

什么是STM32

- 1、ST— 意法半导体，是一个公司名，即SOC厂商
- 2、M— Microelectronics的缩写，表示微控制器，大家注意微控制器和微处理器的区别
- 3、32— 32bit的意思，表示这是一个32bit的微控制器

诞生背景：技术更替，市场需求，ST的努力

STM32能做什么

STM32属于一个微控制器，自带了各种常用通信接口，功能非常强大

- 1、串口—USART，用于跟串口接口的设备通信，比如：USB转串口模块、ESP8266 WIFI、GPS模块，GSM 模块，串口屏、指纹识别模块 STM32属于一个微控制器，自带了各种常用通信接口，功能非常强大
- 2、内部集成电路—I2C(Inter-Integrated Circuit)，用于跟I2C接口的设备通信，比如：EEPROM、电容屏、陀螺仪MPU6050、0.96寸OLED模块
- 3、串行通信接口—SPI(Serial Peripheral Interface)，用于跟SPI接口的设备通信，比如：串行FLASH、以太网 W5500、音频模块VS1053
- 4、SDIO、FSMC的超级、I2S、SAI、ADC、GPIO

STM32怎么选型

STM32分类

表格 5-2 STM8 和 STM32 分类

CPU 位数	内核	系列	描述
32	Cortex-M0	STM32-F0	入门级
		STM32-L0	低功耗
	Cortex-M3	STM32-F1	基础型，主频 72M
		STM32-F2	高性能
		STM32-L1	低功耗
	Cortex-M4	STM32-F3	混和信号
		STM32-F4	高性能，主频 168M
		STM32-L4	低功耗
	Cortex-M7	STM32-F7	高性能
8	超级版 6502	STM8S	标准系列
		STM8AF	标准系列的汽车应用
		STM8AL	低功耗的汽车应用
		STM8L	低功耗

stm32命名法

—	ST M32	F	103	R	C	T	6
家族	STM32 表示 32bit 的 MCU						
产品类型	F 表示基础型						
具体特性	基础型						
引脚数目	R 表示 64pin, 其他常用的为 C 表示 48, V 表示 100, Z 表示 144, Z 表示 144, B 表示 208, N 表示 216						
FLASH 大小	C 表示 256KB, 其他常用的为 B 表示 128, E 表示 512, I 表示 2048						
封装	T 表示 QFP 封装, 这个是最常用的封装						
温度	6 表示温度等级为 A: -40~85°						

STM32

F

051

R

8

T

6

X

XX

Family

STM32 32-bit MCUs

STM8 8-bit MCUs

Product type

A Automotive

F Foundation

L Ultra-low power

S Standard

T Touch sensing

W Wireless

xP Fastrom

Specific features (3 digits)

(Depends of product series
None exhaustive list)

STM32x ...

051 Entry-level

103 STM32 foundation

303 103 upgraded with DSP and Analog

407 High-performance and DSP with FPU

152 Ultra-low-power

STM8x .../STM8Ax ...

103 Mainstream access line

F52 Automotive CAN

L31 Automotive low-end

Code size (Kbytes)

0	1
1	2
2	4
3	8
4	16
5	24
6	32
7	48
8	64
9	72
A	96 or 128*
B	128
Z	192
C	256
D	384
E	512
F	768
G	1024
H	1536
I	2048

Note:

* For STM8A only

Package

B Plastic DIP*

D Ceramic DIP*

G Ceramic QFP

H LFBGA /TFBGA

I UFBGA Pitch 0.5**

J UFBGA Pitch 0.8**

K UFBGA Pitch 0.65**

M Plastic SO

P TSSOP

Q Plastic QFP

T QFP

U UFQFPN

V VFQFPN

Y WLCSP

* Dual-in-Line package

** For new product serie only for existing product marketing serie please use H letter

Firmware Royalties

U Universal Not for production (Sampling and tools)

V MP3 decoder

W MP3 Codec

J 0.80 mm

D IS2T JAVA

Option

xxx Fastrom code

or

xTR Tape and Real

Dxx No RTC (STM8L)

Dxx BOR OFF with Special bonding + Boot standard

Dxx BOR OFF with Boot I2CS (Special)

Sxx BOR OFF

bx BOR ON

No Letter BOR ON + Boot standard

or

Yxx Die rev (Y)

Pin count (pins for STM8 and STM32)

D 14 pins	C 48 & 49 pins	A 169 pins
Y 20 pins (STM8)	U 63 pins	I 176 & 201 (176+25) pins
F 20 pins (STM32)	R 64 & 66 pins	B 208 pins
E 24 & 25 pins	J 72 pins	N 216 pins
G 28 pins	M 80 pins	X 256 pins
K 32 pins	O 90 pins	Auto
T 36 pins	V 100 pins	8 48
H 40 pins	Q 132 pins	9 64
S 44 pins	Z 144 pins	A 80

Temperature range (°C)

6 and A - 40 to + 85

7 and B - 40 to + 105

3 and C - 40 to + 125

D - 40 to + 150

选择合适的MCU

- 1、选择哪种内核的芯片，内核越高意味着功耗也越高
- 2、选择多少引脚的芯片，引脚多少决定了资源的多少，也影响价格
- 3、选择多少RAM和FLASH的芯片，FLASH越大，价格越贵
- 4、还要考虑所选型号采购是否容易，供货是否稳定

分配原理图引脚

引脚分类	引脚说明说明
电源	(VBAT)、(VDD VSS)、(VDDA VSSA)、(VREF+ VREF-)等
晶振 IO	主晶振 IO，RTC 晶振 IO
下载 IO	用于 JTAG 下载的 IO：JTMS、JTCK、JTDI、JTDO、NJTRST
BOOT IO	BOOT0、BOOT1，用于设置系统的启动方式
复位 IO	NRST，用于外部复位
上面 5 部分 IO 组成的系统我们也叫做最小系统	
GPIO	专用器件接到专用的总线，比如 I2C，SPI，SDIO，FSMC，DCMI 这些总线的器件需要接到专用的 IO
	普通的元器件接到 GPIO，比如蜂鸣器，LED，按键等元器件用普通的 GPIO 即
	如果还有剩下的 IO，可根据项目需要引出或者不引出

手册	主要内容	说明
参考手册	片上外设的功能说明和寄存器描述	对片上每一个外设的功能和使用做了详细的说明，包含寄存器的详细描述。编程的时候需要反复查询这个手册。
数据手册	功能概览	主要讲这个芯片有哪些功能，属于概括性的介绍。芯片选型的时候首先看这个部分。
	引脚说明	详细描述每一个引脚的功能，设计原理图的时候和写程序的时候需要参考这部分。
	内存映射	讲解该芯片的内存映射，列举每个总线的地址和包含有哪些外设。
	封装特性	讲解芯片的封装，包含每个引脚的长度宽度等，我们画 PCB 封装的时候需要参考这部分的参数。

数据手册中对引脚的定义

Table 5. High-density STM32F103xx pin definitions											
Pins 1						Pin name 2	Type ⁽¹⁾ 3	I/O Level 4	Main function ⁽³⁾ (after reset) 5	Alternate functions ⁽⁴⁾ 6	
LFPGA144	LFPGA100	WLCSP64	LQFP64	LQFP100	LQFP144					Default	Remap 7
A3	A3	-	-	1	1	PE2	I/O	FT	PE2	TRACECK/FSMC_A23	
A2	B3	-	-	2	2	PE3	I/O	FT	PE3	TRACED0/FSMC_A19	
B2	C3	-	-	3	3	PE4	I/O	FT	PE4	TRACED1/FSMC_A20	
B3	D3	-	-	4	4	PE5	I/O	FT	PE5	TRACED2/FSMC_A21	
B4	E3	-	-	5	5	PE6	I/O	FT	PE6	TRACED3/FSMC_A22	
C2	B2	C6	1	6	6	V _{BAT}	S		V _{BAT}		

