

1. Mục tiêu (Mission Objective)

Hệ thống đo nhiệt độ & độ ẩm từ cảm biến AHT20, hiển thị tại chỗ trên LCD, truyền dữ liệu qua Wi-Fi (ESP8266) lên nền tảng IoT (ThingSpeak). Ngoài ra, hệ thống có cảnh báo tại chỗ (buzzer) khi giá trị vượt ngưỡng cấu hình.

2. Functional Requirement

Number	Description	Note
R1.1	Đọc được giá trị nhiệt độ & độ ẩm từ cảm biến AHT20 qua I ² C.	
R1.2	Hiển thị dữ liệu đo được lên LCD (Temp xx.x °C, Humi xx.x %RH).	Cập nhật mỗi chu kỳ
R1.3	Kết nối Wi-Fi bằng ESP8266.	UART
R1.4	Gửi dữ liệu (Temp, Humi) lên ThingSpeak bằng API Key.	Field1 = Temp, Field2 = Humi
R1.5	Chu kỳ gửi dữ liệu ≥ 15 giây.	Theo giới hạn ThingSpeak
R1.6	Lưu dữ liệu khi mất Wi-Fi.	EEPROM/Flash
R1.7	Tự động đồng bộ dữ liệu khi Wi-Fi phục hồi.	FIFO
R1.8	Người dùng có thể xem dữ liệu theo thời gian thực trên ThingSpeak.	Web/App
R1.9	Kích hoạt báo động (buzzer) khi nhiệt độ hoặc độ ẩm vượt ngưỡng.	Ngưỡng cài đặt trước

2.1. Đọc giá trị từ AHT20

- Giao tiếp: I²C.
- Chu kỳ đọc: 15 giây/lần (cấu hình được).
- Dải đo:
 - Nhiệt độ: -40 ~ +85 °C, sai số ± 0.3 °C.
 - Độ ẩm: 0 ~ 100 %RH, sai số ± 2 %RH.
- Nếu cảm biến trả về lỗi (CRC hoặc không ACK) \rightarrow hệ thống báo lỗi.

2.2. Hiển thị LCD

- Người dùng có thể quan sát trực tiếp nhiệt độ tại thiết bị.
- LCD 16x2 hiển thị định dạng
 - Dòng 1: Temp: xx.x °C.
 - Dòng 2: Humi yy.y%
 - Nếu lỗi cảm biến hiển thị "Sensor Error"
- Cập nhật theo chu kỳ đọc cảm biến.
- Nếu mất tín hiệu cảm biến \rightarrow hiển thị lỗi.
- Ký tự hiển thị rõ ràng, không vượt quá độ trễ 2 giây sau khi đo.

2.3. Wi-Fi (ESP8266)

- Hệ thống phải truyền dữ liệu qua mạng Wi-Fi để kết nối Internet.
- ESP8266 giao tiếp với Atmega 328 qua UART (AT Command hoặc firmware custom).
- Hỗ trợ chuẩn Wi-Fi 802.11 b/g/n, 2.4 GHz.

- Nếu Wi-Fi mất → hệ thống phải thử kết nối lại định kỳ.
- Cấu hình Wi-Fi (SSID, Password) lưu trong bộ nhớ Atmega 328 hoặc ESP8266

2.4. Gửi dữ liệu ThingSpeak

- Dữ liệu được truyền đến nền tảng IoT để giám sát từ xa.
- Dữ liệu gửi bằng HTTP POST hoặc MQTT (ThingSpeak hỗ trợ).
- Kèm API Key để xác thực người dùng.
- Mỗi bản tin gồm: thời gian (timestamp), nhiệt độ, độ ẩm.
- Tuân thủ giới hạn ThingSpeak (tối thiểu 15 giây giữa 2 lần gửi).

2.5. Chu kỳ gửi dữ liệu

- Đảm bảo dữ liệu cập nhật kịp thời nhưng không vượt giới hạn của server.
- Atmega 328 có bộ định thời (Timer) để điều khiển chu kỳ gửi.
- Tần suất mặc định: 15 giây, nhưng có thể cấu hình ≥ 15 giây.
- Nếu gửi lỗi → thử lại sau 15 giây tiếp theo.
- Không vi phạm policy ThingSpeak (không gửi nhanh hơn).

