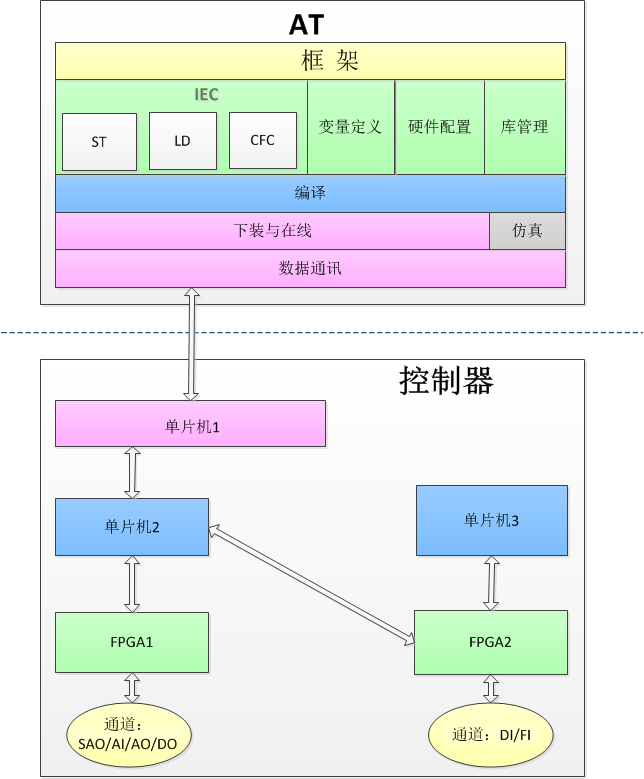
ETM在线下装总结

# ETM架构



AT

1. 框架：AutoThink的用户操作界面
2. IEC：为用户提供三种IEC语言（ST、LD、CFC）的编程环境，并将IEC程序以POU为单位转换为AutoThink内部的一种存储结构，即中间语言或中间代码。
3. 变量定义：支持用户定义或修改变量
4. 硬件组态：支持用户进行硬件模块配置，如主控单元，I/O模块等，并生成控制器所需的硬件模块信息。
5. 库管理：支持用户使用库函数，如添加，删除算法库，或在算法库中查找POU的声明
6. 编译：该模块通过调用HCC编译器，将每一个用中间语言描述的POU编译成二进制指令，然后将这些二进制形式的POU链接成控制器可加载的文件形式（\*.AT），并进行CRC32校验。
7. 下装与在线：
8. 下装：用户可以将AutoThink编译生成的\*.AT文件下装给控制器，该文件包含二进制指令和硬件组态信息；
9. 在线：当控制器正在运行IEC任务时，AT可以连接控制器，监视二进制程序的运行，并实时显示运行结果。
10. 仿真：在AT内部模拟运行和调试IEC程序

控制器

1. 单片机1：主要实现TCP/IP协议解析（与AT交互）
2. 单片机2：主要实现IEC运算和与AT软件的交互
3. 单片机3：主要实现profisafe协议解析（与ITCC交互）
4. FPGA1：主要实现一套ETM281系统中三个模块间的同步、高速数据通讯和部分通道功能
5. FPGA2：主要实现profisafe协议解析和部分通道功能

# 在线功能



1. 设置通讯参数：从总控启动时获取站号和域号，采用自适用IP的做法，用以实现与控制器通讯。
2. 下装：下装组态信息到控制器。
3. 初始化下装：即首次下装，下装完整的组态信息到控制器中，IEC运算需要停止。
4. 增量下装：下装组态逻辑变化的部分到控制器，IEC运算不停止。
5. 在线：程序处于在线状态，实时监视变量的在线值。
6. 离线：程序处于离线状态，断开与控制器的连接。
7. 强制：在线状态下，强制某变量为一固定值。
8. 释放
   1. 部分释放：对若干已强制的变量解除强制限制。
   2. 全部释放：对全部已强制的变量解除强制限制。
9. 写入：在线状态下，对某变量赋值，该变量按新值参与IEC运算。
10. 仿真：将组态逻辑下装到仿真控制器中，并在线监视。
11. 日志：记录使用软件的一些操作过程和错误信息等。

# 在线状态机

在线状态机状态：

1. IDLE：空闲态
2. ToMonitor：在线中间态：在线操作由IDLE态跳转为ToMonitor态，有任何异常再跳转为IDLE态；离线结束由ToMonitor态跳转为IDLE态。
3. OnMonitor：在线状态：离线操作由OnMonitor态跳转为ToMonitor态；ToMonitor态跳转为OnMonitor态不由用户操作触发，由在线后台处理（读取强制列表值应答后跳转）。
4. ToSimu：仿真中间态：仿真操作由IDLE态跳转为ToSimu态，有任何异常再跳转为IDLE态
5. OnSimu：仿真态：
6. OnDownLoad：下装态：下装操作由IDLE态跳转为OnDownLoad态，下装结束或任何异常再跳转为IDLE态



