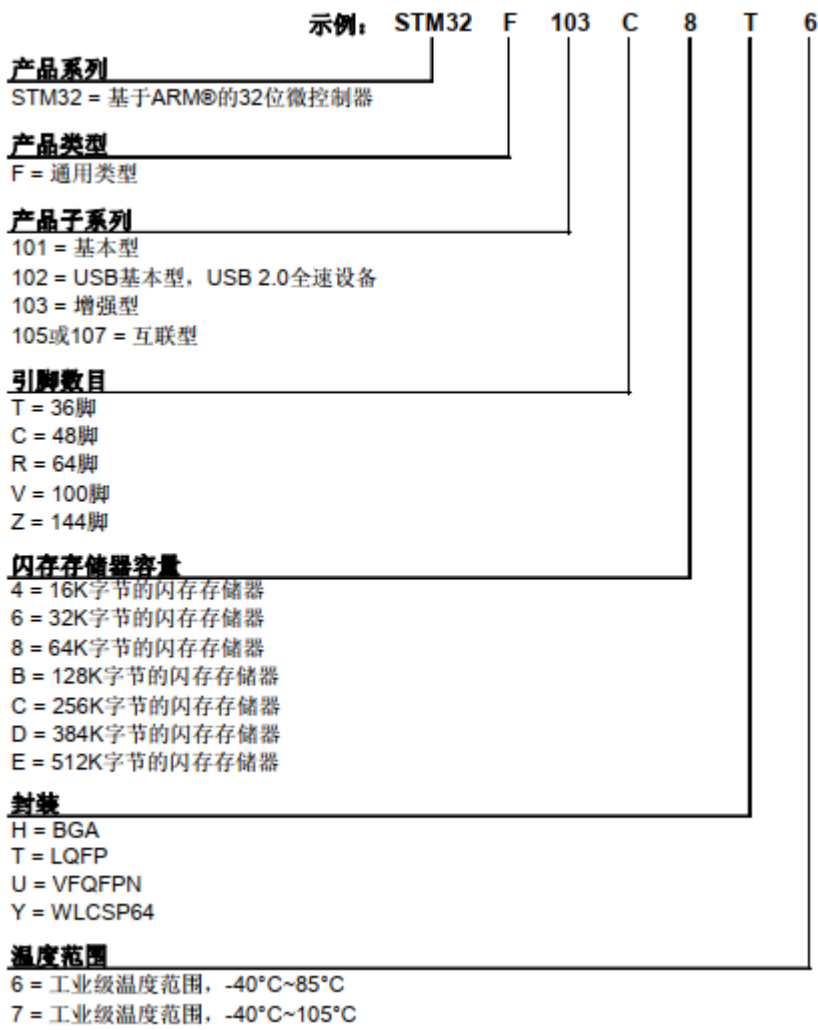


STM32学习笔记

作者：唐鹏

一、基础知识

1. stm32命名规则



2. 寄存器描述表

| | |
|--------------------------------|------------------------------------|
| read / write (rw) | 软件能读写此位。 |
| read-only (r) | 软件只能读此位。 |
| write-only (w) | 软件只能写此位，读此位将返回复位值。 |
| read/clear (rc_w1) | 软件可以读此位，也可以通过写'1'清除此位，写'0'对此位无影响。 |
| read / clear (rc_w0) | 软件可以读此位，也可以通过写'0'清除此位，写'1'对此位无影响。 |
| read / clear by read (rc_r) | 软件可以读此位；读此位将自动地清除它为'0'，写'0'对此位无影响。 |
| read / set (rs) | 软件可以读也可以设置此位，写'0'对此位无影响。 |
| read-only write trigger (rt_w) | 软件可以读此位；写'0'或'1'触发一个事件但对此位数值没有影响。 |
| toggle (t) | 软件只能通过写'1'来翻转此位，写'0'对此位无影响。 |
| Reserved(Res.) | 保留位，必须保持默认值不变 |

3. 术语表

术语表

- **小容量产品**是指闪存存储器容量在16K至32K字节之间的STM32F101xx、STM32F102xx和STM32F103xx微控制器。
- **中容量产品**是指闪存存储器容量在64K至128K字节之间的STM32F101xx、STM32F102xx和STM32F103xx微控制器。
- **大容量产品**是指闪存存储器容量在256K至512K字节之间的STM32F101xx和STM32F103xx微控制器。
- **互联型产品**是STM32F105xx和STM32F107xx微控制器。

