**SAMA图概要设计文档**

**1 引言**

**1.1 目的**

本文档依据《MACS V65X软件技术改进与开发\_SAMA图需求规格说明书》进行设计，描述了SAMA图实现的关键技术、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 V65X软件技术改进与开发\_SAMA图需求规格说明书》

[2] 《MACS V65X软件技术改进与开发\_SAMA图导出可行性分析报告》

**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。



图 1 系统结构图

**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：框架模块**

**2：CFC模块**

**本次新增模块如下：**

**1：SAMADLL**

**2：VisioAPI**

**3：基础工程文件**

由于新增模块SAMADLL与VisioAPI,所以AT的架构调整如下图所示：



下文以模块为单位阐述设计方案。

**4.2 框架模块**

**4.2.1 单个控制站级导出**

在框架的【工程】菜单里面增加“导出SAMA图”菜单项，如下图所示：



对应的菜单资源为：

IDR\_AUTOTHINK\_CFC

IDR\_AUTOTHINK\_LD

IDR\_AUTOTHINK\_SFC

IDR\_AUTOTHINK\_ST

增加的导出SAMA图的菜单项ID为： IDM\_EXPORT\_SAMA

在框架模块的Resource.h里面增加该ID的定义：

#define IDM\_EXPORT\_SAMA 974

在MainFrm.h定义该消息响应函数：

class CMainFrame : public CMDIFrameWnd

{

……………………………………………

……………………………………………

……………………………………………

protected:

//{{AFX\_MSG(CMainFrame)

……………………………………………

***afx\_msg void OnExportSAMA();***

***afx\_msg void OnUpdateExportSAMA(CCmdUI\* pCmdUI);***

……………………………………………

//}}AFX\_MSG

……………………………………………

};

在MainFrm.cpp定义该消息响应函数：

ON\_COMMAND(***IDM\_EXPORT\_SAMA,OnExportSAMA***)

ON\_UPDATE\_COMMAND\_UI(***IDM\_EXPORT\_SAMA, OnUpdateExportSAMA***)//该函数控制【导出SAMA图】菜单项是否使能。

函数void OnExportSAMA()的实现流程图如下所示：



点击导出SAMA图菜单项，弹出导出SAMA图对话框，如下：



**注：导出SAMA图菜单的使能前提条件是该站的工程编译通过。**

图纸大小默认是A0

转换的两种方式：

①如果不点击选择转换的POU，直接点击确定，则按文件夹层级分级导出，导出路径为工程底下的工程名文件夹，并导出到对应的站号下。

②如果点击了【选择转换的POU】按钮，则弹出【选择POU】对话框，用户可以选择想要转换的POU，并选择相应的导出路径，通过这种方式的操作，所有选择的POU导出为一个VSD文件，每个POU占一个sheet页。

增加导出SAMA图的对话框类如下：

class CVisioActiveXDlg : public CDialog

{

public:

CVisioActiveXDlg(CWnd\* pParent = NULL); // standard constructor

//{{AFX\_DATA(CVISIOTOATXMLDlg)

enum { IDD = IDD\_VISIO\_ACTIVEX };

CDrawingControl m\_visio\_drawing;//Visio OLE对象

CStringArray m\_arrSelectPOUs; //用户选择的POU集合

//}}AFX\_DATA

//{{AFX\_VIRTUAL(CVISIOTOATXMLDlg)

protected:

virtual void DoDataExchange(CDataExchange\* pDX); // DDX/DDV support

//}}AFX\_VIRTUAL

protected:

//{{AFX\_MSG(CVISIOTOATXMLDlg)

virtual BOOL OnInitDialog();

virtual void OnOK();

virtual void OnCancel();

afx\_msg void OnSelectPaper();//选择纸张函数

afx\_msg void OnSelectPou()//打开选择转换的POU对话框

//}}AFX\_MSG

DECLARE\_MESSAGE\_MAP()

public:

void ExportFOLDERPouToSAMA

(CPOUTreeNode\*ptPOUTreeNode,

CVDocument VDocument,

CPouNodeList &tCFCPOUNodeList);

void ExportPouToSAMA

(CPouNodeList &tCFCPOUNodeList,

CVDocument VDocument,

CString SavePath);//转换CFC POU到SAMA

void Generateddirectory(CVDocument PVDocument);//构建SAMA图目录页

void FBdescription (CVDocument PVDocument);//构建SAMA图功能块说明页

public:

