Automated and Sustainable Irrigation System using an ESP-8266 module

Georgios Sigalas
School of Computing
University of Portsmouth
Portsmouth, UK
up933287@myport.ac.uk

***Abstract*—Water scarcity is a major issue in the Aegean islands of Greece, particularly in the agricultural sector where water consumption is high. This paper proposes a low-cost IoT solution to tackle the water supply challenges in these regions. The solution utilises an ESP-8266 WiFi module that enables real-time monitoring of soil moisture levels and provides accurate watering based on the specified moisture levels. This approach aims to reduce water consumption, improve crop yields, and promote sustainable agricultural practices in these areas.**

The paper begins by discussing the water supply challenges faced by rural communities and the agricultural sector in the Aegean islands of Greece. It then describes the proposed IoT solution, including the subsystems and sub-modules, the chosen tools and protocols, and the cost estimation of the implementation.

The available literature shows that the proposed IoT solution has the potential to significantly reduce water consumption, improve crop yields, and promote sustainable agriculture practices, making a significant contribution to promoting sustainable agriculture practices and improving the quality of life for rural communities.

***Index Terms*—smart irrigation, MQTT, ESP8266, water management**

I. Introduction

Water scarcity is a major challenge in many rural areas, including the islands of Greece, where the agricultural sector heavily relies on water resources. For example, in Crete, Greece, 84.5% of the water is used for agricultural needs (Chartzoulakis et al., 2001). This exorbitant need for water in addition to the expensive cost of desalination methods (Nthunya et al., 2022), highlights the urgent need to find efficient ways to manage water supply in irrigation systems. In addition, the increased tourist activity in the southern Aegean has increased the demand for potable water which further challenges irrigation efforts (Papapostolou et al., 2020). One solution to reduce water consumption and improve water management is to implement smart irrigation systems based on real-time data on soil moisture levels (Obaideen et al., 2022).

This paper implements a smart irrigation system that can provide accurate and timely watering based on soil moisture levels. The system is comprised of a central hub that manages individual watering nodes, which are equipped with water and power supply interfaces, as well as sensors that detect soil moisture levels. The system communicates through the MQTT protocol, widely used in the IoT industry (Yassein et al., 2017).

This approach focuses on creating a cost-effective and practical solution to the water scarcity problem in rural areas by utilising the generally cost-effective ESP-8266 WiFi module. This smart irrigation system can potentially reduce water consumption, improve crop yields, and promote sustainable agriculture practices. In the following sections, the design will be described, including the decomposition into subsystems/sub-modules, the chosen tools and protocols, and the cost estimation of the implementation.

II. Design

The system's implementation will use the soil moisture level as the deciding factor on when to start pumping water as the available literature (Zhao et al., 2019), (Nigatu et al., 2022), and (Holzman et al., 2014) finds a strong correlation between soil moisture levels and crop yields. It should be noted though, that the estimation of crop yields is a complex task, and depends on a variety of variables such as the crop and soil type, as well as the climate. As such, a production-ready system would be able to gather various data should from each node, by utilising different sensors. These sensors include a soil moisture sensor, a humidity and temperature sensor (such as a DHT11 sensor), and a light sensor (such as the BH1750FVI light sensor).

In addition, the individual nodes should utilise a relay module that controls the watering system and is controllable by a WiFi-capable module.

Furthermore, the module subscribes to a central hub that acts as an MQTT broker. After that, the module should be able to read the raw sensor data and publish them to the broker. It is worth noting, that typically the ESP-8266 module does not include multiple analogue input pins. Hence, since the project requires the analogue input of multiple sensors, a better alternative would be the ESP-32 module. Alternatively, analogue-to-digital converters can be implemented, although because their cost is quite significant, they are not recommended.

Finally, the central hub can be any device that can transmit a WiFi signal and host a docker container that runs an MQTT broker server.

A. Challenges

1) Environmental: Since the system is designed to operate in an outdoor setting, weatherproofing the equipment and elec-

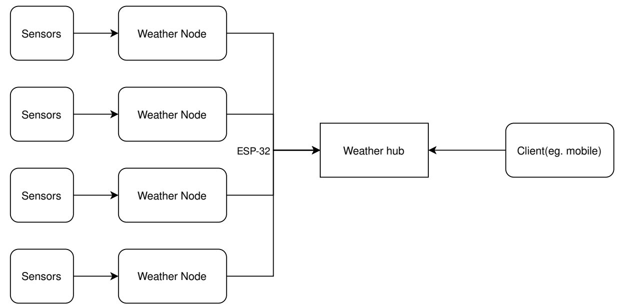

Fig. 1. General design architecture

tronics is a crucial aspect of designing the system. Exposure to rain, dust and extreme temperatures can cause damage to the hardware. As such, it is of great importance to weatherproof the system.

