TRƯỜNG ĐẠI HỌC VINH

**VIỆN KỸ THUẬT VÀ CÔNG NGHỆ**



**TIỂU LUẬN THIẾT KẾ HỆ THỐNG NHÚNG**

Họ và Tên: Nguyễn Xuân Bảo

MSSV: 1755252021600013

Lớp: 58K-KTĐK & TĐH

GVHD: TS. Mai Thế Anh

Nghệ An, 06-2021

MỤC LỤC

[**PHẦN TRÌNH BÀY NỘI DUNG BÀI TẬP** 4](#_Toc75222977)

[**Câu 1.** 4](#_Toc75222978)

[-Code: 4](#_Toc75222979)

[-Mô phỏng: 4](#_Toc75222980)

[-Code 2: 5](#_Toc75222981)

[-Mô phỏng 2: 7](#_Toc75222982)

[**Câu 2.** 8](#_Toc75222983)

[-Code: 8](#_Toc75222984)

[-Mô phỏng: 9](#_Toc75222985)

[-Code 2: 10](#_Toc75222986)

[-Mô phỏng 2: 12](#_Toc75222987)

[**Câu 3.** 14](#_Toc75222988)

[-Code: 14](#_Toc75222989)

[-Mô phỏng: 18](#_Toc75222990)

[-Code 2: 19](#_Toc75222991)

[-Mô phỏng 2: 21](#_Toc75222992)

[**Câu 4.**. 22](#_Toc75222993)

[-Code: 22](#_Toc75222994)

[-Mô phỏng: 24](#_Toc75222995)

[**Câu 5.** Hãy trình bày: 26](#_Toc75222997)

[1. Trình bày quy trình thiết kế hệ thống nhúng sử dụng vi điều khiển. 26](#_Toc75222998)

[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. 27](#_Toc75222999)

LỜI NÓI ĐẦU

Hệ thống nhúng (*embedded system*) là một thuật ngữ để chỉ một hệ thống có khả năng tự trị được nhúng vào trong một môi trường hay hệ thống mẹ. Đó là các hệ thống tích hợp cả phần cứng và phần mềm phục vụ các bài toán chuyên dụng trong nhiều lĩnh vực công nghiệp, tự động hoá điều khiển, quan trắc và truyền tin. Đặc điểm của các hệ thống nhúng là hoạt động ổn định và có tính năng tự động hoá cao.

Hệ thống nhúng thường được thiết kế để thực hiện một chức năng chuyên biệt nào đó. Khác với các máy tính đa chức năng, chẳng hạn như [máy tính cá nhân](https://vi.wikipedia.org/wiki/M%C3%A1y_t%C3%ADnh_c%C3%A1_nh%C3%A2n), một hệ thống nhúng chỉ thực hiện một hoặc một vài chức năng nhất định, thường đi kèm với những yêu cầu cụ thể và bao gồm một số thiết bị máy móc và phần cứng chuyên dụng mà ta không tìm thấy trong một máy tính đa năng nói chung. Vì hệ thống chỉ được xây dựng cho một số nhiệm vụ nhất định nên các nhà thiết kế có thể tối ưu hóa nó nhằm giảm thiểu kích thước và chi phí sản xuất. Các hệ thống nhúng thường được sản xuất hàng loạt với số lượng lớn. Hệ thống nhúng rất đa dạng, phong phú về chủng loại. Đó có thể là những thiết bị cầm tay nhỏ gọn như đồng hồ kĩ thuật số và máy chơi nhạc MP3, hoặc những sản phẩm lớn như đèn giao thông, bộ kiểm soát trong nhà máy hoặc hệ thống kiểm soát các máy năng lượng hạt nhân. Xét về độ phức tạp, hệ thống nhúng có thể rất đơn giản với một vi điều khiển hoặc rất phức tạp với nhiều đơn vị, các thiết bị ngoại vi và mạng lưới được nằm gọn trong một lớp vỏ máy lớn.

Để có được sự thành công trong đề tài này là sự giúp sức hướng dẫn nhiệt tình của TS. Mai Thế Anh. Em gửi lời cảm ơn đến thầy đã giúp em hoàn thành đề tài một cách hoàn chỉnh và thành công nhất.

# PHẦN TRÌNH BÀY NỘI DUNG BÀI TẬP

## 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:

// NGUYEN XUAN BAO 1755252021600013

#include <REGX52.H>

sbit L7S1 = P3^0;

sbit L7S2 = P3^1;

#define LED7SEG P2;

int main()

{ P2= 0xC0;

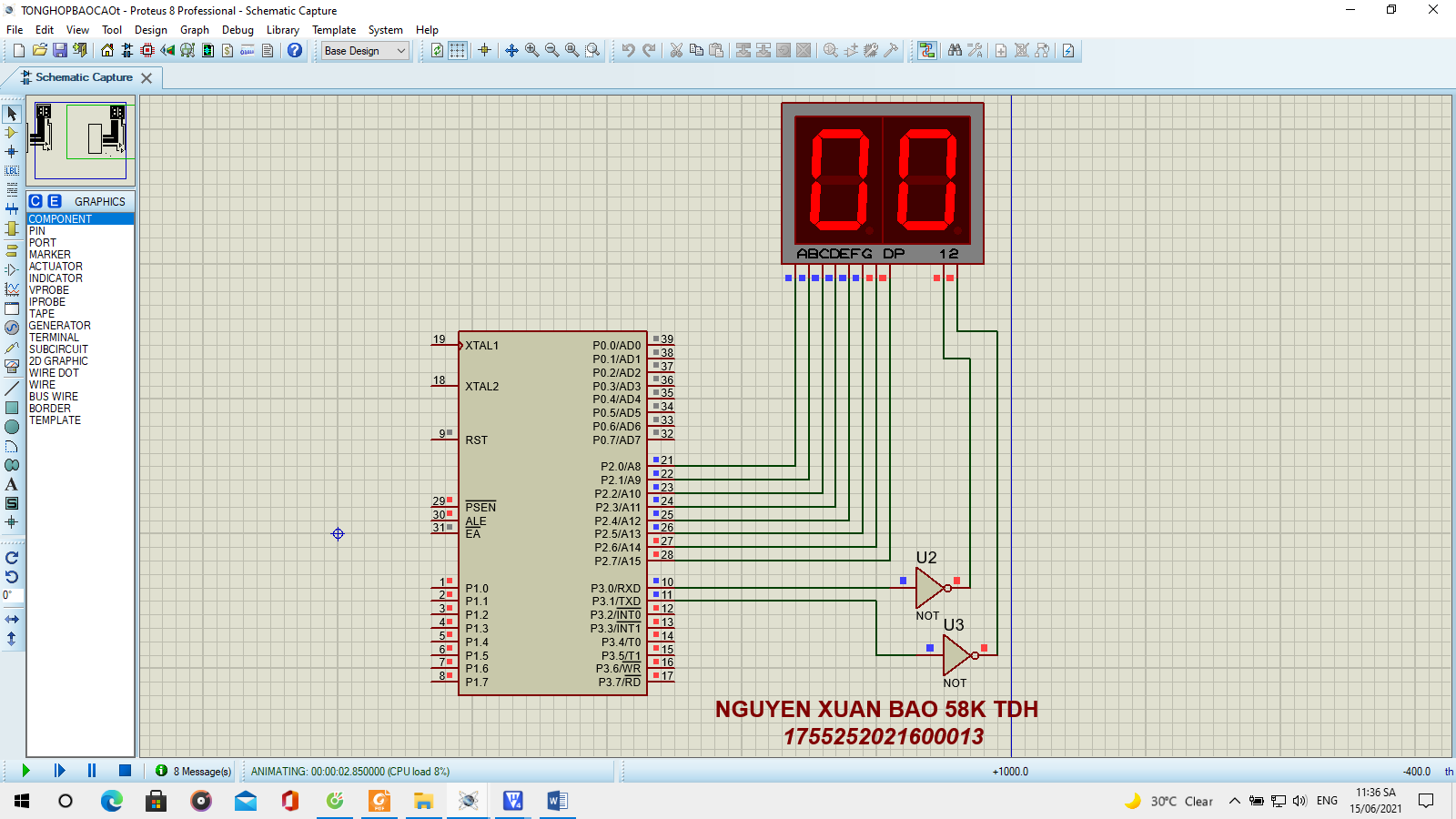
L7S1=0;

L7S2=0;

while(1) ;

}

### -Mô phỏng:



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

- **Số báo danh Nguyễn Xuân Bảo: 02 -> 2+20 =22.**

### -Code 2:

// NGUYEN XUAN BAO 1755252021600013

#include <REGX52.H>

sbit L7S1 = P3^0;

sbit L7S2 = P3^1;

char LED7SEG[11] = {0xC0,0xF9,0xA4,0xB0,0x99,0x92,0x82,0xF8,0x80,0x90};

void Fn\_Delay (unsigned int \_vrui\_Time)

{

while(\_vrui\_Time--);

}

void tangsbd()

{

char i;

int j;

for(i = 0; i < 23; i++)

{

for (j = 0; j < 30; j++)

{

P2 = LED7SEG[i/10];

L7S1 = 0;

Fn\_Delay(500);

L7S1 = 1;

P2 = LED7SEG[i%10];

L7S2 = 0;

Fn\_Delay(500);

L7S2 = 1;

}

}

}

int main ()

{

L7S1 = 0;

L7S2 = 0;

while(1)

