2021

INDOOR CLIMATE CONTROL SYSTEM FINAL DESIGN REPORT OF THE BIO-ISO PROJECT

JIACONG LI

Student nr.	**: 82749**
Internship company	**: HZ University of Applied Sciences**
Major	**: Engineering Mechatronics**
Completion date	**: 9 June 2021**
Version	**: Final 7.0**

Student/intern

Name	**: Jiacong Li**
Student nr.	**: 82749**
Major	**: Mechatronics**
Study year	**: 4**
University	**: HZ University of Applied Sciences**
Thesis company	**: HZ University of Applied Sciences**
Email	**: li0042@hz.nl**
Tel.	**: +31(0)645637440**

In-company mentor

Name	**: W.C.M Heeren**
Function	**: Researcher**
Department	**: Lectorate of bio-based building**
Company	**: HZ University of Applied Sciences**
Email	**: marijn.heeren@hz.nl**
Tel.	**: +31(0)6487560610**

1st examiner

Name	**: J.P.C. Wubben**
Email	**: wubb0002@hz.nl**
Tel.	**: +31(0)627369234**

2nd examiner

Name	**: P. van der Heide**
Email	**: p.van.der.heide@hz.nl**

Content List

Figure list

Table list

Foreword

In daily life, the amount of carbon dioxide emitted from building heating, cooling and electricity consumption accounts for 47% of the total building carbon dioxide emissions. This makes people pay too much attention to energy conservation and emission reduction, which leads to the neglect of a pleasant indoor living environment.

Healthy living needs to start with a healthy indoor climate. Biobased insulation (Bio-Iso) products can make a significant contribution to this. Biobased insulation products, in contrast to synthetic and mineral products, have so-called hygric properties. The insulation material can absorb or release moisture depending on the relative humidity of the environment. In the 'Bio-Iso' project, biobased insulation materials are scientifically tested in various combinations. With these proven favorable properties, producers of biobased insulation materials can distinguish themselves in existing and new markets. (CoEBBE, 2020)

This report shows one of the most important parts of the 'Bio-Iso' project, the designing process of the project test setup, indoor climate control system.

The whole report is divided into thirteen chapters. The first chapter makes a short introduction of the whole project and tells the develop direction and desired results of the project. The second chapter contains the theories might be used in the project. The third chapter shows the method used in the whole project. From chapter 5 to chapter 11, these chapters show the process how the ideas are generated, transformed into concepts, and realized in the real life. The final products will be shown and introduced at the end of the report.

The test plans and results are contained are presented in the Appendix. For all the program codes can be found in the attachments.

1 Introduction

1.1 Background

Nowadays, building contractors always choose traditional building materials in most construction projects such as concrete and bricks of clay. This kind of materials have relatively low prices compared with bio-based materials and the functionality and reliability of them have been proved because of their widespread usage.

As a kind of innovative material, bio-based materials are, in contrast to synthetic and mineral products, healthier and have so-called hygric properties which are known as the ability to 'breathe'. The insulation material can absorb or release moisture depending on the relative humidity of the environment. What is more, bio-based materials are non-toxic or less toxic (Zoeter, 2021).

But the novelty of the materials also becomes the reason bio-based materials are less chosen. According to EU regulations, a contractor is responsible for the warranty of the safety of the construction even years after completion of the building. This means the functionality and reliability of bio-based materials need to be proved.

1.2 Problem statement

In order to prove the functionality and reliability of the biobased materials, the properties of them in different temperatures and moistures need to be tested. So the sketch of the container in figure 1 will be used as the test place for researching the properties of a bio-based material wall.

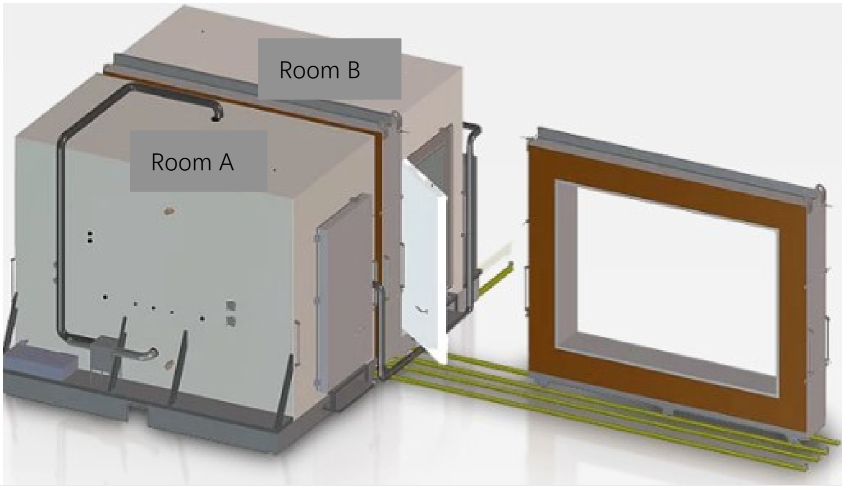


Figure 1 The container for the bio-based material properties tests. (Technik, 2021)

This container can be separated into two parts by inserting a frame filled with a bio-based material wall into the middle of the container. One room (room A) will be used to simulate an environment (outdoor environment) with extreme temperature and moisture (-10°C-35°C, 15%-100% RH) situation. The other room (room B) will be used to simulate an environment (indoor environment) with normal temperature and moisture situation (15°C-25°C, 50% RH) which is related to this project. So an air condition system needs to be build

But because of the special requirements for the test, the air-conditioning devices and systems sold in the market now are not fit for this project. In the following paragraphs, the features that the test requires but also the search for the air-conditioning devices and systems for sale now prove where the problems are and are explained why the required devices are not available yet on the market.

First, the test not only focuses on temperature and moisture control, the high accuracy (accuracy with resolutions of 0.01°C and 0.01% RH are high) container ambient and bio-based material wall internal temperature and moisture monitoring functions are also needed. But the air-conditioning devices and systems for sale now only have low accuracies with resolutions of 0.5°C and 1% and do not have the function of bio-based material wall temperature and moisture monitoring.

Second, before the test, several groups of temperature and moisture data with a fixed chronological order will be set. The controller of the air-conditioning system should control the temperature and moisture in the container according to these groups of data at a different time and repeat the same action cycle and cycle. But the air-conditioning devices and systems for sale now can only realize semi-automatization which means only one command can be executed once and the devices will stop and wait for the next command.

Third, because the whole test should not be affected by external environmental factors. This means the container should be enclosed and people cannot go in unless there are emergencies. And the long-distance remote control and wireless temperature and moisture data transmission are needed. But the air-conditioning devices and systems for sale now can only be remotely controlled in a short distance and cannot realize temperature and moisture data storage and transmission.

With these problems need to be solved, the following main and sub-questions are given to help to understand the project:

1.2.1 Main question

What design is feasible and affordable to realize the functions of temperature and moisture control, monitoring, and related data storage and wireless transmission in an enclosed 20ft container (L * W * H = 6.1 * 2.4 * 2.6 with a surface of 14.64 m^2 and a volume of 38.064 m^3)

1.2.2 Theoretical sub-questions

- What components can be used to control temperature?
- What components can be used to adjust moisture?
- In what way will the collected data be stored and uploaded to a data logger?

1.2.3 Empirical sub-questions

- What is a possible way to build communication between sensors, controllers, and the climate control components?
- What is a possible way to control moisture at a constant value when the environment temperature always changes?
- What is a possible way to measure the moisture contained in the bio-based material rather than only measuring the moisture of the surfaces of this material?

1.2.4 Analytical sub-questions

- What is a possible way to use the temperature control components to adjust the temperature rapidly with low money and power cost?

1.3 Goals and objectives

This paragraph introduces the goal and objectives which will be reached at the end of the project.

1.3.1 Goal

The goal of the project is to design and build a climate control system that can be used to simulate and monitor an indoor environment with different temperatures and moistures for a room of 14.64 m² (2.6 m high), meanwhile, measure the temperature and moisture inside the bio-based material wall and storage all the temperature and moisture data.

1.3.2 Objectives

Build a long term used stable indoor climate control system with the following functions:

- Rapid and accurate environment temperature and moisture control (temperature from 15 °C to 25°C and moisture at 50%).
- Container environment and bio-based material temperature and moisture monitoring.
- All the temperature and moisture data storage and transmission.
- Automatically cyclic operation according to set temperature and moisture data and time.

1.4 Boundaries

The boundary of the project is limited to designing a climate control system in order to simulate an indoor environment rather than researching the physical and chemical properties other than temperature and humidity, such as ductility, the flexibility of bio-based materials themselves.

Figure 2 shows the assignment division of the container climate control system:

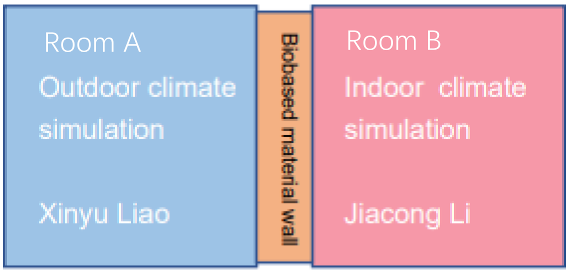


Figure 2 Sketch container test system of the BIO-ISO project

The red part is the part of the container used for the indoor climate simulation and is related to this project. In this project, the realization of the following functions is within the project boundaries: container ambient temperature and moisture control, container ambient and bio-based material wall internal temperature and moisture monitoring, all temperature and moisture data storage and transmission. The contents of the blue part (Room A), outdoor climate simulation are out of the project boundaries and Xinyu Liao, another team member, will be in charge of it. And there is no direct connections between Room A and Room B. Two rooms are only used to simulate two different environments at the same time.

1.5 Clients requirements list

This paragraph lists in table 1 the requirements from the client. These requirements are separated into two types, demands, and wishes. The demands are requirements that need to be realized and wishes are requirements that can be skipped during the project. All these requirements are summarized by myself after discussion with clients and approved by clients.

For the wishes, these are some requirements that do not have to be met because of the consumption of money or time which might not be able to realize because of not enough time or budget, or the special requests needed to realize the requirements. Such as the measurement of the heat flux. It needs high precision and sensitivity sensors which are not easy to buy and use.

Table 1 Clients requirements

List of clients' requirements		Demand	Wish
1	The system needs the function of temperature control.	√	
2	The system needs the function of moisture control.	√	
3	The system needs the function of temperature monitoring.	√	
4	The system needs the function of moisture monitoring.	√	
5	The system needs the functions of data storage and display.	√	
6	The temperature control ranging is from 15 °C up to 25°C.	√	
7	The moisture needs to be controlled at 50%.	√	
8	The system needs to adjust the temperature according to the preset serial of data automatically and do it as a circulation.	√	
9	The heat flux in the bio-based material wall needs to be measured		√
10	The system needs to simulate the day and night cycles.		√
11	The system needs to simulate the season's cycles.		√
12	The system needs to be built with completely new components.		√
13	The system can control up to 14.64 m² (2.6 m high)	√	

2 Theoretical framework

This chapter introduces all the theory might be used during the project.

2.1 Basic definition

This paragraph introduces the basic definitions of the relevant physical entities and values.

2.1.1 Temperature

Temperature is a physical quantity that expresses hot and cold. It is the manifestation of thermal energy, present in all matter, which is the source of the occurrence of heat, a flow of energy when a body is in contact with another that is colder or hotter. (ThoughtCo, 2010)

2.1.2 Moisture

Moisture is the presence of a liquid, especially water, often in trace amounts. The characteristic of being slightly wet or damp. Moisture also refers to the amount of water vapor present in the air. (Biology online, 2021)

2.1.3 Heat flux

Heat flux or the thermal flux, sometimes also referred to as heat flux density, heat-flow density, or heat flow rate intensity is a flow of energy per unit of area per unit of time. Heat flux also is the thermal energy transferred from one substance to another per unit time and area denoted by temperature change measured in watts per meter squared units. In simple terms, it is the heat transferred per unit area. (CORROSION, 2020)

2.2 Temperature and moisture measure

This paragraph introduces some temperature and moisture measure components and their working principles.

2.2.1 Temperature measure

Three kinds of thermometers can be used in the project: metal expansion-based sensor, thermistor temperature sensor, thermocouple temperature sensor.

Metal expansion-based sensor: The metal develops a corresponding stretch in response to a change in ambient temperature, so the sensor can signal this reaction in different ways to realize the temperature measurement.

Thermistor temperature sensor: Thermistors are semiconductor materials, most of which have a negative temperature coefficient, that is, the resistance value decreases with increasing temperature. It is the most sensitive temperature sensor because of the large resistance changes caused by temperature changes (see figure 3).

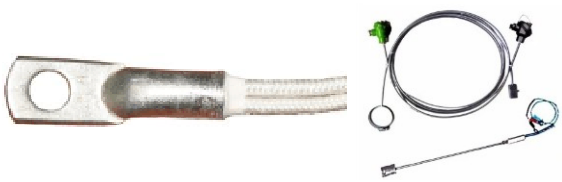


Figure 3 Thermistor and thermocouple temperature sensors (cc, 2019)

2.2.2 Moisture measure

The moisture sensor is divided into two kinds: resistive and capacitive.

The characteristic of moisture-sensitive resistance is that the substrate is covered with a film made of moisture-sensitive material. When the water vapor in the air is adsorbed on the moisture-sensitive film, the resistivity and resistance value of the element will change, and the moisture can be measured by using this characteristic.

Moisture-sensitive capacitors are generally made of polymer film capacitors. Common polymer materials include polystyrene, polyimide, tyrosine acetate fiber, and so on. When the ambient moisture changes, the dielectric constant of the moisture-sensitive capacitor changes, so that its capacitance also changes, and the capacitance change is proportional to the relative moisture. (cc, 2019)

For solid moisture measurement, Pin-type moisture meters can be used. These devices have two pins on the instrument, which are used to penetrate the test surface at the desired depth. The moisture is measured at the depth of the head of the contact pins. These meters use the principle of electrical resistance to measure the moisture by measuring the conductivity between the pins. The tips of the pins are relatively sharp, uninsulated, and penetrate the surface for a sub-surface reading (see figure 4). (GRAINGER, 2019)

Figure 4 Moisture meter (GRAINGER, 2019)

2.3 Temperature and moisture control

This paragraph introduces some temperature and moisture control components and their working principles.

2.3.1 Temperature control

For the system cooling, the working principle of the fridge and air conditioner must be mentioned firstly. Most cooling systems sold on the market all use the compressor as the core of the cooling system. All air conditioners have four basic components: a pump, an evaporator, a condenser, and an expansion valve. All have a working fluid and an opposing fluid medium as well. (Rude, 2010)There is refrigerating fluid cycled in this system to realize cooling (see figure 5). (Engineering360, 2012)

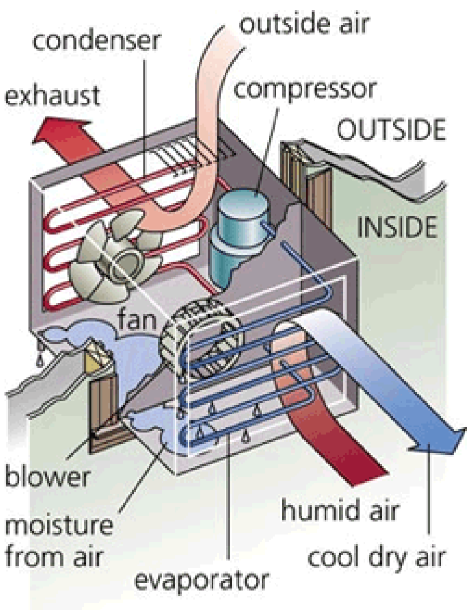


Figure 5 Working principle of an air conditioner (Engineering360, 2012)

The refrigerating fluid is input into the system and the compressor pressurizes the fluid. The fluid dissipates heat at the condenser and absorbs heat at the evaporator. The refrigerating fluid will change from liquid to gas and change back in one cycle to transfer the heat.

Another cooling method is using the Peltier devices which use the thermoelectric cooling method with low money costs (see figure 6).

Figure 6 A Peltier device

Thermoelectric cooling uses the Peltier effect to generate heat flow at the junction of two different materials. The Peltier cooler, heater, or thermoelectric heat pump is a solid-state active heat pump that transfers heat from one end of the device to the other, and the power consumed depends on the direction of the current. It can be simply understood as: under the action of an applied electric field, the electrons have directional movement, bringing a part of internal energy to the other end of the electric field (see figure 7 and 8).

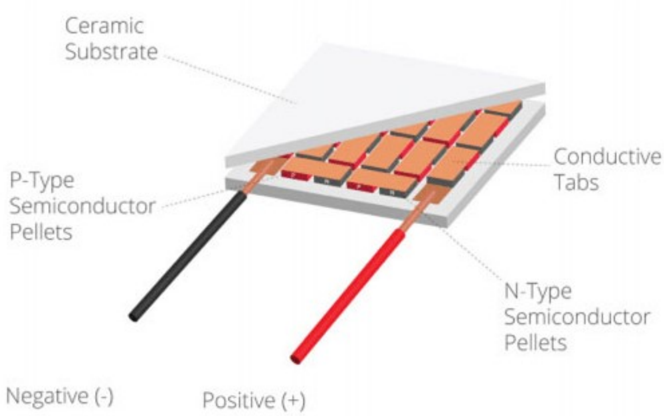


Figure 7 The working principle of a Peltier device 1 (CUIDevices, 2014)

Such instruments are also called Peltier devices, Peltier heat pumps, solid-state refrigerators, or thermoelectric coolers (TEC), and are sometimes called thermoelectric batteries. It can be used for both heating and cooling. (CUIDevices, 2014)

This means the Peltier device can be also used as the heating device of the system in this project. Compared it used as a heating device with using it as a cooling device. There is higher temperature changing efficiency and heating capacity.

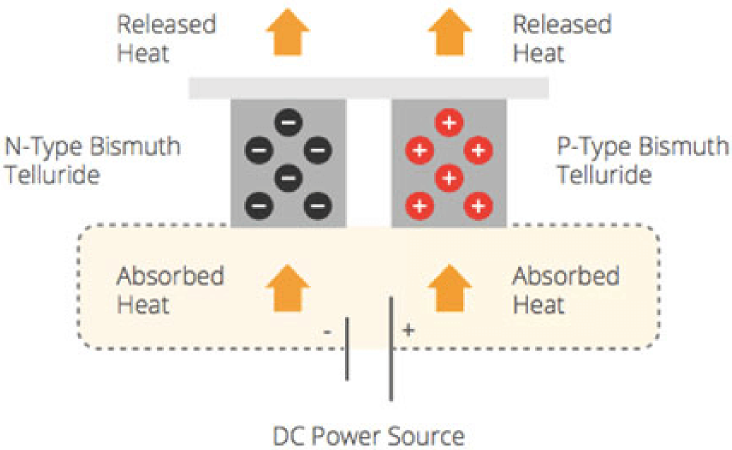


Figure 8 Working principle of a Peltier device 2 (CUIDevices, 2014)

2.3.2 Moisture control

In order to increase the moisture, the most direct and effective method is atomizing water with a component called the atomizer.

Ultrasonic atomizer makes use of electronic high-frequency oscillation (the oscillation frequency is 1.7MHz or 2.4MHz, beyond the range of human hearing, and the electronic oscillation is harmful to the human body) (xiaominge98, 2021). Through the high-frequency resonance of the ceramic atomizer, the liquid water molecular structure is broken to produce natural and elegant water mist, without heating or adding any chemical reagent. (see figure 9)

Figure 9 Atomizers (xiaominge98, 2021)

For decreasing moisture, there are two common methods. The first one is using the condenser. Blowing air into the condenser, and the moisture in the air will freeze and condense. The dehumidified air is then heated and discharged again (see figure 10). (Lingfeng, 2020)

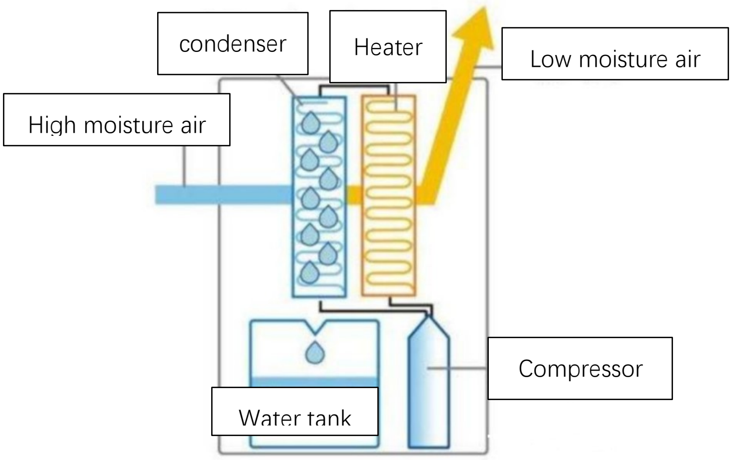


Figure 10 The working principle of a moisture decreasing system (Lingfeng, 2020)

The second method to realize this function is using a desiccant. Blowing the wet air into the containers with desiccant and build several filter circles to ensure the air is fully dried. The common desiccant has silica gel, calcium chloride, and so on.

2.4 Heat flux measure

The heat flux sensor can be used to measure the heat flux. The most common heat flux sensor is the thermopile heat flux sensor. The principle of the type of sensor is as follows: when a heat flow passes through the heat flow sensor, a temperature gradient is generated on the thermal resistance layer of the sensor, and the heat flux through the sensor can be obtained according to the Fourier blade law (see figure 11 and 12).

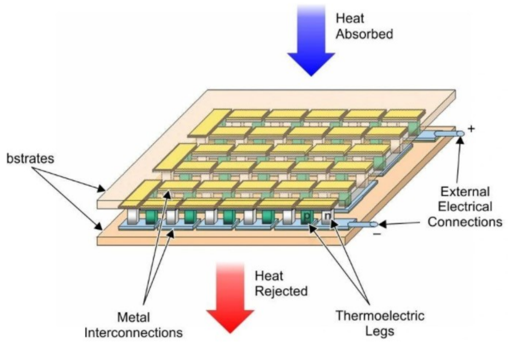


Figure 11 The working principle of a flux meter (Fluxteq, 2021)

In order to improve the sensitivity of the magnitude of a measurable heat flux of sensor, it is necessary to increase the output signal of the sensor, so many thermocouples need to be connected in series to form a thermopile, so the measured temperature signal on both sides of the thermal resistance layer is the series of all the thermocouple signal superposition one by one, the signal can reflect the average characteristics of multiple signals. (Fluxteq, 2021)

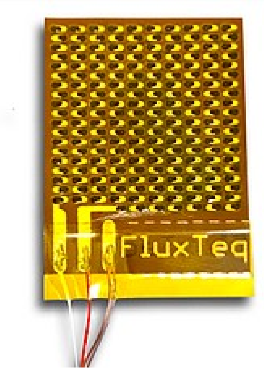


Figure 12 A flux meter (Fluxteq, 2021)

3 Research method

The whole project is executed with the 'Delft Design Method'.

3.1 Structure

The ‘Delft Design Method’ is a drill-down structure method with a total of six phases: start-up phase, analyzing phase, idea phase, concept phase, materialization phase, realization phase. The evolvements of every phase basis on the last phase. (See figure 13)

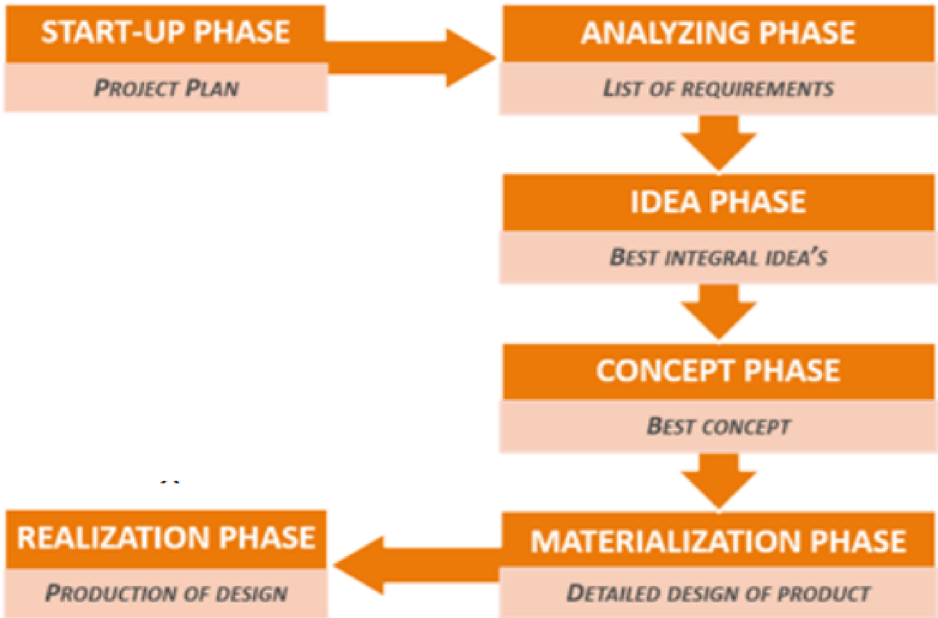


Figure 13 The structure of the Delft Design Method

3.2 Reason of choice

This project mainly focuses on designing a new system rather than a test. Compared with the other methods such as V-model which starts from a shallow layer to a deep layer with a progressive layer test to validate a conclusion and a theory, the ‘Delft Design Method’ can help users to transfer their thinking into reality step by step with several interlocking phases. Especially all the components in a system are all coordinating with each other. All the components need to be considered on the same layer. With this method, tests are still needed in the project to prove the function of intermedia or final products, but they are not used to validate conclusions or theory. What’s more, because I have more experience with this method compared with other methods. So the ‘Delft Design Method’ is chosen.

3.3 Activities and deliverables

This paragraph introduces the activities contained in every phase and corresponding deliverables.

3.3.1 START-UP PHASE

In this phase, with the start of the project, the project problem, aims, boundaries, and requirements from clients need to be clarified.

Deliverables:

i. A project research proposal
ii. An initial project plan

3.3.2 ANALYZING PHASE

In this phase, client requirements need to be analyzed and all the project-related information needs to be looked through to understand the present situation and desired result. Make a list to roughly settle the development directions with every client requirement. An initial system structure flowchart should also be made,

Deliverables:

i. A development directions introduction
ii. System structure flowcharts

3.3.3 IDEA PHASE

In this phase, based on the development direction list from analyzing phase, professional knowledge and information need to be collected. Meanwhile, generate several ideas with every development direction and make a list. An initial material list should also be made. And also, several solutions to the empirical questions will be given in this phase.

Deliverables:

i. An idealist (contains some ideas of the solutions or answers to the empirical questions)
ii. An initial material list

3.3.4 CONCEPT PHASE

In this phase, the idealist needs to be evaluated. Choose the most reasonable and feasible idea and detail these ideas with detailed solutions about them. After making choice, simplify and optimize the material list with the budget. The final solutions to the empirical questions will be given in this phase.

Deliverables:

i. A concept list (final solutions or answers to the empirical questions)
ii. A final material list with budgets

3.3.5 MATERIALIZATION PHASE

Basing on the concept phase, a prototype needed to be designed according to the concepts and material list. After designing and purchasing materials, testing different components respectively to check if every part can work well and conform to concepts. And then, assembling every component as an entirety and design experiments that can verify its functions. Write test plan and report with feedback for every test. Make new designs and correct mistakes if it is necessary. And also, the “make or buy” decision is scheduled. This means that determined is what parts are bought in the market and what parts must be manufactured according to specification.

Deliverables:

i. Components test plans
ii. Component test reports and data logs
iii. An assembled prototype
iv. Result of the “Make or buy” decision

3.3.6 REALIZATION PHASE

After building and validating the function of the prototype, in this phase, the prototypes need to be turned into products. Some details need to be added such as the distribution structure and system appearance in order to optimize the prototypes and ensure they can be used in a long term.

Deliverables:

i. Final products (Build up a real hardware system as a model)
ii. Final products test plan
iii. Final product test report and data logger.

iv. Instructions of the final products
v. A final report (contains the solutions or answers to all the questions)

3.3.7 Deliverable list

In Table 2, all the deliverables in every phase are summarized.

Table 2 Deliverables per phase

START-UP PHASE
A project research proposal
An initial project plan
ANALYZING PHASE
A development directions introduction
System structure flowcharts
IDEA PHASE
An idealist (contains some ideas of the solutions or answers to the empirical questions)
An initial material list
CONCEPT PHASE
A concept list (final solutions or answers to the empirical questions)
A final material list with budgets
MATERIALIZATION PHASE
Components test plans
Component test reports and data logs
An assembled prototype
Prototype test plans
Prototype test reports and data logs
Result of the “Make or buy” decision
REALIZATION PHASE
Final products (Build up a real hardware system)
Instructions of the final products
A final report

3.3.8 Relation between research questions and phases

In Table 3, the relation between research questions and phases is summarized.

Table 3 Relation between research questions and phases

Main question
The main question will be solved in the realization phase and the solutions and answers will be given in the final report
Theoretical sub-questions
Theoretical sub-questions will be solved in the START-UP phase during searching information and writing the theoretical framework of the project proposal.
Empirical sub-questions
Several possible solutions to the empirical sub-questions will be given in the idea phase when generating ideas.
The final solutions of the empirical sub-questions will be determined in the concept phase when converting ideas into concepts.
Analytical sub-questions
The analytical sub-questions will be solved during the test of the components in the materialization phase.

4 Project plan

In figure 14, the project plan is given. It shows the start time and deadline of each step.

