

Câu 1: (3 điểm)

Xét một ứng dụng PIC16F887 ($F_{OSC} = 8\text{MHz}$) tạo xung có: 04 nút nhấn thường hồ tên gọi P1 (kết nối với chân RB1), P2 (kết nối với chân RB2), P3 (kết nối với chân RB3) và STOP (kết nối với chân $\overline{\text{MCLR}}$); 01 LED bảy đoạn (kết nối với PORTD); và một ngõ ra xung PulseOut (kết nối với chân RC2/CCP1) có chu kỳ $T = 2\text{ms}$, chu kỳ công tác **Duty = 75%**. Nhấn nút P1 (LED hiển thị số 1), hoặc P2 (LED hiển thị số 2), hoặc P3 (LED hiển thị số 3) sẽ phát xung ra chân PulseOut. Nhấn nút STOP (LED hiển thị số 0) sẽ tắt xung ra PulseOut; khi đó PulseOut ở mức thấp. Giải thuật phát xung ra chân PulseOut tùy thuộc vào mỗi nút nhấn P1, P2, và P3 như sau:

- Nhấn nút P1: Trì hoãn mức thấp và mức cao sử dụng hàm **delay_us(t)**.
- Nhấn nút P2: Trì hoãn sử dụng **Timer1**.
- Nhấn nút P3: Sử dụng kỹ thuật **PWM**.

- Vẽ lưu đồ. (1.5 điểm)
- Viết chương trình C. (1.5 điểm)

Câu 2: (4 điểm)

Một ứng dụng PIC16F887 đo nhiệt độ 2 kênh (CH1 & CH2) được điều khiển bằng máy tính qua giao tiếp **RS232**, sử dụng cảm biến LM35, môi trường cần đo có nhiệt độ trong phạm vi $[0^{\circ}\text{C} \dots 150^{\circ}\text{C}]$. Nhiệt độ được hiển thị trên một màn hình LCD 16x2 (tùy chọn kết nối PORT). Sử dụng dao động ngoài cho vi điều khiển, $F_{OSC} = 20\text{MHz}$. **CH1** là ký hiệu nhiệt độ Kênh 1 (được nối đến AN1), **CH2** là ký hiệu nhiệt độ Kênh 2 (được nối đến AN2). Một Led đơn (được nối đến RA0).

- Nếu kênh nào có nhiệt độ $\geq 50^{\circ}\text{C}$, gọi là bị quá nhiệt, thì sẽ được ký hiệu [H]; ngược lại thấp nhiệt được ký hiệu [L]. Xem minh họa trên hình: **CH1** đang thấp nhiệt.
- Giao diện trên máy tính có 04 nút: CH1, CH2, CH1&2, và STOP. Nhấp CH1, chỉ đo Kênh 1. Nhấp CH2, chỉ đo Kênh 2. Nhấp CH1&2, đo cả hai Kênh 1 & Kênh 2. Nhấp STOP, không đo cả hai kênh. Kênh nào không được đo sẽ hiện ký hiệu OFF và [x]. Xem minh họa trên hình: **CH2** đang không được đo.
- Bất kể kênh nào đang được đo mà bị quá nhiệt thì Led đơn sáng; ngược lại Led tắt.

- Vẽ mạch nguyên lý. (1.0 điểm)
- Viết chương trình C. (3.0 điểm)

Câu 3: (3 điểm)

Một ứng dụng phát số ngẫu nhiên có một màn hình gồm hai LED bảy đoạn (được nối đến PORTC & PORTD) và một nút nhấn thường hồ ký hiệu RANDOM (được nối đến RE0). Mỗi lần nhấn nút RANDOM sẽ phát sinh một số có giá trị ngẫu nhiên trong phạm vi $[00-99]$ và được hiển thị trên hai LED bảy đoạn. **Lưu ý:** Sinh viên phải tự xây dựng giải thuật để giải quyết vấn đề và không được phép dùng các hàm liên quan đến phát số ngẫu nhiên có sẵn.

- Vẽ lưu đồ. (1.5 điểm)
- Viết chương trình C. (1.5 điểm)

Ghi chú: Cán bộ coi thi không được giải thích đề thi.

Chuẩn đầu ra của học phần (về kiến thức)	Nội dung kiểm tra
[G4.1]: Hiểu và ứng dụng được hệ thống vi điều khiển vào thực tế.	Câu 1, Câu 2, Câu 3
[G1.3]: Thiết kế và lập trình điều khiển dùng ngôn ngữ C cho các hệ thống điều khiển.	Câu 1, Câu 2, Câu 3
[G2.3]: Có kỹ năng thiết kế và xây dựng qui trình điều khiển cho hệ thống dùng vi điều khiển PIC16F887.	Câu 1, Câu 2, Câu 3

Ngày tháng năm 2019

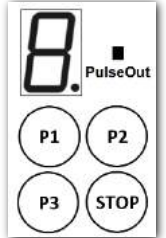
Thông qua bộ môn

(ký và ghi rõ họ tên)

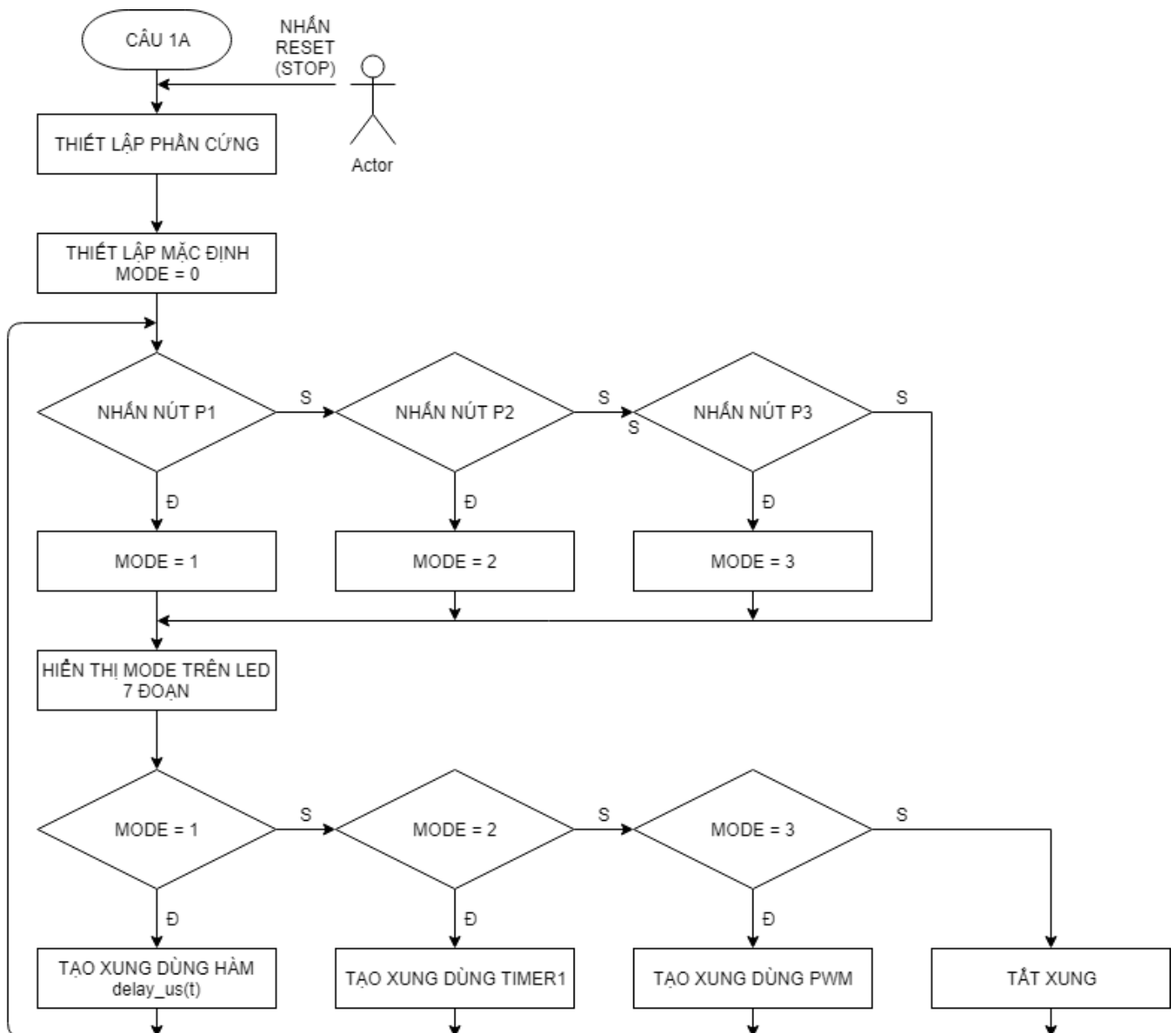
Câu 1: (3 điểm)

Xét một ứng dụng PIC16F887 ($F_{OSC} = 8\text{MHz}$) tạo xung có: 04 nút nhấn thường hở tên gọi P1 (kết nối với chân RB1), P2 (kết nối với chân RB2), P3 (kết nối với chân RB3) và STOP (kết nối với chân MCLR); 01 LED bảy đoạn (kết nối với PORTD); và một ngõ ra xung PulseOut (kết nối với chân RC2/CCP1) có chu kỳ $T = 2\text{ms}$, chu kỳ công tác **Duty = 75%**. Nhấn nút P1 (LED hiển thị số 1), hoặc P2 (LED hiển thị số 2), hoặc P3 (LED hiển thị số 3) sẽ phát xung ra chân PulseOut. Nhấn nút STOP (LED hiển thị số 0) sẽ tắt xung ra PulseOut; khi đó PulseOut ở mức thấp. Giải thuật phát xung ra chân PulseOut tùy thuộc vào mỗi nút nhấn P1, P2, và P3 như sau:

- Nhấn nút P1: Trì hoãn mức thấp và mức cao sử dụng hàm **delay_us(t)**.
 - Nhấn nút P2: Trì hoãn sử dụng **Timer1**.
 - Nhấn nút P3: Sử dụng kỹ thuật **PWM**.
- a) Vẽ lưu đồ. (1.5 điểm)
b) Viết chương trình C. (1.5 điểm)



BÀI GIẢI Câu 1a)



BÀI GIẢI Câu 1a)

```
#include <16F887.h>
#fuses INTRC
#use delay(CLOCK=8MHz)

#define btnP1      PIN_B1
#define btnP2      PIN_B2
#define btnP3      PIN_B3
//-----STOP-----MCLR
//-----LED-----PORTD
#define PulseOut   PIN_C2    //CCP1
#bit T1IF=0x0C.0
//Digit:           '0'   '1'   '2'   '3'   '4'   '5'   '6'   '7'   '8'   '9'
char Code7Seg[10]={0xC0,0xF9,0xA4,0xB0,0x99,0x92,0x82,0xF8,0x80,0x90};
unsigned int8 Mode;

void main()
{
    set_tris_b(0xFF);port_b_pullups(0xFF);
    set_tris_c(0x00);
    set_tris_d(0x00);

    setup_timer_1(T1_INTERNAL|T1_DIV_BY_1);

    //PR2=(Period*Fosc)/(4*DIV)-1=(2*1000*8)/(4*16) -1 = 249
    setup_timer_2(T2_DIV_BY_16,249,1);//T=PWM Period = 2ms; DIV=16;PR=249
    set_timer2(0);

    //value=(Period*Duty*Fosc)/(100%*DIV)=(2ms*75%*8MHz)/(100%*16)
    set_pwm1_duty(750);//Duty =75%
    setup_ccp1(CCP_OFF);

    Mode =0;

    while(TRUE)
    {
        if(input(btnP1)==0) Mode=1;else
        if(input(btnP2)==0) Mode=2;else
        if(input(btnP3)==0) Mode=3;else;
        output_d(Code7Seg[Mode]);
        if(Mode==1) //using delay_us(t)
        {
            setup_ccp1(CCP_OFF);
            output_high(PulseOut);
            delay_us(1500);//delay 1500us
            output_low(PulseOut);
        }
    }
}
```

```

delay_us(500); //delay 500us
}
else
if(Mode==2) //using Timer1
{
setup_ccp1(CCP_OFF);
output_high(PulseOut);
//delay 1.5ms = 1500us
setup_timer_1(T1_INTERNAL|T1_DIV_BY_1);
set_timer1(-1500*8/(4*1));
T1IF=0;
while(T1IF==0);
//-----
output_low(PulseOut);
//delay 0.5ms = 500us
setup_timer_1(T1_INTERNAL|T1_DIV_BY_1);
set_timer1(-500*8/(4*1));
T1IF=0;
while(T1IF==0);
//-----
}
else
if(Mode==3) //using PWM1
{
setup_ccp1(CCP_PWM);
}
else
{
output_low(PulseOut);
}

}
}

```

Câu 2: (4 điểm)

Một ứng dụng PIC16F887 đo nhiệt độ 2 kênh (CH1 & CH2) được điều khiển bằng máy tính qua giao tiếp **RS232**, sử dụng cảm biến LM35, môi trường cần đo có nhiệt độ trong phạm vi [0°C ... 150°C]. Nhiệt độ được hiển thị trên một màn hình LCD 16x2 (tùy chọn kết nối PORT). Sử dụng dao động ngoài cho vi điều khiển, $F_{osc} = 20\text{MHz}$. **CH1** là ký hiệu nhiệt độ Kênh 1 (được nối đến AN1), **CH2** là ký hiệu nhiệt độ Kênh 2 (được nối đến AN2). Một Led đơn (được nối đến RA0).

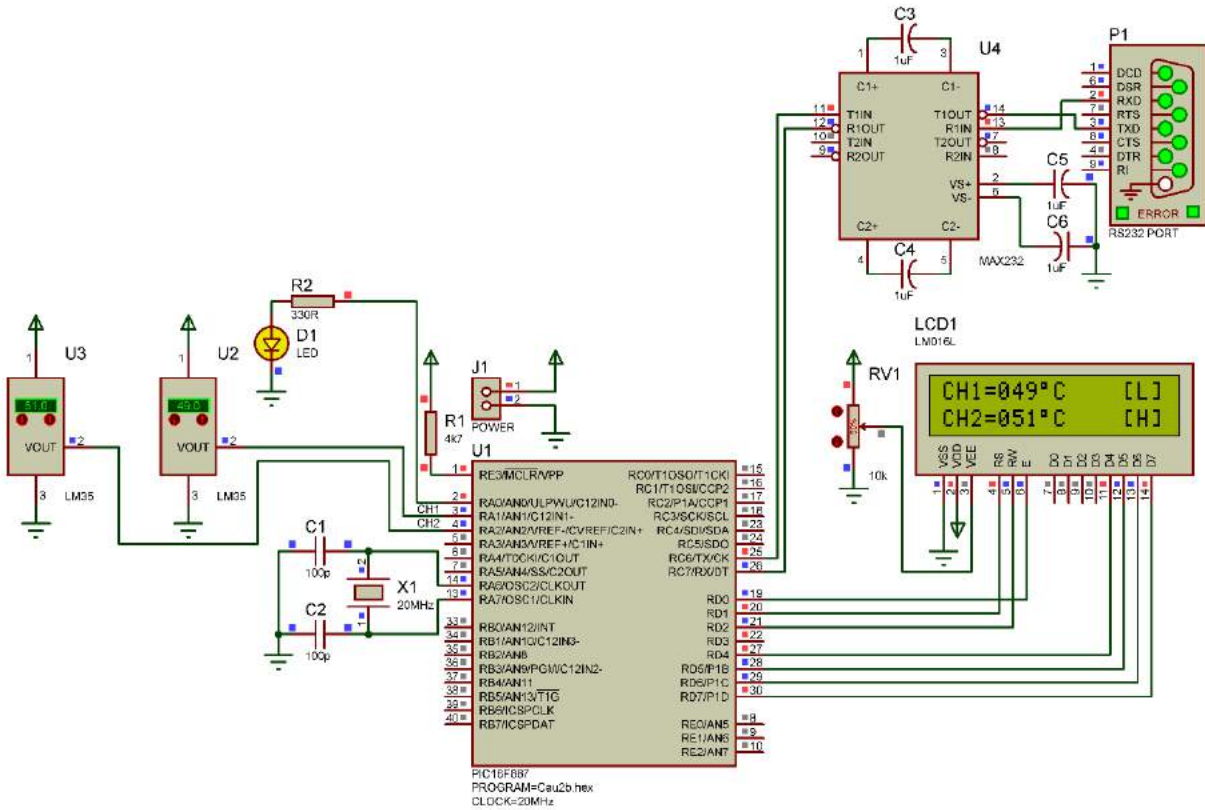
- Nếu kênh nào có nhiệt độ $\geq 50^\circ\text{C}$, gọi là bị quá nhiệt, thì sẽ được ký hiệu [H]; ngược lại thấp nhiệt được ký hiệu [L]. Xem minh họa trên hình: **CH1** đang thấp nhiệt.
- Giao diện trên máy tính có 04 nút: CH1, CH2, CH1&2, và STOP. Nhấp CH1, chỉ đo Kênh 1. Nhấp CH2, chỉ đo Kênh 2. Nhấp CH1&2, đo cả hai Kênh 1 & Kênh 2. Nhấp STOP, không đo cả hai kênh. Kênh nào không được đo sẽ hiện ký hiệu OFF và [x]. Xem minh họa trên hình: **CH2** đang không được đo.
- Bất kể kênh nào đang được đo mà bị quá nhiệt thì Led đơn sáng; ngược lại Led tắt.



