



深圳微精艺电子有限公司

www.vkinging.com

Product:VK70xN 数据采集器

Revision: 6.07

Issued Date: 2022-10-21

DLL 函数使用手册 (自带 TCP 服务器解析多卡连接数据)

Sheet 1 of 68

VK70xNMC 数据采集器 DLL 函数使用说明书

 深圳微精艺电子有限公司 www.vkinging.com	Product:VK70xN 数据采集器 Revision: 6.07 Issued Date: 2022-10-21
DLL 函数使用手册 (自带 TCP 服务器解析多卡连接数据)	Sheet 2 of 68

目 录

.....	3
1. DLL 函数说明 (自带 TCP 服务器解析多卡连接数据)	4
2. DLL 版本信息	6
2.1 获取 DLL 函数的版本信息	6
3. TCP 服务器函数说明	6
3.1 打开服务器	6
3.2 关闭服务器	7
3.3 读取已打开服务器的端口号	8
3.4 读取已连接服务器上的采集卡数量	8
3.5 读取当前采集卡的句柄和 IP 地址	9
3.6 读取当前服务器端接收总字节数	10
4. ADC 数据采集函数说明	12
4.1 初始化已连接的采集卡参数 (适用于 VK701N, VK702N)	12
4.2 初始化已连接的采集卡参数 2 (适用于 VK701N, VK701NSD, VK702N)	13
4.3 开始连续采样	15
4.4 开始 N 点采样	15
4.5 停止采样	16
4.6 读取单通道数据	17
4.7 读取 4 通道采样数据 (适用于 VK701S, VK701N, VK702N, VK701NSD)	19
4.8 读取 8 通道采样数据 (适用于 VK702N)	20
4.9 停止当前采集卡采样	21
4.10 读取 IO2 和 IO3 状态 (适用于 VK701N, VK701NSD 和 VK702N 的 24 位采样模式)	22
4.11 读取带 IO 状态的单通道数据 (适用于 VK701N, VK701NSD 和 VK702N)	23
4.12 读取带 IO 状态的 4 通道采样数据 (适用于 VK701N, VK701NSD 和 VK702N)	25
4.13 读取带 IO 状态的 8 通道采样数据 (适用于 VK702N)	27
4.14 其它说明	29
5. 附加功能函数说明	32
5.1 设置 PWM/DAC/IO/计数器/温度通道	32
5.2 读取 IO 状态	34
5.2.1 读取 PWM 参数	35
5.2.2 读取 DAC 参数	36
5.2.3 读取计数器	37
5.2.4 读取温度值	39
6. SD 卡文件系统函数说明	41
6.1 进入 SD 卡文件下载模式	41
6.2 退出 SD 卡文件下载模式	42

 深圳微精艺电子有限公司 www.vkinging.com	Product:VK70xN 数据采集器 Revision: 6.07 Issued Date: 2022-10-21
DLL 函数使用手册 (自带 TCP 服务器解析多卡连接数据)	Sheet 3 of 68

6.3 读取文件目录	43
6.4 查询 SD 卡状态	44
6.5 读取一个采样记录文件	45
6.6 读取所有采样文件	46
6.7 删除一个采样记录文件	47
6.8 删除所有采样记录文件	48
6.9 设置下载文件缺省存储路径	49
7. 系统设置函数说明	50
7.1 系统模式	50
7.1.1 切换系统模式	50
7.1.2 读取当前系统模式	53
7.2 系统缺省参数	54
7.2.1 设置缺省参数	54
7.2.2 读取缺省参数	56
7.3 网络参数	57
7.3.1 设置网络	57
7.3.2 读取网络参数	60
8. 处理流程	62
8.1 打开默认端口 (8234) 服务器的操作流程	62
8.2 打开自定义端口服务器的操作流程	63
8.3 多卡连接操作	64
8.4 附件功能操作	65
8.5 切换采样模式操作	66
8.6 SD 卡文件系统操作	67

 深圳微精艺电子有限公司 www.vkinging.com	Product:VK70xN 数据采集器 Revision: 6.07 Issued Date: 2022-10-21
DLL 函数使用手册 (自带 TCP 服务器解析多卡连接数据)	Sheet 4 of 68

1. DLL 函数说明 (自带 TCP 服务器解析多卡连接数据)

DLL 函数包括:

VK70xNMC_DAQ2.dll

用于 VC/C 参考文件包括:

VK70xNMC_DAQ2.h

VK70xNMC_DAQ2.lib

【调用头文件】

```
extern "C"
{
    int VK70xNMC_Initialize(int mci, double refvol, int bitmode, int sr, int volrg15, int
volrg26, int volrg37, int volrg48);
    int VK70xNMC_Initialize_2(int mci, double refvol, int bitmode, int sr, int npoints, int
timeinval, int *volrg);
    int VK70xNMC_StartSampling(int mci);
    int VK70xNMC_StartSampling_NPoints(int mci, int Npointsnums);
    int VK70xNMC_StopSampling(int mci);
    //-----
    int VK70xNMC_Set_AdditionalFeature(int mci, int funcNo, int paral, double para2);
    int VK70xNMC_Set_SystemMode(int mci, int modeval, int samplemethod, int sdfilefmt);
    int VK70xNMC_Set_eNetParameter(int mci, char *para, int len);
    int VK70xNMC_Set_DefaultParameter(int mci, int *para);
    //-----
    int VK70xNMC_Get_ADCCConfigParameter(int mci, int *sr, int *npoints, int *timeintervals, int
*bitmode, int *para, int timeout);
    int VK70xNMC_Get_SystemMode(int mci, int *sysmode, int *samplecmd, int *sdfilefmt, int
timeout);
    int VK70xNMC_Get_DeviceDefaultParameter(int mci, char *rdmodol, char *rdsver, char *rdhdver,
char *rdsn, int *para, int timeout);
    int VK70xNMC_Get_eNetParameter(int mci, char *tvalue, int timeout);
    //=====
    int VK70xNMC_GetOneChannel(int mci, int CHNum, double *adcbuffer, int rsamplenum); // for
VK701N, VK701S, VK702N
    int VK70xNMC_GetFourChannel(int mci, double *adcbuffer, int rsamplenum); // for VK701N, VK701S
    int VK70xNMC_GetAllChannel(int mci, double *adcbuffer, int rsamplenum); // for VK702N
    int VK70xNMC_GetOneChannel_WithIOStatus(int mci, int CHNum, double *adcbuffer, int
rsamplenum, int ioenable); // for VK702N
    int VK70xNMC_GetFourChannel_WithIOStatus(int mci, double *adcbuffer, int rsamplenum, int
ioenable); // for VK702N
    int VK70xNMC_GetAllChannel_WithIOStatus(int mci, double *adcbuffer, int rsamplenum, int
ioenable); // for VK702N
    int VK70xNMC_GetFixedFourChannel_CH1CH2_IO2IO3(int mci, double *adcbuffer, int rsamplenum);
    int VK70xNMC_GetFixedFourChannel_CH3CH4_IO2IO3(int mci, double *adcbuffer, int rsamplenum);
    int VK70xNMC_GetIOStatus(int mci, int *iostatus);
    char* VK70xNMC_GetVersionLot(void);
    //-----
    int VK70xNMC_Get_AllIOS_Blocking(int mci, int *iobuffer, int timeout);
    int VK70xNMC_Get_IO1_Blocking(int mci, int *iovalue, int timeout);
}
```



深圳微精艺电子有限公司

www.vkinging.com

Product:VK70xN 数据采集器

Revision: 6.07

Issued Date: 2022-10-21

DLL 函数使用手册 (自带 TCP 服务器解析多卡连接数据)

Sheet 5 of 68

```
int VK70xNMC_Get_I02_Blocking(int mci, int *iovalue, int timeout);
int VK70xNMC_Get_I03_Blocking(int mci, int *iovalue, int timeout);
int VK70xNMC_Get_I04_Blocking(int mci, int *iovalue, int timeout);
int VK70xNMC_Get_PWM_Blocking(int mci, double *dutyval, int *freqval, int timeout);
int VK70xNMC_Get_DAC_Blocking(int mci, double *dacvalue, int timeout);
int VK70xNMC_Get_Counter_Blocking(int mci, int *countervalue, int timeout);
int VK70xNMC_Get_Temperature_Blocking(int mci, double *tempvalue, int timeout);
int VK70xNMC_Get_Counter_Blocking(int mci, int *countervalue, int timeout);
int VK70xNMC_Get_Temperature_Blocking(int mci, double *tempvalue, int timeout);
//-----
int VK70xNMC_Set_SaveFileDeafultPath(int tflag, const char *defaultdir);
int VK70xNMC_Get_SDFileRuningInformation(int mci, int *para);
int VK70xNMC_Enter_SDFileSystem(int mci);
int VK70xNMC_Exit_SDFileSystem(int mci);
int VK70xNMC_Get_SDFileDIR(int mci);
int VK70xNMC_Get_OneSDFile(int mci, int fileindex);
int VK70xNMC_Get_MultiSDFiles(int mci);
int VK70xNMC_Delete_OneSDFile(int mci, int fileindex);
int VK70xNMC_Delete_MultiSDFiles(int mci);
//=====
int VK70xNMC_Get_ConvertBin2Text_Progress(int *precent, int *tstatus); //progress
int VK70xNMC_Start_ConvertBin2Text(const char *filename);
//-----
int VK70xNMC_Set_DeviceEnterUpgradeMode(int mci);
//-----
int Server_TCPClose(int portnumber);
int Server_TCPOpen(int portnumber);
int Server_Get_ServerPort(int *iport);
int Server_Get_ConnectedClientNumbers(int *cnum);
int Server_Get_ConnectedClientHandle(int mci, int *ihadble, char *ipadr);
int Server_Get_RxTotoalBytes(int *totalbytesnum, int clrflag);
int Server_Bind_ConnectedClientIP(int tflag);
```

```
}
```

 深圳微精艺电子有限公司 www.vkinging.com	Product:VK70xN 数据采集器 Revision: 6.07 Issued Date: 2022-10-21
DLL 函数使用手册 (自带 TCP 服务器解析多卡连接数据)	Sheet 6 of 68

2. DLL 版本信息

2.1 获取 DLL 函数的版本信息

【函数原型】

```
char* VK70xNMC_GetVersionLot(void);
```

输入：无

输出：DLL 功能名称，版本号，释放日期

【VC/C 调用例程】

```
extern "C" char* VK70xNMC_GetVersionLot(void);

Char* GetDLLVersionLot(void)
{
    Char *pstrVL;
    pstrVL= VK70xNMC_GetVersionLot();
    printf( "%s", pstrVL); // 显示 DLL 版本和日期
}
```

3. TCP 服务器函数说明

3.1 打开服务器

【函数原型】- 打开 TCP 服务器

```
int Server_TCPOpen(int portnumber);
```

输入：设置参数说明如下表。

参数名称	参数描述	设置范围
portnumber	TCP 服务器端口	范围 0-65535，默认是 8234

输出：返回状态说明如下表。

返回数值	状态说明
0 或大于 0	服务器打开成功
1	服务器已经打开，未连接客户端
2	服务器已有连接的客户端
-13	服务器端口已被占用！
其它小于 0 的数值	异常退出

【VC/C 调用例程】

```
extern "C" int Server_TCPOpen(int portnumber)
```

 深圳微精艺电子有限公司 www.vkinging.com	Product:VK70xN 数据采集器 Revision: 6.07 Issued Date: 2022-10-21
DLL 函数使用手册 (自带 TCP 服务器解析多卡连接数据)	Sheet 7 of 68

```

void OpenTCPServer(void)
{
    Int tstatus=0;
    Unsigned short int serverport;
    Serverport = 8234;
    tstatus = Server_TCPOpen(Serverport);
    if (tstatus >= 0)printf("服务器打开成功！");
    else if (tstatus == 1)printf("服务器已经打开， 未连接客户端！");
    else if (tstatus == 2)printf("服务器已有连接的客户端！");
    else if (tstatus == -13)printf("服务器端口已被占用！");
    else printf("服务器已打开！");
}

```

3.2 关闭服务器

【函数原型】 关闭已打开 TCP 服务器端口

```
int Server_TCPClose(int portnumber);
```

输入：设置参数说明如下表。

参数名称	参数描述	设置范围
portnumber	已打开 TCP 服务器端口	范围 0-65535 如果置 0， 自动关闭已打开的服务器端口

输出：返回状态说明如下表。

返回数值	状态说明
0 或大于 0	服务器关闭成功
-11	服务器未打开
-12	服务器端口不存在
其它小于 0 的数值	异常退出

【VC/C 调用例程】

```

extern "C" int Server_TCPClose(int portnumber)

void CloseTCPServer(void)
{
    Int tstatus=0;

    tstatus = Server_TCPOpen(0); // 自动关闭已打开的服务器端口
    // 注意：如果不为 0，需要先读取当前打开的服务器端口号
    if (tstatus >= 0)printf("服务器关闭成功！");
    else if (tstatus == -11)printf("服务器未打开！");
}

```

 深圳微精艺电子有限公司 www.vkinging.com	Product:VK70xN 数据采集器 Revision: 6.07 Issued Date: 2022-10-21
DLL 函数使用手册 (自带 TCP 服务器解析多卡连接数据)	Sheet 8 of 68

```

    else if (tstatus == -12)printf("服务器端口不存在！");
    else printf("服务器已打开！");
}

```

3.3 读取已打开服务器的端口号

【函数原型】

```
int Server_Get_ServerPort(int *iport)
```

输入：设置参数说明如下表。

参数名称	参数描述	范围
*iport	用于存放读取的已打开服务器的端口号	0-8

输出：返回状态说明如下表。

返回数值	状态说明
0 或大于 0	读取成功
-11	服务器未打开
其它小于 0 的数值	异常退出

【VC/C 调用例程】

```

extern "C" int Server_Get_ServerPort(int *iport)

void GetTCPServerPort(void)
{
    Int tstatus=0;
    Int iportval =0;

    tstatus = Server_Get_ServerPort(&iportval);

    if (tstatus >= 0) printf("服务器端口号是 %d!", iportval);
    else if (tstatus == -11)printf("服务器未打开！");
    else printf("服务器已打开！");
}

```

3.4 读取已连接服务器上的采集卡数量

【函数原型】

```
int Server_Get_ConnectedClientNumbers (int *inum)
```

输入：设置参数说明如下表。

参数名称	参数描述	范围
------	------	----

 深圳微精艺电子有限公司 www.vkinging.com	Product:VK70xN 数据采集器 Revision: 6.07 Issued Date: 2022-10-21
DLL 函数使用手册 (自带 TCP 服务器解析多卡连接数据)	Sheet 9 of 68

*inum	用于存放读取已连接服务器上的采集卡数量	0-8
-------	---------------------	-----

输出：返回状态说明如下表。

返回数值	状态说明
0 或大于 0	读取成功
-11	服务器未打开
-12	无采集卡连接到服务器
其它小于 0 的数值	异常退出

【VC/C 调用例程】

```
extern "C" int Server_Get_ConnectedClientNumbers(int *inum)

void GetTCPServerClientNumber(void)
{
    Int tstatus=0;
    Int inum =0;

    tstatus = Server_Get_ConnectedClientNumbers(&inum);

    if (tstatus >= 0) printf("已连接了 %d 个采集卡!", inum);
    else if (tstatus == -11)printf("服务器未打开!");
    else if (tstatus == -12)printf("无采集卡连接到服务器!");
    else printf("服务器已打开!");
}
```

3.5 读取当前采集卡的句柄和 IP 地址

【函数原型】

```
int Server_Get_ConectedClientHandle(int mci,int *iihanlde,char *ipadr)
```

输入：设置参数说明如下表。

参数名称	参数描述	范围
int mci	已连接到服务器上的采集卡序号	范围：0-7。 注意：目前库函数版本仅仅支持最大连接采集卡 8 个。
int *iihanlde	用于存放读取的当前活动的采集卡的连接句柄指针	1-65535
char *ipadr	用于存放读取的当前活动的采集卡的 IP 地址指针， 要求至少保	字符，1-250 字节

	证有 128 字节空间。	
--	--------------	--

输出：返回状态说明如下表。

返回数值	状态说明
0 或>0	读取成功
-11	服务器未打开
-12	请求的采集卡索引号错误
-13	请求的采集卡未连接或不存在
其它小于 0 的数值	异常退出