TASK	ASSIGNED TO	PROGRESS	START	END	DAYS
Description of task	JL	%			
Phase 1 START-UP Phase					
Research proposal		0%	01-02-21	24-02-21	24
Planning rev1		0%	01-02-21	14-02-21	14
Phase 2 Analyzing Phase					
Development directions instructions		0%	15-02-21	28-02-21	14
System structure flowcharts		0%	15-02-21	28-02-21	14
Phase 3 Idea Phase					
Idea list		0%	01-03-21	14-03-21	14
Initial material list		0%	01-03-21	14-03-21	14
Phase 4 Concept Phase					
Concept list		0%	15-03-21	31-03-21	17
Component test plans		0%	01-04-21	07-04-21	7
Component test, test reports and data logs		0%	08-04-21	30-04-21	23
Final material list with budget		0%	01-05-21	01-05-21	1
Phase 5 Materialization Phase					
Make or buy decision		0%	01-05-21	01-05-21	1
An assembled prototype		0%	01-05-21	17-05-21	17
Final program codes		0%	01-05-21	17-05-21	17
Concept report		0%	05-05-21	12-05-21	8
Phase 6 Relization Phase					
Final products		0%	20-05-21	05-06-21	17
Final products test plan		0%	20-05-21	05-06-21	17
Final products test report and data logger		0%	20-05-21	05-06-21	17
Instructions of the final products		0%	05-06-21	07-06-21	3
Phase 7 Project delivered					
Final report		0%	12-05-21	07-06-21	27

Figure 14 Project plan

5 Project analysis

This chapter will analyze clients' requirements and list future project development directions.

5.1 Client requirements analysis and future development directions

Table 4 is the client requirements list summarized according to the results of the communications and discussions with clients during several meetings.

These requirements are separated into two types, demands and wishes. The demands are requirements that need to be realized and wishes are requirements that can be skipped during the project.

Table 4 Clients requirements

List of clients' requirements		Demand	Wish
1	The system needs the function of temperature control.	√	
2	The system needs the function of moisture control.	√	
3	The system needs the function of temperature monitoring.	√	
4	The system needs the function of moisture monitoring.	√	
5	The system needs the functions of data storage and display.	√	
6	The temperature control ranging is from 15 °C up to 25°C.	√	
7	The moisture needs to be controlled at 50%.	√	
8	The system needs to adjust the temperature according to the preset serial of data automatically and do it as a circulation.	√	
9	The heat flux in the bio-based material wall needs to be measured		√
10	The system needs to simulate the day and night cycles.		√
11	The system needs to simulate the season's cycles.		√
12	The system needs to be built with completely new components.		√
13	The system can control up to 14.64 m² (2.6 m high)	√	

Analyzing according to the requirement 1 to 5, the composition of the main system can be generally divided into 3 sub-systems:

a. Temperature and moisture monitoring system.
b. Temperature and moisture control system.
c. Data transmission and storage system

And according to this partition, a system flow chart is given to show the desired structure and working principle (how to monitor, control the temperature and moisture in Room B) of the indoor climate control system (see figure 15).

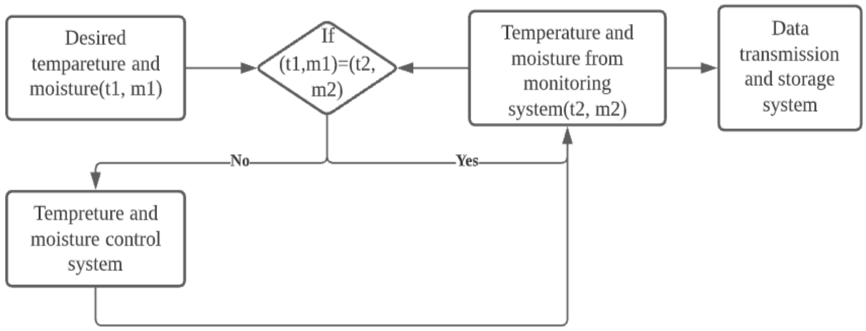


Figure 15 General system flow chart

This flow chart shows the general structure of the system. Users should tell the system the desired temperature and moisture they want. The system will collect the ambient temperature and moisture to make the comparison and adjust the ambient temperature and moisture to the desired value. Meanwhile, the temperature and moisture data will be stored and showed to the users.
KEYWORDS: Cooling and heating systems, humidifier, and dehumidifier.

Analyzing according to requirement 6, the temperature control range is not very wide. This means a rapid and continuous temperature change is available. And analyzing according to requirement 13, because the temperature control area affects the temperature control speed. So the next step of research should focus on how to control the temperature accurately and effectively in such an area. Something like the thermal load should be considered.

KEYWORDS: Small range temperature control and cooling efficiency.

Analyzing according to requirement 7, the moisture only needs to be kept at a constant value rather than being changed frequently. This means the moisture control system doesn't need high power to meet the needs of different moisture. The moisture circulation is a good direction to research.

KEYWORDS: Keep the moisture.

Analyzing according to requirement 8, the working module of the system should be considered. Should it work automatically to control the temperature of the container according to the preset temperature data before the system is launched? Or should it can be interrupted by a sudden order from the users? This will have a big effect on how to program the system controller.

KEYWORDS: Automatic operation, command detection, remote control

Analyzing according to requirement 12, it is not necessary to invent or develop any new components. Buying things from the markets, assembling, and controlling them is our way to do this project.

KEYWORDS: Assemble, control, organize.

Except for these requirements, how to guarantee the system can work a long time, how to check the working status of the system without close encounters, and also how to display the collected data to users are all important research directions in the next phase.

KEYWORDS: Data storage and display.

6 Idea generation

This chapter will show the ideas generated from the project analysis.

6.1 Detailed system flow chart

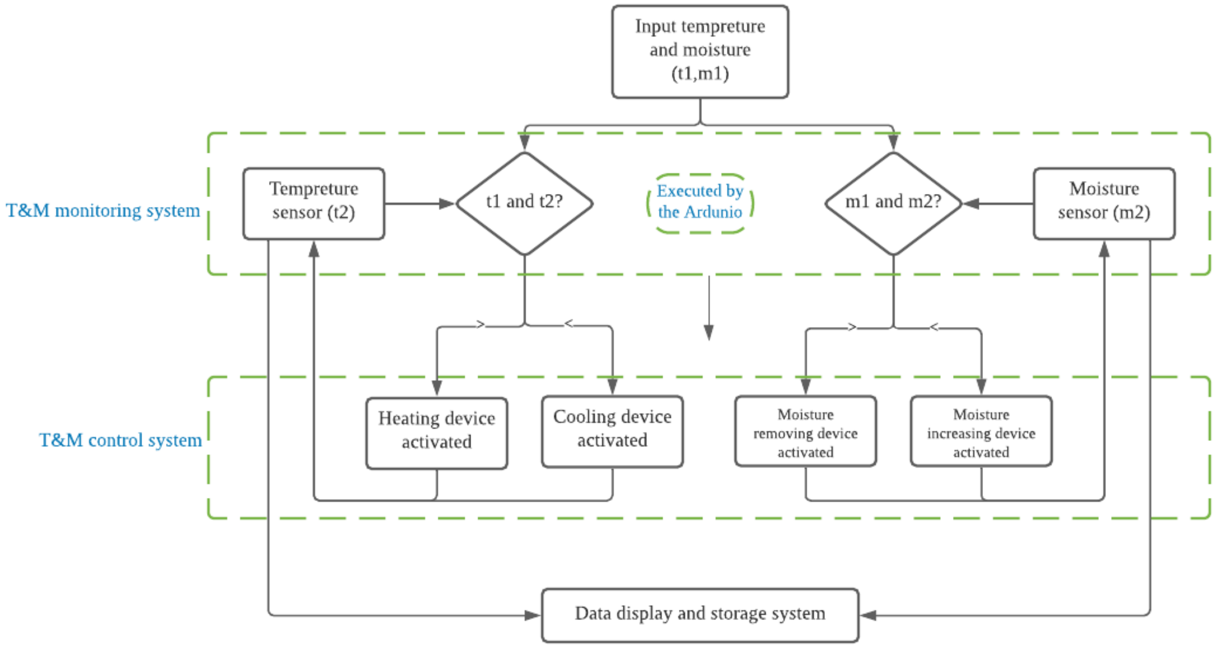


Figure 16 Detailed system flow chart

This is a detailed system flow chart (see figure 16) drawn basing on the general system flowchart. In this system, all the commands and activities are executed and controlled by one or several Arduino controllers.

This flow chart shows the detailed working principle of the whole system. The controller will receive desired data from users or store a serial of data and control the sensor and climate control devices to reach the desired temperature and moisture.

The monitoring system will measure the present temperature and moisture in the container. The controller will compare these data with desired data to execute the next command. The results of comparisons will affect the temperature and moisture control system. It will choose the trends to adjust the temperature and moisture according to the results. And meanwhile, all the data will be transmitted to dataloggers by a data transmission and storage system.

6.2 Idealist

Table 5 lists all ideas generated according to the analysis of clients' requirements and the keywords summarized in Chapter 2 after a lot of information searches are done.

Table 5 Idealist

No.	Ideas
Temperature and moisture control system	
	Cooling system
1.	Air-conditioner (compressor) cooling
2.	Semiconductor (Peltier device) cooling
	Heating system
3.	Air-conditioner heating
4.	Semiconductor (Peltier device) heating
	Moisture control
5.	Atomizer chip with driving circuits increasing moisture
6.	Filter with drying agent decreasing moisture
7.	Dehumidifier decreasing moisture
Temperature and moisture monitoring system	
8.	Temperature and moisture synthetic sensors measuring ambient temperature and moisture.
9.	Thermocouple measuring the temperature of the bio-based material wall.
10.	Solid moisture sensor measuring the moisture of the bio-based material wall.
11.	Solid moisture device measuring the moisture of the bio-based material wall and collecting data from the device.
Data transmission and storage system	
12.	Wireless data transmission replacing wire data transmission.
13.	WIFI module transmitting data to online datalogger.
14.	WIFI module receiving commands from the user and realizing remote control.
15.	SD card offline storing data
System control	
16.	One Arduino controlling the whole system.
17.	Several Arduinos controlling the whole system, one Arduino taking charge of one part of all the functions.

6.3 Idea introduction

This paragraph will introduce details of every idea on different aspects. Such as how will it be realized or what is its advantage or disadvantage?

6.3.1 Temperature and moisture monitoring system

Cooling system

Two possible methods can be chosen to decrease the temperature.

1) Air-conditioner (compressor) cooling
2) Semiconductor (Peltier device) cooling

The first method is using an air conditioner. According to the relationship between the size of the cooling space and the cooling capacity of the air conditioner, a space between 15 ㎡ to 30 ㎡ needs an air conditioner with the power between 1950[W] to 2640[W] to realize lowering the temperature by 10℃. And according to the prices of air conditioners from the internet. The cheapest one is about 260 EUR and with 2600[W] power. But an integrated device is difficult to be controlled from a software aspect. So it needs to be controlled from a hard ware aspect. Using an electromagnet to push the buttons on the device and control the electromagnet is a possible way.

Another way to realize the same function is buying components like the compressor, condenser, and so on, and assembling them by ourselves. A compressor with a cooling capacity of around 2500W is about 100-190 EUR and the other components are about a total of 70 EUR. And also, compared with buying an air conditioner directly, the self-made air conditioner can be easier controlled (control the power supply) and repaired. But it needs more time to assemble all the components and the safety of the system cannot be guaranteed.

Comparing these two ways, the system built firstly is easy to adjust temperature control but difficult to be controlled. The system builds in a second way can be easily controlled but needs more time to be built.

The second method is using the Peltier device. This method should be the final plan (plan B). Although the Peltier device can be easily controlled and has a very low price, its cooling capacity and cooling efficiency are very low. Especially it is related to the environment temperature which means its cooling capacity is not very stable. What's more, the working principle of the Peltier device is in fact the transfer of the heat from one side of the Peltier device to another side, which means if one side is cold, the other side is very hot. For example, if the cold side is 25℃, the hot side can be about 65℃. If keep this high temperature for several hours the Peltier device will be burned. So the heat dissipation devices are needed. But the test container is an enclosed container, if the Peltier device is used for cooling, its heat dissipation should not affect the temperature of the container. So how to guarantee the heat from the Peltier device can be dissipated out of the container, and without effects on the container inside temperature become a big problem.

Heating system

For heating systems, it is easier to realize compared with the cooling system. If the air conditioners are used, the heating function can be easily realized. Another way is to use Peltier devices. The heat dissipation problem now becomes an advantage. The only thing that needs to be careful about is keeping the air flowing to take away the heat from the Peltier device. Radiators can be used to improve heating efficiency. Radiators can increase the contact area with area to speed up the heat transfer to the air.

Moisture control

For moisture control, because the moisture only needs to be controlled at a fixed value, rather than changing rapidly in a range. So the moisture control doesn't need a big system to produce the moisture or remove moisture.

For increasing moisture, an atomizer with driving circuits will be used. Its working principle is atomizing water and mixing water particles into the air to increase the moisture. The volume needs to drive the atomizer is small and can be controlled by a controller like an Arduino directly.

For decreasing moisture, the first method is to use the filter with a drying agent and blow air into it. The drying agent will absorb the moisture in the air and blow dry air out. This is not an efficient way to change the moisture rapidly, but useful to keep constant at the same value. The disadvantage of it is the drying agent needs to be renewed which means it cannot work continuously. The second method is directly using the dehumidifier. With this method, the dehumidifier can work continuously when the water draining problem is solved.

6.3.2 Temperature and moisture monitoring system

The monitoring part can be separated into two parts. **The first part** is environment temperature and moisture monitoring. This part is relatively easy to realize with temperature and moisture synthetic sensors. Sensors are controlled by an Arduino controller.

The second part is bio-based material internal temperature and moisture monitoring. For temperature, an embedded thermal couple controlled with the Arduino can be used. But for moisture, as far as the present research, there is no efficient way to measure the material internal moisture with the Arduino controllable sensor. But there are independent devices sold on the market which contain the monitor, sensors, and data storage components. So, there should be a way to record or transfer the data from these independent devices into data that can be processed by Arduino.

6.3.3 Data transmission and storage system

With this subsystem, in order to guarantee the data of the whole climate control system can be monitored whenever the users want, a cloud database or an online data logger is needed. To realize this function, a WIFI module is needed to transmit data. And also, there is still the risk of data loss, so data backup is needed. SD cards, as the most convenient device which also has high data capacity, can be used to store data.

The data transmission system is not only used to display and store data but also used for command transmission between controllers. Because the system will be installed in a room of about 14.64 m² the long wire connection for a micro-controller is a little unwieldy. So the wireless data transmission is necessary.

The third function of the data transmission system is used for remote control. It needs an IoT platform for receiving commands from users and transmitting commands through the internet to the controller to control and monitor the system.

6.3.4 System control

The initial plan is to use Arduino as the controller to collect data from the sensor and control the temperature and moisture control system. As a kind of commonly used device, Arduino is easy to use and has a lot of support equipment, codes, software, or hardware designed by other Arduino users. This means it is easy to realize data transmission between sensor and controller and commands transmission between controller and controlled devices.

There are mainly two selections for Arduino control. **The first selection** is connecting all sensors and devices on one Arduino. This selection can guarantee the integrality of the whole system. Easy to install and integrated control are its advantages. **The second selection** is using several controllers to take charge of different functions. The advantages of this method are every part is almost independent, so the codes for every controller will not be complex. And also, the position of different sensors and devices will not be limited by the wires used to connect them. Wireless data transmission will be used in this selection.

7 Concepts from ideas

This chapter will introduce how the final concepts are transferred from ideas. According to concepts, the final product will be designed according to these concepts.

7.1 Final system flow chart

Figure 17 shows the final system flow chart. This flow chart is designed according to the final concepts which will be discussed in paragraph 7.2 &7.3. The final product will work according to this flow chart.

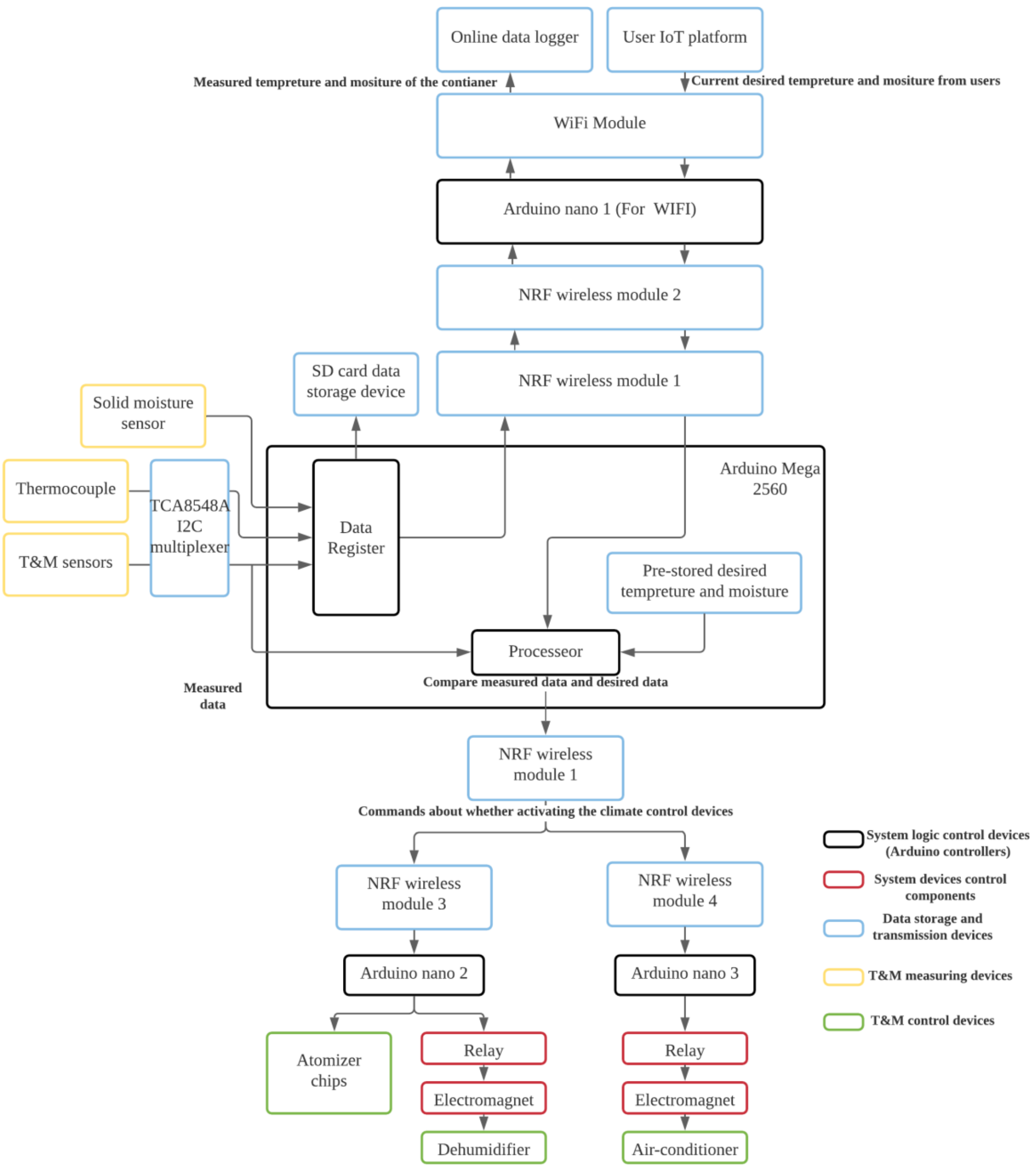


Figure 17 Final system flow chart

7.2 Final concept list

Table 6 lists all concepts filtered from the idea list in Chapter 3 after consulting detailed information from devices or components solders, researching about these devices and components, and testing them. Some ideas which are too difficult to realize or not fit for the project are abandoned and the rest of the ideas are summarized into concepts as follow:

Table 6 Concept list

No.	Ideas
Temperature and moisture control system	
Temperature control	
1.	Air-conditioner cooling and heating
Moisture control	
2.	Atomizer chip with driving circuits increasing moisture
3.	Dehumidifier decreasing moisture
Temperature and moisture monitoring system	
4.	Temperature and moisture synthetic sensors measuring ambient temperature and moisture.
5.	Thermocouple measuring the temperature of the bio-based material wall.
6.	Solid moisture sensor measuring the moisture of the bio-based material wall.
Data transmission and storage system	
7.	Wireless data transmission replacing wire data transmission.
8.	WIFI module transmitting data to online datalogger.
9.	WIFI module receiving commands from the user and realizing remote control.
10.	SD card offline storing data
System logic control	
11.	Several Arduinos controlling the whole system, every Arduino taking charge of one part of all the functions.
System device control	
12.	Electromagnets are used to press the buttons on the temperature and moisture control devices such as air-conditioner and dehumidifier.

7.3 Reasons for choosing the final concepts.

Part reasons of why these solutions are chosen for the final concepts will be explained in the paragraph.

7.3.1 Temperature and moisture monitoring system

Temperature control

As it mentioned in Chapter 2, two possible methods can be chosen to change the temperature.

1) Air-conditioner (compressor) cooling and heating

2) Semiconductor (Peltier device) cooling and heating

Finally, the first method, the air-conditioner temperature adjustment method is chosen. Because after a lot of information searches, buying compressors and other components to assemble them needs more time cost compared with using an air-conditioner. Especially, there is no supplier around Europe which can supply high-power compressors. It means buying compressors from the place out of Europe needs high money and time consumption. And an air-conditioner is enough for the project to adjust the temperature between 15℃-25℃. What's more, the insulation of an air-conditioner is easier than assembling all components.

When it comes to the usage of the semiconductor temperature adjustment, the Peltier devices are cheap and can be both used for cooling and heating, but its cooling efficiency is too low to realize the 10℃ temperature difference in a room about 30 ㎡ with only a few Peltier devices. If a lot of Peltier devices are used, the power consumption will be the biggest problem, and also the heat dissipation problem cannot be solved easily. Electrical safety cannot be guaranteed.

Moisture control

Using an atomizer to increase the air moisture can be easily controlled compared with using a humidifier. Arduino only needs to provide an atomizer with a power supply, the atomizer can work.

For the moisture decreasing, the drying agent is not fit for the project. Because the container used to simulate the indoor climate cannot be affected by the outside environmental elements. This means people cannot go into the container very often. So the refreshing of the drying agent becomes a big problem. That's why the dehumidified is chosen.

7.3.2 Data transmission and storage system

With this sub-system, wireless data transmission methods are frequently used. The first reason is to make sure all the devices can be placed anywhere in the container. So the wire data transmission methods are not fit for this requirement.

And also, because the place of the container might be far from the building which has WIFI signal around. So the data needs to be transmitted by wireless data transmission modules to cross the no WIFI signal area. On the contrary, Arduino needs to receive commands from users' IoT platforms. These commands also need to be transmitted by a WIFI signal and go through the wireless channel to reach the Arduino.

7.3.3 System logic control

For the system control, several Arduinos are decided to be used. The main reason is the port number on one Arduino is not enough to connect all sensors and devices. Especially the wireless data transmission method is used. So several controller is used to control different wireless data transmission models. What's more, different controllers take their own function will not cause too much clutter in the programming code. So there will be four Arduinos in the system.

The first one is an Arduino mega 2560. It will be used to do the most calculation, receiving measured data from the sensor, comparing measured data with desired results, and data communication between the other Arduinos, dataloggers, and users IoT platform.

The last two Arduino will be used for temperature control and moisture control and long-distance data transmission to online data logger with WIFI signal. They will execute the commands from the Arduino mega 2560 to activate the air-conditioner, atomizers, and dehumidifier, and also communicate with online data logger.

7.3.4 System devices control

The system devices control mainly means the control of the air-conditioner and dehumidifier. These devices are sold on the market and contain a lot of integration circuits which means the circuits of these devices cannot be changed and the devices cannot be controlled with simply code programming. So the relays and electromagnets are used.

Arduino will control the relay to control the power supply of the electromagnets. When the relay is powered, the electromagnet will be powered and triggered. The electromagnet will press the buttons on the air-conditioner and dehumidifier to realize device control. For more details of relays and electromagnets please check **'Appendix 11.3 System device control components'**

8 Final product

This chapter shows the final product designed according to the final concepts. It contains the introduction of the final product and the test of the final product. Before reading this chapter, please check Appendix 1 & 2 about details of the hardware and program used for the final product. This can help users to understand this chapter better.

8.1 Introduction of the final product

Figure 18 is the picture of the final product with its setup. The final product contains four main parts. And table 7 shows the compositions of these four parts. Figure 19-22 show the details of these four parts.

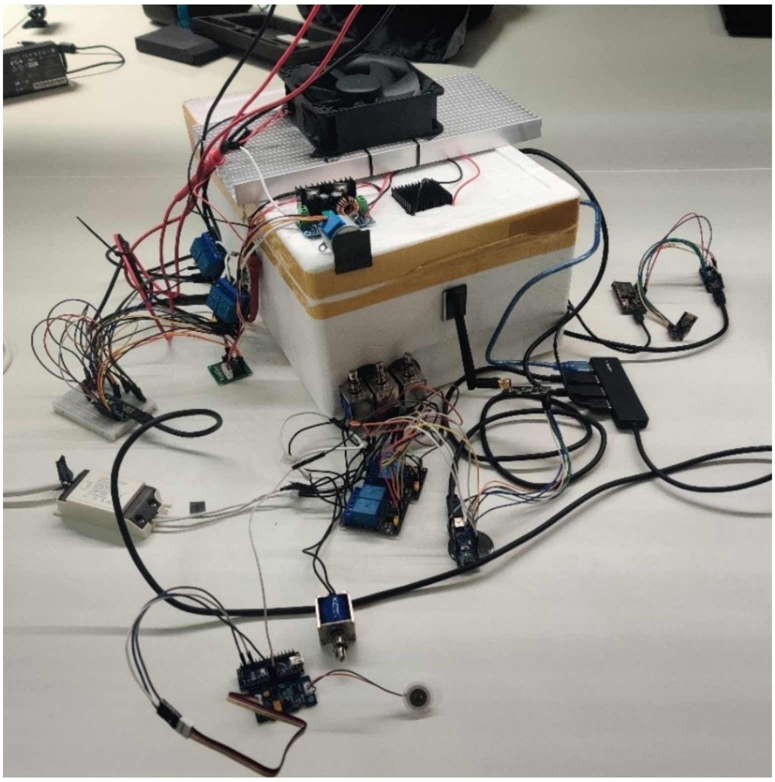


Figure 18 Final product test setup

Table 7 The composition of the final product

Part	Usage
Part 1	Main controller Arduino Mega 2560 with sensors
Part 2	Arduino nano 1 for online data storage and command reception.
Part 3	Arduino nano 2 for moisture adjusting devices control
Part 4	Arduino nano 3 for temperature adjusting devices control

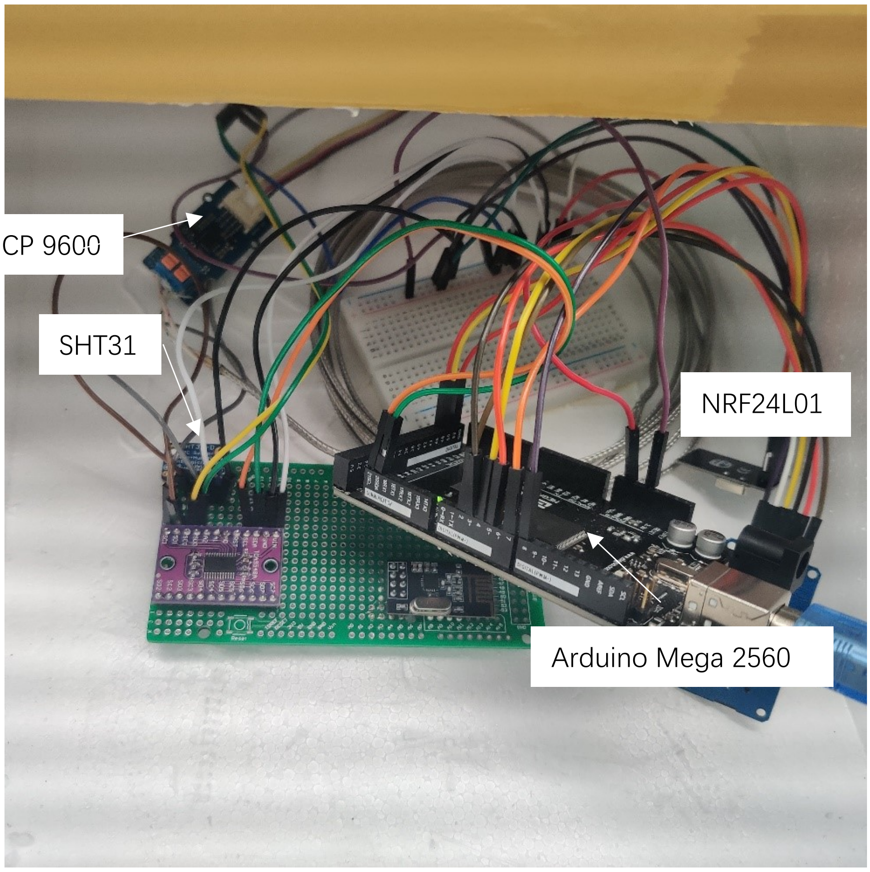


Figure 19 Final product part 1 (Be put in the foam box, which is not shown in Fig 18)

