

**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 : Nguyễn Trọng Ngãi
Giảng viên hướng dẫn : ThS Đỗ Anh Dũng

Hải Phòng -2022

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

**THIẾT KẾ HỆ THỐNG ĐO KHOẢNG CÁCH DÙNG
CẢM BIẾN SIÊU ÂM**

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

**Sinh viên thực hiện: Nguyễn Trọng Ngãi
Giảng viên hướng dẫn: ThS Đỗ Anh Dũng**

Hải Phòng - 2022

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

NHIỆM VỤ ĐỀ TÀI TỐT NGHIỆP

Sinh viên : Nguyễn Trọng Ngãi - **MSV :** 2013102012

Lớp : DCL 2401

Ngành: Điện Tự Động Công Nghiệp

Tên đề tài : Thiết kế hệ thống đo khoảng cách dùng cảm biến siêu âm

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 tính toán.

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

3. Địa điểm thực tập tốt nghiệp.

.....

CÁC CÁN BỘ HƯỚNG DẪN ĐỀ TÀI TỐT NGHIỆP

Họ và tên :

Học hàm, học vị :

Cơ quan công tác : Trường Đại học quản lý và công nghệ Hải Phòng

Nội dung hướng dẫn:

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

Đề tài tốt nghiệp được giao ngày 20 tháng 6 năm 2022

Yêu cầu phải hoàn thành xong trước ngày 10 tháng 9 năm 2022

Đã nhận nhiệm vụ ĐTTN

Sinh viên

Đã giao nhiệm vụ ĐTTN

Giảng viên hướng dẫn

Hải Phòng, ngày tháng năm 2022

TRƯỞNG KHOA

TS. Đoàn Hữu Chức

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: ThS Đỗ Anh Dũng

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

Họ và tên sinh viên: Nguyễn Trọng Ngãi

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 2022

Giảng viên hướng dẫn

(ký và ghi rõ họ tên)

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 phản biện

Hải Phòng, ngày.....tháng.....năm 2022

Giảng viên chấm phản biện

(ký và ghi rõ họ tên)

MỤC LỤC

Nội dung	Số trang
Chương 1. Tổng quan về cảm biến và Arduino	1
1.1 Sóng siêu âm	1
1.2 Cảm biến siêu âm	2
1.3 Cảm biến siêu âm SRF10	6
Giới thiệu Arduino	10
1.4. Giới thiệu Arduino	11
1.4.1 Một số KIT tiêu biểu	12
a. Kit Arduino Uno R3	
b. Arduino Mega 2560	13
1.5. Cấu tạo LCD 1602A	20
Chương 2. Thiết kế hệ thống đo dùng cảm biến siêu âm	27
2.1. Thiết kế hệ thống thu phát siêu âm đo khoảng cách	
2.2 Thiết kế phần cứng giao tiếp cảm biến và Kit Arduino	27
2.3. Khởi hiển thị	28
2.4. Thiết kế khối cảnh báo	30
2.5 Khối nguồn	31
2.6. Lưu đồ thuật toán	32
Kết luận	34
Lời cảm ơn	35
Phụ lục	36
Tài liệu tham khảo	39

Chương 1. Tổng quan về cảm biến và Arduino

1.2 Sóng siêu âm.

a) **Định nghĩa:** Sóng siêu âm là những sóng cơ học có tần số lớn hơn 20Khz. Bởi vậy nó nằm ngoài khoảng âm thanh mà tai ta có thể nghe được. Một số loài vật như ong, dơi có thể cảm nhận được sóng siêu âm.

b) **Tính chất:**

Sóng siêu âm là sóng dọc: tức là giao động cùng chiều với chiều lan truyền sóng. Siêu âm chỉ truyền trong môi trường giãn nở (trừ chân không). Sóng âm tạo nên một sức ép làm thay đổi áp lực môi trường. Tại một vị trí nào đó trong môi trường, ở nửa chu kỳ đầu của sóng áp lực tại đó tăng, trong nửa chu kỳ sau lại giảm gây ra hiệu ứng cơ học của siêu âm. Sự chênh lệch áp suất giữa hai pha này là rất lớn, và tỷ lệ với tần số siêu âm. Nước và tổ chức cơ thể chịu sự biến thiên áp suất dễ bị phá huỷ ở pha giãn nở, gây nên hiệu ứng tạo lỗ.

Tốc độ lan truyền của siêu âm phụ thuộc vào bản chất và nhiệt độ môi trường truyền âm, không phụ thuộc vào tần số. Tốc độ truyền âm trong không khí là rất thấp khoảng 342m/s, trong cơ thể khoảng 1540m/s.

Năng lượng siêu âm: là động năng dao động và thế năng đàn hồi của các phần tử trong môi trường, được tính theo công thức sau:

$$E = \frac{1}{2} r. \omega^2 . a^2 = k. a^2 . f^2 \quad (1)$$

e: năng lượng siêu âm;

r: mật độ môi trường;

$\omega = 2\pi f$;

f: tần số siêu âm;

a: biên độ siêu âm.

Cường độ siêu âm: là năng lượng siêu âm truyền qua một đơn vị diện tích đặt vuông góc với phương truyền sóng:

$$I = \omega \cdot v = k \cdot a^2 \cdot f^2 \cdot v \quad (2)$$

I: cường độ siêu âm (W/cm²);

v: vận tốc truyền âm.

Như vậy cường độ siêu âm tỷ lệ thuận với bình phương của tần số và bình phương của biên độ sóng.

c) **Ứng dụng của sóng siêu âm trong cuộc sống**

Sóng siêu âm đang được ứng dụng trong nhiều ngành nghề khác nhau nhưng các nhà nghiên cứu cho rằng, giới hạn ứng dụng của chúng vẫn có thể tiếp tục mở rộng.

Sử dụng sóng siêu âm đuổi chuột, diệt côn trùng

Như đã đề cập ở trên, siêu âm là loại sóng có tần số cao hơn những gì tai người có thể nghe thấy, nhưng các loài xâm nhập có thể phát hiện ra chúng. Chính vì vậy, sóng siêu âm đã được sử dụng để xua đuổi chuột và các loài côn trùng gây hại như muỗi, gián,...

Máy đuổi chuột là những thiết bị nhỏ phát ra sóng âm tần số cao mà loài động vật này không hề thích thú. Khi thiết bị hoạt động, sự xuất hiện của siêu âm giống như chiếc khoan lớn với âm thanh chói tai, khó có thể dung hoà, khiến chúng khiếp sợ và phải bỏ đi.

Ứng dụng của sóng siêu âm trong y học

Ứng dụng sóng siêu âm trong y học đóng vai trò quan trọng, chúng được ứng dụng chủ yếu trong hoạt động siêu âm, thăm khám cho người bệnh được gọi là “siêu âm chẩn đoán”.

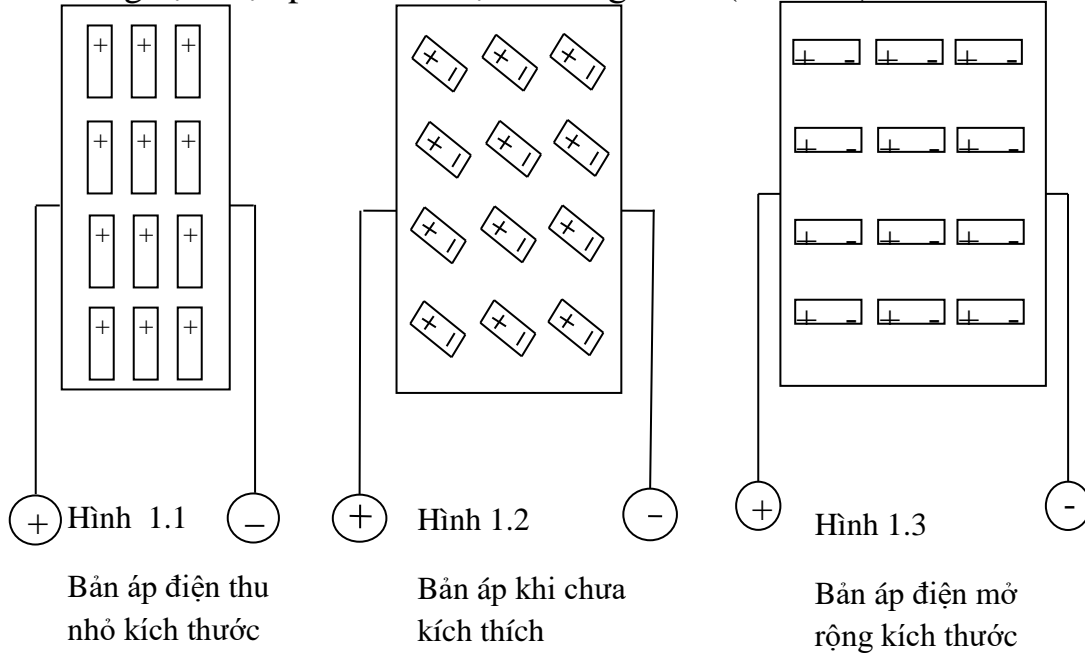
1.2 Cảm biến siêu âm.

a) **Khái niệm:**

Cảm biến siêu âm là thiết bị điện tử được dùng để xác định vị trí của các vật thông qua phát sóng siêu âm.

b) Cấu tạo:

Cảm biến thu phát siêu âm chủ yếu xây dựng trên tinh thể áp điện (Piezoelectric). Với các nguồn thu phát siêu âm công suất thường sử dụng kim loại giảo (magnetostrictive) hoặc gốm áp từ. Thạch cao là tinh thể áp điện thông dụng nhất. Trong quá trình gia công vật liệu, tinh thể áp điện được hình thành với các lưỡng cực điện phân tử có định hướng trước (hình 1.2).



Bộ cảm biến điện có thể mắc trong mạch kiểu cộng hưởng hoặc có thể kích thích phát siêu âm bằng tín hiệu sin ở tần số thích hợp. Việc kích thích bằng xung vuông góc cũng có tác dụng tương tự như sóng sin. Xung kích thích tác dụng vào cảm biến áp điện sẽ sắp xếp thẳng hàng các lưỡng cực điện phân tử dẫn đến làm dao động độ rộng của tinh thể. Các hình trên mô tả phản ứng của cảm biến với các xung kích thích : cảm biến bị thu hẹp (hình 2) khi các lưỡng cực phân tử sắp xếp theo chiều dọc, còn trên hình 3 cảm biến mở rộng khi các lưỡng cực phân tử sắp xếp theo chiều ngang. Dao động cơ học này được truyền ra môi trường xung quanh như sóng áp suất siêu âm. Khi tần số kích thích bằng tần số dao động riêng, hiệu suất phát sóng siêu âm là cực đại.

Ngược lại, khi có áp suất đặt vào giữa hai mặt của cảm biến và hình thành thế ra, cảm biến điện hoạt động như một bộ thu siêu âm (giống như loa nhỏ có thể làm loa hoặc micro).

c) Nguyên lý xác định khoảng cách của cảm biến siêu âm

Sóng siêu âm được truyền đi trong không khí với vận tốc khoảng 343m/s. Nếu 1 cảm biến phát ra các sóng siêu âm và thu về các phản xạ đồng thời, đo được khoảng thời gian từ lúc phát đi tới lúc thu về thì bộ xử lý có thể tính được quãng đường mà sóng đã di chuyển trong không gian. Quãng đường di chuyển của sóng sẽ bằng 2 lần khoảng cách từ cảm biến tới chướng ngại vật, theo hướng phát của sóng siêu âm. Hay khoảng cách sẽ được tính theo nguyên lý TOF(time of flight)

Nguyên lý TOF: là nguyên lý đo khoảng cách bằng thời gian chuyển của sóng phương pháp này được đặc biệt ứng dụng với các thiết bị sử dụng sóng siêu âm do vận tốc di chuyển của sóng trong không khí và trong các vật liệu khác tương đối chậm, và người ta có thể đo được khoảng cách với sai số tương đối nhỏ. Khoảng cách từ cảm biến tới chướng ngại vật được tính bằng vận tốc của sóng trong môi trường tương ứng nhân với 1 nửa thời gian truyền của sóng.