【VC/C 调用例程】

```
extern "C" int Server_Get_ConnectedClientHandle(int mci,int *iihanlde,char
*ipadr)

void GetTCPServerHandleAndIP(void)
{
    Int tstatus=0;
    Int ihandle =0;
    char ipadr[250];

    tstatus = Server_Get_ConnectedClientHandle(0,&ihandle, ipadr);

    if (tstatus >= 0) printf("当前 0 号采集卡连接句柄是 %d; IP 地址 %s!",
ihandle, ipadr);
    else if (tstatus == -11)printf("服务器未打开!");
    else if (tstatus == -12)printf("请求的采集卡索引号错误!");
    else if (tstatus == -13)printf("请求的采集卡未连接或不存在!");
    else printf("服务器已打开!");
}
```

3.6 读取当前服务器端接收总字节数

【函数原型】

```
int Server_Get_RxTotoalBytes(int *totalbytesnum, int clrflag)
```

输入：设置参数说明如下表。

参数名称	参数描述	范围
int *totalbytesnum	用于存放读取的当前服务器接收总字节数的指针	Int
int clrflag	读取完成清除接收字节数	0: 累计，不清零 1: 读取完成后，清零

 深圳微精艺电子有限公司 www.vkinging.com	Product:VK70xN 数据采集器 Revision: 6.07 Issued Date: 2022-10-21
DLL 函数使用手册 (自带 TCP 服务器解析多卡连接数据)	Sheet 11 of 68

输出：返回状态说明如下表。

返回数值	状态说明
0 或大于 0	读取成功
其它小于 0 的数值	异常退出

【VC/C 调用例程】

```
extern "C" int  Server_Get_RxTotoalBytes(int *totalbytesnum, int clrflag)

void GetTCPServerRXbytesNumber(void)
{
    Int tstatus=0;
    Int itotalnum =0;

    tstatus = Server_Get_RxTotoalBytes(&itotalnum, 1);// 读取完成清零再计数

    if (tstatus >= 0) printf("当前服务器数据传输速率 %d KB/S! ", itotalnum/
(rdintervaltimes*1000));
    else printf("服务器已打开!");
}
```

 深圳微精艺电子有限公司 www.vkinging.com	Product:VK70xN 数据采集器 Revision: 6.07 Issued Date: 2022-10-21
DLL 函数使用手册 (自带 TCP 服务器解析多卡连接数据)	Sheet 12 of 68

4. ADC 数据采集函数说明

4.1 初始化已连接的采集卡参数(适用于 VK701N, VK702N)

【函数原型】

```
int VK70xNMC_Initialize(int mci,double refvol,int bitmode, int sr, int volrg15,
int volrg26,int volrg37,int volrg48) ;
```

输入：设置参数说明如下表。

参数名称	参数描述	设置范围
int mci	已连接到服务器上的采集卡序号	范围：0-7。 注意：目前库函数版本仅仅支持最大连接采集卡 8 个。
Double refvol	选择采集卡模式	VK701N (PGA) : 4.000 或 1 VK702N (NOPGA) : 2.442 或 0 VK702S (NOPGA) : 2.442 或 0
int bitmode	采样精度	0: 8 位 1: 16 位 2 或 3: 24 位
int sr	采样率	1-100K 采样率。 注：VK701N 如果 24 位采样精度，最高采样率为 50KHZ
int volrg15,	通道 1,5 电压输入范围	VK701N : 0~7, 参考【其它说明】章节。 VK702N : 0~4, 参考【其它说明】章节。
int volrg26,	通道 2,6 电压输入范围	
int volrg37,	通道 3,7 电压输入范围	
int volrg48	通道 4,8 电压输入范围	

输出：返回状态说明如下表。

返回数值	状态说明
0 或大于 0	设置成功
-11	服务器未打开
-12	采集卡未连接或不存在
-13	采集卡未连接或不存在
其它小于 0 的数值	异常退出

【VC/C 调用例程】

```
extern "C" int VK70xNMC_Initialize(int mci,double refvol,int bitmode, int sr,
int volrg15, int volrg26,int volrg37,int volrg48)
```

```
Void IntiVK70xNMC(void)
{
    int rstatus;
```

 深圳微精艺电子有限公司 www.vkinging.com	Product:VK70xN 数据采集器 Revision: 6.07 Issued Date: 2022-10-21
DLL 函数使用手册 (自带 TCP 服务器解析多卡连接数据)	Sheet 13 of 68

```

int  uCH15VR,uCH26VR,uCH37VR,uCH48VR;
double refvol=2.442;
int  tmpbitmode,tmpsr;

uCH15VR=0; //设置通道 1,5 输入电压范围为-10V~10V
uCH26VR=1; //设置通道 2,6 输入电压范围为-5V~5V
uCH37VR=2; //设置通道 3,7 输入电压范围为-2.5V~2.5V
uCH48VR=4; //设置通道 4,8 输入电压范围为-500mV~500mV
tmpbitmode=1;//16 位
tmpsr=1000; /// 1KHZ 采样率

rstatus= VK70xNMC_Initialize(0,refvol, tmpbitmode , tmpsr,
uCH15VR,uCH26VR,uCH37VR,uCH48VR); // 初始化已连接的 0 号采集卡
if (rstatus >= 0) printf("%s", '操作成功!');
else if (tstatus == -11)printf("服务器未打开!");
else if (tstatus == -12 || tstatus == -13)printf("采集卡未连接或不存在!");
else printf("%s", '异常退出!');
}

```

4.2 初始化已连接的采集卡参数 2(适用于 VK701N, VK701NSD, VK702N)

【函数原型】

```
int VK70xNMC_Initialize_2(int mci, double refvol, int bitmode, int sr, int
npoints, int timeinval, int *volrg);
```

输入：设置参数说明如下表。

参数名称	参数描述	设置范围
int mci	已连接到服务器上的采集卡序号	范围：0-7。 注意：目前库函数版本仅仅支持最大连接采集卡 8 个。
Double refvol	选择采集卡模式	VK701N (PGA)：4.000 或 1 VK701NSD (PGA)：4.000 或 1 VK702N (NOPGA)：2.442 或 0 VK702S (NOPGA)：2.442 或 0
int bitmode	采样精度	0： 8 位 1： 16 位 2 或 3： 24 位
int sr	采样率	1-100K 采样率。 注：VK701N 如果 24 位采样精度，最高采样率为 50KHZ
int npoints	N 采样待采样点数	N 采样待采样点数
int timeinval	定时间隔 N 采样时间间隔	单位为秒。最小 1 秒间隔

 深圳微精艺电子有限公司 www.vkinging.com	Product:VK70xN 数据采集器 Revision: 6.07 Issued Date: 2022-10-21
DLL 函数使用手册 (自带 TCP 服务器解析多卡连接数据)	Sheet 14 of 68

int *volrg	通道电压输入范围	VK701N/VK701NSD : 0~7, 参考【其它说明】章节。 VK702N : 0~4, 参考【其它说明】章节。 Volrg 1 为通道 1,5 电压输入范围 Volrg 2 为通道 2,6 电压输入范围 Volrg 3 为通道 3,7 电压输入范围 Volrg 4 为通道 4,8 电压输入范围
------------	----------	---

输出：返回状态说明如下表。

返回数值	状态说明
0 或大于 0	设置成功
-11	服务器未打开
-12	采集卡未连接或不存在
-13	采集卡未连接或不存在
其它小于 0 的数值	异常退出

【VC/C 调用例程】

```
extern "C" int VK70xNMC_Initialize_2(int mci, double refvol, int bitmode, int sr, int npoints, int timeinval, int *volrg)
```

```
Void IntiVK70xNMC(void)
```

```
{
```

```
    int rstatus;
```

```
    int Volrg[4];
```

```
    int Npoints,Timeinval;
```

```
    double refvol=2.442;
```

```
    int tmpbitmode,tmpsr;
```

```
    Volrg[0]=0; //设置通道 1,5 输入电压范围为-10V~10V
```

```
    Volrg[1]=1; //设置通道 2,6 输入电压范围为-5V~5V
```

```
    Volrg[2]=2; //设置通道 3,7 输入电压范围为-2.5V~2.5V
```

```
    Volrg[3]=4; //设置通道 4,8 输入电压范围为-500mV~500mV
```

```
    tmpbitmode=1; //16 位
```

```
    tmpsr=1000; /// 1KHZ 采样率
```

```
    Npoints=10000; // N 采样 10000 点
```

```
    Timeinval=5; //定时采样间隔 5 秒
```

```
    rstatus= VK70xNMC_Initialize_2(0,refvol, tmpbitmode , tmpsr,
    Npoints,Timeinval,Volrg); // 初始化已连接的 0 号采集卡
    if (rstatus >= 0) printf("%s", '操作成功!');
    else if (tstatus == -11)printf("服务器未打开!");
```

 深圳微精艺电子有限公司 www.vkinging.com	Product:VK70xN 数据采集器 Revision: 6.07 Issued Date: 2022-10-21
DLL 函数使用手册 (自带 TCP 服务器解析多卡连接数据)	Sheet 15 of 68

```

    else if (tstatus == -12 || tstatus == -13)printf("采集卡未连接或不存在！");
    else printf("%s", '异常退出！');
}

```

4.3 开始连续采样

【函数原型】

```
int VK70xNMC_StartSampling(int mci);
```

输入：设置参数说明如下表。

参数名称	参数描述	设置范围
int mci	已连接到服务器上的采集卡序号	范围：0-7。 注意：目前库函数版本仅仅支持最大连接采集卡 8 个。

输出：返回状态说明如下表。

返回数值	状态说明
0 或大于 0	操作成功，第一次开始采样
-11	服务器未打开
-12	未检测到采集卡
-13	采集卡未连接或不存在
其它小于 0 的数据	异常退出

【VC/C 调用例程】

```
extern "C" int VK70xNMC_StartSampling (void);
```

```
Void VK70xNMC_StartSampling_ActivedClient (void)
```

```

{
    int rstatus;
    rstatus = VK70xNMC_StartSampling(0); //开始 0 号采集卡采样
    if (rstatus >= 0) printf("%s", '操作成功，开始采样！');
    else if (rstatus == -11) printf("%s", '服务器未打开！');
    else if (tstatus == -12 || tstatus == -13) printf("%s", '未检测到采集卡！');
    else printf("%s", '异常退出！');
}

```

4.4 开始 N 点采样

【函数原型】

```
int VK70xNMC_StartSampling_NPoints(int mci,int Npointsnums)
```

输入：参数说明如下表。

参数名称	参数类型	参数描述
int mci	已连接到服务器上的采集卡序号	范围：0-7。 注意：目前库函数版本仅仅支持最大连接采集卡 8 个。
int Npointsnums	N 点参数	0- 连续模式

 深圳微精艺电子有限公司 www.vkinging.com	Product:VK70xN 数据采集器 Revision: 6.07 Issued Date: 2022-10-21
DLL 函数使用手册 (自带 TCP 服务器解析多卡连接数据)	Sheet 16 of 68

		1- 停止采样
		2- N: N 点采样

输出：返回状态说明如下表。

返回数值	状态说明
0 或大于 0	操作成功，第一次开始采样
-11	服务器未打开
-12	未检测到采集卡
-13	采集卡未连接或不存在
其它小于 0 的数据	异常退出

【VC/C 调用例程】

```
extern "C" int  VK70xNMC_StartSampling_NPoints(int mci,int Npointsnums);

Void VK70xNMC_ StartSamplingNP_ActivedClient (void)
{
    int rstatus;
    rstatus  = VK70xNMC_StartSampling_ NPoints (0,10000); //开始 0 号采集卡采样
10000 点
    if (rstatus >= 0) printf("%s", '操作成功，开始采样!');
    else if (rstatus == -11) printf("%s", '服务器未打开!');
    else if (tstatus == -12 || tstatus == -13) printf("%s", '未检测到采集卡!
');
    else printf("%s", '异常退出!');
}
```

4.5 停止采样

【函数原型】

```
int  VK70xNMC_StopSampling(int mci);
```

输入：设置参数说明如下表。

参数名称	参数描述	设置范围
int mci	已连接到服务器上的采集卡序号	范围：0-7。 注意：目前库函数版本仅仅支持最大连接采集卡 8 个。

输出：返回状态说明如下表。

返回数值	状态说明
0 或大于 0	操作成功，采样停止
-11	服务器未打开
-12	未检测到采集卡
-13	采集卡未连接或不存在

 深圳微精艺电子有限公司 www.vkinging.com	Product:VK70xN 数据采集器 Revision: 6.07 Issued Date: 2022-10-21
DLL 函数使用手册 (自带 TCP 服务器解析多卡连接数据)	Sheet 17 of 68

其它小于 0 的数据	异常退出
------------	------

【VC/C 调用例程】

```
extern "C" int VK70xNMC_StopSampling (void);

Void VK70xNMC_StopSampling _ActivatedClient (void)
{
    int rstatus;
    rstatus = VK70xNMC_StopSampling (0); //停止 0 号采集卡
    if (rstatus >= 0) printf("%s", '操作成功, 采样停止!');
    else if (tstatus == -12 || tstatus == -13) printf("%s", '服务器未打开!');
    else printf("%s", '异常退出!');
}
```

4.6 读取单通道数据

【函数原型】

```
int VK70xNMC_GetOneChannel(int mci, int CHno, double *adcbuffer, int rsamplenum)
```

输入：参数说明如下表。

参数名称	参数类型	参数描述
int mci	已连接到服务器上的采集卡序号	范围：0-7。 注意：目前库函数版本仅仅支持最大连接采集卡 8 个。
CHno	32 位整形	通道序号。 0-读取采样通道 1; 1-读取采样通道 2; 2-读取采样通道 3; 3-读取采样通道 4。 4-读取采样通道 5; 5-读取采样通道 6; 6-读取采样通道 7; 7-读取采样通道 8。
adcbuffer	双精度浮点指针	读数数据首地址指针
rsamplenum	32 位整形	准备读取采样点数

输出：返回状态说明如下表。

返回数值	状态说明
>0	实际读出采样点数。
0	无数据。
其它小于 0 的数据	异常退出

 深圳微精艺电子有限公司 www.vkinging.com	Product:VK70xN 数据采集器 Revision: 6.07 Issued Date: 2022-10-21
DLL 函数使用手册 (自带 TCP 服务器解析多卡连接数据)	Sheet 18 of 68

【VC/C 调用例程】

```
extern "C" int VK70xNMC_GetOneChannel(int mci,int CHno, double *adcbuffer,
int rsamplenum);
#define SAMPLENUMLEN 100 // 定义每次采样 100 点
```

```
Void ReadOneChannelADCSample(void)
```

```
{
    int rstatus;
    int CHno, rsamplenum, waittime,i;
    double SaveADCBuffer[SAMPLENUMLEN];

    CHno = 2; // 准备读取通道 3 ADC 采样数据
    Rsamplenum = SAMPLENUMLEN; // 准备读取 0 号采集卡采样 100 点
    rstatus= VK70xNMC_GetOneChannel (0,CHno, SaveADCBuffer, Rsamplenum);
    if (rstatus > 0)
    {
        Rsamplenum= rstatus;
        printf("读取通道%d 新采样数=%d\n", CHno ,Rsamplenum);
        For(i=0;i< Rsamplenum;i++)
        {
            printf("采样点%d 的数据是%f\n", i,SaveADCBuffer[i]);
        }
    }
    else if (rstatus < 0) printf("%s", '数据异常!');
}
```

 深圳微精艺电子有限公司 www.vkinging.com	Product:VK70xN 数据采集器 Revision: 6.07 Issued Date: 2022-10-21
DLL 函数使用手册 (自带 TCP 服务器解析多卡连接数据)	Sheet 19 of 68

4.7 读取 4 通道采样数据（适用于 VK701S, VK701N, VK702N, VK701NSD）

【函数原型】

```
int VK70xNMC_GetFourChannel (int mci, double *adcbuffer, int rsamplenum)
```

输入：参数说明如下表。

参数名称	参数类型	参数描述
int mci	已连接到服务器上的采集卡序号	范围：0-7。注意：目前库函数版本 仅仅支持最大连接采集卡 8 个。
adcbuffer	双精度浮点指针	读数采样数据首地址指针
rsamplenum	32 位整形	准备读取采样点数

