**TRƯỜNG ĐẠI HỌC NÔNG LÂM TP.HCM**

**KHOA CƠ KHÍ – CÔNG NGHỆ**

**BỘ MÔN CÔNG NGHỆ KĨ THUẬT OTO**

****

**ĐỒ ÁN**

**LẬP TRÌNH ĐIỀU KHIỂN Ô TÔ**

**ĐỀ TÀI**

**Công tắc vị trí tay số điều khiển hệ thống khởi động**

Sinh viên thực hiện: Lê Duy Quốc MSSV: 18154104

Nguyễn Văn Toàn MSSV: 18154125

Võ Văn Vũ MSSV: 18154135

Giáo viên hướng dẫn: Nguyễn Trịnh Nguyên

Ký tên:

Ngày hoàn thành:

***Tp.HCM, tháng 01/2021***

## Kiến trúc vi điều khiển:

AVR là họ vi điều khiển 8 bit theo công nghệ mới, với những tính năng rất mạnh được tích hợp trong chip của hãng Atmel theo công nghệ RISC, nó mạnh ngang hàng với các họ vi điều khiển 8 bit khác như PIC, Pisoc.Do ra đời muộn hơn nên họ vi điều khiển AVR có nhiều tính năng mới đáp ứng tối đa nhu cầu của người sử dụng, so với họ 8051 89xx sẽ có độ ổn định, khả năng tích hợp, sự mềm dẻo trong việc lập trình và rất tiện lợi.

\* Tính năng mới của họ AVR:

* + Giao diện SPI đồng bộ.
  + Các đường dẫn vào/ra (I/O) lập trình được.
  + Giao tiếp I2C.
  + Bộ biến đổi ADC 10 bit.
  + Các kênh băm xung PWM.
  + Các chế độ tiết kiệm năng lượng như sleep, stand by..vv.
  + Một bộ định thời Watchdog.
  + 3 bộ Timer/Counter 8 bit.
  + 1 bộ Timer/Counter 16 bit.
  + 1 bộ so sánh analog.
  + Bộ nhớ EEPROM.
  + Giao tiếp USART..vv.

## Giới thiệu vi điều khiển Atmega16L:

Atmelga16L có đầy đủ tính năng của họ AVR, về giá thành so với các loại khác thì giá thành là vừa phải khi nghiên cứu và làm các công việc ứng dụng tới vi điều khiển. Tính năng**:**

* + Bộ nhớ 16K(flash) . - 512 byte (EEPROM). - 1 K (SRAM).
  + Đóng vỏ 40 chân , trong đó có 32 chân vào ra dữ liệu chia làm 4 PORT A,B,C,D. Các chân này đều có chế độ pull\_up resistors.
  + Giao tiếp SPI. - Giao diện I2C. - Có 8 kênh ADC 10 bit.
  + 1 bộ so sánh analog. - 4 kênh PWM.
  + 2 bộ timer/counter 8 bit, 1 bộ timer/counter1 16 bit.
  + 1 bộ định thời Watchdog.
  + 1 bộ truyền nhận UART lập trình được.

Mô tả các chân:

* Vcc và GND 2 chân cấp nguồn cho vi điều khiển hoạt động.
* Reset đây là chân reset cứng khởi động lại mọi hoạt động của hệ thống.
* 2 chân XTAL1, XTAL2 các chân tạo bộ dao động ngoài cho vi điều khiển, các chân này được nối với thạch anh (hay sử dụng loại 4M), tụ gốm (22p).
* Chân Vref thường nối lên 5v(Vcc), nhưng khi sử dụng bộ ADC thì chân này được sử dụng làm điện thế so sánh, khi đó chân này phải cấp cho nó điện áp cố định, có thể sử dụng diode zener

- Chân Avcc thường được nối lên Vcc nhưng khi sử dụng bộ ADC thì chân này được nối qua 1 cuộn cảm lên Vcc với mục đích ổn định điện áp cho bộ biến đổi.

## Phần mềm lập trình CodeVisionAVR V2.05.0:

Lựa chọn phần mềm : đây là phần mềm được sử dụng rất rộng dải bởi nó được xây dựng trên nền ngôn ngữ lập trình C, phần mềm được viết chuyên nghiệp hướng tới người sử dụng bởi sự đơn giản, sự hổ trợ cao các thư viện có sẵn.

