GPIO用法及应用流程（也含其他设备的用法和配置）

（GPIO用于STM32的IO处电平、信号的输入输出）

1)定义一个GPIO结构体 GPIO\_InitTypeDef GPIO\_InitStructure; （为了向GPIO外设内写数据）

2)开启GPIOx口的时钟外设：（ 可以一次设置多个用（。。。|。。。）形式）

RCC\_APB2PeriphClockCmd( RCC\_APB2Periph\_GPIOx,ENABLE);——GPIO配置时都是用APB2，也就说这个语句在GPIO中就是这个样子。

3)选择引脚 GPIO\_InitStructure.GPIO\_Pin = GPIO\_Pin\_0;

4)设置GPIO输出模式 GPIO\_InitStructure.GPIO\_Mode = GPIO\_Mode\_Out\_PP;

（1）浮空输入：In\_Floating

     （2）带上拉输入：IPU（In Push\_Up）

（3）带下拉输入：IPD（In Push\_Down）

（4）模拟输入：AIN（Anolog In）

（5）开漏输出：OUT\_PD（OD 代表开漏，既Open Drain，OC代表开集，Open Collect）

（6）推挽输出：OUT\_PP（PP代表推挽式，Push\_Pull）效果例：三个灯轮流亮每刻有且只有一个灯亮

（7）复用功能推挽输出：AF\_PP（AF代表复用功能，Alternate-Fuction）

（8）复用功能开漏输出：AF\_OD

5)设置引脚速率 GPIO\_InitStructure.GPIO\_Speed = GPIO\_Speed\_50MHz;

6)调用库函数初始化GPIO（注：每次只能初始化一个引脚） GPIO\_Init(GPIOB, &GPIO\_InitStructure);

（如果要控制其他引脚，重复上面的3和6有时也会用到4步，总的来说配置一个GPIO至少要有1—6步）

另外：

GPIO\_SetBits(GPIOx, GPIO\_Pin\_i|GPIO\_Pin\_j); ——对所有端口进行置1操作（高电平）

GPIO\_ResetBits(GPIOx，GPIO\_Pin\_i); ——对所有端口进行置0操作（低电平）

GPIO\_WriteBits(GPIOx, GPIO\_Pin\_i,1); ——对一个端口进行写1（也可以写0，把括号内的1改为0即可）操作

GPIO\_Write(GPIOA, 0x1101);——向指定GPIO数据端口写入数据

**GPIO库函数：**

**函数名**   **描述**

GPIO\_DeInit 将外设GPIOx寄存器重设为缺省值

GPIO\_AFIODeInit 将复用功能（重映射事件控制和EXTI设置）重设为缺省值 GPIO\_Init 根据GPIO\_InitStruct中指定的参数初始化外设GPIOx寄存器 GPIO\_StructInit 把GPIO\_InitStruct中的每一个参数按缺省值填入 GPIO\_ReadInputDataBit 读取指定端口管脚的输入

GPIO\_ReadInputData(GPIOx,GPIO\_Pin\_i) 读取指定的GPIO端口输入

GPIO\_ReadOutputDataBit(GPIOx,GPIO\_Pin\_i) 读取指定端口管脚的输出位（0|1）

GPIO\_ReadOutputData(GPIOx,GPIO\_Pin\_i) 读取指定的GPIO端口输出

GPIO\_SetBits 设置指定的数据端口位

GPIO\_ResetBits 清除指定的数据端口位

GPIO\_WriteBit 设置或者清除指定的数据端口位

GPIO\_Write 向指定GPIO数据端口写入数据

GPIO\_PinLockConfig 锁定GPIO管脚设置寄存器

GPIO\_EventOutputConfig 选择GPIO管脚用作事件输出

GPIO\_EventOutputCmd 使能或者失能事件输出

GPIO\_PinRemapConfig 改变指定管脚的映射

GPIO\_EXTILineConfig(GPIO\_PortSourceGPIOA, GPIO\_PinSource0); 选择GPIO管脚用作外部中断线路 (例子是让PA0引脚作为中断线路),中断触发时该线将中断输出。

注：GPIO引脚与端口：

SYM32中将GPIO引脚分为了GPIOA、GPIOB……GPIOG不同的组，每组中每个引脚叫端口（例如：可以说GPIOA中的Pin\_1端口，在开发板上可以看见写着A1字样的位置即是），每组端口分为0~15，共16个不同的引脚（此处真实数据依芯片型号的datasheet），端口就是开发板上伸出来的一堆针，引脚是ARM芯片与电路板焊接的地方，综上也可以理解为“端口”就是“引脚”在组里的“昵称”。