ĐẠI HỌC FPT

VIỆN ĐÀO TẠO QUỐC TẾ FPT

THIẾT KẾ VI MẠCH

THIẾT KẾ VÀ TRIỂN KHAI HỆ THỐNG THU THẬP DỮ LIỆU MÔI TRƯỜNG DÙNG FPGA VÀ GIAO THỨC I2C

Tài liệu thiết kế - Học kỳ 2

Giáo viên hướng dẫn: Nguyễn Văn Thành Lộc

Lớp: C1.2411.M0

Tên nhóm: Nhóm 1

	Tên đầy đủ	Mã số Sinh viên
1.	*Nguyễn An Phú*	C1S2408003
2.	*Trần Thông Triết*	C1S2407005
3.	*Nguyễn Trung Tín*	C1S2407008
4.	*Nguyễn Hồng Ngọc*	C1S2409001

Tháng 09/2025

MỤC LỤC

A. Lý thuyết giao thức i2c:

I. Giới thiệu:

Trong lĩnh vực điện tử và hệ thống nhúng, các chuẩn giao tiếp nối tiếp như UART, SPI và I2C đóng vai trò quan trọng trong việc trao đổi dữ liệu giữa các vi mạch. Mỗi giao thức có đặc điểm kỹ thuật, ưu điểm và phạm vi ứng dụng riêng. Trong số đó, I2C (Inter-Integrated Circuit) nổi bật nhờ tính đơn giản, khả năng mở rộng và khả năng kết nối nhiều thiết bị chỉ qua hai đường tín hiệu. I2C được sử dụng rộng rãi trong các ứng dụng yêu cầu truyền thông khoảng cách ngắn, đặc biệt là trong các thiết bị đo lường, cảm biến, bộ nhớ EEPROM, và các mô-đun hiển thị. Với cấu trúc bus gồm một hoặc nhiều bộ điều khiển chính **(master)** và nhiều thiết bị phụ **(slave)**, I2C cho phép truyền dữ liệu đồng bộ, đảm bảo tính toàn vẹn và độ tin cậy cao.

II. Cơ sở lý thuyết:

1. Giao thức I2C là gì:

Giao thức I2C là một giao thức truyền thông đồng bộ, sử dụng hai đường tín hiệu để kết nối và trao đổi dữ liệu giữa các thiết bị. Giao thức I2C được dùng phổ biến để liên lạc ở khoảng cách gần thông qua 2 đường tín hiệu. Trong hệ thống I2C, có thể tồn tại nhiều thiết bị master và nhiều thiết bị slave, tất cả đều chia sẻ chung hai đường tín hiệu chính:

- SCL (Serial Clock): là tín hiệu xung nhịp do master tạo ra, dùng để đồng bộ hóa quá trình truyền dữ liệu.
- SDA (Serial Data): là đường truyền dữ liệu hai chiều, nơi các thiết bị gửi và nhận dữ liệu.

2. Nguyên lý hoạt động của giao thức I2C:

2.1 Địa chỉ trong I2C:

Khi master muốn giao tiếp với một slave cụ thể, nó sẽ gửi khung địa chỉ lên đường SDA và tiếp theo đó là khung dữ liệu mà master cần trao đổi. Các slave sẽ so sánh địa chỉ đó với địa chỉ riêng của mình; nếu trùng khớp với địa chỉ thì slave sẽ phản hồi lại bằng tín hiệu ACK(tức kéo đường tín hiệu SDA xuống mức thấp), ngược lại thì không làm gì tương ứng tín hiệu NACK(đường SDA vẫn ở mức cao).

2.2 Bit xác định thao tác đọc/ghi dữ liệu:

Trong khung truyền địa chỉ, 7 bit cao nhất tương ứng là 7 bit địa chỉ và 1 bit cuối cùng sẽ tương ứng với thao tác mà master muốn thực hiện với slave. Nếu master muốn ghi dữ liệu đến slave, bit cuối cùng này sẽ mang giá trị là 0(mức điện áp thấp). Nếu master muốn đọc dữ liệu từ slave, bit cuối cùng này sẽ mang giá trị là 1(mức điện áp cao).

2.3 Nguyên lý truyền dữ liệu trong I2C:

Maste***r*** có vai trò tạo xung trên đường tín hiệu SCL, việc đọc tín hiệu trên SDA sẽ được đọc trong xung cạnh lên của SCL. Master thực hiện điều khiển thời điểm truyền/nhận, slave chỉ có vai trò đáp ứng lại dữ liệu. Bắt đầu quá trình truyền dữ liệu khi gặp *start condition* và ngừng lại khi có *stop condition:*

- **start condition:** thực hiện bằng cách đưa chân SDA xuống mức 0 trước khi đưa chân SCL xuống mức 0.
- **stop condition:** thực hiện bằng cách đưa chân SDA lên mức 1 sau khi chân SCL lên mức 1.

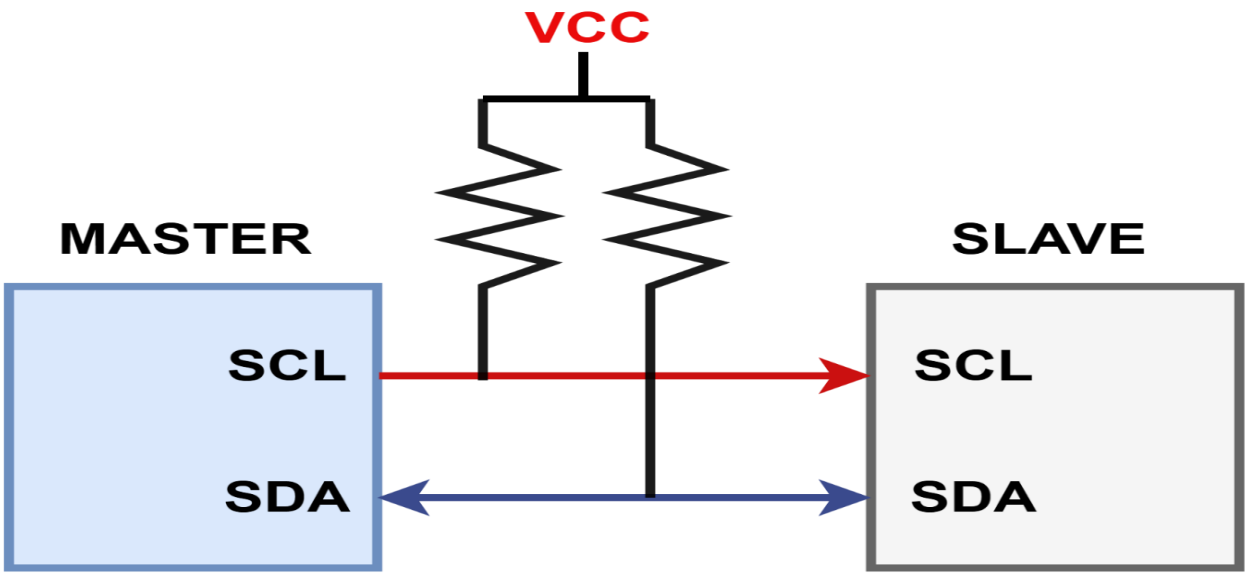
Quá trình truyền trong I2C sẽ thực hiện gửi MSB trước, master gửi 7 bit địa chỉ đến slave + 1 bit thể hiện thao tác cần đọc hay ghi dữ liệu. Nếu có slave khớp với địa chỉ mà master gửi thì slave đó phản hồi lại bằng cách kéo chân SDA xuống mức 0 để thông báo đến master. Dựa trên bit đọc/ghi để xác định thao thác thực hiện, nếu master cần ghi thì slave sẽ nhận dữ liệu và ngược lại. Với mỗi khung dữ liệu trong quá trình truyền thì luôn có bit phản hồi Ack/Nack từ slave hoặc master tùy trường hợp, nếu dữ liệu được truyền đi thành công thì bit Ack sẽ được phản hồi trở lại từ thiết bị nhận, ngược lại sẽ gửi Nack.

