# 中断编程顺序

一、使能中断请求

二、配置中断优先级分组

三、配置NVIC寄存器，初始化NVIC\_InTypeDef

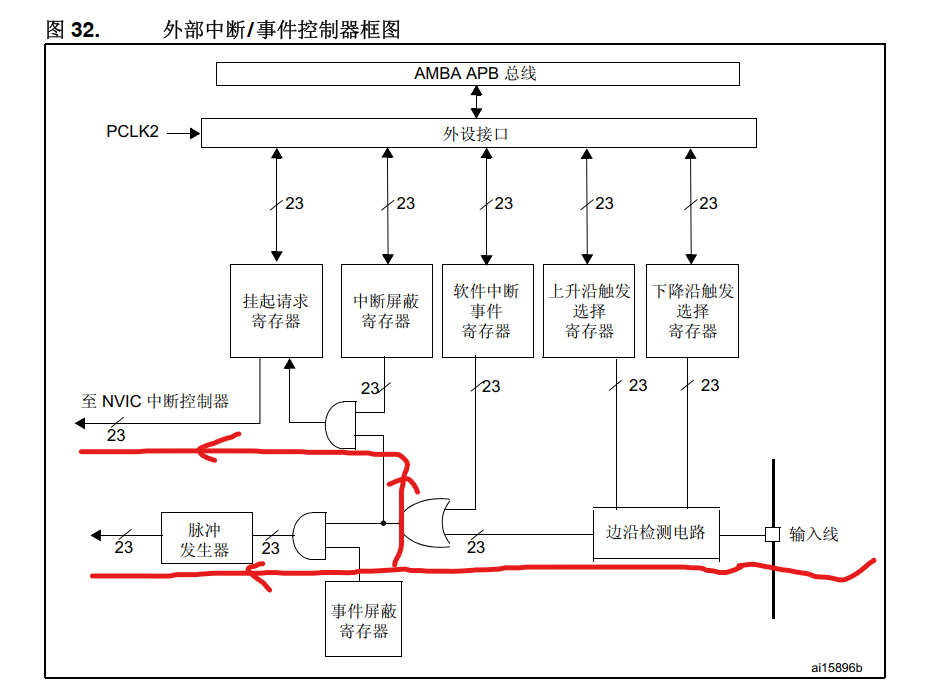
NVIC\_IRQChannel：中断源

NVIC\_IRQChannelPreemptionPriority：抢占优先级

NVIC\_IRQChannelSubPriority：子优先级

NVIC\_IRQChannelCmd：使能或者失能

四、编写中断服务函数



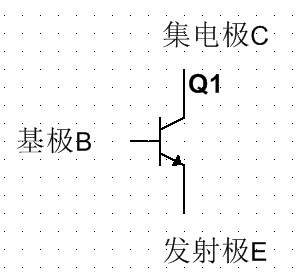
# 基本定时器

功能简介：

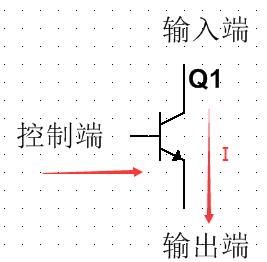
1. 只有TIM6和TIM7，只能向上计数，计数器16bit，挂载在APB1总线上
2. 没有外部的GPIO，是内部资源，只能用来定时
3. 时钟来自PCLK1，可实现1~~65535分频

# 三极管

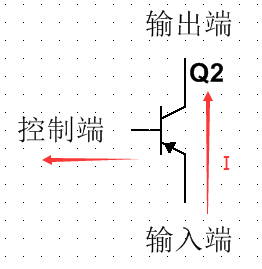
把基极B视为**控制端**，集电极C视为**输入端**，发射极E视为**输出端**。这里输入输出是指电流流动的方向。



