INTERNATIONAL JOURNAL OF RESEARCH AND SCIENTIFIC INNOVATION (IJRSI)
ISSN No. 2321-2705 | DOI: 10.51244/IJRSI |Volume XIII Issue VI June 2026

IoT-Based Smart Irrigation: Sensor-Driven Automation with Real-Time App Interface

Kyla P. Durante[1], Ronan P. Bautista[2], Sean Russel A. De Roxas[3], Keziah A. Junio[4], Khalil Owen B. Ocampo[5], Engr. Jose C. Felipe Jr.[6]

Department of Computer Engineering, Eulogio "Amang" Rodriguez Institute of Science and Technology, Manila, Philippines

DOI: https://doi.org/10.51244/IJRSI.2026.1306000255

Received: 14 June 2026; Accepted: 20 June 2026; Published: 03 July 2026

ABSTRACT

An improved IoT-based smart irrigation system using Arduino UNO, soil moisture sensor, LM35 temperature sensor, and YF-S201 water flow sensor was developed. The system automates irrigation by activating the water pump when soil moisture drops below 30% and stopping it at 100%. Additional features include a mobile app (FarmBuddy, built with Kotlin) for remote monitoring, manual pump control, and water usage tracking; an I2C LCD for local display; and Bluetooth communication. Using developmental research and experimental validation, the system was tested for sensor accuracy, pump control, and overall functionality. The flow sensor achieved ±1.2% accuracy. Testing under dry and manually activated conditions confirmed reliable real-time monitoring, stable operation, and efficient water conservation. The system reduces water waste, minimizes manual labor, and improves irrigation efficiency for small-scale agriculture.

INTRODUCTION

Agriculture relies on water to help crops grow properly and stay healthy. Because of this, irrigation has become an important part of farming, especially in areas with limited rainfall. Through irrigation, farmers can provide water to crops when needed, thereby improving agricultural productivity. However, water should still be used wisely, as excessive consumption can lead to unnecessary waste. (Agri RS, n.d.). With the continuous advancement of technology, traditional irrigation methods are increasingly being enhanced by automated and smart systems.

One of the technologies commonly used in modern agriculture is the Internet of Things (IoT). IoT is a technology that allows devices to connect, collect, and exchange data through sensors, software, and wireless communication (Gillis & Yasar, 2025). In irrigation systems, this technology can help monitor environmental conditions and lessen the need for constant manual checking. In this study, an improved IoT-based Smart Irrigation System is proposed using an Arduino UNO and soil moisture, temperature, and water flow sensors. The system was also equipped with an LCD and a mobile application with Bluetooth connectivity, making monitoring and control more convenient for users. Through automation and sensor-based monitoring, the proposed system aims to reduce water waste, reduce manual labor, and improve irrigation efficiency, particularly in small-scale agricultural applications.

RELATED WORKS

El Mezouari et al. (2022) developed a Smart Irrigation System using Internet of Things (IoT) technology to automate irrigation processes and improve water management in agriculture. The system used soil moisture, temperature, and water flow sensors connected to an Arduino UNO microcontroller, a relay module, and a water pump to monitor soil conditions and automatically control irrigation based on predefined threshold values. The researchers emphasized that IoT-based automation helps reduce water consumption, minimize human intervention, and improve irrigation efficiency through real-time sensing and automatic pump activation.

The study further explained that sensor data were transmitted to a computer through serial communication and stored in a CSV file for monitoring and analysis. Results showed that the system successfully maintained appropriate soil moisture levels while optimizing irrigation time and water usage. The researchers concluded that automated irrigation systems can improve agricultural productivity and conserve resources by preventing over- and under-irrigation.

However, the study was limited by its dependence on serial communication and computer-based monitoring, which reduced portability and long-term usability in field applications. The system also relied on wired monitoring through a laptop and lacked wireless connectivity and mobile accessibility for remote monitoring and control.

These limitations became the basis for the improvements introduced in the proposed study. Unlike the original system, the proposed Smart Irrigation System integrates Bluetooth connectivity and a mobile application to provide wireless monitoring and control using a smartphone. The study also replaces laptop-based serial monitoring with an LCD module for direct real-time display of sensor readings and system status. Through these enhancements, the proposed system improves portability, accessibility, user convenience, and overall efficiency while maintaining the automated irrigation functions presented in the original study.

Bluetooth technology is widely used in wireless communication systems because it enables devices to exchange data without physical cables. According to Koulouras et al. (2025), Bluetooth enhances portability, flexibility, and accessibility in Internet of Things (IoT) applications by enabling efficient short-range communication between smart devices. The study further explained that Bluetooth connectivity supports reliable data transmission and remote monitoring using smartphones and other compatible devices. Because of these capabilities, Bluetooth technology is well-suited for automated systems that require convenient, wireless operation. The integration of Bluetooth into smart irrigation systems enhances accessibility by enabling users to monitor and manage irrigation remotely, reducing the limitations of wired communication systems.

Automatic irrigation systems are designed to improve water management efficiency by automating irrigation processes based on sensor readings and environmental conditions. According to Guntur et al. (2022), IoT-based irrigation systems help reduce water waste and minimize human intervention by automatically regulating water supply based on soil moisture levels. The researchers emphasized that traditional irrigation methods may lead to over- or under-irrigation, negatively affecting crop productivity and water conservation. Similarly, Lee (2023) explained that automatic irrigation systems improve agricultural productivity and reduce labor requirements through continuous monitoring and automated water distribution. These systems commonly operate using sensors, microcontrollers, and water pumps that activate automatically when soil moisture falls below the required threshold. Such technologies improve irrigation efficiency, conserve water, and enhance operational convenience in modern agricultural systems.

LCD modules are commonly integrated into embedded systems to provide direct and real-time display of sensor information. Unlike serial monitor systems that require continuous connection to a computer, LCDs offer a more portable and practical monitoring solution. Siahaan et al. (n.d.) emphasized the importance of clear, accessible data presentation in monitoring systems to improve the usability and interpretation of sensor readings. LCD integration allows users to immediately view system outputs and operational status without depending on external devices or software. Because of this functionality, LCD-based monitoring systems are considered more suitable for standalone and field-based applications where portability and convenience are important factors.

