

**BỘ GIÁO DỤC VÀ ĐÀO TẠO  
TRƯỜNG ĐẠI HỌC QUẢN LÝ VÀ CÔNG NGHỆ HẢI PHÒNG**



**ĐỒ ÁN TỐT NGHIỆP**  
**NGÀNH ĐIỆN TỰ ĐỘNG CÔNG NGHIỆP**

**Sinh viên: Đỗ Đại Dương**

**Giảng viên hướng dẫn: ThS. Nguyễn Đoàn Phong**

**HẢI PHÒNG – 2023**

**BỘ GIÁO DỤC VÀ ĐÀO TẠO**  
**TRƯỜNG ĐẠI HỌC QUẢN LÝ VÀ CÔNG NGHỆ HẢI PHÒNG**

---

**ĐỀ TÀI:**  
**THIẾT KẾ GIAO DIỆN ĐIỀU KHIỂN QUA TRÌNH**  
**CHỤP ẢNH TỰ ĐỘNG CỦA MÁY ĐO THÂN NHIỆT**  
**KHÔNG TIẾP XÚC**

**ĐỒ ÁN TỐT NGHIỆP ĐẠI HỌC HỆ LIÊN THÔNG CHÍNH QUY**  
**NGÀNH ĐIỆN TỰ ĐỘNG CÔNG NGHIỆP**

**Sinh viên : Đỗ Đại Dương**  
**Giảng viên hướng dẫn: Th. Nguyễn Đoàn Phong**

**HẢI PHÒNG – 2023**

**BỘ GIÁO DỤC VÀ ĐÀO TẠO**  
**TRƯỜNG ĐẠI HỌC QUẢN LÝ VÀ CÔNG NGHỆ HẢI PHÒNG**

-----

**NHIỆM VỤ ĐỀ TÀI TỐT NGHIỆP**

**Sinh viên :** Đỗ Đại Dương

**MSSV:** 2113102004

**Lớp :** DCL2501

**Nghành :** Điện Tự Động Công Nghiệp

**Tên đề tài:** Thiết kế giao diện điều khiển quá trình chụp ảnh tự động của máy đo thân nhiệt không tiếp xúc.

# NHIỆM VỤ ĐỀ TÀI

**1. Nội dung và các yêu cầu cần giải quyết trong nhiệm vụ đề tài tốt nghiệp (về lý luận, thực tiễn, các số liệu cần tính toán và các bản vẽ).**

.....  
.....  
.....  
.....  
.....  
.....  
.....  
.....  
.....  
.....  
.....  
.....

**2. Các số liệu cần thiết để thiết kế, tính toán.**

.....  
.....  
.....  
.....  
.....  
.....  
.....  
.....  
.....  
.....  
.....  
.....

**3. Địa điểm thực tập tốt nghiệp.....**

.....  
.....  
.....  
.....  
.....  
.....  
.....  
.....  
.....  
.....

# **CÁC CÁN BỘ HƯỚNG DẪN ĐỀ TÀI TỐT NGHIỆP**

**Họ và tên :** Nguyễn Đoàn Phong

**Học hàm, học vị :** Thạc Sĩ

**Cơ quan công tác :** Trường Đại học Quản lý và Công nghệ Hải Phòng

**Nội dung hướng dẫn:** Thiết kế giao diện điều khiển quá trình chụp ảnh tự động của máy đo thân nhiệt không tiếp xúc

Đề tài tốt nghiệp được giao ngày 09 tháng 10 năm 2023

Yêu cầu phải hoàn thành xong trước ngày 20 tháng 1 năm 2024

Đã nhận nhiệm vụ Đ.T.T.N

**Sinh viên**

Đã giao nhiệm vụ Đ.T.T.N

**Cán bộ hướng dẫn Đ.T.T.N**

**Đỗ Đại Dương**

**Th. Nguyễn Đoàn Phong**

Hải Phòng, ngày.....tháng ..... năm 2023.

**TRƯỞNG KHOA**

**Cộng Hòa Xã Hội Chủ Nghĩa Việt Nam**

**Độc lập - Tự do - Hạnh phúc**

**PHIẾU NHẬN XÉT CỦA GIẢNG VIÊN HƯỚNG DẪN TỐT NGHIỆP**

**Họ và tên giảng viên:** Nguyễn Đoàn Phong

**Đơn vị công tác:** Trường Đại học Quản lý và Công nghệ Hải Phòng

**Họ và tên sinh viên:** Đỗ Đại Dương

**Chuyên ngành:** Điện tự động công nghiệp

**Nội dung hướng dẫn :** Toàn bộ đề tài

**1. Tinh thần thái độ của sinh viên trong quá trình làm đề tài tốt nghiệp**

.....  
.....  
.....  
.....

**2. Đánh giá chất lượng của đề án/khóa luận( so với nội dung yêu cầu đã đề ra trong nhiệm vụ Đ.T.T.N, trên các mặt lý luận, thực tiễn, tính toán số liệu... )**

.....  
.....  
.....

**3. Ý kiến của giảng viên hướng dẫn tốt nghiệp**

Được bảo vệ  Không được bảo vệ  Điểm hướng dẫn

Hải Phòng, ngày.....tháng.....năm 2023  
**Giảng viên hướng dẫn**

**Th. Nguyễn Đoàn Phong**

**Cộng hòa xã hội chủ nghĩa Việt Nam**

**Độc lập - Tự do - Hạnh phúc**

-----  
**PHIẾU NHẬN XÉT CỦA GIẢNG VIÊN CHẤM PHẢN BIỆN**

**Họ và tên giảng viên:**.....

**Đơn vị công tác:**.....

**Họ và tên sinh viên:** ..... **Chuyên ngành:**.....

**Đề tài tốt nghiệp:** .....

**1. Phần nhận xét của giảng viên chấm phản biện**

.....  
.....  
.....  
.....

**2. Những mặt còn hạn chế**

.....  
.....  
.....  
.....

**3. Ý kiến của giảng viên chấm phản biện**

Được bảo vệ  Không được bảo vệ  Điểm hướng dẫn

Hải Phòng, ngày.....tháng.....năm 2023  
**Giảng viên chấm phản biện**

## LỜI NÓI ĐẦU

Ngày nay khoa học công nghệ ngày càng phát triển, vi điều khiển AVR , PIC, ARM ngày càng trở lên phổ biến, nhưng có thể nói sự xuất hiện của Arduino đã mở ra một hướng đi mới cho vi điều khiển. Sự xuất hiện của Arduino đã hỗ trợ cho con người rất nhiều trong lập trình và thiết kế, nhất là đối với những người bắt đầu tìm tòi về vi điều khiển mà không có quá nhiều kiến thức, hiểu biết sâu sắc về vật lý và điện tử. Phần cứng của thiết bị đã được tích hợp nhiều chức năng cơ bản và là mã nguồn mở. Ngôn ngữ lập trình trên nền tảng Java lại vô cùng dễ sử dụng tương thích với ngôn ngữ C và hệ thư viện rất phong phú và được chia sẻ miễn phí. Chính vì những lý do như vậy nên Arduino hiện đang dần phổ biến và được phát triển ngày càng mạnh mẽ trên toàn thế giới.

Trong đợt dịch COVID-19 vừa qua, những vấn đề liên quan đến phòng ngừa dịch bệnh trở lên cấp bách, nhất là vấn đề lưu vết những người có thân nhiệt cao. Trên cơ sở kiến thức đã học cùng với sự giúp đỡ của thầy Nguyễn Đoàn Phong em đã quyết định thực hiện đề tài : **Thiết kế giao diện quá trình điều khiển của máy đo thân nhiệt không tiếp xúc** với mục đích tìm hiểu thêm về Arduino, cũng như đóng góp vào quá trình xử lý thông tin, cũng như lưu vết những người có khả năng nhiễm bệnh.

Do kiến thức còn hạn hẹp, thêm vào đó đây là lần đầu em thực hiện đồ án nên chắc chắn không tránh khỏi những thiếu sót , hạn chế vì thế em rất mong có được sự góp ý và nhắc nhở từ thầy giáo để em có thể hoàn thiện đề tài của mình.

Em xin chân thành cảm ơn thầy giáo hướng dẫn Th. Nguyễn Đoàn Phong đã giúp đỡ em rất nhiều trong quá trình tìm hiểu ,thiết kế và hoàn thành đồ án này.

Em xin được trân trọng cảm ơn!

*Hải Phòng, ngày...tháng...năm 20...*  
**Sinh viên**

**Đỗ Đại Dương**



## LỜI CẢM ƠN

Đây là kết quả của quá trình những năm tháng học tập của em nhưng do kinh nghiệm thực tế của bản thân còn chưa nhiều nên khó tránh khỏi nhiều thiếu sót, do đó cần phải có sự hướng dẫn, giúp đỡ của các thầy cô giáo. Qua đây em cũng xin gửi lời cảm ơn chân thành đến quý thầy cô Trường Đại học Quản Lý và Công Nghệ Hải Phòng, khoa Điện – Điện tử, các thầy cô bộ môn lời cảm ơn chân thành nhất, các thầy cô đã tận tình giảng dạy cho em thời gian vừa qua, các thầy cô đã trang bị cho em nhiều kiến thức cơ bản về lĩnh vực điện tự động công nghiệp. Và cuối cùng em xin cảm ơn thầy giáo Th.Nguyễn Đoàn Phong đã giúp đỡ và hướng dẫn em trong suốt quá trình làm đề án tốt nghiệp. Thầy đã tận tình giúp đỡ, định hướng, góp ý và cung cấp những ý tưởng quý báu trong suốt thời gian làm đề án.

Em xin chân thành cảm ơn !

# MỤC LỤC

LỜI CẢM ƠN.....	9
CHƯƠNG 1 TỔNG QUAN VỀ VI XỬ LÝ ARDUINO .....	12
1.1. GIỚI THIỆU VỀ ARDUINO NANO .....	12
1.1.1. Thông số kỹ thuật Arduino Nano .....	12
1.1.2. Arduino Nano Schematic.....	17
1.2. NGUỒN NUÔI CHO ARDUINO NANO VÀ DRIVE A4988.....	18
1.2.1. Chuyển nguồn AC~220V sang DC-12V .....	18
1.2.2. Chuyển nguồn từ DC-12V sang DC-5V .....	18
1.3. MẠCH DAO ĐỘNG CHO ARDUINO NANO .....	20
1.4. RESET .....	20
1.5. GIAO TIẾP MÁY TÍNH .....	21
1.6. LẬP TRÌNH CHO ARDUINO NANO.....	22
CHƯƠNG 2 TỔNG QUAN VỀ C# .....	24
2.1. TÌM HIỂU VỀ NGÔN NGỮ C# .....	24
2.1.1. C# là ngôn ngữ đơn giản.....	24
2.1.2. C# là ngôn ngữ hiện đại .....	24
2.1.3. C# là ngôn ngữ hướng đối tượng .....	25
2.1.4. C# là ngôn ngữ mạnh mẽ và cũng mềm dẻo .....	25
2.1.5. C# là ngôn ngữ ít từ khóa .....	25
2.1.6. C# là ngôn ngữ hướng module .....	26
2.1.7. C# sẽ là một ngôn ngữ phổ biến .....	26
2.1.8. Ngôn ngữ C# và những ngôn ngữ khác.....	26
2.2. CÁC BƯỚC CHUẨN BỊ CHO CHƯƠNG TRÌNH.....	28
2.2.1. Chương trình C# đơn giản .....	29
2.2.2. Lớp, đối tượng và kiểu dữ liệu (type).....	29
2.2.3. Phương thức.....	30
2.2.4. Chú thích.....	31
2.2.5. Ứng dụng Console .....	32
2.2.6. Toán tử ‘.’.....	33
2.2.7. Từ khóa using .....	33
2.2.8. Từ khóa static .....	35
CHƯƠNG 3 THIẾT KẾ GIAO DIỆN ĐIỀU KHIỂN QUA TRÌNH CHỤP ẢNH TỰ ĐỘNG CỦA MÁY ĐO THÂN NHIỆT KHÔNG TIẾP XÚC ..	36
3.1. ĐỘNG CƠ BƯỚC .....	36
3.1.1. Cấu tạo và nguyên lý hoạt động của động cơ bước .....	36
3.1.2. Điều khiển động cơ bước .....	36
3.1.3. Chân ra Driver A4988 .....	37
3.2. CÁC CẢM BIẾN SỬ DỤNG TRONG ĐỒ ÁN .....	39
3.2.1. Cảm biến không tiếp xúc Ztemp TN901 .....	39
3.2.2. Cảm biến hồng ngoại .....	41
3.2.3. Cảm biến siêu âm (HC-SRF04).....	41
3.2.4. Công tắc hành trình .....	42

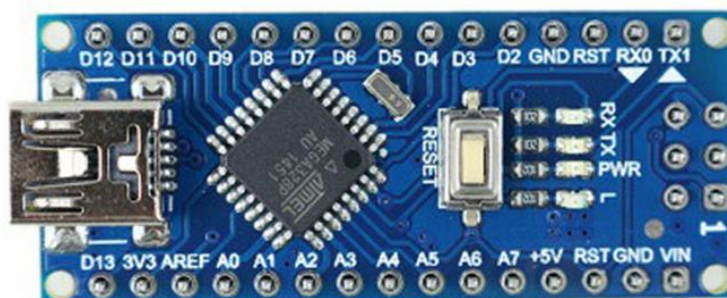
<b>3.3. KẾT NỐI VỚI LCD QUA GIAO THỨC I2C .....</b>	<b>43</b>
<b>3.4. SƠ ĐỒ NGUYÊN LÝ MẠCH VÀ GIAO DIỆN NGƯỜI DÙNG TRÊN C#.....</b>	<b>44</b>
<b>3.5. CODE THAM KHẢO.....</b>	<b>46</b>
<b>KẾT LUẬN.....</b>	<b>55</b>
<b>TÀI LIỆU THAM KHẢO.....</b>	<b>56</b>

# CHƯƠNG 1 TỔNG QUAN VỀ VI XỬ LÝ ARDUINO

## 1.1. GIỚI THIỆU VỀ ARDUINO NANO

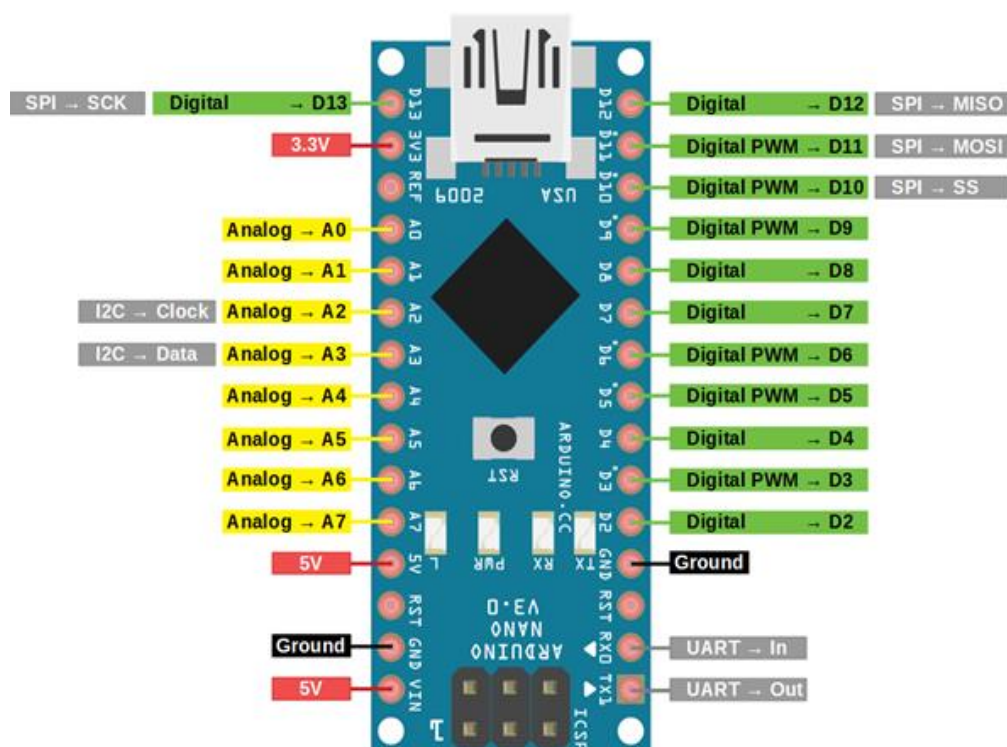
### 1.1.1. Thông số kỹ thuật Arduino Nano

Arduino Nano cảm nhận đó là sự tiện dụng, đơn giản, có thể lập trình trực tiếp bằng máy tính (như Arduino Uno R3) và đặc biệt hơn cả đó là kích thước của nó. Kích thước của **Arduino Nano** cực kì nhỏ chỉ tương đương đồng 2 nghìn gấp lại 2 lần thôi (1.85cm x 4.3cm), rất thích hợp cho các bạn mới bắt đầu học Arduino, vì giá rẻ hơn Arduino Uno R3 nhưng dùng được tất cả các thư viện của mạch này.



