

0.1 Erklärung der LabVIEW Programme

0.1.1 Allgemeines

Allgemeine Kenntnisse in LabVIEW werden vorausgesetzt. VIs, die eigentliche Treiberfunktionen erfüllen, sind in einer Bibliothek (.lib) zusammengefasst und sollten dort auch bleiben. In diesen finden sich auch die "Type-definition" von Controls (als eine Art Template zu verstehen), insbesondere den aufwendigen Popupmenus. Alle VIs und Libraries zu einem Gerät sind in einem nach dem Gerät benannten Ordner abgelegt. Die Kopien der VIs sollten nach dem Öffnen auf keinen Fall gespeichert werden; oft sind Drittbibliotheken eingebunden, teilweise sind die Programme für LabVIEW 4.0 oder 5.0 und zudem für PC oder Mac bestimmt (teilweise kann sonst nur das als TopLevel-VI designierte VI auf der anderen Plattform geöffnet werden).

Die Popupmenus sollten nur geändert werden wenn man sich der Konsequenzen bewusst ist (je nachdem ist die Position oder der Menustring innerhalb des Popupmenus entscheidend). Connector Panes wurden prinzipiell grosszügig gewählt damit auch nachträglich gut weitere Variablen verdrahtet werden können. Wo viele Variablen auftreten werden diese in Cluster zusammengefasst.

Alle VIs enthalten eine History in der die wesentlichen Änderungen kommentiert sind. Die neusten Versionen der VIs sind auf den Messrechnern zu finden, die Dokumentation hier. Mit Ausnahme der VIs für den PREMA 5000 Scanner wurde die Möglichkeiten der Dokumentation oder Hilfe in den VIs nie benutzt.

In [an11.pdf](#) (Application Note 11) sind die Richtlinien von National Instruments zur Aufnahme von Treibern in ihr "Driver Network" klar umschrieben. Diese werden von keinem meiner Programme nur annähernd erfüllt, da sie nicht die VISA-Treiber verwenden (d.h. üblicherweise entweder nur für die serielle Schnittstelle, GPIB oder DAQ ausgelegt sind), nicht die Dokumentationsmöglichkeiten benutzen, nicht anhand eines einheitlichen Schemas auf verschiedene VIs verteilt sind etc.

0.1.2 Gasflussregelung via RS-485

Die für die Erweiterung der Gasmessstation angeschafften Gasflussregler von MKS können sowohl analog (wie die alten Gasflussregler) wie auch digital via RS-485 gesteuert werden. Die Analog-schnittstelle ist nur noch für die Rückwärtskompatibilität und die Spannungsversorgung vorgesehen, die gesamte Kommunikation geschieht via RS-485 (two-wire).

Zur Ansteuerung via PC wird ein geeigneter RS-232 zu RS-485 Adapter benötigt (Unterscheidung zwischen Senden und Empfangen durch RTS, DTR, CTS und DSR), ein Nullmodem (nur Linien 2-3, 4-5 und 6-20 jeweils kreuzweise verbunden) sowie zwei selbstgebaute cross-over RJ-12 Kabel, da die Pinbelegung an den Gasflussreglern und am RS-232 zu RS-485 Adapter nicht übereinstimmen. Die zwei neuen Gasflussregler haben die Adressen "002" und "003". Details dazu im L5 S.50, L6 S.3.

In [MKSMassFloViaRS-485.lib](#) sind sämtliche VI's enthalten, die zur Ansteuerung gebraucht werden. Zuerst muss [InitSerialInterfaceForMFC.vi](#) aufgerufen werden, damit die serielle Schnittstelle richtig initialisiert wird (9600 bauds sowie sämtliche Hardware handshaking auf true). Danach können mit dem VI [SendCommandOrRequest.vi](#) sämtliche möglichen Parameter des Gasflussreglers abgerufen (Request) oder geändert werden (Command), dementsprechend ist

<SendCommandOrRequest.vi> als universales SubVI gedacht zur Kommunikation mit den Gasflussreglern. Ein Pop-upmenu zeigt alle vorhandenen Befehle sowie weitere Informationen (z.B. mögliche Eingabewerte), siehe Fig. 0.1.

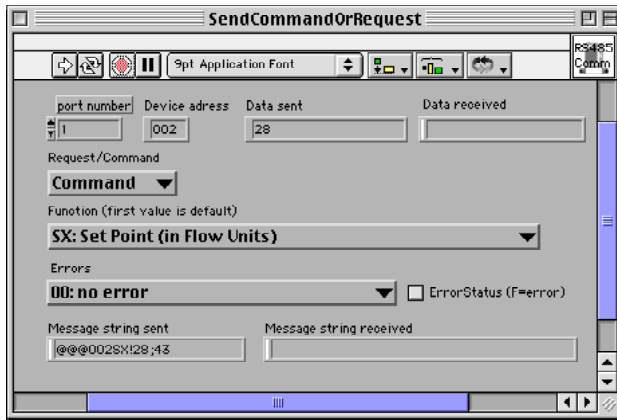


Fig. 0.1 Panel des SenCommandOrRequest.vi

Die Befehle bestehen aus Setup, Control, Flow Sensor, Calibration und Informational Messages gemäss dem Manual "MKS Mass-Flo[®] Device RS-485 Digital Interface - Supplement" [MKS Instruments, 1999#149] und sind dort detailliert erklärt. Das VI <CalculateChecksum.vi> führt die Berechnung der Quersumme sowohl der ausgehenden wie auch eingehenden Botschaften aus gemäss der geräteeigenen Konvention (mit der Einstellung "FF" wird die Ueberprüfung der Quersumme ignoriert, die Quersumme wird gegebenenfalls mit einer Null auf zwei Stellen aufgefüllt). Achtung: bei kleinen Buchstaben arbeitet das VI noch nicht korrekt (ASCII/ANSI-Problem).

<ErrorAnalysis.vi> wertet die Fehlermeldungen der Gasflussregler aus und trennt den Antwortstring ab. Die vom Benutzer eingegebenen Werte werden nicht überprüft, ungültige Parameterwerte ergeben eine Fehlermeldung.

