**TRƯỜNG ĐẠI HỌC VINH**

**VIỆN KỸ THUẬT VÀ CÔNG NGHỆ**



**TIỂU LUẬN**

**THIẾT KẾ HỆ THỐNG NHÚNG**

**Giảng viên: TS. Mai Thế Anh**

**Họ tên sinh viên: Trần Văn Quyết**

**Lớp: 58K – KTĐK & TĐH**

**Mã số sinh viên: 1755252021600011**

**Nghệ An, năm 2021**

**Mục lục**

[**câu 5: hãy trình bày:** 3](#_Toc75677267)

[**Câu 1: sử dụng vi điều khiển AT89C52, mô phỏng trên phần mềm protues và lập trình thực hiện các nhiệm vụ sau:** 5](#_Toc75677268)

[**Câu 2: sử dụng vi điều khiển AT89C52, mô phỏng trên phần mềm protues và lập trình thực hiện các nhiệm vụ sau:** 11](#_Toc75677269)

[**Câu 3: sử dụng vi điều khiển AT89C52 thực hiện các nhiệm vụ sau:** 19](#_Toc75677270)

[**Câu 4: sử dụng vi điều khiển AT89C52 vẽ sơ đồ mô phỏng protues ghép nối với LED D1qua 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 tast BUTTON đến tast LED. Tast LED thực hiện đảo trạng thái của LED D1 khi nhận được tín hiệu tast khác gửi tới.** 31](#_Toc75677271)

# **câu 5: hãy trình bày:**

1. **Trình bày quy trình thiết kế hệ thống nhúng sử dụng vi điều khiển**
2. **Hệ điều hành thời gian thực (RTOS). Ưu , nhược điểm và ứng dụng của hệ điều hành thời gian thực trong thiết kế các hệ thống nhúng.**

**Bài làm**

**1.**

Quy trình: quy trình thiết kế nhằm vào hai mục tiêu đó là cung cấp cho chúng ta các bước khác nhau trong thiết kế hệ thống nhúng trước khi chúng ta đi sâu vào nghiên cứu chi tiết tuèng bước, thứ hai là cho phép chúng ta hiểu hơn về bản thân phương pháp thiết kế.

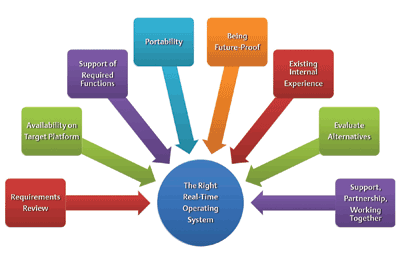
Đầu tiên, cho phép chúng ta giữ 1 bảng ghi chú về 1 thiết kế để đảm bảo rằng chúng ta đã làm mọi thứ chúng ta cần làm.

Thứ hai, cho phép chúng ta phát triển các công cụ thiết kế được hỗ trợ bởi máy tính, phát triển 1 công cụ duy nhất để hỗ trợ toàn bộ quá trình thiết kế 1 hệ thống máy tính nhúng sẽ là 1 nhiệm vụ khó khăn nhưng bằng cách chia quy trình thiết kế thành các bước nhỏ có thể quản lý, chúng ta có thể tạo ra các công cụ tự động hóa quá trình thiết kế cho mỗi bước trong quy ttrìn.

Thứ ba, một phương pháp thiết kế giúp các thành viên của nhóm thiết kế trao đổi với nhau dễ dàng hơn. Bằng cách xác định 1 quy trình thiết kế tổng thể, các thành viên trong nhóm có thể dễ dàng hơn.



**RTOS** tắt của cụm từ Real-time operating system, đây là 1 hệ điều hành (OS) dùng để phục vụ các ứng dụng thời gian thực với khả năng xử lý dữ liệu siêu nhanh. Đối với hệ điều hành thời gian thực RTOS thì các ứng dụng đều được thực thi với thời gian chính xác .



RTOS thường là 1 phân đoạn của chương trình, nó có nhiệm vụ giải quyết và điều phối các tác vụ lập trình và phân mức ưu tiên cho chúng.

Ưu điểm: RTOS xử lý nhanh chóng vì thế nó sẽ dành cho các thiết bị đòi hỏi khả năng xử lý có đỗ 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

Nhược điểm: do phải chờ tast thực thi xong thì tast có mức ưu tiên cao mơí được thực thi, do đó mức đáp ứng của hệ thống thấp

ứng dụng: xử lý ngắt, dịch vụ quản lý thời gian, dịch vụ quản lý thiết bị, dịch vụ quản lý bộ nhớ, dịch vụ quản lý các kết nối vào ra.

