Fachhochschule Merseburg

Bachelor-Studiengang Maschinenbau/Mechatronik/Physiktechnik

Erstellung und Programmierung eines Interfaces zur optimalen Kommunikation und der grafischen Darstellung wichtiger Informationen.

Matrikel 12 BMMP-M

von

Joachim Hähnel

geb. am 22.03.1989

in Leipzig

Matrikel-Nr. 19582

Verantwortlicher Hochschullehrer:	Prof. Dr.-Ing. Rolf Kademann
Bachelorarbeitsbetreuer:	Christian Kratz M.Eng.
Betreuendes Unternehmen:	FRAIMTEC Automation & Anlagenmontage GmbH

Selbstständigkeitserklärung

Ich versichere wahrheitsgemäß, die Bachelorarbeit selbstständig angefertigt, alle benutzten Hilfsmittel vollständig und genau angegeben und alles kenntlich gemacht zu haben, was aus Arbeiten anderer unverändert oder mit Abänderungen entnommen wurde.

……..

Merseburg, den Verfasser

Inhaltsverzeichnis

Abkürzungsverzeichnis

Abb.	Abbildung
Gl.	Gleichung
GmbH	Gesellschaft mit beschränkter Haftung
o. Ä.	oder Ähnliches
HMI	Human Machine Interface
SPS	Speicherprogrammierbare Steuerung
TCP	Tool Center Point
HTML	Hypertext markup Language

Tabellenverzeichnis

Abbildungsverzeichnis

1 Einleitung

1.1 Einleitende Worte

Die industrielle Herstellung von Konsum- und Produktionsgütern hat sich in den vergangenen Jahren stark zu automatisierten Prozessen entwickelt. Dabei ersetzt die Robotertechnik heute vollständig die Arbeitskraft Mensch. Die notwendigen Aufgaben wie Handling, Bearbeitung und Qualitätskontrollen innerhalb einer automatisierten Produktionstrecke werden von Robotern übernommen. Menschliche Arbeitskraft wird heute lediglich zur Bedienung der Anlagen benötigt. Dies geschieht über eine grafische Bedienoberfläche, mit der der Benutzer den Roboter bedienen, programmieren oder sich Informationen über diesen beschaffen kann. Durch die Vielzahl von Herstellern auf diesem Gebiet sind die Bedienoberflächen mit einer Fülle von Funktionen und Informationen überhäuft, sodass eine Vereinfachung der Bedienung und Programmierung verstärkt von Roboterprogrammierern nachgefragt wird. Damit bietet diese Nachfrage eine Grundlage für die Weiterentwicklung einer speziellen Bedienoberfläche für den ausgewählten Hersteller *Fanuc Corporation.*

1.2 Motivation

Um das Studium Maschinenbau/Mechatronik an der Hochschule Merseburg erfolgreich abzuschließen und meinen Abschluss als Bachelor of Engineering zu erhalten, habe ich die Plicht eine Bachelor Thesis zu bearbeiten. In dieser werde ich alle erlernten Kenntnisse und Fähigkeiten, die ich während des Studiums erworben habe verwenden, um eine wissenschaftliche beziehungsweise technische Arbeit zu erstellen.

Die Bachelorarbeit wird den thematischen Inhalt der industriellen Robotik umreißen, einen Bereich der in mir selbst großes Interesse weckt. Nach dem ich während meines Studiums als Instandsetzungshelfer in einer Automobilzulieferungsfirma gearbeitet habe und ich dort schon mit Industrierobotern in Berührung kam, war es mir ein großes Bedürfnis mehr über die Roboter-Branche zu erfahren. Im Juli dieses Jahres absolvierte ich ein

freiwilliges Praktikum in dem in Magdeburg ansässigen Unternehmen *FRAIMTEC – Automation & Anlagenmontage GmbH*. Die Firma widmet sich der Roboterprogrammierung meist im Bereich der Automobilbranche. In diesem Praktikum durfte ich den Beruf des Roboterprogrammierers kennenlernen und Projekte in Unternehmen wie *Skòda* oder *John Deere* begleiten.

Diese wissenschaftliche Arbeit wird nun auch mit Unterstützung der Firma *FRAiMTEC GmbH* von mir bearbeitet und angefertigt. Das Thema befasst sich mit einem internen Projekt des Unternehmens, welches ich in Eigenregie bearbeiten soll.

1.3 Beteiligte Unternehmen

FRAIMTEC – Automation & Anlagenmontage GmbH

Die Firma *FRAIMTEC*, mit Sitz in der Landeshauptstadt von Sachsen-Anhalt in Magdeburg, verhalf mir zu einem vier wöchigen Praktikum, welches ich im Juli 2016 absolvierte. So konnte ich viel über die Firma lernen und nun auch meine Abschlussarbeit mit deren Unterstützung anfertigen.

Abbildung 1.1 Logo der Firma FRAIMTEC GmbH (1)

„Gegründet wurde das noch recht junge Unternehmen im Jahre 2012 von dem Roboterprogrammierer Michael Jürgen als Geschäftsführer und dessen Vater Norbert Jürgen, der die Firma als Prokurist vertritt. Mit seiner Erfahrung in der Branche der Anlagenmontage trägt Norbert Jürgen die Verantwortung für das wichtigste Standbein der Fa. FRAIMTEC GmbH, die Montage und Demontage

großindustrieller Stahlbauten. Mit etwa 20 Monteuren ist dieser Bereich im Unternehmen stärker besetzt, als die Abteilung der Roboterprogrammierung.“ (2)

Bis Anfang des Jahres 2015 war Micheal Jürgen als alleiniger Roboterprogrammierer eingestellt. Heute sind zwei Mitarbeiter in diesem Bereich tätig.

„Von Magdeburg aus, wo die Firma FRAIMTEC GmbH ein Büro sowie eine Werkstatt zur Vorbereitung von Montage-Arbeiten unterhält, werden Projekte in ganz Deutschland, aber auch über die Staatsgrenzen hinaus durchgeführt. Insbesondere Vertreter der fertigenden Industrie, Automobilproduzenten, entsprechende Zulieferer, Chemiebetriebe oder auch Anlagenbauer gehören zum Kundenkreis der Fa. FRAIMTEC GmbH. Insbesondere bei Projekten der Abteilung Roboterprogrammierung tritt die Fa. FRAIMTEC GmbH häufig als dienstleistender Dritter auf, wobei die gesamtheitliche Projektleitung von anderen Unternehmen verantwortet wird.“ (2)

2 Aufgabenstellung

Die Aufgabe, die ich mir im Rahmen dieser wissenschaftlichen Arbeit gestellt habe, beinhaltet die Programmierung und die visuelle Präsentation eines Human Machine Interfaces, kurz HMI, zur besseren Darstellung und Bedienung seitens des Benutzers. Die Robotertypen, die hierfür in Frage kommen, sind FANUC Roboter der neueren Generation wie in Abbildung 2.1 dargestellt mit Farbdisplay auf dem *i Pendant*©.

Abbildung 2.1 i Pendant FANUC (4)

In Abbildung 2.2 zu sehen ist beispielhaft der 6 Achs-Roboter *FANUC R-2000* der für viele Fertigungsaufgaben verwendet wird.

Abbildung 2.2 FANUC R-2000 (3)

Ziel meiner Aufgabenstellung ist es, neben der Erleichterung der Bedienung durch den Endbenutzer auch die aktuellsten Statusinformationen des Roboters visuell darzustellen, um eine schnellere Fehleranalyse oder eine Eingabe seitens des Benutzers zu ermöglichen.

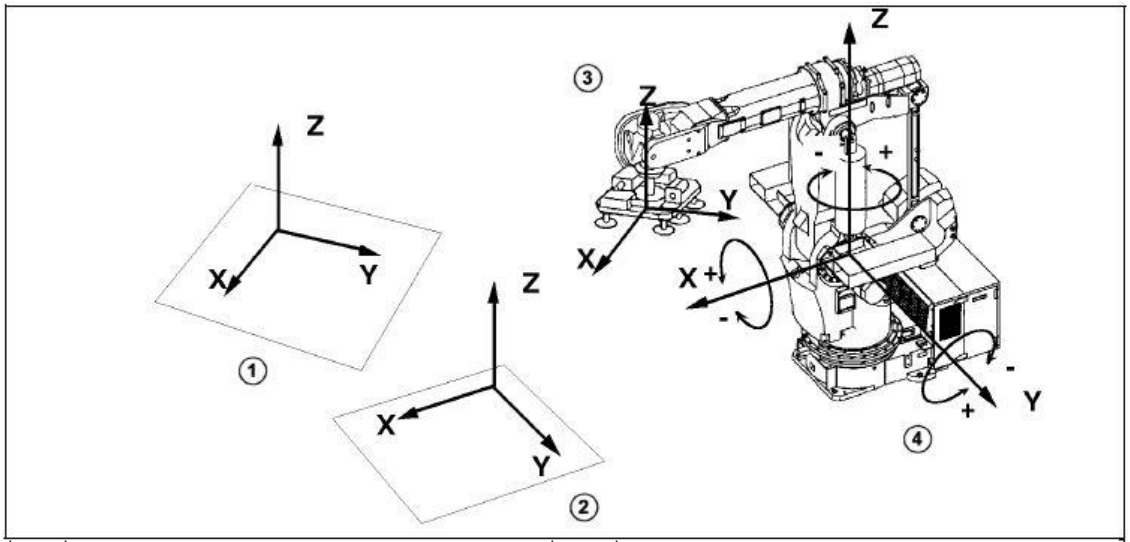
Zur Hauptaufgabe dabei gehört die Programmierung einiger KAREL Grundprogramme und den dazugehörigen HTML Programmtexten um mithilfe des Browsers im Roboterbedienpanel eine visuelle Darstellung zu ermöglichen. Dabei ist eine Ausarbeitung einer Strategie nötig um die erstellten Programme mit der Robotergrundsteuerung kommunizieren zu lassen.

