

KASDI MERBAH UNIVERSITY OF OUARGLA
Faculty of Hydrocarbons and Renewable Energies and Earth
and Universe Sciences



Department of Renewable Energies

Memory

ACADEMIC MASTER

Domain: Science and technology

Field: electrotechnical

Specialty: Renewable energies in electrotechnical

Prepared by: Ilyes ZERGANI

Abdennasser CHEBBOUB

Title:

Design of a hybrid automatic and semi – automatic control system for a model farm using Arduino, sensor, and SMS technology

Publicly supported

on: ... / 06 /2025

Before the jury:

.....	President	UKM-Ouargla
Dr. Abdelkader KHENTOUT	MCA	Supervisor	UKM-Ouargla
Dr. Zhou ABADA	MCB	Assistant Supervisor	UKM-Ouargla
.....	Examiner	UKM-Ouargla

Academic Year: 2024/2025

Acknowledgements 1

Zergani Ilyes

Our foremost thanks, praise to "ALLAH" for giving us health, courage will and faith to accomplish this work.

*Our thanks go to Our project director **Dr.KHANTOUT Abdelkader**, who guided us with his precious advice and suggestions, and the confidence he showed us and also for offering us a pleasant work environment and quality supervision.*

*We warmly thank **Mr. BEN DELLALI Riad** and **Mr. KAFI Ismail** and **Mr.Bensddik Noureddin** for their follow-up and support during the realization of this work, May they find here the expression of our deep gratitude, for the encouragement, advice and trust they have always shown.*

*We also express our sincere thanks to all our teachers from the Department of Renewable Energies at the **University of KASDI MERBAH** in OUARGLA for the effort they have made to ensure our education, for their skills, and especially their modesty.*

We would also like to thank our families, our friends, as well as all those who have contributed directly or indirectly to the realization of this work.

Acknowledgements 2

Chebboube Abd Ennasser

Our foremost thanks, praise to "ALLAH" for giving us health, courage will and faith to accomplish this work.

*Our thanks go to Our project director **Dr.KHANTOUT Abdelkader**, who guided us with his precious advice and suggestions, and the confidence he showed us and also for offering us a pleasant work environment and quality supervision.*

*We warmly thank **Mr. BEN DELLALI Riad** and **Mr. Chebboub Djamal** for their follow-up and support during the realization of this work, May they find here the expression of our deep gratitude, for the encouragement, advice and trust they have always shown.*

*We also express our sincere thanks to all our teachers from the Department of Renewable Energies at the **University of KASDI MERBAH** in OUARGLA for the effort they have made to ensure our education, for their skills, and especially their modesty.*

We would also like to thank our families, our friends, as well as all those who have contributed directly or indirectly to the realization of this work.

List of content

Acknowledgements 1	I
Acknowledgements 2	II
List of content.....	III
List of figures	VII
List of tables	8
General introduction.....	1

Chapter I

General Concepts on Control and Communication Systems in Smart Agriculture

I.1 Introduction.....	3
I.2 Automatic Control Systems	3
I.2.1 Definition of Control Systems	3
I.2.2 Types of Control Systems	3
I.2.3 Components of Control Systems	5
I.2.4 Applications of Control Systems in Agriculture.....	5
I.3 Advantages of Automatic Control Systems.....	6
I.4 Challenges and Considérations.....	6
I.5 Automation in Agriculture.....	6
I.5.1 Definition of Agricultural Automation	7
I.5.2 Goals and Benefits of Automation.....	7
I.5.3 Examples of Smart Agriculture Systems	8
I.6Microcontrollers.....	10
I.6.1 Microcontroller	10
I.6.2 Arduino Platform: Features and Benefits	11
I.6.3 Use of Arduino in Agricultural Applications.....	13
I.7 Remote Communication Technologies.....	14

I.7.1 Basic Concepts of Wireless Communication.....	15
I.7.2 GSM and SMS Technology	16
I.8 Renewable Energy in Smart Systems	18
I.8.1 The Importance of Solar Energy in Agriculture	18
I.8.2 Integration of Renewable Energy with Control Systems.....	19
I.9 Conclusion	20

Chapter II

Technical components of the smart agricultural control system

II.1 Introduction	24
II.2 The Arduino Microcontroller	24
II.2.1 Arduino	24
II.2.2 Main Components of Arduino	25
II.2.3 Programming Arduino with Arduino IDE.....	26
II.2.4 Arduino in Smart Agriculture.....	30
II.3 Sensors in Smart Agriculture.....	31
II.3.1 Types of Agricultural Sensors	31
II.3.2 Environmental Sensors	31
II.3.3 Soil Moisture Sensors	32
II.3.4 Choosing the Right Sensor	32
II.4 Wireless Communication Technologies in Agriculture	33
II.4.1 Overview	33
II.4.2 GSM and SMS Technology.....	34
II.4.3 Sending and Receiving Messages via GSM	34
II.4.4 Comparison of Wireless Communication Technologies	35
II.5 Integrating renewable energies into smart agriculture systems.....	36
II.5.1 The Importance of Renewable Energy	37
II.5.2 Solar Energy as a Primary Source	38
II.5.3 Integrating Solar Power with Control Systems	39

II.6 Smart Agricultural System Design	39
II.6.1 System Requirements Analysis	41
II.6.2 General Architecture of the System.....	42
II.6.3 System Block Diagram	44
II.6.4 Control Strategy.....	44
II.7 Conclusion	45

Chapter III

Simulation and practical implementation of the smart agricultural system

III.1 introduction	46
III.2 simulation environment and tools used	46
III.2.1 Software platforms (proteus,tinkercad,Arduino ide).....	47
III.2.2 Virtual component (Arduino, sensors, gsm module)	48
III.2.3 Setup and configuration of simulation environment	49
III.3 First simulation experiment: automated moisture control system using Arduino	50
III.3.1 Experiment objectives	50
III.3.2 System setup in simulation.....	51
III.3.3 Control algorithm overview	51
III.3.4 Simulation results and performance analysis	52
III.4 Second simulation experiment: semi-automated control via sms using gsm module.....	53
III.4.1 Experiment objectives	53
III.4.2 sms command setup and interaction.....	54
III.4.3 Response analysis in simulation environment	54
III.5 Laboratory model farm: design and real-world reference.....	55
III.5.1 location and setup within the laboratory	55
III.5.2 Reference real farm: location and energy source	56
III.5.3 Physical and environmental characteristics of the model farm.....	57
III.6 components of the system implemented in the model farm.....	58
III.6.1 Hardware components (Arduino, sensors, GSM, solar panels)	58

III.6.2 Wiring and integration diagram	59
III.6.3 system configuration for laboratory environment	60
III.7.2 Measurement Accuracy and System Response	69
III.7.3 System Behavior under Variable Environmental Conditions	69
III.8 Second Practical Experiment: Control via SMS Commands Using GSM.....	70
III.8.1 Sending Commands via SMS to GSM Module	71
III.8.2 System Response within the Model Farm.....	72
III.9 Challenges and Lessons Learned	73
III.10 Conclusion.....	75
General conclusion.....	75
Bibliographic references	خطأ! الإشارة المرجعية غير معرّفة.

List of figures

Figure I.1 open-loop control system	4
Figure I.2 arble TM smart farming system.....	8
Figure II.1 arduino uno.....	25
Figure II.2 main components of arduino	26
Figure II.3 Arduino IDE Interface and Main Sections.....	27
Figure II.4 Toolbar section.....	27
Figure II.5 menu section.....	28
Figure II.6 sketch menu section	29
Figure II.7 tools menu section.....	29
Figure II.8 environmental sensors	31
Figure II.9 YL-69 Soil Moisture Sensor Used in the Smart Irrigation System.....	32
Figure II.10 comprehensive communication.....	33
Figure II.11 Integration of Renewable Energy into the Smart Agriculture System.....	37
Figure II.12 solar energy	39
Figure II.13 The following figure shows the system block diagram[14].....	44
Figure III.1 Icons of the Software Tools Used.....	48
Figure III.2 Full-Automatic System Simulation Using Proteus.....	50
Figure III.3 Semi-Automatic System Simulation Using Proteus	53
Figure III.4 Practical Implementation of the Prototype System.....	55
Figure III.5 Image Showing Wheat Production in the Farm.....	57
Figure III.6 Image of the COSIDER Farm Surroundings	57
Figure III.7 Front View of the COSIDER Farm	57
Figure III.8 COSIDER Agricultural Investment Site.....	57
Figure III.10 Diagram Illustrating the Integration of Arduino, SIM800L, and Sensors	60
Figure III.11 Real Image of the System Wiring and Connections	61
Figure III.13 Image of the Prototype Farm With Network Controller.....	71

List of tables

Table I.1 Summary Table: Smart Agriculture Systems Overview.....	9
Table I.2 Common Microcontroller Families	11
Table I.3 Features of the Arduino Platform	12
Table I.4 Common Agricultural Applications Using Arduino.....	13
Table I.5 Comparison of Wireless Communication Technologies Used in Remote Smart Agriculture Systems.....	14
Table I.6 Core Concepts in Wireless Communication	15
Table I.7 Types of Modulation.....	15
Table I.8 Transmission Modes	16
Table I.9 Summary Table: Wireless Communication Basics	16
Table I.10 Common GSM Modules	17
Table I.11 Applications of GSM and SMS Technology	17
Table II.1 Main Components of Arduino.....	25
Table II.2 Sensor selection	32
Table II.3 Comprehensive Communication for Precision.....	33
Table II.4 Common AT Commands.....	35
Table II.5 Comparison of Wireless Communication Technologies	36
Table III.1 Table Comparing Automatic and Semi-Automatic Control Modes	75

Nomenclature Symbol, Designation, Unit

GSM: Global System for Mobile Communications

SIM: Subscriber Identity Module

SMS: Short Message Service

UART: Universal Asynchronous Receiver-Transmitter

AT Commands: Control commands for GSM modules

Wi-Fi: Wireless Fidelity

LoRa: Long Range wireless technology

BLE: Bluetooth Low Energy

IoT: Internet of Things

IDE: Integrated Development Environment

RTC: Real-Time Clock

LED: Light Emitting Diode

OLED: Organic Light Emitting Diode

LCD: Liquid Crystal Display

kWh: Kilowatt-hour

kW: Kilowatt

MWh: Megawatt-hour

HP: Horsepower (1 HP \approx 0.746 kW)

V: Volt

A: Ampere

I_k: Short-circuit current

I_n: Nominal current

kVA: Kilovolt-ampere – Apparent power unit

MVA: Megavolt-ampere

General introduction

General introduction

The global population is expected to reach around 10 billion by 2050, which will significantly increase the demand for food and agricultural products—by nearly 50% compared to 2013 (FAO, 2017). While agriculture currently occupies about 37.7% of the Earth’s land surface, the sector faces growing challenges related to climate change, water scarcity, and energy limitations—especially in developing countries. In Algeria, agriculture remains a vital pillar of the national economy, contributing approximately 12% of the GDP and serving as the primary source of income for many rural communities.

In arid and semi-arid regions, where water is scarce and temperatures are high, irrigation becomes one of the most critical challenges for farmers, directly affecting crop quality and yield. Algeria, for instance, has over 18 million date palm trees cultivated across more than 167,000 hectares, mostly in areas that require highly efficient irrigation systems sensitive to temperature and soil moisture levels.

As a response to these challenges, integrating smart agriculture technologies has become essential. By using environmental sensors to monitor humidity and temperature, and incorporating microcontroller platforms like Arduino—alongside wireless communication modules such as GSM for SMS alerts—farmers can gain precise control over irrigation processes.

This project aims to design a hybrid automatic and semi-automatic irrigation control system for a model farm, focusing on real-time monitoring of soil moisture and temperature. The system helps reduce water waste, prevents over-irrigation, and improves crop productivity. Furthermore, it enables farmers to remotely control and monitor the system, reducing daily labor and advancing the concept of a smart, sustainable farm.

Chapter I

General Concepts on Control and Communication Systems in Smart Agriculture

I.1 Introduction

The world's agricultural sector is experiencing a digital revolution with the aim of enhancing sustainable food production. Intelligent agriculture, or precision or digital farming, uses control and communication systems to maximize farm activities through data-based decision-making.

Control systems automate critical processes such as irrigation, fertilization, and climate control with real-time feedback to create ideal conditions. Communication systems enable safe data transfer between field devices, cloud platforms, and user interfaces to enable remote monitoring and automation.

Combined, these systems form the foundation of smart farming, with enhanced productivity, resource utilization, and environmental sustainability attained through the use of advanced technology. [1]

I.2 Automatic Control Systems

Automatic control systems play a pivotal role in smart agriculture by enabling the automation of complex farming processes, reducing manual labor, and improving precision in agricultural operations. These systems use advanced technologies to monitor field conditions and regulate machinery or environmental parameters without constant human intervention. [2]

I.2.1 Definition of Control Systems

An automatic control system is a set of mechanical or electronic devices that manage, command, or regulate the behavior of other devices using control loops. In smart agriculture, these systems are designed to maintain desired output levels (e.g., soilmoisture, green house temperature) through real-time adjustments based on sensor feedback.

I.2.2 Types of Control Systems

Control systems are classified into several types based on feedback and operation. The most common are:

- **Open-Loop Control System:** Operates without feedback; control action is independent of the output.

Example: A timer-based irrigation system that waters crops at fixed intervals regardless of actual soil moisture. The figure below illustrates the system architecture

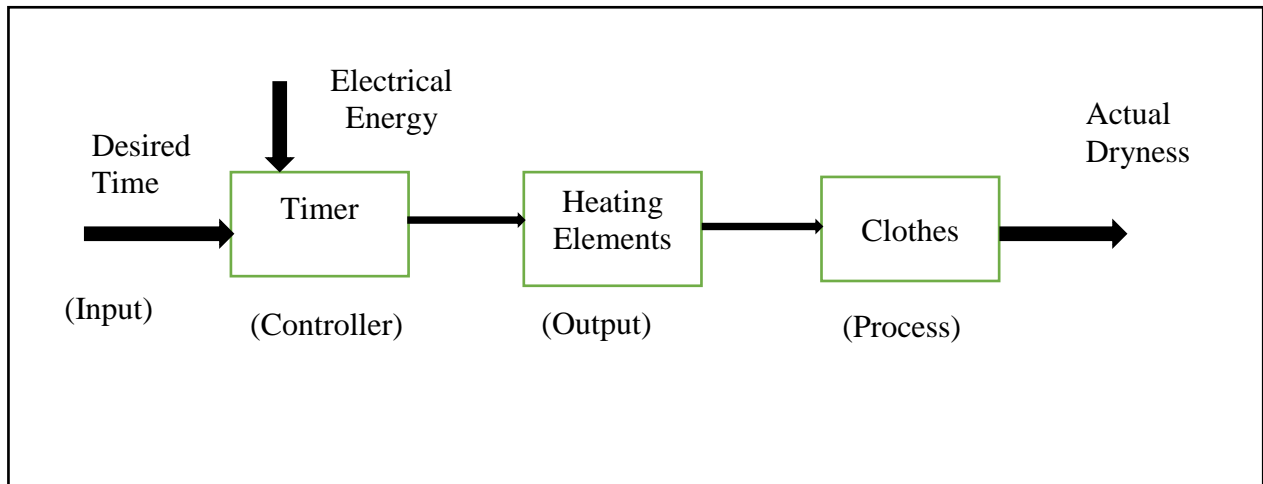


Figure I.1 open-loop control system

We can define the main characteristics of an “Open-loop system” as being:

- There is no comparison between actual and desired values.
- An open-loop system has no self-regulation or control action over the output value.
- Each input setting determines a fixed operating position for the controller.
- Changes or disturbances in external conditions does not result in a direct output change (unless the controller setting is altered manually)
- Closed-Loop (Feedback) Control System: Uses feedback from sensors to adjust control actions and achieve desired outcomes

Example: An irrigation system that adjusts water flow based on real-time soil moisture readings. The figure below illustrates the system architecture

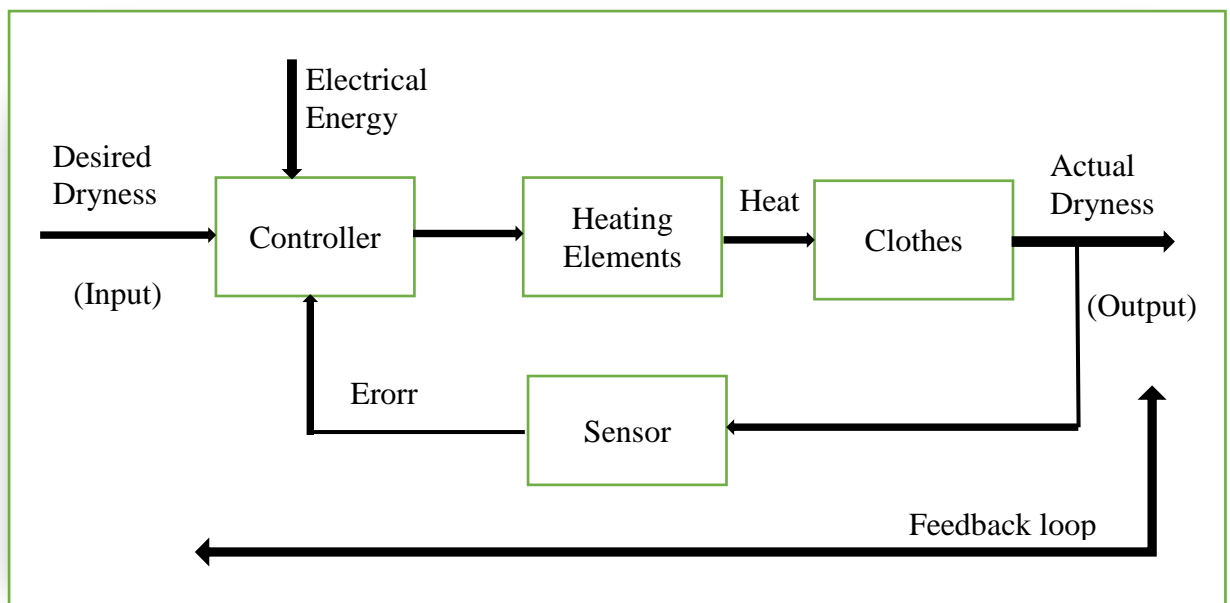


Figure I.2 closed-loop control system

We can define the main characteristics of **Closed-loop Control** as being:

- To reduce errors by automatically adjusting the systems input.
- To improve stability of an unstable system.
- To increase or reduce the systems sensitivity.
- To enhance robustness against external disturbances to the process.
- To produce a reliable and repeatable performance.