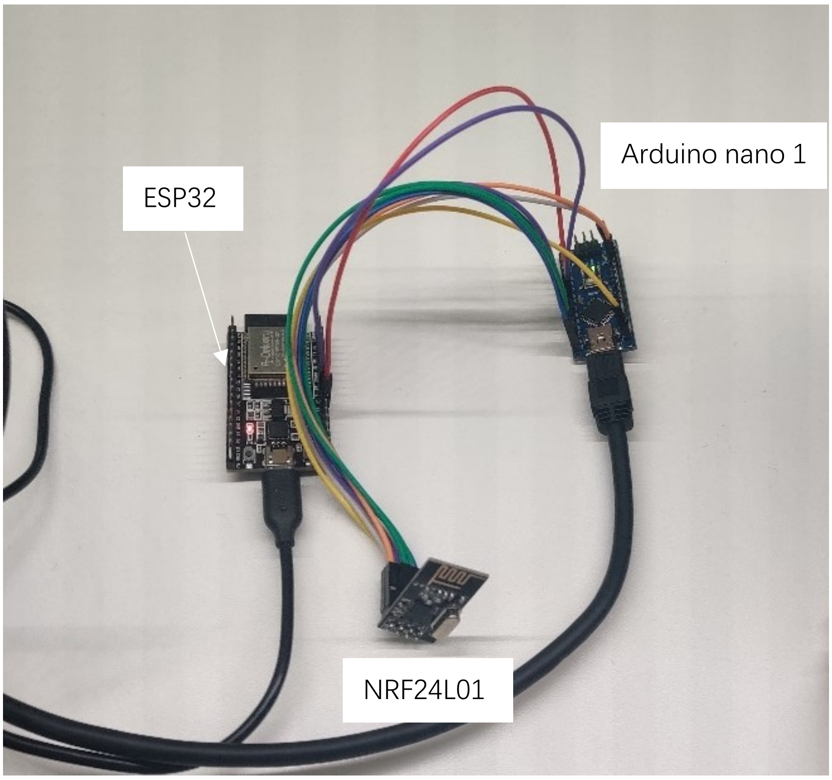


Figure 20 Final product part 2

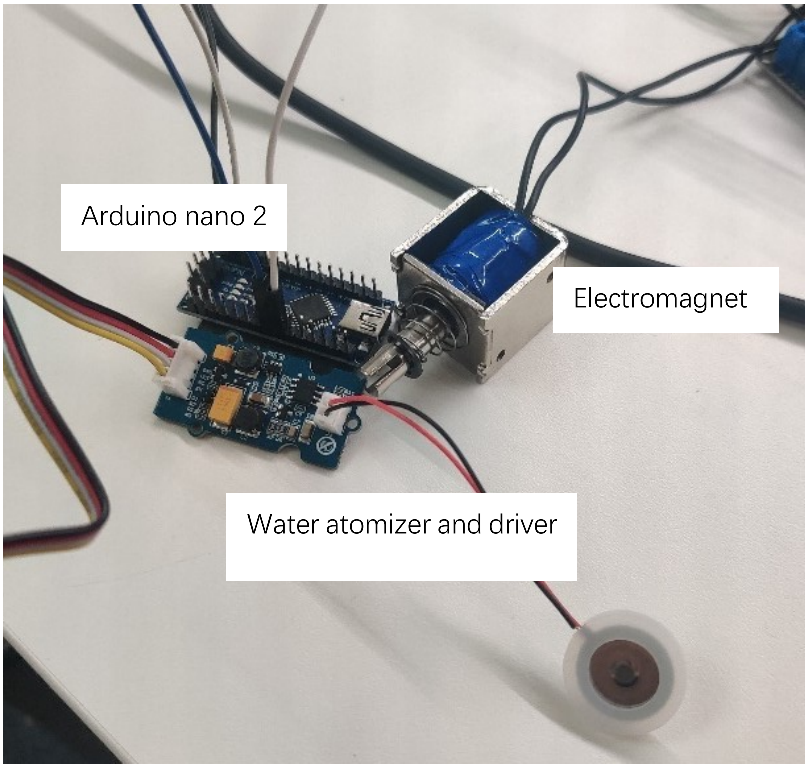


Figure 21 Final product part 3

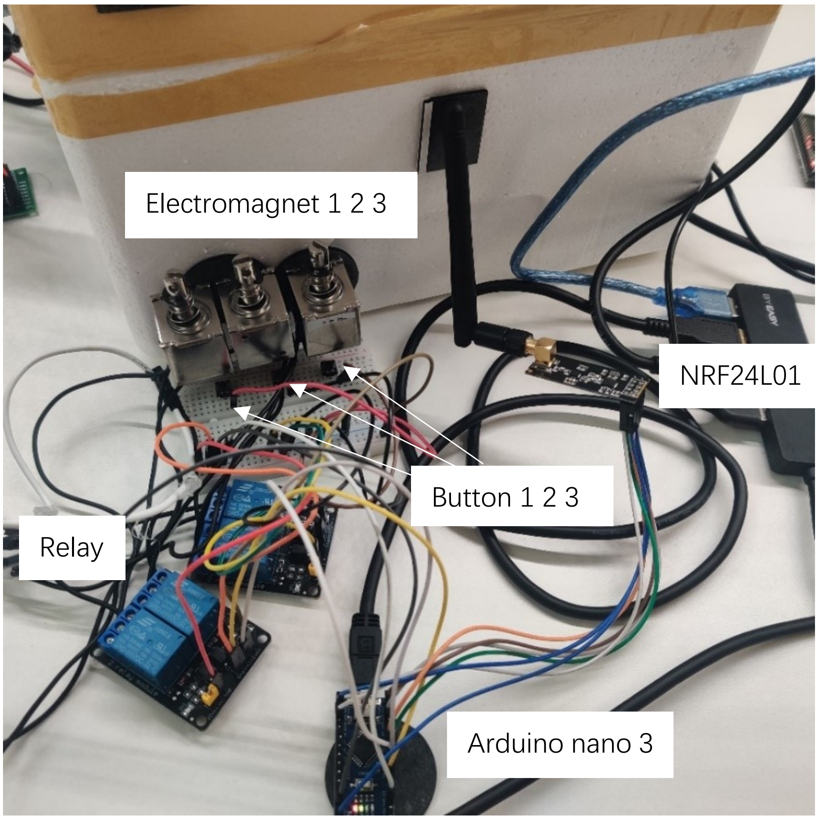


Figure 22 Final product part 4

The main controller Arduino Mega 2560 is set with a serial of temperature values. It will send commands to several sub-controller to adjust the temperature according to those preset values and repeat this in cycles.

The final product has functions of temperature and moisture monitoring, temperature and moisture control, climate data transmission, and storage. Before starting it, a serial of preset values of temperature and reserved execution time for adjusting the ambient temperature from one preset value to the next preset value should be input to the main controller. For example, in the final test, the preset temperatures are 26℃, 27℃, 25℃. The execution time is 5 minutes. This means in the first 10 minutes, the final product should adjust the ambient temperature to 26 ℃ and keep this temperature until the second 5 minutes come. Similarly, the final product will adjust the temperature to 27℃, 25℃ and repeat the whole process in cycles.

And especially in order to save power and avoid power loss, it can switch off the temperature and moisture control devices automatically when the temperature and moisture reach the desired value. Also, these devices will be launched again when the temperature and moisture deviate from the desired value.

8.2 The test of the final product

Because the real bio-based material container cannot be bought before the end of this project. So in order to verify the function of it. An alternative test setup for the final test is built. The test setup contains a foam box with two Peltier devices set on the cover of the foam box. The foam box is used to replace the real enclosed bio-based container room A. The Peltier devices are used to replace the air-conditioner for heating or cooling the container and making a temperature condition system (This temperature condition system is only used in the final test which is different from the temperature control system mentioned in previous chapters.) with relays and a voltage transformer. (The function of moisture control is not tested. Because the working principle of the program of moisture control is the same as temperature control. Especially the moisture only needs to be controlled at a constant. Another reason is there is not enough budget to buy a real dehumidifier to use in this test.)

Figure 22 shows part 4 of the final product which is mentioned in table 7. This part is controlled by Arduino nano 3 and these three electromagnets will press the buttons below them when Arduino nano 3 receives the commands from the main controller Arduino Mega 2560.

The buttons under the electromagnets represent the buttons on a panel of a real air-conditioner. These buttons are connected to the temperature control system used in this test. Table 8 shows the function of different buttons.

Table 8 Button functions

Button	Function
Button 1 (Mode control)	Control the working mode of the temperature control system. First pressing for heating mode. Second pressing for cooling mode. Third pressing for system power off.
Button 2 (Temperature up)	Increase the set temperature of the temperature control system. Pressing once the set temperature will increase by 0.5℃
Button 3 (Temperature down)	Decrease the set temperature of the temperature control system. Pressing once the set temperature will decrease by 0.5℃

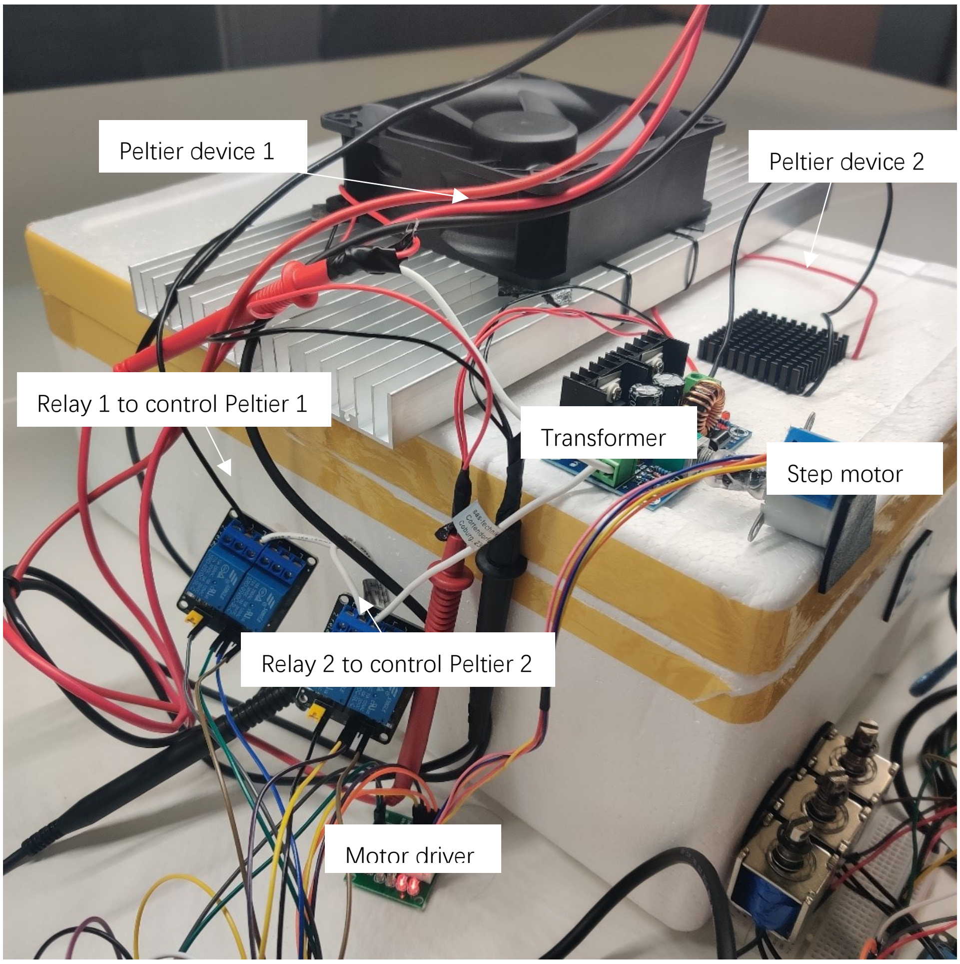


Figure 23 Test setup

Button 1 is connected to two relays used for switching working mode. Two relays are separately connected to two Peltier devices used for controlling their power supply. The Peltier device with a bigger heat sink and a fan on it is used for cooling (See figure 23 & 24). The other Peltier device is for heating.

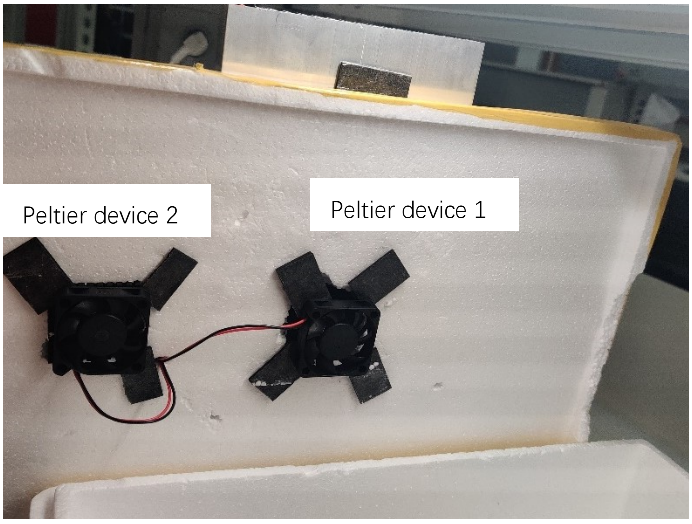


Figure 24 Inside of the test setup

Because the heating and cooling efficiency is related to the voltage of their power supply. So an adjustable voltage transformer is used. When the voltage transformer outputs a high voltage, the heating or cooling efficiency of the Peltier device is high. This means the ambient temperature of the foam box can reach a very high or very low value and vice versa.

The adjusting knob of the transformer is connected with the shaft of a step motor (see figure 25). The step motor is controlled by buttons 2 and button 3. For example, when button 2 (temperature up button) is pressed. The shaft of the motor will rotate a certain angle and drive the transformer adjusting knob to rotate. This will increase the output voltage of the transformer. This means when the system is working in heating mode, the Peltier device can make the ambient temperature of the foam box higher.

This is used to simulate the process of a real air-conditioner increase or decrease the ambient temperature when the set temperature of it is increased or decreased. When button 2 (temperature up button) is pressed, the set temperature of the temperature condition system is increased.

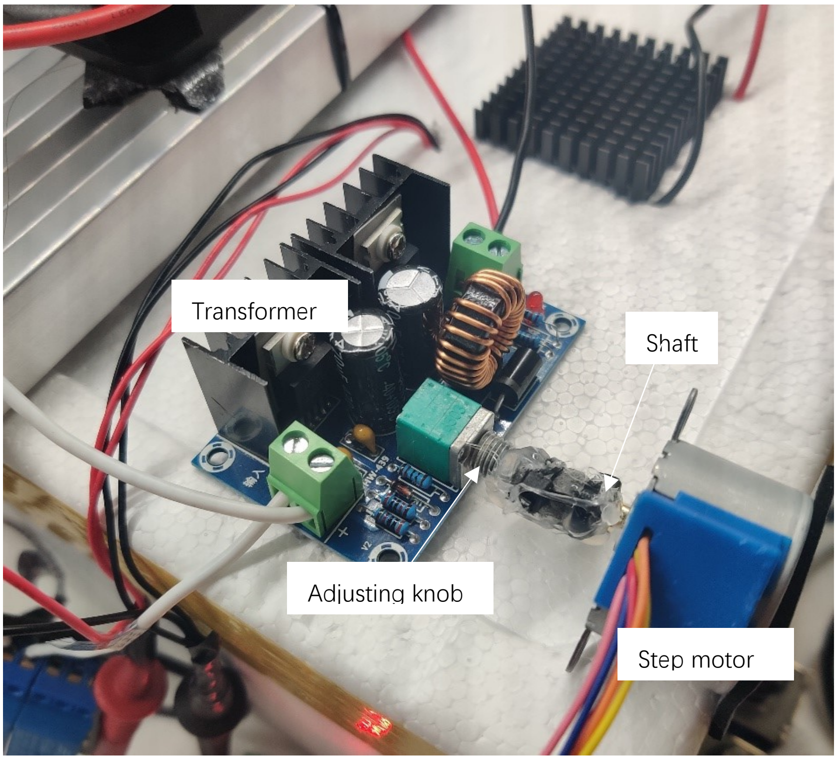


Figure 25 Voltage transformer and step motor

After a serial test of the transformer, it can be proved that by pressing the button five times (shaft rotating five times), the ambient temperature of the foam box can increase a random value from 2.3℃ to 2.7℃. For this test, it can be seen as a 0.5℃ change for every button pressing. Because the temperature of Peltier devices cannot be accurately controlled. So the ambient temperature which has + 0.1 or – 0.2 difference to the desired temperature is considered as the ambient reaching the set value. This means when the sensor measures the ambient temperature has + 0.1 or – 0.2 difference to the desired temperature, the temperature condition system will be switched off.

All the buttons and the transformer are controlled by another Arduino called Arduino nano test. To monitor this Arduino, users can see the changes of the working mode and the changes of the set temperature of the temperature condition system.

To monitor the Arduino Mega 2560, users can see the present ambient temperature of the container, the moisture, the desired temperature, and the progress of the execution period. It will output data once every 15 seconds. (see figure 26)

```
->
-> Temp *C = 25.09
-> Hum. % = 32.76
-> WALL TEMP:25.81
-> Done
-> Desired temp: 26.00
-> time:9
->
-> Temp *C = 25.12
-> Hum. % = 32.74
-> WALL TEMP:25.81
-> Done
-> Desired temp: 26.00
-> time:10
->
```

Figure 26 Monitor of Arduino Mega 2560

To monitor the Arduino nano 3, users can see the process of calculating the set temperature and the next actions of the electromagnets when a new execution period comes. (See figure 27)

```
->
-> **********next temp***********
-> stemp23.50
-> stemp24.00
-> stemp24.50
-> stemp25.00
-> stemp25.50
-> stemp26.00
-> Desired temp: 26.00
-> result temp: 26.00
-> push button times: 6
-> set temp adjustment finished
-> offmode: heatingoff
-> **********next temp***********
-> stemp26.50
-> stemp27.00
-> Desired temp: 27.00
-> result temp: 27.00
```

Figure 27 Monitor of Arduino nano 3

For the details of the test process, please see the **‘Appendix 13.5 Final test plan’**.

8.3 Final product results and analysis

This paragraph shows the final product results and results analysis.

8.3.1 The ideal working process of the final product

According to step c) of the **'Appendix 13.5 Final test plan --- 6. test steps'**. An ideal working process should be written.

So the ideal working process flowchart of the final product when it adjust the ambient temperature from 26℃ to 27℃ to 25℃ is drawn. (Click to see the picture). Table 9 and 10 are the summary of the ideal working flow chart that shows the actions of different electromagnets.

Table 9 Actions of different electromagnets in different execution periods

The working mode of the last execution period	The working mode of the next execution period	Electromagnet 1 (Button 1 Mode control)	Electromagnet 2 (Button 2 Temperature up)	Electromagnet 3 (Button 3 Temperature down)
Off	Heating	Press once	Press	-
	Cooling	Press twice	-	Press
Heating	Cooling	Press once	-	Press
	Off	Press twice	-	-
Cooling	Heating	Press twice	Press	-
	Off	Press once	-	-

Table 10 Actions of different electromagnets in the same execution period

Working process	Electromagnet 1 (Button 1 Mode control)	Electromagnet 2 (Button 2 Temperature up)	Electromagnet 3 (Button 3 Temper-ature down)
When ambient temp reaches the desired value			
Heating → Off	Press twice	-	-
Cooling → Off	Press once	-	-
When ambient temp goes higher or lower than the desired value when the working mode is 'Off'			
Off → Heating	Press once	-	-
Off → Cooling	Press twice	-	-

8.3.2 The data logger of the final test

Figure 28 shows a part of the result data logger of the final test. This is the main result of the final test. All data in the data logger are input by the tester according to the real results read from the monitors and test setup.

Ambient temperature	Trend	Desired temperature	Set temperature	Working mode	Eletromagent 1 (button 1 mode control)	Eletromagent 2 (button 2 temperature up)
			Default: 23	Off	Pressed by hand once(Off→Heating)	
			Default: 23	Heating		Auto press 6 times (23 + 0.5 * 6)
24.50				Heating		
24.54				Heating		
24.60				Heating		
24.73				Heating		
24.99				Heating		
25.23				Heating		
25.45				Heating		
25.67	↑			Heating		
25.88 > 25.80				Heating	Auto press twice(Heating→Cooling→Off)	
25.95		26.00 (**25.80-26.10**)		Off		
26.04			23.00 → 26.00	Off		
26.09				Off		
26.10				Off		
26.10				Off		
26.08				Off		
26.07				Off		
26.06	↓			Off		
26.04				Off		
26.03				Off		
Next execution period				Off	Auto press once(Off→Heating)	
				Heating		Auto press twice (26 + 0.5 * 2)
26.05				Heating		
26.16				Heating		
26.26				Heating		
26.39				Heating		
26.50				Heating		
26.63	↑			Heating		
26.71				Heating		
26.83 > 26.80				Heating	Auto press twice(Heating→Cooling→Off)	
26.96				Off		

Figure 28 Final test result data logger

This data logger records the ambient temperature change measured by SHT31 and output from the Arduino Mega 2560, the change of the desired and set temperature, the change of the working mode of the temperature condition system, and the actions of the different electromagnet in two big cycles (Every cycle has three execution periods). Figure 29 shows the legend of this data logger.

Heating	Heating mode
Cooling	Cooling mode
Off	Switch off
	Working mode change
	Set temperature change
↑	Temperature increasing
↓	Temperature decreasing
→	Keeping
AAA	Reasons cause the action of the electromagnet

Figure 29 Legends of the data logger

8.3.3 Data analysis and conclusions

Ambient temperature	Trend	Desired temperature
24.50		
24.54		
24.60		
24.73		
24.99		
25.23		
25.45		
25.67	↑	
25.88 > **25.80**		
25.95		26.00 (**25.80-26.10**)
26.04		
26.09		
26.10		
26.10		
26.08		
26.07		
26.06	↓	
26.04		
26.03		
Next execution period		

Ambient temperature	Trend	Desired temperature
26.05		
26.16		
26.26		
26.39		
26.50		
26.63	↑	
26.71		
26.83 > **26.80**		
26.96		
27.03		27.00 (**26.80-27.10**)
27.04		
26.99		
26.92		
26.85	↓	
26.81		
26.78 < **26.80**		
26.82 > **26.80**		
26.84	↑	
26.85		
26.83	↓	
Next execution period		

Ambient temperature	Trend	Desired temperature
26.76		
26.59		
26.44		
26.33		
26.20		
26.09		
26.01		
25.92		
25.83		
25.74	↓	25.00 (**24.80 - 25.10**)
25.64		
25.55		
25.48		
25.35		
25.30		
25.26		
25.20		
25.17		
25.15		
25.14		
Next execution period		

Figure 30 Temperatures recorded in data logger

Figure 30 shows the ambient temperature changes of three execution periods during the first working cycle. According to these data, it can be directly proved that the final product can control the temperature condition system to adjust the ambient temperature to the desired temperature. Although in the third period, the temperature cannot be completely decreased to the desired temperature, the ambient temperature is still very close to the desired temperature. This can be avoided by extending the execution time.

The red marks show the working mode changing at these moments. Some changes are caused by the completion of the temperature adjusting, and others are caused by the coming of the new execution period.

	24.73				Heating	
	24.99				Heating	
	25.23				Heating	
0	25.45				Heating	
1	25.67	↑			Heating	
2	25.88 > **25.80**				Heating	Auto press twice(Heating→Cooling→Off)
3	25.95		26.00 (**25.80-26.10**)		Off	
4	26.04			23.00 → 26.00	Off	
5	26.09				Off	
6	26.10				Off	
7	26.10				Off	
8	26.08				Off	

26.63	↑			Heating	
26.71				Heating	
26.83 > **26.80**				Heating	Auto press twice(Heating→Cooling→Off)
26.96				Off	
27.03				Off	
27.04		27.00 (**26.80-27.10**)		Off	
26.99			26.00 → 27.00	Off	
26.92				Off	
26.85	↓			Off	
26.81				Off	
26.78 < **26.80**				Off	Auto press once (Off→Heating)
26.82 > **26.80**				Heating	Auto press twice(Heating→Cooling→Off)
26.84	↑			Off	
26.85				Off	
26.83	↓			Off	
Next execution period				Off	Auto press twice(Off→Heating→Cooling)

Figure 31 Part data of the data logger

Figure 31 shows parts of the data logger which has the information of temperature condition system working mode changes and electromagnets action changes. According to table 9, the changes of the working mode shown in figure 31 are corresponding to the correct action changes of the electromagnets.

The blue circles in figure 31 show the process of working mode change from ‘Heating’ to ‘Off’ when the ambient temperature reaches the desired result. Especially when the temperature changes in a tolerable range. The working mode is still kept ‘Off’. This can prove the desired result of the test: When the ambient temperature reaches the present preset value. The temperature condition system will be switched off automatically.

The red circle in figure 31 shows the process of the ambient temperature fluctuations. The ambient temperature fluctuates around the desired value caused by the heat dissipation of the foam box and repeated start and stop of the temperature condition system. When the ambient temperature changes, the working mode changes accordingly. This can prove the desired result of the test: In the same execution period, after the temperature condition system is switched off, the ambient temperature changes and becomes unequal to the preset values, the temperature condition system will be switch on again to adjust the ambient temperature.

Temperature	24.50
Temperature	24.54
Temperature	24.60
Temperature	24.73
Temperature	24.99
Temperature	25.23
Temperature	25.45
Temperature	25.67
Temperature	25.88
Temperature	25.95
Temperature	26.04
Temperature	26.09
Temperature	26.10
Temperature	26.10
Temperature	26.08
Temperature	26.07
Temperature	26.06
Temperature	26.04
Temperature	26.03
Temperature	26.05
Temperature	26.16
Temperature	26.26

Figure 32 Part results of data stored by SD card

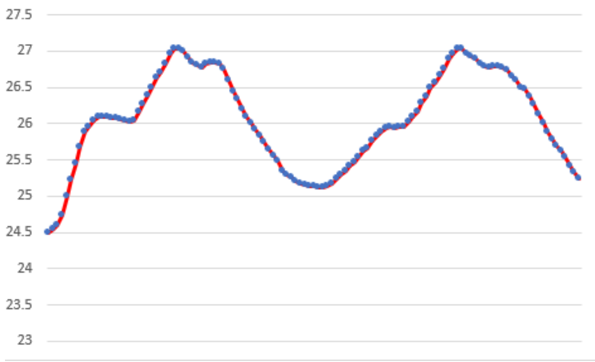


Figure 33 Part results of data stored by the online data logger

Figures 32 and 33 show the data stored in SD card and online data logger. After comparing these data with the data recorded in the data logger. All data are the same. This can prove the desired result of the test: The climate data of the container can be stored in both SD card and online data logger.

For more detailed test results, please check '**Appendix 13.4 Final test result data logger'.**

8.4 Conclusion of the final product test

After analyzing the result data logger, the final conclusion can be drawn: The final product can be used to control and monitoring the climate of a container with fixed volume and store all climate data at the same time. If the real air-conditioner, dehumidifier, and a well-built enclosed container are used with the final product, the temperature and moisture control results will be more accurate and stable.

9 Conclusions and recommendations

In this chapter, all the research questions will be answered with the conclusions drawn during completing the project. And also, the recommendations of the project are contained in this chapter.

9.1 Conclusions of the research questions

In this paragraph, all the research questions mentioned in chapter **'1.2 Problem statement'** will be answered.

9.1.1 Conclusions for the main question

Main question: **What design is feasible and affordable to realize the functions of temperature and moisture control, monitoring, and related data storage and wireless transmission in an enclosed 20ft container (L * W * H = 6.1 * 2.4 * 2.6 with a surface of 14.64 m² and a volume of 38.064 m³)**

The design with four controllers to take charge of different functions and using wireless data transmission to communicate is feasible and affordable. There is one main controller and three sub-controllers used in the final design. The main controller is used to collect climate data from sensors and judge the current situation according to the measured data and preset data to decide what command should be sent to the sub-controller. Three sub-controllers take charge of the function of online data storage and remote control, temperature adjusting devices control, and moisture adjusting devices control. Three sub-controllers will execute the commands from the main controller.

Using Multi controllers can guarantee every function can run efficiently and normally. Using wireless data transmission between controllers can guarantee the arrangements of devices and components are not limited by the length of the wire. All devices and components can be placed anywhere in the container.

All the choices and devices are made after comparing their cost performance. All devices and components can realize the best results with the lowest price.

9.1.2 Conclusions for the theoretical sub-questions

Sub-question 1: **What components can be used to control temperature?**

Air-conditioner; compressor with condensing pipe, evaporation pipe, reversing valve, capillary tubes, and refrigerating fluid; Peltier device.

Sub-question 2: **What components can be used to adjust moisture?**

The drying agent and dehumidifier can be used to decrease moisture. Atomizer and humidifier can be used to increase the moisture.

Sub-question 3: **In what way will the collected data be stored and uploaded to a data logger?**

The SD card storage module can be used to store data and the WIFI signal can be used to upload the data to the online data logger.

9.1.3 Conclusions for the empirical sub-questions

Sub-question 1: **What is a possible way to build communication between sensors, controllers, and the climate control components?**

Between sensors and controllers, the wire connection is enough. The I2C and SPI data transmission protocols are used to communicate between sensors and controllers.

Between controllers, wireless data transmission is used. NRF24L01 module can guarantee efficient and stable data transmissions between controllers.

Between controllers and climate control components. Some components are directly connected to the controller and controlled by electrical level signals from the controller. For the devices and components which can not be connected to the controller directly. The relay module and electromagnets are used. The controller will control the relay to control the electromagnets to press buttons on these devices.

