

Institut für Rechnergestützte Ingenieursysteme
Universität Stuttgart
Universitätsstraße 38
70569 Stuttgart
Deutschland

Diplomarbeit Nr. 3404

**Entwickeln und bewerten eines Konzeptes
zur effizienten Programmierung von
Koordinatenmessgeräten zur Messung
verschiedener Bauteilpaletten in einer
Aufspannung**

Jie Li

Studiengang: Informatik

Prüfer: Univ-Prof. Hon-Prof. Dr. Dieter Roller

Betreuer: Univ-Prof. Hon-Prof. Dr. Dieter Roller

Betreuer: Dipl.-Inf. Hans-Peter Eisele (Daimler AG)

begonnen am: 02.11.2012

beendet am: 04.05.2013

CR-Klassifikation: D.2.2 H.5.2 J.7

Danksagung

Die vorliegende Arbeit entstand in einer Kooperation zwischen dem Institut für Rechnergestützte Ingenieursysteme der Universität Stuttgart und der Abteilung TP/PPH Daimler AG in Stuttgart.

Bei Univ-Prof. Hon-Prof. Dr. Dieter Roller bedanke ich mich sehr herzlich für die Möglichkeit zur Diplomarbeit an seinem Lehrstuhl, für seine Betreuung und seine kontinuierliche Unterstützung dieser Arbeit, insbesondere dabei für die wertvollen fachlichen Hinweise und Ratschläge.

Weiterhin möchte ich mich bei meinem Betreuer Herrn Dipl.-Ing. Hans-Peter Eisele für die Aufnahme in seiner Abteilung während meiner Tätigkeit bei der Daimler AG sehr herzlich bedanken. Herrn Dipl.-Ing. Hans-Peter Eisele und Herrn Michael Krauter gilt mein besonderer Dank für die Einarbeitung und Bereitschaft zur stetigen Beratung und wertvolle Unterstützung bei fachlichen Themen. Allen weiteren Kollegen der Abteilung TP/PPH möchte ich nicht zuletzt auch für das hervorragende Arbeitsklima und ihre Unterstützung danken.

Erklärung

Hiermit versichere ich, diese Arbeit selbstständig verfasst und nur die angegebenen Quellen benutzt zu haben. Wörtliche und sinngemäße Übernahmen aus anderen Quellen habe ich nach bestem Wissen und Gewissen als solche kenntlich gemacht.

Stuttgart, 04.2013

Kurzfassung

Im Laufe der Entwicklung von Verbrennungsmotoren werden geometrische Messdaten der Motorbauteile zu verschiedenen Zeitpunktes der Lebensdauer eines Motors gemessen und bewertet. Eine wichtige Information dabei ist neben der Maßhaltigkeit die Ermittlung von Spielen und Verschleiß aus der Paarung jeweils passender Messwerte entsprechender Motorbauteile.

Für CNC-gestützte Messmaschinen liegt es nahe, durch entsprechende Messprogramm-konfiguration auch die bedienerlose Zeit weitestgehend zu nutzen. Dazu hat man in der Vergangenheit für gleiche Bauteile sogenannte Palettenmessungen eingeführt, die auch für Einzelbauteile eingesetzt werden können. Bei entsprechender Maschinengröße entsteht der Wunsch, durch passende gleichzeitige Anordnung von Paletten unterschiedlicher Bauteile die Maschinenlaufzeit in die bedienerlose Zeit noch konsequenter auszudehnen. Hierfür soll eine passende Systematik auf der Basis der vorhandenen "Einzel"-Palettenprogramme unter Berücksichtigung der zur Messvorbereitung erforderlichen Informationen entwickelt und bewertet werden.

Inhaltsverzeichnis

1	Einleitung	1
1.1	Zielsetzung.....	1
1.2	Problemstellung	2
1.3	Aufbau der Arbeit.....	3
2	Theoretische Grundlagen	5
2.1	Cax-Technologien	5
2.1.1	Abkürzung aus der CAX-Welt.....	5
2.1.2	Modellarten	7
2.1.3	Volumenmodellierung.....	9
2.1.4	Struktur der Bauteilen	12
2.1.5	Koordinatensystem des zumessenden Bauteils	13
2.2	Grundlagen und Stand der Messtechnik	14
2.2.1	Aufbau der 3D Koordinatenmessger äte	14
2.2.2	Tastkopfsystem.....	16
2.2.3	Taststift und Taster.....	17
2.2.4	Tasteraufnahme	18
2.2.5	Tasterwechselbank und Tasterablage.....	19
2.2.6	Würfel und Gelenkst ücke	19
2.2.7	Konfigurationsdiagramm des 3D-Tastsystems	20
2.3	Koordinatenmessger äte bei Abteilung TP/PPH und ihre Randbedingungen	21
2.3.1	Alle verwendeten Tasterb äumen für entsprechenden Bauteilen	21
2.3.2	Verwendete Taster je Bauteil	23
2.3.3	Messfl äche und Messvolumen von KMG	25
2.4	Der theoretischer Messprozess	27
2.5	Programmiersprache "QUINDOS 7"	28
3	Aktueller Messprozess und Entwurf des neuen Verfahrens	30
3.1	Der praktische Messprozess der aktuellen CNC-Messprogramme	30
3.2	Entwurf eines flexiblen und effizienten Messprozesses	33
3.2.1	Auswahlmen ü von Multibauteilmessprogramm	34
3.2.2	Eingaben der Messkenndaten für jeden Bauteil	35
3.2.3	Taster Einmessung für alle Bauteile.....	36
3.2.4	Manuelle Ausrichtung und Messaufgabe Eingabe.....	37
3.2.5	Automatische Messung und Messprotokoll Erstellung.....	37
3.2.6	Wiederholungsmessung	38
3.3	Neues Multibauteilmessprogramm im Vergleich zum aktuellen Messprogramm	39
3.3.1	Messungslist von einen Tag im ganzen Messungsraum	39

3.3.2	Messungslist von Einer Messmaschine in ein paaren Tage	40
3.3.3	Bewertung für die Idee des Multimessprogramms	41
4	Ergebnisse der neuen Messprogramm für Motorenmessung	42
4.1	Versuchsbeispiel Nr.1	42
4.2	Versuchsbeispiel Nr.2.....	47
4.3	Versuchsbeispiel Nr.3.....	54
4.4	Weitere Versuchsergebnisse	59
4.5	Bewertung des entwickelten Verfahrens	60
5	Zusammenfassung und Ausblick	62
A	Anh änge.....	63
A 1:	Messprogramme für Auswahlmen ü.....	63
A 2:	Messprogramme für die Eingabe der Bauteilskennndaten.....	68
A 3:	Ein Beispielprogramme für Taster Einmessen.....	72
A 4:	Ein Beispielprogramme für Manuelle Ausrichtung	74
A 5:	Ein Beispielprogramme für Wiederholungsmessung.....	78
A 6:	Ein Beispielprogramme für Koordinatensystem Benutzung.....	80
	Abbildungsverzeichnis	82
	Tabellenverzeichnis.....	84
	Abk ürzungsverzeichnis	85
	Literaturverzeichnis.....	86

1 Einleitung

Die technische Weiterentwicklung und Optimierung des Verbrennungsmotors spielt in der in der Automobilindustrie neben der Entwicklung alternativer Antriebe eine wichtige Rolle. Obwohl konstruktiv das Grundprinzip des Verbrennungsmotors nur wenig verändert werden kann, reizt er seine Effizienzpotenziale noch nicht aus. Daher gibt es nach wie vor die eindeutige Entwicklungstendenz, bei neuen Fahrzeugmotoren sowohl Kraftstoffverbrauch als auch Abgasemission zu optimieren und damit zu minimieren [1].

Am Entwicklungszyklus in einem Unternehmen sind verschiedene Bereiche und Abteilung beteiligt. Jeder dieser Bereiche leistet seinen Beitrag zur Entwicklungszeitschiene und auch zur Kostenentwicklung. Deshalb sind alle Bereiche zu Kostenbewusstsein und Kostenoptimierung angehalten.

Die Kostenkomponenten, welche so direkt beeinflussbar sind, liegen sicher im prozessualen Bereich und in den entsprechenden Abläufen. Verrechnungssätze können meist nur indirekt beeinflusst werden, wenn man von der Anzahl der Arbeitsstunden absieht. Diese wiederum hängen direkt an den optimierbaren Prozessen und Abläufen. Hierbei werden alle Vorbereitungs-, Einsatz-, und Nachbereitungszeiten für einen Versuch sowie die Wartezeiten bei Messmaschinen betrachtet. Werden die Messdaten erst bei der Auswertung analysiert, wird der Zeit- und Personalaufwand durch die notwendigen Messungen oder Wiederholungsmessungen, insbesondere bei den dauerhaften Messungen, berechnet.

1.1 Zielsetzung

Das Ziel der vorliegenden Arbeit ist, den Einsatz der vorhandenen Ressourcen (Messmaschinen und Mitarbeiter) zu optimieren. Das kann erfolgen durch minimieren des für eine Messung erforderlichen Personalaufwandes oder durch Verlängerung der unbemannten Maschinenlaufzeiten. Insgesamt nehmen dadurch die Kosten für die Messung eines Motors ab.

Die hierfür verwendeten Messungen von Bauteilen sind ein Teil von "Truck Group Produktentstehungsprozess". In der Abbildung 1.1 hat es Hardwareprozess als Gelbfelder von ganzem Zyklus gezeigt. Hardware-Prozess bedeutet von der Programmplanung, über die Prototypbeschaffung und Montage bis zur Erprobung und Befund.

Die Abteilung TP/PPH bei Daimler AG wird für die Aufgaben "Befund" eingesetzt. Die geometrischen Messdaten für Konstruktion werden erst nach den Messungen bekommen. Mit diesen Messdaten können die Fachleute die Prototypen Serien weiter entwickeln. Jeder Bauteil muss mindestens zweimal gemessen werden. Einmal vor dem Versuch müssen die Bauteile Einbaumessung, die Qualität von Produkt prüft, durchgeführt werden. Nach dem Versuch müssen die Bauteile auch einmal Ausbaumessung (alternativ auch Zwischenmessung) die zur Einbaumessung vergleicht, durchgeführt werden.

Truck Group Produktentstehungsprozess

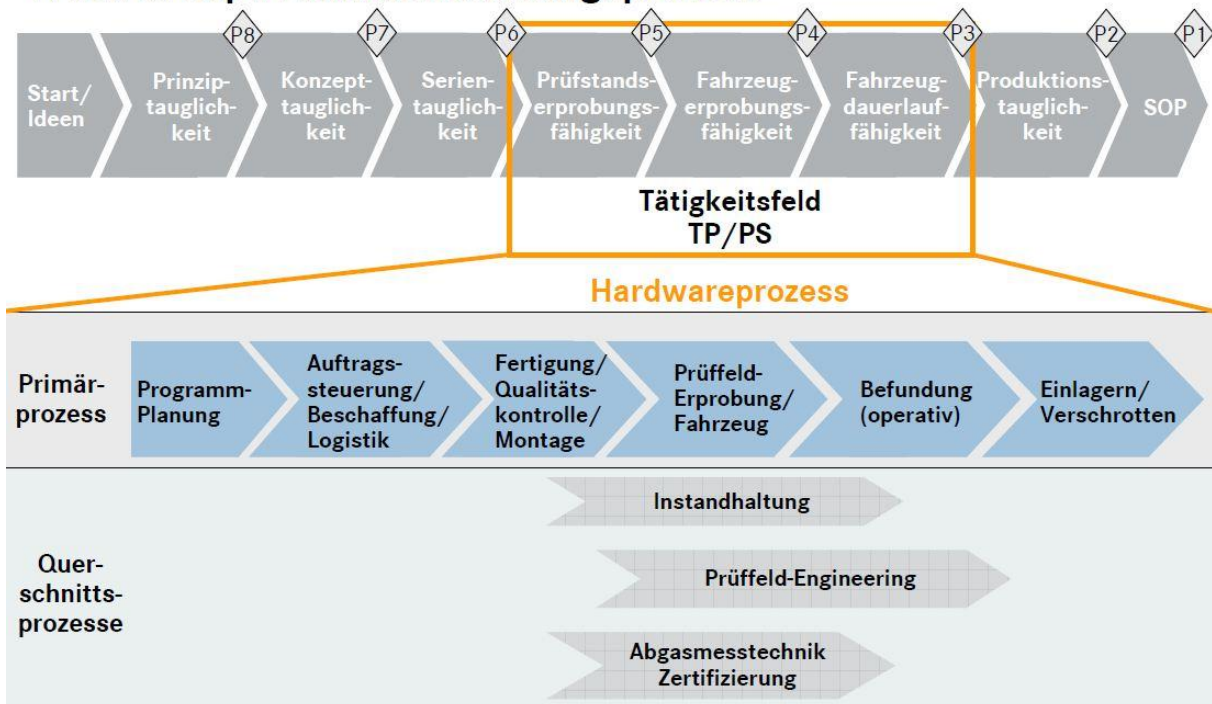


Abb. 1.1 Produktentstehungsprozess [2]

Ziel dieser Diplomarbeit ist der sogenannte Messungszeitaufwand zu optimieren. Dadurch kann die Dauer des Entwicklungszyklus verkürzt werden.

1.2 Problemstellung

Bisher werden alle Aufgaben im Bereich der geometrischen Koordinatenmesstechnik durch Einsatz einzelner, bauteilspezifischer CNC-Messprogramme abgedeckt. Diese Programme wurden teilweise selbst erstellt, teilweise in Form von Familienprogramm für sogenannte Bauteilfamilien käuflich erworben. Insofern ist eine heterogene Programmlandschaft entstanden, deren Teilprogramme dafür sorgen, dass durch Einhaltung gleicher Messstrategien vergleichbare Messwerte ermittelt werden. Eine optimierte Maschinenauslastung wurde dabei nur teilweise mitbetrachtet.

Die zu messenden Bauteile werden entweder in Form von Einzelmessungen vermessen, wobei in einigen Fällen durch gleichzeitiges Aufspannen gleichartiger Bauteile auf Paletten ein Optimierungsansatz zu erkennen ist. Diese Palettenmessungen sind bisher jedoch nicht für gemischte Bauteilbestückungen möglich. Um eine Maschinenbestückung mit verschiedenen Paletten für unterschiedliche Bauteile erreichen zu können, müssen verschiedene Randbedingungen eingehalten werden, wenn zielgerichtet eine deutliche Steigerung der unbemannten Maschinenlaufzeit erreichen soll. Es ist also wichtig, die einzuhaltenden Randbedingungen zu untersuchen, zu bewerten und für deren Einhaltung zu sorgen.

Die programmtechnische Umsetzung muss dazu in der Lage sein, aus den bisher isoliert ablaufenden Programm die relevanten Daten so zur Verfügung zu stellen, dass aus einer Gesamtheit immer die erforderlichen Informationen sicher zur Verarbeitung bereitstehen: Bauteildaten, Messergebnisse, Messorganisation z.B. bei teilweise erforderlichen Wiederholmessungen etc.

Es soll ein neuartiges Messverfahren für die automatische Messung der Bauteile entwickelt und ein anschließendes Messprogramm implementiert werden. Es soll eine Methode gefunden werden, die Effizienz des Messungsprojektes zu realisieren. Das zu bewertende (zu entwickelnde) Verfahren soll für die Mitarbeiter einfach und leicht zu bedienen sein. Das bedeutet, dass eine Veränderung der Messstrategie zu vermeiden ist. Zusätzlich soll eine maximale Unterstützung bei der sicheren Eingabe von Messrandbedingungen zur Fehlervermeidung beitragen. Dies ist vor dem Hintergrund des automatischen Abspeicherns der Messergebnisse in einer Messdatenbank ein wichtiger Aspekt.

Das vorgeschlagene Verfahren ist im Vergleich zu bestehenden Methoden:

- effektiver, da die Messungsaufgaben für verschiedene Bauteilen ohne Pause kontinuierlich gemessen werden ;
- schneller, da keine Trainingszeiten benötigt werden;
- weniger aufwändig, da alle wichtigen Einstellungen automatisiert werden und kein zusätzliches Personal für die Bedienung gebraucht wird;
- flexibler, da das Verfahren erlaubt, unterschiedliche Bauteilen beliebig zusammen zu messen;
- leichter zu modifizieren bzw. zu erweitern, da alle Messprogramme einheitlich "verpackt" werden, wodurch einfach neu "Objekt" erzeugt werden können.

Die Software soll auf QUINDOS 7 implementiert werden.

1.3 Aufbau der Arbeit

Ein umfassender Einblick in die Themenkreise der Messtechnik und des theoretischen Messungsprozesses wird in Kapitel 2 vorgestellt. Die theoretischen Grundlagen sind für das Verständnis aller aktuellen Messstrategie notwendig, um die Messprogramme später zu umstrukturieren zu können.

Das Kapitel 3 stellt die aktuellen Messprozesse bei Daimler AG und dazu neue entwickelte Messprozesse vor. Die Architektur des Messprozesses sowie ihre einzelnen Komponenten werden dargestellt. Die modulare Bauweise des Messprozesses erlaubt eine strukturierte, hierarchische Anordnung der Systemkomponenten. Die Aufgaben jeder Komponente des Messprozesses sowie die verbundenen theoretischen Aspekte und Randbedingungen werden beschrieben. Um einen effizienten Programmablauf zu gewährleisten, wird besondere Berücksichtigung für die Randbedingung vorgeschlagen.

Kapitel 4 beinhaltet eine Auswahl an Versuchsbeispielen, die als experimenteller Beweis für die Leistungsfähigkeit des Verfahrens zur Messung der Multibauteile dienen. Eine Bewertung

des Verfahrens zeigt, wie der praktische Aufwand von neuem Messprogramm ist. Die Stillstandzeit der Maschine mit neuem Messprogramm im Vergleich zum alten Stillstandzeit der Maschine mit aktuellem Messprogramm wird es tatsächlich gesenkt.

Mit einer Zusammenfassung der vorliegenden Arbeit und einem kurzen Ausblick auf zukünftige Entwicklungen im Kapitel 6 schließt der Hauptteil.

Ergänzend folgt ein ausführlicher Anhang. Dieser umfasst QUINDOS-code, XML-code von verschiedenen Programmteilen. Die verwendeten Abkürzungen werden aufgelistet. Ein detailliertes Literaturverzeichnis schließt die Arbeit ab.

2 Theoretische Grundlagen

2.1 Cax-Technologien

Die zusammenfassende Bezeichnung aller Systeme der Rechnerunterstützung in einem Unternehmen lautet CAX-Systeme. Dabei steht „CA“ für "Computer-Aided", also "rechnerunterstützt" und "x" als Platzhalter für eine Vielzahl von Akronymen, die bestimmte Einsatzbereiche näher spezifizieren [3].

2.1.1 Abkürzung aus der CAX-Welt

Die Abkürzung CAX steht für computer-aided x, vereinzelt auch computer-assisted x, dt. meist *computerunterstützt*, *rechnergestützt* und ähnlich - wobei das x ein Platzhalter für verschiedene Buchstaben darstellt, die als Abkürzung für einzelne Ablaufabschnitte und Technologien der Produktion stehen. CAX umfasst die fertigungsbezogene angewandte Informatik in der Produktionswirtschaft und das Product-Lifecycle-Management [4].

- CAD: von engl. computer-aided design, zu Deutsch rechnerunterstütztes Konstruieren. Unter CAD versteht man das Konstruieren eines Produkts mittels EDV. Ursprünglich wurde mit CAD die Verwendung eines Computers als Hilfsmittel beim technischen Zeichnen bezeichnet. Heute sind professionelle CAD-Anwendungen komplexe Expertensysteme für den Entwurf und die Konstruktion technischer Lösungen [4].

Inzwischen ist in fast allen CAD-Anwendungen die dritte Dimension (3D) hinzugekommen. Damit bezeichnet CAD auch die Bildung eines virtuellen Modells dreidimensionaler Objekte mit Hilfe eines Computers. Von diesem können die üblichen technischen Zeichnungen abgeleitet und ausgegeben werden. Ein besonderer Vorteil ist, vom bereits virtuell bestehenden dreidimensionalen Objekt eine beliebige räumliche Abbildung zu erzeugen. Durch die erfassten Materialeigenschaften lassen sich darüber hinaus mit CAD Computer-Modelle unterschiedlicher Natur zum Finden einer geeigneten Lösung nutzen, Fertigungsinformationen (z. B. Rapid Prototyping) ableiten und Produktdokumentationen erstellen.

- CAE: von engl. computer-aided engineering, zu Deutsch rechnerunterstützte Entwicklung. Es umfasst alle Varianten der Rechner-Unterstützung von Arbeitsprozessen in der Technik.
- CAP auch CAPP: von engl. computer-aided process planning, zu Deutsch computergestützte Arbeitsplanung. Es baut auf konventionell oder mit CAD erstellten Konstruktionsdaten auf, um Daten für die Teilefertigungs- und Montageanweisungen zu erzeugen. CAP ist als Bestandteil der computerintegrierten Produktion (Computer-Integrated Manufacturing, CIM) in vielen Systemen zum Enterprise Resource Planning (ERP) implementiert.

- CAM: von engl. computer-aided manufacturing, zu Deutsch rechnerunterstützte Fertigung. Es bezeichnet die Verwendung einer von der CNC-Maschine unabhängigen Software zur Erstellung des NC-Codes. Im Unterschied zur Erstellung des NC-Codes in der Werkstatt (WOP), wird mit dem CAM-System das NC-Programm bereits in der AV erstellt. CAM ist ein wesentlicher Bestandteil der computerintegrierten Produktion CIM (Computer-Integrated Manufacturing).
- CAT: von engl. computer aided testing, zu Deutsch rechnerunterstütztes Testen. Der Erstellte Prototyp wird nun hinsichtlich der Produkthanforderung analysiert. Dabei unterscheidet man zwischen der zerstörungsfreien und der zerstörenden Analyse [6].
- CAQ: von engl. computer-aided quality assurance, zu Deutsch rechnerunterstützte Qualitätssicherung. CAQ-Systeme analysieren, dokumentieren und archivieren qualitätsrelevante Daten zu Fertigungsprozessen. CAQ umfasst computergestützte Maßnahmen zur Planung und Durchführung der Qualitätssicherung.

Die Analyse, Dokumentation und Archivierung qualitätsrelevanter Daten ist für Unternehmen zur Minimierung der Risiken nach dem Produkthaftungsgesetz von sehr hoher Bedeutung. Dazu werden Daten aus messenden, attributiven und visuellen Prüfungen vom CAQ-System mit statistischen Methoden ausgewertet. Die CAQ-Systeme zeigen auch die Prozessfähigkeit von Produktionsprozessen an, d.h. wie stabil und wie gut reproduzierbar Produktionsprozesse sind. Die Verknüpfung solcher qualitätsrelevanter Daten mit der Reklamationsbearbeitung kann zu einer deutlichen Kostenreduzierung führen [4].

Abbildung 2.1 zeigt die typischerweise zu durchlaufenden Schritte bei der Produktentwicklung. In der Diplomarbeit wird CAD und CAT Phasen betrachtet.

Bei CAD werden zunächst alle Bauteile in Simulation 3D Modelle, die später durch zugehörige Koordinatensysteme automatisch ihre Positionen für Messmaschine erkennen, erstellt. Bei CAT werden die Ist-Maß und Soll-Maß der Bauteile verglichen. Es zeigt die Ergebnisse der Messung. Die Analyse des Prototyps kann einerseits bewirken, dass zur Erfüllung der festgelegten Produkthanforderungen eine Konstruktionsänderung nötig ist. Andererseits können diese Analyseergebnisse dazu dienen, bei der auf die Konstruktionsänderung hin erneut notwendigen Konstruktionsanalyse die Kritischen Stellen genauer zu berechnen [5].

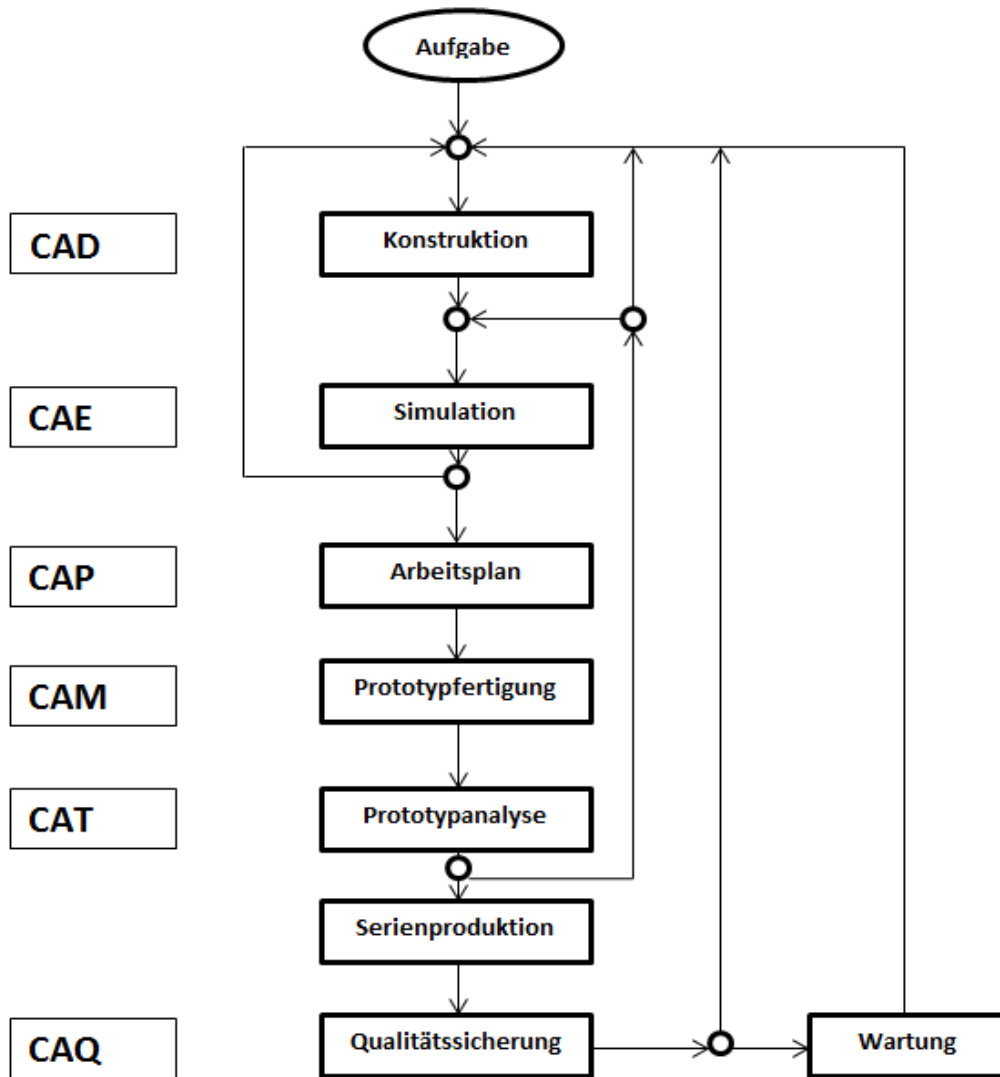


Abb. 2.1 Produktentwicklungsphasen und Konstruktionsänderung [6]

2.1.2 Modellarten

Eine wesentliche Motivation für die dreidimensionale Produktmodellierung (3D Modellierung) stellt die gleichzeitige und durchgängige Verwendung der Modelle in allen Arbeitsschritten des Produktentstehungsprozesses dar [3]. 3D-Modelle bieten im Vergleich zu 2D-Modellen nicht nur in geometrischer Hinsicht eine wesentlich eindeutiger und vollständiger Objektbeschreibung.

Die Repräsentationsformen für 3D-Modelle lassen sich in Hauptgruppen gemäß Abb. 2.2 einteilen, die im Folgenden genauer behandelt werden. Die wichtigsten Vertreter im Bereich CAD sind in Abb. 2.3 dargestellt.

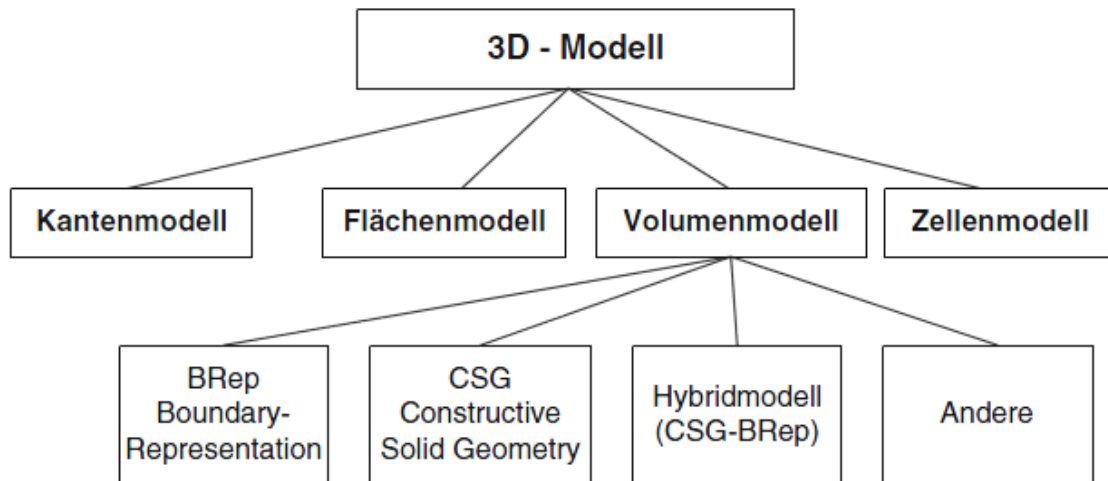


Abb. 2.2 Repräsentationsformen für 3D-Modelle [3]

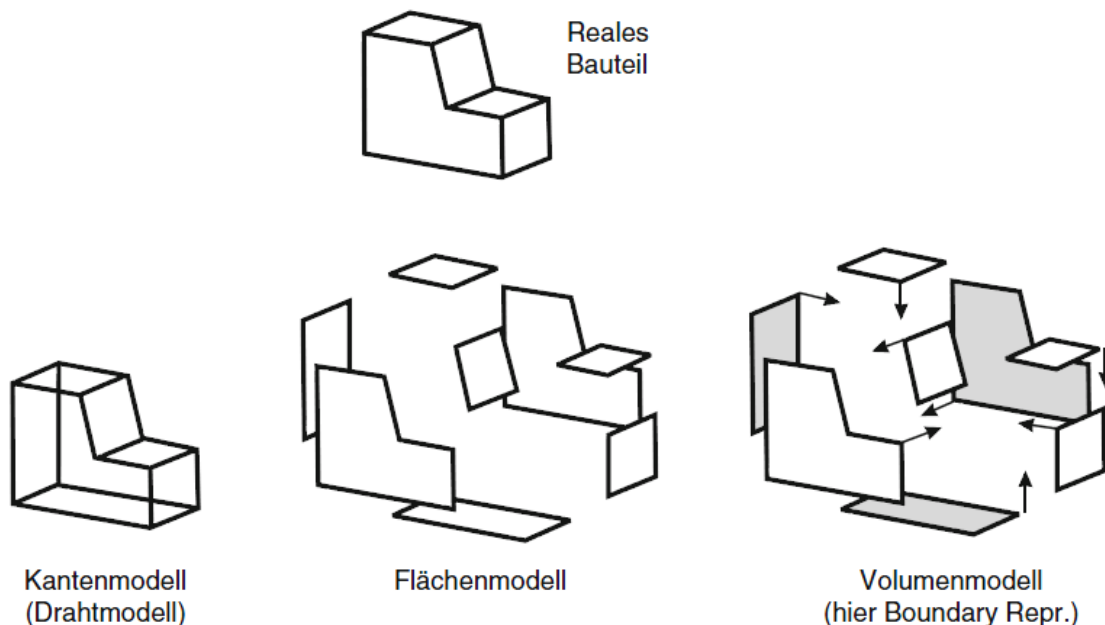


Abb. 2.3 Kanten- Flächen- und Volumenmodell [3]

- **Kantenmodelle:** Ein Objekt wird durch seine Kanten repräsentiert, die als Linienelemente (Strecken, Kreis-/Ellipsenbögen, Freiformkurven) gespeichert werden. Kantenmodelle sind in ihrer bildlichen Darstellung nicht eindeutig und in ihrer geometrischen Beschreibung nicht vollständig. Sie verfügen über keine Volumen- oder Flächeninformationen, weshalb auch keinerlei Volumen oder Flächenoperationen (Sichtbarkeit, Schnitte durch Körper, Durchdringungen, Erzeugung von Schnittkanten, Querschnitts- oder Volumenberechnungen, Kollisionsüberprüfungen, Rundungen, Fasen, schattierte Darstellungen usw.) möglich sind. Die Orientierung im Modell wird durch die "nicht-flächige" Darstellung erschwert. Im Gegensatz zu den beiden anderen Modellklassen erschweren Kantenmodelle bei der Bildschirmdarstellung die eindeutige Identifizierung von "vorne" und "hinten" sowie von "innen" und "außen".

