

API

DX5100.dll

Service functions	4
OpenSerialPort	4
CloseSerialPort	4
CheckConnection	4
SetDeviceAddress.....	5
GetLastDeviceError	5
System commands	6
0x02 CMD_ECHO	6
0x03 CMD_INFO	6
0x04 CMD_GetVer.....	6
0x05 CMD_GetInfo.....	7
0x06 CMD_SetInfo	7
0x07 CMD_SetAdr	7
0x40 CMD_StTel	8
0x41 CMD_getPRM	8
0x42 CMD_setPRM.....	8
0x44 CMD_I2C.....	9
0x45 CMD_Prog_T.....	9
0x46 CMD_get_Tel	11
0x49 CMD_Krt_Ok.....	11
0x4a CMD_St_HW	12
0x4b CMD_Infs_WK	12
0x4d CMD_Dig_Out.....	13
0x4e CMD_Dig_In.....	14
Commands of Work with ADC.....	15
0x10 CMD_ClbrADC.....	15
0x11 CMD_ClbrK_ADC.....	15
0x14 CMD_AskKADC	15
0x15 CMD_AskOfst.....	16
0x16 CMD_StartADC	16
0x17 CMD_Only_1.....	16
0x18 CMD_Sever	17
0x19 CMD_PGA	18
0x1a CMD_Polynom	18

0x1b CMD_ask_Pol.....	19
0x1c CMD_saveTerm.....	19
0x1d CMD_loadTerm.....	20
0x1e CMD_get_TBL	20
0x1f CMD_set_TBL	20
Commands of work with DAC.....	21
0x21 CMD_set_DAC.....	21
0x22 CMD_seth_DAC	21
0x23 CMD_Wr_K_DAC	21
0x24 CMD_AskKDAC.....	22
0x25 CMD_DAC_max.....	22
0x26 CMD_U_Treg.....	22
Commands of work with PID controller	23
0x30 CMD_Pol_TEC	23
0x31 CMD_set_PID.....	23
0x32 CMD_ask_PID	24
0x33 CMD_setCurrT	24
0x34 CMD_askT_PID.....	25
0x35 CMD_strt_PID	26
0x37 CMD_Zmetr	26
0x38 CMD_Zprmtr	26
0x39 CMD_Z_I	27
0x3b CMD_Boot	27
0x3c CMD_Set_LimT.....	28
0x3d CMD_Get_LimT.....	29
0x3e CMD_ResZmtr.....	29
0x51 CMD_PID_fun	29
0x53 CMD_RESET	30

Service functions

```
//-----
```

OpenSerialPort

```
typedef HANDLE __declspec(dllimport) OpenSerialPort(char* PortName, DWORD baud);

OpenSerialPort *ptrFunc;
if (Dll)
{
    ptrFunc = (OpenSerialPort *)GetProcAddress(Dll, "_OpenSerialPort");
    if (ptrFunc)
    {
        HANDLE hCom = ptrFunc("COM1", 19200);
    }
    else
    {
        ShowMessage(SysErrorMessage(GetLastError()));
    }
}
else
    ShowMessage("library not loaded.\n"+SysErrorMessage(GetLastError()));
//-----
```

CloseSerialPort

```
typedef HANDLE __declspec(dllimport) CloseSerialPort();

CloseSerialPort *ptrFunc;
if (Dll)
{
    ptrFunc = (CloseSerialPort *)GetProcAddress(Dll, "_CloseSerialPort");
    if (ptrFunc)
    {
        hCom = ptrFunc();
    }
    else
    {
        ShowMessage(SysErrorMessage(GetLastError()));
    }
}
else
    ShowMessage("library not loaded.\n"+SysErrorMessage(GetLastError()));
//-----
```

CheckConnection

```
typedef bool __declspec(dllimport) CheckConnection(int iTimeOut);

CheckConnection *ptrFunc;
if (Dll)
{
    ptrFunc = (CheckConnection *)GetProcAddress(Dll, "_CheckConnection");
    if (ptrFunc)
    {
        bool Status = ptrFunc(5000);
    }
    else
    {
        ShowMessage(SysErrorMessage(GetLastError()));
    }
}
else
    ShowMessage("library not loaded.\n"+SysErrorMessage(GetLastError()));
```

```
//-----
```

SetDeviceAddress

```
typedef bool __declspec(dllimport) SetDeviceAddress(char Addr);
SetDeviceAddress *ptrFunc;
if (Dll)
{
    ptrFunc = (SetDeviceAddress *)GetProcAddress(Dll, "_SetDeviceAddress");
    if (ptrFunc)
    {
        char NewAddr = 0x0A;
        if(ptrFunc(NewAddr)) ShowMessage("Ok");
        else ShowMessage("Error");
    }
    else
    {
        ShowMessage(SysErrorMessage(GetLastError()));
    }
}
else
    ShowMessage("library not loaded.\n"+SysErrorMessage(GetLastError()));
```

```
//-----
```

GetLastDeviceError

```
typedef DWORD __declspec(dllimport) GetLastDeviceError();
GetLastDeviceError *ptrFunc;
if (Dll)
{
    ptrFunc = (GetLastDeviceError *)GetProcAddress(Dll, "_GetLastDeviceError");
    if (ptrFunc)
    {
        DWORD res = ptrFunc();
        ShowMessage("Last Device Error =" + IntToHex((__int64)res, 8));
    }
    else
    {
        ShowMessage(SysErrorMessage(GetLastError()));
    }
}
else
    ShowMessage("library not loaded.\n"+SysErrorMessage(GetLastError()));
```

System commands

0x02 CMD_ECHO

```

typedef bool __declspec(dllimport) CMD_ECHO(char *s, char len, char *answer);
CMD_ECHO *ptrFunc;
if (Dll)
{
    ptrFunc = (CMD_ECHO *)GetProcAddress(Dll, "_CMD_ECHO");
    if (ptrFunc)
    {
        char res[64] = {0};
        char str[] = "test string";
        if (ptrFunc(str, strlen(str), res)) ShowMessage(res);
        else ShowMessage("error in function CMD_ECHO");
    }
    else
    {
        ShowMessage(SysErrorMessage(GetLastError()));
    }
}
else
    ShowMessage("library not loaded.\n"+SysErrorMessage(GetLastError()));
//-----

```

0x03 CMD_INFO

```

typedef bool __declspec(dllimport) CMD_INFO(char* input);
CMD_INFO *ptrFunc;
if (Dll)
{
    ptrFunc = (CMD_INFO *)GetProcAddress(Dll, "_CMD_INFO");
    if (ptrFunc)
    {
        char res[2] = {0};
        if (ptrFunc(res)) ShowMessage(IntToHex(MAKEWORD(res[1], res[0]), 4));
        else ShowMessage("error in function CMD_INFO");
    }
    else
    {
        ShowMessage(SysErrorMessage(GetLastError()));
    }
}
else
    ShowMessage("library not loaded.\n"+SysErrorMessage(GetLastError()));
//-----

```

0x04 CMD_GetVer

```

typedef bool __declspec(dllimport) CMD_GetVer(char *FirmwareVer);
CMD_GetVer *ptrFunc;
if (Dll)
{
    ptrFunc = (CMD_GetVer *)GetProcAddress(Dll, "_CMD_GetVer");
    if (ptrFunc)
    {
        char res[32] = {0};
        if (ptrFunc(res)) ShowMessage(res);
        else ShowMessage("error in function CMD_GetVer");
    }
    else
    {
        ShowMessage(SysErrorMessage(GetLastError()));
    }
}
else
    ShowMessage("library not loaded.\n"+SysErrorMessage(GetLastError()));

