

Erweiterung des Ablaufs und der Visualisierung einer Prüfvorrichtung für Schraubklemmen

Ausbildungsberuf
Fachinformatiker für Anwendungsentwicklung

Jan-Philipp Töberg
Vahlhauser Straße 35a
32825 Blomberg

Abgabetermin:
13.12.2017

Inhaltsverzeichnis

<i>Inhaltsverzeichnis</i>	<i>I</i>
<i>Tabellenverzeichnis</i>	<i>II</i>
<i>Abbildungsverzeichnis</i>	<i>II</i>
<i>Abkürzungsverzeichnis</i>	<i>II</i>
<i>1. Einleitung</i>	<i>1</i>
1.1 Projektbeschreibung	1
1.2 Projektziel.....	1
1.3 Projektumfeld.....	1
1.4 Projektschnittstellen	2
<i>2. Projektplanung</i>	<i>2</i>
2.1 Projektphasen	2
2.2 Ressourcenplanung	3
2.3 Kosten- & Personalplanung	4
<i>3. Analysephase</i>	<i>4</i>
3.1 IST-Analyse	4
3.2 Wirtschaftlichkeitsanalyse	5
3.3 Anforderungen	5
3.3.1 Anforderungen an den Ablauf	5
3.3.2. Anforderungen an die Visualisierung.....	5
<i>4. Entwurfsphase</i>	<i>5</i>
4.1 Entwurf der Benutzeroberfläche	5
4.2 Entwurf des Ablaufs	6
<i>5. Implementierungsphase</i>	<i>7</i>
5.1 Optimierung des alten Quellcodes	7
5.2 Implementierung des neuen Ablaufes.....	7
5.3 Anpassen der Visualisierung.....	8
<i>6. Fazit</i>	<i>8</i>
6.1 Soll-/Ist-Vergleich	8
6.2 Auswertender Vergleich der Zeitplanung	9
6.3 Erweiterungsmöglichkeiten	9
<i>7. Hinweis auf vertrauliche Daten</i>	<i>10</i>
<i>8. Quellen</i>	<i>10</i>
<i>A Anhang</i>	<i>i</i>

Erweiterung des Ablaufs und der Visualisierung einer Prüfvorrichtung für Schraubklemmen

A1 – Ausführliche Zeitplanung	i
A2 – Entwürfe der Benutzeroberfläche	i
A3 – Screenshots der bisherigen Visualisierung	ii
A4 – Bilder der Prüfvorrichtung.....	iii
A5 – Programmablaufpläne.....	iv
A6 – Programmcode	vi
A6.1 – „ParameterSettings“-Baustein.....	vi
A6.2 – “NumberOfTurns“-Baustein	vii
A6.3 – “TimeToScrew“-Baustein	viii
A6.4 – Ausschnitt aus dem “Pruef_Hand“-Baustein.....	viii
A7 – Eidesstattliche Erklärung	x
A8 – Genehmigter Projektantrag.....	xi

Tabellenverzeichnis

Tabelle 1 – Projektphasen	2
Tabelle 2 - Genutzte Hardware	3
Tabelle 3 - Genutzte Software	3
Tabelle 4 – Personalkostenplanung	4
Tabelle 5 - Ausführliche Zeitplanung.....	i

Abbildungsverzeichnis

Abbildung 1 - Entwurf der Parameterseite	i
Abbildung 2 - Entwurf der Startseite	ii
Abbildung 3 - Der bisherige Parameter-Bildschirm	ii
Abbildung 4 - Die bisherige Startseite	iii
Abbildung 5 - Der Handschrauber	iii
Abbildung 6 - PAP für den Ablauf-Baustein	iv
Abbildung 7 - PAP für die Schrittkette	v

Abkürzungsverzeichnis

ECAR	Servoverstärker für bürstenbehaftete Gleichstrom-Motoren (DC-Motoren) und bürstenlose Gleichstrom-Motoren (EC-Motoren)
HMI	Human-Machine-Interface
EN	Europäische Norm
DIN	Deutsche Industrie-Norm
PAP	Programmablaufplan
PNG	Portable Network Graphics
CSV	Comma-Separated Values
TXT	Textdatei
ST	Strukturierter Text

1. Einleitung

Diese Projektdokumentation behandelt den genauen Ablauf des Projektes „Erweiterung des Ablaufs und der Visualisierung einer Prüfvorrichtung für Schraubklemmen“, welches der Autor im Rahmen seiner Ausbildung zum Fachinformatiker für Anwendungsentwicklung bei Phoenix Contact GmbH & Co. KG durchgeführt hat. Phoenix Contact ist weltweiter Marktführer für Komponenten, Systeme und Lösungen im Bereich der Elektrotechnik, Elektronik und Automation. Das Familien-Unternehmen beschäftigt heute rund 15.000 Mitarbeiter weltweit und hat in 2016 einen Umsatz von 1,97 Mrd. Euro erwirtschaftet. Der Stammsitz ist im westfälischen Blomberg. Zur Phoenix Contact-Gruppe gehören zwölf Unternehmen in Deutschland sowie mehr als 50 eigene Vertriebs-Gesellschaften in aller Welt. Die internationale Präsenz wird zusätzlich durch mehr als 40 Vertretungen in Europa und Übersee verdichtet¹.

Ziel dieser Dokumentation ist es, die durchgeführten Schritte zur Analyse, Planung und Durchführung zusammenzufassen und mit Hilfe von Beschreibungen und Diagrammen zu verdeutlichen.

1.1 Projektbeschreibung

Bei dem Projekt „Erweiterung des Ablaufs und der Visualisierung einer Prüfvorrichtung für Schraubklemmen“ wird mit einer Prüfvorrichtung für Schraubklemmen gearbeitet, welche im Folgenden als Handschrauber bezeichnet wird. Die Aufgabe dieser Maschine ist, bei einer vorgehaltenen Schraubklemme, die Schraube einmal komplett ein- und danach wieder herauszuschrauben. Somit kann auf eine zeitsparende Art und Weise getestet werden, ob die Schrauben der Klemme funktionstüchtig sind.

Dieser beschriebene Handschrauber soll nun im Zuge dieses Projektes um die Funktionalität erweitert werden, die Ausrichtung der Schraube nach dem Prüfvorgang festzulegen. Um dieses Ziel zu erreichen, müssen sowohl der aktuelle Ablauf, als auch die dazugehörige Visualisierung angepasst werden.

1.2 Projektziel

Das Ziel des Projektes ist, wie oben schon angedeutet, die Ausrichtung der Schraube nach dem Prüfvorgang vorher in der Visualisierung einstellen zu können. Dies würde dafür sorgen, dass bei einer getesteten Klemme alle Schrauben gleich ausgerichtet sind, was in der Automatisierung des Schaltschrankaufbaus von Vorteil sein wird, da so die Schrauben auch automatisiert zugeschraubt werden können.

1.3 Projektumfeld

Das Projekt wird in der Abteilung „Corporate Technology & Value Chain, Fixture Engineering“ des Maschinenbaus von Phoenix Contact durchgeführt. Diese Abteilung beschäftigt sich zum einen mit dem Training und der Ausbildung von diversen Ausbildungsberufen

¹ Siehe 8. Quellen – Quelle Nummer 1

(Mechatroniker/In, Elektroniker/In für Geräte und Systeme, Industriemechaniker/In, Technische/r Produktdesigner/In) und zum anderen mit dem Entwurf, dem Aufbau und der Programmierung von technischen Aufbauten zu Testzwecken oder von Messgeräten für bestimmte Prüfsituationen.

