

Fachhochschule Gießen-Friedberg
Fachbereich Elektrotechnik I (IT)

Diplomarbeit

**Aufbau einer PC-geführten Antriebssteuerung für eine Waferzentrifuge zur
Bearbeitung optischer Skalenscheiben mit grafischer Benutzeroberfläche**
Bedienoberfläche & Prozeßüberwachung (Software)

eingereicht von:

Thomas Jockel
E-Mail: Thomas.Jockel@JoJo.Li

Betreuer:

Prof. Dr. Rainer Thüringer

Ausgabe des Themas:

H.W. Feldmann GmbH
Präzisionsteilungen
Bergstraße 31
35578 Wetzlar

Sommer 1998

Vorwort

Die vorliegende Diplomarbeit wurde bei der H.W. Feldmann GmbH Präzisionsteilungen, Bergstraße 31 in 35578 Wetzlar durchgeführt. Sie wurde durch Herrn Prof. Dr. Rainer Thüringer von der Fachhochschule Gießen-Friedberg betreut.

Herrn Prof. Dr. Rainer Thüringer möchte ich an dieser Stelle für die sehr gute und engagierte Betreuung der Diplomarbeit recht herzlich danken.

Ferner gilt mein Dank Herrn Oliver Feldmann, der Herrn Holger Erb und mir eine Diplomarbeit ermöglichte und uns großzügig unterstützte.

Herrn Dipl. Ing. Ralf Leicht danke ich für die wertvollen Anregungen und die stete Hilfsbereitschaft bei Fragen.

Lich, den 10. August 1998

Unterschrift

Inhaltsverzeichnis

1. EINLEITUNG	5
2. AUFGABENSTELLUNG.....	6
3. DIE STEUERUNGSHARDWARE.....	8
3.1. Allgemeines	8
3.2. Realisierung mit SAB 80C535	8
3.3. Steuer - PC.....	11
4. STEUER- UND BEDIENSFTWARE.....	12
4.1. Allgemeines	12
4.2. Programmiersprache	12
4.3. Aufgabe	13
4.4. Funktionsbeschreibung.....	13
4.4.1. Das Menü	13
4.4.2. Programminitialisierung (Steuerung.cfg)	18
4.4.3. Das Programmhauptfenster.....	18
4.4.4. Ein neues Projekt.....	21
4.4.5. Eingabe der Daten	21
4.4.6. Die Projektbeschreibung und die Projektdaten.....	23
4.4.7. Die Optionen und die Schnittstellen.....	24
4.4.8. Die Projektdatei.....	25
4.4.9. Starten des Prozesses.....	26
5. DIE REGELSOFTWARE	31
5.1. Das interne RAM des Controllers.....	31
5.2. Adressierungsarten.....	31
5.2.1. Registeradressierung.....	31
5.2.2. Direktadressierung.....	32
5.2.3. Register-Indirekt-Adressierung	32
5.2.4. Unmittelbare Adressierung	32
5.2.5. Indirekte, indizierte Adressierung mittels Basisregister plus Indexregister	32
5.3. Regelungen.....	34
5.3.1. Der P-Regler	36
5.3.2. Der PI-Regler	37
5.3.3. Der PID-Regler	38
5.4. Programmiersprache	38
5.5. Aufgabe	39
5.6. Funktionsbeschreibung.....	40
5.6.1. Allgemeines	40
5.6.2. Der Regelkreis.....	43
5.6.2.1. Allgemeines.....	43
5.6.2.2. Funktionsweise.....	44
6. ZUSAMMENFASSUNG	45
7. KRITISCHE SCHLUßBETRACHTUNG	47

8. KURZBESCHREIBUNG	48
9. ANHANG.....	49
9.1. Programmlisting der Steuersoftware.....	49
9.2. Programmlisting der Regelsoftware	117
9.3. Abbildungsverzeichnis	125
9.4. Tabellenverzeichnis.....	126
9.5. Gleichungsverzeichnis	126
9.6. Literaturverzeichnis.....	127

1. Einleitung

Immer häufiger werden in der Industrie Arbeitsvorgänge automatisiert und modernisiert. So stand auch die Firma H.W. Feldmann GmbH vor der Frage, ihre bereits vorhandene, aber veraltete Fertigungseinrichtung zu erneuern. Da in diesem Jahr ein Umzug in ein neues Firmengebäude geplant war, entschied man sich unter anderem, die Fertigungsanlagen für die Strichplattenherstellung zu erneuern.

Für die Fertigungseinrichtung werden nur vier Geräte benötigt. So waren die größten zu erwartenden Kosten, die Entwicklungskosten, weshalb seitens der Firma auch ein Interesse für eine Diplomarbeit vorhanden war. Ich stehe schon seit längerer Zeit in engem Kontakt zur Fa. Feldmann und habe schon am Aufbau der alten Fertigungsanlagen mitgearbeitet.

Da sowohl die Hardware als auch die Software sehr umfangreich waren, wurde die Arbeit auf zwei Diplomanden aufgeteilt. Die einfachste Teilung war die in Hard- und Software. So konnten für die Diplomanden unterschiedliche Interessenbereiche gefördert werden, aber trotzdem war eine enge Zusammenarbeit sehr wichtig, um die Software optimal auf die Hardware abzustimmen. Die genaue Spezifikation wird im Abschnitt "Aufgabenstellung" beschrieben.

Im Zeitalter der PCs und der Vernetzung kam man recht schnell auf die Idee, den Fertigungsprozeß von einem PC aus zu steuern. Das hat verschiedene Vorteile: man kann die Benutzeroberfläche sehr übersichtlich und leicht bedienbar gestalten, weiter hat man die Möglichkeit, bei einer geplanten Vernetzung der Steuer-PCs alle Daten und Projekte zentral zu verwalten. Da die Eckdaten eines jeden Prozesses sehr stark auf Erfahrungswerten und Hintergrundwissen aufbauen, ist es auch sehr sinnvoll, diese Projekte zentral zu verwalten und zu erfassen. Der Mitarbeiter an der Fertigungsanlage braucht lediglich das Projekt zu öffnen, und der Fertigungsprozeß kann beginnen.

2. Aufgabenstellung

Anwendungsgebiet:

Die zu entwickelnde Fertigungsvorrichtung wird bei der Firma Feldmann zur Herstellung von Strichplatten auf optischen Substraten (Glasplatten) genutzt. Hierzu wird eine beschichtete Glasplatte, nachdem sie auf einem Belichtungsgerät mit einem spezifischen Strichmuster belichtet wurde, auf eine rotierende Achse gelegt, auf der sie mit einem Unterdruck angesaugt wird. Diese Achse dreht sich mit verschiedenen Geschwindigkeiten und die Glasplatte wird mit Entwickler betropft, um das belichtete Strichmuster auf photochemischem Weg zu entwickeln. Nach einer Spülung bleibt nur noch das belichtete Strichmuster zurück. Anschließend wird das Substrat noch einmal mit einer sehr hohen Drehzahl getrocknet.

Aufgabenstellung:

Die zu entwickelnde Hard- und Software ist in der Firma bereits in stark veralteter Technik im Einsatz. Unsere Aufgabe war es, diese Anlage auf einen modernen, der Zeit entsprechenden Stand mit erweitertem Funktionsumfang und Ausbaumöglichkeiten zu bringen. Hierzu gab es allerdings einige Eckdaten und Vorgaben. So wurde u.a. der Motor vorgegeben, da dieser schon in anderen Prozessen erfolgreich eingesetzt wird. Die komplette Schleudermechanik wurde nach den alten Vorgaben von einer Drittfirma gebaut und uns zur Verfügung gestellt.

Spezifikation der Diplomarbeit:

1. Hardware

- ⇒ Die Drehzahl der Achse zum Aufnehmen der Substrate soll von 0 bis max. 12.000 min^{-1} kontinuierlich einstellbar sein. Eine geeignete, den Ansprüchen entsprechende Antriebseinheit in Bezug auf Zuverlässigkeit, Langlebigkeit und Verfügbarkeit soll entwickelt werden.
- ⇒ Die vorhandene Mechanik zum Handhaben der Flußmittelzufuhr (elektromechanische Hubmagnete) soll ggf. in modifizierter Form in die Neuentwicklung integriert werden.
- ⇒ Zum Meßwert- bzw. Datenaustausch zwischen Meßwertaufnehmer und Stellglieder müssen Interface-Elektroniken entwickelt werden.
- ⇒ Während des Fertigungsablaufs soll der Fertigungsfortschritt kontrolliert werden. Hierzu sind geeignete Sicherheitsmechanismen bzw. Kontrollelemente in Form von Drehzahlgebern, Vakuumsensoren, Durchflußmessern, Füllstandsmessern usw. in der Fertigungsvorrichtung zu installieren.

2. Software

- ⇒ Die Steuer- und Regelsoftware, die den Charakter einer SPS haben kann, soll wahlweise auf einem PC oder auf einem Mikrocontroller betrieben werden.
- ⇒ In beiden Fällen soll die Bedienung ergonomisch und einfach erlernbar sein, so daß sich auch nicht PC-kundige Mitarbeiter schnell in den programmierbaren Steuerablauf einfinden können.
- ⇒ Bedieneroberfläche mit Pulldown-Menü und Schaltflächen. Sie soll sowohl mit Tastatur als auch mit Maus bedienbar sein.
- ⇒ Das Steuer- und Regelprogramm soll einen modularen Aufbau mit Kommentieren der verschiedenen Funktionsgruppen erhalten.
- ⇒ Das Programm soll eine begrenzte Anzahl (max. je 10) von Drehzahl/Zeit-Paaren und Beschichtungsmedium/Zeit/Volumen-Paaren verwalten. Diese Drehzahl- und Zeiteinstellungen sollen in ihrer zeitlichen Abfolge frei programmierbar sein.
- ⇒ In einem gesonderten Programmodul soll der Fertigungsfortschritt mittels Abfrage von o.g. Kontrollelementen auf eine Fehlfunktion im Fertigungsablauf hin überprüft werden. Ergänzend kann hierzu ein Fehlerprotokoll als Dateiablage mit Möglichkeit zur Einsicht erstellt werden.

3. Die Steuerungshardware

3.1. Allgemeines

Die in der Aufgabenstellung geforderten Bedingungen führten zur Auswahl des Mikrocontrollers SAB 80C535 von Siemens. Dieser bietet alle Möglichkeiten, die wir zur Realisierung benötigten. Die integrierte serielle Schnittstelle wurde z.B. zur Kommunikation mit dem PC genutzt.

Da dieser Teil von Holger Erb ausgeführt wurde, werde ich hier nur sehr oberflächlich auf die Hardware eingehen.

3.2. Realisierung mit SAB 80C535

Der verwendete Mikroprozessor SAB 80C535 (μ P) stellte uns alle notwendigen Funktionen zur Verfügung. Er wird bei einer Taktfrequenz von 11,0592 MHz betrieben. Wir hatten zunächst einen 12,0 MHz Quarz, damit waren aber nur Baudraten bis 4.800Bd zu erreichen. So entschieden wir uns für einen 11,0592 MHz Quarz und konnten mit einer internen Taktverdopplung eine Baudrate von 19.200Bd erreichen. Es werden zwar nicht so viele Daten seriell ausgetauscht, daß eine so hohe Geschwindigkeit nötig wäre, aber so ist die Zeit für die Datenübertragung doch erheblich kürzer, und der μ P kann seine Rechenzeit für die viel wichtigere Drehzahlregelung verwenden. Denn während der Zeit einer seriellen Übertragung darf kein Interrupt ausgelöst werden, das heißt es findet auch keine Drehzahlregelung statt, da der Timer den Interrupt nicht auslösen darf.

Die Beschaltung des externen RAM's wurde von einer Entwicklungsschaltung übernommen. Die Bausteine sind recht günstig und es war zu Anfang auch nicht abzusehen, wie umfangreich die Regel-Software wird und ob dieser RAM für das endgültige Programm benötigt wird.

Wichtige Komponenten und Eigenschaften des μ P SAB 80C535 sind:

- ☒ enthält alle Funktionen der 8051-Familie
- ☒ softwarekompatibel zum 8051
- ☒ 8-Bit CPU
- ☒ 1,2 bis 12,0 MHz Oszillatorfrequenz ($f_{\mu P}$)
- ☒ 1 μ s typische Befehlszykluszeit bei einer Oszillatorfrequenz ($f_{\mu P}$) von 12,0 MHz
- ☒ 256 Byte internes RAM, auf max. 64 kByte extern erweiterbar
- ☒ 41 direkt adressierbare Special-Function-Register (SFR)
- ☒ 128 einzeln adressierbare Bits in den SFR
- ☒ 12 Interrupt-Quellen mit vier Prioritätsebenen
- ☒ drei 16-Bit Zähler/Zeitgeber
- ☒ vier 16-Bit Capture-Eingänge oder Compare-Ausgänge

- ☒ 16-Bit Watchdog-Timer
- ☒ sechs 8-Bit Ein-/Ausgabe-Ports, ein Eingabe-Port für digitale oder analoge Signale
- ☒ Analog-/Digitalwandler mit acht Eingängen, 8-Bit Auflösung, 15 μ s Wandlungszeit und programmierbarer Referenzspannung.
- ☒ Serielle Schnittstelle für Duplex-Betrieb mit vier Betriebsarten und variablen Baudraten.
- ☒ Idle- und Power-Down-Modus
- ☒ 68 Pin PLCC-Gehäuse

Nun möchte ich auf einige spezielle Funktionen des μ P näher eingehen, die für dessen Programmierung von Bedeutung sind.

Erzeugen von Pulsmustern (PWM)

Für die Erzeugung von Pulsmustern ist die Betriebsart "Timer 2: Auto-Reload, Compare-Register: Mode 0" vorgesehen. In dieser Betriebsart wird Timer 2 beim Überlauf (FFFF) auf den im Reload-Register abgelegten Wert zurückgesetzt und zählt von dort mit jedem Zeitschritt (1 μ s) um eins hoch. Bei Erreichen des im jeweiligen Compare-Register abgelegten Wertes gibt der zugehörige Komparator ein Setzsignal an seine Ausgangsstufe. Alle Ausgangsstufen erhalten bei Zählerüberlauf gleichzeitig ein Rücksetzsignal. Für unsere Anwendung heißt das:

- ☒ Die Impulsperiodendauer wird um so kürzer, je größer der Wert im Reload-Register ist. Der kleinstmögliche Wert (0000) liefert eine Impulsperiodendauer t_T von 65.536 μ s (16^4), der größtmögliche Wert (FFFF) liefert ein t_T von 1 μ s.
- ☒ Die Werte in den Compare-Registern müssen mindestens so groß sein, wie der Wert im Reload-Register.
- ☒ Der Wert jedes einzelnen Compare-Registers legt die Einschaltzeit t_e seines zugeordneten Ausgangs fest. Diese Einschaltzeit ergibt sich aus der Differenz von FFFF und Inhalt des jeweiligen Compare-Registers:

$$t_e = 65.536 - (\text{Inhalt des Compare - Registers})_{\text{dezimal}} \quad \text{ms}$$

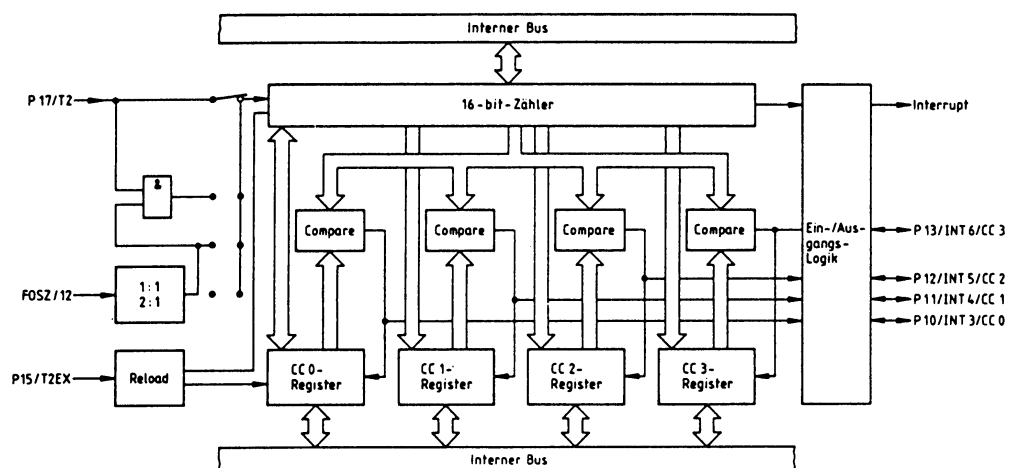


Abbildung 1: Ausschnitt der Architektur des SAB 80C535 (Timer 2, Reload- und 3 Compare-Register)

Das durch diesen Timer entstehende Pulsmuster (Rechteck) nutzen wir zur Ansteuerung des Gleichstrommotors, der uns von Seiten der Firma vorgegeben war. Das an Port 1 anliegende Signal wird durch eine Treiberstufe noch verstärkt und dann direkt dem Motor zugeführt.

Integrierte serielle Schnittstelle

Die integrierte serielle Schnittstelle bietet vier verschiedene Modi, Voll-Duplex-Betrieb und ist in Port 3 des μ P integriert. Bei der μ P-Taktfrequenz von 11.0592 MHz ergibt sich für die Schnittstelle im Mode 1 eine Geschwindigkeit von 19.200Bd.

$$\text{Baudrate} = \frac{2^{SMOD}}{32} * \frac{f_{osc}}{12 * (256 - TH1)} = \frac{2}{32} * \frac{11,0592 \text{ MHz}}{12 * (256 - 253)} = 19.200 \text{ Bd}$$

Die Schnittstelle übernimmt die Kommunikation zwischen Steuerplatine und PC. So kann das Steuerprogramm auf dem PC, und damit der Benutzer, verschiedene Prozeßzustände überwachen und kontrollieren.

Integrierte 8-Bit I/O-Ports

Die zahlreichen digitalen Ein- und Ausgabeports wurden zum Erfassen digitaler Meßwerte und zum Steuern von Schaltern genutzt. So werden mit den Ports 4 und 5 über einen zwischengeschalteten Treiberbaustein die Durchflußventile für Fixierer und Spülung geschaltet, der Schwenkarm für die Flüssigkeiten auf seine Position gefahren und der Unterdruck an der Rotationsachse über einen digitalen Unterdruckgeber geprüft.

Integrierte A/D-Wandler Ports

Die A/D-Wandler Ports (Port 6) wurden zur Auswertung von analogen Größen genutzt. So wertet die Hardware die Motorstromaufnahme, die Tachogeneratorspannung und die Trafotemperaturen aus. Die Motorstromaufnahme und die Trafotemperatur dienen nur der Kontrolle einer Überlastung, wobei die Auswertung der Tachogeneratorspannung eine der wichtigsten Größen im Prozeß ist. Sie wird in einen digitalen Wert umgewandelt und mit dem Sollwert der Drehzahl verglichen. Auf diese Art ist die Regelung der Motordrehzahl realisiert.

Weitere Besonderheiten des SAB 80C535

Port 0 und Port 2 sind für den Daten- und Adreßbus des externen Speichers reserviert. Auch bietet der Prozessor noch einige Stromsparfunktionen wie den Idle-Mode und den Power-Down-Mode.

Die drei integrierten Timer sind sehr wichtig und kommen auch alle drei zum Einsatz. Timer 0 wurde von uns zum Auslösen der Timerinterruptroutine zur Motorregelung genutzt, er besteht aus einem dekrementen Zähler, der immer beim Nulldurchgang einen Interrupt, und damit die entsprechende Routine auslöst. Timer 1 stellt den Reload-Wert

für die serielle Schnittstelle fest und gibt auch einen zeitlichen Takt, und damit die Baudrate vor. Mit dem Reload-Wert von Timer 2 stellen wir den Grundtakt (4 kHz) für das PWM-Signal ein.

3.3. Steuer - PC

Der in dieser Arbeit verwendete PC ist ein handelsübliches Gerät mit Pentium-Prozessor mit einer Taktfrequenz von 166 MHz und MMX Technologie. Installiert sind 32 MB Hauptspeicher (RAM) und eine Festplatte mit einer Speicherkapazität von 1,7 GB. Auf dem Motherboard sind eine parallele (Centronics) und zwei serielle (RS232) Schnittstellen integriert, die die hardwareseitige Kommunikation mit der Steuerplatine ermöglicht. Die PCI-Grafikkarte arbeitet nach dem SVGA-Standard und ist mit einem Chipsatz des Typs S3 ausgestattet. Der eingesetzte VGA-Monitor hat eine Bild-diagonale von 15 Zoll.

Der Rechner wird mit dem Betriebssystem Windows 95 der Firma Microsoft betrieben, welches auch die Netzwerkkommunikation mit der 100Mbit Netzwerkkarte von 3Com (Fast Etherlink 3c905B-TX) im Windows NT-Netzwerk übernimmt.

4. Steuer- und Bediensoftware

4.1. Allgemeines

Die Steuer- und Bediensoftware soll - wie schon in der Aufgabenstellung (Seite 6) beschrieben - einfach zu bedienen sein und während des Prozeßablaufes den momentanen Prozeßzustand anzeigen.

Die Benutzeroberfläche bekommt ein Menü, über welches alle Funktionen des Programms erreichbar sind. Weiter gibt es Schaltflächen, mit denen man den Prozeß starten und stoppen kann. Die Eingabe der Prozeßdaten soll über separate Eingabefenster erfolgen und über die Windowsdialogfenster kann man Projekte speichern, öffnen und drucken.

Die Programm bietet die Möglichkeit, bis zu acht Drehzahl/Zeit-Paare einzugeben. Jedes dieser Paare hat ein bestimmtes Ergebnis während der ablaufenden Zeit auszuführen. Weiter sollen möglichst viele Parameter variabel sein, um spätere Änderungen an der Hardware möglichst einfach anpassen zu können.

Zur Realisierung eines solchen Projektes bietet Windows sehr viele Funktionen. Um z.B. den Prozeßstatus anzuzeigen, habe ich mich für einen herkömmlichen Windows-Statusbalken entschieden.

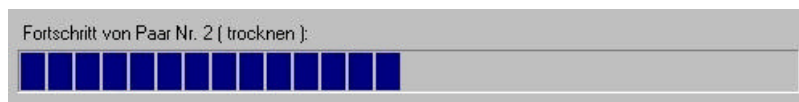


Abbildung 2: Windows-Statusbalken während eines Prozesses

Dieser Balken existiert in Windows bereits, muß praktisch nur noch in seiner Form, Optik und seinen Eckdaten konfiguriert werden.

4.2. Programmiersprache

Da ich mich vor einiger Zeit bereits intensiver mit Turbo Pascal (im Folgenden mit TP bezeichnet) unter MS-DOS beschäftigt hatte, wollte ich gerne die Steuer- und Bediensoftware damit programmieren. Mir waren die Befehle und die Programmierstruktur bekannt und unter TP kann man recht einfach direkt an der Hardware programmieren. Was mir allerdings erst im Laufe der Entstehung des Steuerprogramms klar wurde, ist die Tatsache, daß es recht aufwendig ist, unter TP eine übersichtliche, leicht bedienbare Benutzeroberfläche zu programmieren. Die Oberfläche grafisch aufzubereiten oder mit kleinen Diagrammen zu versehen, ist ebenso schwierig. Auch wurde der Entschluß, die Rechner der Steuerung mit zu vernetzen, erst später gefaßt. Nachdem nun als Betriebssystem Windows 95 vorhanden war und die Software auch auf Netzlaufwerke zugreifen sollte, beschloß ich, nachdem ich bereits unter TP einen

recht großen Teil programmierte hatte, das Steuer-Programm unter Windows neu zu programmieren.

Zur Programmierung unter Windows entschied ich mich aus Kostengründen für Visual Basic 5.0 von Microsoft (im Folgenden mit VB bezeichnet). Dieses Programm war vorhanden und bietet alle notwendigen Programmiermöglichkeiten zur Realisierung des Projektes.

VB in der Version 5 hält auch einige interessante Programmierhilfen bereit. So bekommt man zum Beispiel während des Schreibens eines Befehls diesen in seiner Anwendung erläutert. Das ist besonders hilfreich, wenn man eine Programmiersprache neu erlernt und sich noch in viele Bereiche einarbeiten muß. Besonders die grafische Gestaltung ist unter Windows doch stark vereinfacht. Man muß natürlich auch hier alle Verknüpfungen und Operationen programmieren, braucht sich aber z.B. keine Gedanken um die Gestaltung eines Schaltknopfes zu machen, man kann ihn einfach in Größe und Form definieren und eine Farbe oder einen Hilfetext zuweisen. So kann man seine Konzentration mehr auf die Steuer- und Auswertprogrammierung legen. Nachdem die Entscheidung für Windows 95 gefallen war, und heute Software mit 32bit als Standard eingesetzt wird, programmierte ich auch unsere Software unter einer 32-bit-Umgebung.

4.3. Aufgabe

Das Programm sollte eine benutzerfreundliche, übersichtliche Oberfläche haben und sollte komplett mit der Tastatur bedienbar sein. Es sollte die Möglichkeit bieten 8 Drehzahl/Zeit-Paare zu erfassen, diese unter einem Projektnamen zu speichern und den Fertigungsprozeß in einem Protokoll grafisch und in Text auszudrucken. Weiter soll man während des Prozeßablaufes den aktuellen Status angezeigt bekommen, so daß der Bediener immer verfolgen kann, in welchem Teilabschnitt sich der Prozeß befindet.

4.4. Funktionsbeschreibung

4.4.1. Das Menü

Das Menü des Hauptprogramms (Flußdiagramm Abbildung 4 auf Seite 17) sieht wie folgt aus. Die einzelnen Menüpunkte werden in Abhängigkeit vom Programmzustand aktiviert oder deaktiviert.

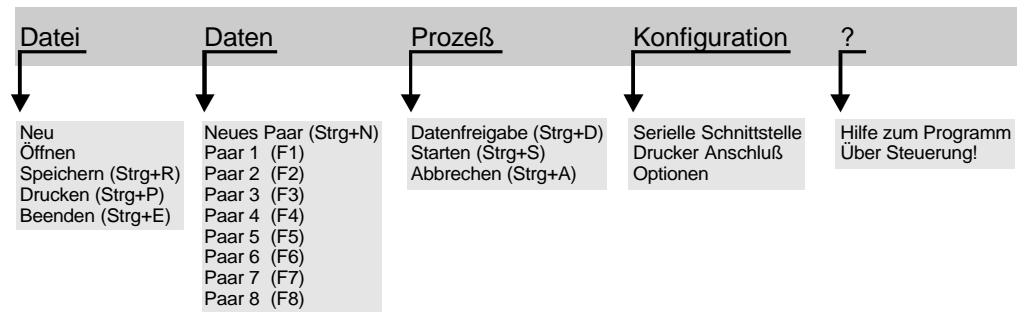


Abbildung 3: Menüstruktur der Steuerprogramms

Menüpunkt "Datei":

☒ "Neu"

Hier werden alle Werte auf die Anfangswerte gesetzt. Sollten geänderte Daten vorhanden sein, so wird der Benutzer über ein Dialogfenster gefragt, ob er diese Daten speichern möchte.

☒ "Öffnen"

Hier öffnet sich das Dialogfenster "Öffnen..." vom Windows Betriebssystem.

☒ Speichern (Strg + R)

Hier öffnet sich das Dialogfenster "Speichern unter..." vom Windows Betriebssystem.

☒ "Drucken (Strg + P)"

Hier wird die Protokolldatei des letzten Prozesses auf den im Betriebssystem als "Standarddrucker" festgelegten Drucker ausgegeben.

☒ "Beenden (Strg + E)"

Hier wird das Programm beendet. Sollten geänderte Daten vorhanden sein, so wird der Benutzer über ein Dialogfenster gefragt, ob er diese Daten speichern möchte. Für diese Funktion gibt es auch einen Button im Programmfenster.

Menüpunkt "Daten":

☒ "Neues Paar (Strg + N)"

Hier wird immer das Eingabefenster geöffnet, um das nächste freie Wertepaar zu erfassen.

☒ "Paar 1 (F1)"

Hier wird das Eingabefenster für Paar 1 geöffnet.*

☒ "Paar 2 (F2)"

Hier wird das Eingabefenster für Paar 2 geöffnet.*

☒ "Paar 3 (F3)"

Hier wird das Eingabefenster für Paar 3 geöffnet.*

☒ "Paar 4 (F4)"

Hier wird das Eingabefenster für Paar 4 geöffnet.*

☒ "Paar 5 (F5)"

Hier wird das Eingabefenster für Paar 5 geöffnet.*

☒ "Paar 6 (F6)"

Hier wird das Eingabefenster für Paar 6 geöffnet.*

☒ "Paar 7 (F7)"

Hier wird das Eingabefenster für Paar 7 geöffnet.*

☒ "Paar 8 (F8)"

Hier wird das Eingabefenster für Paar 8 geöffnet.*

* Sollten sich zwischen dem gerade erfaßten Paar und dem letzten in der Liste ein oder mehrere leere Paare befinden, so wird das gerade erfaßte Paar automatisch an das Ende der Liste gestellt und bekommt die entsprechende Nummer.

Menüpunkt "Prozess":**☒ "Datenfreigabe Paar (Strg + D)"**

Hier werden die aktuellen Daten aus der Liste für den Prozeß freigegeben. Danach kann der Prozeß mit "Starten" im Menü oder mit der Taste "START" am Arbeitsplatz gestartet werden.

Für diese Funktion gibt es auch einen Button im Programmfenster.

☒ "Starten (Strg + S)"

Startet den Prozeßablauf nachdem er durch "Datenfreigabe" freigegeben wurde.

Für diese Funktion gibt es auch einen Button im Programmfenster.

☒ "Abbrechen (Strg + A)"

Bricht den laufenden Prozeß ab.

Für diese Funktion gibt es auch einen Button im Programmfenster.

Menüpunkt "Konfiguration":**☒ "Serielle Schnittstelle"**

Hier wird das Dialogfenster zur Einstellung der seriellen Schnittstelle geöffnet. Man kann allerdings lediglich den Port auswählen, die Parameter werden vom Programm vorgegeben, um keine Kommunikationsprobleme mit der Steuerplatine aufkommen zu lassen.

Diese Funktion ist durch eine Paßwort geschützt.

☒ "Drucker Anschluß"

Hier wird lediglich der Druckerport ausgewählt, auf den die zu druckenden Daten ausgegeben werden sollen.

Diese Funktion ist durch ein Paßwort geschützt.

☒ "Optionen"

Hier öffnet sich das Optionsfenster, in dem verschiedene Eckdaten und Korrekturwerte für den Prozeßablauf eingestellt werden:

- MaxDrehzahl (maximal einzugebende Drehzahl im Eingabefenster der Paare)
- MinDrehzahl (minimal einzugebende Drehzahl im Eingabefenster der Paare)
- serieller_Schleifenzähler (Zähler für das Timeout beim Warten auf serielle Daten)
- MaxChar (Größtes ASCII-Zeichen für die Drehzahlübertragung)
- Passwort (Paßwort für alle passwortgeschützten Dialogfenster)
- RELpause (Pause zwischen den Ereignissen Fixierung, Spülung und Schwenkarm)

- Tfaktor (Korrekturfaktor für die Trafotemperatur)
- Sfaktor (Korrekturfaktor für den Motorstrom)
- Drehzahlkorrekturwerte (Hier kann man das Drehzahlspektrum in beliebig kleine Bereiche teilen, um für diese Bereiche eine Drehzahlkorrektur anzugeben.)

Diese Funktion ist durch ein Paßwort geschützt.

Menüpunkt "?":

"Hilfe zum Programm"

Hier wird eine kleine Hilfedatei geöffnet, um die Funktionen des Programms zu erklären.

"Über Steuerung"

Hier stehen Informationen über Programmversion, Copyright und das Diplomprojekt.