Zwischen dem Schreiben und dem Empfangen von Botschaften via serielle Schnittstelle wird jeweils 100 ms gewartet damit die Antwortstrings sicher nicht abgeschnitten werden. Stimmt die Quersumme der empfangenen Botschaft nicht wird der ganze Vorgang höchstens zweimal wiederholt.

In <TestNOxGasanlage.llb> findet sich eine rudimentäre Version einer bedarfsgesteuerten (d.h. Kommunikation erfolgt nur wenn Sollwerte geändert oder Istwerte abgefragt werden müssen) Gasflusssteuerung.

0.1.3 Gasflussregelung via DDE-Server

Siemens-Cerberus benützt 4 analoge und einen digitalen Gasflussregler von Bronkhorst. Die vier analogen Geräte sind an einer digitalen Auswerteeinheit Bronkhorst E-7000 angehängt, sodass für die Steuerung via FLOW-BUS alle Gasflussregler equivalent erscheinen. Die Kommunikation mit dem FLOW-BUS wird durch einen Dynamic Data Exchange (DDE, ein älteres Microsoft Datenprotokoll zum Datenaustausch zwischen Programmen auf PCs) Server sichergestellt. Die DDE-Server Software muss dabei stets im Hintergrund laufen. Unter Umständen beansprucht LabVIEW selbst zuviel Ressourcen (z.B. ständige Impedanzmessungen), sodass das Aktualisieren sehr langsam wird (bis halbminütige Verzögerungen).

Die Bibliothek *<Gasflussregelung.llb>* enthält alle VIs zur Kontrolle der 5 Gasflussregler via DDE-Server. *<FlowDDEClient.vi>* ist ein primitiver Client, währenddessen *<DDETest.vi>* sämtliche Parameter zur Wartung des Gasflussregelsystems via FLOW-BUS implementiert.

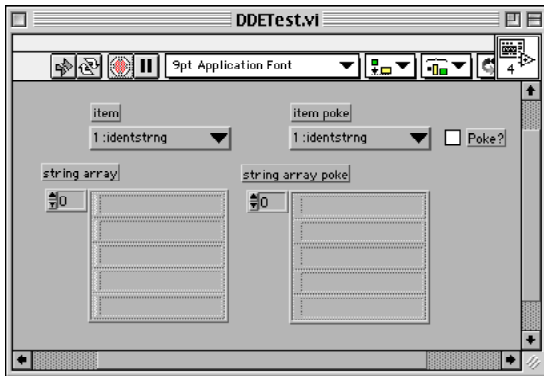


Fig. 0.2 Panel des DDETest.vi

Angaben zur Konfiguration des FLOW-BUS/DDE-Servers bei Siemens-Cerberus finden sich in L5 S46, 48-49., L6 S.6.

0.1.4 PE LS-50B Lumineszenzspektrophotometer

Der Hauptteil der PE LS-50B Parameter die sowohl von Timedrives wie auch Scans benützt werden ist in *<PE LS-50BInit.vi>* festgelegt (Fig. 0.3).

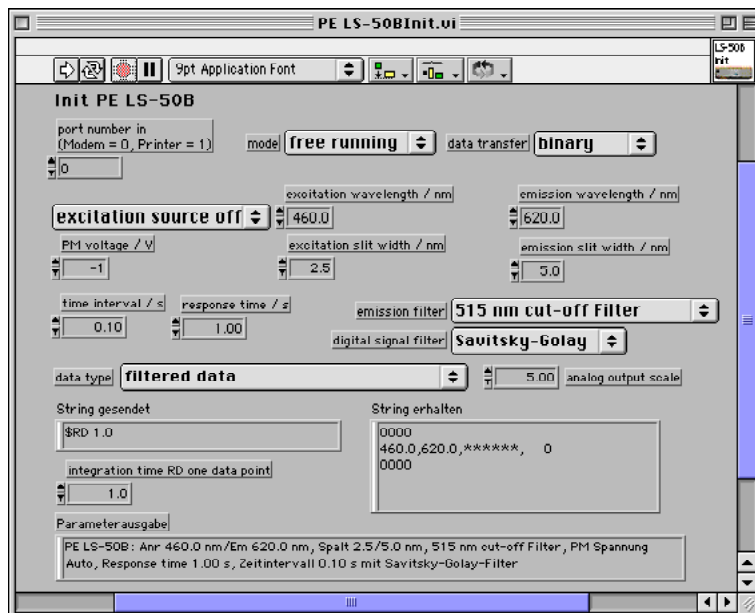


Fig. 0.3 PE LS-50BInit Panel
Gezeigt sind die Standardeinstellungen.

Mit *<PE LS-50BStartScan.vi>* wird ein Scan konfiguriert (Fig. 0.4).

Das *<PE LS-50BSingleScan.vi>* ist als Top-VI gedacht während der Spektrenaufnahme (Fig. 0.5).