Sub-question 2: **What is a possible way to control moisture at a constant value when the environment temperature always changes?**

Keep measuring the moisture when the temperature is changing. So that the moisture control devices and components can be switched on immediately to adjust the moisture.

Sub-question 3: **What is a possible way to measure the moisture contained in the bio-based material rather than only measuring the moisture of the surfaces of this material?**

In the present market, there are mainly two types of solid moisture sensors are sold, surface contact type and embedded type. The embedded type of solid moisture sensor has higher prices but more accurate compared with the surface contact type. It can measure the moisture contained in the bio-based material when the sensor is buried into the material and close contact with the material.

9.1.4 Conclusions for the analytical sub-question

Sub-question: **What is a possible way to use the temperature control components to adjust the temperature rapidly with low money and power cost?**

After information researching and cost performance comparing, using the air-conditioner is the best way to adjust temperature rapidly with low money and power cost for this project. Because the air-conditioner can be easily bought from the market and installed. Especially different air-conditioners with different power and prices can be found. Most air-conditioners comply with European energy usage standards. So these air-conditioners can save power. For using the compressor to adjust the temperature. Combining different components together needs a lot of time and the safety of the self-made compressor temperature adjusting system can not be guarantee. Especially there is no supplier sell small compressor in Europe. So buying compressors needs a lot of money. For using Peltier devices, the cooling efficiency of the Peltier device is too low to decrease the temperature by 10 ℃ in a room about 15 m^2. It will cost a lot of power.

9.2 Recommendations for the project

This paragraph will show the recommendations for the project. These recommendations are about the problem which cannot be solved during the project and the points which can be improved.

9.2.1 Practical operation and testing

Because the company cannot find a supplier who can supply the container which meets their requirements. So the real test in the container with air-conditioner and dehumidifier cannot be operated. Only the simple model which uses a foam box to replace the container and uses an adjustable voltage transformer and two Peltier devices to simulate an air-conditioner is used to test the final products. If it is possible to test in the real container. The test results can be more convincing.

9.2.2 How to expedite the simulation procedure

One client requirement is 'the system needs to adjust the temperature according to the preset serial of data automatically and do it as a circulation.' And the final product is designed according to the serial temperature data provided by the clients. The system can circulate to adjust the temperature according to these preset temperature values and preset time. But if it is possible to shorten the test circulation time? What effects will it have on the results of the characteristic research of the bio-based material wall? Are there results reliable? Although these questions are out of the boundaries of this project, they are worth being researched to improve the system.

9.2.3 The data logger and IoT platform used in the project

For this project, the Thingspeak online data logger and the TalkBack IoT application are used to realize online data storage and remote control. These two tools are professional enough for the data analysis. But the aspect of man-machine interaction is not enough. Two tools are both hard to use which have few guidance for new users. Especially users can use these two tools freely without money cost. So there might be problems with the unstable network connection when using the free version. Because of the shortage of time. These two tools are temporarily used.

But to solve the problem, the IoT platform Arduino Cloud is a better solution. This platform has more useful man-machine interaction interfaces. Which is easier for new users to use.

9.2.4 Future research directions

For this project, there are two directions can be developed in the future research. The first direction is about simulating and controlling more variables of the indoor climates such as the air composition of the container Room B. For example, controlling the concentration of carbon dioxides or methanal in the air and monitor the concentration change of these compositions.

The second direction is to increase the connections between Room A and Room B. Because in the project, two rooms are used to simulate different climates at the same time. And there is no mutual influence between two rooms. Increasing the connections between two rooms and take the biobased material wall as the medium to research what influence will happen to the indoor climate when the outdoor climate changes are important future research directions.

10 References

1. Agency, U. S. (2018, 9). Retrieved from Indoor Air Facts No. 8 Use and Care of Home Humidifiers. Retrieved from https://www.epa.gov/indoor-air-quality-iaq/indoor-air-facts-no-8-use-and-care-home-humidifiers

2. Biology online. (2021, 1). Moisture Definition. Retrieved from https://www.biologyonline.com/dictionary/moisture

3. Buijs, J. (2012, 5). The Delft Innovation Method. Retrieved from http://scholar.google.nl/scholar_url?url=https://www.designsociety.org/download-publication/38532/the_delft_innovation_method_a_design_thinkers_guide_to_innovation&hl=nl&sa=X&ei=T9WeYJnWLsPtmQH3_qPYDg&scisig=AAGBfm036gI5BHEIycOUKwU7ALDvVEqodQ&nossl=1&oi=s

4. cc. (2019, 2). The working principle and application of temperature and humidity sensor. Retrieved from http://www.elecfans.com/d/

5. CORROSION. (2020, 5). Heat Flux. Retrieved from https://www.corrosionpedia.com-/definition/1706/heat-flux

6. CUIDevices. (2014). PRODUCT SPOTLIGHT: PELTIER MODULES. Retrieved from https://www.cuidevices.com/product-spotlight/peltier-modules

7. Engineering360. (2012). Engineering 360. Retrieved from https://www.globalspec.com/learnmore/building_construction/hvac/cooling/room_air_conditioners

8. Fluxteq. (2021, 2). Fluxteq. Retrieved from https://www.fluxteq.com/

9. GRAINGER. (2019, 1). How Do Moisture Meters Work? Retrieved from https://www.grainger.com/know-how/equipment-information/kh-types-of-moisture-meters-qt-346

10. Humidifier, B. (2021, 1). Best Humidifier 2021 [Top Rated] Reviews And Buyer's Guide. Retrieved from https://besthumidifier2020.com/

11. Lingfeng. (2020, 7). What is the principle of a dehumidifier?What are the good brands? Retrieved from https://www.zhihu.com/question/19733283

12. Ltd, C. B. (2015). Thermocouple Accuracy Table | Microlink Measurement & Control. Retrieved from http://www.microlink.co.uk/tctable.html

13. machine, W. (2016, 9). Retrieved from Warm & Dry dehumidifiers heat recovery in dehumidifiers. Retrieved from https://web.archive.org/web/20160924023544/-https://warmdrypools.co.uk/Casestudies/CarbonGobblers

14. Rude, J. (2010). How air-conditioners are made. Retrieved from http://www.madehow.com/Volume-3/Air-Conditioner.html

15. Technik, W. (2021). Calorimetric chamber for materials testing. Retrieved from https://www.weiss-technik.com/en/products/produkte-detail/calorimetric-chamber-for-materials-testing

16. ThoughtCo. (2010, 2). Temperature Definition in Science. Retrieved from https://www.thoughtco.com/temperature-definition-in-science-2699014

17. xiaominge98. (2021, 1). Atomization piece. Retrieved from https://baike.baidu.com/item/%E9%9B%BE%E5%8C%96%E7%89%87/1629739?fr=aladdin

18. Zoeter, L. (2021). Background information of the BIO-ISO project.

11 Appendix 1 Final design hardware details

This chapter is about the introductions and descriptions of the components and devices used in the final product. The flow charts in the chapter are all basing on the paragraph **'7.1 Final system flow chart'**

11.1 Final component and device list

Table 11 shows the result of the "Make or buy" decision after comparing different components and component combinations. Table 11 only contains the components which will be used for code programming or device control. The power supply and circuit connection components are not listed.

Table 11 Final component and device list

No.	Concepts
Temperature and moisture control system (Buy)	
Temperature control	
1.	Adler AD 7916 mobile air conditioner (tentative)
Moisture control	
2.	ANGEEK Ceramic atomizer chip
3.	Seeedstudio Grove - Water atomization driver v1.0
4.	Lonove portable dehumidifier (tentative)
Temperature and moisture monitoring system (Buy)	
5.	Adafruit SHT-31 temperature and humidity sensor
6.	Keenso M6 type K thermocouple
7.	Seeedstudio Grove MCP9600 thermocouple driver
Data transmission and storage system (Buy)	
8.	NRF24L01 wireless data transmission module
9.	ESP32 WIFI module
10.	HW-125 SD memory card storage shield module
11.	TCA9548A I2C multiplexer
12.	Thingspeak online datalogger and TalkBack application

System logic control (Buy)	
13.	Arduino Mega 2560
14.	Arduino nano
15.	DS32S1 real-time clock
System device control (Buy)	
16.	JF-0826B 12V 20N electromagnet
17.	CH2 2 channel relay
Prototype and final products (Make)	
18.	Final test set up(foam box container with Peltier devices).
19.	Integration of sensors and components connected to Arduino Mega 2560 on one PCB board.
20.	Protection cases for all components

Tentative components mean these components or devices cannot be bought to test for proving the feasibility and make the decision of "Make of buy" during the concept phase because of the money cost but are temporarily decided to use in the final design. These components are normally the components or devices used for changing the temperature and moisture. The feasibility of these components or devices can only be roughly judged by the device information given by the seller to see if the function or power of these devices can meet the clients' requirements.

11.2 System logic control components

This paragraph will introduce the components used for system logic control.

11.2.1 Main controller: Arduino Mega 2560

i. Introduction

Arduino Mega 2560 (see figure 34) is the main controller for the whole indoor climate control system. It is used to receive data from the sensor, judge the climate situation (if the measured temperature and moisture are equal to preset values), communication with other Arduino nano controllers to send commands to them, or transmit data to dataloggers. And also, it is used to receive commands from the user and control the system to react to users' commands. Because it has more ports can be used and better processor performance compared with. So Arduino 2560 is used for the main controller.

Figure 34 Arduino Mega 2560

ii. Input and output flow chart

Figure 35 shows the Arduino Mega 2560 input and output flow chart. Table 12 shows Arduino Mega 2560 unique identifiers.

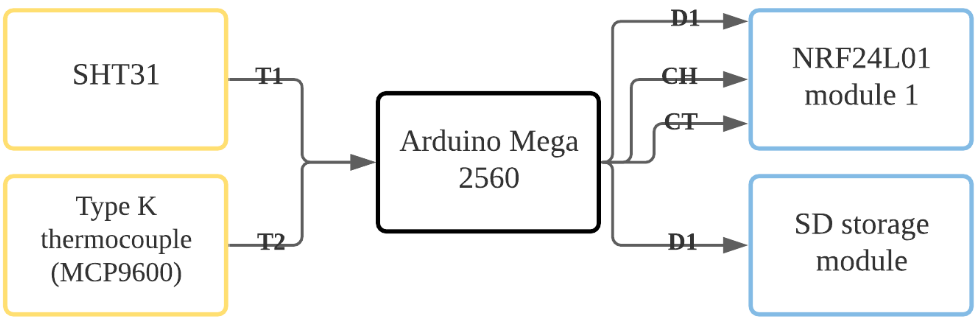


Figure 35 Arduino Mega 2560 input and output flow chart

Table 12 Arduino Mega 2560 unique identifiers

Identifier	Meaning
T1	Container Room B ambient temperature and moisture measured by SHT31
T2	Bio-base material wall temperature measure by a thermocouple (output from the MCP9600)
D1	All temperature and moisture data
CH	Commands to control the moisture control system
CT	Commands to control the temperature control system

iii. Pin usage

Table 13 shows Arduino Mega 2560 used pins.

Table 13 Arduino Mega 2560 used pins

pin	USage
2	Soft CS pin for SD storage module

3	Soft MOSI pin for SD storage module
4	Soft MISO pin for SD storage module
5	Soft SCK pin for SD storage module
7	CSN pin for NRF24L01 wireless module
8	CE pin for NRF24L01 wireless module
SDA	SDA pin for I2C multiplexer (SDA pin for MCP9600 and SHT31)
SCL	SCL pin for I2C multiplexer (SCL pin for MCP9600 and SHT31)
50	MISO pin for NRF24L01 wireless module
51	MOSI pin for NRF24L01 wireless module
52	SCK pin for NRF24L01 wireless module
GND	Power supply
5V	Power supply

iv. Tips

The power supply pin on the Arduino Mega 2560 cannot be directly used for the power supply of the NRF24L01 wireless module. 5V and GND pins must be connected to the breadboard to supply power.

11.2.2 Sub controller: Arduino nano

i. Introduction

Three Arduino nanos (see figure 36) are used as the controller to control the wireless module, temperature adjusting devices, and moisture adjusting devices. **Arduino nano 1** is used to receive temperature and moisture data from Arduino Mega 2560 with NRF24L01 and transmit data to the WIFI module. **Arduino nano 2** is used to receive commands from Arduino Mega 2560 to control the atomizer chips and dehumidifier. **Arduino nano 3** is used to receive commands from Arduino Mega 2560 to control the air-conditioner.

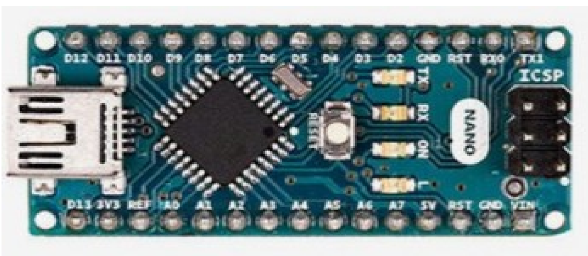


Figure 36 Arduino nano

ii. Input and output flow chart

Figure 37 shows the Arduino nanos input and output flow chart. Table 14 shows Arduino nanos unique identifiers.

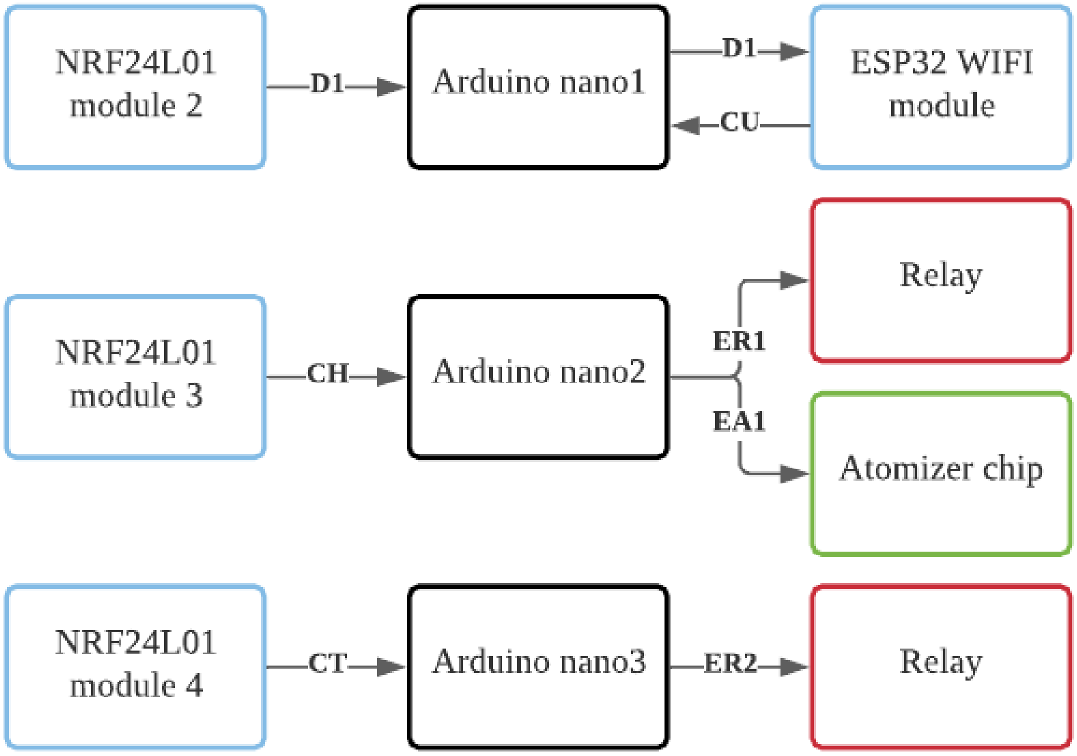


Figure 37 Arduino nano input and output flow chart

Table 14 Arduino nanos unique identifiers

Identifier	Meaning
D1	All temperature and moisture data
CH	Commands to control the moisture control system
CT	Commands to control the temperature control system
CU	Commands receive by WIFI module from users
ER1	The electrical level signal to control the relay connected to the electromagnets used to control the dehumidifier
EA1	The electrical level signal to control the atomizer chips (Seeedstudio Grove - Water atomization driver v1.0)
ER2	The electrical level signal to control the relay connected to the electromagnets used to control the air-conditioner

iii. Pin usage

Table 15 shows Arduino Mega nanos used pins.

Table 15 Arduino nanos used pins

pin	Usage
Arduino nano 1	
5	TX pin of soft serial data transmission to WIFI module
6	RX pin of soft serial data transmission to WIFI module
7	CSN pin for NRF24L01 wireless module 2
8	CE pin for NRF24L01 wireless module 2
11	MOSI pin for NRF24L01 wireless module 2
12	MISO pin for NRF24L01 wireless module 2
13	SCK pin for NRF24L01 wireless module 2
GND	Power supply
5V	Power supply
Arduino nano 2	
5	Output level signal to relay connected to the electromagnets used to control the dehumidifier (control button 'on/off')
6	To the EN pin on the atomizer chip driver to control the atomizer
7	CSN pin for NRF24L01 wireless module 3
8	CE pin for NRF24L01 wireless module 3
11	MOSI pin for NRF24L01 wireless module 2
12	MISO pin for NRF24L01 wireless module 2
13	SCK pin for NRF24L01 wireless module 2
GND	Power supply
5V	Power supply
Arduino nano 3	
3	Output level signal to relay connected to the electromagnets used to control the air-conditioner (control button 'on/off')
4	Output level signal to relay connected to the electromagnets used to control the air-conditioner (control button 'mode')
5	Output level signal to relay connected to the electromagnets used to control the air-conditioner (control button 'up')
6	Output level signal to relay connected to the electromagnets used to control the air-conditioner (control button 'down')

7	CSN pin for NRF24L01 wireless module 4
8	CE pin for NRF24L01 wireless module 4
11	MOSI pin for NRF24L01 wireless module 2
12	MISO pin for NRF24L01 wireless module 2
13	SCK pin for NRF24L01 wireless module 2
GND	Power supply
5V	Power supply

11.3 System device control components

This paragraph will introduce the components used for system devices control. In the final design, the air-conditioner and dehumidifier which are planned to use are the encapsulated devices sold on the market. This means the circuits of these devices cannot be changed and the devices cannot be controlled with simply code programming. So the relays and electromagnets are used. Arduino will control the relay to control the power supply of the electromagnets and using electromagnets to push buttons on the air-conditioner and dehumidifier to realize device control.

Figure 38 & 39 shows the connection between relays and electromagnets.

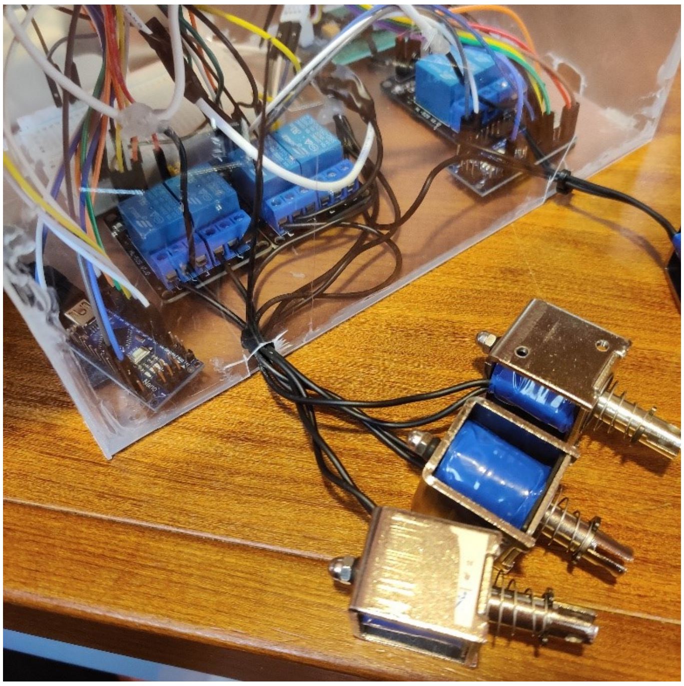


Figure 38 Connection between relays and electromagnets

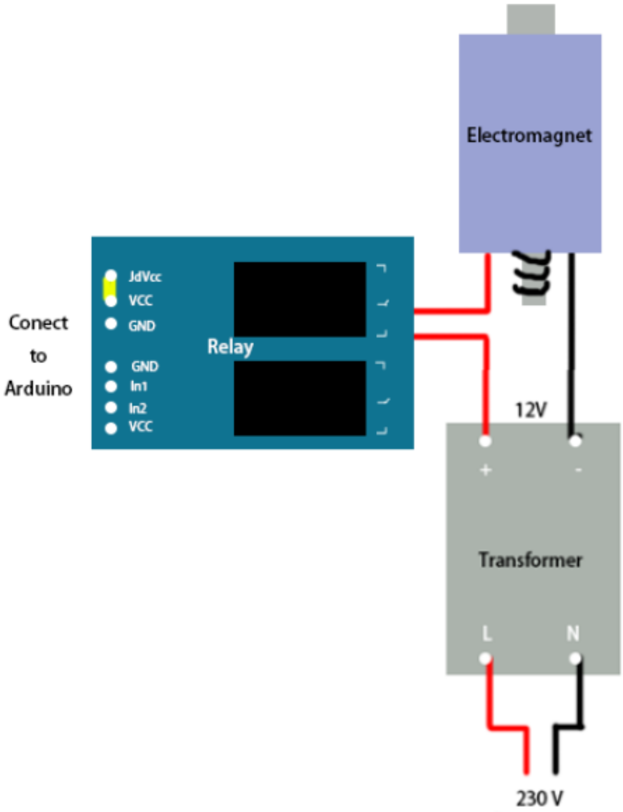


Figure 39 Circuit connection between electromagnets and relays

11.3.1 CH2 two-channel relay module

i. Introduction

The relay module (see figure 40) is used to control the power supply of the electromagnets. It uses low-level power from Arduino to control high power (12V). When the input pin is given a high-level signal, the relay closes, and the circuit connects and the circuit flows through the relay. The electromagnets will be triggered to press the buttons on the air-conditioner and dehumidifier.

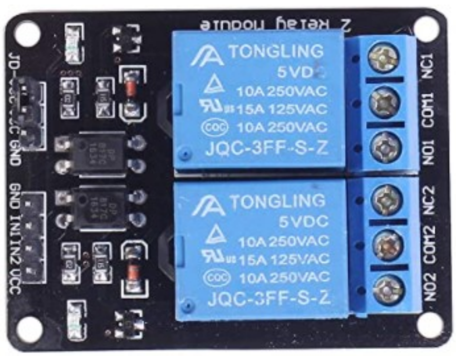


Figure 40 CH2 two-channel relay module

ii. Input and output flow chart

Figure 41 shows the relay module input and output flow chart. Table 16 shows relay module unique identifiers.

Figure 41 Relay module input and output flow chart

Table 16 Arduino nano unique identifiers

Identifier	Meaning
ER	The electrical level signal from Arduino nano

iii. Pin usage

Table 17 shows CH2 two-channel relay module used pins.

Table 17 CH2 two-channel relay module used pins

pin	Usage
IN1 / IN2	Pin to receive the electrical level signal from Arduino nano
VCC	Power supply
GND	Power supply
JDVCC	Connected with pin VCC

iv. Tips

When the relay is powered up and receives a high-level signal from Arduino, the LED on the module will be light up and users can hear a clear ‘click’ from the module.

11.3.2 JF-0826B 12V 20N electromagnet

i. Introduction

The electromagnets (see figure 42) are used to push the buttons on the air-conditioner and the dehumidifier to realize device control. When the electromagnet is powered, the metal stick at the inside of the electromagnets will stretch out. (See figure 43 & 44) And the thrust of the electromagnet is 20N.

Figure 42 Electromagnet

Figure 43 Electromagnet nonpowered

Figure 44 Electromagnet powered

ii. Tips

a) The electromagnet needs a power supply of 12VDC, and the max withstand current is 2A. So 230VAC power cannot be used. A 230VAC to 12VDC transformer is needed.

b) The electromagnet cannot be powered for more than 5 seconds once. Otherwise, the electromagnet will be overheated and burned.

c) The copper coil of the electromagnet is wrapped with insulating tape. Do not remove it and touch it directly.

11.4 Data transmission and storage system components

This paragraph will introduce the components used for data transmission and storage system.

11.4.1 NRF24L01 wireless data transmission module

i. Introduction

NRF24L01 wireless data transmission module (see figure 45) uses a 2.4GHz wireless signal to communicate between different modules. One NRF24L01 can be both used as a receiver and a sender. But at least two NRF modules are needed to realize data transmission. NRF24L01 modules are mainly used to transmit data from the main controller to Thingspeak online data logger and used to transmit commands from the main controller to sub-controllers.

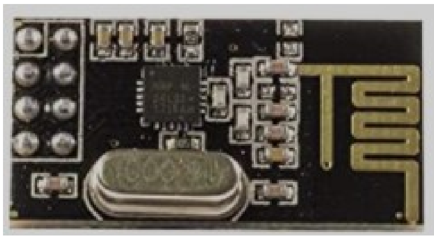


Figure 45 NRF24L01 wireless data transmission module

ii. Input and output flow chart

Figure 46 & 47 show the NRF24L01 input and output flow chart. Table 18 shows relay module unique identifiers.

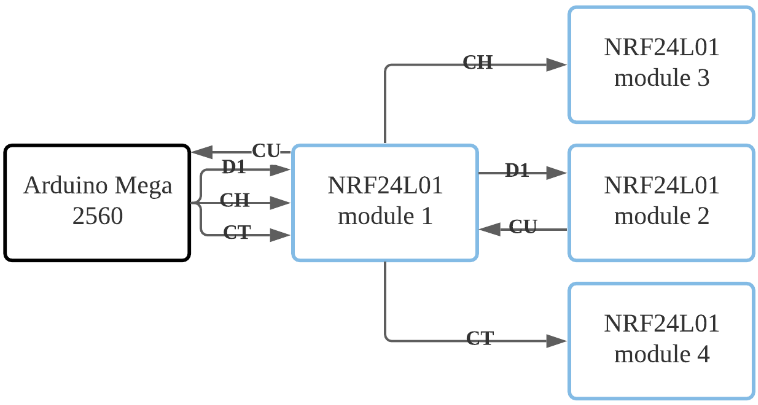


Figure 46 NRF24L01 input and output flow chart 1

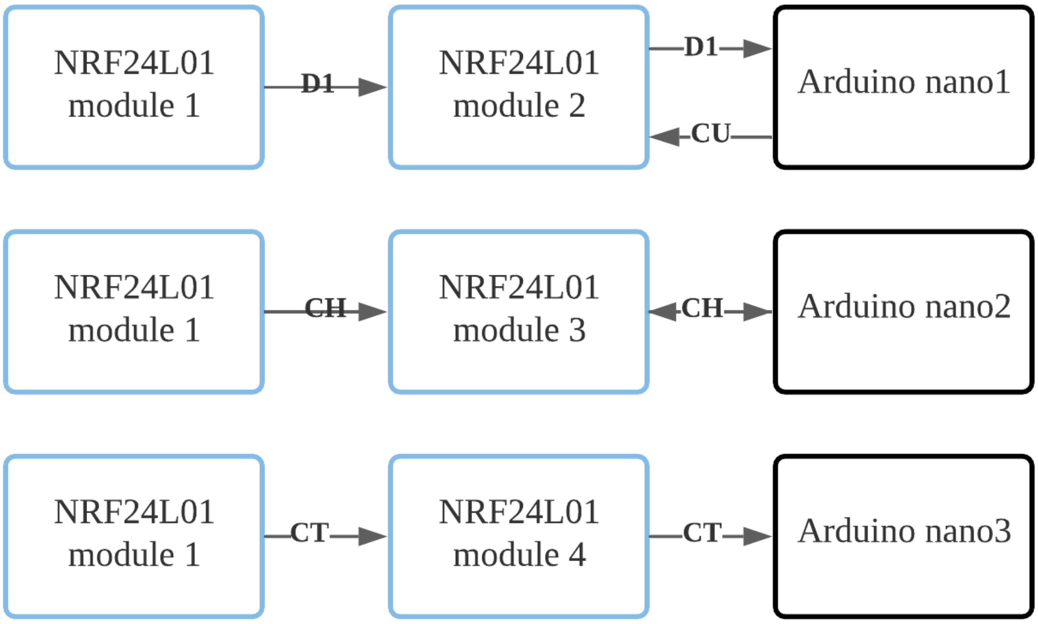


Figure 47 NRF24L01 input and output flow chart 2

Table 18 NRF24L01 unique identifiers

Identifier	Meaning
D1	All temperature and moisture data
CH	Commands to control the moisture control system
CT	Commands to control the temperature control system
CU	Commands receive by WIFI module from users

iii. Pin usage

Table 19 shows NRF24L01 used pins. Figure 48 show pins of NRF24L01.

