IOSR Journal of Electronics and Communication Engineering (IOSR-JECE)
e-ISSN: 2278-2834, p- ISSN: 2278-8735.Volume 21, Issue 6, Ser. 1 (Nov. – Dec. 2026), PP 01-07
www.iosrjournals.org

Design, Implementation, and Evaluation of a Dual-Zone ESP32-Based Smart Adaptive Irrigation System with Cloud Telemetry

K Padmapriya[1] , Khaja Mujeebuddin Quadry[2] , B. Abdul Raheem[3]
[1]*PG Scholar, Department of ECE, Joginpally B.R. Engineering College, Hyderabad, India*
[2]*Professor, Department of ECE, Joginpally B.R. Engineering College, Hyderabad, India*
[3]*Professor, Department of ECE, Joginpally B.R. Engineering College, Hyderabad, India.*

Abstract: *Conventional fixed-schedule or manual irrigation does not respond to actual field conditions and tends to over- or under-supply water. Fields with several sections also require each section to be controlled independently, while sharing one water source, whereas most reported low-cost IoT irrigation systems are single-zone. A dual-zone irrigation controller based on an ESP32 microcontroller was built and operated on a two-tray planted testbed. Each block has its own soil-moisture sensor, water-level sensor, and solenoid valve, and a single pump is shared. A block is irrigated when its soil moisture is below 30% or its water level is below 25%. Rain sensing with hysteresis inhibits irrigation, each valve opens 3 s before the pump starts, each cycle is limited to 300 s, and a 120 s cooldown follows irrigation. The information is displayed on a 16 X 2 LCD and uploaded to ThingSpeak every 15 s for cloud storage. Block-wise functional tests showed that low soil moisture or low water level in a block opened the corresponding valve and started the shared pump. In an 8-day cloud-logged window of 3,291 entries the pump was ON in 2.86% of entries, and Soil Moisture 1 averaged 66.65% (SD 7.57). The prototype achieved independent two-zone irrigation with a shared pump and remote monitoring. Longer logging with the cooldown active and a comparison against fixed-schedule irrigation are needed to quantify water savings.*
Keywords: *smart irrigation; ESP32; Internet of Things; soil moisture sensing; dual-zone control; ThingSpeak; precision agriculture.*

Date of Submission: 02-11-2026 Date of Acceptance: 13-11-2026

I. Introduction

Agriculture depends on reliable water management, since irrigation directly affects plant growth, soil condition and resource use. Manual and fixed-schedule irrigation do not continuously account for variations in soil moisture or environmental conditions. Sensor-based irrigation instead generates decisions from measured field conditions, and Internet of Things (IoT) connectivity extends this by linking field-level sensing and control to cloud platforms for remote monitoring [1,2]

Many low-cost implementations found in the literature focus on a single soil-moisture threshold and a single pump-control decision. This does not address the independent irrigation requirements of several field sections, nor the interaction of moisture, water availability, rainfall, and actuator sequencing that a multi-zone controller must manage [3,4]. This work presents an experimentally implemented two-block irrigation controller based on the ESP32 that monitors each block independently, inhibits irrigation during rain, and drives independent solenoid valves from a common pump.

The paper discusses on the following: (1) a dual-zone hardware architecture combining per-block soil-moisture and water-level sensing, shared rain and ambient sensing, relay-driven valves and pump, and a local LCD; (2) a rain-priority control algorithm with valve-before-pump sequencing, bounded irrigation time and a post-irrigation cooldown, applied independently to each block; and (3) an experimental evaluation on a physical testbed using functional block-wise tests and cloud-logged operational data.

II. Related Work

Several ESP32/ESP8266-based irrigation systems have been reported. Shobana et al.[1] used an ESP8266 NodeMCU with a soil-moisture sensor and DHT11 to automate single-zone irrigation with ThingSpeak logging. Morchid et al.[2] proposed a three-layer IoT architecture (device, cloud, dashboard) using ThingsBoard for telemetry and real-time visualization, with emphasis on cloud architecture. Boukri et al.[3] presented an experimentally validated, low-cost ESP32 irrigation system using capacitive soil-moisture sensors and Blynk monitoring, emphasizing affordability for small-scale farmers. Pereira et al.[4] implemented an ESP32 drip-irrigation system combining soil-moisture, temperature, air-humidity and water-flow sensing with Blynk-based control and reported improved water-use efficiency. Purnama et al.[5] reported an ESP32 irrigation controller

using a percentage-based soil-moisture calibration and Blynk monitoring. Shiva Shankar et al.[6] implemented a low-cost ESP32 system using Adafruit IO/MQTT with soil-moisture, light and temperature sensing, publishing sensor feeds and motor status to a cloud dashboard.

