

Design of a Security System to Monitor Equipment Performance, Diagnose Anomalies, Predict Failures, and Optimize Operations in Isolated GSM Sites through Sensor Technology: Case of GSM Sites in Burkina Faso

Yanogo Kiswendsida Jean Hermann¹, Kahoun Zita Phillipe², Rachid Gaetan Nabollé³, Ouedraogo Tounwendyam Frederic⁴

¹Institute of Computer Engineering and Telecommunication Polytechnic, Ouagadougou, Burkina Faso

²University of Aube Nouvelle, Ouagadougou, Burkina Faso

³University of Lisala, Lisala, Congo

⁴Norbert Zongo University, Koudougou, Burkina Faso

Email: yanogohermann@yahoo.fr

How to cite this paper: Hermann, Y.K.J., Phillipe, K.Z., Nabollé, R.G. and Frederic, O.T. (2026) Design of a Security System to Monitor Equipment Performance, Diagnose Anomalies, Predict Failures, and Optimize Operations in Isolated GSM Sites through Sensor Technology: Case of GSM Sites in Burkina Faso. *Journal of Sensor Technology*, 16, 1-12.

<https://doi.org/10.4236/jst.2026.161001>

Received: December 18, 2025

Accepted: March 14, 2026

Published: March 17, 2026

Copyright © 2026 by author(s) and Scientific Research Publishing Inc.

This work is licensed under the Creative Commons Attribution International License (CC BY 4.0).

<http://creativecommons.org/licenses/by/4.0/>



Open Access

Abstract

The aim of this research is to design a security system capable of monitoring, diagnosing anomalies, predicting failures, and optimizing operations in isolated GSM sites using sensor systems. In remote areas of Burkina Faso, the reliability and security of telecommunications infrastructure is crucial to ensure continuous and efficient connectivity. Operators use GSM stations to provide communication services across the country, even in the most remote areas. To maintain the performance and integrity of equipment, the implementation of supervision and control systems is essential. This research focuses on the physical security of equipment and is important because it will allow monitoring and having real-time information on isolated GSM infrastructures with limited energy sources. We use Arduino, a SCADA system and temperature sensors to design the system architecture and C++ language for coding. The expected result is the implementation of a sensor-based security system to allow administrators to be always alert via a signal relay from GSM sites to the telephone operator.

Keywords

Computer System, Supervision, Security Management, Real-Time Alert, Security System

1. Introduction

SCADA systems are sophisticated technological solutions that enable remote monitoring, control, and management of equipment. By combining various technologies such as sensors, these systems collect real-time data on equipment operation, enabling rapid intervention in the event of failure or malfunction. This continuous monitoring is particularly valuable for GSM stations located in hard-to-reach areas, where manual intervention is often complicated and costly. A SCADA system can control processes remotely, monitor, collect and process data in real time and interact directly with devices (sensors etc.) using human-machine interface (HMI) software. The system also logs events on servers. In addition, a SCADA system can trigger alarms in the event of dangerous situations in facilities. These systems are used to distribute data to enable informed decisions and communicate system issues to reduce downtime.

2. Issues Related to the Management of GSM Sites

Isolated telecommunications sites frequently experience power outages that can severely impact the quality of service provided. These outages can stem from a multitude of factors, often related to the remote nature of these sites and the specific constraints of their power supply. Outages are frequently multifactorial and can be difficult to predict. A comprehensive approach involving regular equipment monitoring and the use of monitoring and control systems is essential to minimize the risk of service interruptions. Power outages remain a persistent threat, often caused by equipment failures, poor management of the transition between different power sources, or adverse weather conditions. The causes of power outages can be numerous, and restoration can take a considerable amount of time. The main causes can include:

- Damaged inverter (DC/AC) or rectifier (AC/DC).
- Rapid battery discharge, insufficient runtime.
 - Surge protector (overvoltage)
 - Faulty room air conditioning, resulting in overheated IT equipment and an increase in temperature beyond the acceptable limit for equipment in the room.
 - Tripping of the circuit breaker due to a ground fault in the lighting.
 - Tripping of thermal-magnetic circuit breakers due to overload or short circuit.
 - Low fuel level in the generator tank.
 - Mechanical failure of the generator. It is not always possible to prevent these problems, but by being able to intervene quickly, it is possible to restore proper operation and eliminate or mitigate some of the causes. The data monitoring and control system solution for energy equipment continuously measures and records room temperature and humidity, as well as key electrical parameters (voltage, current), and the fuel level in the generator tank. This system also monitors battery status, charge level, and input and output voltages.