Trong giao tiếp sử dụng 7 bit địa chỉ để thực hiện chọn slave để tiến hành giao tiếp, tương ứng 7 bit địa chỉ sẽ có thể có 128 slave giúp mở rộng giao tiếp từ master đến nhiều slave. Trường hợp cùng có nhiều master cùng làm việc trong giao tiếp I2C có thể dẫn đến việc xung đột tín hiệu khi cùng gửi dữ liệu vào SDA. Để khắc phục vấn đề này cần thực hiện kiểm tra chân SDA đang ở mức thấp hay cao, nếu là mức thấp thì đã có master khác đang làm việc cần đợi lên mức cao để truyền dữ liệu một cách an toàn.

2.4 Các cấu hình kết nối trong I2C:

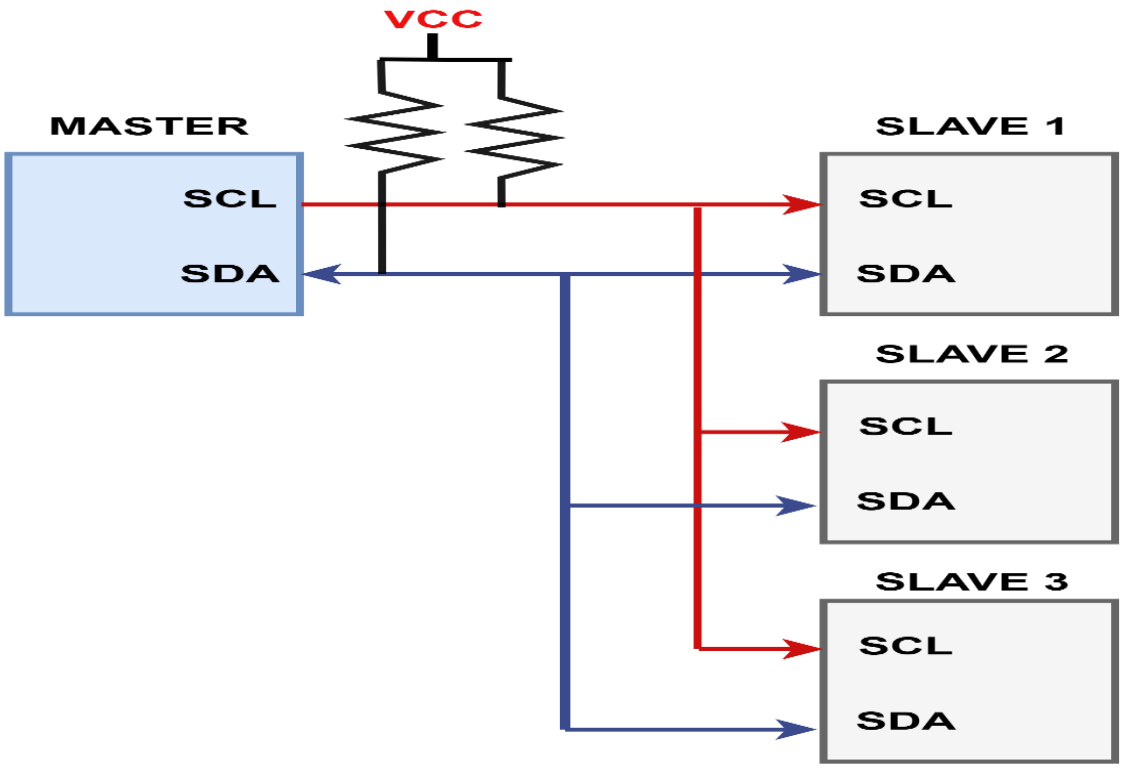
Giao thức I2C chỉ dùng 2 đường tín hiệu là SCL và SDA. Trong quá trình hoạt động thì cả SCL và SDA điều cần điện trở kéo lên (pull-up). Đường truyền kiểu open-drain và open-collector.

2.4.a Sơ đồ kết nối giữa 1 master và 1 slave:



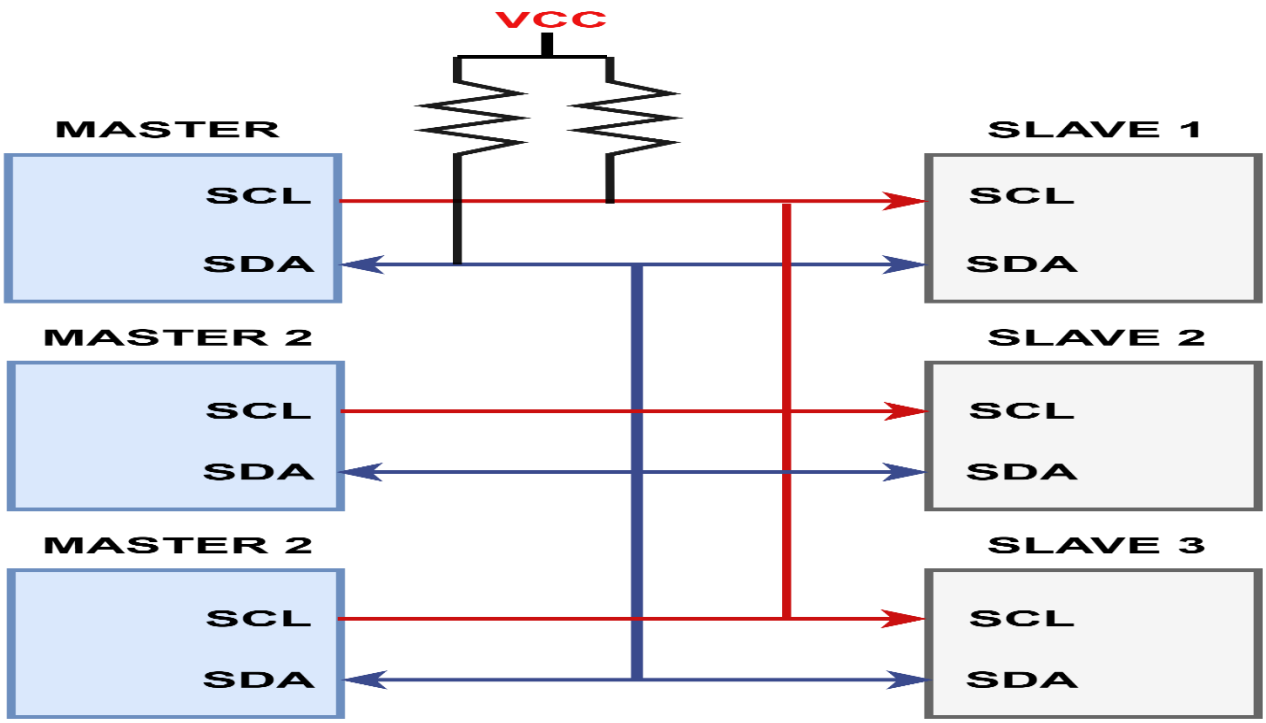
Hình 1: Sơ đồ kết nối 1 master và 1 slave

