

Technische Universität Ilmenau



Projektarbeit

Entwicklung einer Software für ein Ringlaser-Winkelmessgerät

Norman Krebs

August 2003

Die vorliegende Projektarbeit entstand in der Zeit von Mai 2003 bis Juli 2003 während meines Aufenthaltes an der Physikalisch-Technischen Bundesanstalt (PTB) in Braunschweig, im Fachbereich 5.2, AG Winkelmesstechnik.

Verantwortlicher Hochschullehrer: Univ.-Prof. Dr.-Ing. habil. Rainer Grünwald
Betreuer der Hochschule: Dr. Ing. Uwe Gerhardt
Betreuer der PTB: Dipl.-Ing. Andreas Just

Ich bedanke mich bei allen Mitarbeitern der PTB. Insbesondere bei meinem Betreuer Dipl. Ing. Andreas Just, der mich durch die Vermittlung seiner unzähligen praktischen Erfahrungen zu vielen Ideen anregte, sowie Dr. Ing. Reinhard Probst, dessen Kompetenz und Erfahrung ich hoch schätze.

Inhaltsverzeichnis

1	Erläuterungen	3
1.1	Die PTB	3
1.2	Das Projekt	4
2	Grundlagen	6
2.1	Ringlaser	6
2.2	Der Sagnac-Effekt	7
2.3	Fehler des Ringlasers	9
3	Der optische Drehgeber	10
4	Das Messgerät	11
5	Der Messzyklus	14
6	Die Programme	16
6.1	Hardwarezugriffe	16
6.2	Die Messkarte	17
7	Das Programm DevLoad2	18
7.1	<i>RL_IWM2</i> und <i>devload</i>	18
7.2	<i>devload2</i>	18
7.3	Die Datensatzkorrektur	19
7.3.1	Die interne Datendarstellung	21
7.4	Korrektur der Teilungsfehler des Drehgebers	26
7.5	Die Driftkorrektur	27
7.6	Die Programmoberfläche	28
8	Labwindows/CVI	32
8.1	Umgebung und Sprachkonzept	32
8.2	Labwindows - Beispiele	34
9	Zusammenfassung und Ausblick	36

1 Erläuterungen

1.1 Die PTB

Die Physikalisch-Technische Bundesanstalt (PTB) ist das nationale Metrologieinstitut Deutschlands und sie hat folgende wissenschaftlich-technischen Dienstleistungsaufgaben:

- Einheitlichkeit des Messens
- Sicherheit und Schutz der Bürger

Die Tätigkeit der PTB gliedert sich in vier Hauptfelder:

1. Grundlagen der Metrologie (Wissenschaft des Messens)
2. Metrologie für die Wissenschaft
3. Metrologie für die Gesellschaft
4. Internationale Angelegenheiten



Abbildung 1: Die Tätigkeitsfelder der PTB

1.2 Das Projekt

Die PTB hat im Jahre 1996 in Zusammenarbeit mit der DLR¹ Braunschweig und der ETU² St. Petersburg/Russland ein Projekt gestartet, das den Einsatz eines Ringlasers in Kombination mit einem optischen Drehgeber in der Präzisionswinkelmessstechnik zum Gegenstand hat. Zu diesem Zweck wurde ein Ringlaser von der Technischen Universität in St. Petersburg sowie ein Drehgeber der Firma Heidenhain erworben und das vorliegende Messgerät aufgebaut.[1][2]

In den Jahren 2002 und 2003 wurde in St. Petersburg eine neue Zählerlektronik sowie in der PTB eine neue Elektronik für die Signalformung entwickelt, deren wesentlicher Vorteil darin besteht, dass die Ringlaserimpulse nicht nur einfach gezählt, sondern durch eine Interpolation ca. auf einen neunzigstel Impuls genau aufgelöst werden können. Ziel der Projektarbeit ist es, ein Programm zu entwickeln, dass diese Genauigkeitserhöhung seitens der Elektronik auch in eine Erhöhung der Messgenauigkeit umsetzt.

Die derzeit für die Navigation verwendeten Ringlaser-Messgeräte haben nur eine Genauigkeit, die sich im Winkelminutenbereich bewegt. Die Messgenauigkeit ist ausserdem abhängig von der Messzeit. Ringlaser haben folgende Vorteile gegenüber anderen Messmethoden:

1. Der Winkel wird nicht gegen einen Bezugspunkt gemessen, sondern inertial, d.h. unabhängig von einem festen Bezugssystem. Im Vergleich zur Winkelmessung mit Drehgebern hat dies den Vorteil, dass Fehler durch mechanische Ankopplung an eine drehende Welle vermieden werden.
2. Die Messung ist drehpunktunabhängig. Daraus resultieren eine grosse Freiheit bei der Wahl des Messortes sowie der Wegfall von Exzentrizitätsfehlern (Schenkel- und Scheiteldeckungsfehler)
3. die Teilungsschritte des Ringlasers durch die Lichtwellenlänge als Referenz sehr klein und sehr gleichmässig.

Die Vorteile, die der Ringlaser gegenüber anderen Messprinzipien hat, ist aber praktisch nicht ohne weiteres in hohe Messgenauigkeit umzusetzen, weil die Stabilität des Ringlaserskalenfaktors begrenzt ist. Ziel des Projektes der Ringlaserprojektes der PTB ist es herauszufinden, ob die derzeit bestehende Genauigkeitsgrenze von ca. 1 Winkelsekunde beim Einsatz dieses Ringlasers überwiegend technischer Natur ist, und zum Beispiel durch den Einsatz geeigneter Messverfahren oder konstruktiver Änderungen wesentlich in Richtung grösserer Genauigkeit verschoben werden kann, oder ob es sich bereits um physikalische Grenzen des Ringlasers handelt, die die Genauigkeit begrenzen.

Der optische Drehgeber hat zwar ein geringeres Auflösungsvermögen als der Ringlaser aber er zeigt keinerlei Drift und nach jeweils einer Umdrehung ist ungeachtet aller Einzelteilungsfehler der Summenteilungsfehler gleich Null. Bei jeder Umdrehung kann somit der Ringlaser gegen den Vollwinkelbezug des Drehgebers kalibriert werden, während gleichzeitig umgekehrt die Teilungsfehler des Drehgebers gegen die genaue Teilung des Ringlasers kalibriert werden.

Durch die Kombination von Ringlaser und Drehgeber ergänzen sich daher die positiven Eigenschaften beider Systeme und das Leistungsvermögen des Hybridsystems ist höher

¹Deutsches Zentrum für Luft- und Raumfahrt

²Elektrotechnische Universität

als das beider Einzelsysteme. So soll auch die gegenseitige Kalibrierung beider Teilsysteme ein integraler Bestandteil des zu entwickelnden Programmes sein.

Insbesondere die inertielle Winkelmessung und die vergleichsweise kompakte Bauform des vorliegenden Messgerätes könnten es erlauben, ein transportables Winkelmessgerät aufzubauen, dessen Einsatz bei hoher Genauigkeit schnell erfolgen kann, da die bei anderen Winkelmessverfahren nötigen Justagearbeiten um die Drehachsen von Messgerät und zu Messobjekt in Übereinstimmung zu bringen, weitgehend entfallen. Sollte es sich erweisen, dass trotz der getroffenen Massnahmen keine wesentliche Genauigkeitserhöhung eintritt, dann kann eine Verbesserung des Messverhaltens nur noch durch eine Vergrößerung des Ringlasers erzielt werden, was aber den Vorteil der leichten Transportierbarkeit zunichte macht.

2 Grundlagen

2.1 Ringlaser

Gas- und Feststofflaser bestehen im allgemeinen aus einem aktiven Medium, einer Quelle zum optischen „Pumpen“ des Materials sowie aus einem optischen Resonator - zwei parallelen Spiegeln, zwischen denen sich eine stehende Welle ausbildet. Im Rahmen der optischen Gesetze sind auch ringförmig geschlossene Resonatorformen denkbar, die aus mehr als zwei Spiegeln oder Prismen bestehen, wie in Abbildung 2 dargestellt. Um ein axiales oder radiales Auswandern des Laserstrahls zu verhindern, werden häufig gekrümmte Spiegel verwendet. Im vorliegenden Ringlaser sind alle Prismen eben, was aber eine hohe Präzision bei deren Ausrichtung und Justage erfordert. An den beiden Austrittsflächen des aktiven Mediums befinden sich Brewster-Fenster, bei deren Durchtritt das Licht linear polarisiert wird.

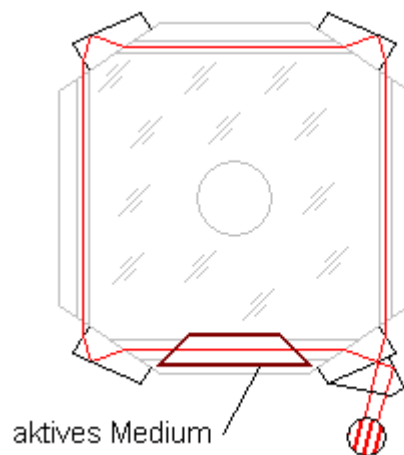


Abbildung 2: Aufbau des vorliegenden Interferometers mit vier Prismen

Durch Nutzung des nach seinem Entdecker benannten Sagnac-Effektes lassen sich solche Laser für die Drehraten-, Winkel- und Lagemessung verwenden. Sie werden bereits seit ca. vierzig Jahren in Land-, See- und Luftfahrzeugen parallel zu anderen Navigationsinstrumenten wie mechanischen Gyroskopen für die autonome Navigation eingesetzt. Die mechanischen Gyroskope werden zunehmend von Ringlasern abgelöst, da diese mechanisch weit weniger anfällig sind. Durch das Aufkommen der Satellitennavigation (GPS ³) hat aber die autonome Navigation etwas an Bedeutung verloren.

³Global Positioning System

2.2 Der Sagnac-Effekt

Abbildung 3 zeigt in schematisierter Form das dem Sagnac-Effekt zugrunde liegende Wirkungsprinzip. Wird ein Lichtstrahl aufgespalten und beide Teilstrahlen jeweils entgegengesetzt im Kreis herum geschickt und dann detektiert, so legen beide dabei einen bestimmten Weg zurück und benötigen eine bestimmte Zeit. Rotiert die Konstruktion, so legt der Detektor, während die Strahlen umlaufen auch einen Weg zurück und läuft so vor dem in Drehrichtung umlaufenden Strahl davon, dessen zurückzulegender Weg vom Emitter bzw. Strahleintrittspunkt in die Konstruktion bis zum Detektor länger wird. Der entgegengesetzt zur Drehrichtung umlaufende Strahl legt einen etwas kürzeren Weg zurück. Betrachtet man die vorauslaufende Welle, so entsteht sie zunächst in dem aktiven Me-

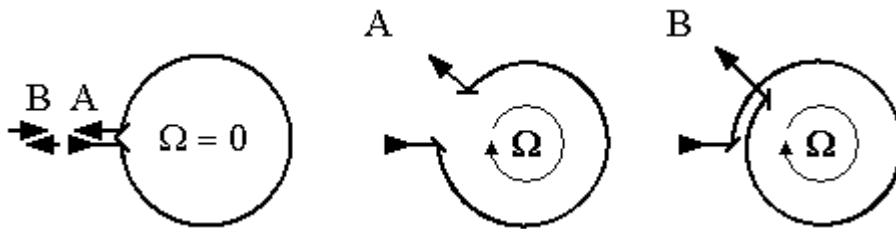


Abbildung 3: optische Weglängen der in Bezug auf die Drehrichtung (Ω) zurück-(A) und vorauslaufenden Teilstrahlen (B) (nach [3])

dium, läuft dann durch die vier Spiegel und kommt wieder im aktiven Medium an. Hier entsteht gerade eine neue Welle, die in der gleichen Richtung läuft und mit der alten Welle kohärent sein muss, damit die Wellen gemeinsam bestand haben und sich nicht auslöschen. Rotiert nun der Resonator, so hat sich das aktive Medium beim Eintreffen der Welle bereits um einen kleinen Weg verschoben und die entstehende Welle ist nicht mehr deckungsgleich zur bereits vorhandenen. Das Licht dieser Wellenlänge wird also nach einigen weiteren Umläufen verschwunden sein. Stattdessen bildet sich eine Welle mit einer grösseren Wellenlänge aus. Der entgegen der Drehrichtung laufende Strahl wird demselben Effekt im umgekehrten Sinne unterworfen und seine Wellenlänge verschiebt sich in kürzere Bereiche. Dieser Effekt wird nach seinem Entdecker, der ihn bereits im Jahre 1913 erfolgreich demonstrierte, Sagnac-Effekt genannt.

Nach (1) ergibt sich als Schwebungsfrequenz der ausgekoppelten und zur Interferenz gebrachten Teilstrahlen mit der von den umlaufenden Strahlen eingeschlossenen Fläche A , der Winkelgeschwindigkeit Ω der Lichtwellenlänge λ und der Umlauflänge der Lichtstrahlen L zu

$$\Delta\nu = \frac{A\Omega 4}{L\lambda} \quad (1)$$

Der Sagnac-Effekt zeigt, dass

$$\Delta\nu \sim \Omega \quad (2)$$

Der Term

$$\frac{4A}{\lambda L} = K \quad (3)$$

wird Skalenfaktor genannt. In ihm sind die relevanten, gerätespezifischen Größen zusammengefasst. Zum Nachweis der Schwebungsfrequenz $\Delta\nu$ werden geringe Anteile der beiden Teilstrahlen über eines der Prismen ausgekoppelt und zur Interferenz gebracht (Abb. 2). Stehen die Strahlen in einem kleinen Winkel zueinander, dann entstehen durchlaufende Interferenzstreifen mit sinoidaler Helligkeitsverteilung, deren Abstand eine Zählung und die Abtastung mit zwei leicht versetzten Fotodetektoren und auch eine Richtungsbestimmung ermöglicht.

