

Đại Học Công Nghiệp Thành Phố Hồ Chí Minh

Bài 4 : Hệ thống điều khiển Timer

Bộ môn: Điện Tử Máy Tính

GV: Lê Lý Quyên Quyên

Mail: lelyquyenquyen@iuh.edu.vn

Modul Timer

Timer 0, Timer 1, Timer 2 : chức năng Timer và Counter

OPTION register

I. Timer 0: (8bit)

$\overline{\text{RBPU}}$	INTEDG	T0CS	T0SE	PSA	PS2	PS1	PS0
--------------------------	--------	------	------	-----	-----	-----	-----

T0CS: chọn nguồn xung clock cho Timer 0 :

+**T0CS=1** : nguồn xung clock bên ngoài : **counter**

+**T0CS=0** : nguồn xung clock bên trong : **Timer**

• T0SE : chọn cạnh kích của nguồn xung clock bên ngoài

+**T0SE =1**: cạnh lên

+**T0SE =0**: cạnh xuống

Timer 0

I.Timer 0:

OPTION register

RBPV	INTEDG	T0CS	T0SE	PSA	PS2	PS1	PS0
------	--------	------	------	-----	-----	-----	-----

- PSA: chỉ định prescaler
 - + PSA=1 : dùng cho WDT
 - + **PSA=0** : dùng cho **Timer 0**

- **PS2, PS1, PS0:**

Các bit chọn tỉ lệ cho Prescaler

PS2	PS1	PS0	Tỉ lệ chia
0	0	0	1:2
0	0	1	1:4
0	1	0	1:8
0	1	1	1:16
1	0	0	1:32
1	0	1	1:64
1	1	0	1:128
1	1	1	1:256

Timer 0

1. Khởi động tính năng định thời cho Timer 0: 4 bước

B1. Tính TMR0 từ công thức:

$$t_{DELAY} = (256 - [TMR0]) \times 4 \times \frac{1}{f_{osc}} \times Pre$$

t_{DELAY} : Thời gian cần định thời (μs)

f_{osc} : Tần số dao động (MHz)

Pre: Giá trị hệ số Prescaler (Pre = 1; 2, 4, ... , 256)

[TMR0]: Giá trị cần ghi vào thanh ghi TMR0

$$0 \leq [TMR0] \leq 255$$

B2. TMR0IF=0 : xóa cờ ngắt

Timer 0

1. Khởi động tính năng định thời cho Timer 0:

B3. Chọn chế độ hoạt động của **Timer 0**:

- + **T0CS=0** : chế độ định thời gian
- + **GIE=PEIE=TMR0IE=0/1** : cấm ngắt or cho phép ngắt
- + **PSA=0** : gán prescaler cho Timer
- + **PS2=0/1; PS1=0/1; PS0=0/1**; chọn giá trị tỉ lệ của prescaler

B4. Xác định thời điểm Timer 0 bị tràn

- + **while (TMR0IF==0)**: kiểm tra cờ TMR0IF=1?
- + viết hàm ISR : **nếu dùng ngắt**

Timer 0

2. Khởi động tính năng đếm xung cho Timer 0: 4 bước

B1. Đặt giá trị ban đầu của bộ đếm -> **TMR0=?**

B2. TMR0IF=0 : xóa cờ ngắt

B3. Chọn chế độ hoạt động của **Timer 0**:

+ **T0CS=1** : chế độ đếm xung

+ **GIE=PEIE=TMR0IE=0/1** : cấm ngắt or cho phép ngắt

+ **PSA=0** : gán prescaler cho Timer

+ **PS2=0/1; PS1=0/1; PS0=0/1**; chọn giá trị tỉ lệ của prescaler

(counter k chọn prescaler -> **PSA=1; PS2=0; PS1=0; PS0=0**)

+ **T0SE=1/0** : chọn kích cạnh lên/cạnh xuống của ng.xung bên ngoài

Timer 0

2. Khởi động tính năng đếm xung cho Timer 0:

- B4. Đọc về và xử lý số xung đếm được trong thanh ghi TMR0
- Dựa vào cờ báo tràn TMR0IF để xử lý các trường hợp số xung đếm vượt quá 255.