{

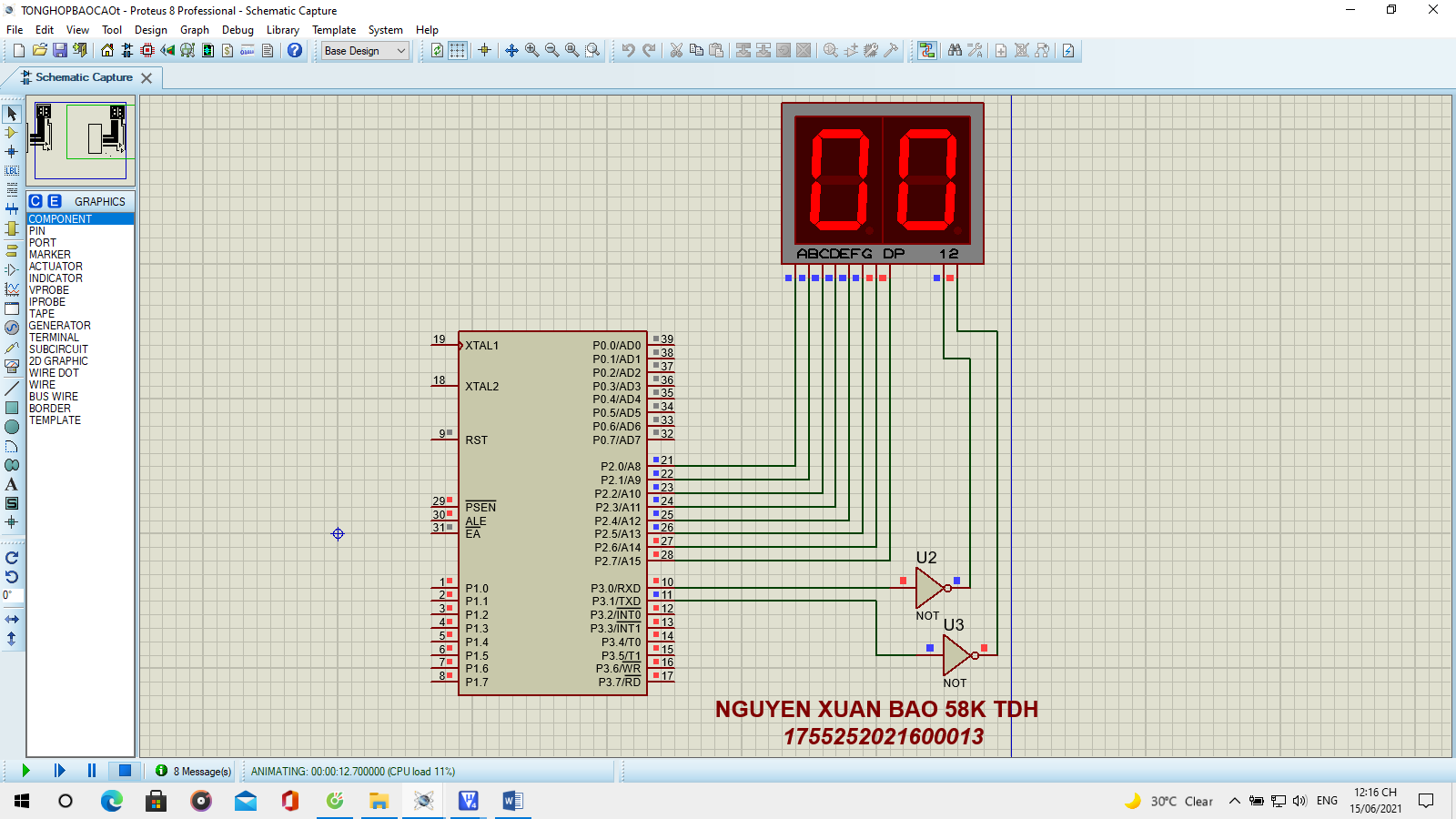
tangsbd();

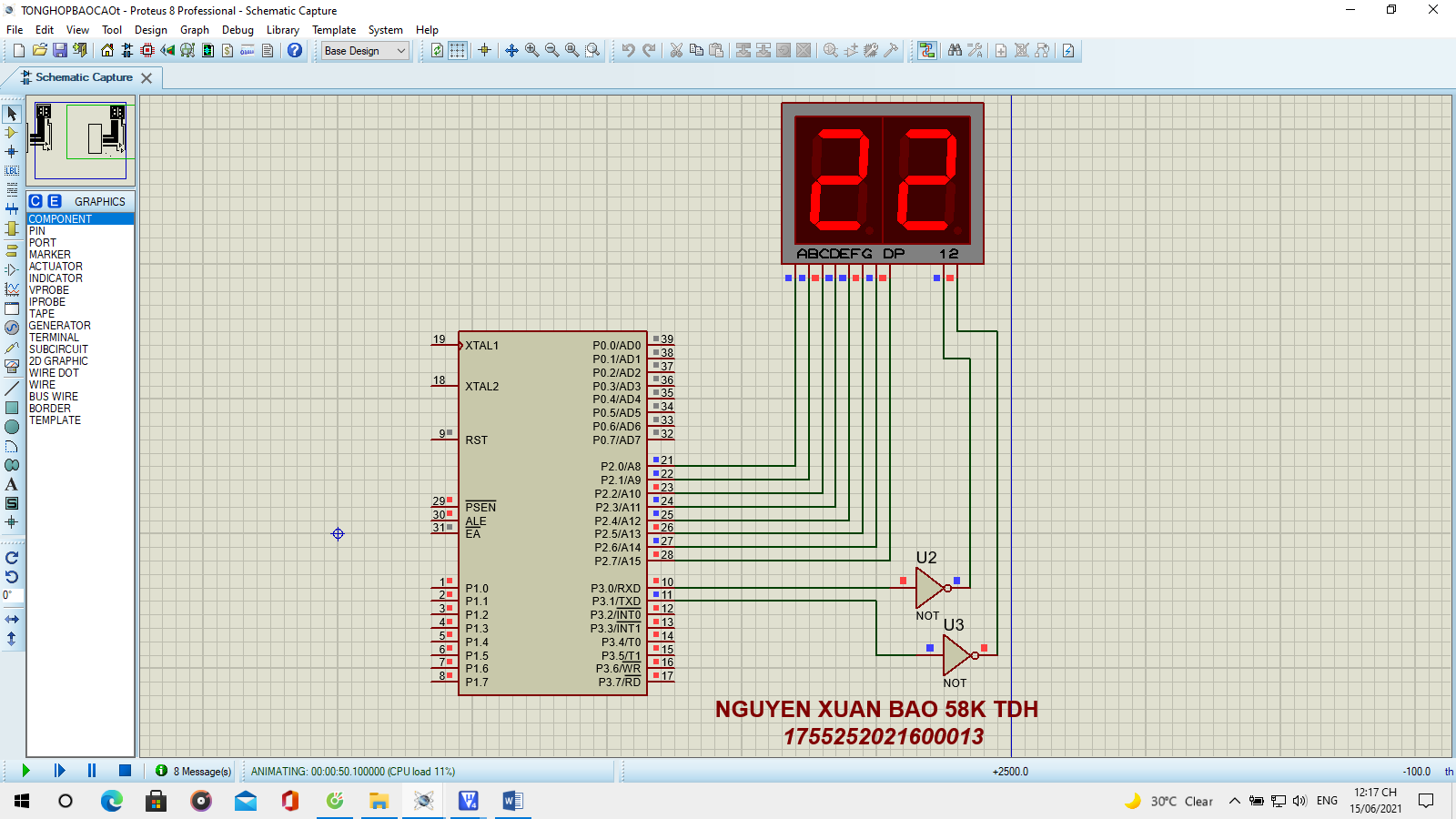
}

return 0;

}

### -Mô phỏng 2:





## 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:

// NGUYEN XUAN BAO 1755252021600013

#include<regx52.h>

void main()

{

IE=0x84;

//cho phép ngat ngoài 0,1(10000100: EA và Ex0,1=1)

EX0=1;

IT0=1;

//Khai báo ngat theo kieu suon xuong

}

void nutbam(void) interrupt 0

//Khai báo trình phuc vu ngat ngoài 0,1(interupt "0,2" ung voi ngat ngoài 0,1)

{

int a=50000;

//Bien dem tre

P1\_3=0;

//Cho Led sáng

while(a--){}

//Tre cho Led sáng vài giây, bang cách thay doi a de tang hay giam thoi gian sáng

