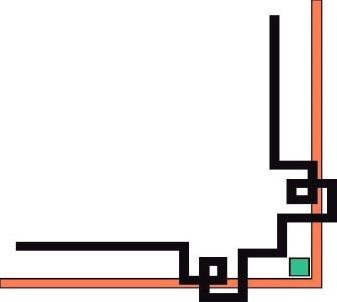
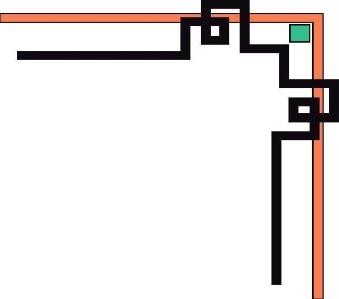
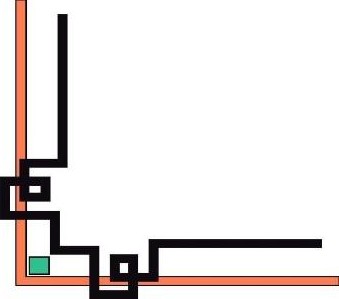
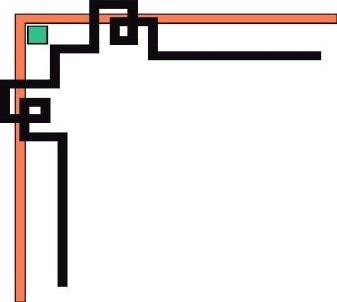
**ĐẠI HỌC QUỐC GIA HÀ NỘI**



**TRƯỜNG ĐẠI HỌC CÔNG NGHỆ**

Khoa Cơ học kỹ thuật & Tự động hóa



**BÁO CÁO BÀI TẬP LỚN**

**Môn Ứng Dụng Máy Tính Trong Đo Lường Và**

**Điều Khiển**

**Đề tài:**

**THIẾT KẾ THIẾT BỊ RADAR SỬ DỤNG SÓNG SIÊU ÂM**

*Giáo viên hướng dẫn:* ThS. Hoàng Văn Mạnh

*Nhóm thực hiện:*

Phạm Hoàng Hải 17020281

Ngô Văn Huy 17020322

Nguyễn Duy Hưng 17020334

*Hà Nội, 06/2020*

**Lời mở đầu**

Ngày nay kĩ thuật vi điều khiển đã trở lên quen thuộc trong các ngành kí thuật và dân dụng. Các bộ vi điều khiển có khả năng xử lí nhiều hoạt động phức tạp mà chỉ cần một chip vi mạch nhỏ, nó đã thay thế các tủ điều khiển lớn và phức tạp thành những mạch điện gọn nhẹ và dễ dàng thao tác sử dụng.

RADAR là viết tắt của từ “**RA**dio **D**etection **A**nd **R**anging”. Chúng có nghĩa là dò tìm và định vị bằng sóng vô tuyến. Về cơ bản, nó là một hệ thống điện từ được sử dụng để phát hiện vị trí và khoảng cách của một vật thể từ điểm đặt RADAR. Nó hoạt động bằng cách tỏa năng lượng vào không gian và theo dõi tín hiệu dội lại hoặc phản xạ từ các vật thể. Nó hoạt động trong phạm vi sóng UHF và vi sóng.

Nhằm vận dụng những kiến thức đã học trên lớp về vi điều khiển và lập trình, chúng em đã nghiên cứu và thực hiện đề tài “***Thiết kế Thiết bị RADAR sử dụng sóng siêu âm*** ” sử dụng vi điều khiển PIC16F887 dưới sự hứng dẫn của ThS. Hoàng Văn Mạnh – giảng viên trường Đại học Công Nghê – ĐHQGHN.

Trong quá trình chúng em thực hiện đề tài vẫn còn có nhiều sai sót,em mong nhận được nhiều ý kiến đóng góp từ thầy cô và các bạn.

**MỤC LỤC**

[CHƯƠNG I: GIỚI THIỆU CHUNG VỀ VI ĐIỀU KHIỂN PIC 16F887 3](#_Toc408389149)

[I. Giới thiệu chung 3](#_Toc408389150)

[1. Sơ đồ khối vi điều khiển PIC 16F887 5](#_Toc408389151)

[2. Đặc điểm vi điều khiển PIC 16F887. 8](#_Toc408389152)

[II. TỔ CHỨC BỘ NHỚ 10](#_Toc408389153)

[1. Bộ nhớ chương trình. 10](#_Toc408389154)

[2. Bộ nhớ dữ liệu 11](#_Toc408389155)

[CHƯƠNG II. PHƯƠNG PHÁP ĐO KHOẢNG CÁCH BẰNG CẢM BIẾN SIÊU ÂM VÀ TÌM HIỂU VỀ CẢM BIẾN SIÊU ÂM SRF05 34](#_Toc408389170)

[I. PHƯƠNG PHÁP ĐO KHOẢNG CÁCH BẰNG SÓNG SIÊU ÂM. 34](#_Toc408389171)

[II. CẢM BIẾN SRF05 34](#_Toc408389172)

[1. Đặc tính kĩ thuật 34](#_Toc408389173)

[2. Các chế độ của SRF05 35](#_Toc408389174)

[CHƯƠNG III. ỨNG DỤNG NGÔN NGỮ LẬP TRÌNH ASSEMBLER,C ĐIỀU KHIỂN 42](#_Toc408389176)

[I. NGÔN NGỮ ASSEMBLER 42](#_Toc408389177)

[II. NGÔN NGỮ LẬP TRÌNH C 42](#_Toc408389178)

[CHƯƠNG IV: THIẾT KẾ PHẦN CỨNG 44](#_Toc408389179)

[I. CÁC LINH KIỆN TRONG ĐỀ TÀI 44](#_Toc408389180)

[1. Điện trở 44](#_Toc408389181)

[2. Biến trở 45](#_Toc408389182)

[3. Tụ điện 45](#_Toc408389183)

[4. Thạch anh 46](#_Toc408389184)

[5. LCD 1602 46](#_Toc408389185)

[CHƯƠNG V: THIẾT KẾ PHẦN MỀM 51](#_Toc408389186)

[KẾT LUẬN 61](#_Toc408389187)

