

Diplomarbeit

Givian Müller

Teststand zur echtzeitfähigen Überwachung von
Werkzeugmaschinen

*Fakultät Technik und Informatik
Department Informations- und
Elektrotechnik*

*Faculty of Engineering and Computer Science
Department of Information and
Electrical Engineering*

Givian Müller

Teststand zur echtzeitfähigen Überwachung von
Werkzeugmaschinen

Diplomarbeit eingereicht im Rahmen der Diplomprüfung
im Studiengang Informations- und Elektrotechnik
Studienrichtung Automatisierungstechnik
am Department Informations- und Elektrotechnik
der Fakultät Technik und Informatik
der Hochschule für Angewandte Wissenschaften Hamburg

Betreuender Prüfer : Prof. Dr. Ing. Reinhard Müller
Zweitgutachter : Prof. Dr. Ing. Henry Reetmeyer

Abgegeben am 18. Dezember 2007

Givian Müller

Thema der Diplomarbeit

Teststand zur echtzeitfähigen Überwachung von Werkzeugmaschinen

Stichworte

Echtzeitfähigkeit, LabVIEW, Profibus, Projektierung, Auswertung

Kurzzusammenfassung

Entwicklung eines Teststandes bestehend aus einem Entwicklungssystem, einer echtzeitfähigen Rechereinheit und einem zu testenden Zielsystem zur echtzeitfähigen Überwachung und Protokollierung des Zeitverhaltens von Messdaten im Bereich der Werkzeugüberwachung.

Entwicklung einer Software zur Generierung von Maschinendaten als Simulationsgrundlage des zeitlichen Verhaltens von Werkzeugmaschinen, Betrachtung und Protokollierung von Profibussignalen während der Messung, sowie zeitkritische Auswertung aller versendeten und empfangenen Profibussignale.

Givian Müller

Title of the paper

Real time test status for monitoring machine tools

Keywords

Real time ability, LabVIEW, Profibus, Project engineering, Evaluation

Abstract

Development of a test status consisting of a development system, a real time computer unit and one target system for real time monitoring and logging of time performance of measuring data, which can be tested in the range of the tool monitoring.

Development of a software for the generation of machine data as simulation basis of the temporal behaviour of machine tools, view and logging of professional bus signals during the measurement and time-critical evaluation of all dispatched and received professional bus signals.

 <p>Hochschule für Angewandte Wissenschaften Hamburg Hamburg University of Applied Sciences</p>	<p>Diplomarbeit Fachbereich Elektrotechnik und Informatik</p>	
<p>Wintersemester 07/08 Abgabetermin: 18.12.07</p>	<p>Givian Müller Matr.Nr.: 1749822</p>	<p>Betreuer: Professor Dr. Ing. Müller</p>

Inhaltsverzeichnis

1	Einleitung	1
1.1	Problematik	2
1.2	Zielsetzung	3
1.2.1	Echtzeitfähigkeit	3
1.2.2	Profibus Master Teststand Software	5
2	Projektplanung	6
2.1	Erstellung des Pflichtenheftes.....	6
2.2	Zeitliche Projekteinteilung.....	6
2.2.1	Entwicklungsrisikoabschätzung	7
2.2.2	Projektphasenplan	7
3	Konzeptphase und Komponentenauswahl	9
3.1	Auswahl einer geeigneten Softwareumgebung	9
3.2	Auswahl einer geeigneten Hardwareumgebung.....	10
3.2.1	Hardwareanforderungen der Entwicklungsumgebung.....	11
3.2.2	Hardwareanforderungen des Realtime PCs.....	11
3.2.3	Hardwareanforderungen des Zielsystems.....	12
4	Konfiguration der Systeme	13
4.1	Hardwarekonfiguration.....	13
4.1.1	Einrichtung der Entwicklungsumgebung	13
4.1.2	Einrichtung der Echtzeitumgebung	14
4.1.3	Einrichtung des Zielsystems	14
4.2	Softwarekonfiguration	14
4.2.1	Verbindungsaufbau zwischen Entwicklungs- und Echtzeitumgebung	14
4.2.2	Einrichtung der Profibusverbindung	16
4.2.3	Kontakt zur Profibuskarte	17
4.2.3.1	Testen der Profibusverbindung.....	18

 <p>Hochschule für Angewandte Wissenschaften Hamburg Hamburg University of Applied Sciences</p>	<p>Diplomarbeit Fachbereich Elektrotechnik und Informatik</p>	
<p>Wintersemester 07/08 Abgabetermin: 18.12.07</p>	<p>Givian Müller Matr.Nr.: 1749822</p>	<p>Betreuer: Professor Dr. Ing. Müller</p>

5	Implementierung der Software	20
5.1	Projekte unter LabVIEW	20
5.2	Erstellen eines Projektes	20
5.2.1	Struktur im Projekt Explorer	21
5.2.2	Grafische Modulgestaltung	23
5.3	Kommunikationsvariablen.....	23
5.3.1	Eigenschaften von Kommunikationsvariablen.....	24
5.3.1.1	Problematik bei zu großer Datenmenge.....	25
6	Programmierung der Software	26
6.1	Module zum Umwandeln von Datenformaten	26
6.1.1	Modulbeschreibung Byte to Bool (SubVI)	27
6.1.2	Modulbeschreibung Byte to Bit (SubVI)	27
6.1.3	Modulbeschreibung Word to Byte-Array (SubVI)	28
6.1.4	Modulbeschreibung DWord to Byte Array (SubVI)	28
6.1.5	Modulbeschreibung Byte-Array to Word (SubVI)	28
6.2	Softwaremodule der Echtzeitumgebung	29
6.2.1	Initialisierung auf der Echtzeitumgebung	29
6.2.1.1	Modulbeschreibung Variable to Bus (SubVI)	30
6.2.2	Durchführung der Messung	31
6.2.3	Speichern der Messwerte	32
6.2.4	Problem der Echtzeitanforderung	33
6.3	Softwaremodule zur Onlinebetrachtung der Messung.....	33
6.3.1	Bedienoberflächen der Onlinebetrachtung.....	34
6.4	Modul zur Projektierung der Signalverläufe	35
6.4.1	Textdateien der gesendeten und empfangenen Signale	35
6.4.2	Bedienoberfläche der Projektierung.....	36
6.4.3	Bedienung der Projektierung	37
6.4.4	Eigenschaften der Projektierung.....	38
6.4.4.1	Auswahl des Kartentyps	38
6.4.4.2	Eingabe des Quellverzeichnisses	39
6.4.4.2.1	Modulbeschreibung Input to Variablen (SubVI).....	40
6.4.4.3	Eingabe von Bezeichnung und Kommentar	40
6.4.4.4	Erstellen einer Kopie der Datei	40
6.4.4.5	Veränderung der Taktung und Leserate	40

 Hochschule für Angewandte Wissenschaften Hamburg Hamburg University of Applied Sciences	Diplomarbeit Fachbereich Elektrotechnik und Informatik	
Wintersemester 07/08 Abgabetermin: 18.12.07	Givian Müller Matr.Nr.: 1749822	Betreuer: Professor Dr. Ing. Müller

6.4.4.5.1	Modulbeschreibung Output Variablen Init (SubVI)	41
6.4.4.6	Veränderung der Wertemenge	41
6.4.4.7	Veränderung der Signalverläufe	42
6.4.4.7.1	Kanal 1 für alle Kanäle übernehmen.....	43
6.4.4.8	Speichern der Daten.....	44
6.4.4.8.1	Modulbeschreibung Input Kopfzeile (SubVI)	44
6.4.4.9	Kanalauswahl des Signalverlaufes	44
6.4.4.9.1	Modulbeschreibung Datei Inputzeile auslesen (SubVI).....	45
6.4.4.9.2	Skalierung des Signalverlaufes.....	45
6.4.4.10	Ansteuerung des Cursors	46
6.4.4.10.1	Modulbeschreibung Tastendruck (SubVI)	46
6.4.4.11	Einhalten der Grenzwerte	47
6.4.4.12	Sperren der Eingabe während der Bearbeitung	47
6.4.4.13	Besonderheit während der Messung	48
6.5	Softwaremodul zur Auswertung der Simulation.....	48
6.5.1	Bedienoberfläche der Auswertung	48
6.5.2	Bedienung der Auswertung.....	51
6.5.3	Eigenschaften der Auswertung	52
6.5.3.1	Auswahl des Kartentyps	52
6.5.3.2	Einlesen der letzten Messung.....	52
6.5.3.2.1	Modulbeschreibung Output Kopfzeile (SubVI)	53
6.5.3.2.2	Modulbeschreibung Variablen to Output (SubVI)	53
6.5.3.3	Zuweisen der Datenquelle beim Starten der Auswertung	53
6.5.3.4	Messwerte in neuer Datei speichern.....	53
6.5.3.5	Veränderung der Datenquellen.....	54
6.5.3.6	Einstellen der darzustellenden Signale	54
6.5.3.7	Analysieren von Input und Output Daten	55
6.5.3.8	Skalierung der Achsen.....	55
6.5.3.8.1	Alternativlösung zur Überprüfung der Grenzwerte	56
6.5.3.9	Ansteuerung des Cursors	56
6.5.3.10	Signalverlauf als Grafik exportieren	56
7	Testphase der Hard- und Software.....	57
7.1	Testen der Echtzeit.....	57
7.1.1	Software zum testen der Echtzeit	57
7.1.2	Oszilloskopauswertung der Echtzeit	58
7.2	Testen der minimalen Laufzeit.....	58
7.2.1	Laufzeittest mit kontinuierlicher Kommunikation	59
7.2.2	Laufzeittest bei vorheriger Initialisierung	59

 Hochschule für Angewandte Wissenschaften Hamburg Hamburg University of Applied Sciences	Diplomarbeit Fachbereich Elektrotechnik und Informatik	
Wintersemester 07/08 Abgabetermin: 18.12.07	Givian Müller Matr.Nr.: 1749822	Betreuer: Professor Dr. Ing. Müller

7.3	Abschließender Funktionstest.....	60
7.3.1	Funktionstest des Projektieren eines Kurvenverlaufes.....	60
7.3.2	Funktionstest der Messdurchführung.....	62
7.3.3	Funktionstest der Signalauswertung	62
7.3.3.1	Vorbereitungen für den Funktionstest der Auswertung	62
7.3.3.1.1	Betrachtung der vorbereiteten Signale auf dem Zielsystem	63
7.3.3.2	Starten der Auswertung ohne vorherige Messung	65
7.3.3.3	Starten der Auswertung mit vorheriger Messung	65
7.3.4	Performancetest von CTM und Genior	66
7.3.4.1	Spezialfall im Performancetest	67
8	Testauswertung	70
8.1	Unterschiedliche Antwortzeiten der Karten	70
8.2	Unterschiedliche Kanalreihenfolge der Alarmsignale	70
9	Abschließender Projektbericht	71
9.1	Einhaltung des Zeitplanes	71
9.2	Beurteilung des entwickelten Teststandes	72
10	Erweiterungsmöglichkeiten des Teststandes	73
10.1	Erweiterung der Hardwareumgebung	73
10.2	Erweiterung der Software	73
10.2.1	Einfügen neuer Profibussignale	73
10.2.2	Überprüfung der Profibusverbindung	76
10.2.3	Kartenunabhängige Softwarelösung	76
10.2.3.1	Kartentyp während der Projektierung ändern.....	76
10.2.3.2	Globales kartenunabhängiges Gesamtkonzept.....	77
10.2.4	Speichern der Daten im Dauerbetrieb.....	77
10.2.5	Automatische Protokollierung der Ergebnisse	77
10.2.6	Teststand im Multiuserbetrieb.....	78
10.2.6.1	Problematik im Multiuserbetrieb.....	78
10.2.6.1.1	Scheduling Verfahren mit festen Prioritäten.....	79
11	Fazit	80

 Hochschule für Angewandte Wissenschaften Hamburg Hamburg University of Applied Sciences	Diplomarbeit Fachbereich Elektrotechnik und Informatik	
Wintersemester 07/08 Abgabetermin: 18.12.07	Givian Müller Matr.Nr.: 1749822	Betreuer: Professor Dr. Ing. Müller

Abbildungsverzeichnis

Abb. 1	Prinzipdarstellung CTM System ₁	2
Abb. 2	Profibus Master Teststand Software Oberfläche	5
Abb. 3	Entwicklungsrisikoabschätzung	7
Abb. 4	Projektphasen nach dem COCOMO Modell	8
Abb. 5	Geplanter Zeitaufwand des Projektes	8
Abb. 6	Aufbau der Hardwareumgebung des Teststandes	10
Abb. 7	Realtime System Check Software.....	12
Abb. 8	Realtime Bootdiskette erstellen	13
Abb. 9	Netzwerkkomponenten hinzufügen.....	15
Abb. 10	Software auf die Echtzeitumgebung laden.....	15
Abb. 11	Einrichtung der Profibusverbindung	16
Abb. 12	Profibus Configuration Download	17
Abb. 13	DF PROFI 2 - RT Installation	17
Abb. 14	Testen der Profibusverbindung	18
Abb. 15	Verbindung zum Zielsystem hergestellt	19
Abb. 16	LabVIEW Projekt erstellen.....	20
Abb. 17	LabVIEW Projekttyp wählen	21
Abb. 18	Struktur im Projekt Explorer.....	22
Abb. 19	Beispiele der Modulgestaltung.....	23
Abb. 20	Programmcode Byte to Bool (SubVI)	27
Abb. 21	Programmcode Byte to Bit (SubVI).....	27
Abb. 22	Programmcode Word to Byte Array (SubVI)	28
Abb. 23	Programmcode DWord to Byte Array (SubVI).....	28
Abb. 24	Programmcode Byte Array to Word (SubVI)	28
Abb. 25	Programmcode Initialisierung der Messung auf dem Target	30
Abb. 26	Programmcode Variable to Bus (SubVI)	31
Abb. 27	Programmcode Durchführung der Messung auf dem Target	32
Abb. 28	Programmcode Messanzahl auf dem Target reduzieren.....	32
Abb. 29	Bedienoberflächen der Onlinebetrachtung der Messung	34
Abb. 30	Genior Input Datei	35

 Hochschule für Angewandte Wissenschaften Hamburg Hamburg University of Applied Sciences	Diplomarbeit Fachbereich Elektrotechnik und Informatik	
Wintersemester 07/08 Abgabetermin: 18.12.07	Givian Müller Matr.Nr.: 1749822	Betreuer: Professor Dr. Ing. Müller

Abb. 31	Genior Output Datei.....	35
Abb. 32	Bedienoberfläche der Projektierung.....	36
Abb. 33	Bedienoberfläche Auswahl des Kartentyps.....	39
Abb. 34	Programmcode Input to Variablen (SubVI)	40
Abb. 35	Programmcode Kurve erzeugen	42
Abb. 36	Programmcode Kanal 1 für alle Kanäle übernehmen (CTM).....	43
Abb. 37	Programmcode Datei Inputzeile auslesen (SubVI).....	45
Abb. 38	Programmcode Ansteuerung des Cursors	46
Abb. 39	Programmcode Tastendruck (SubVI).....	46
Abb. 40	Bedienoberfläche der Auswertung (Initialisieren)	49
Abb. 41	Bedienoberfläche der Auswertung (Single Plot).....	50
Abb. 42	Bedienoberfläche der Auswertung (Multi Plot)	50
Abb. 43	Programmcode Einlesen der letzten Messung.....	52
Abb. 44	Programmcode Messwerte in neuer Datei speichern.....	54
Abb. 45	Programmcode Grenzwerte der Achsskalierung überprüfen.....	55
Abb. 47	Programmcode Echtzeittest.....	58
Abb. 48	Oszilloskopauswertung der Echtzeit	58
Abb. 49	Funktionstest der Projektierung des Kurvenverlaufes	61
Abb. 50	Signalverlauf zum Funktionstest der Auswertung	63
Abb. 51	Lernkurve auf CTM.....	63
Abb. 52	Wiederholter Kurvenverlauf auf CTM.....	64
Abb. 53	Simulierter Bruch auf CTM	65
Abb. 54	Performance Genior 8 Kanäle, BAV ein, unsynchron zum Vorschub.....	67
Abb. 55	Performance Genior 8 Kanäle, BAV aus, synchron zum Vorschub.....	69
Abb. 56	Performance Genior 8 Kanäle, BAV aus, synchron zum Vorschub 2.....	69
Abb. 57	Realer Zeitaufwand des Projektes	71
Abb. 58	Scheduling Prioritäten im Multiuserbetrieb.....	79
Abb. 59	Scheduling Ablaufdiagramm im Multiuserbetrieb	79

 Hochschule für Angewandte Wissenschaften Hamburg Hamburg University of Applied Sciences	Diplomarbeit Fachbereich Elektrotechnik und Informatik	
Wintersemester 07/08 Abgabetermin: 18.12.07	Givian Müller Matr.Nr.: 1749822	Betreuer: Professor Dr. Ing. Müller

1 Einleitung

Die ARTIS GmbH entwickelt, fertigt und vertreibt innovative und anwendungsorientierte Systeme zur Werkzeug-, Prozess- und Maschinenzustandsüberwachung, die ihren Einsatz zur Prozessoptimierung insbesondere in Metall zerspanenden Produktionsanlagen finden. Dabei befassen sich die Mitarbeiter sowohl mit der Entwicklung von Hardware in Form von Sensoren, Sensormodulen, Messkarten und kompletten Überwachungseinheiten, als auch mit der dazugehörigen Softwareentwicklung. In der Produktpalette der Hardwarekomponenten der Firma befinden sich unter anderem zwei Messkarten mit den Bezeichnungen CTM¹ und Genior², die für diese Diplomarbeit von Interesse sind. Beide Messkarten sind PCI³ Einsteckkarten, die in normalen Desktop PCs⁴ oder in speziell konfigurierten Rechneinheiten verbaut werden.

Der Unterschied der Karten liegt sowohl in der Hardwarebeschaffenheit als auch in der Software. Die Genior Karte ist auf Grund ihrer Hardwareeigenschaften wesentlich leistungsfähiger als die CTM Karte. Darüber hinaus ist die Programmstruktur und die generelle Gestaltung der Software eine völlig andere. So verfügt CTM beispielsweise über die Möglichkeit Messsignale von vier verschiedenen Sensoren über entsprechende Kanäle zu überwachen, Genior hingegen unterstützt die Überwachung von bis zu acht Kanälen.

Da eine weitere detaillierte Beschreibung der Eigenschaften und Funktionen der Karten für den Teststand zu diesem Zeitpunkt nicht von Bedeutung sind, sollen diese nur kurz anhand der Prinzipdarstellung einer CTM Karte erläutert werden.

Die Aufgabe der Karte ist es, Messwerte über folgende Sensormodule der Firma zu überwachen:

CFM-4 ⁵ :	Überwachung von Dehnung und Kraft
MU 4 ⁶ :	Überwachung von Wirkleistung
KU-4 ⁷ :	Überwachung von Körperschall
VG-4 ⁸ :	Überwachung von Beschleunigung
DDU4 ⁹ :	Überwachung von Drehmoment und Vorschubkraft
DTA-1-DP/S ¹⁰ :	Sensorlose digitale Überwachung von Signalen direkt über Profibus

¹ Computer Integrated Tool and Machine Monitoring

² Synonym / Produktname

³ Peripheral Component Interconnect

⁴ Personal Computer

⁵ Kollisionskraft Messumformer Serie 4

⁶ Messumformer Serie 4

⁷ Körperschallumformer Serie 4

⁸ Beschleunigungs- und Schwinggeschwindigkeitsumformer Serie 4

⁹ Double Drehmoment Unit / Vorschub und Achsialkraft Umformer Serie 4

¹⁰ Daten Träger Austauschverfahren / Digital Performer

 Hochschule für Angewandte Wissenschaften Hamburg Hamburg University of Applied Sciences	Diplomarbeit Fachbereich Elektrotechnik und Informatik	 ARTIS ®
Wintersemester 07/08 Abgabetermin: 18.12.07	Givian Müller Matr.Nr.: 1749822	Betreuer: Professor Dr. Ing. Müller

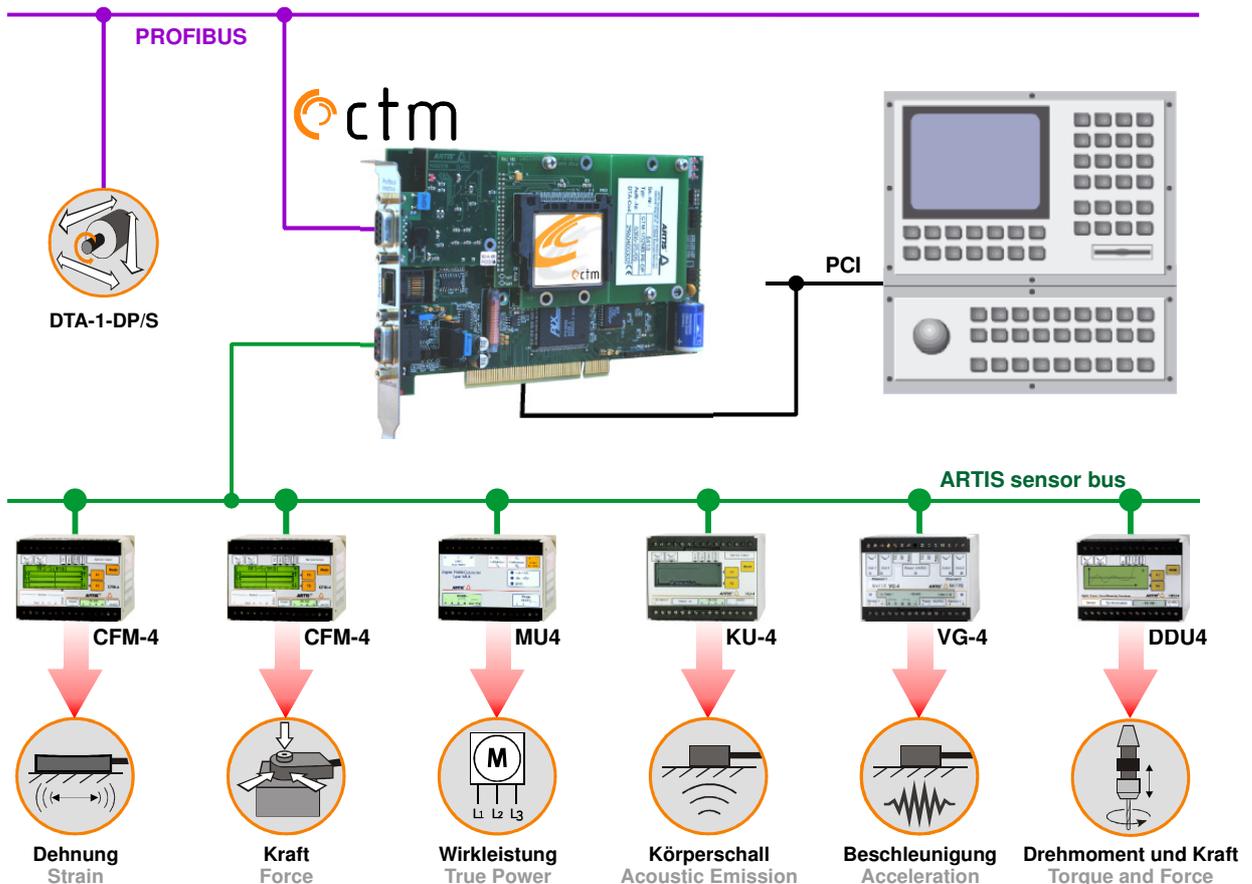


Abb.¹¹ 1 Prinzipdarstellung CTM System₁

1.1 Problematik

In der Testabteilung der Firma Artis soll das Verhalten beider Karten unter möglichst realen Bedingungen getestet werden. Um dieses realisieren zu können, müsste jedem Entwickler oder Tester eine eigene NC¹² Maschine zur Verfügung stehen. Dieses ist aber zum einen aus finanziellen, zum anderen aus platztechnischen Gründen nicht umsetzbar. Dazu kommt noch, dass für unterschiedliche Anforderungen unterschiedliche Maschinen angeschafft werden müssten.

Ein weiteres Problem liegt darin, dass die Maschinen auf Grund der mechanischen Komponente bei gleicher Bearbeitung keine exakt gleichen Daten senden würden, wodurch eine reproduzierbare Messung unmöglich wäre. Ebenso wäre es nicht möglich jeden Mitarbeiter auf jeder Maschine auszubilden, um zu verhindern, dass es bei falscher Bedienung zu ungewollten Störfällen kommt. Im Umkehrschluss dazu ist auch ein gewolltes Herbeiführen eines Störfalles schlecht umzusetzen.

¹ Dr. Dirk Lange (Artis): Concept Rolls Royce in: Concept_Rolls_Royce_2007_08_21_DL, 23.08.2007

¹¹ Abbildung
¹² Numerical Control

 Hochschule für Angewandte Wissenschaften Hamburg Hamburg University of Applied Sciences	Diplomarbeit Fachbereich Elektrotechnik und Informatik	
Wintersemester 07/08 Abgabetermin: 18.12.07	Givian Müller Matr.Nr.: 1749822	Betreuer: Professor Dr. Ing. Müller

Der wesentliche Punkt der zur Erstellung dieser Diplomarbeit geführt hat, ist jedoch die Tatsache, dass die Reaktionszeiten auf verschiedenste auftretende Signalereignisse, bei beiden Karten nicht exakt oder gar nicht getestet werden können. Gefordert sind Reaktionszeiten auf auftretende Ereignisse innerhalb des folgenden Kompilyklustaktes der Karten. Dieser liegt bei beiden Karten bei 10 ms¹³. Ob diese Reaktionszeit erfüllt wird ist jedoch fraglich. Um derartige Tests durchzuführen, müssten sämtliche Profibustelegamente zeitgenau protokolliert werden.

Derzeit ist bereits eine Profibus Master Teststand Software im Einsatz. Allerdings verfügt diese nicht über den gewünschten Funktionsumfang der als Anforderung für diesen Teststand zu Grunde liegt.

1.2 Zielsetzung

Um Signale zeitgenau zu protokollieren, muss eine Echtzeitumgebung geschaffen werden, die zwei wesentliche Grundaufgaben erfüllt.

- a. Echtzeitfähige Überwachung und Protokollierung des Zeitverhaltens aller eingehenden und ausgehenden Profibussignale.
Da die Karten mit einem Zyklustakt von 10 ms arbeiten, sollte diese Laufzeit als Minimalanforderung ebenfalls eingehalten werden. Dadurch wird der Karte ermöglicht zu jedem Zeitpunkt neue Signale senden zu können und ebenso zu jedem Zeitpunkt neue Signale von der Karte erfassen zu können.
- b. Simulation des Verhaltens einer Werkzeugmaschine, um mit den Karten CTM und Genior über die Profibuschnittstelle zu kommunizieren. Ablösung der derzeitig verwendeten Profibus Master Teststand Software.

1.2.1 Echtzeitfähigkeit

Echtzeitfähigkeit wird wie folgt definiert.

*„Von **Echtzeitsystemen** (englisch real-time system) spricht man, wenn ein System ein Ergebnis innerhalb eines vorher fest definierten Zeitintervalles garantiert berechnet, also bevor eine bestimmte Zeitschranke erreicht ist. Die Größe des Zeitintervalles spielt dabei keine Rolle: während bei einigen Aufgaben (Motorsteuerung) eine Sekunde bereits zu lang sein kann, reichen für andere Probleme Stunden oder sogar Tage. Ein Echtzeitsystem muss also nicht nur ein Berechnungsergebnis mit dem richtigen Wert, sondern dasselbe auch noch **rechtzeitig** liefern. Andernfalls hat das System versagt.*

Echtzeit beschreibt das zeitliche Ein- und Ausgangsverhalten eines Systems, sagt aber nichts über dessen Realisierung aus. Ein Echtzeitsystem kann ein Rechner mit einer geeigneten Software, aber auch eine reine Hardwarelösung sein. Für Anforderungen mit so genannten weichen Grenzen werden normalerweise reguläre EDV¹⁴-Systeme verwendet.

¹³ Millisekunden

¹⁴ Elektronische Daten Verarbeitung

 Hochschule für Angewandte Wissenschaften Hamburg Hamburg University of Applied Sciences	Diplomarbeit Fachbereich Elektrotechnik und Informatik	
Wintersemester 07/08 Abgabetermin: 18.12.07	Givian Müller Matr.Nr.: 1749822	Betreuer: Professor Dr. Ing. Müller

Für Anforderungen mit harten Grenzen werden spezielle Architekturen (Hardware und Software) verwendet. Prinzipiell ist auch ein PC echtzeitfähig, allerdings nicht oder nur sehr bedingt, wenn er mit klassischen Multitasking-Betriebssystemen betrieben wird. Echtzeitbetriebssysteme sind oftmals ebenfalls multitaskingfähig, verfügen jedoch über einen anderen Scheduler als konventionelle Systeme. Es gibt auch Lösungen, bei denen ein bestehendes Standardbetriebssystem durch hinzufügen spezieller Software echtzeitfähig gemacht wird. Dies hat den Vorteil, dass nur die wirklich zeitkritischen Vorgänge im Echtzeitsystem ablaufen müssen und für den Rest die normalen APIs¹⁵ (inklusive Compiler oder GUIs¹⁶) des zugrunde liegenden Betriebssystems verwendet werden können.

Die Software, die unter Echtzeitbedingungen läuft, muss einige Eigenschaften, insbesondere bei harten Grenzen, aufweisen:

- *Die maximale Laufzeit eines Moduls muss berechenbar sein und darf keinen oder nur bedingt beeinflussbaren Faktoren unterliegen. Dies ist ein Problem bei komplexer I/O¹⁷ (z.B.¹⁸ Festplatte mit Cache und automatischem Ruhezustand) oder bei einem Netzwerk mit nicht deterministischem Zeitverhalten (z. B. TCP / IP¹⁹). Es muss daher dafür gesorgt werden, dass die Echtzeitmodule von der virtuellen Speicherverwaltung des Betriebssystems unbeeinflusst bleiben und niemals ausgelagert werden (typischerweise verwenden Echtzeitsysteme deshalb überhaupt keine virtuelle Speicherverwaltung).*
- *Bei Rekursion muss die maximale Rekursionstiefe, bei Schleifen muss die maximale Anzahl an Iterationen feststehen.*
- *Der Bedarf an Ressourcen, insbesondere der Bedarf an Speicher muss bekannt sein. Die Laufzeitumgebung und die Hardware müssen den Ressourcenbedarf decken können.*
- *Die Laufzeit von Betriebssystemaufrufen und von Routinen der Laufzeitumgebung muss mit berücksichtigt werden. Problematisch ist hier zum Beispiel der Einsatz automatischer Speicherbereinigung (Garbage Collector), dessen Laufzeit sehr pessimistisch abgeschätzt werden muss.*
- *Das Verhalten bei drohender Zeitüberschreitung muss definiert und vorhersehbar sein“₂*

² o.V²⁰: Comp.realtime: Frequently Asked Questions (FAQs), in:
<http://de.wikipedia.org/wiki/Echtzeitf%C3%A4higkeit>, 20.09.2007

¹⁵ Adaptive Planungsintelligenzen (mathematische Methoden für industrielle Planungsaufgaben)

¹⁶ Grafische Benutzeroberflächen

¹⁷ Input / Output

¹⁸ zum Beispiel

¹⁹ Transmission Control Protokoll / Internet Protokoll

²⁰ ohne Verfasser

1.2.2 Profibus Master Teststand Software

Die „Profibus Master Teststand Software“ ist ähnlich der hier zu entwickelnden Software. Sie dient den Entwicklern zum Übertragen von vorhandenen gespeicherten Maschinendaten über die Profibusschnittstelle zu den Messkarten, ebenso wie der Betrachtung der von den Karten zurückgesendeten Signale. Nahe liegend wäre es, diese Software derartig weiter zu entwickeln, dass sie die gewünschten Anforderungen erfüllt. Dieses ist aber aufgrund mehrerer Punkte nicht realisierbar.

- Es werden nur die zu jedem Zeitpunkt anliegenden Signalwerte dargestellt und keine Daten gespeichert, was ein späteres Betrachten der Signale unmöglich macht.
- Die Software läuft auf einem Desktop PC unter Windows und verfügt somit über keinerlei Echtzeitfähigkeit um zeitkritische Signalverläufe zu analysieren.
- Zum Betreiben der Software wird eine ISA²¹ Einsteckkarte benötigt, die nur noch in älteren Systemen verwendet werden kann. Eine Umrüstung auf eine neue PCI Einsteckkarte würde erheblichen Mehraufwand oder sogar eine völlig neue Programmierung bedeuten.
- Die Software ermöglicht es nicht direkt Signalverläufe zu generieren, sondern kann nur bereits vorhandene Maschinendatensätze verwenden.

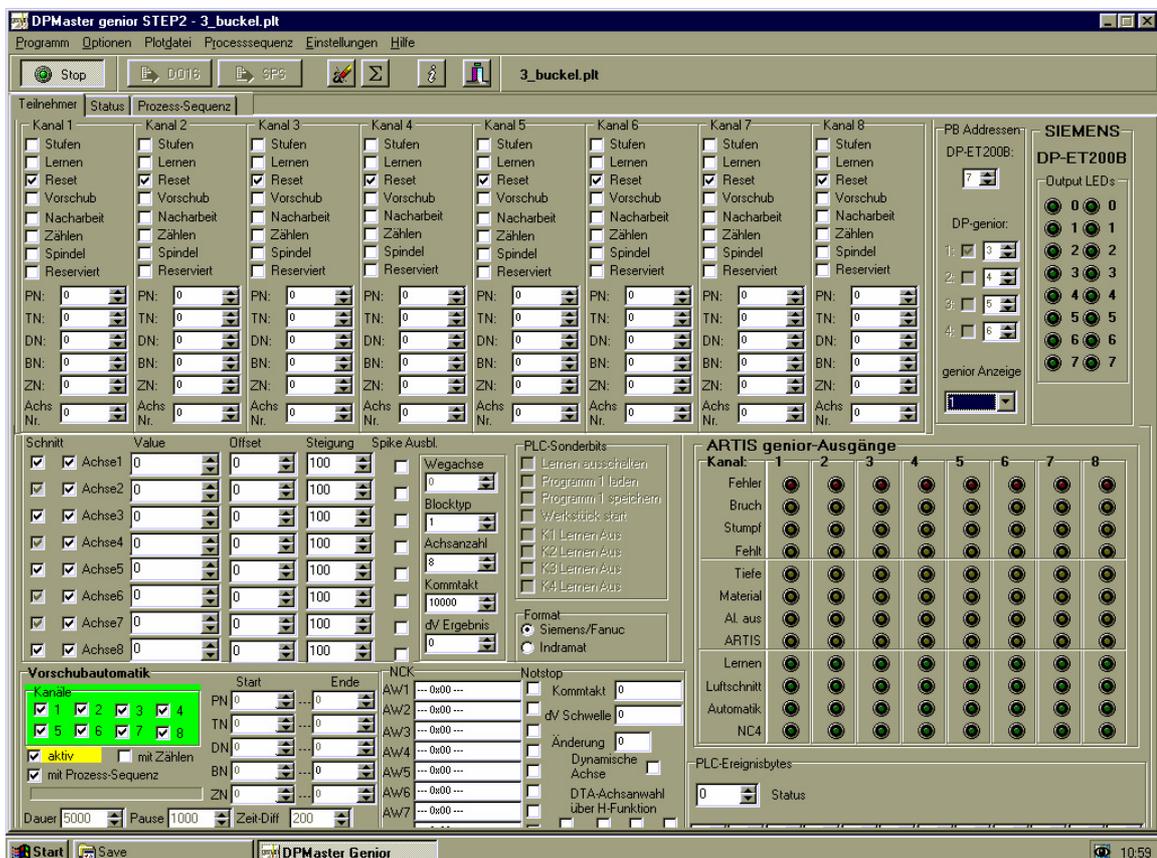


Abb. 2 Profibus Master Teststand Software Oberfläche

²¹ Industry Standard Architecture

 Hochschule für Angewandte Wissenschaften Hamburg Hamburg University of Applied Sciences	Diplomarbeit Fachbereich Elektrotechnik und Informatik	
Wintersemester 07/08 Abgabetermin: 18.12.07	Givian Müller Matr.Nr.: 1749822	Betreuer: Professor Dr. Ing. Müller

2 Projektplanung

Beim optimalen Ablauf eines Projektes ist der gesamte Umfang der Anforderungen im Vorfeld bekannt, so dass zu diesem Zeitpunkt ein Pflichtenheft erstellt werden kann. In diesem Projekt war dieses jedoch nicht möglich, da innerhalb der Firma niemand genaue Vorstellungen davon hatte, was im Themenbereich der Echtzeitfähigkeit im Bezug auf diesen Teststand zu realisieren ist. Um einen Überblick zu bekommen, welche Funktionalitäten für die Mitarbeiter relevant sind, wurde zunächst die bereits in der Firma eingesetzte Profibus Master Teststand Software mit ähnlicher Funktionalität betrachtet. Nachdem mehrere Mitarbeiter, die mit der Software arbeiten, die Funktionen eingehend erläutert haben, ergab sich ein Überblick über ihre Funktionalität. Darüber hinaus ergab sich eine Vielzahl von Anforderungen, die für das zu entwickelnde System als Erweiterung wünschenswert wären. Da zu diesem Zeitpunkt nicht zu überschauen war, welche dieser Anforderungen sowohl technisch als auch zeitlich im Rahmen der Diplomarbeit realisierbar waren, war es nicht möglich diese in unterschiedliche Prioritäten einzuteilen. Deswegen wurde zunächst nur der grobe Rahmen des Projektes im Pflichtenheft festgelegt. Die genaue Anforderungsspezifikation wurde erst im Laufe des Projektes, nachdem die Machbarkeit geprüft wurde, erstellt.

2.1 Erstellung des Pflichtenheftes

In der Ursprungsversion des Pflichtenheftes wurden die grundlegende Beschreibung der Funktionen des Teststandes, sowie die Beschreibung der möglichen Einsatzgebiete beschrieben. In den weiteren Versionen wurden die Art der Kommunikation zwischen den Modulen, die Beschreibung des Funktionsumfanges, die Art der Oberflächengestaltung, sowie die Bedienbarkeit beschrieben. Das letzte Kapitel des Pflichtenheftes beinhaltet eine Auflistung möglicher Erweiterungen des Teststandes.

Siehe Anhang I Pflichtenheft.

2.2 Zeitliche Projekteinteilung

Im Zuge der Projektplanung war im Vorfeld zu prüfen, welcher zeitliche Rahmen für die einzelnen Aufgabenbereiche anzusetzen ist. Um den Fertigstellungstermin einhalten zu können, musste im Vorfeld eine Risikoabschätzung des Projektes erfolgen. Dafür besitzt die Firma eine eigens für diesen Zweck entworfene, auf dem COCOMO²² Modell basierende, Vorlage, mit deren Hilfe unerwartete Risikofaktoren, die das Projekt verzögern könnten, ermittelt wurden. Nach der Eingabe aller bekannten relevanten Größen ergab sich ein Risikofaktor von 1,36. Mit diesem Faktor ist jede zeitliche Konstante zu multiplizieren, um eine realistische Abschätzung zu erhalten.

²² COConstructive COst MOdel

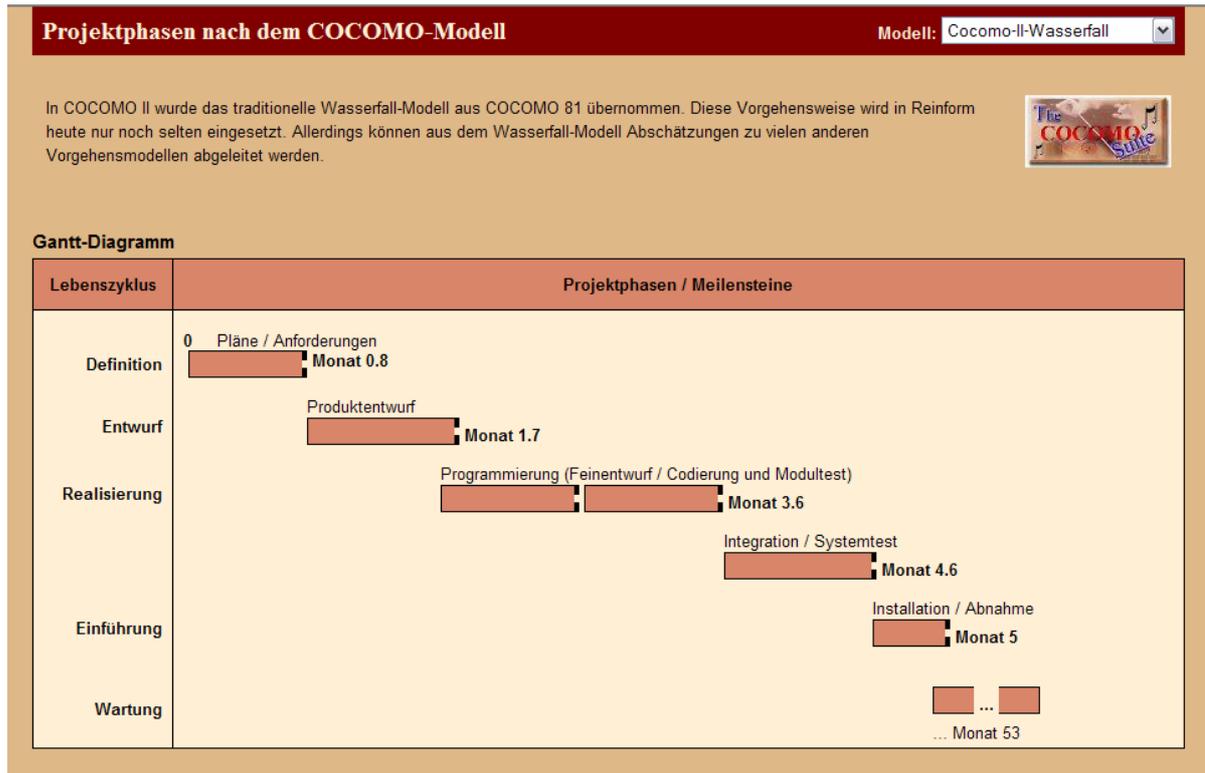


Abb. 4 Projektphasen nach dem COCOMO Modell

Das Modell vermittelt einen Einblick darüber, welcher Zeitaufwand nach der Realisierung des Projektes, in diesem Fall die Programmierung der Software, noch zu kalkulieren ist. Da dieses Diagramm das Anfertigen der Diplomarbeit nicht mit beinhaltet und auch die Einteilung der Projektphasen / Lebenszyklus für die Vielfalt der Arbeitsschritte des Projektes nicht detailliert genug sind, wurde eine weitere Zeiteinteilung erstellt.

Mit Hilfe des Risikofaktors von 1,36, den Erfahrungswerten aus dem COCOMO- Modell und der Einbeziehung der Diplomarbeit ergibt sich folgender geplante Zeitaufwand für die Realisierung des gesamten Projektes.

Geplanter Zeitaufwand

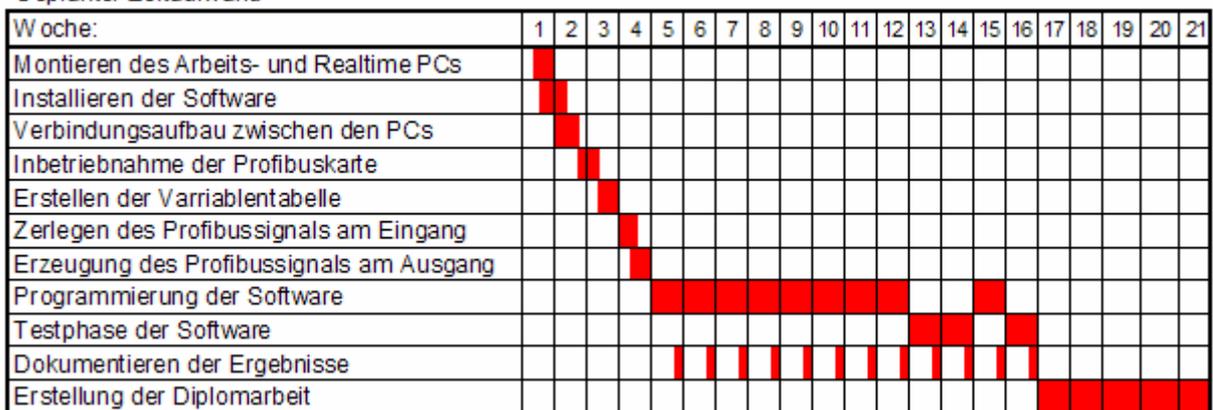


Abb. 5 Geplanter Zeitaufwand des Projektes

 Hochschule für Angewandte Wissenschaften Hamburg Hamburg University of Applied Sciences	Diplomarbeit Fachbereich Elektrotechnik und Informatik	
Wintersemester 07/08 Abgabetermin: 18.12.07	Givian Müller Matr.Nr.: 1749822	Betreuer: Professor Dr. Ing. Müller

3 Konzeptphase und Komponentenauswahl

3.1 Auswahl einer geeigneten Softwareumgebung

Als geeignete Programmiersprache kommen für dieses Projekt gängige textbasierte Programmiersprachen, wie C und C++, oder grafikorientierte Programmiersprachen, wie beispielsweise LabVIEW, in Frage. Für den Teststand wurde sich für die Programmierung unter LabVIEW entschieden. Für diese Entscheidung waren folgende Punkte ausschlaggebend.