## Chương trình CODE viết bằng CodeVisionAVR:

This program was produced by the

CodeWizardAVR V2.05.0 Professional

Automatic Program Generator

© Copyright 1998-2010 Pavel Haiduc, HP InfoTech s.r.l.

http://www.hpinfotech.com

Project :

Version :

Date : 28/12/2020

Author : NeVaDa

Company :

Comments:

Chip type : ATmega16

Program type : Application

AVR Core Clock frequency: 8.000000 MHz

Memory model : Small

External RAM size : 0

Data Stack size : 256

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

#include <mega16.h>

#include <delay.h>

// Alphanumeric LCD Module functions

#include <alcd.h>

// Declare your global variables here

void main(void)

{

// Declare your local variables here

// Input/Output Ports initialization

// Port A initialization

// Func7=In Func6=In Func5=In Func4=In Func3=In Func2=In Func1=In Func0=In

// State7=P State6=P State5=P State4=P State3=P State2=P State1=P State0=P

PORTA=0xFF;

DDRA=0x00;

// Port B initialization

// Func7=In Func6=In Func5=In Func4=In Func3=In Func2=In Func1=In Func0=In

// State7=T State6=T State5=T State4=T State3=T State2=T State1=T State0=T

PORTB=0x00;

DDRB=0x00;

// Port C initialization

// Func7=In Func6=In Func5=In Func4=In Func3=In Func2=In Func1=In Func0=In

// State7=T State6=T State5=T State4=T State3=T State2=T State1=T State0=T

PORTC=0x00;

DDRC=0x00;

// Port D initialization

// Func7=Out Func6=Out Func5=Out Func4=Out Func3=Out Func2=Out Func1=Out Func0=Out

// State7=0 State6=0 State5=0 State4=0 State3=0 State2=0 State1=0 State0=0

PORTD=0x00;

DDRD=0xFF;

// Timer/Counter 0 initialization

// Clock source: System Clock

// Clock value: Timer 0 Stopped

// Mode: Normal top=0xFF

// OC0 output: Disconnected

TCCR0=0x00;

TCNT0=0x00;

OCR0=0x00;

// Timer/Counter 1 initialization

// Clock source: System Clock

// Clock value: Timer1 Stopped

// Mode: Normal top=0xFFFF

// OC1A output: Discon.

// OC1B output: Discon.

// Noise Canceler: Off

// Input Capture on Falling Edge

// Timer1 Overflow Interrupt: Off

// Input Capture Interrupt: Off

// Compare A Match Interrupt: Off

// Compare B Match Interrupt: Off

TCCR1A=0x00;

TCCR1B=0x00;

TCNT1H=0x00;

TCNT1L=0x00;

ICR1H=0x00;

ICR1L=0x00;

OCR1AH=0x00;

OCR1AL=0x00;

OCR1BH=0x00;

OCR1BL=0x00;

// Timer/Counter 2 initialization

// Clock source: System Clock

// Clock value: Timer2 Stopped

// Mode: Normal top=0xFF

// OC2 output: Disconnected

ASSR=0x00;

TCCR2=0x00;

TCNT2=0x00;

OCR2=0x00;

// External Interrupt(s) initialization

// INT0: Off

// INT1: Off

// INT2: Off

MCUCR=0x00;

MCUCSR=0x00;

// Timer(s)/Counter(s) Interrupt(s) initialization

TIMSK=0x00;

// USART initialization

// USART disabled

UCSRB=0x00;

// Analog Comparator initialization

// Analog Comparator: Off

// Analog Comparator Input Capture by Timer/Counter 1: Off

ACSR=0x80;

SFIOR=0x00;

// ADC initialization

// ADC disabled

ADCSRA=0x00;

// SPI initialization

// SPI disabled

SPCR=0x00;

// TWI initialization

// TWI disabled

TWCR=0x00;

// Alphanumeric LCD initialization

// Connections specified in the

// Project|Configure|C Compiler|Libraries|Alphanumeric LCD menu:

// RS - PORTB Bit 0

// RD - PORTB Bit 1

// EN - PORTB Bit 2

// D4 - PORTB Bit 4

// D5 - PORTB Bit 5

// D6 - PORTB Bit 6

// D7 - PORTB Bit 7

// Characters/line: 16

lcd\_init(16);

while (1)

{

if (PINA.0==0) {

if (PINA.7==0) {

PORTD.0=1;

lcd\_clear();

lcd\_gotoxy(3,1);

lcd\_puts("Starting....");

delay\_ms(500);

}

else {

PORTD.0=0;

};

lcd\_clear();

lcd\_gotoxy(8,0);

lcd\_putchar('P');

lcd\_gotoxy(3,1);

lcd\_puts("Ready !!!");

delay\_ms(200);

}

else {

PORTD.0=0;

lcd\_clear();

