变量强制查看原始值

**1.1 目的**

本文档依据《MACS V6.5.5软件开发项目\_MACS V6.5.5B1系统需求规格书》进行设计，描述了强制功能改进实现的关键技术、子系统划分、接口定义。

编写此文档的目的是明确AT软件各功能模块的接口与设计思路，作为详细设计及编码工作的输入，并指导后续的测试和维护工作。

**1.2 预期读者**

本文档的预期读者是：测试人员、开发人员、手册编写人员。

**1.3 术语与缩略语**

AT：AutoThink，HOLLiAS MACS V6 版本中的算法组态软件AutoThink的简称

总控：工程总控

ST：文本语言

**1.4 引用文档**

[1] 《MACS V6.5.5软件开发项目\_MACS V6.5.5B1系统需求规格书》

**2 假设和约束**

**2.1假设**

**2.2 约束**

**3 系统概述**

**3.1 系统结构概述**

HOLLiAS MACS V6.5.4系统主要由以下子系统构成：

1. 控制站：包括控制器和I/O卡件，主要实现采集过程数据，进行数据转换和处理，实施连续、批量或顺序控制的运算和输出控制作用。
2. 历史站：包括I/O 服务，实时数据库服务，历史数据库服务，报警处理服务，主要进行历史数据的收集存储、报警、日志记录的产生。
3. 工程师站：包括数据库组态软件、图形组态软件、控制算法组态软件、报表组态软件，主要完成工程、监视画面、控制算法、报表的组态以及一些系统设置。
4. 操作员站：包括操作员在线监视软件，报表打印服务，提供监视、操作现场设备的人机界面。

系统中各个节点通过网络连接，系统划分为：控制网和系统网。



**4 模块设计**

**4.1 软件结构描述**

AT目前的架构如下图所示：



**本次涉及修改的模块如下：**

**1：公共模块**

**2：点面板模块**

**3：IEC模块（CFC\LD\SFC\ST）**

**4：在线模块**

**5：trg配置文件**

下文以模块为单位阐述设计方案。

**4.2 公共模块**

**4.2.1 APPDATA.dll**

需要对类CbaseDB进行扩展：

class APPDATA\_DLL\_DECL CBaseDB

{

…………………………………

public:

CString m\_strOriginValue;

public:

CString GetOriginValue();

private:

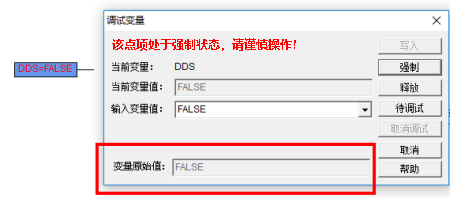
SetOriginValue(CString OriginValue);

…………………………………

}

**4.2.2 COMMONINFO.DLL**

公共模块**COMMONINFO.DLL**针对写值对话框修改，写值对话框达成的效果如下：



增加一个EDIT控件定时刷新在线值。

在**COMMONINFO**的调试变量对话框上增加一个EDIT控件，该控件的ID为：IDC\_PUBLIC\_EDIT3，值为2378。增加一个静态文本框控件，值为“变量原始值”，由于静态文本框只是显示功能，不关联变量，所以ID采用默认ID，值为默认值-1。增加一个静态文本框，ID为IDC\_STATIC\_TIP，值为2379。

对原始值EDIT控件关联成员变量：

class PUBLIC\_DLL\_DECL CDlgWriteVar : public Cdialog

{

…………………………

…………………………

Public:

CString m\_bOriginValue;

…………………………

}

数据交互结构添加如下关联关系：

DDX\_Text(pDX, IDC\_PUBLIC\_EDIT3, m\_bOriginValue);

在类CdlgWriteVar的实现函数增加定时器回调函数。

回调函数声明如下：

VOID CALLBACK TimerProc ( HWND hwnd, UINT message, UINT iTimerID, DWORD dwTime)