4. 启动模式

| 启动模式选择引脚 | | 启动模式 | 说明 |
|----------|-------|--------|---------------|
| BOOT1 | BOOT0 | | |
| X | 0 | 主闪存存储器 | 主闪存存储器被选为启动区域 |
| 0 | 1 | 系统存储器 | 系统存储器被选为启动区域 |
| 1 | 1 | 内置SRAM | 内置SRAM被选为启动区域 |

5. 开漏电路与推挽电路

推挽电路：示意图：一个P型场效应管管接VCC，N型场效应管接GND，输入低电平，P型管导通，输出高电平，输入端高电平时，N型管导通，输出低电平。推挽电路只能输出高低电平。

场效应管 [源自百度百科](#)

场效应晶体管（Field Effect Transistor缩写(FET)）简称场效应管。主要有两种类型：结型场效应管（junction FET—JFET）和金属－氧化物半导体场效应管（metal-oxide semiconductor FET，简称MOS-FET）。由多数载流子参与导电，也称为单极型晶体管。它属于电压控制型半导体器件。具有输入电阻高（10⁷～10¹⁵Ω）、噪声小、功耗低、动态范围大、易于集成、没有二次击穿现象、安全工作区域宽等优点，现已成为双极型晶体管和功率晶体管的强大竞争者。场效应管（FET）是利用控制输入回路的电场效应来控制输出回路电流的一种半导体器件，并以此命名。由于它仅靠半导体中的多数载流子导电，又称单极型晶体管。FET 英文为Field Effect Transistor，简写成FET。

作用：

- 1. 场效应管可应用于放大。由于场效应管放大器的输入阻抗很高，因此耦合电容可以容量较小，不必使用电解电容器。
- 2. 场效应管很高的输入阻抗非常适合作阻抗变换。常用于多级放大器的输入级作阻抗变换。
- 3. 场效应管可以用作可变电阻。
- 4. 场效应管可以方便地用作恒流源。
- 5. 场效应管可以用作电子开关。

灌电流和拉电流：以驱动led的推挽电路为例。当led一端接推挽电路一端接地时（电流从内向外），电流为拉电流；反之，当led一段接led一段接VCC时（电流从外向内），电流称为灌电流。无论灌电流还是拉电流，都是把I/O口当作输出口。

开漏电路：漏指场效应管的漏极，把场效应管的漏极作为输出。做电平的匹配，所以需要外部的电源（VCC）

作用：

1. 做电平的匹配 2. 多个开漏输出引脚接到一个电源上，形成一个与逻辑，一条线上有一个低电平，整条线上就都是低电平，例如IIC，判断IIC工作状态。

二、时钟系统

时钟的作用

1. 计时作用（供给某系数计数器统计时间）
2. 控制时序（串口数据的传输，只能一位一位传输） 注：一位八个字节
3. 控制信号（将时钟的上升下降沿作为独特的控制标志）

三种不同的时钟源可以用来驱动系统时钟：

- HSI振荡器时钟
- HSE振荡器时钟
- PLL时钟

这些设备有以下两种二级时钟源：

- 40kHz低速内部RC，可以用于驱动独立看门狗和通过程序选择驱动RTC。RTC用于从停机/待机模式下唤醒系统。（RTC时钟使用外部纽扣电池供电，可以用来作为时钟，达到类似arduino函数库里millis()的功能）
- 32.768kHz低速外部晶振也可用来通过程序选择驱动RTC（RTCCLK）。当不被使用时，仍一个时钟源都可以被独立地启动或关闭，由此优化系统能耗。

时钟源

stm32时钟源分为以下：

- 低速内部时钟LSI：频率为40KHz
- 高速内部时钟HSI：频率为8MHz
- 低速外部时钟OSC_32: 频率为32.768kHz
- 高速外部时钟OSC：频率范围4-16MHz
- 时钟输出MCO：为其他设备提供时钟源

在对时钟频率要求不高的情况下，我们可以选择内部RC振荡器时钟作为单片机工作的时钟源。如果对时钟精度要求较高，我们要选择外部石英晶体振荡器（晶振），作为单片机的时钟源，因为内部时钟用的是模拟电路组成的振荡器，误差较大，在实时性要求比较高的场合（串口通信、IIC通信等）容易造成不可预知的错误。

