

**TRƯỜNG ĐẠI HỌC GIAO THÔNG VẬN TẢI  
BỘ MÔN KỸ THUẬT ĐIỆN TỬ**

**THỰC HÀNH HỆ THỐNG NHÚNG**

**Hà nội -2016**

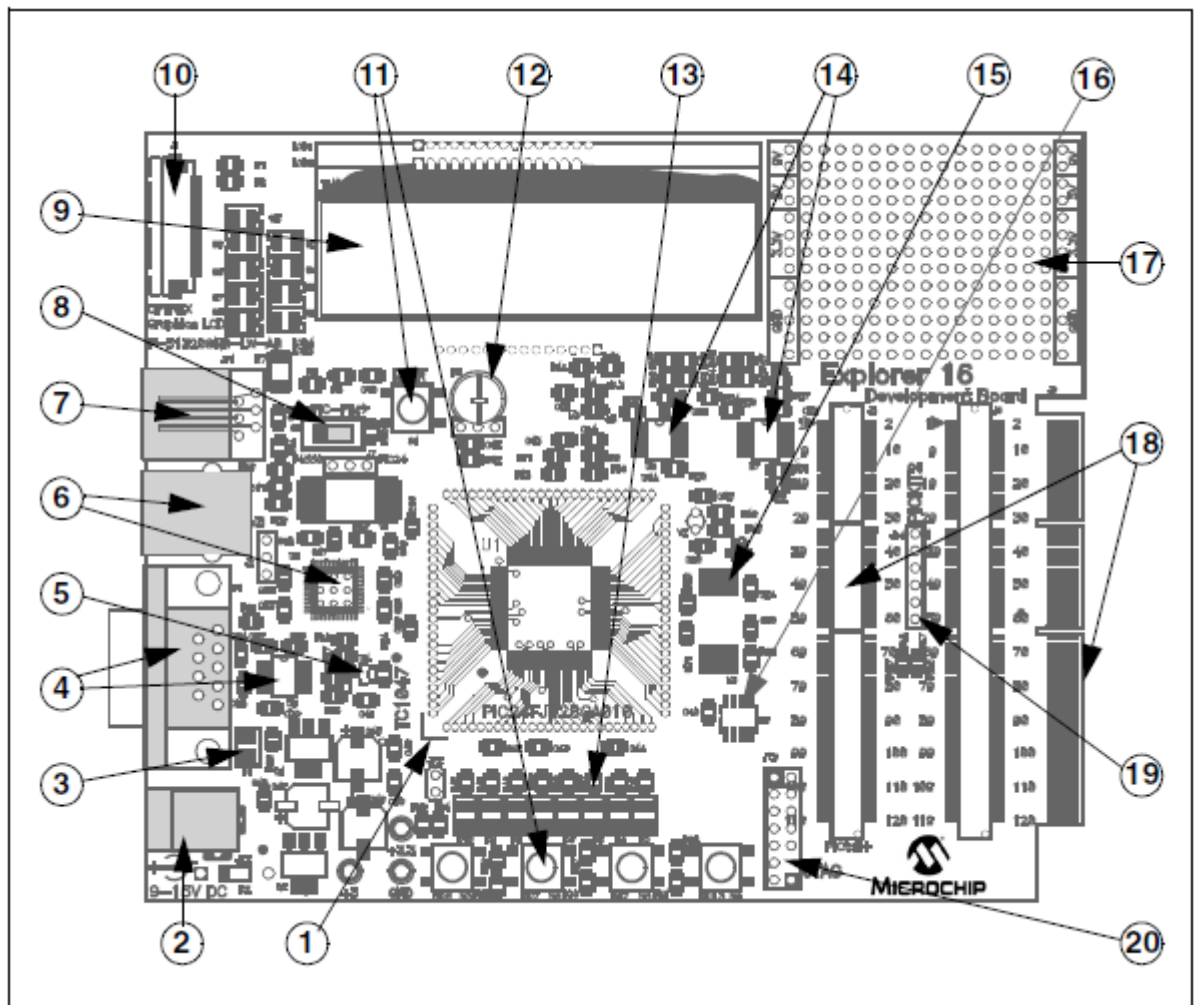
# CHƯƠNG 1. CẤU TRÚC PHẦN CỨNG HỆ VI XỬ LÝ THỰC HÀNH.

## I. Tổng quan về mạch thực hành.

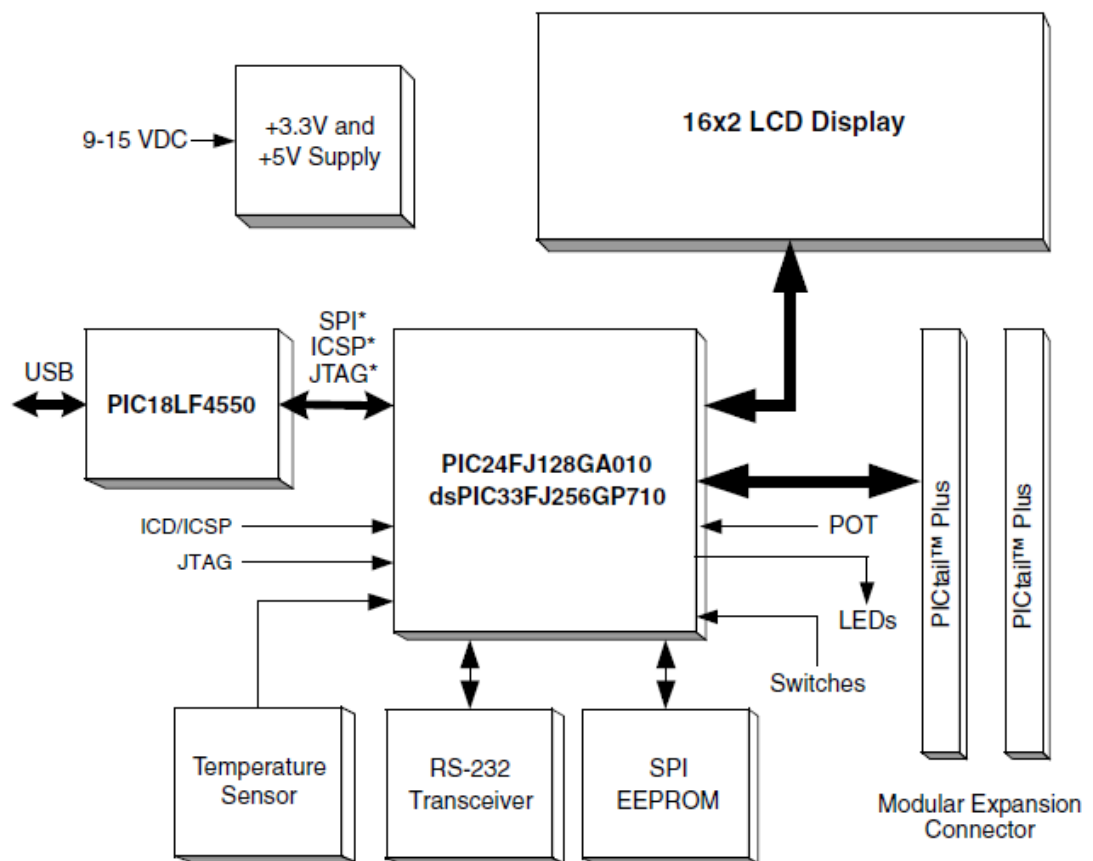


Phần cứng bao gồm các thành phần sau:

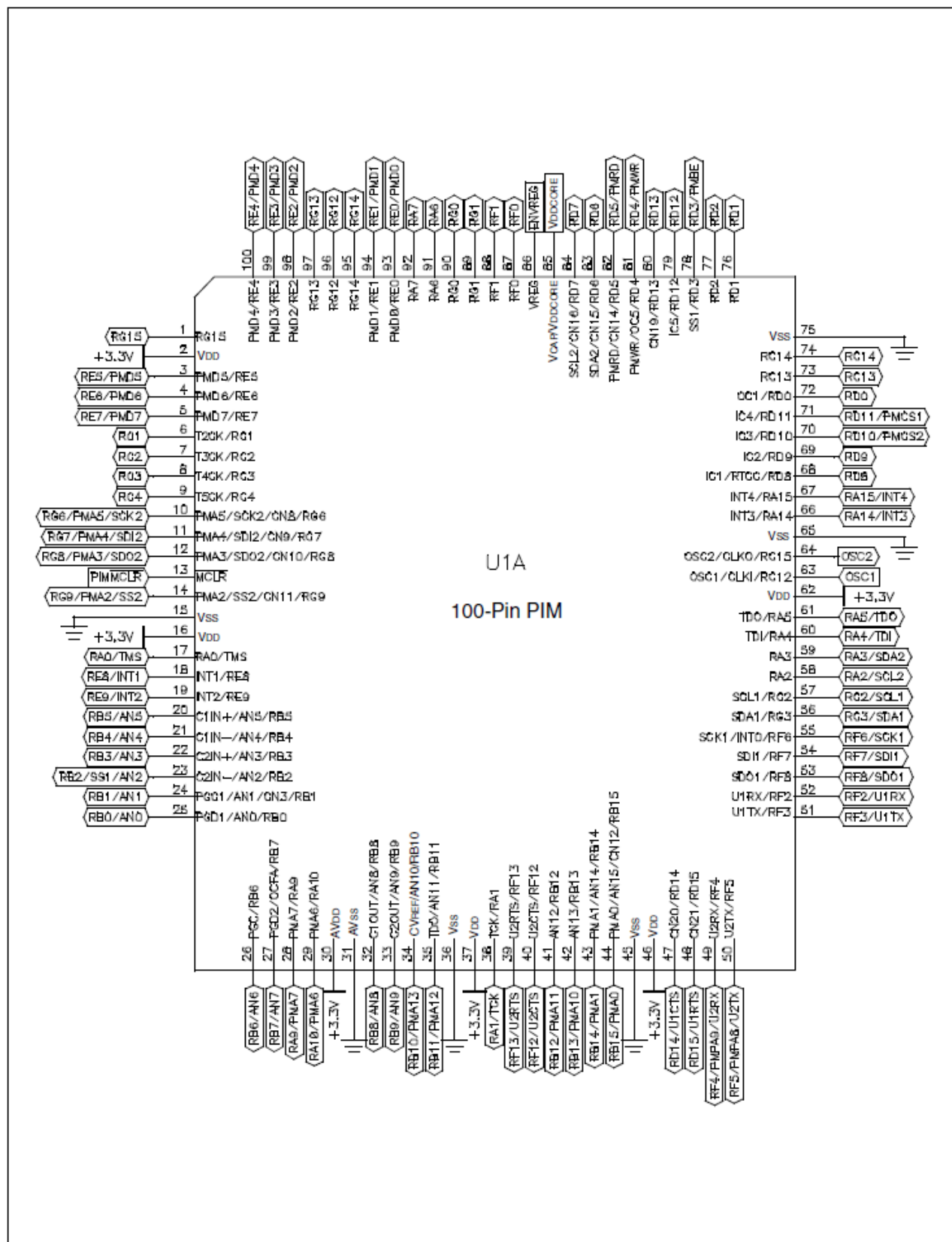
1. Đế cắm 100 chân PIM cho phép ghép nối với các vi xử lý PIC24F/24H/dsPIC33F. Trong mạch sử dụng Chip PIC24FJ128GA010.
2. Giắc cắm nguồn cấp 9VDC, trong mạch tạo ra nguồn 3.3V và 5V
3. Led báo nguồn
4. Cổng nối RS232 và các phần cứng phụ trợ
5. Sensor đo nhiệt
6. Cổng nối USB để truyền thông và phục vụ nạp trình/gỡ rối
7. Cổng kết nối tiêu chuẩn 6 dây ICD (in-Circuit Debugger) để kết nối Module MPLAB ICD 2.
8. Chuyển mạch lựa chọn sử dụng Chip trong mạch hoặc đế cắm.
9. Màn hình LCD 2 dòng 16 ký tự
10. Phần mạch cho phép bố trí màn hình graphic LCD.
11. Các nút ấn có chức năng Reset và người dùng sử dụng
12. Thiết lập kết nối với cổng vào Analog.
- 13.8 Led hiển thị
14. Sử dụng IC 74HCT4053 để lựa chọn cổng truyền thông nối tiếp
15. Serial EEPROM
16. Sử dụng thạch anh ngoài chính xác cao (8Mhz) và thạch anh cho đồng hồ thời gian thực RTCC (32.768 kHz)
17. Mạch phần cứng phục vụ ứng dụng người sử dụng.
18. Giắc cắm mở rộng PICtail
19. Giắc cắm kết nối cổng nạp trình PICkit 2
20. Giắc cắm JTAG

