

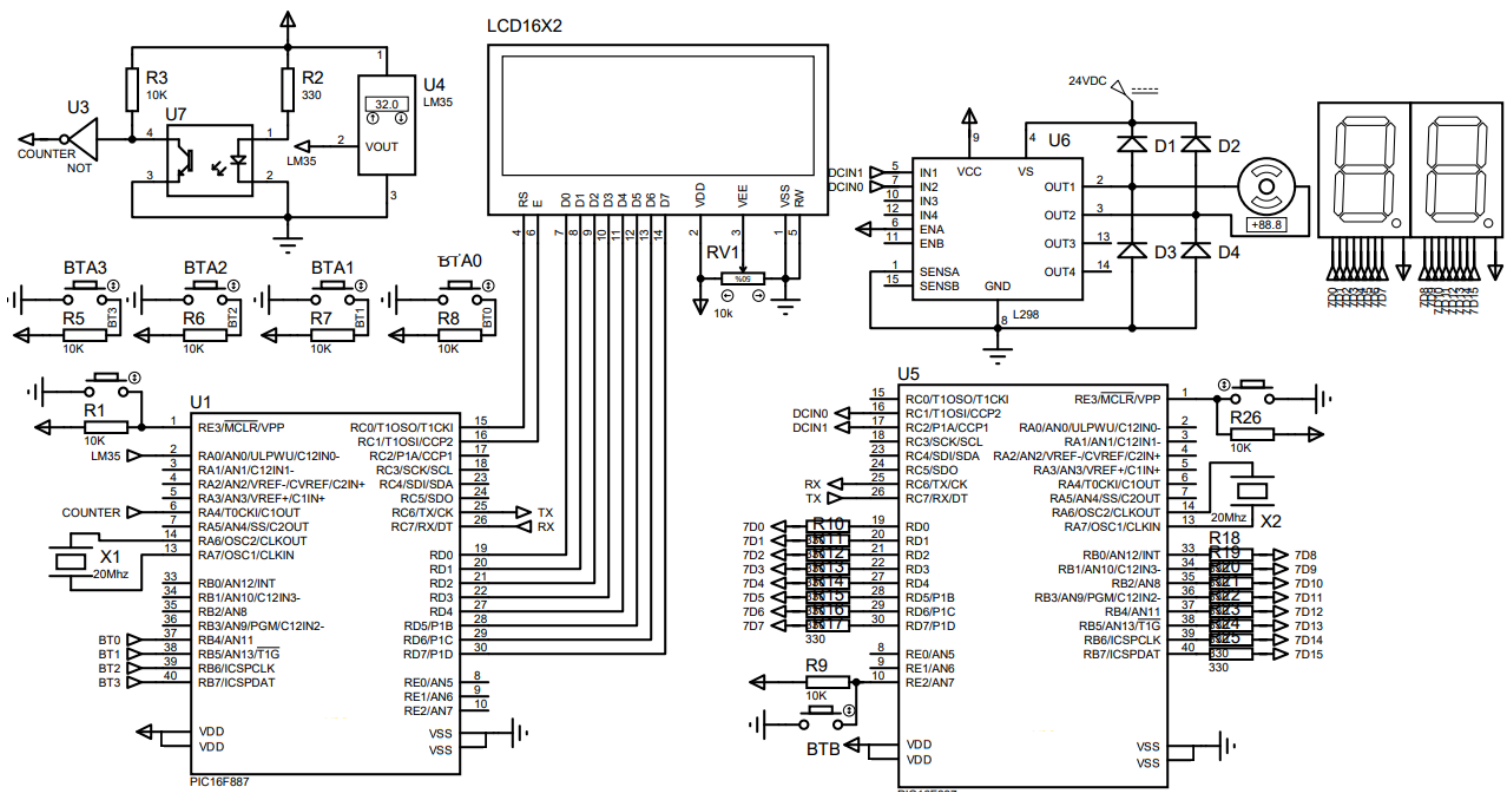
Câu 1 (4 đ). Thiết kế phần cứng

Hệ thống A (HTA) dùng vi điều khiển PIC16F887 có kết nối với các ngoại vi như sau: 1 cảm biến nhiệt LM35 để đo nhiệt độ, một mạch cảm biến tạo xung để đếm sản phẩm dùng Timer0 và 1 LCD 16x2, 4 nút nhấn BTA0, BTA1, BTA2, BTA3.