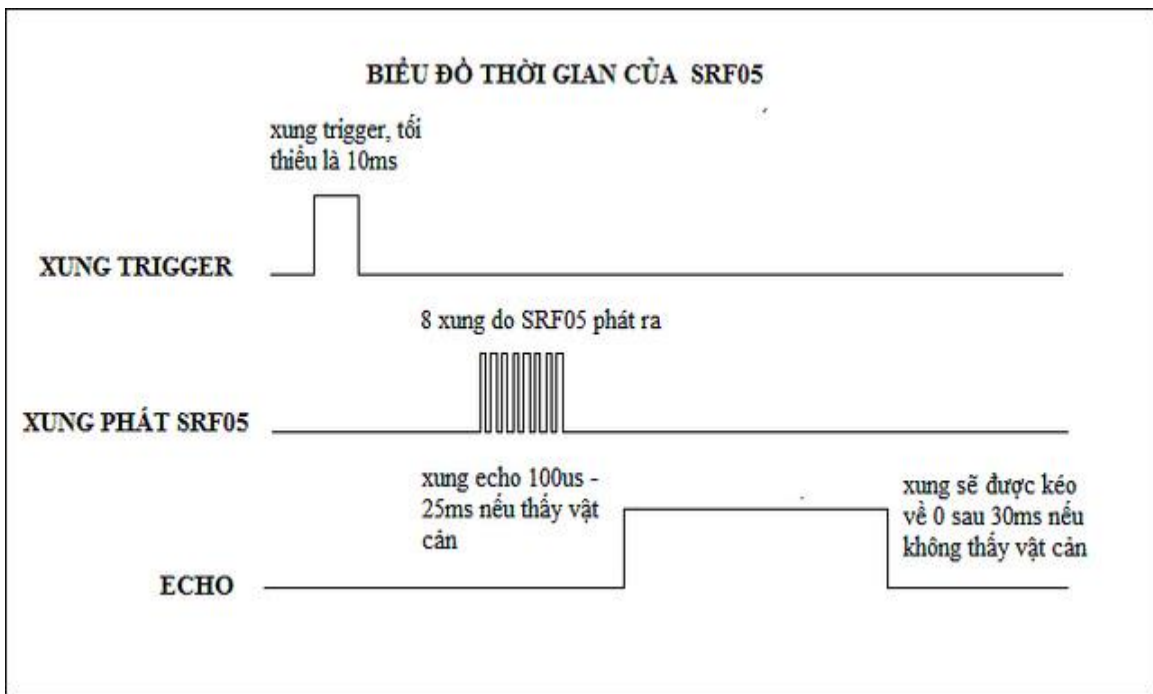
$$d = V \times \frac{t}{2}$$

d: khoảng cách từ cảm biến tới c. (3) ngại vật.

V: vận tốc của sóng; t: thời gian truyền của sóng.

Một ví dụ về hoạt động của cảm biến siêu âm đưa ra trên hình 1.4. 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. **Cảm biến siêu âm** càng xa thì càng bất không chính xác, vì góc quét của cảm biến sẽ mở rộng dần theo hình nón, ngoài ra bề mặt xiên hay xù xì cũng làm giảm độ chính xác của cảm biến, thông số kỹ thuật ghi ở dưới đây là của nhà sản xuất test trong điều kiện lý tưởng, còn thực tế thì tùy theo môi trường làm việc của cảm biến.



Hình 1.4. Cảm biến SRF05

d) Một số loại cảm biến siêu âm thường dùng.

SRF05	
Điện áp	5v
Dòng thấp	4mA
Tần số	40Khz
Phạm vi hoạt động	1cm-4m
Kích thước	43mm x 20mm x 17mm



Hình 1.5. Cảm biến SRF05

SRF10	
Điện áp	5v

Dòng	15mA
Tần số	40Khz
Phạm vi hoạt động	6cm-6m
Đầu nối	Bus tiêu chuẩn I2C
Kích thước	32mm x 15mm x 10mm

Hình 1.6. Cảm biến SRF10



SRF235	
Điện áp	5v
Dòng	25mA
Tần số	235Khz
Hình 1.7. Cảm biến SRF235	
Đầu nối	Bus tiêu chuẩn I2C
Kích thước	34mm x 20mm x 19mm



1.3 Cảm biến siêu âm SRF10

1.3.1 Giao tiếp

Giao tiếp với cảm biến này qua bus I2C. Điều này khả dụng trên những mạch điều khiển thông dụng như OOPic và stamp BS2p cũng như trong chế độ mở rộng của vi điều khiển. Người lập trình SRF10 xử lí giống như đối với series 24XX eeprom thường gặp, ngoại trừ địa chỉ I2C là khác nhau. Địa chỉ mặc định của SRF10 là 0xE0. Nó có thể được thay đổi bằng cách sử dụng 1 địa chỉ nào đó trong số 16 địa chỉ sau: E0, E2, E4, E6, E8, EA, EC, EE, F0, F2, F4, F6, F8, Fa, FC, FE do đó có tới 16 cảm biến có thể được sử dụng.

1.3.2 Ghép nối

Việc nối SRF10 giống như với SRF08. Điểm “do not connect” ở phía trái không nối thông. SDA và SCL mỗi đường có một điện trở 5V trên bus I2C. Chỉ cần 1 đôi điện trở, không cần 1 đôi cho mỗi module. Bình thường, chúng được xác định vị trí với bus chủ nhanh hơn bus tớ. SRF10 luôn là tớ, không bao giờ là chủ. Điện trở được khuyến nghị là 1,8k.

1.3.3 Các thanh ghi

SRF10 có 4 thanh ghi như bảng sau:

Location	Read	Write
0	Software Revision	Thanh ghi lệnh
1	Unused (reads 0x80)	Max Gain Register (default 16)
2	Byte cao	Range Register (default 255)
3	Byte thấp	N/A

Bảng 1: Các thanh ghi

Chỉ vùng 0, 1, 2 là có thể ghi. 0 là thanh ghi điều khiển và được dùng để bắt đầu một phiên làm việc (đo khoảng cách). Việc đọc từ vùng 0 quay trở lại kiểm tra phần mềm SRF10. Mặc định khoảng cách đo kéo dài 65ms, nhưng có thể thay đổi bằng cách ghi vào vùng 2. SRF10 sẽ không trả lời trên bus I2C trong khi nó đo khoảng cách.

Thanh ghi 2 và 3 là 16 bit không dấu là kết quả của phép đo khoảng cách gần nhất – byte cao đầu tiên. Có nghĩa là giá trị của nó phụ thuộc vào lệnh sử dụng và nó có thể là inches hay cm hay us. Giá trị bằng 0 chứng tỏ rằng không có vật nào được phát hiện. Có 3 lệnh để bắt đầu một phiên đo (80 đến 82), để trả về kết quả là inches hay cm hay us. Cũng có thể điều chỉnh câu lệnh để thay đổi địa chỉ I2C.

1.3.4 Chế độ đo khoảng cách

Để bắt đầu đo khoảng cách, ghi lệnh vào một trong các thanh ghi kể trên, và đợi giá trị về thời gian hoàn thành được yêu cầu và đọc kết quả. Bộ đệm phản hồi được xóa khi bắt đầu mỗi phiên đo. Thời gian mặc định và được khuyến nghị để hoàn thành 1 phép đo là 65ms, tuy nhiên ta cũng có thể rút ngắn thời gian bằng cách ghi vào ô nhớ trước khi đưa ra lệnh đo.

1.3.5 Kiểm tra sự hoàn tất phép đo.

Ta không thể dùng việc đếm thời gian trên vi điều khiển để kiểm tra và đợi phép đo kết thúc. Một ưu điểm thiết thực là SRF10 sẽ không trả lời hoạt động của I2C khi đang đo. Do vậy nên cố gắng đọc từ SRF10 (sử dụng phần mềm kiểm tra số vùng 0) sau đó đặt 255(0xFF) trong khi đo. Đó là vì đường dữ liệu I2C(SDA) được pull cao nếu không có gì điều khiển nó. Ngay khi việc đo hoàn tất, SRF10 sẽ trả lời tới bus I2C, duy trì việc đọc thanh ghi đến khi không là 255 nữa. Sau đó có thể đọc dữ liệu trả về. Người lập trình có thể dùng một vài lợi ích của điều đó để thực hiện công việc khác trong khi SRF10 đang đo.

1.3.6 Thay đổi khoảng cách

Khoảng cách tối đa của SRF được lập bởi bộ định thời bên trong. Mặc định nó bằng 65ms, hay tương đương với 11m chiều dài. Khoảng cách này xa hơn so với khả năng thực tế của SRF10. Có thể giảm thời gian SRF10 nhận tín hiệu phản hồi lại, and hence the range, bằng cách ghi vào thanh ghi lệnh ở vùng 2. Khoảng đo có thể được chia thành nhiều bước nhỏ khoảng 43mm cho tới khi đủ 11m. Khoảng đo = (range register x 43mm)+43 mm do vậy đặt = 0(0x00) nhận được khoảng đo max là 43mm, = 1(0x01) nhận được khoảng đo max là 86mm. thông thường, đặt = 24(0x18) – 1m và =93(0x5D) –4m. Khi đặt là 255 được khoảng đo 11m. Có 2 lí do mà bạn sẽ muốn làm giảm khoảng đo:

1. Nhận được thông tin về khoảng đo nhanh hơn.
2. Có thể khởi động SRF10 nhanh hơn.

Nếu chỉ muốn nhận thông tin khoảng đo sớm hơn, sau đó có thể tiếp tục khởi động SRF10 ở 65mm thì tất cả đều ổn. tuy nhiên nếu bạn muốn khởi động SRF10 ở tốc độ cao hơn 65mm thì sẽ cần giảm hệ số khuếch đại. Khoảng đo có thể được đặt cao nhất ở mỗi lần SRF10 được cấp nguồn.

1.3.7 Analogue gain

Thanh ghi Analogue Gain thiết lập giá trị cực đại. Để đặt giá trị cực đại, bạn hãy ghi vào thanh ghi số 1. Trong quá trình đo, analogue gain sẽ bắt đầu từ giá trị min của nó là 40, bước nhảy là 96 us đến khi nào nó đạt được giá trị cực đại ghi tại thanh ghi số 1. Thông thường Gain có thể đạt giá trị cực đại sau khoảng 100mm trong quá trình đo. Mục đích của việc giới hạn giá trị cực đại Gain là để cho phép thời gian phát siêu âm nhanh hơn 65ms. Khi thời gian đo tìm ngắn, một quá trình

đo tìm mới có thể được khởi tạo ngay khi các dữ liệu về quá trình đo trước đó đã được đọc xong. Có thể có rủi ro khi thực hiện đo lần 2 là do lần 2 có thể nhận những phản hồi (dữ liệu) từ lần phát trước đó, điều này đưa đến kết quả sai về một vật thể rất gần nhưng không có thực. Để giảm thiểu khả năng này, Giá trị cực đại của gain được giảm tới giới hạn độ nhạy module để phân biệt được những vật ở gần và không nhận ra những phản hồi yếu hơn do khoảng cách. Việc cài đặt gain max được lưu trữ trong RAM và được khởi tạo lại khi bật điện, vì vậy nếu bạn chỉ muốn đo ở khoảng cách 65ms hoặc dài hơn, bạn có thể bỏ qua các thanh ghi range và thanh ghi gain. Thanh ghi Gain được thiết lập tới 16(khoảng 700) khi bật nguồn. Nó có thể giảm khi có yêu cầu.