CString m\_strPaper; //纸张类型

CString m\_Samapath;//控制站工程路径或者用户选择的导出路径

CString m\_Modulepath;//模板路径

BOOL m\_bflag//用户自定义转换标志

Public:

CList<CSAMAPOU\*, CSAMAPOU\*>m\_SAMAPOUList; //SAMAPOU列表

CMap<CString, LPCTSTR,CSAMAPOU\*, CSAMAPOU\*>m\_SAMAPOUMap; //SAMAPOU Map

void SamaPouCrossReference(CVDocument PVDocument);//SAMA图POU交叉引用

void [Hyperlink](http://www.baidu.com/link?url=dJh756oENd77n7sjhaH5vDav97NumESZZ6eUhPiBQIeM2qmxImS7UCHAYsOr6tNJXSPznVUIb1bmLXtHRuE5TagFBeuk_2xaPm3BFIy2--m)Add(CVDocument PVDocument);//超链接添加

void ExcelPouCrossReference(CVDocument PVDocument); //Excel交叉引用表

void ExcelPouparameter(CVDocument PVDocument); //Excel静态参数表

void RreshCrossReferenceAndParameter(CVDocument PVDocument);//POU拆分刷新

void SpiltDirectoryAndSaveVSDFile(CVDocument PVDocument)//一个总的VSD文件按目录层级拆分多个VSD文件，针对本站所有的CFC PRG POU。

……………………………………………………………………..//辅助成员函数

……………………………………………………………………..//辅助成员函数

……………………………………………………………………..//辅助成员函数

};

IDD\_VISIO\_ACTIVEX为该对话框的资源ID号，定义如下：

#define IDD\_VISIO\_ACTIVEX 1061

该类定义与其成员函数实现分别定义在DlgVisioActiveX.h以及DlgVisioActiveX.cpp中

函数BOOL CVisioActiveXDlg ::OnInitDialog()的实现流程图如下：



函数void OnSelectPou()的实现流程图如下：



函数void CVisioActiveXDlg ::OnOK()的实现流程图如下：



用户选择的POU导出走流程1的处理流程图如下：



默认本站所有CFC PRG POU导出走流程2的处理流程图如下：



函数void Generateddirectory(CVDocument PVDocument) 的实现流程图如下：



函数void FBdescription (CVDocument PVDocument)的实现流程图如下：



函数void ExportPouToSAMA

(CPouNodeList &tCFCPOUNodeList,

CVDocument VDocument,

CString SavePath) 的实现流程图如下：



交叉引用信息只针对输入输出元件对应的图元，对于输出元件需要找对应的输入元件的引用POU，对于输入元件需要找对应的输出元件的引用POU。

函数void SamaPouCrossReference(CVDocument PVDocument)处理流程图如下：



对于超链接则针对三种元件分别是输入元件、输出元件、块元件。其中的引用信息则不分是否为输入引用信息或者输出引用信息。均是全部填充。

函数void [Hyperlink](http://www.baidu.com/link?url=dJh756oENd77n7sjhaH5vDav97NumESZZ6eUhPiBQIeM2qmxImS7UCHAYsOr6tNJXSPznVUIb1bmLXtHRuE5TagFBeuk_2xaPm3BFIy2--m)Add(CVDocument PVDocument) 处理流程图如下：



SAMA图EXCEL导出的交叉引用表，本质上就是超链接的信息导出。因为SAMA图上图元的交叉引用信息颗粒度到POU级别，并且只针对输入图元和输出图元，但是超链接针对的是输入输出以及块元件的引用信息。静态的索引表要求每个变量在POU里面的引用情况。所以针对该需求，需要导出的是SAMA图元的超链接信息。

函数void ExcelPouCrossReference(CVDocument PVDocument)处理流程图如下：



函数void ExcelPouparameter(CVDocument PVDocument)处理流程图如下：



函数void RreshCrossReferenceAndParameter(CVDocument PVDocument)处理流程图如下：



函数void SpiltDirectoryAndSaveVSDFile(CVDocument PVDocument) 处理流程图如下：



其中文档拆分只针对本站所有的CFC PRG POU导出为SAMA图。保存到相应目录是根据文件夹层级进行预先创建的目录。

CDrawingControl类为OLE绘图窗口类，定义如下：

class CDrawingControl : public CWnd

{

protected:

DECLARE\_DYNCREATE(CDrawingControl)

public:

CLSID const& GetClsid()

