

Đại học Quốc Gia TP HCM
Trường Đại học Khoa Học Tự Nhiên
Khoa Vật Lý – Vật Lý Kỹ Thuật
Bộ môn Vật Lý Tin Học



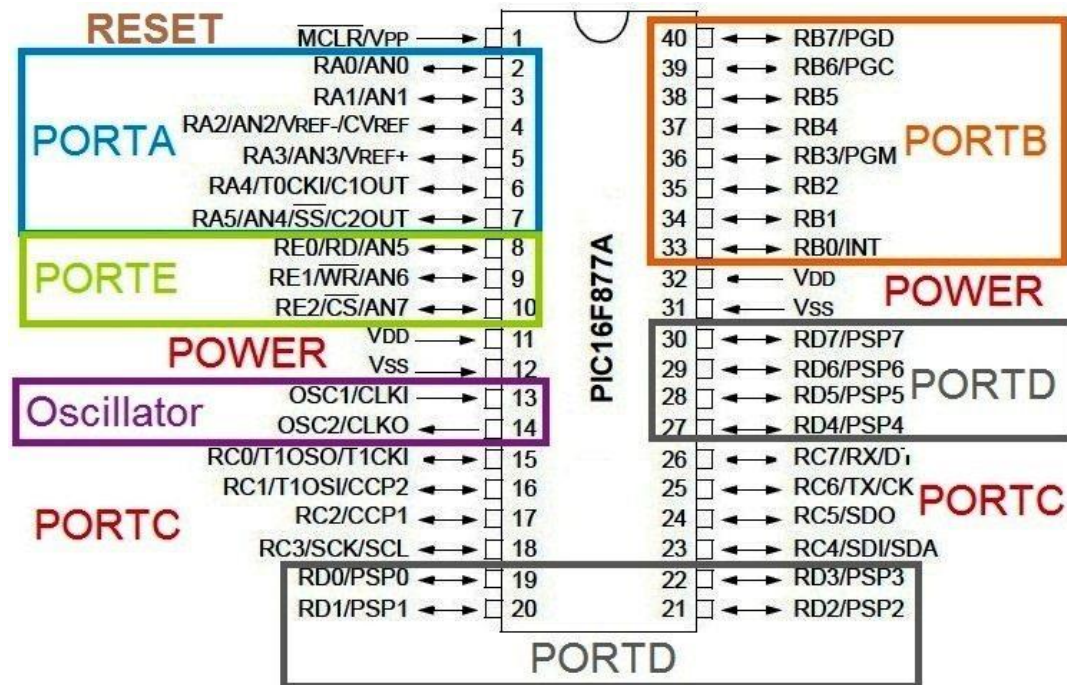
THỰC HÀNH VI ĐIỀU KHIỂN (PHY10605)

CBHD: Võ Hoàng Thủy Tiên
vhttien@hcmus.edu.vn
0937649914

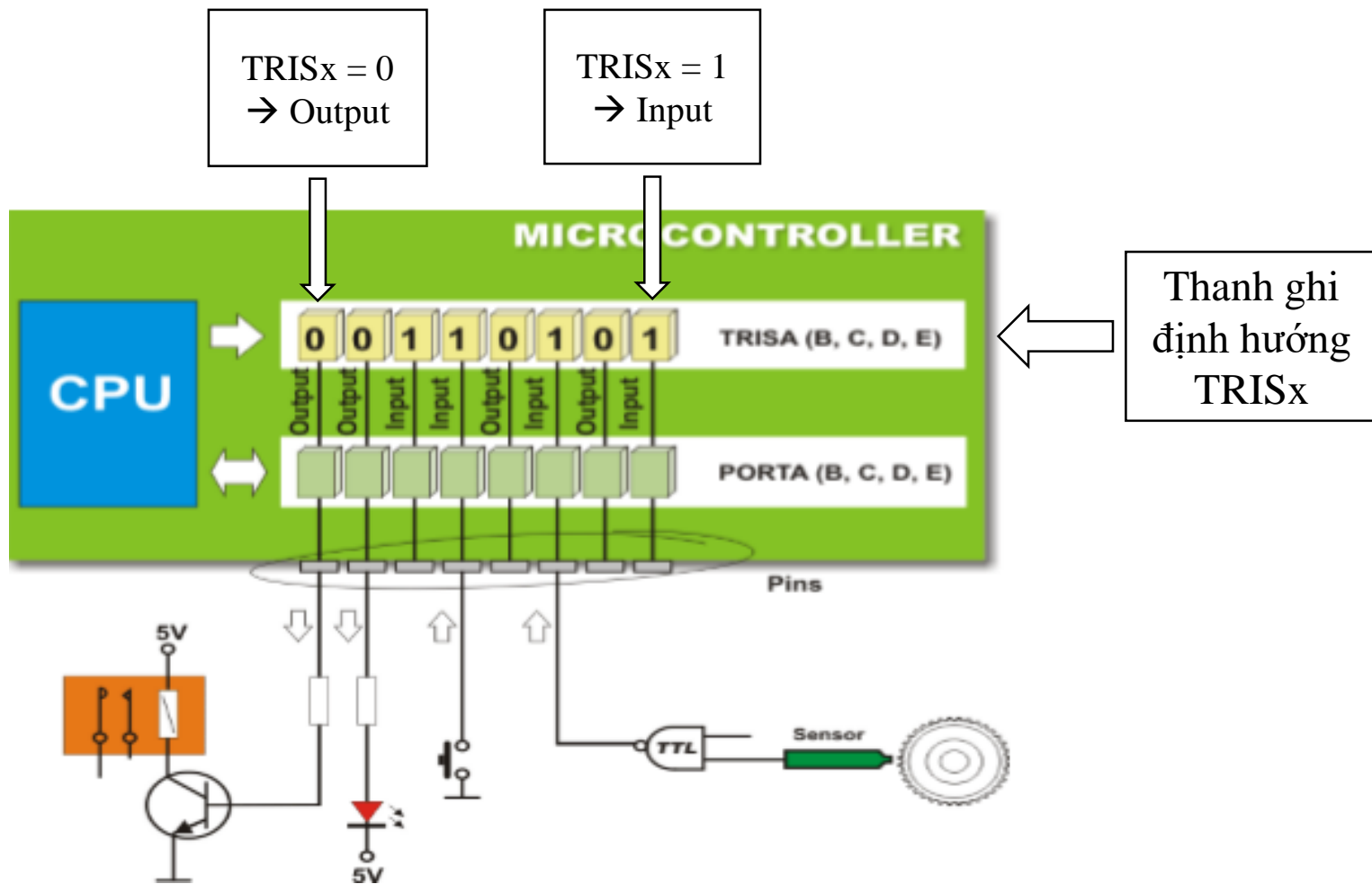
Huỳnh Quốc Việt
hyqviet@hcmus.edu.vn
0349043204