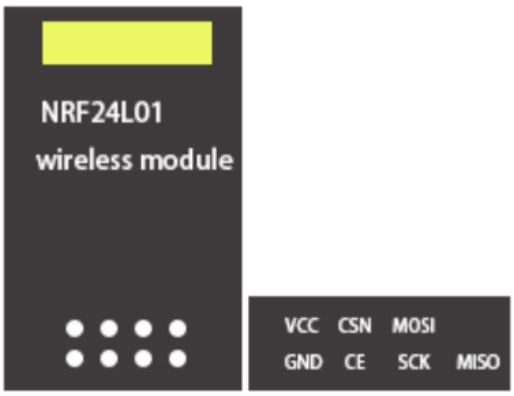


Figure 48 NRF24L01 pin sketch

Table 19 NRF24L01 used pins

pin	Usage
VCC	To 5V on Arduino / breadboard
GND	To GND on Arduino
CSN	To pin 7 on Arduino
CE	To pin 8 on Arduino
MOSI	To pin 51 on Arduino Mega 2560 / pin 11 on Arduino nano
MISO	To pin 50 on Arduino Mega 2560 / pin 12 on Arduino nano
SCK	To pin 52 on Arduino Mega 2560 / pin 13 on Arduino nano

iv. Tips

a) The NRF24L01 module cannot be directly powered by the 5V pin on the Arduino Mega 2560. It must be connected to the power pins on a breadboard.

b) In order to guarantee the receivers can receive data from the corresponding sender, the channel and receiver name set for the send should be the same as those set for the receiver.

c) The max data transmission distance of the NRF24L01 at the open area is around 50m. For longer distances, modules with antennas are needed. And a 10μ capacitor need to be soldered between the GND and VCC pin of the NRF24L01 when using the modules with antennas.

d) Don't keep data transmission in high frequency for a long time. Otherwise, the module will be overheated and burned.

11.4.2 ESP32 WIFI module

i. Introduction

ESP32 WIFI module (see figure 49) can be both used as a WIFI module and a development board. So it is connected to Arduino nano 1 to receive temperature and moisture data transmitted by NRF24L01 module from the main controller measured by SHT31 and thermocouple. After that, it uploads these data to Thingspeak online data logger through a WIFI signal. It can also receive user commands from the TalkBack application and transmit commands to the main controller.

Figure 49 ESP32 WIFI module

ii. Input and output flow chart

Figure 50 shows the ESP32 WIFI module input and output flow chart. Table 20 shows ESP32 WIFI module unique identifiers.

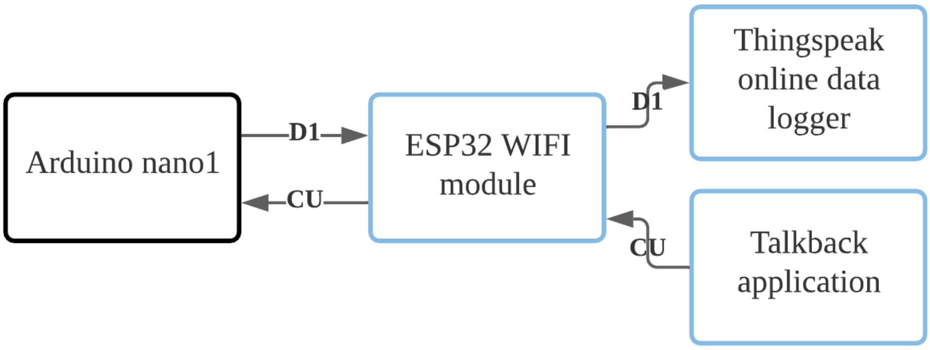


Figure 50 ESP32 WIFI module input and output flow chart

Table 20 ESP32 unique identifiers

Identifier	Meaning
D1	All temperature and moisture data
CU	Commands receive by WIFI module from users

iii. Pin usage

Table 21 shows ESP32 used pins.

Table 21 ESP32 used pins

Identifier	Meaning
16	RX pin for data transmission to Arduino nano 1 (connected to pin 5 on Arduino nano 1)
17	TX pin for data transmission to Arduino nano 1 (connected to pin 6 on Arduino nano 1)

iv. Tips

a) The ESP32 can be programmed in Arduino IDE but the library of the Arduino IDE needs to contain the library file of the ESP32.

b) The ESP32 can be only connected to the 2.4GHz WIFI rather than 5GHZ WIFI.

11.4.3 HW-125 SD memory card storage shield module

i. Introduction

HW-125 SD memory card storage shield module (see figure 51) is used to back up the temperature and moisture data to avoid data loss when uploading data to the Thingspeak data logger.

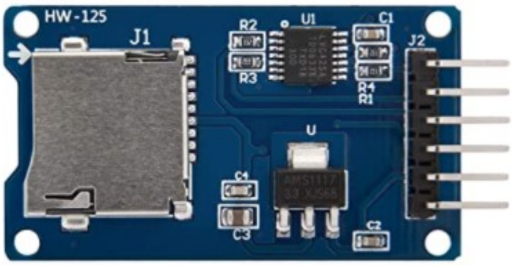


Figure 51 HW-12 SD memory card storage shield module

ii. Input and output flow chart

Figure 52 shows the HW-125 SD memory card storage shield module input and output flow chart. Table 22 HW-125 SD memory card storage shield module unique identifiers.

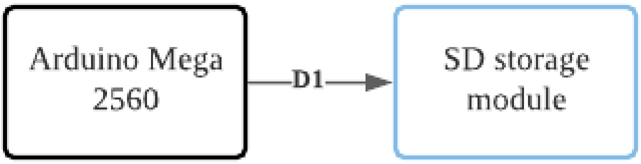


Figure 52 HW-125 SD memory card storage shield module input and output flow chart

Table 22 HW-125 unique identifiers

Identifier	Meaning
D1	All temperature and moisture data

iii. Pin usage

Table 23 shows HW-125 used pins.

Table 23 HW-125 used pins

pin	Usage
CS	To pin 2 on Arduino Mega 2560
MOSI	To pin 3 on Arduino Mega 2560
MISO	To pin 4 on Arduino Mega 2560
SCK	To pin 25 on Arduino Mega 2560
VCC	To pin 5V on Arduino Mega 2560
GND	To pin GND on Arduino Mega 2560

iv. Tips

Because of the conflictions between the library file of HW-125 and NRF24L01, the hard SPI data transmission cannot be used. So the soft SPI data transmission method is used.

11.4.4 TCA9548A I2C multiplexer

i. Introduction

TCA9548A I2C multiplexer (see figure 53) is used to collect data from the SHT31 temperature and humidity sensor and thermocouple (MCP9600) and send these data to the main controller. Because normally one Arduino only has one I2C port. But SHT31 and MCP9600 both use the I2C data transmission protocol. So TCA9548A is used to collect and combine data together to transmit to the controller.

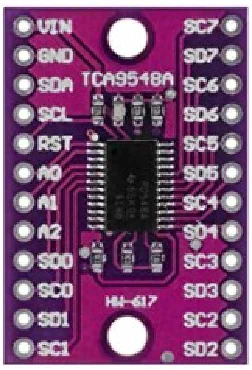


Figure 53 TCA9548A I2C multiplexer

ii. Input and output flow chart

Figure 54 shows the TCA9548A I2C multiplexer input and output flow chart. Table 24 TCA9548A I2C multiplexer unique identifiers.

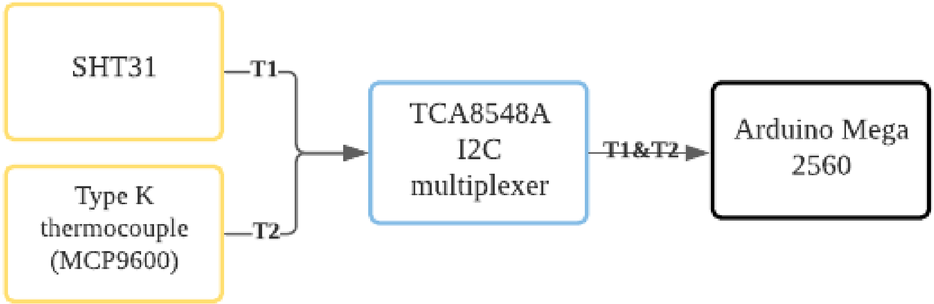


Figure 54 TCA9548A I2C multiplexer input and output flow chart

Table 24 TCA9548A unique identifiers

Identifier	Meaning
T1	Container Room B ambient temperature and moisture measured by SHT31
T2	Bio-base material wall temperature measure by the thermocouple (output from the MCP9600)

iii. Pin usage

Table 25 shows TCA9548A used pins.

Table 25 TCA9548A used pins

pin	Usage
VIN	To pin 5V on Arduino Mega 2560
GND	To pin GND on Arduino Mega 2560
SDA	To pin SDA on Arduino Mega 2560
SCL	To pin SCL on Arduino Mega 2560
SD0	To pin SDA on SHT31
SC0	To pin SCL on SHT31
SD1	To pin SDA on MCP9600
SC1	To pin SCL on MCP9600

11.4.5 Thingspeak online data logger & TalkBack application

i. Introduction

Thingspeak (see figure 55) is an online data logger which can be used to store temperature and moisture data and display these data on users' computer. With this data logger, users can see the temperature and moisture situation of the container Room B whenever and wherever they want.

TalkBack (see figure 56) is the application used to send commands with a WIFI signal to any WIFI modules.

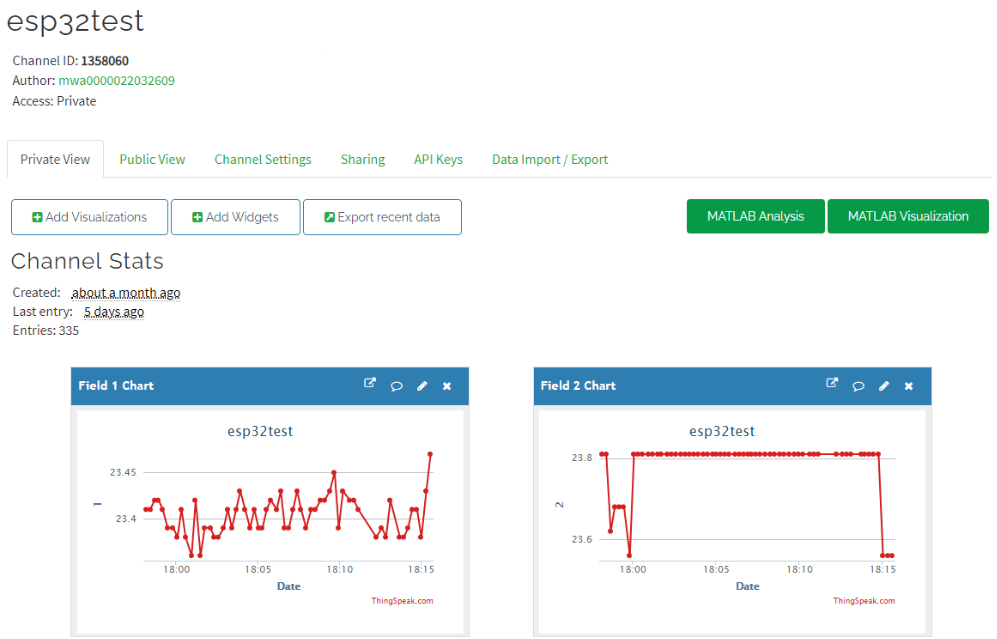


Figure 55 Thingspeak online data logger

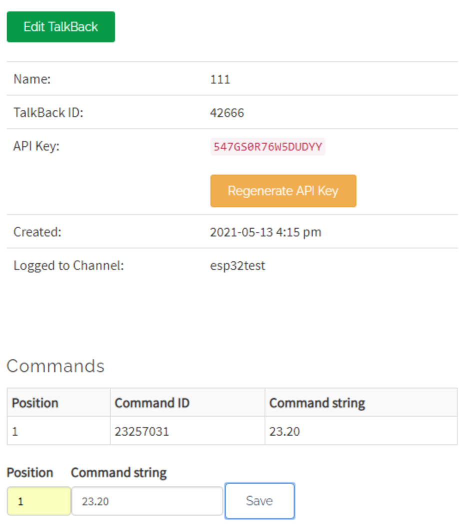


Figure 56 TalkBack application

ii. Input and output flow chart

Figure 57 shows the Thingspeak online data logger & TalkBack application input and output flow chart. Table 26 Thingspeak online data logger & Talk-Back application unique identifiers.

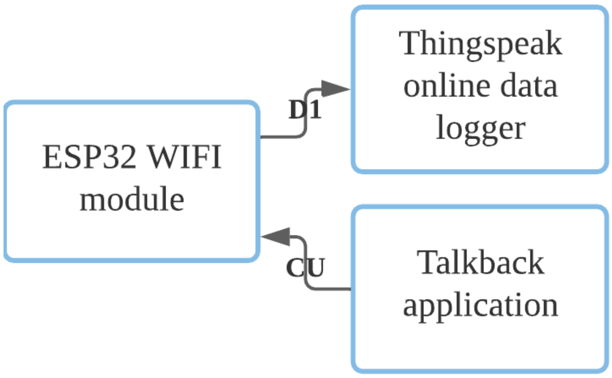


Figure 57 Thingspeak online data logger & TalkBack application Input and output flow chart

Table 26 Thingspeak and Talkback unique identifiers

Identifier	Meaning
D1	All temperature and moisture data
CU	Commands receive by WIFI module from users

iii. Tips

a) The free version Thingspeak online data logger collects the data from the ESP32 WIFI module every 15 seconds. So the data logger cannot receive every data output from sensors

b) The server of the TalkBack application is not very stable. Sometimes the ESP32 WIFI module cannot receive a response from it, so the command transmission delays might happen.

11.5 Temperature and moisture monitoring system components

This paragraph will introduce the components used for temperature and moisture monitoring system.

11.5.1 SHT-31 temperature and humidity sensor

i. Introduction

The SHT-31 module (see figure 58) is used to measure the ambient temperature and moisture of the container Room B. The temperature measurement accuracy is ±0.3℃And the moisture measurement accuracy is ±2%

Figure 58 SHT-31 temperature and humidity sensor

ii. Input and output flow chart

Figure 59 shows the SHT-31 temperature and humidity sensor input and output flow chart. Table 27 SHT-31 temperature and humidity sensor unique identifiers.

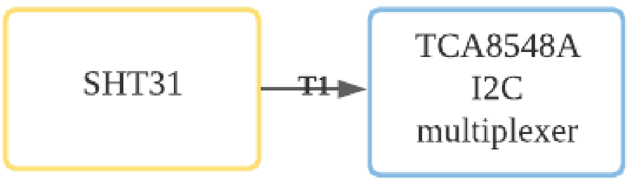


Figure 59 SHT-31 temperature and humidity sensor input and output flow chart

Table 27 SHT-31 unique identifiers

Identifier	Meaning
T1	Container Room B ambient temperature and moisture measured by SHT31

iii. Pin usage

Table 28 SHT-31 used pins

pin	Usage
SCL	To pin SC0 on TCA9548A
SDA	To pin SD0 on TCA9548A
VIN	To pin 5V on Arduino Mega 2560
GND	To pin GND on Arduino Mega 2560

11.5.2 M6 type K thermocouple & MCP9600 thermocouple driver

i. Introduction

The M6 type K thermocouple (see figure 60) is used to measure the temperature of the bio-based material wall between container Room A and Room B. MCP9600 is used to receive the analog signal from the thermocouple and transform it into a digital signal and send it to the main controller. The temperature measurement accuracy is ±0.5℃

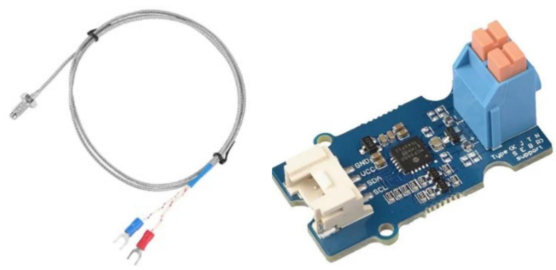


Figure 60 M6 type K thermocouple & MCP9600 thermocouple driver

ii. Input and output flow chart

Figure 61 shows Thermocouple and MCP9600 input and output flow chart. Table 29 Thermocouple and MCP9600 unique identifiers.

Figure 61 M6 type K thermocouple & MCP9600 thermocouple driver input and output flow chart

Table 29 Thermocouple and MCP9600 unique identifiers

Identifier	Meaning
T2A	Bio-base material wall temperature measure by the thermocouple (analog signal output from the M6 type K thermocouple)
T2	Bio-base material wall temperature measure by the thermocouple (digital signal output from the MCP9600)

iii. Pin usage

Table 30 shows MCP9600 used pins.

Table 30 MCP9600 used pins

pin	Usage
+	Connect to the blue line of the thermocouple
-	Connect to the red line of the thermocouple
VCC	To pin 5V on Arduino Mega 2560
GND	To pin GND on Arduino Mega 2560
SCL	To pin SC1 on TCA9548A
SDA	To pin SD1 on TCA9548A

11.6 Temperature and moisture control system components

This paragraph will introduce the components used for the temperature and moisture control system.

11.6.1 Water atomizer chip and driver

i. Introduction

The water atomizer chip and its driver (see figure 62) are used to increase the ambient moisture of the container Room B. It uses high-frequency vibration to make water mist to increase the ambient moisture. But the Arduino cannot supply high-frequency power. So the atomizer chip driver is used to produce the high-frequency signal.

Figure 62 Water atomizer chip and driver

ii. Input and output flow chart

Figure 63 shows Atomizer chip and its driver input and output flow chart. Table 31 Atomizer chip and its driver unique identifiers.

Figure 63 Atomizer chip and its driver input and output flow chart

Table 31 Atomizer chip and its driver unique identifiers

Identifier	Meaning
EA1	The electrical level signal to control the atomizer chip (Seeedstudio Grove - Water atomization driver v1.0)

iii. Pin usage

Table 32 shows Atomizer chip drive used pins.

Table 32 Atomizer chip driver used pins

pin	Usage
+	Connect to the red line of the atomizer chip
-	Connect to the blue line of the atomizer chip
EN	To pin 6 on Arduino nano
VCC	To pin 5V on Arduino nano
GND	To pin GND on Arduino nano

iv. Tips

a) There are two sides of the atomizer chip, one side should contact the water, the other side spray water mist, and this side cannot be submerged by the water. Otherwise, there is no water spray produced.

b) When producing the water spray, do not touch the atomizer chip driver. Because there is the high voltage on it and it's heated to a high temperature.

11.6.2 Air-conditioner and dehumidifier

i. Introduction

The air-conditioner and the dehumidifier (see figure 64) are devices used to adjust temperature and decrease moisture. They are also the devices that the system will finally control. But they will not be controlled directly with the Arduino controller by programming it. Electromagnets are used to push the buttons on their panels. In figure 48, it shows the panels of the air-conditioner which might be used in the final product. There are a total of four buttons that need to be controlled: ‘up’, ‘down’, ‘MODE’, ‘POWER’.

Figure 64 Air-conditioner and dehumidifier

Figure 65 The panel of the air-conditioner

The panel of the dehumidifier only has a button to control the on / off status of the dehumidifier. (See figure 65)

11.7 The circuit diagram of the final product

This paragraph shows the circuit connection diagram of four parts of the final product. (See figure 66-69)

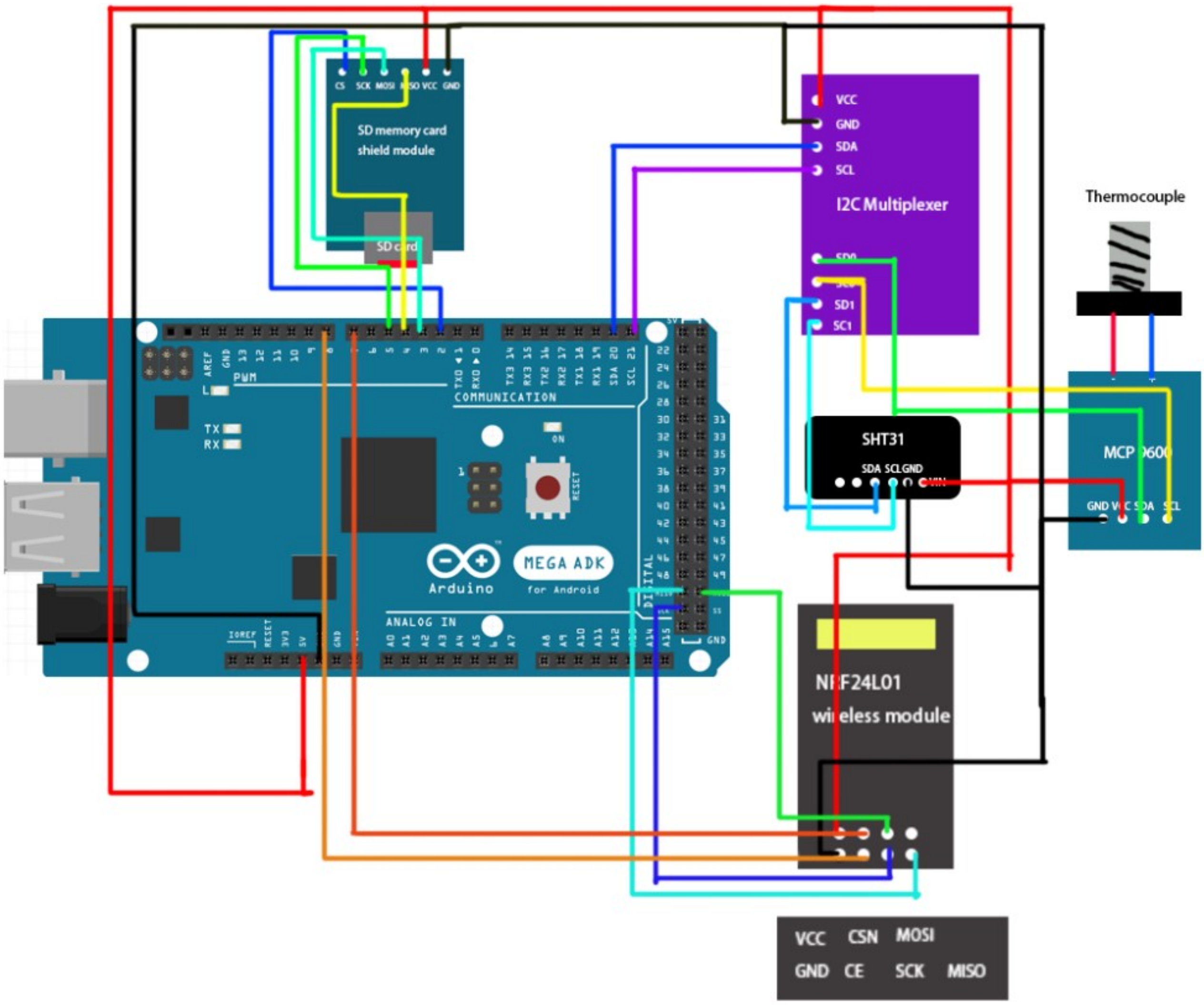


Figure 66 Part 1 circuit diagram

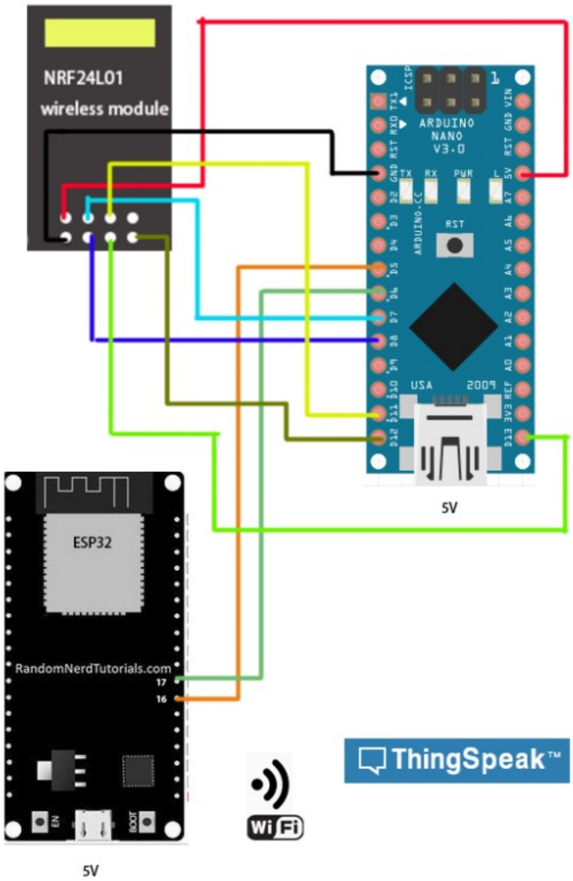


Figure 67 Part 2 circuit diagram

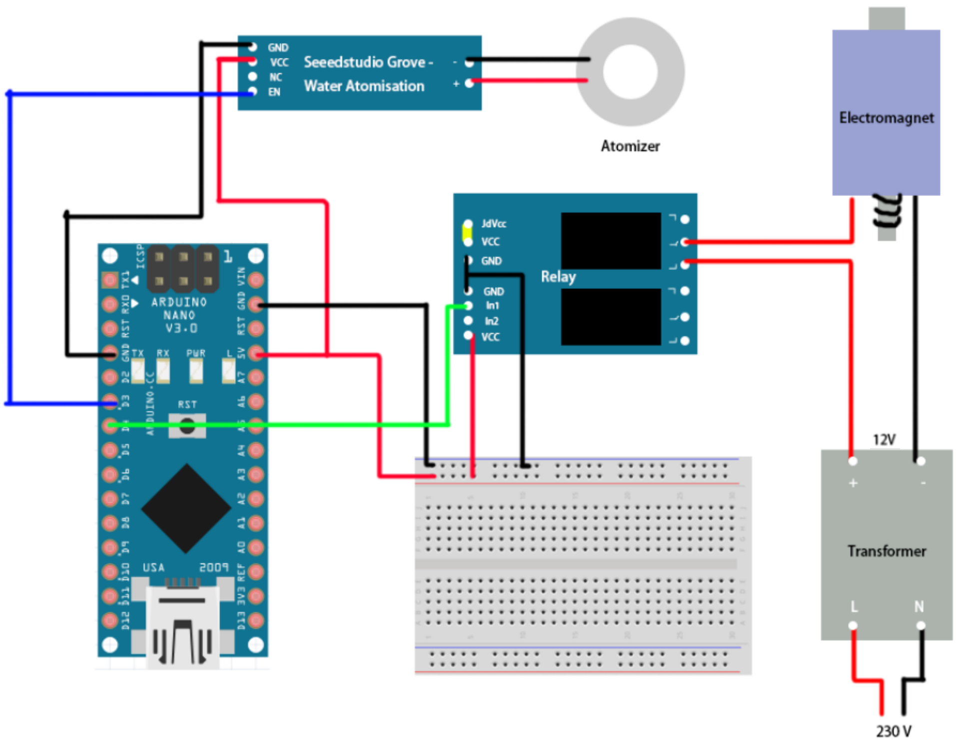


Figure 68 Part 3 circuit diagram

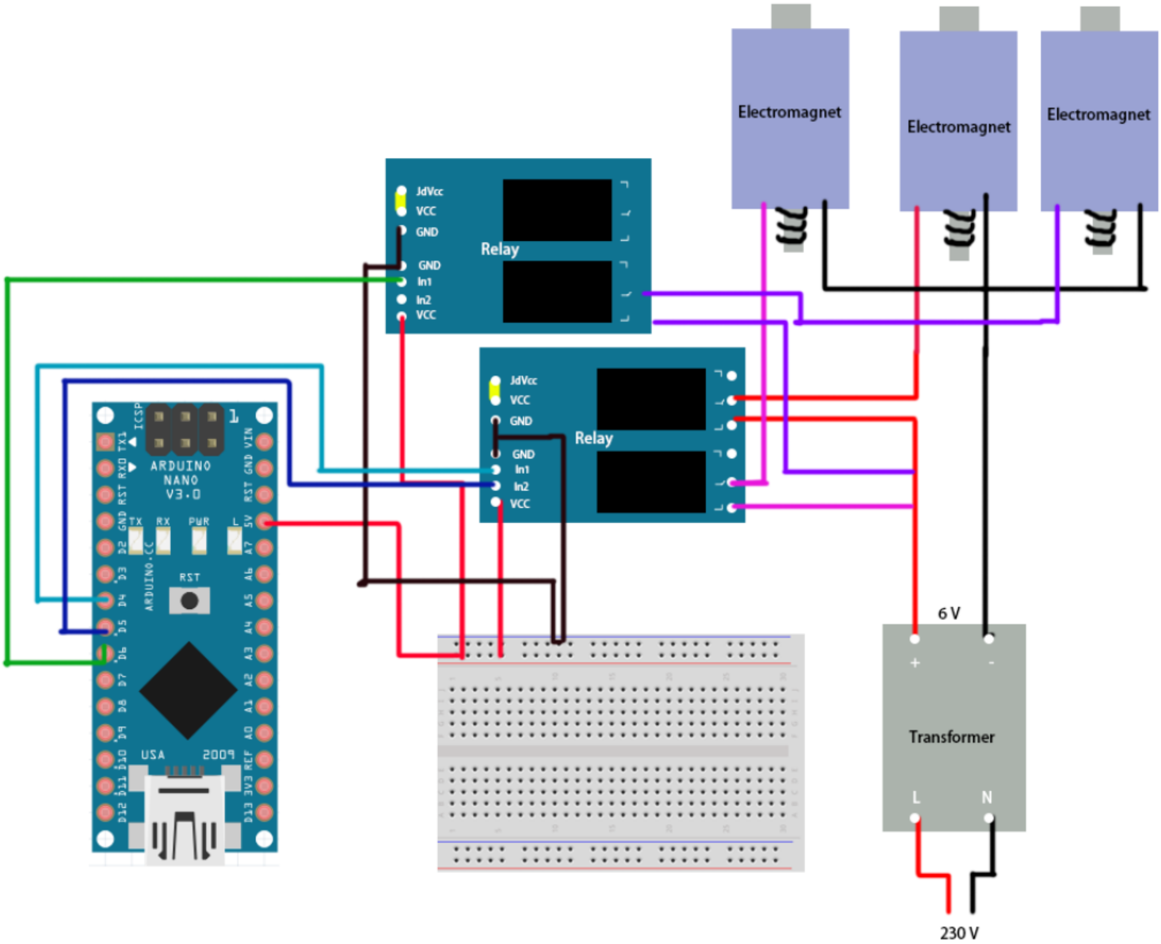


Figure 69 Part 4 circuit diagram

12 Appendix 2 Final design program and logic details

This chapter contains the analysis of programs used in the final product and detailed introduces the design logic of these programs.