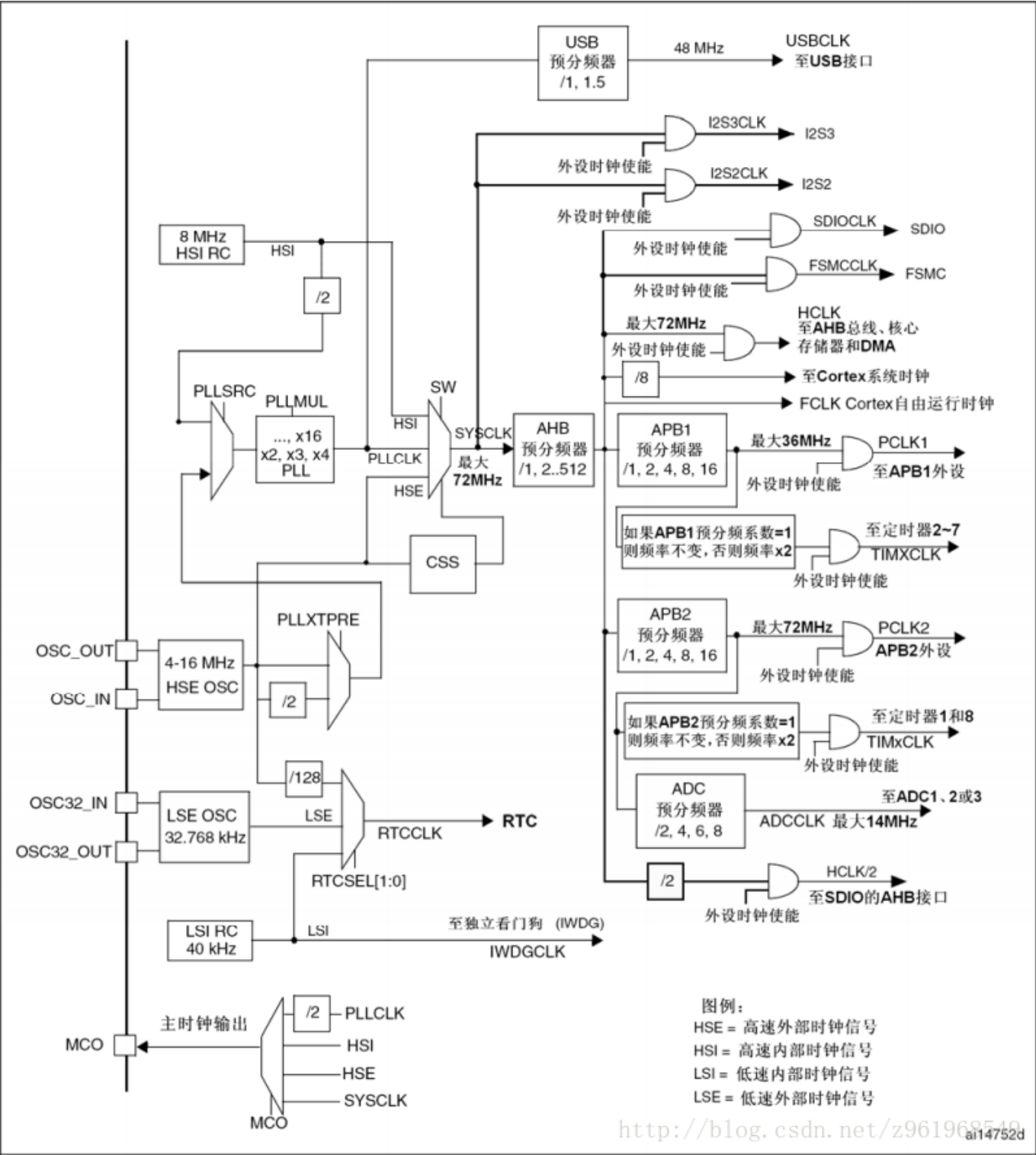
时钟相对复杂的意义主要是为不同外设提供合适的时钟频率，主要目的是为了节能、低功耗。

高速时钟：用于高速外设IO 串口通信 SPI等等

低速时钟：用于低速外设 RTC 看门狗
倍频器：时钟与外设进行时钟适配

stm32系统时钟参考链接（csdn）

stm32的时钟树



占坑以后来补

三、外设

1. GPIO STM32的IO口有以下八种模式：

输入模式

- 输入浮空
- 输入上拉
- 输入下拉
- 模拟输入

输出模式

- 推挽输出
- 开漏输出

复用模式

- 推挽复用
- 开漏复用

每个 IO 口可以自由编程，但 IO 口寄存器必须要按 32 位字被访问。STM32 的很多 IO 口都是 5V 兼容的，这些 IO 口在与 5V 电平的外设连接的时候很有优势