Im Folgenden nun das Flußdiagramm zum Ablauf bei Betätigung eines Menüpunktes:

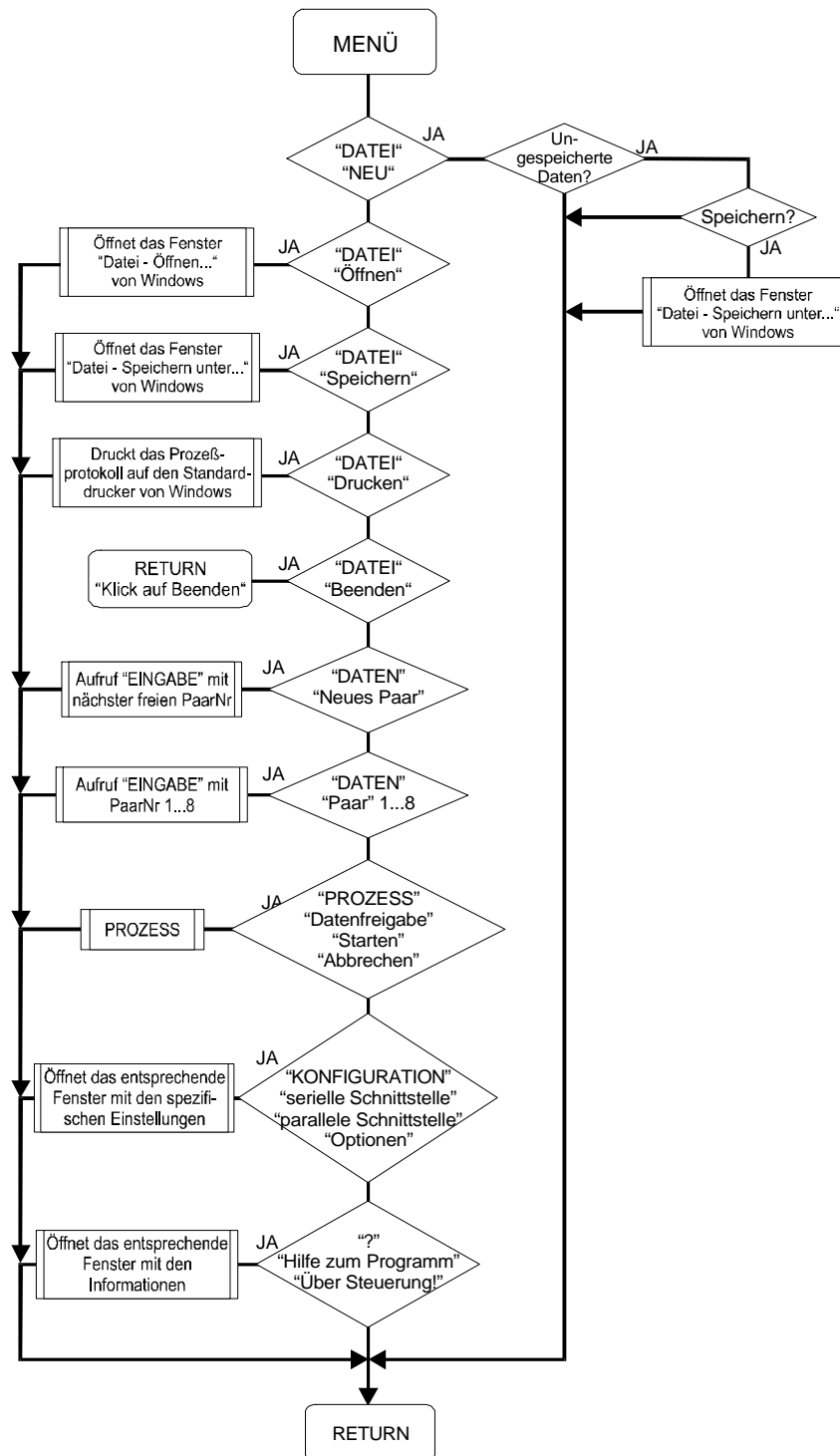


Abbildung 4: Programmfluß wenn ein Menü-Event eintritt

Wir haben am Steuer-PC eine spezielle Tastatur eingesetzt. Sie ist sehr klein und hat einen integrierten Trackball. Trotz des Trackballs sind alle Eingabefelder und Fenster mit der Tastatur zu erreichen, sind aber manchmal mit der Maus oder einem Trackball bequemer zu bedienen.

4.4.2. Programminitialisierung (Steuerung.cfg)

Nach dem Programmstart wird zunächst die Konfigurationsdatei gelesen und es werden die globalen Variablen und Konstanten definiert.

Die Konfigurationsdatei "steuerung.cfg" enthält folgende Parameter:

```
↑ 0          (Port der seriellen Schnittstelle)
↑ 0          (Port der parallelen Schnittstelle)
↑ "20"       (Timeoutzähler für Schnittstellenabfragen)
↑ "245"      (höchster Binärwert 0...255 der zur Hardware gesendet werden kann)
↑ "13000"    (max. Drehzahl)
↑ 1000      (min. Drehzahl)
↑ "kqms"    (verschlüsseltes Paßwort)
↑ "2"       (Relaisverzögerung in Sekunden)
↑ "0,5"     (Temperatur - Korrekturwert)
↑ 20        (Motorstrom - Korrekturwert)
↑ 50        (Anzahl der Korrekturpaare für den Drehzahlbereich)
↑ 260,1     (1. Wertepaar)
↑ 520,1     (2. Wertepaar)
↑ .....
    Drehzahl, Korrekturwert (.. gilt bis zur nächsten Drehzahl)
↑ .....
↑ 12740,1   (49. Wertepaar)
↑ 13000,1   (50. Wertepaar)
↑ "Ende"    (Ende der CFG-Datei)
```

(Die kursiven Texte in Klammern sind lediglich Erläuterungen und stehen nicht in der Datei!)

Diese Datei kann sowohl aus der Steuer-Software heraus bearbeitet werden, als auch mit einem normalen Editor. In diesem Fall muß man natürlich die genaue Bedeutung aller Parameter kennen. Sollte man dabei einen Parameter so verstellen, daß er keinen Sinn mehr ergibt, wird dies beim nächsten Programmstart signalisiert, und eine neue Standard-CFG-Datei wird erstellt.

Nachdem nun die Parameter eingelesen wurden, werden die Startwerte festgelegt, und die serielle Schnittstelle initialisiert. Alle veränderlichen Startwerte stehen in der CFG-Datei wie der serielle und parallele Port auch. Weitere Schnittstellenparameter, außer der Portnummer, kann man nicht einstellen, da die Schnittstellen eine bestimmte Konfiguration benötigen, die nicht verändert werden darf.

4.4.3. Das Programmhauptfenster

Danach wird das Hauptfenster (Abbildung 6) initialisiert und die Steuer-Software ist somit einsatzbereit.

Zunächst muß man sagen, daß Windows-Programme keine Programme sind, die einem strengen Programmablauf folgen, sondern ereignisgesteuert sind.

D.h., wenn man eine Taste betätigt oder die Maus bewegt (also ein Ereignis eintritt), werden Programmteile gestartet, die entsprechend vorher für genau dieses Ereignis definiert wurden.

Auch gibt das Programm zwischendurch immer Ressourcen an das Betriebssystem frei, damit auch andere, parallel laufende Programme weiter arbeiten können. In besonders kritischen Programmteilen sperrt man genau diese Ressourcen, um zu vermeiden, daß genau in diesem Moment ein anderes Programm der eigenen Prozedur Ressourcen wegnimmt oder sperrt.

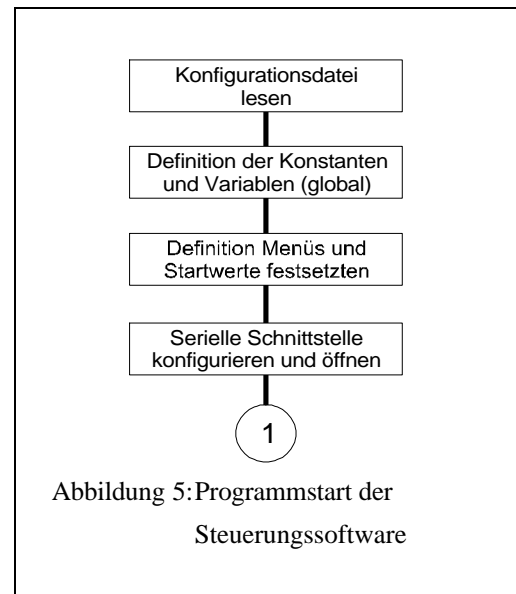


Abbildung 6: Das Programmhauptfenster

Das Hauptfenster des Steuerungsprogramms zeigt auf den ersten Blick alle wichtigen Daten, wie die 8 Drehzahl/Zeit-Paare und die Projektbeschreibung. Zwischen den Buttons "Datenfreigabe" und "Beenden" wird noch die Verbindung zur Hardware signalisiert. Wenn die LED grün leuchtet, wurde die Schnittstelle erfolgreich geöffnet. Im oberen Bereich, in dem grünen Balken, werden verschiedene Hardwarestaten wie Trafotemperatur und Motorstromaufnahme angezeigt.

Im Folgenden (Abbildung 7) sieht man das Flußdiagramm des Hauptfensters. Es sind nur einige Funktionen, die nach dem Start des Programms sinnvoll sind, freigegeben.

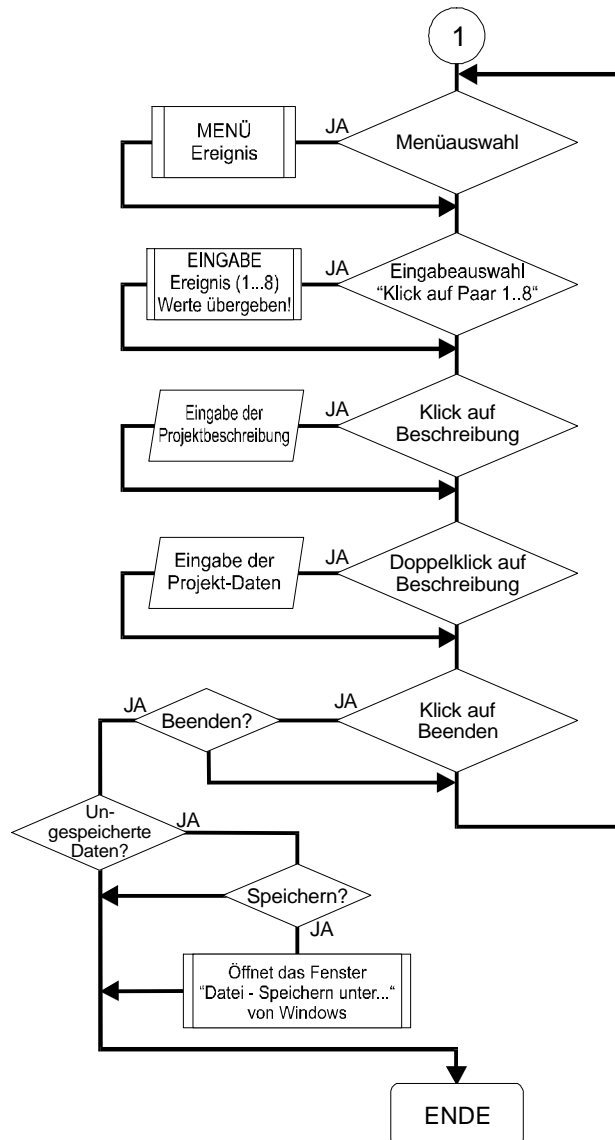


Abbildung 7: Programmablauf der Steuerungssoftware

So ist z.B. der Menüpunkt "Speichern" beim Neustart inaktiv. Er wird erst aktiv, wenn ein neues Datenpaar erfaßt wurde, oder in einem geöffnetem Projekt Daten verändert wurden.

Der Menüpunkt und Button "Datenfreigabe" wird aktiv, wenn ein gültiges Projekt vorhanden ist.

Beim Klick auf "Beenden" wird zunächst ein Dialogfenster mit der Frage "Wollen Sie wirklich beenden?" geöffnet. Wenn man dieses mit "Ja" bestätigt, wird geprüft, ob noch ungespeicherte Daten existieren. Wenn ja, wird automatisch ein Dialogfenster geöffnet, in dem gefragt wird, ob man die Daten noch speichern möchte, erst dann wird das Programm wirklich beendet.

Die Eingabe der Projektbeschreibung und der Projektdaten wird in Kapitel 4.4.6 auf Seite 23 erläutert.

4.4.4. Ein neues Projekt

Nach dem Programmstart befindet man sich im Hauptfenster der Software und hat zwei Möglichkeiten, ein Projekt zu erstellen. Entweder öffnet man ein schon bestehendes Projekt, oder man erfaßt eine neues Projekt. Wenn man ein bestehendes Projekt öffnen möchte, klickt man im Menü auf "Datei" "Öffnen...". In diesem Fall öffnet sich das Dialogfenster "Öffnen..." vom Windows Betriebssystem (Abbildung 8).



Abbildung 8: Windows - Dialogfenster "Öffnen..."

In diesem Fenster kann man nun über das Pulldown-Menü "Suchen in:" den Pfad festlegen, in dem sich die Projektdateien befinden. Die Endung "*.dat" ist sowohl beim Öffnen als auch beim Speichern eines Projektes vorgegeben.

4.4.5. Eingabe der Daten

Im anderen Fall erfaßt man ein neues Projekt. In einem Projekt stehen immer 2 bis 8 Drehzahl/Zeit-Paare, eine Projektbeschreibung, und wenn man auf die Projektbeschreibung doppelt klickt, kann man noch spezifische Daten zum Kunden und Projekt erfassen.

Die Drehzahl/Zeit-Paare bestehen immer aus einer Drehzahl in 100-Umdrehungen, einer Zeit in Sekunden und einem Ereignis. Sie werden in einem speziellen Fenster (Abbildung 9) erfaßt.

Die Drehzahleingabe muß zwischen 6 und 150 (x100) liegen. Beim Bestätigen des Feldes mit "ENTER" (⏏) wird der Wert geprüft. Sollte er außerhalb des Bereiches sein,

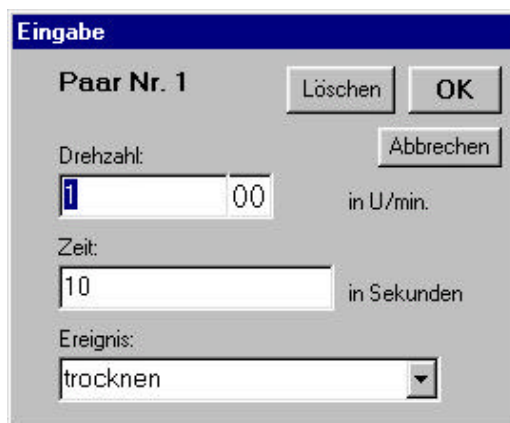


Abbildung 9: Fenster zur Eingabe der Drehzahl/Zeit-Paare

so wird er auf den minimalen oder maximalen Wert gesetzt, und ein Signalton ertönt.

Die Zeit, wie lange die vorher eingegebene Drehzahl gehalten wird, muß zwischen einer und 100 Sekunden liegen. Auch dieser Wert wird beim verlassen des Feldes geprüft und gegebenenfalls korrigiert.

Das Ereignis, welches während dieser Zeit eintritt, (Fixierer, Spülung oder Trocknen) muß in dem PullDown-Menü gewählt werden. Wenn das Ereignis auf

"keine" stehen bleibt, ist das Drehzahl/Zeit-Paar ungültig und wird nicht in das Hauptfenster übernommen.

Das Eingabefenster für die Drehzahl/Zeit-Paare startet mit dem Cursor im Drehzahlfeld, nach einer Eingabe betätigt man "ENTER" (□) oder "TAB" (●). Der Cursor springt dann in das Zeitfeld. Nach dieser Eingabe und "ENTER"/"TAB" kann man im Ereignisfeld mit den Pfeiltasten (□£) noch ein Ereignis auswählen. Nach dem Bestätigen der OK-Taste mit "ENTER", wird das Drehzahl/Zeit-Paar in den Hauptdialog übernommen.

Wenn man ein Drehzahl/Zeit-Paar löschen möchte, muß man lediglich "Löschen" betätigen, d.h. mit der Maus anklicken oder mit "ENTER" bestätigen, und das Paar wird aus dem Hauptdialog entfernt. Anwahl der Buttons "Abbrechen" übergibt die ursprünglichen Werte ins Hauptfenster. Das Flußdiagramm des Eingabefensters ist in Abbildung 12 zu sehen.

Hier kann man alle erfaßten Daten sehen und auch nachbearbeiten. Zum Nachbearbeiten klickt man einfach auf einen dem Drehzahl/Zeit-Paar zugehörigen Wert oder drückt die entsprechende (F)unktions-Taste und das Eingabefenster öffnet sich mit den zugehörigen Daten.

Paar	Drehzahl	Zeit	Ereignis	Drehzahl OK
F1	100 U./min.	10 Sekunden	trocknen	●
F2	4.000 U./min.	3 Sekunden	trocknen	●
F3	5.000 U./min.	3 Sekunden	trocknen	●
F4	6.000 U./min.	3 Sekunden	trocknen	●
F5	7.000 U./min.	3 Sekunden	trocknen	●
F6	8.000 U./min.	3 Sekunden	trocknen	●
F7	10.000 U./min.	3 Sekunden	trocknen	●
F8				

Abbildung 10: Fensterausschnitt mit den Drehzahl/Zeit-Paaren

Die roten LED's visualisieren den Drehzahlzustand. Während des Prozeßablaufes sind sie zunächst auch rot, erst wenn die jeweilige Drehzahl erreicht ist, wird die LED grün. Sollte der Prozeß an einer beliebigen Stelle unter-

Paar	Drehzahl	Zeit	Ereignis	Drehzahl OK
F1	100 U./min.	10 Sekunden	trocknen	●
F2	4.000 U./min.	3 Sekunden	trocknen	●
F3	5.000 U./min.	3 Sekunden	trocknen	●

Abbildung 11: Abbruch eines Prozesses (graues Feld)

brochen werden, so bleibt die

aktuelle Zeile grau, und man kann in einem Prozeßprotokoll nachschauen, an welcher Stelle genau der Abbruch erfolgt ist. Man kann auf Grund der grünen LED schon sagen, daß die Drehzahl auf jeden Fall erreicht war und der Abbruch danach statt fand. Das Protokoll mit dem detaillierten Prozeßablauf kann man auch ausdrucken.

Im folgenden Flußdiagramm (Abbildung 12) ist der Ablauf im Eingabefenster für die Drehzahl/Zeit-Paare noch einmal beschrieben. Beim Aufrufen des Fensters werden alle Werte, die zu einem Paar gehören, in das Eingabefenster übergeben (bei einem neuen Paar "Null"), und am Ende, unmittelbar bevor das Fenster geschlossen wird, werden sie wieder an das Hauptfenster zurück gegeben.

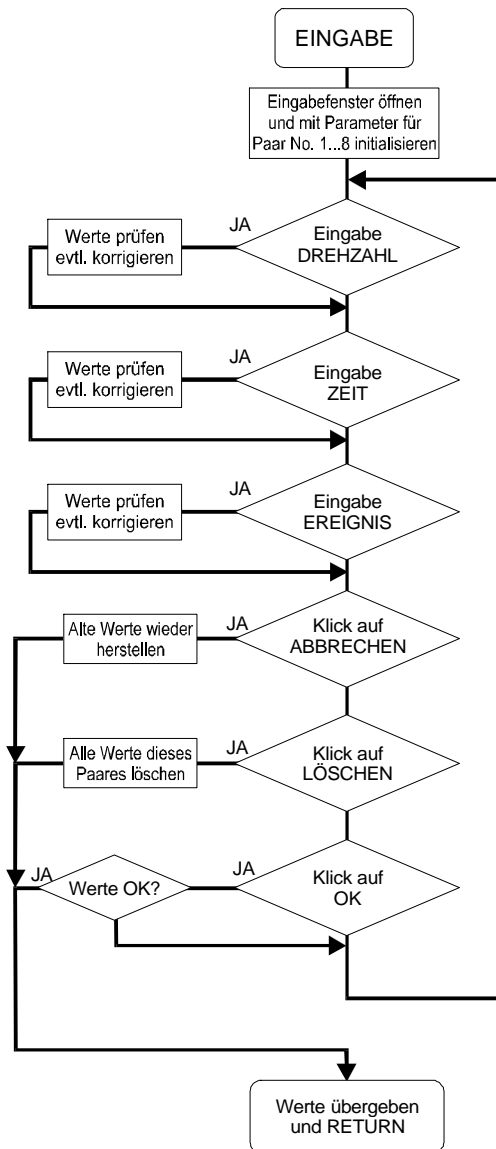


Abbildung 12: Programmfluß des EINGABE-Fensters

4.4.6. Die Projektbeschreibung und die Projektdaten

Die Projektbeschreibung ist ein einfaches Textfeld, das in seiner Größe nicht eingeschränkt ist.

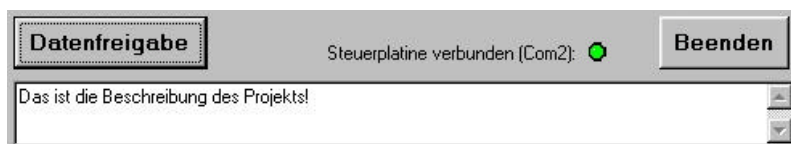


Abbildung 13: Textfeld für die Projektbeschreibung

Hier kann man eine beliebig lange Texteingabe vornehmen, um das aktuelle Projekt zu beschreiben. Während der Eingabe werden die Zeilen automatisch umgebrochen. Wenn man einen Absatz möchte, kann man diesen mit "ENTER" erzwingen.

Bei den spezifischen Projektdaten werden z.B. Kundennummer, Adresse, Projekt-
nummer und Erstellungsdatum erfaßt. Diese Daten sind sehr wichtig, um später ein
Projekt wieder eindeutig zuordnen zu können.

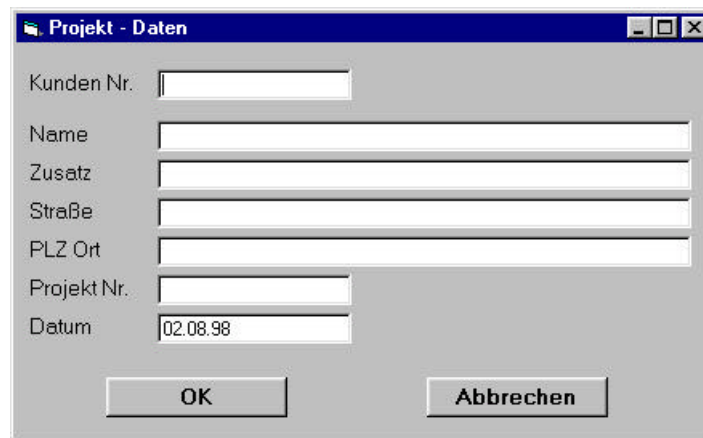


Abbildung 14: Fenster zur Eingabe der projektspezifischen Daten

Dieses Fenster erreicht man, in dem man auf die Projektbeschreibung im Hauptfenster
doppelt klickt. Wie man oben schon sieht, werden hier noch einige sehr wichtige Daten
erfaßt, die auch zu jedem Projekt eingegeben werden sollten.

In den Feldern wechselt man mit der TAB-Taste (●) und bestätigt die Eingaben am
Ende mit "OK". Sie werden beim Speichern mit in die Projektdatei geschrieben.

4.4.7. Die Optionen und die Schnittstellen

Über das Menü sind auch die Optionen zu erreichen.

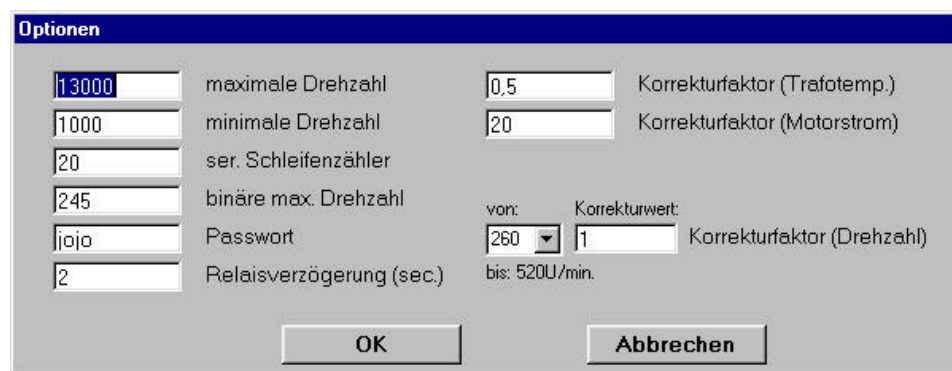


Abbildung 15: Fenster zur Eingabe der Optionen

Die Optionen dienen dem Administrator zur Anpassung an den Prozeß. Wenn sich
Änderungen an der Hardware, wie z.B. dem Motor ergeben, kann es notwendig sein, die
Drehzahl in bestimmten Bereichen zu korrigieren. Da diese Werte direkt in den Prozeß
eingreifen, sollten sie nur von Personen mit dem entsprechenden Hintergrundwissen

verändert werden. Aus diesem Grund sind sie auch durch ein Paßwort geschützt. Sie sind in der CFG-Datei (Seite 18) des Programms gespeichert.

Beim Anwählen des Menüpunktes öffnet sich zunächst ein Fenster, in dem man zur Eingabe des Paßwortes aufgefordert wird. Erst wenn dieses korrekt eingegeben wurde, öffnet sich das Dialogfenster für die Optionen.

Das gleiche Paßwortfenster öffnet sich zunächst, wenn man die serielle oder parallele Schnittstelle auswählen möchte.

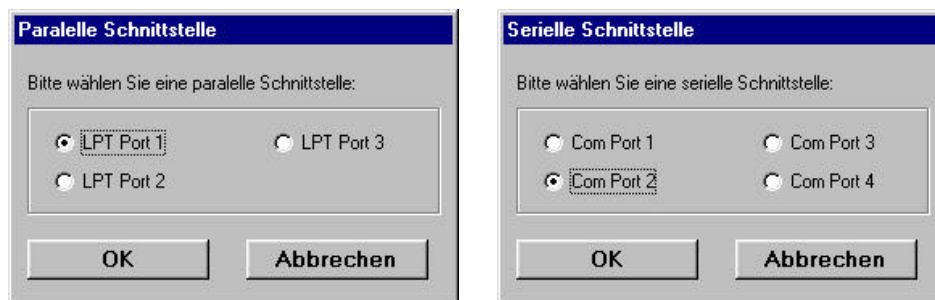


Abbildung 16: Parallele und serielle Schnittstellenfenster

In diesen beiden Fenstern klickt man nur auf den entsprechenden Port, es kann auch immer nur einer angewählt werden, und bestätigt es dann mit "OK".

4.4.8. Die Projektdatei

Die Projektdatei enthält alle Daten für ein Projekt: die Projektbeschreibung, die acht Drehzahl/Zeit-Paare und die spezifischen Projektdaten. Diese Datei muß genau wie die Konfigurationsdatei ein bestimmtes Format haben. Auch diese Datei kann man mit einem normalen Editor bearbeiten, wenn man das Format kennt und sich genau daran hält. Sollte ein Formatfehler vorliegen, so ist die Datei nicht mehr als Projektdatei verwendbar. Im Folgenden die Definition der Projektdateien:

```

↑ "Die Projektbeschreibung geht bis zum 'Gänsefüßchen' .
↑ Sie kann also über mehrere Zeilen...
↑ Gehen und endet hier!",           (3 Zeilen Projektbeschreibung)
↑ 0,100,"10",3                       (Paarnummer, Drehzahl, Zeit, Ereignis)
↑ 1,4000,"3",3                       (Paarnummer, Drehzahl, Zeit, Ereignis)
↑ 2,5000,"3",3                       (Paarnummer, Drehzahl, Zeit, Ereignis)
↑ 3,6000,"3",3                       (Paarnummer, Drehzahl, Zeit, Ereignis)
↑ 4,7000,"3",3                       (Paarnummer, Drehzahl, Zeit, Ereignis)
↑ 5,8000,"3",3                       (Paarnummer, Drehzahl, Zeit, Ereignis)
↑ 6,10000,"3",3                     (Paarnummer, Drehzahl, Zeit, Ereignis)
↑ 7,0,0,0                             (Paarnummer, Drehzahl, Zeit, Ereignis)
↑ "Kundennummer","Firma","Name", " Straße", "PLZ/Ort",
  "Projektnummer", "Datum"           (eine Zeile spezifische Projektdaten)

```

(Die kursiven Texte in Klammern sind lediglich Erläuterungen und stehen nicht in der Datei!)

4.4.9. Starten des Prozesses

Wenn das Projekt fertig erfaßt oder geöffnet wurde, werden automatisch der Menüpunkt und Button "Datenfreigabe" aktiv. Wenn man nun den Ablauf starten möchte, muß man die Daten über genau diesen Menüpunkt oder den Button freigeben. Nach der Freigabe wird zunächst die Funktion "Datenfreigabe" inaktiv und die Funktion "Starten" aktiv. Weiter wird der Steuerungsplatine seriell mitgeteilt, daß die Daten freigegeben sind und somit der "Start"-Taster am Arbeitsplatz aktiviert werden muß.

Danach hat man zwei Möglichkeiten den Prozeß zu starten:

- a) Man startet den Prozeßablauf über das Menü "Prozeß – Starten" oder den Button "Starten".
- b) Man startet den Prozeß über den "Start"-Taster, ein am Arbeitsplatz angebrachter Taster. Dieser "Start"-Taster ist an einem Port des μ P angeschlossen und legt diesen beim Drücken auf Masse. Der μ P sendet dann ein serielles Zeichen an den PC und dort wird dann der Prozeßablauf gestartet.

Nachdem der Prozeßablauf aktiviert wurde, wird zunächst die Funktion "Starten" inaktiv und die Funktion "Abbrechen" aktiv. An der Steuerungsplatine wird der "Start"-Taster deaktiviert und der "Stop"-Taster aktiviert.

Wenn der Prozeßablauf aktiviert wurde, läuft der eigentliche Prozeß ab. Er kann aber jeder Zeit mit dem "Stop"-Taster am Arbeitsplatz oder der Funktion "Abbrechen" am PC unterbrochen werden.

Dieser durch den Benutzer ausgelöste Abbruch wird genau wie ein automatischer Abbruch (bei Fehlern) behandelt (Abbildung 11 auf Seite 22).

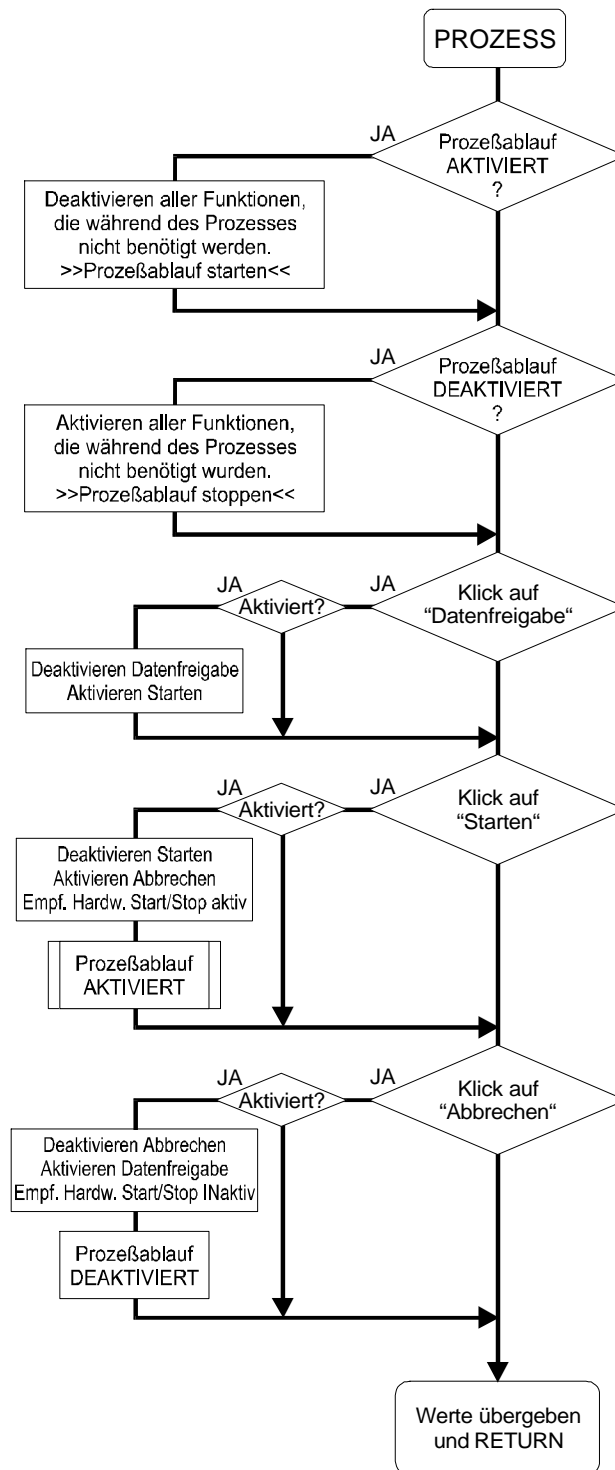


Abbildung 17: Programmfluß beim Freigeben der Prozeßdaten

Wenn der Prozeß abgebrochen wurde, sei es vom Programm oder dem Benutzer, so wird zunächst die Funktion "Abbrechen" und der "Stop"-Taster deaktiviert. Danach wird die Funktion "Datenfreigabe" wieder aktiviert.

Nun hat man die Möglichkeit, die Daten des Prozesses erneut freizugeben, oder sich das Fehlerprotokoll auszudrucken.

