

基于 PIC 单片机的 微机原理与应用实验指导书

徐 静 编

华南理工大学机械工程学院

机械工程专业实验室

2021 年 9 月

目 录

第一章	MPLAB X IDE 软件使用注意事项和调试器简介.....	2
第二章	软件和实验箱的使用简介	4
一、	软件的安装和介绍.....	4
二、	实验箱简介.....	19
第三章	PIC16F877A 软硬件基础实验	22
实验一	PIC 单片机软件的应用	22
实验二	外部中断实验.....	27
实验三	定时器实验.....	29
实验四	计数器实验.....	31
实验五	PWM 控制 LED 实验	33
实验六	数码管驱动显示实验.....	35
实验七	内部模数转换实验.....	40
实验八	矩阵键盘扫描显示实验.....	42
第四章	PIC16F877A 创新综合实验	44
实验一	步进电机控制实验.....	44
实验二	LCD1602 显示实验（IO 方式驱动）	46
实验三	16X16 点阵汉字显示实验	50

第一章 MPLAB X IDE 软件使用注意事项和调试器简介

1. MPLAB X IDE 软件不支持中文路径，所有项目名、文件名以及系统路径名均使用英文和数字来命名。如果要加中文注释，也要按照下面软件的操作说明在建立项目时，选择 GB2312 的语言格式。这样编辑的中文不会乱码。
2. 在进行模拟仿真联调时，要注意正确加载电路图的文件或者在 PROTEUS 里加载正确的编译过的机器语言程序。（编译的程序和电路图文件要在一个文件夹里）
3. 实验系统软件全面支持 PIC 8 位到 32 位系列单片机的实验编程及调试的开发过程。可以通过与 PROTEUS 软件的联调实现多种单片机仿真实验。
4. 实验系统因使用了在线调试功能，使得 PIC16F877A 芯片 PB 口的 PB3、PB6 和 PB7 被占用。
5. PICkit4 的功能介绍



PICkit 4 调试器

用户可以通过 MPLAB X IDE 集成开发环境（Integrated Development Environment, IDE）功能强大的图形用户界面，使用 MPLAB PICkit 4 在线调试器（PG164140）快速方便地调试和编程 Microchip PIC®、dsPIC® 闪存和 CEC（基于 Arm® Cortex®-M4）单片机。MPLAB PICkit 4 还可用作器件生产编程器。

MPLAB PICkit 4 使用高速 2.0 USB 接口连接到计算机，可通过 Microchip8 引脚单列直插式（Single In-Line, SIL）调试连接器连接到目标板。连接器使用两个器件 I/O 引脚和复位线来实现在线调试和在线串行编程（In-Circuit Serial Programming™, ICSP™）。附加 micro SD 卡插槽和通过目标板自供电的能力，意味着可以随身“携带”代码并脱机编程。

由于该调试器系统使用带有内置仿真电路的器件（而非特殊调试器芯片），因此代码的执行与实际器件相似。可以交互访问给定器件的所有可用功能，且可通过 MPLAB X IDE 界面设置和修改这些功能。

MPLAB PICkit 4 在线调试器兼容以下任意平台：

- Microsoft Windows® 7 或更高版本
- Linux®
- macOS™

要使用 MPLAB PICKit 4 在线调试器系统进行调试（设置断点和查看寄存器等），以下关键因素必须正确：

- 调试器必须连接到计算机。它必须由计算机通过 USB 电缆供电，并且必须通过 Micro-B USB 电缆与 MPLAB X IDE 软件通信。

- 调试器必须通过模块化接口电缆（或等效电缆）连接到目标器件的 Vpp、PGC 和 PGD 引脚。

- 目标器件必须通电且具有正常运行的振荡器。如果目标器件出于任何原因未运行，MPLAB PICKit 4 在线调试器都无法调试。

- 目标器件必须正确编程其配置字。这些配置字使用 MPLAB X IDE 来设置。

- 振荡器配置位应对应于 RC 和 HS 等，具体取决于目标设计。
- 对于某些器件，看门狗定时器默认使能，需要禁止。
- 不要使能目标器件的代码保护功能。
- 不要使能目标器件的表读保护功能。
- 要关闭目标器件的低电压编程功能。

第二章 软件和实验箱的使用简介

一、 软件的安装和介绍

(一)、MPLAB X IDE 软件安装

1. 可以在官网

<https://www.microchip.com/development-tools/pic-and-dspic-downloads-archive> 网页下载 MPLAB X-v5.00 版本的软件；

2. 然后双击下载的文件 MPLABX-v5.00-windows-installer.exe 按照提示进行安装，直到安装完成；

在安装过程中，如果用户没有指定安装目录，安装完成后，会在 C: 盘建立一个目录，结构如下：

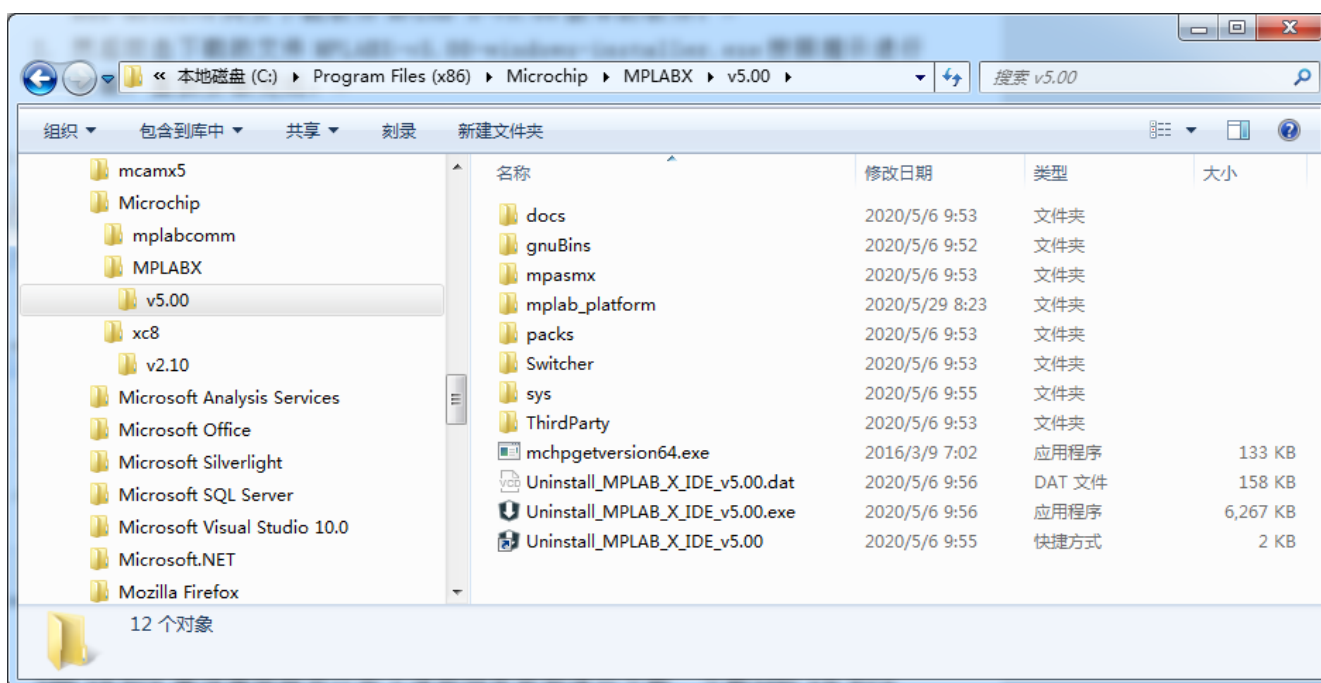


图 2-1 MPLAB X IDE 5.0 目录

(二)、编译器安装

MPLAB XC8 编译器同样可以在上述的网页找到进行下载，下载MPLAB XC8 V2.1版本的编译软件。该软件支持PIC8位系列的单片机。它与Microchip公司的MPLAB X IDE兼容，由于它是作为一个插件使用，不会在桌面单独出现一个图标。MPLAB XC8编译器默认安装路径如下图 2-2所示。

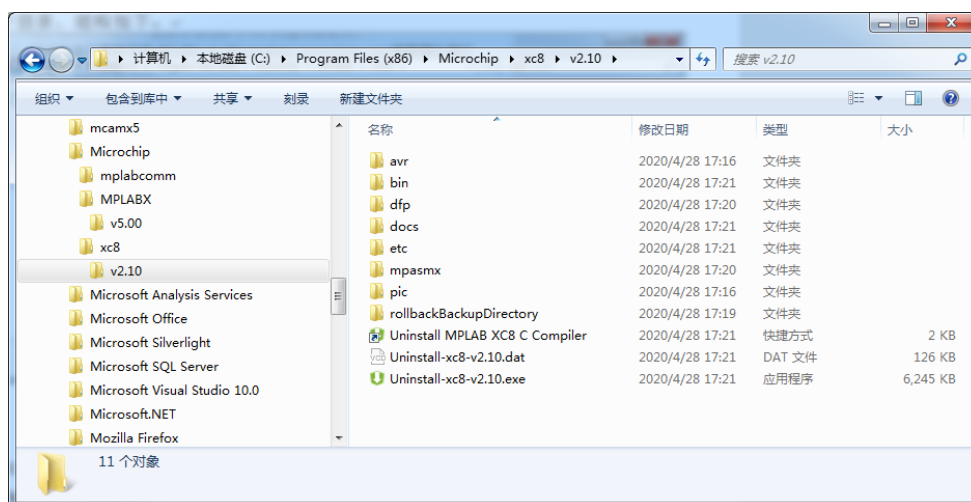


图 2-2 MPLAB XC8 目录

(三)、开发环境简介

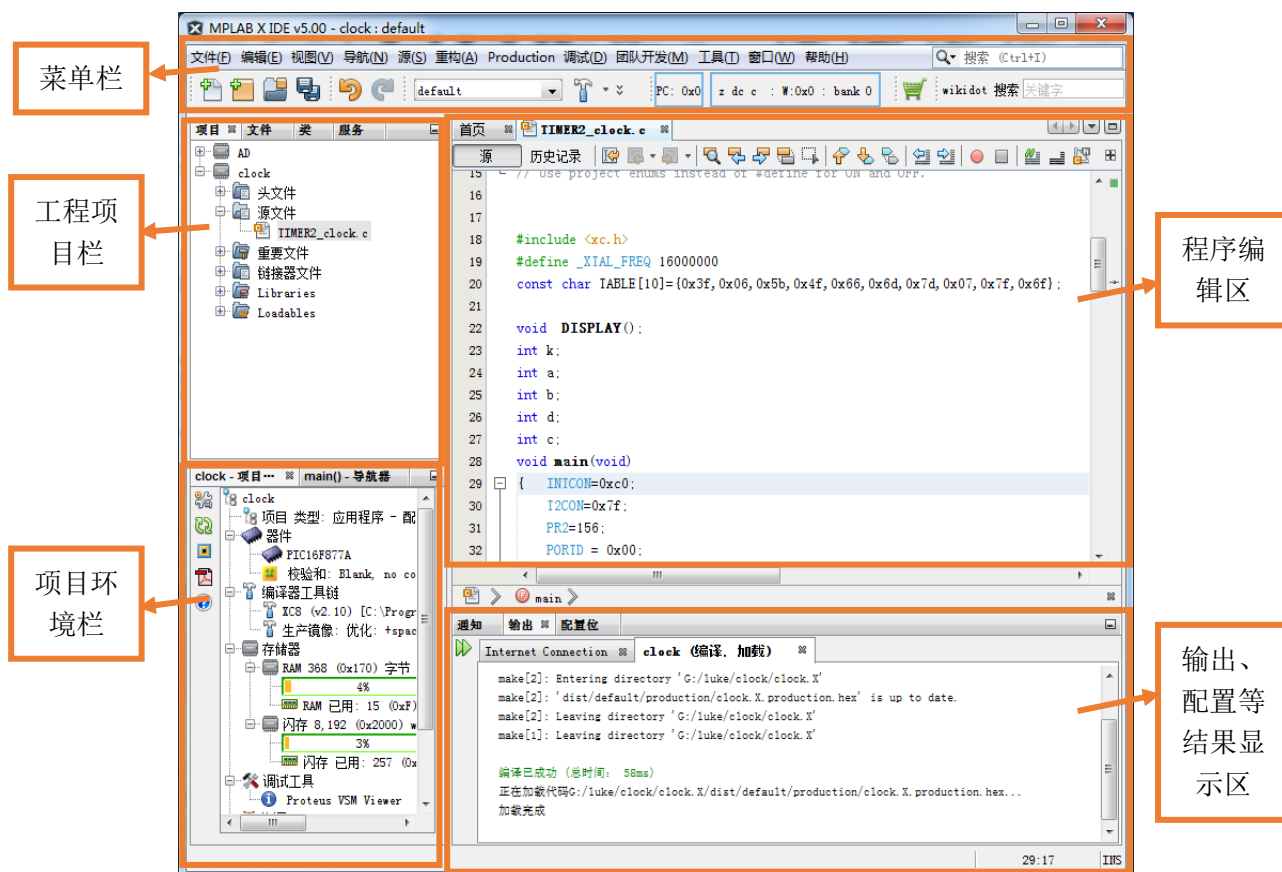


图 2-3 MPLAB X IDE 界面

下面分别介绍图 2-3 MPLAB X IDE 软件界面常用几个菜单的功能

1. 文件 (F)

文件菜单如图 2-4 所示，大部分命令同 WINDOWS 的其它软件类似命令使用相同。其中新建项目是用来创建新的项目文件，新建文件是用来创建一个汇编或者 C 语言的新文件。Import 可以用来导入早期版本或者其它格式的文件或项目。

2. Production

在该菜单如图 2-5 所示可以对项目进行编译，设置主项目。还有对硬件的配置位进行设置-Set Configuration Bits。如图 2-6 所示配置位设置。

3. 调试(D)

调试菜单如图 2-7 所示，要在运行了调试项目才能有调试步骤的选项可选，可以进行单步、断点调试等，可以观察全局变量或专用寄存器的内容。

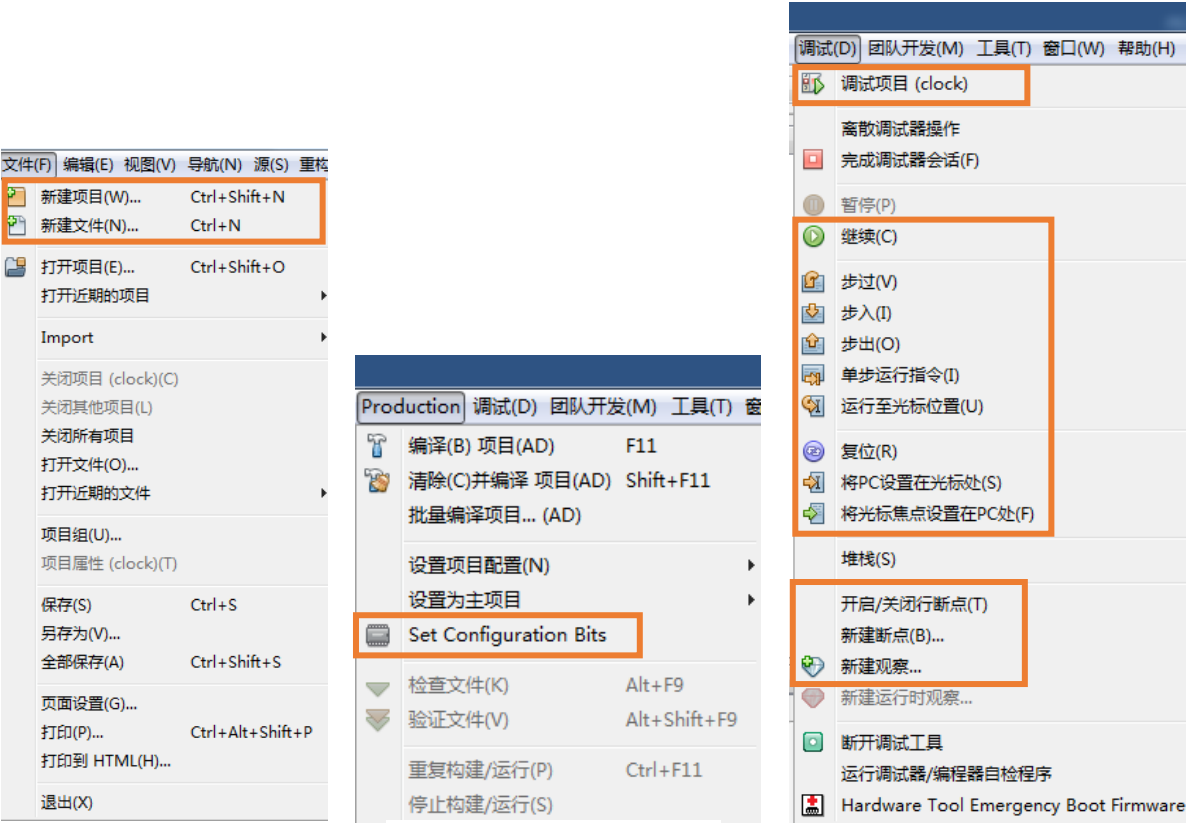


图 2-4 文件菜单

图 2-5 Production 菜单

图 2-7 调试菜单

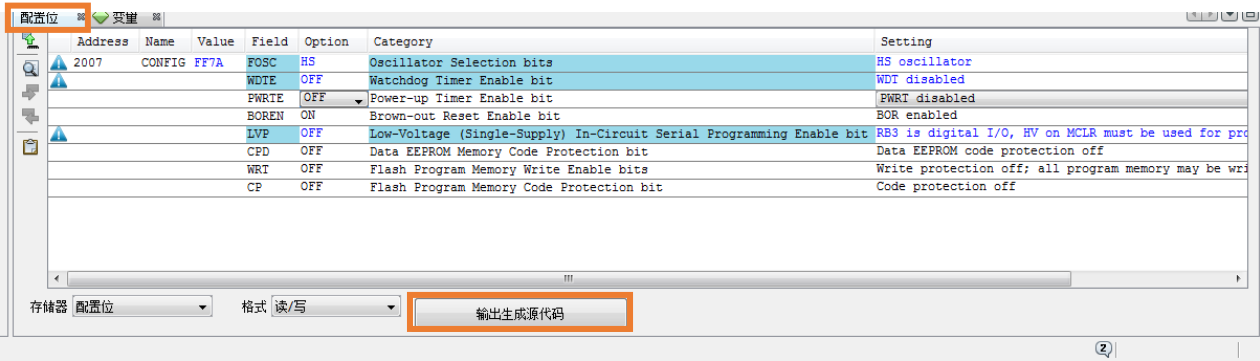


图 2-6 配置位设置

4. 工具(T)

工具菜单可以进行**插件的安装**，比如进行虚拟仿真用到的 Proteus VSM Viewer, 当然必须要确认使用的计算机网络是通畅的。在选项对话框还可以进行一些字体、颜色的设置，在**嵌入式选项**还可以进行**编译工具的设置**，一般在**通用设置**里设置 Debug Reset 为 Main 开始，Debug Startup 为 Halt at Main, 这样程序运行时先停在开始处，然后进行单步、断点或者连续运行调试程序。

5. 窗口(W)

在窗口菜单可以打开或关闭各种窗口信息，比如：项目、文件、项目环境、编辑器等内容。

(四)、MPLAB X IDE 软件的使用步骤

1. 在 PC 机上用鼠标双击 MPLAB X IDE 图标，进入 WINDOWS 调试环境。
2. 通过“文件”菜单，点击“新建项目”来进行新项目的建立，以下是新建项目的步骤，如图 2-8 系列所示。