Số xung thực tế = $(256 \times \text{số lần tràn của Timer 0}) + \text{TMR0}$

Timer 1

II. Timer 1: (16bit)



- **TMR1ON = 1/0** : cho phép/cấm Timer 1
- **TMR1CS** : Chọn nguồn xung clock
 - + **TMR1CS=1** : Bên ngoài (cạnh lên tại T1CKI): **Counter**
 - + **TMR1CS=0** : Bên trong ($F_{osc}/4$): **Timer**
- **T1SYNC** : Đồng bộ xung ck bên ngoài với xung ck bên trong
 - + **T1SYNC=1** : Không đồng bộ
 - + **T1SYNC=0** : Đồng bộ

Timer 1

Timer1 Control Register (T1CON)

T1GINV	TMR1GE	T1CKPS1	T1CKPS0	T1OSCEN	T1SYNC	TMR1CS	TMR1ON
--------	--------	---------	---------	---------	--------	--------	--------

- T1OSCEN : Cho phép bộ dao động LP
(bộ dao động phụ : 32.768Hz)
 - + **T1OSCEN = 1** : Chọn **T1OSC (RC1)**
 - + **T1OSCEN = 0** : Chọn **T1CKI (RC0)**
- **T1CKPS1, T1CKPS0** :
các bit chọn tỉ lệ cho prescaler

T1CKPS1	T1CKPS0	Tỉ lệ
1	1	1:8
1	0	1:4
0	1	1:2
0	0	1:1

Timer 1

1. Khởi động tính năng định thời cho Timer 1: 5 bước

B1. Tính TMR1 suy ra TMR1H, TMR1L từ công thức:

$$t_{DELAY} = (65536 - [TMR1]) \times 4 \times \frac{1}{f_{osc}} \times Pre$$

t_{DELAY} : Thời gian cần định thời (μs)

f_{osc} : Tần số dao động (MHz)

Pre: Giá trị hệ số Prescaler (Pre = 1, 2, 4, 8)

[TMR1]: Giá trị cần ghi vào thanh ghi TMR1

$$0 \leq [TMR1] \leq 65535$$

TMR1H=TMR1/256; TMR1L=TMR1%256;

B2. TMR1IF=0 : xóa cờ ngắt

Timer 1

1. Khởi động tính năng định thời cho Timer 1:

B3. Chọn chế độ hoạt động của **Timer 1**:

- Chế độ định thời gian:
 - + **TMR1CS=0** : Bộ dao động chính
 - + **TMR1CS=1** và **T1OSCEN=1** : Bộ dao động phụ
- **GIE=PEIE=TMR1IE=0/1** : cấm ngắt or cho phép ngắt
- **T1CKPS1=0/1; T1CKPS0=0/1** : chọn giá trị tỉ lệ của prescaler
- **T1SYNC=0/1** : đồng bộ/không đồng bộ xung ck

Timer 1

1. Khởi động tính năng định thời cho Timer 1:

B4. TMR1ON=1: cho phép Timer 1 hoạt động

B5. Xác định thời điểm Timer 1 bị tràn

+ **while (TMR1IF==0):** kiểm tra cờ TMR1IF=1?

+ viết hàm **ISR** : **nếu dùng ngắt**

Timer 1

2. Khởi động tính năng đếm xung cho Timer 1: 5 bước

B1. Đặt giá trị ban đầu của bộ đếm -> **TMR1H=?;TMR1L=?**

B2. **TMR1IF=0** : xóa cờ ngắt

B3. Chọn chế độ hoạt động của **Timer 1**:

- **TMR1CS=1** và **T1OSCEN=0** : chế độ đếm xung
- **GIE=PEIE=TMR1IE=0/1** : cấm ngắt or cho phép ngắt
- **T1CKPS1=0/1; T1CKPS0=0/1** : chọn giá trị tỉ lệ của prescaler
- **T1SYNC=0/1** : đồng bộ/không đồng bộ xung ck

Timer 1

2. Khởi động tính năng đếm xung cho Timer 1:

B4. TMR1ON=1: cho phép Timer 1 hoạt động

**B5. Đọc về và xử lý số xung đếm được trong thanh ghi
TMR1(TMR1H:TMR1L)**

Dựa vào cờ báo tràn TMR1IF để xử lý các trường hợp số xung đếm vượt quá 65535.