Der interne Ablauf des Prozesses:

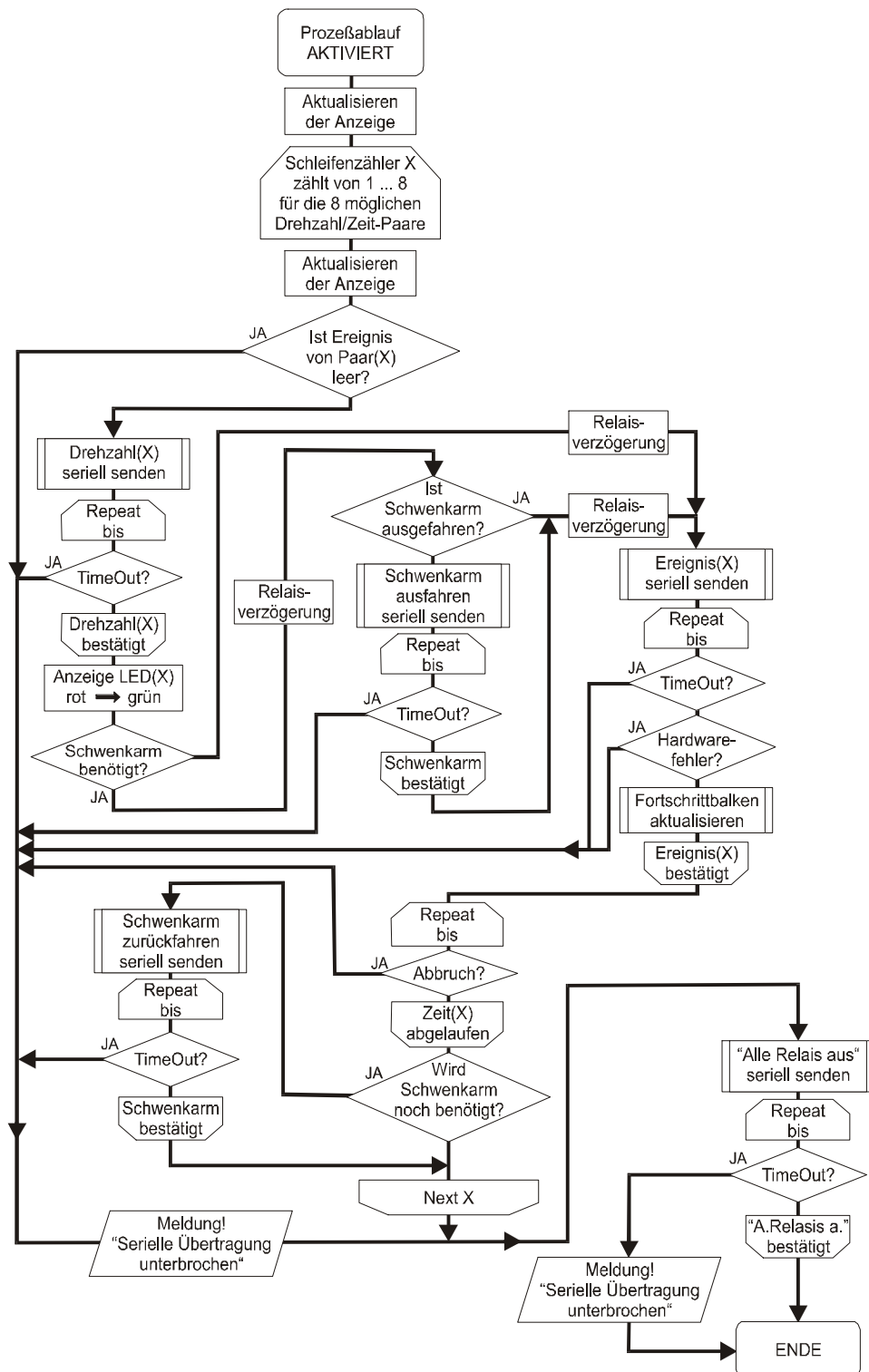


Abbildung 18: Programmfluß während des Prozeßablaufes

Nach Aktivierung des Prozeßablaufes wird dieser gestartet. Zunächst werden die Drehzahl/Zeit-Paare geprüft und es wird die Anzeige aktualisiert.

Danach beginnt eine Schleife, die von 1 bis zur Anzahl der gültigen Drehzahl/Zeit-Paare läuft. Innerhalb dieser Schleife läuft dann der gesamte Prozeß ab.

Zunächst wird die Prozedur für das erste Paar durchlaufen:

- Paar initialisieren -

Die Werte des Fortschrittbalkens werden errechnet und seine Anzeige wird auf "null" gesetzt. Nun wird die Zeile des ersten Paares rot unterlegt und die Drehzahl des ersten Paares seriell zur Steuerplatine gesendet. Nach Erhalt der Drehzahl beschleunigt die Regelsoftware den Motor bis die vorgegebene Drehzahl erreicht ist und dann wird seriell eine Bestätigung zum PC gesendet.

- Schwenkarm positionieren -

Erhält die Steuersoftware diese Bestätigung nicht innerhalb einer bestimmten Zeit, wird von einem Fehler ausgegangen, und der Prozeß wird abgebrochen. Nach Erhalt der Bestätigung wird nun geprüft, ob der Schwenkarm benötigt wird. Bei Bedarf wird er wieder ausgefahren und nach einer definierbaren Verzögerungszeit wird das entsprechende Flüssigkeitsrelais durchgeschaltet. Sowohl beim Schwenkarm, als auch bei den Flüssigkeitsrelais, wird auf eine serielle Bestätigung von der Regelsoftware gewartet. Auch hier muß diese Bestätigung innerhalb einer bestimmten Zeit erfolgen, sonst wird der Prozeß abgebrochen.

- Zeit läuft -

Wurde das Flüssigkeitsrelais durchgeschaltet, beginnt in diesem Moment die eingestellte Zeit abzulaufen. Während dieser Phase wird nur der Fortschrittbalken aktualisiert und die Schnittstelle auf den Empfang eines eventuellen Fehlers oder eines Benutzerabbruchs geprüft.

- Abbruch -

Tritt ein Abbruch auf, wird der Motor gestoppt, alle Relais und der Schwenkarm werden in Ausgangsposition gebracht, und der Abbruch wird am PC signalisiert.

- Paar beenden -

Ansonsten wird nach Ablauf der Zeit das entsprechende Flüssigkeitsrelais geschlossen. Der Schwenkarm wird nur zurück gefahren, wenn er beim zweiten Paar nicht mehr benötigt wird. Sollte er noch benötigt werden, bleibt er auf der Position und wird erst dann zurückgefahren, wenn er im nächsten Paar nicht mehr benötigt wird.

- weitere Paare -

Auf die gleiche Art durchläuft nun jedes gültige Drehzahl/Zeit-Paar die Prozedur. Das letzte Paar wird in der Regel ein Paar sein, welches die Aufgabe des Trocknens hat, hier wird dann der Schwenkarm nach dem letzten Paar zurückgefahren. Bei einem solchen Paar (Trocknen) beginnt die Zeit nach der Drehzahlbestätigung zu laufen.

- Prozeßende -

Wenn alle Paare durchlaufen sind, werden aus Sicherheitsgründen erneut alle Ventile geschlossen, der Schwenkarm wird in die Ausgangsposition gefahren und der Motor bekommt ein Stop-Signal. Danach ist der Fertigungsprozeß beendet.

Nachdem der Prozeß komplett durchlaufen wurde und keine Fehler auftraten, wird dieses mit einer Meldung auf dem Bildschirm signalisiert. Diese Meldung muß vom Anwender bestätigt werden. Die Bestätigung des erfolgreichen Prozesses wird anschließend auch in die Protokoll-Datei aufgenommen.

Die folgende Abbildung zeigt das Hauptfenster während des Programmablaufes:

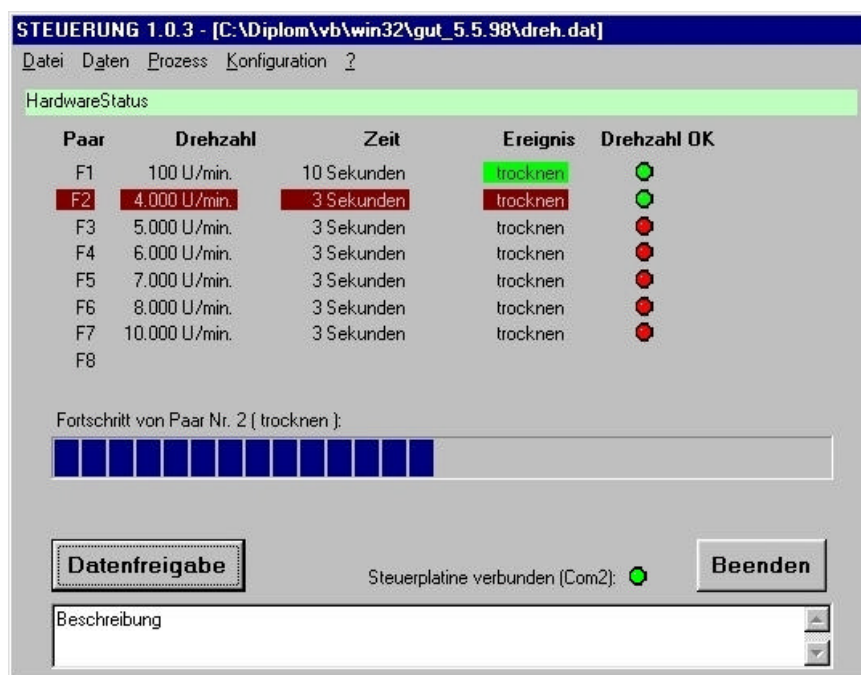


Abbildung 19: Das Hauptfenster während des Prozeßablaufes

Die grüne LED hinter dem "Hardware verbunden:" zeigt, daß die serielle Schnittstelle erfolgreich geöffnet wurde. Das grün markierte Ereignis (Paar 1) zeigt, daß dieses Paar bereits erfolgreich verarbeitet wurde. Der rote Balken hebt das Paar hervor, welches zur Zeit gerade abgearbeitet wird. Wie weit der Fortschritt bei diesem Paar ist, kann man an Hand des Fortschrittbalkens sehen. Die restlichen Paare (rote LED) müssen noch bearbeitet werden.

5.2.2. Direktadressierung

Direktadressierung ist die einzige Möglichkeit, die SFR anzusprechen. Die niederwertigen 128 Byte des internen RAM werden ebenso adressiert.

```
Assembler-Bsp.:  MOV    A,P4      ; (Port 4) ⇒ A
                  INC     30h     ; (30h) + 1 ⇒ 30h
```

Die SFR sind teilweise Direkt-Bit-adressierbar. Die Bits des Bereichs 1Fh bis 2Fh des internen RAM sind Bit-adressierbar.

```
Assembler-Bsp.:  SETB   P1.4     ; Bit 4 des Port 1 setzen (high)
                  CLR    P4.3     ; Bit 3 des Port 4 löschen (low)
```

5.2.3. Register-Indirekt-Adressierung

Mit der Register-Indirekt-Adressierung wird indirekt das interne RAM und der externe Datenspeicher selektiert. Diese Adressierungsart verwendet den Inhalt von R0 oder R1 der selektierten Registerbank als 8-Bit-Zeiger in einem 256-Byte Block. Das sind die 256 Byte des internen RAM, bei Verwendung des MOV-Befehls, oder die niederwertigen 256 Byte des externen Datenspeichers, bei Verwendung des MOVX-Befehls. In Verbindung mit dem MOVX-Befehl ermöglicht der 16-Bit Datenpointer (DPTR) den Zugriff auf den gesamten externen Datenspeicherbereich (max. 64 Kbyte). Der Zugriff auf die SFR ist nicht möglich. Die Befehle PUSH und POP benutzen ebenfalls die Register-Indirekt-Adressierung mittels Stackpointer (SP).

```
Assembler-Bsp.:  MOV     A,@R0    ; ((R0)) ⇒ A
                  MOVX   A,@R1    ; ((R1)) ⇒ A
                  MOVX   @DPTR,A  ; (A) ⇒ (DPTR)
```

5.2.4. Unmittelbare Adressierung

Die Unmittelbare Adressierung bezieht sich auf Konstanten, die Teil eines Befehls aus dem Programmspeicher sind. Die Konstante wird in ein angegebenes Ziel kopiert.

```
Assembler-Bsp.:  MOV     A,#27h   ; 27h ⇒ A
```

5.2.5. Indirekte, indizierte Adressierung mittels Basisregister plus Indexregister

Diese Adressierungsart erlaubt den Zugriff auf ein Byte des Programmspeichers, dessen Adresse aus der Summe des Basisregisterinhaltes (DPTR oder PC) und des

Indexregisterinhaltes (A) gebildet wird. Hiermit wird also der Zugriff auf Tabellen im Programmspeicher erleichtert.

Assembler-Bsp.: `MOVC A,@A+PC ; ((A) + (PC)) ⇒ A`
 `MOVC A,@A+DPTR ; ((A) + (DPTR)) ⇒ A`

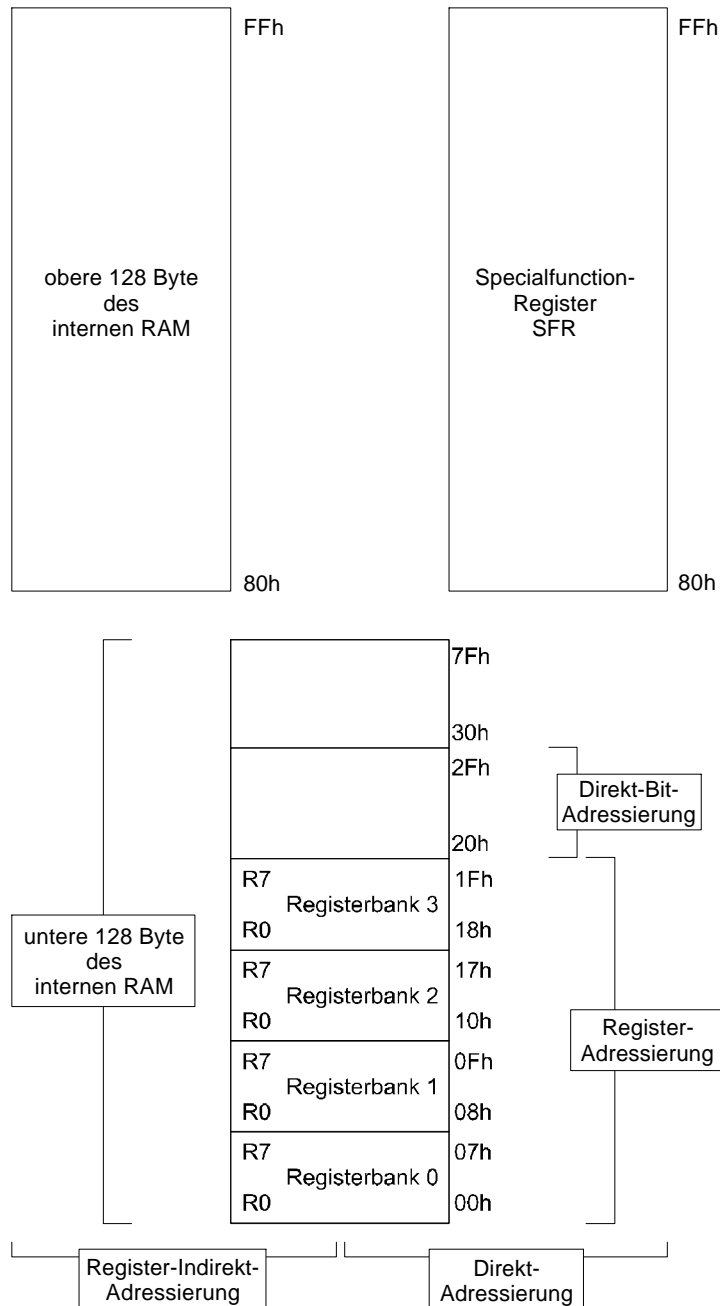


Abbildung 20: Adreßbereich des internen RAM

5.3. Regelungen

Mit der Verbreitung von digitalen Rechensystemen nicht mehr nur zur Bearbeitung von Daten im Verwaltungsbereich, sondern auch in der Prozeßautomatisierung, wurde es immer wichtiger, Prozeßdaten auch in digitaler Form verarbeiten zu können.

Da auch die entsprechenden Mikroprozessoren immer preisgünstiger wurden, lag es nahe, Regelaufgaben zunehmend von, für die Prozeßautomatisierung ohnehin vorhandenen, Rechnern übernehmen zu lassen. Es ist auch durchaus sinnvoll, dafür gesonderte Rechner einzusetzen. Diese bieten zusätzlich zu den mit digitalen Konzepten einhergehenden Vorteilen eine einfache und weitgehend störungsfreie Möglichkeit, beliebige Prozeßdaten von und zu Leitrechnern zu übertragen.

Charakteristisch für diese Regelkonzepte ist, daß sie nicht mehr mit analogen Prozeßwerten zeitgleich zum Prozeß operieren, sondern in diskrete digitale Werte abgebildete Prozeßgrößen "rechtzeitig" verarbeiten¹.

Die Lösungsmittel zur Führung industrieller Prozesse bzw. Anlagen sind Steuerungs- und Regeleinrichtungen. Das besondere Merkmal der Regelkreise ist ihr geschlossener Wirkungsweg, bei dem eine physikalische Größe (Istwert) fortlaufend oder in kurzen zeitlichen Abständen erfaßt und durch Vergleich mit einer anderen Größe (Sollwert) im Sinne einer Angleichung an diese beeinflußt wird².

Man kann Regelkreise auf verschiedene Arten (Transistorschaltung, OP-Schaltung, digital) aufbauen, in unserem Fall entschieden wir uns für einen digitalen Regelkreis.

Wie oben schon zum Teil erwähnt bieten die digitalen Regelkreise einige Vorteile:

- ☒ billige Fertigung, da programmierbare Standardschaltungen benutzt werden
- ☒ zunehmende Genauigkeit, bei linear ansteigenden Kosten
- ☒ hohe Zuverlässigkeit bei sich ändernden Umweltbedingungen
- ☒ die so erfaßten Daten stehen auch anderen Prozessen zur Verfügung

Als Nachteil ist anzuführen, daß wegen der Rechenzeiten digitale Regelsysteme nicht beliebig schnell arbeiten können.

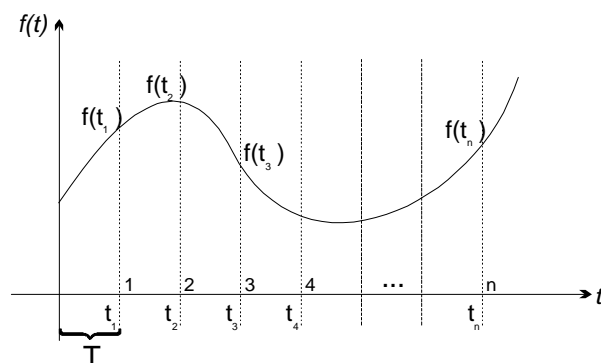


Abbildung 21: Abtastung eines Signals in zeitgleichen Zeitabschnitten

Zur Synchronisierung eines zu regelnden Prozesses mit der für die digitale Verarbeitung benötigten Zeit ist eine Abtastung erforderlich (Abbildung 21). Diese Abtastung erfolgt zweckmäßigerweise zu äquidistanten Zeitpunkten mit der Abtastperiode T . Die Abtastperiode wird einmal von der Komplexität des benutzten Regelalgorithmus und der Rechengeschwindigkeit des

¹ vgl.: Garbrecht, F.W., "Digitale Regelungstechnik", (Literaturverzeichnis (1))

² vgl.: Orlowski, Peter, "Praktische Regelungstechnik", (Literaturverzeichnis (2))

benutzten Digitalrechners vorgegeben.

In diesen Regelkreisen spielen einige Variablen und Konstanten eine große Rolle:

- ☒ w ist der Sollwert, der dem System in der Regel von außen zugeführt wird. Er gilt für die Regelung als Sollwert und sollte möglichst schnell erreicht und gehalten (geregelt) werden.
- ☒ x ist der aktuelle Wert am Ende des Regelkreises. Dieser Wert wird dem Eingang wieder zugeführt und dient als Vergleichswert, um die Abweichung (e) vom Sollwert zu ermitteln.
- ☒ e ist die Differenz aus w und x und somit die Abweichung vom Sollwert.
- ☒ p ist der Verstärkungsfaktor. Er wird in unserem Fall empirisch ermittelt. Man erhöht in einem Versuch p so lange, bis das System zu schwingen beginnt. Den so ermittelten p -Wert halbiert man und stellt ihn beim P-Regler als konstanten Wert ein.

	P	T_i	T_D
P - Regler	$0,5 * P_{krit}$	---	---
PI - Regler	$0,45 * P_{krit}$	$0,85 * T_{krit}$	---
PID - Regler	$0,6 * P_{krit}$	$0,5 * T_{krit}$	$0,12 * T_{krit}$

Tabelle 1: Standardwerte für die Reglervariablen

- ☒ T_i ist die Integrationskonstante, der Zeitwert, an dem die Funktion bei 63% des Maximums angekommen ist.

☒

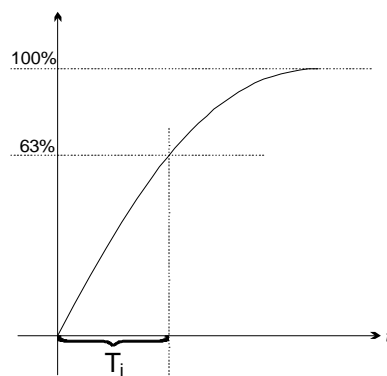


Abbildung 22: Bestimmung der Integrationskonstanten

- ☒ T_D ist die Differentiationskonstante, sie wird wie folgt ermittelt:

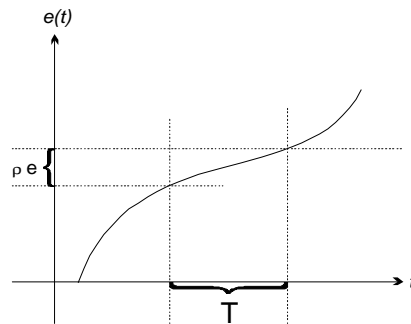


Abbildung 23: Bestimmung der Differentiationskonstante

Wie aus dem Schaubild hervorgeht, wird die Konstante T_D aus ρe und T bestimmt.

Die Formel lautet: $T_D = \frac{\Delta e}{T}$

☒ y ist die Stellgröße, die sich aus dem Produkt der Abweichung und der Verstärkung ($y = (w - x) * p$) ergibt.

Wie diese einzelnen Variablen und Konstanten zum tragen kommen, wird im Folgenden, in den drei Regelkreisen erläutert.

5.3.1. Der P-Regler

Der P-Regler³ (Proportional-) ist der einfachste Regelkreis und sieht wie folgt aus:

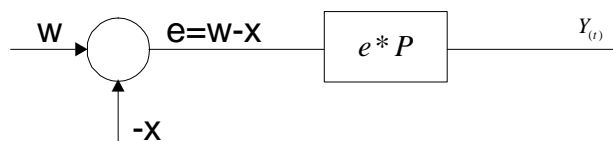


Abbildung 24: Der P-Regelkreis

So wird hier die ermittelte Abweichung lediglich mit einem konstanten Faktor (P) multipliziert und dem Ausgang zugewiesen. Die Formel für den P-Regler ist deshalb auch recht einfach:

Gleichung 1: $y_{(t)} = e_{(t)} * P$ mit $e_{(t)} = w_{(t)} - x_{(t)}$

³ Der P-Regler wird hier nur oberflächlich behandelt, näheres kann man in den im Literaturverzeichnis unter (3), (6) und (7) aufgeführten Büchern nachlesen.

Wird eine Strecke mit Ausgleich mit einem P-Regler geregelt, so sind alle Übertragungsglieder des Regelkreises stabil. Es gibt dann stets eine bleibende Regelabweichung.

5.3.2. Der PI-Regler

Der PI-Regler⁴ als Kombination aus P- und I-Regler (Integral-) vereint die Vorteile beider miteinander. Nach Auftreten einer Störung wirkt er zunächst mit seinem P-Anteil, während langfristig der I-Anteil überwiegt. Es gibt daher, wie auch beim I-Regler, wenn der Regelkreis stationäre Eingangsgrößen hat, keine bleibende Regelabweichung. Einen I-Regler könnte man allerdings nicht allein einsetzen, da er instabil ist. Bei konstanter Regelabweichung würde seine Stellgröße ständig anwachsen und abnehmen; dies wäre kein stationärer Zustand, der nur eintreten kann, wenn die Regelabweichung null ist.

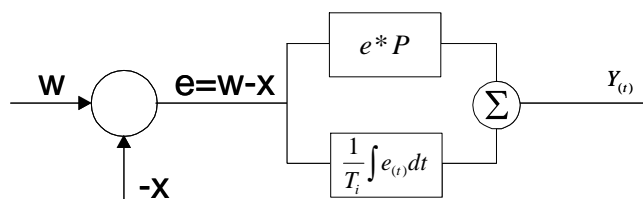


Abbildung 25: Der PI-Regelkreis

Aus der Übersicht ergibt sich folgende Formel für den PI-Regelkreis:

Gleichung 2:
$$y_{(t)} = e_{(t)} * P + \frac{1}{T_i} \sum e_{(t)}$$

Durch den Integralanteil wird der Programmieraufwand und somit auch der zeitliche Aufwand für diesen Regelkreis deutlich größer, was dazu führen würde, daß man die Zeitintervalle T (Abbildung 21, Seite 34) vergrößern müßte, was wiederum zur Folge hätte, daß man die ISR seltener aufrufen kann. Aus diesem Grund haben wir ihn auch nicht eingesetzt.

⁴ Der PI-Regler wird hier nur oberflächlich behandelt, näheres kann man in den im Literaturverzeichnis unter (3), (6) und (7) aufgeführten Büchern nachlesen.

5.3.3. Der PID-Regler

Der PID-Regler⁵ hat genau wie der I- und PI-Regler keine bleibende Regelabweichung (Gleiche Bedingung wie beim PI-Regler beschrieben). Außerdem macht der D-Anteil (Differential-) ihn zu einem schnellen Regler. Auch ein D-Regler allein würde keinen Sinn ergeben, er könnte einer Änderung der Regeldifferenz kurzzeitig entgegenwirken, gegen eine bleibende Regeldifferenz könnte er jedoch nichts ausrichten. Anders, als zusätzlicher D-Anteil in einem P- oder PI-Regler, weil sich damit neben der verminderten bzw. verschwindenden bleibenden Regeldifferenz, zusätzlich eine schnellere Reaktion auf Regeldifferenzänderungen erzielen lässt.

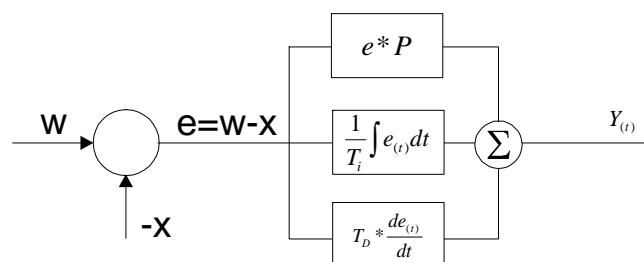


Abbildung 26: Der PID-Regelkreis

Die Formel für den PID-Regler:

Gleichung 3:
$$y_{(t)} = (e_{(t)} * P) + \frac{1}{T_i} * \sum e_{(t)} + T_D * \frac{de_{(t)}}{dt}$$

PID-Regler setzt man in Systemen ein, die extrem schnell und genau regeln müssen. Für unsere Ansprüche ist das nicht notwendig. Natürlich wird dadurch auch der Programmieraufwand größer, was die selben Folgen, wie unter dem PI-Regler beschrieben, hätte.

5.4. Programmiersprache

Der Prozessor arbeitet mit einem Maschinencode, welcher aus verschiedenen Maschinensprachen, aber auch aus Hochsprachen erstellt werden kann. Die wohl meist verbreitete Programmiersprache ist Assembler (ASM51). Wie man oben an den Programmbeispielen beim Adressieren sehen kann, ist es sehr komplex und unübersichtlich, direkt in Assembler zu programmieren. Aus diesem Grund entschied ich mich für die Hochsprache C, hier C51 für diese μ P-Familie. Das bringt einige

⁵ Der PID-Regler wird hier nur oberflächlich behandelt, näheres kann man in den im Literaturverzeichnis unter (3), (6) und (7) aufgeführten Büchern nachlesen.

Vorteile mit sich. Man hat die Möglichkeit, mit C sehr nahe an der Hardware zu programmieren und arbeitet aber trotzdem mit einer Hochsprache, so daß der Quelltext sehr übersichtlich und strukturiert aussieht. Das bringt den großen Vorteil mit sich, daß man die Funktionen des Programms recht leicht logisch durchdenken kann. Auch bei der Fehlersuche ist es sehr viel einfacher, einen Fehler zu finden. C51 verfügt als Hochsprache über eine Vielzahl schon vorhandener Routinen, für Ein-/Ausgabefunktionen, Berechnungen auch höherer Funktionen und Speicheroperationen. In der Bibliothek der Fachhochschule fand ich ein Buch (siehe (5) im Literaturverzeichnis) über Mikrocontroller, in dem sogar der C51-Compiler der Firma Keil auf einer CD-ROM beigelegt war. Man erstellt in einem gewöhnlichen Texteditor den Quelltext und kompiliert ihn dann zu einer Objektdatei, die danach mit einem Linker zum Maschinencode im Intel Hex-Format zusammengeführt wird. Dieser Maschinencode wird dann mit Hilfe einer Kommunikationssoftware zu einem Epromsimulator übertragen, mit dem ich nun das Programm, direkt auf der Hardware, testen kann. Der Epromsimulator war eine große Erleichterung, denn sonst hätte man jedesmal nach einer Programmänderung ein neues Eprom brennen müssen.

5.5. Aufgabe

Das Mikroprozessorprogramm hat vor allem die Aufgabe, die Motordrehzahl zu regeln, alle von der Hardware ermittelten Werte über die serielle Schnittstelle an den PC weiter zu leiten und die Vorgabewerte, die von dem PC kommen, auf die Hardware umzusetzen. Weiter sind einige Sicherheitsmechanismen eingebaut; wenn z.B. Unregelmäßigkeiten auftreten, die den Prozeß stören können, so wird der Prozeß automatisch abgebrochen und eine Meldung an den PC gesendet. Diese Meldung beinhaltet den Grund für den Abbruch und wird in einer Protokolldatei gespeichert.

5.6. Funktionsbeschreibung

5.6.1. Allgemeines

Zunächst der Programmablauf in der Übersicht:

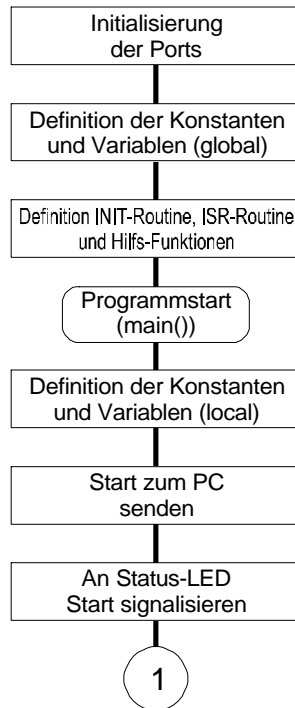


Abbildung 27: Programmstart der Regelungssoftware

Nach dem Programmstart werden zunächst alle Ports, globalen Variablen und globalen Konstanten definiert. Danach werden alle Funktionen und Prozeduren definiert. Jetzt beginnt das Hauptprogramm. Zunächst werden lokale Variablen definiert, danach blinkt die Status-LED 2-Mal und es werden drei Kontrollzeichen zum PC gesendet, um den Programmstart zu signalisieren. Dieser Programmstart kann nur mit dem Einschalten der Anlage oder mit einem Reset von seiten des μ P durchgeführt werden.

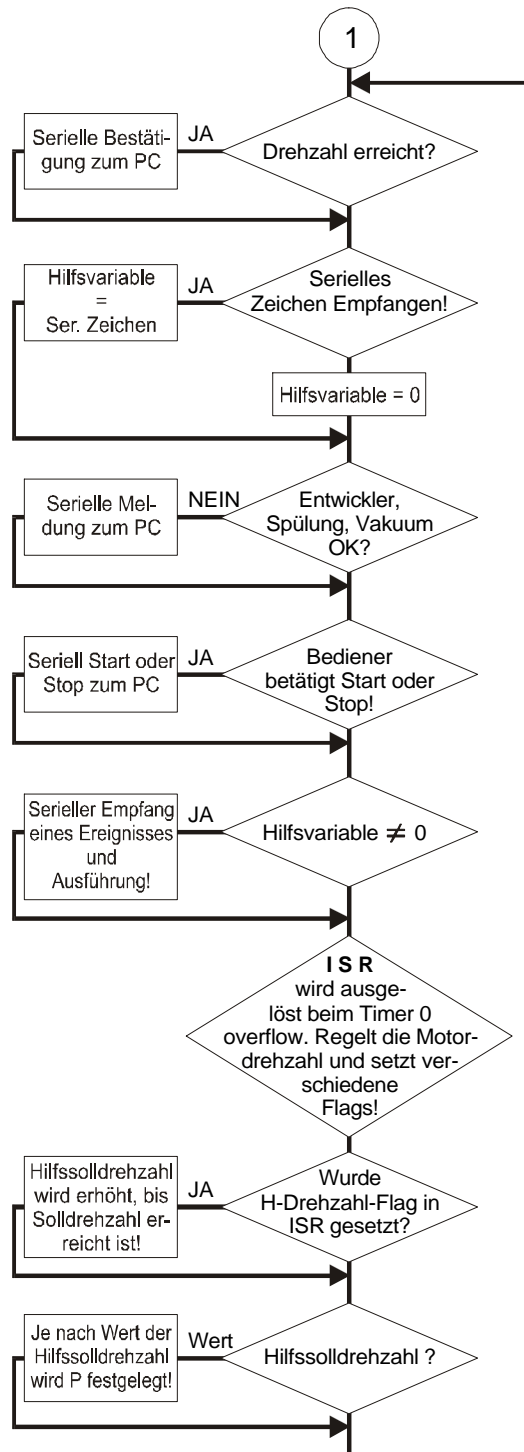


Abbildung 28: Die Endlosschleife der Regelungssoftware

Danach beginnt eine Endlosschleife, in der dann alle benutzten Eingabeports und die serielle Schnittstelle überwacht werden. Parallel dazu läuft ein Timer, welcher alle 10ms einen Interrupt auslöst und eine Interruptserviceroutine (ISR) startet. Der Zeittakt für diesen Interrupt ergibt sich aus dem Reloadwert des TH- und TL-Registers ($10ms = \frac{(10000h - D8F0h)_{decimal}}{1ms}$) des Timers 0. Der Interrupt kann nicht ausgelöst werden, wenn gerade eine serielle Übertragung statt findet. Aus diesem Grund wurde

auch die Baudrate recht hoch gewählt, um die Zeiträume für eine serielle Übertragung möglichst kurz zu halten.