2.4.b Sơ đồ kết nối 1 master và nhiều slave:



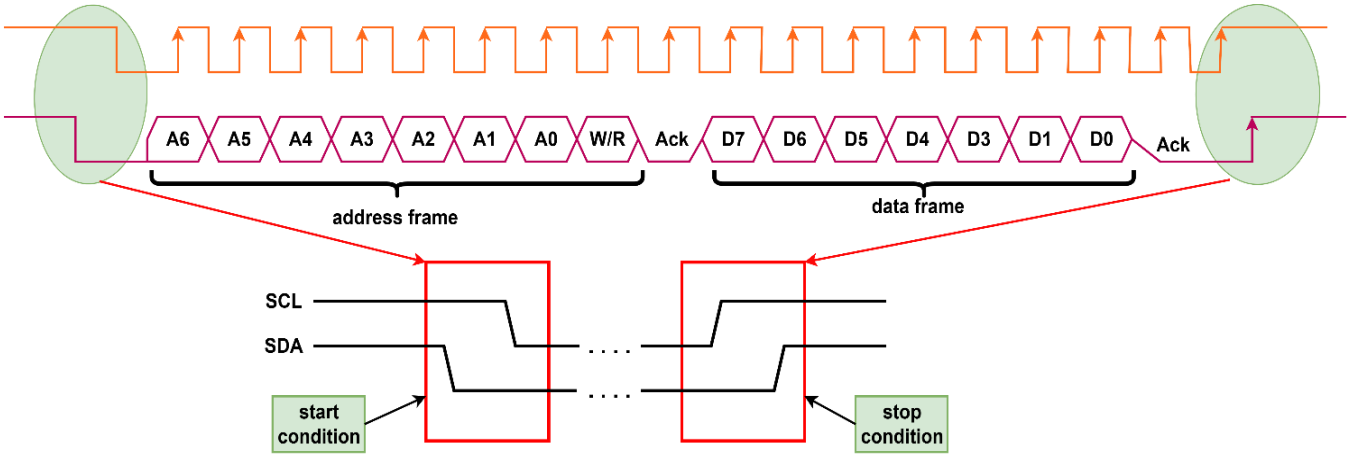
Hình 2: Sơ đồ kết nối 1 master và nhiều slave

2.4.c Sơ đồ kết nối giữa nhiều master và nhiều slave:



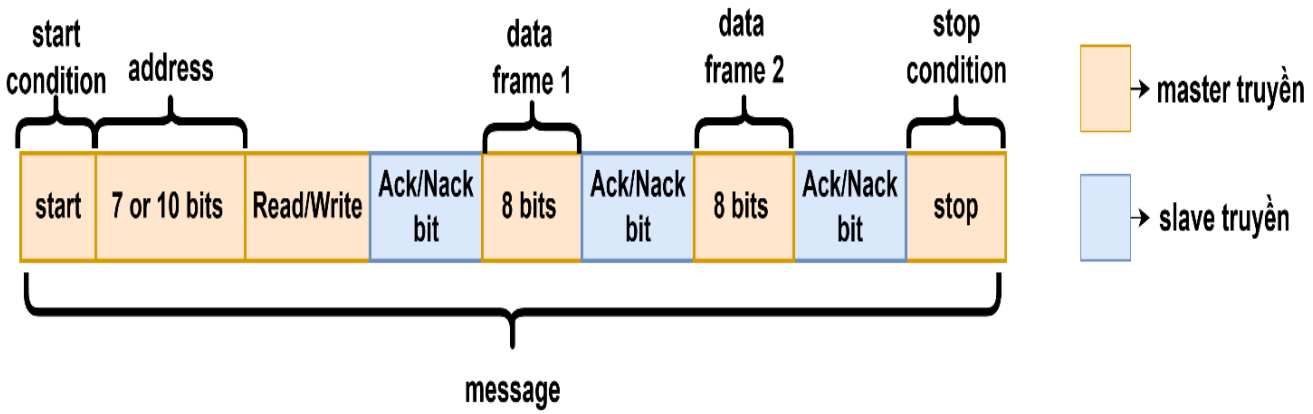
Hình 3: Sơ đồ kết nối giữa nhiều master và nhiều slave

3. Mô tả khung truyền dữ liệu:



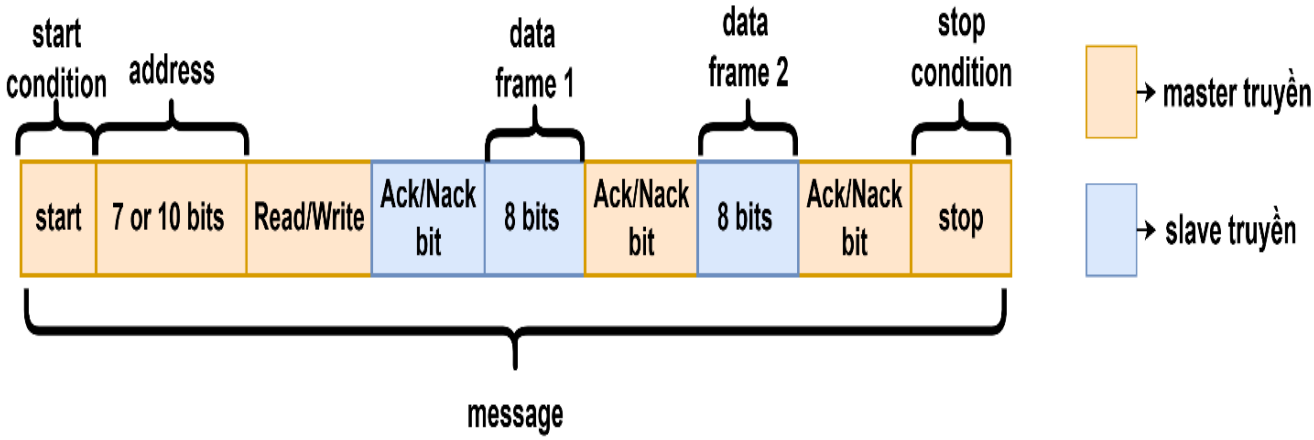
Hình 4: Minh họa quá trình truyền giao thức I2C

