

Schüler-STM

Software – Dokumentation

geschrieben in Microsoft Visual Basic[®] 6.0
von Dipl.-Phys. Marcus Schäfer

Inhaltsverzeichnis

1	Hinweise zur Hardware	1
1.1	Computer	1
1.2	Messwerterfassungskarte	1
2	Dateien im Quell-Code-Paket	3
2.1	Klassendateien	3
2.2	Formulare	3
2.3	Module	3
2.4	Usercontrols	3
2.5	Projektdateien	3
2.6	Bilddateien	4
3	Klassenhierarchie	1
3.1	CADInfo	2
3.2	CEdit_STM_Image	3
3.3	CME300	5
3.4	CPC20TR	7
3.5	CRandom	9
3.6	CScale	11
3.7	CScan_STM_Image	12
3.8	CSTM_Data	14
3.9	CSTMTip	15
3.10	CStopUhr	17
3.11	CLineScan	18
3.12	CSTM_Image	19
4	Formulare	21
4.1	CSTM_Image.frm	21
4.2	CLineScan.frm	22
4.3	frm_coarseApproach.frm	23
4.4	frm_EditImages.frm	24
4.5	frm_ScanFrame.frm	25
4.6	frm_StartUp.frm	26
4.7	frmAbout.frm	27
4.8	Main.frm	28
5	Module	29
5.1	API_Interface	29
5.2	Konstanten	29
5.3	MathModul	29
6	Quellen	Fehler! Textmarke nicht definiert.

Windows und VisualBASIC sind Warenzeichen von Microsoft.

PC20TR ist ein Produkt der BMC-Messsysteme GmbH

ME300 ist ein Produkt der Meilhaus electronic GmbH

Weitere der im Text erwähnten Firmen- und Produktnamen sind eingetragene Warenzeichen der jeweiligen Firmen.

1 Hinweise zur Hardware

1.1 Computer

Welche Systemvoraussetzungen stellt diese Anwendung an den Computer? Die Antwort hierauf ist genauso kurz, wie auch pauschal. Da die Software in Microsoft Visual Basic geschrieben wurde, ist die Ausführungs geschwindigkeit nicht sonderlich hoch. Dieses muss/sollte daher durch einen verhältnismäßig schnellen Computer kompensiert werden. Ich schlage deshalb als untere Grenze eine Pentium II mit 300MHz vor.

Eine zweite, wesentlich Voraussetzung ist jedoch, dass ein ISA-Slot auf dem Mother-Board frei ist. Die hier verwendeten Messwerterfassungskarten (siehe nächstes Kapitel) sind nämlich, aus Kostengründen, noch ISA-Karten. Falls ein neuer Computer für das STM angeschafft werden soll, muss hierauf besonders geachtet werden, da die meisten angebotenen Computer nur noch PCI-Slots haben.

Die minimalen Systemvoraussetzungen in der Zusammenfassung:

Prozessor:	Pentium II mit 300MHz
Motherboard	1 freier ISA-Slot
Betriebssystem:	Microsoft Windows 95/98/2000/NT4.0
Hauptspeicher:	64MB
Festplatte:	10MB freier Speicher für Microsoft Visual Basic 6.0 ca. 100MB zusätzlich
Monitor	Bildschirmdiagonale: 15" Auflösung: 1024 × 768 Pixel ²

1.2 Messwerterfassungskarte

Diese Software ist für zwei problemlos erhältlichen Messwerterfassungskarten geschrieben und getestet worden:

- die ME300 der *Meilhaus Electronic GmbH*
- die PC20TR der *BMC-Messsysteme GmbH*

Beide Karten können direkt bei den herstellenden Firmen bezogen werden, die PC20TR ist zusätzlich auch noch z.B. bei der Firma *Conrad electronic* erhältlich.

Grundsätzlich sollten immer die aktuellsten Treiberdaten von den Homepages der Hersteller heruntergeladen und installiert werden. Dieses gilt auch, wenn man die Karte gerade erst erworben hat, da die beiliegenden Treiber-CDs meistens veraltet sind.

Aus Kostengründen ist momentan (Stand August 2001) die PC20TR zu empfehlen. Aus dem gleichen Grund wurden auch nur ISA-Karten verwendet. Vergleichbare PCI-Karten sind momentan noch ca. dreimal so teuer. Falls das Geld jedoch vorhanden ist, ist die Anschaffung auf lange Sicht betrachtet sinnvoller, da es in Zukunft immer schwieriger sein wird, Motherboards mit ISA-Slots zu kaufen.

Auf dem Markt sind auch Messwerterfassungssysteme erhältlich, die an den Universal Serial Bus (USB) angeschlossen werden. Rein prinzipiell sollten auch diese einsetzbar sein, jedoch ist darauf zu achten, dass sie eine ausreichende Digitalisierungsrate haben. Da ein Bild von (512×512) Pixeln in ca. 2 Minuten gemessen wird, muss die Samplefrequenz des Systems mindestens

$$\frac{(512 \cdot 512) \text{ Pixel}}{120 \text{ s}} = \frac{262.144 \text{ Pixel}}{120 \text{ s}} = 2184,5 \frac{\text{Pixel}}{\text{s}} \quad \text{also ca. } 2.2 \text{ kHz}$$

sein.

Abschließend auch hier eine Tabelle mit den wichtigsten Minimalvoraussetzungen für eine Meßwerterfassungskarte:

	Minimal	PC20TR	ME-300
AD-Eingänge	1	16	16
Messbereich	$\pm 3\text{V}$	$\pm 10\text{V}$	$\pm 10\text{V}$
Auflösung	12 Bit	12 Bit	12 Bit
DA-Ausgänge	2	2	4
Bereich	$\pm 3\text{V}^*$	$\pm 10\text{V}$	$\pm 10\text{V}$
Auflösung	12 Bit	12 Bit	12 Bit
Samplerate	5 kHz	100 kHz	200 kHz
Steckplatz	-----	ISA-16Bit	ISA-16Bit

Adressen der Hersteller/Händler:

Meilhaus Electronic GmbH
Fischerstraße 2
D-82178 Puchheim

Conrad Electronic GmbH
Klaus-Conrad-Strasse 1
D-92240 Hirschau

Dr. Schetter BMC IGmbH
Boschstr. 12
D-82178 Puchheim

<http://www.meilhaus.de>

<http://www.conrad.de>

<http://www.bmc.de>

* Damit der Rasterbereich eine Akzeptable Größe hat, sollten die 3V mittels eines Operationsverstärkers auf 10V verstärkt werden.

2 Dateien im Quell-Code-Paket

2.1 Klassendateien

CADInfo.cls	enthält Informationen zur AD-Wandlung der Karten
CME300.cls	Klasse zum Ansteuern der Meilhauskarte ME300
CEdit_STM_Image.cls	enthält die Funktionen zur Bildbearbeitung
CSTMTip.cls	simuliert quasi der Spitze des STMs
CRandom.cls	Klasse zur Simulation einer AD/DA-Karte
CScale.cls	enthält die Skaleninformationen des Bildes
CScan_STM_Image.cls	enthält die Funktionen zur Bildaufnahme
CSTM_Data.cls	enthält/verwaltet die eingelesenen Daten
CStopUhr.cls	implementiert eine Stoppuhr
CPC20TR.frm	Klasse zum Ansteuern der BMC-Karte PC20TR

2.2 Formulare

CSTM_Image.frm	Formular zur Darstellung des STM-Bildes
CLineScan.frm	Formular zur Darstellung eines Linienprofils
frm_coarseApproach.frm	Hilfe für die Grobannäherung
frm_EditImages.frm	Steuerleiste für die Bildbearbeitung
frm_ScanFrame.frm	Steuerleiste für die Bildaufnahme
frm_StartUp.frm	Splash-Screen
frmAbout.frm	Info-Dialog
Main.frm	Hauptprogramm

2.3 Module

Konstanten.bas	Definition globaler Konstanten
MathModule.bas	Definition einiger mathematischer Funktionen
API_Interface.bas	Definition des API-Funktionen

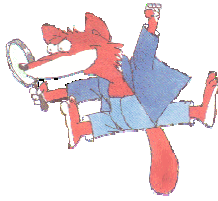
2.4 Usercontrols

Saveimagecontrol.???	Steuerelement zur Einstellung des Abspeicherverhaltens während der Bildaufnahme
ScanFrameOffsetControl.???	Steuerelement zur visuellen Einstellung des Offsets
ScanSizeControl.???	Steuerelement zur Einstellung der Rastergrösse
SngScrollControl.ctl	Wie das Steuerelement HscrollBar, allerdings für Single anstatt für Integer

2.5 Projektdateien

Schueler-STM.vbp
Schueler-STM.vbw

2.6 Bilddateien



fuchslogo.gif

Das Logo der Arbeitsgruppe von Prof. Fuchs



Quark+Co-Logo.gif

Das Logo der Fernsehsendung Quark & Co.