Der Handschrauber wurde von Teilen des ganzen Maschinenbaus entwickelt und gefertigt und wird sowohl intern genutzt als auch weltweit an externe Kunden verkauft. Die wesentlichen Absatzmärkte liegen in Polen, China und Indien.

Ansprechpartner: Viktor Wölk

Telefon: +49 5235 3-48534

E-Mail: vwoelk@phoenixcontact.com

1.4 Projektschnittstellen

Dieses Projekt wird in enger Zusammenarbeit mit dem Projekt „Erweiterung eines Motortreibers für den Maschinenbau“ durchgeführt, da beide Projekte auf der gleichen Maschine aufsetzen. Das Projekt „Erweiterung eines Motortreibers für den Maschinenbau“ befasst sich mit der Entwicklung eines allgemein einsetzbaren Treiberbaustein, mit Hilfe dessen verschiedene Motoren über die eingesetzte ECAR-Klemme genutzt werden können. Aktuell ist der Treiberbaustein auf den eingesetzten Motor und den benötigten Betriebsmodus fest eingestellt, was durch dieses Projekt modularer und flexibler gestaltet werden soll. Im weiteren Verlauf dieser Dokumentation wird dieses Projekt als „Treiberbaustein“ bezeichnet.

2. Projektplanung

Im Folgenden wird auf die genaue Planung des Projektes eingegangen.

2.1 Projektphasen

Projektphase	geplante Zeit (in h)
Einarbeitung in das bisherige Programm	10
Abstimmung mit der Treiberbaustein Entwicklung	5
Programmierung des neuen Ablaufs	20
Erweiterung der Visualisierung	20
Funktionalitätsprüfung mit Fehlerkorrektur & Optimierung	5
Dokumentation des Projekts	10
Summe	70

Tabelle 1 – Projektphasen

Das Projekt beginnt mit einer geplanten Einarbeitungszeit in die bisherigen Funktionalitäten des Handschraubers und die dazugehörige Visualisierung. Da die Maschine zum Start des Projektes bereits voll funktionstüchtig ist, ist für diesen Teil ein verhältnismäßig großer Zeitraum eingeplant. Da dieses Projekt, wie in Abschnitt 1.4 Projektschnittstellen bereits beschrieben, sehr eng mit dem Treiberbaustein-Projekt zusammenhängt, ist ein Teil des Projektes zur Absprache mit diesem Projekt geplant. In dieser Phase wird über die Schnittstelle zwischen dem Ablauf und dem Baustein gesprochen, wie diese aufgebaut ist

und welche Schritte für die Kommunikation unternommen werden müssen. Daraufhin sollten die Grundlagen soweit geklärt worden sein, wodurch mit der Anpassung/Entwicklung des neuen Ablaufs und der Visualisierung angefangen werden kann. Diese Projektphasen laufen parallel ab, da Änderungen im Ablauf direkten Einfluss auf die Visualisierung haben. Nachdem die Anpassungen vorgenommen wurden, wird getestet, ob die Änderungen das gewünschte Ergebnis erzielen und ggf. werden noch weitere Anpassungen vorgenommen. Am Ende des Projektes wird die Dokumentation geschrieben.

Diese Planung wurde vor dem Beginn des Projektes durchgeführt. In Abschnitt 6.2 wird dieser Einschätzung der tatsächliche Projektverlauf gegenüber gestellt und es wird analysiert, wieso eventuelle Diskrepanzen aufgetreten sind.

2.2 Ressourcenplanung

Hardware	
Notebook	Fujitsu Lifebook E754
Maus	Logitech M100
Steuerung	ILC 150 ETH
HMI	Touch-Panel - WP 07T/WS
Motor	Maxon EC-max 30W

Tabelle 2 - Genutzte Hardware

Software	
Betriebssystem	Windows 10 Pro (Version 1607)
E-Mail	Lotus Notes 9
Entwicklungsumgebung	PCWorx 6.30 Build 103
Visualisierung	WebVisit 6.30.04 Professional
Entwürfe und Diagramme	PapDesigner Version 2.2.0.8.02 Evolus Pencil Version 3.0.4
Office	Microsoft Office Standard 2010

Tabelle 3 - Genutzte Software

Bis auf die Steuerung, das HMI und den Motor wird ausschließlich Hardware genutzt, welche das Unternehmen Phoenix Contact allen Auszubildenden zur Verfügung stellt. Die restlichen drei Hardwarekomponenten sind in jedem Handschrauber verbaut und können für das Projekt genutzt werden, da ein gesamter Handschrauber als Testobjekt zur Verfügung steht. Die Software wurde entweder auch von Phoenix Contact zur Verfügung gestellt oder es handelt sich um eine kostenfreie Version

Bei dem Programm „PCWorx“ handelt es sich um ein eigenes Produkt der Phoenix-Contact-Gruppe, welches genutzt wird um Steuerungen zu programmieren. Die dafür zur Verfügung stehenden Programmiersprachen sind in der Norm EN 61131 zusammengefasst und spezifiziert². Das Programm „WebVisit“ wird ebenfalls von Phoenix Contact entwickelt und wird eingesetzt, um Visualisierungen auf HMIs zu erstellen und mit der Steuerung zu verbinden. Dabei wird bei WebVisit eine Website erstellt und auf der Steuerung hinterlegt,

² Siehe 8. Quellen – Quelle Nummer 2

Erweiterung des Ablaufs und der Visualisierung einer Prüfvorrichtung für Schraubklemmen

welche dann bei der Verbindung mit dem HMI angezeigt werden kann. Verarbeitet werden die Änderungen in der Visualisierung dabei direkt von der Steuerung.

Der „PapDesigner“ wurde vom Georg-Friedrich-Ohm-Berufskolleg in Köln entwickelt und erlaubt die Erstellung von Programmablaufplänen nach der DIN 66001. Die Software „Evolus Pencil“ ermöglicht den Entwurf von graphischen Benutzeroberflächen und wurde von der Evolus Company entwickelt, welche das Programm über Spenden finanzieren. Die Programme Lotus Notes und Microsoft Office sollten allgemein bekannt sein und bedürfen keiner weiteren Erklärung.

2.3 Kosten- & Personalplanung

Für die Berechnung der Personalkosten wird ein exemplarischer Stundensatz von Phoenix Contact veranschlagt, da die richtigen Stundensätze nicht veröffentlicht werden. Bei diesem wird für einen Facharbeiter mit 35€ und für einen Auszubildenden mit 10€ die Stunde gerechnet. Für die Ressourcennutzung wird mit einem Stundensatz von 40€ gerechnet.

Verursacher	Arbeitszeit (in h)	Personalkosten (in €)	Ressourcenbeitrag (in €)	Gesamt (in €)
Ansprechpartner / Auftraggeber	4	35,00	40,00	300,00
Auftragnehmer	70	10,00	40,00	3 500,00
Gesamtaufwand				3 800,00

Tabelle 4 – Personalkostenplanung

Da für das Projekt keine Ressourcen extra angeschafft werden müssen, da sowohl die Hardware als auch die Software bereits im Unternehmen zur Verfügung stehen, wird auch keine Kostenberechnung dafür durchgeführt.

An Personal wird für dieses Projekt nur der Auftraggeber, welcher auch gleichzeitig die Rolle des Ansprechpartners innehat, eingeplant. Mit diesem werden zwei Stunden zur zwischenzeitlichen Absprache und zwei Stunden zur Abnahme des Projektes geplant.

3. Analysephase

Nachdem die Planung des Projektes abgeschlossen wurde, wird mit der Analyse der bisherigen Funktionen des Handschraubers und der Anforderungen begonnen.