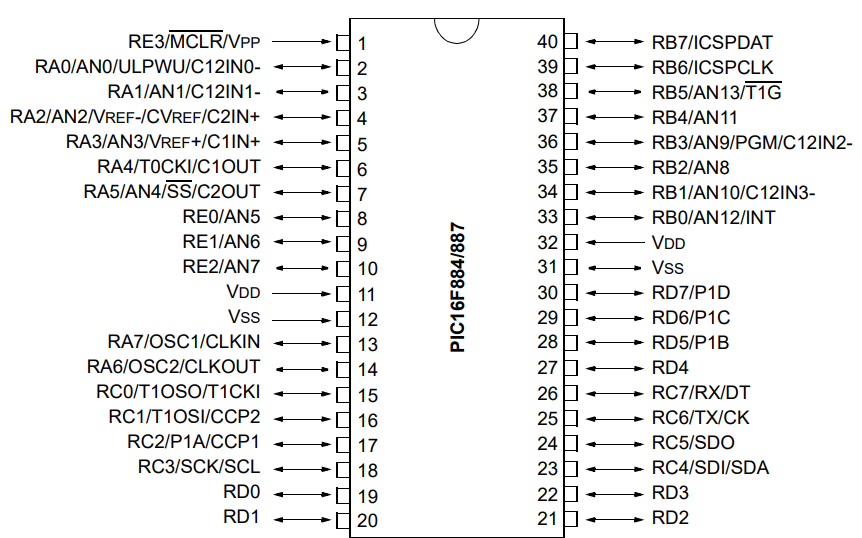
[**TÀI LIỆU THAM KHẢO** 43](#_Toc43648506)

**CHƯƠNG 1: GIỚI THIỆU CHUNG VỀ VI ĐIỀU KHIỂN PIC16F887**

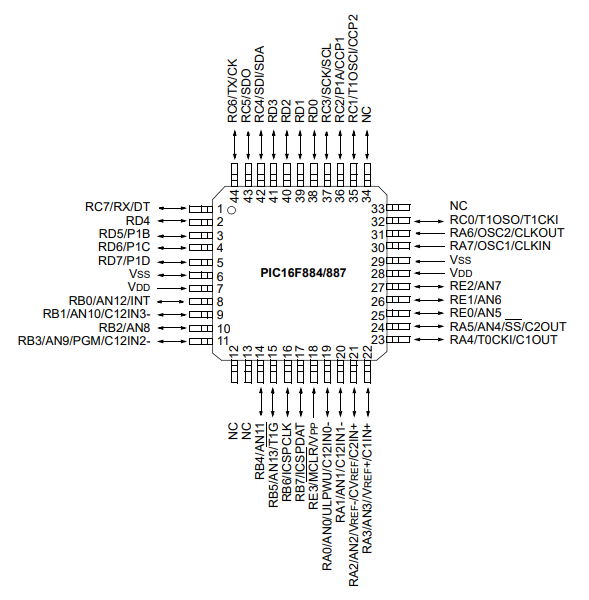
1. **Giới thiệu chung**

Cá dạng sơ đồ chân.

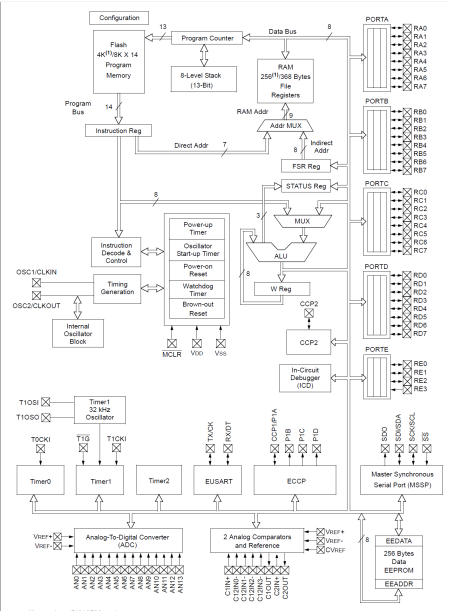
40-pin PDIP



44-pin TQFP



1. Cấu trúc bên trong của vi điều khiển 16F887:



Sơ đồ khối của PIC 16F887, gồm các khối:

* Có khối thanh ghi định cấu hình cho vi điều khiển.
* Có khối bộ nhớ chương trình có nhiều dung lượng cho 5 loại khác nhau.
* Có khối bộ nhớ ngăn xếp 8 cấp.
* Có khối bộ nhớ RAM cùng với thanh ghi FSR để tính toán tạo địa chỉ cho 2 cách truy xuất trực tiếp và gián tiếp.
* Có thanh ghi lệnh (Instruction register) dung để lưu mã lệnh nhận về từ bộ nhớ chương trình.
* Có thanh ghi trạng thái (status register) cho biết trạng thái sau khi tính toán của khối ALU.
* Có thanh ghi FSR.
* Có khối ALU cùng thanh ghi working hay thanh ghi A để xử lý dữ liệu.
* Có khối giải mã lệnh và điều khiển (Instruction Decode and Control).
* Có khối dao động nội (Internal Oscillator Block).
* Có khối dao động kết nối với 2 ngõ vào OSC1 và OSC2 để tạo dao động.
* Có khối các bộ định thời khi cấp điện PUT, có bọ định thời chờ dao động ổn định, có mạch reset khi mất điện, có bộ định thời giám sát watchdog, có mạch reset khi phát hiện sụt giảm nguồn.
* Có khối bộ dao động cho timer1 có tần số 32kHz kết nối với 2 ngõ vào T1OSI và T1OSI.
* Có khối CPP2 và ECCP.
* Có khối mạch gỡ rối (In-Circuit Debugger IDC).
* Có khối timer0 và ngõ vào xung đếm từ bên ngoài là T0CKI.
* Có khối truyền dữ liệu đồng bộ và bất đồng bộ nâng cao.
* Có khối truyền dữ liệu đồng bộ MSSP cho SPI và I2C.
* Có khối bộ nhớ EEPROM 256byte và thanh ghi quản lý địa chỉ EEADDR và thanh ghi dữ liệu EEDATA.
* Có khối chuyển dổi dữ liệu tương tự sang số ADC.
* Có khối 2 bộ so sánh với nhiều ngõ vào ra và điện áp tham chiếu.
* Có khối các port A, B, C, D và E.