In dieser ISR findet die eigentliche Drehzahlregelung statt. Die über die serielle Schnittstelle empfangenen Drehzahlen werden einer Variablen zugewiesen, die in der ISR abgefragt wird und als Regelgrößenvorgabe gilt. In der ISR wird zunächst am A/D-Wandlerport 6.2 ausgelesen, um den momentanen Istwert zu ermitteln, der dann mit dem aktuellen Sollwert verglichen wird.

Da die Regelung sehr schnell ist, wird die Solldrehzahl in einer Hilfsvariablen (Hilfssolldrehzahl) zwischengespeichert. Die eigentliche Solldrehzahl-Variable wird, gesteuert von einem in der ISR gesetzten Flag, in bestimmten Zeittakten erhöht, bis die eigentliche Solldrehzahl erreicht ist. So wird ein langsames Beschleunigen und Abbremsen der Drehzahl erreicht. Dieses Verzögern ist sehr wichtig, denn der Motor hat eine recht hohe Leistung und würde so schnell beschleunigen, daß das aufgelegte Glasblättchen wegfliegen würde.

Die Eingabeports, die in der Endlosschleife überprüft werden, lesen digitale und analoge Werte ein.

Art	Port	Ereignis	Bemerkungen
analog	6.0	Motorstrom	High $\hat{=}$ zu viel Strom \Rightarrow aus
analog	6.1	Temperatur	High $\hat{=}$ zu heiß \Rightarrow aus
analog	6.2	Tachogenerator	
digital	5.1	Unterdruck	Eingang High $\hat{=}$ OK Low $\hat{=}$ Fehler
digital	5.2	Entwickler	Eingang High $\hat{=}$ OK Low $\hat{=}$ Fehler
digital	5.3	Spülung	Eingang High $\hat{=}$ OK Low $\hat{=}$ Fehler
digital	5.4	Start	Eingang High $\hat{=}$ Taster offen Low $\hat{=}$ Taster gedrückt / gedrückt
digital	5.5	Stop	Eingang High $\hat{=}$ Taster offen Low $\hat{=}$ Taster gedrückt / gedrückt

Tabelle 2: Analoge und digitale Eingänge und ihre Funktionen

Wie aus der Tabelle hervorgeht, besteht die Möglichkeit, über digitale Geber auch Unterdruck, Entwicklerfluß und Spülungsfluß zu kontrollieren. Ob diese Funktionen im endgültigen Produkt zum Einsatz kommen, ist noch nicht sicher. Von Hard- und Softwareseite ist die Möglichkeit auf jeden Fall vorgesehen.

Wenn dann an einem solchen Port ein negatives Ereignis eintritt, wird der Programmablauf gestoppt und der Abbruch der Steuersoftware auf dem PC signalisiert. An Port 5.4 und 5.5 sind Taster, die direkt am Arbeitsplatz angebracht sind; hier hat der Bediener die Möglichkeit, den Prozeß zu starten und zu stoppen.

Bei den Ausgabeports handelt es sich nur um digitale Ports und den Port 1.3, an dem die PWM realisiert wird. Da die digitalen Ports über die Masse schalten, sind nach dem Programmstart alle Ports auf HIGH und werden erst, wenn seriell ein Schaltbefehl empfangen wurde, auf LOW, und damit "aktiv" gesetzt.

Art	Port	Ereignis	Bemerkungen
analog	1.3	PWM	Zur Motoransteuerung
digital	4.0	Status LED	Ausgang High $\hat{=}$ aus Low $\hat{=}$ an
digital	4.1	Motor an/aus	Ausgang High $\hat{=}$ Signal aus Low $\hat{=}$ Signal an
digital	4.2	Entwickler	Ausgang High $\hat{=}$ aus Low $\hat{=}$ an (Flüssigkeit läuft)
digital	4.3	Spülung	Ausgang High $\hat{=}$ aus Low $\hat{=}$ an (Flüssigkeit läuft)
digital	4.4	Schwenkarm	Ausgang High $\hat{=}$ aus Low $\hat{=}$ an (Arm schwenkt)

Tabelle 3: Analoge und digitale Ausgänge und ihre Funktionen

An Port 4.0 ist eine Status-LED angeschlossen, die verschiedene Prozeßzustände signalisiert, so daß man sofort sieht, wenn der Prozeß steht oder läuft.

Da auch bei O_{digital} noch ein kleines PWM-Signal anliegt, wurde Port 4.1 nachträglich aufgenommen, um den PWM-Signal-Ausgang ein- bzw. auszuschalten.

An Port 4.2 und 4.3 werden die beiden Flüssigkeiten ein- und ausgeschaltet und mit Port 4.4 wird der Schwenkarm auf seine Position gefahren.

5.6.2. Der Regelkreis

5.6.2.1. Allgemeines

Der Regelkreis war leider nicht rechnerisch zu bestimmen, da die Schleuderzentrifuge uns nicht zur Verfügung stand. So waren wir gezwungen den Regelkreis empirisch zu ermitteln. Dieses Verfahren ist zeitaufwendig, da es sehr viele Faktoren gibt, die sich auf die Regelstrecke auswirken. So bekamen wir im Laufe der Arbeit einen neuen Motor und schon war die ganze Anpassung der Regelung hinfällig. Das Gleiche passierte, als wir kurz vor Ende der Arbeit die Schleuderzentrifuge bekamen. Diese lief so schwergängig, daß der Motor so viel Strom zog, daß verschiedene Sicherheitsmechanismen sofort zum Abbruch des Prozesses führten.

Der PI-Regler und der PID-Regler waren vom Programmieraufwand so hoch, daß wir den Timer zum Aufruf der ISR deutlich verlangsamten mußten. So setzten wir dann einen P-Regler ein, der von der Regelgeschwindigkeit und -genauigkeit völlig ausreichend war.

5.6.2.2. Funktionsweise

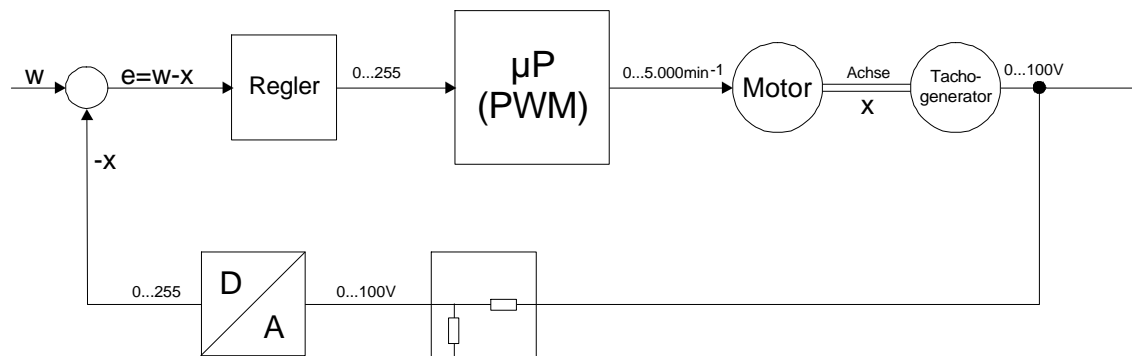


Abbildung 29: Der Regelkreis unserer Steuerungseinheit

Die am PC eingegebene Drehzahl wird seriell als ASCII-Zeichen (0-255d) zur Steuerungshardware gesendet, dort in einer Hilfsvariablen gespeichert und über einen Verzögerungstimer der Sollgeschwindigkeitsvariablen zugewiesen. Über einen A/D-Wandler-Port wird die aktuelle Tachospaltung eingelesen und in einen digitalen Wert (0-255d) gewandelt. In der ISR zur Drehzahlregelung wird nun die Regelabweichung und die neue Stellgröße $y_{(t)}$ errechnet.

Gleichung 4:
$$y_{(t)} = (\text{Sollwert} - \text{Istwert}) * \text{Verstärkung}$$

Bevor die neue Stellgröße dem PWM-Port zugewiesen wird, wird noch geprüft, ob kein Überlauf stattgefunden hat. D.h., daß das bei der Subtraktion des Istwertes vom Sollwert entstandene Ergebnis kein negativer Wert ist. Denn ein negativer Wert entspricht in dem 8bit Register dem komplementären positiven Wert und das würde zu gravierenden Fehlern führen.

6. Zusammenfassung

Hardware

Bei Abgabe der Diplomarbeit war die Anlage noch nicht einsatzbereit, wird derzeit aber in weiteren Arbeitsgängen, bis zur vollen Funktionalität, weitergebaut.

Wie in Kapitel 5.6.2.1 auf Seite 43 bereits erwähnt, war uns der Regelkreis erst am Ende der Arbeit bekannt. So konnten wir das eigentliche Regelverhalten des Systems nicht rechnerisch ermitteln. Das führte natürlich unweigerlich zu einigen Problemen bei der Anpassung des Systems. Aus diesem Grund haben wir versucht, möglichst viele Abschnitte variabel zu bauen und zu programmieren. So hat man die Möglichkeit, das System auch bei sich ändernden Randbedingungen wieder anzupassen.

Ferner bekamen wir mitten in der Arbeit einen neuen Motor, mit dem einige Programmteile nicht mehr korrekt arbeiteten, was aber durch unsere recht flexible Arbeitsweise schnell angepaßt werden konnte.

Die Schleuderzentrifuge wurde von einer Drittfirma gefertigt und stand uns leider erst gegen Ende der Arbeit zur Verfügung (s. Kapitel 5.6.2.1). Diese Schleuderzentrifuge hatte einen Produktionsfehler und war so schwergängig, daß die Steuerung sofort einen Fehler registrierte, da die Stromaufnahme zu hoch war. Nachdem wir die Schleuder zerlegten, entdeckten wir den Fehler sofort, und liesen ihn beseitigen.

Jetzt ist die Anlage komplett und wir können damit beginnen, die Anpassung durchzuführen. Leider stand so zum Ende dieser Arbeit kein endgültiges Produkt zur Verfügung.

Regelsoftware

Bei der Regelsoftware wurden alle geforderten Funktionen realisiert und das Programm ist lauffertig. Es mußte zum Ende der Arbeit lediglich noch in seinen Parametern angepaßt werden.

Nun möchte ich kurz auf die Verwendung der C51-Hochsprache (siehe Programmlisting auf Seite 117) bei der Regelsoftware eingehen. Der Nachteil der Hochsprache C51 ist, daß sie oft mehr Maschinencode als nötig produziert. Als Vorteil muß man allerdings vermerken, daß man mit einer Hochsprache ein deutlich übersichtlicheres Programm erhält, in das sich auch ein Dritter viel einfacher hineindenken kann. Dieser Punkt erschien mir in unserem Fall auch recht wichtig, da vielleicht ein Mitarbeiter der Firma, zu einem späteren Zeitpunkt, noch etwas an der Anlage ändern wird.

Der Nachteil von Assembler liegt einzig vor allem darin, daß man beim Programmieren den Befehlssatz des μP kennenlernen und alle mathematischen Funktionen selbst programmieren muß. Vorteilhafter ist die Assemblersprache (siehe Seite 31) vor allem, weil sie relativ kompakt und damit sehr schnell ist. Sie wird in kostenkritischen Applikationen überwiegend eingesetzt. So hätte man auf jeden Fall den externen Speicher für den μP sparen können. Man hätte also die Regelsoftware besser direkt in Assembler geschrieben.

Steuer- und Bediensoftware

Bei der Steuer- und Bediensoftware wurden alle geforderten Funktionen realisiert. Lediglich einige Parameter mußten noch angepasst werden.

Nach anfänglichen Mißerfolgen mit Turbo Pascal unter DOS (siehe Seite 12) schrieb ich die Bediensoftware in Microsoft Visual Basic 5.0 (32bit) unter Windows (siehe Programmlisting Seite auf 49).

In Kapitel 4.4.9 auf Seite 30 wird beschrieben, daß am Ende des Prozesses aus Sicherheitsgründen alle Ventile geschlossen werden, der Schwenkarm in die Ausgangsposition gebracht wird und der Motor ein Stop-Signal bekommt. Dies wird gemacht, da wir keine Rückmeldung haben, ob eine Flüssigkeit fließt oder nicht. Diese Funktionen sind auf der Steuerplatine und in der Software vorgesehen, sollten aber von seiten der Firma nicht weiter ausgebaut werden.

7. Kritische Schlußbetrachtung

Zum Schluß möchte ich noch ein paar Anmerkungen zur Durchführung dieser Arbeit machen und einige Verbesserungsvorschläge anführen.

Hardware

Rückblickend läßt sich zur Hardware sagen, daß man verschiedene Teile hätte besser lösen können. So wäre es zum Beispiel sehr sinnvoll gewesen, einen Pulsbreitenmodulator mit 10-bit zu verwenden. Damit hätten wir den doch sehr großen Drehzahlbereich, der sich von 600 bis 15.000 min^{-1} erstreckt, in kleinere, feinere Schritte aufteilen können, um eine genauere Drehzahleinstellung zu erreichen. So hätten wir die Drehzahl in 25 min^{-1} -Schritten statt 100 min^{-1} -Schritten steuern können.

Wir hatten weiter einen externen RAM für den Prozessor vorgesehen, welcher sich aber im Laufe der Programmierarbeiten eher als überflüssig herausstellte, da der interne Speicher des SAB 80C535 völlig ausreichte. Da dieser externe RAM nicht teuer ist, wurde er auch nicht mehr entfernt. Sollte die Regelsoftware einmal erweitert werden, kann es durchaus sein, daß das Programm so umfangreich wird, daß der externe RAM doch noch benötigt wird.

Als Steuer-PC hätte auch ein kleineres Modell völlig ausgereicht. Allerdings sollten die Mitarbeiter, von diesen Arbeitsplätzen aus, auf die Auftragsbearbeitungssoftware des Unternehmens Zugriff haben und diese Software stellt sehr hohe Ansprüche an den PC, da sie mit einer SQL-Datenbank arbeitet.

Regelsoftware

Wenn die Regelsoftware erweitert werden sollte, kann es durchaus von Nutzen sein, sie direkt in Assembler zu programmieren. Denn je größer dieses Programm wird umso länger werden die Durchlaufzeiten, und das kann irgendwann zu Störungen bei der Regelung oder der seriellen Kommunikation führen.

Steuer- und Bediensoftware

Hier könnte man noch eine graphische Auswertung des Prozeßablaufes programmieren, die man zusammen mit dem Prozeßprotokoll ausdrucken kann.

Der komplette Druckvorgang stand zum Zeitpunkt der Abgabe der Arbeit noch nicht zur Verfügung.

8. Kurzbeschreibung

Die Diplomarbeit behandelt eine unter Microsoft Visual Basic 5.0 programmierte grafische Benutzeroberfläche für Windows 95. Diese 32bit Anwendung wurde zur Bedienung einer speziell entwickelten Steuerhardware für eine PC-gesteuerte Wafer-Zentrifuge programmiert. Die Steuerhardware basiert auf einem Mikrocontroller Siemens SAB 80535 und kommuniziert mit dem Windowsprogramm über eine serielle Schnittstelle (RS 232). Die Software des Mikrocontrollers wurde in C geschrieben und mit dem C51-Compiler der Firma Keil Electronics GmbH in Maschinencode übersetzt.

This dissertation is about a Windows 95 user interface which is programmed with Microsoft Visual Basic 5.0. This 32bit application was programmed to operate on a specially created Hardware for a PC-controlled Wafer-centrifuge. The control-hardware is based on a Siemens SAB 80535 microcontroller and communicates with the windows-application by a serial interface (RS232). The microcontroller's software was edited in C and was assembled with the C51-Compiler of the Keil Electronics GmbH.

9. Anhang

9.1. Programmlisting der Steuersoftware

Texte, die nach den ' stehen, sind lediglich Kommentare und dienen dem Verständnis beim lesen des Programmes.

FrmAbout (Programmcode)

```
Option Explicit
' Reg Key Security Options...
Const READ_CONTROL = &H20000
Const KEY_QUERY_VALUE = &H1
Const KEY_SET_VALUE = &H2
Const KEY_CREATE_SUB_KEY = &H4
Const KEY_ENUMERATE_SUB_KEYS = &H8
Const KEY_NOTIFY = &H10
Const KEY_CREATE_LINK = &H20
Const KEY_ALL_ACCESS = KEY_QUERY_VALUE + KEY_SET_VALUE + _
                        KEY_CREATE_SUB_KEY + KEY_ENUMERATE_SUB_KEYS + _
                        KEY_NOTIFY + KEY_CREATE_LINK + READ_CONTROL

' Reg Key ROOT Types...
Const HKEY_LOCAL_MACHINE = &H80000002
Const ERROR_SUCCESS = 0
Const REG_SZ = 1 ' Unicode nul terminated string
Const REG_DWORD = 4 ' 32-bit number
Const gREGKEYSYSINFOLOC = "SOFTWARE\Microsoft\Shared Tools Location"
Const gREGVALSYSINFOLOC = "MSINFO"
Const gREGKEYSYSINFO = "SOFTWARE\Microsoft\Shared Tools\MSINFO"
Const gREGVALSYSINFO = "PATH"
Private Declare Function RegOpenKeyEx Lib "advapi32" Alias
"RegOpenKeyExA" (ByVal hKey As Long, ByVal lpSubKey As String, ByVal
ulOptions As Long, ByVal samDesired As Long, ByRef phkResult As Long) As
Long
Private Declare Function RegQueryValueEx Lib "advapi32" Alias
"RegQueryValueExA" (ByVal hKey As Long, ByVal lpValueName As String,
ByVal lpReserved As Long, ByRef lpType As Long, ByVal lpData As String,
ByRef lpcbData As Long) As Long
Private Declare Function RegCloseKey Lib "advapi32" (ByVal hKey As Long)
As Long

Private Sub cmdOK_KeyPress(KeyAscii As Integer)
    If KeyAscii = 27 Then
        cmdOK_Click
    End If
End Sub

Private Sub cmdSysInfo_Click()
    Call StartSysInfo
End Sub

Private Sub cmdOK_Click()
    Unload Me
End Sub

Private Sub Form_Load()
    Me.Caption = "Info's über " & App.Title & "!"
End Sub
```

```

    lblVersion.Caption = "Version " & App.Major & "." & App.Minor & "." &
App.Revision
    lblTitle.Caption = App.Title
    lblDescription.Caption = "Pumpensteuereung für
Strichplattenbeschichtung." & Chr(13) & " Einsatz in der Firma:" &
Chr(13) & Chr(13) & " H.W. Feldmann GmbH" & Chr(13) & " Bergstraße 31" &
Chr(13) & " 35583 Wetzlar" & Chr(13) & "Tel.: 06441 / 50005-0" & Chr(13)
& "Fax.: 06441 / 50005-55"
    lblDisclaimer.Caption = "Erstellt von:" & Chr(13) & Chr(13) & "
Holger Erb, 35428 Langgöns" & Chr(13) & "Thomas Jockel, 35410 Hungen"
End Sub

Public Sub StartSysInfo()
    On Error GoTo SysInfoErr
    Dim rc As Long
    Dim SysInfoPath As String
    ' Try To Get System Info Program Path\Name From Registry...
    If GetKeyValue(HKEY_LOCAL_MACHINE, gREGKEYSYSINFO, gREGVALSYSINFO,
SysInfoPath) Then
    ' Try To Get System Info Program Path Only From Registry...
    ElseIf GetKeyValue(HKEY_LOCAL_MACHINE, gREGKEYSYSINFOLOC,
gREGVALSYSINFOLOC, SysInfoPath) Then
        ' Validate Existance Of Known 32 Bit File Version
        If (Dir(SysInfoPath & "\MSINFO32.EXE") <> "") Then
            SysInfoPath = SysInfoPath & "\MSINFO32.EXE"
        ' Error - File Can Not Be Found...
        Else
            GoTo SysInfoErr
        End If
    ' Error - Registry Entry Can Not Be Found...
    Else
        GoTo SysInfoErr
    End If
    Call Shell(SysInfoPath, vbNormalFocus)
    Exit Sub
SysInfoErr:
    MsgBox "System Information Is Unavailable At This Time", vbOKOnly
End Sub

Public Function GetKeyValue(KeyRoot As Long, KeyName As String, SubKeyRef
As String, ByRef KeyVal As String) As Boolean
    Dim I As Long ' Loop Counter
    Dim rc As Long ' Return Code
    Dim hKey As Long ' Handle To An Open
Registry Key
    Dim hDepth As Long '
    Dim KeyValType As Long ' Data Type Of A Registry
Key
    Dim tmpVal As String ' Tempory Storage For A
Registry Key Value
    Dim KeyValSize As Long ' Size Of Registry Key
Variable
    '-----
    ' Open RegKey Under KeyRoot {HKEY_LOCAL_MACHINE...}
    '-----
    rc = RegOpenKeyEx(KeyRoot, KeyName, 0, KEY_ALL_ACCESS, hKey) ' Open
Registry Key

    If (rc <> ERROR_SUCCESS) Then GoTo GetKeyError ' Handle
Error...

    tmpVal = String$(1024, 0) ' Allocate
Variable Space

```

```

    KeyValSize = 1024                                ' Mark
Variable Size

    '-----
    ' Retrieve Registry Key Value...
    '-----
    rc = RegQueryValueEx(hKey, SubKeyRef, 0, _
                        KeyValType, tmpVal, KeyValSize)  ' Get/Create
Key Value

    If (rc <> ERROR_SUCCESS) Then GoTo GetKeyError      ' Handle
Errors

    If (Asc(Mid(tmpVal, KeyValSize, 1)) = 0) Then      ' Win95 Adds
Null Terminated String...
        tmpVal = Left(tmpVal, KeyValSize - 1)        ' Null Found,
Extract From String
    Else                                              ' WinNT Does
NOT Null Terminate String...
        tmpVal = Left(tmpVal, KeyValSize)            ' Null Not Found,
Extract String Only
    End If

    '-----
    ' Determine Key Value Type For Conversion...
    '-----
    Select Case KeyValType                            ' Search Data
Types...
        Case REG_SZ                                  ' String Registry
Key Data Type
            KeyVal = tmpVal                          ' Copy String Value
        Case REG_DWORD                               ' Double Word
Registry Key Data Type
            For I = Len(tmpVal) To 1 Step -1         ' Convert
Each Bit
                KeyVal = KeyVal + Hex(Asc(Mid(tmpVal, I, 1))) ' Build Value
Char. By Char.
            Next
            KeyVal = Format$("&h" + KeyVal)           ' Convert Double
Word To String
        End Select

    GetKeyValue = True                                ' Return
Success
    rc = RegCloseKey(hKey)                            ' Close
Registry Key
    Exit Function                                     ' Exit

GetKeyError:    ' Cleanup After An Error Has Occured...
    KeyVal = ""                                       ' Set Return
Val To Em
pty String
    GetKeyValue = False                                ' Return
Failure
    rc = RegCloseKey(hKey)                            ' Close
Registry Key
End Function

```

FrmAbout (Fensterdefinon)

```

VERSION 5.00
Begin VB.Form frmAbout

```

```
BorderStyle      = 3  'Fester Dialog
Caption          = "About MyApp"
ClientHeight     = 3555
ClientLeft       = 2340
ClientTop        = 1935
ClientWidth      = 5730
ClipControls     = 0  'False
ControlBox       = 0  'False
LinkTopic        = "Form2"
MaxButton        = 0  'False
MinButton        = 0  'False
ScaleHeight      = 2453.724
ScaleMode        = 0  'Benutzer
ScaleWidth       = 5380.766
ShowInTaskbar    = 0  'False
Begin VB.PictureBox picIcon
    AutoSize      = -1  'True
    ClipControls  = 0  'False
    Height        = 540
    Left         = 240
    Picture       = (Symbol)
    ScaleHeight   = 337.12
    ScaleMode     = 0  'Benutzer
    ScaleWidth    = 337.12
    TabIndex      = 1
    Top          = 240
    Width        = 540
End
Begin VB.CommandButton cmdOK
    Cancel        = -1  'True
    Caption       = "OK"
    Default       = -1  'True
    Height        = 345
    Left         = 4245
    TabIndex      = 0
    Top          = 2625
    Width        = 1260
End
Begin VB.CommandButton cmdSysInfo
    Caption       = "&System Info..."
    Height        = 345
    Left         = 4260
    TabIndex      = 2
    Top          = 3075
    Width        = 1245
End
Begin VB.Line Line1
    BorderColor   = &H00808080&
    BorderStyle   = 6  'Innen ausgefüllt
    Index        = 1
    X1           = 84.515
    X2           = 5309.398
    Y1           = 1687.583
    Y2           = 1687.583
End
Begin VB.Label lblDescription
    Caption       = "Pumpensteuerung für
Strichplattenbeschichtung."
    ForeColor     = &H00000000&
    Height        = 1530
    Left         = 1050
    TabIndex      = 3
    Top          = 765
```

```
        Width          = 3885
    End
    Begin VB.Label lblTitle
        Caption          = "Application Title"
        ForeColor        = &H00000000&
        Height           = 360
        Left              = 1050
        TabIndex         = 5
        Top               = 240
        Width             = 1965
    End
    Begin VB.Line Line1
        BorderColor      = &H00FFFFFF&
        BorderWidth      = 2
        Index             = 0
        X1                = 98.6
        X2                = 5309.398
        Y1                = 1697.936
        Y2                = 1697.936
    End
    Begin VB.Label lblVersion
        Caption          = "Version"
        Height           = 225
        Left              = 3240
        TabIndex         = 6
        Top               = 240
        Width             = 1725
    End
    Begin VB.Label lblDisclaimer
        Caption          = "Erarbeitet von:"
        ForeColor        = &H00000000&
        Height           = 825
        Left              = 255
        TabIndex         = 4
        Top               = 2625
        Width             = 3870
    End
End
```

FrmComPort (Programmcode)

```
Dim altComPort As Byte

Private Sub cmdAbbrechen_Click()
    ComPort = altComPort
    frmSteuerung.test.Caption = frmSteuerung.test.Caption & "Com" & ComPort
+ 1
    Unload frmComPort
End Sub

Private Sub cmdAbbrechen_KeyPress(KeyAscii As Integer)
    If KeyAscii = 27 Then
        cmdAbbrechen_Click
    End If
End Sub

Private Sub cmdOK_Click()
    frmSteuerung.test.Caption = frmSteuerung.test.Caption & "Com" & ComPort
+ 1
    frmSteuerung.lblConnect.Caption = "Steuerplatine verbunden (Com" &
ComPort + 1 & "):"
```

```

    CFG_Speichern
    Unload frmComPort
End Sub

Private Sub cmdOK_KeyPress(KeyAscii As Integer)
    If KeyAscii = 27 Then
        cmdAbbrechen_Click
    End If
End Sub

Private Sub Form_Load()
    altComPort = ComPort
    Option1(ComPort).Value = True           'Marke auf den Option-Punkt
setzen
    For x = 0 To 3
        Option1(x).Caption = "Com Port " & x + 1
    Next x
End Sub

Private Sub Option1_Click(Index As Integer)
    ComPort = Index
End Sub

```

FrmComPort (Fensterdefinon)

```

VERSION 5.00
Begin VB.Form frmComPort
    Caption           = "Serielle Schnittstelle"
    ClientHeight      = 2340
    ClientLeft        = 60
    ClientTop         = 345
    ClientWidth       = 4080
    ControlBox        = 0   'False
    LinkTopic         = "Form1"
    ScaleHeight       = 2340
    ScaleWidth        = 4080
    ShowInTaskbar     = 0   'False
    StartupPosition   = 2   'Bildschirmmitte
    Begin VB.CommandButton cmdAbbrechen
        Caption        = "Abbrechen"
        BeginProperty Font
            Name         = "MS Sans Serif"
            Size         = 9.75
            Charset      = 0
            Weight       = 700
            Underline    = 0   'False
            Italic       = 0   'False
            Strikethrough = 0   'False
        EndProperty
        Height          = 375
        Left            = 2160
        TabIndex        = 7
        Top             = 1800
        Width           = 1695
    End
    Begin VB.CommandButton cmdOK
        Caption         = "OK"
        BeginProperty Font
            Name         = "MS Sans Serif"
            Size         = 9.75
            Charset      = 0

```

```
        Weight          = 700
        Underline       = 0   'False
        Italic          = 0   'False
        Strikethrough   = 0   'False
    EndProperty
    Height              = 375
    Left                = 120
    TabIndex            = 6
    Top                 = 1800
    Width               = 1695
End
Begin VB.Frame Frame1
    Height              = 1095
    Left                = 120
    TabIndex            = 0
    Top                 = 480
    Width               = 3855
    Begin VB.OptionButton Option1
        Caption          = "Com Port "
        Height           = 375
        Index            = 3
        Left             = 2280
        TabIndex         = 4
        Top              = 600
        Width            = 1500
    End
    Begin VB.OptionButton Option1
        Caption          = "Com Port "
        Height           = 375
        Index            = 2
        Left             = 2280
        TabIndex         = 3
        Top              = 240
        Width            = 1500
    End
    End
    Begin VB.OptionButton Option1
        Caption          = "Com Port "
        Height           = 375
        Index            = 1
        Left             = 240
        TabIndex         = 2
        Top              = 600
        Width            = 1500
    End
    End
    Begin VB.OptionButton Option1
        Caption          = "Com Port "
        Height           = 375
        Index            = 0
        Left             = 240
        TabIndex         = 1
        Top              = 240
        Width            = 1500
    End
    End
End
Begin VB.Label Label1
    Caption              = "Bitte wählen Sie eine serielle Schnittstelle:"
    Height              = 255
    Left                = 120
    TabIndex            = 5
    Top                 = 240
    Width               = 3495
End
End
```