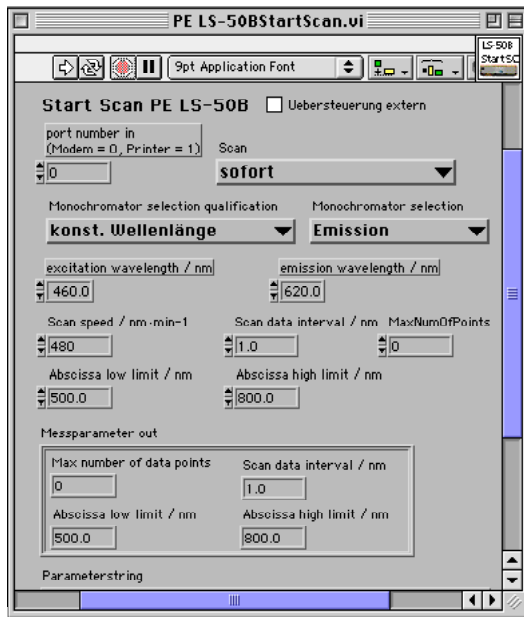


Fig. 0.4 PE LS-50BStartScan Panel

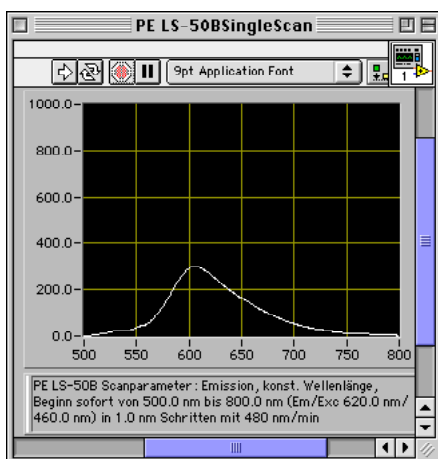


Fig. 0.5 Panel des PE LS-50BSingleScan.vi

0.1.5 Gilson Minipuls 3

Die Gilson Minipuls 3 Peristaltikpumpe kann mit dem <GilsonMinipuls3.llb> (Hauptpanel in Fig. 0.6) vollumfänglich via RS-232 und dem RS-232 Adapter von Gilson angesteuert werden. Die serielle Verbindung ist aus irgendwelchen Gründen schlecht, wodurch die Minipuls 3 Befehle ab und zu erst nach einigen Sekunden ausführt (das LabVIEW Programm beginnt jeden neuen Versuch mit einer Initialisierung). Siehe auch L5 S.5-6.

0.1.6 Lumineszenzmessungen an Sauerstoffmembranen

In der <O2-Lumineszenzmessung.llb> (zusammen mit <PE LS-50B.llb> und <GilsonMinipuls3.llb>) findet sich alles zur Ansteuerung einer kompletten Sauerstoffmembrancharakterisierung.

Auf dem Hauptpanel von <Messautomation.vi> finden sich alle wichtigen Konfigurationseinstellungen.



Fig. 0.6 Hauptpanel des GilsonMinipuls3.llb

Das Popupmenu oben links wählt den Namen einer Programmdatei aus (muss sich im Ordner "Tests", der wiederum auf der gleichen Hierarchieebene wie <O2-Lumineszenzmessung.llb> ist, befinden); wird "manuell" gewählt wird kein Programm eingelesen. Das Stringcontrol darunter dient zur Eingabe eines Beschreibungstextes welcher im Header der Datendatei zusammen mit Datum/Zeit/Name der Datei/Name der Programmdatei abgespeichert wird. Der Pfadindikator darunter zeigt den Pfad der aktuell niedergeschriebenen Datei an.

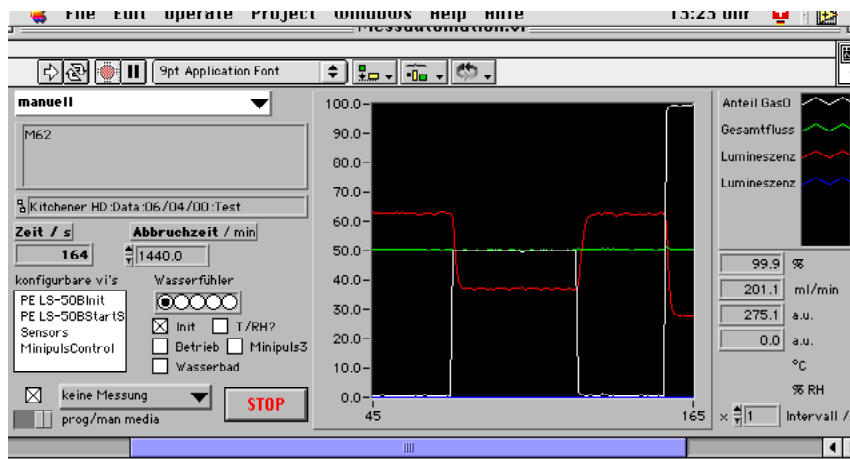


Fig. 0.7 Hauptpanel (Messautomation.vi) mit Konfiguration/Resultaten

In der Liste "konfigurierbare vi's" können die VIs ausgewählt werden, die vorgängig aufgerufen werden sollen (um einzelne Parameter ändern zu können). Der Booleanarray rechts davon gibt den Zustand der 5 implementierten Wasserfühler wieder. Darunter finden sich die Boolean-Einstellungen welche für die Initialisierung gebraucht (Minipuls3 kann auch noch danach gesetzt werden) werden:

"Init" startet das <PE LS-50BInit.vi> (Weglassen bringt Zeitgewinn falls es bereits vorgängig aufgerufen worden ist und die Parameter nicht geändert werden müssen), "T/RH?" falls das rotronic A2 hygrometer, "Minipuls3" falls die Gilson Minipuls 3 benutzt werden soll und "Wasserbad" falls Soll-

und Istwert der Wasserbadtemperatur des *Neslab EX-210* abgelesen werden sollen. "Betrieb" wird automatisch gesetzt (False = Abbruch von *Messautomation.vi*). Sind diese Booleans zu Beginn falsch gesetzt so werden die entsprechenden Werte nicht aufgezeichnet resp. - falls die entsprechenden Geräte nicht in Betrieb sind - nur unsinnige Werte erhalten.

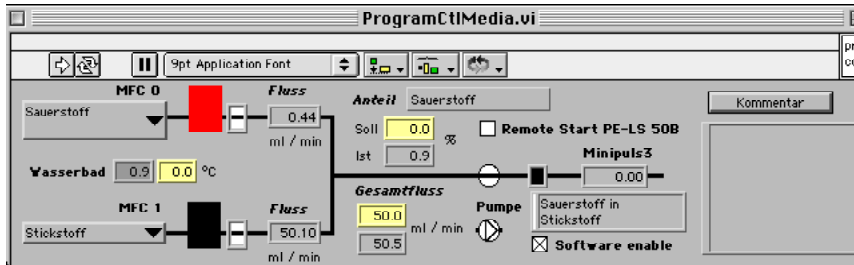


Fig. 0.8 Panel mit den experimentellen Einstellungen (ProgramCtlMedia.vi)

Die Entsprechung mit "PE LS-50B?" (ohne Label, links vom Popupmenu weiter unten) ist eigentlich jetzt unnötig (könnte durch "PE LS-50B? out" ersetzt werden). Ist dieser auf False kann nicht mit dem PE-LS 50B gemessen werden. Das Popupmenu rechts davon wird ferngesteuert. Der "STOP"-Knopf ist der einzige saubere Weg um das ganze Programm vorzeitig zu beenden. Der Graph zeichnet die wichtigsten Parameter auf, desweiteren können diese auch rechts an den jeweils nach Bedarf konfigurierten Indikatoren abgelesen werden. Die Intervallzeit beeinflusst nur die Darstellung auf dem Graph und dient dazu bei Experimenten mit langen Intervallzeiten den Ueberblick zu behalten.

