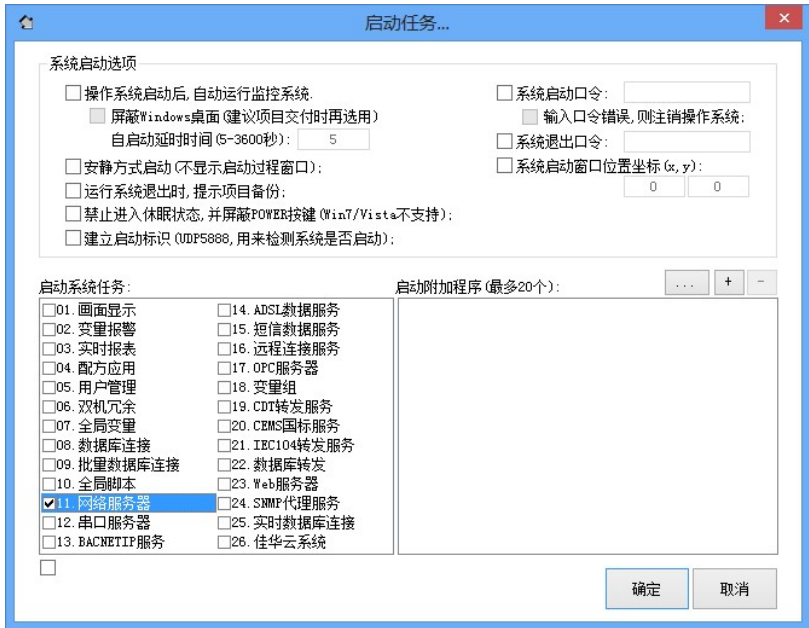


28.1 启动网络服务器

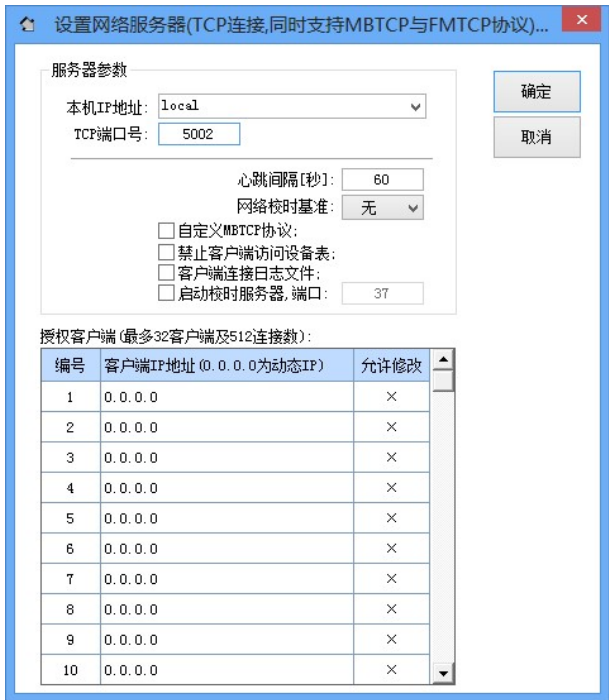
- 选择[我的系统->设置]功能, 执行[2. 启动任务]:



- 选择系统任务中“☒11. 网络服务器”;

28.2 设置网络服务器

- 选择[数据服务->网络服务器]功能, 执行[设置网络服务器]任务:



- 本机 IP 地址:
 - [1]. 设定 local, 服务器启动时自动检测本机网卡 IP 地址, 并使本机所有网卡都提供网络服务;
 - [2]. 便于项目移置, 建议使用 local 参数;
 - [3]. 如果计算机有多块网卡, 须指定某网卡提供服务, 则需正确填写相应网卡 IP 地址;
- TCP 端口号, 缺省服务端口 5002, 取值范围 100-32000;
根据报文格式, 同时提供 FMTCP 或 MBTCP 协议的客户端连接访问;
- 心跳间隔, 以秒为单位, 范围 3-600 秒, 缺省 60 秒;
检测客户端心跳间隔时间段内, 没有进行数据访问, 认为客户端故障, 关闭客户端连接;
- 校时基准, 客户端使用 FMTCP 驱动, 允许客户端和服务端时钟同步:
 - [1]. 以服务器为基准, FMTCP 驱动连接服务器时, 得到服务器时钟作为基准, 设置客户端时钟;
 - [2]. 以客户端为基准, FMTCP 驱动连接服务器时, 发送客户端时钟作为基准, 设置服务器时钟;
 - [3]. 选择[无], 不进行时钟同步;
- 禁止客户端访问设备表, 使用 FMTCP 和 MBTCP 协议, 只能访问变量包数据, 而不允许访问设备表;
- 客户端连接日志文件, 监控系统 Temp 子目录下生成日志文件, 记录客户端连接信息:



- 启动校时服务器, 缺省端口 37, 缺省范围 1-30000;
 - [1]. 网络服务器可同时再开放某端口作为校时服务器;
 - [2]. 连接校时服务器时, 服务器立刻返回当前时间值, 并关闭客户端连接;
 - [3]. 时间值为当前时间距 (1900-1-1 0:0:0) 秒间隔, 高字节在前, 低字节在后;
 - [4]. 使用 TCP 调试测试:



D6 24 CA AB => 2013-11-06 14:03 23

```
CTimeSpan timeSpan(0,0,0,0xD6*0x1000000+0x24*0x10000+
0xCA*0x100+0xAB-3155673600);
CTime curTime=CTime(2000,1,1,0,0,0)+timeSpan;
CString timeDesc=curTime.Format("%Y-%m-%d %H:%M:%S");
```

- 最多支持 32 个客户端, 最大并发连接数 512;

实际支持客户端数量由加密狗授权决定, 默认客户端数量不足时, 需要再单独购买;

变量点数	客户数量	变量点数	客户数量	变量点数	客户数量
30	1	1024	2	150K	20
64	1	2048	2	300/500K	30
128	1	4096	2	绑定软授权	16
256	1	64K	2	变量服务器	32
512	1	100K	20	演示版	30

- 定义客户端访问;

[1]. 默认设置, 任何客户端允许连接服务器, 但只读取服务器数据, 不支持修改;

编号	客户端IP地址 (0.0.0.0为动态IP)	允许修改
1	0.0.0.0	×
2	0.0.0.0	×
3	0.0.0.0	×
4	0.0.0.0	×
5	0.0.0.0	×
6	0.0.0.0	×
7	0.0.0.0	×
8	0.0.0.0	×
9	0.0.0.0	×
10	0.0.0.0	×

[2]. 需要限制客户端访问, 只允许某些具有特定 IP 的客户端连接;

鼠标双击表格固定行第 2 列, 全部清除 0.0.0.0, 双击其他行第 2 列输入指定 IP:

编号	客户端IP地址 (0.0.0.0为动态IP)	允许修改
1	192.168.2.105	×
2	192.168.2.107	×
3		
4		
5		
6		
7		
8		
9		
10		

按 DEL 键可以删除某行设置;

[3]. 允许客户端修改服务器数据, 鼠标双击表格第 3 列, 切换访问方式:

编号	客户端IP地址 (0.0.0.0为动态IP)	允许修改
1	192.168.2.105	×
2	192.168.2.107	✓
3		
4		
5		
6		
7		
8		
9		
10		

编号	客户端IP地址 (0.0.0.0为动态IP)	允许修改
1	0.0.0.0	✓
2	0.0.0.0	✓
3	0.0.0.0	✓
4	0.0.0.0	✓
5	0.0.0.0	✓
6	0.0.0.0	✓
7	0.0.0.0	✓
8	0.0.0.0	✓
9	0.0.0.0	✓
10	0.0.0.0	✓

鼠标双击固定行第 3 列, 批量修改;

28.3 定义数据变量包

□ 网络服务器提供访问数据类型：

- [1]. 设备数据表 (D1-D2000)；
- [2]. AI/AO/AR/VA 模拟变量, DI/DO/DR/VD 开关变量；

服务器设备数据表 (D1-D2000) 允许被客户端直接访问；

AI/AO/AR/VA 变量、DI/DO/DR/VD 变量需要被打包为变量包, 才能被客户端访问；

□ 选择[网络服务器]功能, 执行[定义变量包]任务：

定义通讯变量包[AP/DP]...

地址	AP - 模拟变量包:	数值[R4]	地址	DP - 开关变量包:	状态[位]
0	VA. %VA1		0.0	VD. %VD1	
1	VA. %VA2		0.1	VD. %VD2	
2	VA. %VA3		0.2	VD. %VD3	
3	VA. %VA4		0.3	VD. %VD4	
4			0.4	VD. %VD5	
5			0.5	VD. %VD6	
6			0.6	VD. %VD7	
7			0.7		
8			1.0		
9			1.1		
10			1.2		
11			1.3		
12			1.4		
13			1.5		
14			1.6		
15			1.7		
16			2.0		
17			2.1		
18			2.2		
19			2.3		
20			2.4		
21			2.5		
22			2.6		
23			2.7		

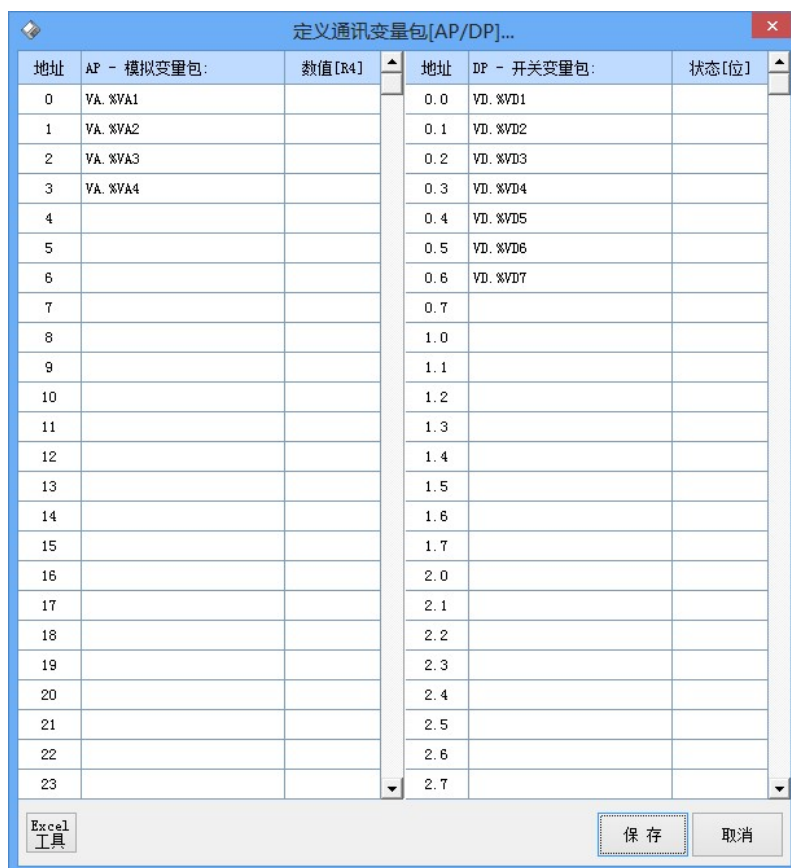
Excel
工具

保存取消

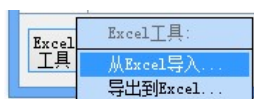
- [1]. 包含两个表格, 分别定义模拟变量包 (AP) 和开关变量包 (DP)；
- [2]. 模拟变量包 (AP), 最多包含 4096 个 32 位浮点数 (R4)；
客户端通过索引地址 (0-4095) 访问模拟变量包；
- [3]. 开关变量包 (DP), 最多包含 512*8 共 4096 个位变量 (bit)；
客户端通过索引地址 (0.0-0.7、...、511.0-511.7) 访问开关变量包；
- [4]. 变量包每个索引地址对应运行数据库某变量；
未对应变量的索引地址, 允许被访问, 但其缺省值/状态总为 0；
- [5]. 鼠标双击变量包索引地址, 从变量列表中连续选择变量直到结束, 按下 Del 键删除某变量；

- 允许通过 Excel 编辑变量包, 提高组态效率;

[1]. 简单编辑变量包:



[2]. 执行[Excel 工具]按钮:



[3]. 执行[导出到 Excel], 导出变量包到 Excel 文件:

	A	B	C	D	E
1		模拟变量包		开关变量包	
2		VA. %VA1		VD. %VD1	
3		VA. %VA2		VD. %VD2	
4		VA. %VA3		VD. %VD3	
5		VA. %VA4		VD. %VD4	
6				VD. %VD5	
7				VD. %VD6	
8				VD. %VD7	
9					

[4]. 使用 Excel 编辑变量包并保存:

	A	B	C	D	E
1		模拟变量包		开关变量包	
2		VA. %VA1		VD. %VD1	
3		VA. %VA2		VD. %VD2	
4		VA. %VA3		VD. %VD3	
5		VA. %VA4		VD. %VD4	
6		VA. %RAND		VD. %VD5	
7		VA. %X=X+1		VD. %VD6	
8				VD. %VD7	
9				VD. %VD8	

[5]. 执行[从 Excel 导入], 选择导入 Excel 文件名称, 导入变量包内容:

地址	AP - 模拟变量包:	数值[R4]	地址	DP - 开关变量包:	状态[位]
0	VA. %VA1		0.0	VD. %VD1	
1	VA. %VA2		0.1	VD. %VD2	
2	VA. %VA3		0.2	VD. %VD3	
3	VA. %VA4		0.3	VD. %VD4	
4	VA. %RAND		0.4	VD. %VD5	
5	VA. %X=X+1		0.5	VD. %VD6	
6			0.6	VD. %VD7	
7			0.7	VD. %VD8	
8			1.0		
9			1.1		
10			1.2		
11			1.3		
12			1.4		
13			1.5		
14			1.6		
15			1.7		
16			2.0		
17			2.1		
18			2.2		
19			2.3		
20			2.4		
21			2.5		
22			2.6		
23			2.7		

Excel 工具 保存 取消

□ 运行状态打开变量包, 监视索引地址对应的数值和状态;

地址	AP - 模拟变量包:	数值[R4]	地址	DP - 开关变量包:	状态[位]
0	VA. %VA1	0	0.0	VD. %VD1	0
1	VA. %VA2	0	0.1	VD. %VD2	1
2	VA. %VA3	0	0.2	VD. %VD3	0
3	VA. %VA4	0	0.3	VD. %VD4	0
4	VA. %RAND	2987.456893	0.4	VD. %VD5	1
5	VA. %X=X+1	284	0.5	VD. %VD6	0
6			0.6	VD. %VD7	0
7			0.7	VD. %VD8	0
8			1.0		
9			1.1		
10			1.2		
11			1.3		
12			1.4		
13			1.5		
14			1.6		
15			1.7		
16			2.0		
17			2.1		
18			2.2		
19			2.3		
20			2.4		
21			2.5		
22			2.6		
23			2.7		

28.4 监视网络服务器

- 运行状态时, 服务器被启动, 选择[网络服务器]功能, 执行[监视网络服务器]任务:

序号	客户端[≤100]	端口号	连接时间	读写计数	允许修改
1	192.168.2.104	61518/TCP	2013-11-11 09:12:22	140	×
2	0.0.0.0				
3	0.0.0.0				
4	0.0.0.0				
5	0.0.0.0				
6	0.0.0.0				
7	0.0.0.0				
8	0.0.0.0				
9	0.0.0.0				
10	0.0.0.0				
11	0.0.0.0				
12	0.0.0.0				
13	0.0.0.0				
14	0.0.0.0				
15	0.0.0.0				
16	0.0.0.0				

客户端连接到服务器, 显示客户端 IP、端口号、连接时间、访问计数等信息;

- 客户端并发连接到服务器, 拖动滚动条到下方, 监视并发连接:

序号	客户端[≤100]	端口号	连接时间	读写计数	允许修改
31	0.0.0.0				
32	0.0.0.0				
[1]	192.168.2.104	61519/TCP	2013-11-11 09:12:22	200	×
[2]	192.168.2.104	61520/TCP	2013-11-11 09:12:22	200	×
[3]	192.168.2.104	61521/TCP	2013-11-11 09:12:22	200	×
[4]	192.168.2.104	61522/TCP	2013-11-11 09:12:22	190	×
[5]	0.0.0.0				
[6]	0.0.0.0				
[7]	0.0.0.0				
[8]	0.0.0.0				
[9]	0.0.0.0				
[10]	0.0.0.0				
[11]	0.0.0.0				
[12]	0.0.0.0				
[13]	0.0.0.0				
[14]	0.0.0.0				

28.5 FMTCP 通讯协议

- 客户端建立 Socket 端口, 以 TCP 方式连接服务器指定端口;
使用 FMTCP 协议与网络服务器同步方式交换数据;

□ 功能码列表

功能代码	功能描述	功能代码	功能描述
0	测试. 协议		
访问. 设备数据表			
1-1000	读取. 设备号 D1-1000	1001-2000	修改. 设备号 D1-1000
11001-12000	读取. 设备号 D1001-2000	12001-13000	修改. 设备号 D1001-2000
0x4D52	读取. 批量设备表	0x4D57	修改. 批量设备表
访问. 运行数据库			
10000	登录. 访问运行数据库口令验证. 如果未设置口令则免验证		
10001	读取. 某变量索引值		
10002	读取. 某变量值	10003	修改. 某变量值
10004	读取. 某批变量值	10005	修改. 某批变量值
10011	读取. 某类变量全部变量名称及对应索引值		
10012	读取. 通过变量索引, 批量读取某类变量全部或某段时间内发生变化的变量值		
10013	修改. 通过变量索引, 批量修改某类变量中某些变量值		
10014	读取. 通过变量索引, 批量读取某类变量中某些变量值		
10015	读取. 批量读取某类变量中某些变量的索引值		
访问. 变量包			
2001	读取. 模拟变量包	2011	修改. 模拟变量包
2002	读取. 开关变量包	2012	修改. 开关变量包
2005-2007	读取. 模拟变量包. 整数值. 系数(1/10/100)		
其他. 特殊功能			
2003	校时. 服务器与客户端间时间		
2008	读取. 数据库记录包		
2010	用户. 登录及注销		
2018	执行. 字符串格式函数命令		

返回错误代码

错误代码	错误描述	错误代码	错误描述
0xFFFF0	接收报文不完整	0xFFFFD	设备号禁止访问
0xFFFF1	访问数据地址不正确	0xFFFFE	运行数据库访问失败
0xFFFF2	访问数据长度不正确	0xFFFFF	无效功能码
0xFFFF3	禁止修改数据		

□ 测试服务器(心跳):

客户端发出请求(16 进制)		
例如:3E 2A 00 00		
Buffer[0][1]	3E 2A	发送标识
Buffer[2][3]	00 00	功能码(0000):测试
服务器正常应答(16 进制)		
例如:3C 2A 00 00		
Buffer[0][1]	3C 2A	接收标识
Buffer[2][3]	00 00	功能码(0000):测试

□ 被设置以客户端为基准, 进行网络校时:

客户端发出校时请求, 把客户端当前时间发送给服务器:		
Buffer[0][1]	3E 2A	协议标识
Buffer[2][3]	07D3 (2003)	功能码=07D3 (2003)
Buffer[4].[10]	年, 月, 日, 时, 分, 秒; 格式:YYYY mm dd HH MM SS	
服务器根据客户端发来的时间改变本地时间后返回:		
3C 2A 07 D3 00 00 00 00 00 00 00 00		

□ 被设置以服务器为基准, 进行网络校时:

客户端发出校时请求, 要求得到服务器时间:		
Buffer[0][1]	3E 2A	发送标识
Buffer[2][3]	07D3 (2003)	功能码=07D3 (2003)
Buffer[4].[10]	年, 月, 日, 时, 分, 秒; 格式:YYYY mm dd HH MM SS	
服务器返回服务器时间, 客户端根据服务器时间改变客户端时间:		
Buffer[0][1]	3C 2A	接收标识
Buffer[2][3]	07D3 (2003)	功能码=07D3 (2003)
Buffer[4].[10]	年, 月, 日, 时, 分, 秒; 格式:YYYY mm dd HH MM SS	

□ 访问运行数据库口令验证. 未设置口令免认证:

客户端请求认证. 例如: 3E 2A 27 10 '12345' 00 00 00 00 00		
Buffer[0][1]	3E 2A	请求标识
Buffer[2][3]	27 10	功能码=2710(10000)
Buffer[4]..[13]	验证口令	10 位口令, 不足补 0
服务器验证返回验证结果. 例如: 3C 2A 27 10 00		
Buffer[0][1]	3C 2A	接收标识
Buffer[2][3]	27 10	功能码=2710(10000)
Buffer[4]	00 FF	验证结果: 成功(00) 失败 (FF)

□ 读取设备号 D1-D1000, 每设备号 1024 字节, 地址表示 0-1023;

客户端发出请求(16 进制)			
例如:3E 2A <u>00 01 00 00 00 0A</u> (设备号 D1, 从字节单元 0 开始, 读 10 个字节)			
Buffer[0][1]	3E 2A	发送标识	异步读取方式
	3E 23(26)		同步读取方式
Buffer[2][3]	[0001-03E8]	功能码(1-1000):读设备号数据 [0001-03E8]对应 D1-D1000	
Buffer[4][5]	开始字节	0-1023	高字节前低字节后
Buffer[6][7]	字节长度	1-1024	
设备号通讯正常, 服务器返回应答(16 进制)			
例如: <u>3C 2A 00 01 00 0A 00 01 02 03 04 05 06 07 08 09</u>			
Buffer[0][1]	3C 2A(23/26)	接收标识	
Buffer[2-3]	[0001-03E8](1-1000)	功能码(1-1000):读设备号数据 [0001-03E8]对应 D1-1000	
Buffer[4][5]	字节长度	1-1024	高字节前低字节后
Buffer[6-n]	返回数据		
设备号通讯不正常, 服务器返回应答(16 进制), 例如: <u>3C 2A 00 01 00 00</u>			

□ 修改设备号 D1-D1000, 每设备号 1024 字节, 地址表示 0-1023;

客户端发出请求(16 进制)			
例如: <u>3E 2A 03 E9 00 01 00 02 01 02</u> (修改设备号 D1, 从字节单元 1 开始的 2 个字节)			
Buffer[0][1]	3E 2A	发送标识	异步修改方式
	3E 23		同步修改方式
Buffer[2][3]	[03E9-07D0]	功能码(1001-2000):修改设备号数据 [03E9-07D0]对应 D1-1000	
Buffer[4][5]	开始字节	0-1023	高字节前低字节后
Buffer[6][7]	字节长度	1-1024	
Buffer[8-n]..	修改数据		
修改设备号成功, 服务器应答(16 进制)			
例如: <u>3C 2A 03 E9</u>			
Buffer[0][1]	3C 2A(23)	接收标识	
Buffer[2][3]	[03E9-07D0]	功能码(1001-2000):修改设备号数据 1001-2000[03E9-07D0]对应 D1-1000	

□ 读取设备号 D1001-D2000, 每设备号 1024 字节, 地址表示 0-1023;

客户端发出请求(16 进制)			
例如: 3E 2A 2A F9 00 00 00 0A （设备号 D1001, 从字节单元 0 开始, 读 10 个字节）			
Buffer[0][1]	3E 2A	发送标识	异步读取方式
	3E 23(26)		同步读取方式
Buffer[2][3]	[2AF9-2EE0]	功能码(11001-12000): 读设备号数据 [2AF9-2EE0] 对应 D1001-D2000	
Buffer[4][5]	开始字节	0-1023	高字节前低字节后
Buffer[6][7]	字节长度	1-1024	
设备号通讯正常, 服务器返回应答(16 进制)			
例如: 3C 2A 2A F9 00 0A 00 01 02 03 04 05 06 07 08 09			
Buffer[0][1]	3C 2A(23/26)	接收标识	
Buffer[2][3]	[2AF9-2EE0]	功能码(11001-12000): 读设备号数据 [2AF9-2EE0] 对应 D1001-D2000	
Buffer[4][5]	字节长度	1-1024	高字节前低字节后
Buffer[6-n]	返回数据		
设备号通讯不正常, 服务器返回应答(16 进制), 例如: 3C 2A 2A F9 00 00			

□ 修改设备号 D1001-D2000, 每设备号 1024 字节, 地址表示 0-1023;

客户端发出请求(16 进制)			
例如: <u>3E 2A 2E E1 00 01 00 02 01 02</u> (修改设备号 D1001, 从字节 1 开始的 2 个字节)			
Buffer[0][1]	3E 2A	发送标识	异步修改方式
	3E 23		同步修改方式
Buffer[2][3]	[2EE1-32C8]	功能码 (12001-13000):修改设备号数据 [2EE1-32C8]对应 D1001-D2000	
Buffer[4][5]	开始字节	0-1023	高字节前低字节后
Buffer[6][7]	字节长度	1-1024	
Buffer[8-n]	修改数据		
修改设备号成功, 服务器应答(16 进制)			
例如: <u>3C 2A 2E E1</u>			
Buffer[0][1]	3C 2A(23)	接收标识	
Buffer[2-3]	[2EE1-32C8]	功能码 (12001-13000):修改设备号数据 [2EE1-32C8]对应 D1001-D2000	

□ 读取模拟变量包 (AP)

客户端发出请求(16 进制)			
例如: <u>3E 2A 07 D1 00 02 00 02</u> (读 2 个模拟变量:AP[2]、AP[3])			
Buffer[0][1]	3E 2A	发送标识	
Buffer[2][3]	07D1 (2001)	功能码=07D1 (2001)	
Buffer[4][5]	开始变量索引号	0-4095	高字节前低字节后
Buffer[6][7]	变量数量	1-1024	
服务器返回应答(16 进制)			
例如: <u>3C 2A 07 D1 00 08 00 00 00 00 00 00 00 00</u>			
Buffer[0][1]	3C 2A	发送标识	
Buffer[2][3]	07D1 (2001)	功能码=07D1 (2001)	
Buffer[4][5]	数据字节长度	(1-1024)*4	高字节前低字节后
Buffer[6-n]	返回数据	浮点数格式, 每变量 4 字节	

