

TRƯỜNG ĐẠI HỌC VINH
VIỆN KỸ THUẬT CÔNG NGHỆ



BÁO CÁO TIỂU LUẬN
HỆ THỐNG NHÚNG

GIẢNG VIÊN HƯỚNG DẪN : TS . MAI THẾ ANH

SINH VIÊN : PHAN TUẤN ĐỨC

MSSV : 1755252021600018

Nghệ An 2021

NỘI DUNG CÁC CÂU HỎI:

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

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;
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).

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

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, tắt BUTTON tắt LED. Thực hiện gửi “signal” từ ngắt USART và task BUTTON đến task LED. Task LED thực hiện đảo trạng thái của Led D1 khi nhận được tín hiệu task khác gửi tới.

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 điểm, nhược điểm và ứng dụng của hệ điều hành thời gian thực trong thiết kế các hệ thống nhúng.

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:

```
#include <REG8253.H>
```

```
sbit L7S1 = P3^0; sbit
```

```
L7S2 = P3^1; #define
```

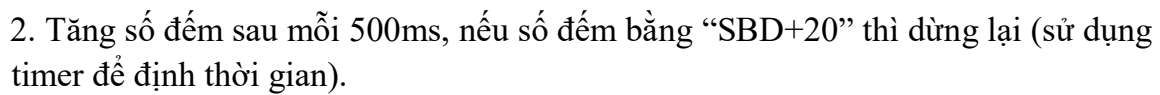
```
LED7SEG P2; int
```

```
main()
```

```
{    P2= 0xC0;
```

```
    L7S1=0;
```

-Mô phỏng:



- Số báo danh Trần Văn Hiệp: 13 -> 13+20 =33.

-Code:

```
#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 < 34; i++)
    {
        for (j = 0; j < 40; j++)
        {
            P2 = LED7SEG[i/10];
            L7S1 = 0;
            Fn_Delay(500);
            L7S1 = 1;
            P2 = LED7SEG[i%10];
            L7S2 = 0;
            Fn_Delay(500);
            L7S2 = 1;
        }
    }
}