# **Câu 1: sử dụng vi điều khiển AT89C52, mô phỏng trên phần mềm protues và lập trình thực hiện các nhiệm vụ sau:**

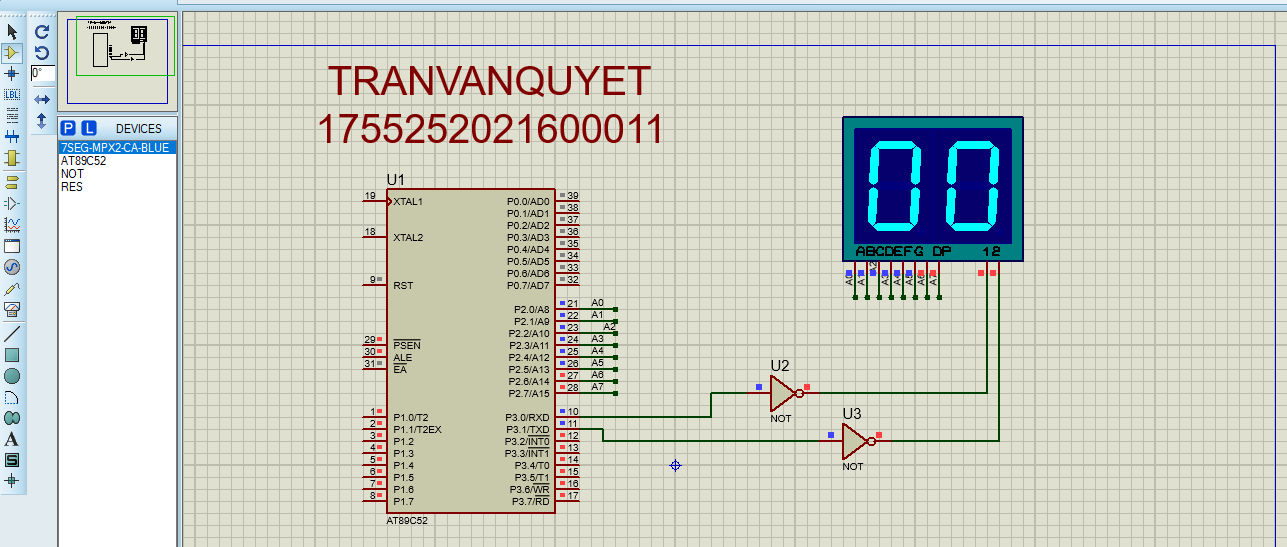
* 1. **Hiển thị số 00 lên hai led 7 thanh nối vào cổng P2 theo phương pháp quét LED**
  2. **Tăng số đếm sau mỗi 500ms, nếu số đếm bằng “ SBD+20” thì dừng lại ( sử dụng timer để định thời gian).**

**Bài làm**

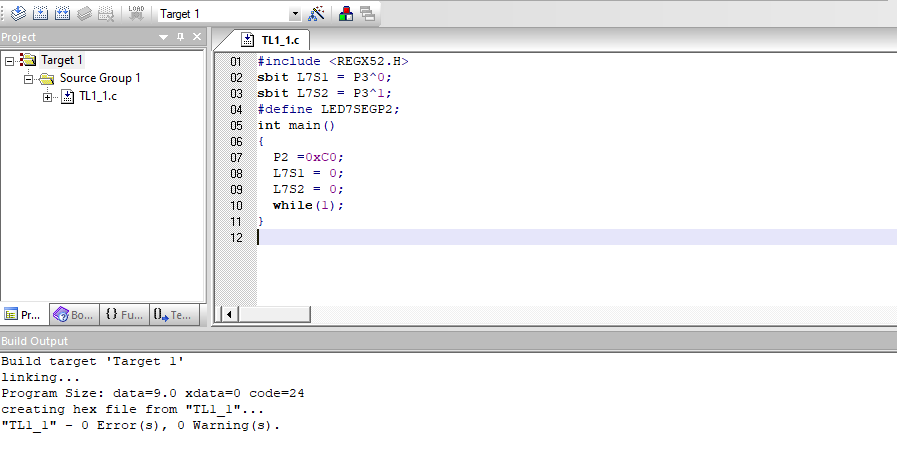
**Phương pháp quét Led:** là thuật ngữ chuyên nghiệp và thường xuyên được đề cập tới trong nghành Led, phương pháp quét là một trong những yếu tố quyết định chất lượng, độ sáng, độ làm mới của màn hình Led. Led sáng là Led được điều khiển bởi IC, ngược lại là Led không được quét.

* 1. Hiển thị số 00 lên hai Led 7 thanh nối vào cổng P2

**Mô phỏng protues**



**Code trên Keil c**



**Code đánh tay**

#include <REGX52.H>

sbit L7S1 = P3^0;

sbit L7S2 = P3^1;

#define LED7SEGP2;

int main()

{

P2 =0xC0;

L7S1 = 0;

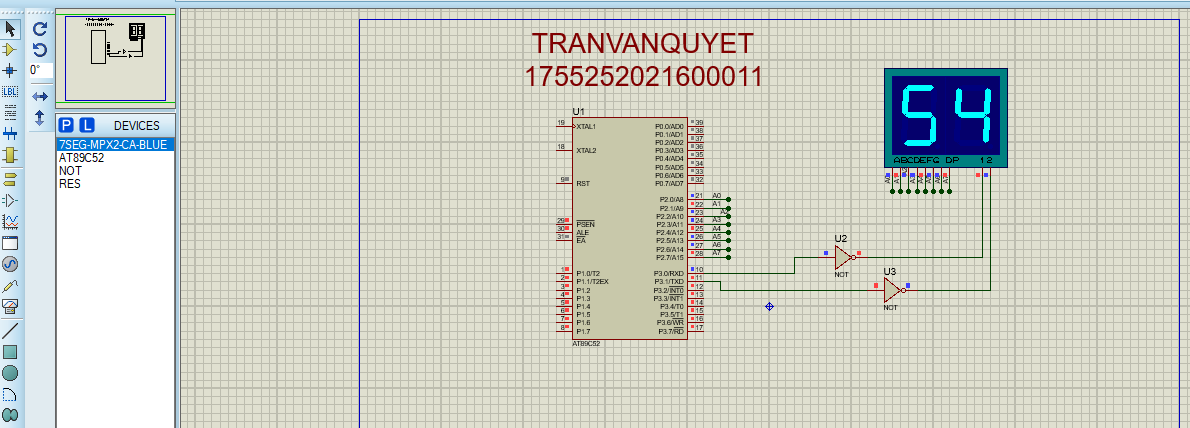
L7S2 = 0;

while(1);