12.1 Final program flow charts

This paragraph shows the logic of the program of the final product. There is a total of five flow charts that are related to the code of Arduino Mega, Arduino 1-3, and ESP32 WIFI module. (See figure 70-72)

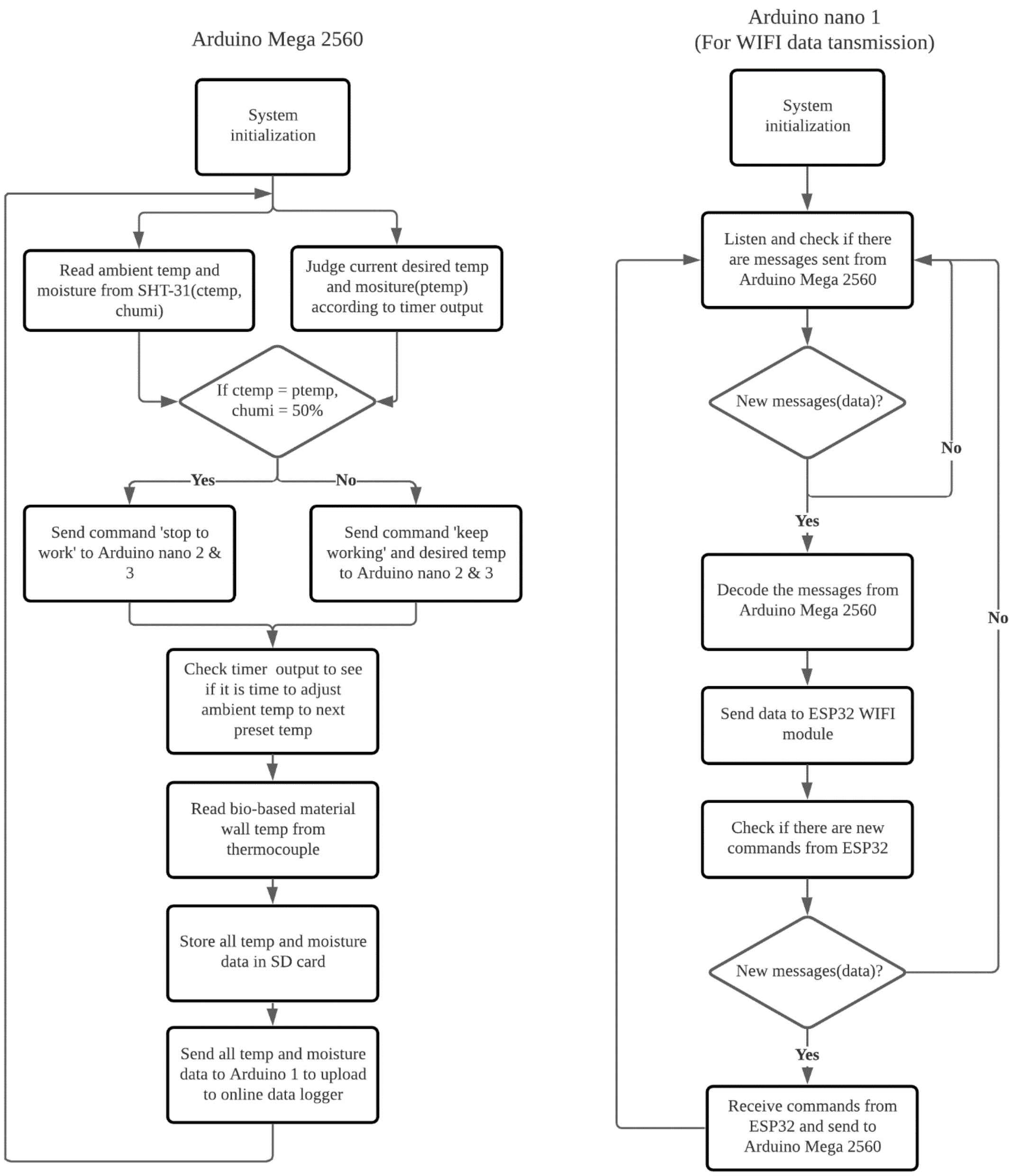


Figure 70 The flow chart of the programs of Arduino Mega 2560 & nano 1

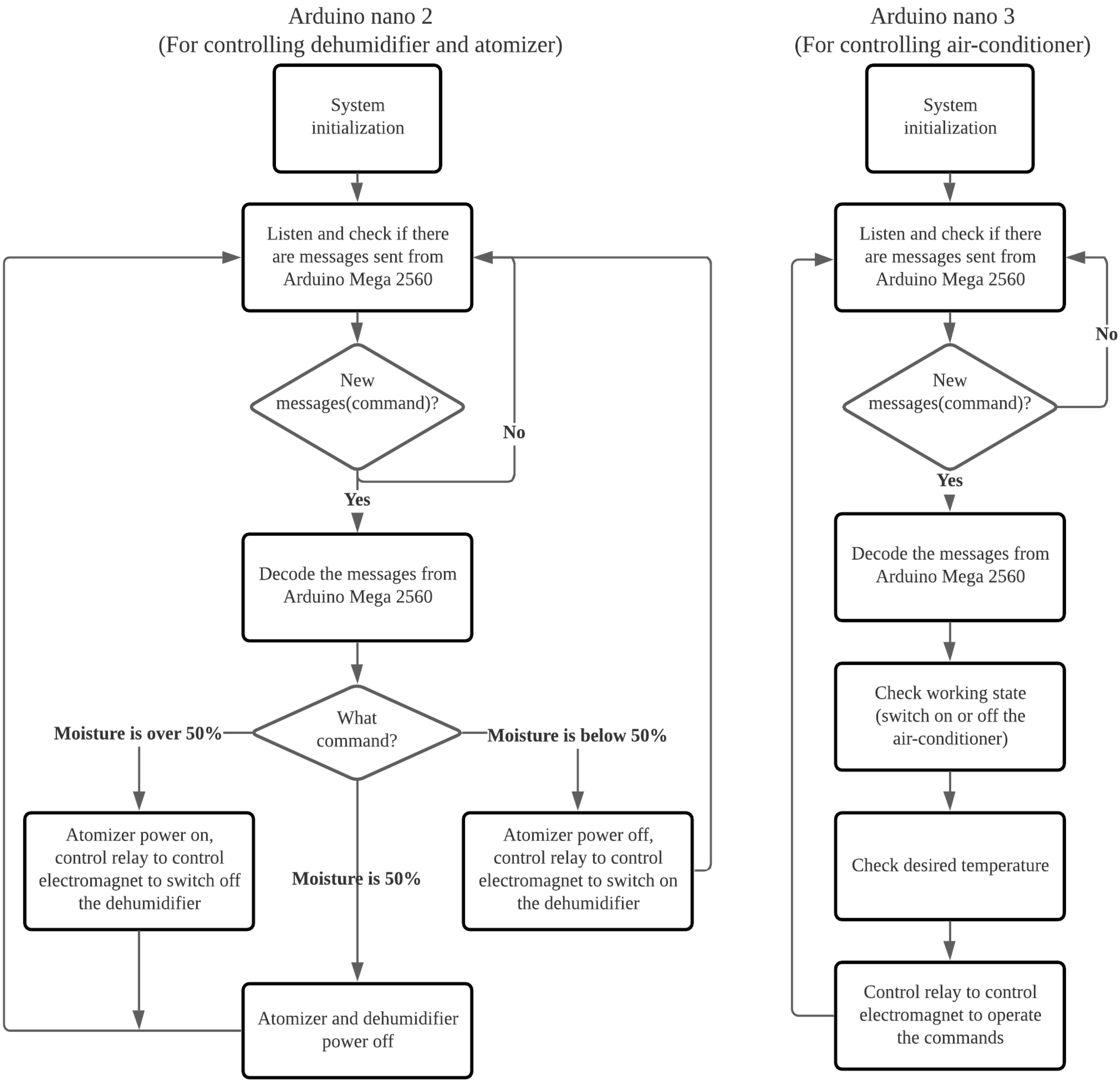


Figure 71 The flow chart of the programs of Arduino nano 2 & 3

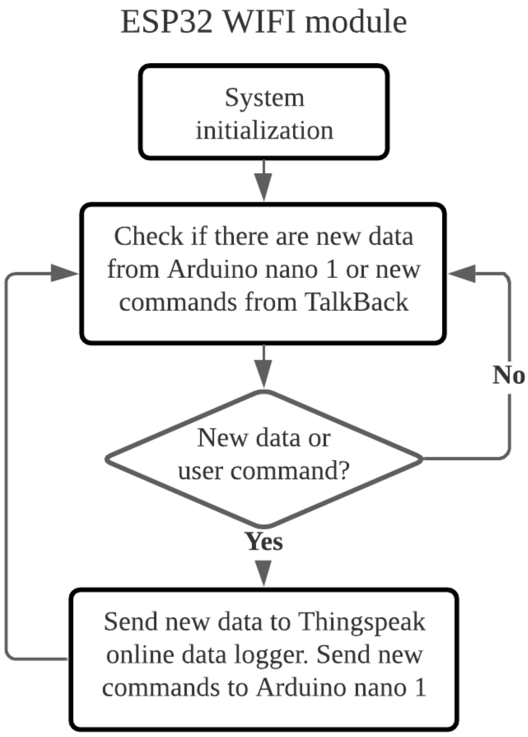


Figure 72 The flow chart of the programs of ESP32 WIFI module

12.2 Global variable list

This paragraph shows all the global variables used in the final program. The definitions of these variables are explained. (See table 33-37)

Table 33 Global variables used in the program of Arduino Mega 2560

Variables	Definition
TCAADDR	The address of the TCA9548A
SERIAL	The hardware serial port for the thermocouple
sensor	Structure-function declaration for MCP9600
sht31	Structure function declaration for SHT-31
sd	Structure-function declaration for Soft SPI for SD
file	Structure-function declaration for SD file read & write
SD_CONFIG	SD card initial function
SD_CS_PIN	CS pin number for Soft SPI for SD
SOFT_MOSI_PIN	MOSI pin number for Soft SPI for SD
SOFT_MISO_PIN	MISO pin number for Soft SPI for SD

SOFT_SCK_PIN	SCK pin number for Soft SPI for SD
ptemp[]	The array which stores the preset temperature
ctemp	The container Room B ambient temperature measured by SHT-31
chumi	The container Room B ambient moisture measured by SHT-31
wtemp	The bio-based wall temperature measured by the thermocouple
acswitch	The power status of the air-conditioner
moswitch	The working status of the atomizer and dehumidifier
command	The commands from the user
commandsign	Value to show if the command is executed
commandtemp	Show the temperature that the command represents

Table 34 Global variables used in the program of Arduino nano 1

Variables	Definition
value	Data received from Arduino Mega 2560 by NRF24L01
command	Commands received from TalkBack by ESP32
a (RX, TX)	Software serial data transmission port on Arduino nano 1

Table 35 Global variables used in the program of Arduino nano 2

Variables	Definition
dhumi	The array which stores the preset temperature
value	Data received from Arduino Mega 2560 by NRF24L01
dehumiswitch	Current working status of the atomizer and dehumidifer

Table 36 Global variables used in the program of Arduino nano 3

Variables	Definition
ptemp[]	The array which stores the preset temperature
value	Data received from Arduino Mega 2560 by NRF24L01
cstate	Current working status of the air-conditioner
corder	Current order of the preset temperature
lstate	Last working status of the air-conditioner
lorder	Last order of the preset temperature

stemp	Start set temperature of the program
mode	The working mode of the air-conditioner
lmode	Last working mode of the air-conditioner
commandsign	Value to show if the command is executed
commandtemp	Show the temperature that the command represents
repeatsign	Use to judge if the new command has been executed or not
judgesign	Use to judge if the new command is a new or old one

Table 37 Global variables used in the program of ESP32

Variables	Definition
RXD2	The RXD2 pin on ESP32 used for hardware serial data transmission with Arduino nano 1
TXD2	The TXD2 pin on ESP32 used for hardware serial data transmission with Arduino nano 1
ssid[]	The ssid of the WIFI ESP32 co
pass[]	The password of the WIFI connected by ESP32
client	Structure-function declaration for ESP32 WIFI
myTalkBackID[]	The ID of the TalkBack application
myTalkBackKey[]	The API key of the TalkBack application
thingspeak_key	The API key of the Thingspeak online data logger
ctemp	The container Room B ambient temperature meas-ured by SHT-31
chumi	The container Room B ambient moisture measured by SHT-31
wtemp	The bio-based wall temperature measured by the thermocouple
command	The commands from the user
commandsign	Value to show if the command is executed
commandtemp	Show the temperature that the command represents

12.3 Custom sub-function list

This paragraph shows all the custom sub-function used in the final program. The definitions of these functions are explained. (See table 38-40)

Table 38 Custom sub-functions used in the program of Arduino Mega 2560

Function name	Usage
tcaselect()	Port selection for TCA9548A
send2WIFI()	Send temperature and moisture data to Arduino nano 1 to upload to the data logger with ESP32
receivefWIFI()	Receive user commands from Arduino nano 1
SDstorage()	Write temperature and moisture data to SD card
wall_temp()	Collect bio-based material wall temperature from MCP9600
ambient_temp	Collect ambient temperature from MCP9600
thermocou-ple_config()	Initialize thermocouple
thermocou-ple_temp	Read analog signal from the thermocouple
acswitch_judge()	Use to judge the next working state of air-conditioner
moswitch_judge()	Use to judge the next working state of atomizer and dehumidifier

Table 39 Custom sub-functions used in the program of Arduino nano 2 & 3

Function name	Usage
judge()	Judge what button should be pressed according to the commands from Arduino Mega 2560
push_button()	Give relay modules electrical signal to control electromagnet to press the button

Table 40 Custom sub-functions used in the program of ESP32

Function name	Usage
httpPOST()	Receive commands from the TalkBack applica-tion. Check the connecting to the server of the TalkBack
upload()	Upload all temperature and moisture data to Thingspeak online data logger.
readdata()	Read data from Arduino nano 1
datadecode()	Decode the data received from Arduino nano 1 according to the custom data transmission proto-col.

12.4 Remarks

This paragraph shows about some points that need users to pay attention to when using the final codes.

12.4.1 The connections between different NRF24L01 modules

In figure 53 and figure 54, are codes for initializing and setting the NRF24L01 modules when this module is used for a receiver and the codes for modules used for a sender. With the function ‘Mirf.setRADDR((byte *)”XXXXR”)’ and ‘Mirf.setTADDR((byte *)”XXXXS”)’, XXXXR and XXXXS are the names of the modules. They are names with five chars and customed by users. A NRF24L01 module can be both used as a sender and a receiver. In order to guarantee the receiver can receive the right messages from corresponding senders. The name set for in the code in figure 73 and 74 must be the same.

```
Mirf.spi = &MirfHardwareSpi;
Mirf.init();
Mirf.setRADDR((byte *)"MAINC");
Mirf.payload = 2;
Mirf.channel = 11;
Mirf.config();
```

Figure 73 Codes for NRF24L01 initialization and setting as a receiver

```
Mirf.setTADDR((byte *)"WIFID");
Mirf.send((byte *)&ambient_temp);
while(Mirf.isSending()) delay(100);
```

Figure 74 Codes for NRF24L01 to send messages as a sender

The transmit data size and transmit channel can also be set by changing the value of ‘Mirf.payload’ and ‘Mirf.channel’. Also, the channels used by the sender and receiver must be the same.

12.4.2 The custom data transmission protocol for NRF24L01 modules 1 & 2

Normally, users do not need to care about the data transmission protocol when using the NRF modules. But if several groups of data need to be transmitted together, a custom data transmission protocol is needed.

For the data transmission between NRF module 1 & 2, it is mainly about the temperature and moisture data. There are three kinds of data, so the protocol is set as follow:

a) The temperature and moisture output by the SHT-31 and MCP9600 are numbers containing two integers and two decimals. So before data transmission, these data should increase a hundredfold. For example:

   24.44 → 2444， 30.19 → 3019.

b) After increasing the number, add headers for different types of data. For Room B ambient temperature, the header is 1. For Room B ambient moisture, the header is 2. For bio-based wall temperature, the header is 3 (see table 41).

c) So the final protocol is:

   1 1111

   1 bit header + 5 bits data contents

   For example: There is a string of data: 24698.

   ‘2’ means this string is about the ambient moisture of Room B. ‘4698’ means, the measured moisture is 46.98%

Table 41 Headers for data transmission protocal of NRF module 1 & 2

Header	Usage
1	This string is about Room B ambient temperature
2	This string is about Room B ambient moisture
3	This string is about bio-based material wall temperature

These kinds of data will be transmitted from Arduino Mega 2560 to Arduino nano 1 to ESP32. And in figure 75 is about the data decoding function according to this protocol in ESP32.

```
void datadecode(int thdata)
{
  if(10000 < thdata && thdata< 20000)
  {
    ctemp = (static_cast<float>(thdata) - 10000)/100;
    Serial.print("ctemp: ");Serial.println(ctemp);
  }
  if(20000 < thdata && thdata< 30000)
  {
    chumi = (static_cast<float>(thdata) - 20000)/100;
    Serial.println(chumi);
  }
  if(30000 < thdata && thdata< 40000)
  {
    wtemp = (static_cast<float>(thdata) - 30000)/100;
    Serial.print("wtemp: ");Serial.println(wtemp);
  }
}
```

Figure 75 Data decoding function used in ESP32

Table 42 shows the protocol used when the main controller controls the atomizer and dehumidifier to adjust moisture.

Table 42 Headers for data transmission protocol of NRF module 1 & 3

Header	Usage
0	Moisture is 50%. Atomizer and dehumidifier should be off
1	Moisture is lower than 50%. Atomizer should be on, and dehumidifier should be off
2	Moisture is lower than 50%. Atomizer should be off, and dehumidifier should be on

Table 43 shows the protocol used when the main controller controls the air-conditioner. When there is no new command. The data sent from main controller should start with two digits. The first digit means the working state of the air-conditioner. 1 means air-conditioner should work. 0 means air-conditioner should stop working. The second digit means the order of the desired preset value. For example, 0 means the temperature is being adjusted according to the first preset value. 1 means temperature is being adjusted according to the second preset value.

Table 43 Headers for data transmission protocol of NRF module 1 & 4

Protocol	Usage
1X	The air-conditioner should work when the order of temperature is X
0X	The air-conditioner should stop working when the order of temperature is X
20	New command is being executed
30	New command has been executed and the main controller is waiting for the next order of the desired temperature
1XXXX	The command temperature sent from NRF 2

12.4.3 The sub-function ‘httpPOST()’ for ESP32 & TalkBack application

The sub-function ‘httpPOST()’ is used to connect ESP32 to the server of the TalkBack application and receive commands from TalkBack. Figure 76 is the declaration of the function:

```
int httpPOST(String uri, String postMessage, String &response)
```

Figure 76 Declaration of the sub-function 'httpPOST'

‘uri’ and ‘postMessage’ contain the TalkBack ID and API-key. This information will be transmitted to the TalkBack server by ESP32. After the server recognizes the data source, it will transmit the command stored in its server given by the users to ESP32. And these commands will be stored in ‘&response’ when ESP32 receives these commands.

After executing this function, it will return a status value, the different value represents different problem (see table 42).

Table 44 Status value for ‘httpPOST()’ sub-function

value	Usage
-301	Fail to connect to TalkBack server
-303	Cannot parse response (do not find HTTP/1.1)
-304	Do not get server response in time
200	No problem

12.4.4 The sub-function ‘upload()’ for ESP32 & Thingspeak online data logger

The sub-function ‘upload’ is used to connect ESP32 to the server of Thing-speak online data logger and upload data to it. Figure 77 is the codes of the function:

```
String url = "/update?key=";
 url += thingspeak_key;
 url += "&field1=";
 url += String(ctemp);
 url += "&field2=";
 url += String(wtemp);
 client.print(String("GET ") + url + " HTTP/1.1\r\n" +
              "Host: api.thingspeak.com\r\n" +
              "Connection: close\r\n\r\n");
```

Figure 77 Codes of sub-function 'upload()'

In the code, the string ‘url’ will be transmitted to the Thingspeak server. This string is mainly combined with three parts: the API-key of the Thingspeak, the field which the users want to display the data, and raw data. Because users can create several fields in Thingspeak to display data. ‘&field1’ means display data in field one. So if users want to create more fields to show more data, they just need to add ‘&field3’ and so on in the code with other data in the form of ‘String(data)’. Figure 78 shows different fields in Thingspeak.

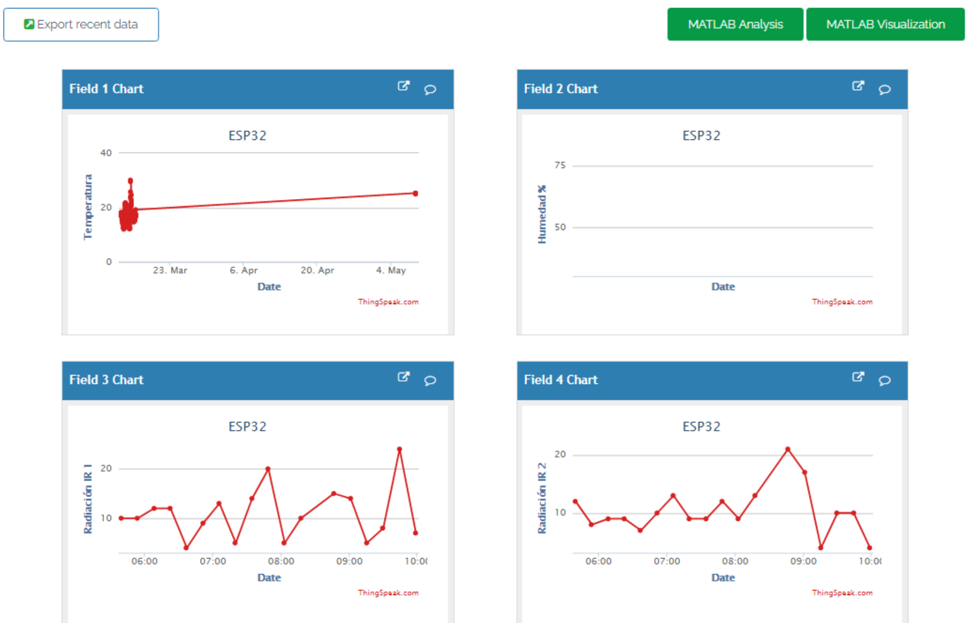


Figure 78 Different fields in Thingspeak to display different data

13 Appendix 3 Attachments and tests

13.1 Components cost performance list

Check out the attachment "**Components cost performance list**" for more details.

13.2 Final material list

Check out the attachment "**Final material list**" for more details.

13.3 Final codes

Check out the attachment "**Final codes**" for more details.

13.4 Final test result data logger

Check out the attachment "**Final test result data logger**" for more details.

13.5 Final test plan

1. Test aim

This test aims to test if the final product can realize the function of temperature monitor, control and related data recording in an enclosed container according to the preset temperature values and execution time.

2. Desired result.

a) The final product can adjust the container temperature according to the preset values and execution time.

b) The climate data of the container can be stored in both the SD card and the online data logger.

c) When the ambient temperature reaches the present preset value. The temperature condition system will be switched off automatically.

d) In the same execution period, after the temperature condition system is switched off, the ambient temperature changes and becomes unequal to the preset values, the temperature condition system will be switch on again to adjust the ambient temperature.

3. Test equipment

Table 45 Final test equipment

Final test set up
Laptop with Arduino IDE

4. Test variables

Table 46 Final test variables

Variables	Statue
Input voltage	5V
Ambient temperature	24℃-28℃
Working mode	Off / Heating / Cooling
State of electromagnet (button)	Pressed / Not pressed

5. Test circuit and codes

Test circuit and codes are almost the same as the final product circuit and final codes.

6. Test steps

a) Connect all the devices to the laptop to power them up.

b) Set the Arduino Mega and Arduino nano 3 with preset temperatures 26℃, 27℃, 25℃, and the execution time of 5 min.

c) Write down the ideal working process of the final product when it adjusts the ambient temperature from 26℃ to 27℃ to 25℃

d) After uploading the code to the Arduinos, open the port monitor of Arduino Mega 2560, Arduino nano 3, Arduino nano test.

e) Reset Arduino Mega 2560 and Arduino nano 3, and press button 1 (Mode control) to start the test.

f) Check the output from the monitor Arduino Mega 2560 to see the ambient temperature of the container. Check the output from the monitor

Arduino nano 3 to see the working state of the final product. Check the output from the monitor Arduino nano test to see the change of the working mode and set temperature of the temperature condition system.

g) When the ambient temperature reaches the desired temperature, check the output from the monitor Arduino nano 3 and the state of the electromagnets.

h) When the ambient temperature becomes unequal to the desired temperature, check the output from the monitor Arduino nano 3 and the state of the electromagnets.

i) When the new execution period comes, check the output from the monitor Arduino nano 3 and the state of the electromagnets.

j) Record all the temperature outputs from Arduino Mega 2560 and actions of different electromagnets to derive the whole working process according to the table 40 ‘button functions’ in paragraph ’10.3 The test of the final product’.

k) Compare the output temperature with desired temperatures. Compare the derivation working process with the ideal working process.

l) Check the data stored in SD card and online data logger.

m) Draw the conclusions.

7. Result assessment

After comparing the real working process with the ideal working process, only when the results meeting the following three requirements, the test can be considered to pass:

a) The final product can adjust the ambient temperature of the foam box to the desired value within an allowable error range and keep this value in the same execution period.

b) The working process (actions of electromagnets) of the final product conforms to the ideal working process. (Electromagnets do the correct actions at different periods)

c) The data recorded in SD card and online data logger should be equal to the data shown with monitors in Arduino IDE.

13.6 Component tests

This chapter contains all the component tests and results during the concept phase. With these results, the components are proved can be used in the final design.

13.1.1 The test of the atomizer and Seeedstudio Grove - Water Atomization v1.0 driving chip

1. Test aim

This test aims to test if the atomizer can work and how does it work.

2. Desired result.

The atomizer can be controlled by an Arduino and atomize liquid water into the water spray. When the atomizer is put on the surface of the water, the atomizer can work.

3. Test equipment

Table 47 Atomizer test equipment

Arduino nano
Seeedstudio Grove - Water Atomization
Atomizer
Breadboard
Jump wires

4. Test variables

Table 48 Atomizer test variables

Variables	Statue
Input voltage	5V

5. Test circuit

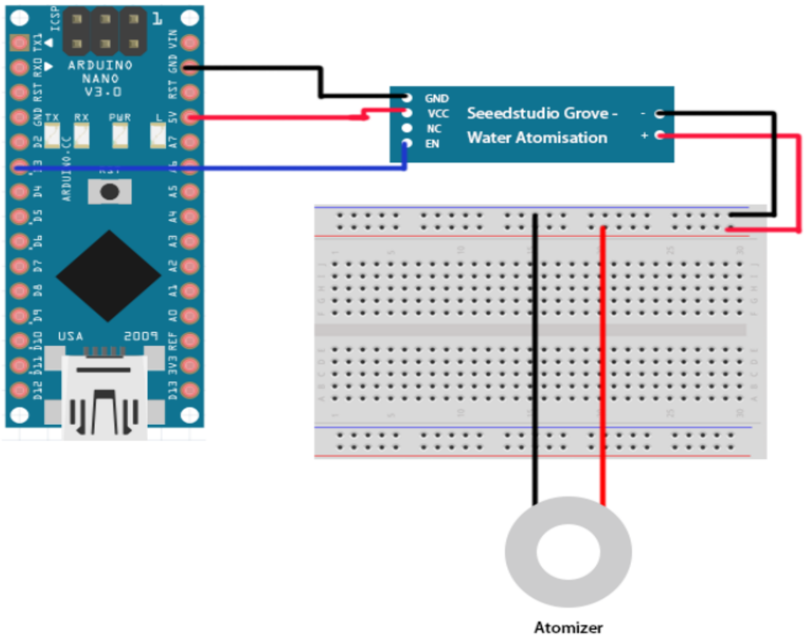


Figure 79 Atomizer test circuit

6. Test codes

```
int atomizationPin = 3;
void setup()
{
  pinMode(atomizationPin, OUTPUT);
}

void loop()
{
 while(1)
 {
    digitalWrite(atomizationPin, HIGH);
    delay(10000);
    digitalWrite(atomizationPin, LOW);
 }
}
```

Figure 80 Atomizer test cods

7. Test steps

a) Connect the equipment like the above test circuit.

b) Program the Arduino nano with the above codes and Arduino IDE. In these codes, Arduino will power up the atomizer for ten seconds. After that, Arduino will stop supplying power and wait for another ten seconds.

c) Put the atomizer on the surface of the water. The convex of the atomizer should be contacted with water.

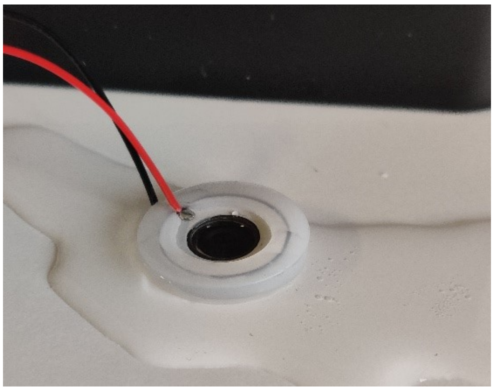


Figure 81 Atomizer test step

d) Observe the working status of the atomizer.