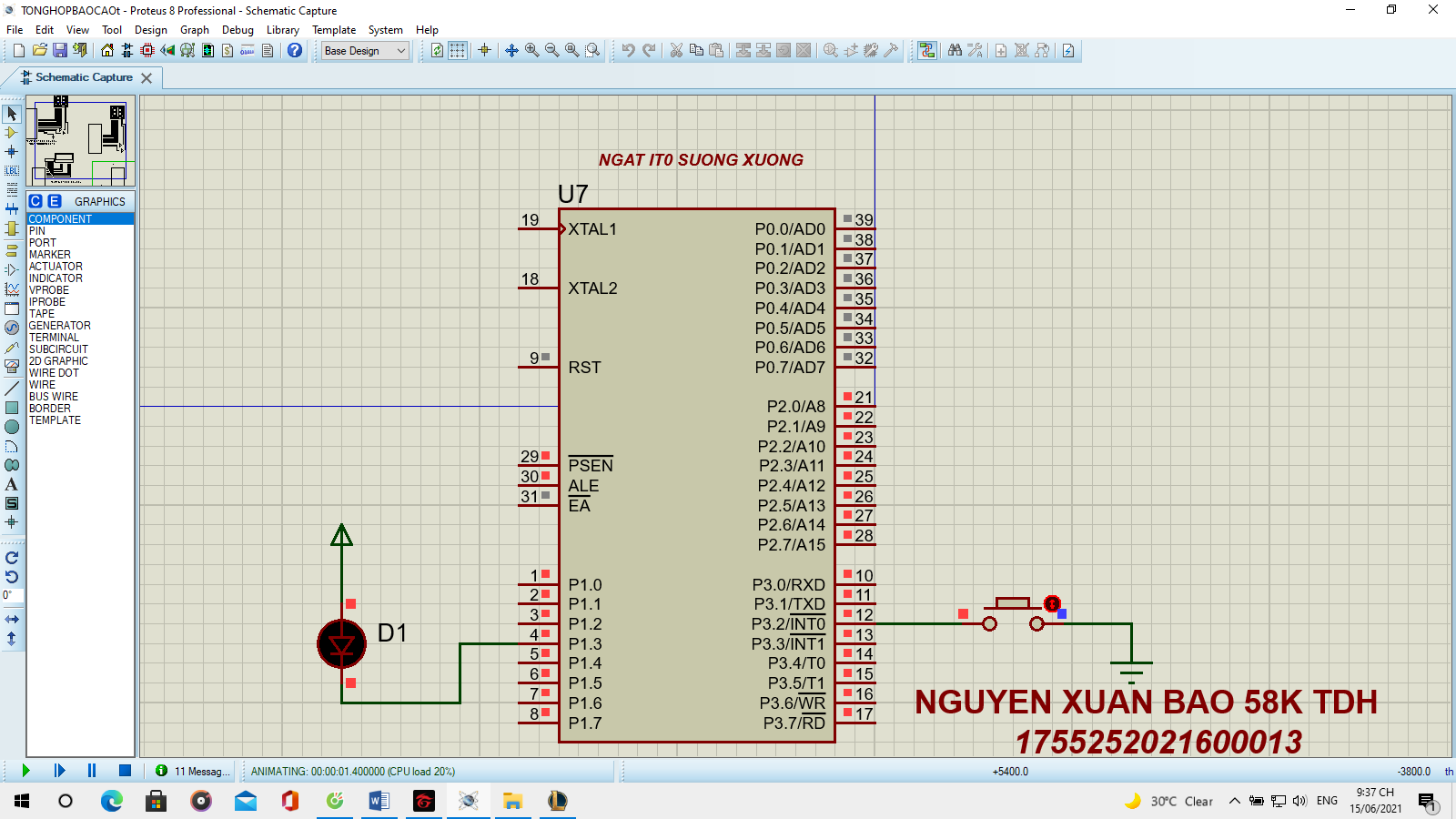
P1\_3=1;

//Tat Led

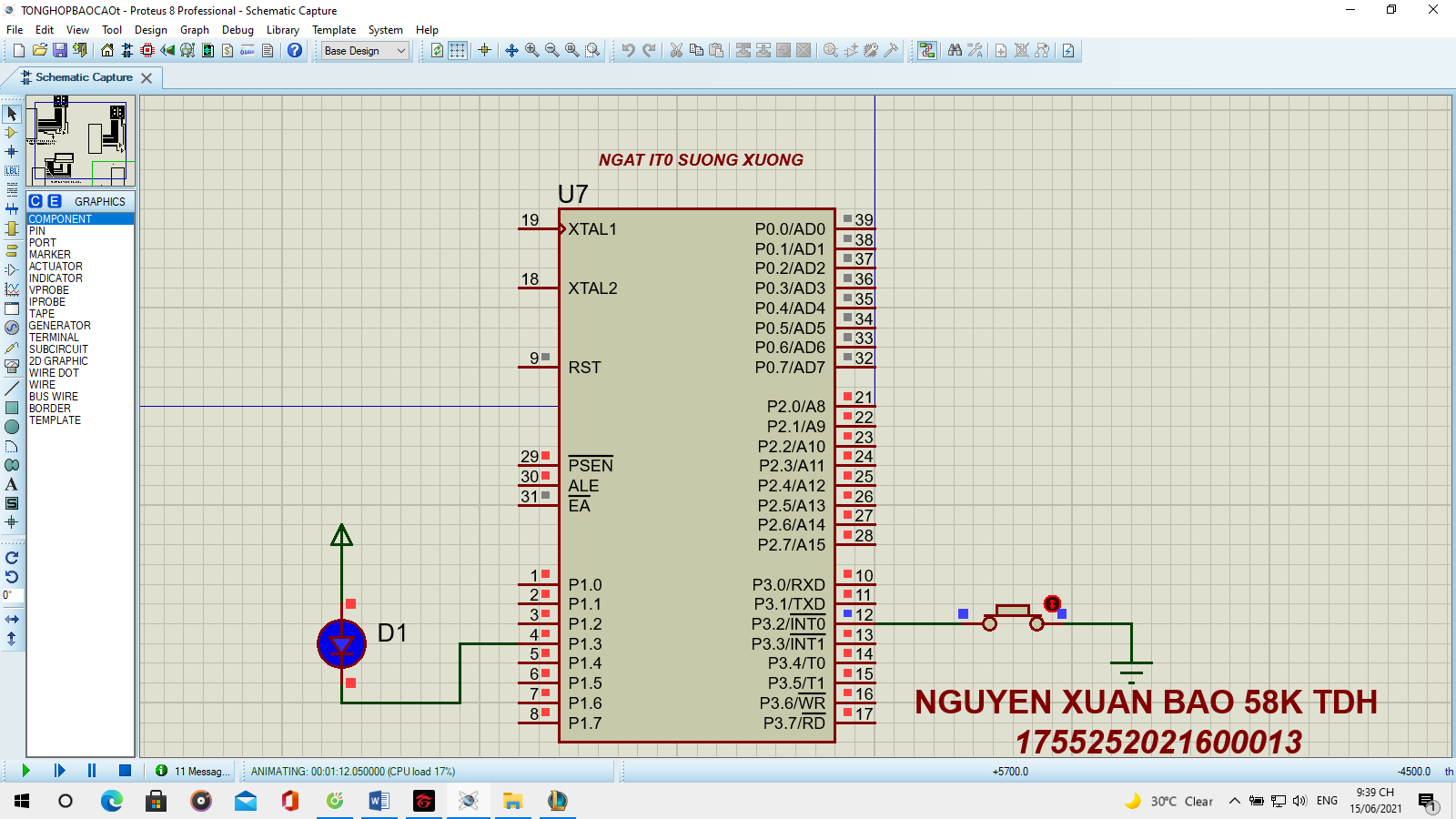
}

### -Mô phỏng:

* Chưa bấm ngắt ngoài INT0 thì led chưa sáng.



* Bấm ngắt ngoài INT0 thì led sáng trong 50000.



* Sau đó thì tắt.

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 2: Số báo danh 02 -> 2+10 = 12.

// NGUYEN XUAN BAO 1755252021600013

#include <REGX52.H>

#define DOUT P2 //Chan xuat data led 7

#define D1 P3\_0 //Quet LED so 1

#define D2 P3\_1 //Quet LED so 2

#define D3 P3\_4 //Quet LED so 3

#define D4 P3\_5 //Quet LED so 4

#define UP P3\_2 //Nut nhan UP

unsigned char Mang[10]={0xC0,0xF9,0xA4,0xB0,0x99,0x92,0x82,0xF8,0x80,0x90};//Mang LED 7

char a,b;

unsigned char dem;

/////////////////////

void delay(unsigned int t){ //Ham tao tre

unsigned int i,j;

for(i=0;i<t;i++)for(j=0;j<100;j++);

}

void quet\_led(){//Ham quet 4 LED 7

D1=0; //Led so 1 sang

DOUT=Mang[a/10];//Xuat data gtri hang chuc cua bien a

delay(1);//Tao tre

DOUT=0xff;//Ngat data (tat bong mo voi led don xep thanh led 7)

D1=1;D2=0; //Led so 1 tat, led 2 sang

DOUT=Mang[a%10];//Xuat data gtri hang don vi cua bien a

delay(1);

DOUT=0xff;

D2=1;D3=0; //Led so 2 tat, led 3 sang

DOUT=Mang[b/10];//Xuat data gtri hang chuc cua bien b

delay(1);

DOUT=0xff;

D3=1;D4=0; //Led so 3 tat, led 4 sang

DOUT=Mang[b%10];//Xuat data gtri hang don vi cua bien b

delay(1);

DOUT=0xff;

D4=1;//Tat Led so 4

}

void nut\_nhan(){//Ham scan nut nhan

if(!UP){//Nut UP dc nhan

a++;//Tang gia tri

if(a>12)a=0;//Set ve 0 khi a>12

while(!UP);//Cho nha phim

}

}

void main(){//Ctr chinh

delay(500);

while(1){//Vong lap ctr

nut\_nhan();//Scan nut nhan

quet\_led();//Scan LED 7

dem++;//Dem tu tang

if(dem>1){

b++;//b tang

dem=0;