CH1=027°C	[L]
CH2=OFF	[x]

- Vẽ mạch nguyên lý.
- Viết chương trình C.

BÀI GIẢI Câu 2a)



BÀI GIẢI Câu 2b)

```
#include<16F887.h>
#device ADC=10
#fuses HS
#use delay(CLOCK=20MHz)
#use RS232(BAUD=9600,BITS=8,STOP=1,PARITY=N, XMIT=PIN_C6,RCV=PIN_C7)
#include<lcd.c>

#define Led      PIN_A0

unsigned int16 xADC1,xADC2;
char RxData;

void ReadCH1();
void ReadCH2();

void main()
{
set_tris_a(0xFE);
setup_adc_ports(sAN1|sAN2|VSS_VDD);
setup_adc(ADC_CLOCK_INTERNAL);
```

```

lcd_init();
RxData='3';xADC1=0;xADC2=0;
while(TRUE)
{
if(kbhit()==TRUE) RxData=getc(); else;
switch(RxData)
{
case '1':ReadCH1();//CH1 only
        lcd_gotoxy(1,2);lcd_putc("CH2=OFF  ");
        lcd_gotoxy(14,2);lcd_putc("[x]");
        if(xADC1>=50) output_high(Led);
        else output_low(Led);
        break;
case '2':ReadCH2();//CH2 only
        lcd_gotoxy(1,1);lcd_putc("CH1=OFF  ");
        lcd_gotoxy(14,1);lcd_putc("[x]");
        if(xADC2>=50) output_high(Led);
        else output_low(Led);
        break;
case '3':ReadCH1();//CH1&CH2
        ReadCH2();
        if((xADC1>=50)|| (xADC2>=50)) output_high(Led);
        else output_low(Led);
        break;
default ://STOP
        lcd_gotoxy(1,1);lcd_putc("CH1=OFF  ");
        lcd_gotoxy(14,1);lcd_putc("[x]");
        lcd_gotoxy(1,2);lcd_putc("CH2=OFF  ");
        lcd_gotoxy(14,2);lcd_putc("[x]");
        output_low(Led);
        break;
}

}

}

void ReadCH1()
{
        set_adc_channel(1);
        delay_us(20);
        xADC1=read_adc()/2.046;
        lcd_gotoxy(1,1);
        printf(lcd_putc,"CH1=%03Lu",xADC1);
        lcd_putc(223);lcd_putc('C');

        if(xADC1>=50)

```

```

    {
        lcd_gotoxy(14,1);lcd_putc("[H]");
    }
    else
    {
        lcd_gotoxy(14,1);lcd_putc("[L]");
    }
}

void ReadCH2()
{
    set_adc_channel(2);
    delay_us(20);
    xADC2=read_adc()/2.046;
    lcd_gotoxy(1,2);
    printf(lcd_putc,"CH2=%03Lu",xADC2);
    lcd_putc(223);lcd_putc('C');

    if(xADC2>=50)
    {
        lcd_gotoxy(14,2);lcd_putc("[H]");
    }
    else
    {
        lcd_gotoxy(14,2);lcd_putc("[L]");
    }
}

```

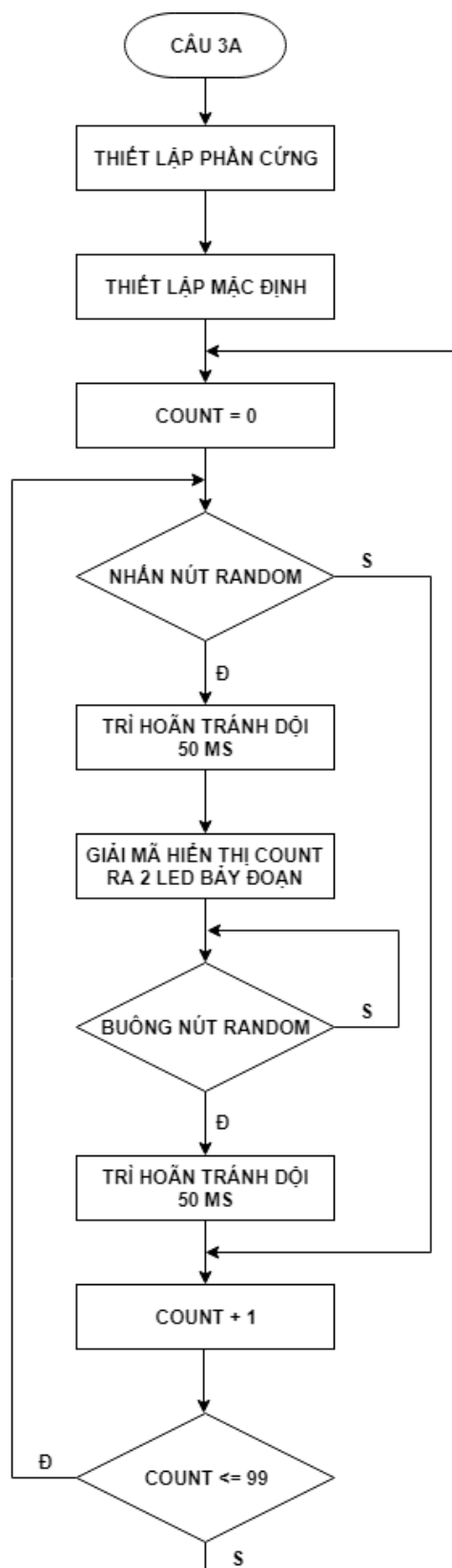
Câu 3: (3 điểm)

Một ứng dụng phát số ngẫu nhiên có một màn hình gồm hai LED bảy đoạn (được nối đến PORTC & PORTD) và một nút nhấn thường hở ký hiệu RANDOM (được nối đến RE0). Mỗi lần nhấn nút RANDOM sẽ phát sinh một số có giá trị ngẫu nhiên trong phạm vi [00-99] và được hiển thị trên hai LED bảy đoạn. **Lưu ý:** Sinh viên phải tự xây dựng giải thuật để giải quyết vấn đề và không được phép dùng các hàm liên quan đến phát số ngẫu nhiên có sẵn.



- Vẽ lưu đồ. (1.5 điểm)
- Viết chương trình C. (1.5 điểm)

BÀI GIẢI Câu 3a)



BÀI GIẢI Câu 3b)

```

#include <16F887.h>
#fuses INTRC_IO
#use delay(CLOCK=8MHz)
#define btnRandom    PIN_E0

//          '0'  '1'  '2'  '3'  '4'  '5'  '6'  '7'  '8'  '9'
const char Code7Seg[10]={0xC0,0xF9,0xA4,0xB0,0x99,0x92,0x82,0xF8,0x80,0x90};
unsigned int8 Count;
  
```



```

void main()
{
setup_adc_ports(NO_ANALOGS);
set_tris_e(0xFF);
set_tris_c(0b00000000);
set_tris_d(0b00000000);
output_c(Code7Seg[Count/10]); //Tens
output_d(Code7Seg[Count%10]); //Units
while(TRUE)
{
for(Count=0;Count<=99;Count++)
{
if(input(btnRandom)==0)
{
delay_ms(50);
output_c(Code7Seg[Count/10]); //Tens
output_d(Code7Seg[Count%10]); //Units
while(input(btnRandom)==0);
delay_ms(50);
}
else;
}
}
}

```

Ghi chú: Cán bộ coi thi không được giải thích đề thi.

Chuẩn đầu ra của học phần (về kiến thức)	Nội dung kiểm tra
[G4.1]: Hiểu và ứng dụng được hệ thống vi điều khiển vào thực tế.	Câu 1, Câu 2, Câu 3
[G1.3]: Thiết kế và lập trình điều khiển dùng ngôn ngữ C cho các hệ thống điều khiển.	Câu 1, Câu 2, Câu 3
[G2.3]: Có kỹ năng thiết kế và xây dựng qui trình điều khiển cho hệ thống dùng vi điều khiển PIC16F887.	Câu 1, Câu 2, Câu 3

Ngày tháng năm 2019

Thông qua bộ môn

(ký và ghi rõ họ tên)

Trang: 1/1

- Chống dội ADJ
 - Xét phạm vi biến bề rộng xung (iWidth)
 - Giải mã và hiển thị ra LED bảy đoạn
- (4) Tạo xung khi ở chế độ chạy (Mode=1) và biến bề rộng xung khác 0 (iWidth \neq 0)
- Đảo xung
 - Nạp bề rộng xung (iWidth x 1000) [ms]
- (5) Đáp ứng nút RUN
- (6) Đáp ứng nút STOP
- (7) Xây dựng hàm trì hoãn đơn vị [ms] có xét nhấn nút STOP: delay_msSTOP(t)

c) Chương trình

```
#include<16F887.h>
#fuses INTRC_IO
#use delay(CLOCK=8MHz)

#define pulseOUT    PIN_C0
#define btnRUN      PIN_C1
#define btnSTOP     PIN_C2
#define btnADJ      PIN_A0

//LED7SEG-----PORTD '0'  '1'  '2'  '3'  '4'  '5'  '6'  '7'  '8'  '9'
const char Code7Seg[10]={0xC0,0xF9,0xA4,0xB0,0x99,0x92,0x82,0xF8,0x80,0x90};
signed int8 iWidth;
int1 Mode;

void delay_msSTOP(unsigned int16 t);
void main()
{
    set_tris_a(0b11111111);
    set_tris_c(0b11111110);
    set_tris_d(0b00000000);

    iWidth=5;
    Mode=0;
    output_d(Code7Seg[iWidth]);

    while(TRUE)
    {
        if((Mode==0)&&(input(btnADJ)==0))
        {
            delay_ms(50);
            iWidth++;
            if(iWidth==10) iWidth=0;else;
            output_d(Code7Seg[iWidth]);
            while(input(btnADJ)==0);
            delay_ms(50);
        }
        else;

        if((Mode==1)&&(iWidth!=0))
```

```

{
output_toggle(pulseOUT);
delay_msSTOP(iWidth*1000);
}
else output_low(pulseOUT);

if(input(btnRUN)==0) Mode=1;
else
if(input(btnSTOP)==0) Mode=0;
else;
}
}

void delay_msSTOP(unsigned int16 t)
{
unsigned int16 i;
for(i=0;i<t;i++)
{
if(input(btnSTOP)==0) break; else;
delay_ms(1);
}
}

```

Câu 2: (3 điểm)

Một ứng dụng PIC16F887 đo nhiệt độ 2 kênh (X & Y), sử dụng cảm biến LM35, môi trường cần đo có nhiệt độ trong phạm vi $[0^{\circ}\text{C} \dots 150^{\circ}\text{C}]$. Nhiệt độ được hiển thị trên một màn hình LCD 16x2 (tùy chọn kết nối PORT). Tùy chọn dao động cho vi điều khiển.

$T_x=025^{\circ}\text{C}$	[L]
$T_y=109^{\circ}\text{C}$	[H]

Hình 1: Minh họa Câu 2

T_x là ký hiệu nhiệt độ kênh X (được nối đến AN0), **T_y** là ký hiệu nhiệt độ kênh Y (được nối đến AN1). Một Led đơn (được nối đến RA2).

- Nếu kênh nào có nhiệt độ $\geq 100^{\circ}\text{C}$, gọi là bị quá nhiệt, thì sẽ được ký hiệu [H], ngược lại được ký hiệu [L]. Xem minh họa trên Hình 1: **T_x** không quá nhiệt, **T_y** bị quá nhiệt.
 - Nếu nhiệt độ hai kênh bằng nhau thì Led sáng, ngược lại thì Led tắt.
- a) Vẽ lưu đồ. (1.0 điểm)
- b) Viết chương trình C. (2.0 điểm)

Bài giải:

a) Lưu đồ

▪ Hình thức

- Sử dụng đúng các ký hiệu lưu đồ
- Nội dung mô tả ở thức mệnh lệnh
- Vẽ ra lưu đồ của các hàm được định nghĩa bởi người dùng (nếu có)

▪ Nội dung

- (1) Thiết lập phần cứng
- (2) Thiết lập mặc định
- (3) Đọc nhiệt độ kênh X và hiển thị lên LCD
- (4) Đọc nhiệt độ kênh Y và hiển thị lên LCD
- (5) Xử lý đáp ứng hiển thị [H]&[L] cho kênh X
- (6) Xử lý đáp ứng hiển thị [H]&[L] cho kênh Y
- (7) Đáp ứng Led theo giá trị hai kênh X&Y

b) Chương trình

```

#include<16F887.h>
#define ADC=10
#define INTRC_IO
#define delay(CLOCK=8MHz)
#include<lcd.c>

#define Led    PIN_A2
//LCD 16x2-----PORTD

unsigned int16 iADCx,iADCy;

void main()
{
    set_tris_a(0xFF);
    setup_adc_ports(NO_ANALOGS|sAN0|sAN1|VSS_VDD);
    setup_adc(ADC_CLOCK_INTERNAL);
    lcd_init();
    while(TRUE)
    {
        set_adc_channel(0);
        delay_us(20);
        iADCx = read_adc()/2.046;
        lcd_gotoxy(1,1);
        printf(lcd_putc,"Tx=%03Lu",iADCx);
        lcd_putc(223);lcd_putc('C');

        set_adc_channel(1);
        delay_us(20);
        iADCy = read_adc()/2.046;
        lcd_gotoxy(1,2);
        printf(lcd_putc,"Ty=%03Lu",iADCy);
        lcd_putc(223);lcd_putc('C');

        if(iADCx>=100) {lcd_gotoxy(14,1);lcd_putc("[H]");}
        else {lcd_gotoxy(14,1);lcd_putc("[L]");}

        if(iADCy>=100) {lcd_gotoxy(14,2);lcd_putc("[H]");}
        else {lcd_gotoxy(14,2);lcd_putc("[L]");}

        if(iADCx==iADCy) output_high(Led);else output_low(Led);
    }
}

```

Câu 3: (3 điểm)

Một ứng dụng đếm sản phẩm có giao tiếp máy tính qua UART dùng chuẩn RS-232. Giao thức 8 bit dữ liệu, 1 bit STOP, không kiểm tra lỗi chẵn lẻ, tốc độ 4800 bps. Giao diện trên máy tính có một nút RUN (mã điều khiển là ký tự '1') và một nút STOP (mã điều khiển là ký tự '0') để điều khiển cho phép đếm (khi nhấp nút RUN) và dừng đếm (khi nhấp nút STOP). Mỗi thùng có 50 sản phẩm. Cảm biến sản phẩm được nối vào chân T1CKI của

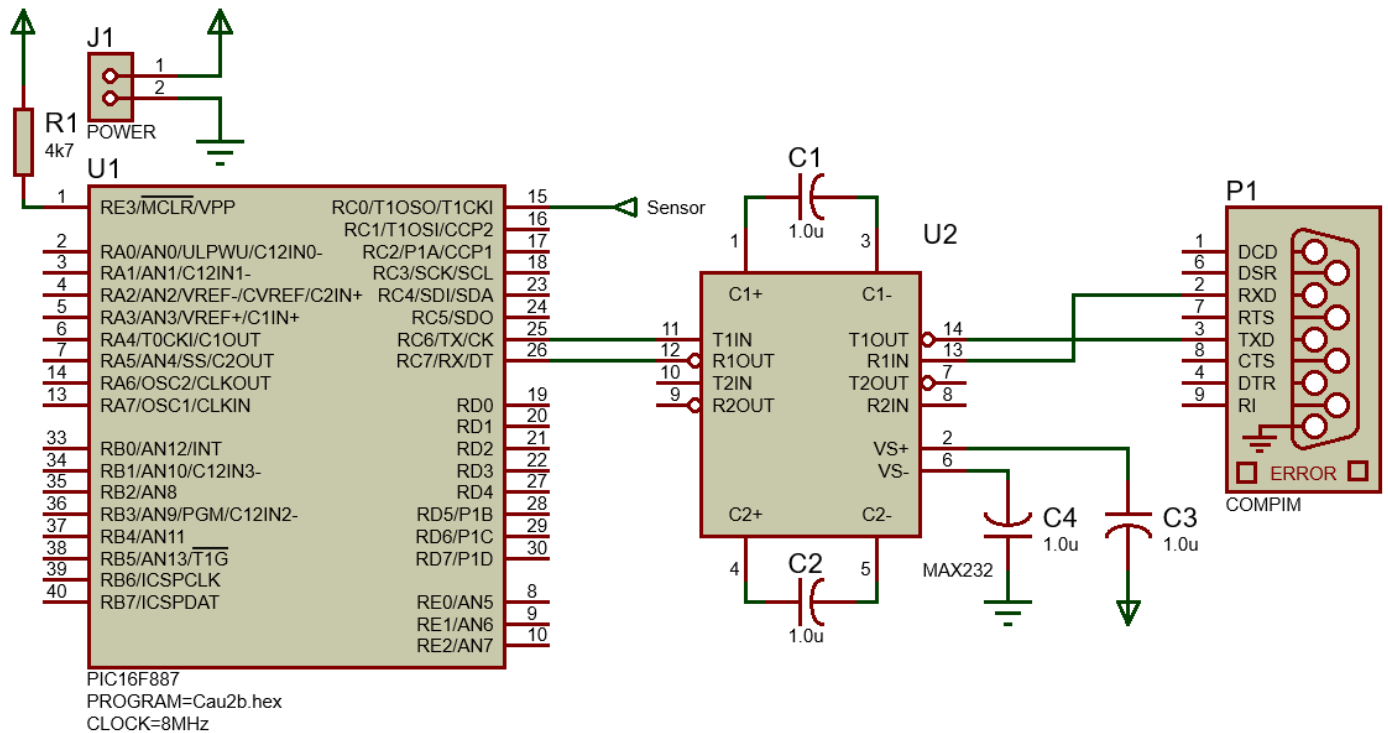
PIC16F887. Tùy chọn dao động cho vi điều khiển. Chỉ khi có sự thay đổi giá trị đếm thì vi điều khiển sẽ gửi lên máy tính khung chuỗi dữ liệu có định dạng: **P=Giá trịSản phẩm,B= Giá trịThùng**

Ví dụ: **P=15,B=00102** là đại diện cho đếm được 15 sản phẩm và 102 thùng.

