FF混合控制方案

实现思路

# 硬件配置模块生成链接关系

## 创建下装数据



Control->Link->Gateway->Slave

通过调用控制器的创建下装数据，调用CreateCfgModule创建数据，之后调用WriteDownloadmemFile下装数据

FF协议中CreateProtocolDownloadData，数据存放在CGeneralSlave::m\_pCfgModule->pSpecialData或

CGeneralLink::m\_pCfgModule->pSpecialData中

## 数据源

数据源是m\_LinkFBApplication，通过链路（主站）对象pLinkObj->GetItemValue(STR\_PROTOCOL\_DATA)获取

数据源定义：

m\_LinkFBApplication

1:n

FBApplication

1:n

FBAppDeviceInfo(m\_DeviceInfoMap)

----1:n

CONNECT\_SUB\_LIST(m\_connectMap)

1:n

CONNECT(uiPubChannelID、uiSubDeviceID、usSubBlockIndex、uiSubChannelID)

----1:n

LINKPUB(m\_listLinkPub) 连接到其它设备，自己加pub，其它加sub

----1:n

LINKTOVFB(m\_listLinkInternal)设备内部连接

----1:n

LINKSUB(m\_listLinkSub)

## 结论

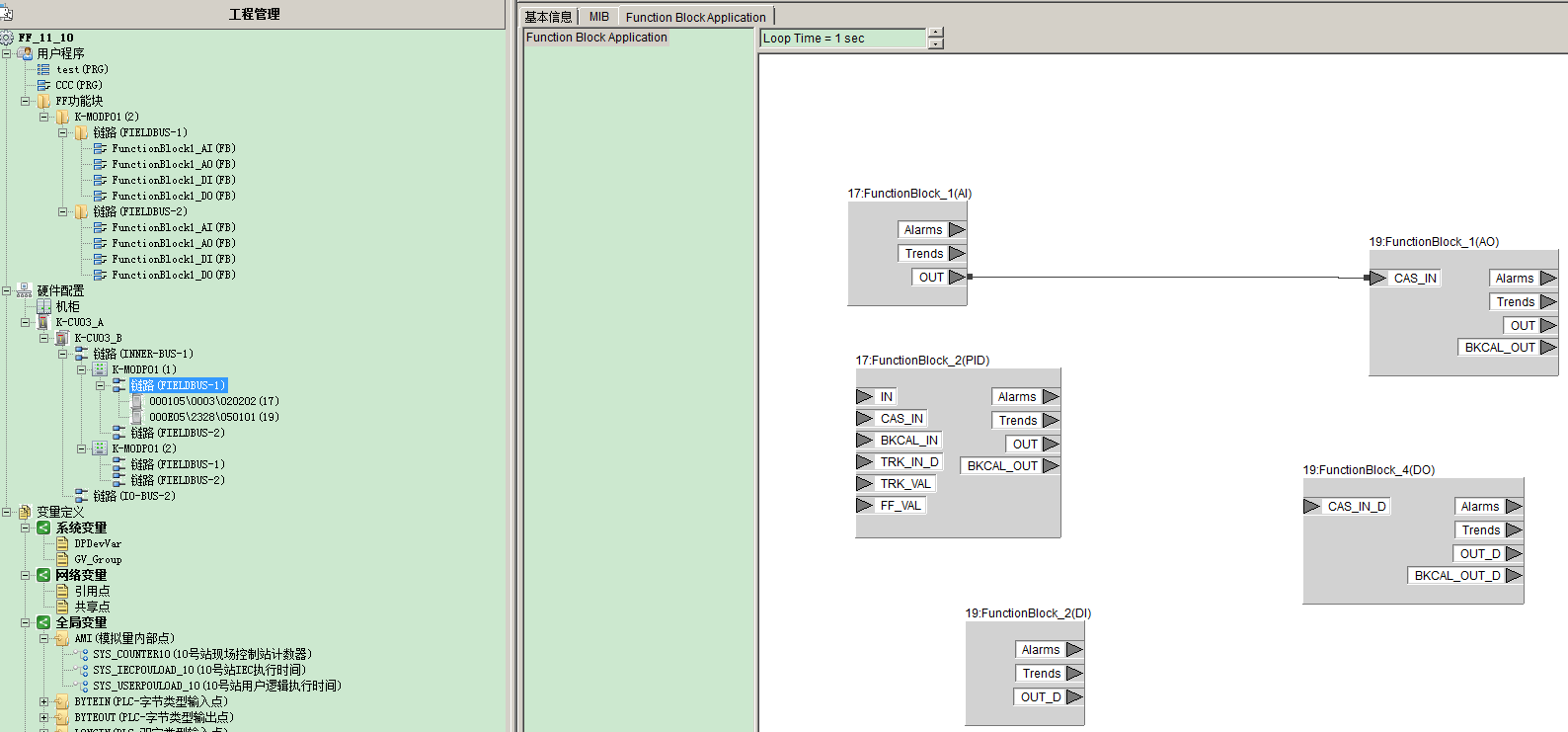
1. 数据源切换（函数块应用）

从站的数据源切换，还有主站的虚拟功能块支持

1. 下装的数据来源、意义

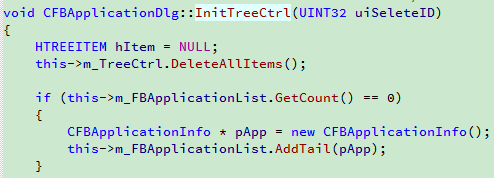
# 从站数据源切换

## Function Block Application构建



### 新建工程





初始装，strData为空，此时m\_FBApplicationList为空，在执行InitTreeCtrl时，会自动创建一个空的FBApplication加入到m\_FBApplicationList，在ChangeFBApp中赋值当前应用。