- a. Die Programmierung unter LabVIEW erfolgt, aufgrund der geringeren Einarbeitungszeit, im Vergleich zu textbasierten Programmiersprachen, wesentlich schneller. Dadurch können Änderung oder Erweiterung schneller und einfacher durchgeführt werden.
- b. LabVIEW verfügt über eine gesonderte Realtime Software, welche die Programmierung echtzeitfähiger Programme ermöglicht, wie sie für diesen Teststand benötigt werden. Für die Echtzeitumgebung wird dadurch keine weitere Software mehr benötigt.
- c. Die Anforderung an das Betriebssystem ist relativ gering. LabVIEW benötigt Windows 2000 oder höher.
- d. Eine Vielzahl von Hardwareherstellern unterstützt die Einbindung ihrer Komponentensoftware in die LabVIEW Programmierung.
- e. Der einzige negative Punkt ist der Preis der benötigten Software. Während beispielsweise eine Programmierung unter C++ keine weiteren Programmkosten verursachen würde, sind für LabVIEW folgende Programmmodule erforderlich.
 1. LabVIEW Update auf Version 8.5 für 1200€
 2. LabVIEW RT 8.5 + Ethernet Karte für 2400€
 3. LabVIEW Application Builder für 1300€

Für die erforderliche Einzelplatzlösung ist der Application Builder zwar noch nicht erforderlich, aber in Hinsicht auf die zukünftige Verwendung von mehreren Mitarbeitern ist er in der Kalkulation mit inbegriffen.

Jeder weitere Arbeitsplatz würde dadurch lediglich eine Runtime Licence für 820€ benötigen.

Der wesentliche Punkt für die Entscheidung die Programmierung unter LabVIEW durchzuführen beruht auf dem Punkt b. Denn dadurch ist, direkt durch das Programm, die Kontrolle der Echtzeitfähigkeit gegeben.

 Hochschule für Angewandte Wissenschaften Hamburg Hamburg University of Applied Sciences	Diplomarbeit Fachbereich Elektrotechnik und Informatik	
Wintersemester 07/08 Abgabetermin: 18.12.07	Givian Müller Matr.Nr.: 1749822	Betreuer: Professor Dr. Ing. Müller

3.2 Auswahl einer geeigneten Hardwareumgebung

Der zu entwickelnde Teststand soll Laufzeiten im Millisekundenbereich garantieren. Dafür muss ein geeignetes Zielsystem geschaffen werden, welches diese Vorgaben ausnahmslos erfüllt. PCs mit dem Betriebssystem Windows verfügen nicht über eine derartige Funktionalität, deswegen muss eine Umgebung ohne gängiges Betriebssystem geschaffen werden. Die einfachste und kostengünstigste Variante ist die Nutzung eines Desktop PCs auf dem eine Echtzeitanwendung direkt im Speicher abläuft.

Als Echtzeitumgebung können auch spezielle echtzeitfähige Baugruppen der Firma NI²³ verwendet werden. Diese sind genau für derartige Messungen konzipiert und laufen stabil mit Zykluszeiten im Mikrosekundenbereich. Nachteil dieser Module ist jedoch der Preis von mindestens 2000 € pro Baugruppe. Aus diesem Grund wurde die Variante des echtzeitfähigen Desktop PCs gewählt.

Bei dieser Variante muss die Steuerung und Auswertung der Messsignale über einen weiteren PC der eigentlichen Entwicklungsumgebung erfolgen. Dieser wird über die Ethernet- Schnittstelle mit dem Realtime PC verbunden. Für die zu messenden Karten CTM und Genior wird ein weiterer PC als Zielsystem benötigt, welcher über Profibus mit dem Realtime PC verbunden wird. Um die Kommunikation über diese Verbindung zu realisieren muss der Realtime PC eine LabVIEW fähige Profibus-Master Karte besitzen.

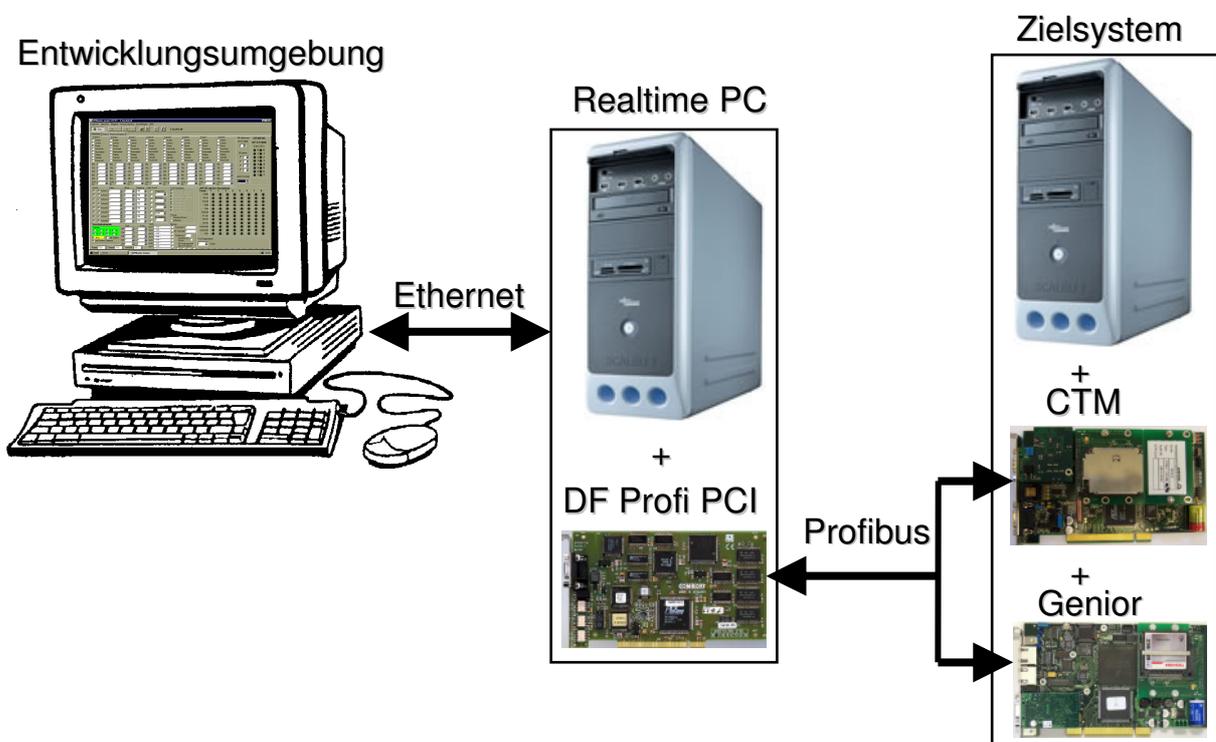


Abb. 6 Aufbau der Hardwareumgebung des Teststandes

²³ National Instruments

 Hochschule für Angewandte Wissenschaften Hamburg Hamburg University of Applied Sciences	Diplomarbeit Fachbereich Elektrotechnik und Informatik	
Wintersemester 07/08 Abgabetermin: 18.12.07	Givian Müller Matr.Nr.: 1749822	Betreuer: Professor Dr. Ing. Müller

3.2.1 Hardwareanforderungen der Entwicklungsumgebung

Die Hardwarevoraussetzungen der Entwicklungsumgebung sind nicht explizit vorgegeben. Daher wurde ursprünglich ein Desktop PC mit einem 1,4 GHz²⁴ Prozessor, 512 MB²⁵ Arbeitsspeicher und Windows 2000 vorgesehen. Im Laufe des Projektes hat sich jedoch gezeigt, dass LabVIEW beim Betreiben von mehreren VIs²⁶ nicht vollständig lauffähig war. Daraufhin wurde ein neuer Desktop PC mit Intel Core 2 Duo Prozessor mit 2 GHz, 1 GB²⁷ Arbeitsspeicher und Windows XP²⁸ angeschafft. Daraufhin war das Betreiben mehrerer VIs ohne Probleme möglich. Die weiteren Voraussetzungen der Hardware beschränken sich auf einen Ethernet Anschluss, um die Verbindung zum Realtime PC herzustellen.

3.2.2 Hardwareanforderungen des Realtime PCs

Für den Realtime PC gelten stärkere Einschränkungen als für die Programmierumgebung. Die Firma NI gibt genaue Hardwarekomponenten vor, damit die Echtzeitapplikation unter LabVIEW reibungslos läuft. Um dieses zu testen, gibt es von NI eine System Check Software. Der PC wird mit dieser Software von Diskette gebootet und führt ein Diagnoseprogramm aus. Dabei werden folgende Komponenten im System gesucht:

- Intel Pentium 4 Prozessor
- Intel Ethernet Karte mit Chip Typ 82540, 82559, 82558, oder 82550
- Parallele ATA²⁹ Festplatte
- FAT³⁰32 Treiber
- 3,5 Zoll Diskettenlaufwerk

Das entscheidende Kriterium der Untersuchung ist die Ethernet Karte, da alle anderen Komponenten in den meisten Standard PCs eingebaut sind. Der erforderliche Chip der Ethernet Karte ist aber nur sehr selten auf gängigen Karten zu finden. Als Alternative gibt es von der Firma NI eine eigene Karte „NI-PCI-8232 GPIB / Gigabit Ethernet“ die exakt für die Echtzeitanbindung ausgelegt ist. Um den reibungslosen Ablauf zu garantieren, wurde für dieses System eine solche Karte angeschafft.

²⁴ Gigaherz

²⁵ Mega Byte

²⁶ Virtuelle Instrumente

²⁷ Giga Byte

²⁸ eXPerience

²⁹ AT-Attachment / Name der IDE-Schnittstelle

³⁰ File Allocation Table

 Hochschule für Angewandte Wissenschaften Hamburg Hamburg University of Applied Sciences	Diplomarbeit Fachbereich Elektrotechnik und Informatik	
Wintersemester 07/08 Abgabetermin: 18.12.07	Givian Müller Matr.Nr.: 1749822	Betreuer: Professor Dr. Ing. Müller

```

CPU: GenuineIntel Pentium II/II Family
PCI BIOS found. Version: 2.10, Hardware Mechanism: 1
Total Memory: 127.62 Mb
Starting up File System... successful.
Available Logical drives:
A: (Removable) C: 7797 Mb, 2233 Mb free
D: 1976 Mb, 714 Mb free File I/O test: successful

Loading Ethernet Drivers:
Intel 8255x Ethernet initialized successfully.
Intel 8254x Ethernet initialized successfully.
Broadcom 57xx Ethernet initialized successfully.
Finding devices for Ethernet drivers:
Intel 8254x Ethernet Driver...found 1 device(s).
Starting up IP Stack... successful.
Initializing network...
Device 1 - MAC Address: 00:0E:0C:5A:E2:A3 - 10.0.60.157
(primary)

System state: Tested configuration is compatible with LabVIEW
Real-Time

```

Abb. 7 Realtime System Check Software

Zusätzlich benötigt der Realtime PC eine Profibusanbindung an das Zielsystem. Dafür wurde eine Profibuskarte „DF-Profi2-PCI“ der Firma Comsoft angeschafft. Diese Karte ist so konfigurierbar, dass sie sowohl als Master, wie auch als Slave betrieben werden kann. Im Handel ist derzeit keine weitere Karte erhältlich, die derartig konfigurierbar ist. Für diesen Teststand braucht die Karte zwar nur als Master betrieben zu werden, aber in Hinsicht auf weitere Projekte in der Firma, wie die Erfassung von Prozessdaten, wurde auf diese Möglichkeit Wert gelegt. Mittlerweile ist diese Karte auch über die Firma NI zu beziehen und die Einbindung fester Bestandteil in der Programmbibliothek der LabVIEW Software. Siehe Anhang I Datenblatt DF-Profi2-PCI.

3.2.3 Hardwareanforderungen des Zielsystems

Die Hardwareanforderungen des Zielsystems hängen von den beiden PCI Karten CTM und Genior ab. Die Anforderungen an die CTM Karte sind wie folgt beschrieben:

- PC mit Pentium Prozessor ab 133 Megahertz
- Betriebssystem Windows 95 oder höher
- 16 MB Arbeitsspeicher
- 10 MB freier Festplattenspeicher

Die gemeinsamen minimalen Anforderungen werden jedoch durch die leistungsstärkere Genior Karte definiert.

- PC mit Pentium 3 Prozessor ab 700 Megahertz
- Betriebssystem Windows XP oder höher
- 256 MB Arbeitsspeicher
- 100 MB freier Festplattenspeicher

Innerhalb des Zielsystems werden keine weiteren Profibuskarten benötigt, da beide Karten über einen derartigen Anschluss verfügen um Kontakt zum Realtime PC herzustellen.

 Hochschule für Angewandte Wissenschaften Hamburg Hamburg University of Applied Sciences	Diplomarbeit Fachbereich Elektrotechnik und Informatik	
Wintersemester 07/08 Abgabetermin: 18.12.07	Givian Müller Matr.Nr.: 1749822	Betreuer: Professor Dr. Ing. Müller

4 Konfiguration der Systeme

4.1 Hardwarekonfiguration

4.1.1 Einrichtung der Entwicklungsumgebung

Bei der Installation von LabVIEW ist zu beachten, dass das Service Pack 2 für Windows XP installiert ist, sowie Virenprogramme ausgeschaltet sind, da diese Konflikte mit LabVIEW erzeugen können. Um mögliche Probleme mit anderen Programmen zu vermeiden, wurde auf dem PC außer LabVIEW kein weiteres Programm installiert. Nach der Installation von LabVIEW 8.5 wurde LabVIEW RT 8.5 installiert. Bei der Realtimesoftware ist darauf zu achten, dass es sich um LabVIEW Realtime RX³¹ handelt, da nur diese Version zum Betreiben einer separaten Echtzeitumgebung geeignet ist. Mit der Standardversion kann lediglich ein echtzeitfähiger Programmkern auf dem Programmierrechner eingerichtet werden, der allerdings keine vergleichbaren Laufzeiten erreicht.

Nachdem die richtige Version installiert ist, wird der MAX³² der Firma NI gestartet, um mit seiner Hilfe eine „RT-Boot Disk“ zu erstellen. Sollten auf dem als Realtime PC genutzten System noch ein Betriebssystem oder andere Daten installiert sein, kann auch eine „Format-Boot Disk“ erstellt werden, die vor dem Hochlaufen des PCs alle vorhandenen Komponenten löscht.

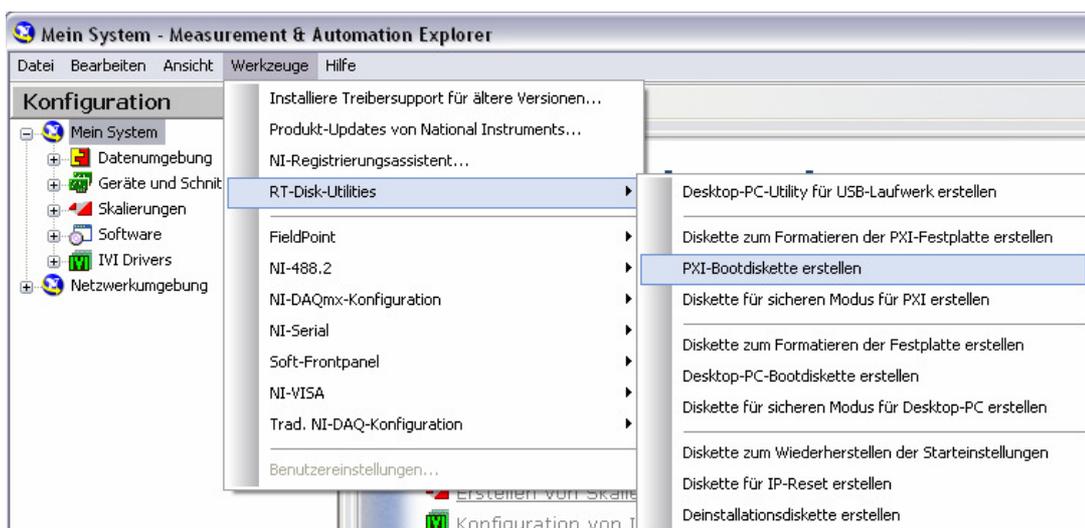


Abb. 8 Realtime Bootdiskette erstellen

³¹ Remote Procedure Call

³² Measurement & Automation Explorer

 Hochschule für Angewandte Wissenschaften Hamburg Hamburg University of Applied Sciences	Diplomarbeit Fachbereich Elektrotechnik und Informatik	
Wintersemester 07/08 Abgabetermin: 18.12.07	Givian Müller Matr.Nr.: 1749822	Betreuer: Professor Dr. Ing. Müller

4.1.2 Einrichtung der Echtzeitumgebung

Um eine Verbindung vom Programmierrechner zum Realtime PC dem so genannten Target, zu erhalten, muss dieser zunächst mit der eingelegten Boot Disk gestartet werden. Ist der Rechner erfolgreich angelaufen, erscheint auf dessen Monitor die CPU³³ Auslastung des Realtime Systems, die MAC³⁴ Adresse des Rechners, sowie die Versionsnummern vom „LabVIEW Realtime 8.5“ und des „NI-VISA³⁵ Servers 4.2“.

4.1.3 Einrichtung des Zielsystems

Die Einrichtung und Konfiguration der Karten CTM bzw.³⁶ Genior wird hier nicht detailliert erläutert, da dieses kein Bestandteil der eigentlichen Diplomarbeit darstellt. Einzig relevant ist, dass die Karten als Slave konfiguriert werden und die Profibusadressen entsprechend der Profibuskonfiguration einzustellen sind. Anschließend kommunizieren die Karten, sobald sie über ihre Profibusadresse angesprochen werden, selbstständig mit der zu entwickelnden Software.

4.2 Softwarekonfiguration

4.2.1 Verbindungsaufbau zwischen Entwicklungs- und Echtzeitumgebung

Auf der Entwicklungsumgebung wird nach dem Hochlaufen des Targets die Verbindung über Ethernet entweder automatisch erkannt oder muss über die „Windows- Netzwerkumgebung“ eingerichtet werden.

Anschließend wird im MAX mit Doppelklick auf Netzwerkumgebung nach neuen Komponenten gesucht und der Target gefunden. Sollte dieses nicht automatisch passieren, kann auch über „rechte Maustaste“ → „Suchen“ der Name des Zielsystems oder optional die IP Adresse eingetragen werden, um nach neuen Komponenten zu suchen.

³³ Central Processing Unit / Hauptprozessor

³⁴ Media Access Control

³⁵ Virtual Installation Solution Architecture

³⁶ Beziehungsweise

 Hochschule für Angewandte Wissenschaften Hamburg Hamburg University of Applied Sciences	Diplomarbeit Fachbereich Elektrotechnik und Informatik	
Wintersemester 07/08 Abgabetermin: 18.12.07	Givian Müller Matr.Nr.: 1749822	Betreuer: Professor Dr. Ing. Müller

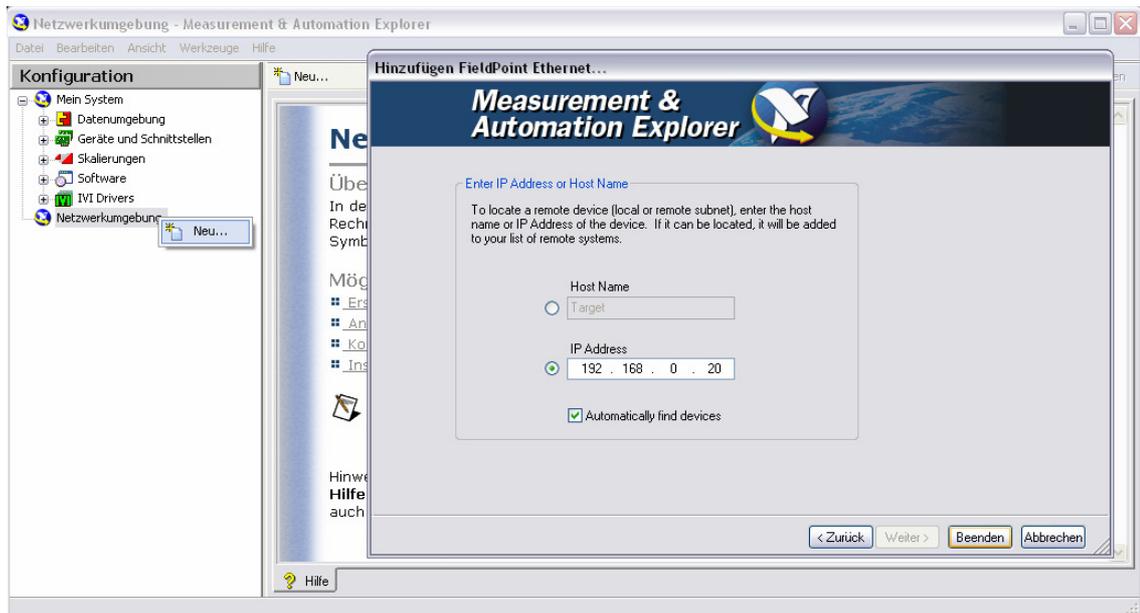


Abb. 9 Netzwerkkomponenten hinzufügen

Nachdem der Target als Zielsystem erkannt wurde enthält er folgende Komponenten:

- Datenumgebung
- Geräte und Schnittstellen
- Skalierungen
- Software

Als nächstes wird über Rechtsklick auf Software → „Software hinzufügen/entfernen“ die Software der LabVIEW Realtimeumgebung auf den Target geladen.

Um den vollen Funktionsumfang zu gewährleisten, werden alle vorhandenen Komponenten übertragen. Die Echtzeitumgebung kann nun über die Programmoberfläche von LabVIEW verwendet werden.

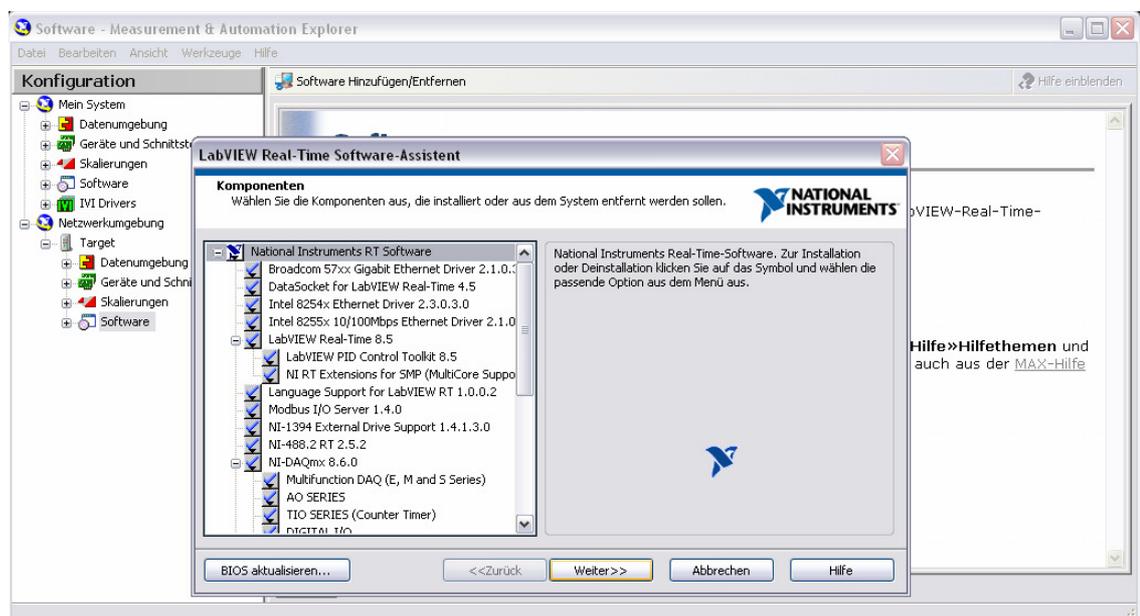


Abb. 10 Software auf die Echtzeitumgebung laden

4.2.2 Einrichtung der Profibusverbindung

Damit die Profibuskarte der Firma Comsoft betrieben werden kann, muss zunächst die mitgelieferte Software auf der Entwicklungsumgebung installiert werden. Nach der Installation befinden sich die entsprechenden Kommunikationsbausteine zum Einbinden in die Programmierung in der LabVIEW Bibliothek. Um diese Bausteine verwenden zu können, muss der Profibus, entsprechend den angeschlossenen Karten, konfiguriert werden. Dafür werden die GSD³⁷ Dateien der Genior und CTM Karte benötigt. Diese müssen in das Stammverzeichnis des Configurators kopiert werden, welches bei Standardinstallation „C:\Dokumente und Einstellungen\All Users\Anwendungsdaten\Configurator2\GSD“ lautet. Anschließend können diese über den Configurator eingebunden werden. Mit Hilfe der GSD Dateien wird automatisch die Anzahl der In- und Output- Bytes des Bussignals vom Slave festgelegt und auf dem Profibus konfiguriert. Lediglich die auf den Karten eingestellten Profibusadressen müssen manuell eingestellt werden.

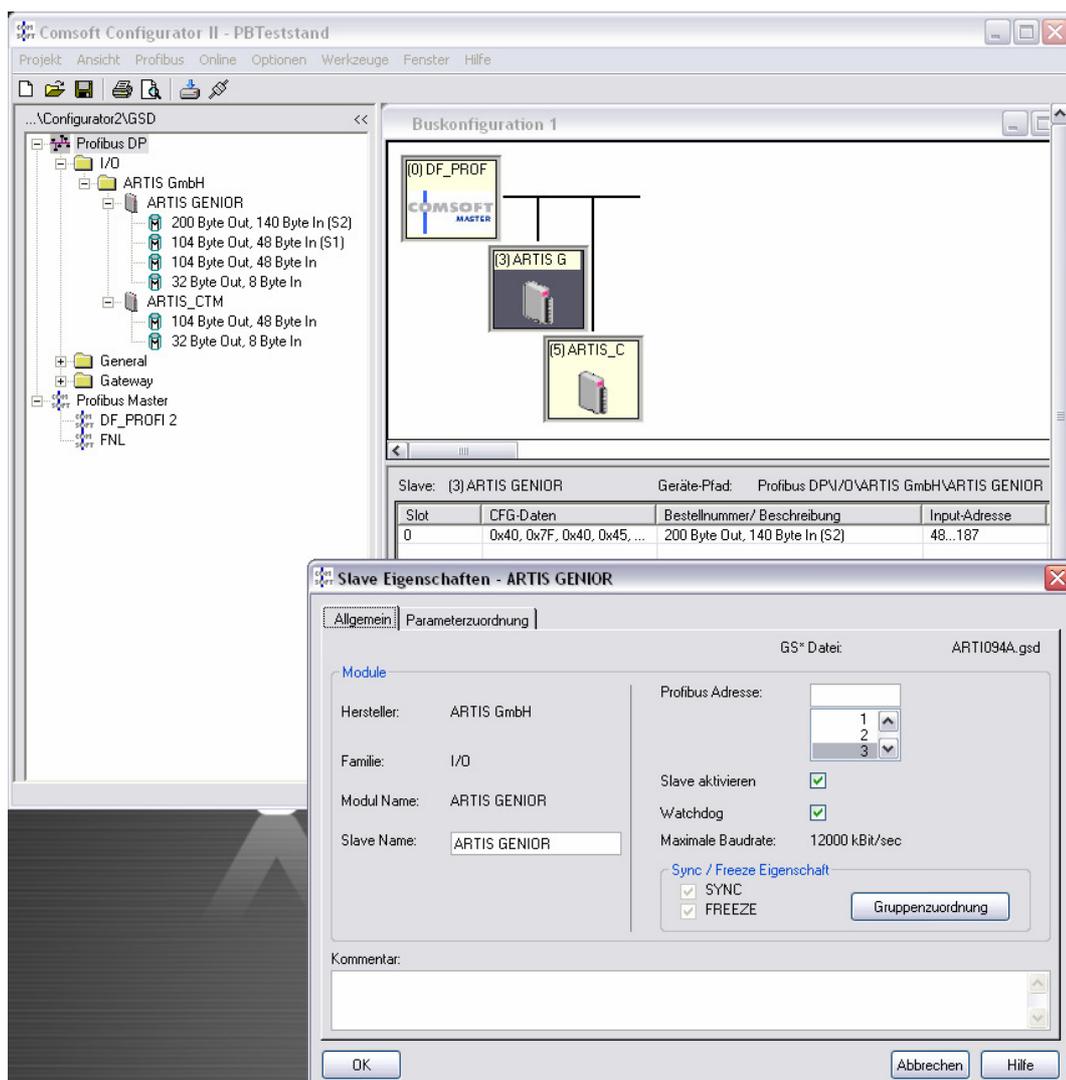


Abb. 11 Einrichtung der Profibusverbindung

³⁷ General Station Description / Gerätstammdaten

 Hochschule für Angewandte Wissenschaften Hamburg Hamburg University of Applied Sciences	Diplomarbeit Fachbereich Elektrotechnik und Informatik	
Wintersemester 07/08 Abgabetermin: 18.12.07	Givian Müller Matr.Nr.: 1749822	Betreuer: Professor Dr. Ing. Müller

4.2.3 Kontakt zur Profibuskarte

Ebenso wie die Programmierumgebung benötigt auch die Echtzeitumgebung Treiber für die Profibuskarte. Diese werden über ein weiteres von der Firma Comsoft mitgeliefertes „DF Profi 2 Profibus Configuration Download“ Programm auf das Echtzeitsystem übertragen. Leider arbeitet dieses Programm in der ausgelieferten Version nicht einwandfrei, so dass die benötigten Komponenten mit Hilfe des MAX, wie unter Kapitel 4.2.1 beschrieben, über „Software Hinzufügen/Entfernen“ manuell übertragen werden müssen.

Diese Fehlfunktion war der Firma Comsoft zu diesem Zeitpunkt bereits bekannt, so dass das Problem mit telefonischer Unterstützung des Servicetechnikers beseitigt werden konnte.



Abb. 12 Profibus Configuration Download

Nachdem die Software auf den Target geladen wurde, kann diese mit dem „DF PROFI 2 – RT Installation“ Programm installiert werden. Mit diesem Programm wird über die IP Adresse auf den Target zugegriffen um dort die Installation zu starten.



Abb. 13 DF PROFI 2 - RT Installation

4.2.3.1 Testen der Profibusverbindung

Wenn alle Treiber installiert sind, und die Verbindung zu den Karten eingerichtet ist, sollte die reibungslose Kommunikation überprüft werden. Dafür wird erneut der „Profibus Configurator“ gestartet und die gespeicherte Konfiguration geladen. Über den in der Menüleiste liegenden Button „Download“ werden die Daten auf den Target geladen. Anschließend wird über „Monitor / Verändern verbinden“ ein Systemcheck durchgeführt.

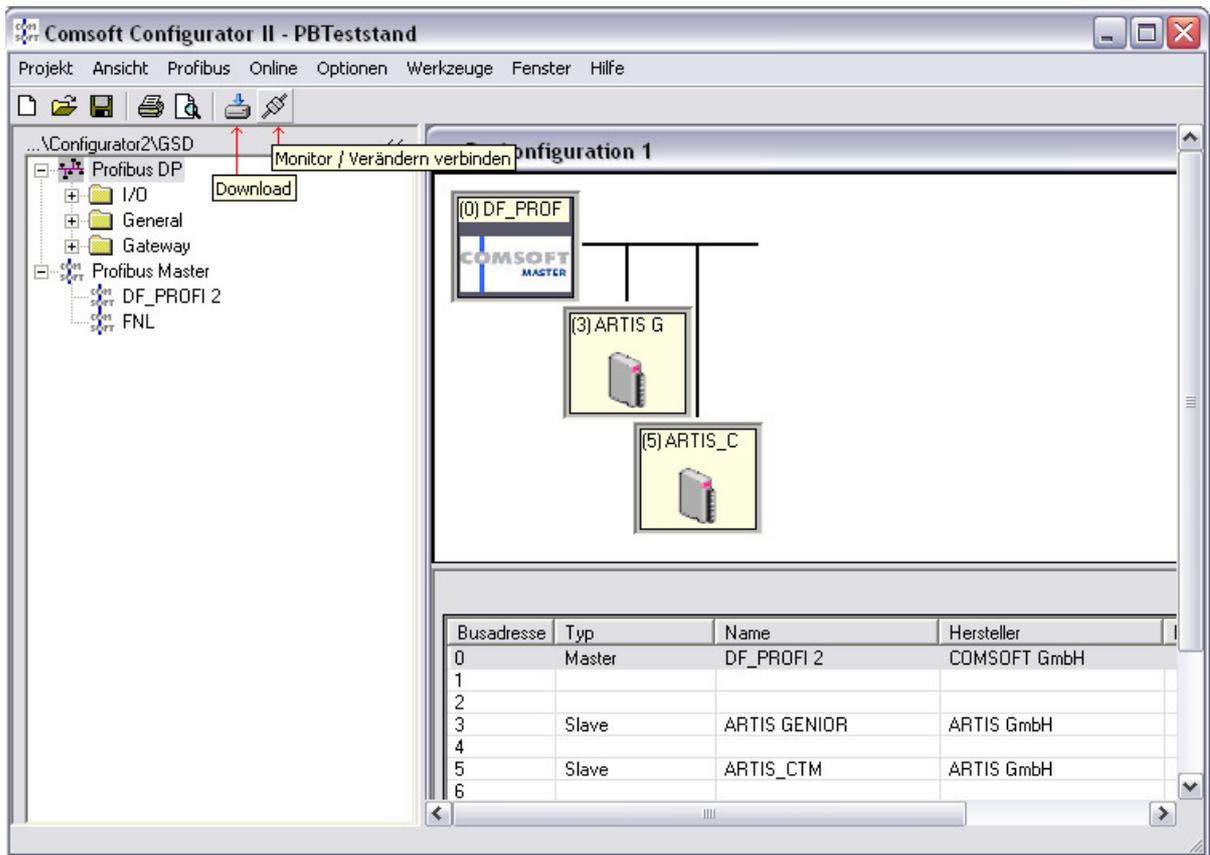


Abb. 14 Testen der Profibusverbindung

Beim Systemcheck sollten beide Module „ARTIS Genior“ und „ARTIS CTM“ mit dem Configurator kommunizieren. Funktionstüchtige Profibusteilnehmer werden mit einem grünen Rahmen dargestellt, während inaktive oder nicht erreichbare Teilnehmer einen roten Rahmen erhalten. Im Falle einer roten Umrandung muss die entsprechende Kartenkonfiguration überprüft werden. Wenn beide Teilnehmer funktionstüchtig sind, ist die Einrichtung erfolgreich beendet. Es kann mit der Programmierung der Software begonnen werden.

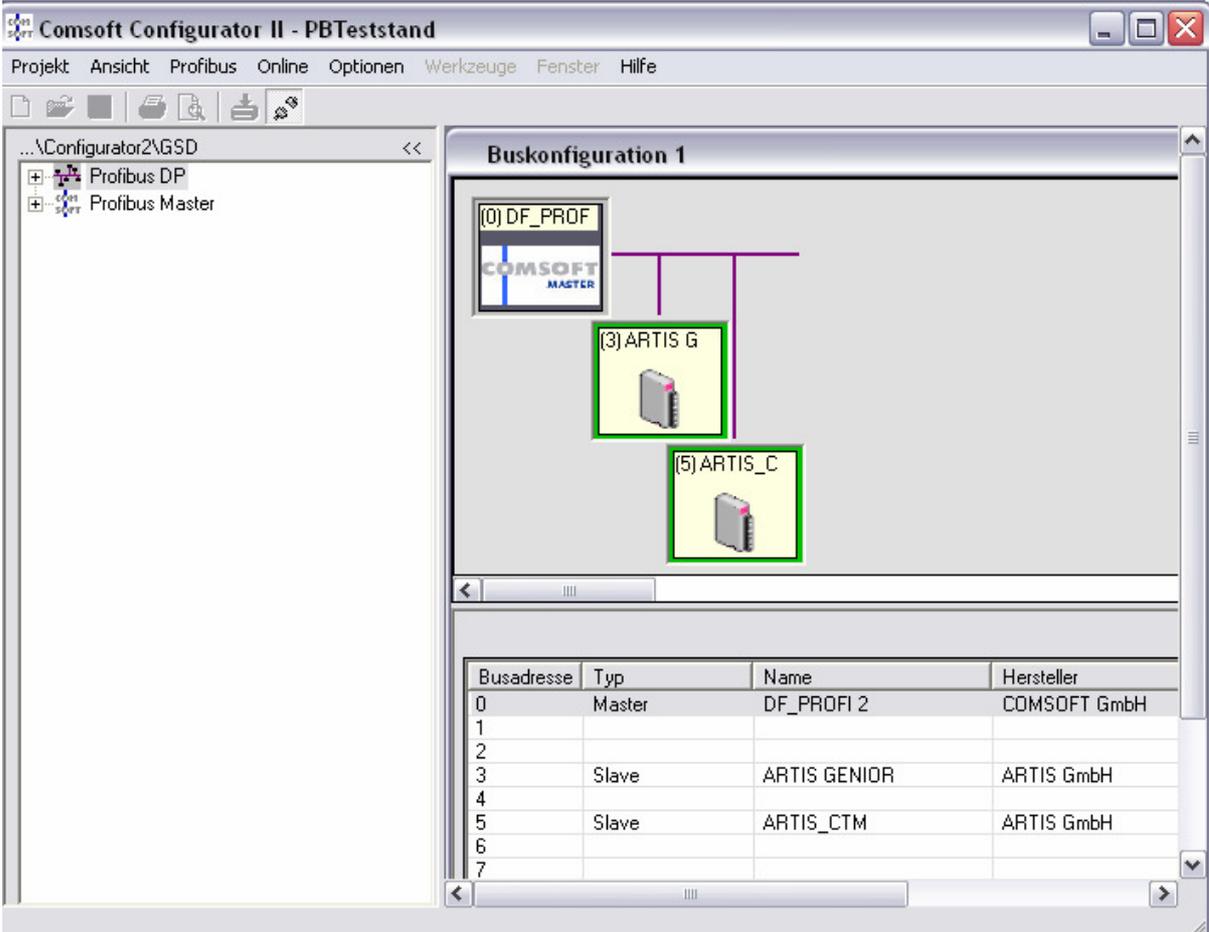


Abb. 15 Verbindung zum Zielsystem hergestellt

 Hochschule für Angewandte Wissenschaften Hamburg <i>Hamburg University of Applied Sciences</i>	Diplomarbeit Fachbereich Elektrotechnik und Informatik	
Wintersemester 07/08 Abgabetermin: 18.12.07	Givian Müller Matr.Nr.: 1749822	Betreuer: Professor Dr. Ing. Müller

5 Implementierung der Software

5.1 Projekte unter LabVIEW

Seit der LabVIEW Version 8.x bietet diese Software die Möglichkeit eine zusammenhängende Ordnerstruktur mit mehreren programmierten Modulen zu erstellen. Das dient zum einen zum Verbessern der Übersicht über die entwickelten Softwaremodule, zum anderen erleichtert es die Kommunikation zwischen den Modulen über globale Kommunikationsvariablen.

5.2 Erstellen eines Projektes

Um ein Projekt unter LabVIEW zu erstellen, muss beim ersten Öffnen des Programms zunächst entschieden werden, welche Art von Programm erstellt werden soll. Für den echtzeitfähigen Teststand wird die Auswahlfläche „Real- Time Project“ gewählt.

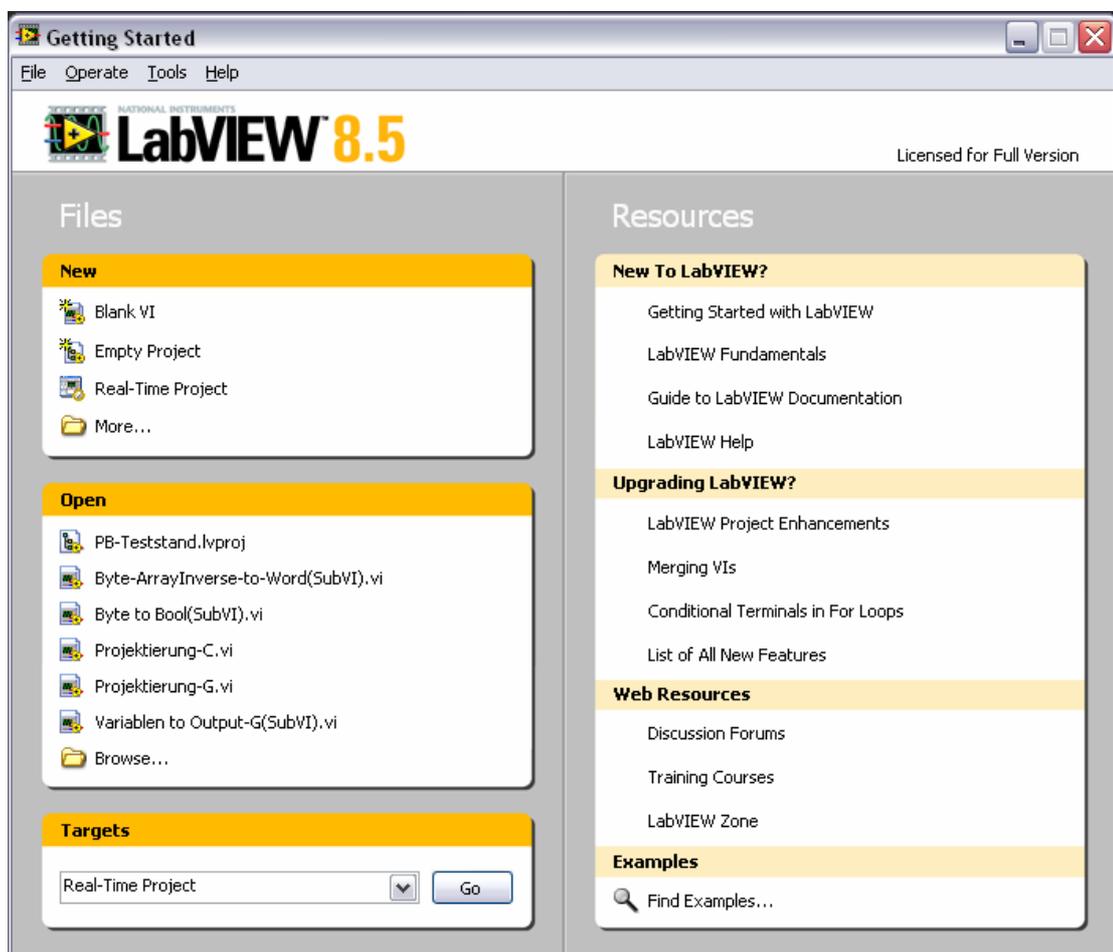


Abb. 16 LabVIEW Projekt erstellen

 Hochschule für Angewandte Wissenschaften Hamburg Hamburg University of Applied Sciences	Diplomarbeit Fachbereich Elektrotechnik und Informatik	
Wintersemester 07/08 Abgabetermin: 18.12.07	Givian Müller Matr.Nr.: 1749822	Betreuer: Professor Dr. Ing. Müller

Anschließend wird der Typ der Echtzeitumgebung eingestellt. Dafür gibt es drei Auswahlmöglichkeiten. „Continuous communication architecture“, „State machine architecture“ und „Custom project“.

Für die Echtzeitumgebung auf einem externen Desktop PC wird „Continuous communication architecture“ ausgewählt. Dieses Modul ermöglicht es, mehrere Applikationen gleichzeitig auszuführen, wobei jeder Applikation eine unterschiedliche Priorität zugeteilt wird. Darüber hinaus kommuniziert die Echtzeitumgebung kontinuierlich mit der Programmierumgebung.