Die Schwebungsfrequenz $\Delta\nu = f$, mit der die Streifen am Detektor vorbeilaufen, beträgt nach (1) und (3)

$$f = K * \Omega \quad (4)$$

Die Frequenz f ist also ein Maß für die Winkelgeschwindigkeit Ω , woraus durch zeitliche Integration (Zählen der Frequenz) folgt, dass das Zählresultat N ein Maß für den Drehwinkel Φ ist.

$$N = \underbrace{\frac{4A}{\lambda L}}_K \Phi \quad (5)$$

In diesem Sinne kann man den Ringlaser als zählendes Laserinterferometer für die Drehbewegung auffassen, in Analogie zum zählenden Laserinterferometer für die Linearbewegung. Beispielsweise beträgt nach (3) bei einem quadratischen Resonator mit der Kantenlänge $0.1m$ bei einer Grundrotation von $240 \text{ }^\circ/s$ und einer Lichtwellenlänge von ca. 633 nm der Skalenfaktor $K = 661736 \pm 11.5Hz$. Wobei die Erdrotation mit ca. 11.5 Impulsen pro Sekunde positiv eingeht, wenn sich das Gerät entgegen dem Uhrzeigersinn dreht, ansonsten negativ.

Im Gegensatz zu vielen anderen Messverfahren, bei denen ein rein analoges Messsignal zunächst in ein zähl- und verarbeitbares Signal umgewandelt werden muss, liefert das Sagnac-Interferometer, wie alle Laser-Interferometer, bereits ein zählbares Signal, welches durch geeignetes Anbringen der Photodetektoren auch zur Detektierung von Vor- und Rücklauf verwendet werden kann. Auch ist durch die Verwendung der Lichtwellenlänge als Längenreferenz dem Messprinzip hinsichtlich der möglichen Genauigkeit ein grosses Potential eigen.

2.3 Fehler des Ringlasers

Die Fehler des Ringlasers lassen sich wie folgt unterteilen:

Lock-In-Effekt ein durch ungewollte Rückstreuung an den Spiegeln verursachter Koppelungseffekt beider umlaufender Wellen, der bei kleinen Drehraten zur Einstellung der gleichen Frequenz und zum Ausbleiben einer Schwebung führt. Abhilfe schafft nur das Aufbringen einer Grunddrehrate, die dann als Konstante vom Messwert abgezogen werden kann, oder die Vergrößerung des Resonators

Änderung des Skalenfaktors - Drift, das Verhältnis von Drehwinkel und Schwebungsfrequenz bzw. gezählten Interferenzimpulsen schwankt und somit wird der falsche Winkel gemessen. Durch Kalibrierung des Lasers mit dem optischen Drehgeber kann diesem Effekt zu einem grossen Teil entgegengewirkt werden.

Änderung der Nullpunktlage

Random Walk Die bestimmenden Größen des Ringlasers unterliegen stochastischen Einflüssen, die die Funktion stören und durch das zählende Messprinzip aufintegriert werden. Der daraus entstehende messzeitabhängige sog. „Random Walk“ ist, weil zufällig, nicht bestimmbar und muss als empirisch ermittelter Random-Walk-Koeffizient mit die Messunsicherheit aufgenommen werden.

Verkippung von Rotationsachse und Ringlaserachse diese Fehler sind systematisch und können bei genauer bestimmung der Verkippung berechnet werden. Die geografische Breite und ihr Einfluss auf den Einfluss der Erdrotation zählt beispielsweise dazu. Durch Änderung des Drehsinns der Grunddrehrate des Ringlasers kann die Verkippung ermittelt werden.

Drehzahlschwankungen nichtsystematische Drehzahlschwankungen der Grunddrehrate zwischen Kalibrier- und Messphase, die dazu führen, dass der Einfluss der Erdrehrate nicht mehr als Konstante betrachtet werden kann.

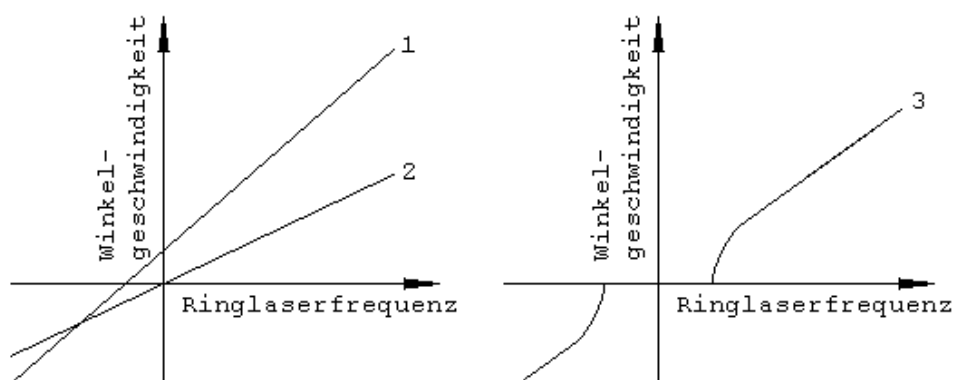


Abbildung 4: Fehler des Ringlasers: 1 - Nullpunktänderung, 2 - Skalenfaktoränderung, 3 - Lock-In-Effekt

3 Der optische Drehgeber

Der eingesetzte optische Drehgeber ist ein Gerätetyp ERO 725 der Firma Heidenhain. Er besteht aus einer Glasscheibe mit einer Radialgitterteilung mit 36000 Teilstrichen und einem Durchmesser von ca. 150 mm. Die Abtastung erfolgt optisch im Durchlichtprinzip über zwei um 180° versetzte, ebenfalls mit einer Radialgitterteilung versehene Abtastköpfe. Dies ermöglicht den teilweisen Ausgleich von Exzentrizitätsfehlern.

Ausserdem verfügt die Scheibe über eine Nullmarke die es ermöglicht, die Messung jeweils an derselben Stelle zu beginnen.

Theoretisch können die ERO-Signale mit dem Faktor 100 interpoliert werden, um die Auflösung zu erhöhen, aber von dieser Möglichkeit wird hier kein Gebrauch gemacht, im Gegenteil: um die Datenrate zu verringern, wird die Drehgeberfrequenz ja mit einem Teilungsfaktor versehen. So ist das Auflösungsvermögen des Drehgebers:

$$1296000''/36000 = 36''$$

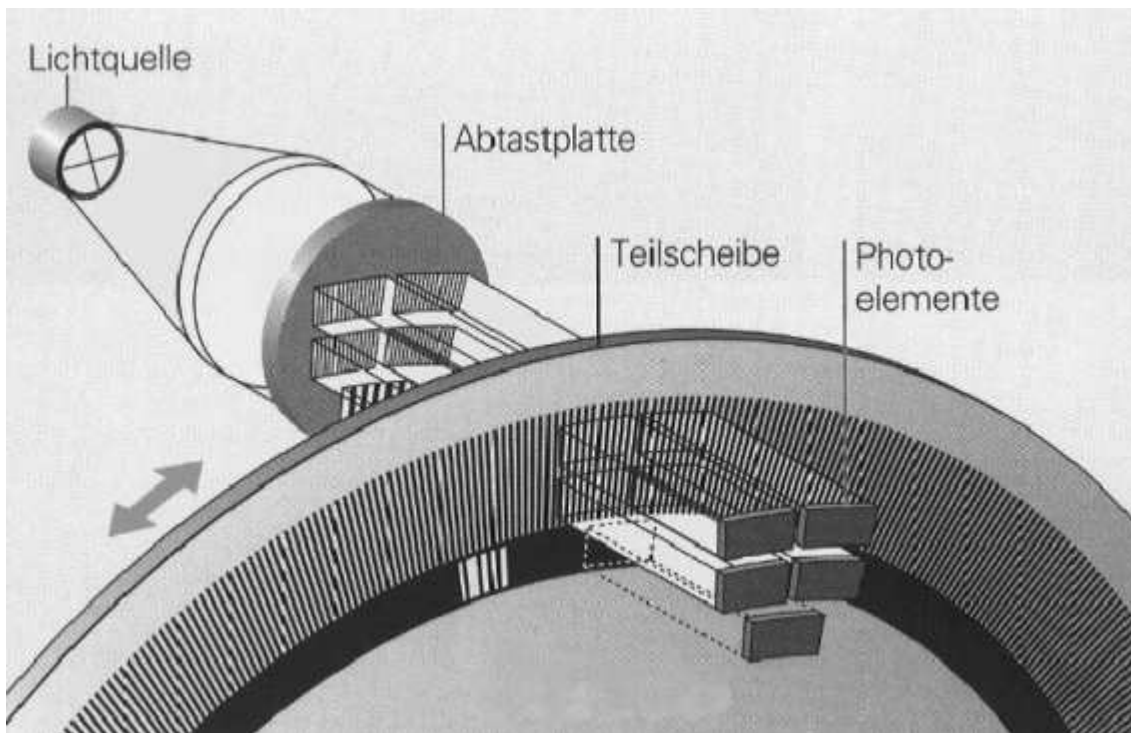


Abbildung 5: Ausschnitt einer Teilkreisscheibe und Schema eines Abtastkopfes sowie einem Nullimpulsgeber (360° – Referenz)

(Quelle: Fa. Heidenhain)

4 Das Messgerät

In das vorliegende Messgerät wurden zwei völlig unterschiedliche Messprinzipien integriert, um die Vorteile beider zu kombinieren. Im oberen Teil des Gerätes befindet sich der Ringlaser, der am oberen Ende einer senkrecht eingebauten, rotierenden Welle sitzt. Die Welle ist in ihrer Mitte mit einem externen Druckluftanschluss luftgelagert. Zwischen Ringlaser und Luftlager befindet sich ein tangentialer Riemenantrieb, der die für die Vermeidung des Lock-In-Effektes notwendige Grundrotation aufbringt. Unter dem Luftlager befindet sich auf der Welle der optische Drehgeber. Ringlaser und Teilscheibe rotieren also stets mit der selben Drehzahl. Unter der Teilscheibe werden die elektrischen Anschlüsse über Schleifkontakte abgeführt (hier nicht dargestellt).

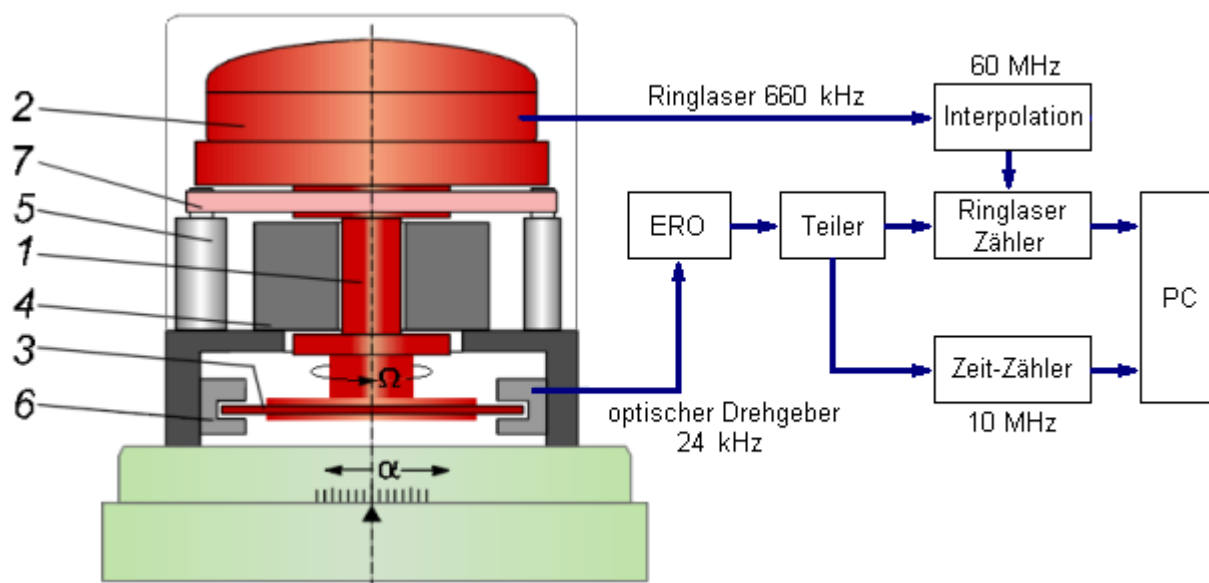


Abbildung 6: Schema und Blockschaltbild des Messgerätes

- 1 - Spindel, 2 - Ringlaser, 3 - Teilkreisscheibe des Drehgebers
 4 - Stator des Luftlagers, 5 - Motor f. Spindelantrieb,
 6 - Abtastköpfe des Drehgebers (2x), 7 - tangentialer Riemen
 (Quelle: A.Just, PTB 2001)

Laut Gleichung (1) berechnet sich die Ausgabefrequenz des Ringlasers zu $\approx 660\text{kHz}$, wenn der Laser mit einer Grundwinkelgeschwindigkeit $\Omega = 240^\circ \cdot \text{s}^{-1}$ rotiert. Die Frequenz des Drehgebers (24kHz) wird mit einem einstellbaren Teiler (z.B. 2..2000) reduziert und zur Triggerung der Ringlasersignale herangezogen. Die resultierende Ausgabefrequenz z.B. mit dem Teiler 100 beträgt 240 Hz. Der Teiler kann durch eine extern ausgeführte Schaltung zugeführt, oder, wie hier geschehen, softwareseitig vorgegeben werden. Die Länge eines Drehgeberimpulses ist kein ganzzahliges Vielfaches der Ringlaserimpulsweite, so dass keine ganzzahlige Anzahl von Ringlaserimpulsen innerhalb eines Drehgeberimpulses liegt. Wenn der Triggerimpuls kommt, dann wird ein Ringlaserimpuls „zerschnitten“. Das Verhältnis zwischen der Ringlaserimpulslänge und dem im Drehgeberimpuls liegenden Anteil des letzten Ringlaserimpulses wird in der Anzahl von Takten des 60MHz Impulsgebers angegeben. ($\frac{B}{A}$ in Abbildung 7)