SIGEL.JPG

Das Sigel der Universität Münster



SCHLOSS.GIF

Das Wahrzeichen der Universität Münster

3 Klassenhierarchie

3.1 Ein paar einleitende Worte

Sicherlich wäre es schön, wenn jetzt hier eine kurze Abhandlung über objektorientiertes Programmieren, Klassen, Methoden usw. stehen würde. Aus mehreren Gründen soll darauf aber verzichtet werden, von denen ich nur zwei nennen möchte:

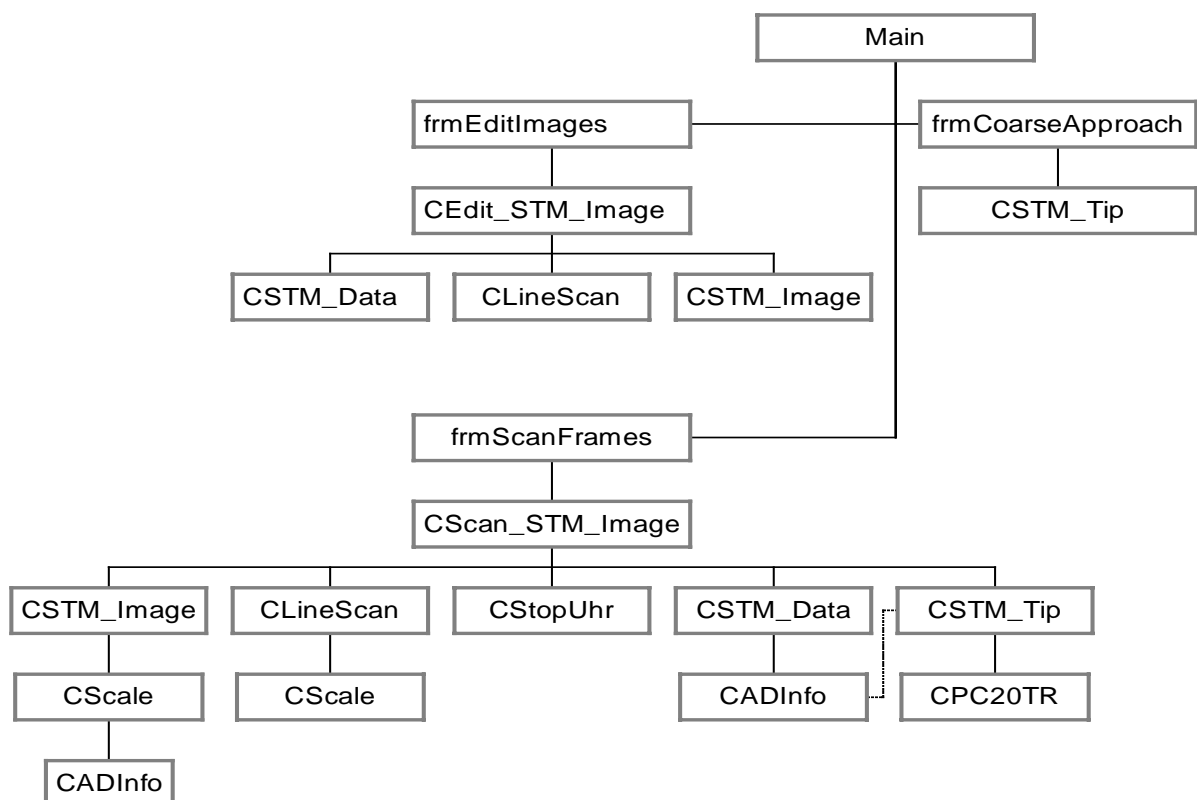
1. Dieses würde eindeutig den Rahmen dieser kurzen Dokumentation sprengen, da hierfür mehrere Seiten notwendig wären
2. Es gehört hier einfach nicht hin, sondern in ein Programmierhandbuch

Die meisten Begriffe sollten einem in VisualBasic (oder einer anderen objektorientierten Hochsprache) geübten Leser jedoch nicht völlig fremd sein. Im Zweifelsfall gibt die VisualBasic-Hilfe einen guten Einstieg.

3.2 Grafische Darstellung

Wie die einzelnen Objekte / Klassen voneinander abhängen lässt sich am besten in einem Diagramm darstellen.

Klassen- und Formularhierarchie



(Komplette Aufschlüsselung nur im Ast „frm_Scan_Frames“)

3.3 CADInfo

3.3.1 Events

keine

3.3.2 Properties (Eigenschaften)

MaxInputVoltage As Single

Die maximale Spannung, die die Karte momentan einlesen kann

MinInputVoltage As Single

Die minimale Spannung, die die Karte momentan einlesen kann

MaxSkalenteile As Long

in wie vielen Skalenteilen diese Spannungsdifferenz aufgelöst wird (bei einer 12Bit Karte sind dies in der Regel $4096 = 0xFFFF$)

OP_Gain As Single

Verstärkungsfaktor des Tunnelverstärkers in V pro nA

3.3.3 Public

Friend Sub GetInfoFromSTMTip(Optional ByVal STMTip As CSTM_Tip = Nothing)

Setzt alle Eigenschaften anhand der Informationen aus der Klasse CSTM_Tip

Friend Sub SetInfo(ByVal MaxInputVolt As Single, ByVal MinInputVolt As Single, ByVal MaxSkalenteil As Long, ByVal OPV_Gain As Single)

Setzt alle Eigenschaften auf einmal

Friend Function Skalenteile2NanoAmpere(ByVal ST As Long) As Single

Rechnet die übergeben Skalenteile in nA um

Friend Function Skalenteile2Label(ByVal ST As Long) As String

Erzeugt aus den übergebenen Skaltenteilen eine Beschriftung der Form „0.0 nA“

3.3.4 Private

Private varMaxSkalenteile As Long

speichert die zugehörige Eigenschaft

Private varMaxInputVoltage As Single

speichert die zugehörige Eigenschaft

Private varMinInputVoltage As Single

speichert die zugehörige Eigenschaft

Private varOPGain As Single

speichert die zugehörige Eigenschaft

Private Sub Class_Initialize()

Initialisiert die Klasse mit Standardwerten

3.4 CEdit_STM_Image

3.4.1 Events

DataManipulationStarted(ByVal typ As DataManipulationMethods)

wird ausgelöst, wenn eine Datenmanipulationsroutine aufgerufen wird. Die Variable typ enthält den Code der entsprechenden Routine.

DataManipulationProgress(ByVal typ As DataManipulationMethods, ByVal Percent As Single)

wird in regelmäßigen Abständen während der Datenmanipulation ausgelöst und ermöglicht somit die Steuerung einer ProgressBar. Typ gibt wiederum die Art der Routine und Percent die Fortschritt der Routine zurück

DataManipulationFinished(ByVal typ As DataManipulationMethods)

Wird nach Beendigung der Datenmanipulation aufgerufen.

ScaleHasChanged(ByVal AbsMin As Single, ByVal min As Single, ByVal max As Single, ByVal AbsMax As Single)

Wird ausgelöst, sobald sich die Skala des Bildes ändert. Min und max enthalten die begrenzenden Werte.

3.4.2 Properties (Eigenschaften)

ADInfo() As CADInfo

Enthält die AD-Informationen der Karte

CommonDialogControl() As CommonDialog

Enthält das CommonDialog-Steuerelement des übergeordneten Formulars

Resolution() As Long

Die aktuelle Größe des Bildes in Pixel

MaxResolution() As Long

Die gemäß der Daten maximale Größe des Bildes

STMFilesDirectory() As String

Das Verzeichnis mit den Datendateien

Visible() As Boolean

Sichtbarkeit des Bildes

SampleInfo() As String

Kommentar zur Probe

Filename() As String

Aktueller Dateiname

ImageScale() As Cscale

Aktuelle Bildskala

3.4.3 Public

Friend Sub ExecuteDataManipulation(typ As DataManipulationMethods, Optional ByVal PercentOfScale As Single = 100)

Startet die Datenmanipulation am aktuelle Bild

Typ: Code für die Arte der Manipulation

PercentOfScale: zu wieviel Prozent die Skala ausgenutzt werden soll

Friend Sub ResetScale()

Setzt die Skala auf Standardwerte zurück

Friend Sub OptimizeScale(Optional ByVal PercentOfScale As Single = 100)

Optimiert die Ausnutzung der Skala

PercentOfScale: zu wieviel Prozent die Skala ausgenutzt werden soll

Friend Sub SubtractLine(Optional ByVal PercentOfScale As Single = 100)

Zieht von jeder Zeile den Untergrund ab und passt danach die Skala an

PercentOfScale: zu wieviel Prozent die Skala ausgenutzt werden soll

Friend Sub FlattenData(Optional ByVal PercentOfScale As Single = 100)

Zieht vom gesamt Bild eine Untergrundebeine ab und passt danach die Skala an

PercentOfScale: zu wieviel Prozent die Skala ausgenutzt werden soll

Friend Sub SmoothData()
Glättet die Daten, in dem ein gewichteter Mittelwert für jeden Punkt berechnet wird.