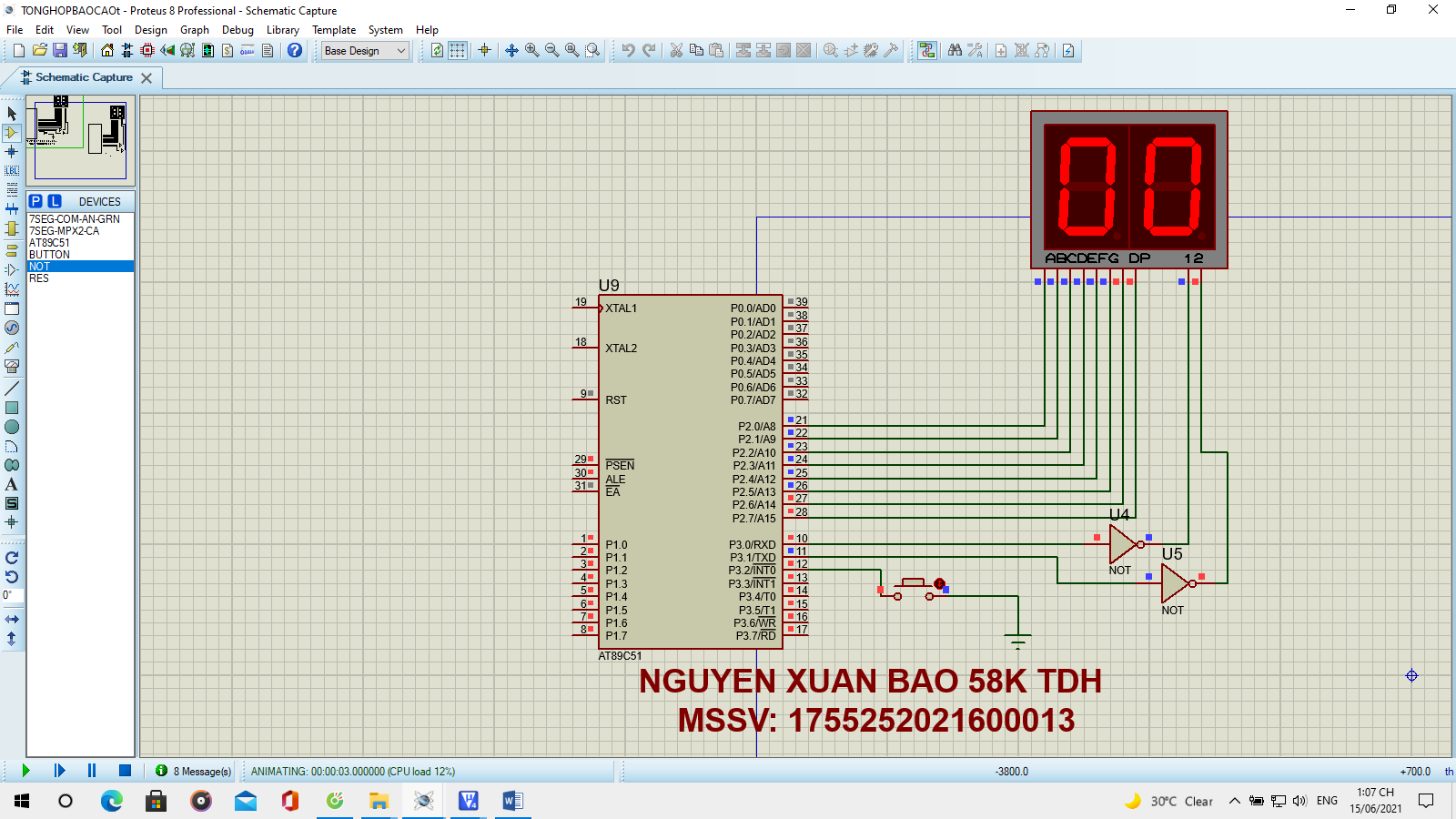
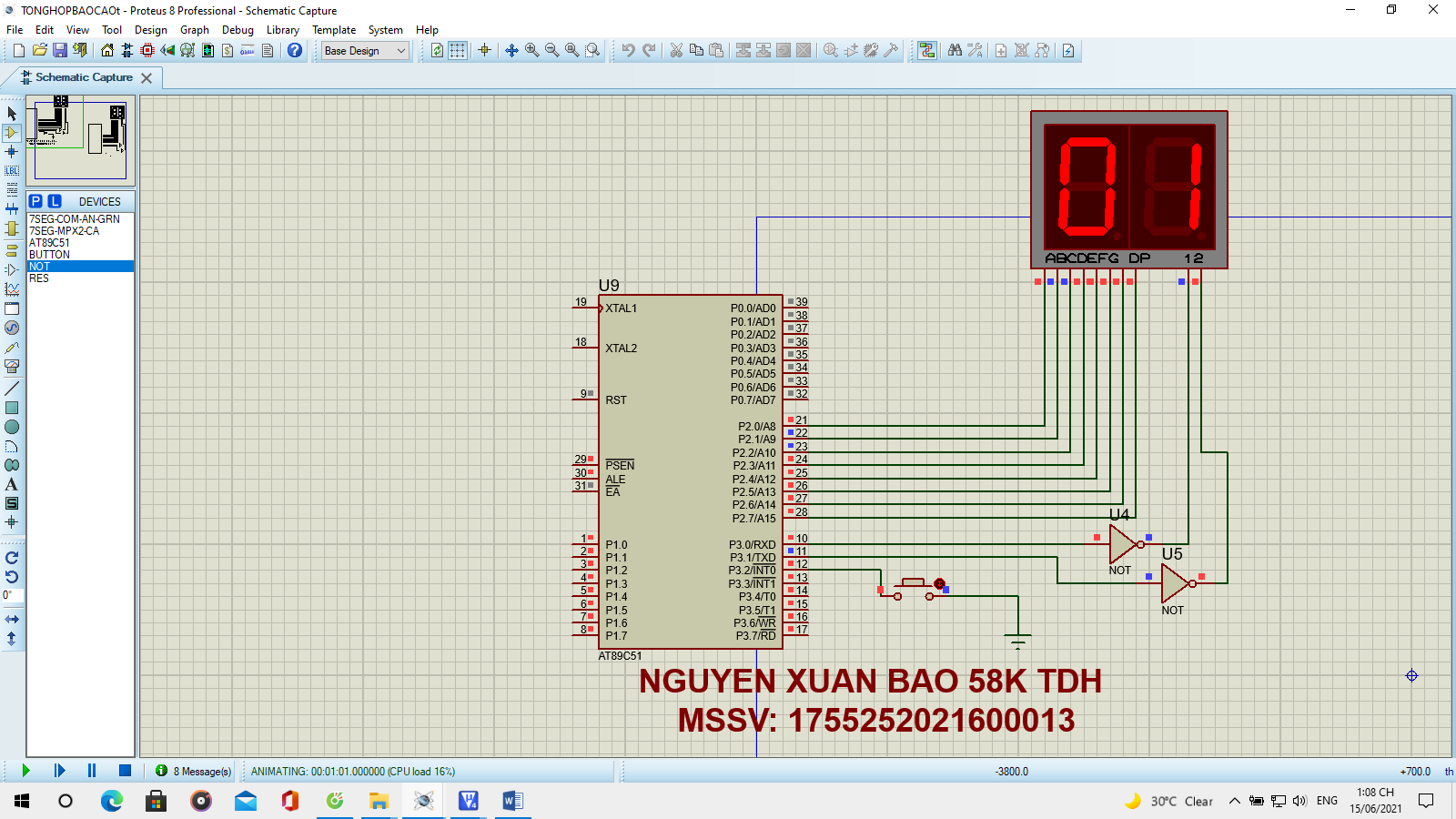
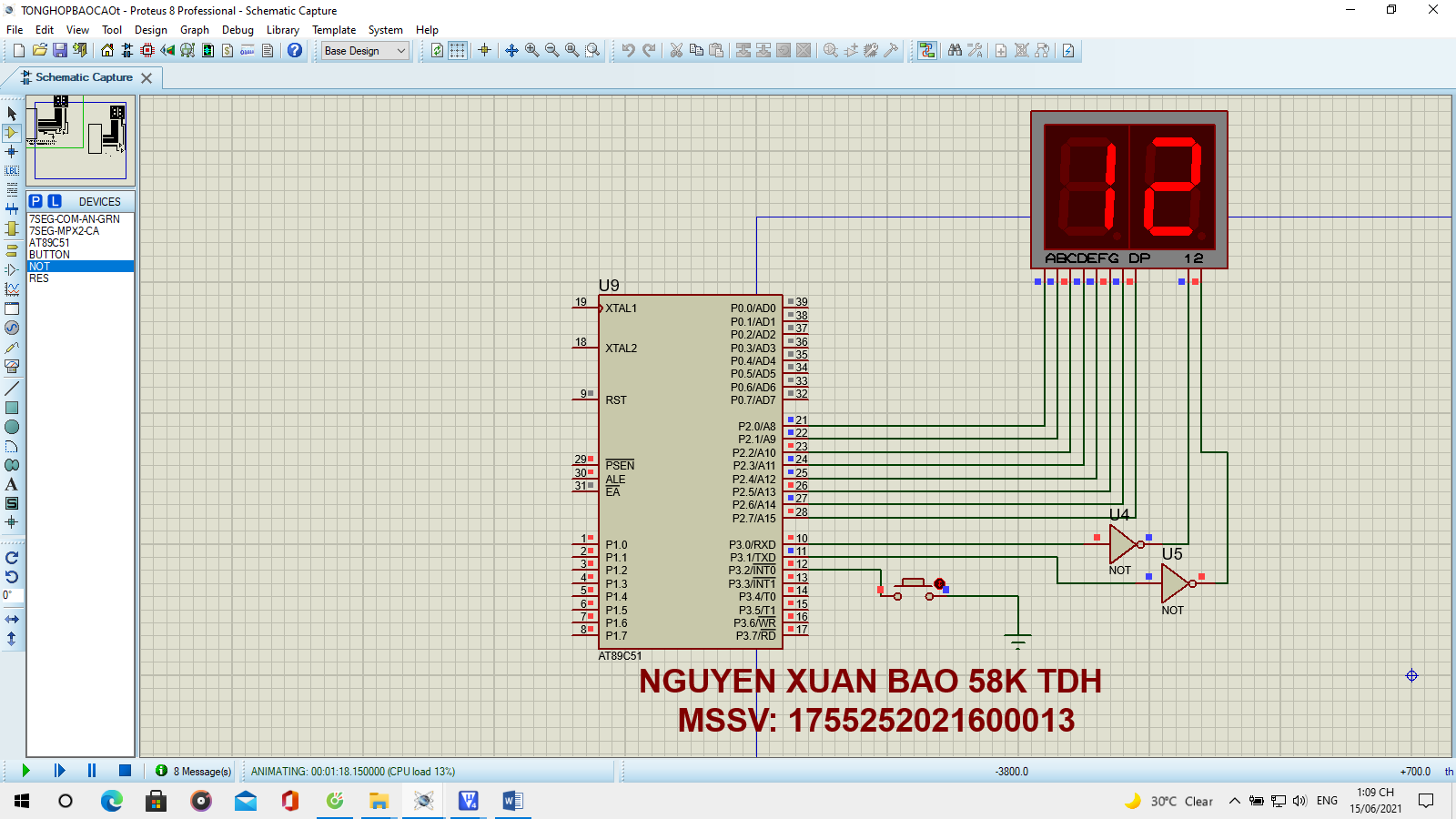
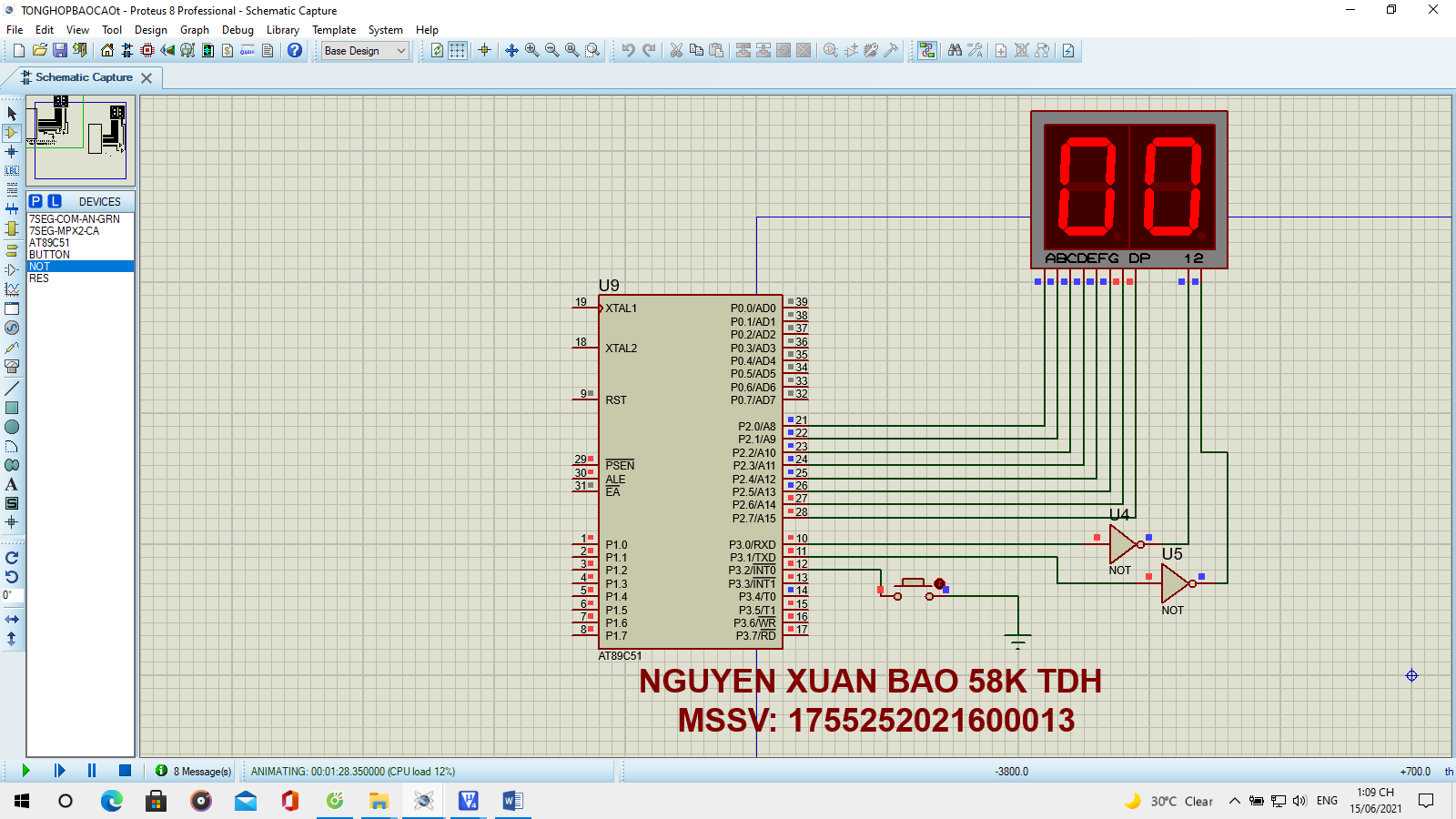
if(b>12)b=0;//Set lai gia tri

