



INTERNATIONAL JOURNAL OF SCIENTIFIC RESEARCH AND INNOVATIVE STUDIES

ISSN: 2820-7157

Advancing Knowledge, Inspiring Innovation

www.ijsrisjournal.com



PEER REVIEWED
Rigorous & Transparent

DOI: CROSSREF
Member

OPEN ACCESS
For Global Visibility

BI-MONTHLY
Timely Publication

WIDE READERSHIP
High Impact & Visibility

AUTHOR FRIENDLY
Fast | Supportive | Reliable

RECEIVED
17 August 2026
 ACCEPTED
09 September 2026
 PUBLISHED
1 October 2026

Pages from:

October 2026

Volume 5

Number 5

228 to 242

Design of an embedded temperature control system for a portable medicine storage kit

KAMBALE VAYIRWE Jeannot¹, MUMBERE MATHE Papy¹, TASIVIWE POSHO Anne-Marie¹, KASOKI SAGHASA Premice¹

¹Lecturer and Researcher, Higher Institute of Commerce in Beni, Beni, Democratic Republic of the Congo (DRC)

LICENSE



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

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



PREFIX :

10.63883

IJSRIS JOURNAL

OPEN ACCESS



This is an open access article
distributed under the terms of
open access.

ABSTRACT

Our research focuses on **the design of an embedded temperature control system for a portable medicine storage kit**. The aim of this research is to develop an embedded computerised refrigeration and temperature control system capable of storing medicines requiring a specific ambient temperature in environments where a reliable

electricity supply is not available, to create an embedded computer system capable of automatically regulating the temperature within a medicine storage kit for items such as vaccines, insulin and other medical products that must be stored at ambient temperature to prevent deterioration.

Temperature sensors, the cooling unit and the temperature controller will greatly assist in the realisation of this research, thanks to instant monitoring via the LCD screen on the kit,

ensuring that the normal storage conditions for the product inside the kit are not exceeded. The advantage of this system is that it prevents shortages of certain medical products in areas without full electricity supply. Thanks to the Peltier module and the temperature sensor connected to a microcontroller, this system will enable automatic cooling and instantaneous control. The integration of C programming within the Arduino development environment, in conjunction with C#, will enable the microcontroller to be developed (coded).

Keywords: Embedded system, temperature control, portable kit, medicines.

1. Introduction

Computer science is recognised as a standard science, a science of processing, particularly through automated information processing systems in scientific, technical, industrial, economic and social fields, enabling and facilitating automation in various sectors, including the healthcare sector, which also benefits from it (Michel Robert Discal, 2004, p. 14) . It continues to stand above other sciences by playing a role in most vital areas, owing to competitiveness across all sectors and even the evolution of society, through its impact in fields such as healthcare, the economy, accountancy, education, leisure, the environment... and no one can claim to be exempt from the use of this technology, namely information technology. Its role is of paramount importance as it helps numerous businesses, institutions, organisations and hospitals to ensure the control of information flow. Nowadays, the trend is towards adopting increasingly intelligent systems with the aim of maximising information processing speed, facilitating the exchange of information and reducing costs – not only in terms of labour but also of equipment – for the benefit of corporate profit and profitability.

As the world moves towards the development of expert and intelligent systems capable of carrying out tasks with minimal human effort, it is essential to recognise that many sectors, such as healthcare, certainly require autonomous systems capable of storing medical products – including pharmaceuticals – either in a cold store or at room temperature for preservation (Didikey, 2021) .

Similarly, in Africa, certain African countries – such as those in the Sahel – and the inhabitants of the Sahelian regions in particular are subject to enormous problems of discomfort, generally during hot periods. Indeed, most Sahelian countries are situated in the solar belt, where normal direct solar radiation exceeds 2,000 kWh/m²/year. Furthermore, average

daily temperatures in these regions hover around 35°C, with maximum temperatures reaching around 45°C. The sun shines for an average of 8 hours a day and the average relative humidity is approximately 48 per cent. Consequently, the storage of medical and pharmaceutical products poses a problem, given that the majority of the population in this region is poor. Due to these high temperatures and intense sunshine, some of these countries use Peltier technology for their ceiling lights and iceboxes (NOUADJE, 2011) .

Thus, our country, the Democratic Republic of the Congo, being a developing nation and equally affected by the reality of new technology that is transforming all businesses, nevertheless requires these technological advancements to increase productivity within businesses, particularly in the medical sector – a key sector requiring greater automation in several departments, not least the pharmaceutical department. The impact of automation on the medical and pharmaceutical sectors and their management is no longer in doubt, as in these sectors certain services – such as the pharmaceutical departments of many healthcare facilities – still use archaic methods to store certain products, such as insulin and vaccines. Indeed, automation plays a role in many areas of our daily lives; as it develops, it provides increasing support to the operations of numerous businesses and sectors. This means that the march towards technological development in the pharmaceutical and medical sectors is a matter of daily and tactical automation (Jean-Marie Domini & Lucien, 2013, p. 179) , hence the need for more firmly established technological progress; its development must, first and foremost, be more thoroughly applied in both sectors, where machines and tools must be implemented to support people's daily activities, with the aim of creating expert systems that will enable them to solve problems related to the storage of various products, in order to ensure better protection of certain medical products, blood banks, and other products requiring refrigeration systems, which unfortunately result in additional expenditure on electricity and refrigeration equipment.

At the regional level, such as in the outskirts of major urban areas, the preservation and protection of certain pharmaceutical products still pose problems due to the high cost of purchasing equipment and storage facilities that maintain the correct ambient conditions, the lack of a constant electricity supply to operate the equipment or refrigeration units, or the inability to procure them. Consequently, many facilities in rural areas face problems in storing certain products such as vaccines and certain antibiotics, which are often stored in bottles of water when they actually require a controlled storage environment.