8. Test results

When the atomizer is powered up and placed on the surface of the water, there is plenty of water spray sprinkling from above the atomizer. When the Arduino stops supplying power, there is no water spray.

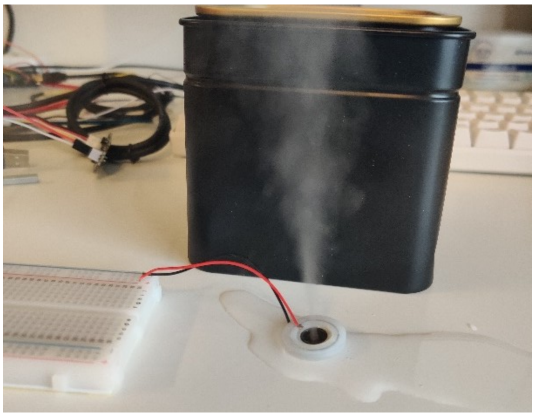


Figure 82 Atomizer test result

9. Test conclusion

The atomizer can be controlled by the Arduino to atomize liquid water into the water spray. When there is a distance between the atomizer and the surface of the water, or the atomizer is immersed into the water, no water spray will be generated. The atomizer can only work when there is water contact to its downside.

13.1.2 The test of the relay module

1. Test aim

This test aims to test if the relay module can work under the control of the Arduino.

2. Desired result.

The relay module can control the power supply of the appliances connected to the output of the relay with the commands from the Arduino.

3. Test equipment

Table 49 Relay module test equipment

Arduino nano
Relay module
LED
Breadboard
Jump wires

4. Test variables

Table 50 Relay module test variables

Variables	Status
Input voltage	5V
Working state of the LED	ON/OFF

5. Test circuit

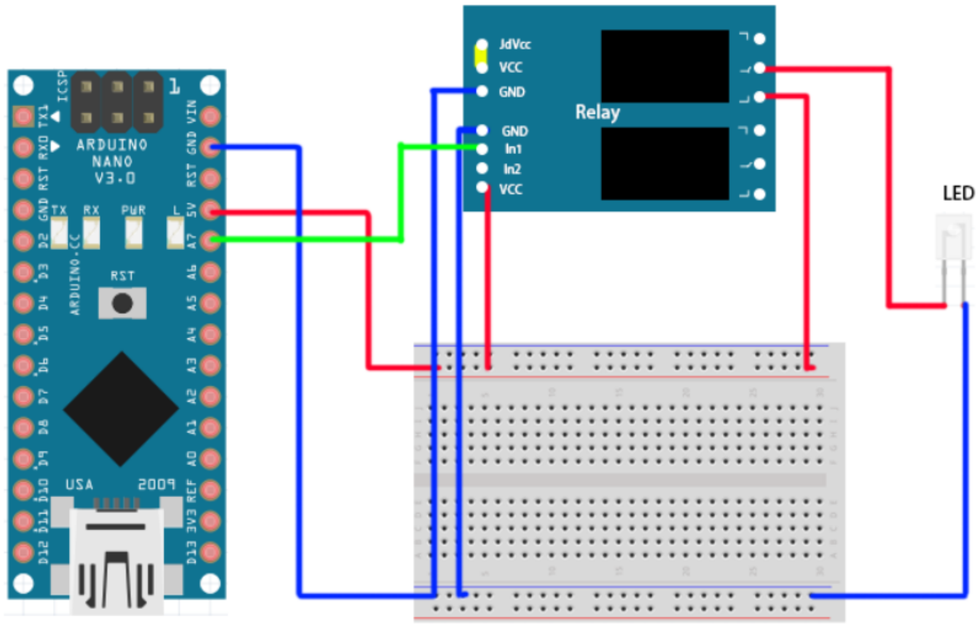


Figure 83 Relay module test circuit

6. Test codes

```
void setup()
{
pinMode(7, OUTPUT);
digitalWrite(7, HIGH);
}

void loop()
{
  digitalWrite(7, LOW);
  delay(3000);
  digitalWrite(7, HIGH);
  delay(3000);
}
```

Figure 84 Relay module test code

7. Test steps

a) Connect the equipment like the above test circuit.

b) Program the Arduino nano with the above codes and Arduino IDE. In these codes, Arduino will power up the relay for three seconds. After that, Arduino will stop supplying power and wait for another three seconds.

c) Observe the working status of the relay and LED.

8. Test results

When the relay is powered up, there is the clear sound 'CLICK', the red light on the relay, and the LED are lighted. After three seconds, the Arduino stop to supply power, the red light on the relay, and the LED extinguish.

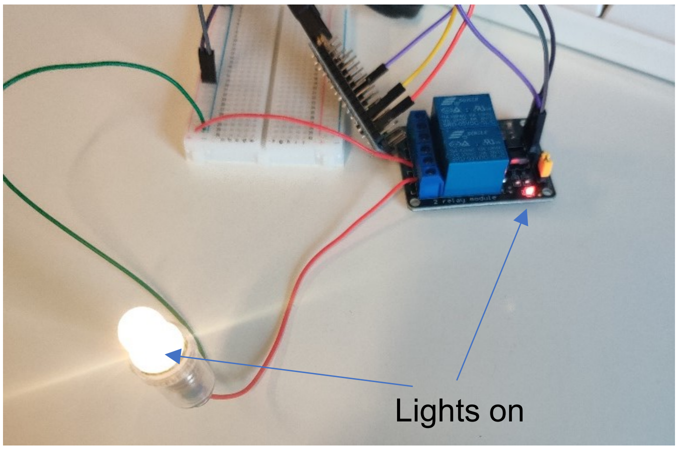


Figure 85 Relay module test result 1

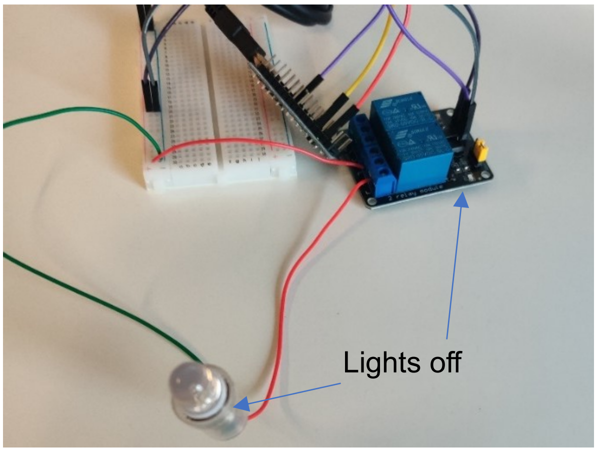


Figure 86 Relay module test result 2

9. Test conclusion

The relay module can control the power supply of the appliances connected to the output of the relay according to the commands from the Arduino.

13.1.3 The test of the electromagnet

1. Test aim

The first test aim is to prove the electromagnet can work under the control of the Arduino with the power supply of a 230V-12V voltage transformer. The second test aim is proving the electromagnet can be used to push buttons.

2. Desired result.

The electromagnet can work according to the status of the relay. When the electromagnet is powered up, it has enough force to push the button.

3. Test equipment

Table 51 Electromagnet test equipment

Arduino nano
Relay module
Electromagnet
230V-12V voltage transformer
Power bank
Breadboard
Jump wires

4. Test variables

Table 52 Electromagnet test variables

Variables	Status
Input voltage of the relay	5V
Input voltage of the voltage transformer	230V
Input voltage of the electromag-net	12V
Light on the power bank	ON/OFF

5. Test circuit

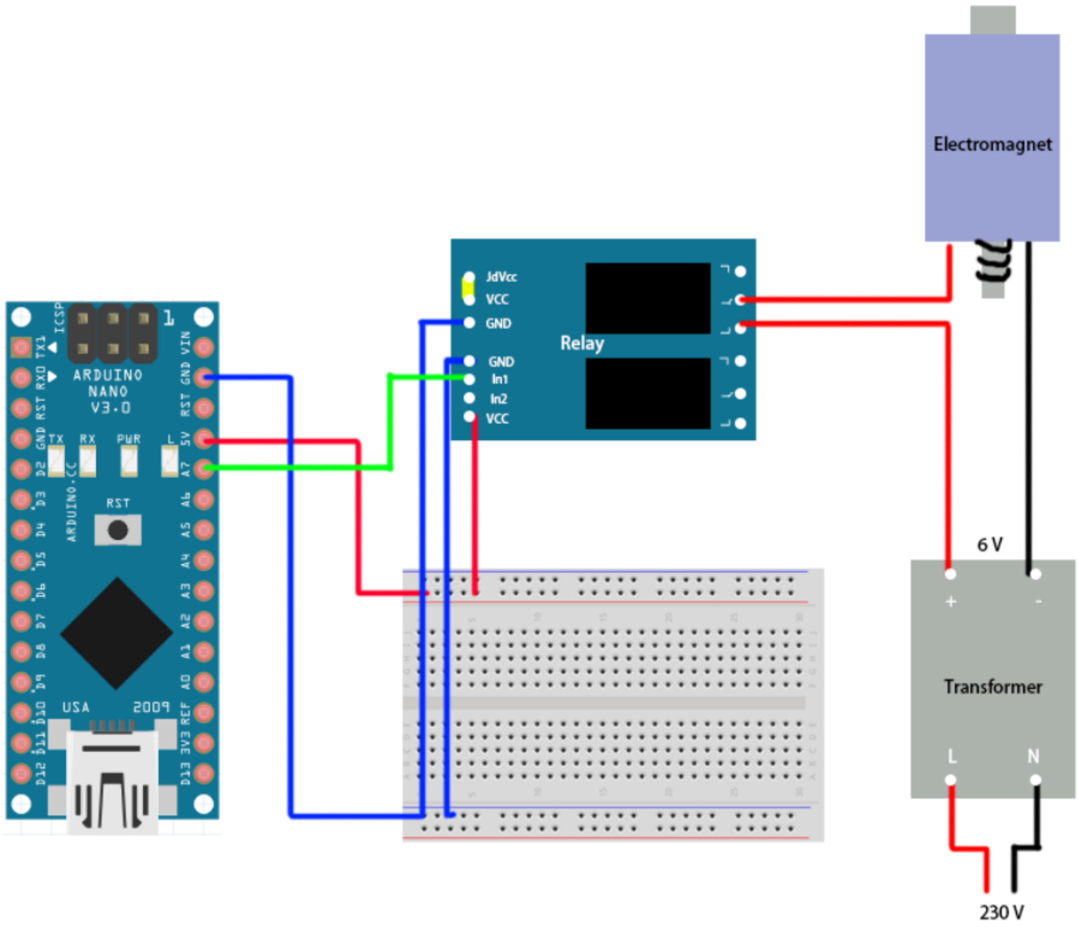


Figure 87 Electromagnet test circuit

6. Test codes

```
void setup()
{
pinMode(7, OUTPUT);
digitalWrite(7, HIGH);
}

void loop()
{
  digitalWrite(7, LOW);
  delay(3000);
  digitalWrite(7, HIGH);
  delay(3000);
}
```

Figure 88 Electromagnet test codes

7. Test steps

a) Connect the equipment like the above test circuit when all the power is cut off.

b) Program the Arduino nano with the above codes and Arduino IDE. In these codes, Arduino will power up the relay for three seconds. After that, Arduino will stop supplying power and wait for another three seconds.

c) Place the electromagnet above the button of the power bank. At this moment, the lights of the power bank should be off. If the button is pushed, the lights will be on.

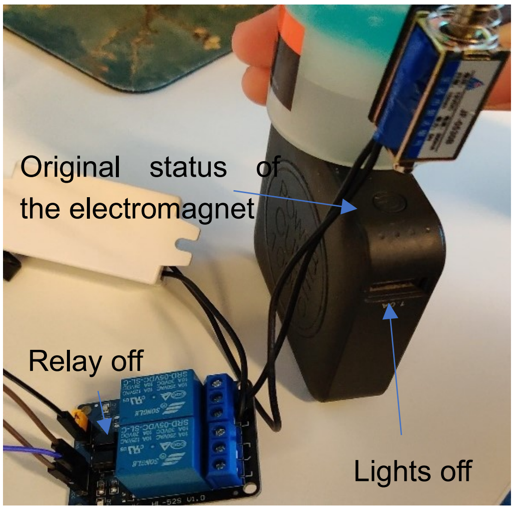


Figure 89 Electromagnet test result 1

d) Power up the transformer and observe the working status of the relay and LED.

8. Test results

When the relay is powered up, there is the clear sound ‘CLICK’, the red light on the relay is lighted. The electromagnet is triggered. The metal stick in the electromagnet stretches. The button on the power bank is pushed and the lights are illumed. After three seconds, Arduino stop to supply power, the red light on the relay and the lights on the power bank extinguish. The electromagnet recovers to its original status.

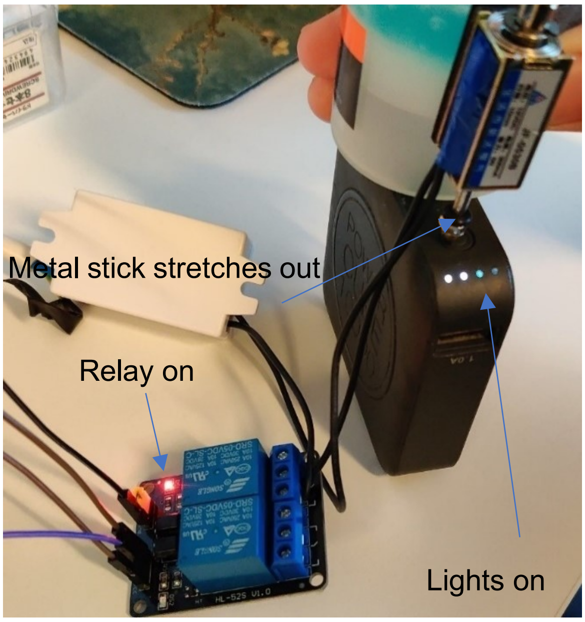


Figure 90 Electromagnet test result 2

9. Test conclusion

The electromagnet can work according to the status of the relay with the control of the Arduino. When the electromagnet is powered up, it has enough force to push the button.

13.1.4 The test of the SHT31 air temperature and moisture sensor

1. Test aim

This test aims to test if the SHT31 can measure the ambient temperature and moisture with high accuracy.

2. Desired result.

SHT31 can output data with an average error of about 0.5℃ on the temperature and 1% on the moisture. The Arduino can collect data from the SHT31 sensor.

3. Test equipment

Table 53 SHT31 test equipment

Arduino nano
SHT31 temperature and moisture sensor
Temperature and moisture meter
Jump wires

4. Test variables

Table 54 SHT31 test variables

Variables	Status
Input voltage	5V
Output temperature	-40℃-125℃
Output moisture	0%-100%

5. Test circuit

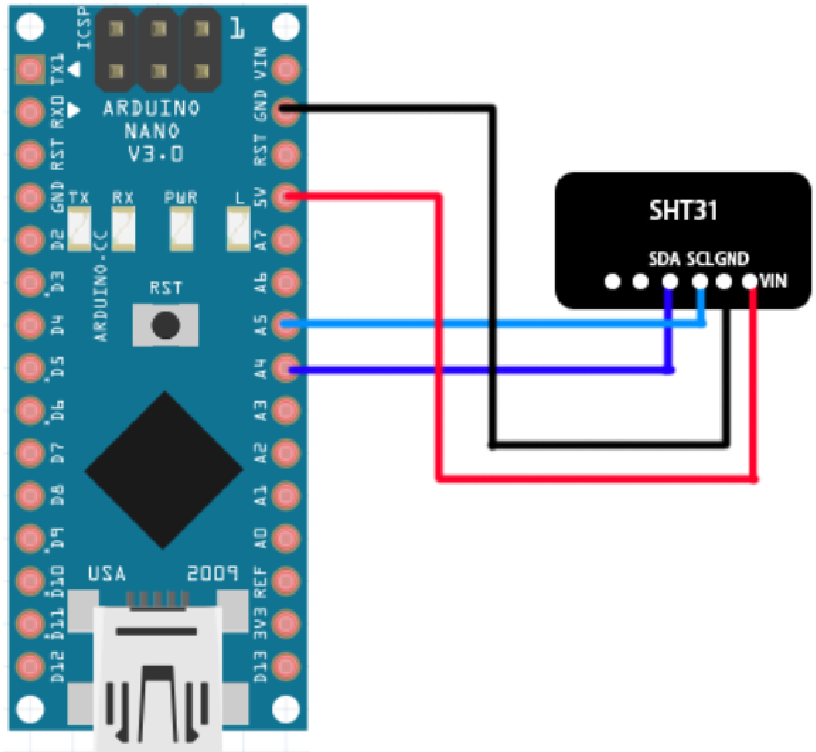


Figure 91 SHT31 test circuit

6. Test codes

```
#include <Arduino.h>
#include <Wire.h>
#include "Adafruit_SHT31.h"

Adafruit_SHT31 sht31 = Adafruit_SHT31();

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

  while (!Serial)
    delay(10);

  Serial.println("SHT31 test");
  if (! sht31.begin(0x44))
  {
    Serial.println("Couldn't find SHT31");
    while (1) delay(1);
  }
}
```

Figure 92 SHT31 test code 1

```
void loop()
{
  float t = sht31.readTemperature();
  float h = sht31.readHumidity();

  if (! isnan(t))
  {  // check if 'is not a number'
    Serial.print("Temp *C = "); Serial.print(t); Serial.print("\t\t");
  }
  else
  {
    Serial.println("Failed to read temperature");
  }

  if (! isnan(h))
  {  // check if 'is not a number'
    Serial.print("Hum. % = "); Serial.println(h);
  }
  else
  {
    Serial.println("Failed to read humidity");
  }

  delay(1000);
}
```

Figure 93 SHT31 test code 2

7. Test steps

a) Connect the equipment like the above test circuit.

b) Program the Arduino nano with the above codes and Arduino IDE. In these codes, Arduino will power up the SHT31 and collect data from it every second.

c) Observe the working status of the atomizer.

d) Put the sensor in the room and record data from the sensor five times every hour and a total of three hours.

e) Meanwhile, record the data from the temperature and moisture meter.

f) Compare the data from the sensor and the temperature and moisture meter.

8. Test results

When the SHT31 is powered up, it measures the ambient temperature and moisture and transmits data to the Arduino every second.

```
COM4

21:59:05.817 -> SHT31 test
21:59:05.910 -> Temp *C = 24.14        Hum. % = 38.93
21:59:06.920 -> Temp *C = 24.14        Hum. % = 38.94
21:59:07.949 -> Temp *C = 24.12        Hum. % = 38.97
21:59:09.021 -> Temp *C = 24.12        Hum. % = 38.97
21:59:10.047 -> Temp *C = 24.12        Hum. % = 38.98
21:59:11.077 -> Temp *C = 24.12        Hum. % = 38.98
21:59:12.153 -> Temp *C = 24.12        Hum. % = 38.99
21:59:13.181 -> Temp *C = 24.09        Hum. % = 38.98
21:59:14.207 -> Temp *C = 24.09        Hum. % = 39.01
21:59:15.282 -> Temp *C = 24.08        Hum. % = 39.06
21:59:16.310 -> Temp *C = 24.06        Hum. % = 39.05
21:59:17.340 -> Temp *C = 24.06        Hum. % = 39.03
21:59:18.418 -> Temp *C = 24.06        Hum. % = 39.11
21:59:19.443 -> Temp *C = 24.05        Hum. % = 39.07
21:59:20.495 -> Temp *C = 24.05        Hum. % = 39.14
```

Figure 94 SHT31 test result 1

After recording data fifteen times, and comparing the different data, the following list is summarized.

Temp SHT31	Temp meter	Error	Mois SHT31	Mois meter	Error
23.61	23.6	0.01	59.61%	59.5%	0.11%
23.57	23.7	-0.13	59.59%	59.5%	0.09%
23.94	23.8	0.14	59.44%	59.5%	-0.06%
23.88	23.8	0.08	59.30%	59.4%	-0.10%
23.78	23.8	-0.02	59.41%	59.4%	0.01%
22.77	22.9	-0.13	59.42%	59.4%	0.02%
22.56	22.9	-0.34	59.33%	59.4%	-0.07%
22.91	22.7	0.21	59.10%	59.3%	-0.20%
22.23	22.3	-0.07	59.04%	59.2%	-0.16%
22.12	22.1	0.02	59.01%	59.1%	-0.09%
21.89	22	-0.11	58.99%	59.1%	-0.11%
21.87	22	-0.13	58.91%	59.1%	-0.19%
21.65	21.9	-0.25	58.03%	59.0%	-0.97%
21.69	21.9	-0.21	58.05%	59.0%	-0.95%
21.6	21.8	-0.2	58.01%	59.0%	-0.99%
	Average error	0.08		Average error	0.24%

Figure 95 SHT31 test result 2

The error was calculated with the subtraction of the data from SHT31 and temperature and moisture meter. The average error of temperature measurement is only 0.08 and the average error of moisture measurement is only 0.24%

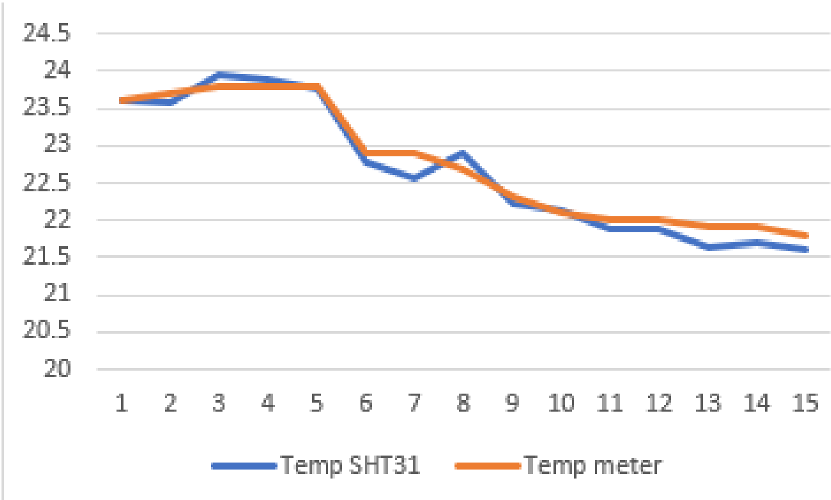


Figure 96 SHT31 test result 3

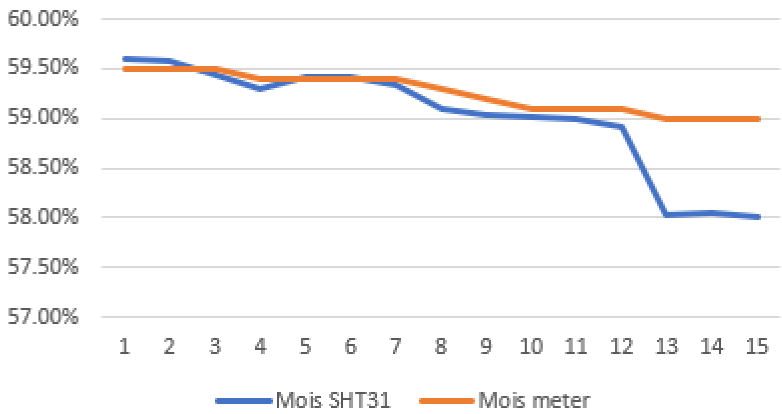


Figure 97 SHT31 test result 4

The above two graphs show the comparison between the data from SHT31 and the temperature and moisture meter.

9. Test conclusion

SHT31 can measure the ambient temperature and moisture, and output data with an average error of about 0.08℃ on the temperature and 0.24% on the moisture. The accuracy is enough for the project. The Arduino can successfully collect data from the SHT31 sensor.

.

13.1.5 The test of the thermocouple and MCP9600 thermocouple signal transformer

1. Test aim

This test aims to test if the thermocouple can test the solid temperature and transmit high accuracy data to Arduino with the MCP9600 thermocouple signal transformer.

2. Desired result.

The thermocouple can output data with an average error of about 0.5℃.

The Arduino can collect data from the thermocouple.

3. Test equipment

Table 55 Thermocouple test equipment

Arduino nano
K type thermocouple
MCP9600 thermocouple signal transformer
Infrared thermometer
Jump wires

4. Test variables

Table 56 Thermocouple test variables

Variables	Status
Input voltage	5V
Output temperature	0℃-200℃
Weather	Sunny/ Cloudy

5. Test circuit

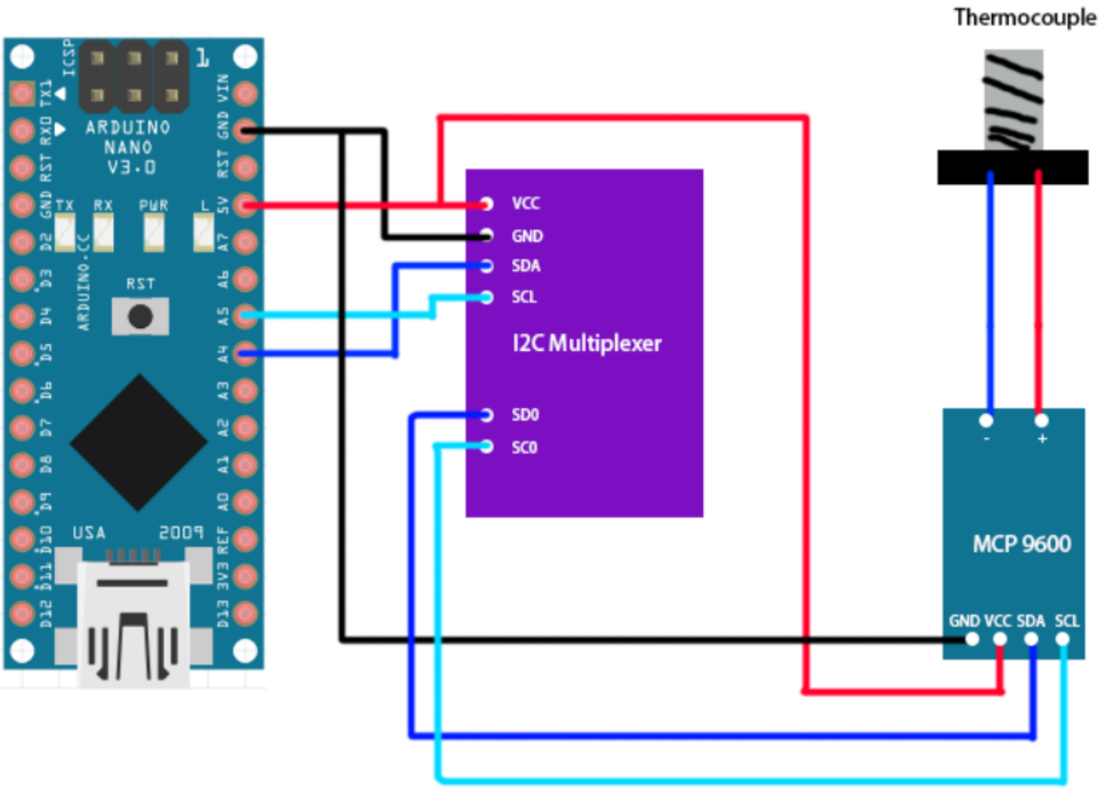


Figure 98 Thermocouple test circuit

6. Test codes

```
#include "Seeed_MCP9600.h"
#include <Arduino.h>
#include "Wire.h"
extern "C"
{
#include "utility/twi.h"  // from Wire library, so we can do bus scanning
}

#define TCAADDR 0x70

#ifdef ARDUINO_SAMD_VARIANT_COMPLIANCE
    #define SERIAL SerialUSB
#else
    #define SERIAL Serial
#endif

MCP9600 sensor;

err_t sensor_INT_config()
{
    err_t ret = NO_ERROR;
    CHECK_RESULT(ret, sensor.set_filt_coefficients(FILT_MID));
```

Figure 99 Thermocouple test code 1

```
    for (int i = 0; i < 4; i++)
    {
        /*Conver temp num to 16bit data*/
        CHECK_RESULT(ret, sensor.set_alert_limit(i, sensor.covert_temp_to_reg_form(28 + i)));
        /*  Set hysteresis.for example,set hysteresis to 2℃,when the INT limitation is 30℃,#
         *  interruption will be generated when the temp ecceed limitation,and the interruption
         flag will stay unless the temp below 30-2(limitation-hysteresis) 28℃. */
        CHECK_RESULT(ret, sensor.set_alert_hys(i, 2));

        /*Set when interruption generated the pin's status*/
        CHECK_RESULT(ret, sensor.set_alert_bit(i, ACTIVE_LOW));

        CHECK_RESULT(ret, sensor.clear_int_flag(i));

        /*default is comparator mode*/
        CHECK_RESULT(ret, sensor.set_alert_mode_bit(i, COMPARE_MODE));

        /*Set alert pin ENABLE.*/
        CHECK_RESULT(ret, sensor.set_alert_enable(i, ENABLE));
    }

    /*device cfg*/
    CHECK_RESULT(ret, sensor.set_cold_junc_resolution(COLD_JUNC_RESOLUTION_0_25));
    CHECK_RESULT(ret, sensor.set_ADC_meas_resolution(ADC_14BIT_RESOLUTION));
    CHECK_RESULT(ret, sensor.set_burst_mode_samp(BURST_32_SAMPLE));
    CHECK_RESULT(ret, sensor.set_sensor_mode(NORMAL_OPERATION));

    return NO_ERROR;
}


err_t get_temperature(float* value)
{
    err_t ret = NO_ERROR;
    float hot_junc = 0;
    float junc_delta = 0;
    float cold_junc = 0;
    bool stat = true;


   CHECK_RESULT(ret, sensor.check_data_update(&stat));
   if (stat)
   {
       CHECK_RESULT(ret, sensor.read_hot_junc(&hot_junc));
       CHECK_RESULT(ret, sensor.read_junc_temp_delta(&junc_delta));

       CHECK_RESULT(ret, sensor.read_cold_junc(&cold_junc));

       *value = hot_junc;
   } else
   {
       SERIAL.println("data not ready!!");
   }

   return NO_ERROR;
}
```

Figure 100 Thermocouple test code 2

```
void setup()
{
  SERIAL.begin(115200);
  Serial.begin(115200);
  delay(10);

  tcaselect(0);
  SERIAL.println("serial start!!");
  if (sensor.init(THER_TYPE_K))
  {
      SERIAL.println("sensor init failed!!");
  }
  sensor_INT_config();
}

void loop()
{
   tcaselect(0);
   thermocouple();
}

void tcaselect(uint8_t i)
{
  if (i > 7) return;

  Wire.beginTransmission(TCAADDR);
  Wire.write(1 << i);
  Wire.endTransmission();
}

void thermocouple()
{
 float temp = 0;
    u8 byte = 0;
    u8 stat = 0;

    tcaselect(0);
    get_temperature(&temp);
    SERIAL.print("temperature ==============================>>");
    SERIAL.println(temp);

    sensor.read_INT_stat(&stat);

    SERIAL.println(" ");
    SERIAL.println(" ");
    delay(1000);
}
```

Figure 101 Thermocouple test code 3

7. Test steps

a) Connect the equipment like the above test circuit.

b) Program the Arduino nano with the above codes and Arduino IDE. In these codes, Arduino will power up the thermocouple and the MCP9600 signal transformer and collect data from them every second.

c) Drill a hole on a bio-based material board and insert the thermocouple into the board.

