Uses ADODB, ActiveX

//上传数据

function TClsDevice.UploadProcessData: Boolean;

var

i, iTrayID, iChanID, lStepID: Integer;

ip,database,user,pass, lFile: string;

qry: TADOQuery;

lRlt: Integer;

begin

Result:= True;

ip:= string(GSystemParam.DBServer.IP); //服务器ip（或名称）

database:= string(GSystemParam.DBServer.Database); //数据库名

user:= string(GSystemParam.DBServer.User); // 用户名

pass:= string(GSystemParam.DBServer.Pass); //密码

CoInitialize(nil);

qry:= TADOQuery.Create(nil);

try

try

if not SQLDataTest(ip,database,user,pass) then //连接检测

begin

Result:=False;

AppendLog('数据库连接异常', etWarnning, True);

Exit;

end;

for i:= 1 to C\_MAX\_CELL\_CNT\_PER\_DEV do

begin

ConvertIDToTrayID(iTrayID, iChanID, i, C\_DeviceType.dtCellCountPerTray);

if Trim(FTrays[iTrayID-1].TrayBarCode.CellCodes[iChanID]) = '' then Continue;

with qry, FTrays[iTrayID-1].TrayChannelDetailDatas[iChanID] do

begin

Close;

ConnectionString:= Format(C\_SQL\_CONN\_STR,[pass,user,database,ip]);

CommandTimeout:= 15;

SQL.Clear;

SQL.Add('exec UpdateFormationRecord :BarcodeNum,:EquipmentID,:ChamberID,:ChannelID,:StepNum,:StepProcess,:Voltage,:Current,:Capacity,'+

':Energy,:StepTime,:FlowTime,:CaptureTime,:Alarm,:Temperature,:PressForce,:CycleNum,:LineVoltage');

Parameters.ParamByName('BarcodeNum').Value := Trim(GetTray(iTrayID)^.TrayBarCode.CellCodes[iChanID]);

Parameters.ParamByName('EquipmentID').Value := 'Formation001';

Parameters.ParamByName('ChamberID').Value := IntToStr(FDeviceID);

Parameters.ParamByName('ChannelID').Value := IntToStr(i);

Parameters.ParamByName('StepNum').Value := IntToStr(ddStepNo);

Parameters.ParamByName('StepProcess').Value := GetStepStatusByValueEn(ddStepStatus);

Parameters.ParamByName('Voltage').Value := FloatToStr(ddVoltage);

Parameters.ParamByName('Current').Value := FloatToStr(ddCurrent);

Parameters.ParamByName('Capacity').Value := FloatToStr(ddCapacity);

Parameters.ParamByName('Energy').Value := FloatToStr(ddEnergy);

Parameters.ParamByName('StepTime').Value := MSecondToTimeFormat(ddStepTime);

Parameters.ParamByName('FlowTime').Value := MSecondToTimeFormat(ddScheduleTime);

Parameters.ParamByName('CaptureTime').Value := FormatDateTime('yyyy-MM-dd hh:mm:ss', ddTestTime);

Parameters.ParamByName('Alarm').Value := IntToStr(ddChanInfo);

Parameters.ParamByName('Temperature').Value := FloatToStr(ddT);

Parameters.ParamByName('PressForce').Value := FloatToStr(ddPressure);

Parameters.ParamByName('CycleNum').Value := IntToStr(ddCycleNo);

Parameters.ParamByName('LineVoltage').Value := FloatToStr(ddVoltage\_I);

Sleep(50);

Result:= ExecSQL >0;

end;

end;

except

on E: Exception do

begin

Result:= False;

raise Exception.Create('UploadProcessData ' + E.Message);

end;

end;

finally

FreeAndNil(qry);

CoUninitialize;

end;

end;

//数据库连接测试

function SQLDataTest(var ip,database,user,pass: string): Boolean;

var

dmd: TADOConnection;

begin

try

Result:= False;

CoInitialize(nil);

dmd:= TADOConnection.Create(nil);

try

dmd.Connected:=False;

dmd.ConnectionString:= Format(C\_SQL\_CONN\_STR,[pass,user,database,ip]);

dmd.ConnectionTimeout:=1;

dmd.Connected:=True;

Result:= True;

except

Result:= False;

end;

finally

FreeAndNil(dmd);

CoUninitialize;

end;

end;

C\_SQL\_CONN\_STR =

'Provider=SQLOLEDB.1;Password=%s;' + //密码

'Persist Security Info=True;User ID=%s;' + //用户

'Initial Catalog=%s;' + //数据库

'Data Source=%s';