

ĐẠI HỌC QUỐC GIA TP.HCM
TRƯỜNG ĐẠI HỌC BÁCH KHOA
KHOA ĐIỆN – ĐIỆN TỬ
BỘ MÔN ĐIỆN TỬ



ĐỒ ÁN MÔN HỌC
ĐO NHIỆT ĐỘ, ĐỘ ẨM BẰNG CẢM BIẾN DHT22
VÀ HIỂN THỊ TRÊN LCD VỚI ATMEGA324P

LỚP TN03 --- NHÓM 05 --- HK 242
GVHD: NGUYỄN LÝ THIÊN TRƯỜNG

STT	Họ và tên	MSSV	Đánh giá
1	Cao Tiến Dũng	2310537	33%
2	Nguyễn Đình Quyền	2312890	34%
3	Nguyễn Hoàng Anh Thắng	2313185	33%

THÀNH PHỐ HỒ CHÍ MINH – 6/2025

LỜI CẢM ƠN

Chúng em xin gửi lời cảm ơn đến Trường Đại Học Bách Khoa TP.HCM đã tạo điều kiện cho chúng em tiếp cận kiến thức một cách chủ động và sáng tạo, được có cơ hội làm các đề tài mang tính thực hiện như này thật sự là một trải nghiệm đáng quý và chắc chắn sẽ là hành trang bổ ích cho tương lai sau này.

Chúng em xin gửi lời cảm ơn chân thành đến giảng viên hướng dẫn thầy Nguyễn Lý Thiên Trường đã tận tâm quan tâm, giúp đỡ chúng em trong suốt quá trình nghiên cứu và hoàn thành bài báo cáo này. Thầy đã cung cấp cho chúng em đầy đủ các kiến thức cơ bản, nhờ đó việc tìm hiểu và làm đề tài này không còn khó khăn nữa.

Đối với chúng em, đề tài đồ án với cảm biến DHT22 lần này còn khá là mới mẻ và lạ lẫm nên không tránh khỏi những thiếu sót về sự hiểu biết, cũng như một số hạn chế về kiến thức nên chúng em mong nhận được sự chỉ bảo, đóng góp ý kiến của thầy để có thể nâng cao kiến thức và kinh nghiệm của bản thân để phục vụ trong quá trình học tập sau này.

Tp. Hồ Chí Minh, ngày 01 tháng 06 năm 2025

Nhóm 05 – TN03

TÓM TẮT ĐỒ ÁN

Đồ án này trình bày quá trình xây dựng một hệ thống đo và hiển thị nhiệt độ, độ ẩm môi trường sử dụng cảm biến DHT22, vi điều khiển ATmega324P, và màn hình LCD 16x2.

Trong quá trình thực hiện, đồ án vận dụng các kiến thức lý thuyết cơ bản về cảm biến, vi điều khiển và hiển thị, đồng thời sử dụng phần mềm Microchip Studio để lập trình cho ATmega324P và Proteus để mô phỏng phần cứng và kiểm thử hoạt động của hệ thống...

Đồ án gồm 6 chương chính được trình bày như sau:

1. Giới thiệu: Trình bày tổng quan đề tài, nhiệm vụ thực hiện và phân chia công việc trong nhóm.
2. Cơ sở lý thuyết: Giới thiệu vi điều khiển ATmega324P, cảm biến DHT22 nguyên lý giao tiếp 1 dây của DHT22, nguyên lý giao tiếp với LCD 16x2 ở chế độ 4-bit và chức năng ngắt Timer1 ở chế độ CTC.
3. Thiết kế và thực hiện phần cứng: Phân tích yêu cầu, thiết kế mạch, tính toán và xây dựng sơ đồ mạch chi tiết của hệ thống.
4. Thiết kế và thực hiện phần mềm: Phân tích yêu cầu phần mềm, lưu đồ giải thuật tổng quát và lưu đồ giải thuật chi tiết.
5. Kết quả thực hiện: Mô tả quá trình thực nghiệm mô phỏng trên Proteus và phân tích kết quả thu được.
6. Kết luận: Tổng kết những kết quả đạt được

MỤC LỤC

1. GIỚI THIỆU	1
1.1. Tổng quan	1
1.2. Nhiệm vụ đề tài.....	1
1.3. Phân chia nhiệm vụ.....	2
2. LÝ THUYẾT	3
2.1. Vi điều khiển ATmega324P và code Assembly VAR.....	3
2.2. Cảm biến nhiệt độ và độ ẩm DHT22	5
2.2.1 Tóm tắt quy trình giao tiếp tổng thể giữa MCU và cảm biến DHT22.....	6
2.2.2 MCU gửi tín hiệu Start Signal đến DHT22	7
2.2.3 Phản hồi của DHT22 khi nhận tín hiệu Start từ MCU	7
2.3. Sử dụng dữ liệu thu được xuất ra LCD.....	8
2.3.1. Các chân điều khiển: RS, RW, E.....	9
2.3.2. Kết nối LCD giao tiếp 4 bit	10
2.3.3. Các lệnh điều khiển LCD	11
2.3.4. Các bước hiển thị dữ liệu nhiệt độ và độ ẩm từ cảm biến DHT22 lên LCD 16x2 bằng chế độ 4-bit	12
2.4. Ngắt OCF1A, Timer 1, chế độ CTC.....	13
3. THIẾT KẾ VÀ THỰC HIỆN PHẦN CỨNG.....	14
3.1. Phân tích yêu cầu	14
3.1.1. Đo nhiệt độ môi trường	14
3.1.2. Đo độ ẩm môi trường.....	14
3.1.3. Hiển thị thông tin	14
3.1.4. Vi điều khiển xử lý	15
3.1.5. Nguồn cấp và điện trở kéo	15
3.1.6. Tần suất lấy mẫu	15
3.1.7. Môi trường hoạt động ổn định.....	15
3.2. Phân tích thiết kế.....	16
3.3. Tính toán và vẽ sơ đồ mạch chi tiết	17
3.3.1. Tính toán các thành phần chính	17
3.3.2. Sơ đồ mạch chi tiết	17

4. THIẾT KẾ VÀ THỰC HIỆN PHẦN MỀM.....	18
4.1. Yêu cầu đặt ra cho phần mềm.....	18
4.2. Yêu cầu kỹ thuật	18
4.3. Phân tích	18
4.4. Lưu đồ giải thuật tổng quát.....	19
4.5. Lưu đồ giải thuật chi tiết.....	21
5. KẾT QUẢ THỰC HIỆN	26
5.1. Cách thức đo đạc và thực nghiệm.....	26
5.1.1. Chuẩn bị linh kiện trong phần mềm mô phỏng.....	26
5.1.2. Kết nối các linh kiện	26
5.2. Phân tích kết quả thu được.....	28
6. KẾT LUẬN.....	29
TÀI LIỆU THAM KHẢO	30
PHỤ LỤC.....	31

DANH SÁCH HÌNH MINH HOẠ

Hình 2.1. Sơ đồ chân ATmega324P (nguồn: datasheet ATmega324P)	3
Hình 2.2. Các khối chức năng của ATmega324P (nguồn: datasheet ATmega324P).....	4
Hình 2.3. Module DHT22 và sơ đồ kết nối (nguồn: datasheet DHT22)	5
Hình 2.4. Quy trình giao tiếp tổng thể (Nguồn: datasheet DHT22)	6
Hình 2.5. MCU gửi tín hiệu bắt đầu và DHT22 phản hồi (Nguồn: datasheet DHT22) ...	7
Hình 2.6. DHT22 biểu diễn bit dữ liệu "0" (Nguồn: datasheet DHT22).....	7
Hình 2.7. DHT22 biểu diễn bit dữ liệu "1" (Nguồn: datasheet DHT22).....	8
Hình 2.8. LCD 16x2	8
Hình 2.9. Các chân và chức năng	9
Hình 2.10. Sơ đồ giao tiếp LCD (nguồn: Tài liệu Thí nghiệm Vi Xử Lý HCMUT)	10
Hình 2.11. Các lệnh của LCD	11
Hình 3.1. Sơ đồ khối tổng quát.....	16
Hình 3.2. Sơ đồ mạch chi tiết	17
Hình 5.1. Chuẩn bị linh kiện trong Proteus	26
Hình 5.2. Kết nối phần cứng giữa Atmega324P và DHT22 trên phần mềm Proteus.....	26
Hình 5.3. Kết nối phần cứng giữa Atmega324P và LCD 16x2.....	27
Hình 5.4. Kết nối phần cứng hoàn chỉnh trên phần mềm Proteus.....	27
Hình 5.5. Kết quả mô phỏng lần 1.....	28
Hình 8.1. Kết quả mô phỏng lần 2.....	31
Hình 8.2. Kết quả mô phỏng lần 3.....	31

DANH SÁCH BẢNG SỐ LIỆU

Bảng 4.1. Sơ đồ thuật toán luồng chính	19
Bảng 4.2. Sơ đồ thuật toán chương trình con xử lý ngắt.....	20
Bảng 4.3. Sơ đồ thuật toán cấu hình ngắt OCF1A, TIMER1	21
Bảng 4.4. Sơ đồ thuật toán xử lý dữ liệu của DHT22 gửi qua chân PD0	22
Bảng 4.5. Sơ đồ thuật toán tạo chương trình con read_next_8bit	23
Bảng 4.6. Sơ đồ thuật toán cấu hình LCD.....	24
Bảng 4.7. Sơ đồ thuật toán chương trình hiển thị nhiệt độ lên hàng 1	24
Bảng 4.8. Sơ đồ thuật toán hiển thị độ ẩm lên hàng 2.....	25
Bảng 5.1. Kết quả đo đạc và kiểm chứng lý thuyết qua 3 lần đo	28

1. GIỚI THIỆU

1.1. Tổng quan

Cảm biến DHT22 (hay còn gọi là AM2302) là một thiết bị đo nhiệt độ và độ ẩm kỹ thuật số, được sử dụng rộng rãi trong các ứng dụng đo lường môi trường. Hiện nay, cảm biến DHT22 được ứng dụng rộng rãi trong các nhà máy nhằm đảm bảo an toàn cho các quy trình sản xuất tự động. Qua đó ta thấy được tầm quan trọng cho việc ứng dụng DHT22 cho việc giám sát các quy trình sản xuất hiện nay.

Từ những vấn đề đó, nhóm em thực hiện đề tài “Đo nhiệt độ, độ ẩm môi trường sử dụng cảm biến DHT22” giao tiếp với MCU là ATmega324P và hiện kết quả lên LCD 16x2. Mục tiêu của đề tài nhằm tìm hiểu về nguyên lý hoạt động của DHT22 và ATmega324P và ứng dụng nhằm đo nhiệt độ môi trường, đồng thời đánh giá được hiệu quả làm việc của DHT22 cũng như giao tiếp giữa DHT22 và ATmega324P.

1.2. Nhiệm vụ đề tài

Sau đây là các nội dung chia nhỏ nhằm thực hiện đề tài một cách tối ưu hơn:

Nội dung 1: Tìm hiểu nguyên lý, lý thuyết về lần lượt là vi điều khiển ATmega324P, cảm biến DHT22, màn hình LCD 16x2.

Mục tiêu: Hiểu lý thuyết của VĐK, hiểu sâu về các chân giao tiếp của cảm biến DHT22 cũng như cách thức hoạt động của loại cảm biến này, tìm hiểu lý thuyết và hiểu từng chân giao tiếp của LCD 16x2.

Nội dung 2: Tìm hiểu sâu hơn về giao tiếp giữa MCU ATmega324P và DHT22 (cách truyền bit, mức cao, mức thấp nhận dạng ra sao,)

Mục tiêu: Hiểu thật kỹ hoạt động giao tiếp của 2 thiết bị này, từ đó ứng dụng được cách xử lý bit nhiệt độ và độ ẩm.

Nội dung 3: Phân tích thiết kế và phần cứng của MCU ATmega324P và DHT22

Mục tiêu: Hiểu rõ phần cứng, từng chân của thiết bị có chức năng như thế nào.

Nội dung 4: Vẽ sơ đồ khối chi tiết cho ý tưởng thuật toán.

Mục tiêu: Trình bày, lên ý tưởng thuật toán chi tiết bài toán của đề tài.

Nội dung 5: Thực hiện mô phỏng hóa ý tưởng bằng phần mềm Microchip Studio

Mục tiêu: Thực hiện ý tưởng thuật toán

Nội dung 6: Thực hiện kết nối trên phần mềm mô phỏng và mô phỏng

Mục tiêu: Kiểm chứng lý thuyết và thuật toán.