I.2.3 Components of Control Systems

1. Sensors:

Devices that detect changes in environmental or operational conditions (e.g., soil moisture sensors, temperature sensors, light sensors).

2. Controllers:

Process data from sensors and make decisions. Can be simple microcontrollers (e.g., Arduino, Raspberry Pi) or advanced PLCs (Programmable Logic Controllers) and AI-based systems.

3. Actuators:

Physical devices that perform actions based on controller commands (e.g., motorized valves, fans, pumps, robotic arms).

4. User Interface / Monitoring System:

Allows farmers to observe system status, receive alerts, and manually override settings if needed (e.g., mobile apps, web dashboards).

5. Communication Modules:

Enable remote monitoring and control via wireless protocols such as LoRaWAN, Zigbee, Wi-Fi, or cellular networks.

I.2.4 Applications of Control Systems in Agriculture

- **Automated Irrigation:** Adjusts water delivery based on soil moisture, weather forecasts, and crop needs.
Benefits: Reduces water waste, improves yield, and conserves resources.
- **Climate Control in Greenhouses:** Regulates temperature, humidity, CO₂ levels, and lighting.
Benefits: Enhances plant growth and protects crops from extreme weather.
- **Autonomous Farming Equipment:** Self-driving tractors, harvesters, and drones operate using GPS and onboard sensors.

Benefits: Increases efficiency reduces labor costs, enables 24/7 operation.

- **Livestock Monitoring & Management:** Monitors animal health, feeding habits, and location using wearable sensors and automated feeders.

Benefits: Improves animal welfare, early disease detection, and productivity.

- **Pest and Disease Control:** Activates pesticide sprayers only when pests are detected using image recognition and sensor data.

Benefits: Minimizes chemical use, reduces environmental impact.

I.3 Advantages of Automatic Control Systems

- **Increased Efficiency:** Automates repetitive tasks, saving time and labor.
- **Precision and Consistency:** Ensures uniform application of inputs like water, fertilizers, and pesticides.
- **Real-Time Adaptation:** Responds quickly to changing environmental conditions.
- **Data-Driven Decisions:** Integrates with farm management software for better planning and optimization.
- **Scalability:** Suitable for both small farms and large agribusinesses.

I.4 Challenges and Considerations

- **High Initial Cost:** Investment in sensors, controllers, and communication infrastructure can be expensive.
- **Technical Expertise:** Requires knowledge of electronics, programming, and system integration.
- **Maintenance and Reliability:** Systems must be robust to function in harsh outdoor environments.
- **Cybersecurity Risks:** Connected systems may be vulnerable to hacking or data breaches.
- **Power Supply Issues:** Especially in remote areas where grid power is not available.

I.5 Automation in Agriculture

Automation in agriculture refers to the use of advanced technologies, machinery, and control systems to perform farming tasks with minimal human intervention. It is a core component of smart agriculture and plays a critical role in improving productivity, reducing labor dependency, and optimizing resource usage. [3]

I.5.1 Definition of Agricultural Automation

Agricultural automation involves the application of automated systems such as robotics, sensors, and artificial intelligence (AI), and control systems—to manage various agricultural processes including planting, irrigation, harvesting, monitoring, and pest control.

I.5.2 Goals and Benefits of Automation

Automation in agriculture is driven by the need to increase efficiency, reduce labor dependency, and optimize resource use while improving crop yield and sustainability. Below is a detailed breakdown of the goals and benefits associated with automation.

- **Increase Operational Efficiency:** Perform tasks faster and more accurately than manual labor.
- **Reduce Labor Dependency:** Address labor shortages by using machines for repetitive or physically demanding tasks.
- **Improve Resource Utilization:** Use water, fertilizers, pesticides, and energy more efficiently through precision techniques.
- **Enhance Crop Yield and Quality:** Provide optimal growing conditions and timely interventions to maximize productivity and product quality.
- **Enable Real-Time Monitoring and Control:** Collect and analyze data continuously to support informed decision-making.
- **Support Sustainable Farming Practices:** Minimize environmental impact by reducing waste and overuse of inputs.
- **Facilitate Scalability:** Allow farmers to manage larger areas with the same or fewer resources.
- **Improve Safety and Working Conditions:** Reduce exposure of workers to hazardous chemicals, heavy machinery, and extreme weather conditions.

Benefits of Automation in Agriculture

- ✓ Higher Productivity
- ✓ Cost Savings
- ✓ Precision and Consistency
- ✓ Remote Monitoring & Management
- ✓ Improved Decision-Making
- ✓ Environmental Sustainability
- ✓ Early Detection of Issues

- ✓ Better Risk Management
- ✓ Adaptability to Climate Change
- ✓ Increased Profitability

I.5.3 Examples of Smart Agriculture Systems

Smart agriculture integrates advanced technologies such as the Internet of Things (IoT), artificial intelligence (AI), robotics, and data analytics to optimize farming operations. Below are several real-world examples of smart agriculture systems currently in use across the globe.

1. Smart Irrigation Systems

Purpose: Optimize water usage based on real-time field conditions.

Example: Arable™ Smart Farming System



Figure I.3 arble™ smart farming system

- Uses a single sensor device that measures soil moisture, temperature, rainfall, and crop health.
- Provides actionable insights via a cloud-based platform.
- Helps farmers make informed irrigation decisions and reduce water waste.

Example: Netafim's Precision Irrigation

- Combines drip irrigation with IoT sensors and AI algorithms.
- Adjusts water delivery based on plant needs, weather forecasts, and soil conditions.

Benefits: Water conservation, increased yield, reduced energy costs.

2. Autonomous Tractors & Farm Equipment

Purpose: Perform plowing, planting, spraying, and harvesting without human drivers.

Example: John Deere Autonomous Tractor

- Equipped with GPS, computer vision, and AI for navigation and obstacle detection.
- Can operate 24/7 with minimal supervision.

Example: Fendt Xavier System

- A fleet of small autonomous robots for seeding and monitoring crops.
- Communicates with drones for field mapping and monitoring.

Benefits: Reduced labor dependency, efficient use of time and fuel, precision farming.

3. AI-Powered Farm Management Platforms

Purpose: Analyze farm data and provide insights for better decision-making.

Example: The Climate Corporation’s Field View™

- Aggregates data from equipment, satellites, and sensors to guide planting, spraying, and harvesting.

Example: Farm Beats by Microsoft

- Combines IoT, AI, and cloud computing to help farmers optimize yields using data from drones, sensors, and tractors.

Benefits: Data-driven decisions, predictive analytics, integration with multiple farm tools.

4. Mobile-Based Farm Advisory Services

Purpose: Provide farmers with real-time advice via mobile apps.

Example: Agripa (India)

- Offers weather updates, market prices, pest alerts, and expert consultations through a smartphone app.

Example: Hello Tractor (Africa)

- A digital platform that connects tractor owners with smallholder farmers on demand.

Benefits: Accessible to small-scale farmers, improves knowledge sharing, enhances market access.

Table I.1 Summary Table: Smart Agriculture Systems Overview

SYSTEM TYPE	TECHNOLOGY USED	KEY BENEFITS
Smart Irrigation	IoT, Sensors, Cloud	Water conservation, cost savings
Autonomous Tractors	GPS, AI, Robotics	Labor reduction, efficiency
Agricultural Drones	UAVs, AI, Imaging	Crop monitoring, targeted spraying
Livestock Monitoring	Wearables, AI	Health tracking, productivity

Greenhouse Automation	Sensors, HVAC, AI	Controlled environment, year-round production
Farm Management Platforms	AI, Big Data, Cloud	Decision support, analytics
Mobile Farming Apps	Mobile tech, SMS	Accessibility, advisory services

I.6 Microcontrollers

Microcontrollers are small, programmable computing devices that integrate a processor core, memory, and input/output (I/O) peripherals onto a single chip. They're designed to perform specific tasks in embedded systems from simple control operations to complex automation[4]

I.6.1 Microcontroller

A microcontroller (MCU) is essentially a tiny computer on a single integrated circuit. It's meant to interact with the real-world using sensors, switches, motors, displays, etc., and make decisions based on programmed instructions. [4]

- Central Processing Unit (CPU) – Executes instructions.
- Memory
- Flash Memory: Stores program code (non-volatile).
- RAM: Used for temporary data storage while running (volatile).
- ROM/EEPROM: Sometimes used for permanent or semi-permanent data storage.
- Input/output Ports (GPIOs) – General-purpose input/output pins to interface with external devices.
- Timers & Counters – For timekeeping, PWM generation, event counting.
- Communication Interfaces
- UART, SPI, I²C – Serial communication protocols.
- CAN, USB (in more advanced MCUs).
- Analog-to-Digital Converters (ADC) – Converts analog signals (e.g., from sensors) into digital values.
- Interrupt Controllers – Allow MCU to respond to external/internal events immediately.

Table I.2 Common Microcontroller Families

FAMILY	MANUFACTURER	FEATURES
AVR	Microchip (Atmel)	Easy to use: Arduino uses this (e.g. ATmega328P).
PIC	Microchip	Popular in industrial applications: variety of models.
ARM Cortex-M	STMicroelectronics (STM32), NXP, TI	High performance; widely used in modern embedded systems.
ESP Series	Espressif	Includes Wi-Fi (ESP8266) and Wi-Fi + Bluetooth (ESP32).
MSP430	Texas Instruments	Ultra-low power consumption.

Programming Microcontrollers

- Languages: C/C++ (most common), Assembly (for low-level), sometimes Python (e.g., Micro Python on ESP32/Py board).

Tools:

- Compilers: GCC, Kiel, IAR
- IDEs: Arduino IDE, STM32CubeIDE, MPLAB X, Platform IO
- Debuggers: JTAG, SWD (Serial Wire Debug)

Applications of Microcontrollers

- Home appliances (microwave ovens, washing machines)
- Automotive systems (engine control, ABS, infotainment)
- Medical devices (monitors, infusion pumps)
- IoT devices (smart sensors, wearables)
- Industrial automation (PLCs, motor control)

I.6.2 Arduino Platform: Features and Benefits

The Arduino platform is one of the most popular and accessible tools for working with microcontrollers. It's widely used in hobbyist electronics, education, prototyping, and even professional applications due to its simplicity and versatility. [5]

Table I.3 Features of the Arduino Platform

FEATURE	DESCRIPTION
Open Source	Hardware designs and software are freely available.
User-Friendly IDE	Simple interface for coding, compiling, and uploading programs.
C/C++ Based Language	Uses a simplified version of C++ for programming.
Wide Range of Boards	From basic (Uno, Nano) to advanced (Due, MKR, ESP32-based).
Large Community	Extensive tutorials, forums, and libraries.
Plug-and-Play Compatibility	Shields (add-on boards) easily extend functionality.
Cross-Platform Support	Works on Windows, macOS, Linux.
Integrated Bootloader	No external programmer needed code can be uploaded directly via USB.

Benefits of Using Arduino

1. Ease of Use

- Simplified programming syntax.
- Built-in functions (digital Write(), analog Read(), etc.) abstract away low-level register manipulation.

2. Fast Prototyping

- Plug-and-play shields and modules allow quick testing of ideas.
- Breadboard-friendly design.

3. Rich Ecosystem

- Thousands of libraries available via the Library Manager.
- Pre-built code examples help beginners get started quickly.

4. Affordability

- Most boards are inexpensive (under \$30).
- Clones and compatible boards are widely available at low cost.

5. Educational Value

- Widely used in schools, universities, and maker communities.
- Great for teaching electronics, embedded systems, and programming.

6. Scalability

- Can start simple and move to more advanced platforms or custom PCBs later.

7. Community Support

- Huge online presence: tutorials, forums (e.g., Arduino Forum), GitHub repos, YouTube channels.

I.6.3 Use of Arduino in Agricultural Applications

Arduino is increasingly being used in agricultural applications due to its affordability, ease of use, and flexibility. It enables farmers, researchers, and hobbyists to build smart farming systems that can monitor environmental conditions, automate irrigation, control pests, and more. [6]

Table I.4 Common Agricultural Applications Using Arduino

Benefit	Description
Low Cost	Affordable hardware ideal for small-scale farms or research.
Customizable	Can be tailored to specific agricultural needs.
Sensors Integration	Easy to connect various sensors (soil moisture, temperature, humidity, etc.).
Automation	Enables automation of tasks like irrigation, lighting, and fertilization.
Open Source	Large community support and available libraries for quick development.
Scalable	Systems can be expanded as needed — from a single greenhouse to large fields.

1. Soil Moisture Monitoring System

Purpose: Monitor soil moisture levels to optimize irrigation.

Components:

- Soil moisture sensor (e.g., FC-28 or capacitive sensor)
- Water pump or valve
- Relay module
- LCD or OLED display (optional)

Functionality: Triggers water pump when soil is dry.

```
cpp
1  int sensorPin = A0;
2  int pumpPin = 7;
3
4  void setup() {
5      pinMode(sensorPin, INPUT);
6      pinMode(pumpPin, OUTPUT);
7  }
8
9  void loop() {
10     int moisture = analogRead(sensorPin);
11
12     if (moisture > 500) { // Threshold value depends on sensor calibration
13         digitalWrite(pumpPin, HIGH); // Turn on pump
14     } else {
15         digitalWrite(pumpPin, LOW); // Turn off pump
16     }
17
18     delay(1000);
19 }
```

- Time-based watering schedules
- Rain detection using raindrop sensor
- Remote control via SMS or web app

3. Weather Station

Purpose: Monitor weather conditions affecting crops.

Sensors Used:

- DHT11/DHT22 – Temperature & Humidity
- BMP180/280 – Atmospheric Pressure & Altitude
- Rain sensor
- Anemometer (wind speed)
- Light-dependent resistor (LDR) – Light intensity

Output Options:

- SD card logging
- Display on LCD or OLED
- Transmit data via Wi-Fi (ESP32/ESP8266) to cloud platforms (like ThingSpeak)

4. Fertilizer Dispenser / Nutrient Management

Use Case: Precise delivery of nutrients based on soil readings.

Components:

- EC (Electrical Conductivity) sensor
- Peristaltic pumps
- pH sensor
- Useful in hydroponics and vertical farming.

I.7 Remote Communication Technologies

Table I.5 Comparison of Wireless Communication Technologies Used in Remote Smart Agriculture Systems

BENEFIT	DESCRIPTION
Low Cost	Affordable hardware ideal for small-scale farms or research.
Customizable	Can be tailored to specific agricultural needs.
Sensors Integration	Easy to connect various sensors (soil moisture, temperature, humidity, etc.).
Automation	Enables automation of tasks like irrigation, lighting, and fertilization.
Open Source	Large community support and available libraries for quick development.
Scalable	Systems can be expanded as needed-from a

	single greenhouse to large fields.
--	------------------------------------

I.7.1 Basic Concepts of Wireless Communication

Wireless communication is the transmission of data between devices without the use of physical cables or wires, using electromagnetic waves such as radio waves, infrared, or microwaves.

It forms the foundation for modern technologies like Wi-Fi, Bluetooth, mobile networks (4G/5G), satellite systems, and IoT (Internet of Things) applications. [7]

- Radio Frequency (RF) signals
- Microwave
- Infrared (IR)
- Satellite links

Advantages:

- Mobility and flexibility
- Easy deployment in remote areas
- Scalability
- Reduced cost of installation

Disadvantages:

- Susceptible to interference
- Security risks
- Limited range depending on technology
- Signal degradation (attenuation)

1. Electromagnetic Spectrum

All wireless communication happens over the electromagnetic spectrum , which includes:

Table I.6 Core Concepts in Wireless Communication

BAND	FREQUENCY RANGE	COMMON USE
Radio Waves	3 kHz – 300 MHz	AM/FM, TV, mobile phones
Microwaves	300 MHz – 300 GHz	Satellite, Wi-Fi, Bluetooth
Infrared	~300 GHz – 400 THz	Remote controls, short-range sensors
Visible Light	~400 THz – 800 THz	Li-Fi (experimental)

2. Modulation

Modulation is the process of encoding information onto a carrier wave for transmission.