Kantenmodelle werden nur noch dort eingesetzt, wo Kanten als Randkurven für Flächenmodelle, für definierende Skizzen („Sketches“) bei der Volumenmodellierung (insbesondere bei rotationssymmetrischen oder plattenförmigen Bauteilen) oder als Führungslinien notwendig sind [3].

- Flächenmodelle: Regel- und Freiformflächen werden durch Rotation und Extrusion von 2D-Konturen oder durch Zusammensetzen von 3D-Randkurven erstellt. Aufgrund der Randkurven- bzw. Isolinien-Darstellung verschlechtert sich - ähnlich wie bei der Kantenmodellierung - bei umfangreichen flächigen Gebilden die Orientierung im Modell. Auch wenn die Information über "vorn" und "hinten" gegeben ist, fehlt den Flächenmodellen, wie den Kantenmodellen, eine Information zu "innen" und "außen" [3].
- Volumenmodelle: Volumenmodelle bestehen aus topologisch geschlossenen Flächenverbänden. Sie erfassen nicht nur die "Hülle" von Objekten, sondern besitzen darüber hinaus auch Volumen und sofern eine Dichte zugeordnet wird - Materialinformationen. Sie können damit neben den geometrischen auch physikalische Informationen eines Produktes erfassen. In Bezug auf geometrische Toleranzen gibt es aber durchaus noch Defizite, da in allen Modellen in der Regel die ideale Sollgeometrie gespeichert wird. Volumenmodelle können für die Mehrzahl der Produkte im Maschinenbau eingesetzt werden. Produkte, die aus Grundelementen aufgebaut sind oder fertigungsorientiert konstruiert werden, sind mit der Volumenmodellierung effizient zu erzeugen [3].

2.1.3 Volumenmodellierung

Abbildung 2.2 zeigt drei Modellierungsverfahren des Volumenmodells. Es ist CSG-Modelle, B-Rep-Modelle und Hybrid-Modelle.

- CSG-Modell: entstehen durch mengentheoretische Verknüpfungen (Vereinigung, Subtraktion/Differenz, Durchschnitt/Verschneidung) von Grundvolumina (Primitiva). Diese sind entweder im 3D-System vorgegeben oder werden vom Anwender erzeugt. Die einzelnen Verknüpfungen werden in ihrer logischen und historischen Reihenfolge, dem sogenannten "CSG-Baum", gespeichert. Die Blätter des Baumes verweisen auf Primitiva. Abbildung 2.4 verdeutlicht, wie aus zwei Primitiva durch Vereinigung ein neues Volumen erzeugt wird. In Abb. 2.5 sind einige Volumenoperationen (Vereinigung, Differenz, Durchschnitt) dargestellt, die durch Boole'sche Operationen erzeugt werden. Abbildung 2.6 zeigt, wie durch unterschiedlichen Modellaufbau für ein und denselben Körper zwar verschiedene CSG-Bäume entstehen, sich jedoch in allen Fällen - entsprechend dem erzeugten Körper - äquivalente Boole'sche Ausdrücke ergeben [3,6].

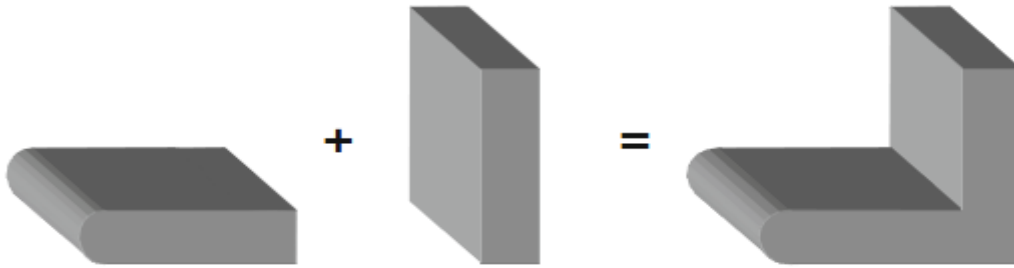


Abb. 2.4 Erzeugen eines Volumens durch Vereinigung [3]

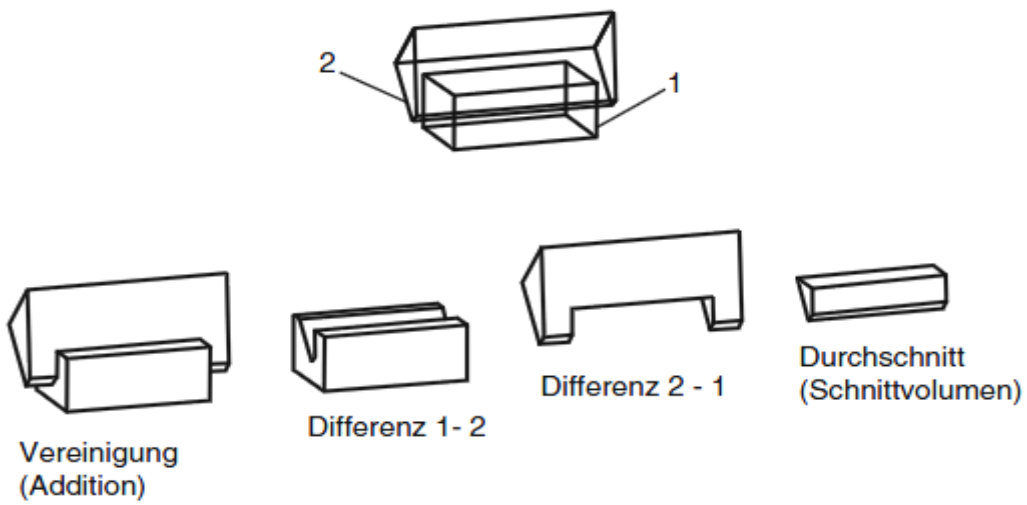


Abb. 2.5 Boole'sche Operationen für zwei Volumina [3]

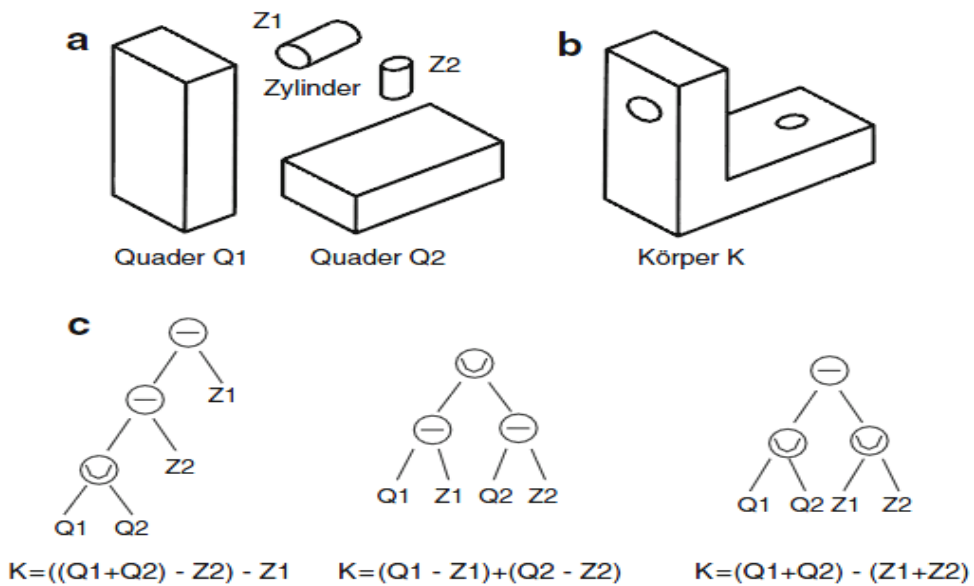


Abb. 2.6 Darstellung eines Objektes durch drei verschiedene CSG-Bäume [3]

- B-Rep-Modelle: beschreiben Volumina rechnerintern durch die sie begrenzenden Flächen und diese wiederum durch die Berandungskanten, wie Abbildung 2.7 zeigt. Neben analytischen Flächen und Kurven (Ebenen, Zylinder bzw. Strecken, Kreise/Ellipsen) werden auch Freiformkurven und -flächen verwendet. Zur eindeutigen Festlegung, auf welcher Seite der Fläche sich das umschlossene Volumen befindet, wird die Richtung des Normalenvektors verwendet, der in Richtung des Materials zeigt. Damit kann beispielsweise das Volumen "Zylinder" entweder eine Welle darstellen (Normalenvektor nach innen) oder eine Bohrung (Normalenvektor nach außen). Die Geschlossenheit der ein Volumen einhüllenden Flächen wird mit der Vektorsumme aller Normalenvektoren geprüft. Diese Summe ist nur dann Null, wenn die Oberfläche vollständig geschlossen ist [3,6].

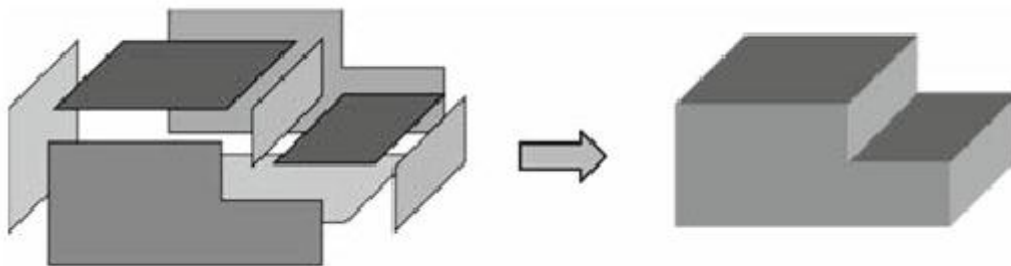


Abb. 2.7 B-Rep-Modell einer L-förmigen Platte [3]

- Hybrid-Modelle: In einem Hybrid-Modell finden sich sowohl CSG-Bäume als auch B-Rep-Darstellungen. Dabei werden B-Rep-Volumina als "komplexe Primitiva" in die CSG-Bäume integriert in Abbildung 2.8. Die strikte Trennung beider Modellierer Typen ist bei den meisten der heute eingesetzten 3D-CAD-Systeme durch die Benutzungsoberfläche nach außen nicht mehr eindeutig erkennbar. Hier besteht die Möglichkeit, Zusammenhänge aus unterschiedlichen Sichten darzustellen: einerseits eine Konstruktionshistorie auf der Basis der CSG-Bäume, andererseits die Relationen aller Objekte untereinander (geometrische, topologische, Constraints usw.) und die Ergebnisse daraus durch die B-Rep-Darstellungen [3,6].

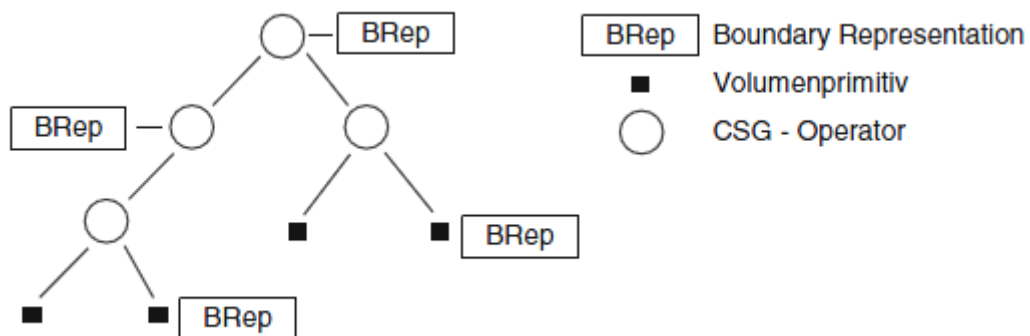


Abb. 2.8 Datenstruktur eines Hybrid-Modells [3]

2.1.4 Struktur der Bauteilen

Jeder Bauteil durch ihre Struktur kann durch zwei Verfahren verteilt werden und 3D Modell erstellt werden.

- Einstufige Struktur: Die Modellierung ist geradlinig aufgebaut, d.h. die Modellierungsschritte sind nacheinander aufgeführt. Der Aufbau des Bauteils entspricht dabei der Reihenfolge der Modellierung.
- Mehrstufige Struktur: Bei der mehrstufigen Modellierung handelt es sich um eine beliebig häufig verzweigte Struktur, d.h. die Modellierungsschritte sind unterschiedlichen Stufen zugeordnet. Diese Form entsteht dadurch, dass bei der Modellierung zuerst die Grobgestalt modelliert wird (Ergebnis: Einstufige Struktur), die danach in unterschiedlichen Teilbereichen detailliert wird. Die detaillierenden Arbeitsschritte bilden dabei tiefere Stufen der Bauteile. Für den Anwender steht die gesamte Struktur sichtbar zur Verfügung. Die Reihenfolge der Modellierungsschritte innerhalb einer Chronologie ist in der Regel nachvollziehbar. (siehe Abb.2.9)

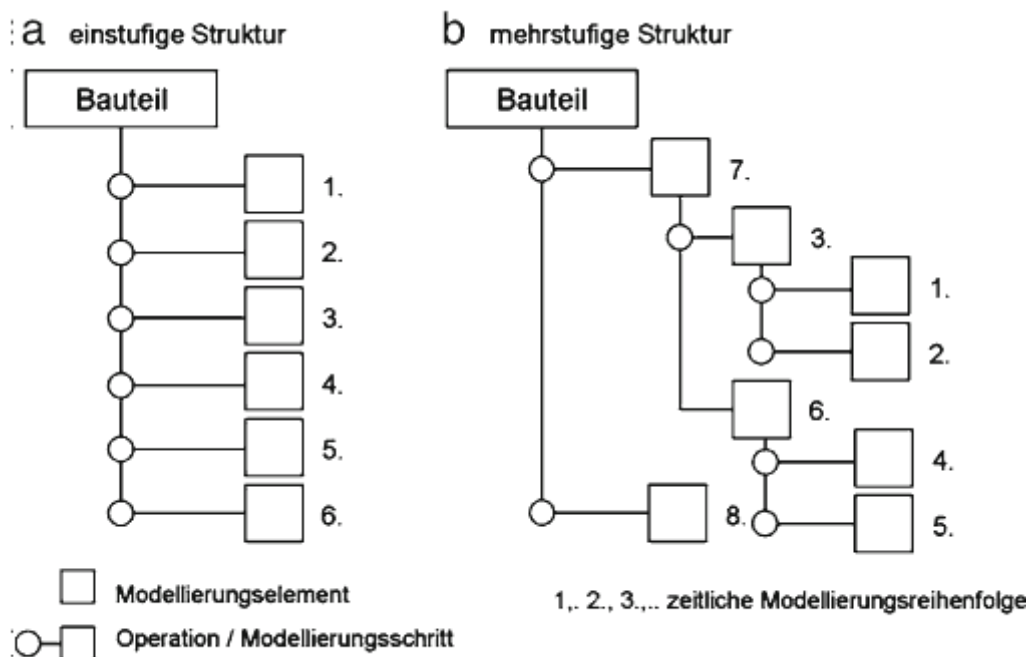


Abb. 2.9 einstufige Struktur und mehrstufige Struktur Modellierung [3]

2.1.5 Koordinatensystem des zumessenden Bauteils

Jeder Bauteil besitzt ein eigenes Koordinatensystem, das sich wiederum auf ein Gesamtkoordinatensystem bezieht. Das bedeutet, dass sich letztendlich jedes Teil (wenn auch häufig indirekt, da über das jeweilige lokale Koordinatensystem) auf das Gesamtkoordinatensystem bezieht. Oder der Bauteil enthält Referenzelemente, die mit entsprechenden Elementen in anderen Bauräumen verknüpft sind, wie Abbildung 2.10 zeigt.

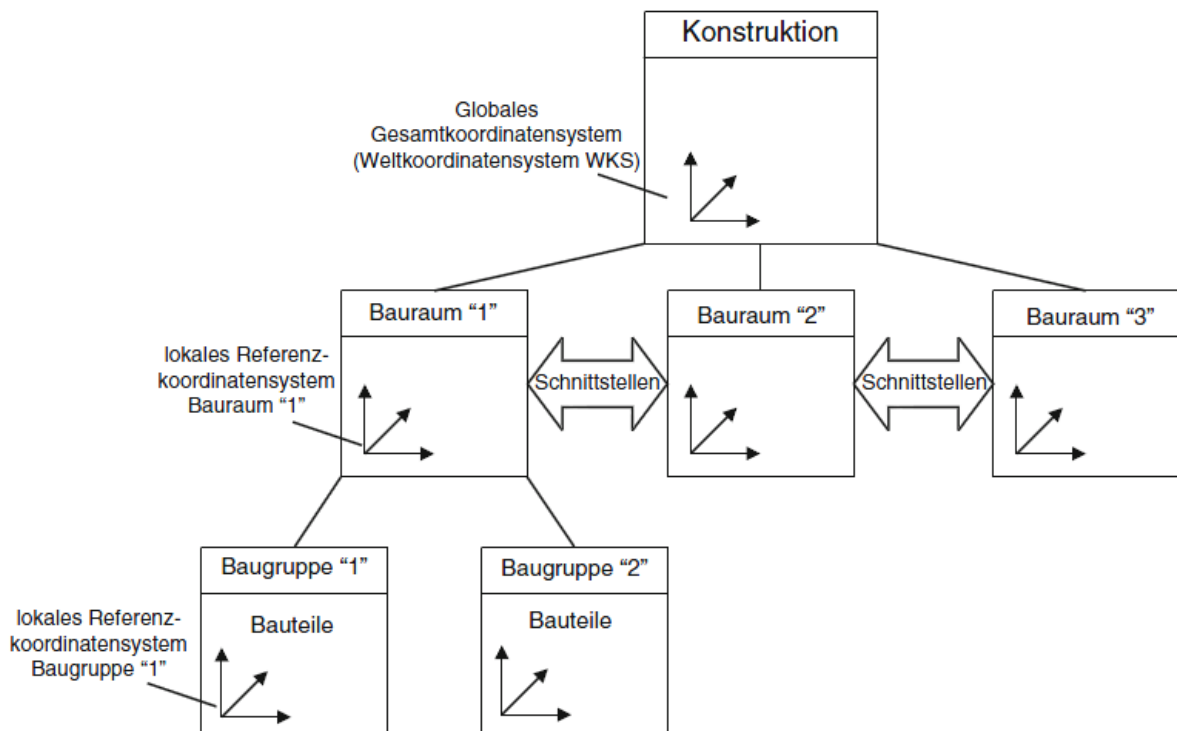


Abb. 2.10 Unterteilung der Bauteile [3]

2.2 Grundlagen und Stand der Messtechnik

Die Messtechnik befasst sich mit der Bestimmung (Messung) physikalischer Größen sowie mit der Entwicklung und Implementierung von Messsystemen und Messmethoden [7].

Messung bedeutet Vergleichen, genauere Definition, ist das Ausführen von geplanten Tätigkeiten zu einer quantitativen Aussage über eine Messgröße durch Vergleich mit einer Einheit. Eine Messgröße ist jene physikalische Größe, der die Messung gilt. Der Wert der Messgröße wird durch das Produkt aus Zahlenwert und Einheit ausgedrückt. Ein Messwert ist der Wert einer Messgröße, der von einem Messgerät oder einer Messeinrichtung geliefert wird [7,8].

Ein Koordinatenmessgerät (KMG) ist ein Messgerät zur Bestimmung der kartesischen Koordinaten (X, Y, Z) eines Punktes an einem Objekt im Raum. Aus der Verbindung von mehreren Messungen weiterer Punkte mit einer Datenverarbeitung ergeben sich dann verschiedene geometrische Größen und Eigenschaften eines Messobjekts [9]. Bei Daimler AG Abteilung TP/PPH werden die Koordinatenmessgeräte vom Unternehmen Hexagon Metrology eingesetzt. Das folgende Kapitel soll einen Überblick über das Koordinatenmessgerät und dessen Funktionsweise liefern.

2.2.1 Aufbau der 3D Koordinatenmessgeräte

In Abhängigkeit von dem jeweiligen zu messenden Werkstück ergeben sich unterschiedliche Anforderungen an das Koordinatenmessgerät. Im Wesentlichen haben sich vier verschiedene Bauartausführungen durchgesetzt. Darüber hinaus sind entsprechend den jeweiligen Anforderungen auch Mischformen möglich.

- Auslegerbauart (mit Tisch, siehe Abbildung 2.11 A)
Bei einer derartigen Bauweise ist ein Maximum an Zugänglichkeit gewährleistet. Allerdings muss hier die Durchbiegung des Auslegers über dem Messbereich kompensiert werden. Beim Einsatz mehrerer Tastköpfe mit unterschiedlichem Gewicht gestaltet sich dies schwierig. Es ist ein relativ großes Messvolumen möglich.
- Ständerbauart (mit Tisch, siehe Abbildung 2.11 B)
Diese Bauart lässt in aller Regel nur relativ kleine Messvolumen zu. Die Messunsicherheit ist im Vergleich zu den anderen Bauarten gering.
- Portalbauart (siehe Abbildung 2.11 C)
Die Portalbauart ist die heutzutage am häufigsten anzutreffende Bauart von Messgeräten. Da sie über eine massiv ausgeführte Werkstückaufnahme verfügt, lassen sich mittlere bis große Teile problemlos messen. Im Rahmen der Portalbauart gibt es zwei verschiedene Ausführungsformen. Im ersten Fall fährt das Portal und der werkstücktragende Tisch steht. Im zweiten Fall bewegt sich der Tisch mit dem gesamten Werkstück und das Portal steht.

- Brückenbauart (siehe Abbildung 2.11 D)
Die Brückenbauart ist die größte aller Bauformen. Sie wird für Großteile sowie im Karosserie- und Formenbereich angewendet. Dahingegen ist sie nicht für Präzisionsmessungen geeignet. [9]

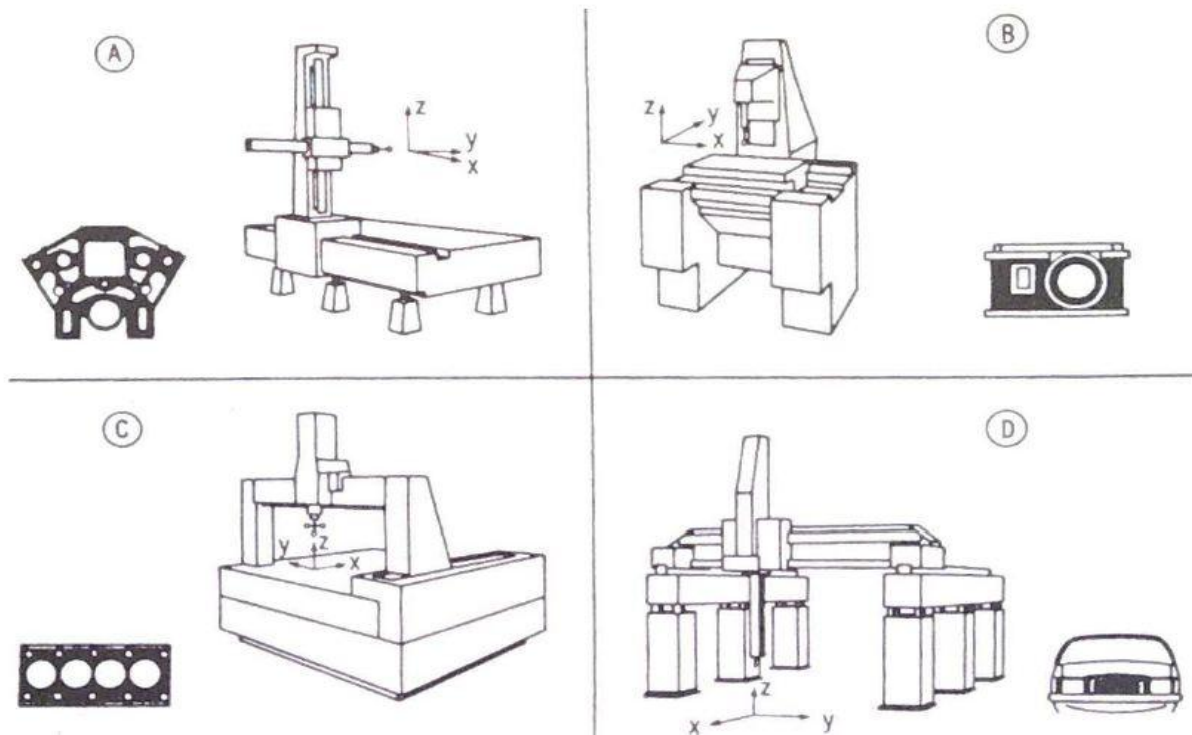


Abb. 2.11 Bauarten von Koordinatenmessgeräten [9]

Jedes Messgerät, das auf der Basis eines geräte-technisch festgelegten Koordinatensystems eine Werkstückoberfläche punktweise optisch oder taktil erfasst und die Koordinatenwerte mit einem Rechner weiterverarbeitet, kann als Koordinatenmessgerät bezeichnet werden. Alle KMG beinhaltet die Funktionsgruppen: Gerätebasis, Werkstückaufnahme, Lagerung, Antrieb, Mess- und Tastsysteme, Computer und Rechnerperipherie, Tastsystemwechsler usw.

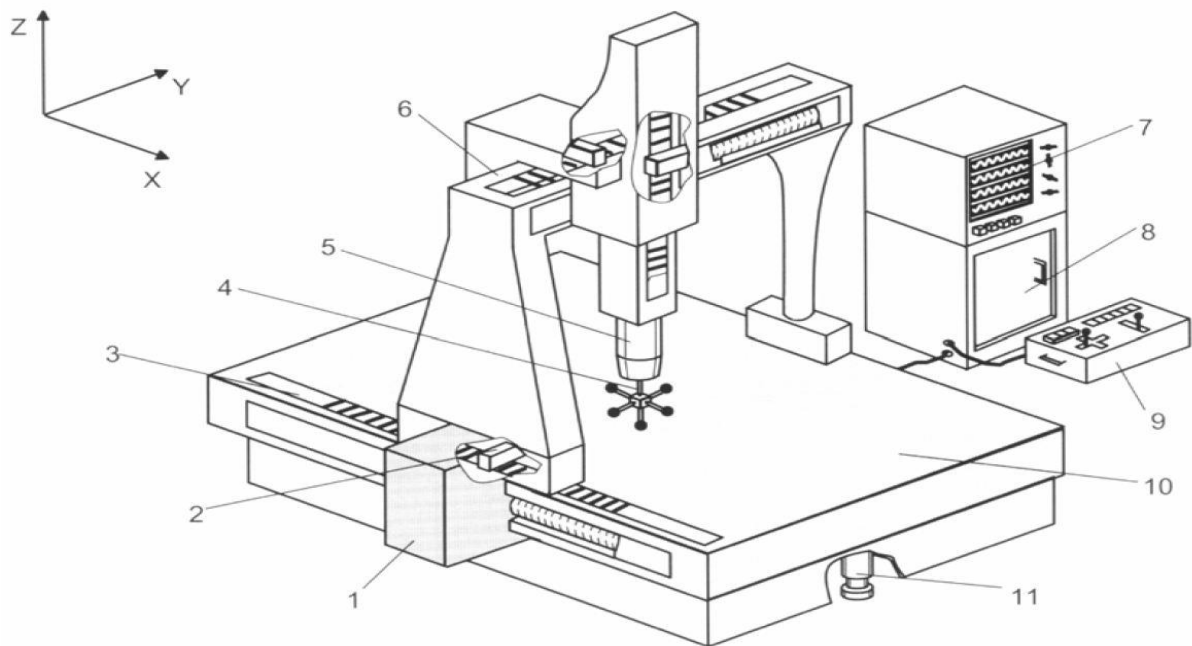


Abb. 2.12 Aufbau eines kartesischen Koordinatenmessgeräts (1 Antrieb für die X – Achse, 2 Ablesesystem für X – Achse, 3 Maßwerkörperung für X – Achse, 4 Taster, 5 3D Tastkopf, 6 Lagerung für Y – Achse, 7 Anzeige, 8 Steuer- und Anpasselektronik, 9 Steuerpult, 10 Werkstückaufnahme, 11. Geräteeinheit) [10]

Abbildung 2.12 zeigt den prinzipiellen Aufbau eines konventionellen kartesischen Koordinatenmessgeräts (Portalbauart). Vorteil von solcher Art ist die hohe Messgenauigkeit und der durch ihre einfache Konstruktion dagegen sehr preisgünstig.

2.2.2 Tastkopfsystem

Das Tastkopfsystem ist ein Element zwischen Pinole und Taster (siehe Abbildung 2.13).

Das Tastkopfsystem leistet:

- ein erlaubt Auslenken des Tasters in allen Richtungen beim Antasten;
- eine in allen Achsen immer wiederkehrende gleichbleibende Nullstellung gewährleisten;
- beim Antasten aus jeder Richtung ein elektrisches Signal erzeugen und sich dabei auf einen unveränderbaren Nullpunkt beziehen;
- mehrere unterschiedliche (lange, kurze, leichte, schwere) Taststifte aufnehmen;
- mit einer definierten Messkraft antasten.

