

# STM32的IO口的8种配置详解 - 全文

来源：网络整理 作者： 2017年11月14日 13:13 关键词： [IO口](#) [STM32](#)

订阅

分享

0

STM32 IO口的8中配置方式解读（推挽输出、开漏输出、复用开漏输出、复用推挽输出以及上拉输入、下拉输入、浮空输入、模拟）

## STM32 IO口的8中配置方式：

(1) GPIO\_Mode\_AIN 模拟输入

(2) GPIO\_Mode\_IN\_FLOATING 浮空输入

(3) GPIO\_Mode\_IPD 下拉输入

(4) GPIO\_Mode\_IPU 上拉输入

(5) GPIO\_Mode\_Out\_OD 开漏输出

(6) GPIO\_Mode\_Out\_PP 推挽输出

(7) GPIO\_Mode\_AF\_OD 复用开漏输出

(8) GPIO\_Mode\_AF\_PP 复用推挽输出

I/O口的输出模式下。有3种输出速度可选（2MHz、10MHz和50MHz），这个速度是指I/O口驱动电路的响应速度而不是输出信号的速度，输出信号的速度与程序有关（芯片内部在I/O口的输出部分安排了多个响应速度不同的输出驱动电路，用户能够依据自己的须要选择合适的驱动电路）。通过选择速度来选择不同的输出驱动模块。达到最佳的噪声控制和减少功耗的目的。

高频的驱动电路。噪声也高，当不须要高的输出频率时，请选用低频驱动电路，这样非常有利于提高系统的EMI性能。

当然假设要输出较高频率的信号，但却选用了较低频率的驱动模块。非常可能会得到失真的输出信号。

关键是GPIO的引脚速度跟应用匹配（推荐10倍以上？）比方：

1、对于串口，假如最大波特率仅仅需115.2k，那么用2M的GPIO的引脚速度就够了，既省电也噪声小。

2、对于I2C接口。假如使用400k波特率，若想把余量留大些，那么用2M的GPIO的引脚速度也许不够，这时能够选用10M的GPIO引脚速度。

3、对于SPI接口，假如使用18M或9M波特率，用10M的GPIO的引脚速度显然不够了。须要选用50M的GPIO的引脚速度。

4、GPIO口设为输入时。输出驱动电路与port是断开。所以输出速度配置无意义。

5、在复位期间和刚复位后，复用功能未开启，I/Oport被配置成浮空输入模式。

6、全部port都有外部中断能力。为了使用外部中断线，port必须配置成输入模式。

7、GPIO口的配置具有上锁功能，当配置好GPIO口后，能够通过程序锁住配置组合，直到下次芯片复位才干解锁。

8、在STM32中怎样配置片内外设使用的IOport

首先，一个外设经过 ①配置输入的时钟和 ②初始化后即被激活（开启）；③假设使用该外设的输入输出管脚，则须要配置相应的GPIOport（否则该外设相应的输入输出管脚能够做普通GPIO管脚使用）；④再对外设进行具体配置。