□ 读取开关变量包 (DP)

客户端发出请求(16 进制)			
例如: <u>3E 2A 07 D2 00 02 00 02</u> (读 2 个字节的开关变量:DP[2]、DP[3])			
Buffer[0][1]	3E 2A	发送标识	
Buffer[2][3]	07D2 (2002)	功能码=07D2 (2002)	
Buffer[4][5]	开始字节	0-511	高字节前低字节后
Buffer[6][7]	字节数量	1-512	
服务器返回应答(16 进制)			
例如: <u>3C 2A 07 D2 00 02 01 02</u>			
Buffer[0][1]	3C 2A	接收标识	
Buffer[2][3]	07D2 (2002)	功能码=07D2 (2002)	
Buffer[4][5]	字节长度	1-512	高字节前低字节后
Buffer[6-n]	返回数据	字节格式	

□ 修改模拟变量包 (AP)

客户端发出请求(16 进制)			
例如: <u>3E 2A 07 DB 00 08 00 08 00 00 F6 42 00 00 E4 43</u> (修改 AP[2]=123、AP[3]=456)			
Buffer[0][1]	3E 2A	发送标识	
Buffer[2][3]	07DB(2011)	功能码=07DB(2011)	
Buffer[4][5]	开始字节	[0-4095]*4	高字节前低字节后
Buffer[6][7]	字节数量	[1-1024]*4	
Buffer[8-n]	修改数据	浮点数据格式，每变量 4 字节	
服务器返回应答(16 进制)			
例如: <u>3C 2A 07 DB</u>			
Buffer[0][1]	3E 2A	接收标识	
Buffer[2-3]	07DB(2011)	功能码=07DB(2011)	

□ 修改开关变量包 (DP)

客户端发出请求(16 进制)			
例如: <u>3E 2A 07 DC 00 00 00 01 FF</u> (修改 DP[0]=FF)			
Buffer[0][1]	3E 2A	发送标识	
Buffer[2][3]	07DC (2012)	功能码=07DC (2012)	
Buffer[4][5]	开始字节	0-511	高字节前低字节后
Buffer[6][7]	字节长度	1-512	
Buffer[8-n]	修改数据	字节格式	
服务器返回应答(16 进制)			
例如: <u>3C 2A 07 DC</u>			
Buffer[0][1]	3E 2A	接收标识	
Buffer[2][3]	07DC (2012)	功能码=07DC (2012)	

□ 读取模拟变量包 (AP) (乘系数, 模拟量由浮点转换为 2 字节整数) :

客户端发出请求(16 进制)			
例如: <u>3E 2A 07 D4 00 02 00 02</u> (读 2 个模拟变量:AP[2]*10、AP[3]*10)			
Buffer[0][1]	3E 2A	发送标识	
Buffer[2][3]	07D5 (2005)	变量包值*1, 转换为整数	
	07D6 (2006)	变量包值*10, 转换为整数	
	07D7 (2007)	变量包值*100, 转换为整数	
Buffer[4][5]	开始索引号	0-4095	高字节前低字节后
Buffer[6][7]	变量数量	1-1024	
服务器返回应答(16 进制)			
例如: <u>3C 2A 07 D4 00 04 00 00 00 00</u>			
Buffer[0][1]	3C 2A	接收标识	
Buffer[2][3]	07D5/6/7 (2005/6/7)	功能码=07D3/4/5 (2003/4/5)	
Buffer[4][5]	数据字节长度高低字节	(1-1024)*2	高字节前低字节后
Buffer[6-n]	返回数据	两字节整数, 每 2 个字节为 1 个变量	

□ 用户管理登录或注销

客户端发出请求

例(登录):3E 2A 07 DA 01 0C "user1,123456"

例(注销):3E 2A 07 DA 02

Buffer[0][1]	3E 2A	发送标识
Buffer[2][3]	07 DA(2010)	功能码
Buffer[4]	01 02	01=登录, 02=注销
Buffer[5]	后续字节数量	注销不需要
Buffer[6]...	"用户名称, 口令"	

服务器返回应答(16 进制)

例如:3C 2A 07 DA 01 00

Buffer[0][1]	3C 2A	接收标识
Buffer[2][3]	07 DA(2010)	功能码
Buffer[4]	01 02	01=登录, 02=注销
Buffer[5]	00=成功, 01=未启动用户管理, 02=用户信息错误,	

□ 批量读取设备数据表

客户端发出请求(16 进制), 例如(读设备号 D2-D3, 单元 0 开始 2 个字节):				
<u>3E 2A 4D 52 00 02 00 03 00 00 00 02</u>				
Buffer[0][1]	3E 2A	'>*'	发送标识	异步读取方式
	3E 23(26)	'>#','>&'		同步读取方式
Buffer[2][3]	4D 52	'MR'	功能码: 批量读取设备号数据	
Buffer[4][5]	开始设备号(n1)		1-2000	高字节前低字节后
Buffer[6][7]	结束设备号(n2)		1-2000	
Buffer[8][9]	设备号开始字节, 高低字节		0-1023, 1028, 1029	
Buffer[10][11]	字节单元长度(1), 高低字节		1-1024, 1	
服务器应答(16 进制)				
例如: <u>3C 2A 4E 20 00 00 06 xx xx xx xx ss ss</u>				
Buffer[0][1]	3C 2A	'<*'	接收标识	异步读取方式
	3C 23(26)	'<#','<&'		同步读取方式
Buffer[2-3]	4D 52	'MR'	功能码: 批量读取设备号数据	
Buffer[4]..[6]	后续字节数		高字节前低字节后	
Buffer[7]..	返回设备号单元数据		字节数=(n2-n1+1)*1	
Buffer[n]..	返回设备号通讯状态		字节数=(n2-n1+1), 0=正常, 1=中断	

□ 批量修改设备数据表

客户端发出请求(16 进制), 例如(修改设备号 D1-D2, 单元 0 开始 2 字节):			
<u>3E 2A 4D 57 00 01 00 02 00 00 00 02 00 00 04 xx xx xx xx</u>			
Buffer[0][1]	3E 2A	'>*'	发送标识
	3E 23	'>#'	
Buffer[2][3]	4D 57	'MW'	功能码: 批量修改设备号数据
Buffer[4][5]	开始设备号(n1)	1-2000	高字节前低字节后
Buffer[6][7]	结束设备号(n2)	1-2000	
Buffer[8][9]	设备号开始字节	0-1023	
Buffer[10][11]	字节长度(1)	1-1024	
Buffer[12][13]	后续字节数量	最多 8192 字节	
Buffer[14]...	修改内容	字节数=(n2-n1+1)*1	
服务器应答(16 进制)			
例如: <u>3C 2A 75 30 00 02 ss ss</u>			
Buffer[0][1]	3C 2A	'<*'	接收标识
	3C 23	'<#'	
Buffer[2-3]	4D 57	'MW'	功能码: 批量修改设备号数据
Buffer[4][5]	后续字节长度, 高低字节	(n2-n1+1)	同步修改方式
Buffer[6]...	修改设备号成功标志	字节数=(n2-n1+1)	
	00=成功, 01=失败, 10=无需修改, 11=中断, 21=只读		

□ 访问运行数据库某变量

[1]. 得到某变量索引值:

客户端发出请求 (16 进制)		
例如: <u>3E 2A 27 11 07 56 41 31 00 ... 00</u> (得到 VA.VA1 的变量索引)		
Buffer[0][1]	3E 2A	发送标识
Buffer[2][3]	27 11	功能码=2711(10001)
Buffer[4]	varType	变量类型: AI=1, AO=2, AR=3, DI=4, DO=5, DR=6, VA=7, VD=8, VT=9
Buffer[5][6]...[25]	varName	变量名称, 最大长度 20, 不足补 0
服务器返回应答 (16 进制)		
例如: <u>3C 2A 27 11 02 00 00 00</u>		
Buffer[0][1]	3C 2A	接收标识
Buffer[2][3]	27 11	功能码=2711(10001)
Buffer[4][5][6][7]	varIndex	变量索引值 (0-300000), 低字节前高字节后 如果变量不存在, 返回-1:

[2]. 得到某变量值:

客户端发出请求 (16 进制)		
例如: <u>3E 2A 27 12 07 02 00 00 00</u>		
Buffer[0][1]	3E 2A	发送标识
Buffer[2][3]	27 12	功能码=2712(10002)
Buffer[4]	varType	变量类型: AI=1, AO=2, AR=3, DI=4, DO=5, DR=6, VA=7, VD=8, VT=9
Buffer[5][6][7][8]	varIndex	变量索引值 (0-300000), 低字节前高字节后
服务器返回应答 (16 进制)		
例如: <u>3C 2A 27 11 02 00 00 00</u>		
Buffer[0][1]	3C 2A	接收标识
Buffer[2][3]	27 12	功能码=2712(10002)
Buffer[4][5]...[12]	08 xx xx xx..xx	模拟变量 (AI/AO/AR/VA), 8 字节双精度
Buffer[4][5]	01 xx	开关变量 (DI/DO/DR/VD), 1 字节整数
Buffer[4][5]...[n]	LL xx xx xx ..xx	文本变量 (VT), 1-200 字节字符串
Buffer[4]	00	变量错误

[3]. 修改某变量值:

客户端发出请求 (16 进制)		
例如: <u>3E 2A 27 13 07 02 00 00 00 08 00 00 00 00 00 00</u>		
Buffer[0][1]	3E 2A	发送标识
Buffer[2][3]	27 13	功能码=2713(10003)
Buffer[4]	varType	变量类型: A0=2, AR=3, D0=5, DR=6, VA=7, VD=8, VT=9
Buffer[5][6][7][8]	varIndex	变量索引值 (0-300000), 低字节前高字节后
Buffer[9][10]..[17]	08 xx xx..xx	模拟变量值 (A0/AR/VA), 8 字节双精度
Buffer[9][10][11][12]	04 xx xx xx xx	模拟变量值 (A0/AR/VA), 4 字节单精度
Buffer[9][10]	01 xx	模拟变量值 (D0/DR/VD), 1 字节整数
Buffer[9][10]..[n]	LL xx xx ..xx	文本变量 (VT), 1-200 字节字符串
服务器返回应答 (16 进制)		
例如: <u>3C 2A 27 13 00</u>		
Buffer[0][1]	3C 2A	接收标识
Buffer[2][3]	27 13	功能码=2713(10003)
Buffer[4]	0/1/2/3/4	返回代码, 修改成功为 0

[4]. 得到多个变量值:

客户端发出请求 (16 进制)		
例如: <u>3E 2A 27 14 08 03 01 00 00 00 02 00 00 00 03 00 00 00</u> (得到 3 个 VD 变量)		
Buffer[0][1]	3E 2A	发送标识
Buffer[2][3]	27 14	功能码=2714(10004)
Buffer[4]	varType	变量类型: AI=1, A0=2, AR=3, DI=4, D0=5, DR=6, VA=7, VD=8
Buffer[5]	varCount	变量数量 (1-255)
Buffer[6][7][8][9]	varIndex_1	变量 1 索引值 (0-300000)
...
Buffer[m]. [m+3]	varIndex_n	变量 n 索引值 (0-300000)
服务器返回应答 (16 进制)		
例如: <u>3C 2A 27 14 06 00 01 00 01 00 00</u>		
Buffer[0][1]	3C 2A	接收标识
Buffer[2][3]	27 14	功能码=2714(10004)
Buffer[4][5]	totalLength	变量值内容总长度: AI/A0/AR/VA (8*n), DI/D0/DR/VD (1*n)
Buffer[6][7]..[14]	00 xx xx ..xx	模拟变量值 (AI/A0/AR/VA):
Buffer[15][16]..[23]	01 xx xx .. xx	标识 (0/1)+8 字节双精度浮点
Buffer[6][7]	00 00	开关变量值 (DI/D0/DR/VD):
Buffer[8][9]	01 00	标识 (0/1)+1 字节

[5]. 修改多个变量值:

客户端发出请求 (16 进制)		
例如: <u>3E 2A 27 15 08 02 01 00 00 00 01 02 00 00 00 01</u>		
Buffer[0][1]	3E 2A	发送标识
Buffer[2][3]	27 15	功能码=2715(10005)
Buffer[4]	varType	变量类型: A0=2, AR=3, D0=5, DR=6, VA=7, VD=8
Buffer[5]	varCount	变量个数 (1-255)
Buffer[6][7][8][9]	varIndex_1	模拟变量 (AO/AR/VA) 8 字节双精度浮点数
Buffer[10][11]..[17]	varValue_1	
Buffer[18][19][20][21]	varIndex_2	
Buffer[22][23]..[29]	varValue_2	开关变量 (D0/DR/VD) 1 字节
Buffer[6][7][8][9]	varIndex_1	
Buffer[10]	varValue_1	
Buffer[11][12][13][14]	varIndex_2	
Buffer[15]	varValue_2	
服务器返回应答 (16 进制)		
例如: <u>3C 2A 27 15 02 00 00</u>		
Buffer[0][1]	3C 2A	接收标识
Buffer[2][3]	27 15	功能码=2715(10005)
Buffer[4]	varCount	变量个数 (0-100)
Buffer[5]	00	各变量修该状态 (0/1/2)
Buffer[6]	01	

