

ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH



THIẾT KẾ HỆ THỐNG NHÚNG
HỆ THỐNG CẢNH BÁO ĐIỂM MÙ XE Ô TÔ
DÙNG SÓNG SIÊU ÂM

Giảng viên: ThS. Nguyễn Phan Hải Phú

Lớp : L03

Nhóm	
Họ và tên	MSSV
Nguyễn Văn Cường	2112971
Võ Nguyễn Tiến Đạt	2111036
Nguyễn Khánh Ly	2114002

TÓM TẮT BÀI TẬP LỚN

Đồ án này trình bày về lý thuyết và quá trình thiết kế Hệ thống cảnh báo điểm mù xe ô tô dùng sóng siêu âm. Đồ án được chia thành các thành phần sau:

Phần 1: Giới thiệu đề tài.

Phần 2: Lý thuyết về tính khoảng cách dùng cảm biến siêu âm, các điểm mù của xe ô tô, cảm biến siêu âm HC-SR04, ATmega8A.

Phần 3: Thiết kế và thực hiện phần cứng: đặc tả hệ thống, sơ đồ khối tổng quát, các linh kiện sử dụng

Phần 4: Thiết kế và thực hiện phần mềm: yêu cầu, lưu đồ giải thuật và xây dựng chương trình

Phần 5: Kết quả thực hiện

Phần 6: Kết luận và hướng phát triển đề tài.

Phần 7: Tài liệu tham khảo.

MỤC LỤC

1. GIỚI THIỆU	1
1.1 Tổng quan.....	1
1.2 Nhiệm vụ đề tài	1
2. LÝ THUYẾT	2
2.1 Đo khoảng cách dựa trên thời gian truyền sóng siêu âm	2
2.2 Điểm mù trên xe ô tô.....	2
2.3 Cảm biến siêu âm HC-SR04.....	3
2.4 Atmel ATmega8A.....	4
3. THIẾT KẾ VÀ THỰC HIỆN PHẦN CỨNG	5
3.1 Đặc tả hệ thống.....	5
3.2 Sơ đồ khối tổng quát	5
3.3 Các linh kiện sử dụng.....	6
3.4 Sơ đồ mạch	7
4. THIẾT KẾ VÀ THỰC HIỆN PHẦN MỀM	8
4.1 Yêu cầu	8
4.2 Lưu đồ giải thuật	8
5.1 Xây dựng chương trình	9
5.1.1 Khai báo các giá trị	9
5.1.2 Bắt đầu phát sóng siêu âm	9
5.1.3 Đếm thời gian truyền sóng.....	10
5.1.4 Khởi tạo LCD	11
5.1.5 Chương trình chính	12
6. KẾT QUẢ THỰC HIỆN.....	14
6.1 Mạch trên breadboard	14
6.2 Minh họa hệ thống hoạt động	14
7. KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN	16

7.1	Kết luận	16
7.2	Hướng phát triển	16
8.	TÀI LIỆU THAM KHẢO.....	17

DANH SÁCH HÌNH MINH HỌA

Hình 2.1. Các điểm mù của xe	2
Hình 2.2. Cảm biến siêu âm HC-SR04	3
Hình 2.3. Sơ đồ chân ATmega8A.....	4
Hình 3.1. Sơ đồ khối tổng quát	5
Hình 3.2. ATmega8A.....	6
Hình 3.3. Cảm biến siêu âm HC-SR04	6
Hình 3.4. LCD 16x2	6
Hình 3.5. LED	7
Hình 3.6. Sơ đồ mạch trên phần mềm mô phỏng Proteus	7
Hình 4.1. Lưu đồ giải thuật.....	8
Hình 5.1. Mạch chạy thử trên breadboard	14
Hình 5.2. Trường hợp có vật thể ở cảm biến thứ nhất	14
Hình 5.3. Trường hợp có vật thể ở cả hai cảm biến.....	15
Hình 5.4. Trường hợp có vật thể ở cảm biến thứ hai	15

DANH SÁCH BẢNG SỐ LIỆU

Bảng 2.1. Chức năng các chân của cảm biến siêu âm HC-SR04	3
---	---

1. GIỚI THIỆU

1.1 Tổng quan

Ngày nay, tình trạng tai nạn giao thông vẫn luôn là một mối lo ngại của người tham gia giao thông nói riêng và của cả xã hội nói chung, trong đó, điểm mù xe ô tô là một trong những nguyên nhân gây ra tai nạn giao thông. Điểm mù là những khu vực mà người điều khiển xe không thể quan sát trực tiếp, tiềm ẩn nguy hiểm, đặc biệt khi chuyển làn hoặc lùi xe. Với sự phát triển của công nghệ hiện nay, tỉ lệ tai nạn giao thông có thể xuống nhờ vào việc áp dụng hệ thống cảnh báo điểm mù dùng sóng siêu âm để hỗ trợ người điều khiển xe có thể chủ động tránh những tình huống không an toàn.

1.2 Nhiệm vụ đề tài

Nội dung 1: Tìm hiểu nguyên lý, lý thuyết về cách tính khoảng cách dựa trên thời gian truyền và nhận sóng siêu âm.

