**TRƯỜNG ĐẠI HỌC VINH**

**VIỆN KỸ THUẬT CÔNG NGHỆ**

**=====\*\*\*=====**



**BÁO CÁO THỰC HÀNH**

MÔN:

*THIẾT KẾ HỆ THỐNG NHÚNG*

Giáo viên: Th.S NCS Lê Văn Chương

Họ & tên: Dương Đình Nghĩa

Ngày sinh: 26/08/1999

MSSV: 1755252021600008

Lớp: 58K – Kỹ thuật ĐK & TĐH

Nghệ An, 2021

Câu 1. Sử dụng vi điều khiển AT89C52, mô phỏng trên phần mềm Proteus và lập trình thực hiện các nhiệm vụ sau:  
*1. Hiển thị số 00 lên 2 LED 7 thanh nối vào cổng P2 theo phương pháp quét LED;*

* *Code chương trình:*

//Duong Dinh Nghia MSSV: 17552520216

#include <REGX52.H>

char code code7seg[] = {0x40,0x79,0x24,0x30,0x19,0x12,0x02,0x78,0x00,0x10,0x89,0x06,0xC7,0x40};

#define led1 P2\_0

#define led2 P2\_1

#define sang 0

#define tat 1

char i;

void delay(int time){

while(time--);

}

void main(){

led1 = led2 = tat;

while (1){

led1 = sang;

P0 = code7seg[0];

delay(100);

led1 = tat;

led2 = sang;

P0 = code7seg[0];

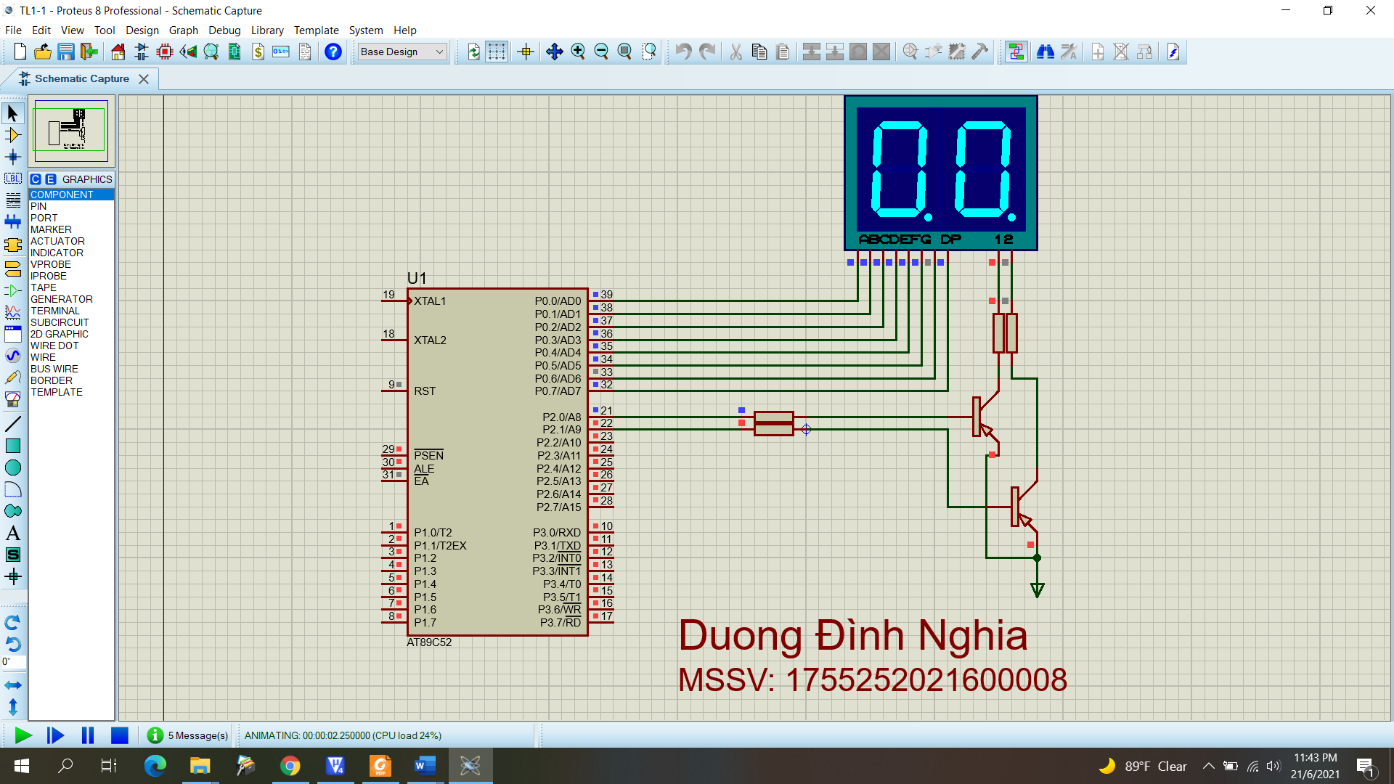
delay(100);

led2 = tat;

}

}

* Mô phỏng trên phần mềm Proteus:



*2. Tăng số đếm sau mỗi 500ms, nếu số đếm bằng “SBD+20” thì dừng lại (sử dụng timer để định thời gian).*