```

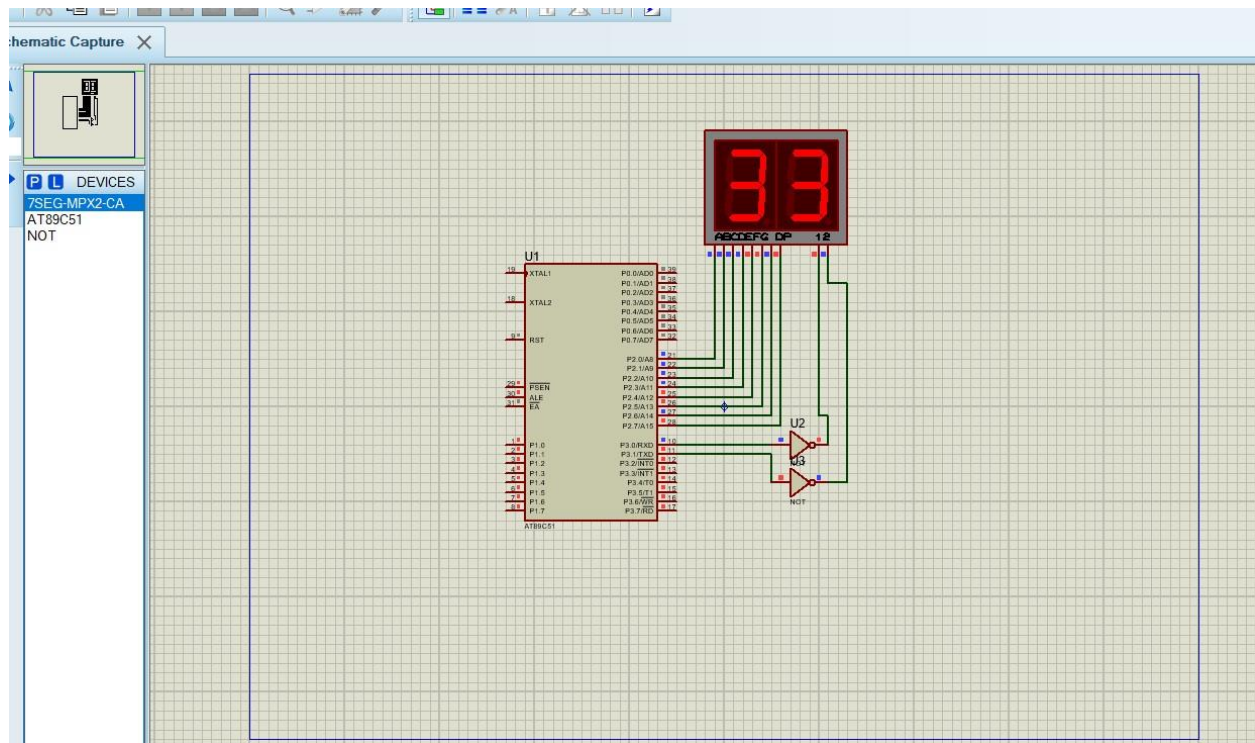
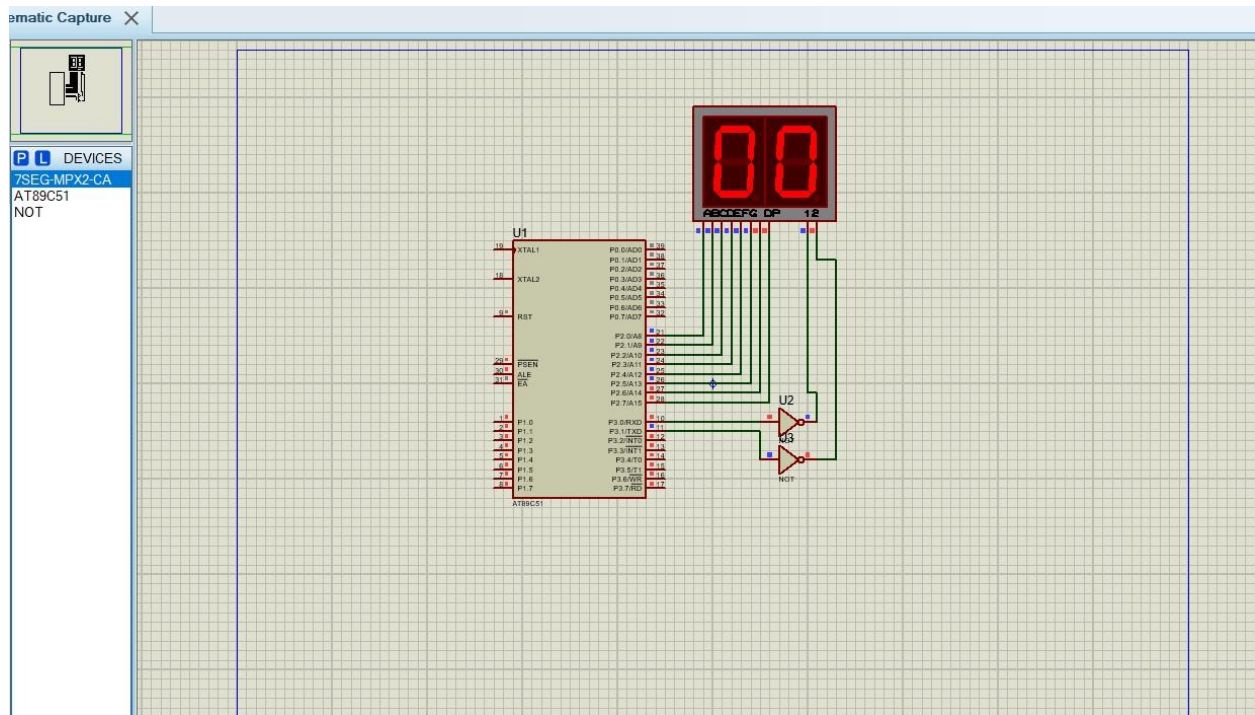
```

int main () {
    L7S1 = 0;
    L7S2 = 0;
    while(1)
    {
        tangsbd();
    }
}

```

```
    }  
    return 0;  
}
```

-Mô phỏng:



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:

-Mô phỏng:

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).