3.1. Cổng Nhập Xuất (I/O Port)

- Cổng xuất nhập (I/O Port) là phương tiện của vi điều khiển dùng để tương tác với các thiết bị bên ngoài
- Vi điều khiển PIC16F877 có 5 cổng nhập xuất được chia thành các khối PortA, PortB, PortC, PortD và PortE.
- Mỗi Port sẽ có khả năng cấp và nhận dòng là 25mA

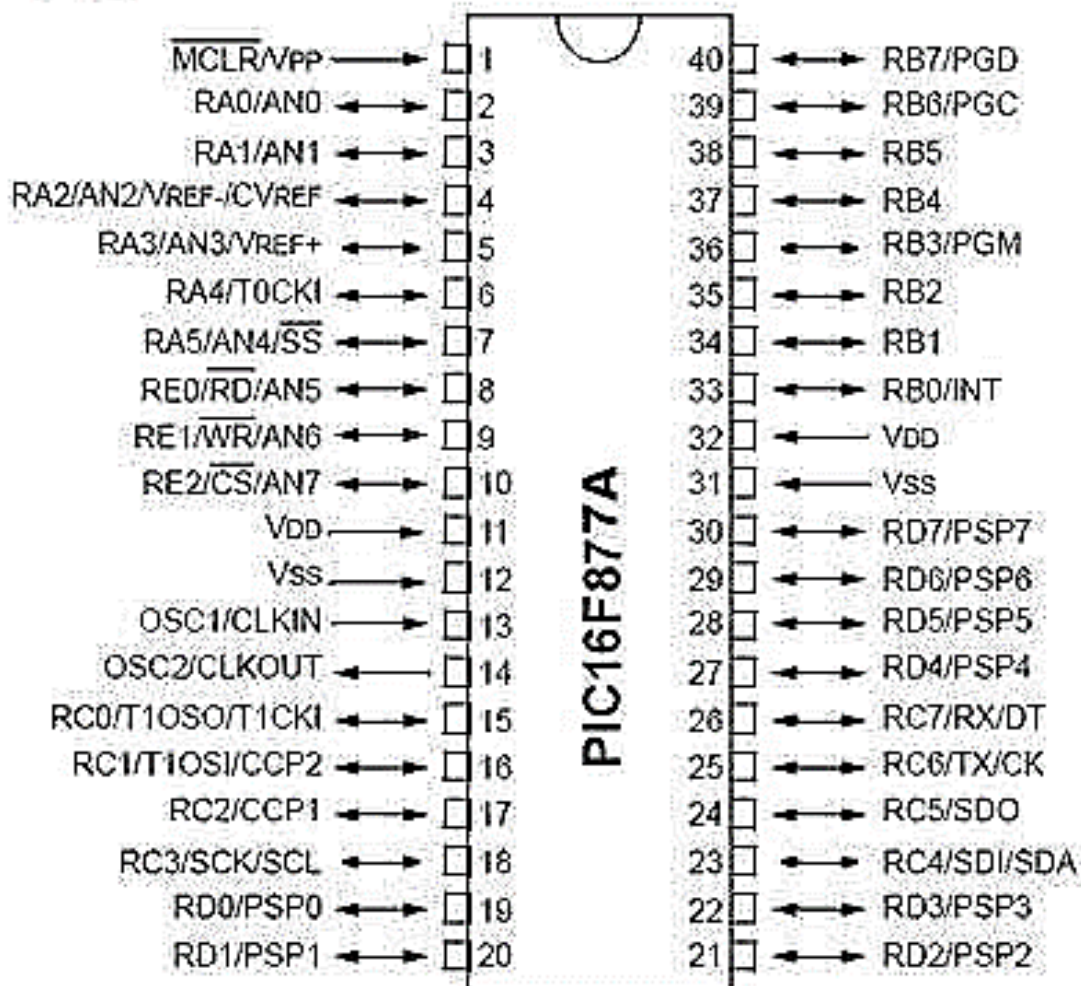


3.1. Cổng Nhập Xuất



3.1. Cổng Nhập Xuất

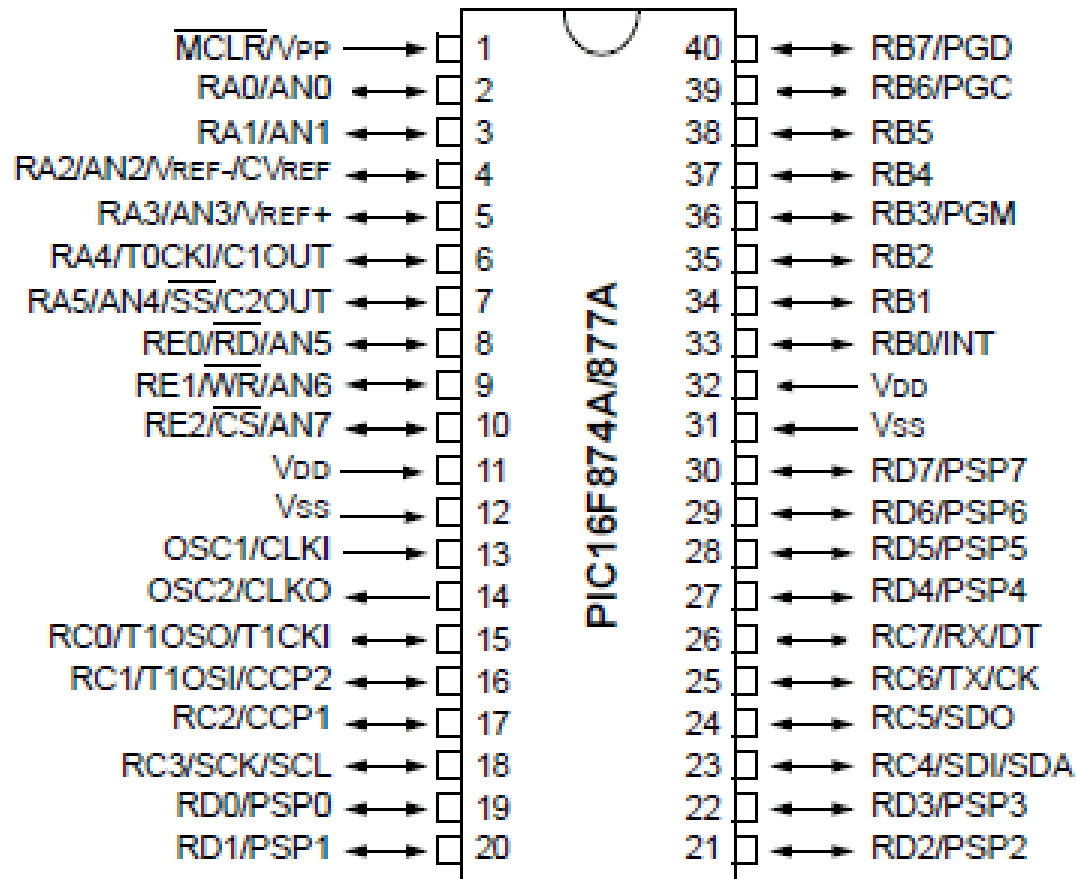
PDIP



3.2. Khảo sát chân PIC 16F877A

Chân	Chức năng
/MCLR	Hoạt động Reset ở mức thấp
RA _x /RB _x /RC _x /RD _x /RE _x	Ngõ vào xuất nhập số
AN _x	Ngõ vào kênh thứ x của khối ADC
V _{REF-} và V _{REF+}	<ul style="list-style-type: none"> - V_{REF-}: Ngõ vào điện áp tham chiếu thấp - V_{REF+}: Ngõ vào điện áp tham chiếu cao
TXCKIx	Ngõ vào xung clock bên ngoài của Timerx
VDD/VSS	Chân nối nguồn/nối đất của PIC
OSC1/OSC2	<ul style="list-style-type: none"> - OSC1: Ngõ vào dao động thạch anh - OSC2: Ngõ ra dao động thạch anh
CLKI/CLKO	<ul style="list-style-type: none"> - CLKI: Ngõ vào xung dao động bên ngoài - CLKO: Ngõ ra xung dao động
RX/TX	<ul style="list-style-type: none"> - RX: Nhận bắt đồng bộ USART - TX: Truyền bắt đồng bộ USART
INT	Ngắt ngoài