# 数据交互



# 在线工作原理





在线模块的所有功能都是在与控制器之间的交互中完成的，在线模块发送相应的服务组合到控制器，并处理控制器的应答，完成该功能。对于每个服务，都是通过请求应答方式进行的。即由在线模块发送服务请求到控制器，控制器对服务请求进行处理，然后将结果返回。

针对仿真模式和非仿真模式，处理方式略有不同。在仿真模式下，通过调用在线模块的仿真通讯接口实现与仿真控制器的通讯，而在非仿真模式下，通过调用通讯模块的接口实现与控制器的通讯。

# 数据结构设计



## CBaseOnlineContainer类

CBaseOnlineContainer是在线的总类，其为站对象CStation的一个私有成员。它提供在线的入口函数。



## CStation类

该类主要定义了收集各模块变量的容器和AT各模块接口函数，CSation类主要提些容器和变量，CbaseOnlineContainer类主要提供了一些在线处理的操作，是CStation类的一个私有成员。

## COnlineContainer类

COnlineContainer类是CBaseOnlineContainer类的派生类



## COnlineGlobal类

该类主要能提供实现一定功能的函数接口。



# 在线关键操作流程

在线模块的状态：在线（监视）、离线、下装、仿真。对于在线、下装和仿真操作都是由一系列服务所组成，而这些服务必须按照严格的顺序执行，任何一个服务出问题，都认为该次操作失败。

COnlineContainer类中的成员m\_arrServiceList，用于存储当前操作的服务列表，进入每个操作的入口时，调用初始化该列表的函数，初始化该操作对应的服务列表。

COnlineContainer类中的成员m\_iServiceNum，用以标识当前服务列表中已经下发的服务号，m\_iServiceListLen成员用于表示当前服务列表的长度。每个操作初始化完成后会在SendNextService中从服务号0开始下发服务；RTS应答后，在ReplyLastService中

m\_arrServiceList[m\_iServiceNum]表示当前等待应答的服务；。

对于每一个服务，在COnlineContainer类中有两个函数，分别用于发送服务处理函数和接收服务处理函数。发送服务处理函数准备要发送数据，然后发送本次服务。每个服务对应唯一的发送服务处理函数；接受服务处理函数接收本次服务的应答数据，处理后根据程序当前的状态再发送不同的后续服务，同样每个服务对应唯一的接受服务处理函数。

在每个操作的入口处查看该列表中的第一个服务，调用该服务发送处理函数，RTS响应后，接受通讯模块消息（Communicator）调用该服务的应答处理函数，接受应答正确，m\_iServiceNum++，再发送列表中的下一个服务，再接收服务应答，直至列表中的服务全部发送完毕或者中间出现服务应答错误或者网络异常。



附录1：在线模块的入口函数，提供给框架或者其他模块调用。

* + - 下装：char HandleDownload()
    - 在线：char HandleOnline()
    - 离线：char HandleOffline()
    - 切换：char HandleOLSwitch()
    - 撤销在线下装：char HandleCancelOLChange()
    - 写入：char HandleWrite(bool bBatch)
    - 强制：char HandleManu(bool bBatch)
    - 部分释放：char HandleUnmanuPart()
    - 全部释放：char HandleUnmanuALL()
    - 下载文件：char HandleWriteProFile()
    - 上传文件：char HandleReadProFile(CString strFilePath)
    - 仿真：char HandleSimu()
    - 退出仿真：char HandleExitSimu()
    - 运行：char HandleRun()
    - 停止：char HandleStop()
    - 冷复位：char HandleResetCold()
    - 热复位：char HandleResetWarm()
    - 清空控制器：char HandleResetHard()

