

# 实验 1 实验板结构和开发工具使用

## 一、实验目的

1. 了解 MSP430G2 实验板结构，掌握实验板的基本检测方法；
2. 掌握单片机 CCS 集成开发环境基本使用方法；
3. 掌握 MSP430G2553 端口 P1~P2 基本输入/输出引脚与相关 I/O 寄存器的关系。

## 二、实验设备

1. 带 USB 接口的 PC 机一台；
2. MSP430G2553 实验系统一套。

## 三、实验任务

### 1. 了解 MSP430G2 实验系统结构，掌握扩展板检测方法

参看“附录 B MSP430G2xxx 实验系统简介”中电路原理图，了解电源、发光二极管、按键、蜂鸣器等电路工作原理，设计检测这些电路的方法，并记录检测结果。

### 2. CCS 开发工具的初步使用

- 1) 确认扩展板上单片机的引脚 P2.2 和 P2.5 已分别与发光二极管的控制端 L3、L6 短接；
- 2) 参看“附录 A CCS 使用入门”中的一至五，在工程空间下建立项目 test，了解 C 语言程序项目建立、编辑、编译、连接、程序下载和运行方法。注意附录 A 源程序 test\_g2553.C 提供有电子版。  
**注意:工程空间名称、项目名称和 C 的源程序、以及存放的文件夹均请勿用中文字符命名。**
- 3) 在 test 项目中，修改源程序 test\_g2553.C，在 test\_g2553.C 中的语句“unsigned int i;”前加一个关键字 register，将变量 i 指定用 CPU 内部的寄存器，并将其放在 main() 函数中，重新编译、连接和下载，观看执行的现象有什么不同。本次实验只看现象，暂不要求解释原因。
- 4) 掌握查看和修改寄存器、存储器单元、变量的方法  
参看“附录 A CCS 使用入门”中的“六、查看当前单片机状态”，掌握查看和修改 CPU 寄存器、存储器单元内容的方法。练习下面操作：
  - (1) View>Registers>Core Registers 将 CPU 寄存器 R6、R7 分别修改为 0x6789，0xABCD；
  - (2) View>Registers>Port\_1\_2 将 P2DIR 修改为 0xFF，P2SEL 改为 0，将 P2OUT 修改为 0xA5；
  - (3) 用 View/Expressions 或 View/Variables>Add 或选中变量 i 右键 add to watch，查看变量 i 的内容，并将 i 当前的值改为 9876。
- 5) 退出 CCS

### 3. 了解端口 P1~P2 基本输入/输出的引脚 Px.7~Px.0 与相关 I/O 寄存器的关系

用短线块将单片机的引脚 P2.7~P2.0 分别与发光二极管的控制端 L8~L1 短接，引脚 P1.7~P1.0 分别与按键的控制端 K8~K1 短接，同时如右图 1-1 将实验板 JP8 处中间两个插针分别用短线块接至 P2.6、P2.7 侧。

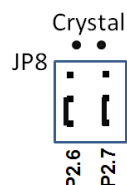


图 1-1 实验板 JP8 跳线连接

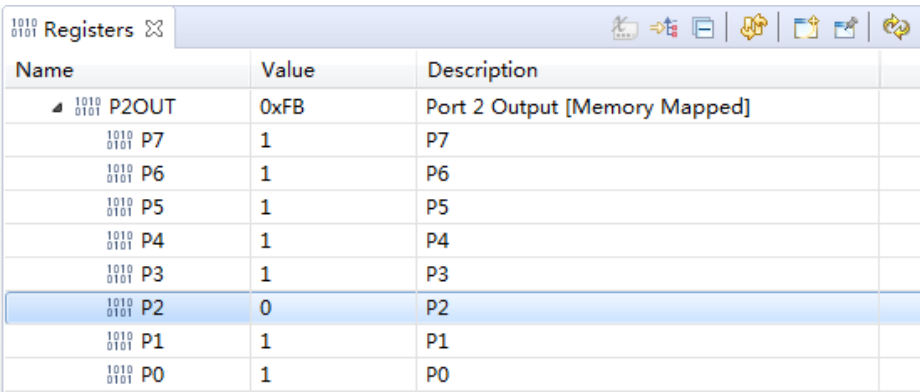
运行 CCS，打开任务 2 的项目 test，进入 DEBUG。。在 View 中打开 Registers 窗口，查看端口 P1/P2 寄存器，修改下面 I/O 寄存器内容，结合端口 P1 和 P2 引脚与外部电路的连接，利用附录 B 中实验板原理图，解释观察到的现象。

