

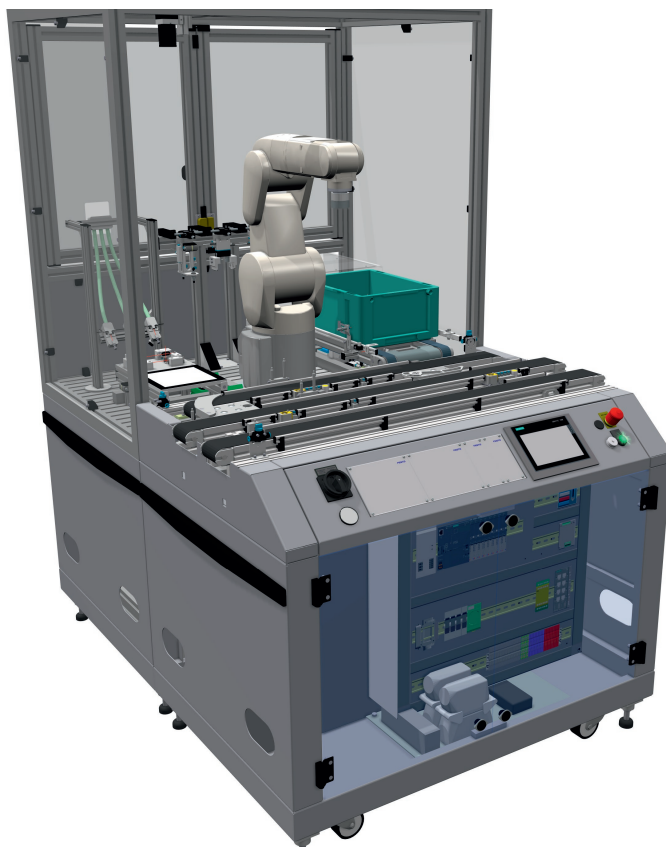
Robotertermontage

FESTO

Industrie 4.0



Arbeitsbuch



Bestell-Nr.: 8075234
Stand: 10/2017
Autor: Dr. Ulrich Karras
Grafik: Dr. Ulrich Karras
Layout: 01/2018, Dimitrios Tsakas

© Festo Didactic SE, Rechbergstraße 3, 73770 Denkendorf, Deutschland, 2017
Alle Rechte vorbehalten.



+49 711 3467-0

+49 711 34754-88500



www.festo-didactic.com

did@festo.com

Der Käufer erhält ein einfaches, nicht-ausschließliches, zeitlich unbeschränktes und geografisch nur auf die Nutzung innerhalb des Standortes/Sitz des Käufers beschränktes Nutzungsrecht wie folgt.

- Der Käufer ist berechtigt, die Inhalte des Werkes zur Fortbildung seiner Mitarbeiterinnen und Mitarbeiter, des Standortes zu nutzen und hierzu auch Teile der Inhalte zur Erstellung eigener Fortbildungsunterlagen zur Fortbildung seiner Mitarbeiterinnen und Mitarbeiter des Standortes unter Angabe der Quelle zu verwenden und für die Fortbildung am Standort zu kopieren. Bei Schulen/Hochschulen und Ausbildungsstätten umfasst das Nutzungsrecht auch die Nutzung für deren Schüler, Lehrgangsteilnehmer und Studenten des Standortes für den Unterricht.
- Ausgeschlossen ist in jedem Fall das Recht zur Veröffentlichung sowie zur Einstellung und Nutzung in Intranet- und Internet- sowie LMS-Plattformen und Datenbanken wie z. B. Moodle, die den Zugriff einer Vielzahl von Nutzern auch außerhalb des Standortes des Käufers ermöglichen.
- Weitere Rechte zu Weitergabe, Vervielfältigungen, Kopien, Bearbeitungen, Übersetzungen, Mikroverfilmungen sowie die Übertragung, Einspeicherung und Verarbeitung in elektronischen Systemen, unabhängig ob ganz oder in Teilen, bedürfen der vorherigen Zustimmung der Festo Didactic.

Hinweis

Soweit in diesem Arbeitsbuch nur von Lehrer, Schüler etc. die Rede ist, sind selbstverständlich auch Lehrerinnen, Schülerinnen etc. gemeint. Die Verwendung nur einer Geschlechtsform soll keine geschlechtsspezifische Benachteiligung sein, sondern dient nur der besseren Lesbarkeit und dem besseren Verständnis der Formulierungen.

Inhalt

Didaktische Umsetzung des Themas „Robotik“ im eigenen Unterricht	VI
Einordnung in den Kontext Industrie 4.0	VI
Allgemeine Voraussetzungen zum Betreiben der Geräte	XIV
Bestimmungsgemäße Verwendung	XIV
Arbeits- und Sicherheitshinweise	XVI
Informationen für Ausbilder/Lehrer und Auszubildende/Studenten zur Nutzung der Unterlage	XXI
Benötigte Komponenten	XXIII

Grundlagen

1	Industrieroboter	3
1.1	Portalroboter	3
1.2	SCARA Roboter	4
1.3	Vertikaler Knickarmroboter	6
1.4	Systemkomponenten eines Industrieroboters	8
1.5	Kenngrößen eines Industrieroboters	10
2	Greifer	14
2.1	Mechanische Greifer	14
2.2	Sauggreifer	16
2.3	Greiferwechselsysteme	18
3	Robotersteuerung	19
3.1	Kartesische Koordinatensysteme	19
3.2	Roboter-Koordinatensysteme	20
3.3	Steuerungsarten der Roboterbewegung	23
4	Sensoren	25
4.1	Induktive Näherungsschalter	25
4.2	Optische Näherungsschalter	26
4.3	Kamera	28
5	Programmierung von Industrieroboter	30
5.1	Online-Verfahren	30
5.2	Offline-Verfahren	31
5.3	CIROS Studio	32
5.4	Hinweise zur Roboterprogrammierung	37
5.4.1	Hindernis vermeiden	38
5.4.2	Beispiel: Werkstück greifen	38
5.4.3	Beispiel Paletten-Funktion	40
6	Sicherheitsmaßnahmen	42
6.1	Allgemeine Sicherheitshinweise	42

6.1.1	Betriebsanleitung _____	42
6.1.2	Verpflichtung des Betreibers _____	42
6.1.3	Verpflichtung des Bedienpersonals _____	42
6.2	Schutzeinrichtungen _____	42
6.2.1	Verriegelte trennende Schutzeinrichtung _____	43
7	Inbetriebnahme einer Roboteranwendung _____	43
7.1	Einleitung _____	43
7.2	Virtuelle Inbetriebnahme _____	43
7.3	Reale Inbetriebnahme der Roboteranwendung _____	43

Praktische Aufgaben

Einleitung	45
Aufgabe 1: Aufbau der CP Factory Station Robotertermontage	57
Aufgabe 2: Funktionsprüfung des Mitsubishi Roboters RV-4FL in der Montagestation	69
Aufgabe 3: Werkzeugwechselsystem für einfache Pick&Place Aufgabe nutzen	79
Aufgabe 4: Platine auf Werkstück montieren und Prüfung des Werkzeugwechselsystems	97
Aufgabe 5: Optische Positionsprüfung eines Werkstücks	113
Aufgabe 6: Anwendung der externen Funktionsbibliothek	125
Aufgabe 7: Montage einer Platine mit Sicherungen unter Anwendung der externen Funktionsbibliothek	131

Didaktische Umsetzung des Themas „Robotik“ im eigenen Unterricht

Einordnung in den Kontext Industrie 4.0

Industrie 4.0 gilt als Schlagwort für die Entwicklung digitaler Wertschöpfungsketten und zur Vernetzung derer Akteure. Der Begriff wurde durch eine Forschungsunion der deutschen Bundesregierung und ein gleichnamiges Projekt in der Hightech-Strategie der Bundesregierung geprägt. Er dient vor allem Marketingzwecken, um sich von dem US-amerikanischen Ansatz der „Cyber-Physical Systems“ abzugrenzen.

Allgemein wird unter Industrie 4.0 eine (erwartete) vierte industrielle Revolution verstanden, die nach der Dampfmaschine (erste Revolution), dem Fließband (zweite Revolution) und der Automatisierung (dritte Revolution) nun durch die intelligente Fabrik (Smart Factory) bestimmt wird. Im Kern geht es darum, bisher passive Bauelemente, Komponenten und Maschinen mit einer eigenen Intelligenz zu versehen und diese miteinander zu vernetzen. So sollen sich Informations- und Kommunikationstechnologie verzahnen und im Zusammenwirken eine neue Qualität bilden.

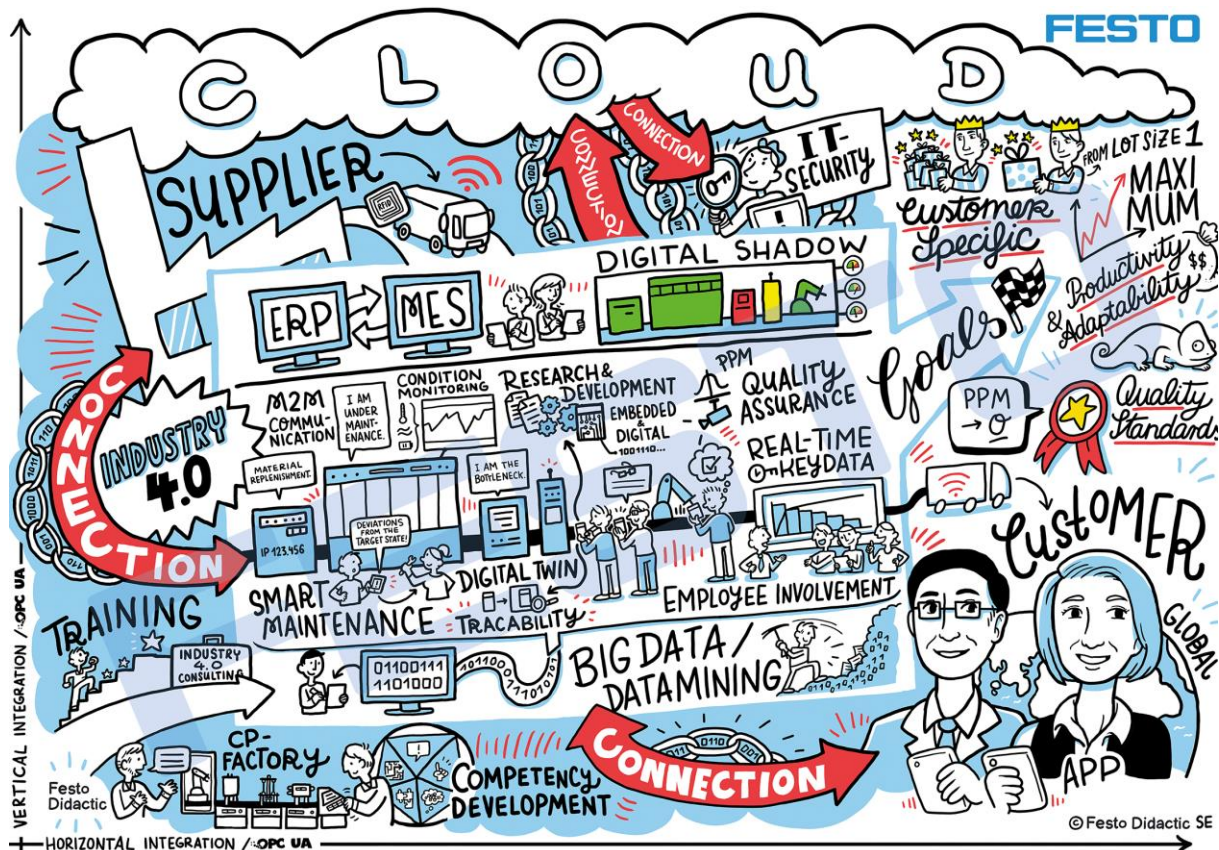
Industrie 4.0 umfasst damit einerseits eine Reihe neuer Technologien. Dazu gehören die selbstständige Kommunikation von Produkten oder Maschinenelementen mit höheren Steuerungssystemen über Radio Frequency Identification (RFID), Near Field Communication (NFC) oder Quick Response Code (QR-Code). Ebenso werden die automatische Verarbeitung von Daten mittels Enterprise Resource Planning (ERP), Manufacturing Execution Systems (MES) oder Supervisory control and data acquisition (SCADA) dazugezählt. Und schließlich sollen an dieser Stelle noch die adaptiven Fertigungstechnologien (3D Druck, Lasersintern, gentelligente Werkstoffe), die neuen Möglichkeiten der Robotik (kollaborierende Roboter, autonomes Fahren) und die Potentiale der Gentechnik hinzugefügt werden.

Andererseits zielt Industrie 4.0 auf die Anwendung der Technologien, also deren Zusammenwirken im Rahmen eines höheren Systems. Das wird auch als vertikale und horizontale Vernetzung bezeichnet. Erstere beschreibt die Verknüpfung von Maschinen- mit den Unternehmensdaten. Die zweite erstreckt sich über die gesamte Wertschöpfungskette und verbindet Zulieferer sowie Kunden mit dem eigenen Fertigungsprozess. Damit sollen im Wesentlichen die klassischen Ziele der produzierenden Industrie in sich schnell ändernden, volatilen Märkten noch besser erreicht werden. Diese Ziele umfassen natürlich Qualität sowie Kosten- und Zeiteffizienz, aber auch Ressourceneffizienz, Flexibilität, Wandlungsfähigkeit sowie Robustheit (oder Resilienz). Idealerweise soll es so möglich werden kundenindividuelle Produkte zu Kosten von Serienprodukten zu fertigen. Der Losgröße 1 kommt dabei eine wichtige Bedeutung zu.

Zusammengefasst lassen sich die Merkmale von Industrie 4.0 in Abgrenzung zur Industrie 3.0 in folgender Form zusammenfassen:

- Die Herstellung individualisierter Produkte unter den Bedingungen einer hoch flexiblen Großserienfertigung.
- Die Integration der Kunden, Lieferanten und anderer möglicher Geschäftspartner in die Wertschöpfungsprozesse.
- Die Kopplung von Produkten und Serviceleistungen zu neuartigen Geschäftsmodellen.

Eine schematische Darstellung zu den neuen Technologien und deren Zusammenspiel für verbesserte bzw. neue Anwendungen finden Sie in nachfolgender Darstellung.



Industrie 4.0: Das Zusammenspiel der Technologien

Industrie 4.0 erobert sich so einerseits Stück für Stück unserer Realität. Andererseits kann sie das nur und so tiefgreifend, wie wir sie gestalten, Potenziale erkennen, Nutzungsmöglichkeiten identifizieren und Anwendungen schaffen.

Dementsprechend sind Sie als Lehrer an beruflichen Schulen bzw. als Ausbilder in der Praxis gefordert. Denn trotz aller technologischen Entwicklungen bleibt der Mensch im Mittelpunkt. Nicht nur als Kunde, sondern ebenso als Gestalter. Der Mensch sorgt für die Integration der Technologien und deren Funktionsfähigkeit. Er übernimmt die Nutzung von neuen Anwendungen und wandelt deren Ergebnisse in eine höhere Qualität, störungsfreiere Prozesse und bessere betriebswirtschaftliche Ergebnisse.

Selbstverständlich wandelt sich dabei die Arbeit. Das betrifft vor allem drei Aspekte:

- Technologisch: Bisher bekannte Technologien werden durch neue ersetzt bzw. durch diese ergänzt.
- Kurzzyklisch: Die Lebenszyklen von Produkten und Technologien verkürzen sich fortwährend.
- Betrachtungsebene: Während bisher eher die Arbeit **im** System im Vordergrund steht, wird die durch die fortschreitende Automatisierung Stück für Stück durch die Arbeit **am** System verdrängt. Menschliche Arbeit rankt sich also in Zukunft weniger darum, Produkte in der richtigen Menge und in der richtigen Qualität herzustellen, als um die Optimierung und Anpassung der automatisierten Systeme, damit diese mit hoher Produktivität und Qualität fertigen.

Das erfordert vor allem die Entwicklung der Kompetenzen der Auszubildenden. Also deren Fähigkeiten, sich in offenen und unüberschaubaren, komplexen und dynamischen Situationen selbstorganisiert zurechtzufinden. Aus dem Ausprägungsgrad dieser Kompetenzen resultiert, wie die aktuellen Auszubildenden und damit späteren Mitarbeiter Signale zur Veränderung wahrnehmen und interpretieren, wie sie in der Lage sind, auftretende Probleme zu analysieren und dafür neue, noch unbekannte Lösungen zu erarbeiten, wie sie deren Wirksamkeit überprüfen und bei auftretenden Abweichungen zum angestrebten Zustand Korrekturen einleiten. Also auf Industrie 4.0 bezogen, wie sie in der Lage sind, Maschinen, Anlagen

und Produkte neuen Typs zu montieren und zu bedienen, einzurichten und instandzuhalten.

Vor diesem Hintergrund wurden aktuell sieben Arbeitsbücher entwickelt. Sie umfassen folgende Themen:

- Produktidentifikation mittels RFID
- Robotik
- Grundlagen der Netzwerktechnik und IT Security
- Manufacturing Execution Systems (MES)
- Advanced SPS Technik
- Human-Machine-Interaction (HMI)
- Smart Maintenance

Weitere Themen werden in den nächsten Monaten folgen.

Relevanz des Themas „Robotik“ zu ausgewählten Berufen

Das Thema Robotik ist vor allem für Ausbildungsberufe relevant, deren Absolventen im zukünftigen Arbeitsleben Roboteranlagen installieren, programmieren, in Betrieb nehmen und einrichten. Das betrifft vorrangig den Mechatroniker, den Elektroniker für Automatisierungstechnik und den Industriemechaniker. In den nachfolgenden Tabellen sind auszugsweise die Rahmenlehrpläne und Ausbildungsrahmenpläne der erwähnten Berufe und deren Bezug zum Thema Robotik dargestellt.

Mechatroniker/Mechatronikerin

Modul	Inhalte des Moduls bzw. der Qualifikation
Rahmenlehrplan: Realisieren mechatronischer Teilsysteme	Steuerkette und Regelkreis, Blockschaltbilder Kenngrößen von Steuerungen und Regelungen Wirkungsweise von Sensoren und Wandlern Signalverhalten von Sensoren und Wandlern Programmierung von einfachen Bewegungsabläufen und Steuerungsfunktionen ... Messen von Signalen Grundsaltungen und Wirkungsweise von Antrieben Darstellung von Antriebseinheiten und Funktionsplänen
Rahmenlehrplan: Design und Erstellen mechatronischer Systeme	Betriebskennwerte und Kennlinien von Antrieben Grenzwerte Funktionsweise, Auswahl und Einstellung von Schutzeinrichtungen Steuern und Regeln von Antrieben Positionierungsvorgänge, Freiheitsgrade Prüf- und Meßverfahren zur Positionsbestimmung ... Einarbeiten von Änderungen an vorhandene Unterlagen Programmieren von Bewegungsabläufen und Steuerungsfunktionen Computersimulation Meßwerterfassung an Schnittstellen
Ausbildungsrahmenplan: Installieren und Testen von Hard- und Softwarekomponenten	Hard- und Softwareschnittstellen, Kompatibilität von Hardwarekomponenten sowie Systemvoraussetzungen für Software prüfen Systemkomponenten zusammenstellen und verbinden Hardware konfigurieren, Software installieren und anpassen Netzwerke und Bussysteme installieren und konfigurieren Signale an Schnittstellen prüfen, Protokolle interpretieren, Systeme testen Versionswechsel von Software durchführen Änderungen in der Hard- und Software dokumentieren

Modul	Inhalte des Moduls bzw. der Qualifikation
Ausbildungsrahmenplan: Aufbauen und Prüfen von Steuerungen	<p>Einrichtungen zur Versorgung mit elektrischer, pneumatischer oder hydraulischer Energie anschließen, prüfen und einstellen</p> <p>...</p> <p>Aufgabenstellung, insbesondere Bewegungsabläufe und Wechselwirkung an Schnittstellen des zu steuernden Systems analysieren</p> <p>Steuerungskonzepte zuordnen und Steuerungseinrichtungen auswählen</p> <p>Sensoren, Aktoren und Wandler installieren</p> <p>das Zusammenwirken von verknüpften Funktionen prüfen und einstellen, Fehler unter Beachtung der Schnittstellen eingrenzen</p>
Ausbildungsrahmenplan: Programmieren mechatronischer Systeme	<p>Steuerungen in unterschiedlichen Realisierungsformen beurteilen</p> <p>Steuerungsprogramme eingeben und ändern, Testprogramme erstellen und anwenden</p> <p>Anwendungsprogramme für Steuerungen erstellen, eingeben und testen</p> <p>Programmablauf in mechatronischen Systemen überwachen, Fehler feststellen und beheben</p>
Ausbildungsrahmenplan: Prüfen und Einstellen von Funktionen an mechatronischen Systemen	<p>Mess- und Prüfverfahren sowie Diagnosesysteme auswählen, elektrische Größen und Signale an Schnittstellen prüfen</p> <p>Signalverarbeitungsbaugruppen anschließen und deren Ein- und Ausgangssignale prüfen</p> <p>Messeinrichtungen zum Erfassen von Bewegungsabläufen, Druck und Temperatur prüfen</p> <p>Einrichtungen zum Erfassen von Grenzwerten, insbesondere Schalter und Sensoren, prüfen und justieren</p> <p>Aktoren nach sicherheitstechnischen Gesichtspunkten beurteilen und einstellen</p> <p>Steuer-, Regel- und Überwachungseinrichtungen prüfen, Regelparameter einstellen</p> <p>Sollwerte von prozessrelevanten Größen, insbesondere von Bewegungsabläufen und Druck einstellen</p> <p>Fehler unter Beachtung der Schnittstellen mechanischer, fluidischer und elektrischer Baugruppen durch Sichtkontrolle, Prüfen und Messen sowie mit Hilfe von Prüfsystemen und Testprogrammen systematisch eingrenzen</p> <p>elektrisch und elektronisch gesteuerte Antriebe prüfen und einstellen</p> <p>Störungen und Fehler auf mögliche Ursachen untersuchen, die Möglichkeiten ihrer Beseitigung beurteilen und die Instandsetzung einleiten</p> <p>Einzel- und Gesamtfunktion prüfen und dokumentieren</p>
Ausbildungsrahmenplan: In Betrieb nehmen und Bedienen mechatronischer Systeme	<p>...</p> <p>Hilfs- und Steuerstromkreise einschließlich zugehöriger Signal- und Befehlsgeber für Mess-, Steuer- und Überwachungseinrichtungen prüfen und in Betrieb nehmen</p> <p>Hauptstromkreise prüfen und schrittweise in Betrieb nehmen, Betriebswerte messen, Sollwerte einstellen</p> <p>Fluidikeinrichtungen in Betrieb nehmen</p> <p>Beweglichkeit, Dichtheit, Laufruhe, Umdrehungsfrequenz,</p> <p>Programme und Daten laden und sichern, Programmablauf prüfen und anpassen</p> <p>Signalübertragungssysteme, insbesondere Feldbusse prüfen und in Betrieb nehmen</p> <p>mechatronische Systeme in Betrieb nehmen, Funktionsprüfung durchführen</p> <p>...</p> <p>Systemparameter bei der Inbetriebnahme ermitteln, mit vorgegebenen Werten vergleichen und einstellen</p> <p>Maschinen und Systeme bedienen, Probelauf bei Nenn- und Grenzwerten durchführen</p>

Elektroniker/Elektronikerin für Automatisierungstechnik

Modul	Inhalte des Moduls bzw. der Qualifikation
Rahmenlehrplan: Steuerungen für Anlagen programmieren und realisieren	Kompakte, modulare und rechnerbasierte Steuerungen, Baugruppen Bussystem auf Feldebene Digitale und analoge Signalverarbeitung Strukturierte Programmierung Entwurfsverfahren Schrittketten Programmiersprachen, auch grafische Variablendeklaration, Instanziierung, symbolische Adressierung Programmsimulation, Fehlersuche, Fehleranalyse Anlagensicherheit durch Hardware und Programmierung Onlinehilfe
Rahmenlehrplan: Antriebssysteme auswählen und integrieren	... Analoge, digitale und programmierbare Sensoren Stromrichter Servoventile Drehzahlregelung, Lageregelung Standardregler Reglereinstellungen ... Kinematik Handhabungsautomaten
Rahmenlehrplan: Automatisierungssysteme in Betrieb nehmen und übergeben	... Diagnoseverfahren Überprüfung von Hard- und Softwarekomponenten Analoge, digitale und programmierbare Sensoren Umrichter- und Reglerparametrierung Prozessvisualisierung Anlagensicherheit, Betriebssicherheit ...
Rahmenlehrplan: Automatisierungssysteme planen	Handbücher, Applikationen, Regelwerke Projektmanagement Pflichtenheft Programmerstellung, Programmtest Wirtschaftlichkeit Anlagen- und Produktgestaltung Normen, Bestimmungen und Vorschriften Recycling Qualitätsmanagement

Ausbildungsrahmenplan: Errichten von Einrichtungen der Automatisierungstechnik	Systeme ändern, anpassen, verdrahten, verbinden, konfigurieren, montieren und demontieren Maschinen, Geräte und sonstige Betriebsmittel aufstellen, ausrichten, befestigen und anschließen Schaltgeräte und Betriebsmittel zum Steuern, Regeln, Messen und Überwachen einbauen, verdrahten und kennzeichnen Sensoren und Aktoren montieren Steuerungen installieren Einrichtungen der Energieversorgung und -verteilung bereitstellen Signal- und Datenübertragungssysteme installieren, prüfen und in Betrieb nehmen Signal- und Datenübertragungseinrichtungen verlegen und montieren elektrische Antriebe montieren, ausrichten, kuppeln und anschließen Baugruppen der Regelungstechnik montieren und justieren
Ausbildungsrahmenplan: Konfigurieren und Programmieren von Automatisierungssystemen	Steuerungsprogramme erstellen Automatisierungsgeräte programmieren analoge und programmierbare Sensorsysteme konfigurieren und parametrieren komplexe Steuerungen anpassen Anwendersoftware zur Maschinen- oder Prozesssteuerung konfigurieren und parametrieren Signal- und Datenübertragungseinrichtungen konfigurieren Netzwerkbetriebssysteme und Netzwerke konfigurieren und parametrieren Komponenten der Informationstechnik und Automatisierungstechnik konfigurieren und parametrieren Anwendungsprogramme für Leitsysteme und Datennetze konfigurieren und parametrieren
Ausbildungsrahmenplan: Prüfen und In Betrieb nehmen von Automatisierungssystemen	... Komponenten der Automatisierungstechnik justieren und prüfen analoge und programmierbare Sensorsysteme in Betrieb nehmen Test- und Diagnosesoftware einsetzen, Signale an Schnittstellen prüfen, netzwerkspezifische Prüfungen durchführen Automatisierungssysteme unter Beachtung der betriebs- und anlagenspezifischen Schutzmaßnahmen in Betrieb nehmen und prüfen Inbetriebnahmeprotokolle erstellen und Anlagen übergeben

Industriemechaniker

Modul	Inhalte des Moduls bzw. der Qualifikation
Rahmenlehrplan: Sicherstellen der Betriebsfähigkeit automatisierter Systeme	... Steuerung Regelung Programmierbare Steuerungen Betriebsarten Ablaufsprache, Funktionsbausteinsprache flexible Handhabungssysteme Schnittstellen Instandhaltungsvorschriften Sicherheitseinrichtungen
Ausbildungsrahmenplan: Herstellen, Montieren und Demontieren von Bauteilen, Baugruppen und Systemen	technische Unterlagen analysieren Montage- und Demontagepläne erstellen und anwenden ... Baugruppen und Bauteile lage- und funktionsgerecht montieren Baugruppen, Systeme oder Anlagen demontieren und kennzeichnen Baugruppen und Bauteile reinigen, pflegen und lagern Maschinen oder Fertigungssysteme umrüsten

Allgemeine Voraussetzungen zum Betreiben der Geräte

Allgemeine Anforderungen bezüglich des sicheren Betriebs der Geräte:

- In gewerblichen Einrichtungen sind die Unfallverhütungsvorschriften der DGUV Vorschrift 3 "Elektrische Anlagen und Betriebsmittel" zu beachten.
- Der Labor- oder Unterrichtsraum muss durch einen Arbeitsverantwortlichen überwacht werden.
 - Ein Arbeitsverantwortlicher ist eine Elektrofachkraft oder eine elektrotechnisch unterwiesene Person mit Kenntnis von Sicherheitsanforderungen und Sicherheitsvorschriften mit aktenkundiger Unterweisung.