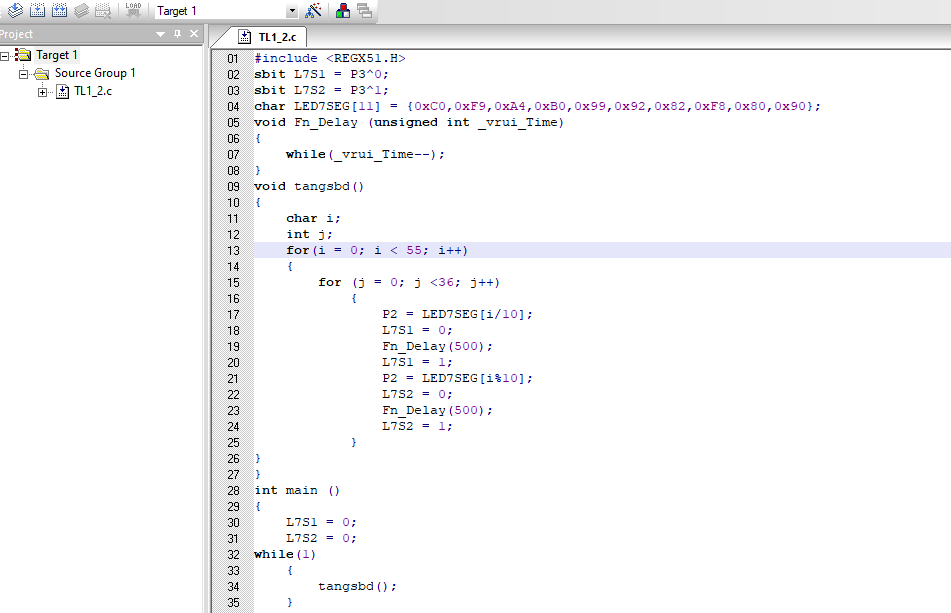
}

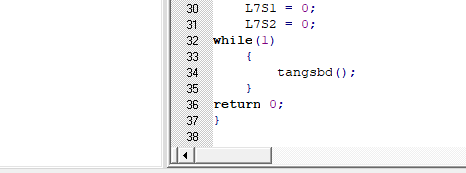
* 1. SBD = 34 + 20 = 54, tăng số đếm mỗi giây là 500ms ( sử dụng timer để định thời gian).

**Mô phỏng protues**



**Code trên keil c**

****

****

**Code đánh tay**

#include <REGX51.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 < 55; i++)

{

for (j = 0; j <36; 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;

}

# **Câu 2: sử dụng vi điều khiển AT89C52, mô phỏng trên phần mềm protues và lập trình thực hiện các nhiệm vụ sau:**

* 1. **cấu hình ngắt ngoài INT0 ở chế độ ngắt sườn xuống**
  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).**

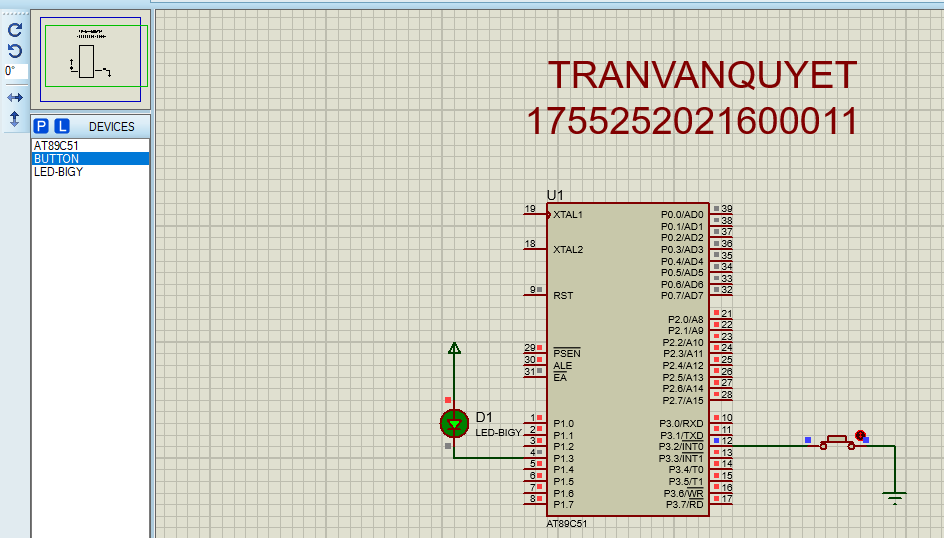
**Bài làm**

**Ngắt :** ngắt là một số sự kiện khẩn cấp bên trong hay bên ngoài của bộ vi điều khiển xảy ra, buộc vi điều khiển tạm dừng thực hiện chương trình hiện tại, phục vụ ngay lập tức mà nhiệm vụ mà ngắt yêu cầu.

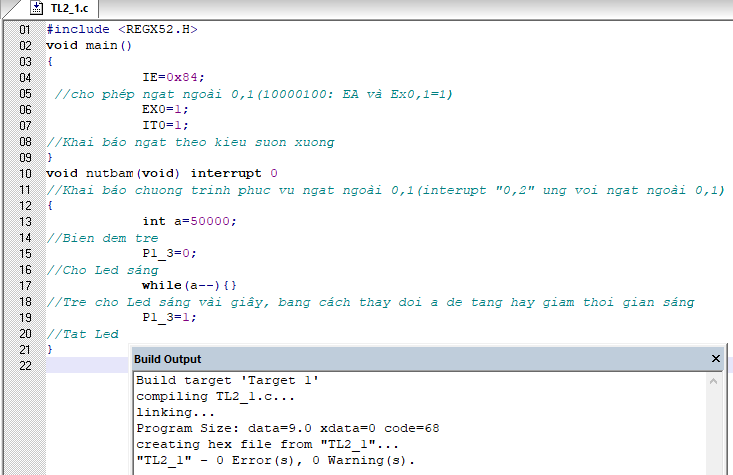
**Ngắt theo sườn:** là ngắt sẽ xảy ra khi có một sườn âm xuất hiện trên các chân ngắt của vi điều khiển, điều bày làm cho ngắt theo sườn khắc phục được nhược điểm của ngắt theo mức.

