INTERNATIONAL JOURNAL OF NOVEL RESEARCH AND DEVELOPMENT (IJNRD)
© 2026 IJNRD | Volume 11, Issue 5, May 2026 | ISSN: 2456-4184 | IJNRD.ORG

Smart Irrigation System

Sanket Phatak[1], Sakshi Sadamate[2], Khushi Mulani[3], Sanika Desai[4], Amol Desai[5]
Project Guide: Mrs. A. S. Yadav[6]
[1–5]B.Tech Electrical Engineering (Final Year) [6]Project Guide

Jaywant College of Engineering & Polytechnic, K. M. Gad, Maharashtra, India

---

Abstract—Water scarcity and improper irrigation scheduling continue to reduce crop productivity globally. This paper presents a solar-powered Smart Irrigation System that automates water delivery based on real-time soil moisture sensing. Four capacitive soil moisture sensors (MS1–MS4) are monitored by an Arduino UNO R3 microcontroller, which actuates four solenoid valves and a BLDC water pump through a five-channel relay module. An HC-05 Bluetooth module paired with a custom MIT App Inventor mobile application enables wireless switching between Automatic (sensor-driven) and Manual (operator-controlled) modes and provides a live status dashboard for all actuators, refreshed every two seconds. The system is powered entirely by a 50 W solar panel through a PWM charge controller and 12 V/7 Ah lead-acid battery, allowing off-grid operation. Experimental field testing verified reliable zone-wise irrigation control and mobile override functionality within a 10-metre Bluetooth range. The proposed system reduces water consumption, eliminates manual intervention, and provides a low-cost, portable, and renewable-energy-based solution suitable for small to medium farms in developing regions.

Keywords—*smart irrigation; soil moisture sensor; Arduino UNO; solar energy; HC-05 Bluetooth; MIT App Inventor; solenoid valve; relay; sustainable agriculture.*

I. INTRODUCTION

Agriculture is the foundation of food security and rural livelihoods, particularly in developing nations such as India, where approximately 58% of the population depends on farming [1]. Traditional flood and furrow irrigation methods are inefficient, often resulting in 40–70% water losses through evaporation, runoff, and percolation [2]. With increasing water stress caused by irregular monsoons, declining groundwater tables, and rising food demand, there is an urgent need for precision irrigation technologies that deliver water only when and where crops require it.

The system proposed in this paper addresses this challenge through three integrated subsystems: (i) a renewable solar power unit providing complete energy autonomy; (ii) a sensor-driven automatic irrigation controller using an Arduino UNO microcontroller and four soil moisture sensors; and (iii) a Bluetooth-enabled mobile application offering real-time monitoring and manual override. The system is mounted on a portable four-wheel trolley, enabling deployment across multiple field sections without fixed infrastructure.

The primary contributions of this work are:

- A fully solar-powered, off-grid irrigation system with zero reliance on grid electricity.
- Zone-wise independent solenoid valve control based on per-zone moisture readings.
- A custom Android application (MIT App Inventor) with live actuator status dashboard and mode-switching via HC-05 Bluetooth.
- A portable, low-cost hardware prototype validated through field-level testing.

II. LITERATURE REVIEW

Chandan Kumar Sahu and Pramitee Behera [3] proposed an Arduino-based low-cost irrigation prototype using soil moisture sensors, a relay module, and a GSM module for SMS alerts. Their system demonstrated effective automatic valve control but lacked wireless real-time monitoring. Veena Divyak et al. [4] extended this concept with GSM notifications for threshold alerts, validating the viability of microcontroller-based approaches for small farms.

García et al. [5] presented a comprehensive survey of IoT-based smart irrigation, highlighting the role of low-cost sensing and wireless protocols (Zigbee, LoRa, Wi-Fi) in precision agriculture. Their work identified Bluetooth as suitable for short-range, low-power farm applications. Morchid et al. [6] demonstrated an IoT irrigation system with cloud monitoring achieving a 35% improvement in water use efficiency, though requiring persistent internet connectivity.