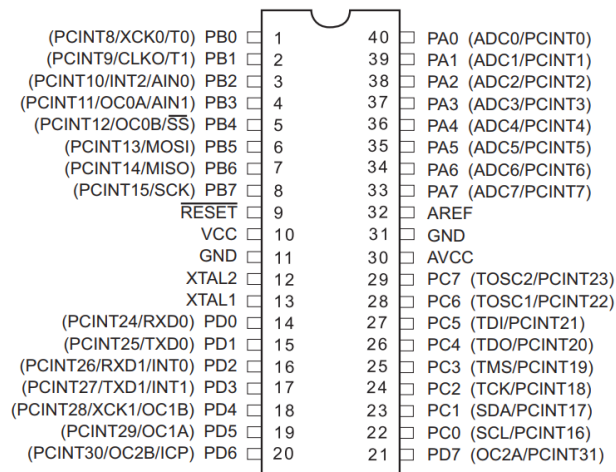
1.3. Phân chia nhiệm vụ

STT	Họ và tên	MSSV	Phân công	Hoàn thành
1	Cao Tiến Dũng	2310537	<p>Xác định tổng quan đề tài và các nhiệm vụ trọng tâm.</p> <p>Rút ra kết luận tổng thể cho đề tài.</p> <p>Hỗ trợ tìm hiểu lý thuyết và phần cứng.</p> <p>Phân tích và đánh giá kết quả mô phỏng.</p> <p>Hoàn thiện và trình bày báo cáo</p>	100%
2	Nguyễn Đình Quyền (nhóm trưởng)	2312890	<p>Phân tích và thiết kế phần mềm, bao gồm việc xây dựng các yêu cầu kỹ thuật và phân tích chi tiết các module phần mềm.</p> <p>Xây dựng lưu đồ giải thuật tổng quát và lưu đồ giải thuật chi tiết cho toàn bộ chương trình.</p> <p>Tổng hợp nội dung cuối cùng</p> <p>Quay video trình bày đề tài.</p>	100%
3	Nguyễn Hoàng Anh Thăng	2313185	<p>Tìm hiểu sâu về lý thuyết liên quan đến vi điều khiển ATmega324P, cảm biến DHT22, giao tiếp 1 dây của DHT22 và giao tiếp LCD.</p> <p>Phân tích chi tiết yêu cầu và thiết kế phần cứng của hệ thống, bao gồm sơ đồ mạch và tính toán các thành phần chính.</p>	100%

2. LÝ THUYẾT

2.1. Vi điều khiển ATmega324P và code Assembly VAR

ATmega324P là vi điều khiển 8-bit thuộc họ AVR của hãng Microchip, hoạt động với xung nhịp tối đa 20 MHz. Với kiến trúc RISC, vi điều khiển có thể thực thi hầu hết các lệnh trong một chu kỳ xung nhịp, giúp tối ưu hiệu năng xử lý. ATmega324P được trang bị bộ nhớ Flash 32KB, RAM 2KB và EEPROM 1KB, phù hợp cho các ứng dụng nhúng vừa và nhỏ như điều khiển tự động, hệ thống thu thập dữ liệu và giám sát môi trường.



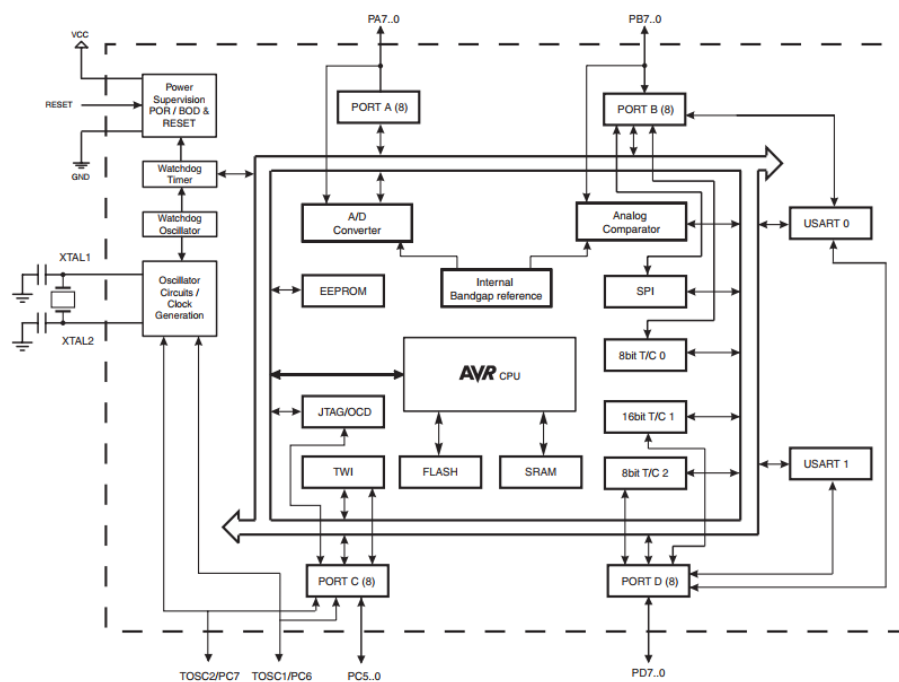
Hình 0.1. Sơ đồ chân ATmega324P (nguồn: datasheet ATmega324P)

Trong các ứng dụng giám sát môi trường, ATmega324P có thể giao tiếp với cảm biến nhiệt độ và độ ẩm để thu thập dữ liệu khí hậu xung quanh. Với khả năng xử lý nhanh và chính xác thời gian ở cấp độ vi giây, vi điều khiển này hoàn toàn đáp ứng được yêu cầu giao tiếp nghiêm ngặt về thời gian của , vốn truyền dữ liệu theo giao thức một dây dựa trên độ rộng xung. Ngoài ra, dung lượng bộ nhớ đủ lớn của ATmega324P cho phép lưu trữ và xử lý dữ liệu đọc từ cảm biến một cách hiệu quả, đồng thời dễ dàng tích hợp thêm các chức năng hiển thị, truyền thông hoặc điều khiển.

Code Assembly AVR là tập hợp các lệnh được viết bằng ngôn ngữ Assembly, một ngôn ngữ lập trình bậc thấp đặc thù dành riêng cho các dòng vi điều khiển thuộc họ AVR do hãng Atmel phát triển (hiện nay thuộc sở hữu của Microchip Technology). Mã Assembly hoạt động rất gần với ngôn ngữ máy, tức là ngôn ngữ mà bộ vi điều khiển trực tiếp hiểu và thực thi.

Khác với các ngôn ngữ lập trình bậc cao như C hay C++, code Assembly cho phép lập trình viên thao tác trực tiếp với phần cứng bên trong vi điều khiển. Các thao tác này bao gồm điều khiển các thanh ghi nội bộ, quản lý chân vào ra (I/O pins), cấu hình và sử dụng các ngoại vi tích hợp như bộ đếm thời gian (Timer), truyền thông nối tiếp (UART), chuyển đổi tương tự sang số (ADC), và nhiều thành phần khác.

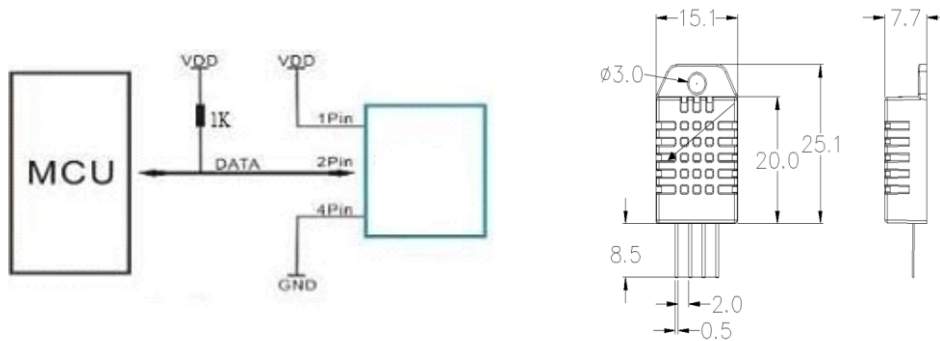
Một trong những ưu điểm nổi bật của ATmega324P là tích hợp nhiều giao tiếp ngoại vi như: Hai cổng USART cho truyền thông nối tiếp, giao tiếp SPI và TWI (I2C) cho kết nối với các IC ngoại vi, bộ chuyển đổi ADC 10-bit với nhiều kênh, ba bộ Timer đa chức năng, ngắt ngoài và ngắt nội, cho phép xử lý sự kiện tức thời như đọc encoder hoặc thời gian truyền nhận dữ liệu cảm biến. Bên cạnh đó, ATmega324P còn dễ dàng làm việc với các ngoại vi như cảm biến và module giao tiếp, giúp thực hiện các chức năng như đọc dữ liệu nhiệt độ, độ ẩm, truyền nhận thông tin với các thiết bị khác, điều khiển hoạt động của hệ thống, cũng như hiển thị thông tin lên các thiết bị như LED, LCD hoặc LED ma trận. Điều này giúp vi điều khiển trở thành một lựa chọn linh hoạt và hiệu quả trong nhiều ứng dụng nhúng thực tiễn.



Hình 0.2. Các khối chức năng của ATmega324P (nguồn: datasheet ATmega324P)

Trong đề tài này, ta chọn sử dụng màn hình LCD để hiển thị các dữ liệu đo được từ cảm biến, vì LCD cho phép trình bày thông tin một cách trực quan, dễ quan sát và tiện lợi cho người dùng. Cảm biến được chọn vì giao tiếp đơn giản qua chuẩn một dây, theo đúng tài liệu datasheet, giúp tiết kiệm chân I/O và dễ dàng tích hợp với vi điều khiển. Để đảm bảo việc đọc dữ liệu và cập nhật thông tin trên LCD được thực hiện chính xác và đều đặn, ta sử dụng Timer1 của vi điều khiển được cấu hình ở chế độ CTC (Clear Timer on Compare Match). Chế độ này tạo ra ngắt định kỳ với tần số ổn định, giúp đồng bộ quá trình lấy mẫu dữ liệu từ cảm biến như làm mới màn hình LCD kịp thời, từ đó nâng cao độ chính xác và độ tin cậy của hệ thống.

2.2. Cảm biến nhiệt độ và độ ẩm DHT22



Hình 0.3. Module DHT22 và sơ đồ kết nối (nguồn: datasheet DHT22)

Cảm biến nhiệt độ và độ ẩm DHT22 là một cảm biến tích hợp, có khả năng đo nhiệt độ và độ ẩm với đầu ra là tín hiệu số đã được hiệu chuẩn. Cảm biến DHT22 sử dụng nguồn điện một chiều từ 3V đến 5.5V. Sau khi cấp nguồn, ta cần chờ ít nhất 2 giây trước khi gửi bất kỳ lệnh nào đến cảm biến, để đảm bảo cảm biến có đủ thời gian ổn định. Để nguồn điện cấp cho cảm biến ổn định hơn và giảm nhiễu, có thể gắn thêm một tụ lọc 100nF giữa chân nguồn (VDD) và chân đất (GND). Khi dây kết nối ngắn hơn 20 mét, ta nên sử dụng một điện trở kéo lên (pull-up) giá trị khoảng 5KΩ để đảm bảo tín hiệu ổn định. Nếu dây dài hơn 20 mét, ta cần chọn điện trở pull-up có giá trị phù hợp tùy theo điều kiện thực tế, nhằm đảm bảo tín hiệu truyền đi không bị suy hao.

Dữ liệu nhiệt độ và độ ẩm từ cảm biến DHT22 được nhân với 10 để chuyển về dạng số nguyên, sau đó được truyền đến vi điều khiển dưới dạng 40 bit, 5 byte, với bit cao được gửi trước. Cấu trúc dữ liệu như sau: 2 byte đầu là giá trị của độ ẩm, 2 byte tiếp theo - biểu thị cho nhiệt độ (bit đầu tiên là dấu của nhiệt độ, 15 bit sau biểu thị độ lớn nhiệt độ). Byte cuối gồm 8 bit kiểm tra (checksum) dùng để xác minh tính hợp lệ của toàn bộ dữ liệu. Byte kiểm tra sẽ bằng tổng của 4 byte đầu tiên. Cách truyền dữ liệu này giúp vi điều khiển dễ dàng đọc và xử lý chính xác các giá trị đo được. Mỗi quá trình truyền dữ liệu kéo dài khoảng 4 mili giây.

Ví dụ, nếu nhiệt độ là $-30,5^{\circ}\text{C}$, độ ẩm là 80% thì DHT22 sẽ gửi giá trị như sau:

03 20 01 31 55 hay 0000 0011 0010 0000 1000 0001 0011 0001 1101 0101

Giải thích

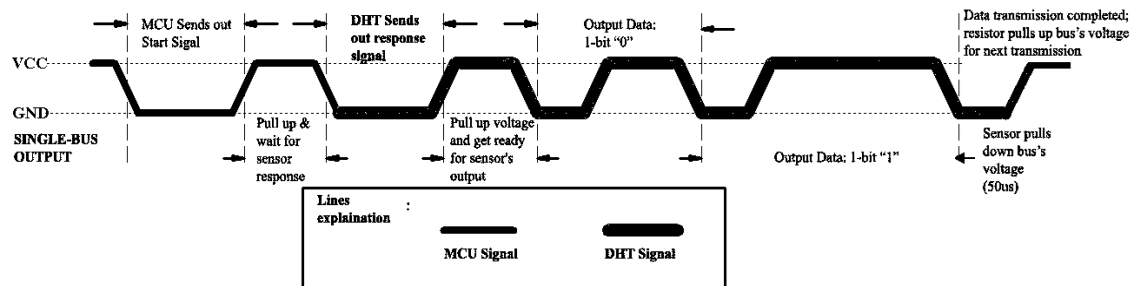
- 2 byte độ ẩm: $80 * 10 = 800_{10} = 03\ 20_{16} = 0000\ 0011\ 0010\ 0000$
- 2 byte nhiệt độ:
 - Độ lớn: $30,5 * 10 = 305_{10} = 01\ 31_{16} = x000\ 0001\ 0011\ 0001$ (15 bit biểu thị độ lớn)
 - Vì nhiệt độ là âm nên 16 bit nhiệt độ có dạng là 1xxx xxx xxxx xxxx
- 1 byte kiểm tra $1101\ 0101 = 000\ 0011 + 0010\ 0000 + 1000\ 0001 + 0011\ 0001$

Trong trường hợp nhiệt độ lớn hơn 0, ta có cấu trúc dữ liệu như sau:

- Byte 1: byte cao của độ ẩm
- Byte 2: byte thấp của độ ẩm
- Byte 3: byte cao của nhiệt độ
- Byte 4: byte thấp của nhiệt độ
- Byte 5: byte kiểm tra checksum

2.2.1 Tóm tắt quy trình giao tiếp tổng thể giữa MCU và cảm biến DHT22

Khi vi điều khiển (MCU) gửi tín hiệu bắt đầu (start signal), DHT22 sẽ chuyển từ chế độ tiêu thụ điện năng thấp sang chế độ hoạt động, và chờ MCU hoàn tất quá trình gửi tín hiệu bắt đầu. Sau khi tín hiệu bắt đầu được hoàn tất, DHT22 sẽ phản hồi bằng một tín hiệu dữ liệu dài 40 bit, chứa thông tin về độ ẩm và nhiệt độ gửi về cho MCU. Nếu không có tín hiệu bắt đầu từ MCU, DHT22 sẽ không phản hồi bất kỳ dữ liệu nào.

