

**REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE**

وزارة التعليم العالي و البحث العلمي

**Ministère de l'Enseignement Supérieur et de la Recherche  
Scientifique**

جامعة أبي بكر بلقايد - تلمسان

Université Aboubakr Belkaïd – Tlemcen –

Faculté de TECHNOLOGIE



## **Dissertation**

Presented for the purpose of obtaining the master's degree  
**In:** Telecommunications

**Speciality:** Optical Telecommunications

**By:** TAIBI Oussama and SAHRAOUI Mohammed Yassine

## **Subject**

Real-Time Power Outage Detection and Notification System using Arduino Uno  
and SIM900 Shield

Publicly defended, on 25/ 06 / 2025 . The committee is composed of:

M/ BORSALI Ahmed Riad  
M/ CHIKH-BLED Hicham  
M/KARIM Fethallah

Professor Tlemcen University  
MCA Tlemcen University  
Professor Tlemcen University

Committee Chair  
Committee Member  
Supervisor

University year :2024/2025

# Dedication

“

*To my dear mother, the light of my path and my hope, who has always given me love and support, and whose presence has always been hope, reliance, and endless energy.*

*To my friend and companion, my gift in this life, whose presence gave me hope, support, and energy like no other,  
W.*

*To my father and my dear brother.*

*To my loyal friends Imad, Seifo, Omar, Salim, and Osama, who were with me through the good and bad times.*

*To my wonderful family, especially my uncle, for always being there for me and loving me.*

*And I don't forget my brother Najib, who was my childhood friend and noble competitor. He pushed me to get better.*

”

***I humbly dedicate this work to everyone who believed in me and helped me.***

# Acknowledgement

First off, I truly want to thank **Allah** from the bottom of my heart for giving me the strength, courage, and patience to get this little project done.

And a huge thank you goes to my supervisor, **M. KARIM Fethallah**. Honestly, this project would not have been possible without your solid direction, consistent support, and those super helpful comments.

I also thank **Mr. CHIKH-BLED Hicham** and **Mr. BORSALI Ahmed Riad** for giving us their precious time to provide a valuable reading and evaluation of my thesis.

And finally, a huge, heartfelt thank you to everyone who helped with this work, whether you were directly involved or offered support from the sidelines. Every bit of your contribution, big or small, has been incredibly important to my academic journey and to making this research a success.

May Allah bless you all.

# Contents

- Dedication ..... I**
- Acknowledgement ..... II**
- General Introduction ..... 1**
- I Alarm Systems and Home Automation ..... 4**
  - I.1 General Introduction ..... 4
    - I.1.1 Remote Control Paradigms ..... 5
    - I.1.2 Conceptual Framework of Smart Systems ..... 5
  - I.2 Alarm Systems..... 9
    - I.2.1 Conceptual Framework of Alarm Systems ..... 9
    - I.2.2 Advantages of Alarm Systems..... 11
  - I.3 Connectivity Modalities of Alarm Systems: Wired versus Wireless .....12
    - I.3.1 Wired Alarm Systems .....12
    - I.3.2 Wireless Alarm Systems.....13
    - I.3.3 First: Wired Alarm Systems .....14
    - I.3.4 Second: Wireless Alarm Systems .....15
  - I.4 Societal and Economic Impact of Alarm Systems .....16
    - I.4.1 Benefits for Individuals and Property.....16
    - I.4.2 Enhanced Emergency Response.....17
  - I.5 Domotics (Home Automation) ..... 18
    - I.5.1 Definition of Domotics ..... 18
    - I.5.2 Core Systems Supported by Domotics..... 18
    - I.5.3 Control Methodologies in Domotics .....19
    - I.5.4 Objectives and Benefits of Domotics.....19
    - I.5.5 Key Components of Domotics Systems.....19
    - I.5.6 Real-World Applications of Domotics .....21
  - I.6 Societal Relevance of Alarm and Communication Systems in Daily Life..... 22
    - I.6.1 Benefits for Individuals and Property..... 22
    - I.6.2 Optimized Emergency Response..... 23
  - I.7 Implications and Foundation for Future Work ..... 24
- II Arduino, Modules, and Board ..... 25**
  - II.1 Arduino Platform Overview ..... 25
    - II.1.1 The ATmega328P Microcontroller: Architecture and Memory..... 26
  - II.2 Arduino Input/Output (I/O) Pins and Peripherals..... 26
    - II.2.1 Digital Pins..... 27

# Contents

II.2.2	Special Functions of Digital Pins .....	27
II.2.3	Analog Pins and Analog-to-Digital Conversion.....	28
II.2.4	Onboard Peripherals .....	28
II.3	Arduino Programming Environment (Arduino IDE) .....	29
II.3.1	Features of the Arduino IDE .....	29
II.3.2	Understanding the Arduino IDE Interface .....	29
II.3.3	Program Development and Upload Workflow .....	30
II.3.4	Illustrative Example: LED Blinking Program.....	30
II.4	Arduino Shields: Concept and Functionality .....	30
II.4.1	Definition and Advantages of Arduino Shields.....	31
II.4.2	Common Shield Examples .....	31
II.4.3	Installation and Usage of Arduino Shields .....	31
II.4.4	Important Considerations for Shield Utilization.....	31
II.5	Introduction to the GSM Shield SIM900.....	32
II.5.1	Overview and Key Features of the SIM900 GSM/GPRS Shield.....	32
II.5.2	Technical Specifications of the SIM900 Shield.....	34
II.6	Connecting and Configuring the SIM900 Shield with Arduino .....	34
II.6.1	Physical Setup and Power Management.....	34
II.6.2	Software Communication: Serial Interface and Baud Rate Synchroni- zation .....	35
II.6.3	Key Considerations for Successful SIM900 Communication.....	36
II.7	Chapter Summary.....	37
<b>III</b>	<b>Development of a Power Outage Detection System Using Arduino and SIM900 Module .....</b>	<b>38</b>
III.1	Introduction.....	38
III.2	System Components .....	38
III.2.1	Arduino UNO Control Board .....	39
III.2.2	GSM SIM900 Communication Module.....	39
III.2.3	Overall System Interconnections .....	41
III.2.4	Arduino UNO Pin Configuration .....	42
III.2.5	Power and Ground Pins .....	43
III.2.6	SIM900 GPRS Shield Features .....	43
III.2.7	UART Connection Configuration for SIM900 .....	45
III.3	System Functionality .....	45
III.3.1	General System Concept.....	45
III.3.2	Power Monitoring Process.....	45
III.3.3	SIM900 Automated SMS Notification System .....	46
III.3.4	User Interaction via SMS Commands .....	46
III.4	System Programming and Implementation .....	47
III.4.1	Code Structure and Initialization .....	47
III.4.2	The loop() Function: Continuous Operation .....	50

<b>Contents</b>	
III.5 System Functionality .....	51
III.5.1 General System Concept.....	51
III.5.2 Power Monitoring Process.....	52
III.5.3 SIM900 Automated SMS Notification System .....	52
III.5.4 User Interaction via SMS Commands .....	53
III.6 System Programming and Implementation .....	53
III.6.1 Code Structure and Initialization Phase .....	53
III.6.2 The loop () Function: Continuous System Operation .....	56
III.7 Conclusion .....	58
III.8 Future Enhancements .....	58
<b>General Conclusion and Perspectives.....</b>	<b>60</b>
<b>66</b> .....	<b>ملخص</b>

# List of Figures

- I.1 Sensors in Smart Systems ..... 6
- I.2 Actuators in Smart Systems ..... 7
- I.3 Diverse Applications of Smart Systems ..... 8
- I.4 Overview of Alarm Systems ..... 9
- I.5 Comparison of Wired and Wireless Alarm System Architectures ..... 12
- I.6 Conceptual Diagram of Wireless Alarm Systems ..... 15
- I.7 Conceptual Overview of Domotics (Home Automation) ..... 18
  
- II.1 Initial setup code using SoftwareSerial..... 35
  
- III.1 Sensors in Smart Systems ..... 40
- III.2 Sensors in Smart Systems ..... 40
- III.3 GSM Arduino Working Prototype ..... 41
- III.4 Arduino UNO SIM900 Wiring Diagram ..... 42
- III.5 Arduino GSM Interface Diagram..... 44
- III.6 Code Snippet: SoftwareSerial Initialization for SIM900 Communication ..... 47
- III.7 Code Snippet: SMS Power Status Variables and Logic ..... 48
- III.8 Code Snippet: Arduino GSM Setup Code with AT Commands ..... 49
- III.9 Code Snippet: AT Command Configuration for SMS and Time Synchronization..... 50
- III.10Sensors in Smart Systems ..... 50
- III.11Sensors in Smart Systems..... 51
- III.12Sensors in Smart Systems..... 51
- III.13Code Snippet: SoftwareSerial Initialization for SIM900 Communication ..... 54
- III.14Sensors in Smart Systems..... 54
- III.15Code Snippet: Arduino GSM Setup Configuration with AT Commands..... 55
- III.16Code Snippet: AT Command Configuration for SMS Mode and Network Time Synchronization ..... 56
- III.17Sensors in Smart Systems..... 56
- III.18Sensors in Smart Systems ..... 57
- III.19Sensors in Smart Systems..... 57

# General Introduction

## Contexte

In the current era of rapid technological progress, automation and intelligent security systems have become central to the development of modern smart environments. These technologies aim to enhance safety, optimize energy usage, and offer user-friendly solutions across domestic and commercial settings. Among the most accessible and versatile platforms supporting this innovation is Arduino—a widely adopted open-source microcontroller system that integrates hardware and software to develop intelligent, scalable solutions.

This project aims to design and implement a low-cost, autonomous power monitoring system capable of continuously detecting electrical outages. Operating in the background 24/7, the system is engineered to instantly detect power loss and transmit a real-time notification to the user via GSM-based SMS communication. Such an immediate alert mechanism is particularly valuable in critical environments where electrical stability is paramount and any disruption can lead to safety hazards or operational failure.

By leveraging embedded systems, wireless communication protocols, and automation strategies, the proposed solution demonstrates how microcontroller-based platforms can effectively address practical engineering problems and contribute to enhanced situational awareness in home infrastructure monitoring.

## Problematic

Despite significant progress in the domain of automation and remote monitoring, several technical and operational challenges persist. Chief among them is the dependency on high-quality, diverse datasets required to train and validate intelligent systems, which are often costly and difficult to acquire.

Moreover, real-time monitoring systems deployed in high-risk environments such as home energy management or industrial automation must demonstrate high reliability and fault tolerance. In such contexts, a system failure—whether in hardware, communication, or detection—can result in critical safety compromises and substantial financial loss.

This research addresses these concerns by developing a robust and responsive monitoring system based on Arduino and GSM modules. The system emphasizes immediate notification capabilities to mitigate risks associated with unanticipated power interruptions. Additionally, the work touches on modern approaches such as Bayesian deep learning, which have shown promise in enhancing system reliability and decision-making under

uncertainty—especially when integrated with low-power embedded platforms.

In doing so, the study illustrates the intersection of automation, embedded systems, and security technologies, offering a practical and efficient solution to a common but often overlooked infrastructure vulnerability.

## Contribution

This thesis presents a practical implementation of a power outage detection and alert system tailored for domestic environments. Centered around the Arduino Uno microcontroller, the system integrates readily available hardware components and a GSM shield to create a cost-effective and accessible smart monitoring solution.

The main contributions are outlined as follows:

- **Hardware-Software Integration:** The project demonstrates how to construct a fully functional embedded system using Arduino, focusing on both hardware design and firmware development. This includes an in-depth discussion of the hardware architecture, module selection, and interfacing techniques.
- **Real-Time Notification Mechanism:** A key feature of the system is its ability to detect power outages in real time and instantly notify users via SMS. This functionality is essential in applications where early detection and rapid response to power disruptions are critical.
- **System Deployment and Validation:** The implemented system was tested under real-world conditions to assess reliability and responsiveness. The validation process also evaluated the feasibility of deploying such a system at scale within residential or small office environments.
- **Application in Smart Home Automation:** Beyond outage detection, the design contributes to the broader field of home automation by demonstrating the potential of integrating Arduino-based platforms with wireless communication modules to enhance monitoring and control.

These contributions collectively serve to bridge the gap between theoretical system design and practical application in embedded smart systems.

## Plan of the Thesis

The aim of this study is to explore the design, development, and deployment of a microcontroller-based power outage detection system, within the broader context of home automation and smart alerting mechanisms.

The thesis is structured into the following three chapters:

### 1. Chapter 1: Alarm Systems and Home Automation

This chapter provides a foundational overview of security alarm systems and their evolution into intelligent automation frameworks. It highlights how modern smart homes employ sensor networks, control logic, and communication technologies to ensure safety and energy efficiency.

### **2. Chapter 2: Arduino and Associated Modules**

This section introduces the Arduino platform and its key hardware modules. It details the technical capabilities of the Arduino Uno and discusses how shields and peripheral components can extend its functionality. Programming tools, data acquisition, and serial communication protocols are also explored.

### **3. Chapter 3: Design and Implementation of the Power Monitoring System**

The final chapter presents the proposed system in detail. It explains the system architecture, discusses communication via the SIM900 GSM module, and outlines the software routines used for event detection and message transmission. The chapter concludes with system validation and performance analysis.

# **Chapter I**

## **Alarm Systems and Home Automation**

## I.1 Introduction

Recent decades have witnessed profound and transformative advancements in information and communication technologies (ICT) [9]. This rapid progression has significantly enhanced various facets of human life, particularly contributing to increased safety and improved convenience in daily operations.

A foundational development stemming from these technological advancements was the widespread emergence of remote-control technology [1]. This innovation enabled the management and operation of distant devices, initially via wired connections such as cables and subsequently through wireless transmission methods. The advent of remote control fundamentally simplified complex operations, facilitating the efficient and rapid execution of tasks such as device activation/deactivation and setting adjustments.

The capabilities introduced by remote control paved the way for the development of sophisticated smart systems [2]. Defined as autonomous entities, smart systems are designed to process environmental data acquired through sensors, leverage advanced communication protocols, and execute complex computational algorithms to reason about their surroundings and perform actions with minimal human intervention [14]. The overarching goal of smart systems is to augment automation, thereby reducing the necessity for manual operations. This approach inherently improves user comfort, enhances safety protocols, and optimizes resource utilization [3].

Within the broader domain of smart systems, alarm systems constitute a critical component [4]. These systems are specifically engineered to safeguard lives and property by providing instantaneous alerts regarding anomalous occurrences or potential threats. This chapter will delve into the principles of remote control, the characteristics and components of smart systems, and a detailed exploration of alarm systems, culminating in a practical application with a smart power outage detection system utilizing the Arduino platform and SIM900 communication module [4].

### I.1.1 Remote Control Paradigms

Remote control refers to technology enabling the management and operation of devices from a distant location. This functionality is achieved by transmitting commands through various communication media, including wired methods (e.g., electrical cables) or, more commonly, wireless technologies such as Radio Frequency (RF), Wi-Fi, and Bluetooth [1].

Remote control technology plays a pivotal role across diverse sectors, encompassing industrial processes, domestic applications, and military operations. It empowers users to interact with complex subsystems in a more flexible, secure, and efficient manner [9].

Illustrative examples of remote control applications include:

- **Consumer Electronics:** Television remote controls for channel changing and volume adjustment.
- **Unmanned Aerial Vehicles (UAVs):** Remote piloting of drones for surveillance, delivery, or recreational purposes.
- **Smart Home Systems:** Wireless dimming of lights or adjustment of climate control settings.

Initially conceived as a simple command transmission mechanism, remote control has evolved significantly to facilitate the execution of automated tasks based on sophisticated data analysis. This evolution has led directly to the emergence of smart home remote control systems [3]. In this context, "domotics", or home automation, refers to the integration of technology to automatically manage and control various household functions, such as lighting, heating, ventilation, and security systems [7].

### I.1.2 Conceptual Framework of Smart Systems

Smart systems represent advanced electronic architectures designed to operate autonomously by acquiring and interpreting data from their surrounding environment [2]. Their intelligence is derived from the ability to process sensed information and adapt their behavior accordingly.