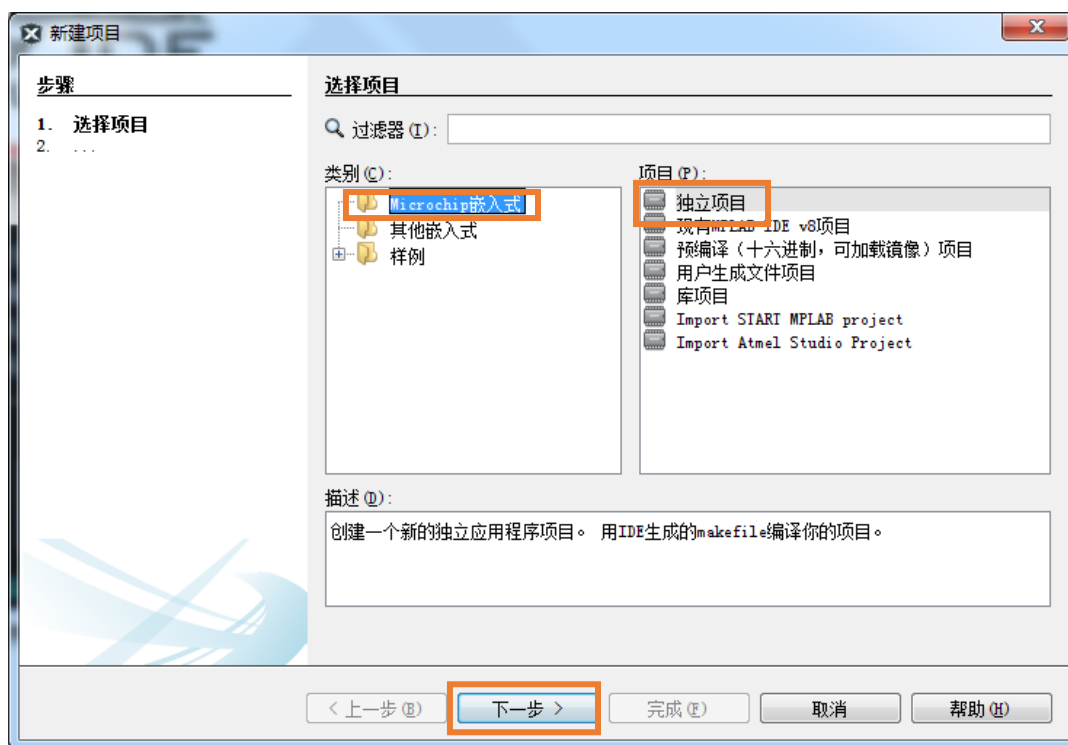


图 2-8-1 新建项目导向一



图 2-8-2 新建项目导向二

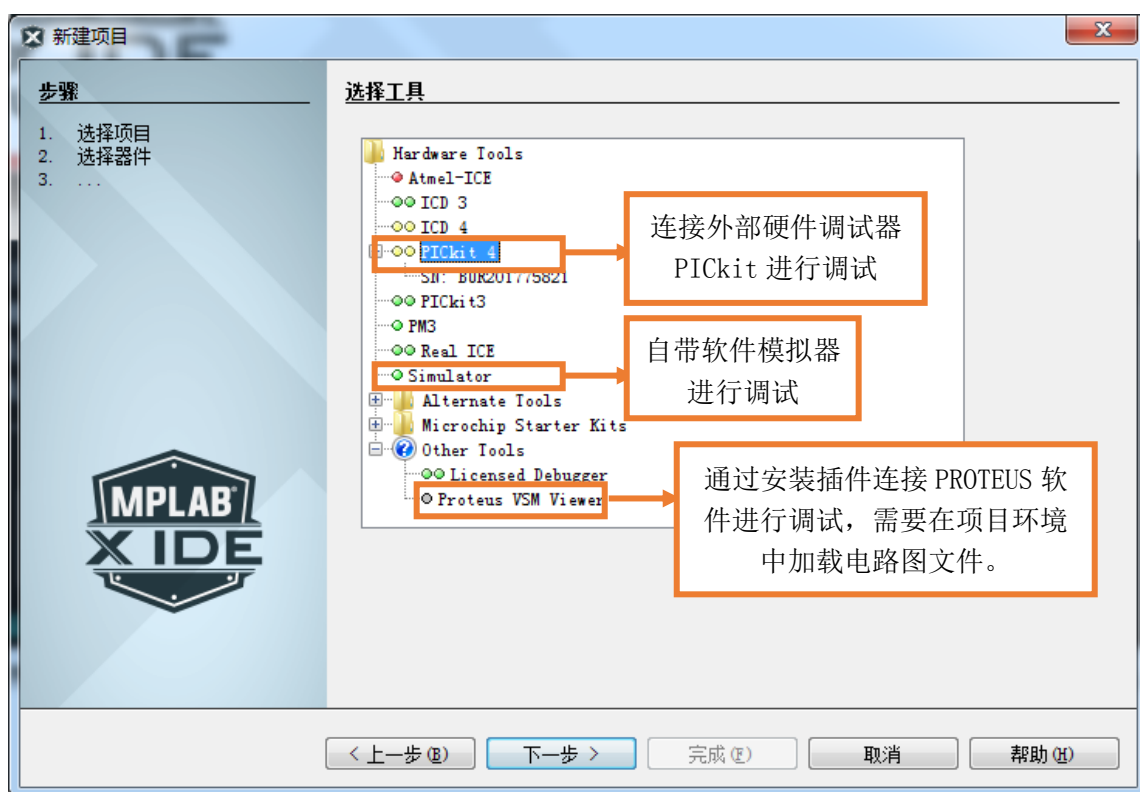


图 2-8-3 新建项目导向三

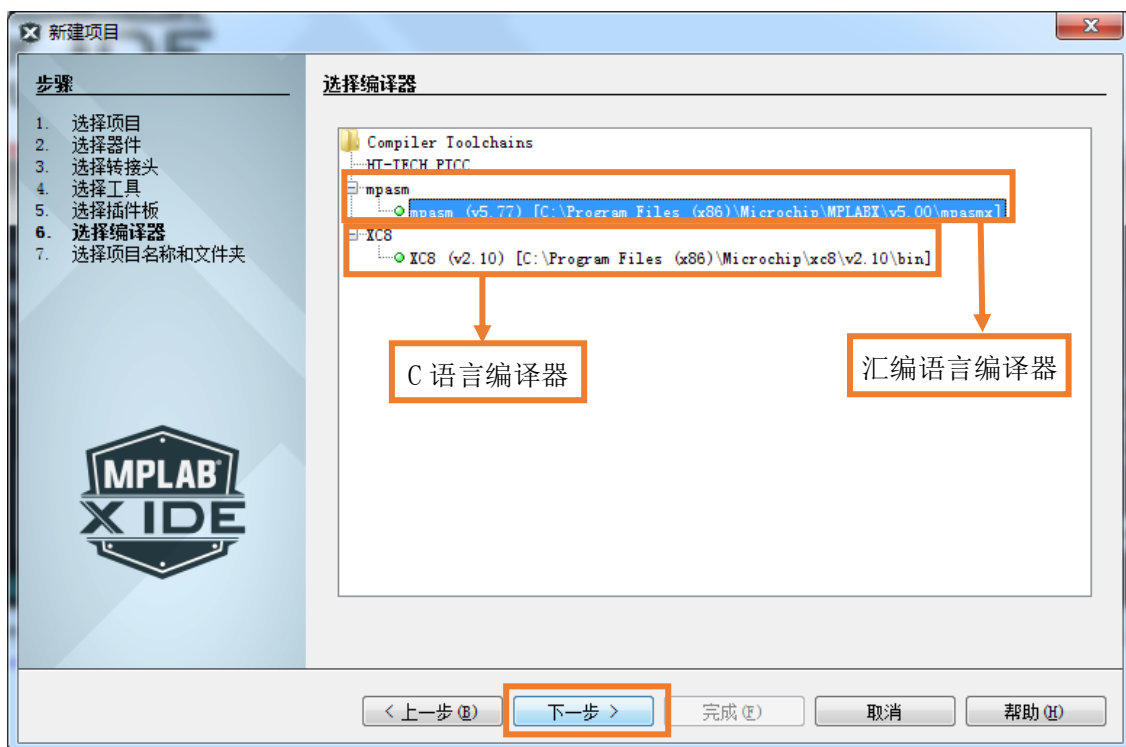


图 2-8-4 新建项目导向四



图 2-8-5 新建项目导向五

3. 通过“文件”菜单，点击“新建文件”来进行新文件的建立。如下图 2-9-1 到 2-9-3 所示，也可以在新建的项目里，通过源文件点击右键选择要建的项目，如图 2-9-4 所示。

1) 汇编文件的新建

选择 Microchip Embedded→MPASM assembler→pic_8b_simple.asm，如果选择 pic_8b_general.asm 文件，将会给出 PIC16 系列和 PIC18 系列的中断矢量入口程序。

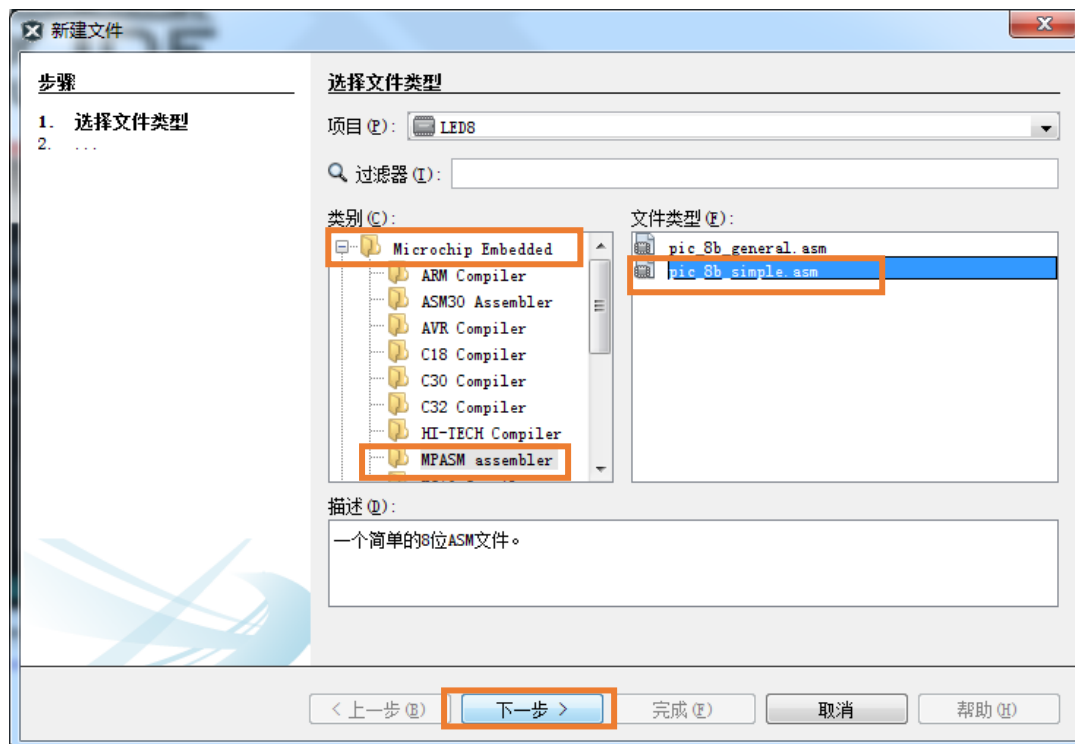


图 2-9-1 新建汇编文件导向一



图 2-9-2 新建汇编文件导向二

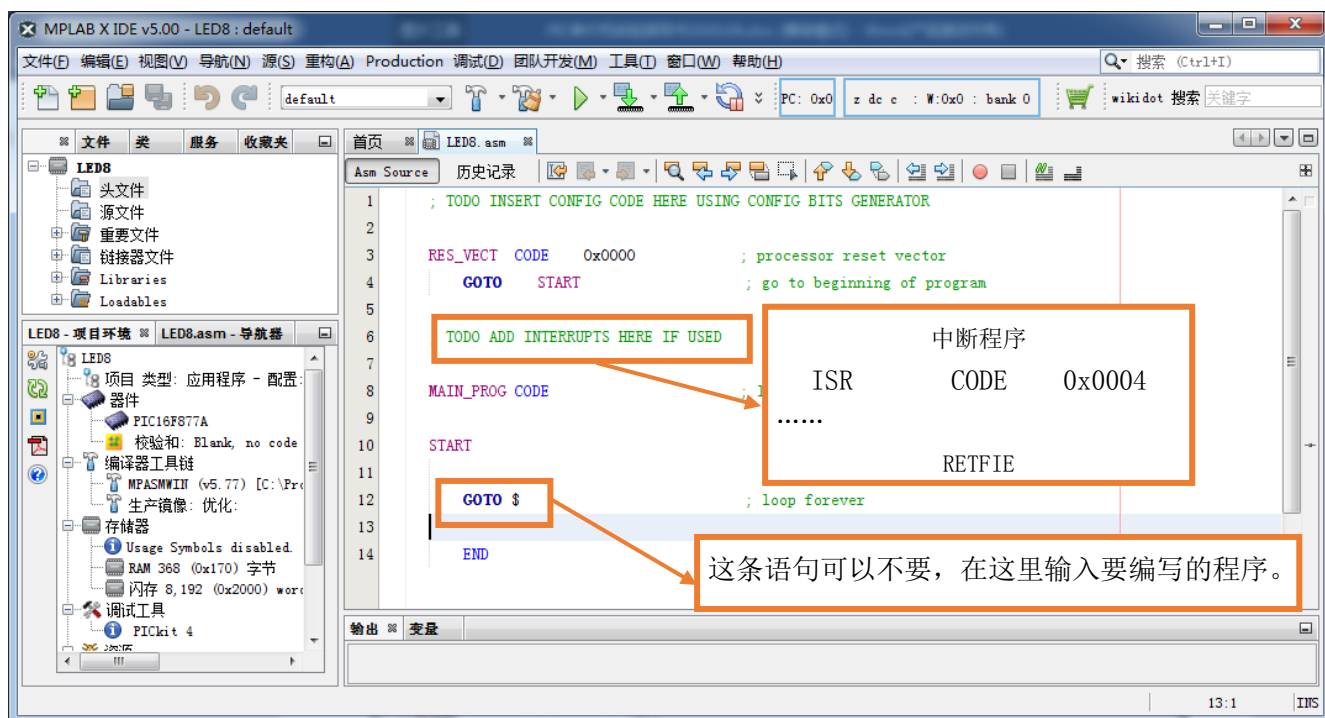


图 2-9-3 新建汇编文件导向三

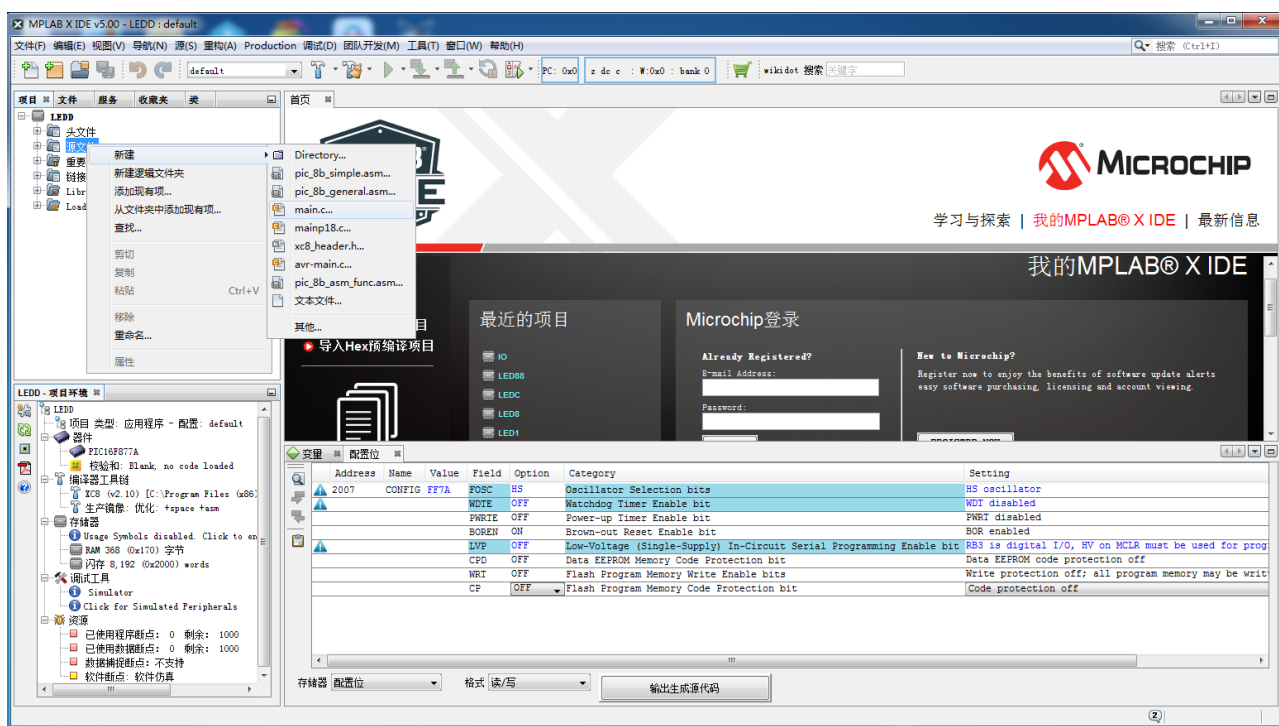


图 2-9-4 新建汇编文件导向四

2) 配置位的设置

按照前面图 2-5 和图 2-6 的操作说明设置配置位，步骤如下：①晶振选择高速晶振 HS，看门狗 WDT disables，低电压编程 LVP OFF，其余不用改变。②点击左下角配置位选择配置

位，就保存了配置位的设置，③点击输出生成源代码，④点击上方配置位选项，⑤把光标移到编辑窗口程序的最上方，然后点击插入源代码到编辑窗口。如图 2-10 所示。编辑完成后程序如图 2-11 所示。