3. Methodology

The methodology consists of modeling the system, implementing a prototype, coding with the C++ language and testing the system in a production environment.

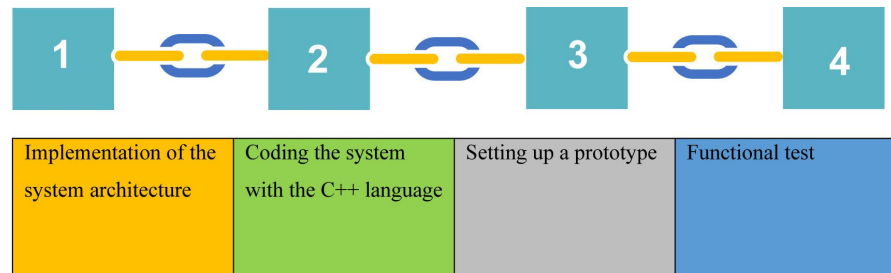


Figure 1. Methodology used.

Figure 1 shows the steps of our working methodology which respect the following points:

- Implementation of the system architecture.
- System coding with C++ language.
- Implementation of a prototype.
- Functional test.

4. Functional Analysis of the System

Functional analysis of the system allows us to understand and describe the essential functions that the system must perform to meet the monitoring, control, and management needs of equipment such as isolated GSM sites. To carry out this analysis, we used the octopus diagram, a graphical tool used in engineering to represent the main functions and constraints of a complex system, highlighting the interactions between the different functions and requirements. In the context of designing and implementing a SCADA system for isolated GSM sites, the octopus diagram helps clarify the system's main functions, as well as the associated constraints. Below is a description of the main functions and constraints for such a system. **Table 1** below show the main function of the system.

Table 1. Main functions and constraint functions of the system.

The Main Function 1	Measure real-time data from sensors (temperature, humidity, battery voltage, fuel level, etc.).
Main Function 2	Display the collected data on an interface to allow operators to monitor the status of the sites.
The Main Function 3	Detect anomalies, generate alerts, and notify operators if critical thresholds are exceeded.
The Main Function 4	Allow remote control of equipment (activation/deactivation, settings, etc.).
The constraint function 1	Interruptions in communication can affect real-time monitoring and remote control
The constraint function 2	Low energy consumption and an integrated battery to minimize the impact on local resources
The constraint function 3	Compatible with equipment for data collection (solar panels, batteries, inverters).

To define all the system's constructive functions, based on the service functions it must perform, we developed a FAST (Functional Analysis System Technique) diagram. Furthermore, this FAST diagram will clarify the technical choices for the energy equipment monitoring and control system by establishing the technical functions.

Table 2. Service functions (FS) and Technical functions (FT).

Function Service (FS)	Technical function (FT)	Associated Components
	FT1: Acquire the measurements	Sensors
FS1: Detect the anomaly	FT2: Process the measurements	Methods of treatment
	FT3: Centralize the data	local network
	FT4: Communicating information remotely	Communication protocols & the internet
FS2: Notify the maintenance manager	FT1: Report the anomaly	Alert and message
	FT2: Display related measures	UI
FS3: Acting on the system	FT31: Reconfigure operating parameters	Actionneurs (climatisation éclairage, source, etc.)
Archive data and events	FT32: Stop the production system	Commands
	FT41: Store the measurements	Servers and databases
	FT42: Recording Events	Servers and storage systems

5. Setting up the System Architecture

The system architecture consists of connecting the various sensors to the Arduino system. The one used is the Arduino Uno which is a microcontroller board featuring an ATmega2560 processor, 14 digital I/O pins (6 PWM), 6 inputs, a 16 MHz crystal, and USB/power connections. It supports UART, I2C, and SPI communication protocols. Key connections include a USB Type-B port for programming and communicating with sensors. These sensors will transmit the status of the lighting, the temperature functionality, and the fuel level used as an energy source on GSM sites. The signal will be retransmitted to a server which will be used by a computer or a smart phone connected to it as shown in the diagram below. Sensors are real mechanical devices that convert one kind of electricity into another. Sensors turn an actual thing into an electrical current to carry out the specified activity. In outdoor lighting, the system's sensors, for instance, can measure the brightness of the light by converting it into an electrical signal [1]. Sensor technology is arguably one of the most significant inventions in the past couple of decades. With the ever-evolving engineering behind the sensor, its role in the power system landscape is dynamic and vital in nature. In recent years, the transmission and distribution systems have relied heavily on sensor technology.