- Vẽ mạch nguyên lý. (1.0 điểm)
- Viết chương trình C. (2.0 điểm)

Bài giải:

a) Mạch nguyên lý



b) Chương trình

```
#include<16F887.h>
#fuses INTRC_IO
#use delay(CLOCK=8MHz)
#use RS232 (BAUD=4800,BITS=8,PARITY=N,STOP=1,XMIT=PIN_C6,RCV=PIN_C7)

unsigned int8 valRx;
unsigned int16 valOldProduct,valProduct,valBox;
void main()
{
    set_tris_c(0b10111111);
    //setup_timer_1(T1_EXTERNAL|T1_DIV_BY_1);
    setup_timer_1(T1_DISABLED|T1_DIV_BY_1);
    set_timer1(0);
    valOldProduct=0;
    valProduct=0;
    valRx='0';
    while(TRUE)
    {
        if(kbhit()==TRUE)
        {
```

```

valRx=getc();
if(valRx=='1')setup_timer_1(T1_EXTERNAL|T1_DIV_BY_1);
else
if(valRx=='0')setup_timer_1(T1_DISABLED|T1_DIV_BY_1);
else;
}
else;

valProduct=get_timer1();
if(valProduct>=50+1) {valProduct=1;valBox++;set_timer1(1);}else;
if(valProduct!=valOldProduct)
{
printf("P=%02Lu,B=%05Lu",valProduct,valBox);putc(13);
valOldProduct=valProduct;
}
else;
}
}

```

Ghi chú: Cán bộ coi thi không được giải thích đề thi.

Chuẩn đầu ra của học phần (về kiến thức)	Nội dung kiểm tra
[G4.1]: Hiểu và ứng dụng được hệ thống vi điều khiển vào thực tế.	Câu 1, Câu 2, Câu 3
[G1.3]: Thiết kế và lập trình điều khiển dùng ngôn ngữ C cho các hệ thống điều khiển.	Câu 1, Câu 2, Câu 3
[G2.3]: Có kỹ năng thiết kế và xây dựng qui trình điều khiển cho hệ thống dùng vi điều khiển PIC16F887.	Câu 1, Câu 2, Câu 3

Ngày tháng năm 2018

Thông qua bộ môn
(ký và ghi rõ họ tên)

Câu 1: (3.5 điểm)

Một ứng dụng tạo xung PWM tại ngõ ra chân CCP1 của PIC16F887 ($F_{OSC} = 8 \text{ MHz}$) có chu kỳ xung $T = 1 \text{ ms}$, chu kỳ công tác thay đổi được thông qua một nút nhấn ADJ (được nối đến chân RB0) và một LED bảy đoạn (được nối đến PORTD). Mười giá trị [0; 1; 2; 3; 4; 5; 6; 7; 8; 9] hiển thị trên LED bảy đoạn tương ứng với mười cấp chu kỳ công tác [0%; 10%; 20%; 30%; 40%; 50%; 60%; 70%; 80%; 90%]. Mỗi lần nhấn nút ADJ, giá trị trên LED bảy đoạn sẽ tăng một đơn vị. Lên tới 9, nhấn ADJ nữa thì LED bằng 0. Mặc định lúc mới bật điện, LED bằng 5 và tắt xung PWM. Xung PWM chỉ được xuất ra sau khi nhấn nút ON (được nối đến chân RB1). Nhấn nút OFF (được nối đến chân RB2) sẽ tắt xung PWM nhưng LED bảy đoạn vẫn giữ nguyên giá trị. Khi tắt xung PWM, yêu cầu ngõ ra CCP1 ở mức Logic thấp.

- a) Vẽ lưu đồ. (1.5 điểm)
b) Viết chương trình C. (2.0 điểm)

Câu 2: (3.5 điểm)

Một ứng dụng PIC16F887 ($F_{OSC} = 20 \text{ MHz}$) đo nhiệt độ 2 kênh (X & Y), sử dụng cảm biến LM35, môi trường cần đo có nhiệt độ trong phạm vi $[0^\circ\text{C} \dots 120^\circ\text{C}]$. Nhiệt độ được hiển thị trên một màn hình LCD 16x2 (tùy chọn kết nối PORT).

$T_x = 025^\circ\text{C}$	[L]
$T_y = 109^\circ\text{C}$	[H]

Hình 1: Minh họa Câu 2

T_x là ký hiệu nhiệt độ kênh X (được nối đến AN5), T_y là ký hiệu nhiệt độ kênh Y (được nối đến AN6). Một Led đơn (được nối đến RE2) để chỉ thị quá nhiệt cho hai kênh. Chỉ cần một trong hai kênh có nhiệt độ lớn hơn 100°C thì Led sẽ nhấp nháy liên tục (sáng $\frac{1}{2}$ giây và tắt $\frac{1}{2}$ giây) và kênh quá nhiệt tương ứng sẽ được ký hiệu [H] trên màn hình LCD, ngược lại thì Led tắt và ký hiệu [L] trên màn hình LCD (xem minh họa trên Hình 1: T_x không quá nhiệt, T_y bị quá nhiệt). Chu kỳ cập nhật giá trị nhiệt độ cho cả hai kênh là 100 ms.

- a) Vẽ mạch nguyên lý hoàn chỉnh. (1.5 điểm)
b) Viết chương trình C. (2.0 điểm)

Câu 3: (3 điểm)

Một ứng dụng đếm sản phẩm có giao tiếp máy tính qua UART. Giao thức 8 bit dữ liệu, 1 bit STOP, không kiểm tra lỗi chẵn lẻ, tốc độ 2400 bps. Giao diện trên máy tính có một nút RUN và một nút STOP để điều khiển cho phép đếm (khi nhấp nút RUN) và dừng đếm (khi nhấp nút STOP). Mỗi thùng có 24 sản phẩm. Cảm biến sản phẩm được nối vào chân T0CKI của PIC16F887 (F_{OSC} tùy chọn). Chỉ khi có sự thay đổi giá trị đếm thì vi điều khiển sẽ gửi lên máy tính khung chuỗi dữ liệu có định dạng: **P=Giá trị Sản phẩm, B= Giá trị Thùng**

Ví dụ: **P=18, B=107** là đại diện cho đếm được 18 sản phẩm và 107 thùng.

- a) Vẽ mạch nguyên lý hoàn chỉnh. (1.0 điểm)
b) Viết chương trình C. (2.0 điểm)

Ghi chú: Cán bộ coi thi không được giải thích đề thi.

Ngày tháng năm 2018

Chuẩn đầu ra của học phần (về kiến thức)	Nội dung kiểm tra
[G4.1]: Hiểu và ứng dụng được hệ thống vi điều khiển vào thực tế.	Câu 1, Câu 2, Câu 3
[G1.3]: Thiết kế và lập trình điều khiển dùng ngôn ngữ C cho các hệ thống điều khiển.	Câu 1, Câu 2, Câu 3
[G2.3]: Có kỹ năng thiết kế và xây dựng qui trình điều khiển cho hệ thống dùng vi điều khiển PIC16F887.	Câu 1, Câu 2, Câu 3

Thông qua bộ môn
(ký và ghi rõ họ tên)

Câu 1: (3.5 điểm)

Một ứng dụng tạo xung PWM tại ngõ ra chân CCP1 của PIC16F887 ($F_{OSC} = 8 \text{ MHz}$) có chu kỳ xung $T = 1 \text{ ms}$, chu kỳ công tác thay đổi được thông qua một nút nhấn ADJ (được nối đến chân RB0) và một LED bảy đoạn (được nối đến PORTD). Mười giá trị [0; 1; 2; 3; 4; 5; 6; 7; 8; 9] hiển thị trên LED bảy đoạn tương ứng với mười cấp chu kỳ công tác [0%; 10%; 20%; 30%; 40%; 50%; 60%; 70%; 80%; 90%]. Mỗi lần nhấn nút ADJ, giá trị trên LED bảy đoạn sẽ tăng một đơn vị. Lên tới 9, nhấn ADJ nữa thì LED bằng 0. Mặc định lúc mới bật điện, LED bằng 5 và tắt xung PWM. Xung PWM chỉ được xuất ra sau khi nhấn nút ON (được nối đến chân RB1). Nhấn nút OFF (được nối đến chân RB2) sẽ tắt xung PWM nhưng LED bảy đoạn vẫn giữ nguyên giá trị. Khi tắt xung PWM, yêu cầu ngõ ra CCP1 ở mức Logic thấp.

a) Vẽ lưu đồ.

(1.5 điểm)

- Vẽ đúng các ký hiệu lưu đồ
- Ghi thức mệnh lệnh trong mô tả lưu đồ
- Vẽ được cấu trúc tổng thể
- Vẽ được khả năng chống dội của nút nhấn ADJ
- Vẽ được khả năng điều chỉnh giá trị các cấp chu kỳ công tác của nút ADJ
- Vẽ được mối quan hệ giữa giá trị các cấp chu kỳ công tác với các thiết lập PWM
- Vẽ được khả năng giải mã và hiển thị trên LED bảy đoạn
- Vẽ được các lưu đồ chương trình con (nếu có)
- Vẽ được các yêu cầu phụ

b) Viết chương trình C.

(2.0 điểm)

- Sử dụng đúng cú pháp, tên lệnh, tên hàm, tên biến, toán tử
- Mô tả được cấu trúc tổng thể
- Mô tả được khả năng chống dội của nút nhấn ADJ
- Mô tả được khả năng điều chỉnh giá trị các cấp chu kỳ công tác của nút ADJ
- Mô tả được mối quan hệ giữa giá trị các cấp chu kỳ công tác với các thiết lập PWM
- Mô tả được khả năng giải mã và hiển thị trên LED bảy đoạn
- Mô tả được các chương trình con (nếu có)
- Mô tả được các yêu cầu phụ

Câu 2: (3.5 điểm)

Một ứng dụng PIC16F887 ($F_{OSC} = 20 \text{ MHz}$) đo nhiệt độ 2 kênh (X & Y), sử dụng cảm biến LM35, môi trường cần đo có nhiệt độ trong phạm vi $[0^\circ\text{C} \dots 120^\circ\text{C}]$. Nhiệt độ được hiển thị trên một màn hình LCD 16x2 (tùy chọn kết nối PORT).

$T_x = 025^\circ\text{C}$	[L]
$T_y = 109^\circ\text{C}$	[H]

Hình 1: Minh họa Câu 2

T_x là ký hiệu nhiệt độ kênh X (được nối đến AN5), T_y là ký hiệu nhiệt độ kênh Y (được nối đến AN6). Một Led đơn (được nối đến RE2) để chỉ thị quá nhiệt cho hai kênh. Chỉ cần một trong hai kênh có nhiệt độ lớn hơn 100°C thì Led sẽ nhấp nháy liên tục (sáng $\frac{1}{2}$ giây và tắt $\frac{1}{2}$ giây) và kênh quá nhiệt tương ứng sẽ được ký hiệu [H] trên màn hình LCD, ngược lại thì Led tắt và ký hiệu [L] trên màn hình LCD (xem minh họa trên Hình 1: T_x không quá nhiệt, T_y bị quá nhiệt). Chu kỳ cập nhật giá trị nhiệt độ cho cả hai kênh là 100 ms.

a) Vẽ mạch nguyên lý hoàn chỉnh.

(1.5 điểm)

- Vẽ đúng ký hiệu linh kiện
- Ghi đúng giá trị linh kiện

- *Vẽ đúng cấu hình mạch*
 - *Ghi tên linh kiện (không bắt buộc)*
- b) Viết chương trình C. (2.0 điểm)
- *Sử dụng đúng cú pháp, tên lệnh, tên hàm, tên biến, toán tử*
 - *Mô tả được cấu trúc tổng thể*
 - *Mô tả được tính năng đọc nhiệt độ 2 kênh*
 - *Mô tả được tính năng hiển thị thông tin trên LCD*
 - *Mô tả được mối quan hệ giữa nhiệt độ và Led đơn*
 - *Xử lý được thời gian nhấp nháy Led và chu kỳ cập nhật nhiệt độ*

Câu 3: (3 điểm)

Một ứng dụng đếm sản phẩm có giao tiếp máy tính qua UART. Giao thức 8 bit dữ liệu, 1 bit STOP, không kiểm tra lỗi chẵn lẻ, tốc độ 2400 bps. Giao diện trên máy tính có một nút RUN và một nút STOP để điều khiển cho phép đếm (khi nhấp nút RUN) và dừng đếm (khi nhấp nút STOP). Mỗi thùng có 24 sản phẩm. Cảm biến sản phẩm được nối vào chân T0CKI của PIC16F887 (FOSC tùy chọn). Chỉ khi có sự thay đổi giá trị đếm thì vi điều khiển sẽ gửi lên máy tính khung chuỗi dữ liệu có định dạng: **P=Giá trị Sản phẩm, B= Giá trị Thùng**

Ví dụ: **P=18, B=107** là đại diện cho đếm được 18 sản phẩm và 107 thùng.

- a) Vẽ mạch nguyên lý hoàn chỉnh. (1.0 điểm)
- *Vẽ đúng ký hiệu linh kiện*
 - *Ghi đúng giá trị linh kiện*
 - *Vẽ đúng cấu hình mạch*
 - *Ghi tên linh kiện (không bắt buộc)*
- b) Viết chương trình C. (2.0 điểm)
- *Sử dụng đúng cú pháp, tên lệnh, tên hàm, tên biến, toán tử*
 - *Mô tả được cấu trúc tổng thể*
 - *Mô tả được giao thức truyền dữ liệu*
 - *Mô tả được tính năng đếm sản phẩm*
 - *Mô tả được tính năng gửi dữ liệu lên máy tính khi có sự thay đổi*

Ghi chú: Cán bộ coi thi không được giải thích đề thi.

Ngày tháng năm 2018

Chuẩn đầu ra của học phần (về kiến thức)	Nội dung kiểm tra
[G4.1]: Hiểu và ứng dụng được hệ thống vi điều khiển vào thực tế.	Câu 1, Câu 2, Câu 3
[G1.3]: Thiết kế và lập trình điều khiển dùng ngôn ngữ C cho các hệ thống điều khiển.	Câu 1, Câu 2, Câu 3
[G2.3]: Có kỹ năng thiết kế và xây dựng qui trình điều khiển cho hệ thống dùng vi điều khiển PIC16F887.	Câu 1, Câu 2, Câu 3

Thông qua bộ môn
(ký và ghi rõ họ tên)

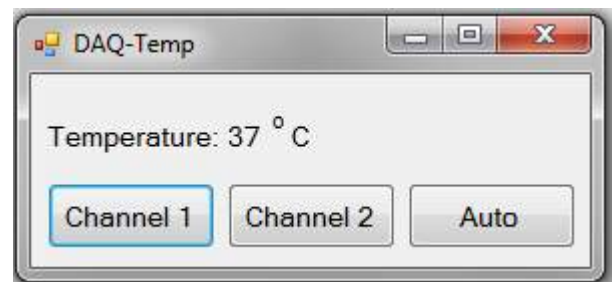
Câu 1: (4 điểm)

Một ứng dụng tạo xung vuông có điều khiển tại ngõ ra chân RC1/CCP2 của PIC16F887 có chu kỳ xung $T = 1 \text{ ms}$, chu kỳ công tác $DT = 70\%$. Chọn dao động nội $F_{OSC} = 8 \text{ MHz}$. Các câu sau đây là độc lập nhau. Hãy viết chương trình C thực hiện ứng dụng này theo các yêu cầu khác nhau như sau đây:

- Yêu cầu 1: Sử dụng hàm `delay_us(t)` để đếm thời gian. (1 điểm)
Đóng công tắc SW (được nối với RB0) cho phép ra xung, mở công tắc SW ngắt xung.
- Yêu cầu 2: Sử dụng Timer0 để đếm thời gian. (1 điểm)
Nhấn nhả nút ON (được nối với RB0) cho phép ra xung. Nhấn nhả nút OFF (được nối với RB1) ngắt xung.
- Yêu cầu 3: Sử dụng PWM để tạo xung. (1 điểm)
Nhấn nhả nút ON/OFF (được nối với RB0) cho phép ra/ngắt xung.
- Yêu cầu 4: Sử dụng ngắt Timer1 để đếm thời gian. (1 điểm)
Xung ra liên tục không có điều khiển.