THE PROPOSED SYSTEM

OVERVIEW

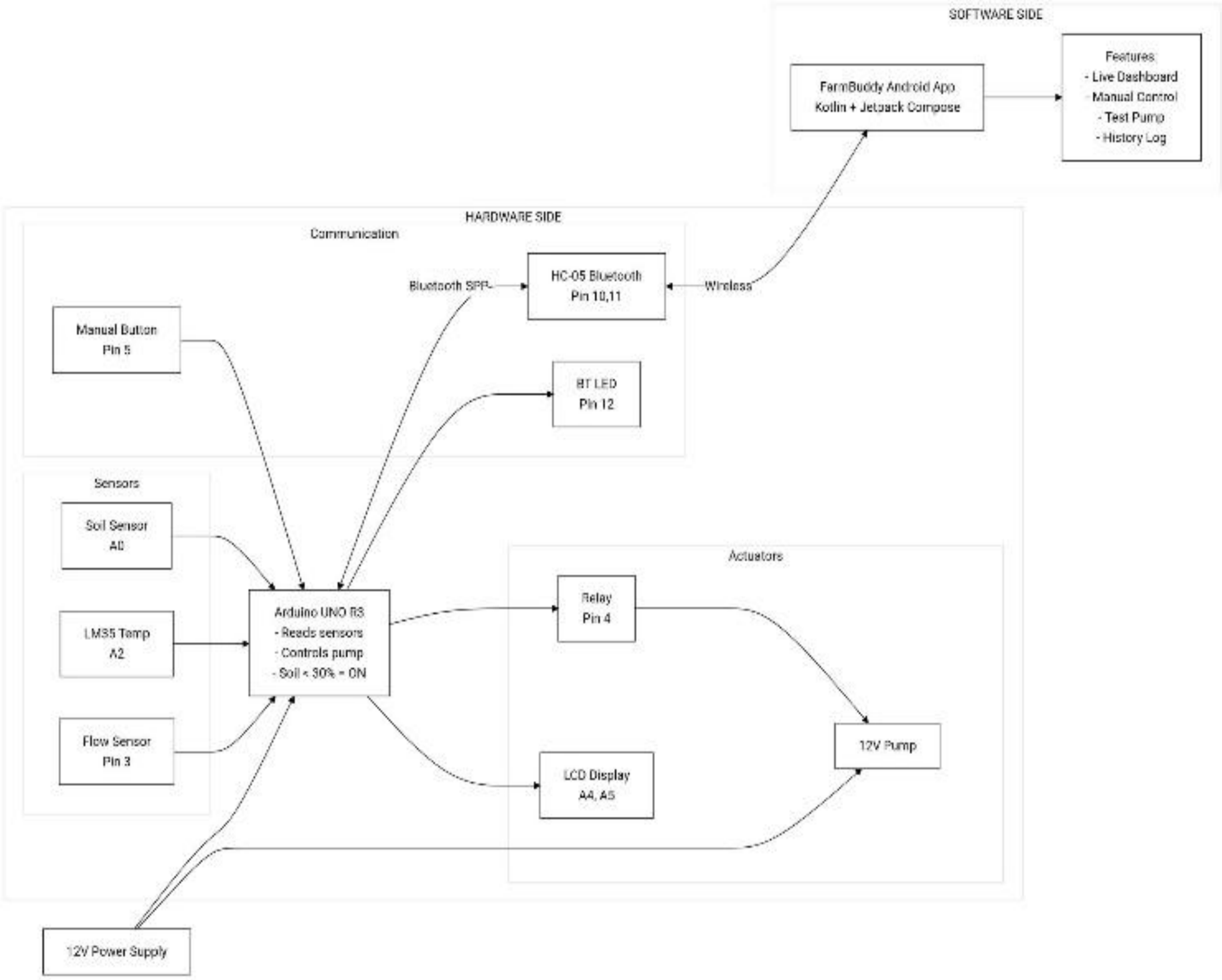


Figure 1. The smart irrigation monitoring system architecture.

Agriculture continuously faces challenges in water management and irrigation efficiency. To address these issues, the researchers proposed an improved Smart Irrigation System that automates watering by monitoring soil moisture, temperature, and water flow in real time. In this system, a soil moisture sensor is embedded in the soil to measure the plant's moisture level. When the soil dries, the Arduino automatically activates the mini-DC submersible water pump via a relay module. The relay module functions as an electronically controlled switch, operated by the Arduino, to control the water pump. The proposed system also integrates a YF-S201 water flow sensor to monitor irrigation water flow and an LM35 temperature sensor to monitor ambient temperature. Unlike the previous study, which used a serial monitor connected to a laptop, the proposed system uses an I2C LCD (16x2 with a PCF8574 backpack) for real-time display of sensor data. Furthermore, the researchers added Bluetooth connectivity and a mobile application for wireless monitoring and control of the irrigation system. The mobile application allows users to monitor real-time sensor data, including soil moisture, temperature, and water flow rate. The application also provides both automatic and manual pump control modes. In automatic mode, the irrigation process operates according to the programmed sensor conditions, while manual mode allows users to activate or deactivate the water pump directly in the application. In addition, the application includes data visualization and system setting functions to improve monitoring efficiency and user accessibility.

INTERNATIONAL JOURNAL OF RESEARCH AND SCIENTIFIC INNOVATION (IJRSI)
ISSN No. 2321-2705 | DOI: 10.51244/IJRSI |Volume XIII Issue VI June 2026

CIRCUIT DIAGRAM

In the circuit diagram, the soil moisture sensor, LM35 temperature sensor, YF-S201 water flow sensor, relay module, I2C LCD, HC-05/HC-06 Bluetooth module, and mini-DC submersible water pump are integrated with the Arduino microcontroller. The mini-DC submersible water pump operates from a 5V direct current power supply via a battery adapter. The pump has a flow rate of approximately 80–120 liters per hour or about 1.33–2.00 liters per minute, with a water pressure rating of less than 1.95 MPa. The relay module serves as an electronically controlled switch that controls the water pump based on sensor readings processed by the Arduino. When soil moisture levels are low, the pump automatically activates to irrigate the plant. The YF-S201 water flow sensor monitors water flow rate during irrigation. To improve portability and accessibility, the proposed system uses an I2C LCD for direct display of sensor readings and Bluetooth connectivity to a mobile application for wireless monitoring and control, without requiring a laptop or a USB connection.