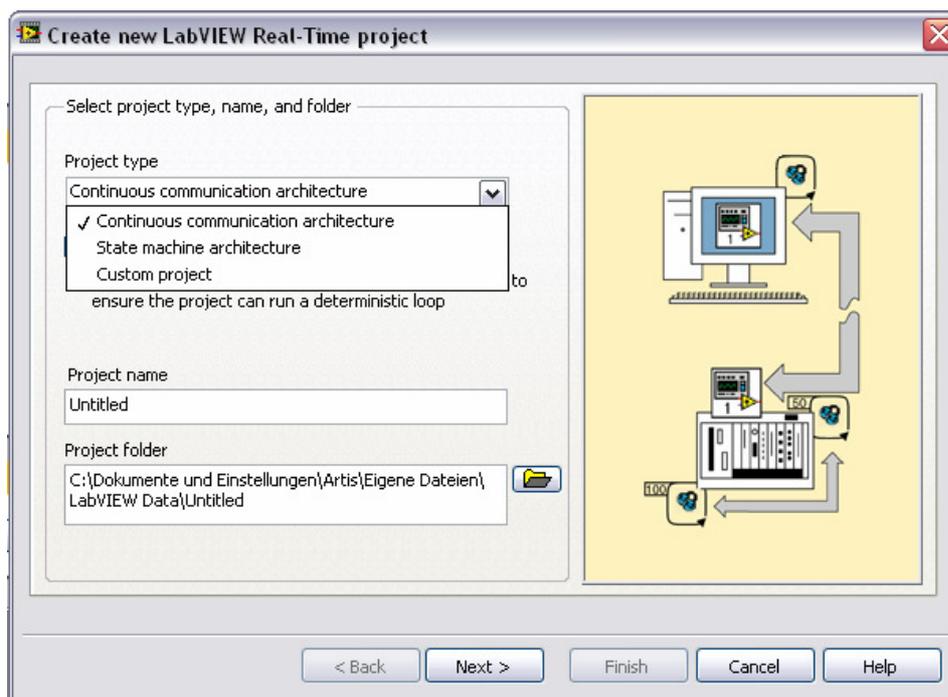


Abb. 17 LabVIEW Projekttyp wählen

Nachdem der Projektname, der für dieses Projekt „PB-Teststand“ lautet, eingestellt ist und mit Finish bestätigt wurde, öffnet sich der Projekt Explorer mit dem neu erstellten Projekt „PB-Teststand“. In der Übersicht des Explorers ist zum einen der Programmier PC unter „My Computer“ sowie der Echtzeitrechner unter „Target (MAC Adresse)“ zu finden. In beiden Bereichen ist es möglich, Programme, Ordner und Variablen zu erstellen.

5.2.1 Struktur im Projekt Explorer

Um mögliche Fehlerquellen, wie beispielsweise falsch zugewiesene Variablen, zu vermeiden, muss eine gegliederte Ordnerstruktur erstellt werden. Auf dem Programmierrechner besitzen beide Kartentypen je einen eigenen Ordner mit ihrem Namen, in denen sich die kanalabhängigen Bedienmodule sowie die Unterordner Daten und SubVIs³⁸ befinden.

Der Datenordner enthält alle Variablenlisten und Unterprogramme, die für die Übertragung zuständig sind. Die Variablenlisten sind zusätzlich nach Typen sortiert. So haben die

³⁸ Untergeordnetes Virtuelles Instrument / Unterprogramm

 Hochschule für Angewandte Wissenschaften Hamburg Hamburg University of Applied Sciences	Diplomarbeit Fachbereich Elektrotechnik und Informatik	
Wintersemester 07/08 Abgabetermin: 18.12.07	Givian Müller Matr.Nr.: 1749822	Betreuer: Professor Dr. Ing. Müller

Variablen von der Maschine zur Karte und die Variablen von der Karte zur Maschine je eine eigene Bibliothek. Darüber hinaus sind diese Bibliotheken in Kanäle unterteilt.

Der Ordner SubVI enthält alle kanalabhängigen Unterprogramme, die in den globalen Bedienmodulen verwendet werden.

Zusätzlich befinden sich auf dem Programmierrechner die kanalunabhängigen Module „Projektierung“ und „Auswertung“, sowie ein weiterer Ordner „SubVI“ in dem sich kanalunabhängige Unterprogramme befinden.

Auf dem Target befinden sich die Module „Target-C“ (für CTM) und „Target-G“ (für Genior), sowie ein Ordner „SubVI“ in dem die Unterprogramme zur Umwandlung der Datenformate zu finden sind.

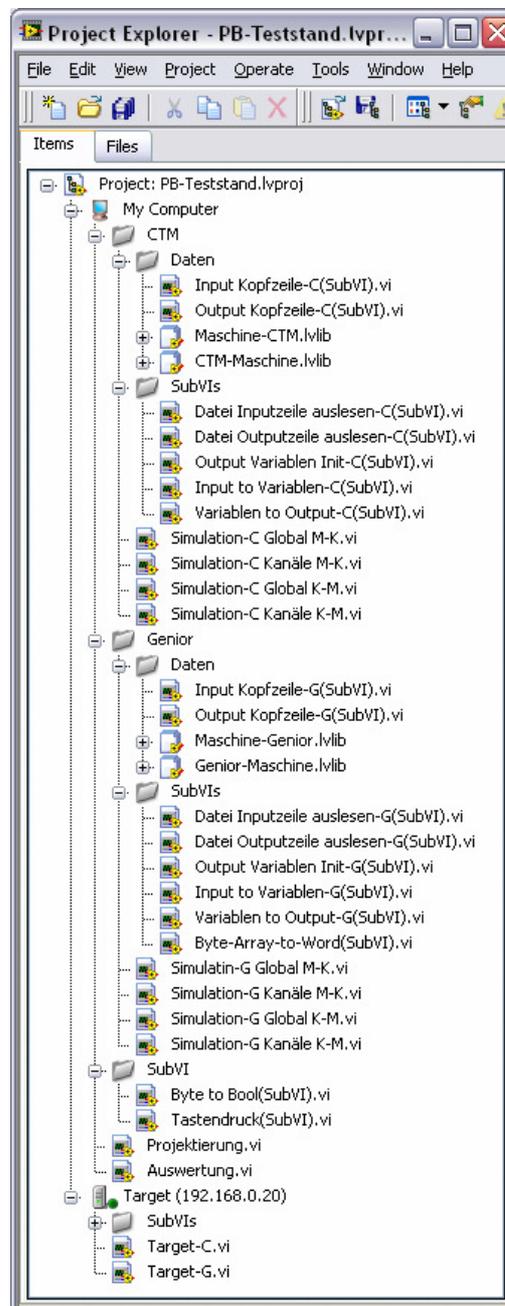


Abb. 18 Struktur im Projekt Explorer

 Hochschule für Angewandte Wissenschaften Hamburg Hamburg University of Applied Sciences	Diplomarbeit Fachbereich Elektrotechnik und Informatik	
Wintersemester 07/08 Abgabetermin: 18.12.07	Givian Müller Matr.Nr.: 1749822	Betreuer: Professor Dr. Ing. Müller

5.2.2 Grafische Modulgestaltung

Die LabVIEW Software ermöglicht es die grafische Darstellung der Programmbausteine eines jeden Moduls zu verändern. Allerdings besitzt dieses Feature nur sehr eingeschränkte Funktionen. In diesem Zusammenhang ist im Support Bereich auf der Homepage der Firma NI ein neues Modul namens „Icon Editor“ verfügbar, welches einen wesentlich erweiterten Funktionsumfang aufweist. Nachdem das neue Modul installiert ist, wird beim Bearbeiten der Grafiken anstatt des alten Editors automatisch der neue gestartet.

Mit diesem neuen Feature wurden sämtliche Grafiken der Module ihrer entsprechenden Funktion nach angepasst.

Um innerhalb des Programmcodes alle eigenen verwendeten Module besser unterscheiden zu können, wurden diese farblich getrennt in drei Untergruppen unterteilt. Alle Module die kartenunabhängig arbeiten wurden mit einem grünen, alle Module der CTM Karte mit einem orangen und alle Module der Genior Karte mit einem blauen Header versehen.

Darüber hinaus sind alle Module die eine bestimmte Funktionalität erfüllen mit grafischen Darstellungen entsprechend ihrer Funktionalität ergänzt worden. Module die keiner Grafik zuzuordnen sind, enthalten einen kurzen Text der ihre Funktion beschreibt.

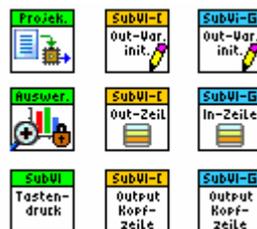


Abb. 19 Beispiele der Modulgestaltung

5.3 Kommunikationsvariablen

Die einzige Kommunikation zwischen dem laufenden Programm auf der Programmierumgebung und dem Target erfolgt über Variablen. Diese können von beiden Systemen gelesen und geschrieben werden. Um sicherzustellen, dass jedes Signal einzeln gelesen und geschrieben werden kann, wird für jedes Signal von der Karte zur Maschine und von der Maschine zur Karte eine Variable erstellt. Durch die unterschiedliche Buskonfiguration von CTM und Genior entstehen folgende Variablen:

87 Variablen von der Maschine zur Karte für CTM

64 Variablen von der Karte zur Maschine für CTM

142 Variablen von der Maschine zur Karte für Genior

138 Variablen von der Karte zur Maschine für Genior

Siehe Anhang III Profibusadressstabellen.

 Hochschule für Angewandte Wissenschaften Hamburg Hamburg University of Applied Sciences	Diplomarbeit Fachbereich Elektrotechnik und Informatik	
Wintersemester 07/08 Abgabetermin: 18.12.07	Givian Müller Matr.Nr.: 1749822	Betreuer: Professor Dr. Ing. Müller

5.3.1 Eigenschaften von Kommunikationsvariablen

Variablen können in verschiedenen Formaten erstellt werden. Zur Verfügung stehen:

- Double (Gleitkommazahl 64 Bit)
- String (endliche Zeichenfolge)
- Boolean (Binäre Einheit „True“ oder „False“)
- Int8 (8 Bit vorzeichenbehaftete Integerzahl [-128 bis 127])
- UInt8 (8 Bit vorzeichenlose Integerzahl [0 bis 255])
- Int16 (16 Bit vorzeichenbehaftete Integerzahl [-32.768 bis 32.767])
- UInt16 (16 Bit vorzeichenlose Integerzahl [0 bis 65.535])
- Int32 (32 Bit vorzeichenbehaftete Integerzahl [-2.147.483.648 bis 2.147.483.647])
- UInt32 (32 Bit vorzeichenlose Integerzahl [0 bis 4.294.967.295])
- Int64 (64 Bit vorzeichenbehaftete Integerzahl [-1e¹⁹ bis 1e¹⁹])
- UInt 64 (64 Bit vorzeichenlose Integerzahl [0 bis 2e¹⁹])

Darüber hinaus kann jeder Variablentyp auch als Variablenarray eingerichtet werden. Da die Signale von CTM und Genior aus verschiedenen Typen unterschiedlicher Länge bestehen und veränderbar über die Laufzeit sein müssen, wurde jede Variable entsprechend ihres Typs als Variablenarray eingerichtet. Daraus ergaben sich folgende Konfigurationen:

Bit	Boolean
Byte	UInt8
Word	UInt16
Doppel Word (DWord)	Int32

Die Achswerte sind die einzigen Signale, die im DWord Format verschickt werden. Da diese als einzige negative Werte annehmen können, wurden für sie vorzeichenlose Variablen verwendet.

Um die Laufzeit der Messungen variabel zu gestalten, wird die Größe des Arrays nicht explizit vorgegeben. Jeder Index der Arrays der einzelnen Variablen entspricht einem neuen zu übertragenen Wert.

Ursprünglich war für beide Systeme (CTM und Genior) die Nutzung gemeinsamer globaler Variablen vorgesehen. Davon wurde aber abgesehen, da die parallele Betreibung mehrerer Karten hierdurch nicht möglich wäre. Zu den globalen und kanalabhängigen Variablen beider Karten kommen noch je zehn programmrelevante Kommunikationsvariablen.

- a. Index (UInt 64)
Der „Index“ gibt während der Laufzeit ständig die aktuelle Zeile der übertragenen Arrays an.
- b. Wertemenge (UInt 64)
Die „Wertemenge“ gibt die Länge der zu übertragenen Arrays an.
- c. Inputdatei (String)
In der Variablen „Inputdatei“ wird der Dateipfad gespeichert aus dem die Variablenarray ihre Werte erhalten haben.
- d. Start (Boolean)
Die Variable „Start“ signalisiert das Starten einer neuen Messung.

 Hochschule für Angewandte Wissenschaften Hamburg Hamburg University of Applied Sciences	Diplomarbeit Fachbereich Elektrotechnik und Informatik	
Wintersemester 07/08 Abgabetermin: 18.12.07	Givian Müller Matr.Nr.: 1749822	Betreuer: Professor Dr. Ing. Müller

- e. Taktung (UInt 32)
Die Variable „Taktung“ überträgt den Zyklus Takt des Compilers zum Senden des Profibussignals des Echtzeitsystems.
- f. Leserate (UInt 16)
Die „Leserate“ beschreibt das Vielfache der „Taktung“ zum Empfangen des Profibussignals auf dem Echtzeitsystem.
- g. Ladeindex (UInt 32)
Der „Ladeindex“ verweist auf den aktuellen Index der Initialisierungsroutine auf dem Echtzeitsystem.
- h. Neue Messung (Boolean)
Die Variable „Neue Messung“ signalisiert das Ende einer Messung.
- i. Messanzahl (UInt 64)
„Messanzahl“ übergibt die noch durchzuführende Anzahl der kontinuierlich nacheinander durchzuführenden Messungen.
- j. Input Time (Array of UInt 64)
„Input Time“ enthält ein Array aus Zeitwerten mit einem vorher eingestellten Takt beginnend bei 0 ms. Die Länge des „Input Time“ Arrays entspricht immer der „Wertemenge“.

5.3.1.1 Problematik bei zu großer Datenmenge

Ursprünglich war es vorgesehen jeder Variable anstatt ein ganzes Array von Werten nur einen Wert zuzuteilen. Über die Laufzeit sollten dann die Werte in 10 ms Schritten verändert werden. Dadurch wäre es nötig alle Variablen in 10 ms Schritten abzufragen und zu speichern, was die volle Auslastung des Echtzeitsystems zur Folge hätte.

Durch das Anlegen eines Arrays für jede Variable können die gesamten Werte vor der Messung eingelesen werden und erst nach der Messung in einem Block gespeichert werden. Dadurch sinkt die Auslastung des Echtzeitsystems unter die 100 % Auslastungsgrenze. Eine weiterführende Problembeschreibung zur Gewährleistung der Echtzeit befindet sich im Kapitel 6.2.4.

 Hochschule für Angewandte Wissenschaften Hamburg Hamburg University of Applied Sciences	Diplomarbeit Fachbereich Elektrotechnik und Informatik	
Wintersemester 07/08 Abgabetermin: 18.12.07	Givian Müller Matr.Nr.: 1749822	Betreuer: Professor Dr. Ing. Müller

6 Programmierung der Software

6.1 Module zum Umwandeln von Datenformaten

Nötig werden Konvertierungen der Signale durch die Verwendung der Profibuschnittstelle. Profibussignale werden ausschließlich im Byte Format versendet und empfangen. Das bedeutet sie entsprechen einem Wertebereich von 0 bis 255.

Dafür beinhaltet LabVIEW 8.5 eine Vielzahl von Modulen zum Umwandeln von Datenformaten. Verwendet wurden beispielsweise Bibliotheksbausteine zum Zusammensetzen von Bit Signalen in ein Byte. Allerdings gibt es bei diesem Projekt auch Konvertierungen in das Byte Format, die in der Bibliothek nicht vorgesehen sind.

Da die Signale auch höhere Werte als 255 annehmen können, müssen diese entsprechend auf mehrere Bytes aufgeteilt werden. Diese Aufteilung erfolgt für einen Wert im Format UInt16 in ein HB³⁹ und ein LB⁴⁰ oder für einen Wert im Format Int32 in ein HHB⁴¹, ein HLB⁴², ein LHB⁴³ und ein LLB⁴⁴. Damit der Nutzer der Software nicht selbst jeden gewünschten höheren Wert umrechnen muss, erledigt das die Software für ihn.

Ebenso ist im umgekehrten Fall nicht vorgesehen die Signale eines Bytes in seine Bits zu zerlegen, oder empfangene Werte, die aus mehreren Bytes bestehen, zu ihrem Wert zusammenzufügen.

³⁹ High Byte

⁴⁰ Low Byte

⁴¹ High High Byte

⁴² High Low Byte

⁴³ Low High Byte

⁴⁴ Low Low Byte

6.1.1 Modulbeschreibung Byte to Bool (SubVI)

Das Modul „Byte to Bool“ wandelt einen UInt8 Wert in ein Boolean Array und zerlegt dieses in seine 8 einzelnen Bits. Die Ausgänge nehmen dabei entweder den Wert „True“ oder „False“ an.

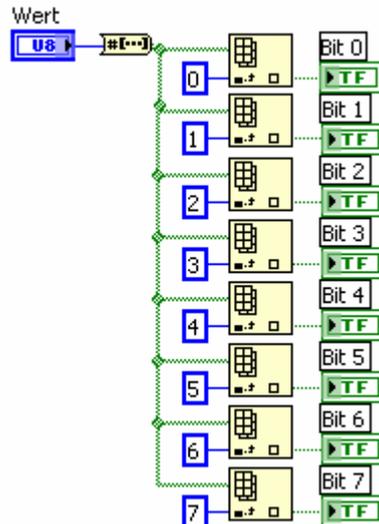


Abb. 20 Programmcode Byte to Bool (SubVI)

6.1.2 Modulbeschreibung Byte to Bit (SubVI)

Das Modul „Byte to Bit“ besitzt dieselbe Funktionalität wie „Byte to Bool“, gibt jedoch die Ausgabewerte im Format UInt8 aus. Die Ausgänge nehmen entweder den Wert 0 oder den Wert 1 an.

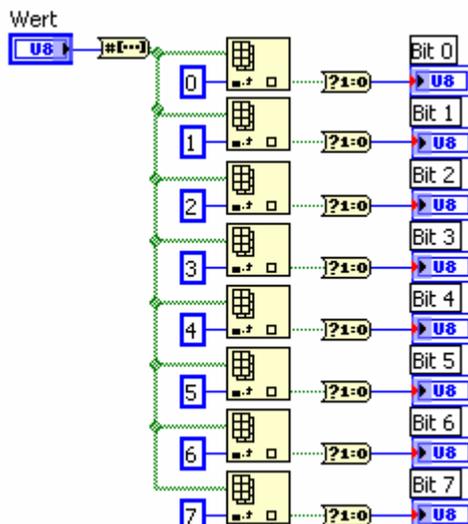


Abb. 21 Programmcode Byte to Bit (SubVI)

6.1.3 Modulbeschreibung Word to Byte-Array (SubVI)

Das Modul „Word to Byte Array“ zerlegt einen Wert im Format UInt16 in sein LB und sein HB im Format UInt8.

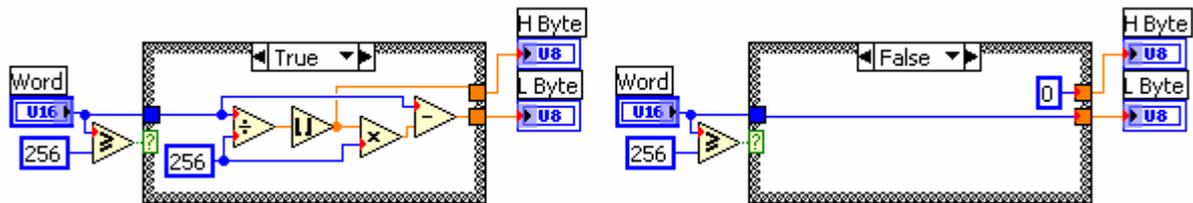


Abb. 22 Programmcode Word to Byte Array (SubVI)

6.1.4 Modulbeschreibung DWord to Byte Array (SubVI)

Das Modul „DWord to Byte Array“ zerlegt einen Wert im Format Int32 in seine vier Byte HHB, HLB, LHB und LLB im Format UInt8.

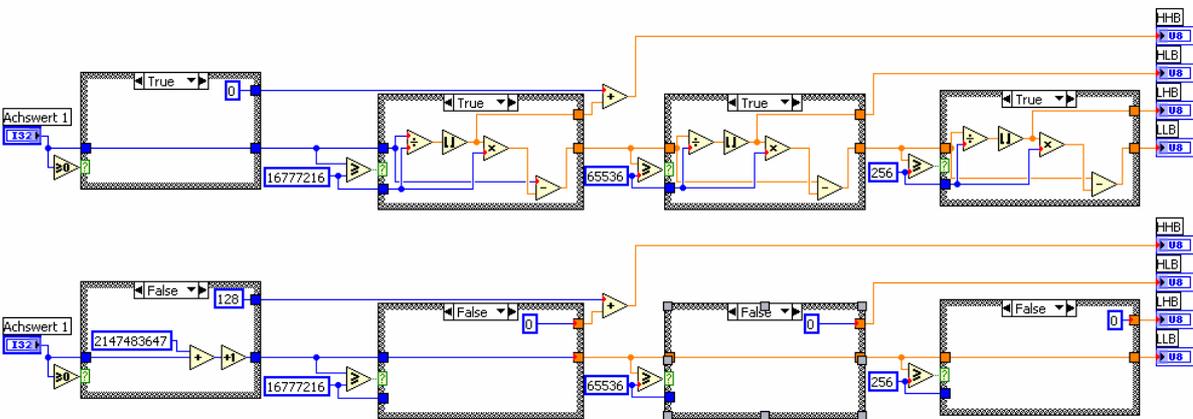


Abb. 23 Programmcode DWord to Byte Array (SubVI)

6.1.5 Modulbeschreibung Byte-Array to Word (SubVI)

Das Modul „Byte Array to Word“ bildet aus einem zusammen gehörenden HB und LB im Format UInt8 einen Wert im Format UInt16.

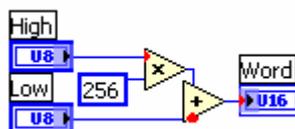


Abb. 24 Programmcode Byte Array to Word (SubVI)

 Hochschule für Angewandte Wissenschaften Hamburg Hamburg University of Applied Sciences	Diplomarbeit Fachbereich Elektrotechnik und Informatik	
Wintersemester 07/08 Abgabetermin: 18.12.07	Givian Müller Matr.Nr.: 1749822	Betreuer: Professor Dr. Ing. Müller

6.2 Softwaremodule der Echtzeitumgebung

Die echtzeitfähigen Module „Target-C“ für CTM und „Target-G“ für Genior besitzen grundlegend die gleiche Funktion. Der Unterschied liegt nur in der Adressliste der Buskonfiguration. Ansonsten ist sowohl die Funktionsweise, als auch die Programmstruktur identisch. Aus diesem Grund wird im folgenden Verlauf lediglich allgemein das Modul „Target“ beschrieben, ohne auf die einzelnen Kartentypen einzugehen.

Das Modul „Target“ verfügt über keinerlei Bedienoberflächen, da diese kontinuierlich abgefragt werden müssten, wodurch die Zykluszeit des Moduls erheblich beeinflusst werden würde.

Die Programmierung des Moduls ist so aufgebaut, dass theoretisch mehrere Module dieser Art parallel auf dem Echtzeitsystem lauffähig sind, um getrennt voneinander Messungen durchzuführen. Um dabei gewährleisten zu können, dass die Echtzeitanforderung eingehalten wird, muss jedes einzelne Modul und jede einzelne Schleifenstruktur eine unterschiedliche Priorität erhalten. Mehr Informationen über diesen Multiuserbetrieb befinden sich im Kapitel 10.2.6.

Ebenso wie mehrere Messung gleichzeitig möglich sind, können mit dem Modul „Target“ beliebig viele Messungen nacheinander durchgeführt werden. Jedoch bleiben zur Auswertung nur die empfangenen Signale der letzten Messung im Speicher der Variablenarrays (Karte zur Maschine) erhalten. Für eventuelle Langzeittests könnte das Modul derart erweitert werden, dass lediglich Änderungen der Signale dokumentiert und in einer Datei zur Auswertung abgelegt werden. Konkrete Pläne in diese Richtung liegen jedoch derzeit noch nicht vor.

Allgemein kann die Programmbeschreibung der Echtzeitumgebung in drei wesentliche Punkte unterteilt werden.

- Die Initialisierung zum Bilden der Signale
- Die Durchführung der Messung
- Die Speicherung der empfangenen Messsignale

6.2.1 Initialisierung auf der Echtzeitumgebung

Nachdem das Modul gestartet wurde, befindet es sich in einem Wartezyklus der alle 100 ms die Variable „Start“ abfragt. Wechselt diese Variable auf „True“ wird mit einer weiteren While-Schleife, deren Schleifendurchläufe dem Wert der Variablen „Wertemenge“ entspricht, im ms Takt ein zweidimensionales Array gebildet. Dieses Array beinhaltet das zu versendende Bussignal über den gesamten Zeitverlauf der Messung. Innerhalb der While-Schleife wird mit dem Modul „Variablen to Bus (SubVI)“ aus den Variablen (Maschine zur Karte) das Bussignal für je einen Zeitpunkt generiert. Den nötigen Zeilenindex erhält das Modul von der Schleife selbst. Zusätzlich werden der Variablen „Ladeindex“ die Anzahl der Schleifendurchläufe und der aktuelle Schleifenindex zugewiesen. Dem Nutzer der Software wird dadurch ein Überblick über den aktuellen Status der Initialisierung mitgeteilt.

 Hochschule für Angewandte Wissenschaften Hamburg Hamburg University of Applied Sciences	Diplomarbeit Fachbereich Elektrotechnik und Informatik	
Wintersemester 07/08 Abgabetermin: 18.12.07	Givian Müller Matr.Nr.: 1749822	Betreuer: Professor Dr. Ing. Müller

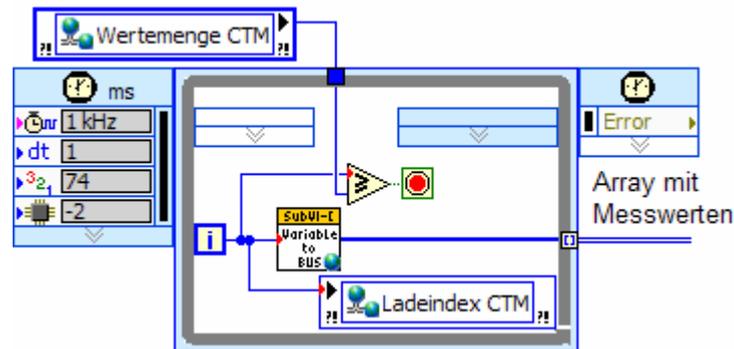


Abb. 25 Programmcode Initialisierung der Messung auf dem Target

Ebenso wie die zu versendenden Signale müssen auch die Wertebereiche für die zu empfangenen Signale eingerichtet werden. Dafür werden alle Variablen (Karte zur Maschine) an ein „Shift Register“ einer weiteren While-Schleife, die für die Kommunikation zuständig ist, übergeben. Zuvor ist es allerdings erforderlich die Variablenarrays mit Werten zu initialisieren, um die gewünschte Länge der Messung vorzugeben.

6.2.1.1 Modulbeschreibung Variable to Bus (SubVI)

Für die Bildung des Bussignals wird vom Modul „Target.vi“ der aktuelle Schleifenindex übergeben. Dieser Index gibt an, welches Element aus den einzelnen Variablenarrays (Maschine zur Karte) gesendet werden soll. Die Elemente der Variablen vom Typ DWord, Word und Bit werden in das Format UInt8 umgewandelt. Entsprechend dem Profibusprotokoll von CTM und Genior werden anschließend alle Werte im Format UInt8 zu einem Array zusammengefasst. Dabei ist darauf zu achten, dass alle unbenutzten Profibusadressen mit Füllbytes, die den Wert Null enthalten, zu belegen sind, da sich sonst das Array verkürzen würde und die entsprechende Information auf dem falschen Byte liegen würde.

Dieses Array wird anschließend wieder dem Modul „Target.vi“ übergeben.

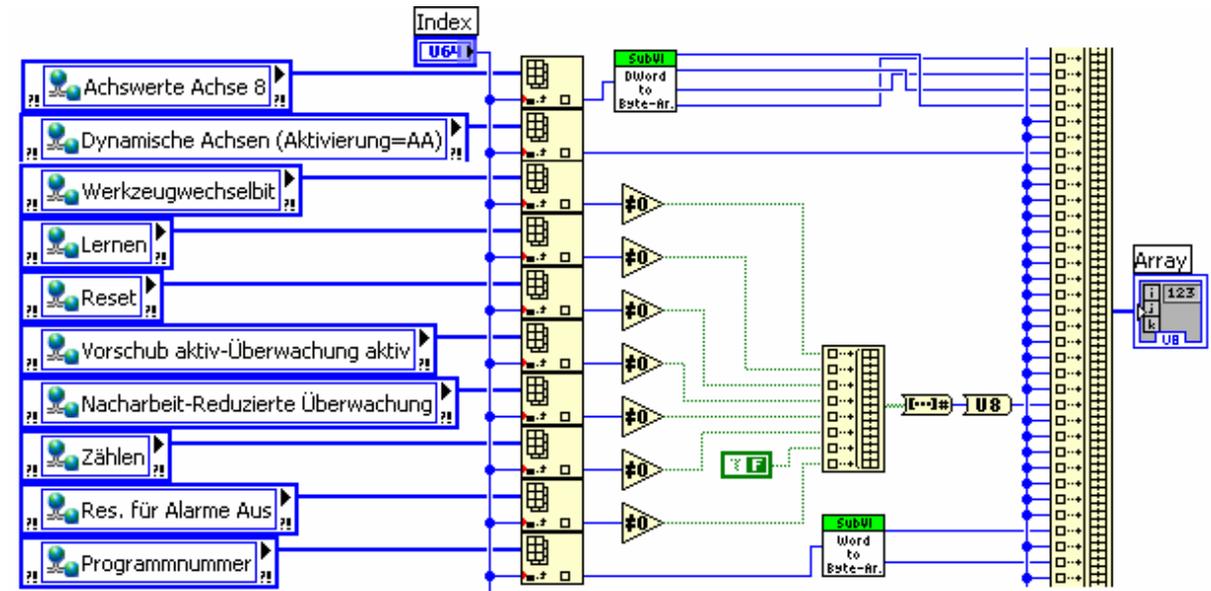


Abb. 26 Programcode Variable to Bus (SubVI)

6.2.2 Durchführung der Messung

Nach der Initialisierung wird die bereits erwähnte While-Schleife mit einem berechneten Zyklustakt gestartet. Der Zyklustakt ergibt sich aus der Variablen „Taktung“ multipliziert mit der Variablen „Leserate“. Die Abtastung der empfangenen Signale erfolgt innerhalb dieser Schleife um das Vielfache der „Leserate“ häufiger als das Ändern des Index der zu versendenden Signale.

Die Anzahl der Schleifendurchläufe entspricht der Variablen „Wertemenge“ multipliziert mit der Variablen „Leserate“. Innerhalb dieser Schleife wird aus dem gebildeten zweidimensionalen Variablenarray das aktuelle Profibussignal gelesen und mit Hilfe des Bausteines der Firma Comsoft auf den Bus geschrieben. Der Index des versendeten Variablenarrays wird zusätzlich an die Variable „Index“ übergeben.

Derselbe Baustein zum Senden des Profibussignals dient ebenso zum Empfangen des Profibussignals von der Karte. Dieses Array wird in jedem Schleifendurchlauf an die Variable „Datenbus“ übergeben.

Darüber hinaus werden entsprechend des Profibusprotokolls von CTM und Genior alle relevanten Bytes mit Ihrer Adresse angesprochen und auf die entsprechende Variable im „Shift Register“ geschrieben. Innerhalb des Registers werden dadurch nacheinander alle bestehenden Werte mit den neuen Messwerten überschrieben. Variablen die mehrere Bytes belegen, werden mit Hilfe des Moduls „Byte-Array-to-Word (SubVI)“ zu einem Wert zusammengefasst, während Variablen die nur aus Bits bestehen mit dem Modul „Byte-to-Bit (SubVI)“ zerlegt werden.

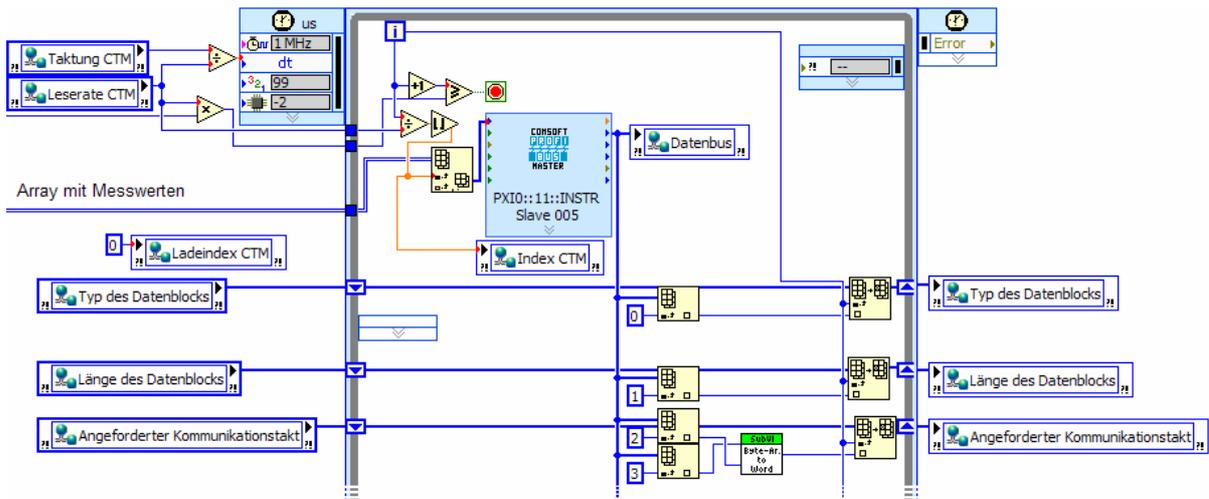


Abb. 27 Programmcode Durchführung der Messung auf dem Target

6.2.3 Speichern der Messwerte

Nach dem letzten Schleifendurchlauf werden alle neu gebildeten Arrays aus den „Shift Registern“ zurück auf die Variablenarrays (Karte zur Maschine) übertragen.

Für eine eventuelle Auswertung wird der Variablen „Neue Messung“ der Wert „True“ zugewiesen. Ebenso wird die Variable „Index“ wieder auf den Wert Null gesetzt.

Anschließend wird die Variable „Messanzahl“ überprüft. Dabei gibt es drei mögliche Zustände.

- Ist der Wert der Variablen „Messanzahl“ größer eins, wird die Anzahl um eins reduziert und die Messung kann von vorn beginnen.
- Ist der Wert der Variablen „Messanzahl“ Eins, wird die Variable „Start“ auf „False“ gesetzt und die Messung ist beendet.
- Ist der Wert der Variablen „Messanzahl“ Null, beginnt die Messung kontinuierlich von vorn ohne die Anzahl zu reduzieren.

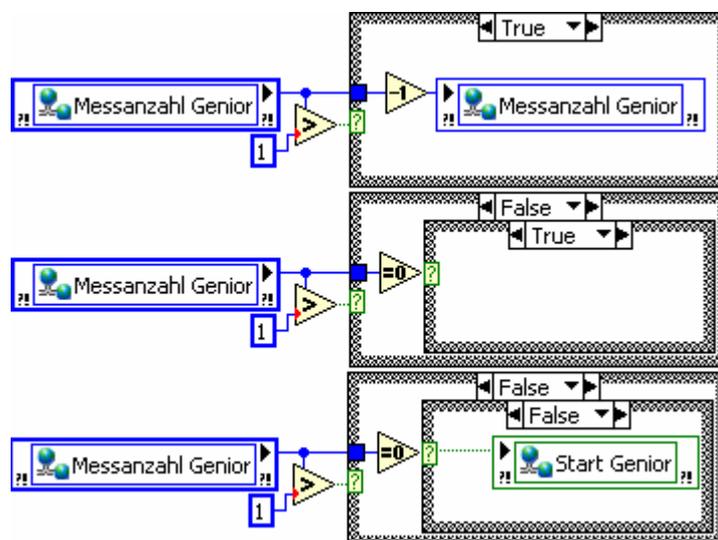


Abb. 28 Programmcode Messanzahl auf dem Target reduzieren

 Hochschule für Angewandte Wissenschaften Hamburg Hamburg University of Applied Sciences	Diplomarbeit Fachbereich Elektrotechnik und Informatik	
Wintersemester 07/08 Abgabetermin: 18.12.07	Givian Müller Matr.Nr.: 1749822	Betreuer: Professor Dr. Ing. Müller

6.2.4 Problem der Echtzeitanforderung

Ursprünglich war geplant nur eine While-Schleife zu betreiben, welcher der gewünschte Zyklustakt vorgegeben wird. Innerhalb der Schleife sollte dann für jeden Schleifenindex sowohl die Bildung des Bussignals aus den Variablen (Maschine zur Karte) wie auch die Zerlegung des Bussignals und Zuweisung auf die Variablen (Karte zur Maschine) erfolgen. Dadurch wurde der Kommunikation die gleiche Priorität wie der eigentlichen Messung zugeteilt. Dies hatte zur Folge, dass der gewünschte Zyklustakt nicht eingehalten wurde. Das Problem wurde durch die Auslagerung der Kommunikation aus der echtzeitfähigen While-Schleife und der Verwendung der „Shift Register“ gänzlich beseitigt. Denkbar wäre auch eine getrennte Programmierung von Kommunikation und Messung mit unterschiedlicher Priorität. Die Übergabe der Werte zwischen den Schleifen würde dabei durch FIFOs⁴⁵ geschehen. Jedoch zeigte sich diese Variante als wesentlich umfangreicher und rechenintensiver als die derzeitige Lösung.

6.3 Softwaremodule zur Onlinebetrachtung der Messung

Eine der Anforderungen des Teststandes im Pflichtenheft ist die Onlinebetrachtung der versendeten und empfangenen Signale. Über den aktuellen Schleifenindex im Modul „Target.vi“, welcher auf die Indexvariable übertragen wird, kann das Element der versendeten Signale, da es bereits vor der Messung bekannt ist, direkt ermittelt werden. Da die empfangenen Signale während der Messung auf dem Target lediglich im „Shift Register“ abgelegt werden und erst am Ende der Messung auf die Variablen übertragen werden, ist ein Betrachten dieser Werte während der Messung nicht möglich. Aus diesem Grund wird im Modul „Target.vi“ der gesamte empfangene Datenbus zusätzlich auf ein Datenbusarray geschrieben, welches immer die aktuellen Werte der Messung enthält. Das Speichern dieses einen Arrays in der Variablen „Datenbus“ belastet die Echtzeitumgebung nur geringfügig, so dass die eingestellte Laufzeit eingehalten wird. Der Datenbus kann dann unabhängig vom Echtzeitsystem auf der Programmierumgebung während der laufenden Messung betrachtet werden.

⁴⁵ First In First Out

6.3.1 Bedienoberflächen der Onlinebetrachtung

Die Betrachtung der Signale ist laut Forderung des Pflichtenheftes in vier Module aufgeteilt.

- a. Simulation Global M-K
Globale Signale von der simulierten Maschine zur Karte
- b. Simulation Kanäle M-K
Kanalabhängige Signale von der simulierten Maschine zur Karte
- c. Simulation Global K-M
Globale Signale von der Karte zur simulierten Maschine
- d. Simulation Kanäle K-M
Kanalabhängige Signale von der Karte zur simulierten Maschine

Alle Werte die einen Zahlenwert annehmen sind als numerische Anzeige dargestellt, während die Variablen im Bit Format den jeweiligen Zustand „True“ oder „False“ als grüne LED⁴⁶ darstellen.

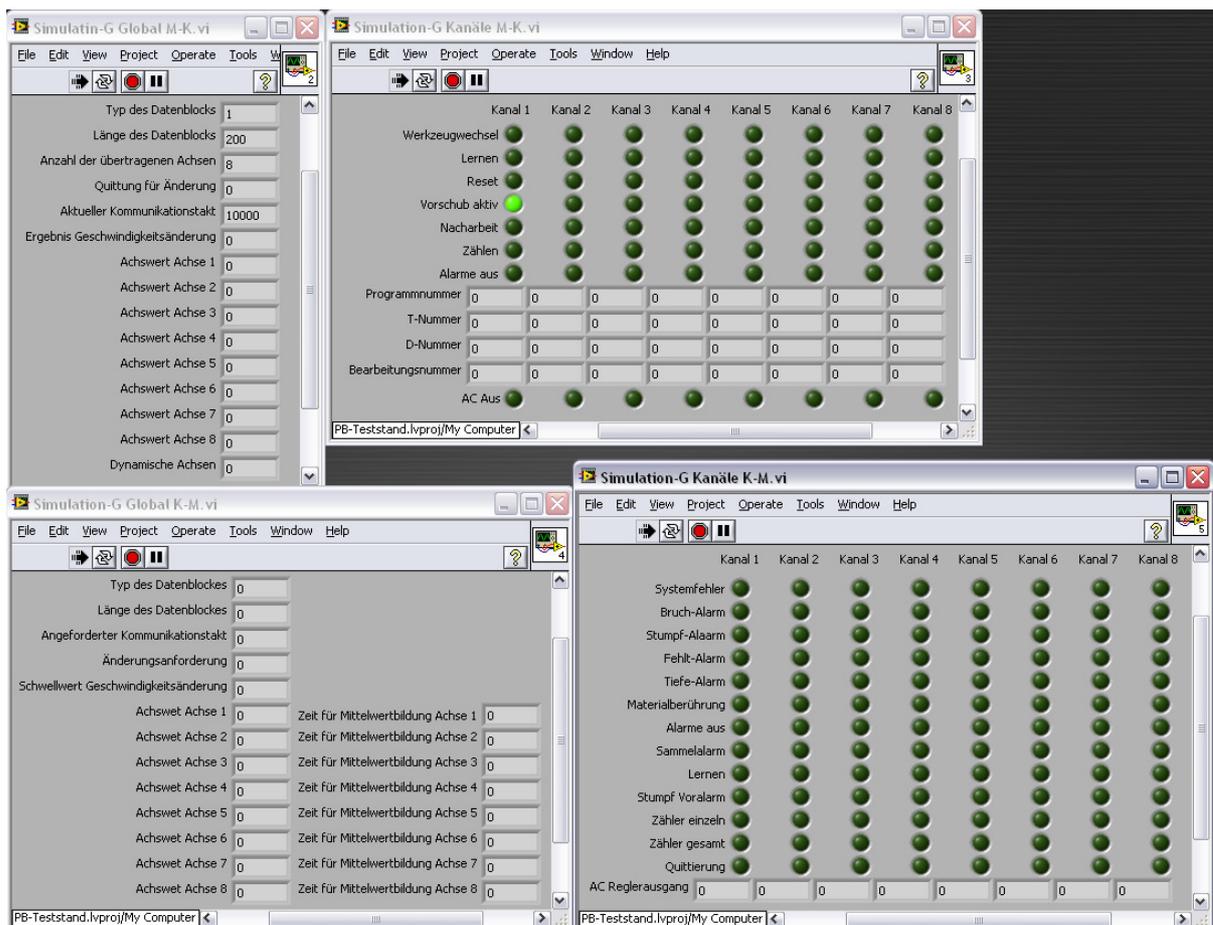


Abb. 29 Bedienoberflächen der Onlinebetrachtung der Messung

⁴⁶ Leuchtdiode / Lumineszenz Diode

 Hochschule für Angewandte Wissenschaften Hamburg Hamburg University of Applied Sciences	Diplomarbeit Fachbereich Elektrotechnik und Informatik	
Wintersemester 07/08 Abgabetermin: 18.12.07	Givian Müller Matr.Nr.: 1749822	Betreuer: Professor Dr. Ing. Müller

6.4 Modul zur Projektierung der Signalverläufe

Das Modul „Projektierung.vi“ ist die Grundlage für jede Messung. Mit diesem Modul ist es möglich vorhandene Signalverläufe beider Kartentypen einzulesen und zu verändern, sowie neue Signalverläufe zu generieren. Jeder generierte Signalverlauf wird in einer Textdatei gespeichert. Dies hat den Vorteil, dass Werte auch direkt in der Textdatei verändert werden können, um anschließend wieder eingelesen zu werden. Beispielsweise könnten auch komplexe Maschinendaten, die von Hand nur schwer zu konfigurieren wären, direkt in die Datei kopiert werden. Ebenso wie die zu sendenden Signale werden auch alle empfangenen Signale in einer Textdatei abgelegt.

6.4.1 Textdateien der gesendeten und empfangenen Signale

Die Textdateien für die Eingabe und die Ausgabe enthalten je einen Header mit folgenden Konfigurationsdaten:

- a. Datum der letzten Änderung der Datei
- b. Uhrzeit der letzten Änderung der Datei
- c. Bezeichnung der Messdaten in der Datei (vom Nutzer einzugeben)
- d. Kommentar der Messdaten in der Datei (vom Nutzer einzugeben)
- e. Name des in der Zeile befindlichen Signals

Darüber hinaus enthält die Ausgabedatei zwischen Punkt d. und e. noch die bei der Messung verwendete Leseratte, um bei der Auswertung die Messwerte in den richtigen Bezug zueinander zu bringen.

Im Anschluss an den Header folgt in beiden Dateien ein zweidimensionales Array mit den projizierten Signalen. Die erste Spalte enthält den Index der Zeile beginnend bei Null. Die restlichen Spalten enthalten die Signale, die dem entsprechenden Signalnamen aus dem Header zugeordnet sind.