Knowledge about the fluctuating nature of the system parameters such as the range of currents, operating voltages, working temperatures, dynamic electric, and magnetic fields to even the moisture content, and the acoustics emanating from various transmission and distribution line equipment are essential for efficient monitoring and control purposes [2].

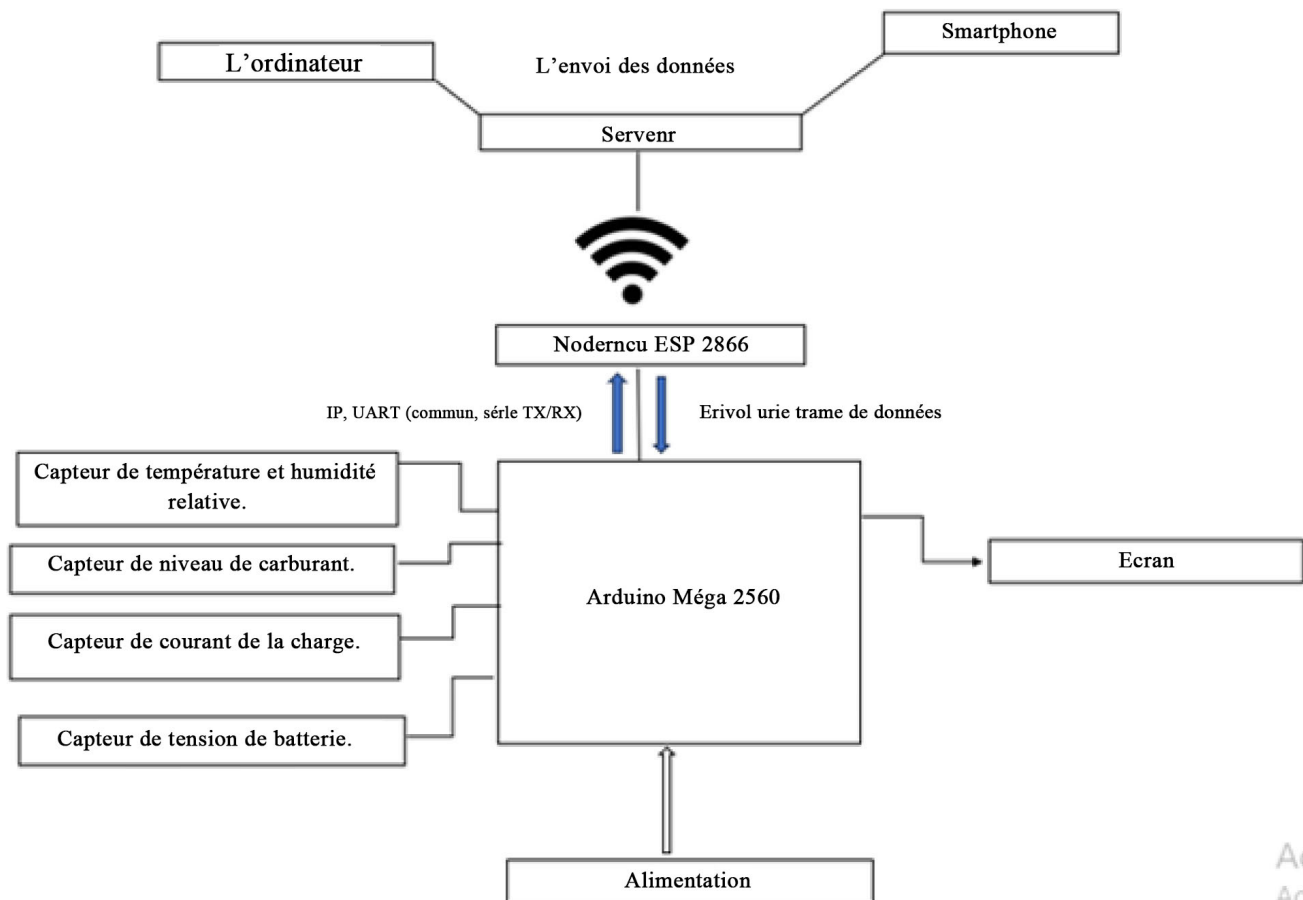


Figure 2. System architecture.