注:stm32IO口耐压值不是全都是5v，也有3.3v，具体要看对应芯片的数据手册中的管脚描述章节。FT为5v兼容

STM32 的每个 IO 端口都有 7 个寄存器来控制。他们分别是：配置模式的 2 个 32 位的端口配置寄存器 CRL 和 CRH；2 个 32 位的数据寄存器 IDR 和 ODR；1 个 32 位的置位/复位寄存器 BSRR；一个 16 位的复位寄存器 BRR；1 个 32 位的锁存寄存器 LCKR。

端口配置低寄存器CRL

8.2.1 端口配置低寄存器(GPIOx_CRL) (x=A..E)

偏移地址: 0x00
复位值: 0x4444 4444

| | | | | | | | | | | | | | | | |
|-----------|----|--|----|-----------|----|------------|----|-----------|----|------------|----|-----------|----|------------|----|
| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| CNF7[1:0] | | MODE7[1:0] | | CNF6[1:0] | | MODE6[1:0] | | CNF5[1:0] | | MODE5[1:0] | | CNF4[1:0] | | MODE4[1:0] | |
| RW | RW | RW | RW | RW | RW | RW | RW | RW | RW | RW | RW | RW | RW | RW | RW |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| CNF3[1:0] | | MODE3[1:0] | | CNF2[1:0] | | MODE2[1:0] | | CNF1[1:0] | | MODE1[1:0] | | CNF0[1:0] | | MODE0[1:0] | |
| RW | RW | RW | RW | RW | RW | RW | RW | RW | RW | RW | RW | RW | RW | RW | RW |
| 位31:30 | | CNFy[1:0]: 端口x配置位(y = 0...7) (Port x configuration bits) | | | | | | | | | | | | | |
| 27:26 | | 软件通过这些位配置相应的I/O端口, 请参考表17端口位配置表。 | | | | | | | | | | | | | |
| 23:22 | | 在输入模式(MODE[1:0]=00): | | | | | | | | | | | | | |
| 19:18 | | 00: 模拟输入模式 | | | | | | | | | | | | | |
| 15:14 | | 01: 浮空输入模式(复位后的状态) | | | | | | | | | | | | | |
| 11:10 | | 10: 上拉/下拉输入模式 | | | | | | | | | | | | | |
| 7:6 | | 11: 保留 | | | | | | | | | | | | | |
| 3:2 | | 在输出模式(MODE[1:0]>00): | | | | | | | | | | | | | |
| | | 00: 通用推挽输出模式 | | | | | | | | | | | | | |
| | | 01: 通用开漏输出模式 | | | | | | | | | | | | | |
| | | 10: 复用功能推挽输出模式 | | | | | | | | | | | | | |
| | | 11: 复用功能开漏输出模式 | | | | | | | | | | | | | |
| 位29:28 | | MODEy[1:0]: 端口x的模式位(y = 0...7) (Port x mode bits) | | | | | | | | | | | | | |
| 25:24 | | 软件通过这些位配置相应的I/O端口, 请参考表17端口位配置表。 | | | | | | | | | | | | | |
| 21:20 | | 00: 输入模式(复位后的状态) | | | | | | | | | | | | | |
| 17:16 | | 01: 输出模式, 最大速度10MHz | | | | | | | | | | | | | |
| 13:12 | | 10: 输出模式, 最大速度2MHz | | | | | | | | | | | | | |
| 9:8, 5:4 | | 11: 输出模式, 最大速度50MHz | | | | | | | | | | | | | |
| 1:0 | | | | | | | | | | | | | | | |

stm32有A-G 7组IO, 每组IO有16个引脚。CRL寄存器控制低八位的引脚, 也就是0到7号引脚。同样的CRH寄存器控制的是高八位的引脚, 也就是8到15号引脚。每个引脚都有一个字节(八位数据)来配置它的模式功能。MODE[1: 0]是用来配置引脚的输入输出模式以及输出模式时的速度。而CNF[1:0]则是用来配置具体的输入输出模式, 例如开漏输出, 上下拉输入。

端口配置高寄存器CRH

8.2.2 端口配置高寄存器(GPIOx_CRH) (x=A..E)