	A	B	C	D	E	F	G	H	I
1	Datum	27.09.2007							
2	Uhrzeit	15:09							
3	Bezeichnung	Probdatei							
4	Kommentar	Versuch 1							
5	Time	Typ des Date	Länge des Da	Anzahl der ut	Quittung für A	Aktueller Kon	Ergebnis Ges	Achswert Acl	Achswert Acl
6	0	1	200	8	0	10000	0	0	0
7	10	1	200	8	0	10000	0	1	0

Abb. 30 Genior Input Datei

	A	B	C	D	E	F	G	H	I
1	Datum	27.09.2007							
2	Uhrzeit	15:30							
3	Bezeichnung								
4	Leserate	10							
5	Kommentar	Versuch 1							
6	Time	Typ des Date	Länge des Da	Änderungsan	Änderungsan	Schwelle Ges	Zeit für Mittel	Achswert Acl	Zeit für Mittel
7	0	1	1	1	1	1	1	1	1
8	10	1	1	1	1	1	1	1	1

Abb. 31 Genior Output Datei

 Hochschule für Angewandte Wissenschaften Hamburg Hamburg University of Applied Sciences	Diplomarbeit Fachbereich Elektrotechnik und Informatik	
Wintersemester 07/08 Abgabetermin: 18.12.07	Givian Müller Matr.Nr.: 1749822	Betreuer: Professor Dr. Ing. Müller

6.4.2 Bedienoberfläche der Projektierung

Die Bedienoberfläche der Projektierung besitzt im oberen linken Bereich Eingabefelder für die gewünschte Pfadadresse, sowie der Bezeichnung und des Kommentars zu dieser Datei. Unterhalb dieser Module befindet sich ein weiteres Eingabefeld zum Einstellen einer Pfadadresse, zum Erstellen einer Kopie, sowie eine Schaltfläche „Kopie erstellen“ zum Anlegen dieser Kopie.

Rechts daneben befindet sich die Anzeige der aktuellen Laufzeit der eingestellten Datei, welche von den darüber liegenden Eingabefeldern „Sende Takt [us⁴⁷]“, „Leserate“ und „Wertemenge“ abhängig ist.

Im oberen rechten Bereich liegen die Eingabefelder zum Verändern der Signale aus der Datei. Dazu zählen „Zeit von [ms]“, „Zeit bis [ms]“, „Signalart“, „Amplitude“, „Offset“, „Rauschen“ sowie die Schaltflächen „Wert erzeugen“ und „Kanal 1 für alle Kanäle“. Unterhalb dieses Bereiches sind die Bedienmodule zum Einstellen zweier Kanäle aus der Liste der Profibusadressen, die im darunter liegenden Signalverlaufsgrafen betrachtet werden können. Beide Kanäle verfügen über einen Zoomfaktor, der es ermöglicht Signale in der Skalierung anzupassen.

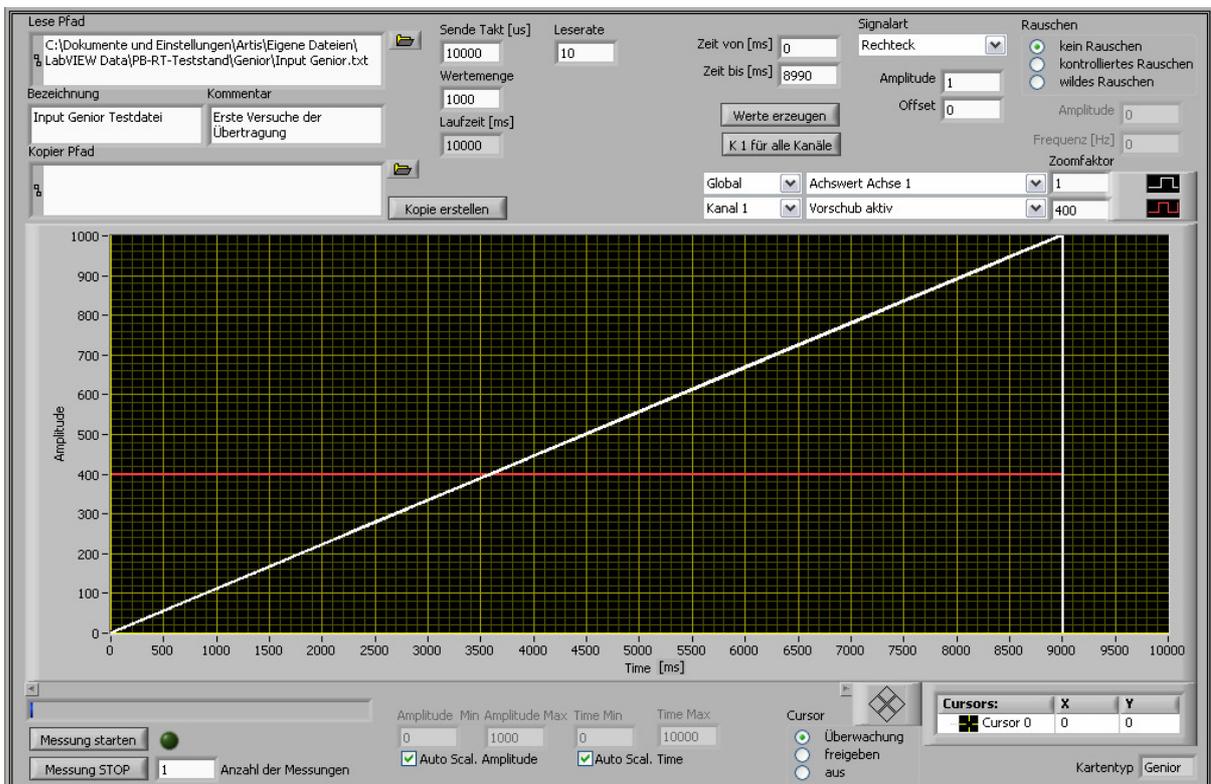


Abb. 32 Bedienoberfläche der Projektierung

Mittig unterhalb des Signalverlaufes befindet sich der Einstellbereich zum Skalieren der Grafik. Das Minimum und Maximum von Time [ms] und Amplitude wird durch Checkboxes entweder automatisch auf den maximalen Wertebereich eingestellt, oder zur manuellen

⁴⁷ Mikro Sekunden

 Hochschule für Angewandte Wissenschaften Hamburg Hamburg University of Applied Sciences	Diplomarbeit Fachbereich Elektrotechnik und Informatik	
Wintersemester 07/08 Abgabetermin: 18.12.07	Givian Müller Matr.Nr.: 1749822	Betreuer: Professor Dr. Ing. Müller

Skalierung freigegeben. Wahlweise ist auch ein Cursor einzublenden, der entweder frei verschoben werden kann oder als Betrachtungsindex der Messung fungiert.

Im unteren linken Bereich befindet sich ein Ladebalken der signalisiert, dass das Programm mit der Projektierung eines Wertebereiches beschäftigt ist oder die Messung auf dem Echtzeitsystem initialisiert wird. Unterhalb des Ladebalkens liegt die Schaltflächen „Messung starten“ und „Messung STOP“ zum Ein- und Ausschalten der Messung, eine Signalleuchte die aktiv ist, solange die Messung andauert, sowie ein Eingabefeld für die Anzahl der durchzuführenden Messungen.

In der unteren rechten Ecke befindet sich das Anzeigefeld „Kartentyp“ welches signalisiert, welcher Kartentyp (CTM oder Genior) aktuell projiziert wird.

6.4.3 Bedienung der Projektierung

Nach dem Start des Programms wird zunächst der Benutzer der Software mit einer „Message Box“ gefragt welcher Kartentyp (CTM oder Genior) projiziert werden soll. Diese Einstellung kann während der Projektierung nicht mehr geändert werden. Sollte der falsche Typ gewählt werden, muss das Modul erneut gestartet werden. Welcher Kartentyp gerade projiziert wird erscheint als Name im Textfeld unten rechts.

Anschließend muss der „Lese Pfad“ der zu verwendenden Textdatei eingestellt werden. Solange keine Datei angegeben wurde ist auch keine weitere Projektierung möglich, denn alle relevanten Schaltflächen zum Verändern der Daten sind zu diesem Zeitraum nicht zu bedienen. Sie werden erst durch das Auswählen einer Datei schaltfähig.

Die einzustellende Datei kann bereits vorhanden sein, in diesem Fall werden alle benötigten Daten aus der Datei gelesen. Falls sie noch nicht besteht, wird vom Programm eine neue Textdatei mit diesem Namen erstellt.

Wenn eine neue Textdatei erstellt wurde, kann anschließend die Bezeichnung und der Kommentar der Datei eingetragen werden. Danach wird die Laufzeit der Datei definiert, dieses geschieht durch Einstellen des Sendetaktes und der Wertemenge. Für die Abtastrate der zu empfangenen Signale muss die Leserate definiert werden. Beim Starten der Software wird diese auf das zehnfache des Sendetaktes eingestellt.

Anschließend kann mit dem Bearbeiten der Werte begonnen werden. Dafür wird das zu ändernde Signal auf den oberen Kanal des Diagramms eingestellt, da alle Änderungen nur für diesen Kanal wirksam werden.

Zum Ändern des Signalverlaufes wird zunächst der Zeitbereich, der verändert werden soll, mit „Zeit von [ms]“ und „Zeit bis [ms]“ eingestellt. Anschließend wird die Signalart „Rechteck“, „Rampe“ oder „Kurve“ mit der entsprechenden Amplitude gewählt. Bei einem eingestellten Achswert kann wahlweise ein Offset oder Rauschen dazugeschaltet werden. Beim Rauschen unterscheidet man zwischen „kontrolliertem Rauschen“ mit einzustellender Amplitude und Frequenz und „wildem Rauschen“ mit einstellbarer Amplitude und zufällig variierender Frequenz.

Nachdem alle Einstellungen vorgenommen wurden, um den Signalverlauf zu ändern, wird der eingestellte Wertebereich mit der Schaltfläche „Werte erzeugen“ verändert. Dieser Vorgang kann beliebig oft und eigenständig für jedes Signal und jedes Intervall im

 Hochschule für Angewandte Wissenschaften Hamburg Hamburg University of Applied Sciences	Diplomarbeit Fachbereich Elektrotechnik und Informatik	
Wintersemester 07/08 Abgabetermin: 18.12.07	Givian Müller Matr.Nr.: 1749822	Betreuer: Professor Dr. Ing. Müller

vorhandenen Bereich wiederholt werden. Um einen besseren zeitlichen Bezug zwischen den Signalen zu bekommen, kann während der Projektierung der Messsignale ein weiteres Signal auf den zweiten Kanal gegeben werden. Da allerdings ein Vergleich eines Signals im Bereich von 0 bis 10000 mit einem Signal von 0 bis 1 nicht zu erkennen ist, können die Pegel innerhalb der Grafik mit dem eingestellten „Zoomfaktor“ vergrößert werden. So multipliziert sich das eingestellte Signal mit dem eingestellten Zoomfaktor.

Ebenso kann die Autoskalierung abgeschaltet werden, so dass mit Ändern der minimalen und maximalen Grenzwerte der Grafikbereich angepasst werden kann.

Für den Fall, dass die Projektierung des ersten Kanals für alle restlichen Kanäle übereinstimmt, kann mit der Schaltfläche „K1 für alle Kanäle“ der projektierte Wertebereich von Kanal 1 auf alle Kanäle übertragen werden. Nach der Übertragung auf alle Kanäle können weiterhin alle Signale unabhängig voneinander bearbeitet werden.

Nachdem alle Messsignale projektiert sind, wird die gewünschte Häufigkeit der Messung im Feld „Anzahl der Messungen“ eingetragen. Wird eine dauerhafte Wiederholung der Messung gewünscht, so wird dieser Wert auf 0 gesetzt. Durch anschließendes Betätigen der Schaltfläche „Messung starten“ wird die Messung begonnen.

Durch Betätigen der Schaltfläche „Messung STOP“ wird nach Beenden der aktuellen Messung keine weitere Messung mehr durchgeführt.

Beim frühzeitigen Beenden der Software wird die vorgegebene Messanzahl ohne weitere Aufforderungen des Nutzers durchlaufen.

Während der Messung läuft, bei eingeschalteter Überwachung, der Cursor über die Grafik und signalisiert den aktuell übertragenen Wert der eingestellten Kanäle. Ebenso reduziert sich die Anzahl der Messungen nach jedem Messdurchlauf, bis sie den Wert 1 erreicht hat.

6.4.4 Eigenschaften der Projektierung

Im Folgenden sind sowohl alle Eigenschaften der Projektierung aufgeführt, die im Pflichtenheft gefordert wurden, als auch alle nützlichen nennenswerten Eigenschaften, die die Eingabe und Projektierung erleichtern.

6.4.4.1 Auswahl des Kartentyps

Durch die Verwendung von unterschiedlichen Profibusprotokollen der Systeme CTM und Genior ist es nicht möglich für beide Kartentypen ein kartenunabhängiges Modul zur Projektierung zu benutzen. Ursprünglich war für jede Karte ein eigenständiges Modul geplant, jedoch mit dem Nachteil, dass Erweiterungen der Software immer in beiden Modulen umzusetzen wären. Das würde einen erheblich größeren Aufwand für den Programmierer bedeuten. Aus diesem Grund wurde für die Projektierung lediglich ein Modul entwickelt, welches mit beiden Kartentypen kompatibel ist. Beim Starten des Moduls wird der Nutzer der Software mit einer Auswahlbox aufgefordert den zu projektierenden Kartentyp anzugeben. Sollte dieser Vorgang unterbrochen werden, indem beispielsweise das Fenster geschlossen wird, ist standardmäßig Genior Projektierung eingeschaltet.

 Hochschule für Angewandte Wissenschaften Hamburg Hamburg University of Applied Sciences	Diplomarbeit Fachbereich Elektrotechnik und Informatik	
Wintersemester 07/08 Abgabetermin: 18.12.07	Givian Müller Matr.Nr.: 1749822	Betreuer: Professor Dr. Ing. Müller



Abb. 33 Bedienoberfläche Auswahl des Kartentyps

Nachdem der Kartentyp ausgewählt wurde, ist es nicht mehr möglich diesen zu ändern. Eine Änderung kann nur über Neustart des Moduls erfolgen. Der Grund hierfür ist, dass nach der Auswahl des Kartentyps zunächst einige kartenabhängige Grundkonfigurationen geladen werden, die anschließend nicht wieder aufgerufen werden. Der erneute Aufruf dieser Grundkonfiguration während der Laufzeit würde einem Neustart des Moduls gleichkommen. Innerhalb des Programmablaufes werden alle kartenspezifischen Programmierungen mit Case-Bedingungen auf den jeweiligen gewählten Kartentyp geschaltet.

6.4.4.2 Eingabe des Quellverzeichnisses

Sobald der „Lese Pfad“ gewählt oder verändert wird, überprüft das Programm ob es sich um einen gültigen Pfad handelt. Es handelt sich um Dateien mit der Endung „.txt⁴⁸“. Wird die Datei erkannt, werden die einzelnen Elemente aus der Datei gelesen und mit dem Modul „Input to Variablen (SubVI)“ auf die entsprechenden Variablen kopiert. Selbiges gilt für die Bezeichnung und den Kommentar der Datei. Darüber hinaus werden die Wertemenge und die Laufzeit aus der Datei ermittelt. Um später alle einzelnen Signale betrachten zu können ohne jedes Mal erneut Werte aus der Datei zu lesen, wird zusätzlich ein zweidimensionales Array angelegt, in das sämtliche Signalverläufe kopiert werden.

Handelt es sich bei der Eingabe des „Lese Pfades“ nicht um eine bereits bestehende Textdatei wird diese mit dem angegebenen Namen erstellt. In diesem Fall wird sowohl dem Array als auch der Wertemenge der Wert 0 übergeben, wodurch eine nicht definierte Laufzeit entsteht. Um dadurch entstehende Probleme sowohl in der Darstellung als auch in der Berechnung zu vermeiden, wird in diesem Fall automatisch eine Änderung der Wertemenge auf den Wert 1 vorgenommen.

Zusätzlich werden die Wertemenge auf die Variable „Wertemenge“ und der Name der projektierten Datei auf die Variable „Inputdatei“ übertragen. Darüber hinaus werden die Signaldaten aus der Datei mit dem Modul „Input to Variablen (SubVI)“ auf die Kommunikationsvariablen (Maschine zur Karte) übertragen. Ebenso wird mit der Wertemenge, multipliziert mit der eingestellten Leserate, die Länge der benötigten Arrays der Kommunikationsvariablen (Karte zur Maschine) bestimmt.

⁴⁸ Textdatei

 Hochschule für Angewandte Wissenschaften Hamburg Hamburg University of Applied Sciences	Diplomarbeit Fachbereich Elektrotechnik und Informatik	
Wintersemester 07/08 Abgabetermin: 18.12.07	Givian Müller Matr.Nr.: 1749822	Betreuer: Professor Dr. Ing. Müller

6.4.4.2.1 Modulbeschreibung Input to Variablen (SubVI)

In diesem Modul wird aus der Datei des übergebenen Dateipfades der Header mit dem zweidimensionalen Signalarray eingelesen. Da nur die Signalwerte für die Messung von Bedeutung sind, wird der Header aus dem Wertebereich entfernt und nicht weiter betrachtet. Das resultierende Signalarray wird dann in die einzelnen eindimensionalen Teilarrays zerlegt, welche anschließend den entsprechenden Variablen zugewiesen werden.

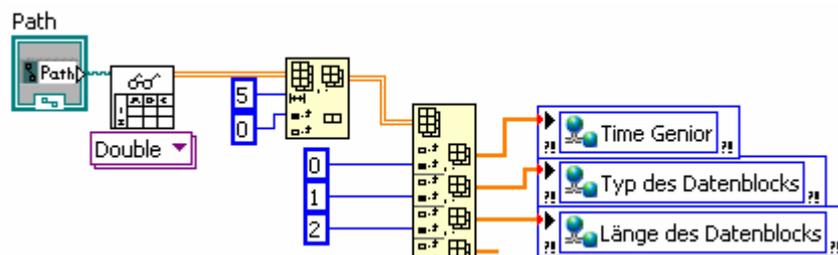


Abb. 34 Programmcode Input to Variablen (SubVI)

6.4.4.3 Eingabe von Bezeichnung und Kommentar

Jede Änderung in den Eingabefeldern „Bezeichnung“ und „Kommentar“ wird direkt in die eingestellte Textdatei übernommen. Bei der Initialisierung einer neuen Textdatei wird automatisch der eingetragene Dateiname zur Bezeichnung der Datei.

6.4.4.4 Erstellen einer Kopie der Datei

Im Laufe der Projektierung kann es vorkommen, dass eine projektierte Datei für eine spätere Messung erhalten bleiben soll, allerdings selbige Werte der Datei als Grundlage für eine neue Messung dienen sollen. Um in diesem Fall keine komplett neue Datei erstellen zu müssen, kann im „Kopier Pfad“ ein Speicherort vorgegeben werden und mit der Schaltfläche „Kopie erstellen“ die derzeit verwendete Datei unter neuem Namen abgespeichert werden. Die Überprüfung auf die Endung „.txt“ erfolgt wie bereits bei der Eingabe des Speicherpfades.

Anschließend wird der neue Dateiname in das Feld des Speicherpfades übernommen, um neue Änderungen der Werte in dieser Datei vorzunehmen.

6.4.4.5 Veränderung der Taktung und Leserate

Der Sendetakt beschreibt die Taktrate in der eine Änderung der zu verschickenden Signale auf dem Profibus erfolgt. Beim Starten des Moduls wird er auf 10000 us eingestellt. Eine Änderung der Taktung kann jederzeit vorgenommen werden und wirkt sich direkt auf die Laufzeit aus. Einzugeben ist der Takt in Mikrosekunden. Jede Änderung der Taktung wird

 Hochschule für Angewandte Wissenschaften Hamburg Hamburg University of Applied Sciences	Diplomarbeit Fachbereich Elektrotechnik und Informatik	
Wintersemester 07/08 Abgabetermin: 18.12.07	Givian Müller Matr.Nr.: 1749822	Betreuer: Professor Dr. Ing. Müller

direkt auf das Echtzeitsystem übertragen. Der minimal einzustellende Takt liegt etwa bei 700 us. Die Ermittlung dieses Wertes ist im Kapitel 7.2 beschrieben.

Die Leserate wird beim Starten des Moduls auf den Wert Zehn gesetzt. Das bedeutet, es werden zehnmal öfter Daten vom Profibus gelesen als neue Werte auf ihn geschrieben. Eine Erhöhung dieses Wertes hat zu Folge, dass noch schneller gelesen wird, während eine Verkleinerung, bis beispielsweise zum Minimalwert 1, ein gleiches Verhältnis vom Lesen zum Schreiben zur Folge hätte. Durch eine höhere Leserate als 1 müssen die Variablenarrays (Karte zur Maschine) um das Vielfache der Variablen größer sein als die der Variablenarrays (Maschine zur Karte). Eine Änderung dieses Wertes reserviert mit dem Modul „Output Variablen Init (SubVI)“ den benötigten Speicherbereich der Variablen. Ebenso wird der Wert direkt auf die Variable „Leserate“ übertragen.

6.4.4.5.1 Modulbeschreibung Output Variablen Init (SubVI)

Diesem Modul wird die neu generierte Laufzeit übergeben, so dass jedes Variablenarray (Karte zur Maschine) mit dieser Länge und dem Wert 1 initialisiert wird. Die Art des Wertes ist dabei allerdings völlig irrelevant, da bei der Messung sämtliche Werte innerhalb des Arrays überschrieben werden.

6.4.4.6 Veränderung der Wertemenge

Eine Änderung der Wertemenge wird entweder von dem Programm selbst bei der Initialisierung vorgenommen oder manuell durch den Nutzer der Software. Sobald die Wertemenge geändert wurde, wird zunächst überprüft, ob sie vergrößert oder verkleinert wurde. Akzeptiert werden nur ganzzahlige Änderungen der Wertemenge. Jegliche anderweitige Eingabe wird auf die nächste ganze Zahl abgerundet.

Wurde die Wertemenge verkleinert, wird dem Array die neue Länge zugewiesen. Das hat zur Folge, dass alle hinteren, nicht mehr benötigten Werte aus dem Array, nicht mehr adressiert und somit für das Programm nicht mehr bekannt sind.

Wurde die Wertemenge vergrößert, was auch bei der Initialisierung der Fall ist, wird mit Hilfe einer For-Schleife das „Timearray“ erstellt. Die Anzahl der Schleifendurchläufe entspricht dabei der Zeitdifferenz zwischen der alten und der neuen eingestellten Zeit. Alle anderen Variablen in dem Array werden mit Standardwerten gefüllt, die in einer weiteren Textdatei hinterlegt sind. Diese heißt „CTM Init.txt“ für CTM und „Genior Init.txt“ für Genior. Während die Werte erzeugt werden läuft der Ladebalken entsprechend der Anzahl der Schleifendurchläufe von 0 bis 100 %. Um zu gewährleisten, dass während der Messung der Wertebereich der Variablenarrays (Karte zur Maschine) die richtige Länge besitzt, wird an dieser Stelle wiederum mit dem Modul „Output Variablen Init (SubVI)“ der Wertebereich der Variablen (Karte zur Maschine) reserviert. Anschließend wird die Wertemenge neu berechnet und die Daten werden in den Variablenarrays abgelegt.

Aus der Wertemenge, multipliziert mit der eingestellten Taktung, ergibt sich die aktuelle Laufzeit der projektierten Messung.

 Hochschule für Angewandte Wissenschaften Hamburg Hamburg University of Applied Sciences	Diplomarbeit Fachbereich Elektrotechnik und Informatik	
Wintersemester 07/08 Abgabetermin: 18.12.07	Givian Müller Matr.Nr.: 1749822	Betreuer: Professor Dr. Ing. Müller

6.4.4.7 Veränderung der Signalverläufe

Grundsätzlich kann jedes Signal, welches in der Adressbusliste von CTM und Genior vorhanden ist, geändert werden. Dabei ist zu beachten, dass nicht jedes Signal jeden Wert annehmen darf. Bei der Änderung der Werte innerhalb der Textdatei kann dieses sicherlich nicht abgefangen werden. Das Programm hingegen lässt nur Werte zu, die auch für den entsprechenden Wert geeignet sind. Sollten Signale innerhalb der Datei auf Werte geändert werden, die für das entsprechende Signal unmöglich sind, werden diese vom Programm erkannt und für die Messung auf die maximalen Grenzwerte gesetzt. Eine Änderung in der Datei erfolgt allerdings nicht.

Um einen Wert mit der Software zu ändern, muss die entsprechende Variable auf dem oberen Kanal des Signalverlaufes eingestellt werden. Änderungen erfolgen immer im Bereich des eingestellten Zeitintervalls „Zeit von [ms]“ bis „Zeit bis [ms]“. Wie auch bei der Wertemenge können hier nur ganzzahlige Werte eingestellt werden. Jede andere Eingabe wird auf den nächsten ganzzahligen Wert gerundet. Haben beide Eingabefelder denselben Wert, wird nur der eine eingestellte Impuls des Signals verändert.

Zum Ändern der Werte muss die Amplitude und die Art des Signals eingestellt werden. Für die Art des Signals stehen drei Signaltypen zur Verfügung.

- a. Rechteck
 Ein Rechtecksignal erzeugt einen konstanten Wert über den eingestellten Zeitverlauf. Die Höhe des Wertes entspricht der Amplitude. Rechtecksignale sind für alle Signalvariablen möglich.
- b. Rampe
 Eine Rampe erzeugt eine gleichmäßig ansteigende Schräge von 0 bis zum Amplitudenwert über den eingestellten Zeitverlauf. Rampen können nur für Achswerte erzeugt werden.
- c. Kurve (Parabel)
 Beim Erzeugen einer Kurve wird eine invertierte Parabel erzeugt, die einen Maximalwert in der Höhe der Amplitude besitzt. Die Nullstellen der Kurve liegen bei den eingestellten Grenzwerten der Zeit. Kurven können nur für Achswerte erzeugt werden. Für die Generierung des Kurvensignals wurde ein Frequenzgenerator nach folgendem Schema, dargestellt in Abb. 35, verwendet. Die Variable Zeit gibt die eingestellte Signallänge an.

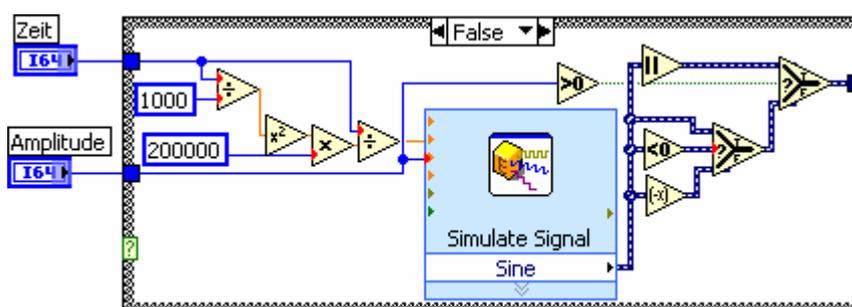


Abb. 35 Programmcode Kurve erzeugen

Die Achswerte können zusätzlich noch mit einem Offset versehen werden, der alle Werte über die Länge des eingestellten Zeitbereiches erhöht oder bei negativem Offset verringert. Darüber hinaus ist es auch möglich eine negative Amplitude einzugeben, wodurch dieselbe Kurve im negativen Bereich erzeugt wird.

Ebenso kann der Achswert durch ein Rauschsignal überlagert werden. Eingestellt werden kann ein „kontrolliertes Rauschen“ mit vorgegebener Amplitude und Frequenz oder aber ein „wildes Rauschen“ mit generierten Zufallswerten im Bereich der eingestellten Amplitude.

6.4.4.7.1 Kanal 1 für alle Kanäle übernehmen

Da grundsätzlich Änderungen der Signalverläufe nur einzeln möglich sind, kommt es bei der Erstellung einer Messreihe unter Verwendung aller Kanäle zu erheblicher Bearbeitungszeit. Für diesen Fall wurde die Schaltfläche „Kanal 1 für alle Kanäle“ geschaffen. Nach betätigen dieser Schaltfläche werden alle eingestellten Werte von Kanal 1 auf die restlichen Kanäle übernommen. Sicherlich macht die Verwendung dieses Features nicht bei jeder Messung Sinn, jedoch kommt es mehrfach vor, alle Kanäle mit denselben Signalen zu versorgen. Darüber hinaus kann nach dem Kopieren weiterhin jeder Kanal angepasst werden.

Bei der Ausführung dieses Features werden die entsprechenden Spalten von Kanal 1 mit Hilfe einer For-Schleife entsprechend der Kanalanzahl auf die Spalten der anderen Kanäle übertragen und anschließend in der Datei gespeichert.

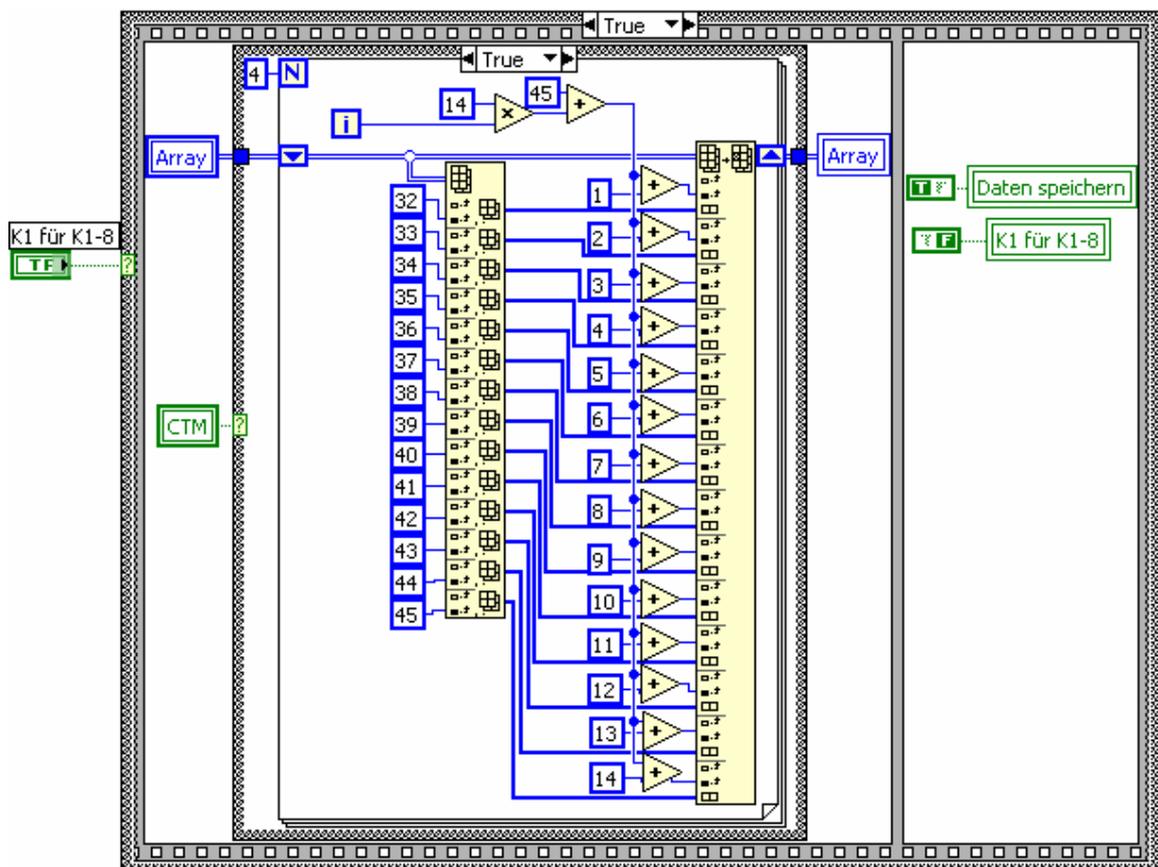


Abb. 36 Programmcode Kanal 1 für alle Kanäle übernehmen (CTM)

 Hochschule für Angewandte Wissenschaften Hamburg Hamburg University of Applied Sciences	Diplomarbeit Fachbereich Elektrotechnik und Informatik	
Wintersemester 07/08 Abgabetermin: 18.12.07	Givian Müller Matr.Nr.: 1749822	Betreuer: Professor Dr. Ing. Müller

6.4.4.8 Speichern der Daten

Das Speichern der Daten wird durch Verändern der Wertemenge, der Bezeichnung oder des Kommentars gestartet. Der Nutzer des Programms bekommt davon während der Projektierung nichts mit, da dieses vollständig im Hintergrund der Software abläuft.

Beim Speichern der Daten wird die vorhandene eingestellte Textdatei komplett überschrieben. Würden die alten Werte nur an geänderter Stelle überschrieben werden, würde bei einer Verkürzung der Wertemenge ein alter, nicht gewünschter Bereich in der Datei vorhanden bleiben.

Zunächst wird nach dem Speicheraufruf der Header erstellt. Dabei werden das aktuelle Datum und die aktuelle Uhrzeit, sowie die eingetragene Bezeichnung und der Kommentar zu einem Array zusammengefasst. Die letzte, fünfte Zeile dieses Headers wird durch das Modul „Input Kopfzeile (SubVI)“ erstellt. Danach wird der Header in die Datei geschrieben. Anschließend wird das im Speicher befindliche zweidimensionale Array mit den aktuellen Signalwerten an den Header in der Datei angehängt. Daraufhin wird, wie bei der Änderung des Speicherpfades, mit dem Modul „Input to Variablen (SubVI)“ das Array mit den aktuellen Signalwerten auf die Variablenarrays (Maschine zur Karte) geschrieben.

6.4.4.8.1 Modulbeschreibung Input Kopfzeile (SubVI)

Dieses Modul bildet ein Array aus Strings mit allen Signalnamen die als Variablen (Maschine zur Karte) existieren. Die Länge des Arrays entspricht der Anzahl der Spalten des zweidimensionalen Signalarrays aus dem Modul „Projektierung“.

6.4.4.9 Kanalauswahl des Signalverlaufs

In der Grafik des Moduls Projektierung können zwei Kanäle angezeigt werden. Jedes vorhandene Signal der Variablenarrays (Maschine zur Karte) kann dargestellt werden. Der untere Kanal dient nur zur Anzeige, während der obere Kanal auch den zu ändernden Wert signalisiert.

Alle Signale die dargestellt werden sind aufgeteilt in die Kanäle der Karten. CTM verfügt über globale Kanäle und Kanal 1 bis Kanal 4. Genior verfügt ebenso über globale Kanäle, sowie über Kanal 1 bis Kanal 8. Abhängig davon, welcher Kanal eingestellt ist, können die globalen Variablen oder die kanalabhängigen Variablen eingestellt werden. Dazu werden die inaktiven Schaltflächen ausgeblendet.

Die eingestellten Werte der Variablen (Kanal, globale Variable und Variable) werden an das Modul „Datei Inputzeile auslesen (SubVI)“ übergeben. Dieses gibt dann den Spaltenindex, den Namen im Format Sting sowie den Typ des ausgewählten Signals zurück.

 Hochschule für Angewandte Wissenschaften Hamburg Hamburg University of Applied Sciences	Diplomarbeit Fachbereich Elektrotechnik und Informatik	
Wintersemester 07/08 Abgabetermin: 18.12.07	Givian Müller Matr.Nr.: 1749822	Betreuer: Professor Dr. Ing. Müller

Mit Hilfe des Zeilenindexes wird aus dem gespeicherten Array das entsprechende Teilarray des Signals ermittelt und in der Grafik dargestellt. Um beispielsweise Achswerte, die im Bereich von mehreren tausend liegen können, mit Vorschubsignalen, die 0 oder 1 sein können, zu vergleichen kann jeder eingestellte Wert mit dem Zoomfaktor vergrößert werden. Der Zoomfaktor beeinflusst dabei nicht den Wert in der Datei oder im Variablenarray, sondern dient lediglich zur besseren Darstellung in der Grafik.

6.4.4.9.1 Modulbeschreibung Datei Inputzeile auslesen (SubVI)

Dieses Modul ermittelt mit Hilfe mehrerer Case-Strukturen den Spaltenindex der eingestellten Variablen. Abhängig vom gewählten Kanal ist das für CTM ein Wert von 1 bis 87 und für Genior ein Wert von 1 bis 111. Darüber hinaus wird der Name der eingestellten Variablen im Format String zurückgegeben. Ebenso wird der vordefinierte Typ der Variablen zurückgegeben. Dabei steht 0 für Bit, 1 für UInt8, 2 für UInt16 und 3 für Int32.

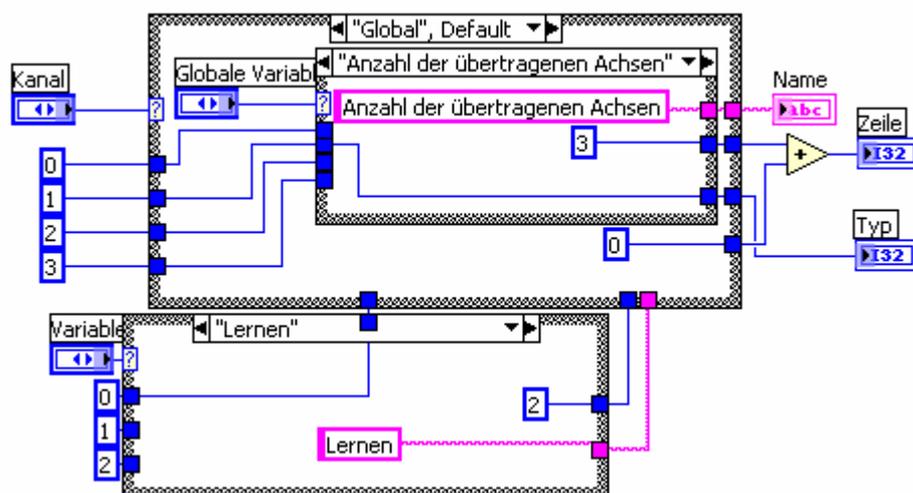


Abb. 37 Programmcode Datei Inputzeile auslesen (SubVI)

6.4.4.9.2 Skalierung des Signalverlaufes

Die Skalierung des Signalverlaufes erfolgt über den minimalen und maximalen Wert der Grafik. Diese Werte können entweder manuell eingegeben, oder separat für die Amplitude und die Zeit automatisch ermittelt werden. Bei der automatischen Ermittlung der Grenzwerte wird für die Zeitachse das Minimum immer 0 und das Maximum entspricht immer der Wertemenge.

Für die Grenzwerte der Amplitude werden das Minimum und das Maximum der aktuellen Signale in der Grafik ermittelt und eingestellt.

6.4.4.10 Ansteuerung des Cursors

Der Cursor der Grafik kann bei Freigabe entweder über „drag and drop“ direkt in der Grafik oder über die Pfeiltasten unterhalb der Grafik verschoben werden. Zusätzlich kann der Cursor mit den Pfeiltasten des Keyboards angesteuert werden. Dafür wurden die Werte der Cursorpositionen bei jedem Durchlauf um einen Index verändert. Die Abfrage der Pfeiltasten erfolgt mit dem Modul „Tastendruck (SubVI)“.

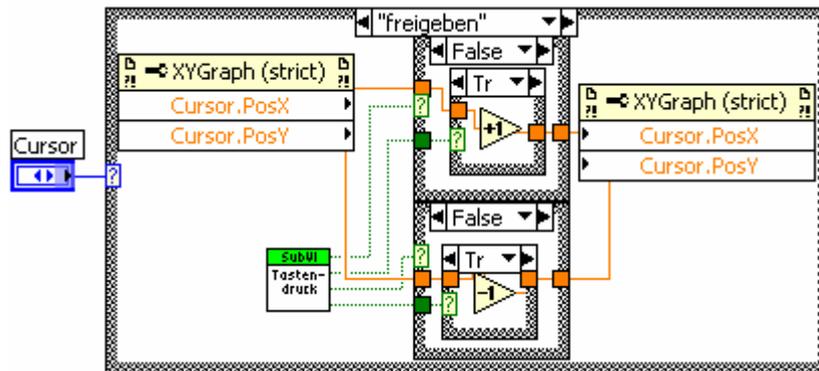


Abb. 38 Programmcode Ansteuerung des Cursors

6.4.4.10.1 Modulbeschreibung Tastendruck (SubVI)

Das Modul „Tastendruck (SubVI)“ initialisiert zunächst die Keyboardabfrage mit dem Baustein „Initialize Keyboard.vi“ und liest anschließend alle Tasten mit dem Baustein „Acquire Input Data.vi“ aus. Mit der Abfrage der Variablennamen der vier Pfeiltasten Left, Right, Up und Down werden anschließend deren Werte ausgelesen und auf Abweichung von minus 1 überprüft. Wird eine Taste betätigt, gibt dieses Modul über den entsprechenden Ausgang Links, Rechts, Hoch und Runter ein „True“ Signal zurück, andernfalls bleibt dieser Ausgang „False“.

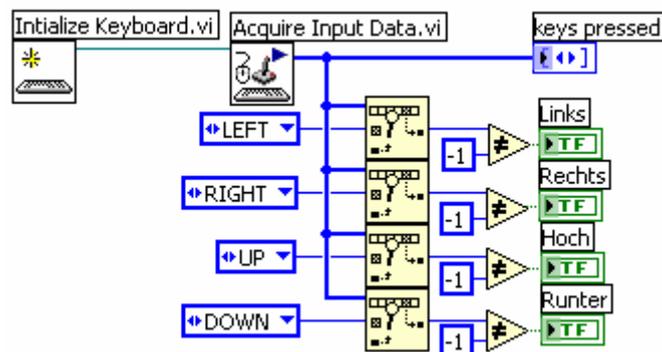


Abb. 39 Programmcode Tastendruck (SubVI)

 Hochschule für Angewandte Wissenschaften Hamburg Hamburg University of Applied Sciences	Diplomarbeit Fachbereich Elektrotechnik und Informatik	
Wintersemester 07/08 Abgabetermin: 18.12.07	Givian Müller Matr.Nr.: 1749822	Betreuer: Professor Dr. Ing. Müller

6.4.4.11 Einhalten der Grenzwerte

Grenzwerte existieren für alle einzugebenden Zahlenwerte. Welcher Wert jedoch in welchem Bereich liegen darf, muss für jede Eingabefläche getrennt betrachtet werden.

- a. Grenzen für Wertemenge, Zeit von [ms] und Zeit bis [ms]

Die Werte für Wertemenge, Zeit von [ms] und Zeit bis [ms] können nur ganzzahlig verändert werden.

Die Eingabe von negativen Zahlen wird durch das Verwenden vom Datentyp UInt verhindert.

Damit keine Änderung des Zeitverlaufes außerhalb der eingestellten Wertemenge möglich ist, wird diese als Grenzwert der jeweiligen Variablen gesetzt.
- b. Grenzen für Variablen

Die Höhe der einzustellenden Amplitude ist abhängig von dem eingestellten Kanal, daher wird mit Hilfe des vom Modul „Datei Inputzeile auslesen (SubVI)“ übergebenen Variablentyps der minimale und maximale Wert festgelegt. Im Falle einer Überschreitung dieser Grenzwerte wird der jeweilige Grenzwert auf die Variable übertragen.

Ebenso wird durch die gleiche Abhängigkeit eine für den Kanal unmögliche Signalart oder ein Rauschen zurückgesetzt. Einstellungen aller Signalarten und dazuschalten eines Rauschens sind lediglich für Achswerte möglich.
- c. Grenzen der Skalierung

Bei der Skalierung der Grafik wurde lediglich die Zeitachse begrenzt. Als Grenzen werden der Wert 0 und die Wertemenge festgelegt, da außerhalb dieses Bereiches keine Werte zum Anzeigen verfügbar sind.

6.4.4.12 Sperren der Eingabe während der Bearbeitung

Um zu vermeiden, dass der Nutzer der Software eine Änderung der Einstellungen vornimmt, wie z. B. eine Laufzeitänderung oder auf eine neue Textdatei zugreift, während das Programm Werte ändert, wird die Bedienoberfläche für diesen Zeitraum gesperrt. Der Mauszeiger wird in dieser Zeit zu einer Sanduhr, die lediglich bewegt werden kann, jedoch keine Schaltgewalt über das Programm besitzt. Nachdem diese Änderungsvorgänge erledigt sind wird der Mauszeiger wieder freigegeben.

 Hochschule für Angewandte Wissenschaften Hamburg Hamburg University of Applied Sciences	Diplomarbeit Fachbereich Elektrotechnik und Informatik	
Wintersemester 07/08 Abgabetermin: 18.12.07	Givian Müller Matr.Nr.: 1749822	Betreuer: Professor Dr. Ing. Müller

6.4.4.13 Besonderheit während der Messung

Das Modul „Projektierung“ kann nicht nur zum Einrichten einer Messung verwendet werden, sondern bietet die Möglichkeit, während der Messung den aktuellen übertragenen Wert anzuzeigen. Wird die Messung gestartet, erhöht sich mit jedem übertragenen Element der Variablenarrays auch der Index. Dieser wird unter Berücksichtigung des Sendetaktes direkt auf den Cursor des Signalverlaufes gegeben und wandert somit während der Laufzeit zeitnah, da nicht in Echtzeit, über die Kurve.

