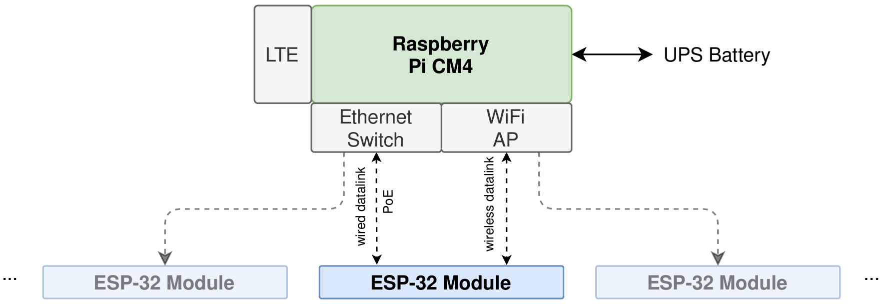
SIWI
Stockholm *Junior* Water Prize
SWITZERLAND

IoT-enabled Unit for Efficient Water Control and Monitoring

Entry to the Stockholm Junior Water Prize 2023



Karim Malik, Frederick Steinmann
International School of Zug and Luzern

Abstract

Our project addresses the urgent need for affordable and accessible water monitoring and management, inspired by the demand at an aquaponics training centre in West Africa. We developed a low-cost, modular, and open platform that leverages single-board computing, mesh networking, and an easy-to-use online interface for cloud monitoring. This innovative solution offers a practical and cost-effective alternative to current systems, empowering communities to take control of their water resources. Our research included a thorough review of literature, state-of-the-art solutions, and network protocols, identifying gaps in implementation and addressing them in our design. During the development process, we benchmarked our system against state-of-the-art multi-thousand-dollar solutions, analysed sensor data, and developed calibration and data analysis software. We exclusively used free and open-source software to ensure easy maintenance and expansion by end-users. Our research included a thorough review of literature, state-of-the-art solutions, and network protocols, identifying gaps in implementation and addressing them in our design.

We combine the ESP-32 and Raspberry Pi microcontrollers with cloud-based databases, offering a practical and affordable option for real-world applications with several benefits. Namely, one can easily add new sensor nodes at a fraction of the cost, using a mesh network for initialization with long-range Wi-Fi and Bluetooth connectivity, at no more than a tenth of the cost of current systems. Additionally, the platform is expandable by end-users and user-friendly due to its open-source nature. Furthermore, the system operates using well-established open-source frameworks, such as Node-RED software and the industry-standard MQTT IoT protocol, ensuring reliable communication and management of sensor readings and actuators.

Our project highlights the urgent need for water quality monitoring and demonstrates a cheaper and more modular solution in comparison to currently available options. Supported by the system's broad applicability, ease of use, and compatibility with various water quality sensors, our applied research project aims to contribute to solving the challenges of food security under water scarcity.

Keywords: Aquaponics; IoT; Water monitoring; Ghana; Cloud-based; Affordable

Acknowledgements

We extend our sincerest appreciation to the individuals and communities who have contributed to the success of this project.

Firstly, we express deep gratitude to the esteemed teacher and mentor, **Mr. Christopher Koch**, for his unwavering dedication to expanding and maintaining the aquaponics system. His expertise and guidance have been invaluable in helping us achieve our goals.

We also extend heartfelt thanks to the wonderful advisor, **Ms. Elisa Bossi**, for her invaluable support and advice throughout this project. Her insights and guidance have been instrumental in shaping this work.

We acknowledge the outstanding support provided by our school, **The International School of Zug and Luzern (ISZL)**, in developing the aquaponics system both locally and internationally. Their generosity and commitment to this project have been instrumental in its success.

We would also like to express sincere appreciation to the open-source community for providing us with the necessary tools and software to achieve these objectives.

Finally, we are grateful to the **Kokrobite Chiltern Centre** non-profit organization for collaborating with us and allowing us to use their system as a testbed. Their involvement has been invaluable.

Contents

Nomenclature

AFE	Analog front end generally refers to a highly sensitive amplifier, which sometimes also digitises an initial signal.
API	An application programming interface is essentially a software library for use with a specific service while abstracting the complexity away.
Baud rate	The baud rate is defined as the rate of bits transferred per second in a serial communication medium.
EMI	Electromagnetic interference is generated by some electrical components in the form of fields, interfering with others and distorting signals.
GPIO	General purpose input/output pins are common on microcontrollers and microprocessors, for interacting or communicating with other components.
IC	Abbreviation for integrated circuit, which is a set of circuits on a solid state semiconductor material.
IoT	Internet of things is a network of interconnected devices, which communicate over the network, exchanging messages with each other.
I^{2}C	I2C, or commonly I^{2}C, is a synchronous, multi controller/multi target serial communication standard for transmitting information between integrated circuits.
Logic level, Serial	In computer logic, logic level voltage refers to the potential difference at which a signal is read to be a “HIGH” or 1 as opposed to a “LOW” or zero. Serial communication, whether wired or not, makes use of this by sending packets of these signals at a previously agreed upon rate.
MCU	Microcontroller unit, in this case usually referring to the main onboard processor of a system-on-chip or single-board computer.
MQTT	MQ Telemetry Transfer protocol (MQ is the name of the IBM product it was developed for). A protocol used for IoT communication between devices.
PCB	A printed circuit board is a piece of electrical equipment that connects various components using copper traces on a non-conducting and usually flame retardant substrate such as FR4. It may have several layers with several embedded integrated circuits.
SoC	A system on a chip is an IC containing all or most elements required for a whole computer.
SPI	SPI, or Serial Peripheral Interface, is a synchronous serial communication standard that utilises a clock signal to synchronise the shifting and sampling of data.
UART	UART, or universal asynchronous receiver-transmitter, is a communication protocol compatible with many different types of serial standards. It is asynchronous, using a previously agreed upon baud rate and an IC’s own oscillator instead of a clock signal to synchronise the sending and receiving devices.
WAP	An access point that wirelessly broadcasts its own LAN, allowing other devices to connect to it. (Wireless Access Point)

1 Introduction

1.1 Context

In 2013, the International School of Zug and Luzern (ISZL) set out to collaborate with the Kokrobite Chiltern Centre, a registered NGO in Kokrobite, Ghana. Kokrobite is a small, traditional fishing village of 8,000 people in southern Ghana, struggling with a shortage of fish due to climate change and foreign companies' over-fishing.

While it's a joyous place, this village is one of many in the coast of Ghana that struggles with an unending cycle of poverty, facing environmental, economic, and social challenges including gender inequality, lack of accessible education, poor sanitation, and a lack of job opportunities, all resulting in stagnant economic growth.

To align with the Ghanaian principle of 'Sankofa', which emphasizes learning from the past while moving forward, the school continues to fund projects in the community. The ultimate objective is to promote the establishment of a self-sustaining local community that does not depend on ongoing funding from affluent nations. ISZL has previously collaborated with the community on various initiatives, such as constructing school buildings, wells, and aquaponics systems.

Since 2018, there has been an effort to implement a fully functioning aquaponics system with several fish tanks, that according to current projections could provide the community with up to 20 tonnes of fish and 50 tonnes of vegetables per annum. Beyond the provision of fresh produce, it can serve as a model and education centre for the local community, providing jobs and economic stimulus.

Aquaponics is a partially closed-loop system consisting of fish, the nitrogen producer, and plants, the nitrogen consumer, where ammonia from waste is converted to nitrite by Nitrosomonas [1]. After this, nitrite is converted to nitrate by Nitrobacter [2], which is in turn delivered to the plants in a circulating water medium.