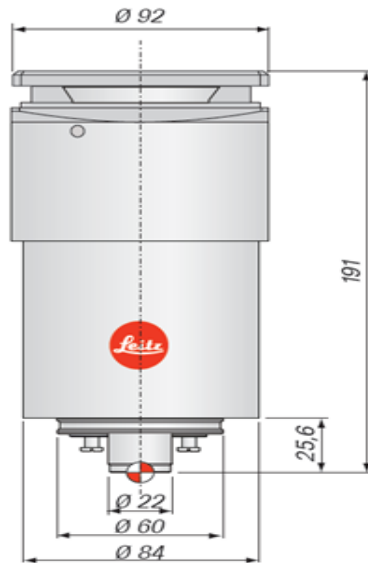


Abb. 2.13 Tasterkopfsystem von Leitz Typ LSP-X5 [11]

Es bietet ein effektives Antikollisionssystem. Der Tastkopf wird schnell und einfach per Schwabenschwanzführung in die Messmaschine eingesetzt. Er bietet effektive Einzelpunktmessung für alle Standardmessaufgaben sowie kontinuierliches High-Speed-Scanning für die Form- und Profilmessung von komplexer Geometrie. Optional kann ein Werkstücktemperatur-Sensor (WTS) in den Tastkopf integriert werden. Auf diese Weise kann die Werkstücktemperatur automatisch im Messprogramm erfasst werden.

2.2.3 Taststift und Taster

Taststift besteht normalerweise aus drei Bestandteilen: Tastelement, Schaft, Adapter (Adapter ermöglichen die Kombination mit Fremdprodukten), wie in der Abbildung 2.14 zeigt.



Abb. 2.14 Taster von Zeiss [10]

Das Tastelement kann, wie in der Abbildung 2.15 zeigt, eine Kugel, eine Scheibe, ein Zylinder oder eine Abwandlung davon sein, wie eine Halbkugel, ein Zylinder mit Kugelfende, usw.

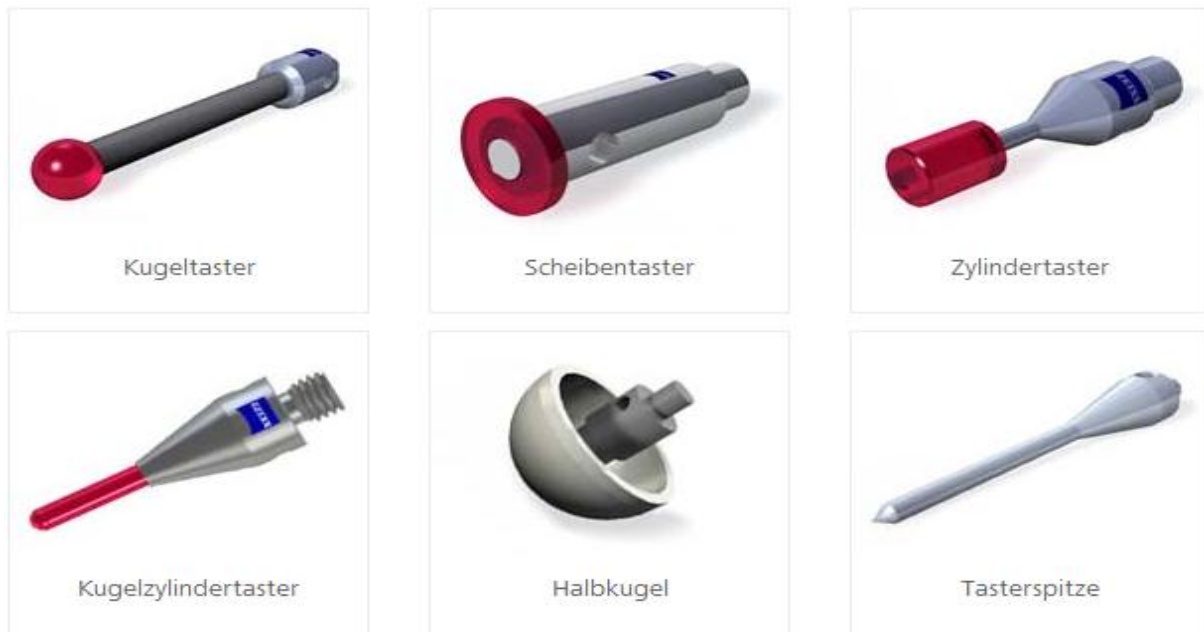


Abb. 2.15 verschiedene Tastspitzen [11]

Für viele Messungen ist eine Standard-Tastkugel ausreichend, je nach Messaufgabe können aber Tastspitzen in vielfältigen Formen verwendet werden. Die Kugel ist von allen geometrischen Formen am besten zum Antasten geeignet, weil ihr Mittelpunkt zu allen Punkten der Kugeloberfläche den gleichen Abstand hat und ihr Radius im Berührungspunkt senkrecht auf der Fläche steht.

An diesem Tastkopfsystem kann nach Bedarf entweder ein einzelner Taststift oder ein so genannter Tasterbaum befestigt werden.

2.2.4 Tasteraufnahme

Die Tasteraufnahme beinhaltet die Hälfte der hochgenauen, reproduzierbaren kinematischen Magnetkupplung, die es erlaubt, den Taster reproduzierbar anzudocken. In der Tasteraufnahme ist ein magnetischer Näherungsschalter integriert, um unerwünschte Schaltsignale während des automatischen Modulwechsels mit dem Wechselmagazin zu unterdrücken. Untere Abbildung 2.16 zeigt eine normale Tasteraufnahme.

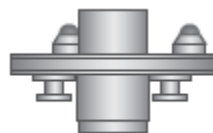


Abb. 2.16 Tasteraufnahme von Leitz (M00-813-101-000) [11]

2.2.5 Tasterwechselbank und Tasterablage

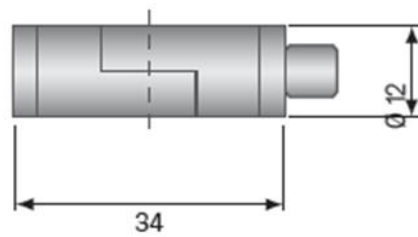
Eine automatische Tasterwechselbank ermöglicht den Tasterwechsel innerhalb eines Messprogramms, ohne dass eine erneute Kalibrierung des Tasters erforderlich ist. Die magnetische oder pneumatische Fixierung von Taststiften am Kopf ermöglicht schnelle, zuverlässige Wechsel. Die Tasterwechselbank ist mit mindestens 3 Tastablagen (Standard) und mehrere Tasterablagen je nach Taster Volumen erhältlich. Die Taster sind nach ihrer eine Magazinposition in der Taster Wechselbank nummeriert. Abbildung 2.17 zeigt Standardtasterwechselbank mit drei Tasterablagen, aber es können auch weiter bis zu 11 Tasterablagen angebracht werden.



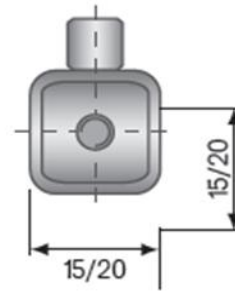
Abb. 2.17 Tasterwechselbank(M00-114-119-000) und Tasterablage(M00-153-286-000) von Leitz [11]

2.2.6 Würfel und Gelenkstücke

Mit Würfelstern sind die Einzeltaster wechselbar, und die alle 5 (+X, +Y, -X, -Y, -Z) Antastrichtungen realisierbar. Mit dem Gelenkstücke ist schräges Antasten realisierbar. Ein Tasterbaum ist eine Tasteraufnahme in Verbindung mit Würfel oder Gelenk mit den verschiedenen Tastern in unterschiedlichen Antastrichtungen. Abbildung 2.18 links zeigt ein Gelenk, und rechts einen Würfel.



Gelenk 0°-90°, Ø 12 mm
060-813.051-000
Gewicht: 21 g



Würfel, Ø 15 mm
060-694.016-000
Gewicht: 11 g

Abb. 2.18 Gelenk und Würfel von Leitz [11]

2.2.7 Konfigurationsdiagramm des 3D-Tastsystems

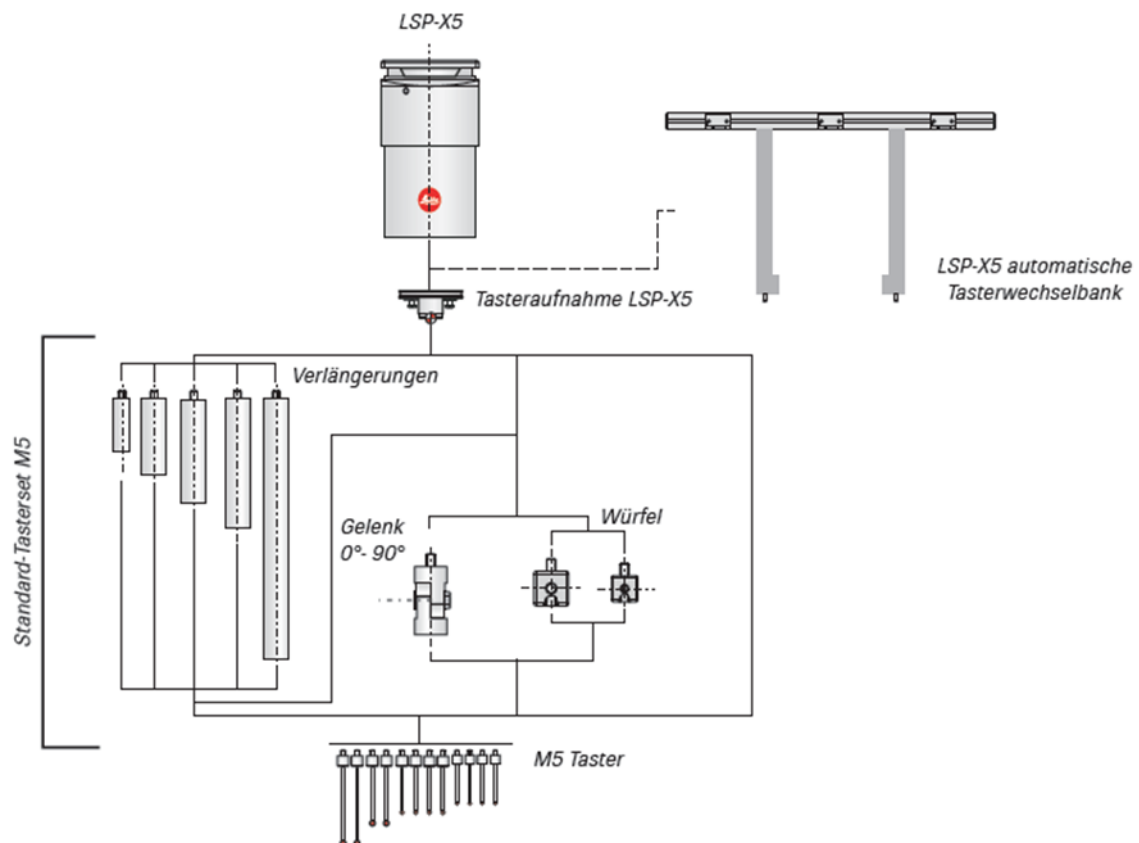


Abb. 2.19 Konfigurationsdiagramm für LSP-X5 Messsystem [11]

Das Messsystem ist hoch flexibel alle Teile zur Verbindung, um einen komplexen Messbauteil zu messen. Mit dem 3D-Tastsystem können hochgenaue und schnelle Einzelpunktantastungen wie die Abbildung 2.19 zeigt. High-Speed-Scanning für form- und Profimessungen sowie Selbstzentrierungen in allen Achsen vorgenommen werden.

2.3 Koordinatenmessgeräte bei Abteilung TP/PPH und ihre Randbedingungen

Die von Abteilung TP/PPH benutzten Koordinatenmessgeräte (KMG) sind von Leitz Serie "PMC" und "CYG". Es sind hochgenaue Messgeräte in monolithischer Portalbauweise für große Werkstücke. Es ist darauf ausgelegt, bei maximaler Genauigkeit einen hohen Durchsatz zu erzielen. Es ist auch mit einem Messvolumen von bis zu 9,6 m³ erhältlich. Sie benötigen kein teures Fundament, ist daher bei Bedarf schnell umsetzbar und bietet serienmäßig aktive Schwingungsdämpfung. Mit dem 3D-Tastsystem Serie "LSP" können hochgenaue und schnelle Einzelpunktastern, High-Speed-Scanning für form- und Profilmessungen sowie Selbstzentrierungen in allen Achsen vorgenommen werden [11].

2.3.1 Alle verwendeten Tasterbäumen für entsprechenden Bauteilen

Die gesamte Anzahl der gleichzeitig auf einem Koordinatenmessgerät eingesetzten Tasterbäume ist begrenzt. Bei KMG Typ PMC321 stehen maximal 11 Tasterablagen für Tasterbäume (Tasteraufnahmen) zur Verfügung, KMG Typ CYG125 und KMG Typ PMC393 stehen maximal 13 Tasterablagen für Tasterbäume zur Verfügung. Abbildung 2.20 zeigt einen kompletten Tasterbaum mit Verlängerung, Tasteraufnahme und Gelenk.



Abb. 2.20 Tasterbaum (Tasteraufnahme, Verlängerung, Gelenk, Taster) und Tasterablage [12]

Die Tasterbaumausstattung einer KMG ist bauteilspezifisch. Das Verändern der Tasterbaumausstattung während einer Messung ist nicht möglich.



Abb. 2.21 Tasterbaumausstattung für Bauteil Kurbelgehäuse [12]

Wenn beispielweise zwei Bauteile nach einander automatisch gemessen werden sollen, müssen es alle Tasterbüumen einmalig in den Tasterablagen eingesetzt werden. Die Anzahl der Tasterbüumen darf nicht die Grenz der maximalen Anzahl der verschiedenen Messmaschinen überschreiten. Abbildung 2.21 zeigt aktuelle benutzte Tasterbaumausstattung für den Bauteil "Kurbelgehäuse". Es ist ein gutes Beispiel, daraus muss man überlegen, wie alle zu benutzenden Tasterbüumen für unterschiedliche Bauteile in ihren Tasterablagen gut verteilt werden. Alle Tastverlängerungen und Antastrichtungen müssen auch damit nachzudenken.

Durch die Analyse von allen Messprogrammen entsteht die untere Tabelle 2.1. Daraus kann man direkt feststellen, welche Bauteile zusammen mit anderen gemessen werden können, wenn dabei die maximale Anzahl der insgesamt verwendeten Tasterbüumen eingehalten wird.

Tab. 2.1 die Anzahl der verwendeten Tastbüumen von entsprechenden Bauteilen

Bauteile	Messprogramm	Anzahl des Tasterbaums
Zylinderlaufbuchsen	Buchsen.wdb	4
Kolben / Kolben zur Sondermessung	Kolben.wdb	4
	Kolben.wdb	8
Kurbelwelle	Kurbelwelle.wdb	3
Lagerzapfen	Lager_Zapfen_KMG.wdb	1
Lagerschale	LASCHA.wdb	2
Nockenwelle	Nockenwelle.wdb	2
Pleuelstange	Pleuel.wdb	1

Bauteile	Messprogramm	Anzahl des Tasterbaums
Zahnrad	Zahnrad.wdb	1
Zahnrad Sondermessung	Zahnrad_SM.wdb	2
Nockenwellerahmen	Nowerahmen_HDEP.wdb	5
	Nowerahmen_MDEG.wdb	3
Kurbelgehäuse	KGH_BR900_PAR.wdb	9
	KGH_SCHWERE_BR_PAR.wdb	9

2.3.2 Verwendete Taster je Bauteil

Jedes Bauteil benutzt verschiedene Taster, die auf Tasterbäume verteilt werden. Durch die Analyse der aktuellen Messprogramme zeigt es die Tasterdaten in der unteren Tabelle 2.2.

Tab. 2.2 Die verwendeten Taster des Bauteils mit selben Tasterbaum

Bauteil	Tasterbaumindex	Tastspitze	Durchmesse(mm)	Achslager
Zylinderlaufbuchsen	1	Kugel	2	+x
		Kugel	2	+y
		Kugel	2	-x
		Kugel	2	-y
	2	Kugel	1	+x
		Kugel	1	+y
		Kugel	1	-x
		Kugel	1	-y
	3	Scheibe	25,007	+x
		Scheibe	25,007	+y
		Scheibe	25,007	-x
		Scheibe	25,007	-y
4	Scheibe	25,476	-z	
Kurbelwelle	1	Kugel	6	+x
		Kugel	6	-x
		Kugel	3	-z
	2 (mit Verlängerung)	Kugel	6	+x
		Kugel	6	-x
		Kugel	3	-z
	3	Kugel	8	+x
		Kugel	8	-x
Lagerzapfen	1	Kugel	5	-z
Nockenwelle	1	Kugel	5	+y
		Kugel	5	-y
		Kugel	5	-z
	2	Kugel	3	+x
		Kugel	3	-x
Pleuelstange	1	Kugel	5	-z

Bauteil	Tasterbaumindex	Tastspitze	Durchmesse(mm)	Achslager
Kolben	1	Kugel	20	-z
	2	Kugel	5	+x
	3	Kugel	5	-x
	4	Kugel	2	+x
		Kugel	2	+y
		Kugel	2	-x
		Kugel	2	-y
Kolben Sondermessung	5	Kugel	1	+y-z
	6	Kugel	1	+x-z
	7	Kugel	1	-y-z
	8	Kugel	1	-x-z
Lagerschale	1	Kugel	3	-z
	2	Kugel	3	+y
Zahnrad	1	Kugel	3	-z
	2	Kugel	1,5	+x
		Kugel	1,5	+y
		Kugel	1,5	-x
		Kugel	1,5	-y
		Kugel	1,5	+x+y
		Kugel	1,5	+x-y
		Kugel	1,5	-x+y
Kugel	1,5	-x-y		
Zahnrad Sondermessung	1	Kugel	3	-z
	2	Kugel	8	+y
		Kugel	5	-y

2.3.3 Messfläche und Messvolumen von KMG

Jeder Messbereich des Messgeräts ist begrenzt, zum Beispiel Messmaschine Typ PMC ist ab 800 x 1000 x 600 bis 2400 x 1600 x 1000 (mm). Die Aufspannung für jeden Bauteil ist unterschiedlich. Bei aktuellem Messprogramm ist jedes Mal nur ein Bauteil gemessen, d.h., dass viele Messbereiche bei Messungsdurchlauf verschwendet werden. Wenn es mehrere Bauteile zusammen gemessen werden, müssen diese im Messbereich der Messmaschine günstig verteilt werden.



Abb. 2.22 Die Aufspannungen jedes einzelnen Bauteils [12]

Abbildung 2.22 zeigt oben verschiedene Aufspannverfahren, aber nur für ein einziges Bauteil. Für das neue entwickelte Verfahren muss es zuerst nachdenken, wie mehrere Bauteile zusammen auf der Messmaschine gleichzeitig aufgespannt werden, welche Bauteile wegen ihrer physikalischen Größe sich nicht zusammen messen lassen. (zum Beispiel: Nockenwellerahmen mit Kurbelwellegehäuse).

Die bisher verwendeten Messprogramme sind streng für die Bewegung des Tastkopfs programmiert. Beim Betrieb der Messmaschine ist darauf zu achten, dass bei allen Verfahrbewegungen ein ausreichender Sicherheitsabstand zum Werkstück und zu anderen Hindernissen im Messvolumen eingehalten wird. Bei Multimesprogramm ist immer darauf zu achten, dass sinnvolle Freifahrpunkte bzw. Zwischenpositionen angelegt werden, so dass keine Kollision erfolgen kann. Abbildung 2.23 zeigt das Volumen einer Messmaschine.

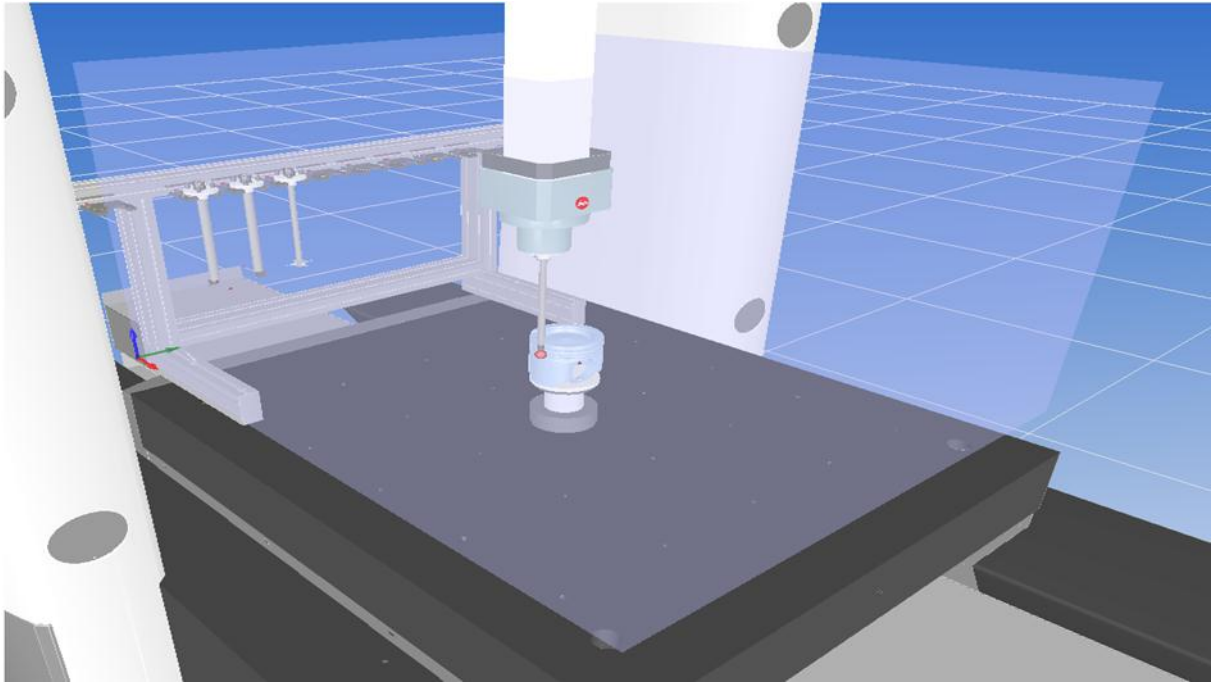


Abb. 2.23 Messungsvolumen von Simulator (Leitz)

2.4 Der theoretischer Messprozess

Eine Messung ist das Ausführen von geplanten Tätigkeiten zu einer quantitativen Aussage über eine Messgröße durch Vergleich mit einer Einheit.

Zum Messen gehören (Reihenfolge):

- Eindeutige Definition der Messaufgabe und der Messgröße: die Aufgabe, das Messobjekt und die physikalische Messgröße sind festzulegen.
- Festlegung der Maßeinheit für das Ergebnis: die Einheit und ihr Einheitenzeichen sind in der Regel gemäß dem SI festgelegt, wählbar sind Vorsätze für Zehnerpotenzen.
- Zusammenstellen der Randbedingungen: als Randbedingungen sind z. B. Eigenschaften des Messobjektes (Werkstoff, Oberflächenbeschaffenheit) und der Umgebung (Temperatur, Schwingungen) zu beachten.
- Wahl einer Messeinrichtung oder eines Messgerätes: ausgehend vom Messprinzip und der Messmethode wird ein Messverfahren entwickelt, das in einer Messeinrichtung verwirklicht wird. Vielfach steht bereits ein fertiges Messgerät für die Messaufgabe zur Verfügung.
- Kalibrieren von Messeinrichtung/Messgerät: dieses wird durch das Verfahren der Messmittelüberwachung gesichert. Dazu soll ein Messgerät in regelmäßigen Abständen kalibriert (eingemessen) werden. Dabei ermittelt man den Messwert (Ausgangsgröße) bei einem als richtig anzusehenden Wert der Messgröße (Eingangsgröße). Falls der Messwert nicht mit dem Wert der Messgröße innerhalb vorgegebener Fehlergrenzen übereinstimmt, ist das Gerät vorerst neu zu justieren.
- Festlegung des Messablaufs: zeitliche bzw. örtliche Abfolge der Messungen: z. B. Reihenfolge von Einzelmessungen, Wiederholungen, Messreihe unter geänderten Bedingungen; räumliche Verteilung der Messpunkte.
- Durchführen der Messung und Ermitteln des Messergebnisses: es können eine Messung oder auch mehrere unter denselben Bedingungen gewonnene Messungen derselben Größe (Vergleichs-/Wiederholmessungen) durchgeführt werden. Dann sind Mittelwert und Standardabweichung zu berechnen.
- Ferner können Messungen verschiedener Größen erforderlich sein, aus denen der Messwert der gesuchten Größe nach festgelegten mathematischen Beziehungen zu berechnen ist.
- Berücksichtigung der Auswirkungen von Einflussgrößen: Korrektur von systematischen Messabweichungen.
- Ermitteln des vollständigen Messergebnisses. [14]

2.5 Programmiersprache "QUINDOS 7"

"QUINDOS 7" ist eine objektorientierte Programmiersprache. QUINDOS ist das leistungsfähigste Analysewerkzeug für viele Arten von Koordinatenmessgeräten und verschiedene Anwendungen, speziell in der Antriebstechnik im Automobilbau, in der Luftfahrt und dem Maschinen- und Kraftwerksbau.

In QUINDOS Datenbanken sind alle aufgenommenen Daten zum Verarbeiten und Darstellen zugänglich. Vielfältige Funktionen verbessern die Effizienz von Messabläufen. QUINDOS 7 stellt eine Kombination aus einem Datenbanksystem und einem Anweisungsinterpreter dar, die zu jeder Zeit - interaktiv oder durch Programmbefehl - den direkten Zugriff auf alle Daten in den Datenbanken ermöglicht [15].

In QUINDOS 7 sind alle Informationen, die zur Durchführung einer Werkstückmessung notwendig sind, als Objekte in Datenbanken abgelegt.

Die folgenden Datenbanken werden beim ersten Start von QUINDOS 7 angelegt:

- Lokale Datenbank (LDB)
- Globale Datenbank (GDB)
- Umgebungs-Datenbank (EDB)
- Sprachunabhängige System-Datenbank (SDB)
- Werkstück-Datenbank (WDB)
- Benutzer-Datenbank (UDB)
- Kundenspezifische Systemdatenbank (ZDB)

Meistens werden die Datenbanken LDB, GDB, und EDB benutzt.

In der lokalen Datenbank (LDB) befindet sich immer das aktuelle Teilemessprogramm, sowie alle Daten, die das Messprogramm direkt betreffen. In der globalen Datenbank (GDB) befinden sich alle Daten, die für mehr als ein Teilemessprogramm gelten, aber benutzereigen sind. In der Umgebungsdatenbank (EDB) befinden sich alle Daten, die für mehr als ein Teilemessprogramm gelten sowie maschinenspezifisch sind (z.B. die Daten der Tasterkombinationen im Tastermagazin). In der Benutzer-Datenbank (UDB) befinden sich alle Daten, die für das Kundenunternehmen typisch sind, wie z.B. Plot-Formulare, Druckerformulare, Firmenzeichen, usw. [15].

Die genaue Beziehungen aller Datenbanken siehe Abbildung 2.24.

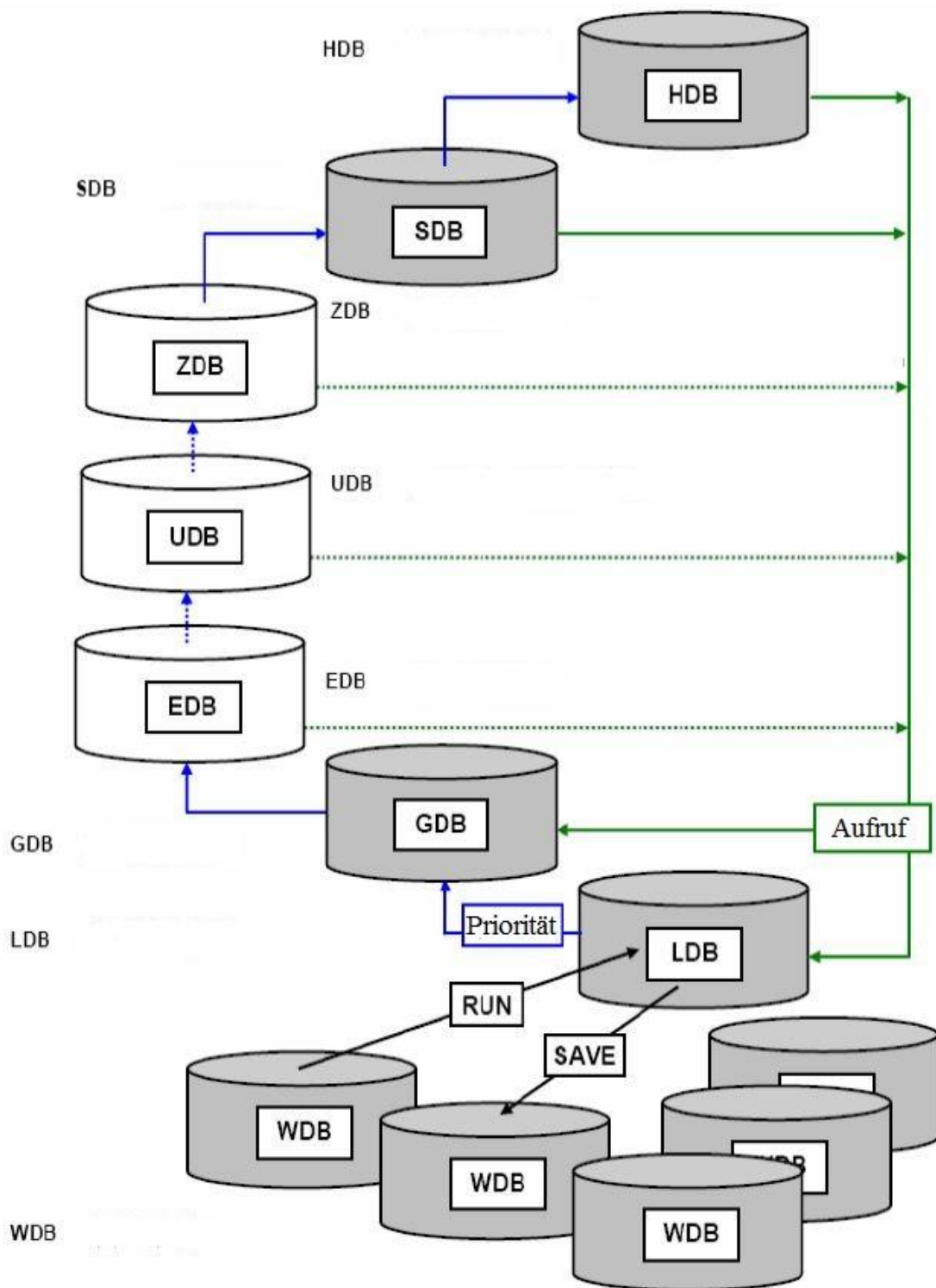


Abb. 2.24 Beziehungen aller Datenbanken in QUINDOS 7

3 Aktueller Messprozess und Entwurf des neuen Verfahrens

Die Einzelmessung von Bauteilen mit den aktuellen CNC-Messprogrammen ist problemlos möglich, da diese entsprechend konfiguriert und optimiert sind.