### 添加函数块

如下图，右键菜单，将Link子设备索引作为ucMainMenuID，子设备模块索引作为ucSubMenuID，拼凑为菜单ID



响应函数里，解析设备索引和模块索引



目的是构造出FBApplication的设备信息和块信息

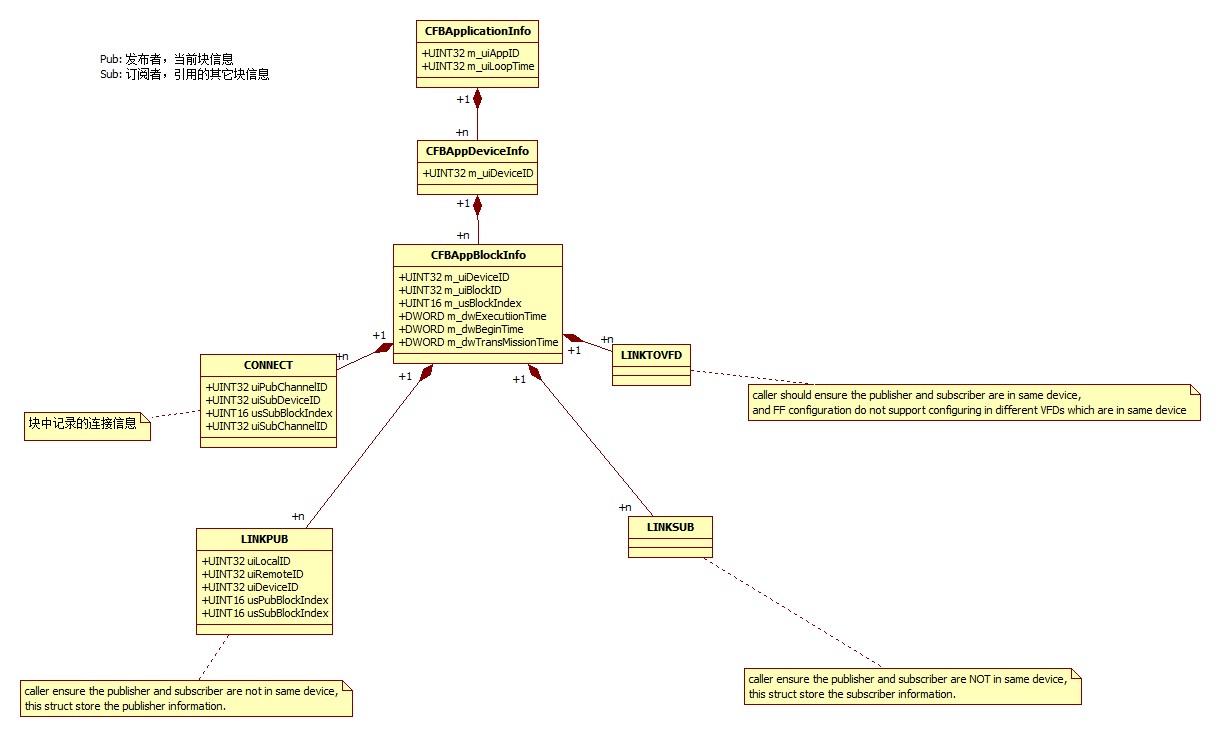


核心信息：链路下设备索引和模块索引

## 数据生成流程

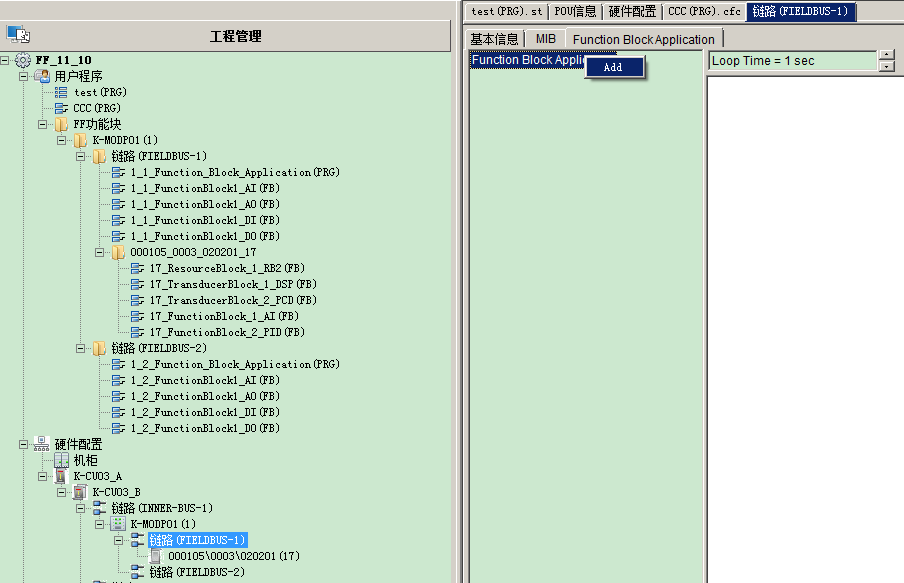
### 数据结构

如下：



### CFBAppliactionInfo生成

双击Link创建初始CFBAppliactionInfo，或在树有右键新增添加



### CFBAppDeviceInfo生成

说明：

设备信息

添加：

右键菜单添加设备时，填充m\_DeviceInfoMap



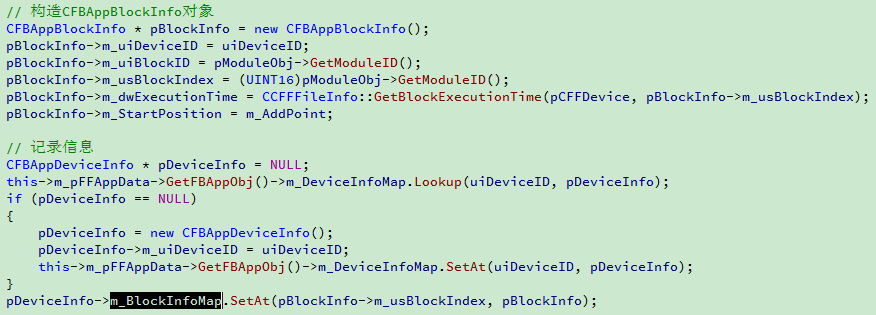
### CFBAppBlockInfo生成

说明：

应用块信息

添加：

生成CFBAppDeviceInfo时，生成对应的BlockInfo，目前逻辑，虽然关系是1:n，但实际是1:1



### CONNECT生成

说明：

被引用信息，即当前块指定引脚（发布者）关联的其它块的引脚信息（订阅者）

添加：

在鼠标拖动完毕时添加



### LINKPUB生成

说明：

无

添加：

不用主动添加，在下装时根据现有数据生成，见：CFBApplicationDlg::UpdateLinkageByConect

### LINKSUB生成

说明：

无

添加：