Nội dung 2: Tìm hiểu về cảm biến siêu âm HC-SR04, vi xử lý ATmega8A.

Nội dung 3: Thiết kế sơ đồ khối để kết nối các khối lại với nhau.

Nội dung 4: Nạp chương trình cho mạch.

Nội dung 5: Đưa ra kết quả thực hiện và nhận xét, đưa ra định hướng phát triển đề tài.

2. LÝ THUYẾT

2.1 Đo khoảng cách dựa trên thời gian truyền sóng siêu âm

Sóng siêu âm là sóng âm có tần số lớn hơn 20kHz và có thể lan truyền trong nhiều môi trường khác nhau. Dựa vào nguyên lý phản xạ sóng siêu âm, ta có thể tính toán khoảng cách dựa trên thời gian truyền và nhận sóng phản xạ. Khoảng cách cần đo giữa cảm biến siêu âm và vật thể có thể tính toán theo công thức sau:

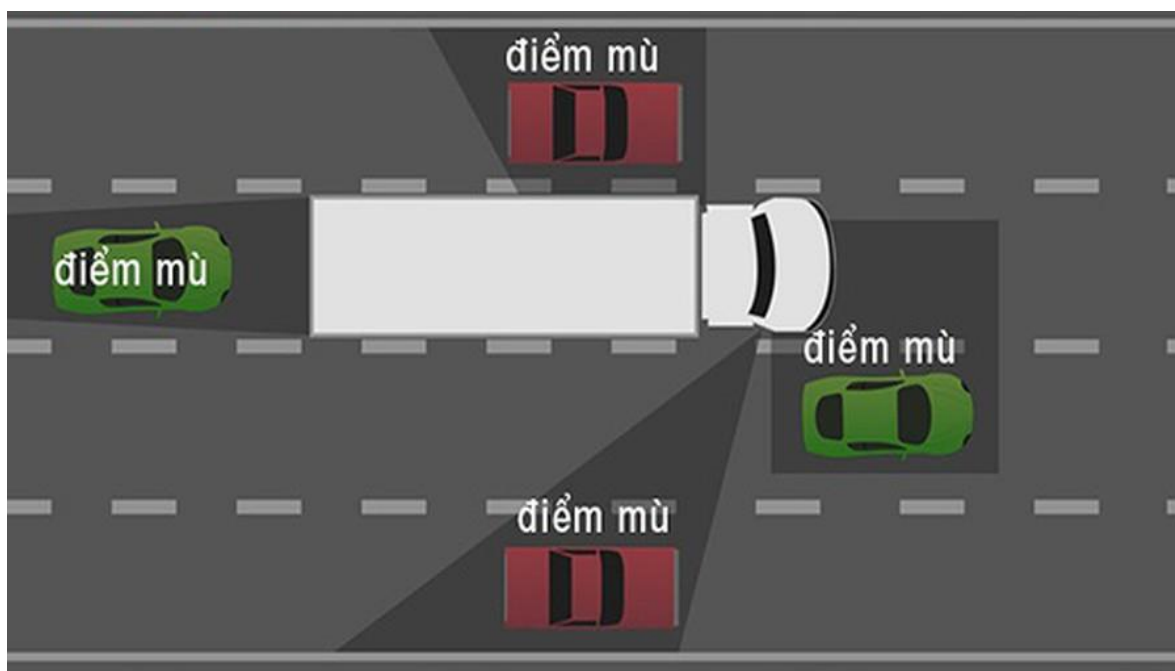
$$d = \frac{s}{2} = \frac{v \cdot t}{2}$$

Trong đó:

- d là khoảng cách giữa cảm biến siêu âm và vật thể.
- s là toàn bộ quãng đường sóng lan truyền từ cảm biến đến vật thể và từ vật thể về cảm biến siêu âm.
- v là tốc độ lan truyền sóng siêu âm.
- t là thời gian từ lúc bắt đầu phát sóng đến lúc nhận về sóng phản xạ.

2.2 Điểm mù trên xe ô tô

Điểm mù trên xe ô tô được hiểu là vùng không gian bên ngoài, xung quanh xe mà người lái xe không thể quan sát trực tiếp hoặc thông qua gương chiếu hậu khi điều khiển phương tiện lưu thông trên đường. Về cơ bản, điểm mù của xe ô tô gồm: điểm mù phía trước xe, điểm mù hai bên hông xe và điểm mù phía sau xe.



Hình 2.1. Các điểm mù của xe

2.3 Cảm biến siêu âm HC-SR04

HC-SR04 là một cảm biến siêu âm phổ biến dùng để xác định khoảng cách trong khoảng cách từ 2 -> 300 cm, với độ chính xác gần như chỉ phụ thuộc vào cách lập trình

Cảm biến hai có bộ phận chính:

- Bộ phát chuyển đổi tín hiệu điện thành sóng siêu âm.
- Bộ thu chuyển đổi tín hiệu siêu âm đó trở lại thành tín hiệu điện.