#### Key Characteristics of Smart Systems

Smart systems exhibit several defining characteristics that distinguish them from conventional automated systems:

- **Adaptability and Learning:** Smart systems possess the capacity to learn from accumulated data and identified patterns over time. Through machine learning algorithms and continuous data acquisition, these systems can refine their predictive capabilities and decision-making accuracy [14]. For instance, a smartphone can learn user habits regarding application usage or frequented locations to optimize notifications and power consumption.
- **Environmental Responsiveness:** A fundamental attribute of smart systems is their ability to dynamically adapt to changes within their operational environment.

In a smart home context, for example, the system can automatically adjust indoor temperature settings based on real-time outdoor weather conditions, thereby optimizing energy consumption and maintaining occupant comfort [7].

- **Continuous Optimization:** The inherent learning and adaptive mechanisms of smart systems enable them to continuously refine their performance and efficiency over extended periods. As more environmental data is gathered and processed, the system's decision-making processes become increasingly nuanced and effective, leading to ongoing optimization of their designated functions [3].

### Core Components of Smart Systems

Smart systems are constituted by several integrated components, each fulfilling a specific role essential for achieving the system's overall objectives. These typically include:

- **Sensors:** These act as the primary data acquisition units of a smart system, perceiving and collecting information about the physical environment [2, 9]. Common types of sensors employed in smart systems include:
  - **Motion Detectors:** Utilized to ascertain the presence or absence of human activity within a defined area [4].
  - **Temperature Sensors:** Employed for measuring ambient thermal conditions [3].
  - **Light Sensors:** Designed to quantify the intensity of ambient illumination.
  - **Humidity Sensors:** Used for detecting the moisture content in the air.
  - **Pressure Sensors:** To measure force applied over a surface area.

These sensors collectively provide real-time environmental data, which forms the basis for the system's intelligent operations [9].

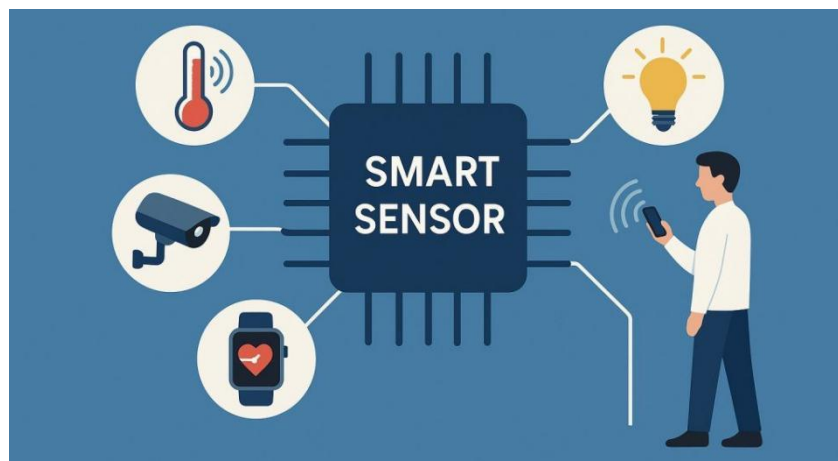


Figure I.1: Sensors in Smart Systems

- **Microcontrollers/Embedded Processors:** Serving as the "brain" of smart systems, these hardware units are responsible for receiving, processing, and interpreting sensor data. They execute programmed algorithms and control logic to make decisions and initiate corresponding actions [4, 9]. Examples include low-cost yet powerful single-board computers like Arduino microcontrollers or Raspberry Pi, which offer significant processing capabilities for embedded applications [4].
- **Actuators:** These are the output components of a smart system, responsible for executing physical actions based on the processed data and decision-making by the microcontroller [14]. For instance, in a smart home system, if a temperature sensor detects a drop below a set threshold, the microcontroller can command an actuator (e.g., a smart thermostat) to activate the heating system. Similarly, if a motion sensor detects presence in a room, an actuator (e.g., a smart light switch) can illuminate the lights [7, 9].



Figure I.2: Actuators in Smart Systems

### Applications and Examples of Smart Systems

Smart systems are increasingly pervasive, integrating into various aspects of modern daily life [2]. Notable applications include:

- **Smartphones:** These devices exemplify smart systems by dynamically adjusting functionalities such as notification delivery, display brightness, and power management based on ambient light conditions, user behavior patterns, and location data





Figure I.4: Overview of Alarm Systems

### I.2.1 Conceptual Framework of Alarm Systems

An alarm system is defined as a synergistic integration of hardware and software components engineered to detect potential threats or unusual anomalies within a monitored environment and subsequently notify designated users or relevant authorities [4]. The primary objective of an alarm system extends beyond mere detection; its fundamental role is to provide timely alerts that enable prompt action, thereby minimizing potential damage or risk associated with incidents such as burglaries, fires, or gas leaks [4].

The operational principle of an alarm system typically follows a structured progression: sensing, detection, alerting, and, in some advanced configurations, automated response [7]. Upon the system's detection of a predefined threat condition, it activates an alert mechanism to notify designated recipients, which may include property owners, security personnel, or emergency services [4].

#### Key Components of Alarm Systems

Alarm systems are composed of several interdependent components that collectively ensure their functionality:

1. **Sensors:** These are specialized devices designed to monitor specific environmental parameters. Common types of sensors found in alarm systems include:
  - **Motion Sensors:** Widely used in security alarms to detect physical movement within a designated area [4].
  - **Heat Detectors/Temperature Sensors:** Employed in fire detection systems to identify abnormal temperature increases indicative of a fire.
  - **Smoke Detectors:** Crucial for fire safety, these initiate alarms upon detecting elevated levels of smoke particles in the air.

- **Gas Detectors:** Specifically designed to identify the presence of hazardous gases, such as carbon monoxide or natural gas, critical in kitchens or industrial settings.
  - **Glass Break Detectors:** These acoustic sensors are often used to protect windows by identifying the sound frequencies associated with breaking glass.
  - **Door/Window Contacts:** Magnetic switches that detect when a door or window is opened.
2. **Control Panel:** Functioning as the central processing unit of the alarm system, the control panel receives signals from all connected sensors. It processes this data against predefined thresholds and rules, and based on its analysis, determines whether to activate an alert or notification system [4]. The control panel may also be configured to trigger automated responses, such as dispatching an emergency response team or making an automated call.
  3. **Alert Mechanisms (Audible and Visual):** These components are responsible for signaling an alarm event. They include:
    - **Siren or Bell:** Generates a loud audible signal to indicate an emergency, deter intruders, and alert occupants or nearby individuals.
    - **Strobe Lights/Flashing Lights:** Provide a visual cue that an alarm has been triggered, particularly useful in noisy environments or for individuals with hearing impairments.
    - **Remote Notifications:** Modern alarm systems offer the capability to transmit alerts via multiple communication channels, including SMS text messages, emails, or push notifications to mobile applications. These notifications ensure that users receive real-time updates on the system's status regardless of their physical location [4].
  4. **Response Systems:** Advanced alarm systems can be directly linked to emergency services or initiate automated pre-programmed actions. Upon detection of a threat, the system can automatically dispatch alerts to emergency responders (e.g., fire department, police) or trigger specific safety measures, such as locking doors, shutting off gas supplies, or deactivating electrical systems to prevent further damage [4].

### Categorization of Alarm Systems by Purpose

Alarm systems can be broadly categorized based on the specific type of threat they are designed to detect:

1. **Burglar Alarm Systems:** These systems are deployed to detect unauthorized entry into a property. They typically integrate various detection technologies, including motion sensors, door/window contacts, and sometimes surveillance cameras [4].

2. **Fire Detection Systems:** Designed to identify the presence of fire, these systems commonly employ smoke detectors (ionization or photoelectric), heat detectors (fixed-temperature or rate-of-rise), and sometimes flame detectors. Their rapid activation is critical for ensuring occupant safety and minimizing property damage in buildings [7].
3. **Gas Leak Detection Systems:** These specialized alarms detect the presence of hazardous gases, such as carbon monoxide, propane, or natural gas, at dangerous concentrations. They are indispensable in residential kitchens, industrial plants, and any areas where gas leaks pose a risk.
4. **Power Failure Alarm Systems:** Specifically engineered to monitor the integrity of the electrical power supply, these systems activate alerts upon detecting an interruption. They are particularly crucial in environments where continuous power is vital, such as medical facilities, data centers, or critical infrastructure, to prevent operational disruptions.
5. **Flood and Water Leak Alarms:** These systems are designed to detect abnormal water levels or leaks, providing early warning to prevent water damage to homes or commercial buildings, especially in areas prone to flooding or where plumbing infrastructure is susceptible to bursts.

### Operational Methodology of Alarm Systems

Alarm systems operate through a multi-layered methodology encompassing detection, processing, activation, and response [4]:

- **Detection Phase:** Sensors continuously monitor their respective environmental parameters, such as movement, temperature fluctuations, or the presence of smoke.
- **Processing Phase:** When a sensor registers data that deviates from normal parameters or indicates an anomaly, this information is transmitted to the central control panel. The control panel then processes this data, comparing it against pre-defined thresholds (e.g., a specific temperature limit for heat sensors, or a detected movement pattern for motion sensors) to determine if a threat condition is met.
- **Activation Phase:** If the processed sensor data correlates with the defined threat characteristics, the control panel initiates the activation of the alert system, triggering audible alarms (e.g., sirens) and/or visual alerts (e.g., flashing lights).
- **Response Phase:** In addition to local alerts, the system can also be configured to deploy an immediate external response. This may involve automatically notifying security services, emergency responders (fire department, police), or initiating pre-programmed safety actions (e.g., remote door locking, utility shutdown).

### I.2.2 Advantages of Alarm Systems

The deployment of alarm systems offers numerous significant advantages across various applications:

- **Enhanced Security for Property and Personnel:** Alarm systems serve as effective deterrents against criminal activity and unauthorized access. In the event of a breach, they provide immediate alerts to designated contacts and authorities, facilitating rapid intervention [4, 7].
- **Early Hazard Detection and Mitigation:** These systems are instrumental in the early identification of hazardous situations such as fires, gas leaks, or floods. This timely detection enables occupants and emergency services to respond promptly, thereby minimizing potential damage, preventing injury, and saving lives [7].
- **Ubiquitous Remote Monitoring and Control:** Contemporary alarm systems frequently integrate with mobile applications, enabling users to monitor their properties and manage security settings remotely from any location with internet connectivity [14, 9]. This provides unparalleled flexibility and control over security operations.
- **Increased Peace of Mind:** The presence of a robust alarm system offers significant psychological comfort and assurance. Users can have confidence that their property and family are under constant surveillance and protection, even when they are away [4].

### I.3 Connectivity Modalities of Alarm Systems: Wired versus Wireless

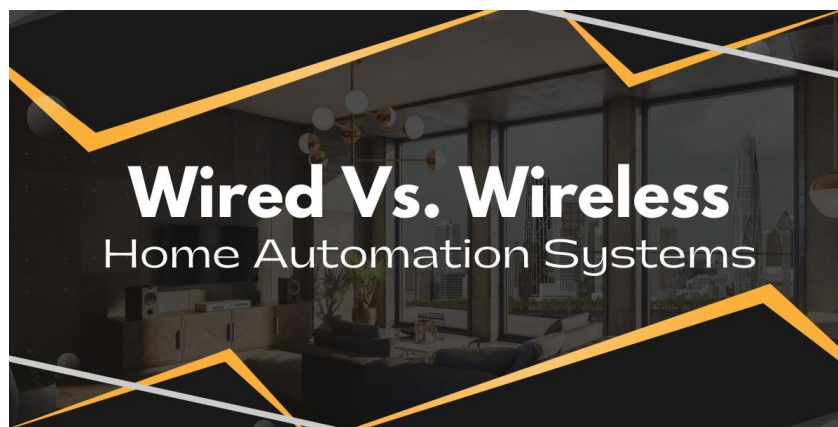


Figure I.5: Comparison of Wired and Wireless Alarm System Architectures

Alarm systems can be fundamentally differentiated by their connectivity modality: wired or wireless. Each type presents distinct advantages and disadvantages concerning installation, reliability, scalability, and security.

### I.3.1 Wired Alarm Systems

Wired alarm systems rely on physical cables to connect all sensors, control panels, and alert devices.

- **Advantages:**

- **High Reliability and Security:** Wired connections are generally less susceptible to signal interference, jamming, or hacking compared to wireless systems, offering a more robust and secure communication pathway.
- **Consistent Power Supply:** Sensors in wired systems often draw power directly from the control panel, eliminating the need for battery replacements and ensuring continuous operation.
- **Cost-Effectiveness for New Builds:** In new constructions, the cost of installing wiring can be amortized into the overall building process.

- **Disadvantages:**

- **Complex Installation:** Installation typically requires drilling, running cables through walls, and professional expertise, making it more intrusive and time-consuming, especially in existing structures.
- **Limited Flexibility:** Relocating sensors or expanding the system can be challenging and costly due to the fixed wiring infrastructure.
- **Vulnerability to Cable Damage:** Physical damage to cables (e.g., cutting) can compromise the system's integrity, though many systems incorporate tamper detection features.

### I.3.2 Wireless Alarm Systems

Wireless alarm systems utilize radio frequencies (RF) to communicate between sensors, control panels, and other components, eliminating the need for physical cabling.

- **Advantages:**

- **Easy and Flexible Installation:** Wireless systems are much easier and faster to install, requiring no drilling or extensive wiring. This makes them ideal for existing homes or rental properties.
- **Scalability and Portability:** Adding new sensors or relocating existing ones is straightforward, and the entire system can often be easily moved to a new location.
- **Aesthetics:** Absence of visible wires contributes to a cleaner aesthetic.

- **Disadvantages:**

- **Battery Dependence:** Most wireless sensors are battery-powered, requiring periodic replacement, which can be a maintenance overhead.
- **Susceptibility to Interference and Jamming:** Wireless signals can be prone to interference from other electronic devices or deliberate jamming at-

---

tempts, potentially compromising security. Advanced wireless systems often employ frequency hopping or encryption to mitigate these risks.

- **Range Limitations:** The range of wireless signals can be limited by distance, obstacles (walls, floors), and building materials, potentially requiring signal repeaters for larger properties.
- **Potential Security Vulnerabilities:** While modern wireless systems use encryption, older or less sophisticated systems might be more vulnerable to unauthorized access or signal interception.

The choice between wired and wireless systems often depends on specific requirements, budget, property type, and security priorities. Hybrid systems, combining both wired and wireless components, are also common, offering a balance of reliability and flexibility.

### I.3.3 First: Wired Alarm Systems

Wired alarm systems are characterized by their physical connections, where sensors (e.g., motion detectors, smoke alarms) are directly linked to the control panel or main unit via cables. This architecture represents a long-standing industry standard, primarily recognized for its inherent reliability and robust performance [7].

#### Advantages of Wired Alarm Systems

- **Enhanced Reliability and Stability:** The reliance on physical wiring minimizes susceptibility to signal interference, electromagnetic noise, or data loss that can plague wireless transmissions. This ensures a highly stable and continuous connection, making wired systems particularly suitable for applications demanding consistent and uninterrupted communication [7].
- **Inherent Security:** Wired systems offer a higher degree of physical security. The integrity of the system is compromised if wires or cables are severed, triggering an immediate alarm. This physical barrier makes them inherently more resistant to signal manipulation or jamming attempts compared to their wireless counterparts [4].
- **Suitability for Large-Scale and Permanent Installations:** Due to their robust and tamper-resistant nature, wired systems are widely deployed in extensive commercial buildings, industrial complexes, and newly constructed residential properties where long-term, dependable security infrastructure is paramount [7].

#### Disadvantages of Wired Alarm Systems

- **Complex and Intrusive Installation:** The installation process for wired alarm systems is often intricate, requiring extensive cable routing through walls, ceilings,

and floors. This can be a time-consuming, labor-intensive, and disruptive task, particularly in existing structures [7].

- **Higher Installation Costs:** The cumulative cost associated with materials (wires, conduits) and the specialized labor required for professional installation typically makes wired systems more expensive to deploy than wireless alternatives. This

initial investment can be a significant factor for budget-conscious projects.