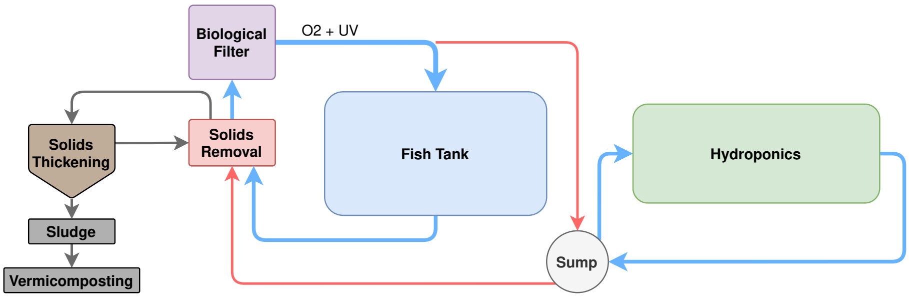


Figure 1 Illustration of a generic aquaponics system [3]

1 shows how water flows through an aquaponics system. In the fish tank, fish consume food and produce waste in the form of ammonia, which is fed to the filter, and then back into the fish tank. This water contains nitrates after filtration, so it is pumped to the plants (the hydroponic element), where the plants absorb the nutrient.

What makes it so well suited to the environment is that clean water is a relatively scarce resource in Ghana, and aquaponics remains closed-loop for water. In fact, it is estimated to be about 90% more water-efficient [4] than conventional farming methods. This is particularly important considering that despite efforts to increase the availability of water, drought is a yearly concern. Crucially, it also allows for year-round production of produce, given somewhat stable exterior temperatures. However, the system relies on a consistent pH and reliable oxygenation such that the plants and bacteria can flourish symbiotically.

The Zug campus setup, which mirrors what was built in Ghana a few years ago, can achieve this easily since the school can afford high-tech industrial-grade equipment to monitor chemical changes in the water and growth medium. For the system in Kokrobite though, several challenges were immediately noticed. Primarily, productivity lagged behind the high-tech system in Zug, and the manually measured chemical values were not ideal for the proliferation of plants and fish. This was a special concern since the imminent expansion of the site in Kokrobite for commercial and training purposes necessitates a well-functioning ecosystem, and measures must be taken to facilitate that.

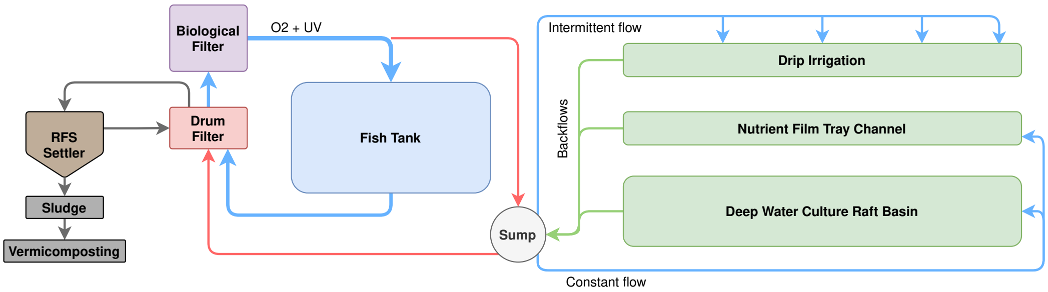


Figure 2 Aquaponics system diagram (Ghana setup) [3]

The layout of aquaponics components can be seen in figure 2, which is specifically the design that is being used in Ghana. As is visible, there are many components or elements involved in the system. Despite the fact that all aquaponics systems rely on the same principle, they can range vastly in the types of configurations they use. For example, there are different types of grow media suitable for different plant types (green elements in figure 2). Due to this range of configurations, it can be difficult to automate systems designed for niche applications, as crucial elements may not be supported by existing control systems available on the market.

An adequate solution from existing providers like Senect® was pursued, which supplied the unit in Zug and many other companies. It was noticed that none of them matched the requirements due to many shortcomings in their designs. In order to find alternative possible solutions in this research work, the following research question was formulated:

> **What specific hardware and software standards are necessary to enable seamless integration of a low-cost and modular aquaponics sensing and actuating unit with existing monitoring systems and databases?**

1.2 The Design Parameters

Before researching possible approaches to solve this problem, the unique challenges of Kokrobite's environment were researched and noted.

1. Due to the remote location of the setup, it is likely that experts will only be able to maintain and visit it in person yearly. It was thus determined that a highly cloud-synchronised approach would allow for the monitoring of readings at all times, and data gathering for training and research purposes. This implies that a cellular modem is needed since the local ISP is not reliable.
2. Because Kokrobite still relies on power from fossil-fuel-powered generators that are highly polluting and prone to failure [5], the power usage of the control system should be minimised where possible.
3. Since the system relies on constant water circulation and aeration performed by air pumps, which in turn

rely on household AC voltage, there must be remote control of relay actuation.

4. As the end goal of this alternative model is aligned with the principle of Sankofa, it is paramount to minimise the cost of such a system. If future expandability is desired, a lower cost is required.
5. Modularity should be a guiding principle, as it allows for future growth and expansion of such a system.
6. It must reliably analyse and present data gathered from an array of sensors on site; including pH, aqueous oxygen, temperature and others. It must also handle measurement errors inherent to pH measurement, along with the need of regular calibration.
7. Because most people maintaining the system will likely have the minimal technical knowledge needed to manipulate the software or hardware, the software frontend must be optimally usable and easy to maintain remotely.
8. In the context of aquaponics, significant deviations in dissolved oxygen or pH can put a fish population in grave danger, consequently, an alarm mechanism must be present to alert supervisors.

1.3 Literature review

The development of the system on the Zug campus was very much guided by the Zurich University of Applied Sciences (ZHAW) in Wädenswil's aquaponics system [6]. It serves as a good example for the educational role such a system can play, serving as a model and educational site for aspiring aquaponics farmers and students alike. Similarly, in the context of central Africa, it may prove particularly useful to provide a hands-on training site, potentially increasing familiarity and inspiring local leaders to pursue this model of farming.

Recent publications and reviews such as Tan *et al.* (2020) [7], and Yanes *et al.* (2020) [8] mention similar solutions to the one discussed in this paper for cloud-synchronised sensor data collection in an aquaponics context. They discuss how MQTT is particularly suitable in an aquaponics context due to its dependable and modular nature, and highlight the potential for total automation.

Common elements from the most recent scientific literature have been incorporated into the design, including but not limited to peer-to-peer communication protocols, automation and data analysis approaches and monitoring techniques. Most literature that is currently present in the field of practical IoT was conceived in a more academic context, while this design, much like work from the MIT Signal Kinetics Lab [9], is targeted towards a real world situation in which one may discover more potential improvements than is possible in an academic context.

The system's design is intended for deployment in Ghana, where unique environmental conditions may affect its functionality beyond laboratory emulation. It is crucial to evaluate not only the impact on community nutrition security but also the economic feasibility, including material reliance for system fabrication, as discussed in Benjamin *et al.* (2021) [10]. Expanding on this, papers such as Kralik *et al.* (2022) [11], and Greenfeld *et al.* (2022) [12] analyse the impact of local aquaponics systems on food security, particularly pertaining to fish and produce production. They conclude a largely positive impact, with noticeable reductions in cost of fresh vegetables [12], further underscoring the potential benefit of lowering the barrier to entry for communities.

