

# Introduce the definition of GPIO and the difference and implementation between different modes of GPIO

/ Control /MCU / Introduce the definition of GPIO and the difference and implementation between different modes of GPIO

---

September 11, 2022    admin    Control /MCU

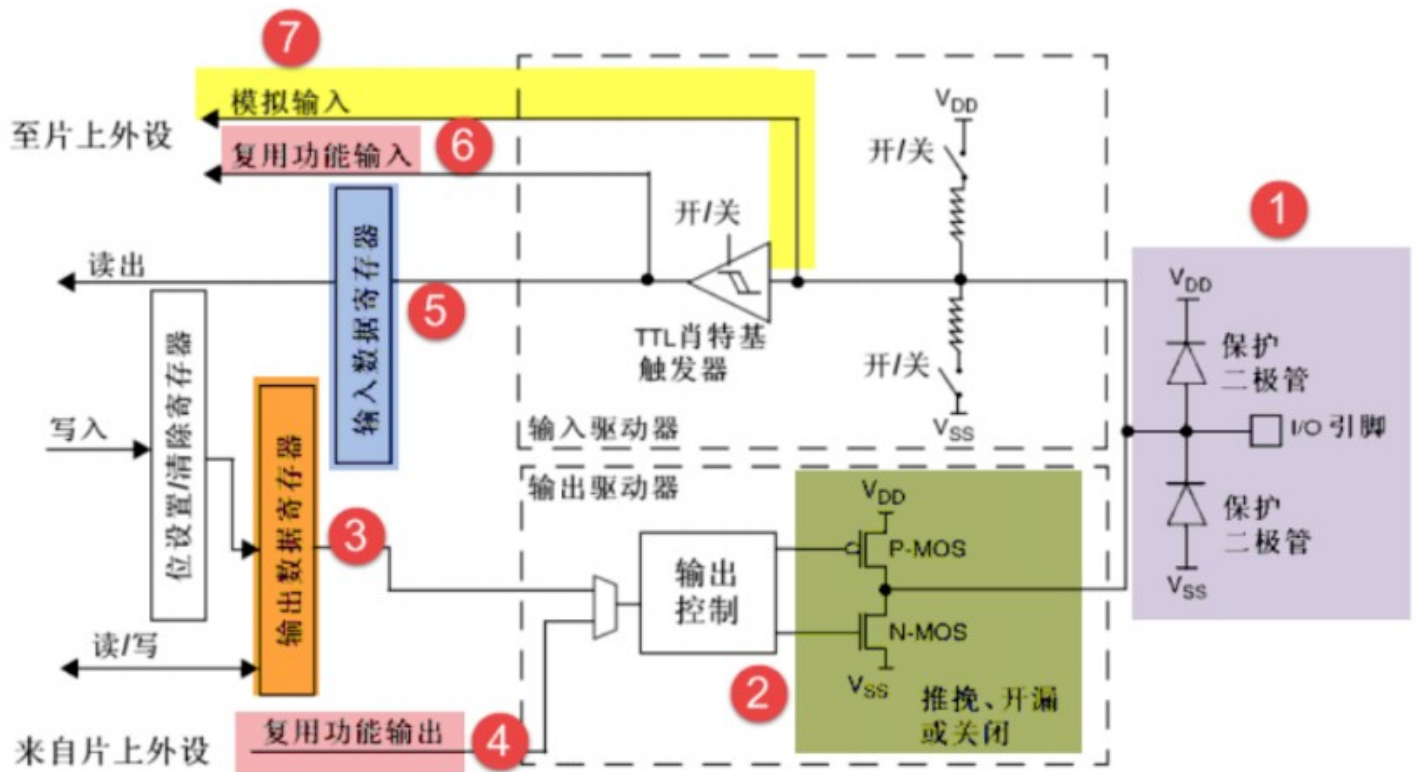
---

## I. Introduction

The full name of GPIO is General Purpose Input Output, which is general input/output. In fact, the essence of GPIO is a pin of the chip. Usually, all I/Os in ARM are general. However, since different peripheral circuits are designed on each development board, the functions of GPIO may be different. Most of the GPIOs have multiplexing functions. For example, some GPIOs may be TX or RX of the serial port, or they may be SCL or SDA line of I2C.