Hình 1. 1. Hình ảnh thực tế của Arduino Nano

+ Arduino Nano Pinout



Hình 1. 2. Các chân đầu vào/ra của Arduino nano

*Bảng 1. 1. Chức năng của các chân*

Thứ tự chân	Tên Pin	Kiểu	Chức năng
1	D1 / TX	I / O	Ngõ vào/ra số Chân TX-truyền dữ liệu
2	D0 / RX	I / O	Ngõ vào/ra số Chân Rx-nhận dữ liệu
3	RESET	Đầu vào	Chân reset, hoạt động ở mức thấp
4	GND	Nguồn	Chân nối mass
5	D2	I / O	Ngõ vào/ra digital
6	D3	I / O	Ngõ vào/ra digital
7	D4	I / O	Ngõ vào/ra digital
8	D5	I / O	Ngõ vào/ra digital
9	D6	I / O	Ngõ vào/ra digital
10	D7	I / O	Ngõ vào/ra digital
11	D8	I / O	Ngõ vào/ra digital
12	D9	I / O	Ngõ vào/ra digital
13	D10	I / O	Ngõ vào/ra digital
14	D11	I / O	Ngõ vào/ra digital
15	D12	I / O	Ngõ vào/ra digital
16	D13	I / O	Ngõ vào/ra digital
17	3V3	Đầu ra	Đầu ra 3.3V (từ FTDI)
18	AREF	Đầu vào	Tham chiếu ADC
19	A0	Đầu vào	Kênh đầu vào tương tự kênh 0
20	A1	Đầu vào	Kênh đầu vào tương tự kênh 1
21	A2	Đầu vào	Kênh đầu vào tương tự kênh 2
22	A3	Đầu vào	Kênh đầu vào tương tự kênh 3
23	A4	Đầu vào	Kênh đầu vào tương tự kênh 4
24	A5	Đầu vào	Kênh đầu vào tương tự kênh 5
25	A6	Đầu vào	Kênh đầu vào tương tự kênh 6
26	A7	Đầu vào	Kênh đầu vào tương tự kênh 7
27	+ 5V	Đầu ra hoặc đầu vào	+ Đầu ra 5V (từ bộ điều chỉnh On-board) hoặc + 5V (đầu vào từ nguồn điện bên ngoài)
28	RESET	Đầu vào	Chân đặt lại, hoạt động ở mức thấp
29	GND	Nguồn	Chân nối mass
30	VIN	Nguồn	Chân nối với nguồn vào

*Bảng 1. 2. Bảng chân ICSP*

Tên pin Arduino Nano ICSP	Kiểu	Chức năng
MISO	Đầu vào hoặc đầu ra	Master In Slave Out
Vcc	Đầu ra	Cấp nguồn
SCK	Đầu ra	Tạo xung cho
MOSI	Đầu ra hoặc đầu vào	Master Out Slave In
RST	Đầu vào	Đặt lại, Hoạt động ở mức thấp
GND	Nguồn	Chân nối đất

+ Các chân: 1, 2, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15 và 16

Như đã đề cập trước đó, Arduino Nano có 14 ngõ vào/ra digital. Các chân làm việc với điện áp tối đa là 5V. Mỗi chân có thể cung cấp hoặc nhận dòng điện 40mA và có điện trở kéo lên khoảng 20-50kΩ. Các chân có thể được sử dụng làm đầu vào hoặc đầu ra, sử dụng các hàm pinMode (), digitalWrite () và digitalRead ().

Ngoài các chức năng đầu vào và đầu ra số, các chân này cũng có một số chức năng bổ sung.

+ Chân 1, 2: Chân nối tiếp

Hai chân nhận RX và truyền TX này được sử dụng để truyền dữ liệu nối tiếp TTL. Các chân RX và TX được kết nối với các chân tương ứng của chip nối tiếp USB tới TTL.

+ Chân 6, 8, 9, 12, 13 và 14: Chân PWM

Mỗi chân số này cung cấp tín hiệu điều chế độ rộng xung 8 bit. Tín hiệu PWM có thể được tạo ra bằng cách sử dụng hàm analogWrite ().

+ Chân 5, 6: Ngắt

Khi chúng ta cần cung cấp một ngắt ngoài cho bộ xử lý hoặc bộ điều khiển khác, chúng ta có thể sử dụng các chân này. Các chân này có thể được sử dụng để cho phép ngắt INT0 và INT1 tương ứng bằng cách sử dụng hàm

attachInterrupt (). Các chân có thể được sử dụng để kích hoạt ba loại ngắt như ngắt trên giá trị thấp, tăng hoặc giảm mức ngắt và thay đổi giá trị ngắt.

+ Chân 13, 14, 15 và 16: Giao tiếp SPI

Khi ta không muốn dữ liệu được truyền đi không đồng bộ, ta có thể sử dụng các chân ngoại vi nối tiếp này. Các chân này hỗ trợ giao tiếp đồng bộ với SCK. Mặc dù phần cứng có tính năng này nhưng phần mềm Arduino lại không có. Vì vậy, ta phải sử dụng thư viện SPI để sử dụng tính năng này.

+ Chân 16: Led

Khi bạn sử dụng chân 16, đèn led trên bo mạch sẽ sáng.

+ Chân 18, 19, 20, 21, 22, 23, 24, 25 và 26 : Ngõ vào/ra tương tự

Như đã đề cập trước đó UNO có 6 chân đầu vào tương tự nhưng Arduino Nano có 8 đầu vào tương tự (19 đến 26), được đánh dấu A0 đến A7. Điều này có nghĩa là ta có thể kết nối 8 kênh đầu vào tương tự để xử lý. Mỗi chân tương tự này có một ADC có độ phân giải 1024 bit (do đó nó sẽ cho giá trị 1024). Theo mặc định, các chân được đo từ mặt đất đến 5V. Nếu ta muốn điện áp tham chiếu là 0V đến 3.3V, có thể nối với nguồn 3.3V cho chân AREF (pin thứ 18) bằng cách sử dụng chức năng analogReference (). Tương tự như các chân digital trong Nano, các chân analog cũng có một số chức năng khác.

+ Chân 23, 24 như A4 và A5: chuẩn giao tiếp I2C

Khi giao tiếp SPI cũng có những nhược điểm của nó như cần 4 chân và giới hạn trong một thiết bị. Đối với truyền thông đường dài, cần sử dụng giao thức I2C. I2C hỗ trợ chỉ với hai dây. Một cho xung (SCL) và một cho dữ liệu (SDA). Để sử dụng tính năng I2C này, chúng ta cần phải nhập một thư viện có tên là Thư viện Wire.

+ Chân 18: AREF

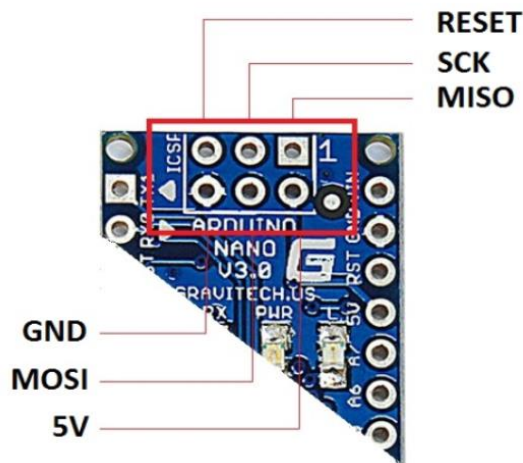
Điện áp tham chiếu cho đầu vào dùng cho việc chuyển đổi ADC.

+ Chân 28 : RESET

Đây là chân reset mạch khi chúng ta nhấn nút rên bo. Thường được sử dụng để được kết nối với thiết bị chuyển mạch để sử dụng làm nút reset.

## ICSP

ICSP là viết tắt của In Circuit Serial Programming , đại diện cho một trong những phương pháp có sẵn để lập trình bảng Arduino. Thông thường, một chương trình bộ nạp khởi động Arduino được sử dụng để lập trình một bảng Arduino, nhưng nếu bộ nạp khởi động bị thiếu hoặc bị hỏng, ICSP có thể được sử dụng thay thế. ICSP có thể được sử dụng để khôi phục bộ nạp khởi động bị thiếu hoặc bị hỏng.



Hình 1. 3. In Circuit Serial Programming

Mỗi chân ICSP thường được kết nối với một chân Arduino khác có cùng tên hoặc chức năng. Ví dụ: MISO của Nano nối với MISO / D12 (Pin 15). Lưu ý, các chân MISO, MOSI và SCK được ghép lại với nhau tạo nên header giao diện SPI.

Chúng ta có thể sử dụng Arduino để lập trình Arduino khác bằng ICSP này.

Arduino là ISP	ATMega328
Vcc/5V	Vcc
GND	GND
MOSI/D11	D11
MISO/D12	D12
SCK/D13	D13

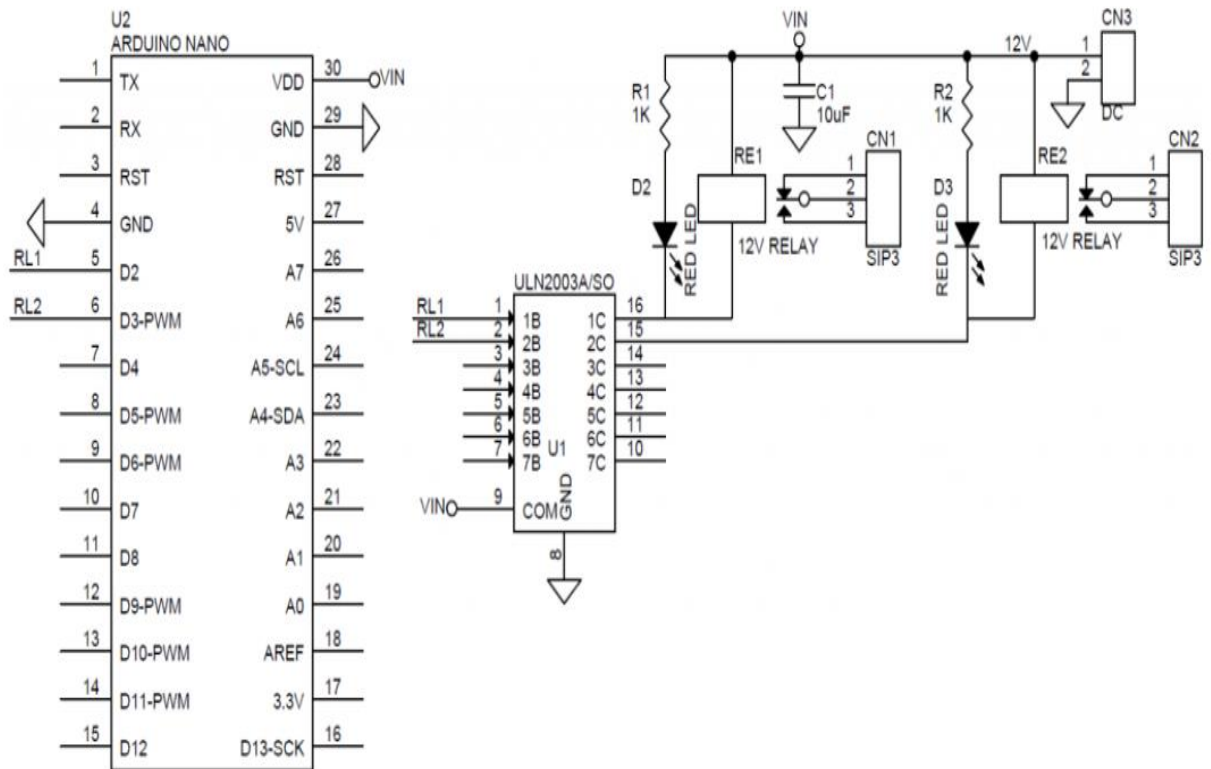


D10	Reset
-----	-------

*Bảng 1. 3. Bảng thông số kỹ thuật của Arduino Nano*

Vi điều khiển	ATmega328
Điện áp hoạt động	5 VDC
Tần số hoạt động	16 MHz
Dòng tiêu thụ	30 mA
Điện áp khuyến dùng	7 - 12 VDC
Điện áp giới hạn	6 - 20 VDC
Số chân Digital I/O	14(6 chân PWM)
Số chân Analog	8 (độ phân giải 10 bit)
Dòng tối đa trên mỗi chân I/O	40 mA
Dòng ra tối đa 5V	500 mA
Dòng ra tối đa 3.3V	50 mA
Bộ nhớ Flash	32 KB (ATmega328) với 2KB dùng bởi bootloader
SRAM	2 KB (ATmega328)
EEPROM	1 KB (ATmega328)
Kích thước	1.85cm x 4.3cm

### **1.1.2. Arduino Nano Schematic**



Hình 1. 4. Sơ đồ mở rộng chân cho Arduino Nano

## 1.2. NGUỒN NUÔI CHO ARDUINO NANO VÀ DRIVE A4988

### 1.2.1. Chuyển nguồn AC~220V sang DC-12V



Hình 1. 5. Mạch chuyển nguồn từ AC~220V sang DC-12V

#### Thông số kỹ thuật

- + Điện áp vào: 100V - 240V AC
- + Điện áp ra: 12V - 2A
- + Tần số: 50 - 60 Hz

### 1.2.2. Chuyển nguồn từ DC-12V sang DC-5V



Hình 1. 6. Bo mạch sử dụng IC LM2569

### Thông số cơ bản của IC LM2569:

IC LM2596 là một IC ổn áp dạng xung DC-DC. Điện áp đầu vào trong dải từ 4,5V-40V. Điện áp đầu ra điều chỉnh được trong khoảng từ 1,5V-37V, dòng điện áp đầu ra đạt 3A hiệu suất cao nhờ cơ chế băm xung ở tần số lên tới 150KHz. Trong quá trình hoạt động LM2596 luôn được đặt trong các chế độ bảo vệ quá nhiệt vào quá dòng.