{

static CLSID const clsid

= { 0xe4615fa3, 0x23b0, 0x4976, { 0xbd, 0x3e, 0xd6, 0x11, 0xdd, 0xbe, 0x33, 0xe } };

return clsid;

}

virtual BOOL Create(LPCTSTR lpszClassName,

LPCTSTR lpszWindowName, DWORD dwStyle,

const RECT& rect,

CWnd\* pParentWnd, UINT nID,

CCreateContext\* pContext = NULL)

{ return CreateControl(GetClsid(), lpszWindowName, dwStyle, rect, pParentWnd, nID); }

BOOL Create(LPCTSTR lpszWindowName, DWORD dwStyle,

const RECT& rect, CWnd\* pParentWnd, UINT nID,

CFile\* pPersist = NULL, BOOL bStorage = FALSE,

BSTR bstrLicKey = NULL)

{ return CreateControl(GetClsid(), lpszWindowName, dwStyle, rect, pParentWnd, nID,

pPersist, bStorage, bstrLicKey); }

// Operations

public:

CVDocument GetDocument();

CVWindow GetWindow();

CString GetSrc();

void SetSrc(LPCTSTR lpszNewValue);

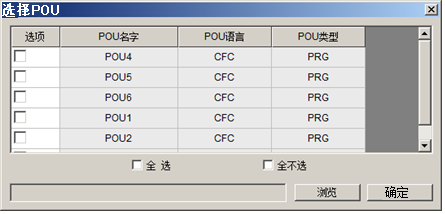
CString GetHostID();

void SetHostID(LPCTSTR lpszNewValue);

};

其中里面的成员函数均为VISIO COM接口调用。

其中点击【选择转换的POU】按钮，会弹出如下对话框：



针对该对话框增加如下类定义：

class CDlgExportPouToSama : public CDialog

{

…………………………………………………

…………………………………………………

public:

enum { IDD = IDD\_POU\_EXPORTTOSAMA };

CString m\_strPath; //用户选择的导出路径

CStringArray m\_arrSelectSets; //用户选择的POU集合

…………………………………………………

protected:

CGridCtrl m\_expotgrid;//grid控件对象

virtual BOOL OnInitDialog();

afx\_msg void OnBrowse();//路径浏览响应函数

afx\_msg void OnOK ();//确定消息响应函数

afx\_msg void OnResetAll();//全不选消息响应函数

afx\_msg void OnCheckAll();//全选消息响应函数

afx\_msg void OnGridEndEdit(NMHDR \*pNotifyStruct, LRESULT\* pResult); //单元格结束编辑。

}

该类定义以及类成员函数实现分别定义在DlgExportPouToSama.h以及DlgExportPouToSama.cpp中，其中IDD\_POU\_EXPORTTOSAMA为该对话框的资源ID，定义如下：

#define IDD\_POU\_EXPORTTOSAMA 1064。

函数CDlgExportPouToSama :: OnInitDialog()的处理流程如下：



函数CDlgExportPouToSama ::OnCheckAll()的处理流程如下：



CDlgExportPouToSama :: OnResetAll()的处理流程如下：

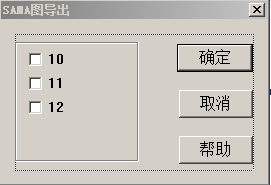


CDlgExportPouToSama :: OnOK()的处理流程如下：



**4.2.2 总控导出SAMA图**

在总控上，也可以针对本域进行所选的站进行导出。以工程和控制站为单位导出时，入口在工程总控处，点击菜单工程/导出SAMA图，弹出如下对话框：



用户选择对应的控制站后，总控记录对应的站号信息，然后在总控编译阶段，由总控依次启动每个站，如果该站已经选择为导出SAMA图，则总控给AT发送一个导出SAMA图的交互ID，AT在对应的消息ID里面导出该站的SAMA图。

在HMIInterface.h里面定义与总控交互的导出SAMA图消息，定义如下：

#define ID\_EXPORT\_SAMA 234

void OnMessageExportSama();

该函数的处理流程图如下：



其中导出SAMA图对话框的ONOK函数处理流程见4.2.1节。

其中【导出SAMA图】对话框以及【选择POU】对话框资源均定义在AutoThink.exe中。

**4.3 CFC模块**

由于CFC模块只是针对CFC语言的一些功能，为了不影响现有CFC模块的稳定性，为了实现导出SAMA图功能与现有的CFC模块解耦，只需要对CFC语言的一些类进行导出设置即可。所有的导出功能均在SAMADLL模块里面实现。