Der Labor- oder Unterrichtsraum muss mit den folgenden Einrichtungen ausgestattet sein:

- Es muss eine NOT-AUS-Einrichtung vorhanden sein.
 - Innerhalb und mindestens ein NOT-AUS außerhalb des Labor- oder Unterrichtsraums.
- Der Labor- oder Unterrichtsraum ist gegen unbefugtes Einschalten der Betriebsspannung bzw. der Druckluftversorgung zu sichern.
 - z. B. durch einen Schlüsselschalter
 - z. B. durch abschließbare Einschaltventile
- Der Labor- oder Unterrichtsraum muss durch Fehlerstromschutzeinrichtungen (RCD) geschützt werden.
 - Betreiben Sie elektrische Geräte (z. B. Netzgeräte, Verdichter, Hydraulikaggregate) nur in Ausbildungsräumen, die mit einer Fehlerstromschutzeinrichtung ausgestattet sind.
 - Als Fehlerstromschutzeinrichtung ist ein RCD-Schutzschalter mit Differenzstrom ≤ 30 mA, Typ B zu verwenden.
- Der Labor- oder Unterrichtsraum muss durch Überstromschutzeinrichtungen geschützt sein.
 - Sicherungen oder Leitungsschutzschalter
- Es dürfen keine Geräte mit Schäden oder Mängeln verwendet werden.
 - Schadhafte Geräte sind zu sperren und aus dem Labor- oder Unterrichtsraum zu entnehmen.
 - Beschädigte Verbindungsleitungen, Druckluftschläuche und Hydraulikschläuche stellen ein Sicherheitsrisiko dar und müssen aus dem Labor- oder Unterrichtsraum entfernt werden.

Bestimmungsgemäße Verwendung

Die Stationen der CP Factory sind nur zu benutzen:

- für die bestimmungsgemäße Verwendung im Lehr- und Ausbildungsbetrieb
- in sicherheitstechnisch einwandfreiem Zustand

Die Stationen sind nach dem heutigen Stand der Technik und den anerkannten sicherheitstechnischen Regeln gebaut. Dennoch können bei unsachgemäßer Verwendung Gefahren für Leib und Leben des Benutzers oder Dritter und Beeinträchtigungen der Komponenten entstehen.

Das Lernsystem von Festo Didactic ist ausschließlich für die Aus- und Weiterbildung im Bereich Automatisierung und Technik entwickelt und hergestellt. Das Ausbildungsunternehmen und/oder die Auszubildenden hat/haben dafür Sorge zu tragen, dass die Auszubildenden die Sicherheitsvorkehrungen, die in diesem Arbeitsbuch beschrieben sind, beachten.

Festo Didactic schließt hiermit jegliche Haftung für Schäden des Auszubildenden, des

Ausbildungsunternehmens und/oder sonstiger Dritter aus, die bei Gebrauch/Einsatz dieses Gerätes außerhalb einer reinen Ausbildungssituation auftreten; es sei denn Festo Didactic hat solche Schäden vorsätzlich oder grob fahrlässig verursacht.

Arbeits- und Sicherheitshinweise



■ Wichtige Hinweise

Grundvoraussetzung für den sicherheitsgerechten Umgang und den störungsfreien Betrieb der CP Factory Stationen ist die Kenntnis der grundlegenden Sicherheitshinweise und der Sicherheitsvorschriften. Dieses Handbuch enthält die wichtigsten Hinweise, um die CP Factory Stationen sicherheitsgerecht zu betreiben.

Insbesondere die Sicherheitshinweise sind von allen Personen zu beachten, die den CP Factory Stationen arbeiten.

Darüber hinaus sind die für den Einsatzort geltenden Regeln und Vorschriften zur Unfallverhütung zu beachten.

■ Verpflichtung des Betreibers

Der Betreiber verpflichtet sich, nur Personen an CP Factory Stationen arbeiten zu lassen, die:

- mit den grundlegenden Vorschriften über Arbeitssicherheit und Unfallverhütung vertraut und in die Handhabung der CP Factory Stationen eingewiesen sind,
- das Sicherheitskapitel und die Warnhinweise in diesem Handbuch gelesen und verstanden haben.

Das sicherheitsbewusste Arbeiten des Personals soll in regelmäßigen Abständen überprüft werden.

■ Verpflichtung der Auszubildenden

Alle Personen, die mit Arbeiten an CP Factory Stationen beauftragt sind, verpflichten sich, vor Arbeitsbeginn:

- das Sicherheitskapitel und die Warnhinweise in diesem Handbuch zu lesen,
- die grundlegenden Vorschriften über Arbeitssicherheit und Unfallverhütung zu beachten.

■ Gefahren im Umgang mit der CP Factory®

Das CP Factory® ist nach dem Stand der Technik und den anerkannten sicherheitstechnischen Regeln gebaut. Dennoch können bei ihrer Verwendung Gefahren für Leib und Leben des Benutzers oder Dritter bzw. Beeinträchtigungen an der Maschine oder an anderen Sachwerten entstehen.

Das CP Factory® ist nur zu benutzen:

- für die bestimmungsgemäße Verwendung und
- in sicherheitstechnisch einwandfreiem Zustand.



Störungen, die die Sicherheit beeinträchtigen können, sind umgehend zu beseitigen!

- Greifen Sie nur bei Stillstand in den Aufbau.
- Achten Sie auf festen Sitz aller Komponenten.
- Beachten Sie Angaben zur Platzierung der Komponenten.

■ Sicher arbeiten

Allgemein

- Die Auszubildenden dürfen nur unter Aufsicht einer Ausbilderin/eines Ausbilders an den Schaltungen arbeiten.
- Betreiben Sie elektrische Geräte (z. B. Netzgeräte, Verdichter, Hydraulikaggregate) nur in Ausbildungsräumen, die mit einer Fehlerstromschutzeinrichtung (RCD) ausgestattet sind.
- Beachten Sie die Angaben der Datenblätter zu den einzelnen Komponenten, insbesondere auch alle Hinweise zur Sicherheit!
- Störungen, die die Sicherheit beeinträchtigen können, dürfen beim Schulungsbetrieb nicht erzeugt werden und sind umgehend zu beseitigen.
- Tragen Sie Ihre persönliche Schutzausrüstung (Schutzbrille, Sicherheitsschuhe), wenn Sie an den Schaltungen arbeiten.

Mechanik

- Energieversorgung ausschalten!
 - Schalten Sie sowohl die Arbeitsenergie als auch die Steuerenergie aus, bevor Sie an der Schaltung arbeiten.
 - Greifen Sie nur bei Stillstand in den Aufbau.
 - Beachten Sie mögliche Nachlaufzeiten von Antrieben.
- Montieren Sie alle Komponenten fest auf die Profilplatte.
- Stellen Sie sicher, dass Grenztaster nicht frontal betätigt werden.
- Verletzungsgefahr bei der Fehlersuche!
Benutzen Sie zur Betätigung der Grenztaster ein Werkzeug, z. B. einen Schraubendreher.
- Stellen Sie alle Komponenten so auf, dass das Betätigen von Schaltern und Trenneinrichtungen nicht erschwert wird.
- Beachten Sie Angaben zur Platzierung der Komponenten.

Elektrik

- Spannungsfrei schalten!
 - Schalten Sie die Spannungsversorgung aus, bevor Sie an der Schaltung arbeiten.
 - Beachten Sie, dass elektrische Energie in einzelnen Komponenten gespeichert sein kann. Informationen hierzu finden Sie in den Datenblättern und Bedienungsanleitungen der Komponenten.
- Verwenden Sie nur Schutzkleinspannungen, maximal 24 V DC.
- Herstellen bzw. Abbauen von elektrischen Anschlüssen
 - Stellen Sie elektrische Anschlüsse nur in spannungslosem Zustand her.
 - Bauen Sie elektrische Anschlüsse nur in spannungslosem Zustand ab.
- Die zulässigen Strombelastungen von Leitungen und Geräten dürfen nicht überschritten werden.
 - Vergleichen Sie stets die Strom-Werte von Gerät, Leitung und Sicherung.
 - Benutzen Sie bei Nichtübereinstimmung eine separate vorgeschaltete Sicherung als entsprechenden Überstromschutz.
- Verwenden Sie für die elektrischen Anschlüsse nur Verbindungsleitungen mit Sicherheitssteckern.
- Verlegen Sie Verbindungsleitungen so, dass sie nicht geknickt oder geschert werden.
- Verlegen Sie Leitungen nicht über heiße Oberflächen.
 - Heiße Oberflächen sind mit einem Warnsymbol entsprechend gekennzeichnet.
- Achten Sie darauf, dass Verbindungsleitungen nicht dauerhaft unter Zug stehen.
- Geräte mit Erdungsanschluss sind stets zu erden.
 - Sofern ein Erdanschluss (grün-gelbe Laborbuchse) vorhanden ist, so muss der Anschluss an Schutz Erde stets erfolgen. Die Schutz Erde muss stets als erstes (vor der Spannung) kontaktiert werden und darf nur als letztes (nach Trennung der Spannung) getrennt werden.
 - Einige Geräte haben einen hohen Ableitstrom. Diese Geräte müssen zusätzlich mit einem Schutzleiter geerdet werden.
- Wenn in den Technischen Daten nicht anders angegeben, besitzt das Gerät keine integrierte Sicherung.
- Ziehen Sie beim Abbauen der Verbindungsleitungen nur an den Sicherheitssteckern, nicht an den Leitungen.

Industrieroboter

- Beachten Sie die Angaben des Sicherheitstechnischen Handbuchs für MELFA Industrieroboter von Mitsubishi.
- Transportieren des Roboters
 - Soll der Roboter nach einer erfolgten Installation erneut transportiert werden, so ist darauf zu achten, dass die Bremsen des Roboterarms gelöst werden und der Arm wieder in die Transportposition gebracht wird. Der Roboter darf niemals mit ausgefahrenem Arm getragen oder transportiert werden.
 - Vor einem erneuten Transport müssen die Transportsicherungen montiert werden.
- Bringen Sie um den Roboter herum eine Sicherheitsbarriere oder Einfriedung an, damit bei unbedachter Annäherung an den Roboter potenzielle Gefahren ausgeschaltet werden.
- Bringen Sie eine Sicherheitsverriegelung an, die das Öffnen der Tür verhindert, wenn ein Bediener versucht, die Barriere (innerhalb des Sicherheitsbereichs) zu passieren, während der Roboter in Betrieb ist, oder durch die beim Öffnen der Tür die Servo-Stromversorgung des Roboters ausgeschaltet und Peripheriegeräte automatisch angehalten werden (die Servo-Stromversorgung ausgeschaltet wird). Schließen Sie die Leitung dieses Sicherheitsverriegelungssignals direkt an den Roboter und die Peripheriegeräte an und nicht an ein Sekundärgerät (z. B. SPS). Schließen Sie die Leitung bei den Robotern von Mitsubishi an die Eingangsklemme des NOT-HALT-Schalters oder des Türschalters an.
- Installieren Sie das Robotersystem so, dass sich das Steuergerät außerhalb des Sicherheitsbereichs befindet.
- Bringen Sie an einer leicht zugänglichen Stelle in Roboternähe einen manuellen NOT-HALT Schalter an und verbinden Sie diesen mit der Notausschaltungs-Eingangsklemme des Steuergerätes.
- Konfigurieren Sie das System so, dass der Betriebsstatus des Roboters (d. h. Ausführungsprogrammbetrieb, Stoppstatus, Servo-Ein-Status, Fehlerstatus usw.) anhand von Sichtanzeigen o.Ä. problemlos extern überprüft werden kann.
- Alle Fehler sollten zur Sicherheit des Bedienungspersonals in unmittelbarer Nähe des Roboters durch Hör- und Sichtsignale angezeigt werden.
- Achten Sie zur Vermeidung von Stromschlag und elektrostatischer Entladung, zur Verbesserung des Störwiderstands und zur Verhinderung unnötiger elektromagnetischer Abstrahlung auf eine ordnungsgemäße Erdung von Roboter und Steuergerät.
- Falls es erforderlich wird, dass ein Bediener zu Schulungs- und Justierzwecken o. Ä. den Bereich hinter der Barriere bei eingeschalteter Servoversorgung betritt, so muss er einen Schutzhelm tragen. Bei der Teach-In-Programmierung muss der Servo-Ein-Betrieb so über den Tormanschalter durchgeführt werden, dass die Servostromversorgung jederzeit ausgeschaltet werden kann und dass genügend Platz für den Körper des Bedieners besteht, um potenzielle Gefahren zu vermeiden.
- Bestimmen Sie bei der Teamarbeit einen Verantwortlichen und stellen Sie sicher, dass Handzeichen und Befehle eingehend bekannt und der Sicherheit dienlich sind.
- Der Anschluss der Kabel darf nur bei ausgeschalteter Spannungsversorgung erfolgen. Es besteht Stromschlaggefahr.
- Berühren Sie während des Betriebes kein bewegliches Teil des Roboters. Schalten Sie den Roboter vor jeder Arbeit in Reichweite des Roboters aus.
- Bewahren Sie eine nicht angeschlossene Teaching-Box wegen der Nichtwirksamkeit der integrierten NOT-HALT Einrichtung nicht in der Nähe des Roboters auf.

Pneumatik

- Drucklos schalten!
 - Schalten Sie die Druckluftversorgung aus, bevor Sie an der Schaltung arbeiten.
 - Prüfen Sie mit Druckmessgeräten, ob die komplette Schaltung drucklos ist.
 - Beachten Sie, dass in Druckspeichern Energie gespeichert sein kann.
Informationen hierzu finden Sie in den Datenblättern und Bedienungsanleitungen der Komponenten.
- Überschreiten Sie nicht den zulässigen Druck von 600 kPa (6 bar).
- Schalten Sie die Druckluft erst ein, wenn Sie alle Schlauchverbindungen hergestellt und gesichert haben.
- Entkuppeln Sie keine Schläuche unter Druck.
- Versuchen Sie nicht, Schläuche oder Steckverbindungen mit den Fingern oder der Hand zu verschließen.
- Verletzungsgefahr beim Einschalten von Druckluft!
Zylinder können selbsttätig aus- und einfahren.
- Unfallgefahr durch ausfahrende Zylinder!
 - Platzieren Sie pneumatische Zylinder immer so, dass der Arbeitsraum der Kolbenstange über den gesamten Hubbereich frei ist.
 - Stellen Sie sicher, dass die Kolbenstange nicht gegen starre Komponenten des Aufbaus fahren kann.
- Unfallgefahr durch abspringende Schläuche!
 - Verwenden Sie kürzest mögliche Schlauchverbindungen.
 - Beim Abspringen von Schläuchen:
Schalten Sie die Druckluftzufuhr sofort aus.
- Pneumatischer Schaltungsaufbau
Verbinden Sie die Geräte mit dem Kunststoffschlauch mit 4 mm oder 6 mm Außendurchmesser. Stecken Sie dabei den Schlauch bis zum Anschlag in die Steckverbindung.
- Schalten Sie vor dem Schaltungsabbau die Druckluftversorgung aus.
- Pneumatischer Schaltungsabbau
Drücken Sie den blauen Lösungsring nieder, der Schlauch kann abgezogen werden.
- Lärm durch ausströmende Druckluft
 - Lärm durch ausströmende Druckluft kann schädlich für das Gehör sein. Reduzieren Sie den Lärm durch den Einsatz von Schalldämpfern oder tragen Sie einen Gehörschutz, falls der Lärm sich nicht vermeiden lässt.
 - Alle Abluftanschlüsse der Komponenten der Gerätesätze sind mit Schalldämpfern versehen. Entfernen Sie diese Schalldämpfer nicht.

Informationen für Ausbilder/Lehrer und Auszubildende/Studenten zur Nutzung der Unterlage

Die Unterlage gliedert sich in einen Theorie- und einen Aufgabenteil. Der Theorieteil umfasst sieben Kapitel und der Aufgabenteil enthält acht Projekte, die sich didaktisch auf die genannten Ausbildungsberufe sowie deren Module aus dem Rahmenlehrplan bzw. deren Qualifikationen aus dem Rahmenlehrplan beziehen.

Wahlweise kann die Unterlage auch für die Technikerausbildung in den genannten Fachrichtungen eingesetzt werden. Da jedoch die Lehrpläne je nach Bundesland stärker differieren, wird an dieser Stelle auf einen entsprechenden Bezug zu den Ausbildungsmodulen verzichtet.

Damit zielt die Unterlage auf die berufliche Erstausbildung bzw. Weiterbildung mit Level drei bzw. vier des Europäischen Qualifikationsrahmens.

Der Theorieteil wurde hardwareneutral formuliert. Er umfasst im Wesentlichen Text und Grafiken. Am Ende jedes Kapitels findet sich ein Verständnistest. Diese können im Rahmen des Unterrichts zur Wissenskontrolle oder auch zur Leistungsüberprüfung eingesetzt werden.

Lösungen in Textpassagen sind mit roter Farbe dargestellt. Lösungen sowie Ergänzungen in Grafiken, Tabellen oder Diagrammen sind rot oder grau hinterlegt.

Der Theorieteil wurde für ca. sechs Unterrichtseinheiten á neunzig Minuten konzipiert. Bei Bedarf und entsprechend der zu vermittelnden Aneignungstiefe können diese aber schnell und unproblematisch erweitert oder verkürzt werden.

Der Theorieteil dient als Vorbereitung zur Durchführung der Projektaufgaben, wobei jede Projektaufgabe jeweils weitere neunzig Minuten praktischen Unterricht ermöglicht. Diese Aufgabenblöcke wurden fachlich auf die CP Factory Robotertermontagezelle von Festo Didactic ausgerichtet. Da diese in der Regel kundenspezifisch bestellt und geliefert wurden, sollten vor Beginn des praktischen Unterrichts die Aufgaben auf die konkrete Anlage angepasst werden.

Die Aufgabenblöcke beinhalten eine Anzahl von Teilaufgaben, die in ihrem Schwierigkeitsgrad gestaffelt sind. Sie beginnen in der Regel mit einfacheren Problemstellungen und steigern sich im Anschluss. Je nach Ausbildungsberuf und Stand des Wissens bzw. Könnens der Auszubildenden haben Sie als Ausbilder bzw. Lehrer so die Möglichkeit, die aus Ihrer Sicht geeigneten Teilaufgaben auszuwählen.

Prinzipiell gehen wir pro Aufgabenblock von neunzig Minuten Lernumfang im praktischen Unterricht aus. Dieser Umfang kann aber leicht durch Sie erweitert werden, wenn Sie alle Teilaufgaben durch die Auszubildenden bearbeiten lassen.

Zur besseren Einordnung in Ihren Unterricht sind vor jedem Aufgabenblock folgende Hinweise eingefügt:

- Lernziele des Aufgabenblocks
- Problemstellung
- Arbeitsaufträge
- Arbeitshilfen
- Hinweis für den Lehrer / Ausbilder

Die Gesamtlernzeit für das Thema Robotik wird dementsprechend mit vierzehn Einheiten zu je neunzig Minuten angegeben.

Haben Sie Tipps, Anregungen oder Vorschläge zur Verbesserung dieses Arbeitsbuchs?

Dann senden Sie diese bitte per E-Mail an did@festo.com.

Die Autoren und Festo Didactic freuen sich auf Ihre Rückmeldung.

Benötigte Komponenten

Im Folgenden sind die Minimal-Anforderungen an Komponenten und die Software aufgelistet, die Sie für die Bearbeitung aller Arbeitsaufträge mit der CP Factory Roboterstation benötigen.

Komponente	Beschreibung	Bestell-Nr.	Menge
CP Factory Montagezelle mit Roboter	Die Montagezelle umfasst einen Roboter vom Typ Mitsubishi RV-4FL, ein Greiferwechselsystem mit 3 Greifern, ein Transportband mit einer Box, die Platinen bereitstellt, ein LED-Feld mit einer Festo Kamera SBOC, ein Montageplatz, ein Magazin für Sicherungen zur Bestückung von Platinen. Die Montagezelle besitzt ein Sicherheitsgehäuse mit Sicherheitstür vorne und hinten.		1
CP Factory Grundmodul Band mit Bypass	Das CP Factory Band transportiert Warenträger und über den Bypass zur Bearbeitungsposition und anschließend weiter zur nächsten Station.		
Bedienfeld mit TouchPanel	Das Bedienfeld umfasst Start-, Stopp- und Richten-Taste, Betriebsarten-Wahlschalter (Man/Auto), 2 Leuchtmeldern und einem Not-Halt-Schlagaster, sowie ein integriertes Touch-Panel.		1
Warenträgersatz	Der Warenträgersatz besteht aus Warenträger und Palette.		1
Werkstücksatz	Der Werkstücksatz besteht aus folgenden Werkstücken: 1xFrontschale, 1xRückschale, 1xPlatine, 2xSicherungen		1
CIROS® Studio	Virtuelle Lernumgebungen erstellen	8038980-SSL	1
Optional CIROS® Education	Virtuelle Lernumgebungen anwenden	8038980-EL12	1

Lernziele

■ Aufgabe 1: Aufbau der CP Factory Station Roboterontage

Wenn Sie diese Aufgabe bearbeitet haben,

- kennen Sie die technische Dokumentation des Mitsubishi Robotersystems RV-4FL.
- können Sie die wesentlichen Komponenten der Roboterontagestation beschreiben.
- kennen Sie die wesentlichen Bedienungsfunktionen der Robotersteuerung und des Programmierhandgerätes (PGH).
- können Sie den Roboter für die Inbetriebnahme vorbereiten.
- können Sie die Robotersteuerung und das PGH verkabeln und mit Spannung versorgen.
- können Sie die NOT-HALT Einrichtungen der Station identifizieren.
- können Sie mit dem PGH die Roboterachsen bewegen.

■ Aufgabe 2: Funktionsprüfung des Mitsubishi Roboters RV-4FL in der Montagestation

Wenn Sie diese Aufgabe bearbeitet haben,

- können Sie einen Prüfplan für ein komplexes technisches System erstellen.
- können Sie eine Funktionsprüfung eines komplexen technischen Systems durchführen.
- können Sie die Struktur des Roboterarms eines vertikalen Knickarmroboters beschreiben.
- können Sie interne Parameter der Robotersteuerung anzeigen und ändern.
- können Sie den Roboter in seinen verschiedenen Koordinatensystemen mit dem PGH steuern.
- können Sie die NOT-HALT Einrichtungen der Station bedienen.
- kennen Sie die Grundlagen zur Bedienung des 3D-Simulationssystems CIROS.
- können Sie das virtuelle Modell der CP Factory Station Roboterontage bedienen.