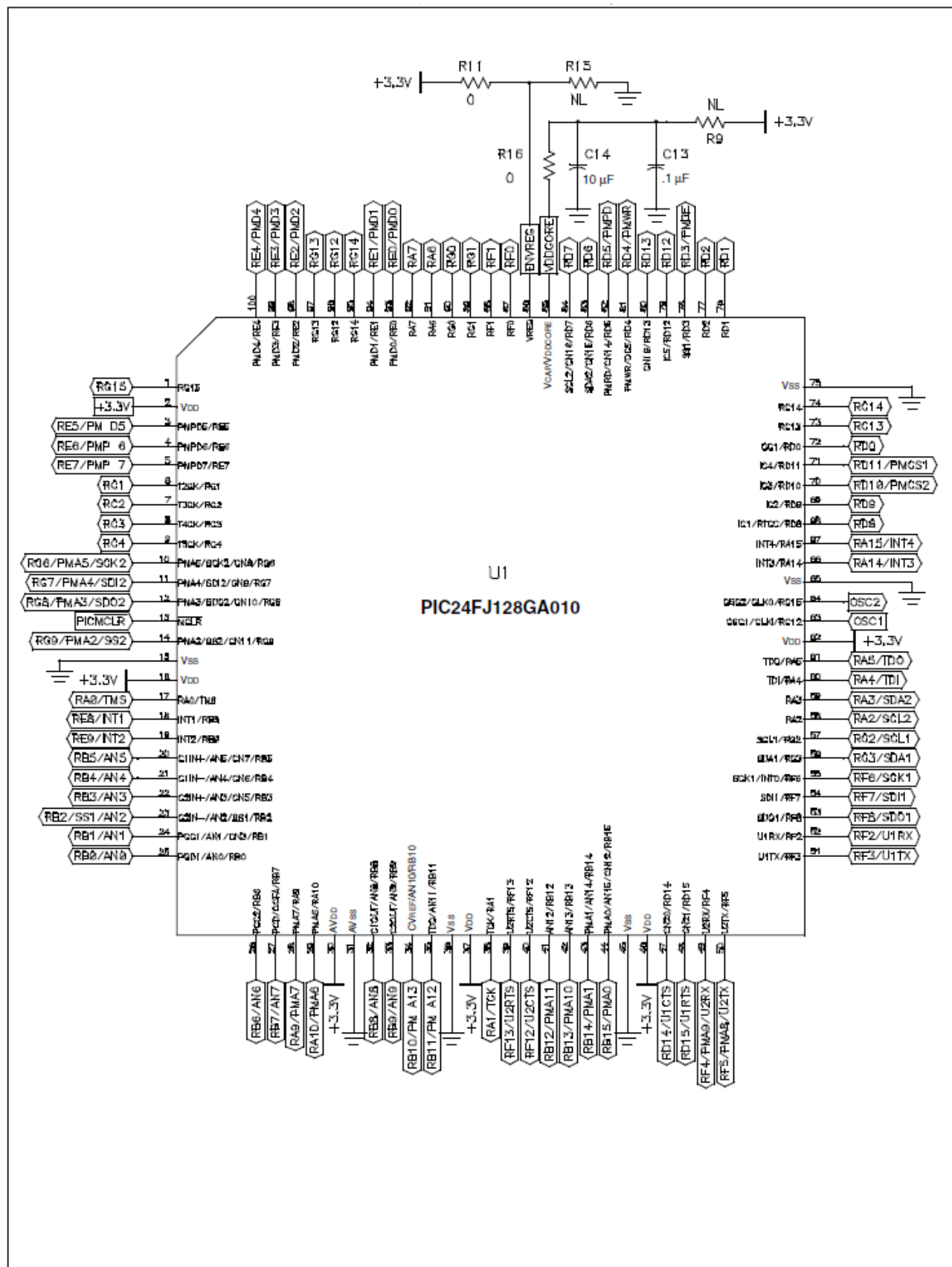


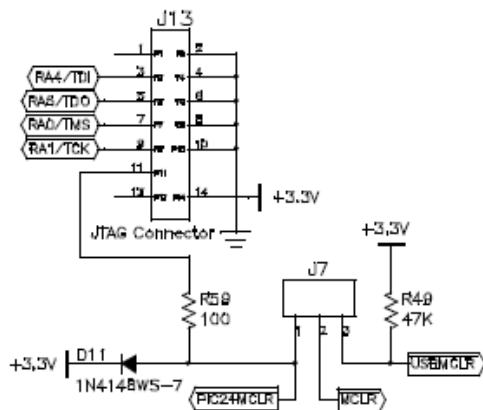
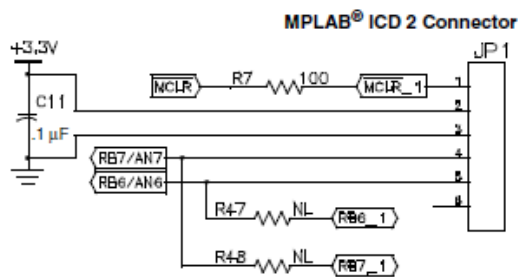
II. Sơ đồ mạch:



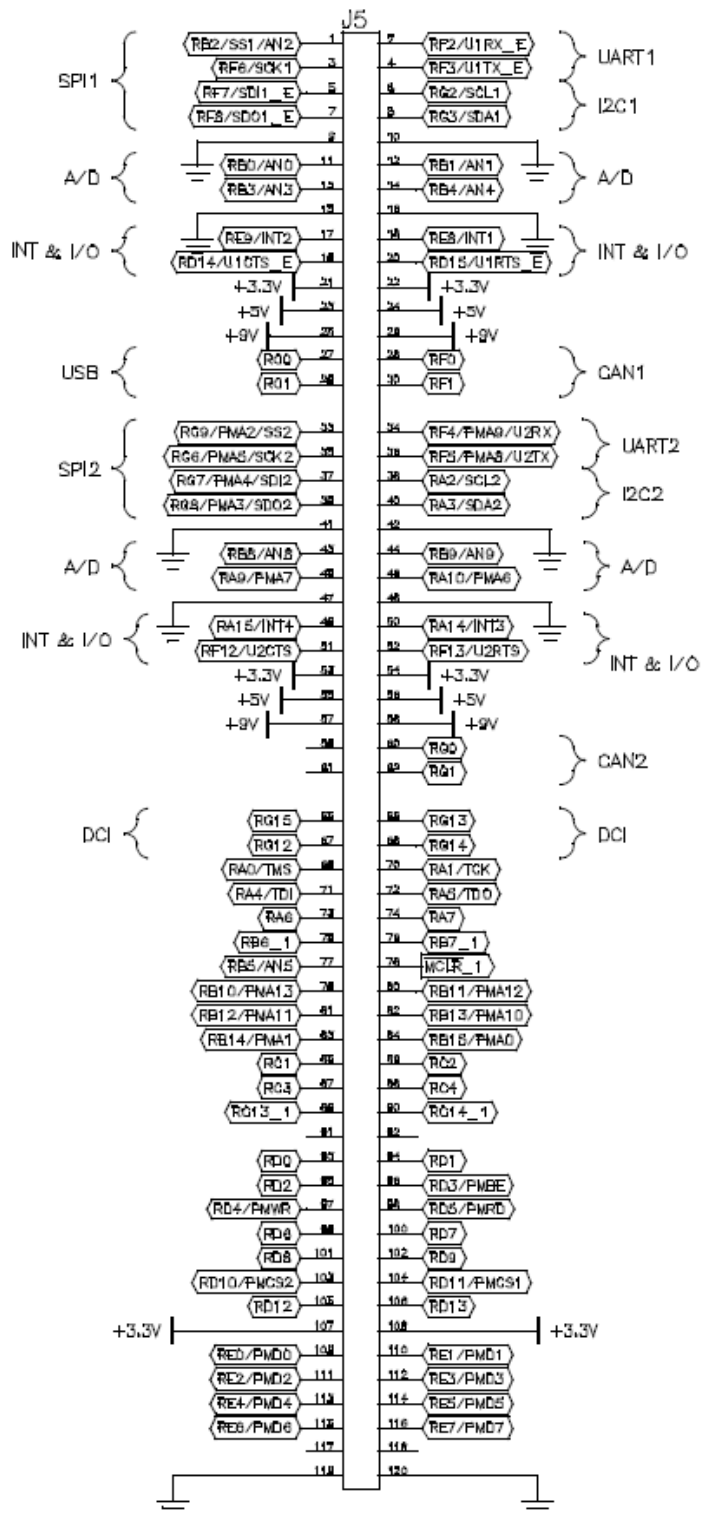
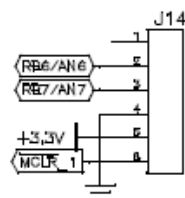
\* Hardware support only; firmware support for SPI™, JTAG and ICSP™ via USB are not available at this time.