Table I.7 Types of Modulation

TYPE	DESCRIPTION	USE CASES
Amplitude Modulation (AM)	Varies amplitude of carrier wave	Old radio broadcasting
Frequency Modulation (FM)	Varies frequency of carrier wave	FM radio, Bluetooth

Phase Modulation (PM)	Varies phase of carrier wave	Digital communications
Quadrature Amplitude Modulation (QAM)	Combines amplitude & phase modulation	Wi-Fi, LTE, cable modems

3. Antennas

An antenna is a device that converts electrical signals into electromagnetic waves (and vice versa).

4. Transmission Modes

How data is sent between devices:

Table I.8 Transmission Modes

MODE	DESCRIPTION	EXAMPLE
Simplex	One-way communication	Radio broadcast
Half-duplex	Two-way, but not at same time	Walkie-talkies
Full-duplex	Simultaneous two-way communication	Mobile calls, Wi-Fi

Table I.9 Summary Table: Wireless Communication Basics

CONCEPT	DESCRIPTION
Medium	Electromagnetic waves (radio, microwave, infrared)
Modulation	Encoding data onto carrier wave (AM, FM, PM, QAM)
Antenna	Converts signals to waves and vice versa
Transmission Mode	Simplex, half-duplex, full-duplex
Multiplexing	TDM, FDM, CDM
Noise/Interference	Signal degradation factors
Propagation	LOS, ground wave, sky wave, multipath
Bandwidth	Determines data capacity
Standards	IEEE, 3GPP, Bluetooth SIG, LoRaWAN

I.7.2 GSM and SMS Technology

GSM (Global System for Mobile Communications) is one of the most widely used digital mobile telephony standards in the world. It provides voice, data, and SMS (Short Message Service) communication over a cellular network. [8]

GSM is a second-generation (2G) digital cellular network that allows:

- Voice calls
- Text messaging (SMS)
- Low-speed data transfer (GPRS)

It uses Time Division Multiple Access (TDMA) technology to divide frequency bands into time slots, allowing multiple users to share the same channel efficiently.

SMS (Short Message Service) is a text messaging service component of most telephone, internet, and mobile device systems.

Features of SMS:

- Maximum message length: 160 characters per message
- Supports text only, not multimedia
- Works even in low-signal conditions
- Reliable delivery mechanism

How GSM and SMS Work Together

Sending an SMS:

- 1-The sender types and sends the message.
- 2-The message is routed through the Mobile Switching Center (MSC) .
- 3-The message is stored in the Short Message Service Center (SMSC) .
- 4-The SMSC forwards the message to the recipient's network.
- 5-If the recipient is unavailable (e.g., phone off), the message is stored and retried periodically.

SMS Transmission Modes:

- Point-to-point (PP): Between two mobile devices
- Cell Broadcast (CB): One-to-many broadcast within a geographic area
- Short Message Peer-to-Peer (SMPP): Used by applications and servers to send/receive SMS

GSM Module in Embedded Systems

In embedded systems like Arduino, Raspberry Pi, or industrial control systems, GSM modules are used to enable remote communication via:

- SMS
- Voice calls
- GPRS-based internet access

Table I.10 Common GSM Modules

MODULE	MANUFACTURER	FEATURES
SIM800 / SIM900	Simcom	SMS, GPRS, voice
SIM7600	Simcom	Supports 4G, GPS, Bluetooth
A6/A7	AI Thinker	Cost-effective, basic SMS & call support
Quectel M95	Quectel	Compact, reliable, supports SMS/GPRS

Table I.11 Applications of GSM and SMS Technology

FIELD	APPLICATION
Agriculture	Remote irrigation alerts, soil moisture notifications
Security	Intrusion detection, alarm systems

Healthcare	Patient monitoring, emergency alerts
Home Automation	SMS-based control of lights, appliances
Industrial	Machine status updates, fault alerts
IoT Devices	Remote sensors sending data via SMS or GPRS

I.8 Renewable Energy in Smart Systems

Renewable energy plays a vital role in the development of smart systems, especially in applications like smart cities, IoT, agriculture, and remote monitoring devices. By integrating renewable energy sources such as solar, wind, hydro, and bioenergy, smart systems can operate independently, sustainably, and cost-effectively. [9]

how can we use wind or hydro energy in smart systems?.

- Smart Agriculture
- Environmental Monitoring
- Smart Cities
- Remote Surveillance
- Industrial IoT
- Healthcare

I.8.1 The Importance of Solar Energy in Agriculture

Solar energy has become a game-changer in agriculture, offering farmers an affordable, sustainable, and reliable power source for various operations. As the agricultural sector faces increasing demands due to population growth and climate change, solar energy provides a clean solution to reduce dependency on fossil fuels and grid electricity while improving productivity and cost-efficiency. [9]

Applications of Solar Energy in Agriculture

1. Solar-Powered Irrigation

- Drip and sprinkler systems powered by solar pumps.
- Eliminates diesel generator costs and pollution.
- Ideal for remote fields with no access to grid electricity.

2. Greenhouse Climate Control

- Solar energy powers fans, heaters, cooling systems, and LED grow lights.
- Helps maintain optimal temperature, humidity, and light conditions for crops.

3. Crop Drying & Processing

- Solar dryers and ovens are used to dehydrate fruits, vegetables, and grains.

- Prevents spoilage and improves food preservation.

4. Livestock Farming

- Powers automatic feeders, water dispensers, ventilation systems, and lighting in barns/poultry farms.

5. Cold Storage Units

- Solar-powered refrigeration units help preserve perishable crops and dairy products.

6. Agricultural Sensors & IoT Devices

- Solar panels power smart sensors that monitor soil moisture, temperature, humidity, and weather.

I.8.2 Integration of Renewable Energy with Control Systems

The integration of renewable energy with control systems is a key development in modern engineering, enabling smart, sustainable, and automated power management across various applications such as agriculture, industry, smart homes, and IoT devices. [10]

This integration allows for:

- Efficient energy use
- Automation of energy generation, storage, and consumption
- Remote monitoring and control
- Improved reliability and sustainability[10]

Practical Applications

1. Solar Water Pumping System

- **Control Action:** Start pump when solar irradiance is sufficient; stop when tank is full or sun sets.
- **Components:** Arduino, water level sensor, relay, solar panel, submersible pump.

2. Smart Home Energy Management

- **Control Action:** Distribute solar energy among appliances, charge batteries, switch to grid if necessary.
- **Components :** ESP32, current/voltage sensors, relays, mobile App.

3. Agricultural Irrigation

- **Control Action:** Activate irrigation system based on soil moisture and available solar energy.
- **Components:** Soil moisture sensor, Arduino, GSM module, water pump.

Benefits of Integrating Renewable Energy with Control Systems

- ✓ Energy Efficiency – Optimized generation, storage, and usage
- ✓ Automation – Reduces human intervention
- ✓ Remote Monitoring – Enables control from anywhere via IoT
- ✓ Cost Savings – Lowers dependency on diesel/grid electricity
- ✓ Scalability – Easily expandable with more panels or storage
- ✓ Environmental Impact – Promotes clean and green energy practices

I.9 Conclusion

Smart agriculture, based on intelligent control systems and reliable communication networks, is reshaping the way food is produced globally. It is not merely a technological trend, but a strategic necessity to ensure sustainable food security and to strike a balance between meeting the needs of a growing population and preserving environmental resources for future generations. [11]

Chapter II

Technical components of the smart agricultural control system

II.1 Introduction

The apparatus enables the collection, analysis, and real-time automated decision making. A detailed description is given in this chapter regarding the operation of the system which includes microcontrollers, sensors, wireless communication, and information technologies as well as alternative energy sources.

Research relied upon the practical example of the analysis framework built upon the functionality of system components throughout the engineering process. [1]

II.2 The Arduino Microcontroller

II.2.1 Arduino

Arduino is an open-source electronics platform based on easy-to-use hardware and software. It is designed to make the process of building interactive electronic systems accessible to a wide range of users, from beginners to professionals. The platform primarily consists of a programmable microcontroller board (such as the Arduino UNO or Mega) and an integrated development environment (Arduino IDE) used for writing and uploading code to the board. among its main components: [4]

1. Microcontroller:

- The core-processing unit of the board, responsible for executing the program instructions.
- A widely used model is the ATmega328P in Arduino UNO boards.

2. Digital and Analog Input/Output Pins:

- These pins allow the connection of sensors, actuators, LEDs, and other devices.
- Digital pins handle binary signals (HIGH/LOW), while analog pins read varying voltage levels from analog sensors.

3. Power Pins:

- Provide regulated voltage (e.g., 5V, 3.3V) and ground (GND) to external components.

4. Programming and Communication Interface:

- Typically, via a USB port to upload programs from a computer.
- Supports serial communication (UART) to interface with external modules like GSM, Bluetooth, or Wi-Fi

Figure II.1 arduino uno

II.2.2 Main Components of Arduino

This table 2.2 shows all the components we will be using in our project:

Table II.2 Main Components of Arduino

THE COMPONENTS	FUNCTION
Breadboard	Assembling electronic circuits without soldering
LED	Emits light when electric current flows through it
Humidity & Temp Sensor	Measures ambient humidity and temperature
Relay Module	Acts as a switch between low-powered digital electronics and high-powered devices
Soil Moisture Sensor	Measures the moisture level of the soil
Water Level Sensor	Detects the water level in tanks or containers
Pump 5v	Water pumping

The smart agriculture system relies on an array of sensors and electronic components that collect environmental data (such as humidity, temperature, and soil moisture) and process it using a microcontroller. This allows the system to make automated decisions—like activating the water pump when necessary—within a flexible, programmable framework. [5]

Here is the image that demonstrates the components mentioned:



Figure II.2 main components of arduino

II.2.3 Programming Arduino with Arduino IDE

Arduino boards are programmed using the Arduino IDE, which supports a simplified version of C/C++. It offers a rich library of built-in functions and examples that simplify sensor integration and control logic. [4]

The interface of the program consists of 6 sections:

- Toolbar Section
- Menus Section
- Code Editor Section
- Status Bar Section
- Program Notification Section
- Serial Port Selections & Board Section

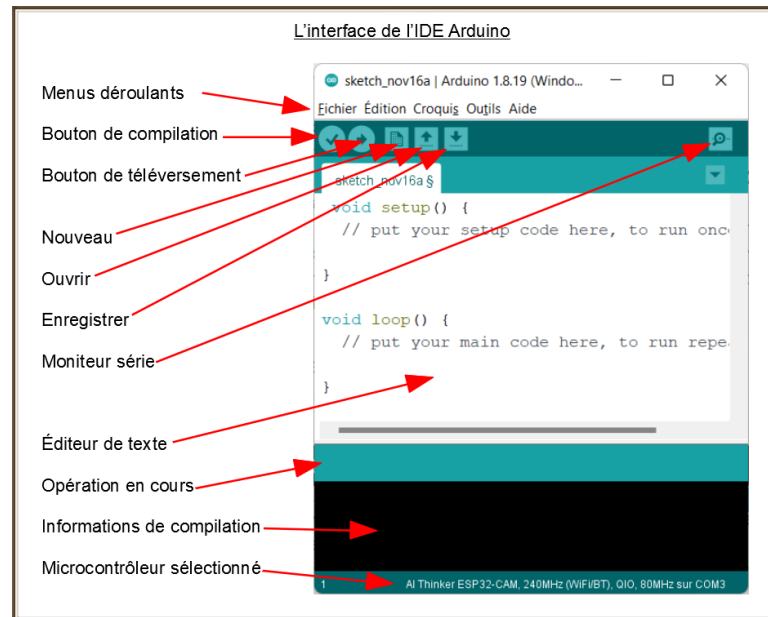


Figure II.3 Arduino IDE Interface and Main Sections
Arduino IDE Program Sections

A. Toolbar

This is one of the most important sections of the Arduino IDE, as it contains the essential tools you will frequently use when programming the Arduino board:

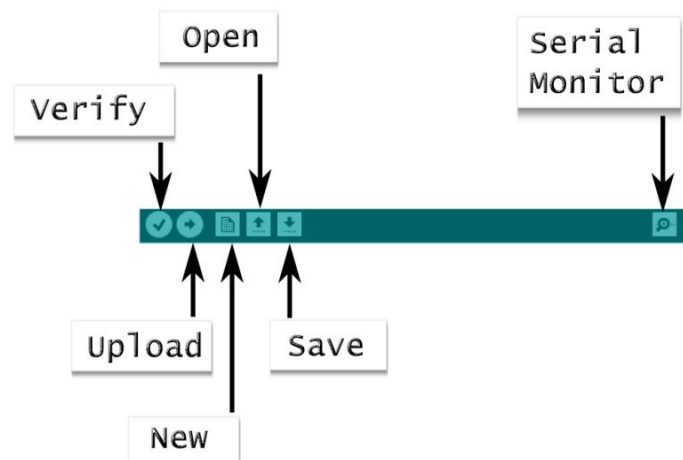


Figure II.4 Toolbar section

- **Upload**: This button is used to upload your code to the Arduino board.
- **New**: Use this to create a new project or sketch.
- **Open**: Allows you to open an existing sketch from the Sketchbook.
- **Save**: Saves the current sketch to your Sketchbook.
- **Serial Monitor**: Displays data sent from the Arduino board.

B. Menus

These are the main menus of the Arduino IDE. There are five of them: **File**, **Edit**, **Sketch**, **Tools**, and **Help**. They are used to add or modify the code you are writing. The **Tools** menu also allows you to configure board settings.

B.1 File Menu

This menu includes several options such as opening, saving, and printing sketches or code files. It also contains the **Preferences** option, which allows you to customize the interface — for example, changing the language, font size, and other settings.

B.2 Edit Menu

The Edit menu offers tools for modifying your code. You can copy, paste, select all, format your code, and add comments. It also provides options for adjusting text size and searching within your sketch.

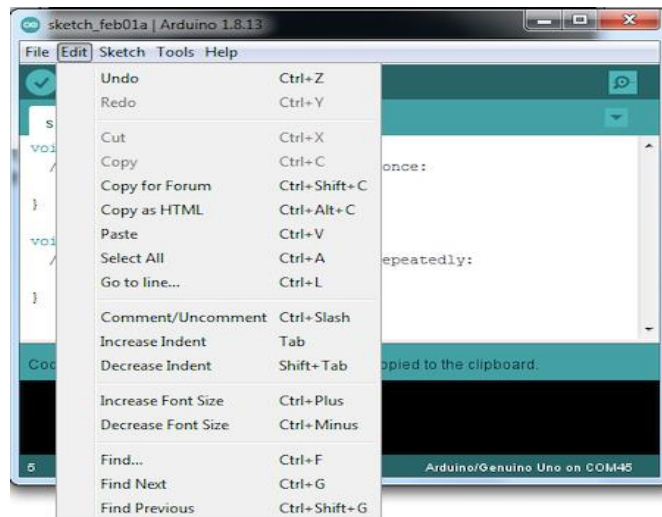


Figure II.5 menu section

B.3 Sketch Menu

The Sketch menu contains several important options, such as:

- **Verify/Compile:** This command reviews the written code and checks it for errors. "Compiling" refers to converting the code into machine language that the Arduino microcontroller can understand.
- **Upload:** This uploads the compiled code directly to the Arduino board.
- **Show Sketch Folder:** Opens the folder where the current sketch (code file) is saved.
- **Include Library:** Allows you to add predefined libraries (sets of code) to your project, such as libraries for the keyboard or motor control.



Figure II.6 sketch menu section

B.4 Tools Menu

The Tools menu offers various options, including:

- **Board:** Used to select the type of Arduino board you are working with.
- **Port:** Lets you choose the communication port used to connect the Arduino to your computer.
- **Auto Format:** Automatically formats your code to improve readability.
- **Archive Sketch:** Compresses your sketch into a ZIP file and saves it to your computer.
- **Burn Bootloader:** Used to burn a new bootloader onto the Arduino's microcontroller.

Note: The bootloader is a small piece of code that makes the microcontroller compatible with the Arduino IDE. This option is only used if you have an AVR programmer and have replaced the microcontroller on the Arduino board.

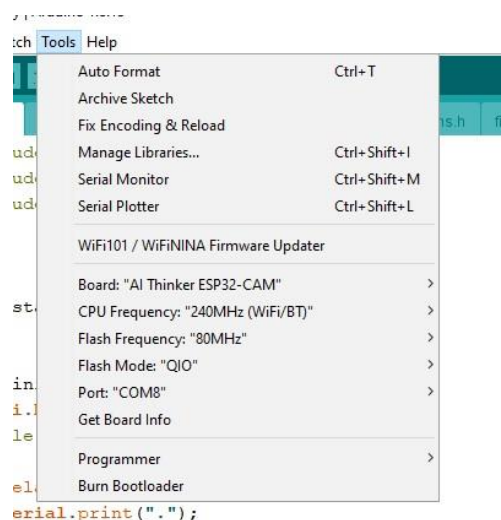


Figure II.7 tools menu section

B.5 Help Menu

This menu provides access to documentation about the Arduino IDE and helpful resources for troubleshooting and learning.

C. Code Editor

The Code Editor is the white workspace where you write and modify your Arduino code.

D. Status Bar

The Status Bar is located at the bottom of the Code Editor. It shows the status of current operations, such as compiling and uploading the code.

E. Program Notifications

This section displays error messages or warnings if there are problems during the coding or uploading process. It indicates the type and cause of the error, and may provide suggestions to help you fix it.

F. Serial Port & Board Selection

This area displays the currently selected communication port and Arduino board type, confirming the hardware configuration used for programming.