Cần chú ý rằng mối quan hệ giữa giá trị thanh ghi Gain được thiết lập và gain thực tế không tuyến tính. Do vậy không có cách thức nào chỉ ra việc ‘dùng giá trị gain thiết lập này với khoảng đo kia’. Nó phụ thuộc vào kích thước, hình khối, chất liệu của các vật và những cái khác xung quanh phòng. Hãy cố

Gain Register		Maximum Analogue Gain
Decimal	Hex	
0	0x00	Set Maximum Analogue Gain to 40
1	0x00	As above - Analogue Gain to 40
2	0x01	Set Maximum Analogue Gain to 50
3	0x02	Set Maximum Analogue Gain to 60
4	0x03	Set Maximum Analogue Gain to 70
5	0x04	Set Maximum Analogue Gain to 80
6	0x05	Set Maximum Analogue Gain to 100
7	0x06	Set Maximum Analogue Gain to 120
8	0x07	Set Maximum Analogue Gain to 140
9	0x08	Set Maximum Analogue Gain to 200
10	0x09	Set Maximum Analogue Gain to 250
11	0x0A	Set Maximum Analogue Gain to 300
12	0x0B	Set Maximum Analogue Gain to 350
13	0x0C	Set Maximum Analogue Gain to 400
14	0x0D	Set Maximum Analogue Gain to 500
15	0x0E	Set Maximum Analogue Gain to 600
16	0x0F	Set Maximum Analogue Gain to 700

Bảng 2. Thanh ghi hệ số khuếch đại

gắng làm nhiều lần với các thiết lập khác nhau cho tới khi có được kết quả bạn mong muốn. Nếu bạn thấy các chỉ số có vẻ sai thì nó có thể là các phản hồi từ trước, hãy cố gắng khởi tạo lại SFR10 ở 65ms hay dài hơn. Nếu bạn nghi ngờ về range và thanh ghi gain, hãy nhớ chúng được thiết lập một cách tự động tới giá trị mặc định của nó bởi SRF10 khi bật nguồn. Bạn có thể bỏ qua hoặc quên nó đi và SFR10 sẽ hoạt động tốt, phát hiện những vật xa 6m ở 65ms hay chậm hơn.

1.3.8 Led

Led đỏ được dùng để thể hiện 1 mã trong địa chỉ bus I2C. Ngoài ra nó cũng sáng trong khi thực hiện đo.

1.3.9 Thay đổi địa chỉ bus I2C

Để thay đổi địa chỉ bus I2C của SRF10, bạn chỉ cần có 1 cảm biến trên bus. Hãy ghi 1 dãy 3 lệnh theo thứ tự đúng như trong địa chỉ. Ví dụ, để thay đổi địa chỉ của 1 cảm biến từ 0xE0(địa chỉ mặc định) thành 0xF2, viết : 0xA0, 0xAA,

Address		Long Flash	Short flashes
Decimal	Hex		
224	E0	1	0
226	E2	1	1
228	E4	1	2
230	E6	1	3
232	E8	1	4
234	EA	1	5
236	EC	1	6
238	EE	1	7
240	F0	1	8
242	F2	1	9
244	F4	1	10
246	F6	1	11
248	F8	1	12
250	FA	1	13
252	FC	1	14
254	FE	1	15

Bảng 3. Địa chỉ của bus I2C SRF10.

0xA5, 0xF2. Các lệnh này phải được gửi đúng thứ tự để thay đổi địa chỉ I2C, thêm vào đó, không một lệnh nào khác có thể chen vào giữa dãy. Dãy này phải được gửi tới thanh ghi lệnh ở vùng 0, which means 4 separate write transactions on the I2C bus. Khi gửi được, bạn có thể đánh dấu (dán nhãn) cảm biến với địa chỉ của nó, tuy nhiên nếu bạn quên, hãy chỉ cấp nguồn cho nó mà không gửi 1 lệnh nào. SRF10 sẽ thể hiện địa chỉ của nó trên Led. 1 lần led nhấp sáng lâu đi kèm với 1 số lần nhấp định led nhấp sáng trong thời gian ngắn chỉ thị địa chỉ của nó. Việc nhấp sáng sẽ kết thúc ngay lập tức khi gửi một lệnh tới SFR10.

Chú ý không thiết lập cho nhiều hơn 1 cảm biến cùng một địa chỉ giống nhau, sẽ có sự xung đột và khó dự đoán được kết quả.

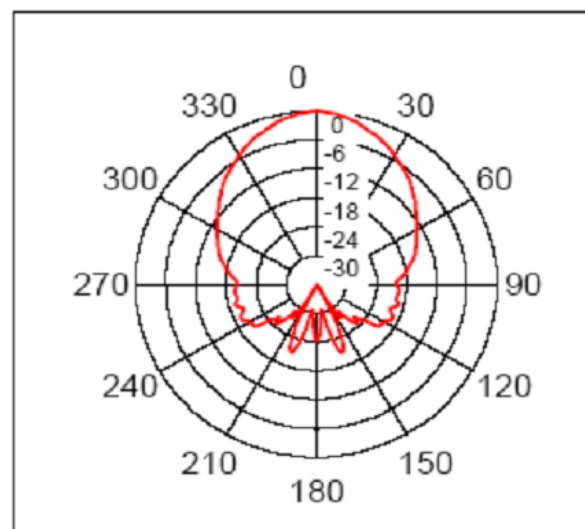
1.3.10 Thay đổi hình dạng và độ rộng búp sóng.

Không hề dễ dàng để giảm hay thay đổi hình dạng và độ rộng búp sóng mà bạn biết. Búp sóng của SRF10 có dạng nón với độ rộng là 1 hàm của diện tích bề mặt bộ cảm biến và nó được cố định. Với hình dạng của búp sóng như nhà sản xuất cho biết, ta thấy khi đo những vật ở xa sẽ sảy ra nhiều. Khi xuất hiện những vật ở lân cận sẽ sảy ra hiện tượng phản xạ ngược trở lại cảm biến và cho kết quả không như mong muốn. Có thể làm cho cảm biến bớt nhạy đi với những vật xung quanh bằng cách giảm tối đa thanh ghi gain từ 16 tới mức thấp hơn. Đó là 1 giải pháp cho khoảng đo ngắn hơn, tuy nhiên, hầu hết những robot nhỏ không cần khoảng đo tới 6m. Giá trị 8(max gain 140) thực tế có thể giảm khoảng đo tới khoảng 2 m nhưng nó sẽ làm giảm độ nhạy đối với những vật ở chính diện. Hình dạng chùm tia của bộ cảm biến được sử dụng trên SRF10 được lấy từ các nhà sản xuất được thể hiện ở hình bên 1.8.

1.4 Arduino

1.4.1. Giới thiệu Arduino

Theo định nghĩa từ www.arduino.cc , **Arduino** là nền tảng điện tử mã nguồn mở, dựa



Hình 1.8. Dạng sóng của SRF10.

Trên board còn có 1 nút reset, 1 cổng kết nối với máy tính qua cổng USB và 1 đường cấp nguồn sử dụng jack 2.1mm lấy năng lượng trực tiếp từ AC-DC adapter hay thông qua ắc-quy nguồn.



Hình 1.10. Mặt trước, mặt sau và bố trí chân thực tế của KIT Arduino Uno R3.

Bảng 4. Các thông số kỹ thuật của KIT Arduino Uno R3.

Vi điều khiển	ATmega328P
Điện áp hoạt động	5V
Điện áp vào khuyến dùng	7-12V
Điện áp vào giới hạn	6-20V
Digital I/O pin	14 (trong đó 6 pin có khả năng băm xung)
PWM Digital I/O Pins	6
Analog Input Pins	6
Cường độ dòng điện trên mỗi I/O pin	20 mA
Cường độ dòng điện trên mỗi 3.3V pin	50 mA
Flash Memory	32 KB (ATmega328P) 0.5 KB được sử dụng bởi bootloader
SRAM	2 KB (ATmega328P)
EEPROM	1 KB (ATmega328P)
Tốc độ	16 MHz
Chiều dài	68.6mm
Chiều rộng	53.4mm
Trọng lượng	25g

d. Arduino Mega 2560

Arduino Mega2560 là một hệ thống sử dụng vi điều khiển ATmega2560.

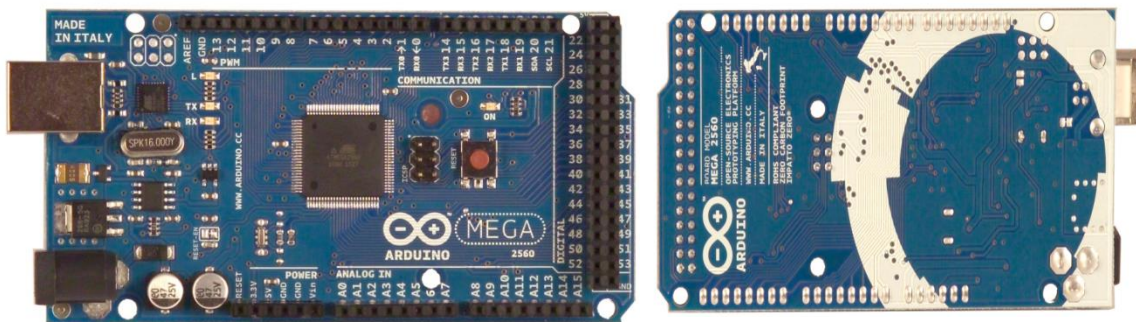
Hệ thống bao gồm:

- 54 chân digital (15 có thể được sử dụng như các chân PWM)
- 16 đầu vào analog,
- 4 UARTs (cổng nối tiếp phân cứng),
- 1 thạch anh 16 MHz,
- 1 cổng kết nối USB,
- 1 jack cắm điện,
- 1 đầu ICSP,
- 1 nút reset.

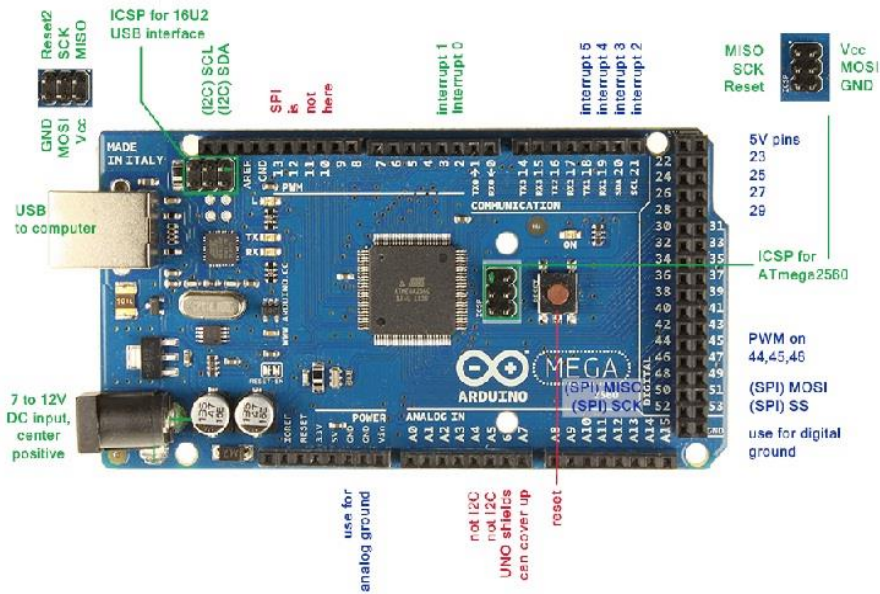
Nó chứa tất cả mọi thứ cần thiết để hỗ trợ các vi điều khiển.

Arduino Mega2560 khác với tất cả các vi xử lý trước giờ vì không sử dụng FTDI chip điều khiển chuyển tín hiệu từ USB để xử lý. Thay vào đó, nó sử dụng ATmega16U2 lập trình như là một công cụ chuyên đổi tín hiệu từ USB. Như vậy các kit này sẽ được lập trình và nạp trực tiếp qua các cổng USB. Ngoài ra, Arduino Mega2560 cơ bản vẫn giống Arduino Uno R3, chỉ khác số lượng chân và có nhiều tính năng mạnh mẽ hơn, nên vẫn có thể lập trình cho vi điều khiển này bằng chương trình lập trình cho Arduino Uno R3.

Sơ đồ chi tiết chân vào/ra của hệ thống Arduino Mega 2560 như hình 1.11 và 1.12 dưới đây.



Hình 1.11. Mặt trước và sau của Arduino Mega 2560 thực tế.



Hình 1.12. Bố trí chân vào/ra Arduino Mega 2560 thực tế.

Bảng 5. Thông số kỹ thuật của Arduino Mega 2560

Vi điều khiển	ATmega2560
Điện áp hoạt động	5V
Điện áp vào (đề nghị)	7V-15V
Điện áp vào (giới hạn)	6V-20V
Cường độ dòng điện trên mỗi 3.3V pin	50 mA
Cường độ dòng điện trên mỗi I/O pin	20 mA
Flash Memory	256 KB
SRAM	8 KB
EEPROM	4 KB
Clock Speed	16 MHz

Sơ đồ chi tiết các chân chức năng của Kit Arduino Mega 2560 trên hình 1.13.