CFC需要设置导出的类如下：

/////////////////////////////////////////////////////////////////////////////////导出类宏定义

#undef CFC\_DLL\_DECL

#ifdef \_CFC\_DLL\_DECL

#define CFC\_DLL\_DECL \_declspec(dllexport)

#else

#define CFC\_DLL\_DECL \_declspec(dllimport)

#endif

/////////////////////////////////////////////////////////////////////////////////导出的元件类

class CFC\_DLL\_DECL CCFCElement : public CElement

class CFC\_DLL\_DECL CCFCBox : public CCFCElement

class CFC\_DLL\_DECL CCFCInput : public CCFCElement

class CFC\_DLL\_DECL CCFCOutput : public CCFCElement

class CFC\_DLL\_DECL CCFCLabel : public CCFCElement

class CFC\_DLL\_DECL CCFCReturn : public CCFCElement

class CFC\_DLL\_DECL CCFCJump : public CCFCElement

class CFC\_DLL\_DECL CCFCComment : public CCFCElement

/////////////////////////////////////////////////////////////////////////////////导出的引脚类

class CFC\_DLL\_DECL CCFCPin : public CObject

class CFC\_DLL\_DECL CCFCOutputPin : public CCFCPin

class CFC\_DLL\_DECL CCFCInputPin : public CCFCPin

//导出连线类

class CFC\_DLL\_DECL CCFCLine : public CObject

**4.4 SAMADLL模块**

SAMADLL为新增模块，所有的SAMA图翻译均在该模块里面实现，该模块为扩展MFC动态链接库，由于SAMADLL继承自CFC，所以，所以在SAMADLL里面定义如下POU类SAMAPOU.

#undef SAMA\_DLL\_DECL

#ifdef \_SAMA\_DLL\_DECL

#define SAMA\_DLL\_DECL \_declspec(dllexport)

#else

#define SAMA\_DLL\_DECL \_declspec(dllimport)

#endif

其中预编译宏\_SAMA\_DLL\_DECL在SAMADLL的工程属性里面定义。

定义VISIO图元的中心点坐标结构体如下：

typedef struct CStructXYmm

{

CString inchxmm;

CString inchymm;

}CStructfloatXYmm;

Typedef Clist<int,int> Idlist;

CSAMAPOU的类定义如下

class SAMA\_DLL\_DECL CSAMAPOU : public CCFCPOU

{

DECLARE\_SERIAL(CSAMAPOU)

public:

CSAMAPOU();

virtual ~CSAMAPOU();

void CopyPouInfo(CCFCPOU\*pCFCPOU);

void TransSAMA(CVDocument PVDocument, int CurrentPageNo,int TotalPageNo);

void CFCElementsTransSAMA(CVDocument PVDocument);//SAMA图元处理

void CFClinesStore();//存储元件之间的线连接

void SAMAElementsLink(CVDocument PVDocument);//SAMA图元之间的线连接

void DrawGraphConentSet(CVDocument PVDocument,

intCurrentPageNo,int TotalPageNo)//施工图处理函数

void InputTransSAMA(CCFCInput\*pCFCInput,CVDocument PVDocument);

void OutputTransSAMA(CCFCOutput\*pCFCOutput,CVDocument PVDocument);

void BoxTransSAMA(CCFCBox\*pBox,CVDocument PVDocument);

void LabelTransSAMA(CCFCLabel\*pLabel,CVDocument PVDocument);

void ReturnTransSAMA(CCFCReturn\*pReturn,CVDocument PVDocument);

void CommentTransSAMA(CCFCComment\*pComment,CVDocument PVDocument);

void JumpTransSAMA(CCFCJump\*pJump,CVDocument PVDocument);

void FBTransSAMA(CCFCBox\*pFBBox,CVDocument PVDocument);

void FUNTransSAMA(CCFCBox\*pFUNBox,CVDocument PVDocument);

void OpratorTransSAMA(CCFCBox\*pOpratorBox,CVDocument PVDocument);

void PrgTransSAMA(CCFCBox \*pPrgBox,CVDocument PVDocument);