2.6. Lưu dữ liệu khi mất Wi-Fi

- Không mất dữ liệu trong thời gian Wi-Fi gián đoạn.
- Dữ liệu nhiệt độ lưu tạm vào EEPROM/Flash.
- Ghi kèm timestamp (từ NTP hoặc RTC nội bộ).
- Bộ nhớ tối thiểu phải lưu được ~100 bản ghi khi mất mạng dài hạn.
- Bộ nhớ giới hạn, cần cơ chế ghi vòng tròn (FIFO).

2.7. Đồng bộ khi Wi-Fi khôi phục

- Đảm bảo dữ liệu liên tục, không mất gói khi giám sát từ xa.
- Khi Wi-Fi trở lại → hệ thống lần lượt gửi các bản ghi lưu trong EEPROM.
- Thứ tự đồng bộ: FIFO (theo thời gian đo).
- Khi bộ nhớ đầy → dữ liệu cũ nhất sẽ bị ghi đè.
- Không làm gián đoạn việc gửi dữ liệu hiện tại.

2.8. Xem dữ liệu trên ThingSpeak

- Người dùng theo dõi dữ liệu qua giao diện IoT.
- ThingSpeak hiển thị biểu đồ line chart theo thời gian.
- Dữ liệu cập nhật đúng chu kỳ gửi.
- Người dùng truy cập từ trình duyệt web hoặc app di động.
- Yêu cầu Internet và API Key hợp lệ để xem kênh dữ liệu.

2.9. Báo động

- Có buzzer cảnh báo khi vượt ngưỡng:
 - Temp > Tmax (ví dụ: 40 °C)
 - Temp < Tmin (ví dụ: 0 °C)
 - Humi > Hmax (ví dụ: 80 %RH)
 - Humi < Hmin (ví dụ: 20 %RH)
 - Ngưỡng có thể cấu hình trong firmware.-

- Báo động dừng khi giá trị quay về trong ngưỡng an toàn hoặc có nút Reset.

3. Non-function Requirement

Number	Description	Note
R2.1	Độ chính xác: Nhiệt độ $\pm 0.3\text{ }^{\circ}\text{C}$, Độ ẩm $\pm 2\text{ \%RH}$	
R2.2	ThingSpeak hiển thị biểu đồ line chart (Temp & Humi)	
R2.3	Độ trễ từ lúc đo đến khi gửi dữ liệu ≤ 2 giây.	
R2.4	Hoạt động trong môi trường $-30\text{ }^{\circ}\text{C}$ đến $80\text{ }^{\circ}\text{C}$.	
R2.5	Nguồn DC 12V-1A, ổn áp xuống 5V/3.3V.	
R2.6	Sử dụng giao thức HTTP/MQTT chuẩn do ThingSpeak hỗ trợ.	
R2.7	Chi phí linh kiện $< 500.000\text{ VND}$.	
R2.8	Báo động phát trong ≤ 1 giây kể từ khi vượt ngưỡng.	

3.1. Độ chính xác đo nhiệt độ sai số không vượt quá $\pm 1\text{ }^{\circ}\text{C}$

- Đảm bảo dữ liệu đo có độ tin cậy cho giám sát và phân tích.
- Cảm biến AHT20 có sai số:
 - Nhiệt độ: $\pm 0.3\text{ }^{\circ}\text{C}$ (trong dải $-40\text{ }^{\circ}\text{C} \rightarrow 85\text{ }^{\circ}\text{C}$).
 - Độ ẩm: $\pm 2\text{ \%RH}$ (trong dải $20\text{ \%} \rightarrow 80\text{ \%RH}$).
- Cần hiệu chỉnh (calibration) ban đầu để giảm sai số hệ thống.
- Vì AHT20 xuất dữ liệu digital (I²C) nên không phụ thuộc vào ADC, giúp tránh nhiễu và sai số do nguồn tham chiếu.

3.2. Giao diện ThingSpeak phải hiển thị biểu đồ dữ liệu kiểu line chart

- Người dùng dễ dàng theo dõi sự biến đổi nhiệt độ theo thời gian.
- ThingSpeak kênh dữ liệu phải bật chế độ “Line Chart”.
- Biểu đồ line chart:
 - Field1 = Nhiệt độ ($^{\circ}\text{C}$).
 - Field2 = Độ ẩm (\%RH).
- Trục X: thời gian (timestamp), trục Y: giá trị cảm biến.
- Biểu đồ có thể hiển thị theo ngày, giờ hoặc thời gian thực.
- Phụ thuộc vào tính năng visualization của ThingSpeak.