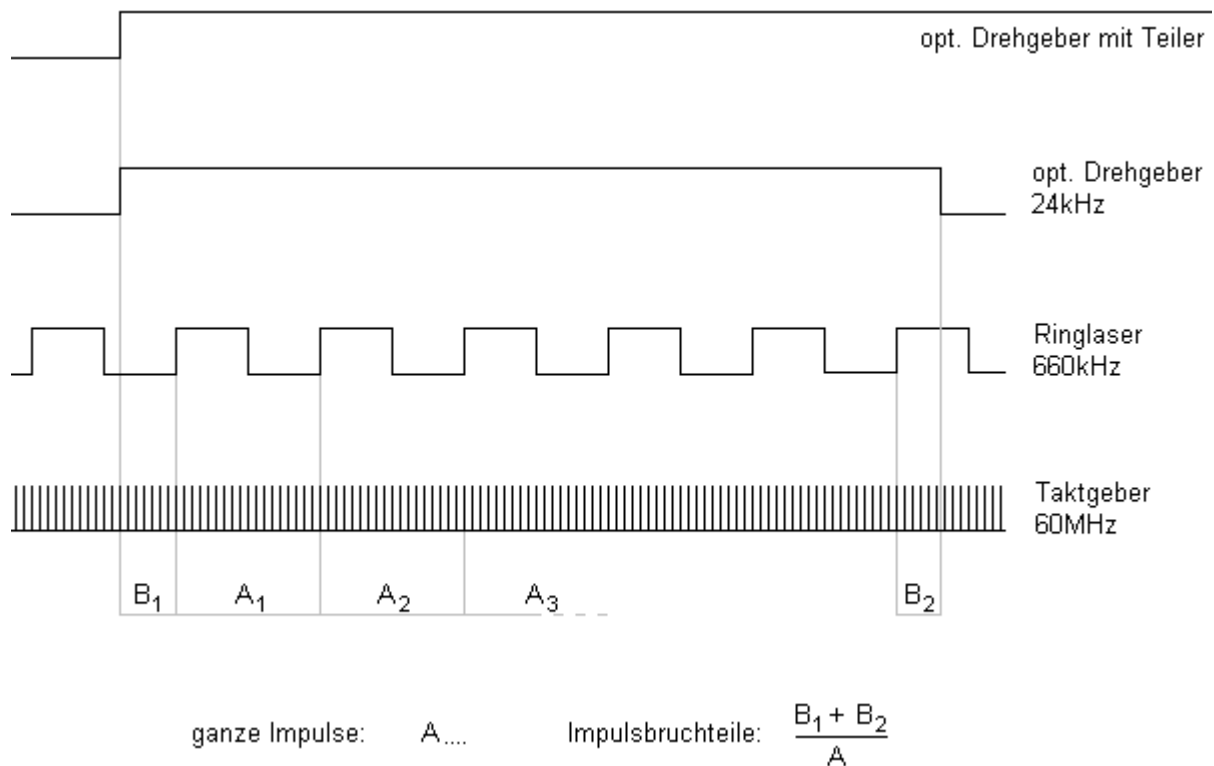


Abbildung 7: Interpolation, A - ganze, B - Bruchteile von Taktgeberimpulsen pro Ringlaserimpuls (nicht masstäblich)

Durch den Teilungsfaktor wird der Vollwinkel in Sektoren eingeteilt, wobei am Ende jedes Sektors die Zählerstände von Ringlaserimpuls- und Drehgeberimpulszählern gelesen werden. Abbildung 8 zeigt die Teilscheibe mit 36 Sektoren (Teiler = 1000). Am Ende jedes Sektors wird ein Messwert ausgelesen. Die ERO-Zählerstände von einem Sektor zum darauffolgenden unterscheiden sich jeweils um exakt die Grösse des angegebenen Teilers. Anhand dieser Information kann festgestellt werden, um welchen Sektor es sich jeweils handelt. In diesem Beispiel würden bei einer Messung über zehn Umdrehungen 10×36 Werte ausgelesen werden. Wenn der ERO-Startwert (Sektor 0) 354 betrüge, dann müssten die darauffolgenden ERO-Werte jeweils 1345 (Sektor 1), 2345 (Sektor 2), 3345 (..) 4345 usw. betragen und die Messung würde mit dem ERO-Wert 36345 enden. Sobald die Differenz zweier aufeinanderfolgender ERO-Werte vom Teilerwert abweicht, ist dies ein sicheres Indiz dafür, dass ein oder mehrere Messwerte verloren wurden.

Bei 1.5 s pro Umdrehung werden die Ringlaserimpulse mit ca. 660 kHz ausgegeben und es ergeben sich ca. 990000 Ringlaserimpulse. Pro Sektor ergibt sich ein Sollwert von ca. 28000 Impulsen.

Exakte Angaben sind hier unsinnig, da sich der jeweilige Wert durch die Skalenfaktordrift ändert. Durch die Registrierung der Laserimpulse pro Sektor können leichte Änderungen über mehrere Umdrehungen gemittelt werden. Einzelteilungsfehler der Teilkreisscheibe schlagen sich als unterschiedlich grosse Sektormittelwerte nieder.

Die Wahl des Teilers wird von zwei Faktoren bestimmt: den Anforderungen an die Messgeschwindigkeit, bzw. Dynamik und dem im Rechner vorhandenen Arbeitsspeicher. Je kleiner der Teiler ist, um so besser können dynamische Effekte, wie Schwingungen und

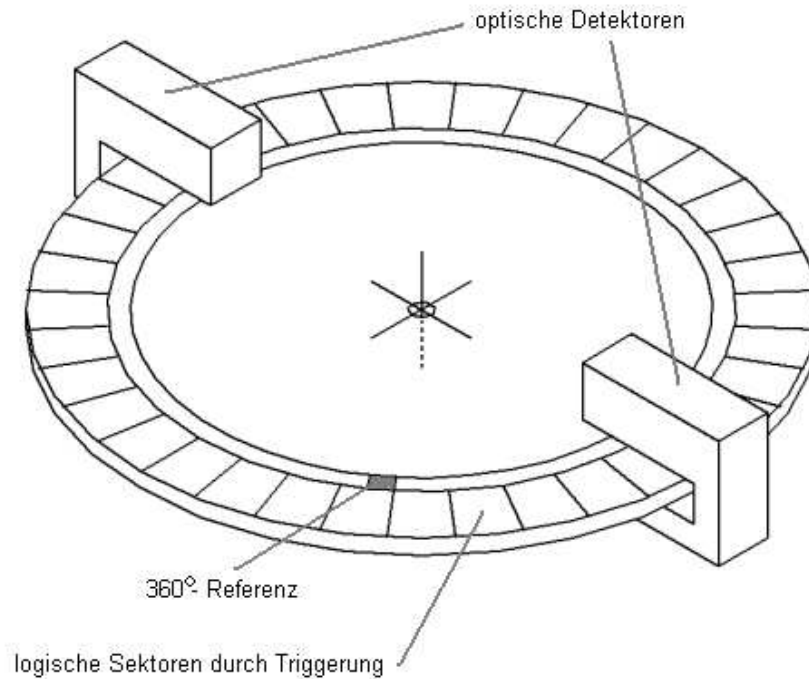


Abbildung 8: Teilkreisscheibe mit logischen Sektoren

Störimpulse, die auf das Messsystem wirken, aufgelöst werden. Gleichzeitig wachsen aber auch die Datenmenge und die betriebssystembedingten Messdatenausfälle reziprok zum Teiler, so dass das praktische Maximum bei einer Sektoranzahl von ca. 1000, was einem Teiler von 36 entspricht, liegen dürfte.

Wie in Abbildung 9 dargestellt, besteht das Gestell des Gerätes aus einer massiven Stahlplatte, die auf drei (damit ein Kippen ausgeschlossen ist) ebenfalls massiven Stahlfüßen ruht. An den Ecken der Platte sitzen vier Aluminiumleisten, die zusammen mit vier seitig angeschraubten Aluminiumblechen und einem aufzusetzenden Abdeckblech das Gehäuse bilden. Die Leisten reichen nicht bis auf die Standfläche hinunter, so dass die Konstruktion nur auf den drei Füßen und nicht auf den Leisten steht. Die massive Ausführung des Grundgestells ist notwendig, damit keine Drehschwingungen auftreten, die dann als zufällige Fehler in das Messergebnis eingehen würden. Auf die Grundplatte sind das Luftlager, die Antriebseinheit und die Detektoren des ERO montiert. Die Kabel des Ringlagers werden durch die hohlgebohrte Welle nach unten geführt und unter dem ERO über Schleifringe abgenommen.

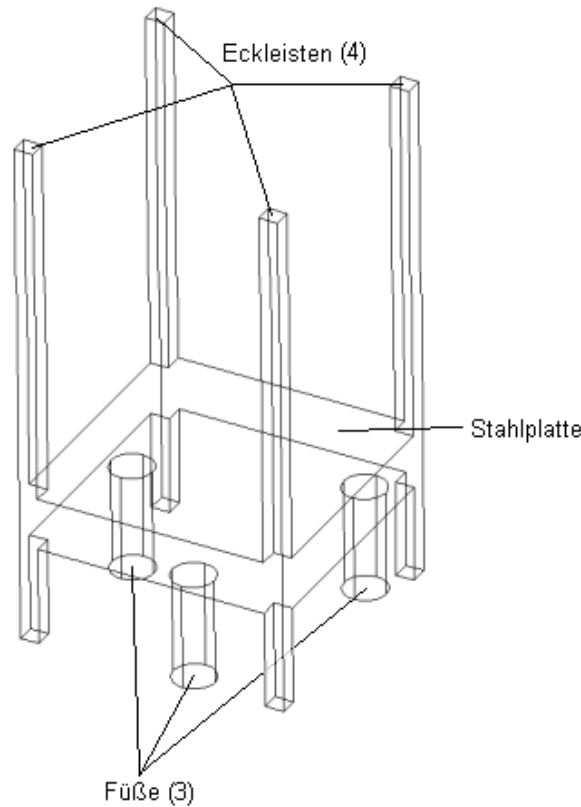


Abbildung 9: das Gestell des Gerätes (schematisch)

5 Der Messzyklus

Rotiert das System nur mit der Grunddrehrate Ω_0 (die externe Rotation $\Omega_E = 0$), so ist das Erreichen des Vollwinkels 2π mit dem Drehgeber gut feststellbar. Diesen Winkel sollte im Idealfall auch der Ringlaser liefern, er tut es im realen Betrieb aber nicht. Wirkt keine äussere Rotation auf das System, dann kann nach jeweils einer Umdrehung der Laser kalibriert und die Drift festgestellt werden.

Durch den Vergleich des Skalenfaktors vor und nach einer Messung kann die Drift des Skalenfaktors und somit ein Korrekturwert berechnet werden. Für die Feststellung der Drift muss natürlich sichergestellt sein, dass das Gerät keiner äusseren Rotation unterworfen ist. Die Erdrotation die stets mit in das Ergebnis eingeht, ist ein zeitabhängiger Einfluss der sehr gleichmässig ist. Da die Grundrotation des Ringlasers aber durch leichte Drehzahlschwankungen variiert, ist auch der Einfluss der Erddrehrate auf den Skalenfaktor variabel. Durch eine parallele Zeitmessung kann der quantitative Einfluss der Erdrotation bestimmt, und die Messergebnisse entsprechend korrigiert werden, auch wenn die Grunddrehrate nicht immer konstant ist.

Um die Drift innerhalb der Messphase bestimmen zu können, besteht ein Messzyklus aus den drei Phasen, die in den Abbildungen 10 und 11 dargestellt sind.

Die Gesamtdrehrate ergibt sich wie folgt

$$\Omega = \Omega_0 + \Omega_E \text{ bzw. } \Omega_E = \Omega - \Omega_0 \quad (6)$$

- Kalibrieren ($\Omega_E = 0$),

- Messen ($\Omega_E \neq 0$)
- Rekalibrieren ($\Omega_E = 0$)

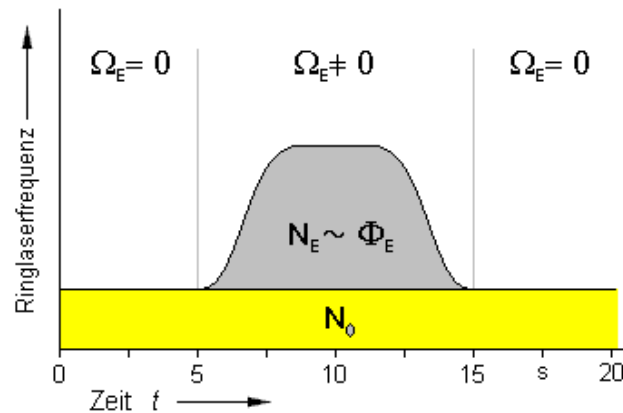


Abbildung 10: Prinzip des Messzyklus ohne Skalendrift

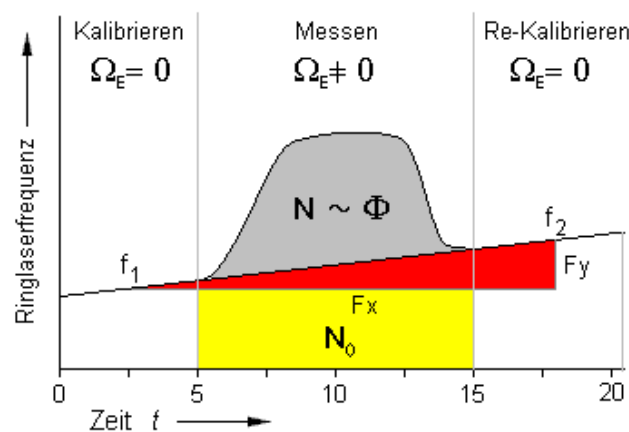


Abbildung 11: Prinzip des Messzyklus mit (als linear angenommener) Skalendrift

Die gelb dargestellten Flächen entsprechen der stationären Rotation und müssen ebenso wie der rot dargestellte Einfluss der Drift vom Ergebnis abgezogen werden, um die externe Rotation bzw. Winkeländerung zu erhalten. Unter der Annahme, dass die Drift des Skalenfaktors innerhalb des Messzyklus linear verläuft, berechnet sich der Driftfaktor aus $\frac{F_y}{F_x}$ und kann herangezogen werden, um die Messwerte zu korrigieren. Insbesondere kurz nach der Inbetriebnahme des Lasers ist die Drift besonders hoch und nichtlinear und durch die Erwärmung des Gerätes kommen häufig Modensprünge des Lasers vor. Nach einer Erwärmungszeit von ca. zwei Stunden hat sich das System stabilisiert. Der in den Abbildungen 10 und 11 dargestellte Zusammenhang $N \sim \Phi$ stellt den Zusammenhang zwischen dem Zählerwert der Ringlaserimpulse und dem Winkel, um den das gesamte System gedreht wurde, dar. Die Fläche unter der Kurve stellt das Integral über die Winkelgeschwindigkeit und somit den Winkel dar, gemessen als Zählerwert N .