Die weitere Aufgabe besteht darin, die erstellten Programme zu testen und im Praxisversuch zu verwenden um eventuelle Fehler zu erkennen. Eine Analyse der Nutzbarkeit und der Sinnhaftigkeit einiger Elemente ist durchzuführen und auszuwerten. Insbesondere die Doppelung einiger Statusinformationen, die schon die anlagenspezifische SPS anzeigt, sollte vermieden werden.

3 Theoretische Grundlagen

Die Definition der Industrieroboter wurde vom *Verein Deutscher Ingenieure (VDI)* festgelegt und als „universell einsetzbare Bewegungsautomaten mit mehreren Achsen, deren Bewegungen hinsichtlich Bewegungsfolge und Wegen […] frei programmierbar und gegebenenfalls sensorgeführt sind. Sie sind mit Greifern, Werkzeugen oder anderen Fertigungsmitteln ausrüstbar und können Handhabungs- und/oder Fertigungsaufgaben ausführen.“ (13)

Im Normalfall besitzen solche Roboter sechs Achsen, welche von speziell angesteuerten Schrittmotoren angetrieben werden. Ebenso besitzt der klassische Industrieroboter sechs Freiheitsgrade. Dadurch ergibt sich die Erreichbarkeit jedes Punktes innerhalb des Arbeitsbereiches, des am Roboterflansch befestigten Werkzeugs.



1	Benutzer-Koordinatensystem 2	3	Werkzeug-Koordinatensystem
2	Benutzer-Koordinatensystem 1	4	Welt-Koordinatensystem

Abbildung 3.1 Schema Roboterachsen (5)

Die bekanntesten Hersteller im Bereich Industrierobotik sind, *ABB Robotics* aus der Schweiz, *FANUC* mit Sitz in Japan, oder der deutsche Hersteller *KUKA* mit Sitz in Augsburg.

Für die automatisierte Arbeitsweise eines Roboters sind nicht nur elektronische Bauteile wie Aktorik oder Sensorik notwendig, sondern weitere elektromechanische Anbauteile. Dazu gehören unter anderem ein

Schaltschrank, ein Flansch, Kabellage und das schon genannte Bedienpanel. Die Abbildung 3.2 stellt schematisch diesen Aufbau dar. Erst die Gesamtheit aller Komponenten ermöglicht eine Bewegung und Bedienung des Roboters. Geschultes Fachpersonal kann mit vom Hersteller vorgegebenen Programmiersprachen einen Bewegungsablauf bestimmen und ändern. Weiterhin ist eine Kommunikation über definierte Schnittstellen zu anderen Steuerungen und Peripherien möglich.

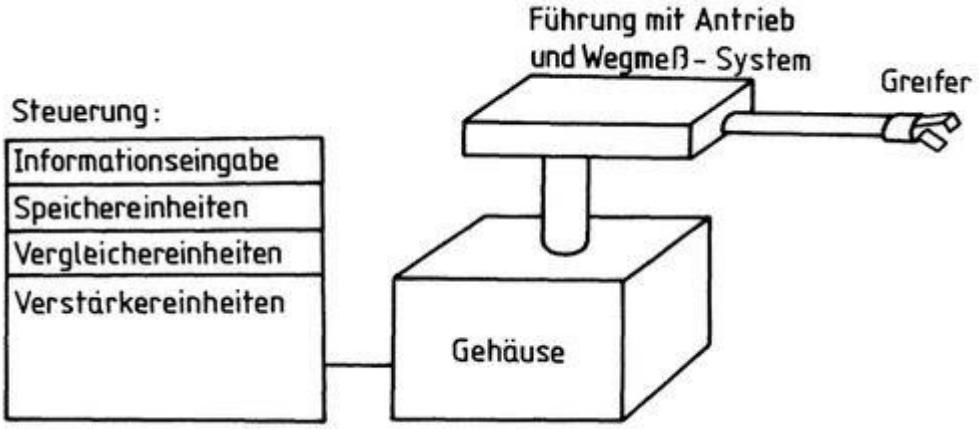


Abbildung 3.2 Hauptbaugruppen Industrieroboter (7)

Das Kommunikationsmittel innerhalb dieser Schnittstellen ist die jeweilige Programmiersprache. Davon gibt es Vielfältige, die von der Struktur und dem Ablauf stark variieren können, aber über festgelegte Befehle einer externen Steuerung miteinander arbeiten können. Im weiteren Verlauf unter Punkt 7.1 werden ausgewählte Programmiersprachen genauer erklärt.

Neben der Programmierung am Roboter selber, ist auch die virtuelle Bedienung und Simulation am Computer oder Laptop nötig. Dafür gibt es eine Vielzahl von Anwendungssoftware. In Zusammenhang mit dieser Arbeit kommt die von *FANUC Corporation* entwickelte Software *ROBOGUIDE* zum Einsatz.
ROBOGUIDE bietet die Möglichkeit virtuelle Roboterzellen zu erstellen, beziehungsweise aus einem Backup eines bestehenden Roboters eine Zelle virtuell samt dem Bewegungsablauf zu generieren. Die Software unterstützt den Programmierer beim Einstellen einiger Systemfunktionen des Robotersystems, wie beispielsweise eine Kollisionsprüfung mit der Umgebung, eine Anpassung von Datenregistern oder Systemvariablen.

Die Standardoberfläche sieht wie folgt aus:

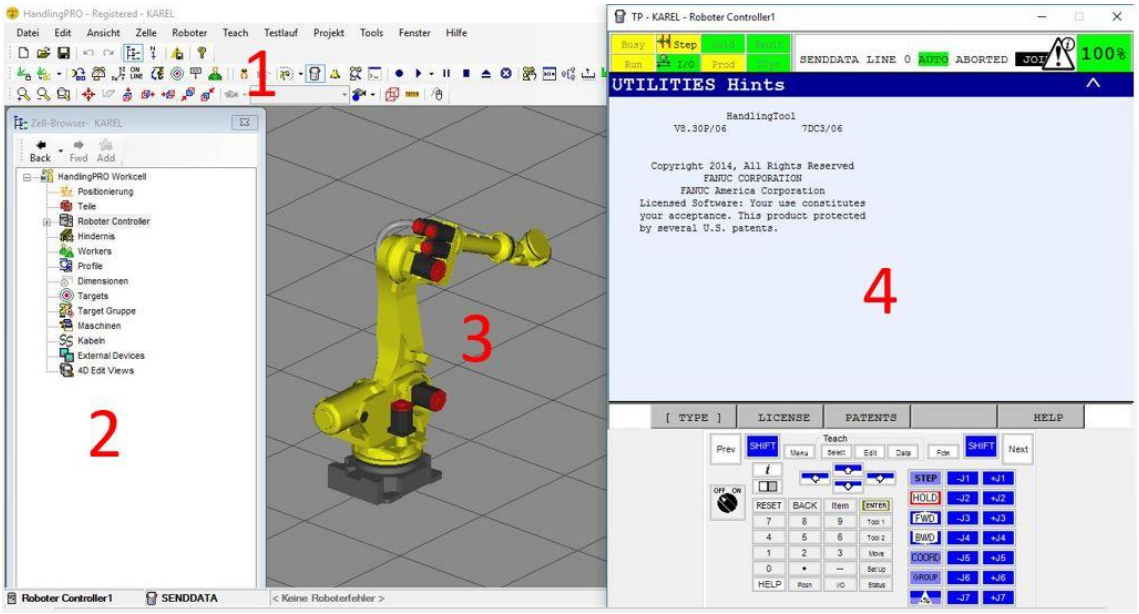


Abbildung 3.3 Benutzeroberfläche ROBOGUIDE

Die Symbolleiste **1** enthält viele Funktionen wie z.B. die Ansicht wechseln oder zur virtuellen Steuerung des Roboters. Auf der linken Seite befindet sich der Zellbrowser **2** welcher alle auf dem Systemspeicher des Roboters befindlichen Programme und Variablen enthält und anzeigt. Alle Programme und virtuelle Funktionen sind anpassbar und veränderbar. In der Mitte ist der Ausschnitt des virtuellen Arbeitsraums **3** zu sehen. Hier können die Spacefunctions angezeigt werden, die als Quader beziehungsweise als Sphären oder anderen geometrischen Figuren dargestellt und angepasst werden können. Die Ansicht kann individuell angepasst werden um eine optimale Sicht auf alle Funktionen zu gewährleisten, wie zum Beispiel die Verfahrwege der einzelnen Achsen und die Lage der Koordinatensysteme der Userframes und der Toolframes. Spacefunction beschreibt die Räume, in die, wenn der Roboter in den festgelegten X/Y/Z Bereich einfährt, ein definiertes Ausgangsignal oder Eingangssignal ausschaltet. Um den Roboter herum sind die Bereiche, die wenn sie sich während der Bewegung außerhalb des vorher festgelegten Zellenraums befinden, die Dual Check Safety kurz DCS auslösen und den Roboter zum sofortigen Stillstand bringen.