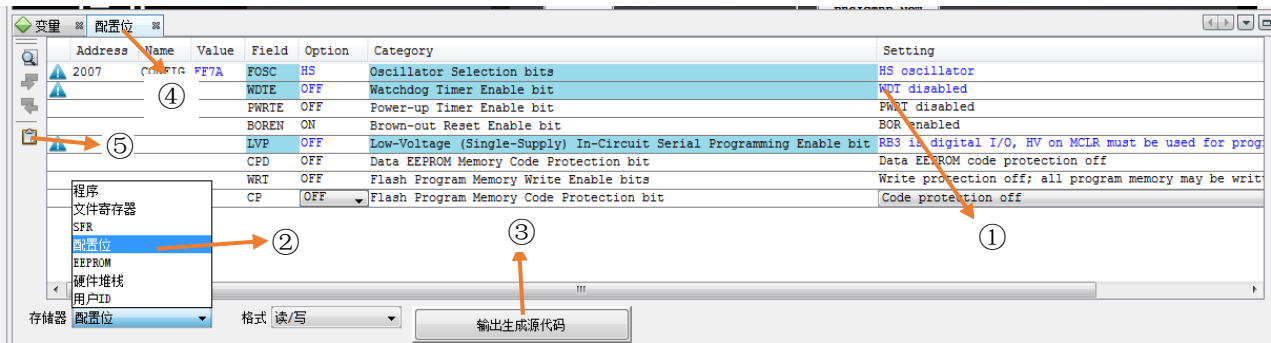


图 2-10 配置位的设置步骤

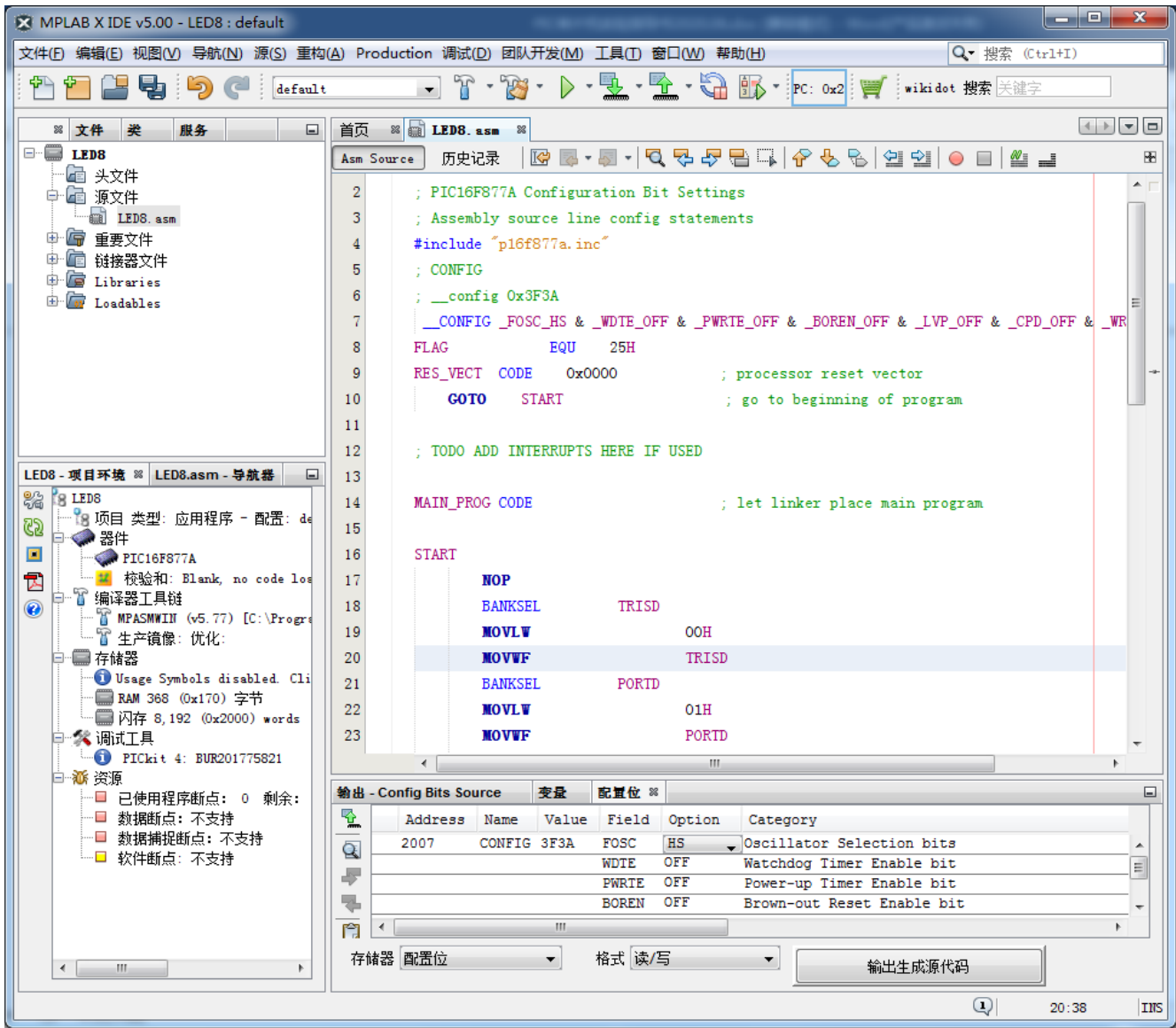


图 2-11 汇编语言编辑完成的界面

3) 调试项目

按照如图 2-12 的步骤进行主项目的编译调试。汇编语言在编程后，目标板程序就开始独立运行，要停止运行需要按复位按钮，也可以用 simulator 调试器进行单步、断点方式调试。C 语言在编译主项目后，可以在线进行调试主项目，也可以编程后独立运行程序。

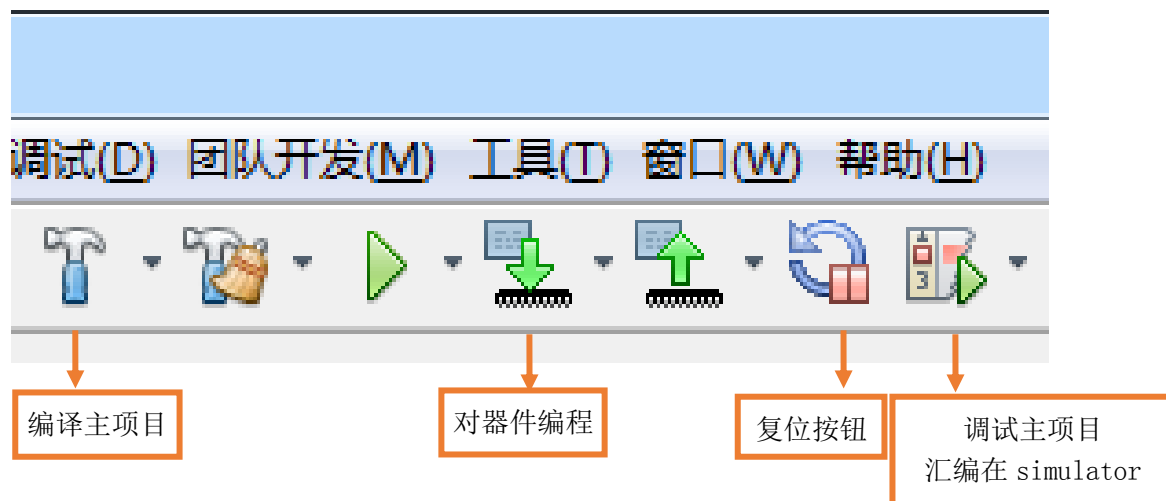


图 2-12 编译调试项目步骤

4) C 语言文件的新建

首先要新建项目，步骤如图 2-8 导向图，只是在后面选择编译器时要选择 XC8 的编译器，其它步骤一样。

新建 C 语言文件，通过文件菜单→新建文件，选择 Microchip Embedded→XC8 Compiler→main.c，只是在后面选择编译器时要选择 XC8 的编译器，其它步骤一样，如图 2-13 所示，也可以通过如图 2-9-4 所示通过源文件新建 main.c 的文件，完成结果如图 2-14 所示。

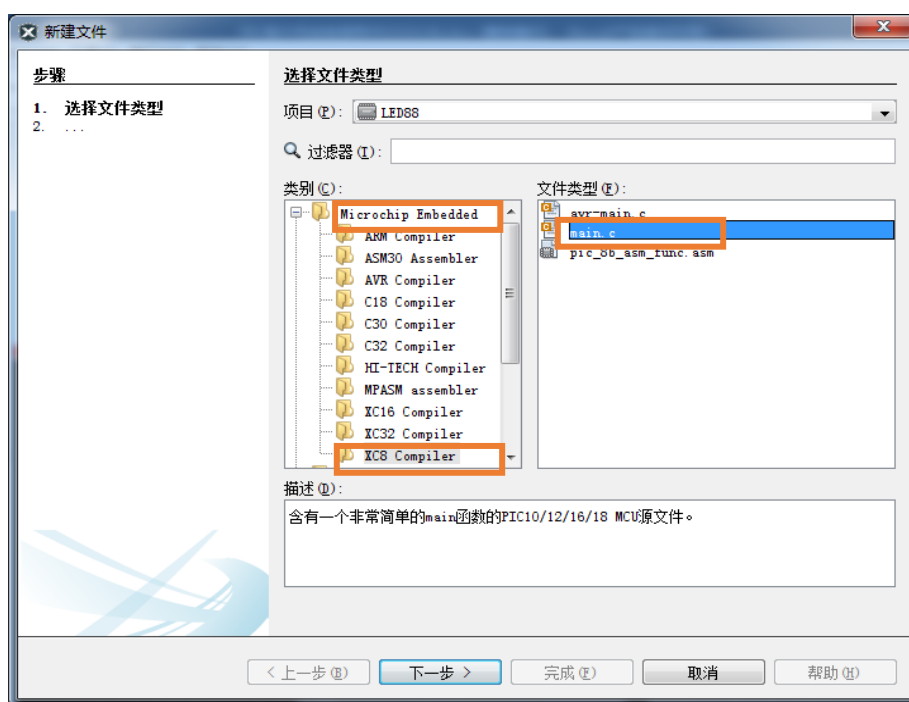


图 2-13 新建 C 语言文件步骤

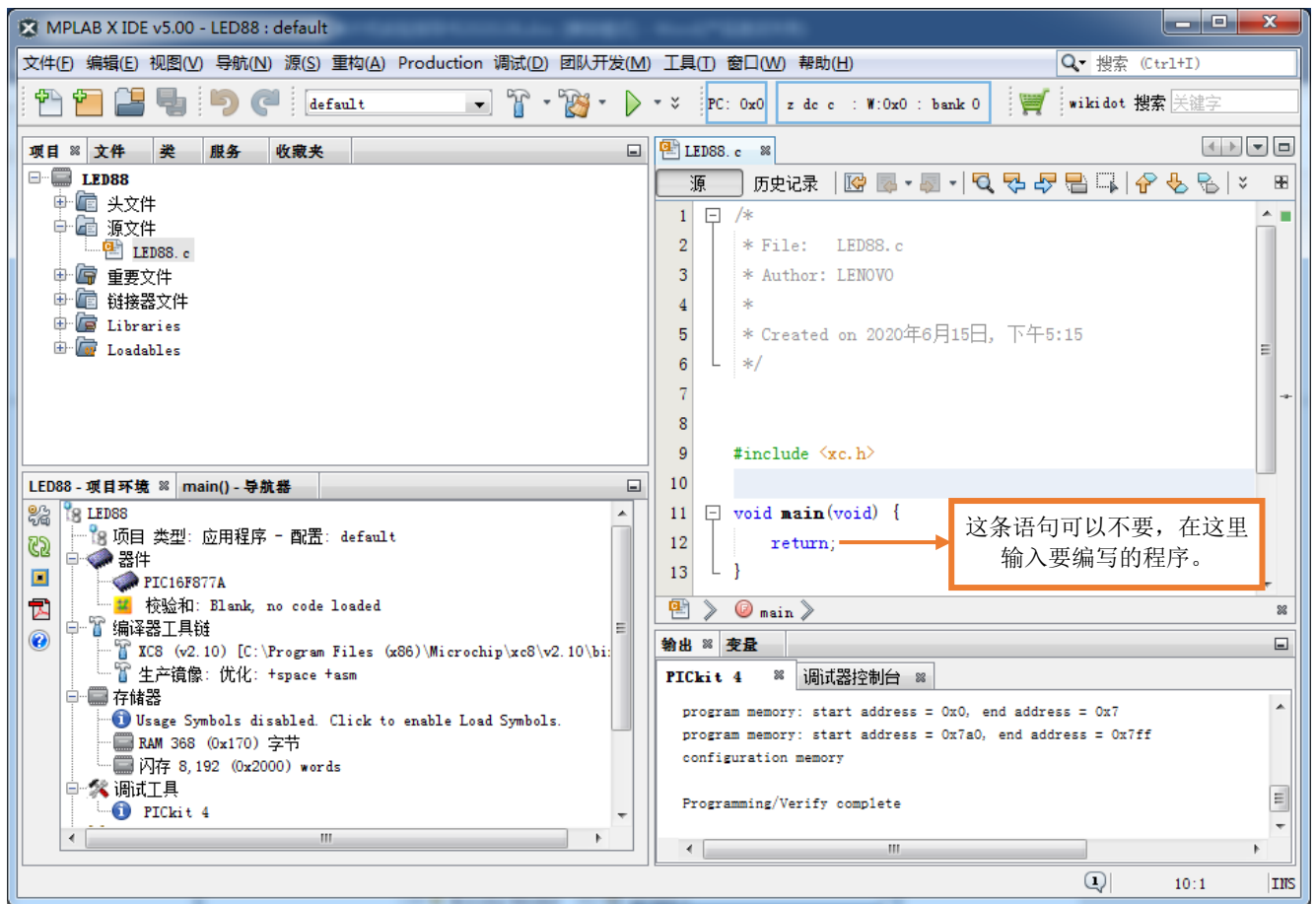


图 2-14 新建 C 语言文件界面

输入程序，然后设置配置位，步骤如图 2-10 所示。然后进行编译程序、对目标板编程、调试程序，如图 2-12 所示。打开调试菜单，找到“新建观察”，选择 SFR 后，把需要观察的变量选中，如图 2-15 所示，然后确定。执行了调试主项目后，会出现调试菜单项，如图 2-16 所示，可以对该程序进行单步运行、断点运行、连续运行等方式进行调试，观察变量值的变化。