Câu 2: (3 điểm)

Một ứng dụng đo nhiệt độ 2 kênh, sử dụng cảm biến LM35, môi trường cần đo có nhiệt độ trong phạm vi $[0^\circ\text{C} \text{ --- } 50^\circ\text{C}]$. Nhiệt độ được hiển thị luân phiên trên 2 LED bảy đoạn (được kết nối với PORTB & PORTD). Hai Led đơn LD1 (được nối với RC1) và LD2 (được nối với RC2) chỉ thị số thứ tự của kênh nhiệt độ: LD1 sáng chỉ thị đang đo Kênh 1 (được nối với AN5) và LD2 sáng chỉ thị đang đo Kênh 2 (được nối với AN6). Giao diện DAQ-Temp trên máy tính điều khiển chọn đo Kênh 1 (Channel 1, mã nút bằng 01h) hoặc Kênh 2 (Channel 2, mã nút bằng 02h) hoặc luân phiên hai kênh (Auto, mã nút bằng 00h) cách nhau 3s. Mặc định lúc mới bật điện là đo Kênh 1. Cho $F_{OSC} = 11.0592 \text{ MHz}$.



Yêu cầu nhiệt độ phải được đọc và lấy trung bình giá trị trong 300 lần trước khi hiển thị ra LED bảy đoạn và chỉ gửi giá trị nhiệt độ lên máy tính khi có sự thay đổi nhiệt độ qua cổng UART theo giao thức: [9600 bps, 8 bit dữ liệu, 1 STOP, không kiểm tra chẵn lẻ]. Giao tiếp đường truyền RS232 sử dụng vi mạch MAX232.

- Vẽ mạch nguyên lý hoàn chỉnh. (1 điểm)
- Vẽ lưu đồ. (1 điểm)
- Viết chương trình C. (1 điểm)

Câu 3: (3 điểm)

Một ứng dụng đếm sản phẩm. Mỗi thùng có 50 sản phẩm. Nhấn nhả nút RUN (được nối với RB0) cho phép đếm; nhấn nhả nút STOP (được nối với RB1) dừng đếm; nhấn nhả nút RESET (được nối với $\overline{\text{MCLR}}$) tác dụng giống lúc mới bật điện: dừng đếm và xóa các kết quả đếm về 0. Dưới đây là giao diện của LCD 16x2 (được nối với PORTD) ở 3 chế độ RUN, STOP, và RESET: P ký hiệu cho sản phẩm (Product); Box cho thùng. Khi đếm được 50.000 thùng thì hệ thống dừng đếm và hiện lên chữ STOP ở góc trên và FULL ở góc phải dưới của LCD. Chọn dao động nội $F_{OSC} = 8 \text{ MHz}$.

P=09	RUN
Box=00027	

P=15	STOP
Box=00027	

P=00	RESET
Box=00000	

- Vẽ lưu đồ. (1.5 điểm)
- Viết chương trình C. (1.5 điểm)

Ghi chú: Cán bộ coi thi không được giải thích đề thi.

Chuẩn đầu ra của học phần (về kiến thức)	Nội dung kiểm tra
[G4.1]: Hiểu và ứng dụng được hệ thống vi điều khiển vào thực tế.	Câu 1, Câu 2, Câu 3
[G1.3]: Thiết kế và lập trình điều khiển dùng ngôn ngữ C cho các hệ thống điều khiển.	Câu 1, Câu 2, Câu 3
[G2.3]: Có kỹ năng thiết kế và xây dựng qui trình điều khiển cho hệ thống dùng vi điều khiển PIC16F887.	Câu 1, Câu 2, Câu 3

Ngày 29 tháng 12 năm 2017

Thông qua bộ môn

(ký và ghi rõ họ tên)

Câu 1: (4 điểm)

Một ứng dụng tạo xung vuông có điều khiển tại ngõ ra chân RC1/CCP2 của PIC16F887 có chu kỳ xung $T = 1 \text{ ms}$, chu kỳ công tác $DT = 70\%$. Chọn dao động nội $F_{OSC} = 8 \text{ MHz}$. Các câu sau đây là độc lập nhau. Hãy viết chương trình C thực hiện ứng dụng này theo các yêu cầu khác nhau như sau đây:

a) Yêu cầu 1: Sử dụng hàm `delay_us(t)` để đếm thời gian. (1 điểm)

Đóng công tắc SW (được nối với RB0) cho phép ra xung, mở công tắc SW ngắt xung.

```
#include<16f887.h>
#fuses INTRC
#use delay(CLOCK=8MHz)
#define PulseOut PIN_C1
#define SW PIN_B0
void main()
{
    set_tris_c(0b11111101);
    set_tris_b(0b11111111);port_b_pullups(0b11111111);
    while(TRUE)
    {
        if(input(SW)==0)
        {
            output_high(PulseOut);
            delay_us(700);
            output_low(PulseOut);
            delay_us(300);
        }
        else output_low(PulseOut);
    }
}
```

b) Yêu cầu 2: Sử dụng Timer0 để đếm thời gian. (1 điểm)

Nhấn nhả nút ON (được nối với RB0) cho phép ra xung. Nhấn nhả nút OFF (được nối với RB1) ngắt xung.

```
#include<16f887.h>
#fuses INTRC
#use delay(CLOCK=8MHz)
#define PulseOut PIN_C1
#define ON PIN_B0
#define OFF PIN_B1
#bit T0IF=0x0B.2
```

```

int1 Mode;
void main()
{
set_tris_c(0b11111101);
set_tris_b(0b11111111);port_b_pullups(0b11111111);
while(TRUE)
{
if(input(ON)==0)Mode=1;else
if(input(OFF)==0)Mode=0;else;
if(Mode==1)
{
output_high(PulseOut);
T0IF=0;
setup_timer_0(RTCC_INTERNAL|RTCC_DIV_8);
set_timer0(-700*8/(4*8));//Th=DT*T=70%*1ms=0.7ms=700us
while(T0IF==0);

output_low(PulseOut);
T0IF=0;
setup_timer_0(RTCC_INTERNAL|RTCC_DIV_4);
set_timer0(-300*8/(4*4));//Tl=T-Th=1ms-0.7ms=0.3ms=300us
while(T0IF==0);
}
else output_low(PulseOut);
}
}

```

c) Yêu cầu 3: Sử dụng PWM để tạo xung.

(1 điểm)

Nhấn nhả nút ON/OFF (được nối với RB0) cho phép ra/ngắt xung.

```

#include<16f887.h>
#fuses INTRC
#use delay(CLOCK=8MHz)
#define PulseOut PIN_C1
#define ONOFF PIN_B0

int1 Mode;
void main()
{
set_tris_c(0b11111101);
set_tris_b(0b11111111);port_b_pullups(0b11111111);

```

```

//PR2=(Period*Fosc)/(4*DIV)-1=(1*1000*8)/(4*16) -1 = 124
setup_timer_2(T2_DIV_BY_16,124,1);//T=PWM Period = 1ms; DIV=16;PR=124
//value=(Period*Duty*Fosc)/(100%*DIV)=(1ms*70%*8MHz)/(100%*16)
set_pwm2_duty(350);//Duty =70%
while(TRUE)
{
if(input(ONOFF)==0)
{
delay_ms(50);
Mode=~Mode;
while(input(ONOFF)==0);
delay_ms(50);
}
else;
if(Mode==1) setup_ccp2(CCP_PWM);
else
{
setup_ccp2(CCP_OFF);
output_low(PulseOut);
}
}
}

```

d) Yêu cầu 4: Sử dụng ngắt Timer1 để đếm thời gian.
Xung ra liên tục không có điều khiển.

(1 điểm)

```

#include<16F887.h>
#fuses INTRC
#use delay(CLOCK=8MHz)
#define PulseOut PIN_C1

unsigned int8 i;

#INT_TIMER1
void ISR_TMR1()
{
i++;
if(i==10) i=0;else;
clear_interrupt(INT_TIMER1);
setup_timer_1(T1_INTERNAL|T1_DIV_BY_1);
set_timer1(-100*8/(4*1));
}

void main()

```

```

{
setup_timer_1(T1_INTERNAL|T1_DIV_BY_1);
set_timer1(-100*8/(4*1));

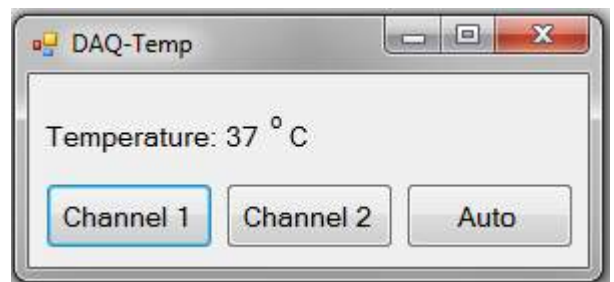
set_tris_c(0x00);
enable_interrupts(GLOBAL);
enable_interrupts(INT_TIMER1);

while(TRUE)
{
if(i<7) output_high(PulseOut);
else output_low(PulseOut);
}
}

```

Câu 2: (3 điểm)

Một ứng dụng đo nhiệt độ 2 kênh, sử dụng cảm biến LM35, môi trường cần đo có nhiệt độ trong phạm vi [0°C --- 50°C]. Nhiệt độ được hiển thị luân phiên trên 2 LED bảy đoạn (được kết nối với PORTB & PORTD). Hai Led đơn LD1 (được nối với RC1) và LD2 (được nối với RC2) chỉ thị số thứ tự của kênh nhiệt độ: LD1 sáng chỉ thị đang đo Kênh 1 (được nối với AN5) và LD2 sáng chỉ thị đang đo Kênh 2 (được nối với AN6). Giao diện DAQ-Temp trên máy tính điều khiển chọn đo Kênh 1 (Channel 1, mã nút bằng 01h) hoặc Kênh 2 (Channel 2, mã nút bằng 02h) hoặc luân phiên hai kênh (Auto, mã nút bằng 00h) cách nhau 3s. Mặc định lúc mới bật điện là đo Kênh 1. Cho $F_{OSC} = 11.0592 \text{ MHz}$.



Yêu cầu nhiệt độ phải được đọc và lấy trung bình giá trị trong 300 lần trước khi hiển thị ra LED bảy đoạn và chỉ gửi giá trị nhiệt độ lên máy tính khi có sự thay đổi nhiệt độ qua cổng UART theo giao thức: [9600 bps, 8 bit dữ liệu, 1 STOP, không kiểm tra chẵn lẻ]. Giao tiếp đường truyền RS232 sử dụng vi mạch MAX232.

- a) Vẽ mạch nguyên lý hoàn chỉnh. (1 điểm)


```

set_tris_e(0x0F);setup_adc_ports(sAN5|sAN6);
setup_adc(ADC_CLOCK_INTERNAL);//Frc

set_tris_c(0b10000000);
set_tris_d(0x00);
set_tris_b(0x00);

ChanNum=1;
while(TRUE)
{
if(kbhit()==TRUE) ChanNum=getc()-0x30;else;//for Simulation
//if(kbhit()==TRUE) ChanNum=getc();else;
if(ChanNum==0x01)
{
Temp1();
output_high(LD1);output_low(LD2);
}
else
if(ChanNum==0x02)
{
Temp2();
output_low(LD1);output_high(LD2);
}
else
{
Temp1();
output_high(LD1);output_low(LD2);
delay_ms(3000);
Temp2();
output_low(LD1);output_high(LD2);
delay_ms(3000);
}
}

void Temp1()
{
unsigned int i;
set_adc_channel(5);
delay_us(20);
Temp=0;

```

```

for(i=0;i<300;i++) {Temp=Temp+read_adc();delay_us(20);}
Temp=(Temp/300)/2.046;
output_b(Code7Seg[Temp/10]);
output_d(Code7Seg[Temp%10]);
if(OldTemp1!=Temp)
{
printf("%01u#",1);
printf("%02Lu",Temp);
OldTemp1=Temp;
}

}

void Temp2()
{
unsigned int16 i;
set_adc_channel(6);
delay_us(20);
Temp=0;
for(i=0;i<300;i++) {Temp=Temp+read_adc();delay_us(20);}
Temp=(Temp/300)/2.046;
output_b(Code7Seg[Temp/10]);
output_d(Code7Seg[Temp%10]);

if(OldTemp2!=Temp)
{
printf("%01u#",2);
printf("%02Lu",Temp);
OldTemp2=Temp;
}
}

```

Câu 3: (3 điểm)

Một ứng dụng đếm sản phẩm. Mỗi thùng có 50 sản phẩm. Nhấn nhả nút RUN (được nối với RB0) cho phép đếm; nhấn nhả nút STOP (được nối với RB1) dừng đếm; nhấn nhả nút RESET (được nối với $\overline{\text{MCLR}}$) tác dụng giống lúc mới bật điện: dừng đếm và xóa các kết quả đếm về 0. Dưới đây là giao diện của LCD 16x2 (được nối với PORTD) ở 3 chế độ RUN, STOP, và RESET: P ký hiệu cho sản phẩm (Product); Box cho thùng. Khi đếm được 50.000 thùng thì hệ thống dừng đếm và hiện lên chữ STOP ở góc trên và FULL ở góc phải dưới của LCD. Chọn dao động nội Fosc = 8MHz.

P=09	RUN
Box=00027	

P=15	STOP
Box=00027	

P=00	RESET
Box=00000	

a) Vẽ lưu đồ. (1.5 điểm)

- Đúng ký hiệu
- Khớp với chương trình
- Vẽ cả lưu đồ cho chương trình con (nếu có)

- Có đầy đủ các khối quá trình như trong đề bài yêu cầu
- Nội dung các mô tả ở thức mệnh lệnh

b) Viết chương trình C.

(1.5 điểm)

```
#include<16F887.h>
#fuses INTRC
#use delay(CLOCK=8MHz)
#include<lcd.c>

#define RUN  PIN_B0
#define STOP PIN_B1
//  RESET--MCLR\

unsigned int8 iProduct;
unsigned int16 iBox;
void main()
{
set_tris_c(0xFF);set_tris_b(0xFF);port_b_pullups(0b00000011);
setup_timer_1(T1_DISABLED);
set_timer1(0);
iProduct=0;iBox=0;
lcd_init();
lcd_gotoxy(12,1);
lcd_putc("RESET");
while(TRUE)
{
if(input(RUN)==0)
{
setup_timer_1(T1_DIV_BY_1|T1_EXTERNAL);
lcd_gotoxy(12,1);
lcd_putc("RUN ");
}
else
if(input(STOP)==0)
{
setup_timer_1(T1_DISABLED);
lcd_gotoxy(12,1);
lcd_putc("STOP ");
}
else;
//-----
iProduct=get_timer1();
```

```

if(iProduct==50)
{
iBox++;
iProduct=0;
set_timer1(0);
}
else;

lcd_gotoxy(1,1);
printf(lcd_putc,"P=%02u",iProduct);

lcd_gotoxy(1,2);
printf(lcd_putc,"Box=%05Lu",iBox);

//-----
if(iBox==50000)
{
setup_timer_1(T1_DISABLED);
lcd_gotoxy(12,1);lcd_putc("STOP ");
lcd_gotoxy(12,2);lcd_putc("FULL ");
}
else;

}
}

```

Ghi chú: Cán bộ coi thi không được giải thích đề thi.