Um eine virtuelle Darstellung des Hand Panels im Programm zu gewährleisten, wird, wie hier das Teach Pendant **4** vom Roboter angezeigt, indem man die gleiche Ansicht hat wie in der Realität. Hier kann man die gleichen Funktionen nutzen wie auch in der physischen Version.

Die Benutzeroberfläche ist die beste Möglichkeit, eine Roboterzelle virtuell zu erstellen und so alle Funktionen und Bewegungsabläufe vorab zu beschreiben und sich anzuschauen. Jegliche Funktionsparameter können dann für die Wirklichkeit angepasst werden. Die Simulation kann so für den kompletten Arbeitsablauf für die Roboterzelle in der realen Welt genutzt werden.

4 Stand der Technik

4.1 Roboterbauarten und ihre Aufgabenbereiche

Roboter werden für unterschiedliche Aufgaben eingesetzt. Um alle diese Aufgabenbereiche abzudecken, wurde eine Vielzahl unterschiedlicher Roboterarten entwickelt. Sie unterscheiden sich in ihrem Aufbau, wie auch in ihren Fähigkeiten. Im industriellen Bereich werden Standartmäßig 5 Robotertypen eingesetzt, welche ich im Folgenden nach ihrem mechanischen Aufbau und dem daraus resultierenden Bewegungsraum unterteilen möchte.

1. Schwenkarm Roboter / SCARA:

„Besteht in der Regel aus vier Achsen, normalerweise aus Dreh-achsen. Sie sind für gewöhnlich durch einen nierenförmigen Bewegungsbereich gekennzeichnet.“ (12)

2. Portal / Linear Roboter:

„Besteht aus linearen Schubachsen die auf einer Ebene liegen, so dass sich die Bewegungsrichtungen vor und zurück sowie von links nach rechts ergeben.“ (12)

3. Zylindrische Roboter :

„Die Gelenke sind so angeordnet, dass sich ein Zylinderförmiger Bewegungsraum ergibt, in dem Objekte gegriffen werden können.“ (12)

4. Sphärische Roboter:

„Ähnlicher Aufbau wie beim zylindrischen Roboter, nur das der Bewegungsraum beliebige Bewegungen im Radius einer Kugel ermöglicht.“ (12)

5.Knickarmroboter:

„besitzt häufig 6 Freiheitsgrade oder mehr: Die Gelenke können je nach Aufgabenbereich und benötigter Anzahl an Gelenken individuell aufgebaut werden.“ (12)

Zusätzlich zu diesen 5 Roboterbauarten gibt es auch noch Parallelroboter mit hexapodischem Antrieb, welche eine Mikropositionierung ermöglicht. (6)

Bei dem in dieser Arbeit verwendetem und angesprochen Roboter handelt es sich um ein *Fanuc* Roboter. Diese Firma stelle überwiegend nur die Knickarmroboter her mit sechs Achsen. Alle weiteren Achsen die verwendet werden sind meisten externe Linearachsen. Farblich hat jeder Roboterhersteller eine andere Lackierung. Bei *Fanuc* ist diese ein spezielles Gelb und erinnert an die Farbe Rapsgelb. Der Farbcode für dieses Gelb wird aber unter Verschluss gehalten.

Aktuelle *Fanuc* Roboter sind unteranderem die Baureihen der M und R Serie. Wie z.B. der R-2000iB.(3) Diese Sechs-Achs-Roboter sind laut Aussage der vorangegangen Quelle ein „*Alleskönner*“ auf dem Gebiet der Robotertechnik. Diese Roboter werden aber ausschließlich für den industriellen Einsatz gebaut und sind keine mobilen Roboter. Für Handlingsaufgaben sind diese Knickarmroboter aber die am weitesten verbreitetsten und auch vom Payload die besten auf Ihrem Gebiet. Mit Payload ist das Gewicht gemeint, welches ein Roboter an der 6. Achse als maximales Gewicht tragen kann.

4.2 Marktanalyse bestehender Mensch-Maschine-Schnittstellen

Allgemein gibt es unzählige verschiedene HMIs, von haptischen über akustischen und visuellen. Die genauere Erklärung des HMIs wird im Abschnitt 5. Erläuterung Mensch Maschine Schnittstelle behandelt. Im Bereich der Industrieroboter und deren Steuerung werden hauptsächlich visuelle beziehungsweise grafische Benutzerschnittstellen mit Eingabemöglichkeiten über Knöpfe verwendet. Die älteren mechanischen HMIs werden kaum noch verwendet heutzutage. Als Beispiel für *Fanuc* Roboter ist im Abschnitt (2. Aufgabenstellung) ein *i Pendant* dargestellt. Diese sind genau wie jegliche Technik, seit den Anfängen stetig weiterentwickelt worden. Die ersten *i Pendants* hatten ein Schwarz-weiß Bildschirm. Der Nachfolger war mit Farbdisplay ausgestattet, aber erst die neueste Version besitzt ein

Farbbildschirm mit Touch Funktion. Für mein Projekt lassen sich nur die neuesten Modelle nutzen, da diese über ein Touchpad verfügen und so eine Eingabe per Fingerdruck realisieren lässt. In meinem Projekt sind so nur die *iPendant* Typen R-30iA und R-30iB als Hardware zu verwenden. Um eine detaillierte Analyse zu erstellen, ist eine Aufschlüsselung nach Angebot und Nachfrage nötig. Zurzeit gibst es keine mir bekannte Mensch Maschine Schnittstelle, welche die benötigte Nutzerfreundlichkeit an den Tag legt. Um eine Einschätzung abgeben zu können, wie hoch die Nachfrage nach einer Vereinfachung der Bedienung ist, wurden einige Nutzer von *Fanuc* Robotern befragt. Da dazu aber keine belegbare Dokumentation vorliegt, gehe ich anhand der Aussagen davon aus, dass hier eine hohe Nachfrage bei den Benutzern vorliegt. Um eine Aussage über die Sinnhaftigkeit des Projektes zu treffen, werde ich im letzten Punkt dieser Arbeit eine Abschlussanalyse durchführen. Ich habe bei diesem Projekt ein Konzept ausgewählt unter diversen Bewertungskriterien, welches im Punkt 6 näher erläutert wird.

5 Erläuterung Mensch Maschine Schnittstelle

5.1 Arten von HMIs und deren Definition

5.1.1 Variante 1: Mechanische HMI

Schon in früheren Zeiten mit Beginn der Industriellen Revolution wurden Mensch-Maschine Schnittstellen gebaut, um die Bedienung und die Manipulation der Maschinenbewegung/Steuerung zu gewährleisten. Dazu zählen unter anderem ganz normale Hebel, die z.B. zum Starten oder zum Stoppen der Maschine zuständig sind. Wie im folgenden Bild zu sehen ist verfügt eine alte Dampflok zur Steuerung und Bedienung lediglich Ventile und Schieberegler und Druckanzeigen.

Abbildung 5.1 Bedienung einer Dampflok (8)

Die Maschinen wurden immer komplexer und immer mehr Hebel/Schalter/Knöpfe wurden integriert. Um die Bedienung zu vereinfachen, wurden Elektronische Schaltungen entwickelt. Darauf gehe ich im nächsten Abschnitt ein. Da die sich mit Entwicklung der Technik auch immer mehr die englische Sprache durchsetzte, spricht man nun von einem Human-Machine-Interface kurz HMI.