II.2.4 Arduino in Smart Agriculture

Arduino can be effectively utilized in the field of smart agriculture through the following applications:

1. Detection of Irrigation and Fertilization Needs:

Arduino can be employed to measure soil moisture levels, ambient temperature, and humidity. Based on pre-defined thresholds, the system can be programmed to notify the user when irrigation or fertilization is required, thereby optimizing water and nutrient management.

2. Monitoring Plant Growth:

By integrating various sensors and cameras, Arduino can function as a surveillance system that monitors plant development. It can help identify growth abnormalities or stress conditions that may require additional attention.

3. Temperature and Humidity Control:

Arduino can be used to automate environmental control systems by activating irrigation, ventilation, or heating devices according to programmable parameters, ensuring optimal microclimatic conditions for plant growth.

4. Light Regulation:

Using light sensors, Arduino can control artificial lighting systems to maintain suitable illumination levels for plant photosynthesis. The system can be programmed to automatically respond to changes in natural light intensity.

5. Air Quality Monitoring:

Arduino-based systems can monitor ambient air quality by detecting harmful gases or pollutants. In the event of hazardous conditions, real-time alerts can be generated to ensure timely intervention.

II.3 Sensors in Smart Agriculture

Agricultural sensors form the backbone of smart farming systems. They represent the first step in the data collection chain, upon which intelligent agricultural management decisions are based. These sensors act as electronic sensory organs, precisely monitoring environmental conditions, soil properties, and crop health. [2]

II.3.1 Types of Agricultural Sensors

II.3.2 Environmental Sensors

These sensors collect data on climate conditions inside the field or greenhouse:

- DHT11/DHT22: Temperature and humidity
- LDR: Light intensity

They help optimize crop growth conditions.



Figure II.8 environmental sensors

II.3.3 Soil Moisture Sensors

Soil moisture sensors, such as YL-69, measure the water content in soil, allowing automated irrigation when levels drop below a threshold.



Figure II.9 YL-69 Soil Moisture Sensor Used in the Smart Irrigation System

II.3.4 Choosing the Right Sensor

Sensor selection depends on: [12]

Table II.2 Sensor selection

Criterion	Requirements	Practical Examples
Accuracy	±1–2% for critical measurements	±0.5°C for temperature readings
Operating Range	-20°C to +60°C	For harsh agricultural conditions
Protection Rating	Minimum IP67	Suitable for outdoor installation
Power Consumption	≤10mA for battery-operated systems	LoRaWAN-based technologies

II.4 Wireless Communication Technologies in Agriculture

II.4.1 Overview

Wireless communication technology in smart agriculture refers to the use of wireless communication devices such as local wireless technology (WiFi), mobile communication technologies (GSM, 4G, 5G), sensor networks (LoRaWAN, Zigbee), and low-energy wireless networks (BLE) to exchange data between sensors and a central hub in smart farming systems. This technology is utilized to collect data related to crops and their surrounding environment, which is then analyzed to enhance productivity and enable more efficient and effective farm management.

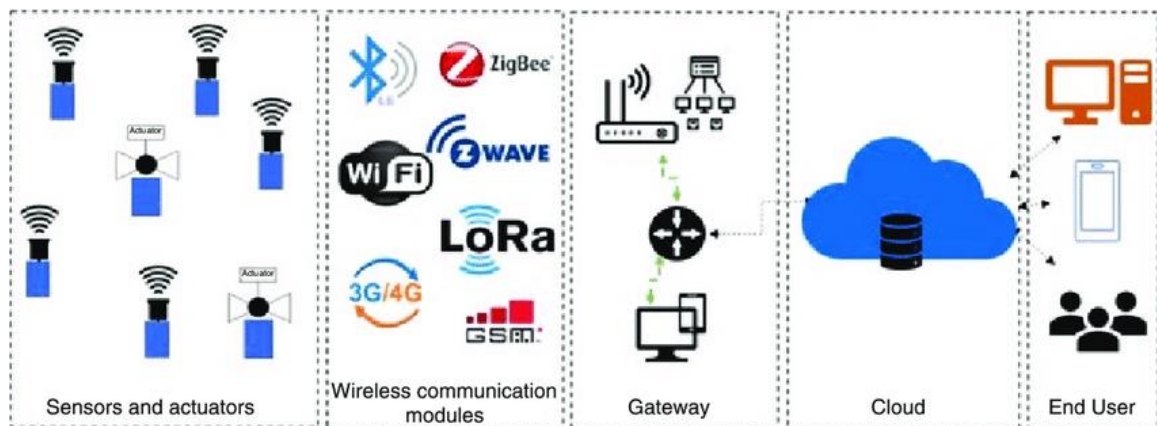


Figure II.10 comprehensive communication

This table highlights the various wireless technologies used in smart irrigation systems:

Table II.3 Comprehensive Communication for Precision

Communication Protocols	Standard Rate	Transmission Range	Data Rate	Topology	Power
BLE	IEEE 802.15.1	30 m	1–24 mbps	Star, bus	Low
Zigbee	IEEE 802.15.4	10–100 m	20–250 kbps	Star, mesh cluster	Low
Wi-Fi	IEEE 802.11	10–200 m	2–54 mbps	Star	Medium
4G LTE	LTE	<15 km	100 mbps	M2M, star	High
LoRaWAN	IEEE 802.11ah	5–10 km	0.3–50 kbps	M2M, star	Medium

II.4.2 GSM and SMS Technology

GSM (Global System for Mobile Communications) is a digital cellular telephony standard widely applied worldwide. GSM supports voice call reception and transmission, SMS (Short Message Service), and data. GSM is especially reliable in rural or remote areas where internet connection is weak. [8]

One of the primitive services provided by GSM is SMS, which can be used to send short text messages up to a size of 160 characters. It finds common application in remote monitoring and control scenarios in embedded systems.

By integrating GSM with SMS capabilities through modules such as SIM800L, effective remote-control systems can be established. They enable command execution as well as data retrieval in the absence of the internet, making them especially suitable for smart agriculture and IoT-based projects.

II.4.3 Sending and Receiving Messages via GSM

1. System Components

- User: Sends and receives SMS messages over a mobile phone.
- GSM Network: Provides messages to and from the SIM800L module.
- SIM800L Module: Interacts with the microcontroller to connect with the GSM network.
- Microcontroller (e.g., Arduino): Runs received instructions and controls actuators.
- Sensors and Actuators: Read environmental information and perform actions (e.g., water pump on).

2. Receiving SMS from the User

- The user sends an SMS command (e.g., "ON1") to the SIM card number integrated into the SIM800L.
- The message is routed via the GSM network to the SIM800L module.
- The module passes the message to the microcontroller via UART (serial communication).
- The microcontroller reads the message using the AT+CMGR command and AT+CMGD to delete it later.
- The microcontroller reads the message and the right action is performed.

3. Sending SMS to the User

The microcontroller reads data from sensors (e.g., temperature, humidity, water level).

- A message is formed which contains the data (e.g., "Temp: 28°C, Humidity: 45%, Water: Full").

- The message is sent using AT commands:
 - AT+CMGF=1 for setting text mode.
 - AT+CMGS="recipient_number" and then the message and Ctrl+Z to send.

4. Communication Flow Overview

[USER] → SMS → [GSM Network] → [SIM800L] → [Microcontroller] → Perform Action
 [Microcontroller] → Sensor Reading → Build SMS → [SIM800L] → [GSM Network] → [USER]

5. Common AT Commands

Table II.4 Common AT Commands

command	Function
AT	TEST COMMUNICATION WITH SIM800L
AT+CMGF=1	ENABLE SMS TEXT MODE
AT+CMGR=1	READ SMS FROM MEMORY
AT+CMGS=1	SEND SMS TO THE SPECIFIED NUMBER
AT+CMGD=1	DELET SMS FROM MEMORY

6. Advantages of Using This Mechanism

- Does not require internet connectivity.
- Effective in remote or rural areas.
- Easy to program and integrate with microcontrollers.
- Cost-efficient compared to internet-based solutions.

II.4.4 Comparison of Wireless Communication Technologies

Table II.5 Comparison of Wireless Communication Technologies

Technology	Range	Power Consumption	Cost	Best Use Case
GSM	Long (global)	Medium	Medium	Remote farms with no Wi-Fi
Wi-Fi	Short	High	Low	Close to router
LoRa	Very Long	Low	High	Large, isolated areas
ZigBee	Medium	Low	Medium	Local sensor networks

Observations:

- **GSM** is ideal when global coverage is needed and cellular networks are available. It’s useful for remote control and data logging where no local internet infrastructure exists.
- **Wi-Fi** works well in environments with existing infrastructure but consumes more power, which makes it less suitable for battery-powered devices.
- **LoRa** shines in wide-area, low-power applications where only small amounts of data need to be sent periodically—like agricultural field monitoring.
- **ZigBee** is optimal for mesh networks of nearby devices (e.g., sensors) that don’t need long-range communication and must preserve power

II.5 Integrating renewable energies into smart agriculture systems

Renewable energies, when integrated into smart agriculture systems, are naturally sustainable, replenish able energy sources (e.g., solar, wind, and geothermal energy) used for the powering of automatic monitoring, and control systems in agriculture. The systems leverage the convergence of Internet of Things (IoT) technologies, smart sensors, and computing algorithms for improved efficiency in the use of water and energy resources as well as reduced fossil fuel dependency. It is characterized by the ability to enable. [9]

Sustainable dynamic management of the farm through data and real-time decision-making based on crop and environmental conditions

The next figure shows

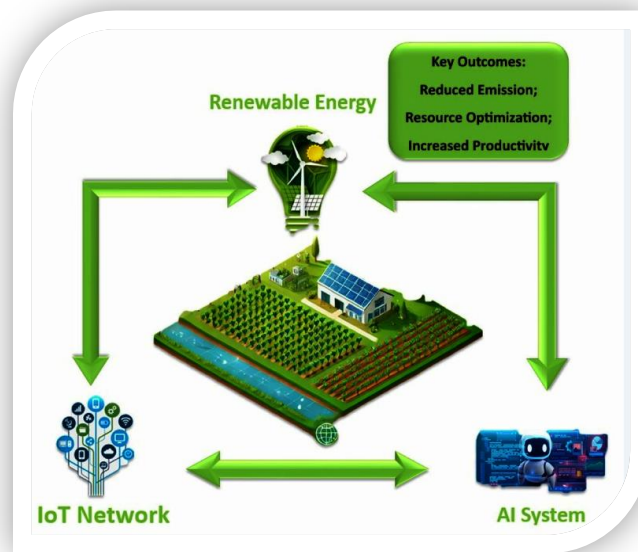


Figure II.11 Integration of Renewable Energy into the Smart Agriculture System

II.5.1 The Importance of Renewable Energy

Renewable energy is a key element in smart farm development since it assists towards achieving environmental sustainability and optimum economic efficiency. The following are the key factors that highlight its importance:

Energy Savings and Cost Reduction

- Lower electricity bills: The integration of solar panels or wind turbines minimizes reliance on traditional power grids.
- Smart irrigation systems: Solar-powered water pumps reduce fuel consumption and operational costs.
- Reduced energy waste: Intelligent control systems distribute energy based on real-time demands, increasing efficiency.

Environmental Sustainability

- Reduced carbon emissions: The use of clean energy sources (solar, wind, biomass) helps decrease greenhouse gas emissions.
- Efficient water management: Solar-powered irrigation systems optimize water usage and minimize waste.
- Organic waste recycling: Agricultural waste is converted into biogas or used as natural fertilizer, promoting circular resource use.

Increased Productivity and Crop Quality

- Smart monitoring technologies: Renewable energy powers sensors and cameras that track soil conditions, humidity, and temperature.
- Climate control in smart greenhouses: Solar energy is used to regulate temperature and humidity for optimal crop growth.
- LED lighting systems: Energy-efficient lighting enhances plant development, especially in vertical farming environments.

Energy Independence and Reliability

- Energy storage solutions: Excess energy can be stored using batteries or green hydrogen systems, ensuring uninterrupted farm operations.
- Off-grid capability: Remote agricultural areas can operate independently using renewable energy sources.
- Climate adaptability: Hybrid renewable systems (solar and wind) offer resilience and stability in the face of climate variability.

II.5.2 Solar Energy as a Primary Source

Solar energy is considered a primary source of renewable energy due to its abundance, sustainability, and potential to meet the growing energy demands of various sectors, particularly agriculture. Unlike fossil fuels, solar energy is clean and inexhaustible, making it an ideal solution for long-term energy needs.

- **Abundance and Accessibility:** Solar radiation is available worldwide, with regions having abundant sunlight offering significant energy production potential. This makes solar energy an accessible source, especially in rural and remote areas where traditional grid connections are often limited.
- **Efficiency and Low Operational Costs:** Solar power systems, such as photovoltaic (PV) panels, convert sunlight into electricity efficiently with minimal operational costs. Once installed, solar panels require very little maintenance and can operate for decades, ensuring long-term, cost-effective energy generation.
- **Environmental Benefits:** Solar energy produces no greenhouse gas emissions during its operation, making it a clean alternative to fossil fuels. Its use significantly reduces the carbon footprint of agricultural operations, contributing to environmental sustainability by decreasing air and water pollution.
- **Energy Autonomy for Remote Areas:** Solar energy systems enable agricultural operations in off-grid areas to become energy self-sufficient. By integrating solar power into smart

farming solutions, farmers can rely on a stable and renewable energy source to power irrigation systems, sensors, climate control systems, and more.

- **Scalability and Flexibility:** Solar energy systems can be scaled according to the specific energy needs of a farm. From small-scale installations for individual sensors or irrigation pumps to large solar farms powering entire agricultural operations, solar energy provides the flexibility to adapt to various farm sizes and requirements.



Figure II.12 solar energy

II.5.3 Integrating Solar Power with Control Systems

It is the use of solar energy as a renewable power source to operate smart control systems managed by controllers (such as Arduino or PLC) for monitoring and controlling agricultural or industrial processes.

Through this integration, it is possible to achieve:

- **Energy independence** in rural or remote areas.
- **Reduced operational costs** (no need for grid electricity).
- **Enhanced environmental sustainability.**
- **Continuous and uninterrupted operation** of smart devices.

II.6 Smart Agricultural System Design

Designing a smart agriculture system involves building an integrated structure based on modern technologies to collect data, analyze it, and make automated decisions with the goal of improving agricultural productivity, reducing waste, and increasing efficiency. Below is a scientific and organized explanation of the system components: [13]

Sensors:

They act as the system's eyes, collecting data from the agricultural environment, such as:

- 1. Humidity and Temperature Sensor (DHT11 or DHT22)**

- Measured Parameters: Relative Humidity (%RH), Temperature (°C)
- Humidity Range: 20–90% RH \pm 5%
- Temperature Range: 0–50°C \pm 2°C
- Voltage Supply: 3.3–5.5V DC
- Output: Digital signal (single-wire serial)

2. Soil Moisture Sensor

- Measured Parameter: Soil moisture content (analog value)
- Working Voltage: 3.3–5V DC
- Output: Analog and digital output
- Features:
- Corrosion-resistant probes
- Adjustable sensitivity

3. Water Level Sensor

- Measured Parameter: Water level in tanks
- Working Voltage: 3–5V DC
- 4. Relay Module (1-Channel 5V)
- Input Voltage: 5V DC
- Control Signal: Digital (from Arduino)
- Load Capacity: 10A 250V AC / 10A 30V DC
- Isolation: Optocoupler isolation for safety

5. Water Pump (5V Mini Submersible)

- Operating Voltage: 3–5V DC
- Flow Rate: ~80–120 L/h
- Power Consumption: ~0.4–0.8W
- Lifting Height: 40–110 cm

Controller Unit:

The system's brain, such as:

- Arduino, Raspberry Pi, or PLC: These devices process data from sensors and make automatic decisions (e.g., turning on a pump or sending an alert).

Actuators:

They execute commands sent by the controller, including:

- Water pumps
- Irrigation valves
- Fans or heating/cooling units in greenhouses

Power Supply:

- Solar energy: Used to power the system in remote areas.
- Batteries: Store energy for night-time or cloudy-day operation.

Communication Modules:

Used to transmit data to users or servers, such as:

- SIM800L (GSM): Sends alerts or reports via SMS or the internet.
- Wi-Fi, LoRa, or ZigBee: Depending on the infrastructure and distance.

User Interface:

Could be:

- A mobile app or web dashboard
- Allows users to monitor the system, read data, and control it manually when needed.

Cloud Platform:

- Stores and analyzes data over time (e.g., trend analysis, identifying recurring issues)

II.6.1 System Requirements Analysis

Analyzing requirements is a crucial step in designing and implementing any smart system. It helps define the necessary hardware, software, and system functions to ensure that the system achieves its goals efficiently and effectively.

❖ Functional Requirements

These are the operations or tasks the system must perform:

- Continuous soil moisture measurement.
- Automatic irrigation activation when moisture drops below a certain threshold.
- Monitoring temperature and humidity in fields or greenhouses.
- Sending alerts via SMS or internet when faults occur (e.g., low water level).
- Storing agricultural condition data in a local or cloud database.
- Remote manual control via a user interface (app or control panel).
- Full operation using solar energy to ensure energy independence.

❖ Non-Functional Requirements

These are qualities or characteristics the system must possess:

1. Reliability: The system should operate continuously without failure.
2. Sustainability: Low energy consumption with reliance on renewable sources.
3. Security: Secure data exchange using encryption, especially when using the internet.
4. Scalability: Ability to add more sensors or units in the future without redesign.
5. Usability: A simple and clear interface suitable for farmers.