NPN型晶体管：只要基极B有输入电流，就有电流从集电极进入并从发射极输出。

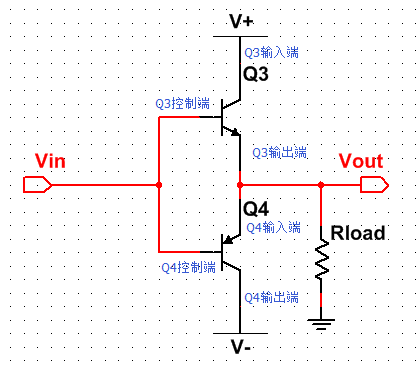


PNP型晶体管：当有电流从基极B流出时，就会有电流从集电极流到发射极。

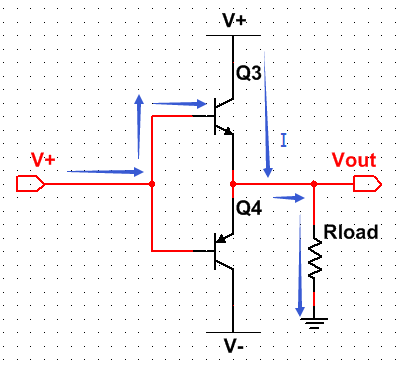


# 推挽输出

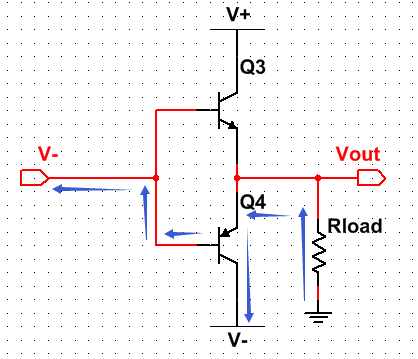
推挽输出电路图：



上面为**N型**三极管，下面为**P型**三极管。



经过上面的N型三极管提供电流给负载，这就叫「**推**」



经过下面的P型三极管提供电流给负载，这就叫「**挽**」。

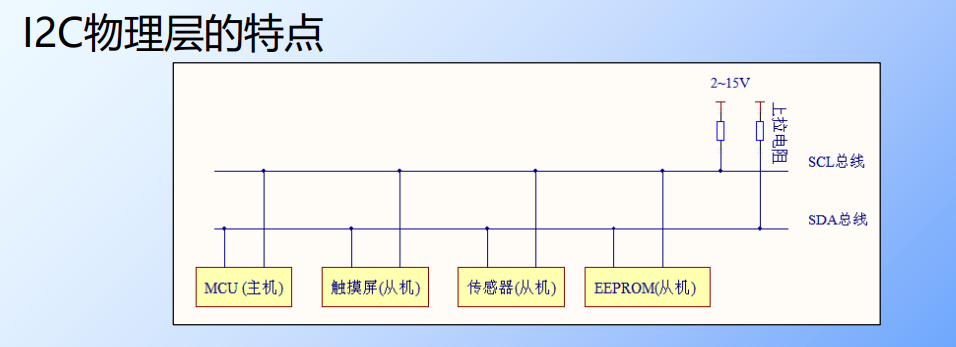
# ADC

**独立模式-单通道-中断（不使用DMA）**

1. 初始化ADC的GPIO（单通道配置一个即可）
2. 配置ADC初始化结构体
3. 配置通道的转换顺序、配置中断、打开ADC、触发ADC开始转换
4. 编写main函数，中断服务函数

# I2C-读写EEPROM

***I2C物理层的特点***



• 它是一个支持多设备的总线。“总线”指多个设备共用的信号线。 在一个I2C通讯总线中，可连接多个I2C通讯设备，支持多个通讯主机及多个通讯从机。

• 一个I2C总线只使用两条8总线线路，一条双向串行数据线(SDA)，一条串行时钟线 (SCL)。数据线即用来表示数据，时钟线用于数据收发同步。

• 每个连接到总线的设备都有一个独立的地址，主机可以利用这个地址进行不同设备之间的访问。

• 总线通过上拉电阻接到电源。当I2C设备空闲时，会输出高阻态，而当所有设备都空闲，都输出高阻态时，由上拉电阻把总线拉成高电平。

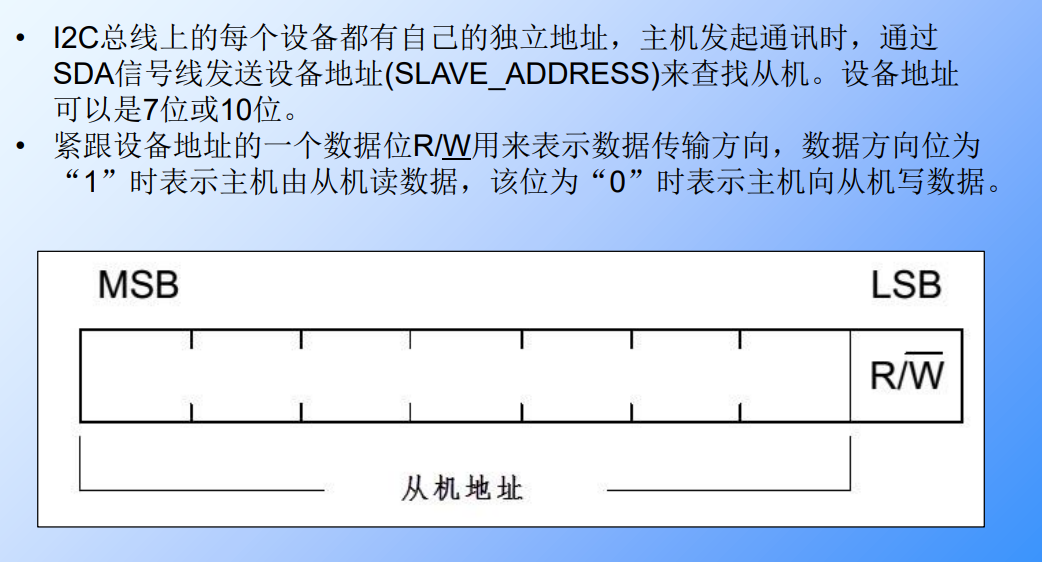
• 多个主机同时使用总线时，为了防止数据冲突，会利用仲裁方式决定由哪个设备占用总线。