3.3. Từ lúc đo đến lúc gửi dữ liệu không quá 2 giây

- Đảm bảo độ trễ thấp, dữ liệu cập nhật gần với thời gian thực.
- AHT20 có thời gian đo $\sim 80\text{--}100\text{ ms}$.
- Xử lý dữ liệu & hiển thị LCD $< 500\text{ ms}$.
- Gửi dữ liệu ThingSpeak qua Wi-Fi < 1.5 giây (nếu mạng ổn định).
- Tổng độ trễ < 2 giây.

3.4. Hệ thống phải hoạt động trong dải nhiệt độ môi trường $-30\text{ }^{\circ}\text{C}$ đến $80\text{ }^{\circ}\text{C}$

- Cho phép thiết bị hoạt động ổn định trong nhiều điều kiện môi trường.
- Linh kiện chọn loại chịu nhiệt độ công nghiệp ($-40\text{ }^{\circ}\text{C} \rightarrow +125\text{ }^{\circ}\text{C}$).
- Vỏ bảo vệ chống ẩm, bụi.
- Nguồn adapter và LCD có thể hạn chế dải hoạt động ($0-70\text{ }^{\circ}\text{C}$), cần chọn loại phù hợp.

3.5. Sử dụng nguồn một chiều 12V-1A

- Cung cấp đủ năng lượng cho hệ thống (ATmega 328 + ESP8266 + LCD).
- Dùng adapter 12V-1A, có IC ổn áp xuống 5V và 3.3V.
- Công suất tối thiểu 12W, đủ cho các module hoạt động đồng thời.
- Cần mạch bảo vệ chống ngắn mạch, quá dòng.

3.6. Sử dụng giao thức HTTP/MQTT chuẩn do ThingSpeak hỗ trợ

- Đảm bảo khả năng tích hợp với nền tảng IoT phổ biến.
- Hệ thống hỗ trợ HTTP POST hoặc MQTT publish.
- Có khả năng thêm Header chứa API Key để xác thực.
- Giới hạn tốc độ gửi dữ liệu của ThingSpeak (≥ 15 giây/lần).

3.7. Chi phí linh kiện ít hơn 500.000 VND

- Đảm bảo hệ thống rẻ, dễ triển khai cho nghiên cứu/DIY.
- ATmega328, AHT20, LCD, ESP8266, buzzer/LED, EEPROM/Flash, nguồn, PCB.
- Tổng chi phí < 500k VND (chưa tính công lắp ráp).

3.8. Thời gian đáp ứng báo động ≤ 1 giây

- Khi giá trị nhiệt độ hoặc độ ẩm vượt ngưỡng, buzzer phải được kích hoạt trong vòng 1 giây.
- Báo động duy trì cho đến khi thông số trở về ngưỡng an toàn hoặc người dùng tắt thủ công bằng nút Reset.

4. Yêu cầu hệ thống (System Requirements)

4.1. Phần cứng (Hardware)

- Vi điều khiển: ATmega 328 hoặc tương tự, có khả năng giao tiếp UART/SPI/I2C.
- Cảm biến nhiệt độ và độ ẩm: AHT20(I²C).
- LCD 16x2.
- Kết nối Wi-Fi: Module ESP8266 để truyền dữ liệu lên Internet.
- Nguồn cấp: 5V DC qua adapter.
- Báo động: Buzzer.
- Nút nhấn Reset/Stop alarm
- Bộ nhớ: EEPROM/Flash để lưu trữ tạm dữ liệu khi mất kết nối mạng.
- Nguồn 12V-1A, IC ổn áp xuống 5V và 3.3V

4.2. Phần mềm nhúng (Embedded Software)

- Xử lý dữ liệu: nhiệt độ, độ ẩm.
- Điều khiển LCD hiển thị.
- Gửi dữ liệu qua Wi-Fi.
- So sánh với ngưỡng → bật/tắt buzzer/LED.
- Lưu dữ liệu khi mất Wi-Fi, đồng bộ khi phục hồi.

4.3. Kết nối IoT (IoT Connectivity)

- Wi-Fi: Chuẩn 2.4 GHz, cấu hình qua AT command.
- ThingSpeak: Field1 = Temp, Field2 = Humi.
- HTTP POST hoặc MQTT.
- Gắn timestamp bằng NTP hoặc RTC.