One way to protect the hardware is by using weatherproof enclosures. These are designed to protect the enclosed hardware from dust, moisture, sand, small debris, and water damage.

Another challenge that needs to be considered is the location of the hardware. The weather nodes should be within the radio distance of the hub and also spread apart so that they cover a greater range. In addition, the hardware should avoid being exposed to direct sunlight in order to reduce the risk of overheating.

Finally, the hardware should be mounted securely to prevent them from moving or shaking due to wind or other external forces.

2) Electronics: One challenge when it comes to powering the components is the difference in voltage requirements and electronic noise. For example, the ESP-8266/ESP-32 modules require a 3.3v input and output a low voltage, while the other components require 5v. In addition, depending on the implementation of the watering system more voltage requirements may be needed. For the current needs of the current implementation, a simple voltage step-down circuit from 5v to 3.3v is sufficient. However, if more voltage is required, a more sophisticated transformer may be needed.

3) Assumptions: For the sake of simplicity, the paper will assume that the power and weatherproofing requirements of the system have been met. This could have been done by running power cables along the water pipes or utilising solar panels and rechargeable batteries on each node. In addition, the pumping requirements can also be assumed to have been met. The pumping mechanism could be a sprinkler system in that each individual sprinkler is connected to a node's relay. In either case, the requirements for the voltage that powers the system and the watering interface are assumed to have been met.

B. Equipment

The following components are designed to be used in the system:

- NodeMCU ESP32 - The NodeMCU ESP32 is a low-cost microcontroller that has built-in WiFi capabilities. It also includes 6 ADC pins (4 of which are input-only) that can be used for reading the raw data of the sensors (Amazon.co.uk, 2023d). Alternatively, the NodeMCU ESP8266 module can be used but there needs to be an analog-to-digital converter as it only includes 1 analogue pin.
- Capacitive Soil Moisture Sensor v1.2 - This soil sensor is a low-cost sensor that measures the content of water in the soil providing the moisture level as an analogue output. (Amazon.co.uk, 2023b)
- DHT11 - The temperature and moisture sensor that reads the ambient values. (Amazon.co.uk, 2023c)
- BH1750FVI light sensor - The light sensor that reads the ambient value of light. (Amazon.co.uk, 2023e)
- 1-Channel 5V Relay Module - This relay is used to control the water flow, acting as a switch for the water pump. This module is controlled by the microcontroller. (Amazon.co.uk, 2023a)

C. Subsystems

1) Central Hub: The central hub subsystem is responsible for receiving data from the soil moisture sensors, communicating with the water pump subsystem, and providing a user interface for monitoring and controlling the irrigation system. It consists of a Raspberry Pi 4, an MQTT broker, and a web server.

The Raspberry Pi 4 is used as the central processing unit to manage the irrigation system. A docker container is used as the MQTT broker to handle the communication between the soil moisture sensors and the water pump subsystem. The Mosquitto MQTT broker is selected due to its reliability and widespread use in the IoT industry.

A web server is also implemented on the Raspberry Pi 4 to provide a user interface for monitoring and controlling the irrigation system. The web server displays real-time data from the soil moisture sensors and allows the user to set the threshold for turning on the water pump subsystem.

Pseudo code for the central hub:

```
while True:
    receive data from watering nodes
    receive user commands

    if user commands to turn on pump:
        send message to turn on pump

    if user commands to stop the pump:
        send message to turn off pump

    if user configures a node:
        send configuration information

    if data received:
        save in database
```

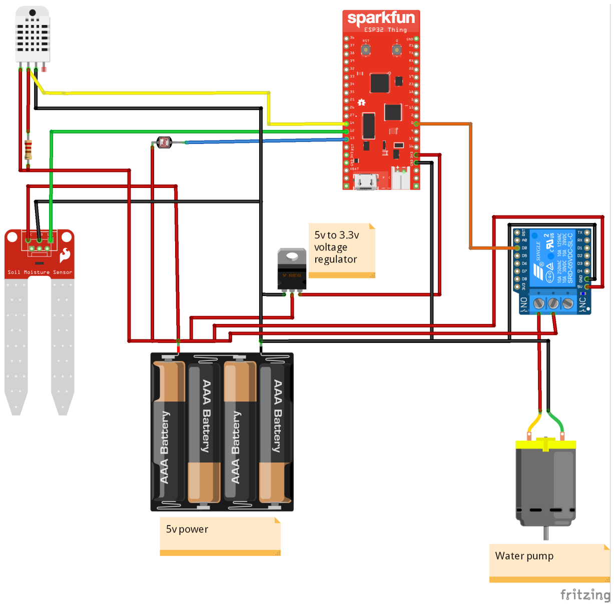

Fig. 2. Wiring of the watering node

```
    if atmospheric data met:
        send message to turn on pump
```

2) Watering node: The watering node is responsible for monitoring the sensor data and controlling the water flow. As seen in figure 2 the watering node is a simple wiring endeavour. Specifically, the components used in the diagram include the soil, temperature, moisture, and light sensors, as well as the ESP32 module, the relay and a voltage regulator to step down the 5v voltage to the module's accepted 3.3v input.