■ Aufgabe 3: Werkzeugwechselsystem für einfache Pick&Place Aufgabe nutzen

Wenn Sie diese Aufgabe bearbeitet haben,

- kennen Sie die Grundfunktionen der Mitsubishi Roboterprogrammiersprache Melfa Basic
- können Sie den Ablaufplan für eine einfache Roboteranwendung erstellen.
- können Sie Positionen in CIROS für eine Roboteranwendung einlernen (teachen) und in eine Positionsliste eintragen.
- kennen Sie den unterschiedlichen Einsatz von PTP und linearen Bewegungen eines Robotersystems.
- kennen Sie die Funktionalität eines Greiferwechselsystems und können Sie in einer einfachen Roboteranwendung in CIROS anwenden, testen und optimieren.
- können Sie die Kommunikationsschnittstelle zwischen CIROS und der realen Robotersteuerung konfigurieren und den Download von Programmen und Positionslisten in die reale Steuerung durchführen.
- kennen Sie die Sicherheitsregeln bei der Kommunikation zwischen PC und der realen Steuerung.
- können Sie das Programm im Debug-Modus analysieren und optimieren.

■ Aufgabe 4: Platine auf Werkstück montieren und Prüfung des Werkzeugwechselsystems

Wenn Sie diese Aufgabe bearbeitet haben,

- können Sie Projekte mit mehreren Programmen erstellen.
- kennen Sie den Einsatz globaler Variablen.
- können sie die Positionen in CIROS für eine Roboteranwendung einlernen (teachen) und in eine Positionsliste eintragen.
- können Sie Testszenarien für Roboterfunktionen erstellen.
- können Sie die Kommunikation mit einer externen SPS anwenden.
- können Sie die Kommunikationsschnittstelle zwischen CIROS und der realen Robotersteuerung konfigurieren und den Download von Programmen und Positionslisten in die reale Steuerung durchführen.
- können Sie Programm im Debug-Modus im Abgleich mit der Hardware analysieren und optimieren.

■ Aufgabe 5: Optische Positionsprüfung eines Werkstücks

Wenn Sie diese Aufgabe bearbeitet haben,

- können Sie den Ablaufplan für eine komplexe Anwendung erstellen.
- können Sie den Einsatz der Kamera für eine Anwendung konzipieren.
- können Sie die Kameraauswertung ins Programm übernehmen.
- können sie die Sensorik bei der Montage identifizieren und in den Ablauf integrieren.
- können sie die Kollisionsprüfung in der Simulation durchführen.
- Können Sie die modulare Programmerstellung durchführen.
- können Sie die Kommunikationsschnittstelle zwischen CIROS und der realen Robotersteuerung konfigurieren und den Download von Programmen und Positionslisten in die reale Steuerung durchführen.
- können Sie umfangreiche Positionsliste in der Hardware testen und optimieren.
- können Sie ein einfaches Programm im Debug-Modus im Abgleich mit der Hardware analysieren und optimieren.

■ Aufgabe 6: Anwendung der externen Funktionsbibliothek

Wenn Sie diese Aufgabe bearbeitet haben,

- können Sie eine komplexe Funktionsbibliothek analysieren und die Anwendungsmöglichkeiten dokumentieren.
- können Sie die Funktionsbibliothek zur optischen Kontrolle der Montage eines Werkstücks anwenden.
- können Sie verschiedene Fehlersituationen in der Robotik identifizieren und im Programmablauf berücksichtigen.
- können Sie das Konzept einer externen universellen Liste von Positionsvariablen anwenden.
- beherrschen Sie die Methode der modularen Programmerstellung.
- können Sie die Kommunikationsschnittstelle zwischen CIROS und der realen Robotersteuerung konfigurieren und den Download von Programmen und Positionslisten in die reale Steuerung durchführen.
- können Sie das Programm im Debug-Modus im Abgleich mit der Hardware analysieren und optimieren.

■ **Aufgabe 7: Montage einer Platine mit Sicherungen unter Anwendung der externen Funktionsbibliothek**

Wenn Sie diese Aufgabe bearbeitet haben,

- können Sie den Ablaufplan unter Anwendung der Funktionsbibliothek erstellen.
- können Sie die Funktionsbibliothek für eine geplante Anwendung effizient einsetzen und ihren Nutzen beurteilen.
- beherrschen Sie die Methode der modularen Programmerstellung.
- Können Sie die Kommunikationsschnittstelle zwischen CIROS und der realen Robotersteuerung konfigurieren und den Download von Programmen und Positionslisten in die reale Steuerung durchführen.
- können sie das Programm im Debug-Modus im Abgleich mit der Hardware analysieren und optimieren.

Grundlagen

1	Industrieroboter	3
1.1	Portalroboter	3
1.2	SCARA Roboter	4
1.3	Vertikaler Knickarmroboter	6
1.4	Systemkomponenten eines Industrieroboters	8
1.5	Kenngößen eines Industrieroboters	10
2	Greifer	14
2.1	Mechanische Greifer	14
2.2	Sauggreifer	16
2.3	Greiferwechselsysteme	18
3	Robotersteuerung	19
3.1	Kartesische Koordinatensysteme	19
3.2	Roboter-Koordinatensysteme	20
3.3	Steuerungsarten der Roboterbewegung	23
4	Sensoren	25
4.1	Induktive Näherungsschalter	25
4.2	Optische Näherungsschalter	26
4.3	Kamera	28
5	Programmierung von Industrieroboter	30
5.1	Online-Verfahren	30
5.2	Offline-Verfahren	31
5.3	CIROS Studio	32
5.4	Hinweise zur Roboterprogrammierung	37
5.4.1	Hindernis vermeiden	38
5.4.2	Beispiel: Werkstück greifen	38
5.4.3	Beispiel Paletten-Funktion	40
6	Sicherheitsmaßnahmen	42
6.1	Allgemeine Sicherheitshinweise	42
6.1.1	Betriebsanleitung	42
6.1.2	Verpflichtung des Betreibers	42
6.1.3	Verpflichtung des Bedienpersonals	42
6.2	Schutzeinrichtungen	42
6.2.1	Verriegelte trennende Schutzeinrichtung	43
7	Inbetriebnahme einer Roboteranwendung	43
7.1	Einleitung	43
7.2	Virtuelle Inbetriebnahme	43
7.3	Reale Inbetriebnahme der Roboteranwendung	43

1 Industrieroboter

Es gibt zwei Klassen von Roboter

- Stationäre Roboter und
- Mobile Roboter

Mobile Roboter haben keinen festen Standort, sondern müssen sich selbstständig in einer im allgemeinen unbekannten Umgebung zu vorgegebenen Zielpunkten bewegen. Dies erfordert ein leistungsfähiges Navigationssystem, das folgende Aufgaben lösen muss:

- Die Ermittlung der Position des Roboters
- Die Planung einer optimalen Bahn zu dem gewünschten Zielpunkt
- Die Berechnung von Ausweichmanövern und gegebenenfalls die Planung eines Ersatzweges zum Zielpunkt.

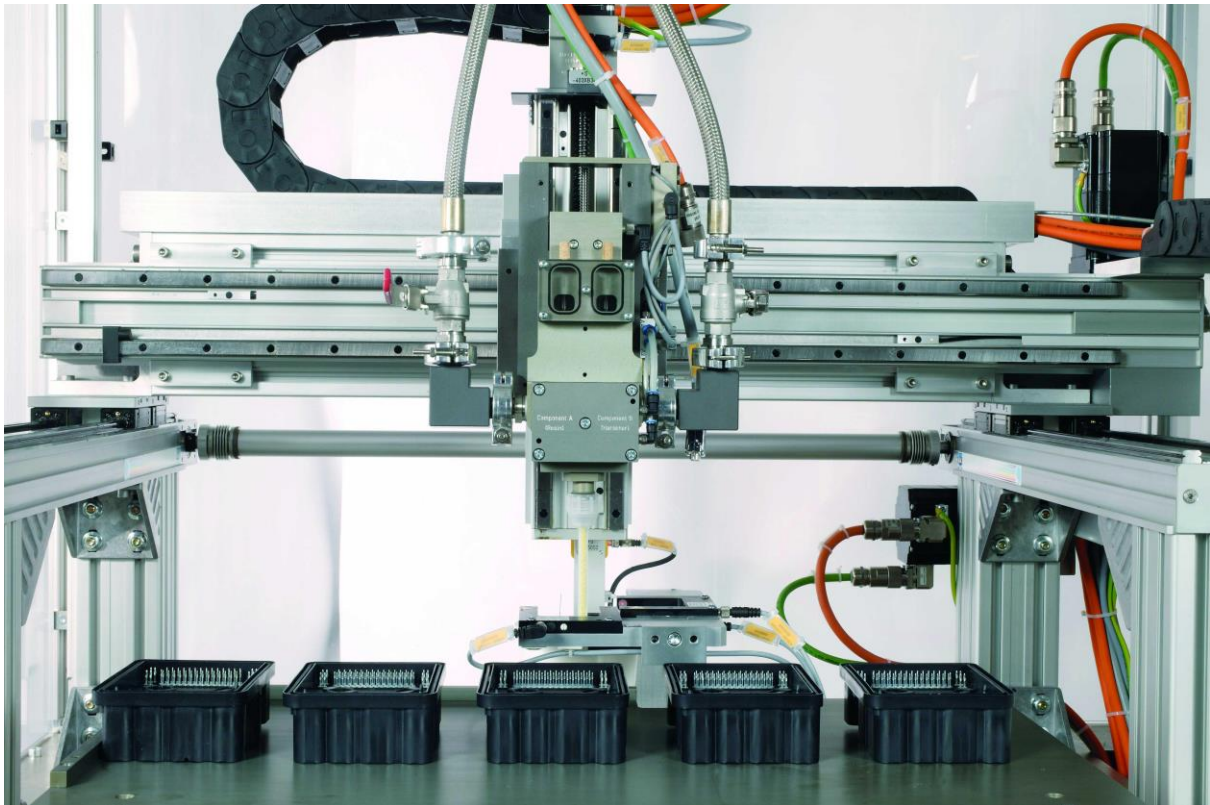
Stationäre Roboter sind an einem festen Ort positioniert. Sie umfassen insbesondere die Klasse der Industrieroboter. Nach VDI Richtlinie 2860 sind Industrieroboter universell einsetzbare Bewegungsautomaten mit mindestens 3 Achsen, deren Bewegungen hinsichtlich Bewegungsfolge und Wegen bzw. Winkeln frei (d. h. ohne mechanischen bzw. menschlichen Eingriff) programmierbar und gegebenenfalls sensorgeführt sind. Sie sind mit Greifern, Werkzeugen oder anderen Fertigungsmitteln ausrüstbar und können Handhabungs- und/oder Fertigungsaufgaben ausführen. Diese Definition ist nahezu identisch mit der Definition nach der europäischen Norm EN775.

Die Struktur eines Industrieroboters ist bestimmt durch seine Kinematik aus rotatorischen und/oder translatorischen Achsen mit den entsprechenden Gelenken. Durch die Anordnung dieser Achsen entstehen verschiedene Konfigurationen, wobei folgende besonders häufig eingesetzt werden:

- Portalroboter
- SCARA Roboter
- Vertikale Knickarmroboter

1.1 Portalroboter

Portalroboter sind eine effiziente Variante zur Automatisierung von Produktionsprozessen mittels Robotertechnik. Durch die Beladung von oben (z. B. über Ladeluken) bleibt die Zugänglichkeit zur Maschine erhalten. Dies ist besonders wichtig für Rüstvorgänge und Überwachungstätigkeiten. Portalroboter bestehen in der Regel aus 2-3 Linearachsen. Der Greifer wird über einen linearen vertikalen Arm gesteuert. Zur Positionierung werden 1-2 horizontalen Linearachsen eingesetzt. Portalroboter dienen vorwiegend dem Transport von schwerem Material über große Entfernungen.



Portalroboter mit zwei horizontalen und einer vertikalen Linearachse

1.2 SCARA Roboter

SCARA steht für die Abkürzung des englischen Begriffes „Selective Compliance Assembly Robot Arm“. SCARA Roboter werden auch als Schwenkarmroboter bezeichnet. Der SCARA Roboter hat eine serielle Kinematik mit typischerweise vier Achsen, die nur eine Handhabung in planparallelen Flächen zulässt. Die erste Drehachse bildet der um den Roboterfuß drehbare Schwenkarm, dessen Gelenk die zweite Achse. Am Ende des Schwenkarms ist die vertikale Z-Achse zur Aufnahme eines Greifers oder von Werkzeugen befestigt, die als vierte Achse auch um ihre Längsachse gedreht werden kann. Durch ihre Bauart bieten SCARA-Roboter eine hohe Steifigkeit in vertikaler Richtung, können jedoch nur geringere Kräfte in XY-Richtung aufnehmen.



SCARA Roboter

SCARA Roboter sind universell verwendbare Systeme zur schnellen und sehr genauen Pick-und-Place-Handhabung in XYZ-Richtung innerhalb von planparallelen Flächen. Kippbewegungen sind nicht möglich. Einsatzschwerpunkt sind Anwendungen, bei denen eine Reduzierung auf senkrechte Füge- und Handhabungsbewegungen möglich ist. Eine Handhabung von Punkt zu Punkt dient zum Handhaben beispielsweise durch Ablegen eines Teils auf Werkstückträgern, zum Fügen z. B. durch Einlegen, Einpressen oder Herstellen von Schnappverbindungen, sowie zum Palettieren und Verpacken.

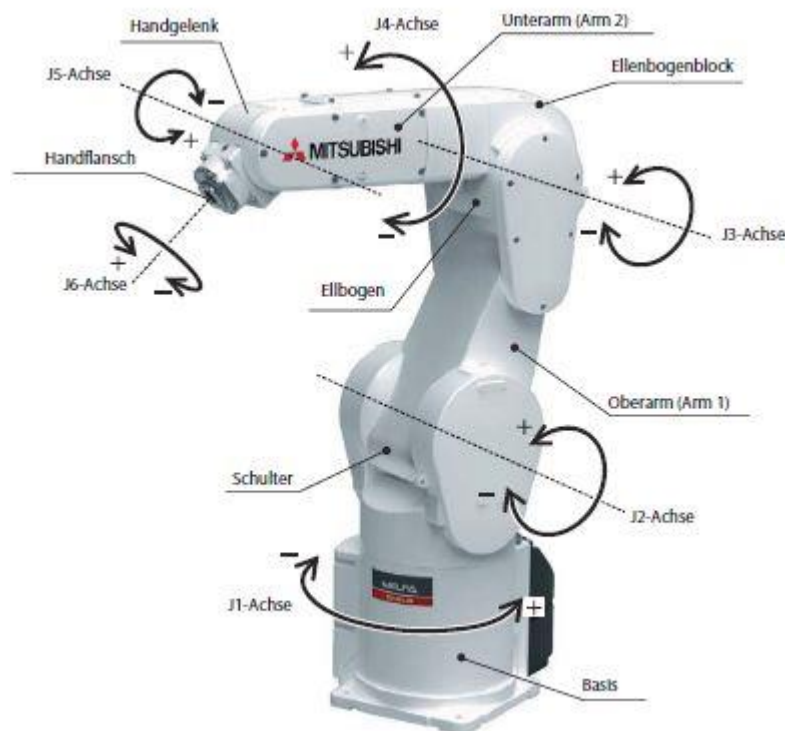
Als Abwandlung dieses Prinzips haben Doppelarm-SCARA Roboter eine Parallelkinematik aus zwei an den Enden verbundenen Gelenkarmen, die in einem kleinen Arbeitsfeld sehr hohe Geschwindigkeit und Positioniergenauigkeit ermöglichen



Doppelarm-SCARA Roboter mit Parallelkinematik

1.3 Vertikaler Knickarmroboter

Ein vertikaler Knickarmroboter besteht aus einer seriellen Folge von vertikalen und horizontalen rotatorischen Achsen mit den entsprechenden Gelenken. Der typische Aufbau sieht wie folgt aus:



Vertikaler Knickarmroboter mit 6 rotatorischen Achsen

Der in der Grafik gezeigte Roboter hat drei vertikale rotatorische Achsen, nämlich die Achsen J1, J4 und J6, und drei horizontale (nicht fluchtende) rotatorische Achsen, nämlich J2, J3 und J5.

Dieser Typ von Industrieroboter zeichnet sich dadurch aus, dass er bis auf singuläre Ausnahmen einen Punkt im Arbeitsraum mit jeder beliebigen Orientierung anfahren kann. Daher sind diese Roboter in der Fertigung sehr universell einsetzbar und haben zahlreiche Anwendungsbereiche wie

- Beschichten und Lackieren
- Schweißen (Punkt- und Bahnschweißen)
- Entgraten
- Kleben
- Flexible Montage
- Palettieren
- Verpacken
- Kabel verlegen
- usw.

Wir betrachten in diesem Arbeitsbuch vorwiegend 6-achsige vertikale Knickarmroboter.

■ Kontrollfragen

a) Bei welchen Aufgaben setzt man in der industriellen Praxis vorzugsweise Scara-Roboter ein?

Bei der Bestückung von Leiterplatten, da hierbei der Roboter nur in der Ebene arbeiten muss und Scara Roboter deutlich effektiver und schneller als Knickarmroboter sind.

b) Warum werden in der Fertigungsautomatisierung vorwiegend 6-achsige Knickarmroboter eingesetzt?

Allgemeine Aufgaben in der Fertigungsautomatisierung erfordern es, dass der Roboter Punkte und Bahnen im Arbeitsraum mit jeder beliebigen Orientierung an- bzw. abfahren kann. Dies wird durch 6-achsige vertikale Knickarmroboter gewährleistet.

1.4 Systemkomponenten eines Industrieroboters

Die wichtigsten Komponenten, aus denen ein Industrieroboter besteht, sind:

- **Kinematik der Achsen und Gelenke**
 Die Struktur eines Roboters ist durch die Kinematik seiner Achsen und Gelenke festgelegt. Ein wichtiger Begriff in der Kinematik ist der Begriff des **Freiheitsgrades** eines Körpers:
 Der Freiheitsgrad f ist die Anzahl der möglichen unabhängigen Bewegungen.
 Also hat ein im Raum frei beweglicher starrer Körper den Freiheitsgrad $f = 6$, der sich aus drei translatorischen (in x , y , z -Richtung) und drei rotatorischen (Drehung um x -Achse, Drehung um y -Achse, Drehung um z -Achse) Bewegungsmöglichkeiten zusammensetzt. Mit den translatorischen Bewegungen bestimmt man die Position (x , y , z) des Körpers und mit den rotatorischen Bewegungen die Orientierung (A , B , C) des Körpers im Raum. Folglich benötigt ein Roboter mindestens 6 Achsen, um ein Objekt beliebig im Raum zu positionieren und zu orientieren. In diesem Zusammenhang sagt man, dass ein Roboter in einem Punkt seines Arbeitsraums den Freiheitsgrad $f = 6$ hat, wenn er in der Nähe des Arbeitspunktes ein Objekt in jede beliebige Position mit einer beliebig vorgegebenen Orientierung bewegen kann. Vertikale Knickarmroboter mit 6 Achsen besitzen in fast allen Punkten des Arbeitsraumes den Freiheitsgrad $f = 6$, aber es gibt auch sogenannte **Singularitäten**, in denen der Roboter einen oder mehrere Freiheitsgrade verliert.
- **Steuerung**
 Im Wesentlichen koordiniert die Steuerung das Zusammenwirken der Roboterachsen. Neben dieser Hauptaufgabe ist die Steuerung auch für das Editieren, Speichern und Abarbeiten der Programme für die Prozessabläufe zuständig. Zusätzlich bietet die Steuerung i.a. zahlreiche Schnittstellen, um eine Kommunikation zu externen Sensoren, z.B. einer Kamera, oder anderen Steuerungssystem wie einer SPS aufzubauen und auszuwerten.
- **Programmierhandgerät (PGH)**
 Das PGH liefert die Schnittstelle für den Bediener, um mit dem Roboter zu kommunizieren. Über das PGH hat der Bediener im Allgemeinen die Möglichkeit, alle internen Daten der Robotersteuerung zu überwachen, Parametereinstellungen zu ändern und Programme zu optimieren und an die aktuellen Bedingungen anzupassen. Eine zentrale Aufgabe des Bedieners bei der Inbetriebnahme einer Roboteranwendung ist es, mit dem PGH die notwendigen Arbeitspunkte für den Bewegungsablauf zu teachen.
- **Endeffektor**
 Mit dieser Teilkomponente wird die eigentliche Handhabungsaufgabe durchgeführt. Man unterteilt die Effektoren in die beiden Hauptgruppen Greifer und Werkzeuge. Es gibt vom einfachen Greifer, der nur die beiden Zustände auf und geschlossen besitzt, bis hin zu komplexen Werkzeugen, die sensorgeführt eine Bearbeitung durchführen, eine breite Palette von Anwendungen.
- **Antriebssysteme**
 Industrieroboter besitzen die Möglichkeit, auf mathematisch definierten Bahnen im Raum zu fahren. Hierzu müssen im allgemeinen Fall Achsen getrennt bzgl. Weg und Geschwindigkeit geregelt werden. Dies bedeutet, dass für jede Achse eine eigene Antriebseinheit notwendig ist. Hierzu werden meistens AC- und DC-Servoantriebe eingesetzt.
- **Wegmesssysteme**
 Jede angetriebene Achse eines Industrieroboters besitzt ein eigenes Wegmesssystem, das in einen Positionsregelkreis integriert ist. Man unterscheidet absolute und inkrementale Wegmesssysteme. Bei inkrementalen Wegmesssystemen wird der gesamte Bewegungsbereich einer Achse in kleine Schritte – den Inkrementen – aufgeteilt. Bei der Bewegung der Roboterachse wird jeder Schritt durch eine elektronische Einrichtung registriert und in einen elektronischen Impuls umgewandelt. Der

zurückgelegte Weg der Roboterachse entspricht der Summe der Impulse. Ist die Ausgangsposition bekannt, kann aus der Zahl der Inkremente und deren Richtung die aktuelle Lage der Achse bestimmt werden. Bei absoluten Wegmesssystemen liefert das Messsystem für jede Position der Achse in ihrem ganzen Bewegungsbereich ein eindeutiges Signal, das unabhängig von der vorherigen Bewegung ist und auch beim Abschalten der Steuerung oder bei Störfällen nicht verloren geht. Absolute Messwertsysteme sind aber relativ groß und sehr teuer, daher werden in der Robotertechnik vorwiegend inkrementale Wegmesssysteme eingesetzt. Um das Problem zu umgehen, bei jedem Neustart die Ausgangsposition zu identifizieren, verwenden die meisten Hersteller folgende Strategie: Es werden Batterien eingesetzt, um die jeweils aktuellen Werte der Wegmesssysteme relativ zum Nullpunkt zu speichern.

- Peripherie wie Sensoren und Sicherheitseinrichtungen
Kaum eine industrielle Roboteranwendung kommt ohne externe Sensoren aus, die programmbeeinflussende Parameter zur Steuerung senden, wie z.B. Werkstück vorhanden oder nicht, Positionskorrekturen, Andrucksensor beim Schleifen, sensorgeführtes Schweißen usw.
Werker, die in der Nähe von oder mit Roboterapplikationen arbeiten, müssen vor den von diesen Anlagen ausgehenden Gefahren geschützt werden. Dazu gehören Sicherheitseinrichtungen, die vor Lichtbögen beim Schweißen schützen, aber auch Einrichtungen, die bei großen Massen und großen Geschwindigkeiten die Anlage zum sofortigen Stillstand führen, wenn eine Person in den Gefahrenbereich kommt. Ebenso gibt es Sicherheitseinrichtungen, die den Roboter selbst vor Defekten schützen, z.B. durch Sensoren, die bei Kollisionen die Bestromung der Antriebe abschalten oder drastisch reduzieren, um somit Beschädigungen des Roboters zu verhindern.

■ Kontrollfrage

- Wie kann erreicht werden, dass inkrementelle Wegmesssysteme wie absolute Messsysteme arbeiten?
 - Es wird bei der Inbetriebnahme des Roboters eine Nulllage definiert und über Batterien gespeichert. Im weiteren Betrieb kann der über inkrementelle Wegmesssysteme die relative Position zur Nulllage bestimmt und über die Batterien gespeichert werden, so dass bei jedem Neustart der Roboter seine Ausgangsposition kennt.

1.5 Kenngrößen eines Industrieroboters

Der VDI Richtlinie 2861 werden verschiedene Kenngrößen für Industrieroboter definiert, um Herstellern und Anwendern einheitliche Kriterien für die Bewertung unterschiedlicher Robotersysteme zur Verfügung zu stellen. Die Richtlinie teilt die Kenngrößen nach folgender Systematik auf:

Geometrische Kenngrößen	Belastungskenngrößen	Kinematische Kenngrößen	Genauigkeitskenngrößen
Mechanische Systemgrenzen	Nennlast	Geschwindigkeit	Wiederholgenauigkeit (Position und Orientierung)
Raumaufteilung	Maximallast	Beschleunigung	Wiederholgenauigkeit (Bahn)
Arbeitsbereich		Überschwingweite	
		Ausschwingzeit	
		Verfahrzeit	
		Zykluszeit	

Geometrische Kenngrößen

- Mechanische Systemgrenzen

Man unterteilt einen Industrieroboter in **Haupt- und Nebenachsen und Werkzeuge (Effektoren)**. Bei einem sechssachsigen vertikalen Knickarmroboter sind J1 bis J3 die Hauptachsen. Sie dienen dazu, die Position anzufahren. Die Achsen J4 bis J6 sind die Nebenachsen. Sie dienen dazu, die Orientierung des Handhabungsobjektes im Raum festzulegen. Die Schnittstelle zwischen der Basis des Industrieroboters und der Arbeitsumgebung ist in der Regel der untere Flansch des Roboters, mit dem er entweder auf dem Boden oder an der Wand oder an der Decke verschraubt wird. Der Mittelpunkt dieser Flanschfläche wird in der Regel als der Nullpunkt des Basiskoordinatensystems für den Roboter festgelegt. Die Schnittstelle zwischen den Haupt- und Nebenachsen liegt beim sechssachsigen Knickarmroboter im Flansch oder Gelenk zwischen den Achsen J3 und J4. Daneben gibt es eine weitere Schnittstelle, die zwischen den Nebenachse und dem Endeffektor liegt. Sie befindet sich in der Regel im Anschlussflansch für den Endeffektor und wird als Nullpunkt des **Tool Center Points (TCP)** festgelegt.