引脚的功能定义解读

名称	缩写	说明
① 引脚序号	阿拉伯数字表示 LQFP 封装，英文字母开头的表示 BGA 封装。引脚序号这里列出了有 8 种封装型号，具体使用哪一种要根据实际情况来选择。	
② 引脚名称	指复位状态下的引脚名称	
③ 引脚类型	S	电源引脚
	I	输入引脚
	I/O	输入/输出引脚
④ I/O 结构	FT	兼容 5V
	TTa	只支持 3V3，且直接到 ADC
	B	BOOT 引脚
	RST	复位引脚，内部带弱上拉
⑤ 注意事项	对某些 IO 要注意的事项的特别说明	
⑥ 复用功能	IO 的复用功能，过 GPIOx_AFR 寄存器来配置选择。一个 IO 口可以复用为多个功能，即一脚多用，这个在设计原理图和编程的时候要灵活选择。	
⑦ 额外功能	IO 的额外功能，通过直连的外设寄存器配置来选择。个人觉得在使用上跟复用功能差不多。	

开始分配原理图IO

确定MCU型号，封装形式，在数据手册上找到封装的引脚定义，根据引脚序号，复制整理成 excel表。

置位1用或，置位0用与（加取反）。

GPIO=general purpose input output通用输入输出端口

定义的是结构体指针的话，访问里面的元素就用p->结构体成员；或者用(*p).结构体成员`

定义的是结构体的话，访问里面的元素就用p.结构体成员就可以

用keil编辑STM32点灯—遇到灯常亮不闪烁的问题

- options-c/c++选项 更改优化级别为-O0
- while循环中出现的变量，定义时加上关键字volatile

STM32固件库文件分析

1-汇编编写的启动文件

startup-stm32f10x_hd.s：设置堆栈指针，设置PC指针、初始化中断向量表、配置系统时钟、对用C库函数_main最终去到c的世界

2-时钟配置文件

system_stm32f10x.c: 把外部时钟HSE=8M, 经过PLL倍频为72M

3-外设相关的

stm32f10x.h: 实现了内核之外的外设寄存器映射

xxx: GPIO、USART、I2C、SPI、FSMC

stm32f10x_XXX.c: 片上外设的驱动函数库文件

stm32f10x_XXX.h: 存放外设的初始化结构体, 外设初始化结构体成员的参数列表, 外设固件库函数的声明

4-内核相关的

CMSIS - Cortex微控制器软件接口标准

core_cm3.h: 实现了内核里面外设的寄存器映射

core_cm3.c: 内核外设的驱动固件库

NVIC (嵌套向量中断控制器)、SysTick (系统滴答定时器)

misc.h

misc.c

5-头文件的配置文件

stm32f10x_conf.h: 头文件的头文件

6-专门存放中断服务函数的C文件

stm32f10x_it.h

stm32f10x_it.c

中断服务函数可以随意放在其他地方, 并不是一定要放在stm32f10x_it.c

^异或,C语言的一个二进制运算符

与1异或改变，与0异或不变

0x00到0x01是一个字节的变化，字节Byte，比特bit

！ GPIO点亮LED灯编程要点

- 使能GPIO端口时钟
- 初始化GPIO目标引脚为推挽输出模式、
- 编写主函数，控制GPIO引脚输出高低电平

！ GPIO按键检测编程要点

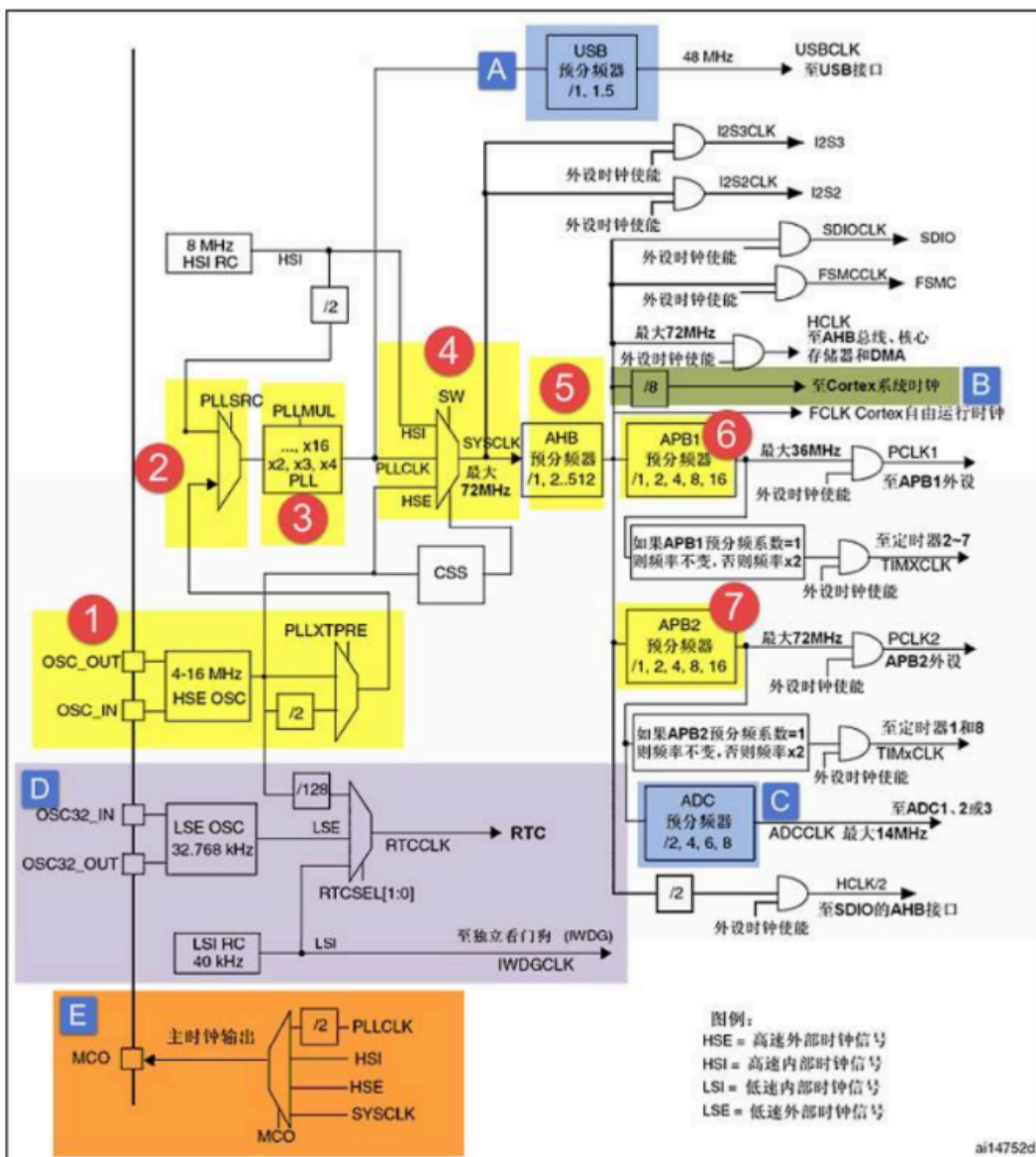
- 使能GPIO端口时钟
- 初始化GPIO目标引脚为输入模式（浮空输入）
- 编写主函数，检测按键状态，实现按键控制LED灯

启动文件代码做了什么

1. 初始化堆栈指针
2. 初始化中断向量表
3. 执行复位程序
4. 中断服务程序
5. 用户堆栈初始化

RCC相关头文件和固件库源文件

1. 时钟使能配置
2. 时钟源相关配置
3. 分频系数选择配置
4. 外设时钟使能
5. 其他外设时钟配置
6. 状态参数获取函数
7. RCC中断相关函数



ai14752d

！ 使用HSE配置时钟编程要点

1. 使能HSE/HSI，并等待其稳定
2. 设置AHB、APB2、APB1预分频因子
3. 设置PLL时钟来源，设置PLL倍频因子
4. 使能PLL并等待PLL稳定
5. 选择PLL作为系统时钟来源