4.4. Giao diện người dùng (User Interface)

- Trên LCD: Temp & Humi.
- ThingSpeak: Line chart (Temp & Humi).
- Báo động tại chỗ (Buzzer).

Web/Ứng dụng IoT:

- Hiển thị dữ liệu nhiệt độ và độ ẩm theo thời gian thực.
- Biểu đồ dữ liệu theo ngày/giờ.
- Cảnh báo khi nhiệt độ hoặc vượt ngưỡng (qua email/app).
- Nút nhấn (nếu có): Reset kết nối Wi-Fi hoặc cấu hình lại thiết bị.

4.5. Hiệu năng (Performance)

- Độ chính xác nhiệt độ: $\pm 0.3^{\circ}\text{C}$
- Độ chính xác độ ẩm: $\pm 0.2\% \text{ RH}$
- Tần suất đo và gửi: 1 lần/10s (cấu hình được).
- Độ trễ dữ liệu: $< 2\text{s}$ (trong điều kiện Wi-Fi ổn định).
- Độ tin cậy: Hệ thống phải tự động kết nối lại khi mất Wi-Fi.

5. Ràng buộc (Constraints)

- Chỉ hoạt động trong vùng có Wi-Fi.
- Giới hạn bởi khả năng xử lý của vi điều khiển ATmega 328.
- Lưu trữ cục bộ chỉ mang tính tạm thời.

6. Khả năng mở rộng (Future Expansion)

- Hỗ trợ nhiều cảm biến (nhiệt độ, độ ẩm, ánh sáng).
- Tích hợp Bluetooth cho cấu hình ban đầu.
- Hỗ trợ pin và năng lượng mặt trời cho ứng dụng di động.

7. Use case modelling

7.1. Log Temperature & Humidity Data to ThingSpeak

Use Case Name	Log Temperature & Humidity Data to ThingSpeak
Use Case ID	UC001
Scope	System based on ATmega328 microcontroller (MCU), AHT20 sensor (I ² C), Wi-Fi module (ESP8266), and ThingSpeak API for IoT data storage and visualization.
Primary Actor(s)	<ul style="list-style-type: none"> • ATmega328 MCU • AHT20 Sensor (Temperature + Humidity) • Wi-Fi Module (ESP8266) • ThingSpeak Cloud
Stakeholders and Interests	<ul style="list-style-type: none"> • User: Muốn giám sát nhiệt độ và độ ẩm từ xa thông qua ThingSpeak. • System: Đọc dữ liệu từ AHT20, xử lý và gửi đúng định dạng API. • ThingSpeak Server: Nhận dữ liệu hợp lệ để lưu và hiển thị trên kênh.
Preconditions	<ul style="list-style-type: none"> • ATmega328 được cấp nguồn và khởi tạo. • Module Wi-Fi kết nối thành công với mạng Wi-Fi. • Tài khoản ThingSpeak có API Key hợp lệ. • AHT20 được kết nối I²C (SDA, SCL).
Postconditions	<ul style="list-style-type: none"> • Dữ liệu nhiệt độ (°C) và độ ẩm (%) lưu trên ThingSpeak channel. • Người dùng có thể giám sát dữ liệu qua web.
Main Flow of Events	<ul style="list-style-type: none"> • ATmega328 khởi tạo giao tiếp I²C và UART. • ATmega328 gửi lệnh khởi tạo AHT20 (0xBE, 0x08, 0x00). • ATmega328 gửi lệnh trigger đo (0xAC, 0x33, 0x00). • Chờ 75ms, sau đó đọc 6 byte dữ liệu từ AHT20. • Tính toán độ ẩm và nhiệt độ. • ATmega328 gửi dữ liệu qua UART đến ESP8266. • ThingSpeak server lưu dữ liệu vào channel. • Người dùng truy cập để xem biểu đồ nhiệt độ và độ ẩm.
Alternative Flow – Wi-Fi Error	<ul style="list-style-type: none"> • Nếu không kết nối được Wi-Fi → thử lại sau 5s. • Nếu vẫn lỗi → báo bằng LED nhấp nháy.
Alternative Flow – Sensor Error	<ul style="list-style-type: none"> • Nếu AHT20 không phản hồi → gửi giá trị mặc định (-1, -1). • Vẫn update ThingSpeak để biết trạng thái lỗi.
Exception Flows	<ul style="list-style-type: none"> • Mất nguồn → hệ thống khởi động lại và tiếp tục đo. • ThingSpeak từ chối dữ liệu → MCU ghi nhận lỗi và thử lại sau 20s.
Includes	<ul style="list-style-type: none"> • UC002: Read Temp & Humidity from AHT20 (I²C) • UC003: Connect to Wi-Fi (ESP8266)
Extends	<ul style="list-style-type: none"> • UC004: Visualize Temp & Humidity on ThingSpeak
Special Requirements	<ul style="list-style-type: none"> • Chu kỳ gửi dữ liệu ≥ 15 giây (giới hạn ThingSpeak free).