* *Code chương trình:*

//Duong Dinh Nghia MSSV:1755252021600008

#include <REGX52.H>

char code code7seg[] = {0x40,0x79,0x24,0x30,0x19,0x12,0x02,0x78,0x00,0x10,0x89,0x06,0xC7,0x40};

char i;

int dem;

unsigned char Chuc, Donvi;

#define Led1 P2\_0

#define Led2 P2\_1

#define sang 0

#define tat 1

void delay(int time)

{

while(time--);

}

void main()

{

Led1 = Led2 = tat;

while (1){

for (dem=0;dem<=47;dem++)

{

Chuc = dem/10;

Donvi = dem%10;

for (i=0; i<=10; i++)

{

Led1 = sang;

P0 = code7seg[Chuc];

delay(1000);

Led1 = tat;

Led2 = sang;

P0 = code7seg[Donvi];

delay(1000);

Led2 = tat;

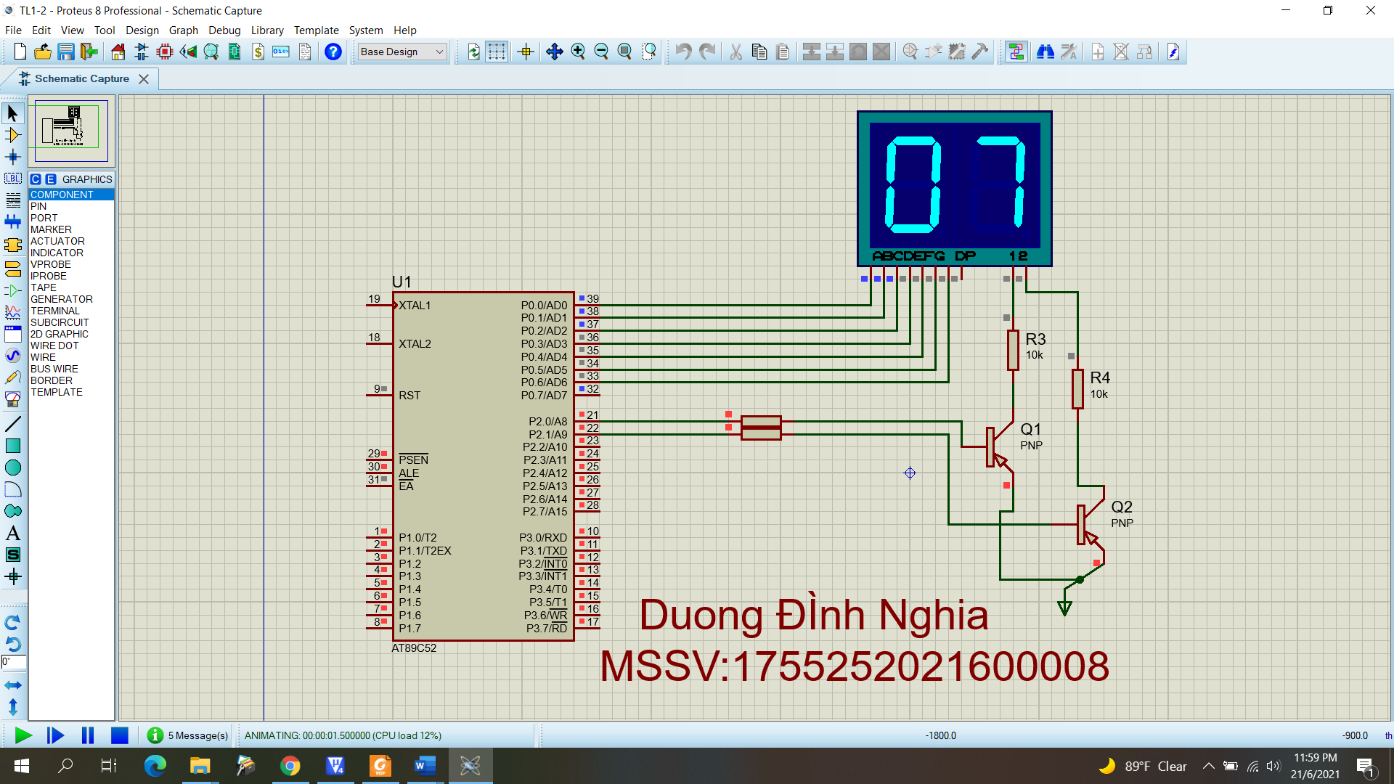
}

}

}

}

* *Mô phỏng trên Proteus:*



Câu 2. Sử dụng vi điều khiển AT89C52, mô phỏng trên phần mềm Proteus và lập trình thực hiện các nhiệm vụ sau:  
*1. Cấu hình ngắt ngoài INT0 ở chế độ ngắt sườn xuống;*

* *Code chương trình:*

//Duong Dinh Nghia MSSV:1755252021600008

#include <REGX52.H>

void delay\_ms(int ms)

{

while(ms--)

{

TH0 = 0xFC;

TL0 = 0x18;

TR0 = 1;

while(!TF0);

TF0 = 0;

TR0 = 0;

}

}

void main()

{

EX0 = 1;

IT0 = 1;

EA = 1;

while(1)