1. Đặc điểm vi điều khiển PIC 16F887.

Đây là vi điều khiển thuộc họ PIC16Fxxx với tập lệnh gồm 35 lệnh có độ dài 14bit. Thời gian thực hiện tất cả các lệnh là 1 chu kỳ máy, ngoại trừ lệnh rẽ nhánh là 2. Tốc độ hoạt động ngõ vào xung clock có tần số 20MHz, chu kỳ lệnh thực hiện là 200ns. Bộ nhớ chương trình là 8Kx14bit, bộ nhớ dữ liệu là 368 byte SRAM và 256 byte EEPROM. Số port I/O là 5 với 35 pin I/O. Có 14 kênh chuyển đổi A/D 10 bit.

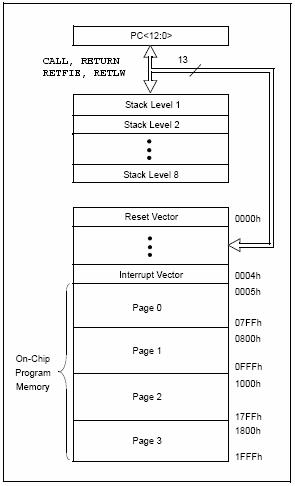
1. **Tổ chức bộ nhớ**

Cấu trúc bộ nhớ của vi điều khiển PIC 16F887 bao gồm bộ nhớ chương trình và bộ nhớ dữ liệu.

1. Bộ nhớ chương trình

Bộ nhớ chương trình có chức năng dùng để lưu chương trình và các dữ liệu cố định, khi vi xử lý truy xuất bộ nhớ chương trình để lấy mã lệnh về xử lý thì chỉ truy xuất một từ dữ liệu hay một ô nhớ duy nhất và thực hiện trong một chu kỳ lệnh duy nhất.

Bộ nhớ có dung lương 8K ô nhớ sẽ có 13 bit địa chỉ, để quản lý địa chỉ của bộ nhớ chương trình thì do thanh ghi bộ đếm chương trình ( Program Counter \_ PC) đảm nhận. Bộ nhớ được chia thành 4 trang ô nhớ, mỗi trang 2K ( từ page 0 đến page 3).

Khi vi điều khiển PIC bị reset thì thanh ghi

PC có giá trị 0000H và PIC sẽ bắt đầu thực

hiện chương trình tại địa chỉ 0000H (reset

vector).

Khi có bất kỳ ngắt nào tác động thì vi điều

khiển PIC sec thực hiện chương trình phục

vụ ngắt tại địa chỉ 0004H (Interrupt Vector).

Bộ nhớ chương trình không bao gồm bộ nhớ

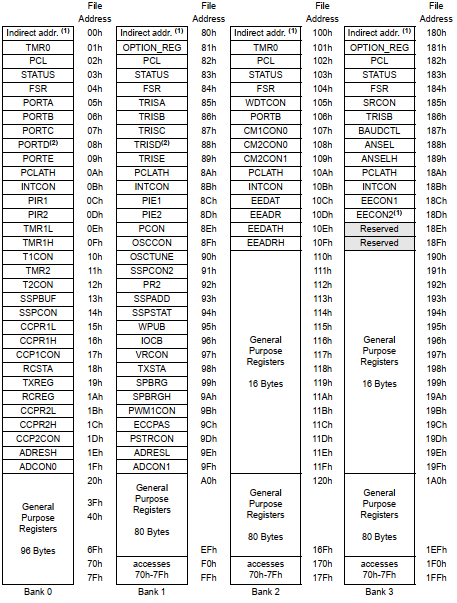
Stack và không được địa chỉ hóa bởi bộ đếm

chương trình.

1. Bộ nhớ dữ liệu

Bộ nhớ dữ liệu được phân chia thành 4 bank, mỗi bank có 128byte bao gồm một số thanh ghi chức năng đặc biệt, còn lại là các ô nhớ thông dụng có chức năng lưu trữ dữ liệu.

Bộ nhớ RAM có 512 ô nhớ và khi đó số bit địa chỉ được dùng là 9 bit (2^9 = 512), mỗi một ô nhớ có một địa chỉ duy nhất.



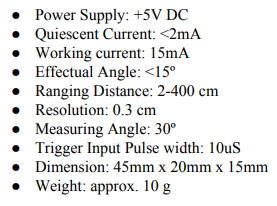
**CHƯƠNG 2: Tìm hiểu về cảm biến siêu âm HC\_SR04 và Phương pháp đo khoảng cách bằng cảm biến siêu**

Siêu âm là dạng sóng âm được ứng dụng rộng rãi trong việc đo khoảng cách và định vị vật thể. Sóng siêu âm được truyền đi trong không khí với vận tốc khoảng 343m/s. Nếu một cảm biến phát ra sóng siêu âm và thu về các sóng phản xạ đồng thời đo được khoảng thời gian từ lúc phát đi đến lúc thu về, thì máy tính có thể xác định được quãng đường mà sóng đã truyền đi trong không gian. Quãng đường di chuyển của sóng sẽ bằng 2 lần khoảng cách từ cảm biến đến vật thể theo hướng phát của sóng siêu âm.

1. Cảm biến HC\_SR04



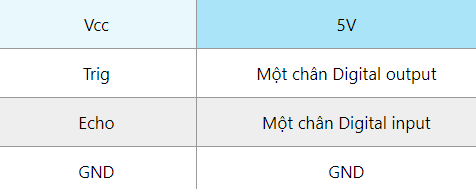
Thông số cơ bản:



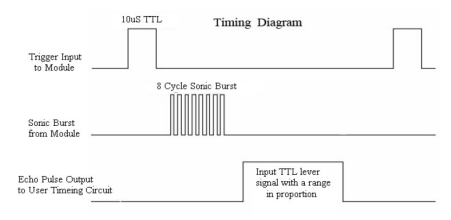
1. Đặc tính kỹ thuật

HC\_SR04 được sử dụng rất phổ biến để xác định khoảng cách vì rẻ và chính xác. Dải đo nằm trong khoảng 2 -> 300 cm, với độ chính xác gần như chỉ phụ thuộc vào cách lập trình.