Ashraf [7] reviewed solar-pump irrigation systems and reported significant reductions in operational cost, noting that battery storage is critical for night-time operation. Okasha et al. [1] designed a low-cost capacitive soil moisture sensor operated by solar cells, validating sensor accuracy for greenhouse irrigation management.

In contrast to the above works, the present system uniquely combines (a) solar-only power with battery backup, (b) zone-wise solenoid control across four independent channels, and (c) a dedicated Bluetooth mobile application providing live status feedback without requiring internet access — making it well-suited for remote, off-grid agricultural environments.

III. SYSTEM ARCHITECTURE

The overall system architecture is illustrated in the block diagram (Fig. 1). It comprises three functional segments:

Power Generation & Supply, Sensing & Decision, and Irrigation Actuation, interconnected through the Arduino microcontroller.

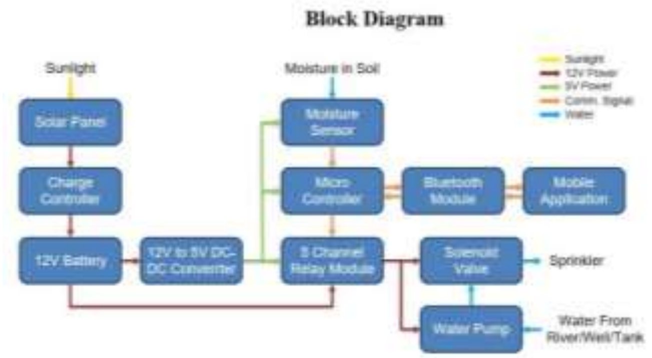

Fig. 1. Block diagram of the proposed Smart Irrigation System.

A. Power Generation & Supply

A 12 V, 50 W monocrystalline solar panel generates DC power from solar irradiance. A PWM charge controller regulates charging current, prevents overcharging, and maintains battery health. A 12 V, 7 Ah sealed lead-acid battery provides energy storage for night-time and cloudy-day operation. A 12 V-to-5 V DC-DC buck converter derives the 5 V rail required by the Arduino, relay module, soil moisture sensors, and HC-05 Bluetooth module. A 10 A Fuse provides short-circuit and overload protection.

B. Sensing & Decision

Four resistive soil moisture sensors (MS1–MS4) are embedded in the root zone of each irrigation section. Each sensor outputs a digital signal — LOW (logic 0) when soil moisture falls below the preset threshold (dry condition) and HIGH (logic 1) when adequate moisture is present — to analog input pins A0–A3 of the Arduino. The onboard potentiometer of each sensor allows threshold calibration for different soil types. The Arduino executes the irrigation decision logic on every program cycle.

C. Irrigation Actuation

A five-channel relay module (5 V coil, SPDT, 10 A contact rating) interfaces the 5 V Arduino outputs with the 12 V actuators. Four solenoid valves (5 V DC, normally closed, G1/2") independently control water flow to each zone. A 12 V, 660 L/h BLDC centrifugal pump draws water from the source and delivers it through 1/2" PVC pipes and sprinkler heads (15-foot spray diameter) to the field zones.

D. Bluetooth & Mobile Application

An HC-05 UART Bluetooth module (9600 baud, 10 m range) connected to Arduino pins D0/D1 (SoftwareSerial) enables bidirectional communication with a custom Android application. The app transmits text commands and the Arduino broadcasts a delimited status string every two seconds, enabling live monitoring and remote control from a smartphone.

IV. CIRCUIT DESIGN

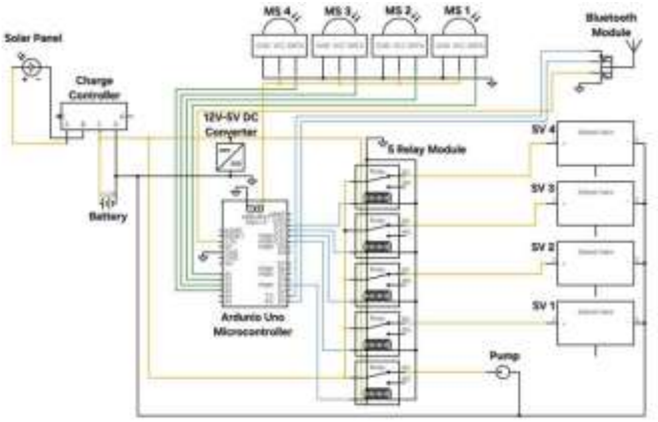

Fig. 2. Labeled circuit diagram of the Smart Irrigation System.