The design in Sunehra and Srinidhi (2020) [13] is comparable to the aforementioned one, with both using Node-RED linked to MQTT and a Raspberry Pi, but their solution employs an Arduino Uno (ATmega 328p) for control. However, we improved upon this idea by using a custom circuit board with an Arduino-compatible ESP-32 board, which features network communication permitting more expandability. In comparison, the Arduino Uno has no built-in wireless communication capability. As explained before, other factors such as cost, modularity, and long-term support were taken into consideration, which were not explored enough in the literature compared to the more practical approach that is required in Ghana.

Some papers, such as Karimanzira and Rauschenbach (2019) [14], and Misra *et al.* (2022) [15] shed light on the value of the data gathered as a consequence of constant monitoring. They emphasize the considerable insight gained from large volumes of data, which has already proven itself valuable in an industrial and small-scale settings. In this paper, we show how online databases visualize these data and can then suggest actionable protocols to the user, such as changing growing parameters or improving the water chemistry.

A significant roadblock on the way to affordable water monitoring remains to be the sensors. Currently, research is underway to find more compact and low-cost sensor solutions using Ion-Sensitive Field-Effect Transistors sensors (ISFETs). ISFETs work by detecting changes in the electrical properties of semiconductor material in response to the presence of ions in a solution. The basic structure of an ISFET sensor consists of a field-effect transistor (FET) with an ion-sensitive layer on the gate terminal. Primarily, they are cheaper than conventional glass electrode sensors and feature a quicker response time [16]. Although the technology is still nascent, there have been efforts to achieve greater accuracy than conventional sensors, even enhanced through machine learning in an IoT context [17].

2 Design of the System

The demands imposed by the design parameters led to a first approach, featuring an Arduino Uno, with an ATmega328P as a main computing unit, along with a Fona Sim800l GSM module feeding to a small 3" TFT LCD screen as a user interface. Some preliminary trials with this hardware were conducted, but several limitations of the Arduino Uno were rapidly encountered (specifically with an ATmega328P MCU). While implementing the LCD screen drivers, it was quickly noticed that the functional memory/SRAM (only 2 kB) [18] was insufficient to store the needed global variables and would not allow for more than one sensor to be successfully interfaced. Additionally, the TFT LCD is small and does not provide comfortable amounts of working space needed to implement a user interface.

After this trial, the search for alternative processors began, and several units such as the STM-32, PJRC Teensy, Arduino Yun, ESP-8266, Particle Photon, Arduino Portenta H7, and ESP-32 were considered (see figure 3).

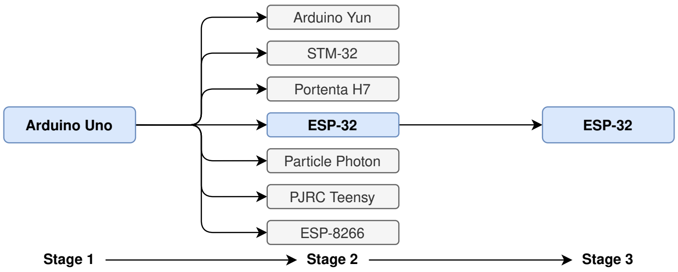


Figure 3 Selection process of micro-controller

After a thorough evaluation of these options, the ESP-32 stood out as especially modular and easily integrable with a highly interconnected IoT environment, being conceived for such applications. In general, IoT applications require diverse wireless communication standards, which the ESP-32 supports. Besides its variety of compatible connectivity methods, it is available in a wide range of computing configurations, like single or dual-core processors and expandable RAM/PSRAM.

For the graphical user interface, driving larger LCD panels from the main microprocessor was considered, but eventually, the aforementioned option was abandoned due to driver complexity and power consumption. Instead, it was opted to run the GUI from a Raspberry Pi, since GNU/Linux-based systems offer better performance and customisability for user interfaces. This also opens the possibility of monitoring data live with a web-based time-series database, which could then directly be displayed as widgets on a screen.

For web-based monitoring of sensor values, at first, a custom solution via Amazon Web Services was considered, which could then be integrated with the Google suite through various APIs such as the Google Drive and Sheets API. This method was previously used to log data, however, it is rather inconvenient as there are quotas on transfer frequency. The connection was unreliable for long-term use due to API key expiration, which is a problem that was encountered when trialing this approach.

Although it was initially planned to integrate additional sensors for water conductivity and nitrate concentration, research and consultation with an aquaponics expert suggested that the added cost of roughly 500 CHF, which is about 23% of the per capita income in Ghana [19], was not worth it. In fact, optimal efficiency can still be achieved as long as nitrite and nitrate measurements are still taken regularly with a testing kit, as these values cannot fluctuate as rapidly as pH or dissolved oxygen. Thus, it was chosen to only implement pH and dissolved oxygen sensors.

The software can run on almost any recent Raspberry Pi model. There are, however, differences between the models in terms of RAM for example, which is particularly important in this case, as it is intended to run many programs simultaneously. An estimated minimum of 2 GB of RAM is sufficient for efficient operation.

A Pi Zero 2W was initially considered for its low cost and small form factor, however, it only features 512 MB of RAM [20], and no native Ethernet support. For development purposes, a Pi model 3B was used, as its specifications are adequate, but the design is now planned to use a Compute Module 4. Compute modules are cheaper to implement in a final networking PCB module, and have a variety of options with different hardware specifications, while still allowing future expandability for any new revisions.

To guarantee wireless connectivity, an initial approach was to use a Fona SIM800L GSM module, however using SMS messaging for communication was rather inefficient at transmitting useful data, so it was decided to use LTE/4G networking as it is more versatile due to its high uplink and downlink speeds. Furthermore, 2G/GSM is nearing its end of life and has been phased out in many countries. In fact, many developing countries omit the installation of legacy networks, instead installing more modern LTE networks. With internet connectivity, cloud-based platforms and messaging methods such as e-mail or Telegram can be used, which are much more intuitive for a user to interact with.

2.1 Shortcomings of Existing Products & Matching Solutions

Costing nearly 3’000 CHF [21], the Senect® Control System (figure 4) supports industrial sensors and functions as expected, however soon it was noticed that it lacked extensibility in software, as only a mobile app is available. This required a very untidy workaround to get data to log into the cloud. After researching, it was noted that using

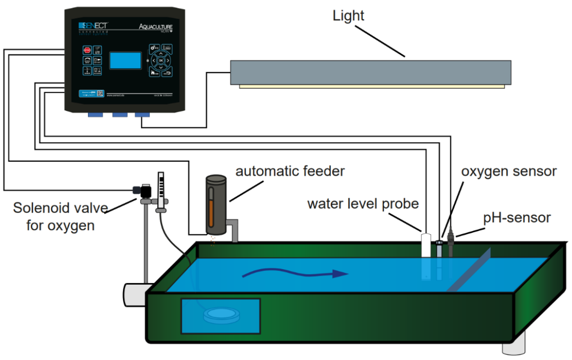


Figure 4 Senect$^{®}$ Control System sample diagram [21]

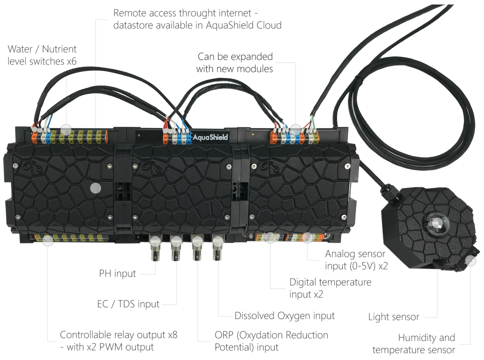


Figure 5 AquaShield sample diagram [22]

Atlas Scientific electrodes would satisfy the needs, as they are reliable and of good quality. However, to interface them with the digital hardware, costly proprietary EZO circuits were required for signal conversion. Their high price (60$ per sensor) made them undesirable for this design as it would demand most of the planned parts budget. Instead it was decided that designing an application-specific analogue front-end (AFEs) implementation was desirable, as existing ICs can fulfill the same role and cost significantly less.