### Thông số cơ bản:

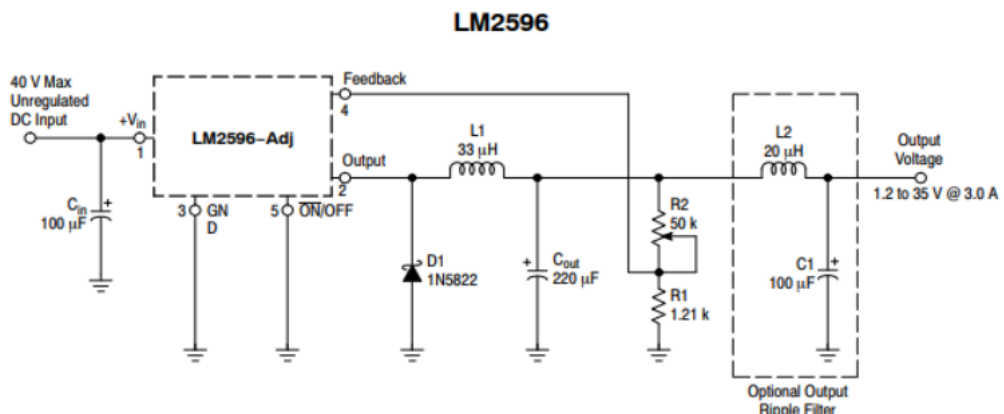
#### IC LM 2596 có 5 chân

- + Chân 1: Vin từ 4,5 ~ 40V
- + Chân 2: Vout
- + Chân 3: GND
- + Chân 4: Feedback ( chân phản hồi điện áp)
- + Chân 5: ON/OFF chân tắt bật mức logic

Điện áp đầu vào: 4.5-40V DC ( điện áp khuyến khích sử dụng <30V).

Điện áp đầu ra: 1,5~37V.

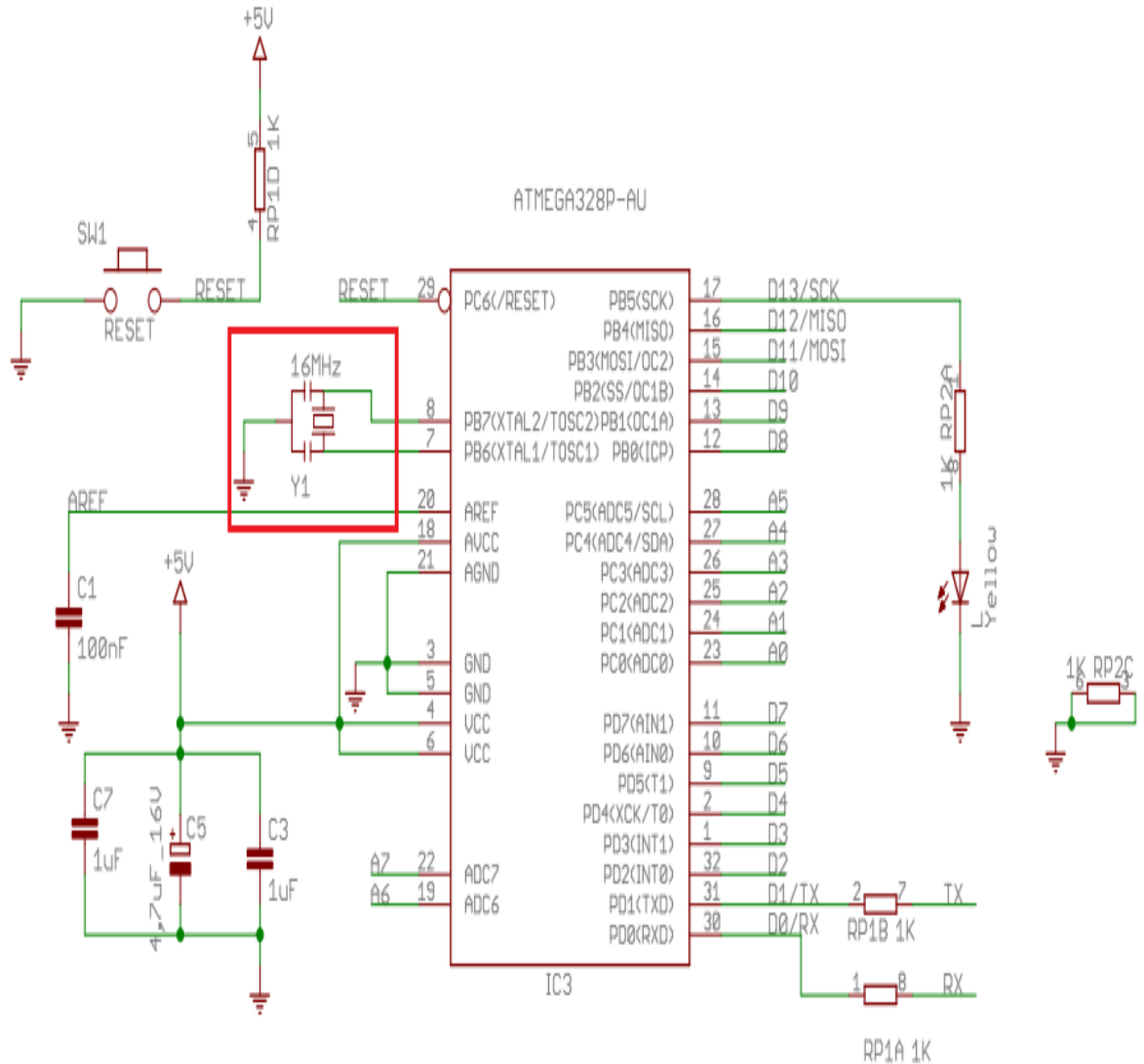
Dòng ra max( 3A).



Hình 1. 7. Sơ đồ mạch nguyên lý LM2596

### 1.3. MẠCH DAO ĐỘNG CHO ARDUINO NANO

Mạch dao động tạo ra các xung Clock giúp cho vi điều khiển hoạt động thực thi lệnh... Board mạch Arduino Nano sử dụng thạch anh 16Mhz làm nguồn lao động.



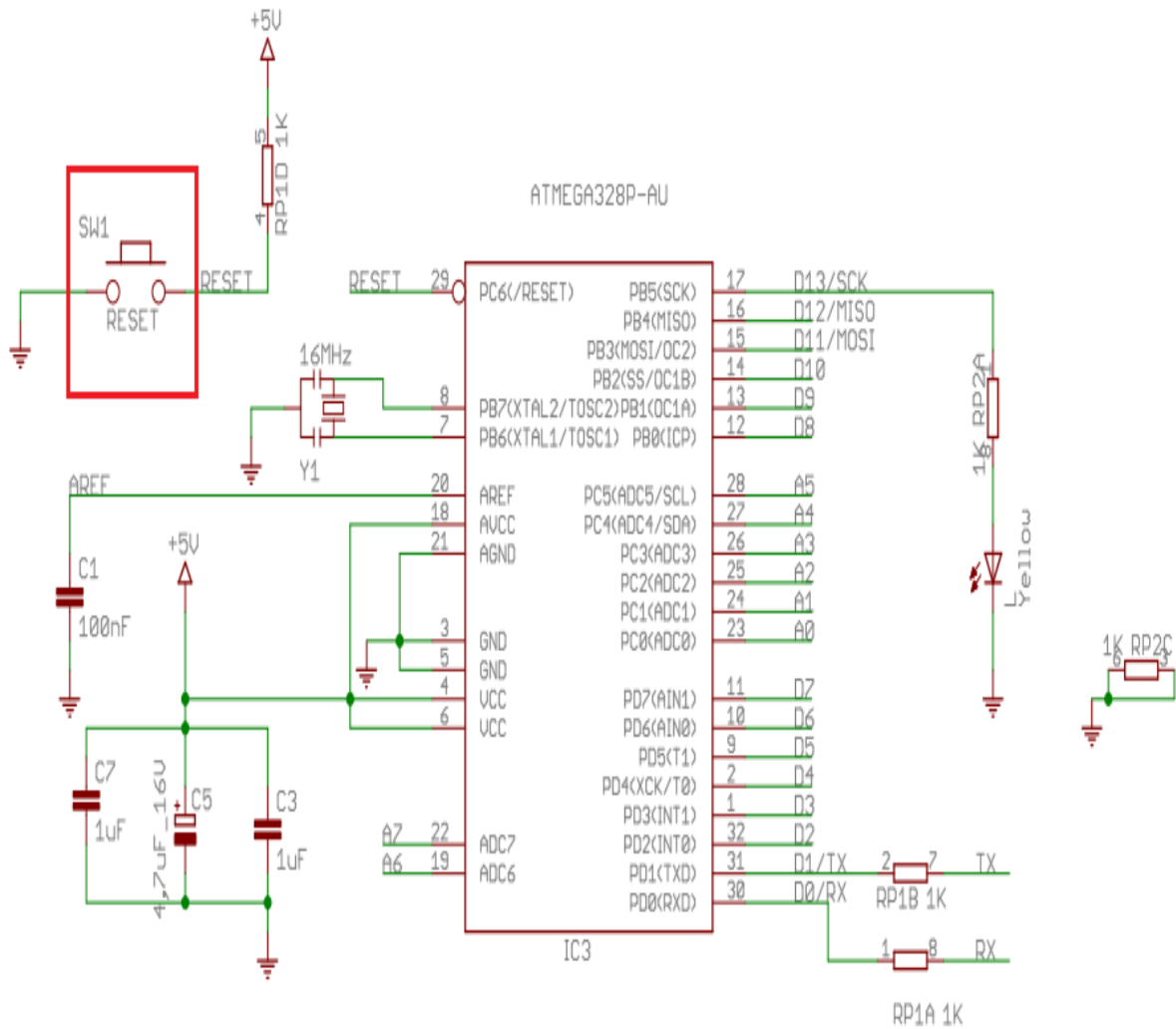
Hình 1. 8. Sơ đồ mạch nguyên lý Arduino Nano sử dụng dao động thạch anh

### 1.4. RESET

Để vi điều khiển thực hiện khởi động lại thì chân RESET phải ở mức logic LOW (~0V) trong 1 khoảng thời gian đủ yêu cầu.

Mạch reset của board Arduino Nano phải đảm bảo được 2 việc:

- Reset bằng tay: Khi nhấn nút, chân RESET nối với GND, làm cho MCU RESET. Khi không nhấn nút chân Reset được kéo 5V.



Hình 1. 9. Sơ đồ mạch Reset

- Reset tự động: Reset tự động được thực hiện ngay khi cấp nguồn cho vi điều khiển nhờ sự phối hợp giữa điện trở nối lên nguồn và tụ điện nối đất. Thời gian tụ điện cho chân RESET ở mức LOW trong 1 khoảng thời gian đủ để vi điều khiển thực hiện reset.
- Khởi động vi điều khiển trước khi nạp chương trình mới.

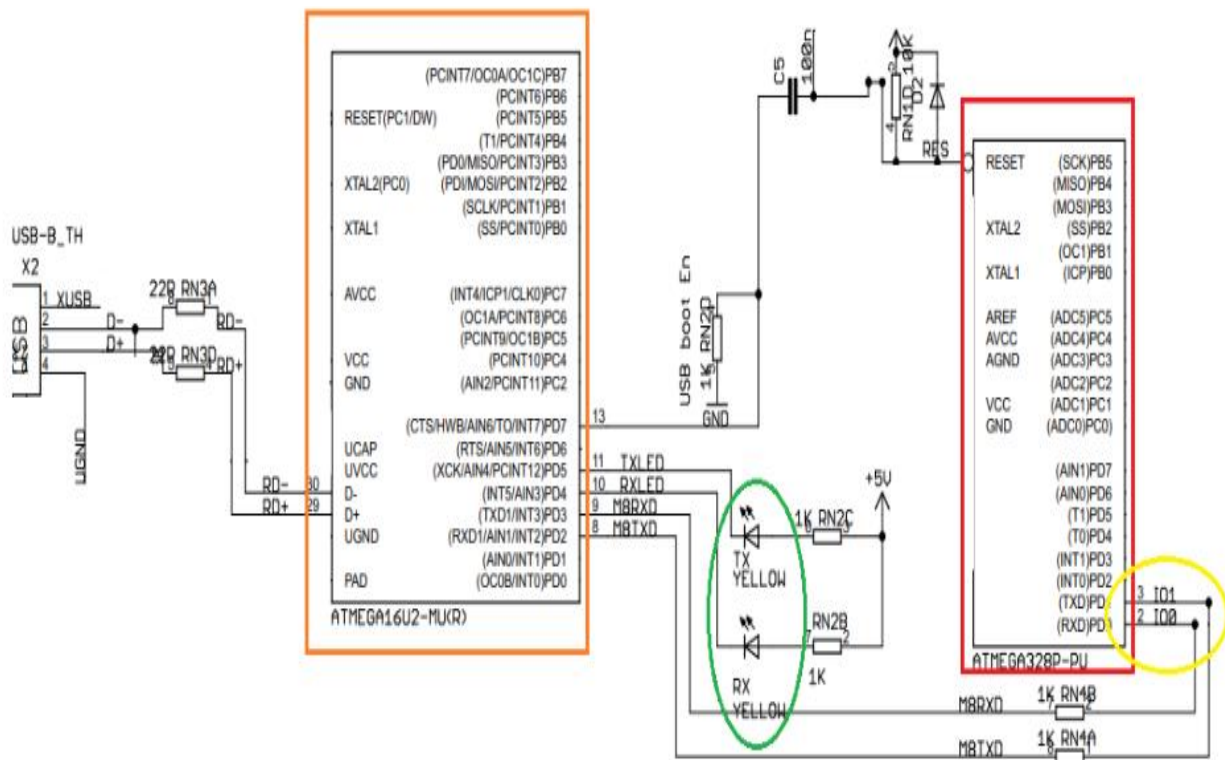
### 1.5. GIAO TIẾP MÁY TÍNH

Vi điều khiển Atmega328P trên Board Arduino Nano đã được nạp sẵn 1 bootloader, cho phép nhận chương trình mới thông qua chuẩn giao tiếp UART (Chân 0 và 1) ở giữa những giây đầu sau khi vi điều khiển Reset.

- Máy tính giao tiếp với Board mạch Arduino Nano qua chuẩn giao tiếp USB (D+/D-), thông qua một IC Driver. IC này có nhiệm vụ chuyển đổi

giao tiếp USB thành chuẩn giao tiếp UART để nạp chương trình hoặc giao tiếp truyền nhận dữ liệu với máy tính (Serial).