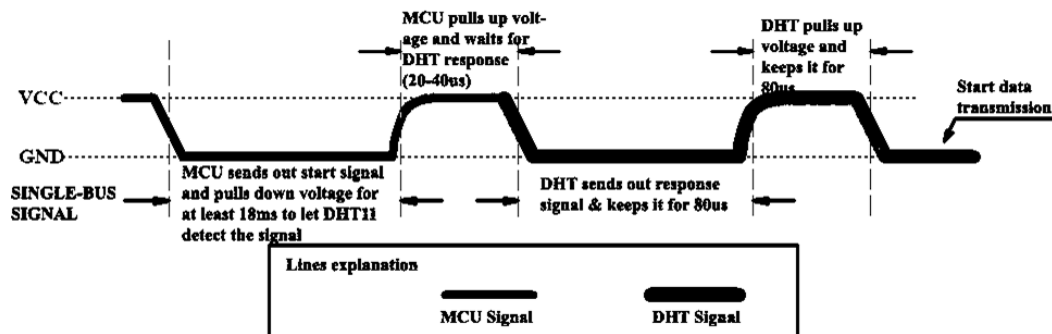


Hình 0.4. Quy trình giao tiếp tổng thể (Nguồn: datasheet DHT22)

Sau khi truyền dữ liệu xong, DHT22 sẽ trở lại chế độ tiết kiệm năng lượng, và sẽ chờ tín hiệu bắt đầu tiếp theo từ MCU để thực hiện lần truyền sau.

2.2.2. MCU gửi tín hiệu Start Signal đến DHT22

Khi chưa giao tiếp, đường truyền dữ liệu (Data Single-bus) ở mức điện áp cao. Khi bắt đầu giao tiếp giữa vi điều khiển (MCU) và cảm biến DHT22, chương trình trong MCU sẽ kéo điện áp của đường dữ liệu từ mức cao xuống mức thấp và giữ mức thấp này trong ít nhất 18ms để DHT22 có thể nhận biết tín hiệu khởi động từ MCU. Sau đó, MCU sẽ kéo điện áp trở lại mức cao và chờ trong khoảng 20–40 micro giây để DHT22 phản hồi.



Hình 0.5. MCU gửi tín hiệu bắt đầu và DHT22 phản hồi (Nguồn: datasheet DHT22)

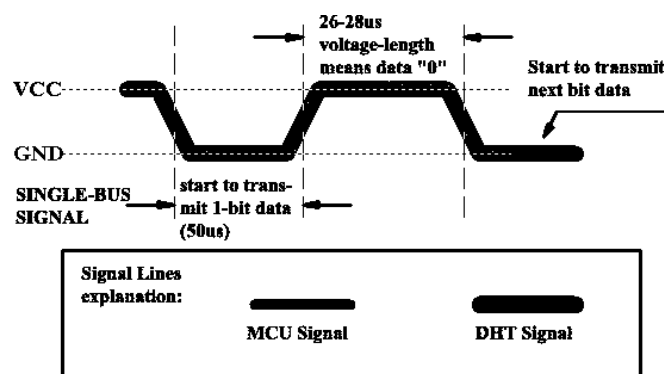
2.2.3 Phản hồi của DHT22 khi nhận tín hiệu Start từ MCU

Khi DHT22 phát hiện tín hiệu khởi động, nó sẽ phản hồi bằng một tín hiệu ở mức điện áp thấp, kéo dài trong 80 micro giây (80µs). Sau đó, DHT22 sẽ kéo điện áp của đường dữ liệu lên mức cao và giữ mức này trong 80µs để chuẩn bị gửi dữ liệu.

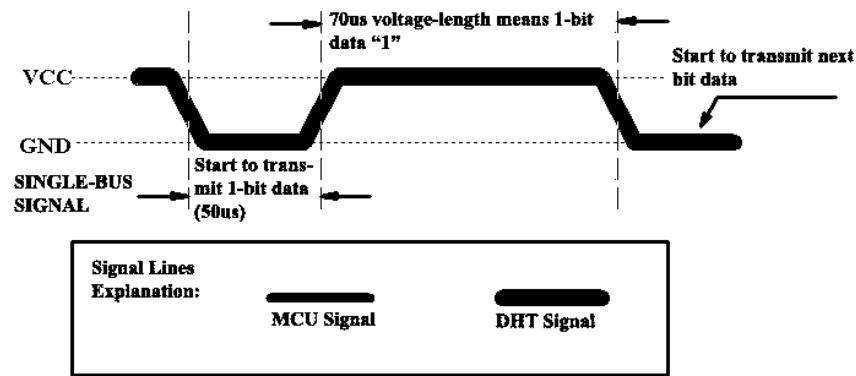
Khi đường dữ liệu (Data Single-bus) đang ở mức điện áp thấp, điều đó cho biết DHT22 đang gửi tín hiệu phản hồi. Ngay sau khi kết thúc tín hiệu phản hồi, DHT22 sẽ đưa điện áp lên mức cao trong 80µs để sẵn sàng bắt đầu quá trình truyền dữ liệu.

Khi truyền dữ liệu đến MCU, mỗi bit dữ liệu đều bắt đầu bằng một xung điện áp thấp kéo dài 50µs. Sau đó, độ dài của xung điện áp cao tiếp theo sẽ quyết định giá trị của bit:

- Nếu xung mức cao ngắn (khoảng 26 -28us) → bit dữ liệu là “0”
- Nếu xung mức cao dài (khoảng 70us) → bit dữ liệu là “1”



Hình 0.6. DHT22 biểu diễn bit dữ liệu "0" (Nguồn: datasheet DHT22)



Hình 0.7. DHT22 biểu diễn bit dữ liệu "1" (Nguồn: datasheet DHT22)

Nếu tín hiệu phản hồi từ DHT22 luôn ở mức cao, điều đó có nghĩa là cảm biến DHT22 không hoạt động đúng — cần kiểm tra lại dây nối hoặc kết nối giữa DHT22 và vi điều khiển (MCU).

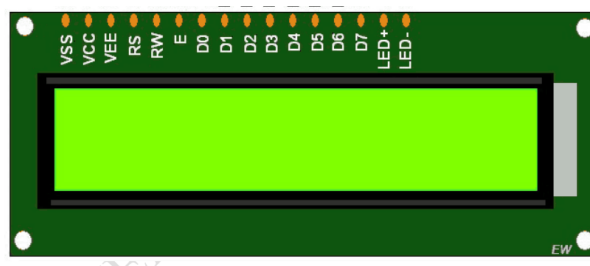
Sau khi gửi xong bit dữ liệu cuối cùng, DHT22 sẽ kéo đường dữ liệu xuống mức thấp và giữ ở đó trong khoảng 50 micro giây. Tiếp theo, điện trở kéo lên (pull-up resistor) sẽ đưa đường dữ liệu trở lại mức cao, tức là trở về trạng thái chờ để sẵn sàng cho lần giao tiếp tiếp theo.

Vì dữ liệu thu được từ cảm biến là dạng số nhị phân 16 bit (bao gồm 2 byte cho độ ẩm và 2 byte cho nhiệt độ), nên chúng ta tiến hành tách riêng từng byte và bit dấu nhiệt độ, sau đó kết hợp lại thành giá trị 16 bit. Tiếp theo, ta chia giá trị này cho 10 để đưa về dạng thập phân có 1 chữ số sau dấu phẩy, rồi hiển thị lên LCD.

2.3. Sử dụng dữ liệu thu được xuất ra LCD

Để hiển thị dữ liệu đã xử lý từ cảm biến lên màn hình, trước hết chúng ta cần hiểu về LCD và cách điều khiển nó. LCD (Liquid Crystal Display) hay màn hình tinh thể lỏng là thiết bị hiển thị rất phổ biến trong các hệ thống nhúng, dùng để hiển thị thông tin như giá trị cảm biến, trạng thái thiết bị, v.v.

Module LCD đã được thiết kế chuẩn để cho phép ta có thể giao tiếp với LCD do một hãng bất kỳ sản xuất với điều kiện là các LCD có sử dụng cùng IC điều khiển HD44780. Phần lớn các module LCD sử dụng giao tiếp 14 chân trong đó có 8 đường dữ liệu, 3 đường điều khiển và 3 đường cấp nguồn. Kết nối được bố trí dưới dạng 1 hàng 14 chân hoặc 2 hàng 7 chân.



Hình 0.8. LCD 16x2

2.3.1. Các chân điều khiển: RS, RW, E.

- RS (Register Select) – Chọn thanh ghi:
 - RS = 0: Truy cập thanh ghi lệnh (Instruction Register – IR), dùng để gửi lệnh điều khiển LCD như xóa màn hình, di chuyển con trỏ, chọn chế độ hiển thị, v.v.
 - RS = 1: Truy cập thanh ghi dữ liệu (Data Register – DR), dùng để gửi dữ liệu hiển thị (các ký tự) lên màn hình.
- RW (Read/Write) – Chọn chế độ đọc/ghi:
 - RW = 0: Chế độ ghi (Write) – vi điều khiển gửi lệnh hoặc dữ liệu đến LCD.
 - RW = 1: Chế độ đọc (Read) – vi điều khiển đọc dữ liệu từ LCD (ít dùng, thường RW luôn nối GND để chỉ ghi).
- E (Enable) – Tín hiệu kích hoạt truyền dữ liệu:
 - Cạnh xuống (E từ 1 → 0): Cho phép ghi lệnh hoặc dữ liệu vào LCD.
 - Cạnh lên (E từ 0 → 1): Cho phép đọc từ LCD (nếu RW = 1).

Mỗi lần truyền dữ liệu/lệnh, cần tạo một xung E để LCD chốt và thực hiện thao tác. Các chân data/command: D0 – D7.

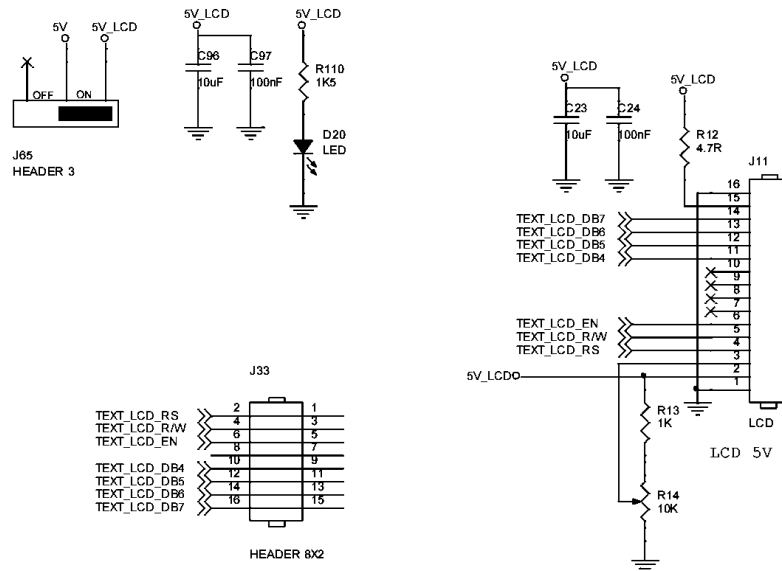
Chân	Ký hiệu	Chức năng	Mô tả
1	VSS	Nguồn	GND
2	VDD	Nguồn	Nguồn(5V)
3	VEE	Điều khiển	Điều chỉnh độ tương phản, thường mắc qua chiết áp thay đổi từ 0-5V
4	RS	Điều khiển	RS=0: truy xuất lệnh RS=1: truy xuất data
5	RW	Điều khiển	RW=1: truy xuất đọc RW=0: truy xuất ghi
6	E	Điều khiển	E=↓ cho phép ghi LCD E=↑ cho phép đọc LCD
7-14	D0 - D7	Data/ Command	RS=1: data RS=0: lệnh
15	LEDA	Anode LED	Anode LED nền
16	LEDK	Cathode LED	Cathode LED nền

Hình 0.9. Các chân và chức năng

Với cấu hình này, LCD có thể dễ dàng hiển thị dữ liệu từ cảm biến hoặc các giá trị cần theo dõi trong hệ thống.

2.3.2. Kết nối LCD giao tiếp 4 bit

Trong đề tài này, ta sử dụng loại LCD 16x2, tức là màn hình có 2 dòng, mỗi dòng hiển thị được tối đa 16 ký tự. LCD 16x2 sử dụng IC điều khiển HD44780, hỗ trợ giao tiếp với vi điều khiển ở hai chế độ: 8-bit và 4-bit. Để tiết kiệm chân I/O trên vi điều khiển, ta chọn chế độ giao tiếp 4-bit, chỉ cần sử dụng 4 chân dữ liệu (D4 đến D7).



Hình 0.10. Sơ đồ giao tiếp LCD (nguồn: Tài liệu Thí nghiệm Vi Xử Lý HCMUT)

2.3.3. Các lệnh điều khiển LCD

Để giao tiếp với module LCD, ta được điều khiển thông qua một tập lệnh chuẩn. Bảng sau tóm tắt các lệnh điều khiển LCD.

Lệnh	Chức năng
0F	LCD bật, con trỏ bật, con trỏ nhấp nháy bật
01	Xoá toàn màn hình
02	Quay về màn hình chính
04	Giảm con trỏ
06	Tăng con trỏ
0E	Màn hình bật, con trỏ nhấp nháy tắt
80	Bắt con trỏ trở về vị trí đầu tiên của hàng 1
C0	Bắt con trỏ trở về vị trí đầu tiên của hàng 2
38	Sử dụng 2 hàng và ma trận 5x7
83	Con trỏ hàng 1 vị trí 3
3C	Kích hoạt dòng 2
08	Tắt màn hình hiển thị và con trỏ
C1	Nhảy đến dòng 2 vị trí 1
0C	Bật màn hình hiển thị, tắt con trỏ
C2	Nhảy đến hàng 2, vị trí 2
0C	Display ON, cursor OFF
C1	Jump to second line, position 1
C2	Jump to second line, position 2

Hình 0.11. Các lệnh của LCD

Để gửi một lệnh ra LCD, ta đặt mã lệnh lên các tín hiệu DATA, tín hiệu RS = 0 để chọn thanh ghi lệnh, tín hiệu RW bằng 0 để chọn chế độ ghi, tín hiệu EN ban đầu bằng 0. Sau đó, ta tạo 1 xung trên EN bằng cách đưa tín hiệu EN lên 1 và xuống 0 để ghi vào thanh ghi lệnh.