Chuẩn đầu ra của học phần (về kiến thức)	Nội dung kiểm tra
[G4.1]: Hiểu và ứng dụng được hệ thống vi điều khiển vào thực tế.	Câu 1, Câu 2, Câu 3
[G1.3]: Thiết kế và lập trình điều khiển dùng ngôn ngữ C cho các hệ thống điều khiển.	Câu 1, Câu 2, Câu 3
[G2.3]: Có kỹ năng thiết kế và xây dựng qui trình điều khiển cho hệ thống dùng vi điều khiển PIC16F887.	Câu 1, Câu 2, Câu 3

Ngày 29 tháng 12 năm 2017

Thông qua bộ môn
(ký và ghi rõ họ tên)

Câu 1: (3 điểm)

a) Cách 1: Sử dụng hàm delay_us() (1 điểm)

```
#include<16F887.h>
#fuses INTRC_IO
#use delay(CLOCK=8MHz)

#define PulseOut PIN_C2

void main()
{
    set_tris_c(0b11111011);
    while(TRUE)
    {
        output_high(PulseOut);
        delay_us(1800);
        output_low(PulseOut);
        delay_us(200);
    }
}
```

b) Cách 2: Sử dụng Timer0 (1 điểm)

```
#include <16F887.h>
#fuses INTRC_IO
#use delay(CLOCK=8MHz)

#bit T0IF=0x0B.2
#define PulseOut PIN_C2

void main()
{
    set_tris_e(0b11111011);
    setup_timer_0(RTCC_INTERNAL);
    T0IF=0;
    while(TRUE)
    {
        output_low(PulseOut);
        setup_timer_0(RTCC_DIV_2);//DIV=2
        set_timer0(-200);//N=-0.2msx8MHz/(4x2)
```

```

T0IF=0;
while(T0IF==0);

output_high(PulseOut);
setup_timer_0(RTCC_DIV_16);//DIV=16
set_timer0(-225);//N=-1.8msx8MHz/(4x16)
T0IF=0;
while(T0IF==0);
}
}

```

c) Cách 3: Sử dụng PWM (1 điểm)

```

#include<16F887.h>
#fuses INTRC_IO
#use delay(CLOCK=8MHz)

void main()
{
set_tris_c(0b11111011);
//PR2=(Period*Fosc)/(4*DIV)-1=(2*8*1000)/(4*16)-1 = 249
setup_timer_2(T2_DIV_BY_16,249,1);//T=PWM Period = 2ms; DIV=16
set_timer2(0);

//value=(Period*Duty*Fosc)/(100%*DIV)=(2ms*90%*8MHz)/(100%*16)
set_pwm1_duty(900);//Duty =90%
setup_ccp1(CCP_PWM);
while(TRUE);
}

```

Câu 2: (3 điểm)

a) Vẽ mạch nguyên lý hoàn chỉnh (1 điểm)


```

set_tris_e(0x0F); //or 0xFF

setup_adc_ports(NO_ANALOGS|sAN5|sAN6);
setup_adc(ADC_CLOCK_INTERNAL);

channelNumber=0; //Channel = 1
output_b(Code7Seg[channelNumber+1]);
while(TRUE)
{
if(input(CHANNEL)==0)
{
delay_ms(50);
channelNumber=~channelNumber;
output_b(Code7Seg[channelNumber+1]);
while(input(CHANNEL)==0);
delay_ms(50);
}
else
if(channelNumber==0)
{
set_adc_channel(5);
delay_us(20);
adcValue=0;

for(loopCount=0;loopCount<100;loopCount++)
{
adcValue=read_adc()/2.046 + adcValue;
}
adcValue=adcValue/100;

output_c(Code7Seg[adcValue/10]);
output_d(Code7Seg[adcValue%10]);
}
else
{
set_adc_channel(6);
delay_us(20);
adcValue=0;
for(loopCount=0;loopCount<100;loopCount++)
{
adcValue=read_adc()/2.046 + adcValue;

```

```

}
adcValue=adcValue/100;

output_c(Code7Seg[adcValue/10]);
output_d(Code7Seg[adcValue%10]);
}
}
}

```

Câu 3: (4 điểm)

- a) Vẽ lưu đồ cho cả hai bộ vi điều khiển (2 điểm)
- Lưu đồ phải phù hợp với chương trình
 - Phải sử dụng động từ khi mô tả trong mỗi ký hiệu lưu đồ
 - Sử dụng đúng ký hiệu lưu đồ
 - Nếu lưu đồ sử dụng chương trình con thì phải vẽ ra lưu đồ của chương trình con đó
- b) Viết chương trình C cho cả hai bộ vi điều khiển (2 điểm)

Vi điều khiển 1:

```

#include<16F887.h>
#fuses HS
#use delay(CLOCK=11.0592MHz)
#use rs232(BAUD=9600,BITS=8,STOP=1,PARITY=N,RCV=PIN_C7,XMIT=PIN_C6)
unsigned int8 OldProduct,Product;
unsigned int8 Mode;
void main()
{
set_tris_c(0b10111111);

Mode=0;
set_timer1(0);
while(TRUE)
{
if(kbhit()==1)
{
Mode=getc();
if(Mode==0)
{
setup_timer_1(T1_DISABLED);
set_timer1(0);
}
else
if(Mode==1) setup_timer_1(T1_EXTERNAL|T1_DIV_BY_1);
else setup_timer_1(T1_DISABLED);
}
else;

```

```

if(Mode==1)
{
Product=get_timer1();output_d(Product);
if(Product==24+1)
{
Product=1;
set_timer1(1);
}
else;
if(Product!=OldProduct)
{
putc(Product);output_d(Product);
OldProduct=Product;
}
else;
}
else;
}
}

```

Vi điều khiển 2:

```

#include<16F887.h>
#fuses HS
#use delay(CLOCK=11.0592MHz)
#use rs232(BAUD=9600,BITS=8,STOP=1,PARITY=N,RCV=PIN_C7,XMIT=PIN_C6)
#include<lcd.c>

```

```

#define RUN  PIN_B6
#define STOP PIN_B7
//  RESET--MCLR

```

```

unsigned int8 Mode;
unsigned int8 Product;
unsigned int16 Box;
void main()
{
set_tris_c(0b10111111);
set_tris_b(0b11111111);
port_b_pullups(0b11111111);
Product=0;
Box=0;
Mode=0;putc(Mode);
lcd_init();

while(TRUE)
{
if(input(RUN)==0) {Mode=1;putc(Mode);}

```

```

else
if(input(STOP)==0) {Mode=2;putc(Mode);}
else;
if(kbhit()==1) Product=getc();else;
if(Product==24) Box++; else;
if(Mode==0)
{
lcd_gotoxy(1,1);
printf(lcd_putc,"P=%02u",Product);
lcd_gotoxy(12,1);
lcd_putc("RESET");

lcd_gotoxy(1,2);
printf(lcd_putc,"Box=%05Lu",Box);
}
else
if(Mode==1)
{
lcd_gotoxy(1,1);
printf(lcd_putc,"P=%02u",Product);
lcd_gotoxy(12,1);
lcd_putc("RUN ");

lcd_gotoxy(1,2);
printf(lcd_putc,"Box=%05Lu",Box);

}
else //(Mode==2)
{
lcd_gotoxy(1,1);
printf(lcd_putc,"P=%02u",Product);
lcd_gotoxy(12,1);
lcd_putc("STOP ");

lcd_gotoxy(1,2);
printf(lcd_putc,"Box=%05Lu",Box);
}

}
}

```

Ngày 21 tháng 05 năm 2017
 Thông qua bộ môn
(ký và ghi rõ họ tên)

- Giá trị trên LED bảy đoạn được cập nhật từ xa bằng ngắt UART. Máy tính được kết nối với bộ vi điều khiển qua cổng RS-232; giao thức: 8 bit dữ liệu, tốc độ 4800 bps, 1 STOP, không kiểm tra chẵn lẻ.
- Giá trị trên máy tính gửi xuống vi điều khiển ở dạng mã ASCII: '0', hoặc '1', hoặc '2', ..., hoặc '9' đại diện cho 0s, 1s, ..., 9s.
 - a) Vẽ mạch nguyên lý. (1.0 đ)
 - b) Viết chương trình. (2.0 đ)

Ghi chú: Cán bộ coi thi không được giải thích đề thi.

Chuẩn đầu ra của học phần (về kiến thức)	Nội dung kiểm tra
[G4.1]: Hiểu và ứng dụng được hệ thống vi điều khiển vào thực tế.	Câu 1, Câu 2, Câu 3
[G1.3]: Thiết kế và lập trình điều khiển dùng ngôn ngữ C cho các hệ thống điều khiển.	Câu 1, Câu 2, Câu 3
[G2.3]: Có kỹ năng thiết kế và xây dựng qui trình điều khiển cho hệ thống dùng vi điều khiển PIC 16F887.	Câu 1, Câu 2, Câu 3

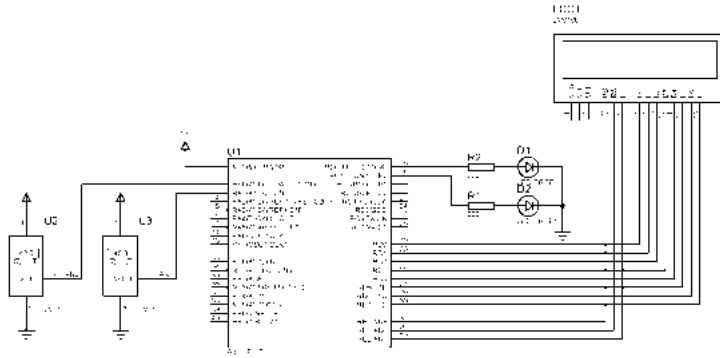
Ngày 03 tháng 06 năm 2016

Thông qua bộ môn
(ký và ghi rõ họ tên)

ĐÁP ÁN MÔN VI XỬ LÝ

THI NGÀY 21-12-2015

Câu 1:



```
#INCLUDE <TV_16F887.C>
#include <TV_LCD.C>
#define LEDA PIN_C0
#define LEDB PIN_C1
unsigned int8 DV_0, CH_0, DV_1, CH_1, I;
unsigned int16 KQADC_0, KQADC_1;

//VIET HAM CON
void GM_LCD()
{
    DV_0 = KQADC_0%10 + 0X30;
    CH_0 = KQADC_0/10 + 0X30;

    DV_1 = KQADC_1%10 + 0X30;
    CH_1 = KQADC_1/10 + 0X30;
}

void DOC_ADC_0()
{
    SET_ADC_CHANNEL(0);
    DELAY_US(20);
    KQADC_0 = 0;
    for(I=0; I<100; I++)
        KQADC_0 = KQADC_0 + READ_ADC();
    KQADC_0 = KQADC_0 / 100 / 2.046;
}

void DOC_ADC_1()
{
    SET_ADC_CHANNEL(1);
    DELAY_US(20);
    KQADC_1 = 0;
    for(I=0; I<100; I++)
        KQADC_1 = KQADC_1 + READ_ADC();
    KQADC_1 = KQADC_1 / 100 / 2.046;
}

void HT_LCD()
{
    //HIEN THI HANG 2
```

```
LCD_COMMAND(0XC0);
LCD_DATA(CH_0);
LCD_DATA(DV_0);
LCD_COMMAND(0XC0+14);
LCD_DATA(CH_1);
LCD_DATA(DV_1);
}

/*----- TINH STEP SIZE
Vref= (Vref+)-(Vref-)=5V.
So buoc nhay = 2^N-1 .
So bit ADC N=10 => So buoc nhay = 1023.
Do Phan Giai = Vref/(2^N-1)= 4.8876 mV.
/*-----*/

void MAIN()
{
    SET_TRIS_C(0X00); //LCD
    SET_TRIS_D(0X00);

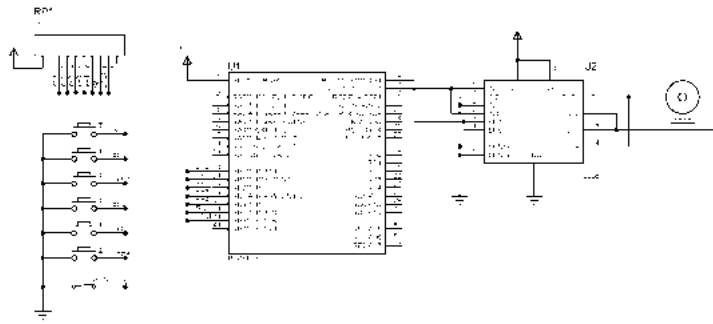
    //KHOI TAO ADC
    SETUP_ADC(ADC_CLOCK_DIV_32);
    SETUP_ADC_PORTS( SAN0 | SAN1 | VSS_VDD);
    //2 KENH

    //KHOI TAO LCD
    LCD_SETUP();
    LCD_COMMAND(0X80);
    LCD_DATA("NA-0902680065");

    while (TRUE)
    {
        KQADC_0=0;
        KQADC_1=0;
        DOC_ADC_0();
        DOC_ADC_1();
        if (KQADC_0 > 40)
            OUTPUT_HIGH(LEDA);
        else
            OUTPUT_LOW(LEDA);
        if (KQADC_1 > 45)
            OUTPUT_HIGH(LEDB);
        else
            OUTPUT_LOW(LEDB);

        GM_LCD();
        HT_LCD();
    }
}
```

Câu 2:



SSSS

```
#include <16F887.h>
#DEVICE ADC=10
#FUSES NOWDT, PUT, HS, NOPROTECT, NOLVP
#USE DELAY(CLOCK = 10M)

#define ENABLE  PIN_C0
#define BTN_1_6  PIN_B0
#define BTN_2_7  PIN_B1
#define BTN_3_8  PIN_B2
#define BTN_4_9  PIN_B3
#define BTN_5_10 PIN_B4
#define SW       PIN_B5
#define BTN_STOP PIN_B6

UNSIGNED INT16 MAXSPEED=500;

VOID KT_PHIM()
{
    //PHIM STOP
    IF(INPUT(BTN_STOP) == 0) OUTPUT_LOW(ENABLE);

    //PHIM BTN_1_6
    IF(INPUT(BTN_1_6) == 0)
    {
        OUTPUT_HIGH(ENABLE);
        IF(INPUT(SW) == 1) SET_PWM2_DUTY(MAXSPEED*6/10);
        ELSE              SET_PWM2_DUTY(MAXSPEED*1/10);
    }

    //PHIM BTN_2_7
    IF(INPUT(BTN_2_7) == 0)
    {
        OUTPUT_HIGH(ENABLE);
        IF(INPUT(SW) == 1) SET_PWM2_DUTY(MAXSPEED*7/10);
        ELSE              SET_PWM2_DUTY(MAXSPEED*2/10);
    }

    //PHIM BTN_3_8
    IF(INPUT(BTN_3_8) == 0)
    {
        OUTPUT_HIGH(ENABLE);
        IF(INPUT(SW) == 1) SET_PWM2_DUTY(MAXSPEED*8/10);
        ELSE              SET_PWM2_DUTY(MAXSPEED*3/10);
    }
}
```

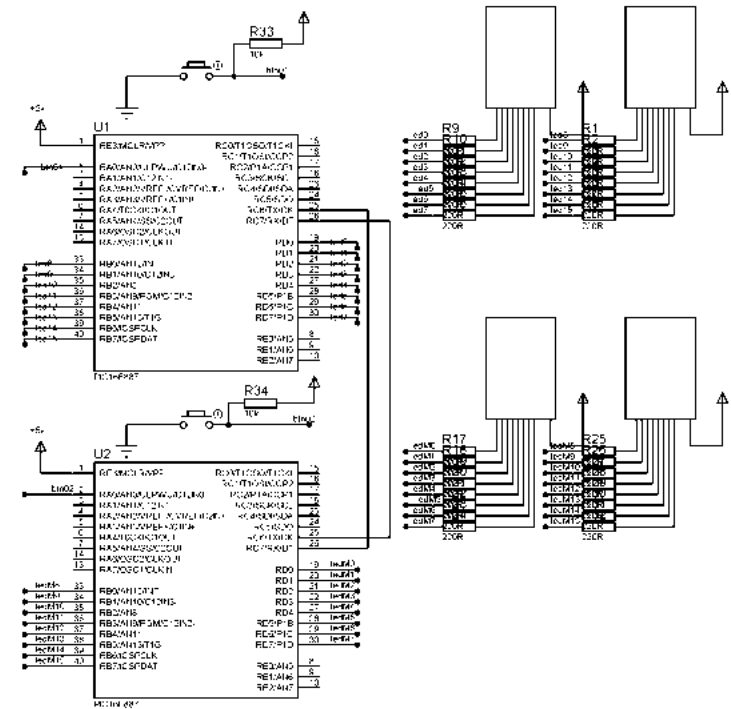
```
//PHIM BTN_4_9
IF(INPUT(BTN_4_9) == 0)
{
    OUTPUT_HIGH(ENABLE);
    IF(INPUT(SW) == 1) SET_PWM2_DUTY(MAXSPEED*9/10);
    ELSE              SET_PWM2_DUTY(MAXSPEED*4/10);
}

//PHIM BTN_5_10
IF(INPUT(BTN_5_10) == 0)
{
    OUTPUT_HIGH(ENABLE);
    IF(INPUT(SW) == 1) SET_PWM2_DUTY(MAXSPEED*10/10);
    ELSE              SET_PWM2_DUTY(MAXSPEED*5/10);
}
}

VOID MAIN()
{
    /*
    * Oscillator Frequency Fosc = 10000000
    * Clock Frequency Fclk = 2500000
    * PWM Freq = 1250 = 0.8ms
    * Prescaler Value = 16
    * PR2 = 124
    * Maximum duty value = 500
    */
    SET_TRIS_C(0X00);
    OUTPUT_LOW(ENABLE);

    SETUP_CCP2(CCP_PWM);
    SETUP_TIMER_2(T2_DIV_BY_16, 124, 16);
    WHILE(TRUE)
    {
        KT_PHIM();
    }
}
```


