



TRƯỜNG ĐẠI HỌC FPT

FPT UNIVERSITY - HOA LAC CAMPUS
DEPARTMENT OF ELECTRONIC
COMMUNICATIONS
GRADUATION THESIS FINAL REPORT

ENVIRONMENTAL DATA COLLETION AND MONITORING

Pham Minh Thang

A thesis submitted in part fulfillment of the
degree of BSc. in
Electronic Communications with
the supervision and moderation of Mr. **Hoang Xuan Son**

Hoa Lac FPT University, Ha Noi
December 15th, 2021

Abstract

Environmental pollution is not a new phenomenon, yet it remains the world's greatest problem facing humanity, and the leading environmental causes of morbidity and mortality. Man's activities through urbanization, industrialization, mining, and exploration are at the forefront of global environmental pollution. Both developed and developing nations share this burden together, though awareness and stricter laws in developed countries have contributed to a larger extent in protecting their environment. Despite the global attention towards pollution, the impact is still being felt due to its severe long-term consequences.

Environmental pollution is an international serious, acute, health and social problem that humans are responsible for. Rapid technology evolution could provide us solutions to predict and monitor environmental parameters and thus minimize or even eliminate environmental pollution. Using a wireless sensor network (WSN) and an Internet of Things (IoT) implementation, we can create an eco-friendly and controlled environment. This paper presents an embedded and standalone system that can monitor, store and analyze environmental data as well as calculate energy consumption in specific settings. Information is available via a web application with central management abilities. Our appliance can work standalone as well as part of a smart ICT grid of appliances. The system aims to determine and designate environmental spaces, in which human activities could be harmful. More specific, it can be used at world cultural heritage and archaeological places, building, constructions where allowance of human intervention is prohibited or very limited. Collecting and analyzing data can help us estimate the environmental consequences of human actions.

Acknowledgement

I would like to express the deep and sincere gratitude to the Electronic and Communications Department at FPT University. Without their supporting and facilities this thesis would not have been possible.

I am also grateful to all of whom I have had the pleasure to collaborating on this and other related projects. I would extremely thankful to my supervisor, Mr. **Hoang Xuan Son**, for his kindness help and expertise in developing the research problem and methodology. Your insightful feedback pushed me to improved my thought and lever- age my work to a higher level.

Most importantly, I want to express my deepest appreciation to my parents, whose love and support are with me in whatever I pursue.

Contents

1 Introduction	7
1.1 Introduction	7
1.2 Related Works	8
1.2.1 RMS	8
1.2.2 Cority's Environmental Management System	9
1.3 Main contribution	9
1.4 Outline	10
2 Project Management Plan	11
2.1 Overview	11
2.1.1 WBS & Estimation	11
2.1.2 Project Objective	12
2.1.3 Project Risk	12
2.2 Management Approach	13
2.2.1 Project Progress	13
2.2.2 Quality Management	14
2.2.3 Training Plan	14
2.3 Master Schedule	15
2.4 Project Communication	15
2.4.1 Communication Plan	15
2.4.2 External Interface	16
2.5 Configuration Management	16
2.5.1 Tools & Infrastructures	16
2.5.2 Document Management	16
3 NodeMCU, sensors and Blynk	17
3.1 NodeMCU	17
3.1.1 What is NodeMCU	17

3.1.2	NodeMCU Development Kit/Board	19
3.1.3	ESP8266 WiFi Module	21
3.2	Sensors	22
3.2.1	Introduction	22
3.2.2	Some type of Sensors	24
3.3	Blynk	29
3.3.1	What is Blynk?	29
3.3.2	Blynk Server, Blynk App and Blynk Libraries	29
3.3.3	Alternatives to Blynk	31
4	Implement Phase	32
4.1	Architecture	32
4.2	Implement	33
4.2.1	Setup device and setup Blynk.	33
4.2.2	Connect device to Blynk	35
4.3	Code	37
5	Project Result	39
5.1	Packaging Project	39
5.2	Practical Result.	40
6	Conclusion and Future plan	42
6.1	Conclusion	42
6.2	Future Plan.	43

List of Figures

3.1a	NodeMCU Development Board/kit v1.0 (Version2)	18
3.1b	NodeMCU Development Board/kit v0.9 (Version1)	18
3.2	NodeMCU Dev Kit v1.0(V2) Pinouts	20
3.3	ESP8266-01 WiFi Module	21
3.4	Arduino Sensors	23
3.5	Grove – Air Quality Sensor v1.3	24
3.6	Dust Sensor	25
3.7	Water level sensor	25
3.8	Rain drop sensor	26
3.9	Sound sensor	27
3.10	DHT11 temperature-humidity sensor	28
3.11	Blynk	29
3.12	Blynk Web Dashboard	30
3.13	Blynk Mobile Dashboard	30
4.1	Project Architecture	32
4.2	Circuit	33
4.3	Datastreams configuration	34
4.4	Web Dashboard Setup	34
4.5	Mobile Dashboard Setup	35
4.6	Blynk Template ID & Device Name	36
4.7	“Adding New Device” Screen	37
5.1	Project device packaging.	39
5.2a	Serial monitor	40
5.2b	Web interface	40
5.3	Mobile interface and Alarm notification	41

CHAPTER *1*

Introduction

"Nature is not simply a technical or economical resource, and human beings are not mere numbers. To suggest that one can somehow align all the squabbling institutions of science, environmental management, government and diplomacy in an alliance of convenience to regulate the global climate seems to me optimistic."

James Buchan

1.1 Introduction

Pollution has significant impacts on human health, the environment, and even on how some of the Earth's systems, such as the climate, are functioning. Pollution touches all parts of the planet. It is affecting our health through the food we eat, the water we drink and the air we breathe. Approximately 19 million premature deaths are estimated to occur annually as a result of the way we use natural resources to support global production and consumption and which impact the environment

As such, environmental monitoring is the continuous, automated observation of changes in the environment. Powerful models, methods, and systems that have been developed for the purpose of environmental monitoring are now leaving the laboratory setting, and some of them are already being applied. When they are applied in countries that have well established environmental authorities and a high population density, these applications are actually additional measures to

protect the environment. However, the situation in less developed or less densely populated countries is quite different. For these countries, the new techniques of environmental monitoring can offer totally new opportunities for better environmental protection.

Over the past decade, only a few countries of Europe and Central Asia have been able to maintain existing monitoring activities. The monitoring of urban air pollution — an important human health risk — is poor in many cities of the sub region. Solid and hazardous waste monitoring is weak and industrial emissions are also not well monitored, reducing the effectiveness of policy instruments such as emissions charges and fines. Monitoring of transboundary air pollution also needs strengthening. Moreover, many European and Central Asian countries lack uniform national methodologies across different monitoring areas, and their classification systems are often incompatible with international standards.

The three main types of environmental monitoring are soil, atmosphere, and water. Some techniques of environmental scanning and monitoring include filtration, sedimentation, electrostatic samples, impinges, absorption, condensation, grab sampling, and composite sampling.

Environmental monitoring solutions have evolved over the years into Smart Environmental Monitoring (SEM) systems that now incorporate modern sensors, Machine Learning (ML) techniques, and the Internet of Things (IoT). Technologies such as IoT devices and wireless sensor networks have made advanced environmental monitoring using IoT a more streamlined and Artificial Intelligence-controlled process.

Data captured by IoT environmental monitoring sensors from a wide variety of environmental conditions can be integrated via the Wireless Sensor Network (WSN) into one, cloud-based environmental system, in which IoT devices embedded with ML can record, characterize, monitor, and analyze elements in a specific environment.

1.2 Related Works

The advantages of environmental monitoring lie in its ability to improve the quality of life for society by highlighting the relationship between the environment and health. Transforming environmental monitoring data into information and communicating actionable insights to the community in a timely manner is crucial for keeping citizens informed of the state of their environment.

Below is some example of environmental monitoring system

1.2.1 RMS

The RMS software allows for a real time monitoring of any parameter required. Rotronic offer solutions for relative humidity, temperature, dew and frost point, differential pressure, pressure, flow, lux and CO₂. For all other parameters, Rotronic offer an analogue to digital converter so that all analogue signals can be monitored. The RMS-Converter will allow you to integrate any digital product (with a LAN connection) into the RMS software giving you the highest flexibility possible.

— The RMS hardware itself gives you the possibility to log down to 10s intervals and the data can be viewed via the RMS webpage if you choose the cloud solution, but also via your own server should you choose the server version.

The software offers charts and graphics, as well as an alarming function (Email, SMS, Telephone call). The analysis tool will allow for you to establish daily, weekly, monthly or personalized reports with all of the statistical data required.

1.2.2 Cority's Environmental Management System (EMS)

Cority's Environmental System helps companies centralize and streamline the tracking and collection of key corporate environmental health and safety data and satisfy Environmental Management Systems (EMS) requirements.

Easily collect and report on quantitative and qualitative data to track air, water and waste, and to gain actionable insights to improve operational performance. Their solution provides a 360° view of all programs and makes it easy for managers to collaborate, measure, control, and reduce environmental impacts.

Cority's powerful reporting capabilities turn strategic and operational objectives into measurable success indicators to determine the effectiveness of your Operational Excellence program. Their approach enables collaboration across different business units so you gain a holistic view of operations to drive faster alignment on key issues, reduce waste and costs.

EMS designed to help you streamline, standardize, and improve compliance management to ensure a closed-loop compliance plan in a single, configurable, and unified platform. Their EHS solution supports a 'management systems' approach to improve management and operations for regulations, permits, international standards and company policies in one system.

1.3 Main contribution

In this works, we demonstrate the ability to observe the change of environment, be able to know and making a timely response whenever the sudden change in environment which can be dangerous is occur. With the power of IoT, by using sensors and proper application, we can make a device that fulfill the requirement. All of related works mention above have the same disadvantage of high cost and only good when use in big organization. But in our project, the cost is much more cheap and individual friendly, with same function or even more modern if you invested time and effort to it.

Specifically, we make the following contributions:

- Our work can observe the change of data collect from environment, such as temperature, humidity, etc. in a real time from anywhere, even if you are far away from the device you use to collecting data.

-
- Our works can identify and make a warning to the user if the data collected surpass the safe threshold of environment. User then can have a correct solution to respond to the change of environment.