6 Die Programme

Zweck der Projektarbeit war es, ein Programm für die Messwertaufnahme und -auswertung zu entwickeln. Damit verbunden war zunächst die Analyse der vorliegenden Programme, um die Funktionsweise der Zählerkarte und der Datenauswertung zu erfahren sowie die Arbeit mit den Entwicklungssystemen LabWindows/CVI und Microsoft Visual C++.

6.1 Hardwarezugriffe

Wenn ein Anwendungsprogramm auf im Rechner verbaute Hardware zugreifen will, dann bieten sich prinzipiell zwei Möglichkeiten:

direkter Hardwarezugriff , wenn die Maschinenadresse, unter der die jeweilige Hardware ansprechbar ist bekannt ist dann kann, unter genauer Kenntnis des Verhaltens des entsprechenden Gerätes, die Hardware direkt angesprochen werden. Dies wird aber in modernen Multitasking-Betriebssystem wie Windows NT, 2000, XP, Linux, Unix usw. vermieden, da durch den gleichzeitigen Zugriff mehrerer Anwendungsprogramme auf dieselbe Hardware, bzw. mögliche aber unzulässige Operationen das System sehr leicht zum Absturz zu bringen ist.

indirekter Hardwarezugriff , um direkte Hardwarezugriffe zu vermeiden, wurden Mechanismen geschaffen, die es Anwendungsprogrammen erlauben, in einem sicheren Kontext und unter der Obhut des Betriebssystems auf die Hardware zuzugreifen. Zunächst wurden, um die Anwendungsprogramme und das Betriebssystem voneinander zu schützen, getrennte, virtuelle Adressräume geschaffen. Selbst bei Kenntnis der Hardwareadresse eines Gerätes sind nun keine direkten Zugriffe mehr möglich, da die Speicheradresse, von einem Anwendungsprogramm aufgerufen, nicht zu der adressierten Hardwareadresse führt, sondern an eine beliebige Stelle im RAM . Ein direkter Hardwarezugriff seitens des Anwendungsprogrammes funktioniert so nicht mehr und führt, infolge des Zugriffes auf geschützte Speicherbereiche, zu einer sog. Schutzverletzung und zum Abbruch des Programms. Das Betriebssystem beinhaltet modulare, hardwarespezifische Programme, die keine eigenen Threads haben und im Maschinenadressraum laufen, sog. Treiber, die hard- und softwareseitig Schnittstellen bieten, um Daten zu übertragen. Auf die softwareschnittstelle setzt meist noch eine weitere Softwareschicht auf, ein sogenanntes API. ⁴ Das Konzept aus getrennten Adressräumen und Treibern ist sehr sicher und restriktiv. Um die Kompatibilität der Betriebssysteme Windows 95, 98 und ME mit dem alten Betriebssystem DOS zu gewährleisten, dessen Anwendungsprogramme oft fast nur direkte Hardwarezugriffe verwendeten, wurde das Sicherheitskonzept gelockert, so dass sich nun alle Anwendungen die ersten zwei Gigabyte des Arbeitsspeichers teilen und hier direkt auf den Maschinenspeicher zugreifen können. Dies beeinträchtigt zwar die Stabilität des Systems, aber die Programmentwicklung wird erheblich vereinfacht, da die Entwicklung eines Hardwaretreibers den Rahmen kleiner Softwareprojekte oft sprengt.

⁴Application Programming Interface

6.2 Die Messkarte

Die Messkarte ist eine PC-Einsteckkarte für den ISA-Bus und sie bietet mehrere Zähler und drei relevante Register, die an die Position 0x300 (und folgend) des Maschinenspeichers eingeblendet werden:

- CSR** das Statusregister, in dieses Register wird der Arbeitsmodus der Karte eingetragen: externe oder, wie hier, vorgegebene Teiler
- NI** liefert Messwerte die mit einem einstellbaren Teiler getriggert werden. Der Teilerwert wird sowohl in dieses Register geschrieben, als auch Daten (Zählerstände) ausgelesen.
- RQ** -nicht verwendet-

Alle Register können sowohl gelesen als auch geschrieben werden.

Die Datenstruktur **va**, zeigt die Daten, die von der Karte geliefert werden:

```
typedef struct {
    DWORD dwLG;           // Ringlaser, ganze Impulse
    DWORD dwLGo;         // Ringlaser, Bruchteile
    DWORD dwODU1;        // optischer Drehgeber, Detektor 1, ganze Impulse
    DWORD dwODU1o;       // optischer Drehgeber, Detektor 1, Bruchteile
    DWORD dwODU2;        // optischer Drehgeber, Detektor 2, ganze Impulse
    DWORD dwODU2o;       // optischer Drehgeber, Detektor 2, Bruchteile
    DWORD dwTIM;         // Zeitgeber der Messkarte, [Mikrosekunden]
} va;
```

Die gebrochenen Impulse werden aus der Interpolation mit dem 60MHz-Schwinger gewonnen.

Sie besteht somit aus sieben Doppelwörtern á 4 Bytes, bzw. vierzehn Wörtern á 2 Bytes. Die Assembler-Routine, mit der die Messdaten aus dem Register NI ausgelesen werden:

```
cli           keine Interrupts mehr zulassen
mov edi, va  Adresse der Datenstruktur in EDI laden
mov edx, NI  Adresse des NI-Registers in EDX laden
mov ecx, 14  Anzahl der Wörter in ECX laden
rep insw    Wort auslesen und wiederholen, bis ECX=0
sti         Interrupts zulassen
```

Zunächst wird gewartet, bis Daten eingehen. Dazu wird regelmässig das Register CSR überprüft, in dem dann ein entsprechender Wert steht. Dann wird ein Datensatz gelesen. Die sieben Doppelwörter werden wortweise (1 Wort = 2 Bytes) hintereinander aus dem Register NI bzw. RQ der Karte ausgelesen.

7 Das Programm DevLoad2

7.1 *RL_IWM2* und *devload*

Als Ausgangspunkt für die Arbeit standen zwei Programme zur Verfügung:

- das Programm „devload“, ein Win32-Programm, das zusammen mit der Messkarte geliefert worden war und die Messdaten lediglich aufnimmt und in roher Form in eine Datei ablegt. Es wurde mit Microsoft-Visual-C++ entwickelt.
- das Programm „RL_IWM2“, ein DOS-Programm, welches von meinem Betreuer, Herrn Just, zusammen mit Herrn Schulz von der PTB, entwickelt worden ist. Es richtet sich weitgehend nach dem ANSI-C Standard und wurde mit einem Watcom-C-Compiler übersetzt.

Beide Programme liegen im Quelltext vor, wobei „devload“ zeigt, wie die aktuelle Hardware, also die Messkarte, anzusprechen ist. „RL_IWM2“ ist noch für eine ältere Messelektronik konzipiert worden, so dass es nicht mehr funktioniert, es zeigt aber die Vorgehensweise zur Aufarbeitung der Messwerte, die Korrektur von Erdrotation und Ringlaserdrift.

Zunächst wurde der „devload“-Quelltext analysiert. Bei diesem Programm wird der Hardwarezugriff in eine dll.-Datei ausgelagert und geschieht in einer Inline-Assembler Prozedur, wie unter (6.2) gezeigt. Die Prozedur wartet zunächst auf ein Null-Indikator-Signal vom Optischen Drehgeber und liest dann eine vorgegebene Anzahl von Datensätzen ein, wobei vor dem Auslesen jedes Datensatzes auf eine Statusmeldung im CSR-Register gewartet wird. Dies muss im Pollingbetrieb erfolgen, da das Auslesen der Datensätze aufgrund der möglicherweise recht hohen Frequenz der Messdaten zeitkritisch sein kann. Wenn aus irgendwelchen Gründen keine Signale vom Ringlaser oder vom Drehgeber kommen sollten, dann bricht das Programm den Messprozess nach einer einstellbaren Zeit ab.

Ein gravierender Nachteil des Programmes „devload“ besteht darin, dass das Programm während des Einlesens der Messwerte komplett „hängt“ und maximale CPU-Last erzeugt, da auf Windows-Nachrichten während der Polling-Schleife nicht reagiert wird. Dies zeigt sich darin, dass die Programmoberfläche nicht neugezeichnet wird und Bedienelemente (Buttons, Checkboxes) nicht auf Eingaben reagieren. So kann eine Messung, einmal gestartet, nicht mehr abgebrochen werden, wobei zum Beispiel bei einer Messung über 40 Umdrehungen eine Messzeit von einer Minute vergehen würde, bevor das Programm wieder reagiert.

Auch fehlt eine Rückmeldung während der Messung bezüglich z.B. der Messwertanzahl und Standardabweichung. Die aufgenommenen Daten liegen schliesslich nur „roh“ vor und die Umrechnung in Winkelwerte muss extern erfolgen, das heisst, das Prinzip der initalen Winkelmessung ist nicht realisiert.

7.2 *devload2*

In Anlehnung an die bereits vorhandenen Messdatenaufnahmeprogramme entstand das Programm „devload2“. Es wurde mit der RAD⁵-Umgebung Labwindows/CVI von National Instruments entwickelt, welches die Programmierung in mehrererlei Hinsicht vereinfacht. Da Labwindows/CVI keine Assembler-Sequenzen zulässt, wurde das Auslesen der Daten sowie die Hardwarezugriffe in die .dll-Datei rddx2.dll ausgelagert, die mit Microsoft Visual C++ 6.0 entstand.

⁵Rapid Application Development

Um die Bedienbarkeit während der Messphase des Programms zu vermeiden, wurden rechenintensive Programmteile in separate Threads ausgelagert. So werden die für die Programmbedienung nötigen Aktionen nicht verdrängt. Ein weiterer Vorteil von Threads besteht darin, dass bei Multi-Prozessorsystemen jeder Thread auf einer CPU gerechnet werden kann. Um solche Rechner optimal zu nutzen, ist eine Programmierung mit mehreren Threads unumgänglich. Die Nachteile des Multithreading sind die unter Umständen sehr vielfältigen Sharing- und Timing-Probleme, die sich ergeben, wenn mehrere Threads gleichzeitig auf begrenzte Ressourcen zuzugreifen versuchen.

7.3 Die Datensatzkorrektur

Während des Auslesens der Daten kann anhand der Drehgeberzählwerte festgestellt werden, ob pro Umdrehung genau so viele Datensätze eingelesen wurden, wie logische Sektoren vorhanden sind.

Die Zuordnung von Datensätzen zu einem bestimmten Sektor, wie in Abbildung 16 dargestellt, kann über mehrere Umdrehungen nur dann erfolgen, wenn kein einziger Datensatz verloren geht. Während der Programmentwicklung trat aber dieses Problem auf. Insbesondere bei kleinen Teilern von 100 und darunter, also grosser Sektoranzahl und hoher Messwertausgabefrequenz, verlieren sich ein oder mehrfach pro Umdrehung einer bis zwanzig Messwerte oder auch noch mehr. Der Effekt nimmt umgekehrt zur Teilergrösse zu und tritt ab ca. einem Teiler von 1000 aufwärts nicht mehr auf. Um diesem Problem auf den Grund zu gehen, wurde ein kleines Kommandozeilenprogramm mit Delphi geschrieben, welches nichts weiter tut, als Messwerte auszulesen und auf einen Datensatzverlust hin zu prüfen. Auch hier trat das Problem auf, und nach einer kleinen Modifikation zur Sichtbarmachung des Effekts, auch bei dem vorhandenen Programm „devload“. Der Verdacht lag nahe, dass es sich um eine Störung handelt, die vom Betriebssystem eingebracht wird, da sie auch, wenn keine weiteren Anwenderprogramme laufen, auftritt. Ein weiteres, kleines DOS-Programm, das funktionell identisch mit dem Delphi-Programm ist, bestätigte den Verdacht, denn dieses Programm las auch bei einem Teiler von 2, das heisst: 18000 Sektoren und 12000 Messwerten pro Sekunde (knapp jede zehntausendstel Sekunde ein Datensatz), alle Werte fehlerfrei aus.

Windows ist, wie alle modernen PC-Betriebssysteme, nicht echtzeitfähig, das heisst, eine Reaktion auf ein Ereignis innerhalb von wenigen Millisekunden kann nicht gewährleistet werden. Multitasking- und Multithreadingbetriebssysteme verarbeiten stets mehrere Programmthreads quasiparallel. Da nur eine oder zwei CPUs vorhanden sind, können die Threads nur seriell gerechnet werden, so dass stets die überwiegende Mehrheit der rechenwilligen Threads „steht“. Wenn während der Messwertaufnahme ein weiterer Thread Rechenzeit verlangt, dann erhält er diese auch nach gewisser Zeit, so dass der Messdaten aufnehmende Thread gestoppt wird und Datensätze verlorengehen. Dies lässt sich weitgehend vermeiden, indem der zeitkritische Thread eine maximale Priorität erhält.

Unter Windows existieren für Threads sieben mögliche Prioritäten, die zusammen mit einer Prioritätsklasse und einem dynamisch veränderlichen Anteil die aktuelle Priorität des jeweiligen Threads ergeben, die der Scheduler seiner Entscheidung zugrundelegt, welcher Thread als nächster zur Verarbeitung gelangt. Ein normales Anwenderprogramm hat die Prioritätsklasse `NORMAL_PRIORITY_CLASS` und die Priorität `THREAD_PRIORITY_NORMAL`.