So we not only need to know that GPIO can output high and low levels, but also understand why some GPIOs can reuse certain functions, while others cannot.

## 2. Internal structure of GPIO



When we use GPIO, we may not think about why we can control the high and low levels of a pin by writing code or manipulating registers.

Today let's take a look at why we can control pin input or output by manipulating registers (in fact, the process of writing code is to manipulate registers).

If we want to control a GPIO port, we need to operate 7 registers, namely CRL, CRH, IDR, ODR, BRR, BSRR, LCKR. Our operation of GPIO is essentially to read and write these registers. The following are these The name of the register:

GPIOx\_CRL(x = A..E)端口配置低寄存器 32位寄存器  
GPIOx\_CRH(x = A..E)端口配置高寄存器 32位寄存器  
GPIOx\_IDR(x = A..E)端口输入数据寄存器 32位寄存器但仅用低16位  
GPIOx\_ODR(x = A..E)端口输出数据寄存器 32位寄存器但仅用低16位  
GPIOx\_BRR(x = A..E)端口位清除寄存器 16位寄存器  
GPIOx\_BSRR(x = A..E)端口位设置/清除寄存器 16位寄存器  
GPIOx\_LCKR(x = A..E)端口配置锁定寄存器 32位寄存器

We first analyze the above structural circuit:

**1. Protection diode:** The protection diode is not difficult to think from its name that it is used to protect the system. The conduction of the two diodes can prevent the external input voltage of the pin from being too low or too high. When the voltage is too high, the upper protection diode conducts. When the voltage is too low, the diode below is turned on to prevent the abnormal voltage from being introduced into the chip and causing the chip to burn.

**2. P MOS tube and N MOS tube:** After being protected by two diodes, GPIO flows upward into the input mode and downward into the output mode, and the control of the output mode is a unit circuit composed of a P MOS tube and an N MOS tube, which mainly controls the output mode. , planned by the structure dog to your unit circuit with **Push-pull output** and **Open drain output** Two modes.

**When the system is configured in push-pull output mode:**

**When the system is configured in open-drain output mode:**

**Push-pull output mode** Typically used when output levels are 0 and 3.3V and require **When switching state at high speed**. **Open-drain output mode** Generally used in I2C, SMBUS communication, etc. **line with function in the bus circuit.**

In addition to the occasions where the open-drain mode must be used in the microcontroller, the push-pull output mode is generally used.

**3. Output data register:** We know earlier that the input signal of the double MOS tube structure circuit is provided by the GPIO output data register GPIOx\_ODR, so we can modify the output level of the GPIO pin by modifying the value of the output data register. The set/reset register GPIOx\_BSRR can affect the output of the circuit by modifying the value of the output data register.

**4. Multiplexing function output:** Multiplexing in multiplexing function output means that other on-chip peripherals of STM32 control the GPIO pins. At this time, the GPIO pins are used as part of the peripheral function, which is regarded as the second purpose. The multiplexed function output signal drawn from other peripherals and the data register of the GPIO itself are connected to the input of the dual MOS tube structure.

### 3. Input and output mode

GPIO has a total of 8 input and output modes, namely: pull-up input, pull-down input, floating input, analog input, open-drain output, push-pull output, open-drain multiplexed output, and push-pull multiplexed output. The above eight input and output modes.

