



苏州大学

SOOCHOW UNIVERSITY



# 第4章 GPIO及程序框架

**4.1 通用I/O接口基本概念及连接方法**

**4.2 GPIO模块的编程结构**



苏州大学

SOOCHOW UNIVERSITY



## 4.1

# 通用I/O接口 基本概念与连接方法





## 4.1

### 通用I/O接口基本概念

#### I/O 接口

即为输入/输出（Input/Output）接口，是MCU与外界进行交互的重要桥梁。MCU与外界的数据交换需要借助I/O接口实现

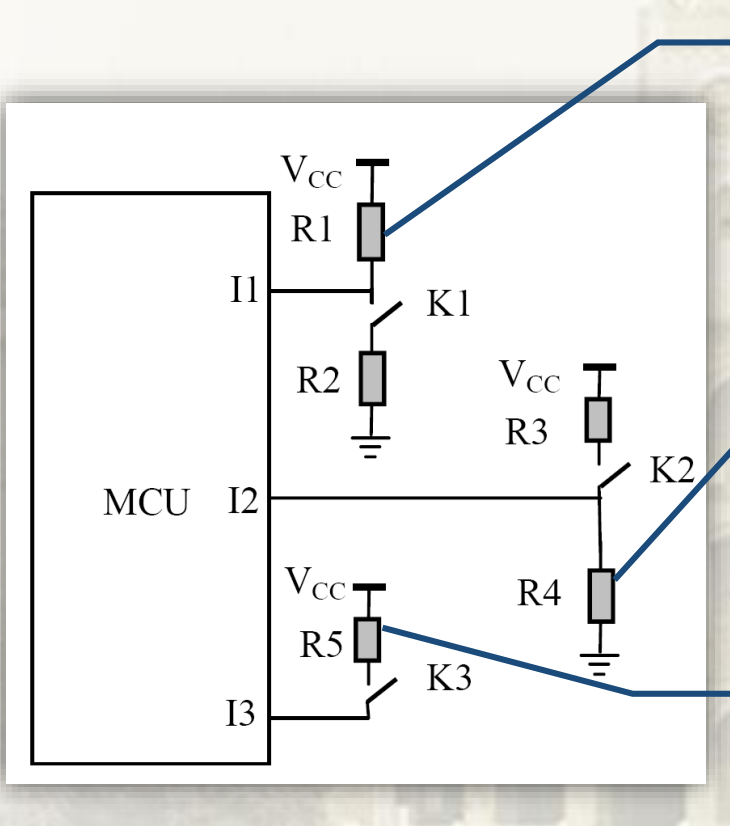
#### 通用 I/O

简称GPIO，是I/O的最基本形式。一般使用正逻辑，高电平对应数字信号“1”，低电平对应数字信号“0”  
引脚作为输入时，MCU通过端口寄存器获取引脚状态；引脚作为输出时，MCU通过内部寄存器控制引脚的输出状态