```

```
//-----
```

0x05 CMD_GetInfo

```
typedef bool __declspec(dllimport) CMD_GetInfo(char *DevInfo);

CMD_GetInfo *ptrFunc;
if (Dll)
{
    ptrFunc = (CMD_GetInfo *)GetProcAddress(Dll, "_CMD_GetInfo");
    if (ptrFunc)
    {
        char res[32] = {0};
        if (ptrFunc(res)) ShowMessage(res);
        else ShowMessage("error in function CMD_GetInfo");
    }
    else
    {
        ShowMessage(SysErrorMessage(GetLastError()));
    }
}
else
    ShowMessage("library not loaded.\n"+SysErrorMessage(GetLastError()));
//-----
```

0x06 CMD_SetInfo

```
typedef bool __declspec(dllimport) CMD_SetInfo(unsigned char* sDeviceInfo);
CMD_SetInfo *ptrFunc;
if (Dll)
{
    ptrFunc = (CMD_SetInfo *)GetProcAddress(Dll, "_CMD_SetInfo");
    if (ptrFunc)
    {
        char res[] = "My New Device";
        if (ptrFunc(res)) ShowMessage(res);
        else ShowMessage("error in function CMD_SetInfo");
    }
    else
    {
        ShowMessage(SysErrorMessage(GetLastError()));
    }
}
else
    ShowMessage("library not loaded.\n"+SysErrorMessage(GetLastError()));
//-----
```

0x07 CMD_SetAdr

```
typedef bool __declspec(dllimport) CMD_SetAdr(char Addr);
CMD_SetAdr *ptrFunc;
if (Dll)
{
    ptrFunc = (CMD_SetAdr *)GetProcAddress(Dll, "_CMD_SetAdr");
    if (ptrFunc)
    {
        char NewAddr = 0xA;
        if (ptrFunc(NewAddr)) ShowMessage("Ok");
        else ShowMessage("error in function CMD_SetAdr");
    }
    else
    {
        ShowMessage(SysErrorMessage(GetLastError()));
    }
}
else
    ShowMessage("library not loaded.\n"+SysErrorMessage(GetLastError()));
```

```
//-----
```

0x40 CMD_StTel

```
typedef bool __declspec(dllimport) CMD_StTel(char d, WORD Status);
CMD_StTel *ptrFunc;
if (Dll)
{
    ptrFunc = (CMD_StTel *)GetProcAddress(Dll, "_CMD_StTel");
    if (ptrFunc)
    {
        char d = 10;           // telemetry output period (10 = 100 ms)
        WORD Status = 0x307F;  // telemetry mask
        if (ptrFunc(d, Status)) ShowMessage("Ok");
        else ShowMessage("error in function CMD_StTel");
    }
    else
    {
        ShowMessage(SysErrorMessage(GetLastError()));
    }
}
else
    ShowMessage("library not loaded.\n"+SysErrorMessage(GetLastError()));
//-----
```

0x41 CMD_getPRM

```
typedef bool __declspec(dllimport) CMD_getPRM(char* OutputFileDir);
CMD_getPRM *ptrFunc;
if (Dll)
{
    ptrFunc = (CMD_getPRM *)GetProcAddress(Dll, "_CMD_getPRM");
    if (ptrFunc)
    {
        if (ptrFunc("c:\\temp")) ShowMessage("Ok");
        else ShowMessage("error in function CMD_getPRM");
    }
    else
    {
        ShowMessage(SysErrorMessage(GetLastError()));
    }
}
else
    ShowMessage("library not loaded.\n"+SysErrorMessage(GetLastError()));
//-----
```

0x42 CMD_setPRM

```
typedef bool __declspec(dllimport) CMD_setPRM(char* FileName);
CMD_setPRM *ptrFunc;
if (Dll)
{
    ptrFunc = (CMD_setPRM *)GetProcAddress(Dll, "_CMD_setPRM");
    if (ptrFunc)
    {
        if (ptrFunc("c:\\temp\\#S213 14.05.2012.txt")) ShowMessage("Ok");
        else ShowMessage("error in function CMD_setPRM");
    }
    else
    {
        ShowMessage(SysErrorMessage(GetLastError()));
    }
}
else
    ShowMessage("library not loaded.\n"+SysErrorMessage(GetLastError()));
```

```
//-----
```

0x44 CMD_I2C

CMD_I2C_SET

```
typedef bool __declspec(dllimport) CMD_I2C_SET(char NumDeviceI2C, WORD
AddrDeviceI2C, char Val);
CMD_I2C_SET *ptrFunc;
if (Dll)
{
    ptrFunc = (CMD_I2C_SET *)GetProcAddress(Dll, "_CMD_I2C_SET");
    if (ptrFunc)
    {
        // for example: enable Digital Input for Channel #1
        if (ptrFunc(4,0,0xFB)) ShowMessage("Ok");
        else ShowMessage("error in function CMD_I2C_SET");
    }
    else
    {
        ShowMessage(SysErrorMessage(GetLastError()));
    }
}
else
    ShowMessage("library not loaded.\n"+SysErrorMessage(GetLastError()));
//-----
```

CMD_I2C_GET

```
typedef bool __declspec(dllimport) CMD_I2C_GET(char NumDeviceI2C, WORD
AddrDeviceI2C, char &Val);
CMD_I2C_GET *ptrFunc;
if (Dll)
{
    ptrFunc = (CMD_I2C_GET *)GetProcAddress(Dll, "_CMD_I2C_GET");
    if (ptrFunc)
    {
        // for example: read Digital Input for Channel #1
        char res;
        if (ptrFunc(4,0,res)) {
            bool bEnabled = !(res & 0x04);
            ShowMessage("Digital Input ChN1 = " + String(bEnabled ? "TRUE":"FALSE"));
        }
        else ShowMessage("error in function CMD_I2C_GET");
    }
    else
    {
        ShowMessage(SysErrorMessage(GetLastError()));
    }
}
else
    ShowMessage("library not loaded.\n"+SysErrorMessage(GetLastError()));
//-----
```

0x45 CMD_Prog_T

```
typedef bool __declspec(dllimport) CMD_Prog_T (char &Mode, char &PrgNum,
char &LineNum, float &T, unsigned short &Time, char &ModePrg, char &NextLine);
CMD_Prog_T *ptrFunc;
if (Dll)
{
    ptrFunc = (CMD_Prog_T *)GetProcAddress(Dll, "_CMD_Prog_T");
    if (ptrFunc)
    {
        char Mode = 1; // 0-write, 1-read, 2-set status, 3-get status
        char PrgNum = 0;
        char LineNum = 0;
        float T = 0;
        WORD Time = 0;
        char ModePrg = 0; // Most significant nibble - mode of current line
                        // 0 Interdiction of regulation
                        // 2 T-regulation
                        // 3 Temperature maintenance (PID)
                        // 4 Constant voltage
                        // Least significant nibble - number of
                        // program to go (0-15) after
    }
}
```

```
char NextLine = 0;
if (ptrFunc (Mode, PrgNum, LineNum, T, Time, ModePrg, NextLine))
{
    ShowMessage ("Mode="+IntToStr (Mode)
        +"\nPrgNum="+IntToStr (PrgNum)
        +"\nLineNum="+IntToStr (LineNum)
        +"\nT="+FormatFloat ("#.##", T)
        +"\nTime="+IntToStr (Time)
        +"\nModePrg="+IntToStr (ModePrg>>4)
        +"\nNextPrg="+IntToStr (ModePrg&0x0F)
        +"\nNextLine="+IntToStr (NextLine));
}
else ShowMessage ("error in function CMD_Prog_T");
}
else
{
    ShowMessage (SysErrorMessage (GetLastError ()));
}
}
else
    ShowMessage ("library not loaded.\n"+SysErrorMessage (GetLastError ()));
```

