CHAPTER II
LITERATURE REVIEW

2.1 Previous Research

Table 2. 1 Previous Research

No	Author	Title	Key Findings	Relevance & Research Gap
1	Damayanti et al.	Prediction of Robusta green bean coffee moisture content based on bioelectric properties with artificial neural network method	High accuracy (R = 0.991; MSE = 0.1047); non-destructive method	Demonstrates the potential of ANNs for predicting moisture content, but has not yet been integrated with IoT and real-time systems.
2	Jakkaew et al.	Adata-driven approach to improve coffee drying: Combining environmental sensors andchemical analysis	A significant correlation between drying conditions and chemical quality (caffeine, chlorogenic acid)	Using cloud monitoring, but incorporating AI-based predictions (ANFIS) and a pre-roasting alarm system.
3	Pramono et al.	Low Cost Telemonitoring Technology Of Semispherical Solar Dryer	Moisture content dropped from 49.59% to 10% in 69 hours;	Provides a foundation for IoT integration, but expands upon it with ANFIS predictive models and real-time

No	Penulis	Judul	Hasil Utama	Relevansi & Research Gap
		For Drying Arabica Coffee Beans	automated cloud system	moisture level notifications.
4	Anokye-Bempah et al.	Design, calibration, and validation of an inline green coffee moisture estimation system using time-domain reflectometry	High accuracy (R^2 = 0.99) compared to oven drying	While demonstrating accurate measurement techniques, this study developed a low-cost IoT alternative that is better suited for smallholder farmers.
5	Cervini C et al.	Interacting climate change factors (CO_2 and temperature cycles) effects on growth, gene expression, and ochratoxin A production by Aspergillus carbonarius	Increases in temperature and CO_2 concentration significantly enhance the growth of Aspergillus carbonarius and the production of ochratoxin A (OTA), indicating that environmental conditions have a significant impact on the	This indicates that temperature and CO_2 affect fungal growth and the risk of contamination, making them important factors in moisture control. However, this research is still limited to laboratory settings and has not yet integrated IoT or real-time ANFIS predictions.

No	Penulis	Judul	Hasil Utama	Relevansi & Research Gap
			risk of contamination mold on food	
6.	Halil Nusret Bulus	Adaptive Neuro-Fuzzy Inference System and Artificial Neural Network Models for Predicting Time-Dependent Moisture Levels in Hazelnut Shells and Prina	ANFIS and ANN are capable of predicting moisture content with very high accuracy (R^2 = 0.9813–0.999) and very low RMSE. ANFIS performs better at predicting moisture content, while ANN performs better at predicting moisture ratio.	There is currently no integration between IoT sensors and the ANFIS model

In general, previous research indicates that the detection and control of moisture content in green coffee beans have been extensively studied, but remain fragmented in terms of methods, context, and technological accessibility. Damayanti et al. utilized an ANN based on bioelectrical properties to predict the moisture content of Robusta coffee with very high accuracy, thereby demonstrating that artificial intelligence can serve as a reliable non-destructive method [12]. However, this approach remains laboratory-based, focuses on the drying process, and is not yet integrated with IoT monitoring systems or real-time alarms in storage environments. The study by Jakkaew et al. underscores the importance of quality monitoring via environmental sensors; however, both studies remain focused on drying control and rely on relatively expensive instruments, making them less suitable for direct application by smallholder farmers or in simple storage facilities[16].

On the other hand, studies by Pramono et al. and Anokye-Bempah et al. show that the integration of technology for automatic moisture content measurement is already feasible, whether through IoT-based telemonitoring of solar dryers or TDR techniques with very high accuracy [17]. Furthermore, a study by Halil Nusret Bulus found that the ANFIS method is suitable for predicting moisture content in hazelnuts and olives with high accuracy [13]. On the other hand, a study on fungal growth by Cervini et al. shows that environmental factors such as temperature and CO_2 concentration significantly affect fungal growth and mycotoxin production, which are directly related to the moisture content of the material [14]. However, this solution still has its limitations: TDR systems are better suited for industrial-scale applications due to the high cost of the instruments, while existing IoT telemonitoring systems are not yet equipped with artificial intelligence layers such as ANFIS, do not provide intelligent moisture content predictions, and generally do not include fungal risk indicators as quality parameters [17].

From a quality dynamics perspective, the findings of Błaszkiewicz et al. confirm that green coffee beans are highly hygroscopic: their moisture content fluctuates easily in response to changes in ambient temperature and humidity during storage. This implies that a single measurement is insufficient to guarantee quality, as green beans can absorb or release moisture again after the drying process is complete [6]. This means that a system that measures moisture content only once at the end of the drying process is unable to detect the risk of quality degradation that arises during the pre-roasting storage phase. This is where the need for continuous, real-time, and networked monitoring becomes critical, especially when considering international quality standards and the risk of mold growth, which directly impacts food safety and export eligibility.

Based on these gaps, the novelty of this study lies in the development of a low-cost IoT-based monitoring system for the moisture content of green coffee beans prior to roasting, integrated with an ANFIS model, using a combination of environmental sensors (temperature, RH, CO_2) and capacitive sensors as indicators of the beans' physical condition. Unlike previous studies, which generally only: (1) focused on the drying stage, (2) used expensive instruments such as TDR, or (3) performed monitoring without predictive intelligence, this study offers a solution that:

1. Affordable and practical
2. Continuous and real-time
3. Intelligent and interpretable (the ANFIS model predicts moisture content while also providing condition categories and indications of potential mold growth), and
4. Tailored to the needs of farmers and coffee industry professionals in Indonesia, who have long relied on traditional methods without calibrated measuring instruments.