5.1.2 Variante 2: Elektronische/Elektromechanische HMI

Um die immer weiter steigende Komplexität der Maschinen zu vereinfachen wurden elektromechanische HMIs entwickelt und in die vorhandenen Maschinen integriert. Diese bestehen meist aus Tastern oder Schaltern die mithilfe einer elektronischen Schaltung dahinter, eine Reaktion der Maschine auslösen. Meistens sind die Schalter direkt mit Motoren oder anderen mechanischen Manipulatoren verbunden. Mit Entwicklung der Computertechnik und der damit zusammenhängenden Programmierung einzelner Programme wurde die Ansteuerung der mechanischen Manipulatoren immer mehr vom eigentlichen HMI abgekoppelt. Heutzutage und in Zukunft werden immer öfter rein elektrische HMIs verwendet, da die Maschine seine Ansteuerung selbst übernimmt. Laut dem Internetlexikon IT Wissen (9) wird ein HMI als eine Benutzerschnittstelle beschrieben, welche überall dort zu finden ist, wo ein Menü auf einem Display dargestellt wird und so ein Dialog zwischen Mensch und Maschine erfolgt. Dies bewirkt eine Interaktion. In den frühen Jahren der Computertechnik waren die Benutzereingaben textorientiert und sehr speziell, was eine erschwerte Kommunikation bedeutete. Um solche Probleme zu beheben wurden visualisierte Schnittstellen entwickelt, die durch Sprach oder Gesteneingaben erweitert wurden und werden. Da die Art des HMIs immer auf den jeweiligen Fall angepasst werden, gibt es viele Varianten die Ein und Ausgabe von Informationen zu gewährleisten.

Abbildung 5.2 Abbildung eines Elektrischen HMIs (10)

6 Konzeption der Bedienoberfläche

6.1 Konzept 1

Zunächst dient eine vereinfachte Skizze der geplanten Bedienoberfläche zur Orientierung. Abbildung 6.1 zeigt schematisch die Anordnung der einzelnen Funktionen.

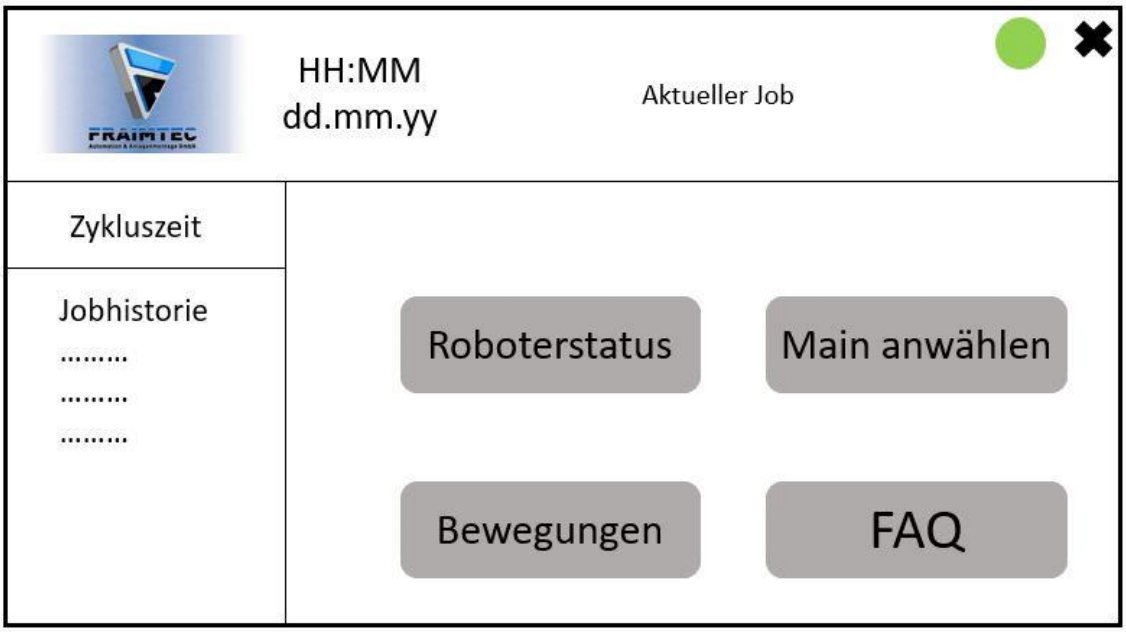


Abbildung 6.1 Skizze Bedienoberfläche

Der Header im oberen Bereich ist immer derselbe, egal wie oft man innerhalb der unteren Fenster durch die Menus klickt. Neben dem firmeneigenen Logo wird die systeminterne Uhrzeit beziehungsweise das aktuelle Datum angezeigt. Jeder Roboter bekommt von einer externen Steuerung, in den meisten Fällen eine Speicherprogrammierbare Steuerung, eine Jobnummer geschickt. Diese wird als Integerwert eingelesen und wird dann im Programm verarbeitet. Der Benutzer kann dann anhand der im Header angezeigten Jobnummer ablesen, welche Aufgabe der Roboter gerade ausführt. Oben rechts als grüner Punkt dargestellt ist das Livebit des Roboters. Dieser abwechselnd grün und rot blinkende Punkt soll den Benutzer darüber informieren, ob der Roboter auch betriebsbereit ist. Auf der linken Seite zu sehen, ist eine Übersicht über die aktuelle Zykluszeit. Das bedeutet, wie lange z.B. ein Handlingsroboter benötigt für einen kompletten Durchlauf durch alle Programme. Darunter ist eine Jobhistorieliste angelegt, welche dem Benutzer mitteilt, was für Jobs der

Roboter vor dem aktuell zu bearbeitenden Job durchgeführt hat. Den größten Bereich nimmt das Fenster zur Auswahl einiger Funktionen ein. Darunter sind z.B. „Main Anwählen“. Diese Funktion bricht alle gerade pausierten Programme ab und versetzt den Programmzeiger in das Hauptprogramm, welches in fast allen Fällen mit „Main“ betitelt wird. Der „FAQ“ Button leitet den Benutzer auf eine Seite mit häufig gestellten Fragen weiter, woraus er dann per Link wählen kann. Der Button „Bewegungen“ leitet den Benutzer auf eine Seite weiter, auf der er den Roboter, wenn er nicht auf Automatiklauf gestellt ist, per Hand z.B. in eine vordefinierte Grundstellung fahren kann.

Um eine Bewertung des Konzeptes abzugeben ist eine Aufschlüsselung nach verschiedenen Gesichtspunkten nötig. In der folgenden Tabelle sind von mir ausgewählte Punkte mit einer Wertung von 1 (schlecht) bis 10 (sehr gut) versehen um einen Vergleich mit dem nachfolgenden Konzept zu ermöglichen.

Bewertungskriterien	Punktzahl
Benutzerfreundlichkeit	10
Programmieraufwand	5
Kosten/Nutzen	4
Intuitive Bedienung	8
Ausmaß des grafischen Feedbacks	10
Gesamtpunktzahl:	37

Tabelle 1 Bewertungskriterien Konzept 1

Die Auswertung der ausgewählten Bewertungskriterien wird im Absatz 6.3 dieser Bachelorarbeit erklärt und durchgeführt.

6.2 Konzept 2

Hinter dem zweiten Konzept steht eine zeilenbasierte Umsetzung des HMIs. Im Folgenden ist ein Beispiel zu sehen:

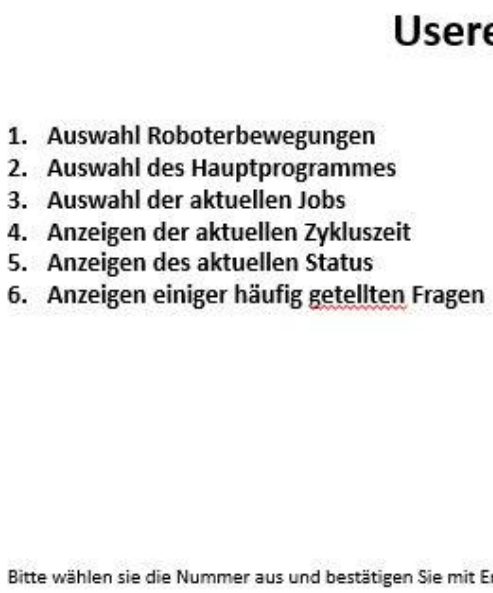


Abbildung 6.2 Skizze Bedienoberfläche zeilenbasiert

Hier wird das Hauptfenster dargestellt als Auswahlliste. Der Nutzer wird hier aufgefordert per Knopfdruck auf dem Bedienpanel eine Nummer einzugeben und mit der Taste „Enter“ zu bestätigen. Die Funktionen die hinter den Auswahlpunkten stehen, sind die gleichen wie schon im 1. Konzept beschrieben. Nur ist hier eine grafische Darstellung wichtiger Informationen nicht möglich beziehungsweise nicht vorgesehen. Um so schnell an wichtige Informationen für den Benutzer zu kommen, muss immer erst eine Auswahl per Knopfdruck getätigt werden. Je nach Auswahl des Benutzers werden immer wieder neue Unterpunkte beziehungsweise neue Seiten mit reinem Text ausgegeben, um so die benötigten Informationen wiederzugeben. Eine Eingabemöglichkeit per Touch Pad ist in diesem Fall auch nicht vorgesehen, da es keine grafischen Elemente wie Buttons oder Texteingabefelder gibt. Möglichkeiten der farblichen Gestaltung der Texte und Auswahlpunkte ist möglich, aber ist auch mit weiterem Programmieraufwand verbunden.