Sơ đồ chân của cảm biến siêu âm HC-SR04:

Số chân	Tên chân	Mô tả
1	Vcc	Cấp nguồn cho cảm biến, thường là +5V.
2	Trigger	Là chân điều khiển phát. Chân này phải được giữ ở mức cao trong 10us để cảm biến bắt đầu phát sóng siêu âm.
3	Echo	Là chân nhận sóng siêu âm phản xạ về. Khi chưa nhận về sóng phản xạ, mức logic của chân này ở mức cao.
4	Ground	Nối đất.

Bảng 2.1. Chức năng các chân của cảm biến siêu âm HC-SR04



Hình 2.2. Cảm biến siêu âm HC-SR04

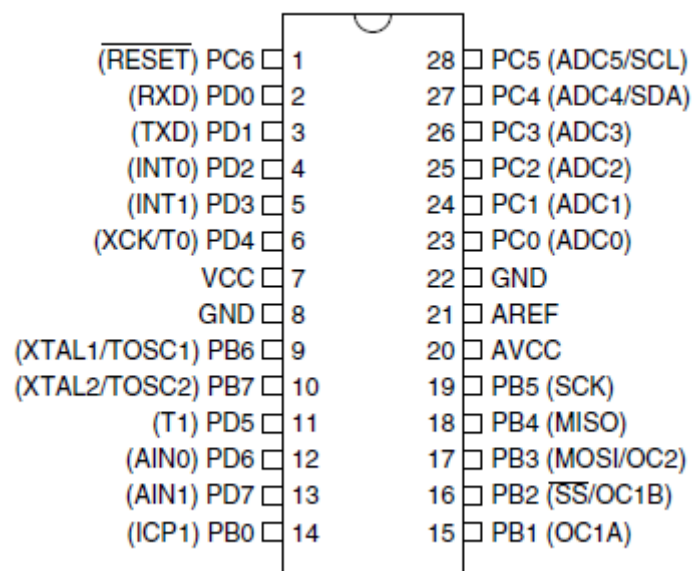
Cảm biến hoạt động dựa trên nguyên lý phản xạ sóng siêu âm. Đầu tiên, cảm biến siêu âm sẽ phát ra sóng siêu âm với tần số 40kHz. Khi sóng siêu âm được truyền đi, nếu xuất hiện chướng ngại vật trên đường đi, sóng siêu âm sẽ phản xạ lại và truyền về module nhận sóng. Khoảng cách từ cảm biến đến chướng ngại vật được tính bằng công thức sau:

$$\text{Khoảng cách} = \frac{\text{Thời gian} \times \text{Vận tốc âm thanh (340m/s)}}{2}$$

2.4 Atmel ATmega8A

Tổng quan về ATmega8A:

- Bộ nhớ Flash 8Kb.
- Bộ nhớ SRAM 1Kb.
- EEPROM 512 Bytes.
- Tần số hoạt động: 0-16MHz.
- Timer/Counter: 2 Timer 8 bits, 1 Timer 16bit.
- 23 chân I/O lập trình được.
- 3 kênh PWM.
- 6 kênh ADC 10 bit.



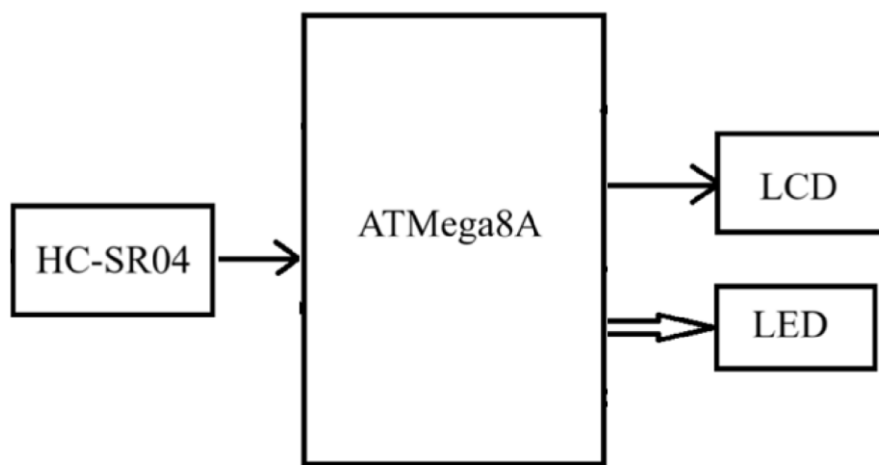
Hình 2.3. Sơ đồ chân ATmega8A

3. THIẾT KẾ VÀ THỰC HIỆN PHẦN CỨNG

3.1 Đặc tả hệ thống

- Tên (Name): Hệ thống cảnh báo điểm mù xe ô tô dùng sóng siêu âm.
- Mục đích (Purpose): phát cảnh báo khi phát hiện vật thể ở điểm mù.
- Ngõ vào/ngõ ra (Input/Output):
- Input: cảm biến siêu âm HC-SR04.
- Output: LCD, Led.
- Trường hợp sử dụng (Use cases): Nếu cảm biến siêu âm phát hiện có vật thể ở khoảng cách dưới 3m thì hệ thống sẽ bật đèn Led để cảnh báo và hiển thị khoảng cách vật thể trên LCD.
- Chức năng (Functions): Cảnh báo và hiển thị khoảng cách dựa theo khoảng cách cảm biến siêu âm đo được.
- Hiệu năng (Performance): Độ trễ giới hạn cho phép: 0,5s
- Giá thành sản phẩm (Manufacturing cost): Dưới 300.000VNĐ