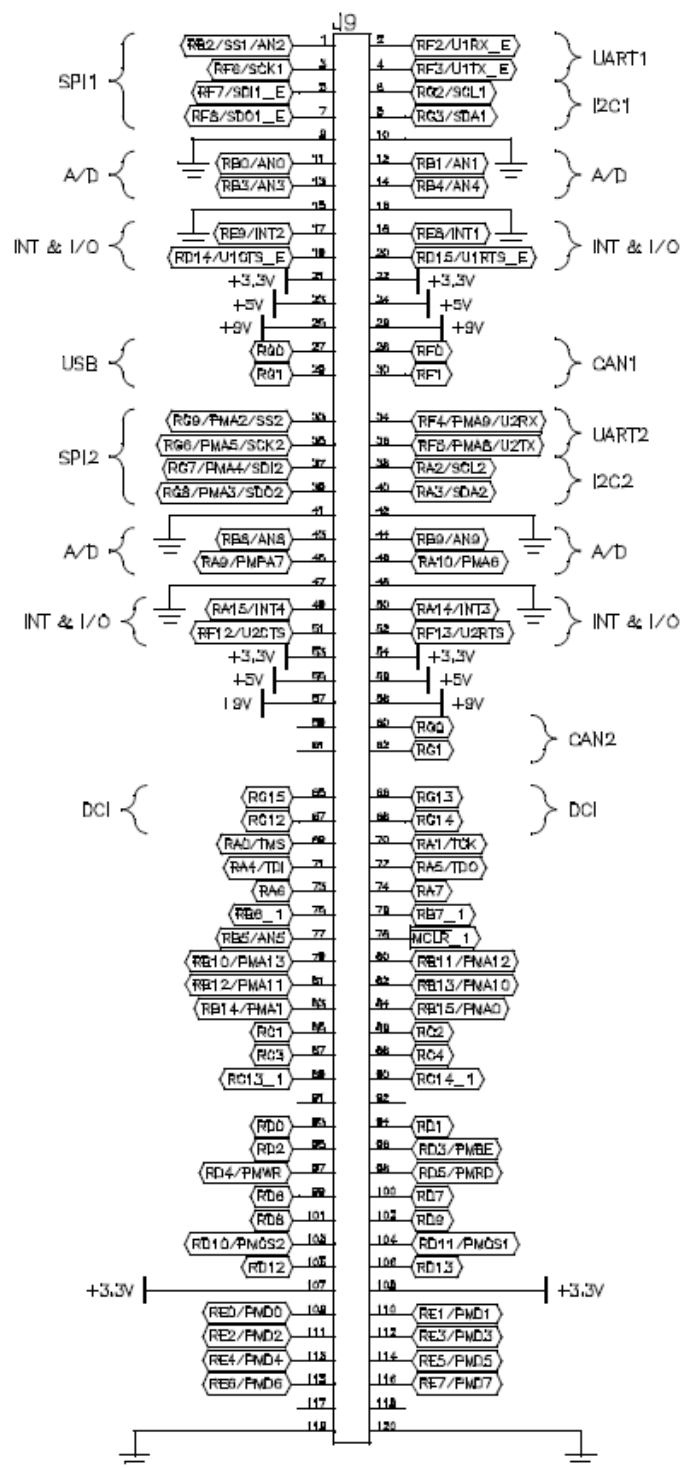
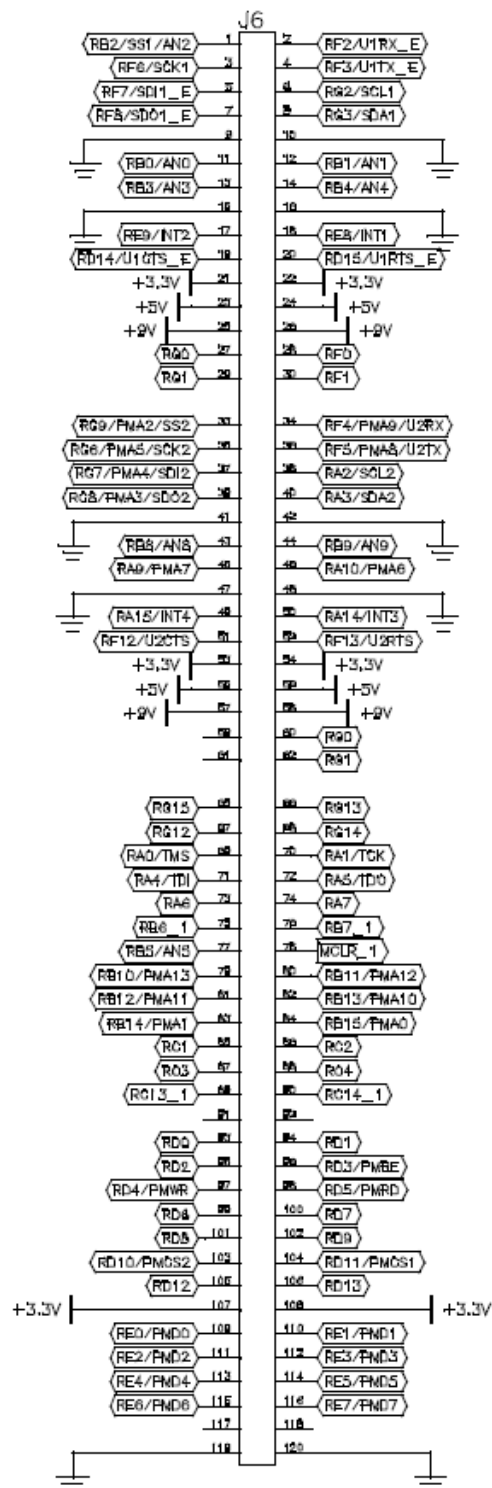




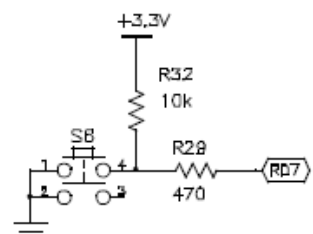
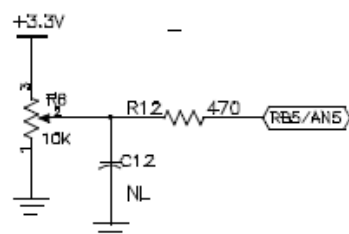
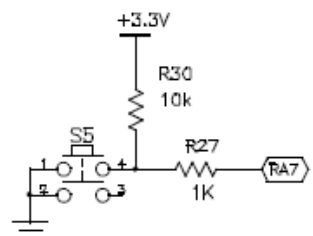
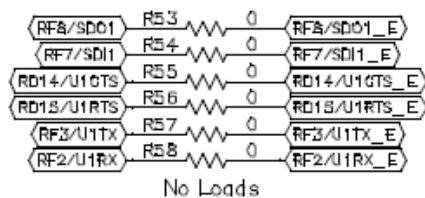
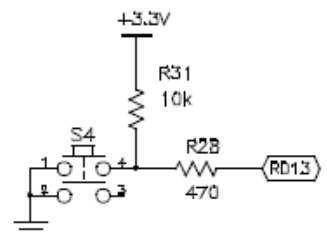
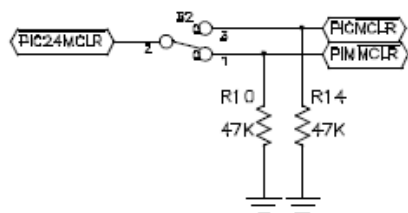
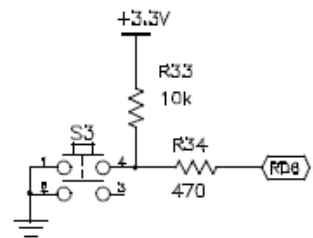
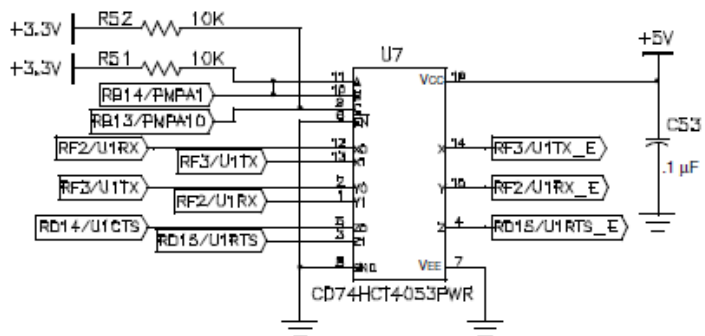
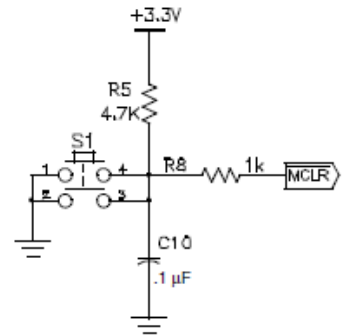
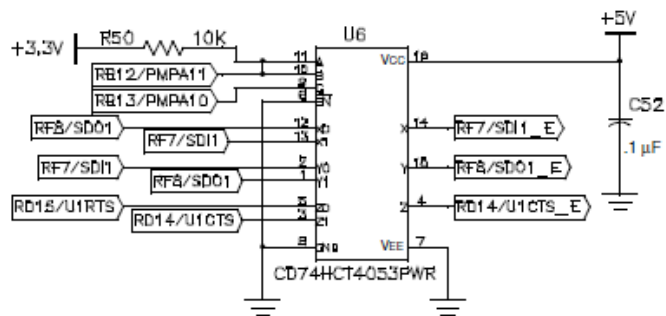