**Pull up input:**GPIO\_Mode\_IPU

**Drop down input:**GPIO\_Mode\_IPD

**Floating input:**GPIO\_Mode\_IN\_FLOATING

**Analog input:**GPIO\_Mode\_AIN

**Open-drain output:**GPIO\_Mode\_Out\_OD

**Push-pull output:**GPIO\_Mode\_Out\_PP

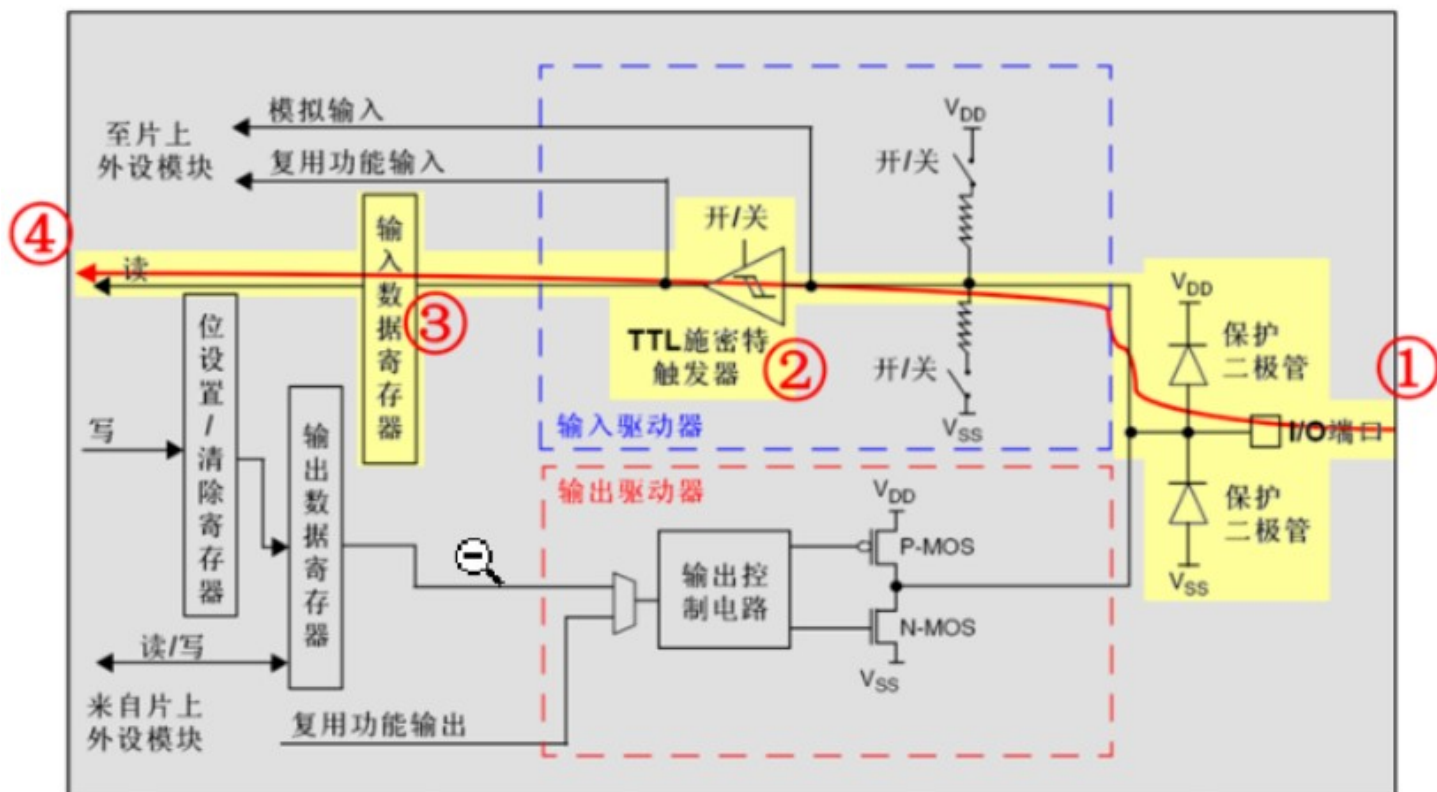
**Open-drain multiplexing:**GPIO\_Mode\_AF\_OD

**Push and reuse:**GPIO\_Mode\_AF\_PP

Here are a few common patterns in detail:

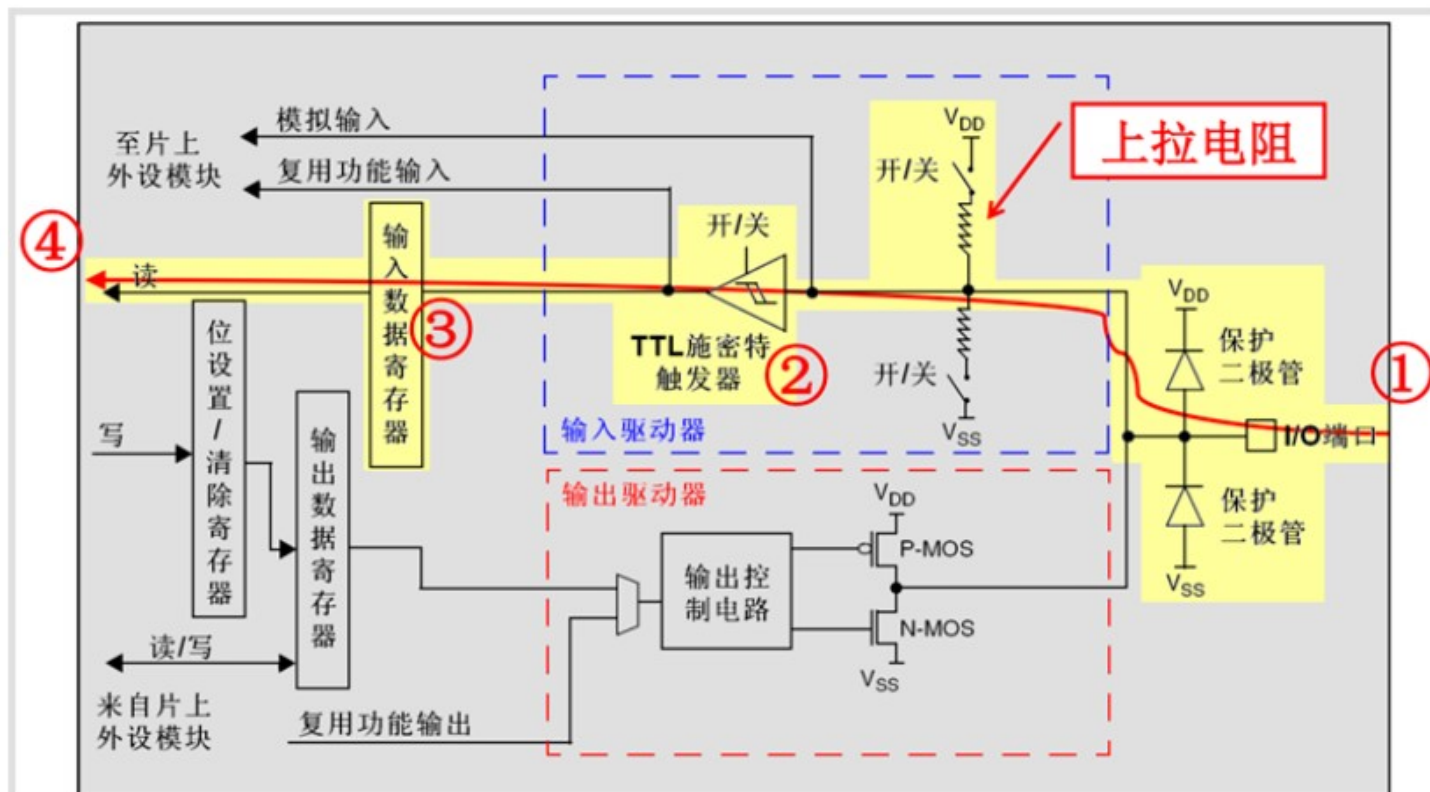
#### 3.1 Floating input

In floating input mode, the level signal of the I/O port directly enters the input data register. That is to say, the level state of the I/O is indeterminate and completely determined by the external input; if the pin is left floating (in the absence of signal input), the level of reading the port is indeterminate, Usually used in IIC, USART and other bus devices.