Figure 2 shows the system architecture. It explains the different level of security that will be check.

6. System Coding

The language used is C++, which is used on the Arduino IDE. C++ is one of the best language for programming on Arduino [3]. The code allows you to check a specific temperature and send alerts if the lighting is faulty or the air conditioning system does not meet the standards. The following code defines a threshold of 30.0°C and a brightness of 350°C. Depending on the temperature and lighting in the site, an alert is transmitted through FH. Arduino technology embraces hardware and software to support the designing and production of creative solutions [4].

```

// --- Broches actionneurs ---
#define BUZZER 8
#define LED_ROUGE 9
#define LED_JAUNE 10
#define LED_BLEU 11
#define BOUTON 3

// --- Seuils ---
#define HI_MAX 30.0 // seuil de température ressentie (°C)
#define LUM_SEUIL 350 // seuil luminosité

// --- Variables d'état ---
bool alerteHI = false;
bool alerteLum = false;
bool acquitte = false;
unsigned long acquitStart = 0; // moment de l'acquitement
const unsigned long delaiAcquit = 15000; // pause après acquitement

// --- Fonction calcul indice de chaleur ---
float computeHeatIndex(float T, float RH) {
  return T - 0.55 * (1 - RH/100.0) * (T - 14.5);
}

void setup() {
  Serial.begin(9600);
  dht.begin();

  pinMode(LED_ROUGE, OUTPUT);
  pinMode(LED_JAUNE, OUTPUT);
  pinMode(LED_BLEU, OUTPUT);
  pinMode(BOUTON, INPUT); // montage en pull-down

```

Figure 3. Code for temperature and fixed lighting at 30 and 350.

Figure 3 shows in detail the different configuration steps that are uploaded to Arduino while respecting certain operating rules.

```

// --- Broches actionneurs ---
#define BUZZER 8
#define LED_ROUGE 9
#define LED_JAUNE 10
#define LED_BLEU 11
#define BOUTON 3

// --- Seuils ---
#define HI_MAX 10.0 // seuil de température ressentie (°C)
#define LUM_SEUIL 350 // seuil luminosité

// --- Variables d'état ---
bool alerteHI = false;
bool alerteLum = false;
bool acquitte = false;
unsigned long acquitStart = 0; // moment de l'acquitement
const unsigned long delaiAcquit = 15000; // pause après acquitement

// --- Fonction calcul indice de chaleur ---
float computeHeatIndex(float T, float RH) {
  return T - 0.55 * (1 - RH/100.0) * (T - 14.5);
}

void setup() {
  Serial.begin(9600);
  dht.begin();

  pinMode(LED_ROUGE, OUTPUT);
  pinMode(LED_JAUNE, OUTPUT);
  pinMode(LED_BLEU, OUTPUT);
  pinMode(BOUTON, INPUT); // montage en pull-down

  digitalWrite(LED_ROUGE, LOW);

```

Figure 4. Fixed temperature and brightness code at 10 and 350.

Figure 4 code defines a temperature threshold of 10 and a brightness threshold of 350.

The hardware that Arduino uses is known as the Arduino development board while the software for the code development is the Arduino Integrated Development Environment (IDE). Built up with an 8-bit or 32-bit Atmel microcontroller, Arduino can be easily programmed by using variants of the C or C++ language in the Arduino IDE [5].

7. Prototype Setup

After the various phases of coding and uploading the code to the Arduino circuit, we obtain a prototype ready for testing. Prototypes are a fundamental part of any development. There are many ways to perform a test for a trial. Some trials have several tests in their procedures. Without Tests & Trails, a new idea does not advance in its technological maturity (TRL—Technology Readiness Level) and thus lengthens path of becoming a product [6]. The figure above shows the prototype of the system, which is now ready.