Falls der Cursor jedoch nur für die Projektierung benutzt werden soll, kann er dafür frei geschaltet werden. Sollte er gar nicht benötigt werden, kann er ausgeschaltet werden.

6.5 Softwaremodul zur Auswertung der Simulation

Das Modul „Auswertung“ dient dem Nutzer der Software zum grafischen Betrachten der Messergebnisse. Dargestellt werden können sowohl die gesendeten als auch die empfangenen Signale.

Darüber hinaus speichert das Modul die Messergebnisse ähnlich wie bei der Projektierung in einer Textdatei, was ein späteres Betrachten oder Vergleichen mehrerer Messergebnisse ermöglicht.

6.5.1 Bedienoberfläche der Auswertung

Die Bedienoberfläche ist auf Grund der Vielzahl von einstellbaren Funktionen und Grafiken in folgende drei Reiter aufgeteilt:

a. Initialisieren (Abb. 40)

Beim Starten der Software wird immer auf diesen Reiter umgeschaltet. Im oberen linken Bereich wird der Dateiname mit den zugehörigen Headerdaten der „Inputdatei“ angezeigt. Im mittleren linken Bereich befinden sich selbige Anzeigen der „Outputdatei“, sowie die dazugehörige Schaltfläche „Werte neu einlesen“. Im unteren linken Bereich liegen Eingabeflächen zum Speichern der Messdaten für eine weitere „Outputdatei“ mit zugehöriger Bezeichnung und Kommentar, ebenso wie die Schaltfläche „Outputdatei speichern“ und einer Signalleuchte, die das Speichern der Messdaten signalisiert. Im rechten Bereich wird ausgewählt, welche Signale auf den einzelnen Kanälen der Grafiken auf den folgenden Seiten betrachtet werden sollen. Die maximale Anzahl der zu betrachtenden Signale ist 16. Dazu zählen 8 Ausgänge für die Variablenarrays (Maschine zur Karte) und 8 Eingänge für die Variablenarrays (Karte zur Maschine).

Wie auch bei der Projektierung kann zur besseren Ansicht jeder Kanal mit einem Zoomfaktor vergrößert dargestellt werden.

Ebenso wie bei der Projektierung wird in der unteren rechten Ecke der Kartentyp, der aktuell ausgewertet wird, angegeben.

 Hochschule für Angewandte Wissenschaften Hamburg Hamburg University of Applied Sciences	Diplomarbeit Fachbereich Elektrotechnik und Informatik	
Wintersemester 07/08 Abgabetermin: 18.12.07	Givian Müller Matr.Nr.: 1749822	Betreuer: Professor Dr. Ing. Müller

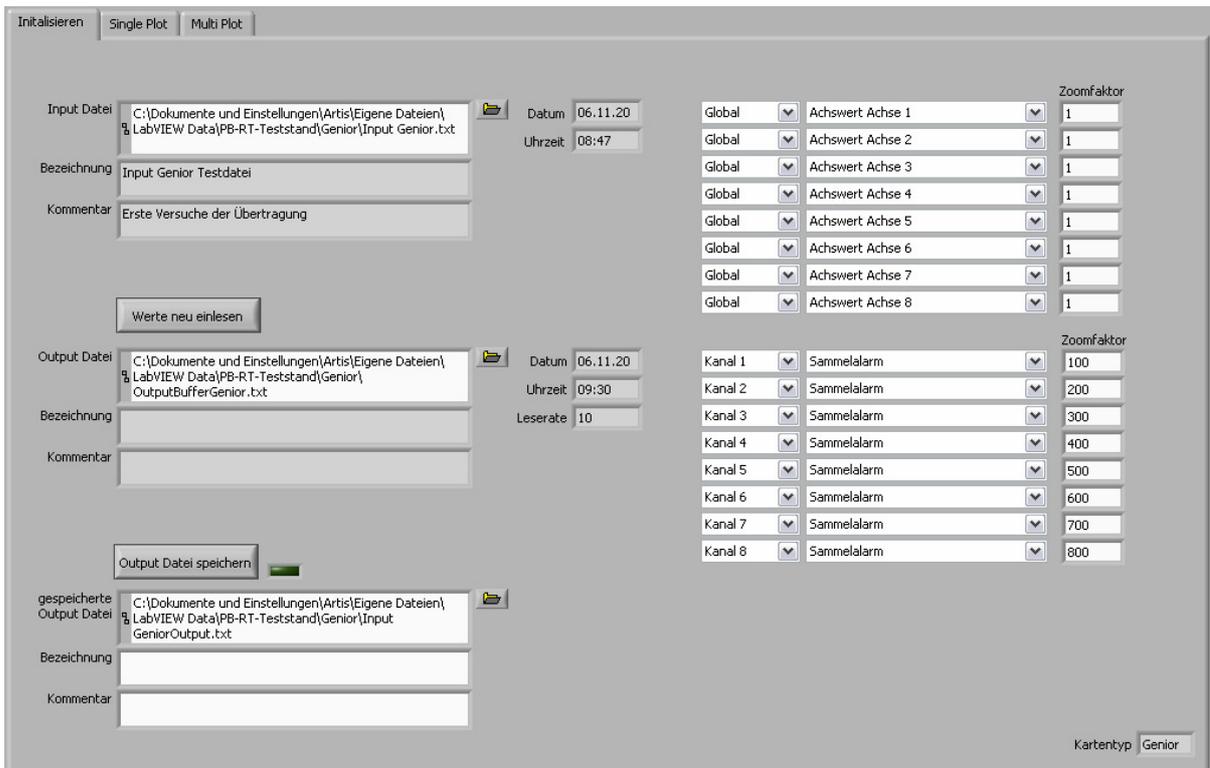
b. Single Plot (Abb. 41)

Auf dieser Seite werden alle ausgewählten Signalverläufe in einem Diagramm angezeigt. Wie schon bei der Projektierung ist die Grafik entweder frei oder automatisch skalierbar sowie der Cursor ein- und ausschaltbar. Am rechten Rand der Grafik befindet sich eine Legende, die anzeigt, welche Signale auf welchen Kanal geschaltet sind. Jeder Kanal kann mit der rechts daneben liegenden Checkbox ein- und ausgeblendet werden.

Im rechten unteren Bereich befindet sich die Bedienung zum Exportieren der Grafik. Dazu zählen das Auswahlfenster „Ablage“, die Schaltfläche „OK“ und das Eingabefeld „Dateiname“.

c. Multi Plot (Abb.42)

Ähnlich wie beim Single Plot werden auf dieser Seite die Signalverläufe dargestellt. Jedoch befinden sich hier nicht alle Signale in einer Grafik, sondern immer nur je ein Eingangssignal und ein Ausgangssignal pro Grafik. Alle Namen der Signale der acht Diagramme werden in den Legenden jeder Grafik angezeigt. Die Skalierung der Diagramme ist hier nur über die Zeitachse möglich und erfolgt über die Eingabefelder „Minimum“ und „Maximum“.



The screenshot shows a software interface with three tabs: 'Initialisieren', 'Single Plot', and 'Multi Plot'. The 'Initialisieren' tab is active. It contains several sections for configuring data input and output, and axis scaling.

Input Section:

- Input Datei: C:\Dokumente und Einstellungen\Artis\Eigene Dateien\LabVIEW Data\PB-RT-Teststand\Genior\Input Genior.txt
- Datum: 06.11.20
- Uhrzeit: 08:47
- Bezeichnung: Input Genior Testdatei
- Kommentar: Erste Versuche der Übertragung
- Buttons: Werte neu einlesen

Output Section:

- Output Datei: C:\Dokumente und Einstellungen\Artis\Eigene Dateien\LabVIEW Data\PB-RT-Teststand\Genior\OutputBufferGenior.txt
- Datum: 06.11.20
- Uhrzeit: 09:30
- Leserate: 10
- Buttons: Output Datei speichern

Axis Scaling Section:

Global	Achswert Achse	Zoomfaktor
Global	Achswert Achse 1	1
Global	Achswert Achse 2	1
Global	Achswert Achse 3	1
Global	Achswert Achse 4	1
Global	Achswert Achse 5	1
Global	Achswert Achse 6	1
Global	Achswert Achse 7	1
Global	Achswert Achse 8	1

Channel Selection Section:

Kanal	Sammelalarm	Zoomfaktor
Kanal 1	Sammelalarm	100
Kanal 2	Sammelalarm	200
Kanal 3	Sammelalarm	300
Kanal 4	Sammelalarm	400
Kanal 5	Sammelalarm	500
Kanal 6	Sammelalarm	600
Kanal 7	Sammelalarm	700
Kanal 8	Sammelalarm	800

Stored Output Section:

- gespeicherte Output Datei: C:\Dokumente und Einstellungen\Artis\Eigene Dateien\LabVIEW Data\PB-RT-Teststand\Genior\Input GeniorOutput.txt
- Bezeichnung: (empty)
- Kommentar: (empty)

Bottom right: Kartentyp Genior

Abb. 40 Bedieneroberfläche der Auswertung (Initialisieren)

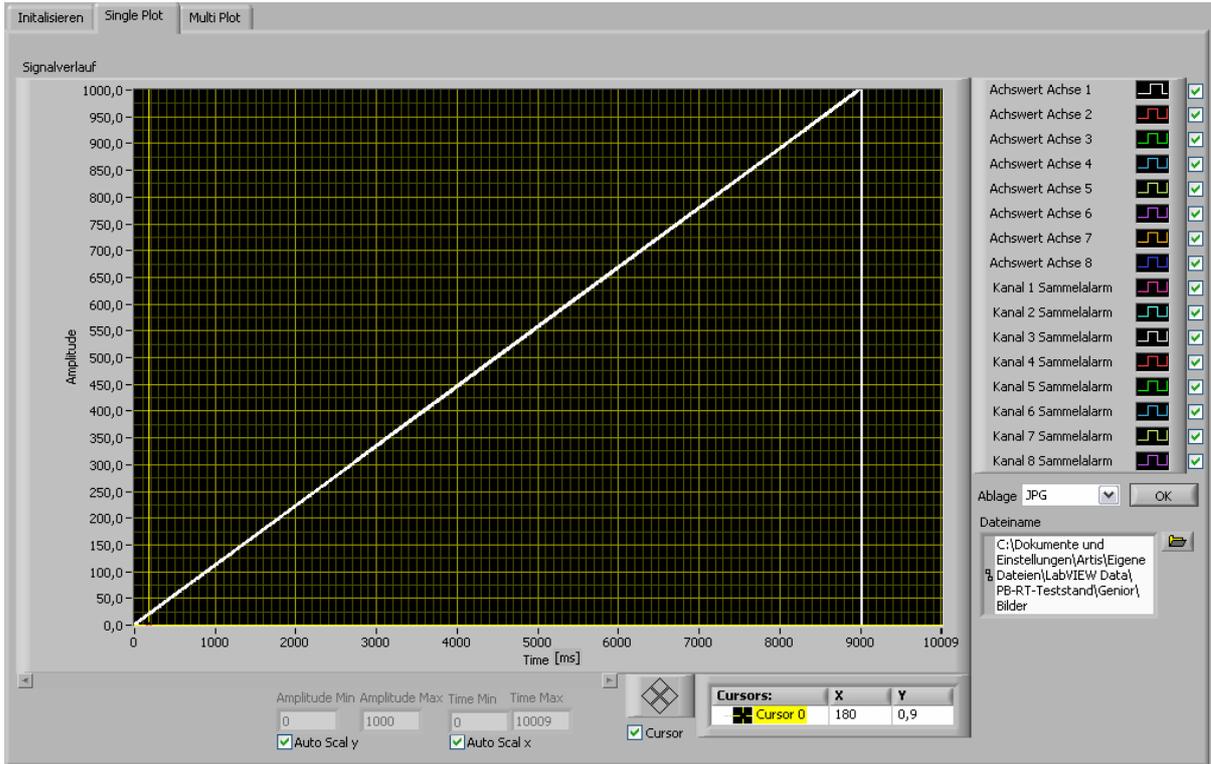


Abb. 41 Bedienoberfläche der Auswertung (Single Plot)

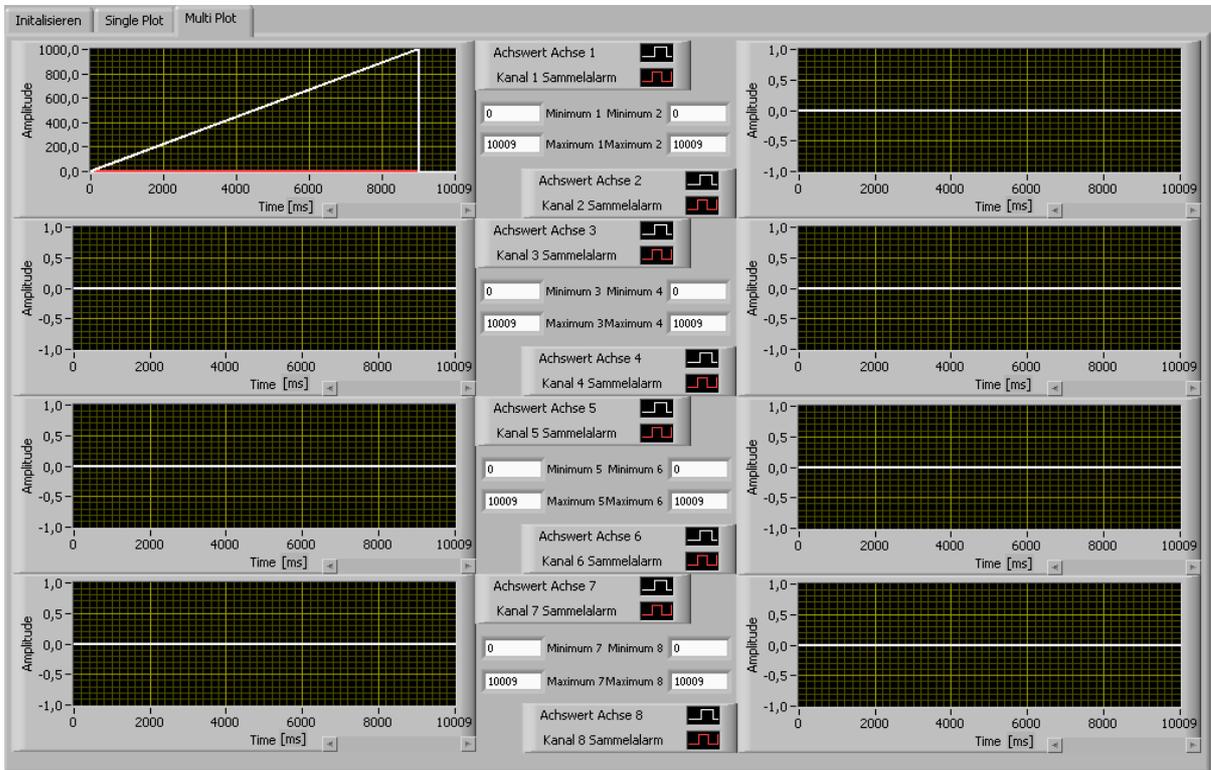


Abb. 42 Bedienoberfläche der Auswertung (Multi Plot)

 Hochschule für Angewandte Wissenschaften Hamburg Hamburg University of Applied Sciences	Diplomarbeit Fachbereich Elektrotechnik und Informatik	
Wintersemester 07/08 Abgabetermin: 18.12.07	Givian Müller Matr.Nr.: 1749822	Betreuer: Professor Dr. Ing. Müller

6.5.2 Bedienung der Auswertung

Nach dem Starten des Moduls „Auswertung“ wird, wie bei der Projektierung, der Benutzer der Software durch eine „Message Box“ aufgefordert den zu projektierenden Kartentyp zu wählen. Auch hier kann diese Einstellung während der Laufzeit nicht mehr geändert werden. Sollte der falsche Typ gewählt werden, muss das Modul erneut gestartet werden. Welcher Kartentyp gerade ausgewertet wird, erscheint namentlich im Textfeld unten rechts auf der Seite der Initialisierung.

Nach der Wahl des Kartentyps hat der Nutzer der Software entweder die Möglichkeit die Messergebnisse der letzten Messung auszuwerten oder bereits gespeicherte ältere Messungen auszuwerten. Soll die letzte Messung untersucht werden, muss der Nutzer nichts an den Einstellungen der Dateien ändern, da beim Starten des Moduls die Messergebnisse der Variablen (Karte zur Maschine) aus der letzten Messung in eine „Bufferdatei“ geschrieben werden und diese als „Outputdatei“ zugewiesen wird. Sollen diese Messergebnisse für eine spätere Auswertung erhalten bleiben, kann ein neuer Speicherpfad in „gespeicherte Output Datei“ angegeben werden, und mit der Schaltfläche „Outputdatei speichern“ gespeichert werden. Ebenso sind die Bezeichnung und der Kommentar einzutragen.

Sollte eine ältere Messung ausgewertet werden, ist die „Output Datei“ anzugeben.

Ist für diese Messung eine andere „Input Datei“ verwendet worden als die zuletzt verwendete, ist diese ebenso neu einzustellen.

Nachdem die Quelldateien ausgewählt wurden, werden im rechten Bereich der Seite „Initialisierung“ die Signale eingestellt, die betrachtet werden sollen. Dabei ist darauf zu achten, dass die Zoomfaktoren entsprechend der Signalarten einzustellen sind.

Anschließend können auf den weiteren Seiten die Signalverläufe betrachtet werden. Auf der Seite „Single Plot“ befinden sich alle eingeschalteten Signale in einer Grafik. Diese passt sich von der Skalierung des Wertebereiches der Signale an. Um spezielle Signalverläufe besser darzustellen, können alle nicht relevanten Signale ausgeblendet werden, wodurch sich bei geringerem Wertebereich die Grafik feiner skaliert. Um zeitrelevante Änderungen der Signalverläufe zu betrachten, kann die Autoskalierung ausgeschaltet werden und mit den Eingabefeldern „Amplitude Min“, „Amplitude Max“, „Time Min“ und „Time Max“ der gewünschte Wertebereich eingestellt werden.

Für den Fall, dass bei der Auswertung der Signalverläufe eine grafische Darstellung erzeugt wird, die derart interessant ist, dass sie als Dokumentation erhalten bleiben soll, besteht die Möglichkeit die Grafik zu exportieren. Dafür wird im Auswahlfeld „Ablage“ die Exportart eingestellt und mit der Schaltfläche „OK“ bestätigt. Die Grafik kann im JPEG⁴⁹ Format abgelegt werden, wofür der Speicherort unter „Dateiname“ eingegeben werden muss, als Bild im Webbrowser dargestellt, oder direkt ausgedruckt werden.

Im „Multi Plot“ ist aus Gründen der Übersichtlichkeit lediglich die Skalierung der Zeitachse freigestellt. Die Amplitude wird automatisch dem Wertebereich der eingestellten Kanäle angepasst.

⁴⁹Bildformatstandard / Joint Photographic Experts Group

 Hochschule für Angewandte Wissenschaften Hamburg Hamburg University of Applied Sciences	Diplomarbeit Fachbereich Elektrotechnik und Informatik	
Wintersemester 07/08 Abgabetermin: 18.12.07	Givian Müller Matr.Nr.: 1749822	Betreuer: Professor Dr. Ing. Müller

6.5.3 Eigenschaften der Auswertung

Im Folgenden sind alle Eigenschaften der Auswertung aufgeführt, die im Pflichtenheft gefordert wurden. Hinzu kommen alle relevanten nützlichen Eigenschaften, die die Auswertung und Sicherung der Daten erleichtern.

6.5.3.1 Auswahl des Kartentyps

Aus dem selben Grund wie bereits bei der Projektierung in Kapitel 6.4.4.1 beschrieben wurde für die Auswertung lediglich ein Modul entwickelt, welches mit beiden Kartentypen kompatibel ist. Beim Starten des Moduls „Auswertung“ wird der Nutzer der Software mit selbiger „Auswahlbox“ wie bei der Projektierung aufgefordert, den auszuwertenden Kartentyp anzugeben. Ebenso ist es während der Laufzeit nicht mehr möglich den Kartentyp zu ändern. Innerhalb des Programmablaufes werden alle kartenspezifischen Programmierungen mit Case-Abfragen auf den zur Auswertung gewählten Kartentyp geschaltet.

6.5.3.2 Einlesen der letzten Messung

Nach der Auswahl des Kartentyps, nach dem Ende einer neuen Messung oder durch Betätigen der Schaltfläche „Werte neu einlesen“ werden die Messwerte der letzten Messung, die sich in den Variablen (Karte zur Maschine) befinden, in eine „Bufferdatei“ geschrieben. Diese heißt für CTM „OutputBufferCTM.txt“ und für Genior „OutputBufferGenior.txt“. Dafür wird zuerst der Header mit leerer Bezeichnung und Kommentar und der eingestellten „Leserate“ erstellt. Die sechste Zeile des Headers bildet das Modul „Output Kopfzeile (SubVI)“. Daraufhin wird der Header in die Datei geschrieben. Abschließend werden alle Variablenarrays (Karte zur Maschine) mit dem Modul „Variablen to Output (SubVI)“ an die Daten in der Datei angehängt. Zusätzlich wird die eingestellte Quelldatei mit neuer Endung „Output.txt“ als Zielverzeichnis für die „gespeicherte Output Datei“ gewählt.

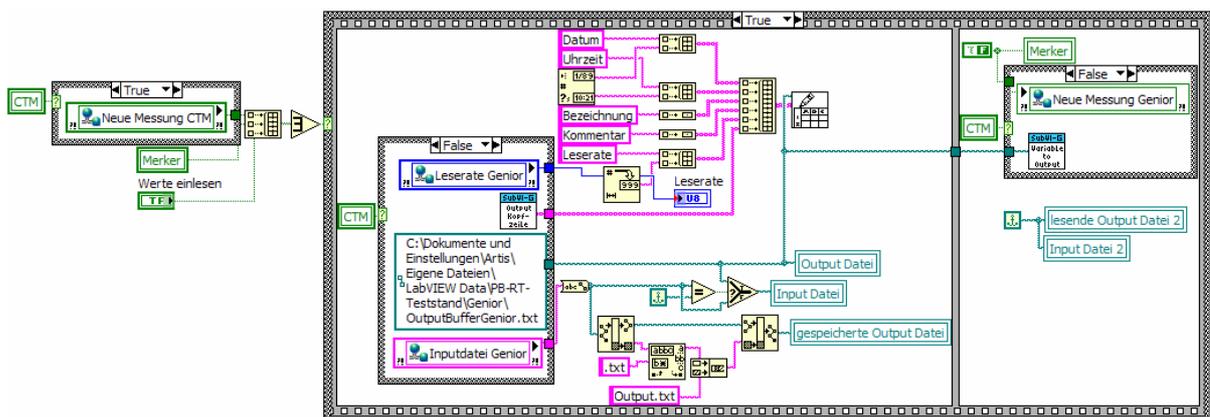


Abb. 43 Programmcode Einlesen der letzten Messung

 Hochschule für Angewandte Wissenschaften Hamburg Hamburg University of Applied Sciences	Diplomarbeit Fachbereich Elektrotechnik und Informatik	
Wintersemester 07/08 Abgabetermin: 18.12.07	Givian Müller Matr.Nr.: 1749822	Betreuer: Professor Dr. Ing. Müller

6.5.3.2.1 Modulbeschreibung Output Kopfzeile (SubVI)

Ähnlich wie beim Modul „Input Kopfzeile (SubVI)“ wird bei diesem Modul ein eindimensionales Array aus Strings mit den Namen aller Variablen (Karte zur Maschine) erstellt.

6.5.3.2.2 Modulbeschreibung Variablen to Output (SubVI)

Dieses Modul liest alle Variablenarrays (Karte zur Maschine) ein und bildet aus ihnen, unter Berücksichtigung der selbigen Reihenfolge der Kopfzeile, ein zweidimensionales Array. Anschließend wird dieses Array in die dem Modul vorgegebene Textdatei geschrieben.

6.5.3.3 Zuweisen der Datenquelle beim Starten der Auswertung

Um die versendeten Signale mit den empfangenen Signalen vergleichen zu können, müssen beide Datenquellen angegeben werden. Manuell kann jede Textdatei sowohl als „Inputdatei“ als auch als „Outputdatei“ eingelesen werden. Da im Normalfall aber die letzte Messung betrachtet werden soll, wird beim Starten des Moduls der in der Variablen „Inputdatei“ gespeicherte Name der Datei eingelesen und als Datenquelle zugewiesen.

Wie unter Punkt 6.5.3.2 beschrieben, wird als Quelle der „Outputdatei“ die zuletzt erstellte „Output Buffer Datei“ zugewiesen. Wird vor dem Starten des Moduls keine Messung durchgeführt, so dass die Variable „Inputdatei“ keinen Wert enthält, wird ebenfalls die „Output Buffer Datei“ zugewiesen. Der Grund hierfür ist, dass die Zuweisung keiner oder einer nicht existierenden Datei eine Fehlermeldung erzeugen würde. Lediglich die „Output Buffer Datei“ ist dauerhaft existent.

6.5.3.4 Messwerte in neuer Datei speichern

Da nach jeder neuen Messung die durchgeführt wird die „Output Buffer Datei“ überschrieben wird, ist es nötig Messergebnisse, die weiterhin erhalten bleiben sollen, in einer weiteren Datei abzulegen. Diese kann daraufhin jederzeit wieder eingelesen und betrachtet werden. Um die Daten zu speichern, wird der Speicherpfad im Feld „gespeicherte Output Datei“ angegeben. Zusätzlich kann die Datei mit einer Bezeichnung und einem Kommentar versehen werden. Den Befehl des Speicherns erteilt der Nutzer der Software über die Schaltfläche „Output Datei speichern“.

Sobald die Schaltfläche betätigt wird, überprüft die Software den eingegebenen Dateinamen auf die Endung „.txt“. Besteht diese Datei bereits, wird sie überschrieben, existiert diese Datei noch nicht, wird sie erstellt.

Anschließend erstellt das System, wie für die „Output Buffer Datei“, den Header und schreibt ihn in die Datei. Danach wird das zweidimensionale Signalarray aus der eingestellten

„Outputdatei“ in die neue Datei übertragen. Nachdem die neue Datei gesichert wurde, wird sie als neue „Outputdatei“ deklariert. In der Anzeige signalisiert eine grüne LED den korrekt durchgeführten Speichervorgang.

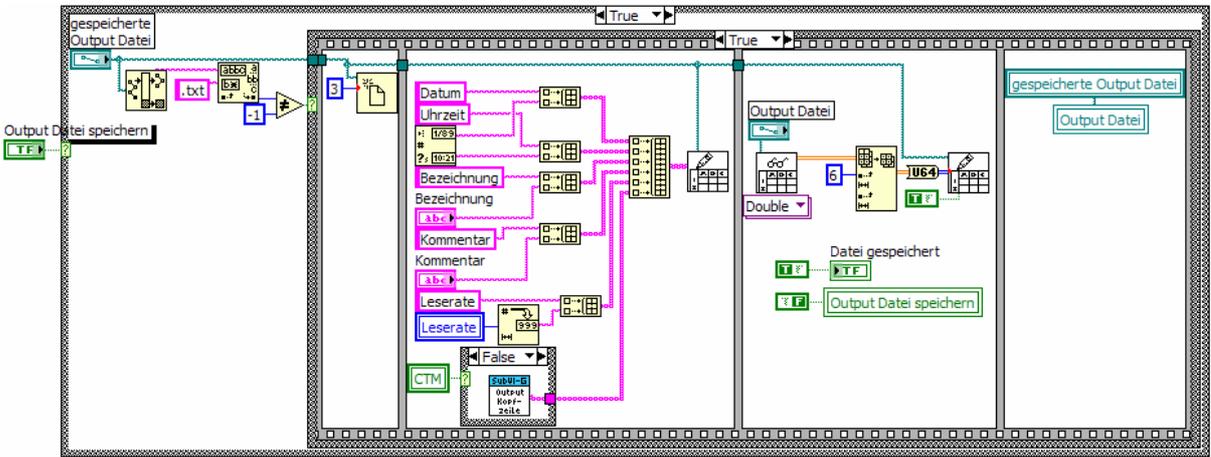


Abb. 44 Programmcode Messwerte in neuer Datei speichern

6.5.3.5 Veränderung der Datenquellen

Bei jeder Änderung der „Input Datei“ oder „Output Datei“ wird aus der neu eingestellten Datei der gesamte Inhalt gelesen und in seine Teilelemente zerlegt. Die Headerdaten werden auf die jeweiligen Anzeigen geschaltet und die Werte werden in zweidimensionale Arrays gespeichert. Darüber hinaus definiert das Wertearray der „Outputdatei“ die „Leserate“ und „Wertemenge“. Die Wertemenge wird durch die „Output Datei“ definiert, weil diese immer um den „Lesefaktor“ größer ist als die „Input Datei“.

6.5.3.6 Einstellen der darzustellenden Signale

Die Auswahl der einzelnen Signale auf den einzelnen Kanälen erfolgt wie auch schon bei der Projektierung durch die drei Variablen (Kanal, Globale Variable und Variable). Je nach eingestelltem Kartentyp werden die entsprechenden Werte für die „Inputdatei“ wieder an das Modul „Datei Inputzeile auslesen (SubVI)“ und für die „Outputdatei“ an das Modul „Datei Outputzeile auslesen (SubVI)“ übergeben. Beide Module funktionieren nach dem selben Prinzip, welches schon unter Punkt 6.4.4.9.1 beschrieben wurde.

Der Ausgang „Name“ der einzelnen Module, welcher den Namen des eingestellten Signals im String Format zurückgibt, dient zum Anzeigen der Signalnamen innerhalb der Legende der Grafiken.

 Hochschule für Angewandte Wissenschaften Hamburg Hamburg University of Applied Sciences	Diplomarbeit Fachbereich Elektrotechnik und Informatik	
Wintersemester 07/08 Abgabetermin: 18.12.07	Givian Müller Matr.Nr.: 1749822	Betreuer: Professor Dr. Ing. Müller

6.5.3.7 Analysieren von Input und Output Daten

Für die Analyse der Messung stehen die gesendeten und empfangenen Signale der ausgewählten Dateien über die gesamte Laufzeit der Messung zur Verfügung. Die beiden zweidimensionalen Arrays aus Signalwerten werden kontinuierlich, entsprechend der vorgewählten Kanäle, in Teilarrays zerlegt, mit dem Zoomfaktor multipliziert und auf die Diagramme zur Anzeige gegeben. Die Einteilung der Zeitachse erfolgt für die „Output Datei“ direkt aus dem Zeitarray aus der Datei. Für die „Input Datei“ werden die Elemente aus dem Zeitarray jedoch mit der entsprechenden Leserate multipliziert. Dadurch wird sichergestellt, dass beide Datensätze in der Anzeige den vollen Wertebereich ausnutzen und entsprechend der korrekten Zeit im Verhältnis zueinander stehen, was eine Auswertung der Werte erst ermöglicht.

Zusätzlich wird ein weiteres Array aus allen eingestellten Signalen erzeugt, welches zur Bildung des minimalen und maximalen Wertes für die automatische Skalierung dient.

6.5.3.8 Skalierung der Achsen

Bei der automatischen Skalierung der Zeitachse wird kontinuierlich das Minimum auf Null und das Maximum auf die Wertemenge gesetzt. Die Eingabeflächen dieser Werte sind über diesen Zeitraum nicht zu verstellen und werden grau dargestellt. Ebenso geschieht dieses bei der Amplitude. Lediglich der maximale und minimale Wert werden anders berechnet. Diese werden aus dem angelegten Array, beschrieben in Kapitel 6.5.3.7, ermittelt. Um zu verhindern, dass beide den selben Wert annehmen, wird in diesem Fall der Maximalwert um eins erhöht.

Bei der Umschaltung auf manuelle Skalierung werden die Eingabeflächen von Minimum und Maximum wieder aktiv. Sie können dann in den Grenzen des Wertebereiches eingegeben werden. Alle Eingaben außerhalb dieses Bereiches werden auf die Grenzwerte gesetzt. Um eine dauerhafte Abfrage der Grenzwerte zu vermeiden, wird mit Hilfe von Buffervariablen nur bei einer Änderung der Werte die Einhaltung der Grenzen überprüft.

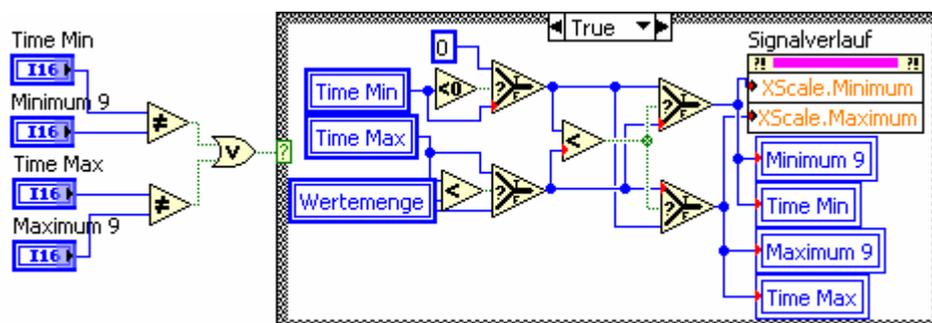


Abb. 45 Programmcode Grenzwerte der Achsskalierung überprüfen

 Hochschule für Angewandte Wissenschaften Hamburg Hamburg University of Applied Sciences	Diplomarbeit Fachbereich Elektrotechnik und Informatik	
Wintersemester 07/08 Abgabetermin: 18.12.07	Givian Müller Matr.Nr.: 1749822	Betreuer: Professor Dr. Ing. Müller

6.5.3.8.1 Alternativlösung zur Überprüfung der Grenzwerte

Denkbar wäre, für die Abfrage ob sich ein Wert für „Time Min“ oder „Time Max“ geändert hat, die Verwendung von „Eventstrukturen“, welche nur auf Änderungen von Werten reagieren. Dadurch würden keine Puffervariablen zum Vergleichen des vorherigen Wertes benötigt, wodurch sich der nötige Speicherbedarf verkleinert. Das Problem dabei ist allerdings, dass diese Strukturen nur Änderungen der Werte erfassen, die vom Nutzer vorgenommen werden und nicht solche, die vom Programm selbst durchgeführt werden. In diesem Fall würde durch das Einschalten der automatischen Skalierung der Wertebereich der Grafik nicht mehr angepasst, wodurch eine umfangreichere Änderung des Programms erforderlich wäre.

6.5.3.9 Ansteuerung des Cursors

Wie bereits bei der Projektierung kann der Cursor entweder per „drag and drop“, über die Navigationstasten unterhalb der Grafik oder mit Hilfe des Moduls „Tastendruck (SubVI)“ direkt über die Pfeiltasten des Keyboards verschoben werden.

6.5.3.10 Signalverlauf als Grafik exportieren

Beim Exportieren der Grafik wird abhängig von der Einstellung der Variablen „Ablage“ entweder ein „Standard Report“ oder ein „HTML⁵⁰ Report“ generiert. Anschließend wird die Grafik in den Report integriert und abhängig von der Variablen „Ablage“ weiterbearbeitet.

Soll die Datei als JPEG gespeichert werden, wird der Report mit dem LabVIEW Baustein „Save Report to File“ in ein Bild umgewandelt und an dem eingestellten Dateipfad mit der Endung „.jpg“ abgelegt. Leider war es an dieser Stelle nicht möglich direkt einen Namen für das Bild vorzugeben, da der Baustein den Namen direkt aus dem Report generiert. Um dieses zu ermöglichen, müsste der Baustein selbst neu geschrieben werden, was jedoch in keinem Verhältnis von Leistung zu Nutzen stehen würde.

Wenn die Grafik als Webpage geöffnet werden soll, wird der Report mit dem Baustein „Open HTML Report in Browser“ umgewandelt. In diesem Fall öffnet sich direkt nach Betätigen der Schaltfläche „OK“ eine Internetseite mit dem eingestellten „Web Browser“, die die exportierte Grafik anzeigt. Diese kann dann beispielsweise mit dem Programm Frontpage weiter verwendet werden.

Soll die Datei gedruckt werden, wird sie mit dem Baustein „Print Report“ an den Default Drucker des Rechners gesendet.

⁵⁰ Hypertext Markup Language / Hypertext-Auszeichnungssprache

 Hochschule für Angewandte Wissenschaften Hamburg Hamburg University of Applied Sciences	Diplomarbeit Fachbereich Elektrotechnik und Informatik	
Wintersemester 07/08 Abgabetermin: 18.12.07	Givian Müller Matr.Nr.: 1749822	Betreuer: Professor Dr. Ing. Müller

7 Testphase der Hard- und Software

Funktionstests wurden im Laufe des Projektes des Öfteren durchgeführt, um auftretende Fehler direkt beheben zu können. Im Folgenden sind das Testen der Echtzeitfähigkeit, sowie mehrere Testdurchläufe beschrieben, bei denen die Performance der Echtzeitumgebung, die korrekte Funktion der Module, sowie das Abfangen fehlerhafter Bedienungen simuliert wurde.

7.1 Testen der Echtzeit

Um die Echtzeitfähigkeit des aufgebauten Systems zu überprüfen, müssen die Signale auf dem Profibus analysiert werden. Um die Profibussignale, die anschließend mit einem Oszilloskop betrachtet werden sollen, in digitale Signale umzuwandeln, wird eine zusätzliche I / O Baugruppe benötigt. Für diesen Test wurde eine „ET 200 B - 16 DO⁵¹“ Baugruppe der Firma Siemens verwendet. Diese verfügt über 2 Bytes am Ausgang, die als 16 einzelne Bits digital ausgegeben werden.

Zum betreiben der Baugruppe, muss diese mit einer Versorgungsspannung von 24 V⁵² versorgt, und in den Bus integriert, werden. Anschließend wird, wie bei der CTM- und Genior Karte, die entsprechende GSD Datei, wie unter Kapitel 4.2.2 beschrieben, mit dem „Configurator 2“ eingebunden und mit einer Profibusadresse versehen. Die Adresse dieses Teilnehmers ist die 6. Nach dem Einbinden werden die neuen Konfigurationen auf das Echtzeitsystem übertragen und die Verbindung zur neuen Baugruppe überprüft. Bei korrekter Verbindung kann mit der Programmierung der Ausgänge begonnen werden.

7.1.1 Software zum testen der Echtzeit

Zum Überprüfen der Echtzeit, bei einer geforderten Zykluszeit von 10 ms, benötigt das Oszilloskop bei jedem Zyklustakt einen erneuten Triggerimpuls. Das Modul „Echtzeittest.vi“ erzeugt dafür innerhalb einer While- Schleife, die im 10 ms Takt läuft, abwechselnd ein True- und ein False Signal auf zwei Bits des Ausganges. Dieses geschieht durch Rotieren des Wertearrays.

⁵¹ Digital Output

⁵² Volt

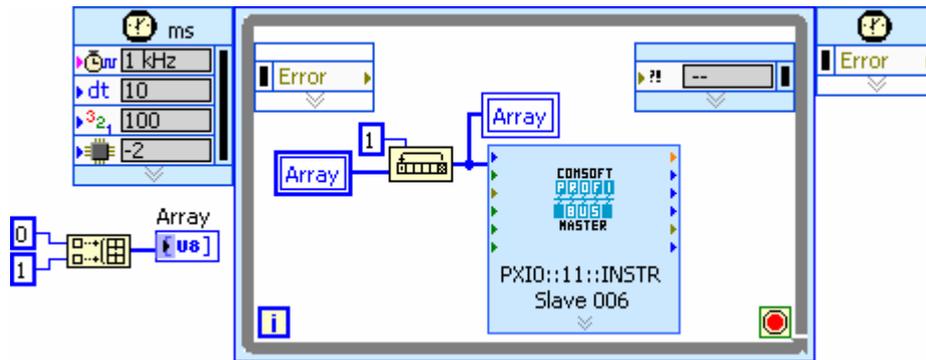


Abb. 46 Programmcode Echtzeitest

7.1.2 Oszilloskopauswertung der Echtzeit

Während das Modul „Echtzeitest.vi“ auf dem Echtzeitsystem läuft, wird einer der geschalteten Kanäle der Siemens Baugruppe auf ein Oszilloskop des Typs „TPS 2024“ von Tektronix geschaltet.

Festzustellen ist dabei, dass die geforderte Zykluszeit von 10 ms kontinuierlich eingehalten wird. Bei selbigem Test mit unterschiedlichen Zykluszeiten wurden Kurvenverläufe mit selbiger Genauigkeit aufgenommen.

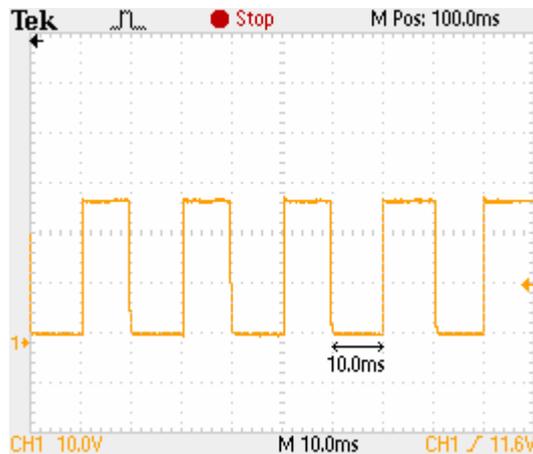


Abb. 47 Oszilloskopauswertung der Echtzeit

7.2 Testen der minimalen Laufzeit

Im Laufe des Projektes war es mehrfach nötig, die Durchlaufzeit, die im Kapitel 1.2 beschrieben wurde, zu testen. Dabei ergaben sich, um die Leistungsfähigkeit weiter zu steigern, fortlaufend Änderungen. Die wesentlichste Änderung im Modul „Target“ ist die Auslagerung des Variablenzugriffes aus der echtzeitfähigen While-Schleife. In den folgenden Kapiteln ist die Durchführung der Tests vor und nach den Änderungen beschrieben.

 Hochschule für Angewandte Wissenschaften Hamburg Hamburg University of Applied Sciences	Diplomarbeit Fachbereich Elektrotechnik und Informatik	
Wintersemester 07/08 Abgabetermin: 18.12.07	Givian Müller Matr.Nr.: 1749822	Betreuer: Professor Dr. Ing. Müller

7.2.1 Laufzeittest mit kontinuierlicher Kommunikation

Um festzustellen welche minimale Laufzeit das Echtzeitsystem leisten kann, wurde die Laufzeit der zeitgesteuerten While-Schleife im Modul „Target“ auf 1 ms herabgesetzt. Dadurch ist es möglich, weiterhin einen bereits projektierten Zeitverlauf zu verwenden, wobei lediglich die Zeit auf ein zehntel der eigentlichen Laufzeit reduziert wird. Die zuvor projektierte Kurve von 20 s sollte dadurch auf 2 s gestaucht werden.

Beim ersten Test dieser Art mit der Genior Karte ergab sich jedoch eine Laufzeit von 12 s, was darauf schließen lässt, dass die minimal einzuhaltende Laufzeit des Echtzeitsystems um den Faktor sechs größer ist. Das bedeutet, sie würde 6 ms betragen. Daraufhin wurden als Laufzeit für die While-Schleife sowohl 5 ms als auch 6 ms eingestellt. Dabei war festzustellen, dass bei 5 ms, mit einer gewünschten Laufzeit von 10 s, eine tatsächliche Laufzeit von etwa 11 s entsteht, während bei 6 ms die erwarteten 12 s eingehalten werden. Nun ist noch zu klären, ob die Verzögerung durch das Echtzeitsystem oder vielleicht durch die zu messende Karte selbst entsteht.

Dass die Verzögerung am Echtzeitsystem liegt, ist an mehreren Punkten ersichtlich. Zum einen ist bei der Messung mit 1 ms Laufzeit die Anzeige der Auslastung des Echtzeitsystems im Vollanschlag auf 100 %. Zum anderen durchläuft das Echtzeitsystem vom Zeitpunkt des Betätigens des Starttasters bis zum Wiedererscheinen der Schaltfläche exakt dieselbe Zeit wie von der Genior Karte ermittelt. Ebenso durchläuft der Cursor aus dem Modul „Projektierung“ die Grafik des Zeitverlaufes innerhalb selbiger Zeit.

Gleicher Test mit der CTM Karte ergab eine Laufzeit von 4 bis 5 ms. Daher ist neben der Echtzeitumgebung selbst der Umfang des Programmcodes der zweite wesentliche Aspekt der Laufzeit. Der einzige Unterschied zwischen CTM und Genior an dieser Stelle der Software ist die Tatsache, dass Genior Signale von acht Kanälen verarbeiten muss, während CTM nur vier Kanäle verarbeitet.