Public Function LnData(Optional ByVal PercentOfScale As Single = 100)
Logarithmiert die Daten
PercentOfScale: zu wieviel Prozent die Skala ausgenutzt werden soll

Friend Sub Hide()
Versteckt das Bild

Friend Sub Show()
Zeigt das Bild

Friend Sub ShowLineScanZeile(ByVal Line As Long)
Zeigt einen Linienprofil für Zeile *Line*

Friend Sub ShowLineScanSpalte(ByVal Line As Long)
Zeigt einen Linienprofil für Spalte *Line*

Friend Sub HideLineScan()
Versteckt das Linienprofil

Friend Sub SaveBMPImage(Optional ByVal Fname As String = "")
Speichert das Bild als BMP-Datei
Fname: enthält den Dateiname. Wird kein Dateiname übergeben, wird gefragt

Friend Sub SaveSTMDData(Optional ByVal Fname As String = "")
Speichert das Bild als STM-Daten-Datei
Fname: enthält den Dateiname. Wird kein Dateiname übergeben, wird gefragt

Friend Sub LoadSTMDData(Optional ByVal Fname As String = "")
Lädt eine STM-Daten-Datei
Fname: enthält den Dateiname. Wird kein Dateiname übergeben, wird gefragt

Friend Sub ReloadData()
Lädt die aktuellen Daten erneut ein.

3.4.4 Private

Private Sub Class_Initialize()
Der Konstruktor der Klasse

Private Sub Class_Terminate()
Der Destruktor der Klasse

Private Function AskForFilename(ByVal Extension As String, ByVal Filter As String) As String
Fragt nach einem Dateiname. Wird gegebenfalls von *SaveBMPImage*, *SaveSTMDData*, *LoadSTMDData* aufgerufen

Private WithEvents STMDData As CSTM_Data
Enthält die Bilddaten

Private WithEvents STMImage As CSTM_Image
Zeigt das Bild an

Private LineScan As ClineScan
Zeigt das Linienprofil an

Private mvarFilename As String 'lokale Kopie
Name der aktuellen Datendatei

Private STMFilesDir As String
Speichert die zugehörige Eigenschaft

Private Cdlg As CommonDialog
Speichert die zugehörige Eigenschaft

3.5 CME300

3.5.1 Events

InitFailed()

Wird ausgelöst, falls die Initialisierung der Karte fehlschlägt

*AnalogInputRangeChanged(ByVal ChannelNr As ME300InChannelNumbers,
ByVal fromRange As ME300AnalogInputRanges,
ByVal toRange As ME300AnalogInputRanges)*

Wird ausgelöst, sobald der AD-Bereich eines Kanals geändert wird

ChannelNr: Nummer des betroffenen Kanals

FromRange: alte AD-Bereich

ToRange: neu AD-Bereich

*AnalogOutputRangeChanged(ByVal ChannelNr As ME300OutChannelNumbers,
ByVal fromRange As ME300AnalogOutputRanges,
ByVal toRange As ME300AnalogOutputRanges)*

Wird ausgelöst, sobald der DA-Bereich eines Kanals geändert wird

ChannelNr: Nummer des betroffenen Kanals

FromRange: alte DA-Bereich

ToRange: neu DA-Bereich

3.5.2 Properties (Eigenschaften)

BoardNumber() As ME300CardNumbers

Nummer des zu benutzenden ME300-Boards

MaxSkalenteile() As Long

Auflösung der AD/DA-Karte in Skalenteilen. Die ME300-Karte hat eine Auflösung von 12Bit, so dass dieser Wert immer 4095=0xFFFF ist

3.5.3 Public

Public Sub InitCard(Optional ByVal nr As ME300CardNumbers = ME300NoCard)

Initialisiert die ME300-Karte mit der Nummer *nr*

Public Sub SetInputChannelRange(ByVal ChannelNr As ME300InChannelNumbers, ByVal range As ME300AnalogInputRanges)

Setzt den AD-Wandelbereich von Kanal *ChannelNr*

Public Function GetInputChannelRange(ByVal ChannelNr As ME300InChannelNumbers) As ME300AnalogInputRanges

Gibt den AD-Wandelbereich von Kanal *ChannelNr* zurück

Public Function MaxInputVoltage(ByVal ch As ME300InChannelNumbers) As Single

Gibt die auf Kanal *ch* maximal einlesbare Spannung in V zurück

Public Function MinInputVoltage(ByVal ch As ME300InChannelNumbers) As Single

Gibt die auf Kanal *ch* minimal einlesbare Spannung in V zurück

Public Sub SetOutputChannelRange(ByVal ChannelNr As ME300OutChannelNumbers, ByVal range As ME300AnalogOutputRanges)

Setzt den DA-Wandelbereich von Kanal *ChannelNr*

Public Function GetOutputChannelRange(ByVal ChannelNr As ME300OutChannelNumbers) As ME300AnalogOutputRanges

Gibt den DA-Wandelbereich von Kanal *ChannelNr* zurück

Public Function MaxOutputVoltage(ByVal ch As ME300OutChannelNumbers) As Single

Gibt die auf Kanal *ch* maximal ausgebbare Spannung in V zurück

Public Function MinOutputVoltage(ByVal ch As ME300OutChannelNumbers) As Single

Gibt die auf Kanal *ch* minimal ausgebbare Spannung in V zurück

Public Sub AnalogOutput(ByVal channel As ME300OutChannelNumbers, ByVal signal As Single, Optional ByVal OutputRange As ME300AnalogOutputRanges = AO_NORange)
 Gibt eine Spannung auf einem Kanal der ME300 aus
channel: Nummer des Ausgabekanals
signal: auszugebende Spannung in V
OutputRange: DA-Bereich des Kanals (optional)

Public Sub DirectAnalogOutput(ByVal channel As ME300OutChannelNumbers, ByVal value As Long, Optional ByVal OutputRange As ME300AnalogOutputRanges = AO_NORange)
 Gibt eine Spannung auf einem Kanal der ME300 aus
channel: Nummer des Ausgabekanals
value: auszugebende Spannung in digitaler Form, d.h.: 0x000 – 0xFFFF
OutputRange: DA-Bereich des Kanals (optional)

Public Function AnalogInput(ByVal channel As ME300InChannelNumbers, Optional ByVal InputRange As ME300AnalogInputRanges = AI_NORange) As Single
 Liest eine Spannung auf einem Kanal der ME300 ein
channel: Nummer des Eingabekanals
InputRange: AD-Bereich des Kanals (optional)
Rückgabe: Spannungswert in Volt

Public Function DirectAnalogInput(ByVal channel As ME300InChannelNumbers, Optional ByVal InputRange As ME300AnalogInputRanges = AI_NORange) As Integer
 Liest eine Spannung auf einem Kanal der ME300 ein
channel: Nummer des Eingabekanals
InputRange: AD-Bereich des Kanals (optional)
Rückgabe: digitales Signal im Bereich 0x000 – 0xFFFF

Public Function ConvertInSignal(ByVal value As Integer, ByVal InputRange As ME300AnalogInputRanges) As Single
 Konvertiert ein digitales Signal in ein analoges
value: digitales Signal
InputRange: AD-Bereich, in dem das Signal gewandelt wurde
Rückgabe: Spannungswert in Volt

Public Sub FillComboBox(cb As ComboBox, IODirection As ME300Directions, Optional ByVal DynamicAllowed As Boolean = False)
 Füllt eine ComboBox mit Spannungsbereichen
Cb: die zufüllende ComboBox
IODirection: Bereich für die DA-Wandlung (Ausgabe) oder AD-Wandlung (Eingabe)
DynamicAllowed: Ob eine Dynamische Anpassung des Bereich auswählbar sein soll