## 相应到外设的输入输出功能有下述三种情况：

一、外设相应的管脚为输出：须要依据外围电路的配置选择相应的管脚为复用功能的推挽输出或复用功能的开漏输出。

二、外设相应的管脚为输入：则依据外围电路的配置能够选择浮空输入、带上拉输入或带下拉输入。

三、ADC相应的管脚：配置管脚为模拟输入。

假设把port配置成复用输出功能。则引脚和输出寄存器断开。并和片上外设的输出信号连接。将管脚配置成复用输出功能后，假设外设没有被激活，那么它的输出将不确定。

3 通用IOport（GPIO）初始化：

1）GPIO初始化

1、RCC\_APB2PeriphClockCmd（RCC\_APB2Periph\_GPIOA | B | C，ENABLE）：使能APB2总线外设时钟

2、RCC\_APB2PeriphResetCmd（RCC\_APB2Periph\_GPIOA | B | C，DISABLE）：释放GPIO复位

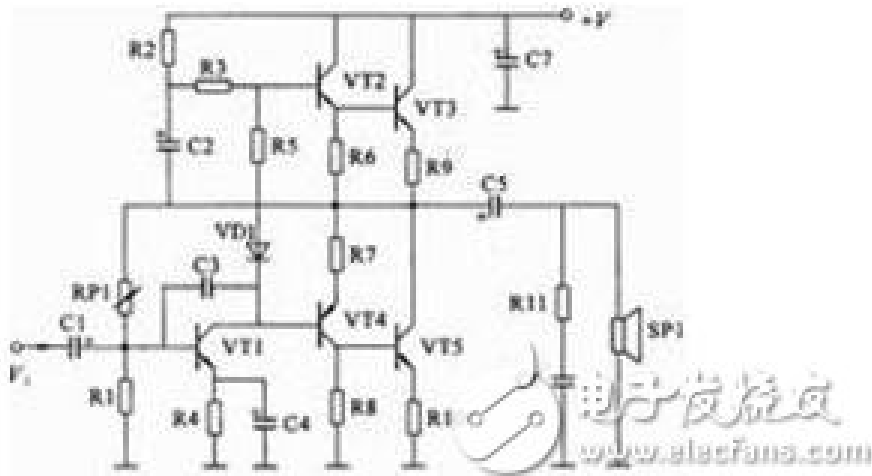
2）配置各个PINport（模拟输入\_AIN、输入浮空\_IN\_FLOATING、输入上拉\_IPU、输入下拉\_IPD、开漏输出\_OUT\_OD、推挽式输出\_OUT\_PP、推挽式复用输出\_AF\_PP、开漏复用输出\_AF\_OD）

3）GPIO初始化完毕

推挽输出：可以输出高，低电平，连接数字器件；推挽结构一般是指两个三极管分别受两互补信号的控制，总是在一个三极管导通的时候另一个截止。高低电平由IC的电源低定。

推挽电路是两个参数相同的三极管或MOSFET，以推挽方式存在于电路中，各负责正负半周的波形放大任务，电路工作时，两只对称的功率开关管每次只有一个导通，所以导通损耗小、效率高。输出既可以向负载灌电流，也可以从负载抽取电流。推拉式输出级既提高电路的负载能力，又提高开关速度。

详细理解：



如图所示，推挽放大器的输出级有两个“臂”（两组放大元件），一个“臂”的电流增加时，另一个“臂”的电流则减小，二者的状态轮流转换。对负载而言，好像是一个“臂”在推，一个“臂”在拉，共同完成电流输出任务。当输出高电平时，也就是下级负载门输入高电平时，输出端的电流将是下级门从本级电源经VT3拉出。这样一来，输出高低电平时，VT3一路和VT5一路将交替工作，从而减低了功耗，提高了每个管的承受能力。又由于不论走哪一路，管子导通电阻都很小，使RC常数很小，转变速度很快。因此，推拉式输出级既提高电路的负载能力，又提高开关速度。

开漏输出：输出端相当于三极管的集电极。要得到高电平状态需要上拉电阻才行。适合于做电流型的驱动，其吸收电流的能力相对强（一般20ma以内）。

## 开漏形式的电路有以下几个特点：

1.利用外部电路的驱动能力，减少IC内部的驱动。当IC内部MOSFET导通时，驱动电流是从外部的VCC流经R pull-up，MOSFET到GND。IC内部仅需很小的栅极驱动电流。

2.一般来说，开漏是用来连接不同电平的器件，匹配电平用的，因为开漏引脚不连接外部的上拉电阻时，只能输出低电平，如果需要同时具备输出高电平的功能，则需要接上拉电阻，很好的一个优点是通过改变上拉电源的电压，便可以改变传输电平。比如加上上拉电阻就可以提供TTL/CMOS电平输出等。（上拉电阻的阻值决定了逻辑电平转换的沿的速度。阻值越大，速度越低功耗越小，所以负载电阻的选择要兼顾功耗和速度。）

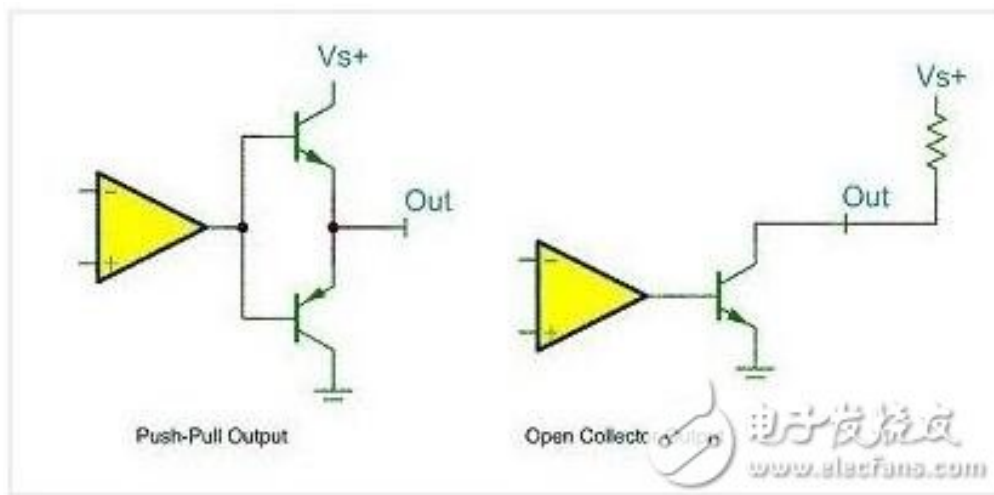
3.OPEN-DRAIN提供了灵活的输出方式，但是也有其弱点，就是带来上升沿的延时。因为上升沿是通过外接上拉无源电阻对负载充电，所以当电阻选择小时延时就小，但功耗大；反之延时大功耗小。所以如果对延时有要求，则建议用下降沿输出。