The Arduino UNO R3 (ATmega328P, 16 MHz) serves as the central processing unit. Digital output pins D10–D13 drive relay coil inputs for solenoid valves SV1–SV4 respectively. Pin D3 (PWM-capable) drives the fifth relay controlling the water pump (M). The relay module uses active-HIGH triggering: a HIGH from the Arduino energizes the relay coil, closing the NO (Normally Open) contact to complete the 12 V actuator circuit.

Moisture sensors are powered from the regulated 5 V rail. The HC-05 module is powered from 5 V with its TX/RX connected to Arduino D0/D1 via a voltage divider (HC-05 RX tolerates 3.3 V max; a 1 kΩ–2 kΩ divider protects it from the Arduino's 5 V TX). The DC-DC converter output supplies the 5 V rail to all logic components, while the 12 V battery line feeds the relay contact power, solenoid valves, and pump directly.

V. COMPONENTS & SPECIFICATIONS

Table I lists the key hardware components used in the prototype.

TABLE I. Key Components

Component	Qty	Specification
Solar Panel	1	12 V, 50 W Mono
Charge Controller	1	12 V, 10 A PWM
Battery	1	12 V, 7 Ah
DC-DC Converter	1	12 V → 5 V
Arduino UNO R3	1	ATmega328P, 5 V
HC-05 Bluetooth	1	UART, 9600 bps
5-Ch Relay Module	1	5 V coil, 10 A
Soil Moisture Sensor	4	5 V, Digital Out
Solenoid Valve	4	5 V DC, NC, G1/2"
BLDC Pump	1	12 V, 660 L/h, 3 m
Sprinkler Head	4	PVC, 1/2" dia.
Fuse	1	10 A

VI. MOBILE APPLICATION

The Android application was designed in MIT App Inventor 2 and communicates with the HC-05 module over Bluetooth SPP (Serial Port Profile). The app interface (Fig. 3) provides:

- Bluetooth connection picker — scans paired devices and connects to HC-05.
- Mode selector — Auto (green highlight) and Manual (yellow highlight) toggle buttons.
- Motor ON / Motor OFF buttons — enabled only in Manual mode.
- Live Status panel — five labelled indicators (SV1–SV4, M) with color-coded ON (green) / OFF (yellow) background.

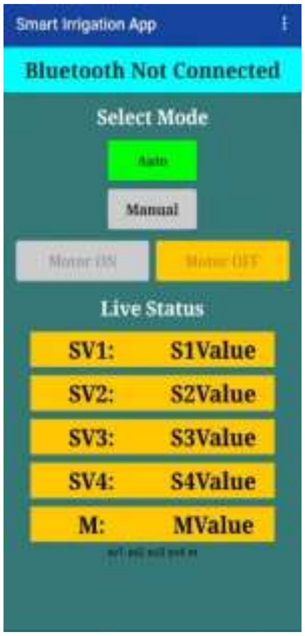

Fig. 3. Smart Irrigation Android application UI.

A. Communication Protocol

Commands from app to Arduino are plain text strings terminated with '\n': "Auto\n", "Manual\n", "M_ON\n", "M_OFF\n". The Arduino responds by broadcasting a slash-delimited status string every 2 s:

`SV1/SV2/SV3/SV4/M`

Each token is "1" (actuator energised / ON) or "0" (de-energised / OFF). The app parses the string by splitting on "/" and maps each token to its corresponding status label and background color.

B. App Inventor Block Logic

Fig. 4 shows the Screen1.Initialize block (sets delimiter byte to 10, sends "Auto\n" on startup, initialises global variables). Fig. 5 shows the Clock1.Timer block that polls Bluetooth, splits the received string, and updates all five status labels with color feedback.

Fig. 4. App Inventor Screen1.Initialize and Bluetooth connection blocks.