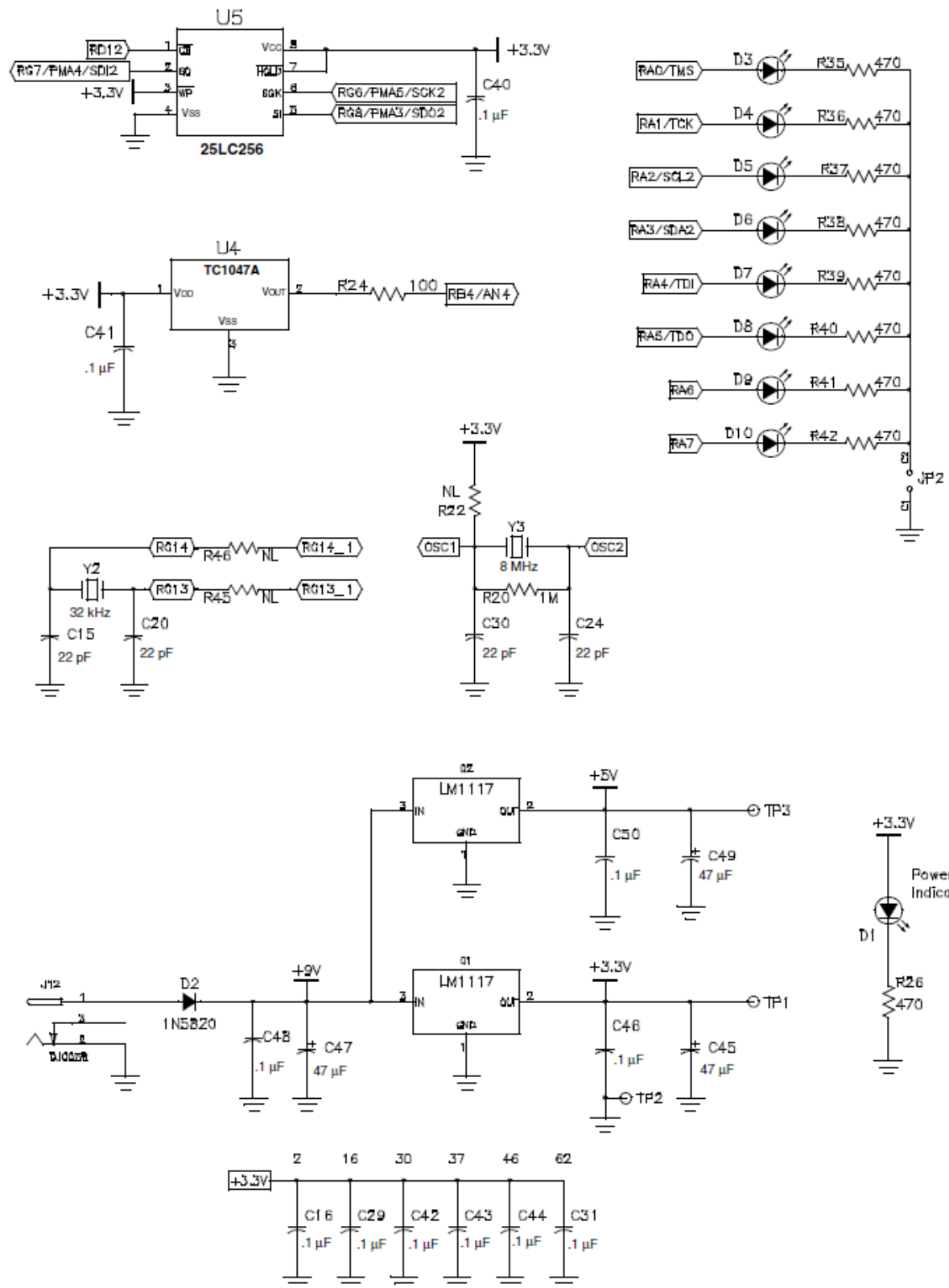
**PICkit™ 2 Programmer**

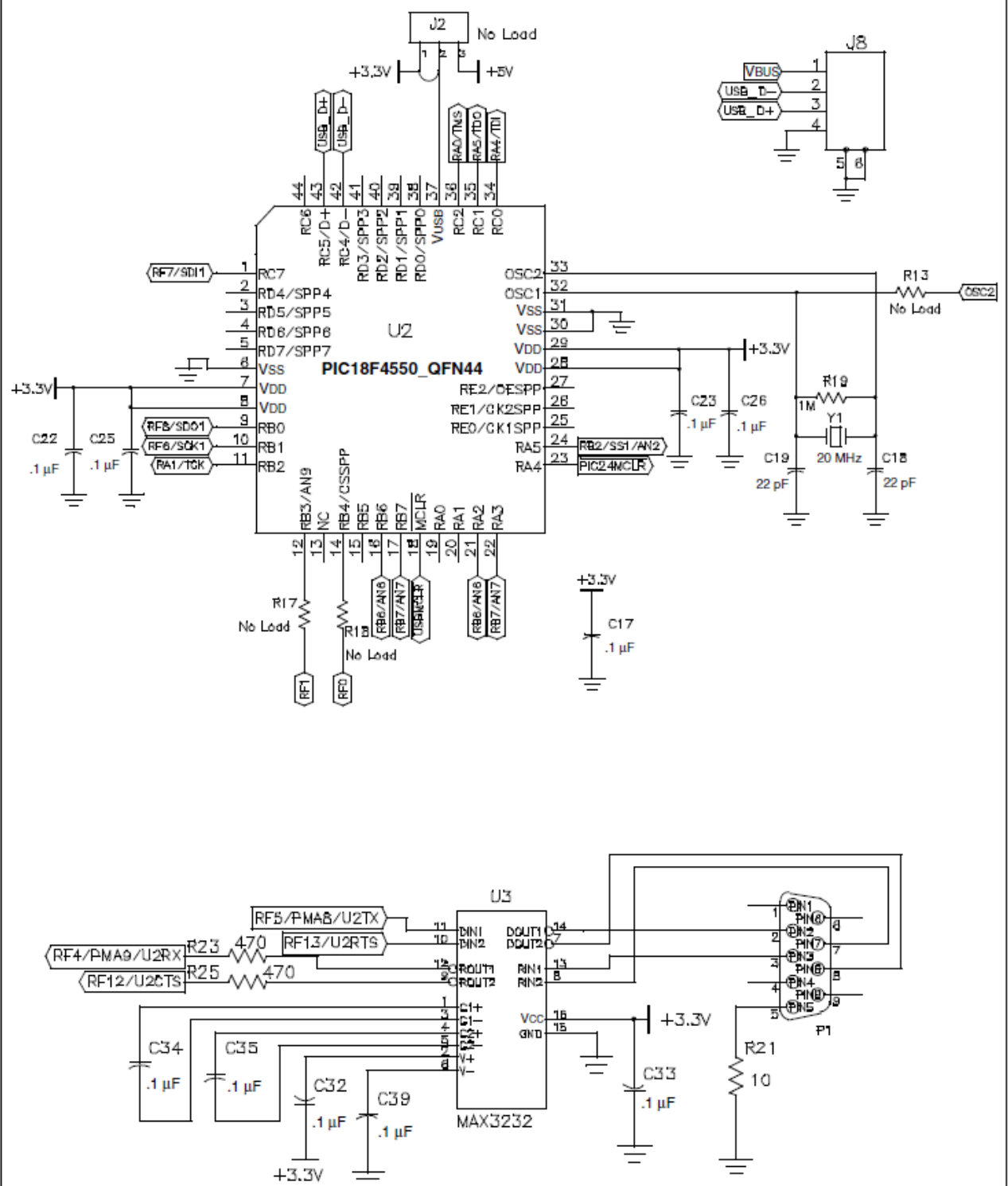


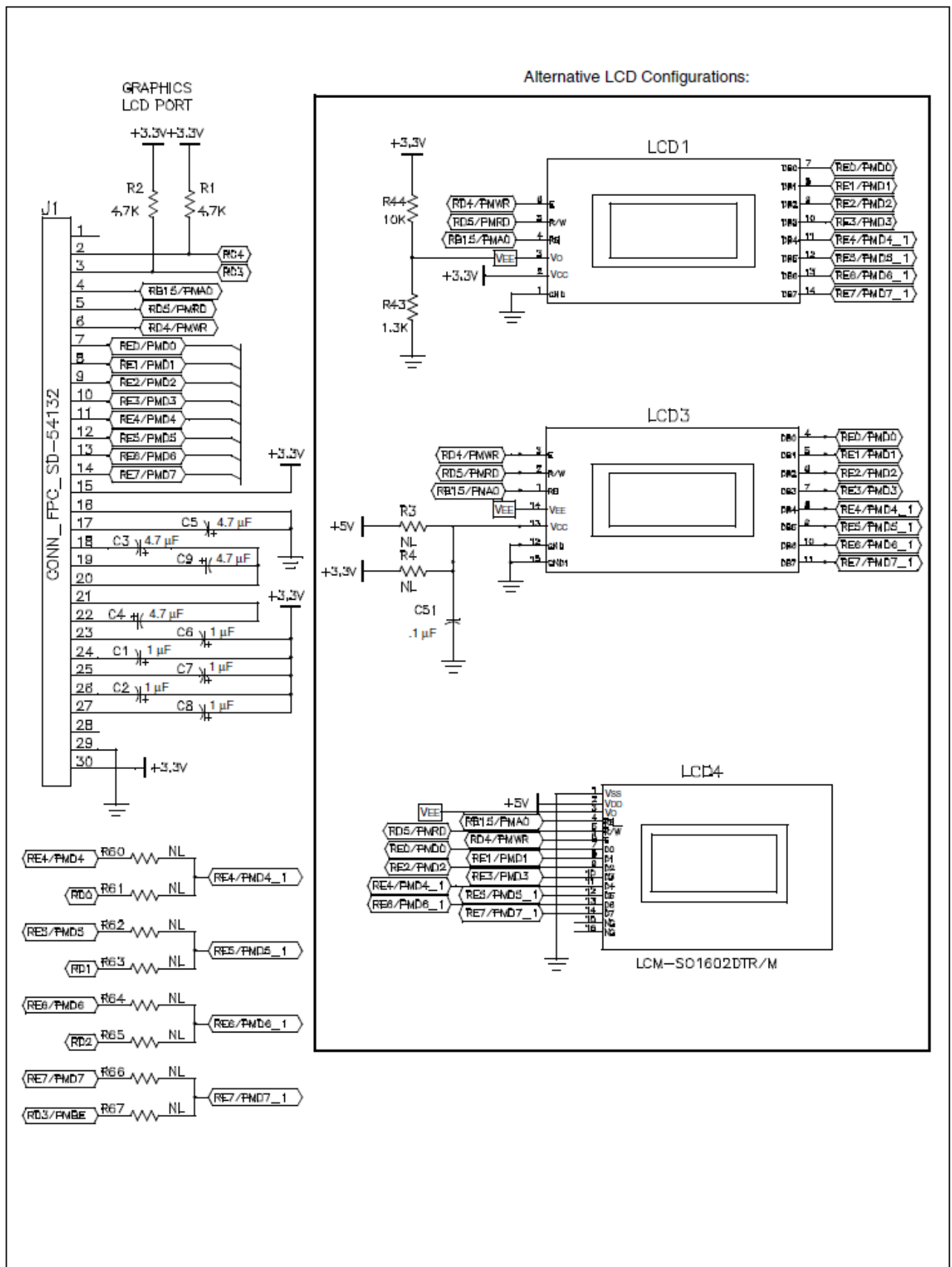












## CHƯƠNG 2. XÂY DỰNG CHƯƠNG TRÌNH PHẦN MỀM TRÊN MẠCH THỰC NGHIỆM.

### I. Tổng quan.

Trong chương này mô tả các thức sử dụng hệ phát triển MPLAB IDE và MPLAB ICD2 để xây dựng phần mềm và chạy trên mạch thực nghiệm Explorer 16. Các chương trình được viết cho Chip PIC24FJ128GA010.

Nội dung bao gồm:

- Khởi tạo 1 Project sử dụng Project Wizard.
- Biên dịch và liên kết chương trình nguồn và thiết lập cấp hình cho Chip.
- Cài đặt MPLAB IDE để MPLAB ICD2.
- Nạp trình cho chip sử dụng MPLAB ICD2.