- Raumaufteilung

Der **Hauptarbeitsraum** ist derjenige Teil des Arbeitsraumes, der von der Schnittstelle Haupt-/Nebenachsen gebildet wird, indem man alle Hauptachsen in ihre jeweilige Maximal- und Minimalstellungen verfährt.

Der **Nebenarbeitsraum** ist entsprechend dem Hauptarbeitsraum für die Nebenachsen definiert.

Der **Arbeitsraum** ist die Summe von Hauptarbeitsraum und Nebenarbeitsraum. Er ist also der Raum, der von der Schnittstelle Nebenachse/Endeffektor (TCP) mit der Gesamtheit aller Achsbewegungen erreicht werden kann.

- Arbeitsbereich

Der **nicht nutzbare Raum** eines Industrieroboters entsteht durch Gelenke, Kabel oder Achsbauteile, die beim Fahren im Raum mitbewegt werden müssen, dadurch auch mit dem Umfeld (Mensch oder

Hardware) kollidieren können und somit eine erhebliche Gefahr darstellen.

Der **feste Bewegungsraum** ist die Summe aus Arbeitsraum und nicht nutzbarem Raum. Er ist somit der Raum, der von allen bewegten Komponenten des Industrieroboters mit der Gesamtheit aller Achsbewegungen beschrieben wird.

Der **variable Bewegungsraum** eines Industrieroboters ist der Raum, der durch die Bewegungen des Endeffektors erzeugt wird. Dieser Raum ist variabel, da die Abmessungen des Endeffektors unbestimmt sind.

Der **Arbeitsbereich** ist die Summe aus variablen und festen Bewegungsraum. Er ist unter sicherheitstechnischen Aspekten identisch mit dem Gefahrenraum.

Belastungskenngrößen

- Nennlast

Die Trennstelle zwischen dem Eigengewicht des Industrieroboters und der zu bewegend Last ist in der Regel der Anschlussflansch an der letzten Achse. Man unterscheidet folgende Begriffe:

Die **Werkzeug-/Effektor-/Greiferlast** ist die Last, die entweder als Werkzeug oder als Greifer am Anschlussflansch der letzten Achse angebracht wird.

Die **Nutzlast** ist die Last, die zusätzlich zur Werkzeuglast bewegt werden kann, ohne dass dabei Einschränkungen an die maximalen Geschwindigkeiten oder maximalen Beschleunigungen einer Achse gemacht werden müssen. Ebenso dürfen auch keine Einschränkungen bezüglich der Genauigkeit und des Arbeitsbereiches erfolgen.

Die **Nennlast** ist die Last, die sich als Summe aus der Werkzeuglast und der Nutzlast berechnet.

Die **zusätzliche Nutzlast** ist die Last um die die Nutzlast überschritten werden darf, wobei hierbei aber Einschränkungen hinsichtlich Genauigkeit, maximale Geschwindigkeit und Beschleunigung oder maximaler Arbeitsbereich des Industrieroboters gemacht werden müssen.

- Maximallast

Die **maximale Nutzlast** ist die Summe aus Nutzlast und zusätzlicher Nutzlast.

Die **Maximallast** ist die Summe aus maximaler Nutzlast und Werkzeug-, Effektor- bzw. Greiferlast.

Kinematische Kenngrößen

- Geschwindigkeit

Die maximalen Geschwindigkeiten und maximalen Beschleunigungen der Achsen eines Industrieroboters sind die Faktoren, die die Zykluszeit einer Handhabungsaufgabe hauptsächlich bei Einhaltung der geforderten Genauigkeit bestimmen. Daneben unterscheidet man folgende Begriffe: Unter der **Bahngeschwindigkeit** versteht man die Geschwindigkeit, mit der TCP entlang einer definierten Bahn im Umgebungsraum bewegt wird.

- Überschwingweite

Die **Überschwingweite** gibt an, um wieviel mm oder Grad eine anzufahrende Position oder Orientierung überschwingt.

- Ausschwingzeit

Die **Ausschwingzeit** gibt an, wie lange der Einschwingvorgang in die Sollposition nach dem ersten Erreichen der Sollposition dauert.

- **Verfahrzeit**
Die **Verfahrzeit** ist die Zeit, die ein Roboter bei Nennlast benötigt, um eine Sollposition zu erreichen.
- **Zykluszeit**
Die **Zykluszeit** ist die Zeit, die ein Roboter benötigt, um einen vollständigen Ablauf bei Nennlast durchzuführen.

Genauigkeitskenngrößen

- **Wiederholgenauigkeit**
Genauigkeitskenngrößen von Industrieroboter sind **Wiederholgenauigkeiten**, d.h. sie machen eine Aussage darüber, wie genau der Industrieroboter eine zuvor bekannte Position erneut anfährt und welche Genauigkeit beim Nachfahren einer Bahn im Umgebungsraum erreicht wird. Die Genauigkeitskenngrößen werden achsunabhängig und unter Nennlast ermittelt. Die Kenngrößen müssen für den gesamten Arbeitsraum gelten.
Zur Ermittlung der Wiederholgenauigkeit beim Positionieren und Orientieren müssen nach der VDI Richtlinie mindestens 30 Messorte nach einer empfohlenen Verteilung im Arbeitsraum mehrfach aus verschiedenen Richtungen unter Nennlast angefahren werden. Hieraus wird dann die mittlere Standardabweichung bzgl. Positionierung und Orientierung ermittelt. Die Aussage „Der Industrieroboter hat eine Wiederholgenauigkeit von $\pm 0.02\text{mm}$ “ besagt, dass ein Roboter eine bekannte Position stets so anfährt, dass der Abstand zur Ausgangsposition maximal 0,02 mm beträgt. Diese Aussage muss auch bei maximaler Achsgeschwindigkeit unter Nennlast gelten, wobei dabei natürlich die entsprechenden Ausschwingzeiten berücksichtigt werden müssen.
- **Absolute Genauigkeit**
Es ist wichtig zu beachten, dass die Wiederholgenauigkeit keine Aussage über die reale Genauigkeit des Roboters in seinem Arbeitsraum macht. Mit anderen Worten: Falls die Raumkoordinaten eines Roboters z.B. (300.5, -400, 280.5) betragen, dann würde es theoretisch bedeuten, dass der TCP vom Nullpunkt des Roboter- Basiskoordinatensystems einen Abstand von 573,57 mm hat. Tatsächlich weicht dieser Wert in der Realität vielleicht mehr als 1 mm ab. Diese Abweichung hängt von zahlreichen Faktoren wie z.B. der Wärmeentwicklung des Systems und der Umgebung oder der Länge des Verfahrwegs ab.
- **Wiederholgenauigkeit bei Bahnen**
Viele der heutigen Anwendungen verlangen von einem Industrieroboter eine Bahnsteuerung wie z.B. für das Schweißen entlang einer vorgegebenen Naht. Hier muss die Schweißpistole mit definierter Geschwindigkeit eine definierte Bahn im Raum mit hoher Genauigkeit abfahren, um die Güte der Schweißnaht zu gewährleisten. Hierfür liefert die zuvor erwähnte Positionier- und Orientierungsgenauigkeit kein Bewertungskriterium. Die VDI Richtlinie 2861 definiert hierzu zahlreiche Kenngrößen wie mittlerer Bahnabstand, mittlere Bahn-Orientierungsabweichung, mittlerer Überschwingfehler usw.

■ Kontrollfragen

- a) Was besagt die Aussage, dass die Nennlast eines Roboters 2 kg beträgt?

Die Nennlast ist die Summe aus Werkzeuglast und Nutzlast, wobei diese Last weder eine Einschränkung auf die maximale Geschwindigkeit oder Beschleunigung der Achsen noch auf die Wiederholgenauigkeit haben darf.

- b) Was versteht man unter der Zykluszeit eines Roboters?

Die Zykluszeit eines Roboters ist die Zeit, die der Roboter für einen fest definierten Arbeitsablauf benötigt.

- c) Erklären Sie den Begriff Wiederholgenauigkeit eines Roboters?

Die Wiederholgenauigkeit ist die Genauigkeit, mit der der Roboter eine eingelernte Position inkl. Orientierung aus einer beliebigen Richtung im Arbeitsraum mit Maximalgeschwindigkeit anfahren kann.

2 Greifer

Ein Effektor ist Teil des Robotersystems, der die eigentliche Handhabungsaufgabe ausführt. Er wird am Flansch der letzten Roboterachse befestigt und an Energie- und Steuerleitungen angeschlossen. Je nach Handhabungsaufgabe kommen noch Sensoren, Fügehilfen, zusätzliche Leitungen (Schweißstrom) oder bei Lackierrobotern Schläuche für die Farbzufuhr hinzu.

Dieses Kapitel beschränkt sich auf Greifer, die vorwiegend in Handhabungsaufgaben als Effektor eingesetzt werden. Der Greifer lässt sich in weitere Subsysteme untergliedern:

- Wirk- oder Haltesystem: Übertragung der Greifkraft auf das Greifobjekt.
- Trägersystem am Flansch des Roboters.
- Antriebssystem: Bereitstellung der Energie zur Erzeugung der Greifkraft.
- Kinematisches System: Die Aufgabe der Kinematik ist es, die Ausgangsbewegung des Antriebs in eine Bewegung des Haltesystems umzusetzen.
- Steuerungssystem: Auswertung von Sensorinformationen, Greifkraftregulierung und automatische Greifweiteneinstellung.
- Sensorsystem: Objekt- und Materialerkennung, Greifkraftbestimmung.
- Schutzsystem: Kollisionskontrolle.

2.1 Mechanische Greifer

Greifer stellen entweder eine kraftschlüssige oder formschlüssige Verbindung zum Handhabungsobjekt her. Ein Antrieb ist nur notwendig, wenn die Greifkraft durch ein kinematisches System erbracht wird. Die mechanischen Greifer bilden diese Klasse von Greifern. Sie stellen eine formschlüssige Verbindung zum Objekt her, während z.B. Sauggreifer eine kraftschlüssige Verbindung herstellen. Folgende Antriebe werden genutzt:

- Pneumatisch
- Hydraulisch
- Elektromagnetisch
- Elektromotorisch

In nachfolgender Grafik sind verschiedene Greifertypen dargestellt. Alle Greifertypen besitzen einen doppelwirkenden Kolbenantrieb und sind selbstzentrierend. Berührungslose Positionserkennung ist mit Näherungsschaltern möglich. Durch externe Greiffinger sind die Greifer vielseitig einsetzbar.



Parallel-Greifer, Winkel-Greifer, Radial-Greifer, 3-Punkt-Greifer

Pneumatische Greifer – Fotos und Symbole

Ganz neue Maßstäbe setzen Greifer, die den Fin Ray Effect® nutzen (siehe nachfolgende Grafik). Sie passen sich der Kontur des zu greifenden Objektes an und können deshalb multifunktional eingesetzt werden. Dadurch wird sicheres und zerstörungsfreies Greifen selbst von zerbrechlichen oder unregelmäßig geformten Werkstücken möglich.



Pneumatischer Greifer mit Fin Ray Effect®

2.2 Sauggreifer

Sauggreifer arbeiten nach dem Prinzip der Vakuumerzeugung, d.h. zwischen Objektoberfläche und Saugnapf wird ein Vakuum erzeugt, mit dem das Objekt gehalten wird. Es handelt sich hier also um eine rein kraftschlüssige Verbindung.

Folgende Arten werden nach dem Prinzip der Vakuumerzeugung unterschieden:

- **Vakuumsauger:** Unterdruck wird mittels Vakuumpumpe erzeugt
- **Luftstromsauger:** Unterdruck wird durch Venturidüsen und Druckluft erzeugt
- **Haftsauger:** Die Luft zwischen Saugnapf und Objekt wird durch Aufpressen des Saugers auf das Objekt ausgepresst. Um das Vakuum aufzuheben, wird ein Druckimpuls benötigt. Haftsauger sind nur bei sehr glatten und luftundurchlässigen Objektoberflächen wie z.B. Glas einsetzbar.

Die Haltekräfte der Sauggreifer sind abhängig von

- der Objektoberfläche,
- der Deformierung des Saugers,
- dem Druckdurchfluss.

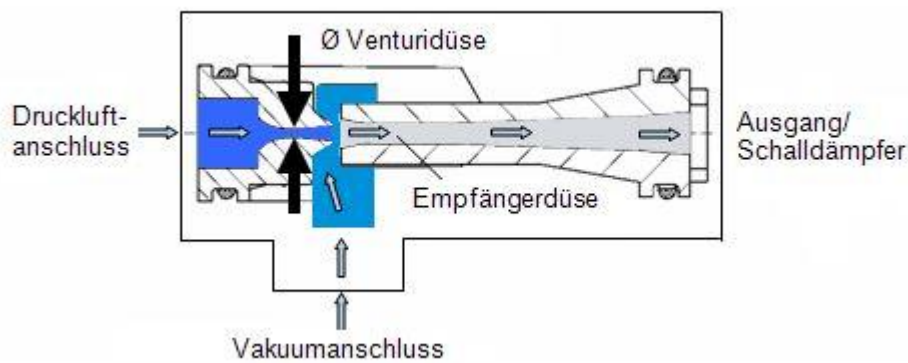
Saugnäpfe erlauben das Handhaben von unterschiedlichen Werkstücken mit Gewichten von nur wenigen Gramm bis mehreren hundert Kilogramm. Es gibt sie in den unterschiedlichsten Formen, z. B. Universal-, Flach-, oder Faltenbalgsaugnäpfe.



Sauggreifer

Nachfolgender Abbildung verdeutlicht das Venturiprinzip der Vakuumerzeugung mittels Ejektor. Die Druckluft strömt vom Druckanschluss in den Ejektor. Durch die Querschnittsverengung in der Venturidüse wird

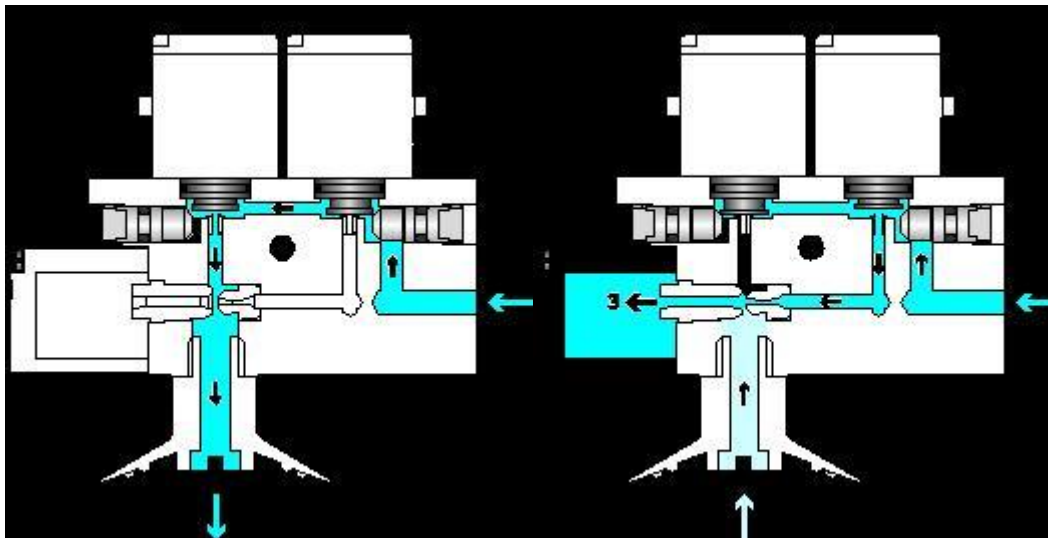
die Strömungsgeschwindigkeit der Luft auf Überschallgeschwindigkeit erhöht. Nach dem Austritt aus der Venturidüse expandiert die Luft und strömt durch die Empfängerdüse in den Ausgang (Schalldämpfer). Dabei entsteht in der Kammer zwischen Venturi - und Empfängerdüse ein Vakuum, das dazu führt, dass Luft vom Vakuumananschluss angesaugt wird.



Funktionsweise einer pneumatischen Vakuumsaugdüse

Im folgenden Bild ist die Funktionsweise einer elektropneumatischen Vakuumsaugdüse dargestellt. Das linke Bild zeigt den Betriebszustand “Ansaugen”. Das elektrisch betätigte 2/2-Wegeventil -QM2 ist geöffnet. Die Druckluft strömt vom Anschluss 1 durch die Strahldüse zum Schalldämpfer 3. Dadurch wird am Sauger 2 ein Unterdruck erzeugt, und das Werkstück wird angesaugt.

Das rechte Bild zeigt den Betriebszustand “Abblasen”. Das 2/2-Wegeventil -QM1 ist geöffnet. Die Druckluft wird direkt dem Sauger zugeführt. Durch einen Druckstoß vom Anschluss 1 über das 2/2-Wegeventil -QM1 werden die angesaugten Werkstücke schneller vom Sauger abgestoßen.



Funktionsweise einer elektropneumatischen Vakuumsaugdüse

2.3 Greiferwechselsysteme

Soll die große Flexibilität der Industrieroboter sinnvoll ausgenutzt werden, sind auch flexible Greifersysteme erforderlich. Greifer mit hoher Flexibilität sind in der Regel technisch aufwendig und somit häufig zu teuer. Als Alternative können Wechselsysteme eingesetzt werden, die es ermöglichen einfache Greifer von verschiedenem Typus einzusetzen.

Ein Greiferwechselsystem ist eine Vorrichtung zum Austausch von Werkzeugen und Greifern am Flansch des Roboterarms. Das Wechselsystem besteht aus einem am Roboterflansch montierten Oberteil und mindestens zwei Unterteilen zur Aufnahme von Werkzeugen oder Greifern. Unterteil und Oberteil verriegeln beim Werkzeugwechsel miteinander. Die Bewegungen beim Verriegeln und Entriegeln werden vom Roboter selbst oder von einer zusätzlichen Leistungseinheit ausgeführt.

■ Kontrollfrage

- Nennen Sie ein Vorteil und ein Nachteil eines Greiferwechselsystems?
 - Vorteil: Es können standardisierte und daher kostengünstigere Greifersysteme eingesetzt werden.
 - Nachteil: Die Zykluszeit bei der Bearbeitung wird größer, da der Roboter den Greifer während der Bearbeitung wechseln muss.

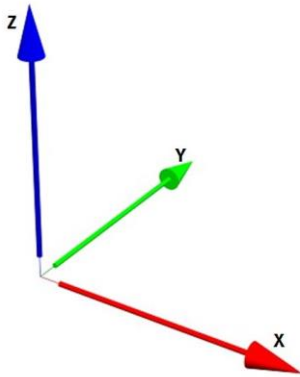
3 Robotersteuerung

Die Hauptaufgaben jeder Robotersteuerung sind:

- Koordinatentransformation
- Achssteuerung (Lage und Geschwindigkeit)
- Bahninterpolation (linear, kreisförmig, Spline-Interpolation)
- Kommunikation mit dem **PGH** (Handbediengerät, Teachpanel, Teach Box)
- Programmverwaltung (Editor, Speicherverwaltung)
- Kommunikation mit den Schnittstellen (Ein- und Ausgänge)

3.1 Kartesische Koordinatensysteme

Zur Beschreibung der Lage und Orientierung eines Körpers im Raum wird das 3-dimensionale kartesische Koordinatensystem benutzt:



Kartesisches Koordinatensystem

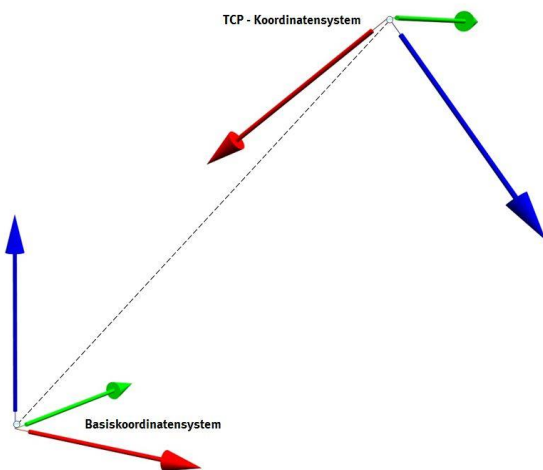
Wir benutzen im Text durchgehend folgende farbliche Markierung der Koordinatenachsen:

Rot = x-Achse

Grün = y-Achse

Blau = z-Achse

Für die weiteren Berechnungen ist es notwendig ein festes **Referenz-Koordinatensystem** für den Raum zu wählen. Um die **Lage** (= Position und Orientierung) eines starren Körpers in diesem Raum zu beschreiben wird der Körper mit einem eigenen Koordinatensystem – das **Objekt-Koordinatensystem** – versehen:



Koordinatentransformation

Man definiert nun:

- Der Körper ist in Position P mit den Koordinaten (x_P, y_P, z_P) , falls der Nullpunkt des Objekt-Koordinatensystems identisch mit P ist. Der Körper hat im Punkt P die Orientierung $O = (A_P, B_P, C_P)$, falls die Orientierung des Objektkoordinatensystems aus dem Referenz-Koordinatensystem durch Drehung um die x-Achse mit dem Winkel A_P , um die y-Achse mit dem Winkel B_P und um die z-Achse mit dem Winkel C_P hervorgeht.

Mathematisch lässt sich die Lage des Körpers, d.h. des Objektkoordinatensystems, durch eine recht einfache Transformationsmatrix beschreiben. Der Vorteil ist nun, dass es zur Beschreibung der Bewegung des Körpers ausreichend ist, die Bewegung im körpereigenen Objektkoordinatensystem festzulegen.

3.2 Roboter-Koordinatensysteme

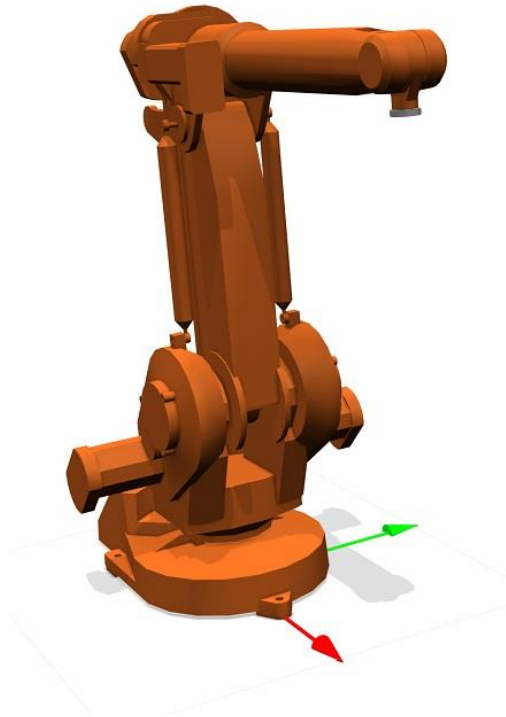
Maßgeblich für die Lage des Roboters im Raum, ist der **Tool-Center-Point (TCP)**. Je nach Anwendung kann der Roboter verschiedene TCP's haben. Ist der TCP festgelegt, dann ist die Lage natürlich eindeutig durch die Werte der Achsen bestimmt. Bei einem 6-achsigen Knickarmroboter ist der TCP durch die sechs Winkelwerte $(\alpha_1, \dots, \alpha_6)$ bestimmt. Nehmen wir jetzt zur Vereinfachung an, dass der TCP im Nullpunkt des Raum-Koordinatensystems liegt. Dann wird die scheinbar **einfache** Aufgabenstellung, die Winkelwerte anzugeben, um den TCP in x-Richtung um 40 cm zu bewegen, für das menschliche Gehirn schon zu einer **unlösaren** Aufgabe. Daher ist es notwendig, eine Transformation des Achsen-Koordinatensystems bestehend aus den Achswerten des Roboters in die kartesische Koordinatenwelt zu finden.

In der Praxis werden folgende verschiedene Typen von kartesischen Koordinatensystemen zur Beschreibung einer Anwendung eingesetzt:

- Welt-Koordinatensystem
- Roboter Basiskoordinatensystem
- Tool Center Koordinatensystem oder Werkzeug-Koordinatensystem
- Objekt-Koordinatensystem

Zunächst ist das Referenzkoordinatensystem für die Anwendungsumgebung festzulegen. Dieses Koordinatensystem wird auch das **Welt-Koordinatensystem** genannt. Jeder Roboter im Anwendungsbereich

erhält ein **Basis-Koordinatensystem**. Der Nullpunkt des Basis-Koordinatensystems liegt in der Regel im Mittelpunkt des Roboterfußes. Die Ausrichtung der Achsen erfolgt herstellerspezifisch:



Das Roboter Basiskoordinatensystem

Zur Festlegung des **TCP** ist es sinnvoll einen **Referenz-TCP** festzulegen. Ein **TCP** besteht aus einem kartesischen Koordinatensystem, wobei der Nullpunkt der sogenannte **Tool Center Point** ist. Bei Knickarmrobotern ist in der Regel der Nullpunkt des **Referenz-TCP** der Mittelpunkt des Endflansches vom Roboter. Das Koordinatensystem des **Referenz-TCP** wird so gewählt, dass bei Nullstellung aller Achsen des Roboters es aus einer reinen Translation des Basis-Koordinatensystems hervorgeht. Jeder andere TCP entsteht dann aus einer Translation und einer Verdrehung des Referenz-TCP Koordinatensystem. Bei einer Anwendung wird man in der Regel den TCP in den Mittelpunkt des Greifers oder in die Spitze eines Werkzeugs legen. Aus obiger Definition über die Lage eines Körpers im Raum können wir folgende Festlegung ableiten:

- Ein Roboter ist in Position P mit den Koordinaten (x, y, z) und der Orientierung (A, B, C) genau dann, wenn der Nullpunkt des **TCP-Koordinatensystems** identisch mit P ist und wenn die Orientierung des **TCP-Koordinatensystems** aus einer Drehung des Basis-Koordinatensystems um die x-Achse mit dem Winkel A, um die y-Achse mit dem Winkel B und um die z-Achse mit dem Winkel C hervorgeht.