Figure 102 Thermocouple test step example 1

Figure 103 Thermocouple test step example 2

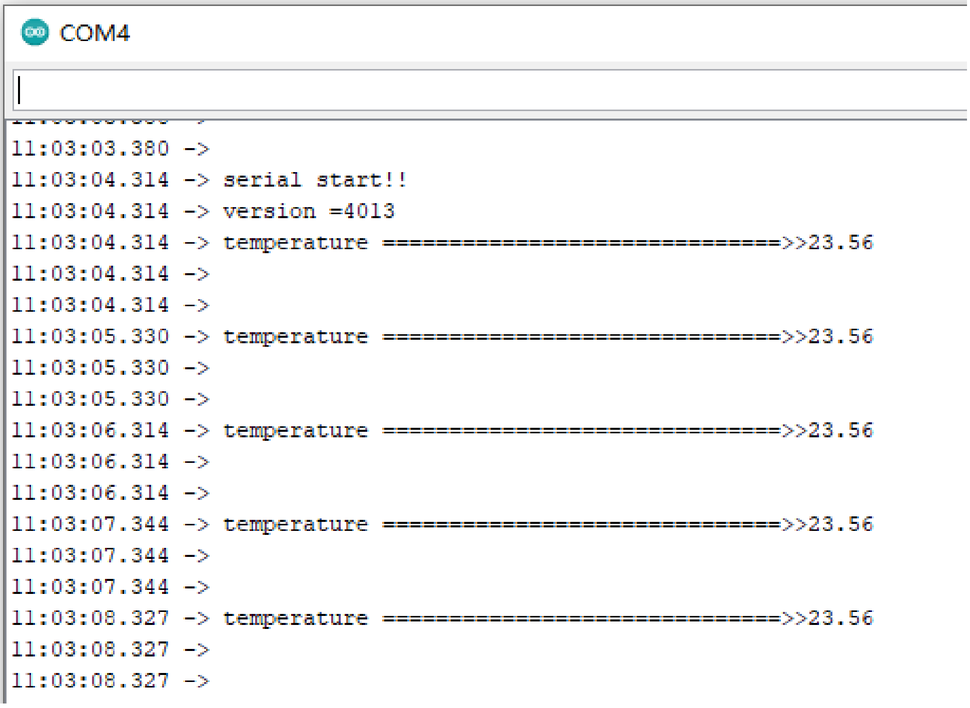
d) Drill another hole on the board and measure the temperature inside the hole. Meanwhile, record the data from the temperature and moisture meter.

e) Put the board in the sun, so that the temperature of the board can be effect by the sunlight intensity at different times. Record data every hour and totally record seven times.

f) Compare the data from the thermocouple and the Infrared thermometer.

8. Test results

When the thermocouple and the MCP9600 signal transformer are powered up, the thermocouple measures the solid temperature and transmits data to the Arduino every second.



```
COM4
11:03:03.380 ->
11:03:04.314 -> serial start!!
11:03:04.314 -> version =4013
11:03:04.314 -> temperature ==============================>>23.56
11:03:04.314 ->
11:03:04.314 ->
11:03:05.330 -> temperature ==============================>>23.56
11:03:05.330 ->
11:03:05.330 ->
11:03:06.314 -> temperature ==============================>>23.56
11:03:06.314 ->
11:03:06.314 ->
11:03:07.344 -> temperature ==============================>>23.56
11:03:07.344 ->
11:03:07.344 ->
11:03:08.327 -> temperature ==============================>>23.56
11:03:08.327 ->
11:03:08.327 ->
```

Figure 104 Thermocouple test result 1

After recording data seven times, and comparing the different data, the following list is summarized.

		Thermocouple	Temp meter	Error
SUNNY	11:10am	29.69	29.8	0.11
SUNNY	12:10am	38.31	38.9	0.59
CLOUDY	1:10pm	33.81	34	0.19
CLOUDY	2:10pm	28.75	28.6	0.15
SUNNY	3:10pm	30	29.3	0.7
CLOUDY	4:10pm	27.44	27.2	0.24
CLOUDY	5:10pm	22.75	22	0.75
			average error	0.39

Figure 105 Thermocouple test result 2

The error was calculated with the subtraction of the data from the thermocouple and temperature meter. The average error of temperature measurement is only 0.39.

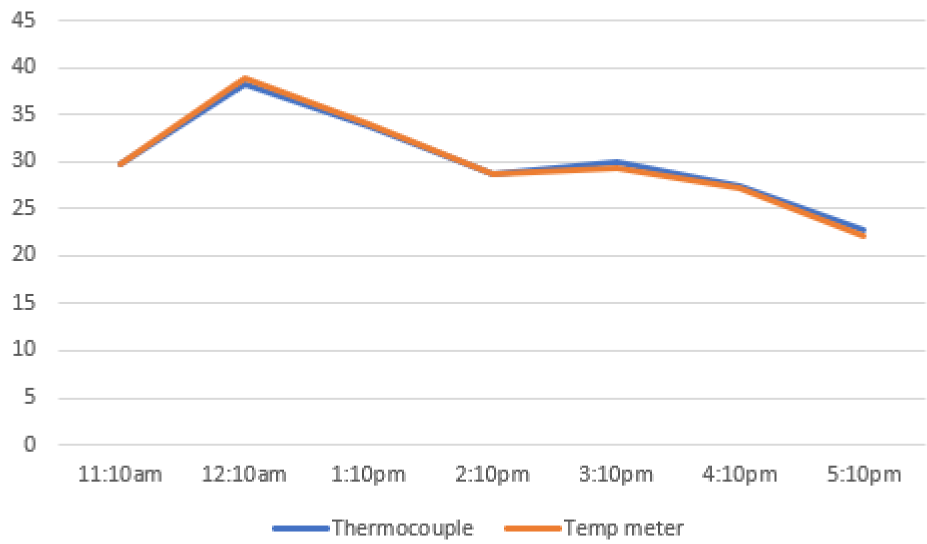


Figure 106 Thermocouple test result 3

The above graph shows the comparison between the data from the thermocouple and temperature meter.

9. Test conclusion

The thermocouple and the MCP9600 signal transformer can output high accuracy data to the Arduino with an average error of 0.39.

13.1.6 The test of the SD memory card shield module

1. Test aim

This test aims to test if the SD memory card shield module can work record data from the Arduino.

2. Desired result.

When the Arduino output some data, the SD memory card shield can record this data.

3. Test equipment

Table 57 SD memory module test equipment

Arduino nano
SD memory card shield module
Micro SD card
Jump wires

4. Test variables

Table 58 SD memory module test variables

Variables	Status
Input voltage	5V
Arduino data output	1-99

5. Test circuit

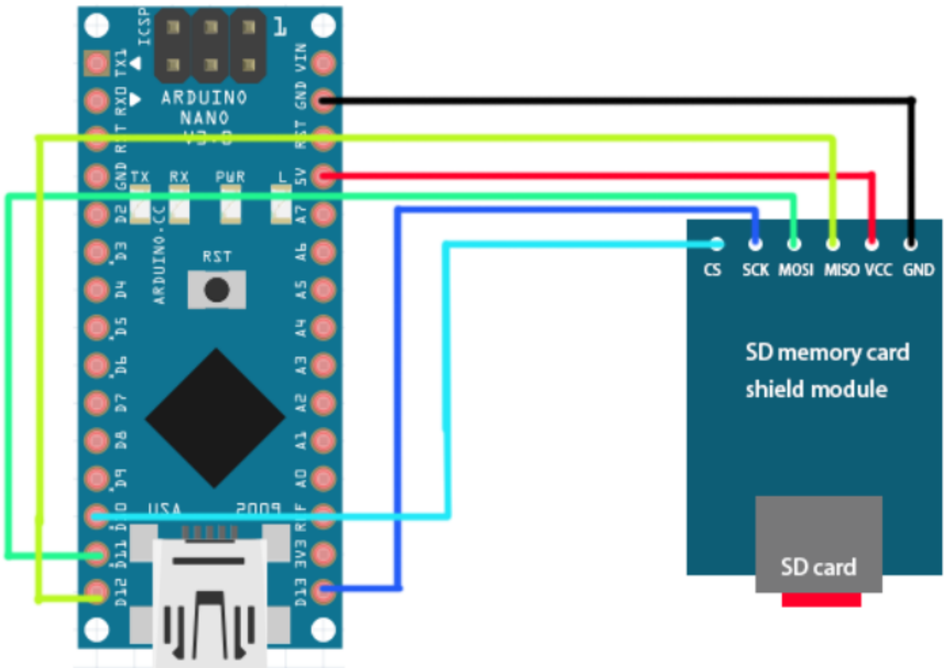


Figure 107 SD memory module test circuit

6. Test codes

```
#include <SPI.h>
#include <SD.h>
#define CS 10
long value = 0;
File logFile;
boolean initCard()
{
  Serial.print("Connecting to SD card... ");
  // Initialize the SD card
  if (!SD.begin(CS))
  {
    Serial.println("\nError: Could not connect to SD card!");
    return false;
  }
  else
    Serial.println("Done!");
  return true;
}
```

Figure 108 SD memory module test code 1

```
void setup()
{
  Serial.begin(9600);
  if (!initCard())
    while (1);

  logFile = SD.open("log.txt", FILE_WRITE);
  if (!logFile)
  {
    Serial.println("Could not open logfile!");
    while (1);
  }
}
void loop()
{
  value = value + 1;
 if(i++ < 99)
 {
    logFile.println("Value =");
    logFile.println(value);
    Serial.println("Writing finished!");
 }
 else
 {
    logFile.close();
    Serial.println("Done");
 }
  delay(500);
}
```

Figure 109 SD memory module test code 2

7. Test steps

a) Connect the equipment like the above test circuit.

b) Program the Arduino nano with the above codes and Arduino IDE. In these codes, Arduino will power up the SD memory card shield module and output a number every second. This number is incremented by one for every output from the beginning. There will be a hint 'Writing finished' to tell the user the data is output.

c) Record how many times does the Arduino output data.

d) Check the file on the SD card on the computer. And check if the value of the recorded data is the same as the data output times.

8. Test results

After counting the Arduino outputs data 18 times, check the .txt file in the SD card. There are eighteen numbers recorded which start from 1 to 18.

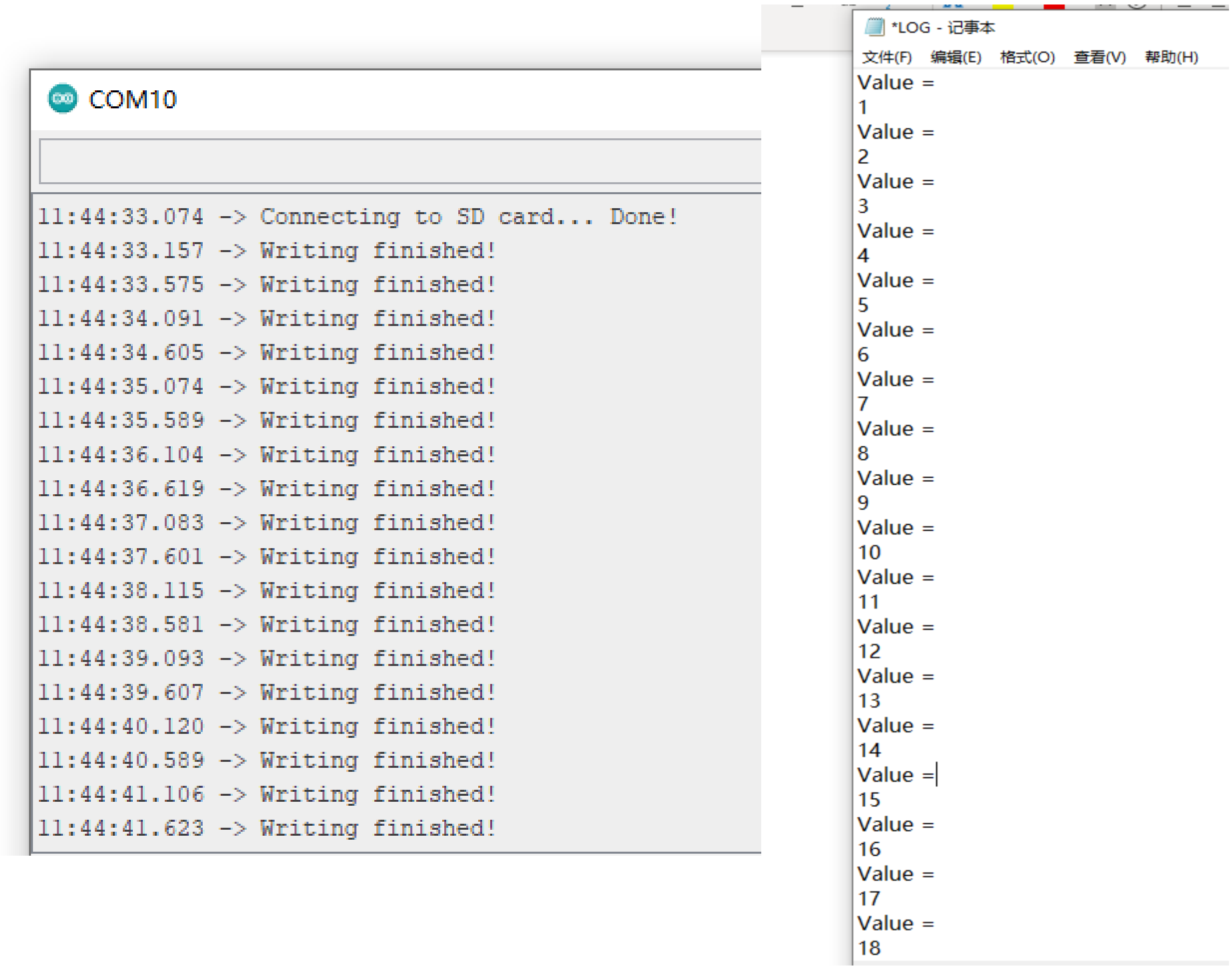


Figure 110 SD memory module test result

9. Test conclusion

The SD memory card shield can record the data output from the Arduino.

13.1.7 The test of the NRF24L01 wireless data transmission module

1. Test aim

This test aims to test if the NRF24L01 wireless data transmission module can communicate with the Arduino and transmit data from one module to another.

2. Desired result.

When the Arduino output some data, the NRF24L01 sender can receive data and send data to the receiver. The received data can be monitored by the Arduino which is connected to the receiver.

3. Test equipment

Table 59 NRF24L01 test equipment

Arduino nano
NRF24L01 wireless data transmission module
Jump wires

4. Test variables

Table 60 NRF24L01 test variables

Variables	Status
Input voltage	5V
Arduino data output	1-99
NRF24L01 sender data output	1-99
NRF24L01 receiver data input	1-99
Arduino data input	1-99

5. Test circuit

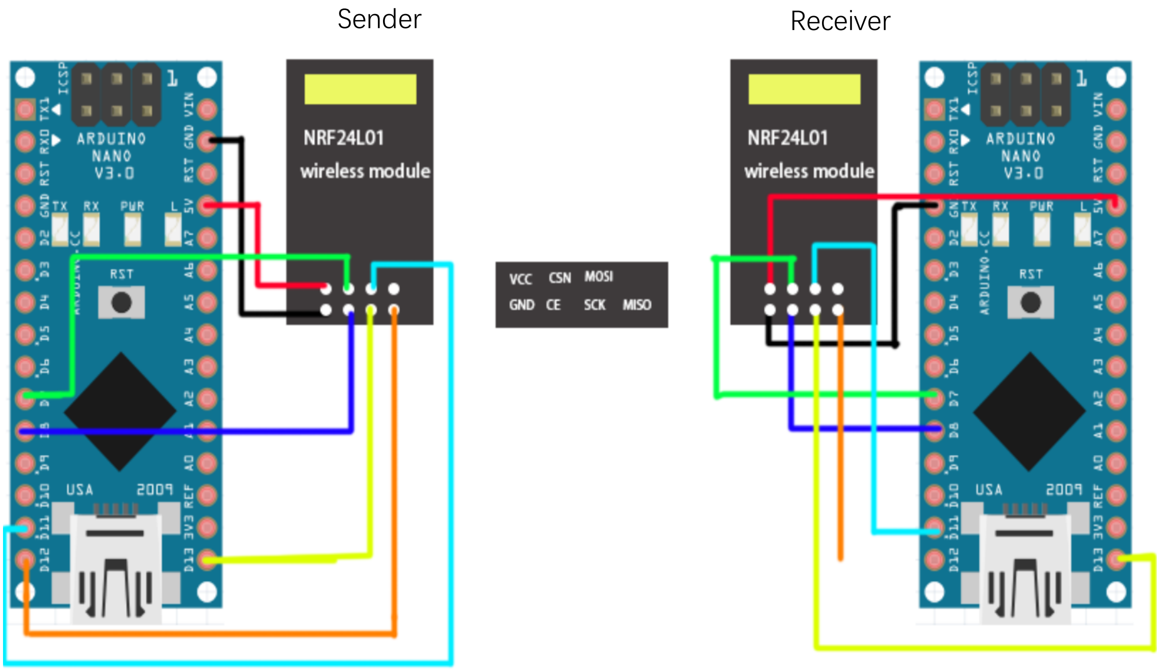


Figure 111 NRF24L01 test circuit

6. Test codes

Sender:

```
#include <SPI.h>
#include <Mirf.h>
#include <nRF24L01.h>
#include <MirfHardwareSpiDriver.h>

int value = 0;

void setup()
{
  Serial.begin(9600);
  Mirf.spi = &MirfHardwareSpi;
  Mirf.init();

  Mirf.setRADDR((byte *)"SENDE");
  Mirf.payload = 2;
  Mirf.channel = 11;
  Mirf.config();
}

void loop()
{
  Mirf.setTADDR((byte *)"RECEI");
  value = value+1;
  Serial.println(value);
  Mirf.send((byte *)&value);
  while(Mirf.isSending()) delay(1);

  delay(1000);
}
```

Figure 112 NRF24L01 test code sender

Receiver:

```
#include <SPI.h>
#include <Mirf.h>
#include <nRF24L01.h>
#include <MirfHardwareSpiDriver.h>
#include <SoftwareSerial.h>

int value;

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

  Mirf.spi = &MirfHardwareSpi;
  Mirf.init();

  Mirf.setRADDR((byte *)"RECEI");
  Mirf.payload = 2;
  Mirf.channel = 11;
  Mirf.config();
  Serial.println("Listening...");
}

void loop()
{
  if(!Mirf.isSending() && Mirf.dataReady())
  {
    Mirf.getData((byte *) &value);

    Serial.println(value);
  }
}
```

Figure 113 NRF24L01 test code receiver

7. Test steps

a) Connect the equipment like the above test circuit.

b) Program the Arduino nano with the above codes and Arduino IDE. In these codes, Arduino will power up the NRF24L01 wireless data transmission module and output a number every second. This number is incremented by one for every output from the beginning.

c) Using the port monitor in the Arduino IDE to monitor the output from the Arduino connected to the sender and input from the Arduino connected to the receiver.

d) Compare the data from sender and receiver.

8. Test results

After the Arduino connected to the sender outputs a number, the Arduino connected to the receiver receives this number after about 50ms.

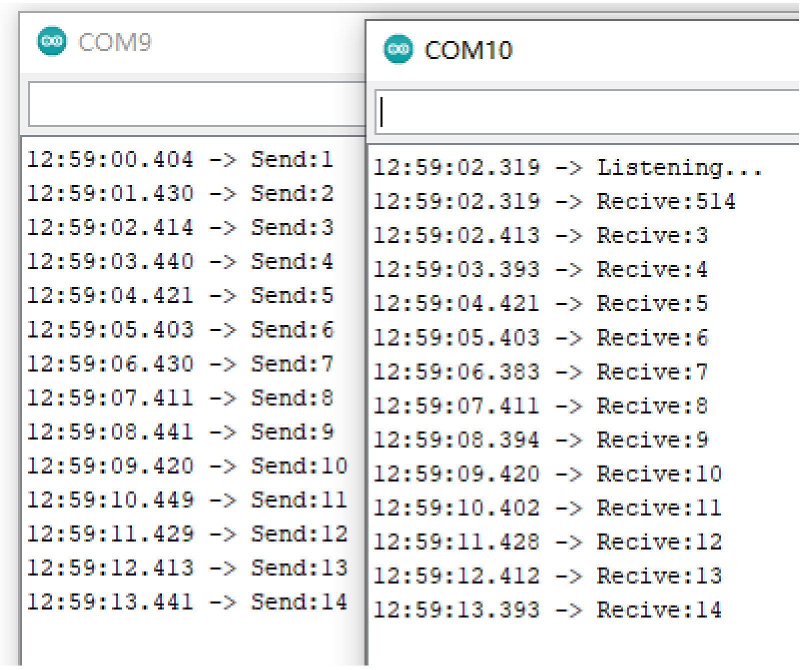


Figure 114 NRF24L01 test result

The number received by the receiver is the same as the number sent by the sender.

9. Test conclusion

The NRF24L01 wireless data transmission system can communicate with the Arduino and can be used for wireless data transmission in the project.

13.1.8 The test of the ESP32 WIFI module and the Thingspeak online datalogger

1. Test aim

This test aims to test if the ESP32 WIFI module can upload data to the online data logger and test if the data can be displayed on the Thingspeak online datalogger.

2. Desired result.

The Thingspeak datalogger can receive the data sent by the ESP32 WIFI module and display those data.

3. Test equipment

Table 61 ESP32 test equipment

ESP32 WIFI module
Thingspeak online data logger account
Jump wires

4. Test variables

Table 62 ESP32 test variables

Variables	Status
Input voltage	5V
ESP32 data output	1-99/255-0
Thingspeak online datalogger data input	1-99/255-0

5. Test circuit

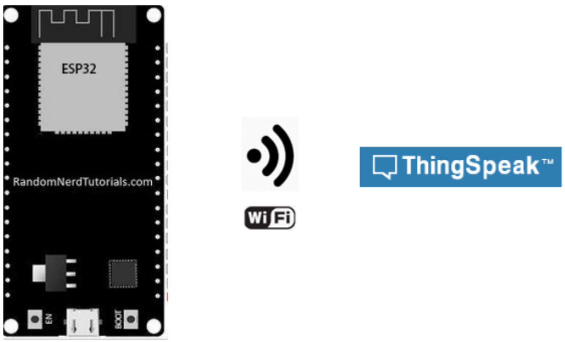


Figure 115 ESP32 test circuit

6. Test codes

```
#include <WiFi.h>
#include <Wire.h>
#include <SPI.h>
#include <HardwareSerial.h>

WiFiClient  client;

 /*HardwareSerial serialESP(1);*/

String thingspeak_key = "VNKYOOFQ43GV1OGL";
char wifi_SSID[]="DeRuyter2.4GHz";
char wifi_Password[]="student2017";
int value=0;
int value2=250;
void setup()
{
  Serial.begin(9600);
   /*serialESP.begin(9600,SERIAL_8N1,5,18);*/
  WiFi.mode(WIFI_STA);
  delay(500);
  WiFi.begin(wifi_SSID,wifi_Password);
  Serial.print("\nconnecting wifi");
  while(WiFi.status() != WL_CONNECTED)
  {
    delay(2500);
    Serial.print(".");
  }
  Serial.print("\nconnected: ");
  Serial.print(WiFi.SSID());
  Serial.print("\nIP: ");
  Serial.print(WiFi.localIP());
}
```

Figure 116 ESP32 test code 1

```
void loop()
{
 /* value = serialESP.read();*/
/* value = Serial.read();*/
  value=value+1;
  value2=value2-1;
  Serial.println(value);

  if (client.connect("api.thingspeak.com", 80))
  {
   String url = "/update?key=";
   url += thingspeak_key;
   url += "&field1=";
   url += String(value);
   url += "&field2=";
   url += String(value2);
   client.print(String("GET ") + url + " HTTP/1.1\r\n" +
               "Host: api.thingspeak.com\r\n" +
               "Connection: close\r\n\r\n");
  }
  delay(3000);

}
```

Figure 117 ESP32 test code 2

7. Test steps

a) Connect the equipment like the above test circuit.

b) Program the ESP32 with the above codes and Arduino IDE. In these codes, ESP32 will transmit data to the data logger through the WIFI every 3 seconds. Two groups of numbers will be transmitted. The first group of numbers is decreased by one for every output of the ESP32 from the beginning of 250. The second group of numbers is incremented by one for every output from the beginning of 0.

c) Observe the status of the ESP32 with the Arduino IDE.

d) Log in the Thingspeak datalogger.

e) Observe the number change in the datalogger.

8. Test results

In a few seconds, after the ESP32 is programmed, the port monitor in the Arduino IDE shows the ESP32 has successfully connected to the WIFI and starts to output numbers.

```
13:47:21.214 -> connecting wifi..
13:47:26.251 -> connected: DeRuyter2.4GHz
13:47:26.251 -> IP: 172.19.0.141value =
13:47:26.300 -> 1
13:47:26.300 -> value2 =
13:47:26.300 -> 249
13:47:29.436 -> value =
13:47:29.436 -> 2
13:47:29.436 -> value2 =
13:47:29.436 -> 248
13:47:32.519 -> value =
13:47:32.519 -> 3
13:47:32.519 -> value2 =
13:47:32.519 -> 247
```

Figure 118 ESP32 test result 1

The numbers received by the datalogger are displayed in two graphs:

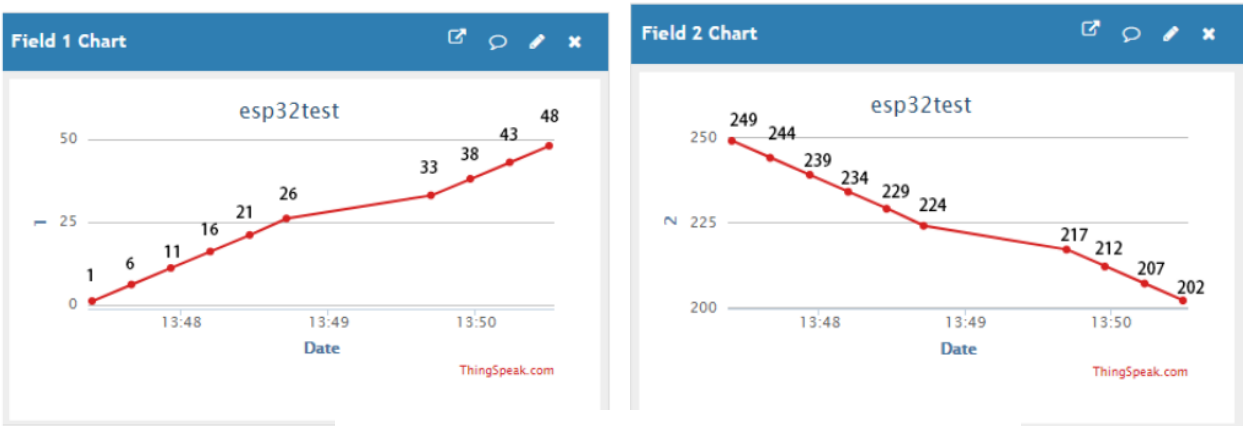


Figure 119 ESP32 test result 2

Because of the limit of the upload speed of the datalogger, the datalogger can only receive data every 15 seconds. That is why the number increases by 5 every time. And when there is a net delay, the data input delay happens (between numbers 26-33 in the first graph and numbers 224-217 in the second graph).

9. Test conclusion

The ESP32 WIFI module can upload data to the Thingspeak online datalogger. The data can be displayed in the Thingspeak online datalogger.