Chân điều khiển:

RESET: Arduino Mega Mega 2560 có sẵn mạch reset với nút ấn để thiết lập lại hệ thống và chân này có thể được sử dụng khi kết nối các thiết bị khác để thiết lập lại bộ điều khiển.

XTAL1, XTAL2: Thạch anh(16Mhz) được kết nối với xung clock cung cấp cho bộ điều khiển.

AREF: Chân này được dùng khi sử dụng ADC để chuyển đổi tín hiệu với điện áp tham chiếu bên ngoài mà không muốn sử dụng điện áp tham chiếu nội bộ 1.1V hoặc 5V.

Các chân Digital (70):

Chân số: Từ 0-53 (số) và 0-15 (tương tự) có thể được sử dụng làm đầu vào hoặc đầu ra cho thiết bị được thiết lập bằng các hàm Mode (), digitalWrite (), digitalRead ().

Ứng dụng:

Thiết bị đầu ra: Relay, LED, buzzer, LCD và các thiết bị khác.

Thiết bị đầu vào: Nút ấn, cảm biến siêu âm, cần điều khiển và các thiết bị khác

Thí dụ:

Đầu ra tín hiệu thấp trên Arduino mega

```
pinMode (0, OUTPUT);
```

```
digitalWrite (0, LOW);
```

Đầu vào đọc tín hiệu trên Arduino mega

```
pinMode (0, INPUT);
```

```
digitalRead (0);
```

Chân tương tự (16):

Từ 0-15 (analog) có thể được sử dụng như chân đầu vào tương tự cho bộ ADC, nếu không sử dụng nó hoạt động như chân digital bình thường. Nó được thiết lập bởi các hàm pinMode () khai báo chân, analogRead () để đọc trạng thái chân và nhận giá trị kỹ thuật số cho tín hiệu analog. Lưu ý phải cẩn thận để lựa chọn điện áp tham chiếu bên trong hoặc bên ngoài và chân Aref.

Ứng dụng :

Thiết bị đầu vào: Cảm biến nhiệt độ, cảm biến (như ldr, irtled và độ ẩm) và các thiết bị khác

Thí dụ :

```
pinMode (0, INPUT);
```

```
analogRead (0);
```

Chân có Chức năng thay thế:

Chân SPI: Chân 22-SS, 23_SCK, 24-MOSI, 25-MISO

Các chân này được sử dụng cho giao tiếp nối tiếp với giao thức SPI để liên lạc giữa 2 thiết bị trở lên. SPI cho phép bit phải được thiết lập để bắt đầu giao tiếp với các thiết bị khác.

Ứng dụng: Lập trình điều khiển AVR, giao tiếp với những người khác ngoài vi như LCD và thẻ SD.

Chân I2C:

Chân 20 cho SDA và 21 cho SCK (Tốc độ 400khz) để cho phép liên lạc hai dây với các thiết bị khác. Hàm được sử dụng là `wire.begin ()` để bắt đầu chuyển đổi I2C, với `wire.Read ()` để đọc dữ liệu i2c và `wire.Write ()` để ghi dữ liệu i2c.

Ứng dụng:

Thiết bị đầu ra: LCD và liên lạc giữa nhiều thiết bị với hai dây.

Thiết bị đầu vào: RTC và các thiết bị khác.

Thí dụ:

I2C chỉ đọc dữ liệu

```
Wire.begin ();
```

```
Wire.requestFrom (2, 1); // dữ liệu 1byte
```

```
Wire.Read ();
```

PWM chân :

Chân 2-13 có thể được sử dụng như đầu ra PWM với hàm `analogWrite ()` để ghi giá trị pwm từ 0-255.

Ứng dụng:

Thiết bị đầu ra: Điều khiển tốc độ của động cơ, ánh sáng mờ, pid cho hệ thống điều khiển hiệu quả.

Thí dụ:

Ngõ ra tín hiệu tương tự trên bảng mạch Arduino

pinMode (0, OUTPUT);

analogWrite (0,255);

Chân USART:

Chân 0 - RXD0, chân 1 - TXD0

Chân 19 - RXD1, chân 18 - TXD1

Chân 17 - RXD2, chân 16 - TXD2

Chân 15 - RXD3, chân 14 - TXD3

Chân này được sử dụng cho giao tiếp nối tiếp giữa bo mạch với máy tính hoặc hệ thống khác để chia sẻ và ghi dữ liệu. Nó được sử dụng với hàm serialBegin () để cài đặt tốc độ truyền và bắt đầu truyền thông với serial.Println () để in mảng ký tự (mảng char) ra thiết bị khác.

Ứng dụng:

Bộ điều khiển truyền thông và máy tính

Ví dụ :

Serial.begin (9600);

Serial.Println (“hello”);

Chân ngắt :

Chân digital: 0,22,23,24,25,10,11,12,13,15,14

Chân analog: 6,7,8,9,10,11,12,13,14,15

Chân này được sử dụng để ngắt. Để kích hoạt chân ngắt phải cài đặt bật ngắt toàn cục.

Ứng dụng :

Bộ mã hóa vòng quay, nút bấm dựa trên ngắt và các nút khác.

Thí dụ :

```
pinMode (0, OUTPUT);
```

```
pinMode (1, INPUT_PULLUP);
```

```
attachInterrupt (digitalpinToInterrupt (1), LOW, CHANGE);
```

Chân ngắt phần cứng:

Chân 18 - 21, 2, 3 ngắt phần cứng được sử dụng cho các ứng dụng ngắt. Ngắt phần cứng phải được bật với tính năng ngắt toàn cục để ngắt quãng từ các thiết bị khác.

Ứng dụng:

Nhấn nút cho chương trình ISR, đánh thức bộ điều khiển bằng thiết bị bên ngoài như cảm biến siêu âm và các thiết bị khác.

Thí dụ:

```
pinMode (0, OUTPUT);
```

```
pinMode (1, INPUT_PULLUP);
```

```
attachInterrupt (digitalpinToInterrupt (1), LOW, LOW);
```

Các phần của Arduino Mega:

Giắc cắm nguồn DC:

Cấp nguồn cho Arduino Mega từ 7-12V qua cổng này. Arduino Mega R3 có bộ điều chỉnh điện áp nguồn cấp 5V và 3.3V cho bộ điều khiển Arduino và bộ cảm biến.

AVR 2560:

Đây là vi điều khiển chính được sử dụng để lập trình và chạy tác vụ cho hệ thống. Đây là bộ não của hệ thống để điều khiển tất cả các thiết bị khác trên mạch.

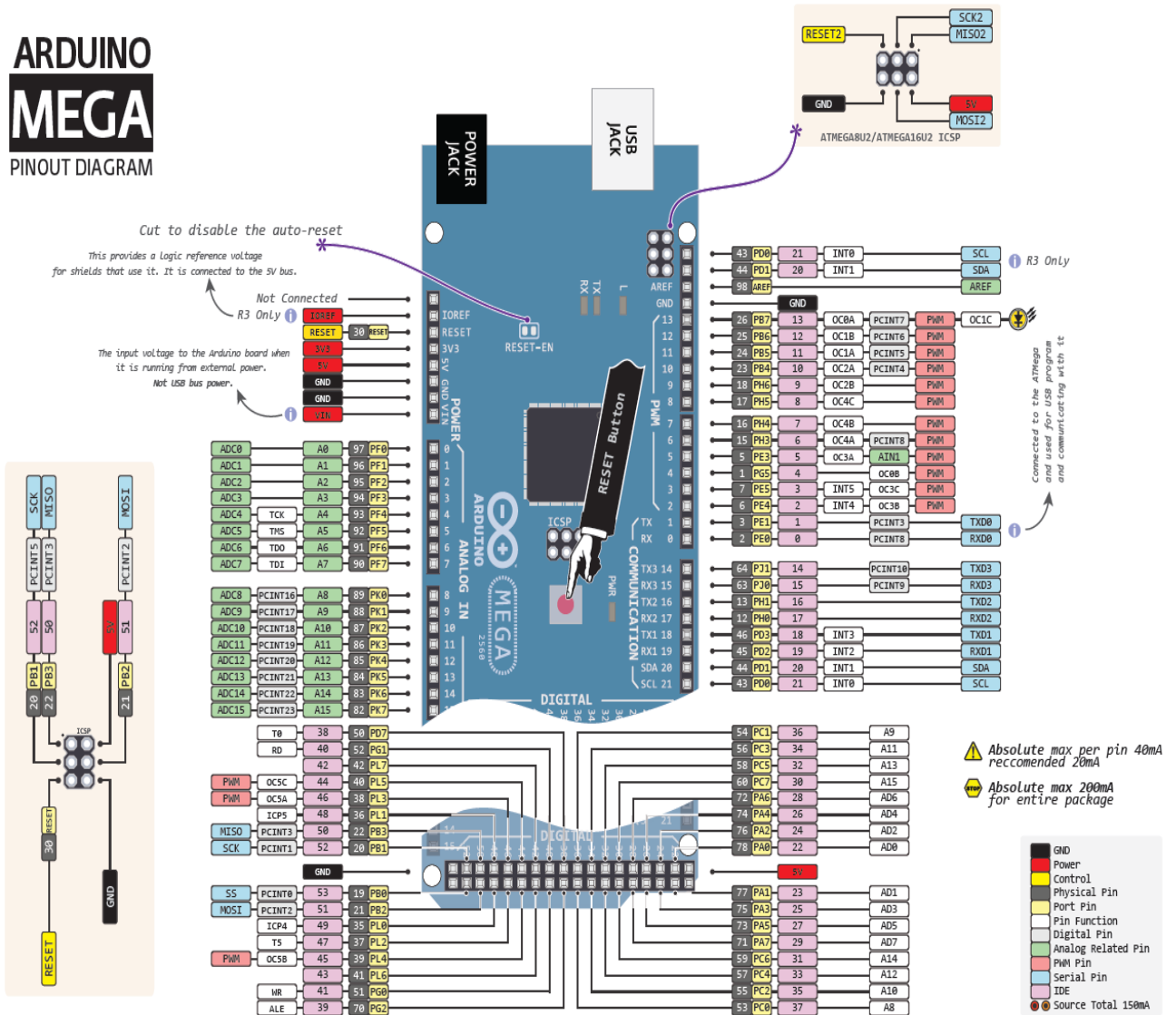
ATmega8:

Vi điều khiển này được sử dụng để liên lạc giữa bộ điều khiển chính và các thiết bị khác. Bộ điều khiển này được lập trình cho giao tiếp USB và các tính năng lập trình nối tiếp.

ICSP 1 (ATmega8) và 2 (AVR 2560):

Nó có các tính năng của lập trình sử dụng bus nối tiếp với lập trình AVR sử dụng giao tiếp SPI. AVR 2560 được lập trình để chạy hệ thống và ATmega 8 được lập trình để liên lạc và lập trình nối tiếp.