//-----

0x46 CMD_get_Tel

```

typedef bool __declspec(dllimport) CMD_get_Tel(OUTPUT_TELEMETRY &TelData);
CMD_get_Tel *ptrFunc;
if (Dll)
{
    ptrFunc = (CMD_get_Tel *)GetProcAddress(Dll, "_CMD_get_Tel");
    if (ptrFunc)
    {
        OUTPUT_TELEMETRY tel;
        if (ptrFunc(tel))
        {
            ShowMessage("Time="+IntToStr(tel.Time)
                +"\nVoltageInput="+FormatFloat("#.##",tel.VoltageInput)
                +"\nVoltageTec[0]="+FormatFloat("#.##",tel.VoltageTec[0])
                +"\nCurrentTec[0]="+FormatFloat("#.##",tel.CurrentTec[0])
                +"\nTempTec[0]="+FormatFloat("#.##",tel.TempTec[0])
                +"\nTempRefTec[0]="+FormatFloat("#.##",tel.TempRefTec[0])
                +"\nPidStatus[0]="+IntToHex(tel.PidStatus[0],2)
                +"\nDevStatus="+IntToStr(tel.DevStatus));
        }
        else ShowMessage("error in function CMD_get_Tel");
    }
    else
    {
        ShowMessage(SysErrorMessage(GetLastError()));
    }
}
else
    ShowMessage("library not loaded.\n"+SysErrorMessage(GetLastError()));

```

//-----

0x49 CMD_Krt_Ok

```

typedef bool __declspec(dllimport) CMD_Krt_Ok(char ChN, char D1, char D2, float
Value);
CMD_Krt_Ok *ptrFunc;
if (Dll)
{
    ptrFunc = (CMD_Krt_Ok *)GetProcAddress(Dll, "_CMD_Krt_Ok");
    if (ptrFunc)
    {
        {
            if (ptrFunc(0,20,10,0.15)) ShowMessage("Ok");
            else ShowMessage("error in function CMD_Krt_Ok");
        }
        else
        {
            ShowMessage(SysErrorMessage(GetLastError()));
        }
    }
}
else
    ShowMessage("library not loaded.\n"+SysErrorMessage(GetLastError()));

```

//-----

0x4a CMD_St_HW

```

typedef bool __declspec(dllimport) CMD_St_HW(DEVICE_CONF &DevConf,
                                              PID_STATUS &ChlStatus1,
                                              PID_STATUS &ChlStatus2);

CMD_St_HW *ptrFunc;
if (Dll)
{
    ptrFunc = (CMD_St_HW *)GetProcAddress(Dll, "_CMD_St_HW");
    if (ptrFunc)
    {
        DEVICE_CONF dev;
        PID_STATUS st1;
        PID_STATUS st2;
        if (ptrFunc(dev, st1, st2))
        {
            ShowMessage("PID 1\nRunning="+IntToStr(st1.Running)
                        +"\nTempInRange="+IntToStr(st1.TempInRange)
                        +"\nHeating="+IntToStr(st1.Heating)
                        +"\nChlPresent="+IntToStr(st1.ChlPresent)
                        +"\nMode="+IntToStr(st1.Mode));
        }
        else ShowMessage("error in function CMD_St_HW");
    }
    else
    {
        ShowMessage(SysErrorMessage(GetLastError()));
    }
}
else
    ShowMessage("library not loaded.\n"+SysErrorMessage(GetLastError()));

```

//-----

0x4b CMD_Infs_WK

CMD_Infs_WK_SET

```

typedef bool __declspec(dllimport) CMD_Infs_WK_SET(char IntfNum, char BaudRate);
CMD_Infs_WK_SET *ptrFunc;
if (Dll)
{
    ptrFunc = (CMD_Infs_WK_SET *)GetProcAddress(Dll, "_CMD_Infs_WK_SET");
    if (ptrFunc)
    {
        //interface:    0-RS232; 1-RS485
        //baudrate:     0-9600; 1-19200; 2-38400; 3-57600; 4-115200
        if (ptrFunc(0,1)) ShowMessage("Ok");
        else ShowMessage("error in function CMD_Infs_WK_SET");
    }
    else
    {
        ShowMessage(SysErrorMessage(GetLastError()));
    }
}
else
    ShowMessage("library not loaded.\n"+SysErrorMessage(GetLastError()));

```

//-----

CMD_Infs_WK_GET

```

typedef bool __declspec(dllimport) CMD_Infs_WK_GET(char &IntfNum, char &BaudRate,
char &Mode);
CMD_Infs_WK_GET *ptrFunc;
if (Dll)
{
    ptrFunc = (CMD_Infs_WK_GET *)GetProcAddress(Dll, "_CMD_Infs_WK_GET");
    if (ptrFunc)
    {
        char interface_num, baud, mode;
        //interface:    0-RS232; 1-RS485
        //baudrate:     0-9600; 1-19200; 2-38400; 3-57600; 4-115200
        //WAKE mode:    0-binary; 1-symbol
    }
}

```

```

        if (ptrFunc(interface_num,baud,mode)) ShowMessage("interface =
0x"+IntToHex(interface_num,2)
                                                    +"\nbaud =
0x"+IntToHex(baud,2)
                                                    +"\nmode =
0x"+IntToHex(mode,2));
        else ShowMessage("error in function CMD_Infs_WK");
    }
    else
    {
        ShowMessage(SysErrorMessage(GetLastError()));
    }
}
else
    ShowMessage("library not loaded.\n"+SysErrorMessage(GetLastError()));
//-----

```

0x4d CMD_Dig_Out

CMD_Dig_Out_SET

```

typedef bool __declspec(dllimport) CMD_Dig_Out_SET(char &val);
CMD_Dig_Out_SET *ptrFunc;
if (Dll)
{
    ptrFunc = (CMD_Dig_Out_SET *)GetProcAddress(Dll, "_CMD_Dig_Out_SET");
    if (ptrFunc)
    {
        char NewVal = 1; //0-disable; 1-enable
        if (ptrFunc(NewVal)) ShowMessage("Digital output = 0x"+IntToHex(NewVal,2));
        else ShowMessage("error in function CMD_Dig_Out_SET");
    }
    else
    {
        ShowMessage(SysErrorMessage(GetLastError()));
    }
}
else
    ShowMessage("library not loaded.\n"+SysErrorMessage(GetLastError()));
//-----

```

CMD_Dig_Out_GET

```

typedef bool __declspec(dllimport) CMD_Dig_Out_GET(char &val);
CMD_Dig_Out_GET *ptrFunc;
if (Dll)
{
    ptrFunc = (CMD_Dig_Out_GET *)GetProcAddress(Dll, "_CMD_Dig_Out_GET");
    if (ptrFunc)
    {
        char CurrentVal; //0-disable; 1-enable
        if (ptrFunc(CurrentVal)) ShowMessage("Digital output =
0x"+IntToHex(CurrentVal,2));
        else ShowMessage("error in function CMD_Dig_Out_GET");
    }
    else
    {
        ShowMessage(SysErrorMessage(GetLastError()));
    }
}
else
    ShowMessage("library not loaded.\n"+SysErrorMessage(GetLastError()));

```

//-----

0x4e CMD_Dig_In

CMD_Dig_In_SET

```

typedef bool __declspec(dllimport) CMD_Dig_In_SET(char &ChN, char &ProgNum);
CMD_Dig_In_SET *ptrFunc;
if (Dll)
{
    ptrFunc = (CMD_Dig_In_SET *)GetProcAddress(Dll, "_CMD_Dig_In_SET");
    if (ptrFunc)
    {
        char ChN = 0;    // channel number
        char ProgNum = 0x01;
                        // D0..D3 number of program, which run on falling
                        // D4..D7 number of program, which run on rising
        if (ptrFunc(ChN,ProgNum)) ShowMessage("Channel #"+IntToStr(ChN)+
                                                "\nRun program on falling = "
+IntToStr(ProgNum & 0x0F)
                                                "\nRun program on rising = "
+IntToStr(ProgNum >> 4));
        else ShowMessage("error in function CMD_Dig_In_SET");
    }
    else
    {
        ShowMessage(SysErrorMessage(GetLastError()));
    }
}
else
    ShowMessage("library not loaded.\n"+SysErrorMessage(GetLastError()));