};

if (PINA.1==0) {

if (PINA.7==0) {

lcd\_clear();

lcd\_gotoxy(3,0);

lcd\_puts("WARNING!!!");

lcd\_gotoxy(2,1);

lcd\_puts("Not allowed");

delay\_ms(500);

}

else {

lcd\_clear();

};

lcd\_clear();

lcd\_gotoxy(8,0);

lcd\_putchar('R');

delay\_ms(200);

}

else {

lcd\_clear();

};

if (PINA.2==0) {

if (PINA.7==0) {

PORTD.0=1;

lcd\_clear();

lcd\_gotoxy(3,1);

lcd\_puts("Starting....");

delay\_ms(500);

}

else {

PORTD.0=0;

};

lcd\_clear();

lcd\_gotoxy(8,0);

lcd\_putchar('N');

lcd\_gotoxy(2,1);

lcd\_puts("Ready !!!");

delay\_ms(200);

}

else {

PORTD.0=0;

lcd\_clear();

};

if (PINA.3==0) {

if (PINA.7==0) {

lcd\_clear();

lcd\_gotoxy(3,0);

lcd\_puts("WARNING!!!");

lcd\_gotoxy(2,1);

lcd\_puts("Not allowed");

delay\_ms(500);

}

else {

lcd\_clear();

};

lcd\_clear();

lcd\_gotoxy(8,0);

lcd\_putchar('1');

delay\_ms(200);

}

else {

lcd\_clear();

};

if (PINA.4==0) {

if (PINA.7==0) {

lcd\_clear();

lcd\_gotoxy(3,0);

lcd\_puts("WARNING!!!");

lcd\_gotoxy(2,1);

lcd\_puts("Not allowed");

delay\_ms(500);

}

else {

lcd\_clear();

};

lcd\_clear();

lcd\_gotoxy(8,0);

lcd\_putchar('D');

delay\_ms(200);

}

else {

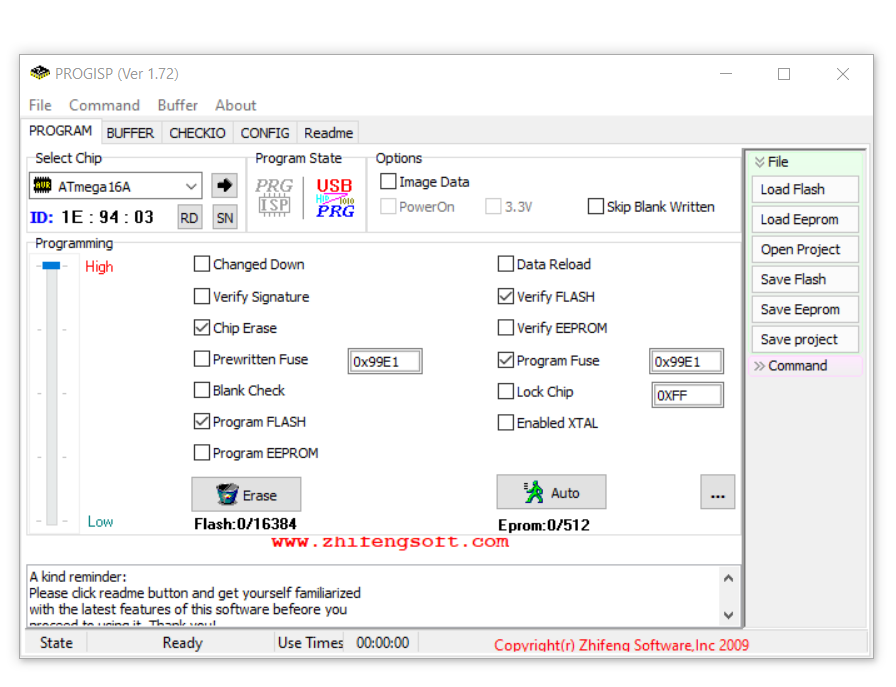
lcd\_clear();

};