* 1. khi bấm nút nhấn đèn sẽ sáng, thể hiện ngắt ngoài INT0 chế độ ngắt sườn xuống.

**Mô phỏng protues**

****

**Code trên keil c**

****

**Code viết tay**

#include <REGX52.H>

void main()

{

IE=0x84;

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

EX0=1;

IT0=1;

//Khai báo ngắt theo kiểu sườn xuống

}

void nutbam(void) interrupt 0

//Khai báo chương trình phục vụ ngắt ngoài 0,1(interupt "0,2" ứng với ngắt ngoài 0,1)

{

int a=50000;

//Biến đếm trễ

P1\_3=0;

//Cho Led sáng

while(a--){}

//Tre cho Led sáng vài giây, bằng cách thay đổi a để tăng hay giảm thời gian sáng

P1\_3=1;

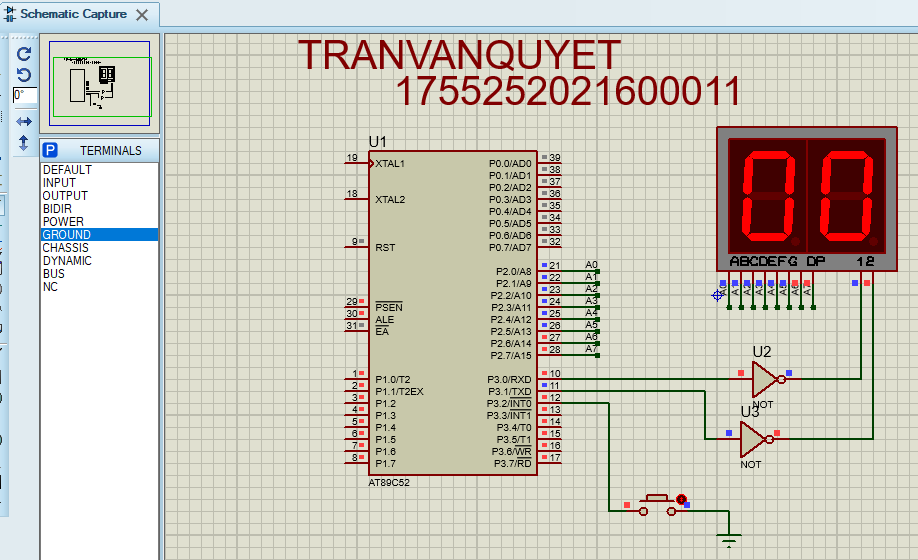
//Tat Led

}

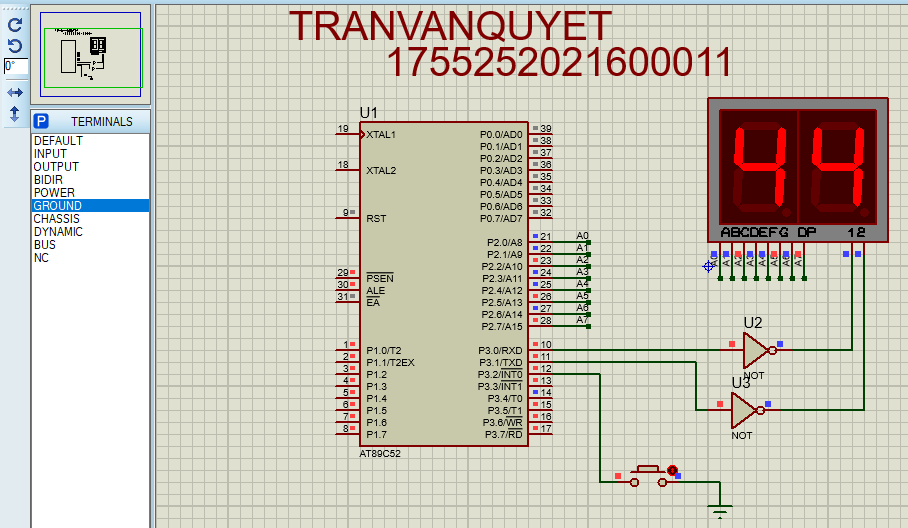
* 1. số báo danh là 34 nên sẽ là 34 + 10 = 44, đếm số lần nút bấm vào chân INT0 được bấm