Sehr wichtig ist der Umschalter "prog/man media". Damit kann jederzeit von einer programm-gesteuerten Messung (Kontrolle durch <ProgramCtlMedia.vi>) zu einer manuell-gesteuerten (Kontrolle durch <ManualCtlMedia.vi>) und zurück gewechselt werden. Für ersteres gilt die Programmabbruchzeit, für letzters die manuelle Abbruchszeit (ist beim Umschalten die jeweilige Zeit schon überschritten wird das Programm sofort abgebrochen!). Durch den Umschalter werden die quasi identischen Panels von <ProgramCtlMedia.vi> (Fig. 0.8) resp. von <ManualCtlMedia.vi> aufgerufen.

Alle relevanten experimentellen Parameter werden hier gezeigt. Gelb dargestellt sind die Soll-, grau die Ist-Werte. Falls Soll- und Ist-Werte signifiant voneinander abweichen werden die Gasflussreglersymbole rot. Ventile und Pumpen sind als Rings mit verständlichen Symbolen versehen die den jeweiligen Zustand veranschaulichen. Die eigentlich vorgesehen Auswahl der Gasparameter via Gaspopupmenu ist nicht implementiert. Durch Drücken des "Kommentar"knopfs wird der Wert des Stringindicators in die Fehlerzeile der gespeicherten Daten geschrieben.

Durch <Sensors.vi> (Panel in Fig. 0.9) können Fehlermeldungen konfiguriert werden wenn ein bestimmter Kanal im Inputport C auf logisch high oder low geht. Diese werden dann in die Fehlerzeile der Daten abgespeichert.

0.1.7 PREMA 5000 Scanner

Die PREMA 5000 Scanner sind Multimeter bei Siemens-Cerberus die via GPIB nach IEEE 488 betrieben werden.

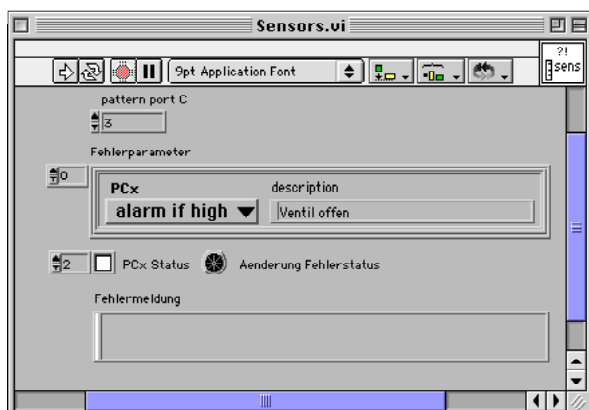


Fig. 0.9 Setzen der Sensoren in Sensors.vi

In *<MultiPREMA5000Messung.llb>* befinden sich sämtliche VIs die zur Messung von Spannungen, Widerständen etc. auf mehreren Kanälen und PREMA Multimetern nötig sind. Als TopVI wird das *<MultiPREMA5000Messung.vi>* benützt. Erwähnenswert ist *<OpenDataFile.vi>*, welches die Daten nach Ueberschreiten einer gewissen Zeilenanzahl und/oder Zeit in eine neue, fortlaufendnummerierte Datei schreibt.

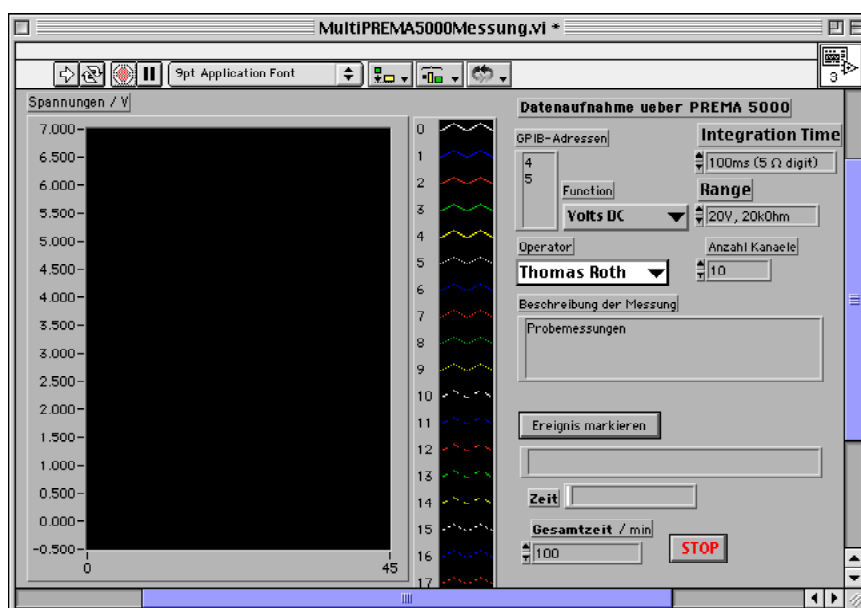


Fig. 0.10 Panel des MultiPREMA5000Messung.vi

Zu einem Fehler im PREMA 5000 Scanner siehe L5 S.15, zum Programm auch L5 S.40.

0.1.8 HP 4274A

Das HP 4274A ist eine ältere Frequenzmessbrücke von Hewlett-Packard die bei Siemens-Cerberus im Einsatz ist. Sie kann an 12 diskreten Frequenzen zwischen 100 Hz und 100 kHz Impedanz, Widerstand, Leitfähigkeit, Kapazität, Induktivität etc. messen. Die Ansteuerung erfolgt über GPIB nach IEEE 488.