Câu 3:



VĐK A:

```
#INCLUDE <TV_16F887.C>
#USE RS232(BAUD = 9600, XMIT=PIN_C6, RCV=PIN_C7)
```

```
#define SP PIN_A0
unsigned char DEM;
```

```
VOID PHIM_SP()
{
    IF(INPUT(SP) == 0)
    {
        DELAY_MS(20);
        IF(INPUT(SP) == 0)
        {
            //XU LY CHUC NANG
            DEM++;
            IF(DEM == 100) DEM = 1;

            //TRUYEN SO SAN PHAM
            PUTC(DEM);

            WHILE(INPUT(SP) == 0);
        }
    }
}
```

```
VOID MAIN()
{
    SET_TRIS_D(0X00);
    SET_TRIS_B(0X00);
    SET_TRIS_A(0X01);

    DEM = 0;
    WHILE(TRUE)
    {
```

```
//NHAN DU LIEU
IF(KBHIT())
{
    DL_NHAN = GETCH();
    IF(DL_NHAN == 0)
        DEM = 0;
}

//KT PHIM
PHIM_SP();

//GIAI MA - HIEU THI
OUTPUT_D(MA7DOAN[DEM/10]);
OUTPUT_B(MA7DOAN[DEM%10]);
}
```

VĐK B:

```
#INCLUDE <TV_16F887.C>
#USE RS232(BAUD = 9600, XMIT=PIN_C6, RCV=PIN_C7)
#define RST PIN_A0
unsigned char DEM;
```

```
VOID PHIM_RST()
{
    IF(INPUT(RST) == 0)
    {
        DELAY_MS(20);
        IF(INPUT(RST) == 0)
        {
            //XU LY CHUC NANG
            PUTC(0);
            DEM = 0;
            WHILE(INPUT(RST) == 0);
        }
    }
}
```

```
VOID MAIN()
{
    SET_TRIS_D(0X00);
    SET_TRIS_B(0X00);
    SET_TRIS_A(0X01);
    DEM = 0;
    OUTPUT_D(MA7DOAN[DEM/10]);
    OUTPUT_B(MA7DOAN[DEM%10]);
    WHILE(TRUE)
    {
        //NHAN DU LIEU
        IF(KBHIT()) DEM = GETCH();

        //GIAI MA - HIEU THI
        OUTPUT_D(MA7DOAN[DEM/10]);
        OUTPUT_B(MA7DOAN[DEM%10]);

        //KT PHIM
        PHIM_RST();
    }
}
```

Câu 1: (3đ)

Một hệ thống đo nhiệt độ 2 kênh (A và B): sử dụng 1 vi điều khiển PIC16F887, 2 cảm biến LM35 (LM35_A và LM35_B), 1 LCD16x2, 2 LED đơn (LED_A và LED_B) với các yêu cầu như sau:

Hàng 1 của LCD hiển thị tên và mã số sinh viên của bạn.

Hàng 2 của LCD hiển thị nhiệt độ kênh A tận cùng bên trái, kênh B tận cùng bên phải.

Đo lần lượt từng kênh và đo trung bình 100 lần cho mỗi kênh.

Nhiệt độ kênh A lớn hơn 40 độ thì LED_A sáng, ngược lại tắt.

Nhiệt độ kênh B lớn hơn 45 độ thì LED_B sáng, ngược lại tắt.

- Hãy vẽ sơ đồ mạch giao tiếp vi điều khiển với cảm biến, LCD, LED, tùy chọn port. (0.5đ)
- Hãy tính toán độ phân giải, sử dụng điện áp tham chiếu $V_{ref+}=V_{dd}=5V$, $V_{ref-}=V_{ss}=0V$. (0.5đ)
- Hãy viết chương trình thực hiện các yêu cầu trên. (2đ)

Câu 2: (4đ)

Một hệ thống vi điều khiển PIC16F887 dùng PWM của khối CCP2 để điều khiển 1 động cơ (ĐC) thay đổi tốc độ 10 cấp (không tính cấp 0), dùng IC giao tiếp công suất L298, 6 nút nhấn thường hở (BTN_1_6, BTN_2_7, BTN_3_8, BTN_4_9, BTN_5_10, BTN_STOP) và 1 switch gạt SW (tạo 2 mức logic 0 và 1 tương ứng 2 vị trí on và off). Chu kỳ PWM là 0.8ms, thạch anh sử dụng là 10MHz.

Khi mới cấp điện hoặc khi nhấn BTN_STOP thì động cơ ngừng.

Khi SW ở vị trí on và nếu nhấn BTN_1_6 thì động cơ sẽ chạy cấp tốc độ 1.

Khi SW ở vị trí off và nếu nhấn BTN_1_6 thì động cơ sẽ chạy cấp tốc độ 6.

Tương tự cho các nút nhấn còn lại (2 số theo sau chính là tốc độ 2 cấp tương ứng).

- Hãy vẽ sơ đồ mạch giao tiếp vi điều khiển với L298, động cơ, nút nhấn và switch. (0.5đ)
- Hãy tính toán các thông số PR2, PV và hằng số tương ứng với tốc độ cực đại. (0.5đ)
- Hãy vẽ lưu đồ giải thuật điều khiển theo yêu cầu. (1đ)
- Hãy viết chương trình. (2đ)

Câu 3: (3đ)

Một hệ thống quản lý bao gồm hệ thống A và hệ thống B như sau:

Hệ thống A: có chức năng đếm số sản phẩm thi công, khi thi công xong thì công nhân sẽ nhấn 1 cái nút để số sản phẩm tăng lên 1. Số sản phẩm nằm trong giới hạn từ 00 đến 99 (khi bằng 99 và nếu nhấn nữa thì quay về 1) hiển thị trên 2 led 7 đoạn anode chung.

Hệ thống B: Số lượng sản phẩm từ hệ thống A được gởi về phòng quản lý để hiển thị trên 2 led 7 đoạn anode chung. Có 1 nút nhấn reset và khi nhấn thì sẽ xóa sản phẩm đếm được về 0 của cả 2 hệ thống.

- a. Hãy thiết kế phần cứng cho mỗi hệ thống đều dùng vi điều khiển PIC16F887 và tùy chọn port, hai hệ thống giao tiếp với nhau dùng chuẩn UART, tốc độ 9600 baud. (0.75đ)
- b. Hãy vẽ lưu đồ cho 2 vi điều khiển. (0.5đ)
- c. Hãy viết các chương trình thực hiện các yêu cầu trên. (1.75đ)

Chú ý không sử dụng counter để đếm cho hệ thống A.

Cán bộ coi thi không được giải thích đề thi, cho đề thi vào túi bài thi để chấm bài.

Chuẩn đầu ra của học phần (về kiến thức)	Nội dung kiểm tra
[G 4.1]: Hiểu và ứng dụng được hệ thống vi điều khiển vào thực tế.	Câu 1
[G 1.3]: Thiết kế và lập trình điều khiển dùng ngôn ngữ C cho các hệ thống điều khiển	Câu 2
[G 2.3]: Có kỹ năng thiết kế và xây dựng qui trình điều khiển cho hệ thống dùng vi điều khiển PIC 16F887.	Câu 3

Tp. Hồ Chí Minh, ngày 15 tháng 12 năm 2015
Thông qua bộ môn


NGUYỄN ĐÌNH PHÚ

ĐẠI HỌC SƯ PHẠM KỸ THUẬT TP.HCM
KHOA: ĐIỆN - ĐIỆN TỬ
BỘ MÔN: ĐIỆN TỬ CÔNG NGHIỆP

ĐỀ THI MÔN: VI XỬ LÝ
Mã môn học: MICR330363
ĐỀ SỐ: 01 Đề thi có 01 trang
Thời gian: 90 phút

SV chỉ được tham khảo trong 01 tờ giấy A4
mang theo, được viết tay 2 mặt, không photo.

Lưu ý: Tất cả các câu trong đề thi này đều sử dụng bộ vi điều khiển 16F887 dùng dao động 8MHz (tùy chọn trong hoặc ngoài). Lập trình bằng PCWH Compiler của hãng CCS.

Câu 1: (3 điểm)

Xét một ứng dụng có một Led đơn (được nối với RB0) và hai nút nhấn ON và OFF (tương ứng được nối với RB1 và RB2). Khi mới bật điện, mặc định Led tắt. Nhấn ON, Led nhấp nháy liên tục theo chu kỳ: Sáng 1s, Tắt 0.5s. Nhấn OFF, Led bị tắt ngay.

- a) Vẽ lưu đồ. (1.5đ)
- b) Viết chương trình. (1.5đ)

Câu 2: (3 điểm)

Xét một ứng dụng đo nhiệt độ và tạo xung PWM. Sử dụng cảm biến LM35, được nối với kênh AN5, trong môi trường có nhiệt độ phạm vi $0^{\circ}\text{C} < T < 100^{\circ}\text{C}$. Xung PWM, dùng ngõ ra CCP1, có chu kỳ xung 1 ms và chu kỳ công tác 70%. Giá trị nhiệt độ được hiển thị trên 2 LED bảy đoạn (được nối với PORT B cho chữ số hàng chục và PORT D cho chữ số hàng đơn vị). Xung PWM chỉ được xuất ra khi nhiệt độ vượt quá 50°C , ngược lại thì ngõ ra PWM ở mức thấp.

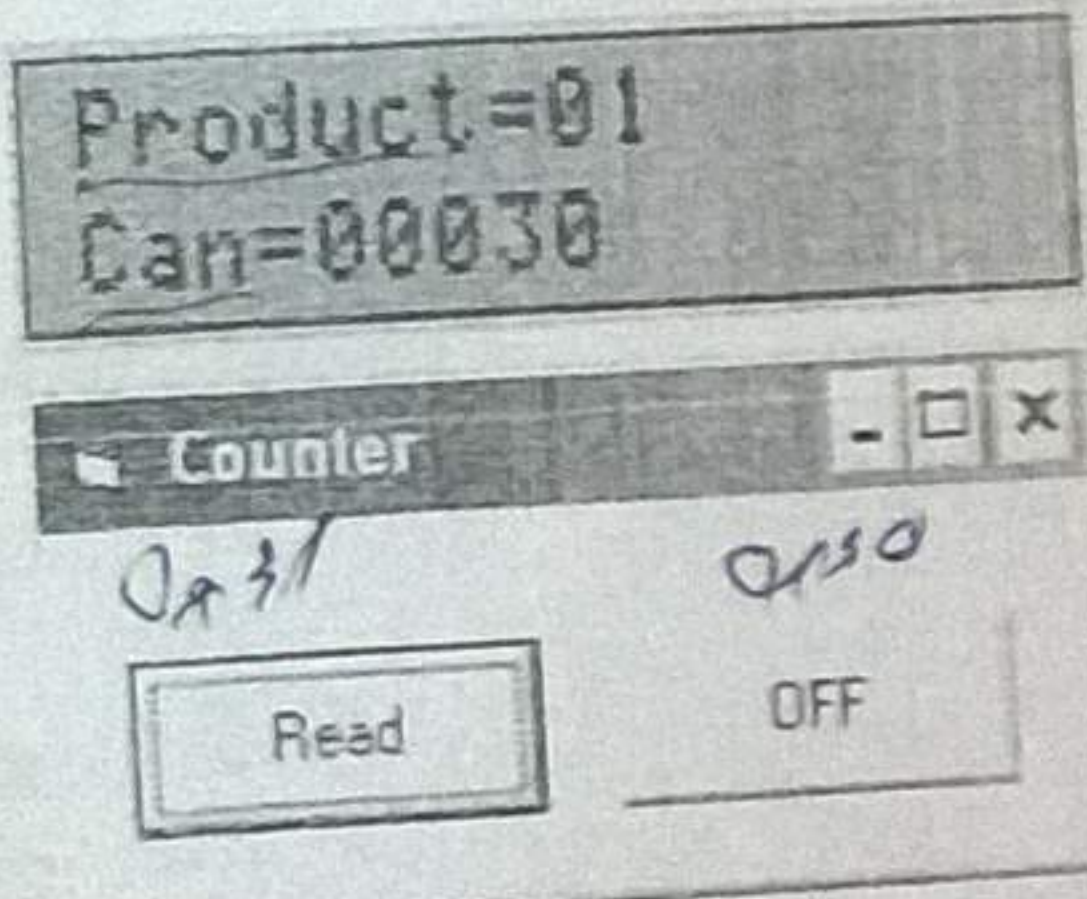
- a) Vẽ lưu đồ. (1.0đ)
- b) Viết chương trình. (1.5đ)
- c) Đánh giá sai số tuyệt đối của xung PWM. (0.5đ)

$$\frac{1000}{0.5 \cdot \text{peak}} \leq \Delta t$$

$$\frac{1}{10} \leq \Delta t$$

Câu 3: (4 điểm)

Xét một ứng dụng đếm sản phẩm có giao tiếp máy tính. Mỗi thùng sản phẩm có 24 sản phẩm. LCD 16x2 để hiển thị kết quả đếm sản phẩm (Product) và thùng (Can), được cập nhật liên tục. Minh họa giao diện LCD như hình bên.



Giao diện trên máy tính như sau: Ngay sau khi nút Read (mã lệnh ASCII = '1'=31h) được nhấn thì kết quả đếm được gửi lên máy tính cho đến khi nào nút OFF (mã lệnh ASCII = '0'=30h) được nhấn thì không gửi nữa. Khi mới bật điện mặc định không gửi giá trị đếm lên máy tính.

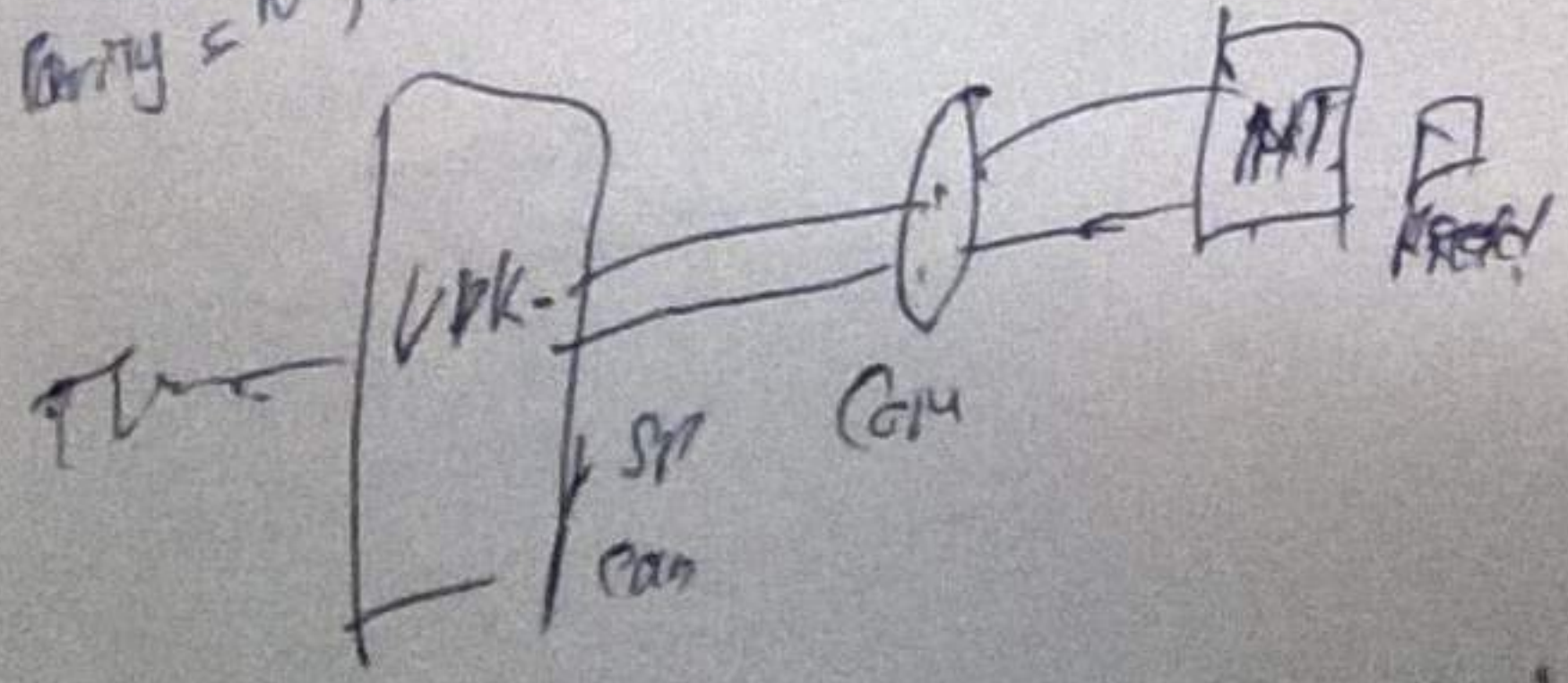
Giao thức truyền UART: tốc độ 9600, bps; 8 bit dữ liệu; 1 bit STOP; không kiểm tra lỗi chẵn lẻ. parity \neq chẵn
Giao tiếp LCD sử dụng PORT D và PORT E nếu chọn kiểu 8 bit. Nếu kiểu 4 bit thì chỉ sử dụng PORT D.
Cảm biến đếm sản phẩm được kết nối đến ngõ vào Counter 0.