6. Real-time responsiveness: The system must react to live data and make instant decisions.

❖ Hardware Requirements

- Microcontroller such as Arduino or Raspberry Pi.
- Sensors:
 1. Soil moisture
 2. Temperature
 3. Air humidity
 4. Water level
- Communication modules like SIM800L, LoRa, or Wi-Fi Module.
- Solar power system + battery for energy storage.
- Irrigation valves and water pump.

❖ Software Requirements

- Firmware for the microcontroller to read sensor data and execute control logic.
- User interface (dashboard) to display data and allow user control.
- Database system (local or cloud-based) for data storage.
- Control and decision-making algorithms based on environmental thresholds.

II.6.2 General Architecture of the System

The system consists of an integrated set of units that work in coordination to collect environmental data, process it, and make smart control decisions automatically. The structure includes the following

Processing Unit - Arduino

The Arduino unit is the main board of the system, functioning as the central processor. It reads data from the connected sensors, such as soil moisture and temperature sensors, and then makes appropriate decisions based on this data. **Arduino** controls the operation of other devices like pumps or relays to perform the required tasks, such as turning on the pump when the soil needs irrigation.

Sensors

❖ Soil Moisture Sensor

The **Soil Moisture Sensor** measures the moisture level in the soil by assessing its electrical conductivity. The more moisture present, the greater the soil's ability to conduct electricity. This data is sent to the **Arduino** processing unit, which uses it to make decisions, such as turning on the pump if the soil moisture is low.

❖ Water Level Sensor

The **Water Level Sensor** is used to measure the water level in a tank or pond. This sensor helps ensure that the water supply in the irrigation system doesn't run out. It sends the information to the **Arduino** unit, which can take action if the water level is low, such as turning on the water pump or sending an alert to the farmer.

❖ Temperature and Humidity Sensor

The **Temperature and Humidity Sensor** measures the environmental conditions surrounding the plants. It monitors the temperature and humidity levels, helping to determine if there's a need to adjust the climate in greenhouses or agricultural rooms. This data is sent to **Arduino**, which uses it to control processes like turning on air conditioning or heating.

Pump

The Pump is a device that provides water to the plants when needed. It is typically used in smart irrigation systems that operate automatically. The Arduino controls the pump's operation based on data gathered from sensors like the soil moisture or water level sensor, ensuring that the pump only runs when the soil is dry or the water level is low.

Relay

A Relay is an electrical switch used to control high-voltage devices using a signal from the Arduino. The relay can be used to turn devices such as pumps or motors on or off. When the sensors detect a need for adjustment (like watering the soil or running equipment), the Arduino sends a signal to the relay to carry out the necessary change

SIM800L Module (GSM Module)

- **Description:**

The **SIM800L** is a GSM/GPRS communication module that provides wireless connectivity via mobile networks. It can send and receive data through SMS (Short Message Service), voice calls, and internet data via 2G networks (GPRS).

- **Functions:**

- ✓ **Send SMS:** The system can send text messages to the farmer when there's a need to make adjustments in the system (e.g., when water levels are low or soil moisture is insufficient).
- ✓ **Remote Control:** The farmer can send text messages to the system to control devices, such as turning the pump on or off.
- ✓ **Data Transfer:** Data collected from sensors can be transmitted to servers or the farmer via the internet using GPRS, allowing continuous system monitoring.

- ✓ **Communication in Remote Areas:** In rural or remote areas where internet networks are unavailable, the **SIM800L** module provides connectivity through the GSM network, enabling remote monitoring and control of the system

II.6.3 System Block Diagram

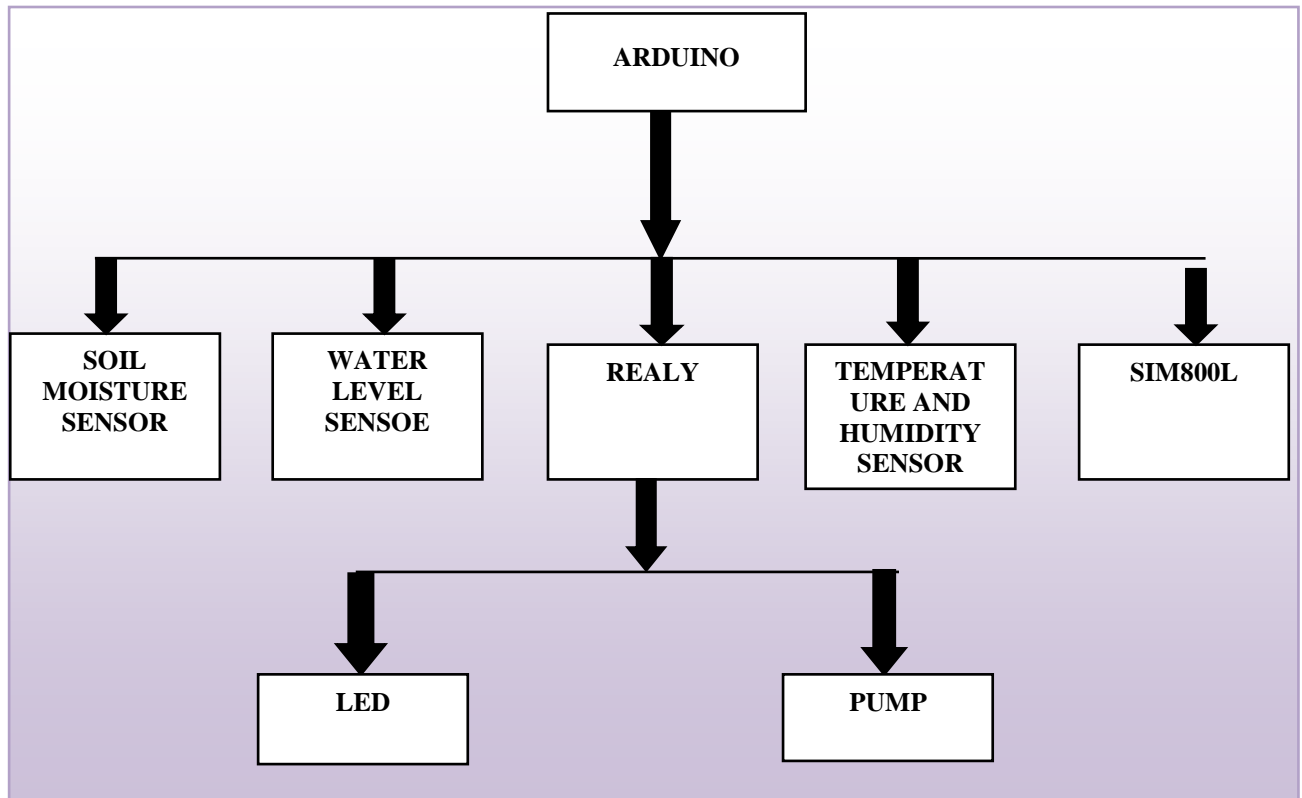


Figure II.13 The following figure shows the system block diagram[14]

II.6.4 Control Strategy

1. **Automatic Mode (Auto Mode):** The Arduino periodically reads soil moisture.
 - If the moisture level drops below 30%, the pump is activated automatically.
 - If the moisture level rises above 40%, the pump is turned off.
 - A notification message is sent automatically whenever the pump is turned on or off.
2. **Manual Mode:**
 - The pump does **not** operate automatically.
 - Only on and off commands sent via SMS are executed.
3. **inf Command:** Sends a message containing the current temperature, soil moisture level, current mode (auto/manual), and pump status (on/off).

II.7 Conclusion

This chapter provided a detailed analysis of the core technical components behind the smart agricultural control system. From the Arduino microcontroller and various sensors to wireless communication and renewable energy integration, these components are vital for the system's functionality and sustainability. The next chapter will move toward real-world implementation and testing of the system. [11]

Chapter III

Simulation and practical implementation of the smart agricultural system

III.1 introduction

In light of accelerating climate change and increasing pressure on natural resources—particularly water—traditional agriculture has become incapable of meeting the demands of sustainable development and ensuring food security. Consequently, there has been a growing need to integrate modern technologies into agricultural systems in order to enhance productivity, optimize resource consumption, and reduce human intervention. This has given rise to what is known as smart agriculture, which relies on remote monitoring and control systems.

This applied project emerges as a direct response to these challenges. It involves the design and implementation of an intelligent agricultural control system that combines automated and semi-automated modes for irrigation and lighting management. The system operates based on the principle of sensing and responding, using sensors connected to a central processing unit (Arduino), in addition to a SIM800L communication module that enables remote interaction via Short Message Service (SMS).

Section one: virtual simulation environment

III.2 simulation environment and tools used

This phase focused on building a virtual environment that accurately simulates the real system, with the goal of testing the behavior of its components before proceeding to on-site implementation. Simulation helps verify electrical connections, the logic of the program, and the system's responsiveness to various environmental variables, thus minimizing the likelihood of technical errors or costly malfunctions later on.

The simulation was carried out using Proteus, one of the most widely used software platforms for simulating electronic circuits and embedded systems. It enables accurate simulation of Arduino microcontrollers alongside components such as relays, sensors, communication modules, and actuators in a visual and interactive manner.

The tools used in the simulation environment include:

- Arduino UNO board: Serves as the main control unit responsible for executing the programmed logic.
- SIM800L communication module: Simulates the sending and receiving of SMS messages in the semi-automated control mode.
- Soil moisture sensors: Measure soil moisture levels in the left and right zones of the field.

- Temperature and humidity sensors (DHT11 or DHT22): Monitor environmental conditions in both zones.
- Five relays: Control the activation of three water pumps (two for irrigation and one for drawing water from a well) and two lamps.
- Virtual pumps and lamps: Act as actuators that are turned on or off based on sensor readings or received commands.
- Arduino IDE: Used to develop the control program and generate the executable hex file for simulation within Proteus

III.2.1 Software platforms (proteus, tinkercad, Arduino ide)

To design a realistic and efficient virtual simulation, it is essential to rely on comprehensive and integrated software platforms that support electronic circuit design and functional system modeling. These platforms enable researchers and developers to visualize, simulate, and validate system performance before transitioning to the practical implementation phase. Among the most important of these platforms are:

- ✓ **Tinkercad** Tinkercad is a browser-based 3D design and simulation tool developed by Autodesk. It includes a simplified Arduino simulator that allows users to quickly prototype and test basic circuits and code in a virtual environment. Tinkercad was initially used during the early design phase to visualize circuit behavior and validate simple interactions between sensors, actuators, and the Arduino board.
- ✓ **Proteus Design Suite** Proteus is a professional-grade electronic design automation (EDA) tool widely used in embedded systems engineering. It was the main platform for simulating the complete circuit of the project. Proteus allows for real-time simulation of Arduino code along with detailed representations of sensors, relays, communication modules, and other components. It provided precise feedback regarding circuit behavior before transitioning to physical implementation.
- ✓ **Arduino IDE** The Arduino Integrated Development Environment (IDE) is the official software used to write, compile, and upload code to Arduino boards. It supports C/C++ programming and provides a simple interface for writing embedded control logic. In this project, the Arduino IDE was used to develop both the automatic and semi-automatic control algorithms, which were then exported to Proteus for simulation.

The following are the desktop icons representing the software tools utilize:

tinkercad



proteus



Arduino ide



Figure III.1 Icons of the Software Tools Used

III.2.2 Virtual component (Arduino, sensors, gsm module)

In the virtual simulation environment created using Proteus software, a set of essential components was integrated to accurately represent the real-world system. These components allow for the simulation of control and response mechanisms based on various environmental conditions, whether in automatic or semi-automatic mode. Below is a detailed overview of the key components:

In the virtual simulation environment created using Proteus software, a set of essential components was integrated to accurately represent the real-world system. These components allow for the simulation of control and response mechanisms based on various environmental conditions, whether in automatic or semi-automatic mode. Below is a detailed overview of the key components:

- ✓ **Arduino UNO Board** The Arduino UNO board serves as the central processing unit of the system. It acts as the brain that receives data from sensors and issues activation or deactivation either commands to actuators such as pumps and lamps— autonomously or based on commands received via SMS. In Proteus, a virtual Arduino UNO board was used, and the control program was uploaded using a HEX file for real-time simulation.
- ✓ **Sensors** A variety of virtual sensors were included to simulate field conditions:
Soil Moisture Sensors: These are used to measure the moisture levels in the soil, with one sensor placed in each side of the field. When the moisture drops below the required threshold, the corresponding pump is automatically activated. Temperature and Humidity

Sensors (DHT11 or DHT22): These sensors monitor the surrounding environmental conditions such as temperature and humidity in both zones of the farm.

- ✓ **GSM Communication Module (SIM800L)** The SIM800L module is used to simulate wireless communication via SMS, enabling remote control of the system. The module is virtually connected to the Arduino board to simulate receiving control commands (e.g., “start pump” or “turn off lights”) and sending confirmation or alert messages to the user.

III.2.3 Setup and configuration of simulation environment

The process of setting up the simulation environment represents a critical step in the system development phases, as it allows for accurate modeling of component behavior and interactions within a virtual environment prior to field implementation. In this project, Proteus and Arduino IDE were utilized to create a comprehensive simulation environment that integrates both virtual hardware and software components.

Relays:

- D2 → Relay 1: Pump for the first side
- D3 → Relay 2: Pump for the second side
- D4 → Relay 3: Lamp for the first side
- D5 → Relay 4: Lamp for the second side
- D6 → Relay 5: Pump from the well to the reservoir

Analog Sensors:

- A0 → Soil Moisture Sensor – first side
- A1 → Soil Moisture Sensor – second side
- A2 → Water level sensor in the reservoir

GSM Communication Module (SIM800L or Virtual Terminal):

- D9 → TX (Arduino transmitting data to GSM)
- D10 → RX (Arduino receiving data from GSM)

Temperature and Humidity Sensors (e.g., DHT11 or DHT22):

- D11 → Temperature/Humidity Sensor – first side
- D12 → Temperature/Humidity Sensor – second side

Note: In this simulation, 12V LEDs were used to represent the water pumps.

III.3 First simulation experiment by proteus (automated moisture control system using Arduino):

In this simulation, a soil moisture sensor is used to monitor the water content in the soil. The sensor sends analog data to the Arduino, which processes the readings and controls the irrigation process automatically. This experiment demonstrates the basic principle of automated moisture control in smart agriculture using proteus software.

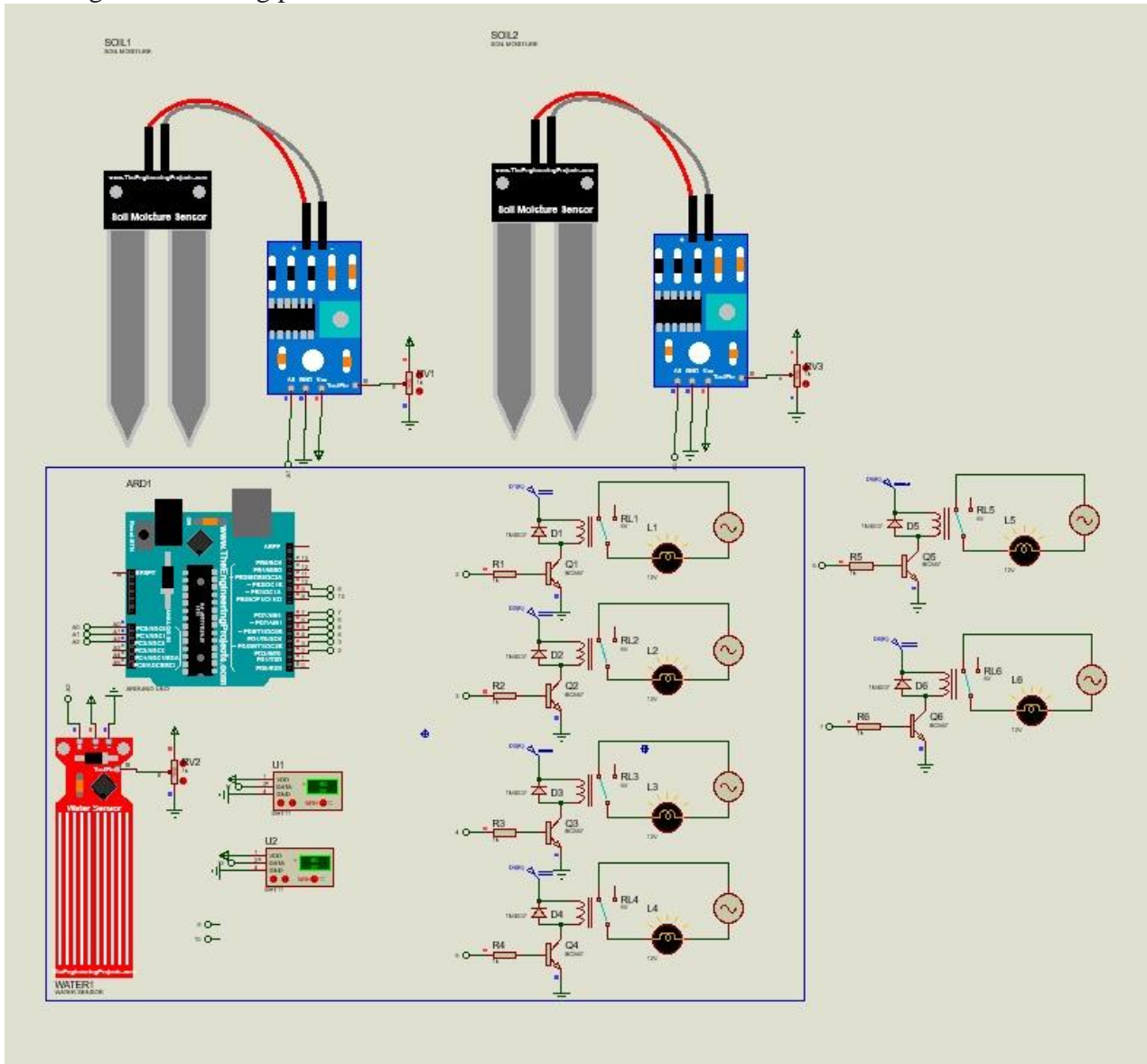


Figure III.2 Full-Automatic System Simulation Using Proteus

III.3.1 Experiment objectives

This experiment aims to implement and validate an autonomous irrigation control system based on soil moisture levels. The primary goals are as follows:

- To configure the components of the system within a virtual simulation environment using Proteus and Arduino UNO.