7.2.2 Laufzeittest bei vorheriger Initialisierung

Nachdem die Änderungen des Moduls „Target“ vorgenommen wurden, wodurch die Variablenarrays vor der Messung eingelesen werden, um das Wertearray zu bilden, wurde der Test ähnlich wie im Kapitel 7.2.1 beschrieben erneut durchgeführt. Mittlerweile wurde die Programmierung der Software derart angepasst, dass es auch möglich ist, den Sendetakt direkt im Modul „Projektierung“ vorzugeben, was die Messdurchführung wesentlich vereinfacht.

Festzustellen ist, dass bei einer eingestellten Laufzeit von 10 ms eine Auslastung der Echtzeitumgebung bei 4 % für CTM und 7 % für Genior liegt. Auch die mit den Karten empfangenen Signalverläufe stimmten zeitlich mit den gesendeten überein. Da hier bereits eine erhebliche Verbesserung festzustellen ist, wurde derselbe Kurvenverlauf mit einer Laufzeit von 1 ms erneut gesendet. Auch dieser Verlauf kam mit einer Systemauslastung von 30 % für CTM und 58 % für Genior zeitgenau bei den Karten an.

 Hochschule für Angewandte Wissenschaften Hamburg Hamburg University of Applied Sciences	Diplomarbeit Fachbereich Elektrotechnik und Informatik	
Wintersemester 07/08 Abgabetermin: 18.12.07	Givian Müller Matr.Nr.: 1749822	Betreuer: Professor Dr. Ing. Müller

Wie erwartet ist die minimale Laufzeit des Gesamtsystems von der Genior Karte abhängig, was einzig und allein an der Anzahl der Kanäle liegt. Durch weitere Verringerungen der Laufzeit wurde der Grenzwert von etwa 700 us ermittelt. Bei diesem Wert ist eine korrekte, zeitgenaue Ankunft auf beiden Karten garantiert. Die Auslastung des Systems liegt dabei bei 46 % für CTM und 80% für Genior.

Denkbar wäre dadurch die Eingabe eines geringeren Sendetaktes im Modul „Projektierung“ zu verhindern, jedoch ist die minimale Laufzeit auch von der Hardwareeigenschaft des Echtzeitsystems abhängig und somit von System zu System unterschiedlich. Da die Software später auf mehreren Systemen in Betrieb genommen werden soll, wird auf eine derartige Verriegelung verzichtet, so dass der Nutzer der Software diese Grenze selbst zu beachten hat.

7.3 Abschließender Funktionstest

Um das korrekte Verhalten der Module und alle Funktionalitäten aus der Profibus Master Teststand Software zu analysieren, wird der komplette Ablauf einer Messung mit allen Einstellmöglichkeiten durchlaufen. Da der Ablauf für beide Karten identisch ist, wird am Beispiel der CTM Karte eine Messung durchgeführt.

7.3.1 Funktionstest des Projektieren eines Kurvenverlaufes

Zunächst wird der richtige Kartentyp CTM gewählt, da eine Auswahl des falschen Kartentyps durch die Verwendung getrennter Variablenarrays keine Fehlfunktion, sondern lediglich keine Funktion der Karte zur Folge hat.

Als nächstes stellt man den „Lese Pfad“ ein, da vorher keine Projektierung möglich ist. Bei der Wahl eines „Lese Pfades“ darf jedes unter Windows gängige Verzeichniszeichen verwendet werden. Beim Weglassen der Endung der Datei wird diese vom Programm automatisch ergänzt. Das bedeutet, selbst eine ausgewählte Datei im falschen Format (z.B. JPEG) wird mit der Endung [.txt] neu erstellt.

Anschließend werden die Bezeichnung und der Kommentar ergänzt. In diesen beiden Feldern kann eine beliebige Zeichenfolge in beliebiger Länge eingetragen werden.

Zum Kopieren der Datei muss der Zielpfad angegeben sein, sonst ist die Schaltfläche „Kopie erstellen“ nicht aktiv. Ebenso wie beim „Lese Pfad“ wird eine fehlende Endung [.txt] ergänzt. Nach dem Erstellen der Kopie ist diese wie gewünscht der neue „Lese Pfad“.

Die Veränderung der Laufzeit durch den Sendetakt und der Wertemenge kann ohne Einschränkungen erfolgen. Zu beachten ist lediglich, dass die minimale Laufzeit nicht unterschritten wird.

Für die Einstellung der Leserate können nur ganzzahlige Vielfache eingestellt werden. Das minimale Verhältnis vom Sendetakt zum Lesetakt ist 1 / 1. Für den Funktionstest an der Karte werden ein Sendetakt von 10000 us, eine Wertemenge von 2000 und eine Leserate von 10 eingestellt. Daraus ergibt sich eine Laufzeit von 20 s mit einem Sende- zu Leseverhältnis von 1:10.

 Hochschule für Angewandte Wissenschaften Hamburg Hamburg University of Applied Sciences	Diplomarbeit Fachbereich Elektrotechnik und Informatik	
Wintersemester 07/08 Abgabetermin: 18.12.07	Givian Müller Matr.Nr.: 1749822	Betreuer: Professor Dr. Ing. Müller

Als Kurvenverlauf für den Achswert 1 werden sämtliche Funktionen verwendet, die durch die Software ermöglicht werden. Nacheinander werden folgende Signale generiert: Im Wertebereich von 1000 bis 4990 wird eine Rampe mit Offset und kontrolliertem Rauschen erzeugt. Im Wertebereich von 6000 bis 11990 wird eine Kurve erzeugt. Im Wertebereich von 14000 bis 15990 und 17000 bis 18990 wird ein Rechtecksignal mit wildem Rauschen erzeugt. Der dazwischen liegende Bereich erhält denselben Signalverlauf, allerdings mit negativer Amplitude. Derartige Signalverläufe lassen sich nur für Achswerte projektieren. Das Vorschubsignal wird entsprechend dieser Signalfolge auf den Wert Eins gesetzt und anschließend mit der Funktion „Kanal 1 für alle Kanäle“ vervielfältigt. Um dieses Signal im Vergleich zum Achswert besser darstellen zu können, wird der Zoomfaktor auf 100 gesetzt. Für die Eingabe des Zoomfaktors ist auch jeder ganzzahliger Wert größer Null denkbar. Das Ausschalten der automatischen Skalierung, sowie das Ändern des sichtbaren Wertebereiches funktionieren einwandfrei. Jede andere Eingabe für die minimalen und maximalen Werte, wie auch für die Zeiten ist nicht möglich. Ebenso funktioniert das Freigeben, Ausschalten, sowie das Bewegen des Cursors einwandfrei. Nachdem alle Funktionen zur Projektierung getestet wurden, kann mit der Messung begonnen werden. Als Messanzahl kann dafür jeder Wert eingetragen werden. Für den Test werden fünf Messungen durchgeführt. Zur Überwachung der Messung wird der Cursor wieder auf Überwachung geschaltet und die vier Module „Simulation Global M-K“, „Simulation Kanäle M-K“, „Simulation Global K-M“ und „Simulation Kanäle K-M“ zur Onlinebetrachtung gestartet.

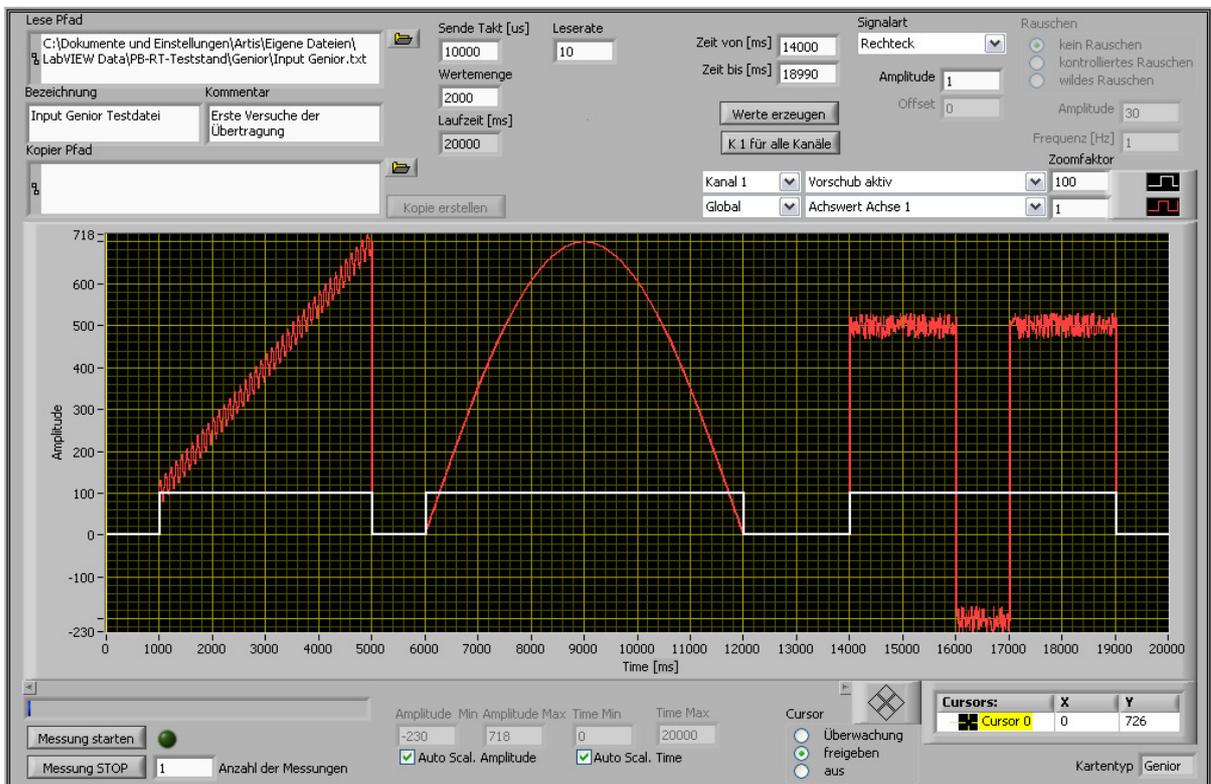


Abb. 48 Funktionstest der Projektierung des Kurvenverlaufes

 Hochschule für Angewandte Wissenschaften Hamburg Hamburg University of Applied Sciences	Diplomarbeit Fachbereich Elektrotechnik und Informatik	
Wintersemester 07/08 Abgabetermin: 18.12.07	Givian Müller Matr.Nr.: 1749822	Betreuer: Professor Dr. Ing. Müller

7.3.2 Funktionstest der Messdurchführung

Um die Messung zu starten wird im Modul „Projektierung“ die Schaltfläche „Messung starten“ betätigt. Daraufhin beginnt der Ladebalken den Fortschritt der Projektierung auf dem Echtzeitsystem anzuzeigen. Die Auslastung des Echtzeitsystems liegt dabei bei etwa 89 %. Sobald dieser Vorgang abgeschlossen ist, beginnt der Cursor über die Grafik zu laufen und signalisiert den aktuellen übertragenen Wert des Vorschubsignals.

In den vier Modulen der Onlinebetrachtung werden alle Werte angezeigt die zu dem entsprechenden Zeitpunkt gesendet und empfangen werden.

Nachdem die erste Messung durchlaufen wurde, reduziert sich die Anzahl der Messungen wie erwartet auf 4 und erneut wird der Signalverlauf auf dem Echtzeitsystem projiziert. Dieser Vorgang wiederholt sich exakt fünf Mal, bis die Anzahl der Messung auf eins reduziert wird. Wird während der Messung die Schaltfläche „Messung STOP“ betätigt, wird keine weitere Messung durchgeführt. Bei der Eingabe 0 als Anzahl der Messungen wird die Messung dauerhaft bis zum Betätigen der Schaltfläche „Messung STOP“ durchlaufen.

7.3.3 Funktionstest der Signalauswertung

Um die Auswertung einer vollständigen Funktionsprüfung zu unterziehen, muss ein anderer Zeitverlauf projiziert werden, als er es für den Test des Moduls „Projektierung“ nötig war. Welcher Zeitverlauf benötigt wird, ist im folgenden Kapitel dargestellt.

7.3.3.1 Vorbereitungen für den Funktionstest der Auswertung

Simuliert werden soll ein sich wiederholendes Vorschubsignal, welches zunächst von der Karte eingelernt und anschließend wieder erkannt werden soll. Anschließend soll ein auftretender Werkzeugbruch innerhalb des Verlaufes festgestellt werden. Als Wertebereich wird wieder 2000 gewählt mit einem Sendetakt von 10 ms und dem zehnfachen Lesetakt. Die drei Signalblöcke erhalten je eine Länge von fünf Sekunden und einer Amplitude von 1000. Beim Bruch erhöht sich der Achswert um 50 % auf 1500. Um von der Karte einen Bruch gemeldet zu bekommen, muss die Bruchgrenze auf unter 50 % eingestellt werden. Für die Messung wird diese auf 49 % gesetzt. Die Grundeinstellungen der Bruchgrenzen der Karten liegen bei 200 %.

Zusätzlich zum Achswert wird der Vorschubwert über denselben Zeitverlauf aller drei Signalblöcke auf 1 gesetzt. Das Lernbit wird lediglich beim ersten Block auf den Wert 1 eingestellt.

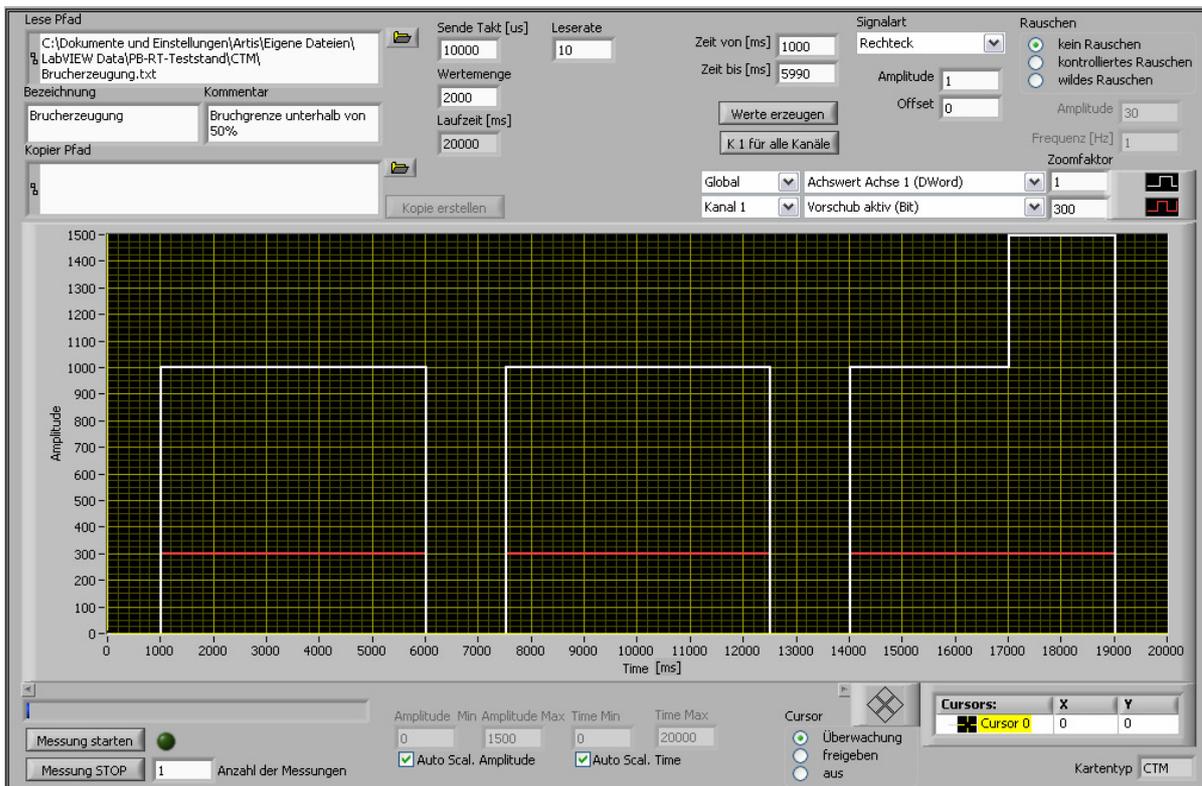


Abb. 49 Signalverlauf zum Funktionstest der Auswertung

7.3.3.1.1 Betrachtung der vorbereiteten Signale auf dem Zielsystem

Die CTM Karte im Zielsystem erkennt alle drei Signalblöcke einwandfrei. Dadurch, dass das Vorschubsignal in den Pausen unterbrochen wird, erkennt die Karte ein neu eintreffendes Vorschubsignal und startet die Visualisierung der Grafik jedes Mal neu. In den folgenden Grafiken sind alle drei Signalblöcke einzeln dargestellt.

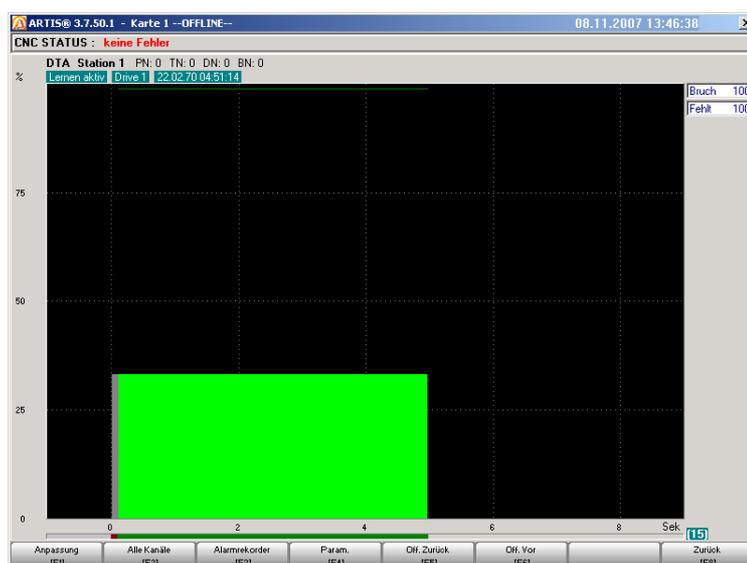


Abb. 50 Lernkurve auf CTM

 Hochschule für Angewandte Wissenschaften Hamburg Hamburg University of Applied Sciences	Diplomarbeit Fachbereich Elektrotechnik und Informatik	 ARTIS ®
Wintersemester 07/08 Abgabetermin: 18.12.07	Givian Müller Matr.Nr.: 1749822	Betreuer: Professor Dr. Ing. Müller

Zunächst wird der Kurvenverlauf durch das gesetzte Bit „Lernen“ eingelernt. Der Maximalwert der Kurve wird dabei immer auf ein Drittel der gesamten Anzeige gebracht. Die gelernte Kurve wird als grüne Fläche dargestellt und erscheint anschließend bei jeder weiteren Messung wiederum im Hintergrund.

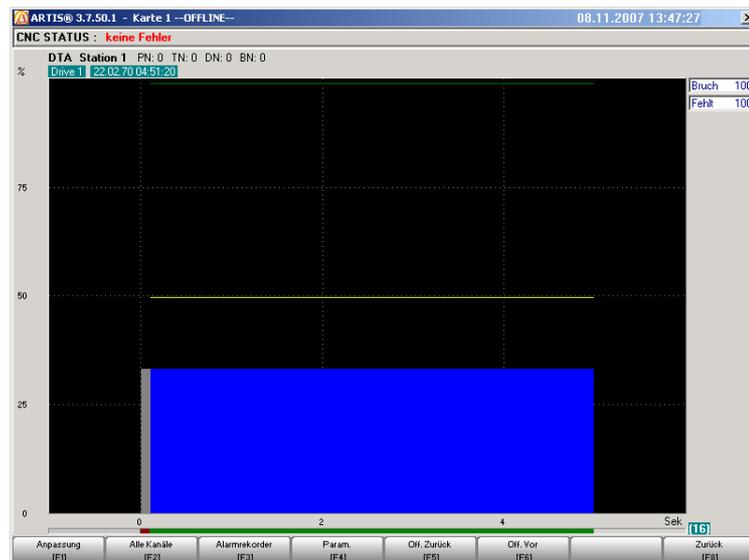


Abb. 51 Wiederholter Kurvenverlauf auf CTM

Beim Durchlauf des zweiten Achswertabschnittes ist das Lernbit ausgeschaltet, so dass keine neue Lernkurve mehr aufgenommen wird. Die projizierte Kurve wird in der Darstellung als blaue Fläche dargestellt. Bei jeder auftretenden Abweichung zur eingelernten Kurve würde entweder bei einem neuen, kleineren Wert die grüne Kurve sichtbar oder bei einem größeren Wert der Überhang als rote Fläche angezeigt. Festzustellen ist bei der aufgenommenen Kurve, dass sie exakt mit der gelernten übereinstimmt. Das ist die grundlegende Voraussetzung um reproduzierbare Messungen zu simulieren. Keine in der Firma verfügbare Maschine wäre sonst in der Lage derartige reproduzierbare Messergebnisse zu liefern.

Die Einteilung der Zeitachse wurde für die neue Kurve optimiert dargestellt, da im Gegensatz zur Lernkurve die Zeitbasis bereits bekannt ist. Ebenso wird die eingestellte, obere Bruchgrenze als gelbe Linie oberhalb des Kurvenverlaufes eingeblendet. Eine Überschreitung dieser Linie hat einen Alarm zur Folge.

Der Durchlauf des dritten Achswertabschnittes beginnt genau wie der des zweiten, bis zu dem Zeitpunkt des simulierten Bruches. An dieser Stelle wird die Abweichung vom gelernten Kurvenverlauf rot dargestellt. Da an dieser Stelle die eingestellte Bruchgrenze überschritten wird, wird in der Grafik eine durchgehende grüne Meldelinie eingeblendet. Zu diesem Zeitpunkt sendet die Karte schnellstmöglich den erkannten Alarm über die Profibusverbindung zum Echtzeitsystem. In der Software der CTM Karte ist es möglich dieses Senden zu verzögern, da dies bei bestimmten Maschinen erforderlich sein kann.

 Hochschule für Angewandte Wissenschaften Hamburg Hamburg University of Applied Sciences	Diplomarbeit Fachbereich Elektrotechnik und Informatik	
Wintersemester 07/08 Abgabetermin: 18.12.07	Givian Müller Matr.Nr.: 1749822	Betreuer: Professor Dr. Ing. Müller



Abb. 52 Simulierter Bruch auf CTM

7.3.3.2 Starten der Auswertung ohne vorherige Messung

Wenn vor dem Starten des Moduls „Auswertung“ noch keine Messung durchgeführt wurde, ist dem Modul keine „Inputdatei“ zugewiesen. In diesem Fall ist die „Output Buffer Datei“ als Verzeichnis für Input und Output zugewiesen.

7.3.3.3 Starten der Auswertung mit vorheriger Messung

Wird das Modul nach einer Messung gestartet oder wird eine neue Messung nach dem Starten des Moduls beendet, wird automatisch die „Inputdatei“ neu zugewiesen, sowie die „Output Buffer Datei“ überschrieben und neu eingelesen. Die Headerdaten beider Dateien werden dabei fehlerfrei eingetragen. Wie erwartet wird dem Eingabefeld „gespeicherte Output Datei“ der Name der „Inputdatei“ mit angehängtem Namen „Output“ zugewiesen. Mit der Schaltfläche „Output Datei speichern“ wird eine Kopie der „Output Buffer Datei“ angelegt. Eine Änderung der Dateien für „Input Datei“ oder „Output Datei“ bewirkt umgehend das erneute Einlesen der Werte aus diesen Dateien.

Um nach einer Messung möglichst ohne viele Eingaben direkt die wichtigsten Signale zu betrachten, werden als Eingangssignale die Achswerte und als Ausgangssignale die wichtigsten Alarme wie Bruchalarme und Sammelalarme dargestellt. Ebenso stellen sich die Zoomfaktoren der binären Signale der „Output Datei“ zum besseren Vergleich auf verschiedene Werte in Hunderterschritten ein.

Im „Multi Plot“ werden die acht Grafiken für alle 16 Signale automatisch dem Maximalwert der Kurve angepasst. Die Umskalierung der Zeitachse ist mit den jeweiligen minimalen und maximalen Werten innerhalb des Wertebereiches möglich. Eine genauere Betrachtung ist allerdings nur im „Single Plot“ möglich. Hier funktioniert die automatische und auch manuelle Skalierung wie bei der Projektierung einwandfrei. Alle nicht verwendeten Kanäle können wie gewünscht ausgeschaltet werden.

 Hochschule für Angewandte Wissenschaften Hamburg Hamburg University of Applied Sciences	Diplomarbeit Fachbereich Elektrotechnik und Informatik	
Wintersemester 07/08 Abgabetermin: 18.12.07	Givian Müller Matr.Nr.: 1749822	Betreuer: Professor Dr. Ing. Müller

Festzustellen ist bei dieser Messung, dass der Bruchalarm des ersten Kanals bei dem dritten Signalblock ausgelöst wird. Durch Vergrößern der Auflösung wird sichtbar, dass es zu einer Zeitverzögerung vom Auftreten des Bruches bis zum Alarm von 17 ms kommt.

Diese Feststellung beweist zwei grundlegende Dinge. Zum einen funktioniert die zehnfach höhere Leserete im Vergleich zum Sendetakt, da eine Abtastung im Millisekundenbereich möglich ist, sowie die synchrone Darstellung beider Verläufe über die Zeit. Zum anderen erkennt die Karte den erzeugten Bruch und setzt umgehend den Bruchalarm. Mehrere Wiederholungen dieser Messung ergaben allerdings unterschiedliche Antwortzeiten. Wodurch diese entstehen wird im folgenden Kapitel beschrieben.

Nachdem die Auswertung abgeschlossen ist, wird die Speicherfunktion getestet. Wie beschrieben wird das Bild entweder als JPEG im angegebenen Verzeichnis gespeichert, als Webpage ausgegeben oder ausgedruckt. Falls kein Drucker eingerichtet sein sollte, erzeugt das Programm eine Fehlermeldung, die nicht abgefangen werden kann.

7.3.4 Performancetest von CTM und Genior

Für eine Testreihe an CTM und Genior, wie sie mit der Software in Zukunft mehrfach durchgeführt werden soll, wird erneut der projektierte Kurvenverlauf aus dem Kapitel 7.3.3.1 verwendet. Getestet wird der erzeugte Werkzeugbruch für beide Kartentypen sowohl für einen Kanal als auch für alle verfügbaren Kanäle. Zusätzlich wurden beide Kurventypen bei Genior mit ausgeschalteter BAV⁵³ und eingeschalteter BAV von 0 ms getestet. Erwartet werden dabei immer dieselben Ergebnisse. Resultierend aus diesen Messungen ergaben sich bei zehn Wiederholungen jedoch folgende Reaktionszeiten der Bruchsignale:

CTM 1 Kanal:	13 – 23 ms
CTM 4 Kanäle:	13 – 23 ms
Genior 1 Kanal, BAV aus:	10 – 11 ms
Genior 1 Kanal, BAV ein:	16 – 26 ms
Genior 8 Kanäle, BAV aus:	17 – 21 ms für Kanal 1 7 – 11 ms für Kanal 8
Genior 8 Kanäle, BAV ein:	23 – 35 ms für Kanal 1 14 – 25 ms für Kanal 8

⁵³ Bruch Alarm Verzögerung

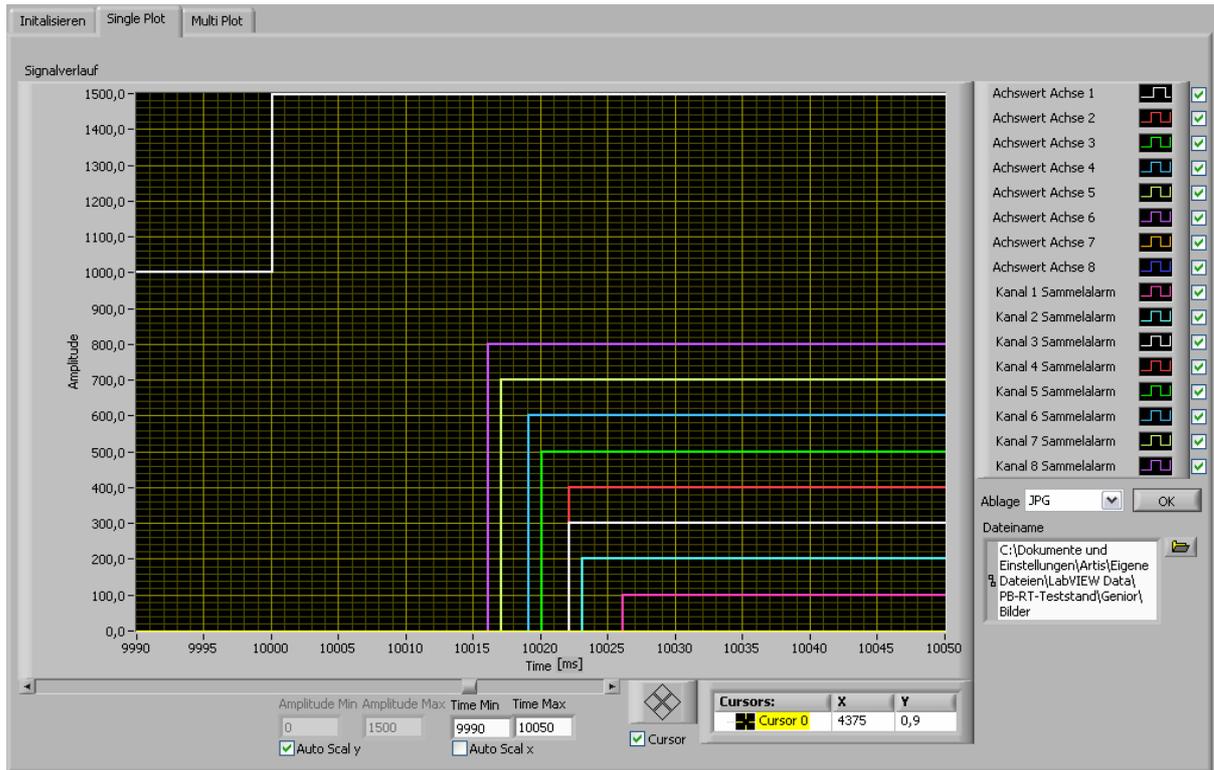


Abb. 53 Performance Genior 8 Kanäle, BAV ein, unsynchron zum Vorschub

Bei dieser Messreihe ist festzustellen, dass die CTM Software unabhängig von der Kanalanzahl agiert. Alle empfangenen Bruchalarme wurden synchron nach derselben Zeit gesendet. Durch die Abtastrate von 10 ms der Karte ist auf eine Reaktionszeit von 13 ms zu schließen.

Die Genior Software hingegen funktioniert nicht kanalsynchron. Sie arbeitet zwar bei einem eingestellten Kanal schneller als die CTM Karte, nämlich mit 10 ms, ist aber bei allen acht Kanälen auf dem ersten Kanal mit 17 ms wesentlich langsamer. Merkwürdig ist dabei, dass die Alarme in umgekehrter Reihenfolge (Kanal 8 bis Kanal 1) gesendet werden. Ebenso ungewöhnlich ist die Tatsache, dass Kanal 8 mit ermittelten 7 ms schneller ist, als bei nur einem eingestellten Kanal.

Die unterschiedlichen Zeiten zwischen ein- und ausgeschaltetem BAV resultieren aus den unterschiedlichen Programmroutinen auf der Karte. Gewünscht ist auch hier ein synchrones Verhalten.

7.3.4.1 Spezialfall im Performancetest

Nach Rücksprache mit einigen Mitarbeitern kommt es bei beiden Karten unter ganz bestimmten Umständen zu einer viel größeren Alarmverzögerung. Dies ist der Fall, wenn ein Bruch zum selben Zeitpunkt wie ein neues Vorschubsignal auftritt. Um dieses Problem zu analysieren wurde der Kurvenverlauf in der Art verändert, dass der gesamte dritte Achswertblock auf 150 % erhöht wurde. Dadurch wird der Bruch synchron zum Beginn des Vorschubsignals simuliert.

 Hochschule für Angewandte Wissenschaften Hamburg Hamburg University of Applied Sciences	Diplomarbeit Fachbereich Elektrotechnik und Informatik	
Wintersemester 07/08 Abgabetermin: 18.12.07	Givian Müller Matr.Nr.: 1749822	Betreuer: Professor Dr. Ing. Müller

Aus dieser Messreihe ergaben sich bei wiederum zehn Wiederholungen der Messung folgende Reaktionszeiten der Bruchsignale:

CTM 1 Kanal:	137 – 147 ms
CTM 4 Kanäle:	175 – 185 ms
Genior 1 Kanal, BAV aus:	111 – 121 ms
Genior 1 Kanal, BAV ein:	126 – 136 ms
Genior 8 Kanäle, BAV aus:	180 – 213 ms für Kanal 1 212 – 271 ms für Kanal 8
Genior 8 Kanäle, BAV ein:	196 – 216 ms für Kanal 1 220 – 236 ms für Kanal 8

Festzustellen ist, dass die Reaktionszeiten beider Karten sich bei diesem Test dramatisch verschlechtert haben. Der Grund dafür ist die Programmroutine auf den Karten, die zunächst neue Vorschubssignale verarbeitet bevor Alarme gesetzt werden.

Im Vergleich beider Kartentypen ist auch bei diesem Test die CTM Karte wesentlich stabiler. Ihre Reaktionszeit liegt immer im Tolleranzbereich der Abtastrate von 10 ms und darüber hinaus werden alle Alarme synchron gesendet.

Die Genior Karte hingegen ist nur beim Betrieb eines Kanals im Tolleranzbereich. Beim Betrieb aller acht Kanäle kommt es zu unterschiedlichen Reaktionszeiten die bis zu 60 ms von einander abweichen. Woran diese Schwankungen von Messung zu Messung liegen ist leider nicht zu erklären. Diese ist durch die Entwickler der Genior Software ausgiebig zu erforschen.

Merkwürdig ist auch die Tatsache, das bei dieser Messreihe die Reihenfolge der versendeten Alarmsignale entgegengesetzt zur vorherigen Messung liegen. Dieses Mal wird zunächst der Alarm von Kanal 1 gesetzt und erst zuletzt der Alarm von Kanal 8.

Leider ist auch dieses Messergebnis keine reproduzierbare Messauswertung. Bei mehrfacher Durchführung selbiger Messung änderte sich die Reihenfolge der Alarmsignale. Dabei kam es mehrfach vor, dass die Alarme in den Kanalreihenfolgen (4-3-2-1-5-6-7-8), (3-2-1-4-5-6-7-8) oder (4-3-2-5-1-6-7-8) versendet wurden. Bei genauerer Betrachtung dieser Reihenfolgen ist ein deutliches Schema zu erkennen wie vermutlich die Routine der Software der Karte arbeitet. Erläuterungen dazu sind im Kapitel 8.2 zu finden.

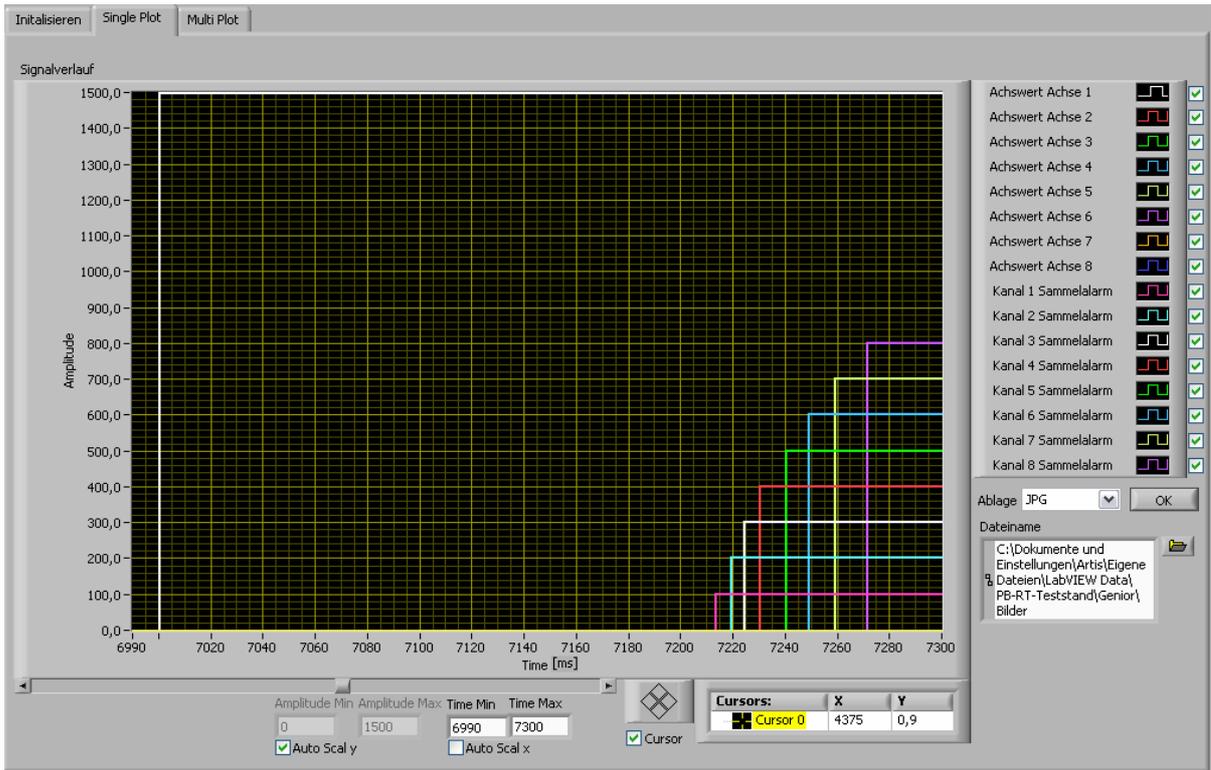


Abb. 54 Performance Genior 8 Kanäle, BAV aus, synchron zum Vorschub

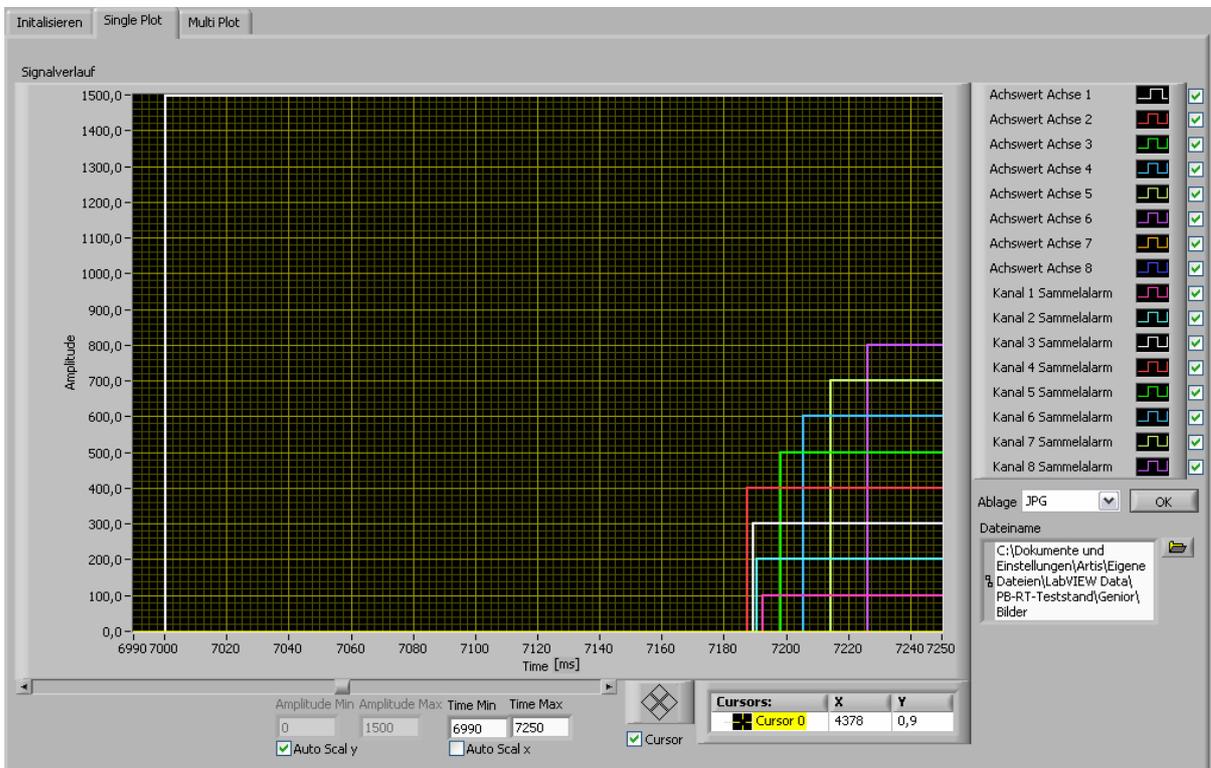


Abb. 55 Performance Genior 8 Kanäle, BAV aus, synchron zum Vorschub 2

 Hochschule für Angewandte Wissenschaften Hamburg Hamburg University of Applied Sciences	Diplomarbeit Fachbereich Elektrotechnik und Informatik	
Wintersemester 07/08 Abgabetermin: 18.12.07	Givian Müller Matr.Nr.: 1749822	Betreuer: Professor Dr. Ing. Müller

8 Testauswertung

8.1 Unterschiedliche Antwortzeiten der Karten

Die gemessenen Unterschiede in den Antwortzeiten, bei gleicher Messung, liegen im Optimalfall bei beiden Karten immer im Bereich von 0 bis 10 ms. Dieser Unterschied kommt zustande, weil der Takt in der die Comsoft Karte und die CTM oder Genior Karte kommunizieren nicht synchron läuft. Dadurch, dass beide Karten bei der Kommunikation im 10 ms Takt senden, kann es entweder sein, das die CTM oder Genior Karte eine Änderung sofort erhält oder gerade verpasst hat und erst im nächsten Takt, also etwa 10 ms verspätet, erhält. Um ein möglichst genaues Messergebnis zu erhalten, müssen daher mehrere Messungen durchgeführt werden. Das genaue Messergebnis ist ermittelt, wenn eine Zeitspanne von 10 ms zwischen dem minimal und maximal gemessenen Wert aufgetreten ist. Der Minimalwert bestimmt dann die Antwortzeit der Karte. Denkbar wäre es diesen Messvorgang zu automatisieren. Mehr dazu ist im Kapitel 10.2.5 zu finden.

8.2 Unterschiedliche Kanalreihenfolge der Alarmsignale

Die Unterschiedliche Kanalreihenfolge die im Kapitel 7.3.4.1 festgestellt wurde, könnte folgendermaßen begründet werden. Die Software der Genior Karte kommt vermutlich innerhalb ihrer Berechnungsroutine nicht mit dem Versenden der Alarme nach und schreibt die Alarme zunächst in einen Speicher. Unter Annahme des LIFO⁵⁴ Prinzips ergibt sich das Verhalten, wodurch immer, sobald Rechenzeit zur Verfügung steht, das zuletzt in den Speicher geschriebene Alarmsignal zuerst bearbeitet wird. Dieses Prinzip würde alle empfangenen Reihenfolgen der Alarme erklären. Sicherlich ist dieses nicht die beste Art, um Alarmsignale zu versenden. Das Ändern der Software zu einem FIFO Prinzip wäre durchaus ratsam.

Anhand dieser Messergebnisse ist es nun möglich, gezielt die Programmroutine der Karten auf derartige Programmierung zu untersuchen und eventuelle Schwächen des Systems zu beheben. Denkbar wären weitere zahlreiche anderweitige Untersuchungen der Performance der Karten die in Zukunft von den Testern durchgeführt werden können.

Angestrebtes Ziel der Entwicklung ist eine kontinuierliche Reaktionszeit aller möglichen Signalantworten der Karten im Bereich von 10 ms. Um dieses zu erreichen, bildet das System eine ideale echtzeitfähige Testumgebung.

⁵⁴ Last In First Out

9 Abschließender Projektbericht

9.1 Einhaltung des Zeitplanes

Der im Kapitel 2.2.2 geplante Zeitaufwand diente als Grundlage der Zeiteinteilung für das gesamte Projekt. Festzustellen ist, dass der reale Zeitaufwand annähernd dem geplanten Zeitaufwand entspricht. Während der Entwicklung des Projektes sind drei ausschlaggebende Zwischenfälle aufgetreten, die den optimalen Verlauf beeinflusst haben.