输出：返回状态说明如下表。

返回数值	状态说明
>0	实际读出采样点数。
0	无数据。
其它小于 0 数据	异常退出

【VC/C 调用例程】

```
extern "C" int VK70xNMC_GetFourChannel(int mci, double *adcbuffer, int
rsamplenum);
#define SAMPLENUMLEN 100 // 定义每次采样 100 点

Void ReadALLADCsSample(void)
{
    int rstatus;
    int rsamplenum, i;
    double SaveADCBuffer[SAMPLENUMLEN*4];

    Rsamplenum = SAMPLENUMLEN; // 准备读取 0 号采集卡采样 100 点
    Rstatus = VK70xNMC_GetFourChannel(0, SaveADCBuffer, Rsamplenum);
    if (rstatus > 0)
    {
        Rsamplenum= rstatus;
        printf("读取采样数=%d\n", Rsamplenum);
        For(i=0; i< Rsamplenum; i++)
        {
            printf("通道 1 采样点%d 的数据是%f\n", i, SaveADCBuffer[4*i]);
            printf("通道 2 采样点%d 的数据是%f\n", i, SaveADCBuffer[4*i+1]);
            printf("通道 3 采样点%d 的数据是%f\n", i, SaveADCBuffer[4*i+2]);
            printf("通道 4 采样点%d 的数据是%f\n", i, SaveADCBuffer[4*i+3]);
        }
    }
}
```

 深圳微精艺电子有限公司 www.vkinging.com	Product:VK70xN 数据采集器 Revision: 6.07 Issued Date: 2022-10-21
DLL 函数使用手册（自带 TCP 服务器解析多卡连接数据）	Sheet 20 of 68

```

    else if (rstatus < 0) printf("%s", '数据异常!');
  }

```

4.8 读取 8 通道采样数据（适用于 VK702N）

【函数原型】

```
int VK70xNMC_GetAllChannel(int mci,double *adcbuffer, int rsamplenum)
```

输入：参数说明如下表。

参数名称	参数类型	参数描述
int mci	已连接到服务器上的采集卡序号	范围：0-7。 注意：目前库函数版本仅仅支持最大连接采集卡 8 个。
adcbuffer	双精度浮点指针	读数采样数据首地址指针
rsamplenum	32 位整形	准备读取采样点数

输出：返回状态说明如下表。

返回数值	状态说明
>0	实际读出采样点数。
0	无数据。
其它小于 0 数据	异常退出

【VC/C 调用例程】

```

extern "C" int VK70xNMC_GetAllChannel(int mci,double *adcbuffer, int
rsamplenum);
#define SAMPLENUMLEN 100 // 定义每次采样 100 点

Void ReadALLADCSample(void)
{
    int rstatus;
    int rsamplenum, i;
    double SaveADCBuffer[SAMPLENUMLEN*8];

    Rsamplenum = SAMPLENUMLEN; // 准备读取 0 号采集卡采样 100 点
    Rstatus = VK70xNMC_GetAllChannel(0, SaveADCBuffer, Rsamplenum);
    if (rstatus > 0)
    {
        Rsamplenum= rstatus;
        printf("读取采样数=%d\n",Rsamplenum);
        For(i=0;i< Rsamplenum;i++)
        {
            printf("通道 1 采样点%d 的数据是%f\n", i,SaveADCBuffer[8*i]);
            printf("通道 2 采样点%d 的数据是%f\n", i,SaveADCBuffer[8*i+1]);
            printf("通道 3 采样点%d 的数据是%f\n", i,SaveADCBuffer[8*i+2]);

```

 深圳微精艺电子有限公司 www.vkinging.com	Product:VK70xN 数据采集器 Revision: 6.07 Issued Date: 2022-10-21
DLL 函数使用手册 (自带 TCP 服务器解析多卡连接数据)	Sheet 21 of 68

```

        printf("通道 4 采样点%d 的数据是%f\n", i, SaveADCBuffer[8*i+3]);
        printf("通道 5 采样点%d 的数据是%f\n", i, SaveADCBuffer[8*i+4]);
        printf("通道 6 采样点%d 的数据是%f\n", i, SaveADCBuffer[8*i+5]);
        printf("通道 7 采样点%d 的数据是%f\n", i, SaveADCBuffer[8*i+6]);
        printf("通道 8 采样点%d 的数据是%f\n", i, SaveADCBuffer[8*i+7]);
    }
}
else if (rstatus < 0) printf("%s", '数据异常!');
}

```

4.9 停止当前采集卡采样

【函数原型】

Int VK70xNMC_StopSampling(int mci);

输入：参数说明如下表。

参数名称	参数类型	参数描述
int mci	已连接到服务器上的采集卡序号	范围：0-7。 注意：目前库函数版本仅仅支持最大连接采集卡 8 个。

输出：返回状态说明如下表。

返回数值	状态说明
0 或>0	操作成功
-11	服务器未打开
-12	请求的采集卡索引号错误
-13	请求的采集卡未连接或不存在

【VC/C 调用例程】

```

extern "C" int VK70xNMC_StopTask(int mci);
int StopLANDatumDisposalTask(void)
{
    int status;
    status = VK70xNMC_StopTask(0); // 停止 0 号采集卡采集
    //if (status==1) printf("成功停止线程任务"); // 成功停止线程任务
    //else printf("线程未启动或停止异常"); // 线程未启动或停止异常!!
    Return status;
}

```

 深圳微精艺电子有限公司 www.vkinging.com	Product:VK70xN 数据采集器 Revision: 6.07 Issued Date: 2022-10-21
DLL 函数使用手册 (自带 TCP 服务器解析多卡连接数据)	Sheet 22 of 68

4.10 读取 I02 和 I03 状态（适用于 VK701N, VK701NSD 和 VK702N 的 24 位采样模式）

【函数原型】

int VK70xNMC_GetIOStatus(int mci,int *iostatus);

输入：用于存放 I01, I02, I03, I04 指针地址

输出：状态

输入：参数说明如下表。

参数名称	参数类型	参数描述
int mci	已连接到服务器上的采集卡序号	范围：0-7。注意：目前库函数版本仅仅支持最大连接采集卡 8 个。
iostatus	整型指针	读数 I01, I02, I03, I04 首地址指针

输出：返回状态说明如下表。

返回数值	状态说明
0	读取成功。IO 状态如下： 0：低电平； 1：高电平； -1：未知状态
-1	采集卡工作模式不支持 IO 状态返回
-2	采集卡不支持 IO 状态返回
其它小于 0 数据	异常退出

警告：此函数支持 VK701N 和 VK702N 采集卡，如果此函数读取 I02 和 I03 状态必要要求采集卡工作在 24bit 采集模式下，而且保证初始化函数 VK70xNMC_Initialize(int mci,double refmode, int uCH15VR, int uCH26VR,int uCH37VR,int uCH48VR)中 refmode 等于 2.442，即选择 VK702N 采集卡类型；等于 4.000，即选择 VK701N 采集卡类型。否则的话，而且要保证采集卡的 IO 设置为输入状态，读出 IO 状态是无意义。

【VC/C 调用例程】

```
extern "C" int VK70xNMC_GetIOStatus (int mci,int *iostatus);
int GetIO2andIO3Status(void)
{
    int IOstatus[4], status;
    status = VK70xNMC_GetIO2andIO3Status(0,&IOstatus);
    //if (status==0) printf("I02 和 I03 状态");//0：低电平；1：高电平；-1：未知
    //else if (status==-1) printf("采集卡工作模式不支持 IO 状态返回");
    //else if (status==-2) printf("采集卡不支持 IO 状态返回");
    //else printf("状态异常");
    Return status;
}
```

 深圳微精艺电子有限公司 www.vkinging.com	Product:VK70xN 数据采集器 Revision: 6.07 Issued Date: 2022-10-21
DLL 函数使用手册 (自带 TCP 服务器解析多卡连接数据)	Sheet 23 of 68

4.11 读取带 IO 状态的单通道数据（适用于 VK701N, VK701NSD 和 VK702N）

【函数原型】

```
int VK70xNMC_GetOneChannel_WithIOStatus(int mci,int CHno, double *adcbuffer,
int rsamplenum,int IOenable)
```

输入：参数说明如下表。

参数名称	参数类型	参数描述
int mci	已连接到服务器上的采集卡序号	范围：0-7。 注意：目前库函数版本仅仅支持最大连接采集卡 8 个。
CHno	32 位整形	通道序号。 0 读取采样通道 1； 1-读取采样通道 2； 2-读取采样通道 3； 3-读取采样通道 4。 4-读取采样通道 5； 5-读取采样通道 6； 6-读取采样通道 7； 7-读取采样通道 8。
adcbuffer	双精度浮点指针	读数数据首地址指针
rsamplenum	32 位整形	准备读取采样点数
IOenable	32 位整形	0：不读取 IO 状态 1：读取 IO2 状态 2：读取 IO3 状态 3：读取 IO2 和 IO3 状态 其它值：不读取 IO 状态

输出：返回状态说明如下表。

返回数值	状态说明
>0	实际读出采样点数。
0	无数据。
其它小于 0 数据	异常退出

警告：此函数支持 VK701N 和 VK702N 采集卡，如果此函数读取 IO2 和 IO3 状态必要要求采集卡工作在 24bit 采集模式下，而且保证初始化函数 VK70xNMC_Initialize(int mci,double refmode, int uCH15VR, int uCH26VR,int uCH37VR,int uCH48VR)中 refmode 等于 2.442，即选择 VK702N 采集卡类型；等于 4.000，即选择 VK701N 采集卡类型。否则的话，而且要保证采集卡的 IO 设置为输入状态，读出 IO 状态是无意义。

【VC/C 调用例程】

```
extern "C" int VK70xNMC_GetOneChannel(int mci,int CHno, double *adcbuffer,
int rsamplenum,int IOenable);
#define SAMPLENUMLEN 100 // 定义每次采样 100 点
```

 深圳微精艺电子有限公司 www.vkinging.com	Product:VK70xN 数据采集器 Revision: 6.07 Issued Date: 2022-10-21
DLL 函数使用手册 (自带 TCP 服务器解析多卡连接数据)	Sheet 24 of 68

```

Void ReadOneChannelADCSample(void)
{
    int rstatus;
    int CHno, rsamplenum, waittime,i;
    double SaveADCBuffer[SAMPLENUMLEN * 3]; //准备 1 路 ADC 数据+2 路 IOzh 状态
buffer

    CHno = 2; // 准备读取通道 3 ADC 采样数据
    Rsamplenum = SAMPLENUMLEN; // 准备读取采样 100 点
    rstatus= VK70xNMC_GetOneChannel_WithIOStatus (0,CHno, SaveADCBuffer,
Rsamplenum,3); //同时读取 IO2 和 IO3 状态
    if (rstatus > 0)
    {
        Rsamplenum= rstatus;
        printf("读取通道%d 新采样数=%d\n", CHno ,Rsamplenum);
        For(i=0;i< Rsamplenum;i++)
        {
            printf("采样点%d 的通道 3 数据是%f\n", i, SaveADCBuffer[3*i]);
            printf("采样点%d 的 IO2 状态是%f\n", i, SaveADCBuffer[3*i+1]);
            printf("采样点%d 的 IO3 状态是%f\n", i, SaveADCBuffer[3*i+2]);
        }
    }
    else if (rstatus < 0) printf("%s", '数据异常!');
}

```

重要提醒:

如果不读 IO 时，直接把 IOenble 参数置 0，同时 SaveADCBuffer 指针数据中也返回不带 IO 状态的数据；如果把 IOenble 参数置 1 时，SaveADCBuffer 指针数据中也返回带 1 通道 IO2 状态的数据；如果把 IOenble 参数置 2 时，SaveADCBuffer 指针数据中也返回带 1 通道 IO3 状态的数据；如果把 IOenble 参数置 3 时，SaveADCBuffer 指针数据中也返回带 2 通道 IO2 和 IO3 状态的数据。

切记调试此函数要注意读取缓冲 buffer 长度设置，如读 100 个采样点，设置 IOenble 参数为 3 时，此时 buffer 长度一定要是平时 3 倍；设置 IOenble 参数为 1 或 2 时，此时 buffer 长度一定要是平时 2 倍；其它设置不变。

 深圳微精艺电子有限公司 www.vkinging.com	Product:VK70xN 数据采集器 Revision: 6.07 Issued Date: 2022-10-21
DLL 函数使用手册 (自带 TCP 服务器解析多卡连接数据)	Sheet 25 of 68

4.12 读取带 I0 状态的 4 通道采样数据 (适用于 VK701N, VK701NSD 和 VK702N)

【函数原型】

```
int VK70xNMC_GetFourChannel_WithIOStatus(int mci, double *adcbuffer, int rsamplenum, int IOenable)
```

输入：参数说明如下表。

参数名称	参数类型	参数描述
int mci	已连接到服务器上的采集卡序号	范围：0-7。注意：目前库函数版本仅仅支持最大连接采集卡 8 个。
adcbuffer	双精度浮点指针	读数采样数据首地址指针
rsamplenum	32 位整形	准备读取采样点数
IOenable	32 位整形	0：不读取 I0 状态 1：读取 I02 状态 2：读取 I03 状态 3：读取 I02 和 I03 状态 其它值：不读取 I0 状态

输出：返回状态说明如下表。

返回数值	状态说明
>0	实际读出采样点数。
0	无数据。
其它小于 0 数据	异常退出

警告：

此函数支持 VK701N 和 VK702N 采集卡，如果此函数读取 I02 和 I03 状态必要要求采集卡工作在 24bit 采集模式下，而且保证初始化函数 VK70xNMC_Initialize(int mci, double refmode, int uCH15VR, int uCH26VR, int uCH37VR, int uCH48VR) 中 refmode 等于 2.442，即选择 VK702N 采集卡类型；等于 4.000，即选择 VK701N 采集卡类型。否则的话，而且要保证采集卡的 I0 设置为输入状态，读出 I0 状态是无意义。

【VC/C 调用例程】

```
extern "C" int VK70xNMC_GetFourChannel_WithIOStatus (int mci, double *adcbuffer, int rsamplenum);
```

```
#define SAMPLENUMLEN 100 // 定义每次采样 100 点
```

```
Void ReadALLADCSample(void)
{
    int rstatus;
    int rsamplenum, i;
```

 深圳微精艺电子有限公司 www.vkinging.com	Product:VK70xN 数据采集器 Revision: 6.07 Issued Date: 2022-10-21
DLL 函数使用手册 (自带 TCP 服务器解析多卡连接数据)	Sheet 26 of 68

```

double SaveADCBuffer[SAMPLENUMLEN* (4 + 1)]; // 4 通道数据+1 通道 I0 状态

Rsamplenum = SAMPLENUMLEN; // 准备读取采样 100 点
Rstatus = VK70xNMC_GetFourChannel_WithIOStatus (0, SaveADCBuffer, Rsamplenum,
1); // 设置读取 I02 状态
if (rstatus > 0)
{
    Rsamplenum= rstatus;
    printf("读取采样数=%d\n", Rsamplenum);
    For(i=0; i< Rsamplenum; i++)
    {
        printf("通道 1 采样点%d 的数据是%f\n", i, SaveADCBuffer[5*i]);
        printf("通道 2 采样点%d 的数据是%f\n", i, SaveADCBuffer[5*i+1]);
        printf("通道 3 采样点%d 的数据是%f\n", i, SaveADCBuffer[5*i+2]);
        printf("通道 4 采样点%d 的数据是%f\n", i, SaveADCBuffer[5*i+3]);
        printf("采样点%d 的 I02 状态是%f\n", i, SaveADCBuffer[5*i+4]);
    }
}
else if (rstatus < 0) printf("%s", '数据异常!');
}

```

重要提醒:

如果不读 I0 时，直接把 I0enble 参数置 0，同时 SaveADCBuffer 指针数据中也返回不带 I0 状态的数据；如果把 I0enble 参数置 1 时，SaveADCBuffer 指针数据中也返回带 1 通道 I02 状态的数据；如果把 I0enble 参数置 2 时，SaveADCBuffer 指针数据中也返回带 1 通道 I03 状态的数据；如果把 I0enble 参数置 3 时，SaveADCBuffer 指针数据中也返回带 2 通道 I02 和 I03 状态的数据。

切记调试此函数要注意读取缓冲 buffer 长度设置，如读 100 个采样点，设置 I0enble 参数为 3 时，此时 buffer 长度一定比平时多 2 个 I0 状态通道数据；设置 I0enble 参数为 1 或 2 时，此时 buffer 长度一定比平时多 1 个 I0 状态通道数据；其它设置不变。