```

//-----

CMD_Dig_In_GET

```

typedef bool __declspec(dllimport) CMD_Dig_In_GET(char &ChN, char &ProgNum);
CMD_Dig_In_GET *ptrFunc;
if (Dll)
{
    ptrFunc = (CMD_Dig_In_GET *)GetProcAddress(Dll, "_CMD_Dig_In_GET");
    if (ptrFunc)
    {
        char ChN = 0;    // channel number
        char ProgNum;    // D0..D3 number of program, which run on falling
                        // D4..D7 number of program, which run on rising
        if (ptrFunc(ChN,ProgNum)) ShowMessage("Channel #"+IntToStr(ChN)+
                                                "\nRun program on falling = "
+IntToStr(ProgNum & 0x0F)
                                                "\nRun program on rising = "
+IntToStr(ProgNum >> 4));
        else ShowMessage("error in function CMD_Dig_In_GET");
    }
    else
    {
        ShowMessage(SysErrorMessage(GetLastError()));
    }
}
else
    ShowMessage("library not loaded.\n"+SysErrorMessage(GetLastError()));

```

//-----

Commands of Work with ADC

//-----

0x10 CMD_ClbrADC

```
typedef bool __declspec(dllimport) CMD_ClbrADC(char Num, char Type);
CMD_ClbrADC *ptrFunc;
if (Dll)
{
    ptrFunc = (CMD_ClbrADC *)GetProcAddress(Dll, "_CMD_ClbrADC");
    if (ptrFunc)
    {
        // for example: calibration offset & gain ADC chn5 (temperature Channel #1)
        if (ptrFunc(5,1)) ShowMessage("Ok");
        else ShowMessage("error in function CMD_ClbrADC ");
    }
    else
    {
        ShowMessage(SysErrorMessage(GetLastError()));
    }
}
else
    ShowMessage("library not loaded.\n"+SysErrorMessage(GetLastError()));
//-----
```

0x11 CMD_ClbrK_ADC

```
typedef bool __declspec(dllimport) CMD_ClbrK_ADC(char Num, float Value);
CMD_ClbrK_ADC *ptrFunc;
if (Dll)
{
    ptrFunc = (CMD_ClbrK_ADC *)GetProcAddress(Dll, "_CMD_ClbrK_ADC");
    if (ptrFunc)
    {
        // for example: calibration ADC chn3 (I=2.12A Channel #1)
        if (ptrFunc(3,2.12)) ShowMessage("Ok");
        else ShowMessage("error in function CMD_ClbrK_ADC");
    }
    else
    {
        ShowMessage(SysErrorMessage(GetLastError()));
    }
}
else
    ShowMessage("library not loaded.\n"+SysErrorMessage(GetLastError()));
//-----
```

0x14 CMD_AskKADC

```
typedef bool __declspec(dllimport) CMD_AskKADC(char Num, float &CalibC
                                                , char &FilterC
                                                , char &Pga);

CMD_AskKADC *ptrFunc;
if (Dll)
{
    ptrFunc = (CMD_AskKADC *)GetProcAddress(Dll, "_CMD_AskKADC");
    if (ptrFunc)
    {
        // for example: request of ADC input (ch5) factor (temperature Channel #1)
        float val; char filter, pga;
        if (ptrFunc(5, val, filter, pga))
            ShowMessage("val="+FormatFloat("0.000000e+00", val)+
                        "\nfilter= 0x"+IntToHex((int)filter, 2)+
                        "\npga= 0x"+IntToHex((int)pga, 2));
        else ShowMessage("error in function CMD_AskKADC");
    }
    else
    {
        ShowMessage(SysErrorMessage(GetLastError()));
    }
}
else
    ShowMessage("library not loaded.\n"+SysErrorMessage(GetLastError()));
//-----
```

0x15 CMD_AskOfst

```
typedef bool __declspec(dllimport) CMD_AskOfst(char Num, char &r1
                                                , char &r2
                                                , char &r3);

CMD_AskOfst *ptrFunc;
if (Dll)
{
    ptrFunc = (CMD_AskOfst *)GetProcAddress(Dll, "_CMD_AskOfst");
    if (ptrFunc)
    {
        // for example: registers of ADC chn5 offset (temperature Channel #1)
        char r1,r2,r3;
        if (ptrFunc(5,r1,r2,r3)) ShowMessage("r1= 0x" +IntToHex(r1,2)+
                                             "\nr2= 0x"+IntToHex(r2,2)+
                                             "\nr3= 0x"+IntToHex(r3,2));

        else ShowMessage("error in function CMD_AskOfst");
    }
    else
    {
        ShowMessage(SysErrorMessage(GetLastError()));
    }
}
else
    ShowMessage("library not loaded.\n"+SysErrorMessage(GetLastError()));

//-----
```

0x16 CMD_StartADC

```
typedef bool __declspec(dllimport) CMD_StartADC(char ChN,DWORD &hexVal,
                                                float &RVal, float &TVal);

CMD_StartADC *ptrFunc;
if (Dll)
{
    ptrFunc = (CMD_StartADC *)GetProcAddress(Dll, "_CMD_StartADC");
    if (ptrFunc)
    {
        // for example: measurement ADC chn5 (temperature Channel #1)
        DWORD dw; float f1,f2;
        if (ptrFunc(5,dw,f1,f2))
            ShowMessage("hex val = 0x" +IntToHex((__int64)dw,8)+
                        "\nRVal = "+FormatFloat("0.000000e+00",f1)+
                        "\nTVal = "+FormatFloat("0.000000e+00",f2));

        else ShowMessage("error in function CMD_StartADC");
    }
    else
    {
        ShowMessage(SysErrorMessage(GetLastError()));
    }
}
else
    ShowMessage("library not loaded.\n"+SysErrorMessage(GetLastError()));

//-----
```

0x17 CMD_Only_1

CMD_Only_1_GET

```
typedef bool __declspec(dllimport) CMD_Only_1_GET(char &val);

CMD_Only_1_GET *ptrFunc;
if (Dll)
{
    ptrFunc = (CMD_Only_1_GET *)GetProcAddress(Dll, "_CMD_Only_1_GET");
    if (ptrFunc)
    {
        char val = 0;
        if (ptrFunc(val)) ShowMessage("Boost mode = 0x"+IntToHex(val,2));
        else ShowMessage("error in function CMD_Only_1_GET");
    }
    else
    {
        ShowMessage(SysErrorMessage(GetLastError()));
    }
}
else
```

```

        ShowMessage("library not loaded.\n"+SysErrorMessage(GetLastError()));
//-----

```

CMD_Only_1_SET

```

typedef bool __declspec(dllimport) CMD_Only_1_SET(char Ch, bool bOnOff);
CMD_Only_1_SET *ptrFunc;
if (Dll)
{
    ptrFunc = (CMD_Only_1_SET *)GetProcAddress(Dll, "_CMD_Only_1_SET");
    if (ptrFunc)
    {
        char ChN = 5;        // ADC channel number
        bool bVal = true;
        if (ptrFunc(ChN, bVal)) ShowMessage("Ok");
        else ShowMessage("error in function CMD_Only_1_SET");
    }
    else
    {
        ShowMessage(SysErrorMessage(GetLastError()));
    }
}
else
    ShowMessage("library not loaded.\n"+SysErrorMessage(GetLastError()));
//-----