不用主动添加，在下装时根据现有数据生成，见：CFBApplicationDlg::UpdateBlockLinkSub

### LINKTOVFD生成

说明：

无

添加：

不用主动添加，在下装时根据现有数据生成，见：CFBApplicationDlg::UpdateLinkageByConect

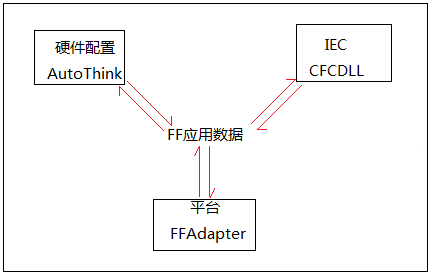
### 结论

关注数据有效性和添加方式

|  |  |  |
| --- | --- | --- |
| **数据** | **添加方式** | |
| **Function Block Application** | **IEC** |
| CFBAppliactionInfo | 双击Link或右键树添加 | 双击Link = 创建POU时自动生成 右键树添加 = 暂无方法，待考虑 |
| CFBAppDeviceInfo | 添加设备时添加 | 拖拽函数块添加 |
| CFBAppBlockInfo | 添加设备时添加 | 拖拽函数块添加 |
| CONNECT | 拖拽连线时添加 | 拖拽连线时添加 |
| LINKPUB | 下装时自动生成 | 无 |
| LINKSUB | 下装时自动生成 | 无 |
| LINKTOVFD | 下装时自动生成 | 无 |

## 数据切换

当前FF应用数据，在平台FFApdpter中，抽提后，需要保证硬件配置、IEC、平台都可以访问，首先在AT硬件配置操作添加主站、从站，FFAdapter将消息发送至框架，在框架里实现FF文件夹和POU、FB的创建，同时生成FF应用数据，之后在IEC操作中，填充FF设备信息、FF块信息、FF连接等信息，见下图：