{

P2 = 0;

delay\_ms(1000);

P2 = 0xFF;

delay\_ms(1000);

}

}

void ngat() interrupt 0

{

long a = 50000;

P1\_3 = 0;

while(a--)

{

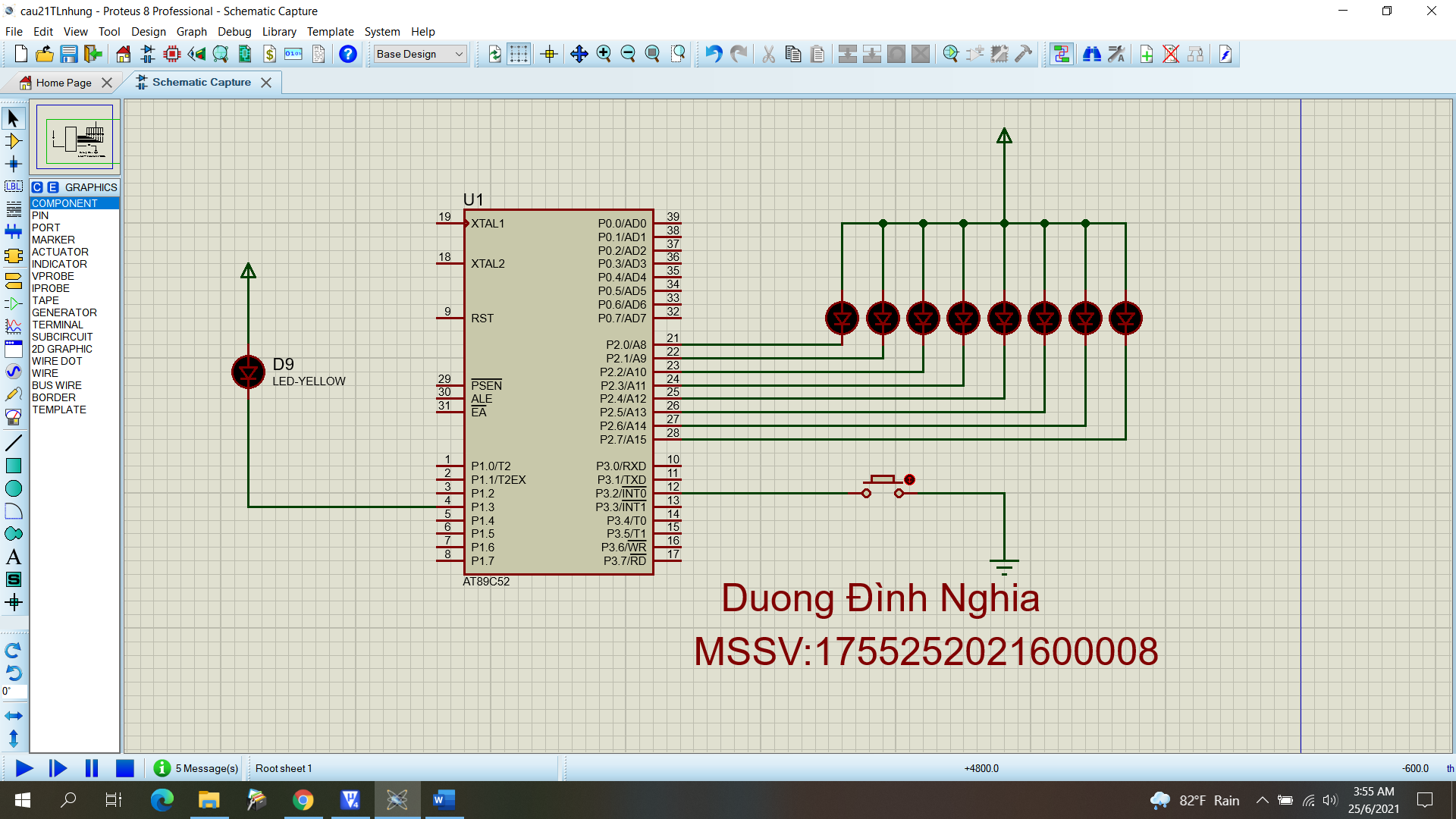
;

}

P1\_3 = 1;

}

* *Mô phỏng trên Proteus:*



*2. Đếm số lần nút bấm nút nối vào chân INT0 được bấm, hiển thị kết quả lên 2 LED 7 thanh nối vào cổng P2 theo phương pháp quét LED (nếu số lần bấm bằng “SBD+10” thì quay về 0).*

* *Code chương trình:*

//Duong Dinh Nghia MSSV:1755252021600008

#include <regx52.h>

#define LED1 P2\_0

#define LED2 P2\_1

#define sang 0

#define tat 1

sbit UP = P3^2;

//khai bao mang led 7 thanh

unsigned char code code7seg[] = {0xC0,0xF9,0xA4,0xB0,0x99,0x92,0x82,0xF8,0x80,0x90 };

char count;

unsigned char chuc, donvi;

void delay\_ms (int time)

{

while(time--)

{

TMOD = 0x01;

TH0 = 0xFC;

TL0 = 0x18;

TR0 = 1;

while(!TF0);

TF0 = 0;

TR0 = 0;