Cảm biến HC\_SR04 có 4 chân: Vcc, GND, Trì, Echo

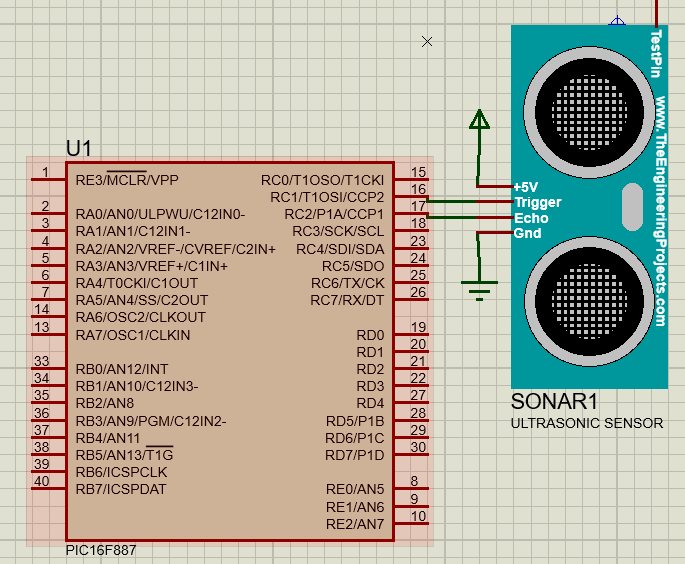


Biểu đồ thời gian của HC\_SR04:



Để đo khoảng cách tới vật cản, ta kích một xung vào chân Trigger của module cảm biến. Bình thường ta giữ chân Trigger ở mức thấp, khi ta cần phát một sóng âm đi để đo khoảng cách, cho chân Trigger này lên mức 1 và giữ ở mức 1 tối thiểu 10us. Sau đó ta cho chân Trigger trở lại mức thấp. Sau khi trở lại mức thấp, bộ phận phát trên HC\_SR04 sẽ phát ra chuỗi gồm 8 xung sóng âm với tần số 40kHz. Chân Echo đang ở mức thấp sẽ được tự động kéo lên mức cao. Sau khi 8 xung sóng âm phát đi gặp vật cản và quay trở lại bộ phận thu thì lập tức chân Echo trở lại mức thấp hoặc nếu không có sóng âm quay trở lại (không gặp vật cản) thì sau khoảng 30ms chân Echo sẽ tự động trở lại mức thấp. Như vậy chúng ta đo quãng thời gian chân Echo ở mức cao.

**Giao tiếp với vi điều khiển để đo khoảng cách:**



Đo khoảng cách chính là đo thời gian chân Echo ở mức cao.

Để đo thời gian chân Echo ở mức cao ta sử dụng Timer1 của vi điều khiển.

Các bước thực hiện:

* Kích chân Trigger: Xuất mức 1 ra chân trigger và delay tối thiểu 10us.
* Đợi chân Echo lên mức cao.
* Khi chân Echo lên mức cao, kích hoạt Timer1, cho phép bắt cạnh xuống sử dụng khối CCP1 ở chế độ capture.
* Khi chân Echo xuống mức thấp, dừng Timer1 và tính toán giá trị thời gian từ Timer1, từ đó suy ra khoảng cách.
* Tắt ngắt Timer1 và cho phép bắt cạnh lên trong khối CCP1 ở chế độ capture để chuẩn bị cho lần đo tiếp theo.

**Cách Tính ra khoảng cách từ thời gian đo được:**

Gọi S là quãng đường đi cuả sóng âm.

S = 2 x d.

Gọi V là vận tốc của sóng âm.

V = 344 m/s = 34400 cm/s.

V = 0.0334 cm/us.

Và gọi T bằng thời gian truyền ( được tính bằng cách sử dụng Timer1, micro giây), thì ta có:

S = 2 x d = V x T.

D = V x T/2 = 0.0344 x T/2 = T/58.

Như vậy: d = t/58. (cm)

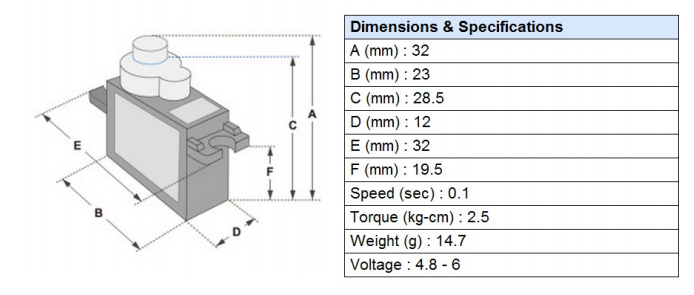
**CHƯƠNG 3: Tìm hiểu về Động cơ Servo**

1. Servo SG90:

Động cơ Servo SG90 có góc quay từ 0 đến 180 độ. Cấu tạo bên trong gồm một động cơ DC và một mạch điều khiển. Bên trên động cơ được đặt một hệ thống bánh răng (hộp giảm tốc).



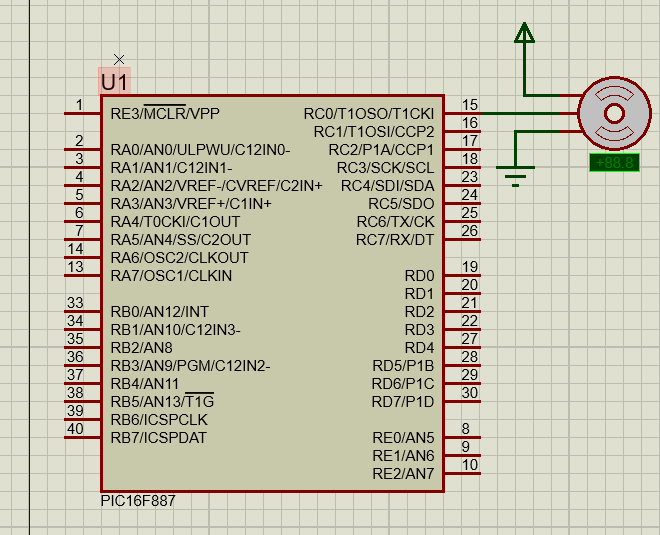
Thông số kỹ thuật:



Nguyên lý hoạt động:

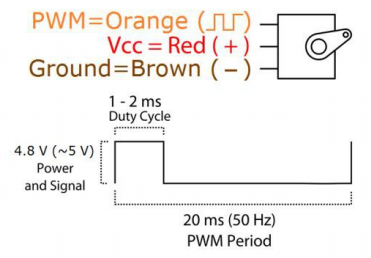
Một chiết áp được gắn vào đầu ra của trục quay (bánh răng cuối cùng), khi động cơ quay thì chiết áp cũng quay theo. Chiết áp sẽ tạo ra một điện áp và điện áp này có mối quan hệ với góc quay. Trong mạch điều khiển, điện áp từ chiết áp sẽ được so sánh với điện áp từ tín hiệu điều khiển. Bộ điều khiển sẽ kích hoạt mạch cầu H để cho phép động cơ quay theo một hướng xác định cho đến khi hai tiến hiệu này sai khác nhau bằng 0.