Realer Zeitaufwand

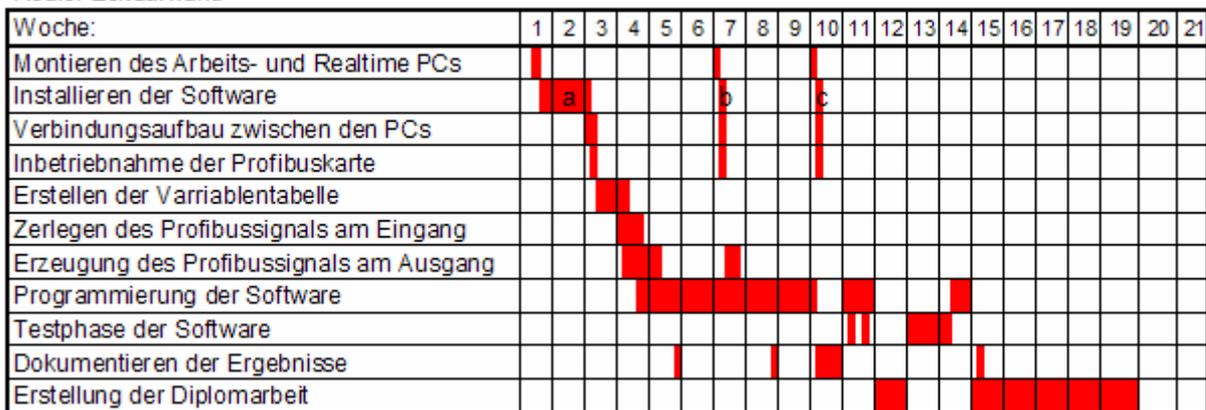


Abb. 56 Realer Zeitaufwand des Projektes

- Die Installation der Software LabVIEW RT verzögerte sich um einige Tage, da vom Hersteller eine falsche Version geliefert wurde. Die zunächst erhaltende Version war zum Betreiben einer annähernd echtzeitfähigen Applikation auf demselben Computer auf dem auch programmiert wird geeignet. Die Verzögerung lag daran, dass, aufgrund schlechter Dokumentation der Software, nicht festzustellen war, dass es sich um die falsche Version handelte.
- Die Echtzeitumgebung musste durch ein leistungsstärkeres System ersetzt werden, da die Auslastung des Systems während der Messung an ihre Grenzen gestoßen ist. Auf dem ersten System waren lediglich Laufzeiten im Bereich von 14 ms realisierbar. Im Nachhinein stellte sich raus, dass sicherlich auch das erste System ausgereicht hätte, da durch die Initialisierung vor der Messung die Leistungsfähigkeit des Target Moduls wesentlich verbessert wurde. Allerdings würden keine Laufzeiten im Bereich von 700 us erreicht werden.
- Die Entwicklungsumgebung musste durch ein leistungsstärkeres System ausgetauscht werden, da die Auslastung des Systems an ihre Grenzen gestoßen ist. Festzustellen war dieses beim Betreiben mehrerer Module zur gleichen Zeit. Auf dem ersten System konnten lediglich drei Module gleichzeitig stabil aktiv sein, während auf dem zweiten System sämtliche Module zur selben Zeit aktiv sein können. Ebenso war festzustellen, dass die Onlinebetrachtung während der Messung nur unregelmäßig Daten empfangen hat.

 Hochschule für Angewandte Wissenschaften Hamburg Hamburg University of Applied Sciences	Diplomarbeit Fachbereich Elektrotechnik und Informatik	
Wintersemester 07/08 Abgabetermin: 18.12.07	Givian Müller Matr.Nr.: 1749822	Betreuer: Professor Dr. Ing. Müller

9.2 Beurteilung des entwickelten Teststandes

Für die abschließende Beurteilung der Software unterscheidet man zum einen die Art der Bedienung und zum anderen die Einsatzvielfalt des Teststandes.

a. Bedienung der Software

Während der Programmierung der Software war festzustellen, dass sich die Bedienung mehrerer Module sehr umständlich gestaltete. Das machte sich zum einen durch viel zu viele auszuführende Schritte bemerkbar, und zum anderen durch denkbar überflüssige Aufforderungen.

Daraufhin wurde die Eingabe bis auf das Nötigste vereinfacht, was einen erheblichen Mehraufwand der Programmierung verursachte. Der Funktionsumfang wurde dadurch nicht beeinflusst, aber die Bedienung wurde derart vereinfacht, dass keine unnötigen Aufforderungen oder Eingaben mehr erforderlich sind.

Die letztlich entstandene Version überzeugt durch Übersichtlichkeit und Einfachheit der Bedienung und steht in der Bedienung der Profibus Master Teststand Software in nichts nach. Die Geschwindigkeit und der Funktionsumfang sind sogar erheblich verbessert worden.

b. Einsatzvielfalt der Software

Bereits bei den ersten Testversuchen der Software stellte sich heraus, dass diese ideal zum durchführen von individuellen Tests, sowie reproduzierbaren Tests geeignet ist.

Die Einfachheit der Handhabung und die Schnelligkeit neue Messwerte zu projektieren machen den Teststand ebenso attraktiv wie effizient.

Die Messergebnisse aus dem Kapitel 7.3 veranlassten bereits die Softwareentwickler die Genior Software auf eventuelle Schwächen zu untersuchen.

Da jeder der Mitarbeiter der Softwareabteilung weitere Tests mit diesem Teststand plant, ist es nötig weitere Systeme dieser Art anzuschaffen.

 Hochschule für Angewandte Wissenschaften Hamburg Hamburg University of Applied Sciences	Diplomarbeit Fachbereich Elektrotechnik und Informatik	
Wintersemester 07/08 Abgabetermin: 18.12.07	Givian Müller Matr.Nr.: 1749822	Betreuer: Professor Dr. Ing. Müller

10 Erweiterungsmöglichkeiten des Teststandes

Durch den modularen Aufbau sind als Erweiterungen für den Teststand viele Möglichkeiten denkbar. Im Folgenden werden Änderungen sowohl von der Hardware als auch von der Software beschrieben.

10.1 Erweiterung der Hardwareumgebung

Grundsätzlich ist eine Erweiterung der Hardwareumgebung immer sinnvoll. Durch einen neuen Rechner, mit schnelleren leistungsfähigeren Prozessoren und größerem Arbeitsspeicher, wird die Auslastung des Systems sowohl von der Entwicklungsumgebung als auch auf dem Echtzeitsystem verringert.

Dadurch resultierend wäre eine geringere Laufzeit auf dem Echtzeitsystems im Bereich von 10 us denkbar.

Die Programmierung unter LabVIEW ermöglicht es zusätzlich neue Rechnersysteme mit Mehrprozessorkernen zu verwenden. Dafür müsste in der Software jeder Berechnungsroutine der jeweilige Prozessor zugewiesen werden.

10.2 Erweiterung der Software

Änderungen der Software umfassen zum einen eine Erweiterung dieser Einzelplatzlösung, sowie eine mögliche Erweiterung auf mehrere Karten oder Rechnersysteme. Ebenso gibt es Erweiterungen die durch neue Signale im Profibusprotokoll resultieren und zum anderen Erweiterungen der Funktionalität der Software.

Eine Beschreibung der Änderung der Programmierung für mögliche Erweiterungen kann hier nicht erfolgen, da es sich bei diesen Anforderungen meist um individuelle Anpassungen handelt. Derartige mögliche Änderungen werden daher lediglich aufgeführt.

Hingegen kann eine Beschreibung der Änderung der Programmierung bei einer Erweiterung des Profibusprotokolls detailliert aufgeführt werden.

10.2.1 Einfügen neuer Profibussignale

Eine Erweiterung des Profibussignals kann sowohl für Signale (Maschine zur Karte) als auch für Signale (Karte zur Maschine) erfolgen. In beiden Fällen ist sowohl eine Erweiterung der Variablenliste im Projektexplorer wie im Kapitel 5.3 beschrieben als auch eine Anpassung mehrerer Module erforderlich. Im Folgenden wird eine Beschreibung der Anpassung einer Änderung beider Signalarten beschrieben:

 Hochschule für Angewandte Wissenschaften Hamburg Hamburg University of Applied Sciences	Diplomarbeit Fachbereich Elektrotechnik und Informatik	
Wintersemester 07/08 Abgabetermin: 18.12.07	Givian Müller Matr.Nr.: 1749822	Betreuer: Professor Dr. Ing. Müller

- a. „Input Kopfzeile (SubVI)“
In diesem Modul muss die Liste der Variablen im Format String ergänzt werden. Dafür wird das zu bildende Array um die zu ergänzenden Variablen nach unten erweitert unabhängig davon welchem Kanal die Variablen angehören. Möglich wäre auch ein Einfügen in die Liste der Variablen, dadurch würden sich allerdings die Änderungen in den Modulen „Projektierung“ und „Auswertung“ erheblich komplizieren.
- b. „Output Kopfzeile (SubVI)“
Dieses Modul ist entsprechend der Beschreibung bei „Input Kopfzeile (SubVI)“ anzupassen.
- c. „Datei Inputzeile auslesen (SubVI)“
Hier ist sowohl eine Anpassung der Oberfläche als auch der Programmierung erforderlich. In der Oberfläche des Moduls sind die Auswahlfelder „Globale Variable“ und „Variable“ mit den entsprechenden Signalen zu ergänzen. Da im Programmcode diese Auswahlfelder Grundlage für die Case-Schleifen sind, müssen auch die Schleifen um die entsprechenden Variablen ergänzt werden. Anschließend muss im neuen Case die Leitung des entsprechenden Typs der Variablen verdrahtet werden. Ebenso werden der Spaltenindex und der Name ausgegeben.
Hier wird erneut sichtbar, dass ein Einfügen der neuen Variablen in die Liste zu einem Verschieben sämtlicher Zeilenindexe führen würde.
- d. „Datei Outputzeile auslesen (SubVI)“
Bei dem Modul ist die Programmierung ebenso wie bei „Inputzeile auslesen (SubVI)“ anzupassen. Lediglich der Typ der Variablen muss nicht angegeben werden, da dieser nur für Variablen (Maschine zur Karte) relevant ist.
- e. „Output Variablen Init. (SubVI)“
Die Liste der Variablenarrays durch die neuen Variablen (Karte zur Maschine) ist zu ergänzen. Da alle Variablen bei der Initialisierung durch dieses Modul denselben Wert erhalten ist es egal an welcher Stelle die neuen Variablen eingefügt werden.
- f. „Input to Variablen (SubVI)“
In diesem Modul ist der Baustein zum Zerlegen des eingelesenen Arrays zu erweitern. Die Position spielt dabei keine Rolle, da die Spalte der Variablen über den bereits unter Punkt c. beschriebenen Spaltenindex angesprochen wird. Anschließend wird der neue Ausgang der neuen Variablen zugewiesen.
- g. „Variablen to Output (SubVI)“
Der Baustein zum Bilden des Arrays in diesem Modul ist um die entsprechenden Variablen nach unten zu erweitern, so dass diese ans Ende des Arrays geschrieben werden. Die Position entspricht dabei wieder dem neuen Zeilenindex.
- h. „Projektierung“
Die Auswahlfelder aller Kanäle sind identisch mit den Änderungen aus Punkt c. für beide Kartentypen zu erweitern. Darüber hinaus sind die Änderung der Module „Datei Inputzeile auslesen (SubVI)“ für beide Kartentypen zu übernehmen. Dieses geschieht über rechts klick auf das Modul und „Relink To SubVI“.

 Hochschule für Angewandte Wissenschaften Hamburg Hamburg University of Applied Sciences	Diplomarbeit Fachbereich Elektrotechnik und Informatik	
Wintersemester 07/08 Abgabetermin: 18.12.07	Givian Müller Matr.Nr.: 1749822	Betreuer: Professor Dr. Ing. Müller

- i. "Simulation Global M-K"
Die Bedienoberfläche des Moduls ist durch entsprechende neue Anzeigefelder zu ergänzen. Diese erhalten nach gleichem bereits in dem Modul verwendeten Schema über den aktuellen Index Werte von der Variablen.
- j. "Simulation Kanäle M-K"
Die Änderung ist ebenso wie bei "Simulation Global M-K" durchzuführen.
- k. "Simulation Global K-M"
Genau wie beim Modul "Simulation Global M-K" wird die Bedienoberfläche durch entsprechende Anzeigeelemente ergänzt. Diesen neuen Elementen wird entsprechend des Zeilenindexes der Wert des Datenbusses zugewiesen. Falls erforderlich werden die Werte über eines der Formatmodule in einen Wert umgewandelt.
- l. "Simulation Kanäle K-M"
Dieses Modul ist ebenso wie "Simulation Global K-M" zu ergänzen.
- m. "Auswertung"
Wie bereits unter Punkt h. beschrieben müssen die Auswahlfelder aller Kanäle beider Kartentypen identisch zu den Änderungen aus Punkt c. erweitert werden. Darüber hinaus sind die Änderungen der Module „Datei Inputzeile auslesen (SubVI)“ und „Datei Outputzeile auslesen (SubVI)“ zu übernehmen. Dieses geschieht über rechts klick auf das jeweilige Modul und „Relink To SubVI“.
- n. "Variable to Bus (SubVI)"
Hier muss die wesentlichste Änderung der Software durchgeführt werden. Die neue Variable muss auf die entsprechende Adresse des Profibusses geschrieben werden. Die Reihenfolge der Arraysignale entspricht der Reihenfolge der Bytes auf dem Profibus. Um eine neue Variable einzufügen wird das entsprechende Byte vom derzeit angeschlossenen Füllbyte getrennt und über den Index mit der neuen Variablen verbunden. Je nach Signaltyp der Variablen kann eine Umwandlung der Variablen in mehrere Bytes erforderlich sein.
- o. "TargetVI"
Innerhalb dieses Moduls findet die Zerlegung des empfangenen Profibussignals statt. Neue Variablen werden daher nach vorhandenem Schema vor der zeitgesteuerten Schleife eingelesen und nach der Schleife wieder zugewiesen. Innerhalb der Schleife wird anhand der neuen Profibusadresse der entsprechende Wert auf die Feedback Node der Variablen geschrieben. Je nach Signaltyp der Variablen kann eine Umwandlung des Bytes in mehrere Variablen oder ein Zusammenfügen mehrere Bytes in einen Wert erfolgen.

 Hochschule für Angewandte Wissenschaften Hamburg Hamburg University of Applied Sciences	Diplomarbeit Fachbereich Elektrotechnik und Informatik	
Wintersemester 07/08 Abgabetermin: 18.12.07	Givian Müller Matr.Nr.: 1749822	Betreuer: Professor Dr. Ing. Müller

10.2.2 Überprüfung der Profibusverbindung

Um während der Messung, Fehler durch Unterbrechungen in der Kommunikation zu vermeiden, wäre es denkbar, diese kontinuierlich zu überprüfen und zu protokollieren. Dafür müsste der Verbindungsstatus des eingebundenen Bausteins der Firma Comsoft abgerufen und kontinuierlich überprüft werden. In der derzeitigen Software wurde auf dieses Feature aufgrund der nicht bekannten, eventuell auftretenden Laufzeitverzögerung verzichtet. Fraglich ist auch, wie eine derartige Unterbrechung der Kommunikation dokumentiert werden kann und soll.

10.2.3 Kartenunabhängige Softwarelösung

Eine der grundlegendsten Anpassungen der Software ist die Verbindung aller kartenabhängigen Module zu allgemeingültigen Modulen. Bereits zu Beginn des Projektes war eine derartige Lösung im Gespräch, allerdings war zu diesem Zeitpunkt die Art der Umsetzung der gesamten Software noch nicht abzuschätzen, so dass auf eine kartenabhängige Variante der Programmierung zurückgegriffen wurde. Nachdem die Programmierung der Software weitestgehend abgeschlossen wurde, ist eine mögliche Zusammenführung beider Kartenmodule in zwei Varianten denkbar. Zum einen eine geschälerte Version in der der Kartentyp zwar gewählt werden muss, aber während der Laufzeit geändert werden kann, und zum anderen eine umfangreich globale Lösung bei der der Kartentyp keine Rolle spielt.

10.2.3.1 Kartentyp während der Projektierung ändern

Bei dieser Variante der Zusammenführung würde die Programmierung der Module erhalten bleiben und lediglich durch Case-Abfragen, wie sie bereits beim Modul „Projektierung“ verwendet werden, gegliedert werden. Der zuletzt verwendete Kartentyp würde dabei als Variable gespeichert werden, um so für alle Module zur Verfügung zu stehen.

Das Problem bei dieser Art der Zusammenführung ist, dass bei einer Auswertung älterer Messungen der richtige Kartentyp ermittelt werden müsste. Dieser würde dann beispielsweise in jedem Header gespeichert werden.

Viel Ausschlaggebender ist aber die Tatsache, dass dadurch der Programmcodeumfang eines jeden Moduls auf die doppelte Datenmenge anwachsen würde, und dadurch während der Laufzeit ein wesentlich größerer Arbeitsspeicherbereich belegt werden würde.

 Hochschule für Angewandte Wissenschaften Hamburg Hamburg University of Applied Sciences	Diplomarbeit Fachbereich Elektrotechnik und Informatik	
Wintersemester 07/08 Abgabetermin: 18.12.07	Givian Müller Matr.Nr.: 1749822	Betreuer: Professor Dr. Ing. Müller

10.2.3.2 Globales kartenunabhängiges Gesamtkonzept

Bei dieser Lösung der Zusammenführung beider Kartensysteme müssten einige Module ebenso wie in Kapitel 10.2.3.1 beschrieben durch Case-Abfragen unterteilt werden. Allerdings würde es sich hierbei nur um SubVIs handeln, die nicht direkt vom Nutzer der Software bedient werden würden. Die Hauptmodule Projektierung und Auswertung müssten derart umprogrammiert werden, dass alle Elemente für beide Kartentypen gleich funktionieren.

Um die unterschiedlichen Bussignale mit den identischen Bussignalen zu kombinieren müsste eine gemeinsame Quelldatei angelegt werden, in der jeder Kartentyp die entsprechende Zuweisung auf seine Elemente hat. Dadurch würde sich für eine gemeinsame Nutzung der Umfang der Daten etwa um den Faktor 1,7 erhöhen. Das bedeutet sowohl die Erstellung als auch die Projektierung von Signalwerten würde um diesen Faktor länger dauern.

Auf dem Echtzeitsystem würden dadurch beide Systeme kontinuierlich aktiv sein und auf das jeweilige Startsignal des Kartentyps warten. Dadurch würde es auf dem Echtzeitsystem ebenso zu einem größeren Speicherbedarf kommen.

10.2.4 Speichern der Daten im Dauerbetrieb

Durch das Eingeben von 0 Messungen im Modul „Projektierung“ ist es möglich unendlich viele Messungen nacheinander zu durchlaufen. Das Problem dabei ist, dass lediglich die Messergebnisse der letzten Messung protokolliert werden. Eine denkbare Erweiterung zur Überprüfung der Signale bei der Durchführung von Dauertests wäre die Protokollierung aller Änderungen der empfangenen Signale. Denn nur die Speicherung der Flankenwechsel würde ein Überlauf der Wertemengen verhindern.

Welche Signale dabei überwacht werden sollten ist zu diesem Zeitpunkt jedoch noch nicht klar. Denkbar wären auf jeden Fall alle Alarmer sowie die globalen Bussignale abgesehen von den Achswerten.

10.2.5 Automatische Protokollierung der Ergebnisse

Als denkbare Erweiterung und wesentliche Vereinfachung für den Benutzer der Software wäre eine Automatisierung des Auswerteprozesses möglich. Beispielsweise könnte mit einer Funktion die Zeitverzögerung zwischen Achswertänderung und Alarmausgangsänderung ermittelt und automatisch gespeichert werden. Dadurch könnten auch die im Kapitel 8.1 auftretenden Verzögerungen durch mehrfache Messung ermittelt werden.

Sobald beispielsweise eine Differenz der Alarmsignale einer Messung von 10 ms auftritt, könnte die Messreihe gestoppt werden und der ermittelte Verzögerungswert gespeichert werden.

 Hochschule für Angewandte Wissenschaften Hamburg Hamburg University of Applied Sciences	Diplomarbeit Fachbereich Elektrotechnik und Informatik	
Wintersemester 07/08 Abgabetermin: 18.12.07	Givian Müller Matr.Nr.: 1749822	Betreuer: Professor Dr. Ing. Müller

Derartige Ergebnisse könnten dann beispielsweise als Ergebnisprotokoll gedruckt oder gespeichert werden.

Für welche Art von Signalreaktionen diese sinnvoll wäre, ist zu diesem Zeitpunkt noch nicht geklärt. Aufgrund der Unterschiede der möglichen Testmöglichkeiten wäre eine allgemeine Auswertung dieser Art nicht möglich. Denkbar wäre eher eine individuelle Lösung für häufig wiederkehrende Messungen.

10.2.6 Teststand im Multiuserbetrieb

Derzeit handelt es sich bei der Entwicklung der Software um eine generelle Einplatzlösung. Das bedeutet, dass an einem System immer nur ein Mitarbeiter arbeiten kann. Es besteht zwar die Möglichkeit eine CTM und eine Genior Karte gleichzeitig zu betreiben, jedoch ist dafür die Verwendung von zwei Projektierungsmodulen nötig.

Eine denkbare Erweiterung des Systems wäre eine Vervielfältigung der Module „Target“ von CTM und Genior, wobei jedes Modul eine andere Kartenadresse zugeteilt bekäme. Darüber hinaus wäre für jede Karte ein weiterer Variablensatz nötig.

Dadurch wäre es möglich, bei der derzeitigen Leistungsfähigkeit des Echtzeitsystems, bis zu zehn Karten parallel zu betreiben.

10.2.6.1 Problematik im Multiuserbetrieb

Bei der Erweiterung des Systems zu einer Multiusernutzung kommt es zu mehreren Problemen. Zum einen kann über Ethernet immer nur ein Entwicklungsrechner auf das Echtzeitsystem zugreifen, dadurch müsste die Kommunikation nacheinander ablaufen, so dass es zu Laufzeitverzögerung in der Onlinebetrachtung als auch in der allgemeinen Kommunikation kommen würde.

Der wesentliche Aspekt jedoch ist die Tatsache, dass auf dem Echtzeitsystem ein kontrolliertes Scheduling stattfinden müsste. Die Art des Scheduling auf dem System ist durch Prioritäten geregelt. Das bedeutet jede Karte würde eine eigenständige Priorität erhalten und ebenso würde die Projektierung einer Messung jeder Karte eine niedrigere Priorität als die eigentlichen Messungen der Karten erhalten.

Sollte ein derartiges System eingerichtet werden, ist es für jeden einzelnen Benutzer der Software nicht zu erkennen, was gerade auf dem Echtzeitsystem abläuft und ob nicht vielleicht doch durch die Mehrfachnutzung die Auslastungsgrenze erreicht ist. Um derartige eventuelle Fehler zu vermeiden ist es ratsam mehrere Einzelplatzlösungen zu schaffen.

 Hochschule für Angewandte Wissenschaften Hamburg Hamburg University of Applied Sciences	Diplomarbeit Fachbereich Elektrotechnik und Informatik	
Wintersemester 07/08 Abgabetermin: 18.12.07	Givian Müller Matr.Nr.: 1749822	Betreuer: Professor Dr. Ing. Müller

10.2.6.1.1 Scheduling Verfahren mit festen Prioritäten

In der folgenden Übersicht ist die zeitliche Startabfolge von 4 Karten (CTM 1, CTM 2, Genior 1 und Genior 2) beschrieben. Darüber hinaus unterscheiden sich alle Karten in der Zeit der Bearbeitung und in der Priorität sowohl für die Initialisierung als auch für die eigentliche Messung.

Die Zeit wird ohne Bezug auf die Größe (Sekunden, Minuten oder Stunden) angegeben, da das Schedulingverfahren unabhängig von der Länge der Zeit arbeitet. Bei der Priorität bedeutet eine hohe Zahl eine hohe Priorität.

Kartenname (Task)	Ankunftszeit	Bearbeitungszeit		Priorität	
		Initialisierung	Messung	Initialisierung	Messung
CTM 1	0	10	10	30	80
CTM 2	3	15	20	29	79
Genior 1	20	15	25	50	100
Genior 2	5	10	10	49	99

Abb. 57 Scheduling Prioritäten im Multiuserbetrieb

Nach dem Starten der Karte CTM 1 wird die Initialisierung begonnen. Wenn nach drei Zeiteinheiten der Startbefehl von der Karte CTM 2 erfolgt, wird dieser zunächst zurückgestellt. Erst der Startbefehl von Karte Genior 1 unterbricht den Task von CTM 1. Nachdem die Initialisierung von Genior 1 abgeschlossen ist, beginnt direkt die Messung dieser Karte. Auch der Startbefehl der Karte Genior 2 wird dabei erst einmal zurückgestellt. Nach Beenden der Messung von Genior 1 durchläuft Genior 2 den kompletten Prozess. Anschließend kann CTM 1 die Initialisierung und anschließende Messung fortführen. Erst zum Schluss beginnt CTM 2 mit der Initialisierung und Messung.

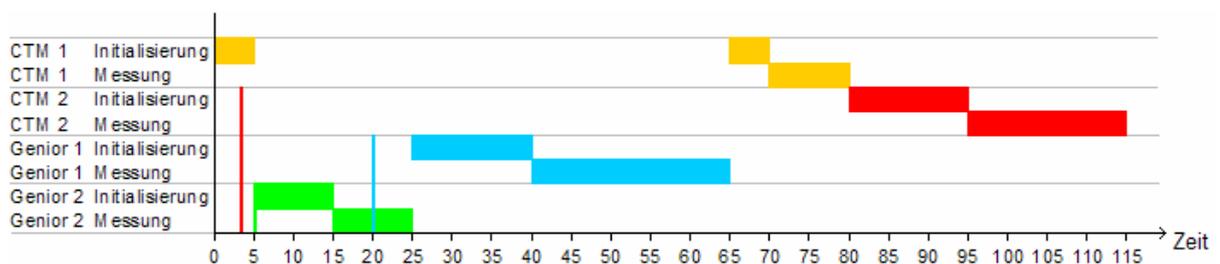


Abb. 58 Scheduling Ablaufdiagramm im Multiuserbetrieb

Dadurch, dass sämtliche Messungen eine höhere Priorität als die Projektierungen besitzen, wird sichergestellt, dass immer alle Messungen ohne Unterbrechungen durchlaufen werden. Unterbrechungen während der Initialisierung wirken sich nicht auf die Ausführungen der Messungen aus.

 Hochschule für Angewandte Wissenschaften Hamburg Hamburg University of Applied Sciences	Diplomarbeit Fachbereich Elektrotechnik und Informatik	
Wintersemester 07/08 Abgabetermin: 18.12.07	Givian Müller Matr.Nr.: 1749822	Betreuer: Professor Dr. Ing. Müller

11 Fazit

Abschließend kann ich behaupten, dass das Projekt absolut gelungen ist. Durch den erstellten Zeitplan konnte ich ständig den zeitlichen Verlauf im Auge behalten, so dass ich zu keinem Zeitpunkt in Termenschwierigkeiten geraten bin. Weiterhin habe ich durch die Vielseitigkeit der Funktionen der entwickelten Software sehr viel über die Programmierung unter LabVIEW gelernt. Es gibt nur wenige Bausteine innerhalb der LabVIEW Bibliotheken, deren Funktion während der Programmierung nicht mindestens einmal getestet wurden. In diesem Zusammenhang war die Thematik zur Entwicklung eines echtzeitfähigen Systems sehr interessant, zumal derartige Systeme in Zukunft in vielen Bereichen eingesetzt werden. Ebenso wurde mir ein tief greifender Einblick in die Systeme der CTM und Genior Karten ermöglicht, was mir die Thematik der Werkzeugüberwachung sehr nahe gebracht hat. Das Wichtigste für mich persönlich ist jedoch die Tatsache, dass ein Teststand entwickelt wurde, der von den Mitarbeitern tatsächlich verwendet wird, weil das System eine schon länger geforderte Erweiterung der Testmöglichkeiten darstellt. Aus diesem Grund beinhaltet die Diplomarbeit auch eine detaillierte Beschreibung der eigentlichen Programmierung, da sie ebenso als Handbuch für die weitere Benutzung dient.

 Hochschule für Angewandte Wissenschaften Hamburg Hamburg University of Applied Sciences	Diplomarbeit Fachbereich Elektrotechnik und Informatik	
Wintersemester 07/08 Abgabetermin: 18.12.07	Givian Müller Matr.Nr.: 1749822	Betreuer: Professor Dr. Ing. Müller

Abkürzungsverzeichnis

Abb.	Abbildung
API	Adaptive Planungsintelligenzen (mathematische Methoden für industrielle Planungsaufgaben)
ATA	AT- Attachment / Name der IDE-Schnittstelle
BAV	Bruch Alarm Verzögerung
bzw.	Beziehungsweise
CFM-4	Kollisionskraft Messumformer Serie 4
COCOMO	COConstructive COst MOdel
CPU	Central Processing Unit / Hauptprozessor
CTM	Computer Integrated Toll and Machine Monitoring
DDU4	Double Drehmoment Unit / Vorschub und Achsialkraft Umformer Serie 4
DO	Digital Output
DTA	Daten Träger Austauschverfahren / Digital Performer
EDV	Elektronische Daten Verarbeitung
FAT	File Allocation Table
FIFO	First In First Out
GB	Giga Byte
Genior	Synonym / Produktname
GHz	Gigaherz
GSD	General Station Description / Gerätestammdaten
GUI	Grafische Benutzeroberflächen
HB	High Byte
HHB	High High Byte
HLB	High Low Byte
HTML	Hypertext Markup Language / Hypertext-Auszeichnungssprache
ISA	Industry Standard Architecture
I / O	Input / Output
JPEG	Bildformatstandard / Joint Photographic Experts Group
KU-4	Körperschallumformer Serie 4
LB	Low Byte
LED	Leuchtdiode / Lumineszenz Diode
LHB	Low High Byte
LIFO	Last In First Out
LLB	Low Low Byte
MAC	Media Access Control
MAX	Measurement & Automation Explorer
MB	Mega Byte
ms	Millisekunden
MU 4	Messumformer Serie 4
NC	Numerical Control
NI	National Instruments

 Hochschule für Angewandte Wissenschaften Hamburg Hamburg University of Applied Sciences	Diplomarbeit Fachbereich Elektrotechnik und Informatik	
Wintersemester 07/08 Abgabetermin: 18.12.07	Givian Müller Matr.Nr.: 1749822	Betreuer: Professor Dr. Ing. Müller

o.V.	ohne Verfasser
PC	Personal Computer
PCI	Peripheral Component Interconnect
RX	Remote Procedure Call
SubVI	Untergeordnetes Virtuelles Instrument / Unterprogramm
TCP / IP	Transmission Control Protokoll / Internet Protokoll
txt	Textdatei
us	Mikro Sekunden
V	Volt
VG-4	Beschleunigungs- und Schwinggeschwindigkeitsumformer Serie 4
VI	Virtuelle Instrumente
VISA	Virtual Installation Solution Architecture
XP	eXPerience
z. B.	zum Beispiel

 Hochschule für Angewandte Wissenschaften Hamburg Hamburg University of Applied Sciences	Diplomarbeit Fachbereich Elektrotechnik und Informatik	ARTIS [®] 
Wintersemester 07/08 Abgabetermin: 18.12.07	Givian Müller Matr.Nr.: 1749822	Betreuer: Professor Dr. Ing. Müller

Literaturverzeichnis

- 1 Dr. Dirk Lange (Artis): Concept Rolls Royce in:
Concept_Rolls_Royce_2007_08_21_DL, 23.08.2007
- 2 o.V.: Comp.realttime: Frequently Asked Questions (FAQs), in:
<http://de.wikipedia.org/wiki/Echtzeit%C3%A4higkeit>, 20.09.2007

 Hochschule für Angewandte Wissenschaften Hamburg Hamburg University of Applied Sciences	Diplomarbeit Fachbereich Elektrotechnik und Informatik	
Wintersemester 07/08 Abgabetermin: 18.12.07	Givian Müller Matr.Nr.: 1749822	Betreuer: Professor Dr. Ing. Müller

Glossar

Applikation	Ein Computerprogramm, das eine für den Anwender nützliche Funktion ausführt (Anwendungsprogramm)
Array	zu Deutsch Feld, ist der Sammelbegriff für eine Anordnung, Aufstellung, Reihe von gleichen Elementen in festgelegter Art und Weise
Cache	ein schneller Pufferspeicher
Deterministisch	eindeutig definierter Folgezustand
Drag and Drop	Methode zur Bedienung graphischer Benutzeroberflächen von Rechnern durch das Bewegen grafischer Elemente mittels eines Zeigegerätes
Header	zu Deutsch Kopf, Einleitung, Vorspann, Metadaten am Anfang einer Datei oder eines Datenblocks
Initialisierung	Ladevorgang, in dem der zur Ausführung benötigte Speicherplatz (z. B. Variablen, Code, Puffer, ...) für das Programm reserviert und mit Startwerten gefüllt wird
Intervall	Zeitabschnitt, Zeitspanne oder Zeitintervall, Teil der Zeit, abhängig vom Wechsel der Ereignisse und Eindrücke oder vom Verlauf der Geschehnisse
Iteration	schrittweise, wiederholter Zugriff auf Datenstrukturen
Master-Slave	Form der Verwaltung des Zugriffs auf eine gemeinsame Ressource
Offset	konstanter additiver oder subtraktiver systematischer Fehler oder Abweichung einer Zustandsgröße oder eines Messwertes
Performance	Zeitverhalten von Programmen (Software) und Geräten (Hardware)
Projektierung	Vorbereitung eines Ereignisses, eines Prozesses oder eines realen Konstruktes
Scheduling	Zeitablaufsteuerung, Erstellung eines Ablaufplanes, der Prozessen zeitlich begrenzt Ressourcen zuweist
Support	Beratung, Problemorientierte Beratungstätigkeit
Webbrowser	Computerprogramme zum Betrachten von Webseiten im World Wide Web
Zyklustakt	periodisch wiederkehrendes Ereignis

 Hochschule für Angewandte Wissenschaften Hamburg Hamburg University of Applied Sciences	Diplomarbeit Fachbereich Elektrotechnik und Informatik	
Wintersemester 07/08 Abgabetermin: 18.12.07	Givian Müller Matr.Nr.: 1749822	Betreuer: Professor Dr. Ing. Müller

Anhang

I. Pflichtenheft

Auf den folgenden neun Seiten befindet sich die freigegebene letzte Version des Pflichtenheftes für diesen Teststand. Erstellt wurde die Ursprungsversion von Herrn Axel Rauther einem der Softwareentwickler der Firma. Diese wurde anschließend gemeinsam in mehreren Schritten ergänzt.

Pflichtenheft

Simulator

Simulation des Verhaltens einer Maschine an der Schnittstelle zum Genior/CTM System, sowie Analyse der Reaktion des Genior/CTM Systems gegenüber einer Maschine

ARTIS GmbH
Sellhorner Weg 28
29646 Behringen

mail: info@artis.de

Aufgestellt:	Axel Rauther
Status:	<Freigabe>
Kategorie:	< >
Datum:	08.08.2007
e-mail:	rauther@artis.de
Telefon:	05194/950-0
Version:	1.05
Ergänzt:	
Letzte Änderung:	08.11.07

1 Inhalt

1	Inhalt	2
2	Änderungsdokumentation	3
3	Referenzpapiere	3
4	Motivation	4
4.1	Ziele	4
5	Beschreibung	4
5.1	Use Case Model	5
5.2	Allgemeine Oberflächenprogrammierung	5
5.3	Projektierung des Schnittstellenverhaltens der Maschine	5
5.3.1	Projektierung Sequenzen globale Ausgänge	6
5.3.1.1	Projektierung Sequenzen DTA-Achswert Profile	6
5.3.2	Projektierung Sequenzen PLC-Kanäle	6
5.4	Durchführung der Simulation	7
5.4.1	Versenden und Empfangen der projektierten Schnittstellenparameter	7
5.4.2	Betrachtung der Signale während der Simulation	7
5.4.3	Datenspeicherung während der Simulation	8
5.5	Auswertung der Simulationsergebnisse	8
5.5.1	Sicherung der Messdaten	8
5.5.2	Darstellung der Messergebnisse	8
5.5.3	Zeitverhalten	8
6	Anforderungen für zukünftige Erweiterungen	9
6.1.1	Schnittstellenparameter während der Simulation verändern	9
6.1.2	Simulatorverhalten auf Reaktion von Genior/CTM	9
6.1.3	Simulation Maschinenbenutzerverhalten	9

2 Änderungsdokumentation

Index	Inhalt der Änderung	Änderungsgrund	Bearbeiter	Datum
1	Ursprungsversion		Axel Rauther	10.08.2007
1.1	Anforderungsspezifikation	Vertiefung der Anforderungen	Givian Müller	14.09.2007
1.2	Anpassung der Bedienbarkeit	Erleichterung der Bedienung	Givian Müller	09.10.2007
1.3	Genauere Ausführung der Anforderung	Vertiefung der Begrifflichkeit	Givian Müller	11.10.2007
1.4	Korrektur der Grafiken		Axel Rauther	22.10.2007
1.5	Pfade der Referenzpapiere erweitert	Fehlende Referenzen	Axel Rauther	08.11.2007

3 Referenzpapiere

Dok.-Nr.	Dokument	Verwendung in / für
1	\\Artis6\alle\Forschung und Entwicklung\Produkte\5041 – Genior\Pflichtenheft\Software\7448_Simulator\Simulator.eap	Enterprise Architekt/UML
2	\\Artis6\alle\Forschung und Entwicklung\Produkte\5041 - Genior\Software\Profibus interface\Interface_Genior.xls	PB Step 2 Genior
3	\\Artis6\alle\Forschung und Entwicklung\Produkte\CTM\6030 - CTM 1 V3\Dokumentation\Schnittstelle.xls	PB Step 2 CTM

4 Motivation

Um das Verhalten unseres Genior bzw. CTM-Systems in einer realistischen Umgebung zu testen, müssten wir das System an eine Maschine anschließen und dort entsprechende Bearbeitungen fahren.

Dies hat jedoch Nachteile, wie z.B.:

1. Eine NC Maschine ist teuer, daher ist es z.B. nicht möglich jedem Entwickler/Tester eine eigene zur Verfügung zu stellen
2. Der Platzbedarf einer Maschine ist erheblich
3. Es gibt nicht „Die“ Maschine, da wir unterschiedlichste Maschinen von unterschiedlichen Herstellern überwachen (z.B. Dreh- / Fräs-Maschinen/ Transferstrassen usw.)
4. Die Maschine liefert keine 100% reproduzierbare Sensordaten (trotz gleichen Werkzeuges und Materials ist jede Bearbeitung leicht bis erheblich unterschiedlich)
5. Die Bedienung einer NC Maschine ist kompliziert und Aufwendig
6. Fehlbedienung kann zu Schäden an der Maschine führen (man muss daher z.B. vorsichtig sein beim Erforschen von Grenzwerten)
7. Es ist nicht möglich eine definierte Störung während einer Bearbeitung zu erzeugen (z.B. Bruchalarm auf Grund eines Sensorwertes von 167% für 100ms der 2,4 s nach Überwachungsstart die Bruchgrenze von 120% überschreitet)
8. Es ist z.B. nicht möglich zu Prüfen nach welcher Zeit das Genior-System auf eine Störung reagiert.
9. Es ist z.B. auch nicht möglich versendetete Profibustelegramme sich anzuschauen

4.2 Ziele

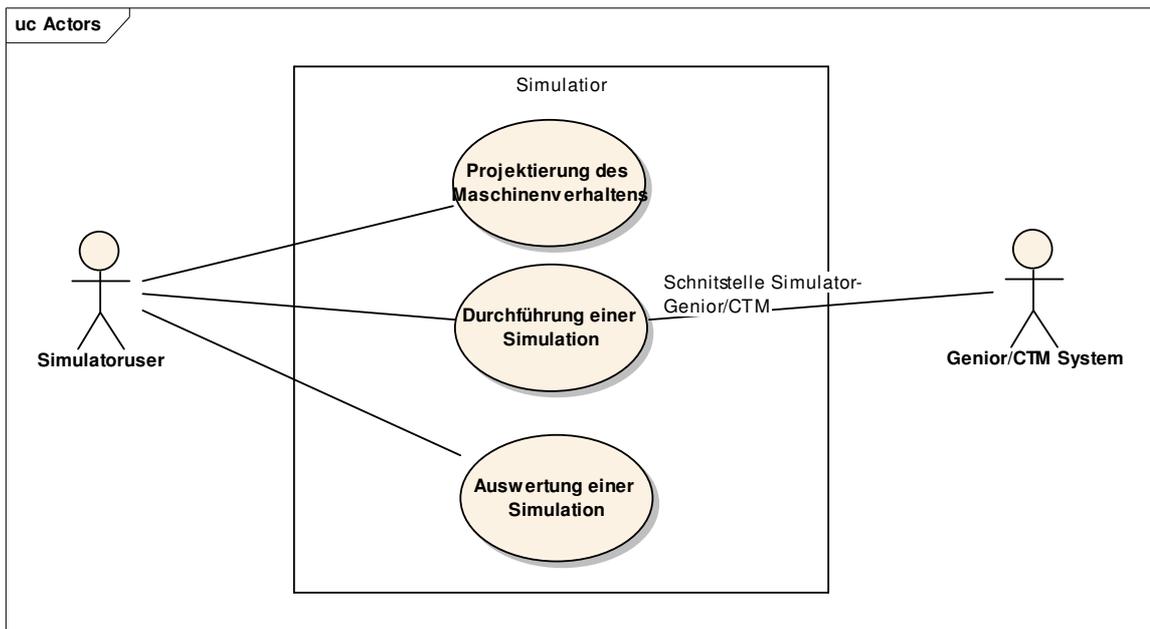
Aus der Motivation ergeben sich zwei wesentliche Ziele für den Simulator

1. Simulation des Verhaltens einer Maschine an der Schnittstelle zum Genior/CTM System
2. Analyse der Reaktion des Genior/CTM Systems gegenüber einer Maschine

5 Beschreibung

1. Oberflächenprogrammierung mit einer Auflösung von 1024x768
2. Jederzeit mögliche Programmunterbrechung
3. Variable Simulationslaufzeit von 10ms bis ca. 60000ms
4. Veränderung des Profibustaktes
5. kontinuierlicher Kommunikationstakt von 10ms

5.2 Use Case Model



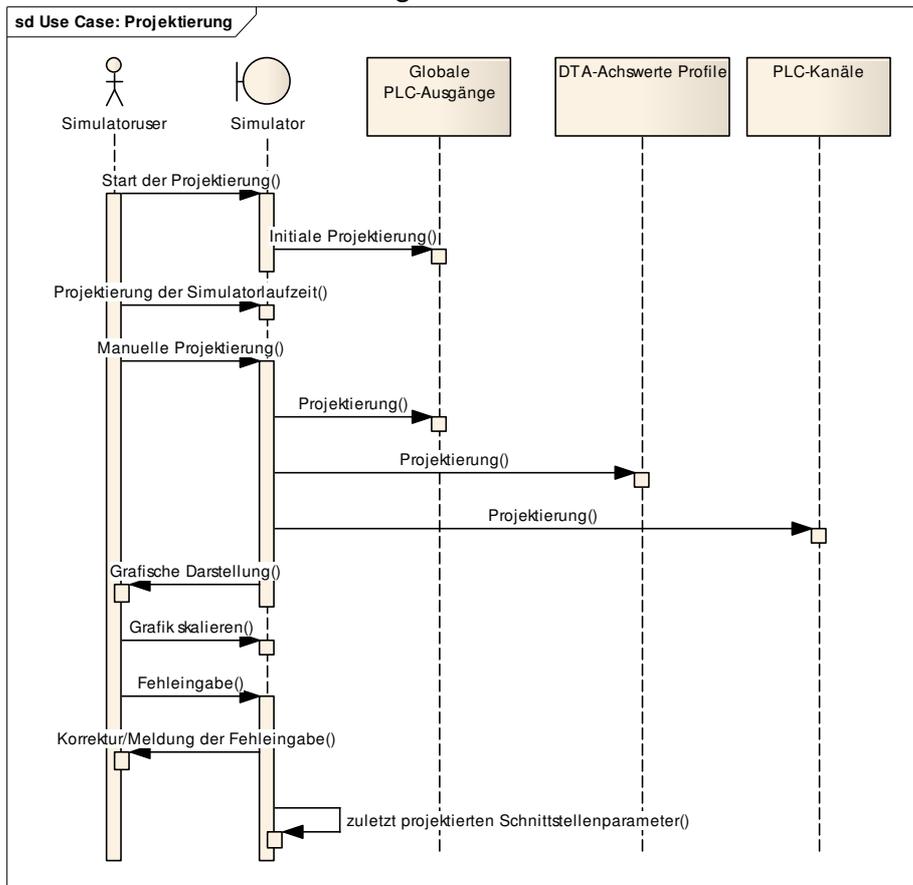
5.3 Allgemeine Oberflächenprogrammierung

1. Schlüssige Modulbenennung um die Übersicht in der Menüleiste zu gewährleisten
2. Aussagekräftige Grafiken in der Menüleiste, insofern möglich

5.4 Projektierung des Schnittstellenverhaltens der Maschine

1. Generieren und Einbinden einer vielseitig nutzbaren Eingabequelle im Textformat
 - a. Speicherpfad frei wählbar und jederzeit konfigurierbar
 - b. Konfigurierbare Header- Daten wie Bezeichnung und Kommentar
 - i. Initiale Bezeichnung generiert sich aus Dateiname
 - c. Sicherung des Zeitstempels der letzten Bearbeitung in der Datei
 - d. Manuelle Änderung der Datei ermöglichen um Beispielsweise vorhandene Maschinendaten zu integrieren
2. Initialisierungswerte aus einer separaten Datei einlesen
 - a. Initialisierungsdatei konfigurierbar
3. Frei definierbare Messdauer in 10ms Schritten
 - a. Laufzeit jederzeit änderbar
4. Projektierung aller Schnittstellenparameter für jeden Takt ermöglichen
5. Grafische Anzeige von zwei projektierten Parameterdaten
 - a. Signalnamen in der Grafik darstellen
 - b. Multidimensional bewegbare Cursor einbinden
6. Skalierbare Grafik
 - a. Initiale Grafikeinstellung an die Schnittstellenparameter anpassen
 - b. Einstellbare Grafikhöhe
 - c. Einstellbare Grafikbreite
 - d. Gesamter Zeitverlauf scrollbar
7. Abfangen aller eventuellen Fehleingaben

8. Die zuletzt projizierten Schnittstellenparameter werden für die Durchführung der Simulation bereitgehalten



5.4.3 Projektierung Sequenzen globale Ausgänge

1. Definierter Zustand der globalen PLC-Ausgänge über die Messdauer.

5.4.3.1 Projektierung Sequenzen DTA-Achswert Profile

Definierter Zustand der Achswerte für jede Bearbeitung jeder Achse über die Messdauer

1. Signalverlauf während Überwachung
2. Signalverlauf während Pause
3. Sowohl positive als auch negative Werte realisierbar
4. Erzeugung von Signalverläufen wie (Rechteck/Rampen/Kurven)
5. Zuschalten von Offset/Rauschen ermöglichen

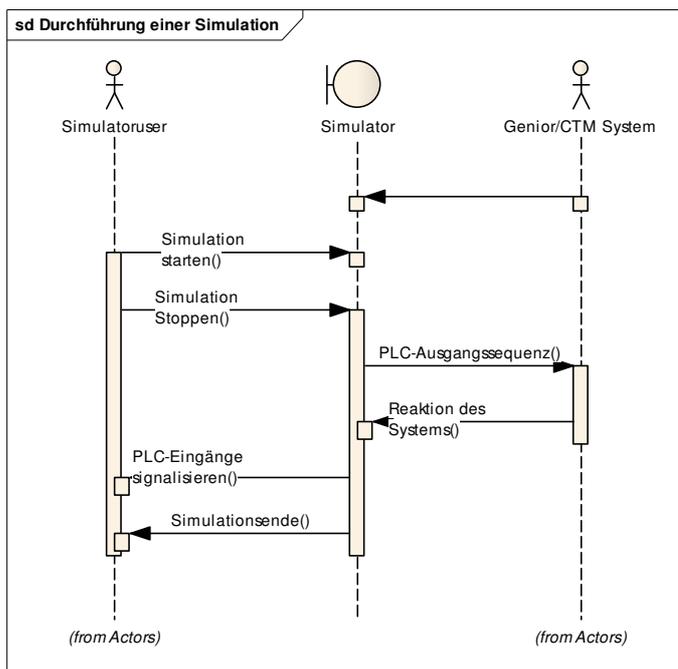
5.4.4 Projektierung Sequenzen PLC-Kanäle

Definierter Zustand der PLC-Kanäle über die Messdauer

z.B.: 5000 ms nach dem Start das Lernenbit für 100 ms setzen.