Given the boundless innovations in digital information and communication technology, we have observed that most hospitals in the country's outlying areas are not reaping the benefits offered by this new technology. The pharmaceutical and medical sector within rural healthcare facilities requires the implementation of a rational yet rapid-response storage kit utilising programmable electronic components with low power consumption; we refer here to direct current (DC) solar power, which can operate autonomously thanks to an intelligence system embedded in a microcontroller, tailored to the specific product in question.

In light of the above, the following question merits our attention:

What cost-effective strategy or mechanism can be applied to address the problem of storing medical and pharmaceutical products that require a controlled ambient environment?

The design of an embedded temperature-control system within a low-power-consumption medicine storage kit will address the challenges of storing medical products in rural areas. In practical terms, this embedded computing system can automatically regulate the temperature within a medicine storage kit — specifically, a kit containing programmable electronic components, with low power consumption, capable of accommodating several types of medicines according to their required storage conditions, utilising a Peltier module for cooling combined with a fan and heat sink to regulate the temperature via a programme integrated into a microcontroller.

2. Literature Review

In this section, we present the theories relevant to our research topic with the aim of contextualising our research in order to establish its unique contribution. To justify the originality of our research, we have analysed certain studies undertaken by our predecessors in our field of research. During this analysis, we have focused on the limitations of these studies.

Astrid Brigitte MEDJO (2011), in her research on combining solar energy with Peltier modules for cooling in the Sahel, focused on the production of cooling capable of providing a comfortable climate in certain parts of the Sahel region by using solar energy and Peltier modules to generate cooling for comfort. Consequently, the question of how to produce a low-cost air-conditioning system in the Sahel region by utilising the region's natural climate was addressed. The researcher took a keen interest in the methodology of cooling via mechanical and electronic compression using the Peltier module; thanks to this, he achieved a significant result

whereby the direct current produced by photovoltaic panels is utilised by Peltier modules to generate cooling (NOUADJE, 2011). Lucien Quaranta (1997), in his embedded system for heat generation, temperature control and environmental cooling based on liquid nitrogen, demonstrated certain methods of temperature control using a laser thermostat (Lucien Quaranta, 1997).

Traore David (2014), in his embedded system for phosphorus-based heat generation and the maintenance of a constant temperature using a sea breeze sensor in a server room, demonstrated certain methods of temperature monitoring using an MVD display. (TraoreDavid, 2014, p. 29)

Marie François (1824), in her research on temperature verification in an attic undergoing adiabatic transformation, sought to examine the isothermal and thermodynamic processes — or the physical transformation of the system — in order to determine whether the system's temperature remains constant and uniform in the attic undergoing adiabatic transformation, using pressure and temperature sensors.

Nevertheless, we find these systems insufficient to resolve the problems faced by rural communities — particularly those without electricity, as in the case of our research — given the findings of individual researchers who have focused solely on issues relating to the monitoring, monitoring and temperature control in their own ways and within their respective research, using embedded automated systems equipped with intelligence to detect abnormal movements and provide reliable results for their fields of investigation. In our research, we will develop an embedded temperature control system within a medicine storage kit, capable of displaying the results on an LCD screen.

Using the SysML modelling language, we will analyse and model our system in order to represent its operation in a simplified and structured manner, either before or during its design (Pierre Alain Muller and Nathalie, 2003). In our case, this involves an embedded temperature control system within a medicine storage kit, which will assist various healthcare organisations and pharmacies in correctly storing pharmaceutical and medical products requiring a specific ambient temperature, depending on the product's storage method, and displaying the temperature regularly on a screen. We will adopt a similar approach to previous work whilst incorporating real-time temperature monitoring on the LCD screen; using the DHT22 temperature sensor, we will take regular temperature readings, which will be regulated by the Peltier effect sensor and an energy-efficient heat sink, and will regularly display temperature fluctuations on a screen, generating a log that shows temperature trends over the

storage period. For healthcare professionals, patients and communities in remote areas, the main objective is to ensure the availability of safe, undamaged and effective medicines and vaccines, even in regions without continuous access to electricity.

For the beneficiary, this translates into:

- **Health safety:** The assurance that temperature-sensitive products (insulin, vaccines) retain all their therapeutic properties without any risk of deterioration.
- **Continuity of care:** Local access to essential treatments without the risk of stock shortages due to power cuts or cold chain issues.
- **Peace of mind:** Direct visual monitoring of the temperature via the LCD screen, allowing instant verification that storage conditions are being maintained.

3. Methods

In this section, we will discuss the methodology and present the tools we have used. In presenting the methodology, we will highlight the Unified Process (UP) and the general principles of UML, in particular the language derived from it known as SysML – an object-modelling language that will enable us to demonstrate how we went about formalising the system.

A method is the set of reasoned steps (or procedures) followed in a given activity (or adopted in a given field) (LeRobert, 2005).

The Unified Process (UP) is a generic software development method; ‘generic’ means that UP must be adapted to the context of the project, the team, the domain and/or the organisation. (Jean-Louis Bénard, 2002). The Unified Process is a software development process: it brings together the activities required to transform a user’s requirements into a software system.

The following key characteristics and reasons lead us to opt for this method:

- It is risk-driven. The major causes of software project failure must be addressed as a priority. The primary cause may stem from the technical architecture’s inability to meet operational constraints, whilst a secondary cause relates to the development failing to

meet user requirements. Consequently, we first identify any risks that could prevent us from achieving our research objectives and take measures to avoid them (Roques P & Vallée F, 2007).

- It is iterative and incremental. Defining development iterations is, in fact, best practice for managing both technical and functional risks. This characteristic helps us avoid falling into the trap of believing we have covered everything. We therefore constantly bear in mind that requirements may change and that we should be prepared to adapt.
- It is user-centred, as the specification and design are based on the expected usage patterns of the system’s stakeholders. The primary objective of a system is to serve its users; it is therefore essential to fully understand users’ desires and needs.