The following code demonstrates the watering node functionality:

```
if configuration data received:
    save configuration data locally
    apply data
    reset module

if atmospheric conditions met:
    turn on relay 1
    wait for 10 seconds
    turn off relay 1
    send sensor data to the central hub
    send watering complete command

if water command received:
    turn on relay 1
    wait for 10 seconds
    turn off relay 1
    send sensor data to the central hub
    send watering complete command

if stop command received:
    turn off relay 1
    send watering interrupted command
```

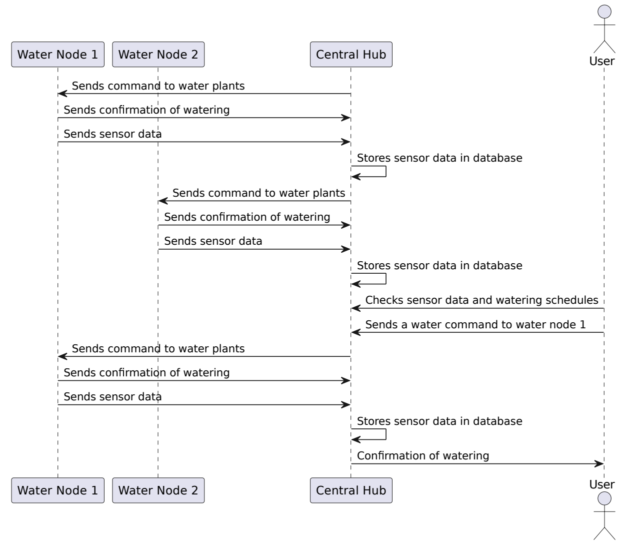

Fig. 3. General data flow

D. Data flow

The data flow of the system is illustrated in figure 3. The sensors are connected to the central hub subsystem through WiFi communication using the ESP32. The central hub subsystem receives data from the sensors and sends commands to the water pump subsystem to turn on or off the water pump. The web server on the central hub subsystem provides a user interface for monitoring and controlling the irrigation system.

E. Cost

The cost of the system electronics is relatively low, totalling £17 for the cost of a water node. It is also worth noting, that this cost estimation does not include any additional costs such as the cost of any tools required for the assembly and installation, and the shipping and handling. In addition, this cost analysis does not include the weatherproofing equipment, the pump, or the power supply, since neither of them has been considered in the design.

Water Node:

- NodeMCU ESP32s x2: £ 14
- Soil Moisture Sensor x6: £ 9
- DHT11 Sensor x5: £ 9.29
- BH1750FVI Light sensor x2: £ 9.29
- 5V Relay Module x4: £ 8

Central Hub:

- Raspberry Pi 4 Model B x1: £ 40

III. Implementation

Even though the design specifies several components, because of the limited resources and time, the implementation shall use slightly different components. Specifically, the implemented system shall use the following equipment:

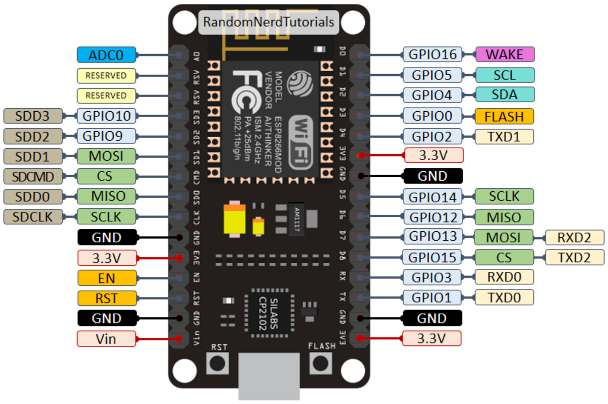

Fig. 4. The NodeMCU pinout

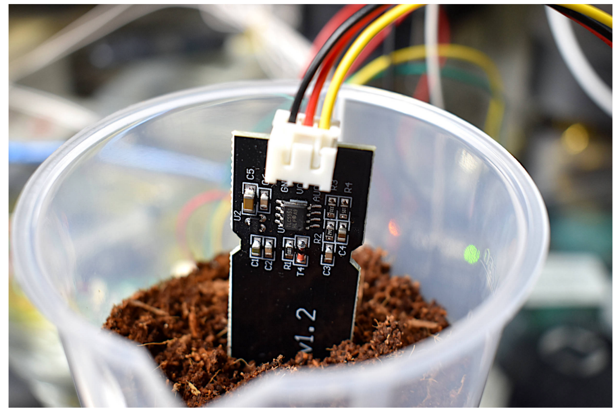

Fig. 5. A capacitive soil moisture sensor, as seen in Hrisko, 2020

- ESP-8266 NodeMCU
- Capacitive Soil Moisture Sensor
- DHT11 sensor
- Photoresistor
- 1-Channel 5V Relay Module
- 6v battery pack
- 4.5v battery pack
- 5v to 3.3v voltage step-down circuit
- 3.3v-6v water pump
- 2N2222A transistor
- LM393 comparator with a potentiometer to convert the analogue signal to digital

A. Equipment

1) ESP-8266 12-E NodeMCU: The ESP8266 12-E NodeMCU is a variant of the popular NodeMCU development board based on the ESP8266 Wi-Fi module. It offers enhanced features and capabilities compared to the standard NodeMCU board. Here's a detailed explanation of the ESP8266 12-E NodeMCU:

ESP8266 Wi-Fi Module: The ESP8266 12-E NodeMCU utilises the ESP8266 Wi-Fi module as its core component. The ESP8266 module integrates a microcontroller unit (MCU) with Wi-Fi connectivity, allowing for wireless communication and IoT capabilities (Systems, 2023).

Microcontroller: The ESP8266 12-E is powered by an ESP8266EX microcontroller, which is based on the Tensilica Xtensa LX106 architecture. It operates at a clock speed of 80 MHz and provides sufficient processing power for IoT applications.

GPIO Pins: The ESP8266 12-E NodeMCU features 16 general-purpose input/output (GPIO) pins, which can be used for digital input/output, analogue input, and various other purposes. These pins allow for easy integration with external sensors, actuators, and other peripheral devices.

USB Interface: The ESP8266 12-E NodeMCU board is equipped with a built-in USB-to-serial chip (such as CH340 or CP2102), enabling easy communication with a computer for programming, debugging, and serial data transfer.

Memory: The ESP8266 12-E NodeMCU offers 4MB (32Mbit) of flash memory for storing firmware, program code, and other data. This ample memory capacity allows for more complex applications and data storage on the board.

Power Supply: The ESP8266 12-E NodeMCU can be powered via a micro USB connector or an external power source connected to the VIN pin. It operates at a voltage of 3.3V, so caution must be exercised to ensure that any external components or sensors used are compatible with this voltage level.

Wi-Fi Connectivity: The ESP8266 12-E NodeMCU provides built-in Wi-Fi connectivity, allowing it to connect to 2.4GHz Wi-Fi networks. This feature enables seamless integration into IoT networks, cloud services, and remote communication with other devices.

Programming and Development: The programming can be done using the Arduino IDE, which supports the ESP-8266 family of modules using the provided Board Manager. Once the module is programmed, it can be used independently.

2) Soil Moisture Sensor: A soil moisture sensor is an electronic device that measures the amount of moisture in the soil. There are various types of soil moisture sensors, but most of them operate on the principle of measuring the electrical resistance of the soil (Dukes et al., 2009). A typical soil moisture sensor consists of two metal probes that are inserted into the soil. These probes are usually made of stainless steel or another corrosion-resistant metal. When the probes are inserted into the soil, the moisture in the soil creates a path of electrical conductivity between the probes.

The sensor contains an electronic circuit that measures the resistance of the soil between the probes. This resistance value is directly related to the moisture content of the soil. When the soil is dry, it has a higher resistance, while when it is moist, the resistance is lower. An example of a soil moisture sensor can be seen in figure 5.

3) Relay module: A 1-channel 5V relay module is an electronic switch that allows a low-voltage signal to control a high-voltage circuit. It has one input channel and is designed to be driven by a 5V DC signal from a microcontroller, such as an Arduino.