Public Function Combo2RangeConst(cb As ComboBox, IODirection As ME300Directions)
 Wandelt den Auswahlwert einer ComboBox in die zugehörige Bereichskonstante um
Cb: die ComboBox
IODirection: Bereich für die DA-Wandlung (Ausgabe) oder AD-Wandlung (Eingabe)
Rückgabe: Bereichskonstante

3.5.4 Private

Private Sub Class_Initialize()
 Konstruktor der Klasse

Private Sub Class_Terminate()
 Destruktor der Klasse

Private Function ConvertOutSignal(ByVal signal As Single, ByVal OutRange As ME300AnalogOutputRanges) As Integer
 Konvertiert einen analogen Wert in einen digital
signal: analoger Spannungswert in V
OutRange: DA-Bereich, in dem das Signal gewandelt werden soll
Rückgabe: digitaler Wert

3.6 CPC20TR

3.6.1 Events

InitFailed()

Wird ausgelöst, falls die Initialisierung der Karte fehlschlägt

*AnalogInputRangeChanged(ByVal ChannelNr As PC20TR_InChannelNumbers,
ByVal fromRange As PC20TR_AnalogInputRanges,
ByVal toRange As PC20TR_AnalogInputRanges)*

Wird ausgelöst, sobald der AD-Bereich eines Kanals geändert wird

ChannelNr: Nummer des betroffenen Kanals

FromRange: alte AD-Bereich

ToRange: neu AD-Bereich

*AnalogOutputRangeChanged(ByVal ChannelNr As PC20TR_OutChannelNumbers,
ByVal fromRange As PC20TR_AnalogOutputRanges,
ByVal toRange As PC20TR_AnalogOutputRanges)*

Wird ausgelöst, sobald der DA-Bereich eines Kanals geändert wird

ChannelNr: Nummer des betroffenen Kanals

FromRange: alte DA-Bereich

ToRange: neu DA-Bereich

3.6.2 Properties (Eigenschaften)

BoardNumber() As PC20TR_CardNumbers

Nummer des zu benutzenden PC20TR_-Boards

MaxSkalenteile() As Long

Auflösung der AD/DA-Karte in Skalenteilen. Die PC20TR-Karte hat eine Auflösung von 12Bit, so dass dieser Wert immer 4095=0xFFFF ist

3.6.3 Public

Public Sub InitCard(Optional ByVal nr As PC20TR_CardNumbers = PC20TR_NoCard)

Initialisiert die PC20TR_-Karte mit der Nummer *nr*

Public Sub SetInputChannelRange(ByVal ChannelNr As PC20TR_InChannelNumbers, ByVal range As PC20TR_AnalogInputRanges)

Setzt den AD-Wandelbereich von Kanal *ChannelNr*

Public Function GetInputChannelRange(ByVal ChannelNr As PC20TR_InChannelNumbers) As PC20TR_AnalogInputRanges

Gibt den AD-Wandelbereich von Kanal *ChannelNr* zurück

Public Function MaxInputVoltage(ByVal ch As PC20TR_InChannelNumbers) As Single

Gibt die auf Kanal *ch* maximal einlesbare Spannung in V zurück

Public Function MinInputVoltage(ByVal ch As PC20TR_InChannelNumbers) As Single

Gibt die auf Kanal *ch* minimal einlesbare Spannung in V zurück

Public Sub SetOutputChannelRange(ByVal ChannelNr As PC20TR_OutChannelNumbers, ByVal range As PC20TR_AnalogOutputRanges)

Setzt den DA-Wandelbereich von Kanal *ChannelNr*

Public Function GetOutputChannelRange(ByVal ChannelNr As PC20TR_OutChannelNumbers) As PC20TR_AnalogOutputRanges

Gibt den DA-Wandelbereich von Kanal *ChannelNr* zurück

Public Function MaxOutputVoltage(ByVal ch As PC20TR_OutChannelNumbers) As Single

Gibt die auf Kanal *ch* maximal ausgebbare Spannung in V zurück

Public Function MinOutputVoltage(ByVal ch As PC20TR_OutChannelNumbers) As Single

Gibt die auf Kanal *ch* minimal ausgebbare Spannung in V zurück

Public Sub AnalogOutput(ByVal channel As PC20TR_OutChannelNumbers, ByVal signal As Single, Optional ByVal OutputRange As PC20TR_AnalogOutputRanges = AO_NORange)
 Gibt eine Spannung auf einem Kanal der PC20TR_ aus
channel: Nummer des Ausgabekanals
signal: auszugebende Spannung in V
OutputRange: DA-Bereich des Kanals (optional)

Public Sub DirectAnalogOutput(ByVal channel As PC20TR_OutChannelNumbers, ByVal value As Long, Optional ByVal OutputRange As PC20TR_AnalogOutputRanges = AO_NORange)
 Gibt eine Spannung auf einem Kanal der PC20TR_ aus
channel: Nummer des Ausgabekanals
value: auszugebende Spannung in digitaler Form, d.h.: 0x000 – 0xFFFF
OutputRange: DA-Bereich des Kanals (optional)

Public Function AnalogInput(ByVal channel As PC20TR_InChannelNumbers, Optional ByVal InputRange As PC20TR_AnalogInputRanges = AI_NORange) As Single
 Liest eine Spannung auf einem Kanal der PC20TR_ ein
channel: Nummer des Eingabekanals
InputRange: AD-Bereich des Kanals (optional)
Rückgabe: Spannungswert in Volt

Public Function DirectAnalogInput(ByVal channel As PC20TR_InChannelNumbers, Optional ByVal InputRange As PC20TR_AnalogInputRanges = AI_NORange) As Integer
 Liest eine Spannung auf einem Kanal der PC20TR_ ein
channel: Nummer des Eingabekanals
InputRange: AD-Bereich des Kanals (optional)
Rückgabe: digitales Signal im Bereich 0x000 – 0xFFFF

Public Function ConvertInSignal(ByVal value As Integer, ByVal InputRange As PC20TR_AnalogInputRanges) As Single
 Konvertiert ein digitales Signal in ein analoges
value: digitales Signal
InputRange: AD-Bereich, in dem das Signal gewandelt wurde
Rückgabe: Spannungswert in Volt

Public Sub FillComboBox(cb As ComboBox, IODirection As PC20TR_Directions, Optional ByVal DynamicAllowed As Boolean = False)
 Füllt eine ComboBox mit Spannungsbereichen
Cb: die zufüllende ComboBox
IODirection: Bereich für die DA-Wandlung (Ausgabe) oder AD-Wandlung (Eingabe)
DynamicAllowed: Ob eine Dynamische Anpassung des Bereich auswählbar sein soll

Public Function Combo2RangeConst(cb As ComboBox, IODirection As PC20TR_Directions)
 Wandelt den Auswahlwert einer ComboBox in die zugehörige Bereichskonstante um
Cb: die ComboBox
IODirection: Bereich für die DA-Wandlung (Ausgabe) oder AD-Wandlung (Eingabe)
Rückgabe: Bereichskonstante

3.6.4 Private

Private Sub Class_Initialize()
 Konstruktor der Klasse

Private Sub Class_Terminate()
 Destruktor der Klasse

Private Function ConvertOutSignal(ByVal signal As Single, ByVal OutRange As PC20TR_AnalogOutputRanges) As Integer
 Konvertiert einen analogen Wert in einen digital
signal: analoger Spannungswert in V
OutRange: DA-Bereich, in dem das Signal gewandelt werden soll
Rückgabe: digitaler Wert

3.7 CRandom

Diese Klasse ist analog zur Klasse CME300 aufgebaut und soll eine ME300-Karte simulieren.

CRandom war zu Beginn des Projektes erforderlich, um mittels simulierter Daten die Funktionalität der Programmelemente zu testen. Im weiteren Entwicklungsverlauf wurden jedoch zur Klasse CME300 Methoden hinzugefügt, die jedoch nicht in CRandom implementiert wurden. Soll diese Klasse wieder eingesetzt werden, müssen die Funktionen unter Umständen nachträglich übertragen werden.