The methodology we will use in this research (UP) will consist of employing models that provide a simplified representation of reality and an abstract representation of the data within an information system. This representation will generally be expressed using a graphical language known as ‘formalism’ (*data model*) (GILLES ROY, 2009, p. 30). This graphical language provides an abstract and simplified representation (*i.e.* one that omits certain details) of a real-world entity (phenomenon, process, system, etc.) with a view to describing, explaining or predicting it. It is a common, precise language known to all team members and is therefore, as such, a preferred means of communicating theories, but with a practical connotation of an action-oriented theory that it is intended to serve (Laurent AUDIBERT, 2009, p. 14). In our field of software engineering, the use of this model will enable us to distribute tasks more effectively and to automate some of them. It also helps to reduce costs and lead times (LAURENT, 2009, p. 15).

4. System modelling procedure using SysML

The diagrams will be drawn up using the SysML ‘System Modelling Language’, which is a graphical modelling language derived from UML. We will use SysML for the following reasons:

SysML is a graphical modelling language derived from UML; it is used to model complex systems, particularly hardware. As has been rightly pointed out, SysML is a graphical

modelling language, which means that every element of the system's description must be represented using a diagram (eduscol, 2026) . We might therefore say that SysML is a language, that is to say, a set of model elements, their meanings and the way in which they are represented. This language is essentially graphical: the elements are boxes connected by lines, which may or may not be directed, depending on the context. The boxes and lines have types, indicated by graphical conventions or keywords; the type of a box indicates whether it represents a component, a behaviour, etc., whilst the type of a line specifies the nature of the relationship between the boxes. Furthermore, the boxes bear names used to identify the elements of the model they represent: two boxes bearing the same name are two (in principle different) representations of the same element, and these representations cannot contradict one another (though they may complement one another). SysML has a hierarchical structure, consistent with systems theory; many boxes may contain other boxes linked by lines, and the entire model is itself a box. The representation of a box's contents is called a SysML diagram. Diagrams, just like the boxes they represent, have types that determine the nature of the information they contain (structural, behavioural, etc.). Each diagram must be drawn within a frame featuring a header that specifies its type, followed by the type and name of the box represented, and finally a free-form description. Thus, a SysML diagram is, by definition, a view of the model, that is to say, a graphical representation of a part of the model (in this case, a box) from a particular perspective (determined by its type); the header enables this view to be identified precisely. Conversely, a SysML model is represented by a collection of interlinked diagram (Gendre L, 2013) . Finally, SysML does not require exhaustiveness: a diagram represents only those elements necessary to achieve the objective (whether that be to communicate or to simulate), and the absence of an element from a given diagram in no way implies that that element does not exist (it may well appear in other diagrams). Two diagrams representing the same element in different ways should therefore be regarded as complementary; however, if they prove to be contradictory, then this constitutes an error, as explained above. In our research, we will present the main SysML diagrams, which are: the block definition diagram, the internal block diagram and the requirements diagram, as modelling a system prior to its implementation enables a better understanding of how the system works. It is also a good way of managing its complexity and ensuring its consistency. (Laurent AUDIBERT, 2009, p. 13) It should be noted that modelling using the UML language has enabled us, from a:

Functional perspective, to illustrate the use cases and the actors involved in them. In this instance, this involved constructing the use case diagram – the first high-level UML

model linking actors and use cases – and the sequence diagram or activity diagram, which clarifies the functional perspective by detailing the various ways in which actors can use the system. (<https://slideplayer.fr/slide/1313429/>, 2019)

4.1. Functional modelling

4.1.1. Actors

For our system, we have identified the following actors:

Main

- Curator;

Secondary

- Display (LCD);
- LM35
- DHT11
- Peltier Cooler.

4.1.2. Use Cases

For the purposes of our study, the use cases below will enable us to model the system:

- Connecting the sensors;
- Start the system;
- Display the temperature reading;
- Measure the humidity level of the kit;
- Generate cooling;
- Measure the temperature;
- Detect temperature;
- Detect current.

4.1.3. Use case diagram

Use case diagrams are UML diagrams used to provide an overview of the functional behaviour of a software system. They are useful for presentations to management or project stakeholders, but for development purposes, use cases are more appropriate. A use case represents a discrete unit of interaction between a user (human or machine) and a system.

It is a meaningful unit of work. In a use case diagram, users are referred to as actors; they interact with the use cases. (https://www.ibm.com/, 2019) . The use case diagram enables requirements to be gathered, analysed and organised, and the system’s functionalities to be identified. This diagram consists of actors, use cases and the associations or relationships between them.

Here is our use case diagram:

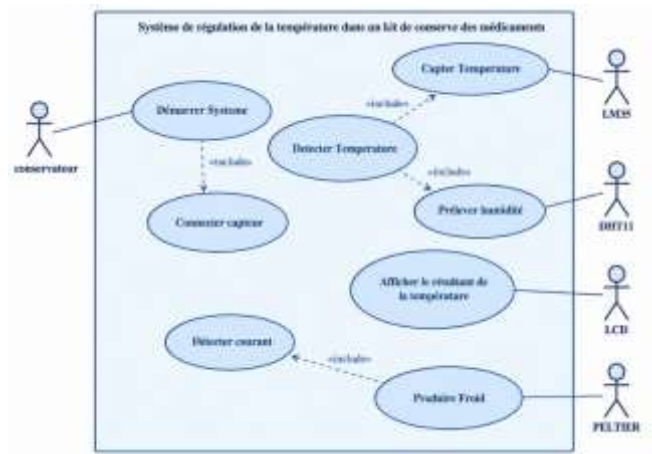


Figure1: Use-case diagram.

4.1.4. Textual description of use cases

Each use case must have a clearly defined lifespan (a start and an end). The description of the use cases details the sequences of actions (sequences) that are executed from start to finish. These sequences of steps, which describe an interaction between the user and the system enabling the user to achieve an objective, will be the focus of this section. (For each use case, we will present a fact sheet detailing the sequences of actions that take place. Let us begin with the use case fact sheet:

Table 1: Case description – connecting the sensors.