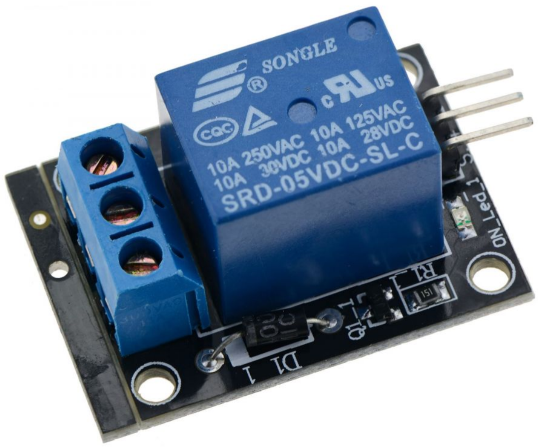

Fig. 6. A 1-channel 5v relay module, as seen in "Switch Electronics", 2023

The relay module consists of a small low-power coil that when energised by the input signal, creates a magnetic field that pulls a metal armature towards the coil, closing a set of contacts in the process. These contacts are designed to handle higher voltage and current levels than the input signal, allowing them to control larger loads such as lights, motors, or appliances.

As can be seen in figure 6 the module typically has three pins: VCC, GND, and Signal. VCC and GND are connected to the power supply, while Signal is connected to the microcontroller output pin. When the microcontroller sends a signal to the Signal pin, it activates the coil and switches the relay contacts on or off depending on the desired operation.

4) AMS1117 Voltage Regulator: The circuit consists of the AMS1117 voltage regulator that steps down the 5v of the power supply to the 3.3v required by the ESP-8266. The AMS1117 is part of the AMS117 series of voltage regulators produced by Advanced Monolithic Systems (AMS) and is a low dropout (LDO) regulator (Advanced Monolithic Systems, 2009).

5) LM393 Comparator: The LM393 is a low-power dual voltage comparator integrated circuit. A voltage comparator is a device that compares two input voltages and outputs a digital signal indicating which one is larger. The LM393 has two independent voltage comparators, and each of them has a non-inverting input (+) and an inverting input (-).

When the voltage at the non-inverting input is greater than the voltage at the inverting input, the output of the comparator goes high (logic level 1). Conversely, when the voltage at the inverting input is greater than the voltage at the non-inverting input, the output of the comparator goes low (logic level 0).

The LM393 is commonly used in applications where a digital signal is required based on a comparison of two input voltages. (Instruments, 2014)

6) 2N2222A transistor: In the circuit setup, the 2N2222A transistor is configured as follows: the emitter is connected to the common ground of the system, the collector is connected to the input signal of the 5V relay and the base is connected to the output signal of the NodeMCU microcontroller board. The primary objective of this configuration is to enable control of the 5V relay using the ground potential of the 5V power supply, as the NodeMCU's output signal operates at a lower voltage level.

By employing the 2N2222A transistor as a switch, the NodeMCU microcontroller can effectively control the state of the relay by modulating the base current. When the NodeMCU output signal is in a high state (logic level "1"), a sufficient base current is supplied to the transistor, causing it to enter the saturation region. This results in a low resistance path between the collector and the emitter of the transistor, allowing current to flow through the relay's input signal and activate the relay.

Conversely, when the NodeMCU output signal is in a low state (logic level "0"), the base current is cut off, driving the transistor into the cutoff region. In this state, the transistor exhibits a high resistance between the collector and the emitter, effectively isolating the relay's input signal from the ground potential, thereby deactivating the relay.

By utilising the 2N2222A transistor as an intermediary between the NodeMCU and the 5V relay, this circuit arrangement allows for the effective control of the relay using a low-voltage signal while leveraging the higher voltage capabilities of the 5V power supply.

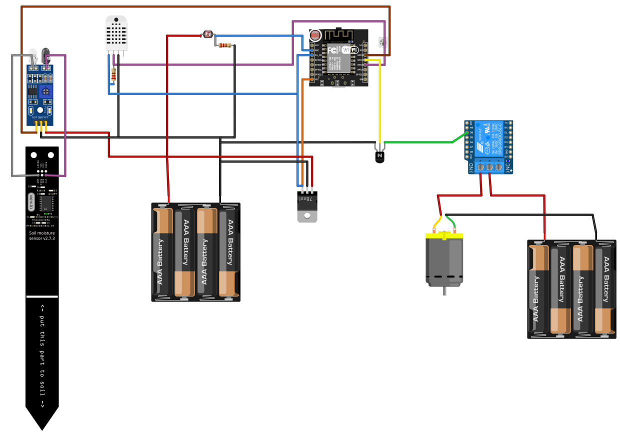

Fig. 7. The current implementation of the water node

B. Water node

Hardware

Input: As can be seen in figure 7, the water node input sensors in the current implementation consist of a soil sensor, a DHT11, and a photoresistor sensor, all of which, are connected to the digital general-purpose i/o pins, besides the photoresistor which is connected to the only analogue pin. All these sensors are responsible for reading the data of the crops' climate and ultimately deciding, whether or not the pump should start.