- **Limited Flexibility and Scalability:** Once installed, the physical wiring infrastructure of a wired alarm system offers limited flexibility for future modifications or expansions. Adding new sensors or reconfiguring the system layout often necessitates additional intrusive wiring work, which can be both complex and costly.

### I.3.4 Second: Wireless Alarm Systems



Figure I.6: Conceptual Diagram of Wireless Alarm Systems

Wireless alarm systems operate through the transmission of signals via various radio frequency (RF) protocols, including Wi-Fi, Bluetooth, or dedicated cellular networks. Sensors communicate wirelessly with a central control panel or alarm unit, eliminating the need for physical cabling [2, 9]. This system architecture has gained significant traction due to its ease of installation and inherent flexibility [14].

#### Advantages of Wireless Alarm Systems

- **Streamlined Installation Process:** Wireless systems negate the necessity for physical cable connections between devices. This significantly simplifies and accelerates the installation process, as there is no requirement for drilling holes or running cables through structural elements, making them ideal for rapid deployment [14].
- **Enhanced Flexibility and Scalability:** The absence of fixed wiring provides exceptional flexibility. Users can readily add new sensors, cameras, or other smart devices to the system without concern for additional cabling. This inherent scalability makes wireless systems highly suitable for dynamic environments such as residential

---

properties, rental units, or temporary installations [3].

- **Ideal for Retrofit Applications:** For existing buildings or homes where installing new wiring would be impractical, disruptive, or cost-prohibitive, wireless alarm systems offer an optimal solution. They can be integrated seamlessly without requiring significant modifications to the existing infrastructure.
- **Aesthetic Integration:** From an aesthetic perspective, wireless systems often present a cleaner and more unobtrusive appearance. The absence of visible wires and cables enhances the visual appeal of modern homes and commercial spaces.

### Disadvantages of Wireless Alarm Systems

- **Susceptibility to Signal Interference:** A primary vulnerability of wireless systems is their susceptibility to signal interference. Environmental factors such as nearby electronic devices, dense building materials (e.g., thick walls), or competing wireless signals can disrupt communication between sensors and the control panel, potentially degrading system performance [9].
- **Battery Dependence and Maintenance:** Wireless sensors typically rely on batteries for power, necessitating periodic replacement or recharging. Neglecting proper battery maintenance can lead to sensor failure, rendering parts of the system ineffective or unreliable [14].
- **Potential Security Vulnerabilities:** While modern wireless systems employ sophisticated encryption and anti-jamming techniques, they can still be theoretically vulnerable to signal manipulation, unauthorized access, or jamming attacks. Malicious actors who can intercept or disrupt wireless signals might potentially compromise system functionality. However, continuous advancements in security protocols are aimed at mitigating these threats [4].

## I.4 Societal and Economic Impact of Alarm Systems

Alarm systems play a pivotal role in safeguarding human lives and valuable assets, yielding significant benefits for individuals and the broader community [4, 7].

### I.4.1 Benefits for Individuals and Property

- **Proactive Threat Prevention:** Alarm systems provide a crucial layer of defense by deterring potential intruders or criminals, thereby preventing unauthorized access and mitigating threats to personal safety and property before incidents occur [4].
- **Timely Early Warning:** These systems deliver real-time alerts to occupants or designated authorities upon detecting an anomaly. This immediate notification provides critical time for early intervention, such as evacuating a building during a

---

fire or contacting emergency services during a burglary [7].

- **Reduction in Injuries and Casualties:** By facilitating rapid response to hazardous events, alarm systems significantly reduce the likelihood of severe injuries or fatalities associated with fires, gas leaks, or other life-threatening incidents.
- **Asset Protection and Loss Mitigation:** The systems actively safeguard valuable assets by detecting theft or attempted unauthorized access, thereby minimizing financial losses and potential damage to property [4].
- **Deterrence of Criminal Activity:** The mere visible presence of an alarm system, often accompanied by audible alerts, serves as a powerful deterrent to criminals, making properties less attractive targets for illicit activities.
- **Prevention of Vandalism:** Integrated alarm systems, especially those incorporating surveillance cameras, can effectively deter acts of vandalism by providing both a warning mechanism and evidence capture capabilities [4].

Collectively, these benefits contribute to:

- An elevated sense of personal and property security, fostering an environment of reduced criminal vulnerability.
- An increase in overall security levels, further augmented by the increasing sophistication of smart home alarm systems.
- Enhanced remote monitoring capabilities through the integration of sensors, cameras, and mobile applications, empowering users with ubiquitous oversight of their properties [9].
- Greater user control over family and property safety, ensuring continuous presence and overall enhanced security [14].

#### I.4.2 Enhanced Emergency Response

Alarm and communication systems significantly improve the efficacy and timeliness of emergency service responses by facilitating rapid notification and intervention during critical events [4, 7].

Upon the detection of any threat, be it a fire or an intrusion, these systems immediately alert both the user and designated emergency monitoring centers. This instantaneous communication ensures that all relevant parties are updated in real-time, enabling prompt decision-making.

Crucially, even when occupants are not present, these systems can dramatically reduce emergency response times for police, fire departments, and ambulance services by directly notifying them as soon as a threat is detected [4]. This direct linkage minimizes the impact of emergencies and can be instrumental in saving lives and mitigating extensive damage.

For specific emergencies, such as fires, automated system responses can include unlocking doors for quicker evacuation or cutting off power supplies to prevent further hazards. Such automated actions minimize human intervention, reduce delays, and limit potential harm.

Furthermore, the advent of remote control capabilities allows users to interact with these systems from any location via live video feeds or mobile applications. This real-time access and control enable swift actions to enhance safety during emergency situations [9].

## I.5 Domotics (Home Automation)



Figure I.7: Conceptual Overview of Domotics (Home Automation)

### I.5.1 Definition of Domotics

Domotics, a portmanteau derived from the Latin word "*domus*" (house) and "robotics" or "informatics," refers to the application of technology to control and automate various systems and appliances within a residential environment. This encompasses a wide range of functionalities, including lighting, heating, ventilation, security, and entertainment

---

systems. It is effectively synonymous with smart home technologies [17, 7, 9].

### I.5.2 Core Systems Supported by Domotics

Domotics systems integrate and manage several key household functionalities:

- **Lighting Control:** Automation of lighting based on predefined schedules, ambient light levels, or motion detection to optimize energy consumption and ambiance [14].
- **Heating, Ventilation, and Air Conditioning (HVAC):** Intelligent climate control through smart thermostats that manage heating, cooling, and air circulation to maintain optimal indoor temperatures and energy efficiency.
- **Environmental Quality Management:** Monitoring and regulation of indoor air quality, including ventilation systems to manage airflow and pollutant levels.
- **Appliance and Electronics Control:** Automated operation and monitoring of household appliances and electronic devices for convenience and energy

management.

- **Integrated Security Systems:** Comprehensive integration of smart cameras, alarms, and various sensors to provide enhanced surveillance, threat detection, and alert capabilities [9, 7].

### I.5.3 Control Methodologies in Domotics

Domotics systems primarily employ two principal methods for controlling automated functions:

- **Scheduled Events:** Devices and systems are configured to operate automatically at predetermined times or intervals, such as lights turning on at dusk or irrigation systems activating in the morning.
- **Environmental Triggers:** Systems respond dynamically to changes detected by sensors in the environment, such as activating lighting when motion is detected, or adjusting heating based on temperature fluctuations [14].

### I.5.4 Objectives and Benefits of Domotics

The implementation of domotics systems aims to achieve several critical objectives:

- **Enhanced Comfort and Convenience:** By automating routine tasks and centralizing control of home systems, domotics significantly improves user comfort and simplifies daily living.
- **Optimized Energy Efficiency:** Intelligent management of lighting, HVAC, and appliances leads to substantial reductions in energy consumption, translating into lower utility bills and a reduced environmental footprint [9].

- **Improved Security and Safety:** Integrated security features provide continuous monitoring, immediate alerts for emergencies or suspicious activities, and automated responses, thereby augmenting the overall safety of the premises.
- **Increased Accessibility and Independent Living:** Domotics systems can be tailored to assist individuals with limited mobility or disabilities, enabling them to control their home environments more easily and facilitating independent living [17].

### I.5.5 Key Components of Domotics Systems

Domotics systems are inherently interconnected and comprise three essential components that collaborate to provide comprehensive home automation and control [7].

#### Control Systems

The control system functions as the central processing unit and "brain" of the domotics architecture. It is responsible for interpreting data received from sensors, applying pre-defined logic, and issuing commands to actuators. Control systems typically include:

- A main control unit, which can range from a dedicated embedded microcontroller (e.g., Arduino, Raspberry Pi) to a more powerful server or cloud-based platform.
- Automated management software, running either locally on the control unit or as a cloud-based service, which orchestrates the system's functionalities.

#### Sensing Systems

Sensing systems comprise various sensors that continuously monitor and detect changes within the home's environment. These sensors acquire real-time data and transmit it to the control system for analysis and decision-making [9, 14]. Common types of sensors in domotics include:

- **Motion Sensors:** Employed to detect movement within rooms or designated areas, frequently integrated into security systems for intruder detection.
- **Temperature and Humidity Sensors:** Used to gauge indoor climatic conditions, allowing for intelligent adjustment of heating, ventilation, and cooling systems.
- **Door/Window Sensors:** Notify the system of the open or closed status of entry points, commonly integrated into security protocols to alert users to unauthorized access.
- **Smoke Detectors:** Critical safety devices that detect the presence of smoke or fire, activating alarms and sending notifications to occupants and emergency services.
- **Carbon Monoxide Detectors:** Essential for safety, detecting hazardous levels of

---

CO, an odorless and colorless gas.

### Communication Systems

The communication system forms the backbone of a domotics network, enabling seamless connectivity and data exchange among all devices within the home. These systems facilitate the flow of information between sensors, control units, actuators, and user interfaces (e.g., smartphones, tablets, or computers) [3, 9]. Major communication technologies commonly employed in domotics include:

- **Wi-Fi:** A ubiquitous wireless local area networking technology widely used for connecting smart devices within a home.
- **Bluetooth:** Primarily used for short-range wireless communication between devices, often for direct control or pairing.
- **Low-Power Wireless Mesh Networks (e.g., Zigbee):** Protocols designed for low-power, low-data-rate applications, highly suitable for interconnected home automation devices that form a mesh network for increased reliability.
- **Z-Wave:** A proprietary wireless communication protocol specifically developed for home automation, emphasizing low power consumption and ease of use.
- **GSM/GPRS Systems:** Such as those utilizing SIM900 modules, enabling remote communication over cellular networks. This allows the system to send SMS alerts or access data services via mobile networks, extending control and monitoring capabilities beyond the local network.

### I.5.6 Real-World Applications of Domotics

The widespread adoption of domotics has significantly enhanced daily living by offering practical applications that contribute to comfort, security, energy efficiency, and overall convenience [13].

#### Automated Lighting Control

Domotics enables intelligent management of lighting systems, adapting illumination based on dynamic needs:

- **Occupancy-Based Lighting:** Lights can be automatically switched on or off in

---

response to detected occupancy in a room, optimizing energy use and convenience [14].

- **Scheduled and Adaptive Lighting:** Users can establish time-based routines for lighting, such as automatic activation at dusk or deactivation at bedtime. Systems can also adapt to natural light levels.
- **Dynamic Brightness Control:** Lighting levels can be adjusted to create specific ambiances or reduce energy consumption when full brightness is not required, offering granular control over illumination.

### Intelligent Heating, Ventilation, and Cooling (HVAC) Management

Domotics systems provide comprehensive control over HVAC components, optimizing both indoor comfort and energy efficiency [8, 7, 14]:

- **Temperature Optimization:** The system can automatically adjust indoor temperatures based on real-time occupancy, user preferences, or external weather conditions, ensuring comfort while minimizing energy waste. For instance, air conditioning can activate automatically on warm days to maintain a comfortable internal environment [14].
- **Centralized Climate Control:** Intelligent management of central heating and air conditioning units allows for precise temperature regulation across different zones of a property.
- **Energy Conservation Modes:** Domotics can significantly conserve energy by automatically reducing HVAC operation when a property is unoccupied or during off-peak hours (e.g., lowering heating at night) [9].
- **Adaptive Climate Response:** Heating and cooling systems operate only when necessary, responding to real-time environmental conditions and occupancy data, leading to a more environmentally conscious energy footprint [7].

### Advanced Security and Surveillance Systems

Smart home security applications are significantly enhanced through domotics integration [4, 7, 9]:

- **Comprehensive Home Security:** The seamless integration of various sensors, alarms, and surveillance technologies significantly increases the overall safety and protection of a property.

- **Remote Surveillance via Smart Cameras:** Smart cameras can be strategically placed indoors and outdoors, accessible remotely over the internet. This allows users to monitor their property from virtually anywhere at any time using a smartphone or computer.
- **Automated Intruder Detection and Alerts:** Cameras and motion sensors can detect unusual movement. In the event of potential intrusions or other threats, the system can automatically trigger alarms, send immediate alerts to users, or even notify emergency services directly.
- **Smart Locks and Access Control:** Smart locks enable remote locking and unlocking of doors via mobile applications, providing convenience for granting access to trusted individuals while away and ensuring security upon departure without manual checks.
- **Integrated Emergency Response:** Most smart home security systems are designed to link directly with alarm monitoring centers and emergency response services, ensuring instantaneous alerts and rapid deployment of assistance in critical situations.

## I.6 Societal Relevance of Alarm and Communication Systems in Daily Life

Alarm and communication systems provide fundamental benefits for both individuals and the collective society, particularly concerning personal safety and asset protection [4, 7].

### I.6.1 Benefits for Individuals and Property

- **Proactive Threat Mitigation:** These systems offer comprehensive protection against intruders, theft, and other potential threats to individuals and property, emphasizing prevention rather than mere reaction [4].
- **Timely Notification and Intervention:** Alarm systems deliver real-time alerts to occupants or relevant authorities, providing crucial time for early intervention, such as evacuating premises or summoning assistance [7].
- **Reduction of Injury Risk:** By facilitating rapid response to hazardous events, these systems significantly reduce the potential for severe injuries and life-threatening accidents.
- **Asset Preservation:** Systems safeguard assets by detecting theft, attempted unauthorized access, and other damaging incidents, thereby minimizing financial losses and property damage [4].
- **Deterrence of Malicious Intent:** The visible presence and audible activation of alarm systems often serve as a strong deterrent, discouraging criminals from

targeting premises.

- **Vandalism Prevention:** Integrated alarm systems, particularly those with surveillance capabilities, are highly effective in deterring acts of vandalism [4].

These integrated benefits collectively contribute to:

- An elevated sense of safety and security, leading to a reduction in perceived and actual vulnerability to criminal activities.
- A heightened level of security assurance, particularly as smart home alarm systems continue to evolve in sophistication.
- Enhanced remote oversight, enabling users to continuously monitor their homes and properties through integrated sensors, cameras, and mobile applications [9].
- A significant increase in overall security by empowering users to effectively manage and monitor family safety and property integrity, ensuring continuous presence and overall enhanced protection [14].

### I.6.2 Optimized Emergency Response

Alarm and communication systems critically enhance the efficiency and speed of emergency service responses by facilitating prompt and precise information dissemination during emergencies [4, 7].

In the event of various threats, from fire to intrusion, detection systems immediately alert both the user and designated emergency response centers. This instantaneous communication ensures that all pertinent parties are kept informed in real-time, enabling

rapid and coordinated action.

Crucially, even when individuals are not physically present, these systems can dramatically improve emergency response times (e.g., for police, fire departments, or medical services) by automatically notifying them as soon as a threat is detected [4]. This direct and swift linkage is paramount in minimizing the impact of emergencies and can be pivotal in saving lives and mitigating extensive property damage.

For specific emergency scenarios, such as a fire, automated system responses can include actions like unlocking automated doors to facilitate quicker evacuation or cutting off electrical power to prevent further hazards. These automated functions minimize delays inherent in human response and reduce the potential for cascading damage.

A significant modern advantage is the integration of remote control capabilities: users can interact with these systems from any location through live video feeds or dedicated mobile applications, enabling them to take swift action to enhance safety during emergencies, even when physically distant [9].