**mô phỏng protues**

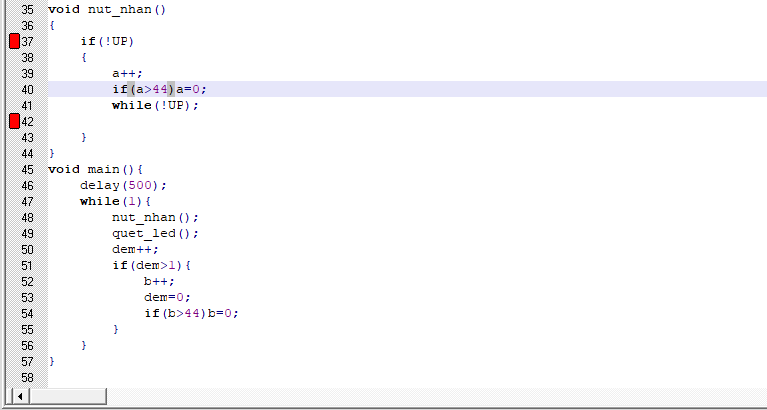
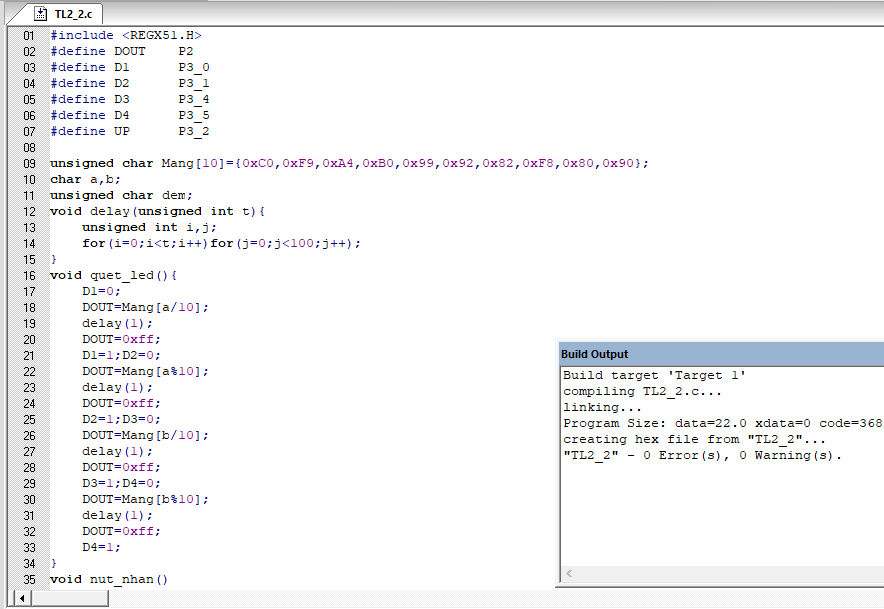
chưa nhấp



Nhấp 44 lần



**Code trên keil c**

****

**Code đánh tay**

#include <REGX51.H>

#define DOUT P2

#define D1 P3\_0

#define D2 P3\_1

#define D3 P3\_4

#define D4 P3\_5

#define UP P3\_2

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

char a,b;

unsigned char dem;

void delay(unsigned int t){

unsigned int i,j;

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

}

void quet\_led(){

D1=0;

DOUT=Mang[a/10];

delay(1);

DOUT=0xff;

D1=1;D2=0;

DOUT=Mang[a%10];

delay(1);

DOUT=0xff;

D2=1;D3=0;

DOUT=Mang[b/10];

delay(1);

DOUT=0xff;

D3=1;D4=0;

DOUT=Mang[b%10];

delay(1);

DOUT=0xff;

D4=1;

}

void nut\_nhan()

{

if(!UP)

{

a++;

if(a>44)a=0;

while(!UP);

}

}

void main(){

delay(500);

while(1){

nut\_nhan();

quet\_led();

dem++;

if(dem>1){

b++;

dem=0;

if(b>44)b=0;

}

}

}

# **Câu 3: sử dụng vi điều khiển AT89C52 thực hiện các nhiệm vụ sau:**

**3.1 vẽ sơ đồ mô phỏng trên protues ghép nối với LCD theo chế độ 4 bit, hiển thị họ tên, mã số sinh viên lên LCD.**

**3.2 vẽ sơ đồ mô phỏng trên protues, lập trình hiển thị họ và tên và mã số sinh viên qua chuẩn truyền thông UART.**

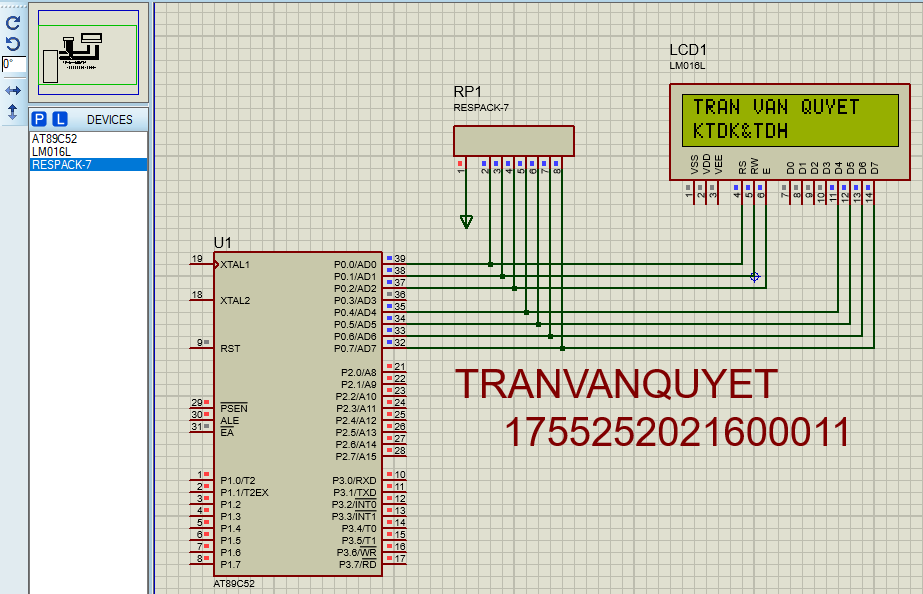
**Bài làm**

**LCD:** màn hình LCD hay còn gọi là màn hình tinh thể lỏng, được cấu tạo bởi các điểm ảnh chứa tinh thể lỏng có thể thay đổi tính phân cực của ánh sáng hay thay đổi cường độ ánh sáng truyền qua.