- To develop and apply a control algorithm that autonomously regulates water pumping based on real-time soil moisture readings.
- To evaluate the system's performance and responsiveness under different environmental conditions, simulating field scenarios.
- To analyze the reliability and efficiency of the automatic mode before transitioning to the real-world implementation phase

III.3.2 System setup in simulation

The simulation environment was built using Proteus, a powerful tool for electronic circuit design and embedded system testing. The system setup included the following components:

- Arduino UNO as the central control unit.
- Two soil moisture sensors connected to analog inputs A0 and A1.
- A water level sensor connected to A2, used to prevent pump activation when the water level in the tank is low.
- Three 12V LEDs representing the water pumps, providing a visual indication of their operating status.
- Three relays connected to digital outputs D2, D3, and D6, used to automatically activate the simulated pumps.
- The control code was uploaded through Arduino IDE, allowing for seamless integration
Between the simulation environment and the programmed logic.

Through this setup, the automatic irrigation simulation is effectively executed, enabling real-time observation and validation of the system's behavior.

III.3.3 Control algorithm overview

The control logic is designed to maintain optimal soil moisture levels without human intervention. It follows a simple yet effective decision-making structure:

1. Continuous Monitoring: The Arduino reads analog values from two soil moisture sensors (connected to A0 and A1) at fixed intervals.
2. Threshold Evaluation: If the moisture level falls below a predefined threshold (indicating dry soil), the corresponding relay (connected to D2 or D3) is activated, switching on the water pump (represented by a 12V LED in the simulation).
3. Auto Shut-off: Once the sensor detects moisture above the threshold, the pump is automatically deactivated to prevent over-irrigation.

4. Fail-safe Conditions: The water level in the reservoir is monitored (via the sensor connected to A2) to prevent the tank from running dry.

This algorithm ensures precise, energy-efficient irrigation tailored to real-time soil conditions.

III.3.4 Simulation results and performance analysis

Upon launching the simulation:

- ✓ The system successfully detected changes in soil moisture levels in real-time. The relay-controlled pumps (simulated via 12V LEDs) responded immediately to sensor input, confirming the effectiveness of the control logic.
- ✓ The use of Proteus allowed observation of voltage levels, relay switching behavior, and LED status—providing detailed insight into the operation of each subsystem.
- ✓ The system maintained a stable and responsive feedback loop, demonstrating reliability under varying simulated soil conditions.
- ✓ The autonomous nature of the system significantly reduces human intervention, aligning with the principles of smart agriculture and sustainable resource management.

III.4 Second simulation experiment: semi-automated control via SMS using GSM module

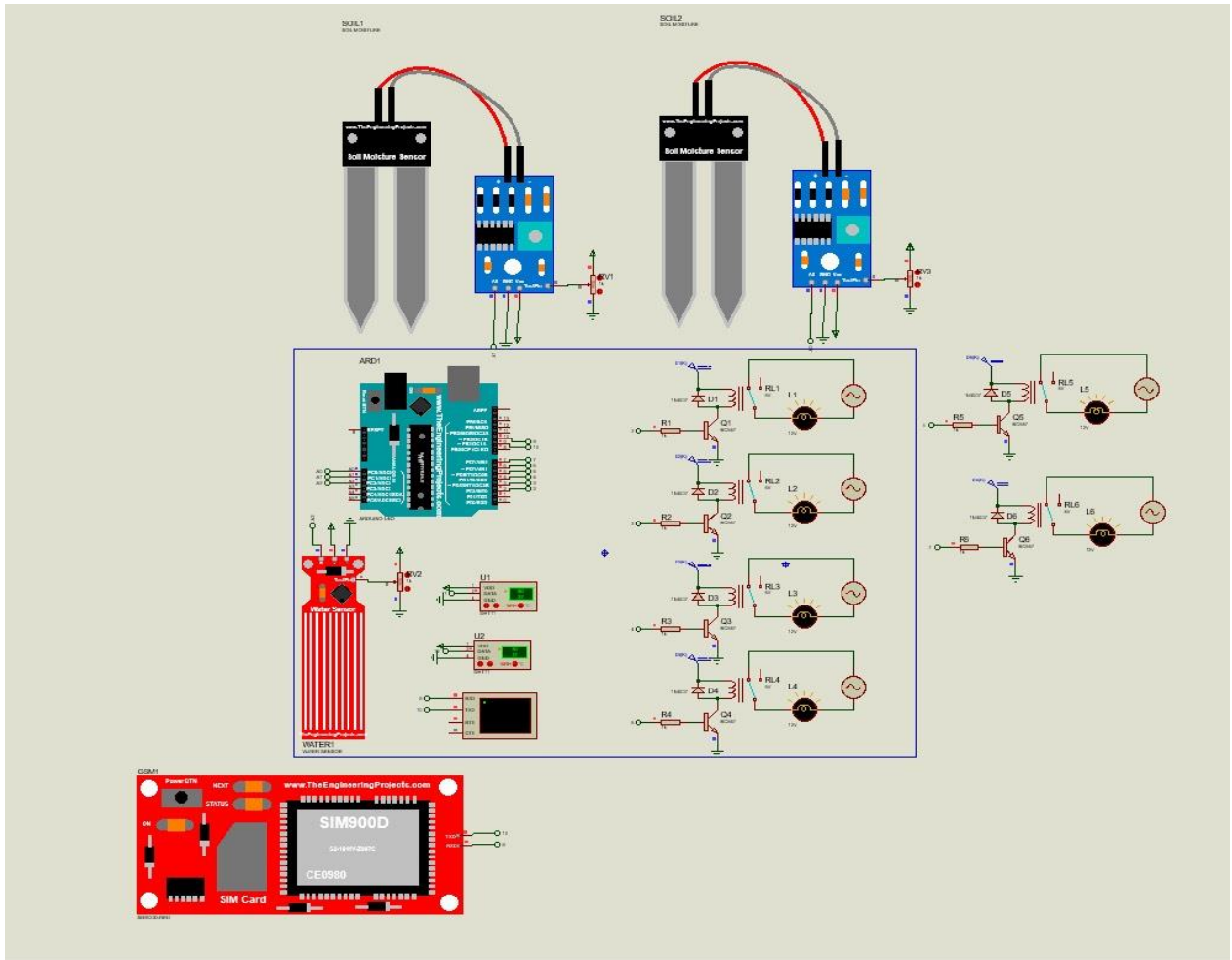


Figure III.3 Semi-Automatic System Simulation Using Proteus

III.4.1 Experiment objectives

This experiment aims to test and evaluate the semi-automatic control mode of the smart agricultural system through the transmission and reception of SMS commands using the SIM800L GSM communication module. The key objectives include:

- ✓ Developing and implementing a set of control commands that can be sent from a mobile phone to the system via SMS.
- ✓ Enabling the user to remotely monitor the real-time status of various system components, including sensors and actuators.
- ✓ Verifying the Arduino controller's ability to correctly interpret incoming commands and analyzing the accuracy and speed of its responses within the simulation environment.
- ✓ Supporting a hybrid control model that combines autonomous operation (automatic mode) with human flexibility (semi-automatic mode).

III.4.2 sms command setup and interaction

A predefined set of SMS commands has been adopted to enable manual control of the system's key components remotely. These commands are intentionally designed to be simple, clear, and effective, particularly in areas with limited digital infrastructure. The supported commands are as follows:

General Information Command:

- Inf

Upon receiving this command, the system sends a real-time status report including:

- Temperature and humidity values for each zone of the farm.
- Water level in the reservoir.
- Current status of all pumps (ON/OFF).
- Current status of all lights (ON/OFF).

Manual Pump Control Commands:

- on1 / off1: Turn ON/OFF the pump for Zone 1.
- on2 / off2: Turn ON/OFF the pump for Zone 2.
- on5 / off5: Turn ON/OFF the pump transferring water from the well to the reservoir.

Manual Lighting Control Commands:

- on3 / off3: Turn ON/OFF the light in Zone 1.
- on4 / off4: Turn ON/OFF the light in Zone 2.

Each message is processed via the SIM800L module, and the content is interpreted by the Arduino controller, which immediately activates or deactivates the corresponding components. A confirmation message is then sent back to the user to acknowledge the execution of the command.

III.4.3 Response analysis in simulation environment

The exchange of SMS commands was simulated using the Virtual Terminal in Proteus. The key outcomes of this experiment are:

- The SIM800L module responded reliably to the user's SMS commands.
- All commands were accurately interpreted by the Arduino, and the corresponding components were triggered as intended.
- Feedback messages were sent back to the user with high precision, reinforcing system transparency and trustworthy control.
- The simulation demonstrated that this control model offers high flexibility, particularly useful in emergency situations or where internet connectivity is unavailable.

Section tow: practical implementation in a laboratory-scal model farm

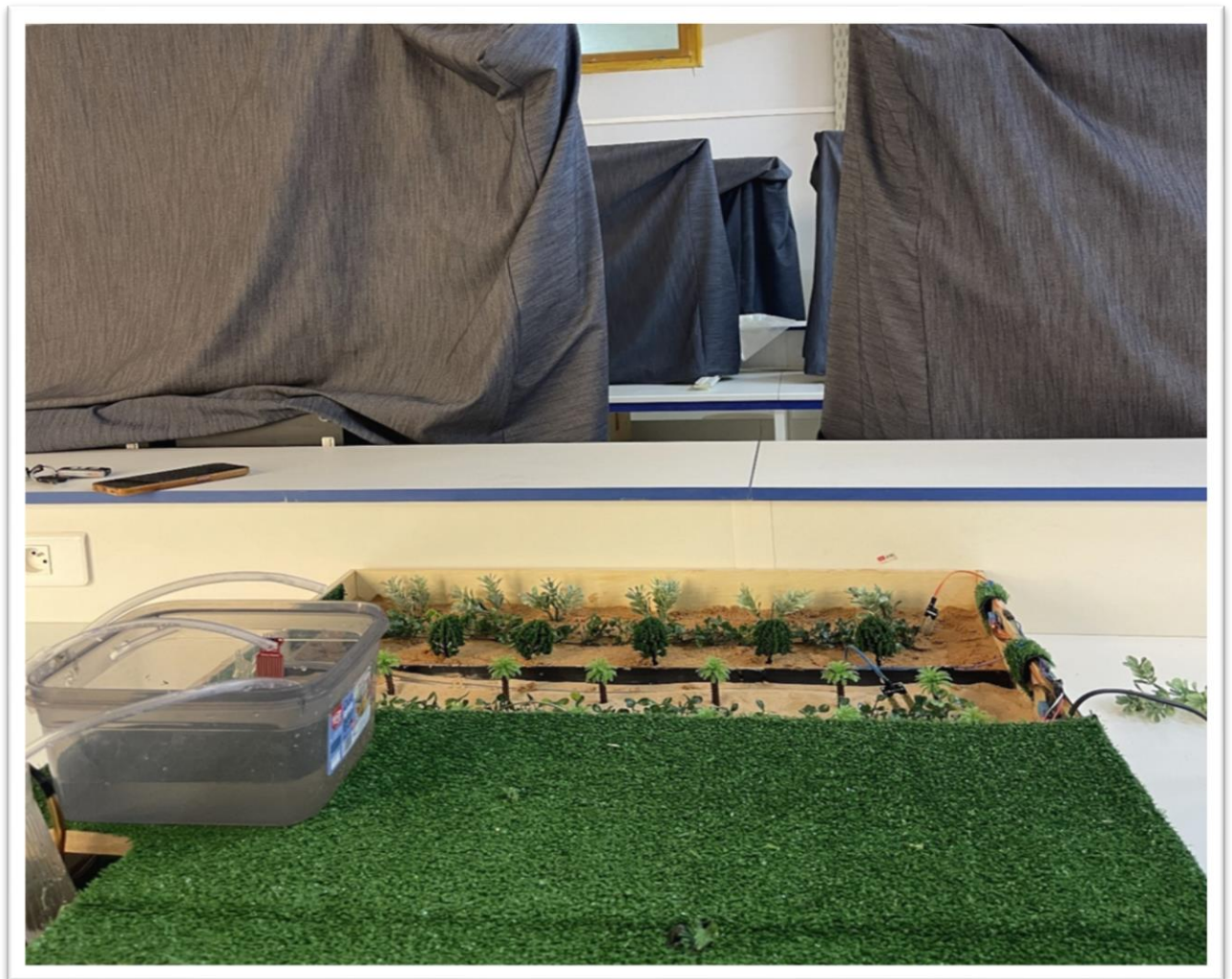


Figure III.4 Practical Implementation of the Prototype System

III.5 Laboratory model farm: design and real-world reference

In this work, a smart model farm was designed in a laboratory environment. This design was inspired by a case study of a real farm belonging to the COSIDAR agricultural group.

The aim of linking the laboratory model with the real farm is to demonstrate the potential for transferring intelligent control and monitoring technologies to large-scale agricultural environments.

III.5.1 location and setup within the laboratory

The smart farm prototype was developed and installed within a controlled laboratory environment, aiming to simulate real agricultural conditions on a smaller scale. This laboratory setup includes key components such as soil moisture sensors, temperature sensors, water level

detectors, relay modules, a control unit based on Arduino, and a SIM800L module for remote communication and monitoring. This configuration enables the testing of irrigation control scenarios, environmental monitoring, and energy management under conditions that closely resemble those of an actual farm.

III.5.2 Reference real farm: location and energy source

The reference agricultural farm “COSIDAR Agriculture” is located in the El-Hadjira area within the El-Gueddachi perimeter, in Touggourt Province, near National Road No. 3. It is one of the agricultural projects developed by the COSIDAR Agriculture Group under the framework of land reclamation and development in the El-Gueddachi zone. The farm covers a total area of 4022 hectares, of which approximately half (around 2000 hectares) has been developed and is currently operational in the southern section, while the northern section is still under development. The irrigation system is based on center pivot irrigation, with a total of 43 pivots, of which 23 are currently operational. Water is supplied to the irrigation network through a single main pump with a power rating of 100 HP (approximately 75 kW) and a starting current of 130 A, responsible for feeding all the active pivots. Electrically, the farm is supplied through a medium-voltage line (36 kV) exceeding 200 km in length, connected to a transformer station with a capacity of 30 MVA. Each pivot is equipped with an individual transformer rated at 160 kVA, amounting to a total installed capacity of approximately 5000 kVA.

To ensure protection and safety, electrical breakers are used, including a main breaker with a breaking capacity (I_k) of 16 kA and a nominal current (I_n) of 630 A, along with secondary breakers rated at 250 A. Operationally, the main pump consumes around 75 kW, which translates to about 1.8 MWh per day if operated continuously for 24 hours. The administrative facilities include offices, computing equipment, surveillance camera systems, and communication devices, with an estimated daily energy consumption of approximately 150 kWh.

The farm currently focuses on wheat cultivation as a primary crop, aligned with national food security strategies. This makes it an ideal case study for evaluating the implementation of automated control systems.

Based on this real-world agricultural site, we saw the opportunity to apply both fully and semi-automated control technologies. Thus, we developed a smart prototype farm, which can serve as a scalable model to be applied in the actual site, aiming to optimize performance, reduce energy consumption, and ensure equitable water distribution.

The following images provide a visual representation of the farm:



Figure III.5 Image Showing Wheat Production in Farm



Figure III.6 Image of the COSIDER Farm Surroundings



Figure III.7 COSIDER Agricultural Investment Site



Figure III.8 Front View of the COSIDER Farm

III.5.3 Physical and environmental characteristics of the model farm

Physical Characteristics:

1. Miniature agricultural plots representing real farmland.
2. Use of soil similar to the original type (sandy, clayey, etc.).

3. A water tank representing the main irrigation source.
4. Drip or sprinkler irrigation system to simulate pivot irrigation.
5. Infrastructure including pump, valves, pipes, and electrical wiring.

Environmental Characteristics:

1. Soil moisture monitoring using sensors.
2. Temperature control to simulate desert climate conditions.
3. Use of LED lighting to mimic sunlight.
4. Simulated airflow or wind using fans.
5. Cultivation of real small-scale crops (such as wheat or beans) or similar plants.