These systems are predominantly single-zone and focus on sensing and cloud connectivity. The system presented here addresses the multi-zone case: two blocks with independent soil and water sensing and dedicated valves share one pump, which requires explicit rules for valve-before-pump sequencing, shutdown ordering when only one block finishes, rain inhibition, bounded run time, and cooldown.

III. Materials And Methods

The wok is an study with **e**xperimental development and evaluation of a physical dual-zone irrigation prototype carried out at the Department of Electronics and Communication Engineering, Joginpally B.R. Engineering College (JBREC), Hyderabad, India.

The system uses two irrigation blocks, each with a soil-moisture sensor and a water-level sensor, plus shared rain and DHT11 temperature/humidity sensing, two relay-driven solenoid valves (one per block), and a single relay-driven pump that draws water from a reservoir and supplies both valves. An ESP32-WROOM board is the central controller: it interfaces all sensors and relays, drives a 16x2 I2C LCD, and publishes telemetry to ThingSpeak over Wi-Fi as shown in Fig. 1. The fabricated controller board is shown in Fig. 2 and the experimental testbed in Fig. 3. The GPIO assignment is given in Table 1 below.

Table 1: ESP32 pin assignments.

Device	Function	GPIO
DHT11	Temperature / humidity	4
Soil Sensor 1	Block 1 soil moisture	34
Water Sensor 1	Block 1 water level	33
Soil Sensor 2	Block 2 soil moisture	32
Water Sensor 2	Block 2 water level	35
Rain Sensor	Rain detection	39
Valve 1 Relay	Block 1 valve	27
Valve 2 Relay	Block 2 valve	26
Pump Relay	Pump control	25

Each analog channel is sampled five times with a 2 ms inter-sample delay and averaged. The averaged ADC value is mapped linearly from a dry calibration point (ADC = 4095, 0%) to a wet calibration point (ADC = 753, 100%) and constrained to 0%–100%. The rain channel (GPIO 39) was read as a raw ADC value; rain was declared when the reading falls below 1800 and cleared when it rose above 2800, and between these thresholds the previous state was retained, which provides hysteresis against signal fluctuation.