Số lần bấm = sbd+10=13+10=23 -Code:

```
#include <REGX51.H>
```

```
#define DOUT      P2    //Chan xuất data led 7
```

```
#define D1        P3_0  //Quét LED số 1
```

```
#define D2        P3_1  //Quét LED số 2
```

```
#define D3        P3_4  //Quét LED số 3
```

```
#define D4        P3_5  //Quét LED số 4
```

```
#define UP        P3_2  //Nút nhấn 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 tạo trễ
```

```
unsigned int i,j;
```

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

```
}
```

```
void quet_led(){ //Ham quét 4 LED 7
```

```
    D1=0; //Led số 1 sáng
```

```
    DOUT=Mang[a/10]; //Xuất data gtri hàng chục của biến a
```

```
    delay(1); //Tạo trễ
```

```
    DOUT=0xff; //Ngắt data (tắt bóng mờ với led đơn xếp thành led 7)
```

```
    D1=1;D2=0; //Led số 1 tắt, led 2 sáng
```

```
    DOUT=Mang[a%10]; //Xuất data gtri hàng đơn vị của biến a
```

```
    delay(1);
```

```
    DOUT=0xff;
```

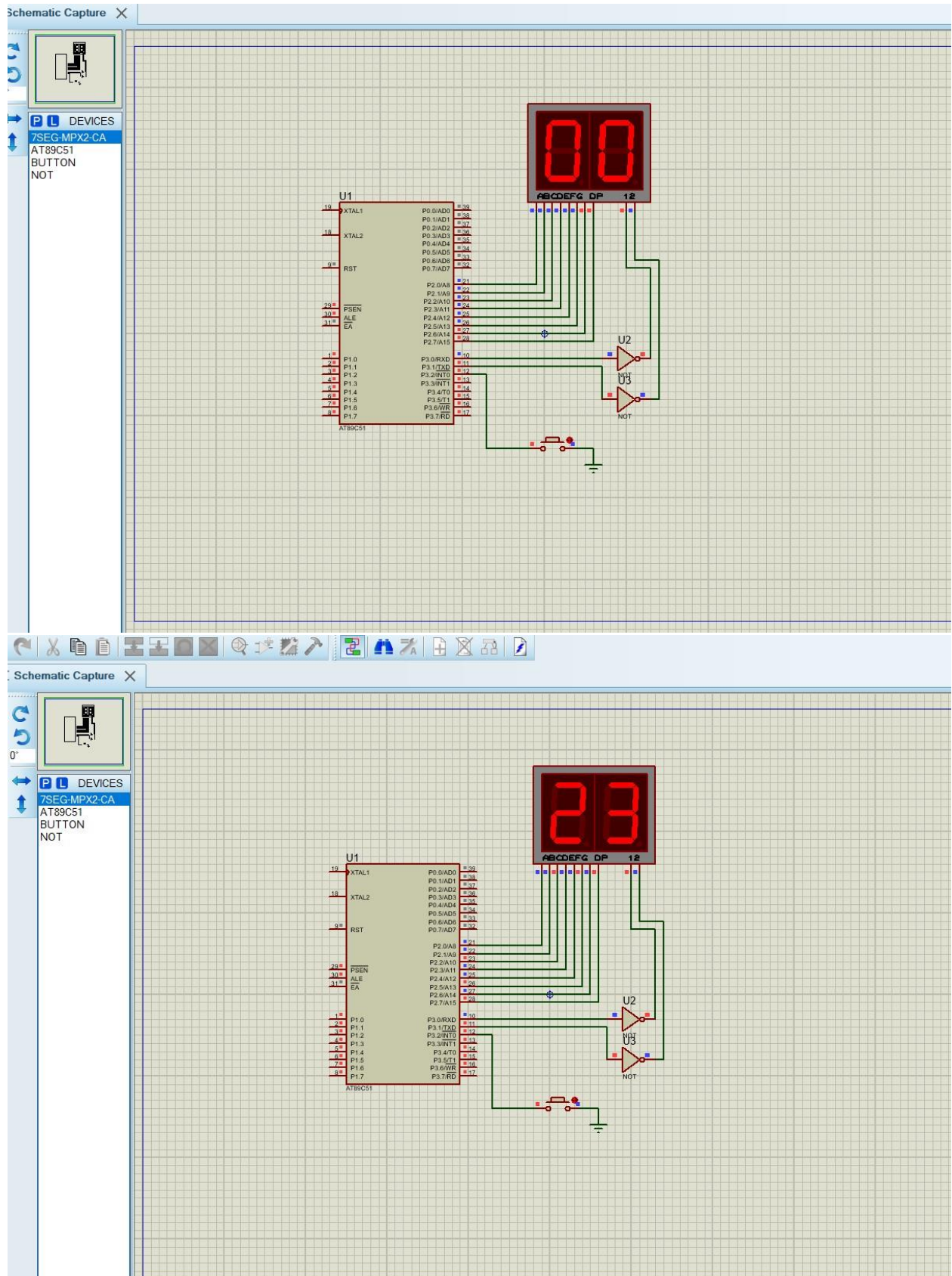
```
    D2=1;D3=0; //Led số 2 tắt, led 3 sáng
```

```

        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>23) a=0; //Set
        ve 0 khi a>23                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>23) b=0; //Set lai gia tri
            }
        }
    }
}