## I.7 Implications and Foundation for Future Work

This chapter has demonstrated how technological advancements have fundamentally re-

---

shaped the landscape of security and home management. The synergistic integration of traditional alarm system principles with the capabilities of domotics technology is driving a revolution in the creation of intelligent and secure environments for both residential and commercial spaces [17, 7].

Modern solutions empower individuals with more effective and safer control over their living spaces, offering substantial advantages such as:

- Improved personal and familial security, fostering a sense of safety and peace of mind.
- Rapid and efficient incident response, minimizing potential harm and losses.
- Enhanced user assurance through continuous automation and ubiquitous monitoring.

These integrated systems provide robust frameworks for addressing contemporary security and environmental challenges faced by individuals and communities alike. The convergence of alarm systems with domotics is therefore indispensable for fostering safer and smarter premises.

This comprehensive overview serves as a foundational context for subsequent chapters, which will delve into the specific technologies and methodologies employed in the practical implementation of smart alarm systems, particularly focusing on platforms such as the Arduino and communication modules like the SIM900.

## **Chapter II**

# **Arduino, Modules, and Board**

The increasing prevalence of remote-control applications in modern life is directly attributable to advancements in smart technology. This evolution has lowered the barrier to entry for electronics, fostering greater innovation. The Arduino platform is a prominent open-source tool for rapid prototyping and automation, valued for its ease of use in controlling devices, acquiring measurements, and establishing communication links [17].

A key advantage of Arduino is its extensibility through “shields”—modular hardware additions supported by an extensive ecosystem of software libraries. Shields augment project functionalities without requiring intricate modifications to the main board. For example, the GSM Shield SIM900 facilitates mobile network communication, such as SMS transmission, directly from the Arduino.

This chapter will delve into the architecture and programming of Arduino. Furthermore, a detailed examination of the SIM900 Shield will be provided, given its critical role in the proposed system.

## II.1 Arduino Platform Overview

Arduino is an open-source electronics platform established in 2005 by Massimo Banzi and his colleagues. Its primary objective was to provide an accessible tool for students and hobbyists to learn electronics and programming [6].

A typical Arduino board fundamentally comprises the following core components:

- A microcontroller, which functions as the central processing unit (CPU) of the system, executing programmed instructions and managing input/output operations.
- A comprehensive set of input/output (I/O) pins, encompassing both digital and analog capabilities, designed for seamless interfacing with external components such as sensors, actuators (e.g., motors, LEDs), and other electronic devices.
- A USB interface for facilitating program uploads and serial communication with a host computer.
- Power management circuitry to regulate and supply the necessary electrical power to the board and connected components.

Programming of Arduino boards is executed using the Arduino Integrated Development Environment (IDE), which employs a simplified dialect of the C/C++ programming language. The user-developed program within the Arduino IDE is referred to as a “sketch.” Upon successful upload to the microcontroller, the sketch executes continuously, enabling real-time interaction and control of the connected hardware [5].

### II.1.1 The ATmega328P Microcontroller: Architecture and Memory

At the core of many Arduino boards, notably the Arduino Uno, lies the ATmega328P

microcontroller manufactured by Atmel (now a division of Microchip Technology) [11]. This compact integrated circuit integrates essential computational and memory elements:

- **Central Processing Unit (CPU):** Executes programmed instructions.
- **Memory Architecture:** The ATmega328P features distinct memory types crucial for its operation:
  - **Flash Memory (32 KB):** Non-volatile storage for the uploaded program code (firmware), ensuring persistence even without power.
  - **SRAM (Static Random-Access Memory) (2 KB):** Volatile memory utilized for temporary data storage, including variables and dynamic data, during active program execution. SRAM offers fast access times for real-time operations.
  - **EEPROM (Electrically Erasable Programmable Read-Only Memory) (1 KB):** Non-volatile memory designated for persistent data storage, such as configuration parameters or calibration settings that must be retained across power cycles.
- **Processing Speed:** Operates at a frequency of 16 MHz, providing sufficient computational power for a wide range of embedded tasks.
- **Input/Output Capabilities:** Equipped with multiple digital and analog I/O pins, offering flexible interfacing for a broad spectrum of external electronic components, sensors, and actuators.

This integrated memory structure and versatile I/O system enable the ATmega328P to efficiently manage and execute user-defined programs for various embedded applications.

## II.2 Arduino Input/Output (I/O) Pins and Peripherals

The I/O pins on an Arduino board serve as the primary interface for connecting to external circuitry, often in conjunction with a breadboard and connecting wires. These pins are typically equipped with headers that facilitate straightforward connection of jumper wires. Arduino boards feature various types of pins, each clearly labeled on the board itself to indicate its specific function.

### II.2.1 Digital Pins

The Arduino Uno, for instance, is equipped with 14 digital pins, labeled from 0 to 13. These pins can be configured as either an OUTPUT (to transmit control signals to actuators) or an INPUT (to read binary values from external digital devices). Digital pins operate in two discrete logical states:

- **HIGH:** Corresponding to a voltage level of approximately +5 volts.

- **LOW:** Corresponding to a voltage level of approximately 0 volts (ground).

## II.2.2 Special Functions of Digital Pins

Beyond basic digital I/O, several digital pins offer specialized functionalities that enhance the board's versatility [6]. The two most prominent are UART communication and Pulse Width Modulation (PWM).

### UART (Universal Asynchronous Receiver-Transmitter)

UART provides a robust mechanism for serial communication, enabling the Arduino to exchange data with other devices (e.g., computers, sensors, other microcontrollers). On the Arduino Uno, digital pins 0 (RX) and 1 (TX) are dedicated for this purpose, forming the hardware serial port.

- **TX (Transmit):** Sends data from the Arduino to an external device.
- **RX (Receive):** Receives data from an external device into the Arduino.

This serial connection is fundamental for debugging, program uploading, and interfacing with serial-enabled devices [6].

### PWM (Pulse Width Modulation)

PWM is a technique that allows the Arduino to simulate an analog output using digital signals. By rapidly toggling a digital pin between HIGH and LOW states at a fixed frequency, and by varying the proportion of time the signal remains HIGH (the "duty cycle"), the Arduino can effectively control analog-like behaviors. This includes seamlessly adjusting LED brightness, regulating DC motor speed, or generating analog voltage levels, offering finer control than simple ON/OFF states [5].

- **UART:** Facilitates asynchronous serial communication for transmitting and receiving data between devices.
- **PWM:** Enables the simulation of analog control by precisely adjusting the duty cycle of a digital signal.

## II.2.3 Analog Pins and Analog-to-Digital Conversion

The Arduino Uno is equipped with six analog input pins, labeled A0 through A5. These pins are specifically designed to read analog signals—continuous voltage values typically generated by sensors like thermistors (temperature) or photoresistors (light intensity).

Since the Arduino is a digital device, it employs an Analog-to-Digital Converter (ADC) to transform these continuous analog voltages (typically 0 to 5V) into discrete numerical

values. The Uno's ADC features a 10-bit resolution, meaning it can represent the input voltage as one of  $2^{10} = 1024$  distinct integer values, ranging from 0 to 1023 [3]. For example, an input voltage of 2.5V will yield a numerical value approximating 512, while 5V will be represented by 1023.

The choice of a 10-bit ADC for the ATmega328P microcontroller is a pragmatic design decision influenced by several factors:

- **Optimal Balance:** A 10-bit resolution achieves an optimal balance between conversion speed and measurement precision, providing sufficient accuracy for most embedded projects while maintaining an adequate conversion rate for real-time applications.
- **Energy Efficiency:** 10-bit ADCs are notably efficient in terms of power consumption, which is crucial for embedded systems and battery-operated devices.
- **Industry Standard:** The adoption of 10-bit ADCs is a prevalent standard across many microcontrollers, fostering compatibility with a wide array of existing components and simplifying development.
- **Sufficient Granularity:** With 1024 discrete levels, a 10-bit ADC offers ample detail for most common physical measurements, balancing granularity with computational simplicity.

## II.2.4 Onboard Peripherals

Arduino boards are also equipped with several highly practical auxiliary components that enhance control and provide immediate feedback:

- **USB Port:** The USB port serves multiple essential functions: it is the primary interface for uploading compiled code, facilitates real-time debugging via serial communication, and provides power to the entire board during development and operation when connected to a host computer.
- **Voltage Regulator:** A built-in voltage regulator ensures that the Arduino's sensitive electronic components receive a consistent and stable power supply (typically 3.3V or 5V), shielding the board from fluctuating input voltages.
- **Auxiliary LEDs and Buttons:**
  - **Reset Button:** This tactile button allows for the immediate restart of the program executing on the Arduino microcontroller without requiring a power cycle.
  - **LED Indicators:** An integrated LED (commonly connected to digital pin 13) can be controlled by user programs for basic testing. Dedicated TX (Transmit) and RX (Receive) LEDs visually indicate data flow over the serial communication (USB) connection, blinking when data is being sent or received.

## II.3 Arduino Programming Environment (Arduino IDE)

To simplify and democratize the process of programming Arduino boards, the Arduino IDE (Integrated Development Environment) was developed. It functions as the primary tool for users to seamlessly write, verify, and upload code to their Arduino boards, eliminating the need for complex configurations or supplementary software installations [amestica2019experimental].

### II.3.1 Features of the Arduino IDE

The Arduino IDE is distinguished by a rich set of features that effectively cater to a wide spectrum of users, from novice programmers to experienced professionals [amestica2019experimental]

- **Intuitive Interface:** A clean and user-friendly text editor where code is written, ensuring users can quickly commence programming.
- **Integrated Compiler:** Automatically translates C/C++ based Arduino code into machine-readable instructions, eliminating manual compilation steps.
- **Extensive Libraries:** A comprehensive collection of pre-built software libraries provides ready-to-use functions for interfacing with various sensors, motors, and displays, significantly reducing development time.
- **Effortless Uploading:** Streamlined, single-click process for transferring compiled programs to the Arduino board via USB.
- **Serial Monitor:** A highly valuable real-time console for displaying data output from the Arduino board, indispensable for debugging, monitoring sensor readings, and understanding program execution.
- **Seamless Expansion and Updates:** Simplifies installing new board drivers and external libraries, allowing effortless integration of new hardware and functionalities.

### II.3.2 Understanding the Arduino IDE Interface

The Arduino IDE interface comprises several key components:

- **Toolbar:** Contains frequently used buttons for verifying (compiling) code, uploading code, and standard file operations.
- **Text Editor:** The primary workspace for writing and editing Arduino sketches, featuring syntax highlighting.
- **Message Window:** Displays compilation messages, error notifications, warnings, and success messages, crucial for identifying and rectifying coding errors. **Serial Monitor Window:** Provides a real-time display of data transmitted from the Arduino board via its serial connection, invaluable for debugging and data logging.
- **Board and Port Settings:** Users must accurately select the specific Arduino board model and the serial port to which it is connected via the "Tools" menu to ensure correct communication.

### II.3.3 Program Development and Upload Workflow

The process of programming and uploading a sketch to an Arduino board is generally straightforward once the foundational steps are understood [2]:

1. **Code Development:** Compose the program code, focusing initially on functionality.
2. **Board Selection:** Navigate to ‘Tools > Board’ and select the specific Arduino model (e.g., Arduino Uno).
3. **Port Configuration:** Identify and select the correct USB serial port (via ‘Tools > Port’) where the Arduino is connected.
4. **Code Verification:** Use the ‘Verify’ function (checkmark icon) to compile the code and check for syntax errors, preventing the upload of non-functional programs.
5. **Program Upload:** Once verified, click the ‘Upload’ button (right arrow icon) to transfer the compiled program onto the Arduino’s microcontroller.
6. **Real-time Monitoring:** Open the ‘Serial Monitor’ to observe the Arduino’s real-time output, facilitating debugging and operational analysis.

The Serial Monitor is an integral and often indispensable feature embedded within the Arduino IDE. It provides a real-time textual interface, effectively acting as a direct communication channel with the Arduino board during program execution. This allows developers to observe incoming sensor data, status messages, and error information, significantly streamlining the debugging process [amestica2019experimental, 2].

### II.3.4 Illustrative Example: LED Blinking Program

As a foundational example, this section demonstrates the implementation of a program designed to control an integrated LED, commonly found on the Arduino Uno board. The program is configured to repeatedly toggle the LED between an ON state for one second and an OFF state for another second, illustrating a fundamental concept in microcontroller programming.

## II.4 Arduino Shields: Concept and Functionality

The increasing adoption of Arduino in electronic projects necessitated a scalable and user-friendly method for expanding its capabilities without resorting to complex circuit design and construction. This demand led to the development of the Arduino Shield concept.

Shields are specialized electronic boards engineered to interface directly with Arduino main boards. They are designed to add specific functionalities, thereby simplifying project development and accelerating prototyping [15].

### II.4.1 Definition and Advantages of Arduino Shields

An Arduino Shield is an extension board that physically stacks atop an Arduino main board. Its pin configuration is designed to align precisely with the Arduino's headers, enabling direct mechanical and electrical connection without the need for additional wiring [15].

The widespread appeal of Arduino Shields can be attributed to several key advantages [15]:

- **Simplified Wiring:** Shields eliminate the need for intricate wiring by directly plugging into the Arduino's existing pin headers.
- **Expanded Functionality:** They provide pre-packaged solutions for adding diverse functionalities (e.g., Wi-Fi, motor control, display interfaces).
- **Accelerated Development:** By offering ready-to-use modules, shields significantly reduce the time and effort required to integrate complex features.
- **Guaranteed Compatibility:** Designed specifically for Arduino, ensuring high levels of hardware and software compatibility.
- **Stackability:** Multiple shields can often be stacked, allowing for the integration of several functionalities within a single project.

### II.4.2 Common Shield Examples

A wide variety of shields are available, each tailored to specific application requirements. Examples include Ethernet Shields, Motor Driver Shields, LCD Shields, and GPS Shields, among many others [15].

### II.4.3 Installation and Usage of Arduino Shields

The installation of a shield typically involves physically pressing it onto the Arduino board's headers until a secure connection is established. Subsequent steps involve:

- **Software Integration:** Incorporate the specific software library designed for the shield into the Arduino sketch, usually available through the Arduino IDE's Library Manager.
- **Circuit Interfacing:** Ensure that any external components connected to the shield align with its pinout and the logic defined in the code.
- **Functional Verification:** Upload a basic test program to the Arduino to confirm the shield's correct operation.

### II.4.4 Important Considerations for Shield Utilization

While shields simplify development, certain considerations are crucial for their effective and safe utilization:

- **Voltage Compatibility:** Verify the operating voltage requirements of the shield.

Some shields may operate at 3.3V, while the Arduino Uno typically uses 5V, necessitating appropriate level shifting or voltage regulation to prevent damage.

- **Pin Conflicts:** Be aware of potential pin conflicts when stacking multiple shields. If two shields attempt to utilize the same Arduino pins, conflicts can arise, leading to unexpected behavior or system malfunction.
- **Power Consumption:** High-power shields may draw significant current. Ensure that the power supply to the Arduino (or an external power source) is sufficient to meet the combined power demands of the board and all connected shields.

## II.5 Introduction to the GSM Shield SIM900

In the contemporary landscape of pervasive connectivity, the ability of embedded systems to communicate over wide areas is paramount. GSM (Global System for Mobile Communications) shields enable Arduino-based projects to leverage cellular networks for data transmission. The GSM Shield SIM900 is a widely adopted communication module for Arduino projects, providing connectivity to standard cellular infrastructure for functionalities such as text messaging and internet access.

GSM is a globally recognized standard for digital cellular networks, constituting the underlying technology employed by mobile phones for voice communication and SMS (Short Message Service). When integrated with an Arduino via a GSM shield, the system gains the capability to:

- Transmit and receive text messages.
- Initiate and receive voice calls.
- Access the internet via cellular data connections.

The GSM Shield SIM900 connects directly to the Arduino board and is controlled through a standardized set of text-based commands known as AT commands, allowing for programmatic management of its various functions [10].