}

}

}

### -Mô phỏng 2:

* Chưa nhấn: 
* Nhấn 1 lần 
* Nhấn đến số báo danh 12
* Trả về số 0 khi nhấn lần thứ 13

## 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: //NGUYEN XUAN BAO 1755252021600013

#include <REGX52.H>

/\*\*\*\*\*\*\*\*Khai bao chan giao tiep\*\*\*\*\*\*\*\*\*\*\*\*/

#define LCD\_RS P0\_0

#define LCD\_RW P0\_1

#define LCD\_EN P0\_2

#define LCD\_D4 P0\_4

#define LCD\_D5 P0\_5

#define LCD\_D6 P0\_6

#define LCD\_D7 P0\_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++);

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*Ctr 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();

}

void LCD\_Clear(){// Ham Xoa Man Hinh LCD

LCD\_SendCommand(0x01);

delay\_us(10);

}

// 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++;

}

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*Ctr chinh\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

void main(){

LCD\_Init();//Khoi tao LCD

delay\_ms(10);

LCD\_Puts("NGUYEN XUAN BAO");//Gui chuoi len LCD

delay\_ms(1000);

LCD\_Clear();//Xoa man hinh

LCD\_Gotoxy(0,8);//Tro toi vi tri

LCD\_Puts("K58 TDH");

delay\_ms(1000);

LCD\_Gotoxy(0,0);

LCD\_Puts("NGUYEN XUAN BAO");

delay\_ms(1000);

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

LCD\_Puts("1755252021600013");

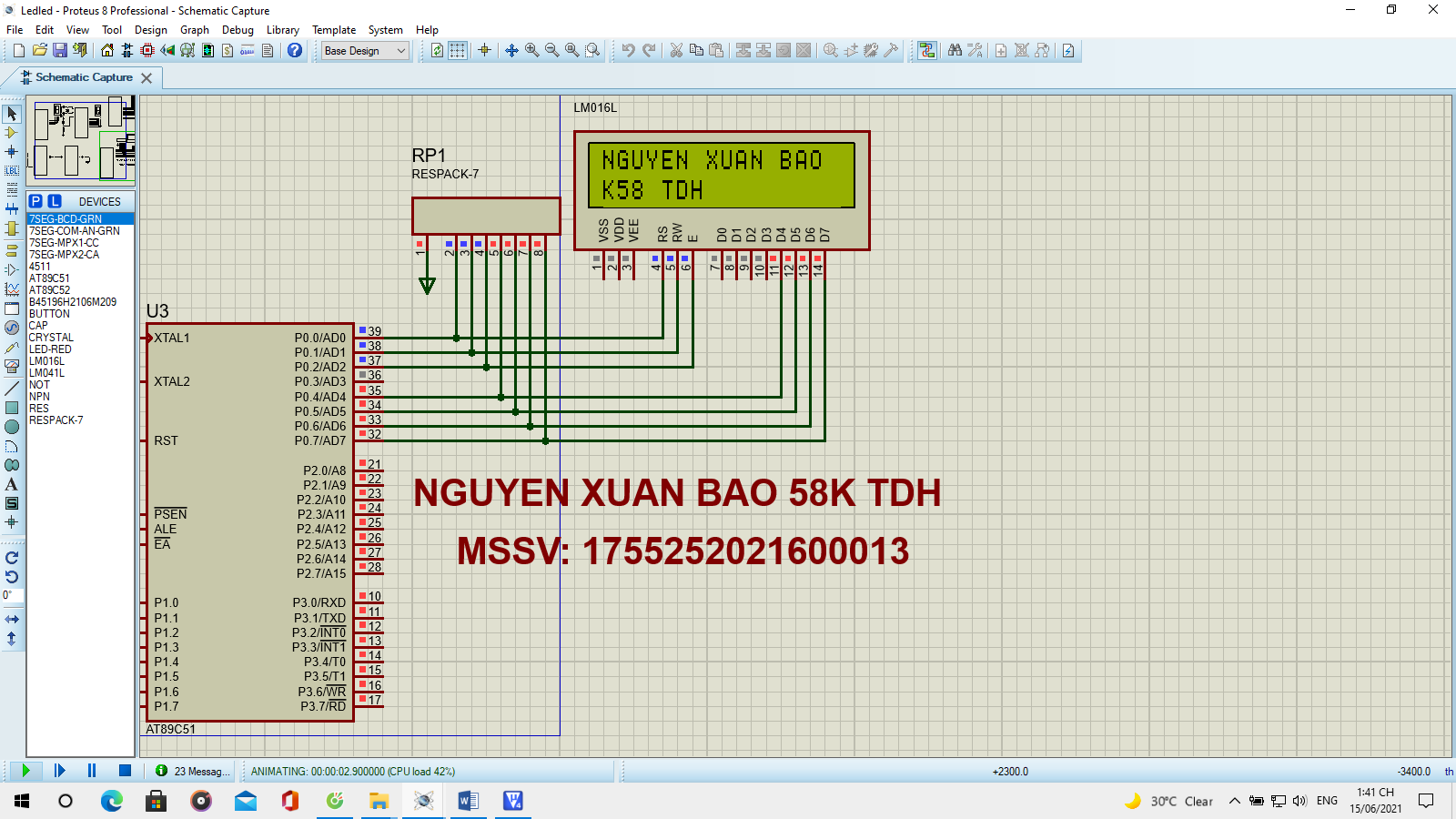
delay\_ms(1000000);