3.1 IST-Analyse

Wie in Abschnitt 1.1 Projektbeschreibung bereits kurz beschrieben, bietet der Handschrauber aktuell die Möglichkeit, eine Schraube testweise ein- und wieder herauszuschrauben. Dabei besitzt die Maschine insgesamt neun verschiedene interne Programme, welche alle auf eine unterschiedliche Art und Weise diesen Vorgang durchführen. Diese Programme können vor Beginn des Projektes entweder über die Visualisierung, oder einen Drehschalter an der Frontseite der Maschine ausgewählt werden. Je nach gewähltem Programm wird der Start mit Hilfe eines Fußkippschalters realisiert,

welcher an der Maschine angeschlossen ist, oder automatisch durch das Andrücken der Schraube an den Kopf des Schraubers.

3.2 Wirtschaftlichkeitsanalyse

Im Folgenden würde analysiert werden, in wie weit die vom Projekt verursachten Kosten gerechtfertigt sind und wann bzw. ob diese von dem Nutzen des Projektes aufgefangen werden. Diese Überlegungen sind für dieses Projekt aber nur schwer bzw. gar nicht durchführbar. Das liegt daran, dass kein neues Produkt entwickelt wird und auch keine Funktion implementiert wird, die große Zeiteinsparungen bringen würde. Dieses Projekt implementiert bei seiner Fertigstellung nur einen weiteren Vorteil des Produkts, welches als Kaufgrund genannt werden könnte. Aber selbst dann ist die neue Funktionalität nicht für alle Kunden interessant.

3.3 Anforderungen

Die Anforderungen des Auftraggebers an den neuen Ablauf und an die Visualisierungen werden im Folgenden stichpunktartig dokumentiert. Teilweise sind diese Anforderungen nicht direkt vom Auftraggeber spezifiziert, sondern ergeben sich aus einer der anderen Anforderungen.

3.3.1 Anforderungen an den Ablauf

- Ø Ausschraubvorgang nur bis bestimmter Winkel erreicht
- Ø Nutzung des neuen Treiberbausteins
- Ø Optimierung des alten Programmcodes

3.3.2. Anforderungen an die Visualisierung

- Ø Einstellen der gewünschten Ausrichtung
- Ø Einstellen der neuen Parameter, welche durch den Treiberbaustein dazukommen
- Ø Anpassung der Parameter aktualisieren

4. Entwurfsphase

Nachdem die genauen Umstände des Projektes in der Analysephase spezifiziert wurden, können nun erste Entwürfe angefertigt werden, welche in der Implementierungsphase dann umgesetzt werden. Dabei werden die Entwürfe für die Benutzeroberfläche und den Ablauf im Folgenden getrennt voneinander betrachtet.

4.1 Entwurf der Benutzeroberfläche

Bei der Benutzeroberfläche müssen die beiden Bildschirme „Startseite“ und „Parameter“ angepasst werden, da bei diesen beiden Bildschirmen Änderungen getätigt werden müssen. Bei der Anpassung und gerade der Erstellung von neuen Bildschirmen muss jedoch auf die Einschränkung der Bildschirmauflösung durch das genutzte HMI geachtet werden. Dieses hat eine native Auflösung von 800 mal 480 Pixeln, weshalb die Bildschirme diese Größe nicht überschreiten sollen.

Die geplanten Änderungen an der Startseite sind, wie in den Abschnitten A2 – Entwürfe der Benutzeroberfläche und A3 – Screenshots der bisherigen Visualisierung zu sehen, minimal. Anstatt das maximale Drehmoment anzugeben, soll sich nun die gewünschte Ausrichtung in Grad angeben lassen. Außerdem wird zwischen der Beschreibung des aktuellen Programms und dem derzeitigen Schritt der Schrittkette die Beschreibung des momentanen Betriebsmodus angezeigt. Diese Neuerung beruht auf der Anpassung des Treiberbausteins, welcher ermöglicht, dass der Motor in allen einstellbaren Modi betrieben werden kann und nicht ausschließlich im Drehzahl-Modus.

Die größte Veränderung zeigt sich jedoch beim neuen Bildschirm zur Parameter-Einstellung. Für diesen wurde von Grund auf ein neues Design erarbeitet, da, wieder begründet durch den neuen Treiberbaustein, mehr Parameter existieren, welche eingestellt werden können. Dementsprechend wird der Parameter-Bildschirm um die Möglichkeit erweitert die Seite zu wechseln, um mehr als die ursprünglichen zwölf Parameter einstellen zu können. Des Weiteren soll das, von diesem Parametersatz genutzte, Programm über diesen Bildschirm einstellbar sein und nicht mehr über das PCWorx-Projekt. Dies wurde über den zusätzlichen, 13ten Parameter realisiert.

4.2 Entwurf des Ablaufs

Zur besseren Implementierung des Ablaufs wurde für diesen ein Programmablaufplan erstellt, welcher im Abschnitt A5 – Programmablaufpläne zu sehen ist. Dieser behandelt den Aufbau des sogenannten „Pruef_Hand“-Bausteins, welcher den wichtigsten Aspekt des Ablaufs darstellt. Der Baustein beginnt mit dem einmaligen Setzen der Standardwerte über den „DefaultParameters“-Baustein. Daraufhin wird ein Teil der Werte an die Visualisierung übertragen und, je nach Stellung des Drehschalters, werden die restlichen Werte entweder übertragen oder gelesen. Danach wird die Anzahl der benötigten Umdrehungen berechnet, bis die Schraube ganz ein- oder ausgeschraubt wurde. Dieser Wert wird dann direkt verwendet, um zu bestimmen, wie lange der Befehl zum Schrauben gegeben werden soll. Diese beiden Berechnungen erfolgen über zwei unterschiedliche Bausteine, welche in den Abschnitten A6.2 – „NumberOfTurns“-Baustein und A6.3 – „TimeToScrew“-Baustein eingesehen werden können.

Bis zu diesem Punkt muss der vorherige Ablauf nicht angepasst werden, da keine Änderungen vorgenommen werden müssen. An diesem Punkt beginnt jedoch die Schrittkette, welche aktuell eines von neun unterschiedlichen Programmen auswählt und durchläuft. In diesem Projekt wird ein neues Programm entwickelt, welches die neuen Funktionalitäten demonstrieren soll. Die Möglichkeit weitere Programme hinzuzufügen soll dabei beibehalten werden. Dementsprechend wird in der Schrittkette, nachdem anfangs einmalig die Parameter übertragen werden, die Schrittnummer des benötigten Programms berechnet. Dazu wird die eingestellte Programmnummer mit 100 multipliziert.

Der zu implementierende Ablauf beginnt als einziges Programm dementsprechend bei Schritt 100 und geht in Fünferschritten weiter. Begonnen wird nun mit der rechtsläufigen Schlitzsuche, welche solange läuft bis durch die Erhöhung des Drehmoments eine Schraubklemme erkannt wird. Sobald diese erkannt wird und der Schraubkopf des Motors eingerastet ist, soll die Schraube über eine bestimmte Zeit mit hoher und die letzten Millimeter mit geringer Geschwindigkeit eingeschraubt werden. Sobald die Schraube

komplett auf ein bestimmtes Drehmoment eingeschraubt ist, wird die Drehrichtung des Motors geändert und es wird angefangen, die Schraube wieder auszuschauben. Dieser Vorgang soll solange laufen, bis eine bestimmte Ausrichtung des Schraubkopfes erreicht ist.

5. Implementierungsphase

In dieser Phase des Projektes werden die in der vorherigen Phase geplanten Änderungen umgesetzt und implementiert. Die Implementation findet dabei in der Programmiersprache ST statt, einer von fünf unterschiedlichen Programmiersprachen mit denen nach der EN 61131 bei PCWorx programmiert werden kann³.