3.7.1 Events

*AnalogInputRangeChanged(ByVal ChannelNr As ME300InChannelNumbers,
ByVal fromRange As ME300AnalogInputRanges,
ByVal toRange As ME300AnalogInputRanges)*

Wird ausgelöst, sobald der AD-Bereich eines Kanals geändert wird

ChannelNr: Nummer des betroffenen Kanals

FromRange: alte AD-Bereich

ToRange: neu AD-Bereich

*AnalogOutputRangeChanged(ByVal ChannelNr As ME300OutChannelNumbers,
ByVal fromRange As ME300AnalogOutputRanges,
ByVal toRange As ME300AnalogOutputRanges)*

Wird ausgelöst, sobald der DA-Bereich eines Kanals geändert wird

ChannelNr: Nummer des betroffenen Kanals

FromRange: alte DA-Bereich

ToRange: neu DA-Bereich

3.7.2 Properties (Eigenschaften)

MaxSkalenteile() As Long

Auflösung der „AD/DA-Karte“ in Skalenteilen. Dieser Wert ist analog zur ME300 immer 4095=0xFFFF.

3.7.3 Public

Public Sub InitCard(ByVal c as long)

Dummy-Funktion für die Kompatibilität zur Klasse CME300

Public Sub SetInputChannelRange (ByVal ch As Long, ByVal range As Long)

Setzt der Wert der lokalen Variable *miv* entsprechend der Angabe in *range*

Public Function MaxInputVoltage(ByVal ch As Long) As Single

Gibt die auf Kanal *ch* maximal einlesbare Spannung in V zurück (= +*miv*)

Public Function MinInputVoltage(ByVal ch As Long) As Single

Gibt die auf Kanal *ch* minimal einlesbare Spannung in V zurück (= -*miv*)

Public Sub SetOutputChannelRange (ByVal c As Long, ByVal r As Long)

Dummy-Funktion für die Kompatibilität zur Klasse CME300

Public Function MaxOutputVoltage(ByVal ch As Long) As Single

Gibt die auf Kanal *ch* maximal ausgebbare Spannung in V zurück (= +*miv*)

Public Function MinOutputVoltage(ByVal ch As Long) As Single

Gibt die auf Kanal *ch* minimal ausgebbare Spannung in V zurück (= -*miv*)

Public Sub AnalogOutput(ByVal ch As Long, ByVal signal As Single)

Schreibt in die lokale Variable *x* den Wert *Signal*

Public Sub DirectAnalogOutput (ByVal ch As Long, ByVal value As Long)

Dummy-Funktion für die Kompatibilität zur Klasse CME300

Public Function AnalogInput(ByVal channel As ME300InChannelNumbers, Optional ByVal InputRange As ME300AnalogInputRanges = AI_NORange) As Single
 Berechnet einen Spannungswert aus beliebigen Parametern. Diese Funktion simuliert die Datenerfassung der ME300
channel: Nummer des Eingabekanals
InputRange: AD-Bereich des Kanals (optional)
Rückgabe: Spannungswert in Volt

Public Function DirectAnalogInput(ByVal channel As ME300InChannelNumbers, Optional ByVal InputRange As ME300AnalogInputRanges = AI_NORange) As Integer
 Berechnet einen Spannungswert aus beliebigen Parametern. Diese Funktion simuliert die Datenerfassung der ME300
channel: Nummer des Eingabekanals
InputRange: AD-Bereich des Kanals (optional)
Rückgabe: Spannungswert in Volt

Public Function ConvertInSignal(ByVal value As Integer, ByVal InputRange As ME300AnalogInputRanges) As Single
 Konvertiert ein digitales Signal in ein analoges
value: digitales Signal
InputRange: AD-Bereich, in dem das Signal gewandelt wurde
Rückgabe: Spannungswert in Volt

Public Sub FillComboBox(cb As ComboBox, IODirection As ME300Directions, Optional ByVal DynamicAllowed As Boolean = False)
 Füllt eine ComboBox mit Spannungsbereichen
Cb: die zufüllende ComboBox
IODirection: Bereich für die DA-Wandlung (Ausgabe) oder AD-Wandlung (Eingabe)
DynamicAllowed: Ob eine Dynamische Anpassung des Bereich auswählbar sein soll

Public Function Combo2RangeConst(cb As ComboBox, IODirection As ME300Directions)
 Wandelt den Auswahlwert einer ComboBox in die zugehörige Bereichskonstante um
Cb: die ComboBox
IODirection: Bereich für die DA-Wandlung (Ausgabe) oder AD-Wandlung (Eingabe)
Rückgabe: Bereichskonstante

3.7.4 Private

Private miv As Single
 Enthält die maximale absolute Spannung Eingangsspannung

Private Sub Class_Initialize()
 Konstruktor der Klasse

3.8 CScale

3.8.1 Events

ScaleHasChanged(ByVal AbsMin As Single, ByVal min As Single, ByVal max As Single, ByVal AbsMax As Single)

Wird ausgelöst, wenn sich die Bildskala geändert hat. Die Variablen enthalten die neuen Werte der Skala

ColormapChanged()

Wird ausgelöst, wenn sich die Farbzuzuordnung der Skala geändert hat.

3.8.2 Properties (Eigenschaften)

ADInfo()

enthält die aktuellen AD-Informationen

AbsoluteMaximumValue() As Single

Enthält das absolute Maximum der Skala in nA

AbsoluteMinimumValue() As Single

Enthält das absolute Minimum der Skala in nA

ScaleMinLabel() As String

Gibt die Beschriftung für das Skalenminimum zurück

ScaleMinValue() As Single

Enthält den aktuellen Maximalwert der Skala in nA

ScaleMaxLabel() As String

Gibt die Beschriftung für das Skalenmaximum zurück

ScaleMaxValue() As Single

Enthält den aktuellen Minimalwert der Skala in nA

3.8.3 Public

Friend Sub DrawScale(ByVal left As Long, ByVal top As Long, ByVal width As Long, ByVal window As Form)

Zeichnet eine Skala an die Position (*left*, *top*) des Fensters *window*

Friend Sub ResetScale()

Setzt die Skala auf Standardwerte zurück

Friend Sub SetAbsScaleValues(ByVal AbsMinValue As Single, ByVal AbsMaxValue As Single)

Setzt die absoluten Maximalwert der Skala

Friend Sub SetAllScaleValues(ByVal AbsMin As Single, ByVal min As Single, ByVal max As Single, ByVal AbsMax As Single)

Setzt alle Maximalwerte (absolute und aktuelle) der Skala

Friend Sub SetScaleValues(ByVal MinValue As Single, ByVal MaxValue As Single)

Setzt die aktuellen Maximalwert der Skala

Friend Sub SetDefaultColormap()

Setzt die Standard-Farbpalette

Friend Function GetColor(ByVal vdata As Single) As Long

Gibt die Farbe aus der Farbpalette für den Wert *vdata* zurück

3.8.4 Private

Private Sub Class_Initialize()

Konstruktor der Klasse

Private Sub Class_Terminate()

Destruktor der Klasse

Private Sub CalcScaleFactor()

3.9 CScan_STM_Image

3.9.1 Events

ScanSizeHasChanged()

Wird ausgelöst, wenn sich die Größe des Rasterfensters geändert hat

ResolutionHasChanged(ByVal oldRes As Long, ByVal NewRes As Long)

Wird ausgelöst, wenn sich die Auflösung des Rasterfensters geändert hat

OldRes: vorherige Pixelzahl des Bildes

NewRes: neue Pixelzahl des Bildes

DataSaved()

Wird ausgelöst, wenn das Bild gespeichert wurde

NewScanVelocity(ByVal t As Double)

Wird ausgelöst, wenn eine neue Rastergeschwindigkeit gemessen wurde

t: Geschwindigkeit in Sekunden pro Zeile

3.9.2 Properties (Eigenschaften)

ScanAngle() As Single

Der Winkel in deg, der der Rasterrahmen gedreht ist

LogarithmSignal() As Boolean

Soll das Signal direkt Logarithmiert werden?

LinesPerSecond() As Double

aktuelle Rastergeschwindigkeit in Zeilen pro Sekunde

DelayAfterPixel() As Long

Pause in ms nach jedem Pixel

Get Tip() As CSTM_Tip

Gibt die Daten der Spitze als Klasse zurück

DatafilePath() As String

Verzeichnis der Datendateien

FitScaleToFirstLine() As Boolean

Soll die Skala an die Dynamik der ersten Bildzeile angepasst werden?

ScanDirection() As ScanDirectionConstants

Richtung der Bildaufnahme: Links->Rechts; Links<-Rechts oder Links<->Rechts

Resolution() As Long

Auflösung des Bildes in Pixel

SampleInfo() As String

Kommentar zur Probe

XOffset() As Long

X-Offset des Rasterrahmens in nm

YOffset() As Long

Y-Offset des Rasterrahmens in nm

ScanSize() As Long

Größe des Rasterbereichs in nm

ImageVisible() As Boolean

Bild sichtbar?