Số xung thực tế = $(65536 \times \text{số lần tràn của Timer 1}) + \text{TMR1H:TMR1L}$

Timer 2

III.Timer 2: (8bit)



- **TMR2ON=1/0** :cho phép/cấm Timer 2
- **T2CKPS1=0/1, T2CKPS0=0/1**
Các bit chọn tỉ lệ cho Prescaler

T2CKPS1	T2CKPS0	Tỉ lệ
0	0	1:1
0	1	1:4
1	X	1:16

Timer 2

III.Timer 2:

Timer2 Control Register (T2CON)

	TOUTPS3	TOUTPS2	TOUTPS1	TOUTPS0	TMR2ON	T2CKPS1	T2CKPS0
--	---------	---------	---------	---------	--------	---------	---------

- TOUTPS3=1/0; TOUTPS2=1/0;
TOUTPS1=1/0; TOUTPS0=1/0
Các bit chọn tỉ lệ cho postscaler

Timer 2 ít khi được dùng để định thời gian hoặc đếm sự kiện như Timer 0,1. Nó thường dùng cho modul CCP để tạo xung PWM.

Kỹ thuật vi xử lý

TOUTP S3	TOUTP S2	TOUTP S1	TOUTP S0	Tỉ lệ
0	0	0	0	1:1
0	0	0	1	1:2
0	0	1	0	1:3
0	0	1	1	1:4
0	1	0	0	1:5
0	1	0	1	1:6
0	1	1	0	1:7
0	1	1	1	1:8
1	0	0	0	1:9
1	0	0	1	1:10
1	0	1	0	1:11
1	0	1	1	1:12
1	1	0	0	1:13
1	1	0	1	1:14
1	1	1	0	1:15
1	1	1	1	1:16

Các bài tập ví dụ

VÍ DỤ 1 : Viết chương trình con tạo trễ tên **delay3s** có $t_{\text{delay}} = 3\text{s}$ dùng **Timer 0**, $f_{\text{osc}} = 4\text{MHz}$, chọn prescaler = 4.

Tính giá trị TMR0 theo công thức, chọn $t_{\text{delay}} = 100\mu\text{s}$, suy ra:

$$t_{\text{DELAY}} = (256 - [\text{TMR0}]) \times 4 \times \frac{1}{f_{\text{osc}}} \times \text{Pre}$$

TMR0=231

t_{DELAY} : Thời gian cần định thời (μs)

f_{osc} : Tần số dao động (MHz)

Pre: Giá trị hệ số Prescaler (Pre = 1; 2, 4, ... , 256)

[TMR0]: Giá trị cần ghi vào thanh ghi TMR0

$$0 \leq [\text{TMR0}] \leq 255$$

```
void delay100us(unsigned int n)
```

```
{ while(n--)
```

```
{
```

```
    TMR0 = 231;
```

```
    TMR0IF = 0; //xóa cờ ngắt
```

```
    T0CS = 0; //chế độ định thời
```

```
    GIE=PEIE =TMR0IE = 0;//cấm ngắt
```

```
    PSA = 0; PS2 = 0; PS1 = 0; PS0 = 1; //prescaler=4
```

```
    while(TMR0IF==0); } }//kiểm tra cờ TMR0IF=1?
```

```
void delay3s(void)
```

```
{
```

```
    delay100us(30000);
```

```
}
```

Các bài tập ví dụ

VÍ DỤ 2 : Viết chương trình con tạo trễ tên **delay3s** có $t_{\text{delay}} = 3\text{s}$ dùng **Timer 1**, $f_{\text{osc}} = 4\text{MHz}$, chọn prescaler = 4.