5.1 Optimierung des alten Quellcodes

Bevor der neue Ablauf implementiert werden kann, wird der alte Programmcode angesehen und teilweise optimiert. Dabei werden neben einigen kleineren Änderungen folgende nennenswerte Anpassungen vorgenommen.

Die erste, größere Anpassung befasst sich mit der Variablendefinition. Bei PCWorx sind alle Variablen in Tabellen organisiert, was die Übersicht verdeutlichen kann. Diese Tabelle ist jedoch unsortiert und unkommentiert, weshalb verschiedene Variablengruppen eingeführt werden und die Kommentare der bereits vorhandenen Variablen ergänzt werden.

Die zweite Änderung befasst sich mit der Einstellung der einzelnen Parameter. Für diesen Vorgang existiert ein einzelner Baustein, der am Anfang des Ablaufs einmal aufgerufen und damit ausgeführt wird. Dieser Baustein stellt die Standardwerte für die verschiedenen Parametersätze ein, welche in der Visualisierung dann angepasst werden können. Diese Einstellung verlief in der alten Version des Quellcodes allerdings auf eine ineffiziente Art und Weise, da für alle neun Parametersätze (für jedes Programm einen) jede der zwölf Variablen untereinander gesetzt wurde. Dabei haben elf dieser Variablen für alle neun Programme den gleichen Wert und die zwölfte Variable ist eine hochzählende Nummer. Dieses Setzen der Variablen wird nun in einer Zählschleife zusammengefasst.

Des Weiteren existiert im normalen Ablauf ein Abschnitt, wo für jeden Parametersatz bestimmt wird, welche der zwölf Parameter letzten Endes verwendet werden. Dieser Teil wird ebenfalls in den Baustein ausgelagert, um alle grundlegenden Einstellungen, die die Parameter betreffen an einem Ort zu sammeln. Dieser Baustein, welcher vorher als „DefaultParameters“ bezeichnet wurde, wird nun „ParameterSettings“ genannt und findet sich im Anhang.

5.2 Implementierung des neuen Ablaufes

Nachdem der alte Code weitestgehend optimiert wurde, kann damit begonnen werden, den neuen Ablauf zu implementieren. Um wie gewollt den neu-entwickelten Treiberbaustein nutzen zu können, müssten die alten Programme der Schrittkette angepasst werden. Da dies allerdings nicht Aufgabe dieses Projektes ist, werden die Programme ausgeschnitten und in

³ Siehe 8. Quellen – Quelle Nummer 2

einer txt-Datei abgespeichert, welche als Sicherheitskopie für zukünftige Anpassungen zur Verfügung steht.

Der neue Ablauf kann dementsprechend als Programm mit der Nummer Eins implementiert werden. Aufbauend auf dem in Abschnitt 4.2 Entwurf des Ablaufs vorgestellten Entwurf wird damit begonnen, die neuen Parameter des Treiberbausteins an die Visualisierung zu übertragen bzw. die neuen Parameter innerhalb der Visualisierung (z.B. die Ausrichtung) auszulesen. Nachdem dieser Teil angepasst wurde, wird der Programmteil der Schrittkette bearbeitet. Dieser beginnt - ähnlich einem der alten Programme - mit der Steuerung des Motors im Drehzahl-Modus. Dabei wird anfangs mit moderater Drehzahl gedreht, bis durch Veränderung des Drehmoments erkannt wird, dass eine Klemme unter den Schraubkopf gehalten wird. Daraufhin beginnt dieser die Schraube zuerst mit hoher, dann mit geringer Geschwindigkeit einzuschrauben. Sobald die Schraube eingeschraubt ist, wird die Drehrichtung geändert. Ab diesem Punkt beginnt der neu erstellte Teil des Projektes, welcher über einen Parameter den Modus wechselt, um den Motor über den Referenzfahrt-Modus zu steuern. In diesem Modus kann der Motor genau positioniert werden.

Die genaue Position, zu welcher der Motor die Schraube ausrichten soll, wird über die Zeit berechnet, die der Motor drehen soll. So existiert eine Konstante, welche speichert, wie viele sogenannte Counts der aktuelle Motor für eine Umdrehung berechnet. Mit Hilfe dieser Konstanten und der gewünschten Ausrichtung in Grad wird berechnet, wie viele Umdrehungen der Motor machen muss, um die gewünschte Ausrichtung einzustellen. Sobald die Anzahl der Umdrehungen bekannt ist, kann mit Hilfe des „TimeToScrew“-Bausteins (*siehe Anhang A6.3 – „TimeToScrew“-Baustein*) bestimmt werden, wie lange die Schraube gedreht werden muss.

5.3 Anpassen der Visualisierung

Die Anpassung der Visualisierung verlief problemlos, da die in Abschnitt 4.1 Entwurf der Benutzeroberfläche geplanten Änderungen ohne weitere Einschränkungen umgesetzt und eingefügt werden konnten. Lediglich der Umgang mit den unterschiedlichen Internationalisierungsmöglichkeiten wurde in der vorherigen Phase nicht geplant. Diese Möglichkeiten belaufen sich auf mehrere csv-Dateien, welche für bestimmte Platzhalter-Strings einen Text auf der dazugehörigen Sprache hinterlegt haben. Diese Dateien existieren für die Sprachen Deutsch, Englisch, Polnisch und Türkisch, jedoch wurden lediglich die Sprachen Deutsch und Englisch angepasst.

6. Fazit

Abschließend wird auf den Verlauf des Projektes zurückgeblickt und es wird eingeschätzt, wie erfolgreich und effizient das Projekt durchgeführt wurde.

6.1 Soll-/Ist-Vergleich

In diesem Abschnitt wird eingeschätzt, ob die in Abschnitt 3.3 Anforderungen spezifizierten Anforderungen an dieses Projekt erfolgreich umgesetzt wurden oder nicht. Dabei werden die Anforderungen an die Visualisierung und an den Ablauf wieder getrennt betrachtet.

Bei den Anforderungen an die Visualisierung lässt sich abschließend sagen, dass diese voll funktionstüchtig umgesetzt wurden und einsatzbereit sind. Bei den Anforderungen an den Ablauf gestaltet sich diese Einschätzung allerdings schwierig. So wurden zwar alle Änderungen umgesetzt, jedoch konnten diese im Rahmen dieses Projektes nicht auf ihre Korrektheit getestet werden. Das liegt an der Abhängigkeit vom Treiberbaustein-Projekt. Da kein voll funktionsfähiger Baustein zum Endzeitpunkt zur Verfügung stand, dieser jedoch für die Funktion des Ablaufs benötigt wird, konnten die Funktionalitäten nicht getestet werden.

Dementsprechend lässt sich keine abschließende Einschätzung der Korrektheit dieses Projekt durchführen.

6.2 Auswertender Vergleich der Zeitplanung

Wie in Abschnitt A1 – Ausführliche Zeitplanung einzusehen, existieren größere Diskrepanzen zwischen der geplanten Dauer der einzelnen Projektphase und der letzten Endes investierten Zeit. Das hat mehrere unterschiedliche Gründe, die im Folgenden dargelegt werden.

Der erste Grund hängt mit der Visualisierung zusammen. Die Änderungen, die vorgenommen werden mussten, waren weniger aufwendig als vorher angenommen. Das lag zum einen an der guten Unterstützung, welche die Software WebVisit dem Benutzer bietet und zum anderen an der Überschätzung der Komplexität der Änderungen.

