文件名称：AT与总控交互总体设计

文件编号：

项目名称：过程自动化产品平台研究

项目编号：R-281501L03

物料编码：

版 本 号：A

文件密级：机密

文件状态：CFC

受控标识：受控

|  |  |
| --- | --- |
| 拟制： | 李保霖 2015年 05 月 05 日 |
| 审核： | 李宗杰 2015年 月 日 |
| 会签： |  |
|  |  |
| 批准： | 李宗杰 2015年 月 日 |

**文件发放范围**：

**修订页**

| 序号 | 版本号 | 修订内容简述 | 拟制/日期 | 审核 | 批准 |
| --- | --- | --- | --- | --- | --- |
|  | A | 创建 | 李保霖/ 2015.05.05 | 李宗杰 |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |

目 录

[1. 介绍 2](#_Toc464129576)

[1.1 目的 2](#_Toc464129577)

[1.2 文档概述 2](#_Toc464129578)

[1.3 术语与缩略语 2](#_Toc464129579)

[1.4 引用文档 2](#_Toc464129580)

[2. 主要功能点设计 2](#_Toc464129581)

[2.1 硬件配置数据上传总控 2](#_Toc464129582)

[2.2.1 硬件配置导出格式定义 3](#_Toc464129583)

[2.2.2 数据结构 3](#_Toc464129584)

[2.2.3 数据结构定义 4](#_Toc464129585)

[2.2.4 与总控交互 6](#_Toc464129586)

[2.2.5 数据库导出 6](#_Toc464129587)

[2.2 硬件配置数据导入AT 7](#_Toc464129588)

[2.2.1 旧硬件配置导入AT结构（参考） 7](#_Toc464129589)

[2.2.2 新硬件配置导入AT结构 7](#_Toc464129590)

[2.2.3 与总控交互 9](#_Toc464129591)

[2.2.4 数据库导入 9](#_Toc464129592)

[2.3 类结构中增加“位置”项 9](#_Toc464129593)

[2.4 诊断点名称同步 9](#_Toc464129594)

[2.5 K-CU01切换K-CU03 10](#_Toc464129595)

# 介绍

## 目的

本文档是基于多总线结构平台的AT系统总体设计，提取出来的AT与总控交互的子模块设计。相较于AT的系统设计，该子模块设计是AT系统设计的进一步深化和细化，为模块的概要设计和开发确定目标和方向。

本文档是进行后续软件设计与开发、测试的依据。

本文档的预期读者是开发设计人员。

## 文档概述

本文档主要基于多总线硬件配置平台结构调整下，描述DCS\_AT与总控的交互涉及的改动点以及工程升级和兼容性等的处理。

## 术语与缩略语

POU:(Program organization unit)程序组织单元. 可以是功能、功能块或程序.（

注：此术语可以指类型或实例.)

PW： Powerlink协议缩写

DP： Profibus-DP的简称

AT: AutoThink的缩写，IEC组态软的名称.

新硬件配置：特指K-CU03主控之前开发的所有控制器型号

旧硬件配置：特指K-CU03主控之后开发的所有控制器型号，包括K-CU03主控。

## 引用文档

1、《K-CU03.SSD.01.AT 系统设计说明书》

2、《多总线结构平台组件使用说明书》

# 主要功能点设计

## 硬件配置数据上传总控

新旧主控的硬件配置上传总控的数据格式一致。即旧主控在新平台下上传给总控的硬件配置数据也按照新的结构上传。旧版本没有的数据可以按结构中的顺序填充为“NO\_NEED” ，“LOCAL”。另外，单个控制站的硬件配置数据最大支持2M，超过2M不支持上传总控。

### 硬件配置导出格式定义

新硬件配置数据导出到数据库文件仍旧采用Excel表格的形式呈现。增加“MODULE”sheet页，默认隐藏。硬件配置格式如下：

旧的硬件配置格式（参考）：

1. 旧硬件配置格式

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 站号 | 设备号 | 模块类型 | 是否冗余 | 组 | 同步冻 结 | LINK仪表配置 | 用户参数 | IO配置 |

新的硬件配置格式：

1. 新硬件配置格式

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 站号 | 模块类别 | 协议号 | 模块名称 | 模块号 | 链路位置 | 配置文件版本号 | 描述文件版本号 | 冗余标志 | 特殊数据个数 | 特殊数据信息 |

### 数据结构

新旧硬件配置通过AT框架上传给总控的数据采用一致结构。定义如下：

1. 新硬件配置结构



其中：

模块类别：指模块是属于控制器、网关还是从站；

协议号：指DP协议，PWlink协议还是ModuBus协议；

链路位置：指模块所在的链路信息，例如“1\_2\_1”，含义为第1条链路下的2#模块的第1条链路。只引申到链路层。

配置版本号：设备的版本号

描述文件版本号：设备所在描述文件的版本号（可能存在一个描述文件中存在多个设备）。

特殊数据个数：指设备的一些参数信息的个数；

### 数据结构定义

typedef struct \_tagHMIHead

{

BYTE uiTag; /\*配置名称 0x00000000\*/

BYTE uiStationNo; /\*控制器号\*/

UINT32 uiVersion; /\*版本号\*/

UINT32 uiCheckSum; /\*检验和\*/

UINT32 uiCount; /\*模块个数\*/

BYTE ucPackageFlag; /\*第一包数据是1，其余的都是0，分包发的时候

HMI第一包数据需要一些处理\*/

}HMIHead;