Output: In addition to all the sensors, the system should be able to start/stop a water pump. This is a slightly different circuit, in which the pump is connected directly to another

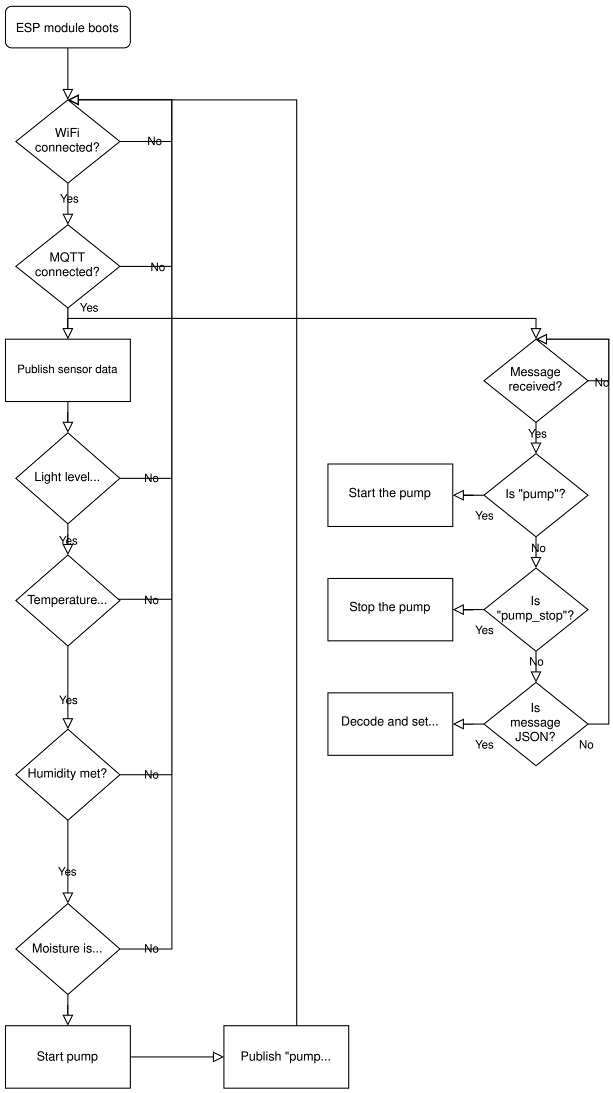

Fig. 8. Water node flow diagram

power supply and the relay module. The reason for this is to avoid the sudden current draw and electrical noise that the pump motor introduces when it starts and stops. Furthermore, the relay module is connected to the 5v power supply and its signal pin is connected to the transistor's collector, which, in turn, is connected to the NodeMCU module which controls the pump by allowing the current to go through the transistor.

Software

The watering node system utilises the Arduino IDE to develop software that manages the sensors and the pump. The main logic of the system can be seen in figure 8, based on this the software can be developed. Please refer to the attached source code file for more information.

Libraries: The code includes various libraries such as DHT (for DHT temperature and humidity sensors), ESP8266WiFi (for ESP8266 Wi-Fi functionality), PubSubClient (for MQTT communication), and ArduinoJson (for working with JSON data).

Custom Library: There is a custom library included named "WiFiConfig.h". This library contains the default configuration of the watering node in a single file.

Constants and Variables: The code defines some constants and variables, including DHT_TYPE (the type of DHT sensor being used), and a boolean variable pumping to track the pump state.

Pin Configuration: The setupPins() function sets the pin modes for various sensors and the pump. It also sets the initial state of the pump to LOW.

JSON Utilities: The code defines a function convertSensorDataToJsonString() that takes sensor data (light level, temperature, humidity, and moisture) and converts it into a JSON string using the ArduinoJson library.

Sensor Logic: The publishSensorData() function takes sensor data, converts it to a JSON string, and publishes it to the MQTT broker using the mqttClient.publish() function.

The setConfiguration() function is responsible for setting the sensor thresholds (light level, temperature, humidity) based on the provided values. It also publishes a message to the MQTT broker to indicate that the configuration is being set.

The turnPumpOn() function turns on the pump by setting the pumping flag to true, activating the pump output pin, delaying for a specified timeout, and then calling the turnPumpOff() function.

The turnPumpOff() function turns off the pump by deactivating the pump output pin and setting the pumping flag to false.

Network: The onMessageReceived() function is the MQTT message callback. It is triggered whenever a message is received from the subscribed topics. It checks the received message and performs actions accordingly. If the message contains "pump", it calls the turnPumpOn() function. If it contains "stop_pump", it calls the turnPumpOff() function. If the message is a JSON message and the topic is "config", it calls the setConfiguration() function with the provided values.

