**TRƯỜNG ĐẠI HỌC GIAO THÔNG VẬN TẢI**



**BÁO CÁO TỔNG KẾT**

**Hệ thống giám sát thời tiết ESP32**

**Sinh viên thực hiện**

**Nguyễn Tiến Hải – 201403932**

Nguyễn Văn Dũng **–** 201403900

Nguyễn Mạnh Dương **–** 201405992

Nguyễn Hữu Ngọc **–** 201404041

Trần Đức Hoàn **–** 201403956

Lớp: KTĐT-THCN2-K61 Khoa: Điện-Điện tử

Giảng viên: TS.Ngô Thanh Bình

**MỤC LỤC**

[**Danh mục bảng biểu** 2](#_Toc150180863)

[**Danh mục video test linh kiện** 2](#_Toc150180864)

[**PHẦN 1. TỔNG QUAN** 3](#_Toc150180865)

[**1.** **Đặt vấn đề.** 3](#_Toc150180866)

[**2.** **Lý do chọn đề tài.** 3](#_Toc150180867)

[**3.** **Mục tiêu nghiên cứu.** 3](#_Toc150180868)

[**4.** **Phần mềm sử dụng.** 4](#_Toc150180869)

[**5.** **Đối tượng nghiên cứu.** 4](#_Toc150180870)

[**6.** **Phạm vi nghiên cứu.** 4](#_Toc150180871)

[**7.** **Phân công nhiệm vụ các thành viên trong nhóm.** 5](#_Toc150180872)

[**PHẦN 2. GIỚI THIỆU LINH KIỆN** 5](#_Toc150180873)

[**1.** **Linh kiện sử dụng.** 5](#_Toc150180874)

[**2.** **Test linh kiện.** 10](#_Toc150180875)

[**2.1.** **ESP kết nối với màn hình LCD.** 10](#_Toc150180876)

[**2.2.** **ESP kết nối với cảm biến DHT22.** 12](#_Toc150180877)

[**2.3.** **ESP kết nối với cảm biến LDR.** 15](#_Toc150180878)

[**2.4.** **ESP kết nối với cảm biến BMP180.** 17](#_Toc150180879)

[**2.5.** **ESP kết nối với cảm biến mưa.** 21](#_Toc150180880)

[**3.** **Sản phẩm thực tế.** 24](#_Toc150180881)

[**PHẦN 3. KẾT LUẬN VÀ KIẾN NGHỊ** 28](#_Toc150180882)

[**1.** **Kết luận** 28](#_Toc150180883)

[**2.** **Kiến nghị.** 29](#_Toc150180884)

# **Danh mục bảng biểu**

[Hình 1: Bảng ESP 6](#_Toc150178389)

[Hình 2: Cảm biến LDR 7](#_Toc150178390)

[Hình 3: Cảm biến BMP180 8](#_Toc150178391)

[Hình 4: Cảm biến DHT22 8](#_Toc150178392)

[Hình 5: Cảm biến mưa 9](#_Toc150178393)

[Hình 6: Màn hình LCD và mô-đun I2C 9](#_Toc150178394)

[Hình 7: Sơ đồ mô phỏng ESP kết nối với màn hình LCD 10](#_Toc150178395)

[Hình 8: Sơ đồ thuật toán ESP kết nối với màn hình LCD 11](#_Toc150178396)

[Hình 9: Kết quả mô phỏng ESP kết nối với màn hình LCD 12](#_Toc150178397)

[Hình 10: Sơ đồ mô phỏng ESP kết nối với cảm biến DHT22 12](#_Toc150178398)

[Hình 11: Sơ đồ thuật toán ESP kết nối với cảm biến DHT22 13](#_Toc150178399)

[Hình 12: Sơ đồ mô phỏng ESP kết nối với cảm biến LDR 15](#_Toc150178400)

[Hình 13: Sơ đồ thuật toán ESP kết nối với cảm biến LDR 16](#_Toc150178401)

[Hình 14: Sơ đồ mô phỏng ESP kết nối với cảm biến BMP180 18](#_Toc150178402)

[Hình 15: Sơ đồ thuật toán ESP kết nối với cảm biến BMP180 19](#_Toc150178403)

[Hình 16: Sơ đồ mô phỏng ESP kết nối với cảm biến mưa 21](#_Toc150178404)

[Hình 17: Sơ đồ thuật toán ESP kết nối với cảm biến mưa 22](#_Toc150178405)

[Hình 18: Sơ đồ mô phỏng hệ thống giám sát thời tiết ESP32 24](#_Toc150178406)

[Hình 19: Sơ đồ thuật toán giám sát thời tiết ESP32 25](#_Toc150178407)

# **Danh mục video test linh kiện**

Video 1: Kết quả mô phỏng ESP kết nối với cảm biến DHT22 .........................................14

Video 2: Kết quả mô phỏng ESP kết nối với cảm biến LDR 17

Video 3: Kết quả mô phỏng ESP kết nối với cảm biến BMP180 21

Video 4: Kết quả mô phỏng ESP kết nối với cảm biến mưa 23

Video 5: Kết quả mô phỏng hệ thống giám sát thời tiết ESP32 28

# **PHẦN 1. TỔNG QUAN**

## **Đặt vấn đề.**

Sự phát triển của hệ thống giám sát thời tiết là một chủ đề quan trọng trong linh vực dự báo thời tiết và quản lý thiên tai. Sự phát triển của công nghệ đã cung cấp các thiết bị và công cụ tiên tiến hơn để thu thập dữ liệu thời tiết. Các vệ tinh, radar, cảm biến khí tượng ngày càng chính xác và giúp nâng cao chất lượng thời tiết. Sử dụng khai thác dữ liệu lớn và học máy đã cải thiện khả năng dự báo thời tiết. Phân tích dữ liệu từ nhiều nguồn khác nhau giúp tạo ra dự báo thời tiết chính xác hơn và kịp thời. Hệ thống giám sát thời tiết ngày càng cải thiện khả năng cảnh báo sớm về các hiện tượng thiên tai như bão, lở đất, lụt lớn, cháy rừng. Điều này giúp cảnh báo dân cư và cơ quan chính phủ để chuẩn bị và phản ứng kịp thời.

