

通过对AUTOSAR标准文档《AUTOSAR\_SWS\_FunctionInhibitionManager》的分析，总结如下：

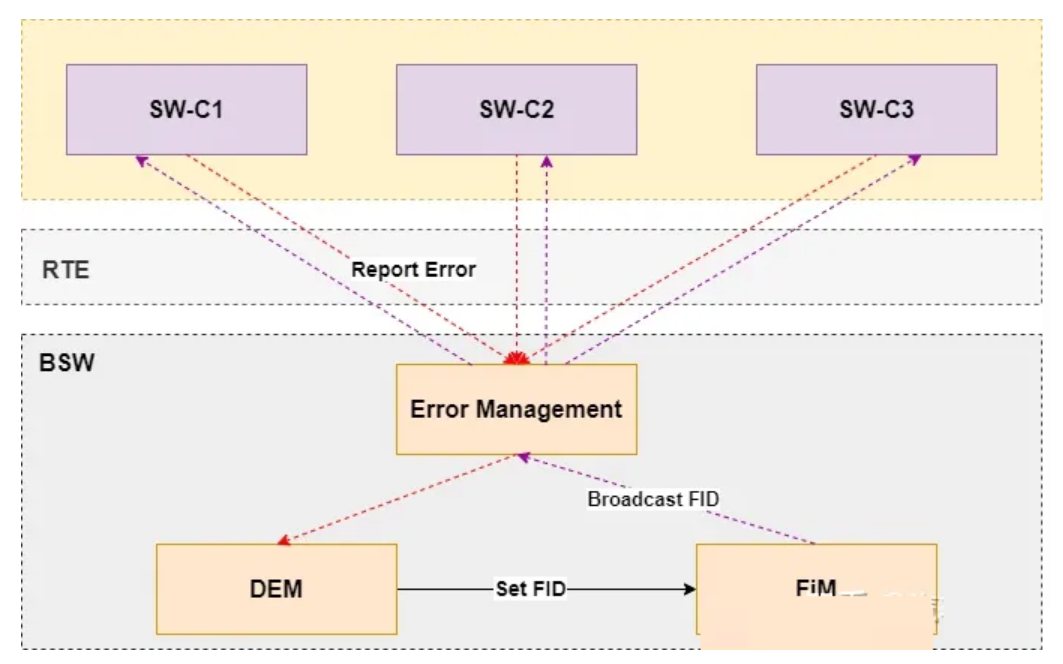
FiM模块提供了一种基于抑制场景的功能降级策略，同时该抑制场景可配置；

可通过特定的Function Identifier来实现相应的抑制场景；

抑制场景可基于Dem模块的Event在FiM模块中根据降级需求完成映射；

FiM模块为BSW以及SW-C各模块的抑制场景提供了一种有效的降级手段；

简而言之，当SW-C或者BSW层发生故障发生时，会将故障即Event上报给到DEM模块，DEM模块会通过某种关系通知到FiM模块，FiM模块通过Function Identifier再次去告知SW-C以及BSW层做出相应的降级行为。如下图所示，以SW-C模块上报故障为例，较为清晰了表示了彼此之间的作用关系：

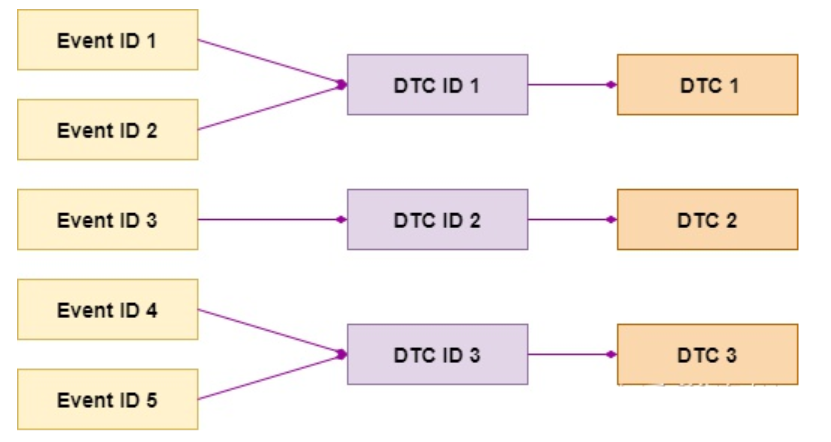


其中，红色虚线表示上报故障Event，蓝色虚线表示广播FID状态。

基于上图，清楚了抑制场景可以通过Function Identifier(以下简称FID)来表示，某个抑制场景FID跟某个特定故障Event进行Mapping（即一对一关系），也可以是某个FID与多个Event进行Mapping（即一对多关系）。