- 1) 修改寄存器 P2SEL=0x00，P2SEL2=0x00，P2DIR=0xFF，设置端口 P2 为基本输出功能，按表 1-1，修改寄存器 P2OUT 的值，记录看到发光二极管 L8~L1 的现象。分析这些寄存器与引脚 P2.7~P2.0、L8~L1 之间相互关系。

表 1-1 I/O 寄存器 P2OUT 和引脚 P2.7~P2.0 的关系

|            | L8~L1 的状态 | P2.7~P2.0 的逻辑值 |
|------------|-----------|----------------|
| P2OUT=0xA6 |           |                |
| P2OUT=0x9C |           |                |
| P2OUT=0x53 |           |                |
| P2OUT=0xB1 |           |                |

- 2) 在 1) 的基础上，如图 1-2，练习按位修改 P2OUT 的 D2、D5 的值，并观察引脚 P2.2、P2.5 连接的 LED 受 P2OUT 修改值的影响变化。用杜邦线将 P1.4 接蜂鸣器控制端 BUZZ，在 DEBUG 下需要操作哪些寄存器的哪些位，可以控制蜂鸣器的发声与否？



| Name            | Value | Description                   |
|-----------------|-------|-------------------------------|
| 1010 0101 P2OUT | 0xFB  | Port 2 Output [Memory Mapped] |
| 1010 0101 P7    | 1     | P7                            |
| 1010 0101 P6    | 1     | P6                            |
| 1010 0101 P5    | 1     | P5                            |
| 1010 0101 P4    | 1     | P4                            |
| 1010 0101 P3    | 1     | P3                            |
| 1010 0101 P2    | 0     | P2                            |
| 1010 0101 P1    | 1     | P1                            |
| 1010 0101 P0    | 1     | P0                            |


图 1-2 按位显示和修改端口寄存器的值

- 3) 修改寄存器 P1SEL=0x00，P1SEL2=0x00，P1DIR=0x00，P1REN=0xFF，P1OUT=0xFF，设置端口 P1 为基本输入功能，并置内部上拉电阻有效，按照表 1-2，按下相应的按键，打开 Register 窗口查看 P1IN 的值，可用按位查看的方式查看 P1IN 各位的值，如图 1-3。分析和理解这些寄存器与引脚 P1.7~P1.0、按键 K8~K1 之间相互关系。

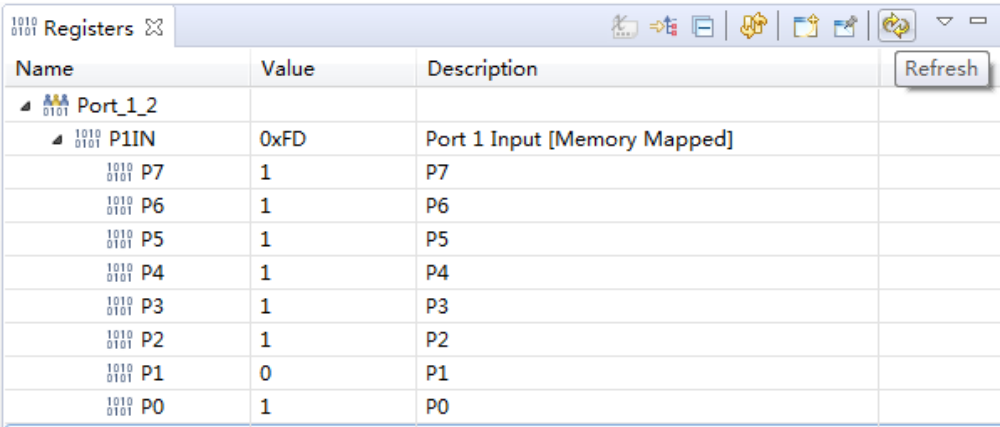
表 1-2 I/O 寄存器 P1IN 和引脚 P1.7~P1.0 的关系

| 操作          | P1IN 的值 |
|-------------|---------|
| 按下 K1       |         |
| 按下 K6       |         |
| 同时按下 K3， K5 |         |