3.2 Sơ đồ khối tổng quát



Hình 3.1. Sơ đồ khối tổng quát

Chức năng các khối:

- Nguồn: Cung cấp điện áp cho các linh kiện hoạt động.
- Clock: Tạo xung cho ATmega8A hoạt động.
- ATmega8A: Xử lý các tín hiệu vào và ra.
- HC-SR04: Phát sóng siêu âm và nhận sóng phản xạ.
- LCD: Hiển thị kết quả đo.
- LED: Sáng khi có vật thể ở khoảng cách gần.

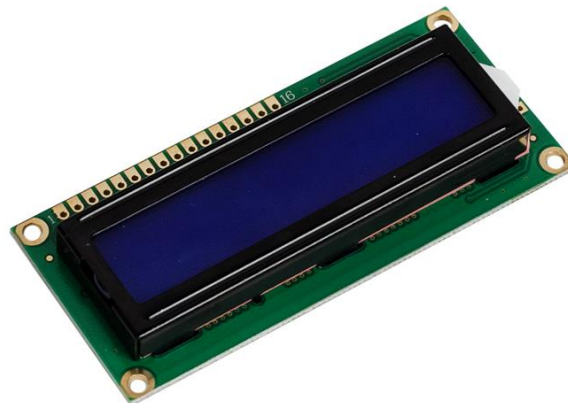
3.3 Các linh kiện sử dụng



Hình 3.2. ATMega8A



Hình 3.3. Cảm biến siêu âm HC-SR04

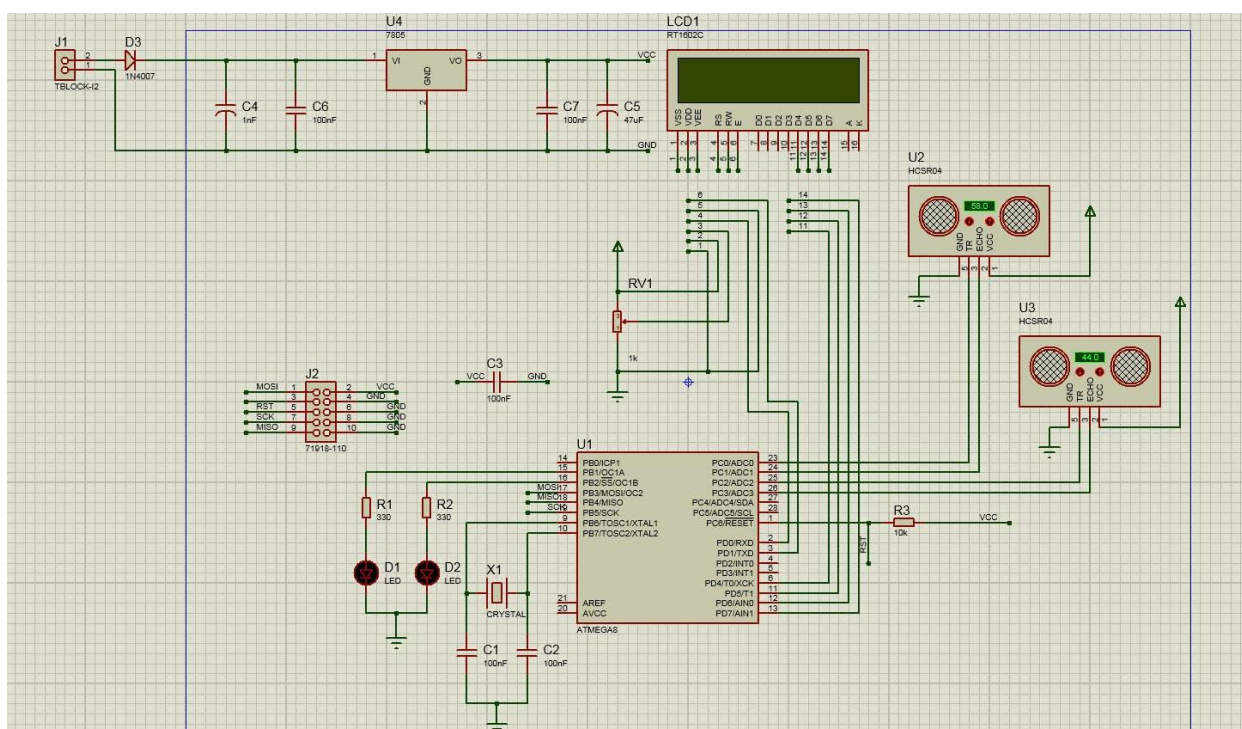


Hình 3.4. LCD 16x2



Hình 3.5. LED

3.4 Sơ đồ mạch



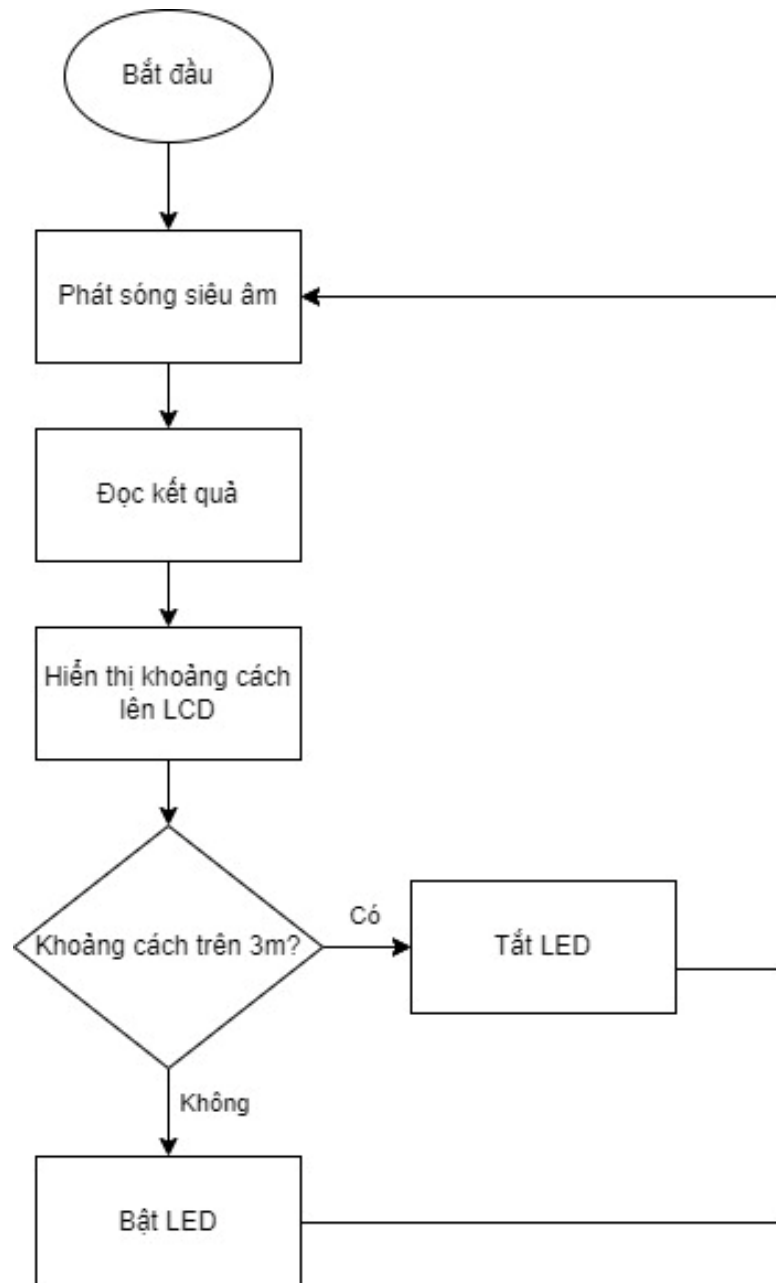
Hình 3.6. Sơ đồ mạch trên phần mềm mô phỏng Proteus

4. THIẾT KẾ VÀ THỰC HIỆN PHẦN MỀM

4.1 Yêu cầu

Hệ thống sẽ dùng cảm biến siêu âm HC-SR04 để quét các vật thể ở điểm mù của xe, nếu có vật thể trong khoảng cách dưới 3m sẽ phát sáng LED để cảnh báo cho người lái xe.

4.2 Lưu đồ giải thuật



Hình 4.1. Lưu đồ giải thuật