```

0x18 CMD_Seaver

CMD_Seaver_GET

```

typedef bool __declspec(dllimport) CMD_Seaver_GET(char &Mask);
CMD_Seaver_GET *ptrFunc;
if (Dll)
{
    ptrFunc = (CMD_Seaver_GET *)GetProcAddress(Dll, "_CMD_Seaver_GET");
    if (ptrFunc)
    {
        char mask;
        // mask      ADC channel
        // -----
        // 0x01      supply voltage
        // 0x02      TEC 1 voltage
        // 0x04      TEC 2 voltage
        // 0x08      TEC 1 current
        // 0x10      TEC 2 current
        // 0x20      TEC 1 temperature
        // 0x40      TEC 2 temperature
        if (ptrFunc(mask)) ShowMessage("Mask = 0x"+IntToHex(mask,2));
        else ShowMessage("error in function CMD_Seaver_GET");
    }
    else
    {
        ShowMessage(SysErrorMessage(GetLastError()));
    }
}
else
    ShowMessage("library not loaded.\n"+SysErrorMessage(GetLastError()));
//-----

```

CMD_Seaver_SET

```

typedef bool __declspec(dllimport) CMD_Seaver_SET(char Mask);
CMD_Seaver_SET *ptrFunc;
if (Dll)
{
    ptrFunc = (CMD_Seaver_SET *)GetProcAddress(Dll, "_CMD_Seaver_SET");
    if (ptrFunc)
    {
        char mask = 0x7F;
        // mask      ADC channel
        // -----
        // 0x01      supply voltage
        // 0x02      TEC 1 voltage
        // 0x04      TEC 2 voltage

```

```

    // 0x08    TEC 1 current
    // 0x10    TEC 2 current
    // 0x20    TEC 1 temperature
    // 0x40    TEC 2 temperature
    if (ptrFunc(mask)) ShowMessage("Ok");
    else ShowMessage("error in function CMD_Seaver_SET");
}
else
{
    ShowMessage(SysErrorMessage(GetLastError()));
}
}
else
    ShowMessage("library not loaded.\n"+SysErrorMessage(GetLastError()));
//-----

```

0x19 CMD_PGA

```

typedef bool __declspec(dllimport)  CMD_PGA(char Num, char Pga);
CMD_PGA *ptrFunc;
if (Dll)
{
    ptrFunc = (CMD_PGA *)GetProcAddress(Dll, "_CMD_PGA");
    if (ptrFunc)
    {
        // for example: set PGA ADC chn5 (temperature Channel #1)
        if (ptrFunc(5,0x04)) ShowMessage("Ok");
        else ShowMessage("error in function CMD_PGA");
    }
    else
    {
        ShowMessage(SysErrorMessage(GetLastError()));
    }
}
else
    ShowMessage("library not loaded.\n"+SysErrorMessage(GetLastError()));
//-----

```

0x1a CMD_Polynom

CMD_Polynom_Power

```

typedef bool __declspec(dllimport)  CMD_Polynom_Power(char ChN,
                                                    unsigned char Power) ;
CMD_Polynom_Power *ptrFunc;
if (Dll)
{
    ptrFunc = (CMD_Polynom_Power *)GetProcAddress(Dll, "_CMD_Polynom_Power");
    if (ptrFunc)
    {
        // for example: set Channel #1 polynomial function type
        if (ptrFunc(0,5)) ShowMessage("Ok");
        else ShowMessage("error in function CMD_Polynom_Power");
    }
    else
    {
        ShowMessage(SysErrorMessage(GetLastError()));
    }
}
else
    ShowMessage("library not loaded.\n"+SysErrorMessage(GetLastError()));

```

```
//-----
```

CMD_Polynom_Coeff

```
typedef bool __declspec(dllimport) CMD_Polynom_Coeff(char ChN,
                                                    unsigned char PowerIndex, float C);

CMD_Polynom_Coeff *ptrFunc;
if (Dll)
{
    ptrFunc = (CMD_Polynom_Coeff *)GetProcAddress(Dll, "_CMD_Polynom_Coeff");
    if (ptrFunc)
    {
        // for example: set Channel #1 polynomial coeff. A0=3.46180e+01
        if (ptrFunc(0,0,3.46180e+01)) ShowMessage("Ok");
        else ShowMessage("error in function CMD_Polynom_Coeff");
    }
    else
    {
        ShowMessage(SysErrorMessage(GetLastError()));
    }
}
else
    ShowMessage("library not loaded.\n"+SysErrorMessage(GetLastError()));
//-----
```

0x1b CMD_ask_Pol

```
typedef bool __declspec(dllimport) CMD_ask_Pol(char ChN, unsigned char &Power,
                                                unsigned char PowerIndex, float &C);

CMD_ask_Pol *ptrFunc;
if (Dll)
{
    ptrFunc = (CMD_ask_Pol *)GetProcAddress(Dll, "_CMD_ask_Pol");
    if (ptrFunc)
    {
        // for example: get Channel #1 polynomial coeff. A2
        unsigned char Power; float fVal;
        if (ptrFunc(0,Power,2,fVal)) ShowMessage("Power = " + IntToStr(Power) +
                                                "\nA2 = " +
FormatFloat("0.00000e+00",fVal));
        else ShowMessage("error in function CMD_ask_Pol");
    }
    else
    {
        ShowMessage(SysErrorMessage(GetLastError()));
    }
}
else
    ShowMessage("library not loaded.\n"+SysErrorMessage(GetLastError()));
//-----
```

0x1c CMD_saveTerm

```
typedef bool __declspec(dllimport) CMD_saveTerm(char ChN);

CMD_saveTerm *ptrFunc;
if (Dll)
{
    ptrFunc = (CMD_saveTerm *)GetProcAddress(Dll, "_CMD_saveTerm");
    if (ptrFunc)
    {
        // save current settings into thermistors table
        if (ptrFunc(0)) ShowMessage("Ok");
        else ShowMessage("error in function CMD_saveTerm");
    }
    else
    {
        ShowMessage(SysErrorMessage(GetLastError()));
    }
}
else
    ShowMessage("library not loaded.\n"+SysErrorMessage(GetLastError()));
```

```
//-----
```

0x1d CMD_loadTerm

```
typedef bool __declspec(dllimport) CMD_loadTerm(char ChN, char StringNum);
CMD_loadTerm *ptrFunc;
if (Dll)
{
    ptrFunc = (CMD_loadTerm *)GetProcAddress(Dll, "_CMD_loadTerm");
    if (ptrFunc)
    { // select thermistor settings from table (line 0C) & reset
      // This command causes the reboot controller
      if (ptrFunc(0,0x0C)) ShowMessage("Ok");
      else ShowMessage("error in function CMD_loadTerm");
    }
    else
    {
        ShowMessage(SysErrorMessage(GetLastError()));
    }
}
else
    ShowMessage("library not loaded.\n"+SysErrorMessage(GetLastError()));
//-----
```

0x1e CMD_get_TBL

```
typedef bool __declspec(dllimport) CMD_get_TBL(char* OutputFileDir);
CMD_get_TBL *ptrFunc;
if (Dll)
{
    ptrFunc = (CMD_get_TBL *)GetProcAddress(Dll, "_CMD_get_TBL");
    if (ptrFunc)
    { // save thermistor file
      if (ptrFunc("c:\\temp")) ShowMessage("Ok");
      else ShowMessage("error in function CMD_get_TBL");
    }
    else
    {
        ShowMessage(SysErrorMessage(GetLastError()));
    }
}
else
    ShowMessage("library not loaded.\n"+SysErrorMessage(GetLastError()));
//-----
```

0x1f CMD_set_TBL

```
typedef bool __declspec(dllimport) CMD_set_TBL(char* FileName);
CMD_set_TBL *ptrFunc;
if (Dll)
{
    ptrFunc = (CMD_set_TBL *)GetProcAddress(Dll, "_CMD_set_TBL");
    if (ptrFunc)
    { // save thermistor file
      if (ptrFunc("c:\\temp\\#S213 14.05.2012 (Thermistor table).txt"))
ShowMessage("Ok");
      else ShowMessage("error in function CMD_set_TBL");
    }
    else
    {
        ShowMessage(SysErrorMessage(GetLastError()));
    }
}
else
    ShowMessage("library not loaded.\n"+SysErrorMessage(GetLastError()));
```