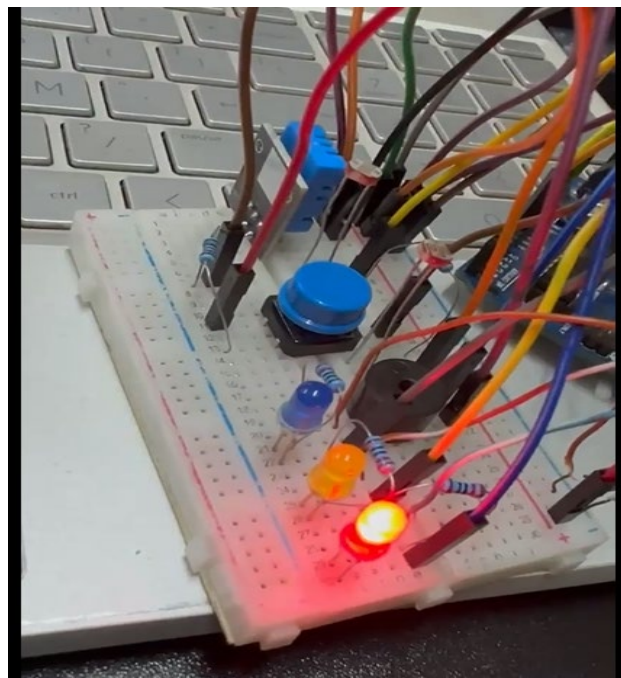


Figure 5. Prototype of the functional system.

Figure 5 shows the physical system as functional and ready for testing. It comprises all the components involved in the system's operation. Note that the LED is illuminated as specified in the code and that the device is functional. Battery Energy Storage Systems (BESS) combined with low-energy, renewable sources minimize local environmental impact by smoothing, storing, and discharging energy during peak demand. These integrated, systems reduce reliance on the grid, enhance energy resilience, and improve efficiency.

8. Results Obtained

After following all the steps to implement the system, the results based on the random values we used as conditions in our code accurately confirm the optimal functioning of the system based on the rules and formulas we considered.

System Functionality Test

The system is mainly tested on a university laboratory with a GSM infrastructure integrated. With a test temperature of 300 and a brightness of 350 as shown in **Figure 3**, the following results are obtained, with a clear timestamp from the system specifying the various alerts to the nearest second.

—Threshold
 Temperature: 30°C
 Brightness: 35°F
 —Expected Result
 Status: Brightness problem
 LED: Yellow
 Message: Brightness alert

```

COM4
16:57:11.877 -> 🚨 ALERTE ÉCLAIRAGE !
16:57:11.924 -> 📡 Luminosité mesurée : 321 (valeur analogique)
16:57:11.970 -> 📡 Seuil minimum requis : 350
16:57:12.017 -> 🚨 Vérifier le circuit d'éclairage (lampe défectueuse ou coupure) !
16:57:12.110 -> -----
16:57:14.085 -> 🚨 ALERTE ÉCLAIRAGE !
16:57:14.132 -> 📡 Luminosité mesurée : 247 (valeur analogique)
16:57:14.179 -> 📡 Seuil minimum requis : 350
16:57:14.225 -> 🚨 Vérifier le circuit d'éclairage (lampe défectueuse ou coupure) !
16:57:14.321 -> -----
16:57:16.295 -> 🚨 ALERTE ÉCLAIRAGE !
16:57:16.341 -> 📡 Luminosité mesurée : 255 (valeur analogique)
16:57:16.388 -> 📡 Seuil minimum requis : 350
16:57:16.434 -> 🚨 Vérifier le circuit d'éclairage (lampe défectueuse ou coupure) !
16:57:16.528 -> -----
16:57:18.538 -> 🚨 ALERTE ÉCLAIRAGE !
16:57:18.538 -> 📡 Luminosité mesurée : 247 (valeur analogique)
16:57:18.584 -> 📡 Seuil minimum requis : 350
16:57:18.631 -> 🚨 Vérifier le circuit d'éclairage (lampe défectueuse ou coupure) !
16:57:18.725 -> -----
16:57:20.735 -> 🚨 ALERTE ÉCLAIRAGE !
16:57:20.735 -> 📡 Luminosité mesurée : 236 (valeur analogique)
16:57:20.829 -> 📡 Seuil minimum requis : 350
16:57:20.829 -> 🚨 Vérifier le circuit d'éclairage (lampe défectueu
  
```

Figure 6. Lighting alerts.

Figure 6 shows that in our test we used a minimum required lighting threshold of 350. However, the sensors continuously measured 321, 247, 255, and 236, resulting in a series of alerts that will be relayed to administrators for decision-making.

For a test conforming to the characteristics of **Figure 4**, we obtain the following results, as shown in **Figure 7**.