Khi bắt đầu hoạt động, ATmega8A sẽ phát tín hiệu mức cao cho chân TRIGGER của cảm biến siêu âm HC-SR04 để bắt đầu phát sóng siêu âm. Khi bắt đầu phát sóng siêu âm, chân ECHO sẽ giữ ở mức cao cho đến khi nhận về sóng phản xạ. ATmega8A sẽ dựa trên thời gian chân ECHO của cảm biến siêu âm ở mức cao để tính toán khoảng cách và hiển thị ra LCD, nếu có vật thể trong khoảng cách 3m, ATmega8A sẽ phát tín hiệu để sáng đèn LED cảnh báo.

4.3 Xây dựng chương trình

4.3.1 Khai báo các giá trị

```
#include <avr/io.h>
#include <util/delay.h>
#include <stdlib.h>

#define lcd_port PORTD
#define lcd_data_dir DDRD
#define rs PD0
#define en PD1

#define US_PORT PORTC
#define US_PIN PINC
#define US_DDR DDRC

#define US_TRIG_POS_1 PC0
#define US_ECHO_POS_1 PC1
#define US_TRIG_POS_2 PC2
#define US_ECHO_POS_2 PC3
#define led1 PB1
#define led2 PB2
#define US_ERROR -1
#define US_NO_OBSTACLE -2

int distance, previous_distance, distance2, previous_distance2;
```