void SplitSAMA(CVDocument PVDocument);//SAMA图拆分

void PageSkipping(CVDocument PVDocument);//页内跳转

void Eleparameteradd(CVDocument PVDocument);//图元参数项添加

void LogicRotate(CVDocument PVDocument);//逻辑旋转

void SamaPouCrossReference(CVDocument PVDocument);//SAMA图POU交叉引用

void [Hyperlink](http://www.baidu.com/link?url=dJh756oENd77n7sjhaH5vDav97NumESZZ6eUhPiBQIeM2qmxImS7UCHAYsOr6tNJXSPznVUIb1bmLXtHRuE5TagFBeuk_2xaPm3BFIy2--m)Add(CVDocument PVDocument);//超链接添加

void ExportExcelHyperlink(CVDocument PVDocument); //Excel交叉引用表

void ExportExcelParameter(CVDocument PVDocument); //Excel静态参数表

void RreshCrossReferenceAndParameter(CVDocument PVDocument,

CList<CSAMAPOU\*, CSAMAPOU\*>&SAMAPOUList);//POU拆分刷新

…………………………………………………………//辅助功能函数

…………………………………………………………//辅助功能函数

…………………………………………………………//辅助功能函数

public:

CMap<int,int,CVShape,CVShape> m\_SAMAEleMap;//ID与图元的映射关系

CString m\_strPaperSize;//存放纸张类型

CArrayString<CString, CString>m\_PouNameSplit;//子POU名字拆分数组

CMap< CString, LPCTSTR, Idlist, Idlist&> m\_SonPouIDList//子POU元件链表

}

函数void CopyPouInfo(CCFCPOU\*pCFCPOU)的功能流程图如下：



函数void TransSAMA(CVDocument PVDocument,int Index,int Total);

的功能流程图如下：



函数void CFCElementsTransSAMA(CVDocument PVDocument,int Index,int Total)的功能流程图如下：



函数void InputTransSAMA(CCFCInput\*pCFCInput,CVDocument PVDocument)处理流程图如下：



函数void OutputTransSAMA(CCFCInput\*pCFCInput,CVDocument PVDocument)处理流程图如下：



函数 void CommentTransSAMA(CCFCComment\*pComment,CVDocument PVDocument) 处理流程图如下：



函数void LabelTransSAMA(CCFCLabel\*pLabel,CVDocument PVDocument) 处理流程图同理

函数void JumpTransSAMA(CCFCJump\*pJump,CVDocument PVDocument) 处理流程图



函数void ReturnTransSAMA(CCFCLabel\*pReturn,CVDocument PVDocument) 处理流程图

同理

函数void BoxTransSAMA(CCFCInput\*pCFCInput,CVDocument PVDocument)处理流程图如下：



对于功能块，有些是业界共识的SAMA通用图元的，比如PID，以及手操器等，所以针对PID、MAN手操器采用业界共识的SAMA图元。其中块元件比较特殊，目前块元件分为FB(功能块)、FUN（函数块）、Operator(操作符块)、PRG（PRG块）。

对于FB来说，其中凡是存在SAMA图元的，采用模具文件里面的SAMA图元，凡是不存在SAMA图元的，则采用默认的块元件进行表示。

对于FUN以及Operator同理，其中PRG块采用默认的块元件进行表示。

其中对于FB类型的块元件，由于其关联一个FB型的变量，该变量的一些参数项需要显示，方便用户查看，对于具体的FB类型，显示哪些参数项，由提供的配置文件决定，在添加FB型类型的参数项时，需要读取系统提供的参数项配置文件，读取相应的参数项以及初始值，并将该参数项以及对应的初始值添加到形状数据里面，参数项配置文件需要添加到基础工程里面，作为基础工程平台文件存在，具体的存放目录位置以及该配置文件的参数内容的格式见第4.6章节。

函数void FBTransSAMA(CCFCBox\*pFBBox,CVDocument PVDocument)处理流程图如下：



函数 void FUNTransSAMA(CCFCBox\*pFUNBox,CVDocument PVDocument) 处理流程同理

函数void OpratorTransSAMA(CCFCBox\*pOpratorBox,CVDocument PVDocument)处理流程同理

由于PRG块没有对应的行业SAMA图标准，所以在SAMA图元模具文件里面是不存在PRG块对应的SAMA图元的，所以采用一般的普通块表示PRG块的SAMA图元。

函数void PrgTransSAMA(CCFCBox \*pPrgBox,CVDocument PVDocument)处理流程如下：



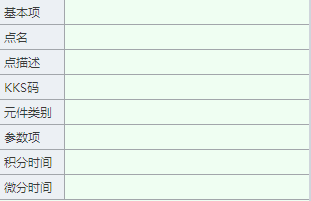
函数void PageSkipping(CVDocument PVDocument)处理流程图如下：