 深圳微精艺电子有限公司 www.vkinging.com	Product:VK70xN 数据采集器 Revision: 6.07 Issued Date: 2022-10-21
DLL 函数使用手册 (自带 TCP 服务器解析多卡连接数据)	Sheet 27 of 68

4.13 读取带 IO 状态的 8 通道采样数据（适用于 VK702N）

【函数原型】

```
int VK70xNMC_GetAllChannel_WithIOStatus(int mci,double *adcbuffer, int rsamplenum,int IOenable)
```

输入：参数说明如下表。

参数名称	参数类型	参数描述
int mci	已连接到服务器上的采集卡序号	范围：0-7。 注意：目前库函数版本仅仅支持最大连接采集卡 8 个。
adcbuffer	双精度浮点指针	读数采样数据首地址指针
rsamplenum	32 位整形	准备读取采样点数
IOenable	32 位整形	0：不读取 IO 状态 1：读取 IO2 状态 2：读取 IO3 状态 3：读取 IO2 和 IO3 状态 其它值：不读取 IO 状态

输出：返回状态说明如下表。

返回数值	状态说明
>0	实际读出采样点数。
0	无数据。
其它小于 0 数据	异常退出

警告：

此函数仅仅支持 VK702N 采集卡，如果此函数读取 IO2 和 IO3 状态必要要求采集卡工作在 24bit 采集模式下，而且保证初始化函数 VK70xNMC_Initialize(int mci,double refmode, int uCH15VR, int uCH26VR,int uCH37VR,int uCH48VR)中 refmode 等于 2.442，即选择 VK702N 采集卡类型。否则的话，而且要保证采集卡的 IO 设置为输入状态，读出 IO 状态是无意义。

【VC/C 调用例程】

```
extern "C" int VK70xNMC_GetAllChannel_WithIOStatus (int mci,double
*adcbuffer, int rsamplenum, int IOenable);
#define SAMPLENUMLEN 100 // 定义每次采样 100 点
```

```
Void ReadALLADCSample(void)
{
    int rstatus;
    int rsamplenum, i;
```

 深圳微精艺电子有限公司 www.vkinging.com	Product:VK70xN 数据采集器 Revision: 6.07 Issued Date: 2022-10-21
DLL 函数使用手册（自带 TCP 服务器解析多卡连接数据）	Sheet 28 of 68

```

double SaveADCBuffer[SAMPLENUMLEN*(8+1)];//读取 I03 通道数据

Rsamplenum = SAMPLENUMLEN; // 准备读取采样 100 点
Rstatus = VK70xNMC_GetAllChannel_WithIOStatus (0,SaveADCBuffer, Rsamplenum,
2);// 设置读取 I03 通道数据
if (rstatus > 0)
{
    Rsamplenum= rstatus;
    printf("读取采样数=%d\n",Rsamplenum);
    For(i=0;i< Rsamplenum;i++)
    {
        printf("通道 1 采样点%d 的数据是%f\n", i, SaveADCBuffer[9*i]);
        printf("通道 2 采样点%d 的数据是%f\n", i, SaveADCBuffer[9*i+1]);
        printf("通道 3 采样点%d 的数据是%f\n", i, SaveADCBuffer[9*i+2]);
        printf("通道 4 采样点%d 的数据是%f\n", i, SaveADCBuffer[9*i+3]);
        printf("通道 5 采样点%d 的数据是%f\n", i, SaveADCBuffer[9*i+4]);
        printf("通道 6 采样点%d 的数据是%f\n", i, SaveADCBuffer[9*i+5]);
        printf("通道 7 采样点%d 的数据是%f\n", i, SaveADCBuffer[9*i+6]);
        printf("通道 8 采样点%d 的数据是%f\n", i, SaveADCBuffer[9*i+7]);
        printf("采样点%d 的 I03 状态是%f\n", i, SaveADCBuffer[9*i+8]);
    }
}
else if (rstatus < 0) printf("%s", '数据异常!');
}

```

重要提醒:

如果不读 I0 时，直接把 I0enable 参数置 0，同时 SaveADCBuffer 指针数据中也返回不带 I0 状态的数据；如果把 I0enable 参数置 1 时，SaveADCBuffer 指针数据中也返回带 1 通道 I02 状态的数据；如果把 I0enable 参数置 2 时，SaveADCBuffer 指针数据中也返回带 1 通道 I03 状态的数据；如果把 I0enable 参数置 3 时，SaveADCBuffer 指针数据中也返回带 2 通道 I02 和 I03 状态的数据。

切记调试此函数要注意读取缓冲 buffer 长度设置，如读 100 个采样点，设置 I0enable 参数为 3 时，此时 buffer 长度一定比平时多 2 个 I0 状态通道数据；设置 I0enable 参数为 1 或 2 时，此时 buffer 长度一定比平时多 1 个 I0 状态通道数据；其它设置不变。

 深圳微精艺电子有限公司 www.vkinging.com	Product:VK70xN 数据采集器 Revision: 6.07 Issued Date: 2022-10-21
DLL 函数使用手册 (自带 TCP 服务器解析多卡连接数据)	Sheet 29 of 68

4.14 其它说明

【数据格式】

VK70xNMC_GetOneChannel 读取采样数据的缓冲区存取格式如下：

采样点 1	采样点 2	...	采样点 N
读取对应的通道	读取对应的通道	...	读取对应的通道

VK70xNMC_GetFourChannel 读取采样数据的缓冲区存取格式如下（适用于 VK701N, VK701S）：

采样点 1				采样点 2				...	采样点 N			
通道 1	通道 2	通道 3	通道 4	通道 1	通道 2	通道 3	通道 4	...	通道 1	通道 2	通道 3	通道 4

VK70xNMC_GetAllChannel 读取采样数据的缓冲区存取格式如下（适用于 VK702N）：

采样点 1				采样点 2				...	采样点 N			
通道 1	通道 2	...	通道 8	通道 1	通道 2	...	通道 8	...	通道 1	通道 2	...	通道 8

VK70xNMC_GetOneChannel_WithIOStatus 读取采样数据的缓冲区存取格式如下 (IOEnable=1)：

采样点 1				采样点 2				...	采样点 N			
读取对应的通道		I02 状态		读取对应的通道		I02 状态		...	读取对应的通道		I02 状态	

VK70xNMC_GetFourChannel_WithIOStatus 读取采样数据的缓冲区存取格式如下 (IOEnable=1)：

采样点 1				采样点 2				...	采样点 N			
通道 1	...	通道 4	I02 状态	通道 1	...	通道 4	I02 状态	...	通道 1	...	通道 4	I02 状态

VK70xNMC_GetAllChannel_WithIOStatus 读取采样数据的缓冲区存取格式如下 (IOEnable=1)：

采样点 1				采样点 2				...	采样点 N			
通道	...	通道	I02 状	通道	...	通道	I02 状	...	通道	...	通道	I02 状

1		8	态	1		8	态		1		8	态
---	--	---	---	---	--	---	---	--	---	--	---	---

VK70xNMC_GetOneChannel_WithIOStatus 读取采样数据的缓冲区存取格式如下 (IOEnable=2):

采样点 1				采样点 2				...	采样点 N			
读取对应的通道		I03 状态		读取对应的通道		I03 状态		...	读取对应的通道		I03 状态	

VK70xNMC_GetFourChannel_WithIOStatus 读取采样数据的缓冲区存取格式如下 (IOEnable=2):

采样点 1				采样点 2				...	采样点 N			
通道 1	...	通道 4	I03 状态	通道 1	...	通道 4	I03 状态	...	通道 1	...	通道 4	I03 状态

VK70xNMC_GetAllChannel_WithIOStatus 读取采样数据的缓冲区存取格式如下 (IOEnable=2):

采样点 1				采样点 2				...	采样点 N			
通道 1	...	通道 8	I03 状态	通道 1	...	通道 8	I03 状态	...	通道 1	...	通道 8	I03 状态

VK70xNMC_GetOneChannel_WithIOStatus 读取采样数据的缓冲区存取格式如下 (IOEnable=3):

采样点 1				采样点 2				...	采样点 N			
读取对应的通道		I02 状态	I03 状态	读取对应的通道		I02 状态	I03 状态	...	读取对应的通道		I02 状态	I03 状态

VK70xNMC_GetFourChannel_WithIOStatus 读取采样数据的缓冲区存取格式如下 (IOEnable=3):

采样点 1					采样点 2					...	采样点 N				
通道 1	...	通道 4	I02 状态	I03 状态	通道 1	...	通道 4	I02 状态	I03 状态	...	通道 1	...	通道 4	I02 状态	I03 状态

VK70xNMC_GetAllChannel_WithIOStatus 读取采样数据的缓冲区存取格式如下 (IOEnable=3):

采样点 1					采样点 2					...	采样点 N				
通道	...	通道	I02 状	I03 状	通道	...	通道	I02 状	I03 状	...	通道	...	通道	I02 状	I03 状

 深圳微精艺电子有限公司 www.vkinging.com	Product:VK70xN 数据采集器 Revision: 6.07 Issued Date: 2022-10-21
DLL 函数使用手册 (自带 TCP 服务器解析多卡连接数据)	Sheet 31 of 68

1		8	态	态	1		8	态	态		1		8	态	态
---	--	---	---	---	---	--	---	---	---	--	---	--	---	---	---

【VK702N、VK701S 电压输入范围】

设置参数	电压输入范围说明
0	-10V~10V
1	-5V~5V
2	-2.5V~2.5V
3	-1V~1V
4	-500mV~500mV

【VK701N 电压输入范围】

设置参数	电压输入范围说明
0	-10V~10V
1	-5V~5V
2	-2.5V~2.5V
3	-1V~1V
4	-500mV~500mV
5	-100mV~100mV
6	-20mV~20mV
7	-1mV~1mV

 深圳微精艺电子有限公司 www.vkinging.com	Product:VK70xN 数据采集器 Revision: 6.07 Issued Date: 2022-10-21
DLL 函数使用手册 (自带 TCP 服务器解析多卡连接数据)	Sheet 32 of 68

5. 附加功能函数说明

5.1 设置 PWM/DAC/IO/计数器/温度通道

【函数原型】

int VK70xNMC_Set_AdditionalFeature(int mci,int funcNo,int para1,double para2);

输入：设置参数说明如下表。

参数名称	参数类型	参数描述
int mci	已连接到服务器上的采集卡序号	范围：0-7。 注意：目前库函数版本仅仅支持最大连接采集卡 8 个。
int funcNo	功能参数	定义见下表(注：11 表示十进制，0x11 表示 16 进制)
int para1	参数 1	定义见下表
double para2	参数 2	定义见下表

名称	功能参数： int funcNo	参数 1： int para1	参数 2： double para2
PWM	1	PWM 通道 1 频率 0~1000 000Hz	PWM 通道 1 占空比，百分比是 0.1%~100.0%
	2	PWM 通道 2 频率 0~1000 000Hz	PWM 通道 2 占空比，百分比是 0.1%~100.0%
DAC	11	无意义	对应电压值 0~3.3V
	12	无意义	对应电压值 0~3.3V
IO	21	IO1 电平, 0 为低, 1 为高, 2 为输入, 0xFF 无效	无意义
	22	IO2 电平, 0 为低, 1 为高, 2 为输入, 0xFF 无效	无意义
	23	IO3 电平, 0 为低, 1 为高, 2 为输入, 0xFF 无效	无意义
	24	IO4 电平, 0 为低, 1 为高, 2 为输入, 0xFF 无效	无意义
计数器	31	0-复位计数器	无意义
温度	41	0-4: 设置温度通道, 0 为采集卡内部通道	无意义
保留	其它数值	无意义	无意义

 深圳微精艺电子有限公司 www.vkinging.com	Product:VK70xN 数据采集器 Revision: 6.07 Issued Date: 2022-10-21
DLL 函数使用手册 (自带 TCP 服务器解析多卡连接数据)	Sheet 33 of 68

输出：返回状态说明如下表。

返回数值	状态说明
0 或大于 0	设置成功
-11	服务器未打开
-12	采集卡未连接或不存在
-13	采集卡未连接或不存在
其它小于 0 的数值	异常退出

【VC/C 调用例程】

```
extern "C" int VK70xNMC_Set_AdditionalFeature(int mci,int funcNo,int
para1,double para2)
```

```
Void SetupVK70xNMC_AdditionalFeature(void)
{
    int rstatus;

    rstatus= VK70xNMC_Set_AdditionalFeature(0,1,10000,50.5);// PWM 1 ,
10KHZ,50.5%
    rstatus= VK70xNMC_Set_AdditionalFeature(0,2,10000,50.5);// PWM 2 ,
10KHZ,50.5%

    rstatus= VK70xNMC_Set_AdditionalFeature(0,11,0,0.25);// DAC 1 , 0.25V
    rstatus= VK70xNMC_Set_AdditionalFeature(0,12,0,1.56);// DAC 2 , 1.56V

    rstatus= VK70xNMC_Set_AdditionalFeature(0,21,1,0);// IO1=high
    rstatus= VK70xNMC_Set_AdditionalFeature(0,22,0,0);// IO2=low
    rstatus= VK70xNMC_Set_AdditionalFeature(0,23,2,0);// IO3= 输入
    rstatus= VK70xNMC_Set_AdditionalFeature(0,24,1,0);// IO4= high

    rstatus= VK70xNMC_Set_AdditionalFeature(0,31,0,0);// 计数器清零

    rstatus= VK70xNMC_Set_AdditionalFeature(0,41,0,0);// 选择内部温度通道

    if (rstatus >=0) printf("%s", '操作成功!');
    else if (tstatus == -11)printf("服务器未打开!");
    else if (tstatus == -12 || tstatus == -13)printf("采集卡未连接或不存在!");
    else printf("%s", '异常退出!');
}
```

 深圳微精艺电子有限公司 www.vkinging.com	Product:VK70xN 数据采集器 Revision: 6.07 Issued Date: 2022-10-21
DLL 函数使用手册（自带 TCP 服务器解析多卡连接数据）	Sheet 34 of 68

5.2 读取 IO 状态

【函数原型】

```
int VK70xNMC_Get_AllIOS_Blocking(int mci, int *iobuffer, int timeout);  
int VK70xNMC_Get_IO1_Blocking(int mci, int *iovalue, int timeout);  
int VK70xNMC_Get_IO2_Blocking(int mci, int *iovalue, int timeout);  
int VK70xNMC_Get_IO3_Blocking(int mci, int *iovalue, int timeout);  
int VK70xNMC_Get_IO4_Blocking(int mci, int *iovalue, int timeout);
```

输入：设置参数说明如下表。

参数名称	参数描述	范围
int mci	已连接到服务器上的采集卡序号	范围：0-7。 注意：目前库函数版本仅仅支持最大连接采集卡 8 个。
int *iovalue 或 int * iobuffer	整型指针	存放 IO 状态地址，需要提供数组长度为 4 或 1
int timeout	整型指针	读取等待时间，单位为毫秒，如 200 相当于在函数内延时等待 200ms 读取结果。

输出：返回状态说明如下表。

返回数值	状态说明
大于或等于 0	成功读取 IO 状态
-1	读取失败，IO 状态仍未更新，请延时 100ms 后再尝试
-11	服务器未打开
-12	采集卡未连接或不存在
-13	采集卡未连接或不存在
其它小于 0 的数值	异常退出

【VC/C 调用例程】

```
extern "C" int VK70xNMC_Get_AllIOS_Blocking(int mci, int *iobuffer, int timeout);  
extern "C" int VK70xNMC_Get_IO1_Blocking(int mci, int *iovalue, int timeout);  
extern "C" int VK70xNMC_Get_IO2_Blocking(int mci, int *iovalue, int timeout);  
extern "C" int VK70xNMC_Get_IO3_Blocking(int mci, int *iovalue, int timeout);  
extern "C" int VK70xNMC_Get_IO4_Blocking(int mci, int *iovalue, int timeout);
```