### II.5.1 Overview and Key Features of the SIM900 GSM/GPRS Shield

The SIM900 GSM/GPRS Shield is a communication module built around the SIM900 chip, a product of SIMCom Wireless Solutions. SIMCom, established in 2002 and headquartered in Shanghai, China, specializes in designing and manufacturing wireless communication modules for technologies including 5G, 4G, LTE, GSM, and GNSS (Global Navigation Satellite System). SIMCom modules are extensively deployed across diverse industries, such as the Internet of Things (IoT), automotive telematics, and smart metering solutions [10].

The SIM900 Shield empowers an Arduino board to establish connections to mobile networks and execute several critical functions:

- Initiate and receive voice calls.

- Send and receive SMS messages.
- Facilitate data transfer over the internet using GPRS (General Packet Radio Service). GPRS is a packet-switched technology enabling the transmission and reception of small data packets over existing mobile phone networks. This allows devices like the SIM900 Shield to utilize internet connectivity for modest data amounts (e.g., sensor readings) even without a high-speed broadband connection, making it suitable for many IoT projects characterized by low data exchange volumes [10].

The SIM900 Shield features a compact form factor designed for straightforward integration onto the Arduino board and provides full support for AT commands. AT commands are textual strings used to control and configure communication modules. The prefix "AT" signifies "attention," signaling to the module that a command follows. These commands allow users to perform operations like initiating calls, sending SMS messages, or establishing internet connections programmatically.

#### **Examples of Basic AT Commands:**

- 'AT': Used to check the module's responsiveness and operational status.
- 'AT+CMGF=1': Configures the module to text message mode (PDU mode 0).
- 'AT+CSQ': Queries the module for the current signal strength indication.

AT commands provide a simple yet powerful interface for interacting with the GSM module.

The SIM900 Shield is well-suited for applications requiring remote communication capabilities. Its key features include [10]:

- **Quad-band Operation (850/900/1800/1900 MHz):** This feature ensures broad compatibility with GSM networks across most global regions [10].
- **GPRS for Internet Connectivity:** Provides essential internet access, suitable for low-bandwidth data transfers, such as sensor data telemetry [10].
- **SMS Functionality:** Enables bidirectional SMS messaging directly through Arduino code [10].
- **Voice Call Capabilities:** Supports initiating and receiving voice calls, along with DTMF (Dual-Tone Multi-Frequency) tone recognition [12]. DTMF is a signaling system where pressing numbers on a telephone generates a unique sound composed of two distinct pure-tone sinusoidal frequencies. This allows the receiving end to decode the pressed number, enabling remote control or interaction with systems.
- **UART Communication Interface:** The SIM900 module communicates with the Arduino board primarily through serial communication, utilizing AT commands [10]. UART (Universal Asynchronous Receiver-Transmitter) is a hardware communication protocol enabling two devices to exchange data serially, typically using two dedicated wires: TX (Transmit) for sending data and RX (Receive) for receiving data. Data is transmitted as electrical pulses at a predefined speed (baud rate), ensuring synchronized understanding.

- **Low Power Consumption in Idle State:** The SIM900 module is designed for energy efficiency, particularly when in standby or idle mode. Its low power consumption in these states is advantageous for battery-powered applications, contributing to extended operational longevity [10].
- **Integrated LEDs:** The module features integrated Light Emitting Diodes (LEDs) that provide visual indicators for network registration status and power supply, aiding users in easily monitoring the system's operational condition [leekongxue2021applications].

## II.5.2 Technical Specifications of the SIM900 Shield

The salient technical specifications of the SIM900 Shield are summarized in Table II.1.

Table II.1: Technical Features of the SIM900 Shield

Feature	Value
Operating Voltage	5V
Standby Power Consumption	Approximately 2.5 mA
Interface	UART Serial (Rx/Tx)
Default Baud Rate	19200 bps
SMS Support	Send and receive SMS messages
Voice Support	Voice calls and DTMF transmission
Internet Support	TCP/UDP protocols over GPRS
Control	Via AT Commands set

## II.6 Connecting and Configuring the SIM900 Shield with Arduino

Integrating the SIM900 Shield with an Arduino board requires careful physical connection and appropriate software configuration to ensure correct operation. This includes ensuring proper voltage levels, accurate baud rate settings, and correct handling of serial communication signals [10].

### II.6.1 Physical Setup and Power Management

Setting up the SIM900 Shield with an Arduino involves several fundamental steps:

- **Shield Attachment:** The SIM900 Shield is designed for direct mounting onto the Arduino board's headers. Proper pin alignment and gentle, even pressure will secure the connection. For standalone SIM900 modules, jumper wire connections for both serial communication and power are necessary.
- **Power Management:** Adequate power supply is crucial, as the SIM900 module can exhibit significant current draw, particularly during active communication phases (e.g., making calls or transferring data). The Arduino board's 5V output typically does not provide sufficient current to power the module reliably [10]. Therefore, it is strongly advised to utilize an external power supply that provides the correct voltage and possesses an adequate current rating. This external supply

should ideally be coupled with a suitable voltage regulator to deliver stable power to the SIM900 Shield, preventing unexpected resets or operational instability caused by insufficient current [10, 16].

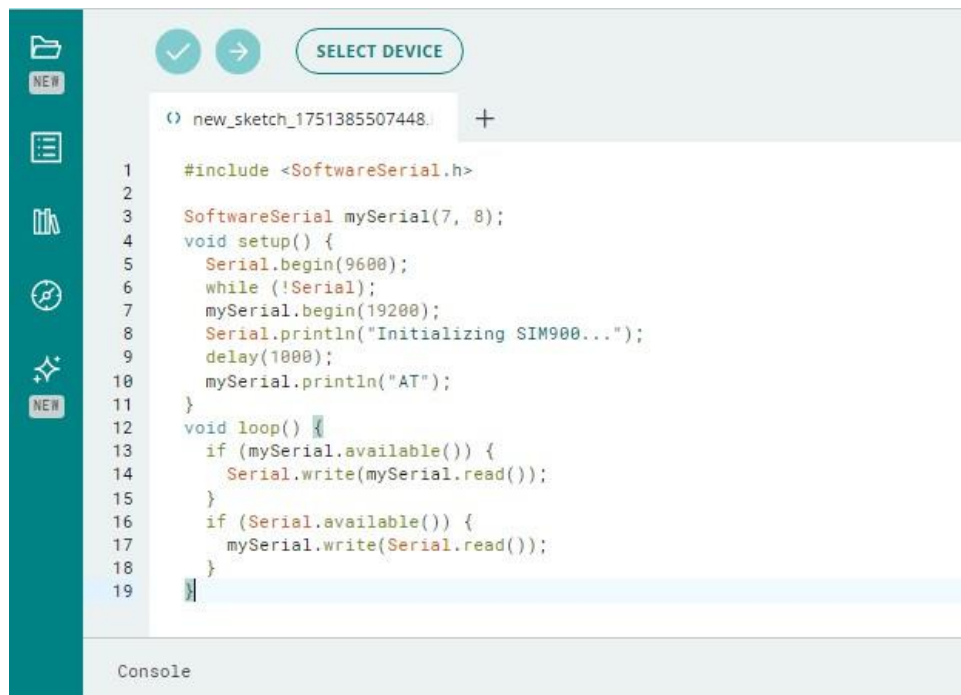
- **SIM Card Insertion:** A functional SIM card with an active service plan is required. Ensure correct orientation when inserting the SIM card into the designated slot on the shield. It is imperative to disable any SIM card Personal Identification Number (PIN) configurations, as a locked SIM card will prevent the module from registering with the mobile network.

## II.6.2 Software Communication: Serial Interface and Baud Rate Synchronization

### Utilizing the ‘SoftwareSerial’ Library for Virtual Serial Ports

The Arduino Uno board is equipped with only one hardware serial port, located on digital pins 0 (RX) and 1 (TX). This port is concurrently used for both program uploading and communication with the Serial Monitor. To enable communication with the SIM900 Shield without interfering with these essential functions, a virtual serial port can be created using the ‘SoftwareSerial’ library [1]. This library allows developers to designate almost any pair of digital pins on the Arduino board to function as additional RX and TX pins. This capability enables the Arduino to establish serial communication with the SIM900 module via these newly defined pins, thereby freeing the primary hardware serial port for debugging activities and program uploads [1].

#### Example of initial setup code using ‘SoftwareSerial’:



```
1  #include <SoftwareSerial.h>
2
3  SoftwareSerial mySerial(7, 8);
4  void setup() {
5      Serial.begin(9600);
6      while (!Serial);
7      mySerial.begin(19200);
8      Serial.println("Initializing SIM900...");
9      delay(1000);
10     mySerial.println("AT");
11 }
12 void loop() {
13     if (mySerial.available()) {
14         Serial.write(mySerial.read());
15     }
16     if (Serial.available()) {
17         mySerial.write(Serial.read());
18     }
19 }
```

Figure II.1: Initial setup code using SoftwareSerial

### Importance of Baud Rate Synchronization

The baud rate is a critical parameter in serial communication, defining the rate at which data bits are transmitted per second. For the Arduino and SIM900 module to communicate effectively, their baud rates must be synchronized. A mismatch in baud rates will lead to communication failures, as one device transmits data at a speed the other cannot correctly interpret.

The SIM900 module typically operates with a default baud rate of 19200 bps as configured by the manufacturer. This rate provides a balanced performance for most common applications. The rationale for 19200 bps as a common standard includes:

- **Compatibility:** A baud rate of 19200 bps is widely supported across various microcontrollers, computers, and peripheral devices, minimizing potential communication issues when integrating different components.
- **Reliability:** While higher baud rates (e.g., 115200 bps) are achievable, they are more susceptible to data corruption or dropped connections, especially in suboptimal conditions (e.g., longer wire lengths, electrical noise). 19200 bps generally offers a more robust and dependable communication link.
- **Stability:** For critical functions such as sending SMS, making calls, or transferring data, a stable and predictable communication speed is paramount. 19200 bps provides a good balance of speed and operational stability for these tasks.

### II.6.3 Key Considerations for Successful SIM900 Communication

To guarantee reliable communication between the SIM900 shield and the Arduino, the following key points must be observed:

- **Voltage Level Shifting on RX Pin:** The SIM900 module's RX (Receive) pin typically operates at a 3.3V logic level, whereas Arduino Uno pins output 5V signals. Directly connecting the Arduino's TX pin (5V) to the SIM900 RX pin (3.3V) can damage the module. It is therefore imperative to use a voltage divider circuit or a suitable series resistor to safely step down the 5V signal to 3.3V [16].
- **Network Registration and Compatibility:** Ensure the SIM card is properly inserted, active, and supports 2G GSM networks (as the SIM900 does not support 3G/4G connectivity). It should also have sufficient credit/balance for SMS or data services if planned [10]. Allow adequate time for the SIM900 module to register with the mobile network after power-up, which may take a few seconds. This process is commonly indicated by a specific blinking pattern on an integrated LED, changing once a stable network connection is established [16].
- **Test Communication with AT Commands:** After initial setup, it is crucial to test communication by transmitting simple AT commands (e.g., 'AT' or 'AT+CSQ') using the Arduino Serial Monitor or within your code. Receiving an "OK" response confirms the module is responsive and connected [10].

## II.7 Chapter Summary

In this chapter, we conducted an in-depth exploration of the Arduino platform, emphasizing its extensibility through the use of modular add-ons known as "shields." Particular attention was given to the SIM900 GSM/GPRS shield, including a review of its primary features, relevant technical specifications, and the essential steps for successful integration and communication with the Arduino board. We also clarified key concepts like AT commands, GPRS, and baud rates which are fundamental for its operation.

This foundational understanding of hardware basics is indispensable for anyone intending to implement more advanced embedded systems. In the context of this work, they serve as the foundation for the development of a smart monitoring application—specifically, a power outage detection system.

In the following chapter, we will transition from theory to practice, detailing the systematic construction of the proposed system. We will cover both hardware assembly and software development, including comprehensive testing procedures, clear wiring diagrams, and practical coding examples to ensure the system functions as expected.

## **Chapter III**

# **Development of a Power Outage Detection System Using Arduino and SIM900 Module**

### **III.1 Introduction**

Unpredictable power outages present significant challenges in various regions, particularly for residential and remote facilities. These interruptions regularly interfere with daily life and can cause considerable damage to sensitive electronic equipment [9]. Even short-term outages, when not accurately monitored, may lead to financial losses or equipment damage, often exacerbated by false negative alarms [14].

This challenge necessitates the development of intelligent monitoring systems capable of real-time power status detection and immediate user notification from diverse locations [2]. Remote monitoring not only facilitates immediate action but also enables the logging and statistical analysis of electrical events. Such data can be instrumental in developing improved infrastructure and proactive prevention strategies [3].

To address this critical need, a technical solution was developed utilizing an Arduino UNO microcontroller and a GSM communication module (SIM900). This system continuously monitors electrical current and automatically sends SMS alerts to the user upon detection of a power outage or restoration [7]. This automated approach offers a more practical and reliable alternative to continuous manual supervision [4].

This chapter will provide a comprehensive description of each subsystem, detailing wiring and connection methods, the monitoring algorithm, and experimental results. It will then discuss the effectiveness of the developed system in addressing the aforementioned problem.

### **III.2 System Components**

This section delineates the core hardware components integral to the proposed power outage detection system, emphasizing their individual functionalities and collective roles.

Table III.1: Core Components of the Power Monitoring System

<b>Component</b>	<b>Function</b>	<b>Model/Spec</b>
Arduino UNO	Central controller	ATmega328P
GSM Module	SMS communication	SIM900
Voltage Adapter	Step-down voltage source	AC to 5V/12V
Power Supply	System input	9V–12V DC
Connecting Wires	Signal transmission	Male-to-male jumpers

#### **III.2.1 Arduino UNO Control Board**

The Arduino UNO serves as a widely recognized and fundamental electronic board for embedded systems and automatic control applications. Based on the ATmega328P microcontroller, it offers a broad spectrum of capabilities [13]. Its architecture includes fourteen digital input/output (I/O) pins, six of which support Pulse Width Modulation (PWM), alongside six analog input pins for converting continuous signals.

Additional essential components of the Arduino UNO encompass a USB port for power

### **Chapter III. Development of a Power Outage Detection System Using Arduino and SIM900 Module**

---

supply and programming, a voltage regulator ensuring stable operation, a timing crystal for clock synchronization, a dedicated power jack, and a reset button.

In this project, the Arduino UNO functions as the central control unit, orchestrating the entire system's operation. It continuously monitors the electrical current status via a dedicated digital input pin (specifically, pin 12) [17]. Based on the acquired readings, the Arduino processes information and commands the SIM900 communication module to dispatch SMS messages to the user. Furthermore, the board is programmed to receive and process user-initiated commands sent via SMS (e.g., requests for current system status), enabling dynamic modification of system functionality as required by different operational scenarios [17].

#### **III.2.2 GSM SIM900 Communication Module**

The SIM900 is a prominent GSM module widely employed in mobile wireless communication projects, supporting both GSM and GPRS technologies. It extends the capabilities of microcontrollers, such as the Arduino UNO, by enabling functionalities like sending and receiving SMS (Short Message Service) and initiating or answering phone calls [11].

Communication between the SIM900 module and the Arduino board is established via the UART (Universal Asynchronous Receiver-Transmitter) protocol, utilizing its serial port. The module's comprehensive functionality is controlled by a standardized set of text-based instructions known as AT commands.

Within the proposed system, the SIM900 module is exclusively responsible for handling all external communication related to user alerts. It dispatches automated SMS messages upon the detection of a power outage or restoration event. Additionally, the module is configured to receive and interpret incoming SMS messages from the user containing specific commands (e.g., querying the system's current status), subsequently forwarding these commands to the Arduino for processing. This integration significantly enhances the system's reliability and accessibility by enabling remote monitoring capabilities at anytime and from any location.

#### **GSM Network Communication Mechanism**

The SIM900 module operates akin to a standard mobile phone, requiring a functional SIM card to establish a connection with the nearest local GSM mobile network. Upon receiving an adequate power supply (typically 12V with sufficient current capacity), the module autonomously initiates a network scan and registers itself. The network registration status is visually indicated by a dedicated status LED on the hardware, providing immediate feedback on the module's connection state.

Once successfully registered on the network, the SIM900 is prepared for communication. Its operations are controlled by AT commands transmitted from the Arduino, typically implemented through software serial libraries to manage the communication interface. The combination of UART and AT commands provides a robust and straightforward communication interface between the Arduino microcontroller and the SIM900 module.