4.3.2 Bắt đầu phát sóng siêu âm

```
void HCSR04Init()
{
    US_DDR |= (1<<US_TRIG_POS_1);
    US_DDR |= (1<<US_TRIG_POS_2);
}

void HCSR04Trigger()
{
    US_PORT |= (1<<US_TRIG_POS_1);
    _delay_us(15);
    US_PORT &= ~(1<<US_TRIG_POS_1);
}

void HCSR04Trigger2()
{
    US_PORT |= (1<<US_TRIG_POS_2);
    _delay_us(15);
    US_PORT &= ~(1<<US_TRIG_POS_2);
}
```

4.3.3 Đếm thời gian truyền sóng

```
uint16_t GetPulseWidth()
{
    uint32_t i,result;
    for(i=0;i<600000;i++)
    {
        if(!(US_PIN & (1<<US_ECHO_POS_1)))
            continue;
        else
            break;
    }

    if(i==600000)
        return US_ERROR;
    TCCR1A=0x00;
    TCCR1B=(1<<CS11);
    TCNT1=0x00;

    for(i=0;i<600000;i++)
    {
        if(US_PIN & (1<<US_ECHO_POS_1))
        {
            if(TCNT1 > 60000) break; else continue;
        }
        else
            break;
    }

    if(i==600000)
        return US_NO_OBSTACLE;

    result=TCNT1;
    TCCR1B=0x00;

    if(result > 60000)
        return US_NO_OBSTACLE;
    else
        return (result>>1);
}

uint16_t GetPulseWidth2()
{
    uint32_t t, result;
    for(t=0;t<600000;t++)
    {
        if(!(US_PIN & (1<<US_ECHO_POS_2)))
            continue;
        else
            break;
    }

    if(t==600000)
        return US_ERROR;

    t=0;
    while(US_PIN & (1<<US_ECHO_POS_2))
        t++;

    if(t==600000)
        return US_NO_OBSTACLE;
```



```

    result=t;
    t=0;

    if(result > 60000)
        return US_NO_OBSTACLE;
    else
        return (result>>1);
}

```

4.3.4 Khởi tạo LCD

```

void initialize (void)
{
    lcd_data_dir = 0xFF;
    _delay_ms(15);
    lcd_command(0x02);
    lcd_command(0x28);
    lcd_command(0x0c);
    lcd_command(0x06);
    lcd_command(0x01);
    _delay_ms(2);
}

void lcd_command( unsigned char cmnd )
{
    lcd_port = (lcd_port & 0x0F) | (cmnd & 0xF0);
    lcd_port &= ~ (1<<rs);
    lcd_port |= (1<<en);
    _delay_us(1);
    lcd_port &= ~ (1<<en);
    _delay_us(200);
    lcd_port = (lcd_port & 0x0F) | (cmnd << 4);
    lcd_port |= (1<<en);
    _delay_us(1);
    lcd_port &= ~ (1<<en);
    _delay_ms(2);
}

void lcd_clear()
{
    lcd_command (0x01);
    _delay_ms(2);
    lcd_command (0x80);
}

void lcd_print (char *str)
{
    int i;
    for(i=0; str[i]!=0; i++)
    {
        lcd_port = (lcd_port & 0x0F) | (str[i] & 0xF0);
        lcd_port |= (1<<rs);
        lcd_port |= (1<<en);
        _delay_us(1);
        lcd_port &= ~ (1<<en);
        _delay_us(200);
        lcd_port = (lcd_port & 0x0F) | (str[i] << 4);
        lcd_port |= (1<<en);
        _delay_us(1);
    }
}

```

```

        lcd_port &= ~(1<<en);
        _delay_ms(2);
    }
}

void lcd_setCursor(unsigned char x, unsigned char y)
{
    unsigned char adr[] = {0x80, 0xC0};
    lcd_command(adr[y-1] + x-1);
    _delay_us(100);
}

```

4.3.5 Chương trình chính

```

int main()
{
    initialize();
    char numberString[4];
    while(1)
    {
        uint16_t r,r2;
        _delay_ms(100);
        HCSR04Init();
        DDRB |= (1 << led1) | (1 << led2);
        while(1)
        {
            HCSR04Trigger();
            r= GetPulseWidth();
            if(r==US_ERROR)
            {
                lcd_setCursor(1, 1);
                lcd_print("Error!");
            }
            else
            {
                distance=(r*0.034/2.0);
                if (distance != previous_distance)
                {
                    lcd_clear();
                }
                lcd_setCursor(1, 1);
                lcd_print("Distance = ");
                lcd_setCursor(12, 1);
                itoa(distance, numberString, 10);
                lcd_print(numberString);
                lcd_setCursor(15, 1);
                lcd_print("cm");
                if(distance<300)
                PORTB |= (1<<led1);
                else
                PORTB &= ~(1<<led1);
                previous_distance = distance;
            }

            HCSR04Trigger2();
            r2 = GetPulseWidth2();
            if(r2==US_ERROR)
            {
                lcd_setCursor(1, 2);
                lcd_print("Error!");
            }
            else

