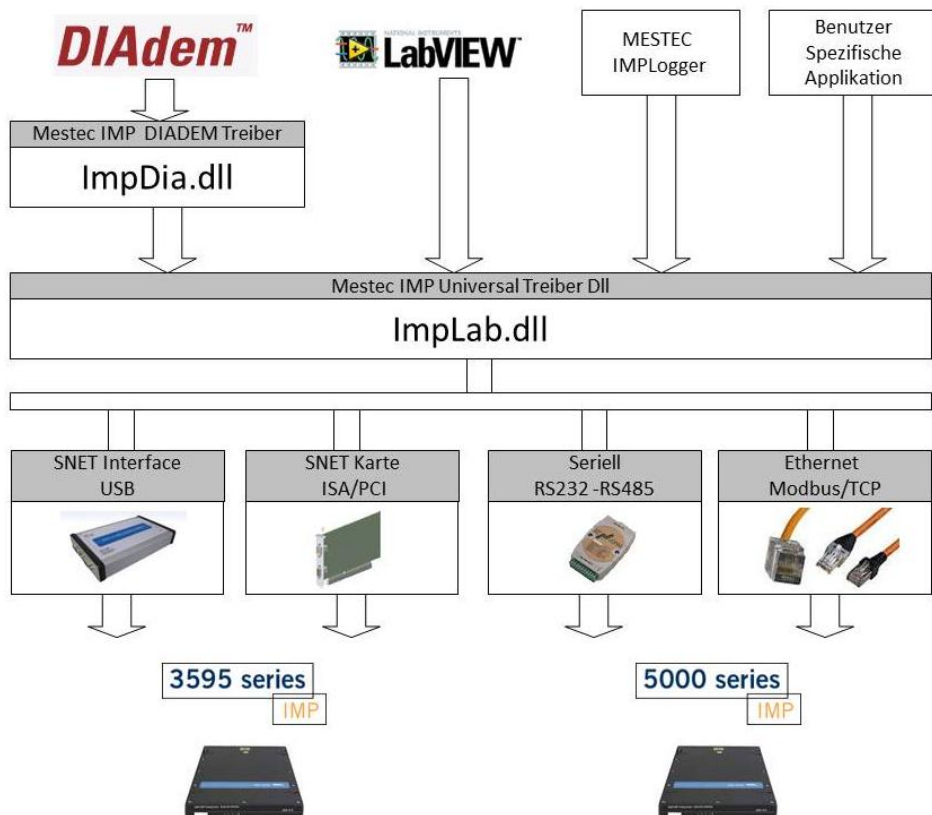


IMP Labview Treiber V5

(C) PMS / MESTEC 2012

Übersicht MESTEC IMP Treiber Version 5



Bedienungshandbuch zu IMP Labview Treiber

Letzte Änderung am 10.April 2012

Autor – Dipl. Phys. Christian Frank

© MESTEC 2012

Inhalt

Übersicht	1
Produkt.....	1
Funktion.....	1
Anwendung.....	1
Fehlerbehandlung	2
Registrierung	2
IMP Modul Übersicht	2
IMP Serie 3595.....	2
IMP Serie 5000.....	3
IMP 5000 Gateway Modus.....	3
Installation	4
Lieferumfang	4
System Voraussetzung.....	5
Hardware Voraussetzung.....	5
Software Voraussetzung	5
Installieren	6
Anwendung	7
LabView	7
Übersicht	7
Beispiel “Demo_ImpLabV5.Vi”.....	7
Bedienung des VI:.....	7
Aufbau des Labview Programms	9
Beispiel für die Einbindung einer Dll Funktion in Labview.....	10
Funktionsreihenfolge	11
C++	11
Beispiel “CallImpLab.EXE”	11
Übersicht.....	15
ImpLab_GetVersion	15
ImpLab_GetDongleInfo.....	15
ImpLab_GetSnetInfo	15
ImpLab_Reset.....	16
ImpLab_SetMeasSrate.....	16
ImpLab_SetIfcTyp.....	17
ImpLab_SetIfcParam	17
ImpLab_SetImp	18
ImpLab_SetImpDefault	19
ImpLab_GetImp	20
ImpLab_SetImpChan.....	20
ImpLab_AddImpChan.....	21
ImpLab_Download	22
ImpLab_StartMeasure	23
ImpLab_StopMeasure.....	23
ImpLab_ReadOutImps.....	24
ImpLab_ReadImpData.....	24
ImpLab_Disconnect.....	25
ImpLab_IsMeasureRunning	25

ImpLab_GetErrorReport.....	25
ImpLab_GetErrorReportSize.....	26
ImpLab_ResetErrorReport.....	26
ImpLab_GetErrorString.....	26
ImpLab_SetDebugMode.....	27
Anhang	28
Kanal Codes für Messung.....	28
Fehlercodes.....	30
TCP/IP Dll.....	31
Solartron DLL.....	31
Glossar	32
Index	33

Übersicht

Produkt

Der IMP Labview Treiber ist eine 32-Bit DLL für Windows zur Konfiguration und Datenerfassung der SOLARTRON IMP Module.

Untertstützt werden die Serien 3595 und 5000.

Der Treiber arbeitet mit SNET-ISA, SNET-PCI, SNET-4U (USB) , Ethernet und Seriellem RS485 Interface.

Funktion

Der Treiber bietet einheitliche Funktionsaufrufe für die Parametrierung der Messung mit den verschiedenen IMP Typen der Serien 3595 und 5000.

Dabei muss sich der Anwender nicht um die unterschiedlichen Interface Typen kümmern. Die Funktionsaufrufe sind einfach in die Anwendung einzubinden.

Das Auslesen der Daten erfordert ebenfalls nur geringen Programmieraufwand.

Mit den Set Funktionen werden für die Interface-Ports, die IMPs und die IMP Kanäle Datenstrukturen angelegt.

Erst mit der Download Funktion werden die Solartron bzw. TCPIP DLL geladen und die Hardwareverbindung getestet. Ebenso werden die Parameter an die IMP-Module gesendet.

Falls der Download fehlerfrei war, wird der Start durchgeführt.

Ab Start können die Meßdaten mit der Funktion ReadDataCopy Impweise abgeholt werden.

Anwendung

Der Treiber kann in beliebige 32-Bit Windows Programme eingebunden werden.

Beispiele sind LabView, LabWindows, MFC C++, C#, Delphi usw.

Für LabView und C++ sind ausführliche Beispiele vorhanden.

Fehlerbehandlung

Zusätzlich zum Rückgabewert der Treiberfunktionen wird jeder interne Fehler in einem Reportstring protokolliert:

Die Fehlerstrings werden, jeweils durch ein Zeilenende-Zeichen getrennt, aneinandergelängt und können mit der Funktion "ImpLab_GetErrorReport()" ausgelesen werden.

Der Reportstring wird durch die Funktion "ImpLab_ResetErrorReport()" gelöscht.

Bei gesetzter Debug Option (Funktion "ImpLab_SetDebug()") werden alle Fehler als Meldungbox angezeigt.

Zu den Rückgabewerten der Funktionen kann mit der Funktion "ImpLab_GetErrorString()" der Fehlerstring angefordert werden.

Eine Liste der Fehlercodes findet sich im Anhang.