}

}

void tang() interrupt 0

{

count++;

if(count>37) count=0;//SBD:27+10=37 kiem tra neu count lon hon 37 thi gan count=0

}

void main()

{

EA = 1;//cho phep ngat toan cuc

EX0 = 1;//cho phep ngat ngoài 0

IT0 = 1; //cai dat ngat kich phat suon cho ngat ngoai 0

while(1)

{

chuc = count/10; //chia cho 10 lay phan nguyen

donvi = count%10; //chia cho 10 lay phan du

LED1 = sang;

P0 = code7seg[chuc];

delay\_ms(10);

LED1 = tat;

LED2 = sang;

P0 = code7seg[donvi];

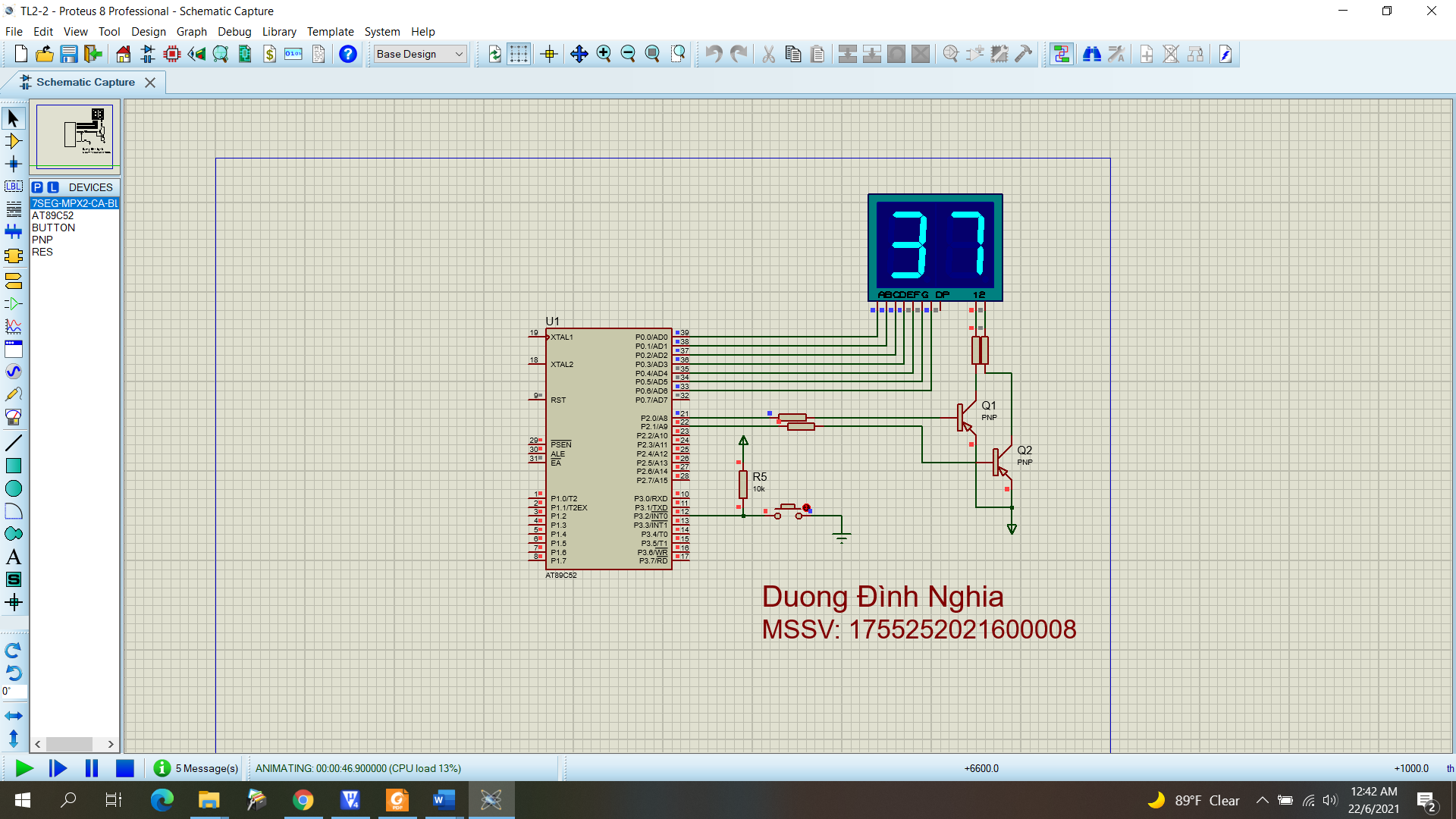
delay\_ms(10);

LED2 = tat;

}

}

* *Mô phỏng trên Proteus:*



Câu 3. Sử dụng vi điều khiển AT89C52, thực hiện các nhiệm vụ sau:  
*1. Vẽ sơ đồ mô phỏng trên Proteus ghép nối với LCD theo chế độ 4bit, hiển thị họ tên, mã số sinh viên lên LCD.*

* *Code chương trình:*

#include <REGX52.H>

//Khai bao chan giao tiep

#define LCD\_RS P2\_0

#define LCD\_RW P2\_1

#define LCD\_EN P2\_2

#define LCD\_D4 P2\_4

#define LCD\_D5 P2\_5

#define LCD\_D6 P2\_6

#define LCD\_D7 P2\_7

void delay\_us(unsigned int t){

unsigned int i;

for(i=0;i<t;i++);