## **Lý do chọn đề tài.**

Hiện nay thời tiết có ảnh hưởng đến cuộc sống hàng ngày của mọi người và đóng vai trò quan trọng trong nhiều khía cạnh của đời sống con người, từ nông nghiệp đến giao thông và quản lý thiên tai. Sự phát triển của hệ thống giám sát thời tiết có ảnh hưởng trực tiếp đến khả năng dự báo thời tiết và cảnh báo sớm về thiên tai, giúp bảo vệ tính mạng và tài sản của người dân. Vì vậy nhóm nghiên cứu chúng em đã lựa chọn đề tài:

**“Hệ thống giám sát thời tiết**”

## **Mục tiêu nghiên cứu.**

* Xây dựng hệ thống giám sát thời tiết sử dụng ESP32 và các cảm biến.
* Đo và ghi lại nhiệt độ, độ ẩm, áp suất không khí, độ sáng và lượng mưa.
* Hiển thị thông tin thu thập được trên màn hình LCD.
* Cung cấp khả năng theo dõi thời tiết từ xa qua ứng dụng Blynk.

## **Phần mềm sử dụng.**

**Arduino IDE** (Integrated Development Environment) - thực hiện việc viết chương trình - là một phần mềm mã nguồn mở được sử dụng để lập trình và nạp code cho các bo mạch Arduino. Nó cung cấp một môi trường phát triển đơn giản và dễ sử dụng cho việc viết và tải lên code vào bo mạch Arduino.



Dưới đây là một số điểm nổi bật về Arduino IDE:

* Giao diện đồ họa thân thiện: Arduino IDE có giao diện đồ họa đơn giản và thân thiện, giúp người dùng dễ dàng thao tác và lập trình cho bo mạch Arduino.
* Editor code: Arduino IDE đi kèm với một trình soạn thảo code tích hợp, cho phép người dùng viết mã Arduino dễ dàng. Nó hỗ trợ các tính năng như kiểm tra cú pháp, tô màu cú pháp, và gợi ý từ khóa.
* Thư viện và ví dụ: Arduino IDE cung cấp một bộ thư viện phong phú cho các chức năng và module phổ biến, như điều khiển động cơ, giao tiếp không dây, đọc cảm biến, và nhiều hơn nữa. Ngoài ra, nó cũng cung cấp các ví dụ code để người dùng có thể học và áp dụng vào dự án của mình.
* Nạp code dễ dàng: Arduino IDE cho phép người dùng nạp code từ máy tính vào bo mạch Arduino thông qua cổng kết nối USB. Người dùng có thể chọn bo mạch Arduino mục tiêu và cổng COM tương ứng để nạp code.
* Hỗ trợ đa nền tảng: Arduino IDE có sẵn cho nhiều hệ điều hành như Windows, macOS và Linux, giúp người dùng có thể sử dụng trên nhiều nền tảng khác nhau.

## **Đối tượng nghiên cứu.**

* Các cảm biến vật lý ghi nhận được các thông số môi trường.
* Hệ thống tiếp nhận và xử lý dữ liệu thu được.

## **Phạm vi nghiên cứu.**

Tập trung vào cách ghi nhận thông tin môi trường xung quanh thiết bị từ các cảm biến đọc và truyền tải thông tin dữ liệu về môi trường từ các cảm biến và truyền tải thông tin qua mạng Lan, mạng Internet và truyền tải dữ liệu lên các hệ thống điện toán đám mây để giám sát dữ liệu và phân tích theo thời gian thực tại một trạm quan sát.

## **Phân công nhiệm vụ các thành viên trong nhóm.**

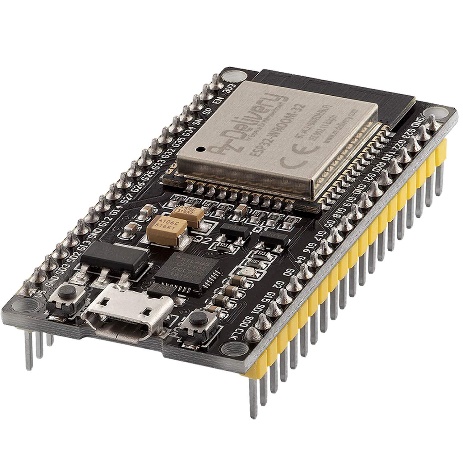
|  |  |
| --- | --- |
| Tên thành viên | Nhiệm vụ |
| Nguyễn Tiến Hải | Code, hàn và lắp ráp linh kiện |
| Nguyễn Văn Dũng | Làm word |
| Trần Đức Hoàn | Làm powerpoint |
| Nguyễn Mạnh Dương | Sơ đồ thuật toán |
| Nguyễn Hữu Ngọc | Sơ đồ thuật toán |

# **PHẦN 2. GIỚI THIỆU LINH KIỆN**

## **Linh kiện sử dụng.**

* **Bảng ESP32:**

Là một series các vi điều khiển trên một vi mạch giá rẻ, năng lượng thấp có hỗ trợ WiFi và dual-mode Bluetooth. Dòng ESP32 sử dụng bộ vi xử lý Tensilica Xtensa LX6 ở cả hai biến thể lõi kép và lõi đơn, và bao gồm các công tắc antenna tích hợp, RF balun, bộ khuếch đại công suất, bộ khuếch đại thu nhiễu thấp, bộ lọc và module quản lý năng lượng. ESP32 được chế tạo và phát triển bởi Espressif Systems, một công ty Trung Quốc có trụ sở tại Thượng Hải, và được sản xuất bởi TSMC bằng cách sử dụng công nghệ 40 nm. ESP32 là sản phẩm kế thừa từ vi điều khiển ESP8266.



Hình : Bảng ESP

Đặc điểm:

* Bộ xử lý:

o CPU: Bộ vi xử lý Xtensa lõi kép (hoặc lõi đơn) 32-bit LX6, hoạt động ở tần số 240 MHz (160 MHz cho ESP32-S0WD và ESP32-U4WDH) và hoạt động ở tối đa 600 MIPS (200 MIPS với ESP32-S0WD/ESP32-U4WDH)

o Bộ đồng xử lý (co-processor) công suất cực thấp (Ultra low power, viết tắt: ULP) hỗ trợ việc đọc ADC và các ngoại vi khi bộ xử lý chính (main processor) vào chế độ deep sleep.

* Hệ thống xung nhịp: CPU Clock, RTC Clock và Audio PLL Clock.
* Bộ nhớ nội:

o 448 KB bộ nhớ ROM cho việc booting và các tính năng lõi.

o 520 KB bộ nhớ SRAM trên chip cho dữ liệu và tập lệnh.