Hệ thống B (HTB) dùng vi điều khiển PIC16F887 có kết nối với các ngoại vi như sau: 2 led 7 đoạn Anode chung theo phương pháp trực tiếp, một IC L298 để điều khiển 1 động cơ DC 24V-1A dùng PWM có chu kỳ là 0,8 ms và 1 nút nhấn BTB.

Cả 2 đều dùng tụ thạch anh có tần số 20Mhz, hai hệ thống giao tiếp với nhau theo chuẩn truyền dữ liệu UART, tốc độ 9600 baud.

- a. Hãy thiết kế sơ đồ nguyên lý kết nối hệ thống A và B tùy ý chọn port giao tiếp, chỉ vẽ các port sử dụng, các port không sử dụng thì không cần vẽ. (1.5+1.5đ)



//Mỗi lỗi trừ 0.25đ

- b. Hãy tính toán độ phân giải của ADC tùy ý chọn Vref+ và Vref-. (0.25đ)

$$\text{StepSize} = \frac{vref^+ - vref^-}{2^{n-1}}, \quad t^o = \frac{dout * \text{StepSize}}{0.01}$$

Nếu chọn n=10(độ phân giải ADC), $vref^+ = 5V$, $vref^- = 0V$

$$\Rightarrow \text{StepSize} = 0.004887 V$$

$$\Rightarrow t^o = dout * 0.4887 \text{ hoặc } t^o = \frac{dout}{2.046}$$

Nếu chọn $n=8$, $vref^+ = 5V$, $vref^- = 0V$

⇒ $StepSize = 0.0196 V$

⇒ $t^0 = dout * 1.96$ hoặc $t^0 = \frac{dout}{0.51}$

Trong đó : $dout$ là giá trị số đọc được từ ADC

- c. Hãy tính toán lựa chọn các thông số cho khối PWM1 để có thể điều khiển động cơ với 3 cấp tốc độ là 25%, 50% và 75%. (0.75đ)

$$T = 4(PR2+1) * T_{OSC} * T2_{PV}$$

Ta có: $T = 0.8ms$, $T_{OSC} = 1/20M$

⇒ Chọn $T2_{PV} = 16$

⇒ $PR2 = 249$

⇒ Hệ số chu kỳ tối đa ($HSCK_{max}$): $HSCK_{max} = (PR2+1) * 4 = 1000$

⇒ $HSCK_{25\%} = 25\% * 1000 = 250$

⇒ $HSCK_{50\%} = 50\% * 1000 = 500$

⇒ $HSCK_{75\%} = 75\% * 1000 = 750$

//Mỗi cấp độ đúng 0.25đ

Câu 2 (3.5 đ). Thiết kế phần mềm:

Sử dụng HTA ở câu 1 để viết một chương trình điều khiển theo các yêu cầu sau:

- Đo nhiệt độ dùng cảm biến LM35 hiển thị ở trên LCD, tùy ý chọn vị trí hiển thị và tùy ý chọn đo 1 lần hay đo trung bình nhiều lần. (1.25đ).
- Đếm sản phẩm hiển thị trên LCD cũng tùy ý chọn vị trí nhưng không trùng vị trí đã hiển thị nhiệt độ, khi giá trị đếm bằng 101 thì quay về 1 hoặc khi nhận mã xóa là 0xBB từ HTB thì xóa kết quả đang đếm về 0, có xóa số vô nghĩa. (1+0.25đ)
- Khi nhấn nút BTA0 thì gửi mã 0xA0 sang HTB, khi nhấn nút BTA1 thì gửi 0xA1 sang HTB, khi nhấn nút BTA2 thì gửi mã 0xA2 sang HTB, khi nhấn nút BTA3 thì gửi 0xA3 sang HTB. (1đ)