Commands of work with DAC

0x21 CMD_set_DAC

```
typedef bool __declspec(dllimport) CMD_set_DAC(char ChN, float Voltage);
CMD_set_DAC *ptrFunc;
if (Dll)
{
    ptrFunc = (CMD_set_DAC *)GetProcAddress(Dll, "_CMD_set_DAC");
    if (ptrFunc)
    {
        // set Channel#1 1V
        // When the command is executed, the set voltage does not exceed
        // maximal for a chosen channel.

        if (ptrFunc(0,1.0)) ShowMessage("Ok");
        else ShowMessage("error in function CMD_set_DAC");
    }
    else
    {
        ShowMessage(SysErrorMessage(GetLastError()));
    }
}
else
    ShowMessage("library not loaded.\n"+SysErrorMessage(GetLastError()));
//-----
```

0x22 CMD_seth_DAC

```
typedef bool __declspec(dllimport) CMD_seth_DAC(char ChN, unsigned short Mode);
CMD_seth_DAC *ptrFunc;
if (Dll)
{
    ptrFunc = (CMD_seth_DAC *)GetProcAddress(Dll, "_CMD_seth_DAC");
    if (ptrFunc)
    {
        // Attention! Check for excess of maximal voltage is not done!
        // The command is used for calibration.
        // It should be used at voltage up to 8 V.

        if (ptrFunc(0,1500)) ShowMessage("Ok");
        else ShowMessage("error in function CMD_seth_DAC");
    }
    else
    {
        ShowMessage(SysErrorMessage(GetLastError()));
    }
}
else
    ShowMessage("library not loaded.\n"+SysErrorMessage(GetLastError()));
//-----
```

0x23 CMD_Wr_K_DAC

```
typedef bool __declspec(dllimport) CMD_Wr_K_DAC(char ChN, float A0, float A1);
CMD_Wr_K_DAC *ptrFunc;
if (Dll)
{
    ptrFunc = (CMD_Wr_K_DAC *)GetProcAddress(Dll, "_CMD_Wr_K_DAC");
    if (ptrFunc)
    {
        // The coefficients determine the function by which a value
        // loaded to DAC is obtained, depending on voltage needed.

        if (ptrFunc(0,6.116621E+004, -7.935489E+003)) ShowMessage("Ok");
        else ShowMessage("error in function CMD_Wr_K_DAC");
    }
    else
    {
        ShowMessage(SysErrorMessage(GetLastError()));
    }
}
else
    ShowMessage("library not loaded.\n"+SysErrorMessage(GetLastError()));
//-----
```

0x24 CMD_AskKDAC

```
typedef bool __declspec(dllimport) CMD_AskKDAC(char ChN, float &A0, float &A1, float
&MAX);
CMD_AskKDAC *ptrFunc;
if (Dll)
{
    ptrFunc = (CMD_AskKDAC *)GetProcAddress(Dll, "_CMD_AskKDAC");
    if (ptrFunc)
    { // Sending Coefficient and DAC Maximal Values
        float a0,a1,max;
        if (ptrFunc(0,a0,a1,max))
            ShowMessage("A0 = " + FormatFloat("0.000000e+00",a0) +
                "\nA1 = " + FormatFloat("0.000000e+00",a1) +
                "\nMax = " + FormatFloat("0.000000e+00",max));
        else ShowMessage("error in function CMD_AskKDAC");
    }
    else
    {
        ShowMessage(SysErrorMessage(GetLastError()));
    }
}
else
    ShowMessage("library not loaded.\n"+SysErrorMessage(GetLastError()));
//-----
```

0x25 CMD_DAC_max

```
typedef bool __declspec(dllimport) CMD_DAC_max(char ChN, float Value);
CMD_DAC_max *ptrFunc;
if (Dll)
{
    ptrFunc = (CMD_DAC_max *)GetProcAddress(Dll, "_CMD_DAC_max");
    if (ptrFunc)
    { // Writing Maximal Allowable Voltage
        if (ptrFunc(0,4.5)) ShowMessage("Ok");
        else ShowMessage("error in function CMD_DAC_max");
    }
    else
    {
        ShowMessage(SysErrorMessage(GetLastError()));
    }
}
else
    ShowMessage("library not loaded.\n"+SysErrorMessage(GetLastError()));
//-----
```

0x26 CMD_U_Treg

CMD_U_Treg_GET

```
typedef bool __declspec(dllimport) CMD_U_Treg_GET(char ChN, float &Value);
CMD_U_Treg_GET *ptrFunc;
if (Dll)
{
    ptrFunc = (CMD_U_Treg_GET *)GetProcAddress(Dll, "_CMD_U_Treg_GET");
    if (ptrFunc)
    { // Voltage of T-reg
        float fVal;
        if (ptrFunc(0,fVal))
            ShowMessage("T-reg value = " + FormatFloat("0.00",fVal));
        else ShowMessage("error in function CMD_U_Treg_GET");
    }
    else
    {
        ShowMessage(SysErrorMessage(GetLastError()));
    }
}
else
    ShowMessage("library not loaded.\n"+SysErrorMessage(GetLastError()));
//-----
```

CMD_U_Treg_SET

```

typedef bool __declspec(dllimport) CMD_U_Treg_SET(char ChN, float Value) ;
CMD_U_Treg_SET *ptrFunc;
if (Dll)
{
    ptrFunc = (CMD_U_Treg_SET *)GetProcAddress(Dll, "_CMD_U_Treg_SET");
    if (ptrFunc)
    { // Voltage of T-reg
        float fVal;
        if (ptrFunc(0,2.0)) ShowMessage("Ok");
        else ShowMessage("error in function CMD_U_Treg_SET");
    }
    else
    {
        ShowMessage(SysErrorMessage(GetLastError()));
    }
}
else
    ShowMessage("library not loaded.\n"+SysErrorMessage(GetLastError()));
//-----

```

Commands of work with PID controller

0x30 CMD_Pol_TEC

```

typedef bool __declspec(dllimport) CMD_Pol_TEC(char ChN, char Mode);
CMD_Pol_TEC *ptrFunc;
if (Dll)
{
    ptrFunc = (CMD_Pol_TEC *)GetProcAddress(Dll, "_CMD_Pol_TEC");
    if (ptrFunc) // Setting of TEC Polarity
    { // 0- TEC is off
      // 1- TEC is heating
      // 2- TEC is cooling

        if (ptrFunc(0,2)) ShowMessage("Ok");
        else ShowMessage("error in function CMD_Pol_TEC");
    }
    else
    {
        ShowMessage(SysErrorMessage(GetLastError()));
    }
}
else
    ShowMessage("library not loaded.\n"+SysErrorMessage(GetLastError()));
//-----

```

0x31 CMD_set_PID

```

typedef bool __declspec(dllimport) CMD_set_PID(char ChN,
                                                float Kp,
                                                float Ki, float Kd);

CMD_set_PID *ptrFunc;
if (Dll)
{
    ptrFunc = (CMD_set_PID *)GetProcAddress(Dll, "_CMD_set_PID");
    if (ptrFunc) // Writing Parameters of PID Controller
    {
        if (ptrFunc(0,2.353643, 0.108696, 0.230000)) ShowMessage("Ok");
        else ShowMessage("error in function CMD_set_PID");
    }
    else
    {
        ShowMessage(SysErrorMessage(GetLastError()));
    }
}
else
    ShowMessage("library not loaded.\n"+SysErrorMessage(GetLastError()));
//-----