}

void delay\_ms(unsigned int t){

unsigned int i,j;

for(i=0;i<t;i++)

for(j=0;j<125;j++);

}

//Chuong trinh giao tiep LCD 16x2 4bit

void LCD\_Enable(void){

LCD\_EN =1;

delay\_us(3);

LCD\_EN=0;

delay\_us(50);

}

//Ham Gui 4 Bit Du Lieu Ra LCD

void LCD\_Send4Bit(unsigned char Data){

LCD\_D4=Data & 0x01;

LCD\_D5=(Data>>1)&1;

LCD\_D6=(Data>>2)&1;

LCD\_D7=(Data>>3)&1;

}

// Ham Gui 1 Lenh Cho LCD

void LCD\_SendCommand(unsigned char command){

LCD\_Send4Bit(command >>4);// Gui 4 bit cao

LCD\_Enable();

LCD\_Send4Bit(command); //Gui 4 bit thap

LCD\_Enable();

}

// Ham Khoi Tao LCD

void LCD\_Init(){

LCD\_Send4Bit(0x00);

delay\_ms(20);

LCD\_RS=0;

LCD\_RW=0;

LCD\_Send4Bit(0x03);

LCD\_Enable();

delay\_ms(5);

LCD\_Enable();

delay\_us(100);

LCD\_Enable();

LCD\_Send4Bit(0x02);

LCD\_Enable();

LCD\_SendCommand( 0x28 ); // giao thuc 4 bit, hien thi 2 hang, ki tu 5x8

LCD\_SendCommand( 0x0c); // cho phep hien thi man hinh

LCD\_SendCommand( 0x06 ); // tang ID, khong dich khung hinh

LCD\_SendCommand(0x01); // xoa toan bo khung hinh

}

void LCD\_Gotoxy(unsigned char x, unsigned char y){

unsigned char address;

if(!y)address=(0x80+x);

else address=(0xc0+x);

delay\_us(1000);

LCD\_SendCommand(address);

delay\_us(50);

}

void LCD\_PutChar(unsigned char Data){//Ham Gui 1 Ki Tu

LCD\_RS=1;

LCD\_SendCommand(Data);

LCD\_RS=0 ;

}

void LCD\_Puts (char \*s){//Ham gui 1 chuoi ky tu

while (\*s){

LCD\_PutChar(\*s);

s++;

}

}

void main(){

LCD\_Init();//Khoi tao LCD

LCD\_Gotoxy(0,0);//Tro toi vi tri

LCD\_Puts("Duong Dinh Nghia");