—Threshold
 Temperature: 10°C

Brightness: 35°F
 —Expected Result
 Status: Brightness and temperature problem
 LED: Yellow and red
 Message: Brightness alert and temperature alert

```

16:58:49.613 -> ☺ Humidité relative : 63.50 %
16:58:49.613 -> ☺ Température ressentie (indice de chaleur) : 26.97 °C
16:58:49.695 -> ⚠ Vérifier immédiatement le système de ventilation !
16:58:49.742 -> -----
16:58:49.790 -> 🚨 ALERTE ÉCLAIRAGE !
16:58:49.836 -> ☺ Luminosité mesurée : 246 (valeur analogique)
16:58:49.885 -> ☺ Seuil minimum requis : 350
16:58:49.932 -> ⚠ Vérifier le circuit d'éclairage (lampe défectueuse ou coupure) !
16:58:49.977 -> -----
16:58:52.032 -> 🚨 ALERTE VENTILATION !
16:58:52.032 -> ☺ Température mesurée : 30.10 °C
16:58:52.079 -> ☺ Humidité relative : 63.50 %
16:58:52.125 -> ☺ Température ressentie (indice de chaleur) : 26.97 °C
16:58:52.172 -> ⚠ Vérifier immédiatement le système de ventilation !
16:58:52.267 -> -----
16:58:52.315 -> 🚨 ALERTE ÉCLAIRAGE !
16:58:52.315 -> ☺ Luminosité mesurée : 264 (valeur analogique)
16:58:52.361 -> ☺ Seuil minimum requis : 350
16:58:52.408 -> ⚠ Vérifier le circuit d'éclairage (lampe défectueuse ou coupure) !
16:58:52.503 -> -----
16:58:54.511 -> 🚨 ALERTE VENTILATION !
16:58:54.560 -> ☺ Température mesurée : 30.10 °C
16:58:54.560 -> ☺ Humidité relative : 63.50 %
16:58:54.608 -> ☺ Température ressentie (indice de chaleur) : 26.97 °C
16:58:54.700 -> ⚠ Vérifier immédiatement le système de ventilation !
16:58:54.746 -> -----
16:58:54.793 -> 🚨 ALERTE ÉCLAIRAGE !
16:58:54.841 -> ☺ Luminosité mesurée : 267 (valeur analogique)
16:58:54.888 -> ☺ Seuil minimum requis : 350
16:58:54.888 -> ⚠ Vérifier le circuit d'éclairage (lampe défectueuse ou coupure) !
16:58:54.982 -> -----
16:58:56.996 -> 🚨 ALERTE VENTILATION !
16:58:57.043 -> ☺ Température mesurée : 30.10 °C
16:58:57.089 -> ☺ Humidité relative : 63.50 %
16:58:57.089 -> ☺ Température ressentie (indice de chaleur) : 26.97 °C
16:58:57.182 -> ⚠ Vérifier immédiatement le système de ventilation !
16:58:57.229 -> -----
16:58:57.276 -> 🚨 ALERTE ÉCLAIRAGE !
16:58:57.322 -> ☺ Luminosité mesurée : 248 (valeur analogique)
16:58:57.368 -> ☺ Seuil minimum requis : 350
16:58:57.414 -> ⚠ Vérifier le circuit d'éclairage (lampe défectueuse ou coupure) !
16:58:57.460 -> -----

```

Figure 7. Brightness and temperature alert.

Figure 7 shows a lighting alert related to either a power outage or a faulty lamp, and also a ventilation alert.

The system for controlling temperature is achieved by using Arduino Uno-based microcontroller system. Arduino Uno due to its increased popularity is used on various system. Temperature sensor and Arduino Uno are the hardware used interfaced with computer, and the temperature is controlled in the room [7].

For a test with the following characteristics, we obtain these results:

—Threshold
 Temperature: 10°C
 Brightness: 150°F
 —Expected Result
 Status: Temperature problem
 LED: Red
 Message: Temperature alert

Figure 8 shows ventilation alerts related to a temperature problem. These alerts are important for enabling appropriate actions to be taken at GSM sites. This helps to better secure their system. An arduino system is defined as a collection of sen-