 深圳微精艺电子有限公司 www.vkinging.com	Product:VK70xN 数据采集器 Revision: 6.07 Issued Date: 2022-10-21
DLL 函数使用手册（自带 TCP 服务器解析多卡连接数据）	Sheet 35 of 68

```
Void Read_I01_I04_Status(void)
{
    int rstatus;
    int IOStatus[4];
    //=====
    //Rstatus = VK70xNMC_Get_I01_Blocking (0, &IOStatus[0],200); //延时 200ms
    //Rstatus = VK70xNMC_Get_I02_Blocking (0, &IOStatus[1],200);
    //Rstatus = VK70xNMC_Get_I03_Blocking (0, &IOStatus[2],200);
    //Rstatus = VK70xNMC_Get_I04_Blocking (0, &IOStatus[3],200);
    Rstatus = VK70xNMC_Get_AllIOS_Blocking(0, IOStatus,200);
    if (rstatus ==1) printf("成功读取 IO 状态: I01=%d; I02=%d; I03=%d; I04=%d;",
IOStatus[0], IOStatus[1], IOStatus[2], IOStatus[3]);
    else if (rstatus ==0)printf("IO 状态未更新, 请延时尝试多次读取!");
    else if (tstatus == -11){printf("服务器未打开!");return;}
    else if (tstatus == -12 || tstatus == -13) {printf("采集卡未连接或不存在!");return;}
    else {printf("异常退出!");return;}
}
```

5.2.1 读取 PWM 参数

【函数原型】

```
int VK70xNMC_Get_PWM_Blocking(int mci, double *dutyval, int *freqval,int timeout);
```

输入：设置参数说明如下表。

参数名称	参数描述	范围
int mci	已连接到服务器上的采集卡序号	范围：0-7。 注意：目前库函数版本仅仅支持最大连接采集卡 8 个。
double *dutyval,	双精度浮点指针	存放 PWM 占空比地址, 需要提供数组长度为 2
int *freqval	无符号整型指针	存放 PWM 频率地址, 需要提供数组长度为 2
int timeout	整型指针	读取等待时间, 单位为毫秒, 如 200 相当于在函数内延时等待 200ms 读取结果。

输出：返回状态说明如下表。

返回数值	状态说明
大于或等于 0	成功读取 PWM 参数
-1	读取失败, PWM 参数仍未更新, 请延时 100ms 后再尝试

 深圳微精艺电子有限公司 www.vkinging.com	Product:VK70xN 数据采集器 Revision: 6.07 Issued Date: 2022-10-21
DLL 函数使用手册 (自带 TCP 服务器解析多卡连接数据)	Sheet 36 of 68

-11	服务器未打开
-12	采集卡未连接或不存在
-13	采集卡未连接或不存在
其它小于 0 的数值	异常退出

【VC/C 调用例程】

```
extern "C" int VK70xNMC_Get_PWM_Blocking(int mci, double *dutyval, int *freqval, int timeout);
```

```
Void Read_PWM_Parameter(void)
{
    int rstatus;
    int freq[2];
    double      tduty[2];
    Rstatus = VK70xNMC_Get_PWM_Blocking (0, tduty, freq, 200); //延时 200ms
    if (rstatus ==1) printf("成功读取 PWM 参数: PWM1 频率=%d HZ; PWM1 占空比=%lf%%; PWM2 频率=%d HZ; PWM2 占空比=%lf%%;", freq[0], tduty[0], freq[1], tduty[1]);
    else if (rstatus ==0)printf("PWM 参数未更新, 请延时尝试多次读取!");
    else if (tstatus == -11){printf("服务器未打开!");return;}
    else if (tstatus == -12 || tstatus == -13) {printf("采集卡未连接或不存在!");return;}
    else {printf("异常退出!");return;}
}
```

5.2.2 读取 DAC 参数

【函数原型】

```
int VK70xNMC_Get_DAC_Blocking(int mci, double *dacval, int timeout);
```

输入：设置参数说明如下表。

参数名称	参数描述	范围
int mci	已连接到服务器上的采集卡序号	范围：0-7。 注意：目前库函数版本仅仅支持最大连接采集卡 8 个。
double *dacval,	双精度浮点指针	存放 DAC 电压值地址, 需要提供数组长度为 2
int timeout	整型指针	读取等待时间, 单位为毫秒, 如 200 相当于在函数内延时等待 200ms 读取结果。

 深圳微精艺电子有限公司 www.vkinging.com	Product:VK70xN 数据采集器 Revision: 6.07 Issued Date: 2022-10-21
DLL 函数使用手册 (自带 TCP 服务器解析多卡连接数据)	Sheet 37 of 68

输出：返回状态说明如下表。

返回数值	状态说明
大于或等于 0	成功读取 DAC 参数
-1	读取失败，DAC 参数仍未更新，请延时 100ms 后再尝试
-11	服务器未打开
-12	采集卡未连接或不存在
-13	采集卡未连接或不存在
其它小于 0 的数值	异常退出

【VC/C 调用例程】

```
extern "C" int VK70xNMC_Get_DAC_Blocking (int mci, double *dacval, int timeout);
```

```
Void Read_DAC_Parameter(void)
{
    int rstatus;
    double tDAC[2];
    Rstatus = VK70xNMC_Get_DAC_Blocking(0, tDAC, 200); //延时 200ms

    if (rstatus == 1) printf("成功读取 DAC 参数: DAC1=%lf; DAC2=% lf ", tDAC[0],
tDAC[1]);
    else if (rstatus == 0) printf("DAC 参数未更新，请延时尝试多次读取！");
    else if (tstatus == -11) {printf("服务器未打开！"); return;}
    else if (tstatus == -12 || tstatus == -13) {printf("采集卡未连接或不存在！"); return;}
    else {printf("异常退出！"); return;}
}
```

5.2.3 读取计数器

【读取计数器方法】

目前采集卡仅仅 I04 支持外部中断计数器，需要先使用

```
VK70xNMC_Set_AdditionalFeature(int mci, unsigned int funcNo, unsigned int para1, double para2);
```

设置 I04 作为外部中断计数模式，函数调用方法为：

```
VK70xNMC_Set_AdditionalFeature(int mci, 31, 0, 0); //mci 为已连接采集卡序号
```

【函数原型】

```
int VK70xNMC_Get_Counter_Blocking(int mci, int *countervalue, int timeout);
```

 深圳微精艺电子有限公司 www.vkinging.com	Product:VK70xN 数据采集器 Revision: 6.07 Issued Date: 2022-10-21
DLL 函数使用手册 (自带 TCP 服务器解析多卡连接数据)	Sheet 38 of 68

输入：设置参数说明如下表。

参数名称	参数描述	范围
int mci	已连接到服务器上的采集卡序号	范围：0-7。 注意：目前库函数版本仅仅支持最大连接采集卡 8 个。
int *countervalue	无符号整型指针	存放计数器值地址, 需要提供数组长度为 1
int timeout	符号整型, 等待时间	最大阻塞等待时间 (ms) 0 或小于 0: 无阻塞返回; 大于 0: 最大等待时长, 单位毫秒

输出：返回状态说明如下表。

返回数值	状态说明
大于或等于 0	成功读取计数器值
-1	读取失败, 计数器值仍未更新, 请延时 100ms 后再尝试
-11	服务器未打开
-12	采集卡未连接或不存
-13	采集卡未连接或不存
其它小于 0 的数值	异常退出

【VC/C 调用例程】

```
extern "C" int VK70xNMC_Get_Counter_Blocking(int mci, int *countervalue, int
timeout);
Void Read_Counter(void)
{
    int rstatus;
    unsigned int tcounter;

    Rstatus = VK70xNMC_Get_Counter_Blocking(0, tcounter, 200); // 最大等待时间
    200ms
    if (rstatus == 1) printf("成功读取计数器值: %d ", tcounter);
    else if (rstatus == 0) printf("计数器值未更新, 请延时尝试多次读取!");
    else if (tstatus == -11) {printf("服务器未打开!"); return;}
    else if (tstatus == -12 || tstatus == -13) {printf("采集卡未连接或不存
"); return;}
    else {printf("异常退出!"); return;}
}
```

 深圳微精艺电子有限公司 www.vkinging.com	Product:VK70xN 数据采集器 Revision: 6.07 Issued Date: 2022-10-21
DLL 函数使用手册 (自带 TCP 服务器解析多卡连接数据)	Sheet 39 of 68

5.2.4 读取温度值

【读取温度值方法】

读取温度值之前，需要先使用

VK70xNMC_Set_AdditionalFeature(int mci, int funcNo, int para1, double para2);
 设置正确的传感器通道值，函数调用方法为：

函数调用	说明
VK70xNMC_Set_AdditionalFeature(int mci, 41, 0, 0);	选择采集卡内部通道温度传感器
VK70xNMC_Set_AdditionalFeature(int mci, 41, 1, 0);	选择 IO1 作为温度传感器通道
VK70xNMC_Set_AdditionalFeature(int mci, 41, 2, 0);	选择 IO2 作为温度传感器通道
VK70xNMC_Set_AdditionalFeature(int mci, 41, 3, 0);	选择 IO3 作为温度传感器通道
VK70xNMC_Set_AdditionalFeature(int mci, 41, 4, 0);	选择 IO4 作为温度传感器通道

【函数原型】

int VK70xNMC_Get_Temperature_Blocking(int mci, int *tempvalue, int timeout);

输入：设置参数说明如下表。

参数名称	参数描述	范围
int mci	已连接到服务器上的采集卡序号	范围：0-7。注意：目前库函数版本仅仅支持最大连接采集卡 8 个。
int * tempvalue	无符号整型指针	存放温度值地址，需要提供数组长度为 1
int timeout	符号整型，等待时间	最大阻塞等待时间（ms） 0 或小于 0：无阻塞返回； 大于 0：最大等待时长，单位毫秒

输出：返回状态说明如下表。

返回数值	状态说明
大于或等于 0	成功读取温度值
-1	读取失败，温度值仍未更新，请延时 100ms 后再尝试
-11	服务器未打开
-12	采集卡未连接或不存
-13	采集卡未连接或不存
其它小于 0 的数值	异常退出

【VC/C 调用例程】

 深圳微精艺电子有限公司 www.vkinging.com	Product:VK70xN 数据采集器 Revision: 6.07 Issued Date: 2022-10-21
DLL 函数使用手册 (自带 TCP 服务器解析多卡连接数据)	Sheet 40 of 68

```
extern "C" int VK70xNMC_Get_Temperature_Blocking (int mci, int *tempvalue, int
timeout);
```

```
Void Read_Temperature(void)
```

```
{
```

```
    int rstatus;
```

```
    int temperature;
```

```
    Rstatus = VK70xNMC_Get_Temperature_Blocking(0, temperature, 200);// 最大等
待时间 200ms
```

```
    if (rstatus ==1) printf("成功读取温度值: %d ", temperature);
```

```
    else if (rstatus ==0)printf("温度值未更新, 请延时尝试多次读取!");
```

```
    else if (tstatus == -11) {printf("服务器未打开!");return;}
```

```
    else if (tstatus == -12 || tstatus == -13) {printf("采集卡未连接或不存在!
");return;}
```

```
    else {printf("异常退出!") ;return;}
```

```
}
```


 深圳微精艺电子有限公司 www.vkinging.com	Product:VK70xN 数据采集器 Revision: 6.07 Issued Date: 2022-10-21
DLL 函数使用手册 (自带 TCP 服务器解析多卡连接数据)	Sheet 41 of 68

6. SD 卡文件系统函数说明

6.1 进入 SD 卡文件下载模式

【函数原型】

`int VK70xNMC_Enter_SDFileSystem(int mci);`

输入：设置参数说明如下表。

参数名称	参数描述	设置范围
int mci	已连接到服务器上的采集卡序号	范围：0-7。 注意：目前库函数版本仅仅支持最大连接采集卡 8 个。

输出：返回状态说明如下表。

返回数值	状态说明
0 或大于 0	设置成功
-11	服务器未打开
-12	采集卡未连接或不存在
-13	采集卡未连接或不存在
其它小于 0 的数值	异常退出

【VC/C 调用例程】

extern "C" `int VK70xNMC_Enter_SDFileSystem(int mci);`

```

Void Enter_VK70xNMC_SystemMode(void)
{
    int rstatus;
    rstatus = VK70xNMC_Enter_SDFileSystem (0);

    if (rstatus >=0) printf( "%s", '操作成功!' );
    else if (tstatus == -11)printf( "服务器未打开!" );
    else if (tstatus == -12)printf( "采集卡未连接或不存在!" );
    else printf( "%s", '异常退出!' );
}

```

 深圳微精艺电子有限公司 www.vkinging.com	Product:VK70xN 数据采集器 Revision: 6.07 Issued Date: 2022-10-21
DLL 函数使用手册 (自带 TCP 服务器解析多卡连接数据)	Sheet 42 of 68

6.2 退出 SD 卡文件下载模式

【函数原型】

```
int VK70xNMC_Exit_SDFileSystem(int mci);
```

输入：设置参数说明如下表。

参数名称	参数描述	设置范围
int mci	已连接到服务器上的采集卡序号	范围：0-7。 注意：目前库函数版本仅仅支持最大连接采集卡 8 个。

输出：返回状态说明如下表。

返回数值	状态说明
0 或大于 0	设置成功
-11	服务器未打开
-12	采集卡未连接或不存在
-13	采集卡未连接或不存在
其它小于 0 的数值	异常退出

【VC/C 调用例程】

```
extern "C" int VK70xNMC_Exit_SDFileSystem(int mci);
```

```
Void Exit_VK70xNMC_SystemMode(void)
{
    int rstatus;
    rstatus = VK70xNMC_Exit_SDFileSystem (0);

    if (rstatus >=0) printf( "%s", '操作成功!' );
    else if (tstatus == -11)printf( "服务器未打开!" );
    else if (tstatus == -12)printf( "采集卡未连接或不存在!" );
    else printf( "%s", '异常退出!' );
}
```

 深圳微精艺电子有限公司 www.vkinging.com	Product:VK70xN 数据采集器 Revision: 6.07 Issued Date: 2022-10-21
DLL 函数使用手册（自带 TCP 服务器解析多卡连接数据）	Sheet 43 of 68

6.3 读取文件目录

重要提示：如果读取成功后，

（1）将在文件夹“\vk70xnsdfiles”目录下自动保存“vk70xn_sdfile_dir_information.txt”文件，用户可查看采集卡的文件名称，文件大小，sd 容量，sd 卡使用状态，sd 卡存储文件数量。

（2）也可以调用函数 VK70xNMC_Get_SDFileRuningInformation(int mci, int *para) 查询 sd 卡的空间大小，文件数量和状态。

【函数原型】

```
int VK70xNMC_Get_SDFileDIR(int mci);
```

输入：设置参数说明如下表。

参数名称	参数描述	设置范围
int mci	已连接到服务器上的采集卡序号	范围：0-7。注意：目前库函数版本仅仅支持最大连接采集卡 8 个。

输出：返回状态说明如下表。

返回数值	状态说明
0 或大于 0	设置成功
-11	服务器未打开
-12	采集卡未连接或不存在
-13	采集卡未连接或不存在
其它小于 0 的数值	异常退出

【VC/C 调用例程】

```
extern "C" int VK70xNMC_Get_SDFileDIR(int mci);
```

```
Void Get_VK70xNMC_SDFileDIR(void)
{
    int rstatus;
    rstatus = VK70xNMC_Get_SDFileDIR(0);

    if (rstatus >= 0) printf(“%s”, ‘操作成功!’);
    else if (tstatus == -11) printf(“服务器未打开!”);
    else if (tstatus == -12) printf(“采集卡未连接或不存在!”);
    else printf(“%s”, ‘异常退出!’);
}
```

 深圳微精艺电子有限公司 www.vkinging.com	Product:VK70xN 数据采集器 Revision: 6.07 Issued Date: 2022-10-21
DLL 函数使用手册 (自带 TCP 服务器解析多卡连接数据)	Sheet 44 of 68

6.4 查询 SD 卡状态

【函数原型】

int VK70xNMC_Get_SDFileRuningInformation(int mci, int *para);

输入：设置参数说明如下表。

参数名称	参数描述	读取参数描述
int mci	已连接到服务器上的采集卡序号	范围：0-7。注意：目前库函数版本仅仅支持最大连接采集卡 8 个。
int *para	用于存取 sd 容量大小，sd 卡存放文件数量，sd 正在读取文件数量进度的整形指针	[Para 0]：SD 卡容量大小 [Para 1]：>0, 正在读取 sd 文件计数器；0x00-无操作或读取完成或未同步 [Para 2]：>0, SD 卡文件数量；0x00-sd 卡无文件或未同步 [Para 3]：0x01-正在下载或读取或删除中；0x00-操作完成、空闲状态 [Para 4]：0x02-正在连续下载文件中；0x00 或其它-其它操作