6. 读取时钟切换状态位，确保PLLCLK被选为系统时钟

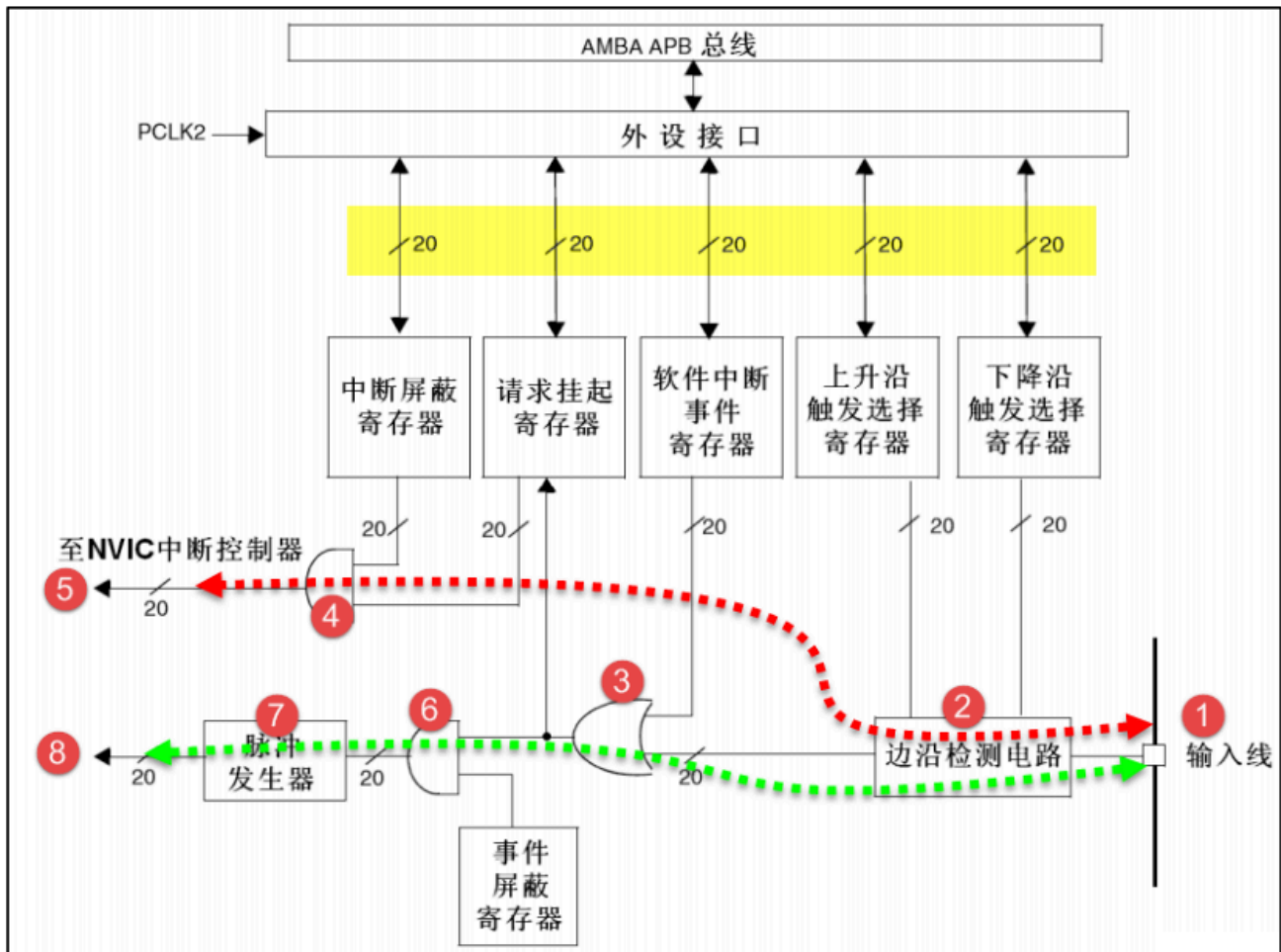
中断应用概览

优先级的设定，要先分组，再配置主优先级和次优先级

中断源从stm32f10x.h的IRQn_Type结构体里面找

中断服务函数名称的编写，要和启动文件里的中断向量表名称要一样，能确保函数的执行，而不是跳到预先写好的空的中断服务函数一直在循环。

EXTI功能框图



！ EXTI外部中断控制实验编程要点

建议对按键和EXTI进行宏定义

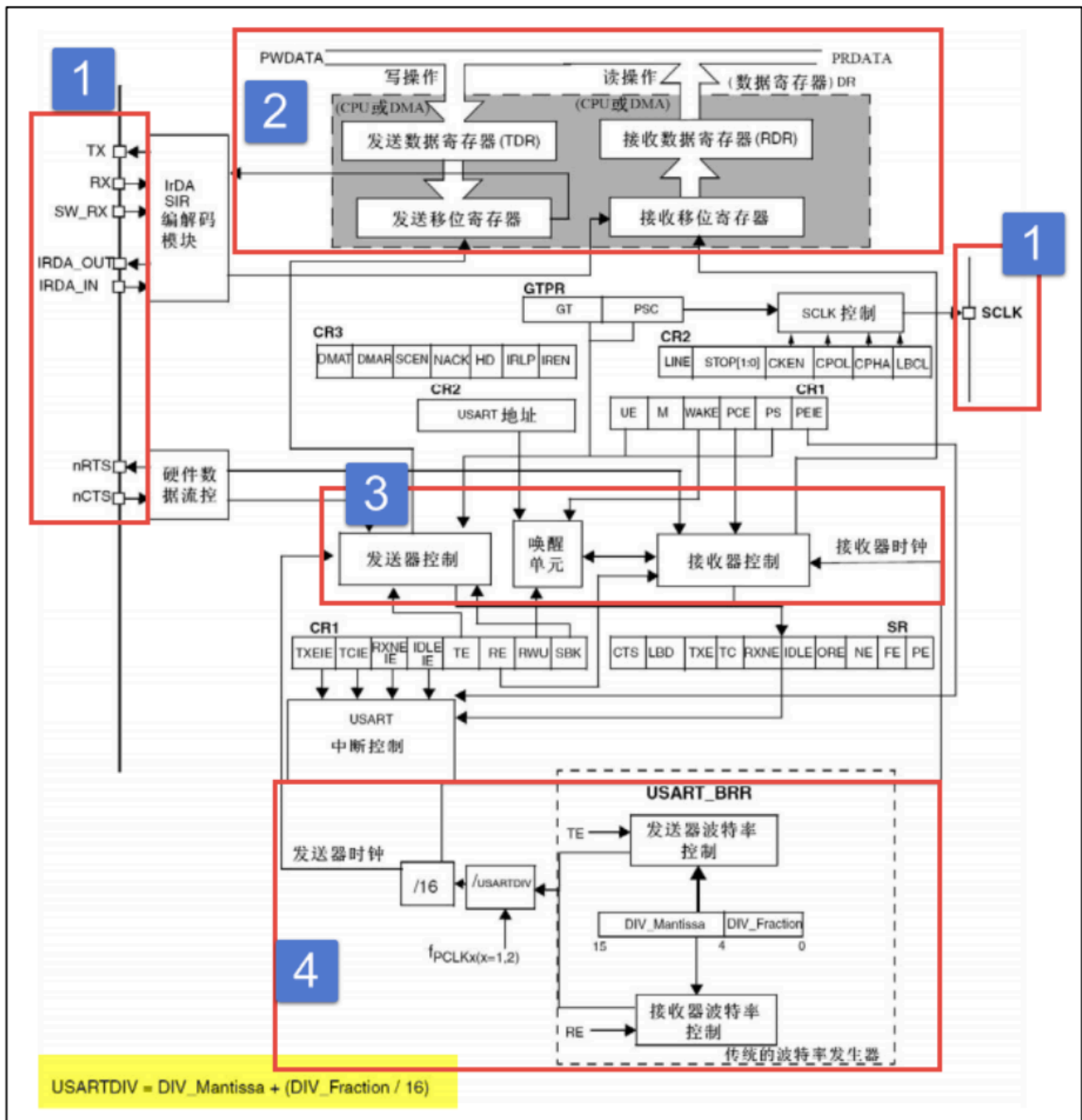
初始化结构体和初始化库函数配合使用是标准库精髓所在

1. 初始化用来产生中断的GPIO
2. 初始化EXTI
3. 配置NVIC（嵌套向量中断控制器）
4. 编写中断服务函数

！ SysTick定时实验编程要点

1. 设置重装载寄存器的值LOAD
2. 清除当前数值寄存器的值VAL
3. 配置控制与状态寄存器CTRL

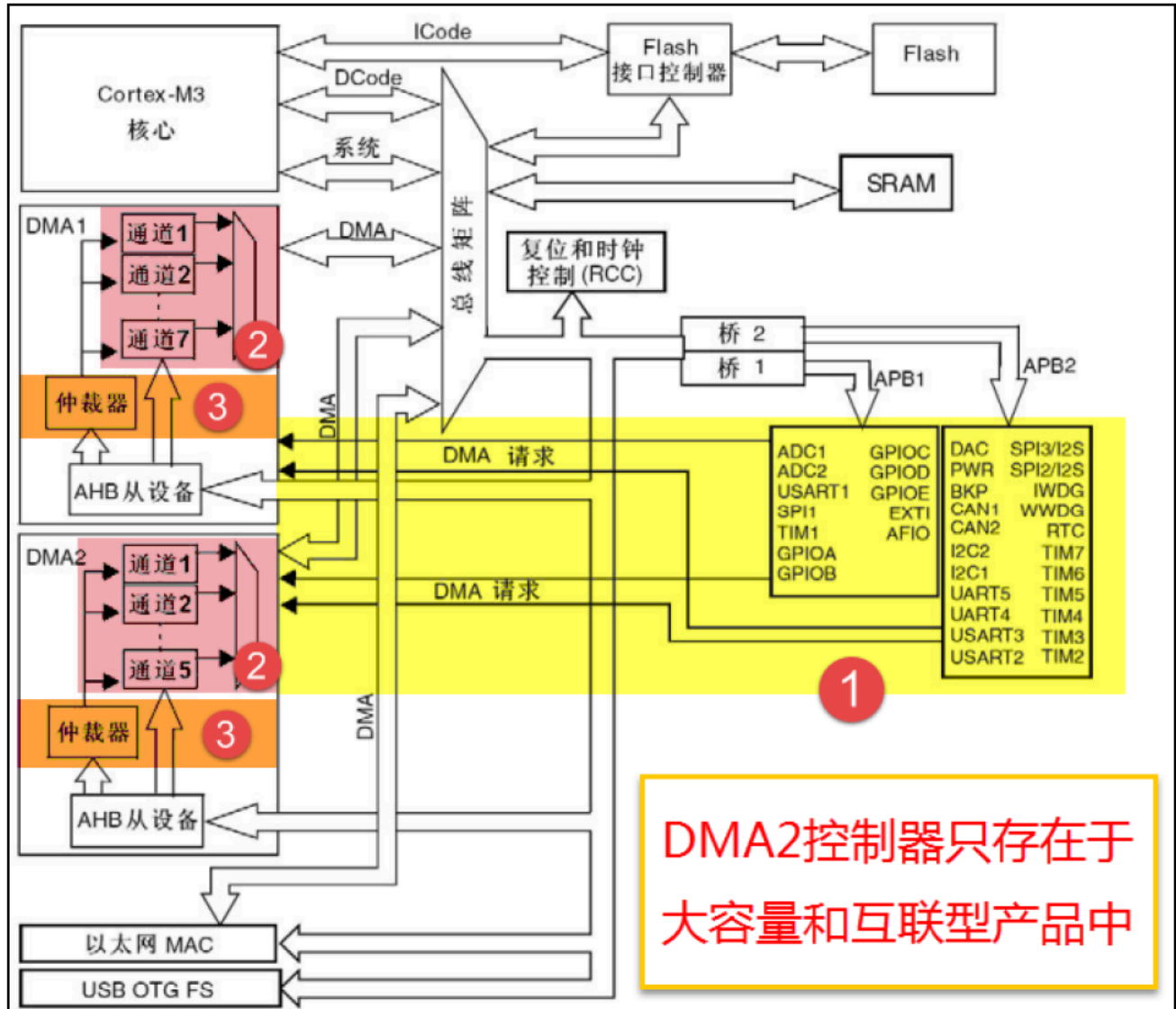
USART功能框图



！ 串口通信接发实验编程要点

1. 初始化串口读写端用到的GPIO配置
2. 初始化串口配置，USART_InitTypeDef (波特率，帧数据字长，停止位，校验位等等)
3. 串口中断配置 (中断优先级配置，接收中断使能)
4. 使能串口
5. 编写发送和接收函数
6. 编写中断服务函数