## 4.1

### 通用I/O接口的连接方法-输入引脚



上拉电阻的连接：通过一个电阻连接到电源（ $V_{CC}$ ），可以设置引脚初始电平为高电平。（ $R1 \gg R2$ ）

下拉电阻的连接：通过电阻连接到地（ $GND$ ）上，可设置引脚初始电平为低电平。（ $R3 \ll R4$ ）

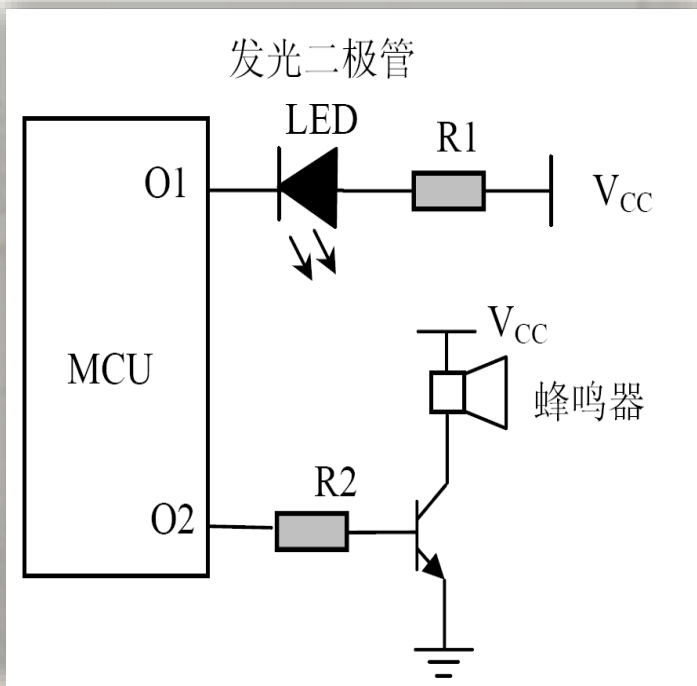
“悬空”连接：即不连接电阻，此时初始电平不确定，一般不使用

上拉电阻与下拉电阻的取值范围在  $1 \sim 10K\Omega$   
需要考虑静态电流与系统功耗



## 4.1

### 通用I/O接口的连接方法-输出引脚



MCU内部程序控制引脚输出高电平或低电平

不同的驱动电路对电流要求不同，使用的接法也不同

如O1与O2采用不同的方式驱动外部器件



苏州大学

SOOCHOW UNIVERSITY



**4.2**

## **GPIO模块的编程结构**







## 4.2 GPIO模块的编程结构-引脚概述

**MSP432P401R ( 100引脚封装 ) 的GPIO引脚分为11个端口，标记为1~10和J，共84个引脚。端口引脚作为 ( GPIO ) 引脚时，逻辑1对应高电平；逻辑0对应低电平**

- ( 1 ) 1口有8个引脚，分别记为P1.0~P1.7；
- ( 2 ) 2口有8个引脚，分别记为P2.0~P2.7；
- ( 3 ) 3口有8个引脚，分别记为P3.0~P3.7；
- ( 4 ) 4口有8个引脚，分别记为P4.0~P4.7；
- ( 5 ) 5口有8个引脚，分别记为P5.0~P5.7；
- ( 6 ) 6口有8个引脚，分别记为P6.0~P6.7；
- ( 7 ) 7口有8个引脚，分别记为P7.0~P7.7；
- ( 8 ) 8口有8个引脚，分别记为P8.0~P8.7；
- ( 9 ) 9口有8个引脚，分别记为P9.0~P9.7；
- ( 10 ) 10口有6个引脚，分别记为P10.0~P10.5；
- ( 11 ) J口有6个引脚，分别记为PJ.0~PJ.5；



## 4.2 GPIO模块的编程结构-寄存器地址分析

每个（GPIO）口均有12个寄存器，分别是：

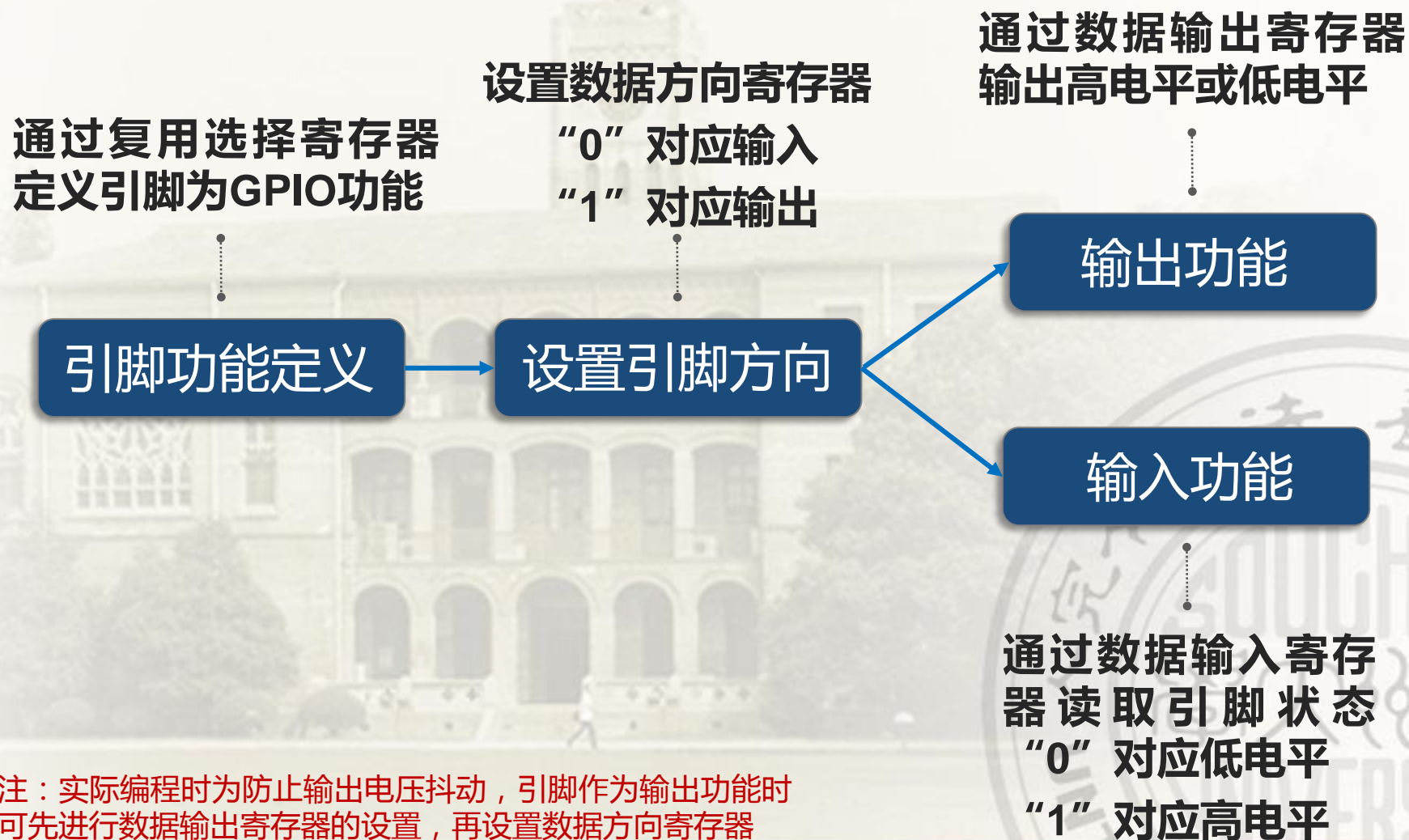
- |          |          |           |           |
|----------|----------|-----------|-----------|
| <b>1</b> | 数据输入寄存器  | <b>7</b>  | 数据输入寄存器   |
| <b>2</b> | 数据输出寄存器  | <b>8</b>  | 功能同变化寄存器  |
| <b>3</b> | 数据方向寄存器  | <b>9</b>  | 中断标志寄存器   |
| <b>4</b> | 上下拉使能寄存器 | <b>10</b> | 中断向量寄存器   |
| <b>5</b> | 驱动选择寄存器  | <b>11</b> | 中断边沿选择寄存器 |
| <b>6</b> | 复用选择寄存器0 | <b>12</b> | 终端使能寄存器   |