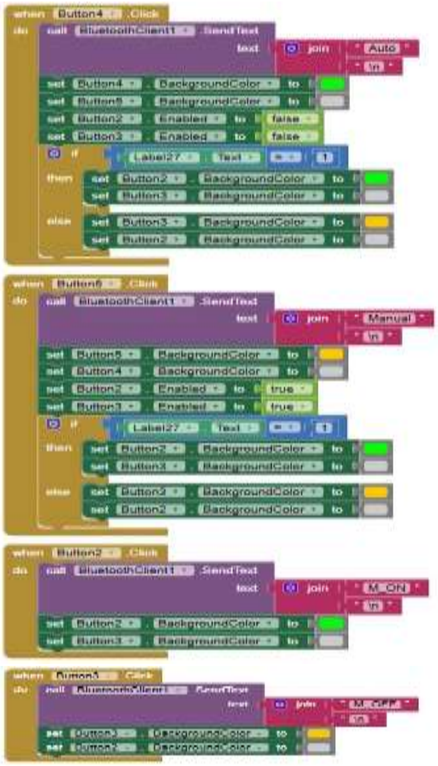

Fig. 5. Auto/Manual mode button and Motor ON/OFF command blocks.

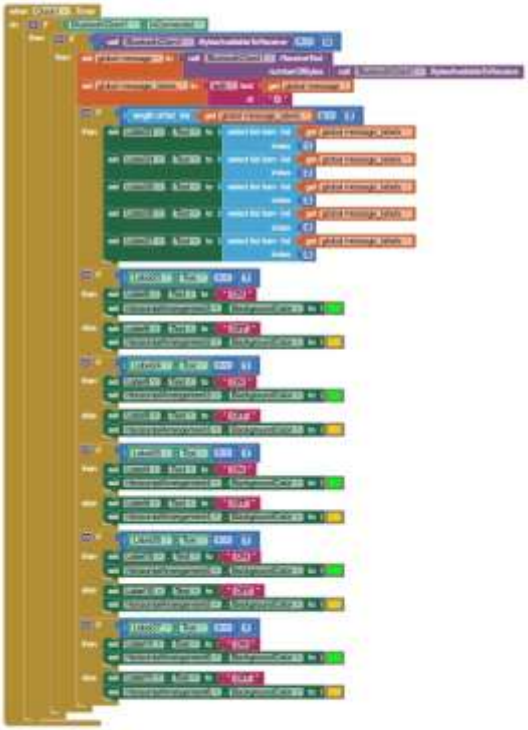

Figure 6: Motor OFF Command Block

VII. ARDUINO FIRMWARE

The firmware is written in C++ (Arduino IDE 2.x). The main loop executes continuously, reading all four sensors and driving actuators accordingly. Bluetooth commands are processed asynchronously via the SoftwareSerial buffer. A non-blocking 2 s timer (millis() comparison) handles status broadcasting without interrupting sensor polling.

```
#include <SoftwareSerial.h>

// HC-05 Bluetooth (RX, TX)
SoftwareSerial BT(0, 1);

// Sensor pins
int MS1 = A0;
int MS2 = A1;
int MS3 = A2;
int MS4 = A3;

// Sprinkler pins
int SV1 = 10;
int SV2 = 11;
int SV3 = 12;
int SV4 = 13;

// Motor pin
int M = 3;

//Status
String SV1_Status;
String SV2_Status;
String SV3_Status;
String SV4_Status;
String M_Status;

// Mode
bool manualMode = false;

// Timing
unsigned long previousMillis = 0;

void setup() {
  BT.begin(9600);    // Bluetooth start
  Serial.begin(9600);

  pinMode(MS1, INPUT);
  pinMode(MS2, INPUT);
  pinMode(MS3, INPUT);
  pinMode(MS4, INPUT);

  pinMode(SV1, OUTPUT);
  pinMode(SV2, OUTPUT);
  pinMode(SV3, OUTPUT);
  pinMode(SV4, OUTPUT);

  pinMode(M, OUTPUT);
}

void loop() {
  // Read sensors
  bool ms1 = digitalRead(MS1);
  bool ms2 = digitalRead(MS2);
  bool ms3 = digitalRead(MS3);
  bool ms4 = digitalRead(MS4);

  // Sprinkler control (inverted logic)
  if (ms1 == 0) {
    digitalWrite(SV1, HIGH);
    SV1_Status = "0";
  }
  else {
    digitalWrite(SV1, LOW);
    SV1_Status = "1";
  }

    if (ms2 == 0) {
    digitalWrite(SV2, HIGH);
    SV2_Status = "0";
  }
  else {
    digitalWrite(SV2, LOW);
    SV2_Status = "1";
  }

    if (ms3 == 0) {
    digitalWrite(SV3, HIGH);
    SV3_Status = "0";
  }
  else {
    digitalWrite(SV3, LOW);
    SV3_Status = "1";
  }

    if (ms4 == 0) {