4.可以将多个开漏输出的Pin，连接到一条线上。通过一只上拉电阻，在不增加任何器件的情况下，形成“与逻辑”关系。这也是I2C，SMBus等总线判断总线占用状态的原理。补充：什么是“线与”？：

在一个结点（线）上，连接一个上拉电阻到电源VCC或VDD和n个NPN或NMOS晶体管的集电极C或漏极D，这些晶体管的发射极E或源极S都接到地线上，只要有一个晶体管饱和，这个结点（线）就被拉到地线电平上。因为这些晶体管的基极注入电流（NPN）或栅极加上高电平（NMOS），晶体管就会饱和，所以这些基极或栅极对这个结点（线）的关系是或非NOR逻辑。如果这个结点后面加一个反相器，就是或OR逻辑。

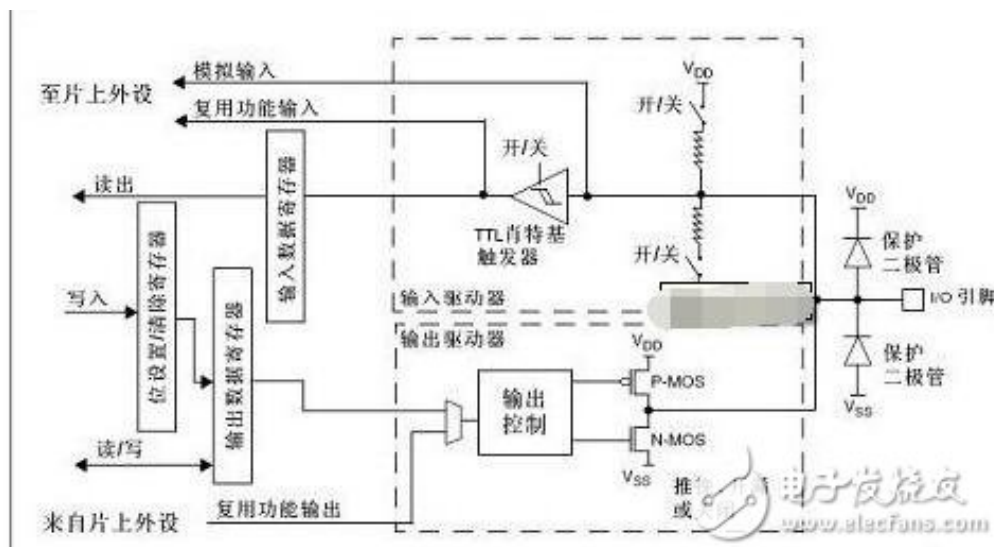
其实可以简单的理解为：在所有引脚连在一起时，外接一上拉电阻，如果有一个引脚输出为逻辑0，相当于接地，与之并联的回路“相当于被一根导线短路”，所以外电路逻辑电平便为0，只有都为高电平时，与的结果才为逻辑1。

关于推挽输出和开漏输出，最后用一幅最简单的图形来概括：



该图中左边的便是推挽输出模式，其中比较器输出高电平时下面的PNP三极管截止，而上面NPN三极管导通，输出电平VS+；当比较器输出低电平时则恰恰相反，PNP三极管导通，输出和地相连，为低电平。右边的则可以理解为开漏输出形式，需要接上拉。

浮空输入：对于浮空输入，一直没找到很权威的解释，只好从以下图中去理解了



由于浮空输入一般多用于外部按键输入，结合图上的输入部分电路，我理解为浮空输入状态下，IO的电平状态是不确定的，完全由外部输入决定，如果在该引脚悬空的情况下，读取该端口的电平是不确定的。