LineScanVisible() As Boolean

Linienprofile sichtbar?

SaveData() As SaveDataConstants

Modus der Datenspeicherung: nie, dieses oder nächstes Bild

3.9.3 Public

Friend Sub ShowImage()

Zeigt das STM-Bild

Friend Sub HideImage()
Versteckt das STM-Bild

Friend Sub ShowLineScan()
Zeigt das Linienprofil

Friend Sub HideLineScan()
Versteckt das Linienprofil

Friend Sub HideAll()
Versteckt Linienprofil und STM-Bild

Friend Sub SaveSTMData()
Speichert die aktuellen Daten

Friend Sub StopScan()
Beendet das Rastern

Friend Sub StartScan(Optional ByVal Direction As ScanDirectionConstants = None)
Startet das Rastern

3.9.4 Private

Private Sub Class_Initialize()
Konstruktor der Klasse

Private Sub Class_Terminate()
Destruktor der Klasse

Private Sub refresh_ADInfo()
Aktualisiert die AD-Informationen gemäß der Angaben in STMTip

Private Sub FitScaleToLine(ByVal y As Long, Optional ByVal PercentOfScale As Single = 100)
Paßt die Skala an die Zeile y an. *PercentOfScale* gibt dabei an, wieviel Prozent der Skala ausgenutzt werden sollen

Private Sub ProcessCurrentSignal(Optional signal As Single = -234)
Ließt den aktuellen Tunnelstrom ein und verarbeitet den Wert

Private Sub Scan_LR()
Führt eine Rasterbewegung inklusive Datenerfassung vom linken zum rechten Rand durch

Private Sub Scan_RL()
Führt eine Rasterbewegung inklusive Datenerfassung vom rechten zum linken Rand durch

3.10 CSTM_Data

3.10.1 Events

ResolutionHasChanged(ByVal oldRes As Long, ByVal NewRes As Long)

Wird ausgelöst, wenn sich die Auflösung des Rasterfensters geändert hat

OldRes: vorherige Pixelzahl des Bildes

NewRes: neue Pixelzahl des Bildes

3.10.2 Properties (Eigenschaften)

STMDataFileExtension() As String

Erweiterung des STM-Datenfiles

ScanSize() As Long

Größe des Rasterbereichs in nm

Resolution() As Long

Auflösung des Bildes in Pixel

SampleInfo() As String

Kommentar zur Probe

ADInfo() As CADInfo

AD-Informationen

3.10.3 Public

Public Sub CopyInto(ByRef newData As CSTM_Data)

Kopiert die Daten des Klasseninstanz in die Instanz *newData*

(also sozusagen der Kopiekonstruktor)

*Friend Function Save(ByVal path As String, Optional ByVal PathContainsFilename As Boolean = False)
As STMDataFileResult*

Speichert die aktuellen Daten als STM-Datenfile

Path: Pfad der Datei

PathContainsFilename: gibt an, ob der Pfad einen Dateinamen enthält (optional)

Friend Function Load(ByVal Filename As String) As STMDataFileResult

Lädt die angegebene Daten-Datei

Friend Sub SetDataPoint(ByVal x As Long, ByVal y As Long, ByVal signal As Single)

Setzt den Wert eines Punktes im Rasterrahmen

(*x,y*): Koordinaten des Punktes

signal: Tunnelstrom in nA

Friend Function getDataPoint(ByVal x As Long, ByVal y As Long) As Single

Gibt des Tunnelstrom in nA an Punkt (*x,y*) zurück

3.10.4 Private

Private Sub Class_Initialize()

Konstruktor der Klasse

Private Sub Class_Terminate()

Destruktor der Klasse

Die folgenden 5 Funktionen dienen nur zur Datei-Ein-Ausgabe und werden daher nicht weiter erklärt.

Private Sub GotoEndOfFileHeader()

Private Function PutText(s As String)

Private Sub GetString(s As String)

Private Function ExtractInt(s As String) As Long

Private Function ExtractSng(s As String) As Single

3.11 CSTMTip

3.11.1 Events

FrameBorderReached(position As CSTFBR)

Wird ausgelöst, wenn der Rand des Rasterrahmens erreicht wurde. *Position* gibt dabei an, ob der obere oder untere Rand erreicht wurde.

TipPropertiesChanged()

Wird ausgelöst, wenn sich eine Eigenschaft der Klasse CSTMTip geändert hat

3.11.2 Properties (Eigenschaften)

ScanAngle() As Single

Der Winkel in deg, unter dem gerastert wird

Logarithm() As Boolean

Sollen die Daten direkt logarithmiert werden?

MaxInputVoltage() As Single

Maximale Eingangsspannung (hängt von der IO-Karte ab)

MinInputVoltage() As Single

Minimale Eingangsspannung (hängt von der IO-Karte ab)

MinOutputVoltage() As Single

Minimale Ausgangsspannung (hängt von der IO-Karte ab)

MaxOutputVoltage() As Single

Maximale Ausgangsspannung (hängt von der IO-Karte ab)

MaxSkalenteile() As Long

Maximale Anzahl von Skalenteilen (hängt von der IO-Karte ab)

XOffset() As Long

X-Offset des Rasterrahmens in nm

YOffset() As Long

Y-Offset des Rasterrahmens in nm

ScanSize() As Long

Größe des Rasterrahmens in nm

Resolution() As Long

Auflösung des Bildes in Pixel

y() As Long

aktuelle Y-Position des Spitze in Pixel

x() As Long

aktuelle X-Position des Spitze in Pixel

wasFirstLineInFrame() As Boolean

war die zuletzt gerasterte Zeile die erste im Bild?

IOKarte() As _____

Gibt eine Instanz der IO-Karten-Klasse zurück

3.11.3 Public

SetInputVoltageRange(ByVal v As Single)

Setzt dem Eingangsspannungsbereich so, dass eine Spannung von *v* Volt optimal digitalisiert wird.

Friend Sub SetXYOffset(ByVal deltaX As Long, ByVal deltaY As Long)

Setzt den Offset des Rasterrahmens (Werte in nm)

Friend Sub setXY(ByVal Xvalue As Long, ByVal Yvalue As Long)

Setzt die aktuelle Spitzenposition (Werte in Pixel)

Friend Sub NextLine()

Bewegt die Spitze zur nächsten Zeile

Friend Sub MoveHome()

Bewegt die Spitze zur (1,1)-Position

Friend Sub MoveToFirstCol()

Bewegt die Spitze zur ersten Spalte

Friend Sub MoveToLastCol()

Bewegt die Spitze zur letzten Spalte

Friend Function getSignal() As Single

Liest den Tunnelstrom ein und logarithmiert gegebenenfalls

Rückgabe: Spannungswert des Tunnelverstärkers

3.11.4 Private

Private Sub SetTip()

Setzt die Spitze gemäß aller Parameter (x, y, xoffset, yoffset, angle)

Private Function getI() As Single

Liest den Tunnelstrom ein

Rückgabe: Spannungswert des Tunnelverstärkers

3.12 CStopUhr

3.12.1 Events

keine

3.12.2 Properties (Eigenschaften)

TimeElapsed() As Double

Gibt die gestoppte Zeit in Sekunden zurück

3.12.3 Public

Friend Sub Start()

Startet den Timer

Friend Sub Stopp()

Stoppt den Timer

3.12.4 Private

Private Function SystemTime2Seconds(ST As SYSTEMTIME) As Double

Rechnet die Systemzeit in Sekunden um

3.13 CLineScan

3.13.1 Events

3.13.2 Properties

ADInfo() As CADInfo

Resolution() As Long

ScanSize() As Long

ImageScale() As CScale

3.13.3 Public

Friend Sub DrawPoint(x As Long, y As Long, signal As Single)

3.13.4 Private

Private Sub Form_Paint()

Private Sub Resize_Form()

3.14 CSTM_Image

3.14.1 Events

Public Event ColormapHasChanged()

Public Event ScaleHasChanged(ByVal AbsMin As Single, ByVal min As Single, ByVal max As Single, ByVal AbsMax As Single)

3.14.2 Properties

ImageScale() As Cscale

Skala

ADInfo() As CADInfo

AD-Infos der Spitze

UserCanUnload() As Boolean

Kann der Benutzer das Formular schliessen?