□ 批量读写运行数据库变量

[1]. 获取某类型全部变量名称及变量索引;

客户端发出请求(16 进制)			
例如:3E 2A 27 1B 01 00 00 00 00 00 00			
Buffer[0][1]	3E 2A	请求标识	
Buffer[2][3]	27 1B	功能码=271B(10011)	
Buffer[4]	变量类型	AI=01, AO=02, AR=03, VA=07 DI=04, DO=05, DR=06, VD=08 VT=09	获取变量名
		AI=11, AO=12, AR=13, VA=17	获取变量名. 附量程
		AI=21, AO=22, AR=23	获取变量名. 附量程及报警
Buffer[5][6][7]	开始索引	0-500000	高字节前, 低字节后
Buffer[8][9][10]	变量数量	0:最大数量, 1-500000:部分	
服务器返回应答(16 进制)			
例如:3C 2A 27 1B 01 00 00 00 02 07 00 00 01 'var1' 07 00 00 02 'var2' 00			
Buffer[0][1]	3C 2A	应答标识	
Buffer[2][3]	27 1B	功能码=271B(10011)	
Buffer[4]	变量类型	1/2/3/4/5/6/7/8/9, 11/12/13/17, 21/22/23	
Buffer[5]	状态代码	00:ok, xx:error	
Buffer[6][7][8]	实际数量	0-500000	高字节前, 低字节后
varName1	08 [index] 'P1_UA'	变量名	
varName2	08 [index] 'P1_UB'		
varName3	08 [index] 'P1_UC'		
...	Length [index] 'varname'		
00	结束		
或			
varName1	0F [index] 'P1_UA{0, 100}'	变量名+量程	
varName2	0F [index] 'P1_UB{0, 300}'		
...	length [index] 'varname{min, max}'		
或			
varName1	0F [index] 'P1_UA{0, 10} {F, 0, 2, 8, 9}'	变量名+量程+报警设定	
varName2	12 [index] 'P1_UB{0, 10} {T, 0, 2, 8, 9}'		
...	length [index] 'varname{range} {alarm}'		
00	结束		

[2]. 通过变量索引, 批量读取某段时间内发生变化的变量值

客户端发出请求(16 进制)				
例如:3E 2A 27 1C 11 00 00 64 00 00 02 00 05				
Buffer[0][1]	3E 2A	请求标识		
Buffer[2][3]	27 1C	功能码(271C) (10012)		
Buffer[4]	变量类型	模拟(R8): AI=01, AO=02, AR=03, VA=07 模拟(R4): AI=11, AO=12, AR=13, VA=17 开关(I1): DI=04, DO=05, DR=06, VD=08 文本(ST): VT=09 处于报警状态变量: 0x50+XX		
Buffer[5][6][7]	开始索引	0-500000		高字节前 低字节后
Buffer[8][9][10]	变量数量	0:全部, 1-500000 个		
Buffer[11][12]	秒时间段	0:全部, 1-3600 秒		
服务器返回应答(16 进制)				
例如:3C 2A 27 1C 11 00 00 10 00 00 64 00 xx xx xx xx 00 00 65 00 xx xx xx xx 00 00				
Buffer[0][1]	3C 2A	应答标识		
Buffer[2][3]	27 1C	功能码(271C) (10012)		
Buffer[4]	变量类型	01/02/03/04/05/06/07/08/09, 11/12/13/17		
Buffer[5]	状态代码	00:ok, xx:error		
Buffer[0][1]	数据包长度	1-8192	高字节前, 低字节后	
varValue11	模拟(R8): [index] Qty(0/1) xx xx xx xx xx xx xx xx		第 1 包数据	
varValue12	模拟(R4): [index] Qty(0/1) xx xx xx xx			
varValue13	开关(I1): [index] Qx (0x/8x)			
varValue14	文本(ST): [index] Length(0-240) 'xxxxxx'			
.....	[index]: 变量索引编号, 3 字节, 高字节前, 低字节后			
Buffer[0][1]	数据包长度	1-8192	高字节前, 低字节后	
varValue21	变量内容		第 2 包数据	
varValue22				
.....				
Buffer[0][1]	数据包长度	1-8192	高字节前, 低字节后	
varValue31	变量内容		第 3 包数据	
varValue32				
.....				
.....				第 N 包数据
00 00	结束			

[3]. 通过变量索引, 选择读取某些变量值

客户端发出请求 (16 进制), 例如: <u>3E 2A 27 1E 13 00 00 00 06 00 00 01 00 00 02</u>				
Buffer[0][1]	3E 2A	请求标识		
Buffer[2][3]	27 1E	功能码 (271D) (10014)		
Buffer[4]	变量类型	模拟 (R8): AI=01, AO=02, AR=03, VA=07 模拟 (R4): AI=11, AO=12, AR=13, VA=17, 开关 (I1): DI=04, DO=05, DR=06, VD=08, 文本 (ST): VT=09 处于报警状态变量: 0x50+XX		
Buffer[5][6]	秒时间段	0: 全部, 1-3600 秒, 3601: 处于报警状态		
Buffer[7][8]	索引字节数	0-8192	高字节前, 低字节后	多个索引编号
varIndex1	最多 2730 个变量索引编号 每变量索引编号, 3 字节, 高字节前, 低字节后			
varIndex2				
...				
服务器返回应答 (16 进制), 例如: <u>3C 2A 27 1E 13 00 00 10 00 00 01 00 00 00 00 00 00 00 02 00 00 00 00 00 00 00</u>				
Buffer[0][1]	3C 2A	应答标识		
Buffer[2][3]	27 1E	功能码 (271E) (10014)		
Buffer[4]	变量类型	01/02/03/04/05/06/07/08/09, 11/12/13/17		
Buffer[5]	状态代码	00:ok, xx:error		
Buffer[0][1]	数据包长度	1-8192	高字节前, 低字节后	第 1 包数据
varValue11	模拟 (R8): [index] Qty(0/1) xx xx xx xx xx xx xx xx			
varValue12	模拟 (R4): [index] Qty(0/1) xx xx xx xx			
varValue13	开关 (I1): [index] Qx (0x/8x)			
varValue14	文本 (ST): [index] Length(0-240) 'xxxxxx'			
.....	[index] : 变量索引编号, 3 字节, 高字节前, 低字节后			
Buffer[0][1]	数据包长度	1-8192	高字节前, 低字节后	第 2 包数据
varValue21				
varValue22				
.....				
Buffer[0][1]	数据包长度	1-8192	高字节前, 低字节后	第 3 包数据
varValue31				
varValue32				
.....				
.....				第 N 包数据
00 00	结束			

[4]. 通过变量索引, 批量修改变量值

客户端发出请求(16 进制)			
例如: <u>3E 2A 27 1D 13 00 0E 00 00 0A</u> <u>xx xx xx xx</u> <u>00 00 64</u> <u>xx xx xx xx</u>			
Buffer[0][1]	3E 2A	请求标识	
Buffer[2][3]	27 1D	功能码(271D) (10013)	
Buffer[4]	变量类型	模拟(R8): A0=02, AR=03, VA=07, 模拟(R4): A0=12, AR=13, VA=17, 开关(I1): D0=05, DR=06, VD=08, 文本(ST): VT=09	
Buffer[5][6]	数据包长度	1-8192	高字节前, 低字节后
varValue1	模拟(R8): [index] xx xx xx xx xx xx xx xx		数据包
varValue2	模拟(R4): [index] xx xx xx xx		
varValue3	开关(I1): [index] xx		
varValue4	文本(ST): [index] length(0-240) 'xxxxxx'		
...	[index] : 变量索引编号, 3 字节, 高字节前, 低字节后		
服务器返回应答(16 进制)			
例如: <u>3C 2A 27 1D 13 00 00 01 00 00</u>			
Buffer[0][1]	3C 2A	应答标识	
Buffer[2][3]	27 1D	功能码(271D) (10013)	
Buffer[4]	变量类型	02/03/05/06/07/08/09, 12/13/17	
Buffer[5]	状态代码	00:ok, xx:error	
Buffer[6][7]	成功计数	高字节前, 低字节后	
Buffer[8][9]	失败计数		

[5]. 得到某批变量名称对应索引

客户端发出请求(16 进制), 例如(' VA1 VA2 VA3 '):		
<u>3E 2A 27 1F 07 00 0D 7C 56 41 31 7C 56 41 32 7C 56 41 33 7C</u>		
Buffer[0][1]	3E 2A	请求标识
Buffer[2][3]	27 1F	功能码(271F) (10015)
Buffer[4]	变量类型	AI=1, AO=2, AR=3, VA=7, DI=4, DO=5, DR=6, VD=8, VT=9 0=混合变量, 变量名包含类型, 格式: 'XX.YYYYYY'
Buffer[5][6]	数据包长度	最大 8192 字节, 高字节前, 低字节后
varNames	' '	多变量名, ' ' 分隔
服务器返回应答(16 进制), 例如:		
<u>3C 2A 27 1F 07 00 09 00 00 01 00 00 02 00 00 03</u>		
Buffer[0][1]	3C 2A	应答标识
Buffer[2][3]	27 1F	功能码(271F) (10015)
Buffer[4]	变量类型	AI=1, AO=2, AR=3, VA=7, DI=4, DO=5, DR=6, VD=8, VT=9, 0 混合
Buffer[5]	状态代码	00:ok, xx:error
Buffer[6][7]	数据包长度	最大 8192 字节, 高字节前, 低字节后
varIndex1	xx xx xx	变量索引 3 字节整数, 高字节前, 低字节后 变量名不存在, 返回索引(FF FF FF)
varIndex2	xx xx xx	
varIndex3	xx xx xx	
...	...	

[6]. 错误代码:

00	无错误
01	变量类型不正确
02	变量实际数量零
03	变量开始索引错误
04	读取变量数量错误
05	禁止修改
06	修改变量全部失败
07	修改变量部分失败
08	批量变量名称格式错误
FE	加密狗授权不支持此协议
FF	访问运行数据库失败

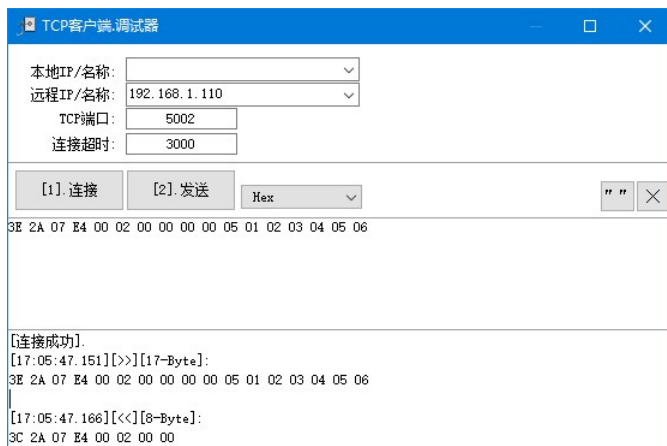
□ 驱动虚拟设备号

安装虚拟驱动(NULLDRV), 设备表中建立虚拟设备号;

客户端连接网络服务器, 通过命令(2020)驱动设备号, 读写并控制设备号;

客户端发送请求			
例(向 D2 设备号写入数据): <u>3E</u> <u>2A</u> <u>07</u> <u>E4</u> <u>00</u> <u>02</u> <u>00</u> <u>00</u> <u>00</u> <u>05</u> <u>xx</u> <u>xx</u> <u>xx</u> <u>xx</u> <u>xx</u>			
例(使 D2 设备号数据无效): <u>3E</u> <u>2A</u> <u>07</u> <u>E4</u> <u>00</u> <u>02</u> <u>01</u>			
例(查询 D2 设备号状态): <u>3E</u> <u>2A</u> <u>07</u> <u>E4</u> <u>00</u> <u>02</u> <u>03</u>			
Buffer[0][1]	3E 2A	发送标识	
Buffer[2][3]	07 E4(2020)	功能码=2020(07E4)	
Buffer[4][5]	设备号	2-2000	
Buffer[6]	操作:0(正常, 设置数据), 1(错误, 数据无效), 2(错误, 数据保持), 3(查询)		
Buffer[7][8]	开始字节单元	0-1023	状态 0 有效
Buffer[9][10]	结束字节单元	0-1023	
Buffer[11]...	数据内容	xx xx xx ...	
服务器应答			
例(D2 设备号正常未被需改): <u>3C</u> <u>2A</u> <u>07</u> <u>E4</u> <u>00</u> <u>02</u> <u>00</u> <u>00</u>			
例(D2 设备号错误数据无效): <u>3C</u> <u>2A</u> <u>07</u> <u>E4</u> <u>00</u> <u>02</u> <u>01</u> <u>00</u>			
例(D2 设备号数据被修改): <u>3C</u> <u>2A</u> <u>07</u> <u>E4</u> <u>00</u> <u>02</u> <u>00</u> <u>01</u> <u>00</u> <u>02</u> <u>00</u> <u>04</u> <u>01</u> <u>02</u> <u>03</u> <u>FF</u> <u>01</u> <u>FF</u>			
Buffer[0][1]	3C 2A	接收标识	
Buffer[2][3]	07 E4(2020)	功能码=2020(07E4)	
Buffer[4][5]	设备号	2-2000	
Buffer[6]	状态:0(正常), 1(错误, 数据无效), 2(错误, 数据保持)		
Buffer[7]	是否被修改:0=no, 1=yes		
Buffer[8][9]	被修改开始字节单元	0-1023	数据被修改有效
Buffer[10][11]	被修改结束字节单元	0-1023	
Buffer[12]...[n]	被修改数据内容	xx xx xx ...	
Buffer[n+1]...[n+m]	被修改数据位标志,	FF 01 00 03 ...	