Um eine Bewertung des Konzeptes abzugeben ist eine Aufschlüsselung nach verschiedenen Gesichtspunkten nötig. In der folgenden Tabelle sind von mir ausgewählte Punkte mit einer Wertung von 1 (schlecht) bis 10 (sehr gut) versehen um einen Vergleich mit dem vorangegangenen Konzept zu ermöglichen. (siehe Seite 18)

Bewertungskriterien	Punktzahl
Benutzerfreundlichkeit	2
Programmieraufwand	8
Kosten/Nutzen	7
Intuitive Bedienung	4
Ausmaß des grafischen Feedbacks	1
Gesamtpunktzahl:	22

Tabelle 2 Bewertungskriterien Konzept 2

Die Auswertung der ausgewählten Bewertungskriterien wird im Absatz 6.3 dieser Bachelorarbeit erklärt und durchgeführt.

6.3 Vergleich und Auswahl des Konzepts

Um einen Vergleich der beiden Konzepte und eine Bewertung abzugeben ist es nötig, die Wertigkeiten der einzelnen Bewertungskriterien zu addieren und festzulegen welches Konzept somit sinnvoller und auch besser ist.

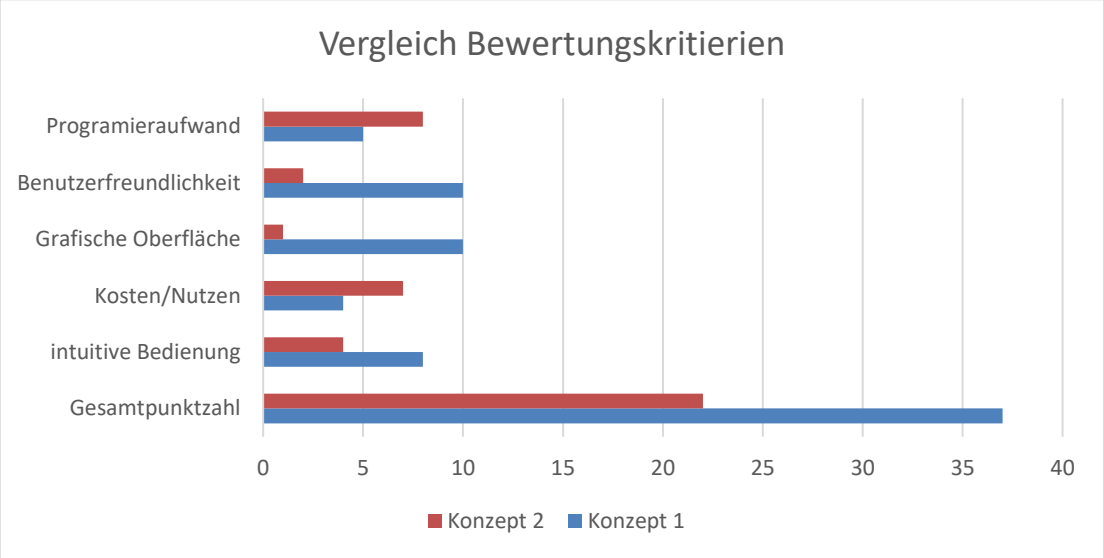


Abbildung 6.3 Diagramm Bewertungskriterien

Die Wertigkeit des ersten Konzeptes liegt bei **37.**

Die Wertigkeit des zweiten Konzeptes liegt bei **22.**

Das von mir ausgewählte Konzept ist das Erste. Zwar ist der Programmieraufwand höher und somit auch die Kosten, aber durch die intuitive Bedienung und die bessere grafische Oberfläche ist automatisch eine höhere Benutzerfreundlichkeit gegeben. Nutzbar ist das Konzept 1 allerdings nur auf den neueren *iPendants* von *FANUC,* da diese auch ein Touchpad besitzen. Das zweite Konzept ist zwar eine starke Vereinfachung, was den Programmieraufwand betrifft, fällt aber bei der Benutzerfreundlichkeit stark ab. Aufgrund der unübersichtlichen Auswahlpunkte und der schlechten Bedienbarkeit, ist das zweite Konzept zwar billig, aber nicht wirklich nutzbar und vereinfacht nur bedingt die Bedienung des Roboters seitens des Benutzers.

Im folgenden Abschnitt werden die verwendeten und entwickelten Programme erklärt und vorgestellt.

7 Beschreibung und Durchführung des Projektes

7.1 Aufbau Programmstruktur

7.1.1 TP Programmstruktur FANUC

Die Programmstruktur der Teach Pendant (TP) Programme besteht aus einer reinen zeilenbasierten Ablaufprogrammierung. Es gibt ein Programmzeiger den man in jede beliebige Zeile des Programms setzen kann. Schaltet man nun in den Automatikablauf des Roboters, beginnt das Programm genau an der Stelle wo man den Programmzeiger gesetzt hat. Wie in jeder zeilenbasierten Programmierung arbeitet das Programm jede Zeile nacheinander ab, bis der Programmzeiger durch Jump/Label Befehle oder Call Befehle in einer neue Zeile oder ein neues Unterprogramm versetzt wird. Ist in einer Zeile eine Abfrage eines digitalen Eingangs gesetzt, so bleibt der Zeiger solange in der Zeile Stehen, bis die vorher definierte Bedingung erfüllt ist. Auch werden hier alle bewegungssteuernden Befehle, mithilfe von Koordinaten, also einem Punkt im Raum, definiert um den Roboter zu bewegen. Man muss ein Hautprogramm erstellen, welches beim Automatikstart aufgerufen wird. Als Beispiel ist hier ein Programmausschnitt gezeigt.

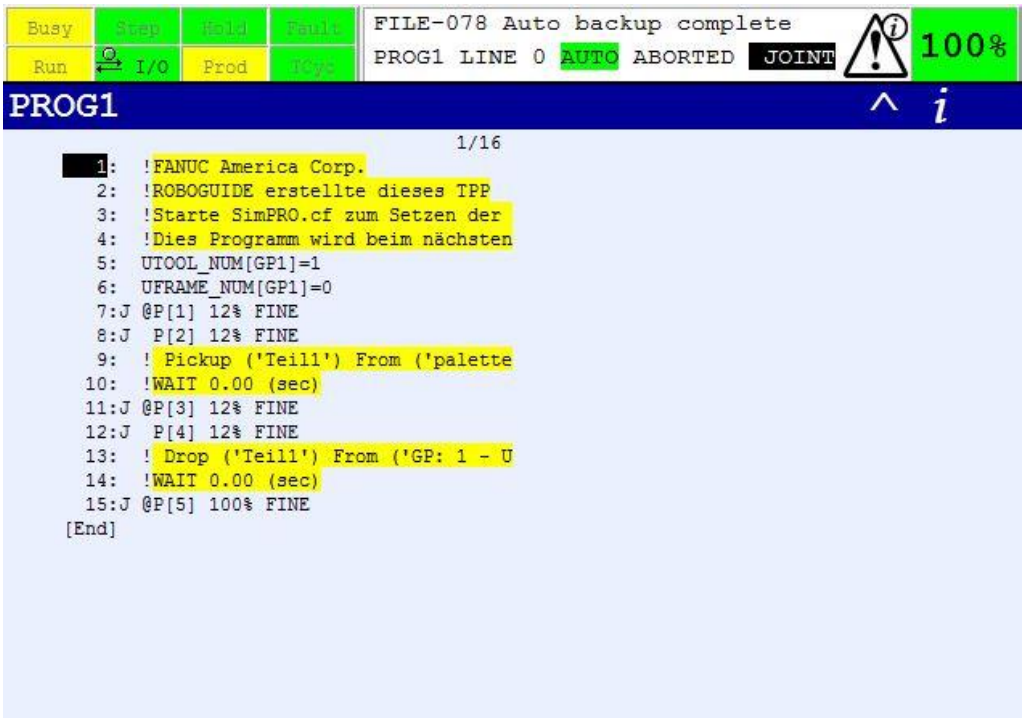


Abbildung 7.1 Programmstruktur TP

Wie in der Abbildung zu sehen, ist das Programm mit Zeilennummern versehen. Die erste Zeilennummer ist schwarz hinterlegt, was bedeutet, dass der Programmzeiger sich in der ersten Zeile befindet. Wenn jetzt ein Automatikstart folgen würde, würde das Programm in Zeile 1 gestartet werden. Die gelben Zeilen sind hier reine Kommentarzeilen und werden im Ablauf wie bei jeder Programmierung ignoriert und sind nur zur Information dargestellt. Die Zeilen mit einem großen „J“ sind hierbei festgelegte Positionen im Raum, die der Roboter mit einer Joint Bewegung erreichen soll. Es existieren 4 Bewegungsverfahren bei *Fanuc*.