- Phần thiết kế mạch nạp có tích hợp thêm 2 đèn LED, nên khi nạp chương trình các bạn sẽ thấy 2LED này nhấp nháy. Còn khi giao tiếp, nếu có giữ liệu từ máy tính gửi xuống vi điều khiển thì đèn LED Rx sẽ nhấp. Còn nếu có dữ liệu từ vi điều khiển gửi lên máy tính thì đèn Tx sẽ nhấp.

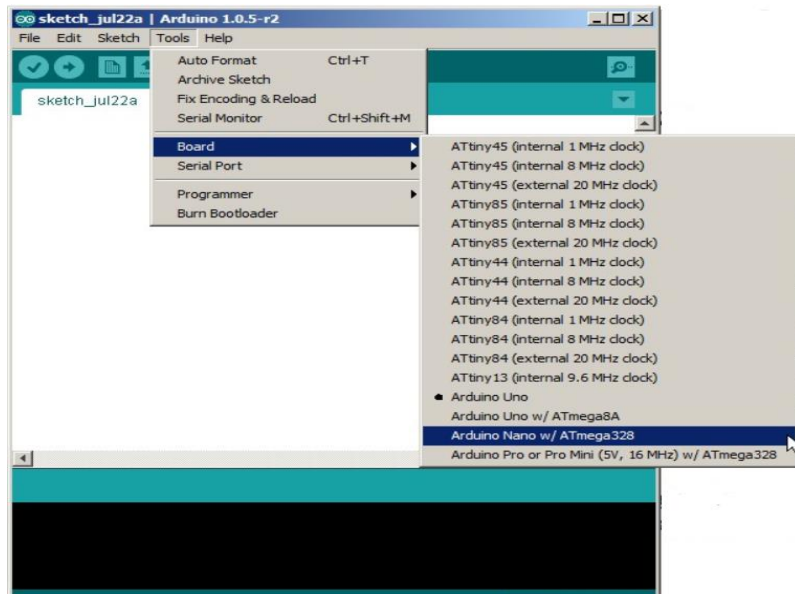


Hình 1. 10. Mạch kết nối máy tính

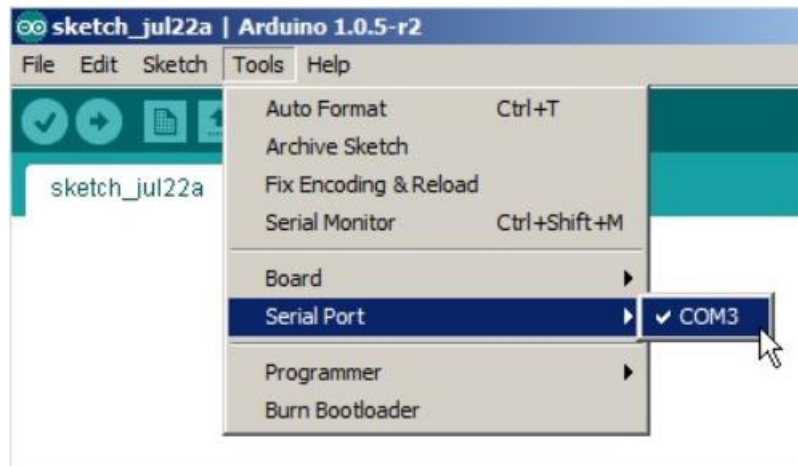
## 1.6. LẬP TRÌNH CHO ARDUINO NANO

**Arduino Nano** sử dụng chương trình Arduino IDE để lập trình, và ngôn ngữ lập trình cho Arduino cũng tên là Arduino. Tuy nhiên, nếu muốn lập trình cho Arduino Nano, bạn cần phải thực hiện một số thao tác trên máy tính.

1. Đầu tiên, bạn cần cài Driver của Arduino Nano và tải về bản Arduino IDE mới nhất cho máy tính, các bước cài đặt hoàn toàn tương tự như Arduino Uno R3. Sau khi cài đặt, bạn sẽ thấy một thông báo dạng cổng COM, sau đó chúng ta chọn COM để kết nối với máy tính.
2. Sau đó, bạn cần lại loại board và cổng Serial mới như hình sau là được.



Hình 1. 11. Lựa chọn dòng Arduino



Hình 1. 12. Chọn cổng giao tiếp cho Arduino

## CHƯƠNG 2 TỔNG QUAN VỀ C#

### 2.1. TÌM HIỂU VỀ NGÔN NGỮ C#

#### 2.1.1. C# là ngôn ngữ đơn giản

C# loại bỏ những phức tạp và rối rắm của những ngôn ngữ như Java và C++, bao gồm việc loại bỏ những cơ sở ảo (virtual base class). Chúng là những nguyên nhân gây ra sự nhầm lẫn hay dẫn đến những vấn đề cho các người phát triển C++. Nếu chúng ta là người học ngôn ngữ này đầu tiên thì chắc chắn là ta sẽ không trải qua những thời gian để học nó! Nhưng khi đó ta sẽ không biết được hiệu quả của ngôn ngữ C# khi loại bỏ những vấn đề trên. Ngôn ngữ C# đơn giản vì nó dựa trên nền tảng C và C++. Nếu chúng ta thân thiện với C và C++ hoặc thậm chí là Java, chúng ta sẽ thấy C# khá giống với diện mạo, cú pháp, biểu thức, toán tử và những chức năng khác được lấy trực tiếp từ ngôn ngữ C và C++, nhưng nó đã được cải tiến để làm cho ngôn ngữ đơn giản hơn. Một vài trong các sự cải tiến là loại bỏ các dư thừa, hay là thêm vào những cú pháp thay đổi.

Trong C++ có ba toán tử làm việc với các thành viên là ::, . , và - >. Để biết khi nào dùng ba toán tử này cũng phức tạp và dễ nhầm lẫn. Trong C# chúng được thay thế với một toán tử duy nhất gọi là (dot). Đối với người mới học thì điều này và những việc cải tiến khác làm bớt nhầm lẫn và đơn giản hơn.

Nếu chúng ta đã sử dụng Java và tin rằng nó đơn giản, thì chúng ta sẽ tìm thấy rằng C# cũng đơn giản. Hầu hết mọi người đều không tin rằng Java là ngôn ngữ đơn giản. Java và C++.

#### 2.1.2. C# là ngôn ngữ hiện đại

Điều gì làm cho một ngôn ngữ hiện đại. Những đặc tính như là xử lý ngoại lệ. Thu gom bộ nhớ tự động, những kiểu dữ liệu mở rộng, và bảo mật mã, nguồn là những đặc tính được mong đợi trong một ngôn ngữ hiện đại. C# chứa tất cả những đặc tính trên. Nếu là người mới tính trên phức tạp và khó hiểu. Tuy nhiên, cũng đừng lo lắng chúng ta sẽ dần dần được tìm hiểu những đặc tính qua các chương trong cuốn sách này.



Con trỏ được tích hợp vào ngôn ngữ C++. Chúng cũng là nguyên nhân gây ra những rắc rối của ngôn ngữ này. C# loại bỏ những phức tạp và rắc rối phát sinh bởi con trỏ. Trong C#, bộ thu gom bộ nhớ tự động và kiểu dữ liệu an toàn được tích hợp vào ngôn ngữ, sẽ loại bỏ những vấn đề rắc rối của C++.

### 2.1.3. C# là ngôn ngữ hướng đối tượng

Những đặc điểm chính của ngôn ngữ hướng đối tượng (Object-oriented language) là sự đóng gói (encapsulation), sự kế thừa (inheritance) và đa hình (polymorphism). C# hỗ trợ tất cả những đặc tính trên. Phần hướng bày chi tiết trong một chương riêng ở phần sau.

### 2.1.4. C# là ngôn ngữ mạnh mẽ và cũng mềm dẻo

Như đã đề cập trước, với ngôn ngữ C# chúng ta chỉ bị giới hạn ở chính bản hay là trí tượng của chúng ta. Ngôn ngữ này không đặt những ràng buộc lên những việc có thể làm. C# được sử dụng cho nhiều các dự án khác nhau như là tạo ra ứng dụng xử lý văn bản, ứng dụng đồ họa, bản tính, hay thậm chí những trình biên dịch cho các ngôn ngữ khác.

### 2.1.5. C# là ngôn ngữ ít từ khóa

C# là ngôn ngữ sử dụng giới hạn những từ khóa. Phần lớn các từ khóa được sử dụng để mô tả thông tin. Chúng ta có thể nghĩ rằng một ngôn ngữ có nhiều từ khóa thì sẽ mạnh hơn.

Điều này không phải sự thật, ít nhất là trong trường hợp ngôn ngữ C#, chúng ta có thể tìm thấy rằng ngôn ngữ này có thể được sử dụng để làm bất cứ nhiệm vụ nào.

Bảng sau liệt kê các từ khóa của ngôn ngữ C#.

abstract	<u>default</u>	<u>foreach</u>	<u>object</u>	<u>sizeof</u>	<u>unsafe</u>
as	<u>delegate</u>	<u>goto</u>	<u>operator</u>	<u>stackalloc</u>	<u>ushort</u>
base	<u>do</u>	<u>if</u>	<u>out</u>	<u>static</u>	<u>using</u>
bool	<u>double</u>	<u>implicit</u>	<u>override</u>	<u>string</u>	<u>virtual</u>
break	<u>else</u>	<u>in</u>	<u>params</u>	<u>struct</u>	<u>volatile</u>
byte	<u>enum</u>	<u>int</u>	<u>private</u>	<u>switch</u>	<u>void</u>
case	<u>event</u>	<u>interface</u>	<u>protected</u>	<u>this</u>	<u>while</u>
catch	<u>explicit</u>	<u>internal</u>	<u>public</u>	<u>throw</u>	
char	<u>extern</u>	<u>is</u>	<u>readonly</u>	<u>true</u>	

<u>checked</u>	<u>false</u>	<u>lock</u>	<u>ref</u>	<u>try</u>	
class	<u>finally</u>	<u>long</u>	<u>return</u>	<u>typeof</u>	
const	<u>fixed</u>	<u>namespace</u>	<u>sbyte</u>	<u>uint</u>	
continue	<u>float</u>	<u>new</u>	<u>sealed</u>	<u>ulong</u>	
decimal	<u>for</u>	<u>null</u>	<u>short</u>	<u>unchecked</u>	

*Bảng 1.2: Từ khóa của ngôn ngữ C#.*

### **2.1.6. C# là ngôn ngữ hướng module**

Mã nguồn C# có thể được viết trong những phần được gọi là những lớp, những lớp này chứa các phương thức thành viên của nó. Những lớp và những phương thức có thể được sử dụng lại trong ứng dụng hay các chương trình khác

Bằng cách truyền các mẫu thông tin đến những lớp hay phương thức chúng ta có thể tạo ra những mã nguồn dùng lại có hiệu quả.

### **2.1.7. C# sẽ là một ngôn ngữ phổ biến**

C# là một trong những ngôn ngữ lập trình mới nhất. Vào thời điểm cuốn sách này được viết, nó không được viết như là một ngôn ngữ phổ biến. Những ngôn ngữ này có một số lý do để trở thành một ngôn ngữ phổ biến. Một trong những dó do chính Microsoft và sự cam kết của NET.

Microsoft muốn ngôn ngữ C# trở nên phổ biến. Mặc dù một công ty không thể làm một sản phẩm trở nên phổ biến, nhưng nó có thể hỗ trợ. Cách đây không lâu Microsoft đã gặp sự thất bại về hệ điều hành Microsoft Bob. Mặc dù Microsoft muốn Bob trở nên phổ biến nhưng thất bại. C# thay thế tốt hơn để đem đến thành công sơ với Bob. Thật sự là không biết khi nào mọi người có trong công ty Microsoft sử dụng Bob trong công việc hằng ngày của họ. Tuy nhiên, với C# thì khác, nó được sử dụng bởi Microsoft. Nhiều sản phẩm của công ty này đã chuyển đổi và viết lại bằng C#. Bằng cách sử dụng ngôn ngữ này Microsoft đã xác nhận khả năng của C# cần thiết cho những người lập trình. Microsoft .NET là một lý do khác để đem đến sự thành công của C#.

NET là một sự thay đổi trong cách tạo và thực thi những ứng dụng.

Ngoài hai lý do trên ngôn ngữ C# cũng sẽ trở nên phổ biến do những đặc tính của ngôn ngữ này được đề cập trong mục trước như: đơn giản, hướng đối tượng, mạnh mẽ...

### **2.1.8. Ngôn ngữ C# và những ngôn ngữ khác**

Chúng ta đã từng nghe đến những ngôn ngữ khác như Visual Basic, C++ và Java. Có lẽ chúng ta cũng tự hỏi sự khác nhau giữa ngôn ngữ C# và những ngôn ngữ đó. Và cũng những ngôn ngữ đó. Và cũng tự hỏi tại sao lại lựa chọn ngôn ngữ này để học mà không chọn một trong những ngôn ngữ kia.

Có rất nhiều lý do và chúng ta hãy xem một số sự so sánh giữa ngôn ngữ C# với những ngôn ngữ khác giúp chúng ta phần nào trả lời được những thắc mắc.

Microsoft nói rằng C# mang đến sức mạnh của ngôn ngữ C++ với sự dễ dàng của ngôn ngữ Visual Basic. Có thể nó không dễ như Visual Basic, nhưng với phiên bản Visual Basic.NET (Version 7) thì ngang nhau. Bởi vì chúng được viết lại từ một nền tảng. Chúng ta có thể viết nhiều chương trình với ít mã nguồn hơn nếu dùng C#.

Mặc dù C# loại bỏ một vài các đặc tính của C++, nhưng bù lại nó tránh được những lỗi mà thường gặp trong ngôn ngữ C++. Điều này có thể tiết kiệm được hàng giờ hay thậm chí hàng ngày trong việc hoàn tất một chương trình. chúng ta sẽ hiểu nhiều về điều này trong các chương của giáo trình.

Một điều quan trọng khác với C++ là mã nguồn C# không đòi hỏi phải có tập tin header. Tất cả mã nguồn được viết trong khai báo một lớp. Như đã nói ở bên trên. NET runtime trong C# thực hiện việc thu gom bộ nhớ tự động. Do điều này nên việc sử dụng con trỏ trong C# ít quan trọng hơn trong C++. Những con trỏ cũng có thể được sử dụng trong C# kho đó những đoạn mã nguồn này sẽ được đánh dấu là không an toàn (unsafe code).

C# cũng từ bỏ ý tưởng đa kế thừa như trong C++. Và sự khác biệt khác là C# đưa thêm thuộc tính vào trong một lớp giống như trong Visual Basic.

Và những thành viên của lớp được gọi duy nhất bằng toán tử "." khác với C++ có nhiều cách gọi trong các tình huống khác nhau.

Một ngôn ngữ khác rất mạnh và phổ biến là Java, giống như C++ và C# được phát triển dựa trên C. Nếu chúng ta quyết định sẽ học Java sau này, chúng ta sẽ tìm được nhiều cái mà học từ C# có thể được áp dụng.

Điểm giống nhau C# và Java là cả hai cùng biên dịch ra mã trung gian: C# biên dịch ra MSIL còn Java biên dịch ra bytecode. Sau đó chúng được thực hiện bằng cách thông dịch hoặc biên dịch just-in-time trong từng máy ảo tương ứng.