Thus, the primary rationale for this research is to bridge the gap between advanced yet expensive and inaccessible moisture content measurement technologies and the need for a monitoring system that is precise and real-time, yet remains cost-effective and easy to implement at the level of farmers and small-scale businesses. The developed IoT & ANFIS system is expected not only to address technical needs (accuracy and automation) but also to serve as a practical guide for maintaining green coffee quality in accordance with ICO/SNI standards, reducing the risk of mold during storage, and strengthening farmers' position in the high-quality coffee supply chain.

2.2 Coffee and Post-Harvest

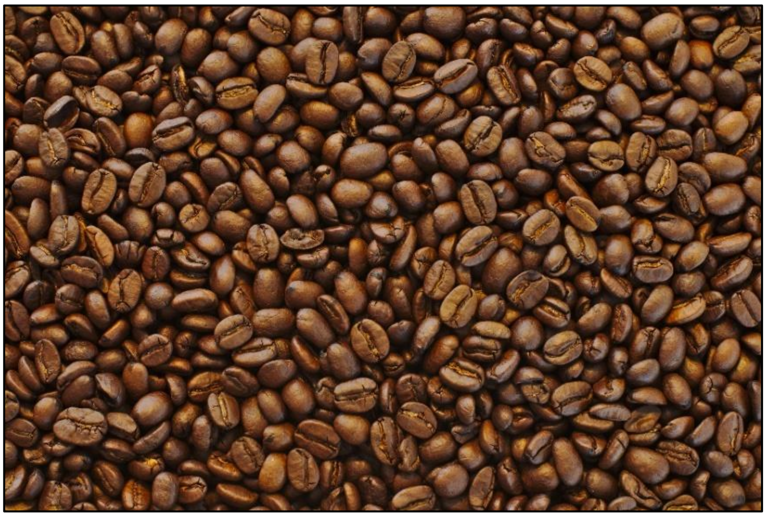

Figure 2. 1 Coffee Beans

Figure 2.1 shows coffee beans. Coffee is an agricultural commodity of significant economic, social, and cultural importance, both globally and nationally. According to a report by the International Coffee Organization (ICO, 2022), global coffee consumption continues to rise each year at an average growth rate of approximately 2% per year, and global production has exceeded 170 million bags (60 kg per bag) [12]. Indonesia ranks as the world's third-largest coffee producer after Brazil and Vietnam, contributing more than 11 million bags per year [16]. This makes coffee one of the country's leading export commodities, contributing to foreign exchange earnings while also serving as the primary source of livelihood for millions of smallholder farmers in coffee-producing regions.

Coffee products traded globally are generally green coffee beans, which are the result of a series of post-harvest processing steps. According to the FAO in 2021, the coffee post-harvest process consists of several key stages: fermentation, washing, drying, sorting, and storage [16]. This stage serves to reduce the moisture content of the coffee beans from the relatively high levels immediately after harvest to a stable level, typically ranging from 10 - 12 percent. The green beans are then stored until they are ready for roasting, a process that will determine the coffee's flavor and aroma. [10].

Proper post-harvest processing can improve coffee quality; conversely, mistakes in post-harvest processing can potentially reduce quality and even lead to economic losses. For example, green beans that are too wet or not fully dried are prone to mold growth, while coffee beans that are too dry become brittle and are prone to cracking during roasting [3]. Therefore, moisture content is a key parameter that must be strictly controlled during the post-harvest stage.

Thus, it can be concluded that coffee, as a strategic commodity, requires a well-controlled post-harvest system. This stage significantly influences the quality of green beans, which ultimately determines the market value of coffee in the global market. In the context of this study, the primary focus is on controlling the moisture content of green beans prior to roasting through IoT-based technology, which is expected to serve as an innovative solution for farmers, cooperatives, and roasteries in maintaining coffee quality.

2.3 Moisture Content of Green Beans

Figure 2. 2 Monitoring Moisture Content

Figure 2.2 shows an example of moisture content monitoring. Moisture content (MC) is a key parameter that determines the quality of green coffee beans prior to the roasting process. The MC value affects storage stability, food safety, and the consistency of the roasted product. Coffee beans with a moisture content above 12% are prone to fungal growth, such as Aspergillus and Penicillium, which can potentially produce ochratoxin A (OTA), while a moisture content below 10% makes the beans brittle and prone to cracking during roasting [5]. Research by Anokye-Bempah et al. confirms that continuous digital MC monitoring is far more representative than instantaneous measurements, primarily because coffee beans are hygroscopic that is, they are capable of absorbing or releasing water vapor from the surrounding air until they reach equilibrium moisture content (EMC) [7]. This explains why changes in relative humidity (RH) and air temperature directly affect the moisture content of coffee beans during storage.

The relationship between EMC and air humidity is described by a sorption isotherm curve, which is a graph showing the equilibrium between water activity (a*w = RH/100) and the moisture content of the material at a specific temperature. The shape of the coffee bean isotherm curve is generally sigmoidal (type II) according to the BET (Brunauer–Emmett–Teller) classification, where the increase in moisture content is slow at low RH but increases sharply at high RH. Empirical models frequently used to describe this phenomenon include GAB (Guggenheim Anderson de Boer), Oswin, and Halsey, with parameters calibrated against oven-drying test results. Recent research by Costa et al. in the study "Water Sorption Isotherms and Mid-Infrared Spectra of Dried Parchment Coffee Beans" found that temperature and post-harvest methods (wet or semi-dry) significantly influence the shape of the curve and the sorption capacity of coffee beans, with Type II isotherms remaining dominant across all treatments[18].