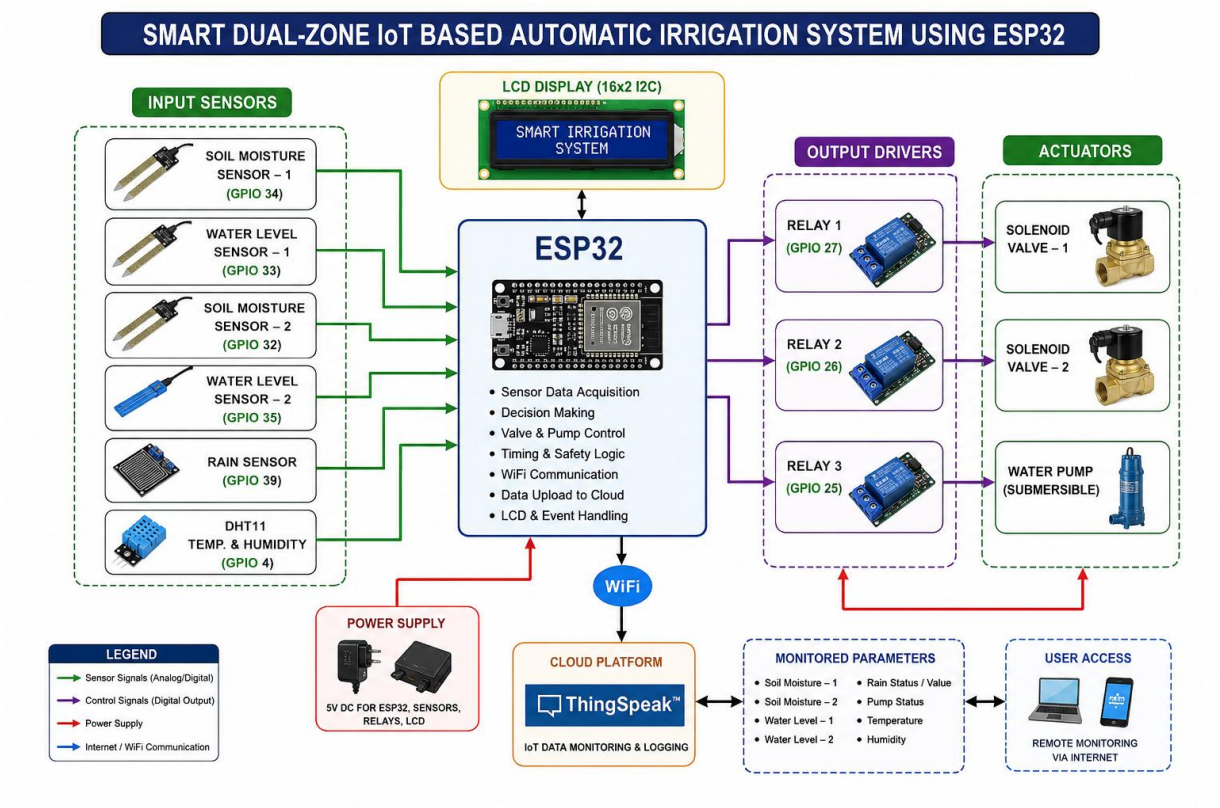

Fig. 1: System block architecture — input sensors, ESP32 controller, relay outputs, actuators, and cloud path.

The valve and pump relay modules are active-low (a LOW level energizes the relay). All relays are driven to the OFF state at start-up so that no actuator is energized before the control loop begins.

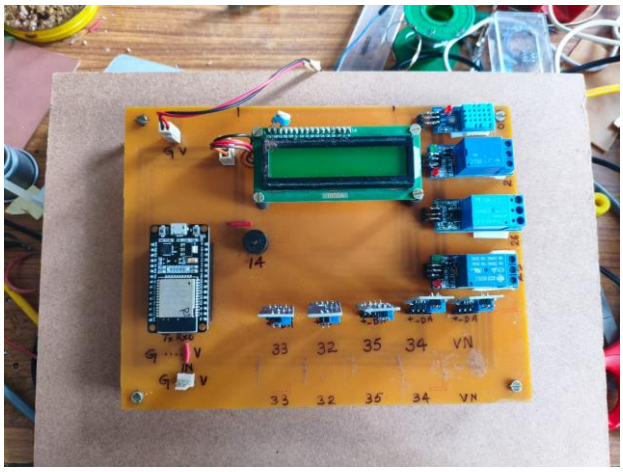

Fig. 2: Fabricated controller board — ESP32-WROOM, 16x2 LCD, DHT11 and relay bank, with labeled sensor and relay connections.

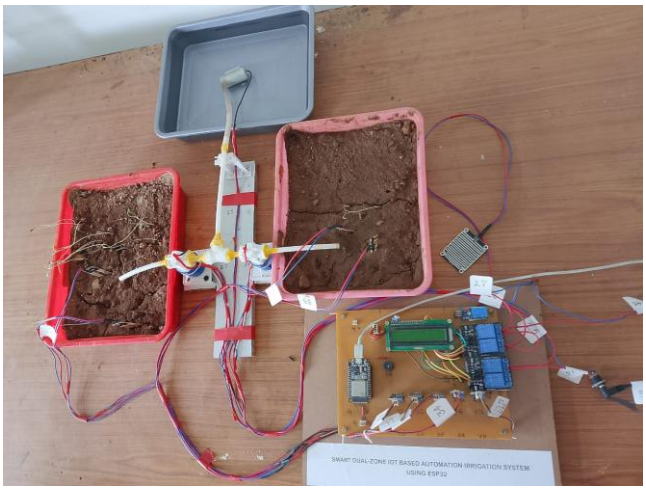

Fig. 3: Experimental testbed — Block 1 and Block 2 planted trays, each with its own soil-moisture and water-level sensors and a dedicated solenoid valve, supplied by a common pump from a shared reservoir; the controller board and rain sensor are also shown.