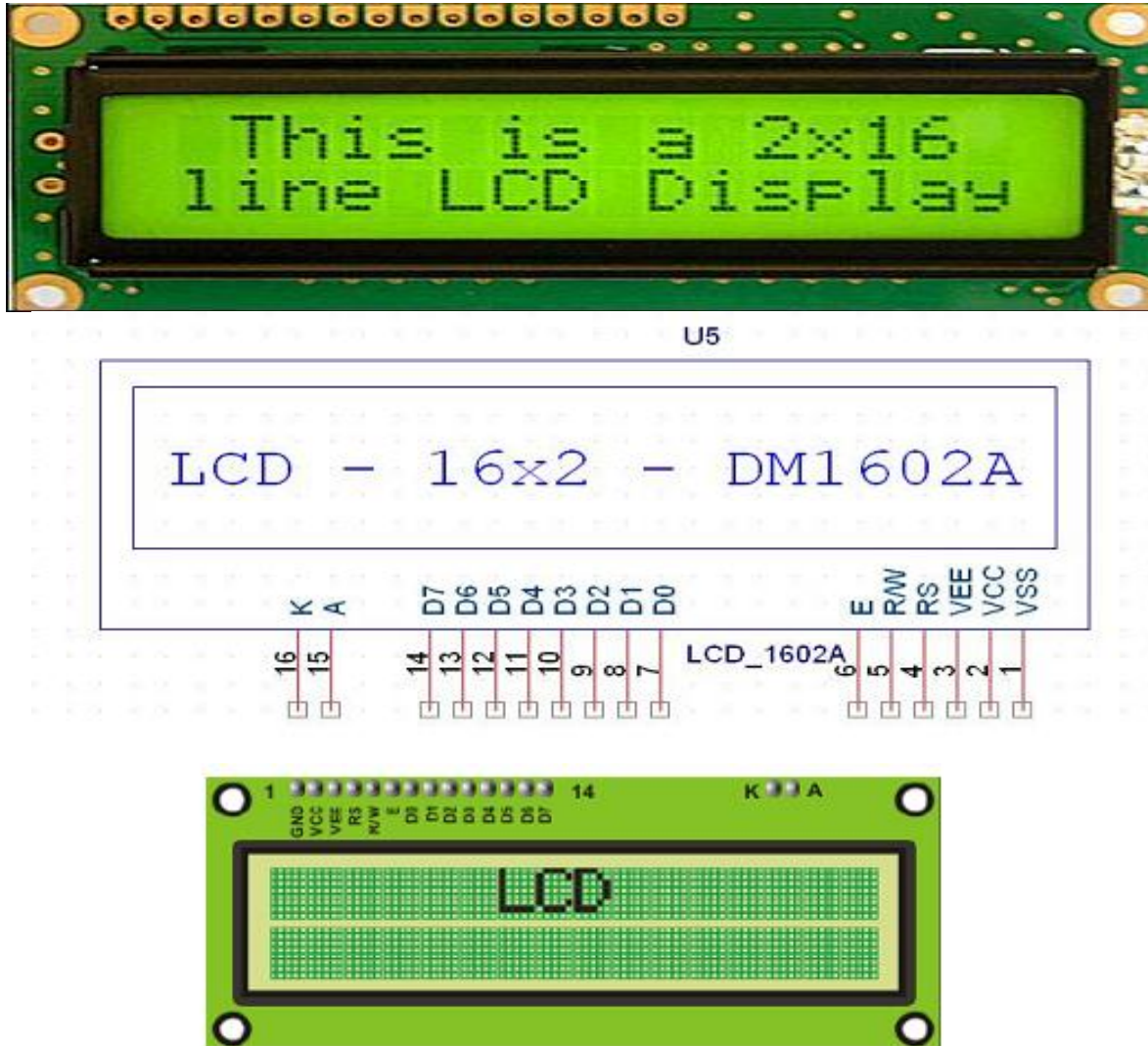
ARDUINO MEGA PINOUT DIAGRAM



Hình 1.13. Sơ đồ chức năng các chân vào/ra Arduino Mega 2560.

1.5. Cấu tạo LCD 1602A

Màn hình LCD được sử dụng để hiển thị các kí tự mong muốn. Trong đồ án này sử dụng loại LCD có thể hiển thị được trên hai hàng và 16 kí tự trên mỗi hàng.



Hình 1.14. Màn hình LCD.

Các thanh ghi

Chân	Kí hiệu	Mức Logic	I/O	Chức năng
1	Vss	-	-	Nguồn (GND)
2	Vcc	-	-	Nguồn (+5V)
3	Vee	-	-	Chỉnh độ tương phản
4	RS	0/1	I	0 = Nhập lệnh 1 = Nhập dữ liệu
5	R/W	0/1	I	0 = Ghi dữ liệu 1 = Đọc dữ liệu
6	E	1, 1->0	I	Tín hiệu cho phép
7	DB0	0/1	I/O	Bus dữ liệu 0
8	DB1	0/1	I/O	Bus dữ liệu 1
9	DB2	0/1	I/O	Bus dữ liệu 2
10	DB3	0/1	I/O	Bus dữ liệu 3
11	DB4	0/1	I/O	Bus dữ liệu 4
12	DB5	0/1	I/O	Bus dữ liệu 5
13	DB6	0/1	I/O	Bus dữ liệu 6
14	DB7	0/1	I/O	Bus dữ liệu 7
15	Lamp-	-	-	Đèn LCD
16	Lamp+	-	-	Đèn LCD

- Thanh ghi IR: Mỗi lệnh được nhà sản xuất LCD đánh địa chỉ rõ ràng. Người dùng chỉ việc cung cấp địa chỉ lệnh bằng cách nạp vào thanh ghi IR.

Ví dụ:

Lệnh “hiển thị màn hình và con trỏ” có mã lệnh là 00001110

- Thanh ghi DR : Thanh ghi DR dùng để chứa dữ liệu 8 bit để ghi vào vùng RAM DDRAM hoặc CGRAM (ở chế độ ghi) hoặc dùng để chứa dữ liệu từ 2 vùng RAM này gửi ra cho MPU (ở chế độ đọc).

- **Cờ báo bận BF: (Busy Flag)**

Khi đang thực thi các hoạt động bên trong, LCD bỏ qua mọi giao tiếp với bên ngoài và bật cờ BF(thông qua chân DB7 khi có thiết lập RS=0, R/W=1) lên để cho biết nó đang “bận”.

- **Bộ đếm địa chỉ AC : (Address Counter)**

Khi một địa chỉ lệnh được nạp vào thanh ghi IR, thông tin được nối trực tiếp cho 2 vùng RAM (việc chọn lựa vùng RAM tương tác đã được bao hàm trong mã lệnh). Sau khi ghi vào (đọc từ) RAM, bộ đếm AC tự động tăng lên (giảm đi) 1 đơn vị.

Bộ nhớ LCD

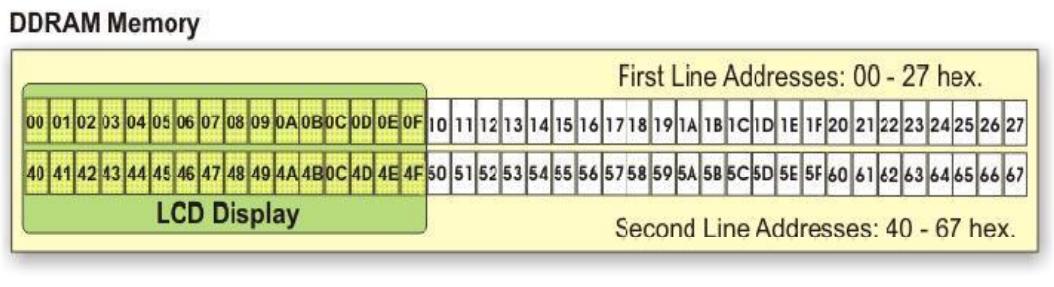
Vùng RAM hiển thị DDRAM : (Display Data RAM)

Vùng CGROM: Character Generator ROM

Vùng CGRAM : (Character Generator RAM)

DDRAM

Đây là vùng RAM dùng để hiển thị, nghĩa là ứng với một địa chỉ của RAM là một ô kí tự trên màn hình.



CGROM

Chứa các mẫu kí tự loại 5x7 hoặc 5x10 điểm ảnh/kí tự, và định địa chỉ bằng 8 bit

		4 higher bits of address																	
		0000	0001	0010	0011	0100	0101	0110	0111	1000	1001	1010	1011	1100	1101	1110	1111		
4 lower bits of address	xxxx0000	CG RAM (1)			0	1	P	~	P					-	9	3	α	p	
	xxxx0001	(2)			!	1	A	Q	a	9					.	7	4	ä	q
	xxxx0010	(3)			"	2	B	R	b	r					「	イ	ツ	β	θ
	xxxx0011	(4)			#	3	C	S	c	s					」	ウ	テ	ε	ω
	xxxx0100	(5)			\$	4	D	T	d	t					、	エ	ト	μ	Ω
	xxxx0101	(6)			%	5	E	U	e	u					.	オ	ナ	υ	Û
	xxxx0110	(7)			&	6	F	V	f	v					ヲ	カ	ニ	ρ	Σ
	xxxx0111	(8)			'	7	G	W	g	w					ア	キ	ヌ	q	π
xxxx1000	(1)			<	8	H	X	h	x					イ	ク	ネ	7	又	
xxxx1001	(2)			>	9	I	Y	i	y					ッ	ケ	ル	ˆ	4	
xxxx1010	(3)			*	:	J	Z	j	z					エ	コ	ハ	j	キ	
xxxx1011	(4)			+	;	K	C	k	<					オ	サ	ヒ	*	5	
xxxx1100	(5)			,	<	L	¥	l	l					ハ	シ	フ	φ	円	
xxxx1101	(6)			-	=	M	J	m	>					ユ	ズ	ヘ	シ	モ	÷
xxxx1110	(7)			.	>	N	^	n	→					ヨ	セ	ホ	ˆ	ñ	
xxxx1111	(8)			/	?	O	_	o	←					ッ	ツ	マ	°	ö	■

										LSB							
A11	A10	A9	A8	A7	A6	A5	A4	A3	A2	A1	A0	O4	O3	O2	O1	O0	
									0	0	0	0	1	0	0	0	0
									0	0	0	1	1	0	0	0	0
									0	0	1	0	1	0	1	1	0
									0	0	1	1	1	1	0	0	1
									0	1	0	0	1	0	0	0	1
									0	1	0	1	1	0	0	0	1
									0	1	1	0	1	1	1	1	0
									0	1	1	1	0	0	0	0	0

Mẫu kí tự đồ họa riêng.

Table 5 Relationship between CGRAM Addresses, Character Codes (DDRAM) and Character Patterns (CGRAM Data)

For 5 × 8 dot character patterns

Character Codes (DDRAM data)				CGRAM Address				Character Patterns (CGRAM data)															
7	6	5	4	3	2	1	0	5	4	3	2	1	0	7	6	5	4	3	2	1	0		
High				Low				High				Low				High				Low			
0 0 0 0 * 0 0 0				0 0 0				0 0 0				* * *				1 1 1 1 0				Character pattern (1)			
																1 0 0 0 1							
																1 0 0 0 1							
																1 1 1 1 0							
																1 0 0 0 1							
0 0 0 0 * 0 0 1				0 0 1				0 0 1				* * *				1 0 0 0 1				Character pattern (2)			
																0 1 0 0							
																0 1 1 1							
																1 0 0 1							
																1 1 0 0							
0 0 0 0 * 1 1 1				1 1 1				1 0 0				* * *				1 0 0 1				Cursor position			
																1 1 0							
																1 1 0							
																1 1 1							
																1 1 1							

- Notes:
- Character code bits 0 to 2 correspond to CGRAM address bits 3 to 5 (3 bits: 8 types).
 - CGRAM address bits 0 to 2 designate the character pattern line position. The 8th line is the cursor position and its display is formed by a logical OR with the cursor. Maintain the 8th line data, corresponding to the cursor display position, at 0 as the cursor display. If the 8th line data is 1, 1 bits will light up the 8th line regardless of the cursor presence.
 - Character pattern row positions correspond to CGRAM data bits 0 to 4 (bit 4 being at the left).
 - As shown Table 5, CGRAM character patterns are selected when character code bits 4 to 7 are all 0. However, since character code bit 3 has no effect, the R display example above can be selected by either character code 00H or 08H.
 - 1 for CGRAM data corresponds to display selection and 0 to non-selection.
- * Indicates no effect.