Ist die Transformation zwischen **Basis-Koordinatensystem** und **TCP-Koordinatensystem** festgelegt, so kann der Anwender die Aufgabenbeschreibung im TCP Koordinatensystem durchführen. Die Robotersteuerung übernimmt die Aufgabe zur Umrechnung in das Basis-Koordinatensystem und zur weiteren Umrechnung in das Achsen-Koordinatensystem des Roboters. Hierbei unterscheidet man zwei Transformationen:

- **Vorwärtstransformation**

Die Vorwärtstransformation berechnet aus den Achswerten die Lage des TCP im Roboter Basiskoordinatensystem. Mit Hilfe der **Denavit-Hartenberg** Parameter, die eine mathematische Beschreibung der Dimension und Funktionalität (rotatorisch oder translatorisch) der Achsen und ihrer Lage zueinander liefert, lässt sich mit überschaubarem Aufwand eine mathematische Transformationsformel zur Berechnung ableiten und recht einfach in die Steuerung implementieren.

- **Rücktransformation**

Die Aufgabe der Rücktransformation ist es aus den kartesischen Positions- und Orientierungsdaten des TCP die zugehörigen Achsstellungen zu bestimmen. Im Falle eines 6-achsigen Knickarmroboters zeigt es sich, dass es keine analytische Lösung für das Problem gibt, sondern nur verschiedene numerische Verfahren zu einer Lösung führen. Eine Hauptschwierigkeit liegt darin, dass es im allgemeinen mehrere Lösungen und in bestimmten Lagen des TCP sogar unendlich viele Lösungen gibt. Zur Implementierung der Rücktransformation nutzt jeder Hersteller seine eigene Optimierung von speziellen numerischen Lösungsverfahren.

Hier im Arbeitsbuch beschränken wir uns auf den Fall, dass jedes Objekt in der Anwendungsumgebung als starrer Körper betrachtet werden kann. Dann kann jedes Objekt mit einem körpereigenen Objekt-Koordinatensystem versehen werden. Sobald die Transformation zwischen Basis-Koordinatensystem und Objekt-Koordinatensystem festgelegt ist, besteht für den Anwender der Vorteil, die Bearbeitung des Objektes durch den Roboter **nur** in dem Objekt-Koordinatensystem beschreiben zu müssen. Die Umrechnung in das Achsen-Koordinatensystem wird von der Steuerung übernommen.

■ Kontrollfrage

- a) Erläutern Sie, was es bedeutet, dass ein Roboter in einer Position P mit den Koordinaten (x,y,z) und der Orientierung (A,B,C) ist.

Ein Roboter ist in Position P mit der vorgegebenen Orientierung genau dann, wenn der Nullpunkt des TCP vom Roboter die Position P hat und die Orientierung des TCP mit der vorgegebenen Orientierung im Punkt P übereinstimmt.

- b) Was berechnet die Vorwärtstransformation?

Die Vorwärtstransformation berechnet aus den Winkelwerten der Roboterachsen die Position und die Orientierung des beim Roboter eingestellten TCP.

3.3 Steuerungsarten der Roboterbewegung

Wir beschränken uns hier auf die Betrachtung von sechssachsigen vertikalen Knickarmrobotern. Es gibt zwei grundsätzlich verschiedene Arten der Roboterbewegung:

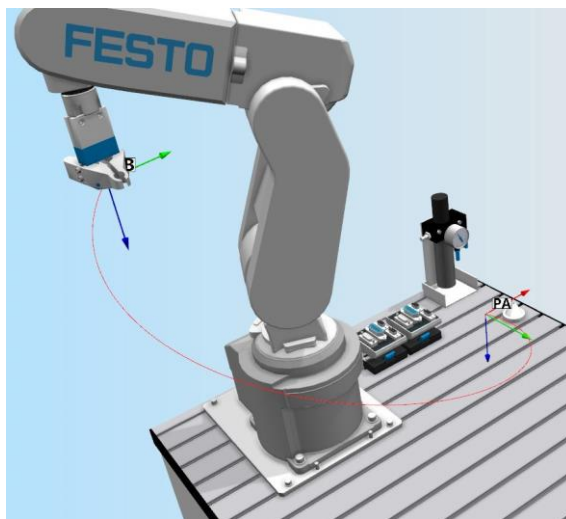
- PTP –Steuerung
- Bahnsteuerung

- PTP Steuerung

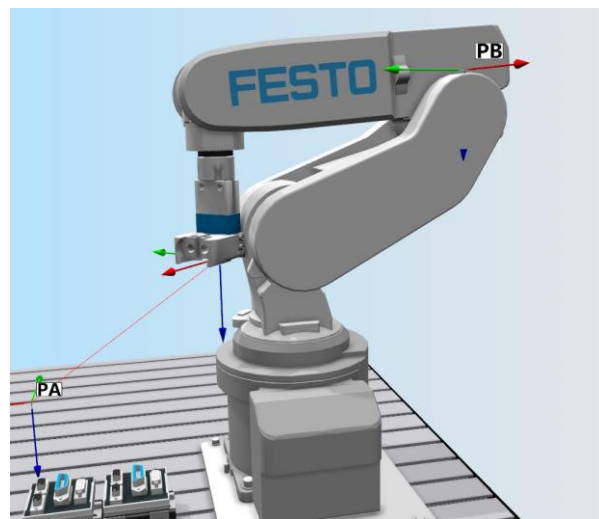
PTP ist die Abkürzung für **Point-to-Point**. Die Steuerungsart besagt folgendes:

Angenommen der Roboter befindet sich in einer Position PA mit einer vorgegebenen Orientierung O_A . Das Ziel ist es den Roboter in eine Position PB mit der Orientierung O_B mit einer vorgegebenen Maximalgeschwindigkeit v der Achsen zu verfahren. Die Lage des Roboters in PA sei durch die sechs Achswinkeln ($\alpha_1, \dots, \alpha_6$) und in PB durch die Achswinkeln (β_1, \dots, β_6) gegeben. Die Steuerung führt nun für jede Achse eine lineare Interpolation zwischen den beiden zugehörigen Achswerten über eine berechnete Zeit t_0 durch, so dass alle Achsen nach t_0 Sekunden den Zielwert erreichen, aber keine Achse sich schneller als die vorgegebene Maximalgeschwindigkeit bewegen muss.

Die Grafik 3.4 zeigt ein Beispiel für die Bahnbewegung eines Roboters bei einer PTP Steuerung von PA nach PB . Diese Bahnbewegung ist dem Anwender im Detail nicht bekannt und kann zu kritischen Kollisionen führen. Daher sind PTP Bewegungen in einer Anwendung nur dann sinnvoll nutzbar, wenn sichergestellt wird, dass keine Hindernisse in dem Bewegungsraum auftreten können.



Die Bahnbewegung bei einer PTP- Steuerung



Lineare Bahnsteuerung

Der Vorteil einer PTP-Steuerung ist, dass in jedem Fall der Roboter den Zielpunkt erreicht, falls natürlich nicht ein Hindernis den Weg versperrt.

- Bahnsteuerung

Im Fall einer Bahnsteuerung ist dies nicht zwingend erfüllt. Wollen wir z.B. im obigen Beispiel, dass der

Roboter auf einer linearen Bahn von PA nach PB fährt, dann passiert folgendes:

Die Robotersteuerung berechnet die lineare Bahn von PA nach PB und die notwendige Bahngeschwindigkeit. Dann werden in einem festen Interpolationstakt online die Achswerte für den nächsten Zwischenpunkt berechnet und der Roboter wird in diese Achsstellung verfahren.

Im obigen Beispiel fährt der Roboter los, aber plötzlich kann er nicht weiter, weil entweder eine Kollision erfolgt oder gewisse Achsstellungen nicht erreicht werden können oder nicht zulässig sind.

Jede glatte Bahnkurve im Raum kann durch zahlreiche lineare Teilstücke approximiert werden, aber es ist oft sehr viel effektiver verschiedene Interpolationsarten zu verwenden, wie z.B. die **kreisförmige Interpolation**. Dies ist eine Methode mit Hilfe von Zwischenpunkten und Kreisabschnitten eine glatte Bewegungsbahn zu beschreiben. Eine Bahnsteuerung ist extrem rechenaufwendig und ist erst durch die gewaltige Leistungssteigerung der Rechner in der Vergangenheit möglich geworden.

■ Kontrollfrage

- Nennen Sie einen Vorteil und einen Nachteil einer PTP Bewegung.
 - Vorteil: Der Roboter führt stets eine PTP Bewegung zum Zielpunkt aus.
 - Nachteil: Die Bewegungsbahn des Roboters ist bei einer PTP Bewegung nicht bekannt und kann daher zu unerwarteten Kollisionen führen

4 Sensoren

Jeder industrielle Roboter besitzt zahlreiche interne Sensoren wie z.B. zur Lage- und Geschwindigkeitsregelung der Achsen oder zur Überwachung der Bestromung der Antriebe. Zusätzlich benötigt nahezu jede industrielle Roboteranwendung zusätzliche externe Sensoren zur Überwachung des Prozessablaufs. Hierbei unterscheidet man

- Messende Sensoren (z.B. Kamera)
- Binär erfassende Sensoren (taktiler oder berührungsloses Erfassen)

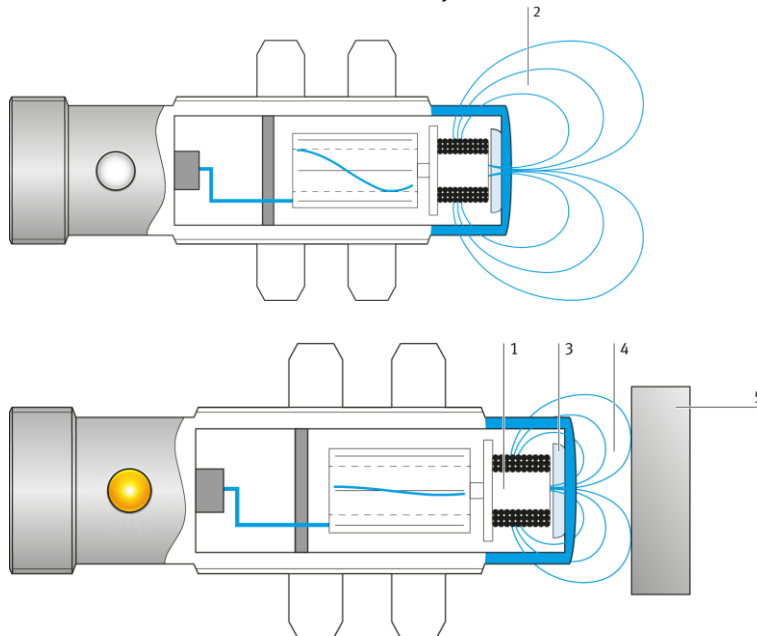
Bei berührungslos binär erfassende Sensoren unterscheidet man optische, kapazitive, induktive und Ultraschallsensoren.

4.1 Induktive Näherungsschalter

Mit induktiven Näherungsschaltern lassen sich nur elektrisch leitfähige Materialien detektieren. Beim Einschalten der Betriebsspannung des induktiven Näherungsschalters schwingt der eingebaute Oszillator an, es fließt ein definierter Ruhestrom.

Tritt nun ein elektrisch leitfähiges Material in die aktive Schaltzone ein, so werden in diesem Wirbelströme erzeugt. Hierdurch wird dem Oszillator Energie entzogen. Die Schwingungen werden gedämpft, es kommt zu einer Änderung der Stromaufnahme des Näherungsschalters.

Die beiden Zustände **Oszillator bedämpft** oder **Oszillator nicht bedämpft** werden elektronisch ausgewertet.



Wirkungsweise eines induktiven Näherungsschalters

- **Schaltabstand**

Je nach Schaltertyp (Schließer, Öffner oder Wechsler) wird die Ausgangsstufe bei der Anwesenheit eines Metalls in der aktiven Schaltzone durchgeschaltet oder gesperrt.

Der Abstand zur aktiven Fläche, bei der ein Signalwechsel des Ausgangssignals erfolgt, bezeichnet man als Schaltabstand.

Der Schaltabstand bei induktiven Näherungsschaltern wird von dem Metallwerkstoff des zu detektierenden Werkstücks beeinflusst. Während gleiche Metalle zu denselben Schaltabständen führen, ändert sich der Schaltabstand, sobald ein anderes Metall zum Einsatz kommt.

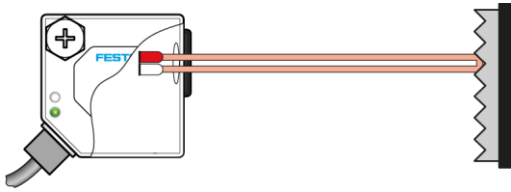
Erst durch den Reduktionsfaktor sind Vergleiche von Schaltabständen sinnvoll möglich. Der Reduktionsfaktor erlaubt die Abschätzung des Schaltabstands, wenn ein anderes Material als Stahl S 235 JR erfasst werden soll.

Material	Reduktionsfaktor
Stahl S 235 JR	1,0
Chrom-Nickel	0,70 – 0,90
Messing	0,35 – 0,50
Aluminium	0,35 – 0,50
Kupfer	0,25 – 0,40

4.2 Optische Näherungsschalter

Optische Näherungsschalter setzen optische und elektronische Mittel zur Objekterkennung ein. Dazu wird rotes oder infrarotes Licht verwendet. Rotes Licht wird bei Versionen mit Lichtleiteranschluss eingesetzt und hat den Vorteil, dass es bei der Justierung der optischen Achsen der verwendeten Näherungsschalter mit bloßem Auge erkannt werden kann. Infrarotes (nicht sichtbares) Licht wird bei Versionen ohne Lichtleiteranschluss verwendet, wo höhere Lichtleistungen benötigt werden, um z. B. größere Strecken zu überbrücken.

Optische Näherungsschalter bestehen grundsätzlich aus zwei Hauptbaugruppen – dem Sender und dem Empfänger. Diese sind entweder in einem gemeinsamen Gehäuse untergebracht (Reflex-Lichttaster und Reflex-Lichtschranken), oder befinden sich in getrennten Gehäusen (Einweg-Lichtschranken).



Reflexlichtschranke



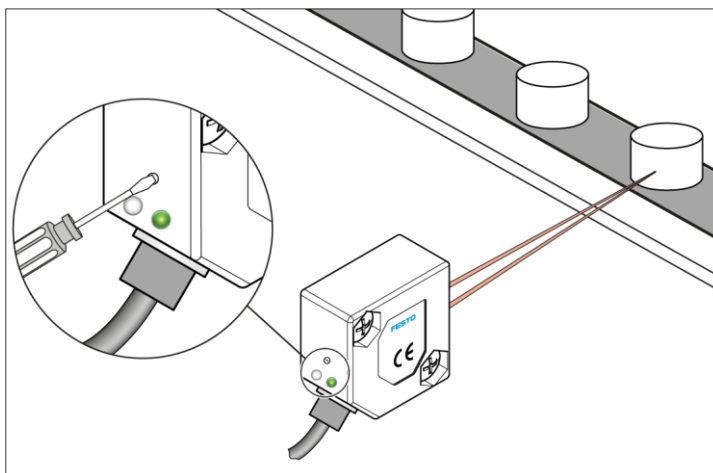
Einweglichtschranke

Besonders zuverlässige Quellen für rotes und infrarotes Licht sind Halbleiter-Leuchtdioden (LEDs), welche beim Durchgang von elektrischem Strom in Durchlassrichtung optische Strahlung aussenden. Sie sind klein und robust, von langer Lebensdauer und einfach modulierbar.

Als Empfangselemente werden Fotodioden oder Fototransistoren eingesetzt.

Die Tastweite von Reflex-Lichttastern hängt unter anderem von der Größe des Objekts ab, da ein großes Objekt mehr Licht reflektiert. Ebenso reflektieren helle Farben das Licht besser als dunkle Farben.

- **Empfindlichkeit optischer Näherungsschalter**
Die Einstellung der Empfindlichkeit an einem optischen Näherungsschalter ist wichtig, um Objekte mit geringem Kontrast oder unter erschwerten Bedingungen zuverlässig erkennen zu können. Die Empfindlichkeit lässt sich mit einem Schraubendreher am Gehäusedeckel einstellen.



Einstellung der Empfindlichkeit

4.3 Kamera

In der modernen Automatisierung von Produktion und Qualitätssicherung nehmen Bildverarbeitungslösungen in die Montagelinien immer häufiger eine zentrale Rolle ein. Eine Bildverarbeitungslösung besteht einerseits aus dem Sensor Kamera, die ein Bild erzeugt, das aber für den Roboter erst mit umfangreicher Software ausgewertet und nutzbar gemacht werden muss. Der Anwendungsbereich ist extrem vielfältig und reicht von der Objektidentifikation (Personen, Geometrie eines Objektes, Lage eines Objektes) bis hin zur Material- und Fehlererkennung.

Ein typisches Beispiel für eine kompakte intelligente Kamera für den Einsatz in der Automatisierungstechnik ist die Festo-Kamera vom Typ SBO...-Q. Sie vereint das Sensorsystem zur Aufnahme der Bilddaten, die komplette Auswerteelektronik, eine integrierte SPS und Schnittstellen zur Kommunikation mit externen Steuerungssystemen wie z.B. einer Robotersteuerung:



Kamera mit integrierter Auswertelektronik

- | | |
|---|-----|
| • Kameragehäuse | [1] |
| • Objektiv | [2] |
| • Ring zur Blendenjustierung | [3] |
| • Entfernungseinstellung | [4] |
| • Ethernet Schnittstelle | [5] |
| • Spannungsversorgung und digitale E/As | [6] |
| • CAN Schnittstelle | [7] |

Folgende Tabelle zeigt eine Liste von Bildverarbeitungsmethoden, um gewisse Anwendungen zu realisieren:

Werkzeug	Beschreibung	Anwendungsbeispiele
ROI	Bereich, in dem alle Pixel als ein zusammenhängendes Objekt gesehen werden. Auf diesem kann man Merkmale wie Schwerpunkt-kordinaten, Abmessungen, Umfang oder Fläche berechnen.	Anwesenheit von Objekteigenschaften Vollständigkeitskontrolle Umfang und Fläche eines Objekts Positions- und Drehlageerkennung eines Objekts
Kreis- und Kantenerkennung	Werkzeuge zur Bestimmung von Ausgleichskreisen(-geraden) von Objektkanten und der zugehörigen Qualitätsmerkmale.	Positionsbestimmung von Objekten Prüfung von Kanten auf Einbrüche
Blobfinder	Das Werkzeug sucht zusammenhängende Pixel, die in dem vorher eingestellten Helligkeits- bzw. Farbbereich liegen. Diese Pixelwolken ergeben dann einzelne Objekte. Die Objekte können gezählt werden und für diese Objekte lassen sich individuelle Merkmale berechnen.	Berechnung der Position und Drehlage mehrerer Objekte Zählen von Objekten Umfangs- und Größenberechnung von Objekten Finden von relevanten Teilen im Suchfenster
Pattern Matching	Werkzeug zum Suchen vorher eingelernter Muster. Bis zu 4 Muster pro Werkzeug lassen sich einlernen. Diese werden dann unabhängig von deren Drehlage und Position im Werkzeugbereich gesucht. Auch wenn sie sich zum Teil berühren oder teilweise verdeckt sind, findet die Kamera diese.	Sortierung von Objekten Übereinstimmung von Objekten mit dem Muster

■ Kontrollfrage

Sie wollen kostengünstig und berührungslos mit dem Roboter überprüfen, ob das Material eines Objektes schwarz ist. Alle Objekte haben die gleiche geometrische Struktur und jeweils eine homogene Farbstruktur, die entweder schwarz oder eindeutig nicht schwarz ist.

– Welchen Sensor setzen Sie ein?

Begründen Sie Ihre Antwort.

- Als Sensor wird ein Reflex-Lichttaster verwendet.
Begründung: Bringe den Sensor mit dem Roboter in einen festen Abstand zum Objekt. Falls das Testobjekt schwarz ist, dann justieren Sie gegebenenfalls den Sensor so, dass keine Reflexion den Sensor aktiviert und dass bei einem andersfarbigen Objekt der Sensor aktiviert wird, d.h. eine Reflexion gemessen wird.

5 Programmierung von Industrieroboter

Es gibt zwei verschiedene Verfahren zur Programmierung:

5.1 Online-Verfahren

Die Programmierung erfolgt über das Programmierhandgerät **PGH** (Teach-Box) bei eingeschalteter Steuerung des Roboters. Der Roboter wird mit dem PGH zu den gewünschten Raumpunkten verfahren, die dann gespeichert werden. Notwendige Zusatzinformation wie Geschwindigkeit, Beschleunigung, Greifer-funktionen u.a. werden über die Tastatur eingegeben.

Jeder Roboterhersteller liefert ein eigenes speziell auf seine Robotersteuerung zugeschnittenes PGH, so liefert z.B. Mitsubishi zwei Handbediengeräte – ein kompaktes und ein komfortables Handbedienterminal mit Farbdisplay und Touch-Screen:



Handbediengeräte

Obwohl sich die Handbediengeräte der Hersteller im Detail unterscheiden, so liefern sie alle die folgenden Grundfunktionen:

- EIN/AUS Schalter
- NOT-AUS Schalter
- Sicherheitsschalter zur Freigabe der Antriebe
- Schalter zum Verfahren der Achsen in verschiedenen Geschwindigkeiten
- Bewegung des Roboters in verschiedenen Koordinatensystemen
- Positionen innerhalb eines Programms speichern
- Programmanzeige mit Editor
- Steuerungsparameter anzeigen und ändern
- Monitor zur Anzeige von E/A-Signalen und Antriebsdaten

Der Nachteil der online-Programmierung ist es, dass die Roboteranwendung verfügbar sein muss und nicht für die Produktion zur Verfügung steht. Daher wird in der Regel das Roboterprogramm zunächst offline am PC erstellt und dann erfolgt bei der realen Inbetriebnahme die Optimierung und Anpassung des Programms mit dem PGH.

■ Kontrollfrage

- Warum ist es im Allgemeinen notwendig mit dem PGH feste Positionen für den Roboter einzulernen (teachen)?

Falls ein Roboter eine Position mit hoher Genauigkeit anfahren muss, dann ist es aufgrund der unbekannten absoluten Genauigkeit erforderlich, dass der Roboter diese Position kennt. Hierzu wird das PGH eingesetzt.

5.2 Offline-Verfahren

Zur Programmierung bieten die Roboterhersteller entsprechende RC-Programmiersysteme an, die auf üblichen Hochsprachen wie BASIC, PASCAL, C oder C++ basieren, aber zahlreiche zusätzliche Funktionen zur

- Roboterbewegung,
- zur Roboterpositionierung und Roboterorientierung,
- zur Verarbeitung von E/A-Signalen
- und zur Kommunikation mit externen Steuerungssystemen benötigen.

Es gab intensive Bemühungen zur Standardisierung von Roboter-Programmiersystemen wie z.B. IRL (Industrial Robot Language), aber alle Versuche sind gescheitert. Beispiele für verschiedene RC-Programmiersprachen sind:

- KRL für Kuka Roboter
- RAPID für ABB Roboter
- MELFA BASIC für Mitsubishi Roboter

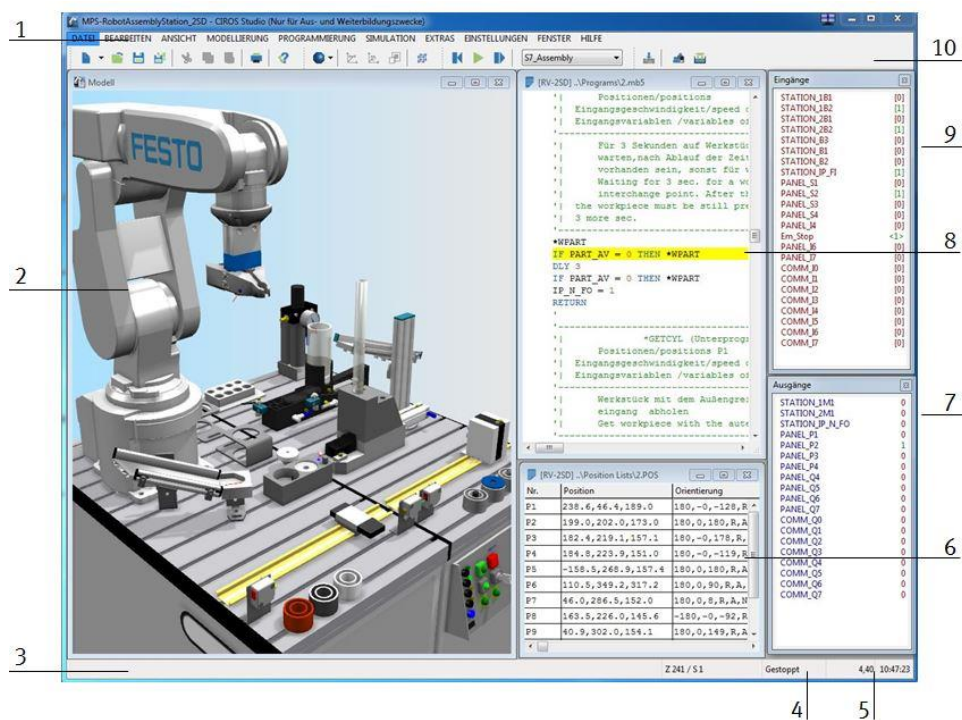
Effektiv kann die Roboterprogrammierung am PC nur dann erfolgen, wenn auch eine adäquate 3D-Simulation der Roboteranwendung zur Verfügung steht, um Bewegungsabläufe zu visualisieren und mögliche Kollisionen schon im Vorfeld zu vermeiden. In diesem Arbeitsbuch wird das recht universelle und sehr leistungsfähige Simulationssystem **CIROS Studio** verwendet, mit dem nicht nur eine Offline-Programmierung, sondern eine virtuelle Inbetriebnahme von Automatisierungssystemen ermöglicht wird.