Another alternative control system known as the AquaShield [22] was considered, however, it has disadvantages of its own. While it provides improved modularity, it lacks software extensibility. This is due to the closed-source software, furthermore, it still costs nearly $1000 per unit. The alternative solution that is presented derives its communication hierarchy from the MQTT protocol, which is highly reliable and was originally designed to control oil pipelines from satellites [23]. It has also been applied to IoT and home automation systems [23]. Coincidentally, aquaponics automation is very similar to home automation, so such protocols can be employed. Using this industry-standard method permits the utilization of third-party IoT devices in the system, which would not have been possible without employing a diverse standard.

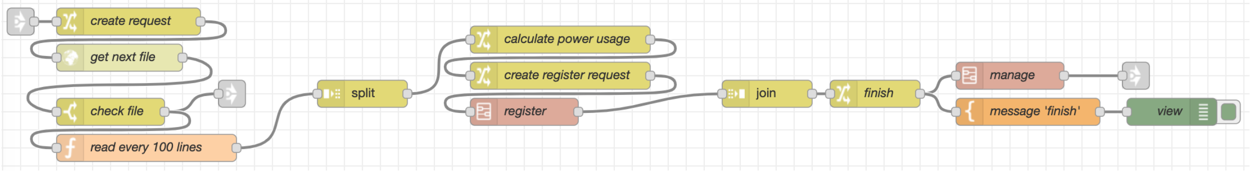


Figure 6 Sample Node-RED flows

Support for well-known software suites such as Node-RED is also included, which is designed for home automation, and was chosen for its intuitive visual scripting capability (see figure 6). In combination with this, Grafana and InfluxDB are included which enable advanced data logging, as this was simply not possible with the Senect$^{®}$ system due to the fact that it was necessary to manually log values into a spreadsheet. Using this software tremendously improves the user experience, as the user is able to easily create automation ‘flows’, which are equivalent to subroutines triggered by events and can then fulfill a wide range of tasks. Effectively, it means that high-level routines and tasks may be implemented without needing to piece code together for APIs. This is a problem for many of the alternatives, as sometimes (like in the aforementioned case of the ISZL campus system) the greenhouse is situated a story above the fish tank, which requires more complex pumping cycles that can lead to flooding if not managed properly. This was possible with the Senect$^{®}$ system, however, it is error-prone and was not user-friendly to set up. This is due to the purposely closed-source code and lack of community support.

It is also important to notice that most aquaponics control systems available on the market are designed for 'urban farming', which, as the name implies, means that the systems are designed to operate in an urban environment, where internet connectivity is accessible and reliable. This can pose many reliability issues in rural or remote locations, especially Kokrobite since internet connections can be dropped and power outages are not a rare occurrence [5]. Therefore, the presented design incorporates GSM and LTE/4G networking capabilities for long-range communication, as well as the option of expanding the system with further sensor modules, which may also be linked wirelessly via Bluetooth or WAN.

2.2 Methodology

The design is intended to be open-source, so that the designs can be improved by other users. Only open-source libraries and platforms were considered. This allows development to be decentralised, and not reliant on a single organization which may cease production, rendering a product effectively useless, contributing to the e-waste problem worldwide and leaving users without a reliable product.

Iterative design was key to the development path that is presented here, with modularity allowing the swift deployment of improvements and changes.

3 Technical Implementation of the System

Fundamentally, the system's design is based on the MQTT network architecture. MQTT (MQ Telemetry Transport) is a message transfer protocol, commonly used in IoT applications due to its high reliability and efficiency.

3.1 MQTT Network Architecture

MQTT utilises a publisher/subscriber model [23], in which all clients communicate with one another by exchanging messages under specific topics. Clients subscribing to a topic will receive messages from other clients publishing to that same topic. All message relaying is handled by an MQTT broker.

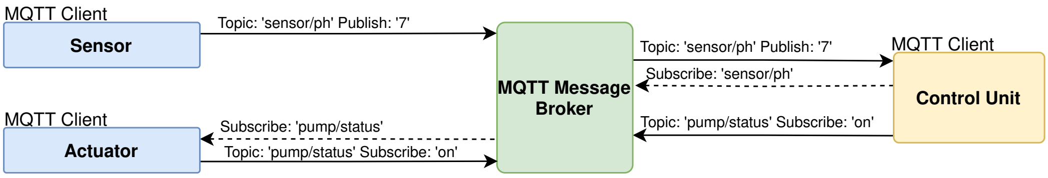


Figure 7 MQTT messages in a simplified aquaponics system

In the context of aquaponics, there are usually three stages to automation. First, there is a control unit that is running a monitoring script. Then, it samples and reads sensors until it can finally respond with actuator events. To monitor sensor values (for example pH as shown in figure 7), the control unit subscribes to any messages under the topic '`sensor/ph`'. The pH sensor then regularly publishes its value under that same topic. These messages are relayed to the control unit through the broker and are then received.

With this information, the control unit is able to respond with its own message. For example, the unit may have logic programmed to turn on the pumps given the sensor value. To do so, it publishes a message under the topic '`pump/status`', which corresponds to the same topic that the pump is subscribed to. In a similar way, the air pump receives the message and alters its state in accordance with the Control Unit's request.

3.2 Advantages of MQTT for IoT

The MQTT protocol offers increased scalability, simplicity, extensibility, and reliability. Increased extensibility is achieved with the use of message topics, which can be assigned multiple levels (‘`level1/level2/level3/...`’). This allows for greater organisation and flexibility when adding new features to a given system, as they categorise data or inputs, making it easier to maintain and less prone to bugs or human error.

Space decoupling is an important aspect of the protocol, which means that the publishing and subscribing clients do not need to know each other’s endpoints (ip addresses/ports), rather they just need to know the topic, while the rest is handled by the protocol thus simplifying development and maintenance. Time and synchronisation decoupling are also aspects that allow for increased reliability during message transmission as messages can be received asynchronously.

Network reliability is also ensured by the protocol, as various ‘Quality of Service’ (QoS) levels are available (0, 1, 2), where level 2 ensures that the message is delivered exactly one time, which is at the cost of increased network traffic. Usually, QoS is only an issue with thousands of devices on a single network. However, in the context of Ghana’s relatively weak infrastructure, as is described in [5], hedging against such occurrences is good practice.

Still, such a guarantee is essential, especially when dealing with actuators in an aquaponics system. For example, if a signal to turn a pump off is not received, it is possible for a disaster to occur. To combat unreliable networks where connections may be dropped unexpectedly, the ‘Last Will and Testament’ (LWT) is used. It broadcasts a message on disconnection, which can perform emergency actions to prevent further problems, improving the fault tolerance of any system. For example, the ‘will’ of the control unit could be to disable any actuators, thereby mitigating any risk associated with them (i.e. flooding, pump overheating, etc.).