Tập lệnh:

Tập lệnh	Mã nhị phân										Mô tả	Thời gian thực thi	
	RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0			
Xoá hiển thị	0	0	0	0	0	0	0	0	0	0	1	Xoá hiển thị và đưa con trỏ về vị trí ban đầu (địa chỉ 0).	1.64mS
Con trỏ ban đầu	0	0	0	0	0	0	0	0	0	1	*	Trả con trỏ về vị trí ban đầu (địa chỉ 0). Ngoài ra đưa hiển thị đã bị dịch chuyển về vị trí ban đầu. Nội dung bộ nhớ hiển thị dữ liệu(DDRAM) không thay đổi.	1.64mS
Thiết lập chế độ	0	0	0	0	0	0	0	0	1	I/D	S	Thiết lập hướng di chuyển của con trỏ tăng/giảm(I/D=0:giảm,I/D=1:tăng), chỉ rõ dịch chuyển hiển thị (S=0:không dịch chuyển hiển thị,S=1:dịch chuyển hiển thị). Hoạt động này được thực hiện trong suốt quá trình đọc/ghi dữ liệu	40uS
Điều khiển hiển thị	0	0	0	0	0	0	0	1	D	C	B	Bật/tắt hiển thị (D=0:tắt,D=1:bật) nhưng dữ liệu vẫn lưu trong DDRAM, bật/tắt con trỏ(C=0:tắt,C=1:bật) và bật tắt con trỏ nhấp nháy tại vị trí của kí tự (B=0:tắt,B=1:bật).	40uS
Dịch chuyển con trỏ/hiển thị	0	0	0	0	0	0	1	S/C	R/L	*	*	Dịch chuyển con trỏ/hiển thị qua trái /phải mà không phải đọc/ghi lại dữ liệu (S/C=0:di chuyển con trỏ,S/C=1:di chuyển hiển thị), (R/L=0:dịch trái,R/L=1:dịch phải). nội dung DDRAM không thay đổi.	40uS
Thiết lập chức năng	0	0	0	0	1	DL	N	F	*	*	*	Khởi tạo giao diện của độ dài dữ liệu (DL=0:độ dài 4 bit,DL=1:8 bit), số hàng hiển thị(N=0:1 hàng,N=1 : 2 hàng) và phòng chữ(F=0:5x7,F=1:5x10).	40uS
Thiết lập địa chỉ CGRAM	0	0	0	1	Địa chỉ CGRAM						Thiết lập địa chỉ bộ nhớ tạo kí tự (CGRAM.), dữ liệu được gửi/nhận sau thiết lập này	40uS	

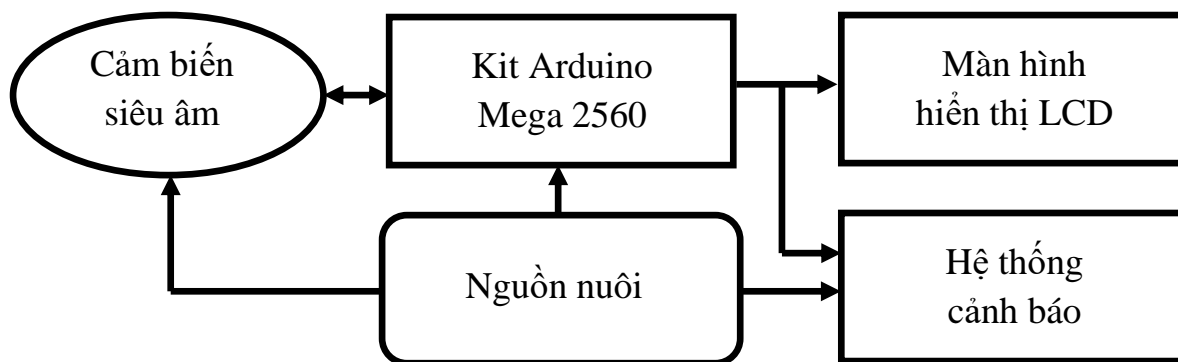
Tập lệnh	Mã nhị phân										Mô tả	Thời gian thực thi
	RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0		
Thiết lập địa chỉ DDRAM	0	0	1	Địa chỉ DDRAM						Thiết lập địa chỉ bộ nhớ tạo kí tự (DDRAM.), dữ liệu được gửi/nhận sau thiết lập này		40uS
Đọc cờ bận	0	1	BF	Địa chỉ CGRAM/ DDRAM						Đọc cờ bận Busy-flag (BF), kiểm tra xem hệ thống có đang thực thi 1 lệnh đã được nhận trước đó không. (BF=1 :hệ thống đang thực hiện tác vụ bên trong, khi BF=0 thì lệnh tiếp theo mới được thực thi)		0uS
Ghi dữ liệu đến CGRAM/ DDRAM.	1	0	Ghi dữ liệu						Ghi dữ liệu đến CGRAM/ DDRAM.		40uS	
Đọc dữ liệu từ CGRAM/ DDRAM.	1	1	Đọc dữ liệu						Đọc dữ liệu từ CGRAM/ DDRAM.		40uS	

Mã (Hex)	Lệnh đến thanh ghi của LCD
1	Xoá màn hình hiển thị
2	Trở về đầu dòng
4	Giả con trỏ (dịch con trỏ sang trái)
6	Tăng con trỏ (dịch con trỏ sang phải)
5	Dịch hiển thị sang phải
7	Dịch hiển thị sang trái
8	Tắt con trỏ, tắt hiển thị
A	Tắt hiển thị, bật con trỏ
C	Bật hiển thị, tắt con trỏ
E	Bật hiển thị, nhấp nháy con trỏ
F	Tắt con trỏ, nhấp nháy con trỏ
10	Dịch vị trí con trỏ sang trái
14	Dịch vị trí con trỏ sang phải
18	Dịch toàn bộ hiển thị sang trái

Chương 2. Thiết kế hệ thống đo dùng cảm biến siêu âm

2.1. Thiết kế hệ thống thu phát siêu âm đo khoảng cách

Phần này sẽ trình bày toàn bộ thiết kế hệ thống đo khoảng cách sử dụng cảm biến siêu âm. Sơ đồ khối hệ thống được trình bày trên hình 2.1.



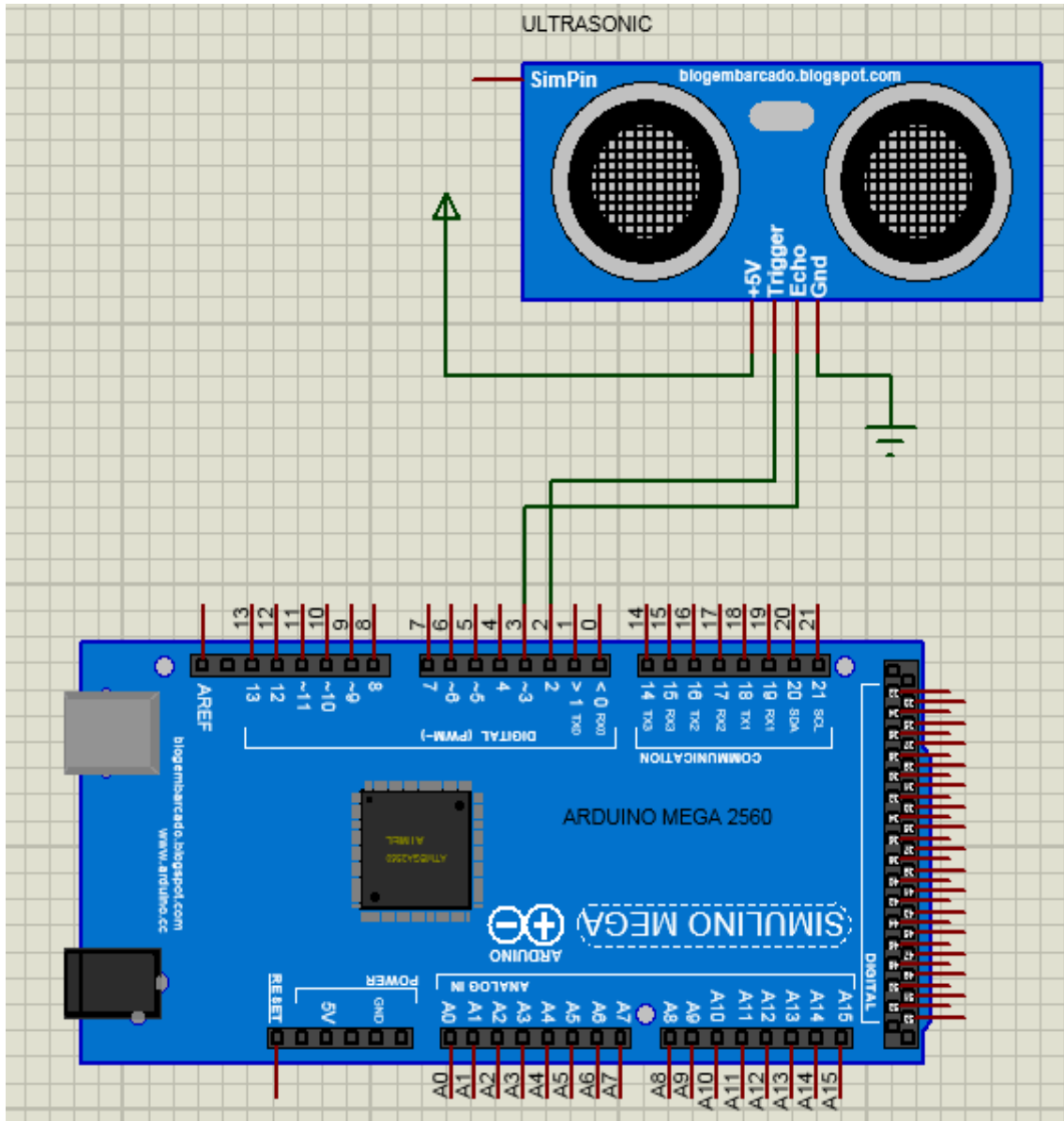
Hình 20. Sơ đồ khối hệ thống đo.

Để thiết kế chi tiết hệ thống, đề án trình bày sơ đồ khối hệ thống bao gồm các khối cảm biến, Kit Arduino (sử dụng Kit Mega 2560), khối hiển thị, khối nguồn nuôi và cảnh báo.

Hệ thống đo dùng cảm biến siêu âm được ứng dụng nhiều trong thực tiễn. Chúng ta có thể sử dụng cảm biến để đo khoảng cách, đo mức hoặc cảnh báo. Tùy thuộc loại cảm biến như đã trình bày trong chương 1 mà ta có được khoảng cách đo khác nhau. Khi sử dụng cho các mục đích cảnh báo như đo mức, đo khoảng cách tiệm cận và báo động chống trộm thì ta thiết kế thêm hệ thống cảnh báo bằng âm thanh, hình ảnh v.v.

2.2 Thiết kế phần cứng giao tiếp cảm biến và Kit Arduino

Các cảm biến siêu âm SRF04, SRF05 hay SRF10 đều cho tín hiệu dạng nhị phân, do đó khi thiết kế giao tiếp ta có thể kết nối trực tiếp các đường tín hiệu của cảm biến với các chân vào/ra của Kit Arduino. Trong thiết kế ứng dụng của đề án sử dụng Kit Arduino Mega 2560 và cảm biến siêu âm SRF05. Thiết kế phần cứng của hệ thống được trình bày trên hình 2.2 dưới đây.



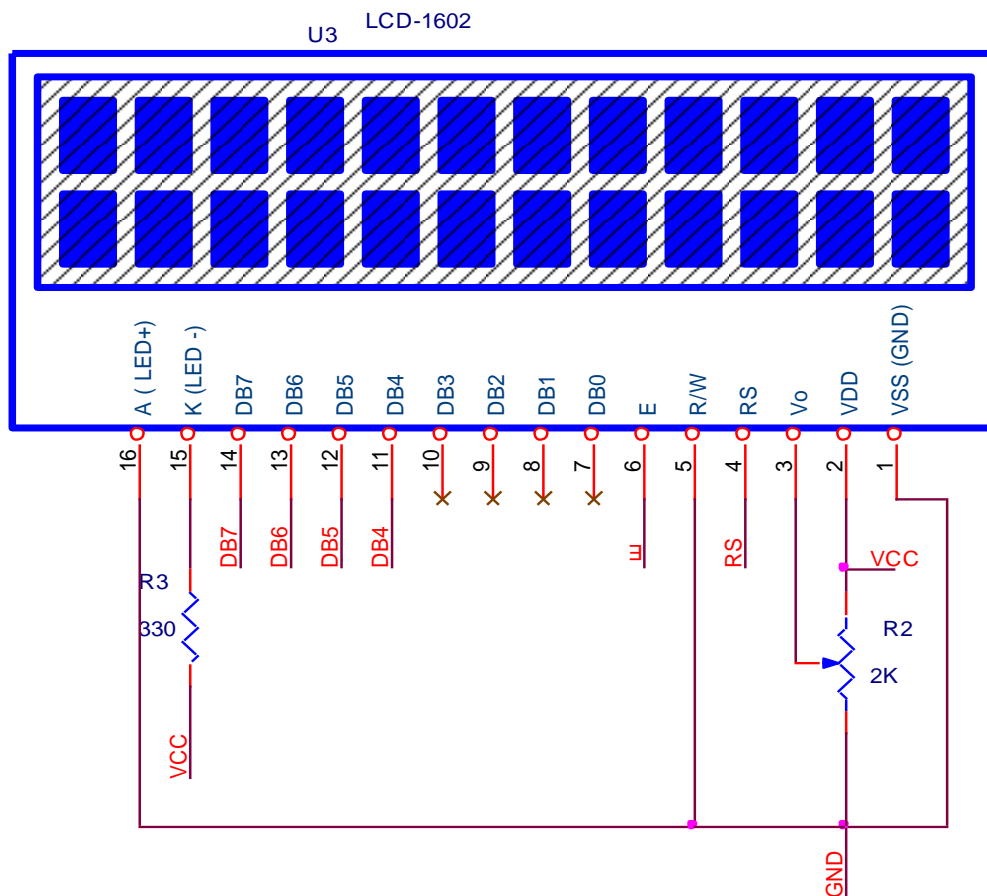
Hình 2.2. Sơ đồ giao tiếp phần cứng giao tiếp cảm biến siêu âm và Kit Arduino.