5.3 CIROS Studio

Mit CIROS Studio können Roboterzellen und automatisierte Fertigungsanlagen modelliert und simuliert werden. CIROS Studio spannt einen Bogen über die Bereiche Anlagenplanung, Konstruktion, Elektroplanung, Steuerungsprogrammierung, Inbetriebnahme, Bedienertraining, Vertrieb und Marketing.

An dieser Stelle sollen einige grundlegende Bedienelemente zur Handhabung eines virtuellen Modells zusammengestellt werden. Eine vollständige Dokumentation und auch Tutorials zur Arbeit mit CIROS findet man in der umfangreichen Online-Hilfe.

- Bedienoberfläche

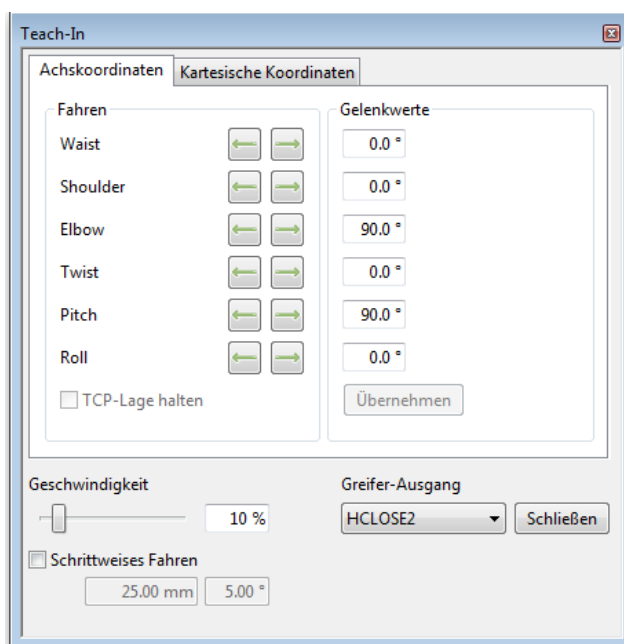


Bedienoberfläche CIROS

1: Menüleiste, 2: Arbeitszellenfenster, 3: Statusleiste, 4: Start/Stop Simulation, 5: Simulationszeit, 6: Positionsliste, 7: Roboter Ausgänge, 8: Roboterprogramm, 9: Roboter Eingänge, 10: Werkzeugleiste

- Ansicht des Modells im Arbeitszellen-Fenster verändern
 - Verwenden Sie die folgenden Tastenkombinationen, um die Ansicht im Arbeitszellenfenster zu ändern:
 Verschieben: **Umschalt + linke Maustaste**
 Drehen: **Strg + linke/rechte Maustaste**
 Zoomen: **Mausrad**
 - Über das Kontextmenü im Arbeitszellen-Fenster (rechte Maustaste) können Sie verschiedene Standardansichten über Buchstaben auswählen, z. B. **A** steht für Aufsicht und **O** für eine festgelegte Ausgangsansicht.

- Programm starten und stoppen
 - Start/ Stopp eines vorhandenen Programms: **F5**
 - Modell in Ursprung zurücksetzen (RESET): **Strg + F5**
- Einsatz des virtuellen Handbediengeräts – **Teach-In** Fenster (Taste **F8**):
 Ansicht **Achskoordinaten**:
 - Durch Mausklick auf die Pfeiltasten kann die jeweilige Achse in positive oder negative Richtung bewegt werden
 - Mit der Option Schrittweise Fahren wird bei jedem Mausklick nur ein Bewegungsschritt von der gewählten Größe durchgeführt.
 - Der Greifer kann über die Taste Schließen/Öffnen geschlossen oder geöffnet werden.
 - Im rechten Teil des Fensters werden die aktuellen Gelenkwerte dargestellt. Hier können Sie auch Werte direkt eintragen, die dann mit der Taste **Übernehmen** auf den Roboter übertragen werden.



Teach-in-Fenster- Achskoordination

Das Bezugssystem zur Roboterbewegung ist hier das Roboter Basiskoordinatensystem.

Mit den linearen Pfeiltasten neben der Achse wird der TCP des Roboters in positive/negative Richtung der entsprechenden Achse des Roboter Basiskoordinatensystems bewegt.

Mit den gebogenen Pfeiltasten wird der Roboter so bewegt, dass sich das TCP-Koordinatensystem um die x (y, z) -Achse des Basiskoordinatensystems dreht und die Lage des TCP unverändert bleibt.

Im rechten Teil des Fensters wird jeweils die aktuelle Lage und Orientierung des TCP relativ zur Roboterbasis angezeigt. Diese Werte können Sie auch manuell ändern



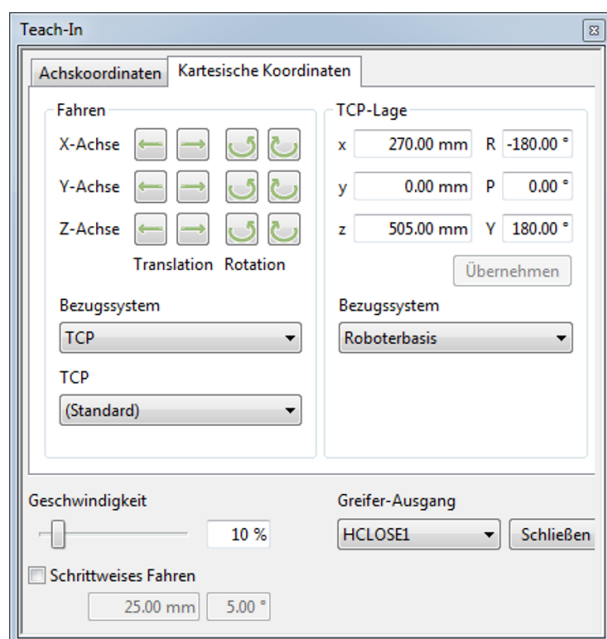
Teach-In Fenster- Roboterkoordinaten

Das Bezugssystem zur Roboterbewegung ist hier das Werkzeug (TCP)-Koordinatensystem.

Mit den linearen Pfeiltasten neben der Achse wird der TCP des Roboters in positive/negative Richtung der entsprechenden Achse des TCP-Koordinatensystems bewegt.

Mit den gebogenen Pfeiltasten wird der Roboter so bewegt, dass sich das TCP-Koordinatensystem um die x (y, z)-Achse des TCP-Koordinatensystems dreht und die Lage des TCP unverändert bleibt.

Als Bezugssystem kann auch ein beliebiges Objekt-Koordinatensystem gewählt werden. Dies wird in CIROS als **3D-Marker** bezeichnet.



Teach-In – TCP Koordinatensystem

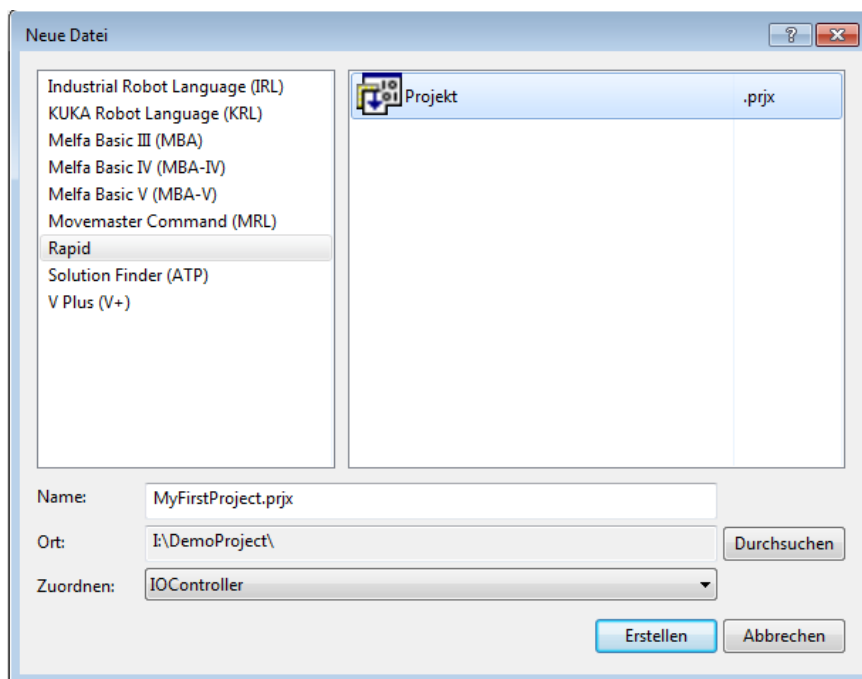
- Projektverwaltung und Programmerstellung

Zur Programmerstellung in CIROS muss zunächst ein Projekt angelegt werden. Der Anwender legt fest, in welcher Robotersprache und zu welcher Robotersteuerung das Programm gehören soll. Hierzu sind folgende Schritte notwendig:

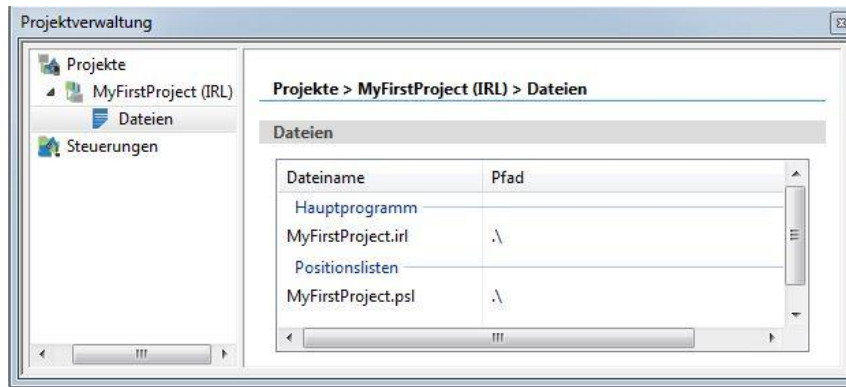
Öffnen Sie das Menü **PROGRAMMIERUNG -> Projektverwaltung**. Sie erhalten das Fenster zur Projektverwaltung.



Öffnen Sie das Kontextmenü zu **Projekte** und wählen Sie **Neu**. Sie werden aufgefordert einen Namen für das Projekt einzugeben und die Programmiersprache auszuwählen, z.B. **IRL**.

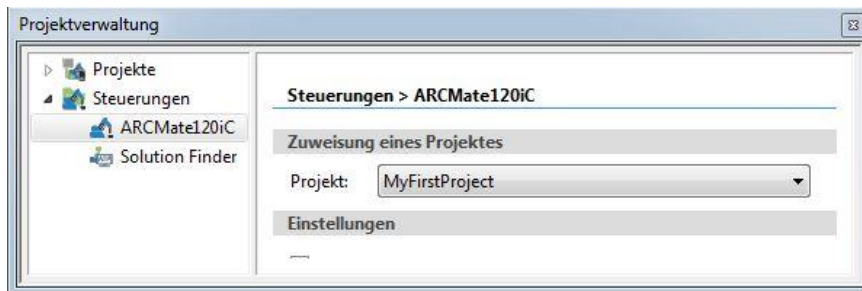


Es erscheint jetzt das Projektfenster mit dem Namen und fordert Sie auf, die Dateien – also Programme und Positionslisten – zum Projekt anzugeben:



Wählen Sie das Programm aus, dann können Sie über das Kontextmenü das Fenster mit dem entsprechenden Editor zur Programmeingabe öffnen. Entsprechend können Sie die Positionsliste öffnen, um Positionen einzutragen.

- Virtuelle Robotersteuerung

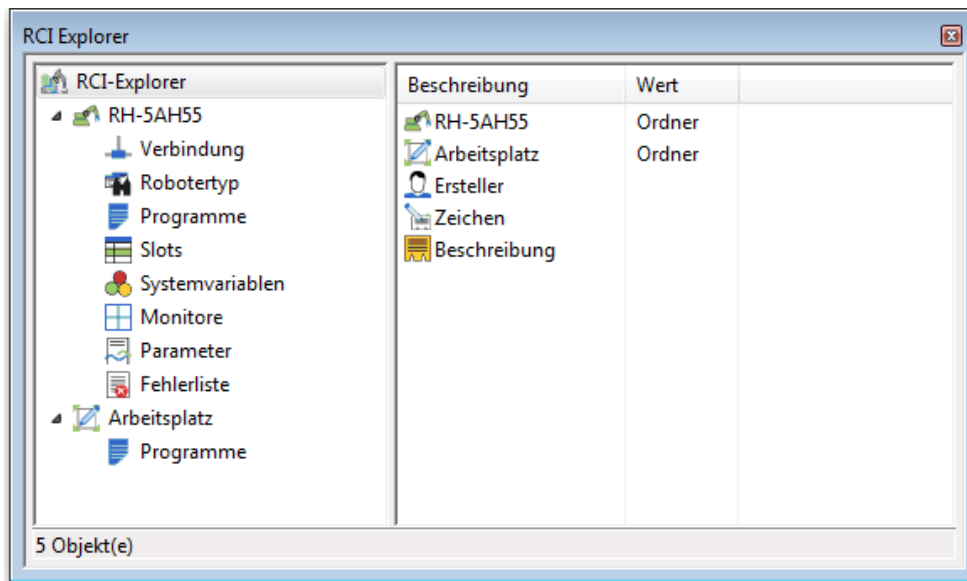


Zum Abschluss müssen Sie Ihrem Projekt die gewünschte Steuerung zuordnen. In dem vorliegenden Beispiel gibt es nur eine Fanuc Steuerung vom Typ ARCMate120iC.

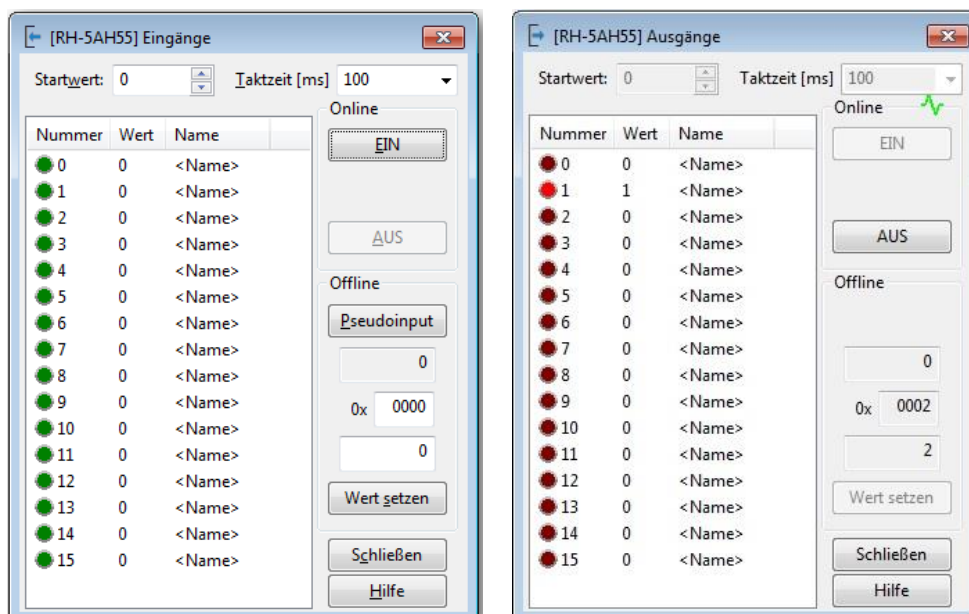
CIROS enthält eine universelle virtuelle Robotersteuerung, die nahezu jede Kinematik steuern kann. Das Verhalten dieser Steuerung ist natürlich nicht identisch zur entsprechenden realen Steuerung, aber bildet im Allgemeinen das Verhalten sehr gut nach.

Zur Simulation muss das erstellte Programm in den IRDATA-Maschinencode der virtuellen Steuerung kompiliert werden (**PROGRAMMIERUNG -> Kompilieren**). Ist der Kompilierungsvorgang erfolgreich, dann kann das Programm gestartet werden: (**F5**) oder (**SIMULATION -> Start/Stop**).

- Online Kopplung zur Mitsubishi Robotersteuerung



Via einer Ethernet Verbindung können Sie mittels des **RCI – Explorers** eine online Verbindung zu jedem Mitsubishi Roboter herstellen. Im Rahmen einer Online-Kopplung haben Sie die Möglichkeit Programme und Positionslisten in die Steuerung zu laden, ein online Debugging ihres Programms durchzuführen, und Sie haben Zugriff auf alle internen Parameter, Variablen, Motordaten und Ein- und Ausgänge:

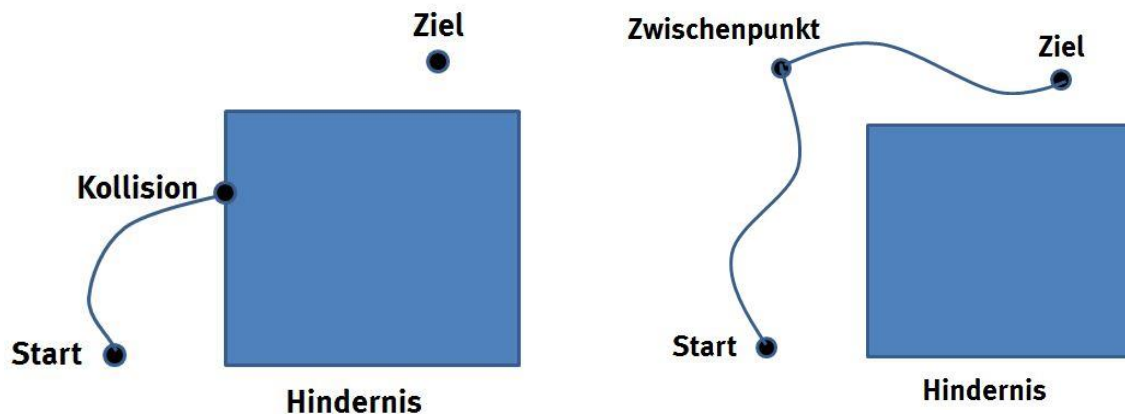


5.4 Hinweise zur Roboterprogrammierung

Eine umfangreiche Darstellung zur Roboterprogrammierung finden Sie in den Dokumentationen zur entsprechenden Programmiersprache. An dieser Stelle soll nur auf zwei typische Vorgänge hingewiesen werden, die zu Beginn auftreten.

5.4.1 Hindernis vermeiden

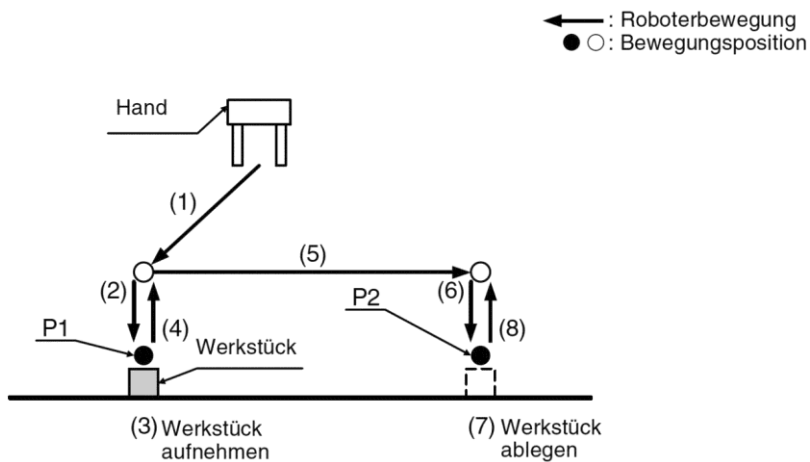
Es sind Start- und Zielposition mittels des PGH für den Roboter eingelesen worden. Dann ist im Programm darauf zu achten, dass es keine Hindernisse in der Bewegungsbahn vom Roboter gibt:



Durch das Einlernen zusätzlicher Zwischenpunkte können Hindernisse umfahren und somit Kollisionen mit Hindernissen vermieden werden.

5.4.2 Beispiel: Werkstück greifen

Roboter soll ein Werkstück greifen:



Programmschritt	Beschreibung
Start	Greifer öffnen
Schritt 1	Greifer senkrecht oberhalb der Zielposition (ca. 50 mm) bewegen
Schritt 2	Zielposition linear anfahren (Geschwindigkeit reduzieren), um Kollisionen mit dem Werkstück zu vermeiden.
Schritt 3	In Zielposition kurze Verzögerungszeit (ca. 0.5 sek) abwarten, um Regelungsschwingungen zu vermeiden. Werkstück greifen und mit einer kurzen Verzögerungszeit das Schließen des Greifers

	abwarten.
Schritt 4	Linear zurückfahren (ca. 50 mm), um Kollisionen mit dem Umgebungsbereich des Werkstücks zu vermeiden.
Schritt 5	Greifer mit PTP Bewegung senkrecht oberhalb der Ablageposition P2 (ca. 50 mm) verfahren
Schritt 6	Lineares Anfahren der Ablageposition P2 (Geschwindigkeit reduzieren)
Schritt 7	In Zielposition P2 kurze Verzögerungszeit (ca. 0.5 sek) abwarten, um Regelungsschwingungen zu vermeiden. Greifer öffnen und mit einer kurzen Verzögerungszeit das Öffnen des Greifers abwarten.
Schritt 8	Linear zurückfahren (ca. 50 mm), um Kollisionen mit dem Umgebungsbereich des Werkstücks zu vermeiden.

Kontrollfrage

- Erstellen Sie ein Roboter-Programm, z.B. in Melfa Basic V, das die obige Ablaufsequenz realisiert.

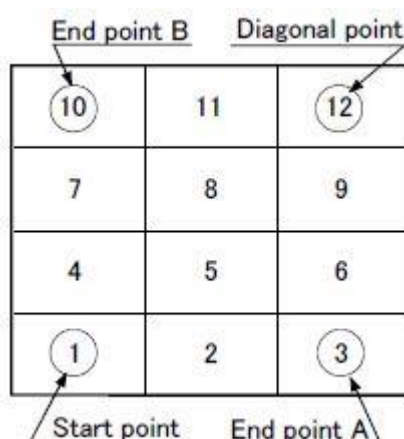
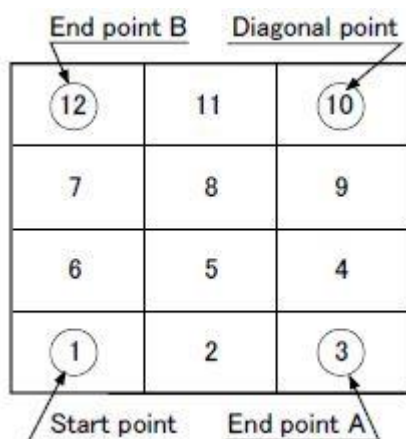
Melfa Basic V Programm:

```

REM Start
HOPEN 1
REM Schritt 1
MOV P1,-50
REM Schritt 2
SPD 80
MVS P1
REM Schritt 3
DLY 0.5
HCLOSE 1
DLY 0.5
REM Schritt 4
MVS P1,-50
REM Schritt 5
MOV P2,-50
REM Schritt 6
MVS P2
REM Schritt 7
DLY 0.5
HOPEN 1
DLY 0.5
REM Schritt 8
MVS P2
  
```

5.4.3 Beispiel Paletten-Funktion

Roboter soll Werkstücke aus einer Palette entnehmen oder dort ablegen.



Es wird vorausgesetzt, dass die Abstände der Werkstückpositionen in den Zeilen und Spalten der Palette gleichmäßig sind. Dann ist es nicht notwendig, jede einzelne Position der Palette einzulernen, sondern es reicht aus, die Eckpunkte der Palette einzulernen. Wichtig ist noch die Nummerierung der Elemente der Palette festzulegen, damit der Roboter auch das gewünschte Element finden kann. Es gibt zwei typische Verfahren zur Nummerierung:

- c) Die Zeilen werden jeweils von links nach rechts fortlaufend nummeriert, siehe rechte Grafik.
- d) Die Zeilen werden abwechselnd von links nach rechts und dann von rechts nach links nummeriert, siehe linke Grafik.

Diese Festlegungen müssen Sie in der Definition der Palette zu Beginn dem Programm mitteilen.