```

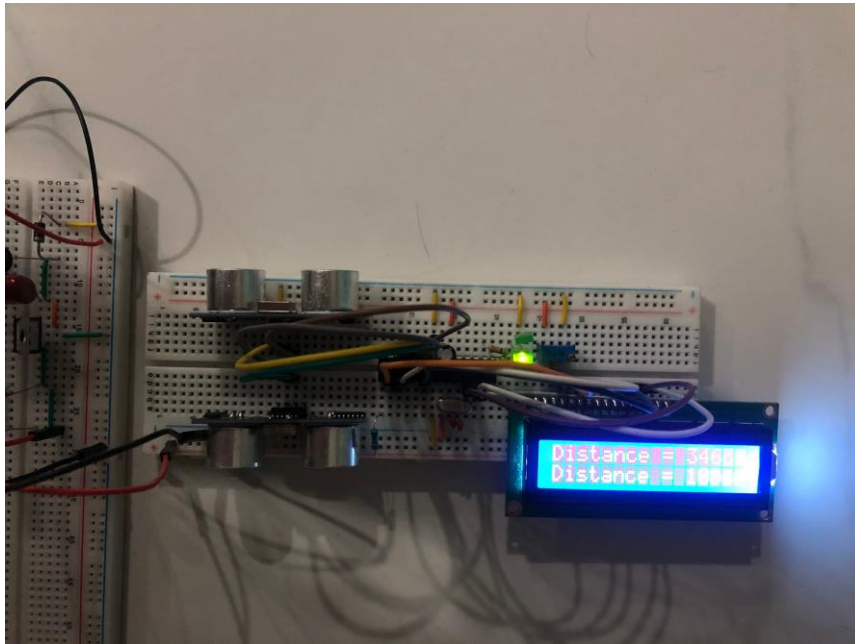
```

    {
        distance2=(r2*0.034/2.0);
        if (distance2 != previous_distance2)
        {
            lcd_clear();
        }
        lcd_setCursor(1, 2);
        lcd_print("Distance = ");
        lcd_setCursor(12, 2);
        itoa(distance2, numberString, 10);
        lcd_print(numberString);
        lcd_setCursor(15, 2);
        lcd_print("cm");
        if(distance2<300)
        PORTB |= (1<<led2);
        else
        PORTB &= ~(1<<led2);
        previous_distance2 = distance2;
    }
}
}