Technological developments in 2024–2025 indicate that sorption isotherms can be modeled quantitatively using both statistical and machine learning approaches. Research by Collazos-Escobar et al. also found that microstructural changes resulting from heat treatment alter sorption capacity; however, the relationship between water activity and moisture content can still be consistently estimated using a Type II isotherm model [19].

Thus, the moisture content of green beans serves not only as an indicator of physical quality but also as the basis for determining storage strategies and mitigating OTA risks. This principle of sorption isotherms forms the scientific basis for the use of temperature and humidity (RH/T) sensors, such as the SHT31, in IoT systems. These sensors can detect changes in RH that are directly proportional to increases in moisture content up to the EMC. When RH/T data is integrated with the ANFIS artificial intelligence model, the system can estimate moisture content in real-time and non-destructively, providing a measurable solution for small-scale coffee farmers and the industry. This approach aligns with the research by Jakkaew et al. and Damayanti et al., which emphasizes the importance of integrating IoT smart sensors and data-driven modeling to ensure sustainable post-harvest coffee quality.

2.4 Internet of Thinks (IOT)

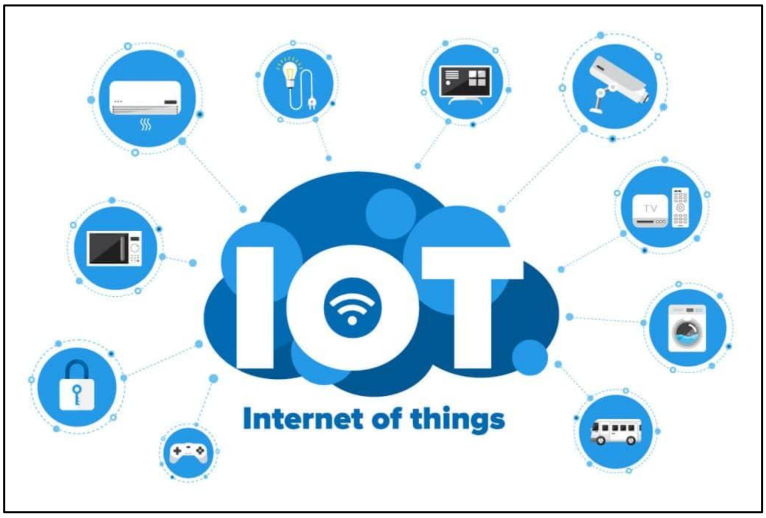

Figure 2. 3 Internet of Thinks

The Internet of Things (IoT) is a concept of connectivity among smart devices that enables sensors, actuators, and control systems to automatically exchange data over the internet. In the context of agriculture, IoT serves as a cornerstone of precision agriculture, which focuses on collecting environmental data to improve production efficiency and quality. According to Soussi, the integration of smart sensors with IoT systems allows farmers to monitor critical variables such as temperature, humidity, and soil moisture in real-time through a digital platform. The general IoT architecture consists of four layers: the perception layer (sensors/actuators), the network layer (data communication), the middleware layer (processing), and the application layer (user interface). This structure ensures that data from sensors can be efficiently transmitted to a cloud dashboard for visualization and AI-based analysis. [20].

Advances in hardware such as the ESP32 drive high efficiency in modern IoT systems due to its ability to integrate Wi-Fi, Bluetooth, and edge computing for local data processing. According to Miller, the combination of IoT and AI has entered a revolutionary phase in the agricultural sector, where machine learning-based predictive models and fuzzy inference are capable of learning sensor data patterns to automatically determine adaptive actions [21]. The implementation of this system has been shown to reduce post-harvest yield losses by up to 25% and improve the energy efficiency of the sensor network. In this study, IoT serves as the backbone of the pre-roasting coffee moisture content detection system, where data from the SHT31 and soil moisture sensors is sent to a cloud dashboard to be processed by an ANFIS model and triggered by real-time alarms. Thus, the implementation of IoT is not merely a monitoring tool but a data-driven adaptive system capable of sustainably maintaining the quality of green beans.

2.5 Arduino IDE

Figure 2. 4 Arduino IDE Logo

The Arduino Integrated Development Environment (IDE) is a development environment for writing, compiling, and uploading programs to microcontroller boards (including the ESP32). Version 2 features a modern editor with autocompletion/Quick Suggestions, code navigation, and a live debugger, making the process of writing and debugging more efficient for IoT projects. The official documentation also provides a Getting Started guide and a Board Manager for installing the necessary board packages directly from the IDE. This is relevant to the research because it accelerates code iteration for RH/T (SHT31) readings, data transmission to the cloud, and real-time alarm testing [22]. Figure 2.4 shows the Arduino IDE logo.

For the ESP32, Arduino support is provided by the Arduino-ESP32 core from Espressif and can be installed via the Boards Manager in the Arduino IDE. Espressif’s official guide explains the installation steps, board selection, and compatibility/migration notes, ensuring that the toolchain setup and firmware upload processes follow standard procedures. The combination of the Arduino IDE and the Arduino-ESP32 core provides a stable workflow: editor → compilation →

upload → serial monitor, which aligns with the architecture of the research IoT system [23].

2.6 Broker MQTT

Figure 2. 5 MQTT Logo

An MQTT broker is the central component of a publish/subscribe architecture that receives messages from devices (publishers), temporarily stores them according to quality of service (QoS 0/1/2) rules, and then distributes them to subscribing clients (subscribers). Popular broker implementations in research and industry include Eclipse Mosquitto (lightweight, open-source) and EMQX (large-scale, enterprise-featured), both of which support MQTT 5.0 as well as core features such as Retained Messages and Last Will and Testament (LWT) for system reliability. The official documentation for Mosquitto and EMQX details protocol version support, retained message management, and configuration options (e.g., autosave, check_retain_source) relevant to the real-time system design in this research [24], [25]. Figure 2.5 shows the MQTT broker's logo.