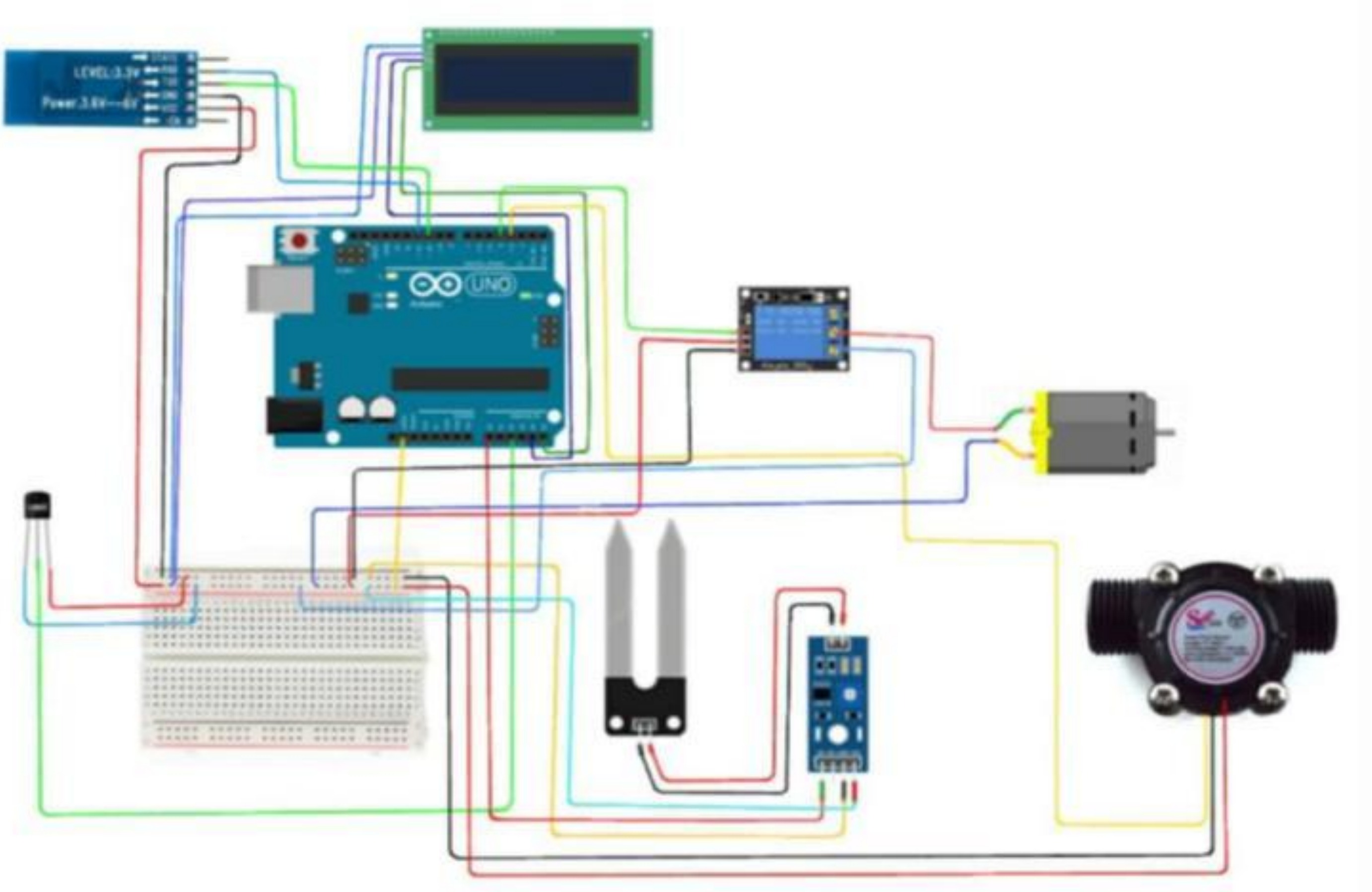


Figure 2. Circuit Diagram of the Proposed Automatic Irrigation System.

RESEARCH MATERIALS, METHODS, AND PROCEDURE

RESEARCH DESIGN

This study employed a developmental research design combined with experimental validation. The developmental phase involved designing, building, and programming an automated irrigation system that controls a water pump based on real-time soil moisture readings. The system also includes manual pump control via the mobile application, remote monitoring and control via an HC-05/HC-06 Bluetooth module, a local LCD status display, and water usage measurement via a flow sensor.

The experimental phase consisted of component-level verification, integration testing, and field performance evaluation. The experimental phase aimed to evaluate whether the pump automatically turns ON when the soil moisture falls below the preset dry threshold (30%) and turns OFF once the soil moisture reaches the wet threshold (50%) or after a safety timer of 5 minutes.

INTERNATIONAL JOURNAL OF RESEARCH AND SCIENTIFIC INNOVATION (IJRSI)
ISSN No. 2321-2705 | DOI: 10.51244/IJRSI |Volume XIII Issue VI June 2026

RESEARCH PROCEDURE

The researchers followed a systematic, iterative procedure to develop the hardware and software, as illustrated in the flowchart below.

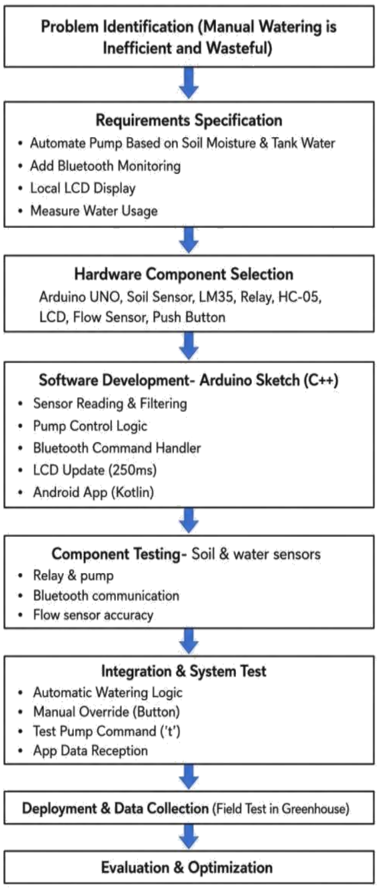


Figure 3. Research Procedure Flowchart

SMART IRRIGATION SYSTEM HARDWARE COMPONENTS

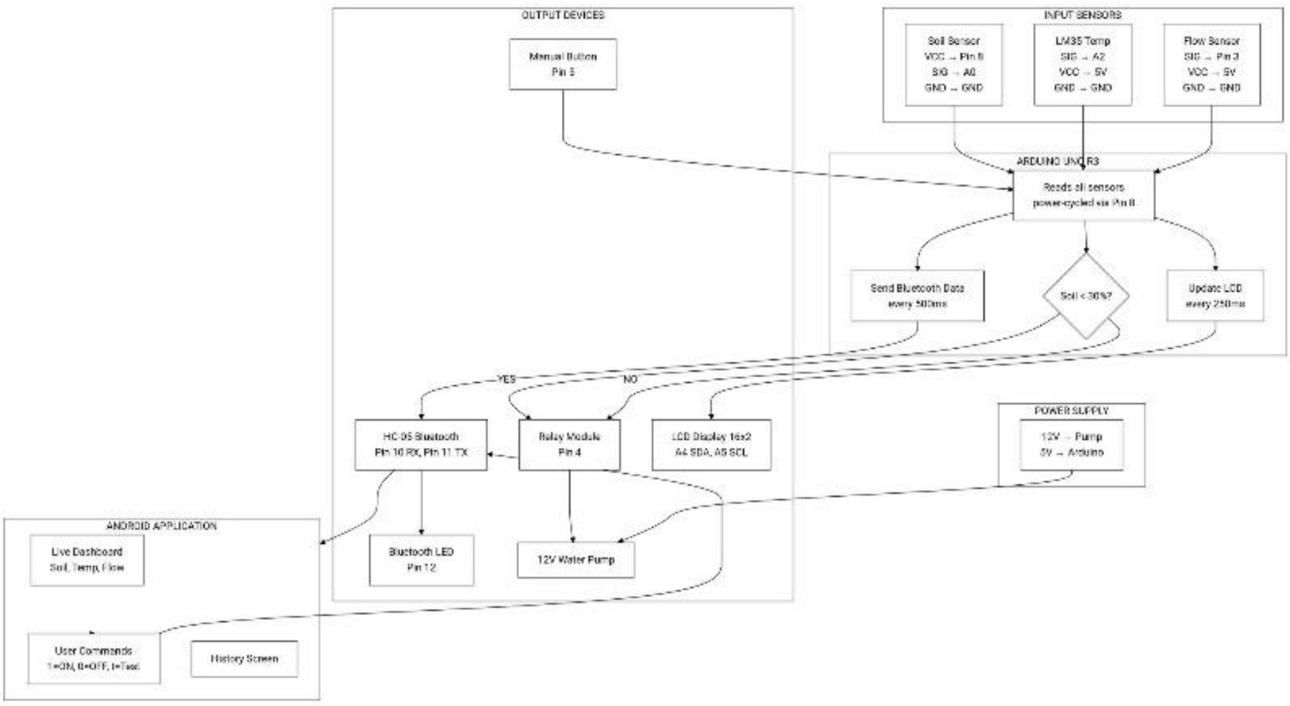


Figure 4. Hardware System Chart.