Registrierung

Der Treiber ist mit einem USB Dongle geschützt. Für den Betrieb des Dongles muss die Software CBUSETUP von Marx installiert sein (siehe Kunden CD)

IMP Modul Übersicht

IMP Serie 3595

Die Serie 3595 besteht aus unterschiedlichen Modultypen nach folgender Tabelle.

Typ	Kanalzahl	Anwendung
1A	20	Analogeingang - Spg., Strom, Thermopaar - Kanalumschalter mit Halbleiter
1B	10	Analogeingang - Spg., Widerst, DMS - Kanalumschalter mit Relais
1C	20	Analogeingang - Spg., Strom, Thermopaar - Kanalumschalter mit Relais Iso max. 200V
1D	4	Analogausgang - Spg., Strom
1E	20	Analogeingang - Spg., Strom, Thermopaar - Kanalumschalter mit Relais Iso max. 500V
1H	20	Analogeingang - Spg, Widerst, Thermopaar, PT100, Status Ein/Aus (2K) - Kanalumschalter mit Relais Iso max. 200V
1J	20	Analogeingang - Spg, Widerst, Thermopaar, PT100, Status Ein/Aus (2K) - Kanalumschalter mit Relais Iso max. 500V
2A	20	Digital Ein/Aus - Status, Frequenz, Periode
2B	32	Switch - Status

IMP Serie 5000

Die Serie IMP 5000 besteht aus nur einem Modultyp.

Typ	Kanalzahl	Anwendung
1KE	20	Analogeingang - Spg, Widerst, Thermopaar, PT100, Status Ein/Aus (2K)- Kanalumschalter mit Relais Iso max. 500V

Mit der Serie 5000 ist ein Universal Meßmodul hinzugekommen, das im Gegensatz zur 3595 Serie keine spezielles Interfacekarte benötigt.

Damit kann die Messung über jeden PC erfolgen, der einen LAN-Anschluß (Ethernet) oder ein seriellen Anschluß besitzt.

IMP 5000 Gateway Modus

Besonders vorteilhaft ist der Gateway Modus, bei dem das erste IMP 5000 direkt über LAN an den PC verbunden wird. Dieses Imp arbeitet zusätzlich zur Messung als Ethernet / RS485 Gateway, an dem die weiteren IMPs mit einer verdrehten 2-Draht Leitung verbunden sind.

Installation

Lieferumfang

Der IMP Labview Treiber wird als Installations Paket geliefert.
Auf der Installations CD befinden sich die folgenden Dateien:

Datei	Funktion
Installer für MESTEC Labview Treiber	Installationsprogramm für Imp Treiber Setup.Exe
Beispielcode C	Beispielcode in C++ für die Einbindung des Treibers, Verzeichnis für MSC 6.0 und für Visual Studio
Beispiel in Labview	Beispiel VIs für die Einbindung in Labview Beispiel VI für eine komplette Messung in Labview
Software für Marx USB Dongle	Installation für Marx Usb Dongle Treiber
Solartron SNET PCI Treiber	Installationsfiles für den Solartron PCI Karten Basis Treiber

System Voraussetzung

Der Treiber setzt ein Windows Betriebssystem voraus wie XP, Vista oder Windows 7.

Hardware Voraussetzung

Der Treiber unterstutzt die IMP Serien 3595 und 5000.

Fur die 3595 IMPs wird eine SNET Karte ISA, PCI oder USB benotigt.

Fur die 5000 IMPs wird ein Ethernet LAN Anschlu oder eine serielle Schnittstelle RS485 benotigt.

Der RS485 Bus kann auch mit einem RS232/RS485 Konverter oder einem USB / RS485 Konverter an einem Standard PC zur Verfugung gestellt werden.

Software Voraussetzung

Bei den SNET Einbaukarten 3595-4B ISA oder 3595-4C PCI muss der aktuelle Solartron Basistreiber „Impdrv.dll“ installiert sein.

Bei dem SNET-4U USB Interface erfolgt die Kommunikation uber einen virtuellen COM-Port im PC.

Fur die 5000 IMPs mit Ethernet Anschlu wird ein TCP/IP Treiber und das Windows Winsock Interface benotigt:

Der TCPIP Treiber ist Bestandteil des Universal Treibers.

Das Winsock Interface ist Bestandteil des Windows Betriebssystems.

Fur die 5000 IMPs mit Seriellen Anschlu wird kein zusatzlicher Treiber benotigt.

Installieren

Die Installation erfolgt durch Aufruf der Datei "SETUP.EXE" im Ordner „Installer für MESTEC Labview Treiber“ auf der Installations CD.

Danach startet das Installations Programm Inno-Setup .

Nach der Installation befinden sich die Dateien wie unter "Lieferumfang" aufgeführt im benutzerdefinierten Programmverzeichnis bzw. im Windows System Verzeichnis.

Nach der Installation befinden sich folgende Dateien auf dem PC im Programmverzeichnis unter \PMSLTD bzw. im Windows System Verzeichnis.

Programmverzeichnis:

Datei	Typ	Funktion
implab.dll	DLL	Datei mit der Treiberfunktionalität
implab.lib	LIB	Datei zur Einbindung in den Compiler
implab_exp.h	Header	Datei mit den Definitionen, Konstanten, den Funktionsnamen und Funktionsparametern
globaldef.h	Header	Datei mit Konstanten und TypDefinitionen
impdef.h	Header	Datei mit Konstanten und TypDefinitionen
Tcpip5000.dll	DLL	Datei mit den TCP/IP Funktionen des Treibers

Windows System Verzeichnis:

Datei	Typ	Funktion
implab.dll	DLL	Datei mit der Treiberfunktionalität
tcpip5000.dll	DLL	Datei mit den TCP/IP Funktionen des Treibers
Ftd2xx.dll	DLL	Datei mit den Treibern für den virtuellen USB COM Port von FTDI.

Die Beispielpprogramme und der Labview Beispielcode müssen von der CD kopiert werden.

Anwendung

LabView

Übersicht

Zur Einbindung des Treibers in LabView wird die Dll dynamisch über den DLL-Namen eingebunden.

Die Beispiel Vis wurden mit LabView 11 erstellt und getestet.

Beispiel "Demo_ImplabV5.Vi"

Öffnen Sie das VI in Labview

Für das Beispiel wird ein IMP 1C mit Adresse 1 verwendet. Auf Kanal 1-4 wird die Temperatur eines Thermoelementes Typ K gemessen.