* Kết nối không dây:

o Wi-Fi: 802.11 b/g/n.

o Bluetooth: v4.2 BR/EDR và BLE (chia sẻ sóng vô tuyến với Wi-Fi)

* 34 GPIO pad vật lý với các ngoại vi:

o ADC SAR 12 bit, 18 kênh.

o DAC 2 × 8-bit.

o 10 cảm biến cảm ứng (touch sensor) (GPIO cảm ứng điện dung).

o 3 SPI (SPI, HSPI và VSPI) hoạt động ở cả 2 chế độ master/slave. Module ESP32 hỗ trợ 4 ngoại vi SPI với SPI0 và SPI1 kết nối đến bộ nhớ flash của ESP32 còn SPI2 và SPI3 tương ứng với HSPI và VSPI.[10] Các GPIO đều có thể được dùng để triển khai HSPI và VSPI.[1

o 2 I²S

o 2 I²C, hoạt động được ở cả chế độ master và slave, với chế độ Standard mode (100 Kbit/s) và Fast mode (400 Kbit/s). Hỗ trợ 2 chế độ định địa chỉ là 7-bit và 10-bit. Các GPIO đều có thể được dùng để triển khai I²C.

o 3 UART (UART0, UART1, UART2) với tốc độ lên đến 5 Mbps

o SD/SDIO/CE-ATA/MMC/eMMC host controller

o SDIO/SPI slave controller

o Ethernet MAC interface cho DMA và IEEE 1588 Precision Time Protocol (tạm dịch: Giao thức thời gian chính xác IEEE 1588)

o CAN bus 2.0

o Bộ điều khiển hồng ngoại từ xa (TX/RX, lên đến 8 kênh)

o PWM cho điều khiển động cơ

o LED PWM (lên đến 16 kênh)

o Cảm biến hiệu ứng Hall

o Bộ tiền khuếch đại analog công suất cực thấp (Ultra low power analog pre-amplifier)

* Bảo mật:

o Hỗ trợ tất cả các tính năng bảo mật chuẩn IEEE 802.11, bao gồm WFA, WPA/WPA2 và WAPI.

o Secure boot (tạm dịch: khởi động an toàn)

o Mã hóa flash

o 1024-bit OTP, lên đến 768-bit cho khách hàng

o Tăng tốc mã hóa phần cứng: AES, SHA-2, RSA, elliptic curve cryptography (ECC, tạm dịch: mật mã đường cong ellip), random number generator (viết tắt: RNG, tạm dịch: trình tạo số ngẫu nhiên)

* Quản lý năng lượng:

o Hỗ trợ 5 chế độ hoạt động với mức tiêu thụ năng lượng khác nhau: Active, Modem¬-sleep, Light-¬sleep, Deep-¬sleep và Hibernation

o Bộ ổn áp nội với điện áp rơi thấp (internal low-dropout regulator)

o Individual power domain (tạm dịch: Miền nguồn riêng) cho RTC

o Trở lại hoạt động từ ngắt GPIO, timer, đo ADC, ngắt với cảm ứng điện dung

* **Cảm biến LDR:**

A blue circuit board with black text

Description automatically generated

Hình : Cảm biến LDR

Điện áp hoạt động: 5VDC

Chuẩn giao tiếp: Analog

Điện áp giao tiếp: 0~3.3VDC

Đo cường độ ánh sáng bằng quang trở (LDR-Light Dependent Resistor)

Sử dụng trực tiếp an toàn với các board mạch giao tiếp ở cả hai mức điện áp 3.3VDC và 5VDC như: Arduino, Raspberry Pi, Jetson Nano, Micro:bit, ....

Bổ sung thêm các thiết kế ổn định, chống nhiễu.

Chuẩn kết nối: connector XH2.54 3Pins.

* **Cảm biến BMP180:**

A blue circuit board with white text

Description automatically generated

Hình : Cảm biến BMP180

Phạm vi áp suất: 300 hPa đến 1100 hPa (hectopascal).

Phạm vi nhiệt độ hoạt động: -40°C đến 85°C.

Độ chính xác áp suất: ±1 hPa.

Độ chính xác nhiệt độ: ±0.5°C.

Độ phân giải áp suất: 0.01 hPa (tương đương với 8 cm độ chênh áp).

Độ phân giải nhiệt độ: 0.1°C.

Giao tiếp: I2C hoặc SPI (Serial Peripheral Interface).

Điện áp hoạt động: Thường là 1.8V đến 3.6V.

Tiêu thụ năng lượng: Thấp, trong khoảng vài microampe.

Thời gian đáp ứng: Nhanh, thường dưới 5 ms cho áp suất và nhiệt độ.

* **Cảm biến DHT22:**

A small white square object with metal pins

Description automatically generated

Hình : Cảm biến DHT22

Phạm vi đo nhiệt độ: -40°C đến 80°C.

Độ chính xác nhiệt độ: Thường là ±0.5°C.

Phạm vi đo độ ẩm: 0% đến 100%.

Độ chính xác độ ẩm: Thường là ±2-5%, tùy thuộc vào điều kiện hoạt động.

Thời gian đáp ứng: Thường dưới 2 giây.

Điện áp hoạt động: 3.3V hoặc 5V DC.

Giao tiếp: Sử dụng giao thức số, thường là 1-wire hoặc 2-wire.

* **Cảm biến mưa:**

A small electronic device with wires

Description automatically generated

Hình : Cảm biến mưa

Điện áp: 5VDC

Led báo nguồn: Màu xanh

Led cảnh báo mưa : Màu đỏ

Hoạt động dựa trên nguyên lý Nước rơi vào board sẽ tạo ra môi trường dẫn điện

Dạng tín hiệu: Analog (AO) và Digital (DO)

IC sử dụng: LM358 để chuyển AO --> DO

Kích thước Boaảd: Bằng biến trở

* **Màn hình LCD và mô-đun I2C:**

Là một mô-đun hiển thị LCD 16x2 (16 cột x 2 hàng) có tích hợp một module chuyển đổi giao diện I2C. Module chuyển đổi I2C giúp đơn giản hóa quá trình kết nối và điều khiển LCD thông qua giao diện I2C, giảm số lượng dây kết nối cần thiết.