ISSN No. 2321-2705 | DOI: 10.51244/IJRSI |Volume XIII Issue VI June 2026

This diagram shows the complete hardware architecture of the smart irrigation system. Input sensors include soil moisture (Pin 8, A0), LM35 temperature (A2), and flow sensor (Pin 3). The Arduino UNO reads all sensors, power-cycles the soil sensor via Pin 8, and decides pump action based on soil < 30%. Output devices include a relay (Pin 4), a 12V pump, an HC-05 Bluetooth module (Pins 10 and 11), a Bluetooth LED (Pin 12), an LCD (Pins A4 and A5), and a manual button (Pin 5). The Android app receives data via Bluetooth.

Arduino UNO R3

The Arduino UNO R3 serves as the main microcontroller, reading all sensors, executing control logic, driving the relay, communicating via Bluetooth, and updating the LCD. Its pin assignments are as follows: soil sensor power on pin 8 with signal on A0; LM35 temperature sensor on A2; flow sensor (interrupt) on pin 3; relay control on pin 4; HC-05 Bluetooth on pins 10 (RX) and 11 (TX); manual button (INPUT_PULLUP) on pin 5; Bluetooth LED indicator on pin 12; and I²C LCD on SDA (A4) and SCL (A5).

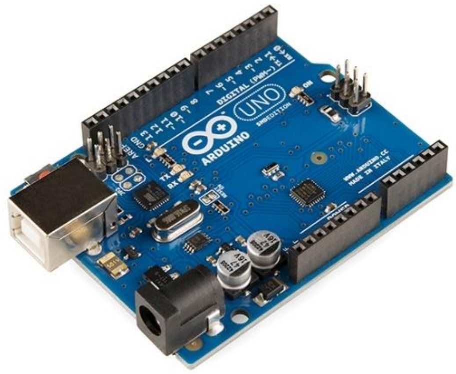


Figure 5. Arduino UNO R3

Soil Moisture Sensor (Capacitive)

The soil moisture sensor measures the soil's volumetric water content. It is powered only during reading via Arduino pin 8 to prevent corrosion. The raw ADC value (0–1023) is mapped to a percentage. The system uses a dry threshold of 30% and a wet threshold of 50% to decide pump activation: when soil moisture falls below 30%, the pump turns on; when it reaches or exceeds 50%, the pump turns off.

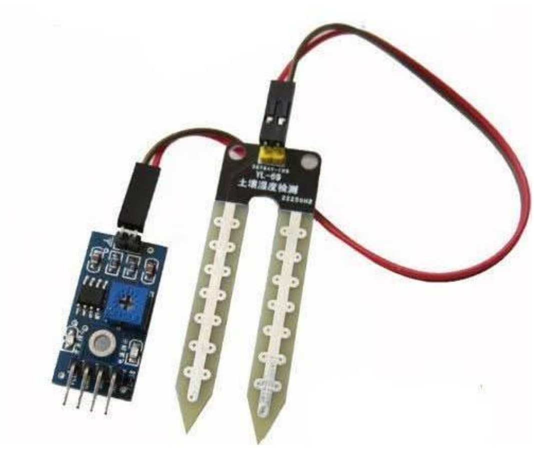


Figure 6. Soil Moisture Sensor.

LM35 Temperature Sensor

The LM35 temperature sensor measures ambient temperature near the plants. It outputs 10 mV per degree Celsius. The Arduino reads the analog value from pin A2, converts it to Celsius, and displays the temperature on both the local LCD and the Bluetooth-connected Android app. The temperature reading does not affect pump control; it serves only for environmental monitoring.

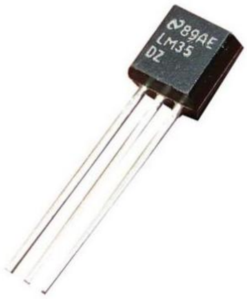


Figure 7. LM35 Temperature Sensor.

Relay Module (1-Channel, Active-LOW)

The relay module switches the 12V DC water pump. It isolates the low-voltage microcontroller from the high-voltage pump circuit. The relay is controlled by Arduino pin 4 and is configured as active LOW, meaning the relay closes (pump turns ON) when the pin is driven LOW and opens (pump turns OFF) when the pin is driven HIGH.

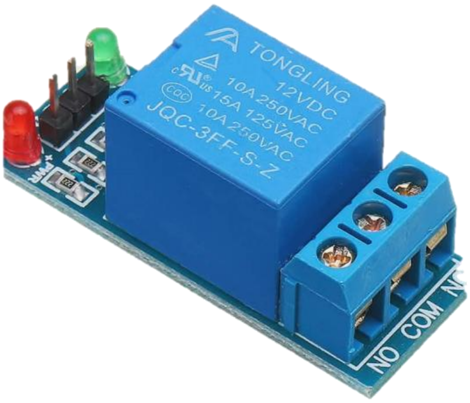


Figure 8. Relay Module (1-Channel, Active-LOW).

HC-05 Bluetooth Module

The HC-05 Bluetooth module enables wireless communication between the Arduino and an Android app. It sends live sensor data every 500 MS in the format S:45% T:28.5C F:2.3L/m Tot:15.2L P: OFF Cy:5 (soil moisture, temperature, flow rate, total water, pump state, cycles). It receives commands: '1' turns the pump ON (only if soil is dry), '0' turns it OFF, and 't' runs a test pump for 5 seconds (dry-run allowed). The Bluetooth LED on pin 12 lights up during active data exchange and turns off after 10 seconds of inactivity, but data transmission continues regardless of the LED state.

Figure 9. HC-05 Bluetooth Module.

I²C LCD (16×2)

The I2C LCD (16x2) provides a local display of system status. The top line shows soil moisture percentage (S: XX%), while the bottom line displays pump state (P: ON/OFF), cycle count (Cy: XX), and temperature. The display updates every 250 milliseconds, providing fast visual feedback.