1.4 Outline

In thesis, we giving a details of the project, specifically:

In **Chapter 1**, we give a gentle introduction and related works about environmental data collection and monitoring

In **Chapter 2**, we give a details about our management plan to be able to make this project easier to approach and have a proper insight of what we have to do to be successful.

In **Chapter 3**, we give an introduction of the hardware and software that can potentially be use for environmental monitoring and collecting data.

In **Chapter 4**, we make a details about the implement phase with step by step of what we do, what we will use in this project, with architecture, codes and the step of how to setup.

In **Chapter 5**, we showed our results of project, what we have accomplished.

In **Chapter 6**, we concluded the thesis and thought about the future of project.

CHAPTER 2

Project Management Plan

2.1 Overview

2.1.1 WBS & Estimation

#	WBS Item	Complexity	Est.Effort (man-days)
1	Researching		
1.1	NodeMCU	Medium	15
1.2	Sensors	Medium	10
1.3	Blynk	Complex	30
2	Function		
2.1	Data Collecting	Medium	7
2.2	Data Saving	Medium	4
2.3	Notify when over the threshold	Easy	2
2.4	Export Data	Medium	5
3	Setup and Testing		
3.1	Setting up circuit	Easy	1
3.2	Testing and fixing	Complex	20

Total Estimated Effort (man-days) 93

2.1.2 Project Objectives

Project must be finished before 15/12/2021 with these main deliverables:

#	Quality Stage	No. of Defects	% of Defect	Notes
1	Complete the project plan within 1 weeks	9	11.8	
2	Complete the research within 8 weeks	40	52.6	
3	Complete the Architecture design within 7 weeks	8	10.5	
4	Complete code phase within 4 weeks	9	11.8	
5	Complete unit testing phase within 13 weeks	10	13.2	

Total

76

100%

2.1.3 Project Risks

#	Risk Description	Impact	Possibility	Response Plans
R01	Lack of technical skill and knowledge	Can't do project properly	High	Have a meeting with supervisor to understand requirement and know what to do
R02	Difficulty of the requirements	Cannot met the expected requirement	High	Change the requirement according to ability
R03	Failure to meet deadline	Project got delay, don't have enough time to	Medium	Find the root cause of the problem Focus on important

		finish		function first
R04	Sudden illness	Project delay	Medium	Try to work overtime to fill the gap in the ill time
R05	Components doesn't work correctly	Make project difficult to find out if the bug is from code or components	High	Always have spare components to replace the broken

2.2 Management Approach

2.2.1 Project Progress

Organization structure above includes 4 sub-works: Analysis & Design, Coding, Documentation and Testing.

This is the detailed description about responsibility in the organization structure:

Role	Name	Responsibility
Supervisor	Hoang Xuan Son	Give instruction for the project Verify deliverables User guide Q&A document
Project Member	Pham Minh Thang	Have overall responsibility of the project Assign task Tracking work Report working status to the instructor
Analysis & Design		
Member	Pham Minh Thang	Drawing Architecture design Screen design Workflow Build circuit and packing Use case specification
Coding		
Member	Pham Minh Thang	Decide technique and tools to be used Keeping track of development work done Coding Convention
Documentation		

Member	Pham Minh Thang	User guide Tracking work Q&A document Doing task document
Testing		
Member	Pham Minh Thang	Create Test Plan. Create Test Cases. Execute Test. Log test report Fix bugs Review Test-cases and reports

2.2.2 Quality Management

The reasons that we choose Iterative and Incremental model are:

- Can develop prioritized requirements first.
- Initial product delivery is faster.
- Requirements changes can be easily accommodated.
- Easier to test

2.2.3 Training Plan

Training Area	Duration	Walver Criteria
Arduino	24/9-30/9	Mandatory
NodeMCU	1/10-14/10	Mandatory
Blynk	15/10-15/11	Mandatory

2.3 Master Schedule

Main Project Deliverables:

#	Deliverable	Due Date	Deliverable Scope
1	Initiating	19/09/2021	Prepare Project Ideals
2	Project Plan	29/09/2021	Prepare Project Plan, Create Technical Prototype, Technical Training/ Self study
3	Analyzing(SRS)	06/10/2021	Write SRS Document
4	Design	15/10/2021	Architecture Design, Detailed Design
5	Code Package	03/12/2021	Code & Unit test, System test cases
6	Final Package	15/12/2021	Final Codes & documents

2.4 Project Communication

2.4.1 Communication Plan

Communication Item	Target	Purpose	Frequency	Type, Tool, Method(s)
Schedule meeting	With supervisor	Discuss, review project and report	Weekly	Online meeting
Unscheduled meeting	With supervisor	Discuss, review project and report	Random	Online meeting or directly meeting

2.4.2 External Interface

FU Contacts

Function	Contact Person	Contact address	Responsibility
Supervisor	Hoang Xuan Son	SonHX@fe.edu.vn	- Provide document template - Give instruction to project - Review deliverables - Supervise project status

2.5 Project Communication

2.5.1 Tools & Infrastructures

a Hardware

- Minimum Requirement Specifications:
- CPU: Intel® Core™ i5 2.3 GHz
- RAM: 6GB
- Hard Drive: 80GB of free space
- Internet network connection.

b Software

Category	Tools or technique name
Window	Microsoft Windows 10 Professional
	Microsoft Word 2016
	Microsoft Excel 2016
Management tools	Google Drive
Communication tools	Gmail
	Facebook
	Messenger
	Google Meet
Design tools	Blynk
Programming languages	C++
Framework	Wiring
IDE	Arduino
Management bug	Notepad
Process model	Iterative and Incremental Software Process Model

2.5.2 Document Management

We are using Microsoft Word 2016 to manage documents for the Project

CHAPTER 3

NodeMCU, sensors and Blynk

3.1 NodeMCU

3.1.1 What is NodeMCU?

NodeMCU is an open-source LUA based firmware developed for the ESP8266 wifi chip. By exploring functionality with the ESP8266 chip, NodeMCU firmware comes with the ESP8266 Development board/kit i.e. NodeMCU Development board. Since NodeMCU is an open-source platform, its hardware design is open for edit/modify/build.

There is Version2 (V2) available for NodeMCU Dev Kit i.e. NodeMCU Development Board v1.0 (Version2), which usually comes in black colored PCB.

NodeMCU Dev Kit has Arduino like Analog (i.e. A0) and Digital (D0-D8) pins on its board.

It supports serial communication protocols i.e. UART, SPI, I2C, etc.

Using such serial protocols, we can connect it with serial devices like I2C enabled LCD display, Magnetometer HMC5883, MPU-6050 Gyro meter + Accelerometer, RTC chips, GPS modules, touch screen displays, SD cards, etc.

NodeMCU Development board is featured with wifi capability, analog pin, digital pins, and serial communication protocols.

To get started with using NodeMCU for IoT applications first we need to know about how to write/download NodeMCU firmware in NodeMCU Development Boards. And before that where this NodeMCU firmware will get as per our requirement.

There are online NodeMCU custom builds available using which we can easily get our custom NodeMCU firmware as per our requirement.

To write codes for NodeMCU, we can use either ESPlorer IDE or Arduino IDE.

There is a programming language difference we can say while developing an application for NodeMCU using ESPlorer IDE and Arduino IDE.

We need to code in C\C++ programming language if we are using Arduino IDE for developing NodeMCU applications and Lua language if we are using ESPlorer IDE.

Basically, NodeMCU is Lua Interpreter, so it can understand Lua script easily. When we write Lua scripts for NodeMCU and send/upload it to NodeMCU, then they will get executes sequentially. It will not build a binary firmware file of code for NodeMCU to write. It will send Lua script as it is to NodeMCU to get executed.

In Arduino IDE when we write and compile code, the ESP8266 toolchain in the background creates a binary firmware file of code we wrote. And when we upload it to NodeMCU then it will flash all NodeMCU firmware with newly generated binary firmware code. In fact, it writes the complete firmware.

That's the reason why NodeMCU not accept further Lua scripts/code after it is getting flashed by Arduino IDE. After getting flashed by Arduino sketch/code it will be no more Lua interpreter and we got an error if we try to upload Lua scripts. To again start with Lua script, we need to flash it with NodeMCU firmware.

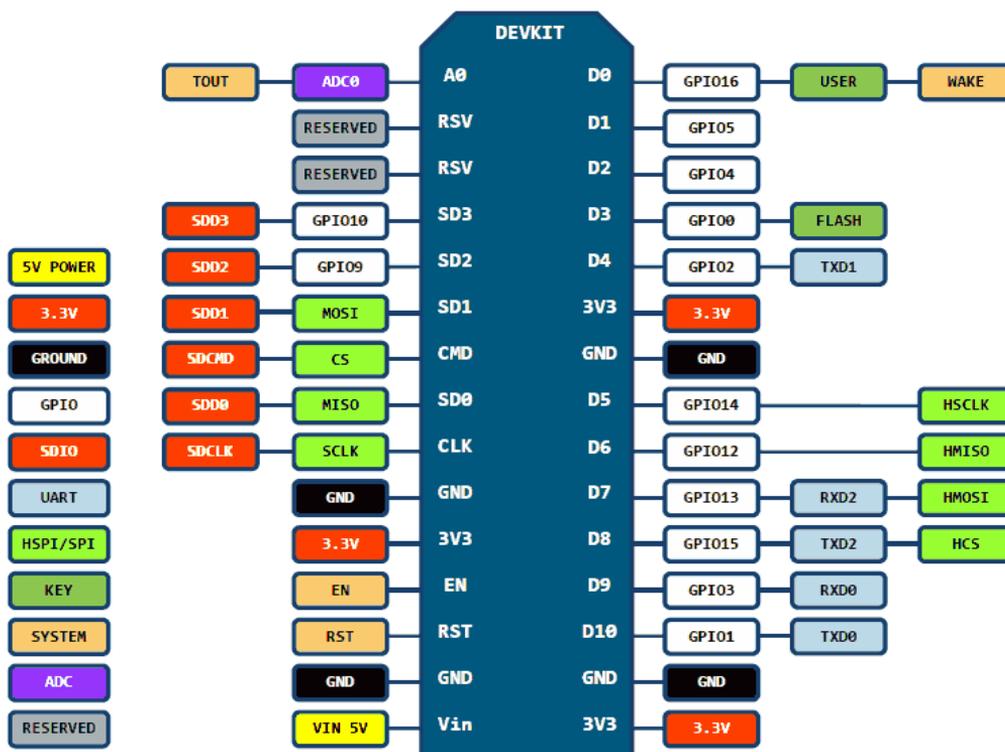
Since Arduino IDE compiles and upload/writes complete firmware, it takes more time than ESPlorer IDE.