Hình : Màn hình LCD và mô-đun I2C

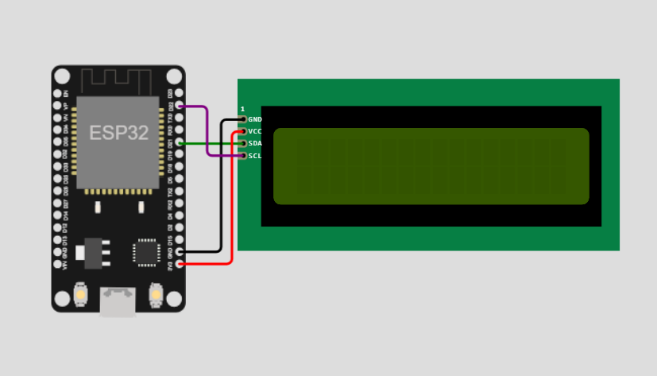
Đặc điểm:

* Kích thước và độ phân giải: Mô-đun LCD 1602 có kích thước 84x44mm và có khả năng hiển thị 16 cột x 2 hàng ký tự, tức là có thể hiển thị tối đa 32 ký tự cùng một lúc.
* Giao diện I2C: Module chuyển đổi I2C "LCD 8050 R7" tích hợp trên mô-đun LCD 1602 cho phép giao tiếp thông qua giao diện I2C. Thông qua đó, mô-đun LCD có thể được kết nối và điều khiển bằng các thiết bị khác sử dụng chỉ hai dây: dây dữ liệu (SDA) và dây đồng hồ (SCL).
* Điều khiển dễ dàng: Với module chuyển đổi I2C, việc điều khiển mô-đun LCD 1602 trở nên dễ dàng hơn. Thay vì phải sử dụng nhiều chân GPIO để điều khiển LCD theo giao thức song song, ta chỉ cần kết nối một cặp dây I2C và sử dụng lệnh giao tiếp I2C để điều khiển hiển thị trên màn hình LCD.
* Đa ngôn ngữ: Mô-đun LCD 1602 với module chuyển đổi I2C "LCD 8050 R7" hỗ trợ nhiều ngôn ngữ và ký tự đặc biệt, cho phép hiển thị các ký tự tiếng Anh, ký tự Unicode, ký tự tùy chỉnh và nhiều ngôn ngữ khác trên màn hình LCD.

## **Test linh kiện.**

#### **ESP kết nối với màn hình LCD.**

1. Sơ đồ phần cứng.



Hình : Sơ đồ mô phỏng ESP kết nối với màn hình LCD

1. Sơ đồ thuật toán.

A screen shot of a cell phone

Description automatically generated

Hình : Sơ đồ thuật toán ESP kết nối với màn hình LCD

1. Chương trình.

#include <Wire.h>

#include <LiquidCrystal\_I2C.h>

LiquidCrystal\_I2C LCD = LiquidCrystal\_I2C(0x27, 16, 2);

void displayTextOnLCD(const char\* text, int row, int column) {

  LCD.setCursor(column, row);

  LCD.print(text);

}

void setup() {

  LCD.init();

  LCD.backlight();

  displayTextOnLCD("test lcd", 0, 0); // Hiển thị "test lcd" ở hàng 0, cột 0

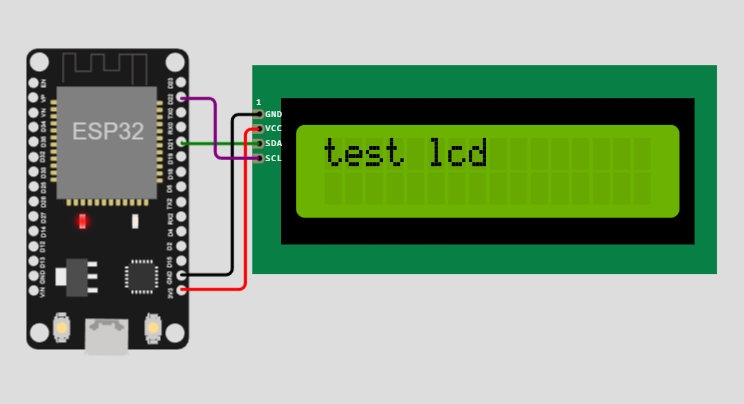
}

void loop() {

  // Không cần thực hiện gì trong hàm loop nếu bạn chỉ muốn hiển thị "test lcd" một lần khi khởi động.

}

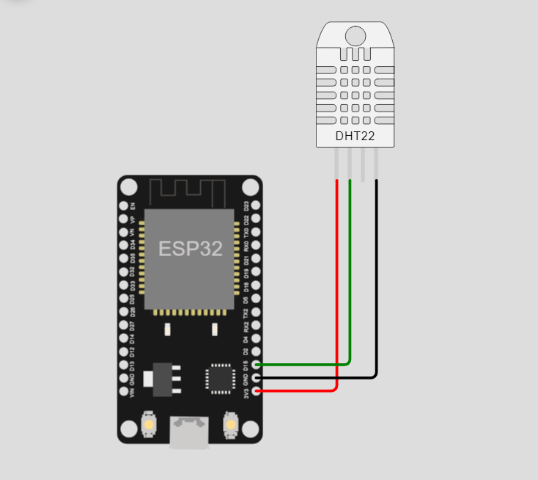
1. Kết quả.



Hình : Kết quả mô phỏng ESP kết nối với màn hình LCD

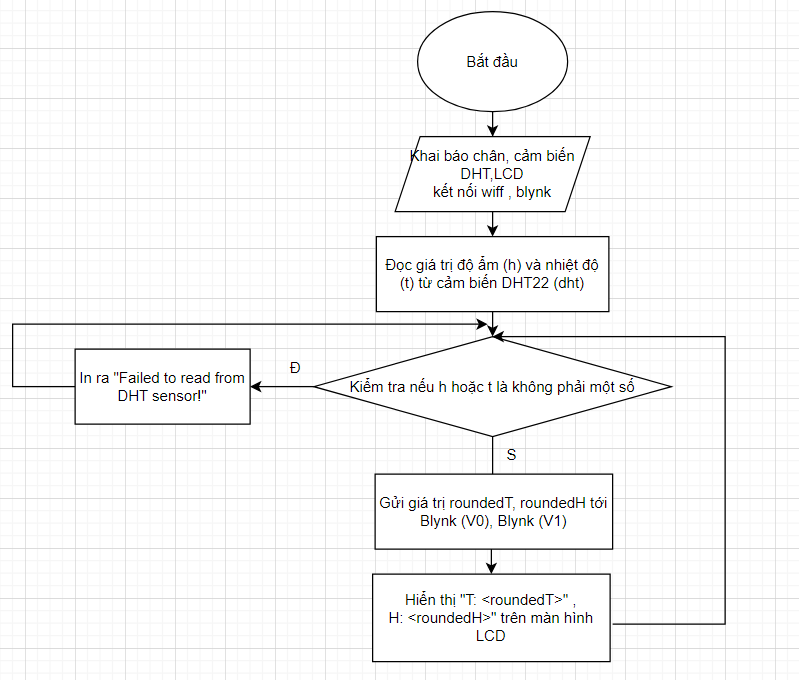
#### **ESP kết nối với cảm biến DHT22.**

* 1. Sơ đồ phần cứng.