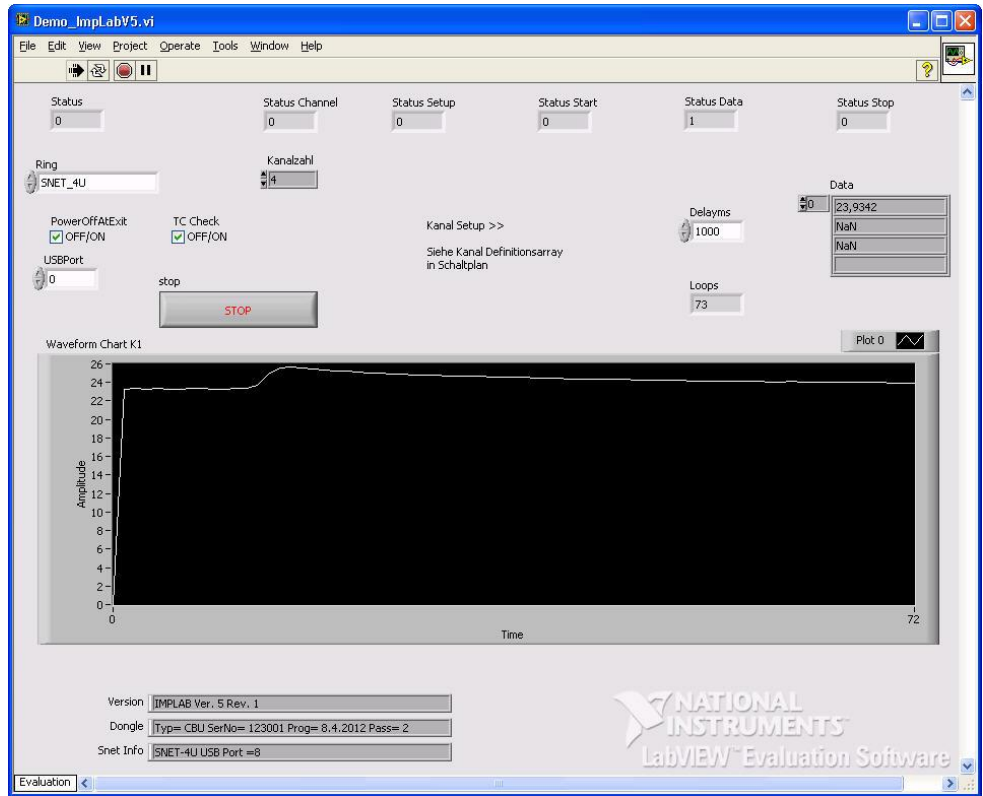
Die Kanäle sind im Abschnitt Messkanal definieren über ein konstantes Array eingestellt.

Die Messrate ist 1Hz.

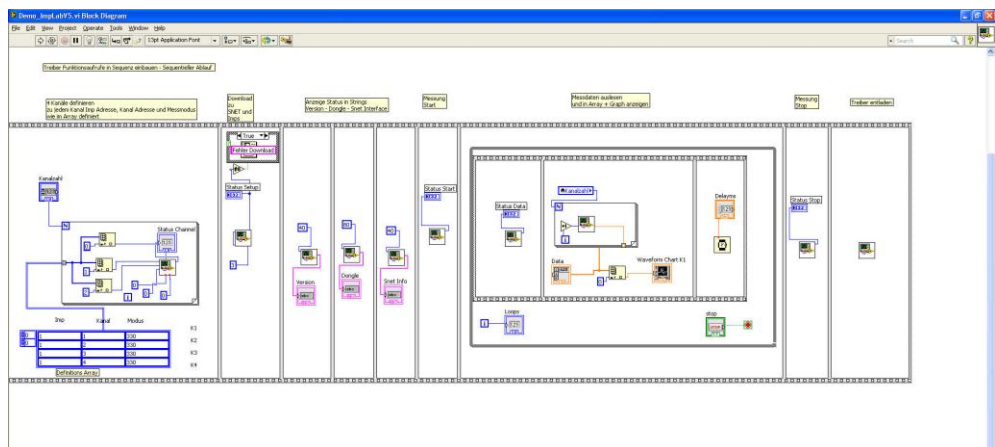
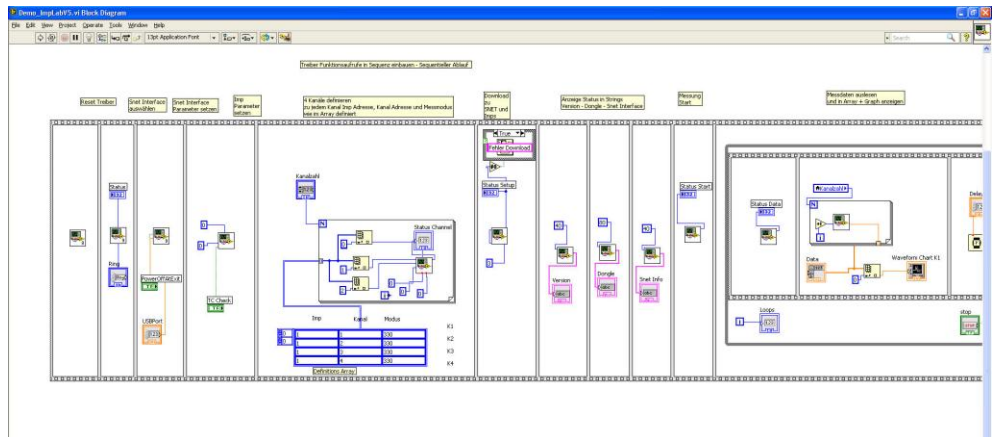
Die Sub-Vis müssen sich im selben Ordner wie das Haupt-VI befinden.

Bedienung des VI:

- Wählen Sie mit dem Ring-Control das Snet Interface.
- Starten Sie die Messung
- Die Status Anzeigen müssen den Wert 0 zeigen, die Messwertanzeige gibt auf 4 Kanälen die Temperatur in Celsius aus.
- Die Graphik zeigt Kanal 1 an.



Labview Frontpanel



Labview Funktionspanel

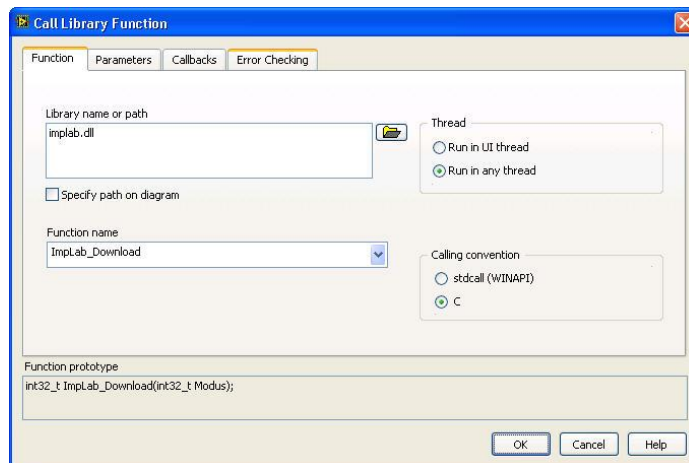
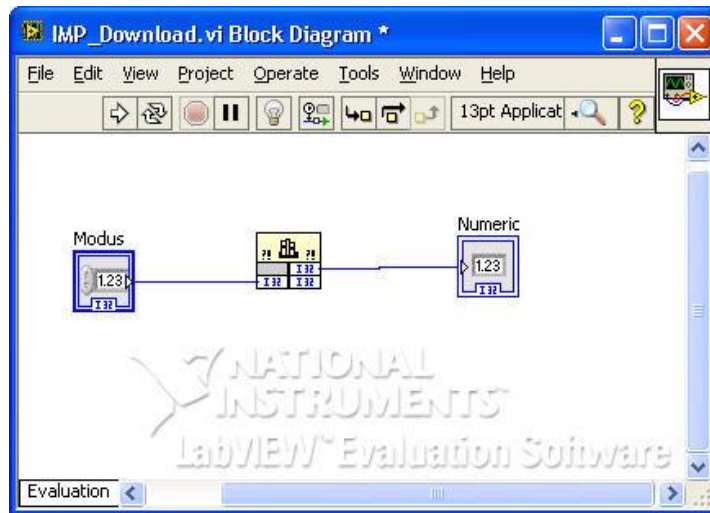
Aufbau des Labview Programms

Das Beispielprogramm Demo_ImplabV5.Vi ruft die einzelnen Treiber – Funktionen nacheinander auf. Dies wird durch die Sequenz Struktur erreicht.