3.1 Trường hợp master muốn ghi dữ liệu đến slave:



Hình 5: Khung truyền dữ liệu cho thao tác ghi

3.2 Trường hợp master cần đọc dữ liệu từ slave:



Hình 6: Khung truyền dữ liệu cho thao tác đọc

4. Ưu và nhược điểm của giao thức I2C:

Ưu điểm:

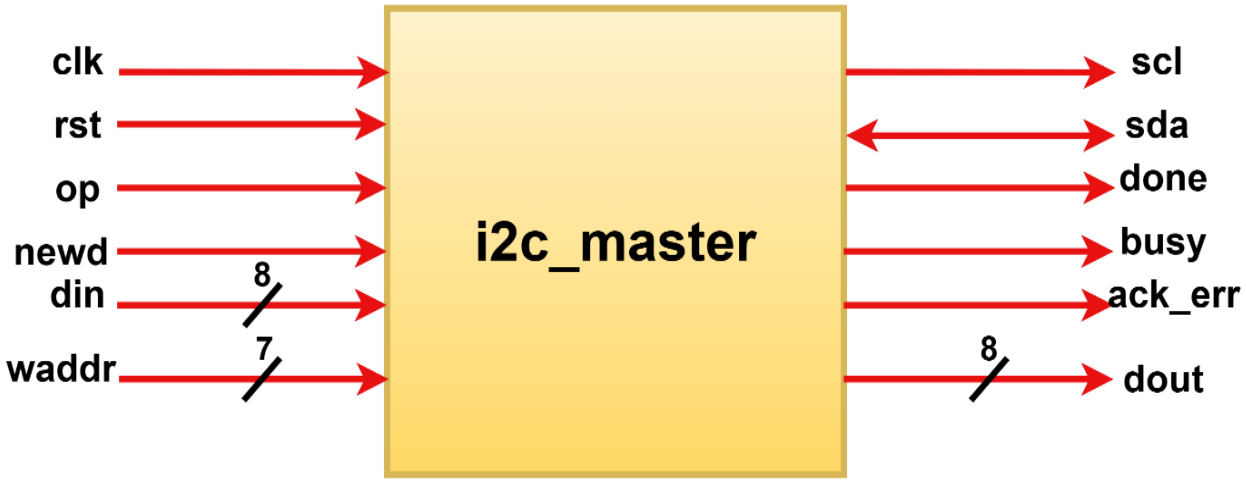
- Chỉ cần sử dụng 2 dây đã có thể thực hiện truyền dữ liệu.
- Giao tiếp được nhiều master và nhiều slave.
- Bit Ack/Nack xác nhận mỗi khung chuyển thành công.
- Là giao thức nổi tiếng và được sử dụng rộng rãi.

Nhược điểm:

- Tốc độ truyền thấp hơn khi so với SPI
- Giới hạn kích thước của khung truyền ở 8 bit.

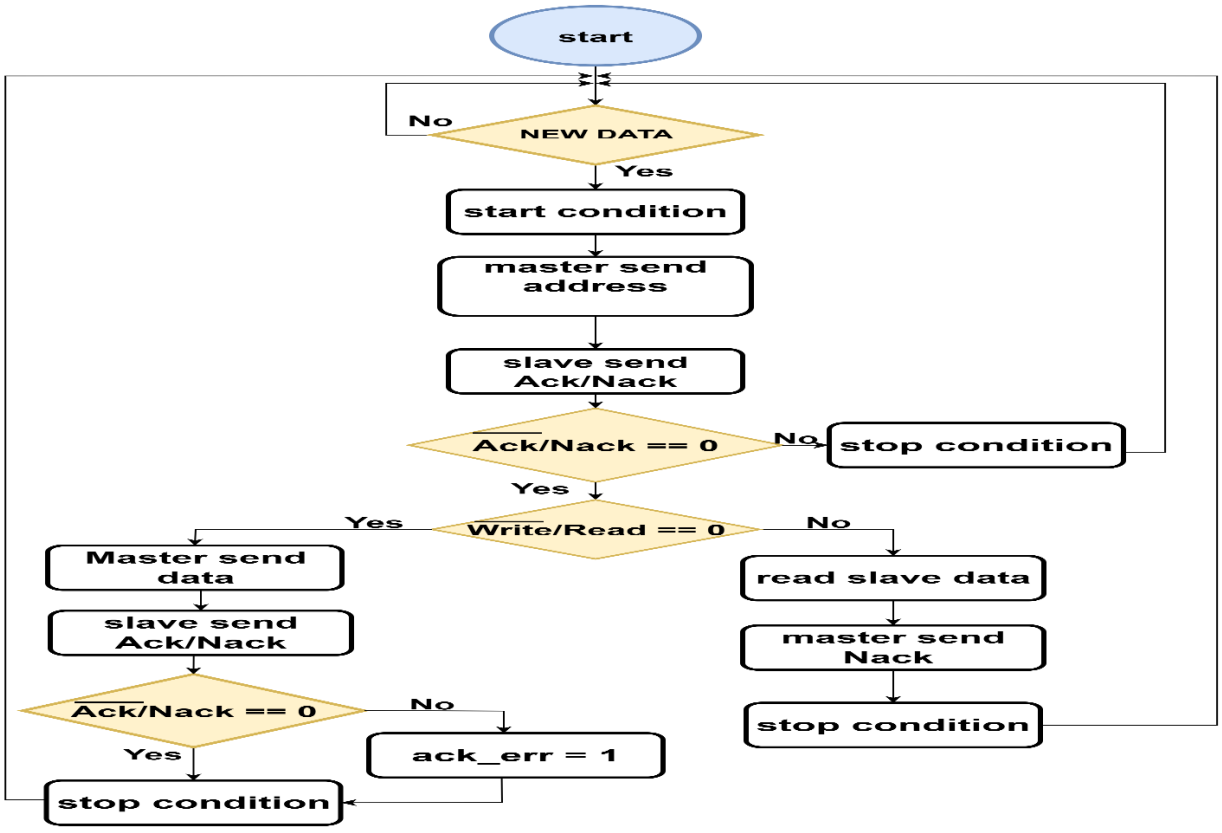
III. Thiết kế:

1. Sơ đồ khối:



Hình 7: Sơ đồ khối module giao tiếp I2C

2. Flowchart mô tả quá trình hoạt động của code Verilog:



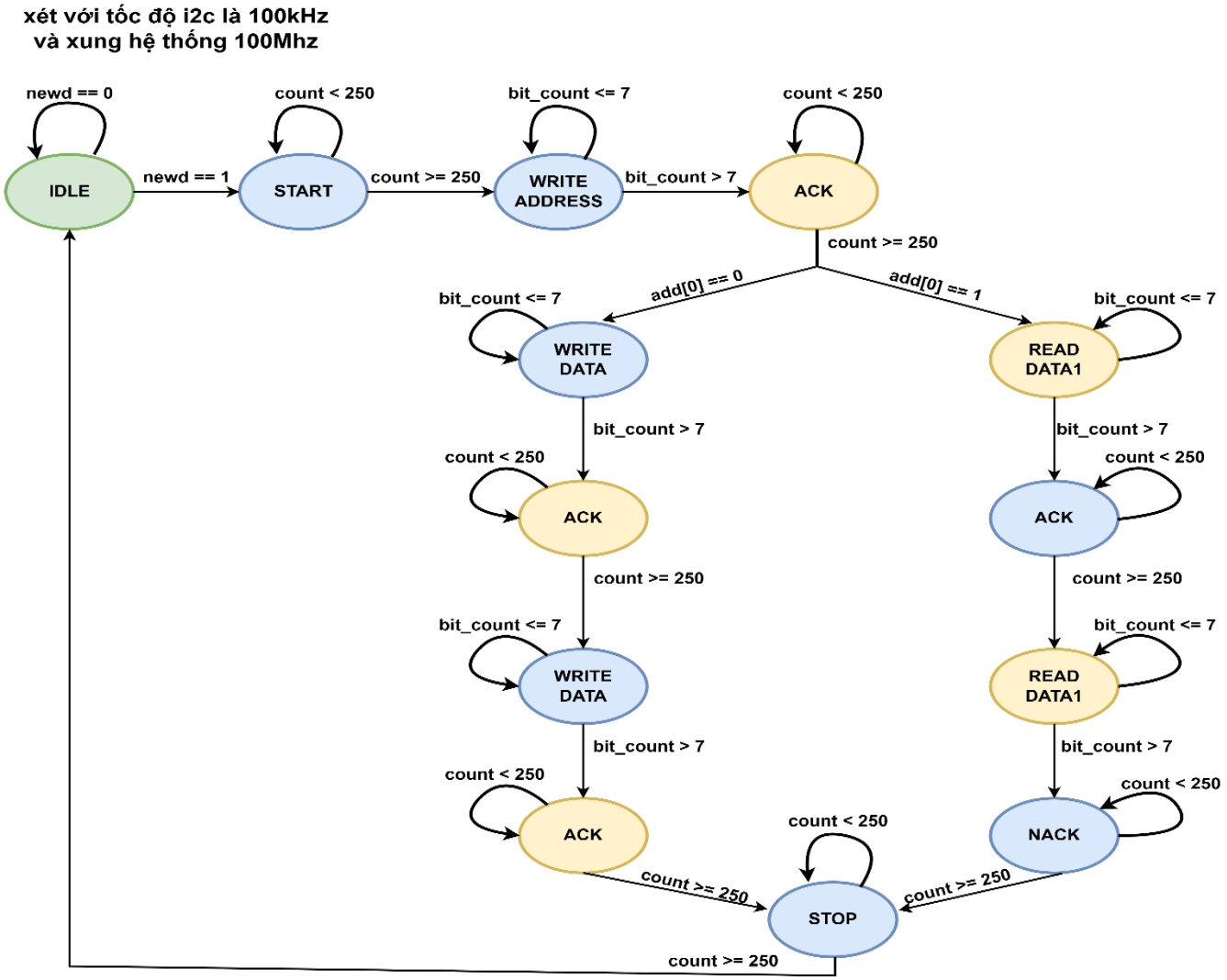
Hình 8: Flowchart mô tả quá trình hoạt động giao thức I2C

3. Bảng mô tả tín hiệu:

Tín hiệu	Hướng truyền	Chức năng
SCL	Master truyền đến slave	Serial Clock – xung đồng hồ do Master tạo, đồng bộ dữ liệu trên SDA
SDA	Cả 2 hướng	Serial Data – đường dữ liệu theo hai chiều, truyền bit theo SCL

Bảng 1: Bảng chú thích về hướng truyền và chức năng

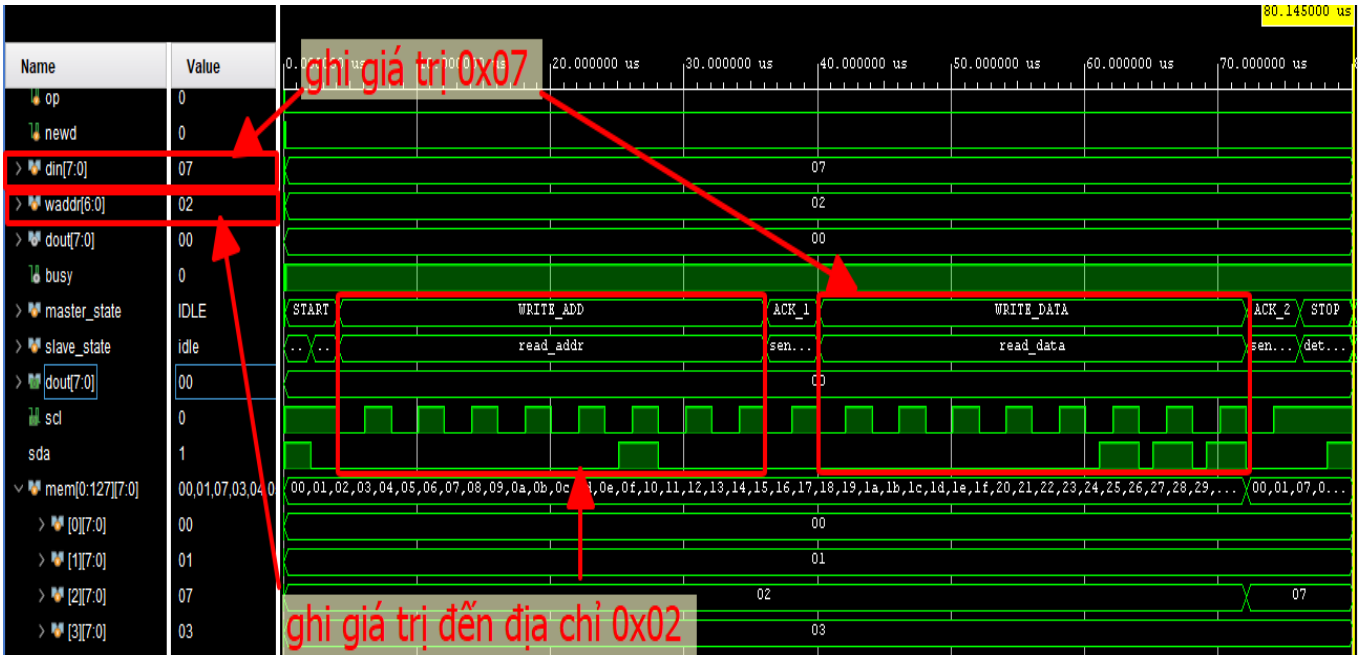
4. Sơ đồ FSM:



Hình 9: Sơ đồ FSM I2C hoạt động ở tốc độ 100kHz

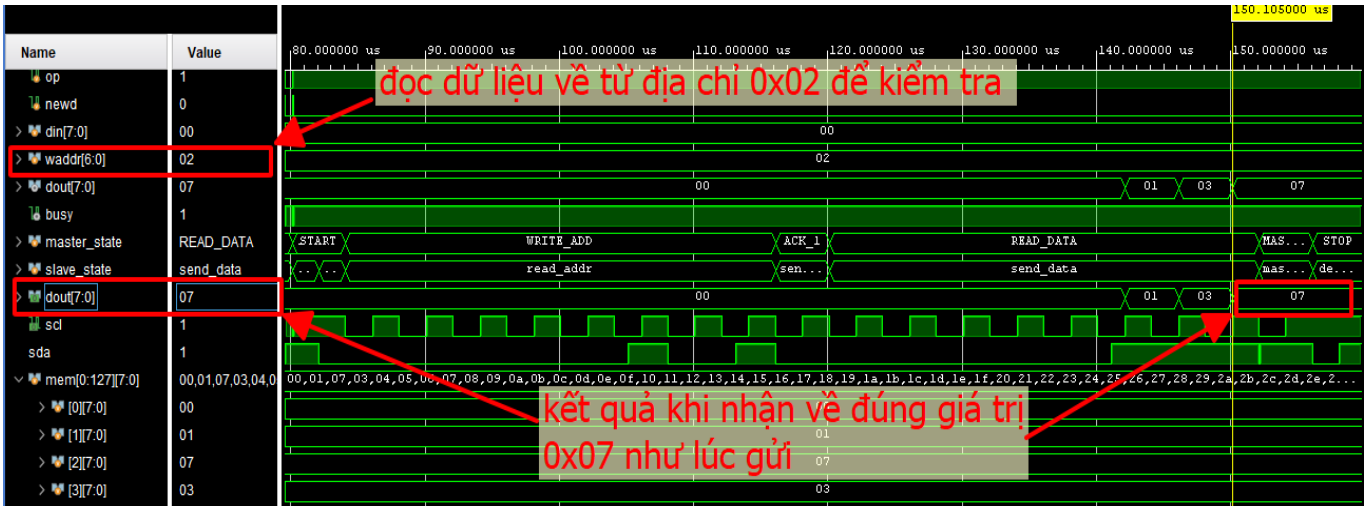
5. Mô phỏng truyền và nhận dữ liệu của master:

5.1. Mô phỏng quá trình master truyền dữ liệu đi:



Hình 10: **Mô phỏng quá trình master truyền dữ liệu đi**

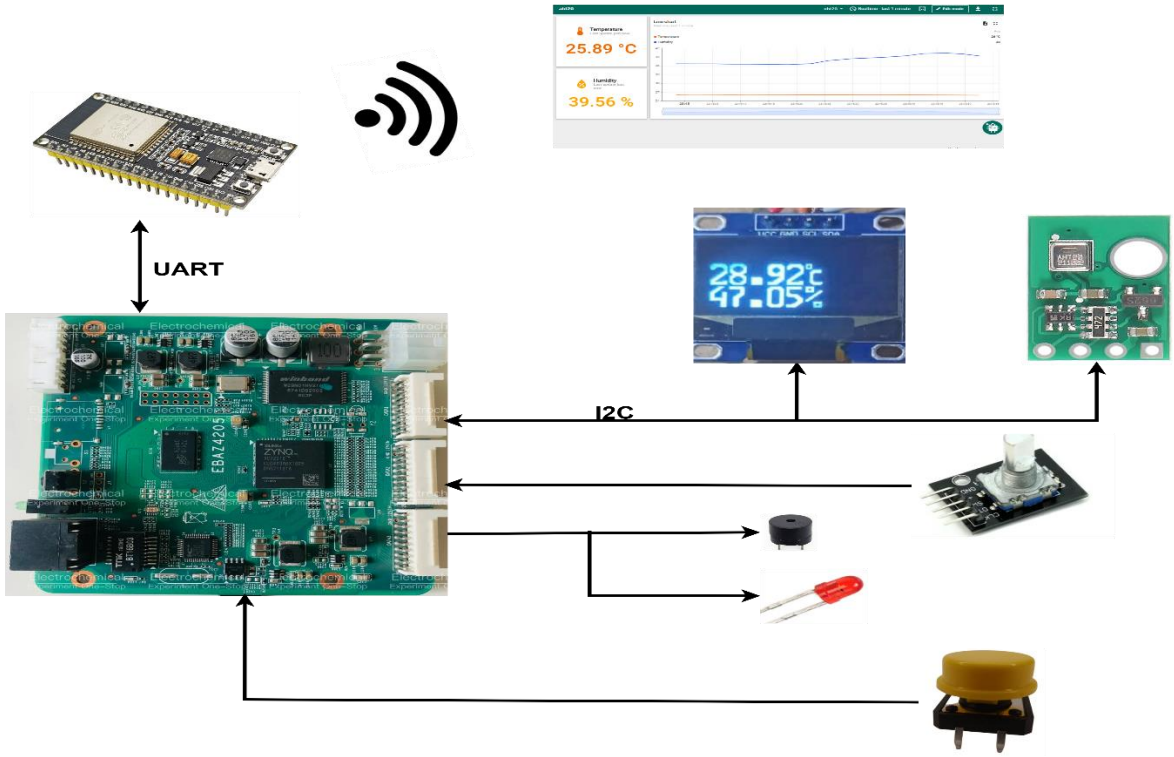
5.2. Mô phỏng quá trình master đọc dữ liệu về