Tuy nhiên, trong ngôn ngữ C# nhiều hỗ trợ được đưa ra để biên dịch mã ngôn ngữ trung gian sang mã máy. biên dịch mã ngôn ngữ C# chứa nhiều kiểu dữ liệu cơ bản hơn Java cũng cho phép nhiều sự mở rộng với kiểu dữ liệu giá trị.

Tương tự như Java, C# cũng từ bỏ tính đa kế thừa trong một lớp, tuy nhiên mô hình kế thừa đơn này được mở rộng bởi tính đa kế thừa nhiều giao diện.

## **2.2. CÁC BƯỚC CHUẨN BỊ CHO CHƯƠNG TRÌNH**

Thông thường, trong việc phát triển phần mềm, người phát triển phải tuân thủ theo quy trình phát triển phần mềm một cách nghiêm ngặt và quy trình này đã được thuần hoá. Tuy nhiên trong phạm vi của chúng ta là tìm hiểu một ngôn ngữ mới và viết những chương trình nhỏ thì không đòi hỏi khắt khe việc thực hiện theo quy trình. Nhưng để giải quyết được những vấn đề thì chúng ta cũng cần phải thực hiện đúng theo các bước sau. Đầu tiên là phải xác định vấn đề cần giải quyết. Nếu không biết rõ vấn đề thì ta không thể tìm được phương pháp giải quyết. Sau khi xác định được vấn đề, thì chúng ta có thể nghĩ ra các kế hoạch để thực hiện. Sau khi có một kế hoạch, thì có thể thực thi kế hoạch này. Sau khi kế hoạch được thực thi, chúng ta phải kiểm tra lại kết quả để xem vấn đề được giải quyết xong chưa. Logic này thường được áp dụng trong nhiều lĩnh vực khác nhau, trong đó có lập trình.

Khi tạo một chương trình trong C# hay bất cứ ngôn ngữ nào, chúng ta nên theo những bước tuần tự sau:

Xác định mục tiêu của chương trình.

Xác định những phương pháp giải quyết vấn đề. Tạo một chương trình để giải quyết vấn đề. Thực thi chương trình để xem kết quả.

Ví dụ mục tiêu để viết chương trình xử lý văn bản đơn giản, mục tiêu chính là xây dựng chương trình cho phép soạn thảo và lưu trữ những chuỗi ký tự hay văn bản. Nếu không có mục tiêu thì không thể viết được chương trình hiệu quả.

Bước thứ hai là quyết định đến phương pháp để viết chương trình. Bước này xác định những thông tin nào cần thiết được sử dụng trong chương trình, các hình thức nào được sử dụng. Từ những thông tin này chúng ta rút ra được phương pháp để giải quyết vấn đề.

Bước thứ ba là bước cài đặt, ở bước này có thể dùng các ngôn ngữ khác nhau để cài đặt, tuy nhiên, ngôn ngữ phù hợp để giải quyết vấn đề một cách tốt nhất

sẽ được chọn . Trong phạm vi của sách này chúng ta mặc định là dùng C#, đơn giản là chúng ta đang tìm hiểu nó!

Và bước cuối cùng là phần thực thi chương trình để xem kết quả.

### 2.2.1. Chương trình C# đơn giản

Để bắt đầu cho việc tìm hiểu ngôn ngữ C# và tạo tiền đề cho các chương sau, chương đầu tiên trình bày một chương trình C# đơn giản nhất.

Ví dụ 2.1 : Chương trình C# đầu tiên.

```
-----  
class ChaoMung{static void Main( ){// Xuat ra man hinh  
System.Console.WriteLine(“Chao Mung”);}}
```

Kết quả:

Chao Mung  
-----

Sau khi viết xong chúng ta lưu dưới dạng tập tin có phần mở rộng \*.cs(C sharp). Sau đó biên dịch và chạy chương trình.

Kết quả là một chuỗi “Chao Mung” sẽ xuất hiện trong màn hình console.

Các mục sau sẽ giới thiệu xoay quanh ví dụ 2.1.

### 2.2.2. Lớp, đối tượng và kiểu dữ liệu (type)

Điều cốt lõi của lập trình hướng đối tượng là tạo ra các kiểu mới.

Kiểu là một thứ được xem như trừu tượng. Nó có thể là một bảng dữ liệu, một tiểu trình, hay một nút lệnh trong một cửa sổ. Tóm lại kiểu được định nghĩa như một dạng vừa có thuộc tính chung (properties) và các hành vi ứng xử (behavior) của nó.

Nếu trong một ứng dụng trên Windows chúng ta tạo ra ba nút lệnh OK,Cancel,Help, thì thực chất là chúng ta đang dùng ba thể hiện của một kiểu nút trong Windows và các nút này cùng chia sẻ các thuộc tính và hành vi chung của chúng với nhau. Ví dụ , các nút có các thuộc tính như kích thước, vị trí, nhãn tên (label) tuy nhiên mỗi thuộc tính của một thể hiện không nhất thiết phải giống nhau, và thường thì chúng khác nhau, như nút OK có nhãn là “OK”, Cancel có nhãn là “Cancel”...Ngoài ra các nút này có các hành vi ứng xử chung như khả năng vẽ, kích hoạt đáp ứng các thông điệp nhân tùy theo từng chức năng đặc biệt

của từng loại thì nội dung ứng xử khác nhau, nhưng tất cả chúng được xem như là cùng một kiểu.

Cũng như nhiều ngôn ngữ lập trình hướng đối tượng khác, kiểu trong C# được định nghĩa là một lớp (class), và các thể hiện của từng lớp được gọi là đối tượng (object). Trong các chương kế tiếp sẽ trình bày các kiểu khác nhau ngoài kiểu lớp như kiểu liệt kê, cấu trúc và kiểu ủy quyền (delegates).

Quay lại chương trình ChaoMung trên, chương trình này chỉ có một kiểu đơn giản là lớp ChaoMung. Để định nghĩa một kiểu lớp trong C# chúng ta phải dùng từ khóa class, tiếp sau là tên lớp trong ví dụ trên lớp là ChaoMung. Sau đó định nghĩa các thuộc tính và hành động của lớp. Thuộc tính và hành động phải nằm trong dấu {}.

Ghi chú: Khai báo lớp trong C# không có dấu ; sau ngoặc } cuối cùng của lớp, và khác với lớp trong C/C++ là chia thành 2 phần header và phần định nghĩa. Trong C#, định nghĩa một lớp được gói trong dấu {} sau tên lớp và trong cùng một tập tin.

### 2.2.3. Phương thức

Hai thành phần chính cấu thành một lớp là thuộc tính hay tính chất và phương thức hay còn gọi là hành động ứng xử của đối tượng. Trong C# hành vi được định nghĩa như một phương thức thành viên của lớp.

Phương thức chính là các hàm được định nghĩa trong lớp. Do đó, ta còn có thể gọi các phương thức thành viên là các hàm thành viên trong một lớp. Các phương thức này chỉ ra rằng cá hành động mà lớp có thể làm được cùng với các cách thức làm hành động đó. Thông thường, tên của phương thức thường được đặt theo tên hành động, ví dụ như DrawLine() hay GetString(). Tuy nhiên trong ví dụ 2.1 vừa trình bày, chúng ta có hàm thành viên là Main() hàm này là hàm đặc biệt, không mô tả hành động nào của lớp hết, nó được xác định làm hàm đầu vào của lớp (entry point) thường được đặt theo tên hành động. ví dụ như DrawLine() hay GetString(). Tuy nhiên trong ví dụ 2.2 vừa trình bày, chúng ta có hàm thành viên Main() hàm này là hàm đặc biệt, không mô tả hành động nào của lớp hết, nó được xác định là hàm đầu vào của lớp (entry point) và được CRL gọi đầu tiên khi thực thi.

Ghi chú: Trong C#, hàm Main() được viết ký tự hoa đầu, và có thể trả về giá trị void hay in. Khi chương trình thực thi CLR gọi hàm Main() đầu tiên, Hàm Main() là đầu của chương trình và mỗi chương trình phải có một hàm Main().

Đôi khi chương trình có nhiều hàm Main() nhưng lúc này ta phải xác định các chỉ dẫn biên dịch để CLR biết đâu là hàm Main() đầu vào duy nhất trong chương trình. Việc chia sẻ phương thức được xem như là một sự giao ước giữa người tạo ra lớp và người sử dụng lớp này. Người xây dựng các lớp cũng có thể là người dùng lớp đó, nhưng không hoàn toàn như vậy. Vì vậy có thể các lớp này được xây dựng thành các thư viện chuẩn và cung cấp cho các nhóm phát triển khác. Do vậy việc tuân thủ theo các qui tắc là rất cần thiết. Để khai báo tên của phương thức, phải xác định kiểu giá trị về, tên của phương thức và cuối cùng là tham số cần thiết cho phương thức thực hiện.

#### 2.2.4. Chú thích

Một chương trình được viết lồi thì cần phải có chú thích các đoạn mã được các đoạn chú thích này sẽ không được biên dịch và cũng không tham gia vào chương trình. Mục đích chính là làm cho đoạn mã nguồn rõ ràng và dễ hiểu. Trong ví dụ 2.1 có một dòng chú thích `://` Xuất ra màn hình. Một chuỗi chú thích trên một dòng thì bắt đầu bằng ký tự `"/"`. Khi trình biên dịch gặp hai ký tự này thì sẽ bỏ qua dòng đó.

Ngoài ra C# còn cho phép kiểu chú thích cho một hay nhiều dòng, và ta phải khai báo `"/"` ở phần đầu chú thích và kết thúc chú thích là ký tự `"/"`.

Ví dụ 2.2 : Minh họa dùng chú thích trên nhiều dòng.

```
-----  
class ChaoMung  
{static void Main(){/* Xuất ra màn hình chuỗi 'chao mung'  
Sử dụng hàm WriteLine của lớp System.Console  
*/System.Console.WriteLine("Chao Mung");}}
```

Kết quả:

Chao Mung  
-----

Ngoài hai kiểu chú thích trên giống trong C/C++ thì C# còn hỗ trợ thêm kiểu thứ ba cũng là kiểu cuối cùng, kiểu này chứa các định dạng XML nhằm xuất ra tập tin XML khi biên dịch để tạo sơ liệu cho mã nguồn. Chúng ta sẽ bàn kiểu này trong các chương trình ở các phần tiếp.

### 2.2.5. Ứng dụng Console

Ví dụ đơn giản trên được gọi là ứng dụng console, ứng dụng nào giao tiếp với người dùng thông qua bàn phím và không có giao diện người dùng (UI), giống như các ứng dụng thường thấy trong Windows.

Trong các chương xây dựng các ứng dụng nâng cao trên Windows hay Web thì ta mới dùng các giao diện đồ họa. Còn để tìm hiểu về ngôn ngữ C# thuần túy thì cách tốt nhất là ta viết các ứng dụng console. Trong hai ứng dụng đơn giản trên ta đã dùng phương thức WriteLine() của lớp Console.

Phương thức này sẽ xuất ra màn hình dòng lệnh hay màn hình DOS chuỗi tham số đưa vào, cụ thể là chuỗi “Chao Mừng”.

Namespace

Như chúng ta đã biết .NET cung cấp một thư viện các lớp đồ sộ và thư viện này có tên là FCL (Framework Class Library).

Trong đó Console chỉ là một lớp nhỏ trong hàng ngàn lớp trong thư viện. Mỗi lớp có một tên riêng, vì vậy FCL có hàng ngàn tên như ArrayList, Dictionary, FileSelector,...

Điều này nảy sinh vấn đề, người lập trình không thể nào nhớ hết được tên của các lớp trong .NET Framework. Tệ hơn nữa là sau này có thể ta tạo lại một lớp trùng với lớp đã có chẳng hạn. Ví dụ trong quá trình phát triển một ứng dụng ta cần xây dựng một lớp từ điển và lấy tên là Dictionary, và điều này dẫn đến sự tranh chấp khi biên dịch vì C# chỉ cho phép một tên duy nhất.

Chắc chắn rằng khi đó chúng ta phải đổi tên của lớp từ điển mà ta vừa tạo thành một cái tên khác chẳng hạn như my Dictionary. Khi đó sẽ làm cho việc phát triển các ứng dụng trở nên phức tạp, cồng kềnh. Đến một sự phát triển nhất định nào đó thì chính là cơn ác mộng cho nhà phát triển.

Giải pháp để giải quyết vấn đề này là việc tạo ra một namespace, namespace sẽ hạn chế phạm vi của một tên, làm cho tên này chỉ có ý nghĩa trong vùng đã định nghĩa.



Giả sử có một người nói Tùng là một kỹ sư, từ kỹ sư phải đi kèm với một lĩnh vực nhất định nào đó, vì nếu không thì chúng ta sẽ không biết được là anh ta là một kỹ sư cầu đường, cơ khí hay phần mềm. Khi đó một lập trình viên C# sẽ bảo rằng Tùng là `CauDuong.KySu` phân biệt với `CoKhi.KySu` hay `PhanMem.KySu`. Namespace trong trường hợp này là `CauDuong` `CoKhi.PhanMem` sẽ hạn chế phạm vi của những từ theo sau. Nó tạo ra một vùng không gian để tên sau đó có nghĩa.

Tương tự như vậy ta cứ tạo các namespace để phân thành các vùng các vùng cho các lớp trùng tên không tranh chấp với nhau. Tương tự như vậy NET Framework có thể xây dựng một lớp Dictionary bên trong namespace `System.Collections`, và tương ứng ta có thể tạo một lớp Dictionary khác nằm trong namespace `ProgramCSharp.DataStructures`, điều này hoàn toàn không dẫn đến sự tranh chấp với nhau.

Trong ví dụ minh họa 1.2 đối tượng Console bị hạn chế bởi namespace bằng việc sử dụng mã lệnh: `System.Console.WriteLine()`;

### **2.2.6. Toán tử '.'**

Trong ví dụ 2.2 trên dấu '.' được sử dụng để truy cập đến phương thức hay dữ liệu trong một lớp ( trong trường hợp này phương thức là `WriteLine()`, và ngăn cách giữa tên lớp đến một namespace xác nhận ( namespace `System` và lớp là `console`). Việc thực hiện này theo từ trên xuống, trong đó mức đầu tiên Namespace là `System`, tiếp theo là lớp `Console`, và cuối cùng là truy cập đến các phương thức hay thuộc tính của lớp. Trong nhiều trường hợp namespace có thể được chia thành các namespace con gọi là subnamespace như `Configuration`, `Collections`, `Data` và còn rất nhiều nữa, hơn nữa trong namespace `Collection` còn chia thành nhiều namespace con nữa. Namespace giúp chúng ta tổ chức và ngăn cách những kiểu. Khi chúng ta viết một chương trình C# phức tạp, chúng ta có thể phải tạo một kiến trúc.

### **2.2.7. Từ khóa using**

Để làm cho chương trình gọn hơn, và không cần phải viết từng namespace Để làm cho chương trình gọn hơn và không cần viết từng namespace C# cung cấp từ khóa là `using`, sau từ khóa này là một namespace hay subnamespace với mô tả đầy đủ trong cấu trúc phân cấp của nó.

Ta có thể dùng dòng lệnh : `using System;` ở đầu chương trình và khi đó trong chương trình nếu chúng ta có dùng đối tượng Console thì không cần phải viết đầy đủ : `System.Console`. mà chỉ cần viết `Console`. thôi.