FrmEingabe (Programmcode)

```
Private Sub cmdAbbrechen_Click()  
    Unload frmEingabe  
End Sub  
  
Private Sub cmdAbbrechen_KeyPress(KeyAscii As Integer)  
    If KeyAscii = 27 Then  
        cmdAbbrechen_Click  
    End If  
End Sub  
  
Private Sub cmdLöschen_Click()  
    WertDrehzahl(paarIdx) = 0  
    frmSteuerung.anzDrehzahl(paarIdx) = 0 & " U/min. "  
    WertZeit(paarIdx) = 0  
    frmSteuerung.anzZeit(paarIdx) = 0 & " Sekunden "  
    WertEreignis(paarIdx) = 0  
    frmSteuerung.anzEreignis(paarIdx) = ComboEreignis.List(0)  
    flgGeändert = True  
    If (Right(frmSteuerung.Caption, 1) <> "*") Then  
        frmSteuerung.Caption = frmSteuerung.Caption & "*"   
    End If  
    AnzeigeAktualisieren  
    Unload frmEingabe  
End Sub  
  
Private Sub cmdOK_Click()  
    If txtDrehzahl.text > 0 Then  
        If txtZeit.text > 0 Then  
            If ComboEreignis.ListIndex = -1 Then  
                WertEreignis(paarIdx) = 0  
                frmSteuerung.anzEreignis(paarIdx) = ComboEreignis.List(0)  
            Else  
                WertEreignis(paarIdx) = ComboEreignis.ListIndex  
                frmSteuerung.anzEreignis(paarIdx) =  
ComboEreignis.List(ComboEreignis.ListIndex)  
            End If  
            If WertEreignis(paarIdx) = 0 Then  
                MsgBox ("Sie haben kein Ereignis gewählt!!")  
                ComboEreignis.SetFocus  
                Exit Sub  
            End If  
            WertDrehzahl(paarIdx) = txtDrehzahl.text * 100  
            WertZeit(paarIdx) = txtZeit.text  
            frmSteuerung.anzDrehzahl(paarIdx) = Format(WertDrehzahl(paarIdx),  
"##,##0") & " U/min. "  
            frmSteuerung.anzZeit(paarIdx) = WertZeit(paarIdx) & " Sekunden "  
            flgGeändert = True  
            If (Right(frmSteuerung.Caption, 1) <> "*") Then  
                frmSteuerung.Caption = frmSteuerung.Caption & "*"   
            End If  
            frmSteuerung.mnuSpeichern.Enabled = True  
            frmSteuerung.mnuDrucken.Enabled = True  
            frmSteuerung.mnuDatenfreigabe.Enabled = True  
            frmSteuerung.cmdDatenfreigabe.Enabled = True  
            AnzeigeAktualisieren  
            Unload frmEingabe  
        Else 'zeit  
            MsgBox ("Die Zeit muß größer 0 sein!!")
```



```
    txtZeit.SetFocus
End If 'zeit
Else 'drehzahl
    MsgBox ("Die Drehzahl muß größer 0 sein!!")
    txtDrehzahl.SetFocus
End If 'drehzahl
End Sub

Private Sub cmdOK_KeyPress(KeyAscii As Integer)
    If KeyAscii = 27 Then
        cmdAbbrechen_Click
    End If
End Sub

Private Sub ComboEreignis_KeyPress(KeyAscii As Integer)
    Select Case KeyAscii
        Case 13
            cmdOK.SetFocus
        Case 27
            cmdAbbrechen_Click
    End Select
End Sub

Private Sub Form_Load()
    frmEingabe.ComboEreignis.List(0) = "keine "
    frmEingabe.ComboEreignis.List(1) = "Entwickler "
    frmEingabe.ComboEreignis.List(2) = "Spülung "
    frmEingabe.ComboEreignis.List(3) = "trocknen "
    lblNr.Caption = "Paar Nr. " & (paarIdx + 1)
    txtDrehzahl.text = WertDrehzahl(paarIdx) / 100
    txtZeit.text = WertZeit(paarIdx)
    ComboEreignis.ListIndex = WertEreignis(paarIdx)
End Sub

Private Sub txtDrehzahl_GotFocus()
    txtDrehzahl.SelStart = 0
    txtDrehzahl.SelLength = Len(txtDrehzahl.text)
End Sub

Private Sub txtDrehzahl_KeyPress(KeyAscii As Integer)
    Select Case KeyAscii
        Case 13
            txtDrehzahl.SetFocus
            txtZeit.SetFocus
        Case 27
            cmdAbbrechen_Click
    End Select
End Sub

Private Sub txtDrehzahl_LostFocus()
Dim H1 As Single
    H1 = txtDrehzahl.text
    If H1 > (MaxDrehzahl / 100) Then
        'Beep
        txtDrehzahl.text = (MaxDrehzahl / 100)
        txtDrehzahl.SetFocus
    End If
    If H1 < (MinDrehzahl / 100) Then
        'Beep
        txtDrehzahl.text = (MinDrehzahl / 100)
        txtDrehzahl.SetFocus
    End If
End Sub
```

```
Private Sub txtZeit_GotFocus()  
    txtZeit.SelStart = 0  
    txtZeit.SelLength = Len(txtZeit.text)  
End Sub  
  
Private Sub txtZeit_KeyPress(KeyAscii As Integer)  
    Select Case KeyAscii  
        Case 13  
            ComboEreignis.SetFocus  
        Case 27  
            cmdAbbrechen_Click  
    End Select  
End Sub
```

FrmEingabe (Fensterdefinon)

```
VERSION 5.00  
Begin VB.Form frmEingabe  
    BorderStyle       = 3   'Fester Dialog  
    Caption           = "Eingabe"  
    ClientHeight      = 2940  
    ClientLeft        = 45  
    ClientTop         = 330  
    ClientWidth       = 3945  
    ControlBox        = 0   'False  
    LinkTopic         = "Form1"  
    MaxButton         = 0   'False  
    MinButton         = 0   'False  
    ScaleHeight       = 2940  
    ScaleWidth        = 3945  
    ShowInTaskbar     = 0   'False  
    StartUpPosition  = 1   'Fenstermitte  
    Begin VB.CommandButton cmdLöschen  
        Caption       = "Löschen"  
        Height        = 375  
        Left          = 2160  
        TabIndex      = 11  
        ToolTipText   = "Löschen des aktuellen Datensatzes"  
        Top           = 120  
        Width         = 855  
    End  
    Begin VB.CommandButton cmdAbbrechen  
        Caption       = "Abbrechen"  
        Height        = 315  
        Left          = 2880  
        TabIndex      = 10  
        ToolTipText   = "Eingabe abbrechen"  
        Top           = 600  
        Width         = 975  
    End  
    Begin VB.ComboBox ComboEreignis  
        BeginProperty Font  
            Name       = "MS Sans Serif"  
            Size       = 9.75  
            Charset    = 0  
            Weight     = 400  
            Underline  = 0   'False  
            Italic     = 0   'False  
            Strikethrough = 0   'False  
        EndProperty  
    End
```

```
Height          = 360
Left            = 360
TabIndex       = 3
Text           = "Ereignis"
ToolTipText    = "Ereignis aus der Liste"
Top            = 2400
Width          = 3015
End
Begin VB.CommandButton cmdOK
Caption        = "OK"
BeginProperty Font
Name          = "MS Sans Serif"
Size         = 9.75
Charset      = 0
Weight       = 700
Underline    = 0 'False
Italic       = 0 'False
Strikethrough = 0 'False
EndProperty
Height        = 375
Left         = 3120
TabIndex     = 5
ToolTipText  = "Werte übernehmen"
Top          = 120
Width        = 735
End
Begin VB.TextBox txtZeit
BeginProperty Font
Name          = "MS Sans Serif"
Size         = 9.75
Charset      = 0
Weight       = 400
Underline    = 0 'False
Italic       = 0 'False
Strikethrough = 0 'False
EndProperty
Height        = 360
Left         = 360
TabIndex     = 2
Text         = "Zeit"
ToolTipText  = "Zeit in Sekunden"
Top          = 1680
Width        = 2175
End
Begin VB.TextBox txtDrehzahl
Alignment     = 1 'Rechts
BeginProperty Font
Name          = "MS Sans Serif"
Size         = 9.75
Charset      = 0
Weight       = 400
Underline    = 0 'False
Italic       = 0 'False
Strikethrough = 0 'False
EndProperty
Height        = 360
Left         = 360
MaxLength    = 3
TabIndex     = 1
Text         = "Dre"
ToolTipText  = "Drehzahl * 100"
Top          = 960
Width        = 1350
```

```
End
Begin VB.TextBox Text1
  BeginProperty Font
    Name           = "MS Sans Serif"
    Size           = 12
    Charset        = 0
    Weight         = 400
    Underline      = 0 'False
    Italic         = 0 'False
    Strikethrough  = 0 'False
  EndProperty
  Height          = 360
  Left            = 1680
  MaxLength       = 3
  TabIndex        = 12
  Text            = "00"
  ToolTipText     = "Drehzahl * 100"
  Top             = 960
  Width           = 375
End
Begin VB.Label Label3
  Caption         = "Ereignis:"
  Height          = 255
  Left            = 360
  TabIndex        = 9
  Top             = 2160
  Width           = 1215
End
Begin VB.Label Label2
  Caption         = "Zeit:"
  Height          = 255
  Left            = 360
  TabIndex        = 7
  Top             = 1440
  Width           = 1215
End
Begin VB.Label Label1
  Caption         = "Drehzahl:"
  Height          = 255
  Left            = 360
  TabIndex        = 6
  Top             = 720
  Width           = 1215
End
Begin VB.Label Label5
  Caption         = "in Sekunden"
  Height          = 255
  Left            = 2640
  TabIndex        = 4
  Top             = 1800
  Width           = 1095
End
Begin VB.Label Label4
  Caption         = "in U/min."
  Height          = 255
  Left            = 2640
  TabIndex        = 8
  Top             = 1080
  Width           = 1095
End
Begin VB.Label lblNr
  Caption         = "Nr"
  BeginProperty Font
```

```

        Name           = "MS Sans Serif"
        Size           = 9.75
        Charset        = 0
        Weight         = 700
        Underline      = 0 'False
        Italic         = 0 'False
        Strikethrough  = 0 'False
    EndProperty
    Height           = 615
    Left             = 360
    TabIndex         = 0
    Top              = 120
    Width            = 1815
End
End

```

FrmFertig (Programmcode)

```

Private Sub cmdOK_Click()
    ' Form Steuerung auf Anfang setzten
    frmSteuerung.cmdDatenfreigabe.Visible = True
    frmSteuerung.cmdAbbrechen.Visible = False
    frmSteuerung.cmdEnde.Enabled = True
    frmSteuerung.cmdEnde.Visible = True
    frmSteuerung.cmdDatenfreigabe.Enabled = True
    frmSteuerung.mnuDatenfreigabe.Enabled = True
    frmSteuerung.mnuAbbrechen.Enabled = False
    frmSteuerung.mnuDaten.Enabled = True
    frmSteuerung.mnuDatei.Enabled = True
    frmSteuerung.mnuKonfiguration.Enabled = True
    frmSteuerung.lblFortschritt.Visible = False
    frmSteuerung.ProgressBar1.Top = 3360
    frmSteuerung.ProgressBar1.Height = 200
    frmSteuerung.ProgressBar1.Visible = False
    frmSteuerung.lblFortschritt.Caption = " Fortschritt:"
    frmSteuerung.cmdDatenfreigabe.SetFocus
    frmSteuerung.cmdStart.Enabled = False
    frmSteuerung.cmdStart.Visible = False
    frmSteuerung.mnuStarten.Enabled = False
    Unload frmFertig
End Sub

```

FrmFertig (Fensterdefinon)

```

VERSION 5.00
Begin VB.Form frmFertig
    BorderStyle   = 1 'Fest Einfach
    Caption      = "Prozess Fertig!"
    ClientHeight = 1770
    ClientLeft   = 45
    ClientTop    = 330
    ClientWidth  = 3495
    ClipControls = 0 'False
    ControlBox   = 0 'False
    LinkTopic    = "Form1"
    MaxButton    = 0 'False
    MinButton    = 0 'False
    ScaleHeight  = 1770
    ScaleWidth   = 3495
    StartUpPosition = 1 'Fenstermitte

```

```
Begin VB.CommandButton cmdOK
  Caption      = "OK"
  BeginProperty Font
    Name        = "MS Sans Serif"
    Size        = 9.75
    Charset     = 0
    Weight      = 400
    Underline   = 0   'False
    Italic      = 0   'False
    Strikethrough = 0   'False
  EndProperty
  Height       = 375
  Left         = 840
  TabIndex     = 0
  Top          = 1200
  Width        = 1815
End
Begin VB.Label Labell
  Alignment    = 2   'Zentriert
  Caption      = "Prozess wurde erfolgreich beendet!"
  BeginProperty Font
    Name        = "MS Sans Serif"
    Size        = 12
    Charset     = 0
    Weight      = 700
    Underline   = 0   'False
    Italic      = 0   'False
    Strikethrough = 0   'False
  EndProperty
  Height       = 735
  Left         = 360
  TabIndex     = 1
  Top          = 240
  Width        = 2655
End
End
```

FrmLPTPort (Programmcode)

```
Dim altLPTPort As Byte

Private Sub cmdAbbrechen_Click()
  lptPort = altLPTPort
  frmSteuerung.test.Caption = frmSteuerung.test.Caption & "LPT" & lptPort
+ 1
  Unload frmLPTPort
End Sub

Private Sub cmdAbbrechen_KeyPress(KeyAscii As Integer)
  If KeyAscii = 27 Then
    cmdAbbrechen_Click
  End If
End Sub

Private Sub cmdOK_Click()
  frmSteuerung.test.Caption = frmSteuerung.test.Caption & "LPT" & lptPort
+ 1
  CFG_Speichern
  Unload frmLPTPort
End Sub
```

```

Private Sub cmdOK_KeyPress(KeyAscii As Integer)
  If KeyAscii = 27 Then
    cmdAbbrechen_Click
  End If
End Sub

Private Sub Form_Load()
  altLPTPort = lptPort
  Option1(lptPort).Value = True
  For x = 0 To 2
    Option1(x).Caption = "LPT Port " & x + 1
  Next x
End Sub

Private Sub Option1_Click(Index As Integer)
  lptPort = Index
End Sub

```

FrmLPTPort (Fensterdefinition)

```

VERSION 5.00
Begin VB.Form frmLPTPort
  Caption           = "Parallele Schnittstelle"
  ClientHeight     = 2340
  ClientLeft       = 60
  ClientTop        = 345
  ClientWidth      = 4080
  ControlBox       = 0 'False
  LinkTopic        = "Form1"
  ScaleHeight      = 2340
  ScaleWidth       = 4080
  ShowInTaskbar    = 0 'False
  StartupPosition  = 2 'Bildschirmmitte
  Begin VB.CommandButton cmdAbbrechen
    Caption         = "Abbrechen"
    BeginProperty Font
      Name          = "MS Sans Serif"
      Size          = 9.75
      Charset       = 0
      Weight        = 700
      Underline     = 0 'False
      Italic        = 0 'False
      Strikethrough = 0 'False
    EndProperty
    Height         = 375
    Left           = 2160
    TabIndex       = 6
    Top            = 1800
    Width          = 1695
  End
  Begin VB.CommandButton cmdOK
    Caption         = "OK"
    BeginProperty Font
      Name          = "MS Sans Serif"
      Size          = 9.75
      Charset       = 0
      Weight        = 700
      Underline     = 0 'False
      Italic        = 0 'False
      Strikethrough = 0 'False
    EndProperty
  End

```

```
EndProperty
Height      = 375
Left        = 120
TabIndex    = 5
Top         = 1800
Width       = 1695
End
Begin VB.Frame Frame1
Height      = 1095
Left        = 120
TabIndex    = 0
Top         = 480
Width       = 3855
Begin VB.OptionButton Option1
Caption     = "LPT Port"
Height      = 375
Index       = 2
Left        = 2280
TabIndex    = 3
Top         = 240
Width       = 1500
End
Begin VB.OptionButton Option1
Caption     = "LPT Port"
Height      = 375
Index       = 1
Left        = 240
TabIndex    = 2
Top         = 600
Width       = 1500
End
Begin VB.OptionButton Option1
Caption     = "LPT Port"
Height      = 375
Index       = 0
Left        = 240
TabIndex    = 1
Top         = 240
Width       = 1500
End
End
Begin VB.Label Label1
Caption     = "Bitte wählen Sie eine parallele
Schnittstelle:"
Height      = 255
Left        = 120
TabIndex    = 4
Top         = 240
Width       = 3495
End
End
```

FrmOptionen (Programmcode)

```
Dim LstIdxAlt As Integer

Private Sub cmdAbbrechen_Click()
Unload frmOptionen
End Sub

Private Sub cmdAbbrechen_KeyPress(KeyAscii As Integer)
```



```
    If KeyAscii = 27 Then
        cmdAbbrechen_Click
    End If
End Sub
```

```
Private Sub cmdOK_Click()
    MaxDrehzahl = Text1.text
    MinDrehzahl = Text2.text
    serieller_Schleifenzähler = Text3.text
    MaxChar = Text4.text
    Passwort = Text5.text
    RELpause = Text6.text
    TMfaktor = Text10.text
    TSfaktor = Text7.text
    Sfaktor = Text8.text
    MaxStromMotor = Text11.text
    'DrehzahlKorrektur = Text9.Text
    CFG_Speichern
    Unload frmOptionen
End Sub
```

```
Private Sub cmdOK_KeyPress(KeyAscii As Integer)
    If KeyAscii = 27 Then
        cmdAbbrechen_Click
    End If
End Sub
```

```
Private Sub Combol_Click()
    ' Neuen Wert Anzeigen
    Text9.text = DrehzahlKorrektur(Combol.ListIndex + 1)
    Text9.Refresh
    PaarNr2.Caption = "bis: " & DrehzahlBereich(Combol.ListIndex + 2) &
    "U/min."
    LstIdxAlt = Combol.ListIndex
End Sub
```

```
Private Sub Form_Load()
Dim I As Integer
    Text1.text = MaxDrehzahl
    Text2.text = MinDrehzahl
    Text3.text = serieller_Schleifenzähler
    Text4.text = MaxChar
    Text5.text = Passwort
    Text6.text = RELpause
    Text7.text = TSfaktor
    Text10.text = TMfaktor
    Text8.text = Sfaktor
    Text11.text = MaxStromMotor
    'ReDim DrehzahlKorrektur(AnzahlDrehzahlen)
    'ReDim DrehzahlBereich(AnzahlDrehzahlen)
    Combol.Clear
    For I = 1 To AnzahlDrehzahlen
        Combol.AddItem DrehzahlBereich(I)
    Next I
    Combol.ListIndex = 0
    LstIdxAlt = 0
    Text9.text = DrehzahlKorrektur(Combol.ListIndex + 1)
    Text9.Refresh
    PaarNr2.Caption = "bis: " & DrehzahlBereich(Combol.ListIndex + 2) &
    "U/min."
End Sub
```

```
Private Sub Text1_GotFocus()
```

```
Text1.SelStart = 0
Text1.SelLength = Len(Text1.text)
End Sub

Private Sub Text1_KeyPress(KeyAscii As Integer)
    Select Case KeyAscii
        Case 13
            Text2.SetFocus
        Case 27
            cmdAbbrechen_Click
    End Select
End Sub

Private Sub Text10_KeyPress(KeyAscii As Integer)
    Select Case KeyAscii
        Case 13
            Text8.SetFocus
        Case 27
            cmdAbbrechen_Click
    End Select
End Sub

Private Sub Text2_KeyPress(KeyAscii As Integer)
    Select Case KeyAscii
        Case 13
            Text3.SetFocus
        Case 27
            cmdAbbrechen_Click
    End Select
End Sub

Private Sub Text3_KeyPress(KeyAscii As Integer)
    Select Case KeyAscii
        Case 13
            Text4.SetFocus
        Case 27
            cmdAbbrechen_Click
    End Select
End Sub

Private Sub Text4_KeyPress(KeyAscii As Integer)
    Select Case KeyAscii
        Case 13
            Text5.SetFocus
        Case 27
            cmdAbbrechen_Click
    End Select
End Sub

Private Sub Text5_KeyPress(KeyAscii As Integer)
    Select Case KeyAscii
        Case 13
            Text6.SetFocus
        Case 27
            cmdAbbrechen_Click
    End Select
End Sub

Private Sub Text6_KeyPress(KeyAscii As Integer)
    Select Case KeyAscii
        Case 13
            Text7.SetFocus
        Case 27
```

```
        cmdAbbrechen_Click
    End Select
End Sub

Private Sub Text7_KeyPress(KeyAscii As Integer)
    Select Case KeyAscii
        Case 13
            Text10.SetFocus
        Case 27
            cmdAbbrechen_Click
    End Select
End Sub

Private Sub Text8_KeyPress(KeyAscii As Integer)
    Select Case KeyAscii
        Case 13
            Text11.SetFocus
        Case 27
            cmdAbbrechen_Click
    End Select
End Sub

Private Sub Text11_KeyPress(KeyAscii As Integer)
    Select Case KeyAscii
        Case 13
            Text9.SetFocus
        Case 27
            cmdAbbrechen_Click
    End Select
End Sub

Private Sub Text9_KeyPress(KeyAscii As Integer)
    Select Case KeyAscii
        Case 13
            cmdOK.SetFocus
        Case 27
            cmdAbbrechen_Click
    End Select
End Sub

Private Sub Text2_GotFocus()
    Text2.SelStart = 0
    Text2.SelLength = Len(Text2.text)
End Sub

Private Sub Text3_GotFocus()
    Text3.SelStart = 0
    Text3.SelLength = Len(Text3.text)
End Sub

Private Sub Text4_GotFocus()
    Text4.SelStart = 0
    Text4.SelLength = Len(Text4.text)
End Sub

Private Sub Text5_GotFocus()
    Text5.SelStart = 0
    Text5.SelLength = Len(Text5.text)
End Sub

Private Sub Text6_GotFocus()
    Text6.SelStart = 0
    Text6.SelLength = Len(Text6.text)
End Sub
```

```
Private Sub Text7_GotFocus()  
    Text7.SelStart = 0  
    Text7.SelLength = Len(Text7.text)  
End Sub  
  
Private Sub Text10_GotFocus()  
    Text10.SelStart = 0  
    Text10.SelLength = Len(Text10.text)  
  
End Sub  
Private Sub Text11_GotFocus()  
    Text11.SelStart = 0  
    Text11.SelLength = Len(Text11.text)  
End Sub  
  
Private Sub Text8_GotFocus()  
    Text8.SelStart = 0  
    Text8.SelLength = Len(Text8.text)  
End Sub  
  
Private Sub Text9_GotFocus()  
    Text9.SelStart = 0  
    Text9.SelLength = Len(Text9.text)  
End Sub  
  
Private Sub Text9_LostFocus()  
    DrehzahlKorrektur(LstIdxAlt + 1) = Text9.text 'Alten Wert speichern  
    Text9.Refresh  
End Sub
```

FrmOptionen (Fensterdefinition)

```
VERSION 5.00  
Begin VB.Form frmOptionen  
    BorderStyle = 1 'Fest Einfach  
    Caption = "Optionen"  
    ClientHeight = 3105  
    ClientLeft = 45  
    ClientTop = 330  
    ClientWidth = 8910  
    ControlBox = 0 'False  
    LinkTopic = "Optionen"  
    MaxButton = 0 'False  
    MinButton = 0 'False  
    ScaleHeight = 3105  
    ScaleWidth = 8910  
    ShowInTaskbar = 0 'False  
    StartUpPosition = 1 'Fenstermitte  
    Begin VB.TextBox Text11  
        BeginProperty Font  
            Name = "MS Sans Serif"  
            Size = 9.75  
            Charset = 0  
            Weight = 400  
            Underline = 0 'False  
            Italic = 0 'False  
            Strikethrough = 0 'False  
        EndProperty  
        Height = 285  
        IMEMode = 3 'DISABLE
```

```
        Left           = 4440
        TabIndex       = 26
        Text           = "Text11"
        ToolTipText    = "Korrekturfaktor zur Anpassung des
Motorstromes"
        Top           = 1200
        Width          = 1215
    End
    Begin VB.TextBox Text10
        BeginProperty Font
            Name           = "MS Sans Serif"
            Size           = 9.75
            Charset        = 0
            Weight         = 400
            Underline      = 0 'False
            Italic         = 0 'False
            Strikethrough  = 0 'False
        EndProperty
        Height           = 285
        IMEMode          = 3 'DISABLE
        Left             = 4440
        TabIndex         = 24
        Text             = "Text10"
        ToolTipText      = "Korrekturfaktor zur Anpassung der
Trafotemperatur"
        Top             = 480
        Width            = 1215
    End
    Begin VB.ComboBox Combol
        Height           = 315
        Left             = 4440
        TabIndex         = 20
        Text             = "Combol"
        Top             = 1800
        Width            = 735
    End
    Begin VB.TextBox Text9
        BeginProperty Font
            Name           = "MS Sans Serif"
            Size           = 9.75
            Charset        = 0
            Weight         = 400
            Underline      = 0 'False
            Italic         = 0 'False
            Strikethrough  = 0 'False
        EndProperty
        Height           = 285
        IMEMode          = 3 'DISABLE
        Left             = 5280
        TabIndex         = 8
        Text             = "Text9"
        ToolTipText      = "Korrekturfaktor zur Anpassung der Drehzahl"
        Top             = 1800
        Width            = 975
    End
    Begin VB.TextBox Text8
        BeginProperty Font
            Name           = "MS Sans Serif"
            Size           = 9.75
            Charset        = 0
            Weight         = 400
            Underline      = 0 'False
            Italic         = 0 'False
```

```
        Strikethrough = 0 'False
    EndProperty
    Height = 285
    IMEMode = 3 'DISABLE
    Left = 4440
    TabIndex = 7
    Text = "Text8"
    ToolTipText = "Korrekturfaktor zur Anpassung des
Motorstromes"
    Top = 840
    Width = 1215
End
Begin VB.TextBox Text7
    BeginProperty Font
        Name = "MS Sans Serif"
        Size = 9.75
        Charset = 0
        Weight = 400
        Underline = 0 'False
        Italic = 0 'False
        Strikethrough = 0 'False
    EndProperty
    Height = 285
    IMEMode = 3 'DISABLE
    Left = 4440
    TabIndex = 6
    Text = "Text7"
    ToolTipText = "Korrekturfaktor zur Anpassung der
Trafotemperatur"
    Top = 120
    Width = 1215
End
Begin VB.TextBox Text6
    BeginProperty Font
        Name = "MS Sans Serif"
        Size = 9.75
        Charset = 0
        Weight = 400
        Underline = 0 'False
        Italic = 0 'False
        Strikethrough = 0 'False
    EndProperty
    Height = 285
    IMEMode = 3 'DISABLE
    Left = 360
    TabIndex = 5
    Text = "Text6"
    ToolTipText = "Verzögerungszeit für die Relais (in x.xx
Sekunden)"
    Top = 1920
    Width = 1215
End
Begin VB.TextBox Text5
    BeginProperty Font
        Name = "MS Sans Serif"
        Size = 9.75
        Charset = 0
        Weight = 400
        Underline = 0 'False
        Italic = 0 'False
        Strikethrough = 0 'False
    EndProperty
    Height = 285
```

```

    IMEMode      = 3  'DISABLE
    Left         = 360
    TabIndex     = 4
    Text         = "Text5"
    ToolTipText  = "Passwort für Optionen"
    Top          = 1560
    Width        = 1215
End
Begin VB.TextBox Text4
  BeginProperty Font
    Name         = "MS Sans Serif"
    Size         = 9.75
    Charset      = 0
    Weight       = 400
    Underline    = 0  'False
    Italic       = 0  'False
    Strikethrough = 0  'False
  EndProperty
  Height       = 285
  Left         = 360
  TabIndex     = 3
  Text         = "Text4"
  ToolTipText  = "maximale binäre Drehzahl"
  Top          = 1200
  Width        = 1215
End
Begin VB.TextBox Text3
  BeginProperty Font
    Name         = "MS Sans Serif"
    Size         = 9.75
    Charset      = 0
    Weight       = 400
    Underline    = 0  'False
    Italic       = 0  'False
    Strikethrough = 0  'False
  EndProperty
  Height       = 285
  Left         = 360
  TabIndex     = 2
  Text         = "Text3"
  ToolTipText  = "Schleifenzähler für den Empfang der seriellen
Daten!"
  Top          = 840
  Width        = 1215
End
Begin VB.TextBox Text2
  BeginProperty Font
    Name         = "MS Sans Serif"
    Size         = 9.75
    Charset      = 0
    Weight       = 400
    Underline    = 0  'False
    Italic       = 0  'False
    Strikethrough = 0  'False
  EndProperty
  Height       = 285
  Left         = 360
  TabIndex     = 1
  Text         = "Text2"
  ToolTipText  = "minimale mögliche Drehzahl"
  Top          = 480
  Width        = 1215
End
```

```
Begin VB.TextBox Text1
  BeginProperty Font
    Name           = "MS Sans Serif"
    Size           = 9.75
    Charset        = 0
    Weight         = 400
    Underline      = 0 'False
    Italic         = 0 'False
    Strikethrough  = 0 'False
  EndProperty
  Height          = 285
  Left            = 360
  TabIndex        = 0
  Text            = "Text1"
  ToolTipText     = "maximale mögliche Drehzahl"
  Top             = 120
  Width           = 1215
End
Begin VB.CommandButton cmdOK
  Caption         = "OK"
  BeginProperty Font
    Name           = "MS Sans Serif"
    Size           = 9.75
    Charset        = 0
    Weight         = 700
    Underline      = 0 'False
    Italic         = 0 'False
    Strikethrough  = 0 'False
  EndProperty
  Height          = 375
  Left            = 2520
  TabIndex        = 9
  Top             = 2640
  Width           = 1695
End
Begin VB.CommandButton cmdAbbrechen
  Caption         = "Abbrechen"
  BeginProperty Font
    Name           = "MS Sans Serif"
    Size           = 9.75
    Charset        = 0
    Weight         = 700
    Underline      = 0 'False
    Italic         = 0 'False
    Strikethrough  = 0 'False
  EndProperty
  Height          = 375
  Left            = 5400
  TabIndex        = 10
  Top             = 2640
  Width           = 1695
End
Begin VB.Label Label11
  Caption         = "Max. Motorstrom (mA)"
  BeginProperty Font
    Name           = "MS Sans Serif"
    Size           = 9.75
    Charset        = 0
    Weight         = 400
    Underline      = 0 'False
    Italic         = 0 'False
    Strikethrough  = 0 'False
  EndProperty
```



```
        Height      = 255
        Left        = 5880
        TabIndex    = 27
        ToolTipText = "Korrekturfaktor zur Anpassung des
Motorstromes"
        Top        = 1200
        Width      = 2655
    End
    Begin VB.Label Label10
        Caption      = "Korrekturfaktor (Trafotemp.Mot.)"
        BeginProperty Font
            Name      = "MS Sans Serif"
            Size      = 9.75
            Charset   = 0
            Weight    = 400
            Underline = 0 'False
            Italic    = 0 'False
            Strikethrough = 0 'False
        EndProperty
        Height      = 255
        Left        = 5880
        TabIndex    = 25
        ToolTipText = "Korrekturfaktor zur Anpassung der
Trafotemperatur"
        Top        = 480
        Width      = 2895
    End
    Begin VB.Label PaarNr2
        Caption      = "bis: "
        Height      = 255
        Left        = 4440
        TabIndex    = 23
        Top        = 2160
        Width      = 1815
    End
    Begin VB.Label WertPaar
        Caption      = "Korrekturwert:"
        Height      = 255
        Left        = 5280
        TabIndex    = 22
        Top        = 1560
        Width      = 1215
    End
    Begin VB.Label PaarNr1
        Caption      = "von:"
        Height      = 255
        Left        = 4440
        TabIndex    = 21
        Top        = 1560
        Width      = 735
    End
    Begin VB.Label Label9
        Caption      = "Korrekturfaktor (Drehzahl)"
        BeginProperty Font
            Name      = "MS Sans Serif"
            Size      = 9.75
            Charset   = 0
            Weight    = 400
            Underline = 0 'False
            Italic    = 0 'False
            Strikethrough = 0 'False
        EndProperty
        Height      = 255
```

```
Left          = 6360
TabIndex     = 19
ToolTipText  = "Korrekturfaktor zur Anpassung der Drehzahl"
Top          = 1800
Width        = 2535
End
Begin VB.Label Label8
Caption      = "Korrekturfaktor (Motorstrom)"
BeginProperty Font
    Name      = "MS Sans Serif"
    Size      = 9.75
    Charset   = 0
    Weight    = 400
    Underline = 0 'False
    Italic    = 0 'False
    Strikethrough = 0 'False
EndProperty
Height      = 255
Left        = 5880
TabIndex    = 18
ToolTipText = "Korrekturfaktor zur Anpassung des
Motorstromes"
Top         = 840
Width      = 2655
End
Begin VB.Label Label7
Caption      = "Korrekturfaktor (Trafotemp.Steu.)"
BeginProperty Font
    Name      = "MS Sans Serif"
    Size      = 9.75
    Charset   = 0
    Weight    = 400
    Underline = 0 'False
    Italic    = 0 'False
    Strikethrough = 0 'False
EndProperty
Height      = 255
Left        = 5880
TabIndex    = 17
ToolTipText = "Korrekturfaktor zur Anpassung der
Trafotemperatur"
Top         = 120
Width      = 3015
End
Begin VB.Label Label6
Caption      = "Relaisverzögerung (sec.)"
BeginProperty Font
    Name      = "MS Sans Serif"
    Size      = 9.75
    Charset   = 0
    Weight    = 400
    Underline = 0 'False
    Italic    = 0 'False
    Strikethrough = 0 'False
EndProperty
Height      = 255
Left        = 1800
TabIndex    = 16
ToolTipText = "Verzögerungszeit für die Relais (in x.xx
Sekunden)"
Top         = 1920
Width      = 2655
End
```

```
Begin VB.Label Label5
Caption      = "Passwort"
BeginProperty Font
    Name      = "MS Sans Serif"
    Size      = 9.75
    Charset   = 0
    Weight    = 400
    Underline = 0 'False
    Italic    = 0 'False
    Strikethrough = 0 'False
EndProperty
Height      = 255
Left        = 1800
TabIndex    = 15
ToolTipText = "Passwort für Otionen"
Top         = 1560
Width       = 2175
End
Begin VB.Label Label4
Caption      = "binäre max. Drehzahl"
BeginProperty Font
    Name      = "MS Sans Serif"
    Size      = 9.75
    Charset   = 0
    Weight    = 400
    Underline = 0 'False
    Italic    = 0 'False
    Strikethrough = 0 'False
EndProperty
Height      = 255
Left        = 1800
TabIndex    = 14
ToolTipText = "maximale binäre Drehzahl"
Top         = 1200
Width       = 2175
End
Begin VB.Label Label3
Caption      = "ser. Schleifenzähler"
BeginProperty Font
    Name      = "MS Sans Serif"
    Size      = 9.75
    Charset   = 0
    Weight    = 400
    Underline = 0 'False
    Italic    = 0 'False
    Strikethrough = 0 'False
EndProperty
Height      = 255
Left        = 1800
TabIndex    = 13
ToolTipText = "Schleifenzähler für den Empfang der seriellen
Daten!"
Top         = 840
Width       = 2175
End
Begin VB.Label Label2
Caption      = "minimale Drehzahl"
BeginProperty Font
    Name      = "MS Sans Serif"
    Size      = 9.75
    Charset   = 0
    Weight    = 400
    Underline = 0 'False
```

```

        Italic          = 0   'False
        Strikethrough   = 0   'False
    EndProperty
    Height              = 255
    Left                = 1800
    TabIndex            = 12
    ToolTipText         = "minimale mögliche Drehzahl"
    Top                 = 480
    Width               = 1935
End
Begin VB.Label Label1
    Caption              = "maximale Drehzahl"
    BeginProperty Font
        Name              = "MS Sans Serif"
        Size              = 9.75
        Charset           = 0
        Weight            = 400
        Underline         = 0   'False
        Italic            = 0   'False
        Strikethrough     = 0   'False
    EndProperty
    Height              = 255
    Left                = 1800
    TabIndex            = 11
    ToolTipText         = "maximale mögliche Drehzahl"
    Top                 = 120
    Width               = 1935
End
End