Der andere Grund hängt mit dem Treiberbaustein-Projekt zusammen, welches parallel zu dem hier vorgestellten Projekt entwickelt wurde. Bei diesem Projekt erwies sich die Einarbeitung in den bisherigen Programmcode und in die Funktionen der genutzten ECAR-Klemme ausführlicher als anfänglich geplant, was dort zu Verzögerungen gesorgt hat. Diese Verzögerungen haben sich direkt in das hier beschriebene Projekt übertragen, da dieses Projekt auf dem Treiberbaustein aufsetzt. Am deutlichsten wird diese Verzögerung bei der geplanten Zeit zur Funktionalitätsprüfung und der anschließenden Fehlerkorrektur. Wie im vorherigen Abschnitt bereits erwähnt, ist Treiberbaustein zum Ende dieses Projektes nicht voll funktionsfähig, wodurch sich die durchgeführten Änderungen am Ablauf auch nicht ausprobieren ließen.

Die Diskrepanz bei der investierten Zeit in die Dokumentation lässt sich über anfängliche Unklarheiten mit der Bedienung von WebVisit feststellen. So bietet WebVisit eine Funktion, die erstellten Bildschirme als Bilddateien im png-Format zu exportieren. Dieser Export-Vorgang überschreibt wiederum alle Daten, welche sich im angegebenen Ordner befinden, in welchem sich zu diesem Zeitpunkt noch die erste Version der Dokumentation befand. Da diese noch zu neu war, gab es kein Backup, weshalb die anfänglichen 2-3 Stunden Fortschritt verloren gegangen sind und aufgearbeitet werden mussten.

6.3 Erweiterungsmöglichkeiten

Mit der Fertigstellung des Projekts endet jedoch nicht die Weiterentwicklung am Handschrauber. So bietet der aktuelle Aufbau des Programms dank der Schrittketten-Programmierung die Möglichkeit, eine Vielzahl von weiteren Programmen und Abläufen zu implementieren und einzubauen. Diese können komplett unabhängig vom Ablauf dieses Projektes sein oder diesen um Erweiterungen ergänzen.

Die gleiche Erweiterbarkeit bietet auch die Visualisierung. Der neue Bildschirm für die Parametereinstellung wurde als Vorlage angelegt und kann, sollten noch mehr Parameter dazukommen, einfach ergänzt werden. Dank der Funktionalität von WebVisit lassen sich auch die Übergänge von einem Bildschirm auf den nächsten leicht ändern und anpassen. Ein weiterer Punkt an der Visualisierung, welcher während dieses Projektes nicht behandelt wurde, sind die unterschiedlichen Sprachen. Wie oben bereits erwähnt bietet WebVisit die Möglichkeit, Platzhalter-Strings zu definieren, welche dann durch Texte, die in unterschiedlichen csv-Dateien hinterlegt sind, ersetzt werden können. Zur Verfügung stehen dabei die Sprachen Deutsch, Englisch, Polnisch und Türkisch. Auf Grund mangelnder Sprachkenntnisse wurden die Texte bisher nicht auf Polnisch oder Türkisch übersetzt. Diese Übersetzungen sollten in Zukunft aber hinzugefügt werden.

7. Hinweis auf vertrauliche Daten

Diese Dokumentation und die nachfolgenden Anlagen enthalten firmenrelevante Daten, die nicht für die Öffentlichkeit bestimmt sind. Aus diesem Grund sind alle Informationen vertraulich zu behandeln.

8. Quellen

1) Phoenix>Contact_Unternehmensprofil.pdf

Autor: Phoenix Contact GmbH & Co. KG

Erstellt am: 09.08.2017

Heruntergeladen am: 05.12.2017

Link: [Phoenix Contact Unternehmensprofil](#)

2) SPS-Programmierung: Speicherprogrammierbare Steuerungen

Autor: Cengiz Ay (info@sps-lehrgang.de)

Erstellt am: 29.01.2015

Website besucht am: 10.12.2017

Link: <https://www.sps-lehrgang.de/sps-programmierung/>

A Anhang

A1 – Ausführliche Zeitplanung

Phase	Benötigte Zeit (in h)
Einarbeitung in das bisherige Programm	15
Einarbeitung in den alten Ablauf	9
Einarbeitung in die Visualisierung	6
Abstimmung mit der Treiberbaustein Entwicklung	5
Festlegen der Parameter für die Schnittstelle	5
Programmierung des neuen Ablaufs	22
Optimierung des alten Quellcodes	7
Entwerfen des neuen Ablaufs	6
Entwicklung des Algorithmus zur Berechnung der Ausrichtung	2
Implementierung der geplanten Änderungen	7
Erweiterung der Visualisierung	14
Erstellen eines Entwurfes	4
Anpassen des Startseite-Bildschirms	2
Erstellen der neuen Parameterbildschirme	6
Anpassen der Sprachtabellen & der genutzten Variablen	2
Funktionalitätsprüfung mit Fehlerkorrektur & Optimierung	0
Dokumentation des Projekts	14
Summe	70

Tabelle 5 - Ausführliche Zeitplanung

A2 – Entwürfe der Benutzeroberfläche

←

→

Parameterkategorie A

Parameter 1

Einstellung 1

Parameter 2

Einstellung 2

...

...

Parameter n

Einstellung n

Parameterkategorie B

Parameter 1

Einstellung 1

Parameter 2

Einstellung 2

...

...

Parameter n

Einstellung n

Programmnummer

Wert 1 .. *

Programmbeschreibung

Bearbeiten

Speichern

-

1 .. *

Parametersatz

+

Reset

Zurück

Abbildung 1 - Entwurf der Parameterseite

Erweiterung des Ablaufs und der Visualisierung einer Prüfvorrichtung für Schraubklemmen

Phoenix Signalkette

Handschrauber V3.0

Parametersatz

- 1..* +

Hinweistext

Aktuelle Drehzahl	Wert	Aktuelles Drehmoment	Wert
Motortyp	Beschreibung	Ausrichtung (in °)	Wert
Programm	Programm-Beschreibung	Modus-Beschreibung	Schritt

☐ Motor ON/OFF
 ☐ Drehmomentgrenze erreicht

☐ Standby

Hardware Info

Deutsch Englisch Polnisch Türkisch

Edit

Abbildung 2 - Entwurf der Startseite

A3 – Screenshots der bisherigen Visualisierung

Drehmoment (Nm):

Drehmoment "Schlitzsuche"	0
Drehmoment "Schrauben"	0
Lösemoment	0
Anzugsmoment	0
Wendemoment	0
Bremsmoment	0

Parametersatz

- 0 +

Drehzahl (1/min):

Drehzahl "Schlitzsuche"	0
Drehzahl "Schliessen"	0
Drehzahl "Öffnen"	0
Drehzahl "langsam Schliessen"	0
Wartezeit	0

Sonstiges:

Motor Typ	0	=	0: Maxon EC 45 flat 30W	Windungszahl	0
Programm	0	=	0: Stop		

♦ = Parameter wird durch das ausgewählte Programm benutzt

Bearbeiten Reset Zurück

Abbildung 3 - Der bisherige Parameter-Bildschirm

PHENIX CONTACT

INSPIRING INNOVATIONS

Handschrauber V2.0

Parametersatz:

-

0

+

Aktuelle Drehzahl	0	Aktuelles Drehmoment	0
Motor Typ	0: Maxon EC 45 flat 30W	Max. Drehmoment	0
Programm	0: Stop		0