<HP 4274A.llb> beinhaltet die Programme <HP4274A Single Measurement.vi> und <HP4274A Frequency Scan.vi>. Beides sind in sich voll funktionsfähige VIs die auch als SubVI benützt werden können.

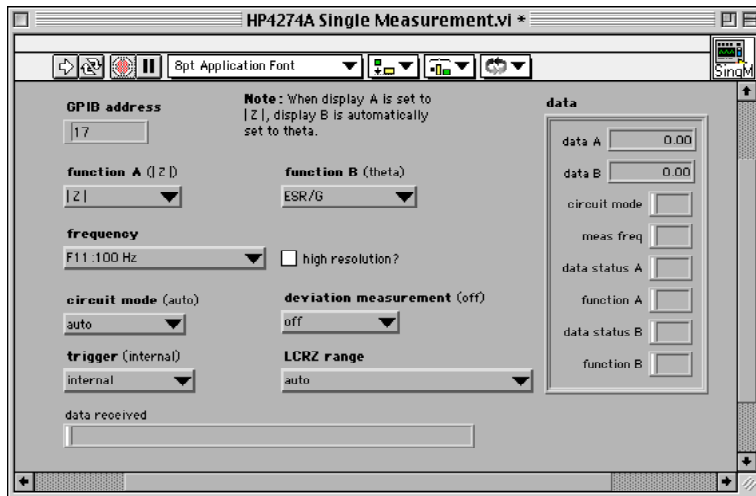


Fig. 0.11 Panel des HP4274A Single Measurement.vi

<HP4274A Single Measurement.vi> konfiguriert das HP 4274A vollständig. Function A und B, frequency, trigger, circuit mode, deviation measurement, LCRZ range oder high resolution mode können analog der Front Panels gewählt werden. Die default-Werte sind für eine automatisierte Messung der Impedanz gedacht. Ein allfälliger Referenzwert für eine deviation measurement oder ein Abgleich der Kabelkapazität muss vorgängig erfolgen!

Das VI wiederholt die Messung bis zu zwei Mal nach einer Pause von jeweils 1 s (nur eine neue Messung auslösen genügt nicht) falls ein Over- oder Underflow aufgetreten ist (Überprüfung anhand des vom Gerät ausgegebenen Strings). Dieses "Versagen" der automatischen Bereichswahl tritt beim Wechsel von hohen zu tiefen Frequenzen auf. In einigen Fällen kann das Gerät in *allen* Messbereichen keinen Wert liefern, z.B. bei Widerstandsmessungen oder natürlich auch wenn ein manuell gewählter Messbereich überschritten wird. In diesen Fällen verzögert sich die Messung um bis zu 3 s, was insbesondere bei Messungen bei mehreren Frequenzen sehr erheblich werden kann.

Zur eigentlichen Kommunikation verwendet das VI das <HP4274A GPIB Communication.vi>, welches den Konfigurationsstring abschickt, wartet bis der Antwortstring bereitliegt und diesen abfragt. Aus dem Datenstring werden in einem Cluster die Werte der Funktionen A und B als Zahlen sowie die weiteren Statusparameter bereitgestellt.

<HP4274A Frequency Scan.vi> verwendet <HP4274A Single Measurement.vi> um bei mehreren Frequenzen hintereinander zu messen. Eine beliebige Kombination von Frequenzen kann aus der Auswahlliste des front panels gewählt werden. Die Frequenzen werden dabei immer von oben nach unten gemessen, d.h. von tieferen zu höheren Frequenzen. Die momentan verwendete Frequenz wird im Statusfenster angezeigt.

Sehr wichtig ist wie das VI die Wahl der Frequenzen handhabt falls deren Reihenfolge geändert werden soll oder die Sonderfrequenzen benutzt werden. Die Wahl der Frequenzen wird aus deren Position in der Auswahlliste bestimmt, der Wert aber durch den String der Auswahl. Würde der

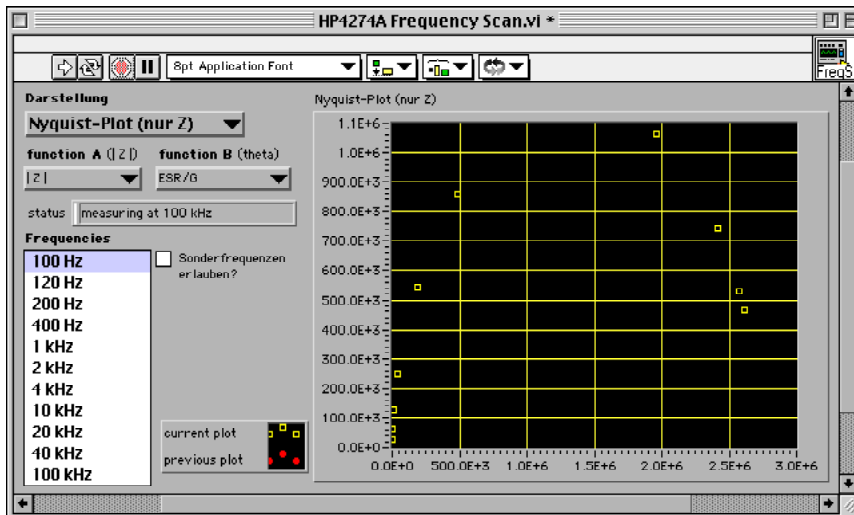


Fig. 0.12 Panel des HP4274A Frequency Scan.vi

String "100 Hz" in der Auswahlliste in "105 Hz" geändert würde das Gerät trotzdem noch bei 100 Hz messen, aber ausgeben es habe bei 105 Hz gemessen. Die Option "Sonderfrequenzen erlauben" blendet diese ein (ein Schutz dass sie nicht aus Versehen angewählt werden), vor der Messung müssen aber UNBEDINGT noch mit dem Textwerkzeug die Frequenzen nach dem Schema "Zahl<Leerschlag>Einheit" (wie bei den anderen Frequenzen in die Auswahlliste) anstelle von "*1" oder "*2" eingetragen werden! Es gibt keine Möglichkeit aus dem Gerät die Größe der Sonderfrequenzen abzufragen! Die verwendete Einheit muss dabei exakt "Hz" oder "kHz" (sonst wird die Frequenz um Faktor 10^3 zu klein) sein.