Tính giá trị TMR1 theo công thức, chọn $t_{\text{delay}} = 100\mu\text{s}$, prescaler = 4
TMR1=65511->TMR1H=65511/256 = 255;

TMR1L=65511%256 = 231;

$$t_{\text{DELAY}} = (65536 - [\text{TMR1}]) \times 4 \times \frac{1}{f_{\text{osc}}} \times \text{Pre}$$

t_{DELAY} : Thời gian cần định thời (μs)

f_{osc} : Tần số dao động (MHz)

Pre: Giá trị hệ số Prescaler (Pre = 1, 2, 4, 8)

[TMR1]: Giá trị cần ghi vào thanh ghi TMR1

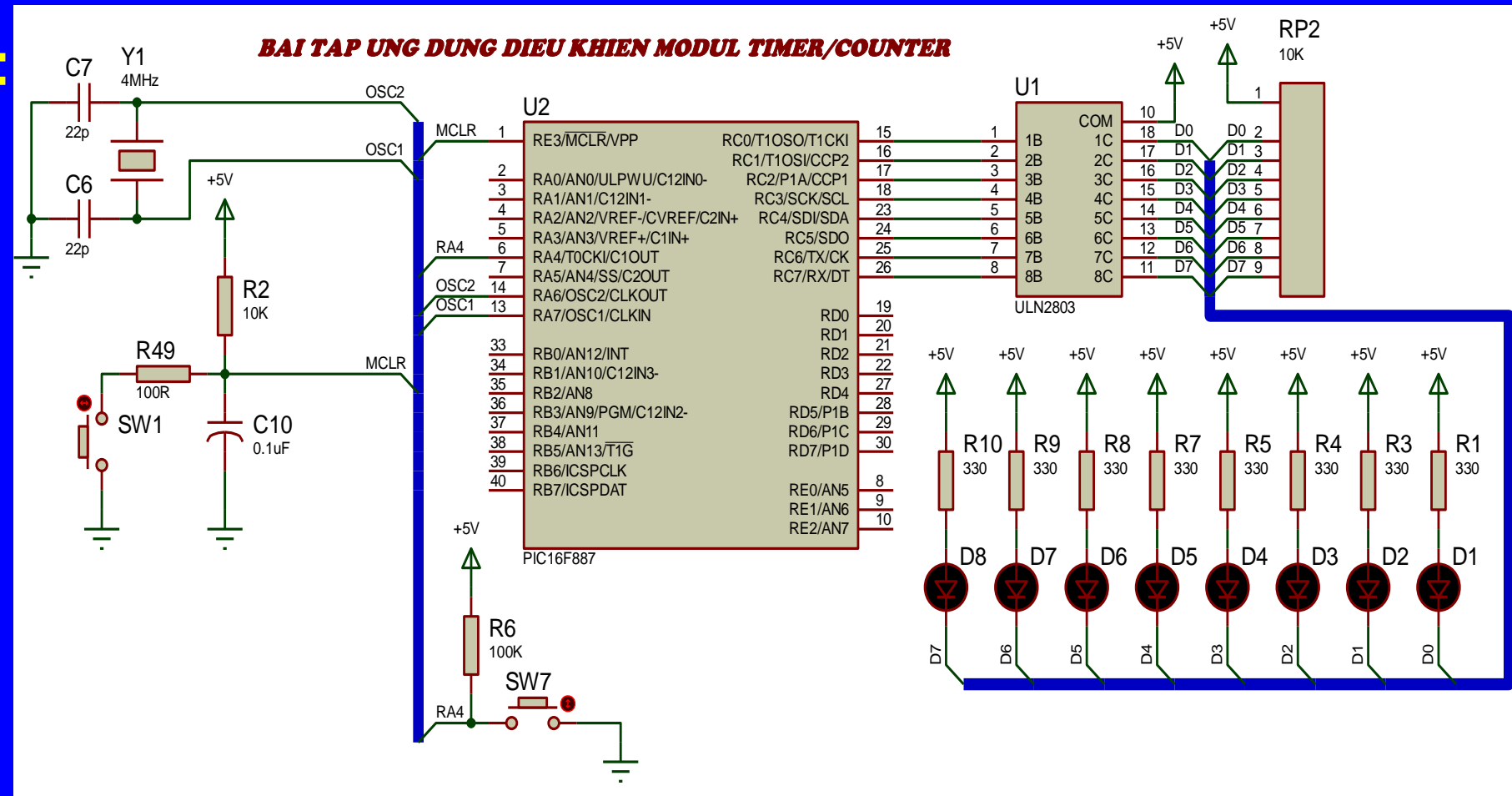
$$0 \leq [\text{TMR1}] \leq 65535$$