```


-Mô phỏng:



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:

```
#include <REGX51.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 chuoai ky
tu while (*s){
        LCD_PutChar(*s);
s++;
        }
}

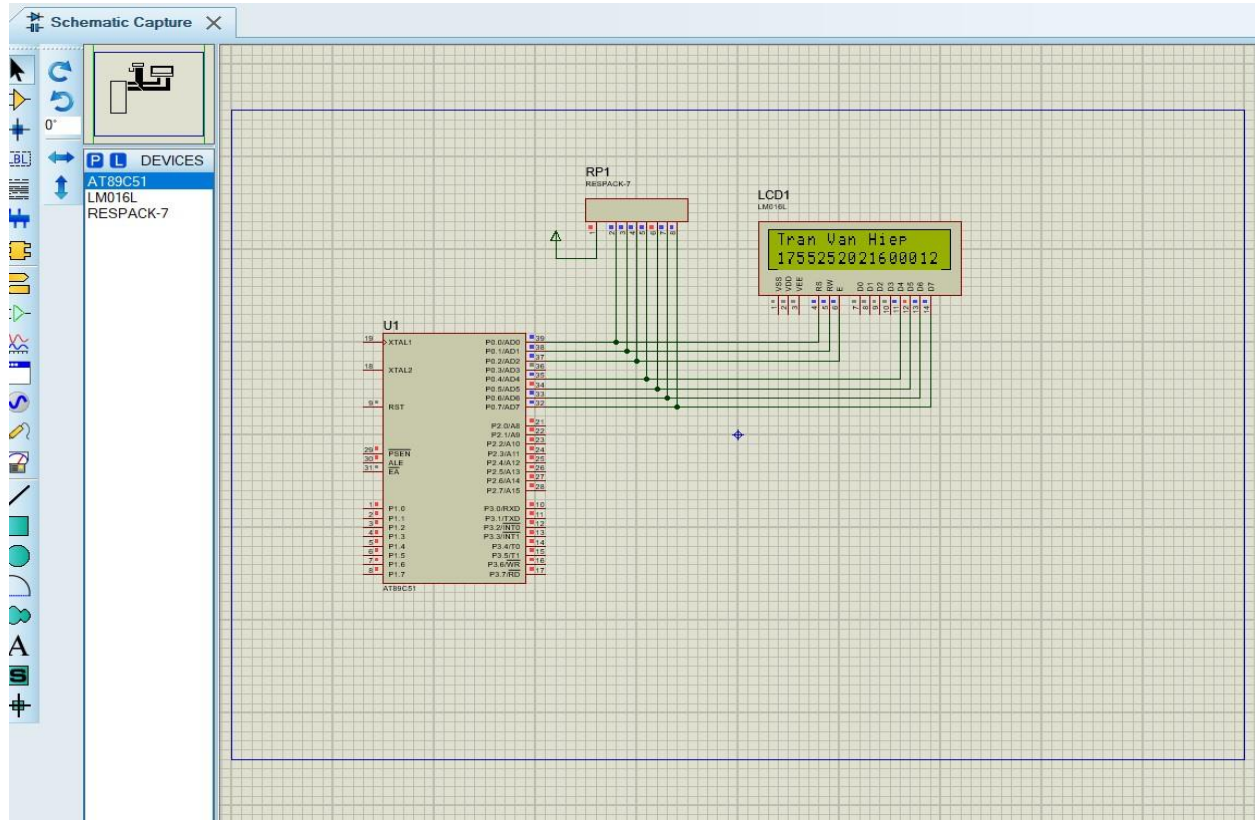
```

```

/*****Ctr chinh*****/ void
main(){
    LCD_Init();//Khoi tao LCD
    delay_ms(10);
    LCD_Puts("Tran Van Hiep ");//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("Tran Van Hiep");
    delay_ms(1000);
        LCD_Gotoxy(0,1);//Tro toi vi tri
    LCD_Puts("1755252021600012");
        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:**