Theo đó chân Echo và Trig của cảm biến lần lượt được giao tiếp với các chân 3 và 2 của Kit Arduino Mega 2560.

2.3. Khởi hiển thị.

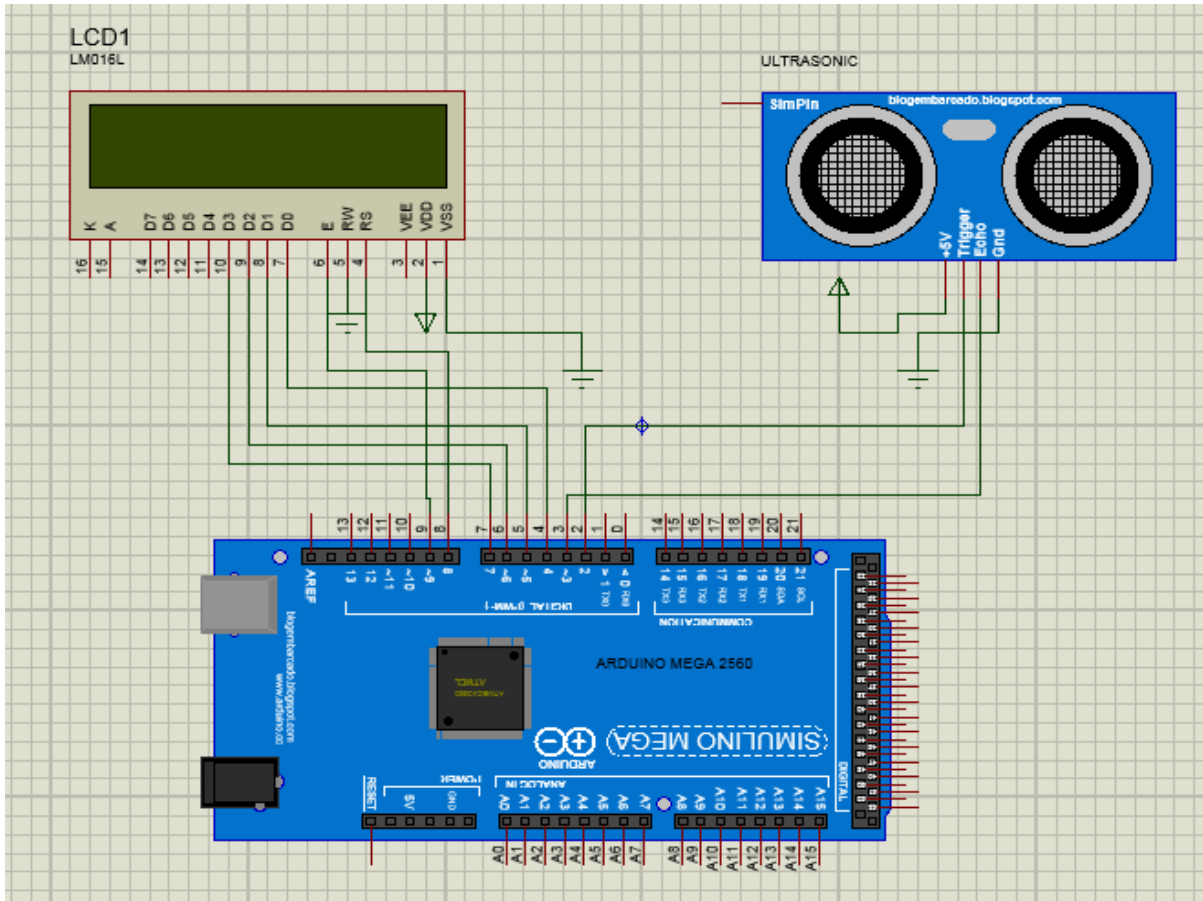
Ngày nay, thiết bị hiển thị LCD (Liquid Crystal Display) được sử dụng trong rất nhiều các ứng dụng của VĐK. LCD có rất nhiều ưu điểm so với các dạng hiển thị khác như nó có khả năng hiển thị kí tự đa dạng, trực quan (chữ, số và kí tự đồ họa), dễ dàng đưa vào mạch ứng dụng theo nhiều giao thức giao tiếp khác nhau, tốn rất ít tài nguyên hệ thống và giá thành rẻ... Trong đồ án này

em sử dụng HD44780 của Hitachi, một loại thiết bị hiển thị LCD rất thông dụng ở nước ta, cụ thể là sử dụng LCD_DM 1602A (1 dòng của HD44780).



Hình 2.3. Sơ đồ nguyên lý của khối hiển thị LCD

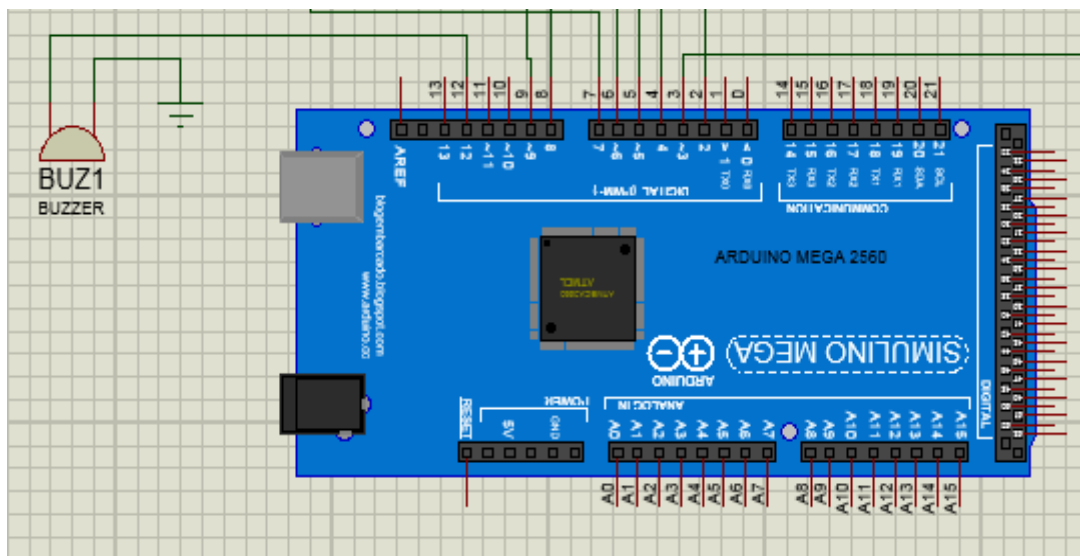
LCD1602 là loại 2 dòng, 16 ký tự, sử dụng nguồn nuôi thấp (từ 2,5 đến 5V). Có thể hoạt động ở hai chế độ 4 bit hoặc 8 bit (trong đồ án này em sử dụng chế độ 4 bit). Với đầu vào 4 bit được lấy từ 4 chân D4→D7 của LCD nối từ các chân 4,5,6 và 7 của Kit Arduino Mega 2560. Chân RW đóng vai trò chọn chế độ đọc ghi cho LCD, mức logic “0” LCD hoạt động ở chế độ ghi, ngược lại ở chế độ đọc. Chân RS của LCD được nối với chân 8 của Kit. Chân E của LCD được nối với chân 9 của Kit. Các tín hiệu điều khiển cho phép hiển thị trên LCD được thực hiện thông qua lập trình. Sơ đồ kết nối LCD và Kit Arduino được trình bày trên hình 2.4.



Hình 2.4. Sơ đồ giao tiếp khởi hiển thị LCD

2.4. Thiết kế khối cảnh báo

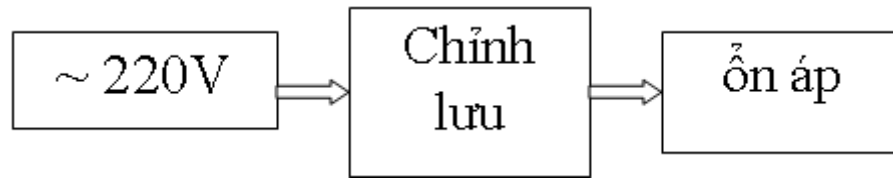
Để có thể cảnh báo, báo động tại chỗ chúng ta có thể sử dụng các loại loa, còi hoặc bất kỳ thiết bị tạo tín hiệu âm thanh nào. Sơ đồ đơn giản như hình 2.5.



Hình 2.5. Sơ đồ cảnh báo bằng âm thanh.

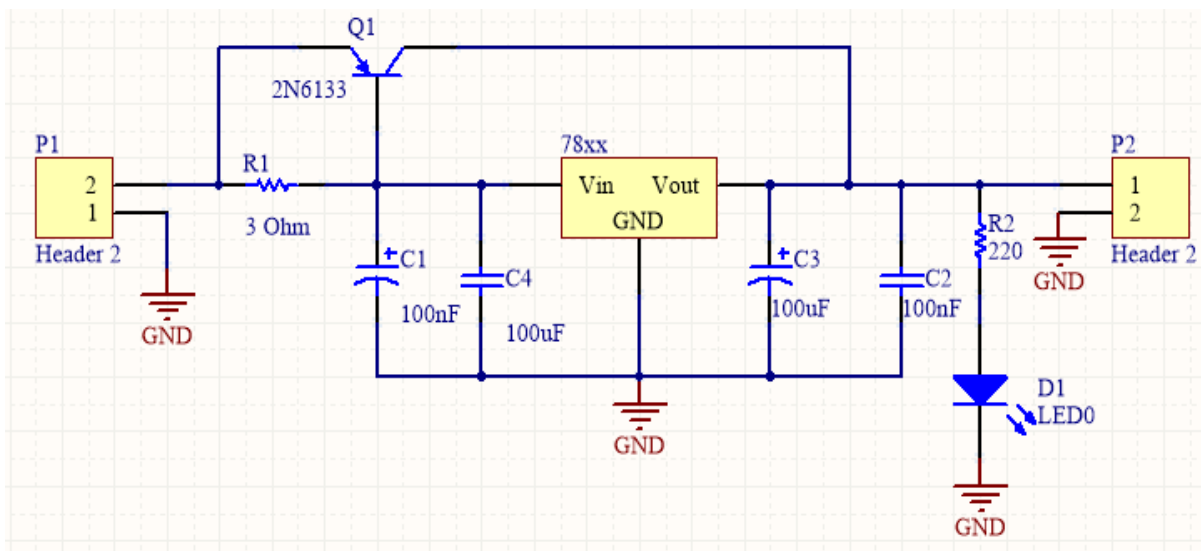
2.5 Khối nguồn.

Khối nguồn làm nhiệm vụ cung cấp nguồn nuôi cho toàn bộ hệ thống. Sơ đồ khối bộ nguồn như trong hình 2.6.



Hình 2.6 Sơ đồ khối bộ nguồn.

Ở đây bộ ổn áp dùng các IC 7805 để tạo các nguồn +5V ổn định cấp cho toàn mạch. Hình 2.7 là sơ đồ bộ nguồn +5V, tụ C1,C2, C3 và C4 để lọc nhiễu, diode D1 có nhiệm vụ báo nguồn.