DMA功能框图



！ DMA——M-M实验编程要点

1. 在FLASH中定义好要传输的数据，在SRAM中定义好用来接收FLASH数据的变量
2. 初始化DMA，主要是配置DMA初始化结构体
3. 编写比较函数
4. 编写main函数

！ DMA——M-P实验编程要点

1. 初始化串口

2. 配置DMA初始化结构体
3. 编写主函数（开启串口发送DMA请求）

基本存储器种类

random access memory随机存储器

Dynamic RAM

Synchronous DRAM时钟同步DRAM

Double Data Rate SDRAM

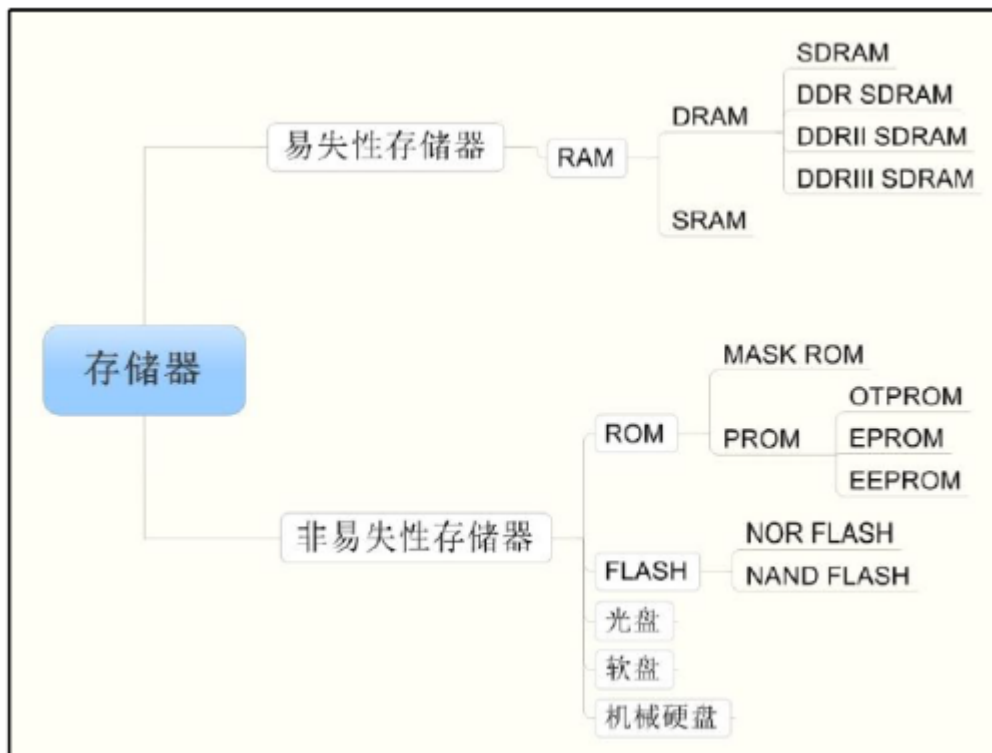
Static RAM

read only memory只读存储器

One Time Programmable ROM

Erasable Programmable ROM

Electrically Programmable ROM



EEPROM芯片和FLASH芯片区别

I2C实验的EEPROM芯片（型号：AT24C02），容量为2Kb（这里是bit），相当于256B（256个字节）。可以按字节为单位修改数据，无需整个芯片擦除。

SPI实验的FLASH芯片（型号：W25Q64）属于NOR FLASH，有8MB存储容量，分为128个64KB的块，每个块包含16个4KB的扇区。以扇区为最小擦除单位，但是可以基于字节读写（NAND FLASH必须以块为单位读写）