In terms of performance and energy efficiency, the choice of QoS directly impacts power consumption and the reliability of sensor data transmission. An experimental study (Sensors, 2023) reports that over TLS connections, QoS 1 tends to be the most energy-efficient compared to QoS 0 and 2, while a 2024 review summarizes the QoS characteristics: at most once (0), at least once (1), and exactly once (2). The MQTT Essentials material from HiveMQ can serve as a conceptual reference for understanding the pub/sub pattern, comparisons with other IoT protocols, and best practices for integration with edge and cloud applications. Thus, for the green-bean moisture IoT system, the combination of Mosquitto/EMQX with QoS 1 and the use of retained/LWT provides reliable notifications and a good initial state on both the Blynk dashboard and the backend service [26].

2.7 ESP32

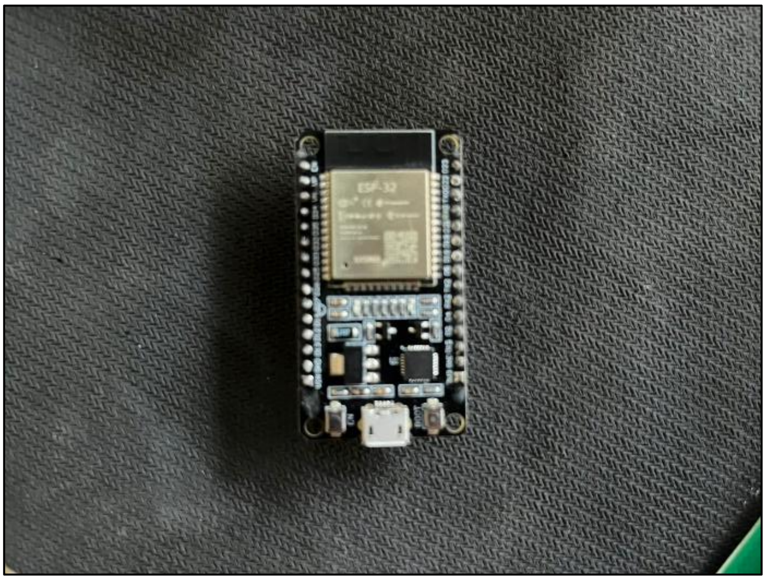

Figure 2. 6 ESP32

The ESP32 is an ESP32 SoC-based development board that integrates 2.4 GHz Wi-Fi and Bluetooth (BR/EDR + BLE), a dual-core Xtensa CPU running up to 240 MHz, internal SRAM, ADC, PWM, SPI, I²C, and UART into a compact, breadboard-ready form factor. Figure 2.6 shows the ESP32 devkit. Espressif's documentation provides the DevKitC user guide, hardware reference, and technical reference manual (TRM) for details on architecture, peripherals, and hardware design guidelines. With support for the official Arduino-ESP32 core, this board can be programmed via Arduino IDE 2.x or ESP-IDF; the installation process for the board package and toolchain is explained in Espressif's online documentation. These features make the ESP32 DevKit a popular choice for IoT research due to its combination of wireless connectivity, rich peripherals, and a mature software ecosystem [27].

In the context of this research, the ESP32 DevKit serves as a sensor node that reads data from the SHT31 (I²C) and a capacitive humidity sensor (ADC), performs edge processing, and then forwards the data to the cloud via MQTT. The Arduino-ESP32 documentation confirms the availability of official cores and a getting-started guide, while recent studies highlight the ESP32's power consumption profile and relevant low-power practices for low-power real-time monitoring. For I²C connections on the Arduino-ESP32, the standard pins commonly used are GPIO 21 (SDA) and GPIO 22 (SCL) (which can be changed via `Wire.begin(SDA, SCL)`), making SHT31 integration simple and stable. With these features, the ESP32 DevKit is well-suited as the backbone of an IoT device for detecting the moisture content of green beans, which requires stable connectivity, multi-interface sensor acquisition, and power-saving options during idle periods [27].

2.8 Expansion Board

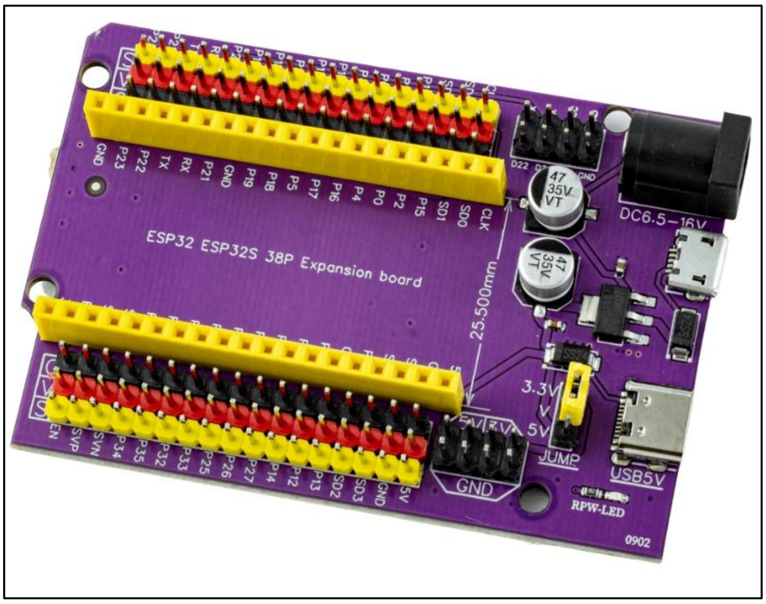

Figure 2. 7 Expansion Board ESP32

An expansion board or shield for the ESP32 serves to expand the I/O interfaces (digital, analog, I²C, SPI, UART) while simplifying wiring and power supply, making the device more stable for hours of monitoring. Figure 2.7 shows an ESP32 expansion board. Modern expansion boards often include threaded terminal blocks to prevent sensor cables from coming loose in field environments, and provide external voltage input (≈7–24 V) with on-board power regulation so the load does not rely entirely on the USB port. DFRobot's documentation highlights an expansion board for the ESP32 FireBeetle 2 family featuring ready-to-use I²C/UART/SPI ports, power disconnection via the EN pin, per-channel LED indicators, and an option to disable LEDs for lower power consumption—a configuration well-suited for sensor nodes in your system [27].