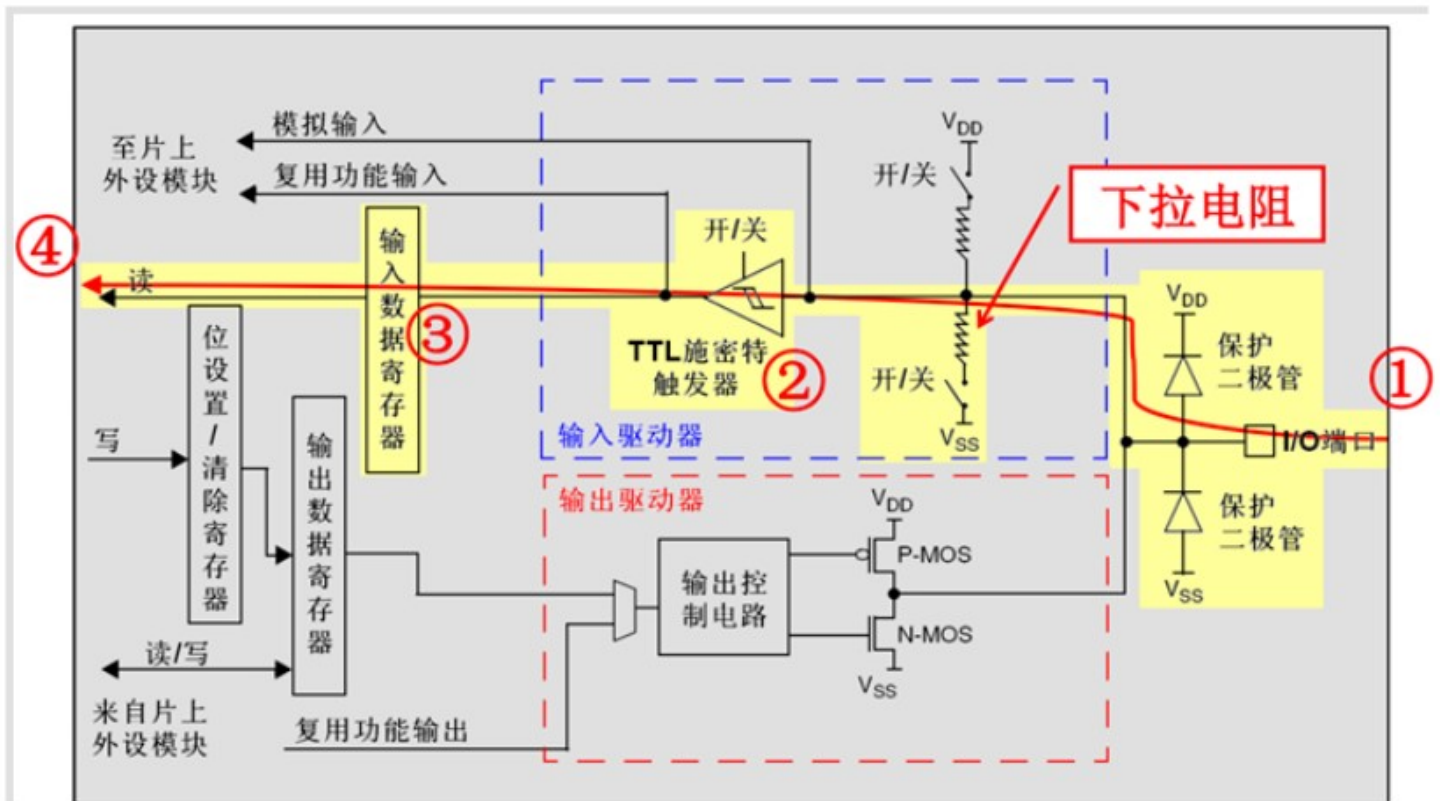
### 3.2 Pull-up input mode

In the pull-up input mode, the level signal of the I/O port directly enters the input data register. However, when the I/O port is floating (in the absence of signal input), the level of the input terminal remains high (and when the input of the I/O port is low, the level of the input terminal is also low)