通过 TCP 调试器测试:



□ 读取数据库记录包

最多支持 4 个数据库记录包, 记录内容须转换至字符串格式, 字段值用 ' | ' 分割, 最大长度 2048 字节, 例如: " | 2011-10-1 0:0:0 | 123 | 45.7 | 0 | 1 | abcd | "

如果客户端读取记录包内容进行实时显示, 而不存储, 则记录包格式自行定义 (16 进制格式);

客户端请求读取记录包		
例如: <u>3E 2A 07 D8 01 04 00</u> (读取第 1 个记录包)		
Buffer[0][1]	3E 2A	发送标识
Buffer[2][3]	07 D8 (2008)	功能码=2008 (07D8)
Buffer[4]	记录包编号	1-4
Buffer[5]	字节长度高字节	1-1023: 自定义记录包格式 1024: 字符串记录包格式
Buffer[6]	字节长度低字节	
服务器收到请求		
如果未准备好记录包, 则返回空记录, 例如: <u>3C 2A 07 D8 01 00 00</u>		
如果已准备好记录包, 则返回记录包:		
例如: <u>3C 2A 07 D8 01 00 24</u> " 2011-10-1 0:0:0 123 456.7 0 1 abcd "		
Buffer[0][1]	3C 2A	接收标识
Buffer[2][3]	07 D8 (2008)	功能码=2008 (07D8)
Buffer[4]	记录包编号	1-4
Buffer[5]	字节长度高字节	记录包实际长度 (1-2048)
Buffer[6]	字节长度低字节	
Buffer[7-n]	记录包内容	字符串或自定义格式
客户端收到记录包, 处理完成, 向服务器发送确认:		
例如: <u>3E 2A 07 D8 81 00 00</u>		
Buffer[0][1]	3E 2A	发送标识
Buffer[2][3]	07 D8 (2008)	功能码=2008 (07D8)
Buffer[4]	81-84, 91-94	处理成功 (81-84), 处理失败 (91-94)
Buffer[5][6]	xx xx	错误代码
服务器收到确认, 进行应答		
Buffer[0][1]	3C 2A	接收标识
Buffer[2][3]	07 D8 (2008)	功能码=2008 (07D8)
Buffer[4]	xx	自定义
Buffer[5][6]	xx xx	自定义
服务器应答后, 对数据库记录进行处理, 清除发送标志或删除已发送记录		

□ 执行字符串格式函数命令

客户端发出请求, 例如: <u>3E 2A 07 E2 00 00 1E</u> <u>ReadVarValues("AI.AI1 VA.VA1")</u>		
Buffer[0][1]	3E 2A	发送标识
Buffer[2][3]	07 E2(2018)	功能码=07E2(2018)
Buffer[4][5][6]	xx xx xx	命令长度(最大 8M), 高字节前, 低字节后
Buffer[7]..[n]	"... .."	命令内容, 字符串格式, 最大长度 8M
服务器返回, 例如: <u>3C 2A 07 E2 00 00 0B 123.4</u> 567.8		
Buffer[0][1]	3C 2A	接收标识
Buffer[2][3]	07 E2(2018)	功能码=07E2((2018)
Buffer[4][5][6]	xx xx xx	返回长度, 高字节在前, 低字节在后
Buffer[7]..[n]	"... .."	返回内容, 字符串格式

支持以下函数命令, 举例说明:

1	读取某类全部变量名称, 返回内容包含变量索引和名称: ReadVarNames(varType), varType - 变量类型: AI/AO/AR/DI/DO/DR/VA/VD/VT	
	请求	ReadVarNames("AI")
	返回	ok(3): 0, AI1 1, AI2 2, AI3
2	读取某些变量值: varValues=ReadVarValues(varNames)	
	请求	ReadVarValues("AI.AI1 VA.VA1 DI.DI1 VT.VT1 DO.DO1 VD.VD1 AI.AI2")
		ReadVarValues("file:varfile.txt"), 文件默认服务器 MyFile 目录
		ReadVarValues("last:")
	返回	ok: 12.3 0.56 1 abcd (error) (none) {0}
3	修改某些变量值: retCodes=WriteVarValues(varNames, varValues)	
	请求	WriteVarValues("AR.AR1 VA.VA1 VT.VT1 AI.AI1", "12.3 0.5 abc 10")
		WriteVarValues("file:varfile.txt", "12.3 0.5 abc 10")
	返回	ok: T F (none) (error)
4	以数组方式读取某批变量: varValues=ReadVarArray(varArray, start, end)	
	请求	ReadVarArray("AI.AI%i", 1, 5)
	返回	ok: 100 200 {0} (none) 500
5	以数组方式修改某批变量: retCodes=WriteVarArray(varArray, start, end, varValues)	
	请求	WriteVarArray("AR.AR%i", 1, 3, "100 200 300")
	返回	ok: T F (none)
6	以索引方式读取某批变量: varValues=ReadVarBlock(varType, start, end)	
	请求	ReadVarBlock("AI", 0, 4)
	返回	ok: 10 20 {0} {0} 50
7	以索引方式修改某批变量: retCodes=WriteVarBlock(varType, start, end, varValues)	
	请求	WriteVarBlock("AR", 0, 4, "10 20 30 40 50")
	返回	ok: T T T F F

8	读取过滤变量值, 返回内容包含变量索引, 变量名称, 变量值, 时间戳: varValues=ReadFilterVarValues (varType, varVame, timeLength, timeStamp) varType - 变量类型: "AI/AO/AR/DI/DO/DR/VA/VD/VT" varName - 变量名称, 包含通配符(*)实现过滤, 如:"*", "P1_*", "*_V1" timeLength - 时间段长度(0-3600 秒), 0:读取全部, 1-3600:读取某段时间内发生变化值 timestamp - 时间戳, 0:不带时间戳, 1:带时间戳			
	请求	ReadFilterVarValues("AI", "P1_*", 0, 0)		
	返回	ok(1-300000): 10, P1_AI1, 10 20, P1_AI2, 20 21, P1_AI3, {0}		
	注册某监控对象作为通讯模版:retCode=RegisterMObject(obj1)			
9	请求	RegisterMObject("P1")		
	返回	ok		
	根据被注册监控对象读取其他监控对象变量:varValues=ReadMObject(obj1, obj2, mode)			
	请求	ReadMObject("P1", "P2", 0)	0 - 返回全部变量值	
	返回	ok: AI. P2_AI1, 10 AI. P2_AI2, 20 DI. P2_DI1, 0 DI. P2_DI2, 1		
	请求	ReadMObject("P1", "P2", 5)	5 - 返回 5 秒内变化变量值	
	返回	ok: AI. P2_AI1, 10 DI. P2_DI1, 0		
	请求	ReadMObjectWithTimeStamp("P1", "P2", 0-3600)	返回带时间戳变量值	
10	返回	ok: AI. P2_AI1, 10, 2015-01-01 03:00:00		
	用户登录:retCode=UserLogon(user, password)			
11	请求	UserLogon("user1", "123456")	返回	ok
	用户注销:retCode=UserLogoff()			
12	请求	UserLogoff()	返回	ok
	读取某设备号字节数值: byteValues=ReadDevnoBytes(devNo, startByte, endByte)			
	请求	ReadDevnoBytes(2, 0, 100)		
13	返回	ok: 1 0 255 0 ... 或 invalid: 0 0 0 0 ...		
	修改某设备号字节数值: retCode=WriteDevnoBytes(devNo, startByte, endByte, values)			
	请求	WriteDevnoBytes(2, 0, 10, "0 1 255 0 ...")		
	返回	ok:modifybytes=10		
14	驱动虚拟设备号: retContent=DriverVirtualDevno(devNo, start, end, values, status) status:0=无错误, 数据有效;1=有错误, 数据无效;2=有错误, 数据保持;3=查询			
	请求	16 进制	DriverVirtualDevno(2, 0, 3, "Hex:01 02 10 20", 0)	
		10 进制	DriverVirtualDevno(2, 0, 3, "Dec:123 100 200 255", 0)	
		字符串	DriverVirtualDevno(2, 0, 7, "Str:ABcd1234", 0)	
		变量值	DriverVirtualDevno(2, 0, 1023, "Var: 0(I1), 0(I2), 0(I4), 0(I8), 0(UI1), 0(UI2), 0(UI4), 0(R4), 0(R8)", 0)	
		有错误	DriverVirtualDevno(2, 0, 7, "", 1/2)	
	返回	无修改	ok:status=0/1/2;modify=no;	
		有修改	ok:status=0;modify=yes;from=1;to=2;hex:10 A0;flag:FF 01	

15	得到当前时间: curTime=GetCurrentTime()	
	请求	GetCurrentTime()
	返回	ok:2010-01-01 12:15:00
16	修改当前时间: curTime=SetCurrentTime("YYYY-mm-dd HH:MM:SS")	
	请求	SetCurrentTime("2010-01-01 12:15:00")
	返回	ok:2010-01-01 12:15:00
17	检测某些设备号是否被修改完成: result=CheckDeviceNosWriteCompleted(devnos, timeout, delay) devnos - 字符串表示的多个设备号, 逗号分隔或横线连接, 例如"2-10, 20, 30" timeout - 超时等待时间(秒) delay - 完成延时时间(秒) result - 返回完成数量和未完成数量, 未完成可能由通讯或超时导致	
	请求	CheckDeviceNosWriteCompleted("2-10, 15", 3, 0.5)
	返回	result:10, 0
18	检测某些设备号是否被读取完成: result=CheckDeviceNosReadCompleted(devnos, timeout, delay) devnos - 字符串表示的多个设备号, 逗号分隔或横线连接, 例如"2-10, 20, 30" timeout - 超时等待时间(秒) delay - 完成延时时间(秒) result - 返回完成数量和未完成数量, 未完成可能由通讯或超时导致	
	请求	CheckDeviceNosReadCompleted("2-10, 15", 3, 0.5)
	返回	result:10, 0
19	读取组态软件版本号: verInfo=GetSystemVersion()	
	请求	GetSystemVersion()
	返回	version:7.6.12.0
20	读取运行数据库各变量数量: varInfo=GetRunDBVarCount()	
	请求	GetRunDBVarCount()
	返回	varCount:AI(2), AO(0), AR(5), DI(3), DO(0), DR(6), VA(50), VD(30), VT(30)
21	获取组态系统启动时间及运行长度: startupTime=GetSystemStartupTime()	
	请求	GetSystemStartupTime()
	返回	startupTime:2015-05-01 12:35:00, timeSpan:10.01:00:50
22	读写服务器中文本文件, 缺省组态 Temp 目录, 须加后缀(txt, csv 等)	
	读服务器文本文件内容:fileContent=ReadTextFile(fileName)	
	请求	ReadTextFile("test.txt")
	返回	ok:ABCDabcd123 或 error:file.invalid
	写文本内容到服务器某文本文件:result=WriteTextFile(fileName, fileContent)	
	请求	WriteTextFile("test.txt", "ABCDabcd123")
	返回	ok:new.length.11 或 ok:append.length.11

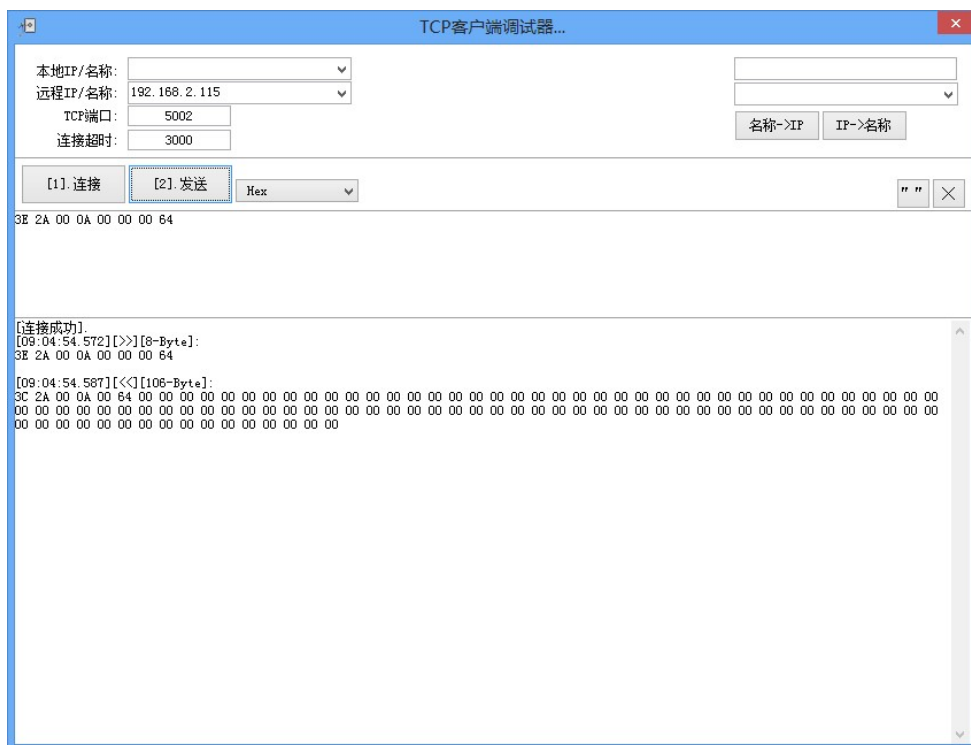
23	设置用户管理和操作级别:result=SetUserManagement(userInfo) UserInfo 内容包含最多 99 个用户信息,用回车换行(\r\n)分隔; 每用户内容为: 用户名,口令,拥有级别(111111....),缺省用户(Y/N);	
	请求	SetUserManagement("user1,1111,11111111,N\r\nuser2,2222,10101010,Y\r\n")
	应答	ok:0 或 error:100
24	设置操作级别: result=SetOperateLevel(levelInfo) LevelInfo 内容包含最多 980 个级别描述,用回车换行(\r\n)分隔;	
	请求	SetOperateLevel("级别 20\r\n 级别 21\r\n")
	应答	ok:0 或 error:100