3.1.2 NodeMCU Development Kit/Board

NodeMCU Development Kit/Board consists of an ESP8266 wifi chip. ESP8266 chip has GPIO pins, serial communication protocol, etc. features on it.

The features of ESP8266 are extracted on the NodeMCU Development board. NodeMCU (LUA based firmware) with Development board/kit that consists of ESP8266 (wifi enabled chip) chip combines NodeMCU Development board which make it a stand-alone device in IoT applications.

PIN DEFINITION



D0(GPI016) can only be used as gpio read/write, no interrupt supported, no pwm/i2c/ow supported.

Figure 3.2: NodeMCU Dev Kit v1.0(V2) Pinouts

NodeMCU Dev Kit pin descriptions

GPIO (General Purpose Input Output) Pins:

NodeMCU has general-purpose input-output pins on its board as shown in the above pinout diagram. We can make it digital high/low and control things like LED or switch on it. Also, we can generate a PWM signal on these GPIO pins.

ADC (Analog to Digital Converter) channel (A0):

NodeMCU has one ADC channel/pin on its board.

SPI (Serial Peripheral Interface) Pins:

NodeMCU based ESP8266 has Hardware SPI (HSPI) with four pins available for SPI communication. It also has SPI pins for Quad-SPI communication. With this SPI interface, we can connect any SPI enabled device with NodeMCU and make communication possible with it.

I2C (Inter-Integrated Circuit) Pins:

NodeMCU has I2C functionality support on ESP8266 GPIO pins. Due to internal functionality on ESP-12E, we cannot use all its GPIOs for I2C functionality. So, do tests before using any GPIO for I2C applications.

UART (Universal Asynchronous Receiver Transmitter) Pins:

NodeMCU based ESP8266 has two UART interfaces, UART0 and UART1. Since UART0 (RXD0 & TXD0) is used to upload firmware/codes to the board, we can't use them in applications while uploading firmware/codes.

We can make difference in the 1st and 2nd versions of the NodeMCU Development board by their board's design and ESP modules on it.

- In 1st version of NodeMCU Dev Kit v0.9, CH341SER USB to Serial converter is used whereas, in 2nd version of NodeMCU Dev Kit v1.0, CP2102 USB to Serial converter is used.

- 1st version uses ESP-12 and 2nd version uses ESP-12E (Enhanced version).

- Extra 6 pins (MTDO, MTDI, SD_3, MTMS, MTCK, SD_2) brought out on the ESP-12E version of ESP-12 modules as shown in the below figure. Though Quad SPI pins are brought out, they are internally used for flash memory access.

- Also, there is a slight antenna design difference in ESP-12 versions like ESP12-E & ESP-12F

3.1.3 ESP8266 WiFi Module

ESP8266 is Wi-Fi enabled system on chip (SoC) module developed by Espressif system. It is mostly used for development of IoT (Internet of Things) embedded applications.

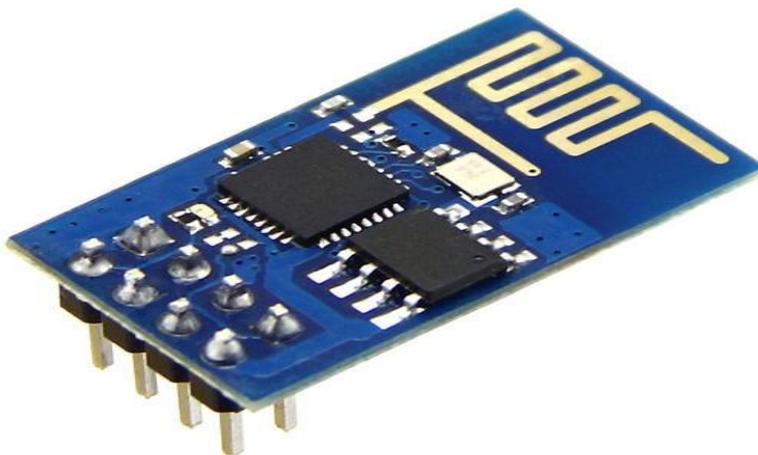


Figure 3.3: ESP8266-01 WiFi Module

— ESP8266 comes with capabilities of

- 2.4 GHz Wi-Fi (802.11 b/g/n, supporting WPA/WPA2),
- general-purpose input/output (16 GPIO),
- Inter-Integrated Circuit (I²C) serial communication protocol,
- analog-to-digital conversion (10-bit ADC)
- Serial Peripheral Interface (SPI) serial communication protocol,
- I²S (Inter-IC Sound) interfaces with DMA (Direct Memory Access)

(sharing pins with GPIO),

- UART (on dedicated pins, plus a transmit-only UART can be enabled on GPIO2), and pulse-width modulation (PWM).

It employs a 32-bit RISC CPU based on the Tensilica Xtensa L106 running at 80 MHz (or overclocked to 160 MHz). It has a 64 KB boot ROM, 64 KB instruction RAM and 96 KB data RAM. External flash memory can be accessed through SPI.

ESP8266 module is low cost standalone wireless transceiver that can be used for end-point IoT developments.

To communicate with the ESP8266 module, microcontroller needs to use set of AT commands. Microcontroller communicates with ESP8266-01 module using UART having specified Baud rate.

There are many third-party manufacturers that produce different modules based on this chip. So, the module comes with different pin availability options like,

- ESP-01 comes with 8 pins (2 GPIO pins) – PCB trace antenna. (shown in above figure)
 - ESP-02 comes with 8 pins, (3 GPIO pins) – U-FL antenna connector.
 - ESP-03 comes with 14 pins, (7 GPIO pins) – Ceramic antenna.
- ESP-04 comes with 14 pins, (7 GPIO pins) – No antenna

3.2 Sensors

3.2.1 Introduction

A sensor is a device that is able to detect changes in an environment. By itself, a sensor is useless, but when we use it in an electronic system, it plays a key role. A sensor is able to measure a physical phenomenon (like temperature, pressure, and so on) and transform it into an electric signal. These three features should be at the base of a good sensor:

- It should be sensitive to the phenomenon that it measures
- It should not be sensitive to other physical phenomena
- It should not modify the measured phenomenon during the measurement process

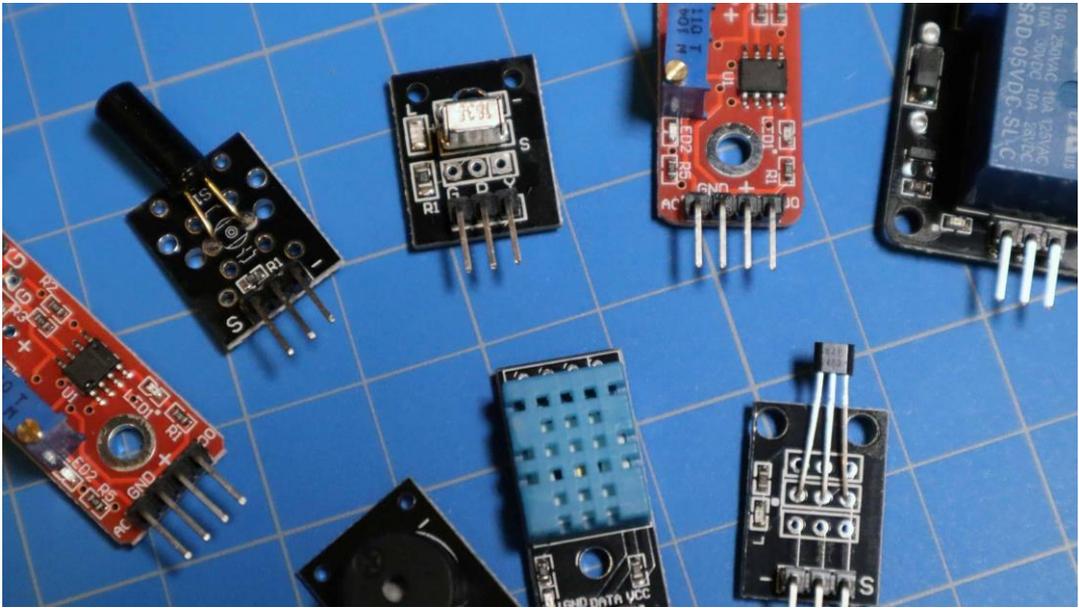


Figure 3.4: Arduino Sensors

There is a wide range of sensors we can exploit to measure almost all the physical properties around us. A few common sensors that are widely adopted in everyday life include thermometers, pressure sensors, light sensors, accelerometers, gyroscopes, motion sensors, gas sensors and many more. A sensor can be described using several properties, the most important being:

- **Range:** The maximum and minimum values of the phenomenon that the sensor can measure.
- **Sensitivity:** The minimum change of the measured parameter that causes a detectable change in output signal.
- **Resolution:** The minimum change in the phenomenon that the sensor can detect.

Sensors can be grouped using several criteria:

- **Passive or Active.** Passive sensors do not require an external power source to monitor an environment, while Active sensors require such a source in order to work.
- **Another classification is based on the method used to detect and measure the property (mechanical, chemical, etc.).**
- **Analog and Digital.** Analog sensors produce an analog, or continuous, signal while digital sensors produce a discrete signal.

There are other ways and methods to group sensors but the classifications shown above are the easiest.

The development of prototyping boards and the low price of sensors allow us easily use them in IoT projects. There are several prototyping boards on the market, suited for different projects depending on features and specifications. In this project, we will use the board: NodeMCU ESP8266 CH340.

It is important to note that every sensor has its own operating voltage range. This parameter is very important because the voltage supplied by the board must

not be higher than the maximum voltage allowed by the sensor. Therefore, it is important to read the sensor data sheet carefully before connecting it to the board to avoid damage. The same principle is valid for the output signal, which must be lower than the maximum voltage that the board can tolerate.