To ensure safe integration, the ESP32's electrical specifications must be followed: the I/O logic operates at 3.3 V, and power delivery from the core board must adhere to the recommendations in the datasheet and the official ESP32-DevKitC schematic. Espressif's documentation confirms the DevKitC pin-header layout, the USB-UART/auto-program path (RTS/DTR → EN/IO0), and the I²C references; while the ESP-IDF/Arduino-ESP32 documentation explains that I²C is flexible (common default: SDA = GPIO21, SCL = GPIO22) and can be remapped via the API if the expansion board remaps the pins. With the combination of the DevKitC core board and the terminal-block expansion board, the ESP32 node can accommodate an SHT31 sensor (I²C), a capacitive humidity sensor (ADC), a buzzer/LED/relay, and a more stable external power supply, making it suitable for monitoring the moisture content of green beans prior to roasting—a task requiring neat wiring, secure connections, and long-term operation [27].

2.9 SHT31

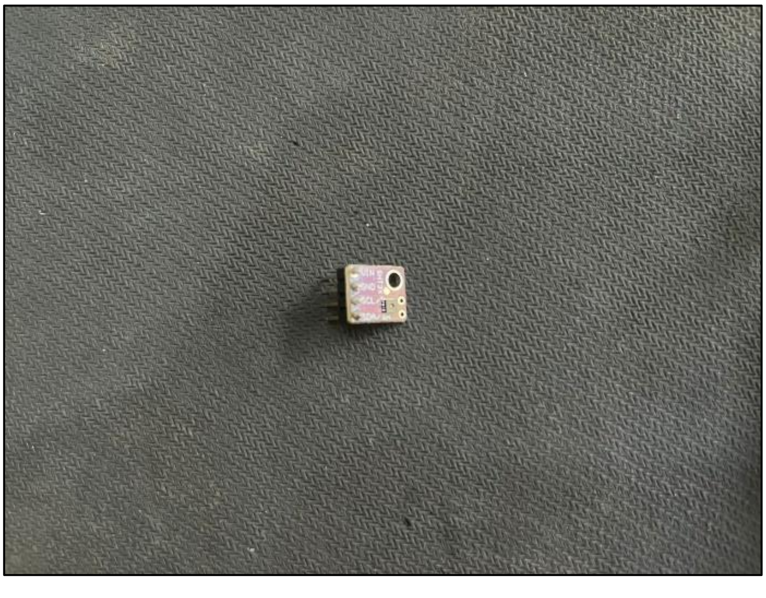

Figure 2. 8 SHT31 Sensor

The SHT31 is the "standard" variant of the SHT3x humidity-temperature sensor series, featuring CMOSens® architecture with a calibrated, linearized, and temperature-compensated digital output. Figure 2.8 shows the SHT31 sensor. According to the official datasheet, the SHT3x supports a 2.15–5.5 V supply, an

I²C interface up to 1 MHz with two selectable addresses, and provides an internal heater option to reduce condensation during RH measurements. The typical accuracy of the SHT31 is in the range of ±2 %RH and ±0.3 °C (depending on the variant and test conditions), with fast startup and measurement times, making it suitable for real-time monitoring in IoT systems. This information confirms that the SHT31 combines good metrological performance with ease of hardware-software integration in the design of the sensor node for this research [28].

In design-in practice, Sensirion recommends paying special attention to sensor placement, PCB/housing design, and thermal management, as local temperature variations will directly affect RH readings (due to the temperature-dependent nature of RH). The latest design guide outlines layout rules (minimizing heat sources around the sensor, ventilation, airflow paths), as well as mechanical options such as membranes or protective covers for dusty or humid environments. On the interface side, the SHT31 uses I²C with a default address of 0x44, which can be changed to 0x45 (e.g., via the ADR pin on the breakout module), making it easy to use two sensors on a single bus. By following the manufacturer's guidelines and the ESP32's default I²C pin mapping (SDA=GPIO21, SCL=GPIO22), integrating the SHT31 is simple, stable, and reliable [29].