~~# Use UART (baud = 9600, parity = N, xmit = pin 16, rx = pin 17)~~
~~bits = 8, stop = 1~~ // mặc định đã vậy

- a) Vẽ mạch nguyên lý hoàn chỉnh. (2.0đ)
- b) Viết chương trình. (2.0đ)

Cán bộ coi thi không được giải thích đề thi.
Tp. Hồ Chí Minh, ngày 12 tháng 12 năm 2014
P. Trưởng bộ môn

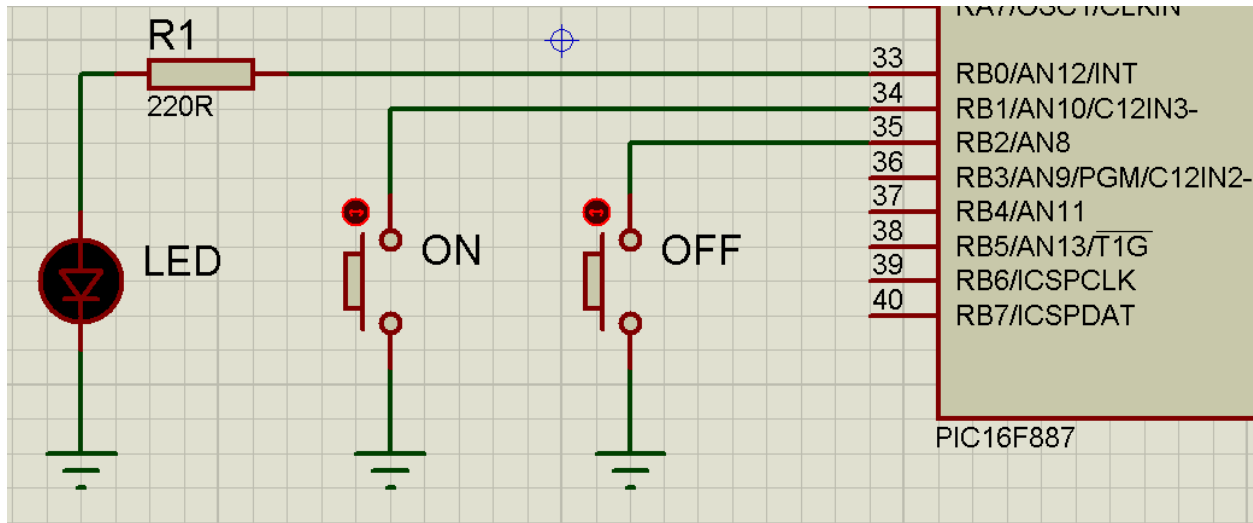
Use RS232 (baud = 9600, bits = 8, stop = 1, parity = N, rx)



Nguyễn Đức Phao

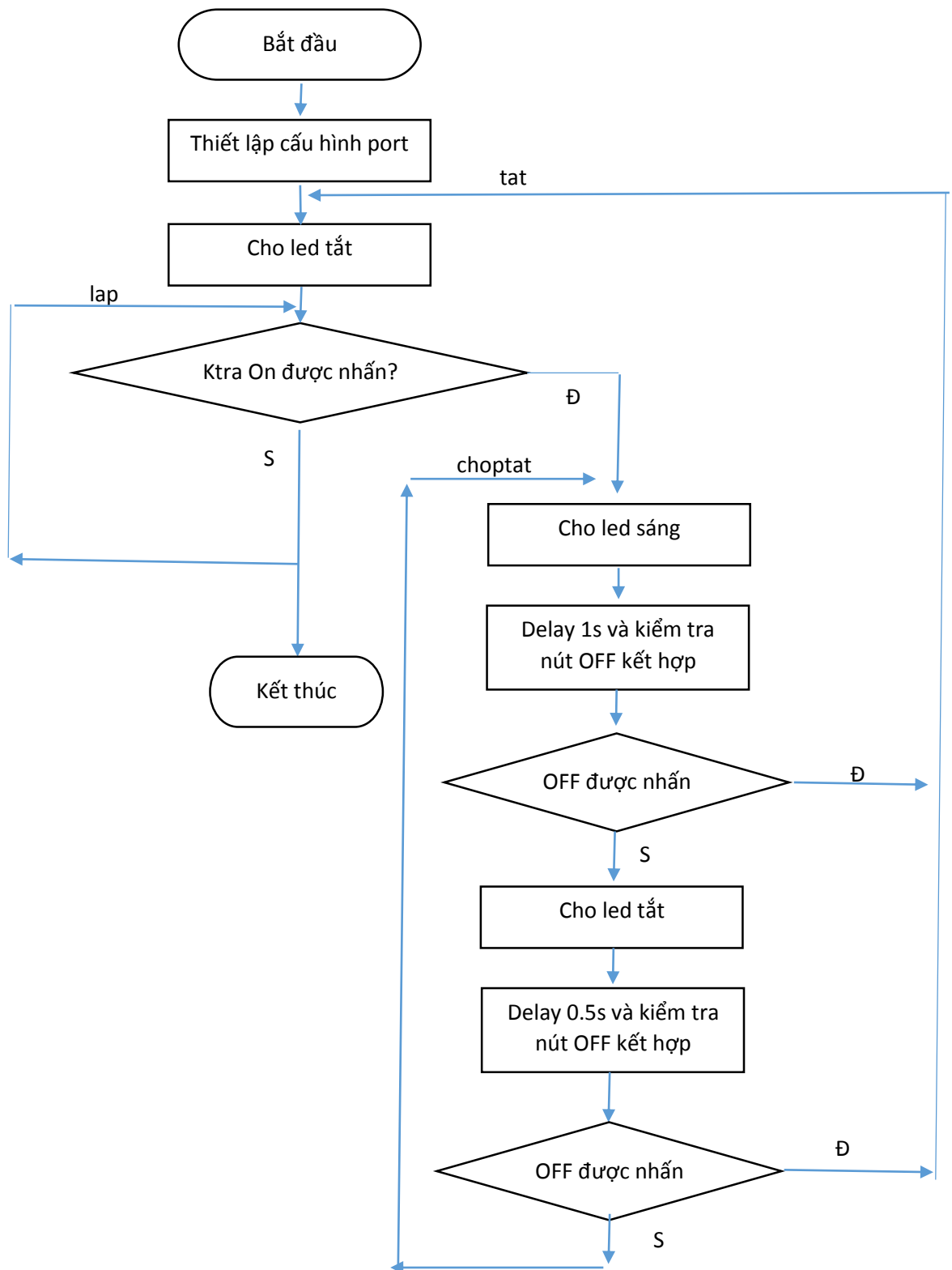
Câu 1:

Sơ đồ phần cứng như sau: Phần này không chấm điểm



a) Lưu đồ chương trình. (1.5đ)

Tùy theo thuật toán của sinh viên mà lưu đồ có thể khác nhau, dưới đây là một lưu đồ gợi ý



b. (1.5 điểm) Chương trình có thể viết theo nhiều cách khác nhau. Dưới đây là gợi ý hai cách viết.

Cách 1: (Viết theo lưu đồ)

Kiểm tra các nút nhấn ngay trong chương trình chính, không cần dùng ngắt

```

#include <16F887.h>
#FUSES NOWDT,PUT,HS,NOPROTECT,NOLVP
#USE DELAY(CLOCK=8M)
#define LED PIN_B0
#define ON PIN_B1
#define OFF PIN_B2

unsigned int8 K;

void main()
{
    set_tris_b(0xFE);
    port_b_pullups(0xFE);
TAT:
    output_low(LED);
LAP:
    if (input(ON)==0)
    {
CHOPTAT:
        output_high(LED);
        for (K=0;K<100;K++)
        {
            delay_ms(10);
            if (input(OFF)==0) goto TAT;
        }
        output_low(LED);
        for (K=0;K<50;K++)
        {
            delay_ms(10);
            if (input(OFF)==0) goto TAT;
        }
        goto CHOPTAT;
    }

    goto LAP;
}

```

Cách 2: (Gợi ý Viết cách khác không theo lưu đồ)

Kiểm tra các nút nhấn trong chương trình ngắt:

```

#include <16F887.h>
#FUSES NOWDT,PUT,HS,NOPROTECT,NOLVP
#USE DELAY(CLOCK=8M)
#define LED PIN_B0
#define ON PIN_B1
#define OFF PIN_B2
unsigned int1 TT;
#INT_TIMER1

```

```

VOID KTRA_NUTNHAN()
{
    IF (INPUT(ON)==0) TT=1;
    IF (INPUT(OFF)==0) TT=0;
}
VOID MAIN()
{
    SET_TRIS_B(0XFE);
    PORT_B_PULLUPS(0XFE);

    ENABLE_INTERRUPTS(GLOBAL);
    ENABLE_INTERRUPTS(INT_TIMER1);

    SETUP_TIMER_1(T1_INTERNAL|T1_DIV_BY_8);
    SET_TIMER1(0);

    TT=0;

    WHILE(TRUE)
    {
        IF (TT==0)
        {
            OUTPUT_LOW(LED);
        }
        ELSE
        {
            OUTPUT_HIGH(LED); DELAY_MS(1000);
            OUTPUT_LOW(LED); DELAY_MS(500);
        }
    }
}

```

Câu 2:

a) Vẽ lưu đồ: (1đ) Sinh viên có thể vẽ theo nhiều cách khác nhau. Lưu ý nếu sinh viên vẽ lưu đồ tóm tắt thì sinh viên phải vẽ thêm lưu đồ cho các chương trình con(nếu cần thiết).

b)

- Sinh viên viết được chương trình thực hiện chuyển đổi ADC đúng kênh, tính được giá trị nhiệt độ : 0.5 điểm.

- Viết được chương trình hiển thị nhiệt độ lên Led 7 đoạn 0.25 điểm.

- Khi nhiệt độ vượt quá 50 độ viết được chương trình tạo xung PWM 1ms, DC=0.7 được 0.5 điểm

Chú ý ba hàm:

- Hàm tạo xung PWM có chu kỳ mong muốn

SETUP_TIMER_2(T2_DIV_BY_16, PERIOD,1);

XTAL=8MHZ → CHIA 4 CÒN LẠI 2 MHZ → CHU KỲ MÁY = 0.5 US

THỰC HIỆN HIỆN BỘ CHIA TRƯỚC CHO TIMER2 LÀ 16 → CHU KỲ XUNG ĐƯA ĐẾN BỘ TIMER2 LÀ $0.5 \times 16 = 8\mu\text{s}$

ĐỂ TẠO XUNG PWM CÓ CHU KỲ $1\text{ms} = 1000\mu\text{s}$ THÌ SỐ XUNG TIMER2 PHẢI ĐẾM SAU MỖI LẦN RESET XUNG PWM LÀ 125 XUNG. DO ĐÓ TIMER2 PHẢI ĐẾM TỪ 0 ĐẾN 124. VÌ VẬY GIÁ TRỊ CỦA THÔNG SỐ PERIOD LÀ 124.

Ghi lại lệnh trên : **SETUP_TIMER_2(T2_DIV_BY_16, 124, 1) ;**

- Hàm thiết lập Duty Cycle:

Để tạo hệ số công tác ta sử dụng lệnh: **SET_PWM1_DUTY(value);**

Trong đó value được tính như sau:

Duty cycle(DC) = $\text{value} / (4 * (\text{số xung timer2 phải đếm để reset xung PWM}))$

Value = $0.7 * (4 * 125) = 350$

- Hàm tắt PWM:

SETUP_CCP1(CCP_OFF);

c. (0.5 đ) Có thể sinh viên chọn hệ số chia trước cho Timer2 khác nên có thể tạo ra sai số PWM. Sinh viên tính sai số này.

Câu 3:

a. Vẽ mạch nguyên lý:

- Vẽ đúng các chân kết nối vi điều khiển đến LCD theo yêu cầu : 0.5 điểm.
- Vẽ đúng kết nối truyền dữ liệu đến máy tính thông qua IC RS232 được 0.5 điểm.
- Vẽ cảm biến đếm sản phẩm đúng chân ngõ vào Counter 0: 0.5 điểm.
- Vẽ đầy đủ các kết nối cơ bản cho vi điều khiển: reset, nguồn, dao động : 0.5 điểm

b. Viết chương trình

- Các phần khai báo các thư viện, delay, FUSE, USE RS232, các biến cần thiết : 0.5 điểm
- Cấu hình timer0 chế độ Counter đếm xung ngoài và đọc được giá trị sản phẩm : 0.5 điểm
- Tính toán và giải mã hiện thị lên LCD số sản phẩm, số thùng: 0.5 điểm
- Viết chương trình xử lý truyền nhận dữ liệu với máy tính: 0.5 điểm

Giáo viên soạn đáp án

Câu 1: (3 điểm)

Cho vi điều khiển Pic16F887 kết nối với một cảm biến A (tại chân RB0), mạch điều khiển đèn B sử dụng điện áp 220VAC (tại chân RB1), mạch điều khiển loa 32 Ohm, 0.5W (tại chân RB2) và một nút nhấn thường hở tên là RST (tại chân RB3). Vi điều khiển sử dụng thạch anh 12 Mhz.

Khi có người xuất hiện trong vùng hoạt động của cảm biến A, ngõ ra cảm biến lên mức 5V (bình thường không có người ngõ ra 0V).

Vi điều khiển xuất mức logic 1 để điều khiển đèn B sáng, mức logic 0 để điều khiển đèn B tắt.

Ban đầu đèn tắt, loa không kêu. Nếu có người xuất hiện trong vùng hoạt động của cảm biến ngay lập tức đèn B sáng, loa được điều khiển phát ra âm thanh với tần số **chính xác** 5Khz, Duty Cycle = 40% (dùng timer tạo xung). Trạng thái này được duy trì cho dù người đó có còn trong vùng hoạt động của cảm biến nữa hay không.

Bất cứ khi nào nhấn nút RST thì đèn tắt, loa không kêu.

a. Vẽ lưu đồ chương trình (1 điểm)

b. Viết chương trình cho vi điều khiển thực hiện đúng yêu cầu trên (2 điểm)

Câu 2: (3.5 điểm)

Cho hệ thống đếm sản phẩm: Vi điều khiển Pic16F887 kết nối một cảm biến phát hiện sản phẩm, LCD 16x2, hai nút nhấn thường hở UP, DW. Bình thường ngõ ra cảm biến là 0V, khi có sản phẩm chắn ngang cảm biến ngõ ra sẽ lên mức 5V.

Giao diện hiển thị của LCD như sau:

GIA TRI DAT: AA SO SAN PHAM: BB

Giá trị đặt AA là số sản phẩm của một thùng cần đóng gói. Giá trị AA có thể điều chỉnh được bởi hai nút nhấn UP (tăng 1 đơn vị), DW (giảm 1 đơn vị). AA nằm trong phạm vi [20-30]. Mặc định AA = 20.

Giá trị BB là số phẩm đếm được.

Khi sản phẩm đếm được BB bằng AA thì LCD sẽ nhấp nháy (sáng 0.5s, tắt 0.5s) chỉ hai dòng chữ như sau:

DA DU SO LUONG DE NGHI DONG GOI

a. Vẽ mạch nguyên lý kết nối (1 điểm) (phần cứng kết nối sinh viên tùy chọn chân thích hợp, LCD có thể giao tiếp với vi điều khiển theo kiểu 8bit dữ liệu hoặc 4 bit dữ liệu. Khối cảm biến chỉ cần vẽ khối, không cần vẽ chi tiết mạch cảm biến).

b. Viết chương trình thực hiện yêu cầu trên (2.5 điểm)

Câu 3: (3.5 điểm)

Cho hai vi điều khiển Pic16F887 (được đặt tên VDKA và VDKB) giao tiếp truyền dữ liệu nối tiếp bất đồng bộ, 8 bit dữ liệu, tốc độ baud = 2400. Cả hai vi điều khiển đều sử dụng thạch anh 12MHz.