Hình 11: Mô phỏng quá trình master đọc dữ liệu về

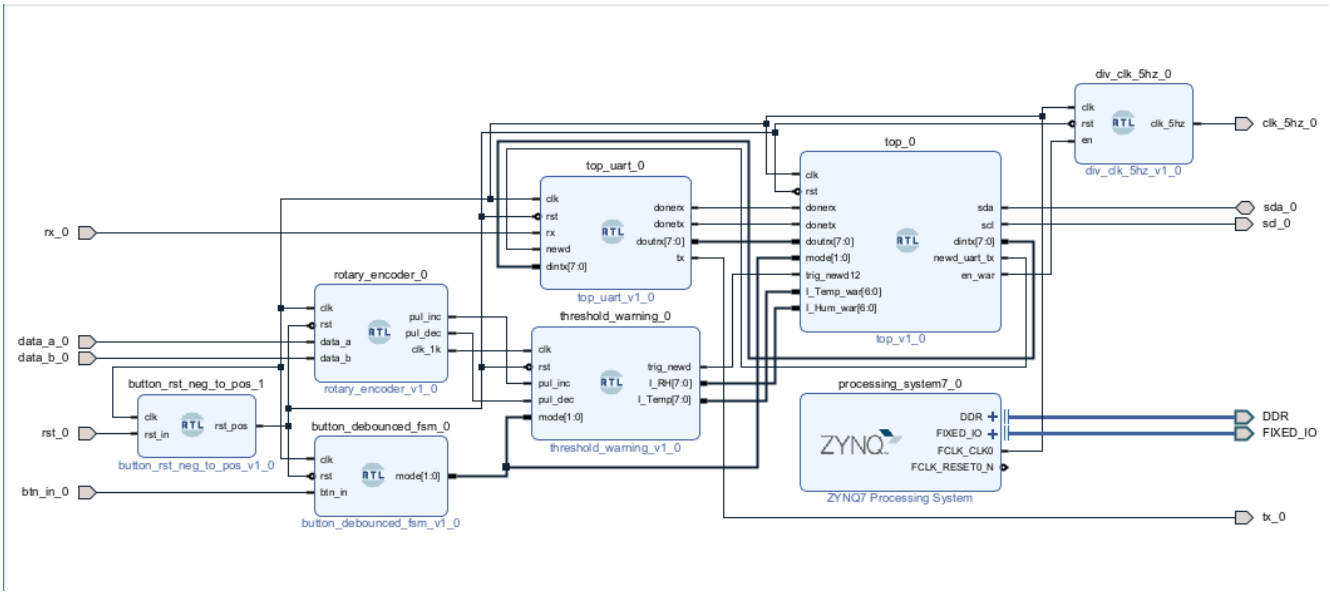
B. THIẾT KẾ SẢN PHẨM:

I. Sơ đồ kết nối các linh kiện:



Hình 12: Sơ đồ tổng quát của sản phẩm

II. Kết nối module trong Verilog:



Hình 13: Kết nối giữa các module trong Vivado

III. Mô tả hoạt động của sản phẩm:

Sản phẩm được phát triển với trọng tâm là thiết kế và lập trình trên FPGA bằng ngôn ngữ Verilog, nhằm khai thác khả năng xử lý song song và thời gian thực của FPGA trong các ứng dụng IoT nhúng. Hệ thống sử dụng giao thức I2C để giao tiếp với cảm biến môi trường AHT20, cho phép đọc dữ liệu nhiệt độ và độ ẩm với độ chính xác cao. Sau khi quá trình thu thập dữ liệu hoàn tất, thông tin đo được sẽ được hiển thị trực quan trên màn hình OLED I2C, giúp người dùng có thể theo dõi tức thời tại chỗ.

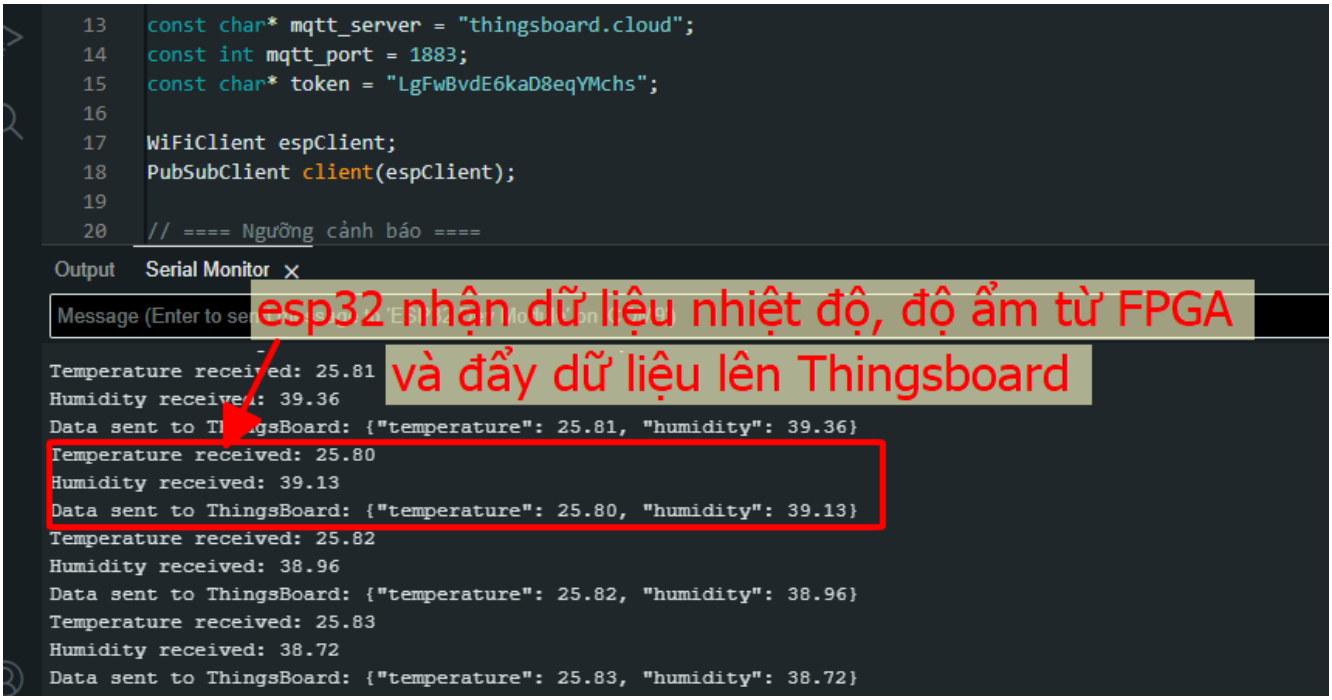
Bên cạnh đó, hệ thống còn tích hợp giao tiếp UART để truyền dữ liệu từ FPGA sang vi điều khiển ESP32. Vai trò của ESP32 là cầu nối, tiếp nhận dữ liệu cảm biến từ FPGA và đưa toàn bộ thông tin này lên nền tảng ThingsBoard thông qua kết nối Wi-Fi. Nhờ đó, người dùng có thể quan sát dữ liệu từ xa ở bất kỳ đâu, đồng thời theo dõi các biểu đồ trực quan về biến thiên nhiệt độ và độ ẩm theo thời gian.