3.2. Khảo sát chân PIC 16F877A



3.2. Khảo sát chân PIC 16F877A

Chân	Tên	Chức năng
1	/MCLR VPP	– /MCLR: Hoạt động Reset ở mức thấp – VPP : ngõ vào áp lập trình
2	RA0/AN0	– RA0 : xuất/nhập số – AN0 : ngõ vào tương tự
3	RA1/AN1	– RA1 : xuất/nhập số – AN1 : ngõ vào tương tự
4	RA2/AN2/V_{REF-}/CV_{REF}	– RA2 : xuất/nhập số – AN2 : ngõ vào tương tự – VREF - : ngõ vào điện áp chuẩn (thấp) của bộ A/D
5	RA3/AN3/V_{REF+}	– RA3 : xuất/nhập số – AN3 : ngõ vào tương tự – VREF+ : ngõ vào điện áp chuẩn (cao) của bộ A/D

3.2. Khảo sát chân PIC 16F877A

Chân	Tên	Chức năng
6	RA4/TOCKI/C1OUT	<ul style="list-style-type: none"> – RA4 : xuất/nhập số – TOCKI : ngõ vào xung clock bên ngoài cho timer0 – C1OUT : Ngõ ra bộ so sánh 1
7	RA5/AN4//SS /C2OUT	<ul style="list-style-type: none"> – RA5 : xuất/nhập số – AN4 : ngõ vào tương tự 4 – SS : ngõ vào chọn lựa SPI phụ – C2OUT : ngõ ra bộ so sánh 2
8	RE0//RD/AN5	<ul style="list-style-type: none"> – RE0 : xuất nhập số – RD : điều khiển việc đọc ở port nhánh song song – AN5 : ngõ vào tương tự
9	RE1//WR/AN6	<ul style="list-style-type: none"> – RE1 : xuất/nhập số – WR : điều khiển việc ghi ở port nhánh song song – AN6 : ngõ vào tương tự
10	RE2//CS/AN7	<ul style="list-style-type: none"> – RE2 : xuất/nhập số – CS : Chip lựa chọn sự điều khiển ở port nhánh song song – AN7 : ngõ vào tương tự

3.2. Khảo sát chân PIC 16F877A

Chân	Tên	Chức năng
11	V_{DD}	Chân nguồn của PIC.
12	V_{SS}	Chân nối đất
13	OSC1/CLKI	<ul style="list-style-type: none">– OSC1 : ngõ vào dao động thạch anh hoặc xung clock bên ngoài.– CLKI : ngõ vào nguồn xung bên ngoài. Luôn được kết hợp với chức năng OSC1.
14	OSC2/CLKO	<ul style="list-style-type: none">– OSC2 : Ngõ ra dao động thạch anh. Kết nối đến thạch anh hoặc bộ cộng hưởng.– CLKO : ở chế độ RC, ngõ ra của OSC2, bằng tần số của OSC1 và chỉ ra tốc độ của chu kỳ lệnh.

3.2. Khảo sát chân PIC 16F877A

Chân	Tên	Chức năng
15	RC0/T1OCO/T1CKI	<ul style="list-style-type: none">– RC0 : xuất/nhập số– T1OCO : ngõ vào bộ dao động Timer 1– T1CKI : ngõ vào xung clock bên ngoài Timer 1
16	RC1/T1OSI/CCP2	<ul style="list-style-type: none">– RC1 : xuất/nhập số– T1OSI : ngõ vào bộ dao động Timer 1– CCP2 : ngõ vào Capture 2, ngõ ra compare 2, ngõ ra PWM2
17	RC2/CCP1	<ul style="list-style-type: none">– RC2 : xuất/nhập số– CCP1 : ngõ vào Capture 1, ngõ ra compare 1, ngõ ra PWM1

3.2. Khảo sát chân PIC 16F877A

Chân	Tên	Chức năng
18	RC3/SCK/SCL	<ul style="list-style-type: none">– RC3 : xuất/nhập số– SCK : ngõ vào xung clock nối tiếp đồng bộ/ngõ ra của chế độ SPI– SCL : ngõ vào xung clock nối tiếp đồng bộ/ ngõ ra của chế độ I2C
19	RD0/PSP0	<ul style="list-style-type: none">– RD0 : xuất/nhập số– PSP0 : dữ liệu port nhánh song song
20	RD1/PSP1	<ul style="list-style-type: none">– RD1 : xuất/nhập số– PSP1 : dữ liệu port nhánh song song
21	RD2/PSP2	<ul style="list-style-type: none">– RD2 : xuất/nhập số– PSP2 : dữ liệu port nhánh song song