    digitalWrite(SV4, HIGH);
    SV4_Status = "0";
  }
  else {
    digitalWrite(SV4, LOW);
    SV4_Status = "1";
  }

    // Bluetooth control
  if (BT.available()) {
    String cmd = BT.readStringUntil('\n');
    cmd.trim();

    if (cmd == "Manual") {
      manualMode = true;
    }
    else if (cmd == "Auto") {
      manualMode = false;
    }
    else if (cmd == "M_ON" && manualMode) {
      digitalWrite(M, HIGH);
    }
    else if (cmd == "M_OFF" && manualMode) {
      digitalWrite(M, LOW);
    }
  }

  // Auto motor control (only when not manual)
  if (!manualMode) {
    if (ms1 == LOW && ms2 == LOW && ms3 == LOW &&
ms4 == LOW) {
      digitalWrite(M, LOW);
```

```
    } else {
      digitalWrite(M, HIGH);
    }
  }

  //Update Status

  if (digitalRead(M) == HIGH) {
  M_Status = "1";
  } else {
  M_Status = "0";
  }

  // Send every 2 sec
  unsigned long currentMillis = millis();

  if (currentMillis - previousMillis >= 2000) {
    previousMillis = currentMillis;

    String data = SV1_Status + "/" + SV2_Status +
"/" +
                  SV3_Status + "/" + SV4_Status +
"/" +
                  M_Status;

    BT.println(data);
    Serial.println(data);
  }
}
```

Listing 1. Arduino UNO firmware.

VIII. SYSTEM CONSTRUCTION

The hardware prototype (Fig. 7–9) is assembled on a four-wheel wooden trolley (600 mm × 400 mm base). The solar panel is mounted on a tiltable frame on top; the control box (Arduino + relay module + DC-DC converter + HC-05) and battery are enclosed in a weatherproof enclosure on the lower shelf. The BLDC pump is fixed at the trolley rear with inlet connected to the water source via flexible hose.

Fig. 7. Portable solar trolley with panel, battery and enclosure.

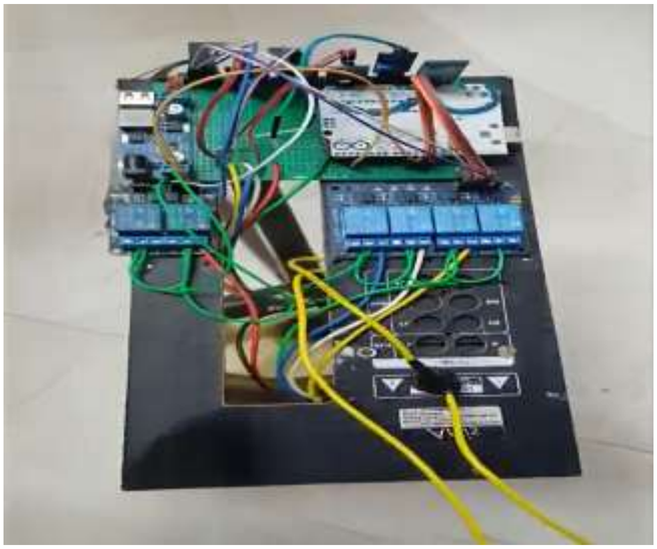

Fig. 8. Control box: Arduino UNO, relay module, HC-05 and DC-DC converter.

Fig. 9. Field pipe network with PVC pipes, solenoid valves and sprinkler risers.