FLASH的存储特性

- 1.在写入数据之前必须先擦除
- 2.擦除时会把数据位全重置为1
- 3.写入数据时只能把为1的数据位改成0
- 4.擦除时必须按最小单位来擦除（一般为扇区）

I2C物理层
Inter - Integrated Circuit

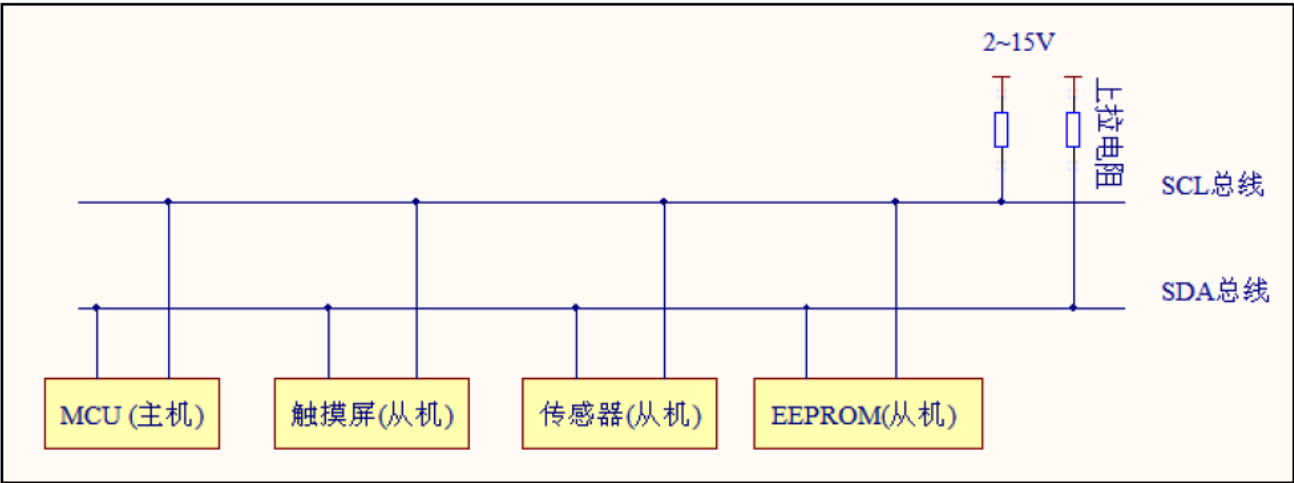
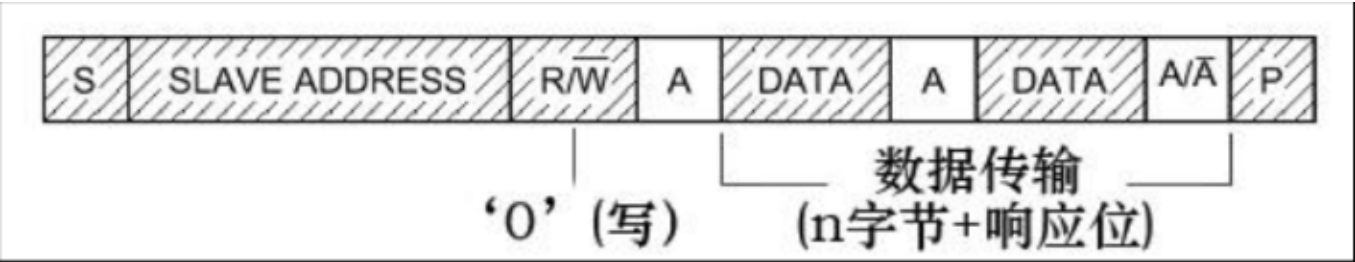
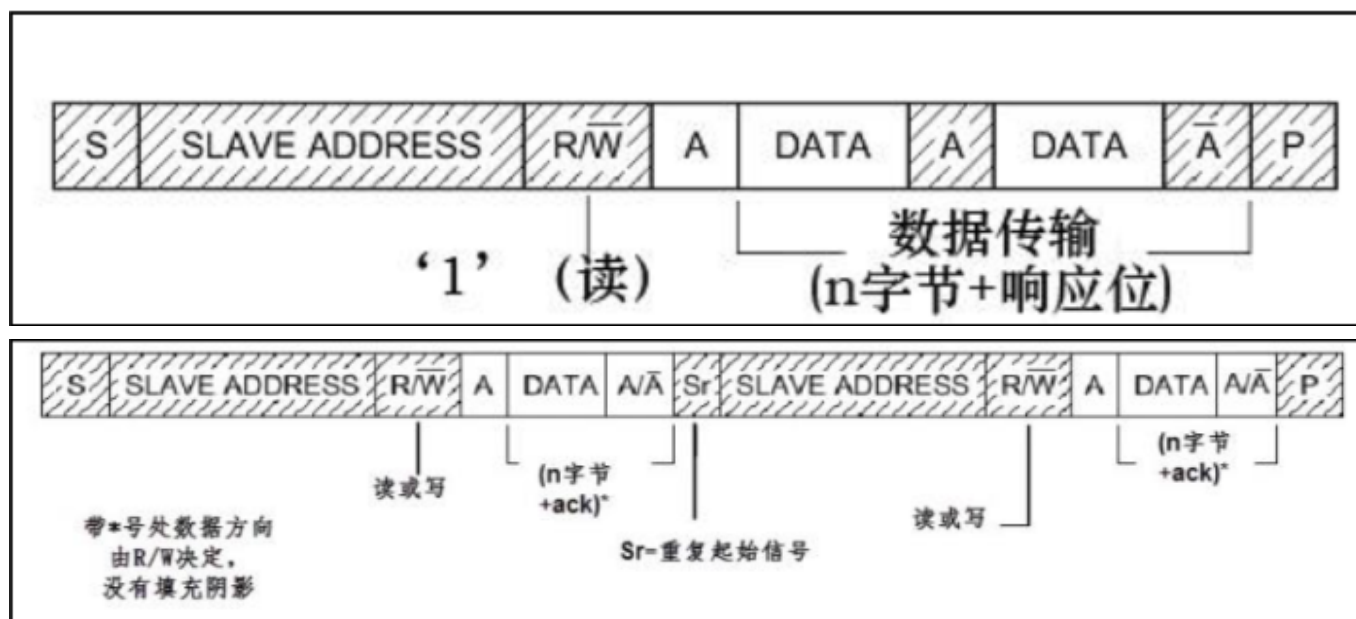