Für die Darstellung des Graphs kann ein Nyquist-Plot gewählt werden (macht nur bei $|Z|$ und Theta-Messung einen Sinn) oder eine Darstellung der Funktion A oder B vs f. Bei ersterem wird automatisch eine lineare, bei den beiden letzteren eine logarithmische Darstellung gewählt und der Graph jeweils entsprechend betitelt. Die zuletzt aufgenommene Messreihe wird ebenfalls angezeigt (dies ist nicht der Fall falls gerade die Darstellungsart, Funktion A oder B geändert wurde). Eine neue Messreihe wird erst dargestellt wenn alle zugehörigen Frequenzen gemessen worden sind. Mit der Funktion Referenz kann zudem zu einem beliebigen Zeitpunkt ein Impedanzspektrum als Referenz aufgenommen werden, das ebenfalls dargestellt wird.

Bei der Verwendung als SubVI gibt *<HP4274A Frequency Scan.vi>* einen eindimensionalen array mit 5 Einträgen aus: Frequenz, Startzeit (relativ zur ersten Messung), Wert der Funktion A, Wert der Funktion B, berechneter Wert 1 und berechneter Wert 2 (bei $|Z|$ und Theta ist dies Z' resp. Z'' ; bei anderer Wahl von A ist dies immer 0). Alle Werte sind in entsprechenden SI-Einheiten Sekunden, Ohm, Farad etc.

0.1.9 NO_x-Messung bei Siemens-Cerberus

Aufbauend auf *<HP 4274A.IIb>* und *<Gasflussregelung.IIb>* wird durch das TopLevel-VI *<NO_x-Messung.vi>* die gesamte Messung koordiniert.

Das VI *<SaveDataForIgorPro.vi>* speichert die Daten in einer für die Weiterverarbeitung mit der Igor-Prozedur *<ImpedanceUtilities>* optimalen Form. Die gemessenen Impedanzspektren werden nur abgespeichert wenn sich die Impedanz relativ oder die Phase absolut um einen gewissen

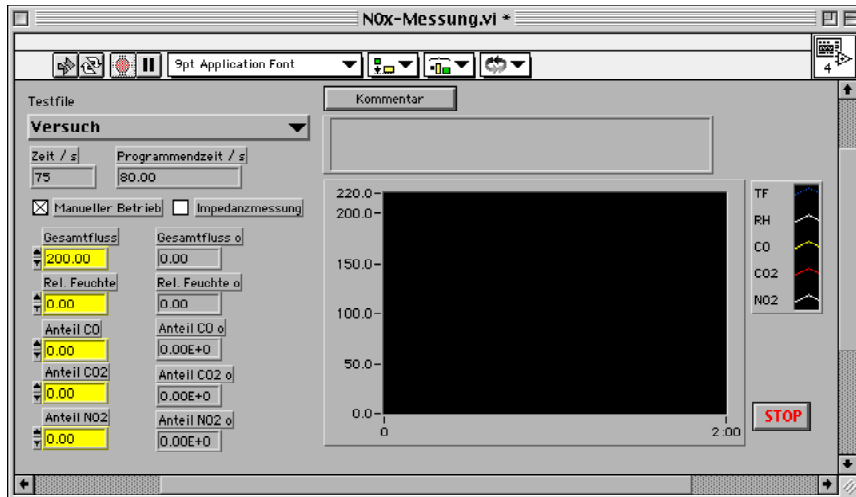


Fig. 0.13 Panel des NOx-Messung.vi

Betrag geändert hat. Dadurch können schon bei der Datenaufnahme unnötige Daten eliminiert werden. Die eingestellten oder abgeleiteten Parameter wie Zeit, Gesamtfluss, relative Luftfeuchtigkeit oder Gaskonzentrationen hingegen werden in regelmäßigen Zeitabständen niedergeschrieben.

Durch die Form einer "Igor Text" -Datei ist es möglich, Daten in ein Igor-Experiment zu laden die - ohne weiteres Zutun - im Experiment mit einem klar definierten Namen zur Verfügung stehen. In folgendem Beispiel

```
Igor
Waves Welle1, Welle2
Begin
1.00, 2.00
End
X Time[0] = 3
```

werden die Wellen "Welle1" und "Welle2" mit ihren Daten (markiert durch "Begin" und "End") geladen und der (zwingend vorhandene!) Welle "Time" wird dem Index 0 der Wert 3 zugewiesen. Hinter einem "X" kann eigentlich jeglicher Igor-Befehl (analog der Kommandozeile) untergebracht werden. In *<SaveDataForIgorPro.vi>* werden damit vom Stamm des Dateinamen abgeleitete Wellen für den Gesamtfluss, Relative Luftfeuchte, die NO₂, CO₂ und die CO-Konzentration generiert mit N = 1000. Im Lauf des Dateneinlesens werden diesen Daten dann analog dem Beispiel Werte zugewiesen. Die Wellen der Impedanzspektren (Endungen "fX", "ZY", "PY" für die Frequenz, Impedanz und Phase in Grad) werden ausgehend vom Stamm des Dateinamen von 000 bis 999 durchnummeriert und nur im Bedarfsfall erzeugt. Wegen der Redundanzprüfung wird deshalb im Extremfall (keine Änderung) nur ein einziges - das erste - Impedanzspektrum gespeichert.

Damit über längere Zeit gemessen werden kann wird nach 1000 Datenwerten automatisch eine neue Datei generiert. Die Dateinamen werden nach dem Muster Stamm + dreistellige Zahl von 0 bis 999 gebildet. Für die Auswertung in IgorPro ermöglicht dies "fortlaufend nummerierte" Wellen zu haben sodass Daten aus verschiedenen aufeinanderfolgenden Dateien sehr einfach integriert werden können. Andererseits bietet die Aufspaltung in einzelne Dateien eine gewisse Datensicherheit

und sichert vernünftige Verarbeitungszeiten falls die Redundanz sehr klein sein sollte und die Dateigrösse leicht in die MegaBytes geht.