编程示例 (VC++):

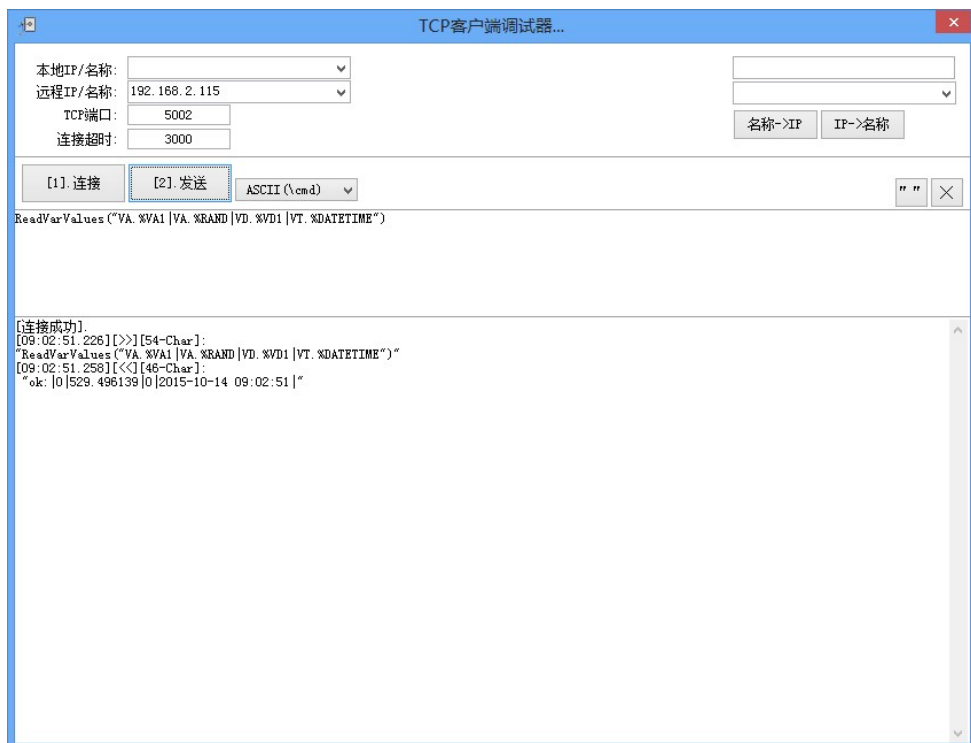
```
//字符串命令
CString s="ReadVarsValue(\"AI.AI1|VA.VA1\")";
DWORD dwValue=s.GetLength();
//报文缓冲区
BYTE Buffer[8192];
memset(Buffer,0,8192);
//报文头:3E 2A
Buffer[0]='>'; Buffer[1]='*';
//功能码(2018)
Buffer[2]=0x07; Buffer[3]=0xE2;
//字符串长度.3字节表示
Buffer[4]=dwValue/0x10000;
Buffer[5]=(dwValue&0xFF00)/0x100;
Buffer[6]=dwValue&0xFF;
//字符串内容
memcpy(&Buffer[7],s,dwValue);
//发送请求报文
cntSocket.SendData(Buffer,dwValue+7);
//接收报文头.并判断
memset(Buffer,0,8192);
int len=cntSocket.ReceiveData(Buffer,7);
if(len==7 && Buffer[0]=='<' && Buffer[1]=='*' && Buffer[2]==0x07 && Buffer[3]==0xE2){
    dwValue=Buffer[4]*0x10000+Buffer[5]*0x100+Buffer[6]; //计算返回字符串长度
    len=cntSocket.ReceiveData(Buffer[7],dwValue); //接收字符串内容
    if(len>dwValue){
        s=&Buffer[7]; //获取字符串
        //.....
    }
}
```

- 使用 TCP 调试器测试 FMTCP 协议:

例 1: 读取设备号 D10 从 B0 开始长度 100;



例 2: 执行字符串格式函数命令:



28.6 MODBUS-TCP 协议

- ☐ 客户端通过 MODBUS/TCP 协议, 访问网络服务器各种数据: 设备数据表、数据包、时钟;
- ☐ 通过 MODBUS 单元地址和功能码, 区分网络服务器不同数据类型或处理方式;

MODBUS/TCP 客户端			网络服务器			
单元	功能码	数据地址	数据类型	地址	数据单位	
寄存器/模入, 按字访问设备设备数据表 D1-100, 反高低字节, I2 (x2, x1)						
1	03, 04, 06, 10	400001-400512	设备号 D1	0-511	字	
2	03, 04, 06, 10	400001-400512	设备号 D2	0-511		
...		
100	03, 04, 06, 10	400001-400512	设备号 D100	0-511		
寄存器/模入, 按字访问设备设备数据表 D1-100, 不反高低字节, I2 (x1, x2)						
101	03, 04, 06, 10	400001-400512	设备号 D1	0-511	字	
102	03, 04, 06, 10	400001-400512	设备号 D2	0-511		
...		
200	03, 04, 06, 10	400001-400512	设备号 D100	0-511		
寄存器/模入, 按字访问模拟变量包和开关变量包, 反高低字节, I2 (x2, x1)						
201 [C9]	03, 04, 06, 10	400001-408192	模拟变量包 (AP)	0-4095	双字 (浮点)	
202 [CA]	03, 04, 06, 10	400001-400256	开关变量包 (DP)	0-0511	字节	
203 [CB]	03, 04, 06, 10	400001-404096	模拟变量包 (AP)	0-4095	字	*1
204 [CC]	03, 04, 06, 10	400001-404096	模拟变量包 (AP)	0-4095	字	*10
205 [CD]	03, 04, 06, 10	400001-404096	模拟变量包 (AP)	0-4095	字	*100
206 [CE]	03, 04, 06, 10	400001-404096	模拟变量包 (AP)	0-4095	字	*1000
寄存器/模入, 按字访问模拟变量包和开关变量包, 不反高低字节, I2 (x1, 2)						
211 [D3]	03, 04, 06, 10	400001-408192	模拟变量包 (AP)	0-4095	双字 (浮点)	
212 [D4]	03, 04, 06, 10	400001-400256	开关变量包 (DP)	0-0511	字节	
213 [D5]	03, 04, 06, 10	400001-404096	模拟变量包 (AP)	0-4095	字	*1
214 [D6]	03, 04, 06, 10	400001-404096	模拟变量包 (AP)	0-4095	字	*10
215 [D7]	03, 04, 06, 10	400001-404096	模拟变量包 (AP)	0-4095	字	*100
216 [D8]	03, 04, 06, 10	400001-404096	模拟变量包 (AP)	0-4095	字	*1000
寄存器/模入, 按双字访问模拟变量包, 反高低字节, R4 (x4, x3, x2, x1)						
221 [DD]	03, 04, 10	400001-408192	模拟变量包 (AP)	0-4095	双字 (浮点)	
继电器, 访问开关变量包, 开始地址须 m*8+1, 长度须 n*8						
1-255	01, 05, 0F	000001-004096	开关变量包 (DP)	0.0-511.7	位 (Bit)	
开关输入, 访问开关变量包, 开始地址须 m*8+1, 长度须 n*8						
1-255	02	100001-104096	开关变量包 (DP)	0.0-511.7	位 (Bit)	
寄存器/模入, 读取或设置服务器时钟						
222 [DE]	03, 04, 10	400001-400004	十进制时间	yyyy mm dd HH MM S*1000		
223 [DF]	03, 04, 10	400001-400004	BCD 码时间	yyyy mm dd HH MM SS 00		

寄存器/模入, 按字访问设备设备数据表 D1-2000				
缺省不反高低字节 I2(x1, x2), 允许设置反高低字节 I2(x2, x1)				
不支持跨设备号访问, m=设备号, n=字单元号;				
231[E7]	03, 04, 06, 10	400001-451200	D1-100	$(m-1)*512+n+1$
232[E8]	03, 04, 06, 10	400001-451200	D101-200	$(m-101)*512+n+1$
233[E9]	03, 04, 06, 10	400001-451200	D201-300	$(m-201)*512+n+1$
234[EA]	03, 04, 06, 10	400001-451200	D301-400	$(m-301)*512+n+1$
235[EB]	03, 04, 06, 10	400001-451200	D401-500	$(m-401)*512+n+1$
236[EC]	03, 04, 06, 10	400001-451200	D501-600	$(m-501)*512+n+1$
237[ED]	03, 04, 06, 10	400001-451200	D601-700	$(m-601)*512+n+1$
238[EE]	03, 04, 06, 10	400001-451200	D701-800	$(m-701)*512+n+1$
239[EF]	03, 04, 06, 10	400001-451200	D801-900	$(m-801)*512+n+1$
240[F0]	03, 04, 06, 10	400001-451200	D901-1000	$(m-901)*512+n+1$
241[F1]	03, 04, 06, 10	400001-451200	D1001-1100	$(m-1001)*512+n+1$
242[F2]	03, 04, 06, 10	400001-451200	D1101-1200	$(m-1101)*512+n+1$
243[F3]	03, 04, 06, 10	400001-451200	D1201-1300	$(m-1201)*512+n+1$
244[F4]	03, 04, 06, 10	400001-451200	D1301-1400	$(m-1301)*512+n+1$
245[F5]	03, 04, 06, 10	400001-451200	D1401-1500	$(m-1401)*512+n+1$
246[F6]	03, 04, 06, 10	400001-451200	D1501-1600	$(m-1501)*512+n+1$
247[F7]	03, 04, 06, 10	400001-451200	D1601-1700	$(m-1601)*512+n+1$
248[F8]	03, 04, 06, 10	400001-451200	D1701-1800	$(m-1701)*512+n+1$
249[F9]	03, 04, 06, 10	400001-451200	D1801-1900	$(m-1801)*512+n+1$
250[FA]	03, 04, 06, 10	400001-451200	D1901-2000	$(m-1901)*512+n+1$

□ 自定义 MODBUS/TCP 协议数据:

设置网络服务器(TCP连接,同时支持MBTCP与FMTC协议)...

服务器参数

本机IP地址: local

TCP端口号: 502

心跳间隔[秒]: 60

网络校时基准: 无

☐ 自定义MBTCP协议:

☐ 禁止客户端访问设备表:

☐ 客户端连接日志文件:

☐ 启动校时服务器, 端口: 37

确定

取消

授权客户端(最多32客户端及512连接数):

编号	客户端IP地址(0.0.0.0为动态IP)	允许修改
1	0.0.0.0	×
2	0.0.0.0	×
3	0.0.0.0	×
4	0.0.0.0	×
5	0.0.0.0	×
6	0.0.0.0	×
7	0.0.0.0	×
8	0.0.0.0	×
9	0.0.0.0	×
10	0.0.0.0	×

选择[自定义 MBTCP 协议]:

自定义MB/TCP协议...