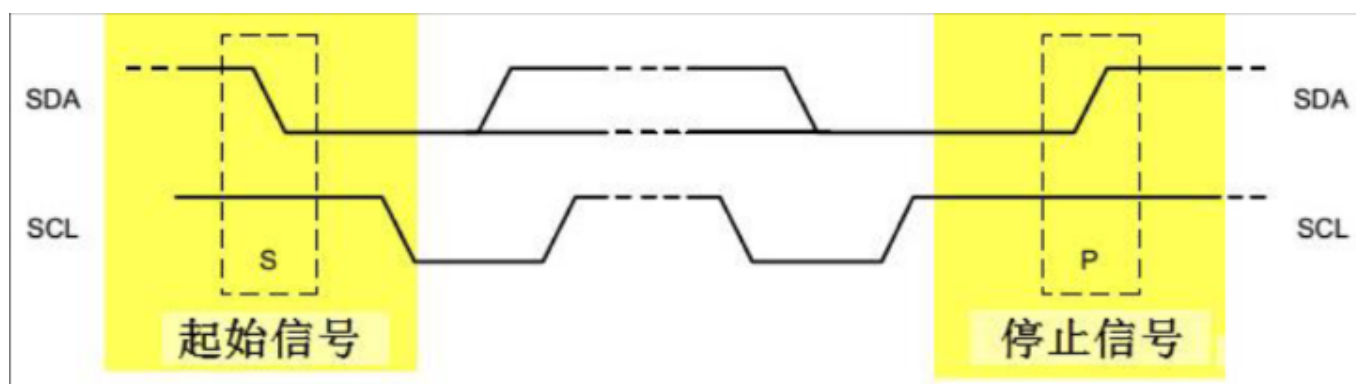
图 24-1 常见的 I2C 通讯系统

I2C协议层——基本读写信号

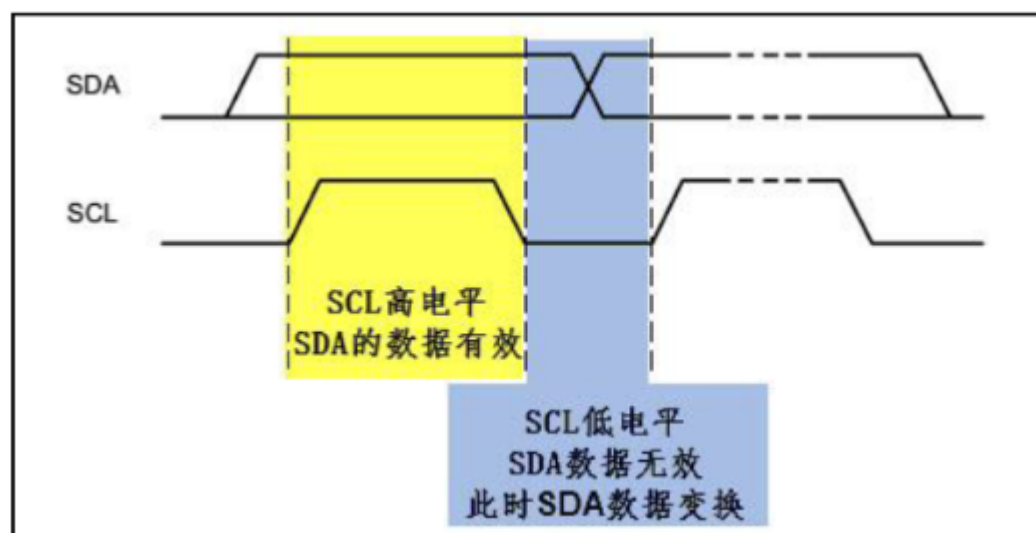




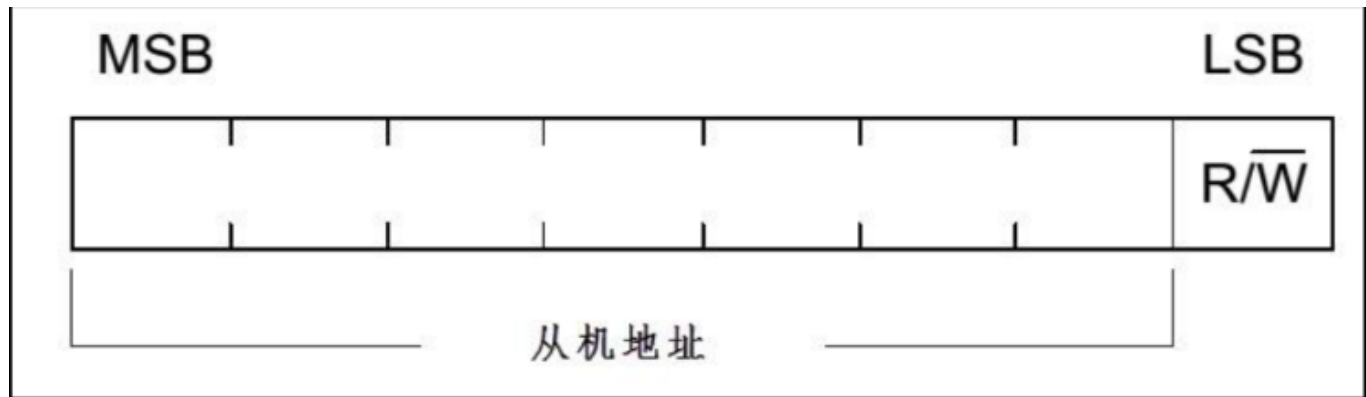
通讯的起始和停止信号



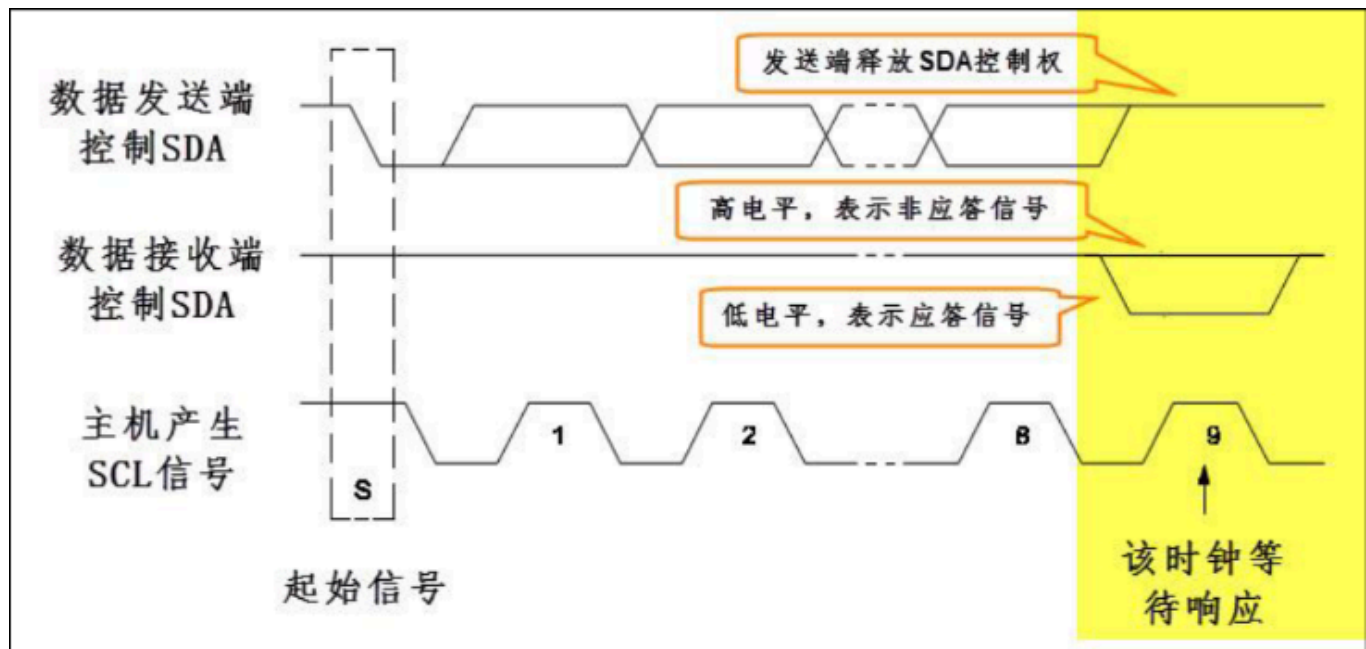
数据有效性



地址及数据方向



响应



I2C架构剖析

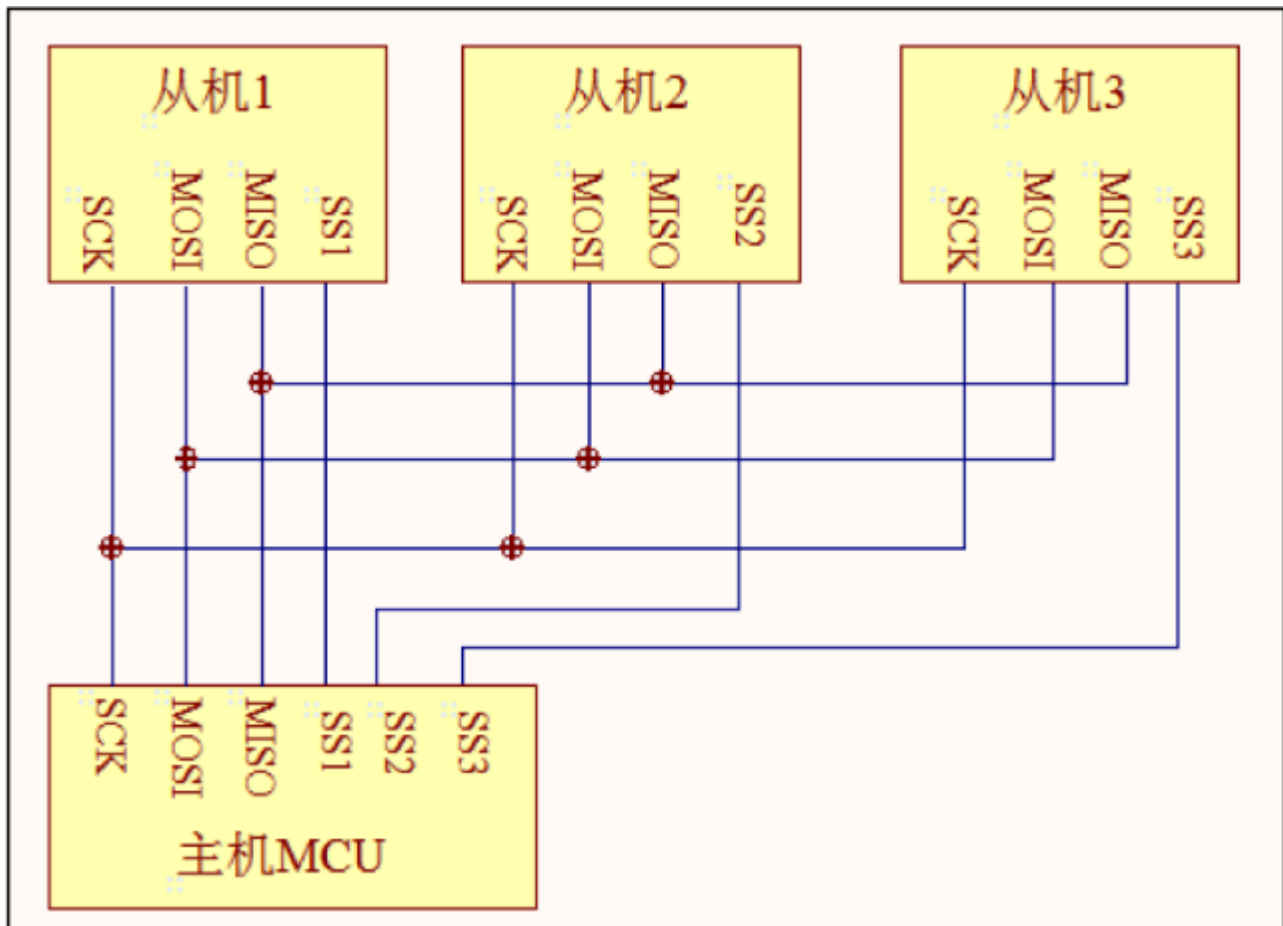
！ I2C读写EEPROM实验编程要点

1. 配置通讯使用的目标引脚为开漏模式
2. 编写模拟I2C时序的控制函数
3. 编写基本I2C按字节收发的函数
4. 编写读写EEPROM存储内容的函数
5. 编写测试程序，对读写数据进行校验

SPI物理层

SPI (Serial Peripheral Interface)