Với hệ thống này còn được thiết kế để thực hiện cảnh báo khi nhiệt độ hoặc độ ẩm vượt mức cho phép. Cảnh báo được phát thông qua còi beep kết hợp với đèn led và biểu tượng cảnh báo hiển thị trên màn hình OLED giúp người dùng xác định được thành phần nào đang vượt mức cho phép. Để tối ưu độ linh hoạt trong các điều kiện môi trường khác nhau, sản phẩm được tích hợp bộ điều chỉnh Rotary Encoder, giúp hiệu chỉnh mức cảnh báo theo ý muốn và hỗ trợ trong các trường hợp khác nhau.

IV. Đánh giá thực tế:

Hình 14: AHT20 dùng đo nhiệt độ, độ ẩm qua giao thức i2c

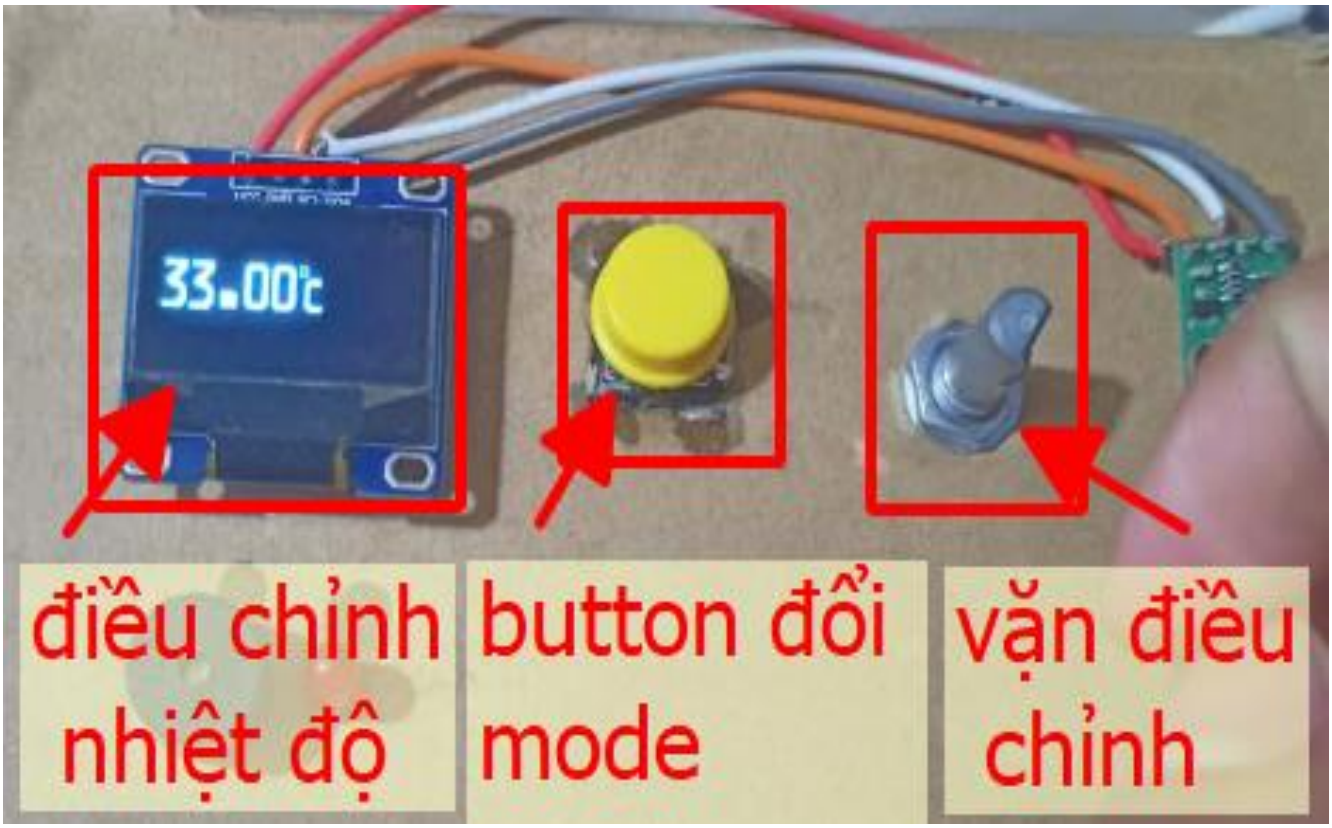
Hình 15: Đọc AHT20 hiển thị lên màn hình OLED



Hình 16: Truyền đến ESP32 qua UART

Hình 17: Hiển thị lên Thingsboard giám sát từ xa

Hình 18: Cảnh báo khi vượt mức cho phép



Hình 19: Chế độ điều chỉnh mức cảnh báo

V. Hướng phát triển và mở rộng:

Định hướng phát triển hệ thống trong tương lai tập trung vào việc mở rộng khả năng kết nối với nhiều loại cảm biến khác nhau như cảm biến ánh sáng, áp suất, khí gas nhằm gia tăng phạm vi ứng dụng trong IoT nhúng. Đồng thời, hệ thống sẽ được tích hợp thêm nhiều giao thức truyền thông như SPI, CAN hoặc Ethernet để đảm bảo tính linh hoạt và khả năng mở rộng. Bên cạnh đó, ứng dụng trí tuệ nhân tạo trực tiếp trên FPGA sẽ giúp phân tích dữ liệu tại chỗ, giảm độ trễ và giảm tải cho server. Ngoài ra, giao diện hiển thị cũng có thể được nâng cấp thành màn hình cảm ứng trực quan.

TÀI LIỆU THAM KHẢO:

[1] Shepherd Tutorials, "*Verilog HDL: VLSI Hardware Design Comprehensive Masterclass*". Truy cập ngày: 10/08/2025, tại: https://www.udemy.com/course/verilog-hdl-vlsi-hardware-design-comprehensive-masterclass/?couponCode=2021PM25

[2] Kumar Khandagle, "*Communication Series P1 : UART, SPI and I2C in Verilog*". Truy cập ngày: 10/08/2025, tại: https://www.udemy.com/course/communication-series-p1-uart-spi-and-i2c-in-verilog/?couponCode=PMNVD2525#instructor-1

[3] Guangzhou Aosong Electronics Co., Ltd. "*AHT20 - Temperature and Humidity Sensor Datasheet*". Truy cập ngày: 10/08/2025 tại: https://files.seeedstudio.com/wiki/Grove-AHT20_I2C_Industrial_Grade_Temperature_and_Humidity_Sensor/AHT20-datasheet-2020-4-16.pdf

[4] SOLOMON SYSTECH, "SSD1306". Truy cập ngày: 12/08/2025, tại: https://soldered.com/productdata/2022/03/Soldered_SSD1306_datasheet.pdf

[5] How To Mechatronics, "How Rotary Encoder Works and How To Use It with Arduino". Truy cập ngày: 18/08/2025, tại: https://youtu.be/v4BbSzJ-hz4?si=RZWPCIjmuhA2KcYd