2.10 MH-Z19

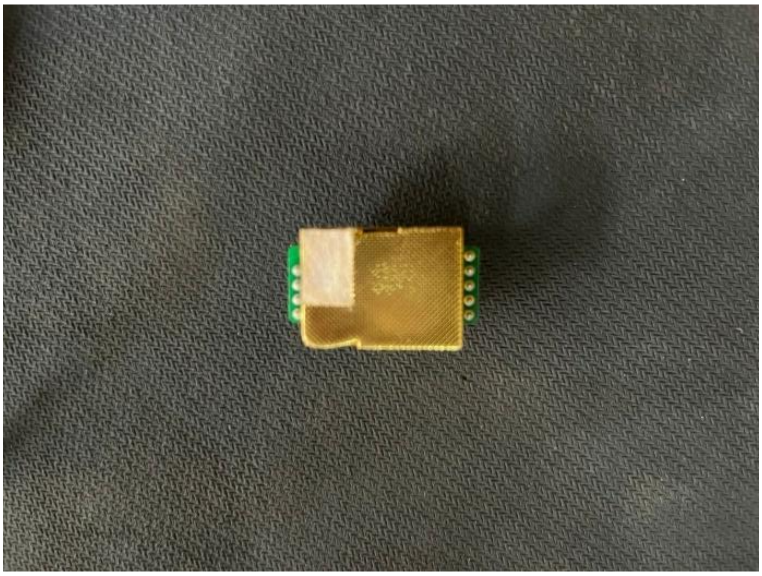

Figure 2. 9 MH-19 Sensor

The MH-Z19 is an NDIR (Non-Dispersive Infrared)-based CO_2 sensor module featuring internal temperature compensation, UART and PWM interfaces, and a typical operating range of 0–5000 ppm; this module is designed for indoor air quality and HVAC applications. Figure 2.9 shows the MH-19 Sensor. Winsen's official datasheet/manual highlights key characteristics such as high selectivity, oxygen-independent operation, a lifespan of >5 years, and operating environmental limits (-10 to 50 °C; 0–95% RH without condensation) that must be adhered to when installing the sensor in an IoT node. The document also outlines the UART command format, PWM timing, and recommendations for zero/span calibration to maintain long-term measurement accuracy. This aligns with the practice in this study, where the sensor is placed in a ventilated air duct, away from local heat sources, and polled via UART to ensure stable CO_2 readings for alarm triggers and dashboard displays [30].

Recent studies identify the MH-Z19 as a viable low-cost CO_2 sensor when combined with calibration and environmental corrections. Research by Barrett et al., 2024, demonstrates the use of the MH-Z19 in field campaigns and reports the manufacturer's accuracy specifications (±50 ppm + 5% of reading), while emphasizing the importance of temperature/humidity correction to minimize bias. Reviews and evaluations of low-cost CO_2 sensors also recommend a statistical/ML-based calibration approach against reference instruments to ensure long-term performance. On the other hand, integrating the MH-Z19 into a multisensor system highlights its long-term durability, internal temperature compensation, and resistance to water vapor, confirming that operation in accordance with the datasheet and calibration routines are key to accuracy in IoT applications. These findings support the selection of the MH-Z19 in your system as a reliable CO_2 data source that can be integrated into ANFIS models [31].

2.11 Buzzer

Figure 2. 10 Buzzer Sensor

A buzzer functions as an audio actuator to provide real-time alarms or warnings when threshold conditions are exceeded (e.g., humidity, temperature, or CO_2 levels). Figure 2.10 shows a Buzzer Sensor. The two common types are piezo (higher voltage, lower current, SPL tends to be higher at the resonance frequency) and magnetic (lower voltage, higher current), selected based on voltage requirements, power consumption, and target sound frequency. In ESP32 implementations, active buzzers can be directly driven via the LEDC's built-in PWM (frequency/duty cycle configuration), while passive buzzers typically require a PWM signal at the target frequency (hundreds of Hz–kilohertz) to produce a stable tone. The ESP-IDF LEDC documentation explains the determination of frequency-resolution trade-offs to generate stable waves, ensuring that the alarm tone remains consistent and undistorted across various PWM channels [32].

In IoT systems, best practices for alarms and notifications recommend defining an effective frequency range and a sound level sufficiently above ambient noise. The latest technical guidelines reference alarm design principles based on IEC 60601-1-8: use an operating range of ±200 Hz–5 kHz, with a focus on 500–3000 Hz, and ensure the sound level is several dB above ambient noise so it is easily audible without being excessive; this serves as a relevant general reference for non-medical audible alerts as well. Recent IoT studies show that buzzers are commonly used as indicators of failure, power status, or process quality, and the integration of the ESP32 in academic and industrial projects positions the buzzer as a low-power actuator responsive to logic rules at the edge or in the cloud. Thus, the buzzer provides a fast feedback channel that complements visual notifications (LEDs, LCDs, dashboards) and push notifications in applications [33].

2.12 Red-Yellow-Green LED Light

Figure 2. 11 Red-Yellow-Green LED Light

Red–yellow–green (RAG) indicator lights are commonly used to intuitively convey system status: red indicates danger or immediate action, yellow or amber indicates a warning or preparation for action, and green indicates a safe condition. This color-semantic principle aligns with the international ISO 22324 (2022) guideline on color-coded alerts, which recommends mapping colors to severity levels to help users make quick decisions. Figure 2.11 shows an RGB LED sensor. Referring to this standard, the IoT system in this study maps water content thresholds (or supporting parameters, such as temperature/CO_2) to RAG statuses so that alarms are uniform and easily understood across users [34].

From an implementation standpoint, the LEDs are driven via PWM to control brightness and flashing patterns (e.g., fast flashing for danger, slow flashing for warnings) using the LEDC peripheral on the ESP32. The ESP-IDF LEDC documentation describes the frequency/resolution range and high-resolution duty cycle control suitable for stable visual indicators. Electrically, each LED requires a current-limiting resistor based on the forward voltage (Vf) and the LED's rated current; manufacturer datasheets (e.g., Vishay/OSRAM) provide typical Vf values per color (red, yellow, green) and thermal considerations necessary to ensure long-term reliability. With a combination of color semantics (ISO 22324) + LEDC control + proper resistor design, the RAG indicator becomes a clear, power-efficient, and durable notification channel [35].