DESCRIPTION OF THE ‘CONNECT SENSORS’ USE CASE
Identification <i>Title:</i> Connecting the sensors <i>Purpose:</i> the aim is to connect the sensors to ensure the temperature control system operates correctly. <i>Actor:</i> Caretaker Sequence

❖ Precondition(s) - Be the curator; - Have knowledge of preservation; - Be familiar with the various sensors in the kit. ❖ Scenario or sequence • Standard sequence 1. The Conservation Officer tests the numerical reading of each sensor and checks the position of each sensor on the system; 2. The Conservator secures the sensors in their respective positions on the kit; 3. The system checks and configures the sensors; 4. The maintenance technician connects the system to the mains power supply. • Alternative sequence 1. Inappropriate current. • Error 1. No cooling output. 2. Failure of the sensors to operate.
Identification <i>Title:</i> Starting the application <i>Objective:</i> to start the kit <i>Person responsible:</i> Conservation Officer Sequence ❖ Precondition(s) - Be the conservation officer; - Have knowledge of conservation; - Understanding how the kit works. ❖ Scenario or sequence • Standard sequence 1. The Conservation Officer checks the sensor’s numerical reading;

2.	The Conservation Officer records the value returned by the verification test;
3.	The Maintenance Officer connects the system to the mains supply;
4.	The system sends a signal confirming connection and power supply;
5.	The Maintenance Officer activates the system.
Alternative sequence	
1.	The sampler checks the sensor's numerical reading
Error	
1.	Sensor not responding;
2.	Connection error.

Table 2: description of the case for measuring temperature and humidity.

CASE DESCRIPTION: MEASURING TEMPERATURE AND HUMIDITY	
<u>Identification</u> <i>Title: Measuring temperature and humidity.</i> <i>Aim:</i> to enable the automatic measurement of temperature and humidity within the kit. <i>Components:</i> LM35 and DHT11 <u>Sequence</u> ❖ Prerequisites - The kit must be powered by a heat source; - The sensing conditions must be strictly adhered to; ❖ Scenario or sequence A. Nominal sequence. - The LM35 and DHT11 sensors detect the temperature from the Peltier module and the humidity of the kit in the form of electrical signals. - Both sensors send these signals to the MICROCONTROLLER. - The microcontroller converts these signal variations into temperature variations. B. Alternative sequence	

1.	Reset the sensors if no temperature is detected.
2.	Restart the system if no signals have been sent to the microcontroller.

Table 3: Case description – displaying the temperature reading.

CASE DESCRIPTION: DISPLAYING THE TEMPERATURE READING	
<u>Identification</u> <i>Title: Display of temperature variation</i> <i>Purpose:</i> to display temperature variation to enable the curator to monitor the condition of the kit whilst it is in storage <i>Component:</i> LCD <u>Sequence</u> ❖ Precondition(s) - Receive the signal from the microcontroller; - Display conditions must be met. ❖ Scenario or sequence • Nominal sequence - The LCD requests temperature and humidity data from the system's microcontroller. - The system synchronises the data. - The system sends the synchronised data back to the LCD. - The LCD displays the changes in temperature and humidity. • Alternative sequence - If temperature and humidity data are missing, the system restarts. - The LCD screen displays nothing; check the display cables. • Error Blue screen error	

Table 4: Description of the case: generating cold.

CASE DESCRIPTION: COOLING	
<u>Identification</u>	
<i>Title: Producing cold</i>	
<i>Aim: to generate cooling to ensure the proper preservation of the product</i>	
<i>Component: Peltier module</i>	
<u>Sequence</u>	
❖ Prerequisite(s)	- Receive the signal from the microcontroller; - Production conditions must be met.
❖ Scenario or sequence	• Nominal sequence 1. The Peltier module requests energy information from the system's microcontroller. 2. The system synchronises the information. 3. The system returns the energy information requested by the Peltier. 4. The Peltier module provides cooling in accordance with the energy request. • Alternative sequence 1. No energy data. 2. The Peltier produces nothing; check the display cables. • Error - Overheating error

4.2. Modelling using the SySML-derived

As we are discussing an embedded system, in this paper we will present some key diagrams using the SySML-derived, which is based on the UML language.

4.2.1. Internal Block Diagram

The internal block diagram illustrates how blocks interact through their connections and exchange flows: **matter**, **energy** and **information**. The internal block diagram (ibd: 'internal block diagram') describes the internal view of a block and is based on the BDD (Block Definition Diagram) to

represent the assembly of blocks. (GENDRE, 2013, p. 11) It introduces the fundamental concept of a **port**, which corresponds to a point of interaction with the outside of the block.

A block may have several ports, each specifying a different point of interaction. These ports can be of two types:

- A **standard port (simple square)** corresponds to an input or output interface for command, control, adjustment, etc.; this type of port enables the description of logical services between blocks, by means of interfaces that group operations together. Example: hmi (human-machine interface)
- A **flow port (square with an arrow)** corresponds to an element through which matter, energy or information passes. The **connectors** (lines) between the ports indicate either associations or flows of matter, energy and information between the different blocks. This type of port allows the circulation of physical flows between the blocks. The nature of what can flow ranges from fluids to data, as well as energy.

The image below shows the internal block diagram:

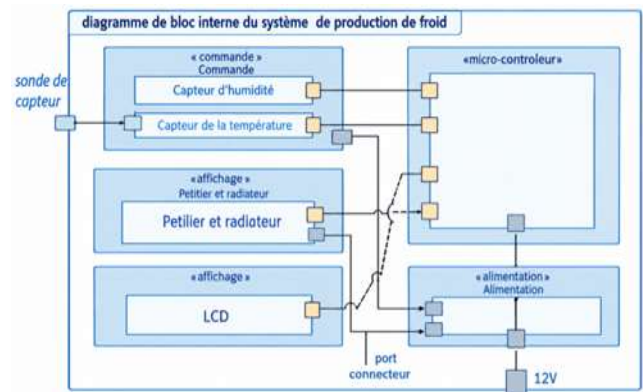


Figure 2: Diagram of the internal blocks.