Để ghi dữ liệu ra LCD để hiển thị, ta đặt dữ liệu lên các tín hiệu DATA, tín hiệu RS = 1 để chọn thanh ghi data, tín hiệu RW bằng 0 để chọn chế độ ghi, tín hiệu EN ban đầu bằng 0. Sau đó, ta tạo 1 xung trên EN bằng cách đưa tín hiệu EN lên 1 và xuống 0 để ghi vào thanh ghi data.

Để biết được LCD có sẵn sàng để nhận lệnh hay không, ta có thể đọc thanh ghi lệnh. Tín hiệu RS = 0 để chọn thanh ghi lệnh, tín hiệu RW bằng 1 để chọn chế độ đọc, tín hiệu EN ban đầu bằng 0. Sau đó, ta tạo 1 xung trên EN bằng cách đưa tín hiệu EN lên 1 và xuống 0. Khi đó dữ liệu của LCD sẽ xuất ra trên các chân dữ liệu.

Ở chế độ 4 bit (được kết nối như trên kit thí nghiệm), dữ liệu mỗi lần truyền/nhận là 1 nibble 4 bit, với nibble cao được truyền/nhận trước. Vì vậy, mỗi quá trình ghi/đọc dữ liệu cần 2 xung EN. Giảm độ xung sau thể hiện quá trình ghi/đọc LCD.

2.3.4. Các bước hiển thị dữ liệu nhiệt độ và độ ẩm từ cảm biến DHT22 lên LCD 16x2 bằng chế độ 4-bit

Sau khi thu thập và xử lý dữ liệu từ cảm biến (ví dụ như nhiệt độ hoặc độ ẩm), vi điều khiển sẽ hiển thị kết quả lên màn hình LCD 16x2 để người dùng dễ quan sát. Việc hiển thị dữ liệu lên LCD gồm các bước sau:

Thứ nhất, ta khởi tạo LCD: LCD được thiết lập làm việc ở chế độ 4-bit để tiết kiệm chân I/O. Quá trình khởi tạo bao gồm gửi các lệnh như chọn chế độ hiển thị 2 dòng, bật màn hình, tắt con trỏ và xóa màn hình. Các lệnh này được gửi thông qua các lệnh điều khiển ($RS = 0$, $RW = 0$) và thực hiện trong chương trình.

Thứ hai, ta chuyển dữ liệu nhị phân sang ký tự ASCII: Dữ liệu đọc được từ cảm biến ở dạng nhị phân (binary), không thể hiển thị trực tiếp. Do đó, cần chuyển đổi sang dạng số thập phân, sau đó cộng với mã ASCII của ký tự '0' (0x30) để có thể hiển thị trên LCD.

Thứ ba, đưa dữ liệu ra LCD: Sau khi chuyển đổi, các ký tự ASCII sẽ được gửi từng ký tự một đến LCD ($RS = 1$, $RW = 0$). Mỗi ký tự được truyền theo giao thức 4-bit: gửi 4 bit cao trước, sau đó gửi 4 bit thấp, kèm theo xung EN để LCD nhận dữ liệu.

Thứ tư, hiển thị thông tin định dạng: Ngoài dữ liệu chính, chương trình cũng hiển thị các ký tự mô tả như "T=" cho nhiệt độ hay "H=" cho độ ẩm để người dùng dễ hiểu. Các ký tự này được gửi trực tiếp bằng cách sử dụng chương trình con nhiều lần với từng mã ASCII tương ứng.

2.4. Ngắt OCF1A, Timer 1, chế độ CTC

Ngắt (Interrupt) là một cơ chế cho phép vi điều khiển tạm thời dừng chương trình chính để xử lý một sự kiện khẩn cấp hoặc ưu tiên cao, sau đó quay lại thực hiện tiếp phần còn lại của chương trình chính. Việc sử dụng ngắt giúp hệ thống phản ứng nhanh chóng và hiệu quả với các tín hiệu ngoại vi hoặc nội bộ, đồng thời giảm thiểu việc kiểm tra liên tục (polling) gây lãng phí tài nguyên. ATmega324P hỗ trợ nhiều loại ngắt như: ngắt ngoại (INT0, INT1), ngắt từ các bộ Timer/Counter (ngắt tràn, ngắt so sánh), ngắt truyền thông (USART, SPI, TWI), ngắt ADC... Khi xảy ra ngắt, vi điều khiển sẽ lưu lại trạng thái hiện tại, chuyển sang thực hiện hàm xử lý ngắt (ISR), sau đó quay lại chương trình chính sau khi xử lý xong. Để đảm bảo việc đọc dữ liệu từ cảm biến diễn ra đều đặn mỗi 2 giây mà không làm gián đoạn các chức năng khác của hệ thống, ta sử dụng ngắt từ Timer1, cấu hình vectơ ngắt tại 0x1A.

Cụ thể, Timer1 được cấu hình hoạt động ở chế độ CTC (Clear Timer on Compare Match). Trong chế độ này, bộ đếm của Timer1 sẽ tự động được đặt về 0 khi đạt đến giá trị so sánh được lập trình trước đó tại thanh ghi OCR1A. Việc này tạo ra một chu kỳ lặp ổn định mà không cần can thiệp thủ công từ chương trình chính. Trong chương trình, giá trị OCR1A được đặt là 15624, tương ứng với khoảng thời gian 2 giây (giả sử tần số xung hệ thống là 8MHz và prescaler của Timer1 được chọn là 1024, ta có: $8.000.000 / 1024 = 15624$, tức mỗi 15624 chu kỳ sẽ là gần đúng 2 giây).

Khi Timer1 đạt đến OCR1A, nó sẽ tự động tạo ra một ngắt so sánh (Compare Match A – TIMER1_COMPA). Trình xử lý ngắt (ISR) tương ứng sẽ được gọi ngay lập tức mà không cần phải đợi chương trình chính phản ứng. Trong ISR này, chương trình thực hiện toàn bộ quá trình đọc tín hiệu từ cảm biến (gửi tín hiệu bắt đầu, đợi phản hồi, đọc 40 bit dữ liệu), sau đó xử lý dữ liệu (tách nhiệt độ, độ ẩm, kiểm tra checksum), thực hiện chuyển đổi dữ liệu nhị phân sang dạng thập phân, rồi tiếp tục chuyển sang mã ASCII để có thể hiển thị trên màn hình LCD.

Việc thực hiện các thao tác này trong ISR đảm bảo rằng dữ liệu luôn được cập nhật chính xác theo đúng chu kỳ 2 giây, đồng thời không ảnh hưởng tới các phần còn lại của chương trình chính. Điều này rất quan trọng trong các ứng dụng có nhiều nhiệm vụ song song như đọc phím nhấn, xử lý giao tiếp UART, hay điều khiển thiết bị ngoại vi. Có thể nói, cơ chế ngắt từ Timer1 giúp chương trình hoạt động hiệu quả, tiết kiệm tài nguyên CPU và đảm bảo tính định kỳ một cách chính xác, ổn định trong việc thu thập và hiển thị dữ liệu từ các cảm biến môi trường.

3. THIẾT KẾ VÀ THỰC HIỆN PHẦN CỨNG

3.1. Phân tích yêu cầu

Hệ thống cần thực hiện các chức năng sau:

- Đo nhiệt độ và độ ẩm môi trường bằng cảm biến DHT22, qua giao tiếp 1-Wire (giao tiếp độc quyền của DHT22, không phải giao tiếp 1-Wire của Dallas).
- Hiển thị dữ liệu đo được lên LCD 16x2:
- Hàng 1: Nhiệt độ, lấy 1 số lẻ (ví dụ: `T: 28.3C`).
- Hàng 2: Độ ẩm, làm tròn đến phần nguyên (ví dụ: `RH: 67%`)

3.1.1. Đo nhiệt độ môi trường

Cảm biến sử dụng: DHT22 (AM2302)

Dải đo nhiệt độ: từ -40°C đến +80°C

Độ phân giải: 0.1°C

Độ chính xác (Accuracy): Nhỏ hơn $\pm 0.5^{\circ}\text{C}$ (trong khoảng nhiệt độ thông thường)

Độ lặp lại (Repeatability): $\pm 0.2^{\circ}\text{C}$

Hiển thị yêu cầu: Định dạng: "T: XX.X°C" (1 chữ số thập phân) hoặc "T: -XX.X°C" nếu nhiệt độ là âm, cập nhật định kỳ 1 lần mỗi 2 giây.

3.1.2. Đo độ ẩm môi trường

Dải đo độ ẩm: 0%RH đến 100%RH

Độ phân giải: 0.1%RH

Độ chính xác (Accuracy): $\pm 2\%RH$ trong điều kiện chuẩn, lệch tối đa $\pm 5\%RH$ theo datasheet

Độ lặp lại (Repeatability): $\pm 1\%RH$

Độ trễ (Hysteresis): $\pm 0.3\%RH$

Ổn định lâu dài (Long-Term Stability): $\pm 0.5\%RH/\text{năm}$

Hiển thị yêu cầu: Làm tròn số nguyên (hiển thị độ ẩm % không có số thập phân), định dạng: "RH: XX%" hoặc RH: 100% nếu độ ẩm là 100%, cập nhật định kỳ 1 lần mỗi 2 giây cùng với nhiệt độ.

3.1.3. Hiển thị thông tin

Thiết bị hiển thị: LCD 16x2 (16 ký tự, 2 dòng), chế độ giao tiếp: 4-bit (tiết kiệm chân I/O)

Thông tin hiển thị:

Dòng 1: "T: [nhiệt độ] °C", [nhiệt độ] là nhiệt độ môi trường đo được, lấy một số lẻ

Dòng 2: "RH: [độ ẩm]%", [độ ẩm] là độ ẩm môi trường đo được, làm tròn đến phần nguyên

Tần suất cập nhật màn hình: 1 lần mỗi 2 giây

3.1.4. Vi điều khiển xử lý

Sử dụng vi điều khiển: ATmega324P

Tần số hoạt động: từ 8 MHz đến 20 MHz

Chân kết nối cảm biến: Chân data DHT22 kết nối vào PD0 để giao tiếp

Chân điều khiển LCD:

- RS \rightarrow PA0
- RS \rightarrow PA1
- E \rightarrow PA2
- D4–D7 \rightarrow PA4–PA7

3.1.5. Nguồn cấp và điện trở kéo lên

Nguồn hoạt động cho toàn bộ hệ thống: 5V DC ổn định

Nguồn hoạt động của cảm biến DHT22: từ 3.3V đến 6V DC, tương thích với 5V MCU

Điện trở kéo lên: Một điện trở khoảng 5 k Ω nối giữa chân Data của DHT22 và Vcc

3.1.6. Tần suất lấy mẫu

Thời gian tối thiểu giữa 2 lần đọc: ≥ 2 giây (theo đặc tả của DHT22)

Tần suất lấy mẫu thiết kế: 1 lần mỗi 2 giây, đảm bảo ổn định và không vi phạm thông số kỹ thuật

3.1.7. Môi trường hoạt động

Vì chạy mô phỏng nên ta chọn dải hoạt động của thông số ở mức tối đa, hay nhiệt độ môi trường hệ thống hoạt động từ -40°C đến 80°C, độ ẩm môi trường hệ thống từ 0%RH đến 100%RH.

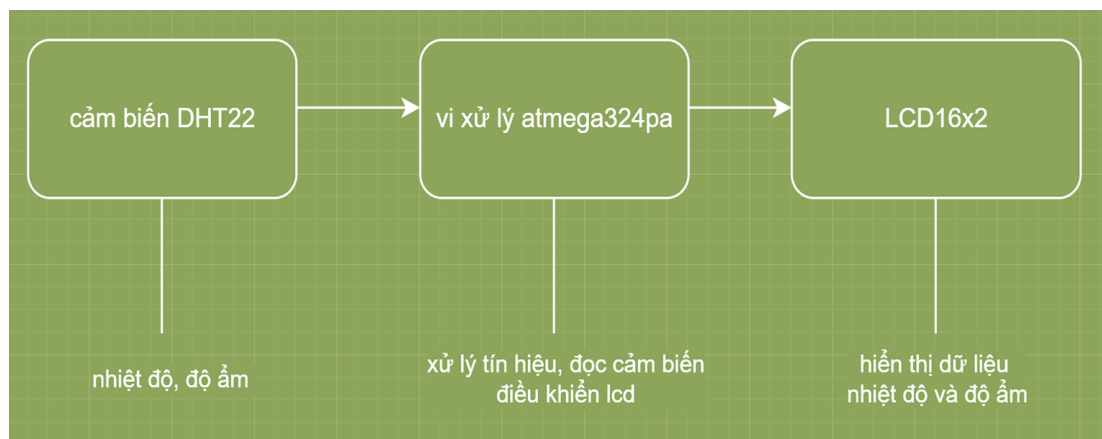
3.2. Phân tích thiết kế

Hệ thống sử dụng vi điều khiển ATmega324P để giao tiếp với cảm biến DHT22 qua giao thức 1-wire. Dữ liệu 40-bit từ cảm biến sẽ được tách thành 2 giá trị: nhiệt độ và độ ẩm. Kết quả sau xử lý sẽ được hiển thị trực tiếp lên LCD 16x2, dòng 1 là nhiệt độ, dòng 2 là độ ẩm.

Ưu điểm: Thiết kế độc lập, không cần máy tính hỗ trợ. Giao diện LCD trực quan, dễ sử dụng ngoài thực tế. Dễ dàng tùy biến, mở rộng tính năng hiển thị hoặc xử lý số liệu. Cho phép định dạng kết quả hiển thị linh hoạt (ví dụ: hiển thị 1 chữ số sau dấu phẩy).