函数void Eleparameteradd(CVDocument PVDocument)处理流程图如下：

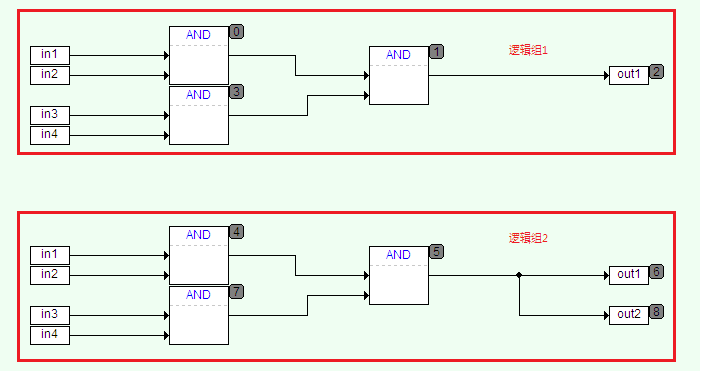


其中功能块类型的参数项信息，由配置文件提供，除了参数项信息，针对输入输出以及块元件对应的图元还有一些基本的信息，如下图所示：



其中基本项可以在制作模具文件的时候静态添加，也可以代码动态添加，为了支持第三方模具文件，所以采用代码动态添加。

由于开关量与模拟量的逻辑表示方法不同，开关量通常采用横向的逻辑表达形式，模拟量通常采用纵向的逻辑表达形式，所以对于CFC的逻辑需要进行相应的旋转转换，旋转转换的单位为逻辑组。逻辑组的概念就是由线连接起来的一组元件集合，如下图所示：



函数void LogicRotate(CVDocument PVDocument)处理流程图如下：



其中如果是功能块则需要在配置文件里面进行查表，如果该功能块类型在配置文件里面存在，则为纵向逻辑表达，否则为横向逻辑表达。该配置文件的存放路径以及格式见后面的第4.6章节所述。

函数void SplitSAMA(CVDocument PVDocument)的处理流程图如下：



函数void SamaPouCrossReference(CVDocument PVDocument) 的处理流程图如下：



函数void [Hyperlink](http://www.baidu.com/link?url=dJh756oENd77n7sjhaH5vDav97NumESZZ6eUhPiBQIeM2qmxImS7UCHAYsOr6tNJXSPznVUIb1bmLXtHRuE5TagFBeuk_2xaPm3BFIy2--m)Add(CVDocument PVDocument)的处理流程图如下：



函数void ExportExcelHyperlink(CVDocument PVDocument)的处理流程图如下：



函数void ExportExcelParameter(CVDocument PVDocument)的处理流程图如下：



函数void RreshCrossReferenceAndParameter(CVDocument PVDocument,

CList<CSAMAPOU\*, CSAMAPOU\*>&SAMAPOUList) 的处理流程图如下：



辅助功能函数根据实际具体实现，进行添加所需要的功能函数。

**4.5 VisioAPI模块**

VisioAPI模块是对Visio软件提供的COM接口进行了一个扩展MFC动态链接库封装。以C++类的组织方式进行了常用接口的封装，并对这些类进行了导出。

其中导出宏如下：

#undef VISIOAPI\_DLL\_DECL

#ifdef \_VISIOAPI\_DLL\_DECL

#define VISIOAPI\_DLL\_DECL \_declspec(dllexport)

#else

#define VISIOAPI\_DLL\_DECL \_declspec(dllimport)