Các bước thực hiện như sau:

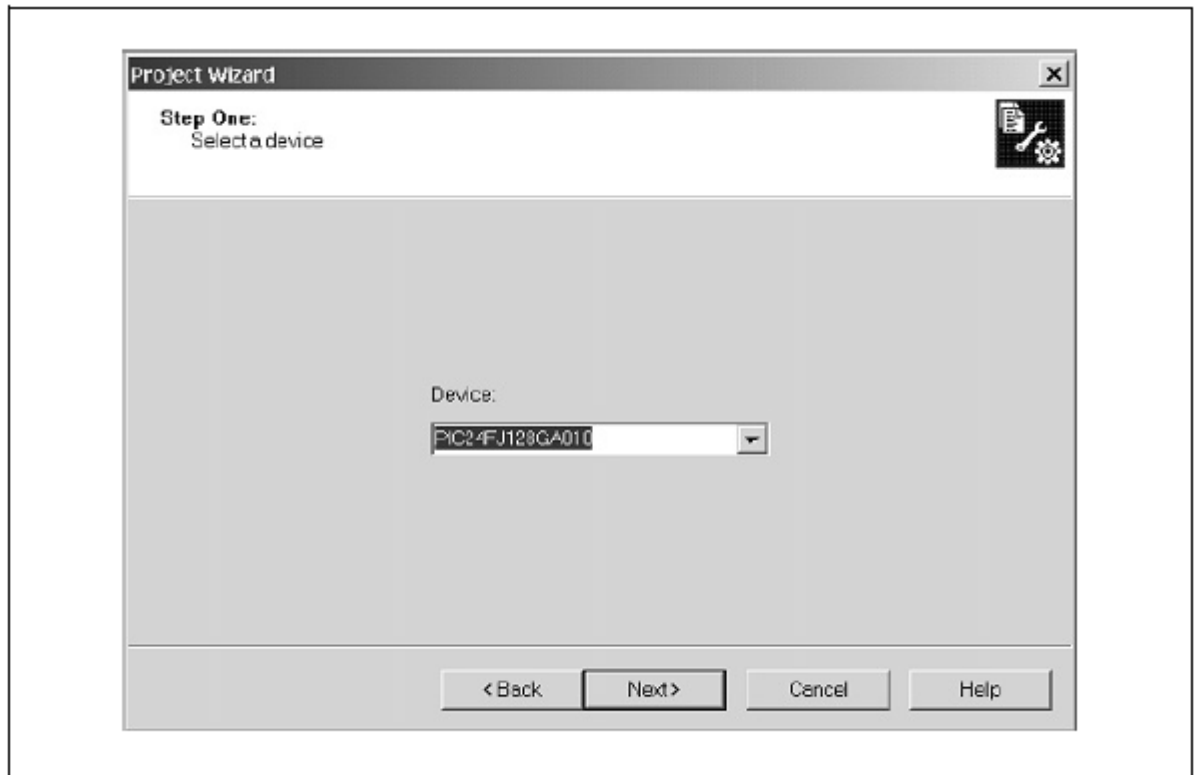
- Tạo 1 Project trong MPLAB IDE
- Dịch và liên kết mã nguồn.
- Nạp trình cho Chip sử dụng MPLAB ICD2.

### II. Khởi tạo Project.

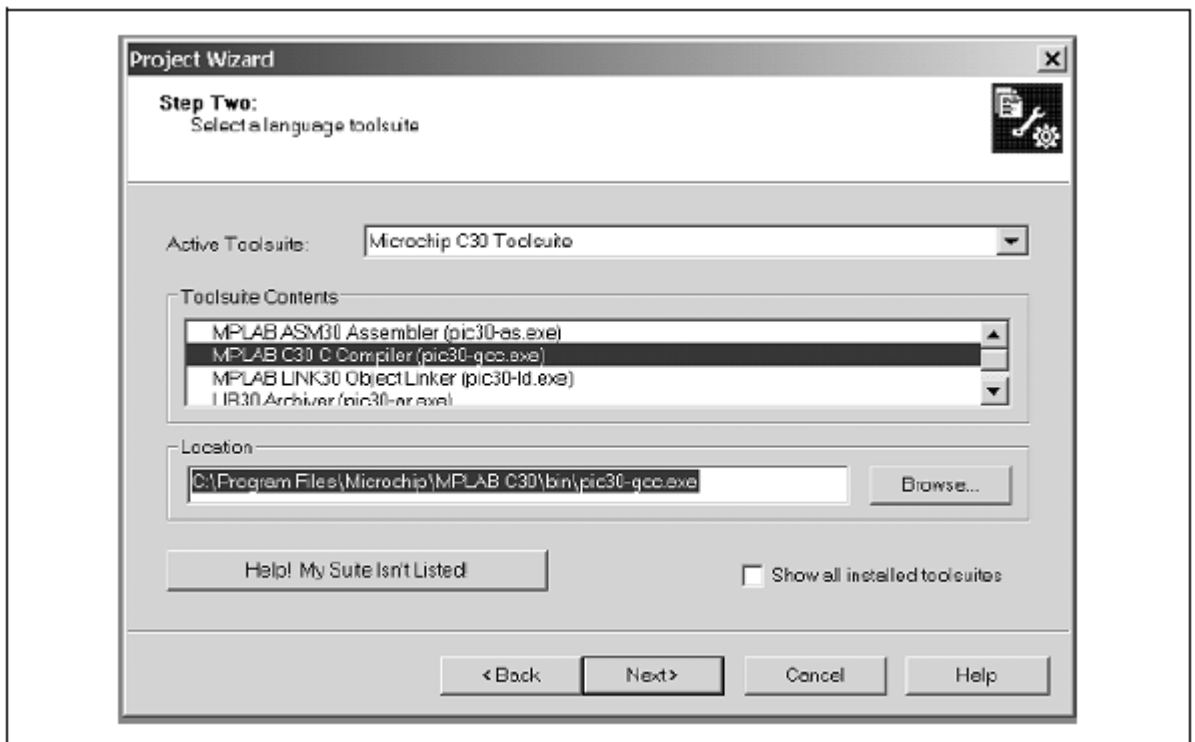
#### 1. Lựa chọn Chip sử dụng:

Thực hiện theo các bước sau:

- Chạy chương trình MPLAB IDE
- Đóng chương trình đang hoạt động hiện tại (File>Close Workspace)
- Từ menu Project chọn Project Wizard
- Trên bảng hội thoại đầu tiên, lựa chọn nút Next. Sẽ hiển thị màn hội thoại sau:



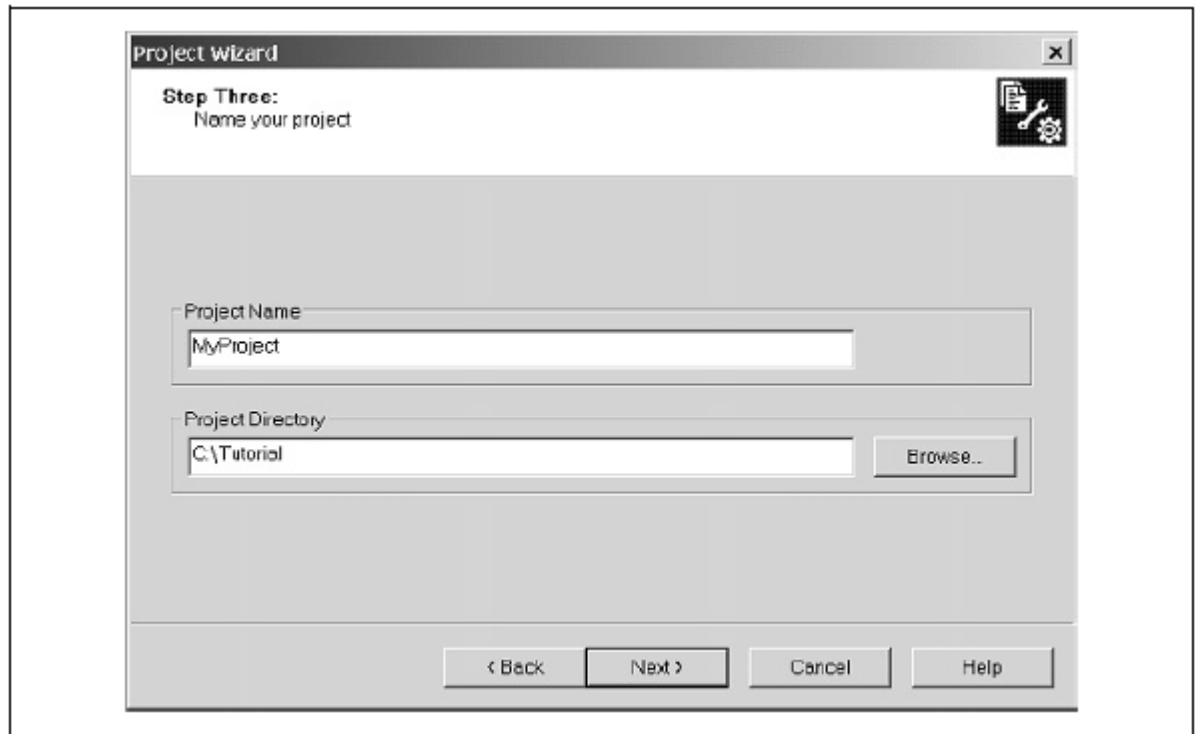
- Từ cửa sổ Device lựa chọn Chip PIC24FJ128GA010, sau đó chọn Next.



## 2. Lựa chọn công cụ lập trình.

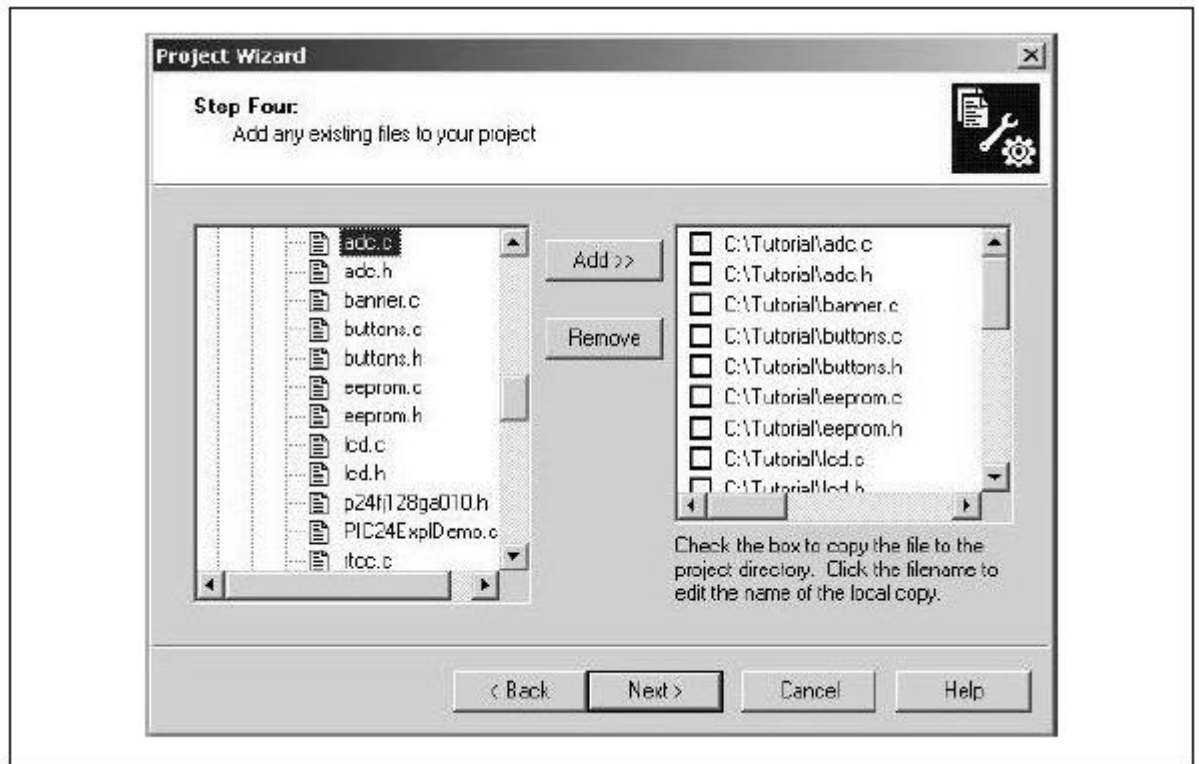
- Từ cửa sổ Active Toolsuite lựa chọn Microchip C30 Toolsuite. Công cụ này dùng để biên dịch và liên kết chương trình nguồn.
- Trong cửa sổ Toolsuite Contents lựa chọn MPLAB C30 Compiler (pic30-gcc.exe)