4.2.2. Block definition diagram

The block definition diagram models the structural architecture – that is, the system's composition and components – by answering the question 'what contains what?'. This diagram allows the system's structure to be visualised at a glance by representing links between blocks at the same level as an association, and between blocks at different levels as a composition. It can also show the main characteristics of each block by displaying its operations and

properties. The image below shows the block definition diagram:

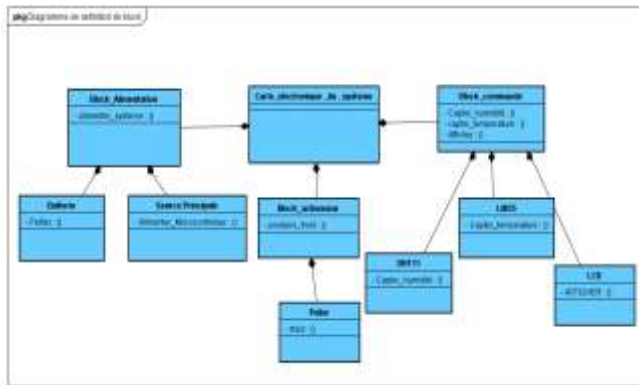


Figure 3: Block definition diagram.

4.2.3. Requirements diagrams

Whether for systems engineering or software development, requirements are commonly used to formalise the system's prerequisites, which translate into functionalities or conditions that must or should be met by the system. The requirements diagram therefore serves to represent all of these.

Purpose: It describes the requirements set out in the functional specification. A requirement expresses a capability or constraint that a system must satisfy. Requirements serve to establish a contract between the client and the developers of the future system.

Below is an image of the requirements diagram for our system:

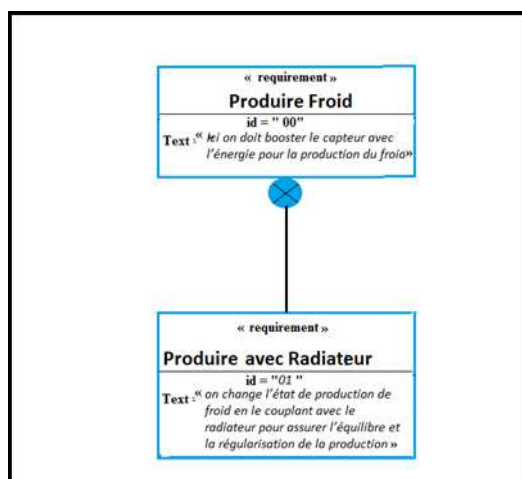


Figure 4: Requirements diagram.

5. Embedded System Hardware Configuration and Prototype Overview

This section aims to present the prototype that we are going to implement. We will begin by presenting the model (the schematic structure of our prototype) and the programming language used; finally, we will present some algorithms for certain methods within the system.

To make the best use of an embedded system model, it is necessary to know the location of the hardware on the microcontroller, the hardware configuration and the software configuration required for it to function correctly. (Simon Landrault, Eskimon, Hippolyte Weisslinger, 2014) That is why, in this section, we outline the minimum hardware and software requirements, as well as the location of each sensor on the Arduino board equipped with a microcontroller, to ensure that the new generator works perfectly. Our microcontroller, which will help us manage our sensors, is connected directly to the Arduino UNO board. Let us show that the basic principle of microcontrollers is based on the integration, within the same package, of the microprocessor and various peripherals, thereby creating a self-contained component. The data and address buses are internal to the component, as are the address decoders for each 'internal peripheral'. The peripheral control registers are grouped within this area of the microcontroller's various configuration registers. The component's pins are therefore essentially reduced to input-output lines, both digital and analogue. (Perrin JP, 2009–2010)

Schematic configuration of our system's hardware on the Arduino UNO board/electronic diagram:

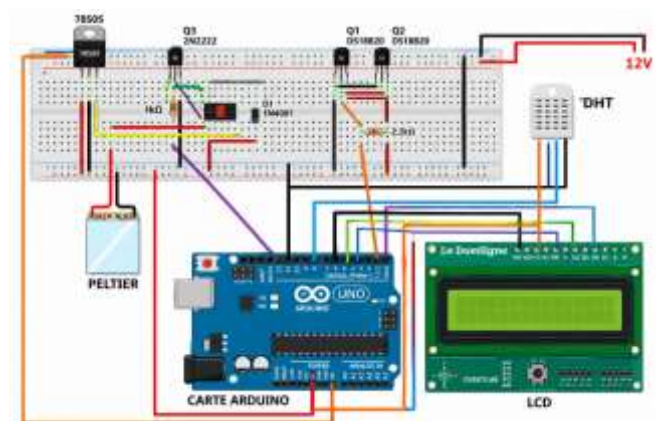


Figure 5: Schematic configuration of the hardware.

5.1. Overview of the prototype's sensors

5.1.1. The LM35 temperature sensor

To measure the temperature provided by the Peltier module, we will use the LM35 temperature sensor. This sensor has three terminals, including:

1. The sensor power supply (4–20 V)
2. The output voltage (OUT)
3. Ground (GND)



Figure 6: LM35 temperature sensor.

The LM35 temperature sensor is an analogue temperature sensor manufactured by Texas Instruments. It is extremely popular in electronics because it is accurate, inexpensive, very easy to use and exceptionally reliable. The LM35 temperature sensor is capable of measuring temperatures ranging from -55°C to $+150^{\circ}\text{C}$ in its most accurate version. The general-purpose versions, which are the ones we are using here, are capable of measuring temperatures between -40°C and $+140^{\circ}\text{C}$.

The sensor's analogue output is proportional to the temperature. You simply need to measure the sensor's output voltage to determine the temperature. Each degree Celsius corresponds to a voltage of $+10\text{ mV}$. For the LM35 sensor, there are two possible circuit configurations: a very simple one for measuring positive temperatures only (2°C to 140°C) and a second, more complex one for measuring temperatures from -40°C to 150°C . In this case, we will use the first configuration, which measures positive temperatures only (2°C to 140°C).