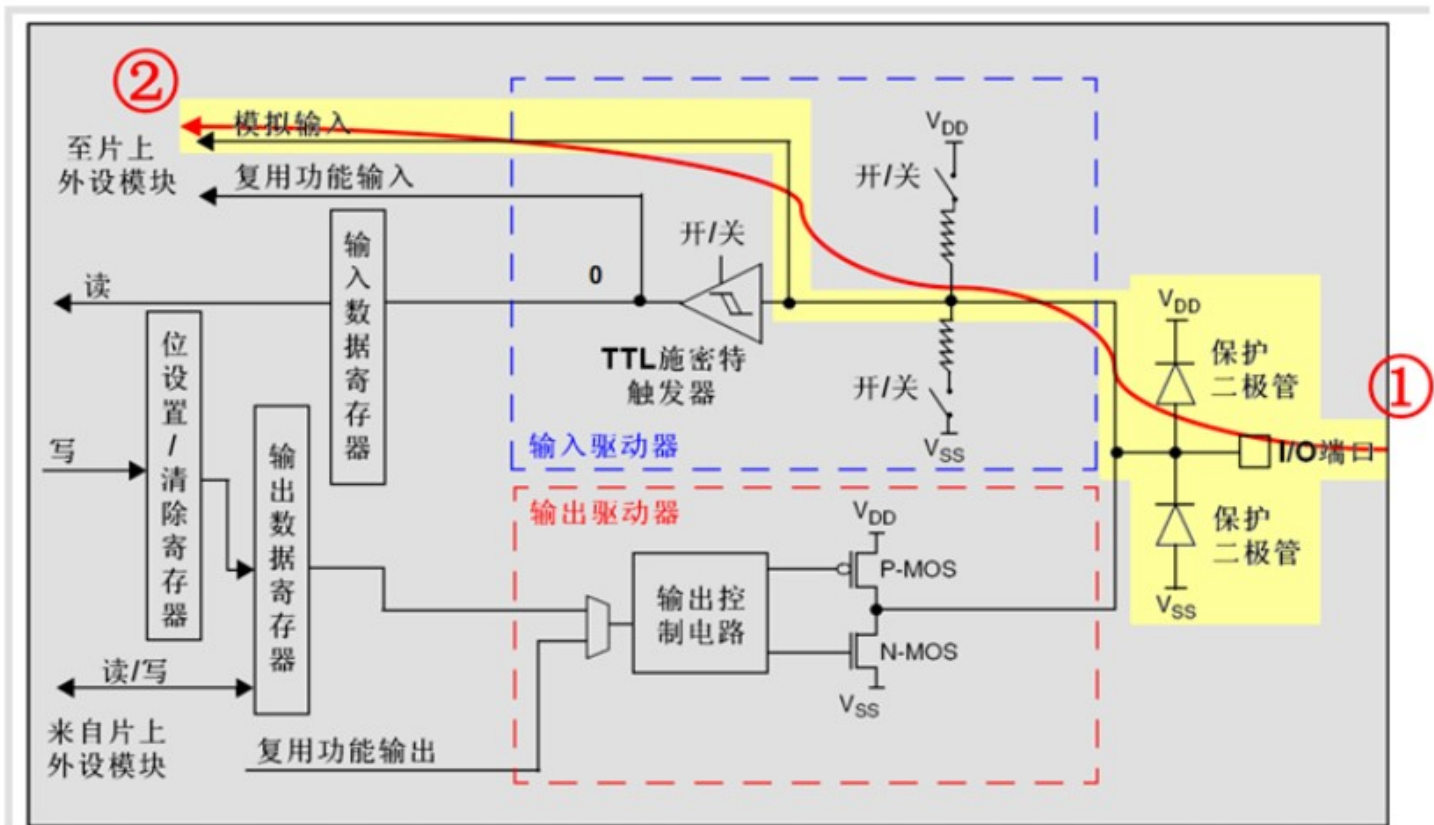
### 3.3 Drop-down input mode

In the pull-down input mode, the level signal of the I/O port directly enters the input data register. However, when the I/O port is floating (no signal input), the level of the input terminal remains low; and when the input of the I/O port is high, the level of the input terminal is also high.



### 3.4 Analog Input Mode

In the analog input mode, the analog signal (voltage signal, not level signal) of the I/O port is directly analog input to the on-chip peripheral modules, such as ADC modules.



### 3.5 Open-Drain Output Mode

See the detailed introduction in the above two;

### 3.6 Open-Drain Multiplexed Output Mode

The open-drain multiplexed output mode is very similar to the open-drain output mode. It is only the source of the high and low levels of the output. Instead of letting the CPU directly write the output data register, it is determined by the multiplexing function output of the on-chip peripheral module. It is generally used for on-chip peripheral functions: TX1, MOSI, MISO, SCK, SS