sors, actuators, communication devices, and computing devices that are connected to each other to provide homeowners with services and applications (e.g., safety and security, automation, entertainment, and energy management) with minimum or no intervention [8].

```

16:55:00.165 -> 🚨 ALERTE VENTILATION !
16:55:00.165 -> ☺ Température mesurée : 30.10 °C
16:55:00.212 -> ☺ Humidité relative : 64.40 %
16:55:00.240 -> ☺ Température ressentie (indice de chaleur) : 27.05 °C
16:55:00.306 -> 🚨 Vérifier immédiatement le système de ventilation !
16:55:00.353 -> -----
16:55:02.409 -> 🚨 ALERTE VENTILATION !
16:55:02.409 -> ☺ Température mesurée : 30.10 °C
16:55:02.455 -> ☺ Humidité relative : 64.40 %
16:55:02.501 -> ☺ Température ressentie (indice de chaleur) : 27.05 °C
16:55:02.548 -> 🚨 Vérifier immédiatement le système de ventilation !
16:55:02.595 -> -----
16:55:04.653 -> 🚨 ALERTE VENTILATION !
16:55:04.653 -> ☺ Température mesurée : 30.10 °C
16:55:04.699 -> ☺ Humidité relative : 64.40 %
16:55:04.754 -> ☺ Température ressentie (indice de chaleur) : 27.05 °C
16:55:04.800 -> 🚨 Vérifier immédiatement le système de ventilation !
16:55:04.846 -> -----
16:55:02.653 -> 🚨 ALERTE VENTILATION !
16:55:02.653 -> ☺ Température mesurée : 30.10 °C
16:55:04.699 -> ☺ Humidité relative : 64.30 %
16:55:04.756 -> ☺ Température ressentie (indice de chaleur) : 27.04 °C
16:55:04.857 -> 🚨 Vérifier immédiatement le système de ventilation !
16:55:04.903 -> -----
16:55:09.891 -> 🚨 ALERTE VENTILATION !
16:55:09.891 -> ☺ Température mesurée : 30.10 °C
16:55:09.938 -> ☺ Humidité relative : 64.30 %
16:55:09.984 -> ☺ Température ressentie (indice de chaleur) : 27.04 °C
16:55:03.850 -> 🚨 Vérifier immédiatement le système de ventilation !
16:55:07.323 -> -----
16:55:01.550 -> 🚨 ALERTE VENTILATION !
16:55:01.550 -> ☺ Température mesurée : 30.10 °C
16:55:11.463 -> ☺ Humidité relative : 64.30 %
16:55:13.693 -> ☺ Température ressentie (indice de chaleur) : 27.04 °C
16:55:13.290 -> 🚨 Vérifier immédiatement le système de ventilation !
16:55:13.838 -> -----
16:55:13.555 -> 🚨 ALERTE VENTILATION !
16:55:13.555 -> ☺ Température mesurée : 30.10 °C
16:55:11.440 -> ☺ Humidité relative : 64.30 %
16:55:11.447 -> ☺ Température ressentie (indice de chaleur) : 27.04 °C
16:55:11.585 -> 🚨 Vérifier immédiatement le système de ventilation !
16:55:11.631 -> -----
16:55:15.845 -> 🚨 ALERTE VENTILATION !
16:55:15.845 -> ☺ Température mesurée : 30.10 °C
16:55:13.950 -> ☺ Humidité relative : 64.30 %
16:55:13.938 -> ☺ Température ressentie (indice de chaleur) : 27.04 °C
16:55:16.050 -> 🚨 Vérifier immédiatement le système de ventilation !
16:55:16.074 -> -----
16:55:18.079 -> 🚨 ALERTE VENTILATION !
16:55:18.079 -> ☺ Température mesurée : 30.10 °C
16:55:18.113 -> ☺ Humidité relative : 64.30 %
16:55:18.212 -> ☺ Température ressentie (indice de chaleur) : 27.04 °C
16:55:18.258 -> 🚨 Vérifier immédiatement le système de ventilation !
16:55:18.307 -> -----
16:55:20.574 -> 🚨 ALERTE VENTILATION !
16:55:20.574 -> ☺ Température mesurée : 30.10 °C
16:55:20.409 -> ☺ Humidité relative : 64.30 %
16:55:20.456 -> ☺ Température ressentie (indice de chaleur) : 27.04 °C
16:55:20.503 -> 🚨 Vérifier immédiatement le système de ventilation !
16:55:20.550 -> -----

```

Figure 8. Ventilation alerts.

In our research, functional analysis is important as **Table 2** shows. It's a method that integrates the need that will mark the product characteristics, technical solutions to satisfy its requirements, as well as the topological analysis that lays the groundwork for the future product development. It enables to characterize the functions offered by a product in order to satisfy the user's needs [9]. We also used FAST in our research because Construction projects require costs as a very important element in their implementation. There are several alternative methods that can be used as a rationale for conducting studies on cost savings. One of the alternatives that can be used for savings by eliminating unnecessary costs so that the value of the project can be reduced is Value Engineering (value engineering). One of the important stages in VE is the analysis of functions whose purpose is to

identify the functions that are most beneficial for the study of VE. The method used in the analysis of this function is the Function Analysis System Technique (FAST) Diagram with the aim of defining the function of each work item analyzed and can review the basic functions that are used as guidelines in the selection of alternative designs for cost savings [10].