3.2. Khảo sát chân PIC 16F877A

Chân	Tên	Chức năng
22	RD3/PSP3	– RD3: xuất/nhập số – PSP3 : dữ liệu port nhánh song song
23	RC4/SDI/SDA	– RC4 : xuất/nhập số – SDI : dữ liệu vào SPI – SDA : xuất/nhập dữ liệu vào I2C
24	RC5/SDO	– RC5 : xuất/nhập số – SDO : dữ liệu ra SPI
25	RC6/TX/CK	– RC6 : xuất/nhập số – TX : truyền bất đồng bộ USART – CK : xung đồng bộ USART
26	RC7/RX/DT	– RC7 : xuất/nhập số – RX : nhận bất đồng bộ USART – DT : dữ liệu đồng bộ USART

3.2. Khảo sát chân PIC 16F877A

Chân	Tên	Chức năng
27	RD4/PSP	– RD4: xuất/nhập số – PSP4 : dữ liệu port nhánh song song
28	RD5/PSP5	– RD5: xuất/nhập số – PSP5 : dữ liệu port nhánh song song
29	RD6/PSP6	– RD6: xuất/nhập số – PSP6 : dữ liệu port nhánh song song
30	RD7/PSP7	– RD7: xuất/nhập số – PSP7 : dữ liệu port nhánh song song
31	V_{SS}	Chân nối đất
32	V_{DD}	Chân nguồn của PIC.

3.2. Khảo sát chân PIC 16F877A

Chân	Tên	Chức năng
33	RB0/INT	– RB0 : xuất/nhập số – INT : ngắt ngoài
34	RB1	xuất/nhập số
35	RB2	xuất/nhập số
36	RB3	– RB3 : xuất/nhập số – PGM: Chân cho phép lập trình điện áp thấp ICPS
37	RB4	– xuất/nhập số

3.2. Khảo sát chân PIC 16F877A

Chân	Tên	Chức năng
38	RB5	– xuất/nhập số
39	RB6/PGC	– RB6 : xuất/nhập số – PGC : mạch vi sai và xung clock lập trình ICSP
40	RB7/PGD	– RB7 : xuất/nhập số – PGD : mạch vi sai và dữ liệu lập trình ICSP

3.3. Một số lệnh truy xuất Port

Câu lệnh	Chức năng
set_tris_x(vaule)	Định cấu hình cho portx
output_low (pin)	Cho 1 chân của port xuống 0
output_high (pin)	Cho 1 chân của port lên 1
output_bit(pin,vaule)	Xuất giá trị vaule ra 1 chân của port
output_x(vaule)	Xuất dữ liệu 8bit ra portx
input_x()	Đọc giá trị tất cả các chân của port gán cho biến
input(pin)	Đọc giá trị 1 chân của port gán cho một biến

3.3. Một số lệnh truy xuất Port

The screenshot displays a microcontroller IDE interface. On the left, the 'Files' pane shows a project structure with folders 'main', 'Source', 'Output', and 'Documentation'. Inside 'Source', there is a sub-folder 'main' containing 'main.c', 'main.h', and '16F877A.h'. An arrow points from the text box 'Thư viện 16F877A' to the '16F877A.h' file. The main editor window shows the content of '16F877A.h', which includes various I/O function prototypes. An arrow points from the text box 'Một số hàm khai báo có trong thư viện' to the function prototypes in the code. The status bar at the bottom indicates the time is 10:17, the mode is 'Insert', and the project is 'Pjt: main'.

Files

main

Source

main

main.c

main.h

16F877A.h

Output

Documentation

main Notes.txt

Thư viện 16F877A

16F877A.h

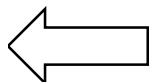
```
33 // PORT_x_PULLUPS(), INPUT(),
34 // OUTPUT_LOW(), OUTPUT_HIGH(),
35 // OUTPUT_FLOAT(), OUTPUT_BIT()
36 // Discrete I/O Prototypes:
37 _bif void set_tris_a(int8 value);
38 _bif void set_tris_b(int8 value);
39 _bif void set_tris_c(int8 value);
40 _bif void set_tris_d(int8 value);
41 _bif void set_tris_e(int8 value);
42 _bif int8 get_tris_a(void);
43 _bif int8 get_tris_b(void);
44 _bif int8 get_tris_c(void);
45 _bif int8 get_tris_d(void);
46 _bif int8 get_tris_e(void);
47 _bif void output_a(int8 value);
48 _bif void output_b(int8 value);
49 _bif void output_c(int8 value);
50 _bif void output_d(int8 value);
51 _bif void output_e(int8 value);
52 _bif int8 input_a(void);
53 _bif int8 input_b(void);
54 _bif int8 input_c(void);
55 _bif int8 input_d(void);
56 _bif int8 input_e(void);
57 _bif int1 input_change_a(void);
```

Một số hàm khai báo có trong thư viện

10:17 Insert Pjt: main

3.3. Một số lệnh truy xuất Port

```
#define PIN_A0 40  
#define PIN_A1 41  
#define PIN_A2 42  
#define PIN_A3 43  
#define PIN_A4 44  
#define PIN_A5 45
```