3.2.2 Some type of Sensors

A. Air-quality Sensor

Grove - Air quality sensor is an example of designed for air quality testing. It can respond to carbon monoxide, alcohol, acetone, thinner, formaldehyde, and other slightly toxic gases. Compatible with 5V and 3.3V power supply, it can work with Arduino and Raspberry Pi. With its long-term stability & low power consumption, it would be a perfect choice for air quality monitoring.

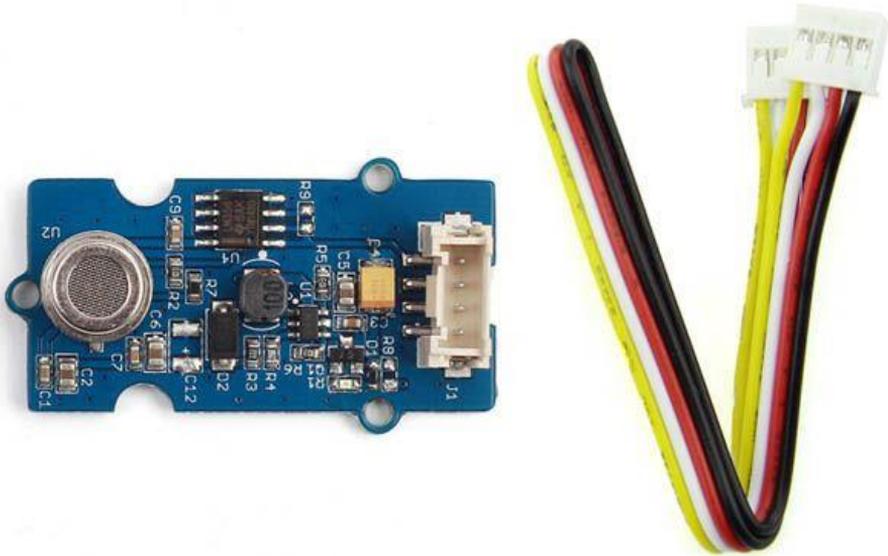


Figure 3.5: Grove – Air Quality Sensor v1.3

B. Dust sensor



Figure 3.6: Dust Sensor

Dust sensor is an optical air quality sensor it is designed to sense dust particles. An infrared emitting diode and a phototransistor are diagonally arranged into this device, to allow it to detect the reflected light of dust in air. It is especially effective in detecting very fine particles like cigarette smoke, and is commonly used in air purifier systems.

C. Water level sensor module

Judge a series of exposed parallel water sutures to measure the water gauge by the water level.



Figure 3.7: Water level sensor

Application rain detection, liquid leakage overflow detector.

Made of printed circuit board electrical components, the internal conductor is pure copper. It is durable and available for you to use.

Widely used in HVAC, dehumidifiers, test equipment, automobiles, automatic control, weather stations, humidity regulators and other related temperature detection and control.

This module can detect the volume of water droplets and the surrounding environment. It is accurate and reliable, and has good long-term stability. With fixing screw holes, easy to install and fix. Small size, low power consumption and fast response speed. High reliability, strong vibration resistance, good electromechanical performance, and long signal transmission distance.

D. Rain drop sensor

Raindrop sensor is basically a board on which nickel is coated in the form of lines. It works on the principal of resistance.

Rain Drop Sensor module allows to measure moisture via analog output pins and it provides a digital output when a threshold of moisture exceeds.



Figure 3.8: Rain drop sensor

The module is based on the LM393 op amp (Operational amplifier). It includes the electronics module and a printed circuit board that “collects” the

rain drops. As rain drops are collected on the circuit board, they create paths of parallel resistance that are measured via the op amp.

The sensor is a resistive dipole that shows less resistance when wet and more resistance when dry. When there is no rain drop on board it increases the Resistance so we get high voltage according to $V=IR$.

When rain drop present it reduces the resistance because water is a conductor of electricity and presence of water connects nickel lines in parallel so reduces resistance and reduces voltage drop across it.

E. Sound sensor

The sound sensor is one type of module used to notice the sound. Generally, this module is used to detect the intensity of sound. The applications of this module mainly include switch, security, as well as monitoring. The accuracy of this sensor can be changed for the ease of usage.



Figure 3.9: Sound sensor

This sensor employs a microphone to provide input to buffer, peak detector and an amplifier. This sensor notices a sound, & processes an o/p voltage signal to a microcontroller. After that, it executes required processing.

This sensor is capable to determine noise levels within DB's or decibels at 3 kHz 6 kHz frequencies approximately wherever the human ear is sensitive. In smartphones, there is an android application namely decibel meter used to measure the sound level.

According to The World Health Organization(WHO), sound become noise pollution when it gets above 65 dB. To be precise, noise becomes harmful when it exceeds 75 dB and is painful above 120 dB. We can use sound sensor to detect and notice whenever the sound excesses allowable level

F. Water quality sensor

The water quality sensor is a general term for multiple sensors that measure PH, residual chlorine, turbidity, suspended solids, COD, BOD,

— conductivity, and dissolved oxygen. Water quality does not refer to a specific day parameter; it contains multiple elements to measure water conditions.

Water quality is a general concept, and it covers a wide range. Therefore, building a complete water quality monitoring system is a complicated project.

G. Temperature-Humidity sensor

A temperature sensor is a device, typically, a thermocouple or RTD, that provides for temperature measurement through an electrical signal. A thermocouple (T/C) is made from two dissimilar metals that generate electrical voltage in direct proportion to changes in temperature. An RTD (Resistance Temperature Detector) is a variable resistor that will change its electrical resistance in direct proportion to changes in temperature in a precise, repeatable and nearly linear manner.

A humidity sensor is an electronic device that measures the humidity in its environment and converts its findings into a corresponding electrical signal. Humidity sensors vary widely in size and functionality; some humidity sensors can be found in handheld devices (such as smartphones), while others are integrated into larger embedded systems (such as air quality monitoring systems). Humidity sensors are commonly used in the meteorology, medical, automobile, HVAC and manufacturing industries.

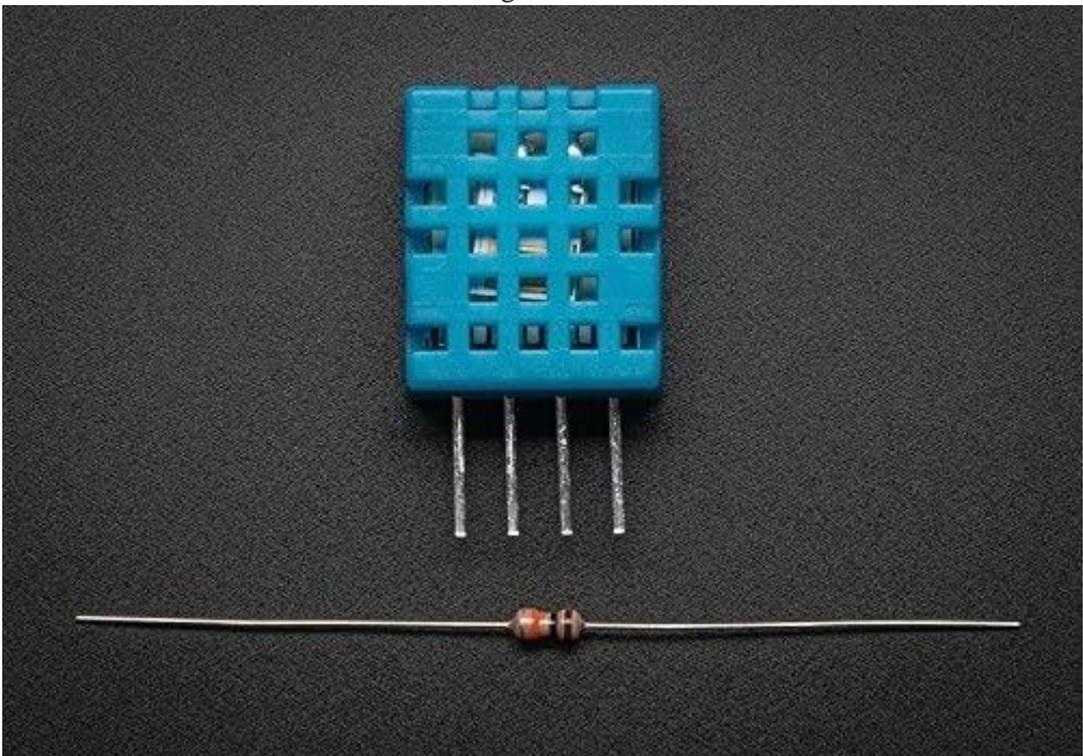


Figure 3.10: DHT11 temperature-humidity sensor

The DHT11 is a basic, ultra low-cost combined digital temperature and humidity sensor. It uses a capacitive humidity sensor and a thermistor to measure the surrounding air, and spits out a digital signal on the data pin (no analog input

pins needed). It's fairly simple to use, but requires careful timing to grab data. The only real downside of this sensor is you can only get new data from it once every 2 seconds, so when using library, sensor readings can be up to 2 seconds old.

3.3 Blynk

3.3.1 What is Blynk?

Blynk is a full suite of software required to prototype, deploy, and remotely manage connected electronic devices at any scale: from personal IoT projects to millions of commercial connected products. Blynk come with iOS and Android apps to control Arduino, ESP8266, Raspberry Pi and the likes over the Internet.

Anyone can connect their hardware to the cloud and build a no-code iOS, Android, and web applications to analyze real-time and historical data coming from devices, control them remotely from anywhere in the world, receive important notifications, and much more...

Blynk is a multi-tenant solution. You can configure how users get access to the data by setting roles and configuring permissions.