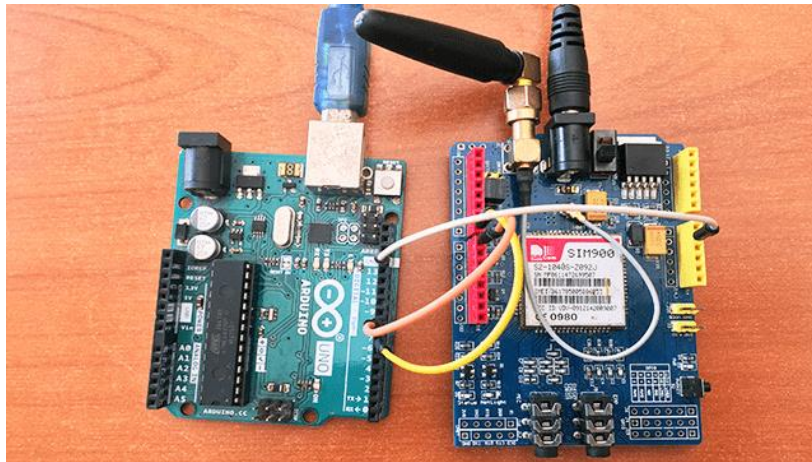


Figure III.1: Sensors in Smart Systems

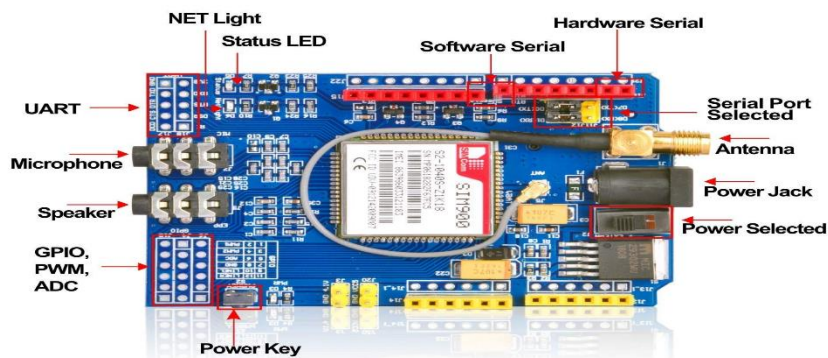


Figure III.2: Communication and power diagram of the SIM900 module

These AT commands empower developers to:

- Enable or disable specific functionalities of the module.
- Configure various operational settings.
- Initiate the sending or receiving of SMS messages.
- Place or receive voice calls.

### **III.2.3 Overall System Interconnections**

This section details the fundamental electrical interconnections between the primary components of the system. While conceptually straightforward, accurate connections are paramount for ensuring proper system behavior and reliable operation.

A robust connection between the Arduino UNO board and the SIM900 communication module must be established, encompassing both power supply and data exchange. The

### Chapter III. Development of a Power Outage Detection System Using Arduino and SIM900 Module

seamless interaction of these two units is critical for dependable real-time monitoring and consistent SMS communication.

The essential interconnections involve:

- Providing a regulated power supply to both the Arduino UNO and the SIM900 module.
- Establishing a serial communication link between the Arduino and the SIM900 module, typically achieved by utilizing specific digital pins via the SoftwareSerial library.
- Maintaining a common and stable Ground (GND) reference across all connected modules to ensure circuit integrity.

These interconnections are visually represented in the following diagrams, facilitating a clear understanding of the module connectivity within the system architecture.

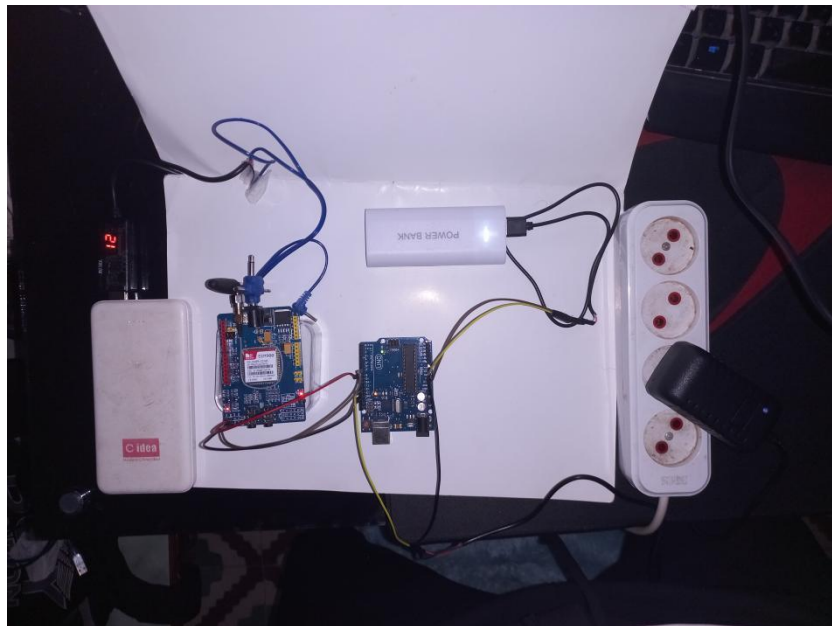


Figure III.3: GSM Arduino Working Prototype

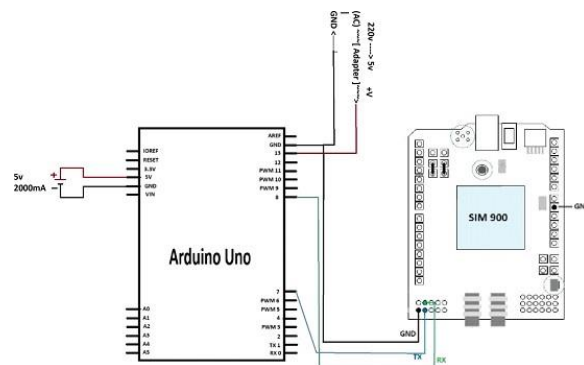


Figure III.4: Arduino UNO SIM900 Wiring Diagram

### **III.2.4 Arduino UNO Pin Configuration**

This subsection details the specific functionalities of the digital and power pins on the Arduino UNO board relevant to the system's operation.

#### **Digital Input/Output (I/O) Pins (0–13)**

The Arduino UNO board features 14 digital I/O pins, numbered from 0 to 13. These pins operate in a binary fashion, capable of assuming one of two discrete logical states at any given moment:

- **High (5V):** Represents a logic bit "1" (active state).
- **Low (0V):** Represents a logic bit "0" (inactive state).

These pins serve two primary purposes:

- **Input Mode:** Used to read digital signals from external devices, such as buttons, motion sensors, or to detect the presence of electrical current from the grid [13].
- **Output Mode:** Used to transmit control signals to actuators, for instance, to illuminate an LED or activate a relay.

**Note on Serial Communication Pins:** Digital pins 0 (RX) and 1 (TX) are directly connected to the onboard USB-to-serial converter. Consequently, it is advisable to avoid utilizing these pins for general-purpose digital I/O when simultaneous serial communication with a host computer or for programming purposes is required. They are primarily reserved for uploading sketches and facilitating serial debugging.

#### **Pulse Width Modulation (PWM) Pins**

On the Arduino UNO, specific digital pins (notably pins 3, 5, 6, 9, 10, and 11) offer Pulse Width Modulation (PWM) capabilities [17]. PWM is a technique that simulates analog output using digital signals by rapidly switching the pin between HIGH and LOW states at a fixed frequency, while varying the proportion of time the signal remains HIGH (known as the "duty cycle"). This method is particularly useful for:

- Dimming Light Emitting Diodes (LEDs).
- Precisely controlling the speed of DC motors.
- Generating specific audio tones.

PWM is especially beneficial in applications where a continuously variable output signal is required, offering finer control than simple ON/OFF states.

### **III.2.5 Power and Ground Pins**

The Arduino UNO board incorporates various power and ground pins, typically grouped

## Chapter III. Development of a Power Outage Detection System Using Arduino and SIM900 Module

---

together, which are essential for supplying power to both the board itself and any connected external components.

### Arduino UNO Power and Ground Pin Functions

- **VIN (Voltage IN):** This input pin allows the board to be powered by an external unregulated power source, such as a battery or wall adapter. The acceptable voltage range typically falls between 7V and 12V [13].
- **5V and 3.3V Pins:** These pins provide regulated voltage outputs (5V and 3.3V, respectively) that can be used to safely power other shields or external low-power electronic components.
- **GND (Ground):** Represents the common electrical ground reference for all components within the circuit, essential for completing electrical paths.
- **IOREF Pin:** This pin outputs the reference voltage of the board (typically 5V for Arduino Uno). It serves as an indicator for connected shields or modules to detect the correct operating voltage level, thereby ensuring safe and compatible operation.

**Important Notice:** It is crucial to note that components designed for lower operating voltages (e.g., 3.3V) may incur damage if directly connected to the 5V pin without appropriate voltage level shifting. The IOREF pin assists connected modules in identifying the safe voltage level for interface.

### III.2.6 SIM900 GPRS Shield Features

The SIM900 GPRS Shield is an add-on module designed to provide Arduino boards with cellular communication capabilities over GSM mobile networks. When integrated with an Arduino, it offers the following essential features:

- **SMS Communication:** Enables the sending and receiving of Short Message Service (SMS) text messages [11].
- **Voice Calling:** Supports the initiation and reception of voice calls, functioning comparably to a standard mobile phone.
- **Internet Connectivity:** Facilitates establishing remote data connections and accessing the Internet via GPRS (General Packet Radio Service) technology, which is suitable for remote monitoring and low-bandwidth data transfer.

### Typical Applications of the SIM900 GPRS Shield

The SIM900 GPRS Shield is particularly well-suited for a variety of applications requiring remote communication and monitoring capabilities, including:

- Remote control and monitoring systems.
- Smart metering and data acquisition systems that rely on SMS or mobile data for transmission.
- Security and alarm systems requiring immediate notification.
- Internet of Things (IoT) sensors deployed in remote locations.

### III.2.7 UART Connection Configuration for SIM900

Accurate configuration of the UART pins is essential for establishing reliable serial communication between the Arduino UNO and the SIM900 shield. Specifically, the RX (Receive) pin of the SIM900 module should be connected to a designated digital pin on the Arduino for data transmission from the Arduino to the SIM900, while the TX (Transmit) pin of the SIM900 module should connect to a different digital pin on the Arduino for data reception by the Arduino.

A common configuration, when using the ‘SoftwareSerial’ library, is as follows:

- Arduino RX (receiving data from SIM900) → SIM900 TX → Arduino Digital Pin 8
- Arduino TX (transmitting data to SIM900) → SIM900 RX → Arduino Digital Pin 7

It is imperative to correctly initialize the ‘SoftwareSerial’ library in the Arduino code to reflect these specific pin assignments, ensuring proper data exchange between the microcontroller and the GSM module.

## III.3 System Functionality

### III.3.1 General System Concept

The proposed system continuously monitors the status of the electrical power supply by sampling a digital input pin on the Arduino, which is interfaced with the power line via a voltage adapter. A HIGH logic state on the input pin signifies the presence of voltage, indicating that the power is ON. Conversely, a LOW logic state denotes the absence of voltage, signaling a power outage [7]. The system is fundamentally designed to interpret the absence of voltage as an indicator of a power supply interruption.

To facilitate real-time monitoring and minimize detection latency, the program utilizes the `digitalRead()` function within the `loop()` function. This function operates at a high frequency, enabling rapid sampling of the input pin and prompt detection of any changes in the power status.

### III.3.2 Power Monitoring Process

The methodology for detecting and monitoring power status involves the following procedural steps:

## Chapter III. Development of a Power Outage Detection System Using Arduino and SIM900 Module

---

1. An Arduino digital input pin (e.g., pin 12) is electrically connected to the output of a low-voltage adapter, which is, in turn, coupled to the main power line.
2. The state of this digital input pin is continuously read using the `digitalRead()` function, executed within the Arduino's primary `loop()` function.
3. The acquired digital state is interpreted based on the following logic:
  - A HIGH state indicates the presence of electrical power.
  - A LOW state signifies a power outage.
4. The duration of power interruptions is precisely measured using the Arduino's `millis()` function, which tracks elapsed time in milliseconds since the Arduino board began running the current program. This allows for quantitative analysis of outage durations.

### III.3.3 SIM900 Automated SMS Notification System

Upon detection of any change in the power status, the Arduino initiates communication with the SIM900 GSM module. This communication is facilitated via AT commands transmitted over the serial connection.

The system is configured to automatically dispatch SMS notifications corresponding to the detected power state:

- When a power outage is identified, an SMS message stating: "The power is OFF." is sent.
- When power is restored, an SMS message stating: "Power is ON." is dispatched.

These critical notifications are transmitted to a pre-configured phone number, ensuring the user receives immediate alerts regarding power status transitions.

### III.3.4 User Interaction via SMS Commands

The system incorporates a remote control capability, enabling user interaction through specific numerical commands sent via SMS. This allows for dynamic querying and control of the system from a remote location.

The defined SMS commands and their corresponding system responses are as follows:

- **Command: 1** Upon receiving this command, the system provides a status response, indicating the current power state:
  - "On Power" is returned if the electrical power supply is active.
  - "The power is off" is returned if a power outage is currently ongoing.
- **Command: 2** This command prompts the system to respond with a statistical overview of recent power interruptions, specifically a bar graph format displaying the number of outages within the preceding 24-hour period:

## Chapter III. Development of a Power Outage Detection System Using Arduino and SIM900 Module

- "The number of cuts in the last 24h: X" where X represents the total count of registered power interruptions.

This robust communication model facilitates effective and user-friendly remote monitoring of the power supply status [3].

Table III.2: User SMS Commands and System Responses

Command	Description	System Response
1	Request current power status	On Power or Power is OFF
2	Request outage stats (last 24h)	Number of cuts: X

### III.4 System Programming and Implementation

The software architecture for the power outage detection system is developed in C/C++ utilizing the Arduino Integrated Development Environment (IDE). The implementation leverages standard Arduino libraries for interfacing with various hardware components, incorporating fundamental functions such as pin Mode () for configuring pin roles and digital Write () for controlling digital outputs. The overarching objective of the program is to generate and transmit alert messages in response to detected irregularities in the electrical current. The developed program was directly uploaded to the Arduino UNO board via the Arduino IDE environment for comprehensive testing and validation.

#### III.4.1 Code Structure and Initialization

##### Defining Libraries and Global Variables

##### Objective: Establishing SIM900 and Arduino UNO Communication

A primary objective in the initial programming phase is to establish robust serial communication between the SIM900 module and the Arduino UNO. This is achieved through the instantiation of a SoftwareSerial object: This configuration assigns Arduino

```
1 #include <SoftwareSerial.h>
2
3 SoftwareSerial SIM900(7, 8);
4 bool smsSent = false;
5 bool powerState = true; // true = power ON, false = power OFF
6
7 #define POWER_PIN 12
8
9
10 unsigned long powerCutTimes[50]; // لتخزين أوقات الانقطاع (بـ millis())
11 int powerCutCount = 0;
12 unsigned long lastMessageTime = 0; // وقت آخر رسالة تم إرسالها
13 unsigned long messageDelay = 60000; // تأخير 1 دقيقة بين الرسائل الأخرى (بـ millis())
14
```

Figure III.6: Code Snippet: SoftwareSerial Initialization for SIM900 Communication

digital pins 7 and 8 for the software-defined serial port, with pin 7 designated for TX (transmit data from Arduino to SIM900) and pin 8 for RX (receive data from SIM900 to Arduino) [11].

**System Status Tracking and Power Monitoring Variables** To preclude the redundant transmission of notification messages, a variable linked to digital pin 12 is utilized to continuously read the current status of the power adapter. The system then records the power state as follows:

- HIGH: Indicates the availability of electrical power.
- LOW: Denotes a power shutdown or outage.

**Message Duplication Prevention Mechanism** The boolean variable smsSent is implemented as a critical flag to prevent repetitive SMS transmissions within the continuous execution cycle of the loop() function. Without this mechanism, the system could send identical alert messages incessantly, leading to excessive consumption of the SIM card's credit and potential service interruption [7].