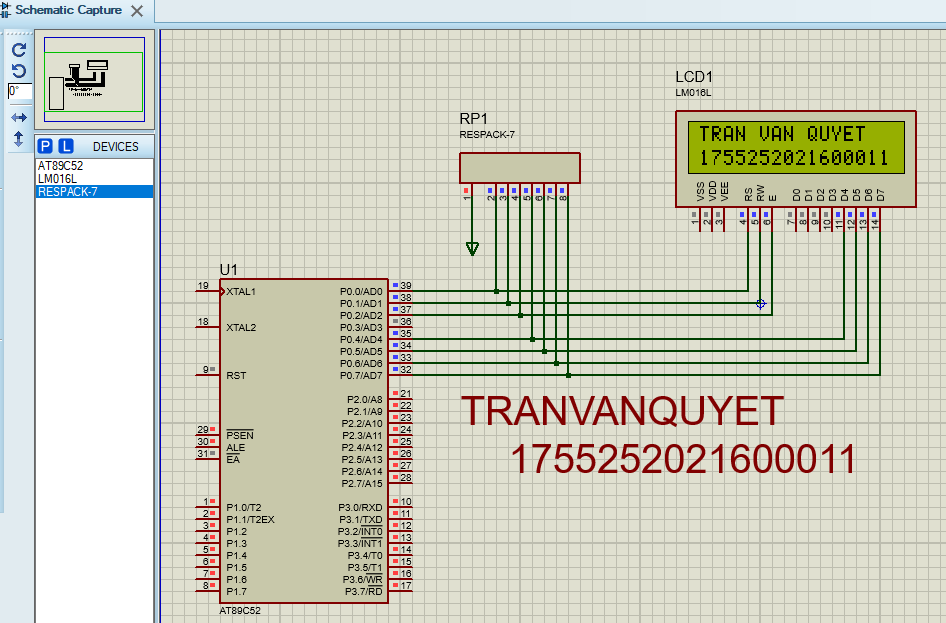
3.1

**Mô phỏng protues**

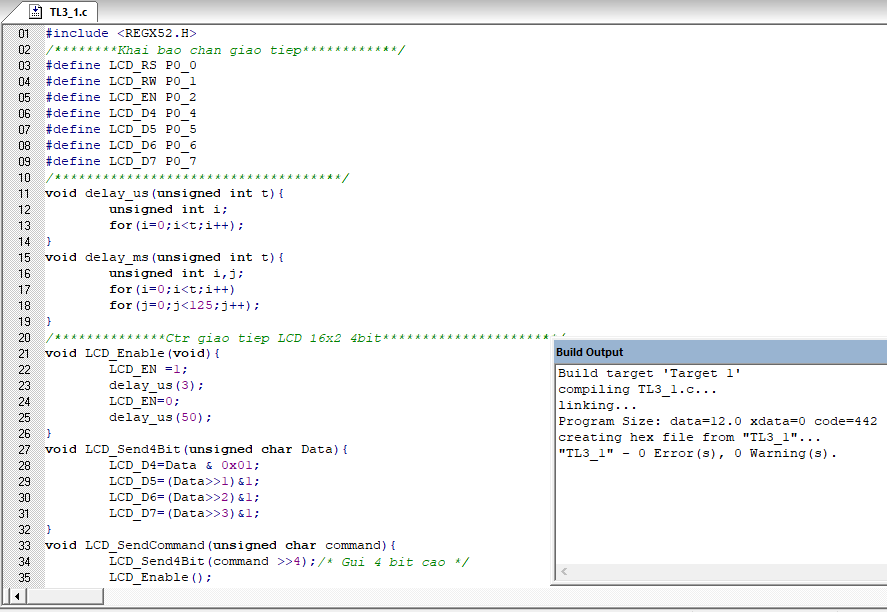
Hiển thị tên và lớp

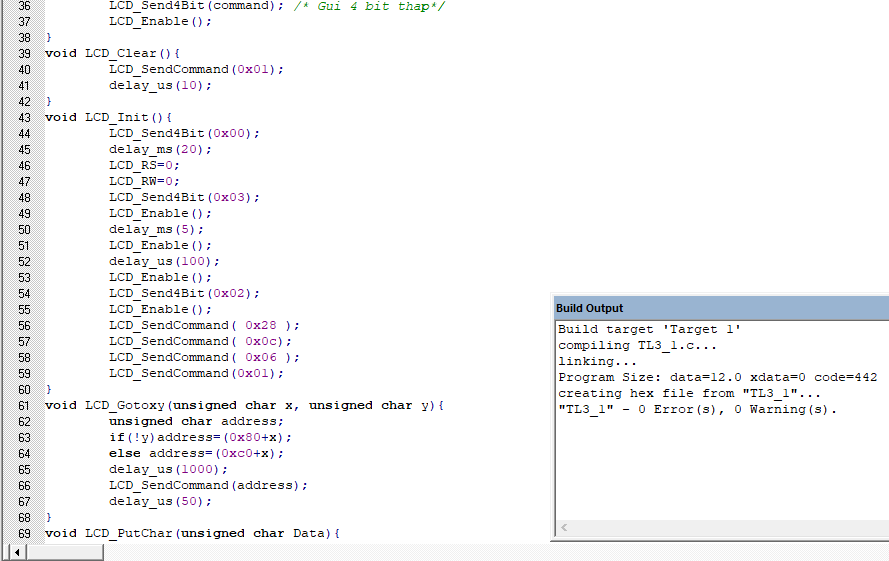
****

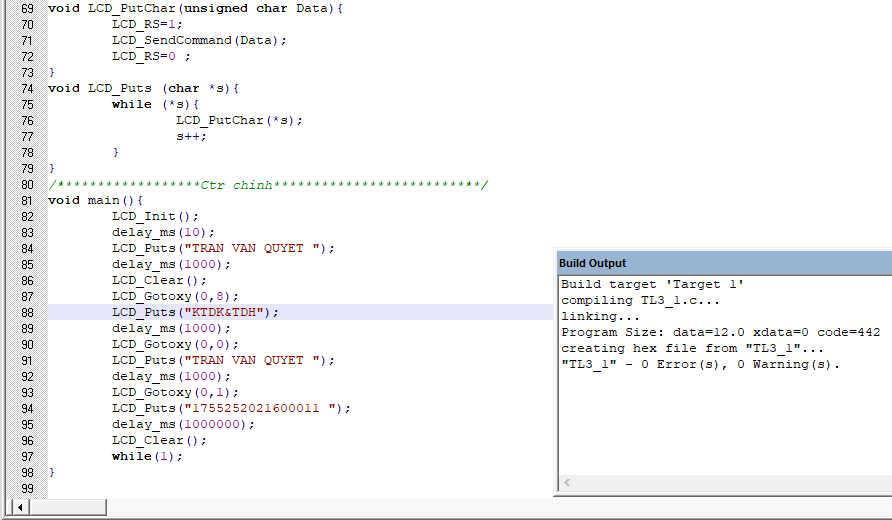
Hiển thị tên và mã số sinh viên



**Code trên keil c**

****

****

****

**Code đánh tay**

#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);