### FF应用数据切换

#### 添加方式1

新建设备，打开链路，初始化FF应用对话框添加

硬件配置添加设备后，自动创建POU，

FBAppInfo初始添加

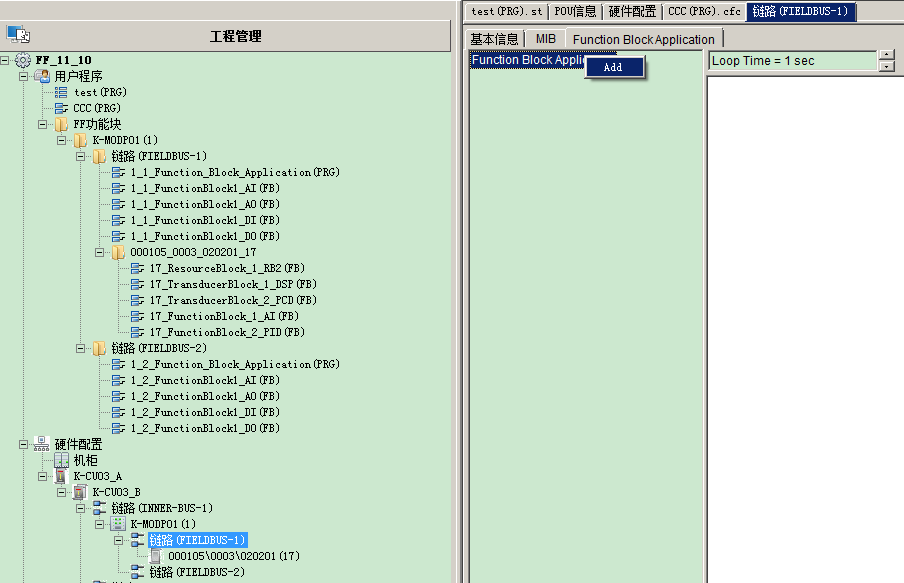




结论：支持，创建主站文件夹时，自动创建

#### 添加方式2

FBAppInfo新增，可以手动新增，即主站下除了默认的FF函数应用，还可以再加



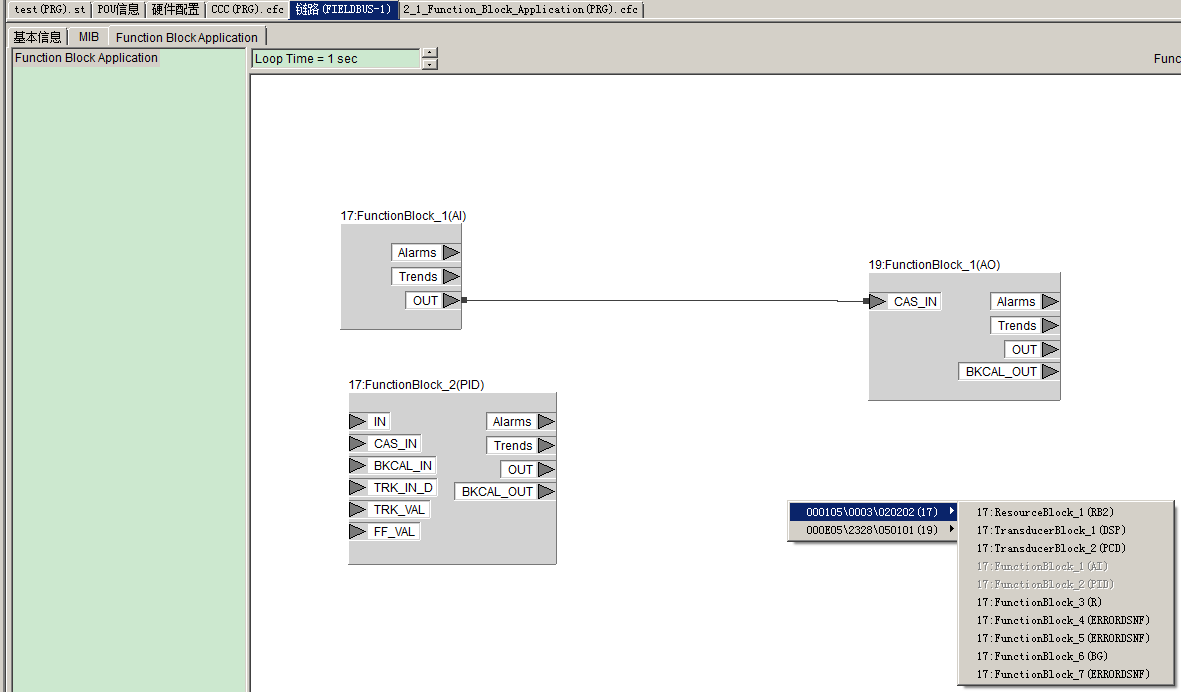