2.13 Soil Moisture

Figure 2. 12 Soil Moisture Sensor

Soil moisture is the amount of water stored within the soil pores and plays a crucial role in plant physiological processes, irrigation efficiency, and agricultural environmental management. Figure 2.12 shows a Soil Moisture Sensor. Hidayat et al. explain that monitoring soil moisture content using low-cost capacitive sensors can be performed with high accuracy if the sensors are calibrated against the physical characteristics of the soil. The study emphasizes that proper calibration enables the sustainable implementation of soil moisture monitoring systems to support precision agriculture, particularly on slopes with artificial rainfall. This approach demonstrates significant potential for the implementation of simple sensors in IoT-based systems that require energy efficiency, low cost, and real-time data transmission capabilities [36].

Furthermore, Abdelmoneim et al. assert that low-cost capacitive sensors can serve as an effective alternative for smart irrigation systems if calibrated according to soil type and salinity levels. The study demonstrated a significant improvement in moisture reading accuracy following the application of an empirical model-based calibration method. Integrating the calibration results into an Internet of Things system enables dynamic moisture monitoring, automated watering schedules, and water savings of over 20%. Thus, recent research indicates that well-calibrated soil moisture sensor technology can support smart decision-making and the sustainable management of water resources in the modern agricultural sector [37].

2.14 LCD I2C 20x4

Figure 2. 13 LCD I2C 20x4

A 20×4 character LCD based on the HD44780 controller is commonly paired with an I^{2}C backpack (typically the PCF8574 expander IC) so that communication requires only two pins (SDA-SCL) on a microcontroller such as the ESP32. This approach reduces the need for parallel pins (D4–D7, RS, E, etc.) to a cable-efficient I^{2}C serial interface, while still maintaining core functions: 20-column × 4-row display, cursor control, and custom characters. The latest PCF8574 datasheet confirms that this chip operates at 2.5–6 V, providing 8-bit I/O expansion for the I^{2}C bus, making it ideal as a bridge between the I^{2}C bus and the digital signals required by the character LCD (4-bit mode), including backlight control via one of the expander pins [38].

In modern embedded system implementations, 20×4 LCDs with I^{2}C interfaces are commonly used to display system status, alarms, or sensor parameters in real time, due to their low cost, ease of integration, and high readability in field environments. The 2023 IJACSA study on vehicle collision detection systems, for example, positions the LCD as a user interface component for displaying warnings and location information—a representative example of the LCD's pivotal role in contemporary IoT/embedded ecosystems. A similar integration in your project (ESP32 + MQTT) enables local (offline) display as a failsafe when internet connectivity is problematic, while also providing instant feedback when water/CO_2 thresholds trigger a buzzer or status changes on the dashboard [39].

2.15 Jumper Cable

Figure 2. 14 Jumper Cable

Jumper wires are flexible conductors with pin or connector ends used to connect points on a breadboard or microcontroller header during prototyping. In the context of embedded systems education and research, jumper wires facilitate rapid assembly, module replacement, and debugging without the need for permanent soldering. An educational study by Govender explicitly states that participants were introduced to Arduino kit components, including breadboards and jumper cables, and then practiced controlling LEDs a concrete example of the role of jumper cables as a learning tool for circuit design and microcontroller programming. [40].

In research on modern IoT prototypes, the use of jumper wires is also clearly documented. The Energies article (2025) details the connection of the BH1750 sensor to the Raspberry Pi via a breadboard "using jumper wires" (SDA to GPIO2, SCL to GPIO3, etc.) as a standard assembly practice; this reflects the I^2C wiring conventions you also use (ESP32–SHT31/LCD I^2C) . In an ESP32-based workplace safety application, Safety (2025) mentions integrating sensors via jumper cables on a breadboard to assemble a smart helmet, emphasizing the speed of integration and flexibility during experimentation [41]. For research and educational purposes, Sensors (2024) introduces the LEWIS platform and highlights the simplification of sensor connections including the use of headers and jumpers to make low-cost wireless experiments easy to replicate [42].

2.16 Power 12v

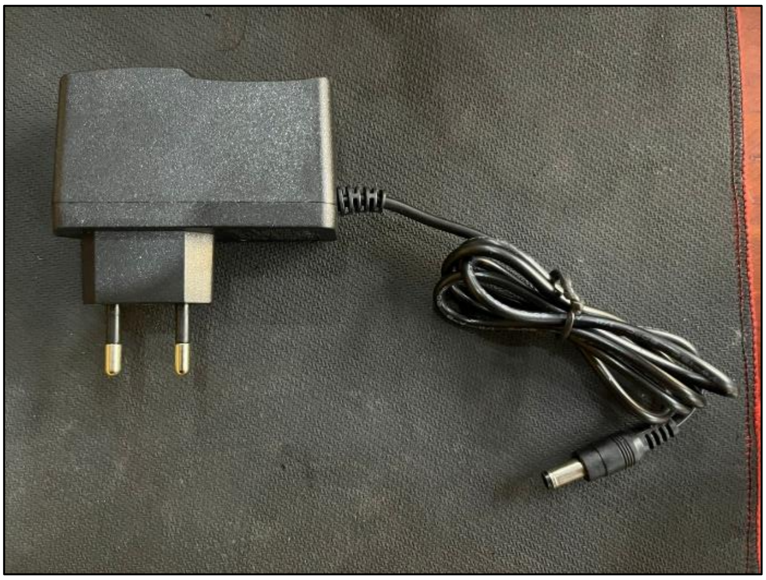

Figure 2. 15 Power 12V

A 12 V DC power adapter is one of the most common power sources in embedded systems and the Internet of Things (IoT). The 12 V voltage is chosen because it provides sufficient power headroom for high-power components such as fans, relay modules, or active buzzers, while also allowing easy step-down conversion to 5 V or 3.3 V for microcontroller logic and sensors. According to Hercog et al. (2023), ESP32-based IoT device designs typically use a 12 V adapter as the primary power rail, which is then fed into an internal step-down regulator. This approach ensures voltage stability for digital subsystems without the need for multiple separate adapters, thereby reducing complexity and the risk of power disruptions [43].