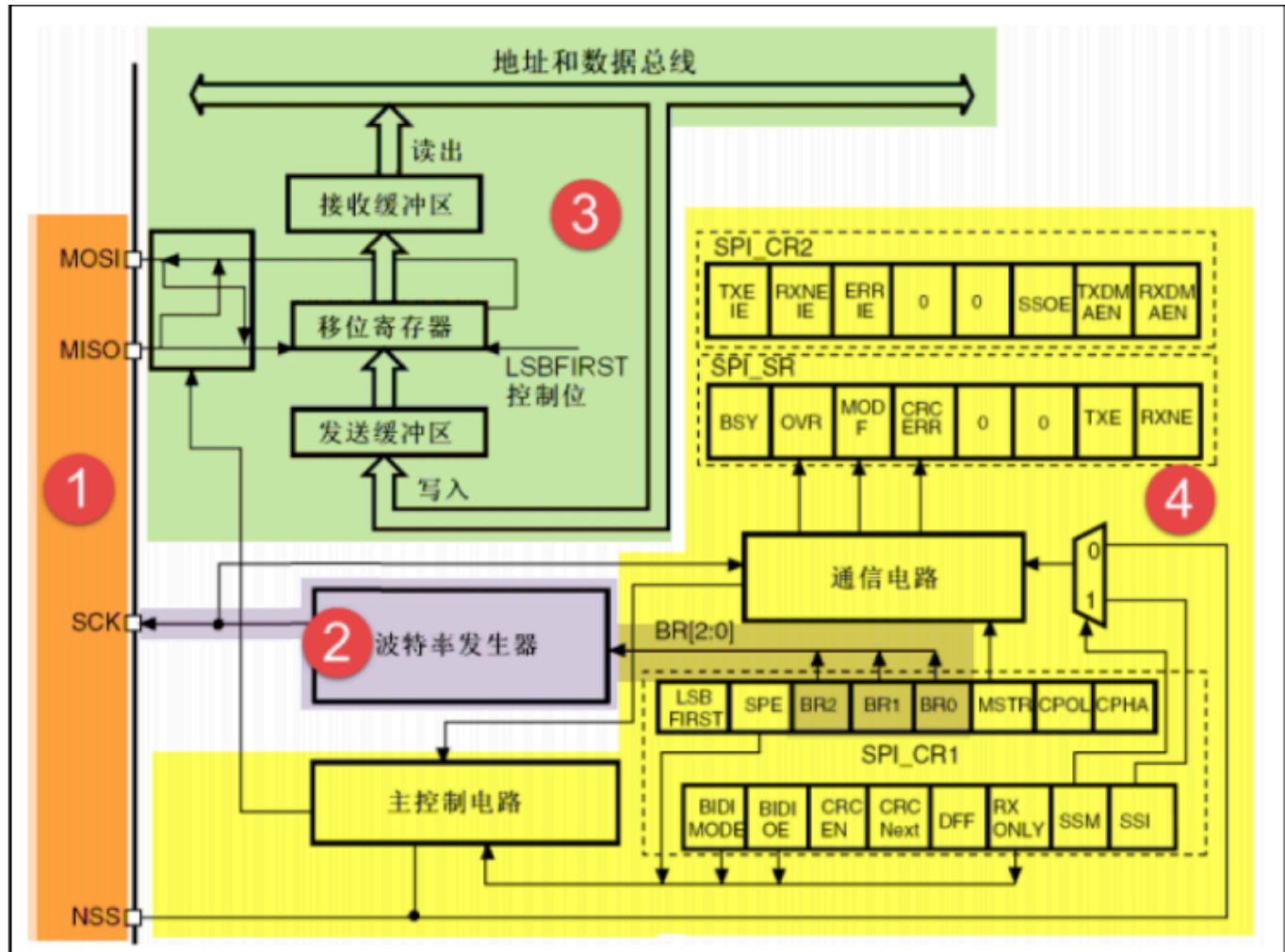
SPI协议层



SPI的四种模式

时钟极性：CPOL=0时，SCK在空闲状态为低电平；CPOL=1时，SCK在空闲状态为高电平

时钟相位：CPHA=0时，信号在SCK时钟线的奇数边沿被采样；CPHA=1时，信号在SCK时钟线的偶数边沿被采样



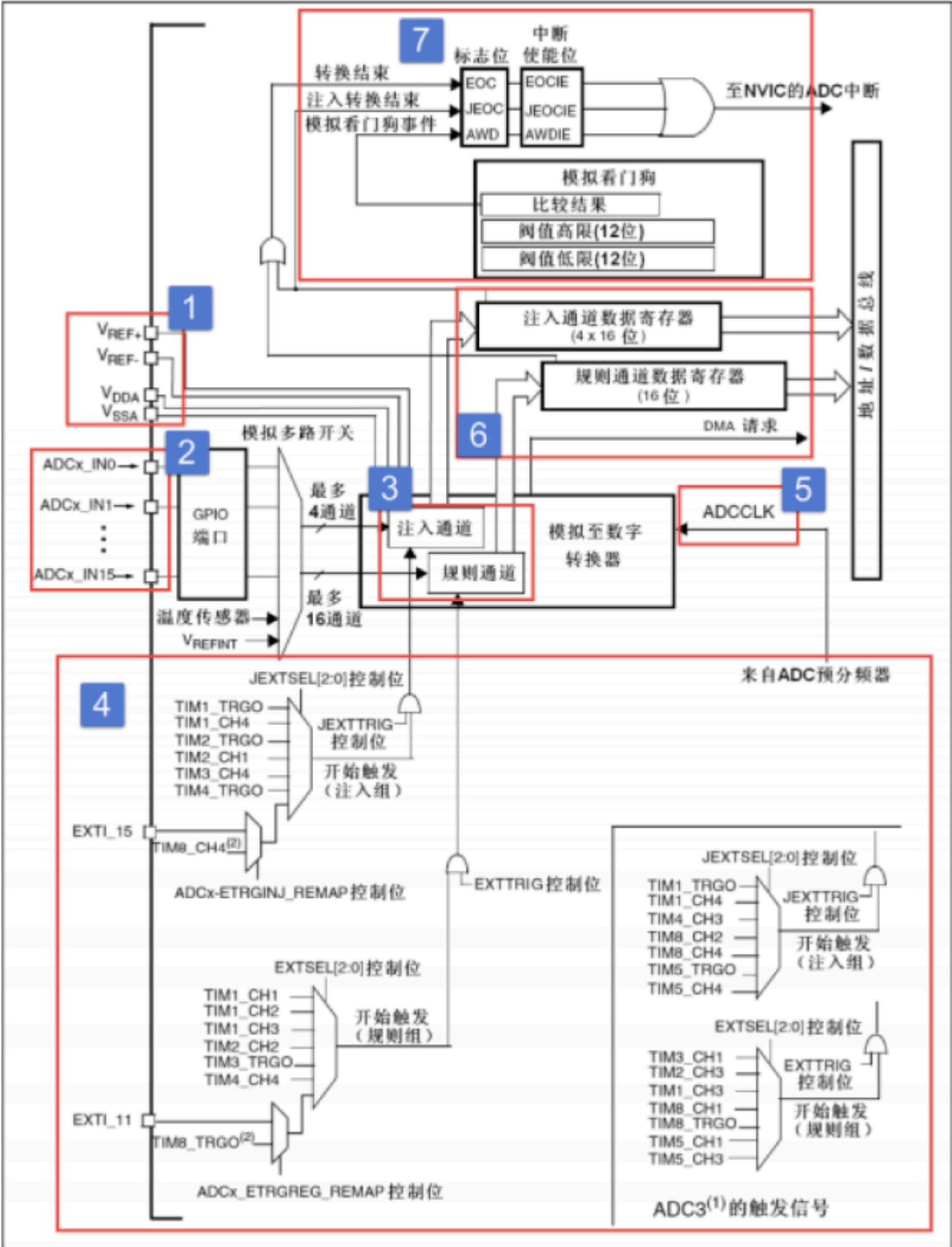
SPI通讯过程

！ SPI读写串行FLASH实验编程要点

- 初始化通讯使用的目标引脚及端口时钟；
- 使能SPI 外设以及引脚相关的时钟，配置并初始化四个引脚；
- 配置SPI 外设的模式、地址、速率等参数，并使能SPI 外设；
- 编写基本SPI 按字节收发的函数；
- 编写对FLASH 擦除及读写操作的的函数；
- 编写测试程序，对读写数据进行校验。