<pre> void delay100us(unsigned int n) { while(n--) { TMR1H= 255; TMR1L= 231; TMR1IF = 0;//xóa cờ ngắt TMR1CS = 0; //chế độ định thời GIE=PEIE =TMR1IE = 0;// cấm ngắt T1CKPS1 = 1; T1CKPS0 = 0;//prescaler= 4 TMR1ON = 1; //cho phép Timer 1 hoạt động T1SYNC=1; //không đồng bộ xung ck while(TMR1IF==0); } } //kiểm tra cờ TMR1IF=1? </pre>	<pre> void delay3s(void) { delay100us(30000); } </pre>
---	--

VÍ DỤ 3

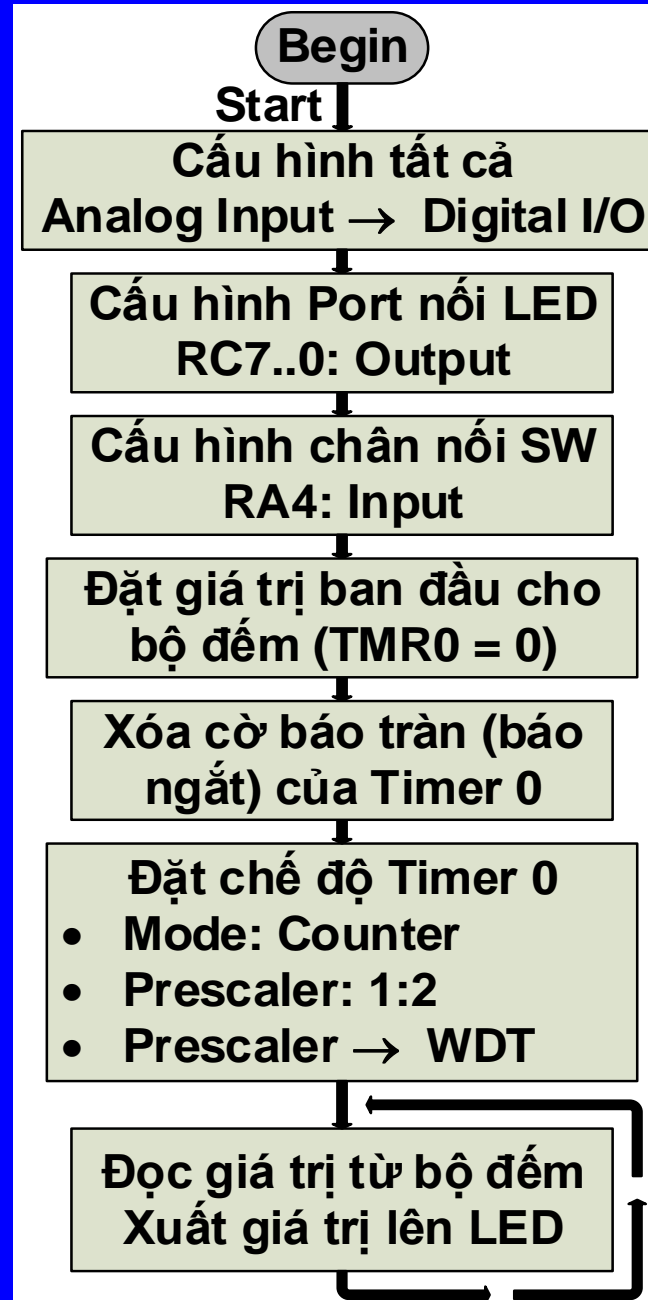
- Dựa vào sơ đồ, viết chương trình điều khiển đếm số lần nhấn–nhả nút SW7 (0 → 255) được nối với ngõ vào của Counter 0 (T0CKI) và hiển thị giá trị này trên 8 LED đơn. Sử dụng modul Counter 0 để đếm.