The LM35 sensor can withstand quite extreme temperatures (down to -55°C and up to $+150^{\circ}\text{C}$).

5.1.2. The LCD display

LCD screens have been in existence since 1971. They have continued to evolve ever since and are now found in many devices with on-board displays (cameras, watches, voice recorders, digital locks, telephones, etc.).

LCD is the abbreviation for 'liquid crystal display'. These displays are used in a great deal of embedded equipment that requires information to be displayed. This technology enables the creation of flat screens that consume less energy. It is worth noting that there are many types of LCD screens available for our project; the LCD screen we will be using in our setup is a 16×2 character display, as shown in the image below.



Figure 7: The LCD display.

5.1.3. The DHT11 temperature and humidity sensor

The DHT11 temperature and humidity sensor communicates with a microcontroller via a serial port. It comes with a pull-up resistor for the data output. (<https://github.com/adafruit/DHT-sensor-library>)

Specifications:

- Power supply: 3 to 5 Vdc
- Maximum current consumption: 2.5 mA

Measurement range:

- Temperature: 0 to $+140^{\circ}\text{C}$
- Humidity: 20 to 100 per cent RH

- Dimensions: 16 x 12 x 7 mm
- Interface: digital

This DHT11 temperature and humidity sensor features a calibrated digital signal output based on the temperature and humidity sensor's capacitance. It is integrated with a high-performance 8-bit microcontroller. Its technology ensures high reliability and excellent long-term stability. This sensor comprises a resistive element and a sensor for NTC humidity-compensated temperature measurement. It offers excellent quality, fast response, anti-interference capability and high performance.

Each DHT11 sensor features extremely precise calibration of the humidity calibration chamber. The calibration coefficients are stored in the OTP programme memory; the internal sensors detect signals during the process, and these should be referred to as calibration coefficients. The single-wire serial interface system is integrated for quick and easy operation. Small size, low power consumption and a signal transmission range of up to 20 metres make it suitable for a wide range of applications, even the most demanding ones. The product is housed in a 4-pin single-row pin header. Convenient connection via ; special packages can be provided according to user requirements. (<https://github.com/adafruit/DHT-sensor-library>)



Figure 8: The DHT11 temperature and humidity sensor.

5.1.4. The Peltier effect module: sensor and cooling unit



Figure 9: Peltier module.

Peltier systems:

Here, the direct current generated by the photovoltaic panels is used by the Peltier modules to produce cooling. Indeed, when a direct electric current passes through the junction connecting two conductors of different types, heat is either released or absorbed depending on the direction of the current. (A. MEDJO N. Brigitte) The heat released is cold on one side of the module and hot on the other. It is the cold side that is used for air conditioning.

These systems have the following advantages and disadvantages.

❖ Advantages

Cooling systems using Peltier modules have the following advantages:

- Use of non-moving parts, therefore low maintenance,
- A service life of around 100,000 hours of steady-state operation,
- No environmentally harmful refrigerants,
- Compact size and low weight,
- Quiet operation,
- Easy to use,
- Quick start-up and easy control,
- Reversible systems: can cool or heat depending on the direction of the applied current,
- Suitable for use in environments that are extremely harsh, highly sensitive or too small for conventional systems,
- Operate regardless of their position,
- DC power supply compatible with a photovoltaic power source.
- Affordable (from €50 for a Mibicool manufactured by Dometic),
- Operates at 12 V, 24 V and 220 V (with or without a transformer, depending on the model),
- Optimised space utilisation,

- Can be used when the boat is unstable (choppy seas, rolling, listing), (A. MEDJO N. Brigitte, p. 15).

❖ Disadvantages

- Low cooling capacity,
- Low efficiency (the system's coefficient of performance is 0.4),
- Condensation on the cold side,
- Poor performance in lowering the temperature (4 to 8°C below room temperature),
- Quite noisy due to the hot air extraction fan,
- Water accumulates in the tank.

5.2. Software and programming language

- C#: We will use the C# programming language to write the source code for our software, which will help to record temperature variations according to the products being stored and adjust the temperature profile according to the product.
- Arduino: We will use this integrated development environment to write the source code for our application, particularly for the sensors we presented in the sections on 'Prototype Overview' and 'Schematic Configuration of the Hardware on the Arduino UNO Board'.

5.3. Algorithms for the main methods

According to (ROHAUT, 2013, p. 67) , in computer science an algorithm describes what a computer must do to achieve a specific result. It is, in fact, the instruction to be given to the computer; instructions written in a language that is clear and intelligible to a human being. In this section, we will present some algorithms for the main methods of our embedded application. We will begin with an algorithmic description of the methods, then we will draw the corresponding flowcharts.

5.3.1. Algorithmic description of the system's main methods

Connecting the sensors to the microcontroller:

Before placing or connecting a sensor to the microcontroller, we must first check its operational status (initial state). There are two possibilities:

- If the sensor has an initial value of 1, we retrieve it and place it on the Arduino board, which will subsequently be controlled by a microcontroller; otherwise, if the sensor returns a value of 0 after verification – indicating zero power – the sensor is faulty.
- If the check returns a value of zero, the sensor is immediately discarded to prevent any malfunction.

5.3.2. Algorithmic diagram of the main methods

Temperature display:

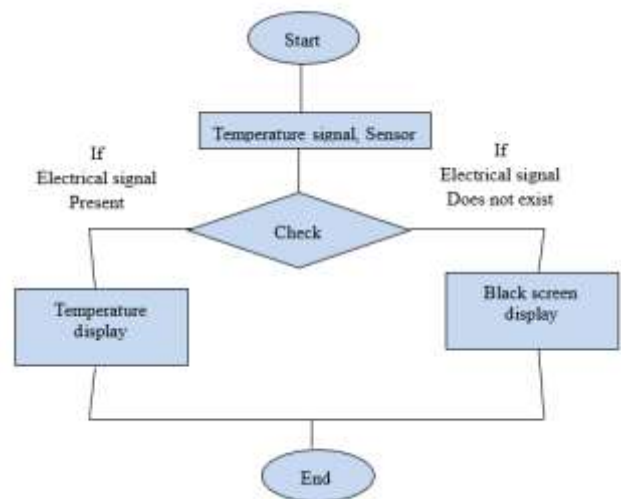


Figure 10: Algorithm diagram Temperature display.