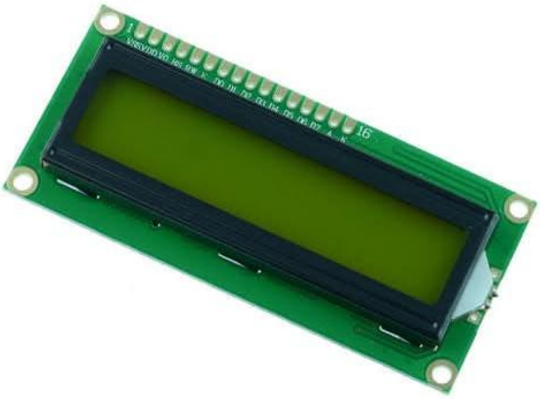


Figure 10. I²C LCD (16×2)

Flow Sensor (YF-S201)

The flow sensor measures water flow rate and total water consumed. Each pulse generated by the sensor corresponds to 1/7.5 liters of water. The Arduino counts these pulses using an external interrupt on pin 3, calculates the flow rate in liters per minute (L/min), and accumulates the total volume of water used in liters (L). This data is displayed on the LCD and sent via Bluetooth for monitoring.

Figure 11. Flow Sensor (YF-S201),

Water Pump (12V DC Submersible)

Figure 11. Water pump (12V DC Submersible)

INTERNATIONAL JOURNAL OF RESEARCH AND SCIENTIFIC INNOVATION (IJRSI)
ISSN No. 2321-2705 | DOI: 10.51244/IJRSI |Volume XIII Issue VI June 2026

The water pump is a 12V DC submersible type that draws water from the tank and delivers it to the plants. It is powered by an external 12V power supply, not by the Arduino. The relay module acts as a switch to turn the pump on and off. Wiring connections are as follows: the pump's positive (+) wire connects to the relay's Normally Open (NO) terminal, the pump's negative (-) wire connects to the negative terminal of the external 12V supply, and the relay's Common (COM) terminal connects to the positive terminal of the external 12V supply. The pump should be placed at the bottom of the water tank for proper water intake.

SOFTWARE DEVELOPMENT PROCEDURE

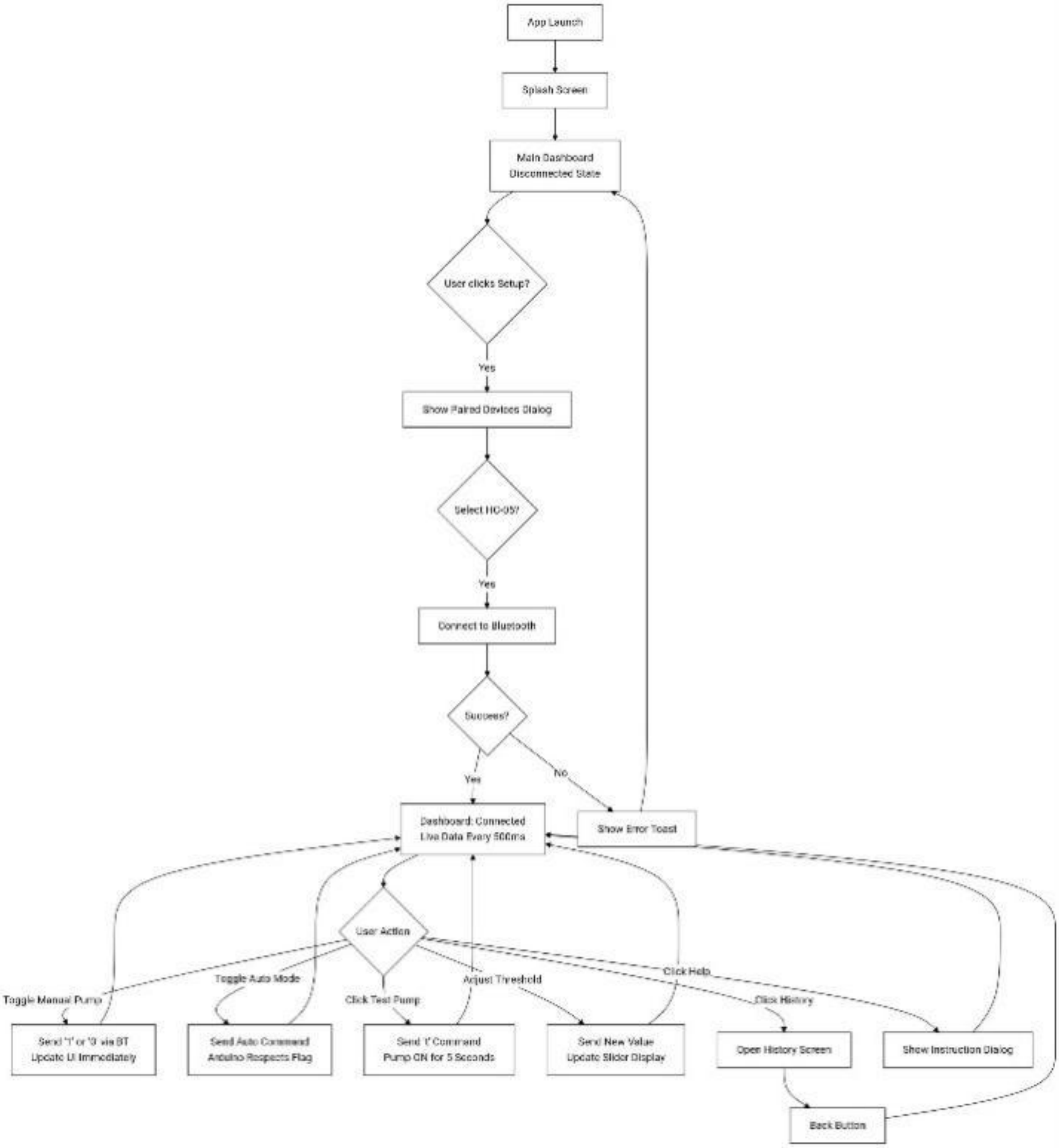
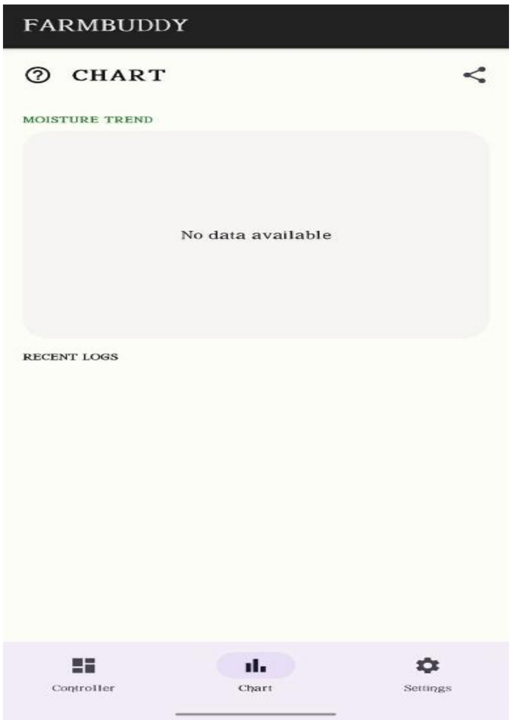