结论：数据切换，暂不支持

### FF设备数据切换

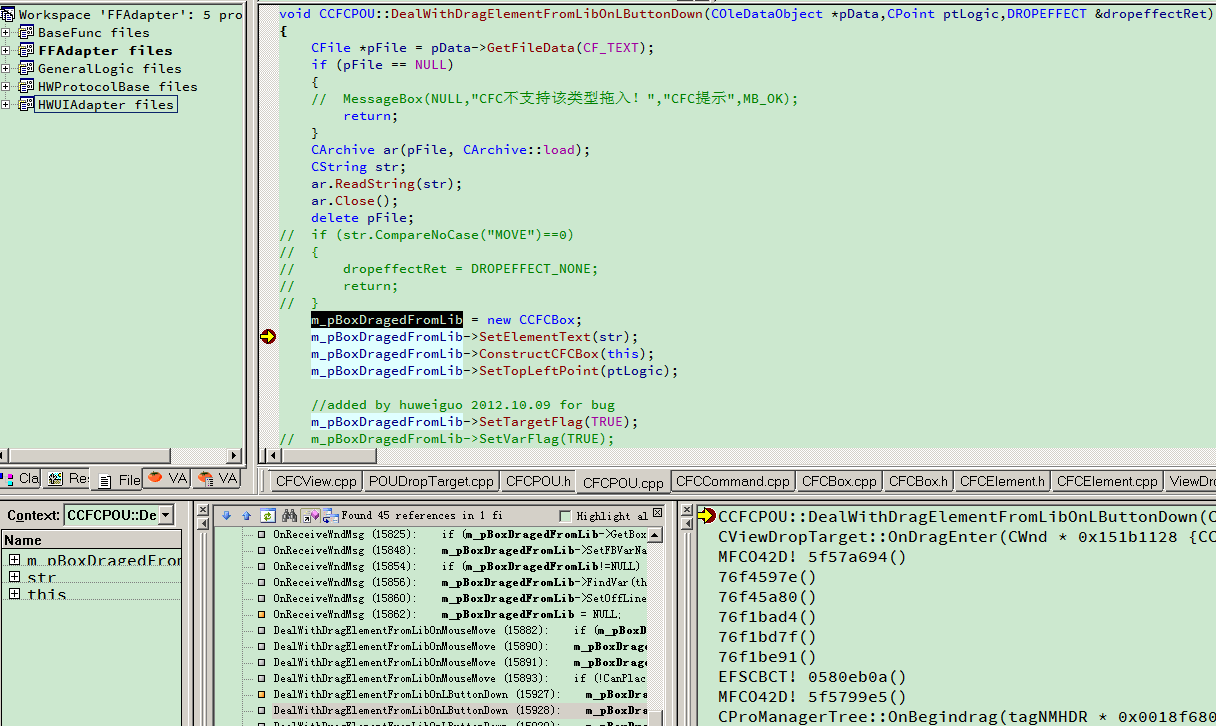
#### 添加方式

右键菜单添加



结论：

1. CFC中通过拖拽添加，拖拽处理函数



1. 拖拽实现



这里需要调整CFCFFInfo的作用域，之前是只AutoThink调用，现在IEC也调用，将该数据结构的定义实现为类，并提为公共类

1. 控制设备数据只能拖入1次



若已包含FF的设备块，则不能再拖入