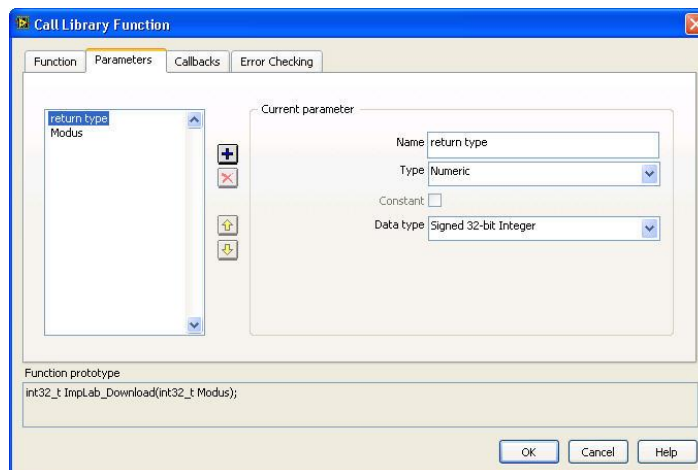
1. Reset Treiber
2. Auswahl des Interface
3. Parameter für Interface setzen
4. Default Parameter für IMP setzen
5. Kanäle hinzufügen. Dazu wird eine for-Schleife verwendet. Die Anzahl der Durchläufe wird über eine lokale Variable definiert, die auch beim Lesen der Daten verwendet wird.
Die Eigenschaften der Kanäle wie IMP Adresse und IMP Kanal werden über ein Konstantes Array an die Funktion zugewiesen.
6. Download der Konfiguration zu den IMPs. Hier wird geprüft, ob die Verbindung mit der Hardware möglich ist.
7. Abfrage der Informationen des Treibers: Version, Dongle-Status und SNET-Interface
8. Start der Messung
9. Abfrage und Anzeige der Daten in einer while-Schleife, solange bis die Stop-Taste gedrückt wird.
Innerhalb der for Schleife wird eine Sequenz verwendet, um die Reihenfolge des Funktions-Aufrufes sicherzustellen:
Zuerst werden aus allen aktiven IMPs die Daten ausgelesen und im Treiber zwischengespeichert. Dann werden die Daten in einer weiteren for-Schleife abgefragt und in ein Array gespeichert. Aus dem Array wird Index 0 in die Graphik gesendet.
Danach erfolgt ein Delay, abhängig von der Messrate, hier 1000ms.
10. Nach Stop in der while-Schleife wird die Messung gestoppt
11. Abmeldung des Treibers

Beispiel für die Einbindung einer DLL Funktion in Labview

Im Block Diagramm wird die Funktion „Call Library Function“ verwendet.



Definition des Funktions Namens



Definition der Parameter

Funktionsreihenfolge

Die Einbindung der Funktionen sollte nach der folgenden Reihenfolge ablaufen:

- Treiber-Reset
- Treiber Versions- und Registrierungsabfrage
- Setzen der Meßparameter
- Setzen der Interface Parameter
- Setzen der IMP Parameter
- Setzen der Kanal Parameter
- Download der Parameter an die Hardware
- Start der Messung
- Auslesen der Meßdaten
- Stop der Messung

Zuordnung

Die Zuordnung der IMP Module zu den Interfaces und die Zuordnung der IMP Kanäle zu den IMP Modulen erfolgt über die Rückgabe Codes beim Setzen der Interface bzw. Setzen der IMPs.

Somit können IMP Module und Kanäle auf einfache Weise zugeordnet werden.

C++

Zur Einbindung des Treibers in ein MFC bzw. C++ Programm wird die Dll über die LIB Datei und die Header Datei definiert.

Die Einbindung kann statisch mit Programmstart oder dynamisch zur Laufzeit erfolgen.

Alle Funktionsaufrufe wurden über die "extern "C" " Anweisung exportiert und führen das Makro AFX... aus.

Die Einbindung erfolgt in den Projekteinstellungen im Feld "Link / Input" durch einen zusätzlichen Eintrag "IMPLAB.LIB" unter "Object/Library Modules".

Die Datei "IMPLAB.DLL" kann sich entweder im EXE Verzeichnis oder im Windows-System32-Verzeichnis befinden.

Beispiel "CallImpLab.EXE"

Das Beispiel-Programm zeigt die Einbindung in ein MFC C++ Programm.

Dieses Programm wurde mit dem Projekt-Wizard als MFC-Appwizard (EXE) mit Namen CallImpLab erstellt, mit Auswahl Single Document und Baseclass = CformView.

Der Wizard erstellt u.a. die Datei "CallImpLabView.CPP" für die Ansichtsklasse "CCallImpLabView". Die Treiberaufrufe wurden in der Datei "ImpLabInp.CPP" zusammengefasst. Dort werden die Treiberfunktionen über die Header Datei "implab_exp.h" eingebunden.

Projekteinstellungen

General / MFC = Use MFC in a shared Dll

C++ Code Generation / Calling Convention = _cdecl

Link / Object-Library Modules = implab.lib

Init

In der Funktion OnInitialUpdate wird der Treiber initialisiert durch Aufruf der Funktion ImplInit(). Diese setzt den Treiber zurück, ruft die Versionsinfo ab und setzt die Fehlerausgabe (optional).

Das Dialogfenster enthält zwei Buttons, mit denen die Messung gestartet und gestoppt wird.

Start

Der Startbutton ist mit der Funktion "OnStart2()" verknüpft, die wiederum die Funktion "ImpDownloadAndStart()" aufruft und den 1 Sekunden Timer freigibt.

Setup

An diese Funktion wird die Snet Adresse übergeben. Dort wird der Treiber Reset aufgerufen, das SNET Interface gesetzt und im weiteren das IMP mit Adresse 1 und die Kanäle 1 bis 4 mit der Funktion "Thermoelement Typ K" gesetzt.

Download und Start

Im folgenden ruft diese Funktion nach fertiger Parameterübergabe die Treiberfunktion "ImpLab_Download()" auf und startet die Messung mit "ImpLab_StartMeasure()".

Abfrage und Anzeige der Meßdaten

In der Funktion "ImpReadout()" wird die Treiberfunktion "ImpLab_ReadOutImps()" aufgerufen, die prüft, ob neue Daten vorliegen. Falls ja, werden diese mit der Funktion ReadOutDataImp ausgelesen. Im Treiber Callback "OnTimer()" der Klasse "CcallImpLabView" werden diese Werte dann in die Member-Variablen "m_strData1" bis .."m_strData4" eingetragen und angezeigt.

Der Button Stop ist mit der Funktion "OnStart3()" verknüpft, die wiederum die Funktion "ImpStop()" aufruft und den Timer beendet.