Nhược điểm: Lập trình tương đối phức tạp, đặc biệt là phần đọc dữ liệu DHT22 đòi hỏi chính xác về thời gian (timing). Cần thêm phần cứng LCD và chiếm nhiều chân I/O nếu không tối ưu

Sơ đồ khối tổng quát



Hình 0.1. Sơ đồ khối tổng quát

Cảm biến DHT22: Cảm biến đo nhiệt độ và độ ẩm môi trường, xuất tín hiệu số theo giao thức 1-wire. Nhiệm vụ gửi dữ liệu 40-bit chứa thông tin nhiệt độ và độ ẩm.

Vi xử lý ATmega324P: Thu nhận tín hiệu số từ DHT22, xử lý tách lấy giá trị nhiệt độ và độ ẩm, chuyển đổi định dạng số theo yêu cầu (ví dụ lấy 1 chữ số thập phân cho nhiệt độ). Đồng thời điều khiển LCD 16x2 hiển thị kết quả.

LCD 16x2: Hiển thị trực quan giá trị nhiệt độ (hàng 1) và độ ẩm (hàng 2) theo định dạng yêu cầu.

3.3. Tính toán và vẽ sơ đồ mạch chi tiết

3.3.1. Tính toán các thành phần chính

a. Nguồn cấp: Nguồn 5V DC ổn định (có thể dùng nguồn USB hoặc bộ nguồn DC 5V).

Nguồn phải đảm bảo đủ dòng để cấp cho vi điều khiển, LCD và cảm biến DHT22.

b. Cảm biến DHT22

Nguồn cấp: 3.3V đến 6V → dùng 5V ổn định

Điện trở kéo lên (Pull-up resistor): chọn giá trị điện trở là 10kΩ nối giữa chân Data của DHT22 và nguồn 5V.

Mục đích: giữ tín hiệu Data ở mức cao khi không có tín hiệu.

c. Vi điều khiển ATmega324P

Tần số hoạt động: 8 MHz

Chân kết nối: PD0 đọc tín hiệu Data từ DHT22.

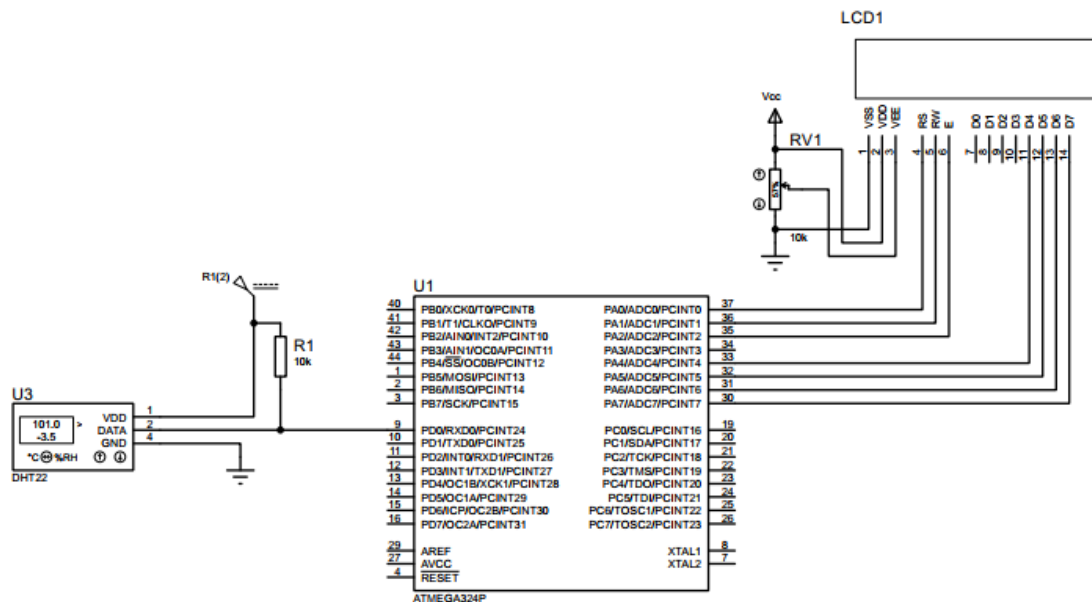
Các chân LCD: RS, RW, EN, D4-D7 nối với PORTA

d. LCD 16x2 (giao tiếp 4-bit)

Điện áp cấp: 5V.

Điện trở thay đổi độ sáng (Contrast): Dùng biến trở 10kΩ nối giữa chân Vo của LCD và GND/Vcc để điều chỉnh độ tương phản.

3.3.2. Sơ đồ mạch chi tiết



Hình 0.2. Sơ đồ mạch chi tiết

4. THIẾT KẾ VÀ THỰC HIỆN PHẦN MỀM

4.1. Yêu cầu đặt ra cho phần mềm

Khởi tạo và giao tiếp được với cảm biến DHT22.

Đọc dữ liệu từ cảm biến mỗi 2 giây.

Tách nhiệt độ và độ ẩm từ dữ liệu nhận được.

- Làm tròn 1 chữ số thập phân cho nhiệt độ
- Làm tròn đến phần nguyên cho độ ẩm

Hiển thị:

Hàng 1 LCD: T: xx.xC

Hàng 2 LCD: RH: xx%

4.2. Yêu cầu kỹ thuật

Độ trễ giữa 2 lần đo: khoảng 2 giây ($\geq 2s$ theo datasheet DHT22).

Thời gian xử lý dữ liệu < 50 ms.

Dữ liệu đọc từ DHT22: 40 bit (16 bit độ ẩm, 16 bit nhiệt độ, 8 bit checksum).

Hiển thị LCD 16x2 chế độ 4-bit.

Giao tiếp cảm biến qua chân PD0: dùng delay để xác định bit 0 bit 1

4.3. Phân tích

DHT22 giao tiếp bằng giao thức 1-wire riêng, tự tay viết.

Quá trình đọc dữ liệu:

- MCU kéo chân DATA xuống 20ms \rightarrow thả ra. (theo datasheet tối thiểu 18ms)
- DHT22 trả lời và gửi 40 bit.

Tách dữ liệu thành 3 byte (24 bit) và dấu của giá trị nhiệt độ lần lượt là

- Cờ T lưu dấu của nhiệt độ
- 8 bit là phần nguyên nhiệt độ
- 8 bit là phần thập phân nhiệt độ
- 8 bit phần nguyên độ ẩm được làm tròn

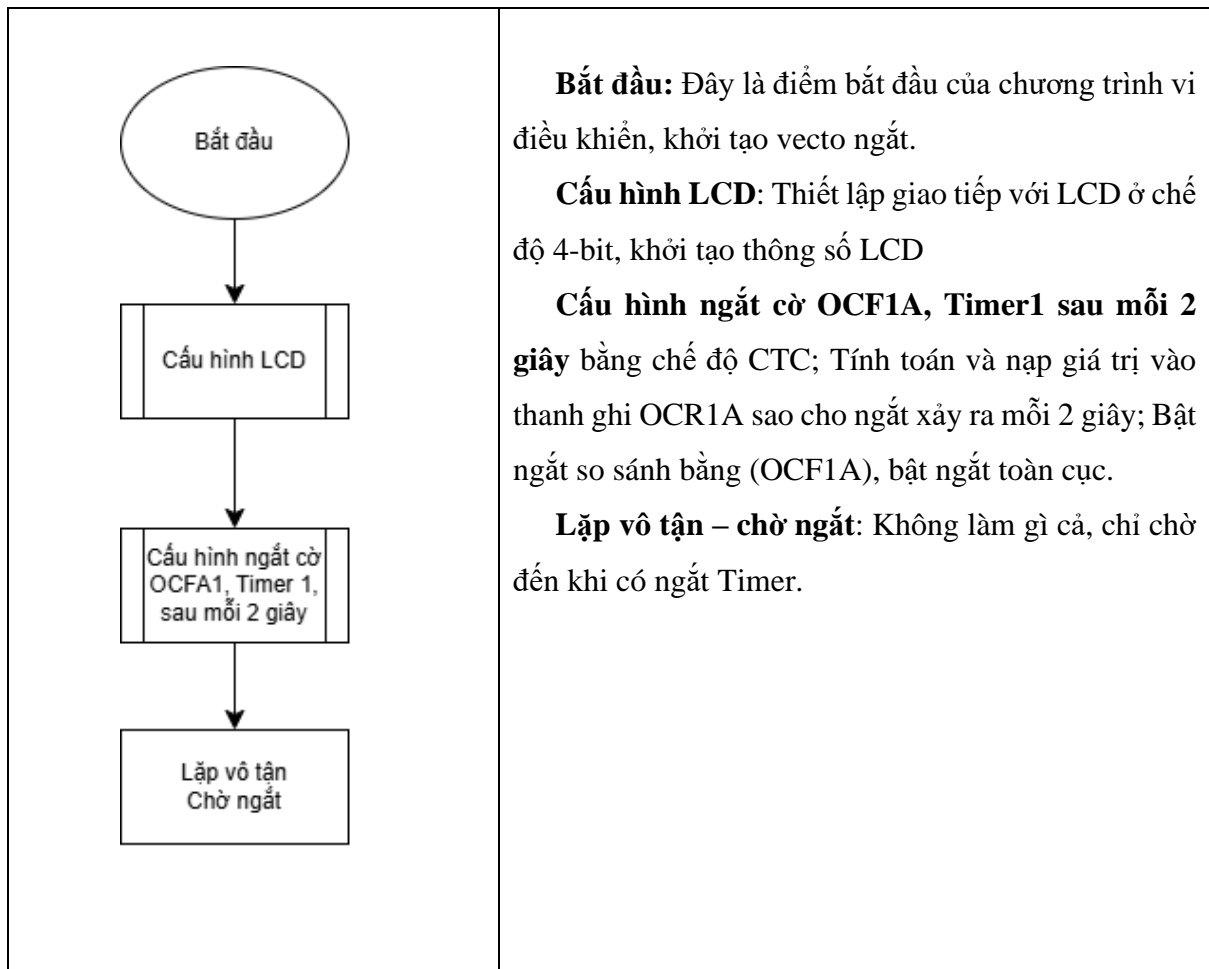
LCD hoạt động ở chế độ 4-bit

4.4. Lưu đồ giải thuật tổng quát

Cứ mỗi 2 giây, chương trình sẽ:

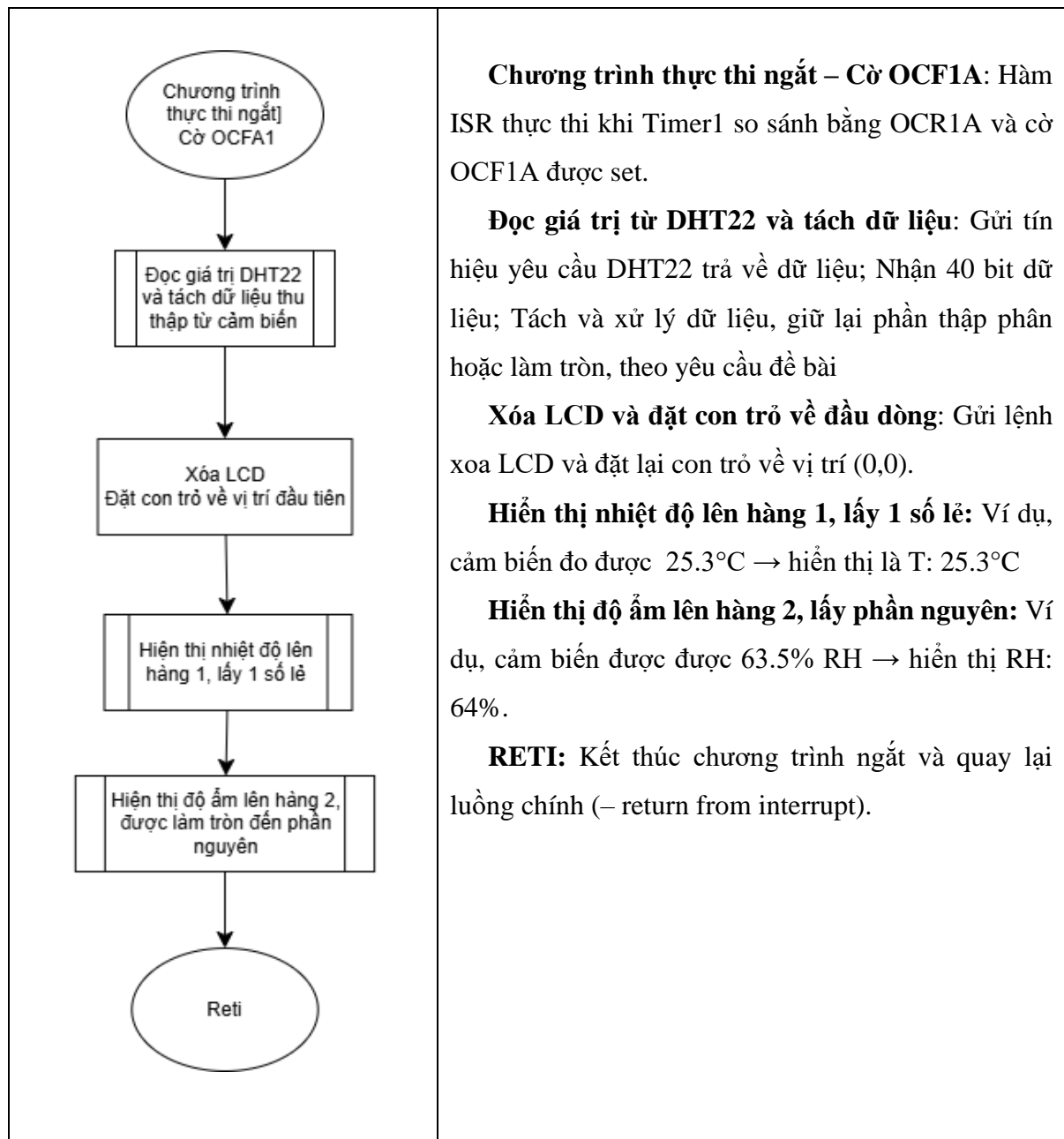
- Đọc nhiệt độ và độ ẩm từ DHT22.
- Hiển thị nhiệt độ (có 1 số lẻ) ở hàng 1.
- Hiển thị độ ẩm (chỉ lấy phần nguyên) ở hàng 2