3.1 Der praktische Messprozess der aktuellen CNC-Messprogramme

Jedes Messprogramm wird neben dem zu messenden Bauteil durch verschiedene Randbedingungen bestimmt (charakterisiert). Das sind im Wesentlichen das relative und das absolute Koordinatensystem zum Bauteil, die zu verwendenden Taster und die Steuerdaten der entsprechenden Bauteile (Parameterblöcke). Der Ablauf einer Messung am KMG unterteilt sich bei Standardmessaufgaben in die nachfolgend aufgeführten Abschnitte: Tastereinmessung, Bauteilaufspannung, manuelle Ausrichtung, automatische Ausrichtung, Einlesen der Sollwerte aus den Parameterblöcken, Durchführung der Messung, Auswertung, Protokollerstellung und Protokollausgabe.

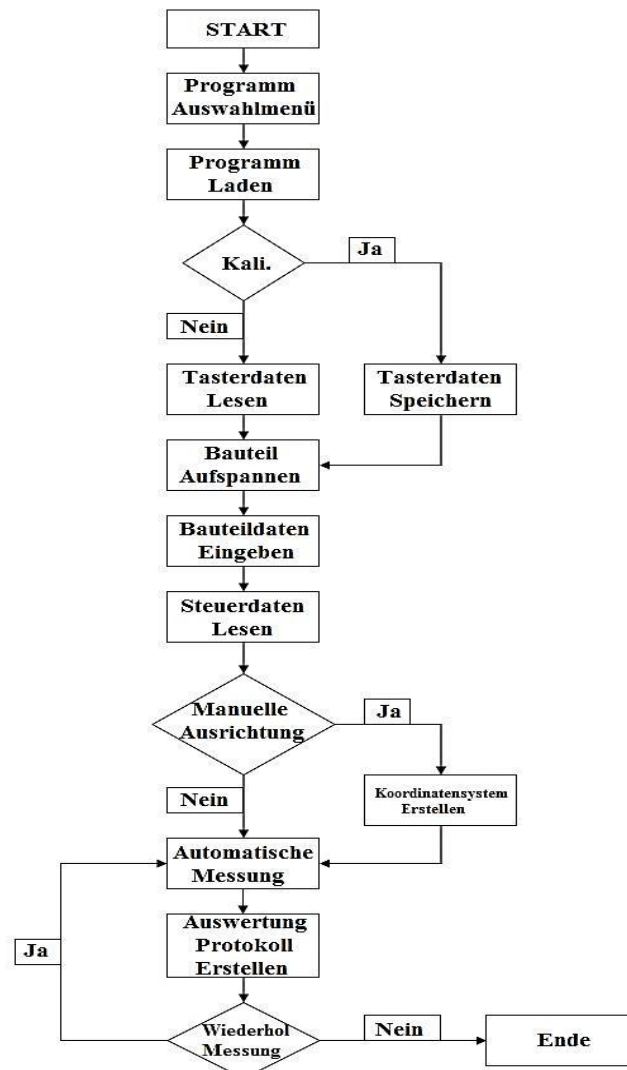


Abb. 3.1 Messungsprozess von aktuelle CNC-Messprogramm

Der Messprozess aller Bauteile ist fast gleich wie in Abbildung 3.1 gezeigt. Alle praktischen Messprozesse von den aktuellen Messprogrammen werden sehr gut geplant im Vergleich zum theoretischen Messungsprozess, weil die Messungsblöcke in Praxis (zum Beispiel: Kalibrierung, Bauteildaten Eingabe, Ausrichtung) flexibel deklariert werden.

Der Ablauf einer CNC-Messung bei Abteilung TP/PPH:

- **Programm Auswählen und Messprogramm Laden**
Im diesen Prozess kann man nur eine einzelne Bauteile auswählen. Es erzeugt eine Referenzzeiger, der ein bestimmtes zugehöriges Messprogramm verbindet. Dann verlässt es die Hauptfunktion, löscht alle aktuellen Variablen, lädt das gezeigte Messprogramm in die lokale Datenbank. Der Vorteil ist, alle Variable sind richtig für die Messbauteil. Wegen der Sicherheit von Messmaschine darf es keine falschen Parameter benutzen.
- **Einmessung von Taster**
Bei Tasterkalibrierung wegen nur eines einzelnen Bauteils soll geprüft werden, wann die letzte Kalibrierung gemessen wurde. Wenn die letzte Kalibrierung eines bestimmten Tastern vor mehr als einer Woche war, d.h. dass diese Messbauteilserie (zum Beispiel: Zahnrad, Nockenwelle, usw.) bei dieser Maschine schon eine Woche nicht gemessen wurde, muss es neu Kalibrieren. Nach der Kalibrierung müssen alle neue Tasterdaten (zum Beispiel: Tasterdurchmesser, Ablageindex von entsprechendem Taster, usw.) in einer spezifischen Datei, die für weitere Messung benutzbar ist, gespeichert werden.
Andererseits wenn keine neue Kalibrierung benötigt wird, liest das Programm die alte Kalibrierdatei der Tasterdaten ein. Hier muss es nur die Tastablageindex prüfen oder korrigieren.
- **Bauteilkenndaten Eingabe und Steuerdaten Einlesen**
Vor der Messung müssen alle Bauteilkenndaten (zum Beispiel: Motortyp, Motornummer, Identnummer, Zeichnungsnummer usw.) eingegeben werden. Bei aktuellem Messprogramm werden jedes Mal eindeutige Kenndaten für ein bestimmtes Bauteil eingegeben, d.h. dass in der gesamten Messung nur diesen Parameter benutzbar sind.
Dabei durch die Zeichnungsnummer kann ein zugehörige Steuerdaten, die genaue physikalische Größe von Messbauteil (Durchmesser Höhe, Tief) definiert, in der LDB eingelesen werden und 3D Modell für die Messung erstellt.
- **Manuelle Ausrichtung und automatische Ausrichtung**
Bei diesem Abschnitt wird das absolute Koordinatensystem für die Messung erzeugt. Das Koordinatensystem ist für spätere Messung sehr wichtig, für die Bewegung der Tasterkopf, für die Auswertung die Messwert, usw.

- **Automatische Messung und Messprotokoll Erstellen**
Automatische Messung heißt durch bestimmte Koordinatensysteme verschiedene Messpunkte zu sammeln. Im Messprotokoll wird der Vergleich der Istwerte zu den Sollwerten erzeugt. Durch entsprechende Auswertung kann visuell schnell erkannt werden, ob Toleranzen eingehalten werden. Dann kann man Entscheidung treffen, ob eine Wiederholungsmessung durchgeführt werden soll.

3.2 Entwurf eines flexiblen und effizienten Messprozesses

Damit verschiedene Bauteile automatisch nacheinander gemessen werden, sollen ohne manuellen Eingriff, um zu Verzapfung der Messprogramme eine neue Struktur erstellt werden.

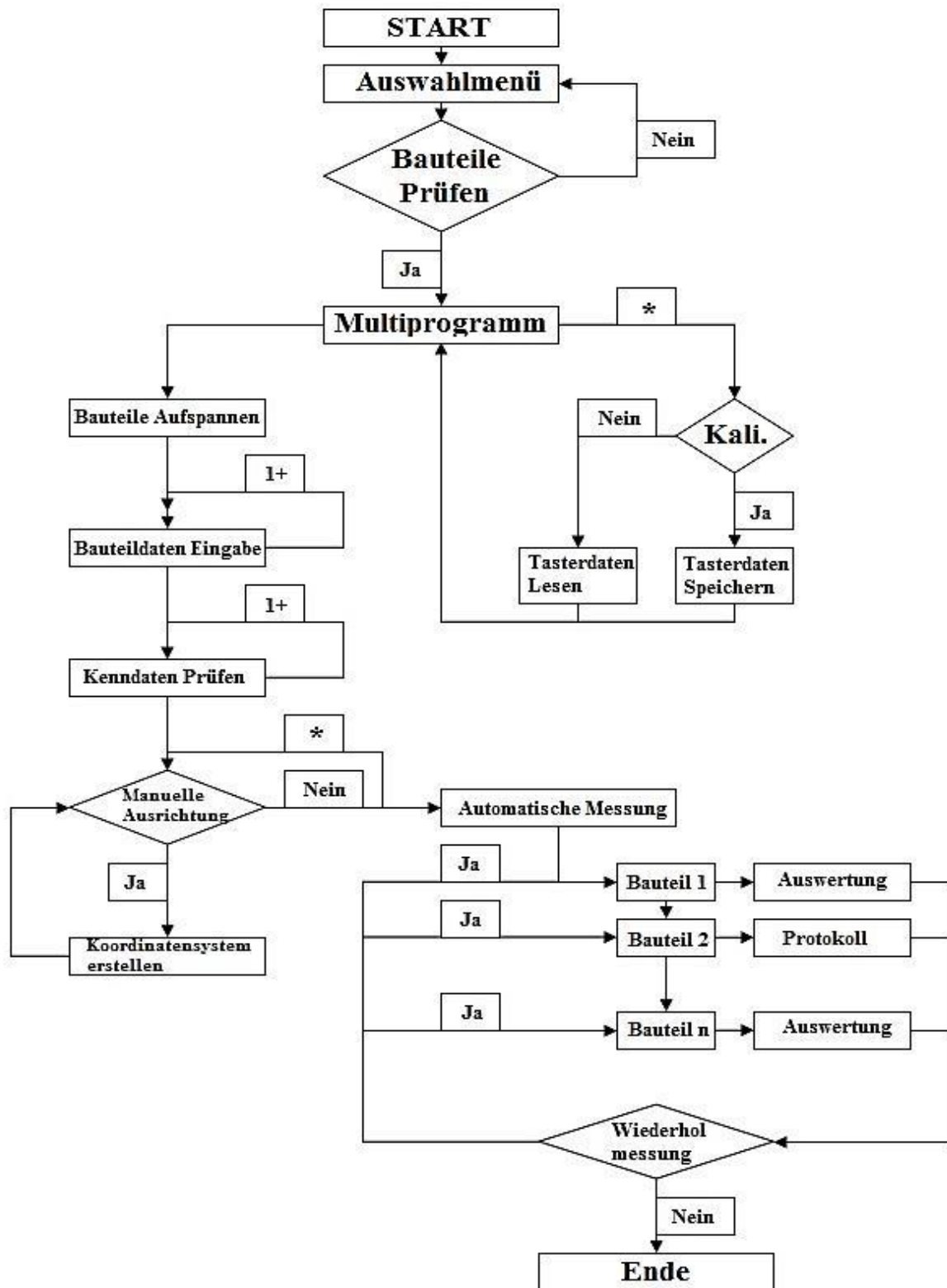


Abb. 3.2 Neuer Entwurf Messungsprozess für Multimesprogramm

Bei der angestrebten Messung verschiedener Bauteile in einer Aufspannung sind verschiedene Kriterien besonders zu berücksichtigen. Dies sind die zu verwendenden Taster, die Aufspannpaletten für die Bauteile und Grundmessfläche sowie Maschinenmessvolumen. Letztere sind relevant in Bezug auf die Verfahrswege der Maschine, die bei der Tastbewegung sicher zu keiner Kollision zwischen Taster und Bauteil führen dürfen. In der Abbildung 3.2 zeigt es, alle Messprozessblöcke sind flexibel verbunden.

In dem alten Prozess sind nur ein einzig Bauteil zu messen, d.h. dass alle Variable sind eindeutig deklariert. Nach der Messung gibt es alle benutzten Speicherplätze wieder frei. Aber bei Multibauteilmessung soll alle Variable in der globalen Datenbank gespeichert werden, Diese Vorteile ist alle aktuelle messende Bauteil nimmt die Variable von globale Datenbank aus zu lokale Datenbank. Wenn die Messung fertig ist, speichert es die Variable zurück zu globale Datenbank. Es ist für Wiederholungsmessung keine falsche Werte zu benutzen.

Multimessprogramm ist durch Entwurf in sechs Teile geteilt, siehe Abbildung 3.3, Hauptprogramm, Bauteil-Auswahl, Kenndaten-Eingabe, Taster-Kalibrierung, manuelle-Ausrichtung und Messaufgabe-Eingabe, automatische Messung und Wiederholungsmessung.

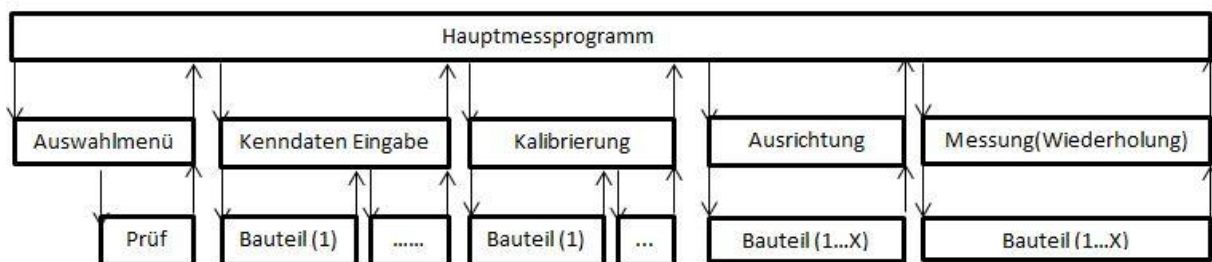


Abb. 3.3 Programmlaufstruktur

3.2.1 Auswahlmenü von Multibauteilmessprogramm

Bei jedem Multibauteilmessungsanfang soll eine technische Messplanung gemacht werden, welche Bauteile zusammen zu messen sind. Im Auswahlmenü können Mitarbeiter alle Bauteile eingeben, durch verschiedene Einschränkungen analysiert es automatische und gibt es ein Ergebnis aus. Wenn es negativ Ergebnis gibt, zeigt es eine Fehlermeldung, d.h. dass alle eingegebenen Bauteile nicht auf einmal zusammen gemessen werden. Das Programm springt zurück zum Auswahlmenü. Es ist eine Zuordnung, bis alle richtig angepasst werden.

Bei Multimessprogramm wird es am Anfang ein Dialog programmiert. Damit kann man definieren, wie viele Bauteile, welcher Bauteil und mit welchem Messgerät. Nach der Zuweisung prüft es automatische, ob es die Anzahl von ausgewählten Bauteilen gleich die Anzahl der zumessenden Bauteilen ist. Wenn es ungleich ist, springt es zurück zu diesem Dialogmenü. Die farbigen Felder sind Pflichtfelder, die unbedingt ausgewählt werden müssen. Wenn es etwas fehlt, springt das Programm auch zurück zum diesen Dialogmenü. Es ist nötig,

alle richtig einzugeben, weil Messgerät streng nach Messprogramm mit eingegebenem Messungsparameter durchlaufen.

The image shows a software dialog box titled 'Auswahlmenü' (Selection Menu) for a multi-measurement program. At the top, there are two dropdown menus: 'Anzahl von Bauteile' (Number of parts) and 'Maschine' (Machine). Below these, there are eight rows, each labeled 'Bauteile:1' through 'Bauteile:8', with a corresponding dropdown menu for each. The background of the dialog box has a dotted grid pattern.

Abb. 3.4 Auswahlmenü für Multimessprogramm

Es wird hier nur achte Bauteile in Dialog geplant. Wie in der Abbildung zeigt, es kann nur aus Popuplist ausgewählt werden, damit alle Schreibefehlen zu vermeiden. Die Zuweisung der Bauteilname ist mit einer kleinen Schnittstelle programmiert. Aber man kann spä auch beliebige Bauteile bei Bedarf erweitern, einfach die Index für Bauteil von dieser Schnittstelle erweitern. Programm für das Auswahlmenü ist im Anhang A1 zu finden.

3.2.2 Eingaben der Messkenndaten für jeden Bauteil

Die vom Auswahlmenü ausgewählte Bauteile muss hier alle zugehörigen Messprogramme laden, darin es alle Prozess Abschnitt definiert werden. Es ist sehr wichtig, weil alle Messprogramme hier verzahnt sind. Wenn es eindeutig definiert, kann es problemlos fließende laufen Man kann hier auch alle Prozessabschnitten als Unterprozeduren programmieren. Dann soll es nur eine statistische Variable deklarieren, als Schlüssel zu verwenden.

Mit allen richtigen zugewiesenen Bauteilnamen kann es jede nach jede zugehörige Eingabedialog von Kenndaten aufgerufen. Jede Messkenndaten werden in einer historischen Datenstruktur in globale Datenbank (GDB) gespeichert. Es ist spezifische Datenstruktur wie

Recorder bei Java oder C. Das ist sinnvolle und einfach, spä bei der automatische Messung alle Variablen wieder zu bekommen.

Die Bauteilkenndaten sind in einer Reihe eingegeben. Es ist möglich, dass der zweite Bauteil oder der dritte Bauteil von dem gleichen Motor ist, d.h. dass die Motornummer oder solche Variablen gleich sind. Hier wird es eine Funktion geschrieben, die eine Popupliste macht und aktuelle Bauteilkenndaten von vorherigem Bauteil auswählen kann, automatische zu übertragen wie in der Abbildung 3.5.

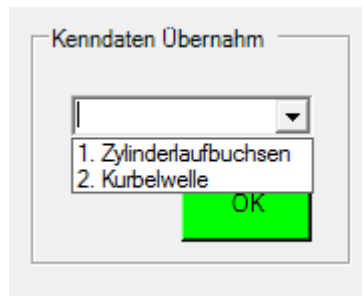


Abb. 3.5 dritter Bauteil Kenndaten Übernahme von erster oder zweiter Bauteil

In jedem Bauteil Kenndateneingabedialog wird das Funktionsdialog eingefügt, die nur durch Butten "OK" aufgerufen wird. Die Popupliste aktualisiert nur nach jeder Kenndateneingabe, d.h. dass es bei erstem Bauteil leer ist. In zwischen prüft das Programm, ob es alle richtig eingegeben wird. Wenn es nicht stimmt, meldet es eine visuelle Fehlermeldung. Damit können alle Mitarbeiter leicht Fehler korrigieren. Programm für die Kenndateneingabe ist im Anhang A2 zu finden.

3.2.3 Taster Einmessung für alle Bauteile

Es wird hier einen Zyklus für alle Tasterkalibrierung eingesetzt. Die kontrolliert, ob es neue Kalibrierung braucht. Von dieser Position muss es jeden Bauteil prüfen, weil die manche Tastdaten noch gültig sind.

Bevor jeder Taster Kalibrierung zeigt es das Programm visuelle Meldung, die richtig zugehöriger Bauteil zeigt. Es kann vermeiden, Taster Kalibrierung für falschen Bauteilen. Nach der Kalibrierung speichern alle Tastdaten in selbe ".PAR" Datei. Es ist nicht wie aktuelles Messprogramm, einfach Tastdaten weiter zu benutzen. In weiterem Messprogrammsteil muss alle Tastdaten, wenn es nötig ist, wieder geladen werden.

Taster Einmessungsvorgehensweise für Multibauteilmessprogramm:

- Im Tastkopf befindlichen Taster angeben
- Einmessungskugel definieren
- Referenztaster einmessen (Weiter Taster einmessen)

- Tastdaten speichern (Unerlöslich)
- Alte Tastdaten in der EDB löschen

Ein Beispielprogramm für Taster einmessen ist im Anhang A3 zu finden.

3.2.4 Manuelle Ausrichtung und Messaufgabe Eingabe

Für jede Messung bei aktuellem Messprogramm erzeugt erst ein zugehöriges relatives Koordinatensystem des zumessenden Bauteils, das von absolutem Koordinatensystem der Messmaschine referenziert. In diesem Abschnitt von Prozess muss es streng alle Tasterbewegungen in der Beziehung von den geometrischen Größen der Tasterbäumen und Messbauteilen, damit es bei automatischer Messung keine Kollision auftaucht.

Im diesen Programmteil ist es vorbereitet für automatische Messung. Die Koordinatensysteme der Bauteile werden unterschiedlich erzeugt. Teilweise Bauteilen sind durch ihre Paletten auf die Messtisch aufgespannt werden. In diesen Fall braucht der Mitarbeiter nur die Positionen der Paletten einzugeben. Andere Bauteilen werden frei auf den Messtisch aufgespannt. In diesen Fall soll der Mitarbeiter durch manuelles Antasten die Messpunkte sammeln, durch die Messpunkte kann es die Koordinatensystem erzeugt. Alle Positionen der Paletten oder die Koordinatensysteme werden in GDB gespeichert. Sie sind weiter in der automatischen Messung bei verschiedenen zugehörigen Bauteilen wieder in LDB einzuladen.

Die Messaufgaben (verschiedene Formprüfen oder unterschiedliche Bauteilmessorte usw.) sind hier eindeutig einzugeben. Es ist ähnliche wie Koordinatensystem erste in GDB zu speichern, nachher durch jeden Bauteil wieder in LDB einzuladen.

Ein Beispielprogramm für Manuelle Ausrichtung und Messaufgabe Eingabe ist im Anhang A4 zu finden.

3.2.5 Automatische Messung und Messprotokoll Erstellung

In diesem Abschnitt wird es so entworfen, dass es zunächst in einer Reihenfolge alle Messbauteile zu messen ist, dazu liefern die Ergebnisse und Protokolle. Die Messungsprogramme sind nur klein verändert. In der Messung laden nur die nötigen Kenndaten, relativ Koordinatensystem und Steuerdaten.

3.2.6 Wiederholungsmessung

Nach der gesamten Messung kontrollieren die Mitarbeiter die Messprotokolle. Es kann sein, manche Messpunkte sind zufällig außer Toleranz. Die Abweichungen zu die Normalform sind sehr groß, d.h. es unnormal gemessen wird. Die Gründe kommen von verschiedenen Ursachen. Deswegen braucht man diese Messpunkt noch einmal messen, um die Form zu prüfen, ob es zufällig oder tatsächlich ist. In diesen Fall sollen hier genaue zu definieren, welche Messpunkte nach gemessen werden sollen, welche nicht. Es ist ganz sinnvolle, weil die gesamte komplette Messung sehr lang dauert. Wenn es nur paar Messpunkte, die nach gemessen werden sollen, ausgewählt werden, kann es viele Messungszeit bei Wiederholungsmessung sparen.

Ein Beispielprogramm für Wiederholungsmessung ist im Anhang A5 zu finden.

3.3 Neues Multibauteilmessprogramm im Vergleich zum aktuellen Messprogramm

Das Programm an sich spart keine Zeit, da die Kalibrierung durchgeführt werden, Messung dauert bei gleicher Messstrategie so lang wie vorher. Die Vorbemerkung würde sogar noch etwas aufwändiger sein, da der Verlust der Messungszeit bei einem Programm Abbruch größer ist als beim aktuellen Programm.

Der Gewinn ist dabei zu sehen, dass die Maschine über einen längeren Zeitraum als vorher unbemannt läuft und der entsprechenden Mitarbeiter in dieser Zeit etwas anderes tun kann.

3.3.1 Messungslist von einen Tag im ganzen Messungsraum

Eine Messungslist wird gemacht, die Zeit von Messungsanfang- und Ende jedes getrennten gemessenen Messbauteilen zu speichern. Damit kann man feststellen, wie die Messmaschine läuft.

Zum Beispiel:

Tab. 3.1 Die normale Messungszeit von Messbauteilen

	Bauteil	Messprogramm	Messmaschine	Anfangszeit	Endzeit	Dauer
1	Lagerzapfen	Zapfen.wdb	PMC 12 10 7 #321	11:13:32	11:47:23	33
2	Zahnrad	Zahnrad_SM.wdb	PMC 12 10 7 #321	12:24:29	12:36:31	12
3	Nockenwelle1	Nockenwelle.wdb	CYG 20 12 10 #125	14:45:59	15:20:54	34
4	Nockenwelle2	Nockenwelle.wdb	CYG 20 12 10 #125	15:55:20	17:23:59	88

Die obige Tabelle 3.1 zeigt, dass die Messbauteile normale bei gleicher Messmaschine in einer Reihenfolge getrennt gemessen werden. Die ersten zwei Zeilen zeigen, dass zwei verschiedene Messbauteilen je nach einander gemessen werden. Die erste ist Lagerzapfen bei Messmaschine "PMC 12 10 7 #321" mit Anfangszeit "11:13:32" und Endzeit "11:47:23", der zweiten Bauteil ist Zahnrad mit Anfangszeit "12:24:29" und Endzeit "12:36:31". Inzwischen halbe Stunde braucht der Mitarbeiter neu Bauteil "Zahnrad" auf die Messmaschine aufzuspannen, zu Kalibrieren und so weiter. Aber die Messungsdauer von Zweite Messbauteil ist nur 12 Minuten. Dann stoppt die Messmaschine um "12:36:31" in der Mittagspause, d.h. dass die Messmaschine bis 13:15 stoppt. Aber bei Multimesprogramm kann man die zwei Messbauteil einmalige zusammen aufspannen, die Beiden ohne Pause automatische gemessen werden können. Vermutlich ist es, dass es 15 bis 20 Minuten sparen kann. Andere Seite, wenn die 20 Minuten gewinnen würden, kurz vor Mittagspause kann man schon weiteren Messbauteil messen. Das ist ein sehr gut Zyklus.

Die dritte und vierte Zeile zeigt, dass der gleiche Bauteil von verschiedenen Motoren mit unterschiedlichen Messaufgaben gemessen wird. Die Endzeit ist "17:23:59", Die Messmaschine stoppt dort bis nächster Tag. Mit Multimesprogramm kann es einfach weiteren Bauteil messen. Es kann auch viele Zeit sparen für nächsten Arbeitstag.

Die Messungsdauer, die meistens automatische Messungszeit ist, steckt eigentlich nahezu kein Optimierungspotential mehr. Es hängt von jeder Messungsaufgabe an (zum Beispiel: bei "Nockenwelle 2" alle komplette Lager zu messen oder "Nockenwelle 1" nur ausgewählte Lager zu messen, usw. Hier hinweist, dass die Messdauer eines Programms neben den aufzumerkenden Messpunkten ganz von der Messstrategie bestimmt wird.).

3.3.2 Messungslist von Einer Messmaschine in ein paaren Tage

Es wird jetzt versucht, eine Messmaschine mit seiner Messungslist zu beobachten.

zum Beispiel:

Tab. 3.2 Eine Messmaschine in ein paaren Tage mit seiner Messungslist

	Bauteil	Messprogramm	Messmaschine	Anfangszeit	Endzeit	Dauer
1	Laufbuchsen1	Buchsen.wdb	PMC 24 12 10 #393	13:38:34	17:10:35	212
2	Pleuelstange1	Pleuel.wdb	PMC 24 12 10 #393	05:38:54	07:13:33	94
3	Pleuelstange2	Pleuel.wdb	PMC 24 12 10 #393	07:17:49	07:27:31	9
4	Nockenwelle1	Nockenwelle.wdb	PMC 24 12 10 #393	11:37:13	12:23:41	46
5	Laufbuchsen2	Buchsen.wdb	PMC 24 12 10 #393	14:48:28	18:14:40	206
6	Pleuelstange3	Pleuel.wdb	PMC 24 12 10 #393	05:46:13	06:57:04	70
7	Nockenwelle2	Nockenwelle.wdb	PMC 24 12 10 #393	07:13:16	07:43:51	30
8	Nockenwelle3	Nockenwelle.wdb	PMC 24 12 10 #393	07:45:44	08:13:09	27
9	Nowerahmen	Nowerahmen.wdb	PMC 24 12 10 #393	08:35:03	09:31:47	56
10	Nockenwelle4	Nockenwelle.wdb	PMC 24 12 10 #393	10:02:36	11:14:46	72
11	Nockenwelle5	Nockenwelle.wdb	PMC 24 12 10 #393	11:40:30	13:53:54	133
12	Laufbuchsen3	Buchsen.wdb	PMC 24 12 10 #393	14:05:58	17:34:37	208

In obiger Tabelle 3.2 zeigt, dass eine Messmaschine in zwei kompletten Tagen durchgelaufen hat. Es ist bemerkbar, dass alle Mitarbeiter schon sehr gute Plan gemacht haben. Sie habe alle Messbauteile, die ganz lange Messdauer benötigt, am Arbeitstagsende zu messen gelassen. Bei aktuellem Messprogramm kann es nicht weiter verbessern. Aber bei Multimessprogramm kann man noch besseren Plan machen. In Messungsraum gibt es immer zumessende Messbauteile für die nächsten paaren Tagen. Das bedeutet, dass es für die obige Messbauteile vermutlich einen neuen Messungsplan machen kann.

Es kann beispielsweise Pleuelstange 1, Pleuelstange 2 und Pleuelstange3 einmal zusammen gemessen werden, Nockenwelle 1 und Nockenwelle 2 einmal zusammen zu messen, Nockenwelle 3 und Nockenwellerahmen einmal zusammen zu messen, Nockenwelle 4 und Nockenwelle 5 einmal zusammen zu messen. Am ersten Abend kann es Zylinderlaufbuchse 1 und Zylinderlaufbuchse 2 zusammen messen, und am zweiten Abend kann es Zylinderlaufbuchse 3 und mit noch anderen Messbauteil zusammen messen. Theoretische kann es halben Arbeitstag bis einen Arbeitstag für die Messmaschine sparen. Es kann man so verstehen, 3 Tagarbeit in 2 bis 2.5 Tage fertig, spart es 16% bis 33% der Stillstandzeit der Maschine.

3.3.3 Bewertung für die Idee des Multimesprogramm

Es ist zwei Seiten zu denken. Eine Seite, bei Arbeitstaganfang kann Mitarbeiter mehrere Bauteilen einmalige aufspannen, die mit Multimesprogramm zu messen. Mansche Messbauteilen (zum Beispiel: Lagerzapfen, Zahnrad) müssen jede 15 bis 20 Minuten gewechselt werden. Aber es kann durch Multimesprogramm diese Störung vermeiden. In zwischen Messungszeit ohne Störung ist der Mitarbeiter frei, kann er etwas anderes tun.

Andere Seite, bei Arbeitstagende kann Mitarbeiter auch mehrere Bauteile, die im optimalen Fall längere Messdauer brauchen, einmalige aufspannen. Danach lassen sie sich durch Multimesprogramm messen. Die Messmaschine läuft automatische am Abend. Es spart die Stillstandzeit der Maschine in der Nacht.

4 Ergebnisse der neuen Messprogramm für Motorenmessung

Die Multimessprogramm ermöglicht die kontinuierliche Messung. Die unterbrochen Aufspannung von einem Bauteil zu anderem Bauteil werden vermieden. Und Zeit- und Personalaufwand wird eingespart.