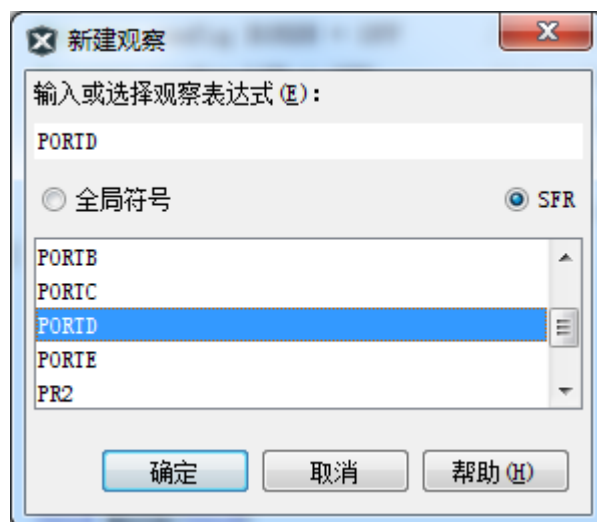


图 2-15 新建观察

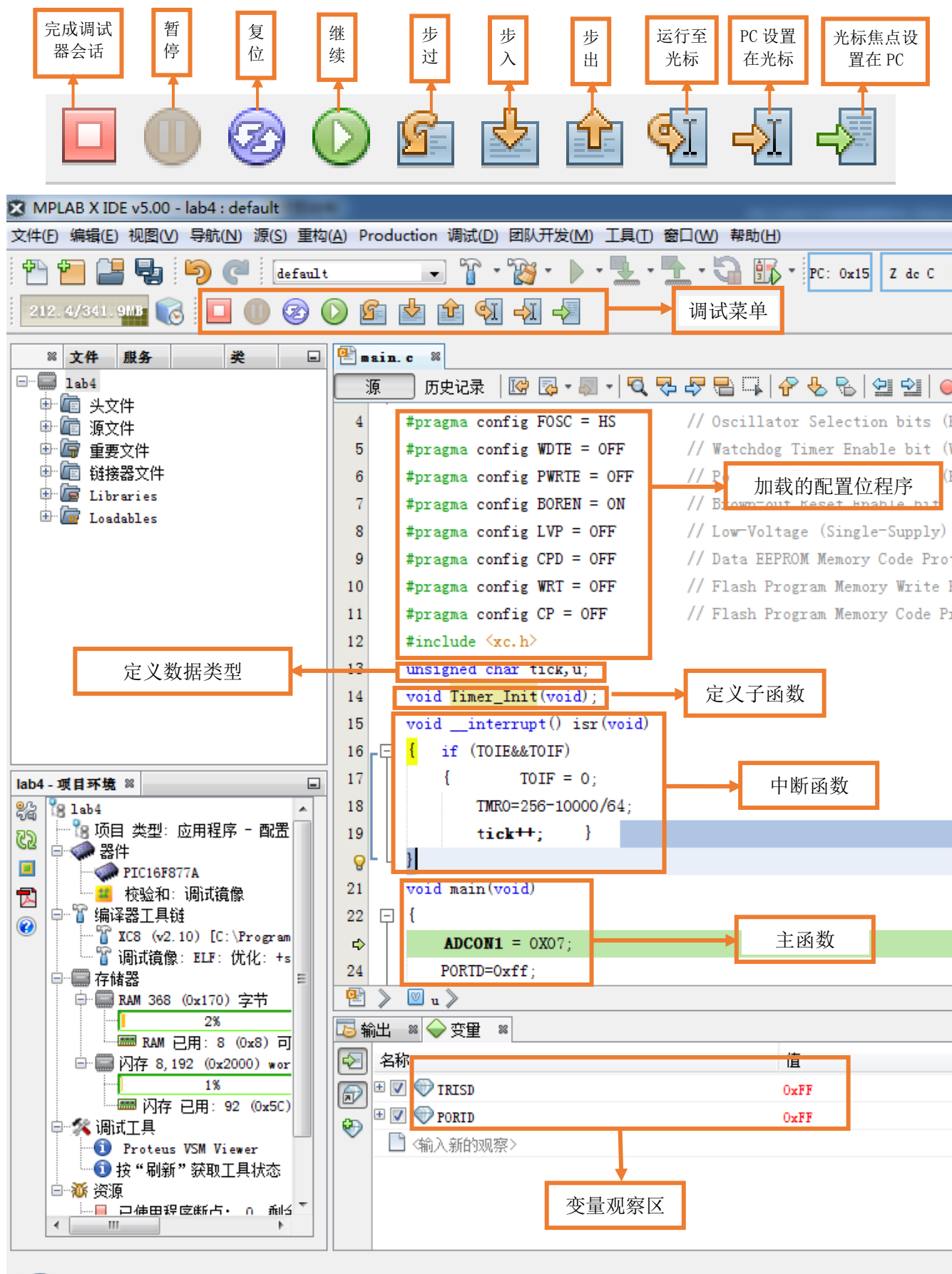


图 2-16 C 语言调试界面

(五)、Proteus 软件的使用步骤

1. 新建原理图

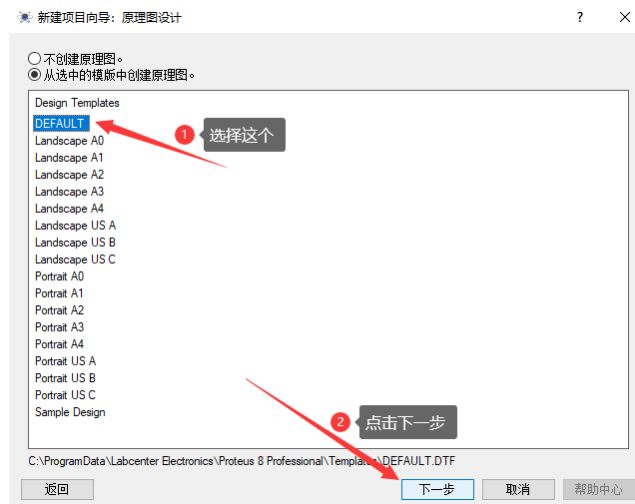
1) 打开Proteus 8，点击新建工程；



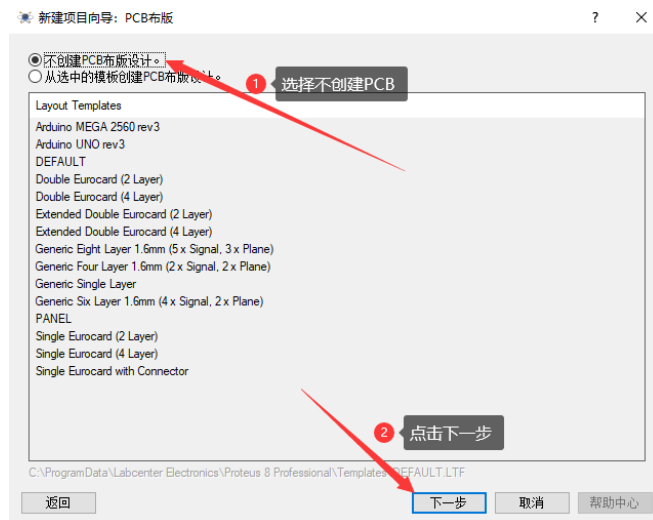
2) 工程命名，最好用英文或数字命名；



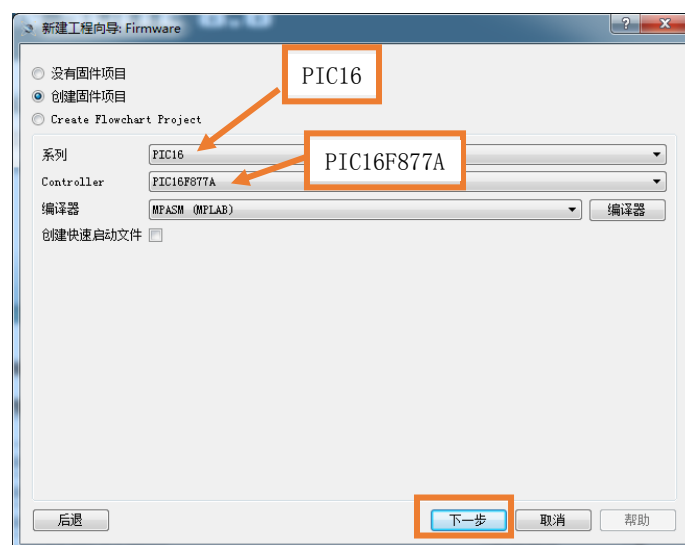
3) 创建原理图，选择DEFAULT，默认模板；



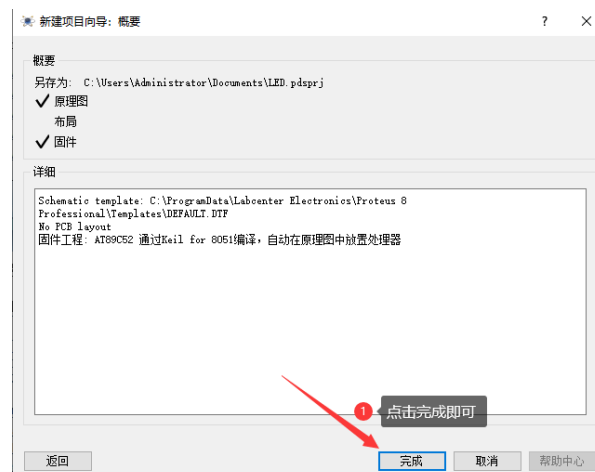
4) 选择不创建PCB设计；



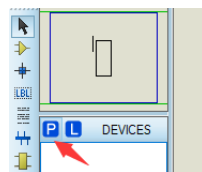
5) 创建固件项目，系列选择 PIC16，控制器选择 PIC16F877A，编译器选择默认；



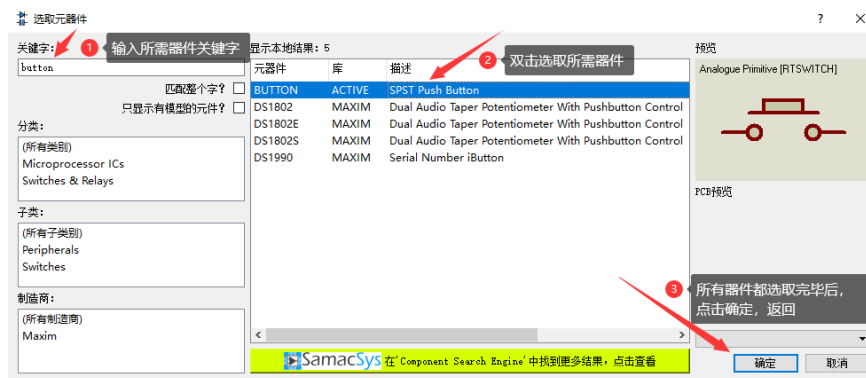
6) 完成原理图创建；



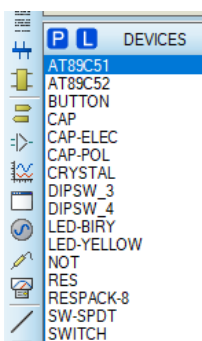
7) 在原理图设计界面，点击左侧蓝色的 P 图标进入元器件选取界面；



8) 在左上角输入所需元器件关键字，并在右侧列表双击选取所需器件，点击确定；



9) 选取完成后，左侧的器件栏窗口会有相应元器件列表；



10) 参考原理图设计步骤完成器件的排布和连线。

2. Proteus 中仿真

在处理器芯片 PIC16F877A 器件上，鼠标右键单击，选择编辑属性选项，或者双击 PIC16F877A 芯片，在 Program File 的浏览选项中添加 MPLAB 软件中生成的 HEX 文件。点击 OK，完成 HEX 文件添加。

点击原理图左下角的运行按键，原理图进入仿真状态，可以在原理图中观察到程序运行的结果。

二、实验箱简介

实验室使用的是广州风标教育技术股份有限公司生产的 FB-EDU-MCU-F 型微控制器仿真实验实训箱。下面介绍实验箱的各组成部分，如图 2-17 所示。

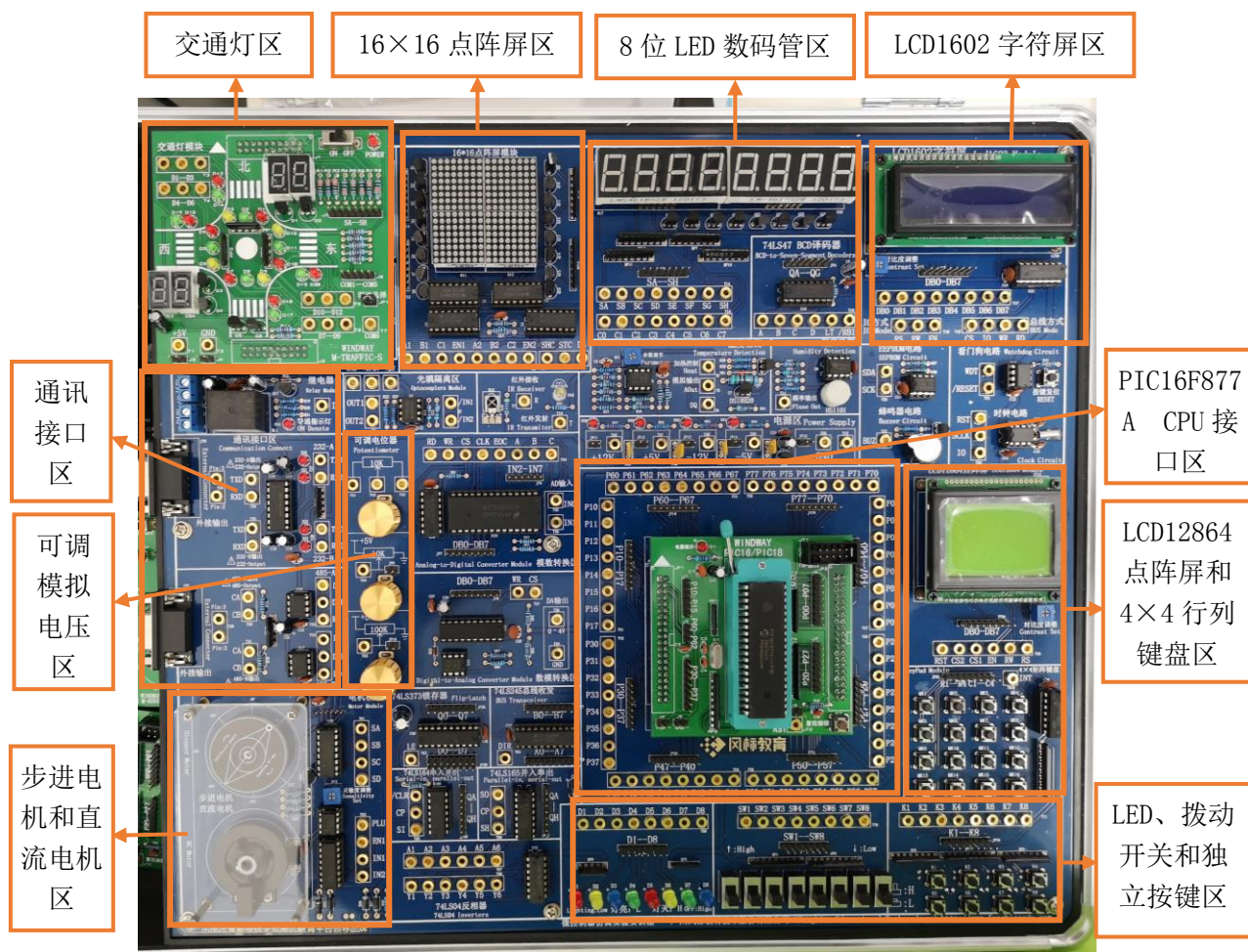


图 2-17 实验箱组成图

1. 实验电路接线

由于实验箱上每一部分电路都是独立的，在实验时，需要通过连接导线把相应的接口连接起来。连接导线有单根线也有排线，具体的使用需要根据接口组成来连接。

由于该实验箱适合多种 CPU 的实验，因此在做 PIC16F877A 为 CPU 实验时，首先要在 CPU 接口区确认插入的接口板是 **PIC16F877A 接口板**。实验时，通过 PICkit 仿真器与计算机连接起来，可以实现程序的下载、调试功能。实验室 PICkit 仿真器有三种，分别是 PICkit4、PICkit3 和 PICkit2，其中 PICkit4 与计算机连接是用透明的白线连接，其它两种是用黑线连接，接口不一样。三种仿真器与 CPU 板的连接都是用扁平电缆通过转接口连接，连接时一定要注意，转接口的三角符号与仿真器的三角符号要对准插入，仿真器上后面多余的口不用接，如图 2-18 所示。

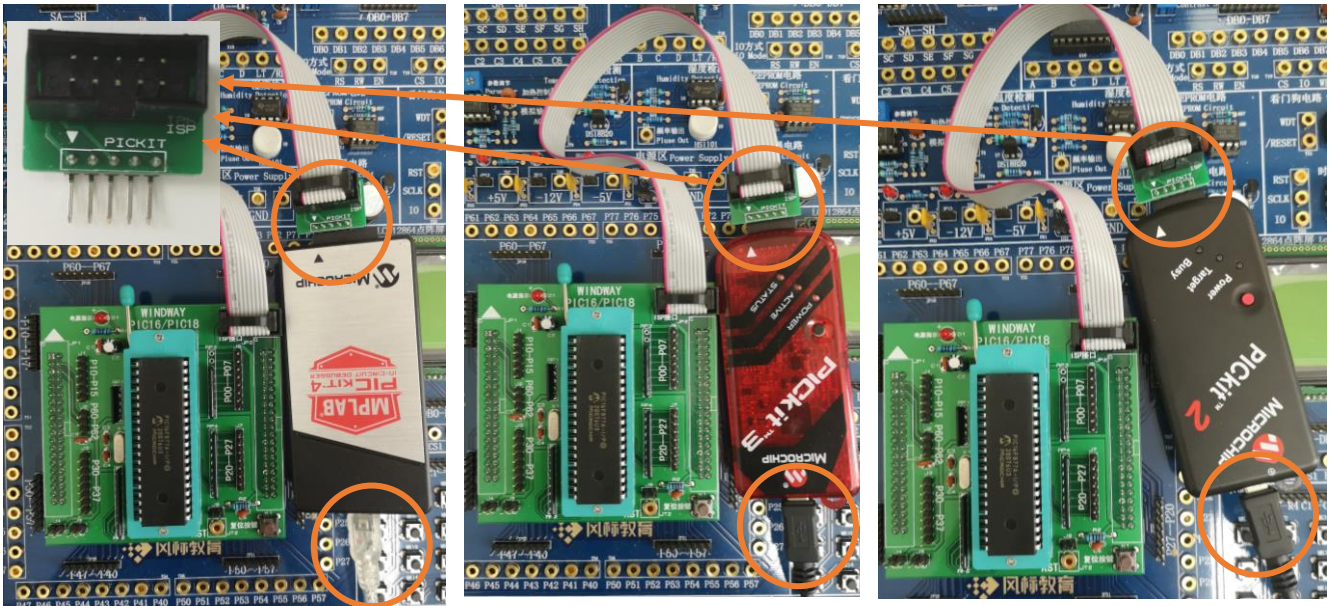


图 2-18 仿真器连接图

2. CPU 板接口介绍

CPU 板上的标识符与实验箱上锁紧插孔的标识符一一对应，连接时插 CPU 板和插实验箱的插孔都可以，根据连接方便选择。实验箱接口的标识符与 PIC16F877A 的接口对应关系见图 2-19 所示，。

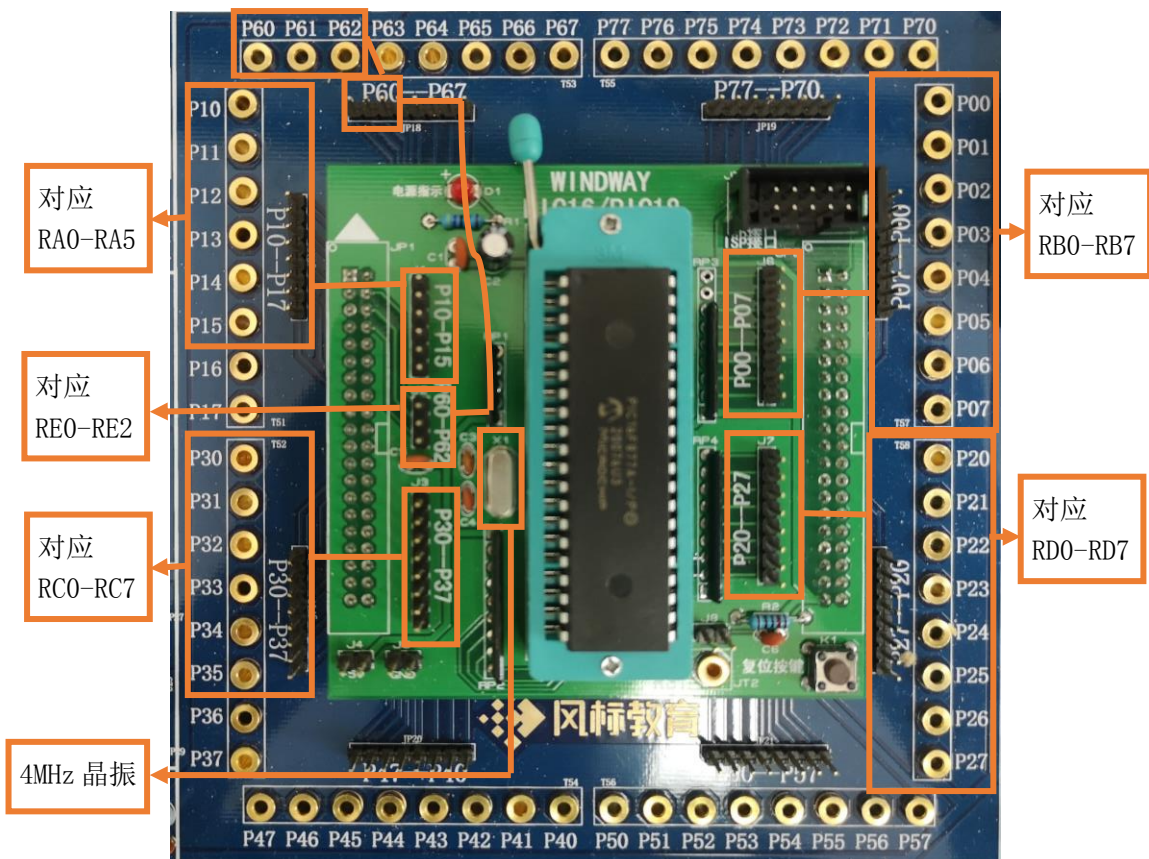


图 2-19 PIC16F877A 接口电路图

引脚对照表

底板丝印	PIC16F877A
P00~P07	RB0~RB7
P10~P15	RA0~RA5
P20~P27	RD0~RD7
P30~P37	RC0~RC7
P60~P62	RE0~RE2
其他控制脚	RST 、 GND、 +5V

第三章 PIC16F877A 软硬件基础实验

实验一 PIC 单片机软件的应用

一、实验目的

1. 熟悉 MPLAB X IDE 软件、XC8 软件、PROTEUS 软件的使用，掌握在 MPLAB X IDE 环境中编写程序、调试程序和 PROTEUS 软件中仿真运行程序的方法。
2. 掌握 I/O 口的编程设置、延时子程序的编写。
3. 掌握在 MPLAB X IDE 环境中如何观察变量及寄存器值的变化。

二、实验设备与器件

1. 计算机 1 台
2. 调试器 PICkit 1 个
3. 实验箱 1 台
4. 导线若干

三、程序清单（8 位 LED 灯流水控制）

汇编程序：

```
;-----  
FLAG      EQU      25H  
  
      NOP  
      BANKSEL      TRISD  
      MOVLW         00H  
      MOVWF         TRISD  
      BANKSEL      PORTD  
      MOVLW         01H  
      MOVWF         PORTD  
      BSF           FLAG, 0  
      BCF           STATUS, 0  
LOOP   BTFSS        STATUS, 0  
      GOTO          LOOP1  
      COMF          FLAG, 1  
LOOP1  BTFSS        FLAG, 0  
      GOTO          LOOP2  
      RLF           PORTD, 0  
      MOVWF         PORTD  
      GOTO          LOOP3  
LOOP2  RRF           PORTD, 0
```

```

                MOVWF      PORTD
LOOP3          CALL      DELAY
                GOTO      LOOP
;-----延时子程序-----
DELAY
                MOVLW      0FFH
                MOVWF      20H
LP0            MOVLW      0FFH
                MOVWF      21H
LP1            DECFSZ     21H, 1
                GOTO      LP1
                DECFSZ     20H, 1
                GOTO      LP0
                RETURN
;-----
                END

```

C 语言程序:

```

;-----
#define uint unsigned int
uint flag;
int i;
void delay(void);
void main(void)
{
    TRISD=0x00;
    PORTD=0x01;
    flag=0x00;
    while(1)
    {
        if (flag==0x00)
        {
            PORTD=(PORTD<<1);
            delay();
            if(PORTD==0x80)
            {flag=0x01;}
        }
    }
}

```



```

    }
else
{PORTD=(PORTD>>1);
  delay();
  if(PORTD==0x01)
    {flag=0x00;}
}
}

void delay(void)
{
  int i;
  for(i=10000;i>0;i--);
}

```

四、实验内容、要求和步骤

图 3-1-1 是本次实验的电路原理图,用 RD 口作为输出口连接 8 个 LED 发光管。用 PROTEUS 软件先进行仿真模拟, 然后按照图 3-1-2 的接线图, 用排线连接 P20-P27 与 D8-D1 (JP32), 下载程序到 CPU, 在实验箱上观察流水灯的运行。