```

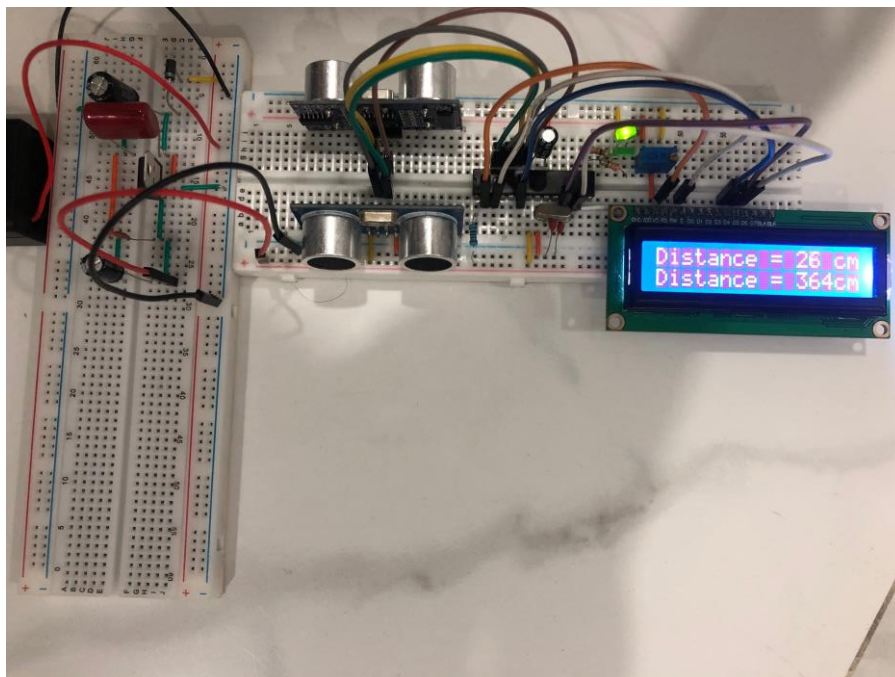
5. KẾT QUẢ THỰC HIỆN

5.1 Mạch trên breadboard

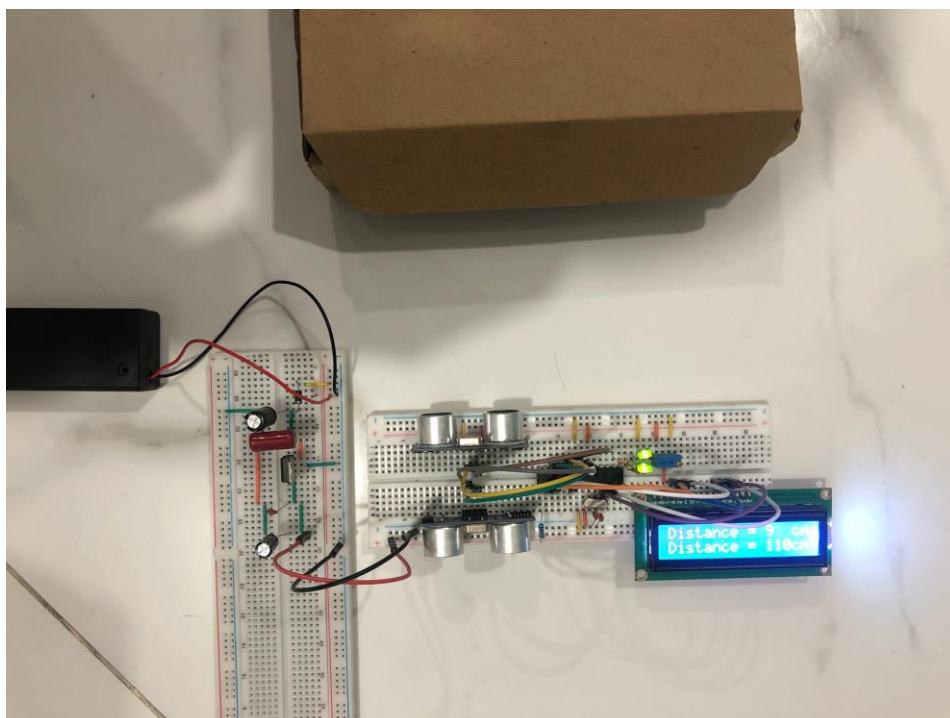


Hình 5.1. Mạch chạy thử trên breadboard

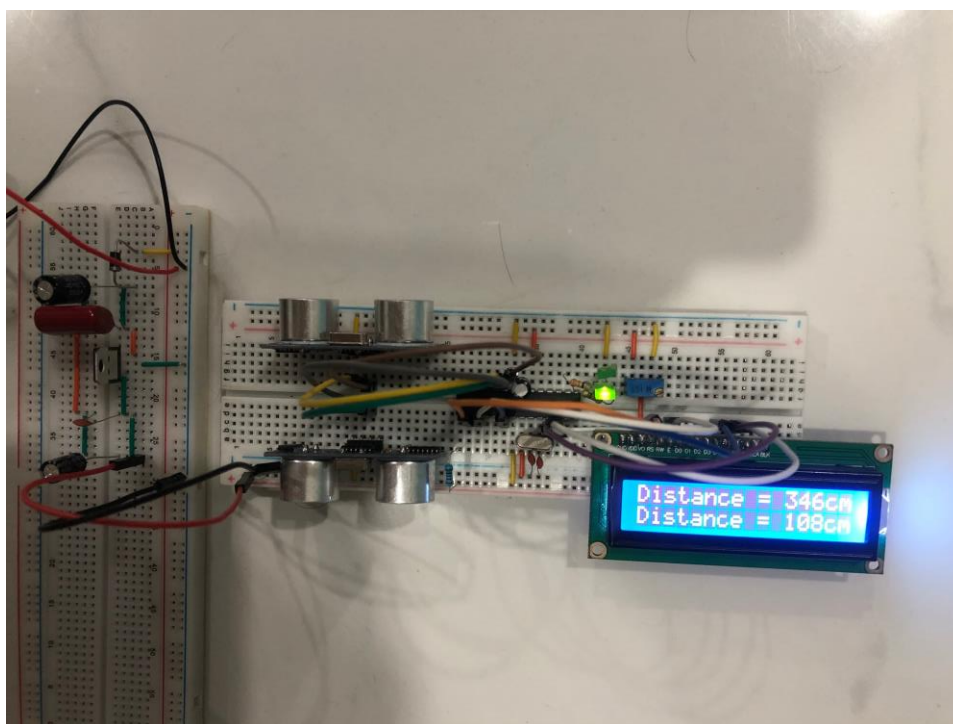
5.2 Minh họa hệ thống hoạt động



Hình 5.2. Trường hợp có vật thể ở cảm biến thứ nhất



Hình 5.3. Trường hợp có vật thể ở cả hai cảm biến



Hình 5.4. Trường hợp có vật thể ở cảm biến thứ hai

6. KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

6.1 Kết luận

Hệ thống đã được hoàn thiện ở mức độ cơ bản và có một số ưu điểm như: giá thành rẻ, dễ lắp đặt và sử dụng, có thể hoạt động không bị ảnh hưởng bởi thời tiết, ngày hoặc đêm. Bên cạnh đó, hệ thống vẫn còn tồn tại các nhược điểm như: khoảng cách có thể đo chính xác chỉ trong khoảng 2-3m và góc quét của cảm biến tương đối nhỏ.

6.2 Hướng phát triển

Một số hướng phát triển:

- Tối ưu tài nguyên phần cứng sử dụng và thời gian xử lý.
- Có thêm chế độ hoạt động khác phụ thuộc vào tốc độ của xe.

7. TÀI LIỆU THAM KHẢO

- [1] Alessio Carullo, Marco Parvis, “An Ultrasonic Sensor for Distance Measurement in Automotive Applications”, IEEE SENSORS JOURNAL, VOL. 1, NO. 2, AUGUST 2001.
- [2] L. Koval, J. Vaňuš, P. Bilík, “Distance Measuring by Ultrasonic Sensor”, IFAC-PapersOnLine 49-25 (2016) 153-158.