上拉输入/下拉输入/模拟输入：这几个概念很好理解，从字面便能轻易读懂。

复用开漏输出、复用推挽输出：可以理解为GPIO口被用作第二功能时的配置情况（即并非作为通用IO口使用）

## 最后总结下使用情况：

在STM32中选用IO模式

(1) 浮空输入\_IN\_FLOATING ——浮空输入，可以做KEY识别，RX1

(2) 带上拉输入\_IPU——IO内部上拉电阻输入

(3) 带下拉输入\_IPD—— IO内部下拉电阻输入

(4) 模拟输入\_AIN ——应用ADC模拟输入，或者低功耗下省电

(5) 开漏输出\_OUT\_OD ——IO输出0接GND，IO输出1，悬空，需要外接上拉电阻，才能实现输出高电平。当输出为1时，IO口的状态由上拉电阻拉高电平，但由于是开漏输出模式，这样IO口也就可以由外部电路改变为低电平或不变。可以读IO输入电平变化，实现C51的IO双向功能

(6) 推挽输出\_OUT\_PP ——IO输出0-接GND，IO输出1 -接VCC，读输入值是未知的

(7) 复用功能的推挽输出\_AF\_PP ——片内外设功能（I2C的SCL，SDA）

(8) 复用功能的开漏输出\_AF\_OD——片内外设功能（TX1，MOSI，MISO.SCK.SS）

## STM32设置实例：

(1) 模拟I2C使用开漏输出\_OUT\_OD，接上拉电阻，能够正确输出0和1；读值时先GPIO\_SetBits（GPIOB，GPIO\_Pin\_0）；拉高，然后可以读IO的值；使用GPIO\_ReadInputDataBit（GPIOB，GPIO\_Pin\_0）；

(2) 如果是无上拉电阻，IO默认是高电平；需要读取IO的值，可以使用带上拉输入\_IPU和浮空输入\_IN\_FLOATING和开漏输出\_OUT\_OD；

通常有5种方式使用某个引脚功能，它们的配置方式如下：

1) 作为普通GPIO输入：根据需要配置该引脚为浮空输入、带弱上拉输入或带弱下拉输入，同时不要使能该引脚对应的所有复用功能模块。

2) 作为普通GPIO输出：根据需要配置该引脚为推挽输出或开漏输出，同时不要使能该引脚对应的所有复用功能模块。

3) 作为普通模拟输入：配置该引脚为模拟输入模式，同时不要使能该引脚对应的所有复用功能模块。

4) 作为内置外设的输入：根据需要配置该引脚为浮空输入、带弱上拉输入或带弱下拉输入，同时使能该引脚对应的某个复用功能模块。

5) 作为内置外设的输出：根据需要配置该引脚为复用推挽输出或复用开漏输出，同时使能该引脚对应的所有复用功能模块。

注意如果有多个复用功能模块对应同一个引脚，只能使能其中之一，其它模块保持非使能状态。

比如要使用STM32F103VBT6的47、48脚的USART3功能，则需要配置47脚为复用推挽输出或复用开漏输出，配置48脚为某种输入模式，同时使能USART3并保持I2C2的非使能状态。

如果要使用STM32F103VBT6的47脚作为TIM2\_CH3，则需要对TIM2进行重映射，然后再按复用功能的方式配置对应引脚。

## GPIO模式配置

1、输入/输出模式（参考stm32手册）

2、GPIO输出模式下，几种速度的区别：

(1) GPIO 引脚速度：GPIO\_Speed\_2MHz（10MHz，50MHz）；

又称输出驱动电路的响应速度：（芯片内部在I/O口的输出部分安排了多个响应速度不同的输出驱动电路，用户可以根据自己的需要选择合适的驱动电路，通过选择速度来选择不同的输出驱动模块，达到最佳的噪声控制和降低功耗的目的。）

可理解为：输出驱动电路的带宽：即一个驱动电路可以不失真地通过信号的最大频率。

（如果一个信号的频率超过了驱动电路的响应速度，就有可能信号失真。失真因素？）

如果信号频率为10MHz，而你配置了2MHz的带宽，则10MHz的方波很可能就变成了正弦波。就好比是公路的设计时速，汽车速度低于设计时速时，可以平稳地运行，如果超过设计时速就会颠簸，甚至翻车。