Bei Stop wird die Treiberfunktion "ImpLab_StopMeasure()" aufgerufen. Der Solartron Treiber wird mit der Treiberfunktion "ImpLab_Disconnect()" entladen.

Beispielcode aus dem Programmpaket Callimplab, Datei implabinp.cpp

```
int ImpDownloadAndStart (int nBaseAdress)
{
    int n1, n2, n3;
    int nErr, nSize;
    int nChanIdx_1, nChanIdx_2, nChanIdx_3, nChanIdx_4;
    int nImpAdress, nIntegration=0, nTempUnit=0, nTcCheck=0, nUnitConversFlag=0;
    CString sgError="";
    char *pChar = NULL;

        nImpAdress = 1;

// IMP Treiber Reset
ImpLab_Reset ();

// Abtastrate setzen
ImpLab_SetMeasSrate (1000);

nErr = ImpLab_SetIfcTyp (cINTERFACE_SNET_USB); //cINTERFACE_SERIAL); //cINTERFACE_SNET);
//cINTERFACE_SNET_USB);
if (!nErr)
{
        // Set Auto Com Port
    ImpLab_SetIfcParam (nSnetIdx, 1, 0, 0);

// IMP an Adresse 1 anmelden mit: Integrationzeit FR1, Einheit Celsius, Fühlerbrucherkennung
nIntegration = IntegrationTimeFR0; //IntegrationTimeFR0;
nTempUnit = TempUnitGradCelsius;
nTcCheck = CheckTC_On; //CheckTC_Off; //CheckTC_On;

ImpLab_SetImpDefault (nIntegration, nTempUnit, nTcCheck);

        if (!nErr)
nErr = ImpLab_AddImpChan (nImpAdress, 1, ModeCode_Thermocouple_Typ_K, nUnitConversFlag, 0.0, 1.0);
        if (!nErr)
nErr = ImpLab_AddImpChan (nImpAdress, 2, ModeCode_Thermocouple_Typ_K, nUnitConversFlag, 0.0, 1.0);
        if (!nErr)
nErr = ImpLab_AddImpChan (nImpAdress, 3, ModeCode_Thermocouple_Typ_K, nUnitConversFlag, 0.0, 1.0);
        if (!nErr)
nErr = ImpLab_AddImpChan (nImpAdress, 4, ModeCode_Thermocouple_Typ_K, nUnitConversFlag, 0.0, 1.0);

        if (nErr)
        {
                // Beispiel für Auslesen der Fehlermeldung aus dem IMPLAB Treiber
                ShowError (nErr, "ImpLab_Download");
        }
}

// Setup Download an IMPs
if (!nErr) {
        nErr = ImpLab_Download (cDownloadModeAll);
        if (nErr)
        {
                // Beispiel für Auslesen der Fehlermeldung aus dem IMPLAB Treiber
                ShowError (nErr, "ImpLab_Download");
        }
}

// Messung starten
```

```

if (!nErr) {
    nErr = ImpLab_StartMeasure ();
    if (nErr)
        {
            ShowError (nErr, "ImpLab_StartMeasure");
        }
}

```

```

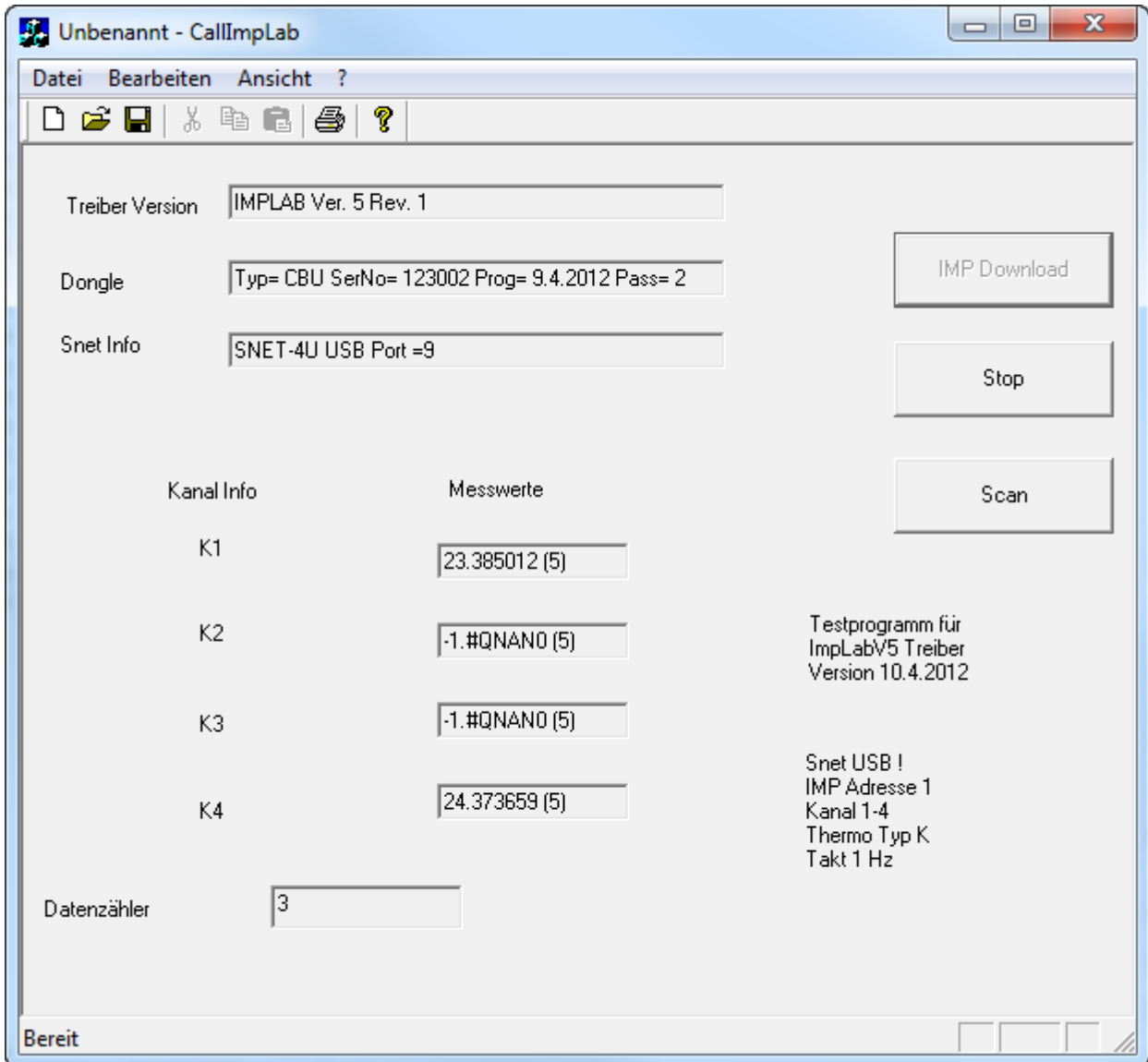
ImpLab_GetSnetInfo (szSnetInfoStr, sizeof szSnetInfoStr);

```

```

// Rückgabe, ob Messung läuft
return ImpLab_IsMeasureRunning ();

```



Übersicht

Alle Funktionen sind in der Datei implab_exp.h definiert. Zur Einbindung wird die Datei implab.lib benötigt.