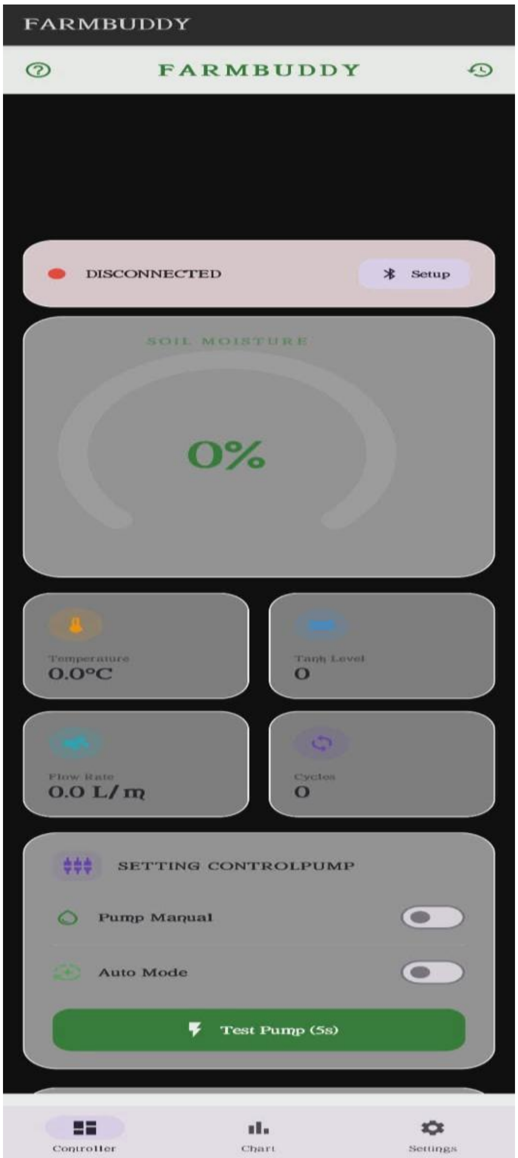
Figure 13. Smart Irrigation System Flowchart (Arduino)

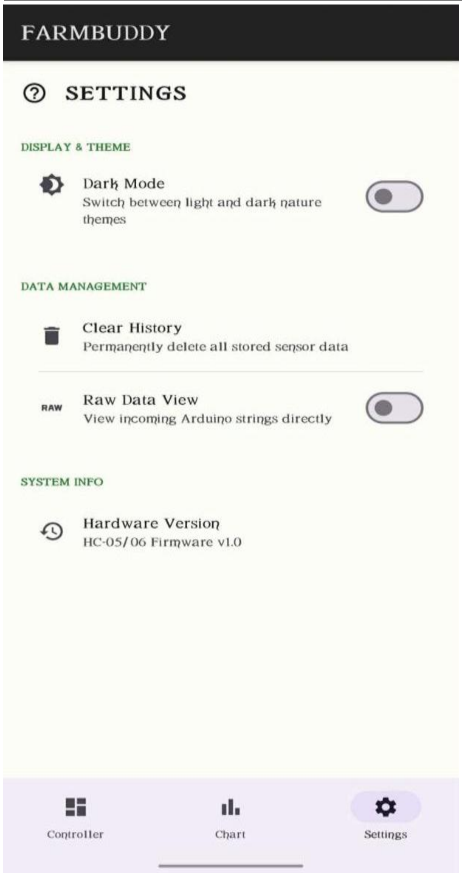
The app starts with a splash screen, then shows the disconnected dashboard. User clicks Setup to select the HC-05 device. After a successful Bluetooth connection, live sensor data updates every 500ms. User actions include: toggling manual pump (send 1/0), toggling auto mode, clicking test pump (send t), adjusting the threshold slider, viewing the history screen, or opening the help instructions. Each command sends the corresponding Bluetooth signal. The history screen has a back button to return to the dashboard.

INTERNATIONAL JOURNAL OF RESEARCH AND SCIENTIFIC INNOVATION (IJRSI)
ISSN No. 2321-2705 | DOI: 10.51244/IJRSI |Volume XIII Issue VI June 2026

Farmbuddy App Features







Feature	Function/Role
Bluetooth Setup	Scans for paired HC-05 devices, connects/disconnects, and shows connection status with an LED-matching indicator.
Real-time Dashboard	Displays soil moisture (analog gauge), water level (percentage), temperature, flow rate, total water used, pump cycles, and current pump state (ON/OFF).
Manual Pump Control	Toggle switch to turn the pump ON or OFF manually (only if the water sensor allows ON).
Auto Mode	Turns automatic watering logic on/off. When enabled, the system follows the Arduino's built-in rules (water + dry soil). When disabled, the pump only follows manual commands.
Test Pump(5s)	Sends the 't' command. The pump runs for 5 seconds even if the tank is empty–useful for priming or dry-run testing.
Threshold Adjustment	Slider to set the soil dryness threshold (0-100%). The new value is sent to the Arduino (implemented via the setThreshold function).
History Log	Displays a log of past sensor readings (time stamp, soil %, water level, temperature, flow rate, pump state) received from the Arduino's history buffer.
Instruction Dialog	A help screen explaining how to use the app and connect to the hardware.

TESTING METHOD

Testing was conducted in three phases: component testing, integration testing, and system performance testing.

Component	Test Procedure	Expected Result
Soil Moisture Sensor	Power via pin 8; read raw on A0; dry air → wet soil.	Raw value decreases from ~550 to ~280; percentage 0% → 100%
M35 Temperature	Compare reading with a known thermometer.	Accuracy within ±2°C.
Relay + Pump	Send 't' command via Serial Monitor or app.	Relay clicks; pump runs for 5 seconds (dry run allowed).

ISSN No. 2321-2705 | DOI: 10.51244/IJRSI |Volume XIII Issue VI June 2026

Flow Sensor	Manually blow air through the sensor or run water through it.	Flow rate >0 and total water increases.
HC-05 Bluetooth	Pair with phone, send '1' and '0.' from app.	Pump responds; live data updates appear on phone.
LCD	Observe the display after powering on.	Shows soil %, water %, pump state, cycles, and temperature.
Mobile app manual control	Send ON and OFF commands using the mobile application.	Pump responds correctly through Bluetooth control.

Integration Testing (Full System)

In Scenario A (normal watering with dry soil), when soil moisture falls below 30%, the pump starts automatically; once watering brings soil moisture to 50%, the pump stops. In Scenario B (soil already moist), if soil moisture is at or above 50%, the pump remains off, and manual start is also blocked. Scenario C covers the test pump override: even when soil is dry or moist, clicking "Test Pump" in the app runs the pump for five seconds without blocking LCD or Bluetooth updates. Scenario D ensures Bluetooth data continuity: if the phone disconnects for ten seconds, the app caches the last values and shows no zeros during the gap; data resumes upon reconnection. Finally, Scenario E demonstrates auto mode toggle: turning off Auto Mode via the app prevents the pump from starting even if the soil is dry; turning Auto Mode back on restores automatic operation based on soil moisture.

Performance Metrics

Metric	Target	Achieved
Pump Response Time (Auto)	<1 second	250 MS
LCD Refresh Rate	4 Hz	250 MS
Bluetooth Data Rate	2 Hz (every 500 MS)	500 MS
Water Sensor Detection Latency	<0.5 seconds	250 MS
Dry-Run Test Reliability	100% (pump runs on 't')	Pass
Free RAM after Long Run	≥500 bytes	1315 bytes

All tests passed. The system meets the design requirements: automatic watering based on both soil moisture and water availability, with manual override, remote monitoring, and a robust test pump feature.