- Sơ đồ nguyên lý:



VÍ DỤ 3

- Giải thuật:



VÍ DỤ 3

- **Cấu hình (Hi-Tech C):**

```
__CONFIG(FOSC_HS & WDTE_OFF & PWRTE_ON &  
MCLRE_ON & CP_OFF & CPD_OFF & BOREN_OFF &  
IESO_OFF & FCMEN_OFF & LVP_OFF & DEBUG_ON);
```

```
#define _XTAL_FREQ 4000000
```

VÍ DỤ 3

- Chương trình (Hi-Tech C):

```
void main (void)
{
    ANSEL = 0;
    ANSELH = 0;

    TRISC = 0;
    TRISA4 = 1;

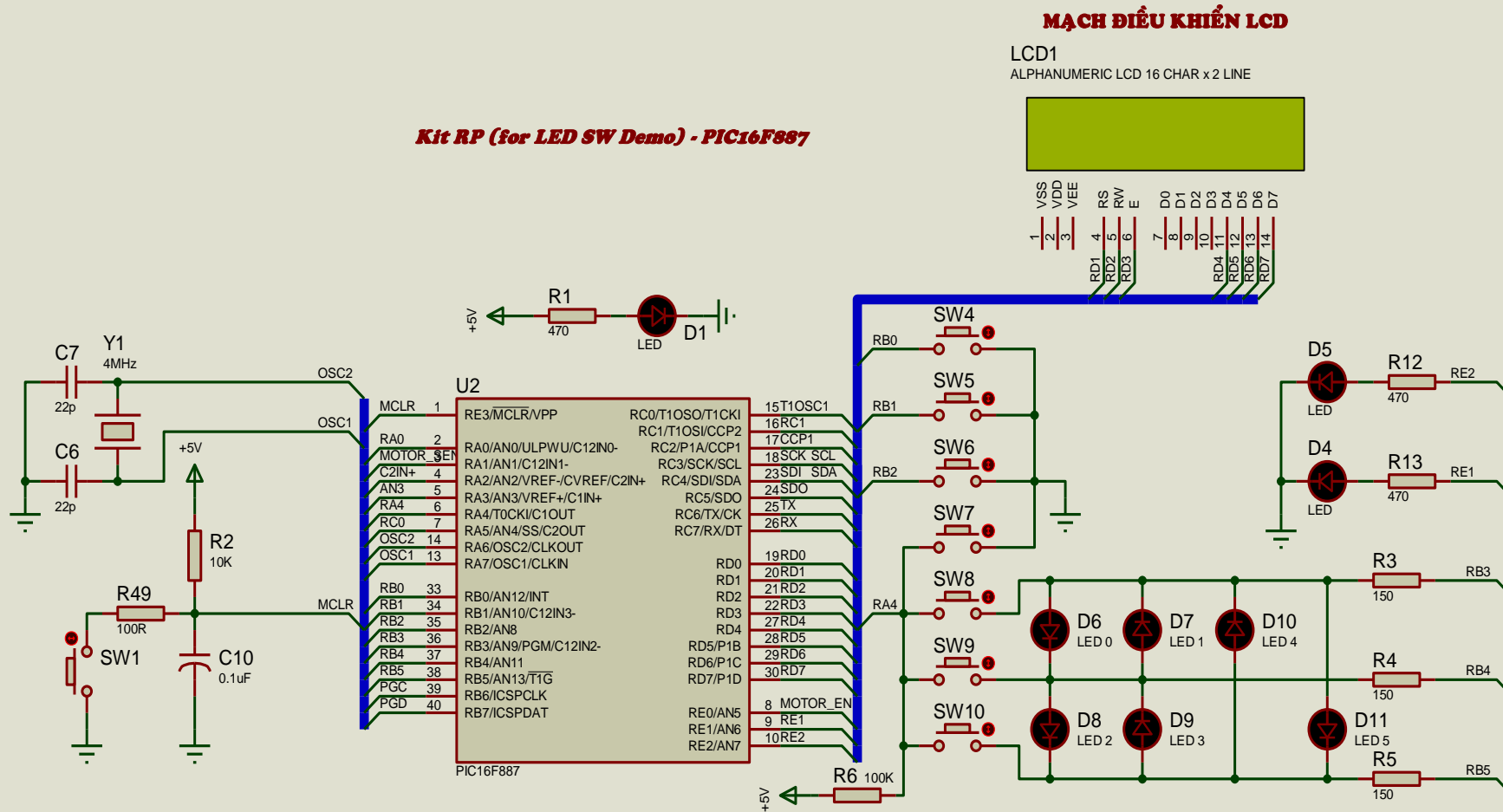
    TMR0 = 0;

    TMR0IF = 0;
```

```
T0CS = 1;
T0SE = 0;
PSA = 1;
PS2 = 0;
PS1 = 0;
PS0 = 0;

while(1)
{
    PORTC = TMR0;
}
}
```