```

#include <REGX51.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

```



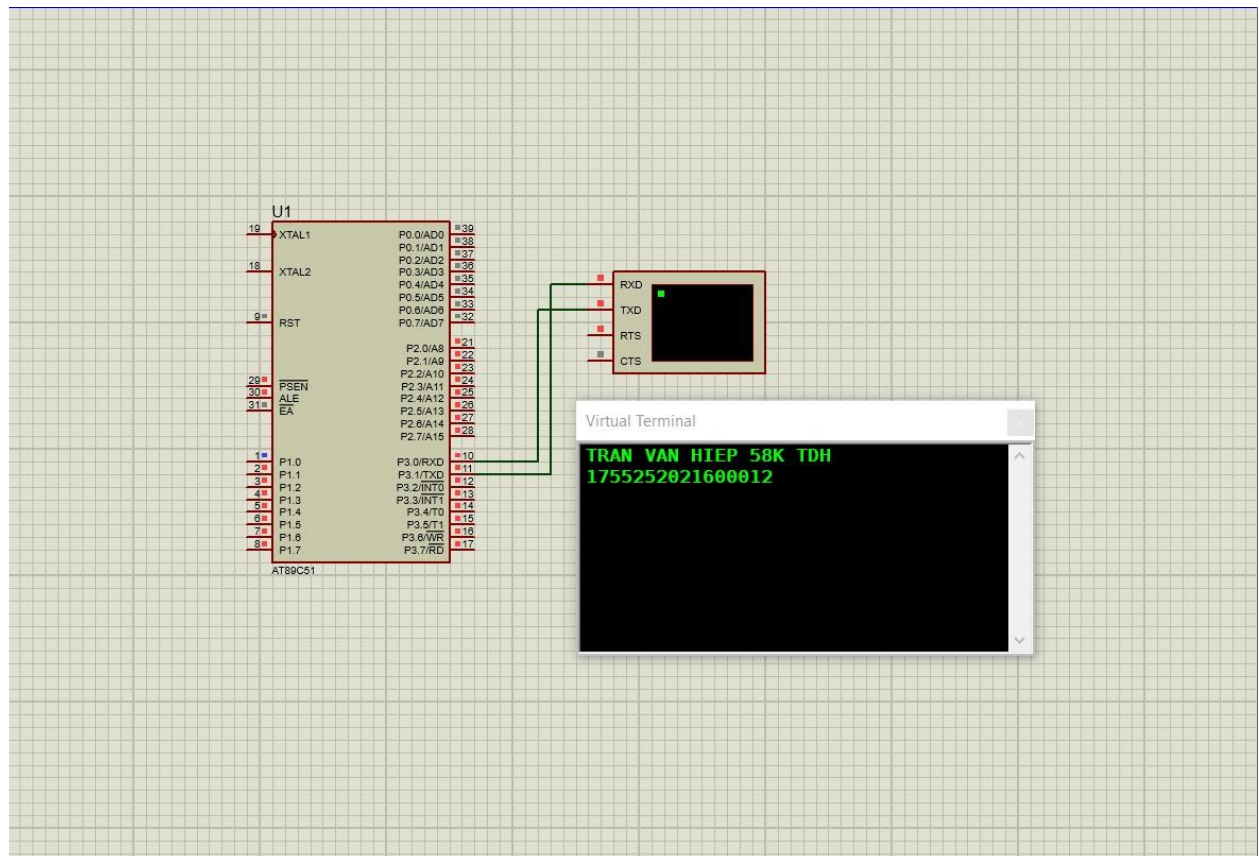
```