Zum Beispiel würde in Melfa Basic V die Definition der Palette im Fall 1 wie folgt aussehen:

```
DEF PLT 1,P1,P2,P3,P4,4,3,1
```

wobei folgendes gilt:

P1 = Start point,

P2 = End point A,

P3 = End point B,

P4 = Diagonal point,

4 = Anzahl der Zeilen,

3 = Anzahl der Spalten,

1 = Nummerierungsverfahren (= 2 im zweiten Fall)

Ein PTP Bewegungsbefehl zur Positionsnummer 6 im ersten Fall würde dann z.B. in Melfa Basic V wie folgt aussehen: **MOV (PLT 1,6)**

■ Kontrollfrage

- Sie haben eine Palette mit 5 Zeilen und 3 Spalten. Erstellen Sie ein Roboter-Programm, das den Roboter an die letzte Position in der dritten Zeile positioniert. Welche Positionsnummern müssen Sie einlernen?

```
DEF PLT 1 P1,P2,P3,P4,5,3,1
```

```
MOV (PLT 1,9)
```

P1 hat die Positionsnummer 1

P2 hat die Positionsnummer 3

P3 hat die Positionsnummer 13

P4 hat die Positionsnummer 15

6 Sicherheitsmaßnahmen

Die Richtlinien zu den Sicherheitsanforderungen beim Einsatz von Industrierobotern ist in der Norm DIN EN ISO 10218 festgelegt. Industrieroboter sind nach heutigem Stand der Technik so gebaut, dass man sie als „betriebssicher“ bezeichnen kann. Dennoch ist trotz Einhaltung aller Sicherheitsvorschriften der Betrieb von Industrierobotern mit einem Restrisiko verbunden. Um dieses Restrisiko so gering wie möglich zu halten, müssen Schutzmaßnahmen und Vorschriften zwingend eingehalten werden.

6.1 Allgemeine Sicherheitshinweise

6.1.1 Betriebsanleitung

Hersteller von Industrieroboter geben in entsprechenden Betriebsanleitungen Sicherheitsrichtlinien für den Umgang mit dem entsprechenden Industrieroboter und weisen ausdrücklich darauf hin, dass vor der Inbetriebnahme alle Anweisungen zum Thema Sicherheit sorgfältig zu lesen sind. Dies gilt entsprechend für die Sicherheitsanweisungen in der technischen Dokumentation einer Gesamtanlage, in der ein oder mehrere Roboter integriert sind.

6.1.2 Verpflichtung des Betreibers

Bei unsachgemäßen Einsatz oder Fehlbedienungen des Robotersystems drohen Gefahren für

- den Bediener
- und für das Robotersystem.

Daher ist der Betreiber verpflichtet. Personen, die im Rahmen der Installation, der Inbetriebnahme, der Programmierung, der Bedienung, der Wartung und der Reparatur tätig sind,

- mit den grundlegenden Vorschriften über Arbeitssicherheit vertraut zu machen,
- in die Handhabung des Robotersystems einzuweisen und
- in regelmäßigen Abständen ihr sicherheitsbewusstes Arbeiten zu überprüfen.

6.1.3 Verpflichtung des Bedienpersonals

Alle Personen, die mit irgendeiner Arbeit am Robotersystem beauftragt sind, müssen

- Die Vorschriften über Arbeitssicherheit beachten,
- und die Hinweise zur Sicherheit des Roboterherstellers und des Maschinenlieferanten lesen und befolgen.

6.2 Schutzeinrichtungen

Es lassen sich nicht alle Gefahrenquellen in einem Maschinensystem mit Industrierobotern beseitigen, so dass es notwendig ist die verbleibenden Risiken durch Schutzeinrichtungen zu sichern:

- **Trennende Schutzeinrichtungen** sind körperliche Sperren wie Verkleidungen, Verdeckungen oder Umzäunungen.
- **Schutzeinrichtungen mit Annäherungsreaktion** sind Schaltmatten, Lichtschranken, Lichtvorhänge und Sensoreinrichtungen, die das Annähern von Personen erkennen.

6.2.1 Verriegelte trennende Schutzeinrichtung

Trennende Schutzeinrichtungen sind solche, die aus der Schutzstellung entfernbar sind. Je nach Häufigkeit des Zugangs zur Gefahrenquelle können diese Schutzeinrichtungen klappbar, schwenkbar oder verschiebbar gestaltet sein. Diese Schutzeinrichtungen müssen so verriegelt sein, dass ein Erreichen der Gefahrenstelle während der gefahrbringenden Bewegung verhindert wird. In zahlreichen Fällen wird dies dadurch erreicht, dass die Schutzeinrichtung eine Tür hat, die bei Öffnung einen NOT-AUS der Anlage bzw. einen NOT-AUS des Robotersystems bewirkt.

7 Inbetriebnahme einer Roboteranwendung

7.1 Einleitung

In diesem Abschnitt gehen wir nicht auf die allgemeine Richtlinie zur Inbetriebnahme eines Automatisierungssystems ein, sondern fokussieren uns auf die Inbetriebnahme einer Roboteranwendung.

7.2 Virtuelle Inbetriebnahme

Aus Sicherheits- und Planungsgründen wird heute in der Industrie zunehmend von der Roboteranwendung zunächst ein 3D-Simulationsmodell erstellt. Hierzu wird das entsprechende CAD-Modell aus dem Konstruktionsbereich eingesetzt und in eine Simulationsumgebung wie z.B. CIROS Studio importiert. In der Simulationssoftware wird das CAD-Modell optimiert, das mechanische und dynamische Verhalten der Roboter und anderer mechanischer Komponenten, die Physik der Sensorik und die elektrische Verdrahtung der Sensorik mit der Steuerung modelliert. Ebenso enthält ein leistungsfähiges Simulationssystem integrierte virtuelle SPS- und Robotersteuerungen, die möglichst viele native Roboterprogrammiersprachen interpretieren können. Es wird dann für das virtuelle Modell der realen Roboteranwendung in der Simulationsumgebung das Programm entwickelt und insbesondere im Hinblick auf Zykluszeiten und Kollisionen getestet.

7.3 Reale Inbetriebnahme der Roboteranwendung

Der erste Schritt zur realen Inbetriebnahme ist der Download der Roboteranwendung in das reale Steuerungssystem. Hierbei sind eventuell schon erste Anpassungen notwendig, da der virtuelle Compiler nicht identisch mit dem Compiler der realen Steuerung ist.

Ist der Download schließlich erfolgreich durchgeführt, dann ist im zweiten Schritt zu prüfen, ob die E/A-Verdrahtung mit der Realität übereinstimmt oder ob weitere Änderungen notwendig sind.

Der finale Schritt besteht in der Überprüfung der eingelernten Positionen. Aufgrund von Fertigungs- und insbesondere Montagetoleranzen stimmt das CAD-Modell mit der Realität nicht zu 100% überein. Daher ist dieser im Allgemeinen recht aufwendige Prüfungs- und Änderungsprozess unbedingt erforderlich.

Nach diesen Vorbereitungen kann dann eine Test-Inbetriebnahme an der realen Hardware durchgeführt und schrittweise optimiert werden.

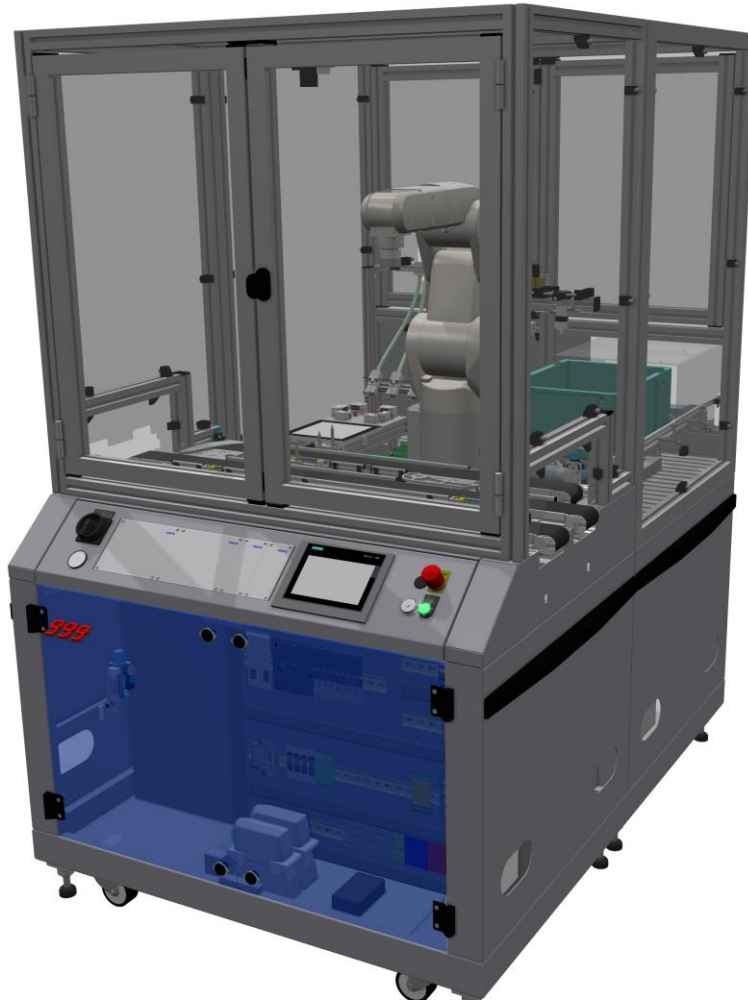
Es ist unbedingt erforderlich, dass alle Änderungen sorgfältig dokumentiert werden müssen.

Praktische Aufgaben

Einleitung	45
Aufgabe 1: Aufbau der CP Factory Station Robotertermontage	57
Aufgabe 2: Funktionsprüfung des Mitsubishi Roboters RV-4FL in der Montagestation	69
Aufgabe 3: Werkzeugwechselsystem für einfache Pick&Place Aufgabe nutzen	79
Aufgabe 4: Platine auf Werkstück montieren und Prüfung des Werkzeugwechselsystems	97
Aufgabe 5: Optische Positionsprüfung eines Werkstücks	113
Aufgabe 6: Anwendung der externen Funktionsbibliothek	125
Aufgabe 7: Montage einer Platine mit Sicherungen unter Anwendung der externen Funktionsbibliothek	131

Einleitung

Sie arbeiten in einem mittelständischen Unternehmen, das Platinen bestückt. Zur Optimierung der Produktionsanlage soll eine neue Robotermontagestation integriert werden. Die Station ist komplett aufgebaut und verdrahtet.



Robotermontagestation

Ihre Aufgabe ist es die Roboteranwendung für diese Station zu entwickeln, zu programmieren und in Betrieb zu nehmen. Der gesamte Prozess soll von Ihnen dokumentiert werden.

Hinweis zur Bearbeitung der Aufgaben:

Die ersten beiden Aufgaben dienen der Einführung in die Arbeit mit dem Mitsubishi Roboter. Die folgenden Aufgaben 3 – 7 sind unabhängig, aber setzen die Grundkenntnisse zur Arbeit mit dem Mitsubishi Roboter voraus.

Es wird empfohlen, die Aufgaben 3 – 7 in dieser Reihenfolge zu bearbeiten, da sie aufeinander aufbauen.

Lesen Sie zuvor sorgfältig die folgenden Informationen.

1. Schnittstelle MES/SPS und Robotersteuerung:

Das Bindeglied zur Kommunikation zwischen MES und der Robotersteuerung bildet die SPS.

Die Robotersteuerung ist mit der SPS über Profinet verbunden. Die Daten sind in einem speziellen Adressbereich der Robotersteuerung abgelegt. Zusätzlich liefert das HMI des Bedienfelds eine interaktive Möglichkeit, Kommunikationsdaten manuell zu generieren. Dieses soll hier zur Vorbereitung der weiteren Arbeit mit dem Roboter im Detail erläutert werden:

1.1. Eingänge und Ausgänge der Robotersteuerung:

	Roboter RV-4FL				
Sensorname	Variablenname	Ein-/Ausgang	Roboter-adresse	Datentyp	Kommentar
BG9	DI_WPClamped	Eingang	16	Bit	Werkstück ist eingespannt
BG10	DI_WPAvail	Eingang	17	Bit	Werkstück verfügbar in Montage
BG11	DI_WPOrientOk	Eingang	18	Bit	Werkstücklage in Montageplatz korrekt.
BG12	DI_WPHoleOk	Eingang	19	Bit	Bohrung im Werkstück im Montageplatz gefunden
BG14	DI_FuseMag1Avail	Eingang	21	Bit	Sicherung in Magazin 1 vorhanden
BG15	DI_FuseMag2Avail	Eingang	22	Bit	Sicherung in Magazin 2 vorhanden
BG16	DI_FuseMag3Avail	Eingang	23	Bit	Sicherung in Magazin 3 vorhanden
BG18	DI_Grp1VacStore	Eingang	25	Bit	Vakuumgreifer in Wechselvorrichtung Nr. 1 vorhanden
BG19	DI_Grp2WrkStore	Eingang	26	Bit	Werkzeuggreifer in Wechselvorrichtung Nr. 2 vorhanden
BG20	DI_Grp3FuseStore	Eingang	27	Bit	Greifer für Sicherungen in Wechselvorrichtung Nr. 3 vorhanden
MB9	DO_WPMountLock	Ausgang	16	Bit	Werkstück einspannen
K_IN0	DO_StartCamera	Ausgang	24	Bit	Kamera wird gestartet

1.2 Sensorinformationen an die Robotersteuerung via Profinet:

	Sensorinformationen via Profinet				
Sensorname	Variablenname	Ein-/Ausgang	Roboter-adresse	Datentyp	Kommentar
BG1	DI_PalletInFront	Eingang	2144	Bit	Platinenbox ist vorne
BG2	DI_PalInLoadPos	Eingang	2145	Bit	Platinenbox ist in Beladeposition
BG3	DI_PalOrientOk	Eingang	2146	Bit	Orientierung der Platinenbox ist ok
BG5	DI_PalIndex1Up	Eingang	2147	Bit	Platinenbox ist durch vorderen Stopper angehalten
BG6	DI_PalIndex2Up	Eingang	2148	Bit	Platinenbox ist durch hinteren Stopper angehalten

SF7	DI_PalChangeAck	Eingang	2149	Bit	Platinenbox ändert Kennung
BG50	DI_WPPalAvail	Eingang	2150	Bit	Transportträger an Stopper-Position
BG51	DI_WPAStopAvail	Eingang	2151	Bit	Werkstück auf Transportträger an Stopper-Position
MA4	DO_ExtIndexBolt	Output	2144	Bit	Stopperzylinder für Platinenbox ausfahren.
MA4	DO_BeltOnInDir	Output	2145	Bit	Platinenbox in den Stationsbereich fahren
SF7	DO_BeltOnOutDir	Output	2146	Bit	Platinenbox aus der Station fahren
	DO_PalAckLampOn	Output	2147	Bit	Warnlampe der Platinenbox einschalten

1.3 Roboter Informationsvariablen via Profinet:

	Informationsdaten via Profinet				
	Variablenname	Ein-/Ausgang	Roboteradresse	Datentyp	Kommentar
	DI_StopperNo	Eingang	2064 - 2071	Byte	Stoppernummer für Werkstückaufnahme
	DI_PCBPalletNo	Eingang	2072 - 2079	Byte	Platinennummer in Transportbox
	IODATA	Eingang	2016 - 2031	Word	Numerischer Eingabewert
	DI_RetryLastStep	Eingang	2032	Bit	Den letzten fehlerhaften Schritt wiederholen
	DI_ExitCurrCycle	Eingang	2033	Bit	Programmschritt verlassen
	IsRobotinoUsed	Eingang	2034	Bit	Robotino ist zum Transport Platinen im Einsatz
	IsRbtArmHome	Ausgang	2032	Bit	Roboterarm ist im Startbereich
	IsRbtAbovePCB	Ausgang	2033	Bit	Roboterarm ist im Bereich der Platinenbox
	ReturnValue	Ausgang	2064 - 2071	Byte	Rückgabewert vom Roboter

Hinweis:

Die Roboteradressen für **Profinet-Daten** sind in der Simulation um den Wert **1900** in die negative Richtung verschoben, also der Sensor **BG1** hat z.B. in der Simulation die Bitadresse = 244.

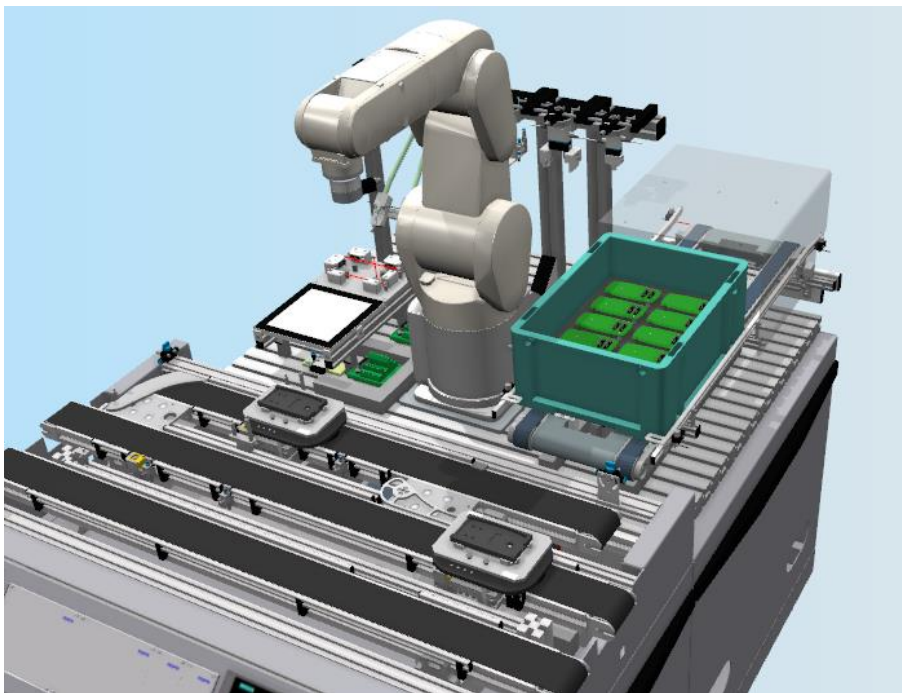
Rückmeldewerte des Roboters an die SPS bzw. MES-System:

Rückgabewert	Kommentar
1	Robotergreifer ist geschlossen
11	Unbekanntes Werkzeug gegriffen
21	Nicht definierte Stopper-Position
22	Unbekannter Programmname

23	Kein Werkstück auf dem Transortträger am Stopper
31	Kamera hat kein Werkstück gefunden
41	Montageplatz ist besetzt
42	Werkstück nicht in der Montage positioniert
43	Werkstück liegt umgedreht
44	Orientierung des Werkstücks ist nicht korrekt
45	Werkstück ist in der Montage nicht eingespannt
52	Unbekannte Platinennummer
53	Keine Platine an der Positionsnummer gefunden

1.4 Manuelle Einstellung einer Anwendungsumgebung für den Roboter in der Montagestation

Folgendes Szenarium wurde als Ausgangsstellung für die Aufgaben 3 -7 gewählt:

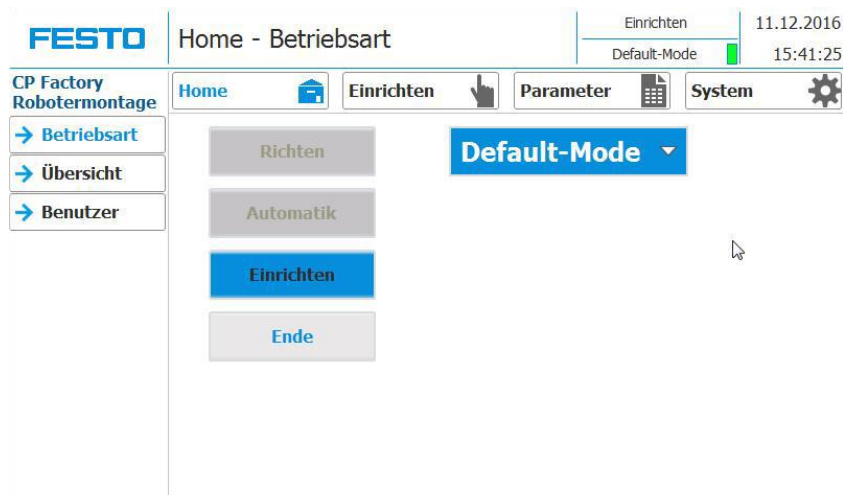


Roboter – Arbeitsumgebung

- Transportträger mit Werkstück ist in Stopper-Position 1 vom Bypass.
- Transportbox ist mit Platinen beladen.
- Transportbox ist in der Station und in der vorderen Position gestoppt.
- Kameraprogramm 1 ist geladen und funktionsfähig.
- Es sind Sicherungen in mindestens einem Magazin vorhanden.

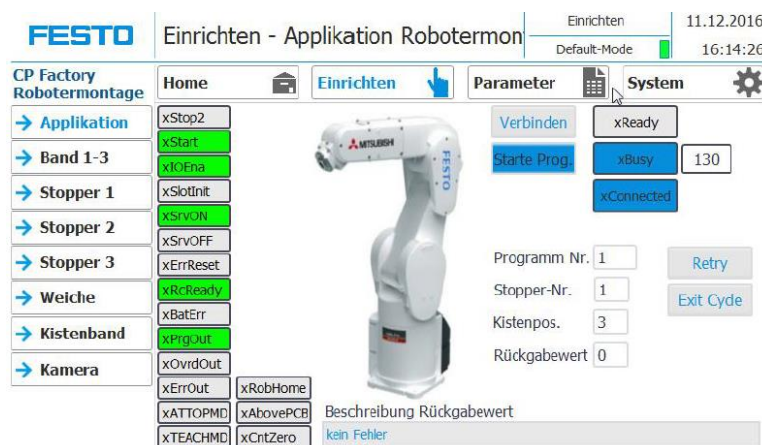
Diese Ausgangsstellung kann manuell wie folgt erreicht werden:

- Sie beladen einen Transportträger mit einem Werkzeug und schieben den Transportträger auf dem Bypass an die erste Stopper-Position.
- Nach dem Einschalten der Spannung mit dem Hauptschalter am Bedienfeld werden automatisch die SPS und die Robotersteuerung gestartet. Die Anzeige auf dem HMI-Touchpanel des Bedienfelds wird aktiviert. Es erscheint folgende Anzeige:



Falls eine Fehlermeldung erscheint (z.B. Roboter ist nicht in einer sicheren Startposition), dann müssen Sie an der Robotersteuerung den **MANUAL**-Modus einschalten, das PRG aktivieren und den Roboter in den Bereich der Startposition bewegen. Falls keine weiteren Fehlermeldungen auftreten, müssen Sie das PRG deaktivieren, den **AUTO**-Modus an der Steuerung wieder einschalten und am HMI den **Default-Mode** und den Modus **Einrichten** wählen. Warten Sie bis in der oberen Zeile der Modus **Einrichten** aktiviert wird. Das Feld wird dann blau markiert.

- Ist der Modus **Einrichten** aktiviert erscheint folgende Ansicht, wenn Sie den Modus **Anwendung** wählen:



Prüfen Sie, dass **xConnected** aktiv ist, denn dies zeigt Ihnen an, dass die Profinet-Verbindung zur SPS funktioniert.

Sie können die Parameter **Stopper-Nr.** und **Kistenpos.** verändern. Der Parameter liefert die Nummer für die Platinauswahl in der Transportbox. Wählen Sie jetzt die Option **Kistenband:**



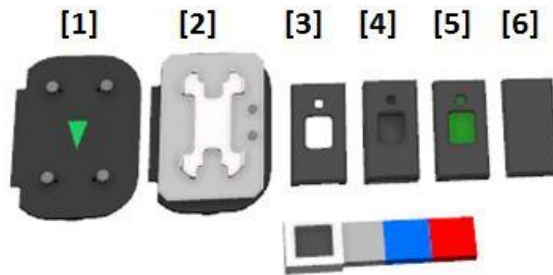
Aktivieren Sie **Box In** und dann **Extend** und warten Sie bis die Box vollständig ausgefahren ist und mittels der ausgefahrenen Stopper angehalten wird. Sind die Stopper ausgefahren, dann drücken Sie nochmals auf **Extend**. Beladen Sie die Box mit Platinen. Sie haben die gewünschte Arbeitsumgebung erstellt.

Hinweis:

- 1 Sie können ein Roboterprogramm über die SPS nur dann starten, wenn der Programmname eine Zahl ist.
- 2 Ein Roboterprogramm kann auch dann auf die Profinet-Daten der SPS zugreifen, wenn es nicht über die SPS gestartet worden ist.
- 3 Sie können die **Stopper-Nr.** (Eingangsbyte DI_StopperNo) und die **Kistenpos.** (Eingangsbyte DI_PCBPalletNo) auch als Daten für andere Anwendungen nutzen.