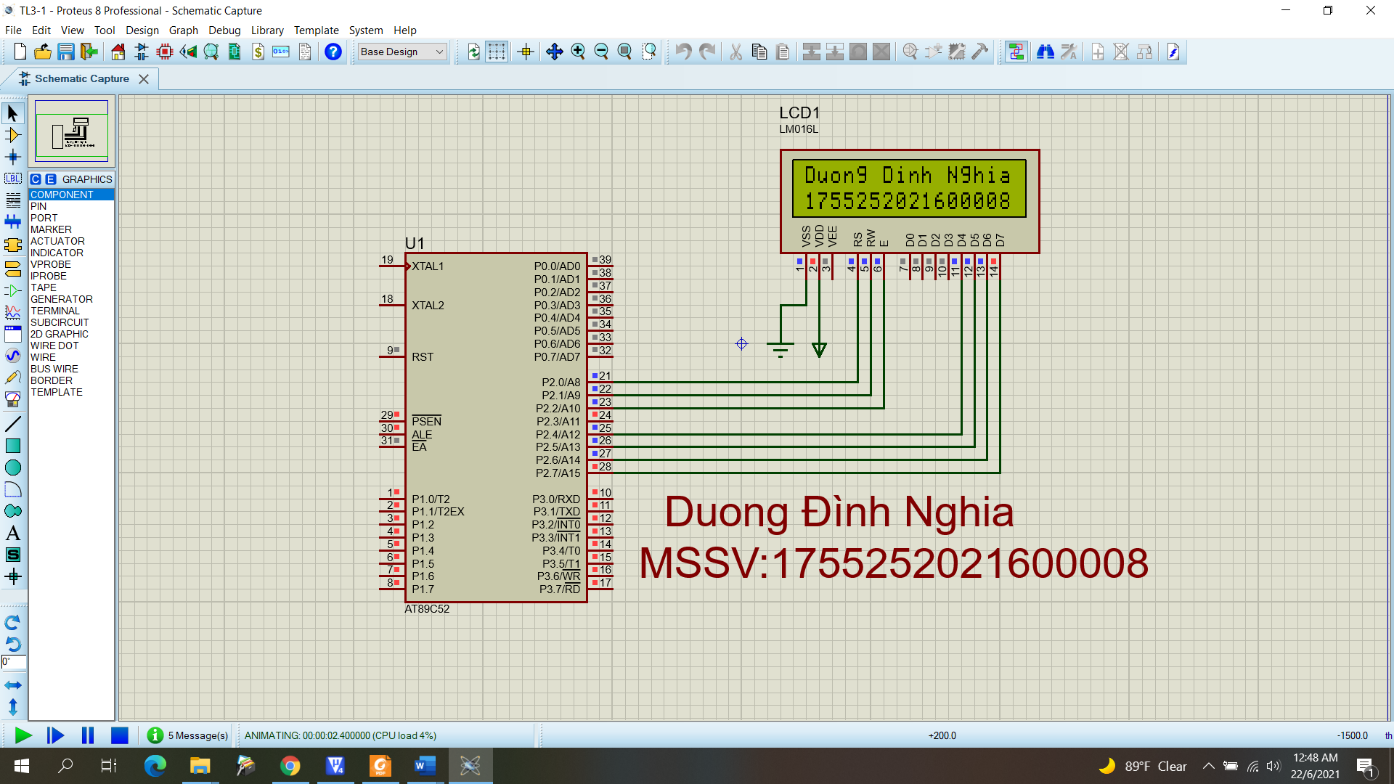
LCD\_Gotoxy(0,1);//Tro toi vi tri

LCD\_Puts("1755252021600008");

while(1);

}

* *Mô phỏng trên Proteus:*



*2. Vẽ sơ đồ mô phỏng trên Proteus, lập trình hiển thị “Họ tên và mã số sinh viên” qua chuẩn truyền thông UART;*

* *Code chương trình:*

//Duong Dinh Nghia MSSV:1755252021600008

#include <REGX52.H>

sbit rs = P2^0;

sbit rw = P2^1;

sbit e = P2^2;

void delay(unsigned int t)

{

unsigned int i, j;

e = 1;

for(i = 0; i < t; i++)

for(j = 0; j < 1275; j++);

e = 0;

}

void cmd1(unsigned char ch)

{

P2 = ch;

rs = 0;

rw = 0;

delay(10);

}

void dat1(unsigned char ch)

{

P2 = ch;

rs = 1;

rw = 0;

delay(10);

}

void cmd(unsigned char a)

{

unsigned char x;

x = a & 0xF0;

cmd1(x);

x = (a << 4) & 0xF0;

cmd1(x);

}

void dat(unsigned char a)

{

unsigned char x;

x = a & 0xF0;

dat1(x);

x = (a << 4) & 0xF0;

dat1(x);

}

void main(void)

{

unsigned char mybyte;

cmd(0x28);

cmd(0x01);

cmd(0x0C);

cmd(0x80);

cmd(0x06);

TMOD = 0x20;

TH1 = 0xFD;

SCON = 0x50;

TR1 = 1;

while(1)

{

while(RI == 0);

mybyte = SBUF;

dat(mybyte);

RI = 0;

if(mybyte == 0x08)

cmd(0x01);

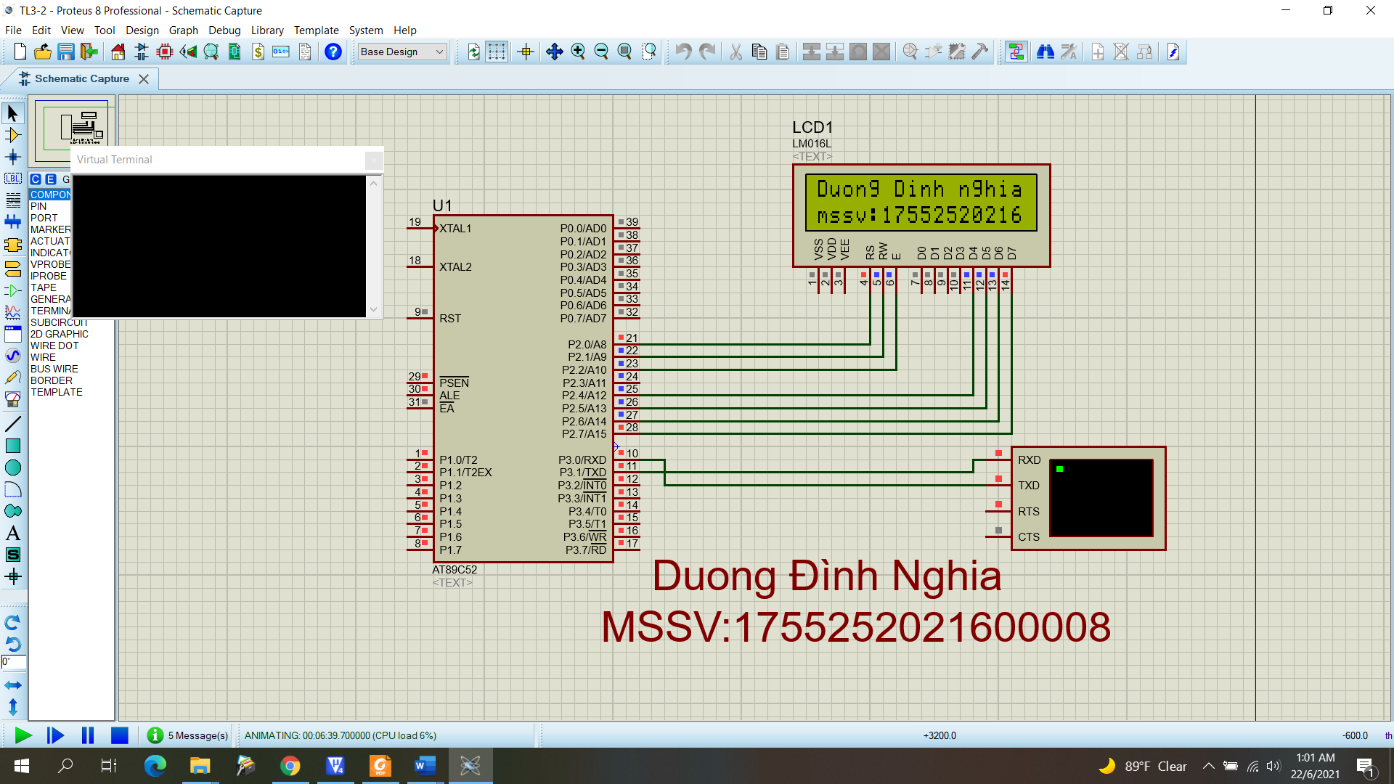
if(mybyte == 0x0D)