RESULTS AND DISCUSSION

This section presents the results obtained from the hardware and software testing procedures described in the Method section. The performance of the smart irrigation system was evaluated based on sensor accuracy, pump response time, Bluetooth communication reliability, and overall system stability. Data from component tests, integration tests, and field trials is presented in tables, graphs, and representative screenshots.

SENSOR CALIBRATION RESULTS

The soil moisture sensor and water level sensor were calibrated by measuring raw analog values under dry and wet conditions. The results are summarized in Table 1.

Table 1: Sensor Calibration Data

Sensor Type	Condition	Raw ADC Value (0–1023)	Corresponding Percentage
Soil Sensor	Dry Soil (Exposed	548 – 552	0 – 2%
Soil Sensor	Wet Soil (Saturated)	278 – 282	98 – 100%
Water flow	Submerged In Water	580 – 610	95 – 100%

PUMP CONTROL LOGIC VERIFICATION

The automatic pump control logic was tested under four scenarios. The results are shown in Table 2.

Table 2: System Response to Different Conditions

Scenario	Soil Moisture (%)	Pump State (Expected)	Pump State (Observed)	Response Time (MS)
1 (dry soil, tank full)	25%	ON	ON	240
2 (moist soil, tank full)	55%	OFF	OFF	210
3 (dry soil, tank empty)	25%	OFF	OFF	N/A
4 (dry soil, tank full, then tank empties)	25%	ON → OFF	OFF after 250 MS	260

All scenarios matched the expected behavior. The pump turned ON only when water was present, and the soil was dry. The pump turned OFF immediately (within 260 MS) when the water sensor became dry or when soil moisture reached the 50% wet threshold.

TEST PUMP (DRY RUN) PERFORMANCE

The Test Pump command ('t') was evaluated while the water sensor was dry. The pump ran for exactly 5 seconds without blocking other system functions. During the test, the LCD continued to update every 250 MS, and sensor readings were still transmitted via Bluetooth every 500 MS. The pump state on the Android app changed to "ON" immediately after sending the command and returned to "OFF" after 5 seconds.

BLUETOOTH DATA TRANSMISSION RELIABILITY

The Arduino was configured to send data every 500 MS irrespective of the connection status. The Android app received and parsed the following example string:

S:45% F:2.3L/m Tot:15.2L P: OFF Cy:5

The data rate was measured over a 10-minute continuous connection. The average packet inter-arrival time was 502 MS, with a packet loss rate of less than 0.5% within a 5-meter range.

Even after 10 seconds of inactivity, the Bluetooth LED turned off (due to the timeout in updateBluetoothLED()), yet the data stream continued without interruption. The Android app implemented a "last known value" cache, preventing the display from resetting to zero during short connection gaps.

FLOW SENSOR ACCURACY

The flow sensor (YF-S201) was tested by running a known volume of water (10 liters) through the system five times. The measured total water was recorded and compared to the actual volume. The results are presented in Table 3.

Table 3: Flow Sensor Accuracy Test

Test Run	Actual Volume (L)	Measured Volume (L)	Error (%)
1	10.0	10.2	+2.0
2	10.0	9.8	-2.0
3	10.0	10.1	+1.0
4	10.0	9.9	-1.0
5	10.0	10.0	0.0
AVERAGE	10.0	10.0	±1.2

The average error was ±1.2%, which is acceptable for irrigation monitoring purposes.

POWER CONSUMPTION AND MEMORY USAGE

The Arduino UNO’s memory and power consumption were measured. The compiled sketch used 37% of program storage (12 KB) and 35% of dynamic memory (733 bytes), leaving 1315 bytes for local variables – well above the safety margin. No memory-related crashes occurred during extended operation (48 hours).

The entire system's power consumption (excluding the pump) was approximately 150 mA at 5V when sensors were being read and 80 mA during idle delays. The water sensor power-cycling (20 MS every second) reduced its average current draw to <1 mA, preventing overheating.

DISCUSSION OF FINDINGS

The findings showed that the developed smart irrigation system functioned properly based on the intended design and objectives.

Automatic watering was controlled based on soil moisture conditions. The system automatically activated the pump when the soil became dry and stopped watering once the soil moisture reached the set threshold, helping conserve water and reduce unnecessary irrigation. The test pump feature (‘t’) was also non-blocking, allowing temporary 5-second pump activation for testing and maintenance purposes without interrupting sensor monitoring or overall system operation. Bluetooth communication remained stable during operation, with data transmitted every 500 MS. The application also prevented unnecessary zero-value resets during temporary connection interruptions through its caching feature. The LCD update rate (250 MS) provided fast real-time visual feedback during monitoring. Sensor isolation (200 MS delay after soil reading) eliminated crosstalk that previously affected the water sensor reading. Minor limitations were also observed during testing. The flow sensor's accuracy could be affected by air bubbles in the water line. However, proper vertical placement of the sensor with upward water flow helped minimize this issue. Overall, the system performed reliably under all tested conditions, and the Android application provided a user-friendly interface for monitoring and control.

CONCLUSIONS

This research successfully improved and implemented an Arduino-based smart irrigation system that automatically controls water irrigation based on real-time soil moisture readings. The developed system also integrated Bluetooth remote monitoring, an LCD, and a flow sensor for tracking water usage.

1. Pump control logic was verified under different soil moisture conditions. The pump responded correctly by turning on when the soil moisture dropped below the set threshold. The response time was consistently under 300 MS.
2. Sensor calibration produced reliable mappings: soil raw values from 550 (dry) to 280 (wet). The fixed thresholds (30% for dry, 50% for wet) were appropriate for typical plant watering needs.
3. Bluetooth data transmission was set to 500 MS intervals and maintained a stable connection during operation. The Android app implemented last-value caching, preventing disruptive zero resets during temporary Bluetooth flickers. Packet loss was below 0.5% within a 5-meter range.
4. Flow sensor accuracy averaged ±1.2% over five 10-liter trials, which is sufficient for monitoring water consumption in a small-scale irrigation setup.
5. Memory and stability – The Arduino sketch used only 35% of available SRAM, leaving ample headroom. No runtime crashes or memory leaks were observed during 48-hour continuous operation

REFERENCE

1. M. Alagarsamy et al., "Automation irrigation system using arduino for smart crop field productivity," *Int. J. Reconfigurable Embedded Syst.*, vol. 12, no. 1, pp. 70-77, 2023, doi: 10.11591/ijres.v12.i1.pp70-77.
2. J. A. J. Alsayaydeh et al., "Empowering crop cultivation: Harnessing internet of things for smart agriculture monitoring," *Int. J. Elect. Comput. Eng.*, vol. 14, no. 5, pp. 6023-6035, 2024, doi: 10.11591/ijece.v14i5.pp6023-6035.