Die Joint Bewegung: Dabei bewegt sich der Roboter mit seinem TCP (kurz für Tool Center Point), welcher sich je nach Werkzeug meistens an der Spitze des Werkzeugs befindet, auf einer nicht festgelegten Bahn. Alle Achsen werden so bewegt, wie der Roboter es am günstigsten berechnet hat.

Die Linear-Bewegung: Diese Bewegung beschreibt eine geradlinige Bewegung des TCPs. Wobei der Roboter alle Achsen so ansteuern muss, dass eine exakt gerade Linie bei der Bewegung zwischen dem vorherigen und dem zu erreichenden Punkt entstehen muss.

Die Circular-Bewegung: Bei dieser Kreisbewegung müssen drei Punkte im Raum festgelegt werden. Ein Startpunkt, ein Punkt auf der Kreisbahn und ein Endpunkt. Die Robotersteuerung berechnet sich aus diesen 3 Punkten nun eine Kreisbahn. Das Ergebnis ist eine kreisrunde Bewegung des TCPs.

Die Circular-Arc-Bewegung: Diese Konfiguration wird sehr selten verwendet, wird aber genutzt wenn eine Kreisbahn nicht mit Punkten sondern mit Winkeln angegeben wird. So ist eine exakte kreisförmige Bewegung des TCPs mit einem vordefinierten Winkel möglich.

7.1.2 Programmstruktur KAREL

KAREL ist eine untergeordnete Programmiersprache, ähnlich, beziehungsweise angelehnt an PASCAL. Zur Verfügung stehen hier Konstanten, Variablen, Funktionen und Routinen, was eine Programmierung für Hintergrundprogramme erheblich erleichtert. Mit dieser Sprache ist viel mehr möglich als mit der TP Struktur am Teach Pendant. KAREL ist eine zu übersetzende Sprache. Das heißt, sie muss erst von einem KAREL Source File (.kl) in den P-Code (.pc) übersetzt werden, bevor ein Programm in den Controller geladen werden kann. Um eigens programmierte Programme auf den Controller laden zu können, ist eine Installation des *Fanuc* eigenen Softwarepaketes notwendig. Da dies von den meisten Kunden nicht erwünscht ist, ist ein Anlegen eigener Routinen nicht möglich, beziehungsweise notwendig.

So ist ein KAREL Programm vom Grunde her aufgebaut:

- **PROGRAM** (Benennung erforderlich)

- Übersetzungsbedingungen (optional)

- **CONST**, **TYPE**, und/oder **VAR** Deklaration (optional)

- **ROUTINE** Deklaration (optional)

- **BEGIN**

- Ausführbare Befehle (optional)

- **END** (Benennung erforderlich genau wie oben bei Program)

- **ROUTINE** Deklaration (optional)

Als erstes muss das Programm einen Rahmen erhalten. In diesem Falle wird das Programm mit dem Syntax **PROGRAM** und **END** gestartet und beendet. Innerhalb des Programms werden Variablen, Konstanten und Routinen deklariert, welche immer wieder genutzt oder in andere KAREL Programme überragen werden können.

Als kurzes Beispiel ein Ausschnitt eines Karelprogramms.

```
041
042 --- Initialisierung ------------------------------------------------
043 ROUTINE init
044 BEGIN
045   bol_ReqMain = FALSE
046   bol_StartCyc = FALSE
047   bol_StopCyc = FALSE
048                                DELETE_FILE ('buttons.stm',FALSE,int_Status)
049                                DELETE_FILE ('Zykluszeit.stm',FALSE,int_Status)
050                                DELETE_FILE ('Winjobs.stm',FALSE,int_Status)
051   GET_TPE_PRM(2,int_DataType, int_Livebit, rel_dummy, str_dummy, int_Status)
052   IF int_DataType <> 1 THEN
053     rError
054   ENDIF
055   RUN_TASK ('BG_Status',0,FALSE,FALSE,0,int_status)
056     -- Rücksetzen von Variablen
057   FOR i=1 TO 5 DO
058     int_JobNr[i]=0
059   ENDFOR
060
061  ----schreiben Main programmstring------
062
063   GET_TPE_PRM(3,int_DataType, int_dummy, rel_dummy, str_main, int_Status)
064   IF int_DataType <> 3 THEN
065     rError
066   ENDIF
067
068
```

Abbildung 7.2 Ausschnitt eines KAREL Programms

Innerhalb einer Routine können andere Programme aufgerufen werden, die im Hintergrund ununterbrochen laufen. Somit ist die Erstellung und Verwendung einer Hintergrundlogik ohne Abhängigkeit vom eigentlich im Controller laufenden TP Programms möglich und auch sinnvoll.

Die Erklärung der verwendeten Programme wird im nächsten Abschnitt dieser Arbeit ausführlich dargelegt.

7.2 Projektdurchführung

Nach der Auswahl des Konzeptes, ist nun die Erstellung der benötigten Programme zu erledigen. Um eine grafische Darstellung überhaupt möglich zu machen, ist das Erstellen einer HTML Seite nötig. Da die Controller von *Fanuc* eine Browserfunktion besitzen, mit der man auch bei verbundenem Ethernet ins Internet gehen kann, ist eine Erstellung dieser HTML Seite die einzige Möglichkeit ein HMI zu erzeugen.

Als erstes wird eine Strategie gesucht, mit der eine Kommunikation zwischen dem Roboter und dem erstellten Programm möglich wird. Dazu ist eine Hierarchie der einzelnen Programmsprachen nötig.

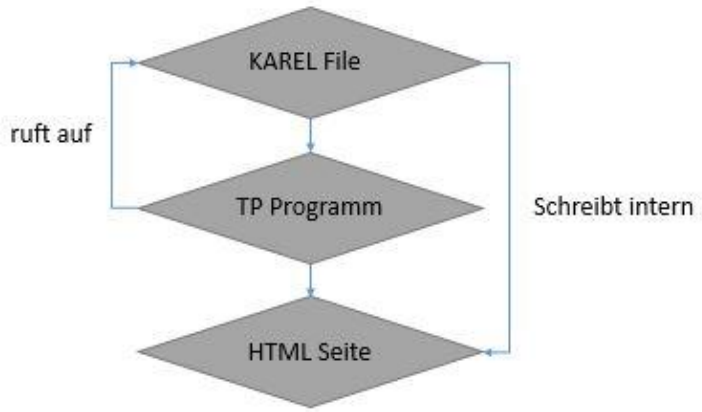


Abbildung 7.3 Hierarchie Programme

Auf dem Controller wird nun innerhalb eines Tp Programms das von mir erstellte KAREL Programm mit dem CALL Befehl ausgeführt. Dieses erstellt nun im Hintergrund eine HTML Seite und öffnet sie im Browserfenster des *iPendants.* Es besteht die Möglichkeit beim Aufrufen des KAREL Programms Argumente zu übergeben.

```
1:  CALL DISP_STATUS('init',5,'Main')
```

Das von mir erstellte Programm heißt DISP_STATUS. Das hier übergeben Argument „init“ wird als String übergeben und startet die Initialisierungsroutine im Programm.

Die übergebene Variable 5? beschreibt die verwendete Bit Nummer für das Livebit des Roboters. Dieses Bit wechselt immer von „On“ auf „Off“. Dieses Bit muss intern im Controller einer Nummer zugewiesen werden. Diese wird dann hier ins Programm übergeben. Das letzte Argument „Main“ wird auch als String übergeben und bezeichnet das verwendete Hauptprogramm, welches bei jedem Automatikstart des Roboters aufgerufen wird. Wenn ein anderer vordefinierter String als erstes Argument übergeben wird, wie zum Beispiel „AddJob“, dann wird eine andere Routine innerhalb des KAREL Programms ausgeführt.

```
   4:  CALL DISP_STATUS('AddJob',R[5:Job Nr])
```

So kann man flexibel das Programm mit mehreren Argumenten starten und je nach Bedarf damit eine andere Funktion aufrufen.

Das Programm DISP_STATUS.PC kann im Controller aber nicht verändert beziehungsweise geöffnet werden. Es kann nur aufgerufen werden. Um Änderungen vorzunehmen muss der KAREL File mit einem Editor geöffnet und nach Anpassung wieder in ein PC File übersetzt werden. Aufgrund der Komplexität des Quellcodes verzichte ich hier auf eine komplette Darstellung des von mir programmierten Programms.