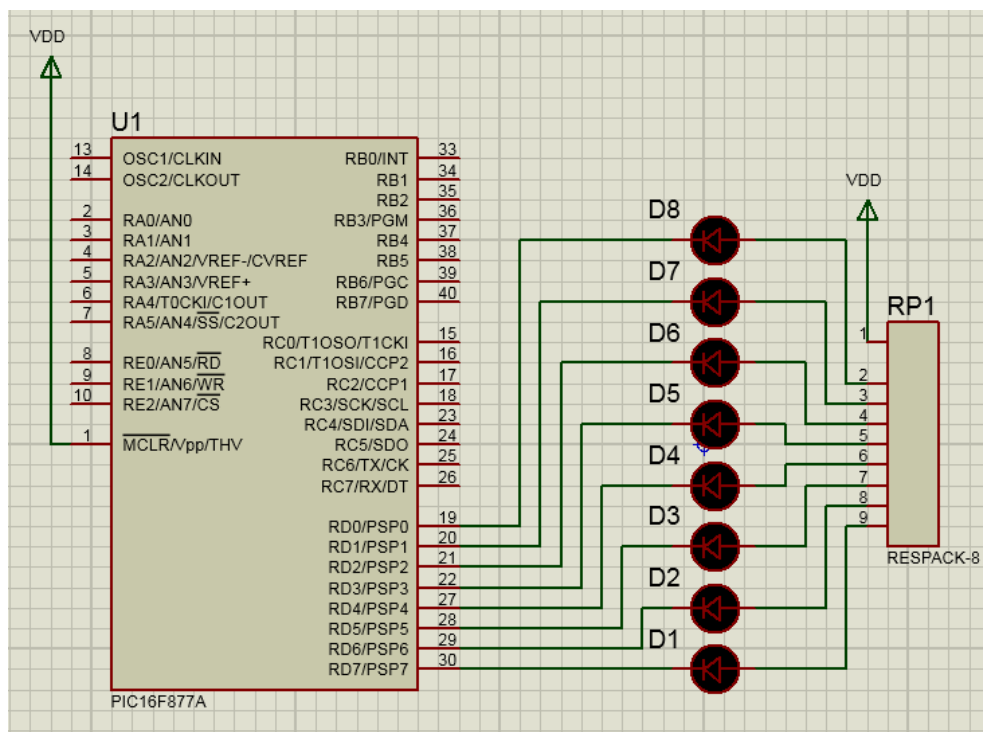


图 3-1-1 8 位 LED 流水灯实验原理图

实验要求: 利用单片机的一个 IO 端口及 8 个发光二极管等器件, 构成一个流水灯控制电路。

单片机最小系统由单片机芯片、时钟电路以及复位电路构成。延时子程序的延时计算，需要根据晶振时钟的大小以及指令周期，计算延时时间所要执行的指令数目，实验箱用到的晶振是 4MHz。

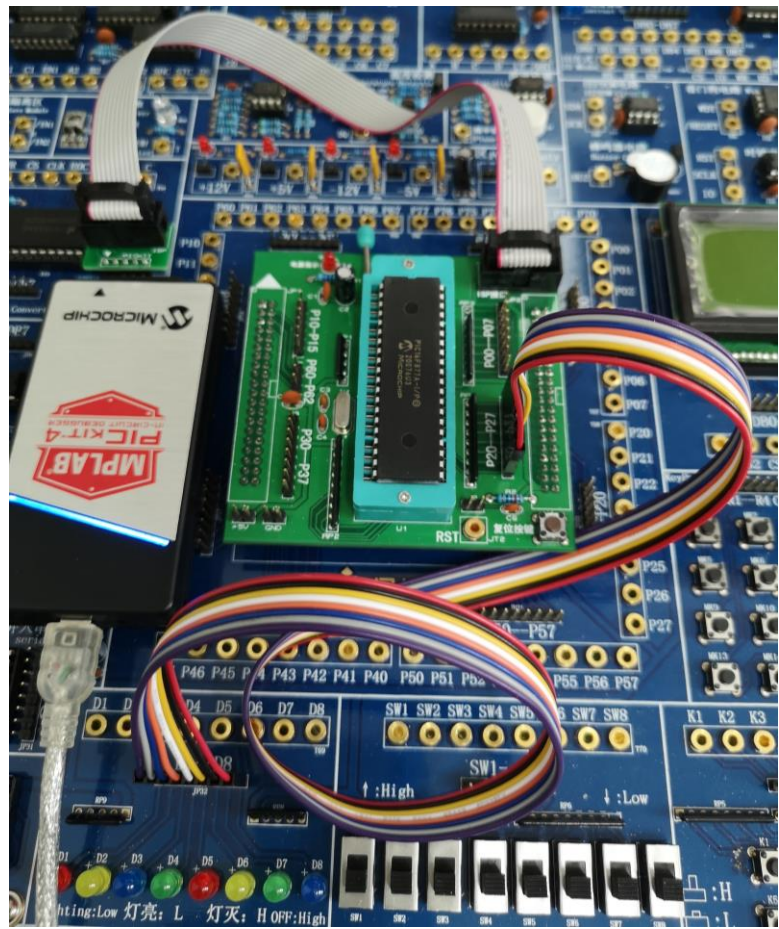


图 3-1-2 8 位 LED 流水灯实验接线图

1. 按照第二章所述 MPLAB X IDE 软件的使用步骤新建汇编语言程序项目和文件，将上述汇编程序录入，设置配置位，编译并排错。程序结构如图 3-1-3 所示。
2. 下载编程到目标板 CPU 中，并观察汇编程序连续运行的结果。**第一次修改程序**，延长汇编程序的时间，观察运行结果，会发现在走马灯到最边上一位时，下一步会出现全亮的情况。**第二次修改程序**，让流水灯朝右移动，始终让一个灯不显示，不能出现全亮的情况。
3. 按照第二章所述 MPLAB X IDE 软件的使用步骤新建 C 语言程序项目和文件，将上述 C 语言程序录入，设置配置位，编译并排错。程序结构如图 3-1-4 所示。
4. 调试 C 语言程序，用单步、连续运行及设置断点方式调试程序；
5. 按变化顺序记录 C 语言程序 PORTD 的值的变化情况。

表 3-1 RD 口的值的记录

端口	左移	右移
PORTD		

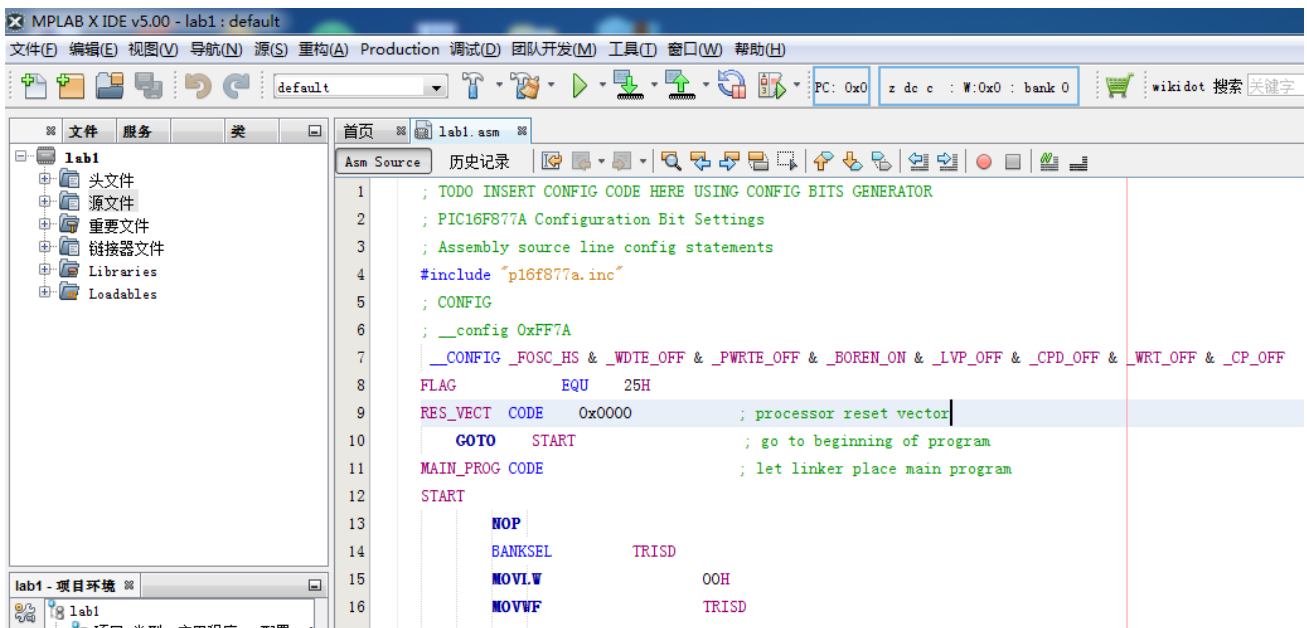


图 3-1-3 汇编语言程序结构

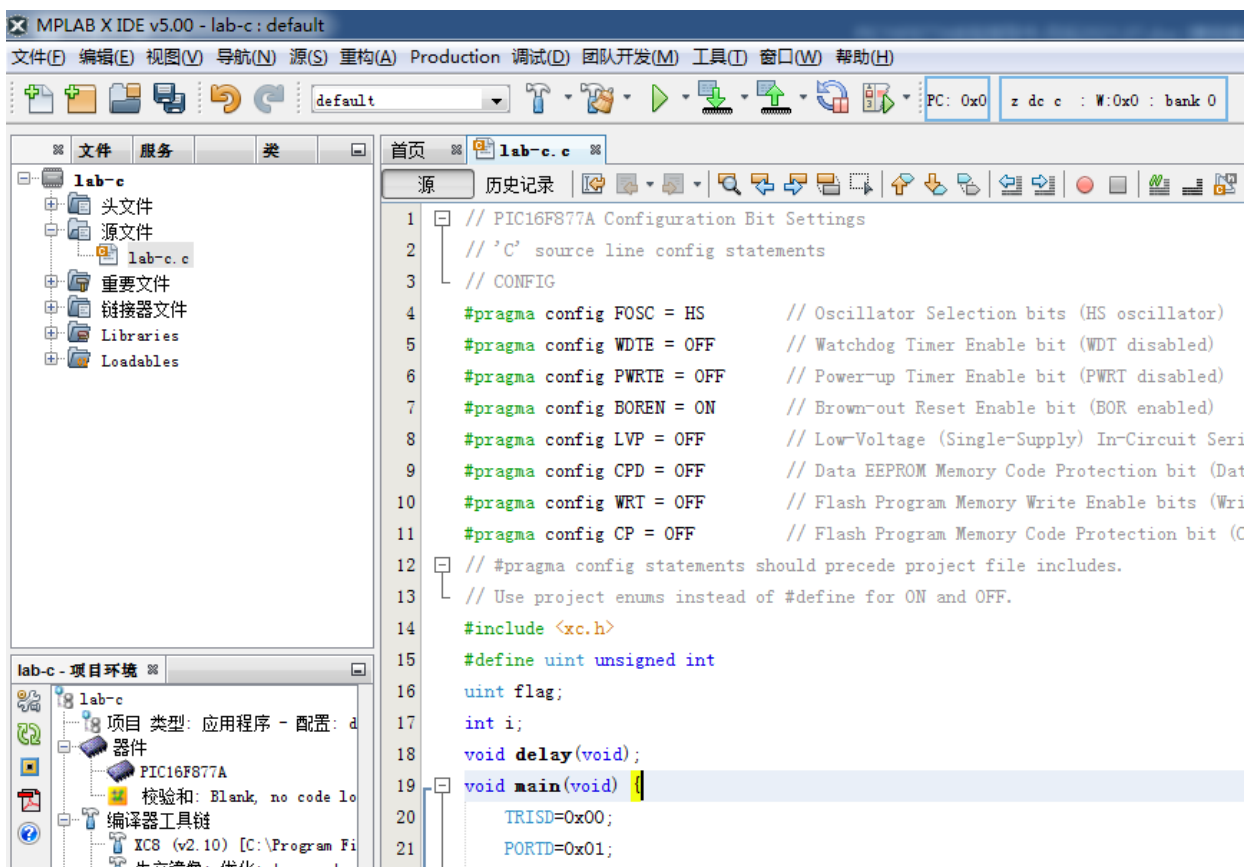


图 3-1-4 C 语言程序结构

五、实验报告要求

1. 根据实验内容中要求修改延时、改变灯向右移动的汇编程序写在实验报告里对主要语句加注释；
2. 记录走马灯左移右移寄存器 PORTD 值并填入表中。

实验二 外部中断实验

一、实验目的

1. 进一步掌握 MPLAB X IDE 软件的使用方法。
2. 掌握 I/O 口的方向配置。
3. 掌握单片机外部中断的寄存器及外部中断处理方法。

二、实验设备与器件

- | | |
|---------------|-----|
| 1. 计算机 | 1 台 |
| 2. 调试器 PICkit | 1 个 |
| 3. 实验箱 | 1 台 |
| 4. 导线若干 | |

三、实验内容

- ### 1、硬件电路原理图

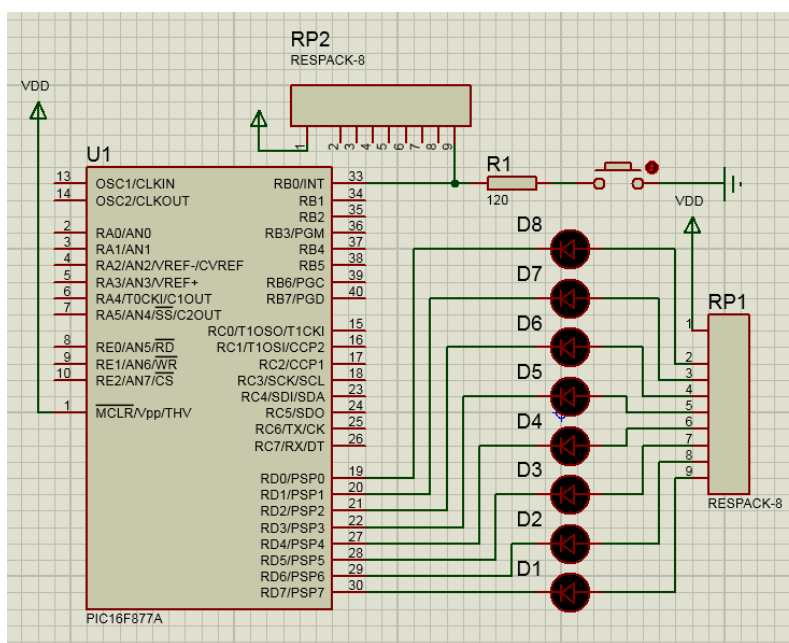


图 3-2-1 8 位独立 LED 和一个独立按键

- ## 2、硬件连接表

MCU-PIC16F877A	8 位独立 LED	独立按键
P20~P27 (RD)	D1~D8	
P00(INT0)(RB0)		K1

四、实验要求

在单片机的外部中断引脚上接一个按键做为外部中断输入，每次有中断时，CPU 处理中断服务程序。

五、实验步骤

1. 按照第二章所述实验箱简介和上述硬件接线表搭建好实验电路（也可参考图 3-1-2）。
2. 按照第二章所述 MPLAB X IDE 软件的使用步骤新建项目和文件，编写程序，设置配置位，编译并排错。
3. 下载编程到目标板 CPU 中，调试程序。

六、实验预习要求

1. 主要知识点概述：本实验利用外部中断 INT0，采用边沿触发方式引发中断。
2. 实验效果说明：当没有中断时 8 个 LED 按照大约 1 秒的方波周期闪烁，当有按键按下产生中断脉冲时，LED 做流水运动，移动时间大约 1 秒一次，每次只有一个灯点亮，1 分钟后没有中断响应恢复 LED 闪烁状态。

3. 汇编语言编写中断程序格式如下：

```
ISR          CODE    0x0004
```

```
.....
```

```
RETFIE
```

4. C 语言编写程序时，中断按照下面的格式来编写，其中的 time1 可以更改，其它格式不要改写。

```
void __interrupt() time1(void)
{
}
```

七、实验思考题

在外部中断中，如何解决按键的抖动而产生多次中断的问题。

实验三 定时器实验

一、实验目的

- 1. 掌握单片机定时器/计数器的寄存器设置；
- 2. 掌握定时器初值时间的计算；
- 3. 掌握定时器中断的设置。

二、实验设备与器件

- 1. 计算机 1 台
- 2. 调试器 PICkit 1 个
- 3. 实验箱 1 台
- 4. 导线若干

三、实验内容

- 1、硬件电路原理图

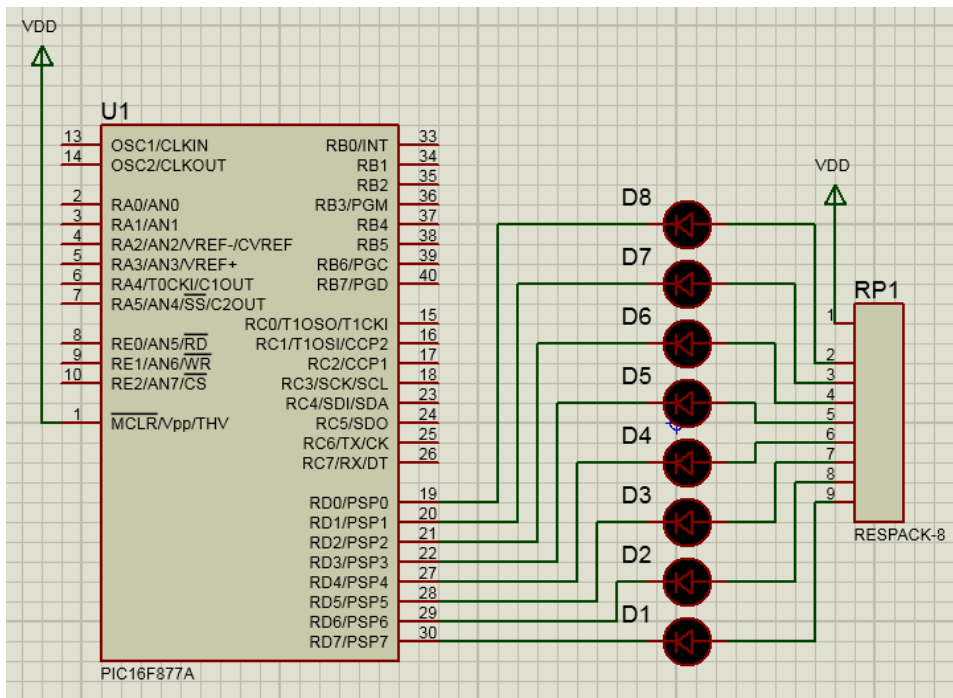


图 3-3-1 8 位独立 LED 和一个独立按键

2、硬件连接表

MCU-PIC16F877A	8 位独立 LED	独立按键
P20~P27 (RD)	D1~D8	

四、实验要求

利用单片机定时器定时 1S。

五、实验步骤

1. 按照第二章所述实验箱简介和上述硬件接线表搭建好实验电路（也可参考图 3-1-2）。
2. 按照第二章所述 MPLAB X IDE 软件的使用步骤新建项目和文件，编写程序，设置配置位，编译并排错。
3. 下载编程到目标板 CPU 中，调试程序。