Tùy ý sử dụng thư viện đã học trong giáo trình.

```
#include<16f887.h>
#define adc = 10
#define hs
#define delay(clock=20M)
#define RS232(baud=9600, xmit = pin_c6, rcv= pin_c7)
#define lcd_rs pin_c0
#define lcd_e pin_c1
#define output_lcd output_d
#include<TV_LCD.c>
unsigned int8 nd,t0,nhan;
void main()
{
    set_tris_a(0x11); set_tris_b(0xf0);
    set_tris_c(0x80); set_tris_d(0);
    lcd_setup();
    setup_adc(adc_clock_div_32); //***** Cau 2a *****
    setup_adc_ports(san0|vss_vdd); //Khoi tao ADC dung 0.5d
    set_adc_channel(0); delay_us(200);
    setup_timer_0(T0_EXT_H_TO_L|T0_DIV_1); //***** Cau 2b *****
    set_timer0(0); //Khoi tao Counter dung 0.25d
    enable_interrupts(int_rda);
    enable_interrupts(global);
    while(true)
    {
```

```

nd = read_adc()/2.046; //***** Cau 2a *****
lcd_goto_xy(0,0); // Do nhiet do dung 0.5d
printf(lcd_data,"Nhiệt Độ: %u ",nd); // Hien thi nhiet do dung 0.25d

t0= get_timer0(); //***** Cau 2b *****
if(t0>=101) set_timer0(t0-101); // Dem dung san pham 0.5d
lcd_goto_xy(1,0);
printf(lcd_data,"San Pham: %u ",t0); // Hien thi va xoa 0VN 0.25d
if(input(pin_b4)==0) putc(0xA0); //***** Cau 2c *****
if(input(pin_b5)==0) putc(0xA1); // Moi nut dung 0.25d
if(input(pin_b6)==0) putc(0xA2);
if(input(pin_b7)==0) putc(0xA3);
}
}
#int_rda //***** Cau 2b *****
void ngatnhan()
{
    nhan = getc();
    if(nhan==0xBB) set_timer0(0); // nhan 0xBB reset ket qua dem 0.25d
}

```

Câu 3 (2.5 đ). Thiết kế phần mềm:

Sử dụng HTB ở câu 1 để viết một chương trình điều khiển theo các yêu cầu sau:

- Đếm giây chính xác dùng Timer1 và dùng ngắt, hiển thị ở 2 led 7 đoạn (1.25đ).
- Khi nhận mã 0xA0 từ HTA thì điều khiển động cơ DC ngừng, nhận mã 0xA1 thì quay thuận 25%, nhận mã 0xA2 thì quay thuận 50%, nhận mã 0xA3 thì quay thuận 75%. (1.đ)
- Khi nhấn nút BTB thì HTB gửi mã 0xBB sang HTA. (0.25đ)

Tùy ý sử dụng thư viện đã học trong giáo trình.

```

#include<16f887.h>
#fuses hs
#use delay(clock=20M)
#use RS232(baud=9600, xmit = pin_c6, rcv= pin_c7)
const unsigned int8 m7d[]={0xc0,0xf9,0xa4,0xb0,0x99,0x92,0x82,0xf8,0x80,0x90};
unsigned int8 bdn=0, giay=0;
unsigned int16 nhan=0xA0;
void main()
{
    set_tris_b(0);set_tris_c(0x80);
    set_tris_d(0);set_tris_e(0x04);
    setup_timer_1(t1_internal|t1_div_by_8); //***** Cau 3a *****
    set_timer1(3036); //Khoi tao timer1 dung 0.5d
    enable_interrupts(int_timer1);
    setup_timer_2(t2_div_by_16,249,1); //***** Cau 3b *****
    setup_ccp1(ccp_pwm); //Khoi tao PWM dung 0.25d
    output_low(pin_c1);
    set_pwm1_duty((int16)0);
    enable_interrupts(int_rda);
    enable_interrupts(global);
    while(true)
    {
        if(bdn>=10) //***** Cau 3a *****
        {
            //Dem giay dung 0.5d
            bdn-=10;
            giay++;
            giay%=60;
        }
        output_b(m7d[giay%10]); //Hien thi giay dung 0.25d
    }
}

```

```

        output_d(m7d[giay/10]);
        if(input(pin_e2)==0) putc(0xBB); //***** Cau 3c *****
    }                                     //Nhan BTB nut goi 0xBB 0.25d
}
#int_timer1    //***** Cau 3a *****
void ngatt1()
{
    set_timer1(3036);
    bdn++;
}

#int_rda    //***** Cau 3b *****
void ngatnhan()                                     //Nhan duoc giu lieu qua UART 0.25d
{
    nhan = getc();                                     //dung 4 cap do 0.5d
    set_pwm1_duty((nhan-0xA0)*250); //Cap 3 SV dieu khien 70% hay 75% deu cham dung
}

```