Erhält der Thread die Priorität `THREAD_PRIORITY_TIME_CRITICAL`, dann gehen praktisch keine Datensätze mehr verloren, aber der Messprozess kann nicht mehr un-

terbrochen werden und die visuellen Labwindows-Steuerelemente werden nicht mehr alle korrekt aktualisiert. Wenn mit kleinen Teilern (hohen Sektorzahlen) gearbeitet wird, dann bestehen also zwei Möglichkeiten:

- hohe Threadpriorität, weitgehende Fehlerfreiheit, dafür Verzicht auf die simultane Messwertausgabe und die Interventionsmöglichkeiten in den Messablauf
- normale Threadpriorität, gute Bedienbarkeit, Inkaufnahme des sporadischen Messdatenverlustes

Unter diesen Gesichtspunkten wurde die Messwertaufnahme des Programmes „devload2“ in einen separaten Thread ausgelagert dessen Priorität einstellbar ist.

Da die Ringlaserwerte jeweils Sektorenweise gespeichert werden, würden beim Verlust eines oder mehrerer Messwerte die folgenden Werte den jeweils falschen Sektoren zugeordnet und damit die Messung wertlos werden. Um diesen Fall abzufangen, wurde in das Programm „devload2“ eine Fehlerkorrektur implementiert. Tritt nun während der ersten drei Umdrehungen ein Datenverlust ein, so werden die verlorengegangenen Ringlaserwerte durch den Mittelwert über alle Sektoren ersetzt. Oberhalb von drei Umdrehungen wird der Messwert aus dem Mittelwert der Ringlaserwerte des jeweiligen Sektors über alle Umdrehungen gebildet. Alle weiteren Werte wie die des Timers und des Drehgebers werden aus den jeweiligen Erwartungswerten gebildet.

Nichtsdestotrotz ist aber Messinformation verlorengegangen, die durch eine Korrektur nicht ersetzbar ist. Es obliegt zunächst dem Messenden, anhand der aufgetretenden absoluten und prozentualen Fehlerhäufigkeit sowie der Fehlerverteilung, zu entscheiden, wie vertrauenswürdig die Messreihe noch ist. Nach der Methode des „scharfen Hinsehens“ scheint diese Art der Korrektur hinreichend zu sein, um aber genaue Ergebnisse zu erhalten, sind noch umfassende Tests und Messungen nötig.

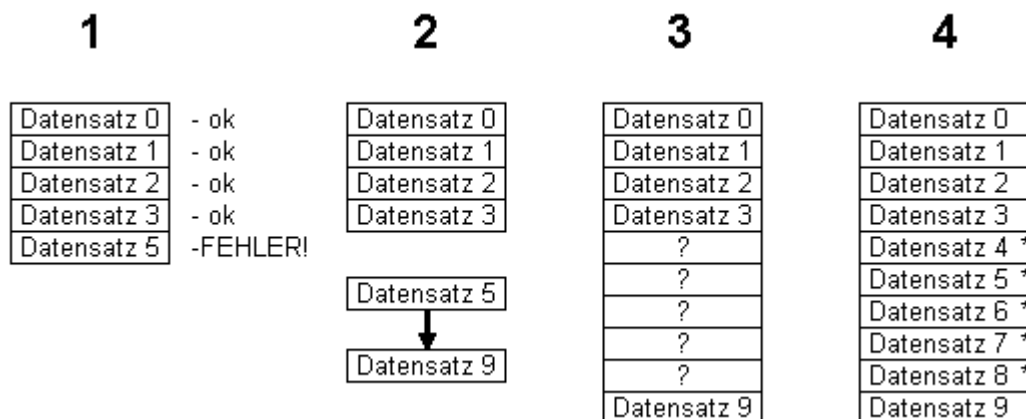


Abbildung 12: die Datensatzkorrektur

(1) Bemerkten des Fehlers, (2+3) Verschieben des aktuellen Datensatzes, (4) Erzeugen der fehlenden Datensätze.

Die Abbildungen 12 und 13 zeigen die Datensatzkorrektur schematisch. Während der Programmentwicklung war dieser Teil der problematischste, da hier leicht logische und Programmierfehler entstehen werden können.

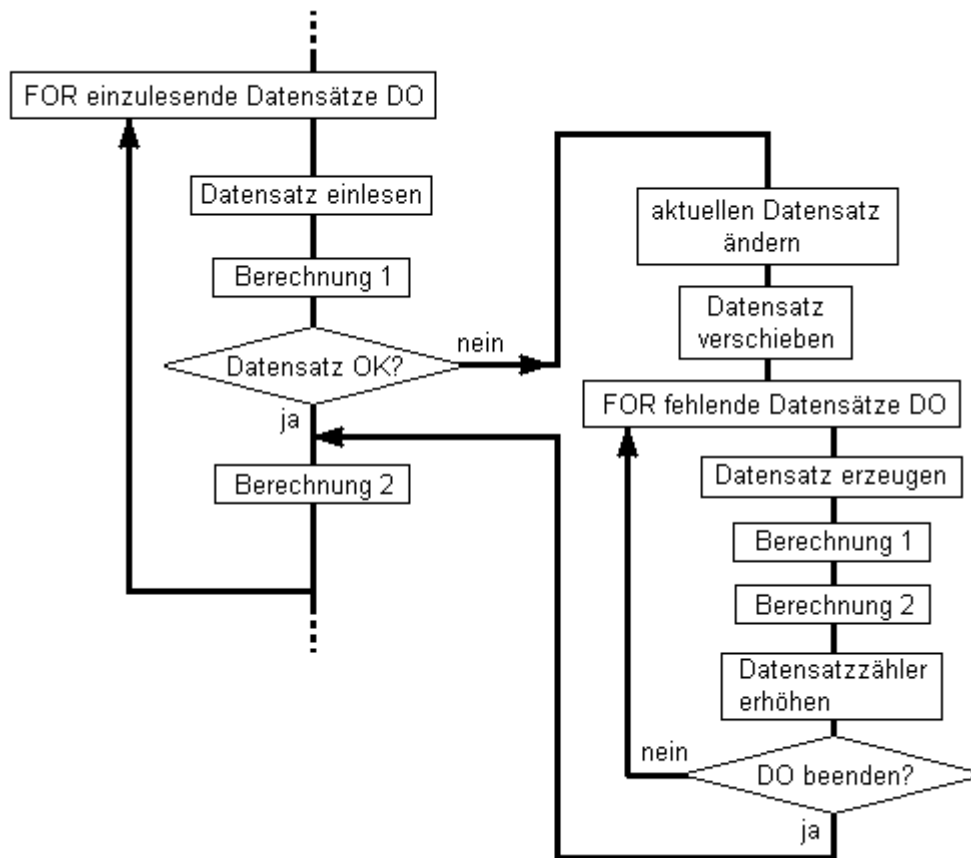


Abbildung 13: die Datensatzkorrektur - Programmablaufplan

7.3.1 Die interne Datendarstellung

Ein Datensatz ist wie folgt aufgebaut:

```

typedef struct {
    DWORD dwLG;           // der Ringlaserwert, ganze Impulse
    DWORD dwLGoever;     // gebrochene Ringlaserimpulse
    DWORD dwODU1;        // Drehgeberwert 1, ganz
    DWORD dwODU1over;    // Drehgeberwert 1, gebrochen
    DWORD dwODU2;        // Drehgeberwert 2, ganz
    DWORD dwODU2over;    // Drehgeberwert 2, gebrochen
    DWORD dwTIME;        // Timer-Zählwert
    int   iCycle;        // Zyklus
    int   iTurn;         // Umdrehung in der Sequenz
    int   iSector;       // Sektor
    int   iSequ;         // Sequenz
    DWORD dwStatus;      // Datensatz korrekter oder erzeugt
    double dfLG;         // Ringlaserwert in Gleitkomma-Darstellung
    double dfLGm;        // Ringlaser-Mittelwert, gleitend
    double zeit;         // Zeitwert
    double dfxi;         // Ringlaserdifferenz
    double dfxi_omega;   // Ringlaserdifferenz, um Erddrehrate korrigiert
}
  
```

```
double dfcal_omega;    // Kalibrierwert des aktuellen Sektors, gleitend
double dfcal_plain;   // Kalibrierwert, nicht korrigiert
double dfw_omega;     // Winkelwert
double dfw_omega_korr; // Winkelwert, Ringlaserdrift-korrigiert
} TnamedDataSetEx;
```

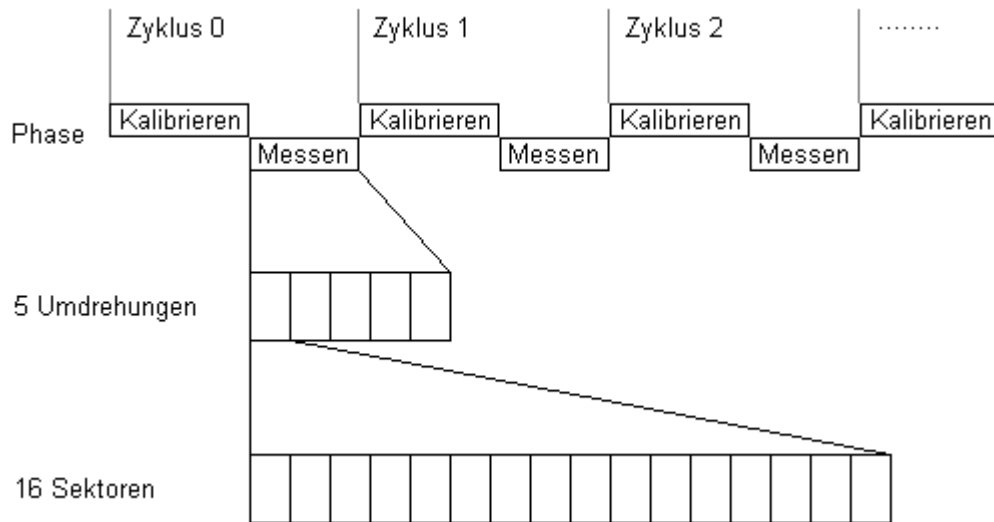


Abbildung 14: Die Darstellung der Messzyklen im Programm

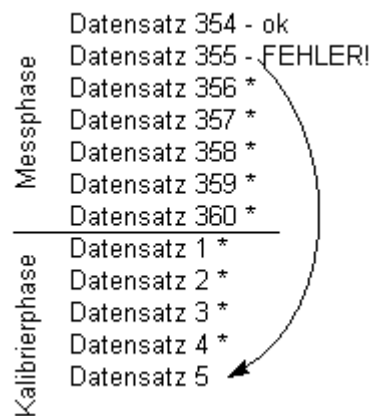


Abbildung 15: Datenkorrektur - Übertrag

Zu Beginn wurde der Messzyklus programmintern so implementiert, wie in Abbildung 11 dargestellt. Es wurden drei Datenpuffer verwendet, die jeweils die Kalibrier-, Mess- und Rekalibrierdaten aufnahmen. Es entstand aber während der Entwicklung der Wunsch, die komplette Messesequenz mehrfach hintereinander ablaufen lassen zu können, insbesondere um das Driftverhalten des Lasers besser beurteilen zu können.

Weiterhin zeigte sich, dass diese Struktur die Datensatzkorrektur sehr kompliziert macht.

Beispiel:

Während einer Messphase mit 5 Umdrehungen und 360 Sektoren ($5 * 360 = 1800$) wird im 355-ten Datensatz der letzten Umdrehung ein Fehler festgestellt, der 10 fehlende Datensätze ausweist, wie in Abbildung 15 dargestellt. Der aktuelle Datensatz stammt nun bereits aus der nächsten Kalibrierphase. So müssen für die Messphase sechs Datensätze korrigiert bzw. neu erzeugt werden und für die darauffolgende Kalibrierphase vier Datensätze.

Da es auch Messungen gibt, bei denen die Messphase aus 0 Datensätzen besteht, also nur kalibriert wird, müssen bei einem solchen Übertrag mehrere Faktoren berücksichtigt werden und es ist einzusehen, dass die Korrektur unter Beachtung aller Eventualitäten kompliziert und fehlerträchtig wird.

Deshalb wurde in einem zweiten Ansatz eine Struktur verwendet, bei der die Datensätze aller hintereinander in nur einen Datenpuffer eingelesen und evtl. korrigiert werden und dann erst die aktuelle Umdrehung, der Sektor und der aktuelle Zyklus, also die genaue Lage des Datensatzes berechnet werden.

Die Datensätze werden in ein Feld, mit einem fortlaufenden Index m eingelesen. Mit der

Sektoranzahl S
 Zyklusanzahl R
 Anzahl der Umdrehungen pro Kalibrierphase U_k
 Anzahl der Umdrehungen pro Messphase U_m

berechnet sich die Größe des Feldes Z und somit die Datensatzanzahl wie folgt:

$$Z = (U_k \cdot S) + R \cdot ((U_k + U_m) \cdot S)$$

die aktuelle Umdrehung u :

$$u = m \mathbf{div} S$$

die aktuelle Sektornummer k :

$$k = m \mathbf{mod} S$$

die aktuelle Umdrehung in der jeweiligen Phase a und die Sequenznummer A :

$$a = u \mathbf{mod} (U_k + U_m) \text{ und}$$

$$a = \begin{cases} a - U_k & \text{für } a \geq U_k \\ a & \text{für } a < U_k \end{cases}$$

$$A = \begin{cases} 2 & \text{für } a \geq U_k \\ 1 & \text{für } a < U_k \end{cases}$$

Ist die Sequenznummer $A = 1$, so handelt es sich um eine Kalibrierphase, ist $A = 2$, ist es eine Messphase. Der aktuelle Zyklus z ist:

$$z = u \mathbf{div} (U_k + U_m)$$

Mit den dargestellten Parametern lässt sich der aktuelle Datensatz aus einem linear angelegten Feld zuordnen, die unterschiedlichen Berechnungen während der Mess- bzw. Kalibrierphase lassen sich nun vornehmen.

Für das Resultat jeder Umdrehung wurde auch ein Feld angelegt, welches unter anderem den Ringlaserzählwert über eine volle Umdrehung enthält - den Skalenfaktor. Weiterhin wird aus zwei jeweils aufeinander folgenden Kalibrierphasen die Ringlaserdrift für die dazwischenliegende Messphase berechnet und in einem Feld abgelegt.