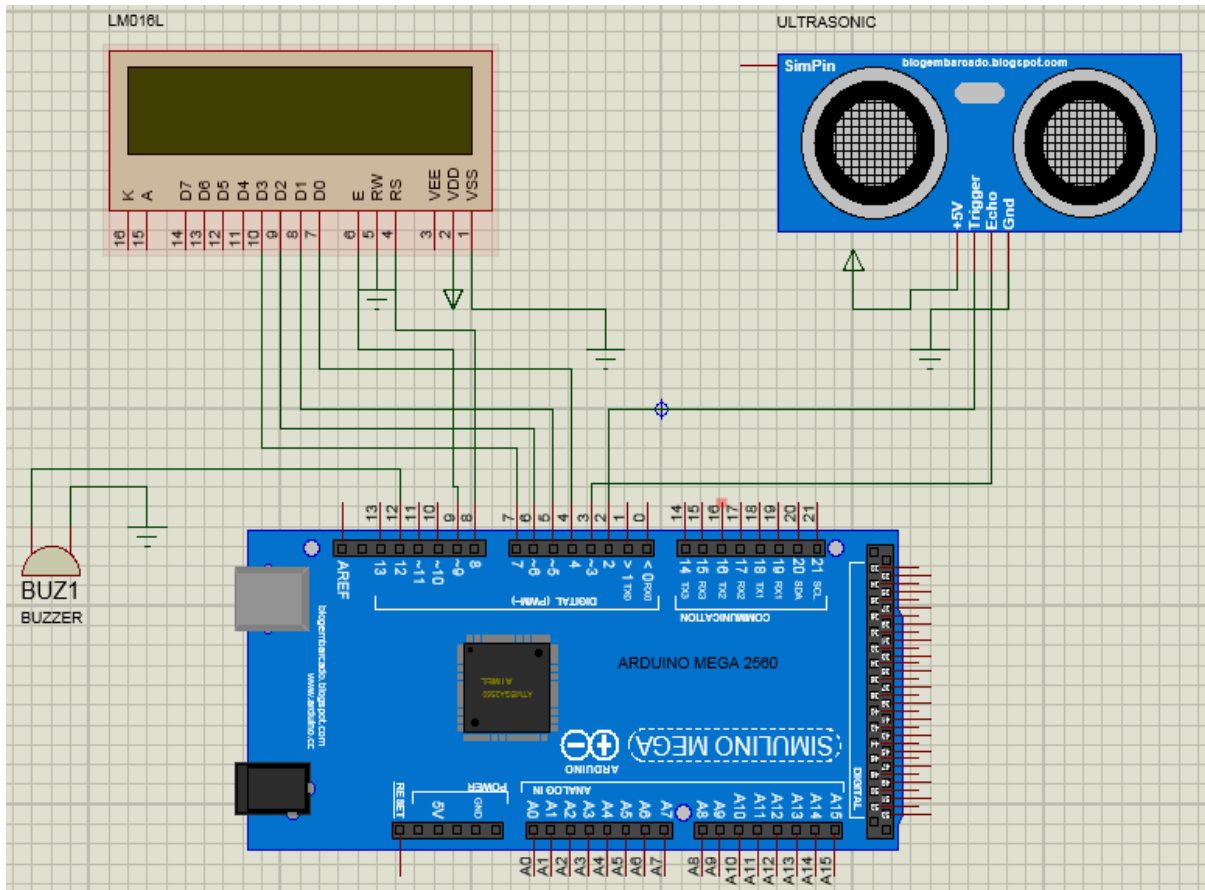


Hình 2.7 Sơ đồ nguyên lý của bộ ổn áp.

Sơ đồ khối của cả hệ thống được đưa ra trên hình 2.8. Bộ nguồn được thiết kế và chế tạo tách rời bộ Kit Arduino Mega 2560. Hoạt động của hệ thống như sau:

Cảm biến siêu âm được kết nối với Arduino thông qua các chân 2, 3 để lấy tín hiệu đo khoảng cách. Tín hiệu này được chuyển đổi thành giá trị tính theo đơn vị Cm và Inch. Giá trị đo được đưa lên hiển thị trên màn hình LCD 1602 kết nối với Kit Arduino Mega2560 qua nửa byte cao từ D4 đến D7 trên

các chân vào ra tương ứng từ 4 đến 7. Tín hiệu cảnh báo đưa ra loa hoặc Buzzer qua chân 12 của Kit. Như một ví dụ ứng dụng cho cảnh báo tiến, lùi và tránh va chạm với các vật thể xung quanh khi tài xế điều khiển xe ô tô, tác giả thêm phần cảnh báo cho hệ thống khi khoảng cách nhỏ hơn 5 cm.



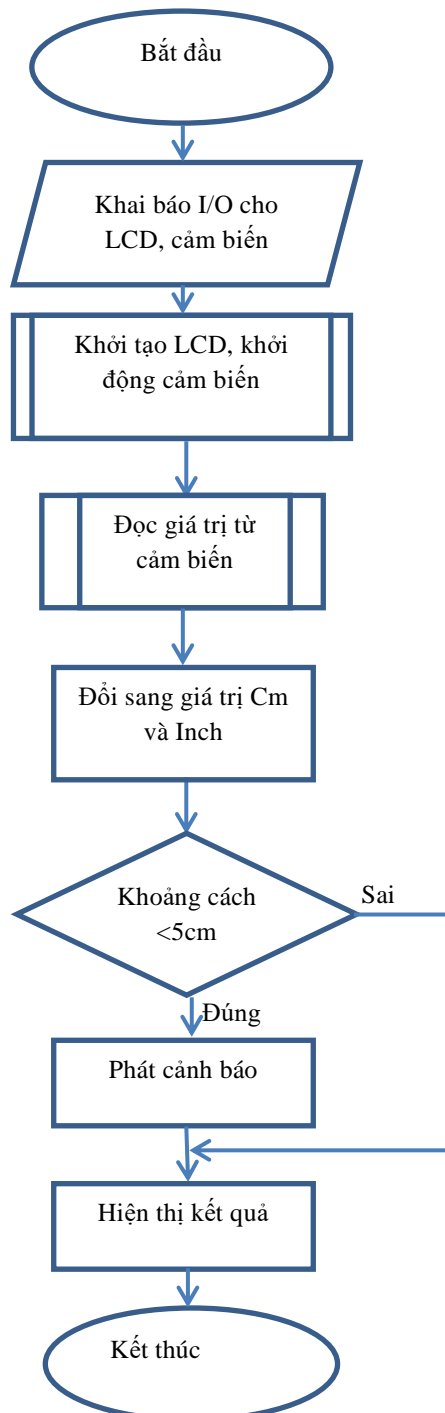
Hình 2.8 Sơ đồ nguyên lý hệ thống đo sử dụng cảm biến siêu âm.

2.6. Lưu đồ thuật toán

Qua việc nghiên cứu và thiết kế phần cứng để có thể điều khiển được cảm biến siêu âm hoạt động tốt thì cần phải viết chương trình thỏa mãn những yêu cầu như:

- Thiết lập được giao tiếp với SRF qua giao tiếp với Arduino.
- Nhận kết quả từ cảm biến và lọc nhiễu.
- Hiển thị kết quả lên LCD.

Lưu đồ thuật toán cho hệ thống trên hình 2.9. Code chương trình được trình bày ở phụ lục 1 của đề án.



Hình 2.9. Lưu đồ đo khoảng cách.

Kết luận

Sau thời gian đồ án, kết quả em đã thực hiện được những nội dung chính như sau:

- Nghiên cứu tổng quan về sóng siêu âm và cảm biến siêu âm;
- Nghiên cứu một số Kit Arduino thường dùng như Arduino Uno, Mega2560;
- Tìm hiểu màn hình LCD 1602
- Thiết kế các thành phần của hệ thống đo dùng cảm biến siêu âm;
- Xây dựng lưu đồ thuật toán và thực hiện viết chương trình cho Kit Arduino.

Lời cảm ơn

Sau thời gian nghiên cứu và được sự hướng dẫn tận tình của thầy Đỗ Anh Dũng và thầy Đoàn Hữu Chúc em đã hoàn thành đồ án tốt nghiệp với đề tài: **“Thiết kế hệ thống đo khoảng cách dùng cảm biến siêu âm”** đúng thời gian quy định. Tuy nhiên do kiến thức còn hạn hẹp nên không thể tránh khỏi những sai sót trong quá trình thực hiện đồ án. Vì vậy em mong các thầy cũng như các bạn trong lớp góp ý để đề tài của em được hoàn hảo hơn.

Trong thời gian học tập tại trường em xin chân thành cảm ơn tất cả các thầy cô giáo trong Khoa Điện – Điện tử đã luôn giúp đỡ và truyền đạt nhiều kiến thức quý báu. Đó là nền tảng giúp em thực hiện đồ án tốt nghiệp này.

Em xin chân thành cảm ơn các thầy !

Hải phòng , ngày tháng năm 2022

Sinh viên thực hiện

Nguyễn Trọng Ngãi

Phụ lục

```
// Chương trình đo khoảng cách dùng cảm biến siêu âm
// và thực hiện cảnh báo với giá trị đặt trước (5 Cm)

#include <LiquidCrystal.h>

LiquidCrystal lcd(8, 9, 4, 5, 6, 7);

const int TRIG_PIN = 2;

const int ECHO_PIN = 3;

int speakerPin = 12;//Chân được nối với loa hoặc buzzer

void setup()
{
  pinMode(1, OUTPUT);
  // Đặt thông số cho cổng nối tiếp
  Serial.begin(9600);
  // Hiển thị LCD
  lcd.begin(16,2);
  lcd.print("HPU va Arduino");
  lcd.setCursor(0,1);
  lcd.print("Khoang cach: ");
  //lcd.createChar(1,degree);
  pinMode(TRIG_PIN,OUTPUT);
  pinMode(ECHO_PIN,INPUT);
}
```

```

void loop()
{
    long chuyen_doi, Bien_Cm, Bien_In;

    digitalWrite(TRIG_PIN, LOW);

    delayMicroseconds(2);

    digitalWrite(TRIG_PIN, HIGH);

    delayMicroseconds(10);

    digitalWrite(TRIG_PIN, LOW);

    chuyen_doi = pulseIn(ECHO_PIN,HIGH);

    // Chuyển đổi từ thời gian sang khoảng cách

    Bien_Cm = chuyen_doi / 29.1 / 2 ;

    Bien_In = chuyen_doi / 74 / 2;

    lcd.setCursor(12,1);

    //lcd.print(round(distanceCm));

    lcd.print(Bien_Cm);

    lcd.print(" ");

    lcd.setCursor(14,1);

    lcd.print("Cm");

    if (Bien_Cm <= 0)

    {

        Serial.println("Ngoai khoang do");

    }
}

```

```
else
{
  Serial.print(Bien_In);
  Serial.print("in: ");
  Serial.print(Bien_Cm);
  Serial.print("cm");
  Serial.println();
}
// Thực hiện báo động nếu khoảng cách nhỏ hơn 5 Cm
if (Bien_Cm<5)
{
  tone(speakerPin, 900, 1000);
}
else
{
  noTone(speakerPin);
}
delay(1000);
}
```

Tài liệu tham khảo

1. Adeel Javed, “Building Arduino Projects for the Internet of Things”, Apress, 2016.
2. Hà Quang Phúc, “Lập Trình Điều Khiển Trên Arduino Cho Hệ Vạn Vật Kết Nối (IoT)”, NXB Thanh Niên, 2020.
3. <http://arduino.vn/>
4. <https://www.arduino.cc/>
5. Jack Purdum, “Beginning C for Arduino”, Apress, 2012.
6. Lê Mỹ Hà, Phạm Quang Huy, “Lập trình IoT với Arduino”, NXB Thanh niên, 2019.
7. Marco Schwartz, “Internet of Things with Arduino Cookbook”, Packt Publishing Ltd, 2016.
8. Nguyễn Vũ Quỳnh, Phạm Quang Huy, “Giáo trình đo lường cảm biến (Lý thuyết – Thực hành)”, NXB Thanh Niên, 2020.
9. TS Đoàn Hữu Chức, Báo cáo tổng hợp kết quả nghiên cứu khoa học đề tài “Ứng dụng cảm biến qua Arduino” .