```

FrmPasswort (Programmcode)

```

Private Sub cmdAbbrechen_Click()
    Text1.text = ""
    PW1 = ""
    Unload frmPasswort
End Sub

Private Sub cmdOK_Click()
    PW1 = Text1.text
    Unload frmPasswort
End Sub

Private Sub Text1_KeyPress(KeyAscii As Integer)
    Select Case KeyAscii
        Case 13
            cmdOK_Click
        Case 27
            cmdAbbrechen_Click
    End Select
End Sub

```

FrmPasswort (Fensterdefinition)

```

VERSION 5.00
Begin VB.Form frmPasswort
    BackColor           = &H000000FF&
    BorderStyle         = 1   'Fest Einfach
    Caption              = "Passwort"
    ClientHeight        = 2220

```

```
ClientLeft      = 45
ClientTop       = 330
ClientWidth     = 3870
ClipControls    = 0 'False
ControlBox      = 0 'False
LinkTopic       = "Form1"
MaxButton       = 0 'False
MinButton       = 0 'False
ScaleHeight     = 2220
ScaleWidth      = 3870
StartPosition   = 1 'Fenstermitte
Begin VB.TextBox Text1
  BeginProperty Font
    Name          = "MS Sans Serif"
    Size          = 12
    Charset       = 0
    Weight        = 400
    Underline     = 0 'False
    Italic        = 0 'False
    Strikethrough = 0 'False
  EndProperty
  Height         = 360
  IMEMode        = 3 'DISABLE
  Left           = 840
  PasswordChar   = "*"
  TabIndex       = 0
  Text           = "Text1"
  Top            = 1095
  Width          = 2295
End
Begin VB.CommandButton cmdAbbrechen
  Caption        = "Abbrechen"
  Height         = 375
  Left           = 2280
  TabIndex       = 2
  Top            = 1680
  Width          = 1335
End
Begin VB.CommandButton cmdOK
  Caption        = "OK"
  Height         = 375
  Left           = 240
  TabIndex       = 1
  Top            = 1680
  Width          = 1335
End
Begin VB.Label Label1
  Alignment      = 2 'Zentriert
  BackColor      = &H000000FF&
  Caption        = "Bitte geben Sie das Passwort ein!"
  BeginProperty Font
    Name          = "MS Sans Serif"
    Size          = 9.75
    Charset       = 0
    Weight        = 400
    Underline     = 0 'False
    Italic        = 0 'False
    Strikethrough = 0 'False
  EndProperty
  Height         = 615
  Left           = 240
  TabIndex       = 3
  Top            = 240
```

```
        Width           = 3375
    End
End
```

FrmProjektDaten (Programmcode)

```
Private Sub cmdAbbrechen_Click()
    Unload frmProjektDaten
End Sub

Private Sub cmdAbbrechen_KeyPress(KeyAscii As Integer)
    If KeyAscii = 27 Then
        cmdAbbrechen_Click
    End If
End Sub

Private Sub cmdOK_Click()
    flgGeändert = True
    If (Right(frmSteuerung.Caption, 1) <> "*") Then
        frmSteuerung.Caption = frmSteuerung.Caption & "*"
    End If
    KD_nr = Text1.text
    KD_adr1 = Text2.text
    KD_adr2 = Text3.text
    KD_str = Text4.text
    KD_ort = Text5.text
    KD_prNr = Text6.text
    PR_dat = Text7.text
    Unload frmProjektDaten
End Sub

Private Sub cmdOK_KeyPress(KeyAscii As Integer)
    If KeyAscii = 27 Then
        cmdAbbrechen_Click
    End If
End Sub

Private Sub Form_Load()
    If PR_dat = "" Then
        PR_dat = Date
    End If
    Text1.text = KD_nr
    Text2.text = KD_adr1
    Text3.text = KD_adr2
    Text4.text = KD_str
    Text5.text = KD_ort
    Text6.text = KD_prNr
    Text7.text = PR_dat
End Sub

Private Sub Text1_KeyPress(KeyAscii As Integer)
    Select Case KeyAscii
        Case 13
            Text2.SetFocus
        Case 27
            cmdAbbrechen_Click
    End Select
End Sub

Private Sub Text2_KeyPress(KeyAscii As Integer)
    Select Case KeyAscii
```

```
Case 13
  Text3.SetFocus
Case 27
  cmdAbbrechen_Click
End Select
End Sub

Private Sub Text3_KeyPress(KeyAscii As Integer)
  Select Case KeyAscii
    Case 13
      Text4.SetFocus
    Case 27
      cmdAbbrechen_Click
  End Select
End Sub

Private Sub Text4_KeyPress(KeyAscii As Integer)
  Select Case KeyAscii
    Case 13
      Text5.SetFocus
    Case 27
      cmdAbbrechen_Click
  End Select
End Sub

Private Sub Text5_KeyPress(KeyAscii As Integer)
  Select Case KeyAscii
    Case 13
      Text6.SetFocus
    Case 27
      cmdAbbrechen_Click
  End Select
End Sub

Private Sub Text6_KeyPress(KeyAscii As Integer)
  Select Case KeyAscii
    Case 13
      Text7.SetFocus
    Case 27
      cmdAbbrechen_Click
  End Select
End Sub

Private Sub Text7_KeyPress(KeyAscii As Integer)
  Select Case KeyAscii
    Case 13
      cmdOK.SetFocus
    Case 27
      cmdAbbrechen_Click
  End Select
End Sub
```

FrmProjektDaten (Fensterdefinition)

```
VERSION 5.00
Begin VB.Form frmProjektDaten
  Caption       = "Projekt - Daten"
  ClientHeight = 3675
  ClientLeft   = 60
  ClientTop    = 345
  ClientWidth  = 6495
```

```
LinkTopic      = "Form1"
ScaleHeight    = 3675
ScaleWidth     = 6495
StartupPosition = 3 'Windows-Standard
Begin VB.TextBox Text1
    Height      = 285
    Left        = 1320
    TabIndex    = 0
    Text        = "Text1"
    Top         = 240
    Width       = 1815
End
Begin VB.TextBox Text2
    Height      = 285
    Left        = 1320
    TabIndex    = 1
    Text        = "Text2"
    Top         = 720
    Width       = 5000
End
Begin VB.TextBox Text3
    Height      = 285
    Left        = 1320
    TabIndex    = 2
    Text        = "Text3"
    Top         = 1080
    Width       = 5000
End
Begin VB.TextBox Text4
    Height      = 285
    Left        = 1320
    TabIndex    = 3
    Text        = "Text4"
    Top         = 1440
    Width       = 5000
End
Begin VB.TextBox Text5
    Height      = 285
    IMEMode     = 3 'DISABLE
    Left        = 1320
    TabIndex    = 4
    Text        = "Text5"
    Top         = 1800
    Width       = 5000
End
Begin VB.TextBox Text6
    Height      = 285
    IMEMode     = 3 'DISABLE
    Left        = 1320
    TabIndex    = 5
    Text        = "Text6"
    Top         = 2160
    Width       = 1815
End
Begin VB.TextBox Text7
    Height      = 285
    IMEMode     = 3 'DISABLE
    Left        = 1320
    TabIndex    = 6
    Text        = "Text7"
    Top         = 2520
    Width       = 1815
End
```



```
Begin VB.CommandButton cmdAbbrechen
Caption      = "Abbrechen"
BeginProperty Font
    Name      = "MS Sans Serif"
    Size      = 9.75
    Charset   = 0
    Weight    = 700
    Underline = 0 'False
    Italic    = 0 'False
    Strikethrough = 0 'False
EndProperty
Height      = 375
Left        = 3840
TabIndex    = 8
ToolTipText = "Abbrechen"
Top         = 3120
Width       = 1695
End
Begin VB.CommandButton cmdOK
Caption      = "OK"
BeginProperty Font
    Name      = "MS Sans Serif"
    Size      = 9.75
    Charset   = 0
    Weight    = 700
    Underline = 0 'False
    Italic    = 0 'False
    Strikethrough = 0 'False
EndProperty
Height      = 375
Left        = 840
TabIndex    = 7
ToolTipText = "Beenden und Daten übernehmen"
Top         = 3120
Width       = 1695
End
Begin VB.Label Label1
Caption      = "Kunden Nr."
BeginProperty Font
    Name      = "MS Sans Serif"
    Size      = 9.75
    Charset   = 0
    Weight    = 400
    Underline = 0 'False
    Italic    = 0 'False
    Strikethrough = 0 'False
EndProperty
Height      = 255
Left        = 120
TabIndex    = 15
Top         = 240
Width       = 1000
End
Begin VB.Label Label2
Caption      = "Name"
BeginProperty Font
    Name      = "MS Sans Serif"
    Size      = 9.75
    Charset   = 0
    Weight    = 400
    Underline = 0 'False
    Italic    = 0 'False
    Strikethrough = 0 'False
```

```
EndProperty
Height      = 255
Left       = 120
TabIndex   = 14
Top        = 720
Width      = 1000
End
Begin VB.Label Label3
Caption     = "Zusatz"
BeginProperty Font
Name       = "MS Sans Serif"
Size      = 9.75
Charset    = 0
Weight    = 400
Underline = 0 'False
Italic    = 0 'False
Strikethrough = 0 'False
EndProperty
Height     = 255
Left      = 120
TabIndex  = 13
Top       = 1080
Width    = 1000
End
Begin VB.Label Label4
Caption     = "Straße"
BeginProperty Font
Name       = "MS Sans Serif"
Size      = 9.75
Charset    = 0
Weight    = 400
Underline = 0 'False
Italic    = 0 'False
Strikethrough = 0 'False
EndProperty
Height     = 255
Left      = 120
TabIndex  = 12
Top       = 1440
Width    = 1000
End
Begin VB.Label Label5
Caption     = "PLZ Ort"
BeginProperty Font
Name       = "MS Sans Serif"
Size      = 9.75
Charset    = 0
Weight    = 400
Underline = 0 'False
Italic    = 0 'False
Strikethrough = 0 'False
EndProperty
Height     = 255
Left      = 120
TabIndex  = 11
Top       = 1800
Width    = 1000
End
Begin VB.Label Label6
Caption     = "Projekt Nr."
BeginProperty Font
Name       = "MS Sans Serif"
Size      = 9.75
```

```

        Charset          = 0
        Weight           = 400
        Underline        = 0   'False
        Italic           = 0   'False
        Strikethrough    = 0   'False
    EndProperty
    Height              = 255
    Left                = 120
    TabIndex            = 10
    Top                 = 2160
    Width               = 1000
End
Begin VB.Label Label7
    Caption             = "Datum"
    BeginProperty Font
        Name              = "MS Sans Serif"
        Size              = 9.75
        Charset           = 0
        Weight            = 400
        Underline        = 0   'False
        Italic           = 0   'False
        Strikethrough    = 0   'False
    EndProperty
    Height              = 255
    Left                = 120
    TabIndex            = 9
    Top                 = 2520
    Width               = 1000
End
End

```

FrmSteuerung (Programmcode)

```

Dim Tick_Zähler As Long
Dim OShilf As String
Dim Abbr_Meldung As Boolean
Dim OSshilf As Integer

Public Sub Maske_INIT()
    Abbr_Meldung = False
    Reset_OK = False
    mnuDrucken.Enabled = False
    mnuSpeichern.Enabled = False
    mnuStarten.Enabled = False
    mnuAbbrechen.Enabled = False
    mnuDatenfreigabe.Enabled = False
    cmdAbbrechen.Enabled = False
    cmdDatenfreigabe.Enabled = False
    cmdStart.Enabled = False
    cmdAbbrechen.Visible = False
    cmdDatenfreigabe.Visible = True
    cmdStart.Visible = False
    flgGeändert = False
    flgGestartet = False
    paarIdx = 0
    WerteKopf.Caption = "Paar           Drehzahl           Zeit
Ereignis   Drehzahl OK"
    WerteKopf.FontBold = True
    DateiName = "Unbenannt"
    TitelNeu (0)
    For x = 0 To 7

```

```

anzNr(x).Visible = True
anzDrehzahl(x).Visible = True
anzZeit(x).Visible = True
anzEreignis(x).Visible = True
imgDrehzahl(x).Visible = False
imgNotDrehzahl(x).Visible = False
anzNr(x).BackColor = &H80000000
anzDrehzahl(x).BackColor = &H80000000
anzZeit(x).BackColor = &H80000000
anzEreignis(x).BackColor = &H80000000
anzNr(x).ForeColor = QBColor(0)
anzDrehzahl(x).ForeColor = QBColor(0)
anzZeit(x).ForeColor = QBColor(0)
anzEreignis(x).ForeColor = QBColor(0)
WertNr(x) = x
WertDrehzahl(x) = 0
WertZeit(x) = 0
WertEreignis(x) = 0
anzNr(x) = "F" & x + 1 & " "
anzDrehzahl(x) = Format(WertDrehzahl(x), "##,##0") & " U/min. "
anzZeit(x) = WertZeit(x) & " Sekunden "
anzEreignis(x) = frmEingabe.ComboEreignis.List(WertEreignis(x))
Next x
Unload frmEingabe
End Sub

Public Sub Abbr_Meld(text As String)
  If Not Abbr_Meldung Then
    Abbr_Meldung = True
    MsgBox (text)
  End If
End Sub

Public Sub HwSt1()
Dim Hil2 As Integer
  Hil2 = Int((SendeVorgang("s", "e", "R", "sS", True) * Sfaktor) + 0.5)
  If Hil2 > MaxStromMotor Then
    cmdAbbrechen_Click
    Abbr_Meld ("ACHTUNG! Maximaler Motorsstrom (" & MaxStromMotor & " mA)
überschritten!!")
  Else
    HardwareStatus1.Caption = "Motor-Strom: " & Hil2 & " mA"
  End If
End Sub

Private Sub Pause(z As Single)
Dim sz As Single
Dim fFf1 As Integer
  sz = Timer
  fFf1 = 0
  Do While (Timer < (sz + z))
    DoEvents
    fFf1 = fFf1 + 1
    If ((fFf1 Mod 30) = 0) Then
      If fFf1 > 30000 Then ffff1 = 0
      HwSt1
    End If
  Loop
End Sub

Public Sub TitelNeu(ff As Integer)
  Select Case ff
    Case 1      'Datei geändert

```

```

    frmSteuerung.Caption = App.EXENAME & " " & App.Major & "." &
App.Minor & "." & App.Revision & " - [" & Dateiname & "]"*
    Case Else 'Normal
        frmSteuerung.Caption = App.EXENAME & " " & App.Major & "." &
App.Minor & "." & App.Revision & " - [" & Dateiname & "]"
    End Select
End Sub
Function SendeVorgang(s1, s2, s3, s4 As String, Sende As Boolean) As
Integer
Dim zZaeX, zZae As Integer
If Len(s4) = 2 Then 'Statusabfrage 2 Zeichen Länge
    zZaeX = 0
    Do
        zZaeX = zZaeX + 1
        'Debug.Print "zZaeX" & zZaeX & "*"
    If Sende Then
        OutString = s1
        ZumMotor
        OutString = s1
        ZumMotor
        OutString = s1
        ZumMotor
        OutString = s2
        ZumMotor
        OutString = s3
        ZumMotor
        If s3 = "s" Then
            OutString = "."
            ZumMotor
        End If
        OutString = ""
    End If
    InString = ""
    zZae = 0
    Do
        zZae = zZae + 1
        'Debug.Print "zZae" & zZae & "*" & serieller_Schleifenzähler & "*"
        OutString = ""
        ZumMotor
        DoEvents
    Loop Until (Left(InString, 2) = s4) Or flgAbbrechen Or (zZae >=
serieller_Schleifenzähler)
    If (zZae >= serieller_Schleifenzähler) Then
        OutString = "."
        ZumMotor
    Else
        If (Left(InString, 2) = s4) Then
            SendeVorgang = Asc(Right(InString, 1))
            Exit Do
        Else
            If (zZaeX >= serieller_Schleifenzähler) Then
                If Sende Then
                    Abbr_Meld ("Die serielle Datenübertragung wurde unterbrochen!
(Keine Sendebestätigung)")
                Else
                    Abbr_Meld ("Die serielle Datenübertragung wurde unterbrochen!
(Kein Empfang)")
                End If
            End If
            SendeVorgang = 0
            flgAbbrechen = True
            Exit Do
        End If
    End If
End Sub

```

```
End If
Loop Until ((Left(InString, 2) = s4) Or flgAbbrechen Or (zZaeX >=
serieller_Schleifenzähler))
Else 'Statusabfrage 3 Zeichen Länge
zZaeX = 0
Do
zZaeX = zZaeX + 1
'Debug.Print "zZaeX" & zZaeX & "*"
If Sende Then
OutString = s1
ZumMotor
OutString = s1
ZumMotor
OutString = s1
ZumMotor
OutString = s2
ZumMotor
OutString = s3
ZumMotor
If s3 = "s" Then
OutString = "x"
ZumMotor
End If
OutString = ""
End If
InString = ""
zZae = 0
Do
zZae = zZae + 1
If ((zZae Mod 8) = 0) Then
HwSt1
End If
Debug.Print "zZae" & zZae & "*" & serieller_Schleifenzähler & "*"
OutString = ""
ZumMotor
DoEvents
Loop Until (InString = s4) Or flgAbbrechen Or (zZae >=
serieller_Schleifenzähler)
If (zZae >= serieller_Schleifenzähler) Then
OutString = "."
ZumMotor
Else
If (InString = s4) Then
SendeVorgang = 1
Exit Do
Else
If Sende Then
Abbr_Meld ("Die serielle Datenübertragung wurde unterbrochen!
(Keine Sendebestätigung)")
Else
Abbr_Meld ("Die serielle Datenübertragung wurde unterbrochen!
(Kein Empfang)")
End If
SendeVorgang = 0
flgAbbrechen = True
Exit Do
End If
End If
Loop Until ((InString = s4) Or flgAbbrechen Or (zZaeX >=
serieller_Schleifenzähler))
End If 'if len(s4) =
End Function
```

```
Private Sub ZumMotor()  
  If OutString <> "" Then  
    MSComm1.Output = OutString  
    ' Debug.Print "OutString:" & OutString & ":"  
  End If  
  Select Case MSComm1.CommEvent  
    ' Ereignis-Meldungen  
    Case comEvReceive  
      InString = MSComm1.Input  
      Debug.Print "InString:" & InString & ":"  
    Case comEvSend  
  End Select  
  If (Not abbrechen) And (cmdStartReset.Visible = False) Then  
    Select Case InString  
      Case "Hey" 'Reset  
        cmdAbbrechen_Click  
        Abbr_Meld ("Reset an der Steuer-Platine!!")  
      Case "*B*" 'Stop  
        cmdAbbrechen_Click  
        Abbr_Meld ("Stop an der Steuer-Platine durch den Benutzer!!")  
      Case "Fel" 'Datenpaket unvollständig  
        cmdAbbrechen_Click  
        Abbr_Meld ("Fehler beim senden der Parameter!!")  
      Case "*G*" 'Start  
        If ((frmFertig.Visible = True) And (cmdDatenfreigabe.Enabled =  
True)) Or (cmdStart.Enabled = True) Then  
          Unload frmFertig  
          cmdStart_Click  
        Else  
          'MsgBox ("Sie müssen die Daten erst freigeben!!")  
        End If  
    End Select  
  End If  
End Sub  
  
Private Sub anzDrehzahl_Click(Index As Integer)  
  paarIdx = Index 'index des gewählten Paares  
  frmEingabe.Show vbModal  
End Sub  
  
Private Sub anzEreignis_Click(Index As Integer)  
  paarIdx = Index 'index des gewählten Paares  
  frmEingabe.Show vbModal  
End Sub  
  
Private Sub anzNr_Click(Index As Integer)  
  paarIdx = Index 'index des gewählten Paares  
  frmEingabe.Show vbModal  
  
End Sub  
  
Private Sub anzZeit_Click(Index As Integer)  
  paarIdx = Index 'index des gewählten Paares  
  frmEingabe.Show vbModal  
End Sub  
  
Private Sub cmdAbbrechen_Click()  
  If Not abbrechen Then  
    MSComm1.InBufferCount = 0  
    InString = ""  
    asw = SendeVorgang("s", "e", "X", "*X*", True)  
    Print #5, "Benutzerabbruch! ! ! !"  
    cmdAbbrechen.Left = 360
```

```
cmdAbbrechen.Visible = False
cmdAbbrechen.Enabled = False
mnuAbbrechen.Enabled = False
cmdEnde.Visible = True
cmdDatenfreigabe.Enabled = True
cmdDatenfreigabe.Visible = True
mnuDatenfreigabe.Enabled = True
cmdStart.Enabled = False
cmdStart.Visible = False
mnuStarten.Enabled = False
flgAbbrechen = True      'Flag zur Prüfung ob AbbrechenButton betätigt
wurde
  InString = ""
  MSComm1.InBufferCount = 0
End If
End Sub

Private Sub cmdDatenfreigabe_Click()
Dim il As Integer
  Abbr_Meldung = False
  flgAbbrechen = False
  il = Sendevorgang("s", "e", "u", "*u*", True)
  flgGestartet = False
  MSComm1.InBufferCount = 0
  InString = ""
  OutString = ""
  ZumMotor
  cmdStart.Enabled = True
  cmdStart.Visible = True
  mnuStarten.Enabled = True
  cmdDatenfreigabe.Enabled = False
  cmdDatenfreigabe.Visible = False
  mnuDatenfreigabe.Enabled = False
  cmdAbbrechen.Left = 5760
  cmdAbbrechen.Visible = True
  cmdAbbrechen.Enabled = True
  mnuAbbrechen.Enabled = True
  cmdEnde.Visible = False
  cmdStart.SetFocus
  InString = ""
  MSComm1.InBufferCount = 0
  Do
    DoEvents
    OutString = ""
    ZumMotor
  Loop Until ((flgGestartet) Or (InString = "*G*") Or (flgAbbrechen))
  MSComm1.InBufferCount = 0
  InString = ""
End Sub

Private Sub cmdEnde_Click()
  If MsgBox("Wollen Sie wirklich beenden?", vbOKCancel + vbQuestion,
"Achtung!") = 1 Then
    If (flgGeändert) Then
      If (MsgBox("Wollen Sie die Daten speichern?", vbYesNo + vbQuestion,
"Achtung!") = 6) Then
        mnuSpeichern_Click
      End If
    End If
  End If
  MSComm1.InBufferCount = 0
  InString = ""
  MSComm1.PortOpen = False
  If Err Then
```



```
        MsgBox Error$, 48
        imgConnect.Visible = True
        imgNotConnect.Visible = False
    Else
        imgConnect.Visible = False
        imgNotConnect.Visible = True
    End If
    Close #5
    End 'ProgrammEnde
End If
End Sub

Private Sub cmdStart_Click()
    Dim i1, I2 As Integer
    'MSComm1.InBufferCount = 0
    'InString = ""
    Abbr_Meldung = False
    Print #5, Dateiname
    Print #5, ""
    MSComm1.InBufferCount = 0
    InString = ""
    flgGestartet = True
    AnzeigeAktualisieren
    flgAbbrechen = False
    cmdStart.Visible = False
    cmdEnde.Visible = True
    cmdAbbrechen.Left = 360
    cmdAbbrechen.Visible = True
    mnuAbbrechen.Enabled = True
    'cmdAbbrechen.SetFocus
    mnuStarten.Enabled = False
    mnuDaten.Enabled = False
    mnuDatei.Enabled = False
    mnuKonfiguration.Enabled = False
    lblFortschritt.Visible = True
    ProgressBar1.Top = 3360
    ProgressBar1.Height = 400
    ProgressBar1.Visible = True
    ProgressBar1.Min = 0
    ProgressBar1.Max = 100
    ProgressBar1.Value = ProgressBar1.Min
    cmdEnde.Enabled = False
    cmdStart.Enabled = False
    For x = 0 To 7 ' setzt die Farben auf Standard
        anzNr(x).BackColor = &H80000000
        anzDrehzahl(x).BackColor = &H80000000
        anzZeit(x).BackColor = &H80000000
        anzEreignis(x).BackColor = &H80000000
        anzNr(x).ForeColor = QBColor(0)
        anzDrehzahl(x).ForeColor = QBColor(0)
        anzZeit(x).ForeColor = QBColor(0)
        anzEreignis(x).ForeColor = QBColor(0)
    Next x
    For x = 0 To 7 'Schleife für Pgm-Ausführung
        HardwareStatus.Caption = "Travo-Steuerung-Temperatur: " &
        Int((SendeVorgang("s", "e", "T", "sT", True) * TSfaktor) + 0.5) & "°C
        Travo-Motor-Temperatur: " & Int((SendeVorgang("s", "e", "V", "sV", True)
        * TMfaktor) + 0.5) & " °C"
        HwSt1
        If WertEreignis(x) > 0 Then
            anzNr(x).BackColor = QBColor(4)
            anzDrehzahl(x).BackColor = QBColor(4)
            anzZeit(x).BackColor = QBColor(4)
```

```
anzEreignis(x).BackColor = QBColor(4)
anzNr(x).ForeColor = QBColor(7)
anzDrehzahl(x).ForeColor = QBColor(7)
anzZeit(x).ForeColor = QBColor(7)
anzEreignis(x).ForeColor = QBColor(7)
lblFortschritt.Caption = " Fortschritt von Paar Nr. " & x + 1 & " ( " &
anzEreignis(x) & "):"
imgDrehzahl(x).Visible = False
imgNotDrehzahl(x).Visible = True
' Schleife für ein Drehzahlpaar
Tick_Zähler = 0
For i1 = 1 To (AnzahlDrehzahlen - 1)
    If ((WertDrehzahl(x) >= DrehzahlBereich(i1)) And (WertDrehzahl(x) <=
DrehzahlBereich(i1 + 1))) Then
        I2 = i1
        Exit For
    End If
Next i1
' Berechnen der Drehzahl
OSshilf = Int(DrehzahlKorrektur(I2) * MaxChar / MaxDrehzahl *
WertDrehzahl(x))
If OSshilf >= MaxChar Then
    OSshilf = MaxChar
End If
If OSshilf <= 0 Then
    OSshilf = 0
End If
OShilf = Chr(OSshilf)
' Senden der Drehzahl
If Sendevorgang("s", "d", OShilf, "*" & OShilf & "*", True) Then
    Print #5, "Drehzahl No." & (x + 1) & " (" & WertDrehzahl(x) & "
U/min.)" & " gesendet"
Else
    Print #5, ">>Fehler<< Drehzahl No." & (x + 1) & " (" &
WertDrehzahl(x) & " U/min.)" & " gesendet"
End If
HwSt1
If Sendevorgang("s", "d", OShilf, "*" & OShilf & "*", False) Then
    Print #5, "Drehzahl " & (x + 1) & " erreicht"
Else
    Print #5, ">>Fehler<< Drehzahl " & (x + 1) & " erreicht"
End If
Pause (RELpause)
HwSt1
'ProgressBar1.Value = 0
imgDrehzahl(x).Visible = True
imgNotDrehzahl(x).Visible = False
test2.Caption = test2.Caption & InString & "*"
' Ereignis auslösen
If Not flgAbbrechen Then
    Select Case WertEreignis(x)
    Case 0 "keine "
    Case 1 "Entwickler ein"
        If (x > 0) Then
            If (WertEreignis(x - 1) <> 1) And (WertEreignis(x - 1) <> 2) Then
                If Sendevorgang("s", "e", "A", "*A*", True) Then
                    Print #5, "Schwenkarm EINSchalten gesendet"
                Else
                    Print #5, ">>Fehler<< Schwenkarm EINSchalten gesendet"
                End If
            End If
        End If
    Else
        If Sendevorgang("s", "e", "A", "*A*", True) Then
```

```
        Print #5, "Schwenkarm EINSchalten gesendet"
    Else
        Print #5, ">>Fehler<< Schwenkarm EINSchalten gesendet"
    End If
End If
Pause (RELpause)
If Sendevorgang("s", "e", "E", "*E*", True) Then
    Print #5, "Entwickler EINSchalten gesendet"
Else
    Print #5, ">>Fehler<< Entwickler EINSchalten gesendet"
End If
InString = ""
Case 2 "Spülung ein"
    If (x > 0) Then
        If (WertEreignis(x - 1) <> 1) And (WertEreignis(x - 1) <> 2) Then
            If Sendevorgang("s", "e", "A", "*A*", True) Then
                Print #5, "Schwenkarm EINSchalten gesendet"
            Else
                Print #5, ">>Fehler<< Schwenkarm EINSchalten gesendet"
            End If
        End If
    Else
        If Sendevorgang("s", "e", "A", "*A*", True) Then
            Print #5, "Schwenkarm EINSchalten gesendet"
        Else
            Print #5, ">>Fehler<< Schwenkarm EINSchalten gesendet"
        End If
    End If
    Pause (RELpause)
    If Sendevorgang("s", "e", "S", "*S*", True) Then
        Print #5, "Spülung EINSchalten gesendet"
    Else
        Print #5, ">>Fehler<< Spülung EINSchalten gesendet"
    End If
    InString = ""
Case 3 "trocknen "
    Print #5, "Trockenvorgang aktiviert"
End Select
End If 'flgAbbrechen
' Timer setzen
HwSt1
zael = 0
zae2 = WertZeit(x) / 100
startzeit = Timer
pauseZeit = WertZeit(x)
DoEvents
Do While (Timer < startzeit + pauseZeit) And (Not flgAbbrechen)
    'test.Caption = Timer
    If Timer > startzeit + zae2 Then
        'test1.Caption = zae2
        zae2 = zae2 + (WertZeit(x) / 100)
        zael = zael + 1
        ProgressBar1.Value = zael
        HwSt1
    End If
    OutString = ""
    ZumMotor
    Select Case InString
        Case "#m#"
            Abbr_Meld ("Es ist ein Fehler am Motor aufgetreten!" & Chr(13) &
"Prozess wurde Abgebrochen!!!")
            Print #5, "Es wurde ein Motorfehler von der Steuerung empfangen!"
            cmdAbbrechen_Click
```

```
Case "#e#"
    Abbr_Meld ("Es ist ein Fehler beim Entwickler aufgetreten!" &
Chr(13) & "Prozess wurde Abgebrochen!!!")
    Print #5, "Es wurde ein Entwicklerfehler von der Steuerung
empfangen!"
    cmdAbbrechen_Click
Case "#s#"
    Abbr_Meld ("Es ist ein Fehler bei der Spülung aufgetreten!" &
Chr(13) & "Prozess wurde Abgebrochen!!!")
    Print #5, "Es wurde ein Spülungsfehler von der Steuerung
empfangen!"
    cmdAbbrechen_Click
Case "#v#"frmSteuerung - 9

    Abbr_Meld ("Es ist ein Fehler beim Vakuum aufgetreten!" & Chr(13)
& "Prozess wurde Abgebrochen!!!")
    Print #5, "Es wurde ein Vakuumfehler von der Steuerung
empfangen!"
    cmdAbbrechen_Click
Case "#n#"
    Abbr_Meld ("Es ist ein Fehler 'NOTSTOP' aufgetreten!" & Chr(13) &
"Prozess wurde Abgebrochen!!!")
    Print #5, "Es wurde ein NOTSTOP von der Steuerung empfangen!"
    cmdAbbrechen_Click
End Select
DoEvents
Loop 'Do While (Timer...
ProgressBar1.Value = 0      'Fortschrittsbalken auf 0
' Ereignis löschen
HwSt1
If Not flgAbbrechen Then
    Select Case WertEreignis(x)
    Case 0 "keine "
    Case 1 "Entwickler aus"
        If Sendevorgang("s", "e", "e", "*e*", True) Then
            Print #5, "Entwickler AUSSchalten gesendet"
        Else
            Print #5, ">>Fehler<< Entwickler AUSSchalten gesendet"
        End If
    Case 2 "Spülung aus"
        If x < 7 Then
            If (WertEreignis(x + 1) <> 1) And (WertEreignis(x + 1) <> 2) Then
                Pause (RELpause)
                If Sendevorgang("s", "e", "a", "*a*", True) Then
                    Print #5, "Schwenkarm AUSSchalten gesendet"
                Else
                    Print #5, ">>Fehler<< Schwenkarm AUSSchalten gesendet"
                End If
            End If
        Else
            Pause (RELpause)
            If Sendevorgang("s", "e", "a", "*a*", True) Then
                Print #5, "Schwenkarm AUSSchalten gesendet"
            Else
                Print #5, ">>Fehler<< Schwenkarm AUSSchalten gesendet"
            End If
        End If
    End Select
    InString = ""
    Case 2 "Spülung aus"
        If Sendevorgang("s", "e", "s", "*s*", True) Then
            Print #5, "Spülung AUSSchalten gesendet"
        Else
            Print #5, ">>Fehler<< Spülung AUSSchalten gesendet"
```

```
End If
If x < 7 Then
  If (WertEreignis(x + 1) <> 1) And (WertEreignis(x + 1) <> 2) Then
    Pause (RELpause)
    If Sendevorgang("s", "e", "a", "*a*", True) Then
      Print #5, "Schwenkarm AUSSchalten gesendet"
    Else
      Print #5, ">>Fehler<< Schwenkarm AUSSchalten gesendet"
    End If
  End If
Else
  Pause (RELpause)
  If Sendevorgang("s", "e", "a", "*a*", True) Then
    Print #5, "Schwenkarm AUSSchalten gesendet"
  Else
    Print #5, ">>Fehler<< Schwenkarm AUSSchalten gesendet"
  End If
End If
InString = ""
Case 3 "trocknen "
  Print #5, "Trockenvorgang DEaktiviert"
End Select
End If
anzNr(x).Caption = Tick_Zähler
If flgAbbrechen Then
  anzNr(x).BackColor = &H80000000
  anzDrehzahl(x).BackColor = &H80000000
  anzZeit(x).BackColor = &H80000000
  anzEreignis(x).BackColor = QBColor(8)
  anzNr(x).ForeColor = QBColor(0)
  anzDrehzahl(x).ForeColor = QBColor(0)
  anzZeit(x).ForeColor = QBColor(0)
  anzEreignis(x).ForeColor = QBColor(0)
Else
  anzNr(x).BackColor = &H80000000
  anzDrehzahl(x).BackColor = &H80000000
  anzZeit(x).BackColor = &H80000000
  anzEreignis(x).BackColor = QBColor(10)
  anzNr(x).ForeColor = QBColor(0)
  anzDrehzahl(x).ForeColor = QBColor(0)
  anzZeit(x).ForeColor = QBColor(0)
  anzEreignis(x).ForeColor = QBColor(0)
End If
End If
If flgAbbrechen Then Exit For
Next x 'Die einzelnen Paare
  ' Ende Drehzahl noch auf 0 setzen
  If Sendevorgang("s", "e", "X", "*X*", True) Then
    Print #5, "Drehzahl ENDE erfolgreich gesendet"
  Else
    Print #5, ">>Fehler<< Drehzahl ENDE erfolgreich gesendet"
  End If
  ' OK Drehzahl 0
If Not flgAbbrechen Then
  Print #5, " F E R T I G ! ! ! !"
  frmFertig.Show
Else
  Print #5, " A B B R U C H ! ! ! !"
  Abbr_Meld ("Benutzerabbruch! ! ! !")
End If
cmdStart.Visible = True
cmdStart.Enabled = True
mnuStarten.Enabled = True
```

```
cmdDatenfreigabe.Enabled = False
cmdDatenfreigabe.Visible = False
mnuDatenfreigabe.Enabled = False
cmdAbbrechen.Left = 5760
cmdAbbrechen.Visible = True
cmdAbbrechen.Enabled = True
mnuAbbrechen.Enabled = True
cmdEnde.Visible = False
cmdStart.SetFocus
lblFortschritt.Visible = False
ProgressBar1.Top = 3360
ProgressBar1.Height = 200
ProgressBar1.Visible = False
lblFortschritt.Caption = " Fortschritt:"
HardwareStatus.Caption = "Travo-Steuerung-Temperatur: " &
Int((SendeVorgang("s", "e", "T", "sT", True) * TSfaktor) + 0.5) & "°C
Travo-Motor-Temperatur: " & Int((SendeVorgang("s", "e", "V", "sV", True)
* TMfaktor) + 0.5) & " °C"
HwSt1
Do Until ((frmFertig.Visible = False) Or (InString = "*G*") Or
(flglAbbrechen))
    DoEvents
    OutStrin3g = ""
    ZumMotor
Loop
If flglAbbrechen Then
    ' Form Steuerung auf Anfng setzten
    cmdDatenfreigabe.Visible = True
    cmdAbbrechen.Visible = False
    cmdEnde.Enabled = True
    cmdEnde.Visible = True
    cmdDatenfreigabe.Enabled = True
    mnuDatenfreigabe.Enabled = True
    mnuAbbrechen.Enabled = False
    mnuDaten.Enabled = True
    mnuDatei.Enabled = True
    mnuKonfiguration.Enabled = True
    lblFortschritt.Visible = False
    ProgressBar1.Top = 3360
    ProgressBar1.Height = 200
    ProgressBar1.Visible = False
    lblFortschritt.Caption = " Fortschritt:"
    cmdDatenfreigabe.SetFocus
    cmdStart.Enabled = False
    cmdStart.Visible = False
    mnuStarten.Enabled = False
End If
End Sub

Private Sub cmdStartReset_Click()
    If (cmdStartReset.Caption = "Abbrechen") Then
        Shapel.Visible = False
        cmdStartReset.Visible = False
        lblStartResetOK.Visible = False
        lblStartReset.Visible = False
        MsgBox "Sie haben an der Steuerplatine keinen Reset gedrückt." &
Chr(13) & "Die Kommunikation zwischen PC und Steuerplatine konnte nicht
geprüft werden!"
    End If
    If (cmdStartReset.Caption = "OK") Then
        Shapel.Visible = False
        cmdStartReset.Visible = False
        lblStartResetOK.Visible = False
    End If
End Sub
```

```
        lblStartReset.Visible = False
    End If
End Sub

Private Sub Form_Load()
    Dim x, dfg As Integer
    Dim ZW_DM As String
    'INIT Grunddaten
    Maske_INIT
    imgConnect.Visible = False
    imgNotConnect.Visible = True
    lblConnect.Caption = "Steuerplatine nicht verbunden (Com" & ComPort + 1
& "):"
    Debug.Print "Programmstart"
    'Laden der CFG Daten
    CFG_Lesen
    MSComm1.CommPort = ComPort + 1 'ComPort(0..3)
    Open "Steuerung.tmp" For Output As #5
    ' COM-Port
    MSComm1.CommPort = ComPort + 1
    MSComm1.Settings = "19200,N,8,1" "9600,N,8,1"
    MSComm1.Handshaking = comXOnXoff
    MSComm1.InputLen = 3
    MSComm1.RThreshold = 3 'CommEvent-Eigenschaft nach 3 Zeichen auf
comEvReceive
    MSComm1.SThreshold = 1 'CommEvent-Eigenschaft bei weniger als 1
Zeichen auf comEvSend
    'MSComm1.InputMode = comInputModeBinary ' Binery für Prozessor
    MSComm1.PortOpen = True
    If Err Then
        MsgBox Error$, 48
        dfg = 1
        Exit Sub
    Else
        Print #5, "Serielle Verbindung über Com" & ComPort + 1 & " " &
MSComm1.Settings
        Print #5, ""
        dfg = 0
    End If
    OutString = ""
    ZumMotor
    frmSteuerung.Show
    Shapel.Visible = True
    cmdStartReset.Caption = "Abbrechen"
    cmdStartReset.Visible = True
    lblStartReset.Visible = True
    Do Until ((InString = "Hey") Or (cmdStartReset.Visible = False))
        DoEvents
        OutString = ""
        ZumMotor
    Loop
    If cmdStartReset.Visible = True Then
        cmdStartReset.Caption = "OK"
        lblStartResetOK.Visible = True
    End If
    If dfg = 1 Then
        lblConnect.Caption = "Steuerplatine nicht verbunden (Com" & ComPort +
1 & "):"
        imgConnect.Visible = False
        imgNotConnect.Visible = True
    Else
        lblConnect.Caption = "Steuerplatine verbunden (Com" & ComPort + 1 &
"):"
```

```
    imgConnect.Visible = True
    imgNotConnect.Visible = False
End If
TSt = Timer
Do Until ((Timer > TSt + 3) Or (cmdStartReset.Visible = False))
    DoEvents
Loop
Shapel.Visible = False
cmdStartReset.Visible = False
lblStartResetOK.Visible = False
lblStartReset.Visible = False
End Sub

Private Sub mnuAbbrechen_Click()
    cmdAbbrechen_Click
End Sub

Private Sub mnuBeenden_Click()
    cmdEnde_Click
End Sub

Private Sub mnuDatenfreigabe_Click()
    cmdDatenfreigabe_Click
End Sub

Private Sub mnuDrucken_Click()
    MsgBox "Hier wird später ein Protokoll gedruckt!"
End Sub

Private Sub mnuDruckerAnschluß_Click()
    If Passwort_Eingabe("Bitte geben Sie das" & Chr(13) & "Passwort ein!",
"Passwort") Then
        frmLPTPort.Show vbModal
    Else
        MsgBox "Falsches Passwort!!"
    End If
End Sub

Private Sub mnuNeu_Click()
    Abbr_Meldung = False
    If (flgGeändert) Then
        If (MsgBox("Wollen Sie die Daten speichern?", vbYesNo + vbQuestion,
"Achtung!") = 6) Then
            mnuSpeichern_Click
        End If
    End If
    Maske_INIT
End Sub

Private Sub mnuNeuesPaar_Click()
    For x = 0 To 7
        If WertEreignis(x) = 0 Then
            paarIdx = x
            Exit For
        End If
    Next x
    frmEingabe.Show vbModal
End Sub

Private Sub mnuöffnen_Click()
    CMDialog1.Filter = "*.dat"
    CMDialog1.filename = "*.dat"
    CMDialog1.ShowOpen
End Sub
```



```
DateiName = CMDialog1.filename
For x = 0 To 7
    anzNr(x).BackColor = &H80000000
    anzDrehzahl(x).BackColor = &H80000000
    anzZeit(x).BackColor = &H80000000
    anzEreignis(x).BackColor = &H80000000
    anzNr(x).ForeColor = QBColor(0)
    anzDrehzahl(x).ForeColor = QBColor(0)
    anzZeit(x).ForeColor = QBColor(0)
    anzEreignis(x).ForeColor = QBColor(0)
Next x
'If CMDialog1.CancelError Then
On Error GoTo OpenError
Open DateiName For Input As #2
Do While Not EOF(2)
    Input #2, Beschreibung, paarSTK
    txtBeschreibung.text = Beschreibung
    For x = 0 To 7
        Input #2, WertNr(x), WertDrehzahl(x), WertZeit(x),
WertEreignis(x)
        anzNr(x) = WertNr(x) + 1 & " "
        anzDrehzahl(x) = Format(WertDrehzahl(x), "##,##0") & " U/min. "
        anzZeit(x) = WertZeit(x) & " Sekunden "
        anzEreignis(x) = frmEingabe.ComboEreignis.List(WertEreignis(x))
    Next x
    Input #2, KD_nr, KD_adr1, KD_adr2, KD_str, KD_ort, KD_prNr, PR_dat
Loop
Close #2
flgGeändert = False
AnzeigeAktualisieren
TitelNeu (0)
'End If
frmSteuerung.mnuSpeichern.Enabled = True
frmSteuerung.mnuDrucken.Enabled = True
frmSteuerung.mnuDatenfreigabe.Enabled = True
frmSteuerung.cmdDatenfreigabe.Enabled = True
Unload frmEingabe
cmdDatenfreigabe.SetFocus
OpenError:
End Sub

Private Sub mnuOptionen_Click()
    If Passwort_Eingabe("Bitte geben Sie das" & Chr(13) & "Passwort ein!",
"Passwort") Then
        frmOptionen.Show vbModal
    Else
        MsgBox "Falsches Passwort!!"
    End If
End Sub

Private Sub mnuPaar_Click(Index As Integer)
    paarIdx = Index
    frmEingabe.Show vbModal
End Sub

Private Sub mnuProgrammhilfe_Click()
    MsgBox "Hier entsteht noch eine Hilfedatei!"
    'CMDialog1.HelpFile = "Steuerung.HLP"
    'CMDialog1.HelpCommand = cdlHelpContents
    'CMDialog1.ShowHelp ' Themenverzeichnis der Hilfedatei zeigen!
End Sub

Private Sub mnuSeriell_Click()
```

```
If Passwort_Eingabe("Bitte geben Sie das" & Chr(13) & "Passwort ein!",
"Passwort") Then
    frmComPort.Show vbModal
Else
    MsgBox "Falsches Passwort!!"
End If
End Sub

Private Sub mnuSpeichern_Click()
    CMDialog1.Filter = "*.dat"
    CMDialog1.ShowSave
    DateiName = CMDialog1.filename
    On Error GoTo SpeichernError
    If (UCASE(Right(DateiName, 4)) <> ".DAT") Then
        DateiName = DateiName & ".dat"
    End If
    Open DateiName For Output As #1
    Write #1, txtBeschreibung.text, paarSTK
    For x = 0 To 7
        Write #1, WertNr(x), WertDrehzahl(x), WertZeit(x), WertEreignis(x)
    Next x
    Write #1, KD_nr, KD_adr1, KD_adr2, KD_str, KD_ort, KD_prNr, PR_dat
    Close #1
    TitelNeu (0)
    mnuSpeichern.Enabled = False
    flgGeändert = False
SpeichernError:
End Sub

Private Sub mnuStarten_Click()
    cmdStart_Click
End Sub

Private Sub mnuüber_Click()
    frmAbout.Show vbModal
End Sub

Private Sub txtBeschreibung_Change()
    Beschreibung = txtBeschreibung.text
    flgGeändert = True
    TitelNeu (1)
End Sub

Private Sub txtBeschreibung_DblClick()
    frmProjektDaten.Show vbModal
End Sub
```