六、实验预习要求

1. 主要知识点概述：本实验涉及到三个知识点：定时器的使用、中断响应以及中断程序。PIC16F877A 单片机内部有 3 个定时器，两个 8 位定时器的分别是 Timer0、Timer2，一个 16 位的定时器是 Timer1。选用哪一个都可以。
2. 实验效果说明：8 个 LED 每 2 秒钟闪烁一次，1 秒亮 1 秒灭。

七、实验思考题

1. 试着用定时器设置不同时间，实现不同的变化效果。
2. 试着用其它定时器实现类似的效果。

实验四 计数器实验

一、实验目的

- 1. 掌握单片机定时器/计数器的寄存器设置；
- 2. 掌握外部脉冲计数的方法；
- 3. 掌握计数器中断的设置。

二、实验设备与器件

- 1. 计算机 1 台
- 2. 调试器 PICkit 1 个
- 3. 实验箱 1 台
- 4. 导线若干

三、实验内容

1、硬件电路原理图

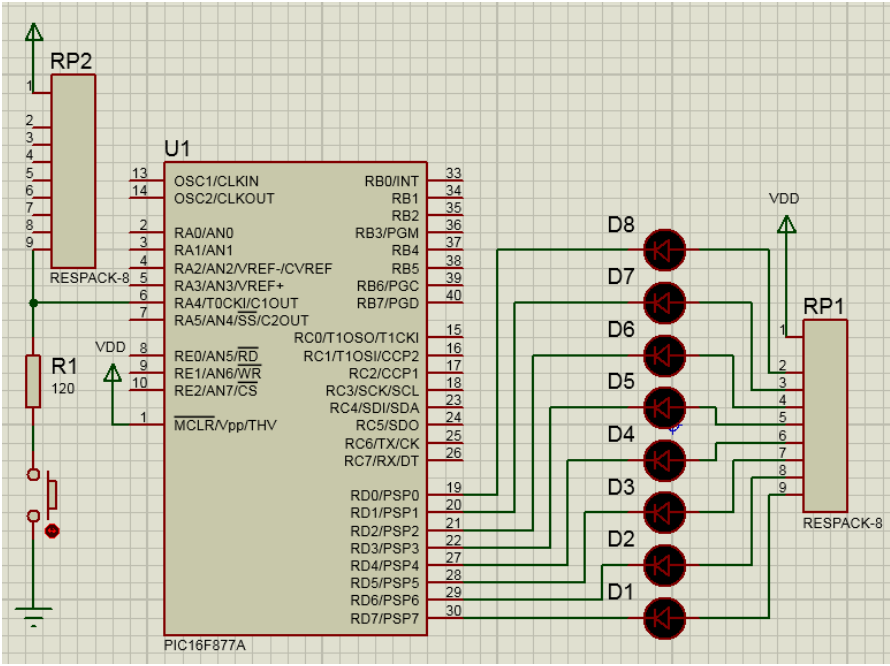


图 3-4-1 8 位独立 LED 和一个独立按键

2、硬件连接表

MCU-PIC16F877A	8 位独立 LED	独立按键
P20~P27 (RD)	D1~D8	
P14 (T0CKI) (RA4)		K1

四、实验要求

掌握定时器计数器的编程方法。

五、实验步骤

1. 按照第二章所述实验箱简介和上述硬件接线表搭建好实验电路（也可参考图 3-1-2）。
2. 按照第二章所述 MPLAB X IDE 软件的使用步骤新建项目和文件，编写程序，设置配置位，编译并排错。
3. 下载编程到目标板 CPU 中，调试程序。

六、实验预习要求

1. 主要知识点概述：

PIC16F877A 单片机内部有 3 个定时器计数器，两个 8 位的分别是 Timer0、Timer2，一个 16 位的是 Timer1。

2. 实验效果说明：

用按键模拟外部时钟发生器产生脉冲信号，用定时器计数器 0 计数并通过 LED 发光二极管显示出来。每次按下按键，计数器加 1。

七、实验思考题

1. 熟悉定时器计数器的其他设置，并编程实现，例如改变分频比，改变按键中断的方式。
2. 试着用 16 位的定时计数器对外部时钟计数，将按键接在 T1CKI（RC0-P30）引脚上，进一步熟练掌握定时器计数器的使用。

实验五 PWM 控制 LED 实验

一、实验目的

1. 掌握单片机定时器的使用;
2. 熟悉 PWM 波形的产生;

二、实验设备与器件

- | | |
|---------------|-----|
| 1. 计算机 | 1 台 |
| 2. 调试器 PICkit | 1 个 |
| 3. 实验箱 | 1 台 |
| 4. 导线若干 | |

三、实验内容

- ### 1、硬件电路原理图

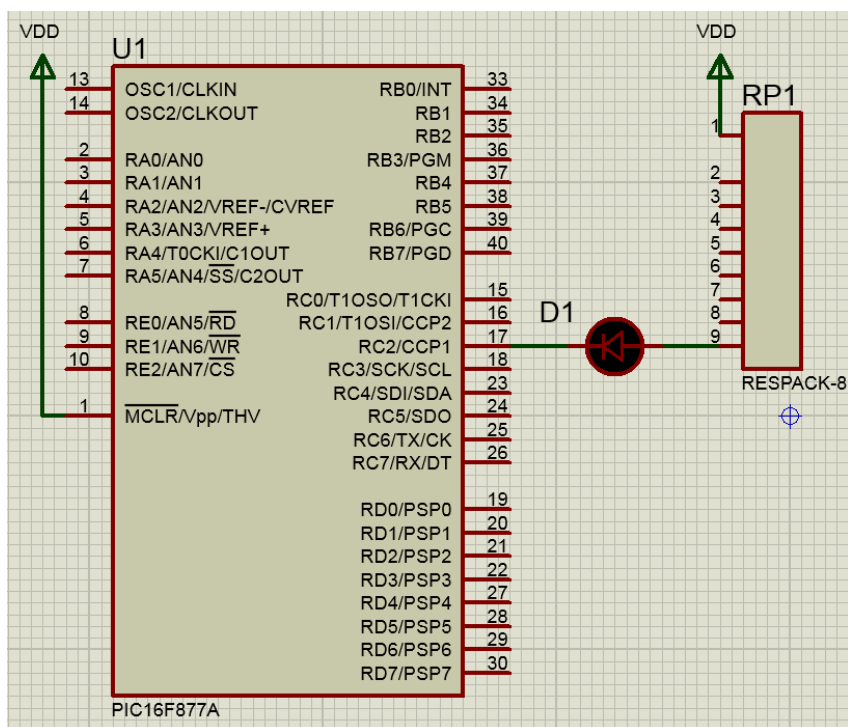


图 3-5-1 PWM 控制 LED 原理图

2、硬件连接表

MCU-PIC16F877A	8 位 LED 电路
P32 (RC2/CCP1)	D1

四、实验要求

用定时器设计 PWM 波形输出驱动控制 LED。

五、实验步骤

1. 按照第二章所述实验箱简介和上述硬件接线表搭建好实验电路（也可参考图 3-1-2）。
2. 按照第二章所述 MPLAB X IDE 软件的使用步骤新建项目和文件，编写程序，设置配置位，编译并排错。
3. 下载编程到目标板 CPU 中，调试程序。

六、实验预习要求

- 1、主要知识点概述：定时器的使用； PWM 的产生原理。
- 2、实验效果说明：PWM 控制 LED 点亮，LED 由暗变亮，由亮变暗。

七、实验思考题

试编写程序，用 PWM 驱动蜂鸣器演奏歌曲，实现电子钢琴功能。

实验六 数码管驱动显示实验

一、实验目的

- 1. 了解数码管显示原理；
- 2. 掌握读表程序的编写。

二、实验设备与器件

- 1. 计算机 1 台
- 2. 调试器 PICkit 1 个
- 3. 实验箱 1 台
- 4. 导线若干

三、实验内容

- 1、硬件电路原理图

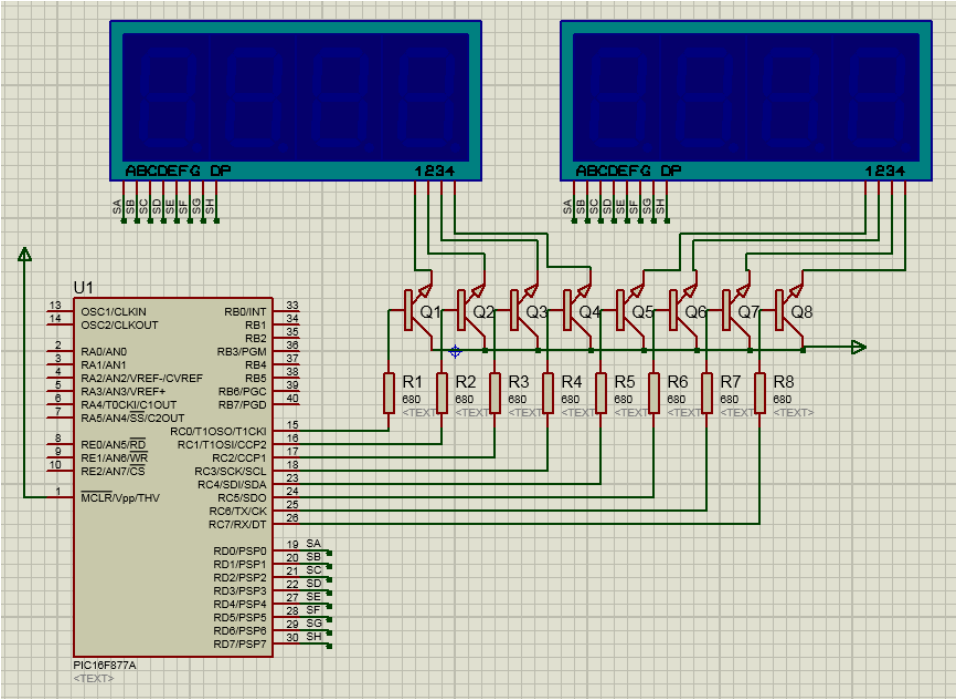


图 3-6-1 8 位共阳数码管显示原理图

- 2、硬件连接表

MCU-PIC16F877A	8 位共阳数码管
P30~P37 (RC0-RC7)	C0~C7
P20~P23 (RD0-RD7)	SA~SH

四、实验要求

利用 I/O 口实现动态扫描数码管显示。

五、实验步骤

1. 按照第二章所述实验箱简介和上述硬件接线表搭建好实验电路（也可参考图 3-1-2）。
2. 按照第二章所述 MPLAB X IDE 软件的使用步骤新建项目和文件，编写程序，设置配置位，编译并排错。
3. 下载编程到目标板 CPU 中，调试程序。

六、实验预习要求

1、主要知识点概述：

LED 数码显示原理：七段 LED 显示器内部由七个条形发光二极管和一个小圆点发光二极管组成，根据各管的极管的接线形式，可分成共阴极型和共阳极型。LED 数码管的 g-a 七个发光二极管因加正电压而发亮，因加零电压而不以发亮，不同亮暗的组合就能形成不同的字形，这种组合称之为字形码，下面给出共阳极的字形码及段码表格。

“0”	0C0H		“8”	80H
“1”	0F9H		“9”	90H
“2”	0A4H		“A”	88H
“3”	0B0H		“b”	80H
“4”	99H		“C”	0B6H
“5”	92H		“d”	0B0H
“6”	82H		“E”	86HH
“7”	F8H		“F”	8EH

由于显示的数字 0—9 的字形码没有规律可循，只能采用查表的方式来完成我们所需的要求。我们按着数字 0—9 的顺序，把每个数字的笔段代码按顺序排好，建立表格：

汇编语言：见后面举例汇编程序中 BMA 查表子程序。

C 语言：见后面举例 C 语言程序中 NUM[10]。

2、实验效果说明：数码管循环显示 0~9，并依次向左滚动，滚动时间不少于 1 秒。可参考后面的举例。

七、实验思考题

- 1、编写代码实现数码管环形流动，流动的方向、速度有规律有节奏的变化；
- 2、编写代码实现数字钟显示，要求显示时、分、秒、星期且之间要用点隔开。

汇编语言编程实现 8 位数码管显示 0-9 的数字：

```
; PIC16F877A Configuration Bit Settings
; Assembly source line config statements
#include "p16f877a.inc"
; CONFIG
; __config 0xFF7A
__CONFIG _FOSC_HS & _WDTE_OFF & _PWRTE_OFF & _BOREN_ON & _LVP_OFF & _CPD_OFF
& _WRT_OFF & _CP_OFF
COUNTER EQU 40H
RES_VECT CODE 0x0000 ; processor reset vector
        GOTO START ; go to beginning of program

; TODO ADD INTERRUPTS HERE IF USED

MAIN_PROG CODE ; let linker place main program

START
        NOP ;MPLAB 专用语句
        BSF STATUS, RP0 ;选择体 1
        CLRF TRISC ;PORTC 设置为输出
        CLRF TRISD ;PORTD 设置为输出
        BCF STATUS, RP0 ;选择体 0
        MOVLW 0FFH
        MOVWF PORTD ; PORTD 不显示
        MOVLW 01H
        MOVWF PORTC
ST CLRF COUNTER ;递增计数常数设置
LS MOVF COUNTER, W ;把循环计数器放入 W
        CALL BMA ;调要显示的数值的编码
        MOVWF PORTD
        CALL DELAY10MS ;动态扫描转换时间控制
        MOVLW 0FFH
        MOVWF PORTD ; 关显示
        BCF STATUS, C
        RLF PORTC
```

```

        BTFSS STATUS, C
        GOTO LT
LD      MOVLW 01H
        MOVWF PORTC
LT      INCF COUNTER          ;循环计算器加 1
        MOVF COUNTER, W      ;把计数次数放入 W
        SUBLW 0AH            ;与 0AH 相减
        BTFSS STATUS, Z      ;判断是否循环 10 次
        GOTO LS              ;返回主程序
        GOTO ST

```

;10MS 延时

DELAY10MS

```

        MOVLW 0FH            ;外循环常数
        MOVWF 20H            ;外循环寄存器
LOOP2   MOVLW 0FFH           ;中循环常数
        MOVWF 21H            ;中循环寄存器
L01     DECFSZ 21H, 1         ;中循环寄存器递减
        GOTO L01             ;继续中循环
        DECFSZ 20H, 1         ;外循环寄存器递减
        GOTO LOOP2           ;继续外循环
        RETURN               ;返回

```

;编码查询

```

BMA     ADDWF PCL, F          ;考察偏移量
        RETLW 0C0H           ;"0" 编码
        RETLW 0F9H           ;"1" 编码
        RETLW 0A4H           ;"2" 编码
        RETLW 0B0H           ;"3" 编码
        RETLW 99H            ;"4" 编码
        RETLW 92H            ;"5" 编码
        RETLW 82H            ;"6" 编码
        RETLW 0F8H           ;"7" 编码
        RETLW 80H            ;"8" 编码
        RETLW 90H            ;"9" 编码
        END

```

C 语言编程实现 8 位数码管显示 0-9 的数字：

```
#include<pic.h>

unsigned char NUM[10]={0XC0, 0XF9, 0XA4, 0XB0, 0X99, 0X92, 0X82, 0XF8, 0X80, 0X90};
unsigned char table[8]={0x01, 0x02, 0x04, 0x08, 0x10, 0x20, 0x40, 0x80};

void Delay_ms(unsigned int time)
{
    while(--time)
    {
        NOP();NOP();NOP();NOP();
    };
}

void Port_init()
{
    PORTD=0XFF;
    PORTC=0XFF;
    TRISC=0;
    TRISD=0;
}

void main()
{
    unsigned int x,y=0;
    Port_init();
    PORTC=0X00;
    while(1)
    {
        for(x=0;x<10;x++)
        {
            PORTC=table[y];
            PORTD=NUM[x];

            y++;
            if(y==8) y=0;
            Delay_ms(10000);
            PORTC=0x00;
        }
    }
}
```


实验七 内部模数转换实验

一、实验目的

- 1. 掌握单片机 10 位模/数转换的原理和编程方法；
- 2. 学会使用与 ADC 有关的特殊功能寄存器；
- 3. 通过编程实现 10 位 A/D 转换和 ADC 输出数据显示；

二、实验设备与器件

- 1. 计算机 1 台
- 2. 调试器 PICkit 1 个
- 3. 实验箱 1 台
- 4. 导线若干

三、实验内容

1、硬件电路原理图

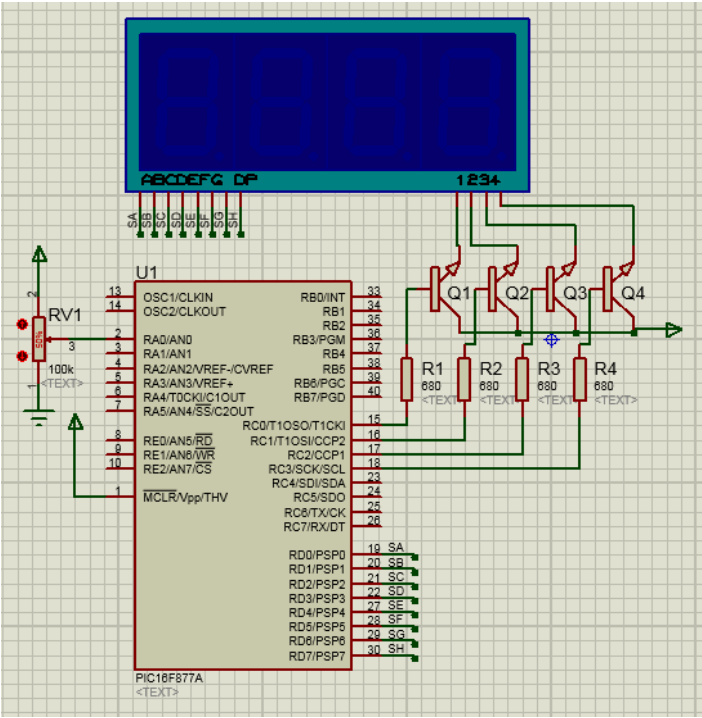


图 3-7-1 单路采样显示电路原理图

2、硬件连接表

MCU-PIC16F877A	4 位共阳数码管	电位器
P30~P33 (RC0-RC3)	C0~C3	
P20~P23 (RD0-RD7)	SA~SH	
P10 (RA0)		100K

四、实验要求

掌握与 A/D 转换有关的寄存器的设置。

五、实验步骤

1. 按照第二章所述实验箱简介和上述硬件接线表搭建好实验电路（也可参考图 3-1-2）。
2. 按照第二章所述 MPLAB X IDE 软件的使用步骤新建项目和文件，编写程序，设置配置位，编译并排错。
3. 下载编程到目标板 CPU 中，调试程序。

六、实验预习要求

1、主要知识点概述：A/D 转换器大致有三类：一是双积分 A/D 转换器，优点是精度高，抗干扰性好，价格便宜，但速度慢；二是逐次逼近 A/D 转换器，精度、速度、价格适中；三是并行 A/D 转换器，速度快，价格也昂贵。

2、实验效果说明：调整电位计，得到不同的电压值，转换后的数据通过数码管输出。

七、实验思考题

1、编程控制 8 路通道进行依次 AD 采样，然后显示，间隔 5 秒采样下一个通道。最左边显示通道号，右边两位显示十进制的电压值（0.0~5.0），保留小数点后面一位。