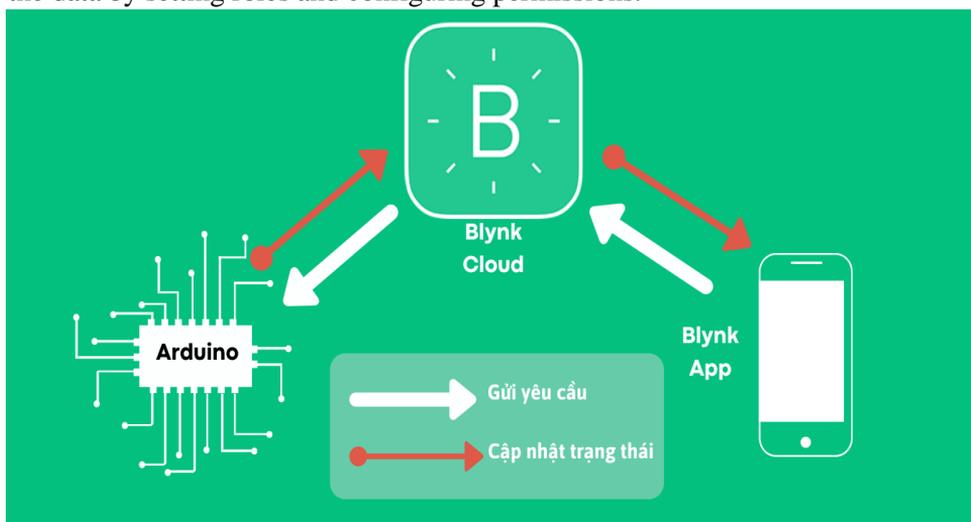


Figure 3.11: Blynk

3.3.2 Blynk Server, Blynk App and Blynk Libraries

Blynk Server is an Open-Source Netty based Java server, responsible for forwarding messages between Blynk mobile application and various microcontroller boards and SBCs (i.e. Arduino, Raspberry Pi. etc).

Before July 2021, Blynk only support develop and control device on iOS/Android through app name Blynk(Legacy). After that they released new app call Blynk Iot which can help develop and control device on both web and mobile with a lot more improvement compares to the previous app. You can no longer create new account on old Blynk app, but for who have created account before July, they can still use old Blynk app although it's not supported by developer

anymore. Noted that the old and new Blynk are difference and not relevant with each other, so for who want to change to new Blynk they have to create new account.

To setup Blynk to control your device, we'll also need to install the Blynk Arduino Library, which helps generate the firmware running on your microcontroller chips.

Below is brief look at Blynk's Web and Mobile Dashboard:

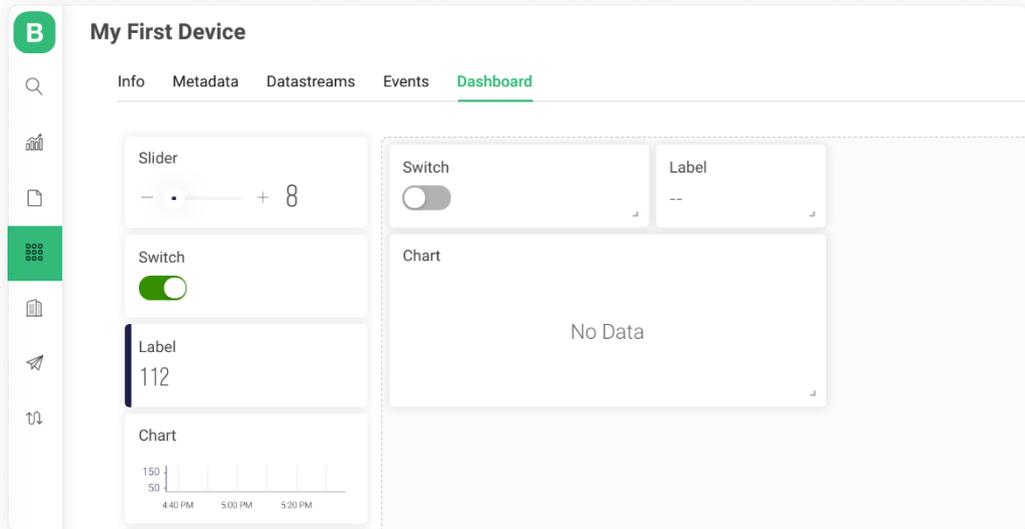


Figure 3.12: Blynk Web Dashboard

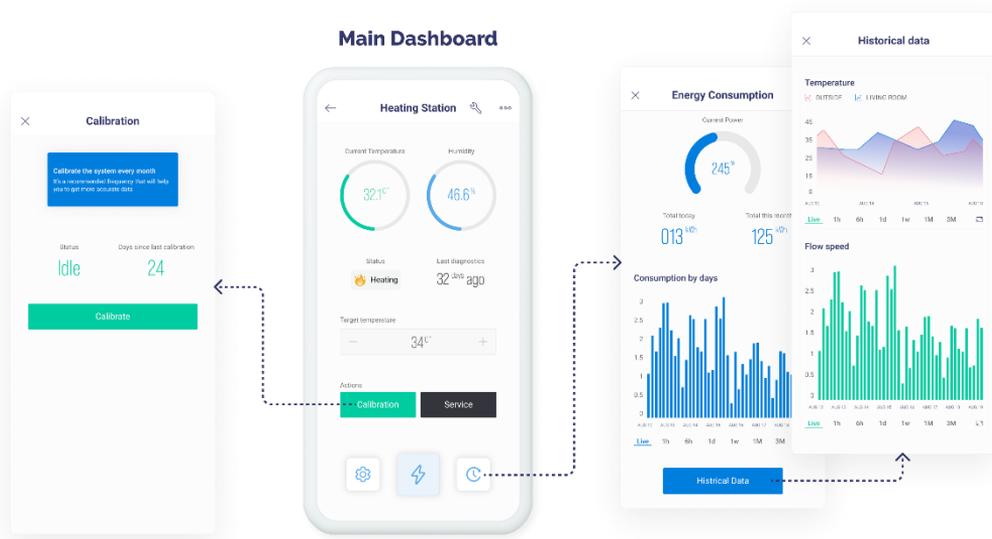


Figure 3.13: Blynk Mobile Dashboard

3.3.3 Alternatives to Blynk

Besides Blynk, there are also a lot of apps which have similar uses and can be Blynk's competitors. One of important factors to consider when researching alternatives to Blynk IoT is include reliability and ease of use, also the size and complexity of project to search for most suitable platform. Here is some example platform that similar to Blynk:

- Particle

Particle is an enterprise IoT (Internet of Things) platform that provides everything you need to power your IoT product, from device to cloud. Particle provides developers with the tools they need to prototype IoT solutions quickly, as well as a reliable and secure platform for enterprises to scale their IoT solutions. As the most widely-used IoT device platform, Particle is used by over 200,000 engineers and by 8,500 companies in more than 170 countries.

- AWS IoT

AWS IoT is a cloud-based platform that connects devices easily and securely interact with cloud applications and other devices, it supports devices and messages, and can process and route those messages to AWS endpoints and to other devices reliably and securely.

- Arduino IoT Cloud

Arduino IoT Cloud is an application that helps makers build connected objects in a quick, easy and secure way. You can connect multiple devices to each other and allow them to exchange real-time data. You can also monitor them from anywhere using a simple user interface.

- IoT-Ignite

IoT-Ignite is a Platform as a Service (PaaS) distinctively designed for realization of Internet of Things. Behind the design, there are modern best practices and a unique architecture. The IoT-Ignite platform enables users to build innovative IoT products and solutions. It provides a secure, reliable gateway connection between your devices and the web and allows you to manage your devices, data flow, data streaming and rule definitions. It's flexible, user friendly; it reduces your technical risk and shortens time to market.

- Fogwing IIoT Platform

Fogwing is the comprehensive Industrial IoT Platform available as a service on subscription basis. Fogwing is a no-code platform that provides features to connect any devices through WiFi, Cellular Wireless (3/4/5g) and LoRaWAN network, data streaming, data rules routing, data alerts, command center, data storage, integration and analytics. With all capabilities, Fogwing can helps to achieve industrial automation. Fogwing also provides IoT solutions as Apps for easy adoption.

CHAPTER 4

Implement Phase

4.1 Architecture

Below is the structure ideal of the project:

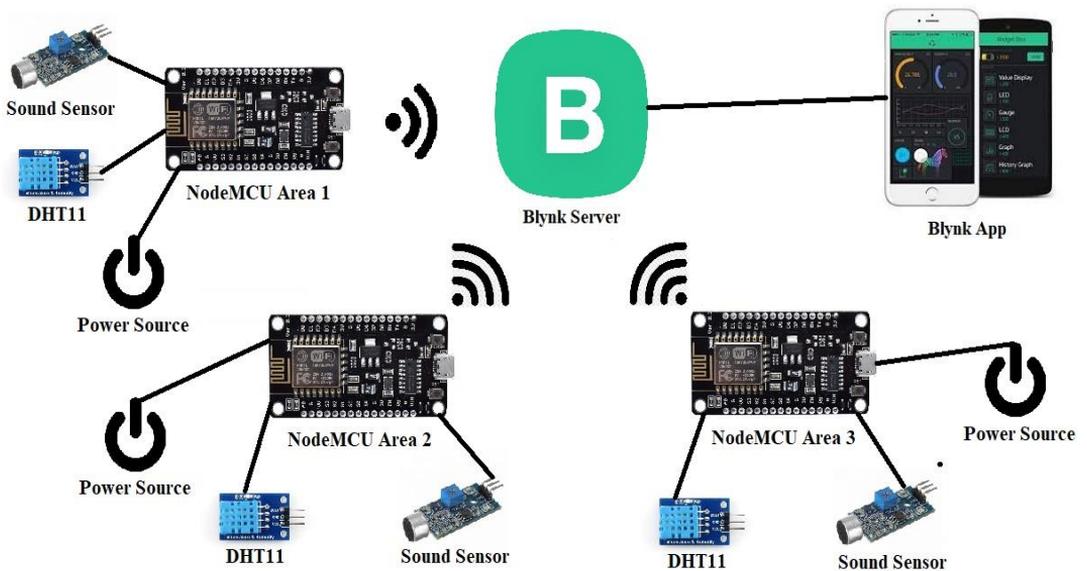


Figure 4.1: Project Architecture

4.2 Implement

4.2.1 Setup devices and setup Blynk

In this section, we giving the details step by step of setting up all devices, design widgets on Blynk Web Dashboard and Mobile Dashboard, and connect between devices and Blynk to be able to collect data. With Blynk, on one account we can have a lot of devices at many location, collecting data from different places with ease, and each device can collect different type of data that we need. Since the base step of setting devices are the same, this section will only demonstrate 1 device

In this project, to demonstrate setting up 1 device we will use:

- 1 KIT WiFi NodeMCU ESP8266
- 1 Temperature & Humidity DHT11 Sensor
- 1 Sound Sensor
- Wires
- Blynk IoT app

With this 2 sensors, we can collect and monitor data of temperature, humidity and noise of environment around the device through Blynk.