FrmSteuerung (Fensterdefinition)

```
VERSION 5.00
Object = "{6B7E6392-850A-101B-AFC0-4210102A8DA7}#1.2#0"; "COMCTL32.OCX"
Object = "{F9043C88-F6F2-101A-A3C9-08002B2F49FB}#1.1#0"; "COMDLG32.OCX"
Object = "{648A5603-2C6E-101B-82B6-000000000014}#1.1#0"; "MSCOMM32.OCX"
Begin VB.Form frmSteuerung
    BackColor      = &H80000000&
    BorderStyle    = 1 'Fest Einfach
    Caption        = "Steuerung"
    ClientHeight   = 5610
    ClientLeft     = 150
    ClientTop      = 435
    ClientWidth    = 8010
```

```
ControlBox      = 0  'False
LinkTopic      = "Form1"
MaxButton      = 0  'False
MinButton      = 0  'False
ScaleHeight    = 5610
ScaleWidth     = 8010
StartPosition   = 2  'Bildschirmmitte
WhatsThisButton = -1 'True
WhatsThisHelp  = -1 'True
Begin VB.CommandButton cmdStartReset
  Caption       = "Abbrechen"
  BeginProperty Font
    Name        = "MS Sans Serif"
    Size        = 12
    Charset     = 0
    Weight      = 400
    Underline   = 0  'False
    Italic      = 0  'False
    Strikethrough = 0  'False
  EndProperty
  Height       = 495
  Left        = 2880
  TabIndex    = 46
  Top        = 2280
  Visible     = 0  'False
  Width      = 2415
End
Begin ComctlLib.ProgressBar ProgressBar1
  Height     = 200
  Left      = 360
  TabIndex  = 29
  Top      = 3360
  Visible   = 0  'False
  Width    = 7300
  _ExtentX = 12885
  _ExtentY = 344
  _Version  = 327682
  Appearance = 1
End
Begin VB.CommandButton cmdDatenfreigabe
  Caption       = "Datenfreigabe"
  BeginProperty Font
    Name        = "MS Sans Serif"
    Size        = 9.75
    Charset     = 0
    Weight      = 700
    Underline   = 0  'False
    Italic      = 0  'False
    Strikethrough = 0  'False
  EndProperty
  Height       = 495
  Left        = 360
  TabIndex    = 44
  ToolTipText  = "Prozessdaten freigeben"
  Top        = 4320
  Width      = 1815
End
Begin VB.CommandButton cmdAbbrechen
  Caption       = "Abbrechen"
  BeginProperty Font
    Name        = "MS Sans Serif"
    Size        = 9.75
    Charset     = 0
```

```
        Weight          = 700
        Underline       = 0   'False
        Italic          = 0   'False
        Strikethrough   = 0   'False
    EndProperty
    Height              = 495
    Left                = 360
    TabIndex            = 28
    ToolTipText         = "Prozess abbrechen"
    Top                 = 4320
    Width               = 1815
End
Begin VB.CommandButton cmdStart
    Caption              = "Start"
    BeginProperty Font
        Name              = "MS Sans Serif"
        Size              = 9.75
        Charset           = 0
        Weight            = 700
        Underline         = 0   'False
        Italic            = 0   'False
        Strikethrough     = 0   'False
    EndProperty
    Height              = 495
    Left                = 360
    MouseIcon           = (Symbol)
    TabIndex            = 25
    ToolTipText         = "Prozess starten"
    Top                 = 4320
    Width               = 1815
End
Begin VB.CommandButton cmdEnde
    Caption              = "Beenden"
    BeginProperty Font
        Name              = "MS Sans Serif"
        Size              = 9.75
        Charset           = 0
        Weight            = 700
        Underline         = 0   'False
        Italic            = 0   'False
        Strikethrough     = 0   'False
    EndProperty
    Height              = 495
    Left                = 6390
    TabIndex            = 26
    ToolTipText         = "Programm beenden"
    Top                 = 4320
    Width               = 1215
End
Begin MSCommLib.MSComm MSComm1
    Left                = 2040
    Top                 = 4340
    _ExtentX            = 1005
    _ExtentY            = 1005
    _Version            = 327681
    DTREnable           = -1   'True
End
Begin MSComDlg.CommonDialog CMDialog1
    Left                = 2700
    Top                 = 4340
    _ExtentX            = 847
    _ExtentY            = 847
    _Version            = 327681
```

```
End
Begin VB.TextBox txtBeschreibung
    Height          = 615
    Left           = 360
    MultiLine      = -1 'True
    ScrollBars     = 2 'Vertikal
    TabIndex       = 39
    ToolTipText    = "Mit Doppel-Klick können Sie die Projekt-Daten
bearbeite
    Top           = 4920
    Width         = 7300
End
Begin VB.Label HardwareStatus1
    BackColor      = &H00C0FFC0&
    Caption        = "Motorstrom"
    Height         = 240
    Left          = 5640
    TabIndex      = 48
    Top           = 120
    Width         = 2250
End
Begin VB.Label lblStartResetOK
    Alignment      = 2 'Zentriert
    BackColor     = &H0080FF80&
    Caption       = "Reset wurde erfolgreich empfangen!"
    BeginProperty Font
        Name          = "MS Sans Serif"
        Size         = 13.5
        Charset      = 0
        Weight       = 400
        Underline    = 0 'False
        Italic       = 0 'False
        Strikethrough = 0 'False
    EndProperty
    ForeColor     = &H000000FF&
    Height        = 495
    Left         = 720
    TabIndex     = 47
    Top          = 1680
    Visible      = 0 'False
    Width        = 6615
End
Begin VB.Label lblStartReset
    Alignment      = 2 'Zentriert
    BackColor     = &H0080FF80&
    Caption       = "Bitte drücken Sie die Resettaste an der
Steuerplatine,
    BeginProperty Font
        Name          = "MS Sans Serif"
        Size         = 13.5
        Charset      = 0
        Weight       = 400
        Underline    = 0 'False
        Italic       = 0 'False
        Strikethrough = 0 'False
    EndProperty
    Height        = 855
    Left         = 720
    TabIndex     = 45
    Top          = 720
    Width        = 6615
End
Begin VB.Shape Shapel
```

```
        BackColor      = &H0080FF80&
        BackStyle     = 1  'Undurchsichtig
        Height        = 2415
        Left           = 240
        Top            = 480
        Visible        = 0  'False
        Width          = 7530
    End
Begin VB.Label HardwareStatus
    BackColor      = &H00C0FFC0&
    Caption         = "Trafotemperaturen"
    Height         = 240
    Left           = 120
    TabIndex       = 43
    Top            = 120
    Width          = 7770
End
Begin VB.Label test2
    Caption         = "test"
    Height         = 195
    Left           = 720
    TabIndex       = 42
    Top            = 3740
    Visible        = 0  'False
    Width          = 6000
End
Begin VB.Label test1
    Caption         = "test"
    Height         = 195
    Left           = 720
    TabIndex       = 41
    Top            = 3910
    Visible        = 0  'False
    Width          = 6000
End
Begin VB.Label lblConnect
    Alignment       = 1  'Rechts
    Caption         = "HardwareConected:"
    Height         = 255
    Left           = 2280
    TabIndex       = 40
    Top            = 4560
    Width          = 3375
End
Begin VB.Image imgNotConnect
    Height         = 180
    Left           = 5760
    Picture        = (Symbol)
    Stretch        = -1  'True
    ToolTipText    = "Toggles Port"
    Top            = 4560
    Width          = 180
End
Begin VB.Image imgConnect
    Height         = 180
    Left           = 5760
    Picture        = (Symbol)
    Stretch        = -1  'True
    ToolTipText    = "Toggles Port"
    Top            = 4560
    Visible        = 0  'False
    Width          = 180
End
```

```
Begin VB.Image imgNotDrehzahl
    Height      = 180
    Index       = 7
    Left        = 5800
    Picture     = (Symbol)
    Stretch     = -1 'True
    ToolTipText = "Toggles Port"
    Top        = 2550
    Visible     = 0 'False
    Width      = 180
End
Begin VB.Image imgNotDrehzahl
    Height      = 180
    Index       = 6
    Left        = 5800
    Picture     = (Symbol)
    Stretch     = -1 'True
    ToolTipText = "Toggles Port"
    Top        = 2295
    Visible     = 0 'False
    Width      = 180
End
Begin VB.Image imgNotDrehzahl
    Height      = 180
    Index       = 5
    Left        = 5800
    Picture     = (Symbol)
    Stretch     = -1 'True
    ToolTipText = "Toggles Port"
    Top        = 2055
    Visible     = 0 'False
    Width      = 180
End
Begin VB.Image imgNotDrehzahl
    Height      = 180
    Index       = 4
    Left        = 5800
    Picture     = (Symbol)
    Stretch     = -1 'True
    ToolTipText = "Toggles Port"
    Top        = 1800
    Visible     = 0 'False
    Width      = 180
End
Begin VB.Image imgNotDrehzahl
    Height      = 180
    Index       = 3
    Left        = 5800
    Picture     = (Symbol)
    Stretch     = -1 'True
    ToolTipText = "Toggles Port"
    Top        = 1545
    Visible     = 0 'False
    Width      = 180
End
Begin VB.Image imgNotDrehzahl
    Height      = 180
    Index       = 2
    Left        = 5800
    Picture     = (Symbol)
    Stretch     = -1 'True
    ToolTipText = "Toggles Port"
    Top        = 1305
```

```
Visible      = 0  'False
Width       = 180
End
Begin VB.Image imgNotDrehzahl
Height      = 180
Index       = 1
Left        = 5800
Picture     = (Symbol)
Stretch     = -1  'True
ToolTipText = "Toggles Port"
Top         = 1050
Visible     = 0  'False
Width       = 180
End
Begin VB.Image imgNotDrehzahl
Height      = 180
Index       = 0
Left        = 5800
Picture     = (Symbol)
Stretch     = -1  'True
ToolTipText = "Toggles Port"
Top         = 795
Visible     = 0  'False
Width       = 180
End
Begin VB.Image imgDrehzahl
Height      = 180
Index       = 7
Left        = 5800
Picture     = (Symbol)
Stretch     = -1  'True
ToolTipText = "Toggles Port"
Top         = 2550
Visible     = 0  'False
Width       = 180
End
Begin VB.Image imgDrehzahl
Height      = 180
Index       = 6
Left        = 5800
Picture     = (Symbol)
Stretch     = -1  'True
ToolTipText = "Toggles Port"
Top         = 2300
Visible     = 0  'False
Width       = 180
End
Begin VB.Image imgDrehzahl
Height      = 180
Index       = 5
Left        = 5800
Picture     = (Symbol)
Stretch     = -1  'True
ToolTipText = "Toggles Port"
Top         = 2050
Visible     = 0  'False
Width       = 180
End
Begin VB.Image imgDrehzahl
Height      = 180
Index       = 4
Left        = 5800
Picture     = (Symbol)
```



```
Stretch          = -1 'True
ToolTipText     = "Toggles Port"
Top             = 1800
Visible        = 0  'False
Width          = 180
End
Begin VB.Image imgDrehzahl
Height         = 180
Index         = 3
Left          = 5800
Picture       = (Symbol)
Stretch       = -1 'True
ToolTipText   = "Toggles Port"
Top           = 1550
Visible      = 0  'False
Width        = 180
End
Begin VB.Image imgDrehzahl
Height         = 180
Index         = 2
Left          = 5800
Picture       = (Symbol)
Stretch       = -1 'True
ToolTipText   = "Toggles Port"
Top           = 1300
Visible      = 0  'False
Width        = 180
End
Begin VB.Image imgDrehzahl
Height         = 180
Index         = 1
Left          = 5800
Picture       = (Symbol)
Stretch       = -1 'True
ToolTipText   = "Toggles Port"
Top           = 1050
Visible      = 0  'False
Width        = 180
End
Begin VB.Image imgDrehzahl
Height         = 180
Index         = 0
Left          = 5800
Picture       = (Symbol)
Stretch       = -1 'True
ToolTipText   = "Toggles Port"
Top           = 800
Visible      = 0  'False
Width        = 180
End
Begin VB.Label anzEreignis
Alignment     = 1  'Rechts
Caption      = "Ereignis"
Height       = 200
Index       = 7
Left        = 4400
TabIndex   = 38
Top        = 2550
Width      = 800
End
Begin VB.Label anzEreignis
Alignment     = 1  'Rechts
Caption      = "Ereignis"
```

```
    Height      = 200
    Index       = 6
    Left        = 4400
    TabIndex    = 37
    Top         = 2300
    Width       = 800
End
Begin VB.Label anzEreignis
    Alignment    = 1 'Rechts
    Caption      = "Ereignis"
    Height       = 200
    Index        = 5
    Left         = 4400
    TabIndex     = 36
    Top          = 2050
    Width        = 800
End
Begin VB.Label anzEreignis
    Alignment    = 1 'Rechts
    Caption      = "Ereignis"
    Height       = 200
    Index        = 4
    Left         = 4400
    TabIndex     = 35
    Top          = 1800
    Width        = 800
End
Begin VB.Label anzEreignis
    Alignment    = 1 'Rechts
    Caption      = "Ereignis"
    Height       = 200
    Index        = 3
    Left         = 4400
    TabIndex     = 34
    Top          = 1550
    Width        = 800
End
Begin VB.Label anzEreignis
    Alignment    = 1 'Rechts
    Caption      = "Ereignis"
    Height       = 200
    Index        = 2
    Left         = 4400
    TabIndex     = 33
    Top          = 1300
    Width        = 800
End
Begin VB.Label anzEreignis
    Alignment    = 1 'Rechts
    Caption      = "Ereignis"
    Height       = 200
    Index        = 1
    Left         = 4400
    TabIndex     = 32
    Top          = 1050
    Width        = 800
End
Begin VB.Label anzEreignis
    Alignment    = 1 'Rechts
    Caption      = "Ereignis"
    Height       = 200
    Index        = 0
    Left         = 4400
```

```
    TabIndex      = 31
    Top           = 800
    Width        = 800
End
Begin VB.Label lblFortschritt
    Caption       = "Fortschritt:"
    Height       = 200
    Left        = 360
    TabIndex     = 30
    Top         = 3100
    Visible     = 0 'False
    Width      = 3000
End
Begin VB.Label test
    Caption       = "test"
    Height       = 195
    Left        = 720
    TabIndex     = 27
    Top         = 4080
    Visible     = 0 'False
    Width      = 6000
End
Begin VB.Label anzZeit
    Alignment    = 1 'Rechts
    Caption     = "Zeit"
    Height     = 200
    Index      = 0
    Left      = 2500
    TabIndex  = 24
    Top       = 800
    Width    = 1200
End
Begin VB.Label anzZeit
    Alignment    = 1 'Rechts
    Caption     = "Zeit"
    Height     = 200
    Index      = 1
    Left      = 2500
    TabIndex  = 23
    Top       = 1050
    Width    = 1200
End
Begin VB.Label anzZeit
    Alignment    = 1 'Rechts
    Caption     = "Zeit"
    Height     = 200
    Index      = 2
    Left      = 2500
    TabIndex  = 22
    Top       = 1300
    Width    = 1200
End
Begin VB.Label anzZeit
    Alignment    = 1 'Rechts
    Caption     = "Zeit"
    Height     = 200
    Index      = 3
    Left      = 2500
    TabIndex  = 21
    Top       = 1550
    Width    = 1200
End
Begin VB.Label anzZeit
```

```
Alignment      = 1  'Rechts
Caption        =  "Zeit"
Height        =  200
Index         =  4
Left          =  2500
TabIndex     =  20
Top          =  1800
Width        =  1200
End
Begin VB.Label anzZeit
Alignment      = 1  'Rechts
Caption        =  "Zeit"
Height        =  200
Index         =  5
Left          =  2500
TabIndex     =  19
Top          =  2050
Width        =  1200
End
Begin VB.Label anzZeit
Alignment      = 1  'Rechts
Caption        =  "Zeit"
Height        =  200
Index         =  6
Left          =  2500
TabIndex     =  18
Top          =  2300
Width        =  1200
End
Begin VB.Label anzZeit
Alignment      = 1  'Rechts
Caption        =  "Zeit"
Height        =  200
Index         =  7
Left          =  2500
TabIndex     =  17
Top          =  2550
Width        =  1200
End
Begin VB.Label anzDrehzahl
Alignment      = 1  'Rechts
Caption        =  "Drehzahl"
Height        =  200
Index         =  0
Left          =  1000
TabIndex     =  16
Top          =  800
Width        =  1100
End
Begin VB.Label anzDrehzahl
Alignment      = 1  'Rechts
Caption        =  "Drehzahl"
Height        =  200
Index         =  1
Left          =  1000
TabIndex     =  15
Top          =  1050
Width        =  1100
End
Begin VB.Label anzDrehzahl
Alignment      = 1  'Rechts
Caption        =  "Drehzahl"
Height        =  200
```

```

    Index          = 2
    Left           = 1000
    TabIndex       = 14
    Top            = 1300
    Width          = 1100
End
Begin VB.Label anzDrehzahl
    Alignment      = 1 'Rechts
    Caption        = "Drehzahl"
    Height         = 200
    Index          = 3
    Left           = 1000
    TabIndex       = 13
    Top            = 1550
    Width          = 1100
End
Begin VB.Label anzDrehzahl
    Alignment      = 1 'Rechts
    Caption        = "Drehzahl"
    Height         = 200
    Index          = 4
    Left           = 1000
    TabIndex       = 12
    Top            = 1800
    Width          = 1100
End
Begin VB.Label anzDrehzahl
    Alignment      = 1 'Rechts
    Caption        = "Drehzahl"
    Height         = 200
    Index          = 5
    Left           = 1000
    TabIndex       = 11
    Top            = 2050
    Width          = 1100
End
Begin VB.Label anzDrehzahl
    Alignment      = 1 'Rechts
    Caption        = "Drehzahl"
    Height         = 200
    Index          = 6
    Left           = 1000
    TabIndex       = 10
    Top            = 2300
    Width          = 1100
End
Begin VB.Label anzDrehzahl
    Alignment      = 1 'Rechts
    Caption        = "Drehzahl"
    Height         = 200
    Index          = 7
    Left           = 1000
    TabIndex       = 9
    Top            = 2550
    Width          = 1100
End
Begin VB.Label anzNr
    Alignment      = 1 'Rechts
    Caption        = "Nr"
    Height         = 200
    Index          = 7
    Left           = 400
    TabIndex       = 8