The firmware executes the control loop as shown in Fig. 4, with the parameters listed in Table 2. On each pass, the controller reads and conditions all sensors, updates the rain state, evaluates the cooldown timer, determines the need of each block, and then starts or stops the valves and pump accordingly.

Table 2: Control parameters.

Parameter	Value
Soil-moisture threshold	< 30 %
Water-level threshold	< 25 %
Rain detect / clear (raw ADC)	< 1800 / > 2800
Valve-to-pump delay	3 s
Pump-stop drain delay	3 s
Maximum irrigation time per block	300 s
Cooldown after irrigation	120 s
DHT read interval	3 s
LCD screen rotation	3 s
ThingSpeak upload interval	15 s

Block 1 requires irrigation when its soil moisture is below 30% or its water level is below 25%. Block 2 is evaluated identically with its own sensors, and the two blocks are evaluated independently on every pass.
When rain is detected or a cooldown is active, irrigation is disallowed. If the pump is running, it is stopped, a 3 s drain delay is applied, and any open valve is closed. The cooldown starts whenever a block completes irrigation and no other valve remains open, and it inhibits all irrigation for 120 s, preventing rapid pump cycling. The LCD screen displays the status RAIN STOP, COOLDOWN, B1 ON, B2 ON, B1+B2 ON or SYSTEM IDLE.
When a block needs irrigation and its valve is closed, the valve opens first. If the pump is not already running, the controller waits for 3 s and then starts it so that the flow path exists before the pump is energized. If both blocks demand water together, the second valve opens without restarting the pump.
A block is complete when its need condition clears or its valve has been open for 300 s. If the other block's valve is still open, only this block's valve is closed, and the pump continues to run. Otherwise, the pump is stopped, the 3 s drain delay is applied, the valve is closed, and the cooldown starts. A safety check stops the pump whenever it is running with both valves closed.
A non-blocking message queue shows event messages (for example valve and pump transitions) on the LCD. Between events, the display rotates every 3 s through block soil/water values, temperature, and humidity, rain value, pump state, and system status. Every 15 s, the controller uploads eight fields to ThingSpeak: soil moisture 1, soil moisture 2, water level 1, water level 2, rain raw value, pump state, temperature, and humidity.