●

Motor ON / OFF

●

Drehmoment Grenze erreicht

●

Standby

Hardware Info









Edit

Abbildung 4 - Die bisherige Startseite

A4 – Bilder der Prüfvorrichtung



Abbildung 5 - Der Handschrauber

A5 – Programmablaufpläne

Pruef_Hand-Baustein

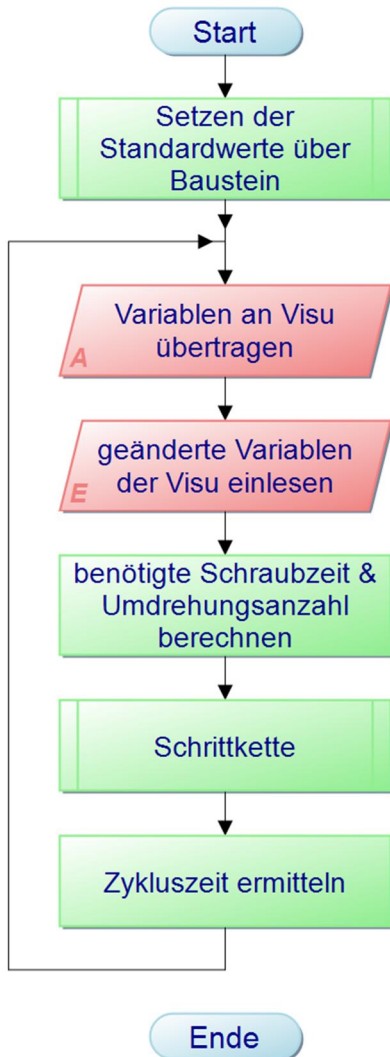


Abbildung 6 - PAP für den Ablauf-Baustein

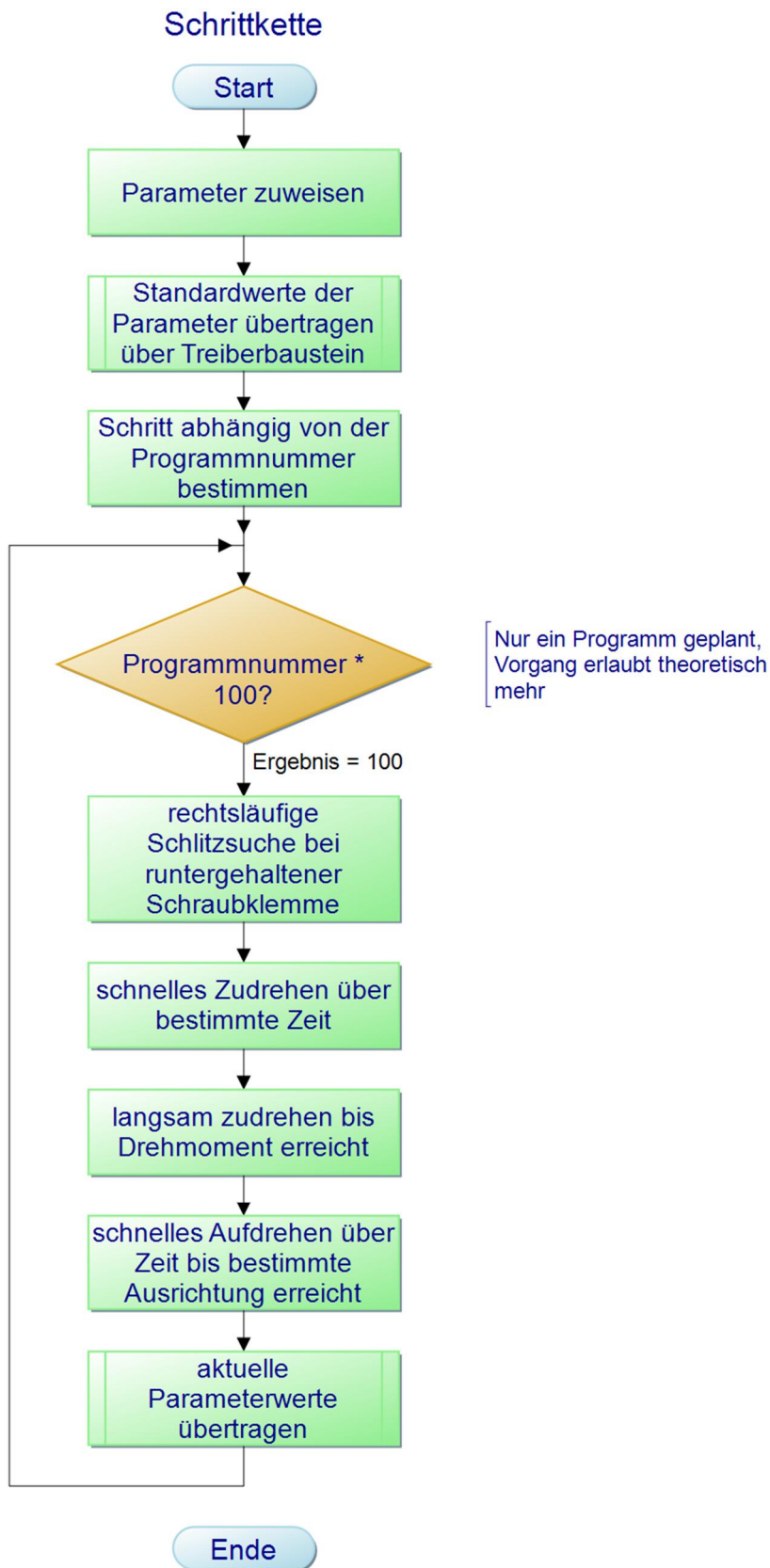


Abbildung 7 - PAP für die Schrittfolge

A6 – Programmcode

A6.1 – „ParameterSettings“-Baustein

```
(* Parameter 0 ist immer "AUS" !*)
(* In Parameter 0 wird kein Motor zugewiesen da dieser vom letzten gelaufenen
Parameter übernommen wird *)
(* Wenn über den Wahlschalter Parameter 0 angewählt wird, gehen die Bedienrechte an
die VISU *)
arrParameter[0].iSpeedFindSlot           := 0;
arrParameter[0].rTorqueFindSlot          := 0.0;
arrParameter[0].iSpeedScrewClose         := 0;
arrParameter[0].rTorqueScrew              := 0.0;
arrParameter[0].iSpeedScrewOpen           := 0;
arrParameter[0].rTorqueBreakOff           := 0.0;
arrParameter[0].rTorqueTurn               := 0.0;
arrParameter[0].rTorqueTighten            := 0.0;
arrParameter[0].rTorqueBreak              := 0.0;
arrParameter[0].rNumberOfTurn             := 0.0;
arrParameter[0].iSpeedScrewCloseSlow      := 0;
arrParameter[0].iProgammmNo               := 0;
arrParameter[0].iWaitingTime              := 1000;

FOR i:=1 To 9 DO
    arrParameter[i].iSpeedFindSlot         := 400;
    arrParameter[i].rTorqueFindSlot         := 0.032;
    arrParameter[i].iSpeedScrewClose        := 2500;
    arrParameter[i].rTorqueScrew            := 0.050;
    arrParameter[i].iSpeedScrewOpen         := 2500;
    arrParameter[i].rTorqueBreakOff         := 0.2;
    arrParameter[i].rTorqueTurn              := 0.1;
    arrParameter[i].rTorqueTighten          := 0.05;
    arrParameter[i].rTorqueBreak            := 0.2;
    arrParameter[i].rNumberOfTurn           := 4.5;
    arrParameter[i].iSpeedScrewCloseSlow    := 280;
    arrParameter[i].iProgammmNo             := i;
    arrParameter[i].iMotorParameter        := 1;
    arrParameter[i].iWaitingTime            := 1000;
END_FOR;

FOR i:=10 To 50 DO
    arrParameter[i].iSpeedFindSlot          := 0;
    arrParameter[i].rTorqueFindSlot          := 0.0;
    arrParameter[i].iSpeedScrewClose         := 0;
    arrParameter[i].rTorqueScrew             := 0.0;
    arrParameter[i].iSpeedScrewOpen          := 0;
    arrParameter[i].rTorqueBreakOff          := 0.0;
    arrParameter[i].rTorqueTurn              := 0.0;
    arrParameter[i].rTorqueTighten           := 0.0;
    arrParameter[i].rTorqueBreak             := 0.0;
    arrParameter[i].rNumberOfTurn            := 0.0;
    arrParameter[i].iSpeedScrewCloseSlow     := 0;
    arrParameter[i].iProgammmNo              := 0;
    arrParameter[i].iMotorParameter          := 1;
    arrParameter[i].iWaitingTime             := 1000;
END_FOR;
```

```
(* Welcher Programmablauf benutzt welche Parameter? *)
(* Programm 1 benutzt folgende Parameter: *)
arrProgrammUses[1].xSpeedFindSlot           := true;
arrProgrammUses[1].xTorqueFindSlot          := true;
arrProgrammUses[1].xSpeedScrewClose        := true;
arrProgrammUses[1].xTorqueTurn              := true;
arrProgrammUses[1].xTorqueScrew            := true;
arrProgrammUses[1].xNumberOfTurn           := true;
arrProgrammUses[1].xSpeedScrewCloseSlow     := true;
arrProgrammUses[1].xTorqueTighten          := true;
arrProgrammUses[1].xTorqueBreak            := true;
arrProgrammUses[1].xSpeedScrewOpen         := true;
arrProgrammUses[1].xTorqueBreakOff         := true;
arrProgrammUses[1].xWaitingTime            := true;
```