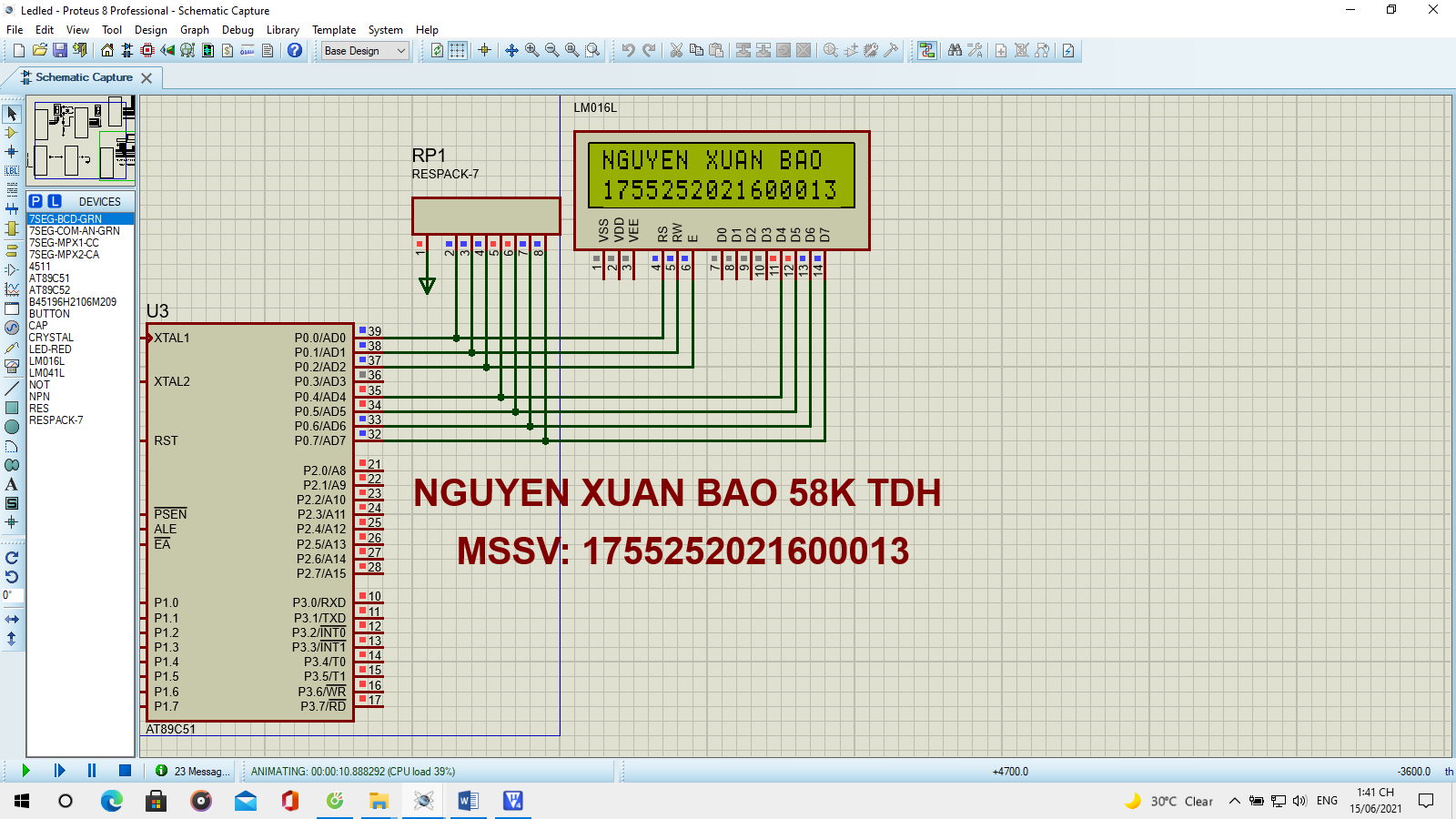
LCD\_Clear();//Xoa man hinh

while(1);//ko lam gi ca

}

### -Mô phỏng:





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 2: //NGUYEN XUAN BAO 1755252021600013

#include <REGX52.H>

#include <string.h>

void delay(unsigned int t){//Ctr delay 50ms dung timer0

unsigned int i;

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

TH0=0x3c; //-50000us

TL0=0xb0;

TR0=1;

while(!TF0); //cho timer0 tran

TF0=TR0=0;

}

}

void send(unsigned char \*s){//Ham gui chuoi ki tu qua UART

unsigned char n,i;

n=strlen(s); //Dem xem co bao nhieu ky tu

for(i=0;i<n;i++){//Vong lap gui tung ky tu 1

SBUF=s[i];//Gui 1 byte

while(!TI); TI=0;//xoa co truyen

}

}

void ngat\_uart()interrupt 4 //Ngat nhan du lieu tu uart

{

if(RI){

P1=SBUF; //Xuat du lieu ra Post 1

}

RI = 0; //Xoa co nhan

}

void main(void){

TMOD=0x21; //Timer 1 che do 8bit nap lai tu dong, timer0 cho delay che do 16bit

SCON=0x50;//01010000 che do 1, cho phep nhan

TH1=TL1=0xFD;//Nap 253 tao baud 9600 ko nhan doi baud

TR1=1; //khoi dong timer1

ES=1; //Ngat UART

EA=1; //Cho phep ngat

delay(20);

send("HO/TEN: NGUYEN XUAN BAO 58K TDH \r");

delay(20);

send("MSSV: 175525201600013 \r");

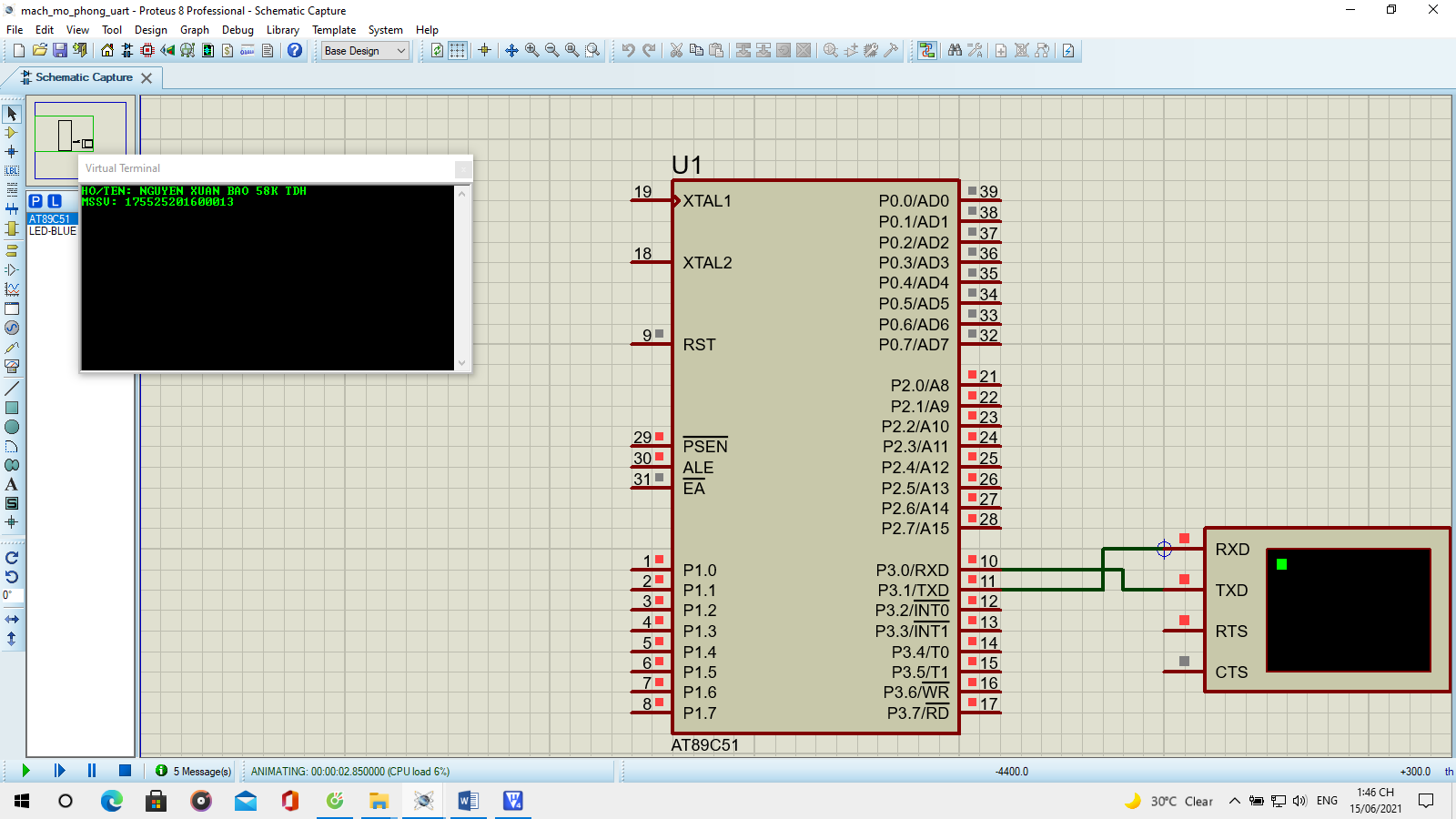
while(1){

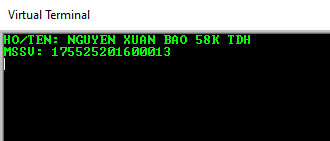
//ko lam gi ca

}

}

### -Mô phỏng 2:



****

## 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:

#include <REGX52.H>

#include <RTX51TNY.H>

#define INIT 0

#define DO 1

#define BUTT 2

sbit LED\_DO = P1^2;

sbit BUTTON = P1^3;

void USART(void) interrupt 4

{

if(RI)