```

```
        Top           = 2550
        Width         = 360
    End
    Begin VB.Label anzNr
        Alignment      = 1 'Rechts
        Caption        = "Nr"
        Height         = 200
        Index          = 6
        Left           = 400
        TabIndex       = 7
        Top            = 2300
        Width          = 360
    End
    Begin VB.Label anzNr
        Alignment      = 1 'Rechts
        Caption        = "Nr"
        Height         = 200
        Index          = 5
        Left           = 400
        TabIndex       = 6
        Top            = 2050
        Width          = 360
    End
    End
    Begin VB.Label anzNr
        Alignment      = 1 'Rechts
        Caption        = "Nr"
        Height         = 200
        Index          = 4
        Left           = 400
        TabIndex       = 5
        Top            = 1800
        Width          = 360
    End
    End
    Begin VB.Label anzNr
        Alignment      = 1 'Rechts
        Caption        = "Nr"
        Height         = 200
        Index          = 3
        Left           = 400
        TabIndex       = 4
        Top            = 1550
        Width          = 360
    End
    End
    Begin VB.Label anzNr
        Alignment      = 1 'Rechts
        Caption        = "Nr"
        Height         = 200
        Index          = 2
        Left           = 400
        TabIndex       = 3
        Top            = 1300
        Width          = 360
    End
    End
    Begin VB.Label anzNr
        Alignment      = 1 'Rechts
        Caption        = "Nr"
        Height         = 200
        Index          = 1
        Left           = 400
        TabIndex       = 2
        Top            = 1050
        Width          = 360
    End
    End
```

```
Begin VB.Label anzNr
    Alignment      = 1   'Rechts
    Caption        = "Nr"
    Height        = 200
    Index         = 0
    Left          = 400
    TabIndex      = 1
    Top           = 800
    Width         = 360
End
Begin VB.Label WerteKopf
    Caption        = "WerteKopf"
    BeginProperty Font
        Name        = "MS Sans Serif"
        Size        = 8.25
        Charset     = 0
        Weight      = 700
        Underline   = 0   'False
        Italic      = 0   'False
        Strikethrough = 0   'False
    EndProperty
    Height        = 255
    Left         = 460
    TabIndex     = 0
    Top          = 480
    Width        = 7000
End
Begin VB.Menu mnuDatei
    Caption        = "&Datei"
    NegotiatePosition= 1   'Links
    Begin VB.Menu mnuNeu
        Caption    = "&Neu"
    End
    Begin VB.Menu mnuöffnen
        Caption    = "ö&ffnen"
    End
    Begin VB.Menu mnuSpeichern
        Caption    = "Speiche&rnr"
        Shortcut   = ^R
    End
    Begin VB.Menu mnuDrucken
        Caption    = "&Drucken"
        Shortcut   = ^P
    End
    Begin VB.Menu mnuSeparator
        Caption    = "-"
    End
    Begin VB.Menu mnuBeenden
        Caption    = "Be&enden"
        Shortcut   = ^E
    End
End
Begin VB.Menu mnuDaten
    Caption        = "D&aten"
    Begin VB.Menu mnuNeuesPaar
        Caption    = "&Neus Paar"
        Shortcut   = ^N
    End
    Begin VB.Menu mnuPaar
        Caption    = "Paar &l"
        Index      = 0
        Shortcut   = {F1}
    End
End
```

```
Begin VB.Menu mnuPaar
  Caption      = "Paar &2"
  Index        = 1
  Shortcut     = {F2}
End
Begin VB.Menu mnuPaar
  Caption      = "Paar &3"
  Index        = 2
  Shortcut     = {F3}
End
Begin VB.Menu mnuPaar
  Caption      = "Paar &4"
  Index        = 3
  Shortcut     = {F4}
End
Begin VB.Menu mnuPaar
  Caption      = "Paar &5"
  Index        = 4
  Shortcut     = {F5}
End
Begin VB.Menu mnuPaar
  Caption      = "Paar &6"
  Index        = 5
  Shortcut     = {F6}
End
Begin VB.Menu mnuPaar
  Caption      = "Paar &7"
  Index        = 6
  Shortcut     = {F7}
End
Begin VB.Menu mnuPaar
  Caption      = "Paar &8"
  Index        = 7
  Shortcut     = {F8}
End
End
Begin VB.Menu mnuProzess
  Caption      = "&Prozess"
  Begin VB.Menu mnuDatenfreigabe
    Caption     = "&Datenfreigabe"
    Shortcut    = ^D
  End
  Begin VB.Menu mnuStarten
    Caption     = "&Starten"
    Shortcut    = ^S
  End
  Begin VB.Menu mnuAbbrechen
    Caption     = "&Abbrechen"
    Shortcut    = ^A
  End
End
Begin VB.Menu mnuKonfiguration
  Caption      = "&Konfiguration"
  Begin VB.Menu mnuSeriell
    Caption     = "&Serielle Schnittstelle"
  End
  Begin VB.Menu mnuDruckerAnschluß
    Caption     = "&Drucker Anschluß"
  End
  Begin VB.Menu mnuOptionen
    Caption     = "&Optionen"
  End
End
End
```



```

Begin VB.Menu mnuHilfe
    Caption          = "&?"
    NegotiatePosition= 3 'Rechts
Begin VB.Menu mnuProgrammhilfe
    Caption          = "Hilfe zum Programm"
End
Begin VB.Menu mnuüber
    Caption          = "über Steuerung!"
End
End
End
End

```

SteuerungModule1 (Programmcode)

```

Public WertZeit(0 To 7), WertDrehzahl(0 To 7), WertNr(0 To 7),
WertEreignis(0 To 7) As Integer
Public DrehzahlKorrektur() As Single
Public DrehzahlBereich(), AnzahlDrehzahlen, MaxStromMotor As Integer
Public paarIdx, paarSTK, serieller_Schleifenzähler, MaxChar, MaxDrehzahl,
MinDrehzahl As Integer
Public RELpause, TSfaktor, TMfaktor, Sfaktor As Single
Public flgAbbrechen, flgGeändert, flgGestartet As Boolean
Public Beschreibung, Dateiname, Passwort As String
Public PW1 As String
Public Reset_OK As Boolean
Public ComPort, lptPort As Byte
Public KD_nr, KD_adr1, KD_adr2, KD_str, KD_ort, KD_prNr, PR_dat As String
Public InString, OutString As String
Dim Ende As String

Function PW_versch(p1 As String, IO As Boolean) As String
Dim I As Integer
Dim s As String
    s = ""
    If IO Then 'VERschlüsseln
        For I = 1 To Len(p1)
            s = s + Chr(Asc(Mid(p1, I, 1)) + I)
            PW_versch = s
        Next I
    Else 'ENTschlüsseln
        For I = 1 To Len(p1)
            s = s + Chr(Asc(Mid(p1, I, 1)) - I)
            PW_versch = s
        Next I
    End If
End Function

Sub CFG_Speichern()
Dim I As Integer
Open "Steuerung.cfg" For Output As #1
Write #1, ComPort
Write #1, lptPort
Write #1, serieller_Schleifenzähler
Write #1, MaxChar
Write #1, MaxDrehzahl
Write #1, MinDrehzahl
Write #1, PW_versch(Passwort, True)
Write #1, RELpause
Write #1, TMfaktor
Write #1, TSfaktor
Write #1, Sfaktor

```

```
Write #1, AnzahlDrehzahlen
Write #1, MaxStromMotor
For I = 1 To AnzahlDrehzahlen
    Write #1, DrehzahlBereich(I), DrehzahlKorrektur(I)
Next I
Write #1, Ende
Close #1
End Sub

Sub Init()
Dim I As Integer
ComPort = 1 'Com2
lptPort = 0 'LPT1
serieller_Schleifenzähler = 20
MaxChar = 245
MaxDrehzahl = 13000
MinDrehzahl = 1000
Passwort = "diplom98"
RELpause = 2 'in sekunden
TMfaktor = 0.5
TSfaktor = 0.5
Sfaktor = 20
AnzahlDrehzahlen = 50
MaxStromMotor = 2500
ReDim DrehzahlKorrektur(AnzahlDrehzahlen)
ReDim DrehzahlBereich(AnzahlDrehzahlen)
For I = 1 To AnzahlDrehzahlen
    DrehzahlKorrektur(I) = 1
    DrehzahlBereich(I) = I * 260
Next I
Ende = "Ende"
End Sub

Sub CFG_Lesen()
Dim I, AD1 As Integer
If Dir("Steuerung.cfg") <> "Steuerung.cfg" Then
    Init
    CFG_Speichern
Else
    On Error GoTo errCFG_Datei
    Open "Steuerung.cfg" For Input As #1
    Input #1, ComPort
    Input #1, lptPort
    Input #1, serieller_Schleifenzähler
    Input #1, MaxChar
    Input #1, MaxDrehzahl
    Input #1, MinDrehzahl
    Input #1, Passwort
    Input #1, RELpause
    Input #1, TMfaktor
    Input #1, TSfaktor
    Input #1, Sfaktor
    Input #1, AD1
    AnzahlDrehzahlen = AD1
    ReDim DrehzahlKorrektur(AnzahlDrehzahlen)
    ReDim DrehzahlBereich(AnzahlDrehzahlen)
    Input #1, MaxStromMotor
    For I = 1 To AnzahlDrehzahlen
        Input #1, DrehzahlBereich(I), DrehzahlKorrektur(I)
    Next I
    Input #1, Ende
    Close #1
    Passwort = PW_versch(Passwort, False)
```

```
End If
Exit Sub
errCFG_Datei:
Close #1
MsgBox "Fehler beim Lesen der CFG-Datei!" & Chr(13) & "Eine neue
Standart-CFG-Datei wurde erstellt!"
Init
CFG_Speichern
End Sub

Sub AnzeigeAktualisieren()
Dim x, xx As Integer
' Werte sortieren (Leerzeilen entfernen)
For xx = 0 To 7
For x = 1 To 7
If (WertEreignis(x - 1) = 0) And (WertEreignis(x) > 0) Then
WertDrehzahl(x - 1) = WertDrehzahl(x)
WertZeit(x - 1) = WertZeit(x)
WertEreignis(x - 1) = WertEreignis(x)
WertDrehzahl(x) = 0
WertZeit(x) = 0
WertEreignis(x) = 0
frmSteuerung.anzEreignis(x - 1) = frmSteuerung.anzEreignis(x)
frmSteuerung.anzDrehzahl(x - 1) = frmSteuerung.anzDrehzahl(x)
frmSteuerung.anzZeit(x - 1) = frmSteuerung.anzZeit(x)
flgGeändert = True
If (Right(frmSteuerung.Caption, 1) <> "*") Then
frmSteuerung.Caption = frmSteuerung.Caption & "*"
End If
End If
Next x
Next xx
' Anzeige aktualisieren
For x = 0 To 7
frmSteuerung.anzNr(x).Visible = True
frmSteuerung.anzNr(x).Caption = "F" & (x + 1)
If WertEreignis(x) > 0 Then
frmSteuerung.anzDrehzahl(x).Visible = True
frmSteuerung.anzZeit(x).Visible = True
frmSteuerung.anzEreignis(x).Visible = True
frmSteuerung.imgDrehzahl(x).Visible = False
frmSteuerung.imgNotDrehzahl(x).Visible = True
Else
frmSteuerung.anzDrehzahl(x).Visible = False
frmSteuerung.anzZeit(x).Visible = False
frmSteuerung.anzEreignis(x).Visible = False
frmSteuerung.imgDrehzahl(x).Visible = False
frmSteuerung.imgNotDrehzahl(x).Visible = False
End If
Next x
End Sub

Function Passwort_Eingabe(str1, str2 As String) As Boolean
frmPasswort.Text1.text = ""
frmPasswort.Label1.Caption = str1
frmPasswort.Text1.PasswordChar = "*"
frmPasswort.Caption = str2
frmPasswort.Show vbModal
If PW1 = Passwort Then
Passwort_Eingabe = True
Else
Passwort_Eingabe = False
End If
```

End Function

9.2. Programmlisting der Regelsoftware

Texte, die zwischen den Zeichen `/*` und `*/` stehen, sind lediglich Kommentare und dienen dem Verständnis beim Lesen des Programmes.

```

/*
Programmname: REGELUNG.C

Funktion      : Steuert das Verhalten für einen Gleichstrommotor
                OK am 24.07.98

erstellt am  : 23.07.98
von          : HE + TJ
*/

#pragma small code debug

#include <REG515.H>
#include <STDIO.H>

/***** Ports - Definitionen *****/
sbit  PORT1_7  = P1^7; /* Reed-Relais fuer Pumpe 1=off 0=on */
sbit  PORT4_0  = P4^0; /* Status LED 1=off 0=on */
sbit  PORT4_1  = P4^1; /* Motorsignal an 555 1=on 0=off */
sbit  PORT4_2  = P4^2; /* Entwickler Ausgang 1=off 0=on */
sbit  PORT4_3  = P4^3; /* Spülung Ausgang 1=off 0=on */
sbit  PORT4_4  = P4^4; /* Schwenkarm Ausgang 1=off 0=on */
sbit  PORT5_1  = P5^1; /* Unterdruck Eingang 1=ok 0=Fehler
*/
sbit  PORT5_2  = P5^2; /* Entwickler Eingang 1=ok 0=Fehler
*/
sbit  PORT5_3  = P5^3; /* Spülung Eingang 1=ok 0=Fehler
*/
sbit  PORT5_4  = P5^4; /* Start Eingang 1=off 0=aktiv */
sbit  PORT5_5  = P5^5; /* Stop Eingang 1=off 0=aktiv */

/***** Definition der globalen Konstanten *****/
#define Keingas_Grenze 1 /* 1 Grenze muß definiert sein 1-255
*/
#define Vollgas_Grenze 250 /* 240 Grenze muß definiert sein 1-
255 */
#define SZ_seriell 10000 /* Schleifenzähler beim Empfang */

/***** Definition der globalen Variablen *****/
int iz = 0; /* Hifszähler für die ISR */
int ib = 0; /* Hifszähler für die ISR zum
Beschleunigen*/
int adhl = 16; /* Multiplikator fuer Reglerwerte */
int flgDrehzahl = 0; /* =1 wenn neue Drehzahl empfangen
*/
int flgAktiv = 0; /* =1 wenn ein Prozess aktiv ist */
int DZerreicht = 0; /* Drehzahl erreicht Flag */
int ErsteDrehzahl = 0; /* ErsteDrehzahl aktiv Flag */
short y = 0; /* Regler-Variabale y Funktion */
short w = 0; /* Regler-Variabale w Sollwert */
short x = 0; /* Regler-Variabale x Istwert */
int p = 1; /* Regler-Variabale p Verstärkung */
signed int ADkorrektur = 0; /* Korrekturfaktor fuer die AD-Wandlung des
Tachogenerators */
data unsigned char soll_gsw = 0; /* Sollgeschwindigkeit */

```

```

data unsigned char Hsoll_gsw = 0; /* Sollgeschwindigkeit zum
Beschleunigen*/
data unsigned char soll_alt = 0; /* alte Sollgeschwindigkeit */
data unsigned char hilf_ad = 0; /* Hilfsvariable für A/D-Wandler
Daten (P6.2) */
data unsigned char pwm = 0;

/***** Funktions-Prototypen *****/
void Timer_0_ISR(void);
void init(void);
void main(void);

/***** Initialisierung der Ports... *****/
void init(void)
{
    PORT1_7 = 1; /* Reed-Relais fuer Pumpe 1=off 0=on */
    PORT4_0 = 1; /* Status LED 1=off 0=on */
    PORT4_1 = 0; /* Motorsignal an 555 1=on 0=off */
    PORT4_2 = 1; /* Entwickler Ausgang 1=off 0=on */
    PORT4_3 = 1; /* Spülung Ausgang 1=off 0=on */
    PORT4_4 = 1; /* Schwenkarm Ausgang 1=off 0=on */
    PORT5_1 = 1; /* Unterdruck Eingang 1=ok 0=Fehler */
    PORT5_2 = 1; /* Entwickler Eingang 1=ok 0=Fehler */
    PORT5_3 = 1; /* Spülung Eingang 1=ok 0=Fehler */
    PORT5_4 = 1; /* Start Eingang 1=off 0=aktiv */
    PORT5_5 = 1; /* Stop Eingang 1=off 0=aktiv */

    /* Initialisierungen für Serielle Schnittstelle und der Timer */

    PCON = 0x80; /* Faktor (x+1) * Baudrate in Register PCON.7
*/
    SCON = 0x50; /* "50h" fuer Windows/DOS (Mode 1) */
    /* Timer 10 Timer 1=2 => Betriebsart 3 8bit Timer mit
AutoReload */
    TMOD = 0x21; /* 0=1 => Betriebsart 2 16-bit Timer */
    TCON = 0x00; /* alle TIMER stop und flags = 0 */
    TH1 = 0xFD; /* Reloadwert Timer 1 */
    TH0 = 0xf0; /* Startwert Timer 0 */
    TL0 = 0xd8; /* Startwert Timer 0 */

    /* Initialisierungen für Interrupts */

    IEN0 = 0x02; /* Generelles Sperren intTimer0 gesetzt */
    IEN1 = 0x00; /* Alle Interrupts sperren */
    IP0 = 0x00;
    IP1 = 0x00;

    /* Initialisierungen für den timer2 */

    T2CON = 0x71; /* Timer-Modus: Compare, interner Zähler,
Reload */
    CCEN = 0x80; /* Ausgang PORT 1.3 als Compare-Ausgang */

    CRCL = 0x00; /* RELOAD-Wert auf 0xFF00 setzen -> ca. 4KHz
*/
    CRCH = 0xFF;

    TL2 = 0x00; /* Timer2 auf hohen Startwert setzen
(0x0000) */
    TH2 = 0x00;

    CCL3 = 0x00;
    CCH3 = 0xFF;

```

```

}

/***** ISR - Interruptsevicieroutine Timer 0... *****/
void Timer_0_ISR(void) interrupt 1
{
    iz++;
    ib++;
    w = soll_gsw * adhl;          /* Sollwert */
    ADCON = 0x02;                /* Einzel-Wandlung, Analogkanal 3 (6.2
= Tachogenerator)*/
    DAPR = 0x00;                 /* Start der Wandlung (0V-5V)
*/
    while (BSY);                /* Warten auf Wandlungsende*/
    x = (ADDAT + ADkorrektur) * adhl; /* AD-Wert in ADDAT*/
    y = (w - x) * p;             /* P-Regler y=(w-x)*p */
    if ((y / adhl) > Vollgas_Grenze) { CCL3 = Vollgas_Grenze; }
    else if ((y / adhl) < Keingas_Grenze) { CCL3 = Keingas_Grenze; }
    else { CCL3 = y / adhl; }
    if (-2 < (w - x) < 2) {DZerreicht = 1;}
}

/***** benutzte Funktionen... *****/
unsigned char ist_temp_tmf(void) /* A/D-Wandlung der Temperatur des
Trafos Motor/Filter */
{
    ADCON = 0x00;                /* Einzel-Wandlung, Analogkanal 2 (6.0)*/
    DAPR = 0x62;                /* Start der Wandlung (3=0,9375V bis 5=1.5625V) */
                                /* (2=0,625V bis 6=1.875V) */
    while (BSY);                /* Warten auf Wandlungsende */
    return (ADDAT);
}

unsigned char ist_temp_str(void) /* A/D-Wandlung der Temperatur des Trafos
Steuerung/Relais */
{
    ADCON = 0x01;                /* Einzel-Wandlung, Analogkanal 2 (6.1)*/
    DAPR = 0x62;                /* Start der Wandlung (3=0,9375V bis 5=1.5625V) */
                                /* (2=0,625V bis 6=1.875V) */
    while (BSY);                /* Warten auf Wandlungsende */
    return (ADDAT);
}

unsigned char ist_strom(void) /* A/D-Wandlung der Stromaufnahme des
Motors */
{
    ADCON = 0x03;                /* Einzel-Wandlung, Analogkanal 1 (6.3)*/
    DAPR = 0x00;                /* Start der Wandlung (0V-5V) */
    while (BSY);                /* Warten auf Wandlungsende */
    return (ADDAT);
}

delay(unsigned int zeit) /* Verzoeigerung */
{
    for (;zeit>0;zeit--); /* warte angegebene Zeit */
}

sendStop(void) /* Bediener Stop */
{
    putchar('*');
    putchar('B');
    putchar('*');
}

```

```
sendStart(void)          /* Bediener Start */
{
    putchar('*');
    putchar('G');
    putchar('*');
}

empfangOK(unsigned char sc) /* Empfangsbestätigung */
{
    putchar('*');
    putchar(sc);
    putchar('*');
}

sendFehler(unsigned char sc) /* Fehlermeldung */
{
    putchar('#');
    putchar(sc);
    putchar('#');
}

/***** Hauptprogramm... *****/
void main(void)
{
    int flgStLED = 0;          /* =1 wenn Blinken aktiv */
    int zae = 0;              /* Hifszaeher für die Regelung */
    data unsigned char hilf_ser = 0; /* Hilfsvariable für Serielle Daten
*/
    data unsigned char hilf_er = 0; /* Hilfsvariable für Serielle Daten
Ereignis*/

    init();
    TR1 = 1;
    TI = 1;
    RI = 1;
    flgDrehzahl = 0;
    flgAktiv = 0;
    ErsteDrehzahl = 0;
    putchar('H');
    putchar('e');
    putchar('y');

    PORT4_0 = 0; /* LED an */
    delay (50000);
    PORT4_0 = 1; /* LED aus */
    delay (50000);
    PORT4_0 = 0; /* LED an */

    /** Timer 0 starten **/
    TR0 = 1;
    ET0 = 1;
    EAL = 1; /* Interrups frei geben */

    /* Programmstart */
    while (1) /* ... bis das der TOD uns scheidet */
    {
        /***** TEST Bereich *****/
        /*putchar(CCL3);*/
        /***** ...lx wenn die Drehzahl erreicht ist *****/
        if ((Hsoll_gsw == soll_gsw) && (DZerreicht) && (flgDrehzahl))
        {
            empfangOK(soll_gsw);
        }
    }
}
```



```

        flgDrehzahl = 0;
        DZerreicht = 0;
    }
    /***** serieller Eingang eines Zeichens *****/
    if (RI) {hilf_ser = _getkey(); RI = 0;}
    else {RI = 1; hilf_ser = 0x00;}
    /***** Status LED *****/
    if (iz > 6)
    {
        iz = 0;
        PORT4_0 = ~PORT4_0; /* Status LED */
    }
    /***** Prüfung der Eingang-PORTs *****/
    if (PORT5_2) /* Entwickler */
    { /* OK */
    }
    else
    {
        sendFehler('e');
    }
    if (PORT5_3) /* Spülung */
    { /* OK */
    }
    else
    {
        sendFehler('s');
    }
    if (PORT5_1) /* Unterdruck/Vakuum */
    { /* OK */
    }
    else
    {
        sendFehler('v');
    }
    if ((!PORT5_4) && (flgAktiv == 0)) /* Bediener Start */
    { /* OK */
        sendStart();
        flgAktiv = 1;
    }
    if ((!PORT5_5) && (flgAktiv == 1)) /* Bediener Stop */
    { /* OK */
        if (ErsteDrehzahl > 0)
        {
            flgAktiv = 0;
            sendStop();
            ErsteDrehzahl = 0;
            PORT4_4 = 1; /* Schwenkarm aus */
            PORT4_3 = 1; /* Spülung aus */
            PORT4_2 = 1; /* Entwickler aus */
            PORT4_0 = 0; /* LED an */
            flgStLED = 0; /* Flag Status LED */
            PORT4_1 = 0; /* Motorsignal aus */
            soll_gsw = 0; /* Sollgeschwindigkeit = 0 */
            Hsoll_gsw = 0; /* Hilfs-Sollgeschwindigkeit = 0 */
        }
    }
    /***** serieller Eingang einer Drehzahl
    oder eines Ereignisses
    *****/
    if (hilf_ser != 0x00)
    {
        if (hilf_ser == 0x73) /* s Start */
        { hilf_ser = 0x00;

```

```

hilf_ser = _getkey();
if (hilf_ser == 0x64) /* d Drehzahl */
{
    hilf_ser = 0x00;
    soll_alt = soll_gsw;
    Hsoll_gsw = _getkey();
    if (Hsoll_gsw != soll_alt) /* neue Drehzahl bestätigen
*/
        {
            ErsteDrehzahl++;
            flgAktiv = 1;
            flgDrehzahl = 1;
            flgStLED = 1; /* Status LED an */
            empfangOK(Hsoll_gsw);
        }
    if (Hsoll_gsw == 0) { PORT4_1 = 0; }
    else { PORT4_1 = 1; }
}
if (hilf_ser == 0x65) /* e Ereignis */
{
    hilf_ser = 0x00;
    hilf_ser = _getkey();
    if (hilf_ser == 0x45) /* E Entwickler AN*/
        {
            hilf_ser = 0x00;
            PORT4_2 = 0;
            empfangOK('E');
        }
    else if (hilf_ser == 0x53) /* S Spülung AN*/
        {
            hilf_ser = 0x00;
            PORT4_3 = 0;
            empfangOK('S');
        }
    else if (hilf_ser == 0x65) /* e Entwickler AUS */
        {
            hilf_ser = 0x00;
            PORT4_2 = 1;
            empfangOK('e');
        }
    else if (hilf_ser == 0x73) /* s Spülung AUS */
        {
            hilf_ser = 0x00;
            PORT4_3 = 1;
            empfangOK('s');
        }
    else if (hilf_ser == 0x41) /* A Schwenkarm AN */
        {
            hilf_ser = 0x00;
            PORT4_4 = 0;
            empfangOK('A');
        }
    else if (hilf_ser == 0x61) /* a Schwenkarm AUS */
        {
            hilf_ser = 0x00;
            PORT4_4 = 1;
            empfangOK('a');
        }
    else if (hilf_ser == 0x75) /* u erste Drehzahl kommt
*/
        {
            hilf_ser = 0x00;
            ErsteDrehzahl = 0;
            flgAktiv = 0;
            empfangOK('u');
        }
    else if (hilf_ser == 0x54) /* T Abfrage Trafotemperatur
Steuerung/Relais */
        {
            hilf_ser = 0x00;
            putchar('s');
            putchar('T');
            putchar(ist_temp_str());
        }
}

```

```

    }
    else if (hilf_ser == 0x56) /* V Abfrage Trafotemperatur
Motor/Filter*/
    { hilf_ser = 0x00;
      putchar('s');
      putchar('V');
      putchar(ist_temp_tmf());
    }
    else if (hilf_ser == 0x52) /* R Abfrage Motorstrom */
    { hilf_ser = 0x00;
      putchar('s');
      putchar('S');
      putchar(ist_strom());
    }
    else if (hilf_ser == 0x58) /* X Stop */
    { hilf_ser = 0x00;
      empfangOK('X');
      flgAktiv = 0;
      ErsteDrehzahl = 0;
      PORT4_4 = 1;
      PORT4_3 = 1;
      PORT4_2 = 1;
      PORT4_0 = 0; /* LED an */
      flgStLED = 0; /* Flag Status LED */
      PORT4_1 = 0; /* Motorsignal aus */
      soll_gsw = 0; /* Sollgeschwindigkeit = 0 */
      Hsoll_gsw = 0; /* Hilfs-Sollgeschwindigkeit = 0
*/
    }
    else /* Fehler */
    { hilf_ser = 0x00;
      putchar('F');
      putchar('e');
      putchar('l');
    }
  }
}
else { hilf_ser = 0x00; }
hilf_ser = 0x00;
} /* if (hilf_ser != 0x00) */
/***** langsame Beschleunigung *****/
if (flgDrehzahl)
{if (ib > 0)
{
  ib = 0;
  if (Hsoll_gsw > (soll_gsw + 1))
  {
    soll_gsw = soll_gsw + 2;
  }
  else if (Hsoll_gsw < (soll_gsw - 1))
  {
    soll_gsw = soll_gsw - 2;
  }
  else
  {
    soll_gsw = Hsoll_gsw;
  }
}
}}
/***** Definition von p *****/
if (soll_gsw > 220) {p = 2; ADkorrektur = -77;}
else if (soll_gsw > 210) {p = 2; ADkorrektur = -70;}
else if (soll_gsw > 200) {p = 2; ADkorrektur = -65;}

```

```
else if (soll_gsw > 180) {p = 2; ADkorrektur = -60;}
else if (soll_gsw > 160) {p = 2; ADkorrektur = -55;}
else if (soll_gsw > 140) {p = 2; ADkorrektur = -50;}
else if (soll_gsw > 120) {p = 2; ADkorrektur = -45;}
else if (soll_gsw > 100) {p = 2; ADkorrektur = -40;}
else if (soll_gsw > 80) {p = 2; ADkorrektur = -35;}
else if (soll_gsw > 76) {p = 2; ADkorrektur = -30;}
else if (soll_gsw > 73) {p = 2; ADkorrektur = -25;}
else if (soll_gsw > 71) {p = 2; ADkorrektur = -20;}
else if (soll_gsw > 70) {p = 1; ADkorrektur = -55;}
else if (soll_gsw > 68) {p = 1; ADkorrektur = -50;}
else if (soll_gsw > 65) {p = 1; ADkorrektur = -45;}
else if (soll_gsw > 60) {p = 1; ADkorrektur = -40;}
else if (soll_gsw > 50) {p = 1; ADkorrektur = -35;}
else if (soll_gsw > 40) {p = 1; ADkorrektur = -30;}
else { p = 1;
      ADkorrektur = -25;}
} /* while(1) */
} /* main */
```

9.3. Abbildungsverzeichnis

Abbildung 1: Ausschnitt der Architektur des SAB 80C535 (Timer 2, Reload- und 3 Compare-Register).....	10
Abbildung 2: Windows-Statusbalken während eines Prozesses.....	12
Abbildung 3: Menüstruktur der Steuerprogramms	14
Abbildung 4: Programmfluß wenn ein Menü-Event eintritt	17
Abbildung 5: Programmstart der Steuerungssoftware	19
Abbildung 6: Das Programmhauptfenster	19
Abbildung 7: Programmablauf der Steuerungssoftware	20
Abbildung 8: Windows - Dialogfenster "Öffnen..."	21
Abbildung 9: Fenster zur Eingabe der Drehzal/Zeit-Paare	21
Abbildung 10: Fensterausschnitt mit den Drehzahl/Zeit-Paaren.....	22
Abbildung 11: Abbruch eines Prozesses (graues Feld).....	22
Abbildung 12: Programmfluß des EINGABE-Fensters.....	23
Abbildung 13: Textfeld für die Projektbeschreibung.....	23
Abbildung 14: Fenster zur Eingabe der projektspezifischen Daten.....	24
Abbildung 15: Fenster zur Eingabe der Optionen	24
Abbildung 16: Parallele und serielle Schnittstellenfenster.....	25
Abbildung 17: Programmfluß beim Freigeben der Prozeßdaten.....	27
Abbildung 18: Programmfluß während des Prozeßablaufes	28
Abbildung 19: Das Hauptfenster während des Prozeßablaufes.....	30
Abbildung 20: Adreßbereich des internen RAM	33
Abbildung 21: Abtastung eines Signals in zeitgleichen Zeitabschnitten	34
Abbildung 22: Bestimmung der Integrationskonstanten.....	35
Abbildung 23: Bestimmung der Differentiationskonstante.....	36
Abbildung 24: Der P-Regelkreis.....	36
Abbildung 25: Der PI-Regelkreis	37
Abbildung 26: Der PID-Regelkreis.....	38
Abbildung 27: Programmstart der Regelungssoftware	40
Abbildung 28: Die Endlosschleife der Regelungssoftware.....	41
Abbildung 29: Der Regelkreis unserer Steuerungseinheit	44

9.4. Tabellenverzeichnis

Tabelle 1: Standardwerte für die Reglervariablen.....	35
Tabelle 2: Analoge und digitale Eingänge und ihre Funktionen.....	42
Tabelle 3: Analoge und digitale Ausgänge und ihre Funktionen.....	43

9.5. Gleichungsverzeichnis

Gleichung 1:	$y_{(t)} = e_{(t)} * P$ mit $e_{(t)} = w_{(t)} - x_{(t)}$	36
Gleichung 2:	$y_{(t)} = e_{(t)} * P + \frac{1}{T_i} \sum e_{(t)}$	37
Gleichung 3:	$y_{(t)} = (e_{(t)} * P) + \frac{1}{T_i} * \sum e_{(t)} + T_D * \frac{de_{(t)}}{dt}$	38
Gleichung 4:	$y_{(t)} = (Sollwert - Istwert) * Verstärkung$	44

9.6. Literaturverzeichnis

- (1) Garbrecht, Friedrich Wilhelm: Digitale Regelungstechnik
VDE-Verlag, Berlin, Offenbach 1991
ISBN 3-8007-1695-X
- (2) Orłowski, Peter F.: Praktische Regelungstechnik
Diagonal, Marburg, 1991
ISBN 3-927165-08-5
- (3) Kluge, Norbert: Lineare zeitdiskrete Systeme
Vorlesungsskript Automatisierungstechnik 2
FH Gießen-Friedberg, Gießen, 1989
- (4) Heesel, Norbert; Reichstein, Werner: Mikrocontroller Praxis
Vieweg Verlag, Braunschweig, 1993
ISBN 3-528-05366-6
- (5) Walter, Jürgen: Mikrocomputertechnik mit der 8051-
Controller-Familie
Springer-Verlag, Berlin, 1994
ISBN 3-540-58272-X
- (6) Mann, Heinz: Einführung in die Regelungstechnik
Hansen Verlag, Wien 1997
ISBN 3-446-17672-1
- (7) Ebel, Tjark: Regelungstechnik
Teubner Verlag, Stuttgart, 1991
ISBN 3-519-00142-X

Eidesstattliche Erklärung

Hiermit erkläre ich, die vorliegende Diplomarbeit selbständig und nur unter Zuhilfenahme der angegebenen Quellen angefertigt zu haben.

Lich, den 10. August 1998

Unterschrift