1. Build the circuit.

Below is the diagram of device's circuit with NodeMCU:

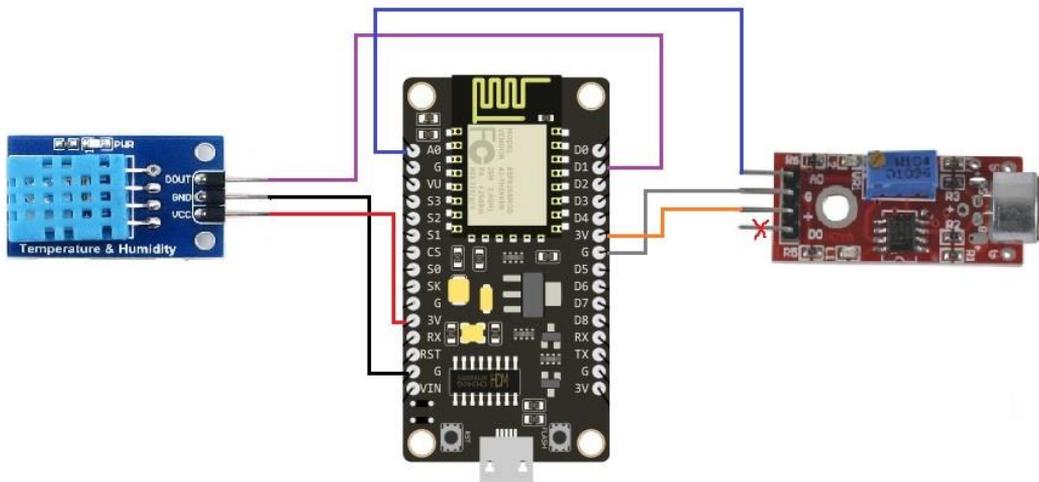


Figure 4.2: Circuit

For DHT11, use wires connect VCC pin to 3V pin, GND pin to Ground pin and OUT pin to D1 pin of NodeMCU.

For Sound sensor, use wires connect + pin to 3V pin, G to Ground and A0 pin to A0 pin of NodeMCU

2. Setting up Blynk

Access to Blynk.cloud, sign in and create new template compatible with ESP8266. In the Datastreams tab, add datastreams to configure how data is sent

between the device and Blynk. In this case, we add 4 datastreams include Led, Temperature, Humidity and Noise. In each datastream settings exclude Led, choose Expose to Automation for setting up notification when data collected surpass the threshold.

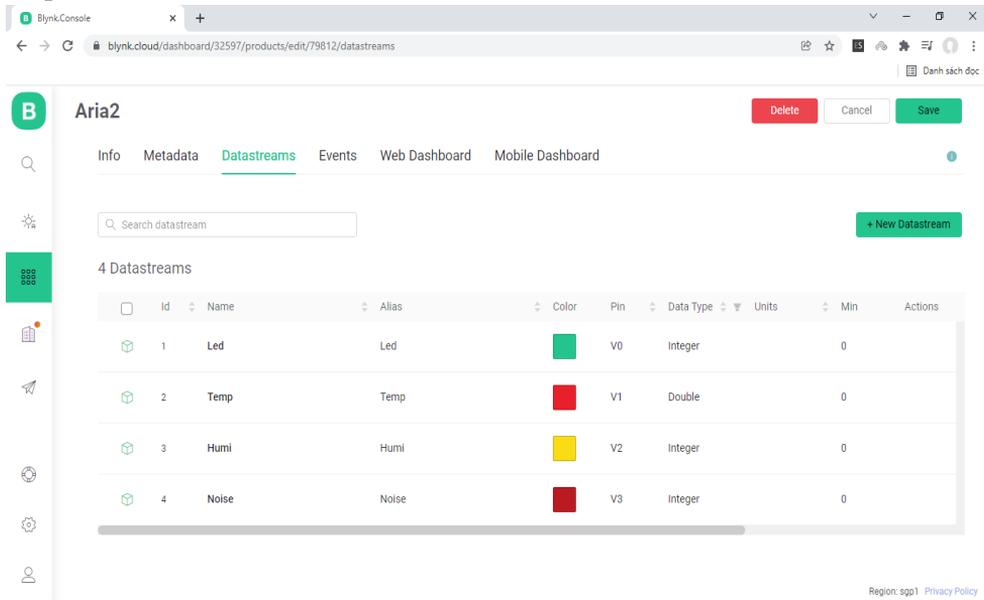


Figure 4.3: Datastreams configuration

Next, on the Web Dashboard tab, add widgets and setting each one to match the pin of each datastream we just created. In our project, 3 Gauge widgets indicate to each data pin: Temperature, Humidity and Noise. Click save to finish the template.

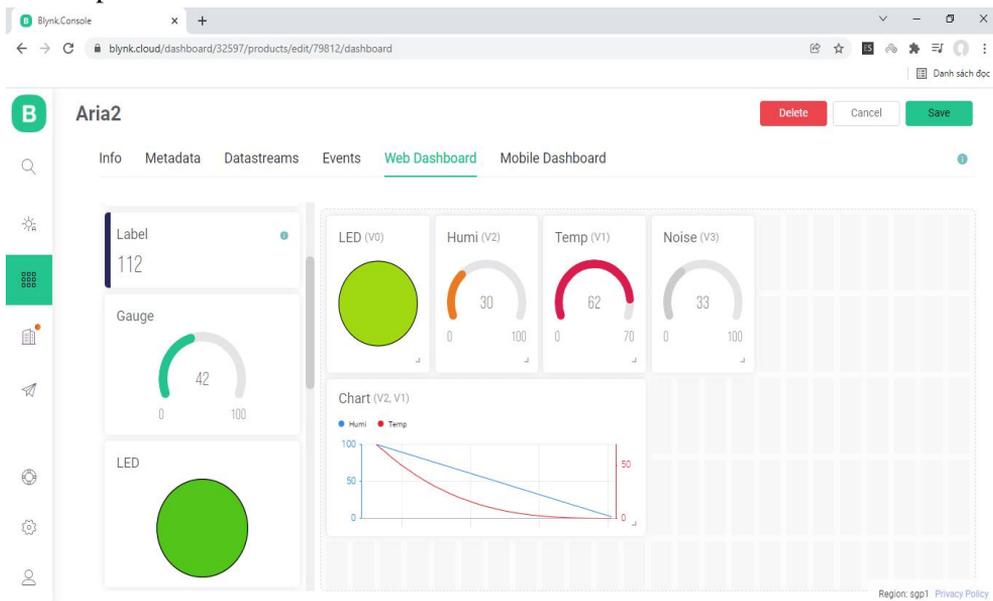


Figure 4.4: Web Dashboard Setup

After finish setting up template on Blynk web, open Blynk IoT app on mobile, sign in, on the top right of the screen click on wrench icon to enter developer

mode. On the screen choose the template that we setup on web before to begin design and adding widgets on mobile dashboard. The operation on mobile are quite the same as operation on the web.

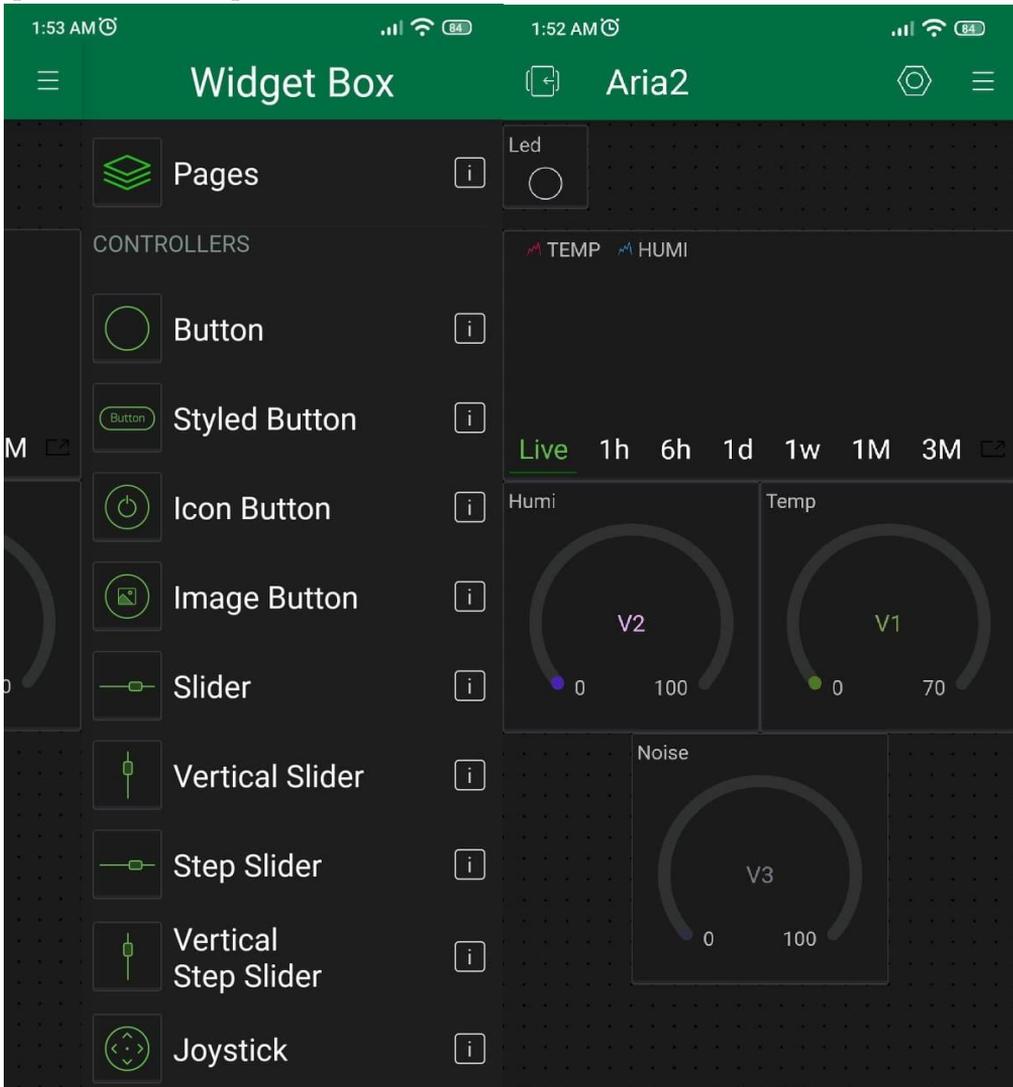


Figure 4.5: Mobile Dashboard Setup

4.2.2 Connect device to Blynk

Open Arduino IDE, to be able to setting up the connection to Blynk, first you need to download the Blynk libraries, then on examples of the file section, choose the Edgent_ESP8266 template in Blynk. In each template on Blynk, there are Firmware configuration section that show template id and device name, we must define that on the main tab of coding to ensure the device connect to the right template of Blynk.