3.3 MQTT Implementation

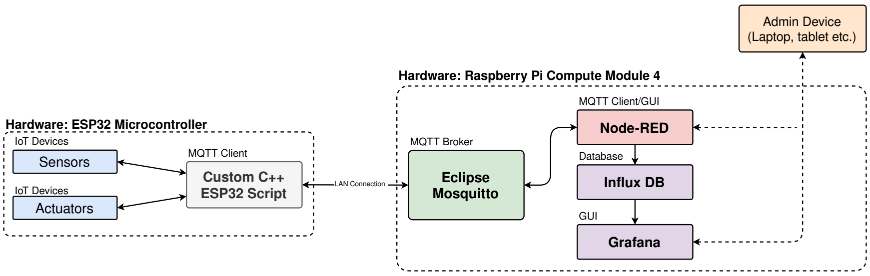


Figure 8 MQTT messages in a simplified aquaponics system

The design uses a variety of software to control the aquaponics system via MQTT messages. It consists of two main components, the ESP-32 microcontroller commonly used in IoT applications, which was chosen for its extensibility and support, as well as its small footprint. This component is tasked with interacting with the sensors directly and allowing them to subscribe and publish messages over the network. All of this takes place on a single board with modular sensor/actuator connectors. Next, there is the Raspberry Pi module, which acts as a hub and can be connected over a LAN/WAN. It runs software capable of controlling and scheduling actuators based on sensory input, operating as an IoT agent.

In figure 8, it can be observed that the software running on the ESP32 is a custom script. This interfaces the low-level processing of the sensors and actuators with the higher-level MQTT messages, allowing sensors and actuators to subscribe to topics. The ESP32 then communicates its messages over Ethernet, connecting to the Mosquitto MQTT broker server developed by the Eclipse Foundation, which is running on the Raspberry Pi. This broker was specifically chosen for its lightweight implementation.

On the Raspberry Pi, the Node-RED client provides logic control over the system, using an intuitive and easy-to-use graphical programming language. It is then linked with the InfluxDB database, specifically designed for use with time-series data. This data is then processed by Grafana, which provides useful insight into the data analytics aspect of the system. All this information can be accessed over the LAN through a computer/smartphone/tablet through a web GUI.

3.4 Debian GNU/Linux System Configuration

In order to simplify and reduce the cost of the system, the networking or routing functions are handled by the Raspberry Pi, using the `hostapd` SystemD service, which allows for the ESP32 client to connect to the broker over Ethernet or WiFi. Mosquitto, Node-RED, InfluxDB, and Grafana are also run as SystemD services since this allows them to start upon boot.

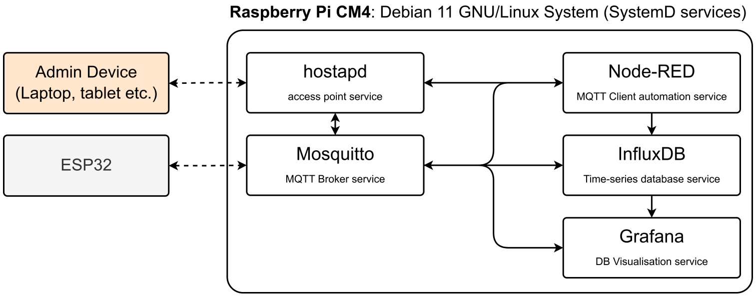


Figure 9 Software configuration on GNU/Linux system

The hosted access point can then be accessed using a laptop to access the web GUI (see figure 9).

3.5 ESP-32 Low-level Computing Configuration

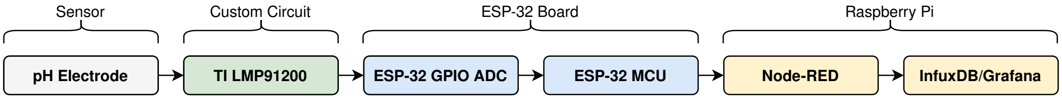


Figure 10 The hierarchical control structure

More generally, the top of the ‘low-level’ computing hierarchy is the ESP32-S3-WROOM2 3.3 V SoC. It has a maximum power draw of 260 mA at its peak clock rate of 240 Mhz [24], and features 32 M Bof Flash as well as 8 MB of PSRAM, which is more than sufficient for this use case.

Since the ESP-32 module only handles the low-level sensor and actuator algorithms, it does not need extreme computing power. Still, to ensure system reliability, a reasonable amount of compute headroom is beneficial. It is programmed in `C++`, and since the range of diverse Arduino libraries have all been ported to the ESP-32 architecture, expansion, and complex functionality is simpler to implement.

Its SoC contains a dual-core Xtensa® 32-bit LX7 microprocessor [24], onboard RAM and PSRAM, a Bluetooth 5 module, and an 802.11 standard WiFi antenna. The exploitation of full dual-core computing power is possible with libraries like FreeRTOS, which dramatically improves computing power by allowing multi-tasking with dynamic memory allocation. Furthermore, its GPIO supports most industry standard communication protocols like UART, I2C, PWM, and SPI. This permits the previously mentioned control and actuation of diverse devices since most analog or embedded devices communicate with these standards.

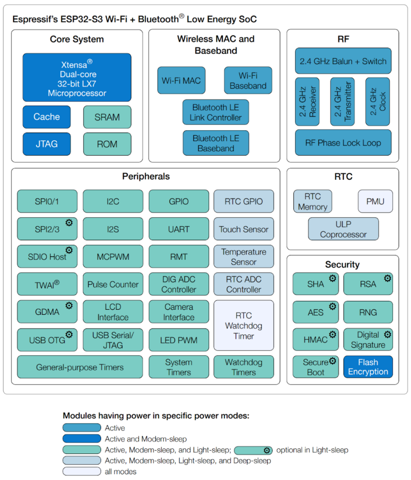


Figure 11 The components of an ESP32-S2-WROOM2 SoC [24]

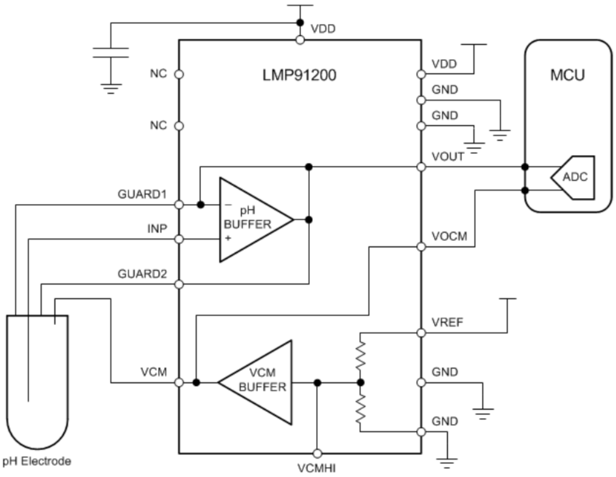


Figure 12 The peripheral circuitry of the AFE

The ESP-32 itself does not handle the raw sensor inputs, that is instead handled by a Texas Instruments LMP-91200 AFE integrated circuit (see figure 12), which consists of a high fidelity amplifier highly suited for the low potential difference created by an electrode, as well as its high impedance [25]. The signal from this amplifier is then passed on to the ESP-32 with the SPI serial protocol for temporary storing. It is then transferred via MQTT over Ethernet or WAN to the Raspberry Pi.

While integrating the AFE circuits on a PCB it was necessary to adhere to good design practices, since weak signals may be corrupted easily by electromagnetic interference from other traces [26]. Such practices include using a separate ground plane for the AFE and system backplane, preventing ground loops from forming, and removing noise on sensor lines. Further, routing clock or logic-level high-frequency signals away from analog signals is highly important. [26].

Broadly speaking, the advantages of this integrated circuit to the Atlas Scientific EZO circuit are reduced cost, higher interoperability with different electrodes, and better open-source documentation.

For the sake of modularity, a PCB with an ESP-32-S3-WROOM2 module and an LMP-91200 coupled to sensors and peripheral circuitry comprises its own self-contained unit capable of measuring values and actuation, therefore it can be independently added to the network architecture, provided it has a power supply. The estimated cost for one such module, without the sensors, is ≈30 CHF at current market prices.