0.1.10 Spekol 1100

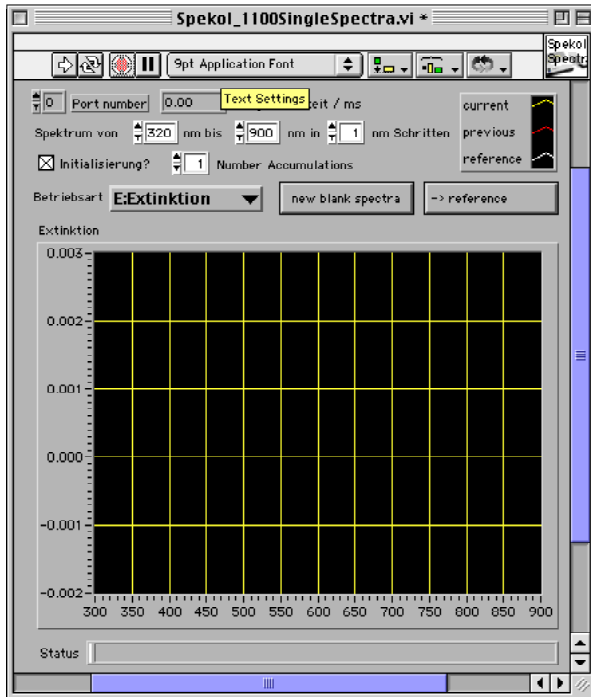


Fig. 0.14 Hauptpanel der Spekol_1100.llb

Die *<Spekol_1100.llb>* ermöglicht die fast vollständige Ansteuerung des *Spekol 1100* (funktioniert auch beim *Spekol 1200*, wenn auch anderer Wellenlängenbereich) gemäss den Spezifikationen die wir von Kurt Merz von der Contrec Technologies AG, Dietikon am 26/5/00 erhalten haben. Verwendet wird ein RS-232 Standardkabel. Die Philosophie lehnt sich stark an das VI *<HP 4274A.llb>*, siehe "HP 4274A" auf S. 7. Auch hier ist *<Spekol_1100SingleSpectra.vi>* (Fig. 0.14) ein Standard-VI, welches sich zur Einbettung als SubVI eignet. Ausgegeben wird ein 2D-array mit den Wellenlängen in nm sowie den Extinktionen.

Die Betriebsart sowie die Wellenlängen-Einstellungen auf dem Panel haben keinen Einfluss (falsche Spezifikation?). Wird der Boolean "Initialisierung?" gesetzt so wird zwingend ein Blank aufgenommen (d.h. Probe muss entfernt sein!) und die Integrationszeit bestimmt, die unter "Integrationszeit" angezeigt wird. Mit dem Knopf "new blank spectra" ist es möglich, auch während Langzeitexperimenten zwischendurch wieder einen Blank aufnehmen zu können.

Die eigentliche Spektrenaufnahme dauert nur rund 3 Sekunden, der Transfer via RS-232 zum Computer hingegen rund 10 Sekunden. Somit können problemlos 4 Spektren/Minute gemessen werden.

Das Messen bei einer Liste von einzelnen Wellenlängen, das Abfragen des Labels oder des Spektralbereichs sowie das Setzen der Integrationszeit sind nicht implementiert. Diverse Parameter werden vom Gerät ignoriert, d.h. es wird bei einer Messung immer die Extinktion im Bereich von 320-900 nm in 1 nm Schritten gemessen. Der Transfer via RS-232 wartet bis zum Senden des *<cr>*. Fehlerprüfungen mit entsprechenden -meldungen werden nur generiert beim Initialisieren der seri-

ellen Schnittstelle oder bei der ausgegebenen Integrationszeit nach der ersten Spekol 1100-Initialisierung.

Das <Spekol_1100SaveDataForIgorPro.vi> ist eine Weiterentwicklung des <SaveDataForIgorPro.vi>. Es können leicht weitere numerische Wellen angehängt werden, deren Suffixe in dem entsprechenden Suffixcontrol nachgeführt werden müssen. Achtung: sämtliche Zeilenumbrüche sind PC-freundlich als <nl> umgesetzt. Damit diese erhalten werden beim Transfer auf den Mac muss die Datei von der PC-Seite (!) herper ftp (ASCII) transferiert werden.

0.1.11 Uvikon 942

Die <Uvikon_942.llb> enthält einen kompletten Befehlssatz für die Uvikon 942-Befehle via RS-232 sowie die Implementation eines Scans. Für den Anschluss an den COM1 des Dell-Laptops wurde ein Flachbandkabel gefertigt, siehe L6 S.24. Der Anschluss erfolgt mit 9600 baud rate, 1 stop bit, 8 data bits, no parity, sw handshake on, hardcopy device HP500 auf Seite des Uvikon 942. Im LabVIEW sind die gleichen Einstellungen mit dem XON/XOFF für In- und Output zu finden.

Das wichtigste VI ist <Uvikon_942_SendSingleCommand.vi> welche mit Ausnahme sehr exotischer Befehle die ganze Funktionalität des Uvikon 942 enthält, siehe Fig. 0.15. Sämtliche Funktionen sind im Manual (Auszug im Ordner "Technische Dokumentation") beschrieben.