    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("TRAN VAN
HIEP 58K TDH \r");        delay(20);
send("175525201600012 \r");    while(1){
        //ko lam gi ca
    }
}
}

```

-Mô phỏng:



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 <REGX51.H>

#include <RTX51TNY.H>    //Sử dụng thư viện RTX51 Tiny Real-Time

#define INIT 0           //Định nghĩa INIT = 0
#define DO 1             //Định nghĩa DO = 1
#define BUTT 2           //Định nghĩa BUTTTON = 2 sbit
LED_DO = P1^2;          //Định nghĩa chân LED_DO sbit
BUTTON = P1^3;           //Định nghĩa chân BUTTON

void USART(void) interrupt 4 //Ngắt nhận USART
{ if(RI)                 //Flag nhận được ki
tu
{
    //Clear flag
    RI=0; //Nhận ki tu    isr_send_signal(DO);
//Gửi signal cho task DO
}
}
```

```

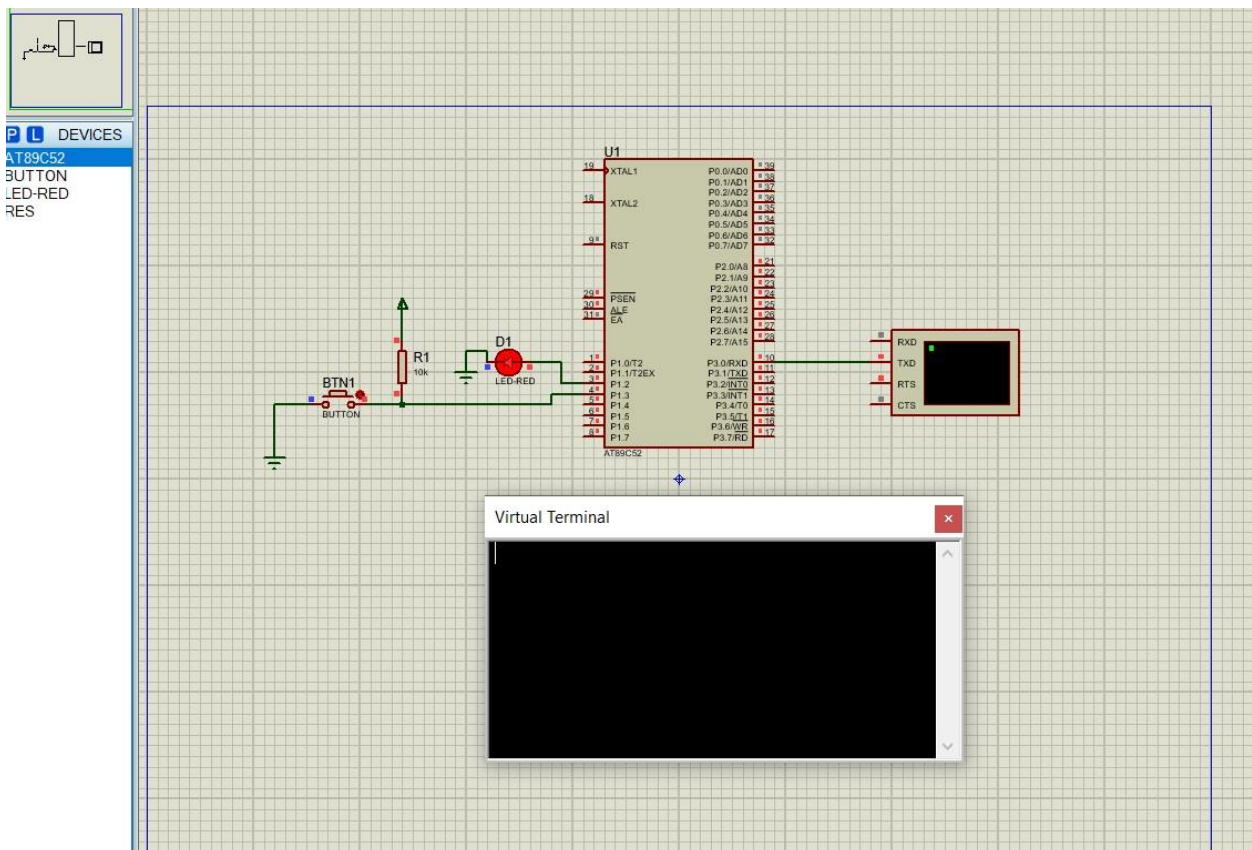
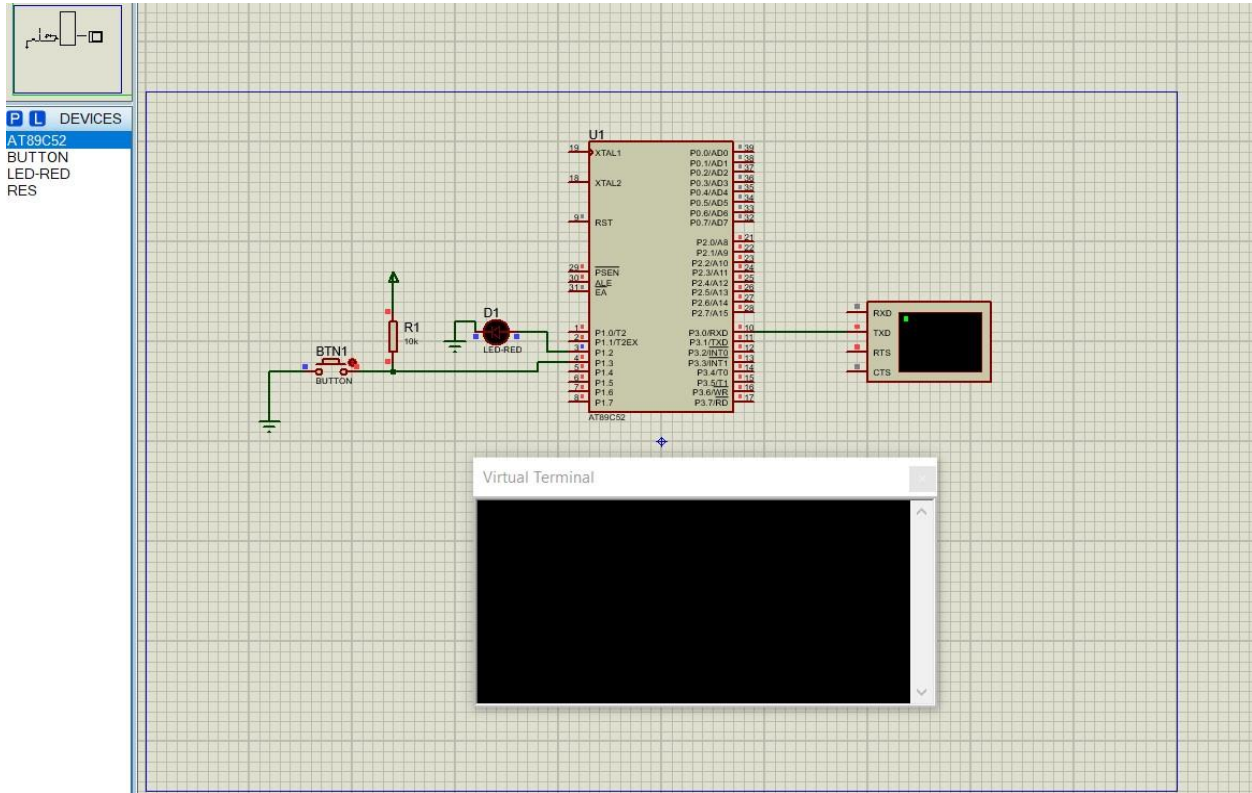
//=====Ham Start up===== void
Startup(void) _task_ INIT
{
    SCON=0x52;          //USART che do 1
    TMOD=0x21;          //Timer 1 mode 2
    TH1=TL1=-3;         //baudrate 9600
    TR1=1;
    IE=0x90;            //Ngat USART  os_create_task
(DO);    //Tao Task_Led_Do  os_create_task
(BUTT);    //Tao Task BUTTON  os_delete_task
(INIT);    //Xoa Task hien tai (Task 0)
} 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:

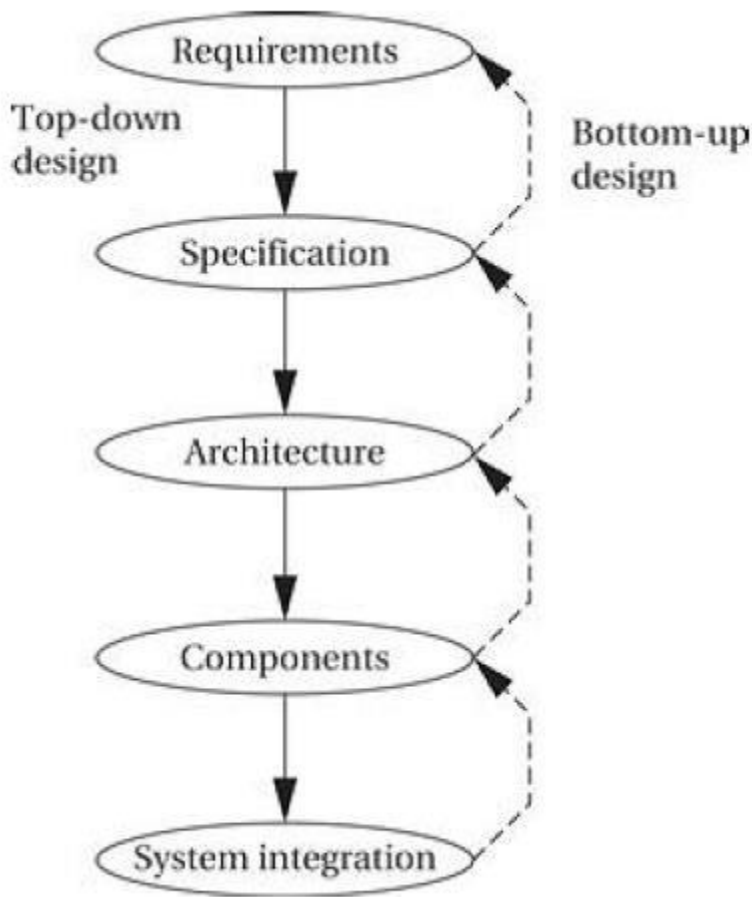


Câu 5. Hãy trình bày:

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