**注意：**

- 1)在 Registers 窗口查看端口寄存器时，如查看 PxOUT、PxIN 时，如果窗口显示未及时更新，可如图 1-3，点击 Registers 窗口右上角处的 Refresh 图标 ，既可看到当前端口寄存器更新后的内容。

2)由于 CCS 对 PxIN 寄存器的显示刷新不及时，如果按刷新按钮，仍不能看到刷新结果，可按  
下实验板上按键，然后点击运行命令（resume），点击暂停，再在 View/registers 下查看 PxIN 寄  
存器中按键对应位的值，即操作了运行命令后，暂停，CCS 会刷新显示。



| Name     | Value | Description                  | Refresh |
|----------|-------|------------------------------|---------|
| Port_1_2 |       |                              |         |
| P1IN     | 0xFD  | Port 1 Input [Memory Mapped] |         |
| P7       | 1     | P7                           |         |
| P6       | 1     | P6                           |         |
| P5       | 1     | P5                           |         |
| P4       | 1     | P4                           |         |
| P3       | 1     | P3                           |         |
| P2       | 1     | P2                           |         |
| P1       | 0     | P1                           |         |
| P0       | 1     | P0                           |         |

图 1-3 点击刷新图标和按位方式查看 P1IN 内容

4.（提高）学习工程空间管理项目的方法

参看“附录 A CCS 使用入门”中的“八、在一个工程空间中管理多个项目”，完成在任务 2 建立的工程空间下，新建 Lab\_1 和 Lab\_2 两个项目，形成附录 A 中图 A-42 的工程空间，各项目文件夹的结构参见附录 A 中的图 A-43。多个项目情况下，鼠标左键双击预操作的项目，项目名将加黑加亮显示，如图 A-42 中的项目 **test**，此时可对选中的项目编译连接和 DEBUG。

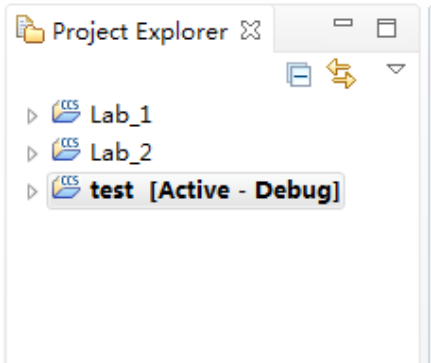


图 A-42 一个工程空间下可建多个项目

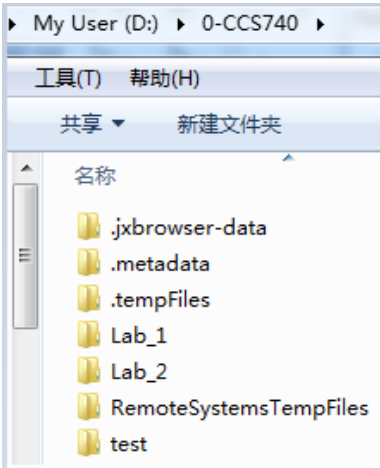


图 A-43 项目空间和项目文件夹结构图

建议在今后的实验中,为便于管理一学期的实验程序,可为每个实验建立一个 Lab\_xx 项目，  
**注意：不是为每个任务都建一个项目。** 实验时，在 Lab\_xx 项目下，采用 Add Files 添加和  
**Exclude from Build 移出**的方式做每次实验安排的各个任务,也就是在每个项目的文件夹下，  
可以有每次实验安排的各个任务的 C 源程序，如 lab2\_task1.C、 lab2\_task2.C、lab2\_task3.C  
等。但每次编译连接时，只有一个带 main()函数的 C 源程序在项目中。  
C 程序的源文件名可根据实验任务的内容用英文单词或英文单词的缩写进行命名。