#endif

导出的封装类如下：

class VISIOAPI\_DLL\_DECL CVApplication : public ColeDispatchDriver

class VISIOAPI\_DLL\_DECL CVShapes : public COleDispatchDriver

class VISIOAPI\_DLL\_DECL CVShape : public COleDispatchDriver

class VISIOAPI\_DLL\_DECL CVWindow : public COleDispatchDriver

class VISIOAPI\_DLL\_DECL CVDocuments : public COleDispatchDriver

class VISIOAPI\_DLL\_DECL CVDocument : public COleDispatchDriver

class VISIOAPI\_DLL\_DECL CVPages : public COleDispatchDriver

class VISIOAPI\_DLL\_DECL CVPage : public COleDispatchDriver

class VISIOAPI\_DLL\_DECL CVMasters : public ColeDispatchDriver

class VISIOAPI\_DLL\_DECL CVMaster : public ColeDispatchDriver

class VISIOAPI\_DLL\_DECL CVCell : public ColeDispatchDriver

class VISIOAPI\_DLL\_DECL CVSelection : public ColeDispatchDriver

class VISIOAPI\_DLL\_DECL CVRow : public ColeDispatchDriver

class VISIOAPI\_DLL\_DECL CVSection : public ColeDispatchDriver

…………………………………………………………………………//待定封装类

…………………………………………………………………………//待定封装类

…………………………………………………………………………//待定封装类

其中这些类均是对Visio COM 接口的C++类封装。

**4.6 基础工程文件**

基础工程文件分为两部分：

①SAMA模具文件

②SAMA配置文件

**4.6.1 SAMA模具文件**

SAMA模具文件的制作，根据需求目前需要做五个模具文件

①普通操作符块与函数块以及功能块模具文件（AND，OR等）

其中该模具文件命名为BaseBlock.VSS

②特殊功能块图元模具文件（PID,MAN等）

其中该模具文件命名为SpecialBlock.VSS

③输入输出元件图元模具文件

其中该模具文件命名为INOUT.VSS

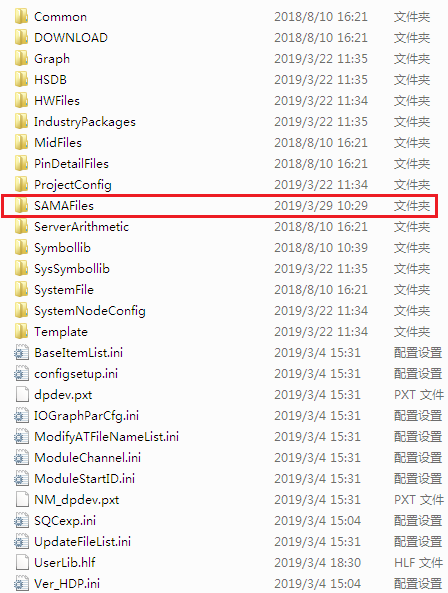
④积木模具文件（构造普通操作符块和函数块以及普通功能块覆盖不到的一些块，例如6个输入管脚的AND,OR等这种块元件）

其中该模具文件命名为BuildingBlock.VSS

其中积木模具文件包含的图元都是一些很简单的图元，用这些很简单的图元后面构造上述三个模具文件不存在的图元。

其中存储路径如下：

基础工程目录如下图所示：



即：..\HOLLiAS\_MACS\ENG\BaseProj\BASEPROJ\_Thermal Power\SAMAFiles

其中该SAMAFiles文件夹下，包含三个子文件夹，如下图所示：

C:\Users\HUWEIG~1.HOL\AppData\Local\Temp\企业微信截图_15538273095835.png

其中：

INI文件夹里面存储SAMA图的一些配置文件

VSS文件夹里面存储SAMA图元的模具文件

⑤制作SAMA模板模具文件，其中包含四种图纸规格，分别为： A2/A3/A0/AX

其中该模具文件命名为DrawingsModule.VSS

**4.6.2 SAMA配置文件**

SAMA配置文件包括：参数表配置文件，横纵向功能块的逻辑配置文件。

①参数表配置文件

参数表配置文件针对的是功能块变量，该配置文件配置了类型，对应的参数项以及参数项的初始值。格式如下：

以PID为例：

[HSPID]

Count =7

NameValue0=OB|100.000000

NameValue1=OT|100.000000

NameValue2=PT|100.000000

NameValue3=TI|30.000000

NameValue4=TD|0.000000

NameValue5=KD|1.000000

NameValue6=IC|0.000000

其中该配置文件的名称为FBPara.ini

②横纵向功能块的逻辑配置文件

横纵向功能块的逻辑配置文件针对的某些功能块需要纵向排布，某些功能块逻辑需要横向排布，格式如下：

[Longitudinal\_logic]

Count=13

Name0=HSPID

Name1=HSINTG

Name2=INTEGRAL

Name3= HSDEV

Name4= HDEV

Name5= DERIVATIVE

Nmae6= HSALGMAN

Name7= HSVMAN

Name8= HSVMAN2

Name9= MANPH

Name10= HSFOP

Name11= HFOP

Name12=SEL

其中该配置文件的名称为Logic\_Config.ini