偏移地址: 0x04
复位值: 0x4444 4444

| | | | | | | | | | | | | | | | |
|------------|-------------|---|-------------|------------|-------------|------------|-------------|----|----|----|----|----|----|----|----|
| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| CNF15[1:0] | MODE15[1:0] | CNF14[1:0] | MODE14[1:0] | CNF13[1:0] | MODE13[1:0] | CNF12[1:0] | MODE12[1:0] | | | | | | | | |
| rW | rW | rW | rW | rW | rW | rW | rW | rW | rW | rW | rW | rW | rW | rW | rW |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| CNF11[1:0] | MODE11[1:0] | CNF10[1:0] | MODE10[1:0] | CNF9[1:0] | MODE9[1:0] | CNF8[1:0] | MODE8[1:0] | | | | | | | | |
| rW | rW | rW | rW | rW | rW | rW | rW | rW | rW | rW | rW | rW | rW | rW | rW |
| 位31:30 | | CNFy[1:0]: 端口x配置位(y = 8...15) (Port x configuration bits) | | | | | | | | | | | | | |
| 27:26 | | 软件通过这些位配置相应的I/O端口, 请参考表17端口位配置表。 | | | | | | | | | | | | | |
| 23:22 | | 在输入模式(MODE[1:0]=00): | | | | | | | | | | | | | |
| 19:18 | | 00: 模拟输入模式 | | | | | | | | | | | | | |
| 15:14 | | 01: 浮空输入模式(复位后的状态) | | | | | | | | | | | | | |
| 11:10 | | 10: 上拉/下拉输入模式 | | | | | | | | | | | | | |
| 7:6 | | 11: 保留 | | | | | | | | | | | | | |
| 3:2 | | 在输出模式(MODE[1:0]>00): | | | | | | | | | | | | | |
| | | 00: 通用推挽输出模式 | | | | | | | | | | | | | |
| | | 01: 通用开漏输出模式 | | | | | | | | | | | | | |
| | | 10: 复用功能推挽输出模式 | | | | | | | | | | | | | |
| | | 11: 复用功能开漏输出模式 | | | | | | | | | | | | | |
| 位9:28 | | MODEy[1:0]: 端口x的模式位(y = 8...15) (Port x mode bits) | | | | | | | | | | | | | |
| 25:24 | | 软件通过这些位配置相应的I/O端口, 请参考表17端口位配置表。 | | | | | | | | | | | | | |
| 21:20 | | 00: 输入模式(复位后的状态) | | | | | | | | | | | | | |
| 17:16 | | 01: 输出模式, 最大速度10MHz | | | | | | | | | | | | | |
| 13:12 | | 10: 输出模式, 最大速度2MHz | | | | | | | | | | | | | |
| 9:8, 5:4 | | 11: 输出模式, 最大速度50MHz | | | | | | | | | | | | | |
| 1:0 | | | | | | | | | | | | | | | |

端口数据输入寄存器

8.2.3 端口输入数据寄存器(GPIOx_IDR) (x=A..E)

地址偏移: 0x08
复位值: 0x0000 XXXX

| | | | | | | | | | | | | | | | |
|--------|-------|--|-------|-------|-------|------|------|------|------|------|------|------|------|------|------|
| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| 保留 | | | | | | | | | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| IDR15 | IDR14 | IDR13 | IDR12 | IDR11 | IDR10 | IDR9 | IDR8 | IDR7 | IDR6 | IDR5 | IDR4 | IDR3 | IDR2 | IDR1 | IDR0 |
| r | r | r | r | r | r | r | r | r | r | r | r | r | r | r | r |
| 位31:16 | | 保留, 始终读为0。 | | | | | | | | | | | | | |
| 位15:0 | | IDRy[15:0]: 端口输入数据(y = 0...15) (Port input data) | | | | | | | | | | | | | |
| | | 这些位为只读并只能以字(16位)的形式读出。读出的值为对应I/O口的状态。 | | | | | | | | | | | | | |

端口数据输出寄存器

8.2.4 端口输出数据寄存器(GPIOx_ODR) (x=A..E)

地址偏移: 0Ch
复位值: 0x0000 0000

| | | | | | | | | | | | | | | | |
|--------|-------|---|-------|-------|-------|------|------|------|------|------|------|------|------|------|------|
| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| 保留 | | | | | | | | | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| ODR15 | ODR14 | ODR13 | ODR12 | ODR11 | ODR10 | ODR9 | ODR8 | ODR7 | ODR6 | ODR5 | ODR4 | ODR3 | ODR2 | ODR1 | ODR0 |
| RW | RW | RW | RW | RW | RW | RW | RW | RW | RW | RW | RW | RW | RW | RW | RW |
| 位31:16 | | 保留，始终读为0。 | | | | | | | | | | | | | |
| 位15:0 | | ODRy[15:0] : 端口输出数据(y = 0...15) (Port output data) 这些位可读可写并只能以字(16位)的形式操作。 注: 对GPIOx_BSRR(x = A..E), 可以分别地对各个ODR位进行独立的设置/清除。 | | | | | | | | | | | | | |