Luồng chính



Bảng 0.1. Sơ đồ thuật toán luồng chính

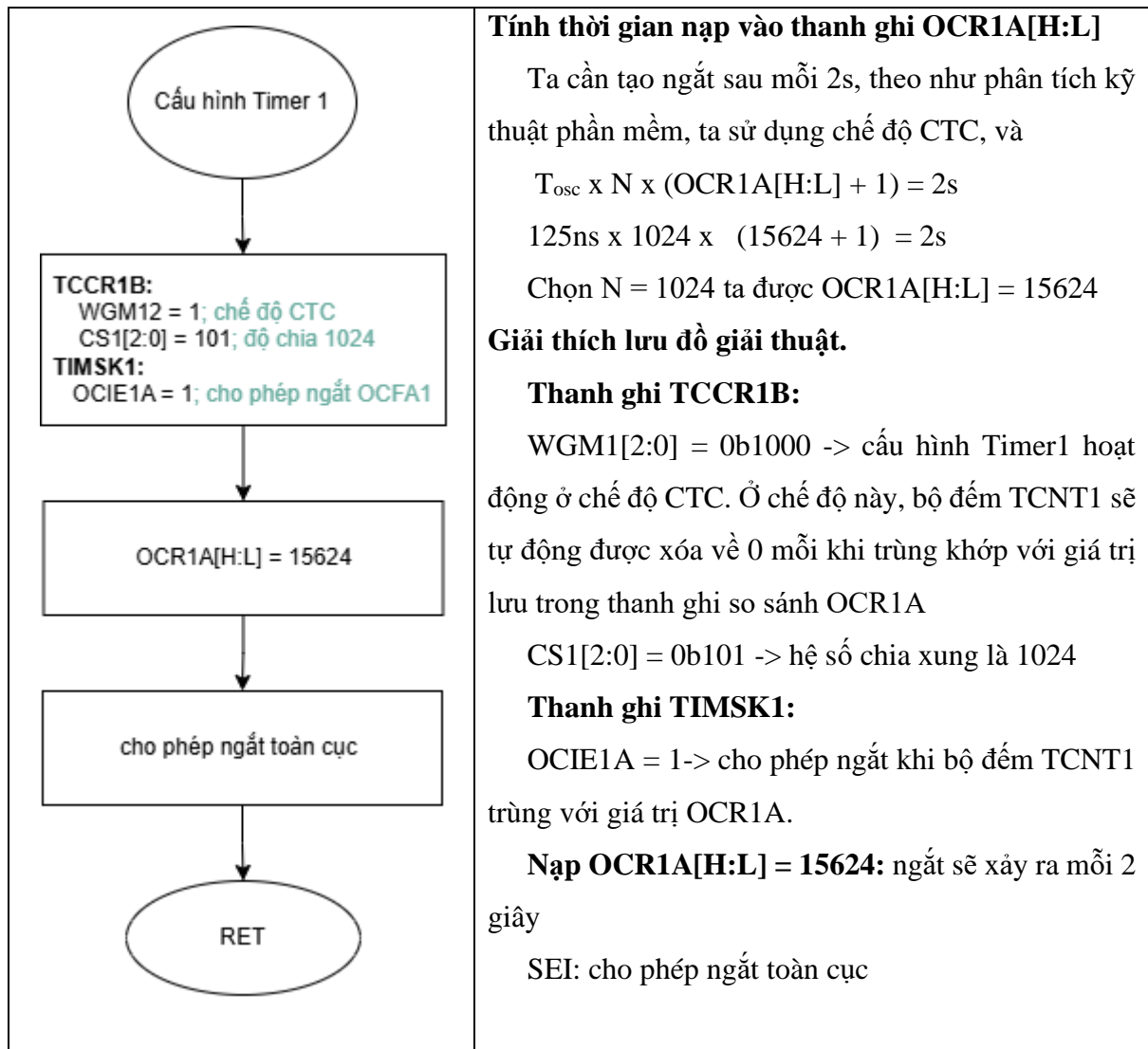
Chương trình con xử lý ngắt



Bảng 0.2. Sơ đồ thuật toán chương trình con xử lý ngắt

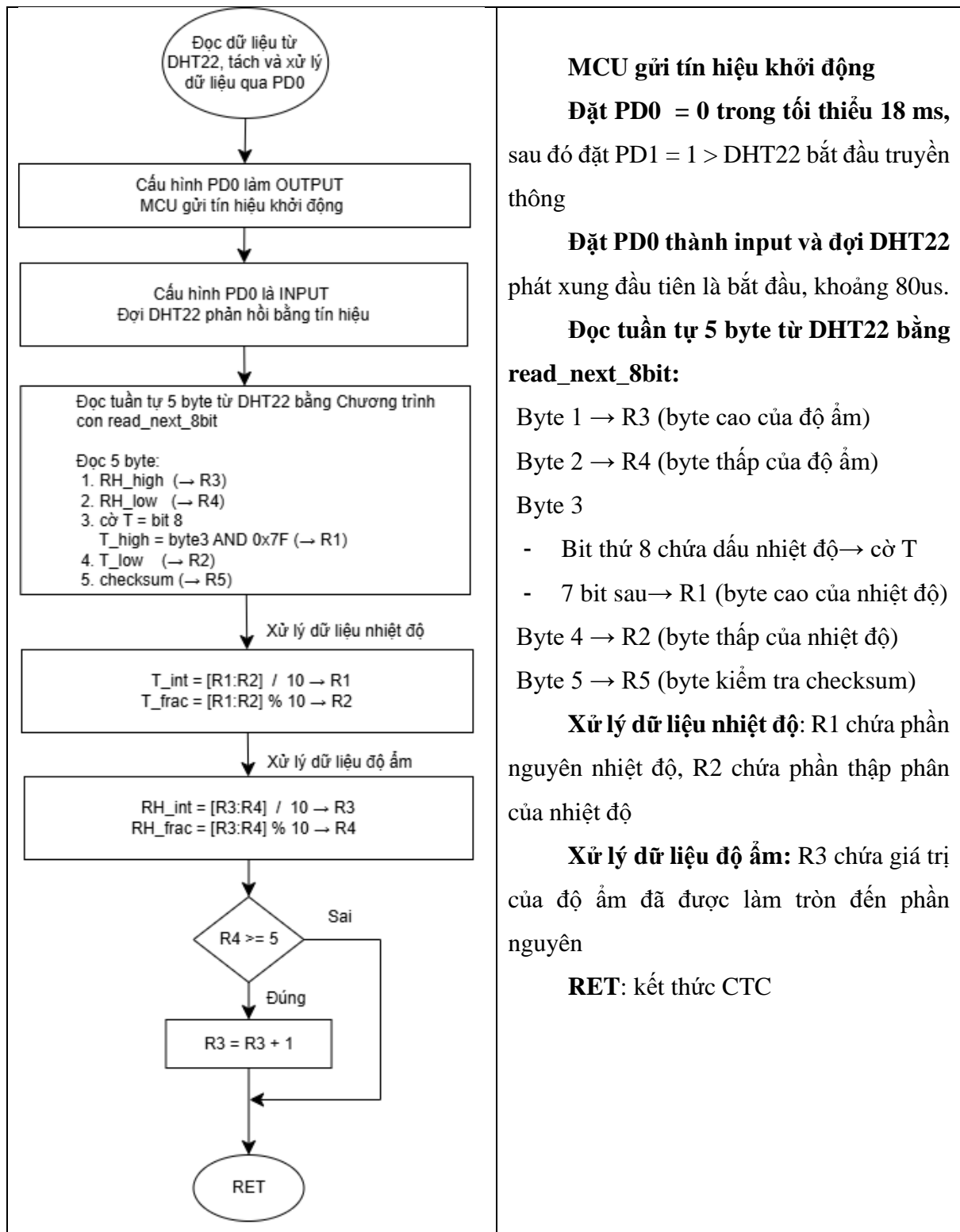
4.5. Lưu đồ giải thuật chi tiết

Cấu hình ngắt OCF1A, timer 1



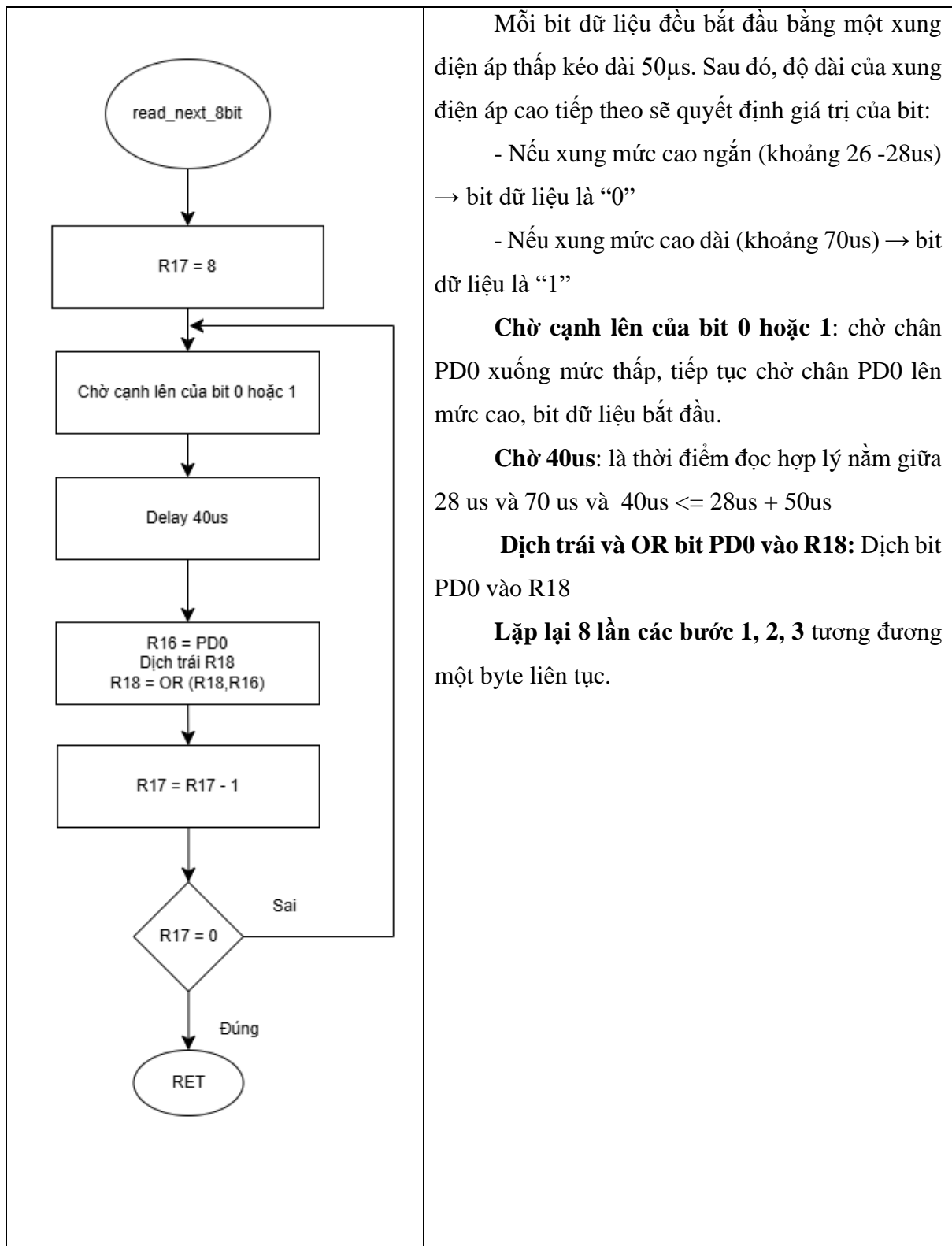
Bảng 0.3. Sơ đồ thuật toán cấu hình ngắt OCF1A, TIMER1

Chương trình đọc dữ liệu từ DHT22 qua chân PD0, tách và xử lý dữ liệu



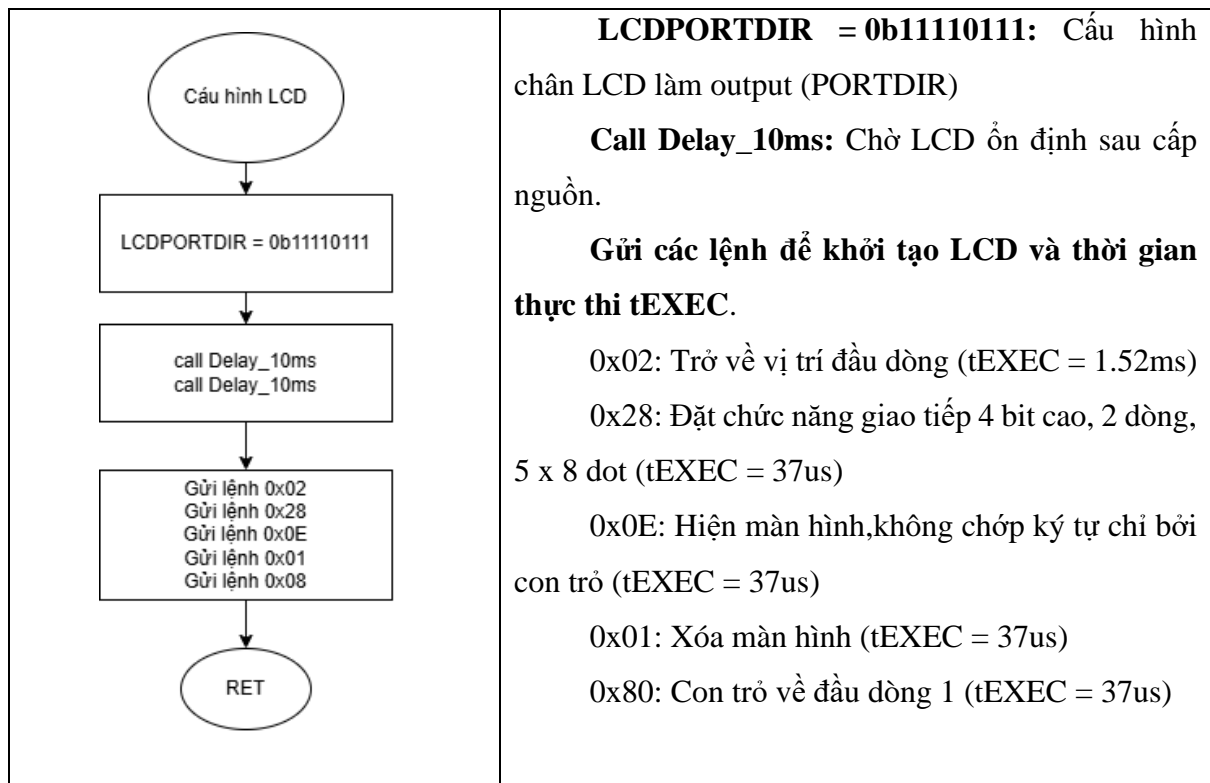
Bảng 0.4. Sơ đồ thuật toán xử lý dữ liệu của DHT22 gửi qua chân PD0