附录2：响应各操作应答入口函数：ReplyAllHandle

|  |  |  |  |
| --- | --- | --- | --- |
| 操作名（handle） | 操作值 | 说明 | 处理函数 |
| HANDLE\_IDLE | 0 | 离线 |  |
| HANDLE\_OFFLINE | 1 | 退出监视 | HandleOfflineReply |
| HANDLE\_ONLINE | 2 | 监视 | HandleOnlineReply |
| HANDLE\_INIT\_DOWNLOAD | 3 | 初始化下装 | ReplyLastService |
| HANDLE\_CHANGE\_DOWNLOAD | 4 | 增量下装 | ReplyLastService |
| HANDLE\_WRITE | 5 | 写入 | HandleWriteReply |
| HANDLE\_MANU | 6 | 手动 | HandleManuReply |
| HANDLE\_UNMANUPART | 7 | 部分自动 | HandleUnmanuPartReply |
| HANDLE\_UNMANUALL | 8 | 全部自动 | HandleUnmanuALLReply |
| HANDLE\_SIMULATION | 12 | 进入仿真 | HandleSimuReply |
| HANDLE\_EXIT\_SIMULATION | 13 | 退出仿真 | HandleExitSimuReply |
| HANDLE\_START | 14 | 运行 | HandleRunReply |
| HANDLE\_STOP | 15 | 停止 | HandleStopReply |
| HANDLE\_RESET\_HARD | 16 | 清空控制器 | ReplyLastService |
| HANDLE\_READPRO | 17 | 从控制器上传工程文件 | ReplyLastService |
| HANDLE\_WRITEPRO | 18 | 下装工程文件到控制器 | ReplyLastService |
| HANDLE\_RESET\_WARM | 19 | 热复位 | HandleResetWarmReply |
| HANDLE\_RESET\_COLD | 20 | 冷复位 | HandleResetColdReply |
| HANDLE\_RESET\_PRG | 21 | 清空存储器 | ReplyLastService |
| HANDLE\_TRANS\_PRG | 22 |  | ReplyLastService |
| HANDLE\_AUTO\_SELECT | 23 | 试探串口是否连接硬件 | HandleAutoSelectReply |
| HANDLE\_SUSPEND | 24 | 暂停？？？？ | HandleSuspendReply |
| HANDLE\_MODIFY\_IP | 25 | 修改IP | HandleModifyIPReply |
| HANDLE\_STOP\_CONTROL | 26 | 停止控制器 | HandleStopControlReply |
| HANDLE\_WANTCH\_DOG | 27 |  | HandleWatchDogReply |
| HANDLE\_ADJUST\_TIME | 28 | 调整时间 | HandleAdjustTimeReply |
| HANDLE\_DEBUG | 29 | 调试 | HandleDebugReply |
| HANDLE\_TRANSFILE\_UNIVERSAL | 30 |  | ReplyLastService |
| HANDLE\_OLSWITCH | 31 | 在线切换 | HandleOLSwitchReply |
| HANDLE\_OLINITVAR | 32 |  | ReplyLastService |
| HANDLE\_READ\_VERSION | 33 | 读取版本句柄 | ReplyLastService |
| HANDLE\_CANCEL\_OLCHANGE | 34 | 撤销在线切换 | HandleCancelOLChangeReply |
| HANDLE\_USER\_CONFIRM | 35 | 用户确认？？? | HandleUserConfirmReply |

附录3：下装ID、编译ID关系：



附录4：RTS编译平台

|  |  |
| --- | --- |
| X86 | 仿真 |
| PPC(LINUX) | DCS |
| XE164C(单片机) | PLC,ETM281 |
| XE164(单片机) |
| ARMV7 | MC |

问题汇总

问题1：

Handle状态HANDLE\_OLINITVAR？

新增变量提示服务

问题2：

在哪里设置初始化下装还是增量下装

在begindownload中设置下装标志（初始化下装、增量下装），在setservicelist中初始化下装服务列表，在sendnextservice中按照顺序发送服务，在replylastservice中接受应答服务并做相应的处理。

问题3：

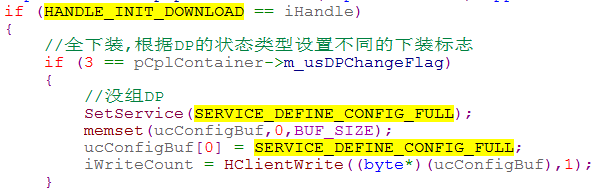
判断RTS状态时，为什么要先断开RTS连接在连接RTS

问题4：

每组操作完成后重新计算日志校验和并更新日志中的校验和及日志内容，再次打开日志后需要计算校验和并与日志中的校验和比较，这么做的意义

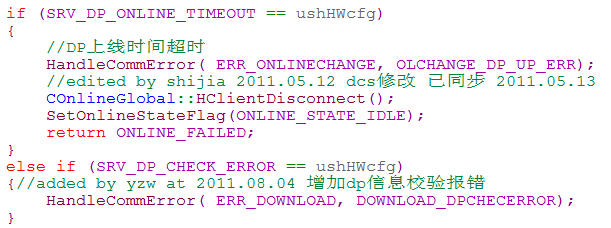
问题5：

发送硬件配置服务SendDefineConfig中，m\_usDPChangeFlag为3时，会设置SERVICE\_DEFINE\_CONFIG\_FULL服务，但在应答中无该服务。



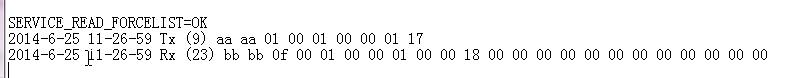
问题6:

应答硬件配置服务ReplyDefineConfig中，



问题7：

读取强制变量列表服务



建议：

* + - 1. 下装过程复杂，最好不要用长函数来实现，重构
      2. 下装、在线服务列表较长，最好不要用case分支，可以用表驱动实现，简单明了
      3. 消灭代码中的魔鬼数字SetOLSwitchConfirm
      4. 建议派专人维护《通讯说明书》，开发人员应以说明书作为开发依据，而非口头约定
      5. 建议在线所有的异常后处理封装成统一函数，初始化所有状态机
      6. PLC代码从DCS代码衍生出来的，许多废弃代码应清理

需要优化函数：

1. HandleDownload
2. SetDBOnlineValue
3. ReplyLogout

……