关键是：GPIO的引脚速度跟应用相匹配，速度配置越高，噪声越大，功耗越大。

带宽速度高的驱动器耗电大、噪声也大，带宽低的驱动器耗电小、噪声也小。使用合适的驱动器可以降低功耗和噪声

比如：高频的驱动电路，噪声也高，当不需要高的输出频率时，请选用低频驱动电路，这样非常有利于提高系统的EMI性能。当然如果要输出较高频率的信号，但却选用了较低频率的驱动模块，很可能会得到失真的输出信号。关键是GPIO的引脚速度跟应用匹配（推荐10倍以上？）。

比如：

- ① USART串口，若最大波特率只需115.2k，那用2M的速度就够了，既省电也噪声小。
- ② I2C接口，若使用400k波特率，若想把余量留大些，可以选用10M的GPIO引脚速度。
- ③ SPI接口，若使用18M或9M波特率，需要选用50M的GPIO的引脚速度。

(2) GPIO的翻转速度指：输入/输出寄存器的0，1 值反映到外部引脚（APB2上）高低电平的速度。手册上指出GPIO最大翻转速度可达18MHz。

@通过简单的程序测试，用示波器观察到的翻转时间：是综合的时间，包括取指令的时间、指令执行的时间、指令执行后信号传递到寄存器的时间（这其中可能经过很多环节，比如AHB、APB、总线仲裁等），最后才是信号从寄存器传输到引脚所经历的时间。

如：有上拉电阻，其阻值越大，RC延时越大，即逻辑电平转换的速度越慢，功耗越大。

(3) .GPIO 输出速度：与程序有关，（程序中写的多久输出一个信号）。

2、GPIO口设为输入时，输出驱动电路与端口是断开，所以输出速度配置无意义。

3、在复位期间和刚复位后，复用功能未开启，I/O端口被配置成浮空输入模式。

4、所有端口都有外部中断能力。为了使用外部中断线，端口必须配置成输入模式。

5、GPIO口的配置具有上锁功能，当配置好GPIO口后，可以通过程序锁住配置组合，直到下次芯片复位才能解锁。

一般应用：

模拟输入\_AIN ——应用ADC模拟输入，或者低功耗下省电。

浮空输入\_IN\_FLOATING ——可以做KEY识别，RX1

开漏输出\_Out\_OD——应用于I2C总线；（STM32开漏输出若外部不接上拉电阻只能输出0）

管脚的复用功能 重映射

1、复用功能：内置外设是与I/O口共用引出管脚（不同的功能对应同一管脚）

STM32 所有内置外设的外部引脚都是与标准GPIO引脚复用的，如果有多个复用功能模块对应同一个引脚，只能使能其中之一，其它模块保持非使能状态。

2、重映射功能：复用功能的引出脚可以通过重映射，从不同的I/O管脚引出，即复用功能的引出脚位是可通程序改变到其他的引脚上！

直接好处：PCB电路板的设计人员可以在需要的情况下，不必把某些信号在板上绕一大圈完成联接，方便了PCB的设计同时潜在地减少了信号的交叉干扰。

如：USART1： 0： 没有重映像（TX/PA9，RX/PA10）； 1： 重映像（TX/PB6，RX/PB7）。

（参考AFIO\_MAPR寄存器介绍） [0, 1为一寄存器的bit值]

【注】 下述复用功能的引出脚具有重映射功能：

- 晶体振荡器的引脚在不接晶体时，可以作为普通I/O口
- CAN模块； - JTAG调试接口； - 大部分定时器的引出接口； - 大部分USART引出接口
- I2C1的引出接口； - SPI1的引出接口；

举例：对于STM32F103VBT6，47引脚为PB10，它的复用功能是I2C2\_SCL和 USART3\_TX，表示在上电之后它的默认功能为PB10，而I2C2的SCL和USART3的TX为它的复用功能；另外在TIM2的引脚重映射后，TIM2\_CH3也成为这个引脚的复用功能。

（1）要使用STM32F103VBT6的47、48脚的USART3功能，则需要配置47脚为复用推挽输出或复用开漏输出，配置48脚为某种输入模式，同时使能USART3并保持I2C2的非使能状态。

（2）使用STM32F103VBT6的47脚作为TIM2\_CH3，则需要对TIM2进行重映射，然后再按复用功能的方式配置对应引脚。

输入输出快速切换

```
#define HD7279_DAT_OUT GPIOB->CRH= (GPIOB->CRH& (~ (0x0000000F  
《《20) ) ) |0x00000003 《《20 //推挽输出
```

```
#define HD7279_DAT_IN GPIOB->CRH= (GPIOB->CRH& (~ (0x0000000F  
《《20) ) ) |0x00000004 《《20 //浮空输入
```