5.4. Assembly and view of the Pharmaceutical and medical product storage kit

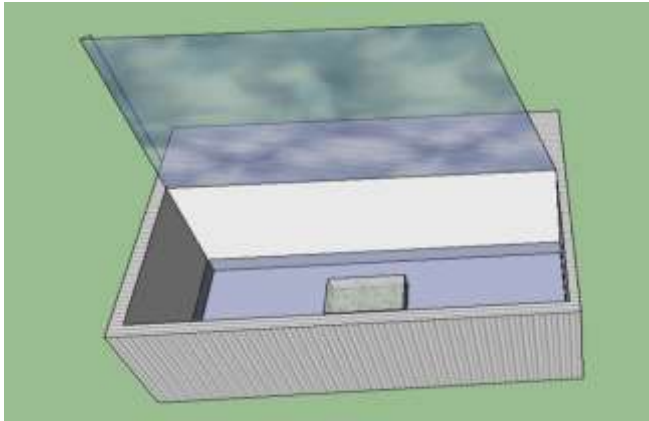


Figure 11: Front panels of the kit opening.

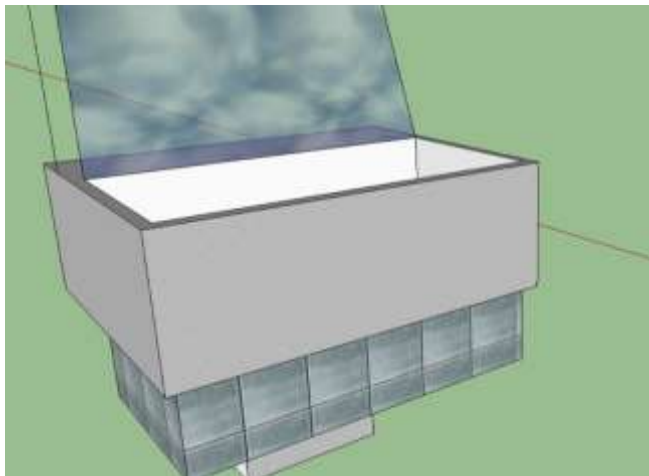


Figure 12: Front views of the kit's opening and radiator view.

5.5. Simulation and presentation of the prototype

5.5.1. Presentation of the models



Figure 13: View of the front.



Figure 14: View from inside the kit.

5.5.2. Presentation of the main form

A screenshot of a data recording form. The form is titled "PRODUIT" and contains several input fields for product information. The fields are: "CODE DU PRODUIT" (set to 001), "NOM DU PRODUIT" (set to antibiotique), "DATE DE CONSIGNA" (set to 27 novembre 2021), "TEMPERATURE DE CONSIGNA" (set to 15), "VARIATION DE LA TEMPERATURE" (set to 14-16), and "CODE CATEGORIE" (set to 1). There is also a "REMARQUE DU PRODUIT" field. The form is displayed in a web browser window.

Figure 15: Data recording form.

5.6. Test results

5.6.1. Temperature check of the kit

As mentioned in previous sections, the temperature check will take place automatically and the results will be displayed on the LCD screen.



Figure 16: Display of the kit verification results.

5.6.2. Temperature and humidity check of the storage kit

To monitor and check the humidity and temperature readings for the kit containing the medical products, the process takes place automatically, but the data relating to this process is always visible on the LCD screen.



Figure 17: Display of monitoring results; verification of the kit's humidity and temperature.

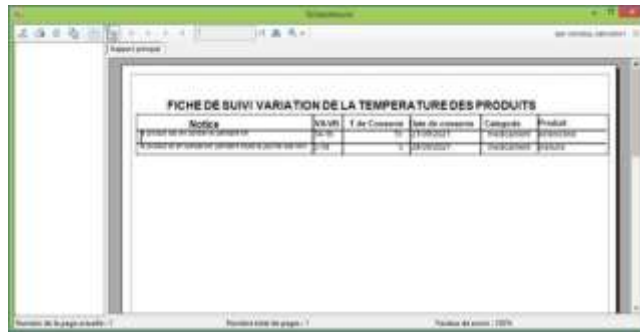


Figure 18: Temperature variation monitoring sheet.

Having simulated and tested the results of our simulation, the prototype is functioning correctly and the microcontroller has been programmed correctly, so that temperature monitoring will not be an issue, as the kit always maintains the temperature in accordance with the user's instructions.

6. Conclusion

At the conclusion of this study, which focused on the design of an embedded temperature control system for a low-power medicine storage kit, it is clear that embedded computing and automatic control technologies can provide a solution to the challenges associated with storing certain medical and pharmaceutical products in environments where access to electricity remains limited. The initial challenge was to identify a more cost-effective strategy for maintaining temperature conditions suitable for the storage of products requiring a controlled environment. To address this concern, the study adopted a methodological approach based on the Unified Process (UP) and SysML modelling. This approach enabled the system to be represented in a structured manner, identifying its various components, their interactions and the requirements the system must meet. The use case diagrams,

internal block diagrams, block definition diagrams and requirements diagrams thus helped to formalise the overall operation of the device prior to its construction. Functionally, the designed system is based primarily on a microcontroller coupled with an Arduino UNO board, temperature and humidity sensors, an LCD screen and a Peltier module designed to generate cooling. In particular, the device enables the sensors to be connected and checked, the system to be started, temperature and humidity to be measured, cooling to be generated, and the collected information to be displayed. This architecture enables the system to automatically monitor the internal conditions of the kit and present the data to the curator on the LCD screen. The use of the Peltier module is also a key element of the proposed solution. This module generates cooling using direct current and has several characteristics suited to an embedded system, notably a design with no moving parts, low maintenance requirements, a compact size, silent operation and the ability to be connected to a photovoltaic power supply. However, the study also highlights certain limitations, notably a relatively low cooling capacity, limited efficiency and a cooling capacity that depends on the ambient temperature.