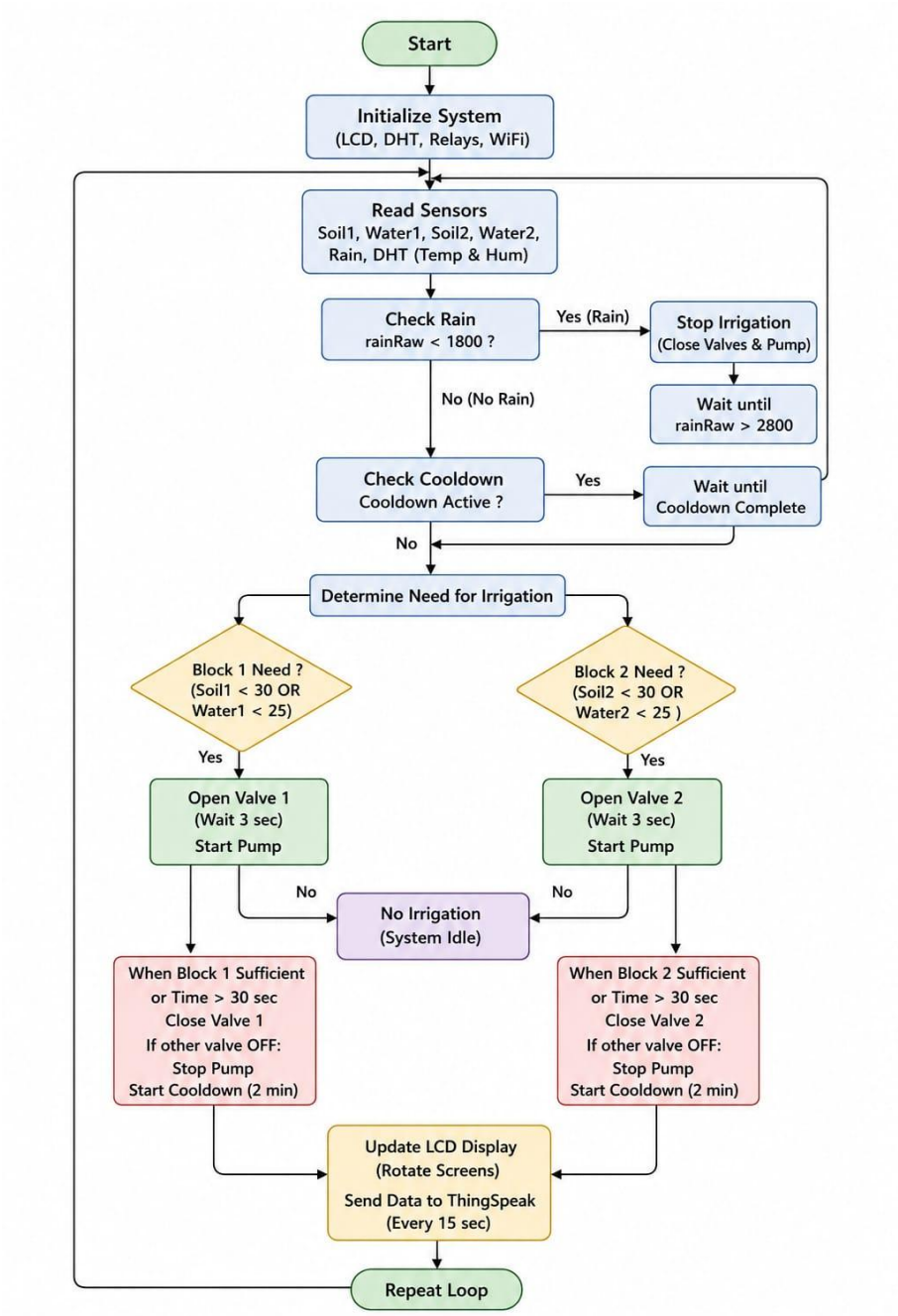

Fig. 4: Firmware control flow — rain check, cooldown gating, per-block need evaluation, valve/pump actuation, and LCD/cloud reporting. The per-cycle time limit in the implemented firmware was 300 s.

First, each block was tested on the testbed of Fig. 3 by inducing low soil-moisture and low water-level conditions at its sensors and observing the response of its valve and the shared pump. In addition, operational data were logged to a ThingSpeak channel. The analysis in this study uses a multi-field export of 3,291 entries covering 5–13 March 2026 (about 8 days) at the 15 s upload interval. An earlier export (8 May–18 August 2025, 1,924 entries) contains only two populated fields from sensor bring-up and is not used for the irrigation statistics.

Descriptive statistics (number of valid samples, minimum, mean, maximum and standard deviation) were computed for each logged field. Eighteen entries containing a non-numeric placeholder string were excluded as missing data. The Soil Moisture 2 and Rain Raw channels were excluded from interpretation because their logged values were affected by sensor connection, replacement, and calibration changes during the logging period. The pump state, a binary field, was summarized as the fraction of entries in which the pump was ON, and compared against Soil Moisture 1 for pump-ON and pump-OFF samples.

IV. Results

In every trial on each block, inducing low soil moisture or low water level caused the corresponding valve to open, and after the valve-to-pump delay, the shared pump to start and deliver water to the block through its valve. These results are based on direct observation during testing and are not a timestamped record.

Table 3 summarizes the logged field analysis. Soil Moisture 1 (mean 66.65%, SD 7.57) and Water Level 1 (mean 19.15, SD 8.72) show the variation expected from sensors responding to irrigation events.

Table 3: Descriptive statistics of logged fields, 5–13 March 2026 (3,291 entries).