Một phương pháp thiết kế rất quan trọng bởi ba lý do như sau. Đầu tiên, nó cho phép chúng ta giữ một bảng ghi chú về một thiết kế để đảm bảo rằng chúng ta đã làm mọi thứ chúng ta cần làm, chẳng hạn như có cần tối ưu hóa hiệu năng hay không hoặc khi nào thực hiện kiểm tra chức năng của thiết kế. Thứ hai, nó cho phép chúng ta phát triển các công cụ thiết kế được hỗ trợ bởi máy tính. Phát triển một công cụ duy nhất để hỗ trợ toàn bộ quá trình thiết kế một hệ thống máy tính nhúng sẽ là một 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 (hoặc ít nhất là bán tự động hóa) quá trình thiết kế cho mỗi bước trong quy trình. 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 một quy trình thiết kế tổng thể, các thành viên trong nhóm có thể dễ dàng hiểu những gì họ phải làm, những gì họ sẽ nhận được từ các thành viên khác trong nhóm vào những thời điểm nhất định và những gì họ phải bàn giao khi hoàn thành các bước được giao. Bởi vì phần lớn các hệ thống máy tính nhúng được thiết kế và thực hiện bởi các nhóm, sự phối hợp nhịp nhàng giữa các nhóm thể hiện vai trò quan trọng nhất trong phương pháp thiết kế.

Hình 1-2 tóm tắt các bước chính trong quy trình thiết kế hệ thống nhúng. Trong luồng thiết kế từ trên xuống, chúng ta bắt đầu với việc đưa ra các yêu cầu thiết kế (Requirements) của hệ thống. Trong bước tiếp theo, đặc tả kỹ thuật (specification), chúng ta tạo ra một bản mô tả chi tiết hơn về những gì chúng ta muốn thiết kế. Tuy nhiên, bản đặc tả chỉ nêu ra cách hệ thống hoạt động mà