Khai báo chân sử
dụng vi điều khiển

```
#define PIN_B0 48  
#define PIN_B1 49  
#define PIN_B2 50  
#define PIN_B3 51  
#define PIN_B4 52  
#define PIN_B5 53  
#define PIN_B6 54  
#define PIN_B7 55
```

```
#define PIN_C0 56  
#define PIN_C1 57  
#define PIN_C2 58  
#define PIN_C3 59  
#define PIN_C4 60  
#define PIN_C5 61  
#define PIN_C6 62  
#define PIN_C7 63
```

```
#define PIN_D0 64  
#define PIN_D1 65  
#define PIN_D2 66  
#define PIN_D3 67  
#define PIN_D4 68  
#define PIN_D5 69  
#define PIN_D6 70  
#define PIN_D7 71
```

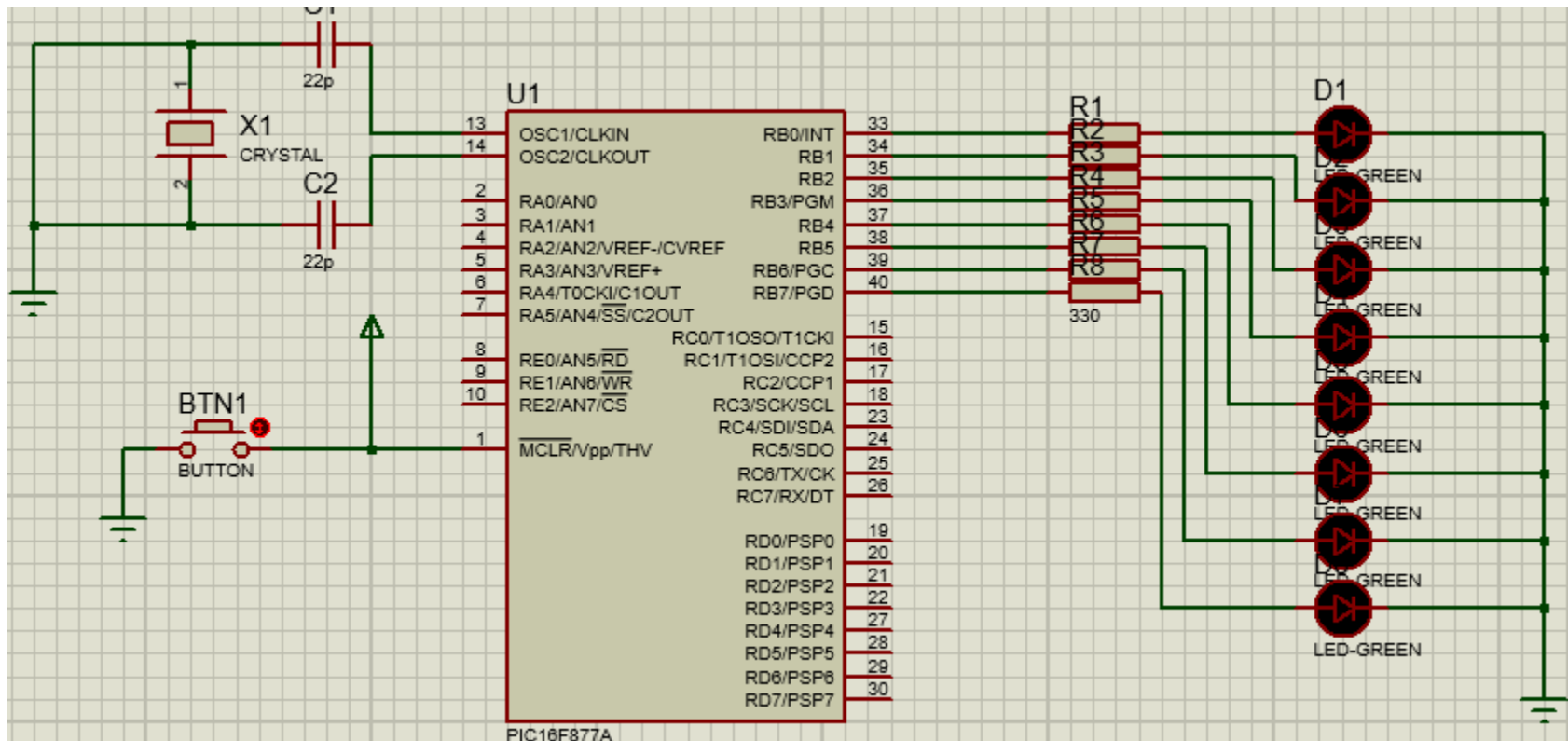
Ví dụ:

- output_low(PIN_D0);
- output_high(PIN_A3);
- output_bit(PIN_D2,1);
- Vaule=input(PIN_B1);

3.4. Thực hành

Bài 1.1: Dùng vi điều khiển 16F877A điều khiển 8 led đơn sáng tắt theo thứ tự từ trên xuống.