其实现流程图如下：



在BOOL CDlgWriteVar::OnInitDialog()里面处理控件显示，以及定时器回调函数设置，其实现流程图如下：



对于释放按钮的响应函数void CDlgWriteVar::OnPublicUnmanu()的处理流程图如下：



**4.3 点面板模块**

AT点面板模块的设计与公共模块COMMONINFO.DLL存在一点区别。PinDetailHeader.h需要对结构体tagItemInfo增加一个成员变量

typedef struct tagItemInfo

{

long lDomainNo; //域号

long lFCSNo; //站号

CString strPin; //点名

CString strName; //点项名

CString strOnlineValue; //点项在线值

/////////////////////////////////////////////////////////

**CString strOnlineOriginValue;//点项在线原始值**

/////////////////////////////////////////////////////////

bool bForced;

bool bNomal; //点项状态是否正常

BOOL bDebugFlg; //点项是否可调式标志

BOOL bString; //点项是否字符串类型

tagItemInfo()

{

lDomainNo = 0;

lFCSNo = 0;

strPin = "";

strName = "";

strOnlineValue = "";

bForced = false;

bNomal = false;

bDebugFlg = FALSE;

bString = FALSE;

}

}ItemInfo;

在函数BOOL CTagInfoSet::GetPinOnlineValueOFTag(CDynCtrlManage\* pDynCtrlManage)里面的处理流程如下：



在类CdlgWriteVar里面增加成员变量ItemInfo\* m\_ItemInfo，用来获取在线值。

对现有接口

void CdlgWriteVar::SetInfo(CItem\*pItem,CString strValue,CDynCtrlManage \*pDynCtrlManage,CPoint pt);

修改为：

void\_CdlgWriteVar::SetInfo(CItem\*pItem,ItemInfo\*pItemInfo,CDynCtrlManage \*pDynCtrlManage,CPoint pt);

所以凡是涉及该成员函数接口的调用，均要传入ItemInfo\*指针。通过该指针，来获取该点项的在线原始值。

增加一个EDIT控件定时刷新在线值。

在**PinDetailEx**的调试变量对话框上增加一个EDIT控件，该控件的ID为：IDC\_PUBLIC\_EDIT3，值为253。增加一个静态文本框控件，值为“变量原始值”，由于静态文本框只是显示功能，不关联变量，所以ID采用默认ID，值为默认值-1。增加一个静态文本框，ID为IDC\_STATIC\_TIP，值为254。

对原始值EDIT控件关联成员变量：

class PUBLIC\_DLL\_DECL CDlgWriteVar : public Cdialog

{

…………………………

…………………………

Public:

CString m\_bOriginValue;

…………………………

}

数据交互结构添加如下关联关系：

DDX\_Text(pDX, IDC\_PUBLIC\_EDIT3, m\_bOriginValue);

在类CdlgWriteVar的实现函数增加定时器回调函数。

回调函数声明如下：

VOID CALLBACK TimerProc ( HWND hwnd, UINT message, UINT iTimerID, DWORD dwTime)

其实现流程图如下：



在BOOL CDlgWriteVar::OnInitDialog()里面处理控件显示，以及定时器回调函数设置，其实现流程图如下：



对于释放按钮的响应函数void CDlgWriteVar::OnPublicUnmanu()的处理流程图如下：



**4.4 IEC模块**

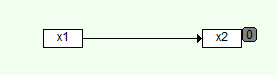
IEC模块主要涉及三个语言的语法树修改，以及界面显示，三个语言分别为CFC、LD、ST，其中涉及语法树修改的POU类型为PRG型POU和FB型POU。

图形语言以CFC为例进行设计说明

设计语法树修改的部分主要是针对赋值语句，其它语句不涉及。

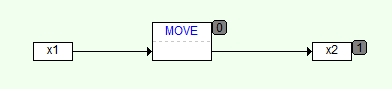
而CFC的赋值语句分为三种形式：

①直接赋值：



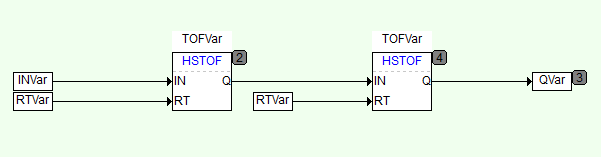
其中x1与x2为简单变量或者复杂变量

②MOVE块赋值



其中x1与x2为简单变量或者复杂变量

③功能块引脚赋值



其中INVar与RTVar、QVar为简单变量或者复杂变量。

这三种形式的赋值语句均要加外部库函数调用。

目前赋值语句正常的伪代码如下：

PRG POU的正常逻辑如下：

if（!val(adr(x2)+ x2\_Force\_offset)）

{

x2=x1;