Chương trình con read_next_8bit



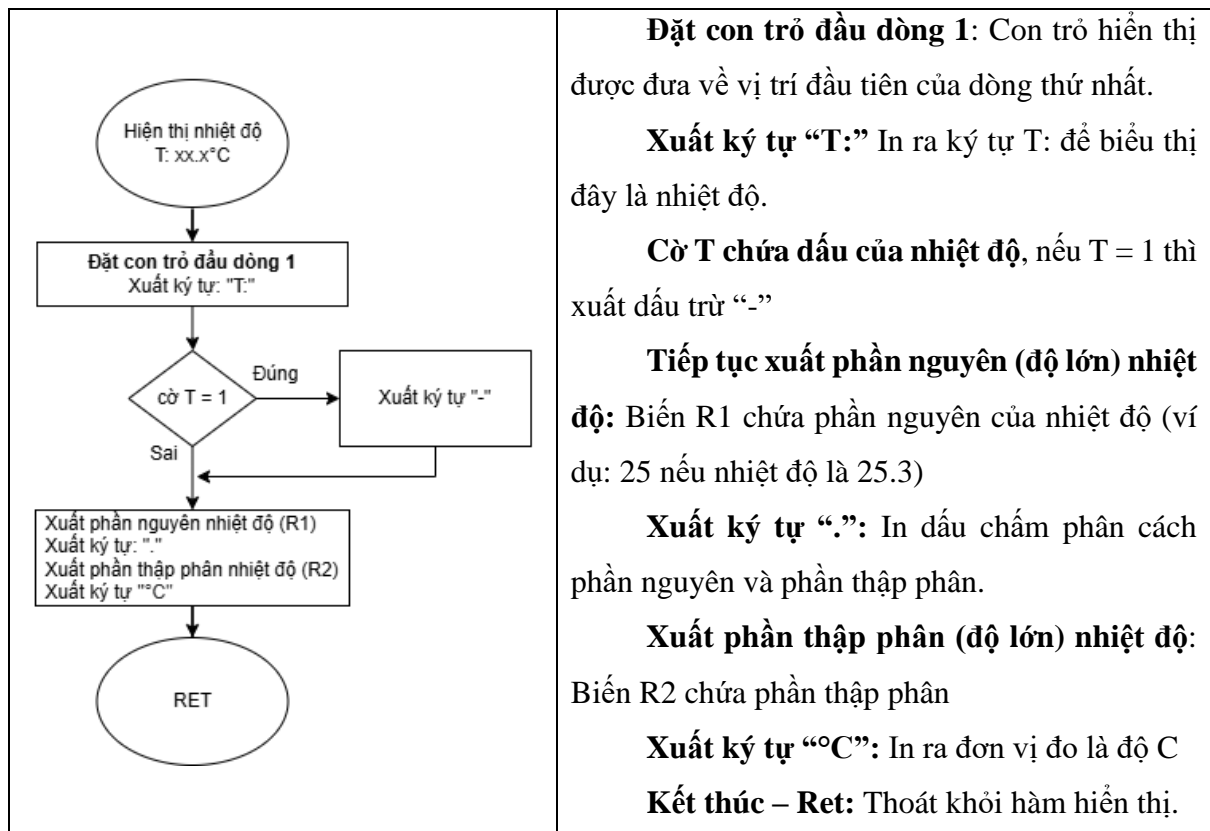
Bảng 0.5. Sơ đồ thuật toán tạo chương trình con read_next_8bit

Cấu hình LCD



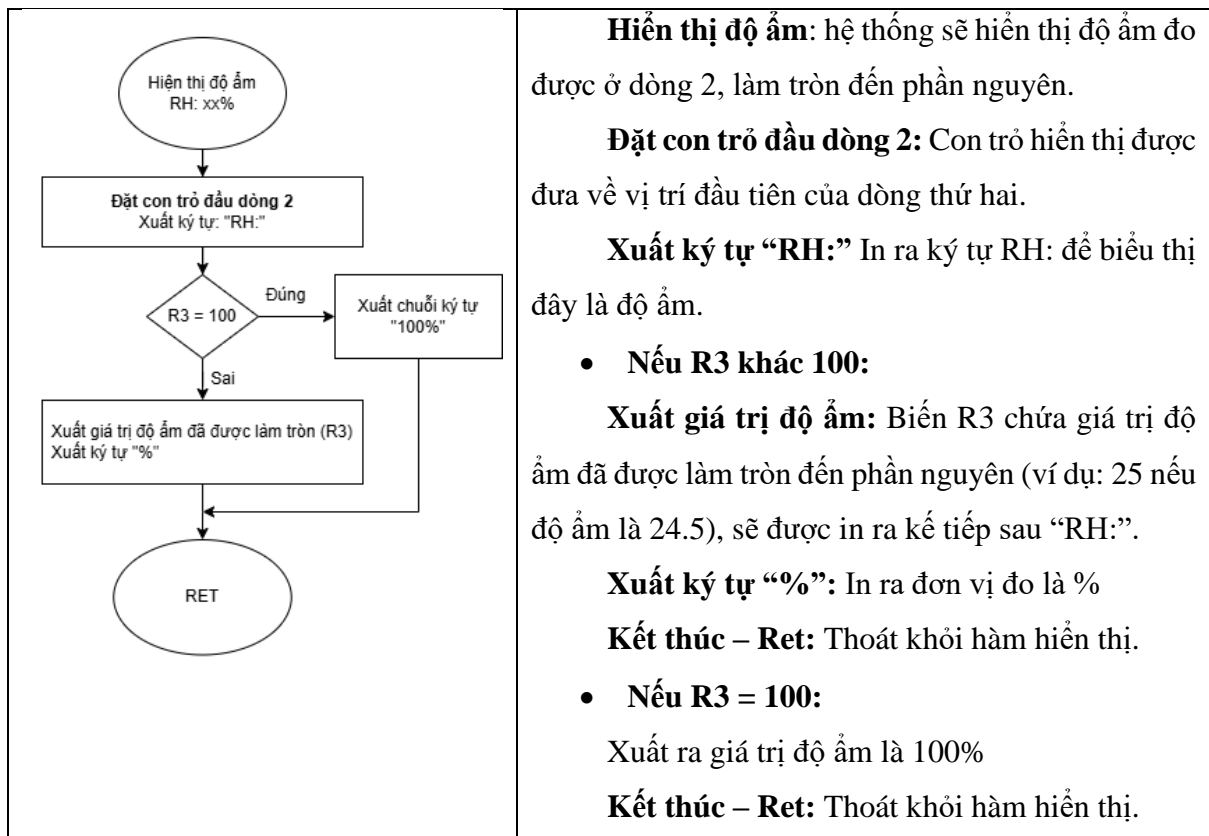
Bảng 0.6. Sơ đồ thuật toán cấu hình LCD

Hiển thị nhiệt độ lên hàng 1



Bảng 0.7. Sơ đồ thuật toán chương trình hiển thị nhiệt độ lên hàng 1

Hiện thị độ ẩm lên hàng 2



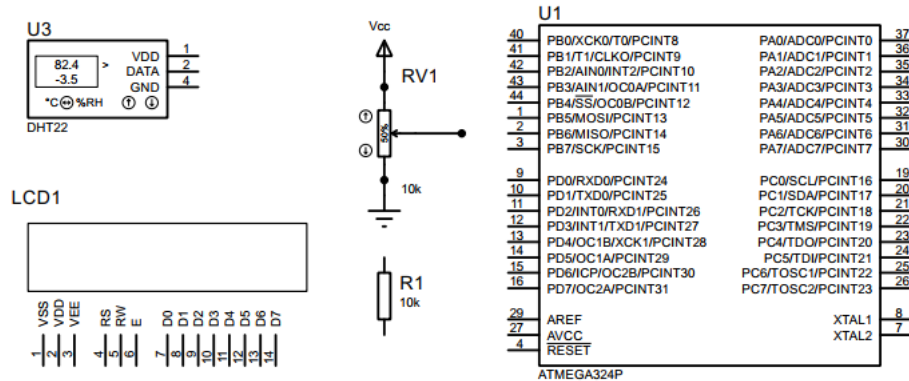
Bảng 0.8. Sơ đồ thuật toán hiện thị độ ẩm lên hàng

5. KẾT QUẢ THỰC HIỆN:

5.1. Cách thức đo đạc và thực nghiệm:

5.1.1. Chuẩn bị linh kiện trong phần mềm mô phỏng

Để thực hiện yêu cầu của đề tài, ta mô phỏng bằng phần mềm Proteus 8.15. Đầu tiên, ta chuẩn bị các linh kiện có sẵn trong thư viện Proteus, bao gồm: ATMEGA324P, cảm biến nhiệt độ DHT22, điện trở 10k Ohm, nguồn điện áp DC, màn hình LCD, nguồn VCC, biến trở, GND.

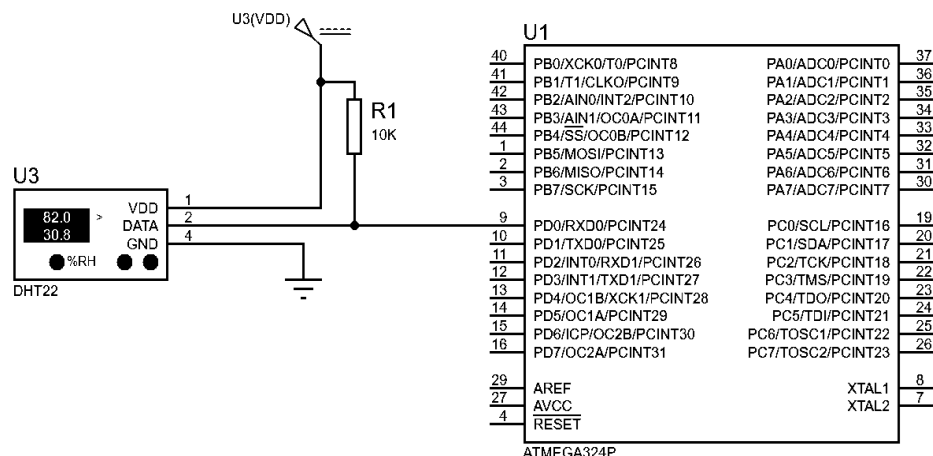


Hình 0.1. Chuẩn bị linh kiện trong Proteus

5.1.2. Kết nối các linh kiện

Kết nối giữa ATMEGA324P và DHT22:

Từ các phân tích nội dung phần lý thuyết, ta có kết nối của ATMEGA324P và DHT22 như hình sau :

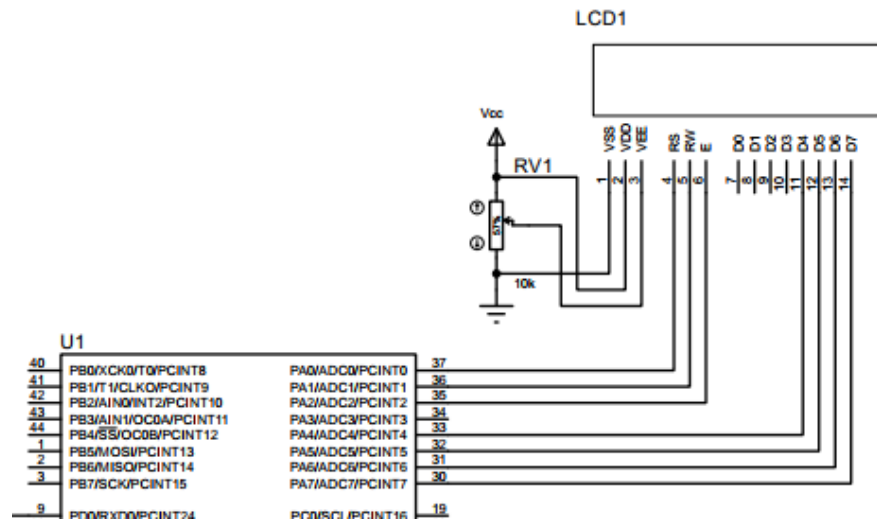


Hình 0.2. Kết nối phần cứng giữa Atmega324P và DHT22 trên phần mềm Proteus

Mô tả kết nối : Chân VDD nối lên nguồn nhằm cấp điện áp cho DHT22 hoạt động. Setup điện trở kéo lên 10kOhm bên ngoài ở chân DATA nhằm xác định mức điện áp nhận vào chân PD0, GND nối đất.