```

0x32 CMD_ask_PID

```
typedef bool __declspec(dllimport) CMD_ask_PID(char ChN,
                                                float &Kp,
                                                float &Ki, float &Kd);

CMD_ask_PID *ptrFunc;
if (Dll)
{
    ptrFunc = (CMD_ask_PID *)GetProcAddress(Dll, "_CMD_ask_PID");
    if (ptrFunc) // Read Parameters of PID Controller
    {
        float p,i,d;
        if (ptrFunc(0,p,i,d))
            ShowMessage("Kp = " + FormatFloat("0.000000e+00",p) +
                        "\nKi = " + FormatFloat("0.000000e+00",i) +
                        "\nKd = " + FormatFloat("0.000000e+00",d));
        else ShowMessage("error in function CMD_ask_PID");
    }
    else
    {
        ShowMessage(SysErrorMessage(GetLastError()));
    }
}
else
    ShowMessage("library not loaded.\n"+SysErrorMessage(GetLastError()));
//-----
```

0x33 CMD_setCurrT

CMD_setCurrT_GET

```
typedef bool __declspec(dllimport) CMD_setCurrT_GET(char ChN, unsigned char
&Current);
CMD_setCurrT_GET *ptrFunc;
if (Dll)
{
    ptrFunc = (CMD_setCurrT_GET *)GetProcAddress(Dll, "_CMD_setCurrT_GET");
    if (ptrFunc) // Get thermistor current
    {
        unsigned char res;
        if (ptrFunc(0,res))
            ShowMessage("thermistor current = " + IntToStr(res));
        else ShowMessage("error in function CMD_setCurrT_GET");
    }
    else
    {
        ShowMessage(SysErrorMessage(GetLastError()));
    }
}
else
    ShowMessage("library not loaded.\n"+SysErrorMessage(GetLastError()));
//-----
```

CMD_setCurrT_SET

```
typedef bool __declspec(dllimport) CMD_setCurrT_SET(char ChN, unsigned char
Current);
CMD_setCurrT_SET *ptrFunc;
if (Dll)
{
    ptrFunc = (CMD_setCurrT_SET *)GetProcAddress(Dll, "_CMD_setCurrT_SET");
    if (ptrFunc) // Set thermistor current
    {
        if (ptrFunc(0,1))
            ShowMessage("Ok");
        else ShowMessage("error in function CMD_setCurrT_SET");
    }
    else
    {
        ShowMessage(SysErrorMessage(GetLastError()));
    }
}
else
    ShowMessage("library not loaded.\n"+SysErrorMessage(GetLastError()));
```

```
//-----
```

0x34 CMD_askT_PID

CMD_askT_PID_GET

```
typedef bool __declspec(dllimport) CMD_askT_PID_GET(char ChN, float &TSet,
                                                    float &dTSet, char &d1, char &d2) ;

CMD_askT_PID_GET *ptrFunc;
if (Dll)
{
    ptrFunc = (CMD_askT_PID_GET *)GetProcAddress(Dll, "_CMD_askT_PID_GET");
    if (ptrFunc) // Sending/Set Setpoints of PID Controller
    {
        float TSet, dTSet; char d1, d2;
        if (ptrFunc(0,TSet, dTSet, d1, d2))
            ShowMessage("TSetpoint = " + FormatFloat("0.00",TSet) +
                        "\ndTSet = " + FormatFloat("0.00",dTSet) +
                        "\nd1 = " + IntToStr(d1) +
                        "\nd2 = " + IntToStr(d2));
        else ShowMessage("error in function CMD_askT_PID_GET");
    }
    else
    {
        ShowMessage(SysErrorMessage(GetLastError()));
    }
}
else
    ShowMessage("library not loaded.\n"+SysErrorMessage(GetLastError()));
//-----
```

CMD_askT_PID_SET

```
typedef bool __declspec(dllimport) CMD_askT_PID_SET(char ChN, float &TSet,
                                                    float &dTSet, char &d1, char &d2) ;

CMD_askT_PID_SET *ptrFunc;
if (Dll)
{
    ptrFunc = (CMD_askT_PID_SET *)GetProcAddress(Dll, "_CMD_askT_PID_SET");
    if (ptrFunc) // Set Setpoints of PID Controller
    {
        float TSet, dTSet; char d1, d2;

        TSet = 270.15; //Value of setpoint temperature of PID
                       //(setting value without restarting the PID).

        if (ptrFunc(0,TSet, dTSet, d1, d2))
            ShowMessage("TSetpoint = " + FormatFloat("0.00",TSet) +
                        "\ndTSet = " + FormatFloat("0.00",dTSet) +
                        "\nd1 = " + IntToStr(d1) +
                        "\nd2 = " + IntToStr(d2));
        else ShowMessage("error in function CMD_askT_PID_SET");
    }
    else
    {
        ShowMessage(SysErrorMessage(GetLastError()));
    }
}
else
    ShowMessage("library not loaded.\n"+SysErrorMessage(GetLastError()));
```

```
//-----
```

0x35 CMD_strt_PID

```
typedef bool __declspec(dllimport) CMD_strt_PID(char ChN, char Mode, float Value);
CMD_strt_PID *ptrFunc;
if (Dll)
{
    ptrFunc = (CMD_strt_PID *)GetProcAddress(Dll, "_CMD_strt_PID");
    if (ptrFunc) // Starting Controller
    {
        //      0 Regulation stop
        //      1 Regulation according to program
        //      2 T-regulation
        //      3 Temperature maintenance - PID starting - setting regulation
        //      4 Constant voltage

        if (ptrFunc(0,3,273.15))
            ShowMessage("Ok");
        else ShowMessage("error in function CMD_strt_PID");
    }
    else
    {
        ShowMessage(SysErrorMessage(GetLastError()));
    }
}
else
    ShowMessage("library not loaded.\n"+SysErrorMessage(GetLastError()));
//-----
```

0x37 CMD_Zmetr

```
typedef bool __declspec(dllimport) CMD_Zmetr(char ChN, char Time, char AdvParam);
CMD_Zmetr *ptrFunc;
if (Dll)
{
    ptrFunc = (CMD_Zmetr *)GetProcAddress(Dll, "_CMD_Zmetr");
    if (ptrFunc) // Starting Z-meter
    {
        //TEC channel number
        //Z-meter measurement time (s) 20...255
        //if "1" - only resistant measurement

        if (ptrFunc(0,20,0))
            ShowMessage("Ok");
        else ShowMessage("error in function CMD_Zmetr");
    }
    else
    {
        ShowMessage(SysErrorMessage(GetLastError()));
    }
}
else
    ShowMessage("library not loaded.\n"+SysErrorMessage(GetLastError()));
//-----
```

0x38 CMD_Zprmtr

```
typedef bool __declspec(dllimport) CMD_Zprmtr();
CMD_Zprmtr *ptrFunc;
if (Dll)
{
    ptrFunc = (CMD_Zprmtr *)GetProcAddress(Dll, "_CMD_Zprmtr");
    if (ptrFunc) // Storage of Z-Metering Parameters
    {
        // By this command the found parameters are stored as
        // reference ones reference for the given object
        if (ptrFunc())
            ShowMessage("Ok");
        else ShowMessage("error in function CMD_Zprmtr");
    }
    else
    {
        ShowMessage(SysErrorMessage(GetLastError()));
    }
}
```

```

    }
}
else
    ShowMessage("library not loaded.\n"+SysErrorMessage(GetLastError()));
//-----

```

0x39 CMD_Z_I

CMD_Z_I_GET

```

typedef bool __declspec(dllimport) CMD_Z_I_GET(float &I);
CMD_Z_I_GET *ptrFunc;
if (Dll)
{
    ptrFunc = (CMD_Z_I_GET *)GetProcAddress(Dll, "_CMD_Z_I_GET");
    if (ptrFunc) //Get Z-Meter Current
    {
        float i;
        if (ptrFunc(i))
            ShowMessage("I = " + FormatFloat("0.0000e+00",i));
        else ShowMessage("error in function CMD_Z_I_GET");
    }
    else
    {
        ShowMessage(SysErrorMessage(GetLastError()));
    }
}
else
    ShowMessage("library not loaded.\n"+SysErrorMessage(GetLastError()));
//-----