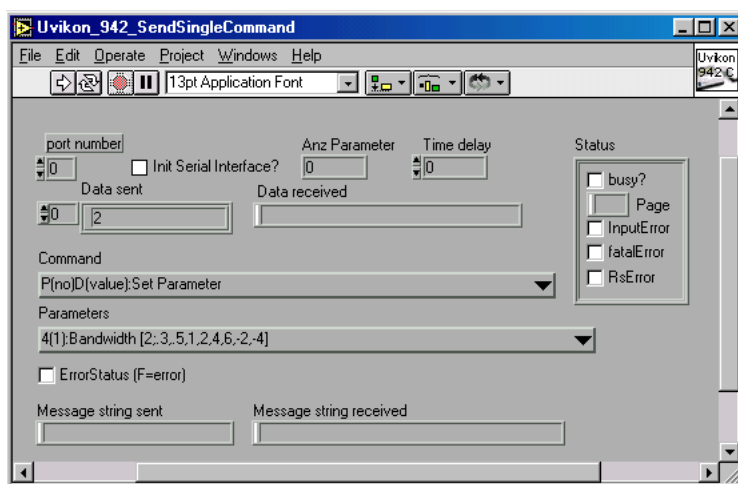


Fig. 0.15 Uvikon_942_SendSingleCommand.vi

Im Popumenu "Command" sind sämtliche Befehle enthalten, in Kombination mit der Befehl "SetParameter" wird das Popumenu "Parameter" darunter benützt, welches sämtliche einstellbaren Geräteparameter mit der Angabe von Standardwerten in []-Klammern enthält. Die Werte der Parameter können im Arraycontrol "Data sent" eingegeben werden. Nach jedem Befehl wird eine Statusabfrage durchgeführt, die rechts oben im Panel angezeigt wird. Dies führt zu einer Verzögerung von einer Sekunde oder so, ermöglicht aber einen reibungslosen Ablauf.

Extrem wichtig ist die Logik der seriellen Ansteuerung des Uvikon 942: es handelt sich um eine eigentliche Makrosprache, die nur Manipulationen erlaubt die auch manuell so ausgeführt werden können.

Mit <Uvikon_942_SingleScan.vi> können einzelne Absorptionsspektren aufgenommen werden, siehe Fig. 0.16. Wichtig ist, dass man sich zu Beginn im "UVIKON spectrophotometer functions"

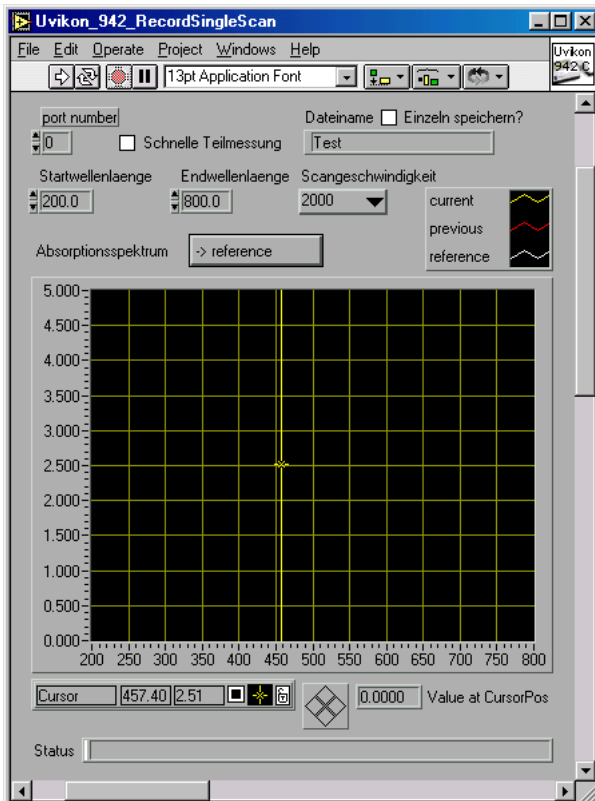


Fig. 0.16 Uvikon_942_SingleScan.vi

befindet. Das VI setzt Anfangs- und Endwellenlänge sowie die Scangeschwindigkeit, nimmt das Spektrum auf, speichert es auf der Harddisk des UVIKON 942 und transferiert es als ASCII -string über die serielle Schnittstelle zum Kontrollcomputer wo das VI es als Array ausgibt.

Die Spektren werden unter dem Namen "Dateiname" (wird auf 8 Zeichen beschnitten) auf dem Uvikon 942 gespeichert. Evtl. vorhandene Wavelength scans gleichen Namens werden einfach überschrieben. Die Option "Einzeln speichern?" ist gedacht zum Aufnehmen und Speichern einzelner Spektren direkt auf dem Kontrollcomputer, womit einiges an manueller Tipparbeit auf der Uvikon 942-Konsole entfällt. Mit der Option "schnelle Teilmessung" entfällt der ASCII-Transfer, womit die Zeit für die Aufnahme eines Spektrums fast halbiert werden kann. Die Daten müssen nachträglich vom Uvikon 942 exportiert werden.

Die graphische Darstellung erfolgt analog der VIs für den HP4274A und dem Spekol 1100. Neben dem aktuell aufgenommenen Spektrum wird auch das vorangegangene sowie ein Referenzspektrum angezeigt, welches durch den Knopf "-> reference" angewählt werden kann. Mit dem Cursor kann eine beliebige Position innerhalb des Spektrums angefahren werden, der Indicator "Value at CursorPos" zeigt dann die Extinktion an dieser Stelle an.

<Uvikon_942_SingleScan.vi> baut auf <Uvikon_942_SendSingleCommand.vi> auf, welches für die einzelnen Schritte aufgerufen wird. Damit können auch beliebige andere Anwendungen wie die Verknüpfung von Timedrives und Spektrenaufnahme mit sehr wenig Aufwand programmiert werden.

Hängt das Programm so kann mittels "break" auf der Tastatur das Ganze gestoppt werden, die serielle Schnittstelle des Uvikon 942 muss dann mit einem Durchlauf durch "Parameter" (für die serielle Schnittstelle) zurückgesetzt werden.

0.1.12 Spektroelektrochemie

Die <Spektroelektrochemie.llb> ermöglicht zusammen mit <Uvikon_942.llb> die lückenlose Aufnahme von Spektren via Uvikon 942 zusammen mit dem E_{out} und I_{out} des Autolab PGSTAT 20. E_{out} und I_{out} werden differentiell zwischen den Kanälen ACH0 und ACH1 resp. ACH2 und ACH3 gemessen. Die Daten werden in einem Igor Text-Format in Analogie zu <SaveDataForIgorPro.vi> gespeichert; die Spektren werden auf dem Uvikon 942 unter dem entsprechenden Namen abgespeichert. Für die automatische Inkrementierung werden die Dateinamen auf 5 Zeichen beschnitten damit die Nummer im Format "nnn" angehängt werden kann.