在此可能会联想起Event与DTC也存在一对一或者一对多的关系。鉴于Dem模块与FiM模块密切相关，所以一并于此梳理下Event 、DTC、FID三者之间的映射关系，防止出现不必要的混淆。

如下图所示，描述了Event ID、DTC ID、DTC三者之间的映射关系。

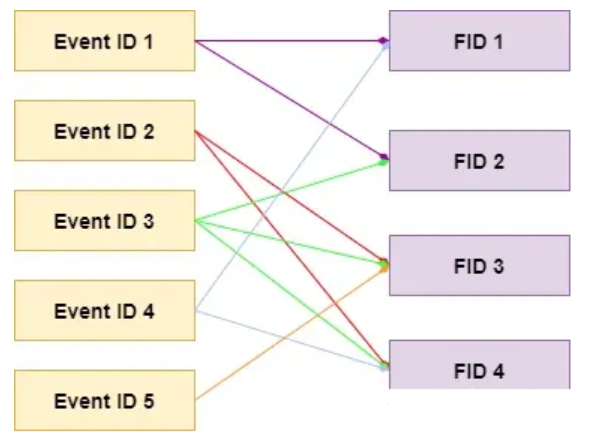


注意：在上图中的DTC ID与常说的DTC会有所区别，DTC ID指的是软件内部自动给每个DTC定义的编号（一般从0...n），这部分编号可以通过配置工具生成，而DTC则就是客户要求定义的三个字节的DTC Number。

由上可知，Event ID与DTC ID既可以是1对1关系，也可以是对对1关系，同时DTC ID与DTC则始终是1对1的关系，最终也就是Event ID与DTC之间既可以是1对1，也可以是多对1关系。

之所以存在1对多关系，主要是为了便于对Event 进行分类整合，减少DTC的数目，以便于我们对故障快速分析，对于具体是哪个故障，可通过相关诊断服务进行获取，不同供应商可能做法会不一致，如通过1906或者内部诊断服务来进行获取相应的Root Cause。

如下图所示，描述了Event ID与FID之间的映射关系。



注意：图中Event ID也是软件内部给每个Event定义的编号（范围从0...n），这部分也是通过工具自动生成。

由此可见，Event ID与FID之间可以实现1对1关系，即某个故障会导致功能层面某个抑制场景；同理，两者也可以是实现多对1关系，即某类events发生时，就都会导致同一功能抑制场景的发生，至于两者如何实现映射关系，后文会进行讲述。

通过上述几个部分的介绍，我们知道FiM的主要功能逻辑就是基于Dem模块上报的Event状态，来触发相应的FID，然后BSW层或者SW-C层相关子模块根据这些FID状态来实现相应的功能抑制场景（俗称：功能降级）。

FiM模块主要功能：获取来自Dem模块的Event Status，置位FiM模块的FID。

**FiM模块基本参数**

在下图看到有Function Inhibition（以下简称FID），Event Monitor Status来自于Dem模块的输出，Inhibition Mask是FID置位的前提条件，它会与Event Monitor Status进行与运算。

Function Identifier Counter（以下简称FID Counter），用于描述当前导致FID置位的Event数目，该Counter会随着导致FID置位的Event数目增加而增加，减少而减少。