}

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;

}

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(){

LCD\_SendCommand(0x01);

delay\_us(10);

}

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 );

LCD\_SendCommand( 0x0c);

LCD\_SendCommand( 0x06 );

LCD\_SendCommand(0x01);

}

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){

LCD\_RS=1;

LCD\_SendCommand(Data);

LCD\_RS=0 ;

}

void LCD\_Puts (char \*s){

while (\*s){

LCD\_PutChar(\*s);

s++;

}

}

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

void main(){

LCD\_Init();

delay\_ms(10);

LCD\_Puts("TRAN VAN QUYET ");

delay\_ms(1000);

LCD\_Clear();

LCD\_Gotoxy(0,8);

LCD\_Puts("KTDK&TDH");

delay\_ms(1000);

LCD\_Gotoxy(0,0);

LCD\_Puts("TRAN VAN QUYET ");

delay\_ms(1000);

LCD\_Gotoxy(0,1);

LCD\_Puts("1755252021600011 ");

delay\_ms(1000000);

LCD\_Clear();

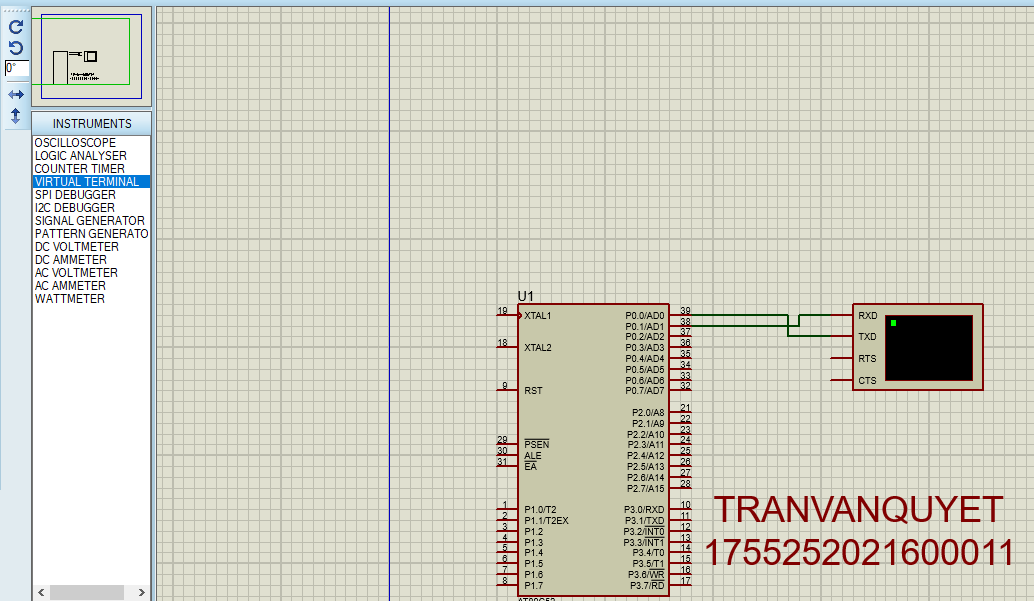
while(1);

}

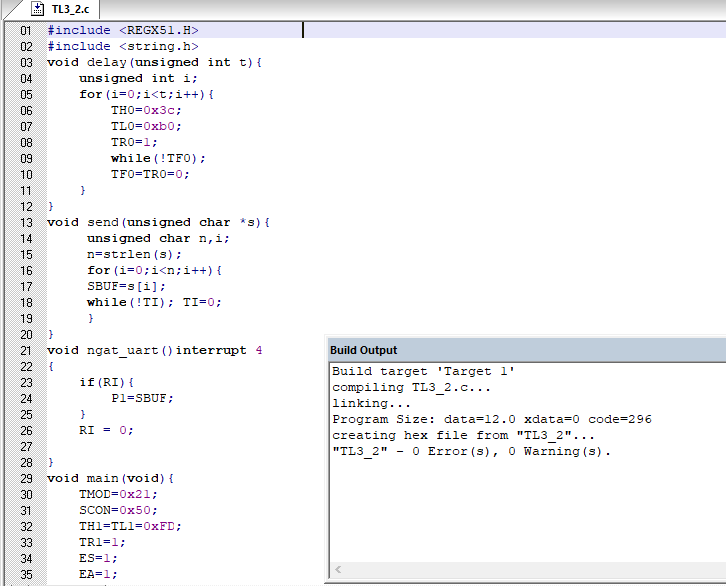
3.2

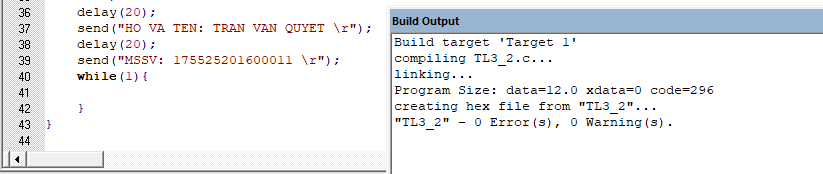
**UART:** là một vi mạch sẵn có trong một điều khiển nhưng không giống như một giao thức truyền thông, chức năng chính của nó là truyền dữ liệu nối tiếp, giao tiếp giữa hai thiết bị có thể được thực hiện theo hai cách là giao tiếp dữ liệu nối tiếp và giao tiếp dữ liệu song song. Sơ đồ khối UART bao gồm hai thành phần là máy phát và máy thu được hiển thị bên dưới.