The connectToWifi() function connects the ESP8266 to the Wi-Fi network using the specified SSID and password.

The connectToMQTTBroker() function sets up the MQTT client, configures the MQTT broker server, and sets the callback function for message handling.

The reconnectMqttClient() function attempts to reconnect to the MQTT broker if the connection is lost. It also subscribes to the appropriate MQTT topics after a successful connection.

Main Functionality: The setup() function is the setup routine that initializes the serial communication, sets up the pins, connects to Wi-Fi, and connects to the MQTT broker.

The loop() function is the main loop that runs continuously. It checks if the MQTT client is connected, and if not, it attempts to reconnect. Then it reads sensor data, checks if the conditions for watering are met, and controls the pump accordingly. It also publishes the sensor data using the "publishSensorData"

MQTT Broker client

In addition to the ESP-8266, the system utilises an MQTT Broker that acts as an intermediate between the water node and the client. Currently, the broker is not doing anything other than receiving and sending messages. In the future, this can be enhanced to include saving sensor data locally and providing a web interface to the client.

IV. Discussion

The smart irrigation system designed in this project provides an efficient solution to the water supply challenges that islands in the Aegean, Greece face. The system makes use of IoT technology to monitor soil moisture levels, temperature, and light intensity, among other factors, to ensure optimal irrigation of crops while conserving water resources. The design is practical, and cost-effective, with the potential to benefit farmers in similar regions.

The main insight of the design is that IoT technology can significantly improve agricultural productivity while reducing water consumption. The use of wireless communication and cloud-based data analysis enables real-time monitoring of crop conditions and allows for quick adjustment of irrigation schedules. The chosen solutions, such as the ESP-8266 WiFi module, the soil moisture sensor, and the 1-channel 5V relay module, are the most appropriate for the given scenario due to their affordability, reliability, and compatibility.

The tangible output of the project is the watering node, which includes the ESP-8266 WiFi module, the soil moisture sensor, the DHT11 sensor, the photoresistor and all the other equipment outlined in the Implementation section. The rest of the system, including the central hub and the cloud-based data analysis platform, was not implemented due to time and resource constraints.

A. Pitfalls

One potential pitfall of the design is the reliance on the Internet and cloud services, which may not be reliable or available in remote areas. In such cases, the system's functionality may be limited, and manual intervention may be necessary. Additionally, the implementation of the design may require technical expertise, which may pose a challenge for some farmers.

B. Future work

Despite these challenges, the advantages of the smart irrigation system outweigh the disadvantages. The system improves crop yields, reduces water consumption, and minimises the manual labour required for irrigation. Moreover, the system can be further improved by integrating additional sensors, such as a pH sensor, to optimise crop growth. Finally, a web interface may be implemented to allow for more user-friendly control of the system.

In conclusion, the smart irrigation system designed in this project demonstrates the potential of IoT technology to address water supply challenges in agriculture. The chosen solutions and tools are appropriate for the given scenario, and the system can be further improved by integrating additional sensors and optimising the data analysis platform.

References

Advanced Monolithic Systems. (2009). AMS1117 voltage regulator: Specifications.

Amazon.co.uk. (2023a). Aediko 4pcs relay module dc 5v 1 channel relay board with optocoupler isolation support high or low level trigger. https://www.amazon.co.uk/AEDIKO-Channel-Optocoupler-Isolation-Support/dp/B09M62VCGX/ref=sr_1_4?crid=1O3BRJAJYA3QC&keywords=1-Channel%5C%2B5V%5C%2BRelay%5C%2BModule&qid=1683237642&s=industrial&sprefix=1-channel%5C%2B5v%5C%2Brelay%5C%2Bmodule%5C%2Cindustrial%5C%2C193&sr=1-4&th=1

Amazon.co.uk. (2023b). Aideepen 6pcs soil moisture sensor hygrometer module v1.2 capacitive,analog capacitive soil moisture sensor corrosion resistant module plants watering diy kits. https://www.amazon.co.uk/Aideepen-Hygrometer-capacitive-Capacitive-Corrosion/dp/B08GCRZVSR

Amazon.co.uk. (2023c). Aitrip 5pcs dht11 temperature humidity sensor module digital temperature humidity sensor 3.3v-5v with wires for arduino raspberry pi 2 3. https://www.amazon.co.uk/AITRIP-Temperature-Humidity-Digital-Raspberry/dp/B08FSY6Z6P/ref=sr_1_3?crid=3G57ZV4ZXDO9F&keywords=DHT11&qid=1683237508&s=industrial&sprefix=dht11%5C%2Cindustrial%5C%2C111&sr=1-3