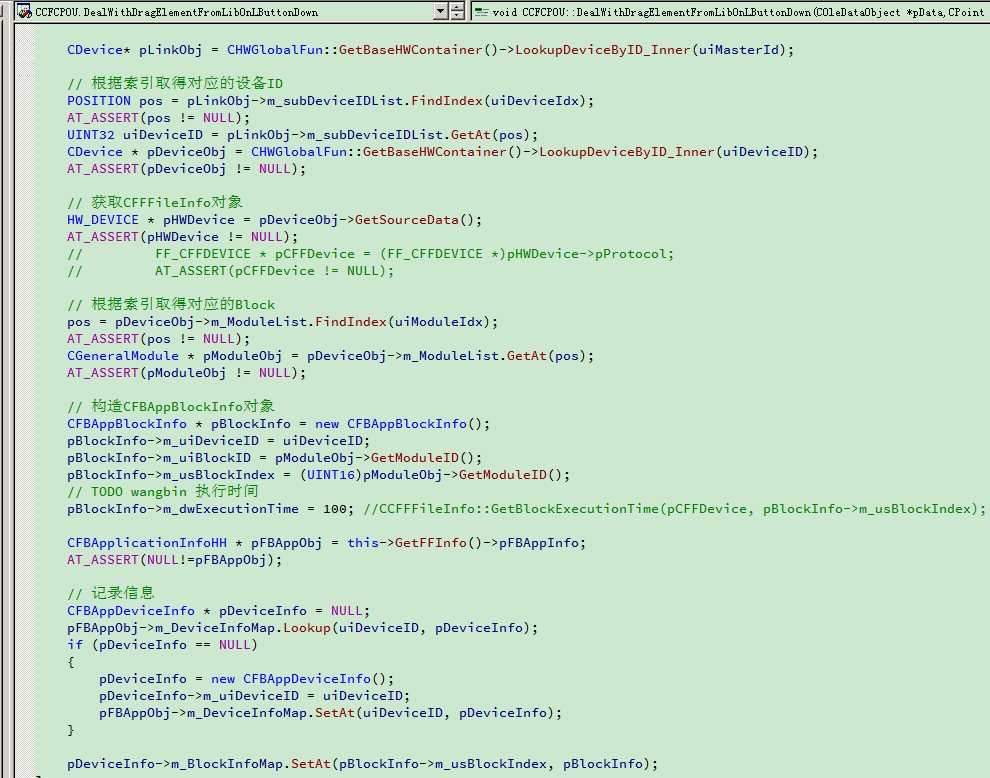
### FF块数据切换

#### 添加方式

同FF设备数据切换添加方式，在添加设备时，1对1生成设备数据信息和块数据信息

结论：

在拖拽CFC函数块时，生成对应的设备数据和块信息数据



#### 删除方式

在删除CFCBox块时，判断是否FF功能块



### FF连接数据切换

#### 添加方式

拉线

点下鼠标左键



放开鼠标左键



结论：

模拟FF连接模式，在CFC连接时，创建连接点



实现