- Trong cửa sổ Location, ấn Browse... và chỉ vào đường dẫn C:\program Files\Microchip\MPLAB C30\bin\pic30-as.exe
- Với bộ liên kết MPLAB LINK C30 (pic30-ld.exe) lựa chọn Toolsuite Contents, ấn Browse... và chỉ vào đường dẫn: C:\program Files\Microchip\MPLAB C30\bin\pic30-ld.exe
- Ấn Next sẽ hiển thị cửa sổ như sau:



### 3. Đặt tên cho Project

- Trong cửa sổ Project Name gõ MyProject.
- Trong cửa sổ Project Directory, ấn Browse ... và chỉ vào đường dẫn C:\Tutorial để đặt Project vào thư mục lựa chọn.
- Ấn Next sẽ hiển thị bảng hội thoại.

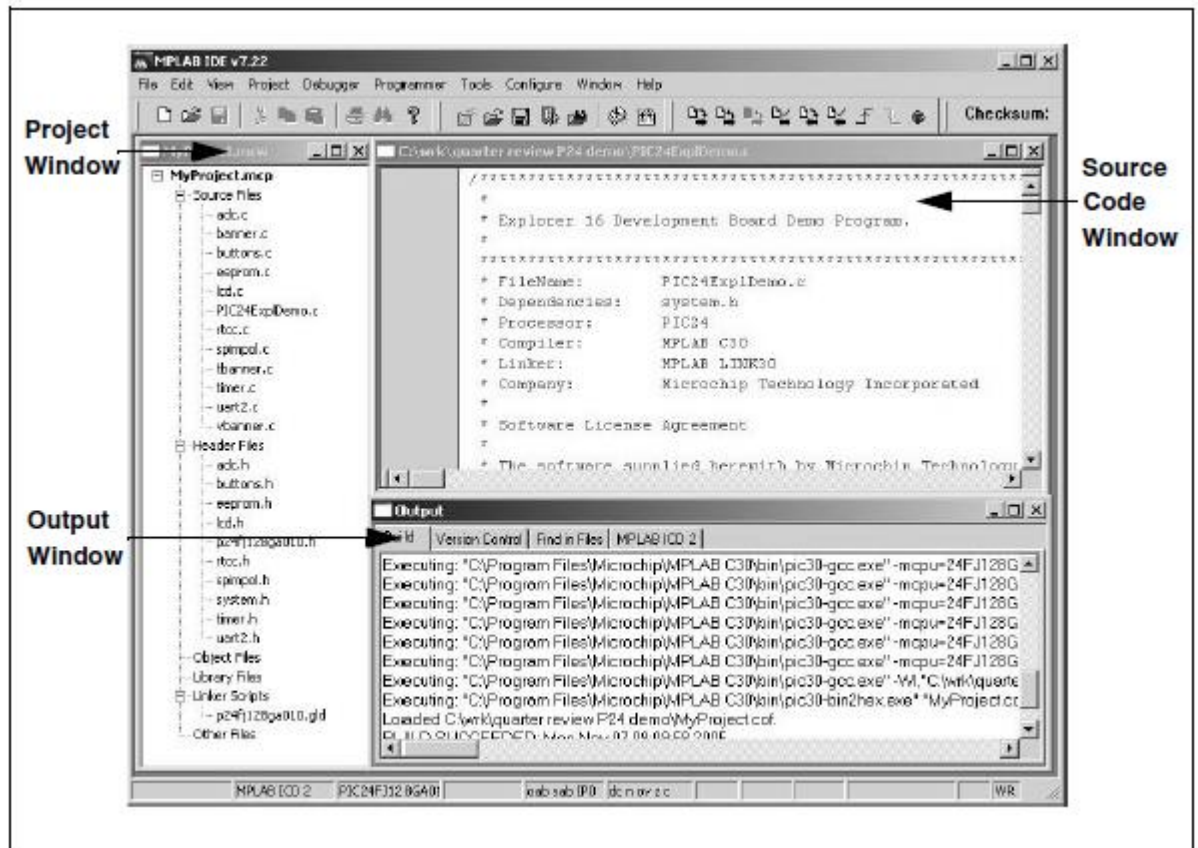


4. Đưa file đã soạn vào Project.

- Lựa chọn các file nguồn (.c) và file header (.h) trong thư mục C:\Tutorial, ấn Add>>
- Lựa chọn file p24f128ga010.gld trong thư mục C:\program Files\Microchip\MPLAB C30\support\gld, ấn Add.
- Ấn Next, sẽ hiển thị màn hình kết thúc, ấn Finish.

Kết quả tạo được Project như hình sau:

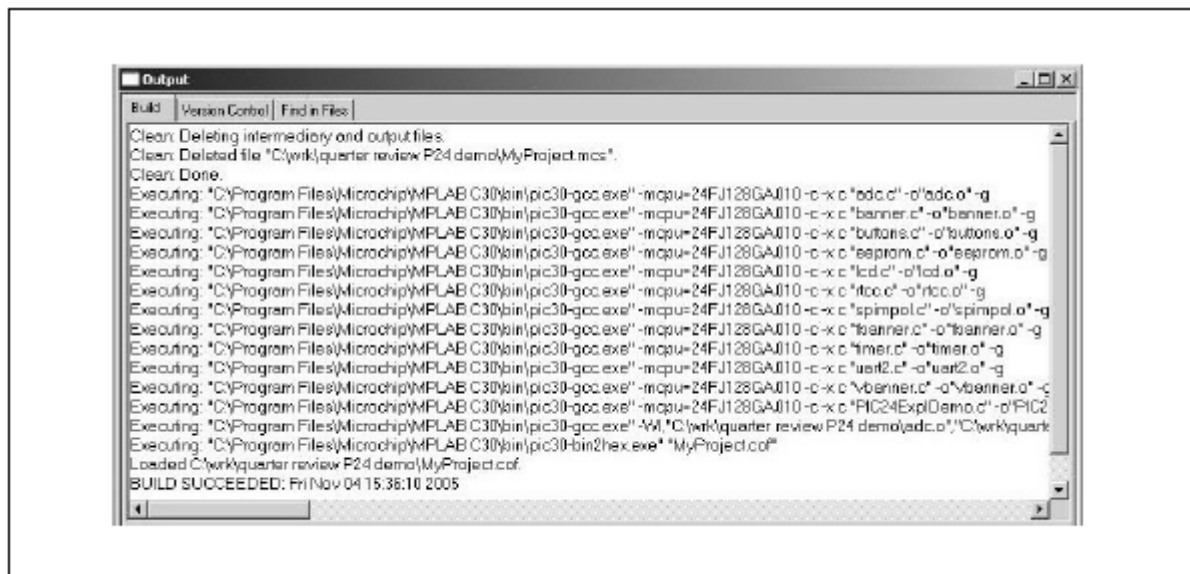




### III. Biên dịch và nạp trình Project

#### 1. Biên dịch

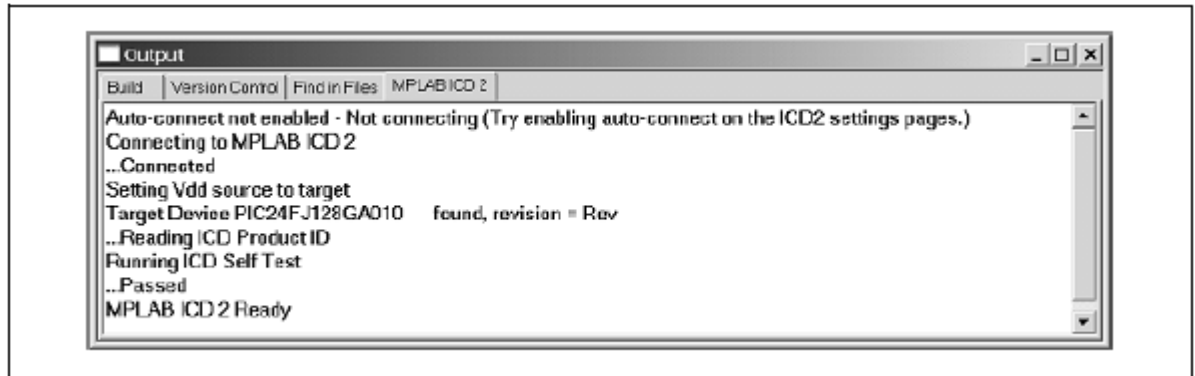
Từ menu chính lựa chọn Project>Make. Khi chưa trình không có lỗi về cú pháp sẽ có thông báo “BUILD SUCCEEDED”, lúc đó sẵn sàng để nạp chương trình cho Chip.