không phải cách nó được xây dựng thế nào. Các chi tiết bên trong của hệ thống bắt đầu hình thành khi chúng ta phát triển kiến trúc – một sơ đồ khối chỉ ra cấu trúc của hệ thống dưới dạng các khối chức năng. Khi chúng ta xác định các khối chức năng chính cấu thành lên hệ thống chúng ta có thể thiết kế các khối chức năng đó, bao gồm cả các khối chức năng phần mềm và bất kỳ khối chức năng phần cứng chuyên dụng nào mà chúng ta cần. Dựa trên các khối chức năng đó, cuối cùng chúng ta có thể tích hợp chúng lại với nhau để xây dựng một hệ thống hoàn chỉnh.



Hình 1-2 Các bước chính trong quy trình thiết kế hệ thống máy tính nhúng. Nhưng các bước trong quy trình thiết kế chỉ là một trục mà chúng ta có thể xem xét quá trình thiết kế hệ thống máy tính nhúng. Bên cạnh đó chúng ta cũng cần xem xét thêm các mục tiêu chính của thiết kế:

- Chi phí sản xuất
- Hiệu năng (bao gồm tốc độ tính toán và thời hạn một tính toán phải hoàn thành) • Sự tiêu thụ năng lượng.

Chúng ta cũng cần xem xét các nhiệm vụ mà chúng ta cần thực hiện ở mọi bước trong thiết kế, cụ thể như:

- Chúng ta phải phân tích thiết kế ở mỗi bước để xác định cách chúng ta có thể thỏa mãn các yêu cầu kỹ thuật trong bản đặc tả
- Sau đó chúng ta phải tinh chỉnh thiết kế để đi sâu vào thiết kế chi tiết
- Chúng ta cần kiểm thử thiết kế để đảm bảo rằng nó vẫn đáp ứng được tất cả các mục tiêu của hệ thống chẳng hạn như chi phí, tốc độ, ...

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.