• 具有三种传输模式：标准模式传输速率为100kbit/s ，快速模式为 400kbit/s，高速模式下可达3.4Mbit/s，但目前大多I 2C设备尚不支持高速模式。

• 连接到相同总线的 IC 数量受到总线的最大电容400pF限制。

***I2C的协议层***

I2C的协议定义了通讯的起始和停止信号、数据有效性、响应、 仲裁、时钟同步和地址广播等环节。



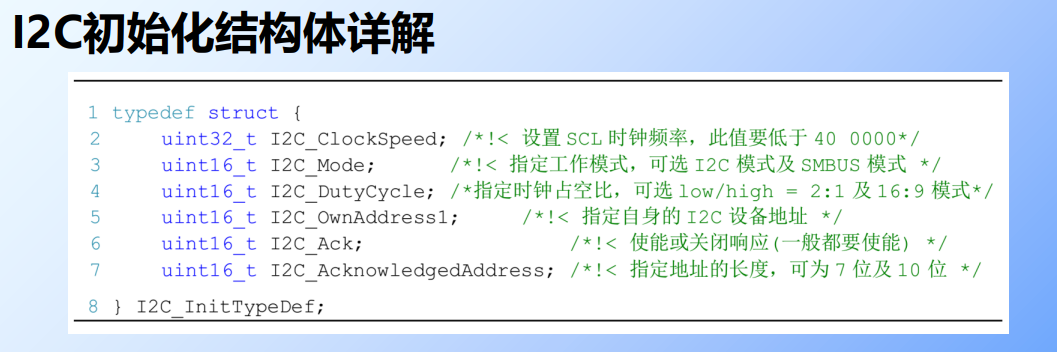
7位设备地址：000 1011 （11）

8位的设备写地址：000 1011 0 （22）

8位的设备读地址：000 1011 1 （23）

8r\_addr = (7addr << 1) | 1 （读）

8w\_addr = (7addr << 1) | 0 （写）



- I2C\_Ack 当设备作为从机时，若没有使能，主机不会收到响应信号。

- I2C\_AcknowledgeAddress STM32添加的额外功能，规定设备地址长度7位或10位。

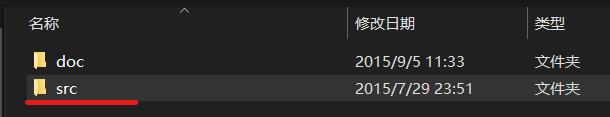
EEPROM写入功能

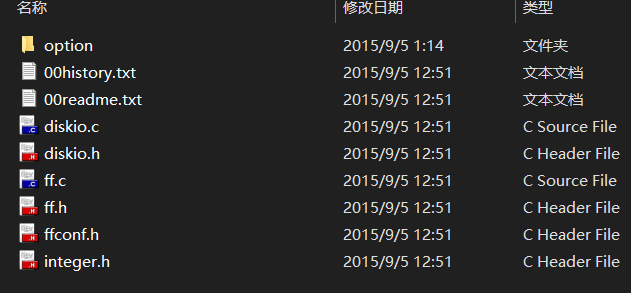
写入一个字节：stm32向EEPROM发送问询信号，得到应答后再次寻址，接着写入一个数据。

页写入：步骤同上，但写入一个数据后可接着写入，一次最多写入八个字节的数据，假如数据溢出将会覆盖之前写入的数据。

# 文件管理系统FATFS

使用的是ff 11a





integer.h 文件中包含了一些数值类型定义

diskio.c 包含底层存储介质的操作函数，这些函数需要用户自己实现，主要添加底层驱动函数

ff.c FatFs核心文件，文件管理的实现方法。该文件独立于底层介质操作文件的函数，利用这些函数实现文件的读写。

Cc936.c 本文件在option目录下，是简体中文支持所需要添加的文件，包含了简体中文的GBK和Unicode相互转换功能函数。

ffconf.h 这个头文件包含了对FatFs功能配置的宏定义，通过修改这些宏定义就可以裁剪FatFs的功能。如需要支持简体中文，需要把ffconf.h中的\_CODE\_PAGE 的宏改成936并把上面的cc936.c文件加入到工程之中。