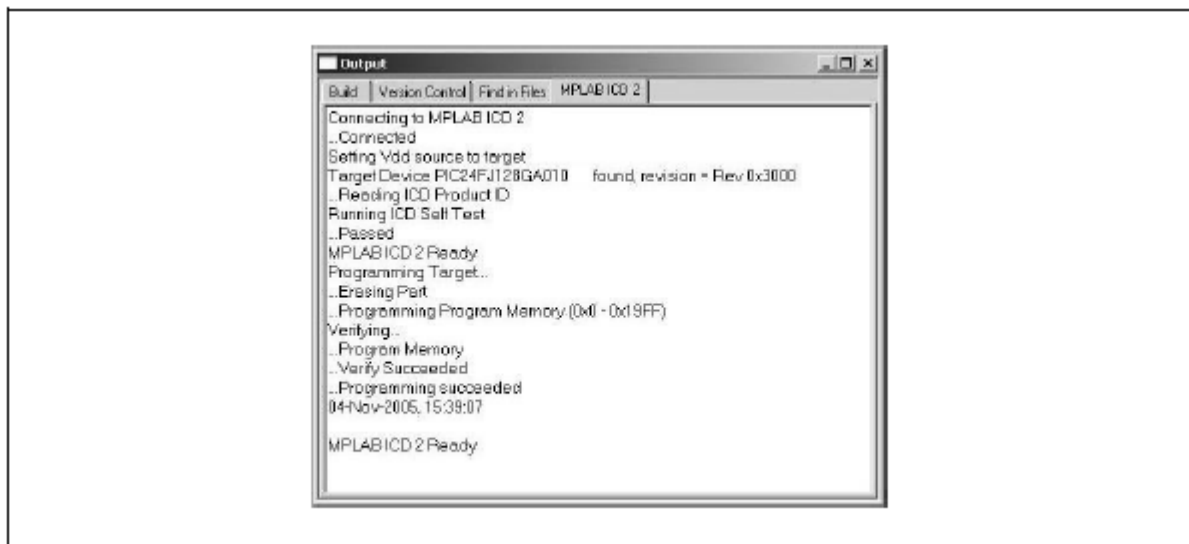
#### 2. Nạp trình

- Kết nối MPLAB ICD2 với máy tính thông qua cổng USB

- Kết nối MPLAB ICD2 với mạch Explorer 16 qua giắc cắm RJ-11
- Cấp nguồn cho mạch Explorer 16
- Từ menu Debugger, ấn Select Tool>MPLAB ICD2
- Từ menu Debugger, chọn Connect sẽ hiển thị hình sau:



- Để nạp trình: từ menu Debugger chọn Program sẽ hiển thị như hình dưới.



#### IV. CÁC TÍNH NĂNG CƠ BẢN CỦA MẠCH THỰC HÀNH

Chương trình mẫu (Demo) bao gồm 3 nội dung được hiển thị trên LCD như sau: Hiển thị các thông số kỹ thuật của vi xử lý PIC24FJ128GA010, hiển thị giá trị điện áp và nhiệt độ, hiển thị đồng hồ thời gian thực RTC.

##### 1. Hiển thị các thông số kỹ thuật của PIC24:

Trên màn hình sẽ hiển thị lần lượt các thông số của Chip xử lý PIC24FJ128GA010, để thoát khỏi chế độ này ấn S4.

##### 2. Hiển thị điện áp và nhiệt độ

Chế độ hiển thị điện áp và nhiệt độ sử dụng chương trình nguồn vbanner.c và ADC.c. Module ADC sẽ đo giá trị điện áp tương tự từ mạch và chuyển đổi chúng lên màn hình hiển thị LCD. Giá trị điện áp được lấy từ triết áp R6 và hiển thị mức điện

áp từ 0.00V đến 3.29V tại dòng thứ nhất của LCD. Giá trị nhiệt độ lấy từ IC TC1074A và được hiển thị tại dòng 2 của LCD. Đồng thời tự động chuyển đổi đơn vị đo giữa độ C và độ F.

Ở chế độ này có thể lưu giá trị nhiệt độ hiện tại vào bộ nhớ EEPROM trên mạch bằng cách ấn F5. Ấn phím S6 trên màn hình sẽ hiển thị giá trị nhiệt độ hiện tại và nhiệt độ đã lưu. Chữ “M” ở bên phải màn hình LCD biểu thị giá trị nhiệt độ đã lưu.

Để thoát khỏi chế độ này ấn phím S4.

### 3. Đồng hồ thời gian thực và lịch.

Chương trình hiển thị đồng hồ thời gian thực và Lịch dùng các file rtcc.c và tbanner.c.

Trong chế độ hiển thị đồng hồ, sử dụng các phím sau để cài đặt:

- S3 để chuyển về chế độ đặt thời gian, cho phép người sử dụng đặt ngày tháng năm giờ phút giây theo thời gian hiện tại.
- S4 để xác nhận giá trị đặt và chuyển sang giá trị tiếp theo.
- S5 để giảm giá trị
- S6 để tăng giá trị.

Những dữ liệu đưa lên màn hình LCD cũng được truyền qua cổng RS232 với giao thức UART. Tốc độ truyền là 19200 baud, 8 bit data, 1 stop bit và không kiểm tra chẵn lẻ.

Lưu đồ hoạt động của chương trình như sau:



## CHƯƠNG 3. CÁC BÀI THỰC HÀNH CƠ BẢN.

### I. THỰC HÀNH ĐIỀU KHIỂN LED.

```
#include <p24fxxx.h>
```

```
#define LED3 LATAbits.LATA0
```

```
#define LED4 LATAbits.LATA1
```

```
#define LED5 LATAbits.LATA2
```

```
#define LED6 LATAbits.LATA3
```

```
#define LED7 LATAbits.LATA4
```

```
#define LED8 LATAbits.LATA5
```

```
#define LED9 LATAbits.LATA6
```

```
#define LED10 LATAbits.LATA7
```

```
#define TRIS_LED3 TRISAbits.TRISA0
```

```
#define TRIS_LED4 TRISAbits.TRISA1
```

```
#define TRIS_LED5 TRISAbits.TRISA2
```

```
#define TRIS_LED6 TRISAbits.TRISA3
```

```
#define TRIS_LED7 TRISAbits.TRISA4
```

```
#define TRIS_LED8 TRISAbits.TRISA5
```

```
#define TRIS_LED9 TRISAbits.TRISA6
```

```
#define TRIS_LED10 TRISAbits.TRISA7
```

```
unsigned int count;
```

```
_CONFIG1( JTAGEN_OFF & GCP_OFF & GWRP_OFF & COE_OFF & FWDTEN_OFF & ICS_PGx2)
```

```
_CONFIG2( FCKSM_CSDCMD & OSCIOFNC_OFF & POSCMOD_XT & FNOSC_PRI)
```

```
void TimerInit(void) // CAI DAT THAM SO CHO TIME 1 HOAT DONG
```

```
{
```

```
    PR1 = 10000; // giá trị so sánh với tín hiệu đầu vào
```

```
    IPC0bits.T1IP = 5; // mức độ ưu tiên của IPC0 ở mức 5
```

```
    T1CON = 0b10000000000010000; // T1con ở vị trí thu 5 của thanh ghi có giá trị 1, bit 5=1 thì hệ số chia 1:8
```

```
    IFS0bits.T1IF = 0; // có ngắt bằng không
```

```
    IEC0bits.T1IE = 1; // cho phép ngắt khi có sự kiện, thì hàm sau được gọi
```

```
}
```

```
void __attribute__((interrupt, shadow, auto_psv)) _T1Interrupt(void) // 50Hz = 20ms
```

```
{
```

```

// Clear flag
IFS0bits.T1IF = 0;
    count++;
    if (count<=100)
    {
        LED3=1;
        LED4=0;
        LED5=0;
    }
    if (count>100&&count<=115)
    {
        LED3=0;
        LED4=1;
        LED5=0;
    }
    if (count>115 && count<=240)
    {
        LED3=0;
        LED4=0;
        LED5=1;
    }
    if(count>240) count=0;

}

void LEDInit(void)// khoi tao cong ra cho cac Led
{
    TRIS_LED3 = 0;
    TRIS_LED4 = 0;
    TRIS_LED5 = 0;
    TRIS_LED6 = 0;
    TRIS_LED7 = 0;
    TRIS_LED8 = 0;
    TRIS_LED9 = 0;
    TRIS_LED10 = 0;
}

int main ()
{

```

```

    LEDInit();
    TimerInit();
    while(1)
{
}
}

```

## II. THỰC HÀNH PHÍM BẤM

```

#include <p24fxxx.h>

#define TRIS_d0    TRISDbits.TRISD0
    #define TRIS_d1    TRISDbits.TRISD1
    #define TRIS_d2    TRISDbits.TRISD2
    #define TRIS_d3    TRISDbits.TRISD3

#define d0        LATDbits.LATD0
    #define d1            LATDbits.LATD1
    #define d2            LATDbits.LATD2
    #define d3            LATDbits.LATD3

#define kb0 PORTDbits.RD6
#define TRIS_KB0    TRISDbits.TRISD6
#define kb1 PORTDbits.RD13
#define TRIS_KB1    TRISDbits.TRISD13
#define kb2 PORTDbits.RD7
#define TRIS_KB2    TRISDbits.TRISD7
#define LED_PORT LATA
    #define TRIS_LED3    TRISAbits.TRISA0
        #define TRIS_LED4    TRISAbits.TRISA1
        #define TRIS_LED5    TRISAbits.TRISA2
        #define TRIS_LED6    TRISAbits.TRISA3
        #define TRIS_LED7    TRISAbits.TRISA4
        #define TRIS_LED8    TRISAbits.TRISA5
        #define TRIS_LED9    TRISAbits.TRISA6
        #define TRIS_LED10    TRISAbits.TRISA7

unsigned int num,i,j,delay;
unsigned long k;
unsigned char donvi,truc,tram,ngin;
char pre_kb0,pre_kb1,pre_kb2;

```

```

unsigned char number[10] = {0xc0,0xf9,0xa4,0xb0,0x99,0x92,0x82,0xf8,0x80,0x90}; //0-9
_CONFIG1( JTAGEN_OFF & GCP_OFF & GWRP_OFF & COE_OFF & FWDTEN_OFF & ICS_PGx2)
_CONFIG2( FCKSM_CSDCMD & OSCIOFNC_OFF & POSCMOD_XT & FNOSC_PRI)

void TimerInit(void)// CAI DAT THAM SO CHO TIME 1 HOAT DONG
{
    PRI = 5000;// giá trị so sánh với tín hiệu đầu vào
    IPC0bits.T1IP = 5; //mức độ ưu tiên của IPCO ở mức 5
    T1CON = 0b1000000000010000;//T1con ở vị trí thu 5 của thanh ghi có giá trị 1, bit 5=1 thì hệ số //chia 1:8
    IFS0bits.T1IF = 0;// có ngắt hay không
    IEC0bits.T1IE= 1; //cho phép ngắt khi có sự kiện, thì hàm sau được gọi
}

void __attribute__((interrupt, shadow, auto_psv)) _T1Interrupt(void) // 50Hz =20ms
{
    // Clear flag
    IFS0bits.T1IF = 0;
}

void LEDInit(void)// khởi tạo cổng ra cho các Led
{
    TRIS_LED3 = 0;
    TRIS_LED4 = 0;
    TRIS_LED5 = 0;
    TRIS_LED6 = 0;
    TRIS_LED7 = 0;
    TRIS_LED8 = 0;
    TRIS_LED9 = 0;
    TRIS_LED10 = 0;
    TRIS_KB0=1;
    TRIS_KB1=1;
    TRIS_KB2=1;
}

int main ()
{
    LEDInit();
    TimerInit();

    while(1)