Nach der Umstellung der Datensatzablage funktionierte die Datensatzkorrektur sehr zufriedenstellend. Messungen mit beispielsweise 360 Sektoren, die, bei einer Priorität des Messthreads unterhalb von `THREAD_PRIORITY_TIME_CRITICAL`, durchschnittlich 20 bis 30 fehlende Datensätze pro Umdrehung aufweisen, können nun problemlos verwendet werden. Bei einer höheren Sektoranzahl von 720 und mehr, treten auch bei maximaler Threadpriorität Fehler auf, die aber nun korrigiert werden.

7.4 Korrektur der Teilungsfehler des Drehgebers

Zunächst werden in der Kalibrierphase, während der das Gerät stillsteht und der Laser nur mit seiner Grunddrehrate und der Erdrotation rotiert, die Sektormittelwerte gebildet. Teilungsfehler der einzelnen Sektoren drücken sich in verschiedenen Mittelwerten aus. Während der Messphase wird dann von den Ringlaserwerten der Mittelwert des entsprechenden Sektors abgezogen. Idealerweise ergibt sich hier während der Kalibrierphase und unter Vernachlässigung der Ringlaserdrift der Wert 0.

Abbildung 16 zeigt die Bildung der Sektormittelwerte an einem Beispiel

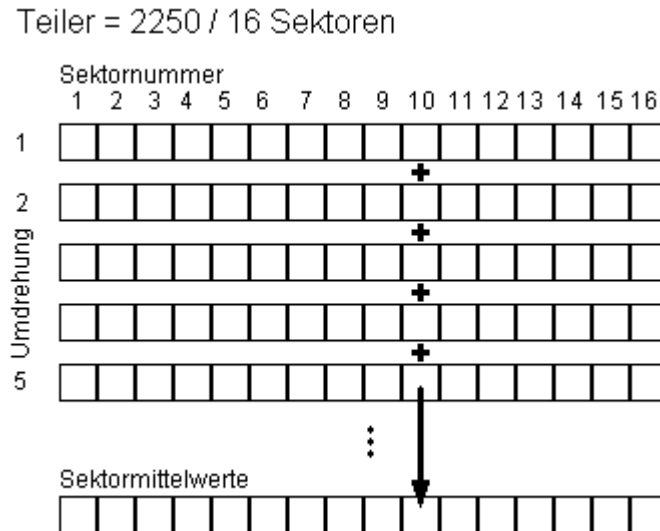


Abbildung 16: Bildung der Sektormittelwerte

Jeder Datensatz LG (Laser Gyro) in einer Phase ist mit der Umdrehungsnummer und der Sektornummer in der Form LG_{ak} ansprechbar. (Indizes: a = Umdrehung, k = Sektor) Der Ringlaser-Mittelwert berechnet sich dann zu:

$$LGM = \frac{1}{U * S} \sum_{a=0}^U \sum_{k=0}^S LG_{ak}$$

mit der Umdrehungsanzahl U und Sektoranzahl S.

Der Sektormittelwert für den k-ten Sektor:

$$LGM_k = \frac{1}{U} \sum_{a=0}^U LG_{ak}$$

Nach jeder Umdrehung wird die Standardabweichung der Sektormittelwerte berechnet und die Werte werden um die Erddrehrate korrigiert und in ein Feld abgelegt.

7.5 Die Driftkorrektur

Abbildung 19 zeigt einen Messzyklus. Es wird hier für die Driftkorrektur die Annahme zugrundegelegt, dass Drift linear verläuft. Es erscheint nicht sinnvoll, für die Driftapproximation Modelle höherer Ordnung anzunehmen, da die Drift sich vom eigentlichen Messwert trennen lassen muss. Die Fläche unter der in Abbildung 19 dargestellten Kurve in der Messphase stellt den Winkel dar, um den gedreht wurde. So kann die Fläche, die aus dem Viereck Φ_{kf} und dem Dreieck Φ_{kv} gebildet wird, als der durch die Drift verursachte Anteil angesehen werden, der eliminiert werden kann.

Die Ringlaserdrift D , der Anstieg der Strecke \overline{AB} , aus den Ringlasermittelwerten LG_{cal1} und LG_{cal2} der beiden Kalibrierphasen:

$$D = \frac{LG_{cal2} - LG_{cal1}}{(U_k + U_m)}$$

Mit dem umdrehungsbezogenen aktuellen Messwertindex $V_{am} = \frac{u \cdot S + k}{S}$ kann nun jeder Ringlaserwert aus der Messphase um seinen entsprechenden Driftanteil korrigiert werden, der, aus einem festen Anteil Φ_{kf} und einem variablen Anteil Φ_{kv} besteht.

$$\Phi_{kf} = U_k \cdot \frac{D}{2} \cdot V_{am}$$

$$\Phi_{kv} = V_{am}^2 \cdot \frac{D}{2}$$

$$\Phi_k = \Phi_{kf} + \Phi_{kv} = \frac{D}{2} \cdot V_{am} \cdot (V_{am} + V_k)$$

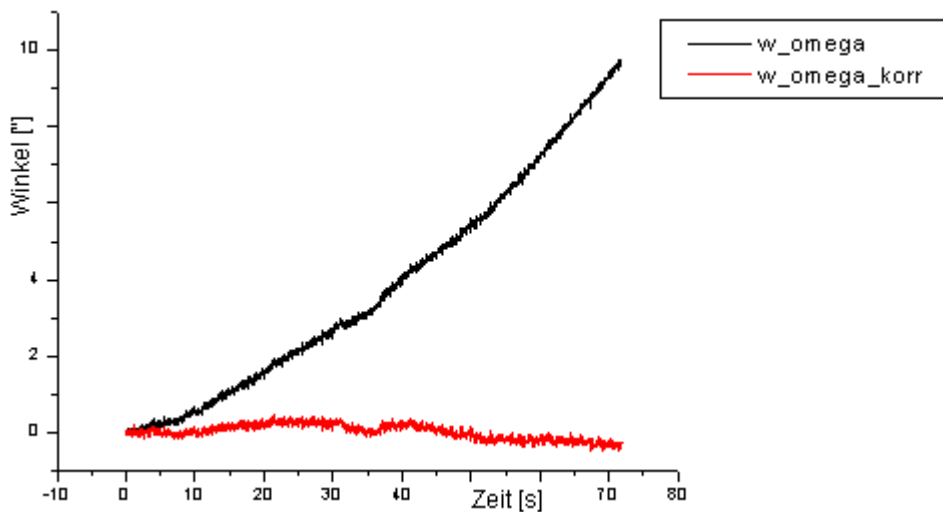


Abbildung 17: Resultat der Driftkorrektur (1) bei 3000 Messwerten (50 Umdrehungen, 60 Sektoren)

Die Abbildungen 17 und 18 zeigen jeweils die Resultate des Driftkorrektur, wobei es sich bei der schwarzen Kurve um die originalen Daten handelt und bei der roten Kurve um die korrigierten.

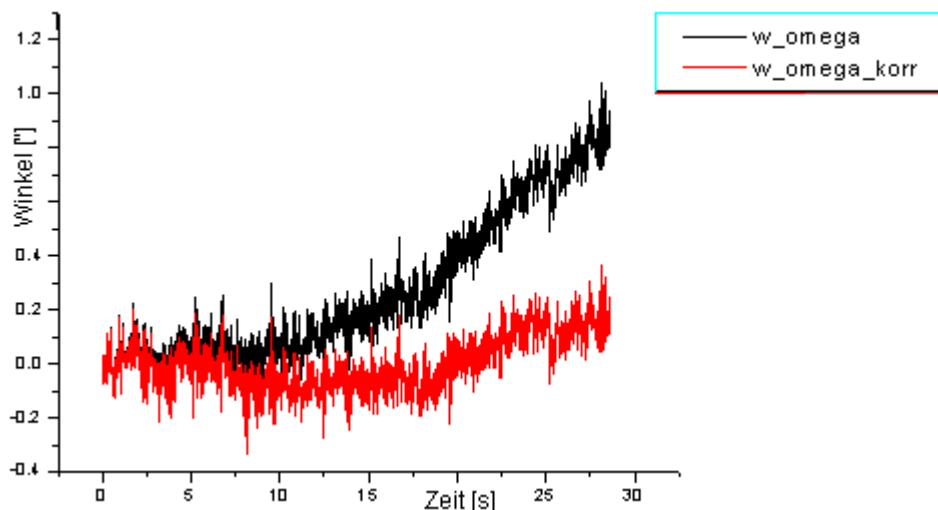


Abbildung 18: Resultat der Driftkorrektur (2) bei 7200 Messwerten (20 Umdrehungen, 360 Sektoren)

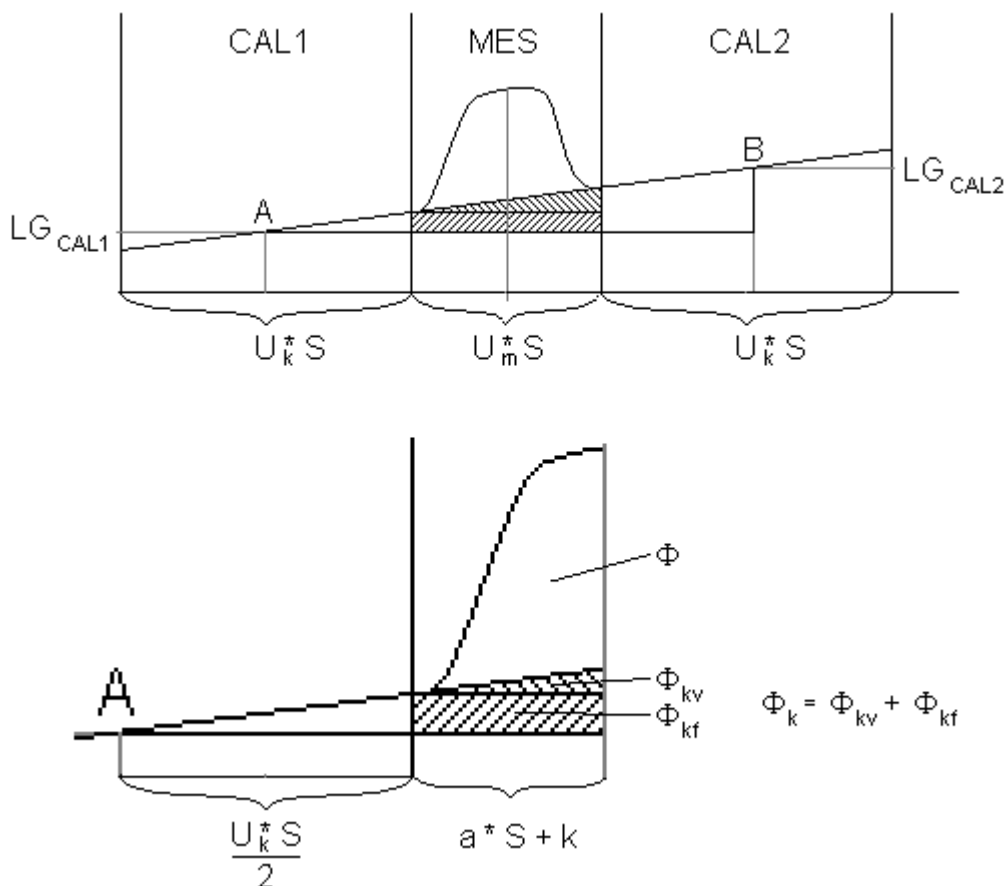


Abbildung 19: die Driftberechnung

7.6 Die Programmoberfläche

Abbildung 20 zeigt das Hauptfenster des Programms „devload2“ nach einer Messung über einen Zyklus mit jeweils drei Umdrehungen pro Kalibrierphase, zwei Umdrehungen

pro Messphase und 360 Sektoren. Sinnvoll bezüglich einer statistischen Auswertung sind so geringe Umdrehungsanzahlen sicher nicht, aber es ging hier lediglich um die Demonstration. Etwas oberhalb der Mitte befinden sich fünf Textausgaben, unter Labwindows „STRING“ genannt. Ganz links, mit der Beschreibung „turn“ befindet sich eine Statusausgabe, mit der Zyklusnummer, dem Kürzel der aktuellen Phase „C“ = Calibration, der aktuellen Umdrehung, 3 und der Anzahl der Umdrehungen pro Kalibrierphase, auch 3. Zum Schluss, in Klammern steht, die Anzahl der korrigierten Datensätze: 142. Bei $(3 + 2 + 3) * 360 = 2880$ Datensätzen entspricht dies einer prozentualen Fehlerrate von 4.9%. Rechts neben diesem Fenster mit der Beschreibung „w_turn“ wird der aktuelle Winkel, um den das ganze Gerät gedreht wurde, in Winkelsekunden angegeben. Dass dieser Wert nicht, wie es im Stillstand zu erwarten wäre, gleich Null ist, beruht auf dem dem Ringlaser eigenen Random-Walk, dem bereits erwähnten Betrieb im um 180 Grad gedrehten Zustand und eventuell auch auf einem nicht linearen Driftverhalten des Lasers. Rechts neben dem Winkelwert wird die Standardabweichung der aktuellen Mess- bzw. Kalibrierphase in Ringlaserimpulsen ausgegeben. Links unten befindet sich die Ausgabe des Drehwinkels in Grad-Minuten-Sekunden und rechts daneben die Ausgabe des Skalenfaktors über eine volle Umdrehung.

Rechts oberhalb der Messwertausgabe befinden sich Eingabefelder für die für die Messung notwendigen Parameter: Anzahl der Zyklen, Anzahlen der Umdrehungen pro Phase, Sektoranzahl und die maximale Messwertanzahl. Der untere Teil des Fensters wird von einer Textbox eingenommen, in die hinein sowohl Statusmeldungen des Programms, als auch die Messwerte ausgegeben werden. Am Ende einer kompletten Messung werden können die Daten automatisch in fünf Dateien abgespeichert werden, die jeweils andere Daten beinhalten und deren Namen automatisch aus dem Datum und einem durchlaufenden Buchstaben generiert werden.