The results are obtained from tests performed on real GSM equipment, the outputs of which are displayed on our screens in our university laboratory.

9. Conclusion

The development of technology has profoundly transformed our way of life, opening up new possibilities in various fields, including surveillance and control [11]. In this research, we established the importance of managing and securing remote sites, which are crucial for telecommunications companies in Burkina Faso. We designed the system, then implemented a prototype, using an object-oriented programming language. The results are convincing and can be used in a production system to secure GSM sites, providing adequate security control and supervision of critical aspects essential to their proper operation. The results demonstrate that alerts are sent to enable administrators to make appropriate decisions. These alerts can be sent to the central system, mobile phones, or servers through FH. With arduino you can build real, working projects like LED blinkers, motion detectors, temperature sensors, and even interactive devices with displays and buzzers—all while mastering the core programming concepts that power them.

Conflicts of Interest

The authors declare no conflicts of interest regarding the publication of this paper.

References

- [1] Arshi, O. and Mondal, S. (2023) Advancements in Sensors and Actuators Technologies for Smart Cities: A Comprehensive Review. *Smart Construction and Sustainable Cities*, **1**, Article No. 18. <https://doi.org/10.1007/s44268-023-00022-2>
- [2] Swain, A., Abdellatif, E., Mousa, A. and Pong, P.W.T. (2022) Sensor Technologies for Transmission and Distribution Systems: A Review of the Latest Developments. *Energies*, **15**, Article 7339. <https://doi.org/10.3390/en15197339>
- [3] Saini, S. and Kaur, M. (2022) *Arduino Solutions Handbook: Design Interesting DIY Projects Using Arduino Uno, C and C++* (English Edition). BPB Publications.
- [4] Tselegkaridis, S., Sapounidis, T., Tokatlidis, C., Rapti, S. and Papakostas, D. (2025) AABC: A Tool for Assessing Arduino Basic Coding Skills. *Computer Applications in Engineering Education*, **33**, e70029. <https://doi.org/10.1002/cae.70029>
- [5] Michailidis, I., Mountzouris, P., Triantis, P., Pagiatakis, G., Papadakis, A. and Dritsas, L. (2025) An Arduino-Based, Portable Weather Monitoring System, Remotely Usable through the Mobile Telephony Network. *Electronics*, **14**, Article 2330. <https://doi.org/10.3390/electronics14122330>
- [6] Wiltgen, F. (2022) Prototypes Are Important. *Transformations/ Transformacje*, **4**, 109-125.
- [7] Khaing, K.K., Srujan Raju, K., Sinha, G.R. and Swe, W.Y. (2020) Automatic Temper-

- ature Control System Using Arduino. In: Raju, K., Govardhan, A., Rani, B., Sridevi, R. and Murty, M., Eds., *Proceedings of the Third International Conference on Computational Intelligence and Informatics*, Springer, 219-226.
https://doi.org/10.1007/978-981-15-1480-7_18
- [8] Sarhan, Q.I. (2020) Systematic Survey on Smart Home Safety and Security Systems Using the Arduino Platform. *IEEE Access*, **8**, 128362-128384.
<https://doi.org/10.1109/access.2020.3008610>
- [9] Bououchma, M. and Herrou, B. (2023) Quality Management, Tools, and Interactions. In: Chana, A.B. and Benazza, H., Eds., *Enhancing Performance, Efficiency, and Security through Complex Systems Control*, IGI Global, 78-91.
<https://doi.org/10.4018/979-8-3693-0497-6.ch005>
- [10] Gede Angga Diputera, I. (2022) Application of Function Analysis Using Function Analysis System Technique (FAST) Diagram on Taman Sari Apartment Construction Project. *Journal of Applied Science, Engineering, Technology, and Education*, **4**, 39-44. <https://doi.org/10.35877/454ri.asci847>
- [11] Lylia, M. (2023) Design and Implementation of a Forest Fire Detection and Alert System Based on an Arduino Uno Board. Ph.D. Thesis, Faculté du Génie Electrique et d'Informatique UMMTO.