Zur Verständlichkeit ist aber zu sagen, dass meine erstellte Anwendung aus 3 einzelnen Dateien besteht. Das Hauptprogramm DISP_STATUS verwaltet alle Routinen die nötig sind, um die Funktionen im erwünschten Endprodukt zu gewährleisten. Das Hintergrundprogramm BG_STATUS verwaltet die Verarbeitung des Livebits und die Berechnung der Zykluszeit. Diese Hintergrundlogik wird einmalig bei der Initialisierung aufgerufen und läuft in Dauerschleife ab. So kann die Anwendung bei jeder Änderung eines Status sofort aktualisiert werden und stellt somit die Informationen in Echtzeit dar. Ein HMI sollte eine grafische Oberfläche besitzen, deshalb wurde eine HTML Seite mit dem Namen HAUPTBILDSCHIRM.stm erstellt. In dieser Datei ist das Layout, also der Rahmen für die Benutzeroberfläche programmiert. Innerhalb dieser Datei wird mit „Frames“ gearbeitet. Das bedeutet, dass nur der Header feststeht. In den „Frames“ werden eigene HTML Seiten aufgerufen die innerhalb des KAREL Programms erzeugt werden.

Das Fertige Produkt sieht folgendermaßen aus:

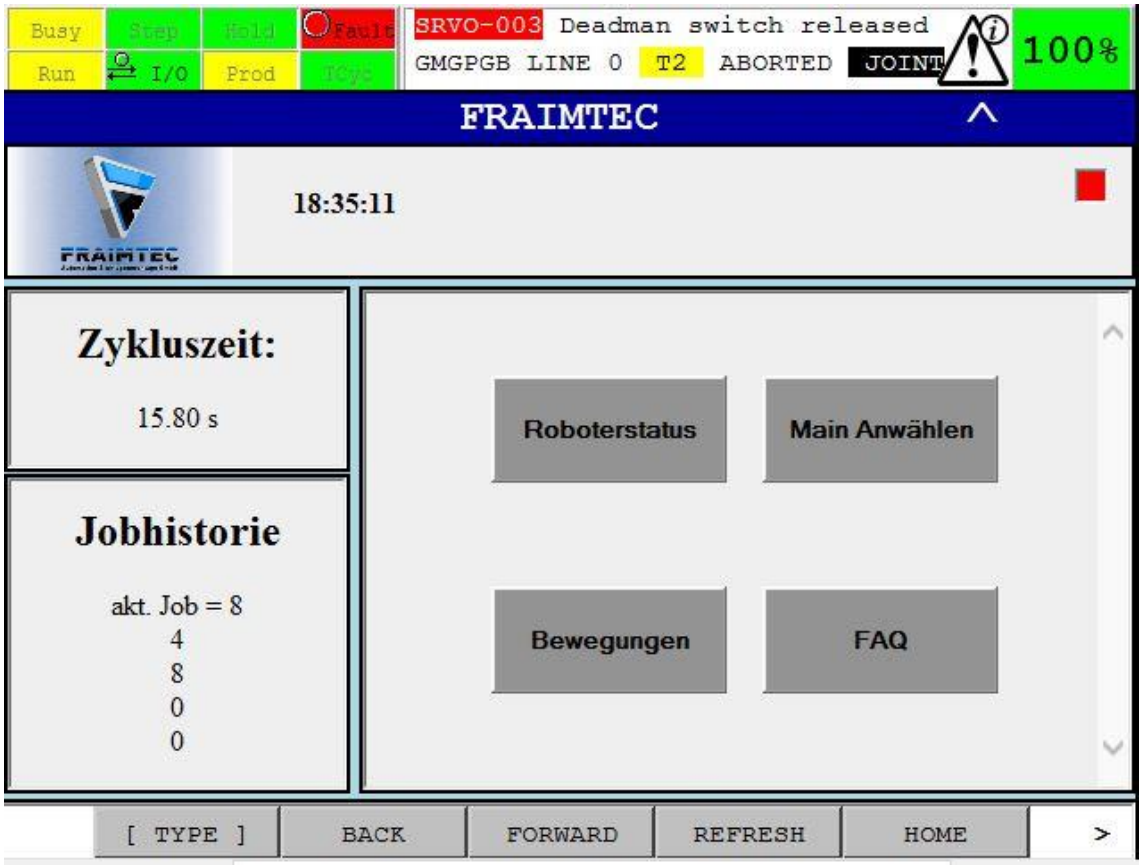


Abbildung 7.4 Ansicht des fertigen HMIs

Hier ist nun ein guter Vergleich mit der Skizze aus dem Konzept möglich. Die elementaren Strukturen und Anordnungen sind übernommen worden. Wie hier zu sehen sind die vorher genannten „Frames“ unterhalb des Headers als eigene HTML Seiten ausgeführt. Die Erklärung der einzelnen Funktionen wurde schon im Punkt 6.1 durchgeführt. Alle hier dargestellten Buttons sind per Touch Pad Eingabe anwählbar und ausführbar. Natürlich handelt es sich hier erstmal nur um einen Prototyp, welcher noch erweiterbar und anpassbar ist. Funktionstüchtig wurde das HMI erst nachdem ich alle vorbereiteten Dateien in den Controller geladen hatte. Mit dem Programm *Roboguide* kann man virtuell sich den Controller anzeigen lassen. So kann man das Programm im Vorfeld gut testen und anpassen.

8 Abschlussanalyse und Zusammenfassung

Inwieweit hat sich die Auswahl des Konzeptes auf die Programmierung und das Endprodukt ausgewirkt? Die grafische Umsetzung der im Punkt 6.1 gezeigten Skizze ist im Großen und Ganzen beim Endprodukt umgesetzt wurden.

Ziel war es eine benutzerfreundliche Oberfläche zu erstellen. Diese Vorgabe wurde erfüllt. Bei der Präsentation des fertigen Programms wurde als Ergebnis festgestellt, dass eine schnellere Auffassung der wichtigen Funktionen seitens der Benutzer gegeben war. Durch die vereinfachte Eingabe per Touch Pad konnte eine große Zeitersparnis bei der Problemlösung ermittelt werden. Mithilfe der FAQ-Funktion ist die Bindung einer Arbeitskraft, die eventuell den Benutzer über Problemlösung aufklären muss, nicht notwendig. Einarbeitungszeit und Schulungsaufwand für Bediener wurden somit auch verringert.

Laut eines Benutzers der Firma *Mondragon Assambly* ist die Verbesserung der Nutzerfreundlichkeit erheblich. *„Es ist viel einfacher zu überblicken, welche Funktionen am Roboter möglich sind, ohne sich durch die Programme wälzen zu müssen“* (11)

Zur Absicherung wurden einige Funktionen redundant ausgeführt. So ist zum Beispiel das Livebit des Roboters auf dem HMI der SPS dargestellt. Zur Kontrolle und zum Ausschluss eventueller Softwarefehler ist eine grafische Darstellung auf dem von mir erstellten HMI notwendig. Auch die Anzeige des aktuell zu bearbeitenden Jobs ist nicht nur von der SPS darzustellen, sondern auch auf der grafischen Benutzeroberfläche des Roboters. Die Begründung dafür ist, dass wenn eine Jobnummer als Integerwert von der SPS an den Roboter übergeben wird, noch nicht feststeht, ob das roboterinterne Programm diese auch eingelesen hat.

Die benötigte Speicherkapazität ist sehr gering im Verhältnis zu Programmen, die zum Ablauf der Jobs nötig sind. Die benötigten HTML Seiten werden in Echtzeit erstellt und bei Verlassen der Seite gelöscht. Damit wird die Speicherbelastung gering gehalten. Ein weiterer Vorteil des geringen

Speicherbedarfs ist, dass für eine Nachrüstung der Anwendung kein zusätzlicher Speicher gekauft beziehungsweise erweitert werden muss.

Aus wirtschaftlicher Sicht ist aufgrund der vorhergenannten Punkte eine Kostenersparnis beim Endkunden zu verzeichnen. Der einmalige Kauf des Produktes ist eine Investition, die vom Endkunden nur einmal getragen werden kann, so dass keine laufenden Kosten für diese Anwendung an der Anlage entstehen.

Als Nachteil zu verzeichnen ist die Gegebenheit, dass alle vorhanden TP Programme angepasst werden müssen. Das bedeutet, dass bei Inbetriebnahme und Installation der Anwendung ein Roboterprogrammierer sich vor Ort befinden muss. Dadurch bietet sich eine Ferninstallation über das Internet nicht an.

Die von mir erstellten Programme sind nur für den Roboterhersteller *Fanuc* und deren Controller zu verwenden, da die Möglichkeit dieses Programm von anderen Herstellern von Industrierobotern zu nutzen nicht gegeben ist. Da die Programmierung auf der Grundlage der Programmiersprache KAREL umgesetzt ist, ist ein einfaches Anpassen des Programmes an die Programmiersprachen anderer Hersteller nur mit kompletter Umprogrammierung möglich.