**FiM模块与Dem模块交互关系**

FiM模块的输入是来自Dem模块的Event Status，输出就是FID的状态。在此需要特别介绍几个参数取值如下：

Event Status: 即Byte的Event status，每一个Bit代表相应的含义；

Inhibition Mask：用于决定什么样的Event Status会触发上述FID置位，有以下三种取值：

Inhibit if Failed：当Event Status Bit0 = 1时，该值就为1，否则为0；

Inhibit if Tested: 当Event Status Bit6 =0时，该值为1，否则为0；

Inhibit if Not Tested: 当Event Status Bit6 =1时，该值为1，否则为0；

FID: 全称Function Identifier，取值就是Bool类型 TRUE or FALSE,举例FID\_1说明如下：

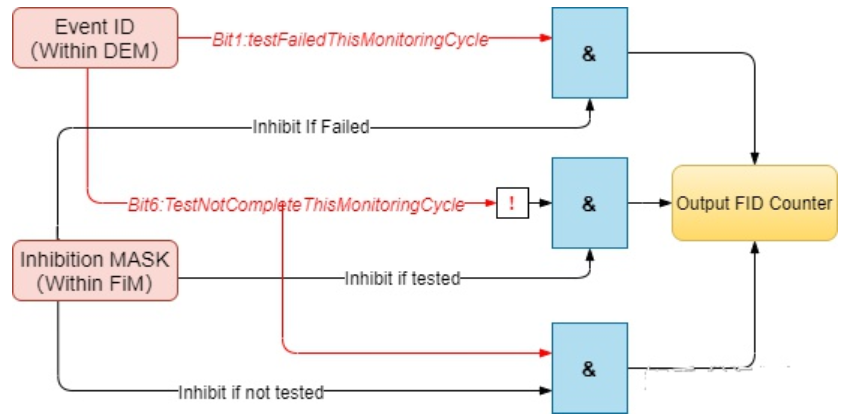
FID\_1 == TRUE: 即抑制场景不激活，功能不降级；

FID\_1== FALSE：即抑制场景激活，功能降级；

FID Counter：用于表征当前存在几个Event同时发生导致了该类型的抑制场景激活，进而功能降级；

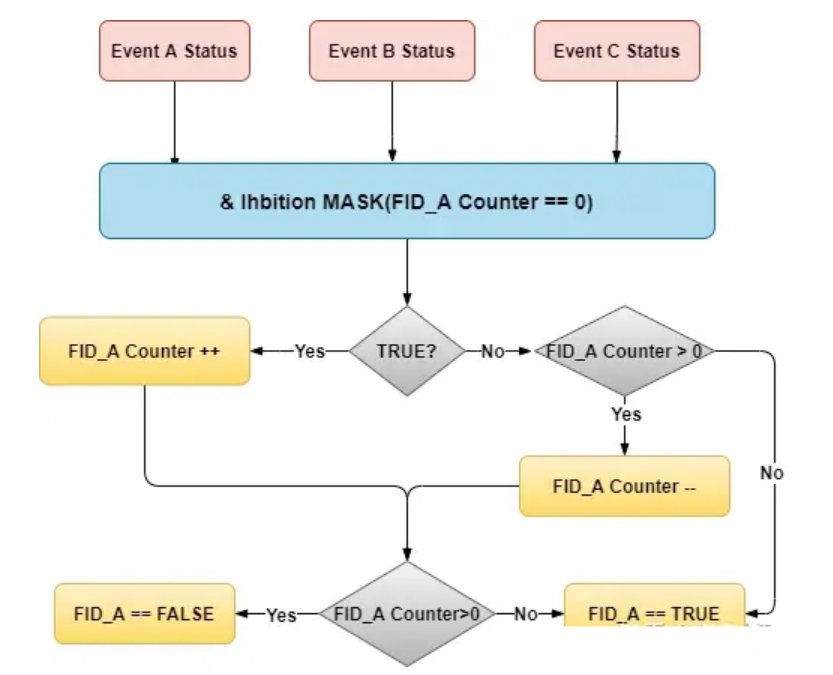
如下图5所示，表示了来自Dem模块的Event Status，与FiM模块中的Inhibition Mask、FID、FID Couner这四者之间的计算关系。

如下图所示，表示了来自Dem模块的Event Status，与FiM模块中的Inhibition Mask、FID、FID Couner这四者之间的计算关系。



由图可知Event ID的状态与Inhibition MASK是位与的关系，如果位与的结果为1，那么相应Mapping 的FID Counter就会加1。其中每个FID只能选择其中一种类型Inhibition MASK，这个是在工具中静态配置，不支持动态更改。

如果多个Event ID发生且与Inhibition MASK位与的结果为1时，那么FID Counter又该如何表现呢，如下图所示：



如上图所示，假设Event A，EventB、EventC均Map了FID\_A,也就意味着当这三个Event中任意一个或多个发生时，FID\_A就会置位，即FID\_A == FALSE。另外，每一个FID Counter初始值均为0且值永远大于等于0。

系统降级算法规则：当FID\_A Counter >0, FID\_A就会置位，即为FALSE,相应的系统降级行为就会发生。

当下列情况时，FID\_A Counter 如何变化呢？



通过上述场景分析得知，只要FID\_A Counter不等于0，FID\_A 就会等于FALSE，系统就会降级。不过在此需要特别提醒的是虽然此时你知道系统降级了，但是从FID你无法知道到底是哪个Event导致的降级，此时就需要通过相关的诊断服务读取内部的Event并且该Event也确实Map了该FID，那么你就可以锁定该Event就是原因所在。