Die Leistungsfähigkeit des entwickelten Verfahrens zur Motorenmessung wird durch die Bewertung folgender Beispiele gezeigt.

4.1 Versuchsbeispiel Nr.1

Im Abschnitt 3 wurde der entwickelte Messungsplan dargestellt. Der Versuch wurde an einem 6-Zylinder-Verbrennungsmotor mit seinen Nockenwellen durchgeführt. Jeder Verbrennungsmotor hat zwei Nockenwellen, eine für die Einlassseite andere für die Auslassseite. Das Ziel des Versuchs war die Realisierung zwei frei aufgespannte Messbauteile sequentiell automatische zu messen.

Die Messungsaufträge sind in Tabelle 4.1 dargestellt.

Tab. 4.1 die Nockenwellen von Motor FH5511

Auftragsnummer	F70590	Auftragsnummer	F70590
Motornummer	FH5511	Motornummer	FH5511
Datum	24.01.2013	Datum	24.01.2013
Auftraggeber	Singer	Auftraggeber	Singer
Von Abteilung	TP/PHC	Von Abteilung	TP/PHC
Bezeichnung	Nockenwelle Auslass	Bezeichnung	Nockenwelle Einlass
Zeichnungsnummer	A4700500501	Zeichnungsnummer	A4700500401
Identnummer	4518804C1	Identnummer	3391310C1
Messauftragsart	Zwischenmessung	Messauftragsart	Zwischenmessung
Laufzeit	1696 Stunden	Laufzeit	1696 Stunden
Messpunkt	051.03.010	Messpunkt	051.04.010
Messpunkt	051.03.013	Messpunkt	051.04.013
Messpunkt	051.03.021	Messpunkt	051.04.021
Messpunkt	051.03.022	Messpunkt	051.04.022
Messpunkt	051.03.023	Messpunkt	051.04.023
Messpunkt	051.03.032	Messpunkt	051.04.032

Die Messpunkte in der Tabelle 4.1 bedeutet die Messaufgabe von dem Auftraggeber, die in der Abteilung TP/PPH zu messen sind. Die Messpunkte für Nockenwelle Auslass in dem Messkatalog sind teilweise wie Tabelle 4.2 gezeigt. In der Tabelle 3.2 wird die letzte Spalte "Dauer" gezeigt. Die Messungsdauer hängt von den Messpunkten ab. Je mehr Messpunkte zu

messen sind, desto längerer Messungsdauer es braucht. Noch sinnvoller für diesen Versuch ist, dass jeder Motor zwei Nockenwellen haben, d.h. dass jeder Auftrag für Nockenwellen zwei Nockenwellen gemessen werden. Wie in der Tabelle 3.2 gezeigt hat, jetzt alle Nockenwellen getrennt gemessen werden, inzwischen es viele Zeit verloren hat. Nach dem Versuch kann man später Messungsplan machen wie in der Vermutung, unnötigen Zeitaufwand zu vermeiden.

Tab. 4.2 Messkatalog für Nockenwelle Auslass

Bauteil	Messpunkt	Prüfmerkmal
Nockenwelle Auslass	051.03.010	Lager, Durchmesser in A+B Richtung in 2 Messebenen
	051.03.011	Lager, Rundheit
	051.03.012	Lager, Mantellinie
	051.03.013	Lager, Rundlauf
	051.03.014	Planlauf vom Nockenwellenrad
	051.03.021	Nockenwinkel, Rundlauf, Grundkreis, Nockenform
	051.03.022	Nockenhöhe und Grundkreis-Ø
	051.03.023	Nockenlauf, Nockenspitze, Nockenablauf Mantellinie
	051.03.032	Winkelstellung, Abstecknut und Absteckfläche zur Indexbohrung
	051.03.033	Nockenabstand

In diesen Fall während ProgrammDurchlauf müssen Bauteilkenndaten, Bauteilpositionen, Bauteilmessaufgaben und Bauteilsteuerdaten durch verschiedene Verfahren in der richtigen Schnittstellen zugewiesen oder definiert werden.

Erste Schritt ist zumessende Bauteile und Messmaschinen auszuwählen, wie in der Abbildung 4.1 zeigt. Die Felder mit der Farbe Khaki bedeutet Pflichtfelder, es muss unbedingt ausgewählt werden. Andere Felder wie zum Beispiel "Bauteil 2" hängen von Anzahl von Bauteil. Wenn der Mitarbeiter das vergessen, auszuwählen. Das Programm läuft nicht weiter, springt es zurück zu diesem Dialog, Fehlereingabe zu vermeiden.

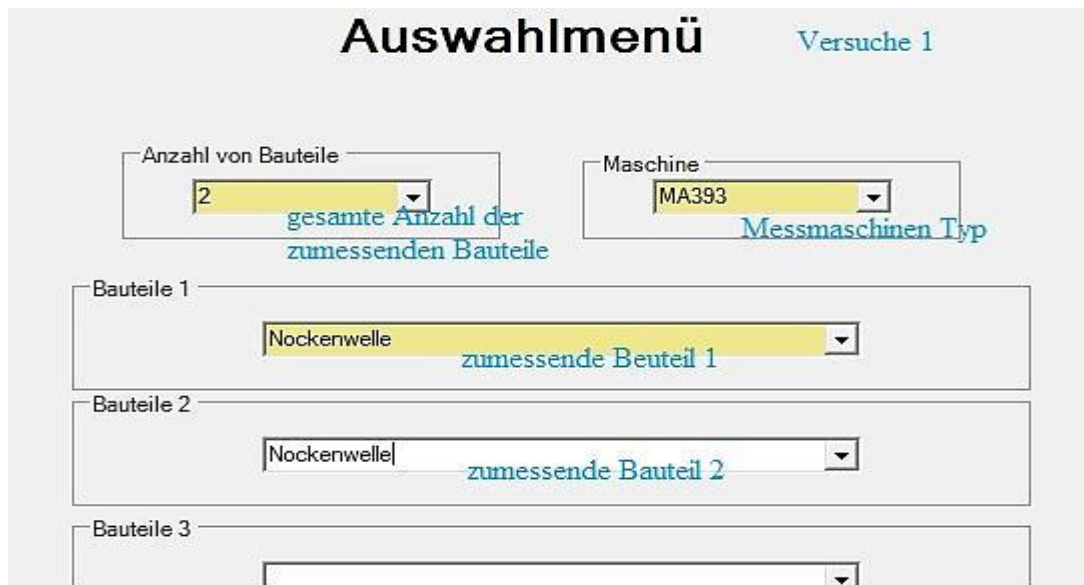


Abb. 4.1 Auswahlmenü für ersten Versuch, zwei Nockenwellen von einem Motor

Nach der Eingabe ruft es die Prozedur "Zuordnung", die prüft, ob alle zumessenden Bauteile nach den Randbedingungen zusammenpassen. Wenn alles in Ordnung ist, ruft die Prozedur "Daten_Eingabe", speichert es die Bauteilkenndaten in GDB, wie in der Abbildung 4.2 zeigt. Alle Bauteilkenndaten in GDB können Daten in der Messung selbst LDB zugreifen, verloren es in der LDB oder WDB Wechsel nicht.

GDBHRC		Daten		Versuch 1													
MTVzLogoBox		ErrorLog		InfoBox		HRC Speicherfelder											
Descriptors : \$A \$B \$C \$D \$E \$F \$G \$H \$I \$J \$K \$L \$M \$N \$O \$P \$Q \$R \$S \$T \$U \$V \$W \$X \$Y \$Z \$a \$b \$c \$d \$e \$f \$g																	
\$A	\$B	\$C	\$D	\$E	\$F	\$G	\$H							\$I	\$J		
Auslass Nockenwelle	4518804C1	A4700500501	~Einlass	A4720510600	A4720510400	OM470	FH551										
Einlass Nockenwelle	3391310C1	A4700500401	A4720510500	~Auslass	~Brems	OM470	FH551										

Abb. 4.2 erster Versuch Bauteilkenndaten in GDB

Bauteilpositionen sind ganz wichtig in diesem Versuch, weil Messmaschinen alle zumessenden Bauteile ihre Positionen in der automatischen Messung erkennen möchten. Bei frei aufgespannten Bauteilen braucht es bestimmt einmal manuelle Ausrichtung, die ein grobes Koordinatensystem erzeugt. Für Nockenwelle hat es so gemacht, dass einmal manuell Antasten für Z Punkt, die X-Y Ebene definiert. Mindestens drei manuelle Antasten an erstem Lager, die generiert Kreis 1, und mindestens drei Antasten an letztem Lager, die generiert Kreis 2, von Kreis 1 und Kreis 2 Mittelpunkt generiert die Achse Z, dann sammelt alle Punkte ein grobes Koordinatensystem erzeugt. Bei dieser Situation kann die Koordinatensystem in GDB mit ihrem Index speichern lassen. Die praktische Aufspannung siehe Abbildung 4.3.

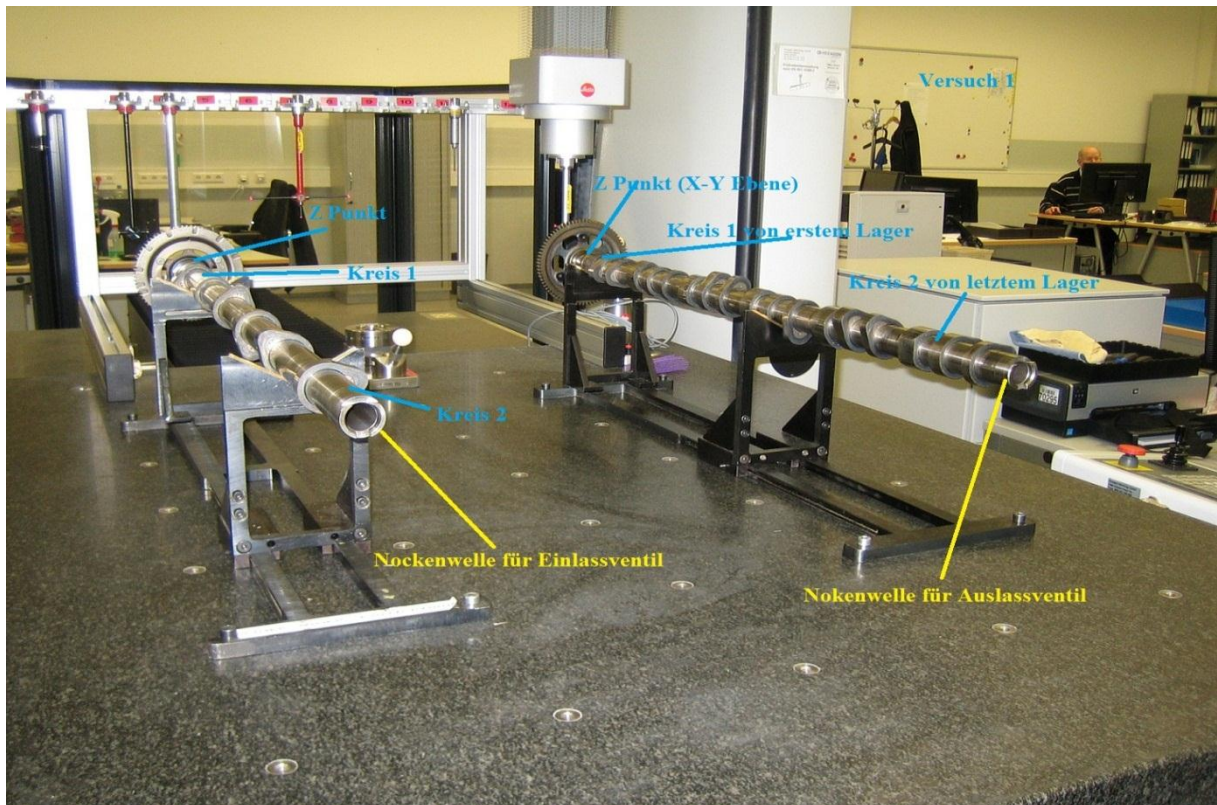


Abb. 4.3 zwei Nockenwellen einmal auf der Messmaschine aufgespannt

Am Ende zeigt es das Messprotokoll siehe Abbildung 4.4. Durch die Messprotokolle kann man visuell schnell erkennen, ob die Toleranzen eingehalten werden oder einen unnormale Istwert auftaucht. Der Mitarbeiter soll befinden die Ursache dieses unnormalen Istwerts und entscheiden, ob es eine Wiederholmessung durchführt. In Abbildung 4.4 zeigt, dass alle Istwerte eingehalten werden, d.h. dass die Messung gültig ist.

Messblatt Nockenwelle					
Daimler AG				Abt: TP/PPH	
Zeichnung-Nr.	A4700500501	Nockenform [E]	~Einlass	Ident-Nr.	4518804C1
Motor-Nr.	FH5511	Nockenform [A]	A4720510800	Prüfer	Novelli
Laufleistung	1696 Stunden	Nockenform [F]		Prüfdatum	28-JAN-2013, 12:04
Ein/Ausbau	Zwischen	Nockenform [B]	A4720510400		
		Bemerkung			

Text	Ausw.	Istmaß	Nennmaß	O.Tol.	U.Tol.	Ist-Soll	Grafik
Auslassnocken							
A11(i)		2DE					
W	ANG_CSX	-257.7208	-257.9000	0.5000	-0.5000	0.1792	
FS	ERR_STP	0.0038	0.0000	0.0150	0.0000	0.0038	
RL	RAD_RUN	0.0078	0.0000	0.0500	0.0000	0.0078	
F1	FORM	0.0105	0.0000	0.1000	0.0000	0.0105	
A1	CAM_HGT	59.1134	59.2040	0.0000	-0.2000	-0.0906	

Abb. 4.4 Messprotokoll von Versuch 1

Messblatt Nockenwelle					
Daimler AG				Abt: TP/PPH	
Zeichnung-Nr.	A9380501001	Nockenform [E]	A9380510200	Ident-Nr.	8259319C
Motor-Nr.	TD3471	Nockenform [A]	~Auslass	Prüfer	Novelli
Laufeistung	0 Stunden	Nockenform [F]		Prüfdatum	29-JAN-2013, 07:50
Ein/Ausbau	Zwischen	Nockenform [B]	~Bremsen		
		Bemerkung			

Text	Ausw.	Istmaß	Nennmaß	O.Tol.	U.Tol.	Ist-Soll	Grafik
------	-------	--------	---------	--------	--------	----------	--------

Einlassnocken

e1(1)		2DE					
W	ANG_CSX	-204.1270	-204.4200	0.3330	-0.3330	0.2930	
FS	ERR_STP	0.0153	0.0000	0.0200	0.0000	0.0153	
RL	RAD_RUN	0.0036	0.0000	0.0300	0.0000	0.0036	
FI	FORM	0.0299	0.0000	0.1000	0.0000	0.0299	
AI	CAM_HGT	53.3968	53.6127	0.0000	-0.2000	-0.2159	

Abb. 4.4 Messprotokoll von Versuch 1

Es zeigt die Messungsdaten, die gesamte Laufzeit und gemessene Bauteile, siehe Tabelle 4.3.

Tab. 4.3.a Messdaten von dem Versuch 1

Bauteil	Messmaschine	Anfangszeit	Endzeit	Dauer (Minute)
Nockenwelle Auslass	PMC 24 12 10 #393	29.JAN.2013 13:56:47	29.JAN.2013 16:23:19	147
Nockenwelle Einlass	PMC 24 12 10 #393	29.JAN.2013 16:23:32	29.JAN.2013 17:26:09	62

Tab. 4.3.b Messdaten von dem Versuch 1

Bauteil 1	Bauteil 2	Messmaschine	Anfangszeit	Endzeit	Dauer(Minute)
Nockenwelle Auslass	Nockenwelle Einlass	PMC 24 12 10 #393	29.JAN.2013 13:56:38	29.JAN.2013 17:26:13	209

Die gesamte Messungsdauer für den Versuch 1 ist 209 Minuten wie in der Tabelle 4.3.b zeigt, Tabelle 4.3.a zeigt die Messungsdauer jeder Nockenwelle. Nach der manuelle Dateneingabe und Ausrichtung maß die Messmaschine die beiden Nockenwellen automatisch, die Endzeit von Nockenwelle Auslass ist Anfangszeit von Nockenwelle Einlass.

4.2 Versuchsbeispiel Nr.2

Der Versuch wurde Pleuelstange von zwei 6-Zylinder-Verbrennungsmotoren durchgeführt. Das Ziel des Versuchs war Realisierung zwei durch Messpalette aufgespannte Bauteile sequentiell automatische zu messen.

Die Messungsaufträge sind in Tabelle 4.4 dargestellt.

Tab. 4.4 Messungsaufträge für Pleuelstange von Motor AE3921 und AE8419

Auftragsnummer	F70736	Auftragsnummer	F70404
Motornummer	AE3921	Motornummer	AE8419
Datum	21.01.2013	Datum	21.01.2013
Auftraggeber	Szattat	Auftraggeber	Singer
Von Abteilung	TP/PHC	Von Abteilung	TP/PHC
Bezeichnung	Pleuelstange	Bezeichnung	Pleuelstange
Zeichnungsnummer	A4710300120	Zeichnungsnummer	A4710300220
Identnummer	4376725A 1-6	Identnummer	9941878C 1-6
Messauftragsart	Ausbaumessung	Messauftragsart	Einbaumessung
Laufzeit	4313 Km	Laufzeit	0 Stunde
Messpunkt	038.01.040	Messpunkt	038.01.020
Messpunkt	038.01.041	Messpunkt	038.01.021
Messpunkt	038.01.042	Messpunkt	038.01.022
Messpunkt	038.01.051	Messpunkt	038.01.040
Messpunkt	038.01.052	Messpunkt	038.01.041
Messpunkt		Messpunkt	038.01.042
Messpunkt		Messpunkt	038.01.050
Messpunkt		Messpunkt	038.01.052

Die Messpunkte in der Tabelle 4.4 bedeutet die Messaufgabe von dem Auftraggeber, die in der Abteilung TP/PPH zu messen sind. Die Messpunkte für Pleuelstange in dem Messkatalog sind teilweise wie Tabelle 4.5 gezeigt hat. Hier zeigt es eindeutig, die Messpunkte für gleiche Bauteil Pleuelstange aber verschiedene Messauftragsart unterschiedlich sind. Das kann am Ende zeigen, dass es die unterschiedliche Messungsdauer gibt.

Tab. 4.5 Messkatalog für Pleuelstange

Bauteil	Messpunkt	Prüfmerkmal
Pleuelstange	038.01.010	Großes Auge, Grundbohrung Durchmesser in A+B+C Richtung in 2 Messebenen
	038.01.011	Großes Auge, Grundbohrung Rundheit
	038.01.020	Pleuellagerschalen eingebaut, \emptyset A+B+C in 3 Messebenen
	038.01.021	Pleuellagerschalen eingebaut, Rundheit
	038.01.022	Pleuellagerschalen eingebaut, Mantellinie in A-Richtung
	038.01.030	Kleines Auge Grundbohrung, Durchmesser in A+B+C Richtung in 2 Messebenen
	038.01.031	Kleines Auge Grundbohrung, Rundheit
	038.01.040	Pleuelbuchse eingepresst, Durchmesser in A+B Richtung
	038.01.041	Pleuelbuchse eingepresst, Rundheit
	038.01.042	Pleuelbuchse eingepresst, Mantellinie
	038.01.050	Achsabstand großes Auge - kleines Auge (nur mit Lagerschalen)
	038.01.051	Achsparallelität zwischen großem und kleinem Auge in Längs- und Querrichtung
	038.01.052	Flucht zwischen großem und kleinem Auge, mit Lagerschalen

In diesen Fall während Programmdurchlauf müssen auch alle Bauteilkenndaten, Bauteilpositionen, Bauteilmessaufgaben und Bauteilsteuerdaten durch verschiedene Verfahren in der richtigen Schnittstellen zugewiesen oder definiert werden.

Erste Schritt ist zuzumessende Bauteile und Messmaschinen auszuwählen, wie in der Abbildung 4.5 zeigt.

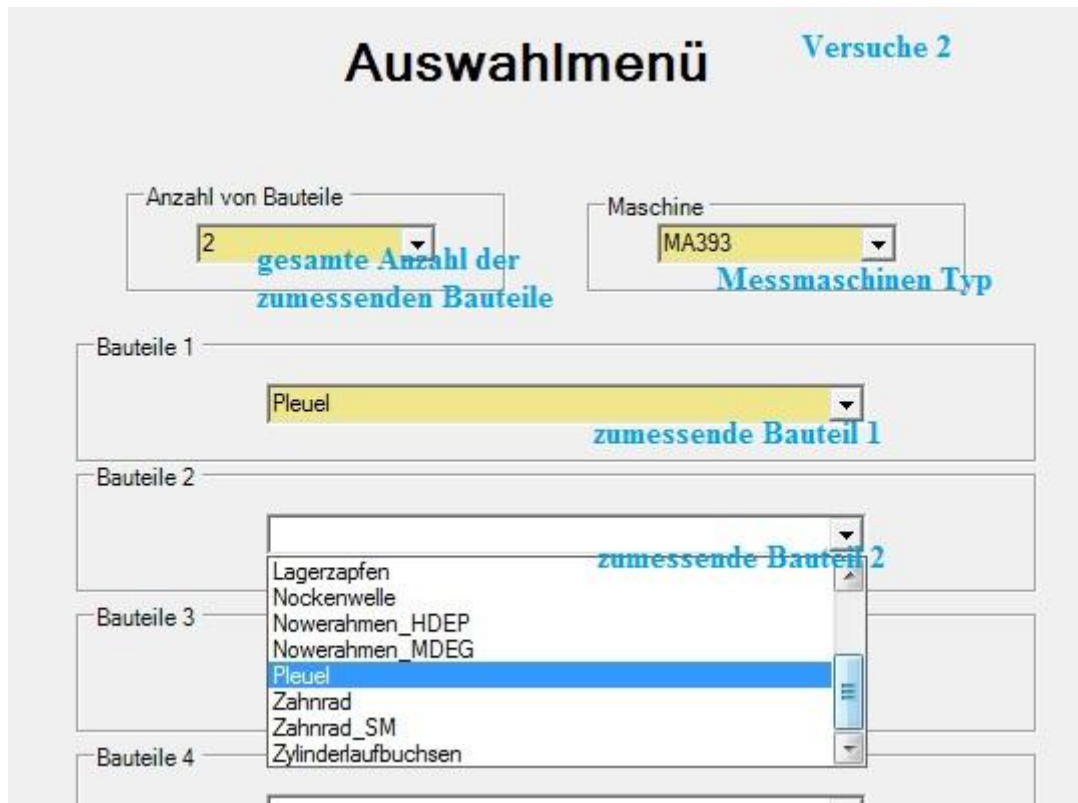


Abb. 4.5 Auswahlmenü für zweiten Versuch, Pleuelstange von zwei Motoren

Nach der Eingabe ruft es die Prozedur "Zuordnung" und "Daten_Eingabe", speichert es die Bauteilkenndaten in GDB, wie in der Abbildung 4.6 zeigt.

GDBHRC		Daten		Versuch 2																	
MTWzLogoBox		ErrorLog		InfoBox																	
Descriptors : \$A \$B \$C \$D \$E \$F \$G \$H \$I \$J \$K \$L \$M \$N \$O \$P \$Q \$R \$S \$T \$U \$V \$W \$X \$Y \$Z \$a \$b \$c \$d \$e \$f \$g \$h																					
HRC Speicherfelder																					
\$B	\$C	\$D	\$E	\$G	\$H	\$I	\$J	\$K	\$M	\$N	\$O	\$P	\$Q	\$R							
~CMM	OM471	AE3921	A4710300120	1	A	Nein	Ja	Nein	0	F70736	Szattat	6	284.00	4313.00							
MA393	OM471	AE8419	A4710300220	1	E	Ja	Ja	Nein	0	F70404	Singer	6	0.00	0.00							

Abb. 4.6 zweiter Versuch Bauteilkenndaten

Bauteilpositionen hängen hier von den Positionen der Paletten ab. Hier wird durch manuelle Ausrichtung Paletten Koordinatensystem erzeugt werden. Für dieses Koordinatensystem soll es einmal antasten die Palettenebene (X-Y Ebene), einmal antasten für Y0 Punkt und zweimal antasten für X-Achse wie die Abbildung 4.7 zeigt. Von diesem Koordinatensystem der Palette kann jede Position der Pleuelstange automatisch auf der Palette verteilt werden. Untere Tabelle 4.6 zeigt die Verteilung der Palette.

Tab. 4.6 automatische Positionsverteilung von dem Koordinatensystem der Palette

Position	X-Position von Referenzkoordinatensystem (mm)	Y-Position von Referenzkoordinatensystem (mm)
1	70	93
2	150	173
3	230	253
4	370	333
5	450	413
6	530	593
7	670	673
8	750	753

Die praktische Aufspannung für zwei Paletten von Pleuelstangen siehe Abbildung 4.7.a. Es ist auf dem Messtisch verstellbar. Durch manuelle Ausrichtung für jede Palette, siehe Abbildung 4.7.b, speichert es nur die Koordinatensysteme von den Paletten. Die Positionen der Pleuelstange werden automatisch durch die Tabelle 4.6 und Koordinatensystem der Palette verteilt.



Abb. 4.7.a zwei Messpaletten einmal auf Messmaschine Aufspannung

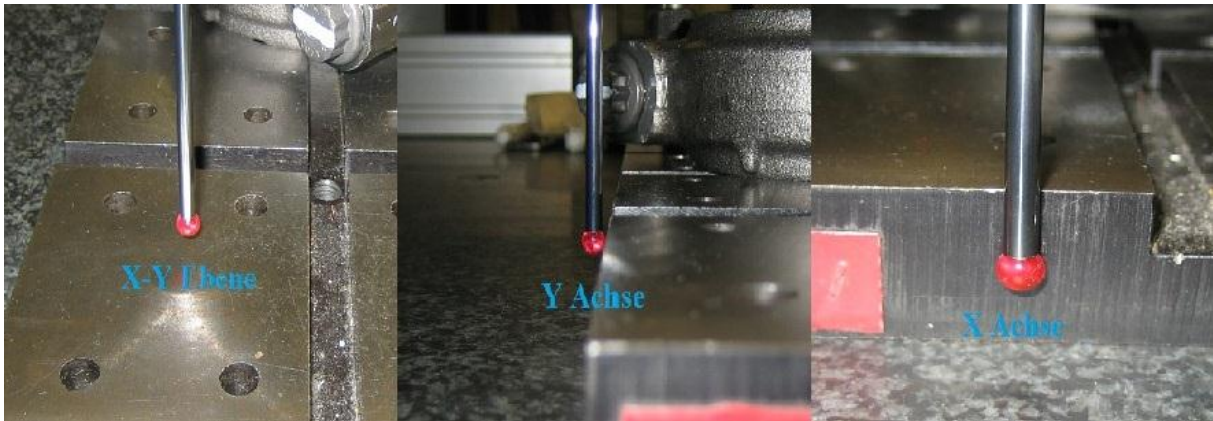


Abb. 4.7.b manuelle Ausrichtung für eine Palette

Nach der automatischen Messung zeigt die Messprotokoll von den beiden Pleuelstangen, siehe Abbildung 4.8. Für die Ausbaumessung der Pleuelstange ist die Durchmesser in B und C Richtungen 0.0029 mm und 0.0006 mm größer als Obertoleranz, es ist normalerweise akzeptiert, d.h. die Messung gütig ist.

Geometrische Motorenmesstechnik		DAIMLER	
Messprotokoll		3D-Anwendungstechnik	
Bezeichnung	Pleuel 1 von 6	Sachnummer	
Zeichnungs-Nr.	A4710300120	Identnummer	4376725A1
Bemerkung		Bearbeitungs-Nr.	246
Motornummer	AE3921	Laufzeit (Std.)	284.00
Motortyp	OM471	Laufstrecke (km)	4313.00
Auftragsnummer	F70736	Losgröße	mit Lager
Einbau/Ausbau/ZW	A1	Stichproben-Nr.	mit Buchse
Auftraggeber	Szattat	Abteilung	TP/PPH
Prüfer	Krauter	Prüfdatum	22-JAN-2013, 11:23:57
Messgerät	PMC 24 12 10 #393	Messprogramm	Pleuel

Text	Ausw.	Istmaß	Nennmaß	O.Tol.	U.Tol.	Ist-Soll	Grafik
Durchmesser grosses Auge							
Winkel der B- und C- Richtung zur Trennfläche in Grad : 30.0							
		ZNR_Deckel	A4710380010				
		ZNR_Stange	A4710380011				
A-oben	DM	95.0882	95.0000	0.1100	0.0640	0.0882	
A-mitte	DM	95.0875	95.0000	0.1100	0.0640	0.0875	
A-unten	DM	95.0795	95.0000	0.1100	0.0640	0.0795	
B-oben	DM	95.1033	95.0000	0.1600	0.0940	0.1033	
B-mitte	DM	95.0972	95.0000	0.1600	0.0940	0.0972	
B-unten	DM	95.0911	95.0000	0.1600	0.0940	0.0911	

Abb. 4.8 Messprotokoll von Versuch 2

Geometrische Motorenmesstechnik		DAIMLER	
Messprotokoll		3D-Anwendungstechnik	
Bezeichnung	Pleuel 1 von 6	Sachnummer	
Zeichnungs-Nr.	A4710300220	Identnummer	9941878C1
Bemerkung		Bearbeitungs-Nr.	03729
Motornummer	AE8419	Laufzeit (Std.)	0.00
Motortyp	OM471	Laufstrecke (km)	0.00
Auftragsnummer	F70404	Losgröße	mit Lager
Einbau/Ausbau/ZW	E1	Stichproben-Nr.	mit Buchse
Auftraggeber	Singer	Abteilung	TP/PPH
Prüfer	Krauter	Prüfdatum	22-JAN-2013, 11:55:47
Messgerät	PMC 24 12 10 #393	Messprogramm	Pleuel



Text	Ausw.	Istmaß	Nennmaß	O.Tol.	U.Tol.	Ist-Soll	Grafik
Durchmesser grosses Auge							
Winkel der B- und C- Richtung zur Trennfläche in Grad : 30.0							
		ZNR_Deckel		A4710380010			
		ZNR_Stange		A4710380011			
A-oben	DM	95.0815	95.0000	0.1100	0.0640	0.0815	
A-mitte	DM	95.0838	95.0000	0.1100	0.0640	0.0838	
A-unten	DM	95.0868	95.0000	0.1100	0.0640	0.0868	
B-oben	DM	95.1298	95.0000	0.1600	0.0940	0.1298	
B-mitte	DM	95.1294	95.0000	0.1600	0.0940	0.1294	
B-unten	DM	95.1207	95.0000	0.1600	0.0940	0.1207	
C-oben	DM	95.1098	95.0000	0.1600	0.0940	0.1098	

Abb. 4.8 Messprotokoll von Versuch 2

Es zeigt die Messungsdaten, die gesamte Laufzeit und gemessene Bauteile, siehe Tabelle 4.7.