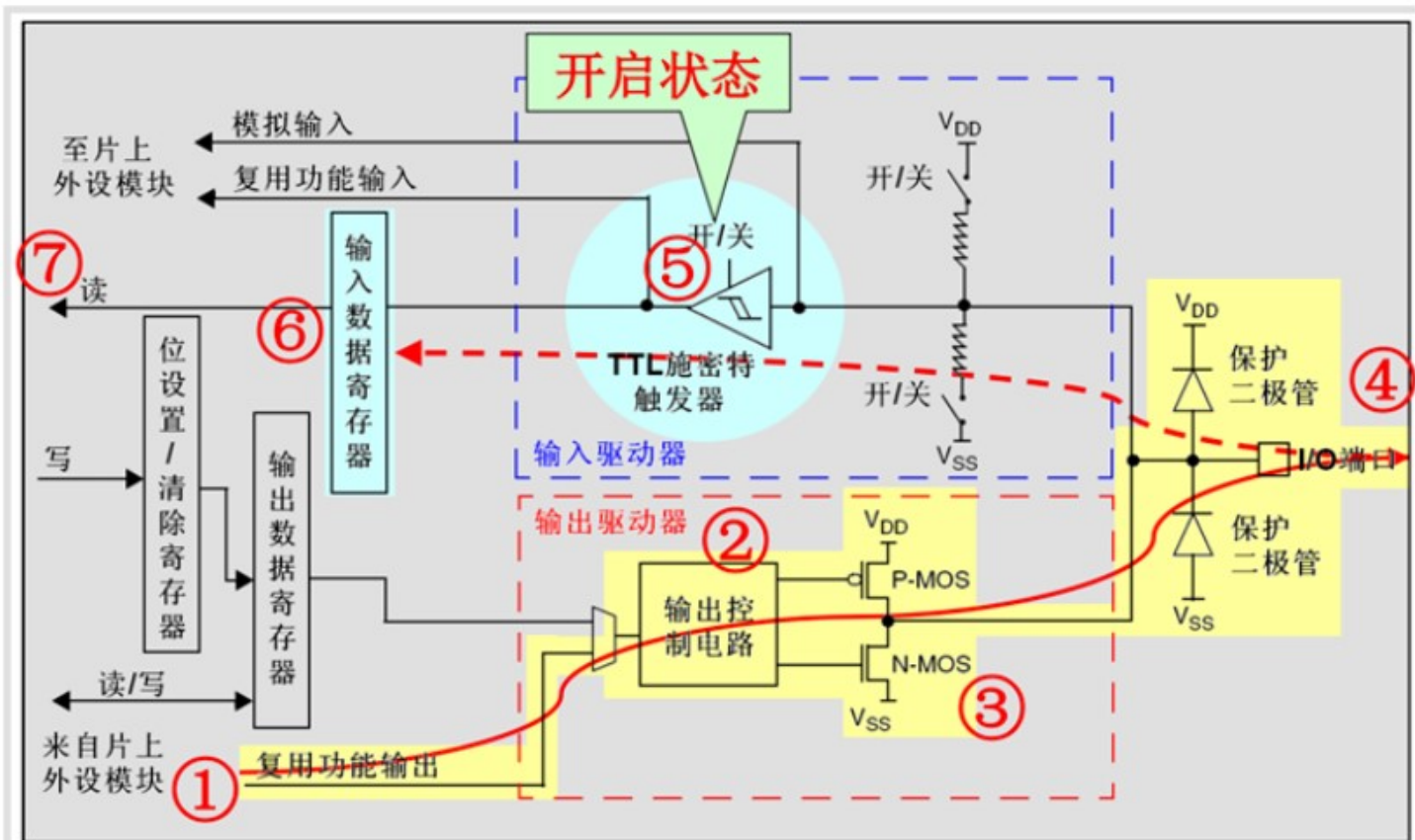
### 3.7 Push-Pull Output Mode

See the detailed introduction in the above two;

### 3.8 Push-Pull Multiplexed Output Mode

The push-pull multiplexed output mode is very similar to the push-pull output mode. It is only the source of the high and low levels of the output. It is not to let the CPU directly write the output data register, but to use the multiplexing function output of the on-chip peripheral module to determine. The common push-pull output mode is generally the SCL of the on-chip peripheral function IIC. , SDL.





#### Fourth, the difference between open-drain output and push-pull output

The difference between open-drain output and push-pull output mode is mainly open-drain output. **Only a strong low level can be output, and a high level must be pulled high by an external resistor.** The output terminal is equivalent to the collector of the triode, which is suitable for current-type driving, and its ability to absorb current is relatively strong (generally within 20ma); push-pull output **Can output strong high and low level**, connect the digital device.