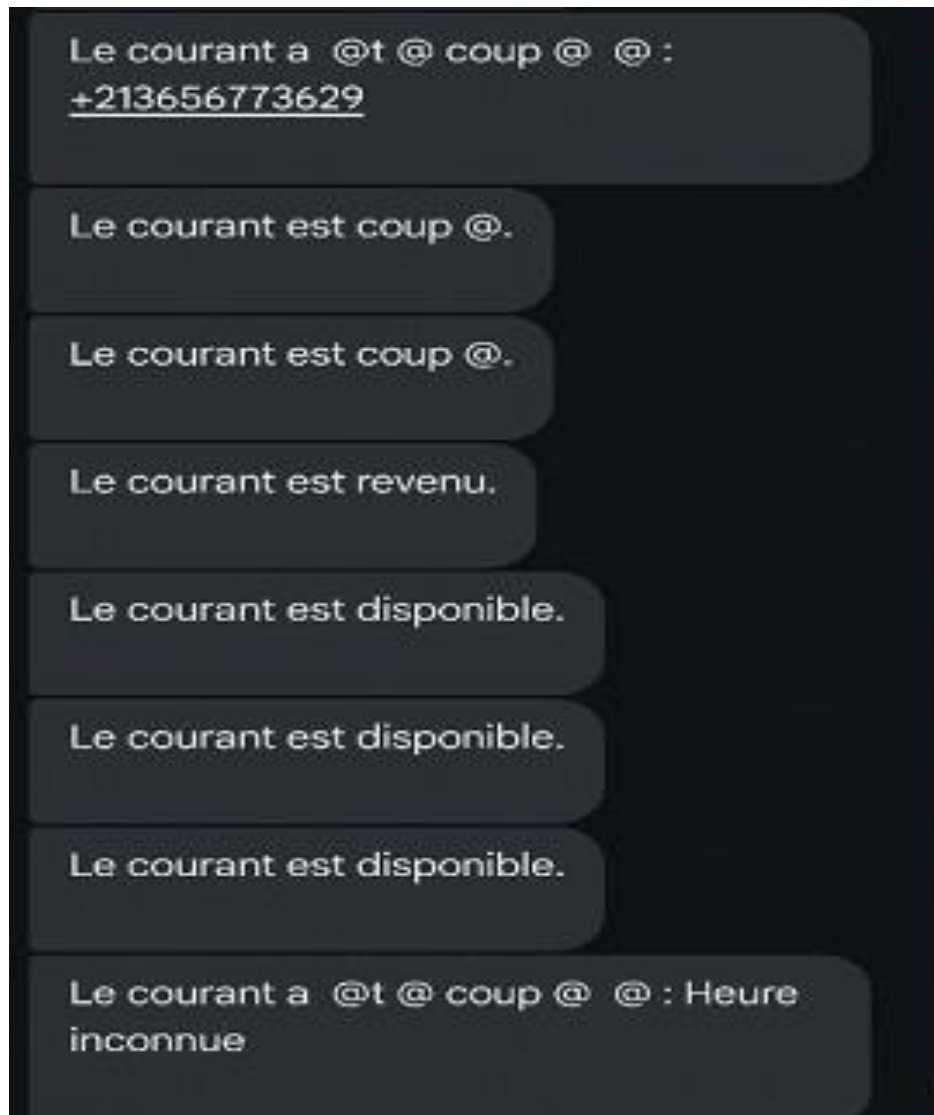


Figure III.7: Code Snippet: SMS Power Status Variables and Logic

### The setup() Function: System Initialization

The setup() function is a fundamental component of Arduino programs, executed precisely once upon the Arduino's power-on or reset event. Its primary role is to perform essential system initializations and configurations, preparing the entire system for subsequent operational tasks [13].

```
15 void setup() {
16
17     pinMode(POWER_PIN, INPUT);
18
19     Serial.begin(9600);
20     SIM900.begin(19200);
21     delay(30000);
22
23     // تفعيل استقبال الوقت من الشبكة
24     SIM900.println("AT+CLTS=1");
25     delay(1000);
26     SIM900.println("AT&W");
27     delay(1000);
28
29     // تفعيل استقبال الرسائل
30     SIM900.println("AT+CMGF=1"); // Text mode
31     delay(1000);
32     SIM900.println("AT+CNMI=1,2,0,0,0"); // عرض الرسائل مباشرة
33     delay(1000);
34
35     Serial.println("Ready.");
36 }
```

Figure III.8: Code Snippet: Arduino GSM Setup Code with AT Commands

The specific purposes of this initialization stage include:

- **Power Input Configuration:**

- The digital input pin, designated as POWERPIN (Pin 12), is configured to receive the electrical current status from the power supply and detect any interruptions or outages in real-time

- **SIM900 Module Initialization:**

- The serial communication interface to the SIM900 module is initialized, ensuring data exchange readiness.

**Serial Communication Baud Rate Configuration:** The baud rates for both the serial monitor (for debugging and user interaction) and the SIM900 module communication are precisely configured, typically set at either 9600 bps or 19200 bps (bits per second) to ensure synchronized data transmission and reception [11].

**AT Command Configuration for SIM900:** During the setup() phase, a series of AT commands are transmitted to the SIM900 module to configure its operational parameters:

1. AT+CLTS=1: This command instructs the SIM900 module to retrieve and synchronize its internal clock with the time provided by the cellular network.
2. AT+CMGF=1: This command configures the module into Text Mode (as opposed to PDU mode) for sending and receiving SMS messages, simplifying message handling.

3. AT+CNMI=1,2,0,0,0: This command configures how the SIM900 module handles incoming SMS messages, typically setting it to display new messages directly on the serial interface.

```
SIM900.println("AT+CMGF=1"); // Text mode
delay(1000);
SIM900.println("AT+CNMI=1,2,0,0,0"); // عرض الرسائل مباشرة
delay(1000);
```

Figure III.9: Code Snippet: AT Command Configuration for SMS and Time Synchronization

Table III.3: Key AT Commands for SIM900 Configuration

Command	Purpose	Expected Response
AT+CMGF=1	Set SMS mode to Text	OK
AT+CNMI=1,2,0,0,0	SMS notification config	OK
AT+CLTS=1	Enable network time sync	OK
AT+CSQ	Signal quality check	+CSQ: X,Y

**Note:** It is important to acknowledge that AT commands are specific instructions for the SIM900 module and their syntax or availability may vary slightly between different GSM shield manufacturers or firmware versions.

### III.4.2 The loop () Function: Continuous Operation

The loop () function is the core execution block of any Arduino program, operating continuously in an infinite cycle immediately after the setup () function has completed its execution. This function is responsible for managing all ongoing tasks, monitoring system performance, and ensuring the system remains active and responsive to various inputs and changes in conditions [9].



Figure III.10: Flowchart to make loop ()

```
if (buttonState == LOW && powerState && !smsSent) {  
    // انقطاع التيار  
  
    powerState = false;  
  
    String time = getTime();  
    sendSMS("Le courant a ete coupe a : " + time);  
    smsSent = true; // تأكد من أننا أرسلنا الرسالة فقط مرة واحدة
```

Figure III.11: detect power outage and send alerts

```
else if (buttonState == HIGH && !powerState) {  
    // عودة التيار  
  
    powerState = true;  
  
    // إرسال رسالة عند عودة التيار فقط إذا لم يتم إرسالها من قبل  
    if (smsSent) {  
        sendSMS("Le courant est revenu.");  
        smsSent = false; // عندما يعود التيار smsSent إعادة تعيين  
    }  
}
```

Figure III.12: Sensors in Smart Systems

**Key Operational Stage:** This stage represents the most critical segment of the system's operation, as it directly processes and responds to the detected state of the electrical current [7]. The activities within this stage include:

- Continuously reading the digital state of the power input pin and storing it in a variable (e.g., ButtonState).
- Accurately tracking the elapsed time using the millis() function, which is essential for measuring power interruption durations and implementing time-based logic within the system [13].

## III.5 System Functionality

### III.5.1 General System Concept

The proposed system continuously monitors the status of the electrical power supply by sampling a digital input pin on the Arduino. This pin is interfaced with the power line through a voltage adapter. When electrical voltage is present, the input pin registers

a HIGH logic state, indicating that the power is ON. Conversely, the absence of voltage results in the pin reading a LOW state, signaling a power outage [7]. Fundamentally, the system interprets the absence of voltage as a definitive indicator of a power supply interruption.

To ensure real-time monitoring and minimize detection latency, the program utilizes the `digitalRead()` function, executed repeatedly within the Arduino's main `loop()` function. This high-frequency sampling approach enables the system to acquire multiple readings swiftly, thereby minimizing any delays in detecting dynamic changes in the power status.

### III.5.2 Power Monitoring Process

The methodology for detecting and monitoring the power status follows a structured, sequential process:

1. An Arduino digital input pin (e.g., pin 12) is electrically connected to the output of a low-voltage adapter, which is coupled to the main power line.
2. The logical state of this configured digital input pin is continuously read using the `digitalRead()` function, which is embedded within the Arduino's `loop()` function for perpetual execution.
3. The acquired digital state is interpreted according to the following logic:
  - A HIGH logic state signifies the active presence of electrical power.
  - A LOW logic state indicates a power outage.
4. The duration of power interruptions is precisely measured and tracked utilizing the Arduino's intrinsic `millis()` function. This function provides the elapsed time in milliseconds since the Arduino board commenced program execution, enabling accurate quantification of outage periods for analytical purposes.

### III.5.3 SIM900 Automated SMS Notification System

Upon the detection of any alteration in the power status (either an outage or restoration), the Arduino initiates communication with the SIM900 GSM module. This inter-module communication is facilitated through the transmission of `**AT commands**` over the established serial connection.

The system is configured to automatically dispatch SMS notifications, tailored to the detected power state:

- When a power outage is identified, a predefined SMS message stating: "The power is OFF." is sent to the designated recipient.
- Conversely, upon the detection of power restoration, an SMS message stating: "Power is ON." is dispatched.

These critical notifications are transmitted to a pre-configured phone number, ensuring the user receives immediate and relevant alerts concerning power status transitions. This automated alert mechanism is crucial for timely intervention and situational awareness, particularly in remote monitoring scenarios.

### **III.5.4 User Interaction via SMS Commands**

The system incorporates robust remote control capabilities, allowing user interaction through the transmission of specific numerical commands via SMS. This feature enables dynamic querying and control of the system from any remote location with cellular network access.

The defined SMS commands and their corresponding system responses are as follows:

- **Command: ‘1’**

Upon receiving this command, the system provides a comprehensive status response, articulating the current power state:

- "On Power" is returned if the electrical power supply is actively present.
- "The power is off" is returned if a power supply interruption is currently ongoing.

- **Command: ‘2’**

This command prompts the system to respond with a statistical overview of recent power interruptions. The response includes a numerical count of outages within the preceding 24-hour period:

- "The number of cuts in the last 24h: X" where ‘X’ precisely represents the total count of registered power interruptions within the specified timeframe.

This robust and user-friendly communication model facilitates highly effective remote monitoring of the power supply status [3], offering both real-time alerts and historical data on power events.

## **III.6 System Programming and Implementation**

The software for the power outage detection system is developed in **C/C++** utilizing the **Arduino Integrated Development Environment (IDE)**. The implementation extensively employs standard Arduino libraries for establishing communication with various peripheral devices, incorporating fundamental functions such as `pinMode()` for configuring digital pin roles and `digitalWrite()` for controlling output states. The overarching objective of the programmed logic is to detect anomalies in the electrical current and subsequently trigger alert messages to the user. The developed program was directly uploaded to the Arduino UNO board through the Arduino IDE environment for comprehensive testing, debugging, and validation.

### **III.6.1 Code Structure and Initialization Phase**

#### **Defining Libraries and Global Variables**

#### **Objective: Establishing SIM900 and Arduino UNO Communication**

A primary objective in the initial programming phase is the establishment of robust serial communication between the SIM900 module and the Arduino UNO. This is achieved through the instantiation of a `SoftwareSerial` object, which emulates a hardware serial port using general-purpose digital pins. This specific configuration designates Arduino

```
1 #include <SoftwareSerial.h>
2
3 SoftwareSerial SIM900(7, 8);
4 bool smsSent = false;
5 bool powerState = true; // true = power ON, false = power OFF
6
7 #define POWER_PIN 12
8
9
10 unsigned long powerCutTimes[50]; // لتخزين أوقات الانقطاع (بـ millis())
11 int powerCutCount = 0;
12 unsigned long lastMessageTime = 0; // وقت آخر رسالة تم إرسالها
13 unsigned long messageDelay = 60000; // تاخير 1 دقيقة بين الرسائل الأخرى (بـ millis())
14
```

Figure III.13: Code Snippet: SoftwareSerial Initialization for SIM900 Communication

digital pins 7 and 8 for the software-defined serial port: pin 7 serves as the **\*\*TX\*\*** (transmit data from Arduino to SIM900), and pin 8 functions as the **\*\*RX\*\*** (receive data from SIM900 to Arduino) [11]. This allows the hardware serial port (pins 0 and 1) to remain available for debugging and communication with the Arduino IDE's Serial Monitor.

**System Status Tracking and Power Monitoring Variables** To preclude the redundant transmission of notification messages, a variable is assigned to capture the digital state of pin 12, which continuously reads the current generated by the power adapter. The system subsequently records the power state as:

- **HIGH:** Signifies the availability of electrical power.
- **LOW:** Denotes a power shutdown or outage.

**Message Duplication Prevention Mechanism** The boolean variable `smsSent` is critically implemented as a flag to prevent the repetitive dispatch of SMS notifications within the continuous execution cycle of the `loop()` function. Without this preventative measure, the system could inadvertently send identical alert messages incessantly, leading to excessive consumption of the SIM card's credit and potential service interruption [7]. This flag is toggled only when a new power state change is detected, ensuring that an SMS is sent only once per change.

### The `setup()` Function: System Initialization

The `setup()` function is a fundamental component of any Arduino program. It executes precisely once upon the Arduino's power-on or reset. Its primary role is to perform all essential system initializations and configurations, preparing the entire embedded system for its subsequent operational tasks [13].

```
15 void setup() {
16
17     pinMode(POWER_PIN, INPUT);
18
19     Serial.begin(9600);
20     SIM900.begin(19200);
21     delay(3000);
22
23     // تفعيل استقبال الوقت من الشبكة
24     SIM900.println("AT+CLTS=1");
25     delay(1000);
26     SIM900.println("AT&W");
27     delay(1000);
28
29     // تفعيل استقبال الرسائل
30     SIM900.println("AT+CMGF=1"); // Text mode
31     delay(1000);
32     SIM900.println("AT+CNMI=1,2,0,0,0,0"); // عرض الرسائل مباشرة
33     delay(1000);
34
35     Serial.println("Ready.");
36 }
```

Figure III.15: Code Snippet: Arduino GSM Setup Configuration with AT Commands

The specific objectives of this initialization stage are:

- **Power Input Configuration:**

- The digital input pin, designated as POWER\_PIN (Pin 12), is configured using `pinMode()` to receive the electrical current signal from the voltage adapter.
- The system's instantaneous power state is subsequently monitored and observed through this designated POWER\_PIN (Pin 12).

- **SIM900 Module Initialization:**

- The serial communication interface (via SoftwareSerial) to the SIM900 module is initialized, establishing the necessary data exchange readiness.

**Serial Communication Baud Rate Configuration:** The baud rates for both the hardware serial monitor (used for debugging and user interaction) and the software serial interface to the SIM900 module are precisely configured. Typical rates, such as **9600 bps** (bits per second) for the Serial Monitor and **19200 bps** for the SIM900 module, are set to ensure synchronized and reliable data transmission and reception [11].

**AT Command Configuration for SIM900 Module:** During the `setup()` phase, a sequence of **AT commands** is transmitted to the SIM900 module to configure its operational parameters for optimal performance within the system:

1. **AT+CLTS=1:** This command instructs the SIM900 module to retrieve and synchronize its internal real-time clock (RTC) with the precise time provided by the cellular network. This is crucial for accurate timestamping of power events.
2. **AT+CMGF=1:** This command configures the SIM900 module to **Text Mode** (as opposed to PDU mode) for sending and receiving SMS messages. Text mode simplifies message handling by allowing direct string manipulation.