## 4.2

## GPIO模块的编程结构-基本编程步骤



注：实际编程时为防止输出电压抖动，引脚作为输出功能时可先进行数据输出寄存器的设置，再设置数据方向寄存器



## 4.2 GPIO模块的编程结构-基本打通程序

设P2口的2引脚接一只发光二极管，高电平点亮。如何点亮该发光二极管步骤如下：

1) 计算所需寄存器地址：寄存器地址=基地址+偏移量

P2端口的基地址	<code>vuint_8 *gpio2_ptr = (vuint_8*)0x40004C01u ;</code>
----------	---

数据方向寄存器	<code>vuint_8 *gpio2_PDIR = (vuint_8*)(gpio2_ptr+4);</code>
---------	---

数据输出寄存器	<code>vuint_8 *gpio2_POUT = (vuint_8*) (gpio2_ptr+2);</code>
---------	--

复用选择寄存器0	<code>vuint_8* gpio2_SEL0 = (vuint_8*) (gpio2_ptr+10);</code>
----------	---

复用选择寄存器1	<code>vuint_8* gpio2_SEL1 = (vuint_8*) (gpio_ptr+12);</code>
----------	--



## 4.2 GPIO模块的编程结构-基本打通程序

设P2口的2引脚接一只发光二极管，高电平点亮。如何点亮该发光二极管步骤如下：

- 1) 计算所需寄存器地址：寄存器地址=基地址+偏移量
- 2) 设置P2.2引脚为GPIO功能

### 程序步骤

将SEL0寄存器的第三位与SEL1寄存器的第三位设置为0，其余位保持

```
*gpio2_SEL0 &= ~(1<<2);  
*gpio2_SEL1 &= ~(1<<2);
```



## 4.2 GPIO模块的编程结构-基本打通程序

设P2口的2引脚接一只发光二极管，高电平点亮。如何点亮该发光二极管步骤如下：

- 1) 计算所需寄存器地址：寄存器地址=基地址+偏移量
- 2) 设置P2.2引脚为GPIO功能
- 3) 设置引脚输出高电平

### 程序步骤

将输出寄存器的相应位赋0，使P2.2引脚为输出状态时输出高电平

```
*gpio2_POUT &= (1<<2);
```

注：实际编程时为防止输出电压抖动，在设置为输出功能时可先进行数据输出寄存器的设置，再设置数据方向寄存器





## 4.2 GPIO模块的编程结构-基本打通程序

设P2口的2引脚接一只发光二极管，高电平点亮。如何点亮该发光二极管步骤如下：

- 1) 计算所需寄存器地址：寄存器地址=基地址+偏移量
- 2) 设置P2.2引脚为GPIO功能
- 3) 设置引脚输出高电平
- 4) 设置引脚为输出

### 程序步骤

通过令方向寄存器的相应位赋0，使P2.2引脚为输出状态

```
*gpio2_ptr->PDIR |= (1<<2);
```





## 4.2

## GPIO模块的编程结构-延伸讨论

### 构件封装的必要性

上述编程过程只是为了理解GPIO的基本编程方法，实际并不实用。芯片引脚较多，不可能一一对应编程。

将对底层硬件的操作以构件的形式封装起来，给出函数名与接口参数，供实际编程时使用，将会提高开发效率与稳定性。



苏州大学

SOOCHOW UNIVERSITY

arm

谢谢!