III.6 components of the system implemented in the model farm

The intelligent system implemented in the model farm consists of an integrated set of technological components that interact to achieve automation, monitoring, and control functions. The Arduino microcontroller serves as the core of the system, receiving signals from distributed environmental sensors, analyzing the data based on predefined algorithms, and executing real-time decisions. These actions include activating or deactivating the water pump, sending alerts via the GSM module, and triggering visual indicators such as LED lights. The system is equipped with pipelines and relay modules to carry out irrigation commands, while the energy supply is ensured through either the electrical grid or solar panels, providing operational continuity and autonomy

III.6.1 Hardware components (Arduino, sensors, GSM, solar panels)

- **Arduino Uno:** An open-source microcontroller board based on the ATmega328P processor. It was selected for its programming flexibility and compatibility with a wide range of sensors and external modules.
- **Sensors:** Soil moisture and temperature sensors are employed to monitor environmental parameters, enabling the system to make informed irrigation decisions.
- **SIM800L GSM Module:** A communication module that operates over the GSM network, allowing the system to transmit data and alerts to the user and receive remote control commands.



Figure III.9 GSM Module SIM800L

- **Solar Panels:** Used as an alternative and eco-friendly energy source to ensure the system's autonomy in the absence of conventional electricity

III.6.2 Wiring and integration diagram

The wiring and integration diagram illustrates the complete architecture of the intelligent agricultural system. The Arduino board serves as the central control unit, receiving data from various environmental sensors, including two soil moisture sensors, two temperature and humidity sensors (DHT type), and a water level sensor in the reservoir. These data are processed in real-time using embedded algorithms, enabling automated decision-making.

Five relays are employed to control the system's actuators: two are assigned to irrigation pumps for each side of the field, a third operates a pump that transfers water from the well to the reservoir, and the remaining two manage LED indicators.

The SIM800L GSM module provides remote communication capabilities, allowing the system to send alerts and receive control commands via SMS.

The system is powered by a dual-source setup, either through the electrical grid or a solar panel connected to a charge controller and a rechargeable battery, ensuring continuous operation in remote agricultural environments

Diagram :

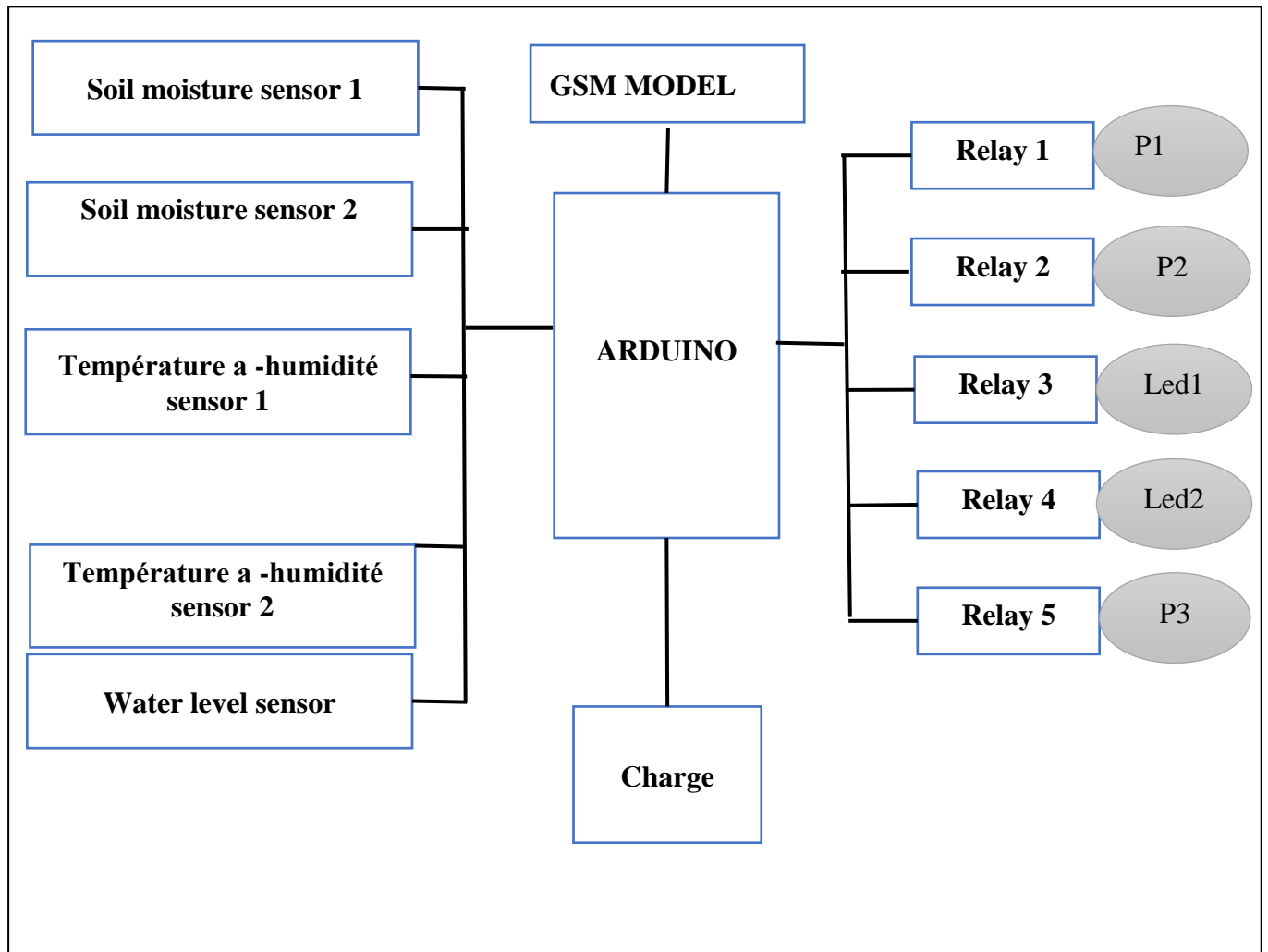


Figure III.10 Diagram Illustrating the Integration of Arduino, SIM800L, and Sensors

III.6.3 system configuration for laboratory environment

The intelligent agricultural control system was assembled and configured in a laboratory environment to verify the correctness of the electronic connections and test the functionality of both hardware and software components before field deployment. The Arduino microcontroller unit was connected to all sensors and actuators according to the following configuration:

- A0: Soil Moisture Sensor 1
- A1: Soil Moisture Sensor 2
- A2: Water Level Sensor (tank)
- D11: DHT11 Sensor 1 (Temperature and Humidity)
- D12: DHT11 Sensor 2 (Temperature and Humidity)

The relay outputs were connected as follows to control external devices:

- D2 → Relay 1: Connected to a water pump for irrigation (side 1)

- D3 → Relay 2: Connected to a second water pump for irrigation (side 2)
- D4 → Relay 3: Connected to an LED indicator
- D5 → Relay 4: Connected to another LED indicator
- D6 → Relay 5: Connected to a pump that transfers water from the well to the tank
- D7 → Relay 6: Reserved for future expansions or additional control

The SIM800L GSM module was integrated with the Arduino through serial communication using:

- D9: TX pin of SIM800L (receives data from Arduino)
- D10: RX pin of SIM800L (sends data to Arduino)

The power supply for the system was provided through the electrical grid in the laboratory setting. However, the design supports alternative power sources, such as a solar energy system consisting of a photovoltaic panel, charge controller, and rechargeable battery, for deployment in off-grid agricultural fields.

Here is image of the system wiring configuration :

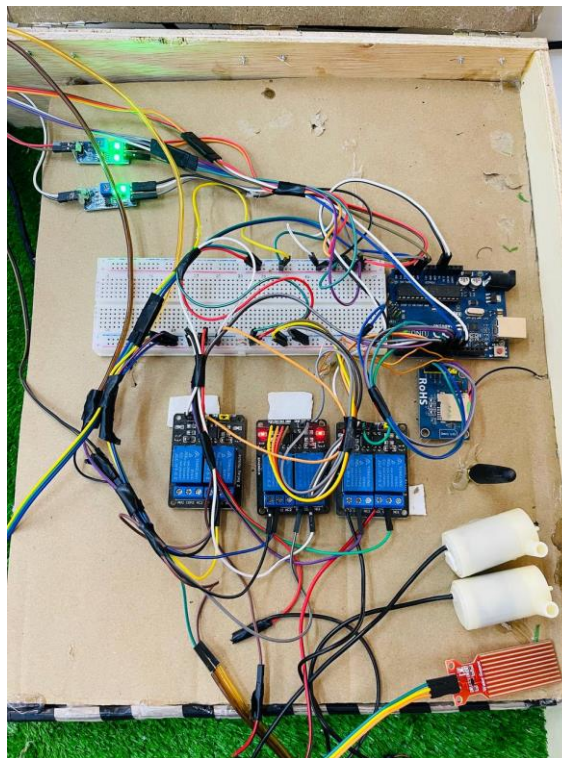




Figure III.11 Real Image of the System Wiring and Connections

Once the components were securely connected and verified, the control code was uploaded to the Arduino board using the Arduino IDE. The program includes logic to read sensor data, evaluate environmental conditions, and trigger actuators accordingly.

Control Code Used:

```
#include <SoftwareSerial.h>
#include <SimpleDHT.h>
SoftwareSerial gsm(9, 10); // RX, TX for SIM800L
const int relayPins[6] = {2, 3, 4, 5, 6, 7};
const int simResetPin = 8;
const int humiditySensor1 = A0;
const int humiditySensor2 = A1;
const int waterLevelSensor = A2;
const int dhtPin1 = 11;
const int dhtPin2 = 12;
SimpleDHT11 dht11_1(dhtPin1);
SimpleDHT11 dht11_2(dhtPin2);
String smsContent = "";
unsigned long lastSensorRead = 0;
const unsigned long sensorInterval = 1000; // 1 second
//  Mode variable
String mode = "man"; // Start in manual mode
void setup() {
  Serial.begin(9600);
  gsm.begin(9600);
  delay(1000);
  pinMode(simResetPin, OUTPUT);
  digitalWrite(simResetPin, HIGH);
  for (int i = 0; i < 6; i++) {
    pinMode(relayPins[i], OUTPUT);
    digitalWrite(relayPins[i], HIGH); // Relays OFF by default
  }
  Serial.println(" # Initializing SIM800L...");
  if (checkGSMConnection()) {
    Serial.println("✔SIM800L is responding.");
    sendSMS("+213660108024", "✔GSM Ready - Mode: MANUAL");
  }
}
```

```
} else {  
  Serial.println("✘ SIM800L not responding. Attempting reset...");  
  resetSIM800L();  
  if (checkGSMConnection()) {  
    Serial.println("✔ SIM800L recovered after reset.");  
    sendSMS("+213660108024", "🔄 SIM800L Reset - GSM Ready.");  
  } else {  
    Serial.println("✘ SIM800L still not responding after reset.");  
  }  
}  
gsm.println("AT+CMGF=1");  
delay(500);  
gsm.println("AT+CNMI=1,2,0,0,0");  
delay(500);  
}  
void loop() {  
  if (gsm.available()) {  
    char c = gsm.read();  
    smsContent += c;  
    if (c == '\n') {  
      smsContent.trim();  
      if (smsContent.length() > 0) {  
        Serial.println("✉ Received SMS: " + smsContent);  
        processSMS(smsContent);  
        smsContent = "";  
      }  
    }  
  }  
  if (millis() - lastSensorRead >= sensorInterval) {  
    lastSensorRead = millis();  
    int hum1_raw = analogRead(humiditySensor1);  
    int hum2_raw = analogRead(humiditySensor2);  
    int waterLevel_raw = analogRead(waterLevelSensor);
```

```
int hum1_percent = map(hum1_raw, 1023, 0, 0, 100);
int hum2_percent = map(hum2_raw, 1023, 0, 0, 100);
int waterLevel_percent = map(waterLevel_raw, 0, 1023, 0, 100);
Serial.print(" 🌿 Humidity 1: "); Serial.print(hum1_percent);
Serial.print(" % | 🌿 Humidity 2: "); Serial.print(hum2_percent);
Serial.print(" % | 🍷 Water Level: "); Serial.println(waterLevel_percent);
//  Auto Mode Control for Relays 1-3
if (mode == "auto") {
  bool changed = false;
  if (hum1_percent < 50 && digitalRead(relayPins[0]) == HIGH) {
    digitalWrite(relayPins[0], LOW);
    changed = true;
  } else if (hum1_percent >= 50 && digitalRead(relayPins[0]) == LOW) {
    digitalWrite(relayPins[0], HIGH);
    changed = true;
  }
  if (hum2_percent < 50 && digitalRead(relayPins[1]) == HIGH) {
    digitalWrite(relayPins[1], LOW);
    changed = true;
  } else if (hum2_percent >= 50 && digitalRead(relayPins[1]) == LOW) {
    digitalWrite(relayPins[1], HIGH);
    changed = true;
  }
  if (waterLevel_percent < 50 && digitalRead(relayPins[2]) == HIGH) {
    digitalWrite(relayPins[2], LOW);
    changed = true;
  } else if (waterLevel_percent >= 50 && digitalRead(relayPins[2]) == LOW) {
    digitalWrite(relayPins[2], HIGH);
    changed = true;
  }
  if (changed) {
    sendSystemStatus("+213660108024"); // Send status after relay state change
  }
}
```

```
}  
// DHT readings  
byte temp1 = 0, humDHT1 = 0;  
if (dht11_1.read(&temp1, &humDHT1, NULL) == SimpleDHTErrSuccess) {  
    Serial.print(" □ DHT11-1: ");  
    Serial.print((int)temp1); Serial.print(" °C, ");  
    Serial.print((int)humDHT1); Serial.println(" %H");  
}  
byte temp2 = 0, humDHT2 = 0;  
if (dht11_2.read(&temp2, &humDHT2, NULL) == SimpleDHTErrSuccess) {  
    Serial.print(" □ DHT11-2: ");  
    Serial.print((int)temp2); Serial.print(" °C, ");  
    Serial.print((int)humDHT2); Serial.println(" %H");  
}  
Serial.print(" # Relay States: ");  
for (int i = 0; i < 6; i++) {  
    Serial.print("R"); Serial.print(i+1);  
    Serial.print(":"); Serial.print(digitalRead(relayPins[i]) == LOW ? "ON" : "OFF");  
    if (i < 5) Serial.print(" | ");  
}  
Serial.println();  
Serial.println("=====");  
}  
}  
void processSMS(String msg) {  
    msg.toLowerCase();  
    if (msg == "auto") {  
        mode = "auto";  
        sendSMS("+213660108024", " Mode changed to AUTO");  
        return;  
    }  
    if (msg == "man") {  
        mode = "man";  
    }  
}
```

```
    sendSMS("+213660108024", " Mode changed to MANUAL");
    return;
}
if (msg == "inf") {
    sendSystemStatus("+213660108024");
    return;
}
for (int i = 1; i <= 6; i++) {
    if (msg == "on" + String(i)) {
        if (mode == "man" || i >= 4) {
            digitalWrite(relayPins[i - 1], LOW);
            sendSystemStatus("+213660108024");
        } else {
            sendSMS("+213660108024", "Δ□ Relay " + String(i) + " is in AUTO mode.");
        }
        return;
    } else if (msg == "off" + String(i)) {
        if (mode == "man" || i >= 4) {
            digitalWrite(relayPins[i - 1], HIGH);
            sendSystemStatus("+213660108024");
        } else {
            sendSMS("+213660108024", "Δ□ Relay " + String(i) + " is in AUTO mode.");
        }
        return;
    }
}
Serial.println("Δ□ Unknown command: " + msg);
}

void sendSystemStatus(String number) {
    int hum1 = map(analogRead(humiditySensor1), 1023, 0, 0, 100);
    int hum2 = map(analogRead(humiditySensor2), 1023, 0, 0, 100);
    int water = map(analogRead(waterLevelSensor), 0, 1023, 0, 100);
    byte t1 = 0, h1 = 0, t2 = 0, h2 = 0;
```

```
String dht1 = "Err", dht2 = "Err";
if (dht11_1.read(&t1, &h1, NULL) == SimpleDHTErrSuccess) dht1 = String(t1) + "C " +
String(h1) + "%";
if (dht11_2.read(&t2, &h2, NULL) == SimpleDHTErrSuccess) dht2 = String(t2) + "C " +
String(h2) + "%";
String r = "";
for (int i = 0; i < 6; i++) r += "R" + String(i+1) + ":" + (digitalRead(relayPins[i]) == LOW ?
"ON " : "OFF ");
String modeStr = mode; // Create a copy of the mode string
modeStr.toUpperCase(); // Convert it to uppercase
String msg = "Mode: " + modeStr + "\nRelays: " + r + "\nH1:" + hum1 + "% H2:" + hum2 +
"% WL:" + water + "%\nT/H1:" + dht1 + "\nT/H2:" + dht2;
sendSMS(number, msg);
}
bool sendSMS(String number, String message) {
gsm.println("AT+CMGF=1");
delay(500);
gsm.print("AT+CMGS=|"); gsm.print(number); gsm.println("|");
delay(500);
gsm.print(message);
delay(500);
gsm.write(26); // CTRL+Z
delay(5000);
return gsm.find("+CMGS");
}
bool checkGSMConnection() {
gsm.println("AT");
delay(1000);
return gsm.find("OK");
}
void resetSIM800L() {
digitalWrite(simResetPin, LOW);
delay(2000);
digitalWrite(simResetPin, HIGH);
```

```
delay(5000);  
}
```

After uploading the code and powering the Arduino using the selected energy source, the system began functioning accurately within the laboratory environment, responding promptly to real-time input values from the sensors, thereby validating its readiness for deployment in real agricultural conditions

III.7 First Practical Experiment: Automated Irrigation Control Based on Soil Moisture

This experiment aims to evaluate the effectiveness of the system in performing automated irrigation based on soil moisture readings obtained from two dedicated sensors

III.7.1 Implementation Steps in the Model Farm

The system was calibrated to acquire real-time data from two soil moisture sensors deployed in distinct zones of the prototype farm. Based on the sensor readings, the controller autonomously determines whether to initiate or terminate the operation of the water pumps assigned to each zone. The control logic is based on a threshold mechanism, wherein a predefined soil moisture value (e.g., less than 400 analog units) is interpreted as an indication of soil dryness. When the moisture level in a given zone falls below this threshold, the Arduino microcontroller issues a digital output signal to the corresponding relay module, thereby activating the associated water pump. Once the moisture level rises above the threshold, the pump is deactivated automatically to prevent over-irrigation.