新硬件配置上传总控数据增加模块信息结构，定义如下：

typedef struct tagSlaveInfoToHMI

{

char cModuleType; //模块类别

char cProtocolType；//协议类别

char cSlaveName[32];//模块名称

unsigned short usAddress；//模块站号

char cLinkPosition[32];//链路位置信息

char szConfigVersion[64];//配置版本号

char szGSDVerSion[64]; //描述文件版本号

char cRedunFlag； //冗余标志

BYTE bySpecialDataNum；//特殊数据的个数

SpeConfigInfo\* pSpeConfigInfo；//特殊配置信息

}

模块配置的特殊数据结构，定义如下：

typedef struct tagSpeConfigInfo

{

DWORD dwDataLength;//数据大小

char\* pDateBuf；//数据内容

}SpeConfigInfo

其中：

模块类别：

#define HW\_CONTROL\_CARD 0x01 表示控制器

#define HW\_CMNCT\_CARD 0x02 表示通讯卡

#define HW\_MST\_CARD 0x03 表示主站

#define HW\_SLAVE\_CARD 0x04 表示从站

#define HW\_GATEWAY\_CARD 0x05 表示网关

协议类别：

#define HW\_PT\_NULL 0x00000000 表示无协议

#define HW\_PT\_DP 0x00000001 表示DP协议

#define HW\_PT\_PW 0x00000002 表示PWLink本地协议

#define HW\_PT\_MDBS 0x00000003 表示modbus协议

#defile HW\_PT\_ETHERCAT 0x00000004 表示EtherCAT协议

冗余标志：

0：非冗余

1：冗余底座

2：SR冗余

3：FR冗余

4：网关冗余切换式

5：网关冗余并联式

6：仪表单接主网关

7：仪表单接备网关

8：仪表接主备网关

数据库导出时，将“是否冗余”拆分为两列，分为为“是否冗余”和“网关接线方式”。

4、5是针对网关设备的，拆分后，是否冗余为1，接线方式为4、5；6、7、8是针对第三方厂家的仪表设备的，拆分后，是否冗余为0，接线方式为6、7、8。

数据内容信息不区分类别，总控统一按照模块参数信息处理（老硬件配置会区分IO配置、用户参数、组、同步冻结、link仪表配置等类别）。硬件配置模块要还原此部分数据时，自行在特殊数据内容中增加类别信息以进行标识。

### 与总控交互

新旧硬件配置都通过Windows消息和共享内存的方式与总控交互。首先将硬件配置数据写入共享内存，再通过发消息的形式通知总控读取共享内存。消息定义如下：

触发时机：保存工程

上传形式：共享内存

消息号：ID\_MODULECONFIG ：61

消息参数：

|  |  |  |
| --- | --- | --- |
| 消息参数 | wParam | lParam |
| 值 | ID\_MODULECONFIG | 控制站号 |

### 数据库导出

旧硬件配置平台是线性结构，而多总线平台的硬件配置是树状结构，这就要求数据库导出的测点清单中增加还原能够树状结构的信息。以两层嵌套的链路结构为例，约定在测点清单中增加三列：控制器链路位置、网关类别和网关链路位置。如下：

1. 新增点项示例

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| PN | SN | DN | CN | … | PRO\_TYPE | GW\_TYPE | LINK\_POS | … |
| 点名 | 站号 | 设备地址 | 通道号 | 控制器链路位置 | 网关类别 | 网关链路位置 |
| PN12\_AAC | 12 | 20 | 1 |  | 1 | 1 | 1 |  |
| PN12\_AAD | 12 | 20 | 2 |  | 1 | 2 | 2 |  |
| PN12\_AAE | 12 | 20 | 3 |  | 2 | 1 | 1 |  |

为了方便用户组态，控制器链路位置和网关类别均采用ID号的形式组态。手册中会逐一说明链路名称与ID、网关类别与ID的映射关系。如下：

1. 控制器链路--ID映射表

|  |  |
| --- | --- |
| 控制器链路名称 | ID号 |
| DP | 1 |
| PW | 2 |

1. 网关类别—ID映射表

|  |  |
| --- | --- |
| 网关类别 | ID号 |
| DP\_PW | 1 |
| PW\_DP | 2 |

控制器链路位置、网关类别和网关链路位置只在测点清单中体现，目的是方便用户组态。导出数据库时，由总控将导出的每个测点的位置项拆分成控制器链路位置、网关类别和网关链路位置这三项。若为单层链路结构，后两项默认值设置为0。

## 硬件配置数据导入AT

新旧硬件配置导入到AT也采用统一的格式。

硬件配置数据是通过总控生成的HD文件的形式传递给AT的，针对旧的硬件配置，导入的结构跟新结构存在差异的地方，由总控统一转换为新的HD文件结构导入到AT。旧结构中缺失的字段默认填充初始值处理。