{ //Clear flag

RI=0; //Nhan ki tu

isr\_send\_signal(DO);

}

}

//=========Ham Start up==========

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 Task\_Led\_Do(void) \_task\_ DO

{

while(1)

{

os\_wait2(K\_SIG ,50); //Cho signal voi time out 50 ticks

LED\_DO ^= 1; //Dao trang thai Led Do

}

}

void Task\_BUTTON(void) \_task\_ BUTT

{

while(1)

{

if(BUTTON == 0) //Nhan nut nhan = 0

{

os\_send\_signal(DO); //Gui signal cho task DO

while(BUTTON==0); //Cho nut nhan = 1(Chong nhieu)

}

os\_wait2(K\_TMO, 10); //Cho 10 ticks = 100ms

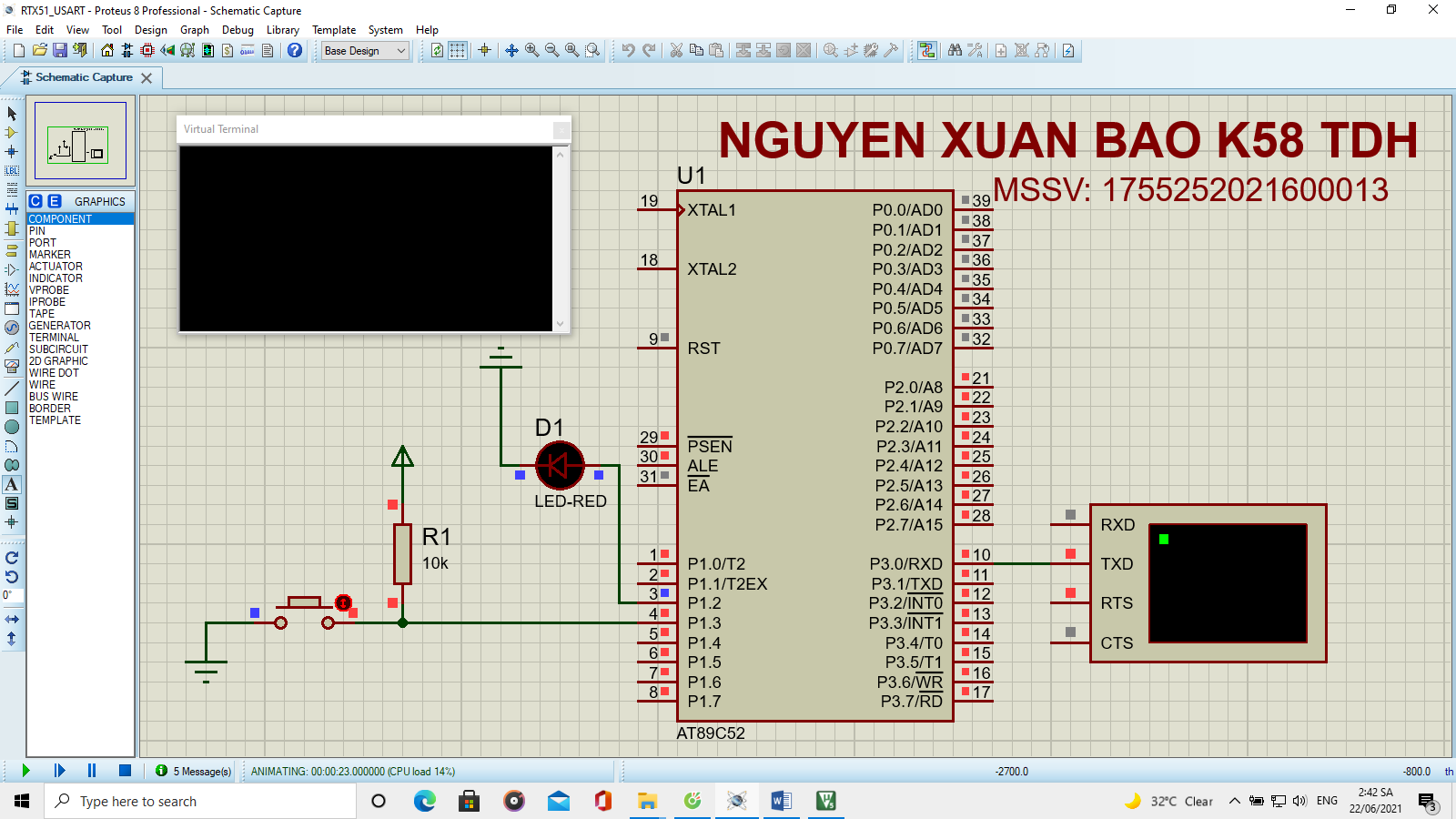
}

}

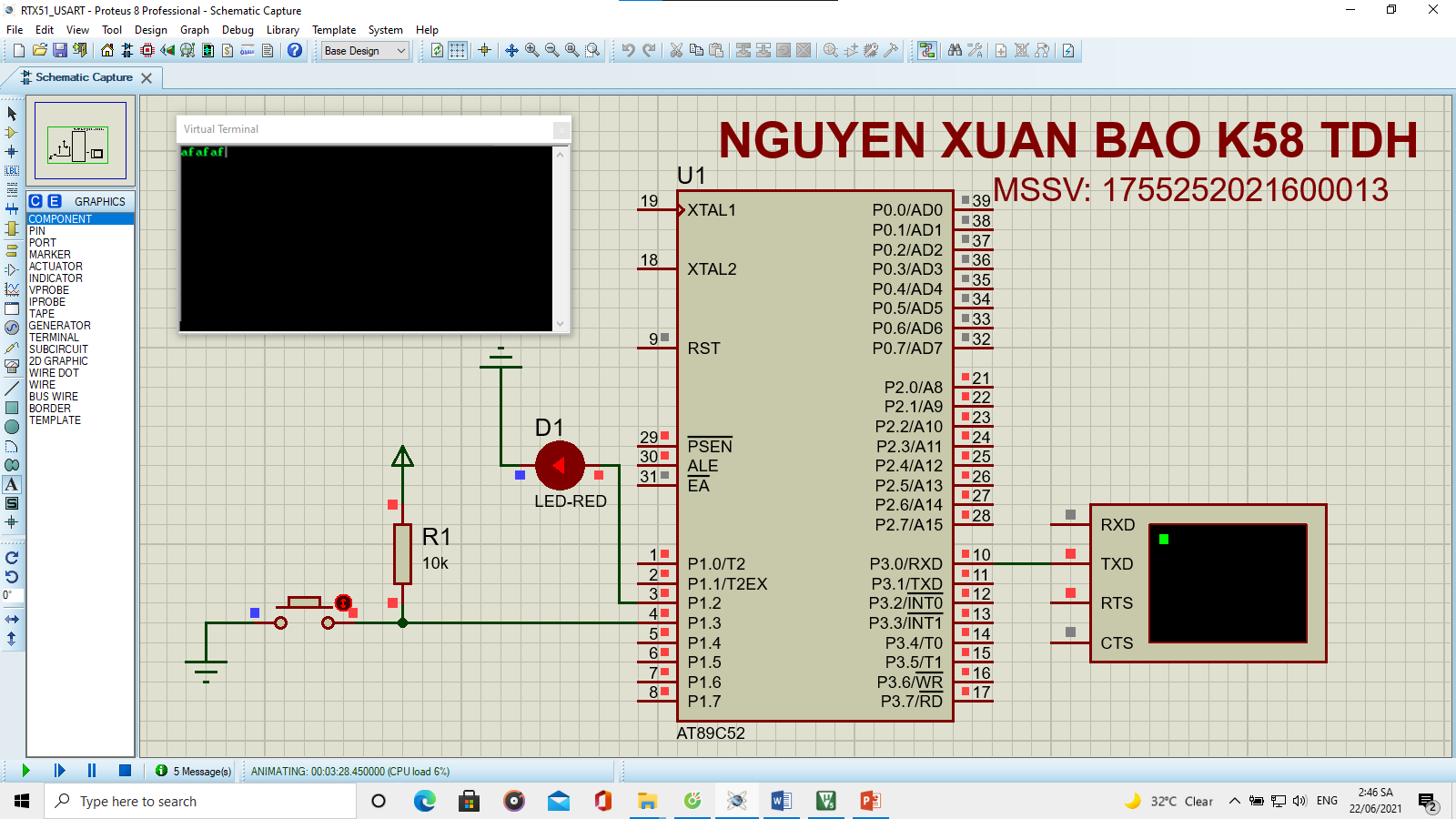
### -Mô phỏng:

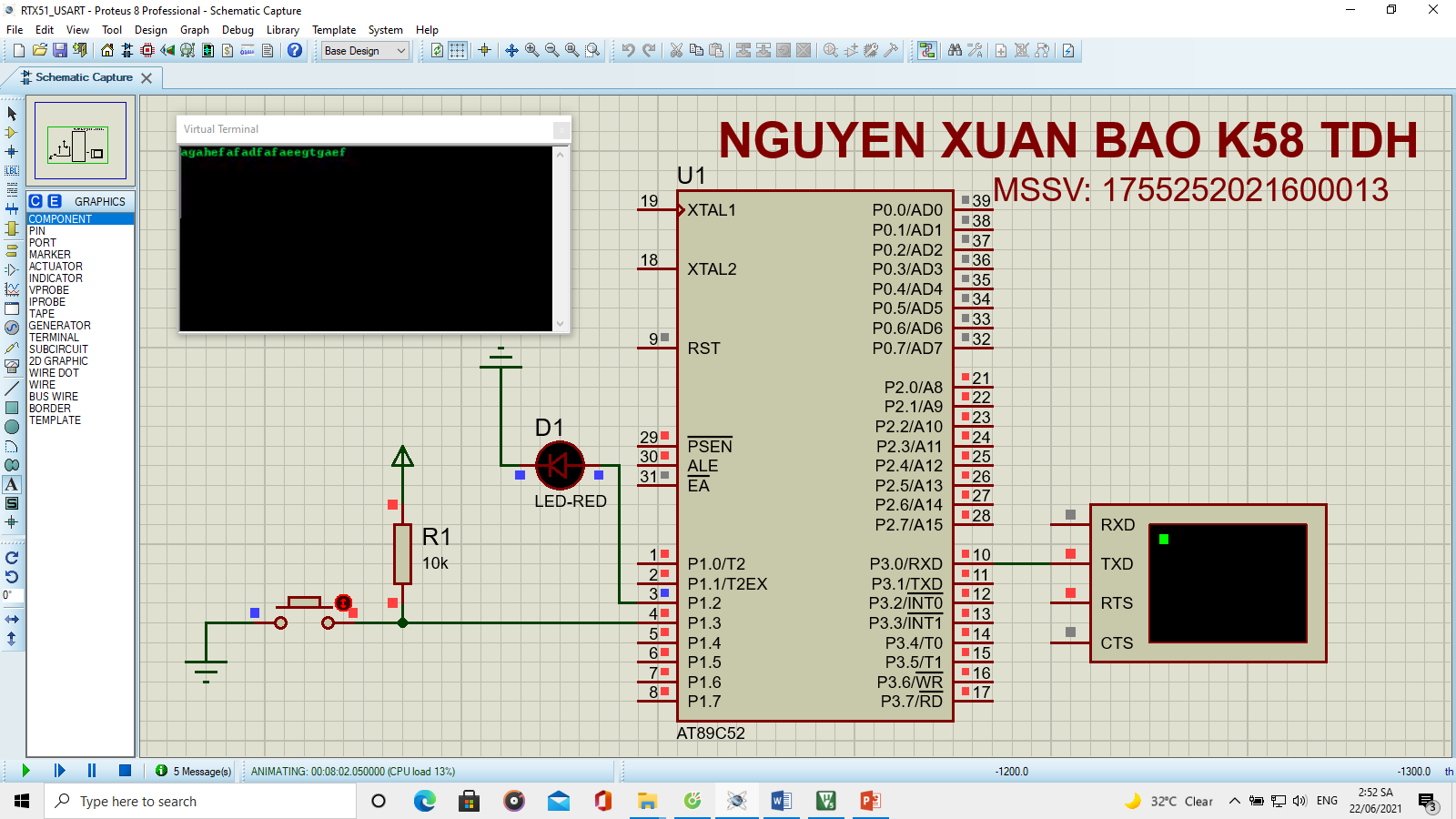
- Tắt bật đèn bằng tín hiệu BUTTON.

### 



- Nhấp chuột vào bảng Virtual Teminal rồi ấn phím máy tính để gửi tín hiệu tắt bật đèn.

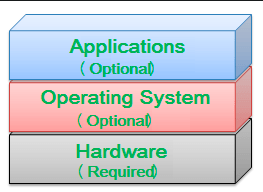




## Câu 5. Hãy trình bày:

### 1. Trình bày quy trình thiết kế hệ thống nhúng sử dụng vi điều khiển.

Mỗi hệ thống nhúng đều có một kiến trúc thổng thể như sau:



**Hardware** Vi xử lý, bộ nhớ, tụ điện, điện trở, mạch tích hợp, bảng mạch in, connector, …. Tất nhiên, đây là thành phần bắt buột phải có cho tất cả các hệ thống nhúng.

Nói thêm về bộ vi xử lý trong Hệ thống nhúng:

* Vi xử lý Bộ xử lý được thiết kế riêng, chỉ bao gồm phần xử lý. Có thể thay đổi thêm bớt các thành phần ngoại vi một cách linh hoạt.
* Vi điều khiển Được tích hợp các thành phần ngoại vi trên chip để giảm kích thước hệ thống.
* SoC (System on Chip) Một vi mạch tích hợp cao, hỗ trợ đa nhân xử lý và nhiều giao tiếp trên 1 chip. Giúp tăng tốc thời gian thiết kế hệ thống. Sử dụng như một mạch tích hợp cho ứng dụng cụ thể (ASIC) hoặc mạch logic khả trình (FPGA)

**Phần mềm hệ thống**

* Không bắt buộc phải có.
* Device driver: UART, Ethernet, ADC…
* Hệ điều hành nhúng: eCos, ucLinux, VxWorks, Monta Vista Linux, BIOS, QNX…
* Quản lý bộ nhớ, quản lý tiến trình, quản lý chia sẽ tài nguyên
* Có thể tái sử dụng trên một hệ thống nhúng khác

**Phần mềm ứng dụng**

* Không bắt buộc phải có.
* Quyết định hành vi (chức năng) của một hệ thống nhúng.
* Khó tái sử dụng trên một hệ thống nhúng khác.

- Thiết kế hệ thống nhúng:

Việc thiết kế hệ thống nhúng khá phức tạp và đòi hỏi người kỹ sư thiết kế phải có nhiều kinh nghiệm. Hầu hết các mô hình được sử dụng trong việc thiết kế hệ thống nhúng đều dựa trên một hoặc sự kết hợp nhiều mô hình trong các các mô hình phát triển sau:

**Mô hình big-bang** Không hề có một kế hoạch cụ thể trước và trong suốt quá trình phát triển hệ thống

**Mô hình code-and-fix** Là một mô hình khá đơn giản, chỉ thích hợp cho các chương trình nhỏ (không đòi hỏi việc bảo trì), không thích hợp với các hệ thống lớn, bao gồm 2 bước:

* Viết code
* Fix các vấn đề phát sinh

**Mô hình waterfall** Trong mô hình này, quá trình phát triển hệ thống được xây dựng theo từng bước, các kết quả của một bước sẽ được sử dụng cho bước kế tiếp.

**Mô hình spiral** Quá trình phát triển hệ thống được chia thành nhiều giai đoạn. Dựa trên sự phản hồi từ các giai đoạn, kết hợp trở lại vào quá trình để lên kế hoạch cho việc thực hiện giai đoạn tiếp theo.

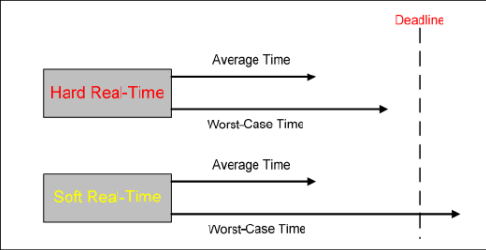
### 2. Hệ điều hành thời gian thực (RTOS). Ưu điểm, nhược điểm và ứng dụng của hệ điều hành thời gian thực trong thiết kế các hệ thống nhúng.

- Hệ thống thời gian thực (RTOS)

* Thời gian ( Time) :  
  - Sự chính xác của hệ thống không chỉ phụ thuộc vào kết quả  
  tính toán logic mà còn phụ thuộc vào thời gian cho ra kết quả.
* Thực ( Real):  
  - Đáp ứng của hệ thống với những sự kiện bên ngoài.
* Thời gian thực ( Real-Time):  
  -Phải đảm bảo các yếu tố :  
  Đáp ứng nhanh  
  Dự đoán được.  
  Các tác vụ ( Real-time Task) được xác định bằng  
  deadline.

Deadline là thời gian tối đa một tác vụ PHẢI hoàn  
thành việc thính toán.

- Thời gian thức cứng và thời gian thực mềm



* Thời gian thực cứng:  
  - Một tác vụ là thời gian thực cứng nếu như thời gian tính toán  
  vượt quá deadtime có thể gây ra sự phá vỡ môi trường điều khiển.
* Thời gian thực mềm:  
  - Một tác vụ là thời gian thực mềm nếu như đảm bảo thực thi  
  trong deadtime cho phép và nếu như không đảm bảo thì sẽ không  
  tạo ra những nguy hại nào đáng kể cho hệ thống và không làm ảnh  
  hưởng đến sự ứng xử của hệ thống.

- Định nghĩa hệ thời gian thực (RTOS).

* Là hệ thống có:  
  - Lịch trình thực thi theo thời gian.

- Quản lý tài nguyên hệ thống.  
- Cung cấp những nền tảng cơ bản để phát triển các ứng dụng.

- Ưu điểm: Ưu điểm lớn của RTOS là xử lý nhanh chóng vì thế nó sẽ dành cho các thiết bị đòi hỏi khả năng xử lý có độ trễ thấp nhất có thể. Lợi ích nó đem lại bao gồm đa nhiệm tốt, ưu tiên các nhiệm vụ và quản lý chia sẻ các tài nguyên. Ngoài ra nó cũng không đòi hỏi nhiều về tài nguyên hay bộ nhớ [RAM](https://fptshop.com.vn/tin-tuc/danh-gia/ram-la-gi-bo-nho-ram-dung-de-lam-gi-57115) quá lớn.  
- Nhược điểm: Tốn thêm bộ nhớ cho RTOS.