Hình : Sơ đồ mô phỏng ESP kết nối với cảm biến DHT22

* 1. Sơ đồ thuật toán.



Hình : Sơ đồ thuật toán ESP kết nối với cảm biến DHT22

* 1. Chương trình.

#define BLYNK\_PRINT **Serial**

#define BLYNK\_TEMPLATE\_ID "TMPL6SQQgPeca"

#define BLYNK\_TEMPLATE\_NAME "Du bao thoi tiet"

#define BLYNK\_AUTH\_TOKEN "KyHKrXtZK8LD4hRdMcv54QF1GB4IYRfs"

#define LED 2

#include <WiFi.h>

#include <WiFiClient.h>

#include <BlynkSimpleEsp32.h>

#include "DHTesp.h"

const int DHT\_PIN = 15;

DHTesp dhtSensor;

char auth[] = "KyHKrXtZK8LD4hRdMcv54QF1GB4IYRfs";

char ssid[] = "Wokwi-GUEST";

char pass[] = "";

BlynkTimer timer;

BLYNK\_WRITE(V0)

{

int pinValue = param.asInt();

digitalWrite(LED,pinValue);

}

void sendSensor()

{

 TempAndHumidity  data = dhtSensor.getTempAndHumidity();

**Serial**.println("Temp: " + String(data.temperature, 2) + "°C");

**Serial**.println("Humidity: " + String(data.humidity, 1) + "%");

**Serial**.println("---");

 Blynk.virtualWrite(V0, data.temperature); //mengirimkan data ke Virtual pin di Blynk Cloud

Blynk.virtualWrite(V1, data.humidity);

}

void setup()

{

**Serial**.begin(9600);

  Blynk.begin(BLYNK\_AUTH\_TOKEN, ssid, pass);

  pinMode(LED, OUTPUT);

  dhtSensor.setup(DHT\_PIN, DHTesp::DHT22);

  timer.setInterval(2000L, sendSensor);

}

void loop()

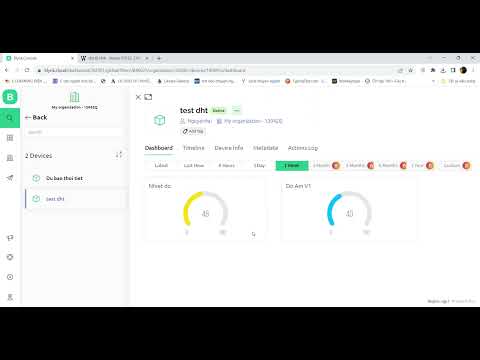
{

  Blynk.run();

  timer.run();

}

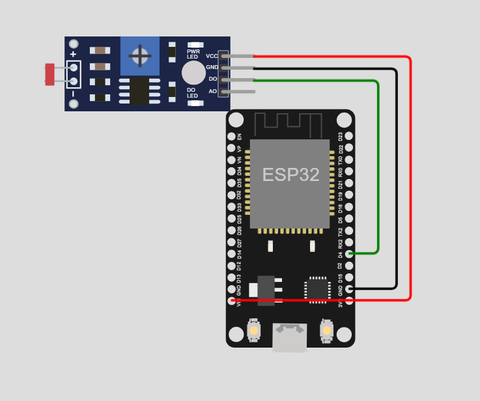
* 1. Kết quả.

[](https://www.youtube.com/embed/Hbd99rwU7zQ?feature=oembed)

Video : Kết quả mô phỏng ESP kết nối với cảm biến DHT22

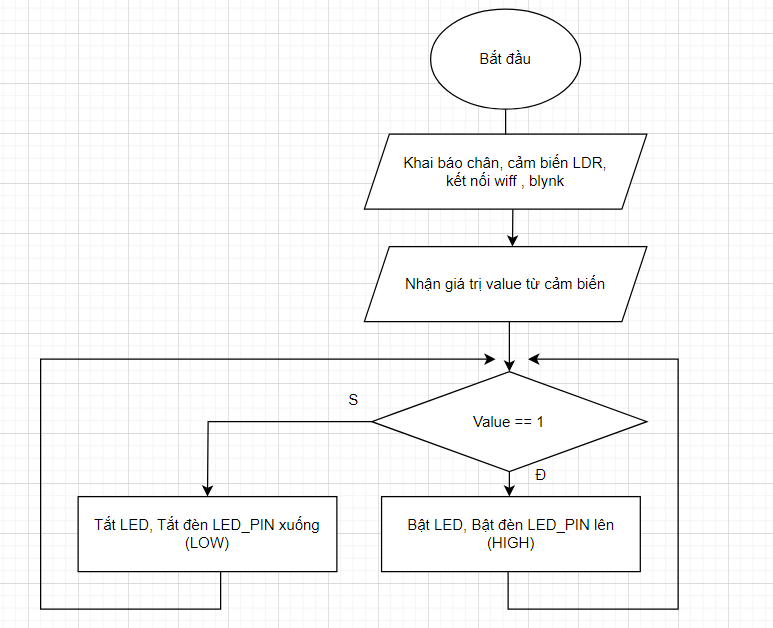
#### **ESP kết nối với cảm biến LDR.**

1. Sơ đồ phần cứng.



Hình : Sơ đồ mô phỏng ESP kết nối với cảm biến LDR

1. Sơ đồ thuật toán.



Hình : Sơ đồ thuật toán ESP kết nối với cảm biến LDR

1. Chương trình.

#define BLYNK\_TEMPLATE\_ID "TMPL6SQQgPeca"

#define BLYNK\_TEMPLATE\_NAME "Du bao thoi tiet"

#define BLYNK\_AUTH\_TOKEN "KyHKrXtZK8LD4hRdMcv54QF1GB4IYRfs"