默认单元地址及数据起始地址

AP/201:	201	AP/211:	211	AP/221:	221
DP/202:	202	DP/212:	212	ST/222:	222
AP/203:	203	AP/213:	213	ST/223:	223
AP/204:	204	AP/214:	214		
AP/205:	205	AP/215:	215		
AP/206:	206	AP/216:	216		

首地址[0/1]: 1

模拟变量包 (AP) 起始地址[4xxxxx]: 1

结束地址[4xxxxx]: 100

开关变量包 (DP) 起始地址[4xxxxx]: 1

结束地址[4xxxxx]: 100

开关变量包 (DP) 起始地址[1xxxxx]: 1

开关变量包 (DP) 起始地址[0xxxxx]: 1

系统时间 (ST) 起始地址[4xxxxx]: 1

☐ 根据单元地址区分访问变量包

模拟变量包 (AP) 偏移量: 32

开关变量包 (DP) 偏移量: 4

确定

取消

[1]. 只访问变量包而不访问设备数据表时, 变量包对应的默认单元地址允许被重新定义;
例如: 欲通过单元地址 201 访问模拟变量包, 而客户端设定的单元地址却只能为 1 时:

AP/201:	1	AP/211:	211	AP/221:	221
DP/202:	202	DP/212:	212	ST/222:	222
AP/203:	203	AP/213:	213	ST/223:	223
AP/204:	204	AP/214:	214		
AP/205:	205	AP/215:	215		
AP/206:	206	AP/216:	216		

[2]. 允许改变变量包对应 MODBUS 的起始地址，默认起始地址为 1；

例如：客户端需要通过寄存器 400100 访问模拟变量包 AP[0]，而不是 400001：

模拟变量包 (AP) 起始地址 [4xxxxx]:	100
结束地址 [4xxxxx]:	200

[3]. 根据单元地址分区访问变量包

☒ 根据单元地址分区访问变量包

模拟变量包 (AP) 偏移量:	10
开关变量包 (DP) 偏移量:	2

需求示意图表：

单元地址	Modbus开始地址		模拟变量包 (AP)	开关变量包 (DP)
1	400001	100001	0, 1, 2, ..., 9	0, 1
2	400001	100001	10, 11, 12, ..., 19	2, 3
3	400001	100001	20, 21, 22, ..., 29	4, 5
4	400001	100001	30, 31, 32, ..., 39	6, 7
...
N	400001	100001	(N-1)*A, ..., N*A-1	(N-1)*B, ..., N*B-1

N=单元地址, A=模拟包区域偏移量, B=开关包区域偏移量

实际定义变量包为：

定义通讯变量包[AP/DP]...

地址	AP - 模拟变量包:	数值[R4]	地址	DP - 开关变量包:	状态[位]
0	VA. %VA1		0.0	VD. %VD1	
1	VA. %VA2		0.1	VD. %VD2	
2			0.2	VD. %VD3	
3	单元1		0.3	VD. %VD4	
4			0.4		
5			0.5	单元	
6			0.6		
7			0.7		
8			1.0		
9			1.1		
10	VA. %VA3		1.2		
11	VA. %VA4		1.3		
12			1.4		
13	单元2		1.5		
14			1.6		
15			1.7		
16			2.0	VD. %VD5	
17			2.1	VD. %VD6	
18			2.2	VD. %VD7	
19			2.3	VD. %VD8	
20			2.4	单元2	
21			2.5		
22			2.6		
23			2.7		

Excel 工具

保存 取消

模拟变量包的数值是浮点数；

缺省格式为按字反高低字节, R4 (b2, b1, b4, b3)；

如果需要不需要反高低字节, R4 (b1, b2, b3, b4) 格式, 则设置AP/211=255:

自定义MB/TCP协议...

默认单元地址及数据起始地址

AP/201:	201	AP/211:	255	AP/221:	221
DP/202:	202	DP/212:	212	ST/222:	222
AP/203:	203	AP/213:	213	ST/223:	223
AP/204:	204	AP/214:	214		
AP/205:	205	AP/215:	215		
AP/206:	206	AP/216:	216		
DP/207:	207				
DP/208:	208				

首地址 [0/1]: 1

模拟变量包 (AP) 起始地址 [4xxxxx]: 1

结束地址 [4xxxxx]: 100

开关变量包 (DP) 起始地址 [4xxxxx]: 1

结束地址 [4xxxxx]: 100

开关变量包 (DP) 起始地址 [1xxxxx]: 1

开关变量包 (DP) 起始地址 [0xxxxx]: 1

系统时间 (ST) 起始地址 [4xxxxx]: 1

☒ 根据单元地址分区访问变量包

模拟变量包 (AP) 偏移量: 10

开关变量包 (DP) 偏移量: 2

确定 取消

如果需要按双字反高低字节, R4 (b4, b3, b2, b1), 则设置AP/221=255:

自定义MB/TCP协议...

默认单元地址及数据起始地址

AP/201:	201	AP/211:	211	AP/221:	255
DP/202:	202	DP/212:	212	ST/222:	222
AP/203:	203	AP/213:	213	ST/223:	223
AP/204:	204	AP/214:	214		
AP/205:	205	AP/215:	215		
AP/206:	206	AP/216:	216		

首地址 [0/1]: 1

模拟变量包 (AP) 起始地址 [4xxxxx]: 100

结束地址 [4xxxxx]: 100

开关变量包 (DP) 起始地址 [4xxxxx]: 1

结束地址 [4xxxxx]: 100

开关变量包 (DP) 起始地址 [1xxxxx]: 1

开关变量包 (DP) 起始地址 [0xxxxx]: 1

系统时间 (ST) 起始地址 [4xxxxx]: 1

☒ 根据单元地址分区访问变量包

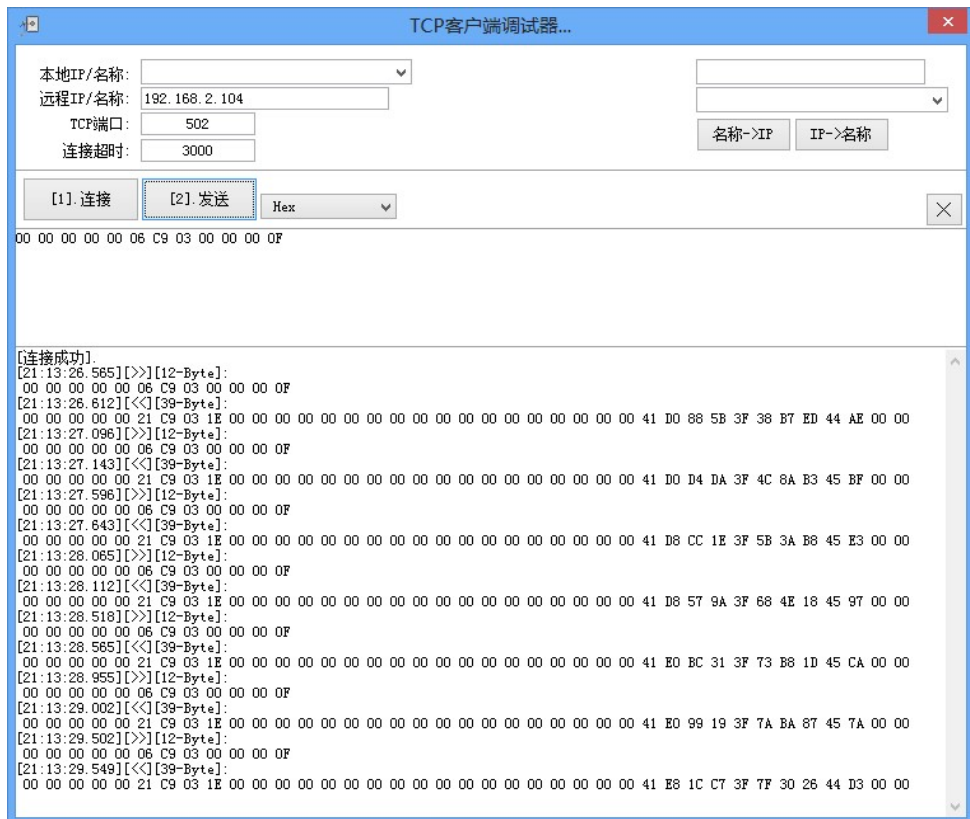
模拟变量包 (AP) 偏移量: 10

开关变量包 (DP) 偏移量: 2

确定 取消

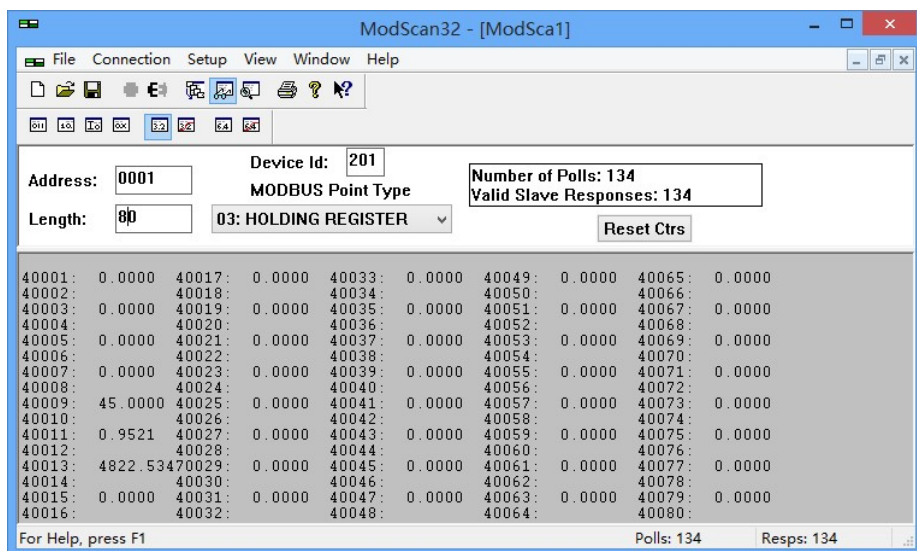
- 使用 TCP 调试测试 MODBUS/TCP 协议：

例如：通过 201 单元地址和功能码 03, 读取 10 个寄存器(5 个模拟变量)；



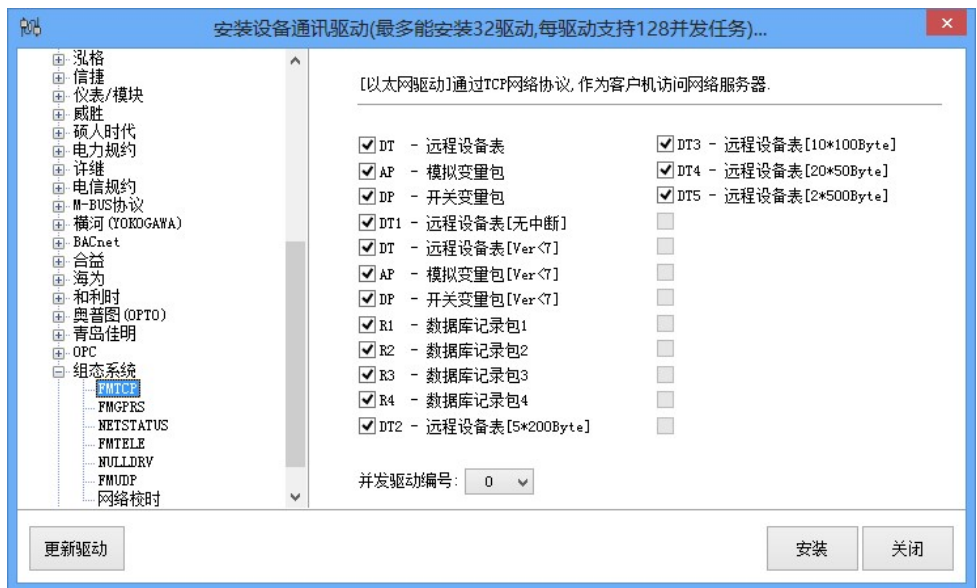
- 使用 MODBUS 调试器 (ModScan) 测试网络服务器：

例如：通过 201 单元地址和功能码 03, 读取 80 个寄存器(40 个模拟变量)；



28.7 FMTCP 驱动实现 C/S 架构

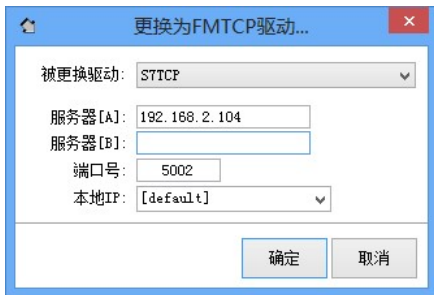
- 将组态完成的服务器项目拷贝到客户机,并调入项目;
- 服务器端定义并启动网络服务器,不需要定义变量包;
- 选择“设备通讯”功能,执行“1. 安装驱动”任务,安装 FMTCP 驱动:



- 选择“设备通讯”功能,执行“设备数据表”任务:



- 设备数据表中执行鼠标右键,选择并执行“转换为 FMTCP”命令:



- 选择需转换驱动, 填写服务器[A/B]IP、端口号、本地 IP, 执行[确定];
设备数据表驱动被转换为 FMTCP 驱动, 数据开始地址总为 0, 长度不变:

设备号	名称	通讯驱动	本地参数	远程参数	数据类型	开始地址	长度
D1	System					0	1024
D2	'D2设备号'	FMTCP	[default]	192.168.2.105,5002	DT - 远程设备表	2	0
D3	'D3设备号'	FMTCP	[default]	192.168.2.105,5002	DT - 远程设备表	3	0
D4	'D4设备号'	FMTCP	[default]	192.168.2.105,5002	DT - 远程设备表	4	0
D5	'D5设备号'	FMTCP	[default]	192.168.2.105,5002	DT - 远程设备表	5	0
D6	'D6设备号'	FMTCP	[default]	192.168.2.105,5002	DT - 远程设备表	6	0
D7							
D8							
D9							
D10							
D11							
D12							
D13							
D14							
D15							
D16							

- 双击某设备号手动修改:

设备号名称:
'D2设备号'

FMTCP

[以太网驱动]通过TCP网络协议,作为客户机访问网络服务器.