Table 1 Bill of materials for one node

ESP-32-S3	4$
LMP-91200	6$
Various capacitors, resistors	5$
DC/DC converters, ports	10$
Assembly	5$
Estimated total	≈30$

3.6 Technical Attributes of the Raspberry Pi

In the current version of the setup, we utilise a Raspberry Pi 4 compute module as the MQTT broker host or network access node. It is a single-board-computer powered by a Broadcom BCM2711 quad-core Cortex-A72 64-bit SoC with a 1.5 GHz maximum clock rate [27], and in this expected configuration it will have 4 GB of RAM, although trials have proven that 2 GB would be sufficient. For future-proofing and greater overhead, however, more RAM is desirable. It is also connected to the internet with a 3G/4G cellular modem, allowing fast uplink and downlink.

To establish an interface, it runs a Debian-based GNU/Linux distribution, which allows it to display a dashboard from Grafana on a monitor. It can also display these dashboards from a wireless access point. Moreover, due to its ability to run full-featured desktop applications, it is possible to remotely access the device via secure shell and upload a new sketch to the ESP-32 boards with `ESPtool` via MQTT or UART for remote software deployment. [28]

3.7 Modularity of Software & Hardware

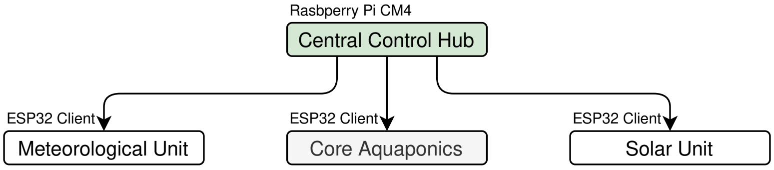


Figure 13 Future expandability of the system

The Raspberry Pi, acting as a hub for all connections, allows for the possibility of creating new ESP32 MQTT client modules with different sensor modules, meaning the aquaponics system can be expanded with new features as illustrated in figure 13.

3.8 User Interface/User Experience Design (UI/UX)

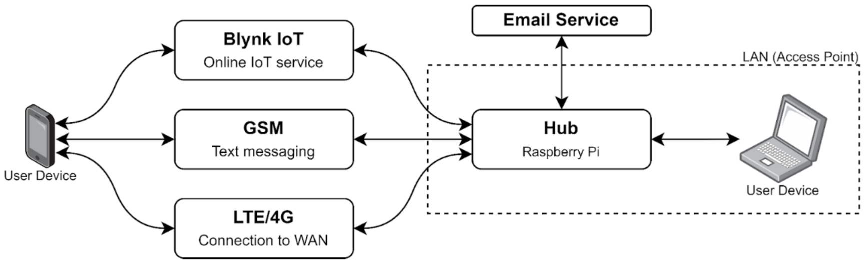


Figure 14 User connectivity and interaction with the system

On a day-to-day basis, the aquaponics system must be monitored. Since the Raspberry Pi ‘hub’ broadcasts its own wireless access point, it can be connected to using a laptop (see figure 14), and the Node-RED and Grafana web-GUI interfaces can be accessed. Through GSM and LTE, the Raspberry Pi is also able to send alerts through SMS or use BlynkIoT, which allows for off-site monitoring and alerting (see figure 14). In a practical case, this necessitates push notifications being relayed to nearby authorized personnel in the case that some sensor metric is out of range, such that a crisis can be averted. Both Grafana and Node-RED have features for email alerting too, which are used for more comprehensive reporting to multiple recipients.

4 Results & Discussion

After elements of the design were implemented, it could be seen that key goals such as affordability and modularity, which are outlined in the hypothesis, were reached. In regard to software, it was possible to run all the required software on the Raspberry Pi model 3B, with memory consumption not exceeding 250 MB at peak load, giving plenty of headroom for additional software, which allows for software upgrades in the future.

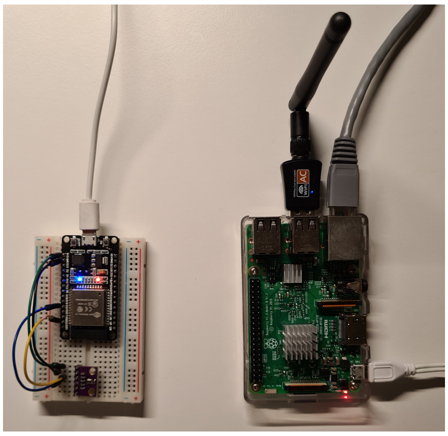


Figure 15 Pi 3B, ESP-32, BME280

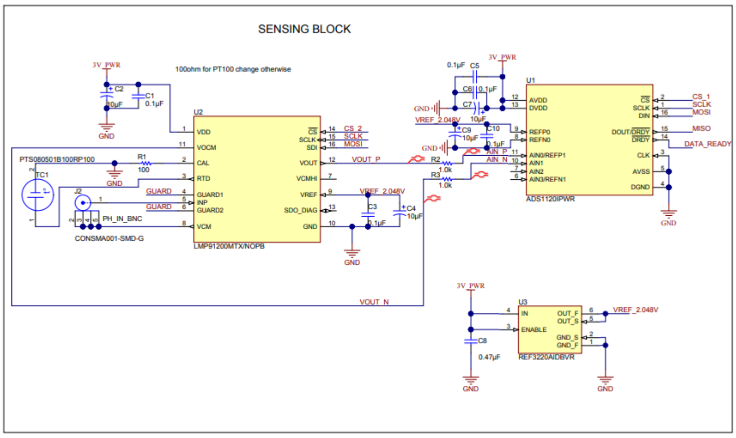


Figure 16 LMP 91200 Sensing block schematic

The setup included a Raspberry Pi 3B broadcasting its own WAP to which the ESP-32 microcontroller connects using its onboard WiFi. For testing, a BME280 sensor was connected to the ESP-32 using I²C to verify that data could be broadcast over the network using the MQTT protocol. To implement this in `C++` the `PubSubClient` library was used as well as the `ArduinoJSON` library to streamline the process of development, mitigating any reliability issues with JSON parsing as this functionality is part of the library.

A Raspberry Pi Zero 2W was also trialed since it can serve as an even more cost-effective single-board computer. However, its limitations did have a noticeable impact on the performance of the software. Notable latency was present when connecting to the web clients and accessing the database. This occurred with data acquired in a time of only ca. 3 hours, which indicated that this model would likely be unsuitable for storing volumes of data acquired over a few days.

```
void setup() {
  Serial.begin(115200);

  mesh.setDebugMsgTypes(ERROR | STARTUP | CONNECTION);
  mesh.init(MESH_PREFIX, MESH_PASSWORD, MESH_PORT, WIFI_AP_STA, 6);
  mesh.onReceive(&receivedCallback);

  mesh.stationManual(STATION_SSID, STATION_PASSWORD);
  mesh.setHostname(HOSTNAME);

  mesh.setRoot(true);
  mesh.setContainsRoot(true);
}
```

Data were gathered from the BME280 sensor (see figure 15), namely temperature, humidity, and pressure. Low-level sensor interfaces worked reliably, and data could be stored in the InfluxDB database for future time-series analysis.

The above snippet of `C++` code was used to initialise the *mesh network* on ESP-32 modules, using the `painlessMesh` library. It permitted setting up a secure, self-repairing peer-to-peer communication environment. A new node is set up by entering the unique `MESH_PASSWORD` to authenticate, which may be performed remotely via `ArduinoOTA` [18].