端口位设置/清除寄存器

8.2.5 端口位设置/清除寄存器(GPIOx_BSRR) (x=A..E)

地址偏移: 0x10
复位值: 0x0000 0000

| | | | | | | | | | | | | | | | |
|--------|------|---|------|------|------|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| BR15 | BR14 | BR13 | BR12 | BR11 | BR10 | BR9 | BR8 | BR7 | BR6 | BR5 | BR4 | BR3 | BR2 | BR1 | BR0 |
| W | W | W | W | W | W | W | W | W | W | W | W | W | W | W | W |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| BS15 | BS14 | BS13 | BS12 | BS11 | BS10 | BS9 | BS8 | BS7 | BS6 | BS5 | BS4 | BS3 | BS2 | BS1 | BS0 |
| W | W | W | W | W | W | W | W | W | W | W | W | W | W | W | W |
| 位31:16 | | BRy : 清除端口x的位y (y = 0...15) (Port x Reset bit y) 这些位只能写入并只能以字(16位)的形式操作。 0: 对对应的ODRy位不产生影响 1: 清除对应的ODRy位为0 注: 如果同时设置了BSy和BRy的对应位, BSy位起作用。 | | | | | | | | | | | | | |
| 位15:0 | | BSy : 设置端口x的位y (y = 0...15) (Port x Set bit y) 这些位只能写入并只能以字(16位)的形式操作。 0: 对对应的ODRy位不产生影响 1: 设置对应的ODRy位为1 | | | | | | | | | | | | | |

端口位清除寄存器

8.2.6 端口位清除寄存器(GPIOx_BRR) (x=A..E)

地址偏移: 0x14
复位值: 0x0000 0000

| | | | | | | | | | | | | | | | |
|------|------|------|------|------|------|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| 保留 | | | | | | | | | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| BR15 | BR14 | BR13 | BR12 | BR11 | BR10 | BR9 | BR8 | BR7 | BR6 | BR5 | BR4 | BR3 | BR2 | BR1 | BR0 |
| W | W | W | W | W | W | W | W | W | W | W | W | W | W | W | W |

stm32输出模式配置

| | |
|------------|----|
| MODE[1: 0] | 意义 |
|------------|----|

| | |
|----|--------------|
| 00 | 保留 |
| 01 | 最大输出速度为10MHz |
| 10 | 最大输出速度为2MHz |
| 11 | 最大输出速度为50MHz |

补:C语言知识

关于置位以及清零

实例: 设置 PC11 为上拉输入, PC12 为推挽输出

`GPIOC->CRH&=0xFFFF00FF;` //清掉这 2 个位原来的设置, 同时也不影响其他位的设置

`GPIOC->CRH|=0X00038000;` //PC11 输入, PC12 输出

`GPIOC->ODR=1<<11;` //PC11 上拉

注: 一个十六进制数等于四位二进制数, 相当于一个十六进制数就能设置一个引脚的模式

置位

四、通信协议

1. SBUS协议

串口做一个电平反向。可以通过三极管电路（反相器） 正常配置和串口配置一样。
以下为一些固定配置：

- 波特率为100K
- 奇偶校验为偶校验
- 帧长 25个字节, 22个数据位
- 停止位

五、应用篇

1. GPIO配置

- 使能定时器
- 引脚配置（模式, 速度等）
- 初始化GPIO
-

2. 定时器中断设置思路

- 中断配置（main中）
- 定时器配置
 1. 使能定时器时钟
 2. 配置定时器参数
 3. 使能定时器中断
 4. 设置中断优先级
 5. 使能定时器
- 编写中断服务函数

3. MOS管开关电路

MOS管三个引脚，N型和P型差别在于导通方向

以NMOS为例

|—— D栅极

G漏极--|

|—— S源极

给栅极和源极之间一个电压，当这个电压大于MOS管本身门限电压时，漏极和源极导通，当小于门限电压时，漏极和源极不导通

注：作为开关需要工作在饱和和截止区，如果是电子负载就需要工作在线性区

需要考虑的几点

1. MOS管的门限电压（单片机给出的电压要大于门限电压才能开启MOS管） 2. 漏极和源极之间最大允许的电压（VDS）和电流（IDS） 3. MOS管的工作温度