5.5 Durchführung der Simulation

1. Für die Durchführung der Simulation werden wohl definierte Ausgangssignale zum Genior/CTM System während der gesamten Zeit benötigt.
2. Nach dem Simulationsstart wird keine Änderung der Parameterdaten übernommen. Diese stehen erst für die nächste Messung bereit.
3. Simulation endet durch
 - a. Abbruch des Benutzers wodurch keine Daten die von der Genior/CTM Karte empfangen wurden zur Auswertung verfügbar sind
 - b. Erreichen des Endes der projektierten Laufzeit wodurch empfangene Daten der Genior/CTM Karte zur Auswertung verfügbar sind



5.5.3 Versenden und Empfangen der projektierten Schnittstellenparameter

1. Die Kommunikation zwischen dem Teststand und den Karten (Genior/CTM) erfolgt über Profibus
2. Die Profibusadresse sowie der Profibustakt sind konfigurierbar
3. Schnittstellenparameter werden nach dem PB Step 2 Genior/CTM Protokoll auf dem Profibus versendet

5.5.4 Betrachtung der Signale während der Simulation

1. Einzeln wählbare zeitnahe Betrachtung der aktuellen
 - a. Globalen Schnittstellenparameter von der Maschine zur Karte
 - b. Kanal Schnittstellenparameter von der Maschine zur Karte
 - c. Globale Schnittstellenparameter von der Karte zur Maschine
 - d. Kanal Schnittstellenparameter von der Karte zur Maschine
2. Grafische zeitnahe Betrachtung des aktuellen Indexes (Cursor/Ladebalken) von 2 Schnittstellenparametern (z.B. Achswert 1 / Vorschub aktiv Kanal 1)

5.5.5 Datenspeicherung während der Simulation

1. Zeitgenaue Protokollierung
 - a. der versendeten Schnittstellenparameter
 - b. der empfangenen Schnittstellenparameter

5.6 Auswertung der Simulationsergebnisse

5.6.3 Sicherung der Messdaten

1. Speichern der von der Genior/CTM Karte empfangenen Schnittstellenparameter in einer Datei
 - a. Speichern des Zeitstempels der letzten Auswertung in der Datei
 - b. Anlegen und bearbeiten der Header-Daten
 - c. Speichern der Datei sowohl vor als auch nach der Betrachtung der Messergebnisse ermöglichen

5.6.4 Darstellung der Messergebnisse

1. Grafische Gegenüberstellung mehrerer Schnittstellenparameter der Ein- und Ausgänge über die gesamte Messzeit, dabei wahlweise
 - a. Eine Grafik mit bis zu acht Ausgangssignalen und bis zu acht Eingangssignalen
 - b. Acht Grafiken mit je einem Ausgangssignal und einem Eingangssignal
2. Skalierbare Grafik
 - a. Einstellbare Grafikbreite
 - b. Einstellbare Grafikhöhe
 - c. Gesamter Zeitverlauf scrollbar
3. Achswerte der Eingänge und Sammelalarme der Ausgänge als Standarddarstellung
4. Signalnamen in den Grafiken darstellen
5. Bereits gespeicherte Dateien jederzeit wieder einlesen können
6. Input- und Outputquelldatei jederzeit konfigurierbar

5.6.5 Zeitverhalten

1. Zeitliches Verhalten der Genior/CTM Karte auf Ereignisse analysieren
 - a. z.B. Zeit vom Setzen eines Achswertes, der zu einem Bruchalarm führt, bis zum Setzen des PLC-Eingangs „Bruchalarm“

6 Anforderungen für zukünftige Erweiterungen

1. Durchführung von längere Messungen von bis zu 2 Stunden
2. Variabler einstellbarer Kommunikationstakt in Schritten von 1ms
3. Protokollierung
 - a. von Kommunikationsfehlern
 - b. ob sich Teilnehmer an und abgemeldet haben

6.2.3 Schnittstellenparameter während der Simulation verändern

1. DTA-Achswerte beeinflussen z.B.
 - a. Offset
 - b. Rauschen (Frequenz/Amplitude einstellbar)
2. Bits PLC-Kanäle ändern
 - a. z.B. Lernen Kanal 1, Reset Kanal 5, Nacharbeit Kanal 8 usw.
3. Globale PLC-Ausgänge ändern
 - a. z.B. Kommtakt (führt dazu, dass die DTA Signale mit einer anderen Samplerate abgetastet werden)

6.2.4 Simulatorverhalten auf Reaktion von Genior/CTM

1. Wie soll sich der Simulator auf eine Reaktion von Genior/CTM verhalten?
 - a. z.B. Es wird der Brucheingang gesetzt-> Simulator stoppt die Simulation, aber nur wenn dal PLC-Eingangsbit „ Alarme Aus“ nicht gesetzt ist.
2. Reaktion auf Taktänderungsanforderung von der Genior/CTM

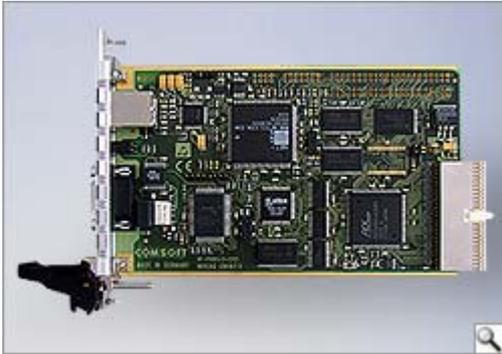
6.2.5 Simulation Maschinenbenutzerverhalten

Es soll das Verhalten eines Maschinennutzers simuliert werden (z.B. nach einem Bruch wird nach 5 Min Reset gedrückt und die Maschine fängt wieder an zu produzieren (in unserem Fall: die Sequenzen fangen wieder von Vorne an)

 Hochschule für Angewandte Wissenschaften Hamburg Hamburg University of Applied Sciences	Diplomarbeit Fachbereich Elektrotechnik und Informatik	
Wintersemester 07/08 Abgabetermin: 18.12.07	Givian Müller Matr.Nr.: 1749822	Betreuer: Professor Dr. Ing. Müller

II. Datenblatt DF-Profi2-PCI

DF PROFII II PCI/CPCI/PC104 High-performance PROFIBUS-DP/DPV1-Baugruppe



Die DF PROFII II ist eine sehr leistungsfähige PROFIBUS DP/DPV1-Baugruppe der neuesten Generation und ist in den Bauformen PCI, Compact-PCI und PC104+ erhältlich. Die DF PROFII II PCI-Baugruppe unterstützt 5V und 3.3V 32Bit-PCI-Slots, so dass auch ein Einsatz in schnellen Server-PC's und Workstations kein Problem darstellt. Neben den Standard PROFIBUS DP/DPV1-Funktionen verfügt die DF PROFII II über sinnvolle Zusatzfunktionen. Die Watchdog-Funktion mit hardwaremäßiger Abschaltung der PROFIBUS-Leitung ermöglicht Master Klasse 1 Redundanz-Konzepte mit gedoppelten PC-Systemen. Das Prozessdatenbild enthält neben den I/O-Daten zusätzlich einen millisekunden genauen Zeitstempel. Die DF PROFII II benötigt für den Austausch des Prozessdatenbildes mit der PC-Applikation bei gängigen Größen (1-3 KByte) nur ca. eine Millisekunde und sie speichert die PROFIBUS-Konfiguration spannungsausfallsicher in ihrem Flashspeicher. Für die Erstellung und den Download der PROFIBUS-Konfiguration steht ein leistungsfähiges und modernes Konfigurationstool zur Verfügung. Die DF PROF II unterstützt neben den gängigen Betriebssystemen Windows 2000/XP und sämtlichen Linux-Varianten (SUSE, Redhat, Fedora) auch Echtzeitbetriebssysteme wie LINUX-RT und LabVIEW-RT.

COMSOFT PROFIBUS-Konfigurationstool CONFIGURATOR II

CONFIGURATOR II ist leistungsfähiges PROFIBUS Konfigurationstool. Es entlastet den Anwender von den schwierigen PROFIBUS-Details und unterstützt neben der vollgraphischen GSD-basierten DP-Slavekonfiguration die komfortable Erstellung von OPC-Tags. Mit den umfangreichen Download, Analyse- und Steuerfunktionen ist eine sofortige Inbetriebnahme und Überprüfung des PROFIBUS-DP-Netzwerkes möglich. Die PROFIBUS-Konfiguration wird im XML-Format gespeichert, was einen problemlosen Zugriff von Third Party-Applikationen ermöglicht.

Technische Daten		
Bauformen	PCI (3.3V and 5V), Compact-PCI, PC104+	
Schnittstellen	Ethernet PROFIBUS	10/100BaseT (Optional) RS485 (DB9)
Baudraten	Ethernet PROFIBUS	max. 100 MBit/s RS485 max. 12 MBit/s
Unterstützte PROFIBUS Protokollvarianten	DPV0/DPV1	Master Klasse 1 (DPV0) und 2 (DPV1), DP-Slave
Hardware	Host-CPU PROFIBUS-Chip Statisches Ram Flash-Speicher PCI-Interface	NET+ARM 40 (33 MHz) ASPC 2 (48 MHz) 2 Mbyte 1 Mbyte PLX PCI-X-kompatibel (Betrieb in 5V und 3.3V-PCI-Slots)
Schicht 2 Dienste	Live List	
DP Dienste	DPV0 Master Klasse 1 und DP-Slave	
DP/DPV1 Dienste	Klasse 2 Master-Slave MSAC2_initiate, MSAC2_read, MSAC2_write, MSAC2_data_transport, MSAC2_abort	
Prozessdatenbild	Max. 8 KByte	
Unterstützte Betriebssysteme	Windows 2000/XP, Linux, Linux RT, LabView PROFIBUS VISA Treiber für LabView RT, LabView PROFIBUS VISA Treiber für Windows 2000/XP	
Add on Software	Redundanter OPC-Server, FDT-1.2 Kommunikations-DTM	

III. Profibusadrestabellen

Profibus Adressen			
CTM	Maschine zur Karte		
Adresse	Bedeutung	Format	Kanal
0.0	Typ des Datenblockes	Byte	Global
1.0	Länge des Datenblockes	Byte	Global
2.0	Anzahl der übertragenen Achsen	Byte	Global
3.0	Quittung für Änderung	Byte	Global
4.0	Aktueller Kommunikationstakt	Word	Global
6.0	Füllbyte	Byte	Global
7.0	Ergebnis Geschwindigkeitsänderung	Byte	Global
8.0	Achswert Achse 1	Dword	Global
12.0	Achswert Achse 2	Dword	Global
16.0	Achswert Achse 3	Dword	Global
20.0	Achswert Achse 4	Dword	Global
24.0	Achswert Achse 5	Dword	Global
28.0	Achswert Achse 6	Dword	Global
32.0	Achswert Achse 7	Dword	Global
36.0	Achswert Achse 8	Dword	Global
40.0	Dynamische Achsen	Byte	Global
41.0	Nummer für Grundeinstellung lernen	Byte	Global
42.0	.0 Lernen-Nacharbeit ausschalten	Bit	Global
	.1	Bit	
	.2	Bit	
	.3 Screenshot speichern	Bit	Global
	.4 Lernen-Nacharbeit ausschalten	Bit	Kanal 1
	.5 Lernen-Nacharbeit ausschalten	Bit	Kanal 2
	.6 Lernen-Nacharbeit ausschalten	Bit	Kanal 3
	.7 Lernen-Nacharbeit ausschalten	Bit	Kanal 4
51.0	Strobebyte Crashrekorder	Byte	Global
52.0	Crashrekorder Datenfeld 1	Byte	Global
53.0	Crashrekorder Datenfeld 2	Byte	Global
54.0	Crashrekorder Datenfeld 3	Byte	Global
55.0	Crashrekorder Datenfeld 4	Byte	Global
56.0	Crashrekorder Datenfeld 5	Byte	Global
57.0	Crashrekorder Datenfeld 6	Byte	Global
58.0	Crashrekorder Datenfeld 7	Byte	Global
59.0	Crashrekorder Datenfeld 8	Byte	Global
60.0	Crashrekorder Datenfeld 9	Byte	Global
61.0	Crashrekorder Datenfeld 10	Byte	Global
62.0	Crashrekorder Datenfeld 11	Byte	Global
63.0	Crashrekorder Datenfeld 12	Byte	Global
64.0	.0 Werkzeugwechsel	Bit	Kanal 1
	.1 Lernen	Bit	Kanal 1
	.2 Reset	Bit	Kanal 1
	.3 Vorschub aktiv	Bit	Kanal 1
	.4 Nacharbeit	Bit	Kanal 1
	.5 Zählen	Bit	Kanal 1
	.6 Spindelüberwachung	Bit	Kanal 1
	.7 Alarme aus	Bit	Kanal 1
65.0	Achsnummer	Byte	Kanal 1
66.0	Programmnummer	Short	Kanal 1
68.0	T-Nummer	Short	Kanal 1
70.0	D-Nummer	Short	Kanal 1
72.0	Bearbeitungsnummer	Short	Kanal 1
74.0	.0 Werkzeugwechsel	Bit	Kanal 2
	.1 Lernen	Bit	Kanal 2
	.2 Reset	Bit	Kanal 2
	.3 Vorschub aktiv	Bit	Kanal 2
	.4 Nacharbeit	Bit	Kanal 2
	.5 Zählen	Bit	Kanal 2
	.6 Spindelüberwachung	Bit	Kanal 2
	.7 Alarme aus	Bit	Kanal 2
75.0	Achsnummer	Byte	Kanal 2

76.0	Programmnummer	Short	Kanal 2
78.0	T-Nummer	Short	Kanal 2
80.0	D-Nummer	Short	Kanal 2
82.0	Bearbeitungsnummer	Short	Kanal 2
84.0	.0 Werkzeugwechsel	Bit	Kanal 3
	.1 Lernen	Bit	Kanal 3
	.2 Reset	Bit	Kanal 3
	.3 Vorschub aktiv	Bit	Kanal 3
	.4 Nacharbeit	Bit	Kanal 3
	.5 Zählen	Bit	Kanal 3
	.6 Spindelüberwachung	Bit	Kanal 3
	.7 Alarme aus	Bit	Kanal 3
85.0	Achsnummer	Byte	Kanal 3
86.0	Programmnummer	Short	Kanal 3
88.0	T-Nummer	Short	Kanal 3
90.0	D-Nummer	Short	Kanal 3
92.0	Bearbeitungsnummer	Short	Kanal 3
94.0	.0 Werkzeugwechsel	Bit	Kanal 4
	.1 Lernen	Bit	Kanal 4
	.2 Reset	Bit	Kanal 4
	.3 Vorschub aktiv	Bit	Kanal 4
	.4 Nacharbeit	Bit	Kanal 4
	.5 Zählen	Bit	Kanal 4
	.6 Spindelüberwachung	Bit	Kanal 4
	.7 Alarme aus	Bit	Kanal 4
95.0	Achsnummer	Byte	Kanal 4
96.0	Programmnummer	Short	Kanal 4
98.0	T-Nummer	Short	Kanal 4
100.0	D-Nummer	Short	Kanal 4
102.0	Bearbeitungsnummer	Short	Kanal 4

Profibus Adressen			
CTM	Karte zur Maschine		
Adresse	Bedeutung	Format	Kanal
0.0	Typ des Datenblockes	Byte	Global
1.0	Länge des Datenblockes	Byte	Global
2.0	Anforderter Kommunikationstakt	Word	Global
4.0	Änderungsanforderung	Byte	Global
5.0	Füllbyte	Byte	Global
6.0	Schwelle Geschwindigkeitsänderung	Word	Global
8.0	Zeit für Mittelwertbildung Achse 1	Word	Global
10.0	Achswert Achse 1	Byte	Global
11.0	Füllbyte	Byte	Global
12.0	Zeit für Mittelwertbildung Achse 2	Word	Global
14.0	Achswert Achse 2	Byte	Global
15.0	Füllbyte	Byte	Global
16.0	Zeit für Mittelwertbildung Achse 3	Word	Global
18.0	Achswert Achse 3	Byte	Global
19.0	Füllbyte	Byte	Global
20.0	Zeit für Mittelwertbildung Achse 4	Word	Global
22.0	Achswert Achse 4	Byte	Global
23.0	Füllbyte	Byte	Global
24.0	Zeit für Mittelwertbildung Achse 5	Word	Global
26.0	Achswert Achse 5	Byte	Global
27.0	Füllbyte	Byte	Global
28.0	Zeit für Mittelwertbildung Achse 6	Word	Global
30.0	Achswert Achse 6	Byte	Global
31.0	Füllbyte	Byte	Global
32.0	Zeit für Mittelwertbildung Achse 7	Word	Global
34.0	Achswert Achse 7	Byte	Global
35.0	Füllbyte	Byte	Global
36.0	Zeit für Mittelwertbildung Achse 8	Word	Global
38.0	Achswert Achse 8	Byte	Global

39.0	Füllbyte	Byte	Global
40.0	Systemfehler	Bit	Kanal 1
	.1 Bruch-Alarm	Bit	Kanal 1
	.2 Stumpf-Alarm	Bit	Kanal 1
	.3 Fehlt-Alarm	Bit	Kanal 1
	.4 Tiefe-Alarm	Bit	Kanal 1
	.5 Materialberührung	Bit	Kanal 1
	.6 Alarme Aus	Bit	Kanal 1
	.7 Sammelalarm	Bit	Kanal 1
41.0	Systemfehler	Bit	Kanal 2
	.1 Bruch-Alarm	Bit	Kanal 2
	.2 Stumpf-Alarm	Bit	Kanal 2
	.3 Fehlt-Alarm	Bit	Kanal 2
	.4 Tiefe-Alarm	Bit	Kanal 2
	.5 Materialberührung	Bit	Kanal 2
	.6 Alarme Aus	Bit	Kanal 2
	.7 Sammelalarm	Bit	Kanal 2
42.0	Systemfehler	Bit	Kanal 3
	.1 Bruch-Alarm	Bit	Kanal 3
	.2 Stumpf-Alarm	Bit	Kanal 3
	.3 Fehlt-Alarm	Bit	Kanal 3
	.4 Tiefe-Alarm	Bit	Kanal 3
	.5 Materialberührung	Bit	Kanal 3
	.6 Alarme Aus	Bit	Kanal 3
	.7 Sammelalarm	Bit	Kanal 3
43.0	Systemfehler	Bit	Kanal 4
	.1 Bruch-Alarm	Bit	Kanal 4
	.2 Stumpf-Alarm	Bit	Kanal 4
	.3 Fehlt-Alarm	Bit	Kanal 4
	.4 Tiefe-Alarm	Bit	Kanal 4
	.5 Materialberührung	Bit	Kanal 4
	.6 Alarme Aus	Bit	Kanal 4
	.7 Sammelalarm	Bit	Kanal 4
44.0	Stumpf-Voralarm	Bit	Kanal 1
45.0	Stumpf-Voralarm	Bit	Kanal 2
46.0	Stumpf-Voralarm	Bit	Kanal 3
47.0	Stumpf-Voralarm	Bit	Kanal 4

Profibus Adressen			
Genior	Maschine zur Karte		
Adresse	Bedeutung	Format	Kanal
0.0	Typ des Datenblockes	Byte	Global
1.0	Länge des Datenblockes	Byte	Global
2.0	Anzahl der übertragenen Achsen	Byte	Global
3.0	Quittung für Änderung	Byte	Global
4.0	Aktueller Kommunikationstakt	Word	Global
6.0	Füllbyte	Byte	Global
7.0	Ergebnis Geschwindigkeitsänderung	Byte	Global
8.0	Achswert Achse 1	Dword	Global
12.0	Achswert Achse 2	Dword	Global
16.0	Achswert Achse 3	Dword	Global
20.0	Achswert Achse 4	Dword	Global
24.0	Achswert Achse 5	Dword	Global
28.0	Achswert Achse 6	Dword	Global
32.0	Achswert Achse 7	Dword	Global
36.0	Achswert Achse 8	Dword	Global
72.0	Dynamische Achsen	Byte	Global
88.0	Werkzeugwechsel	Bit	Kanal 1
	.1 Lernen	Bit	Kanal 1
	.2 Reset	Bit	Kanal 1
	.3 Vorschub aktiv	Bit	Kanal 1
	.4 Nacharbeit	Bit	Kanal 1
	.5 Zählen	Bit	Kanal 1
	.6	Bit	Kanal 1
	.7 Alarme aus	Bit	Kanal 1
89.0		Byte	Kanal 1
90.0	Programmnummer	Short	Kanal 1

92.0	T-Nummer	Short	Kanal 1
94.0	D-Nummer	Short	Kanal 1
96.0	Bearbeitungsnummer	Short	Kanal 1
98.0		Short	Kanal 1
100.0	AC Aus	Bit	Kanal 1
101.0		Byte	Kanal 1
102.0	Werkzeugwechsel	Bit	Kanal 2
	.1 Lernen	Bit	Kanal 2
	.2 Reset	Bit	Kanal 2
	.3 Vorschub aktiv	Bit	Kanal 2
	.4 Nacharbeit	Bit	Kanal 2
	.5 Zählen	Bit	Kanal 2
	.6	Bit	Kanal 2
	.7 Alarme aus	Bit	Kanal 2
103.0		Byte	Kanal 2
104.0	Programmnummer	Short	Kanal 2
106.0	T-Nummer	Short	Kanal 2
108.0	D-Nummer	Short	Kanal 2
110.0	Bearbeitungsnummer	Short	Kanal 2
112.0		Short	Kanal 2
114.0	AC Aus	Bit	Kanal 2
115.0		Byte	Kanal 2
116.0	Werkzeugwechsel	Bit	Kanal 3
	.1 Lernen	Bit	Kanal 3
	.2 Reset	Bit	Kanal 3
	.3 Vorschub aktiv	Bit	Kanal 3
	.4 Nacharbeit	Bit	Kanal 3
	.5 Zählen	Bit	Kanal 3
	.6	Bit	Kanal 3
	.7 Alarme aus	Bit	Kanal 3
117.0		Byte	Kanal 3
118.0	Programmnummer	Short	Kanal 3
120.0	T-Nummer	Short	Kanal 3
122.0	D-Nummer	Short	Kanal 3
124.0	Bearbeitungsnummer	Short	Kanal 3
126.0		Short	Kanal 3
128.0	AC Aus	Bit	Kanal 3
129.0		Byte	Kanal 3
130.0	Werkzeugwechsel	Bit	Kanal 4
	.1 Lernen	Bit	Kanal 4
	.2 Reset	Bit	Kanal 4
	.3 Vorschub aktiv	Bit	Kanal 4
	.4 Nacharbeit	Bit	Kanal 4
	.5 Zählen	Bit	Kanal 4
	.6	Bit	Kanal 4
	.7 Alarme aus	Bit	Kanal 4
131.0		Byte	Kanal 4
132.0	Programmnummer	Short	Kanal 4
134.0	T-Nummer	Short	Kanal 4
136.0	D-Nummer	Short	Kanal 4
138.0	Bearbeitungsnummer	Short	Kanal 4
140.0		Short	Kanal 4
142.0	AC Aus	Bit	Kanal 4
143.0		Byte	Kanal 4
144.0	Werkzeugwechsel	Bit	Kanal 5
	.1 Lernen	Bit	Kanal 5
	.2 Reset	Bit	Kanal 5
	.3 Vorschub aktiv	Bit	Kanal 5
	.4 Nacharbeit	Bit	Kanal 5
	.5 Zählen	Bit	Kanal 5
	.6	Bit	Kanal 5
	.7 Alarme aus	Bit	Kanal 5
145.0		Byte	Kanal 5
146.0	Programmnummer	Short	Kanal 5
148.0	T-Nummer	Short	Kanal 5
150.0	D-Nummer	Short	Kanal 5
152.0	Bearbeitungsnummer	Short	Kanal 5

154.0		Short	Kanal 5
156.0	.0 AC Aus	Bit	Kanal 5
157.0		Byte	Kanal 5
158.0	.0 Werkzeugwechsel	Bit	Kanal 6
	.1 Lernen	Bit	Kanal 6
	.2 Reset	Bit	Kanal 6
	.3 Vorschub aktiv	Bit	Kanal 6
	.4 Nacharbeit	Bit	Kanal 6
	.5 Zählen	Bit	Kanal 6
	.6	Bit	Kanal 6
	.7 Alarme aus	Bit	Kanal 6
159.0		Byte	Kanal 6
160.0	Programmnummer	Short	Kanal 6
162.0	T-Nummer	Short	Kanal 6
164.0	D-Nummer	Short	Kanal 6
166.0	Bearbeitungsnummer	Short	Kanal 6
168.0		Short	Kanal 6
170.0	.0 AC Aus	Bit	Kanal 6
171.0		Byte	Kanal 6
172.0	.0 Werkzeugwechsel	Bit	Kanal 7
	.1 Lernen	Bit	Kanal 7
	.2 Reset	Bit	Kanal 7
	.3 Vorschub aktiv	Bit	Kanal 7
	.4 Nacharbeit	Bit	Kanal 7
	.5 Zählen	Bit	Kanal 7
	.6	Bit	Kanal 7
	.7 Alarme aus	Bit	Kanal 7
173.0		Byte	Kanal 7
174.0	Programmnummer	Short	Kanal 7
176.0	T-Nummer	Short	Kanal 7
178.0	D-Nummer	Short	Kanal 7
180.0	Bearbeitungsnummer	Short	Kanal 7
182.0		Short	Kanal 7
184.0	.0 AC Aus	Bit	Kanal 7
185.0		Byte	Kanal 7
186.0	.0 Werkzeugwechsel	Bit	Kanal 8
	.1 Lernen	Bit	Kanal 8
	.2 Reset	Bit	Kanal 8
	.3 Vorschub aktiv	Bit	Kanal 8
	.4 Nacharbeit	Bit	Kanal 8
	.5 Zählen	Bit	Kanal 8
	.6	Bit	Kanal 8
	.7 Alarme aus	Bit	Kanal 8
187.0		Byte	Kanal 8
188.0	Programmnummer	Short	Kanal 8
190.0	T-Nummer	Short	Kanal 8
192.0	D-Nummer	Short	Kanal 8
194.0	Bearbeitungsnummer	Short	Kanal 8
196.0		Short	Kanal 8
198.0	.0 AC Aus	Bit	Kanal 8
199.0		Byte	Kanal 8

Profibus Adressen			
Genior	Karte zur Maschine		
Adresse	Bedeutung	Format	Kanal
0.0	Typ des Datenblockes	Byte	Global
1.0	Länge des Datenblockes	Byte	Global
2.0	Angefordertes Kommunikationsstakt	Word	Global
4.0	Änderungsanforderung	Byte	Global
5.0	Füllbyte	Byte	Global
6.0	Schwelle Geschwindigkeitsänderung	Word	Global
8.0	Zeit für Mittelwertbildung Achse 1	Word	Global
10.0	Achswert Achse 1	Byte	Global
11.0	Füllbyte	Byte	Global
12.0	Zeit für Mittelwertbildung Achse 2	Word	Global
14.0	Achswert Achse 2	Byte	Global
15.0	Füllbyte	Byte	Global

16.0		Zeit für Mittelwertbildung Achse 3	Word	Global
18.0		Achswert Achse 3	Byte	Global
19.0		Füllbyte	Byte	Global
20.0		Zeit für Mittelwertbildung Achse 4	Word	Global
22.0		Achswert Achse 4	Byte	Global
23.0		Füllbyte	Byte	Global
24.0		Zeit für Mittelwertbildung Achse 5	Word	Global
26.0		Achswert Achse 5	Byte	Global
27.0		Füllbyte	Byte	Global
28.0		Zeit für Mittelwertbildung Achse 6	Word	Global
30.0		Achswert Achse 6	Byte	Global
31.0		Füllbyte	Byte	Global
32.0		Zeit für Mittelwertbildung Achse 7	Word	Global
34.0		Achswert Achse 7	Byte	Global
35.0		Füllbyte	Byte	Global
36.0		Zeit für Mittelwertbildung Achse 8	Word	Global
38.0		Achswert Achse 8	Byte	Global
39.0		Füllbyte	Byte	Global
40.0	.0	Pulsschlag der Karte	Bit	Global
41.0	.0	Überwachung aktiv	Bit	Global
	.1	Datenbank wird synchronisiert	Bit	Global
60.0	.0	Systemfehler	Bit	Kanal 1
	.1	Bruch-Alarm	Bit	Kanal 1
	.2	Stumpf-Alarm	Bit	Kanal 1
	.3	Fehlt-Alarm	Bit	Kanal 1
	.4	Tiefe-Alarm	Bit	Kanal 1
	.5	Materialberührung	Bit	Kanal 1
	.6	Alarme Aus	Bit	Kanal 1
	.7	Sammelalarm	Bit	Kanal 1
61.0	.0	Lernen	Bit	Kanal 1
62.0	.0	Stumpf Voralarm	Bit	Kanal 1
	.1	Zähler einzeln	Bit	Kanal 1
	.2	Zähler gesamt	Bit	Kanal 1
	.3	Quittierung Vorschub aktiv	Bit	Kanal 1
63.0		AC Reglerausgang	Byte	Kanal 1
70.0	.0	Systemfehler	Bit	Kanal2
	.1	Bruch-Alarm	Bit	Kanal2
	.2	Stumpf-Alarm	Bit	Kanal2
	.3	Fehlt-Alarm	Bit	Kanal2
	.4	Tiefe-Alarm	Bit	Kanal2
	.5	Materialberührung	Bit	Kanal2
	.6	Alarme Aus	Bit	Kanal2
	.7	Sammelalarm	Bit	Kanal2
71.0	.0	Lernen	Bit	Kanal2
62.0	.0	Stumpf Voralarm	Bit	Kanal2
	.1	Zähler einzeln	Bit	Kanal2
	.2	Zähler gesamt	Bit	Kanal2
	.3	Quittierung Vorschub aktiv	Bit	Kanal2
73.0		AC Reglerausgang	Byte	Kanal2
80.0	.0	Systemfehler	Bit	Kanal 3
	.1	Bruch-Alarm	Bit	Kanal 3
	.2	Stumpf-Alarm	Bit	Kanal 3
	.3	Fehlt-Alarm	Bit	Kanal 3
	.4	Tiefe-Alarm	Bit	Kanal 3
	.5	Materialberührung	Bit	Kanal 3
	.6	Alarme Aus	Bit	Kanal 3
	.7	Sammelalarm	Bit	Kanal 3
81.0	.0	Lernen	Bit	Kanal 3
82.0	.0	Stumpf Voralarm	Bit	Kanal 3
	.1	Zähler einzeln	Bit	Kanal 3
	.2	Zähler gesamt	Bit	Kanal 3
	.3	Quittierung Vorschub aktiv	Bit	Kanal 3
83.0		AC Reglerausgang	Byte	Kanal 3
90.0	.0	Systemfehler	Bit	Kanal 4
	.1	Bruch-Alarm	Bit	Kanal 4
	.2	Stumpf-Alarm	Bit	Kanal 4
	.3	Fehlt-Alarm	Bit	Kanal 4

	.4	Tiefe-Alarm	Bit	Kanal 4
	.5	Materialberührung	Bit	Kanal 4
	.6	Alarme Aus	Bit	Kanal 4
	.7	Sammelalarm	Bit	Kanal 4
91.0	.0	Lernen	Bit	Kanal 4
92.0	.0	Stumpf Voralarm	Bit	Kanal 4
	.1	Zähler einzeln	Bit	Kanal 4
	.2	Zähler gesamt	Bit	Kanal 4
	.3	Quittierung Vorschub aktiv	Bit	Kanal 4
93.0		AC Reglerausgang	Byte	Kanal 4
100.0	.0	Systemfehler	Bit	Kanal 5
	.1	Bruch-Alarm	Bit	Kanal 5
	.2	Stumpf-Alarm	Bit	Kanal 5
	.3	Fehlt-Alarm	Bit	Kanal 5
	.4	Tiefe-Alarm	Bit	Kanal 5
	.5	Materialberührung	Bit	Kanal 5
	.6	Alarme Aus	Bit	Kanal 5
	.7	Sammelalarm	Bit	Kanal 5
101.0	.0	Lernen	Bit	Kanal 5
102.0	.0	Stumpf Voralarm	Bit	Kanal 5
	.1	Zähler einzeln	Bit	Kanal 5
	.2	Zähler gesamt	Bit	Kanal 5
	.3	Quittierung Vorschub aktiv	Bit	Kanal 5
103.0		AC Reglerausgang	Byte	Kanal 5
110.0	.0	Systemfehler	Bit	Kanal 6
	.1	Bruch-Alarm	Bit	Kanal 6
	.2	Stumpf-Alarm	Bit	Kanal 6
	.3	Fehlt-Alarm	Bit	Kanal 6
	.4	Tiefe-Alarm	Bit	Kanal 6
	.5	Materialberührung	Bit	Kanal 6
	.6	Alarme Aus	Bit	Kanal 6
	.7	Sammelalarm	Bit	Kanal 6
111.0	.0	Lernen	Bit	Kanal 6
112.0	.0	Stumpf Voralarm	Bit	Kanal 6

	.1	Zähler einzeln	Bit	Kanal 6
	.2	Zähler gesamt	Bit	Kanal 6
	.3	Quittierung Vorschub aktiv	Bit	Kanal 6
113.0		AC Reglerausgang	Byte	Kanal 6
120.0	.0	Systemfehler	Bit	Kanal 7
	.1	Bruch-Alarm	Bit	Kanal 7
	.2	Stumpf-Alarm	Bit	Kanal 7
	.3	Fehlt-Alarm	Bit	Kanal 7
	.4	Tiefe-Alarm	Bit	Kanal 7
	.5	Materialberührung	Bit	Kanal 7
	.6	Alarme Aus	Bit	Kanal 7
	.7	Sammelalarm	Bit	Kanal 7
121.0	.0	Lernen	Bit	Kanal 7
122.0	.0	Stumpf Voralarm	Bit	Kanal 7
	.1	Zähler einzeln	Bit	Kanal 7
	.2	Zähler gesamt	Bit	Kanal 7
	.3	Quittierung Vorschub aktiv	Bit	Kanal 7
123.0		AC Reglerausgang	Byte	Kanal 7
130.0	.0	Systemfehler	Bit	Kanal 8
	.1	Bruch-Alarm	Bit	Kanal 8
	.2	Stumpf-Alarm	Bit	Kanal 8
	.3	Fehlt-Alarm	Bit	Kanal 8
	.4	Tiefe-Alarm	Bit	Kanal 8
	.5	Materialberührung	Bit	Kanal 8
	.6	Alarme Aus	Bit	Kanal 8
	.7	Sammelalarm	Bit	Kanal 8
131.0	.0	Lernen	Bit	Kanal 8
132.0	.0	Stumpf Voralarm	Bit	Kanal 8
	.1	Zähler einzeln	Bit	Kanal 8
	.2	Zähler gesamt	Bit	Kanal 8
	.3	Quittierung Vorschub aktiv	Bit	Kanal 8
133.0		AC Reglerausgang	Byte	Kanal 8

 Hochschule für Angewandte Wissenschaften Hamburg Hamburg University of Applied Sciences	Diplomarbeit Fachbereich Elektrotechnik und Informatik	
Wintersemester 07/08 Abgabetermin: 18.12.07	Givian Müller Matr.Nr.: 1749822	Betreuer: Professor Dr. Ing. Müller

IV. Ablaufdiagramm

Diagrammlegende



Hauptmodule die vom Nutzer der Software gestartet werden und in einer Endlosschleife bis zum Schließen des Programms laufen.



Unterprogramme die von den Hauptmodulen aufgerufen werden. Der Nutzer der Software hat keinen direkten Zugriff auf diese Module.



Dauerhaft aufgerufene Module.



Module die erst nach einer Eingabe des Nutzers der Software aktiv werden.



Dateien die von den Modulen angesprochen werden.



Variablen die von den Modulen angesprochen werden.



Aufrufen eines Untermoduls und Übergeben eines Wertes .



Aufruf eines Untermoduls, übergeben eines Wertes und erhalten einer Antwort des Untermoduls.



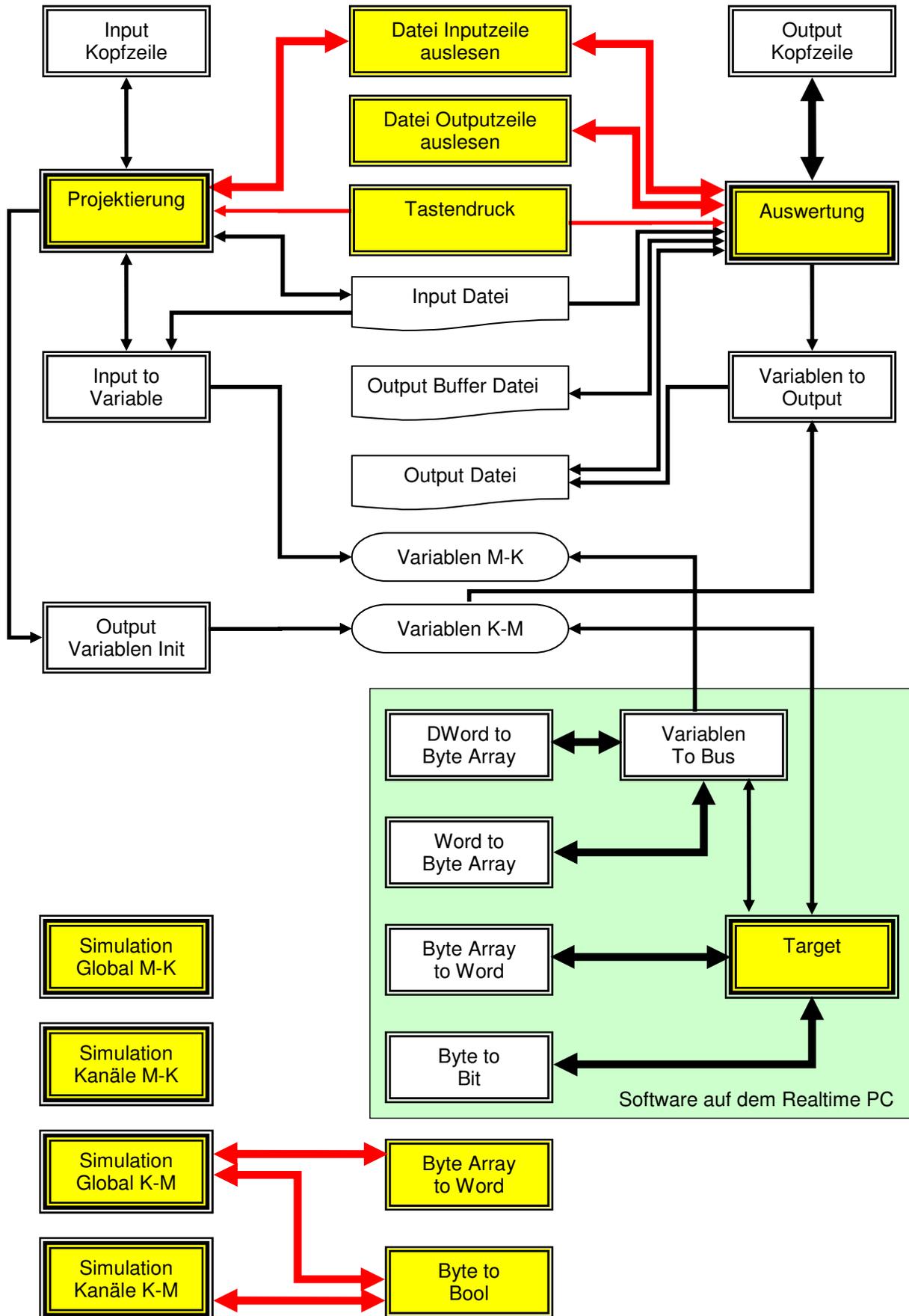
Mehrfaches Aufrufen eines Untermoduls innerhalb eines Schleifendurchlaufs.



Kontinuierliches Aufrufen eines Untermoduls innerhalb eines Schleifendurchlaufs.



Kontinuierliches mehrfaches Aufrufen eines Untermoduls innerhalb eines Schleifendurchlaufs.



 Hochschule für Angewandte Wissenschaften Hamburg Hamburg University of Applied Sciences	Diplomarbeit Fachbereich Elektrotechnik und Informatik	
Wintersemester 07/08 Abgabetermin: 18.12.07	Givian Müller Matr.Nr.: 1749822	Betreuer: Professor Dr. Ing. Müller

Danksagung

Diese Diplomarbeit entstand bei der Artis GmbH im Bereich der Forschung und Entwicklung.

Ich möchte mich an dieser Stelle bei meinem sehr geehrten Prof. Dr. Ing. Müller für die hervorragende Betreuung bedanken.

Besonders möchte ich mich für die Betreuung die mir während der gesamten Arbeit gewährt wurden bei Dipl. Ing. V. Redecker, Dr. Ing. D. Lange sowie allen weiteren Mitarbeitern der Abteilung Forschung und Entwicklung bedanken.

Ebenso möchte ich mich bei meiner Familie für die Unterstützung und vor allem für die Geduld bedanken.

 Hochschule für Angewandte Wissenschaften Hamburg Hamburg University of Applied Sciences	Diplomarbeit Fachbereich Elektrotechnik und Informatik	ARTIS [®] 
Wintersemester 07/08 Abgabetermin: 18.12.07	Givian Müller Matr.Nr.: 1749822	Betreuer: Professor Dr. Ing. Müller

Versicherung über die Selbständigkeit

Hiermit versichere ich, dass ich die vorliegende Arbeit im Sinne der Prüfungsordnung nach §25(4) ohne fremde Hilfe selbständig verfasst und nur die angegebenen Hilfsmittel benutzt habe. Wörtlich oder dem Sinn nach aus anderen Werken entnommene Stellen habe ich unter Angabe der Quellen kenntlich gemacht.

Munster 18.12.2007

Unterschrift