**Giao tiếp với vi điều khiển:**



Cách thức điều khiển:

Tín hiệu điều khiển chính là tín hiệu PWM. Chu kỳ xung PWM điều khiển động cơ là 20ms(50Hz) . Thay đổi độ rộng xung từ 1 đến 2ms (thời gian ở mức cao) thì động cơ sẽ di chuyển từ 0 đến 180 độ. Đây chỉ là số liệu từ nhà sản xuất cung cấp, trên thực tế chúng ta phải thử thì mới biết để điều khiển một cách chính xác.



**CHƯƠNG 4: THIẾT KẾ PHẦN MỀM**

Để lập trình cho vi điều khiển ta có thể dùng nhiều chương trình và ngôn ngữ lập trình khác nhau. Trong đồ án này, nhóm em sử dụng phần mềm CCS C compiler và viết chương trình bằng ngôn ngữ C.

#include<16f887.h>

#include<string.h>

#include<stdlib.h>

#fuses HS

#fuses NOWDT

#use delay(clock=20M)

#use rs232(baud=9600, xmit=pin\_c6, rcv=pin\_c7)

#bit tmr1if = 0x0C.0

#bit tmr1on = 0x10.0

#BYTE trisc=0x87

#BYTE portc=0x07

#bit RC2=0x7.2

// LCD pinout setting

#define LCD\_ENABLE\_PIN PIN\_d2 ////

#define LCD\_RS\_PIN PIN\_d0 ////

#define LCD\_RW\_PIN PIN\_d1 ////

#define LCD\_DATA4 PIN\_D4 ////

#define LCD\_DATA5 PIN\_D5 ////

#define LCD\_DATA6 PIN\_D6 ////

#define LCD\_DATA7 PIN\_D7

// HC-SR04 pinout

#define trigPin PIN\_c1

#define echoPin PIN\_c2

#include<lcd.c>

// Khai bao bien toan cuc

// Can tao ra tan so 50Hz = 20.000us

// Don vi delay nho nhat = 4u ---> dem 5000 lan

// T\_on = [530,2140](us) ---> 2,65% -> 12,05%

// Vay gia tri T\_on = [265,1205]

unsigned char b;

unsigned int16 T\_on, C;

double distance, getVar;

int1 hasEcho;

unsigned char Angle[20];

unsigned char Dist[20];

#int\_RDA

void interrupt\_uart()

{

if(kbhit())

{

b = getch();

delay\_ms(50);

}

}

#int\_CCP1 // Chuong trinh ngat CCP1 o chan RC2

void interrupt\_ccp1()

{

if(input(pin\_c2)) // Neu a xung canh len

{

set\_timer1(0); // Reset timer1 ve 0

setup\_ccp1(ccp\_capture\_fe); // Chuyen CCP1 capture canh xuong

}

if(!input(pin\_c2)) // Neu la xung canh duoi

{

getVar = CCP\_1; // CCP\_1 = tra tri TMR1

//

distance = (getVar\*0.8)/58;

setup\_ccp1(ccp\_capture\_re); // Chuyen CCP1 capture canh len

hasEcho = 1; // Co xung echo tra ve

disable\_interrupts(int\_timer1); // Tat ngat Timer1

}

}

#int\_timer1 // Chuong trinh ngat timer1 tran

void interrupt\_timer1()

{

if(!hasEcho)

{

// Tran Timer 1 ma ko co xung echo thi cho qua( Xung echo toi qua 30ms, tran Timer1 52ms)

hasEcho = 1;

distance = -1;

}

disable\_interrupts(int\_timer1);

}

void Delay\_4us(unsigned int16 t)

{

while(t>0) t--;

}

float Convert(float x, float in\_min, float in\_max, float out\_min, float out\_max)

{

float tam = (out\_max - out\_min)/(in\_max - in\_min);

tam = tam \* (x - in\_min);

tam = tam + out\_min;

return tam;

}

void Write(unsigned int16 goc, signed int16 t)

{

T\_on = Convert(goc,0,180,265,1205);

t=t/20;

for(;t>0;t--)

{

output\_high(pin\_c0);

Delay\_4us(T\_on);

output\_low(pin\_c0);

Delay\_4us(C-T\_on);

}

}

void trig()

{

output\_high(trigPin);

Delay\_4us(3);

output\_low(trigPin);

}

void lcd\_display(unsigned int8 x)

{

lcd\_putc(x/100+0x30);

lcd\_putc(x/10%10+0x30);

lcd\_putc(x%10+0x30);

}

void main()

{

set\_tris\_d(0x00);

set\_tris\_c(0x84); // 0x10000100

set\_tris\_b(0x00);

// Khoi tao timer1

setup\_timer\_1(t1\_internal | t1\_div\_by\_4);

set\_timer1(0);

// Thiet lap CCP 1 bat canh len

setup\_ccp1(ccp\_capture\_fe);

// Khoi tao cho phep ngat

enable\_interrupts(global);

enable\_interrupts(int\_rda);

enable\_interrupts(int\_timer1);

enable\_interrupts(int\_ccp1);

// Khi tao LCD

lcd\_init();

output\_b(0x00);

output\_c(0x00);

T\_on = 0;

C = 10000;

distance = 0;

getVar = 0;

hasEcho=0;

b = 49;

while(true)

{

for(unsigned int8 i=0; i<181; i++)

{

Write(i,20);

lcd\_gotoxy(1,1);

lcd\_putc("GOC QUAY:");

sprintf(Angle,"%d", i);

lcd\_gotoxy(10,1);

lcd\_display(atoi(Angle));

lcd\_gotoxy(13,1);

lcd\_putc(0xDF);

delay\_ms(1);

set\_timer1(0);

enable\_interrupts(int\_timer1);

hasEcho = 0;

trig();

output\_low(pin\_b0);

while(!hasEcho);