### 旧硬件配置导入AT结构（参考）

旧硬件配置是通过HD文件的形式，将硬件配置数据导入到AT。导入数据库，总控编译时通知AT解析HD文件获取数据。HD文件结构如下：

1. 旧硬件配置HD文件结构



### 新硬件配置导入AT结构

新硬件配置导入AT的数据结构跟导出时结构保持一致。



1. 新硬件配置HD文件结构

（1）新硬件配置，总控将导出的结构直接转换为HD文件传给AT即可。

（2）旧硬件配置结构导入到新平台的AT，需要由总控转换为新的HD文件结构传给AT。其中，新结构中与老结构有差异的字段，按照默认值填充。处理如下：

1. 新结构默认值处理

|  |  |
| --- | --- |
| 关键字 | 默认值 |
| 版本 | 默认填充新版本号 |
| 校验和 | 由总控计算 |
| 模块类别 | 5：默认为从站 |
| 协议号 | 1： 默认为DP协议 |
| 链路位置 | 1：DP链路 |
| 配置版本号 | NULL |
| 描述文件版本号 | NULL |
| 数据类型 | 旧结构中数据类型跟数据内容合并 |

### 与总控交互

通过Windows消息和HD文件的形式将硬件配置数据导入到AT。

触发时机：编译总控

上传形式：HD文件

消息号：ID\_WRITEOBJECT：83

消息参数：

|  |  |  |
| --- | --- | --- |
| 消息参数 | wParam | lParam |
| 值 | ID\_MODULECONFIG | HD文件路径 |

### 数据库导入

新硬件配置测点清单按照导出的结构可以直接导入到AT中。测点清单中控制器链路位置、网关类别和网关链路位置这三项，总控解析后将这三项合并为测点位置信息，并作为测点“位置”项导入给AT。总控导入给AT之前，需要对“位置”项做唯一性检查。

旧硬件配置测点清单必须通过总控转换为新的测点清单结构再导入到AT中。以两层嵌套的链路结构为例，旧测点清单中缺少控制器链路位置、网关类别和网关链路位置这三项信息。总控在导入旧清单时，默认情况下要为每个测点增加这三个点项，并设置初始值。

## 类结构中增加“位置”项

类结构需要为硬件测点类型增加“位置”项，暂定项名为LP（LINK POSITION的缩写），代表测点所属模块的链路信息。该项为string类型。组态工程，测点位置项在添加测点的时候默认添加，且不支持用户编辑修改；

（1）配置规则

“位置”项代表模块的链路信息。其填充的规则为：链路号。若模块下仍旧挂链路，则为：链路号\_模块地址\_链路号。比如，值“1\_2\_1”，含义为第1条链路下的2#模块的第1条链路。  
（2）导数据库处理

新硬件配置的数据库中测点位置信息若为空，总控导入数据库时将其按照默认值处理。旧硬件配置数据库中没有位置项信息，总控导入数据库时，将其初始化为0。

（3）升级工程处理

升级旧版本的工程，测点的位置信息默认为空，总控在升级工程时将各测点的位置信息初始化为1。升级工程后，按照AT同步给总控的测点信息更新数据库中原测点位置信息。

## 诊断点名称同步

新硬件配置增加了链路信息，原诊断点名称“FIO\_站号\_设备地址”的结构不能通用。新控制器诊断点名称要求增加链路信息来唯一标识某个模块，比如：FIO\_10\_1\_2\_1\_8，含义为10号站下第1条链路的2#模块的第1条链路的8#地址模块。

1. 生成规则

旧硬件配置按照旧的命名规则生成诊断变量，即：FIO\_站号\_设备地址；新硬件配置按照新规则命名诊断变量，即：FIO\_站号\_链路号\_模块地址。总控自己生成的诊断点，名称也要根据控制器型号的新旧区分采用不同的命名规则。

1. 导数据库处理

旧硬件配置数据库导入，还是按照旧的命名规则生成变量。

新硬件配置的数据库导入，总控需要根据模块的链路位置和模块地址拼接为新的诊断变量名称。

特别说明：旧硬件配置中DN可以唯一标识模块的地址，但在新硬件配置，DN只代表某条链路下某个模块的地址，要唯一标识单个控制站内一个模块的位置信息，必须结合“链路位置+模块地址”。

1. 升级工程处理

旧控制器型号的工程升级，按照旧的命名规则生成诊断变量。若K-CU01切换控制器型号为K-CU03的升级，默认会升级到K-CU03内置链路上，总控需要按照K-CUO3的链路信息和模块地址重新生成诊断点名称。

## K-CU01切换K-CU03

初步规划K-CU01到K-CU03的控制器切换在总控由用户手动触发。大致流程如下：

1. K-CU01切换K-CU03流程



（1）默认启动AT，新旧硬件配置全部加载；

（2）切换控制器型号从总控由用户手动触发；

（3）切换完控制器型号，用户不能打开控制站，必须先触发总控编译此控制站；

（4）更新测点位置项，也可以由AT自行修改，再将变量全部同步给总控。不需要总控导入处理。