Tab. 4.7.a Messdaten von Versuch 2

Bauteil	Messmaschine	Anfangszeit	Endzeit	Dauer (Minute)
Pleuelstange 1	PMC 24 12 10 #393	22.JAN.2013 11:23:43	22.JAN.2013 11:54:58	31
Pleuelstange 2	PMC 24 12 10 #393	22.JAN.2013 11:55:33	22.JAN.2013 13:00:40	65

Tab. 4.7.b Messdaten von Versuch 2

Bauteil 1	Bauteil 2	Messmaschine	Anfangszeit	Endzeit	Dauer(Minute)
Pleuelstange 1	Pleuelstange 2	PMC 24 12 10 #393	22.JAN.2013 11:23:43	22.JAN.2013 13:00:41	96

Die gesamte Messungsdauer für den Versuch 2 ist 96 Minuten wie in der Tabelle 4.7.b zeigt, Tabelle 4.7.a zeigt die Messungsdauer jedes Pleuelstangeauftrags. Nach der manuelle Dateneingabe und Ausrichtung maß die Messmaschine die beiden Pleuelstangeaufträge automatische, die Endzeit von Pleuelstange 1 ist Anfangszeit von Pleuelstange 2.

In dem entwickelten Messprogramm werden die Koordinatensysteme der Paletten in der "TASTER_PLEUEL_MAXXX.PRB" ("XXX" ist die Maschinentyp, 125, 321, 393 und 416) mit ihren Indexnummern, die von Hauptprogramm eingesetzte Bauteilnummer ist, gespeichert. In diesen Fall braucht es für nächste Messung gleiche Position der Palette nicht mehr manuelle Ausrichtung. Das bedeutet, dass es durch die Auswahl des vorherigen gemessenen Messbauteils das Koordinatensystem automatisch laden, wie Abbildung 4.9 zeigt. Programm für das Koordinatensystem speichern und laden ist im Anhang A6 zu finden.

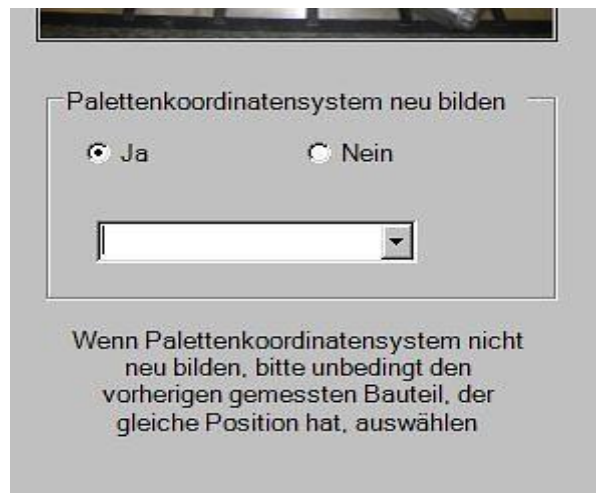


Abb. 4.9 Palettenkoordinatensystem Auswahl

4.3 Versuchsbeispiel Nr.3

Der Versuch wurde Pleuelstange und Zylinderlaufbuchsen von zwei 6-Zylinder-Verbrennungsmotoren durchgeführt. Das Ziel des Versuchs war Realisierung zwei durch Messpalette oder Befestigungsbohrung aufgespannte Bauteile sequentiell automatische zu messen.

Die Messungsaufträge sind in Tabelle 4.8 dargestellt.

Tab. 4.8 Messungsaufträge für Zylinderlaufbuchse und Pleuelstange von zwei Motoren

Auftragsnummer	F70515	Auftragsnummer	F70615
Motornummer	AE8414	Motornummer	AE8413
Datum	23.01.2013	Datum	03.01.2013
Auftraggeber	Walker	Auftraggeber	Lohrengel
Von Abteilung	TP/PHH	Von Abteilung	TP/PHC
Bezeichnung	Zylinderlaufbuchse	Bezeichnung	Pleuelstange
Zeichnungsnummer	A4710111010	Zeichnungsnummer	HNF471030B000
Identnummer	9111284C 1-6	Identnummer	3199715C 1-6
Messauftragsart	Ausbaumessung	Messauftragsart	Ausbaumessung
Laufzeit	0 Stunde	Laufzeit	1642 Stunde
Messpunkt	011.10.040	Messpunkt	038.01.040
Messpunkt		Messpunkt	038.01.041
Messpunkt		Messpunkt	038.01.042
Messpunkt		Messpunkt	038.01.050

Die Messpunkte von Pleuelstange ist wie in der Tabelle 4.5 zeigt. Die Messpunkt von Zylinderlaufbuchse wird in der Tabelle 4.9 zeigt.

Tab. 4.9 Messkatalog für Zylinderlaufbuchse

Bauteil	Messpunkt	Prüfmerkmal
Zylinderlaufbuchse	011.10.010	Einpassen und Bund-Ø, Durchmesser in A+B Richtung
	011.10.012	Feuerring Einstich, Pass-Ø und Tiefe in A+B Richtung
	011.10.020	Bund, Höhe
	011.10.021	Bund, Einstichtiefe
	011.10.022	Bundauflage, Planlauf
	011.10.030	Zylinderbohrung, Bohrungs-Ø in A+B Richtung, Messebenen nach Vorschrift (max.17)
	011.10.031	Zylinderbohrung, Rundheit
	011.10.040	Zylinderlaufbahn, Zwickelmessung

Messprogramm durchläuft wie vorherige zwei Versuche, gemessene Bauteile auszuwählen, Bauteilkenndaten einzugeben, Kenndaten in der GDB zu speichern. Die manuelle Ausrichtung für Palette der Pleuelstange. Für die Positionen der Zylinderlaufbuchsen wird im aktuellen Messprogramm immer Befestigungsbohrung definiert, die von Koordinatensystem der Messmaschinen abhängen. Die Verteilung für die Befestigungsbohrung der Messmaschinentisch ist unveränderbar. Aber in dem entwickelten Programm können die Positionen der zumessenden Zylinderlaufbuchsen ausgewählt werden. Das passt flexible zu anderem Messbauteil, beispielweise wie in diesem Fall "Pleuelstange".

Die praktische Aufspannung für den Versuch 3 siehe Abbildung 4.10.

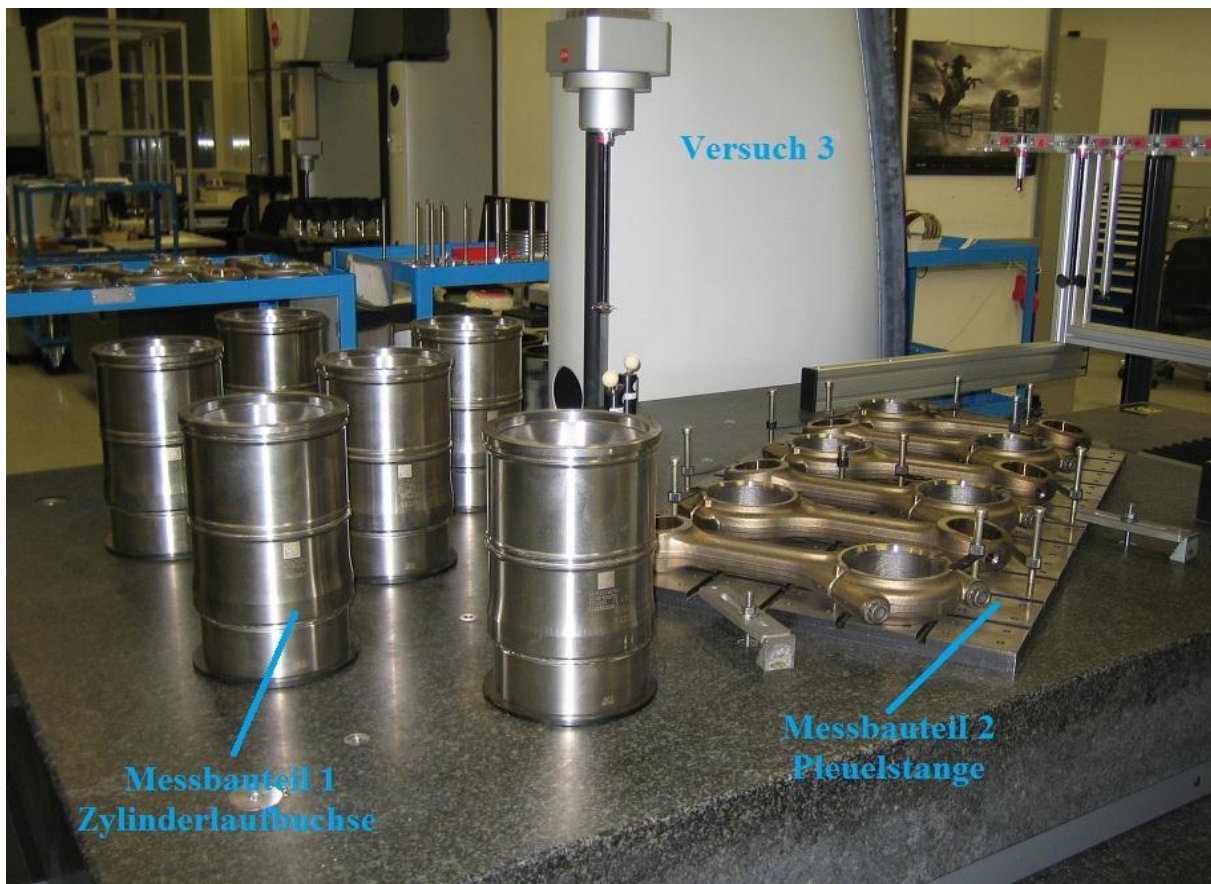


Abb. 4.10 Aufspannungsplan für den Versuch 3

Befestigungsbohrungen werden hier für alle Messmaschinen neu definiert, die später für automatische Messung benutzt werden. Die Positionen der zugehörigen Zylinderlaufbuchsen werden erst vor automatische Messung ausgewählt, wie Abbildung 4.11 zeigt. Es ist jetzt sinnvoll, die Messungsposition kann beliebig auf dem Messtisch verteilt werden.

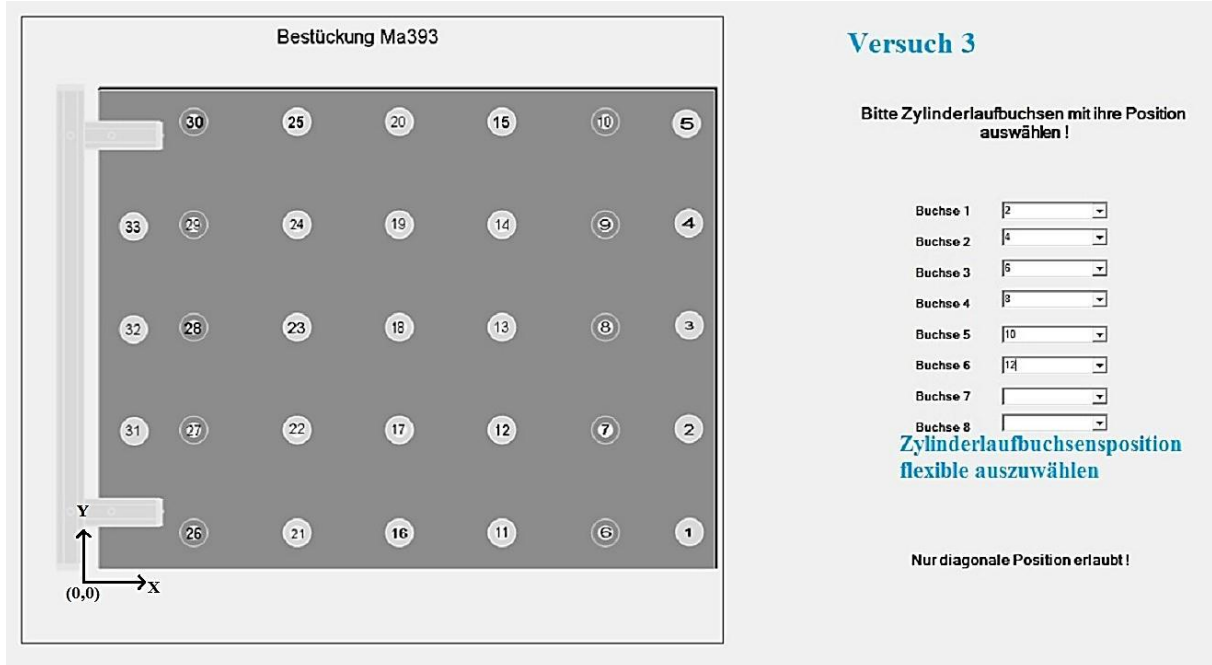


Abb. 4.11 die Dialog für Messungsposition Auswahl

Die Positionen (x,y) der Befestigungsbohrungen werden in der Tabelle 4.10 gezeigt. Hier ist "X_OFF" gleich 610 mm und "Y_OFF" gleich 213 mm, das Abweichen von Messmaschinen Ursprungspunkt.

Tab. 4.10 die Positionen der Befestigungsbohrungen

Position	X-Position von Referenzkoordinatensystem (mm)	Y-Position von Referenzkoordinatensystem (mm)
1	$MX(1)=X_OFF+1200$	$MY(1)=Y_OFF$
2	$MX(2)=X_OFF+1200$	$MY(2)=Y_OFF+200$
3	$MX(3)=X_OFF+1200$	$MY(3)=Y_OFF+400$
4	$MX(4)=X_OFF+1200$	$MY(4)=Y_OFF+600$
...
16	$MX(16)=X_OFF+600$	$MY(16)=Y_OFF$
17	$MX(17)=X_OFF+600$	$MY(17)=Y_OFF+200$
18	$MX(18)=X_OFF+600$	$MY(18)=Y_OFF+400$
19	$MX(19)=X_OFF+600$	$MY(19)=Y_OFF+600$
20	$MX(20)=X_OFF+600$	$MY(20)=Y_OFF+800$
...
32	$MX(32)=X_OFF$	$MY(32)=Y_OFF+400$
33	$MX(33)=X_OFF$	$MY(33)=Y_OFF+600$

Nach der automatischen Messung zeigt die Messprotokolle der beiden Bauteilen, siehe Abbildung 4.12. Durch die Kontrolle der Messprotokolle kann man festlegen, ob die Messungswerte gültig sind.

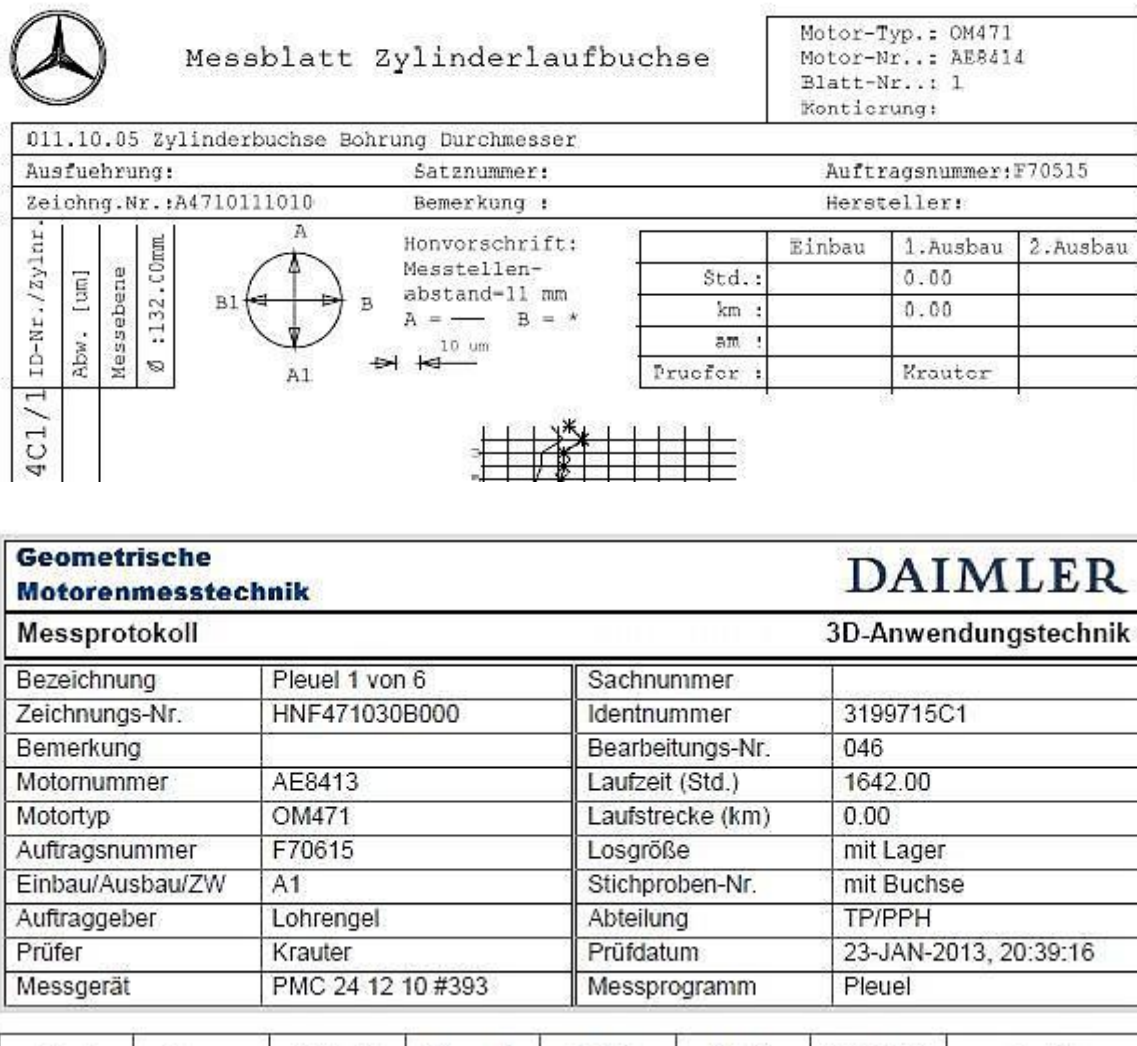


Abb. 4.12 Messprotokoll für Versuch 3

Es zeigt die Messungsdaten, die gesamte Laufzeit und gemessene Bauteile, siehe Tabelle 4.11.

Tab. 4.11.a Messdaten von Versuch 3

Bauteil	Messmaschine	Anfangszeit	Endzeit	Dauer (Minute)
Zylinderlaufbuchse	PMC 24 12 10 #393	23.JAN.2013 17:07:35	23.JAN.2013 20:38:46	211
Pleuelstange	PMC 24 12 10 #393	23.JAN.2013 20:38:58	23.JAN.2013 21:22:01	43

Tab. 4.11.b Messdaten von Versuch 3

Bauteil 1	Bauteil 2	Messmaschine	Anfangszeit	Endzeit	Dauer(Minute)
Zylinderlauf- buchse	Pleuel- stange	PMC 24 12 10 #393	23.JAN.2013 17:07:30	23.JAN.2013 21:22:41	254

Die gesamte Messungsdauer für den Versuch 3 ist 254 Minuten wie in der Tabelle 4.11.b zeigt, Tabelle 4.7.a zeigt die Messungsdauer jedes Bauteils. Nach der manuelle Dateneingabe und Ausrichtung maß die Messmaschine die beiden Bauteilen automatische, die Endzeit von Zylinderlaufbuchse ist Anfangszeit von Pleuelstange.

4.4 Weitere Versuchsergebnisse

Weitere Versuche werden mit verschiedenen Bauteilen durchgeführt, die Zuverlässigkeit des Verfahrens zu prüfen. Versuche werden am Messmaschine PMC 24 12 10 #393 und CYG 20 12 10 #125 getestet. Die Ergebnisse zeigen in der Tabelle 4.12.

Tab. 4.12 Ergebnisse von weiteren Versuchen

Bauteil 1	Bauteil 2	Messmaschine	Anfangszeit	Endzeit	Dauer (Minute)
Zylinderlaufbuchse Nr. 2	Nockenwelle Nr. 1	PMC 24 12 10 #393	25.JAN.2013 15:55:41	25.JAN.2013 21:07:12	312
Nockenwelle Nr. 2	Nockenwelle NR. 3	PMC 24 12 10 #393	06.FEB.2013 06:36:13	06.FEB.2013 10:08:07	211
Nockenwelle Nr. 4	Nockenwelle Nr. 5	PMC 24 12 10 #393	06.FEB.2013 15:41:39	06.FEB.2013 19:16:20	214
Nockenwelle Nr. 6	Nockenwelle Nr. 7	PMC 24 12 10 #393	07.FEB.2013 09:56:21	06.FEB.2013 11:09:06	72
Zylinderlaufbuchse Nr. 2	Zylinderlaufbuchse Nr. 3	PMC 24 12 10 #393	07.FEB.2013 16:37:33	07.FEB.2013 21:46:25	309
Zylinderlaufbuchse Nr. 4	Nockenwelle Nr. 8	PMC 24 12 10 #393	08.FEB.2013 12:13:27	08.FEB.2013 16:20:33	247
Zylinderlaufbuchse Nr. 5	Nockenwelle Nr. 9	CYG 20 12 10 #125	14.FEB.2013 06:15:41	14.FEB.2013 16:05:09	589
Pleuelstange Nr. 1	Pleuelstange Nr. 2	PMC 24 12 10 #393	14.FEB.2013 14:21:10	14.FEB.2013 16:48:13	147
Pleuelstange Nr. 3	Pleuelstange Nr. 4	PMC 24 12 10 #393	15.FEB.2013 13:48:57	15.FEB.2013 15:10:52	81

Bauteil 1	Bauteil 2	Bauteil 3	Bauteil 4	Messmaschine	Anfangszeit	Endzeit	Dauer
Zylinderlaufbuchsen Nr. 6	Nockenwelle Nr.10	Nockenwelle Nr. 11	Kolbenbolzen Nr. 1	CYG 20 12 10 #125	15.FEB.2013 11:59:17	15.FEB.2013 19:38:55	459

4.5 Bewertung des entwickelten Verfahrens

Die oben beschriebenen Versuchsbeispiele und ihre Ergebnisse bei Multimesprogramm bringen einige Schlussfolgerungen über das entwickelte Verfahren.

- Die Bauteile, die unter Einhaltung der Randbedingungen sind, können mit dem neuen Multimesprogramm gemessen werden.
- Die zugehörigen Steuerdaten für jedes zu messende Bauteil werden in den Schnittstellen automatisch zugewiesen. Die Messmaschine kann daraus ein 3D Model erzeugen.
- In der automatischen Messung werden alle benutzten Koordinatensysteme richtig gewechselt. Die Referenzkoordinatensysteme werden in der Messung automatisch gebildet.
- Das Multimesprogramm ist "Intelligent" genug, um Einzelbauteile zu isolieren. Das ermöglicht, dass für spezielle Messaufgabe später nach der automatischen Messung z.B. Wiederholmessungen für die selbe Bauteil möglich.

In den Versuchen wurden die Bauteile "Nockenwelle", "Pleuelstange" und "Zylinderlaufbuchse" mit verschiedenen Kombinationen gemessen. Dabei wurde ein positives Ergebnis erzielt. Der Kapitel 3.3.2 zeigt eine Auflistung von Messungen über einige Tage. Die positiven Effekt des Konzepts, wenn die Bauteile "Nockenwelle", "Pleuelstange", "Zylinderlaufbuchse" und "Nockenwelle Rahmen" so geprüft, können sie mit dem neuen Verfahren gemessen werden. Das reduziert die Stillstandzeit der Maschine. Die Flexibilität des Verfahrens ermöglicht bei entsprechender Vorbereitung längerer Maschinenlaufzeiten in Zeiträumen, die sonst ungenutzt bleiben würden.

5 Zusammenfassung und Ausblick

Mit der vorliegenden Arbeit wurde die Aufgabenstellung erfolgreich bearbeitet und das gesetzte Ziel vollständig erreicht, da das entwickelte Messprogramm die automatische Messung für einmalige Aufspannung verschiedener Bauteile auf der Messmaschine ermöglicht.

Die kontinuierliche Messungsprozesse, die rechtzeitige Datenbankzugriffe im Messprozess, die systematische Überprüfung von Messmaschinenrandbedingungen, die automatische Erkennung von physikalischen Größen der Bauteile, die richtige Definition der Koordinatensysteme vermeiden die Unterbrechung der automatische Messung und können Zeit- und Personalaufwand einsparen, ohne dass in den Ablauf der Messung manuell eingegriffen werden muss.

Eine Beendigung der langdauere Messung würde nur Wiederholenmessung durchgeführt werden, wenn die Messdaten sehr schlecht sind und die Messdaten nicht mehr gültig für weitere Auswertung sind. In diesen Fall soll nur die Wiederholenmessung für die bestimmten Messaufgabe oder bestimmen Bauteilstelle durchgeführt werden.

Die vorgeschlagene Architektur des Verfahrens ist flexibel und erlaubt, unterschiedliche Bauteile zusammen dauerhaft gemessen zu werden. Dazu ist die Zuverlässigkeit des Verfahrens sichert, ohne die Messmaschine zu beschädigen.

Die automatische Erstellung der notwendigen Parameter, Variablen und Bedingungen für alle Messungen, die graphische Darstellung des Programms sowie zusätzlich gemeldete Information über eine Messaufgabe bzw. einen individuellen Prozess machen die Benutzung des Messprogramms hilfreich und bedienungsfreundlich.

Die Ergebnisse der Versuche zeigen, dass die Messungsfähigkeit jeder Messmaschine durch neues Messprogramm ungefähr von 10% bis 30% erhöht. Das reduziert viele unbenötigte Arbeit und steigert die Rentabilität von Motorenmessung deutlich.

Die Programmierung in "QUINDOS 7", die von Anfang Idee bis Ende komplette Verpackung als ein Softwareprodukt ist, ermöglichen künftige Verbesserungen und Erweiterungen der Anwendung, beispielweise:

- Neue Prozedur für die Ausrichtung der verschiedenen Bauteile können immer wieder eingefügt werden, um eine günstigere Verteilung der Bauteile im Messmaschinenvolumen zu realisieren.
- Eine einfache Implementierung weiterer Module, z.B. automatische Bauteilerkennung durch Befestigungsbohrungen der Messmaschine.
- Die Kollision in der automatischen Messung kann vermieden werden, wenn die Messungsgeschwindigkeit durch die erfahrungsbasierten Parameter gesunken sind.

- Eine verbesserte Messpalette für die Aufspannung der unterschiedenen Bauteile kann durch Programmierung automatische zumessende Messbauteilposition erkennen, manuelle Ausrichtung zu vermeiden.