VDKA kết nối với hai cảm biến nhiệt độ LM35 tại chân AN3 (kênh 3), AN4 (kênh 4). Đầu tiên VDKA đo nhiệt độ kênh 3 và gởi dữ liệu qua VDKB, một giây sau VDKA đo nhiệt độ kênh 4 và gởi dữ liệu qua VDKB, một giây sau nữa thì VDKA lại quay lại đo nhiệt độ kênh 3 và gởi dữ liệu đi... Quá trình cứ lặp đi lặp lại. Phạm vi nhiệt độ đo từ 0-99⁰C.

VDKB kết nối với 3 led 7 đoạn được đánh số 1, 2, 3 (ba led 7 đoạn kết nối theo kiểu trực tiếp đến vi điều khiển). Led 3 hiển thị kênh nhiệt độ đang đo, Led 1,2 hiển thị giá trị nhiệt độ đo được. Dữ liệu kênh đo và nhiệt độ đo được nhận từ VDKA gởi qua.

Yêu cầu:

- a. Vẽ mạch nguyên lý kết nối (1 điểm)
- b. Viết chương trình cho VDKA (1.5 điểm)
- c. Viết chương trình cho VDKB (1 điểm)

Ghi chú: Cán bộ coi thi không được giải thích đề thi.

Chuẩn đầu ra của học phần (về kiến thức)	Nội dung kiểm tra
[G 4.1]: Hiểu và ứng dụng được hệ thống vi điều khiển vào thực tế.	Câu 1
[G 1.3]: Thiết kế và lập trình điều khiển dùng ngôn ngữ C cho các hệ thống điều khiển.	Câu 2
[G 2.3]: Có kỹ năng thiết kế và xây dựng qui trình điều khiển cho hệ thống dùng vi điều khiển PIC 16F887.	Câu 3

Ngày tháng năm 20

Thông qua bộ môn

(ký và ghi rõ họ tên)

ĐÁP ÁN ĐỀ THI CUỐI KỲ HỌC KỲ2 - NĂM HỌC 2015

Môn: Vi xử lý (Điện)

Mã môn học: 1146050

CÂU 1: 3 ĐIỂM

a. Vẽ lưu đồ chương trình (1 điểm)

- Về hình thức: lưu đồ có đầy đủ các phần Bắt đầu, xử lý và kết thúc đúng ký hiệu chuẩn : **0.25 điểm**.
- Có nhiều cách vẽ lưu đồ khác nhau, tùy theo thuật toán xử lý của sinh viên. Nếu lưu đồ đúng trọn vẹn nội dung thuật toán: **0.75 điểm**. Các mức độ thấp hơn sẽ do giáo viên chấm quyết định.

b. Chương trình (2 điểm)

Sinh viên có thể viết chương trình theo lưu đồ giải thuật, hoặc nếu không vẽ được lưu đồ, chương trình vẫn được chấm điểm độc lập.

Sinh viên có thể dùng ngắt do tràn timer để tạo xung hoặc dùng delay timer để tạo xung.

- Từ thạch anh 12Mhz, sinh viên tính ra được số xung timer sẽ đếm khi xung ở mức cao và ở mức thấp, viết được chương trình tạo xung: 0.75 điểm.
- Khai báo thư viện, định nghĩa các chân kết nối hợp lý: 0.25 điểm.
- Thiết lập được trạng thái ban đầu của đèn, loa : 0.25 điểm.
- Viết chương trình xử lý điều khiển đèn, loa khi có người: 0.75 điểm. (Nếu có sử dụng ngắt, nếu khai báo đúng ngắt: 0.25 điểm).

Dưới đây là một chương trình mẫu, sinh viên có thể tham khảo:

```
#INCLUDE <16F887.H>
#FUSES NOWDT,HS,PUT,NOPROTECT,NOLVP
#USE DELAY(CLOCK=12M)
#DEFINE SENSOR PIN_B0
#DEFINE LAMP PIN_B1
#DEFINE SPEAKER PIN_B2
#DEFINE RST PIN_B3

#INT_TIMER1
VOID TAOXUNG5KH()
{
    IF (INPUT_STATE(SPEAKER)==0)
    { SET_TIMER1(65296);}
    ELSE
    { SET_TIMER1(65176);}
    OUTPUT_TOGGLE(SPEAKER);
}

VOID MAIN()
{
    SET_TRIS_B(0B00001001);
    ENABLE_INTERRUPTS(GLOBAL);
    ENABLE_INTERRUPTS(INT_TIMER1);
    TD:
    SETUP_TIMER_1(T1_DISABLED);
```

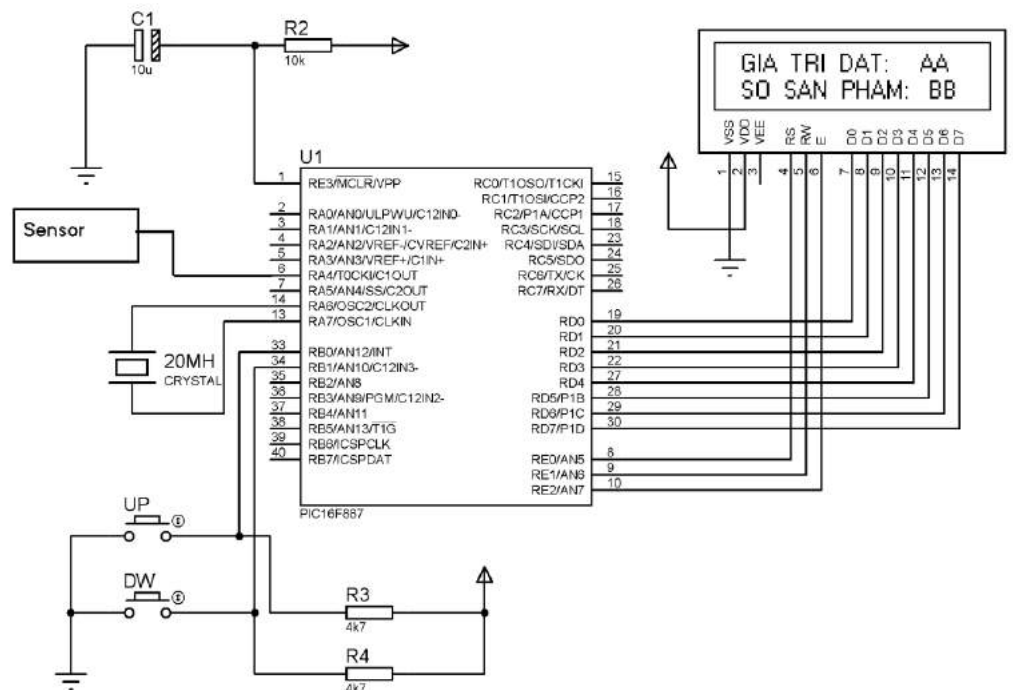
```

OUTPUT_LOW(LAMP);
OUTPUT_LOW(SPEAKER);
LAP:
IF (INPUT(SENSOR)==1)
{ OUTPUT_HIGH(LAMP);
  SETUP_TIMER_1(T1_INTERNAL|T1_DIV_BY_1);
  SET_TIMER1(65176);
  WHILE(INPUT(RST)==1) {}
  GOTO TD;
}
GOTO LAP;
}

```

CÂU 2: 3.5 điểm

- Vẽ mạch nguyên lý kết nối: 1 điểm. Sinh viên có thể chọn cách kết nối phù hợp.
 - Vẽ kết nối đúng với LCD : 0.5 điểm.
 - Vẽ kết nối cho vđk các chân cấp nguồn, reset, thạch anh, nút nhấn, cảm biến sản phẩm: 0.5 điểm
 - Dưới đây là một mạch kết nối tham khảo, chân vcc và gnd của vi điều khiển được lược bỏ do vẽ bằng protues, sinh viên phải vẽ đầy đủ.



- Viết chương trình: 2.5 điểm

- Viết được cái thư viện, khai báo chân sử dụng: 0.25 điểm
- Khai báo được cái biến phục vụ cho hiển thị LCD và thư viện LCD: 0.25 điểm
- Cấu hình timer thích hợp đếm xung ngoài, hệ số chia, cho phép timer đếm: 0.25 điểm
- Viết chương trình LCD hiển thị được cái dòng chữ cố định: “GIA TRI DAT:”, “SO SAN PHAM:” : 0.25 điểm
- Viết chương trình xử lý UP-DW: 0.5 điểm.
- Viết chương trình xử lý chớp tắt dòng chữ trên LCD khi AA=BB: 0.5 điểm

- Lắp ráp hoàn chỉnh chương trình: 0.5 điểm.

Chương trình tham khảo bên dưới:

```
#INCLUDE <16F887.H>
#FUSES NOWDT,HS,PUT,NOPROTECT,NOLVP
#USE DELAY(CLOCK=20M)
#DEFINE OUTPUT_LCD OUTPUT_D
#DEFINE CLEAR 0X01
#DEFINE UP PIN_B0
#DEFINE DW PIN_B1
#INCLUDE <TV_LCD.C>

CONST UNSIGNED INT8 MALCD[10]={"0123456789"};
CONST UNSIGNED INT8 LINE1_1[13]="GIA TRI DAT: ";
CONST UNSIGNED INT8 LINE2_1[13]="SO SAN PHAM: ";
CONST UNSIGNED INT8 LINE1_2[16]=" DA DU SO LUONG ";
CONST UNSIGNED INT8 LINE2_2[16]="DE NGHI DONG GOI";

UNSIGNED INT8 AA,BB,K;

VOID HTHI_AA()
{
    LCD_COMMAND(0X8D);
    LCD_DATA(MALCD[AA/10]);
    LCD_DATA(MALCD[AA%10]);
}

VOID HTHI_BB()
{
    LCD_COMMAND(0XCD);
    LCD_DATA(MALCD[BB/10]);
    LCD_DATA(MALCD[BB%10]);
}

VOID MAIN()
{
    SET_TRIS_D(0X00);
    SET_TRIS_E(0X00);
    SET_TRIS_B(0XFF);

    PORT_B_PULLUPS(TRUE);

    SETUP_TIMER_0(TO_EXT_H_TO_L|TO_DIV_1);
    SET_TIMER0(0);

    LCD_SETUP();

    LCD_COMMAND(0X80);
    FOR(K=0;K<13;K++)
    { LCD_DATA(LINE1_1[K]);}

    LCD_COMMAND(0XC0);
```

```

FOR(K=0;K<13;K++)
{ LCD_DATA(LINE2_1[K]);}

AA=20;
HTHI_AA();

LOOP2:
BB=GET_TIMER0();
HTHI_BB();

IF(INPUT(UP)==0)
{
    DELAY_MS(10);
    IF(INPUT(UP)==0)
    { WHILE(INPUT(UP)==0) {}
      IF (AA<30) AA++;
      HTHI_AA();
    }
}
IF(INPUT(DW)==0)
{
    DELAY_MS(10);
    IF(INPUT(DW)==0)
    { WHILE(INPUT(DW)==0) {}
      IF (AA>20) AA--;
      HTHI_AA();
    }
}

IF(BB==AA)
{
    LOOP1:
    LCD_COMMAND(0X80);
    FOR(K=0;K<16;K++)
    {
        LCD_DATA(LINE1_2[K]);
    }
    LCD_COMMAND(0XC0);
    FOR(K=0;K<16;K++)
    {
        LCD_DATA(LINE2_2[K]);
    }
    DELAY_MS(500);

    LCD_COMMAND(CLEAR);
    DELAY_MS(500);

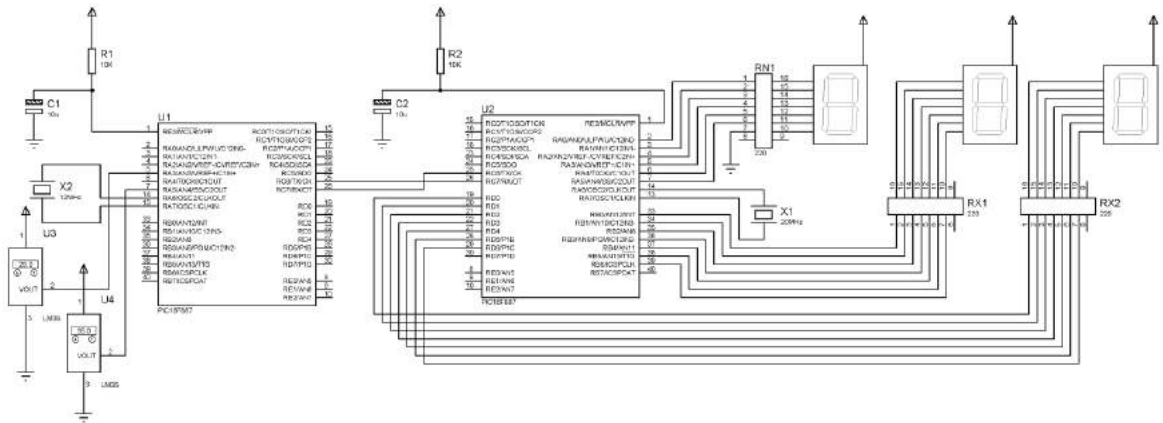
    GOTO LOOP1;
}
GOTO LOOP2;
}

```

CÂU 3: 3.5 điểm

a. Vẽ mạch nguyên lý: 1 điểm

- Vẽ được vi điều khiển A kết nối 2 cảm biến nhiệt độ đúng chân yêu cầu, cấp nguồn, thạch anh, reset đầy đủ, hai chân truyền dữ liệu qua VDKB: 0.5 điểm
- Vẽ được VDKB kết nối các chân cần thiết, kết nối đến 3 led 7 đoạn, ta dùng hai port còn trống đủ chân kết nối 2 led hiển thị nhiệt độ, led hiển thị kênh chỉ hiển thị hai kênh 3,4 nên ta kết nối nó với port ko còn đáp ứng đủ chân, ở đây dùng Port A. Port A mất hai chân kết nối thạch anh, do đó chân G của led 7 đoạn này ta luôn nối GND, chân DP ta luôn nối VCC (bởi vì mã 7 đoạn của số 3, 4 hai đoạn này giống nhau): 0.5 điểm



b. Chương trình cho VDKA

- Khai báo đầy đủ thư viện, cầu chì, device adc, rs232.. : 0.5 điểm
- Khai báo đúng các hàm thiết lập ADC: 0.25 điểm
- Thực hiện và chuyển đổi ADC của từng kênh gọi đi, gọi thông tin kênh và thông tin nhiệt độ, cách nhau 1s: 0.75 điểm

Chương trình tham khảo bên dưới

```
#INCLUDE <16F887.H>
#DEVICE ADC=10
#FUSES NOWDT,HS,PUT,NOPROTECT,NOLVP
#USE DELAY(CLOCK=12M)
#USE RS232(baud=2400, xmit=PIN_C6,rcv=PIN_C7)
```

```
UNSIGNED INT8 KENH, ND;
UNSIGNED INT16 DLADC;
```

```
VOID MAIN()
{
    SETUP_ADC(ADC_CLOCK_INTERNAL);
    SETUP_ADC_PORTS(SAN3|SAN4|VSS_VDD);
    WHILE(TRUE)
    {
        FOR(KENH=3;KENH<5;KENH++)
        {
            SET_ADC_CHANNEL(KENH);
            DELAY_MS(10);
            DLADC=READ_ADC();
            ND=DLADC/2.046;
```



```

        PUTC(KENH);
        PUTC(ND);
        DELAY_MS(1000);
    }
}
}

```

c. Chương trình cho VDKB:1 điểm

- Khai báo đúng thư viện, các biến...: 0.25 điểm
 - Viết được chương trình hiển thị 3 led 7 đoạn:0.25 điểm
 - Viết chương trình nhận lần lượt thông tin từ VDKB gửi qua: 0.5 điểm
- Chương trình tham khảo bên dưới

```

#include <16F887.H>
#FUSES NOWDT,HS,PUT,NOPROTECT,NOLVP
#USE DELAY(CLOCK=12M)
#USE RS232(baud=2400, xmit=PIN_C6,rcv=PIN_C7)

UNSIGNED INT8 KENH, ND;
CONST UNSIGNED INT8
MA7DOAN[10]={0XC0,0XF9,0XA4,0XB0,0X99,0X92,0X82,0XF8,0X80,0X90};

VOID HIENTHI()
{
    OUTPUT_A(MA7DOAN[KENH]);
    OUTPUT_B(MA7DOAN[ND/10]);
    OUTPUT_D(MA7DOAN[ND%10]);
}
VOID MAIN()
{
    SET_TRIS_A(0X00);
    SET_TRIS_B(0X00);
    SET_TRIS_D(0X00);
    WHILE(TRUE)
    {
        IF(KBHIT())
        { KENH=GETC();
          //WHILE(KBHIT()==0) {}
          ND=GETC();

          HIENTHI();
        }
    }
}

```

Giảng viên soạn đáp án
Nguyễn Văn Hiệp