Der Beispielcode in callimplab zeigt die Verwendung der Funktionen

ImpLab_GetVersion

Liest die Versionsnummer und den Versionsstring aus.

Aufruf	ImpLab_GetVersion (char* pszVersion, int nLen);
Return Typ	int
Return-Wert	Versionsnr

Funktionsparameter:

Typ	Wert
char-Zeiger - pszVersion	Versionsstring
Int - nLen	max. Anzahl an Zeichen

ImpLab_GetDongleInfo

Liest Informationen zum USB Dongle aus.

Angezeigt werden der Typ des Dongle, die Seriennummer, das Programmierdatum und der Prüfwert.

Der Prüfwert muss 2 sein, damit der Treiber arbeitet.

Aufruf	ImpLab_GetDongleInfo (char* pszInfo, int nLen);
Return Typ	int
Return-Wert	Status Dongle gefunden

Funktionsparameter:

Typ	Wert
char-Zeiger - pszInfo	Infostring
Int - nLen	max. Anzahl an Zeichen

ImpLab_GetSnetInfo

Liest Informationen zum SNET Interface.

Angezeigt werden der Typ des Interface und die Parameter.

Aufruf	ImpLab_GetSnetInfo (char* pszInfo, int nLen);
Return Typ	int
Return-Wert	0=ok

Funktionsparameter:

Typ	Wert
char-Zeiger - pszInfo	Infostring
Int - nLen	max. Anzahl an Zeichen

ImpLab_Reset

Initialisiert den Treiber:

Löscht alle Interface, IMP und Kanal Strukturen.

Setzt alle Variablen auf Ihre Default Werte

Entlädt die Solartron DLL und die TCPIP DLL, falls diese geladen wurden.

Aufruf	ImpLab_Reset (void);
Return-Typ	int
Return-Wert	0

Funktionsparameter:

Keine

ImpLab_SetMeasSrate

Setzt die Basis Abtastrate für alle IMPs in 1/1000 Sekunden.

Aufruf	ImpLab_SetMeasSrate (int nMilliseconds);
Return Typ	int
Return-Wert	Status: 0 = OK <> 0 = Fehler

Funktionsparameter:

Typ	Wert
Int - nMilliseconds	Basis Abtastrate der Messung in 1/1000 Sekunden Bereich: 100 ms = 10 Messungen / Sekunde bis 10000 ms = 1 Messungen / 10 Sekunden

ImpLab_SetIfcTyp

Meldet die SNET Karte als Interface an und setzt die Parameter

Aufruf	ImpLab_SetIfcTyp (int nInterface, int* pnInterfaceIdx);
Return Typ	Int
Return-Wert	Ok= 0, <>0 Fehler

Funktionsparameter:

Typ	Wert
Int - nInterface	Code für Interface Typ 0=Snet Isa/Pci 1=Snet Usb 2= RS485 3= Ethernet
Int* -pnInterfaceIdx	Zeiger für Interner Index für Interface nur für spezielle Verwendung, mit 0 vorbelegt

ImpLab_SetIfcParam

Übergabe der Parameter für die verschiedenen Interface Typen.

Aufruf	ImpLab_SetIfcParam (int nIfcIndex int nPowerOffAtExit, int nSnetAdress, int nUsbComPort, int nComPort, int nBaudrate, int nSerialFormat, int nTCPPort, const char* pszTCPAdress, const char* pszTCPSubnet);
Return Typ	Int
Return-Wert	ok= 0

Funktionsparameter:

Typ	Wert
Int - nComPort	PC-Port Nr. der seriellen Schnittstelle Bereich cComPortMin .. cComPortMax
Int - nSnetAdress	SNET ISA PC Adresse
Int - nUsbComPort	Virtueller Com Port des USB Interface 0=automatische Zuordnung des Com Port
Int - nBaudrate	Taktrate der Seriellen Übertragung. (muss mit IMP Konfiguration übereinstimmen!) Bereich siehe cBaudrate...

Typ	Wert
Int - nSerialFormat	Format der seriellen Übertragung (muss mit IMP Konfiguration übereinstimmen!) Bereich cSER_FORMAT_ASCII, cSER_FORMAT_RTU
Int - nTCPPort	Port Nr. der TCP/IP Verbindung. (muss mit IMP Konfiguration übereinstimmen!) Default: CnDefTCPPort
Char* - szTCPAdress	Adresse der TCP/IP Verbindung im Format "a.b.c.d". (muss mit IMP Konfiguration übereinstimmen!) Default: CszDefTCPAdress
Char* - szTCPSubnet	Netzmaske der TCP/IP Verbindung im Format "a.b.c.d". (muss mit IMP Konfiguration übereinstimmen!) Default: CszDefTCPSubnet
Int - nPowerOffAtExit	Flag für Stop der Messung bei Stop der Messung 0 – keine Abschaltung 1 - Abschaltung

ImpLab_SetImp

Anmelden des IMPs für die Messung und Setzen der IMP Parameter

Aufruf	ImpLab_SetImp (int nIfcIndex, int nImpAdress, int nIntegrationTime , int nTempUnitSelect, int nTCCheckOnOff);
Return Typ	Int
Return-Wert	IMP Index. Wird bei weiteren Zugriffen auf dieses IMP als Parameter erwartet. > 0 = OK <= 0 = Fehler

Funktionsparameter:

Typ	Wert
Int - nIfcIndex	Index auf das Interface, der bei ImpLab_SetIfc... zurückgegeben wurde
Int - nImpAdress	Adresse des IMPs im Bereich 1..50
Int - nIntegrationTime	Der Wert im Bereich 0..5 legt die interne Integrationszeit des ADC fest: 0 – 20,00 ms (50 Hz) 1 – 16,67 ms (60 Hz) 2 – 5,00 ms (400 Hz) 3 – 4,17 ms 4 – 1,25 ms

	5 – 1,04 ms
Int - nTempUnitSelect	Der Wert legt die Temperatureinheit bei allen Temperaturmessungen fest 0 – Einheit °C 1 - Einheit °F
Int - nTCCheckOnOff	Dieses Flag schaltet eine zusätzliche Testmessung zur Erkennung offener Thermokanäle ein. (Verlangsamt die Messrate) 0 – Aus 1 – Ein

ImpLab_SetImpDefault

Setzt Default Parameter für alle Imps

Aufruf	ImpLab_SetImpDefault (int nIntegrationTime , int nTempUnitSelect, int nTCCheckOnOff);
Return Typ	Int
Return-Wert	0=ok

Funktionsparameter:

Typ	Wert
Int - nIntegrationTime	Der Wert im Bereich 0..5 legt die interne Integrationszeit des ADC fest: 0 – 20,00 ms (50 Hz) 1 – 16,67 ms (60 Hz) 2 – 5,00 ms (400 Hz) 3 – 4,17 ms 4 – 1,25 ms 5 – 1,04 ms
Int - nTempUnitSelect	Der Wert legt die Temperatureinheit bei allen Temperaturmessungen fest 0 – Einheit °C 1 - Einheit °F
Int - nTCCheckOnOff	Dieses Flag schaltet eine zusätzliche Testmessung zur Erkennung offener Thermokanäle ein. (Verlangsamt die Messrate) 0 – Aus 1 – Ein

ImpLab_GetImp

Auslesen der IMP Setup Parameter

Aufruf	ImpLab_GetImp (int nIfcIndex, int nImpAdress, int *pnIntegrationTime , int *pnTempUnitSelect, int *pnTCCheckOnOff);
Return Typ	Int
Return-Wert	Status: 0 = OK <> 0 = Fehler

Funktionsparameter:

Typ	Wert
Int - nIfcIndex	Index auf das Interface, der bei ImpLab_SetIfc... zurückgegeben wurde.
Int - nImpIndex	Index auf IMP, der bei ImpLab_SetImp zurückgegeben wurde
Int* - pnIntegrationTime	Zeiger auf Integer. Der erhaltene Wert liefert die Integrationszeit des ADC
Int* - pnTempUnitSelect	Zeiger auf Integer. Der erhaltene Wert liefert die Temperatureinheit
Int* - pnTCCheckOnOff	Zeiger auf Integer. Der erhaltene Wert liefert das Flag zur Erkennung offener Thermokanäle

ImpLab_SetImpChan

Anmelden eines IMP-Kanals und setzen der Parameter.

