Mfc上位机与PLC通讯的实现

（1）主要是基于西门子提供NET软件提供的OPCServer服务器。

（2）OPC是采用客户端/服务器模式，为工业自动化软件提供统一的标准，定义了微软操作系统基于PC客户端之间的自动化实时数据交换。

（3）西门子NET，提供多个OPC服务器，为数据的外部访问提供接口。

（4）OPC客户端读取数据：同步，异步和订阅

**同步通讯**：客户端向服务器发送请求，客户端必须等到服务器响应全部完成才能返回，客户端在此期间一直处于等待状态，如果进行读操作，必须等待服务器响应后才返回。如果有大量数据操作的时候会造成客户端阻塞。

**异步通讯**：客户端对服务器进行请求，客户端程序请求后立即返回，不用等待服务器响应。OPC服务器响应之后再通知客户端程序，进行读操作，OPC客户端通知服务器之后返回，不等待服务器的读完成，当OPC服务器读完成之后，会自动通知客户程序，把读结果传送给客户端。效率高

**订阅方式**：客户端对服务器进行请求，客户端立即返回，不等待服务器操作，可以进行其他操作。服务器的组在组内有数据发生改变的时候，自动刷新相应的客户端数据。只向服务器发送一次请求。

（5）OPC写数据一般有两种：异步和同步，一般写数据参与控制，采用同步方式。

接口采用：自定义接口是一组COM（组件对象模型）接口，采用C++语言应用程序开发

（1）将PLC通过以太网卡连接到PLC

（2）构建一个OPC服务器端的组态配置

（3）地址和变量的配置

VC程序设计

（5）建立对话框界面

（6）变量定义，

（7）函数定义声明

（8）将变量和函数构建映射

（9）增加OPC外部引用文件

OLE：面向过程过

#include "stdafx.h"

#include "Pre\_OPC.h"

#include "Sync\_RW.h"

#include "Sync\_RWDlg.h"

（10）初始化COM控件，注册OPC组合OPCitem。

（11）同步读；

用C++语言的客户一般采用定制接口方案；OPC 数据存取服务器由三类对象组成:服务器(Server)、组(Group)、数据项(Item)。服务器对象用于指出特定的OPC服务器应用程序名，并作为组对象的容器;组对象存储由若干Item组成的Group信息并逻辑组织数据项;数据项对象()存储具体的Item的定义、数据值、状态值等信息，一个 Item就代表一个具体的过程变量。

建立OPC连接后，客户应用程序一般可以通过三种方式从OPC 服务器读取数据:使用同步接口IOPC- SyncIO，简单有效，适合于只读取少量数据的客户程序;使用接口IOPCCallback的“订阅”的功能OnChange，每当数据有变化时，服务器自动通知客户;使用异步接口IOPCASyncIO2，可以直接与物理设备通讯，速度慢但数据精确度高。

4 VC应用程序作为OPC客户端的程序实现

（1）包含OPC的头文件

（2）初始化COM支持库

（1）结构上的创新点：设计一个环形的胸鳍，（20个鳍条）实现融合推进，确保波形的连续和平滑，可以实现任意方向的矢量推进。

为什么需要20个胸鳍条：因为根据设计的大小，然后结合胸鳍摆动时候的波形，结合相容采样定理，得到一个鳍条数与胸鳍的基线大小关系，从而确定所需的鳍条数目。

而且为了每个鳍条采用一舵机实现摆动的功能，力度大，质量小。

主要采用的是硅胶防水，因为相对于每个鳍条来讲其实就是上下一定角度的摆动，硅胶结合防水胶就可以了

配重盘上水平面上垂直分布四个丝杠滑台机构，实现重心的调整从而实现姿态的变化。然后通过丝杠旋转带动活塞吸排水实现上浮和下沉。（步进电机）

（2）融合方案的创新：在魟鱼胸鳍摆动的基础上，引入尾鳍摆动推进机制，就是摆动和推动的方式同个

STM32的定时器和GPIO引脚控制五个超声传感器检测障碍，

（3）设计一种基于Hopf非线性振荡器的中心式CPG网络拓扑模型，相比于传统的链路式和网状的结构简单，而且耦合性小。所有的节点都只与中心节点相连，其余节点之间不存在耦合，都是通过中心节点计算相位差之后控制每个节点的角度。固定一个节点为0位起点，其他节点的相位。

直线巡游：CPG网络连续输出等幅震荡的控制信号。

原地转弯：利用左右胸鳍的非对称力矩，

//MFC界面开发：

（1）和底层socket通信不同

（2）主要使用的是西门子配套硬件。

（3）西门子提供了一个NET的软件能够在电脑上构建一个虚拟的OPC服务器，

（4）MFC界面的开发主要是对OPC内部的变量进行控制，PLC会通过CP5622板卡与opc服务器进行通信。

（5）在OPC中设置PLC变量的对应地址

（6）MFC中搭建主界面，登录界面

OPC由两套接口组成:客户端和服务器程序员使用的OPC自定义接口(OPC COM Custom Interfaces);支持用高端商业应用开发的客户程序的OPC自动化接口(OPC OLE Automation Interfaces)。COM接口效率高，通过该接口，客户能够发挥OPC服务器的最佳性能，采用C++语言的客户一般采用COM接口方案;自动化接口使解释性语言和宏访问OPC服务器成为可能，采用VB语言的客户一般采用自动化接口。自动化接口简化客户应用程序的实现，但运行时需要进行类型检查，牺牲了程序的运行速度。OPC自定义接口是服务器必须提供的，而自动化接口则不一定提供。典型的OPC体系结构如图1所示:

在Visual C++环境中实现OPC客户应用程序，首先必须从OPC国际基金会官方网站下载OPC头文件("opcda\_i.c"、"opcda.h" 、"opccomn\_i.c"、"opccomn.h")，并在Visual C++工程的“Tool”→“Options”→“Directories”加载头文件。然后再进行登陆COM、连接服务器、数据读写等操作。