}

PRG POU的修改后的逻辑如下

if（!val(adr(x2)+ x2\_Force\_offset)）

{

x2=x1;

}

else

{

int RefID = x2.RefID;

int offset = x2.offset;

int typeindex= x2.typeindex

void\*pvalue=adr(x1);

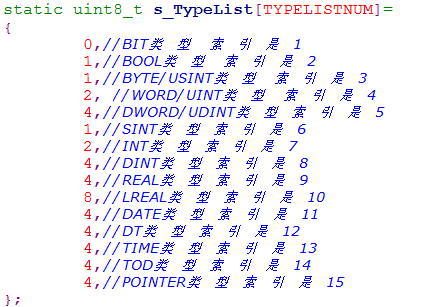
int size = x1.size

sysAssignPrg(RefID,offset, typeindex,pvalue, size);//外部库函数

}

其中变量类型索引typeindex由RTS提供：

类型数据定义如下：



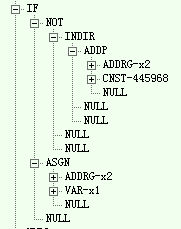
在CPOU里面增加一个成员函数：

Int CPOU::GetBaseDBIndex(CBaseDB\*pBaseDB);

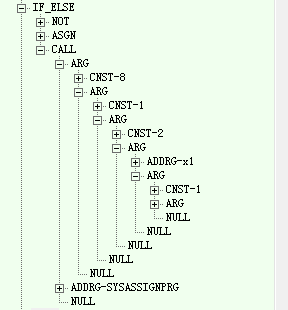
该函数的流程图如下：



目前的语法树构建如下：



修改后的语法树构建如下



FB POU的正常逻辑如下：

if（!val(this+x2\_offset+x2\_Force\_offset)）

{

x2=x1;

}

FB POU的修改后的逻辑如下

if（!val(this+x2\_offset+x2\_Force\_offset)）

{

x2=x1;

}

else

{

int offset = x2.offset;

int typeindex = x2.typeindex;

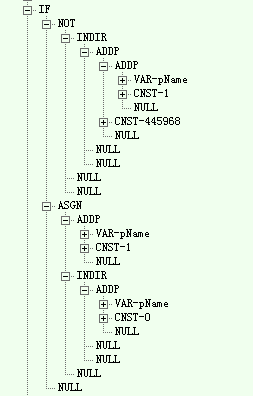
void\*pvalue=this+ x1.offset;

int size= x1.size;

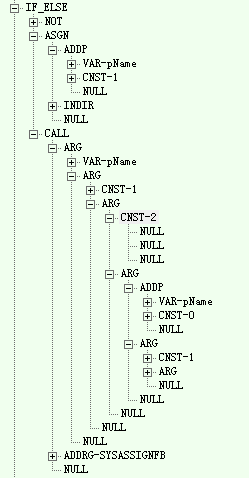
sysAssignFB(this,offset, typeidex,pvalue, size);//外部库函数

}

目前的语法树构建如下：



修改后的语法树如下：



对于ST语言的的赋值语法树修改，同理

**4.5 在线模块**

在线模块需要对读变量服务进行修改，修改的接口：

int COnlineGlobal::SetDBOnlineValue(bool bDecimal)里面处理如下：



**4.6 trg配置文件**

**①**在AT2.trg文件里面增加PRG赋值外部库函数。

AT17=sysAssignPrg

RTS17=sysAssignPrg

增加FB赋值外部库函数

AT18=sysAssignFB

RTS18=sysAssignFB

注：目前对应平台的trg文件均要修改。

其中sysAssignPrg声明如下：

**void sysAssignPrg（int desRefID,int desoffset, int desindex,int void\*psrcvalue,int srcsize）；**

**void sysAssignFB （void\*pdesthis,int desoffset, int desindex, void\*psrcvalue, int srcsize）；**

**②**在AT2.trg文件里面修改版本号。

**[TRGVersion]**

**TargetVarsion = 0xxxxxxxx//规则见说明**