输出：返回状态说明如下表。

返回数值	状态说明
0 或大于 0	设置成功
-7	当前采集卡未工作 SD 下载模式或未同步 sd 卡目录信息
-11	服务器未打开
-12	采集卡未连接或不存
-13	采集卡未连接或不存
其它小于 0 的数值	异常退出

【VC/C 调用例程】

```
extern "C" int VK70xNMC_Get_SDFileRuningInformation(int mci, int *para);
```

```
Void Get_VK70xNMC_SDFileRuningInformation(void)
```

```
{
```

```
int rstatus;
```

```
int tpara[8];
```

```
rstatus = VK70xNMC_Get_SDFileRuningInformation(0, tpara);//
```

```
if (rstatus >=0) printf( "%s", '操作成功! ');
```

```
else if (tstatus == -7)printf( "当前采集卡未工作 SD 下载模式或未同步 sd 卡目录信息! ");
```

```
else if (tstatus == -11)printf( "服务器未打开! ");
```

```
else if (tstatus == -12)printf( "采集卡未连接或不存! ");
```

 深圳微精艺电子有限公司 www.vkinging.com	Product:VK70xN 数据采集器 Revision: 6.07 Issued Date: 2022-10-21
DLL 函数使用手册 (自带 TCP 服务器解析多卡连接数据)	Sheet 45 of 68

```

    else printf( "%s", '异常退出! ');
  }

```

6.5 读取一个采样记录文件

重要提示:

- (1) 如果未同步 SD 目录, 需要先调用 VK70xNMC_Get_SDFileDIR 函数同步一次 SD 文件目录信息。
- (2) 如果读取成功后, 相关文件将自动存放到文件夹 “\vk702hnsdfiles” 中。

【函数原型】

```
int VK70xNMC_Get_OneSDFile(int mci, int fileindex);
```

输入: 设置参数说明如下表。

参数名称	参数描述	设置范围
int mci	已连接到服务器上的采集卡序号	范围: 0-7。注意: 目前库函数版本仅仅支持最大连接采集卡 8 个。
int fileindex	Sd 文件索引	读取 sd 卡第【fileindex】个文件, 此索引一定小于等于文件总数量且大于 0。

输出: 返回状态说明如下表。

返回数值	状态说明
0 或大于 0	设置成功
-5	未同步 sd 卡目录信息
-6	选择文件出错, 文件索引大于文件总数量
-11	服务器未打开
-12	采集卡未连接或不存在
-13	采集卡未连接或不存在
其它小于 0 的数值	异常退出

【VC/C 调用例程】

```
extern "C" int VK70xNMC_Get_OneSDFile (int mci, int fileindex);
```

```

Void Get_VK70xNMC_OneSDFile (void)
{
    int rstatus;
    rstatus = VK70xNMC_Get_OneSDFile(0, 1);

    if (rstatus >=0) printf( "%s", '操作成功! ');
    else if (tstatus == -5)printf( "当前采集卡未同步 sd 卡目录信息! ");
    else if (tstatus == -6)printf( "选择文件出错, 文件索引大于文件总数量" );
    else if (tstatus == -11)printf( "服务器未打开! ");

```

 深圳微精艺电子有限公司 www.vkinging.com	Product:VK70xN 数据采集器 Revision: 6.07 Issued Date: 2022-10-21
DLL 函数使用手册（自带 TCP 服务器解析多卡连接数据）	Sheet 46 of 68

```

else if (tstatus == -12)printf(“采集卡未连接或不存在!”);
else printf(“%s”, ‘异常退出!’);
}

```

6.6 读取所有采样文件

重要提示：

- (1) 如果未同步 SD 目录，需要先调用 VK70xNMC_Get_SDFileDIR 函数同步一次 SD 文件目录信息。
- (2) 如果读取成功后，相关文件将自动存放到文件夹“\vk702hnsdfiles”中。

【函数原型】

```
int VK70xNMC_Get_MultiSDFiles(int mci);
```

输入：设置参数说明如下表。

参数名称	参数描述	设置范围
int mci	已连接到服务器上的采集卡序号	范围：0-7。注意：目前库函数版本仅支持最大连接采集卡 8 个。

输出：返回状态说明如下表。

返回数值	状态说明
0 或大于 0	设置成功
-5	未同步 sd 卡目录信息
-11	服务器未打开
-12	采集卡未连接或不存在
-13	采集卡未连接或不存在
其它小于 0 的数值	异常退出

【VC/C 调用例程】

```
extern “C” int VK70xNMC_Get_MultiSDFiles(int mci);
```

```

Void Get_VK70xNMC_MultiSDFiles(void)
{
    int rstatus;
    rstatus = VK70xNMC_Get_MultiSDFiles(0);

    if (rstatus >=0) printf(“%s”, ‘操作成功!’);
    else if (tstatus == -5)printf(“当前采集卡未同步 sd 卡目录信息!”);
    else if (tstatus == -11)printf(“服务器未打开!”);
    else if (tstatus == -12)printf(“采集卡未连接或不存在!”);
    else printf(“%s”, ‘异常退出!’);
}

```

 深圳微精艺电子有限公司 www.vkinging.com	Product:VK70xN 数据采集器 Revision: 6.07 Issued Date: 2022-10-21
DLL 函数使用手册 (自带 TCP 服务器解析多卡连接数据)	Sheet 47 of 68

6.7 删除一个采样记录文件

重要提示:

- (1) 如果未同步 SD 目录, 需要先调用 VK70xNMC_Get_SDFileDIR 函数同步一次 SD 文件目录信息。
- (2) 删除文件后, 需调用 VK70xNMC_Get_SDFileDIR 函数同步 SD 目录信息。

【函数原型】

```
int VK70xNMC_Delete_OneSDFile(int mci, int fileindex);
```

输入: 设置参数说明如下表。

参数名称	参数描述	设置范围
int mci	已连接到服务器上的采集卡序号	范围: 0-7。注意: 目前库函数版本仅仅支持最大连接采集卡 8 个。
int fileindex	Sd 文件索引	删除 sd 卡第【fileindex】个文件, 此索引一定小于等于文件总数量且大于 0。

输出: 返回状态说明如下表。

返回数值	状态说明
0 或大于 0	设置成功
-5	未同步 sd 卡目录信息
-6	选择文件出错, 文件索引大于文件总数量
-11	服务器未打开
-12	采集卡未连接或不存
-13	采集卡未连接或不存
其它小于 0 的数值	异常退出

【VC/C 调用例程】

```
extern "C" int VK70xNMC_Delete_OneSDFile (int mci, int fileindex);
```

```
Void Delete_VK70xNMC_OneSDFile(void)
{
    int rstatus;
    rstatus = VK70xNMC_Delete_OneSDFile(0, 1);

    if (rstatus >= 0) printf("操作成功!");
    else if (rstatus == -5) printf("当前采集卡未同步 sd 卡目录信息!");
    else if (rstatus == -6) printf("选择文件出错, 文件索引大于文件总数量");
    else if (rstatus == -11) printf("服务器未打开!");
    else if (rstatus == -12) printf("采集卡未连接或不存!");
    else printf("%s", "异常退出!");
}
```

 深圳微精艺电子有限公司 www.vkinging.com	Product:VK70xN 数据采集器 Revision: 6.07 Issued Date: 2022-10-21
DLL 函数使用手册 (自带 TCP 服务器解析多卡连接数据)	Sheet 48 of 68

6.8 删除所有采样记录文件

重要提示：

(1) 如果未同步 SD 目录，需要先调用 VK70xNMC_Get_SDFileDIR 函数同步一次 SD 文件目录信息。

(2) 删除文件后，需调用 VK70xNMC_Get_SDFileDIR 函数同步 SD 目录信息。

【函数原型】

```
int VK70xNMC_Delete_MultiSDFiles(int mci);
```

输入：设置参数说明如下表。

参数名称	参数描述	设置范围
int mci	已连接到服务器上的采集卡序号	范围：0-7。注意：目前库函数版本仅支持最大连接采集卡 8 个。

输出：返回状态说明如下表。

返回数值	状态说明
0 或大于 0	设置成功
-5	未同步 sd 卡目录信息
-6	选择文件出错，文件索引大于文件总数量
-11	服务器未打开
-12	采集卡未连接或不存
-13	采集卡未连接或不存
其它小于 0 的数值	异常退出

【VC/C 调用例程】

```
extern "C" int VK70xNMC_Delete_MultiSDFiles(int mci);
```

```
Void Delete_VK70xNMC_MultiSDFiles(void)
{
    int rstatus;
    rstatus = VK70xNMC_Delete_MultiSDFiles(0);

    if (rstatus >=0) printf(“%s”, ‘操作成功!’);
    else if (tstatus == -5)printf(“当前采集卡未同步 sd 卡目录信息!”);
    else if (tstatus == -11)printf(“服务器未打开!”);
    else if (tstatus == -12)printf(“采集卡未连接或不存!”);
    else printf(“%s”, ‘异常退出!’);
}
```


 深圳微精艺电子有限公司 www.vkinging.com	Product:VK70xN 数据采集器 Revision: 6.07 Issued Date: 2022-10-21
DLL 函数使用手册 (自带 TCP 服务器解析多卡连接数据)	Sheet 49 of 68

6.9 设置下载文件缺省存储路径

重要提示:

- (1) 打开根目录下缺省文件 “vk70xmcsdconfig.ini”，输入相对应存储文件路径即可，设置完成，重启软件。需要注意一点，输入缺省文件路径一定要有效，如果电脑中不存的路径，请手动创建好文件夹再设置，否则会导致 DLL 奔溃！

【函数原型】

```
int VK70xNMC_Set_SaveFileDeafultPath(int tflag, const char *defaultdir);
```

输入：设置参数说明如下表。

参数名称	参数描述	设置范围
int tflag	绑定和解绑标志	范围 0-1: 0x00: 使用缺省工作路径 0x01: 自定义路径
const char *defaultdir	自定义缺省存储文件路径	“D:\\sdcardfiles\\0”

输出：返回状态说明如下表。

返回数值	状态说明
0 或大于 0	服务器关闭成功
-11	服务器未打开
-12	服务器端口不存在
其它小于 0 的数值	异常退出

【VC/C 调用例程】

```
extern “C” int VK70xNMC_Set_SaveFileDeafultPath(int tflag, const char *defaultdir);
```

```
void Set_SaveSDFileDeafultPath(void)
{
    Int tstatus=0;
    //tstatus = VK70xNMC_Set_SaveFileDeafultPath(0, NULL); //同工作目录
    tstatus = VK70xNMC_Set_SaveFileDeafultPath(1, “D:\\sdcardfiles\\0”);

    if (tstatus >= 0)printf(“操作成功!”);
    else if (tstatus == -11)printf(“服务器未打开!”);
    else if (tstatus == -12)printf(“服务器端口不存在!”);
    else printf(“服务器已打开!”);
}
```

 深圳微精艺电子有限公司 www.vkinging.com	Product:VK70xN 数据采集器 Revision: 6.07 Issued Date: 2022-10-21
DLL 函数使用手册 (自带 TCP 服务器解析多卡连接数据)	Sheet 50 of 68

7. 系统设置函数说明

7.1 系统模式

7.1.1 切换系统模式

【函数原型】

`int VK70xNMC_Set_SystemMode(int mci, int sysmodeval, int samplemethod, int sdfilefmt)`

输入：设置参数说明如下表。

参数名称	参数描述	设置范围
<code>int mci</code>	已连接到服务器上的采集卡序号	范围：0-7。 注意：目前库函数版本仅仅支持最大连接采集卡 8 个。
<code>int sysmodeval</code>	选择系统模式 (*)	0x00/0x80/0x81: LAN/WIFI 实时传输模式 0x01/0x8A: SD 文件存储模式 0x02/0x8B: SD 文件下载模式 其它： 保留功能
<code>int samplemethod</code>	选择采样方式	0/x01: LAN/WIFI 实时命令采样 12/0x12/0x80: I04 触发 N 采样 13/0x13/0x99: I04 作为 ADC 时钟采样 21/0x21/0x9A: 定时 N 采样模式 (注：12 代表十进制 12，0x12 代表 16 进制 12，相当于 10 进制 18，下同)
<code>int sdfilefmt</code>	选择 SD 存储文件方式	VK701S, VK701N 和 VK702N 此参数无意义，默认 0; VK701NSD 为 SD 卡文件存储格式： 0x00: Bin 文件格式 1/11/0x11: Text 文件格式 (注：选用此存储格式时，采样率最大设置 8KHZ)

输出：返回状态说明如下表。

返回数值	状态说明
0 或大于 0	设置成功
-11	服务器未打开
-12	采集卡未连接或不存在
-13	采集卡未连接或不存在
其它小于 0 的数值	异常退出

 深圳微精艺电子有限公司 www.vkinging.com	Product:VK70xN 数据采集器 Revision: 6.07 Issued Date: 2022-10-21
DLL 函数使用手册 (自带 TCP 服务器解析多卡连接数据)	Sheet 51 of 68

【VC/C 调用例程】

```
extern "C" int VK70xNMC_Set_SystemMode(int mci, int sysmodeval, int
samplemethod, int sdfilefmt);
Void SwitchVK70xNMCSysSystemMode(void)
{
    int rstatus;
    //rstatus = VK70xNMC_Set_SystemMode(0, 0, 0, 0);
    //切换 LAN/WIFI 实时传输命令触发采样方式
    // rstatus = VK70xNMC_Set_SystemMode(0, 0, 0x80, 0);
    //切换 LAN/WIFI 实时传输 IO 触发采样方式
    rstatus = VK70xNMC_Set_SystemMode(0, 0x8A, 0x80, 0);
    //切换 SD 存储 IO 触发采样方式
    // rstatus = VK70xNMC_Set_SystemMode(0, 0x8B, 0, 0);
    //切换 SD 下载模式

    if (rstatus >=0) printf(“%s”, ‘操作成功!’);
    else if (tstatus == -11)printf(“服务器未打开!”);
    else if (tstatus == -12)printf(“采集卡未连接或不存在!”);
    else printf(“%s”, ‘异常退出!’);
}
```

 深圳微精艺电子有限公司 www.vkinging.com	Product:VK70xN 数据采集器 Revision: 6.07 Issued Date: 2022-10-21
DLL 函数使用手册 (自带 TCP 服务器解析多卡连接数据)	Sheet 52 of 68

(*) 重要提示:

由于不同项目有不同功能, 所以对应模式命令字和采样方式的命令字如下表:

模式关系		VK701/VK7012 /VK700	VK702	VK701N	VK702N	VK701NSD
模式字节	0	/	/	/	/	LAN 传输
	0x80	I04 触发 N 点	I04 触发 N 点	I04 触发 N 点	I04 触发 N 点	LAN 传输
	0x81	工厂模式	工厂模式	I04 做时钟/工厂模式	I04 做时钟/工厂模式	LAN 传输
	0x8A	/	/	/	/	SD 卡
	0x8B	/	/	/	/	下载 SD 文件
	0x8F	/	/	/	/	工厂模式
	其它	/	/	/	/	LAN 传输
采样方式	0	/	/	/	/	TIMERA 中断采样
	0x80	I04 触发 N 点 (与模式值为 0x80 组合)	I04 触发 N 点 (与模式值为 0x80 组合)	I04 触发 N 点 (与模式值为 0x80 组合)	I04 触发 N 点 (与模式值为 0x80 组合)	I04 触发 N 点
	0x99	/	/	I04 做时钟 (与模式值为 0x81 组合)	I04 做时钟 (与模式值为 0x81 组合)	I04 做时钟
	0x9A	/	/	/	定时采样 N 点 (与模式值为 0x81 组合)	定时采样 N 点
	0x7E	工厂模式 (与模式值为 0x81 组合)	工厂模式 (与模式值为 0x81 组合)	工厂模式 (与模式值为 0x81 组合)	工厂模式 (与模式值为 0x81 组合)	TIMERA 中断采样
	其它	/	/	/	/	TIMERA 中断采样
sd 文件格式	0x11	/	/	/	/	Text 文件
	其它	/	/	/	/	Bin 文件

 深圳微精艺电子有限公司 www.vkinging.com	Product:VK70xN 数据采集器 Revision: 6.07 Issued Date: 2022-10-21
DLL 函数使用手册 (自带 TCP 服务器解析多卡连接数据)	Sheet 53 of 68