Cho sơ đồ mạch kit RP như hình vẽ



BÀI TẬP ỨNG DỤNG

Dựa vào sơ đồ mạch kit RP

- **Bài tập 1:** Viết chương trình sử dụng Modul Counter 0 để thực hiện việc tăng biến đếm khi nhấn_nhả SW7. Nếu giá trị biến đếm bằng 25 thì reset giá trị biến đếm về 00 và hiển thị giá trị đếm lên LCD như sau:

Tên sinh viên Counter 0
xx

xx: giá trị biến đếm (00÷25)

line 1_LCD: tên T.Việt không dấu hiển thị sẵn trên LCD.

BÀI TẬP ỨNG DỤNG

Dựa vào sơ đồ mạch kit RP

- **Bài tập 2:** Viết chương trình sử dụng Modul Counter 1 để thực hiện việc tăng biến đếm khi nhấn SW(nối với T1CKI:SV tự bổ sung) Nếu giá trị biến đếm bằng 30 thì reset giá trị biến đếm về 00 và hiển thị giá trị đếm lên LCD như sau:

Tên sinh viên Counter 1
xx

xx: giá trị biến đếm (00÷30)

line 1_LCD: tên T.Việt có dấu hiển thị sẵn trên LCD.

BÀI TẬP ỨNG DỤNG

Dựa vào sơ đồ mạch kit RP

➤ **Bài tập 3:** Viết chương trình hiển thị lên LCD nội dung như sau:

Bài tập Timer0
Họ và tên sinh viên

Yêu cầu :- Nội dung LCD chớp tắt liên tục 3 lần.

- Nội dung LCD: dịch chuyển từ trái sang phải 2 lần**
- Nội dung LCD :dịch chuyển từ phải sang trái 2 lần**
- Sử dụng Timer0 để viết chương trình tạo thời gian trễ, không sử dụng lệnh `__delay_ms()`;**

BÀI TẬP ỨNG DỤNG

Dựa vào sơ đồ mạch kit RP

- **Bài tập 4:** Viết chương trình hiển thị lên LCD nội dung như sau:

Họ và tên sinh viên
Bài tập Timer1

Yêu cầu :- Nội dung LCD chớp tắt liên tục 3 lần.

- Nội dung LCD: dịch chuyển từ trái sang phải 2 lần
- Nội dung LCD :dịch chuyển từ phải sang trái 2 lần
- Sử dụng Timer1 để viết chương trình tạo thời gian trễ, không sử dụng lệnh `__delay_ms()`;

YÊU CẦU BUỔI HỌC

➤ Mỗi sinh viên tự viết code, chạy mô phỏng trên phần mềm Proteus đúng yêu cầu bài toán:

1. Bài tập 1 : 3đ

2. Bài tập 2 : 3đ

3. Bài tập 3 : 2đ

4. Bài tập 4 : 2đ

➤ Thời gian làm bài :

➤ Thời gian chấm bài :