3. AT+CNMI=1,2,0,0: This command configures the SIM900 module's behavior regarding incoming SMS messages. Specifically, it typically sets the module to direct new messages to the serial interface, enabling the Arduino to read them.

```
SIM900.println("AT+CMGF=1"); // Text mode
delay(1000);
SIM900.println("AT+CNMI=1,2,0,0,0"); // عرض الرسائل مباشرة
delay(1000);
```

Figure III.16: Code Snippet: AT Command Configuration for SMS Mode and Network Time Synchronization

**Note:** It is imperative to acknowledge that AT commands are proprietary instructions specific to GSM/GPRS modules, and their precise syntax or availability may exhibit slight variations between different SIM900 shield manufacturers or module firmware versions. Developers should consult the respective module's datasheet for definitive command specifications.

### III.7 Conclusion

The presented project successfully developed and implemented an intelligent system for remotely monitoring power distribution status. This system effectively leverages the capabilities of the Arduino UNO microcontroller for control and processing, and the SIM900 GSM module for robust cellular communication [13]. The core functionality of the system includes automatic detection of power outages and restorations via a sensor connected to a digital input pin, followed by the immediate transmission of SMS notifications to a predefined mobile number to alert the user.

During the development phase, a comprehensive software script was implemented to manage the detection of both power cuts and re-energization events. Furthermore, the system was configured to process incoming SMS commands, enabling users to remotely query the current power status or retrieve a historical log of power outages within a specified timeframe. The SIM900 module's ability to reliably process AT commands was instrumental in ensuring consistent communication with the mobile network [11].

Experimental validation confirmed the system's effective performance. It accurately detected all simulated power outages and subsequent restorations, reliably dispatching appropriate SMS notifications for each event. A minor observable difference was noted in the system's response time, with power outage detection exhibiting a slightly longer latency compared to power restoration. This discrepancy is primarily attributed to the methodology employed for power supply simulation during the testing phase.

Despite these minor performance variations, the system demonstrably fulfilled all its original project objectives, proving its reliability and effectiveness. Consequently, the

proposed solution stands as a viable and cost-effective approach for remote electrical current monitoring, utilizing readily accessible and mature technologies.

## **III.8 Future Enhancements**

To further augment the system's performance, robustness, and user experience, several key areas for future improvement have been identified:

- 1. Integration of a Real-Time Clock (RTC) Module:** Incorporating a dedicated RTC module would ensure highly accurate timekeeping for event logging, independent of network availability or Arduino resets. This would provide precise timestamps for all power status changes, even during prolonged network outages, which is critical for detailed historical analysis and auditing.
- 2. Upgrade of the Power Supply Unit (PSU):** Improving the system's power supply infrastructure, particularly for the SIM900 module, is recommended. A more robust and stable PSU capable of delivering consistent current, especially during peak transmission periods, would likely enhance the system's response time to power outages and improve overall operational stability.
- 3. Implementation of an Optional Data Storage Module:** Integrating an SD card module or similar non-volatile memory would enable comprehensive logging of all power events (outages, restorations, durations) directly on the device. This local data storage would provide a persistent record, serving as a valuable backup and allowing for more extensive long-term analysis, even if network communication is intermittently unavailable.
- 4. Development of a Sophisticated User Interface (UI):** Creating a dedicated mobile application or a web-based dashboard would significantly enhance user interaction and data visualization. Such an interface could provide real-time graphical representations of power status, historical trends, cumulative outage statistics, and offer more intuitive control mechanisms than SMS commands, thereby improving usability and accessibility. This could leverage cloud platforms (e.g., Firebase, AWS IoT) for data storage and retrieval.

Overall, this project establishes a solid foundational framework for future developments in remote monitoring and control systems. It presents significant opportunities for further research and practical application, particularly within the evolving contexts of smart home automation and industrial monitoring solutions.



# **General Conclusion and Perspectives**

### **General Conclusion**

In the contemporary landscape of pervasive technological integration, the development of intelligent systems has become paramount, particularly in enhancing the safety and operational efficiency of residential and commercial environments. This project specifically addressed the prevalent challenge of unscheduled power outages, phenomena that extend beyond mere inconvenience to significantly disrupt daily routines and potentially inflict damage upon sensitive electronic infrastructure. Consequently, the imperative for an autonomous, real-time power status monitoring system became evident.

Our solution materialized as a robust and accessible system, architected around the Arduino UNO microcontroller and the SIM900 GSM communication module. This synergistic integration enabled the creation of a sophisticated yet practical framework capable of continuously monitoring the electrical power supply and promptly dispatching SMS notifications upon detecting any change in its status. This automated approach represents a significant advancement over conventional manual inspection methods, offering enhanced reliability and considerable peace of mind to users.

Chapter II provided a foundational exposition on the architectural and functional principles of the Arduino platform. We meticulously detailed the core components, including the ATmega328P microcontroller, its programming paradigms, and the diverse functionalities offered by various digital and analog pins. Furthermore, the discussion extended to the seamless integration of peripheral shields, specifically highlighting how the SIM900 module facilitates advanced communication features through straightforward interfacing. This systematic breakdown laid the groundwork for understanding the hardware layer upon which our system is built.

Subsequently, Chapter III transitioned into the practical implementation of our power outage detection system. This section delineated the methodology by which the Arduino continuously monitors the power supply, detailing the logic that triggers automated SMS notifications in response to observed changes. Crucially, the system was designed with an emphasis on user interaction; users can send simple text commands to ascertain the current power state or to retrieve a historical record of past outages, thus enhancing its utility beyond mere notification.

The programming section provided an in-depth analysis of the code's structure, including variable definitions, function implementations, and the debugging procedures employed during development. The efficacy of the system was rigorously validated through experimental testing, which unequivocally demonstrated its successful operation. The system accurately detected all simulated power interruptions and subsequent restorations, consistently dispatching timely alerts. While a marginal latency was observed in the detection of power outages, primarily attributable to the specific method of power supply simulation during testing, the system nevertheless achieved all its primary design objectives with commendable reliability.

## **General Conclusion and Perspectives**

---

In essence, this project transcends the mere creation of a functional device. It culminates in the realization of a viable and economically attractive remote power monitoring solution, built entirely upon readily available commercial off-the-shelf technologies. This accomplishment not only addresses a common logistical challenge with practical efficacy but also establishes a foundational framework for future advancements in smart systems, industrial monitoring, and generalized remote control applications.

### Future Enhancements and Perspectives

Building upon the established foundation, several avenues for future research and development are envisioned to further augment the system's performance, robustness, and user experience:

- 1. Integration of a Real-Time Clock (RTC) Module:** To ensure highly precise and continuous timekeeping, especially during periods of network unavailability or system resets, the integration of a dedicated RTC module is imperative. This would enable accurate timestamping of all power events, critical for detailed historical analysis and auditing, independent of cellular network time synchronization.
- 2. Upgrade of the Power Supply Unit (PSU):** An enhanced and more stable power supply unit, particularly for the power-intensive SIM900 module, is recommended. A PSU capable of providing consistent voltage and sufficient current, especially during peak transmission phases, would likely mitigate the observed detection latency for power outages and contribute to overall system stability and reliability.
- 3. Implementation of an Optional Data Storage Module:** Incorporating a non-volatile data storage solution, such as an **SD card module**, would enable the comprehensive logging of all power events (outages, restorations, durations) directly on the device. This local data repository would serve as a robust backup, providing a persistent record for more extensive long-term analysis, even in scenarios where continuous network connectivity is compromised.
- 4. Development of a Sophisticated User Interface (UI):** To significantly enhance user interaction and data visualization, the development of a dedicated mobile application or a web-based dashboard is proposed. Such an interface could provide real-time graphical representations of power status, historical trends, cumulative outage statistics, and offer more intuitive control mechanisms than text-based SMS commands. This would leverage modern **cloud platforms** (e.g., Firebase, AWS IoT) for scalable data storage and seamless retrieval, fundamentally transforming the user's engagement with the system.

- [1] Arduino Documentation. *Arduino SoftwareSerial Library*. <https://www.arduino.cc/en/Reference/softwareSerial>. 2024.
- [2] Arduino Project Hub. *Getting Started with Arduino Uno*. <https://create.arduino.cc/projecthub>. 2023.
- [3] Yusuf A. Badamasi. “The working principle of an Arduino.” In: *Proceedings of the 11th International Conference on Electronics, Computer and Computation (ICECCO) (2014)*, pp. 1–4.
- [4] Massimo Banzi and Michael Shiloh. *Getting Started with Arduino*. 3rd. Maker Media, Inc., 2014.
- [5] Rohith Bloor, Chris Bowen, and Westley Weimer. “Arduino based interactive LED system.” In: *Proceedings of the 2015 IEEE International Symposium on Technologies for Homeland Security (HST)*. IEEE, 2015, pp. 1–6.
- [6] J. M. Hughes Cameron. *Arduino: A technical reference*. O’Reilly Media, Inc., 2019.
- [7] Diane J. Cook and Sajal K. Das. *Smart Environments: Technology, Protocols and Applications*. John Wiley & Sons, 2012.
- [8] Jayavardhana Gubbi, Marimuthu Palaniswami, and Slaven Marusic. “The Internet of Things in building automation and energy management.” In: *Smart Buildings Journal* 4.2 (2017), pp. 63–74.
- [9] Jayavardhana Gubbi et al. “Internet of Things (IoT): A vision, architectural elements, and future directions.” In: *Future Generation Computer Systems* 29.7 (2013), pp. 1645–1660.
- [10] Lei Leekongxue, Michael Choi, and Hao Wang. “Smart irrigation system using GSM and Arduino.” In: *International Journal of Advanced Computer Science and Applications* 11.5 (2020), pp. 475–480.
- [11] Abdul Majid Mirza et al. “Mind-controlled Arduino robot using an EEG headset.” In: *International Journal of Engineering Research and General Science* 3.4 (2015), pp. 45–55.
- [12] Adebayo Oluwole, Anthony Uche, and Abayomi Samuel. “Applications of GSM module in embedded system projects.” In: *Journal of Engineering and Applied Sciences* 16.7 (2021), pp. 155–162.
- [13] Avik Paul and Arijit Kundu. “A survey on smart home automation: Architectures, technologies, and future directions.” In: *Journal of Ambient Intelligence and Smart Environments* 11.5 (2019), pp. 387–418.

## Bibliographie

---

- [14] Sourav Paul and Sugata Sanyal. “Survey on home automation using IoT.” In: *International Journal of Advanced Research in Computer Science and Software Engineering* 6.6 (2016), pp. 115–119.
- [15] Md Saiful Rahman et al. “Development of a weather monitoring system using Arduino and GSM module.” In: *2018 International Conference on Computer, Communication, Chemical, Material and Electronic Engineering (IC4ME2)*. IEEE, 2018, pp. 1–4.
- [16] SIMCom Wireless Solutions. *SIM900 Quad-band GSM/GPRS Module Datasheet*. [https://cdn.sparkfun.com/datasheets/Cellular/GSM/GPRS\\_SIM900.pdf](https://cdn.sparkfun.com/datasheets/Cellular/GSM/GPRS_SIM900.pdf). 2015.
- [17] Guillaume Simonet. “Domotics: Smart homes and the automation of domestic space.” In: *Journal of Smart Technologies* 12.3 (2021), pp. 101–119.

# Abstract

Unscheduled power outages pose considerable challenges in sectors like healthcare and industry, where an uninterrupted power supply is crucial. These disruptions can lead to equipment failure, operational halts, and significant safety risks. This Dissertation details the design and implementation of a cost-effective, real-time system for detecting and notifying about power outages.

The system's architecture leverages an Arduino Uno microcontroller integrated with a SIMCom SIM900 GSM shield. This configuration allows for the continuous monitoring of a power supply and automates the dispatch of SMS notifications when the power status changes.

Chapter 3 provides experimental validation of the system's performance and reliability. During controlled tests, the system effectively sent a "Power Off" notification with an average delay of 37 seconds, and a "Power Restored" notification within 7 seconds. The system also demonstrated interactive functionality, responding to SMS-based status queries in less than 30 seconds. A minor limitation identified during simulated outages was the display of an "Heure inconnue" (Unknown Time) message, stemming from a lack of network time synchronization.

These findings affirm the proposed system as a viable and quantifiable solution for remote power monitoring, making it well-suited for deployment in safeguarding critical infrastructure.

---

**Keywords :** Arduino Uno; SIM900 GSM; power outage detection; remote monitoring; Internet of Things; embedded system.

---

# Résumé

Les coupures de courant imprévues représentent des défis majeurs dans des secteurs critiques comme la santé et l'industrie, où la continuité de l'alimentation électrique est essentielle. Ces interruptions peuvent entraîner des défaillances d'équipements, des arrêts opérationnels et des risques de sécurité importants. Cette thèse présente la conception et la mise en œuvre d'un système basé sur un Arduino Uno et un shield GSM SIM900, capable de détecter les pertes de puissance et d'envoyer automatiquement des alertes par SMS.

Le Chapitre 3 détaille les résultats expérimentaux. Lors de trois tests consécutifs de coupure de courant, le système a envoyé le message « Power Off » avec un délai moyen de 37 secondes. Pour cinq essais de rétablissement, le message « Power Restored » a été transmis en 7 secondes. De plus, le système a démontré sa capacité à répondre à une demande de statut par SMS en moins de 30 secondes. Une limitation a été observée durant une panne simulée de 30 secondes, où le système a affiché « Heure inconnue » en raison d'un manque de synchronisation d'horloge réseau.

Ces résultats confirment que le système proposé constitue une solution concrète et mesurable, prête à être déployée sur site pour la protection des infrastructures critiques.

**Mots clés :** Arduino Uno ; SIM900 GSM ; détection de panne de courant ; surveillance à distance ; Internet des objets; Système embarqué.

## ملخص

تشكل انقطاعات التيار الكهربائي غير المخطط لها تحديات كبيرة في قطاعات حيوية مثل الرعاية الصحية والصناعة، حيث تُعد استمرارية التيار الكهربائي أمراً بالغ الأهمية. يمكن لهذه الانقطاعات أن تؤدي إلى أعطال في المعدات، وتوقف العمليات، ومخاطر جسيمة على السلامة. تقدم هذه الأطروحة تفصيلاً لتصميم وتنفيذ نظام فعال من حيث التكلفة وفي الوقت الفعلي للكشف عن انقطاع التيار الكهربائي وإرسال الإشعارات.

يُصمم النظام بالاعتماد على متحكم دقيق أردوينو Uno مدمجاً مع درع GSM. SIM900 SIMCom يمكن هذا التكوين من المراقبة المستمرة لمصدر الطاقة ويسهل الإرسال التلقائي للإشعارات عبر الرسائل القصيرة (SMS) عند اكتشاف أي تغيير في حالة الطاقة.

يعرض الفصل الثالث من هذه الأطروحة النتائج التجريبية التي تؤكد أداء النظام وموثوقيته. ففي سلسلة من الاختبارات المراقبة، نجح النظام في إرسال إشعار "انقطاع التيار" بمتوسط تأخير بلغ 37 ثانية، وإشعار "استعادة التيار" في غضون 7 ثوانٍ. كما أظهر النظام قدرات تفاعلية، حيث استجاب لاستعلامات الحالة عبر الرسائل القصيرة في أقل من 30 ثانية. وقد تم تحديد قيود بسيطة خلال الانقطاعات المحاكاة، حيث ظهرت رسالة "Heure inconnue" (وقت غير معروف) بسبب عدم وجود مزامنة وقت الشبكة.

تؤكد هذه النتائج أن النظام المقترح يُعد حلاً عملياً وقابلًا للقياس لمراقبة الطاقة عن بعد، مما يجعله مناسباً للنشر الميداني لحماية البنى التحتية الحيوية.

## كلمات مفتاحية :

نظام مضمن، Arduino، SIM900 Uno، GSM، إنترنت الأشياء، كشف انقطاع التيار الكهربائي والمراقبة عن

بعد