cmd(0xC0);

}

}

* *Mô phỏng trên Proteus:*



Câu 4. Sử dụng vi điều khiển AT89C52, vẽ sơ đồ mô phỏng trên Proteus ghép nối với Led D1 qua cổng P1.2, BUTTON B1 qua cổng P1.3. Sử dụng hệ điều hành RTX51 lập trình ngắt USART, tast BUTTON, tast LED. Thực hiện gửi “signal” từ ngắt USART và task BUTTON đến tast LED. Task LED thực hiện đảo trạng thái của Led D1 khi nhận được tín hiệu task khác gửi tới.

* *Code chương trình:*

//Duong Dinh Nghia MSSV:1755252021600008

#include <rtx51tny.h>

#include <REGX51.H>

#include <AT89X51.H>

#include <string.h>

#define INIT 0

#define DO 1

#define BUTT 2

sbit LED = P1^0;

sbit BUTTON = P1^1;

void USART(void) interrupt 4

{

if(RI)

{

RI = 0;

isr\_send\_signal(DO);

}

}

void Startup(void) \_task\_ INIT

{

SCON = 0x52;

TMOD = 0x21;

TH1 = TL1 = -3;

TR1= 1;

IE = 0x90;

os\_create\_task(DO);

os\_create\_task(BUTT);

os\_delete\_task(INIT);

}

void LED\_DO(void) \_task\_ DO

{

while(1)

{

os\_wait2(K\_SIG, 50);

LED ^= 1;

}

}

void Button(void) \_task\_ BUTT

{

while(1)

{

if (BUTTON == 0)

{

os\_send\_signal(DO);

while(BUTTON == 0);