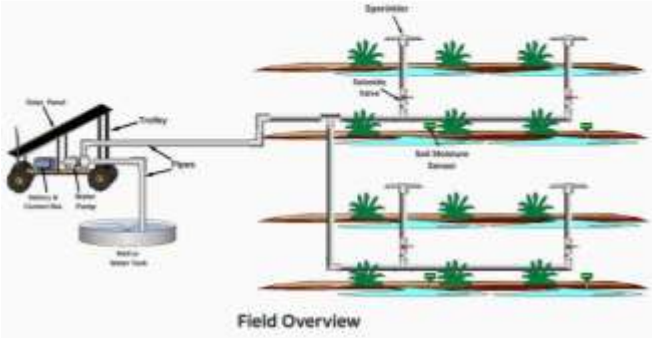

Fig. 10. Animated field layout overview showing system deployment.

The field pipe network consists of 1/2" PVC pipes arranged in a rectangular grid. Solenoid valves are installed at the branch junction of each zone. Soil moisture sensors are buried at 5–10 cm depth (root zone) within each zone. All inter-component wiring is protected with PVC conduit and secured with cable ties.

IX. SYSTEM OPERATION

A. *Automatic Mode*

On power-up, the Arduino initialises all peripherals and defaults to Automatic mode. In each program loop, it reads MS1–MS4. For any sensor reading LOW (dry soil), the corresponding relay is energised, opening the solenoid valve for that zone. If at least one zone is dry, the pump relay is also energised, drawing water from the source through the open valve(s) to the sprinklers. When all sensors return HIGH (adequate moisture), all valve relays and the pump relay are de-energised, halting irrigation. The LCD and Bluetooth status string reflect current actuator states.

B. *Manual Mode*

When the operator selects "Manual" on the mobile app, the Arduino sets manualMode = true. In this mode, solenoid valves continue to respond to sensor readings (zone protection), but the pump is decoupled from the sensor logic and responds only to "M_ON" and "M_OFF" Bluetooth commands. This allows the operator to manually run or stop the pump independent of soil moisture, useful for pre-irrigation or system flushing.

C. *Live Status Monitoring*

The Arduino broadcasts the status string "SV1/SV2/SV3/SV4/M" every 2 s over Bluetooth. The mobile app receives, parses, and displays each value with a green (ON) or yellow (OFF) background on the live dashboard, giving the operator complete real-time visibility of all actuators from a smartphone within 10 m.

X. RESULTS & DISCUSSION

The prototype was tested in a field setup with four irrigation zones, each covering approximately 2 m². Key observations are summarized in Table II.

TABLE II. Performance Summary

Parameter	Result
Sensor response time	< 1 second
Actuator response time	< 500 ms (relay delay)
Bluetooth range (line of sight)	~10 m
Status refresh interval	2 seconds
Solar charging (full sun, 4 h)	Battery fully charged
Battery backup (no sun)	~3 hours operation
Water delivery accuracy	Zone-selective, no cross-flow
App mode switching latency	< 1 second

The system reliably activated the correct solenoid valve and pump in response to sensor dry conditions and correctly stopped irrigation upon moisture restoration. The mobile app successfully switched between Auto and Manual modes and displayed accurate live status for all five actuators. No false triggers or missed events were observed during a 2-hour continuous test. Solar charging maintained full battery level after 4 hours of sunshine, confirming energy autonomy for daytime operation.

XI. ADVANTAGES

- Fully off-grid operation — no grid electricity dependency.
- Zone-wise irrigation prevents water wastage in already-moist areas.
- Real-time mobile monitoring without internet connectivity.
- Portable trolley design suits multi-section farms.
- Low operational cost after initial installation.
- Scalable — additional zones can be added with minimal hardware changes.

XII. LIMITATIONS & FUTURE WORK

Current limitations and proposed solutions are as follows:

- Bluetooth range (~10 m) limits remote operation. Solution: replace HC-05 with an ESP8266 Wi-Fi/MQTT module for internet-based control.
- No water-level sensing in source reservoir — pump may run dry. Solution: add ultrasonic or float sensor with pump-dry protection logic.
- System cannot skip irrigation during rain. Solution: integrate rain sensor or IoT weather API for forecast-based override.
- Trolley is manually moved. Solution: motorised wheels with GPS-guided path control.
- Fixed moisture threshold. Solution: crop-type and season-adaptive thresholds using ML regression on historical sensor data.
- Future integration with drip irrigation for even finer water efficiency.