#include <BlynkSimpleEsp32.h>

#include <WiFi.h>

char auth[] = BLYNK\_AUTH\_TOKEN;

char ssid[] = "Wokwi-GUEST";

char pass[] = "";

#define LDR 4

BlynkTimer timer;

void setup() {

**Serial**.begin(115200);

  Blynk.begin(auth, ssid, pass);

  pinMode(LDR, INPUT);

  analogReadResolution(12);

}

void LDRsensor() {

  bool value = digitalRead(LDR);

  if (value == 1) {

    WidgetLED LED(V4);

    LED.on();

  } else {

    WidgetLED LED(V4);

    LED.off();

  }

}

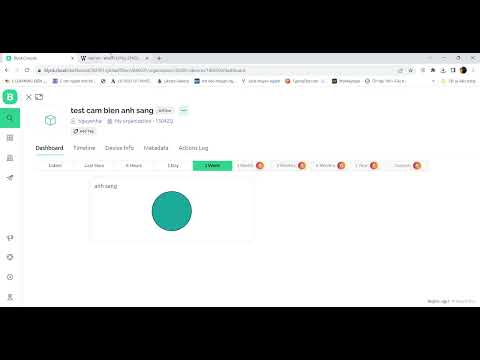
void loop() {

  LDRsensor();

  Blynk.run();

}

1. Kết quả.

[](https://www.youtube.com/embed/vGhue5vhIVs?feature=oembed)

Video : Kết quả mô phỏng ESP kết nối với cảm biến LDR

#### **ESP kết nối với cảm biến BMP180.**

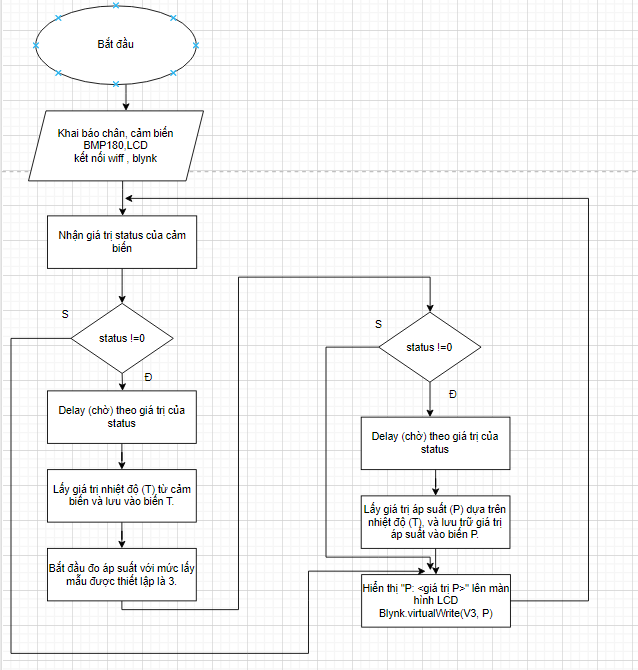
1. Sơ đồ phần cứng.

A circuit board with a screen and wires

Description automatically generated

Hình : Sơ đồ mô phỏng ESP kết nối với cảm biến BMP180

1. Sơ đồ thuật toán.



Hình : Sơ đồ thuật toán ESP kết nối với cảm biến BMP180

1. Chương trình.

#define BLYNK\_TEMPLATE\_ID "TMPL6\_Btna6J2"

#define BLYNK\_TEMPLATE\_NAME "test ap suat"

#define BLYNK\_AUTH\_TOKEN "h7Bp9CwbcAoKQkEpxhCgyS8yLtG3Axce"

#include <LiquidCrystal\_I2C.h>

#include <Wire.h>

#include <WiFiClient.h>

#include <BlynkSimpleEsp32.h>

#include <Adafruit\_Sensor.h>

#include <DHT.h>

#include <SFE\_BMP180.h>

// Tạo biến cho áp suất

double T, P;

char status;

// Khởi tạo màn hình LCD

LiquidCrystal\_I2C lcd(0x27, 16, 2);

// Tạo đối tượng cho cảm biến BMP180

SFE\_BMP180 bmp;

BlynkTimer timer;

char ssid[] = "nguyenhai";

char pass[] = "23456788";

void setup() {

  // Bộ định tuyến đang dùng

  Serial.begin(115200);

  Blynk.begin(BLYNK\_AUTH\_TOKEN, ssid, pass, "blynk.cloud", 80);

  bmp.begin();

  lcd.init();

lcd.backlight();

  analogReadResolution(12);

lcd.setCursor(0, 0);

  lcd.print("He Thong Du Bao");

  lcd.setCursor(4, 1);

  lcd.print("Thoi Tiet");

  delay(4000);

  lcd.clear();

}

// Lấy giá trị áp suất

void pressure() {

status = bmp.startTemperature();

  if (status != 0) {

    delay(status);

    status = bmp.getTemperature(T);

status = bmp.startPressure(3); // 0 đến 3

    if (status != 0) {

      delay(status);

      status = bmp.getPressure(P, T);

      if (status != 0) {

      }

    }

  }

Blynk.virtualWrite(V3, P);

  lcd.setCursor(0, 0);

  lcd.print("Ap Suat P:");

  lcd.setCursor(1, 5);

  lcd.print(P);

}

void loop() {

  pressure();

  Blynk.run(); // Chạy thư viện Blynk

}

1. Kết nối.

[](https://www.youtube.com/embed/TSvlvvIS3MU?feature=oembed)

Video : Kết quả mô phỏng ESP kết nối với cảm biến BMP180

#### **ESP kết nối với cảm biến mưa.**

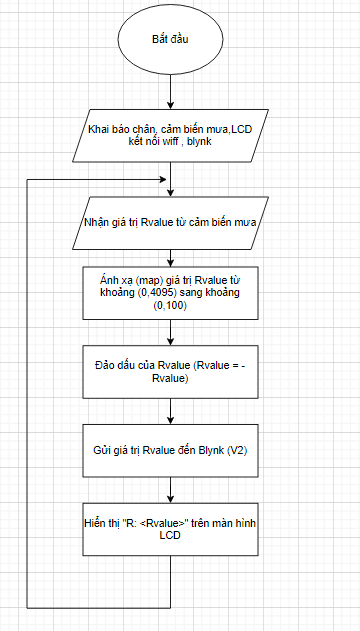
1. Sơ đồ phần cứng.