Ví dụ 2.3: Dùng khóa `using`

```
-----  
using System;class ChaoMung  
{static void Main(){//Xuat ra man hinh chuoi thong bao  
Console.WriteLine(“Chao Mung”);}}
```

Kết quả:

Chao Mung

Lưu ý rằng phải đặt câu `using System` trước định nghĩa lớp `ChaoMung`. Mặc dù chúng ta chỉ định rằng chúng ta sử dụng namespace `System` và không giống như các ngôn ngữ khác, không thể chỉ định rằng chúng ta sử dụng đối tượng `System.Console`

Ví dụ 2.4: Không hợp lệ trong C#.

```
-----  
using System.Console;class ChaoMung{static void Main()  
{//Xuat ra man hinh chuoi thong bao  
WriteLine(“Chao Mung”);}}
```

Đoạn chương trình trên khi biên dịch sẽ được thông báo một lỗi như sau:  
`error CS0138: A using namespace directive can only be applied to namespace; ‘System.Console’ is a class not a namespace.`

Cách biểu diễn namespace có thể làm giảm nhiều thao tác gõ bàn phím, Nhưng nó có thể sẽ không đem lại lợi ích nào bởi vì nó có thể làm xáo trộn những namespace có tên khác nhau. Giải pháp chung là chúng ta sử dụng từ khóa `using` với các namespace đã được xây dựng sẵn, các namespace do chúng ta tạo ra. những namespace này chúng ta nắm chắc suu liệu về nó. Còn đối với namespace do các hãng thứ ba cung cấp thì chúng ta không nên dùng từ khóa `using`.

Phân biệt chữ thường và chữ hoa

Cũng giống như C/C++, C# là ngôn ngữ phân biệt chữ thường với chữ hoa, đi đều này có nghĩa rằng hai câu lệnh WriteLine thì khác với WriteLine và cũng khác với WRITELINE.

Đáng tiếc là C# không giống như VB, Môi trường phát triển C# sẽ không tự sửa các lỗi này. Cũng giống như C/C++, C# là không tự sửa lỗi này, nếu chúng ta viết hai chữ với cách khác nhau thì cũng ta có thể đưa vào chương trình gỡ rối tìm ra các lỗi này. Để tránh việc lãng phí thời gian và công sức, người ta phát triển một số quy ước cho cách đặt tên biến, hằng, hàm, và nhiều định danh khác nữa. Quy ước trong giáo trình này dùng cú pháp lạc đà (camel notation) cho tên biến và cú pháp Pascal cho hàm, hằng, và thuộc tính.

Ví dụ:

Biến myDictionary theo cách đặt tên cú pháp lạc đà. Hàm DrawLine, thuộc tính ColorBackground theo cách đặt tên cú pháp Pascal.

#### **2.2.8. Từ khóa static**

Hàm Main() trong ví dụ minh họa trên có nhiều hơn một cách thiết kế. Trong minh họa này, hàm Main() được khai báo với kiểu trả về là void, tức là hàm này không trả về bất cứ giá trị nào cả. Đôi khi cần kiểm tra chương trình có thực hiện đúng hay không, người lập trình có thể khai báo hàm Main() trả về một giá trị nào đó để xác định kết quả thực hiện của chương trình.

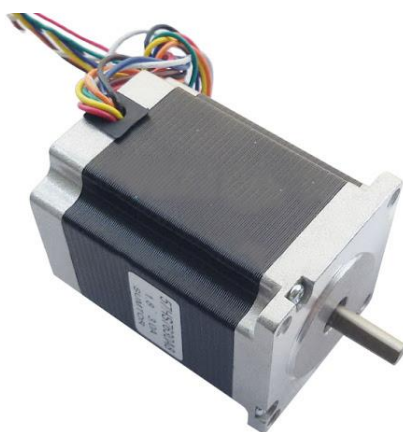
Trong khai báo của ví dụ trên có dùng từ khóa static: `static void Main(){.....}`

Từ khóa này chỉ ra rằng hàm Main() có thể được gọi mà không cần phải tạo đối tượng đối tượng Chào Mừng. Những vấn đề liên quan đến khai báo lớp phương thức, hay thuộc tính sẽ được trình bày chi tiết trong các chương tiếp theo.

# CHƯƠNG 3 THIẾT KẾ GIAO DIỆN ĐIỀU KHIỂN QUA TRÌNH CHỤP ẢNH TỰ ĐỘNG CỦA MÁY ĐO THÂN NHIỆT KHÔNG TIẾP XÚC

## 3.1. ĐỘNG CƠ BƯỚC

### 3.1.1. Cấu tạo và nguyên lí hoạt động của động cơ bước



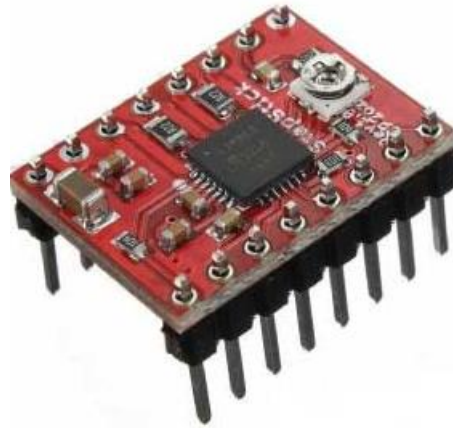
Hình 3. 1. Động cơ bước trong thực tế

Động cơ bước (Step Motor – Stepper Motor hay Stepping Motor) là một thiết bị cơ điện, nó chuyển đổi năng lượng điện thành cơ năng. Ngoài ra, nó là một động cơ điện không có chổi than, đồng bộ, có thể chia một vòng quay đầy đủ thành một số bước mở rộng. Trục của động cơ quay qua một góc cố định cho mỗi xung rời rạc. Khi một chuỗi xung được áp dụng, nó sẽ được chuyển qua một góc nhất định. Góc mà trục động cơ bước quay cho mỗi xung được gọi là góc bước, thường được biểu thị bằng độ.

Nếu góc bước càng nhỏ thì số bước trên mỗi vòng quay càng lớn và độ chính xác của vị trí thu được càng lớn. Các góc bước có thể lớn tới 90 độ và nhỏ đến 0,72 độ, tuy nhiên, các góc bước thường được sử dụng là 1,8 độ, 2,5 độ, 7,5 độ và 15 độ.

### 3.1.2. Điều khiển động cơ bước

A4988 là một trình điều khiển vi bước để điều khiển động cơ bước lưỡng cực có bộ dịch tích hợp để vận hành dễ dàng. Điều này có nghĩa là chúng ta có thể điều khiển động cơ bước chỉ với 2 chân từ bộ điều khiển của chúng ta hoặc một chân để điều khiển hướng quay và chân kia để điều khiển các bước.



Hình 3. 2. A4988 trình điều khiển bước

Driver cung cấp năm độ phân giải bước khác nhau: bước đủ,  $\frac{1}{2}$  bước,  $\frac{1}{4}$  bước,  $\frac{1}{8}$  bước và  $\frac{1}{16}$  bước. Ngoài ra, nó có một biến trở để điều chỉnh đầu ra hiện tại, tắt khi nhiệt độ quá cao và bảo vệ dòng điện chéo.

Nguồn vào của nó là từ 3 đến 5,5 V và dòng điện tối đa trên mỗi pha là 2A nếu được làm mát bổ sung tốt hoặc dòng điện liên tục 1A mỗi pha mà không cần tản nhiệt hoặc làm mát.

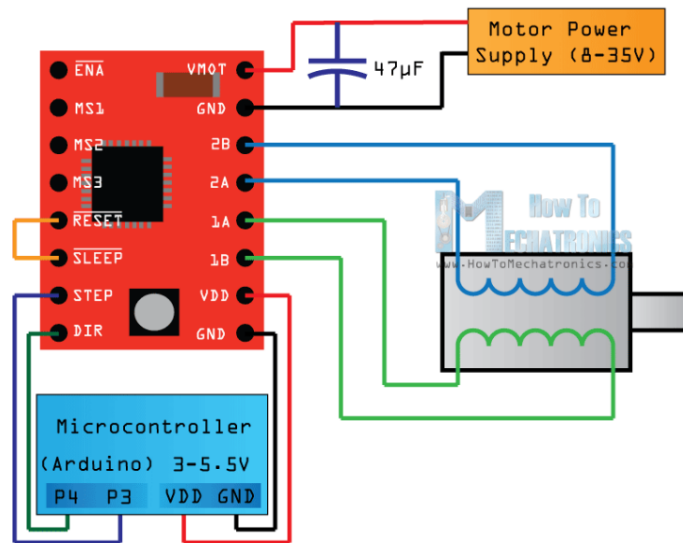
Bảng 3. 1. A4988- Thông số kỹ thuật

Minimum Logic Voltage:	3V
Maximum Logic Voltage:	5.5 V
Continuous current per phase:	1 A
Maximum current per phase:	2 A
Minimum Operating Voltage:	8 V
Maximum Operating Voltage:	35 V

### 3.1.3. Chân ra Driver A4988

Bắt đầu với 2 chân ở nút bên phải để cấp nguồn cho trình điều khiển, chân VDD và Ground mà chúng tôi cần kết nối chúng với nguồn điện từ 3 đến 5,5 V và trong trường hợp của chúng tôi sẽ là bộ điều khiển của chúng tôi, Arduino Board sẽ cung cấp 5 V. 4 chân sau đây để kết nối động cơ. Các chân 1A và 1B sẽ được kết nối với một cuộn dây của động cơ và chân 2A và 2B với cuộn dây khác của động cơ. Để cung cấp năng lượng cho động cơ, chúng tôi sử dụng 2 chân tiếp theo, Ground và VMOT mà chúng tôi cần kết nối chúng với Nguồn

cung cấp từ 8 đến 35 V và chúng tôi cũng cần sử dụng tụ tách rời với ít nhất 47 PhaF để bảo vệ board điều khiển khỏi các xung điện áp.



Hình 3. 3. Sơ đồ kết nối động cơ bước

Hai chân tiếp theo, Step và Direction là các chân mà chúng ta thực sự sử dụng để điều khiển chuyển động của động cơ. Chân Direction điều khiển hướng quay của động cơ và chúng ta cần kết nối nó với một trong các chân kỹ thuật số trên vi điều khiển, hoặc trong trường hợp của tôi, tôi sẽ kết nối nó với chân số 4 của Board Arduino.

Với chân Step, chúng ta điều khiển microsteps của động cơ và với mỗi xung được gửi tới chân này, động cơ sẽ di chuyển một bước. Vì vậy, điều đó có nghĩa là không cần bất kỳ chương trình phức tạp, board chuyển pha, dòng điều khiển tần số, v.v., vì trình dịch tích hợp của Driver A4988 đảm nhiệm mọi thứ. Ở đây chúng ta không nên nói chúng trong chương trình của mình.

Tiếp theo là chân SLEEP và mức logic thấp đặt board ở chế độ nghỉ để giảm thiểu mức tiêu thụ điện khi động cơ không được sử dụng. Chân RESET đặt trình dịch sang trạng thái Home được xác định trước. Trạng thái Home hoặc Vị trí Microstep Home này có thể được nhìn thấy từ Board dữ liệu A4988. Vì vậy, đây là vị trí ban đầu từ nơi động cơ khởi động và chúng khác nhau tùy thuộc vào độ phân giải microstep. Nếu trạng thái đầu vào của chân này ở mức logic thấp, tất cả các đầu vào STEP sẽ bị bỏ qua. Chân Reset là một chân nổi, vì vậy nếu chúng ta không có ý định điều khiển nó trong chương trình của mình, chúng ta cần kết nối nó với chân SLEEP để đưa nó lên mức cao và bật board.

Bảng 3. 2. Bảng mức chân cho MS

MS1	MS2	MS3	Resolution
LOW	LOW	LOW	Full Step
HIGH	LOW	LOW	Half Step
LOW	HIGH	LOW	Quarter Step
HIGH	HIGH	LOW	Eighth step
HIGH	HIGH	HIGH	Sixteenth Step

3 chân tiếp theo (MS1, MS2 và MS3) là để chọn một trong năm độ phân giải theo board ở trên. Các chân này có điện trở bên trong, vì vậy nếu chúng ta ngắt kết nối, board sẽ hoạt động ở chế độ bước đủ.

Cuối cùng, chân ENABLE được sử dụng để bật hoặc tắt các đầu ra FET. Vì vậy, mức cao sẽ giữ cho đầu ra bị vô hiệu hóa.

### 3.2. CÁC CẢM BIẾN SỬ DỤNG TRONG ĐỒ ÁN

#### 3.2.1. Cảm biến không tiếp xúc Ztemp TN901

##### + Mô tả sản phẩm

- Cảm biến nhiệt độ không tiếp xúc ZTEMP TN901 là loại cảm biến nhiệt độ hồng ngoại có độ chính xác và độ nhạy cao với độ nhiễu tín hiệu thấp thì đây là dòng cảm biến thích hợp sử dụng trong các ứng dụng như làm súng bắn nhiệt độ không tiếp xúc, trạm đo nhiệt độ không tiếp xúc, ...
- Hiện nay có rất nhiều loại cảm biến đo nhiệt độ không tiếp xúc nhưng chất lượng đo của sản phẩm đó không ổn định chính vì thế TL901 chính là giải pháp hữu hiệu cho việc tan hiệu năng đo nhiệt độ trên các thiết bị đo nhiệt độ từ xa.

##### + Thông số kĩ thuật

- Điện áp hoạt động: 3.3V
- Phạm vi đo: -33 ~ 220°C

- Sai số:  $\pm 2\%$ ,  $2^{\circ}\text{C}$
- Tỷ lệ đo ( distance:spot) 1:1
- Độ nhạy:  $1/16^{\circ}\text{C} = 0.0625$
- Tần số cập nhật dữ liệu: 1.4Hz
- Bước sóng:  $5\mu\text{m} \sim 14\mu\text{m}$

#### + Tính năng

TN901 được thiết kế nhỏ gọn với nhiều tính năng đặc biệt sau đây:

- Cảm biến này có thể bù nhiệt độ môi trường giúp tăng độ chính xác khi đo nhiệt độ trên con người.
- ZTEMP TL901 tích hợp tính năng SOC ( System On Chip ) để tối ưu hóa thiết kế khiến cảm biến trở nên nhỏ gọn và tiết kiệm chi phí nhất có thể.
- Sử dụng nguồn điện thấp 3.3V trong khi đó các dòng cảm biến khác phải sử dụng điện áp từ 5V trở lên.
- Dữ liệu đo được của loại cảm biến đo nhiệt độ này có thể được lưu vào trong EEPROM.

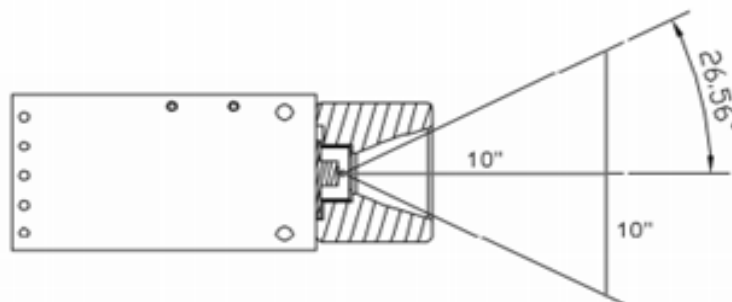
#### + Khoảng cách đo

Cảm biến nhiệt độ không tiếp xúc ZTEMP TN901 có thể đo chính xác trong khoảng từ 5 – 15 cm và tỷ lệ đo là 1:1.