```



```

{
    ngin=num/1000;

    LED_PORT=0xff;
    d3=1;d0=0;d1=1;d2=1;
    LED_PORT=number[ngin];

    for (i=0;i<100;i++);
j=num%1000;
tram=j/100;

    LED_PORT=0xff;
    d0=1;d1=0;d2=1;d3=1;
    LED_PORT=number[tram];
    for (i=0;i<100;i++);
j=j%100;

    truc=j/10;

    LED_PORT=0xff;
    d0=1;d1=1;d2=0;d3=1;
    LED_PORT=number[truc];
    for (i=0;i<100;i++);
    donvi=j%10;

    LED_PORT=0xff;
    d0=1;d1=1;d2=1;d3=0;
    LED_PORT=number[donvi];
    for (i=0;i<100;i++);
    //delay++;

    //if(delay==10)
    if(!kb0)

    pre_kb0=1;
    if(pre_kb0&&kb0)
    {
        delay=0;
    }
    pre_kb0=0;

    num=0;

```

```

    }
    if(!kb1)
        pre_kb1=1;
    if(pre_kb1&&kb1)
        {
            delay=0;

            pre_kb1=0;
            num++;

            if(num==10000)
                num=0;
        }
    if(!kb2)

        pre_kb2=1;
    if(pre_kb2&&kb2)
        {
            delay=0;

            pre_kb2=0;

            if(num>0)

                num--;
        }
    }
}

```

### III. THỰC HÀNH HIỂN THỊ LED 7SEG

```
#include <p24fxxx.h>
```

```

#define TRIS_d0    TRISDbits.TRISD0
#define TRIS_d1    TRISDbits.TRISD1
#define TRIS_d2    TRISDbits.TRISD2
#define TRIS_d3    TRISDbits.TRISD3

```

```

#define d0          LATDbits.LATD0
#define d1          LATDbits.LATD1
#define d2          LATDbits.LATD2
#define d3          LATDbits.LATD3

```

```

#define LED3  LATAbits.LATA0
#define LED4  LATAbits.LATA1
#define LED5  LATAbits.LATA2
#define LED6  LATAbits.LATA3

```

```

#define LED7 LATAbits.LATA4
#define LED8 LATAbits.LATA5
#define LED9 LATAbits.LATA6
#define LED10 LATAbits.LATA7
#define PORTA LATA
#define TRIS_LED3 TRISAbits.TRISA0
#define TRIS_LED4 TRISAbits.TRISA1
#define TRIS_LED5 TRISAbits.TRISA2
#define TRIS_LED6 TRISAbits.TRISA3
#define TRIS_LED7 TRISAbits.TRISA4
#define TRIS_LED8 TRISAbits.TRISA5
#define TRIS_LED9 TRISAbits.TRISA6
#define TRIS_LED10 TRISAbits.TRISA7

unsigned int count,i,j,ngan,tram,chuc,dvi,y,x;
char pre_kb0,pre_kb1,pre_kb2;
unsigned char ma7d[10] = {0xc0,0xf9,0xa4,0xb0,0x99,0x92,0x82,0xf8,0x80,0x90}; //0-9
unsigned int i;
_CONFIG1( JTAGEN_OFF & GCP_OFF & GWRP_OFF & COE_OFF & FWDTEN_OFF & ICS_PGx2)
_CONFIG2( FCKSM_CSDCMD & OSCIOFNC_OFF & POSCMOD_XT & FNOSC_PRI)

void TimerInit(void)// CAI DAT THAM SO CHO TIME 1 HOAT DONG
{
    PR1 = 12500;// giá trị so sánh voi tín hiệu đầu vào 8000000/2/64=62500, 1/6250=0,000016s. 12500 xung
    đầu vào thì sau 0,2s

    IPC0bits.T1IP = 5; //mức độ ưu tiên của IPCO ở mức 5
    T1CON = 0b1000000000100000;//T1con ở vị trí thu 5 của thanh ghi có giá trị 1, bit 5=1 thì hệ số //chia
    1:64
    IFS0bits.T1IF = 0;// có ngắt bằng không
    IEC0bits.T1IE= 1; //cho phép ngắt khi có cờ tràn, thì hàm sau được gọi
}

void __attribute__((interrupt, shadow, auto_psv)) _T1Interrupt(void) // 50Hz = 10ms
{
    // Clear flag
    IFS0bits.T1IF = 0;
    count++;
    if (count==1000)

```

```

    count=0;
}

void LEDInit(void)// khoi tao cong ra cho cac Led
{
    TRIS_LED3 = 0;
    TRIS_LED4 = 0;
    TRIS_LED5 = 0;
    TRIS_LED6 = 0;
    TRIS_LED7 = 0;
    TRIS_LED8 = 0;
    TRIS_LED9 = 0;
    TRIS_LED10 = 0;
    TRIS_d3=0;
    TRIS_d2=0;
    TRIS_d1=0;
    TRIS_d0=0;
}

int main()
{

    LEDInit();
    TimerInit();

    while(1)
    {

        dvi=((count%1000)%100)%10;
        PORTA=0xff;d3=1;d2=0;d1=0;d0=0;
        PORTA=ma7d[dvi];
        chuc=((count%1000)%100)/10;
        tram=(count%1000)/100;
        ngan=count/1000;
        for(i=0;i<1000;i++); //delay
        PORTA=0xff;d3=0;d2=1;d1=0;d0=0;
        PORTA=ma7d[chuc];
        for(i=0;i<1000;i++); //delay
        PORTA=0xff;d3=0;d2=0;d1=1;d0=0;
    }
}

```

```

        PORTA=ma7d[tram];
    for(i=0;i<1000;i++); //delay
        PORTA=0xff;d3=0;d2=0;d1=0,d0=1;
        PORTA=ma7d[ngan];
    for(i=0;i<1000;i++); //delay
    }

}

```

#### IV. THỰC HÌNH ĐIỀU KHIỂN MÁY GIẶT

```
#include <p24fxxx.h>
```

```
#define Start PORTDbits.RD6
```

```
#define Muc_cao PORTDbits.RD13
```

```
#define Muc_thap PORTDbits.RD7
```

```

#define Cap_nuoc      LATAbits.LATA0
#define Xa_phong      LATAbits.LATA1
#define Nhiet_do      LATAbits.LATA2
#define Khoa          LATAbits.LATA3
#define DC_Chay       LATAbits.LATA4
#define DC_chieu      LATAbits.LATA5
#define DC_tocdo      LATAbits.LATA6
#define Xa_nuoc       LATAbits.LATA7

```

```
#define TRIS_LED3  TRISAbits.TRISA0
```

```
#define TRIS_LED4  TRISAbits.TRISA1
```

```
#define TRIS_LED5  TRISAbits.TRISA2
```

```
#define TRIS_LED6  TRISAbits.TRISA3
```

```
#define TRIS_LED7  TRISAbits.TRISA4
```

```
#define TRIS_LED8  TRISAbits.TRISA5
```

```
#define TRIS_LED9  TRISAbits.TRISA6
```

```
#define TRIS_LED10 TRISAbits.TRISA7
```

```
#define TRUE 1
```

```
#define FALSE 0
```

```
char Co_xaphong,Tocdo_cao,Chieu_thuan,Co_nhiet,test;
```

```

float s;
unsigned char Thoigian_giat,st;

typedef enum{IDE,CAP_NUOC,DUN_NUOC,DONGCO,XANUOC} eControl_State;

eControl_State Mode;
int count_i,count;
int count_s=0,count_m;

_CONFIG1( JTAGEN_OFF & GCP_OFF & GWRP_OFF & COE_OFF & FWDTEN_OFF & ICS_PGx2)
_CONFIG2( FCKSM_CSDCMD & OSCIOFNC_OFF & POSCMOD_XT & FNOSC_PRI)

void TimerInit(void)// CAI DAT THAM SO CHO TIME 1 HOAT DONG
{
    PRI = 5000;// giá trị so sánh với tín hiệu đầu vào

    IPC0bits.TIIP = 5; //mức độ ưu tiên của IPCO ở mức 5
    T1CON = 0b10000000000010000;//T1con ở vị trí thu 5 của thanh ghi có giá trị 1, bit 5=1 thì hệ số
chia 1:8
    IFS0bits.TIIF = 0;// có ngắt bằng không
    IEC0bits.TIIE= 1; //cho phép ngắt khi có sự kiện, thì hàm sau được gọi
}

void __attribute__((interrupt, shadow, auto_psv)) _T1Interrupt(void) // 50Hz =20ms
{
    // Clear flag
    IFS0bits.TIIF = 0;
    count++;
    if(count==100) //1s
    {
        count=0;
        count_s++;
    }
    count_i++;
    if(count_i==6000) //60s
    {
        count_i=0;
    }
}

```

```

        count_m++;
    }
}

void LEDInit(void)// khoi tao cong ra cho cac Led
{
    TRIS_LED3 = 0;
    TRIS_LED4 = 0;
    TRIS_LED5 = 0;
    TRIS_LED6 = 0;
    TRIS_LED7 = 0;
    TRIS_LED8 = 0;
    TRIS_LED9 = 0;
    TRIS_LED10 = 0;
}

void Maygiat_Init(const eControl_State START_STATE)
{
    Mode=START_STATE;
}

void Kiemtra_Chedo(void)
{
    Co_xaphong=TRUE;
    Co_nhiet=TRUE;
    Tocdo_cao=TRUE;
    Thoigian_giat=60;
}

void MAYGIAT_Update(void)
{
    switch(Mode)
    {
        case IDE:
            Cap_nuoc=0;
            Xa_phong=0;
            Nhiet_do=0;

```

```

        Khoa =0;
        DC_Chay=0;
        DC_chieu=0;
        DC_tocdo=0;
        Xa_nuoc=0;
        Kiemtra_Chedo();

        if(Start==0) //Dieu khien chuyen che do
            Mode=CAP_NUOC;

        break;
        case CAP_NUOC:
            Cap_nuoc=1;
            Khoa=1;
//            if(Co_xaphong)
                Xa_phong=1;
            if(Muc_cao==0)
            {
                Cap_nuoc=0;
                Mode=DUN_NUOC;
            }
            break;
            case DUN_NUOC:
                Nhiet_do=1;
//                if(Donhiet)
                    {
                        Nhiet_do=0;
                        Mode=DONGCO;
                    }
                count_s=0;
                count_m=0;
            }
            break;
            case DONGCO:
                DC_Chay=1;
//                if(Donhiet)
                    Nhiet_do=0;
//                else
//                Nhiet_do=1;

```



```

        if(Chieu_thuan)
DC_chieu=0;
        else
            DC_chieu=1;
        if(Tocdo_cao)
            DC_tocdo=1;
        else
            DC_tocdo=0;
        if(count_s==10)
        {
            count_s=0;
            Chieu_thuan=~Chieu_thuan;
        }

if(count_m==Thoigian_giat)
{
    count_m=0;
    DC_Chay=0;
    DC_chieu=0;
    DC_tocdo=0;
    Mode=XANUOC;
}
break;
case XANUOC:
Xa_nuoc=1;
    if(Muc_thap==0)
    {
        Xa_nuoc=0;
        Mode=IDE;
    }
    break;
} //swith
} //Update

int main ()
{
    LEDInit();

```

```
    TimerInit();  
Maygiat_Init(IDE);  
  
    while (1)  
    {  
        MAYGIAT_Updata();  
    }  
}
```