A diagram of a circuit board

Description automatically generated

Hình : Sơ đồ mô phỏng ESP kết nối với cảm biến mưa

1. Sơ đồ thuật toán.



Hình : Sơ đồ thuật toán ESP kết nối với cảm biến mưa

1. Chương trình.

#define BLYNK\_TEMPLATE\_ID "TMPL6v1XekkCP"

#define BLYNK\_TEMPLATE\_NAME "testcambienmua"

#define BLYNK\_AUTH\_TOKEN "IjceKeVyL-bBWZHnKFdHjsYUjmQyPFB9"

#include <LiquidCrystal\_I2C.h>

#include <Wire.h>

#include <WiFiClient.h>

#include <BlynkSimpleEsp32.h>

#include <Adafruit\_Sensor.h>

#include <DHT.h>

#include <SFE\_BMP180.h>

#define Rain 36

// Khởi tạo màn hình LCD

LiquidCrystal\_I2C lcd(0x27, 16, 2);

BlynkTimer timer;

char ssid[] = "nguyenhai";

char pass[] = "23456788";

void setup() {

  // Bộ định tuyến đang dùng

Serial.begin(115200);

  Blynk.begin(BLYNK\_AUTH\_TOKEN, ssid, pass, "blynk.cloud", 80);

  lcd.init();

  lcd.backlight();

  pinMode(Rain, INPUT);

  analogReadResolution(12);

  lcd.setCursor(0, 0);

  lcd.print("System");

  lcd.setCursor(4, 1);

  lcd.print("Loading..");

  delay(4000);

  lcd.clear();

}

// Lấy giá trị từ cảm biến mưa

void rainSensor() {

  int Rvalue = analogRead(Rain);

  Rvalue = map(Rvalue, 0, 4095, 0, 100);

  Rvalue = (Rvalue - 100) \* -1;

  Blynk.virtualWrite(V2, Rvalue);

  lcd.setCursor(0, 0);

  lcd.print("Luong mua R:");

  lcd.setCursor(1, 5);

  lcd.print(Rvalue);

}

void loop() {

  rainSensor();

  Blynk.run(); // Chạy thư viện Blynk

}

1. Kết quả.

[](https://www.youtube.com/embed/tLP2MJa9Q_k?feature=oembed)

Video : Kết quả mô phỏng ESP kết nối với cảm biến mưa

## **Sản phẩm thực tế.**

1. Sơ đồ phần cứng.

A diagram of a circuit board

Description automatically generated

Hình : Sơ đồ mô phỏng hệ thống giám sát thời tiết ESP32

1. Sơ đồ thuật toán.

A diagram of a flowchart

Description automatically generated

Hình : Sơ đồ thuật toán giám sát thời tiết ESP32

1. Chương trình.

#define BLYNK\_TEMPLATE\_ID "TMPL6SQQgPeca"

#define BLYNK\_TEMPLATE\_NAME "Du bao thoi tiet"

#define BLYNK\_AUTH\_TOKEN "KyHKrXtZK8LD4hRdMcv54QF1GB4IYRfs"

#include <LiquidCrystal\_I2C.h>

#include <Wire.h>

#include <WiFiClient.h>

#include <BlynkSimpleEsp32.h>

#include <Adafruit\_Sensor.h>

#include <DHT.h>

#include <SFE\_BMP180.h>

#define LDR 4

#define TH 5

#define Rain 36

#define LED\_PIN 2

// Tạo biến cho áp suất

double T, P;

char status;

// Khởi tạo màn hình LCD

LiquidCrystal\_I2C lcd(0x27, 16, 2);

// Tạo đối tượng cho cảm biến BMP180

SFE\_BMP180 bmp;

DHT dht(TH, DHT22);  // Sử dụng DHT22 cho cảm biến DHT

BlynkTimer timer;

char ssid[] = "nguyenhai";

char pass[] = "23456788";

void setup() {

pinMode(LED\_PIN, OUTPUT);

  // Bộ định tuyến đang dùng

Serial.begin(115200);

  Blynk.begin(BLYNK\_AUTH\_TOKEN, ssid, pass, "blynk.cloud", 80);

  bmp.begin();

  dht.begin();

  lcd.init();

  lcd.backlight();

  pinMode(LDR, INPUT);

  pinMode(Rain, INPUT);

  analogReadResolution(12);

  lcd.setCursor(0, 0);

  lcd.print("He Thong Du Bao");

  lcd.setCursor(4, 1);

  lcd.print("Thoi Tiet");

  delay(4000);

  lcd.clear();

}

// Lấy giá trị từ cảm biến DHT22

void DHT22sensor() {

  float h = dht.readHumidity();

  float t = dht.readTemperature();

  if (isnan(h) || isnan(t)) {

    Serial.println("Failed to read from DHT sensor!");

    return;

  }

  int roundedT = int(t);  // Làm tròn nhiệt độ thành số nguyên

  int roundedH = int(h);  // Làm tròn độ ẩm thành số nguyên

  Blynk.virtualWrite(V0, roundedT);

  Blynk.virtualWrite(V1, roundedH);

 lcd.setCursor(0, 0);

  lcd.print("T:");

  lcd.print(roundedT);

lcd.setCursor(8, 0);

  lcd.print("H:");

  lcd.print(roundedH);

}

// Lấy giá trị từ cảm biến mưa

void rainSensor() {

  int Rvalue = analogRead(Rain);

  Rvalue = map(Rvalue, 0, 4095, 0, 100);

  Rvalue = (Rvalue - 100) \* -1;

  Blynk.virtualWrite(V2, Rvalue);

  lcd.setCursor(0, 1);

  lcd.print("R:");

  lcd.print(Rvalue);

  lcd.print(" ");

}

// Lấy giá trị áp suất

void pressure() {

  status = bmp.startTemperature();

  if (status != 0) {

    delay(status);

    status = bmp.getTemperature(T);

    status = bmp.startPressure(3); // 0 đến 3

    if (status != 0) {

    delay(status);

status = bmp.getPressure(P, T);

      if (status != 0) {

}

 }

  }

  Blynk.virtualWrite(V3, P);

  lcd.setCursor(8, 1);

  lcd.print("P:");

  lcd.print(P);

}

// Lấy giá trị từ cảm biến LDR

void LDRsensor() {

  bool value = digitalRead(LDR);

  if (value == 1) {

    WidgetLED LED(V4);