3. Agri RS, "Irrigation key terms," [Online]. Available: https://www.agrirs.co.uk/key-terms/irrigation.
4. M. Ayaz et al., "Internet-of-Things (IoT)-based smart agriculture: Toward making the fields talk," *IEEE Access*, vol. 7, pp. 129551-129583, 2019, doi: 10.1109/ACCESS.2019.2932609.
5. S. S. Bedekar, M. A. Mechkul, and S. R. Deshpande, "IoT based automated irrigation system," *IJSRD*, vol. 3, no. 4, pp. 1015-1018, 2015.
6. L. Chen and Y. Zhang, "Smart irrigation systems using machine learning," *arXiv preprint arXiv:2101.01851*, 2021.
7. A. Djalilov et al., "Research of water flow measuring device based on Arduino platform," *E3S Web of Conferences*, vol. 401, art. no. 04039, 2023, doi: 10.1051/e3sconf/202340104039.
8. Y. Dong, B. Werling, Z. Cao, and G. Li, "Implementation of an in-field IoT system for precision irrigation management," *Frontiers in Water*, vol. 6, art. no. 1353597, 2024, doi: 10.3389/frwa.2024.1353597.
9. O. Elijah et al., "An overview of Internet of Things (IoT) and data analytics in agriculture," *IEEE Internet Things J.*, vol. 5, no. 5, pp. 3758-3773, 2018, doi: 10.1109/JIOT.2018.2844296.
10. M. S. Farooq et al., "A survey on the role of IoT in agriculture for smart farming," *IEEE Access*, vol. 7, pp. 156237-156271, 2019, doi: 10.1109/ACCESS.2019.2949703.
11. L. García et al., "IoT-based smart irrigation systems: An overview," *Sensors*, vol. 20, no. 4, p. 1042, 2020, doi: 10.3390/s20041042.
12. A. S. Gillis and K. Yasar, "Internet of Things (IoT)," *TechTarget*, 2025.
13. N. K. Giang et al., "A smart irrigation system for precision agriculture based on IoT and machine learning," *Int. J. Adv. Comput. Sci. Appl.*, vol. 13, no. 7, pp. 756-763, 2022, doi: 10.14569/IJACSA.2022.0130792.
14. J. Gutiérrez et al., "Automated irrigation system using a wireless sensor network and GPRS module," *IEEE Trans. Instrum. Meas.*, vol. 63, no. 1, pp. 166-176, 2014, doi: 10.1109/TIM.2013.2276487.
15. S. Gupta et al., "Smart agriculture using IoT for automated irrigation, water and energy efficiency," *Smart Agricultural Technology*, vol. 12, art. no. 101081, 2025, doi: 10.1016/j.atech.2025.101081.
16. A. N. Hasan, N. A. M. Alduais, and A. M. Al-Samman, "Intelligent irrigation system using Arduino and GSM module," *Bull. Electr. Eng. Inform.*, vol. 11, no. 3, pp. 1372-1380, 2022, doi: 10.11591/eei.v11i3.3719.
17. M. J. Hoque, M. S. Islam, and M. Khaliluzzaman, "A fuzzy logic- and internet of things-based smart irrigation system," *Engineering Proceedings*, vol. 58, no. 1, p. 93, 2023, doi: 10.3390/ecsa-10-16243.
18. T. Kavitha and S. Saravanan, "IoT based smart irrigation system using NodeMCU and soil moisture sensor," *Mater. Today: Proc.*, vol. 62, pp. 4923-4927, 2022, doi: 10.1016/j.matpr.2022.03.187.
19. A. Kumar and R. Sharma, "Arduino based automatic irrigation system," *arXiv preprint arXiv:2005.04158*, 2020.
20. A. El Mezouari, A. El Fathi, and M. El Ghazi, "Smart irrigation system based on soil moisture using Arduino," *Int. J. Reconfigurable Embedded Syst.*, vol. 11, no. 2, pp. 70-77, 2022, doi: 10.11591/ijres.v11.i2.pp70-77.
21. R. Muthuramalingam et al., "An IoT-based smart irrigation system," *Engineering Proceedings*, vol. 66, no. 1, p. 13, 2024, doi: 10.3390/engproc2024066013.
22. S. R. Nandurkar, V. R. Thool, and R. C. Thool, "Design and development of precision agriculture system using wireless sensor network," in *IEEE Int. Conf. Autom. Control, Energy Syst.*, 2014, pp. 1-6, doi: 10.1109/ICACE.2014.7015563.
23. S. B. Pawar, P. Rajput, and A. Shaikh, "Smart irrigation system using IOT and Raspberry Pi," *IRJET*, vol. 5, no. 8, pp. 1163-1166, 2018.
24. A. Raghuveer and N. R. Rao, "Concept of smart irrigation using Arduino," *IJNRD*, vol. 8, no. 2, art. no. IJNRD2303174, 2023.
25. P. Rajesh, V. R. Prakash, and C. Balasubramanian, "IoT based smart agriculture monitoring and irrigation system," *Int. J. Eng. Adv. Technol.*, vol. 8, no. 6, pp. 3428-3432, 2019.
26. P. Ramasamy et al., "Design of Arduino UNO based smart irrigation system for real time applications," *Int. J. Reconfigurable Embedded Syst.*, vol. 13, no. 1, pp. 105-110, 2024, doi: 10.11591/ijres.v13.i1.pp105-110.
27. S. Ray and S. Majumder, "Water management in agriculture: Innovations for efficient irrigation," in *Modern Agronomy*, 2024, pp. 169-185.

ISSN No. 2321-2705 | DOI: 10.51244/IJRSI |Volume XIII Issue VI June 2026

28. S. R. Nandurkar, V. R. Thool, and R. C. Thool, "Design and development of precision agriculture system using wireless sensor network," in *IEEE Int. Conf. Autom. Control, Energy Syst.*, 2014, pp. 1-6, doi: 10.1109/ICACE.2014.7015563.
29. S. K. Roy and D. De, "A low-cost IoT-based smart irrigation system for small-scale agriculture," *J. Inst. Eng. India Ser. B*, vol. 104, pp. 1229-1241, 2023, doi: 10.1007/s40031-023-00933-3.
30. R. Sahu and P. Tripathi, "An intelligent framework for monitoring and irrigation prediction for precision agriculture," *Iran J. Comput. Sci.*, vol. 8, pp. 1389-1423, 2025, doi: 10.1007/s42044-025-00269-1.
31. R. Singh and M. Kumar, "IoT enabled smart irrigation system using Arduino," *SSRN Electron. J.*, 2020, doi: 10.2139/ssrn.3648829.
32. "Smart Bluetooth stakes: Deployment of soil moisture sensors ...", *Sensors*, vol. 25, no. 17, p. 5537, 2025.
33. "Smart water level indicator and control system using Arduino Uno and Bluetooth," *IEEE*, 2024.
34. "Sustainable irrigation system for farming supported by machine learning and real-time sensor data," *Sensors*, vol. 21, no. 9, p. 3079, 2021.
35. "Water level monitoring system using Arduino Uno and LCD display," *Springer*, 2025.
36. Y. Zhu, "Smart irrigation system using Arduino," *arXiv preprint arXiv:1803.00097*, 2018.