In this experimental setup:

- Soil Moisture Sensor 1 monitored the moisture content in Zone 1 and controlled Relay 1, which actuated the first irrigation pump.
- Soil Moisture Sensor 2 monitored Zone 2 and was connected to Relay 2, which operated the second irrigation pump.



Figure III.12 Image of the Prototype Farm Without Network Controller

III.7.2 Measurement Accuracy and System Response

The efficiency of the system was evaluated in terms of soil moisture measurement accuracy and the responsiveness of the control unit to sensor input. The results demonstrated reliable and consistent performance, making the system suitable for smart irrigation applications.

Recorded Observations:

- Sensors provided stable and repeatable readings under identical environmental conditions.
- The response time between detecting dryness and activating the pump ranged between 1 and 2 seconds.
- No errors were observed in activating or deactivating irrigation units during the test. The control algorithm performed accurately in accordance with the programmed conditions.
- System components functioned cohesively, with no data loss or connection failures.

III.7.3 System Behavior under Variable Environmental Conditions

The system was tested in two distinct environmental phases to ensure its reliability under real operating conditions:

- Phase One took place in a laboratory environment characterized by stable temperature and humidity conditions, ideal for sensor calibration and testing the accuracy of wiring and programming.

- Phase Two was conducted in a hot environment simulating actual agricultural conditions, where environmental parameters fluctuate rapidly and unpredictably, including daytime temperature spikes, humidity variations, and occasional power supply voltage drops. Observed Results:
- In the laboratory environment, the system demonstrated precise and stable performance, with each unit responding within the specified time frame and without errors.
- In the hot environment, despite the challenges, the system exhibited a high degree of adaptability;
- Sensors continuously provided accurate data, and the Arduino unit recorded no faults or processing freezes.
- Relays and pumps responded smoothly even with temperature fluctuations, indicating effective electrical circuitry and proper insulation.
- The decision-making mechanism remained unaffected by changing inputs, thanks to threshold-based programming and averaging analysis.
- This dual environmental testing reinforces user confidence in the system’s capability to operate in harsh agricultural settings without compromising its stability or operational accuracy

III.8 Second Practical Experiment: Control via SMS Commands Using GSM

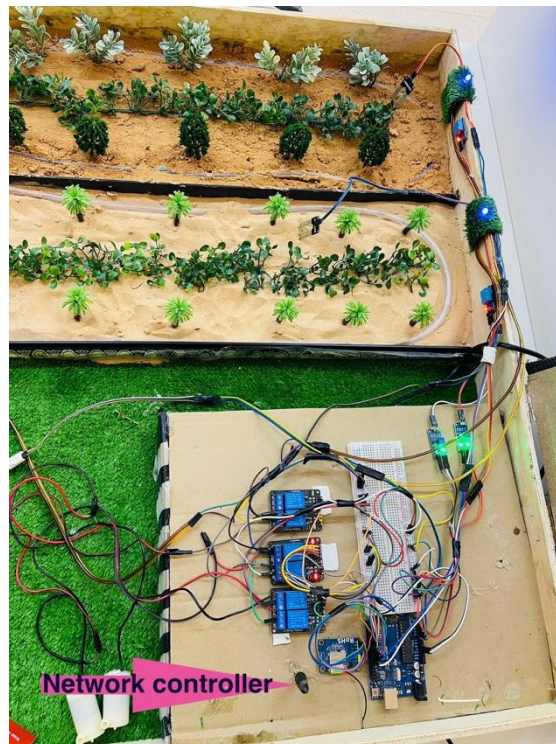


Figure III.13 Image of the Prototype Farm With Network Controller

III.8.1 Sending Commands via SMS to GSM Module

In this system, devices such as pumps or valves are controlled remotely via SMS commands sent to the SIM800L GSM module. The received messages are parsed in the processSMS() function, and commands are executed based on their content and the current operational mode (manual or automatic).

1. Manual Mode Identification

The system is initialized in manual mode by default:

```
String mode = "man"; // Start in manual mode
```

Users can switch to manual mode by sending the SMS command "man":

```
if (msg == "man") {  
    mode = "man";  
    sendSMS("+213660108024", " Mode changed to MANUAL");  
    return;  
}
```

2. Relay Control Commands in Manual Mode

While in manual mode (mode == "man"), users can directly control any of the six relays by sending SMS commands such as:

- "on1" to turne Relay 1 ON
- "off3" to turne Relay 3 OFF
- And so forth for relays 1 to 6

This is handled in processSMS() as follows:

```
for (int i = 1; i <= 6; i++) {  
    if (msg == "on" + String(i)) {  
        if (mode == "man" || i >= 4) {  
            digitalWrite(relayPins[i - 1], LOW); // Activate relay  
            sendSystemStatus("+213660108024");  
        } else {  
            sendSMS("+213660108024", "⚠️ Relay " + String(i) + " is in AUTO mode.");  
        }  
    }  
    return;  
} else if (msg == "off" + String(i)) {  
    if (mode == "man" || i >= 4) {
```

```
    digitalWrite(relayPins[i - 1], HIGH); // Deactivate relay
    sendSystemStatus("+213660108024");
} else {
    sendSMS("+213660108024", "⚠️ Relay " + String(i) + " is in AUTO mode.");
}
return;
}
}
```

Scientific Explanation of the Process:

- The incoming command string is converted to lowercase for consistency.
- For example, when the command "on2" is received, the system verifies that it is currently in manual mode.
- If confirmed, the system activates the corresponding relay by setting its control pin to LOW (active LOW logic).
- After execution, a confirmation SMS with the updated system status is sent back to the user via the sendSystemStatus() function.

Summary

In manual mode:

- Users send text commands like "on1" or "off2" to the SIM800L GSM module.
- The system validates the mode, then switches the specified relay ON or OFF.
- A confirmation message reporting the updated system status is sent back to the user.

This mechanism enables precise and secure remote control of agricultural devices without requiring the user to be physically present on site.

III.8.2 System Response within the Model Farm

The system response within the farm environment refers to the effectiveness and reliability of the control system in reacting to real-time inputs and environmental changes in the agricultural field.

This response includes:

- **Sensor Feedback:** Sensors (soil moisture, temperature, water level) continuously monitor environmental parameters and provide accurate data despite field conditions such as temperature fluctuations, humidity variations, and electrical noise.

- **Command Execution:** Upon receiving commands—either automatically based on sensor thresholds or manually via SMS—the system promptly activates or deactivates actuators such as pumps and valves through relays.
- **Robustness to Environmental Stress:** The system maintains stable operation under harsh outdoor conditions (e.g., high temperatures and power fluctuations), ensuring no interruption or delay in response.
- **Real-time Status Updates:** The system regularly communicates its current status via SMS, informing the operator of sensor readings and actuator states to enable timely decision-making.

This comprehensive response capability ensures efficient water and resource management in the farm, optimizing irrigation and protecting crops under varying conditions.

III.8.3 Response Time and Accuracy Analysis

When sending text commands to the GSM module, the system's response time was observed to be approximately five seconds. This time includes receiving the message, processing it within the Arduino microcontroller, executing the requested command (such as turning a relay ON or OFF), and sending a confirmation reply via SMS.

- **Response Time (5 seconds):**

This duration is acceptable for remote agricultural control systems, as operations like pump activation or irrigation do not require instantaneous response times measured in fractions of a second. The delay arises from several factors, including:

- The GSM module's time to detect and receive the message.
- Processing the message by the microcontroller.
- Mechanical execution of the relay switching.
- Sending confirmation or status update messages.

- **Response Accuracy:**

The system demonstrates good accuracy, executing commands correctly without errors or additional delays. This reflects the effectiveness of the programming and communication setup. The system also verifies the current system state before executing commands, minimizing the risk of conflicting or incorrect instructions.

III.9 Challenges and Lessons Learned

1. Semi-Automatic Approach (Without GSM):

Advantages:

- Simpler in terms of design and coding.

- Fast response time due to local decision-making without communication delays.
- Suitable for small-scale setups or when the user is physically present on-site.

Challenges:

- Requires manual intervention for monitoring and activating components.
- Lacks remote control and automated alert functionalities.
- Not ideal in emergency scenarios or when the user is absent from the field.

Lessons Learned:

- Practical for educational or prototyping purposes.
- Easier maintenance but with limited functionality.

2. Fully Automatic Approach (with GSM Module):

Advantages:

- Enables remote control and monitoring via SMS, ideal for large or remote farms.
- Sends real-time alerts with environmental data (humidity, temperature, water level, etc.).
- Allows switching between manual and automatic modes remotely.

Challenges:

- Delay in response time (approximately 5 seconds or more).
- Dependent on mobile network availability and signal strength.
- Requires more complex programming and incurs additional power and communication costs.

Lessons Learned:

- Power consumption must be optimized, and backup systems should be implemented for signal loss.
- More reliable when powered by stable solar energy and under good GSM coverage.
- Better suited for real-world agricultural applications.

Comparison

Table III.1 Table Comparing Automatic and Semi-Automatic Control Modes

Criterion	Semi-Automatic System	Fully Automatic System (GSM)
Remote Control	Not Supported	Supported via SMS
Reliability	Adequate for local use	High for remote operatio
Response Time	Immediate	5 seconds delay
Programming Complexity	Simple	Moderate to Complex
Cost & Power	Low	Higher (GSM + Power Supply)

III.10 Conclusion

In this chapter, we conducted a series of experiments combining simulation and practical implementation to analyze the fundamental differences between fully automated and semi-automated control systems. The results demonstrated that the absence of a GSAM module or SIM800L chip in the fully automated system significantly limits its functionality, as it operates independently without the capability to receive commands or transmit data remotely. In contrast, the semi-automated system allowed for manual command transmission and remote device control, offering a satisfactory level of operational flexibility.

These experiments underscore the critical role of communication infrastructure in the development of intelligent control systems and pave the way for integrating more advanced technologies in the following chapters to ensure greater autonomy and precision in agricultural environments.

General conclusion

General conclusion

In light of the increasing challenges facing the agricultural sector—especially in arid and semi-arid regions such as Algeria—the need for smart and sustainable solutions has become critical to ensuring effective and balanced food security. This dissertation presented the design and implementation of a hybrid (automatic and semi-automatic) control system for a model farm, utilizing the Arduino platform, a variety of environmental sensors, and SMS-based communication technology to enable remote monitoring and control.

The proposed system represents a practical step toward digital transformation in agriculture by reducing water waste, improving irrigation efficiency, and enhancing crop productivity, while also alleviating the daily workload for farmers. Experimental results have demonstrated the system's adaptability to varying agricultural conditions and its ability to provide accurate, real-time data, thus supporting more informed decision-making.

In conclusion, the outcomes of this work highlight the importance of integrating control and communication technologies into smart farming systems and open new avenues for scaling these solutions to larger and more complex agricultural operations. The study also recommends further research and development in areas such as artificial intelligence and the Internet of Things (IoT) to enhance the performance of such systems and to support sustainable, future-ready agriculture.

Bibliographic references

Bibliographic references

- [1] Al-Khatib, M. (2021). The Role of IoT Systems in Smart Agriculture Development. PhD Thesis, University of Damascus, Faculty of Mechanical and Electrical Engineering.
- [2] Zeidan, N. (2019). Design and Implementation of a Smart Irrigation System Using Microcontrollers and Sensors. PhD Thesis, University of Baghdad.
- [3] Abdulrazak, A. (2020). Applications of Artificial Intelligence in Agricultural Automation. PhD Thesis, Cairo University, Faculty of Agriculture.
- [4] Al-Haddad, Y. (2018). Smart Control System Design Using Arduino for Precision Agriculture. PhD Thesis, University of Aleppo.
- [5] Salem, A. (2019). Using Arduino Microcontroller for Low-Cost Smart Agriculture System Design. PhD Thesis, Tishreen University.
- [6] Khalil, N. (2021). Smart Agricultural System Design Using Arduino and Remote Sensing Technologies. PhD Thesis, University of Constantine 1.
- [7] Cheniti, F.Z. (2020). Wireless Communication Infrastructure in Agricultural IoT Systems. PhD Thesis, University of Blida 1.
- [8] Abdellawi, R. (2018). Remote Control System Development Using GSM Technology in Agricultural Environments. PhD Thesis, University of Annaba.
- [9] Hussein, A. (2020). Integration of Solar Energy with Smart Control Systems in Agriculture. PhD Thesis, University of Mosul.
- [10] Bouzid, N. (2021). Design of a Smart Agricultural System Powered by Solar Energy Using Arduino Microcontroller. PhD Thesis, University of Batna 2.
- [11] Omar, S. (2022). Digital Transformation in Agriculture: Challenges and Opportunities. Al-Fikr Al-Jamii Publishing House, Alexandria.
- [12] Al-Farsi, K. (2023). "Sensor Selection Criteria for Precision Agriculture". Journal of Smart Farming, 15(2), 45-60.
- [13] Mahmoud, S. & Hamed, A. (2022). "System Requirements Engineering for IoT-Based Agricultural Solutions". Cairo: Dar Al-Marefa Publishing.

رسالة دكتوراه: "التصميم الأمثل للأنظمة الزراعية الذكية"، إبراهيم خالد، جامعة الملك فهد للبترول والمعادن، 2023

[14]

Abstract:

This thesis aims to design a hybrid intelligent control system (automatic and semi-automatic) for a model farm, based on the Arduino platform and wireless communication technologies using the SIM800L module. The system allows real-time monitoring and analysis of critical environmental parameters such as temperature, soil moisture, and water level, enabling instant decisions to operate or stop the pump and lighting. These operations can be activated either automatically or via SMS, ensuring flexible control both locally and remotely.

The project relies on renewable energy sources, particularly solar energy, making it suitable for use in rural and isolated areas. The system was initially developed through simulation using the Proteus software, then implemented in a laboratory environment.

The results demonstrated the system's effectiveness in reducing water waste, improving operational efficiency, and minimizing human intervention, positioning it as a practical model for smart and sustainable agriculture. The study also recommends expanding the system in the future to include artificial intelligence and the Internet of Things (IoT) for greater autonomy and precision.

Résumé :

Ce mémoire a pour objectif la conception d'un système de contrôle intelligent hybride (automatique et semi-automatique) pour une ferme modèle, basé sur la plateforme Arduino et les technologies de communication sans fil via le module SIM800L. Le système permet la surveillance et l'analyse en temps réel de paramètres environnementaux essentiels tels que la température, l'humidité du sol et le niveau d'eau, afin de prendre des décisions instantanées pour activer ou désactiver la pompe et l'éclairage. Ces opérations peuvent être déclenchées automatiquement ou par SMS, assurant ainsi un contrôle flexible à distance ou sur place.

Le projet repose sur des sources d'énergie renouvelable, en particulier l'énergie solaire, ce qui le rend adapté aux zones rurales et isolées. Le système a été développé initialement par simulation à l'aide du logiciel Proteus, puis mis en œuvre dans un environnement de laboratoire.

Les résultats ont démontré l'efficacité du système dans la réduction du gaspillage d'eau, l'amélioration de l'efficacité opérationnelle et la diminution de l'intervention humaine, faisant de lui un modèle pratique pour une agriculture intelligente et durable. L'étude recommande également d'élargir ce système à l'avenir en y intégrant l'intelligence artificielle et l'Internet des objets (IoT) pour plus d'autonomie et de précision.

ملخص

يهدف هذا العمل إلى تصميم نظام تحكم ذكي هجين (أوتوماتيكي ونصف أوتوماتيكي) لمزرعة نموذجية، يعتمد على منصة أروينو وتقنيات الاتصال اللاسلكي باستخدام وحدة SIM800L. يتيح هذا النظام المراقبة والتحليل في الوقت الحقيقي لمجموعة من المعطيات البيئية الأساسية مثل درجة الحرارة، رطوبة التربة، ومستوى المياه، من أجل اتخاذ قرارات فورية لتشغيل أو إيقاف المضخة والإضاءة. ويمكن تنفيذ هذه العمليات إما تلقائياً أو عن طريق الرسائل النصية، مما يوفر تحكماً مرناً عن بُعد أو محلياً. يعتمد المشروع على مصادر الطاقة المتجددة، وخاصة الطاقة الشمسية، مما يجعله مناسباً للمناطق الريفية والمعزولة. تم تطوير النظام في البداية من خلال المحاكاة باستخدام برنامج Proteus، ثم تم تطبيقه فعلياً في بيئة مخبرية. أظهرت النتائج فعالية النظام في تقليل هدر المياه، تحسين الكفاءة التشغيلية، وتقليل الحاجة للتدخل البشري، مما يجعله نموذجاً عملياً للزراعة الذكية والمستدامة. كما توصي الدراسة بتطوير النظام مستقبلاً من خلال دمج تقنيات الذكاء الاصطناعي وإنترنت الأشياء من أجل تحقيق مزيد من الاستقلالية والدقة.