实验八 矩阵键盘扫描显示实验

一、实验目的

- 1. 理解矩阵键盘扫描的原理；
- 2. 掌握矩阵键盘与单片机接口的编程方法。
- 3. 进一步掌握数码管显示与单片机接口的动态显示编程方法。

二、实验设备与器件

- 1. 计算机 1 台
- 2. 调试器 PICkit 1 个
- 3. 实验箱 1 台
- 4. 导线若干

三、实验内容

1、硬件电路

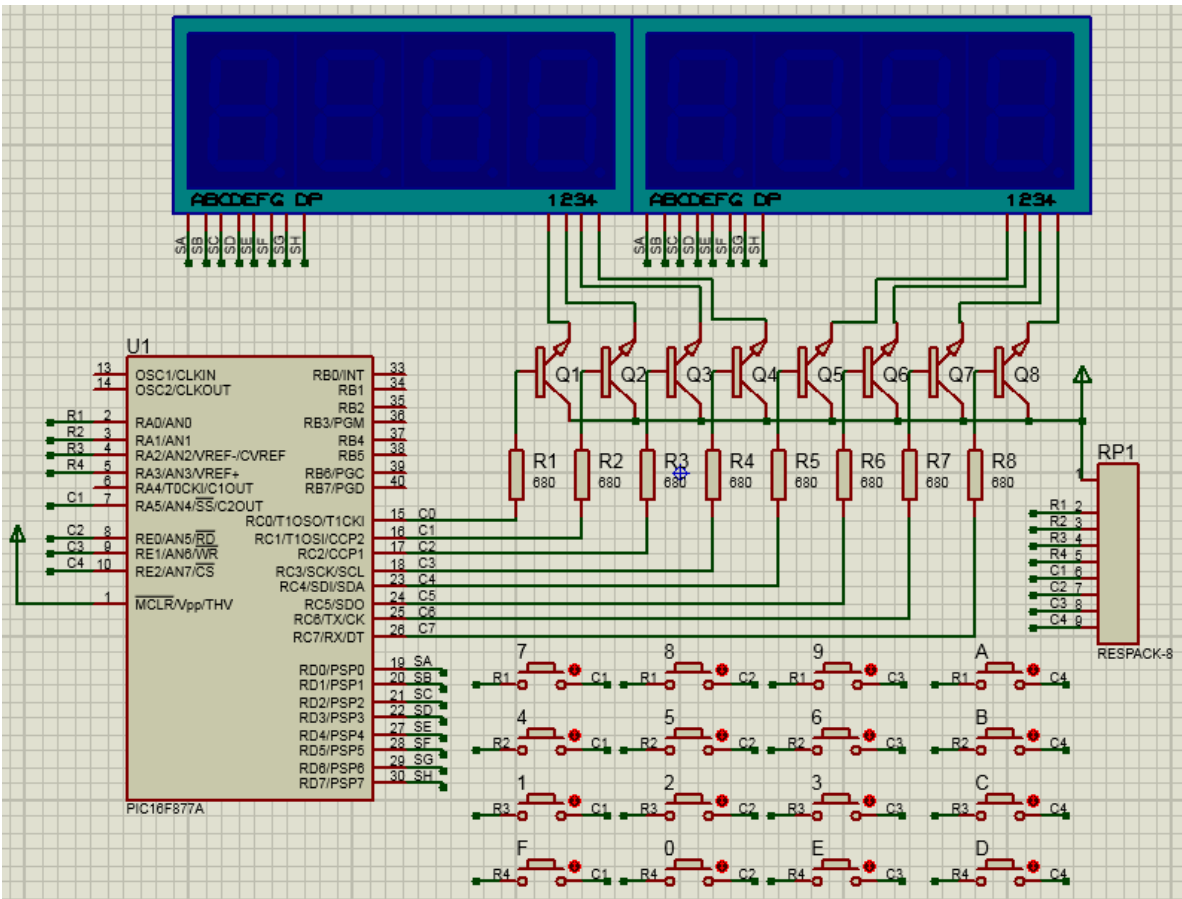


图 3-11-1 矩阵键盘扫描显示电路原理图

2、硬件连接表

MCU-PIC16F877A	8 位共阳数码管	4*4 矩阵键盘
P20~P27 (RD0~RD7)	SA~SH	
P30~P37 (RC0~RC7)	C0~C7	
P10~P13 (RA0~RA3)		(KeyPad) R1~R4
P15, P60~P62 (RA5, RE0~RE2)		(KeyPad) C1~C4

四、实验要求

读取矩阵键盘键值，对键值进行判断和处理，每次按键的键值都显示在最左边，前面的键值显示依次向右滚动。

五、实验步骤

1. 按照第二章所述实验箱简介和上述硬件接线表搭建好实验电路（也可参考图 3-1-2）。
2. 按照第二章所述 MPLAB X IDE 软件的使用步骤新建项目和文件，编写程序，设置配置位，编译并排错。
3. 下载编程到目标板 CPU 中，调试程序。

六、实验预习要求

- 1、主要知识点概述：

本实验阐述了键盘扫描原理，过程如下：首先扫描键盘（通过行扫描读列，或者列扫描读行），判断是否有键按下，再确定是哪一个键，计算键值，输出显示。

- 2、实验效果说明：有按键按下时数码管最左边显示对应的键值，前面的键值显示依次向右滚动。

七、实验思考题

- 1、本例程采用的是查询的方式，也可以使用中断方式进行读键值，需要连接一个 4 或门电路，编写程序测试。
- 2、熟练掌握矩阵键盘后，可编写程序，实现功能键。类似 PC 机的 ALT+其他键，可编写程序测试。

第四章 PIC16F877A 创新综合实验

实验一 步进电机控制实验

一、实验目的

1. 了解步进电机控制的基本原理；
2. 掌握控制步进电机转动的编程方法。

二、实验设备与器件

1. 计算机 1 台
2. 调试器 PICkit 1 个
3. 实验箱 1 台
4. 导线若干

三、实验内容

1. 硬件电路原理图

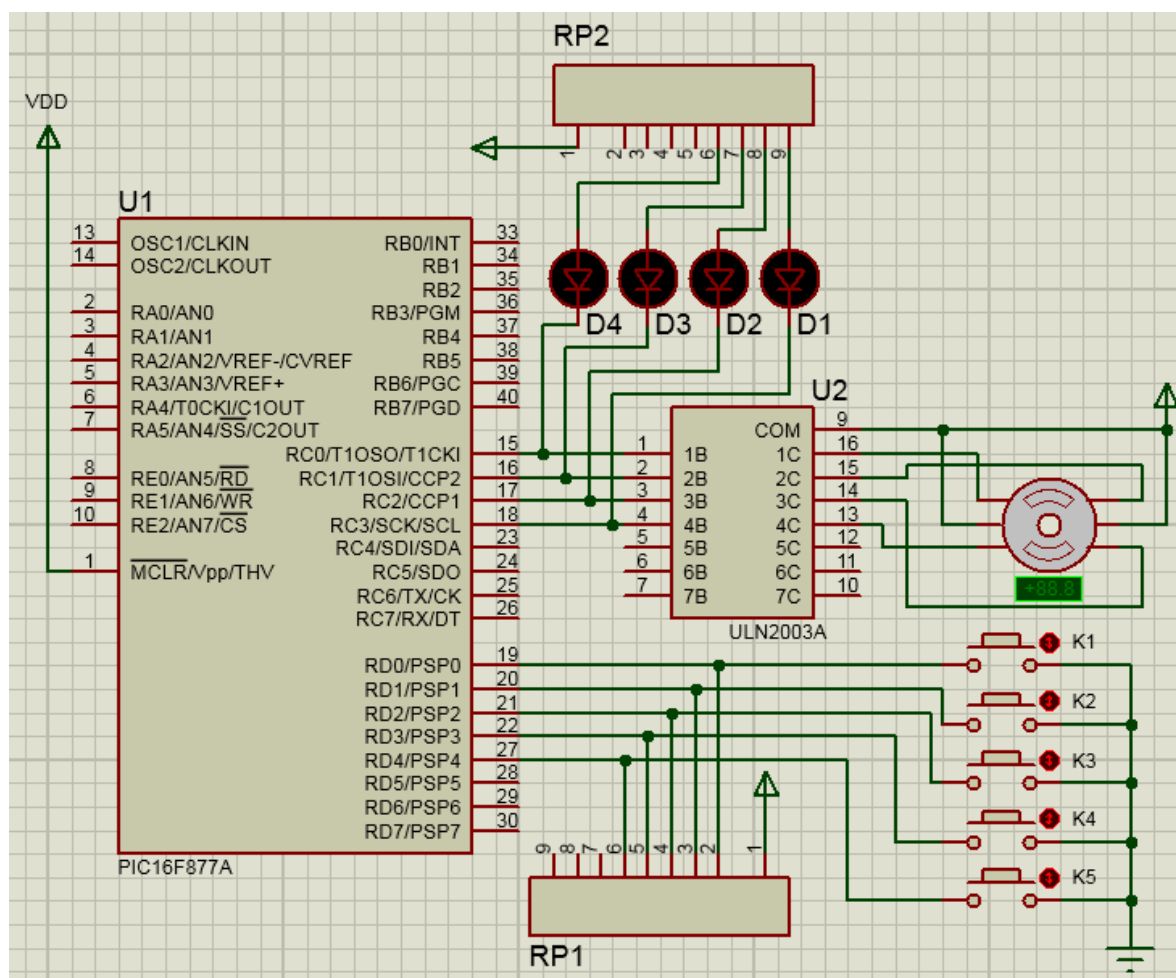


图 3-8-1 步进电机驱动电路原理图

2、硬件连接表

MCU-PIC16F877A	8 位独立按键	电机模块	8 位 LED 电路
P30~P33		SA~SD	D3~D1
P20	K1	正转	
P21	K2	反转	
P22	K3	快速	
P23	K4	慢速	
P24	K5	停止	

四、实验要求

利用单片机实现对步进电机的控制，编写程序，用四路 I/O 口实现环形脉冲的分配，控制步进电机按固定方向连续转动。同时，要求分别用一个触动按钮控制步进电机正转、反转、快速、慢速、停止。该实验用到的电机的步距角是 7.5° /步。

五、实验步骤

1. 按照第二章所述实验箱简介和上述硬件接线表搭建好实验电路（也可参考图 3-1-2）。
2. 按照第二章所述 MPLAB X IDE 软件的使用步骤新建项目和文件，编写程序，设置配置位，编译并排错。
3. 下载编程到目标板 CPU 中，调试程序。

六、实验预习要求

1、主要知识点概述：

步进电机驱动原理是通过对每组线圈中电流的顺序切换，来使电机作步进式旋转。切换是通过单片机输出脉冲信号来实现的。所以调节脉冲信号的频率就可以改变步进电机的转速，改变各相脉冲的先后顺序，就可以改变电机的转向。步进电机的转速应由慢到快逐步加速。

2、实验效果说明：按下 K1 键正转，按下 K2 键反转，按下 K3 键快速（慢速的 10 倍），按下 K4 键慢速，按下停止键电机停止转动。

七、实验思考题

通过不同的节拍方式来控制步进电机。

实验二 LCD1602 显示实验（IO 方式驱动）

一、实验目的

1. 了解字符型液晶显示屏的控制原理和方法；
2. 了解数字和字符的显示原理；
3. 掌握数字和字符的程序编写。

二、实验设备与器件

1. 计算机 1 台
2. 调试器 PICkit 1 个
3. 实验箱 1 台
4. 导线若干

三、实验原理和内容

1、实验原理

LCD1602 也叫 1602 字符型液晶，它是一种专门用来显示字母、数字、符号等的点阵型液晶模块。它由若干个 5X7 或者 5X11 等点阵字符位组成，每个点阵字符位都可以显示一个字符，每位之间有一个点距的间隔，每行之间也有间隔，起到了字符间距和行间距的作用。LCD1602 是指显示的内容为 16X2，即可以显示两行，每行 16 个字符液晶模块（显示字符和数字）。

市面上字符液晶大多数是基于 HD44780 液晶芯片的，控制原理是完全相同的，因此基于 HD44780 写的控制程序可以很方便地应用于市面上大部分的字符型液晶，本实验就是基于 HD44780 来编写程序。下面这个列表是 LCD1602 每个引脚的说明。

引脚号	符号	引脚说明	引脚号	符号	引脚说明
1	VSS	电源地	9	D2	数据端口
2	VDD	电源正极	10	D3	数据端口
3	VEE	偏压信号	11	D4	数据端口
4	RS	命令/数据	12	D5	数据端口
5	RW	读/写	13	D6	数据端口
6	E	使能	14	D7	数据端口
7	D0	数据端口	15	BLA	背光正极
8	D1	数据端口	16	BLK	背光负极

1602 液晶内部带了 80 个字节的显示 RAM，用来存储我们发送的数据，它的结构如下图所示。

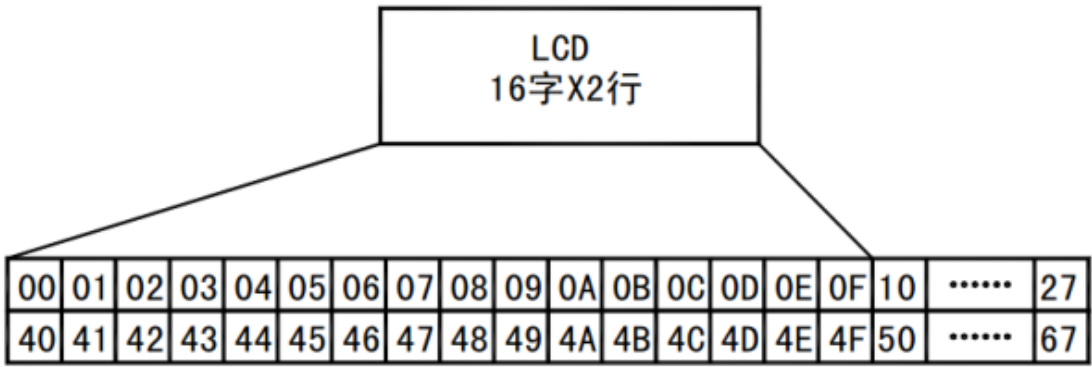


图 3-9-1 1602 内部 RAM 结构图

第一行的地址是 0x00H 到 0x27，第二行的地址从 0x40 到 0x67，其中第一行 0x00 到 0x0F 是与液晶上第一行 16 个字符显示位置相对应的，第二行 0x40 到 0x4F 是与第二行 16 个字符显示位置相对应的。而每行都多出来一部分，是为了显示移动字幕设置的。1602 字符液晶是显示字符的，因此它跟 ASCII 字符表是对应的。比如我们给 0x00 这个地址写一个 ‘a’，也就是十进制的 97，液晶的最左上方的那个小块就会显示一个字母 a。

单片机是通过硬件接口向 LCD 发送各种指令来控制 LCD 显示。LCD1602 的控制指令共有 11 条，指令的格式和功能可以查阅 HD44780 的数据手册。

2、硬件电路原理图

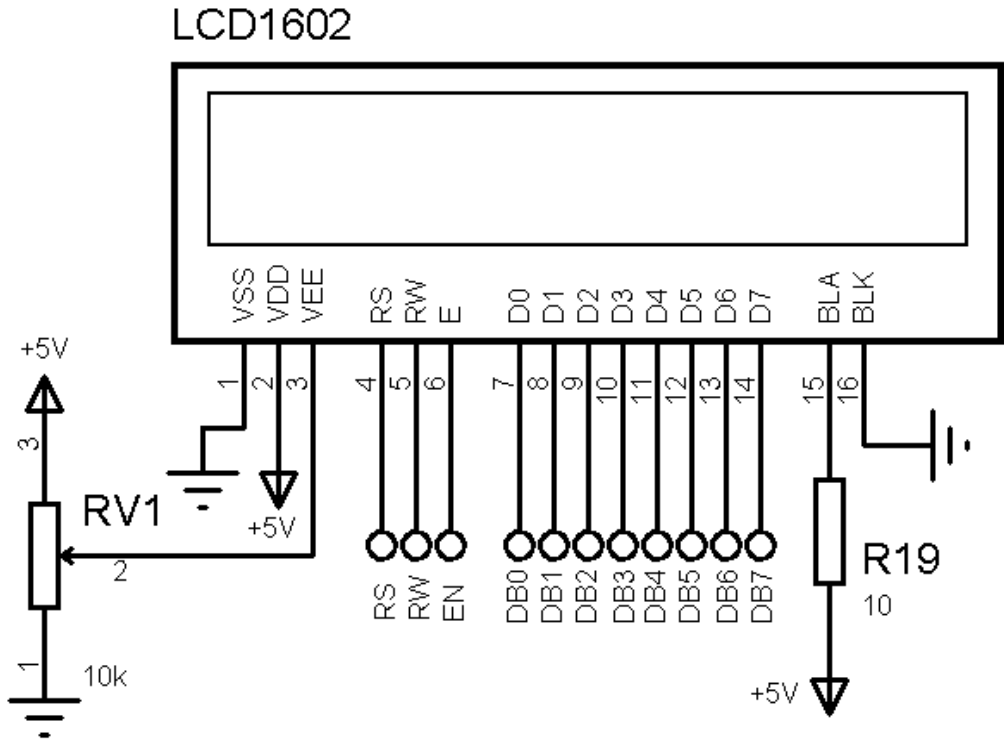


图 3-9-2 LCD1602 字符屏

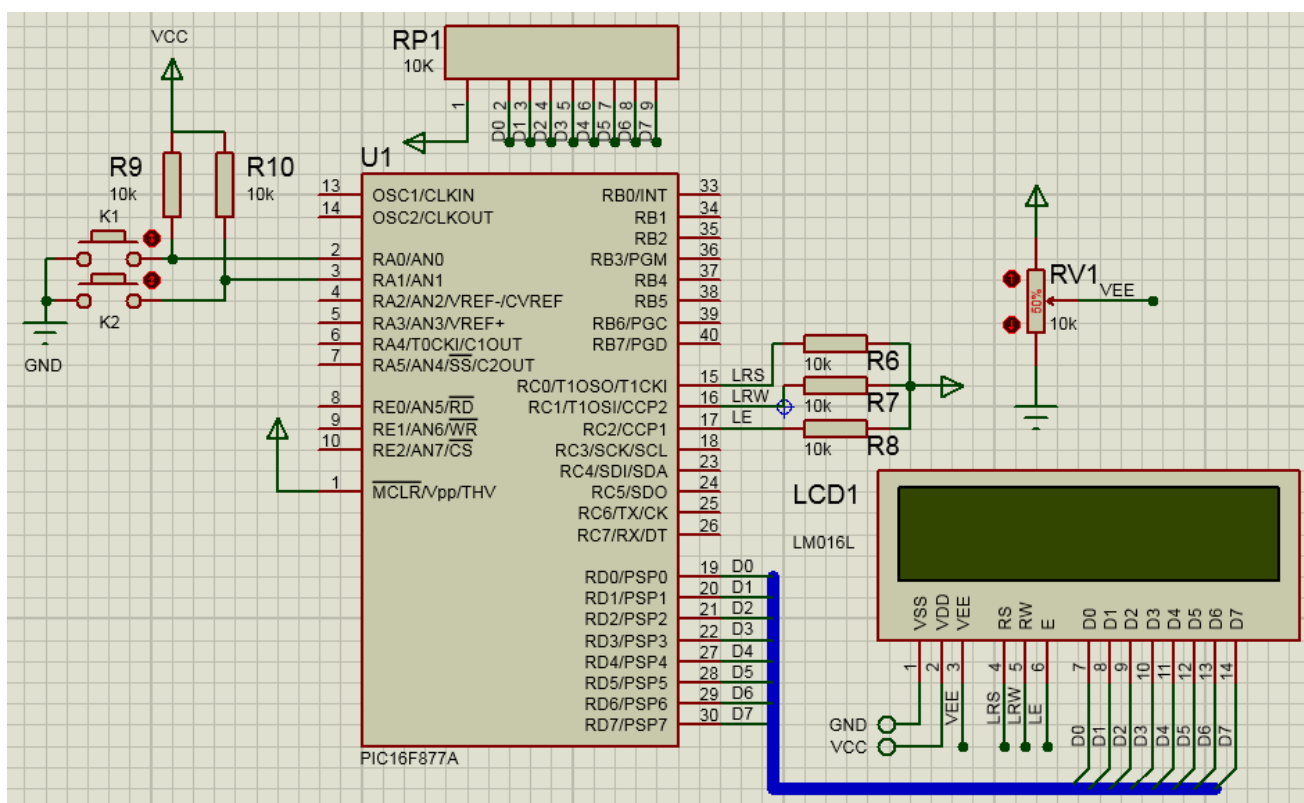


图 3-9-3 LCD1602 显示电路原理图

3、硬件连接表

MCU-PIC16F877A	LCD1602 字符屏
P20~P27 (RD0~RD7)	D0~D7
P30 (RC0)	RS
P31 (RC1)	RW
P32 (RC2)	EN
P10 (RA0)	K1
P11 (RA1)	K1

四、实验要求

利用实验板的液晶显示屏电路，编写程序, 每按下 K1 键一次，在第一行控制输出显示数字 0~9，并依次从左向右显示，每按下 K2 键一次，在第二行控制输出显示英文字符，并依次从左向右显示。