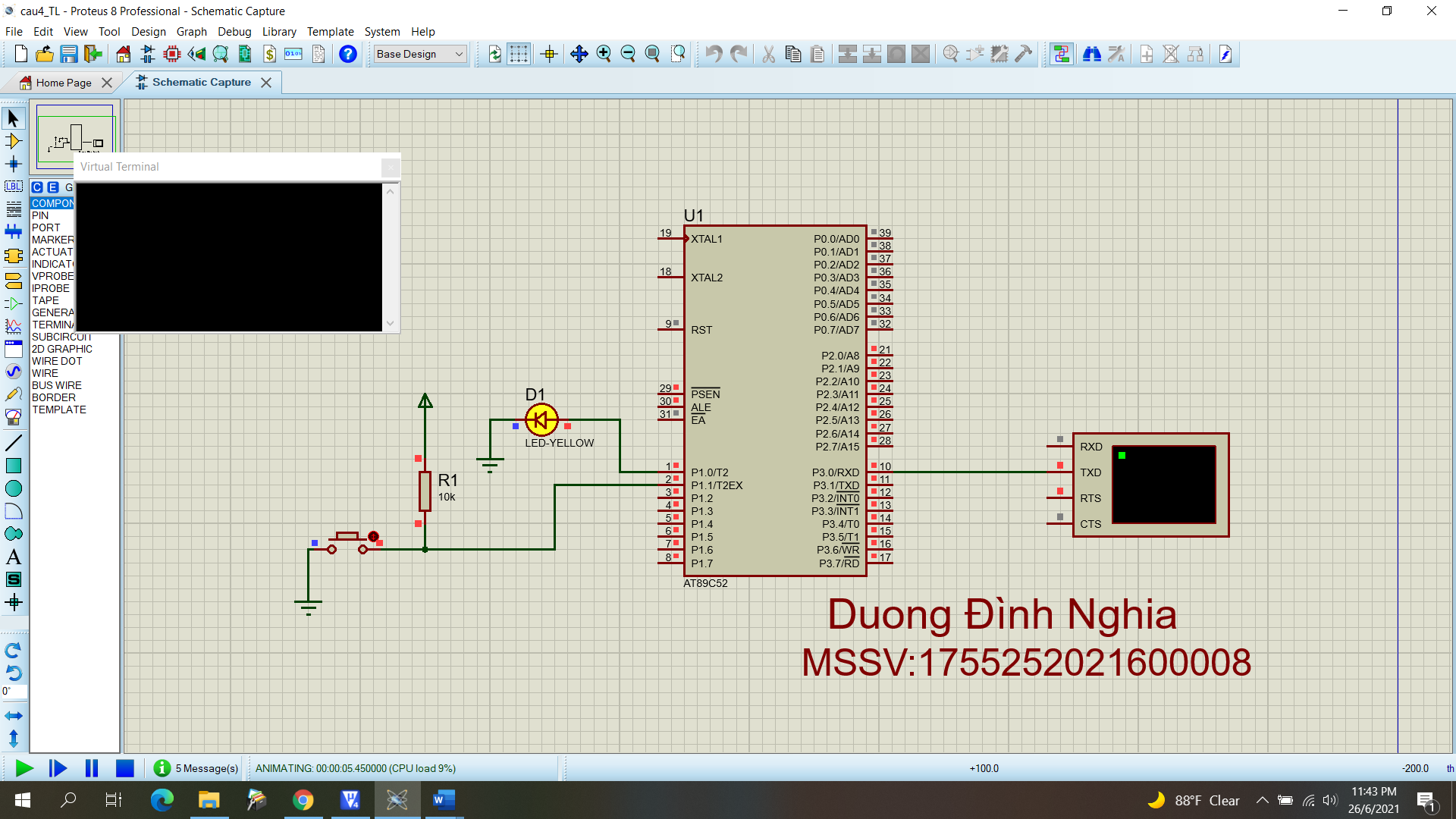
}

os\_wait2(K\_TMO, 10);

}

}

* *Mô phỏng trên Protues:*



Câu 5. Hãy trình bày:  
*1. Trình bày quy trình thiết kế hệ thống nhúng sử dụng vi điều khiển.*

7 giai đoạn của quá trình thiết kế:

* Lập mô tả các tính năng yêu cầu đối với sản phẩm.
* Phân chia thiết kế thành các mảng thiết kế phần cứng và phần mềm.
* Bước đầu thiết kế phần cứng và phần mềm, thực hiện phân chia lại nhiệm vụ giữa phần cứng và phần mềm nếu cần thiết.
* Thiết kế phần cứng và phần mềm được tiến hành song song.
* Cài đặt phần mềm lên phần cứng và tiến hành các thử nghiệm.
* Thử nghiệm hoàn chỉnh các tính năng của sản phẩm theo yêu cầu và đưa sản phẩm ra thị trường.
* Tiếp tục phát triển và nâng cấp sản phẩm sau khi đã đưa ra thị trường.

*2. Hệ điều hành thời gian thực (RTOS). Ưu điểm, nhược điểm và ứng dụng của hệ điều hành thời gian thực trong thiết kế các hệ thống nhúng.*

Hệ điều hành thời gian thực – RealTime Operating Systems(RTOS), là phần mềm điều khiển chuyên dụng thường được dùng trong những ứng dụng điện toán nhúng có tài nguyên bộ nhớ hạn chế và yêu cầu ngặt nghèo về thời gian đáp ứng tức thời, tính sẵn sàng cao và khả năng tự kiểm soát một cách chính xác.

Ưu điểm:

- Điều khiển dễ dàng hơn nhờ dùng độ ưu tiên: trong đó các chu trình xử lý thiết bị thì có cùng độ ưu tiên và bộ xử lý có thể dừng vòng lặp chính để thực thi các ngắt với độ ưu tiên cao.

- Độ ưu tiên của các ngắt cao nên cho phép dừng tác vụ đang thực hiện và yêu cầu chu trình thực hiện ngắt (nên) ngắn cho phù hợp với các thiết bị đơn giản có không gian nhớ nhỏ.

Nhược điểm:

- Phức tạp hơn vòng lặp Round-robin

- Vấn đề lưu trữ và khôi phục dữ liệu bên trong chu trình ngắt trở nên cần thiết khi mà số lượng thanh ghi là nhỏ

- Vấn đề chia sẻ dữ liệu: do quá trình gỡ lỗi là phức tạp. Vấn đề về độ trễ vẫn còn tồn tại (thực thi một tác vụ vẫn phải chờ vòng lặp duyệt qua các trường hợp để tìm đến tác vụ thích hợp).

Ứng dụng:

Hệ điều hành thời gian thực là hệ điều hành đa nhiệm dành cho những ứng dụng thời gian thực. Những ứng dụng này bao gồm hệ thống nhúng, robot công nghiệp, hệ thống điều khiển công nghiệp hay các phương tiện nghiên cứu khoa học. Đây là những phần mềm điều khiển chuyên dụng thường được dùng trong những ứng dụng điện toán nhúng có tài nguyên bộ nhớ hạn chế và yêu cầu ngặt nghèo về thời gian đáp ứng tức thời, tính sẵn sàng cao và khả năng tự kiểm soát một cách chính xác.

Một vài ví dụ về hệ điều hành thời gian thực: Adeos, ART Linux, ChorusOS, eCos, ELinOS, FreeRTOS, iRmx, ITRON, Linux, LynxOS, MicroC/OS-II, Nucleus, OS-9, OSE, OSEK/VDX, pSOS, PikeOS, QNX, RedHawk, RSX-11, VxWorks, Windows CE, Xenomai.