Resolution() As Long

Auflösung des Bildes in Pixeln

ScanSize() As Long

Rastergröße in nm

SampleInfo() As String

Kommentar zur Probe

3.14.3 Public

Friend Sub RedrawScale()

Zeichnet die Skala neu

Friend Sub DrawPoint(ByVal x As Long, ByVal y As Long, ByVal signal As Single)

Malt einen Punkt an die Position (x,y)

3.14.4 Private

Private Sub Form_Paint()

Zeichnet die Skala und beschriftet das gesamte Bild

Private Sub DrawBox(x1 As Long, y1 As Long, x2 As Long, y2 As Long, color As Long, Optional ByVal cleaned As Boolean = False)

Zeichnet einen Kasten

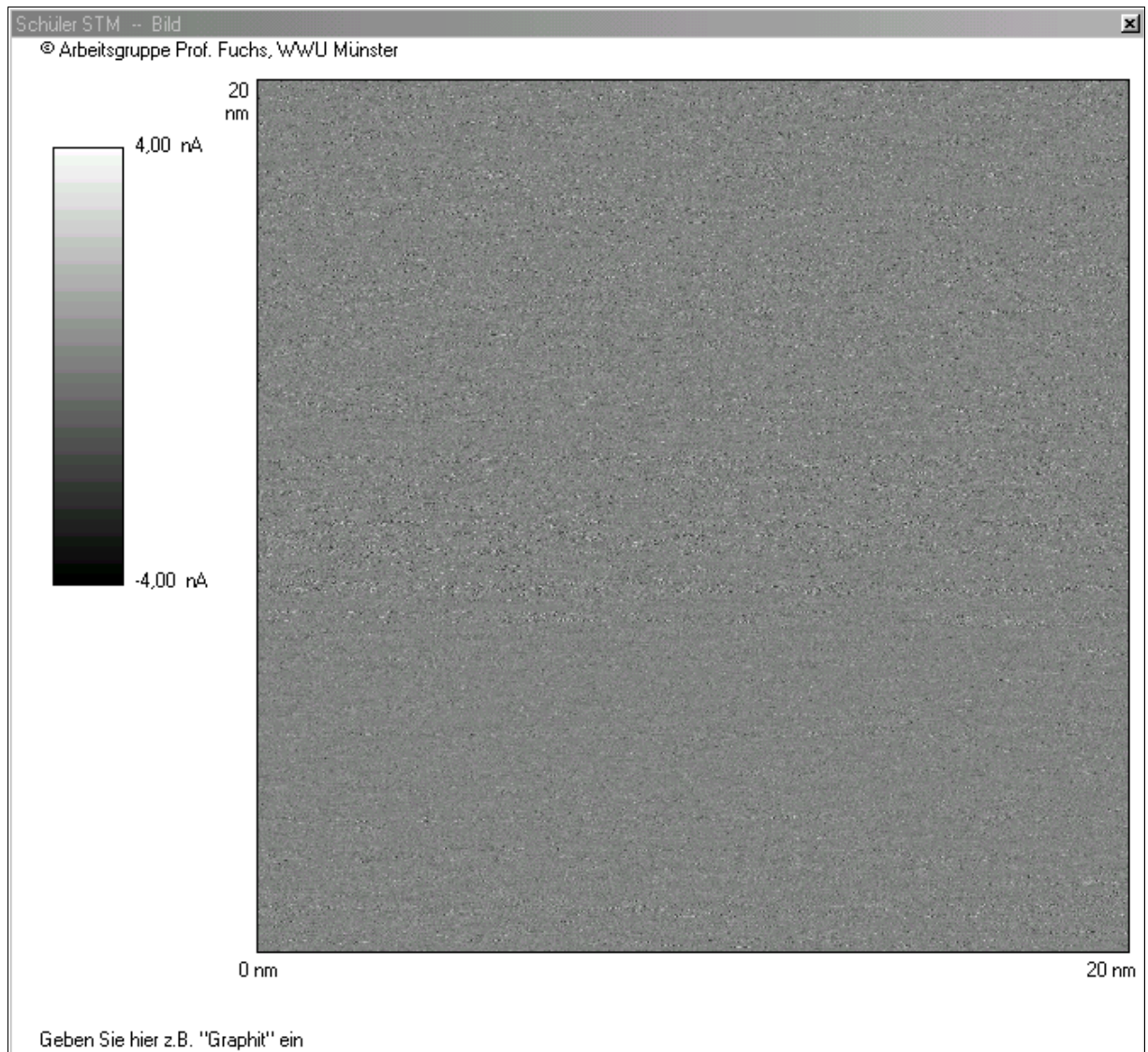
Private Sub WriteText(ByVal x As Long, ByVal y As Long, ByVal t As String, Optional ByVal clean As Boolean = False)

Schreibt einen Text und löscht gegebenenfalls vorher die Stelle

4 Formulare

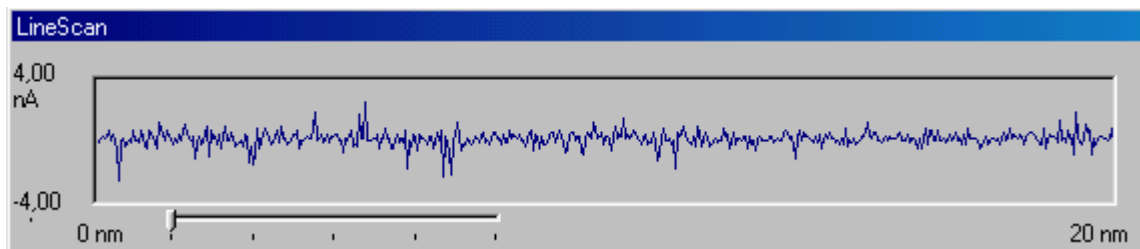
4.1 *CSTM_Image.frm*

Auf diesem Formular wird das STM-Bild dargestellt. Diese Formular wird wie eine Klasse verwendet.



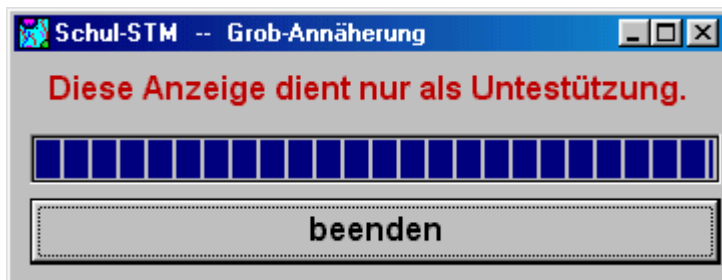
4.2 CLineScan.frm

Auf diesem Formular wird ein Linienprofil dargestellt. Diese Formular wird wie eine Klasse verwendet.