Bài 1.2: Dùng vi điều khiển 16F877A điều khiển 8 led đơn sáng tắt theo thứ tự chẵn/lẻ từ trên xuống.



3.4. Thực hành

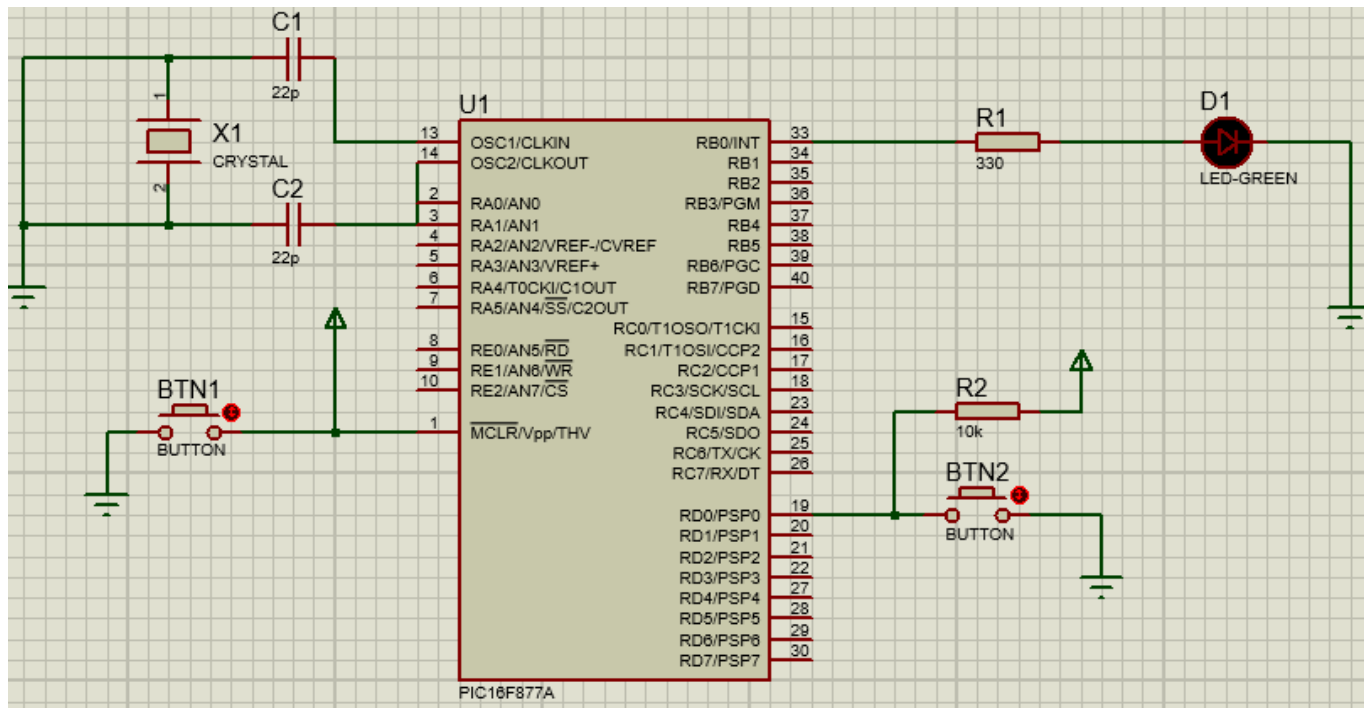
```
#include <...>
void main()
{
    Đặt port b là output
    Tắt tất cả các đèn
    while(TRUE)
    {
        Bật tắt led theo thứ tự
    }
}
```

```
#include <...>
Khai báo mảng giá trị bật tắt từng đèn
void main()
{
    Đặt port b là output
    Tắt tất cả các đèn
    while(TRUE)
    {
        for (i=0;i<x;i++)
        {
            Bật tắt led theo thứ tự
        }
    }
}
```

3.4. Thực hành

Bài 2: Sử dụng nút nhấn để điều khiển đèn.

- A) Khi nút nhấn được nhấn thì đèn sẽ sáng, khi không nhấn thì đèn sẽ tắt
- B) Khi nút nhấn được nhấn thì đèn sẽ thay đổi trạng thái (từ sáng sang tắt và ngược lại)