注：等到发送缓冲区为空时，表示可能存在的上一个数据已经发送完毕，可以调用senddata发送数据了；当接收缓冲区非空时，表示上面的数据发送完毕，且接收缓冲器也收到新的数据，可以调用receivedata读取数据了。接收数据函数只需要调用发送函数发送数据Dummy_Byte即可获取返回值，收发同步进行。

ADC功能框图



！ ADC电压采集实验编程要点

初始ADC用到的GPIO

设置ADC的工作参数并初始化

设置ADC工作时钟

设置ADC转换通道顺序及采样时间

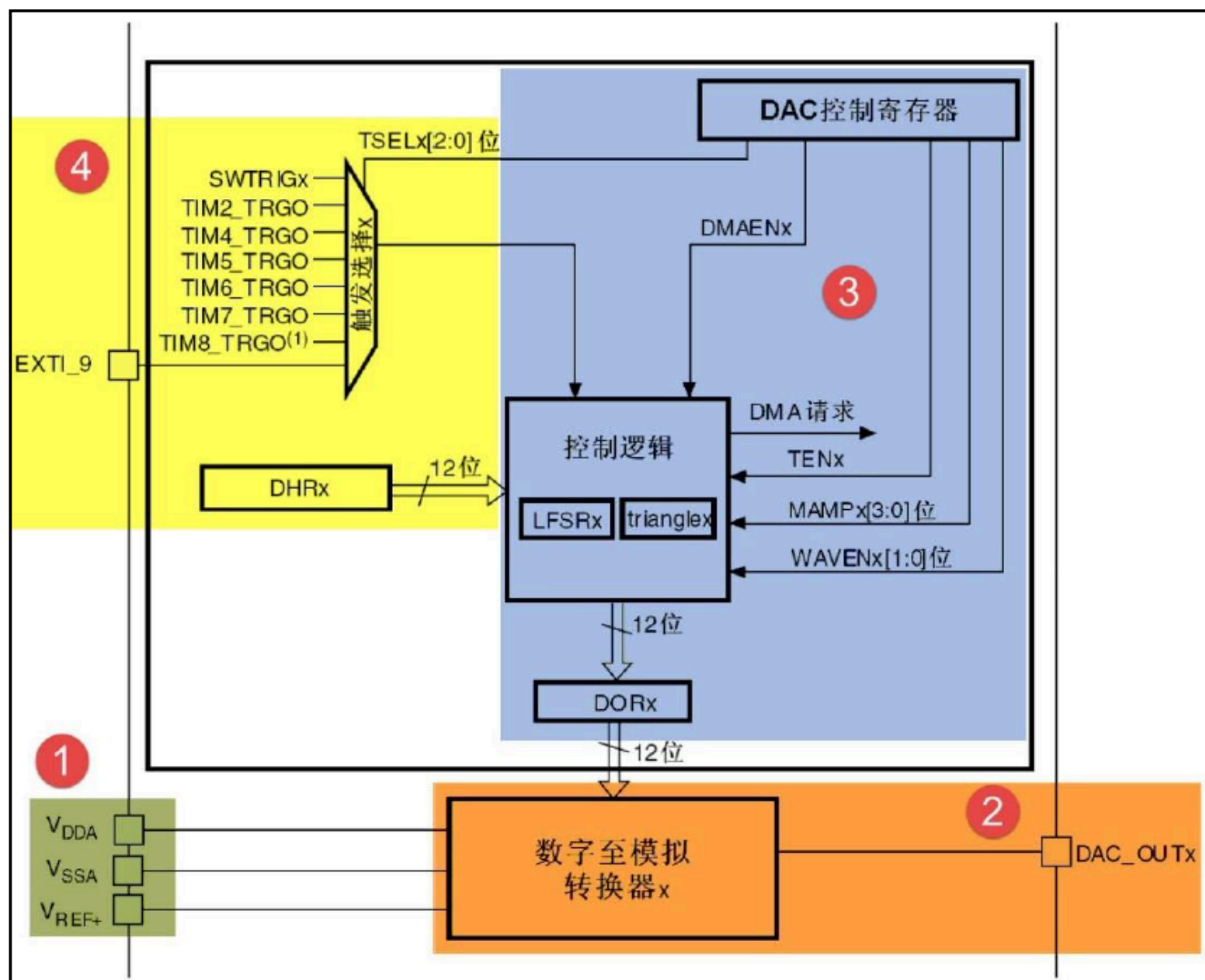
配置使能ADC转换完成中断，在中断内读取转换完的数据

使能ADC

使能软件触发ADC转换

ADC 转换结果数据使用中断方式读取，这里没有使用DMA 进行数据传输。

DAC功能框图



！ DAC输出正弦波实验编程要点

1. 计算获取正弦波数据表
2. 根据正弦波数据表的周期内点数和周期计算定时器触发间隔
3. 初始化DAC输出通道，初始化DAC工作模式
4. 配置触发DAC用的定时器
5. 配置DMA自动转运正弦波数据表

配置完成后，即可在PA4、PA5引脚中检测到信号输出