Intern betrachtet ist der Nutzen für die Firma sehr hoch. Zu sehen ist dieser Vorteil in dem Zeitaufwand für Bedienerschulungen, der sich nicht nur verringert, sondern auch die Verständlichkeit der wiedergegebenen Informationen lässt sich so verbessern. Das eingefügte Logo im feststehenden Header ist als Wiedererkennungswert der Firma *Fraimtec* zu sehen. Diese Visualisierung fördert die Kundenbindung und die Werbung.

Zusammenfassend zu sagen ist, dass eine neue Benutzeroberfläche für das Bedienpanel für *Fanuc* erstellt werden konnte. Dies ist nur ein Prototyp, welcher noch stetig weiterentwickelt und programmtechnisch mit einfachen Mitteln erneuert werden kann. Dieses Programm zeigt eine deutlich vereinfachte Bedienung und eine sichtlich verkürzte Einarbeitungszeit. Die gewählten Bewertungskriterien führten zu der Entscheidung für das Konzept 1. Konzept 1

hebt als priorisierte Wahl hervor, dass die grafische Darstellung ein höheres Niveau an Erkennbarkeit besitzt und des daraus resultierenden Feedbacks eine hohe Benutzerfreundlichkeit an den Tag legt.

Die Kommunikation von Programm zu Roboter wurde in mehreren Arbeitsschritten verwirklicht. Das HMI als solches konnte nicht in einem Programm beziehungsweise einer Programmiersprache geschrieben werden, sondern musste auf die verschiedenen Schnittstellen der von *Fanuc* verwendeten Programme angepasst werden. Dieses Problem wurde mit der Verwendung einer Programmhierarchie gelöst.

Im Hinblick auf die Erweiterung des Programms können folgende Verbesserungen gemacht werden. Die grafische Aufmachung ist aktuell etwas einfach gehalten. Diesen Umstand kann man zukünftig verbessern indem man das Design überarbeitet. Mit Entwicklung neuer Bedienpanels seitens *Fanuc* erhöht sich nicht nur die Auflösung, sondern es verbessert sich auch das Layout. Erweiterungsmöglichkeiten sind zum Beispiel, eine Bedienung per Spracheingabe oder eine haptische Rückmeldung. Möglich wäre auch eine Funktionserweiterung, entweder auf den Kundenwunsch angepasst oder wenn sich Neuerungen in der Bedienung der Robotersteuerung ergeben.

9 Summary

To what extent did the selection of the concept affect the programming and the final product? The graphical implementation of the sketch shown in point 6.1 is largely implemented in the final product.

The goal was to create a user-friendly interface. This requirement was done. When the final program was presented, it was set that there was a faster understanding of the important functions on the part of the users. Due to the simplified input via touch pad, a great time-saving in the problem solution could be determined. The FAQ function is not used to bind another worker. So the user can solve the problem on his own. The training time and the training effort for operators were thus also reduced.

According to a user of Mondragon Assambly, the improvement of user friendliness is significant. " *Es ist viel einfacher zu überblicken welche Funktionen am Roboter möglich sind, ohne sich durch die Programme wälzen zu müssen* " (11)

For protection, some functions have been implemented redundantly. For example, the robot's live bit is displayed on the HMI of the PLC. For the control and exclusion of possible software errors, a graphical representation is necessary on the HMI I created. The display of the current job is not only displayed by the PLC, but also on the graphical user interface of the robot. The reason for this is that if a job number is transferred as an integer value from the PLC to the robot, it is not yet determined whether the robot internal program has also read it in.

The required storage capacity is very small in relation to programs which are necessary for the execution of the jobs. The required HTML pages are created in real time and deleted when the page is exited. Thus the storage load is kept low. A further advantage of the low memory requirement is that no additional memory has to be purchased or extended for retrofitting the application.

From an economic point of view, the end customer has cost savings because of the aforementioned points. The one-time purchase of the product is an investment which can be borne by the end customer only once, so that no running costs for this application are created on the system.
The disadvantage is the fact that all existing TP programs must be adapted. This means that a robot programmer must be on site when commissioning and installing the application. This does not allow remote installation over the Internet.
The programs I have created are to be used only for the robot manufacturer Fanuc and their controllers, because the possibility of using this program by other manufacturers of industrial robots is not given. Since the programming is based on the programming language KAREL, it is only possible to easily adapt the program to the programming languages of other manufacturers with complete reprogramming.
Internally, the benefit to the company is very high. This advantage can be seen in the time expenditure for operator training, which not only reduces, but also improves the intelligibility of the reproduced information. The displayed logo in the fixed header can be seen as a recognition value of Fraimtec. This visualization promotes customer loyalty and advertising.

To summarize, a new user interface for the control panel for Fanuc could be created. This is only a prototype, which is still further developed and can be renewed programmatically by simple means. This program shows a considerably simplified operation and a visibly shorter processing time. The selected evaluation criteria led to the decision for concept 1. Concept 1 emphasizes as a prioritized choice that the graphical representation has a higher level of recognizability and shows the resulting feedback a high user-friendliness. The communication from program to robot was realized in several working steps. The HMI could not be written in one program or one programming language, but had to be adapted to the various interfaces of the programs used by Fanuc. This problem has been solved by using a program hierarchy.
The following improvements can be made with a view to the enlargement of the program. The graphical presentation is currently somewhat simple. This can be

improved in the future by reworking the design. With the development of new control panels by Fanuc, not only does the resolution increase, but the layout also improves. Extending possibilities would be, for example, an operation by means of speech input or a haptic feedback. A function extension would also be possible. Either adapted to the customer's request or when innovations arise in the operation of the robot control.

10 Literaturverzeichnis

(1) 1. **Lübbecke, Dominic.** *Logo.* [USB Flashspeicher] Magdeburg : FRAIMTEC GmbH, 2016.

(2) 2. **Bütof, Paul.** Bachelorarbeit. Merseburg : s.n., 2015.

(3) 3. **Corperation, FANUC America.** products: robots: R-2000. [Online] 2016. [Zitat vom: 06. September 2016.] http://www.fanucamerica.com/cmsmedia/images/tn/R-2000iB_165F_298.jpg.

(4) 4. **Roboworld Molded Products, LLC.** Pendant Armor: Home: Fanuc: A05B-2255: A05B-2255 Bumper. [Online] Resonate Marketing Group, 2016. [Zitat vom: 06. September 2016.] https://store.pendantarmor.com/images/Products/FANUC-A05B-2255-teach-pendant-bumper-protector-FRONT.jpg.

(5) 5. **FANUC Robotics Deutschland GmbH.** *Bedienungshandbuch Roboterserie R-J3iC.* [Portable Document Format] Neuhausen : s.n., 2007.

(6) 6. **Hesse, Stefan.** *Fertigungsautomatisierung: Automatisierungsmittel, Gestaltung und Funktion.* Braunschweig/Wiesbaden : Vieweg Verlagsgesellschaft, 2000. 978-3528039141.

(7) 7. **Raab, Horst H.** *Handbuch Industrieroboter 2. Auflage.* Braunschweig/Wiesbaden : Friedr. Vieweg & Sohn, 1986. 978-3-528-18481-0.

(8) 8. **Martin.** Die Technikfreunde. [Online] Rayk Web Design, 2008-2014. [Zitat vom: 29. September 2016.] http://technikfreun.de/unterwegs/bw-schoeneweide-eisenbahnfest-10/dampflok-regler.jpg.

(9) 9. **ITWissen.** ITWissen. [Online] DATACOM Buchverlag GmbH. [Zitat vom: 6. 10 2016.] http://www.itwissen.info/definition/lexikon/human-machine-interface-HMI-Mensch-Maschine-Schnittstelle.html.

(10) 10. **Weber, Jürgen.** Bonfiglio. [Online] Bonfiglio. [Zitat vom: 6. 10 2016.] http://www.bonfiglioli.de/media/filer_public/2c/5e/2c5e7162-0d79-4d72-9eee-c822d96b4ebd/bmi_it_503x340_3.jpg.

(11) 11. **Bozcan, Hr.** *Aussage Benutzer.* Stahringen : s.n., 2016.

(12) 12. **Schmiedecke, Christoph.** users.informatik.haw-hamburg.de. [Online] 2010. [Zitat vom: 8. 12 2016.] https://users.informatik.haw-hamburg.de/~ubicomp/projekte/master2010-aw2/schmiedecke/bericht.pdf.

(13) 13. **Weller, Rainer.** [Online] [Zitat vom: 11. 12 2016.] http://www.forum-sondermaschinenbau.de/seiten/Industrieroboter.htm.

Anhang