Amazon.co.uk. (2023d). Dubeuyew sp-cow esp32 esp-32s nodemcu-32s esp-wroom-32 development board 2.4 ghz wifi and bluetooth dual cores microcontroller esp-wroom-32 chip for arduino nodemcu (38pin, 2 pcs). https://www.amazon.co.uk/DUBEUYEW-NodeMCU-32S-ESP-WROOM-32-Development-Microcontroller/dp/B0BJPZ7G6D/ref=sr_1_2?crid=2Q3PDHRVLJSVI&keywords=NodeMCU+ESP32&qid=1683237797&s=industrial&sprefix=nodemcu+esp32+%5C%2Cindustrial%5C%2C178&sr=1-2

Amazon.co.uk. (2023e). Youmile 2pack light sensor module chip digital intensity illumination bh1750fvi stable dc 5v light ball tools compatible for arduino. https://www.amazon.co.uk/Youmile-Intensity-illumination-BH1750FVI-Compatible/dp/B07YDH737W/ref=sr_1_3?crid=3JECSMOE1L5IJ&keywords=BH1750FVI&qid=1683237567&s=industrial&sprefix=bh1750fvi%5C%2Cindustrial%5C%2C69&sr=1-3

Chartzoulakis, K., Paranychianakis, N., & Angelakis, A. (2001). Water resources management in the island of crete, greece, with emphasis on the agricultural use. *Water Policy*, *3*(3), 193–205. https://doi.org/https://doi.org/10.1016/S1366-7017(01)00012-5

Dukes, M. D., Shedd, M., & Cardenas-Lailhacar, B. (2009). Smart irrigation controllers: How do soil moisture sensor (sms) irrigation controllers work? *University of*

Florida, Institute of Food and Agricultural Sciences, FL, USA, 1.

Holzman, M., Rivas, R., & Piccolo, M. (2014). Estimating soil moisture and the relationship with crop yield using surface temperature and vegetation index. *International Journal of Applied Earth Observation and Geoinformation*, *28*, 181–192. https://doi.org/https://doi.org/10.1016/j.jag.2013.12.006

Hrisko, J. (2020). Capacitive soil moisture sensor theory, calibration, and testing. *no*, *2*, 1–12.

Instruments, T. (2014). Lmx93-n, lm2903-n low-power, low-offset voltage, dual comparators datasheet.

Nigatu, Z. M., Fan, D., You, W., Melesse, A. M., Pu, L., Yang, X., Wan, X., & Jiang, Z. (2022). Crop production response to soil moisture and groundwater depletion in the nile basin based on multi-source data. *Science of The Total Environment*, *825*, 154007. https://doi.org/https://doi.org/10.1016/j.scitotenv.2022.154007

Nthunya, L. N., Bopape, M. F., Mahlangu, O. T., Mamba, B. B., Van der Bruggen, B., Quist-Jensen, C. A., & Richards, H. (2022). Fouling, performance and cost analysis of membrane-based water desalination technologies: A critical review. *Journal of Environmental Management*, *301*, 113922. https://doi.org/https://doi.org/10.1016/j.jenvman.2021.113922

Obaideen, K., Yousef, B. A., AlMallahi, M. N., Tan, Y. C., Mahmoud, M., Jaber, H., & Ramadan, M. (2022). An overview of smart irrigation systems using iot. *Energy Nexus*, *7*, 100124. https://doi.org/https://doi.org/10.1016/j.nexus.2022.100124

Papapostolou, C. M., Kondili, E. M., Zafirakis, D. P., & Tzanes, G. T. (2020). Sustainable water supply systems for the islands: The integration with the energy problem. *Renewable Energy*, *146*, 2577–2588. https://doi.org/https://doi.org/10.1016/j.renene.2019.07.130

Switch electronics. (2023). https://www.switchelectronics.co.uk/5v-1-channel-relay-board-module

Systems, E. (2023). Esp8266 series. https://www.espressif.com/en/products/modules

Yassein, M. B., Shatnawi, M. Q., Aljwarneh, S., & Al-Hatmi, R. (2017). Internet of things: Survey and open issues of mqtt protocol. *2017 International Conference on Engineering MIS (ICEMIS)*, 1–6. https://doi.org/10.1109/ICEMIS.2017.8273112

Zhao, H., Di, L., Yu, E., Zhang, C., Yu, Z., & Rahman, M. S. (2019). Relationship between soil moisture and crop yield in America during a drought. *AGU Fall Meeting Abstracts*, *2019*, Article B51B-08, B51B–08.