A6.2 – “NumberOfTurns”-Baustein

```
(* Zeit seit der letzten Messung *)
diTime := PLC_SYS_TICK_CNT - OldTickCnt;

(* Ersten Zyklus abbrechen *)
if OldTickCnt = DINT#0 then
    OldTickCnt := PLC_SYS_TICK_CNT;
    return; (* Baustein abbrechen *)
end_if;

(* Speichern der aktuellen "Zeit" *)
OldTickCnt := PLC_SYS_TICK_CNT;

(* Nur Positiv *)
if IN_iSpeed < 0 then
    iSpeed := IN_iSpeed * -1;
else
    iSpeed := IN_iSpeed;
end_if;
rActualSpeed := INT_TO_REAL(iSpeed);

(* Durch 0 Teilen verhindern *)
if rActualSpeed = 0.0 then
    return; (* Baustein abbrechen *)
end_if;

(* Richtungswechsel setzt den Umdrehungszähler zurück *)
if IN_xDirection <> xDirection then
    OUT_rTurns := 0.0;
    xDirection := IN_xDirection;
end_if;

(* Anzahl der Umdrehungen die in einer ms gemacht werden *)
rUmdrehungenMS := rActualSpeed / 60.0 / 1000.0;
(* Umdrehungen die im letzten Zeitinterval gemacht wurden *)
rNewTurns := rUmdrehungenMS * DINT_TO_REAL(diTime);

OUT_rTurns := Out_rTurns + rNewTurns;
```

A6.3 – "TimeToScrew"-Baustein

```
rSetSpeed := INT_TO_REAL(IN_iSpeed);    (* Zeit zum schließen ausrechnen*)

if rSetSpeed = 0.0 then
  rSetSpeed := 0.001;
end_if;

rTime := 1000.0 / (rSetSpeed / 60.0) * IN_rNumberOfTurn;
diTime := REAL_TO_DINT(rTime);
OUT_tTime := DINT_TO_TIME(diTime);

rUmdrehungenMS := rSetSpeed / 60.0 / 1000.0;
rTest := IN_rNumberOfTurn / rUmdrehungenMS;
diTest := REAL_TO_DINT(rTest);
OUT_tTest := DINT_TO_TIME(diTest);
```

A6.4 – Ausschnitt aus dem "Pruef_Hand"-Baustein

```
(*-----
Programm: 1
Beschreibung: Test der Schraubvorrichtung mit Ende auf eingestellter Position
Funktion: - Rechtslauf zur Schlitzsuche bis das Drehmoment ansteigt wenn ein Teil
           drangehalten wird
           - Schnell schließen über Zeit
           - Langsam Schließen bis Drehmoment erreicht
           - Schnell aufschrauben über Zeit
           - Ende wenn Position erreicht
-----*)

100:  (* Rechtslauf Schlitz suchen *)
      iMode           := 2;
      xModeChange     := TRUE;
      xDirection      := TRUE;
      iSetpointSpeed  := arrParameter[iParameter].iSpeedFindSlot;
      rSetpointTorque := arrParameter[iParameter].rTorqueFindSlot;
      xScrew          := FALSE;
      xTorqueMaxReset := TRUE;
      iStep           := 105;
      xRun            := TRUE;

105:  (* Maximales Leerlaufdrehmoment erfassen um einen Autostart erkennen zu
können *)
      xSleep          := TRUE;
      iStep           := 110;

110:
      IF NOT xSleep THEN
        rTorqueDetectStart := rMaxTorque + 0.005;
        iStep              := 115;
      END_IF;

115:  (* Start des eigentlichen Programmablaufs *)
      IF PosFlanke_xStartTorque OR PosFlanke_xFoot THEN
        xStandbyReset := TRUE;
        iStep          := 120;
      END_IF;

120:  (* Drehmoment zuschrauben *)
      iSetpointSpeed := arrParameter[iParameter].iSpeedScrewClose;
      rSetpointTorque := arrParameter[iParameter].rTorqueScrew;
      xScrew          := FALSE;
      iStep           := 125;

125:  (* Nach 100ms Überwachung *)
      xScrew          := TRUE;
      iStep           := 130;
```

```

130:  (* geringe Drehzahl / Drehmoment *)
      IF NOT xScrew THEN
          xTorqueMaxReset      := TRUE;
          iSetpointSpeed       := arrParameter[iParameter].iSpeedScrewCloseSlow;
          rSetpointTorque      := arrParameter[iParameter].rTorqueTighten;
          iStep                := 135;
      END_IF;

135:  (* Drehzahl 0 --> Bremsen mit hohem Moment*)
      IF xTorqueLimit THEN
          iSetpointSpeed       := 0;
          rSetpointTorque      := arrParameter[iParameter].rTorqueBreak;
          iStep                := 140;
      END_IF;

140:  (* Motor steht *)
      IF iActSpeed > -50 THEN
          xDirection           := FALSE;
          iSetpointSpeed       := arrParameter[iParameter].iSpeedScrewOpen;
          rSetpointTorque      := arrParameter[iParameter].rTorqueBreakOff;
          iStep                := 145;
      END_IF;

145:  (* kleines Drehmoment --> Motor läuft wieder*)
      IF iActSpeed > 50 THEN
          iSetpointSpeed       := arrParameter[iParameter].iSpeedScrewOpen;
          rSetpointTorque      := arrParameter[iParameter].rTorqueScrew;
          xScrew               := FALSE;
          iStep                := 150;
      END_IF;

150:  (* Aufschrauben bis bestimmte Position *)
      iMode                    := 6;
      xModeChange              := TRUE;
      xScrew                   := TRUE;
      TimeToScrew_2(
          IN_iSpeed             := iActSpeed,
          IN_rNumberOfTurn      := (iAusrichtung / 360));
      tTimeToScrew             := TimeToScrew_2.OUT_tTime;
      iCounts                   := REAL_TO_INT((iAusrichtung/360)*iCountsProUmdrehung);
      iStep                     := 155;

155:
      IF NOT xScrew THEN
          iSetpointSpeed       := arrParameter[iParameter].iSpeedFindSlot;
          rSetpointTorque      := arrParameter[iParameter].rTorqueFindSlot;
          xSleep               := TRUE;
          iStep                := 160;
      END_IF;

160:  (* Mit hohem Drehmoment wenden*)
      IF NOT xFoot AND NOT xSleep THEN
          xDirection           := TRUE;
          iStep                := 115;
      END_IF;

```


**IHK**Industrie- und Handelskammer
Lippe zu Detmold

Diese Erklärung ist der Dokumentation anzuhängen!