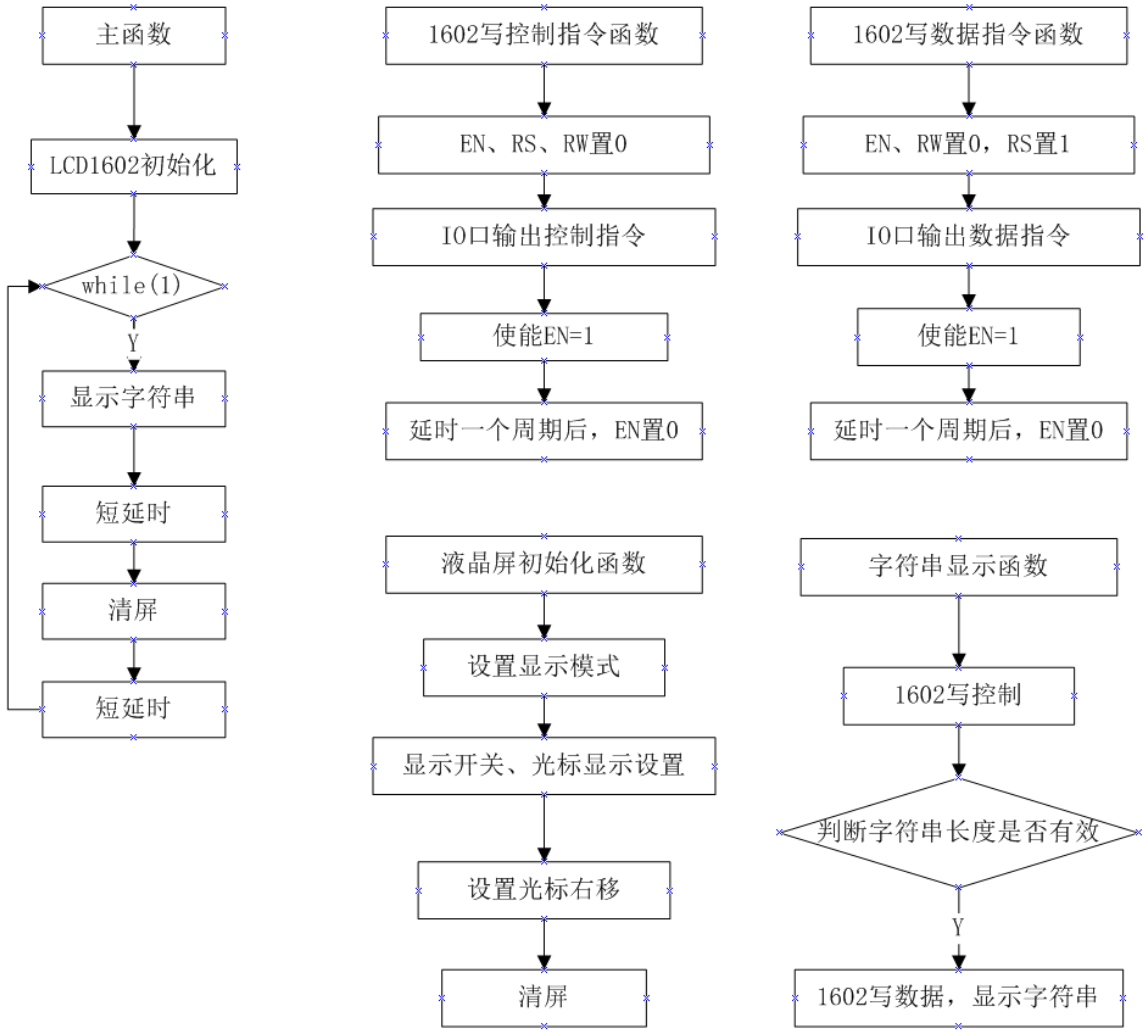
五、实验步骤

1. 按照第二章所述实验箱简介和上述硬件接线表搭建好实验电路（也可参考图 3-1-2）。
2. 按照第二章所述 MPLAB X IDE 软件的使用步骤新建项目和文件，编写程序，设置配置位，编译并排错。
3. 下载编程到目标板 CPU 中，调试程序。

六、实验预习要求

参考以下程序流程图和查阅 HD44780 的数据手册按照上述要求编写程序。每按下 K1 键一

次，在第一行控制输出显示数字 0-9，并依次从左向右显示，每按下 K2 键一次，在第二行控制输出显示英文字符，并依次从左向右显示。



七、实验思考题

怎样增加光标的闪动实现人机界面的输入？

实验三 16X16 点阵汉字显示实验

一、实验目的

1. 了解阵列 LED 屏扫描显示的原理；
2. 掌握汉字、图形取字模软件的使用及编写程序驱动点阵显示。

二、实验设备与器件

1. 计算机 1 台
2. 调试器 PICkit 1 个
3. 实验箱 1 台
4. 导线若干

三、实验内容和原理

1、点阵 LED

点阵 LED 显示屏作为一种现代电子媒体,具有灵活的显示面积(可任意分割和拼装)、高亮度、长寿命、数字化、实时性等特点,应用非常广泛。

一个数码管由 8 个 LED 组成,同理,一个 8*8 的点阵就是由 64 个 LED 小灯组成。图 3-10-1 就是一个点阵 LED 最小单元,即一个 8*8 的点阵 LED。本实验是使用 4 个 8*8 点阵拼成 16*16 的点阵。16X16 点阵共需要 256 个发光二极管组成,且每个发光二极管是放置在行线和列线的交叉点上,当对应的某一系列置 0 电平,某一行置 1 电平时,相应的位置点亮。



图 3-10-1 8*8 的点阵 LED

2、74LS138

74LS138 为 3 线—8 线译码器,如图 3-10-2 所示。

A、B、C: 地址输入端

G1 (E1): 选通端

/G2A (/E2)、/G2B (/E3): 选通端(低电平有效)

/Y0~ /Y7: 输出端(低电平有效)

VCC: 电源正

GND: 地

A、B、C 对应 Y0—Y7; A、B、C 以二进制形式输入,然后转换成十进制,对应相应 Y 的序号输出低电平,其他均为高电平。

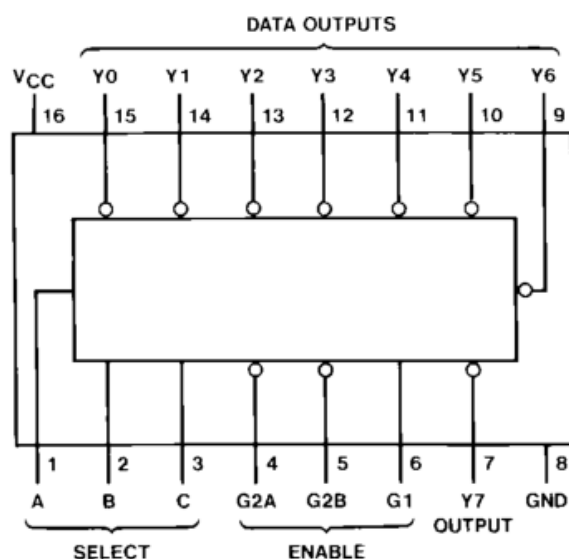


图 3-10-2 74LS138 原理图

74LS138 真值表

Inputs					Outputs							
Enable		Select										
G1	G2 (Note 1)	C	B	A	Y0	Y1	Y2	Y3	Y4	Y5	Y6	Y7
X	H	X	X	X	H	H	H	H	H	H	H	H
L	X	X	X	X	H	H	H	H	H	H	H	H
H	L	L	L	L	L	H	H	H	H	H	H	H
H	L	L	L	H	H	L	H	H	H	H	H	H
H	L	L	H	L	H	H	L	H	H	H	H	H
H	L	L	H	H	H	H	H	L	H	H	H	H
H	L	H	L	L	H	H	H	H	L	H	H	H
H	L	H	L	H	H	H	H	H	H	L	H	H
H	L	H	H	L	H	H	H	H	H	H	L	H
H	L	H	H	H	H	H	H	H	H	H	H	L

3、74HC595

74HC595 是一个 8 位串行输入、并行输出的位移寄存器。如图 3-10-3 所示。

74HC595 引脚功能如下表所示：

符号	引脚	描述
Q0--Q7	第 15 脚，第 1-7 脚	8 位并行数据输出，
GND	第 8 脚	地
Q7'	第 9 脚	串行数据输出
/MR	第 10 脚	主复位（低电平）
SHCP	第 11 脚	数据输入时钟线
STCP	第 12 脚	输出存储器锁存时钟线
/OE	第 13 脚	输出有效（低电平）
DS	第 14 脚	串行数据输入
VCC	第 16 脚	电源

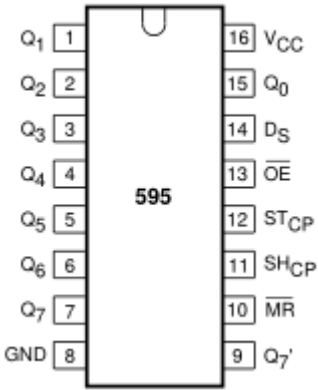


图 3-10-3 74HC595 封装图

74HC595 真值表如下表所示：

INPUTS					OUTPUTS		FUNCTION
SH _{CP}	ST _{CP}	\overline{OE}	\overline{MR}	D _S	Q ₇ '	Q _N	
X	X	L	L	X	L	NC	a LOW level on \overline{MR} only affects the shift registers
X	↑	L	L	X	L	L	empty shift register loaded into storage register
X	X	H	L	X	L	Z	shift register clear. Parallel outputs in high-impedance OFF-state
↑	X	L	H	H	Q ₆ '	NC	logic high level shifted into shift register stage 0. Contents of all shift register stages shifted through, e.g. previous state of stage 6 (internal Q ₆ ') appears on the serial output (Q ₇ ')
X	↑	L	H	X	NC	Q _n '	contents of shift register stages (internal Q _n ') are transferred to the storage register and parallel output stages
↑	↑	L	H	X	Q ₆ '	Q _n '	contents of shift register shifted through. Previous contents of the shift register is transferred to the storage register and the parallel output stages.

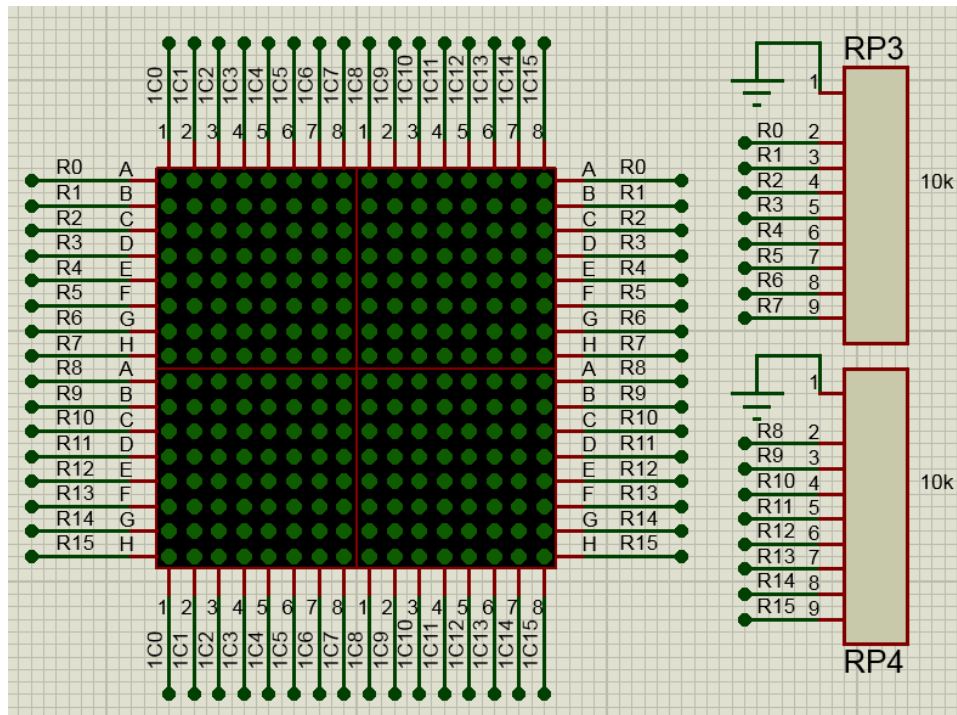


图 3-10-7 16*16 点阵 LED 电路图

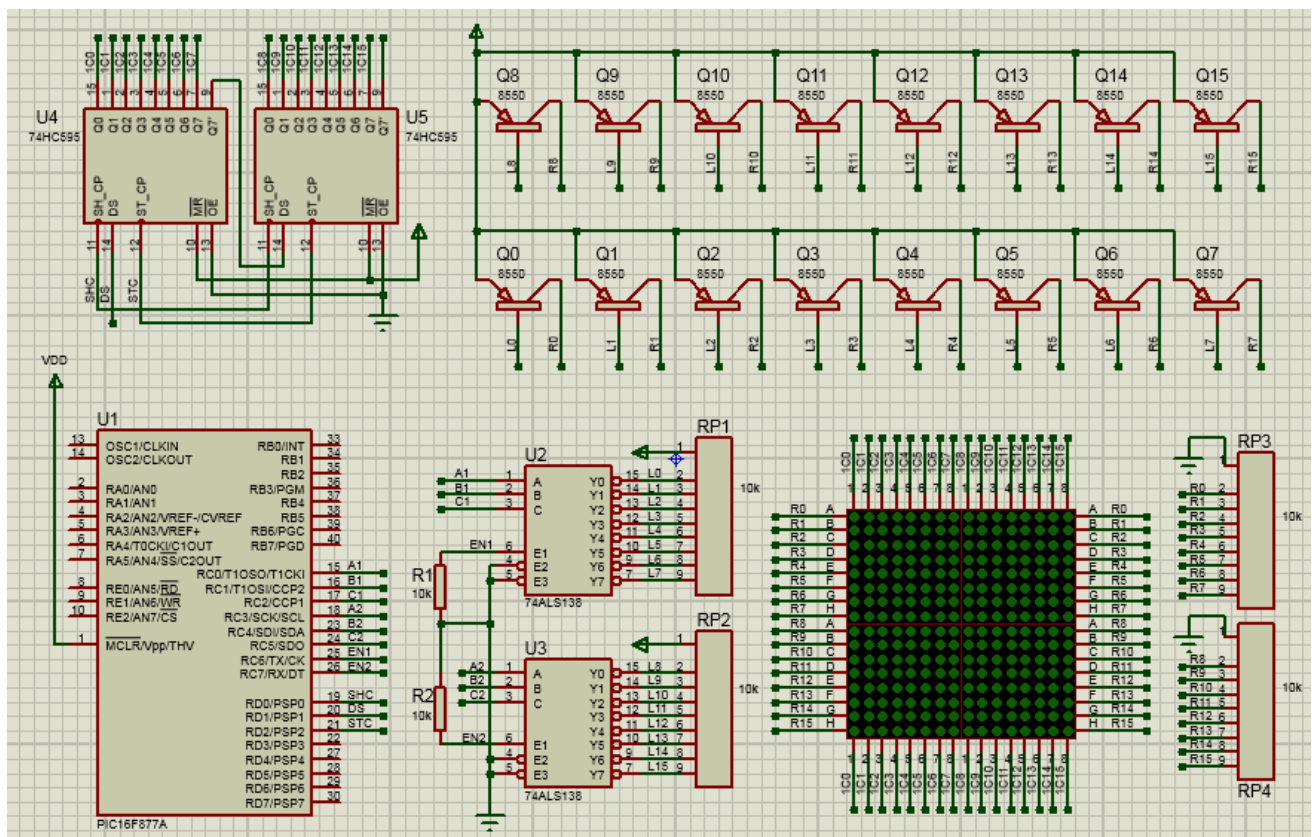


图 3-10-8 16*16 点阵 LED 电路原理图

2、硬件连接表

MCU-PIC16F877A	16X16 点阵模块
P20 (RD0)	SHC
P21 (RD1)	DS
P22 (RD2)	STC
P30 (RC0)	A1
P31 (RC1)	B1
P32 (RC2)	C1
P33 (RC3)	A2
P34 (RC4)	B2
P35 (RC5)	C2
P36 (RC6)	EN1
P37 (RC7)	EN2

四、实验要求

利用 74HC595、74HC138 进行行列扫描驱动 16x16LED 屏，实现汉字和图形的循环显示。

五、实验步骤

1. 按照第二章所述实验箱简介和上述硬件接线表搭建好实验电路（也可参考图 3-1-2）。
2. 按照第二章所述 MPLAB X IDE 软件的使用步骤新建项目和文件，编写程序，设置配置位，编译并排错。
3. 下载编程到目标板 CPU 中，调试程序。

六、实验预习要求

1、主要知识点概述：

16X16 点阵共需要 256 个发光二极管组成，且每个发光二极管是放置在行线和列线的交叉点上，当对应的某一列置 0 电平，某一行置 1 电平时，该点亮，如图 3-10-9 所示。

2、实验效果说明：把程序下载到单片机后，接好线显示“华南理工大学机汽学院”每个字间隔 2 秒。

七、实验思考题

编写程序实现文字的左移入或者是右移入。

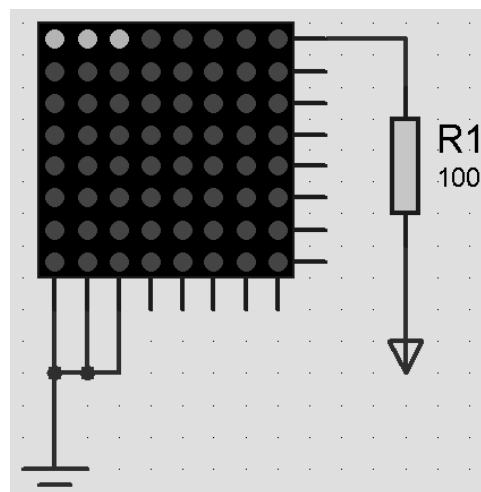


图 3-10-9 点阵 LED 图