```

CMD_Z_SET

```

typedef bool __declspec(dllimport) CMD_Z_I_SET(float I);
CMD_Z_I_SET *ptrFunc;
if (Dll)
{
    ptrFunc = (CMD_Z_I_SET *)GetProcAddress(Dll, "_CMD_Z_I_SET");
    if (ptrFunc) //Set Z-Meter Current
    {
        //By this command the value of calculated current is stored
        //in the non-volatile memory and used for Z-metering calculations
        if (ptrFunc(4.985)) ShowMessage("Ok");
        else ShowMessage("error in function CMD_Z_I_SET");
    }
    else
    {
        ShowMessage(SysErrorMessage(GetLastError()));
    }
}
else
    ShowMessage("library not loaded.\n"+SysErrorMessage(GetLastError()));
//-----

```

0x3b CMD_Boot

CMD_Boot_GET

```

typedef bool __declspec(dllimport) CMD_Boot_GET(char ChN, char &Mode,
                                                float &Value,unsigned short &Delay);

CMD_Boot_GET *ptrFunc;
if (Dll)
{
    ptrFunc = (CMD_Boot_GET *)GetProcAddress(Dll, "_CMD_Boot_GET");
    if (ptrFunc) // Switching On Regulation after Restarting
    {
        // TEC channel number
        // 3 4
        // 0 Regulation stop Not present
        // 1 Regulation according to program 0...15 program to go to
        // 2 T-regulation Temperature to be maintained
        // 3 Temperature maintenance -
        // PID starting - Temperature to be maintained
        // setting regulation (setting value)
    }
}

```

```

// 4 Constant voltage Voltage to be maintained
//
// time (s) after which to proceed to the program
// (only for Regulation according to program)
char mode; float value; unsigned short delay;
    if (ptrFunc(0,mode,value,delay))
        ShowMessage("mode = " + IntToStr(mode) +
                    "\nvalue = " + FormatFloat("0.00",value) +
                    "\ndelay = " + IntToStr(delay));
    else ShowMessage("error in function CMD_Boot_GET");
}
else
{
    ShowMessage(SysErrorMessage(GetLastError()));
}
}
else
    ShowMessage("library not loaded.\n"+SysErrorMessage(GetLastError()));
//-----

```

CMD_Boot_SET

```

typedef bool __declspec(dllimport) CMD_Boot_SET(char ChN, char Mode,
                                                float Value,unsigned short Delay);

CMD_Boot_SET *ptrFunc;
if (Dll)
{
    ptrFunc = (CMD_Boot_SET *)GetProcAddress(Dll, "_CMD_Boot_SET");
    if (ptrFunc) // Switching On Regulation after Restarting
    {
        // TEC channel number
        // 3 4
        // 0 Regulation stop Not present
        // 1 Regulation according to program 0...15 program to go to
        // 2 T-regulation Temperature to be maintained
        // 3 Temperature maintenance -
        // PID starting - Temperature to be maintained
        // setting regulation (setting value)
        // 4 Constant voltage Voltage to be maintained
        //
        // time (s) after which to proceed to the program
        // (only for Regulation according to program)
        if (ptrFunc(0,0,0,0))
            ShowMessage("Ok");
        else ShowMessage("error in function CMD_Boot_SET");
    }
    else
    {
        ShowMessage(SysErrorMessage(GetLastError()));
    }
}
else
    ShowMessage("library not loaded.\n"+SysErrorMessage(GetLastError()));
//-----

```

0x3c CMD_Set_LimT

```

typedef bool __declspec(dllimport) CMD_Set_LimT(char ChN, float Min,
                                                float Max, char D) ;

CMD_Set_LimT *ptrFunc;
if (Dll)
{
    ptrFunc = (CMD_Set_LimT *)GetProcAddress(Dll, "_CMD_Set_LimT");
    if (ptrFunc) // Writing Limiting Temperatures
    {
        if (ptrFunc(0,203,403,10)) ShowMessage("Ok");
        else ShowMessage("error in function CMD_Set_LimT");
    }
    else
    {
        ShowMessage(SysErrorMessage(GetLastError()));
    }
}
else

```

```

        ShowMessage("library not loaded.\n"+SysErrorMessage(GetLastError()));
//-----

```

0x3d CMD_Get_LimT

```

typedef bool __declspec(dllimport) CMD_Get_LimT(char ChN, float &Min,
                                                float &Max, char &D) ;

CMD_Get_LimT *ptrFunc;
if (Dll)
{
    ptrFunc = (CMD_Get_LimT *)GetProcAddress(Dll, "_CMD_Get_LimT");
    if (ptrFunc) // Sending Limiting Temperatures
    {
        float min,max; char d;
        if (ptrFunc(0,min,max,d)) ShowMessage("min = " + FormatFloat("0.00",min)+
                                                "\nmax = " + FormatFloat("0.00",max)+
                                                "\nd = " + IntToStr(d));

        else ShowMessage("error in function CMD_Get_LimT");
    }
    else
    {
        ShowMessage(SysErrorMessage(GetLastError()));
    }
}
else
    ShowMessage("library not loaded.\n"+SysErrorMessage(GetLastError()));
//-----

```

0x3e CMD_ResZmtr

```

typedef bool __declspec(dllimport) CMD_ResZmtr(char &ChN, float &R,
                                                float &Z, float &To) ;

CMD_ResZmtr *ptrFunc;
if (Dll)
{
    ptrFunc = (CMD_ResZmtr *)GetProcAddress(Dll, "_CMD_ResZmtr");
    if (ptrFunc) // Sending Z-metering Results
    {
        char ch = 0;
        float r,z,t;
        if (ptrFunc(ch,r,z,t)) ShowMessage("r = " + FormatFloat("0.00",r)+
                                                "\nz = " + FormatFloat("0.00",z)+
                                                "\nt = " + FormatFloat("0.00",t));

        else ShowMessage("error in function CMD_ResZmtr");
    }
    else
    {
        ShowMessage(SysErrorMessage(GetLastError()));
    }
}
else
    ShowMessage("library not loaded.\n"+SysErrorMessage(GetLastError()));
//-----

```

0x51 CMD_PID_fun

```

typedef bool __declspec(dllimport) CMD_PID_fun(char ChN);

CMD_PID_fun *ptrFunc;
if (Dll)
{
    ptrFunc = (CMD_PID_fun *)GetProcAddress(Dll, "_CMD_PID_fun");
    if (ptrFunc) // Autotuning PID
    {
        // The autotuning function searches the values of proportional,
        // integral and differential coefficients of the PID algorithm.
        if (ptrFunc(0)) ShowMessage("Ok");
        else ShowMessage("error in function CMD_PID_fun");
    }
    else
    {
        ShowMessage(SysErrorMessage(GetLastError()));
    }
}
else

```

```
    ShowMessage("library not loaded.\n"+SysErrorMessage(GetLastError()));  
}  
//-----
```

0x53 CMD_RESET

```
typedef bool __declspec(dllimport) CMD_RESET();  
CMD_RESET *ptrFunc;  
if (Dll)  
{  
    ptrFunc = (CMD_RESET *)GetProcAddress(Dll, "_CMD_RESET");  
    if (ptrFunc) // reset controller  
    {  
        if (ptrFunc()) ShowMessage("Ok");  
        else ShowMessage("error in function CMD_RESET");  
    }  
    else  
    {  
        ShowMessage(SysErrorMessage(GetLastError()));  
    }  
}  
else  
    ShowMessage("library not loaded.\n"+SysErrorMessage(GetLastError()));
```