

# 华大MCU端口使用时的竞争

产品名称	华大MCU端口使用时的竞争
公司名称	沈阳芯硕科技有限公司
价格	.00/个
规格参数	
公司地址	辽宁省沈阳市铁西区建设西路5号(14B13)
联系电话	024-85718900 19972041966

## 产品详情

在使用华大单片机时对GPIO操作是基础的操作，即使这种操作如果不注意还是会掉到坑里去。例如：使用同一组GPIO端口中的两个引脚（PA00和PA01）做输出，PA00在主循环中改变输出状态，PA01通过中断方式改变输出状态。正常的情况应该是PA00只在主循环中改变输出状态，而PA01只会在中断发生时改变输出状态。但是，随着程序运行时间的加长或者在主循环中提高PA00输出的频率，会发现本应该在中断中完成状态改变的PA01，个别时候状态会不发生改变。而在中断服务程序中设置断点，进行debug发现中断可以正常进入，也能正常改变PA01的输出状态。要想分析造成这个情况的原因可以从网上提供的DDL库入手来分析。华大单片机M0+系列芯片在对GPIO端口输出电平操作时，DDL库提供了如下两种方法：

方法1：

```
<p><font face="微软雅黑" size="3">/*****
```

```
** \brief GPIO IO输出值写入
```

```
**
```

```
** \param [in] enPort    IO Port口
```

```
** \param [in] enPin     IO Pin脚
```

```
** \param [out] bVal     输出值
```

```
** \retval en_result_t  Ok    设置成功
```

```
**
```

```
其他值 设置失败 *****/
```

```

en_result_t Gpio_WriteOutputIO(en_gpio_port_t enPort, en_gpio_pin_t enPin, boolean_t bVal)
{
    SetBit(((uint32_t)&M0P_GPIO->PAOUT + enPort), enPin, bVal);
    return Ok;
}

```

方法2：

```

** \brief GPIO IO设置
** \retval en_result_t    Ok    设置成功
**
其他值 设置失败 *****/

en_result_t Gpio_SetIO(en_gpio_port_t enPort, en_gpio_pin_t enPin)
{
    SetBit(((uint32_t)&M0P_GPIO->PABSET + enPort), enPin, TRUE);
    return Ok;
}

** \brief GPIO IO清零
**
其他值 设置失败 *****/

en_result_t Gpio_ClrIO(en_gpio_port_t enPort, en_gpio_pin_t enPin)
{
    SetBit(((uint32_t)&M0P_GPIO->PABCLR + enPort), enPin, TRUE);
    return Ok;
}

```

方法1是对整个PxOUT寄存器进行的操作，查看华大单片机用户手册关于此寄存器的说明；当PxOUT寄存器对应位为1时，对应的引脚输出高电平，反之输出低电平。  
方法2是通过置位寄存器引脚对应位的置1完成引脚输出高电平的操作，寄存器说明。

通过清零寄存器引脚对应位的置1完成引脚输出低电平的操作，寄存器说明

如图：

上述输出不正常的现象是因为使用了方法1进行的操作。在主循环中对PA00输出状态的改变通过PAOUT寄存器来完成，假如PORTA所有引脚都为低电平时，让PA00输出高电平，方法1的操作是把0X0001写入到PAOUT寄存器就可以实现。在ARM的汇编指令中要把0X0001写入到PAOUT必须借助于通用寄存器（r0~r7）来实现。当CPU刚完成0X0001移入到通用寄存器时，中断发生，CPU会把通用寄存器保存起来，然后响应中断，在中断中PA01输出高电平PAOUT值为0X0002，之后退出中断。退出中断后，CPU会恢复中断之前通用寄存器的值（0X0001），再继续把通用寄存器的值存入到PAOUT。这时PAOUT的值是0X0001，只有PA00输出高电平，而PA01没有输出高电平。这种现象就是华大MCU端口使用时的竞争-冒险现象。在使用华大芯片的时候不希望这竞争-冒险现象出现，通过方法2的操作完全可以避免此现象的出现。因为方法2是对寄存器的位进行操作，每次操作的时候只有对应的位进行置位或清零，其它位值为0的时候不影响输出的结果。建议在开发的时候对GPIO端口输出操作时，使用方法2的方式来操作。