The performance of a 12 V DC adapter depends heavily on the converter technology it employs, typically a high-efficiency, low-ripple SMPS (Switch-Mode Power Supply). Ventimiglia et al. (2024) emphasize the importance of monitoring the condition of the MOSFET in a 12 V DC–DC converter, as this component determines the efficiency and reliability of the power supply [44]. Meanwhile, Tan et al. (2024) demonstrate that a buck converter topology with time-based current-mode control is capable of maintaining voltage stability during rapid load changes a factor of great relevance when a 12 V adapter is used to power a microcontroller, an SHT31 sensor, an I²C LCD, and a buzzer in an IoT system [45]. Thus, the 12-volt adapter serves not only as the primary power source but also as the starting point of the power distribution chain, which determines the reliability of the entire electronic system.

2.17 USB Micro-b

Figure 2. 16 USB Micro-b

The USB Micro-B connector is one of the most widely used data and power interfaces in portable electronic devices, including microcontroller boards such as the ESP32 DevKit, Arduino Nano, and sensor module boards. This standard features five main pins: VBUS (5 V), D+, D−, ID, and GND, with the capability to deliver up to 2 A of current for charging or powering low-power microprocessor systems. According to Lee et al. (2023), the Micro-B connector is a popular choice due to its compact size and broad compatibility with various standard USB cables. In the context of IoT, the Micro-B port functions not only as a charging path but also as a serial communication path (USB-to-UART bridge) between a computer and a control board, enabling direct programming, debugging, and data transmission without requiring an additional interface [46].

The Micro-B connector also plays a crucial role in the device's power supply stability. Research by Huang et al. (2024) shows that the quality of the USB connector and the impedance of the power path can affect output voltage stability, particularly at high currents, due to contact resistance and ground fluctuations [47]. In an IoT system like the one you designed, the Micro-B USB connector serves as the primary power source (5 V), which is then stepped down to 3.3 V via the onboard regulator on the ESP32, while also functioning as the communication channel to the host PC via a USB-to-UART conversion chip (e.g., CP2102 or CH340). This combination of power and data functions makes the Micro-B a crucial component in ensuring reliable power supply and ease of programming for modern IoT systems.

2.18 ANFIS

Neuro-fuzzy is a combination of the learning capabilities of artificial neural networks (ANN) and the linguistic reasoning capabilities of fuzzy logic. Architecturally, this system uses neural network layers whose weights can be trained to automatically generate Fuzzy rules (IF-THEN) or adjust membership functions and Fuzzy inference rules based on training data. Thus, ANFIS enables a system that not only classifies or predicts numerically but is also capable of providing linguistic interpretations (“low,” “normal,” “high”) or more human-like recommendations that are highly relevant for water content detection system applications. For example, research by Apiecionek shows that neural networks with fuzzy elements can run on power-constrained IoT devices and still produce adequate accuracy [48].

1. Normalisasi Min-Max

Before the input is processed by the layers, normalization is required to ensure that the values do not vary too widely. For this normalization, the Min-Max Normalization method is used. The formula for this normalization is shown below:

$$x_{norm} = \frac{x - x_{min}}{x_{max} - x_{min}}$$

The initial stage in ANFIS is the fuzzification process, which involves mapping each input variable to membership degrees using membership functions. Gaussian membership functions are widely used in recent research because they have a smooth, continuous curve and are capable of realistically representing variations in sensor data. Mathematically, the Gaussian membership function is expressed as:

$$\mu ij = (x_i) = \exp\left(-\frac{1}{2}\left(\frac{x_i - c_{ij}}{\sigma_{ij}}\right)^2\right) \tag{1}$$

where c_(ij) is the center and σ_(ij) is the width of the membership function.

These membership degree values are then combined to form fuzzy rules. The firing strength of each rule is obtained by multiplying the membership degrees of all inputs, which is expressed as

$$w_r = \prod_{i=1}^{n} u_{i,r}(x_i) \tag{2}$$

These values are then normalized to obtain the relative weights of each rule, namely

$$\bar{w}_r = \frac{w_r}{\sum_{k=1}^{r} w_k} \tag{3}$$

where R denotes the number of fuzzy rules formed.

In the consequent layer, each rule has an output function in the form of a linear equation as follows:

$$f_r(x) = a_{r1}x_1 + a_{r2}x_2 + \cdots + a_{rn}x_n + b \tag{4}$$

The final output of the ANFIS system is obtained using the weighted average method, namely :

$$\hat{y} = \sum_{r=1}^{R} \bar{w}_r\, f_r(x) \tag{5}$$

During the training process, the membership function parameters and consequence coefficients are optimized using a gradient-based learning algorithm with a Mean Squared Error (MSE) objective function, which is expressed as :

$$\text{MSE} = \frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2 \tag{6}$$

The use of adaptive optimizers such as Adam has been shown to accelerate convergence and improve the training stability of ANFIS models on nonlinear and noisy data.

The main advantages of ANFIS over pure ANNs or pure Fuzzy systems are adaptability and interpretability. Pure ANNs have high learning capabilities but low interpretability (black-box), whereas pure Fuzzy systems are easy to interpret but difficult to train automatically when there are many rules or large amounts of data. ANFIS harmonizes these two aspects: sensor data from IoT systems (e.g., humidity and temperature measurements) can be fed as input into a neural network, which then passes through a fuzzy layer to produce a condition classification output, while also enabling automatic tuning as environmental conditions change. Real-world example: an IoT power quality monitoring system using ANFIS (Adaptive Neuro-Fuzzy Inference System) that improves performance by over 20% compared to traditional methods [49], [50].