Die erste Messung am 25.07.2003 würde z.B. folgende Dateinamen erhalten:

a25072003_had.dat	Headerfile, beinhaltet Informationen zur Messung
a25072003_com.dat	Kalibrierphase: um die Erddrehung korrigierte Ringlaserwerte und Kalibrierwerte
a25072003_cxi.dat	Kalibrierphase: Ringlaserwerte, nicht um Erddrehrate korrigiert
a25072003_mes.dat	Messphase: Ringlaser-, Kalibrier- und Winkelwerte
a25072003_ska.dat	Skalenfaktorwerte für jede Umdrehung

Einstellmöglichkeiten, die nicht unmittelbar für die Messung notwendig sind, wurden in ein separates Fenster ausgelagert, das in Abbildung 21 dargestellt ist. Hier kann die Rotationsrichtung der Grunddrehrate angegeben werden, die Priorität des Threads für die Messdatenaufnahme, der Ordner, in den die Dateien geschrieben werden sollen und welche Dateien erzeugt werden sollen. Links in der Mitte befinden sich zwei Checkboxes, mit denen die DTR- bzw. RTS-Leitung einer der beiden seriellen Schnittstellen gesetzt werden können. Dies dient dem Test der entsprechenden Funktionen, die bereits vorbereitet wurden, um später ein externes Gerät auf diese Weise takten zu können. Darunter befindet sich ein Bereich, der dem Test des Aufbaues dient, und anzeigt, ob Ringlaser- und Drehgebersignale in der Messkarte ankommen. Die Einträge in der ganz unten liegenden Textbox werden als Kommentare mit in die Headerdatei geschrieben. Sie können vom Messenden verändert werden.

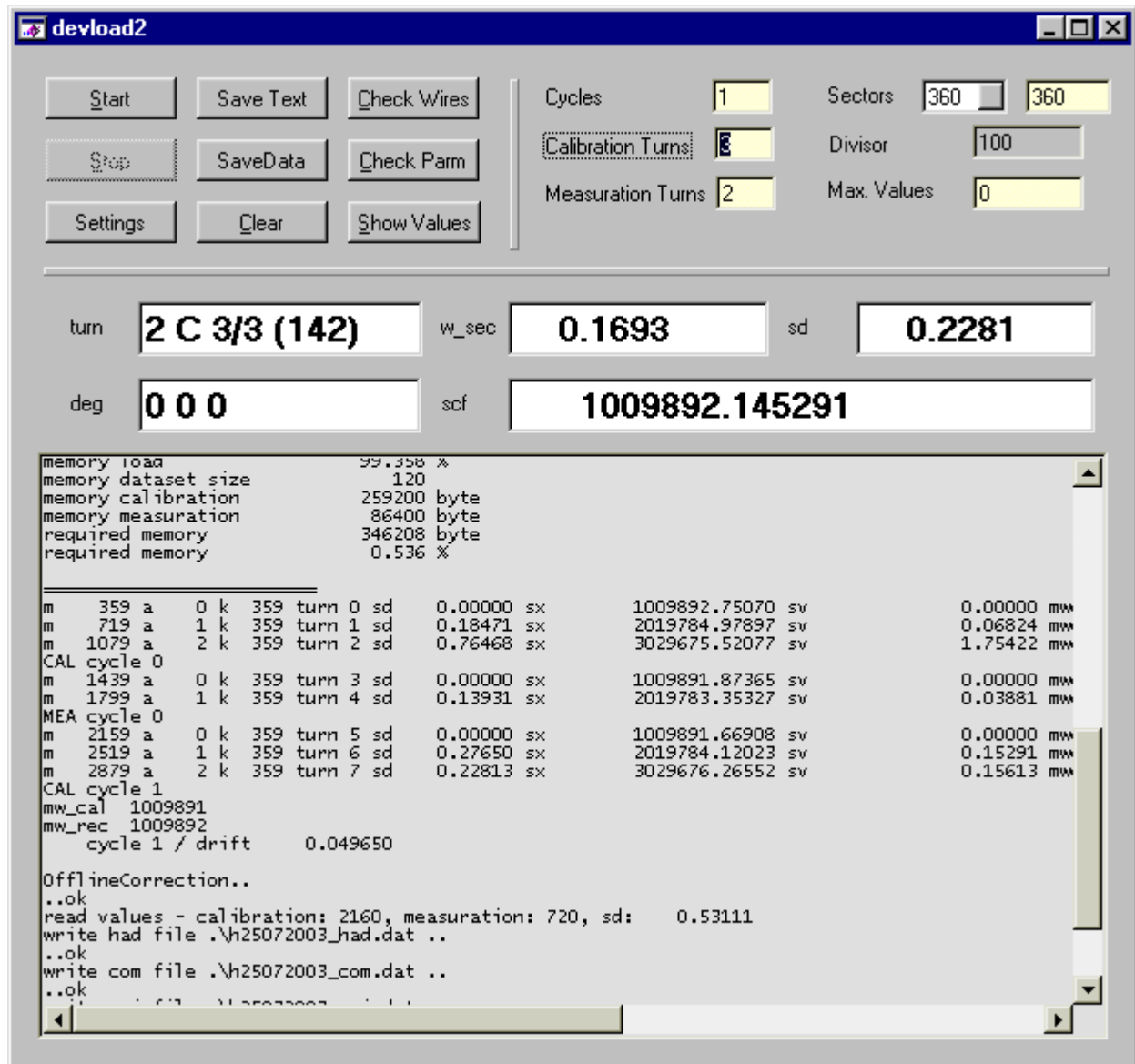


Abbildung 20: Ansicht des Programmfensters

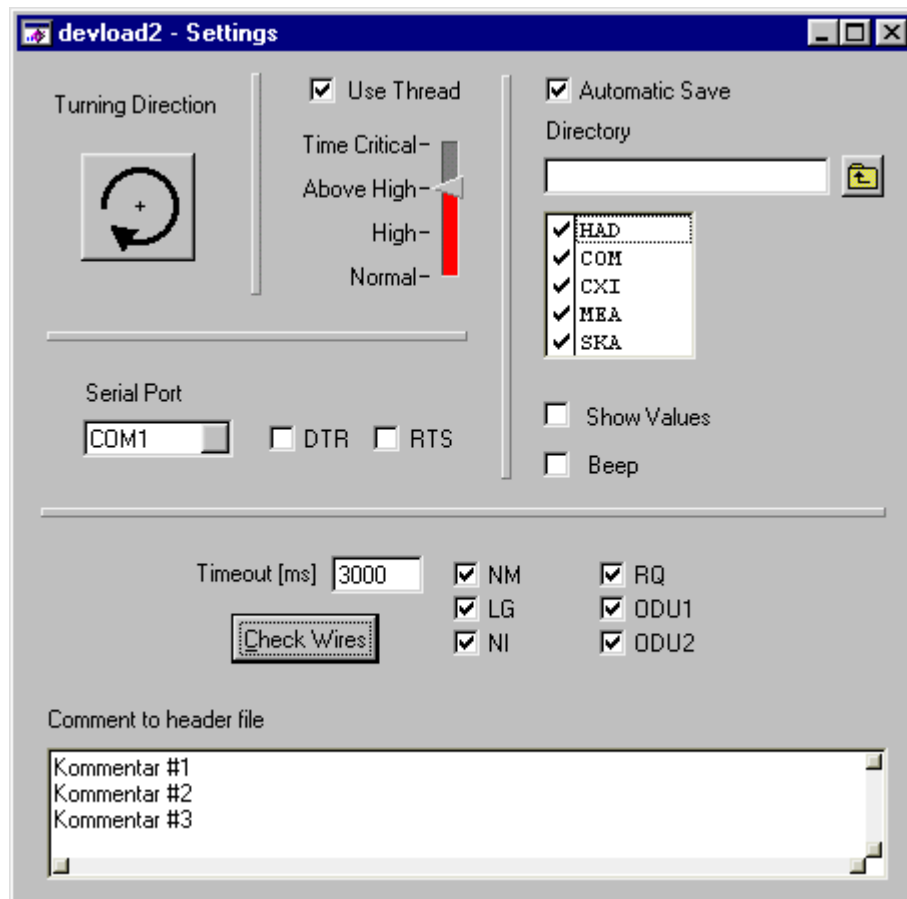


Abbildung 21: Ansicht des Fensters für die Einstellungen

8 Labwindows/CVI

8.1 Umgebung und Sprachkonzept

Labwindows/CVI ist eine Programmierumgebung der Firma National Instruments. Labwindows vereint RAD⁶ mit einer grossen Funktionsbibliothek für den Zugriff auf Messgeräte und für die Darstellung diverser Steuerelemente. Die Programmierung beginnt mit der Entwicklung einer Programmoberfläche, die die grafischen Programmlemente wie Buttons, Ein- und Ausgabeelemente, in Labwindows "Controls" genannt, enthält.

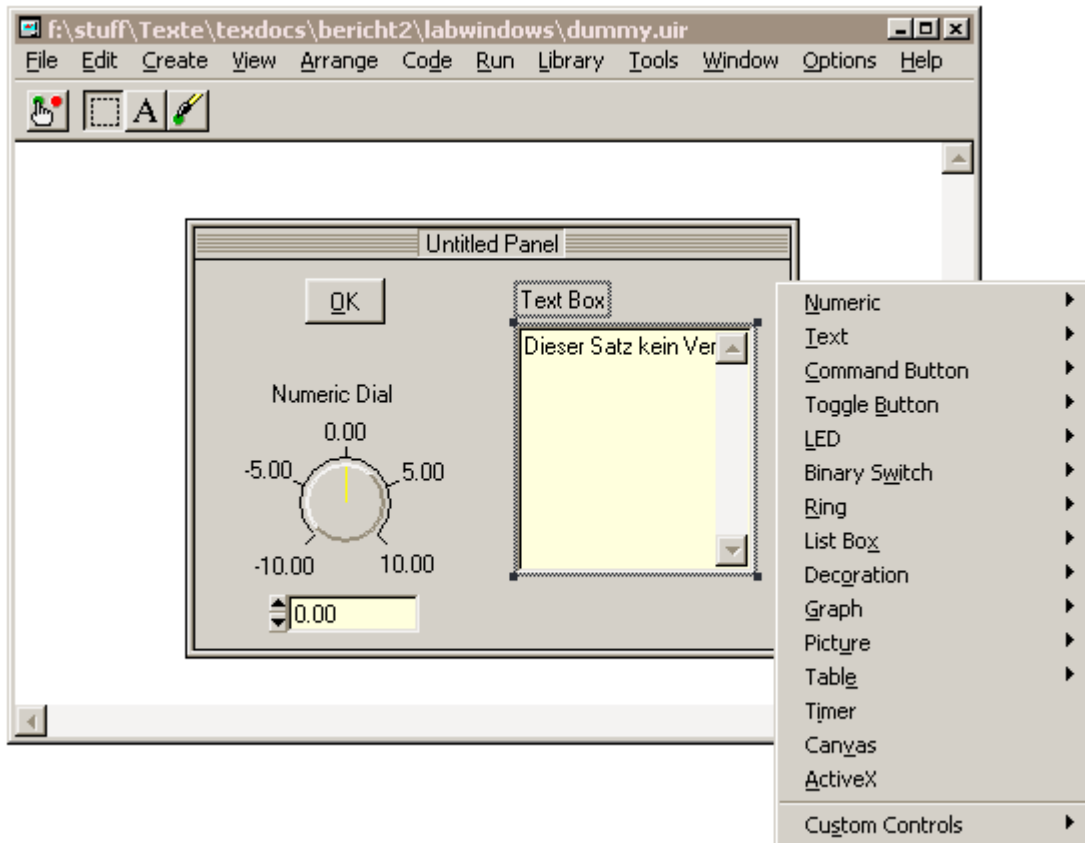
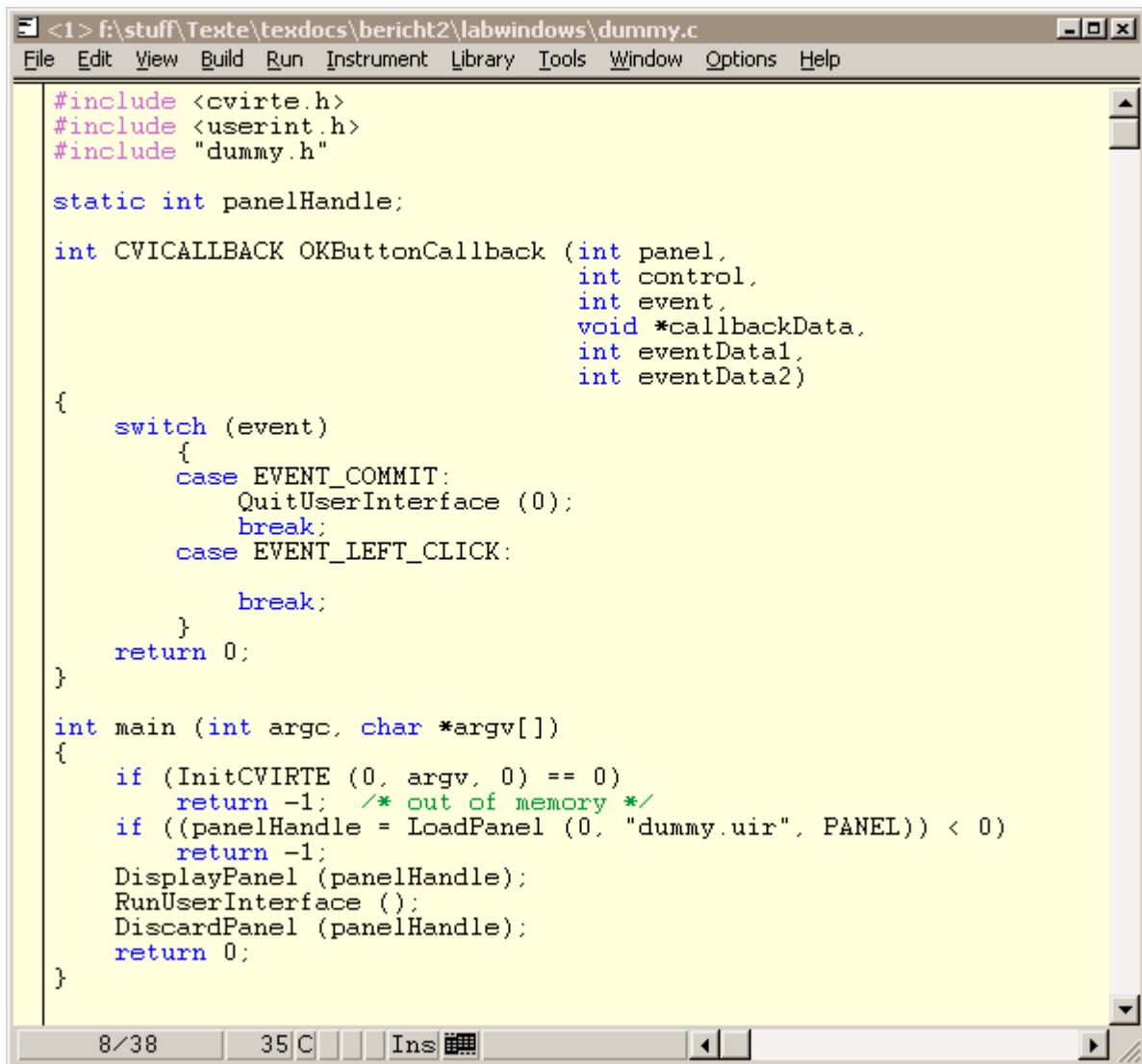