FIRMWARE CONFIGURATION

```
#define BLYNK_TEMPLATE_ID "TMPLDi0kohQ-"  
#define BLYNK_DEVICE_NAME "Aria2"
```

Template ID and Device Name should be included at the top of your main firmware

Figure 4.6: Blynk Template ID & Device Name

After done with the coding phase and upload it to the NodeMCU, on the NodeMCU there are FLASH button, hold it for 10 seconds for NodeMCU return to configuration mode. Then in the main screen of the Blynk app, on the top right of screen click on menu icon, and choose “Add new device” to begin connect the device to Blynk app, follow the instructure and the device is finally connected to Blynk. The device and the phone using Blynk must connect to the same WiFi in order to be able to connect to each other.

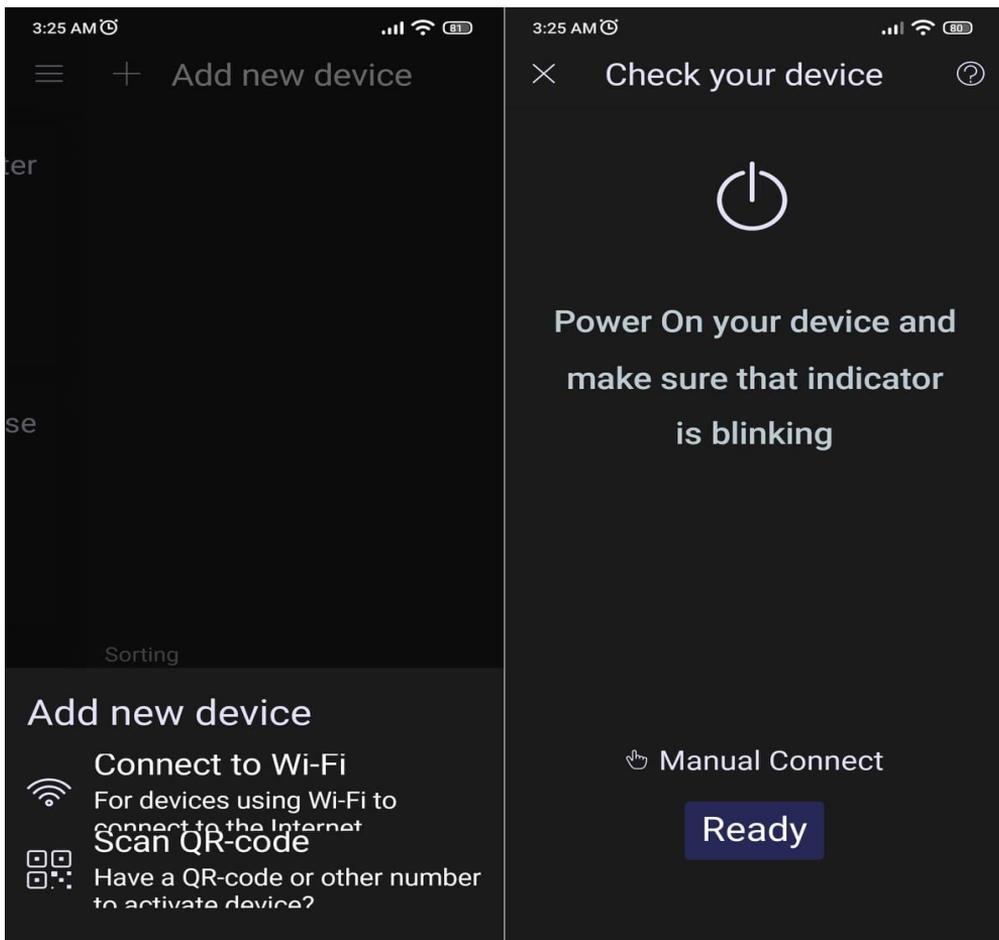


Figure 4.7: “Adding New Device” Screen

4.2.3 Code

Below is the source code of the project

```
// Capstone project
#define BLYNK_TEMPLATE_ID "TMPLLv5bkCcj"
#define BLYNK_DEVICE_NAME "Area 4"

#define BLYNK_FIRMWARE_VERSION "0.1.0"
#define BLYNK_PRINT Serial
// #define BLYNK_DEBUG

#define APP_DEBUG
#define USE_NODE_MCU_BOARD
#include "BlynkEdgent.h"
#include "DHT.h"
#define DHTPIN 5
#define DHTTYPE DHT11
DHT dht(DHTPIN, DHTTYPE);
unsigned long times=millis();
int noise=0, count=0;
boolean runMode=0, ledAlarmStatus=0, canhbaoState=0; // lưu trạng thái cảnh báo
quá ồn hoặc quá nhiệt
int thresholdnoise=70, thresholdtemp=25;

WidgetLED ledconnect(V0);
WidgetLED ledAlarm(V4);
BlynkTimer timer;

// V0 LED Widget is blinking
void blinkLedWidget(){
  if (ledconnect.getValue()) ledconnect.off();
  else ledconnect.on();
}

void setup(){
  Serial.begin(115200);
  delay(100);
  BlynkEdgent.begin();
  dht.begin();
  timer.setInterval(1000L, blinkLedWidget);
  //timerID1 = timer.setInterval(1000L, handleTimerID1);
  // timerID2 = timer.setInterval(10000L, handleTimerID2);
}

void loop() {
  BlynkEdgent.run();
  timer.run();
  if(millis()-times>100){
    noise=analogRead(A0)/4;
    Blynk.virtualWrite(V3, noise);
    if(noise>thresholdnoise) {canhbaoState=1; Serial.println("Không gian làm
```

```

việc đang quá ồn!"));}

// Reading temperature or humidity
count++;
if(count==25)
{
    count=0;
    float t = dht.readTemperature();
    if(t>thresholdtemp) {canhbaoState=1; Serial.println("Nhiệt độ đang quá
ngưỡng!");}

    float h = dht.readHumidity();
    if (isnan(h) || isnan(t)) {
        Serial.println(F("Failed to read from DHT sensor!"));
        return;
    }
    Blynk.virtualWrite(V1,t);
    Blynk.virtualWrite(V2,h);
    Serial.print(F("Temperature: ")); Serial.print(t); Serial.print(F("°C -
"));
    Serial.print(F("Humidity: ")); Serial.print(h); Serial.print(F("% - "));
    Serial.print(F("Noise: ")); Serial.print(noise); Serial.println("% max.");
}
BlinkLedAlarm(); // Nhấp nháy đèn Alarm
times = millis();
}
}

void BlinkLedAlarm()
{
    if(canhbaoState==1)
        if(ledAlarmStatus==0)
        {
            ledAlarmStatus=1;
            ledAlarm.on();
        }
        else
        {
            ledAlarmStatus=0;
            ledAlarm.off();
        }
        else ledAlarm.off();
    canhbaoState=0;
}

BLYNK_WRITE(V5) {
    runMode = param.asInt();
    if(runMode==0)canhbaoState=0;
}

```

CHAPTER 5

Project Result

5.1 Packaging Project

Below shows the finished product when we packaging the project:

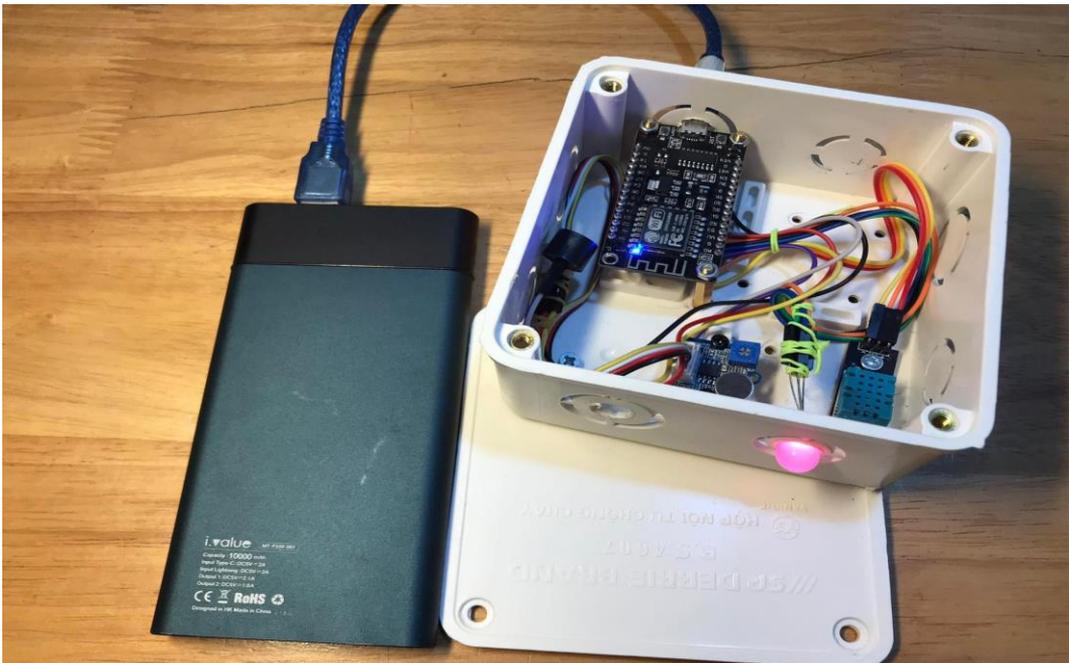


Figure 5.1: Project device packaging

5.2 Practical Result

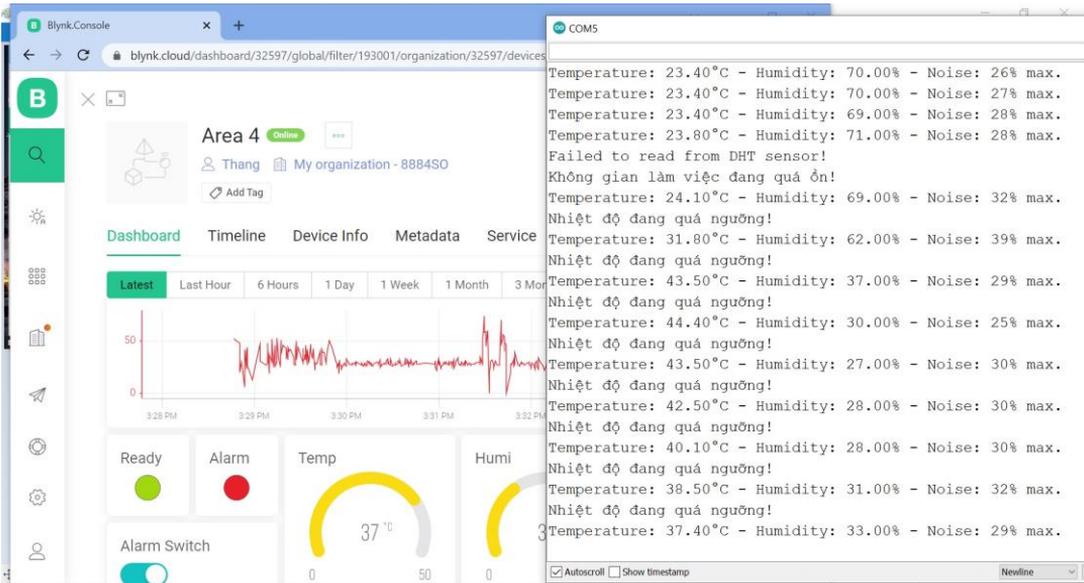


Figure 5.2a: Serial monitor

In Figure 5.2a the data that the sensors of the device capture and show it on serial monitor, when the data collected exceed the threshold, there will be a warning message appear on serial monitor as show on picture above.

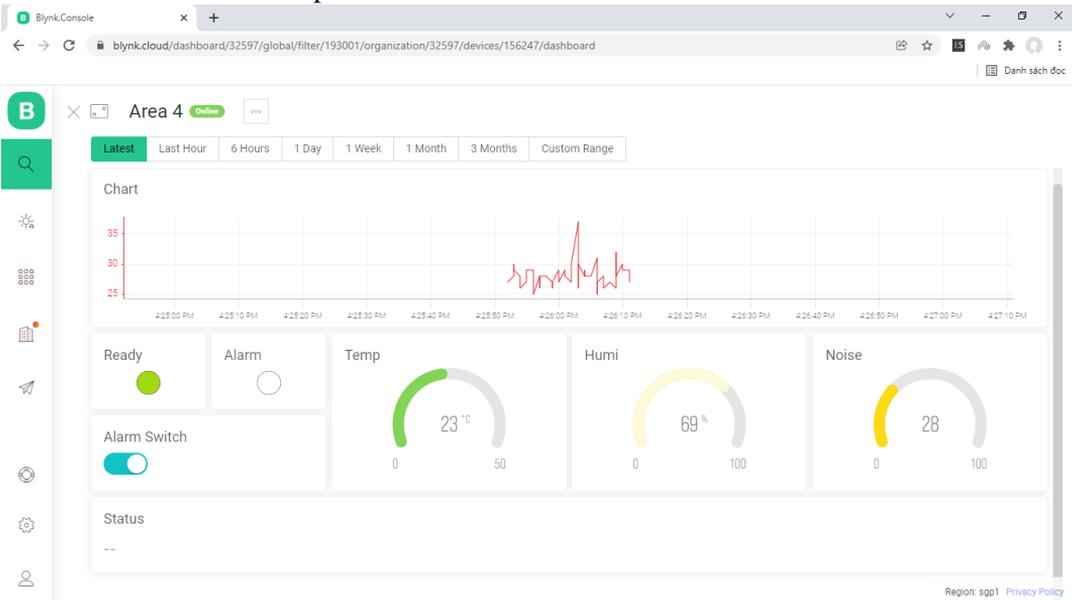


Figure 5.2b: Web Interface

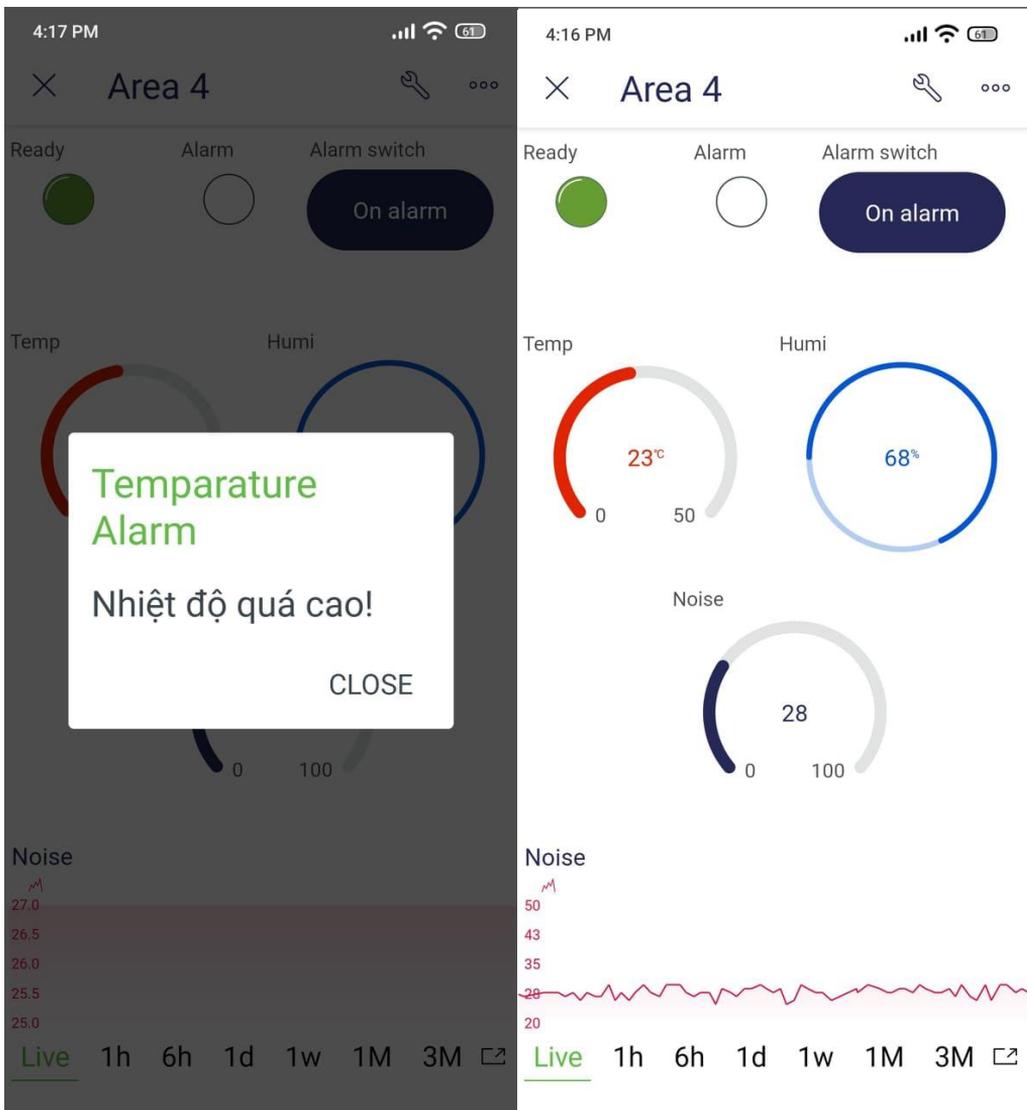


Figure 5.3: Mobile interface and Alarm notification

In figure 5.3. we have the mobile interface of Blynk and the data show in the form of gauges, also with the noise data, we use chart to show the change of the data each second, we can also change the time of data recorded below the chart for more information. When the data collected exceed the threshold, in this case is temperature, app will send a notification as show on figure to let us know.

CHAPTER 6

Conclusion and Future Work

6.1 Conclusion

In this project, we studied about NodeMCU and Blynk to provide an environmental data collection and monitoring system that cheaper and more individual friendly to setup and use. The coding needed for this project is a lot less complicated since a lot of function is included in Blynk which help simple the work needed to be done by a lot

Based on our methods, the data we collected can be flexible as we can choose to export the data for other use, since the data that Blynk server provided can save up to 3 months of historical data storage. With the help of Blynk platform, we can set up a lot of function for the device that normally need coding to be able to work properly. That is one advantage of using existed platform to make monitoring device.

Also this project is only suit for research and study. Because in reality, to be able to make a completed and fully functional environmental monitoring system, it is required a lot of different elements and human power, hence why this project cannot be use practically.

6.2 Future Plan

Although the project now is finished, it still need a lot more to be able to call good. We can add a lot more type of sensors for device to be able to collect more type of data. New function also can be added in to make project more completed.

As mention above, although Blynk provided a lot of tools that make it easier to create device designate to monitoring data, it is also a disadvantage since many things got limited like design of the widget, some other function cannot be change.

But to be able to created own web, app server and design widget, it's required a lot of time and effort. Therefore, we hope in the future we can do this without the help of existed platform.

And as said in above section, because this project is not practically for use, we hope to research and practice more to finally make this project be practical and use commonly.