}

}

## Phần mềm nạp PROGISP 1.72 :



## Mô tả:

* LCD được nối với PORT B có nhiệm vụ hiển thị vị trí tay số hiện tại và hiện thị trạng thái hay cảnh báo cho người dùng
* 5 con LED được nối với PORT D từ chân 1 tới chân 5 có nhiệm vụ thông báo vị trí tay số hiện tại
* Công tắc xoay 5 chế độ (thay thế cảm biến công tắc tay số) được nối với PORT A từ chân 0 tới chân 4 theo kiểu nối mass chung và có điện trở nội kéo lên có nhiệm vụ truyền tín hiệu cho chip ATmega16L
* Nút nhấn nhả (thay thế cho công tắc đề) đucợ nối với PORT A7 có điện trở nội kéo lên có nhiệm vụ truyền tín hiệu cho chip ATmega16L
* Tín hiệu từ công tắc xoay và nút nhấn nhả được gửi đến chip ATmega16L để quyết định có cho phép motor đề quay hay không, đồng thời cũng quyết định thông tin hiển thị trên LCD
* Motor điện 3VDC (thay thế motor đề ) được nối với PORT D0 sẽ nhận tín hiệu từ chip ATmega16L gửi tới để chấp hành nhiệm vụ.
* Nếu sử dụng motor đề thật thì cần phải có điện áp 12V để kích đề, nhưng vì điện áp ngõ ra của ATmega16L chỉ ở mức 5V nên chúng ta phải sử dụng mạch đệm ngoài để có thể dùng điện áp của acquy kích cho motor đề hoạt động

## Hoạt động:

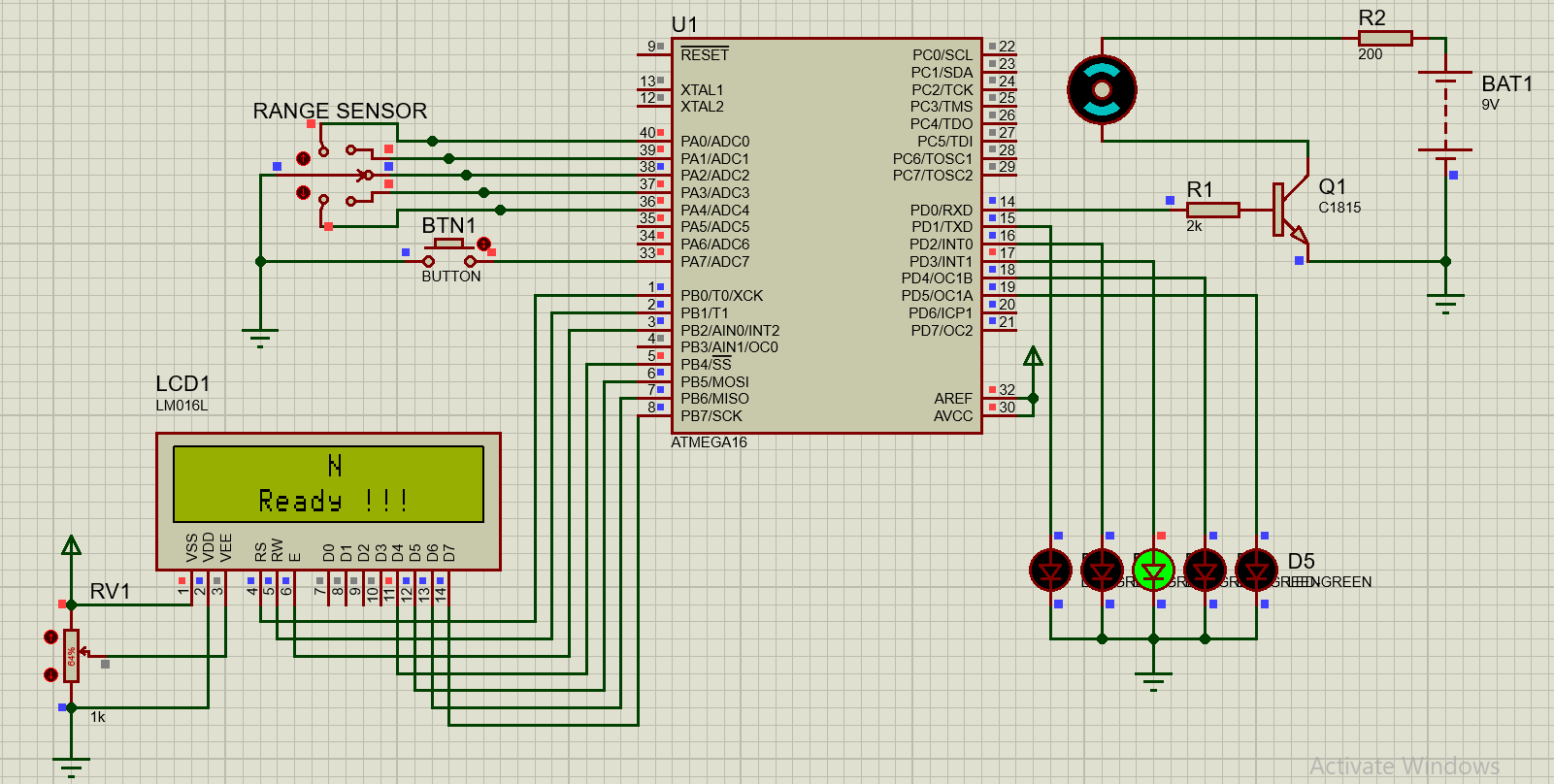
Khi công tắc tay số ở trí P hoặc N thì LCD và Led sẽ hiển thi vị trí tay số hiện tại.