A Anhänge

A 1: Messprogramme für Auswahlmenü

- Haupt Messprogramm:

! Messprogramme Speichersort

```
~Pfad='Q:\QDA 8\CNC QDA8\Multi\  
GDBCHS:~Pfad=~Pfad
```

! alle Programm ist hier verpackt

! Zuweisung: Stringvariable von LDB zum GDB

! Löschen alle ungewünschte Variablen

```
DELCHS (NAM=CHS:~*(), CNF=N, TYP=CHS, NOT=~Pfad)  
DELREA (NAM=REA:~*(), CNF=N)  
DELELE (NAM=Lgd:~*(), CNF=N)  
DELTXT (NAM=TXT:~*(), CNF=N)  
DELUVS (NAM=UVS:~*(), CNF=N)  
DELELE (NAM=GDBHRC:Daten, CNF=N) ! Löschen HRC Variable von GDB  
DELELE (NAM=XML:Bauteil, CNF=N)  
DELELE (NAM=HRC:~*(), CNF=N, NOT=HRC:BESTUECKEN)
```

! Auswahlmenü für alle Bauteilen

```
DELETE (NAM=Lgd:Auswahlmenü, CNF=N)  
EDIT (NAM=Lgd:Auswahlmenü, CRE=Y)
```

! Label 1, Auswahlmenüdialog

```
1:ShowDialog (DLP=Auswahlmenü, OBJ=Lgd:Auswahlmenü, DID=1111)
```

! Bekommen alle Pflichte Variablen, und prüfen, ob alle richtig eingegeben werden

```
GETVALS (OBJ=Auswahlmenü, TYP=Lgd, RDS=a, REA=ges_anz, CDS=D, CHS=~MA)  
IFEXEC (VL1=ges_anz, VL2=0, TYP=EQ, CMD='UNWIND (FRA=1, LAB=1)')  
IFEXECS (ST1=~MA, ST2=' ', TYP=EQ, CMD='UNWIND (FRA=1, LAB=1)')
```

! Bekommen alle weitere Variablen

```
IFTHEN (VL1=ges_anz, VL2=3, TYP=EQ)  
GETVALS (OBJ=Auswahlmenü, TYP=Lgd, CDS=(A,B,C,D), CHS=(~Bauteil(1), ~Bauteil(2), ~Bauteil(3),  
~MA)) ! Felder "A,B,C,D" mit Stringvariable schon verbindet  
ELSEIF (VL1=ges_anz, VL2=2, TYP=EQ)  
GETVALS (OBJ=Auswahlmenü, TYP=Lgd, CDS=(A,B,D), CHS=(~Bauteil(1),~Bauteil(2),~MA))  
ELSEIF (VL1=ges_anz, VL2=1, TYP=EQ)  
GETVALS (OBJ=Auswahlmenü, TYP=Lgd, CDS=(A,D), CHS=(~Bauteil(1),~MA))  
ELSEIF (VL1=ges_anz, VL2=0, TYP=EQ)  
CVSTRTXT (TXT=Ges_anz, LIN=1, STR=")  
CVSTRTXT (TXT=Ges_anz, LIN=2, STR=Prüfen Sie bitte die Eingabe von "Gesamte Anzahl!")  
CVSTRTXT (TXT=Ges_anz, LIN=3, STR=Es muß aus POPUP ausgewählt werden!)
```

! bessere Interaktivität zwischen Menschen und Maschinen

```
ShowPicAndText (TXT=Ges_anz, STL=Normal, WAI=Y, UNI=mm)  
UNWIND (FRA=1, LAB=1)
```

ELSE

```
GETVALS (OBJ=Auswahlmenü, TYP=Lgd, CDS=(A,B,C,D,E,F,G,H,I), CHS=(~Bauteil(1), ~Bauteil(2),  
~Bauteil(3), ~MA, ~Bauteil(4), ~Bauteil(5), ~Bauteil(6), ~Bauteil(7), ~Bauteil(8)))
```

ENDIF

! Prüfen, ob alle richtig eingegeben werden

```
IFTHENS (ST1=~Bauteil(1), ST2=' ', TYP=EQ)  
CVSTRTXT (TXT=Bauteil, LIN=1, STR=")  
CVSTRTXT (TXT=Bauteil, LIN=2, STR=Prüfen Sie bitte die Eingabe von dem Bauteil 1!)  
CVSTRTXT (TXT=Bauteil, LIN=3, STR=Es muß aus POPUP ausgewählt werden!)  
ShowPicAndText (TXT=Bauteil, STL=Normal, WAI=Y, UNI=mm)  
UNWIND (FRA=1, LAB=1)  
ELSEIF (VL1=ges_anz, VL2=1, TYP=GT)  
IFTHENS (ST1=~Bauteil(2), ST2=' ', TYP=EQ)  
CVSTRTXT (TXT=Bauteil, LIN=1, STR=")
```

```

CVSTRTXT (TXT=Bauteil, LIN=2, STR=Prüfen Sie bitte die Eingabe von dem Bauteil 2!)
CVSTRTXT (TXT=Bauteil, LIN=3, STR=Es muß aus POPUP ausgewählt werden!)
ShowPicAndText (TXT=Bauteil, STL=Normal, WAI=Y, UNI=mm)
UNWIND (FRA=2, LAB=1)
ENDIF
ENDIF
IFTHEN (VL1=ges_anz, VL2=3, TYP=GE)
DO (NAM=I, BGN=3, END=ges_anz, DLT=1)
IFTHENS (ST1=~Bauteil(I), ST2=' ', TYP=EQ)
CVSTRTXT (TXT=Bauteil, LIN=1, STR=")
CVSTRTXT (TXT=Bauteil, LIN=2, STR=Prüfen Sie bitte die Eingabe von dem Bauteil !)
CVSTRTXT (TXT=Bauteil, LIN=3, STR=Es muß aus POPUP ausgewählt werden!)
ShowPicAndText (TXT=Bauteil, STL=Normal, WAI=Y, UNI=mm)
UNWIND (FRA=3, LAB=1)
ENDIF
ENDDO
ENDIF

```

! Prüfen, ob Randbedingungen erfüllt wird, Prozedur "Zuordnung aufrufen"

```
INDPRC (NAM=Zuordnung)
```

```
GDBCHS::~MA=~MA
```

! Zuweisung: Stringvariable von LDB zum GDB

- Prozedur "Zuordnung":

! Prozedur Zuordnung

! Prüfen, ob die gesamte Anzahl von Tasterbüme weniger oder gleich Maximale Anzahl der Ablagen ist

! Definieren Integer Variable "MAX" durch verschiedene Messmaschinentyp

```
IFTHENS (ST1=~MA, ST2=321, TYP=EQ)
```

```
MAX=11
```

```
ELSEIFS (ST1=~MA, ST2=125, TYP=EQ)
```

```
MAX=13
```

```
ELSEIFS (ST1=~MA, ST2=393, TYP=EQ)
```

```
MAX=13
```

! neue kommende Messmaschine, die noch nicht in Messprogramm integriert werden

```
ELSEIFS (ST1=~MA, ST2=416, TYP=EQ)
```

```
! MAX=
```

```
ENDIF
```

! initialisieren alle Variablen

```
gesamt_Tasterbüme=0
```

```
Anzahl_Zylinderlaufbuchsen=4
```

```
Anzahl_Kurbelgehäuse=9
```

```
.....
Anzahl_Kolbenbolzen=1
```

```
DO (NAM=X, BGN=1, END=ges_anz, DLT=1)
```

! Wenn es gleicher Bauteil gibt, rechnet die gesamte Anzahl wegen gleichen Tasterbüme nicht

```
IFTHEN (VL1=X, VL2=2, TYP=EQ)
```

```
IFTHENS (ST1=~Bauteil(2), ST2=~Bauteil(1), TYP=EQ)
```

```
UNWIND (FRA=2, LAB=10)
```

```
ENDIF
```

```
ELSEIF (VL1=X, VL2=3, TYP=EQ)
```

```
If (BXP=CHS::~Bauteil(1) == CHS::~Bauteil(2) || CHS::~Bauteil(2)==CHS::~Bauteil(3) ||
```

```
CHS::~Bauteil(1) == CHS::~Bauteil(3))
```

```
UNWIND (FRA=2, LAB=10)
```

```
EndIf
```

```
ELSE
```

```
DO (NAM=I, BGN=4, END=ges_anz, DLT=1)
```

```
IFTHENS (ST1=~Bauteil(I), ST2=~Bauteil(1), TYP=EQ)
```

```
UNWIND (FRA=3, LAB=10)
```

```
ENDIF
```

```

IFTHENS (ST1=~Bauteil(I), ST2=~Bauteil(2), TYP=EQ)
  UNWIND (FRA=3, LAB=10)
ENDIF
IFTHENS (ST1=~Bauteil(I), ST2=~Bauteil(3), TYP=EQ)
  UNWIND (FRA=3, LAB=10)
ENDIF
ENDDO
ENDIF
IFTHENS (ST1=~Bauteil(X), ST2=Zylinderlaufbuchsen, TYP=EQ)
  gesamt_Tasterb äume=gesamt_Tasterb äume+Anzahl_Zylinderlaufbuchsen
ENDIF
IFTHENS (ST1=~Bauteil(X), ST2=Kolben, TYP=EQ)
  gesamt_Tasterb äume=gesamt_Tasterb äume+Anzahl_Kolben
ENDIF
.....
IFTHENS (ST1=~Bauteil(X), ST2=Kolbenbolzen, TYP=EQ)
  gesamt_Tasterb äume=gesamt_Tasterb äume+Anzahl_Kolbenbolzen
ENDIF
10:CONTIN
ENDDO
! Wenn Randbindung nicht erfüllt wird, springt es zurück zu Label 1, ausgewählte Bauteilen zu korrigieren
IFTHEN (VL1=gesamt_Tasterb äume, VL2=MAX, TYP=GT)
  ShowDialog (DLP=Zuordnung)
  UNWIND (FRA=2, LAB=1)
ENDIF
! Löschen alle in Prozedur "Zuordnung" erzeugte Variable
CLEANUP

```

- Dialog "Auswahlmen ü"

```

<?xml version="1.0" encoding="utf-16"?>
<Root>
  <!--Auswahlmen ü in LDB-->
  <Dialog QdsName="LDBDLP:Auswahlmen ü">
    <BackColor>Control</BackColor>
    <NativeFont Name="Microsoft Sans Serif" Size="8.25" Style="Regular" Unit="Point" />
    <ForeColor>ControlText</ForeColor>
    <Size>
      <Width>600</Width>
      <Height>837</Height>
    </Size>

    <!--Beispiel : Groupbox für Bauteil 8 -->
    <GroupBox DsgName="gbx10">
      <AutoSize>False</AutoSize>
      <BackColor>Control</BackColor>
      <NativeFont Name="Microsoft Sans Serif" Size="8.25" Style="Regular" Unit="Point" />
      <ForeColor>ControlText</ForeColor>
      <Text>
        <de>Bauteile 8</de>
      </Text>
      <Location>
        <X>50</X>
        <Y>680</Y>
      </Location>
      <Size>
        <Width>480</Width>
        <Height>65</Height>
      </Size>

```



```

<TabIndex>1</TabIndex>
<ComboBox DsgName="cbx10" InputPatternType="None">
  <AutoSize>False</AutoSize>
  <BackColor>Window</BackColor>
  <DropDownWidth>300</DropDownWidth>
  <NativeFont Name="Microsoft Sans Serif" Size="8.25" Style="Regular" Unit="Point" />
  <ForeColor>WindowText</ForeColor>
  <Location>
    <X>95</X>
    <Y>25</Y>
  </Location>
  <Size>
    <Width>300</Width>
    <Height>21</Height>
  </Size>
  <TabIndex>0</TabIndex>
  <TabStop>True</TabStop>
  <!--Texteingabefelder verbindet mit $I-->
  <BindTo Dsc="$I" UpdImm="Y" Brackets="N">
    <ItemList>
      <!--Element mit nur Attribute -->
      <Item Value="Zylinderlaufbuchsen" />
      <Item Value="Kolben" />
      .....
      <Item Value="Nowerahmen_MDEG" />
      <Item Value="Kolbenbolzen" />
    </ItemList>
  </BindTo>
</ComboBox>
</GroupBox>
.....
<!--Beispiel : Groupbox für Maschinentyp -->
<GroupBox DsgName="gbx5">
  <AutoSize>False</AutoSize>
  <BackColor>Control</BackColor>
  <NativeFont Name="Microsoft Sans Serif" Size="8.25" Style="Regular" Unit="Point" />
  <ForeColor>ControlText</ForeColor>
  <Text>
    <de>Maschine</de>
  </Text>
  <Location>
    <X>305</X>
    <Y>125</Y>
  </Location>
  <Size>
    <Width>190</Width>
    <Height>50</Height>
  </Size>
  <TabIndex>3</TabIndex>
  <ComboBox DsgName="cbx5" InputPatternType="None">
    <AutoSize>False</AutoSize>
    <BackColor>Khaki</BackColor>
    <DropDownWidth>121</DropDownWidth>
    <NativeFont Name="Microsoft Sans Serif" Size="8.25" Style="Regular" Unit="Point" />
    <ForeColor>WindowText</ForeColor>
    <Location>
      <X>35</X>
      <Y>15</Y>
    </Location>
    <Size>
      <Width>121</Width>

```

```

    <Height>21</Height>
  </Size>
  <TabIndex>0</TabIndex>
  <TabStop>True</TabStop>
  <!--Texteingabefelder verbindet mit $D-->
  <BindTo Dsc="$D" UpdImm="Y" Brackets="N">
    <ItemList Pairs="True">
      <!--Element mit nur Attribute und Text-->
      <Item Value="416">MA416</Item>
      <Item Value="321">MA321</Item>
      <Item Value="125">MA125</Item>
      <Item Value="393">MA393</Item>
    </ItemList>
  </BindTo>
</ComboBox>
</GroupBox>
.....
</Dialog>
</Root>

```

Alle Programme sind hier nicht komplett.

A 2: Messprogramme für die Eingabe der Bauteilskenndaten

- Haupt Messprogramm

! Kenndaten Eingeben für jeden Bauteil

INDPRC (NAM=Daten_Eingabe)

! Prozedur "Daten_Eingabe" aufrufen

DELELE (NAM=XML:Bauteil, CNF=N)

! XML Daten löschen

- Prozedur "Daten_Eingabe"

! historische Variable "Daten" erstellen

DELETE (NAM=GDBHRC:Daten, CNF=N, TYP=HRC)

! Felder und ihre Länge von historischer Variable "Daten" definieren

DFNHRC (NAM=GDBHRC:Daten, MOD=NEW,

FLD=(A,B,C,D,E,F,G,H,I,J,K,L,M,N,O,P,Q,R,S,T,U,V,W,X,Y,Z,a,b,c,d,e,f,g,h,i),

FLG=(20,20,20,20,20,20,20,20,20,20,.....,20,20,20,20,20,20,20,20,20,20,20))

! leere Objekte für historische Variable "Daten" erzeugen, Anzahl der Objekte ist Variable "ges_anz"

CRSUBTYP (NAM=GDBHRC:Daten, LIN=ges_anz, TYP=HRC, STY=HIS)

! Bauteil von 1 bis gesamte Anzahl der Bauteilen

DO (NAM=X, BGN=1, END=ges_anz, DLT=1)

! zeigen welche aktuellen Bauteilskenndaten, einzugeben

CVREACHS (NAM=~X, VAL=X, FM2=0, INT=N, ANG=N, SPZ=N, RLS=Y, RTZ=Y)

CONCAT (NAM=~KKALI, STR=(Kenndaten Eingabe für Bauteil,' ',~X,' ',(~Bauteil(X),) !), LMG=0,

RLS=N, INI=Y)

CVSTRTXT (TXT=Ba, LIN=1, STR=")

CVSTRTXT (TXT=Ba, LIN=2, STR=~KKALI)

CVSTRTXT (TXT=Ba, LIN=3, STR='')

ShowPicAndText (TXT=Ba, STL=Normal, UNI=mm)

WAIT (SEC=1.5)

! Kontrolle Variable "Übernahme" für Kenndaten übernahme mit "NEIN" eingesetzt wird

~Übernahm=NEIN

! Beispiel Kenndateneingabe für Bauteil "Zahnrad"

IFTHENS (ST1=~Bauteil(X), ST2=Zahnrad, TYP=EQ)

! initialisieren alle Variablen

If (BXP=REA:X==1 || CHS::~Übernahm=='NEIN')

DELETE (NAM=Lgd:ZAHNRAD\$NR, CNF=N)

EDIT (NAM=Lgd:ZAHNRAD\$NR, CRE=Y)

EndIf

~EIN\$AUS(X)=Einabu

.....

~FQDA(X)=''

PUTVALS (OBJ=ZAHNRAD\$NR, TYP=Lgd, CDS=(D,N,P,R), STR=(~EIN\$AUS(X),

~AUFSPANNUNG(X), ~ZEIT\$KM(X), ~EXIT(X)))

! Eingabedialog für Zahnrad aufrufen, Label ist 9

9:ShowDialog (DLP=ZAHNRAD\$NR, OBJ=Lgd:ZAHNRAD\$NR, DID=1225)

! Variable "springen" ist in Prozedur "Daten_Übernahm" definieren.

! Die kontrolliert, ob Bauteilskenndaten überzunehmen

IFTHENS (ST1=~springen, ST2=JA, TYP=EQ)

~springen=''

UNWIND (FRA=1, LAB=9)

ENDIF

! bekommen alle Bauteilskenndaten von Dialog

GETVALS (OBJ=ZAHNRAD\$NR, TYP=Lgd,

CDS=(A,B,C,D,E,F,G,H,I,J,K,L,M,N,O,P,Q,R,T,a,b,c,d,e,f,g),

CHS=(~BEZEICHNUNG(X), ~ZEICH\$NR(X), , ~Abgasnorm(X), ~Prüfprogramm(X)))

! Prüfen, ob alle Variable richtig sind

```

IFTHENS (ST1=~MOTORTYP(X), ST2=' ', TYP=EQ)
  UNWIND (FRA=1, LAB=8)
ENDIF
If (BXP=CHS:~BEZEICHNUNG(X)=='KW' || CHS:~BEZEICHNUNG(X)=='NW' || ..... )
  UNWIND (FRA=1)
Else
  DELTXT (NAM=BEZ, CNF=N)
  CVSTRTXT (TXT=BEZ, LIN=1, STR=")
  CVSTRTXT (TXT=BEZ, LIN=2, STR=Prüfen Sie bitte die Eingabe von der Bezeichnung!)
  CVSTRTXT (TXT=BEZ, LIN=3, STR=Bezeichnung muß aus POPUP ausgewählt werden!)
  ShowPicAndText (TXT=BEZ, STL=Normal, WAI=Y, UNI=mm)
  UNWIND (FRA=1, LAB=9)
EndIf
.....
! Append HRC to Menu program
! Speichern zum aktuellen Messprogramm
SAVE (OBJ=HRC:BESTUECKEN, STA=APN, SCB=N, FIL=CNC:ZAHNRAD.WDB)
! Speichern zum Multimesprogramm
SAVE (OBJ=HRC:BESTUECKEN, STA=APN, SCB=N, FIL=Q:\QDA 8\CNC
      QDA8\Multi\Auswahlmenü.WDB)
SAVE (OBJ=HRC:BESTUECKEN, STA=APN, SCB=N, FIL=Q:\QDA 8\CNC
      QDA8\Multi\manuelle_Zahnrad.wdb)
ENDIF
! Speichern alle Kenndaten in GDBHRC, jede Bauteilskenndaten als eine Objekt in HRC zu speichern
! Bauteilskenndaten für "Pleuelstange"
IFTHENS (ST1=~Bauteil(X), ST2=Pleuel, TYP=EQ)
  PUTVALS (OBJ=GDBHRC:Daten.HIS(X), TYP=HRC,
           CDS=(A,B,C,D,E,F,G,H,I,J,K,L,M,N,O,P,Q,R,S,T,U,V,W,X,Y,Z,a,b,c,d,e,f,g,h),
           STR=(~DATE(X), ....., ~NORMAL(X), ~MGZ(X)))
ENDIF
! Bauteilskenndaten für "Zahnrad" Standardmessung
IFTHENS (ST1=~Bauteil(X), ST2=Zahnrad, TYP=EQ)
  PUTVALS (OBJ=GDBHRC:Daten.HIS(X), TYP=HRC,
           CDS=(A,B,C,D,E,F,G,H,I,J,K,L,M,N,O,P,Q,R,S,T,U,V,W,X,Y,Z),
           STR=(~BEZEICHNUNG(X), ~ZEICH$NR(X), ....., ~MOTORTYP(X)))
ENDIF
.....
! Xml Dateil für POPUP erstellen
IFTHEN (VL1=X, VL2=1, TYP=EQ)
  CONCAT (NAM=~Bauteil, STR=(1.,',',~Bauteil(X)), LMG=0, INI=Y)
  CVSTRTXT (TXT=POPUP, LIN=1, STR=~Bauteil)
  CnvTxtToItemList (TXT=POPUP, XML=Bauteil)
ELSEIF (VL1=X, VL2=2, TYP=EQ)
  CONCAT (NAM=~Bauteil, STR=(2.,',',~Bauteil(X)), LMG=0, INI=Y)
  CVSTRTXT (TXT=POPUP, LIN=2, STR=~Bauteil)
  CnvTxtToItemList (TXT=POPUP, XML=Bauteil)
ELSEIF (VL1=X, VL2=3, TYP=EQ)
  CONCAT (NAM=~Bauteil, STR=(3.,',',~Bauteil(X)), LMG=0, INI=Y)
  CVSTRTXT (TXT=POPUP, LIN=3, STR=~Bauteil)
  CnvTxtToItemList (TXT=POPUP, XML=Bauteil)
.....
ENDIF
ENDDO
! Löschen Xml Datei "Bauteil"
DELELE (NAM=XML:Bauteil, CNF=N)
! Löschen alle in Prozedur erzeugten Variablen
CLEANUP

```

- "OK" Button Funktion von Dialog Kenndaten übernahme

! "X" ist der Index von Bauteilreihe, hier Beispiel von Zahnrad Standardmessung
! Erste Bauteil gibt es keine Übernahmesmöglichkeit

```
IFTHEN      (VL1=X, VL2=1, TYP=EQ)
  CVSTRTXT  (TXT=HH, LIN=1, STR=")
  CVSTRTXT  (TXT=HH, LIN=2, STR=Es ist der ersten Bauteil !)
  CVSTRTXT  (TXT=HH, LIN=3, STR=")
  ShowPicAndText (TXT=HH, STL=Normal, WAI=Y, UNI=mm)
  ReturnToDialog (DID=1225)
ELSE
  ! Bekommen die ausgewählte Bauteilskenndaten, die gleich Motor ist
  GETVALS   (OBJ=ZAHNRAD$NR, TYP=Lgd, CDS=1, CHS=~BB)
  INDFUNC   (NAM=Daten_übernahm)           ! Prozedur "Daten_Übernahm" aufrufen
  ~springen=JA           ! Variable "springer" als "JA" deklarieren
ENDIF
```

- Prozedur "Daten_Übernahm"

! Daten übernehmen von vorherigem Bauteil

```
DELCHS     (NAM=~BBB, CNF=N, TYP=CHS)
! Prüfen, ob richtig Bauteil ausgewählt wird
SUBSTR     (NAM=~BBB, RSD=~BB, LM2=1, RM2=1)
IFTHENS    (ST1=~BBB, ST2=' ', TYP=EQ)
  CVSTRTXT  (TXT=HH, LIN=1, STR=")
  CVSTRTXT  (TXT=HH, LIN=2, STR=Bitte einen Bauteil auswählen !)
  CVSTRTXT  (TXT=HH, LIN=3, STR=' ')
  ShowPicAndText (TXT=HH, STL=Normal, WAI=Y, UNI=mm)
  ~springen=JA
  UNWIND   (FRA=2)
ENDIF
```

! Variable "BBB" die Bauteilsindex ist, und umwandeln von String zu Integer

```
CVSTRREA   (NAM=T, STR=~BBB)
IFTHEN     (VL1=T, VL2=X, TYP=EQ)
  CVSTRTXT  (TXT=HH, LIN=1, STR=")
  CVSTRTXT  (TXT=HH, LIN=2, STR=Der aktuellen Bauteil ist 2. !)
  CVSTRTXT  (TXT=HH, LIN=3, STR=Es kann die Kenndaten nur von Bauteil 1 übernehmen!)
  ShowPicAndText (TXT=HH, STL=Normal, WAI=Y, UNI=mm)
  ~springen=JA
  UNWIND   (FRA=2)
ENDIF
```

! Kenndaten Übernehmen von ausgewähltem Bauteil

```
IFTHENS    (ST1=~Bauteil(T), ST2=Zylinderlaufbuchsen, TYP=EQ)
  GETVALS   (OBJ=GDBHRC:Daten.HIS(T), TYP=HRC, CDS=(A,B,C,D,N,P),
             CHS=(~DATE_Temp,~CMM_Temp, ..... ,~Prüfer_Temp))
ENDIF
```

```
IFTHENS    (ST1=~Bauteil(T), ST2=Lagerzapfen, TYP=EQ)
  GETVALS   (OBJ=GDBHRC:Daten.HIS(T), TYP=HRC, CDS=(A,B,C,G),
             CHS=(~MOTTYP_Temp,~MOTNR_Temp,~MSO_Temp,~Prüfer_Temp))
ENDIF
```

~Abteilung(X)=TP/PPH

! Kenndatenszuweisung für aktuellen Bauteil

```
IFTHENS    (ST1=~Bauteil(X), ST2=Zylinderlaufbuchsen, TYP=EQ)
  DELETE    (NAM=Lgd:Buchse, CNF=N)
  EDIT      (NAM=Lgd:Buchse, CRE=Y)
  PUTVALS   (OBJ=Lgd:Buchse, CDS=(M,B,T), STR=(0,~CMM,TP/PPH))
  PUTVALS   (OBJ=Buchse, TYP=Lgd, CDS=(A,B,C,D,P,N),
```

```

PUTVALS STR=(~DATE_Temp,~CMM_Temp,~MOTTYP_Temp, ..... , MSO_Temp))
(OBJ=Buchse, TYP=Lgd, CDS=(Q,a,b,c,d,e,f),
STR=(~Verwendung(X),~Zweck(X),~Ölsorte(X), ..... ,~Prüfprogramm(X)))
ENDIF
.....
IFTHENS (ST1=~Bauteil(X), ST2=Zahnrad, TYP=EQ)
DELETE (NAM=Lgd:ZAHNRAD$NR, CNF=N)
EDIT (NAM=Lgd:ZAHNRAD$NR, CRE=Y)
PUTVALS (OBJ=ZAHNRAD$NR, TYP=Lgd, CDS=(F,G,I,T,H),
STR=(~Prüfer_Temp,~MOTNR_Temp,~DATE_Temp,~MOTTYP_Temp,~MSO_Temp))
PUTVALS (OBJ=ZAHNRAD$NR, TYP=Lgd, CDS=(a,b,c,d,e,f,g),
STR=(~Verwendung(X),~Zweck(X), ..... , ~Prüfprogramm(X)))
ENDIF

```

Alle Programme sind hier nicht komplett.

A 3: Ein Beispielprogramme für Taster Einmessen

- Haupt Messprogramm

! jedes Tastereinmessenprogramm aufrufen nach ihre reihfolge

```
DO (NAM=X, BGN=1, END=ges_anz, DLT=1)
  ! zeigen Tastereinmessen für welchen Bauteilen
  CVREACHS (NAM=~X, VAL=X, FM2=0, INT=N, ANG=N, SPZ=N, RLS=Y, RTZ=Y)
  CONCAT (NAM=~KKALI, STR=(Kalibrierung für Bauteil,' ',~X,' ',(~Bauteil(X),) !), LMG=0, RLS=N,
    INI=Y)
  CVSTRTXT (TXT=Ba, LIN=1, STR=")
  CVSTRTXT (TXT=Ba, LIN=2, STR=~KKALI)
  CVSTRTXT (TXT=Ba, LIN=3, STR=' ')
  ShowPicAndText (TXT=Ba, STL=Normal, WAI=Y, UNI=mm)
  ! Jedes Teilprogramm aufrufen
  IFTHENS (ST1=~Bauteil(X), ST2=Zylinderlaufbuchsen, TYP=EQ)
    CALL (FIL=Q:\QDA 8\CNC QDA8\Multi\Kali_Buchsen.wdb, RPO=N)
  ENDIF
  .....
  IFTHENS (ST1=~Bauteil(X), ST2=Pleuel, TYP=EQ)
    CALL (FIL=Q:\QDA 8\CNC QDA8\Multi\Kali_Pleuel.wdb, RPO=N)
  ENDIF
ENDDO
```

- Taster Einmessen für Pleuelstange (Beispiel)

! Kalibrierung für Pleuel

~MA=GDBCHS:~MA ! Messmaschinentype von GDB bekommen

CONCAT (NAM=~CMM, STR=(MA,~MA), LMG=0, RLS=Y, INI=Y)

! Messbauteilsdaten von GDBHRC bekommen

```
GETVALS (OBJ=GDBHRC:Daten.HIS(X), TYP=HRC,
  CDS=(A,B,C,D,E,F,G,H,I,J,K,L,M,N,O,P,Q,R,S,T,U,V,W,X,Y,Z,a,b,c,d,e,f,g,h),
  CHS=(~DATE, ..... , ~PALCSY,~KALI,~NORMAL,~MGZ))
```

! Im Tastkopf befindlichen Taster angeben

```
DELUVS (NAM=UVS:M_?, CNF=N)
DFNUVS (NAM=M_1, VAL=(' ','N,5,-120), DSC=(A,E,C,a,b))
ShowDialog (DLP=PLEUEL, OBJ=UVS:M_1, DID=1)
```

! Einmesskugel definieren

```
DfnArtefact (NAM=Artefact, DIA=Normal_O, SEL=45.0, SDM=10.0)
QualifyTool (NAM=PRB(1), DIA=TDM, NRF=Y, REF=Artefact, SCN=Y, DFT=3, MGZ=~MGZ,
  SNT=TRX, RPT=(0,0,ZAB), DEL=Y, GEO=SPH, UAD=N)
```

! Referenztaster einmessen

```
MECIR (NAM=Einzel_O, CSY=Artefact, MOD=NOE)
BLDCSY (NAM=Ring, TYP=CAR, SPA=Ring_Pla, SDR=+Z, XZE=Ring_Cir, YZE=Ring_Cir,
  ZZE=Ring_Pla)
```

! auf jeweiliges Tasterdatenfile abspeichern

```
IFTHENS (ST1=~CMM, ST2=MA393, TYP=EQ)
  SAVE (OBJ=(EDBPRB:PRB*()),ELE:PRB(), STA=APN, SCB=N,
    FIL=CNC:TASTER_PLEUEL_MA393.PRB)
```

```
ELSEIFS (ST1=~CMM, ST2=MA125, TYP=EQ)
  SAVE (OBJ=(EDBPRB:PRB*()),ELE:PRB(), STA=APN, SCB=N,
    FIL=CNC:TASTER_PLEUEL_MA125.PRB)
```

```

ENDIF
! Wenn es nicht neue Kalirierung braucht, lesen die Daten ein
IFTHENS      (ST1=~KALI, ST2=N, TYP=EQ)
! alte Tasterdaten verwenden
IFTHENS      (ST1=~CMM, ST2=MA393, TYP=EQ)
  LOAD       (OBJ=(PRB:PRB*()),ELE:PRB()), QUE=(CHS:~PRBDAT,~PRBTIM), STA=APN, LCB=N,
             FIL=CNC:TASTER_PLEUEL_MA393.PRB)
ELSEIFS      (ST1=~CMM, ST2=MA125, TYP=EQ)
.....
ENDIF
! upload Tasterdaten in EDB und deckt die Tasterdaten in LDB ab
EDBUPD      (OBJ=PRB:PRB(1))
LISPRB      (NAM=EDBPRB:PRB(1))
ENDIF
! Tasterdaten speichern
IFTHENS      (ST1=~PALCSY, ST2=Y, TYP=EQ)
IFTHENS      (ST1=~CMM, ST2=MA393, TYP=EQ)
  SAVE       (OBJ=CSY:P_CSY, STA=APN, SCB=N, FIL=CNC:TASTER_PLEUEL_MA393.PRB)
ELSEIFS      (ST1=~CMM, ST2=MA125, TYP=EQ)
  SAVE       (OBJ=CSY:P_CSY, STA=APN, SCB=N, FIL=CNC:TASTER_PLEUEL_MA125.PRB)
ELSEIFS      (ST1=~CMM, ST2=MA321, TYP=EQ)
  SAVE       (OBJ=CSY:P_CSY, STA=APN, SCB=N, FIL=CNC:TASTER_PLEUEL_MA321.PRB)
ENDIF
ENDIF
! Freifahren
MOVCM      (TYP=DLT, DST=(0,0,300))
PUTPRB      ! Taster zurück in Ablage lassen (notwendig bei Multimesprogramm)

```

Alle Programme sind hier nicht komplett.