1. Erklärung des Prüfungsteilnehmers / der Prüfungsteilnehmerin

Ich versichere durch meine Unterschrift, dass ich das Projekt und die dazugehörige Dokumentation selbständig und ohne fremde Hilfe angefertigt und alle Stellen, die wörtlich oder annähernd wörtlich aus Veröffentlichungen entnommen habe, als solche kenntlich gemacht habe. Die Arbeit hat in dieser Form keiner anderen Prüfungsinstitution vorgelegen.

Ich nehme zur Kenntnis, dass die Projektarbeit bei Täuschungshandlungen bzw. Ordnungsverstößen mit **null Punkten** bewertet wird. Die Prüfung gilt damit als nicht bestanden.

Blomberg, 12.12.17

Ort, Datum

J.-P. Fölsch

Unterschrift des Prüfungsteilnehmers

2. Erklärung des Ausbildungs- bzw. Praktikumsbetriebes

Wir versichern, dass das Projekt wie in der Dokumentation dargestellt in unserem Unternehmen realisiert worden ist.

PHOENIX CONTACT GmbH & Co. KG

12.12.17 Schieder

Ort, Datum

K. WenzStempel und Unterschriften des
Ausbildungs-/ Praktikumsbetriebes

**IHK**Industrie- und Handelskammer
Lippe zu Detmold**Abschlussprüfung:**Sommer
Winter 2017/2018Sachbearbeiter: Sebastian Prange
Telefon: 05231 760113, Fax: 7601-8013
E-Mail: prange@detmold.ihk.de**Antrag für die betriebliche Projektarbeit**Ausbildungs- bzw. Praktikumsbetrieb (mit Anschrift):Phoenix Contact GmbH & Co. KG
Flachmarktstraße 8
32825 BlombergAntragsteller/in (mit Anschrift):Jan-Philipp Töberg
Vahlhauser Straße 35a
32825 BlombergBerufsbezeichnung/Fachrichtung:

Fachinformatiker für Anwendungsentwicklung

Projektbezeichnung:

Erweiterung des Ablaufs und der Visualisierung einer Prüfvorrichtung für Schraubklemmen

Durchführungszeitraum:

10.10.2017 - 12.12.2017 immer Dienstags

Projektverantwortliche/r im Ausbildungsbetrieb:Viktor Wölk
Telefon: +49 5235 3-48534 Fax: +49 5235 3-41820
E-mail: vwoelk@phoenixcontact.comDer Ausbildungs- bzw. Praktikumsbetrieb erklärt sein Einverständnis zur Durchführung des Projektes. Nach Zulassung des Projektes sind geplante Änderungen zwecks Genehmigung der IHK vorab zu melden.

PHOENIX CONTACT GmbH & Co. KG

Schieder 22.08.17 Wölk

Ort, Datum

Stempel / Unterschrift

Der/Die Prüfungsbewerber/in bescheinigt, die Projektarbeit selbständig anzufertigen:

Schieder, 21.08.17, J.-P. Töberg

Ort, Datum, Unterschrift

Projektgenehmigung

wird von der IHK ausgefüllt!

Projektantrag wird genehmigt:

Die Dokumentation ist in 3-facher Ausfertigung am nächsten Arbeitstag nach Fertigstellung (Ende des persönlichen Durchführungszeitraumes) bei der IHK einzureichen.



Die in der Stellungnahme genannten Nachbesserungen sind zu berücksichtigen (s. Anlage)!

Projektantrag muss geändert bzw. nachgebessert werden (s. Anlage):

Der geänderte bzw. nachgebesserte Projektantrag ist bis zum _____ bei der IHK einzureichen (Änderungen bzw. Nachbesserungen sind nur zu markierten Punkten erforderlich).

Projektantrag wird nicht genehmigt (s. Anlage):

Der neue Projektantrag ist bis zum _____ bei der IHK einzureichen.

Detmold, 25. Sep. 2017

INDUSTRIE- UND HANDELSKAMMER
LIPPE ZU DETMOLD

i.d. Kasse



**IHK**Industrie- und Handelskammer
Lippe zu Detmold**Kurze Projektbeschreibung:**

Im Unternehmen wird eine Prüfvorrichtung für Schraubklemmen eingesetzt, um die Funktionalität von Schrauben in Klemmblocks zu überprüfen, bevor diese ausgeliefert werden. Diese Prüfvorrichtung soll nun so erweitert werden, dass der Nutzer vor dem Prüfvorgang die Ausrichtung des Schraubenkopfes einstellen kann. Diese Einstellung soll zusammen mit weiteren Parametern in einer Visualisierung konfiguriert werden und über das Programm an den Treiber-Baustein weitergegeben werden. Dazu soll auf Basis des bisherigen Programms ein neuer Ablauf, welcher die Ausrichtung der Schraube übernimmt, mit dazugehöriger Visualisierung und einer passenden Schnittstelle zum Treiber-Baustein programmiert/erstellt werden. Die Entwicklung des Treiber-Bausteins ist Teil eines eigenen Projektes, mit welchem die Eigenschaften der Schnittstelle abgestimmt werden müssen.

Bei nicht ausreichendem Platz bitte zusätzliches Blatt beifügen

Projektfeld:

Die Prüfvorrichtung wird aktuell im Unternehmen für Prüfvorgänge bei Klemmblocks eingesetzt. Dazu wird der Gewindegang und das Drehmoment überprüft, indem die Schraube einmal komplett hinein und wieder heraus gedreht wird, wodurch ein einwandfreies Verhalten der Schraubklemme festgestellt werden kann. Dieser Vorgang wird aktuell über eine Visualisierung gestartet und gestoppt, welche ebenfalls die Einstellung gewisser Parameter ermöglicht.

Projektphasen mit Zeitplanung in Stunden:

- 1 - Einarbeitung in das bisherige Programm (10h)
- 2 - Abstimmung mit der Treiber-Baustein Entwicklung (5h)
- 2 - Programmierung des neuen Ablaufs (20h)
- 3 - Erweiterung der Visualisierung (20h)
- 4 - Funktionalitätsprüfung mit Fehlerkorrektur und Optimierung (5h)
- 5 - Dokumentation des Projekts (10h)

Geplante Dokumentation zur Projektarbeit:

Projektdokumentation
Ausschnitte des Programmcodes und der Visualisierung
Bilder der Prüfvorrichtung

Unterstrichene Positionen sind Dokumentationen, die nicht von mir erstellt werden, die aber der Klarheit und zum Gesamtverständnis des Projektes dienen.

evtl. Ausschnitte aus genutzten Bibliotheken/Programmabschnitten des vorherigen Programms

Geplante Hilfsmittel zur Präsentation (zutreffendes bitte markieren):Flipchart ☐Tageslichtprojektor ☐Pinwand ☐Andere Präsentationsmittel: Laptop & Beamer

(sind vom Prüfungsteilnehmer funktionsfähig mitzubringen)