`C++` code is implemented to allow calibration of the pH reading with an available reference or buffer solution. It uses the analog reading that has been processed by the LMP-91200. In principle, the probe's measurement is carried out using the Nernst equation.

All low-level code and processing functioned reliably on an ESP-32 node. Peak power draw did not exceed 300 mA under peak operating loads, and a considerable amount (50&+) of compute headroom was present, even while handling multiple sensor inputs.

The desired attributes listed in the research question and design requirements were reached; a low cost, being 90% cheaper than the cheapest comparable unit (at 30 CHF), and full software and hardware modularity as measured by the ease of introducing new nodes to the system.

5 Conclusion

In summary, this paper has demonstrated a design for a low-cost and accessible aquaponics sensor system that can overcome the shortcomings of comparable systems on the market such as the Senect$^{®}$ system which served as the inspiration for the design described in this paper. The results discussed in this paper positively indicate the feasibility of implementing such a system using commonly available components. The unique aspect of this design is its modularity, which enables it to adapt more effectively to different environments, far beyond Ghana. Generally, all communities can profit from the flexible platform provided by our system to implement water management systems. Residual chlorine sensors and other relevant graywater sensors are easily integrated with this framework for monitoring, allowing it to be easily repurposed as a more general water management system.

The potential social impact of this design would primarily include reduced water consumption for agriculture, leading to a higher proportion of clean water available for sanitation. Further, increased economic opportunities in the community follow from the scalability and profitability of aquaponics systems [29]. Societally, it could contribute to lower costs and better overall nutrition in the community by providing year-round fresh produce. Through 2023, the Kokrobite Chiltern Centre will commence training interested locals on newly constructed tanks and the above-described technology. The process of training is highly important to guarantee success in a sustainable way. Implementation will likely not proceed flawlessly, but the evolution of this system will be closely supervised and modified according to local community needs and feedback.

Likely, this will become a community project maintained by a large group of people, spreading to other villages as the Kokrobrite Chiltern Center trains volunteers and future leaders. The potential spread of aquaponics on a larger scale in West Africa and less developed areas, in general, will be supported by this platform and its robust scalability and affordability (table 1).

6 Outlook

Since the technical framework of the implementation easily permits expandability due to its wide array of communication standards, its functionality is easily added to with a range of ‘modules’. To increase the reliability and uptime of a system, we are planning the integration of an uninterrupted power supply (UPS) module, ensuring the system is able to monitor the aquaponics system at all times in case of power outages. These occur frequently in rural or remote locations. In the worst case, this could lead to a shutdown, requiring manual intervention to reboot critical systems, jeopardising the fish. To avoid this, the Raspberry Pi broker module has an attached battery that is constantly being charged when mains power is available and switches on instantaneously when an outage is detected. To further increase reliability, the cables connecting to ESP-32 sensor nodes can be routed through the UPS module (see figure 18). Since the latest ESP-32-S models feature ultra-low-power coprocessors, they can operate as a finite state machine to report critical out-of-range measurements of water quality while being supplied a fraction of a mAh from the UPS. This UPS will also be integrated on the actuator/pump level with batteries that are charged by solar panels.

The `ESP-OTA` (over the air) library from Espressif allows remote system flashing, enabling the modules to be updated during operation which can reduce downtime during bug fixes/updates. Essentially, it allows for continuous development and implementation. If an end user wants to add a low-level function on the nodes, all of them can be flashed by the Raspberry Pi with new software without having to interact with every single node.

A meteorological monitoring module can be added simply to the system to monitor weather conditions such as temperature, humidity, and sunlight intensity. Over a long period of time, these data provide big data analytical insights necessary for commercial applications and planting schedules [15][30]. For monitoring plant growth, cameras could be installed above the plants and connected to the Raspberry Pi. Machine learning models can then extract data about plant size or health and log it, which was demonstrated in low-power environments [9].

A diversification of cultivation methods is also possible. A new aeroponics system could be controlled with its own associated sensors and actuators, not necessarily requiring the purchase of new sensors, enabling the system to grow crops, specifically tubers, which are particularly suited to aquaponics [31].

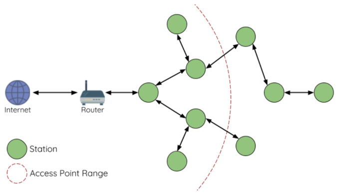


Figure 17 The principle of an interconnected mesh network [32]

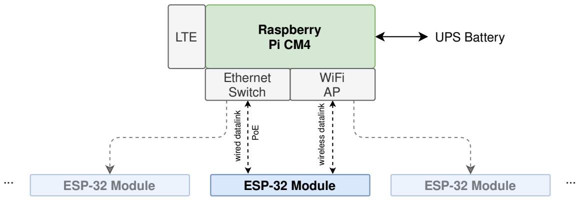


Figure 18 Implementation of the uninterrupted power supply

Should the local community desire to expand the farming unit far beyond its current scope, the capability to do so is maintained due to the aforementioned communication standards. Namely, ESP-32 units reading sensor values can be incorporated into a mesh network [24] with several nodes connected to one root node that is communicating with a router/root node, in this case, the Raspberry Pi. This network can be operated over relatively large distances between individual nodes, making even decentralised expansion past the range of the initial access point possible. It is self-healing, meaning it preserves system integrity when a node is compromised by using an alternate communication path. If a critical software error occurs, the `ArduinoOTA` libraries have built-in rollback safety.

7 References

[1] D. J. Arp, L. A. Sayavedra-Soto, and N. G. Hommes, "Molecular biology and biochemistry of ammonia oxidation by nitrosomonas europaea," *Archives of Microbiology*, vol. 178, no. 4, pp. 250–255, Oct. 2002, ISSN: 1432-072X. [Online]. Available: `https://doi.org/10.1007/s00203-002-0452-0`.

[2] B. Boon and H. Laudelout, "Kinetics of nitrite oxidation by Nitrobacter winogradskyi," *Biochemical Journal*, vol. 85, no. 3, pp. 440–447, Dec. 1962, ISSN: 0006-2936. [Online]. Available: `https://doi.org/10.1042/bj0850440`.

[3] R. Känzig, 2016.

[4] J. M. Ebeling and M. B. Timmons, "Recirculating aquaculture systems," *Aquaculture production systems*, pp. 245–277, 2012.

[5] E. Owusu-Adjapong, "Dumsor: Energy crisis in ghana," *Unpublished paper, Stanford University*, 2018.

[6] A. Graber, N. Antenen, and R. Junge, "The multifunctional aquaponic system at zhaw used as research and training lab," en, in *Prenos inovacij, znanja in izkušenj v vsakdanjo rabo : zbornik izvlečkov = transmission of innovations, knowledge and practical experience into everyday practice : book of abstracts*, M. A. Maček, M. Maček Jerala, and M. Kolenc Artiček, Eds., 3rd Conference VIVUS, Conference on Agriculture, Environmentalism, Horticulture, Floristics, Food Production and Processing, Strahinj, Slovenia, 14–15 November 2014, Strahinj: Biotehniški center Naklo, 2014, pp. 245–255, ISBN: 9789619356432. [Online]. Available: `https://digitalcollection.zhaw.ch/handle/11475/2518`.

[7] E.-K. Tan, Y.-W. Chong, M. Niswar, B.-Y. Ooi, and A. Basuki, "An iot platform for urban farming," in *2020 International Seminar on Intelligent Technology and Its Applications (ISITIA)*, 2020, pp. 51–55.

