# IFC

APT32F102x的内部flash包括两个区域，一个是64K/32K的PFLASH，用于程序代码的存储，一个是4K的DFLASH，用于用户数据的存储。这两个区域最大的区别是扇区大小不同。

无论是DFLASH还是PFLASH，一次写操作之前必需有一次擦除操作（以扇区为单位）。从FLASH寿命角度考虑，擦和写的次数需要对等。正是出于这个考虑，驱动不再支持单独的擦除操作，写操作函数中实现先擦除再写。

## API列表

Table 8‑1 IFC CSI接口函数

|  |  |  |
| --- | --- | --- |
| API | 说明 | 函数位置 |
| csi\_ifc\_read | 读取flash区域的内容，支持PFLASH和DFLASH | ifc.c |
| csi\_ifc\_program | 往flash区域写入内容，支持PFLASH和DFLASH |
| csi\_ifc\_get\_status | 获得flash状态（error, busy） |  |

## API详细说明

### csi\_ifc\_read

csi\_error\_t csi\_ifc\_read(csp\_ifc\_t \*ptIfcBase, uint32\_t wAddr, uint32\_t \*wData, uint32\_t wDataNum)

#### 功能描述

读取flash区域的内容，支持PFLASH和DFLASH。发生以下情况返回错误：

1. 传入地址没有字对齐。
2. 地址不在系统flash（PFLASH+DFLASH）空间

#### 参数说明

|  |  |  |
| --- | --- | --- |
| 参数 | 说明 | 位置 |
| ptIfcBase | 指向IFC控制寄存器结构体的指针，用于进行寄存器操作。 | csp\_ifc.h |
| wAddr | 读取Flash数据的首址 |  |
| wData | 指向目标数据首址的指针 |  |
| wDataNum | 一次读取数据的长度（单位：byte） |  |

#### 返回值说明

|  |  |  |
| --- | --- | --- |
| 返回值类型 | 说明 | 位置 |
| csi\_error\_t | CSI\_ERROR/CSI\_OK | common.h |

### csi\_ifc\_program

csi\_error\_t csi\_ifc\_program(csp\_ifc\_t \*ptIfcBase, uint32\_t wAddr, uint32\_t \*pwData, uint32\_t wDataNum)

#### 功能描述

往flash区域写入内容，带校验功能。支持PFLASH和DFLASH。发生以下情况返回错误：

* 传入地址没有字对齐
* 地址不在系统flash（PFLASH+DFLASH）空间
* 校验失败

#### 参数说明

|  |  |  |
| --- | --- | --- |
| 参数 | 说明 | 位置 |
| ptIfcBase | 指向IFC控制寄存器结构体的指针，用于进行寄存器操作。 | csp\_ifc.h |
| wAddr | 操作Flash的目标首址 |  |
| wData | 指向源数据首址的指针 |  |
| wDataNum | 一次写入的数据的长度（单位：byte） |  |

#### 返回值说明

|  |  |  |
| --- | --- | --- |
| 返回值类型 | 说明 | 位置 |
| csi\_error\_t | CSI\_ERROR/CSI\_OK | common.h |

### csi\_ifc\_get\_status

csi\_ifc\_status\_t csi\_ifc\_get\_status(csp\_ifc\_t \*ptIfcBase)

#### 功能描述

读取Flash的状态。

#### 参数说明

|  |  |  |
| --- | --- | --- |
| 参数 | 说明 | 位置 |
| ptIfcBase | 指向IFC控制寄存器结构体的指针，用于进行寄存器操作。 | csp\_ifc.h |

#### 返回值说明

|  |  |  |
| --- | --- | --- |
| 返回值类型 | 说明 | 位置 |
| csi\_ifc\_status\_t | typedef struct {      uint32\_t busy  : 1;  // busy状态      uint32\_t error : 1;  //错误状态（IFC\_RISR[xxx\_ERR]）标志  } csi\_ifc\_status\_t; | ifc.h |