Abbildung 22: ein Demo-Panel

Abbildung 22 zeigt den Programmoberflächeneditor mit einer Programmoberfläche (Panel) und dem Pull-Down-Menu, das die möglichen grafischen Elemente enthält. Ist die Oberfläche des Programms prinzipiell fertiggestellt, wird Labwindows angewiesen, den zugehörigen C-Quellcode zu erstellen, der dann in einem Quellcodeeditor geändert und erweitert werden kann. Er wird in Abbildung 23 gezeigt und besteht aus der Funktion `main` und der Callback-Funktion `OKButtonCallback`, die beim Betätigen des Buttons „OK“ aufgerufen wird, und nichts weiter tut, als das Programm zu beenden. Der automatisch erstellte Quellcode wird dann per Hand weiter ausgebaut und von Labwindows wiederum nur auf ausdrückliche Anweisung des Anwenders geändert. Der C-Dialekt von Labwindows ist auf einfache und schnelle Verwendbarkeit und Erlernbarkeit hin ausgelegt. Die Labwindows-Bibliotheksfunktionen sind bezüglich des Funktionsumfangs sehr

⁶Rapid Application Development



```

<1> f:\stuff\Texte\texdocs\bericht2\labwindows\dummy.c
File Edit View Build Run Instrument Library Tools Window Options Help
#include <cvirte.h>
#include <userint.h>
#include "dummy.h"

static int panelHandle;

int CVICALLBACK OKButtonCallback (int panel,
                                  int control,
                                  int event,
                                  void *callbackData,
                                  int eventData1,
                                  int eventData2)
{
    switch (event)
    {
        case EVENT_COMMIT:
            QuitUserInterface (0);
            break;
        case EVENT_LEFT_CLICK:
            break;
    }
    return 0;
}

int main (int argc, char *argv[])
{
    if (InitCVIRTE (0, argv, 0) == 0)
        return -1; /* out of memory */
    if ((panelHandle = LoadPanel (0, "dummy.uir", PANEL)) < 0)
        return -1;
    DisplayPanel (panelHandle);
    RunUserInterface ();
    DiscardPanel (panelHandle);
    return 0;
}

```

Abbildung 23: ein Demo-Quellcode mit der Hauptfunktion „main“ und der CallBack-Funktion eines Buttons

leistungsfähig und auch die ANSI-C-Funktionen sind verfügbar. Wahlweise kann eine Release-Version oder eine Debug-Version erstellt werden, die dann unter anderem Arraygrenzen überprüft, Schutzverletzungen abfängt und die Programmentwicklung sehr vereinfacht. Funktionen und Operatoren können nicht überladen werden. Das Konzept ist nicht objektorientiert, sondern rein prozedural. Der Zugriff auf die Werte und Eigenschaften von Steuerelementen erfolgt über das jeweilige Panel, auf denen sich die Elemente befinden und über die Elementnummer. Im in Abbildung 22 dargestellten Beispiel wird das Panel über die Konstante `PANEL = 1`, die im Headerfile des Projektes festgelegt ist, referenziert. Der links oben liegende Button hat die Konstante `PANEL_BUTTON1 = 1`. Um den Wert und die Eigenschaften eines Steuerelementes abfragen bzw. ändern zu können, gibt es die Funktionen:

```

GetCtrlVal(PANEL, PANEL_BUTTON1, &value) bzw.
SetCtrlVal(PANEL, PANEL_BUTTON1, &value)

```

und

```
GetCtrlAttribute(PANEL, PANEL_BUTTON1, ATTR_LEFT, &left) bzw.
SetCtrlAttribute(PANEL, PANEL_BUTTON1, ATTR_TOP, &top)
```

An die Funktionen werden jeweils die für die Referenzierung notwendigen Konstanten und die Referenz des zu setzenden bzw. zu liefernden Wertes übergeben. Selbstverständlich umfasst der Funktionsumfang weit mehr, aber der Zugriff ist im Prinzip einfach und wenn die Modalitäten einmal verstanden wurden, dann lassen sich leicht weitere Programmiermöglichkeiten erarbeiten.

Labwindows bindet auf Wunsch auch wahlweise den Visual-C++-Compiler von Microsoft bzw. den Borland C++-Compiler mit in seine Installation ein. Der Labwindows-Quelltext kann dann auch mit dem externen Compiler übersetzt werden, wovon hier kein Gebrauch gemacht wurde. PTB-Mitarbeiter, die bereits über umfangreiche Labwindows-Erfahrungen verfügen, und deshalb bei Problemen um Rat gefragt, nutzen diese Möglichkeit aber intensiv, da der Maschinencode schneller und effizienter ist.

8.2 Labwindows - Beispiele

Exemplarisch sollen nun einige weitere Funktionen gezeigt werden.

Die Funktion

```
CmtScheduleThreadPoolFunctionAdv(DEFAULT_THREAD_POOL_HANDLE,
                                detect_wires,
                                0, 0, NULL, 0, NULL, NULL,
                                &TID)
```

startet eine Funktion in einem separaten Thread, wobei `detect_wires` der Name der entsprechenden Funktion ist.

Beim Zugriff auf .DLL-Funktionen müssen vom Linker die Namen der exportierten bzw. importierten Funktionen in Adressen aufgelöst werden. Dazu beinhaltet die .DLL-datei eine entsprechende Tabelle, die normalerweise ausreicht, so dass das entgeltige Programm nur aus der ausführbaren .EXE-Datei und der .DLL-Datei besteht. Labwindows benötigt aber für das dynamische Linken zusätzlich noch eine .LIB-Datei, die die Namensauflösung ermöglicht. Eine solche Datei wird beispielsweise von Microsoft Visual-C++ erzeugt. Eine von einer .DLL-Datei exportierte Funktion (`readNIdataSet`) kann nun wie folgt in ein Labwindows-Programm eingebunden werden:

```
// (1) Laden des .DLL-Moduls (bzw. .LIB-Moduls)
    int moduleID = LoadExternalModule("rddx2.lib");
// (2) Funktionsprototyp: ein Zeiger der die Adresse der Funktion enthält
    DWORD (*DLLreadNIdataSet) (void);
// (3) Auflösung Funktionsname --> Adresse
    DLLreadNIdataSet = ResolveFunctionAdress(moduleID, "readNIdataSet");
// (4) Aufruf der Funktion
    (*DLLreadNIdataSet)();
```

Die Ausgabe einer Textzeile in die letzte Zeile einer TextBox und Scrollen zu dieser Zeile ist leider etwas aufwändig, aber eine Funktion, die die Schritte zusammenfasst (nennen wir sie: `AddTextBoxLine()`), ist schnell geschrieben.

```
int vL, tL;
GetNumTextBoxLines(PANEL, PANEL_TEXTBOX, &tL);
InsertTextBoxLine(PANEL, PANEL_TEXTBOX, tL, "Ein Satz kein Verb");
GetCtrlAttribute(PANEL, PANEL_TEXTBOX, ATTR_VISIBLE_LINES, &vL);
GetCtrlAttribute(PANEL, PANEL_TEXTBOX, ATTR_TOTAL_LINES, &tL);
SetCtrlAttribute(PANEL, PANEL_TEXTBOX, ATTR_FIRST_VISIBLE_LINE, tL - vL);
```

In Anlehnung an die ANSI-C Funktion `printf()`, deren Ausgabe durch Formatierungsstrings gesteuert wird, bietet Labwindows die Funktion `Fmt()`. Folgende Codesequenz erzeugt dir Zeichenkette `outstr`, die Festkommazahl `i` und die Gleitkommazahl `d`. Dann werden `i` und `d` mit zehn Stellen und drei Nachkommastellen in `outstr` hinein kopiert.

```
char* ostr = (char*)malloc((size_t)sizeof(char) * 256);
int i = 5;
double d = 3.1415927;
Fmt(ostr, "%s<%d %f[w10p3]", i, d);
```

Äquivalent erfolgt, mit einem vorher zu erzeugenden Datei-Handle, auch eine Ausgabe in eine Datei hinein:

```
FmtFile(filehandle, "%s<%d %f[w10p3]", i, d);
```

Natürlich bietet Labwindows weit mehr, aber bereits mit einigen wenigen Funktionen lässt sich allerhand anfangen. Die Einarbeitungszeit ist recht kurz.

9 Zusammenfassung und Ausblick

Das Ziel der Projektarbeit, ein Programm zu schaffen, das die neuen Möglichkeiten der Elektronik, z.B. in Bezug auf die Interpolation, nutzt, ist erreicht worden. Da das Programm aber noch nicht dem "harten" Messbetrieb unterworfen wurde, lässt sich noch nicht sagen, ob der Programmaufbau den Anforderungen gerecht wird oder noch grössere Änderungen nötig sind. Insbesondere die Datensatzkorrektur und die vorherige Feststellung ihrer Notwendigkeit, und ihre Verbindung mit den während der Datenaufnahme laufenden Berechnungen war recht schwierig zu realisieren - auch unter dem Gesichtspunkt, dass das fertige Programm möglichst wenige „Bugs“ enthalten sollte.

Insbesondere der vielen Zeit, die Herr Just mit mir zusammen bei Erklärungen und Diskussionen verbrachte, ist es zu verdanken, dass die Problemstellungen und Möglichkeiten ihrer programmtechnischen Umsetzung klar und Fehler erkannt und behoben wurden.

Als nächstes müssen zunächst die bereits laufenden Überarbeitungen am Ringlaser abgeschlossen und dann ausgedehnte Experimente und Messungen gemacht werden, um Aussagen darüber machen zu können, ob sich das Messverhalten des Gerätes verbessert hat oder nicht. Als nächster Schritt steht die Erprobung des Ringlaser-Messgerätes mit der neuen Elektronik und der neuen Software an. Es müssen Experimente auf den verschiedenen Messtischen der PTB und unter verschiedenen Betriebsbedingungen durchgeführt werden, um einschätzen zu können, ob die Massnahmen wesentliche Verbesserungen der Messgenauigkeit (Ziel: $\frac{1}{10}Grad$) zur Folge haben.

Literatur

- [1] YU. V. FILATOV, D.P. LOUKIANOV, R. PROBST *Dynamic angle measurement by means of a ring laser* Metrologia, 1997/34, 343-351
- [2] A. JUST, R. PROBST *Investigation of a ring laser angle measuring instrument for dynamic angle measurements* Proceedings of euspen 1999, Bremen 223-226
- [3] F. ARONOWITZ *Laser Applications* ACADEMIC PRESS New York and London 1971
- [4] E.O.SCHULZ-DUBOIS *Alternative Interpretation of Rotation Rate Sensing by Ring Laser* IEEE Journal of Quantum Electronics, Vol. QE-2, No. 8, August 1966
- [5] DANA Z. ANDERSON *Optical Gyroscopes* Scientific American Vol.254, 4/1986, 86-91
- [6] A. JUST, D.SCHULZ *RL_IWM2.C - Programm für die Kalibrierung und Inertiale Winkelmessung* PTB Braunschweig, 2000
- [7] *devload.cpp - Programm für die Messdatenaufnahme, ETU St. Petersburg*
- [8] D. ULLRICH, E. LÜBECK, W. PAHL, V. WETZIG, R. WITTEKOPF *Winkelmessung mit Laserkreisel* Technisches Messen 10/1988
- [9] R. RODLOFF *Der Laserkreisel* Laser und Optoelektronik 2/1985
- [10] H.-J. BÜSCHELBERGER *Der Ringlaser als Kreisel* Lasermagazin 6/1988
- [11] PTB *Jahresbericht 2002*
- [12] HERING, MARTIN, STOHRER *Physik für Ingenieure* VDI Verlag 1992
- [13] HECHT *Optik* Oldenburg Verlag München Wien 2001
- [14] B. WOHAKE, R. MAURUS *80x86/Pentium Assembler* IWT-Verlag GmbH Bonn 1995
- [15] M. TISCHER *PC-Intern 5* Data Becker 1997
- [16] R. SEDGEWICK *Algorithmen* Addison Wesley 1997
- [17] W.BEITZ UND K.-H. GROTE *Dubbel, Taschenbuch für den Maschinenbau* Springer 1997
- [18] C. PETZOLD *Microsoft Windows Programmierung* Microsoft Press, 2000
- [19] J. RICHTER *Microsoft Windows Programmierung für Experten* Microsoft Press, 2000
- [20] D. KRUGLINSKI, G. SHEPERD, S. WINGO *Inside Visual C++ 6.0* Microsoft Press, 1998
- [21] W. DOBERENZ, T. KOWALSKI *Borland Delphi 6* Hanser, 2002
- [22] E. WARKEN *Delphi 6* Addison Wesley, 2002
- [23] SIEGERT, BAUMGARTEN *Betriebssysteme* R.Oldenburg Verlag, 1998