[1]. 远程参数

端口号: 5002

服务器IP地址[A]: 192.168.2.105

服务器IP地址[B]:

通讯超时[MS]: 1000

重试次数: 3

扫描级别[1-100]: 1

☒ 动态扫描级别:

[2]. 本地参数

本机IP地址[1/2]: [default]

设置网卡...

[3]. 通讯数据

数据类型: DT - 远程设备表

访问方式: 读写[RW]

单元格式: 字节[8位] 无符号整数

设备号(01-2000): 2

开始地址: 0 [00]

长度: 100

增强选项:

☐ 无中断标志;
☐ 中断时数据保持;
☒ 尽快恢复通讯;
☐ 设备号变化计数标志(B1038);
☐ 报文日志文件(部分驱动);
☐ 读受D1相应单元控制[0/1/2/...15];

确认

取消

□ 客户端不启动网络服务器功能, 并删除 FMTCP 外的其他驱动程序;

28-39

28.8 数据库记录包

- 假设远程计算机 SQL Server 有 UserDatabase 数据库, 其中有 demo 数据表:

字段名称:	dt	f1	f2	f3
字段类型:	datetime	real	real	Tinyint, 缺省值 1

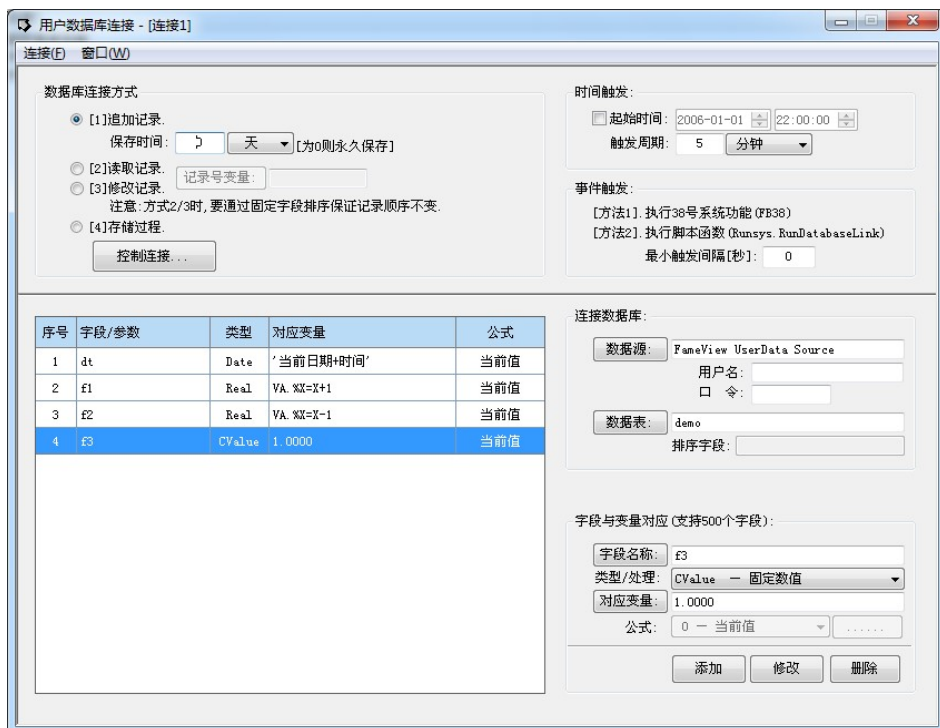
- 通过网络服务器, 查询 demo 表中 f3=1 的记录作为记录包:

客户端依次读取记录包内容, 读取记录成功后使 f3=0;

如果未定义标识字段 f3, 读取记录成功后删除当前记录;

- 服务器端(远程计算机)

[1]. 通过数据库连接向 demo 表追加记录, 建议最小存储间隔 30 秒:



[2]. 应用数据库连接;

[3]. 组态系统 MyFile 目录下建立记录包配置文件(fmDbrPackage.txt), 最多定义 4 个记录包:



记录包内容包括 demo 表中 dt(自动作为打包首字段)、f1、f2 字段组成, f3 是标志字段;

字段类型支持 UI1/I1/UI2/I2/UI4/I4/UI8/I8/R4/R8/BOOL/STR16/STR32/STR64/STRXX/DATE;

记录包报文格式: " | 2011-10-1 0:0:0 | 123 | 456.7 | "

[4]. 缺省方式组态网络服务器:

TCP网络服务器(同时支持MBTCP与FMTCPP协议)

服务器参数

本机IP地址: local

TCP端口号: 5002

心跳间隔[秒]: 60

网络校时基准: 无

同步协议超时[秒]: 5

☐ 自定义MBTCP协议

☐ 禁止访问设备数据表

☐ 启动校时服务器, 端口: 37

☐ 客户端通讯日志文件

限制设备号

授权客户端 (最多32客户端及512连接数):

编号	客户端IP地址 (0.0.0.0为动态IP)	允许修改
1	0.0.0.0	×
2	0.0.0.0	×
3	0.0.0.0	×
4	0.0.0.0	×
5	0.0.0.0	×
6	0.0.0.0	×
7	0.0.0.0	×
8	0.0.0.0	×
9	0.0.0.0	×
10	0.0.0.0	×

[5]. 启动网络服务器和数据库连接:

启动任务

系统启动选项

☐ 操作系统启动, 自动运行监控系统:
☐ 屏蔽桌面 (项目交付时选用)
 自启动延时 [5~300] 秒: 5

☐ 安静方式启动 (不显示启动过程窗口):

☐ 运行系统退出时, 提示项目备份:

☐ 禁止进入休眠状态, 并屏蔽POWER按键:

☐ 建立启动标识 (UDP5688, 用来检测系统是否启动):

☐ 系统启动命令:

☐ 输入命令错误, 注销操作系统:

☐ 系统退出命令:

☐ 系统启动窗口位置坐标 (x, y):
 0 0

☐ 记录系统事件:

启动系统任务:

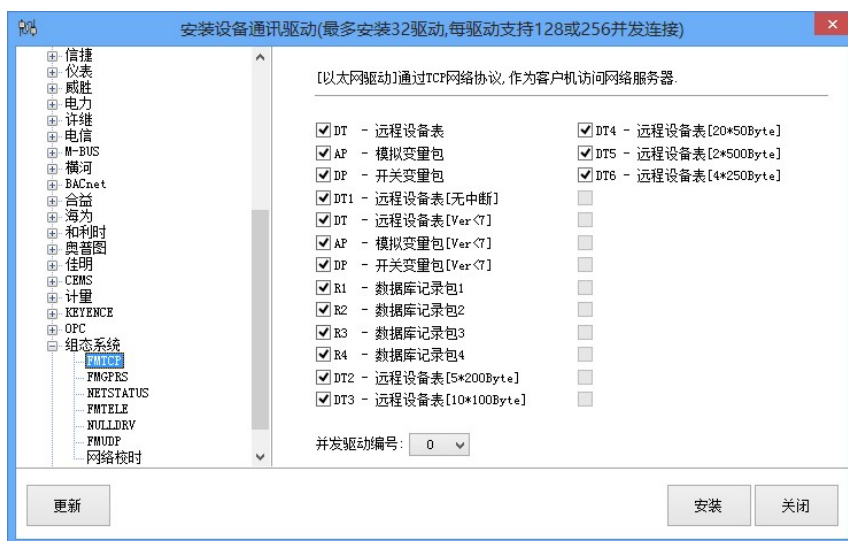
<input type="checkbox"/> 01. 画面显示	<input type="checkbox"/> 14. ADSI数据服务
<input type="checkbox"/> 02. 变量报警	<input type="checkbox"/> 15. 短信数据服务
<input type="checkbox"/> 03. 实时报表	<input type="checkbox"/> 16. 远程连接服务
<input type="checkbox"/> 04. 配方应用	<input type="checkbox"/> 17. OPC服务器
<input type="checkbox"/> 05. 用户管理	<input type="checkbox"/> 18. 变量组
<input type="checkbox"/> 06. 双机冗余	<input type="checkbox"/> 19. CDT转发服务
<input type="checkbox"/> 07. 全局变量	<input type="checkbox"/> 20. CEMS国标服务
<input checked="" type="checkbox"/> 08. 数据库连接	<input type="checkbox"/> 21. IEC104转发服务
<input type="checkbox"/> 09. 批量数据库连接	<input type="checkbox"/> 22. 数据库转发
<input type="checkbox"/> 10. 全局脚本	<input type="checkbox"/> 23. Web服务器
<input checked="" type="checkbox"/> 11. 网络服务器	<input type="checkbox"/> 24. SNMP代理服务
<input type="checkbox"/> 12. 串口服务器	<input type="checkbox"/> 25. 实时数据库连接
<input type="checkbox"/> 13. BACNET/IP服务	<input type="checkbox"/> 26. 佳华云系统

启动附加程序 (1-20):

确定 **取消**

□ 客户端(中心计算机)

[1]. 安装 FMTCP 驱动:



[2]. 使用 FMTCP 驱动读取记录包, 假设设备号 D3, 长度必须 1024:

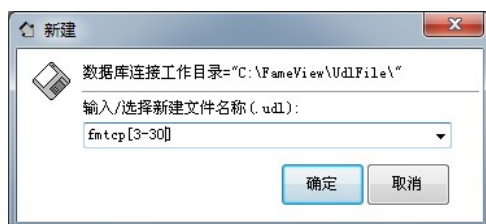


自动触发名称为 FMTCP[m-n] 或 FMTCP[n] 的数据库连接 ($m \leq 3 \leq n$); 如: "FMTCP[2-50], FMTCP[3]";

读取记录包的设备号与读取实时数据的设备号在同一个任务中时, 通讯速度会较慢;

建议为读取记录包的设备号, 单独开设并发通讯任务, 即在服务器 IP 后加", #N"

[3]. 建立数据库连接, 文件名为 FMTCP[3-30] 或 FMTCP[3], 则能够被 D3-D30 设备号触发执行;



[4]. 字段列表顺序须与记录包字段顺序一致, 时间字段须是首字段, 触发周期总是 0:

用户数据库连接 - [FMTCP[3-30]]

连接(F) 窗口(W)

数据库连接方式

- ☒ [1]追加记录:
保存时间: 0 天 [为0则永久保存]
- ☐ [2]读取记录:
记录号变量:
- ☐ [3]修改记录:
注意: 方式2/3时, 要通过固定字段排序保证记录顺序不变.
- ☐ [4]存储过程:

控制连接...

时间触发:

☐ 起始时间: 2006-01-01 22:00:00
触发周期: 0 分钟

事件触发:

[方法1]. 执行38号系统功能 (FB38)
[方法2]. 执行脚本函数 (Runsys.RunDatabaseLink)
最小触发间隔[秒]: 0

序号	字段/参数	类型	对应变量	公式
1	dt	Date	'当前日期+时间'	当前值
2	f1	Real	VA. %VA1	当前值
3	f2	Real	VA. %VA1	当前值

连接数据库:

数据源: FameView UserData Source
用户名:
口令:
数据表: demo
排序字段:

字段与变量对应 (支持500个字段):

字段名称: f2
类型/处理: Real — 实数/单精度
对应变量: VA. %VA1
公式: 0 — 当前值 参数...
添加 修改 删除

[5]. 应用数据库连接 FMTCP[3-30]:

[6]. 启动数据库连接:

启动任务...

系统启动选项

- ☐ 操作系统启动后, 自动运行监控系统:
☐ 屏蔽Windows桌面 (建议项目交付时再选用)
自启动延时时间 (5-3600秒): 5
- ☐ 安静方式启动 (不显示启动过程窗口):
☐ 运行系统退出时, 提示项目备份:
☐ 禁止进入休眠状态, 并屏蔽POWER按键 (Win7/Vista不支持):
☐ 建立启动标识 (UDP5888, 用来检测系统是否启动):
- ☐ 系统启动口令:
☐ 输入口令错误, 则注销操作系统:
☐ 系统退出口令:
☐ 系统启动窗口位置坐标 (x, y): 0 0

启动系统任务:

- ☐ 01. 画面显示
- ☐ 02. 变量报警
- ☐ 03. 实时报表
- ☐ 04. 配方应用
- ☐ 05. 用户管理
- ☐ 06. 双机冗余
- ☐ 07. 全局变量
- ☒ 08. 数据库连接
- ☐ 09. 批量数据库连接
- ☐ 10. 全局脚本
- ☐ 11. 网络服务器
- ☐ 12. 串口服务器
- ☐ 13. BACNET/IP 服务
- ☐ 14. AUST数据服务
- ☐ 15. 短信数据服务
- ☐ 16. 远程连接服务
- ☐ 17. OPC服务器
- ☐ 18. 变量组
- ☐ 19. CDT转发服务
- ☐ 20. CEMS国标服务
- ☐ 21. IEC104转发服务
- ☐ 22. 数据库转发
- ☐ 23. Web服务器
- ☐ 24. SNMP代理服务
- ☐ 25. 实时数据库连接
- ☐ 26. 佳华云系统

启动附加程序 (最多20个):

确定 取消