	推挽输出	开漏输出
高电平驱动能力	强	由外部上拉电阻提供
低电平驱动能力	强	强
电平跳变速度	快	由外部上拉电阻决定，电阻越小，反应越快，功耗越大
线与功能	不支持	支持
电平转换	不支持	支持

## 5. Configuration of GPIO in STM32

In the development of the firmware library, the operation of the registers CRH and CRL to configure the mode and speed of the IO port is done through the GPIO initialization function:

```
void GPIO_Init(GPIO_TypeDef* GPIOx, GPIO_InitTypeDef* GPIO_InitStruct);
/*第一个参数是用来指定 GPIO，取值范围为 GPIOA~GPIOG。
 *第二个参数为初始化参数结构体指针，结构体类型为 GPIO_InitTypeDef
 */
```

The usual format for initializing a GPIO is:

```
GPIO_InitTypeDef GPIO_InitStructure;
```

```
GPIO_InitTypeDef GPIO_InitStructure;

GPIO_InitStructure.GPIO_Pin = GPIO_Pin_5; //LED0-- PB.5 端口配置

GPIO_InitStructure.GPIO_Mode = GPIO_Mode_Out_PP; //推挽输出

GPIO_InitStructure.GPIO_Speed = GPIO_Speed_50MHz; //速度 50MHz

GPIO_Init(GPIOB, GPIO_InitStructure); //根据设定参数配置 GPIO
```

There are three optional values for the IO port speed setting:

```
typedef enum  
  
{  
  
GPIO_Speed_10MHz,  
  
GPIO_Speed_2MHz,  
  
GPIO_Speed_50MHz  
  
}GPIOSpeed_TypeDef;
```

The mode has 8 optional definitions as follows:

```

typedef enum

{
    GPIO_Mode_AIN = 0x0, //模拟输入

    GPIO_Mode_IN_FLOATING = 0x04, //浮空输入

    GPIO_Mode_IPD = 0x28, //下拉输入

    GPIO_Mode_IPU = 0x48, //上拉输入

    GPIO_Mode_Out_OD = 0x14, //开漏输出

    GPIO_Mode_Out_PP = 0x10, //通用推挽输出

    GPIO_Mode_AF_OD = 0x1C, //复用开漏输出

    GPIO_Mode_AF_PP = 0x18 //复用推挽

}GPIO_Mode_TypeDef;

```

## 6. Summary

This article mainly introduces the definition of GPIO and the difference and implementation between different modes of GPIO. After reading this article, I believe you will have a deeper understanding of GPIO. If we are software developers, we don't need to pay too much attention to how GPIO is implemented, but We need to know the characteristics and applications of GPIO in each mode, only in this way can we better configure the most suitable one in practical applications.

If the control output is 1 (it cannot directly output a high level), both the P-MOS and N-MOS tubes are turned off, so **in open-drain output mode, the pin outputs neither high level nor low level, and is in a high-impedance state.**

If we control the output to be 0, low level, the P-MOS tube is turned off, the N-MOS tube is turned on, and the output is grounded;

The P-MOS tube above does not work at all;

When the pin is switched high and low, the two MOS tubes are turned on in turn, the P tube is responsible for sinking current, and the N tube is responsible for sourcing current, so that the load capacity and switching speed are greatly improved compared to the ordinary method. \*\*The low level of the push-pull output is 0V and the high level is 3.3V\*\*.

If a low level is input into this structure, after the reverse, the lower N-MOS transistor is turned on, the upper P-MOS is turned off, and a low level is output externally.

If a high level is input into this structure, after the reverse, the upper P-MOS is turned on, the lower N-MOS is turned off, and a high level is output to the outside.

Reviewing Editor: Liu Qing

Post Views:  
7

Tagged: [GPIO](#) [Low level](#) [MOS tube](#)

## Related Articles

Apte won the "IC Unicorn" Award

CI520 card reader chip is compatible  
with CV520 and supports

### Leave a Reply

What are the main features of the  
BL32F32x2 optical module  
microcontroller

ISO/IEC14443  
Setup time and hold time analysis of  
static timing  
Jihai Semiconductor was successfully  
selected as a Chinese IC unicorn  
enterprise

Your email address will not be  
published. Required fields are  
marked \*

Why can the microcontroller directly  
program the program?