7.1.2 读取当前系统模式

【函数原型】

```
int VK70xNMC_Get_SystemMode(int mci, int *sysmode, int *samplecmd, int *sdfilefmt, int timeout)
```

输入：设置参数说明如下表。

参数名称	参数描述	读取参数描述
int mci	已连接到服务器上的采集卡序号	范围：0-7。 注意：目前库函数版本仅仅支持最大连接采集卡 8 个。
int *sysmode	选择系统模式	0x00: LAN/WIFI 实时传输模式 0x01: SD 文件存储模式 0x02: SD 文件下载模式 其它： 保留功能
int *samplecmd	选择采样方式	0: LAN/WIFI 实时命令采样 0x12: I04 触发 N 采样 0x13: I04 作为 ADC 时钟采样 0x21: 定时 N 采样模式 （注：12 代表十进制 12，0x12 代表 16 进制 12，相当于 10 进制 18，下同）
int *sdfilefmt	采样文件格式	VK701S, VK701N 和 VK702N 此参数无意义，默认 0； VK701NSD 为 SD 卡文件存储格式： 0x00: Bin 文件格式 0x11: Text 文件格式 （注：选用此存储格式时，采样率最大设置 8KHZ）
int timeout	读取超时设置值	单位毫秒。0 不等待，timeout 大于 0 代表等待 timeout ms

输出：返回状态说明如下表。

返回数值	状态说明
0 或大于 0	设置成功
-1	读取超时退出
-11	服务器未打开
-12	采集卡未连接或不存在
-13	采集卡未连接或不存在
其它小于 0 的数值	异常退出

【VC/C 调用例程】

```
extern "C" int VK70xNMC_Get_SystemMode(int mci, int *sysmode, int *samplecmd, int *sdfilefmt, int timeout)
```

 深圳微精艺电子有限公司 www.vkinging.com	Product:VK70xN 数据采集器 Revision: 6.07 Issued Date: 2022-10-21
DLL 函数使用手册 (自带 TCP 服务器解析多卡连接数据)	Sheet 54 of 68

```

Void GetVK70xNMCSystemMode(void)
{
    int rstatus;
    int sysmode,samplecmd,sdfilefmt;
    rstatus = VK70xNMC_Get_SystemMode(0,&sysmode,&samplecmd,&sdfilefmt,200);
    /延时 200mS

    if (rstatus >=0) printf(“%s”, ‘操作成功!’);
    else if (tstatus == -1)printf(“读取超时退出!”);
    else if (tstatus == -11)printf(“服务器未打开!”);
    else if (tstatus == -12)printf(“采集卡未连接或不存在!”);
    else printf(“%s”, ‘异常退出!’);
}

```

7.2 系统缺省参数

7.2.1 设置缺省参数

【函数原型】

int VK70xNMC_Set_DefaultParameter(int mci, int *para);

输入：设置参数说明如下表。

参数名称	参数描述	设置范围
int mci	已连接到服务器上的采集卡序号	范围：0-7。 注意：目前库函数版本仅仅支持最大连接采集卡 8 个。
int *para	设置缺省参数	Para 输入缺省参数，排列顺序如下： para [0]：通道 1、2 的电压输入范围 para [1]：通道 3、4 的电压输入范围 para [2]：通道 5、6 的电压输入范围 para [3]：通道 7、8 的电压输入范围 para [4]：缺省采样率 para [5]：缺省 N 点采样数目 para [6]：缺省定时采集时间间隔 para [7]：缺省采样精度 para [8]：保留，升级扩展 para [9]：缺省系统模式 para [10]：缺省采样模式 para [11]：保留，升级扩展 para [12]：缺省保存的文件格式 para [13]：缺省文件索引流水号

 深圳微精艺电子有限公司 www.vkinging.com	Product:VK70xN 数据采集器 Revision: 6.07 Issued Date: 2022-10-21
DLL 函数使用手册 (自带 TCP 服务器解析多卡连接数据)	Sheet 55 of 68

输出：返回状态说明如下表。

返回数值	状态说明
0 或大于 0	设置成功
-11	服务器未打开
-12	采集卡未连接或不存在
-13	采集卡未连接或不存在
其它小于 0 的数值	异常退出

【VC/C 调用例程】

```
extern "C" int VK70xNMC_Set_DefaultParameter(int mci, int *para)
```

```
Void SetVK70xNMCDefaultParameter(void)
{
    int rstatus, i;
    int para[24];

    para [0]=0;//: 通道 1、2 的电压输入范围
    para [1] =0;//: 通道 3、4 的电压输入范围
    para [2] =0;//: 通道 5、6 的电压输入范围
    para [3] =0;//: 通道 7、8 的电压输入范围
    para [4] =1000;//: 缺省采样率
    para [5] =10000;//: 缺省 N 点采样数目
    para [6] =60;//: 缺省定时采集时间间隔
    para [7] =2;//: 缺省采样精度
    para [8] =0;//: 保留，升级扩展
    para [9] =0;//: 缺省系统模式
    para [10] =0;//: 缺省采样模式
    para [11] =0;//: 保留，升级扩展
    para [12] =0;//: 缺省保存的文件格式
    para [13] =1000;//: 缺省文件索引流水号

    rstatus = VK70xNMC_Set_DeviceDefaultParameter(0, para);

    if (rstatus >=0) printf( "%s", '操作成功! ');
    else if (tstatus == -11)printf( "服务器未打开! ");
    else if (tstatus == -12)printf( "采集卡未连接或不存在! ");
    else printf( "%s", '异常退出! ');
}
```

 深圳微精艺电子有限公司 www.vkinging.com	Product:VK70xN 数据采集器 Revision: 6.07 Issued Date: 2022-10-21
DLL 函数使用手册 (自带 TCP 服务器解析多卡连接数据)	Sheet 56 of 68

7.2.2 读取缺省参数

【函数原型】

```
int VK70xNMC_Get_DeviceDefaultParameter(int mci, char *rdmodol, char *rdsver,
char *rdhdver, char *rdsn, int *rdpara, int timeout);
```

输入：设置参数说明如下表。

参数名称	参数描述	读取参数描述
int mci	已连接到服务器上的采集卡序号	范围：0-7。 注意：目前库函数版本仅仅支持最大连接采集卡 8 个。
char *rdmodol	读取项目名称指针	rdmodol [7]:7 字符的项目名称
char *rdsver	读取软件版本指针	rdsver[7]: 7 个字符的项目名称
char *rdhdver	读取硬件版本指针	rdhdver[7]: 7 个字符的项目名称
char *rdsn	读取项目序列号指针	rdsn[17]: 17 个字符的项目编号
int *rdpara	读取缺省参数指针	读取缺省参数，排列顺序如下： para [0]: 通道 1、2 的电压输入范围 para [1]: 通道 3、4 的电压输入范围 para [2]: 通道 5、6 的电压输入范围 para [3]: 通道 7、8 的电压输入范围 para [4]: 缺省采样率 para [5]: 缺省 N 点采样数目 para [6]: 缺省定时采集时间间隔 para [7]: 缺省采样精度 para [8]: 保留，升级扩展 para [9]: 缺省系统模式 para [10]: 缺省采样模式 para [11]: 保留，升级扩展 para [12]: 缺省保存的文件格式 para [13]: 缺省文件索引流水号
int timeout	读取超时设置值	单位毫秒。0 不等待，timeout 大于 0 代表等待 timeout ms

输出：返回状态说明如下表。

返回数值	状态说明
0 或大于 0	设置成功
-1	读取超时

 深圳微精艺电子有限公司 www.vkinging.com	Product:VK70xN 数据采集器 Revision: 6.07 Issued Date: 2022-10-21
DLL 函数使用手册 (自带 TCP 服务器解析多卡连接数据)	Sheet 57 of 68

-11	服务器未打开
-12	采集卡未连接或不存在
-13	采集卡未连接或不存在
其它小于 0 的数值	异常退出

【VC/C 调用例程】

```
extern "C" int VK70xNMC_Get_DeviceDefaultParameter(int mci, char *rdmodel, char
*rdswver, char *rdhdver, char *rdsn, int *rdpara, int timeout);
```

```
Void GetVK70xNMCDeviceDefaultParameter(void)
{
    int rstatus,i;
    char modelname[7];
    char rdswver[7];
    char rdhdver[7];
    char sn[17];
    int para[24];

    rstatus = VK70xNMC_Get_DeviceDefaultParameter(0, modelname,rdswver,rdhdver,
sn, para);

    if (rstatus >=0) printf( "%s", '操作成功!' );
    else if (tstatus == -11)printf( "服务器未打开!" );
    else if (tstatus == -12)printf( "采集卡未连接或不存在!" );
    else printf( "%s", '异常退出!' );
}
```

7.3 网络参数

7.3.1 设置网络

【函数原型】

```
int VK70xNMC_Set_eNetParameter(int mci, char *para, int len);
```

输入：设置参数说明如下表。

参数名称	参数描述	设置范围
int mci	已连接到服务器上的采集卡序号	范围：0-7。 注意：目前库函数版本仅仅支持最大连接采集卡 8 个。
char	设置缺省参	输入缺省参数，排列顺序如下：

 深圳微精艺电子有限公司 www.vkinging.com	Product:VK70xN 数据采集器 Revision: 6.07 Issued Date: 2022-10-21
DLL 函数使用手册 (自带 TCP 服务器解析多卡连接数据)	Sheet 58 of 68

*para	数	para[0]~ para[3]: 服务器的 IP 地址, 如 192.162.1.188 para[4]~ para[7]: 采集卡的 IP 地址, 如 192.162.1.199 para[8]~ para[11]: 采集卡的网关地址, 如 192.162.1.1 para[12]~ para[15]: 采集卡的子网络掩码, 如 255.255.255.0 para[16]~ para[17]: 服务器端口号, 高字节在前, 如 8234 para[18]~ para[19]: 采集卡端口号, 高字节在前, 如 8234 para[20]: 网络协议; 0: UDP 协议, 1: TCP 协议 para[21]: 网络断开重连请求间隔时间, 默认 10 秒 para[22]: 保留, ADC 缓冲时间, 默认 10 秒 para[23]~ para[28]: 采集卡 MAC 地址, 如 ee:1d:34:34:44:d6 para[29]~ para[31]: 保留
int len	设置参数长度	设置参数长度。 重要提示: (1) 如果 len 小于 23, 将退出设置网络参数; (2) 如果 23<=len<=50, 采集卡 MAC 地址自动忽略不更新; (3) 如果 len 大于 50, 才可更新 MAC 地址; (4) 务必输入正确网络参数, 否则会导致采集卡无法通讯。

输出: 返回状态说明如下表。

返回数值	状态说明
0 或大于 0	设置成功
-5	设置网络参数错误
-11	服务器未打开
-12	采集卡未连接或不存在
-13	采集卡未连接或不存在
其它小于 0 的数值	异常退出

【VC/C 调用例程】

```
extern "C" int VK70xNMC_Set_eNetParameter(int mci, char *para, int len);
```

 深圳微精艺电子有限公司 www.vkinging.com	Product:VK70xN 数据采集器 Revision: 6.07 Issued Date: 2022-10-21
DLL 函数使用手册 (自带 TCP 服务器解析多卡连接数据)	Sheet 59 of 68

```

Void SetVK70xNMCDeviceeNetParameter(void)
{
    int rstatus, i, temp;
    char para[64];
    for(i=0;i<64;i++)
    {
        para [i]=0;
    }
    //-----

    para [0]= 192;//
    para [1] =168;//
    para [2] =1;//
    para [3] =188;// 服务器的 IP 地址, 如 192.162.1.188

    para [4]= 192;//
    para [5] =168;//
    para [6] =1;//
    para [7] =199;// 采集卡的 IP 地址, 如 192.162.1.199

    para [8]= 192;//
    para [9] =168;//
    para [10] =1;//
    para [11] =1;// 采集卡的网关地址, 如 192.162.1.1

    para [12]= 255;//
    para [13] =255;//
    para [14] =255;//
    para [15] =0;// 采集卡的子网络掩码, 如 255.255.255.0

    temp=8234;
    para [16]= (char) (temp>>8);//
    para [17] = (char)temp;// 服务器端口号, 高字节在前, 如 8234

    temp=8234;
    para [18]= (char) (temp>>8);//
    para [19] = (char)temp;// 采集卡端口号, 高字节在前, 如 8234

    para[20]=1;// 网络协议; 0: UDP 协议, 1: TCP 协议
    para[21] =10;// 网络断开重连请求间隔时间, 默认 10 秒
    para[22] =10;// 保留, ADC 缓冲时间, 默认 10 秒

```

 深圳微精艺电子有限公司 www.vkinging.com	Product:VK70xN 数据采集器 Revision: 6.07 Issued Date: 2022-10-21
DLL 函数使用手册 (自带 TCP 服务器解析多卡连接数据)	Sheet 60 of 68

```

para [23]= 0xee;//
para [24] =0x1d;//
para [25] =0x34;//
para [26] =0x34;//
para [27] =0x44;//
para [28] =0xd6;// 采集卡 MAC 地址，如 ee:1d:34:34:44:d6

para [29] =0;//
para [30] =0;//
para [31] =0;//
//rstatus = VK70xNMC_Set_eNetParameter(0, para, 32); //不更新 MAC 地址
rstatus = VK70xNMC_Set_eNetParameter(0, para, 51); //更新 MAC 地址

if (rstatus >=0) printf(“%s”, ‘操作成功!’);
else if (tstatus == -5)printf(“设置网络参数错误!”);
else if (tstatus == -11)printf(“服务器未打开!”);
else if (tstatus == -12)printf(“采集卡未连接或不存在!”);
else printf(“%s”, ‘异常退出!’);
}

```

7.3.2 读取网络参数

【函数原型】

int VK70xNMC_Get_eNetParameter(int mci, char *para, int timeout);

输入：设置参数说明如下表。

参数名称	参数描述	读取参数
int mci	已连接到服务器上的采集卡序号	范围：0-7。 注意：目前库函数版本仅仅支持最大连接采集卡 8 个。
char *para	设置缺省参数	读取缺省参数，排列顺序如下： para[0]~ para[3]：服务器的 IP 地址，如 192.162.1.188 para[4]~ para[7]：采集卡的 IP 地址，如 192.162.1.199 para[8]~ para[11]：采集卡的网关地址，如 192.162.1.1 para[12]~ para[15]：采集卡的子网络掩码，如 255.255.255.0 para[16]~ para[17]：服务器端口号，高字节在前，如 8234 para[18]~ para[19]：采集卡端口号，高字节在

 深圳微精艺电子有限公司 www.vkinging.com	Product:VK70xN 数据采集器 Revision: 6.07 Issued Date: 2022-10-21
DLL 函数使用手册 (自带 TCP 服务器解析多卡连接数据)	Sheet 61 of 68

		前, 如 8234 para[20]: 网络协议; 0: UDP 协议, 1: TCP 协议 para[21]: 网络断开重连请求间隔时间, 默认 10 秒 para[22]: 保留, ADC 缓冲时间, 默认 10 秒 para[23]~ para[28]: 采集卡 MAC 地址, 如 ee:1d:34:34:44:d6 para[29]~ para[31]: 保留 重要提示: 读取缓冲区一定要大于 32 个字节, 否则会发生内存溢出;
int timeout	读取超时设置值	单位毫秒。0 不等待, timeout 大于 0 代表等待 timeout ms

输出: 返回状态说明如下表。

返回数值	状态说明
0 或大于 0	设置成功
-5	设置网络参数错误
-11	服务器未打开
-12	采集卡未连接或不存在
-13	采集卡未连接或不存在
其它小于 0 的数值	异常退出

【VC/C 调用例程】

```
extern "C" int VK70xNMC_Get_eNetParameter(int mci, char *para, int timeout);
```

```
Void GetVK70xNMCDeviceeNetParameter(void)
{
```

```
    Int  rstatus, i, temp;
    char para[64];
```

```
    rstatus = VK70xNMC_Set_eNetParameter(0, para, 200); //延时 200ms
```