* Khi ấn nút đề: motor đề quay và LCD thông báo “Starting…”
* Khi buông nút đề: motor đề dừng và LCD thông báo “Vi trí \_ Ready!!!”

Khi công tắc tay số ở vị trí R, 1 hoặc D LCD và Led sẽ hiển thị vị trí tay số hiện tại

* Khi ấn nút đề: motor đề không quay và LCD cảnh báo “WARNING\_Not allowed”
* Khi buông nút đề: và LCD thông báo “Vi trí”.

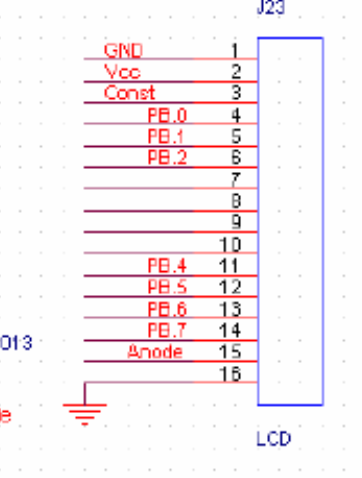
## Mô phỏng bằng phần mềm Proteus 8.8



## Các thiết bị, linh kiện chính để lắp sản phẩm

Ngoài vi điều khiển ATMEGA 16L

### LCD Module 1602A-1 xanh lá

LCD đảm nhân vai trò hiển thị các thông tin mà chúng ta muốn nhập vào hay các thông tin xử lý mà bộ điều khiển đang hoạt động đựoc hiển thị ra màn hình, giúp chúng ta giao tiếp gần hơn với quá trình hoạt đông của hệ thống.

Nó có 16 chân như hình vẽ. Trong đó chúng ta có thể thấy

* 2 chân 1,2 được cấp nguồn cho LCD hoạt động,
* chân thứ 3 (chân VSS) được nối vào đầu ra của biến trở dùng để điều chỉnh độ tương phản (phải điều chỉnh VSS hợp lý thì LCD mới hiển thị được)
* Chân 4 🡪14 là các chân điều khiển được nối với vi điều khiển

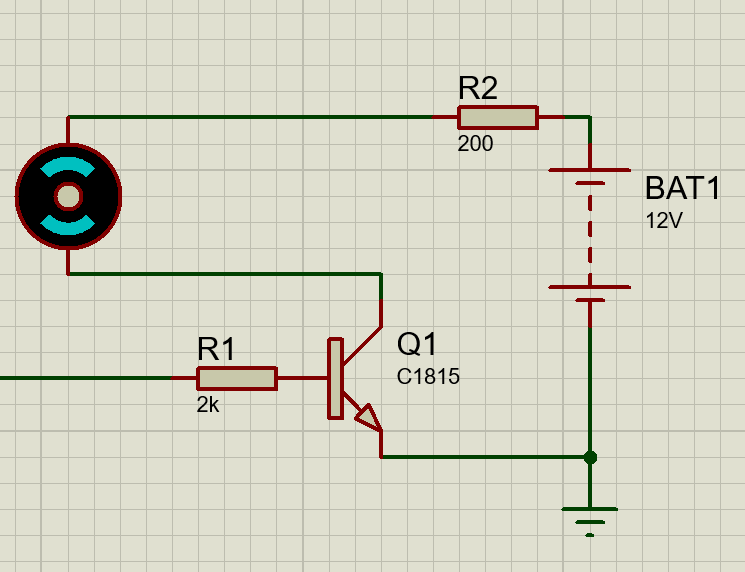
Chân 4,5,6 là để điều khiển hoạt động của LCD,

* các chân còn lại là 8 bit Data dùng để truyền nhận dữ liệu.
* Chân 15,16 là 2 chân cấp nguồn dung để bật đèn của LCD từ

### Motor điện 130 3VDC

Giả lập motor đề:

Mạch đệm khi sử dụng motor đề 12V



### Công Tắc Xoay 5 Chế Độ 2 Cực CT-5R2

Giả lập công tắc tay số với 5 chế độ tương ứng 5 cấp số là P – R – N – 1 – D

## Hình ảnh sản phẩm thực tế