XIII. CONCLUSION

This paper has presented a solar-powered Smart Irrigation System integrating soil moisture sensing, microcontroller-based automation, and Bluetooth mobile control. The Arduino UNO microcontroller manages four independently controlled solenoid valve zones and a water pump based on real-time digital moisture sensor inputs. A custom MIT App Inventor Android application communicates with the system via HC-05 Bluetooth, providing Auto/Manual mode switching and a live actuator status dashboard refreshed every 2 s — all without requiring internet connectivity.

Field testing confirmed reliable zone-selective irrigation, accurate sensor-to-actuator response, and effective smartphone monitoring within 10 m range. The system operates entirely on 50 W solar power with 3-hour battery backup, making it suitable for off-grid and remote agricultural deployments. The portable trolley design allows flexible multi-field use from a single installation.

Future work will focus on IoT cloud integration, AI-driven threshold adaptation, rain-sensor override, and autonomous trolley movement to further enhance system intelligence and usability. The proposed system demonstrates a practical, low-cost pathway toward precision agriculture for small and medium farms in developing regions.

ACKNOWLEDGMENT

The authors sincerely thank Mrs. A. S. Yadav (Project Guide) for her continuous guidance and support throughout this project. We are grateful to the Department of Electrical Engineering and the management of Jaywant College of Engineering & Polytechnic, K. M. Gad, for providing laboratory facilities and technical resources for this work.

REFERENCES

[1] A. M. Okasha, H. G. Ibrahim, A. H. Elmetwalli, K. M. Khedher, Z. M. Yaseen, and S. Elsayed, "Designing low-cost capacitive-based soil moisture sensor and smart monitoring unit operated by solar cells for greenhouse irrigation management," Sensors, vol. 21, no. 16, p. 5387, Aug. 2021. doi: 10.3390/s21165387.

[2] B. Chhabada, V. Narawade, and V. Mane, "Solar-powered smart irrigation system with cloud integration," Int. J. Intell. Syst. Appl. Eng., vol. 11, no. 10s, pp. 655–659, 2023.

[3] C. K. Sahu and P. Behera, "A low-cost smart irrigation control system," in Proc. IEEE Int. Conf. Electron. Circuit Syst. (ICECS), Cairo, Egypt, 2015, pp. 1–4.

[4] V. Divyak, R. Priya, and A. Kumar, "Arduino based smart irrigation system," Int. J. Adv. Res. Comput. Commun. Eng. (IJARCCE), vol. 7, no. 3, pp. 14–18, Mar. 2018.

[5] L. García, L. Parra, J. M. Jimenez, J. Lloret, and P. Lorenz, "IoT-based smart irrigation systems: An overview on the recent technology for smart watering and crop irrigation," Sensors, vol. 20, no. 4, p. 1042, Feb. 2020. doi: 10.3390/s20041042.

[6] A. Morchid, R. El Alami, A. Raezah, and A. Heidari, "IoT-based smart irrigation management system using machine learning and OpenWeatherMap," Appl. Sci., vol. 13, 2023.

[7] A. Ashraf, "Solar-powered irrigation systems: Developments and future trends," Renew. Sustain. Energy Rev., 2022.

[8] R. Kathiresan, M. Janani, K. M. Naveen, E. Narashimabalaji, and R. Rasiga, "Solar energy based smart irrigation system using IoT," J. IoT Social Mobile Anal. Cloud (JISMAC), vol. 5, no. 2, pp. 83–99, 2023.

[9] M. V. Gurao and U. B. Vaidya, "IoT based smart irrigation system," Int. J. Eng. Res. Technol. (IJERT), vol. 11, no. 11, 2022.

[10] P. Tan, J. Liu, H. Guo, and F. Zhu, "Soil moisture monitoring and watering management system using IoT and ARM," IEEE Access, vol. 10, 2022.