//! lcd\_putc("\f");

if(distance >=3 && distance <= 90)

{

if(b == '1') output\_high(pin\_b0);

else output\_low(pin\_b0);

lcd\_gotoxy(1,2);

lcd\_putc("KH/ CACH:");

lcd\_gotoxy(10,2);

sprintf(Dist,"%3.0f",distance);

lcd\_display(atof(Dist));

lcd\_gotoxy(13,2);

lcd\_putc("cm");

delay\_ms(1);

printf("%3.0f\r", distance);

}

else

{

lcd\_gotoxy(1,2);

lcd\_putc(" NO OBJECT ");

printf("%d\r",0);

}

printf("%u\r", i);

}

for(unsigned int8 x=179; x > 0; x--)

{

Write(x,20);

lcd\_gotoxy(1,1);

lcd\_putc("GOC QUAY:");

sprintf(Angle,"%d",x);

lcd\_gotoxy(10,1);

lcd\_display(atoi(Angle));

lcd\_gotoxy(13,1);

lcd\_putc(0xDF);

delay\_ms(1);

set\_timer1(0);

enable\_interrupts(int\_timer1);

hasEcho = 0;

trig();

output\_low(pin\_b0);

while(!hasEcho);

//! lcd\_putc("\f");

if(distance >=3 && distance <= 90)

{

if(b == '1') output\_high(pin\_b0);

else output\_low(pin\_b0);

lcd\_gotoxy(1,2);

lcd\_putc("KH/ CACH:");

lcd\_gotoxy(10,2);

sprintf(Dist,"%3.0f",distance);

lcd\_display(atof(Dist));

lcd\_gotoxy(13,2);

lcd\_putc("cm");

delay\_ms(1);

printf("%3.0f\r", distance);

}

else

{

lcd\_gotoxy(1,2);

lcd\_putc(" NO OBJECT ");

printf("%d\r",0);

}

printf("%u\r", x);

}

}

}

#include<16f887.h>

#include<string.h>

#include<stdlib.h>

#fuses HS

#fuses NOWDT

#use delay(clock=20M)

#use rs232(baud=9600, xmit=pin\_c6, rcv=pin\_c7)

#bit tmr1if = 0x0C.0

#bit tmr1on = 0x10.0

#BYTE trisc=0x87

#BYTE portc=0x07

#bit RC2=0x7.2

// LCD pinout setting

#define LCD\_ENABLE\_PIN PIN\_d2 ////

#define LCD\_RS\_PIN PIN\_d0 ////

#define LCD\_RW\_PIN PIN\_d1 ////

#define LCD\_DATA4 PIN\_D4 ////

#define LCD\_DATA5 PIN\_D5 ////

#define LCD\_DATA6 PIN\_D6 ////

#define LCD\_DATA7 PIN\_D7

// HC-SR04 pinout

#define trigPin PIN\_c1

#define echoPin PIN\_c2

#include<lcd.c>

// Khai bao bien toan cuc

// Can tao ra tan so 50Hz = 20.000us

// Don vi delay nho nhat = 4u ---> dem 5000 lan

// T\_on = [530,2140](us) ---> 2,65% -> 12,05%

// Vay gia tri T\_on = [265,1205]

unsigned char b;

unsigned int16 T\_on, C;

double distance, getVar;

int1 hasEcho;

unsigned char Angle[20];

unsigned char Dist[20];

#int\_RDA

void interrupt\_uart()

{

if(kbhit())

{

b = getch();

delay\_ms(50);

}

}

#int\_CCP1 // Chuong trinh ngat CCP1 o chan RC2

void interrupt\_ccp1()

{

if(input(pin\_c2)) // Neu a xung canh len

{

set\_timer1(0); // Reset timer1 ve 0

setup\_ccp1(ccp\_capture\_fe); // Chuyen CCP1 capture canh xuong

}

if(!input(pin\_c2)) // Neu la xung canh duoi

{

getVar = CCP\_1; // CCP\_1 = tra tri TMR1

//

distance = (getVar\*0.8)/58;

setup\_ccp1(ccp\_capture\_re); // Chuyen CCP1 capture canh len

hasEcho = 1; // Co xung echo tra ve

disable\_interrupts(int\_timer1); // Tat ngat Timer1

}

}

#int\_timer1 // Chuong trinh ngat timer1 tran

void interrupt\_timer1()

{

if(!hasEcho)

{

// Tran Timer 1 ma ko co xung echo thi cho qua( Xung echo toi qua 30ms, tran Timer1 52ms)

hasEcho = 1;

distance = -1;

}

disable\_interrupts(int\_timer1);

}

void Delay\_4us(unsigned int16 t)

{

while(t>0) t--;

}

float Convert(float x, float in\_min, float in\_max, float out\_min, float out\_max)

{

float tam = (out\_max - out\_min)/(in\_max - in\_min);

tam = tam \* (x - in\_min);

tam = tam + out\_min;

return tam;

}

void Write(unsigned int16 goc, signed int16 t)

{

T\_on = Convert(goc,0,180,265,1205);

t=t/20;

for(;t>0;t--)

{

output\_high(pin\_c0);

Delay\_4us(T\_on);

output\_low(pin\_c0);

Delay\_4us(C-T\_on);

}

}

void trig()

{

output\_high(trigPin);

Delay\_4us(3);

output\_low(trigPin);

}

void lcd\_display(unsigned int8 x)

{

lcd\_putc(x/100+0x30);

lcd\_putc(x/10%10+0x30);

lcd\_putc(x%10+0x30);

}

void main()

{

set\_tris\_d(0x00);

set\_tris\_c(0x84); // 0x10000100

set\_tris\_b(0x00);

// Khoi tao timer1

setup\_timer\_1(t1\_internal | t1\_div\_by\_4);

set\_timer1(0);

// Thiet lap CCP 1 bat canh len

setup\_ccp1(ccp\_capture\_fe);

// Khoi tao cho phep ngat

enable\_interrupts(global);

enable\_interrupts(int\_rda);

enable\_interrupts(int\_timer1);

enable\_interrupts(int\_ccp1);