Khi mà khoảng cách là 10 inch thì kích thước điểm đo luôn là 10 inch

Khi mà khoảng cách là 20 inch thì kích thước điểm đo luôn là 20 inch

Nói cách khác, góc nhìn là  $26,6^{\circ} \times 2 = 53,2^{\circ}$ .



Hình 3. 4. Khoảng đo của cảm biến TN901



### 3.2.2. Cảm biến hồng ngoại

Cảm biến vật cản hồng ngoại E18-D80NK dùng ánh sáng hồng ngoại để xác định khoảng cách tới vật cản cho độ phản hồi nhanh và rất ít nhiễu do sử dụng mắt nhận và phát tia hồng ngoại theo tần số riêng biệt. Cảm biến có thể chỉnh khoảng cách báo mong muốn thông qua biến trở, ngõ ra cảm biến ở dạng cực thu hờ nên cần thêm 1 trở treo lên nguồn ở chân Tín hiệu khi sử dụng.

*Thông số kỹ thuật:*

- Nguồn điện cung cấp: 5VDC.
- Khoảng cách phát hiện: 3 ~ 80cm.
- Có thể điều chỉnh khoảng cách qua biến trở.
- Dòng kích ngõ ra: 300mA.
- Ngõ ra dạng NPN cực thu hờ giúp tùy biến được điện áp ngõ ra, trở treo lên áp bao nhiêu sẽ tạo thành điện áp ngõ ra bấy nhiêu.
- Chất liệu sản phẩm: nhựa.
- Có led hiển thị ngõ ra màu đỏ.
- Kích thước: 1.8cm (D) x 7.0cm (L).

*Sơ đồ chân:*

- **Màu nâu:** VCC, nguồn dương 5VDC.
- **Màu xanh dương:** GND, nguồn âm 0VDC
- **Màu đen:** Chân tín hiệu ngõ ra cực thu hờ NPN, cần phải có trở kéo để tạo thành mức cao.

### 3.2.3. Cảm biến siêu âm (HC-SRF04)

Cảm biến siêu âm HC-SR04 (Ultrasonic Sensor) được sử dụng rất phổ biến để xác định khoảng cách vì giá thành rẻ và khá chính xác. Cảm biến siêu âm HC-SR04 sử dụng sóng siêu âm và có thể đo khoảng cách trong khoảng từ 2 -> 300cm.

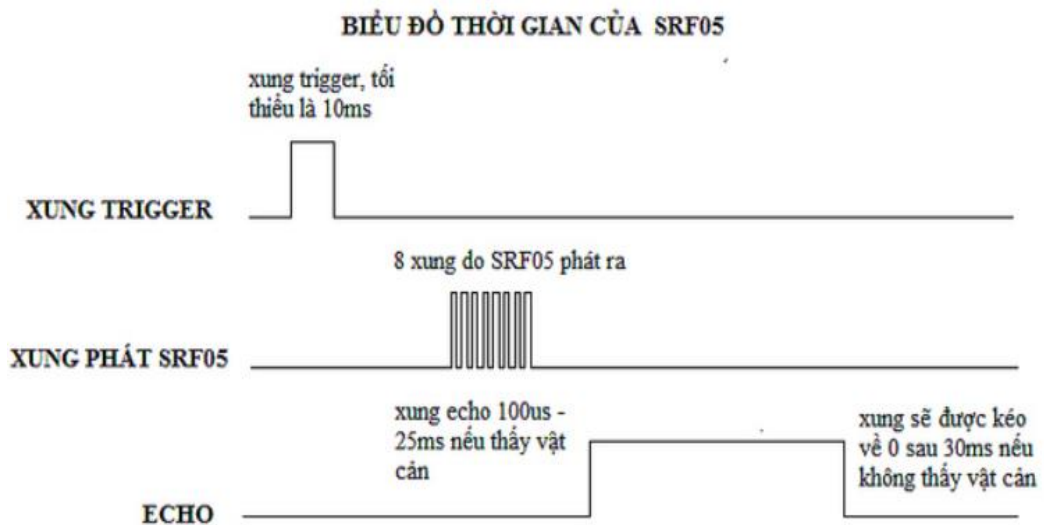


Hình 3. 5. HC-SR04

### Nguyên lý hoạt động:

Để đo khoảng cách, ta sẽ phát 1 xung rất ngắn (5 microSeconds) từ chân Trig. Sau đó, cảm biến siêu âm sẽ tạo ra 1 xung HIGH ở chân Echo cho đến khi nhận lại được sóng phản xạ ở pin này. Chiều rộng của xung sẽ bằng với thời gian sóng siêu âm được phát từ cảm biến và quay trở lại.

Tốc độ của âm thanh trong không khí là 340 m/s (hằng số vật lý), tương đương với 29,412 microSeconds/cm ( $106 / (340 \cdot 100)$ ). Khi đã tính được thời gian, ta sẽ chia cho 29,412 để nhận được khoảng cách.



Hình 3. 6. Biểu đồ thời gian

Bảng 3. 3. Các chân chức năng

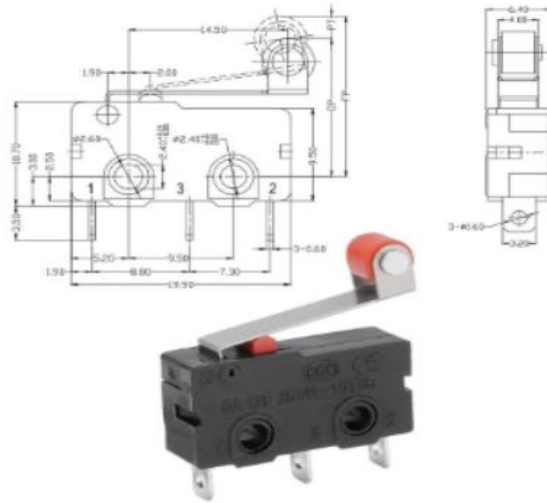
VCC	Cấp nguồn cho cảm biến (5V) hoặc 3.3V ở cảm biến 3V3
TRIGGER	Chân phát sóng âm. Là chu kỳ của của điện cao /thấp diễn ra.
ECHO	Trạng thái ban đầu là 0V, khi có tín hiệu trả về sẽ là 5V và sau đó trở về 0V
GND	Nối cực âm của mạch
OUT	Không sử dụng

#### 3.2.4. Công tắc hành trình

**Nguyên lý hoạt động công tắc hành trình:** ở điều kiện bình thường, tiếp điểm giữa chân COM và chân NC sẽ được đấu với nhau. Khi có lực tác động lên cần tác động thì tiếp điểm giữa chân COM + chân NC sẽ hở và chuyển qua chân COM + chân NO.

Công tắc hành trình là thiết bị giúp chuyển đổi chuyển động cơ thành tín hiệu điện để phục vụ cho quá trình điều khiển và giám sát. Ở đây trong đồ án này,

công tắc hành trình giúp xác định vị trí gốc cho cơ cấu lên xuống của hệ thống đo thân nhiệt tự động.



Hình 3. 7. Nguyên lí hoạt động của công tắc hành trình

### 3.3. KẾT NỐI VỚI LCD QUA GIAO THỨC I2C

#### + Thông số kỹ thuật LCD 16x2:

**LCD 16x2** được sử dụng để hiển thị trạng thái hoặc các thông số.

- LCD 16x2 có 16 chân trong đó 8 chân dữ liệu (D0 - D7) và 3 chân điều khiển (RS, RW, EN).
- 5 chân còn lại dùng để cấp nguồn và đèn nền cho LCD 16x2.
- Các chân điều khiển giúp ta dễ dàng cấu hình LCD ở chế độ lệnh hoặc chế độ dữ liệu.
- Chúng còn giúp ta cấu hình ở chế độ đọc hoặc ghi.

LCD 16x2 có thể sử dụng ở chế độ 4 bit hoặc 8 bit tùy theo ứng dụng ta đang làm.

#### + Module I2C Arduino



Hình 3. 8. Module I2C LCD 16x2

LCD có quá nhiều nhiều chân gây khó khăn trong quá trình đấu nối và chiếm dụng nhiều chân trên vi điều khiển.

**Module I2C LCD** ra đời và giải quyết vấn đề này.

Thay vì phải mất 6 chân vi điều khiển để kết nối với LCD 16x2 (RS, EN, D7, D6, D5 và D4) thì module IC2 bạn chỉ cần tốn 2 chân (SCL, SDA) để kết nối.

Module I2C hỗ trợ các loại LCD sử dụng driver HD44780(LCD 16x2, LCD 20x4, ...) và tương thích với hầu hết các vi điều khiển hiện nay.

+ **Ưu điểm**

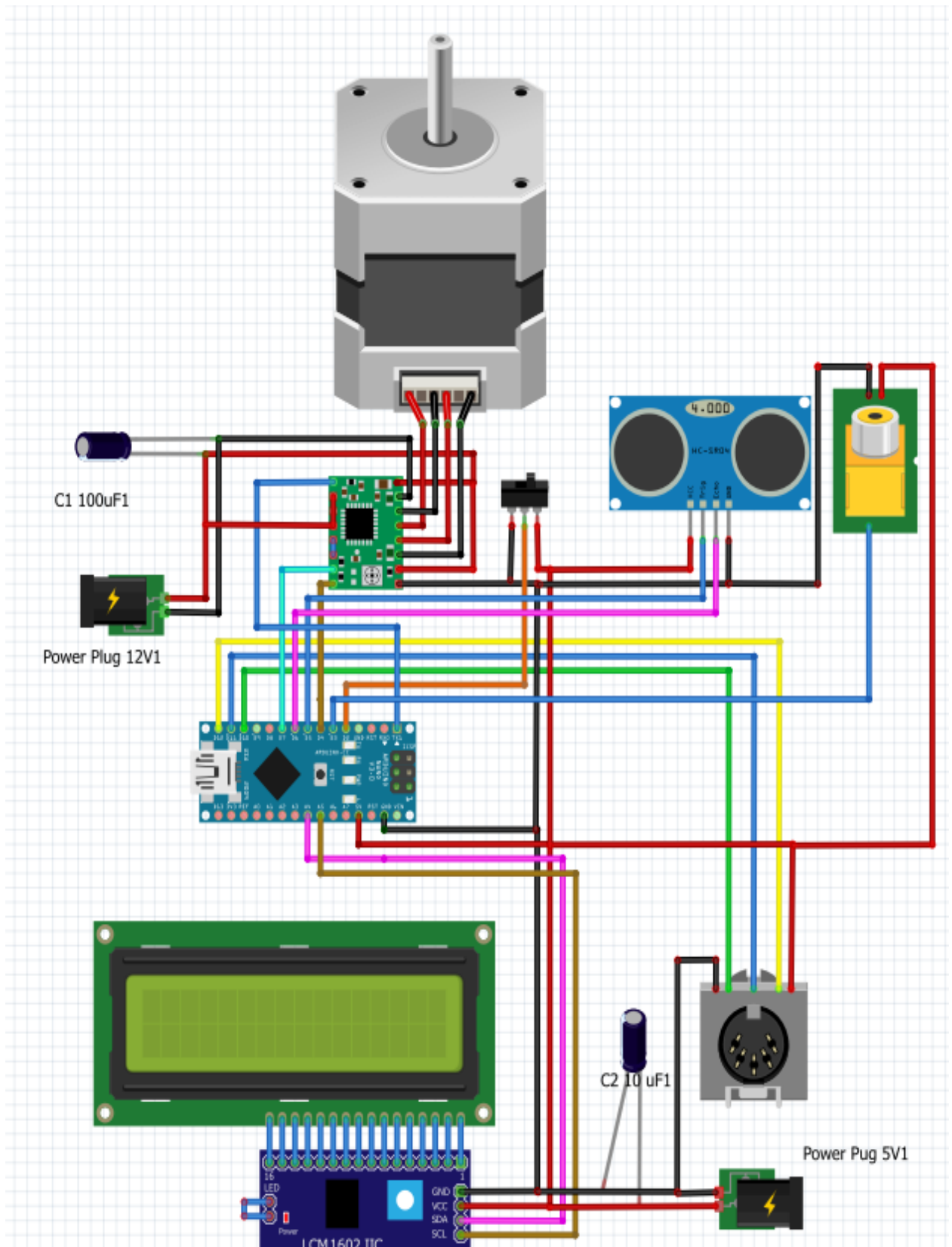
- Tiết kiệm chân cho vi điều khiển.
- Dễ dàng kết nối với LCD.

+ **Thông số kĩ thuật**

- Điện áp hoạt động: 2.5-6V DC.
- Hỗ trợ màn hình: LCD1602,1604,2004 (driver HD44780).
- Giao tiếp: I2C.
- Địa chỉ mặc định: 0X27 (có thể điều chỉnh bằng ngắn mạch chân A0/A1/A2).
- Tích hợp Jump chốt để cung cấp đèn cho LCD hoặc ngắt.
- Tích hợp biến trở xoay điều chỉnh độ tương phản cho LCD.

Để sử dụng màn hình LCD giao tiếp I2C sử dụng Arduino thì ta cần cài đặt thư viện **Liquidcrystal\_I2C**.