Field	N	Min	Mean	Max	SD
Soil Moisture 1 (%)	3291	0.0	66.65	101.0	7.57
Water Level 1	3269	0.0	19.15	45.0	8.72
Water Level 2	3291	0.0	8.05	100.0	22.87

The pump state was strictly binary across all 3,291 entries and was ON in 94 entries (2.86%), a low duty cycle consistent with short, threshold-triggered irrigation bursts rather than continuous running. Mean Soil Moisture 1 during pump-ON samples (65.14%) was slightly below that during pump-OFF samples (66.70%), the direction expected when the pump is driven by low moisture; the difference is small over 94 pump-ON samples and is not statistically established.

V. Discussion

The prototype experimental setup shows that two blocks with independent soil-moisture and water-level sensing can be irrigated through dedicated valves from one shared pump without interfering with each other. The valve-before-pump delay ensures the existence of a flow path before the pump starts, the shutdown ordering keeps the pump running while the other block still requires water, and the per-block run-time limit and cooldown bound pump activity. In contrast to the single-zone systems summarized in the previous section, the above arrangement addresses the actuator sequencing problem that arises specifically when a pump is shared.

The low pump duty cycle (2.86%) indicates that irrigation was driven by threshold crossings and that the pump was idle for most of the logging periods. The small separation between Soil Moisture 1 during pump-ON and pump-OFF samples is consistent with short irrigation bursts that quickly restore moisture above the 30% threshold, but it cannot establish the size of the effect by itself.

Limitations of the present study are as follows:

- The irrigation statistics were obtained from a single 8-day window and were recorded with a firmware revision that preceded activation of the 120 s cooldown interlock. Therefore, they characterize threshold-driven irrigation without the cooldown gate and do not demonstrate cooldown-gated behavior or water savings relative to conventional irrigation.
- The Functional block-wise validation is observational rather than a logged, timestamped record.
- Temperature and humidity were sparsely logged in the available data, so no ambient-condition correlation is reported.
- The Soil Moisture 2 and Rain Raw values need re-calibration and re-logging before quantitative use.

VI. Conclusion

A dual-zone ESP32-based smart irrigation controller was designed, fabricated and operated on a physical testbed. Each block is sensed and actuated independently, rain inhibits irrigation, valve-before-pump sequencing and shutdown ordering protect the shared pump, and per-cycle run-time and cooldown limits bound pump activity. Block-wise tests confirmed correct valve and pump response to low soil moisture and low water level, and an 8-day cloud-logged window showed a 2.86% pump duty cycle. Future work includes a longer continuous logging campaign with the cooldown active, re-calibration of the Block 2 soil probe and the rain channel, denser temperature/humidity logging, and a controlled comparison against fixed-schedule irrigation to quantify water savings.

References

[1]. Shobana S, Sanjana Pandey B, Padmashri R, Triveni U. IoT based smart irrigation system using soil moisture sensor and ESP8266 NodeMCU. International Journal of Computer Science and Information Technology Research. 2021;9(2):52–58.

[2]. Morchid A, Jebabra R, Khalid HM, El Alami R, Qjidaa H, Ouazzani Jamil M. IoT-based smart irrigation management system to enhance agricultural water security using embedded systems, telemetry data, and cloud computing. Results in Engineering. 2024;23:102829.

[3]. Boukri et al. Analysis and experimental implementation of affordable smart irrigation system using IoT to reduce agricultural costs and minimize water usage. Applied Water Science. Springer Nature; 2026.
[4]. Pereira GP, Chaari MZ, Daroge F. IoT-enabled smart drip irrigation system using ESP32. IoT (MDPI). 2023;4(3):232–250.
[5]. Purnama NA, Abyakto R, Rosyid MGN, Irfawan DR, Soli M, Aribowo D. IoT-based irrigation control system with ESP32 for sustainable agriculture. Mars: Jurnal Teknik Mesin, Industri, Elektro dan Ilmu Komputer. 2024;2(6):211–224. doi:10.61132/mars.v2i6.556.
[6]. Shiva Shankar J, Palanivel S, China Venkateswarlu S. IoT based smart irrigation system by using ESP32 and Adafruit IO. In: Proceedings of the International Conference on Emerging Technologies: AI, IoT, and CPS for Science & Technology Applications. CEUR Workshop Proceedings, vol. 3058. NITTTR Chandigarh, India; 2021.