Aufruf	ImpLab_SetImpChan (int nIfcIndex, int nImpIndex, int nImpChan, int nModeCode, int nUnitConversMode, float fUnitConvConst, float fUnitConvSlope);
Return Typ	Int
Return-Wert	IMP Kanal Index. Wird bei weiteren Zugriffen auf diesen Kanal als Parameter erwartet. > 0 = OK <= 0 = Fehler

Funktionsparameter:

Typ	Wert
Int - nIfcIndex	Index auf das Interface, der bei ImpLab_SetIfc... zurückgegeben wurde
Int - nImpIndex	Index auf IMP, der bei ImpLab_SetImp zurückgegeben wurde
Int - nImpChan	Terminal Nummer des IMP Kanals im

	Bereich 1..20
Int - nModeCode	Funktionscode für diesen Kanal. 0 – Kanal inaktiv >0 .. 1000 hex - siehe IMP Manual und Datei "IMPDEF.H"
Int – nUnitConversMode	Der Wert legt die Linearisierung des Meßwertes fest 0 – Meßwert normal 1 - Meßwert mit Linearisierung berechnen (siehe unten)
Float - fUnitConvConst	Linearisierungskonstante
Float - fUnitConvSlope	Linearisierungsfaktor Rückgabe - Meßwert = Linearisierungskonstante + Linearisierungsfaktor * Original-Meßwert

ImpLab_AddImpChan

Fügt einen neuen Kanal hinzu. In den Parametern wird das IMP und der Kanal definiert. .

Aufruf	ImpLab_SetImpChan (int nImpIndex, int nImpIndex, int nImpChan, int nModeCode, int nUnitConversMode, float fUnitConvConst, float fUnitConvSlope);
Return Typ	Int
Return-Wert	0=ok

Funktionsparameter:

Typ	Wert
Int - nImpAdress	Adresse des IMP im Bereich 1..50
Int - nImpChan	Terminal Nummer des IMP Kanals im Bereich 1..20
Int - nModeCode	Funktionscode für diesen Kanal. 0 – Kanal inaktiv >0 .. 1000 hex - siehe IMP Manual und Datei "IMPDEF.H"
Int – nUnitConversMode	Der Wert legt die Linearisierung des Meßwertes fest 0 – Meßwert normal 1 - Meßwert mit Linearisierung berechnen (siehe unten)
Float - fUnitConvConst	Linearisierungskonstante
Float - fUnitConvSlope	Linearisierungsfaktor Rückgabe - Meßwert = Linearisierungskonstante + Linearisierungsfaktor * Original-Meßwert

ImpLab_Download

Sendet alle Parameter an die Hardware:

Bei Snet ISA und PCI: Lädt die Solartron-DLL "IMPDRV.DLL"

Überprüft, ob die angemeldeten SNET Karten vorhanden sind

Überprüft, ob die angemeldeten IMPs am SNET-Bus vorhanden sind

Sendet die Setup Parameter an die IMPs.

Gibt die IMPs zur Messung frei

Aufruf	ImpLab_Download (int nMode);
Return Typ	Int
Return-Wert	Status 0 = OK <> 0 = Fehler

Funktionsparameter:

Typ	Wert
Int - nMode	Modus für Download. Wertebereich: CDownloadModeIfc = nur Interface anmelden cDownloadModeImp = nur IMPs anmelden (nur nach Aufruf von ImpLab_Download (CDownloadModeIfc)) cDownloadModeAll = Alles anmelden

ImpLab_StartMeasure

Startet die Messung bei allen IMPs

(Snet: Senden des Kommandos "CO;TR")

Aufruf	ImpLab_StartMeasure (void);
Return Typ	Int
Return-Wert	Status 0 = OK <> 0 = Fehler

Funktionsparameter:

Keine

ImpLab_StopMeasure

Stoppt die Messung bei allen IMPs.

(Snet: Senden des Kommandos "HA")

Aufruf	ImpLab_StopMeasure (void);
Return Typ	Int
Return-Wert	Status 0 = OK <> 0 = Fehler

Funktionsparameter:

Keine

ImpLab_ReadOutImps

Liest alle Messdaten aus den aktiven Imps und speichert diese Werte im Treiber zwischen.

Aufruf	ImpLab_ReadOutImps (int* pnLastTime)
Return Typ	Int
Return-Wert	Status der Daten 0 - keine Daten bereit 1 - Daten bereit, in DatenListe kopiert <0 - Error

Funktionsparameter:

Typ	Wert
Int * pnLastTime	Zeigt an, ob alle Imps ihre Daten bereitgestellt haben

ImpLab_ReadImpData

Auslesen der zuvor mit ImpLab_ReadOutImps() zwischengespeicherten Messdaten zu jedem einzelnen IMP.

Aufruf	ImpLab_ReadOutImps (nIdx, float *pfData, short *psValid);
Return Typ	Int
Return-Wert	0=ok

Funktionsparameter:

Typ	Wert
Int - nIdx	Index des auszulesenden Imps Bereich 1..50
float* - pfData	Zeiger auf Float Speicher mit dem Messwert
float* - pfValid	Zeiger auf Float Speicher mit dem Gültigkeitswert (Gibt die gültigen Dezimalstellen an)

ImpLab_Disconnect

Trennt alle Interface Verbindungen und setzt Power Off bzw. Messtakt aus.

Aufruf	ImpLab_Disconnect (void);
Return Typ	Int
Return-Wert	Status 0 = OK <> 0 = Fehler

Funktionsparameter:

Keine

ImpLab_IsMeasureRunning

Zeigt Status der Messung an

Aufruf	ImpLab_IsMeasureRunning (void);
Return Typ	Int
Return-Wert	Status der Messung 0 - Messung steht 1 - Messung läuft

Funktionsparameter:

Keine

ImpLab_GetErrorReport

Kopiert den vollständigen Fehlerreport.

Der Fehlerreport enthält alle Fehlerstrings hintereinander, mit Zeilenumbruch getrennt.

Aufruf	ImpLab_GetErrorReport (char *pszError, int nByteSize);
Return	Int
Return-Wert	Anzahl kopierte Bytes

Funktionsparameter:

Typ	Wert
Char* – pszError	Zeiger auf String
Int - nByteSize	Max. Anzahl Bytes zu kopieren

ImpLab_GetErrorReportSize

Liefert die Speichergröße zum Kopieren des Fehlerreports.