**Mô phỏng protues**

****

**Code trên keil c**

****

****

**Code đánh tay**

#include <REGX51.H>

#include <string.h>

void delay(unsigned int t){

unsigned int i;

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

TH0=0x3c;

TL0=0xb0;

TR0=1;

while(!TF0);

TF0=TR0=0;

}

}

void send(unsigned char \*s){

unsigned char n,i;

n=strlen(s);

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

SBUF=s[i];

while(!TI); TI=0;

}

}

void ngat\_uart()interrupt 4

{

if(RI){

P1=SBUF;

}

RI = 0;

}

void main(void){

TMOD=0x21;

SCON=0x50;

TH1=TL1=0xFD;

TR1=1;

ES=1;

EA=1;

delay(20);

send("HO VA TEN: TRAN VAN QUYET \r");

delay(20);

send("MSSV: 175525201600011 \r");

while(1){

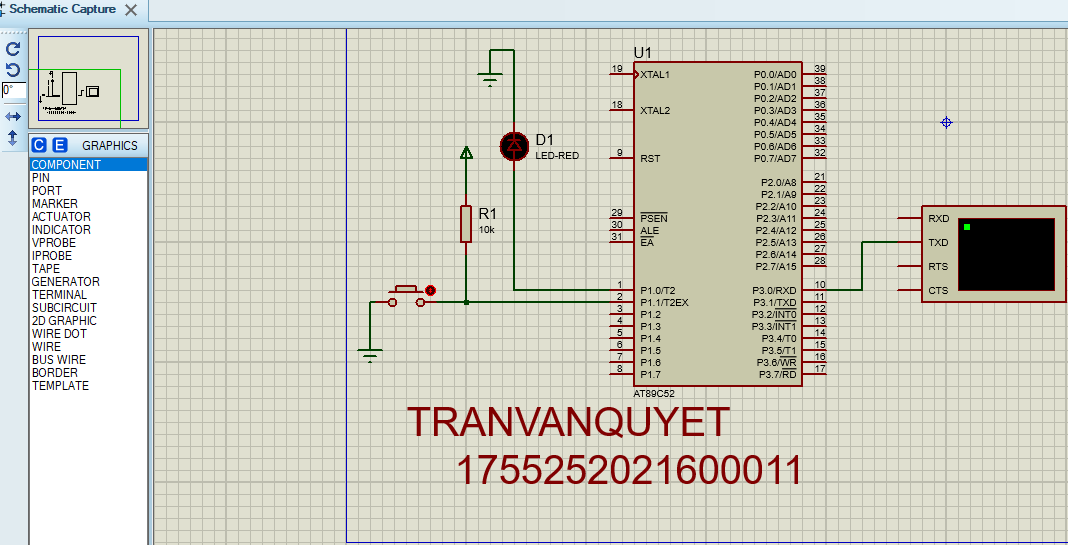
}

}

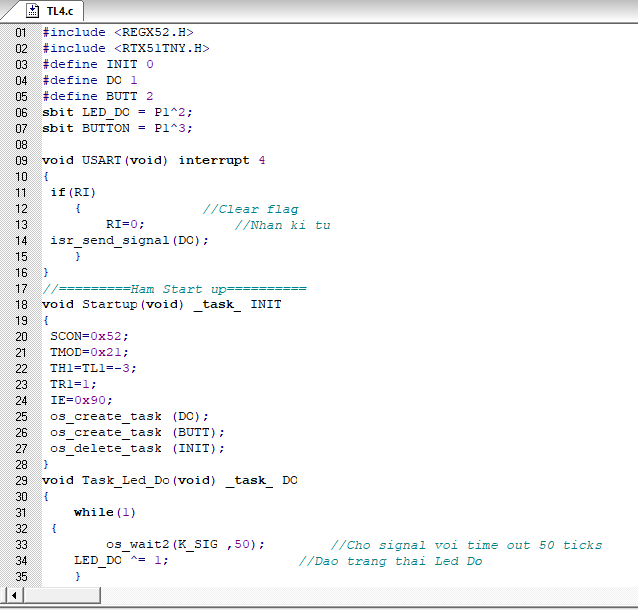
# **Câu 4: sử dụng vi điều khiển AT89C52 vẽ sơ đồ mô phỏng protues ghép nối với LED D1qua cổng P1.2, BUTTON B1 qua cổng P1.3. sử dụng hệ điều hành RTX51 lập trình ngắt USART, tast BUTTON, tast LED. Thực hiện gửi “signal” từ ngắt USART và tast BUTTON đến tast LED. Tast LED thực hiện đảo trạng thái của LED D1 khi nhận được tín hiệu tast khác gửi tới.**

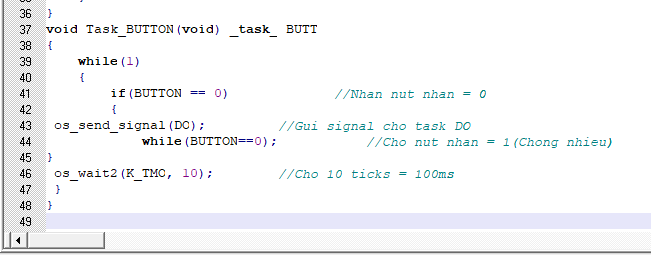
**Bài làm:**

**Mô phỏng protues**

****

**Code trên keil c**

****

****

**Code đánh tay**

#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

}

}