    LED.on();

    digitalWrite(LED\_PIN, HIGH); // Bật LED

  } else {

    WidgetLED LED(V4);

    LED.off();

    digitalWrite(LED\_PIN, LOW); // Tắt LED

  }

}

void loop() {

  DHT22sensor();

  rainSensor();

  pressure();

  LDRsensor();

  Blynk.run(); // Chạy thư viện Blynk

}

1. Mô hình thực tế.

[](https://www.youtube.com/embed/SVeT_Cm26dc?feature=oembed)

Video : Kết quả mô phỏng hệ thống giám sát thời tiết ESP32

# **PHẦN 3. KẾT LUẬN VÀ KIẾN NGHỊ**

1. **Kết luận**

Hệ thống giám sát thời tiết là một phần quan trọng của việc dự đoán và theo dõi thời tiết để đảm bảo an toàn, hỗ trợ quyết định và thúc đẩy sự phát triển bền vững trong nhiều lĩnh vực. Dưới đây là một số điểm kết luận về hệ thống giám sát thời tiết:

* Quan trọng cho An toàn và Bảo vệ: Hệ thống giám sát thời tiết cung cấp thông tin quan trọng để đưa ra cảnh báo và lập kế hoạch cho các sự kiện thời tiết cực đoan như bão, lũ lụt, và cơn bão. Điều này giúp bảo vệ cuộc sống và tài sản của con người.
* Hỗ trợ Quyết định: Dữ liệu từ hệ thống giám sát thời tiết giúp các quyết định liên quan đến nông nghiệp, giao thông, hàng hải, hàng không, và nhiều lĩnh vực khác. Điều này cải thiện hiệu suất và giảm thiểu rủi ro.
* Dự báo Thời tiết: Hệ thống giám sát cung cấp thông tin cơ bản cho các dự báo thời tiết, giúp mọi người lập kế hoạch cho hoạt động hàng ngày và sự kiện dự kiến.
* Khoa học và Nghiên cứu: Dữ liệu thu thập từ hệ thống giám sát thời tiết đóng vai trò quan trọng trong nghiên cứu về biến đổi khí hậu, hiểu biết về các mô hình thời tiết và cung cấp thông tin về khí tượng học và thảm họa thiên nhiên.
* Liên quan Toàn cầu: Thời tiết không bao giờ giới hạn trong một quốc gia hoặc khu vực, và hệ thống giám sát thời tiết là một phần của mạng lưới toàn cầu. Các quốc gia cùng hợp tác và chia sẻ thông tin để dự đoán và ứng phó với các biến đổi thời tiết toàn cầu.
* Công nghệ và Cách tiếp cận Tiến bộ: Sự phát triển trong công nghệ đã cải thiện khả năng giám sát thời tiết, từ việc sử dụng máy bay không người lái cho đến việc áp dụng trí tuệ nhân tạo trong việc phân tích dữ liệu thời tiết.
* Thách thức và Tài trợ: Mặc dù hệ thống giám sát thời tiết rất quan trọng, chúng đôi khi gặp khó khăn về tài trợ và hạ tầng. Việc đảm bảo rằng hệ thống này được duy trì và phát triển là một thách thức quan trọng.

Tổng cộng, hệ thống giám sát thời tiết đóng một vai trò quan trọng trong cuộc sống hàng ngày của chúng ta, hỗ trợ an toàn, quyết định và sự phát triển bền vững, đồng thời mang lại thông tin cần thiết để nghiên cứu và hiểu biết về môi trường và biến đổi khí hậu toàn cầu.

1. **Kiến nghị.**

Dưới đây là một số kiến nghị về cách cải thiện hệ thống giám sát thời tiết:

* Đầu tư vào Hạ tầng Công nghệ: Cải thiện và mở rộng hạ tầng công nghệ để cung cấp dữ liệu thời tiết chính xác và nhanh chóng. Điều này bao gồm việc sử dụng máy bay không người lái, vệ tinh, và các thiết bị cảm biến tiên tiến.
* Hợp tác Quốc tế: Tăng cường hợp tác quốc tế trong việc chia sẻ thông tin thời tiết và dữ liệu giữa các quốc gia để cải thiện khả năng dự đoán và ứng phó với biến đổi thời tiết quốc tế.
* Chú trọng Đào tạo và Nghiên cứu: Đầu tư vào đào tạo và nghiên cứu trong lĩnh vực khí tượng học và hệ thống giám sát thời tiết để có thêm kiến thức và kỹ năng cần thiết cho người làm việc trong ngành này.
* Cải thiện Dự báo Thời tiết Cao cấp: Tạo điều kiện để cải thiện dự báo thời tiết cận kề và dài hạn. Điều này có thể giúp người dân và các tổ chức lập kế hoạch tốt hơn cho các sự kiện thời tiết quan trọng.
* Khảo sát Biến đổi Khí hậu: Tăng cường năng lực để theo dõi và khảo sát biến đổi khí hậu và hiểu biết về tác động của nó đối với môi trường và xã hội.
* Truyền thông Hiệu quả: Cải thiện truyền thông về thông tin thời tiết để đảm bảo rằng dự báo và cảnh báo đến được người dân một cách hiệu quả, đặc biệt là trong trường hợp khẩn cấp.
* Sử dụng Trí tuệ Nhân tạo (AI) và Học Máy: Áp dụng công nghệ AI và học máy để phân tích và dự đoán dữ liệu thời tiết một cách chính xác hơn và nhanh chóng.
* Bảo vệ Dữ liệu và Quyền riêng tư: Đảm bảo rằng dữ liệu thời tiết được bảo vệ và quản lý một cách an toàn, và đảm bảo quyền riêng tư của cá nhân được tuân thủ.
* Thúc đẩy Nghiên cứu ứng dụng: Khuyến khích nghiên cứu và phát triển ứng dụng công nghệ mới trong lĩnh vực giám sát thời tiết để nâng cao khả năng dự đoán và ứng phó với các sự kiện thời tiết cực đoan.
* Hệ thống Giám sát Thời tiết Công bằng: Đảm bảo rằng hệ thống giám sát thời tiết được xây dựng và phát triển một cách công bằng và hỗ trợ cả cộng đồng quốc tế.

Những kiến nghị này có thể giúp nâng cao hiệu suất và tính bền vững của hệ thống giám sát thời tiết, đồng thời cung cấp thông tin quan trọng cho quyết định và an toàn của con người trước biến đổi thời tiết và biến đổi khí hậu.