// Khi tao LCD

lcd\_init();

output\_b(0x00);

output\_c(0x00);

T\_on = 0;

C = 10000;

distance = 0;

getVar = 0;

hasEcho=0;

b = 49;

while(true)

{

for(unsigned int8 i=0; i<181; i++)

{

Write(i,20);

lcd\_gotoxy(1,1);

lcd\_putc("GOC QUAY:");

sprintf(Angle,"%d", i);

lcd\_gotoxy(10,1);

lcd\_display(atoi(Angle));

lcd\_gotoxy(13,1);

lcd\_putc(0xDF);

delay\_ms(1);

set\_timer1(0);

enable\_interrupts(int\_timer1);

hasEcho = 0;

trig();

output\_low(pin\_b0);

while(!hasEcho);

//! lcd\_putc("\f");

if(distance >=3 && distance <= 90)

{

if(b == '1') output\_high(pin\_b0);

else output\_low(pin\_b0);

lcd\_gotoxy(1,2);

lcd\_putc("KH/ CACH:");

lcd\_gotoxy(10,2);

sprintf(Dist,"%3.0f",distance);

lcd\_display(atof(Dist));

lcd\_gotoxy(13,2);

lcd\_putc("cm");

delay\_ms(1);

printf("%3.0f\r", distance);

}

else

{

lcd\_gotoxy(1,2);

lcd\_putc(" NO OBJECT ");

printf("%d\r",0);

}

printf("%u\r", i);

}

for(unsigned int8 x=179; x > 0; x--)

{

Write(x,20);

lcd\_gotoxy(1,1);

lcd\_putc("GOC QUAY:");

sprintf(Angle,"%d",x);

lcd\_gotoxy(10,1);

lcd\_display(atoi(Angle));

lcd\_gotoxy(13,1);

lcd\_putc(0xDF);

delay\_ms(1);

set\_timer1(0);

enable\_interrupts(int\_timer1);

hasEcho = 0;

trig();

output\_low(pin\_b0);

while(!hasEcho);

//! lcd\_putc("\f");

if(distance >=3 && distance <= 90)

{

if(b == '1') output\_high(pin\_b0);

else output\_low(pin\_b0);

lcd\_gotoxy(1,2);

lcd\_putc("KH/ CACH:");

lcd\_gotoxy(10,2);

sprintf(Dist,"%3.0f",distance);

lcd\_display(atof(Dist));

lcd\_gotoxy(13,2);

lcd\_putc("cm");

delay\_ms(1);

printf("%3.0f\r", distance);

}

else

{

lcd\_gotoxy(1,2);

lcd\_putc(" NO OBJECT ");

printf("%d\r",0);

}

printf("%u\r", x);

}

}

}

Phần giao diện giao tiếp giữa người dùng và máy tính chúng em sử dụng C# để thiết kế.

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing; //Cho phép truy cập tính năng đồ họa

using System.IO.Ports; //Chứa các lớp để điều khiển cổng nối tiếp

using System.Linq;

using System.Linq.Expressions;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

using System.Xml;

namespace Group16

{

public partial class Form2 : Form

{

SerialPort UART = new SerialPort(); //Tạo một cổng com mới

Timer t = new Timer();

Graphics g;

Bitmap bmp;

Pen p;

Pen p2;

Pen p3;

int WIDTH = 900, HEIGHT = 900;

int uz = 450;

int rx, ry;

int x = 0, y = 0;

int x1 = 0, y1 = 0;

int k;

int uz2;

//Hàm quét cổng Com

private void Khoitao()

{

try

{

comboBox1.DataSource = SerialPort.GetPortNames(); //Đọc toàn bộ cổng Com của máy tính

if (comboBox1.Items.Count > 0)

{

comboBox1.SelectedIndex = 0;

}

}

catch (Exception)

{

}

}

public Form2()

{

InitializeComponent();

}

//Hàm Load\_Form2

private void Form2\_Load(object sender, EventArgs e)

{

//Cho form (hoặc control) và sử dụng các tài nguyên từ thread khác một cách thoải mái

Control.CheckForIllegalCrossThreadCalls = false;

Khoitao();

// Đại diện cho phương thức sẽ xử lý sự kiện DataReceive của một đối tượng SerialPort.

this.UART.DataReceived += new System.IO.Ports.SerialDataReceivedEventHandler(this.UART\_DataReceived);

//Nhận hoặc đặt khoảng thời gian, được biểu thị bằng mili giây, tại đó để tăng sự kiện Đã trôi qua.

t.Interval = 32;

t.Tick += new EventHandler(this.tick);

t.Enabled = true;

rx = WIDTH / 2 ;

ry = HEIGHT / 2;

pictureBox5.BackColor = Color.Black;

pictureBox6.Image = Group16.Properties.Resources.bell\_\_1\_;

//Khởi tạo một phiên bản mới của lớp Bitmap với kích thước được chỉ định.

bmp = new Bitmap(WIDTH, 500);

}

//Hàm nút nhấn Connect

private void button1\_Click\_1(object sender, EventArgs e)

{

try

{

if (comboBox1.Text.Trim().Length == 0)

{

MessageBox.Show("Select COM");

return;

}

UART.PortName = comboBox1.Text; //Hiển thị những cổng Com của máy tính

UART.Open(); //Mở kết cổng Com

if (UART.IsOpen == true) //True cho biết trạng thái mở của Cổng Com

{

button1.Enabled = false; //Tắt hiển thị nút Connect

button2.Enabled = true; //Mở hiện thị nút Disconnect

}

else

{

button1.Enabled = true; //Mở hiện thị nút Conect

button2.Enabled = false; //Tắt hiển thị nút Disconnect

}

}

catch (Exception)

{

}

}

//Hàm nút nhấn Disconnect

private void button2\_Click\_1(object sender, EventArgs e)

{

try

{

UART.Close();

if (UART.IsOpen == true)

{

button1.Enabled = false;

button2.Enabled = true;

}

else

button1.Enabled = true;

button2.Enabled = false;

}

catch (Exception)

{

}

}

//Hàm nhận dữ liệu từ Pic gửi lên

private void UART\_DataReceived(object sender, SerialDataReceivedEventArgs e)

{

try

{

txtkq1.Text = UART.ReadTo("\r"); //Đọc một chuỗi lên đến giá trị được chỉ định trong bộ đệm đầu vào.