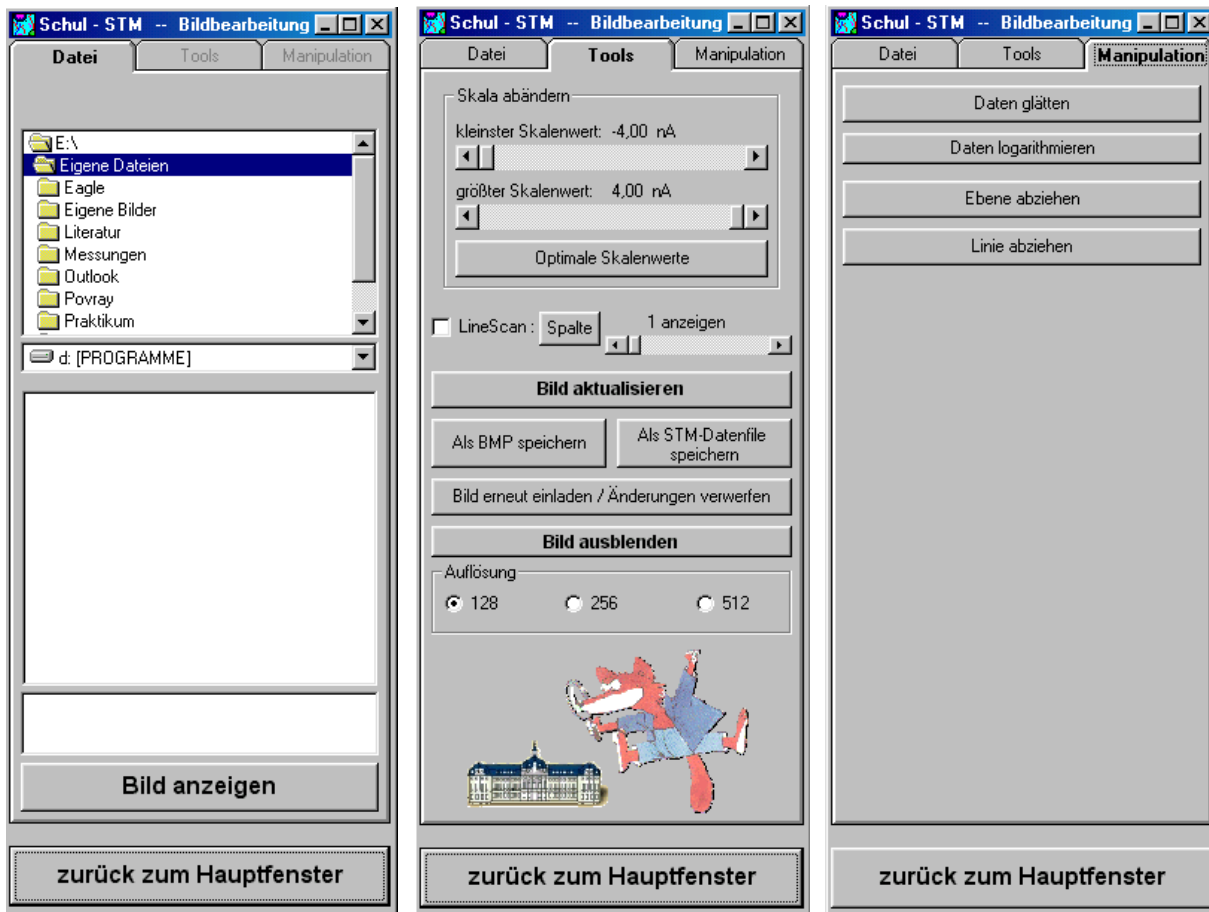
4.3 frm_coarseApproach.frm

Dieses Formular dient zur Unterstützung bei der Annäherung der Spitze an die Probe. Die Balkengrafik zeigt den momentan gemessenen Tunnelstrom an.



4.4 frm_EditImages.frm

Dieses Formular wird aufgerufen, wenn der Benutzer in den Bildbearbeitungsmodus wechselt. Es stellt sämtlich Funktionen zur Verfügung



4.5 frm_ScanFrame.frm

Dieses Formular wird aufgerufen, wenn der Benutzer in den Bildaufnahmemodus wechselt. Es stellt sämtliche Funktionen zur Verfügung

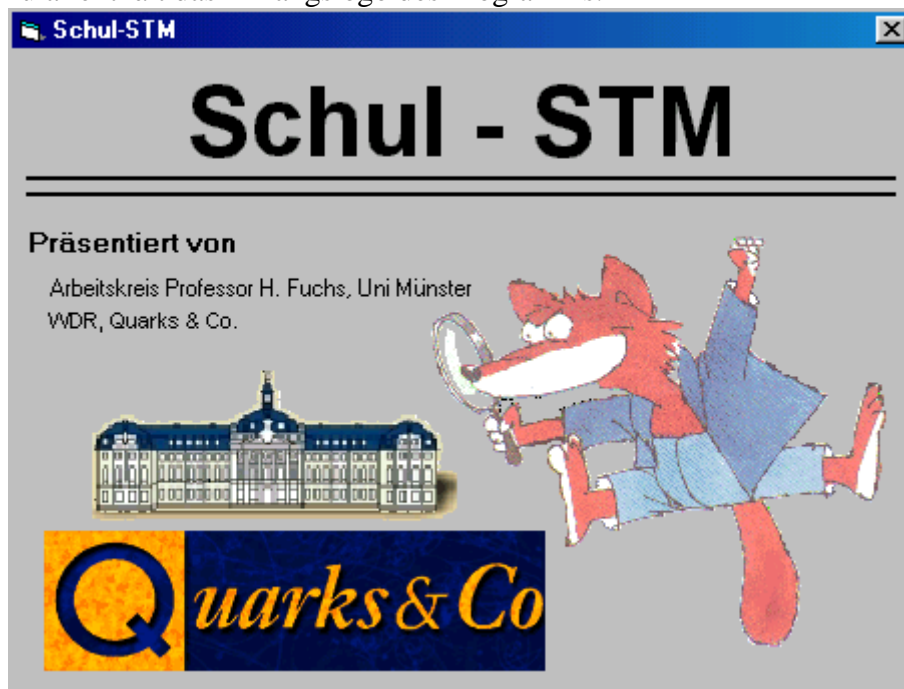
The screenshot shows a software window titled "Schul-STM" with a menu bar containing "Scan starten", "Optionen", and "Exit". The interface is organized into several sections:

- Scanrichtung:** Three radio buttons for "von links", "von rechts", and "Bidirektional" (which is selected).
- Scangeschwindigkeit:** A slider control with a label "? Hz".
- Größe des Scanbereichs:** A text box showing "100 nm x 100 nm" with a slider below it.
- Auflösung – (Punkte pro Linie):** Three radio buttons for "128", "256", and "512" (where "128" is selected).
- Verschiebung des Rasterrahmen:** Two sets of slider controls for "X: 150 nm" and "Y: 150 nm", and a small square icon to the right.
- Bilder speichern:** Four buttons: "kein Bild", "nächstes Bild", "dieses Bild", and "Aktuelles Bild sofort speichern".
- LineScan sichtbar:** A checkbox that is currently unchecked.
- Skala an die erste Zeile anpassen:** A checkbox that is currently unchecked.
- Scan starten:** A large red button with the text "Scan starten".

At the bottom, there is a copyright notice: "© Arbeitsgruppe Prof. Fuchs", "Westfälische Wilhelms-Universität Münster,", and "Physikalisches Institut".

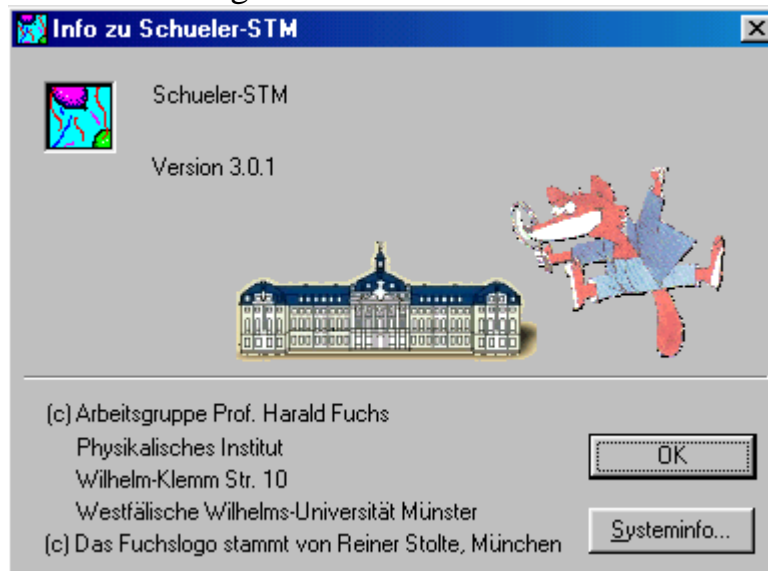
4.6 frm_StartUp.frm

Dieses Formular enthält das Anfangslogo des Programms.



4.7 frmAbout.frm

Der Info-Dialog



4.8 Main.frm

Das Hauptprogramm



5 Module

5.1 API_Interface

Public Declare Sub Sleep Lib "kernel32" (ByVal dwMilliseconds As Long)

Wartet die angegebene Zeitdauer, bevor das Programm weiter ausgeführt wird. Während der Wartezeit werden allerdings keine Messages verarbeitet.

Public Function GetSpecialFolder(RootFolder As ShellSpecialFolderConstants) As String

Ermittlet aus der Registrierung den Pfad des angegebenen Systemsordner.
Die Konstanten sind im Modul in einem Enum-Feld definiert.

Public Sub Beep()

Ruft die API-Funktion Beep auf

5.2 Konstanten

Public Const FileExt = ".STM"

Standard Erweiterung der Datendateien

Public Const MaxScanSize = 400

Gibt den, von den Stapelpiezos vorgegebenen, maximalen Rasterbereich in nm an.

5.3 MathModul

Die folgenden Funktionen sind aus

William H. Press et al.: *Numerical Recipies, The Art of Science Computing (Fortran Version)*,
Cambridge University Press

entnommen. Für eine detaillierte Dokumentation bitte ich dort nachzuschlagen

Public Sub MedFit(x() As Single, y() As Single, a As Single, b As Single)

[Seite 554f]

Public Sub Fit(x() As Single, y() As Single, a As Single, b As Single)

[Seite 508f]