Aufruf	ImpLab_GetErrorStringSize (void);
Return	Int
Return-Wert	Anzahl Bytes

Funktionsparameter:

keine

ImpLab_ResetErrorReport

Setzt Fehlerreport zurück, d.h. der Report-String wird komplett gelöscht.

Aufruf	ImpLab_ResetErrorReport (void);
Return	--
Return-Wert	--

Funktionsparameter:

keine

ImpLab_GetErrorString

Liefert einen Fehlerstring zum Fehlerindex (+ oder - identisch)

Aufruf	ImpLab_GetErrorString (int nErrorIndex, char* pszError, int nStringSize);
Return	--
Return-Wert	--

Funktionsparameter:

Typ	Wert
int – nErrorIndex	Fehler Index, der von einer ImpLab_..Funktion zurückgeliefert wurde.
Char* – pszError	Zeiger auf String
Int - nStringSize	Max. Anzahl Bytes zu kopieren

ImpLab_SetDebugMode

Fehlerausgabe über Messagebox ein- oder ausschalten

Aufruf	ImpLab_SetDebugMode (int nMode);
Return Typ	--
Return-Wert	--

Funktionsparameter:

Typ	Wert
Int - nMode	Modus für Fehlermeldungen. Wertebereich: 0 – keine Ausgabe der Fehlermeldungen 1 - Ausgabe aller Fehlermeldungen in einer Messagebox mit Bestätigung.

Anhang

Kanal Codes für Messung

```
// IMP Command Mode Codes

// voltage measurement
#define ModeCode_VOLTDC_AUTO    0x100
#define ModeCode_VOLTDC_20mV    0x101
#define ModeCode_VOLTDC_200mV   0x102
#define ModeCode_VOLTDC_2V      0x103
#define ModeCode_VOLTDC_10V     0x104

// current measurement
#define ModeCode_CURRENTDC_AUTO  0x500
#define ModeCode_CURRENTDC_200uA 0x501
#define ModeCode_CURRENTDC_2mA   0x502
#define ModeCode_CURRENTDC_20mA  0x503
#define ModeCode_CURRENTDC_100mA 0x504

// resistance measurement
#define ModeCode_RES_4Wire_AUTO   0x200
#define ModeCode_RES_3Wire_AUTO   0x210
#define ModeCode_RES_2Wire_AUTO   0x220

// thermocouple measurement
#define ModeCode_Thermocouple_Typ_E 0x310
#define ModeCode_Thermocouple_Typ_J 0x320
#define ModeCode_Thermocouple_Typ_K 0x330
#define ModeCode_Thermocouple_Typ_R 0x340
#define ModeCode_Thermocouple_Typ_S 0x350
#define ModeCode_Thermocouple_Typ_T 0x360
```



```
#define ModeCode_Thermocouple_Typ_B 0x370
#define ModeCode_Thermocouple_Typ_N 0x380

// RTD 100 Ohm measurement
#define ModeCode_RTD100_4Wire_AUTO 0x400
#define ModeCode_RTD100_3Wire_AUTO 0x410

// RTD 10 Ohm measurement
#define ModeCode_RTD10_4Wire_AUTO 0x420
#define ModeCode_RTD10_3Wire_AUTO 0x430

// Digital Logik measurement
#define ModeCode_LOGIC_TTL 0x700
#define ModeCode_LOGIC_12V 0x701
#define ModeCode_LOGIC_2WIRE 0x702
```

Fehlercodes

Für die Treiberfunktionen, die als Rückgabewert den Status der Funktionsausführung liefern, gelten die Fehlercodes aus "ImpLab_Error.H"

- 102 "SNET kann nicht geöffnet werden";
- 104 "SNET Basisadresse unterhalb Gültigkeitsbereich";
- 105 "SNET Basisadresse oberhalb Gültigkeitsbereich";
- 106 "SNET Basisadresse nicht Vielfaches von 512";
- 107 "SNET Klasse konnte nicht angelegt werden";
- 108 "Kein freier Index für SNET Klasse";
- 109 "SNET Index ausserhalb Bereich";
- 110 "Nullzeiger auf SNET Klasse";
- 111 "SNET Parameter außerhalb Wertebereich";
- 201 "IMP kann nicht geöffnet werden";
- 202 "Keine freier Index für IMP-Klasse";
- 204 "Snet Connect nicht möglich";
- 205 "IMP Parameter ausserhalb Wertebereich";
- 206 "IMP nicht initialisiert";
- 207 "IMP Typ wird nicht unterstützt";
- 208 "Stream Adresse ausserhalb 0..3";
- 209 "Fehler in Funktion -ImpConnect-";
- 210 "IMP Adresse ausserhalb 1..50";
- 211 "Imp nicht am Bus gefunden";
- 212 "IMP Index ausserhalb Bereich";
- 213 "Nullzeiger auf IMP Klasse";
- 301 "Nullzeiger auf IMP-Kanal-Klasse";
- 302 "IMP Kanalparameter ausserhalb Wertebereich";
- 303 "IMP Kanal Adresse ausserhalb Bereich";
- 305 "IMP Kanal nicht initialisiert";
- 401 "Solartron DLL nicht geladen";
- 403 "Solartron DLL Funktionszeiger nicht gesetzt";
- 405 "Solartron DLL nicht entladen";
- 501 "Setup String Überlauf";
- 502 "Setup String Teilungsfehler";
- 601 "Aufruf Nullzeiger";
- 602 "Funktionsaufruf hier nicht erlaubt";
- 701 "String wurde nicht fehlerfrei gesendet";
- 702 "Indexüberlauf bei Zugriff auf IMP-Datentabelle";
- 703 "Timeout bei Teststream";
- 801 "Messung läuft";

- 802 "Messung läuft nicht";
- 888 "Passwort nicht gesetzt";
- 889 "Registrierung nicht gesetzt";

TCP/IP DII

Funktionen zur Kommunikation mit den IMP 5000 Modulen über die Ethernet Schnittstelle.

Verwendet die Windows WINSOCK DLL.

Wird vom IMP Universaltreiber dynamisch beim Download der Parameter bzw. Start der Messung eingebunden.

Solartron DLL

Datei "impdrv.dll" und weitere System- Dateien

Basisfunktionen zum Zugriff auf das SNET PC Interface.

Wird vom IMP Labview Treiber dynamisch beim Download der Parameter bzw. Start der Messung eingebunden.

Header Datei "IMPDEF.H"

Treiber Spezifische Typen und Konstanten.

Die Konstanten können als Werte an die Funktionsaufrufe übergeben werden.

Header Datei "GLOBALDEF.H"

Allgemeine Typen Definitionen

Glossar

LabView

Grafischer Compiler von National Instruments

DLL

Data Link Library zur Einbindung in Windows Programme

IMP

Isolated Measurement Pod, Präzisions Feldbus Modul

Index

3

3595 1–2, 4

5

5000 1–2, 4, 8, 21

E

Ethernet 1–2, 4, 8, 21

I

IMP 1–6, 8–11, 13–14, 19, 21

L

LabView 1, 20

R

RS485 2, 4

T

TCPIP 4–6

Treiber 1, 3–6, 19, 21