[8] A. R. Yanes, P. Martinez, and R. Ahmad, "Towards automated aquaponics: A review on monitoring, iot, and smart systems," *Journal of Cleaner Production*, vol. 263, p. 121 571, 2020, ISSN: 0959-6526. [Online]. Available: `https://www.sciencedirect.com/science/article/pii/S0959652620316188`.

[9] Y. Zhao, S. S. Afzal, W. Akbar, *et al.*, "Towards battery-free machine learning and inference in underwater environments," in *Proceedings of the 23rd Annual International Workshop on Mobile Computing Systems and Applications*, ser. HotMobile '22, Tempe, Arizona: Association for Computing Machinery, 2022, pp. 29–34, ISBN: 9781450392181. [Online]. Available: `https://doi.org/10.1145/3508396.3512877`.

[10] E. O. Benjamin, G. R. Buchenrieder, and J. Sauer, "Economics of small-scale aquaponics system in west africa: A sanfu case study," *Aquaculture Economics & Management*, vol. 25, no. 1, pp. 53–69, 2021. [Online]. Available: `https://doi.org/10.1080/13657305.2020.1793823`.

[11] B. Kralik, F. Weisstein, J. Meyer, K. Neves, D. Anderson, and J. Kershaw, "From water to table: A multidisciplinary approach comparing fish from aquaponics with traditional production methods," *Aquaculture*, vol. 552, p. 737 953, 2022, ISSN: 0044-8486. [Online]. Available: `https://www.sciencedirect.com/science/article/pii/S0044848622000679`.

[12] A. Greenfeld, N. Becker, J. F. Bornman, S. Spatari, and D. L. Angel, "Is aquaponics good for the environment?—evaluation of environmental impact through life cycle assessment studies on aquaponics systems," *Aquaculture International*, vol. 30, no. 1, pp. 305–322, Feb. 2022, ISSN: 1573-143X. [Online]. Available: `https://doi.org/10.1007/s10499-021-00800-8`.

[13] D. Sunehra and M. Srinidhi, "Implementation of smart urban farming using raspberry pi, arduino and node-red platform," in *2020 IEEE International Conference for Innovation in Technology (INOCON)*, 2020, pp. 1–6.

[14] D. Karimanzira and T. Rauschenbach, "Enhancing aquaponics management with iot-based predictive analytics for efficient information utilization," *Information Processing in Agriculture*, vol. 6, no. 3, pp. 375–385, 2019, ISSN: 2214-3173. [Online]. Available: `https://www.sciencedirect.com/science/article/pii/S2214317318303871`.

[15] N. N. Misra, Y. Dixit, A. Al-Mallahi, M. S. Bhullar, R. Upadhyay, and A. Martynenko, "Iot, big data, and artificial intelligence in agriculture and food industry," *IEEE Internet of Things Journal*, vol. 9, no. 9, pp. 6305–6324, 2022.

[16] L. A. V. Zorrilla and J. G. L. Calvo, "Monitoring system for isfet and glass electrode behavior comparison," *2017 IEEE XXIV International Conference on Electronics, Electrical Engineering and Computing (INTERCON)*, pp. 1–4, 2017.

[17] N. Sahu, R. Bhardwaj, H. Shah, R. Mukhiya, R. Sharma, and S. Sinha, "Towards development of an isfet-based smart ph sensor: Enabling machine learning for drift compensation in iot applications," *IEEE Sensors Journal*, vol. 21, pp. 19 013–19 024, 2021.

[18] Arduino, *Arduino uno r3 reference manual*. [Online]. Available: `https://docs.arduino.cc/resources/datasheets/A000066-datasheet.pdf`.

[19] Statista, *Income per capita in ghana 2017-2020*, 2023. [Online]. Available: `https://www.statista.com/statistics/1291251/gross-national-income-per-capita-in-ghana/`.

[20] Raspberry-Pi-Trading-Ltd, *Raspberry pi zero 2w*. [Online]. Available: `https://datasheets.raspberrypi.com/rpizero2/raspberry-pi-zero-2-w-product-brief.pdf`.

[21] SENECT-aquaculture-technology, *Aquaculture|control pro*. [Online]. Available: `https://produkte.senect.de/SENECT-AQUACULTURECONTROL-PRO`.

[22] AquaShield-Control, *Aquashield 2.0*. [Online]. Available: https://www.aquashieldcontrol.com.

[23] MQTT-OASIS, "Mqtt software," [Online]. Available: https://mqtt.org.

[24] Espressif-Systems, *Esp32-s3-wroom2 datasheet*. [Online]. Available: https://www.espressif.com/sites/default/files/documentation/esp32-s3-wroom-2_datasheet_en.pdf.

[25] Texas-Instruments, *Lmp91200 configurable afe for low-power chemical-sensing applications*. [Online]. Available: https://www.ti.com/lit/ds/symlink/lmp91200.pdf?ts=1676150165443%5C&ref_url=https%5C%254A%5C%252F%5C%252Fwww.ti.com%5C%252Fproduct%5C%252FLMP91200.

[26] B. R. Archambeault and J. Drewniak, *PCB design for real-world EMI control*. Springer Science & Business Media, 2013, vol. 696, p. 5.

[27] Raspberry-Pi-Trading-Ltd, *Raspberry pi compute module 4*. [Online]. Available: https://datasheets.raspberrypi.com/cm4/cm4-datasheet.pdf.

[28] Espressif-Systems, *Esptool*. [Online]. Available: https://github.com/espressif/esptool.

[29] R. H. Bosma, L. Lacambra, Y. Landstra, *et al.*, "The financial feasibility of producing fish and vegetables through aquaponics," *Aquacultural Engineering*, vol. 78, pp. 146–154, 2017, ISSN: 0144-8609. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0144860917300821.

[30] R. H. Ip, L.-M. Ang, K. P. Seng, J. Broster, and J. Pratley, "Big data and machine learning for crop protection," *Computers and Electronics in Agriculture*, vol. 151, pp. 376–383, 2018, ISSN: 0168-1699. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0168169917314588.

[31] J. Pasch, S. Appelbaum, H. W. Palm, and U. Knaus, "Growth of basil (ocimum basilicum) in aeroponics, drf, and raft systems with effluents of african catfish (clarias gariepinus) in decoupled aquaponics (s.s.)," *AgriEngineering*, vol. 3, no. 3, pp. 559–574, 2021, ISSN: 2624-7402. [Online]. Available: https://www.mdpi.com/2624-7402/3/3/36.

[32] Espressif-systems, *Esp-wifi-mesh*. [Online]. Available: https://docs.espressif.com/projects/esp-idf/en/latest/esp32/api-guides/esp-wifi-mesh.html#esp-wifi-mesh.

8 Appendix

Open-source Projects Used

Name	**Category**	**Brief Description**	**License Type**
Node-RED nodered.org	Automation control	Graphical scripting MQTT compatible automation software for IoT devices.	Apache 2.0 License
Eclipse Mosquitto mosquitto.org	MQTT broker	Lightweight MQTT broker.	EPL-2.0 License
Influx Database influxdata.com	Database	A database designed specifically for time-series based data (TSDB).	MIT License
Grafana grafana.com	Data visualisation	Software for advanced graphical visualisation and alerting.	AGPL-3.0 License
Pi OS Lite (Debian) raspberrypi.com/software	GNU/Linux OS	Lightweight operating system for the ARM based Raspberry Pi.	MIT License, GPL License
FreeRTOS freertos.org	Hardware resource management	Software for utilising multi-core processor functionality as part of a real-time operating system.	MIT License
painlessMesh MQTT gateway github.com/latonita/painlessMeshMqttGateway	Software control	Software for interfacing IoT/ESP-WIFI-MESH with MQTT broker.	MIT License