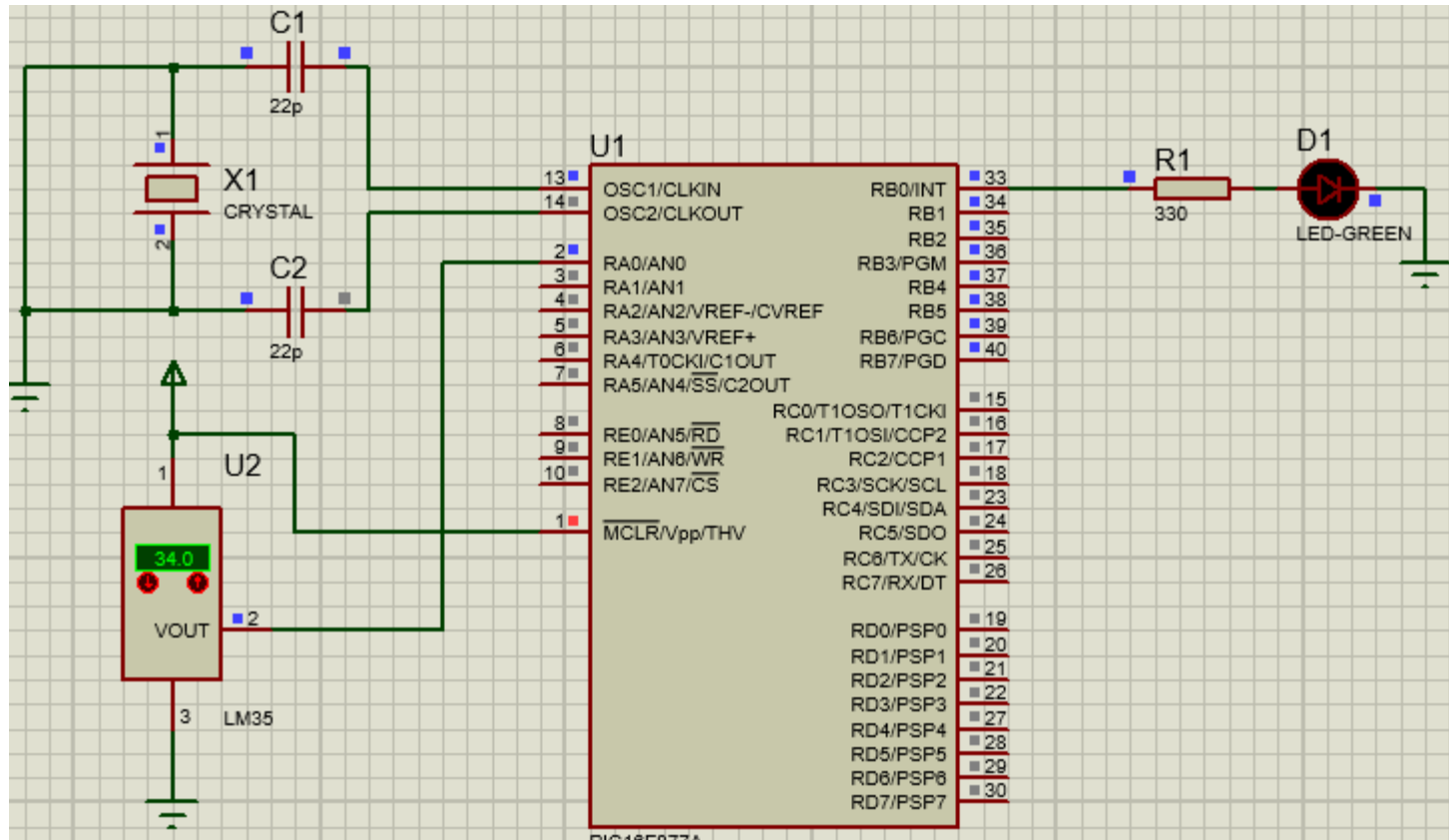
3.4. Thực hành

```
#include <...>
Khai báo các biến sử dụng
void main()
{
    Đặt port x là output
    Tắt tất cả các đèn
```

```
while(TRUE)
{
    Đọc giá trị nút nhấn
    if ()
    {
        Điều khiển LED
    }
    else
    {
        Điều khiển LED
    }
}
```

3.4. Thực hành

Bài 3: Đọc giá trị điện áp nếu điện áp quá cao đèn sẽ bật và ngược lại (sử dụng bộ chuyển đổi ADC)



3.4. Thực hành

Nếu output voltage = 10mV \rightarrow Nhiệt độ = 1°C

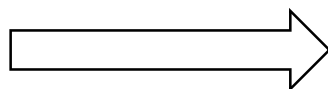
Nếu output voltage = 100mV \rightarrow Nhiệt độ = 10°C

Nếu output voltage = 370mV \rightarrow Nhiệt độ = 37°C

\rightarrow Nhiệt độ = $V_o * 100$

ADC 10 bit:

5V	_____	1023
4.995V	_____	1022
4.990V	_____	1021
4.985V	_____	1020
	_____	1019
.	.	.
	_____	1
	_____	0



$$\text{Nhiệt độ} = V_o * 100 * 5 / 1023$$

3.4. Thực hành

```
#include <...>
```

Khai báo các biến sử dụng

```
void main()
```

```
{
```

Đặt port x là output

Tắt tất cả các đèn

%khởi tạo cho ADC

```
setup_adc(ADC_CLOCK_INTERNAL);
```

```
setup_adc_ports(AN0);
```

```
set_adc_channel(0);
```

```
while(TRUE)
```

```
{
```

Đọc giá trị ADC (read_adc())

Tính toán theo công thức

```
if (nhiệt độ >...)
```

```
{
```

Điều khiển LED

```
}
```

```
else
```

```
{
```

Điều khiển LED

```
}
```

```
}
```