The construction of the prototype enabled the integration of the various hardware and software components required for the system to function. The programming, carried out in the Arduino environment using the C# language, ensures the acquisition, processing, display and recording of temperature data. The algorithms developed specify, in particular, the sensor verification procedure and the process for displaying temperature. Finally, the simulation and tests carried out on the model demonstrate, according to the results presented in this study, that the prototype functions correctly and that the microcontroller has been programmed to ensure automatic temperature monitoring in accordance with the instructions provided by the user. The results are visible on the LCD screen, whilst temperature and humidity data can be monitored automatically. Thus, this research has enabled a progression from identifying a problem with the storage of medical products to the design, modelling and construction of a prototype embedded thermal control system. The proposed solution provides a technological foundation that could help improve the monitoring of storage conditions for medical products in healthcare facilities, particularly in environments facing energy constraints. Future work could focus on improving energy efficiency, integrating an autonomous photovoltaic power supply, optimising the cooling system, adding alert mechanisms in the event of temperature thresholds being exceeded, and testing the device under real-world storage conditions.

Bibliography

- (n.d.). Retrieved from <https://github.com/adafruit/DHT-sensor-library>.
- A. MEDJO N. Brigitte. (n.d.). *Specialised Master's Thesis in GEER. Specialisation: Renewable Energy, written by.*
- Didikey. (25 May 2021). <https://www.digikey.fr/fr/articles/choosing-using-advanced-peltier-modules-thermoelectric-cooling>. Retrieved from [www.digikey.fr: https://www.digikey.fr/fr/articles/choosing-using-advanced-peltier-modules-thermoelectric-cooling](https://www.digikey.fr/fr/articles/choosing-using-advanced-peltier-modules-thermoelectric-cooling)
- eduscol. (23 August 2026). [eduscol.education](https://eduscol.education.fr/sti/sites/eduscol.education.fr/sti/files/ressources/pedagogiques/6369/6369-sysml-un-langage-pour-la-modelisation-des-systemes-ensps.pdf). Retrieved from [https://eduscol.education.fr: https://eduscol.education.fr/sti/sites/eduscol.education.fr/sti/files/ressources/pedagogiques/6369/6369-sysml-un-langage-pour-la-modelisation-des-systemes-ensps.pdf?](https://eduscol.education.fr/sti/sites/eduscol.education.fr/sti/files/ressources/pedagogiques/6369/6369-sysml-un-langage-pour-la-modelisation-des-systemes-ensps.pdf)
- Gendre, L. (2013). *A language for systems modelling*. Éducol STI.
- GENDRE, L. (2013). *SysML: a language for system modelling*. Paris: Université Paris-SACLAY.
- GILLES ROY, J. (2009). *Database Design with UML*. Québec: Presses de l'Université du Québec.
- <https://slideplayer.fr/slide/1313429/> (25 April 2019). Retrieved 25 April 2019, from [https://slideplayer.fr: https://slideplayer.fr/slide/1313429/](https://slideplayer.fr/slide/1313429/)
- [https://www.ibm.com/](https://www.ibm.com/support/knowledgecenter/fr) (17 May 2019). <https://www.ibm.com/support/knowledgecenter/fr>. Retrieved 17 May 2019, from IBM Knowledge Centre: <https://www.ibm.com>
- Jean-Louis Bénard, L. B. (2002). *Project Management: Extreme Programming*. Paris: Eyrolles.
- Jean-Marie Domini & Lucien. (2013). *Thermodynamics and its Applications*. Edition Pierron.
- Laurent AUDIBERT. (2009). *UML 2.0: From Theory to Practice*. Paris.
- LAURENT, A. (2009). *UML 2: From Learning to Practice*.
- LeRobert. (2005).
- Longuet, Delphine, L. (2016–2017). *First-year course, 3rd year, Computer Science specialism: UML Course 2 – Use Cases*. Paris: Polytech Paris-Sud.
- Lucien Quaranta, L. (1997). *Experimental Physics and Thermodynamics*. Paris: Pierron.
- Marie François, S. (1824). *A seminal work for its time on the motoring power of fire*. Paris: Carnot.
- Michel Robert Discal, P. (2004). *Industrial Automation*. Borfon.
- NOUADJE, A. B. (2011). *Combination of Solar and Peltier Modules for Cooling in the Sahel*. Yaoundé: International Institute of Water and Environmental Engineering.
- NOUADJE, M. (2011). *Renewable Energy, Water and the Environment*. Yaoundé: National Institute of Engineering.
- Perrin, J.P., L. (2009–2010). *Les-electronicien*. In L. Perrin, J.P., *Microcontroller Course* (p. 6). Grenoble: Joseph Fourier University.
- Pierre Alain Muller and Nathalie, G. (2003). *Object-Oriented Modelling with UML*. Paris: Eyrolles.
- Raphael Yende, G. (2019). *Course on IT Project Management Methods*. Kinshasa: HAL ID: CEL-02004689.
- ROHAUT, S. (2013). *Algorithms: Fundamental Programming Techniques*. Paris: ENI Editions.
- Roques, P. & Vallée, F. (2007). *From Requirements Analysis to Design (4th ed.)*. Paris: Eyrolles.
- Simon Landrault, Eskimon, Hippolyte Weisslinger, (2014). *Arduino: First Steps in Embedded Computing*. Tunis: Eskimon's Blog, 19 June 2014 edition.
- Traore, David W. (2014). *Phosphorus-based embedded heat generation systems*. Duala.