A 4: Ein Beispielprogramme für Manuelle Ausrichtung

- Haupt Messprogramm

! jedes Teilprogramm aufrufen nach ihre reihfolge

```
DO (NAM=X, BGN=1, END=ges_anz, DLT=1)
  ! zeigen Teilprogramm für welchen Bauteilen
  GDBREA:X=X
  CVREACHS (NAM=~X, VAL=X, FM2=0, INT=N, ANG=N, SPZ=N, RLS=Y, RTZ=Y)
  CONCAT (NAM=~MMANU, STR=(Manuelle Ausrichtung oder Messaufgabe Auswahl für,';~X,'
    ',(~Bauteil(X),!), LMG=0, RLS=N, INI=Y)
  CVSTRTXT (TXT=Ba, LIN=1, STR=")
  CVSTRTXT (TXT=Ba, LIN=2, STR=~MMANU)
  CVSTRTXT (TXT=Ba, LIN=3, STR=' ')
  ShowPicAndText (TXT=Ba, STL=Normal, WAI=Y, UNI=mm)
  ! Teilprogramm aufrufen
  IFTHENS (ST1=~Bauteil(X), ST2=Zylinderlaufbuchsen, TYP=EQ)
    CONCAT (NAM=~LOAD, STR=(~Pfad,manuelle_Buchsen.wdb), LMG=0, RLS=Y, INI=Y)
    CALL (FIL=~LOAD, RPO=N)
  ENDIF
  .....
  IFTHENS (ST1=~Bauteil(X), ST2=Lagerschale von Nockenwelle, TYP=EQ)
    CONCAT (NAM=~LOAD, STR=(~Pfad,manuelle_Lagerschale_N.wdb), LMG=0, RLS=Y, INI=Y)
    CALL (FIL=~LOAD, RPO=N)
  ENDIF
ENDDO
```

- Beispielprogramm für Kurbelwelle

! manuelle Ausrichtung oder Messaufgabe Eingabe für Kurbelwelle

! Messger ätyp und Bauteilindex lesen von GDB

~CMMTYP=GDBCHS:~MA

X=GDBREA:X

! Messkenndaten von GDB lesen

~BEZEICHNUNG=KURBELWELLE

```
GETVALS (OBJ=GDBHRC:Daten.HIS(X), TYP=HRC, CDS=(A,B,C,D,E, ... ,T,U,V,W,X,Y),
  CHS=(~DATE,~CMMTYP,~MOTTYP,~MOTNR,~ZEICHNR,~ID, ... ,~Verschleiss))
```

! Dialog für Merkmalsnummer zeigen, und korrigieren

222:DELETE (NAM=Lgd:MKP, CNF=N)

EDIT (NAM=Lgd:MKP, CRE=Y)

ShowDialog (DLP=MKP, OBJ=Lgd:MKP)

! Messungspositionen bekommen

```
GETVALS (OBJ=Lgd:MKP, TYP=Lgd, CDS=(A,B,C,D,E,F,G,H,I,J,K,L,M,N,O,P,Q,R,S,T,U,V,W,X),
  CHS=(~MERKMAL$NR(1),~MERKMAL$NR(2), ... ,~MERKMAL$NR(24)))
```

! Speichern von LDB zu GDB

```
CPYOBJ (FRM=MKP, TO =MKP_Kurbelwelle(X), TYP=Lgd, SRC=LDB, DST=GDB)
```

! Dialog Merkmal zeigen, und korrigieren

ShowDialog (DLP=~MKM, OBJ=Lgd:Merkmal, DID=3)

IFTHENS (ST1=~MKM, ST2=L4, TYP=EQ)

```
  GETVALS (OBJ=Merkmal, TYP=Lgd, CDS=(A,B,C,D,E,a,b,c,d),
```

```
    CHS=(~LAGER$NR(1),~LAGER$NR(2), ... , ~PLEUEL$NR(4)))
```

ELSEIFS (ST1=~MKM, ST2=L6, TYP=EQ)

.....
ENDIF

! Speichern von LDB zu GDB

```

CPYOBJ      (FRM=Merkmal, TO =Merkmal_Kurbelwelle(X), TYP=Lgd, SRC=LDB, DST=GDB)
.....
GDBCHS:~LAGER$ORG=~LAGER$AUFTR
GDBCHS:~PLEUEL$ORG=~PLEUEL$AUFTR
! Tastdaten Laden
TESTSTR     (NAM=TST$VAL, ST1=~CMMTYP, ST2=125, TYP=IDX)
IFTHEN      (VL1=TST$VAL, VL2=0, TYP=GT)
LOAD        (OBJ=(REA:*(),PRB:*()), STA=APN, LCB=N,
            FIL=CNC:TASTER_KURBELWELLE_MA125.PRB)
ENDIF
.....
! Updaten EDB
EDBUPD      (OBJ=PRB:PRB())
DELPRBS
DELPRB      (NAM=PRB:PRB(), CNF=N)
! Measure fixture
HELP        (NAM=ICN:FIXTURE)
! Manuelle Ausrichtung
USEPRB      (NAM=PRB(13))
CVSTRTXT    (TXT=PNT1, LIN=1, STR='Messe Z-Null(Anlauffläche am Paßlager in +X-Richtung)')
CVSTRTXT    (TXT=PNT1, LIN=2, STR='mit 1 CLP and 1 PRB. Aktiver Taster zeigt in +Y Richtung')
MEPNTX      (NAM=PNT1, CSY=REFR$CSY, MOD=NOE, DEL=YY)
USEPRB      (NAM=PRB(11))
.....
DIPNTPNT    (NAM=AXI2, CSY=REFR$CSY, EL1=AXI1, EL2=CIR3, MOD=NOE)
! Koordinatensystem erzeugen
BLDCSY      (NAM=PAL$CSY, SPA=AXI1, SDR=+Z, PLA=AXI2, PDR=+X, XZE=AXI1, YZE=AXI1,
            ZZE=PNT1)
! Speichern relativ Koordinatensystem von LDB zu GDB
CPYOBJ      (FRM=PAL$CSY, TO =PAL$CSY_Kurbelwelle(X), TYP=CSY, SRC=LDB, DST=GDB)
! Taster in Tastablage lassen
MOVCMM      (TYP=DLT, DST=(0,0,400))
PUTPRB

```

- Beispielprogramm für Nockenwelle

```

! automatische oder Wiederholungs- Messung für Nockenwelle
! Index X und Wiederholungsschlüssel von GDB lesen
X=GDBREA:X
~WB(X)=GDBCHS:~WB(X)
~CMMTYP=GDBCHS:~MA
! Wenn neue Messung, löschen alle unnötige Objekte
IFTHENS     (ST1=~WB(X), ST2=JA, TYP=NE)
  DELCHS    (NAM=CHS:*(), CNF=N)
  DELETE    (NAM=(HRC:*(),UVS:*(),REA:*(),MSG:*(),CMM:*(),ELE:*()), CNF=N)
ENDIF
! für Wiederholungsmessung, Messpunkte neue definieren
IFTHENS     (ST1=~WB(X), ST2=JA, TYP=EQ)
  DELETE    (NAM=Lgd:MKP, CNF=N)
  EDIT      (NAM=Lgd:MKP, CRE=Y)
  ShowDialog (DLP=MKP, OBJ=Lgd:MKP)
  GETVALS   (OBJ=UVS:MKP, CDS=(A,B,C,D,E,F,G,H,I,J,K,L),
            CHS=(~MERKMAL$NR(1),~MERKMAL$NR(2),~Sensor,~Planlauf))
! springen zu Label 330, anfang zu messen
UNWIND      (FRA=1, LAB=330)
CUTDBS     (DBS=UDB)
BOOT        (FIL=NOCKENWELLE:Q7Camshaft_de.ZDB, DBS=ZDB)

```

```

ENDIF
! Messungsumgebung definieren
NOBELL
! Cut UDB
CUTDBS      (DBS=UDB)
! Boot Gessler ZDB
BOOT        (FIL=NOCKENWELLE:Q7Camshaft_de.ZDB, DBS=ZDB)
GDBCHS:~ABTEILUNG=TP/PPH

! Alle nötige Daten von GDB einlesen
CPYOBJ      (FRM=MainDialog(X), TO =MainDialog, TYP=UVS, SRC=GDB, DST=LDB)
GETVALS     (OBJ=UVS:MainDialog, CDS=(A,B,C,D,E,F,G),
            CHS=(~MotorType,~MotorCylinders,~CamTypes,~N_SL(1),~N_SL(2),~VEL$,~DEN$))
CONCAT      (NAM=GDBCHS:~NW$TYP, STR=(~MotorType,~MotorCylinders), RLS=Y, INI=Y)
~NW$TYP=GDBCHS:~NW$TYP
GETVALS     (OBJ=GDBHRC:Daten.HIS(X), TYP=HRC, RDS=a, REA=StopDia,
            CDS=(A,B,C,D,E,F,G,H,I,J,K,L,M,N,O,P,Q,R,S,T,U,V,W,X,Y,Z,a),
            CHS=(~BEZEICHNUNG,~IDENT$NR,~ZEICH$NR, ... ,~ZGS))
GDBCHS:~ZEICH$NR=~ZEICH$NR
CPYOBJ      (FRM=NOCKENWELLE(X), TO =NOCKENWELLE, TYP=UVS, SRC=GDB, DST=LDB)
CPYOBJ      (FRM=MKP_Nockenwelle(X), TO =MKP, TYP=UVS, SRC=GDB, DST=LDB)
GETVALS     (OBJ=UVS:MKP, CDS=(A,B,C,D,E,F,G,H,I,J,K,L),
            CHS=(~MERKMAL$NR(1),~MERKMAL$NR(2), ... ,~Sensor,~Planlauf))
CPYOBJ      (FRM=MerkmalAuswahl_Nockenwelle(X), TO =MerkmalAuswahl, TYP=UVS, SRC=GDB,
            DST=LDB)
GETVALS     (OBJ=UVS:MerkmalAuswahl, CDS=(A,B), CHS=(~INDEX,~GEAR$YN))
! Label für Wiederholungsmessung
330:CONTIN
IFTHENS     (ST1=~WB(X), ST2=JA, TYP=EQ)
.....
IFTHENS     (ST1=~NW$TYP, ST2=MDEGL4, TYP=EQ)
IFTHENS     (ST1=~CamTypes, ST2=Inlet, TYP=EQ)
  CPYOBJ     (FRM=MDEGL4Einlass, TO =MerkmalAuswahl, TYP=DLP, CNF=N)
ELSEIFS     (ST1=~CamTypes, ST2=Exhaust, TYP=EQ)
  CPYOBJ     (FRM=MDEGL4Auslass, TO =MerkmalAuswahl, TYP=DLP, CNF=N)
ELSEIFS     (ST1=~CamTypes, ST2=ExhaustBrake, TYP=EQ)
  CPYOBJ     (FRM=MDEGL4AuslassBre, TO =MerkmalAuswahl, TYP=DLP, CNF=N)
ENDIF
.....
ENDIF
.....
IFTHENS     (ST1=~WB(X), ST2=JA, TYP=EQ)
  ShowDialog (DLP=MerkmalAuswahl, OBJ=UVS:MerkmalAuswahl)
  GETVALS    (OBJ=UVS:MerkmalAuswahl, CDS=(A,B), CHS=(~INDEX,~GEAR$YN))
ENDIF
.....
! Extend UVS for each motor
IFTHENS     (ST1=~WB(X), ST2=JA, TYP=EQ)
  DFNUVS    (NAM=NOCKENWELLE, VAL=(~INDEX, ~GEAR$YN, ~LAGER$AUFTR,
    ~NOCKEN$AUFTR, ~ABSS$AUFTR,~NUT), DSC=(T, U,V,W,X,Y))
ENDIF
CPYOBJ      (FRM=PAL$CSY_Nockenwelle(X), TO =PAL$CSY, TYP=CSY, SRC=GDB, DST=LDB)
USEPRB      (NAM=PRB(11))
!
USECSY      (NAM=CSY)
! Messungsabschnitt
.....
! Nach der Messung, sortieren alle Messdaten und springen zurück zu Hauptmessprogramm
IFTHENS     (ST1=~WB(X), ST2=JA, TYP=EQ)
  DFNQUE     (NAM=$RPO, MBR=GDBQUE:$RPO(X), DEL=Y)
  CPYQUE     (FRM=(CHS:~LAGER$AUFTR(X),~NOCKEN$AUFTR(X),~ABSS$AUFTR(X)),

```

```

                TO = (CHS:~LAGER$AUFTR,~NOCKEN$AUFTR,~ABS$AUFTR), SRC=GDB, DST=LDB)
ENDIF
! Messdaten in Datenbank zu speichern
INDPRC      (NAM=END)
INDPRC      (NAM=Speich_QDA8)
CPYOBJ      (FRM=LAGER, TO =LAGER(X), TYP=HRC, DST=GDB)
DFNQUE      (NAM=GDBQUE:$RPO(X), MBR=$RPO, DEL=Y)
CPYQUE      (FRM=(CHS:~LAGER$AUFTR,~NOCKEN$AUFTR,~ABS$AUFTR),
            TO =(CHS:~LAGER$AUFTR(X),~NOCKEN$AUFTR(X),~ABS$AUFTR(X)), DST=GDB)

! Speichern ganz Programm als temporäre Programm. Das wird erst gelöscht, wenn Hauptprogramm beendet
~Pfad=GDBCHS:~Pfad
X=GDBREA:X
CVREACHS    (NAM=~X, VAL=X, FM2=0, INT=N, ANG=N, SPZ=N, RLS=Y, RTZ=Y)
~MA=GDBCHS:~MA
CONCAT      (NAM=~Ssave, STR=(~Pfad,Bauteil(~X,)_~MA,.wdb), LMG=0, RLS=Y, INI=Y)
SAVE        (FIL=~Ssave)
BOOT        (FIL=Q:\Quindos\Q7_Lib\UDB\Quindos7.UDB, DBS=UDB)

```

A 5: Ein Beispielprogramme für Wiederholungsmessung

- Haupt Messprogramm

```
! Wiederhole Messung
2:CONTIN                                ! Zykluslabel
! Wiederholungsmessung Schlüssel für jeden Bauteil
~WB(1)=NEIN
~WB(2)=NEIN
~WB(3)=NEIN
~Ende=NEIN                               ! Programm Ende Schlüssel
~WB(4)=NEIN
~WB(5)=NEIN
~WB(6)=NEIN
~WB(7)=NEIN
~WB(8)=NEIN
! nicht gemessene Bauteil definieren
DO      (NAM=I, BGN=ges_anz+1, END=8, DLT=1)
  ~Bauteil(I)=kein Bauteil
ENDDO
DELETE  (NAM=Lgd:wieder, CNF=N)
EDIT    (NAM=Lgd:wieder, CRE=Y)
PUTVALS (OBJ=wieder, TYP=Lgd, CDS=(E,F,G,A,B,C,D,H,I,J,K,L,M,N,O,P,Q),
        STR=(~Bauteil(1),~Bauteil(2),~Bauteil(3),NEIN,NEIN,NEIN,NEIN,~Bauteil(4))
ShowDialog (DLP=Widerhole, OBJ=Lgd:wieder)
GETVALS (OBJ=wieder, TYP=Lgd, CDS=(A, B, C, D, M, N, O, P, Q),
        CHS=(~WB(1),~WB(2),~WB(3),~Ende,~WB(4),~WB(5),~WB(6),~WB(7),~WB(8)))
! Wiederhole Messung für Bauteil 1
IFTHENS (ST1=~WB(1), ST2=JA, TYP=EQ)
  X=1
  GDBREA:X=X
  CVREACHS (NAM=~X, VAL=X, FM2=0, INT=N, ANG=N, SPZ=N, RLS=Y, RTZ=Y)
  CONCAT   (NAM=~MMANU, STR=(Wiederholungsmessung für, ',~X,',(,~Bauteil(X),!)), LMG=0,
          RLS=N, INI=Y)
  CVSTRTXT (TXT=Ba, LIN=1, STR=")
  CVSTRTXT (TXT=Ba, LIN=2, STR=~MMANU)
  CVSTRTXT (TXT=Ba, LIN=3, STR=' ')
  ShowPicAndText (TXT=Ba, STL=Normal, WAI=Y, UNI=mm)
  ! GDB Stringvariable definieren für Wiederholungsmessung, bei automatische Messung setzt es leer
  GDBCHS:~WB(1)=~WB(1)
  CONCAT   (NAM=~Lade, STR=(~Pfad,Bauteil(1),~MA,,wdb), LMG=0, RLS=Y, INI=Y)
  CALL     (FIL=~Lade, RPO=N)
ENDIF
! Wiederhole Messung für Bauteil 2
.....
! Programm Ende
IFTHENS (ST1=~Ende, ST2=JA, TYP=EQ)
  ! löschen alle Variable
  DELELE  (NAM=(GDBHRC:Daten,LDBHRC:*()), CNF=N, NOT=LDBHRC:BESTUECKEN)
  DELPRB  (NAM=(EDBPRB:PRB_Buchse(),EDBPRB:PRB_Lagerzapfen(1)), CNF=N)
  .....
  DELQUE  (NAM=GDBQUE:$RPO*(), CNF=N)
  CONCAT  (NAM=~DELFIL, STR=(~Pfad,*.,wdb), LMG=0, RLS=Y, INI=Y)
  PURGE   (FIL=~DELFIL)
  ! Programm Ende, springen zu Label 999999
  UNWIND  (FRA=1, LAB=999999)
ENDIF
! springen zu Label 2, weiter Wiederholungsmessung
```

```
GOTO          (LAB=2)
! Programm Ende
999999:CONTIN
BOOT         (FIL=UDB:Quindos7.UDB, DBS=UDB)
ShowDialog   (DLP=Q7, DID=10)
```

A 6: Ein Beispielprogramme für Koordinatensystem Benutzung

- Haupt Messprogramm

! XML Datei erzeugen, nach jeder Bauteilkenndaten Eingabe, liegt in LDB

! Hier X ist die eingesetzte Bauteilnummer

```
IFTHEN      (VL1=X, VL2=1, TYP=EQ)
  CONCAT    (NAM=~Bauteil, STR=(1,',' ,~Bauteil(X)), LMG=0, INI=Y)
  CVSTRTXT  (TXT=POPUP, LIN=1, STR=~Bauteil)
  CnvTxtToItemLst (TXT=POPUP, XML=Bauteil)
ELSEIF      (VL1=X, VL2=2, TYP=EQ)
  CONCAT    (NAM=~Bauteil, STR=(2,',' ,~Bauteil(X)), LMG=0, INI=Y)
  CVSTRTXT  (TXT=POPUP, LIN=2, STR=~Bauteil)
  CnvTxtToItemLst (TXT=POPUP, XML=Bauteil)
ELSEIF      (VL1=X, VL2=3, TYP=EQ)
  CONCAT    (NAM=~Bauteil, STR=(3,',' ,~Bauteil(X)), LMG=0, INI=Y)
  CVSTRTXT  (TXT=POPUP, LIN=3, STR=~Bauteil)
  CnvTxtToItemLst (TXT=POPUP, XML=Bauteil)
ELSEIF      (VL1=X, VL2=4, TYP=EQ)
  .....
ENDIF
```

! Nach der kompletten Messung, speichern die XML Datei in der GDB

```
DELELE      (NAM=GDBXML:Bauteil, CNF=N)
CPYOBJ      (FRM=Bauteil, TO =Bauteil, TYP=XML, SRC=LDB, DST=GDB)
DELELE      (NAM=LDBXML:Bauteil, CNF=N)
```

- Beispielprogramm für Pleuelstange

! von Dialog bekommen die Position variable, wenn kein neu Palettenkoordinatensystem erzeugt

```
999:ShowDialog (DLP=PLEUEL, OBJ=UVS:M_1, DID=1)
GETVALS      (OBJ=M_1, TYP=UVS, RDS=(a,b), REA=(TDM,ZAB), CDS=(A,B,C,D,E,Z),
              CHS=(~PRUEFER, ~PALCSY, ~KALI, ~NORMAL, ~MGZ, ~Position))
```

! prüft, wenn ~PALCSY gleich N ist, muss unbedingt ~Position auszuwählen.

! wenn ~PALCSY gleich N und ~Position leer ist, springt es zurück zu Dialog Label 999

```
IFTHENS     (ST1=~PALCSY, ST2=N, TYP=EQ)
  IFTHENS   (ST1=~Position, ST2='', TYP=EQ)
    UNWIND  (FRA=2, LAB=999)
  ENDIF
ENDIF
```

! aktuelle X Wert und Maschinentyp ~MA von GDB bekommen

X=GDBREA:X

~MA=GDBCHS:~MA

```
CONCAT      (NAM=~CMM, STR=(MA,~MA), LMG=0, RLS=Y, INI=Y)
GETVAL      (NAM=TDM, OBJ=EDBPRB:PRB(1), DSC=D, TYP=PRB)
```

! neu Koordinatensystem erzeugen und speichern

```
IFTHENS     (ST1=~PALCSY, ST2=Y, TYP=EQ)
  USEPRB    (NAM=PRB(1))
  DELETE    (NAM=(ELE:P_Z0,P_Y0,P_YAX,CSY:P_CSY), CNF=N)
  MEPNT     (NAM=P_Z0, CSY=CMMA$CSY, MOD=NOE, MSG=P_Z0)
  MEPNT     (NAM=P_Y0, CSY=CMMA$CSY, MOD=NOE, MSG=P_Y0)
  MEAXI     (NAM=P_YAX, CSY=CMMA$CSY, MOD=NOE, ITY=GSS, MSG=P_YAX)
```

```

BLDCSY      (NAM=P_CSY(X), TYP=CAR, PLA=P_YAX, PDR=Y, XZE=P_YAX, YZE=P_Y0,
ZZE=P_Z0)
IFTHENS     (ST1=~CMM, ST2=MA393, TYP=EQ)
SAVE        (OBJ=CSY:P_CSY(X), STA=APN, SCB=N, FIL=CNC:TASTER_PLEUEL_MA393.PRB)
ELSEIFS     (ST1=~CMM, ST2=MA125, TYP=EQ)
SAVE        (OBJ=CSY:P_CSY(X), STA=APN, SCB=N, FIL=CNC:TASTER_PLEUEL_MA125.PRB)
ELSEIFS     (ST1=~CMM, ST2=MA321, TYP=EQ)
SAVE        (OBJ=CSY:P_CSY(X), STA=APN, SCB=N, FIL=CNC:TASTER_PLEUEL_MA321.PRB)
ENDIF
CPYOBJ      (FRM=P_CSY(X), TO =P_CSY(X), TYP=CSY, SRC=LDB, DST=GDB)
ELSE
! Wenn vorheriges Koordinatensystem benutzt wird, laden von .PRB Datei
! speichern das aktuelle Koordinatensystem mit Index in der PRB Datei
DELCHS      (NAM=~POS, CNF=N, TYP=CHS)
SUBSTR      (NAM=~POS, RSD=~Position, LM2=1, RM2=1)
CVCHSREA    (NAM=POS, STR=~POS)
IFTHENS     (ST1=~CMM, ST2=MA393, TYP=EQ)
LOAD        (OBJ=CSY:P_CSY(POS), STA=APN, LCB=N, FIL=CNC:TASTER_PLEUEL_MA393.PRB)
SAVE        (OBJ=CSY:P_CSY(X), STA=APN, SCB=N, FIL=CNC:TASTER_PLEUEL_MA393.PRB)
ELSEIFS     (ST1=~CMM, ST2=MA125, TYP=EQ)
LOAD        (OBJ=CSY:P_CSY(POS), STA=APN, LCB=N, FIL=CNC:TASTER_PLEUEL_MA125.PRB)
ELSEIFS     (ST1=~CMM, ST2=MA321, TYP=EQ)
LOAD        (OBJ=CSY:P_CSY(POS), STA=APN, LCB=N, FIL=CNC:TASTER_PLEUEL_MA321.PRB)
ENDIF
CPYOBJ      (FRM=P_CSY(POS), TO =P_CSY(X), TYP=CSY, SRC=LDB, DST=GDB)
IFTHENS     (ST1=~CMM, ST2=MA393, TYP=EQ)
SAVE        (OBJ=GDBCSY:P_CSY(X),          STA=APN,          SCB=N,
FIL=CNC:TASTER_PLEUEL_MA393.PRB)
ELSEIFS     (ST1=~CMM, ST2=MA125, TYP=EQ)
SAVE        (OBJ=GDBCSY:P_CSY(X),          STA=APN,          SCB=N,
FIL=CNC:TASTER_PLEUEL_MA125.PRB)
ELSEIFS     (ST1=~CMM, ST2=MA321, TYP=EQ)
SAVE        (OBJ=GDBCSY:P_CSY(X),          STA=APN,          SCB=N,
FIL=CNC:TASTER_PLEUEL_MA321.PRB)
ENDIF
ENDIF

```


Abbildungsverzeichnis

Abb. 1.1 Produktentstehungsprozess	2
Abb. 2.1 Produktentwicklungsphasen und Konstruktionsänderung	7
Abb. 2.2 Repräsentationsformen für 3D-Modelle	8
Abb. 2.3 Kanten- Flächen- und Volumenmodell	8
Abb. 2.4 Erzeugen eines Volumens durch Vereinigung	10
Abb. 2.5 Boole'sche Operationen für zwei Volumina	10
Abb. 2.6 Darstellung eines Objektes durch drei verschiedene CSG-Bäume	10
Abb. 2.7 B-Rep-Modell einer L-förmigen Platte	11
Abb. 2.8 Datenstruktur eines Hybrid-Modells	11
Abb. 2.9 einstufige Struktur und mehrstufige Struktur	12
Abb. 2.10 Unterteilung der Bauteile	13
Abb. 2.11 Bauarten von Koordinatenmessgeräten	15
Abb. 2.12 Aufbau eines kartesischen Koordinatenmessgeräts	16
Abb. 2.13 Tasterkopfsystem von Leitz Typ LSP-X5	17
Abb. 2.14 Taster von Zeiss	17
Abb. 2.15 verschiedene Tastspitzen	18
Abb. 2.16 Tastaufnahme von Leitz	18
Abb. 2.17 Tastwechselbank und Tastablage von Leitz	19
Abb. 2.18 Gelenk und Würfel von Leitz	20
Abb. 2.19 Konfigurationsdiagramm für LSP-X5 Messsystem	20
Abb. 2.20 Tastbaum (Tastaufnahme, Verlängerung, Gelenk, Taster) und Tastablage	21
Abb. 2.21 Tastbaumausstattung für Bauteil Kurbelgehäuse	22
Abb. 2.22 Die Aufspannungen jedes einzelnen Bauteils	25
Abb. 2.23 Messungsvolumen von Simulator (Leitz)	26
Abb. 2.24 Beziehungen aller Datenbanken in Quindos 7.....	29

Abb. 3.1 Messungsprozess von aktuelle CNC-Messprogramm.....	30
Abb. 3.2 Neuer Entwurf Messungsprozess für Multimesprogramm.....	33
Abb. 3.3 Programmablaufstruktur	34
Abb. 3.4 Auswahlmenü für Multimesprogramm.....	35
Abb. 3.5 dritter Bauteil Kenndaten Übernahme von erster oder zweiter Bauteil	36
Abb. 4.1 Auswahlen für ersten Versuch, zwei Nockenwellen von einen Motor	44
Abb. 4.2 erster Versuch Bauteilkenndaten in GDB	44
Abb. 4.3 zwei Nockenwellen einmal auf der Messmaschine aufgespannt	45
Abb. 4.4 Messprotokoll von Versuch 1	45
Abb. 4.5 Auswahlen für zweiten Versuch, Pleuelstange von zwei Motoren.....	49
Abb. 4.6 zweiter Versuch Bauteilkenndaten	49
Abb. 4.7.a zwei Messpaletten einmal auf Messmaschine Aufspannung	50
Abb. 4.7.b manuelle Ausrichtung für eine Palette	51
Abb. 4.8 Messprotokoll von Versuch 2.....	51
Abb. 4.9 Palettenkoordinatensystem Auswahl.....	53
Abb. 4.10 Aufspannungsplan für den Versuch 3	55
Abb. 4.11 die Dialog für Messungsposition Auswahl	56
Abb. 4.12 Messprotokoll für Versuch 3	57

Tabellenverzeichnis

Tab. 2.1 die Anzahl der verwendeten Tastbäumen von entsprechenden Bauteilen	22
Tab. 2.2 Die verwendeten Taster des Bauteils mit selben Tastbaum.....	23
Tab. 3.1 Die normale Messungszeit von Messbauteilen	39
Tab. 3.2 Eine Messmaschine in ein paaren Tage mit seiner Messungslist	40
Tab. 4.1 die Nockenwelle von Motor FH5511	42
Tab. 4.2 Messkatalog für Nockenwelle Auslass	43
Tab. 4.3.a Messdaten von dem Versuch 1	46
Tab. 4.3.b Messdaten von dem Versuch 1	46
Tab. 4.4 Messungsaufträge für Pleuelstange von Motor AE3921 und AE8419	47
Tab. 4.5 Messkatalog für Pleuelstange.....	48
Tab. 4.6 automatische Positionsverteilung von dem Koordinatensystem der Palette.....	50
Tab. 4.7.a Messdaten von Versuch 2	52
Tab. 4.7.b Messdaten von Versuch 2	52
Tab. 4.8 Messungsaufträge für Zylinderlaufbuchse und Pleuelstange von zwei Motoren	54
Tab. 4.9 Messkatalog für Zylinderlaufbuchse.....	54
Tab. 4.10 die Positionen der Befestigungsbohrungen.....	56
Tab. 4.11.a Messdaten von Versuch 3	57
Tab. 4.11.b Messdaten von Versuch 3	58
Tab. 4.12 Ergebnisse von weiteren Versuchen	59

Abkürzungsverzeichnis

3D	drei Dimensionen
CNC	(Engl.) Computerized Numerical Control
DG	Grundkörper Durchmesser
DK	Kugeldurchmesser
DS	Schaftdurchmesser
EDB	Umgebungs-Datenbank
ERFO	erfolgreich abgeschlossen
ERFT	teilerfolgreich abgeschlossen
FREI	frei für Ressourcenplanung
FRVD	frei für Versuchsdurchführung
FRVV	frei für Versuchsverwendungsplanung
GDB	globale Datenbank
GEPL	geplant
HDEP	(Engl.) Heavy-Duty Engine Platform
KMG	Koordinatenmessgerät
L	Gesamtlänge
LDB	lokale Datenbank
LKW	Lastkraftwagen
MDEG	(Engl.) Medium-Duty Engine Generation
ML	Messlänge
NERF	nicht erfolgreich abgeschlossen
QUINDOS	(Engl.) Quality Inspection of Dimensional Objects and Sizes
SDB	Sprachunabhängige System-Datenbank
SOP	(Engl.) Start of Production
TP/PPH	Abteilung von Daimler Truck Produktentwicklung
UDB	Benutzer-Datenbank
WDB	Werkstück-Datenbank
WTS	Werkstücktemperatur Sensor
ZDB	Kundenspezifische Datenbank
ZRST	zurückgestellt

Literaturverzeichnis

- [1] Verband der Automobilindustrie e.V., "Jahresbericht 2012 VDA", [Online], verfügbar : www.vda.de/de/downloads/1092/, 2012
- [2] Dr. Klaus Waßenhoven, "Truck Group Produktentstehungsprozess", Daimler AG / Untertürkheim, 2009
- [3] S. vajana, C. Weber, "CAx für Ingenieure", Verlag Springer, 2009
- [4] "CAx", [Online], verfügbar : <http://de.wikipedia.org/wiki/CAx>, 2013
- [5] G. Spur, F. Krause, "CAD-Technik. Lehr- und Arbeitsbuch für die Rechnerunterstützung in Konstruktion und Arbeitsplanung", Verlag Hanser, 1984
- [6] Dieter Roller, "CAD effiziente Anpassungs- und Variantenkonstruktion", Verlag Springer, 1995
- [7] Profos P., Pfeifer T., "Handbuch der industriellen Messtechnik", R. Oldenbourg, 1994
- [8] "Grundlagen der Messtechnik - Teil1: Grundbegriffe", DIN 1319-1:1995 Nr. 2.1, 1995
- [9] Pfeifer, T. (Hrsg.), "Koordinatenmesstechnik für die Qualitätssicherung", VDI Düsseldorf, 1992
- [10] Unternehmen Carl Zeiss, "Koordinatenmesstechnik", Verlag Moderne Industrie, Landsberg/Lech, 1990
- [11] Hexagon Metrology, "Taster & Sensoren für Koordinatenmessgeräte Produktkatalog", Deutschland, 2012
- [12] Daimler AG Abteilung TP/PPH, "Koordinatenmessgeräte", 2012
- [13] Messung, <http://de.wikipedia.org/wiki/Messung>, 2013
- [14] "Messung", [Online], verfügbar : <http://de.wikipedia.org/wiki/Messprozess>, 2012
- [15] Hexagon Metrology, "QUINDOS - Software für Koordinatenmessgeräte und Formprüfer", Deutschland, 2012