txtkq2.Text = UART.ReadTo("\r"); //chia dữ liệu cho 2 textbox sau dấu \r

k = 90 + Convert.ToInt32(txtkq2.Text);

uz2 = Convert.ToInt32(txtkq1.Text) \* 5;

if(Convert.ToInt32(txtkq1.Text) != 0)

{

pictureBox6.Image = Group16.Properties.Resources.bell\_ring;

}

else

{

pictureBox6.Image = Group16.Properties.Resources.bell\_\_1\_;

}

}

catch (Exception)

{

}

}

//Xử lý nút nhấn chuông thông báo

bool buzzer = true;

private void button3\_Click\_1(object sender, EventArgs e)

{

pictureBox6.Image = Group16.Properties.Resources.bell\_\_1\_;

try

{

button3.Enabled = false;

button4.Enabled = true;

if (buzzer == true)

{

UART.Write("2");

buzzer = false;

pictureBox6.Image = Group16.Properties.Resources.bell\_\_1\_; //Gán hình ảnh bell\_ring cho picture\_Box

}

}

catch (Exception)

{

}

}

//Hàm Click vào button Turn On

private void button4\_Click\_1(object sender, EventArgs e)

{

pictureBox6.Image = Group16.Properties.Resources.bell\_ring;

try

{

button4.Enabled = false;

button3.Enabled = true;

if (buzzer == false)

{

UART.Write("1");

buzzer = true;

pictureBox6.Image = Group16.Properties.Resources.bell\_ring;

}

}

catch (Exception)

{

}

}

private void tick(object sender, EventArgs e)

{

p = new Pen(Color.Green, 5f);

p2 = new Pen(Color.White, 2f);

p3 = new Pen(Color.Red, 5f);

//Tạo một đồ họa hình ảnh

g = Graphics.FromImage(bmp);

x = rx + (int)(uz \* -Math.Sin(Math.PI \* k / 180));

y = ry + (int)(uz \* Math.Cos(Math.PI \* k / 180));

x1 = rx + (int)(uz2 \* -Math.Sin(Math.PI \* k / 180));

y1 = ry + (int)(uz2 \* Math.Cos(Math.PI \* k / 180));

//Gía trị các cung

g.DrawArc(p2, 0, 0, WIDTH, HEIGHT, -180, 180);

g.DrawArc(p2, 50, 48, WIDTH - 100, HEIGHT - 100, -180, 180);

g.DrawArc(p2, 100, 96, WIDTH - 200, HEIGHT - 200, -180, 180);

g.DrawArc(p2, 150, 150, WIDTH - 300, HEIGHT - 300, -180, 180);

g.DrawArc(p2, 200, 200, WIDTH - 400, HEIGHT - 400, -180, 180);

g.DrawArc(p2, 250, 250, WIDTH - 500, HEIGHT - 500, -180, 180);

g.DrawArc(p2, 300, 300, WIDTH - 600, HEIGHT - 600, -180, 180);

g.DrawArc(p2, 350, 350, WIDTH - 700, HEIGHT - 700, -180, 180);

g.DrawArc(p2, 400, 400, WIDTH - 800, HEIGHT - 800, -180, 180);

// Các đường thẳng

g.DrawLine(p, new Point(rx, ry), new Point(x, y));

g.DrawLine(p3, new Point(rx, ry), new Point(x1, y1));

g.DrawLine(p2, new Point(450, 450), new Point(0, 450));

g.DrawLine(p2, new Point(450, 450), new Point(132, 132));

g.DrawLine(p2, new Point(450, 450), new Point(450, 0));

g.DrawLine(p2, new Point(450, 450), new Point(768, 132));

g.DrawLine(p2, new Point(450, 450), new Point(900, 450));

g.DrawString("0\*", new Font("Arial", 18), Brushes.White, new Point(10, 420));

g.DrawString("45\*", new Font("Arial", 18), Brushes.White, new Point(132, 132));

g.DrawString("90\*", new Font("Arial", 18), Brushes.White, new Point(450, 0));

g.DrawString("135\*", new Font("Arial", 18), Brushes.White, new Point(768, 132));

g.DrawString("180\*", new Font("Arial", 18), Brushes.White, new Point(850, 420));

g.DrawString("90", new Font("Arial", 18), Brushes.Red, new Point(0, 450));

g.DrawString("80", new Font("Arial", 18), Brushes.Red, new Point(50, 450));

g.DrawString("70", new Font("Arial", 18), Brushes.Red, new Point(100, 450));

g.DrawString("60", new Font("Arial", 18), Brushes.Red, new Point(150, 450));

g.DrawString("50", new Font("Arial", 18), Brushes.Red, new Point(200, 450));

g.DrawString("40", new Font("Arial", 18), Brushes.Red, new Point(250, 450));

g.DrawString("30", new Font("Arial", 18), Brushes.Red, new Point(300, 450));

g.DrawString("20", new Font("Arial", 18), Brushes.Red, new Point(350, 450));

g.DrawString("10", new Font("Arial", 18), Brushes.Red, new Point(400, 450));

g.DrawString("0", new Font("Arial", 18), Brushes.Red, new Point(450, 450));

g.DrawString("90", new Font("Arial", 18), Brushes.Red, new Point(880, 450));

g.DrawString("80", new Font("Arial", 18), Brushes.Red, new Point(830, 450));

g.DrawString("70", new Font("Arial", 18), Brushes.Red, new Point(780, 450));

g.DrawString("60", new Font("Arial", 18), Brushes.Red, new Point(730, 450));

g.DrawString("50", new Font("Arial", 18), Brushes.Red, new Point(680, 450));

g.DrawString("40", new Font("Arial", 18), Brushes.Red, new Point(630, 450));

g.DrawString("30", new Font("Arial", 18), Brushes.Red, new Point(580, 450));

g.DrawString("20", new Font("Arial", 18), Brushes.Red, new Point(530, 450));

g.DrawString("10", new Font("Arial", 18), Brushes.Red, new Point(480, 450));

pictureBox5.Image = bmp;

//Giải phóng tài nguyên

g.Dispose();

p.Dispose();

p2.Dispose();

p3.Dispose();

}

}

}

**CHƯƠNG 5: THIẾT KẾ PHẦN CỨNG**