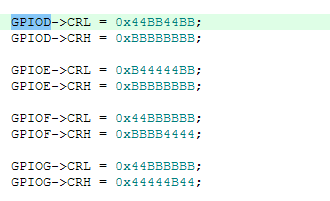
首先从字面上理解就是这个变量容易发生变化，对吧？

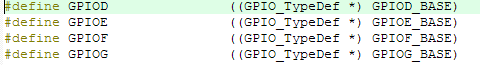
可是变量不总是会发生变化吗？这个易变变量和普通变量到底区别在哪里？

在这里我们可以以STM的IO寄存器访问来学习其中的细节。

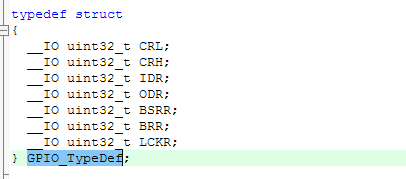
首先这是标准的STM处理器IO寄存器的赋值操作。



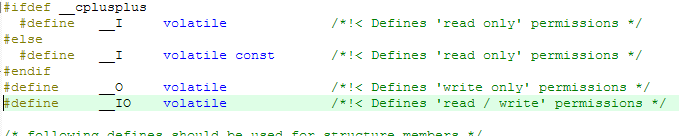
然后我们跟踪到GPIOD的定义



发现GPIO是统一的类的不同实例，没有问题，继续跟踪这个GPIO\_TypeDef类的定义。



好的，这是一个结构体，其中有我们熟知的几个寄存器，CRL CRH高低配置位，IDR ODR 输入输出数据寄存器，BSRR BRR 数据位修改寄存器，LCKR ？？暂时不清楚，然后这里就是寄存器的值了，这里我们发现了一个很常见的自定义修饰符 \_\_IO ，代表着与IO操作相关的变量，那么这个\_\_IO 又是怎么定义的呢？



原来，这个\_\_IO 的实际作用等价于volatile 修饰符。

总结一下也是符合实际情况，IO既可输出，也可输入，程序无法得知在使用过程中IO的值，故用volatile修饰，迫使CPU每次都要去读相应的实际值，而不是普通变量那种事先读取到内存，或者缓存中，方便快速访问的那种类型。

这就是本人对volatile的一点理解，如有错误请指正。