	<ul style="list-style-type: none"> • Điện áp AHT20: 2.0–5.5V (tương thích ATmega328). • I²C tốc độ chuẩn (100kHz hoặc 400kHz).
--	---

7.2. Read Temp & Humidity from AHT20 (I²C)

Use Case Name	Read Temp & Humidity from AHT20
Use Case ID	UC002
Scope	ATmega328 communicates with AHT20 sensor via I ² C to obtain temperature and humidity data.
Primary Actor(s)	<ul style="list-style-type: none"> • ATmega328 MCU • AHT20 Sensor
Stakeholders and Interests	<ul style="list-style-type: none"> • MCU: Cần đọc dữ liệu nhiệt độ và độ ẩm chính xác từ sensor • System: Đảm bảo giao tiếp I²C ổn định và xử lý lỗi phù hợp • User: Nhận được dữ liệu môi trường đáng tin cậy
Preconditions	<ul style="list-style-type: none"> • AHT20 kết nối SDA, SCL với ATmega328. • ATmega328 đã khởi tạo giao tiếp I²C.
Postconditions	<ul style="list-style-type: none"> • MCU nhận được dữ liệu nhiệt độ (°C) và độ ẩm (%RH).
Main Flow of Events	<ul style="list-style-type: none"> • MCU gửi lệnh Initialization (0xBE, 0x08, 0x00) đến AHT20. • MCU gửi lệnh Trigger measurement (0xAC, 0x33, 0x00). • MCU chờ 75ms để AHT20 hoàn thành đo. • MCU đọc 6 byte dữ liệu từ AHT20 qua I²C. MCU tính toán kết quả
Alternative Flow – Sensor Busy	<ul style="list-style-type: none"> • Nếu bit trạng thái [7] = 1 (sensor đang đo) → chờ thêm 10ms và đọc lại.
Exception Flows	<ul style="list-style-type: none"> • Nếu không nhận được ACK từ AHT20 → báo lỗi “Sensor not responding”.

7.3. Connect to Wi-Fi (ESP8266)

Use Case Name	Connect to Wi-Fi via ESP8266
Use Case ID	UC003
Scope	ATmega328 communicates with ESP8266 module over UART using AT commands to establish Wi-Fi connection.
Primary Actor(s)	<ul style="list-style-type: none"> • ATmega328 MCU • ESP8266 Wi-Fi Module
Stakeholders and Interests	<ul style="list-style-type: none"> • MCU: Cần kết nối Wi-Fi ổn định để truyền dữ liệu • System: Đảm bảo kết nối mạng đáng tin cậy • User: Muốn có kết nối internet liên tục để giám sát từ xa
Preconditions	<ul style="list-style-type: none"> • ESP8266 được cấp nguồn 3.3V ổn định. • ATmega328 UART đã khởi tạo (baud rate phù hợp, ví dụ 9600/115200).

Postconditions	<ul style="list-style-type: none"> ESP8266 kết nối thành công đến Access Point (Wi-Fi).
Main Flow of Events	<ul style="list-style-type: none"> MCU gửi lệnh kiểm tra: AT → ESP8266 trả về OK. MCU gửi lệnh đặt chế độ Wi-Fi Station. MCU gửi lệnh kết nối Access Point. ESP8266 trả về WIFI CONNECTED và OK. MCU kiểm tra IP. ESP8266 trả về địa chỉ IP.
Alternative Flow – Wrong Password	<ul style="list-style-type: none"> Nếu ESP8266 trả về FAIL → MCU thử lại tối đa 3 lần, sau đó báo lỗi. <p>Exception Flows</p> <ul style="list-style-type: none"> Nếu ESP8266 không phản hồi → kiểm tra nguồn và UART.

8. System architecture