Kết nối giữa ATMEGA324P và LCD 16x2:

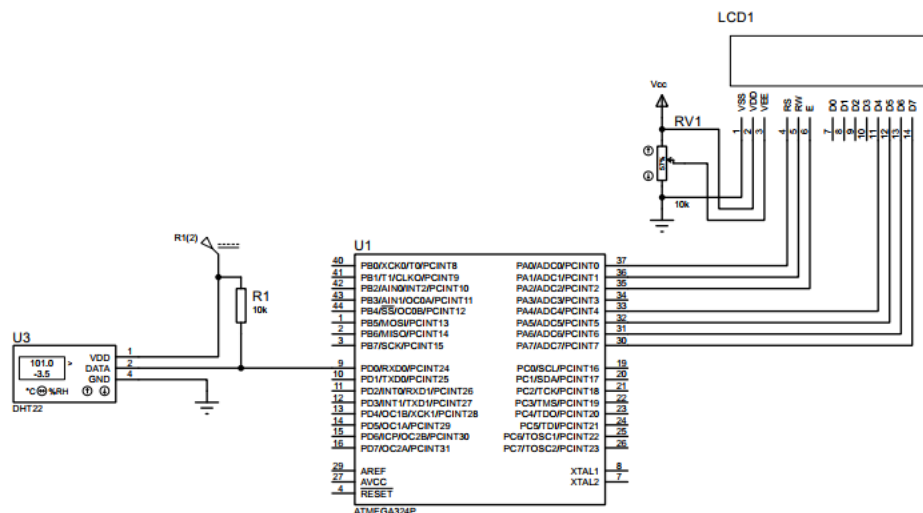
Tương tự, ta có kết nối như hình sau:



Hình 0.3. Kết nối phần cứng giữa Atmega324P và LCD 16x2

Mô tả kết nối: Nối trực tiếp nguồn Vcc với VDD nhằm cấp điện áp cho LCD hoạt động. VSS nối với GND, VEE nối với mức điện áp trên biến trở nhằm điều chỉnh được mức điện áp, từ đó điều khiển được độ sáng tương phản của LCD. Kết nối PA0 - PA2 với lần lượt các chân: RS, RW, E nhằm điều khiển các chân này. Kết nối PA4 – PA7 với D4 – D7 của LCD (giao tiếp 4 bit).

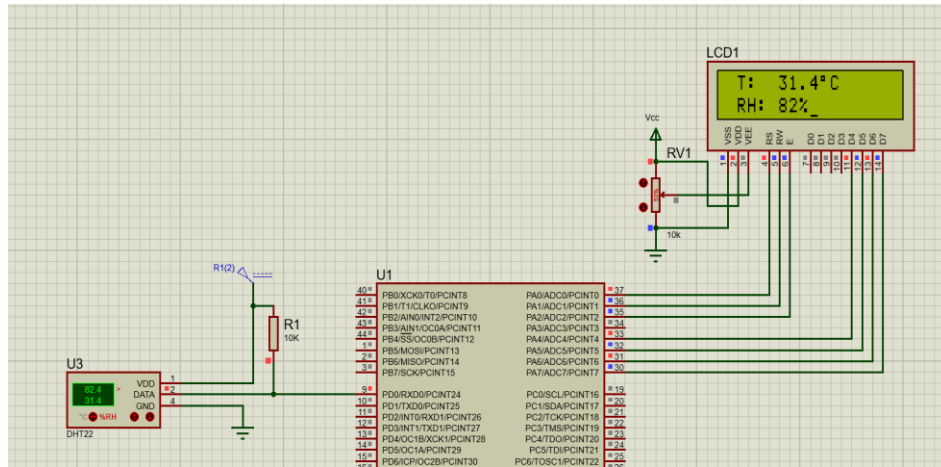
Từ đó, kết nối trên phần mềm hoàn chỉnh để mô phỏng như hình:



Hình 0.4. Kết nối phần cứng hoàn chỉnh trên phần mềm Proteus

5.2. Phân tích kết quả thu được:

Thực hiện mô phỏng, ta thu được kết quả như hình sau:



Hình 0.5. Kết quả mô phỏng lần 1
(Các hình ảnh mô phỏng các lần còn lại ở phần phụ lục)

Giải thích: Khi cảm biến nhiệt độ và độ ẩm DHT22 ghi nhận độ ẩm và nhiệt độ của môi trường thì lần lượt sẽ xuất ra 32 bit data và 8 bit checksum. Cảm biến DHT22 trên nhận được giá trị từ môi trường là 82.4RH– 31.4T, từ đó ta có 32 bit data lần lượt là:

0000 0011 - 0011 1000 – 0000 0001 – 0011 1010

Lần lượt là: 8 bit cao độ ẩm – 8 bit thấp độ ẩm – 8 bit cao nhiệt độ – 8 bit thấp nhiệt độ. Từ 32 bit data trên, ta lưu lần lượt R1 = 1₁₀ (byte cao T), R2 = 58₁₀ (byte thấp T), R3 = 3₁₀ (byte cao RH), R4 = 56₁₀ (byte thấp RH).

Xử lý:

- Phần nhiệt độ:

- +) $[R1:R2]/10 \rightarrow R1 \Leftrightarrow 314/10 \rightarrow R1 \Rightarrow R1 = 31_{10}$ (phần nguyên T)
- +) $[R1:R2]\%10 \rightarrow R2 \Leftrightarrow 314\%10 \rightarrow R2 \Rightarrow R2 = 4_{10}$ (phần thập phân T)
- +) Lần lượt in dấu, R1, “.”, R2 ra LCD

- Phần độ ẩm

- +) $[R3:R4]/10 \rightarrow R3 \Leftrightarrow 824/10 \rightarrow R3 \Rightarrow R3 = 82_{10}$ (phần nguyên RH)
- +) $[R3:R4]\%10 \rightarrow R4 \Leftrightarrow 824\%10 \rightarrow R4 \Rightarrow R4 = 4_{10}$ (Phần thập phân RH)
- +) Nếu $R4 \geq 5 \Rightarrow R3 \leftarrow R3 + 1$ không thì bỏ qua.
- +) Lần lượt in R3, “%” ra LCD

Bảng số liệu:

Lần đo	1	2	3
Lý thuyết	82.4%RH-31.4°C	81.5%RH-29.5°C	82.0%RH-30.0°C
Đo đạc	82%RH – 31.4°C	82%RH – 29.5°C	82%RH – 30.0°C

Bảng 0.1. Kết quả đo đạc và kiểm chứng lý thuyết qua 3 lần đo

6. KẾT LUẬN

Qua quá trình thực hiện đồ án, nhóm đã tích lũy được nhiều kiến thức và kỹ năng quý báu. Cụ thể, nhóm đã nắm vững hơn về nguyên lý hoạt động và quy trình giao tiếp của cảm biến DHT22, đặc biệt là kỹ thuật xử lý tín hiệu theo độ rộng xung trong giao tiếp 1-wire. Việc triển khai giao tiếp giữa ATmega324P và cảm biến DHT22 giúp nhóm nâng cao khả năng lập trình hợp ngữ cho vi điều khiển, cũng như củng cố kiến thức về sử dụng ngắt Timer1 ở chế độ CTC và giao tiếp LCD 16x2 ở chế độ 4-bit.

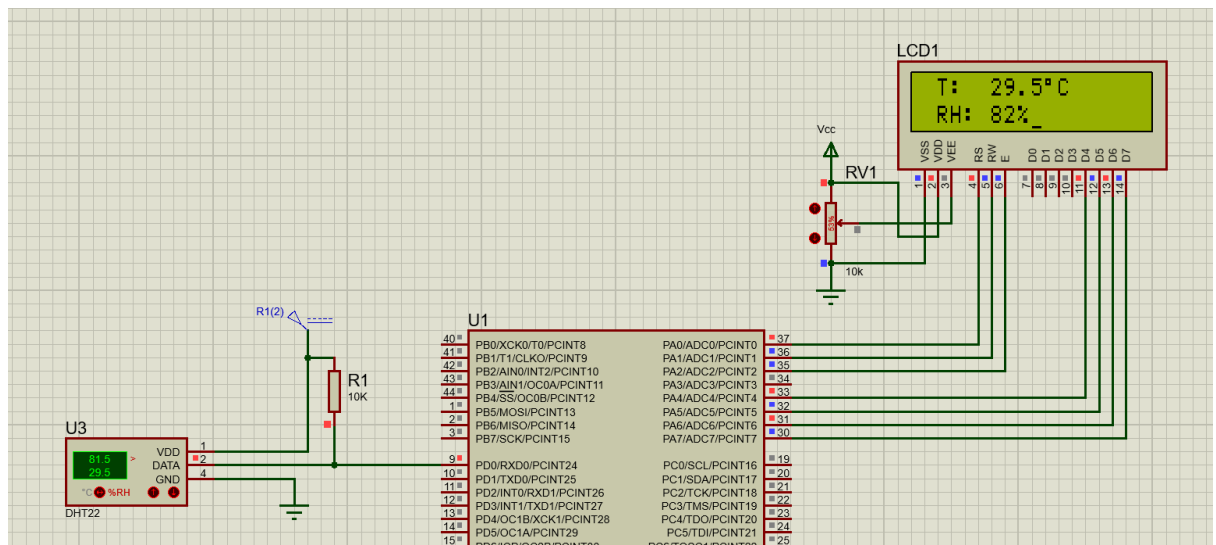
Bên cạnh các kiến thức chuyên môn, đồ án còn giúp nhóm cải thiện kỹ năng làm việc nhóm, quản lý tiến độ, trình bày báo cáo và mô phỏng phần cứng bằng phần mềm Proteus. Đây là những hành trang thiết thực để chuẩn bị cho các đồ án và dự án kỹ thuật sau này.

Tuy nhiên, trong quá trình thực hiện, nhóm cũng gặp phải một số khó khăn và hạn chế. Thứ nhất, do đặc điểm truyền dữ liệu dựa trên độ rộng xung của DHT22, tốc độ lấy mẫu còn thấp, gây độ trễ nhất định trong quá trình hiển thị. Thứ hai, ban đầu nhóm có sự hiểu nhầm về định dạng dữ liệu nhiệt độ, dẫn đến xử lý sai phần giá trị thập phân, ảnh hưởng đến độ chính xác của kết quả. Những hạn chế này đã được rút kinh nghiệm và là cơ sở để nhóm tiếp tục cải tiến và mở rộng hệ thống trong các giai đoạn tiếp theo.

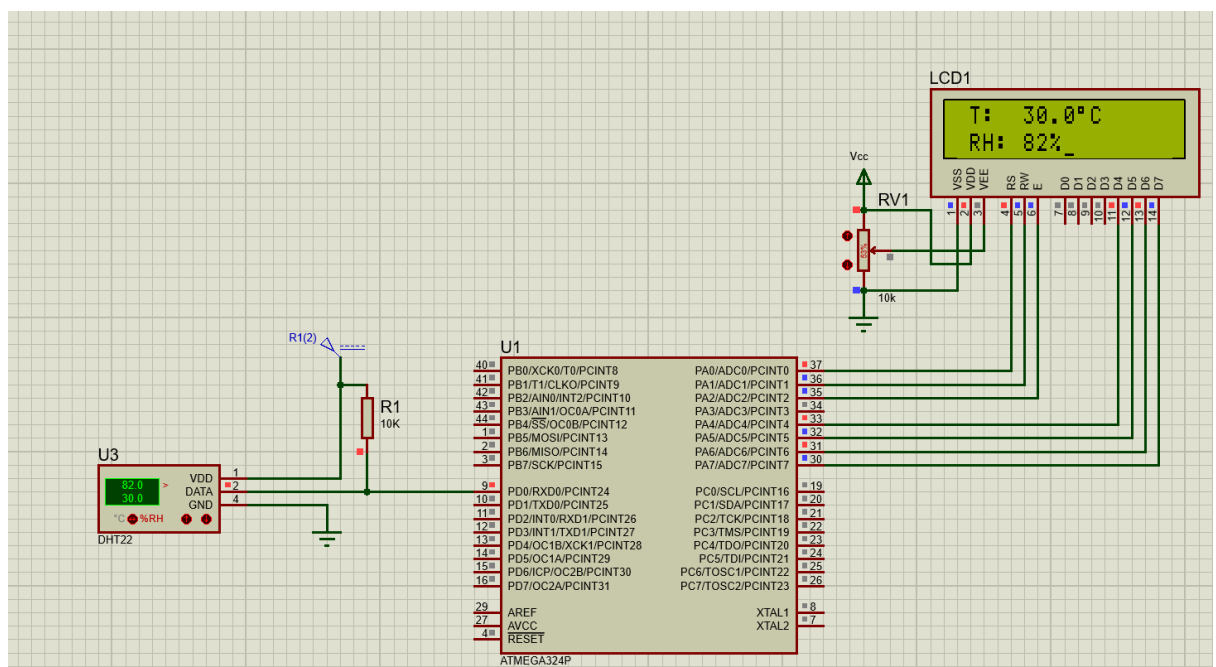
TÀI LIỆU THAM KHẢO

1. Hoàng Trang (Chủ biên), Lưu Phú (Đồng chủ biên), Nguyễn Lý Thiên Trường, Lê Thị Kim Anh, Nguyễn Trọng Luật, Bùi Quốc Bảo, Trần Hoàng Linh. *Giáo trình Vi xử lý*. Đại học Quốc gia TP. Hồ Chí Minh, Trường Đại học Bách Khoa.
2. *Tài liệu thí nghiệm vi xử lý* – Khoa Điện Điện tử.
3. Datasheet: *DHT22 Digital-output humidity & temperature sensor*. Truy cập từ: <https://www.alldatasheet.com/datasheet-pdf/pdf/1132459/ETC2/DHT22.html>.
4. Datasheet: *Digital Humidity and Temperature Sensor AM2302*. Truy cập từ: <https://cdn-shop.adafruit.com/datasheets/Digital+humidity+and+temperature+sensor+AM2302.pdf>.
5. Nguyễn Lý Thiên Trường. Slide Chương 6: Giao tiếp ngoại vi.
6. Datasheet: *DHT22 Temperature-Humidity Sensor DHT22*. Truy cập từ: https://xonstorage.blob.core.windows.net/pdf/waveshare_dht22temperaturehumiditysensor_Lcs01_link.pdf.

PHỤ LỤC



Hình 0.1. Kết quả mô phỏng lần 2



Hình 0.2. Kết quả mô phỏng lần 3