### **3.4. SƠ ĐỒ NGUYÊN LÝ MẠCH VÀ GIAO DIỆN NGƯỜI DÙNG TRÊN C#**



Hình 3. 9. Sơ đồ nguyên lý mạch

### 3.5. CODE THAM KHẢO

```
#include <Wire.h>
#include <LiquidCrystal_I2C.h>
//Include the AccelStepper library:
#include <AccelStepper.h>
// Define stepper motor connections and motor interface type. Motor interface type must be set to
1 when using a driver:
#define dirPin 4
#define stepPin 7
#define IR 3
#define motorInterfaceType 1
#define STEPS 3200
AccelStepper stepper1 = AccelStepper(motorInterfaceType, stepPin, dirPin);
//-----
LiquidCrystal_I2C lcd(0x3f,16,2);
//-----
//-----
const int limitSW=2;
const int LED=13;
const int highSpeed=600;
const int lowSpeed=100;
const long stepMax=42000;
//
const int pingPin = 5; // Trigger Pin of Ultrasonic Sensor
const int echoPin = 6 ; // Echo Pin of Ultrasonic Sensor
//
float Obj,Envi;
char index=0;
//
long steps=0; //so buoc da dich chuyen
//
int out=0;
//
long curPos=0;
void setup() {
  Serial.begin(9600);

  lcd.init(); //Khởi động màn hình. Bắt đầu cho phép Arduino sử dụng màn hình
  lcd.backlight(); //Bật đèn nền
  //lcd.print(" XIN CHAO BAN ");
  //
  pinMode (limitSW,INPUT_PULLUP);

  pinMode (IR,INPUT_PULLUP);

  //
  stepper1.setMaxSpeed(5000);
  stepper1.setAcceleration(5000);
  //
  pinMode(pingPin, OUTPUT);
  pinMode(echoPin, INPUT );
```

```

//
goHome();
}
//-----

//-----
int disMes()
{
    int cm;
    return cm;
}
//-----

int docCbKc()
{
    //do khoang cach
    // neu loi cam bien thi do lai 3 lan
    //tra ve 1000 neu co loi
    //khong loi tra ve gia tri (cm)
    int d=0;
    char doc=0;
    d=disMes();
    if (d==1000)
    {
        for (doc=0;doc++;doc<3) //thu doc lai 3 lan
        {
            // delay(100);
            d=disMes();
            if (d!=1000)
                return d;
        }
        return 1000; //tra ve ma: loi cam bien (1000)
    }
    else
    {
        return d; //tra ve khoang cach neu khong loi
    }
}

//-----

void up(long buoc)
{
    curPos=stepper1.currentPosition();
    ///stepper1.setCurrentPosition(0);
    stepper1.setMaxSpeed(5000);
    stepper1.setAcceleration(5000);
    // chuyen dong len n buoc
    while((stepper1.currentPosition() != (curPos+buoc))&&(stepper1.currentPosition() <=stepMax))
//bao ve qua so buoc
    {
        stepper1.setSpeed(1000);
        stepper1.runSpeed();
    }
}

```

```

}
//-----
void down(long buoc)
{
//stepper1.setCurrentPosition(0);
stepper1.setMaxSpeed(4000);
stepper1.setAcceleration(4000);
// chuyen dong len n buoc
curPos=stepper1.currentPosition();
long s=0;
if (curPos>buoc)
s=curPos-buoc;
else
s=0;

while(stepper1.currentPosition() != s)
{
stepper1.setSpeed(-1000);
stepper1.runSpeed();
if (!digitalRead(limitSW)) //gap cong tac hanh trinh
break;
}
}
//-----
void goHome()
{

stepper1.setMaxSpeed(5000);
stepper1.setAcceleration(5000);
stepper1.setCurrentPosition(0);

while(stepper1.currentPosition() != 0) //chay len 1 doan nho
{
stepper1.setSpeed(400);
stepper1.runSpeed();
}

out=0;
while(out==0) //chay xuong toc do cao
{
stepper1.setSpeed(-3200);
stepper1.runSpeed();
if (!digitalRead(limitSW)) //cong tac hanh trinh
out=1;
}
stepper1.setCurrentPosition(0);
//
while(stepper1.currentPosition() != 300) //chay len toc do cham
{
stepper1.setSpeed(200);
stepper1.runSpeed();
}
}

```



```

}
out=0;
while(out==0) //chay xuong cham den khi cham cong tac hanh trinh
{
  stepper1.setSpeed(-200);
  stepper1.runSpeed();
  if (!digitalRead(limitSW))
    out=1;
}
stepper1.setCurrentPosition(0); //dat vi tri goc
index=0;
steps=0;
}
//-----
float temMes()
{

float ObjTem[3],EnTem[3];
return Obj;
}
//-----
bool checkIR()
{
  //co nguoi tra ve : 1
  //khong nguoi tra ve: 0

  if (!digitalRead(IR))
  {
    delay(2);
    if (!digitalRead(IR)) //check to ensure
      return 1;
    else
      return 0;
  }
  else
    return 0;
  /*
  if ((PIND & 0xf7)==1)
    return 1;
  else
    return 0;
  */
}
// -----chuong trinh chinh-----

void loop() {
float t;
char check=0;
int d=0;
char r=0;
digitalWrite(LED,digitalRead(IR)); //test
if (index==0)

```

```

{
  lcd.setCursor(0,0);
  lcd.print(" MAY DO THAN ");
  lcd.setCursor(0,1);
  lcd.print(" NHIET TU DONG ");
  delay(1000);
  d=docCbKc();
  if ((d >0)&&(d<50)&&(checkIR()))
  {
    delay(500);
    d=docCbKc();
    if ((d >0)&&(d<50)&&(checkIR()))
      index=1;
    // gui du lieu len may tinh de phat ra am thanh
    Serial.print("1");
    lcd.clear();
    lcd.print(" XIN CHAO BAN ");
    delay(1000);
    lcd.setCursor(0,0);
    lcd.print("VUI LONG GIU KC ");
    lcd.setCursor(0,1);
    lcd.print(" 20 CM - 30 CM ");
    steps=0;
  }
}
else
if (index==1)
{

  //stepper1.setCurrentPosition(steps);
  check=0;
  stepper1.setMaxSpeed(5000);
  stepper1.setAcceleration(5000);
  while(stepper1.currentPosition() !=stepMax)
  {
    stepper1.setSpeed(3000);

    //steps++;
    //delayMicroseconds(10);
    //if(!checkIR())
    if (digitalRead(IR)) //neu co nguoi
    {
      delay(2);
      if (digitalRead(IR))
      {
        check=1; //thu check
        break; //out while
      }
    }
    stepper1.runSpeed();
  }
}

```

```

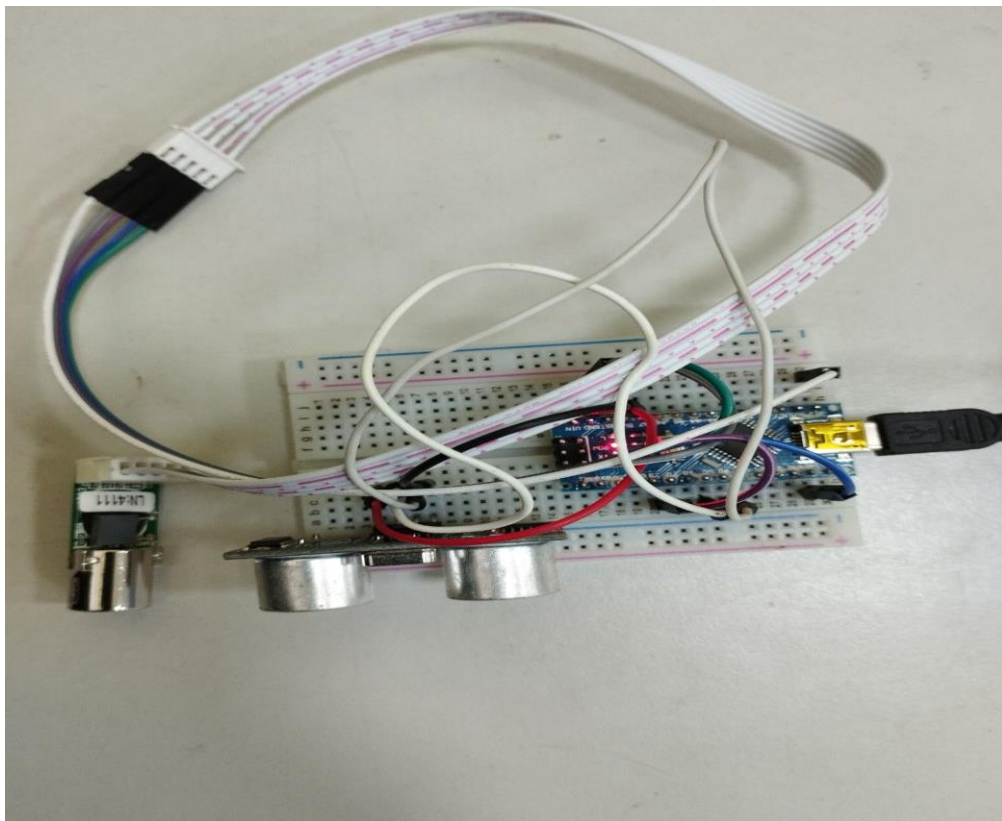
if (check==0)
{
  Serial.print("2");
  lcd.clear();
  lcd.print("NGOAI PHAM VI HT");
  delay(4000);
  index=2; //thoat while
  check=0;
}
else
if (check==1)
{
  up(500);
  if (!checkIR()) //neu ko co nguoi
  {
    down(3600);
    if (checkIR()) //lui xuong thi co nguoi
    {
      Serial.print("3");
      //
      lcd.clear();
      lcd.print("TH. NHIET:");
      t=temMes();
      lcd.print(t);
      lcd.print("oC");
      lcd.setCursor(0,1);
      if (t>=37.8)
      {
        lcd.print(" THAN NHIET CAO ");
        Serial.print("4");
      }
      else
      if (t>=36)
      {
        lcd.print(" THAN NHIET BT ");
        Serial.print("5");
      }
      delay(6000);
      index=2; //to reset part
    }
  }
else
{
  Serial.print("6");
  //
  lcd.clear();
  lcd.print(" KHONG TIM THAY ");
  lcd.setCursor(0,1);
  lcd.print(" DOI TUONG ");
  delay(5000);
  index=2; //to reset
}
}
}

```

```

else
{
  down(500); //tro ve vi tri cu sau khi UP
  check=0; //neu co nguoi thi tiep tục do
  Serial.print("7");
}
}
}
else
if (index==2)
{
  Serial.print("8");
  //
  lcd.clear();
  lcd.print("VE VI TRI GOC ");
  goHome();
  lcd.setCursor(0,0);
  lcd.print("DA XONG ");
  index=0;
  steps=0;
  check=0;
  delay(1000);
}
}

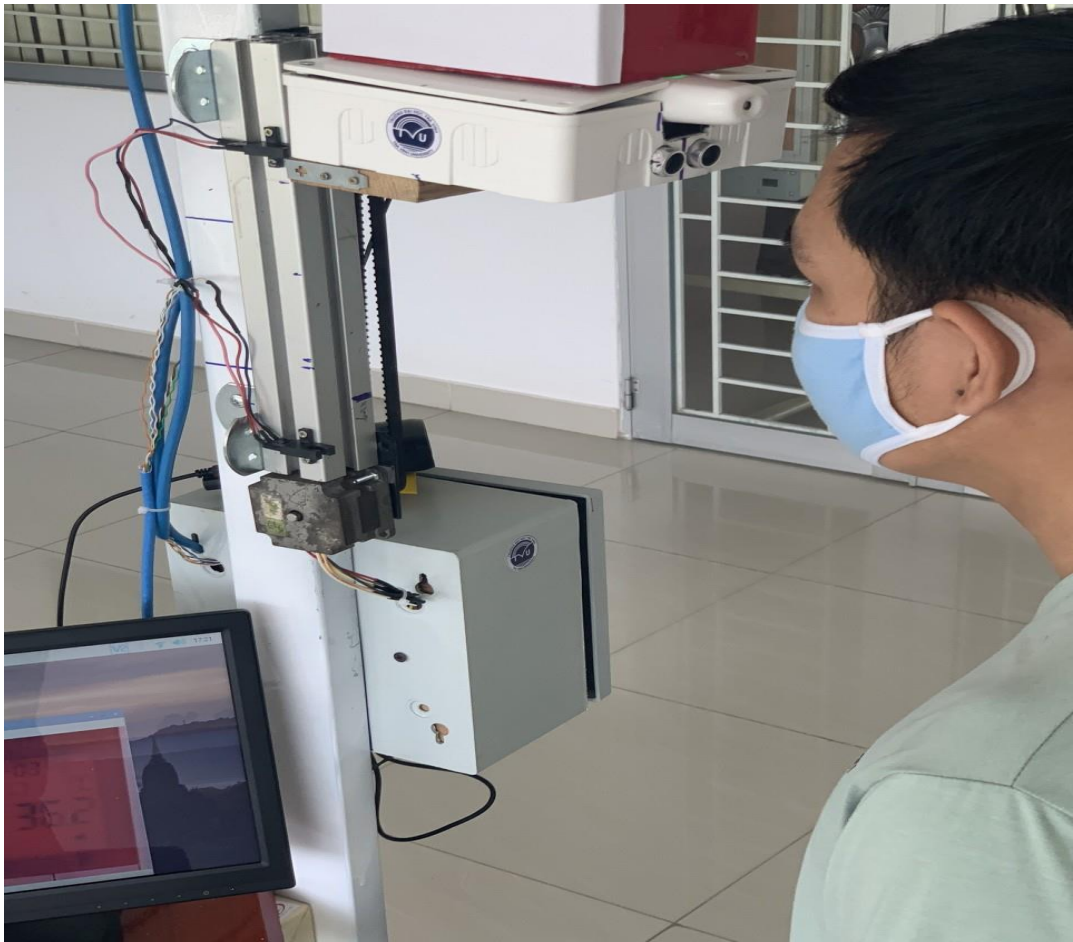
```



1.Hình ảnh mô hình chạy test Arduino Nano của cảm biến siêu âm và cảm biến đo thân nhiệt không tiếp xúc.



1.2 Hình ảnh cảm biến trả kết quả về máy tính.



1.3 Hình ảnh sản phẩm đo thân nhiệt không tiếp xúc.

## KẾT LUẬN

Sau thời gian 3 tháng làm đề án với sự hướng dẫn tận tình của thầy giáo:Th.Nguyễn Đoàn Phong, em đã hoàn thành đề tài được giao “ **Thiết kế giao diện điều khiển quá trình chụp ảnh tự động của máy đo thân nhiệt không tiếp xúc** “ .

Trong đề án này em đã tìm hiểu được các vấn đề:

- Biết được cách sử dụng phần mềm arduino
- Hiểu biết về dùng ngôn ngữ C #

Quá trình thực hiện đề án đã giúp em củng cố lại những kiến thức mà mình đã học. Ngoài ra qua quá trình tìm hiểu thực tế bên ngoài để hoàn thành đề án đã giúp em có thêm những kiến thức thực tế. Do thời gian làm đề án ngắn và kiến thức còn hạn chế nên trong đề án còn có những thiếu sót nhất định. Vì vậy, em rất mong được sự góp ý, bổ sung của các thầy cô giáo để đề án của em được hoàn thiện hơn.

Em xin chân thành cảm ơn !

Hải phòng, ngày....tháng.....năm 2023

Sinh viên

**ĐỖ ĐẠI DƯƠNG**

## TÀI LIỆU THAM KHẢO

- 1) ARDUINO cho người mới bắt đầu – Thạc Sỹ Huỳnh Văn Phú ( 2015 )
- 2) Báo cáo thực hành lập trình hệ thống – Thạc Sỹ Hồ Hải Nghĩa ( 2013)
- 3) Thiết kế và thi công hệ thống iot phục vụ cho nông nghiệp ứng dụng gateway – Sinh Viên Trần Quốc Tiến ( 2015 )
- 4) Nghiên cứu xây dựng mạng cảm biến không dây dựa trên giao thức LEACH và ZigBee – Thạc Sỹ Nguyễn Hoài Sơn ( 2018 )
- 5). Giáo trình cảm biến. (2000)  
Phan Quốc Phô, Nguyễn Đức Chiến. – NXB Khoa học và kỹ thuật.
- 6). Vi điều khiển cấu trúc lập trình và ứng dụng. (2008)  
Kiều Xuân Thực, Vũ Thị Hương, Vũ Trung Kiên – NXB Giáo Dục.
- 7). Kỹ thuật điện tử. (1999) - Đỗ Xuân Thụ. – NXB giáo dục.
- 8). Website <http://arduino.vn/>
- 9). Website <http://codientu.org/>
- 10). <http://www.dientuviệtnam.net/forums>
- 11) <http://www.tailieu.vn>
- 12) <http://luanvan.net.vn>
- 13) <https://123doc.net>