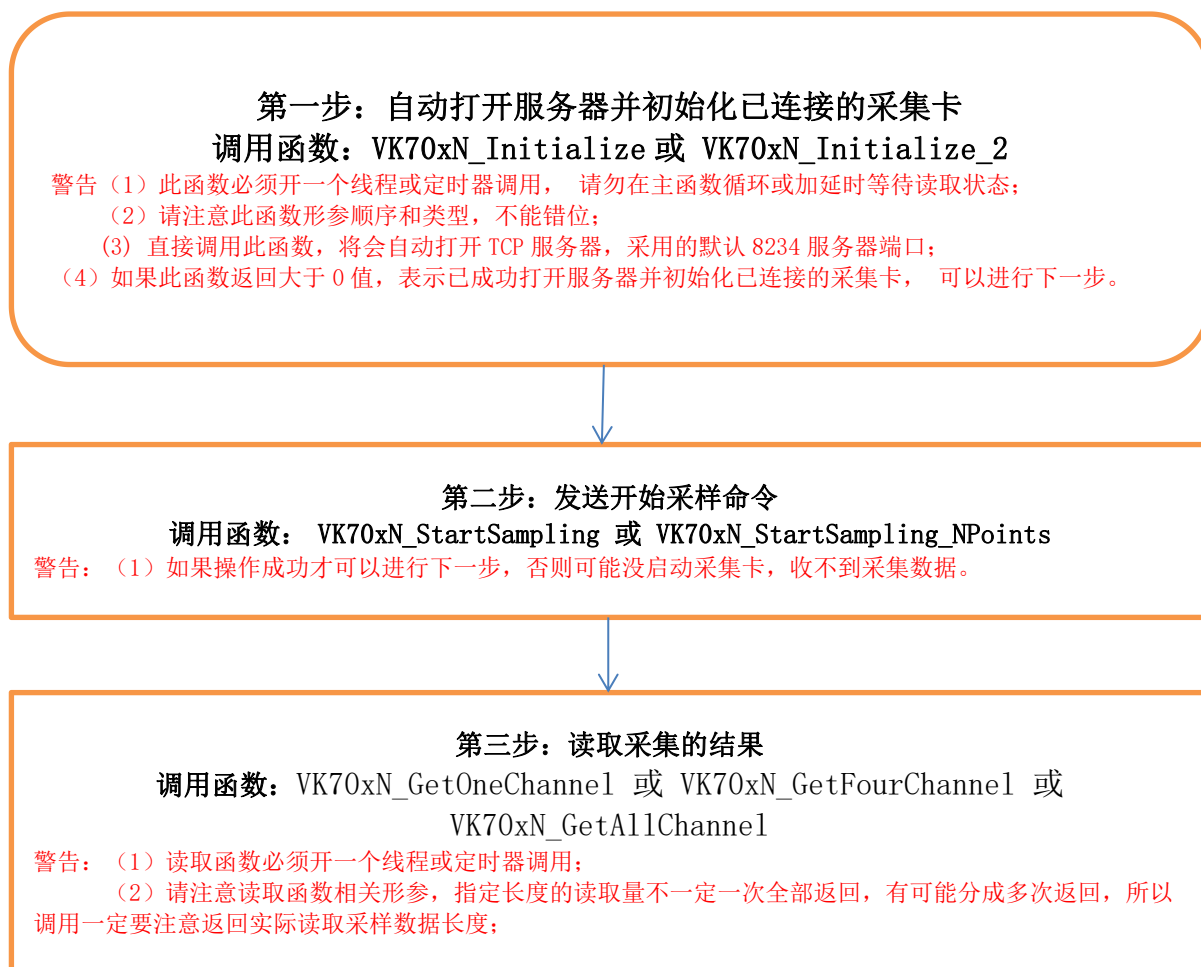
```
    if (rstatus >= 0) printf( "%s", '操作成功! ');
    else if (tstatus == -5) printf( "设置网络参数错误! ");
    else if (tstatus == -11) printf( "服务器未打开! ");
    else if (tstatus == -12) printf( "采集卡未连接或不存在! ");
    else printf( "%s", '异常退出! ');
```

```
}
```

 深圳微精艺电子有限公司 www.vkinging.com	Product:VK70xN 数据采集器 Revision: 6.07 Issued Date: 2022-10-21
DLL 函数使用手册（自带 TCP 服务器解析多卡连接数据）	Sheet 62 of 68

8. 处理流程

8.1 打开默认端口（8234）服务器的操作流程



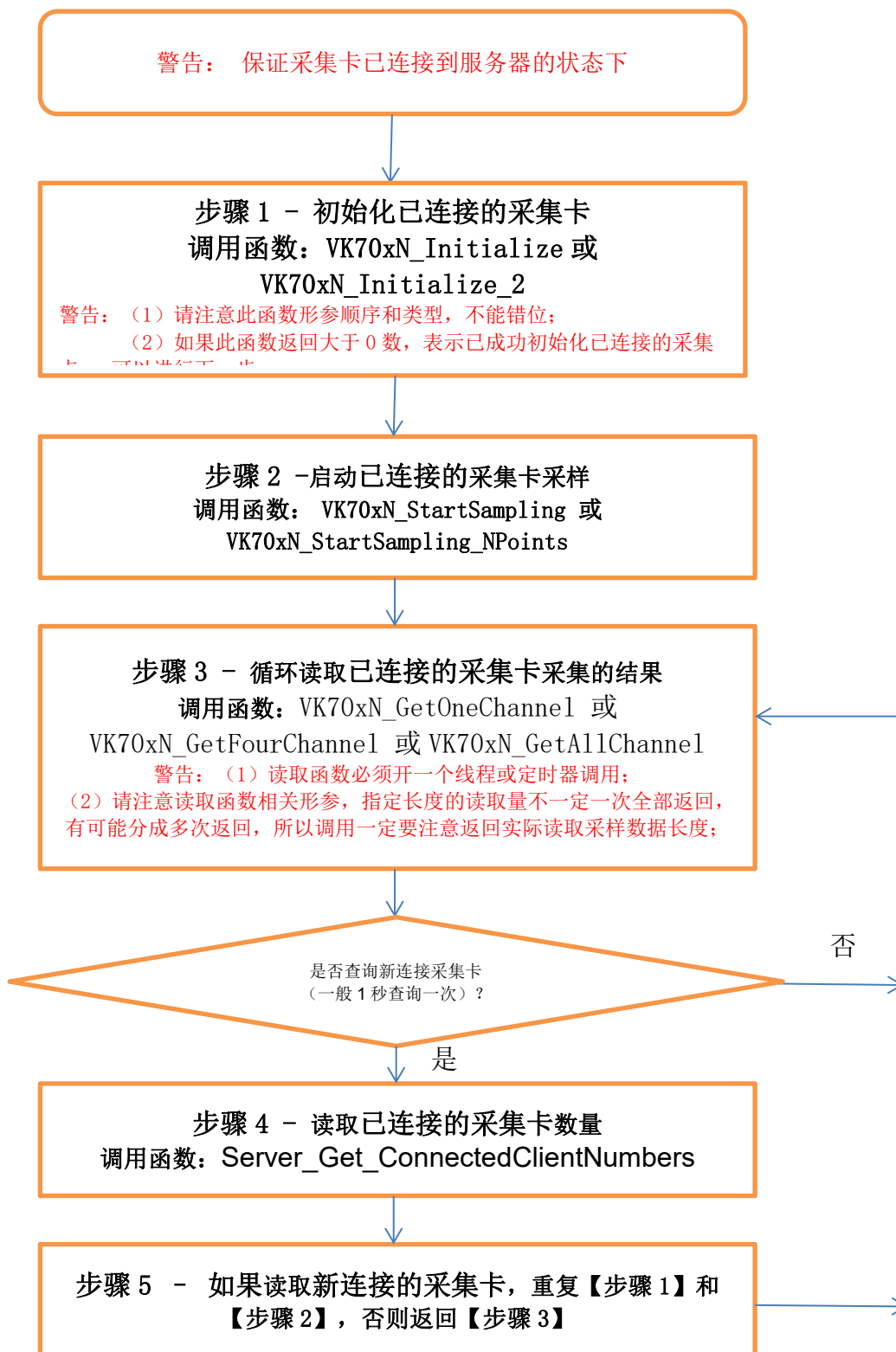
 深圳微精艺电子有限公司 www.vkinging.com	Product:VK70xN 数据采集器 Revision: 6.07 Issued Date: 2022-10-21
DLL 函数使用手册 (自带 TCP 服务器解析多卡连接数据)	Sheet 63 of 68

8.2 打开自定义端口服务器的操作流程



 深圳微精艺电子有限公司 www.vking.com	Product: VK70xN 数据采集器 Revision: 6.07 Issued Date: 2022-10-21
DLL 函数使用手册 (自带 TCP 服务器解析多卡连接数据)	Sheet 64 of 68

8.3 多卡连接操作



 深圳微精艺电子有限公司 www.vkinging.com	Product:VK70xN 数据采集器 Revision: 6.07 Issued Date: 2022-10-21
DLL 函数使用手册（自带 TCP 服务器解析多卡连接数据）	Sheet 65 of 68

8.4 附件功能操作

警告： 保证采集卡已连接到服务器的状态下



调用函数： VK70xN_Set_AdditionalFeature

警告： (1) 函数形参 1 表示选择要设置的功能， 形参 2 和形参 3 表示要设置的参数。如果设置 PWM1, 10000HZ, 50.5%占空比： VK70xN_Set_AdditionalFeature(0, 1, 10000, 50.5);
(2) 详细的功能详见第四章第一节输入参数定义表。



调用函数：

```
int VK70xNMC_Get_AllIOS_Blocking(int mci, unsigned int *iobuffer, int timeout);
int VK70xNMC_Get_IO1_Blocking(int mci, unsigned int *iovalue, int timeout);
int VK70xNMC_Get_IO2_Blocking(int mci, unsigned int *iovalue, int timeout);
int VK70xNMC_Get_IO3_Blocking(int mci, unsigned int *iovalue, int timeout);
int VK70xNMC_Get_IO4_Blocking(int mci, unsigned int *iovalue, int timeout);
int VK70xNMC_Get_PWM_Blocking(int mci, double *dutyval, unsigned int
*freqval, int timeout);
int VK70xNMC_Get_DAC_Blocking(int mci, double *dacvalue, int timeout);
int VK70xNMC_Get_Counter_Blocking(int mci, unsigned int *countervalue, int
timeout);
int VK70xNMC_Get_Temperature_Blocking(int mci, double *tempvalue, int timeout);
```

 深圳微精艺电子有限公司 www.vkinging.com	Product:VK70xN 数据采集器 Revision: 6.07 Issued Date: 2022-10-21
DLL 函数使用手册 (自带 TCP 服务器解析多卡连接数据)	Sheet 66 of 68

8.5 切换采样模式操作

警告： 保证采集卡已连接到服务器的状态下

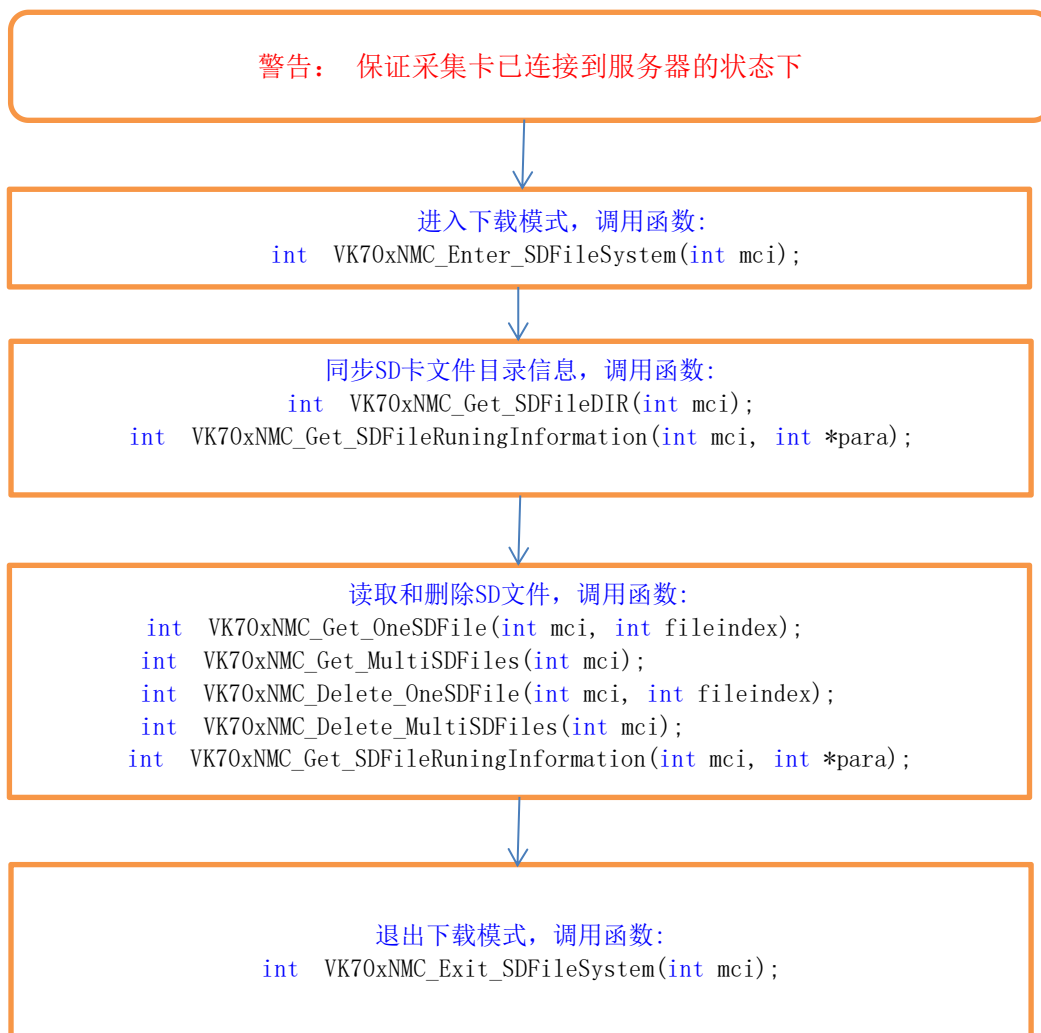


调用函数： VK70xN_Set_SystemMode

警告： (1) 采集卡接收到此命令，将停止当前所有采集任务，自动切换另外一个模式；

 深圳微精艺电子有限公司 www.vkinging.com	Product:VK70xN 数据采集器 Revision: 6.07 Issued Date: 2022-10-21
DLL 函数使用手册 (自带 TCP 服务器解析多卡连接数据)	Sheet 67 of 68

8.6 SD 卡文件系统操作



 深圳微精艺电子有限公司 www.vkinging.com	Product:VK70xN 数据采集器 Revision: 6.07 Issued Date: 2022-10-21
DLL 函数使用手册 (自带 TCP 服务器解析多卡连接数据)	Sheet 68 of 68

修改记录

版本	描述	发布者	修改日期
3.00	第一次准备	Andy	DEC 17, 2017
3.01	增加 VK701N 读取四通道函数	Andy	DEC 19, 2017
3.02	纠正读取 4 通道描述错误和更新 LOGO	Andy	OTC 10, 2018
3.03	增加停止线程处理函数描述	Andy	OTC 21, 2018
3.04	增加在 24 位模式下读取 IO2 和 IO3 状态	Andy	DEC 04, 2019
3.05	增加读取 IO 状态函数	Andy	DEC 16, 2019
3.06	增加 VK701N 读取 IO 状态函数	Andy	DEC 29, 2019
5.00	增加 TCP 服务器操作函数	Andy	Sept 20, 2020
5.01	增加操作函数流程和修正部分函数返回错误描述	Andy	Sept 22, 2020
6.00	更新所有 DLL, 原函数 VK70XN 变成 VK70xNMC, 增加函数操作时采集卡序号, 方便多卡连接时单独控制已连接的采集卡。	Andy	Nov 14, 2020
6.01	更新多卡操作描述	Andy	Nov 17, 2020
6.02	增加读取 IO 状态/DAC 参数/PWM 参数/计数器/温度	Andy	Dec 20, 2020
6.03	更新输入电压描述错误	Andy	Jan 11, 2021
6.04	- 修改 IO 读取函数操作方式 - 增加读写系统参数, 网络参数等 - 增加 SD 文件读取 - 修改模式切换函数 - 增加 ADC 采样设置函数, 适用与 VK701NSD 采集卡	Andy	Dec 12, 2021
6.05	-修正设置函数描述	Andy	May 14, 2022
6.06	-修正开始采样函数参数描述 -增加模式不同项目的命令字描述	Andy	Oct. 08, 2022
6.07	修改缺省设置函数描述	Andy	Oct. 21, 2022