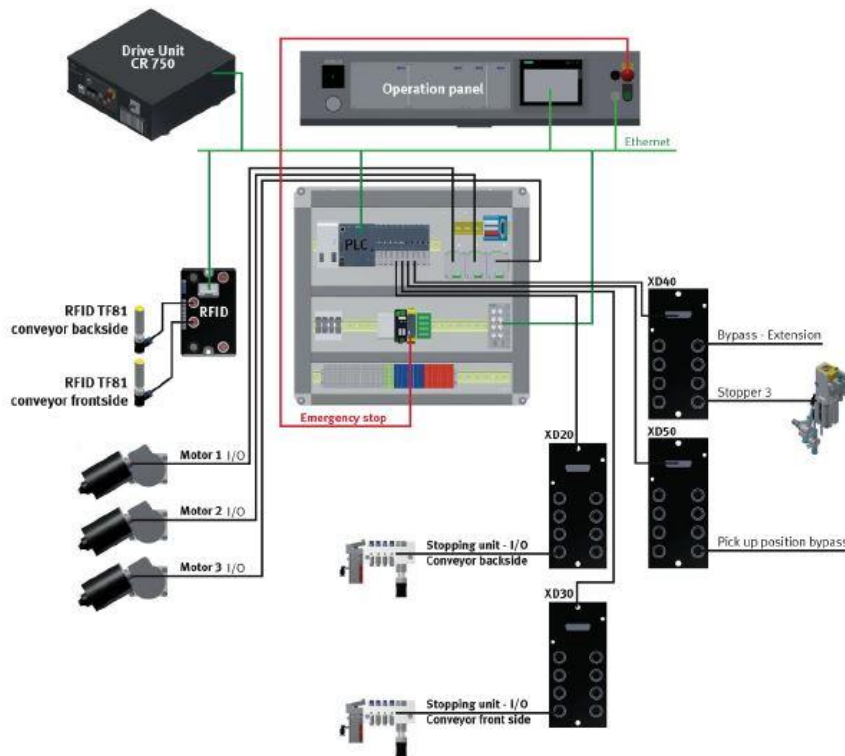
1.5 Manuelle Einstellung einer Anwendungsumgebung für den Roboter in der virtuellen Montagestation

Sie können die virtuelle Robotermontagezelle **CP_RobotAssembly.modx** im Verzeichnis **.../CP_Factory/Rob_Montage** öffnen. Starten Sie die Simulation, dann wird das aktuelle Programm in der virtuellen Steuerung gestartet. Parallel wird automatisch eine virtuelle SPS gestartet, die das Bandsystem startet, die Transportbox mit Platinen beladen lässt und sie in die vordere Ausgangsstellung fährt. Sie können manuell beliebig viele Palettenträger der Station wie folgt hinzufügen:



- Vor der Station befinden sich Taster, mit denen Sie verschiedene Komponenten auf das Bandsystem stellen können:
 [1] Leerer Palettenträger
 [2] Palettenträger mit Palette
 [3] Palettenträger mit Palette und Frontschale
 [4] Palettenträger mit Palette und Frontschale und Rückwandschale
 [5] Palettenträger mit Palette und vollständigem Produkt
 [6] Palettenträger mit Palette und Rohteil
- Sie können die jeweiligen Komponenten auch in den verschiedenen Farben schwarz, grau, blau und rot anliefern lassen. Hierzu verschieben Sie das weiße Rechteck auf eine der Farben. Die Standardeinstellung ist schwarz.
- Der ausgewählte Palettenträger fährt dann bis zum Stopper 1 im Bypass.

2. Verdrahtung Robotersteuerung mit SPS



Verdrahtungsschema der Roboterstation

Hinweis:

Wenn Sie sich mit Ihrem PC mit der Robotersteuerung verbinden möchten, dann stecken Sie das Ethernetkabel vom PC in die Anschlussbuchse am Bedienfeld (siehe Grafik).

Die korrekte IP-Adresse von der Robotersteuerung entnehmen Sie dem Parameter **NETIP** der Robotersteuerung.

3. Manuelle Ansteuerung der Greifer

Die Greifer und die Ankopplung eines Greifers werden über drei 2-Wege Ventile angesteuert, wobei jedes Ventil über zwei Ausgänge des Roboters gesteuert wird. Das sind die Ausgänge Hopen_1, HClose_1, Hopen_2, HClose_2 und Hopen_3, HClose_3.

3.1 Vakuumgreifer:

- PGH:
 - Hand-Modus einschalten
 - Vakuum einschalten: Tasten (-B) + (+C)
 - Vakuum ausschalten: Tasten (-B) + (-C) + kurz (+B) drücken (Teil abwerfen)
- Wählen Sie die Option **Greifer Ausgang** im **Teach-In** Fenster der Simulation:
 - Vakuum einschalten: HOPEN_1 <-> öffnen
 - HCLOSE_1 <-> schließen

- HOPEN_2 <-> schließen
 - HCLOSE_2 <-> öffnen
- Vakuum ausschalten:
 - HOPEN_1 <-> schließen
 - HCLOSE_1 <-> öffnen
 - HOPEN_2 <-> schließen
 - HCLOSE_2 <-> öffnen

3.2 Werkstückgreifer:

- PGH:
 - Hand-Modus einschalten
 - Greifer öffnen: Tasten (-B) + (+C)
 - Greifer schließen: Tasten (-C) + (+B)
- Wählen Sie die Option **Greifer Ausgang** im **Teach-In** Fenster der Simulation:
 - Greifer öffnen:
 - HOPEN_1 <-> öffnen
 - HCLOSE_1 <-> schließen
 - Greifer schließen:
 - HOPEN_1 <-> schließen
 - HCLOSE_1 <-> öffnen

3.3 Greifer für Sicherungen:

Bedienung ist analog zum Werkzeuggreifer

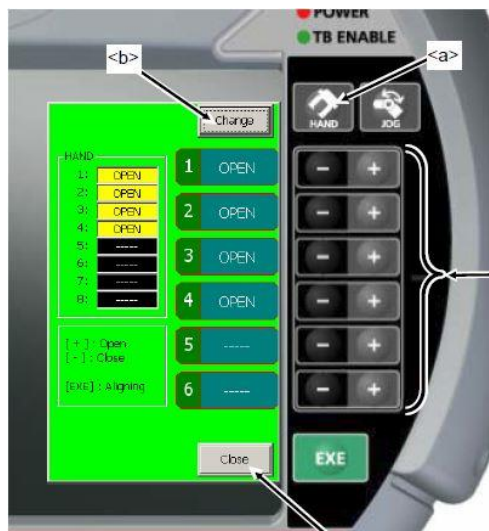
3.4 Greifer in Wechsellagerung zurücklagern:

- PGH:
 - Hand-Modus einschalten
 - Lösen: Taste (+X) drücken und dann kurz Tasten (-B) + (-C) drücken (Luft entweicht)
- Wählen Sie die Option **Greifer Ausgang** im **Teach-In** Fenster der Simulation:
 - Lösen:
 - HOPEN_3 <-> öffnen
 - HCLOSE_3 <-> schließen

3.5 Greifer aus Wechsellagerung entnehmen:

- PGH:
 - Hand-Modus einschalten
 - Verriegeln: Taste (-X) drücken
- Wählen Sie die Option **Greifer Ausgang** im **Teach-In** Fenster der Simulation:
 - Verriegeln:
 - HOPEN_3 <-> schließen
 - HCLOSE_3 <-> öffnen

3.6 Einsatz des R56TB als PGH



Anzeige des Hand-Modus beim R56TB

- Hand-Modus einschalten
- Mit **(+)** wird der entsprechende Hand-Zustand auf **OPEN** geschaltet und mit **(-)** auf **CLOSED**.
- Greifer aus Wechsellvorrichtung entnehmen:
 - Hand 1 und Hand 2 = CLOSED
 - **Hand 6 auf CLOSED** setzen
- Greifer in Wechsellvorrichtung zurücklagern:
 - Hand 1 und Hand 2 = CLOSED
 - **Hand 6 auf OPEN** setzen kurz HAND 1 auf OPEN setzen (Luft entweicht)
- Werkstückgreifer/ Greifer Sicherungen:
 - Greifer öffnen:
 - Hand 1 auf OPEN** und **Hand 2 auf CLOSED** setzen
 - Greifer schließen:
 - Hand 1 auf CLOSED** und **Hand 2 auf OPEN** setzen
- Vakuumsauger:
 - Ansaugen:
 - Hand 1 auf OPEN** und **Hand 2 auf CLOSED** setzen
 - Lösen:
 - Hand 1 auf CLOSED** und **Hand 2 auf CLOSED** setzen und zum „Abwerfen“ kurz **Hand2 auf OPEN** setzen.

4. TCP- Einstellung

Zur Durchführung der Anwendungen in dieser Montagestation benötigt der Roboter verschiedene TCP- Einstellungen:

TCP_Vac	=	(0,0,205,0,0,33.5)	TCP für den Roboter mit dem Vakuumgreifer
TCP_WP	=	(0,0,170,0,0,33.5)	TCP für den Roboter mit dem Werkstückgreifer
TCP_Fuse	=	(0,0,151.5,0,0,33.5)	TCP für den Roboter mit dem Greifer für die Sicherungen
TCP_None	=	(0,0,0,0,0,33.5)	TCP für das Greiferwechselsystem

Hinweis:

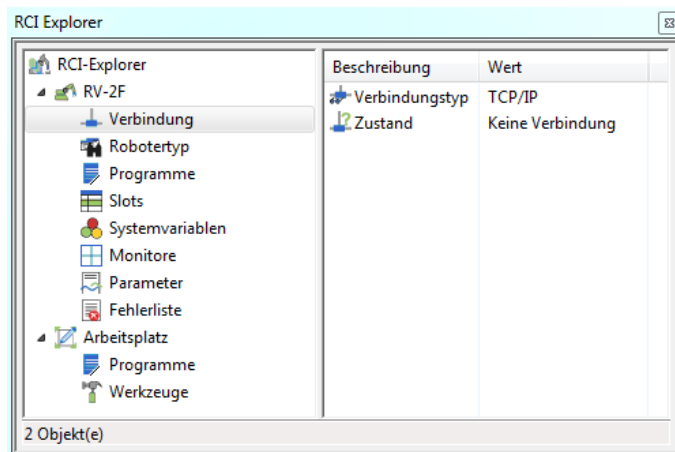
- In dem Simulationsmodell können Sie im Teach-In Fenster (Kartesische Koordinatensysteme) den TCP sehr einfach aus einer **Listbox** wählen.
- Bei der Arbeit mit dem PGH ist es ratsam diese 4 TCP Einstellungen in den Parametern **MEXTL1**, ..., **MEXTL4** zu speichern. Sie können dann während des **Teachen** des TCP sehr einfach wie folgt ändern:
 - Öffnen Sie den Hand-Modus und drücken dann die Taste ca. 3 Sekunden, dann erscheint eine Anzeige **TOOL ([])**, die sie auffordert eine Zahl n zwischen 1 und 4 einzugeben. Wählen Sie die obige Reihenfolge für die Nummerierung. Bestätigen Sie die Eingabe mit **EXE**, dann ist der TCP eingestellt, der im Parameter **MEXTL(n)** angegeben worden ist.
- Im Programmablauf können Sie jederzeit mit dem Befehl **TOOL „Name des TCP“** den gewünschten TCP einstellen.

5. Kommunikationsschnittstelle PC und Mitsubishi Roboter RV-4FL



Lesen Sie sorgfältig und beachten Sie die Mitsubishi Sicherheitshinweise, bevor Sie sich über den PC mit dem realen Roboter verbinden.

1. Spannungsversorgung für den Roboter ist eingeschaltet.
2. Wählen Sie den Modus **Auto** an der Robotersteuerung.
3. Verbinden Sie den PC und die Robotersteuerung mit einem LAN-Kabel via der Buchse am Bedienfeld, siehe **2** der Einleitung.
4. Zur Konfiguration der LAN-Verbindung benötigen Sie die IP-Adresse der Robotersteuerung. Sie ist in dem Parameter **NETIP** hinterlegt, z. B. **192.168.0.20**. Dokumentieren Sie diesen Wert.
5. Konfigurieren Sie am PC die LAN-Verbindung so, dass Sie eine feste IP-Adresse hat, wobei sich die gewählte IP-Adresse lediglich in der letzten Stelle unterscheiden soll, also z. B. 192.168.0.120.
6. Öffnen Sie in **CIROS** mit dem Menü **EINSTELLUNGEN -> Kommunikationsschnittstelle** das Fenster zur Konfiguration der Schnittstelle. Wählen Sie die Option **Netzwerk (TCP/IP)**. Sie erhalten ein Fenster, in dem Sie lediglich die IP-Adresse des Roboters (NETIP) eintragen müssen. Sie bestätigen Ihren Eintrag durch Drücken der Taste **Speichern**.
7. Sie öffnen in **CIROS®** mit dem Menü **EXTRAS -> Online-Kopplung -> RCI-Explorer** den RCI-Explorer. Mit dem RCI-Explorer können Sie die Schnittstelle zwischen PC und Robotersteuerung umfassend bedienen:



Wählen Sie eine Gruppe unter dem Roboterobjekt aus, so wird im rechten Fenster eine Zustandsbeschreibung angegeben und mit der rechten Maustaste können Sie ein Kontextmenü zu der Gruppe öffnen, z. B. können Sie bei **Verbindung** die Aktion **öffnen** oder **abbrechen** wählen. Sie erhalten eine Meldung, ob die Verbindung korrekt geöffnet werden konnte. Ist die Verbindung korrekt geöffnet, dann wählen Sie die Gruppe **Programme**. Im rechten Fenster werden alle Programme angezeigt, die in der Steuerung geladen sind. Die Gruppe **Monitore** stellt zahlreiche online Anzeigen bereit, wie z. B. den aktuellen Zustand der **Ein- und Ausgänge** der Robotersteuerung oder die **Motorströme** usw.

6. Universelle Positionsliste

Es gibt einige Positionen in der Montagestation, bei denen es extrem schwierig ist, diese selbst einzustellen. Daher ist es in diesen Fällen sinnvoll die vorbereiteten Positionen für die eigene Anwendung zu nutzen.

Hierzu gehen Sie wie folgt vor:

- Öffnen Sie in CIROS die Arbeitszelle und verbinden Sie sich mit der Robotersteuerung.
- Öffnen Sie in Ihrem CIROS Modellverzeichnis den Ordner **Pos_Rob**. Öffnen Sie die Datei **UBP.pos**.
- Öffnen Sie das CIROS Menü **EXTRAS -> Online-Kopplung** und starten Sie **Upload Realer Roboter -> PC**. Es wird die Datei **UPB.pos** auf dem PC durch die entsprechende Datei in der Steuerung überschrieben. Die Positionsliste **UBP.pos** enthält nun die entsprechenden Positionsdaten für die reale Hardware.
- Beachten Sie, dass Sie die Daten der Positionsliste in dem Ordner **Pos_Rob** nicht mehr für die Simulation nutzen können!

Aufgabe 1: Aufbau der CP® Factory Station Robotermontage

■ Lernziele

Wenn Sie diese Aufgabe bearbeitet haben,

- kennen Sie die technische Dokumentation des Mitsubishi Robotersystems RV-4FL.
- können Sie die wesentlichen Komponenten der Robotermontagestation beschreiben.
- kennen Sie die wesentlichen Bedienungsfunktionen der Robotersteuerung und des Programmierhandgerätes (PGH).
- können Sie den Roboter für die Inbetriebnahme vorbereiten.
- können Sie die Robotersteuerung und das PGH verkabeln und mit Spannung versorgen.
- können Sie die NOT-HALT Einrichtungen der Station identifizieren.
- können Sie mit dem PGH die Roboterachsen bewegen.

■ Problemstellung

Ihre Führungskraft empfiehlt Ihnen sich zunächst einen Überblick über den Aufbau der Station zu verschaffen und den Roboter für eine erste Funktionskontrolle vorzubereiten.

■ Arbeitsaufträge

1. Informieren Sie sich über die verschiedenen Typen von Industrieroboter und identifizieren Sie den gelieferten Typ.
2. Erstellen Sie eine Liste der verfügbaren technischen Dokumentation zum gelieferten Robotersystem und beschreiben Sie kurz die Zielsetzung der einzelnen Dokumente.
3. Erstellen Sie ein Technologieschema zum gelieferten Robotersystem.
4. Informieren Sie sich über den Aufbau des Systems und erstellen Sie eine Aufbauanleitung.
5. Prüfen Sie die Verkabelung des Robotersystems mit der Robotersteuerung.
6. Beschreiben Sie den Aufbau des Roboterarms.
7. Beschreiben Sie die Bedienungsfunktionen der Robotersteuerung.
8. Beschreiben Sie die Bedienung des PGH.
9. Aktivieren Sie das PGH.
10. Bewegen Sie die Roboterachsen mit dem PGH und prüfen Sie die Bewegungsrichtung.
11. Identifizieren Sie alle NOT-HALT Einrichtungen des gelieferten Robotersystems.

■ Arbeitshilfen

- Software
- Internet
- Medien
 - Technische Unterlagen des Industrieroboters
 - Technische Dokumentation der Robotermontagestation

1. Typen verschiedener Industrieroboter

Information

Industrieroboter sind universell einsetzbare Bewegungsautomaten mit mindestens drei Gelenken, deren Bewegungen hinsichtlich Bewegungsfolgen frei programmierbar sind. Die Gelenke können rotatorische oder translatorische Bewegungen erzeugen.

- a) Nennen Sie drei typische Klassen von Industrieroboter und erläutern Sie ihre charakteristischen Eigenschaften

Klasse der Portalroboter

Sie bestehen aus 3 linear unabhängigen translatorischen Gelenken. Portalroboter dienen vorwiegend dem Transport von schwerem Material über große Entfernungen.



Scara Roboter

Klasse der SCARA-Roboter

Sie bestehen aus 2 bis 3 rotatorischen Gelenken und einem translatorischen Gelenk. Die Grafik zeigt einen typischen Vertreter dieser Klasse mit 2 rotatorischen Gelenken.

Diese Roboter zeichnen sich durch ihre sehr hohe Geschwindigkeit und Steifigkeit in der Linearachse aus. Sie werden vorwiegend in der Montage bei der Bestückung von Leiterplatten eingesetzt. Der Nachteil ist, dass sie nur in einer Ebene arbeiten können.



Vertikaler 6-achsiger Knickarmroboter

Klasse der vertikalen Knickarmroboter

Sie bestehen im Allgemeinen aus 6 rotatorischen Gelenken. Die Grafik zeigt einen typischen Vertreter dieser Klasse mit 6 rotatorischen Gelenken.

Diese Roboter zeichnen sich dadurch aus, dass sie jeden Raumpunkt mit einer beliebigen Orientierung anfahren können. Dadurch sind sie sehr universell in der Automatisierung von Fertigungsprozessen einsetzbar.

- b) Ordnen Sie den gelieferten Roboter einer der genannten Klassen zu.

Das gelieferte Robotersystem ist ein 6-achsiges vertikales Knickarm-Robotersystem.

- c) Was besagt die Mitsubishi Typ Bezeichnung RV-4FL-D1?

Die Bezeichnung RV-4FL-D1 besagt, dass

- es sich um einen Knickarmroboter handelt (RV)
- das maximale Handhabungsgewicht 4 kg beträgt
- das Robotersystem vom Serientyp F ist
- der Roboter lange Arme hat (L)
- das Steuerungssystem vom Typ CR750 ist
- das Robotersystem CE zertifiziert ist (1)

2. Technische Dokumentation

- a) Erstellen Sie eine Liste der technischen Dokumentation zum Mitsubishi Roboter RV-4FL

Die technische Dokumentation besteht aus

- Bedienungsanleitung CP Factory® Roboter montagezelle, Festo Didactic SE
- Mitsubishi Instruction Manual - Robot Arm Setup & Maintenance – RV-4F Series
- Mitsubishi Controller CR750/CR751 - Standard Specifications Manual – RV-4F Series
- Mitsubishi Controller CR750/CR751 – Instruction Manual – Controller setup, basic operations, and maintenance
- Mitsubishi Controller CR750/CR751 – Instruction Manual – Detailed Explanation of Functions and Operations
- Mitsubishi Controller CR750/CR751 – Instruction Manual - Troubleshooting
- MELFA Industrieroboter – Installationsbeschreibung für RV-F-D Serie (DE)

- b) Beschreiben Sie kurz die Zielsetzung des jeweiligen Mitsubishi-Handbuches.

Die Firma Mitsubishi liefert zum Robotersystem RV-2FB eine umfangreiche technische Dokumentation, die aus einer deutschen Installationsbeschreibung und zahlreichen englischen Dokumenten besteht:

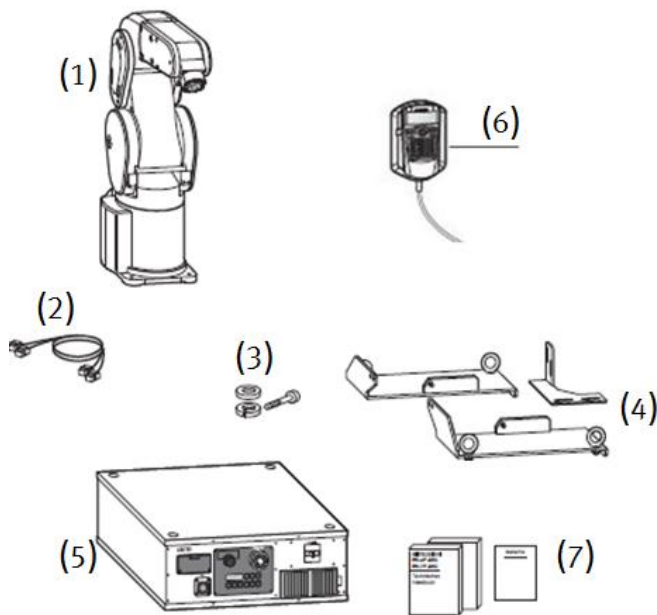
- Bedienungsanleitung CP Factory® Roboter montagezelle, Festo Didactic SE:
 - Sicherheitshinweise
 - Aufbau und Funktion
 - Bedienung
 - Elektrische und mechanische Komponenten

- Mitsubishi Instruction Manual - Robot Arm Setup & Maintenance – RV-4F Series:
 - Englisches Handbuch
 - Detaillierte Beschreibung des Aufbaus vom Roboterarm
 - Beschreibung der Bedienung der PGH
 - Wartungs- und Inspektionsbeschreibung
 - Einstellung des ORIGIN für den Roboterarm
 - Wartung und Austausch der Batterien
- Mitsubishi Controller CR750/CR751 - Standard Specifications Manual – RV-4F Series:
 - Englisches Handbuch
 - Technische Daten des Robotersystems (z. B. Wiederholgenauigkeit, Kollisionserkennung, Arbeitsraum)
 - Technische Daten und Schnittstellen des Steuerungssystems inkl. Verdrahtungsdiagrammen
 - Übersicht Programmiersprache und Parametereinstellung
 - Arbeitssicherheit und Beschreibung von Sicherheitseinrichtungen für die Arbeit mit dem Roboter
 - CE-Zertifizierung und Modelltypenbeschreibung
- Mitsubishi Controller CR750/CR751 – Instruction Manual – Controller setup, basic operations, and maintenance:
 - Englisches Handbuch
 - Installation und Inbetriebnahme des Steuerungssystems
 - Wartung und Austausch der Batterie
- Mitsubishi Controller CR750/CR751 – Instruction Manual – Detailed Explanation of Functions and Operations:
 - Englisches Handbuch
 - Bedienung der Steuerung
 - Programmierung mit der PGH
 - Beschreibung der Programmiersprache Melfa Basic V
 - Beschreibung der Multitasking Funktionalität
 - Mitsubishi Controller CR750/CR751 – Instruction Manual - Troubleshooting:
 - Englisches Handbuch
 - Beschreibung aller Fehlermeldungen und Hinweise zur Fehlerbehebung
- MELFA Industrieroboter – Installationsbeschreibung für RV-F-D Serie (DE):
 - Deutsches Handbuch
 - Detaillierte Darstellung aller Sicherheitshinweise
 - Übersicht Systemkomponenten und deren Aufbau und Verkabelung
 - Übersicht Bedienung der Steuerung CR750
 - Übersicht Bedienung der PGH R28TB, R32TB und R56TB
 - Übersicht Fehlerbeseitigung und Wartung

c) Wo finden Sie die CE-Zertifizierung für den Roboter?

- Mitsubishi Controller CR750/CR751 - Standard Specifications Manual – RV-4F Series.

3. Technologieschema



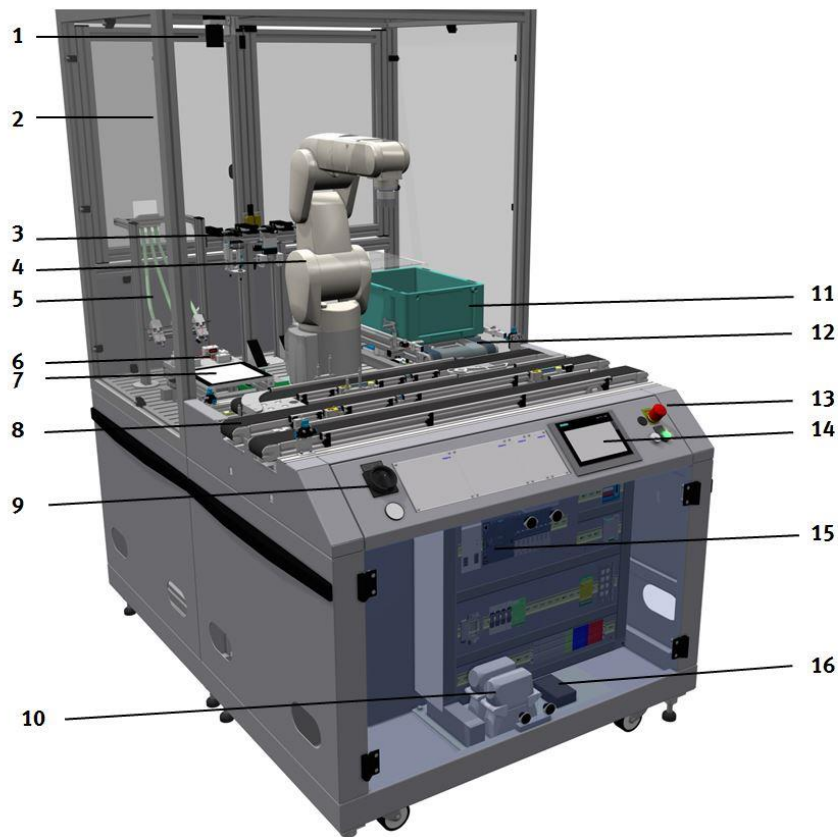
Technologieschema

- Nennen Sie die Bezeichnungen für die einzelnen Komponenten, lesen Sie hierzu die technische Dokumentation der Firma Mitsubishi.

- (1) Roboter RV-4FL
- (2) Kabelsatz
- (3) Befestigungsschrauben
- (4) Transportsicherung
- (5) Steuerung CR750
- (6) Handbediengerät (PGH) R32TB
- (7) Dokumentation

4. Aufbauanleitung

- a) Informieren Sie sich über den Aufbau der Station Roboteranmontagezelle.



Aufbau Roboteranmontagezelle

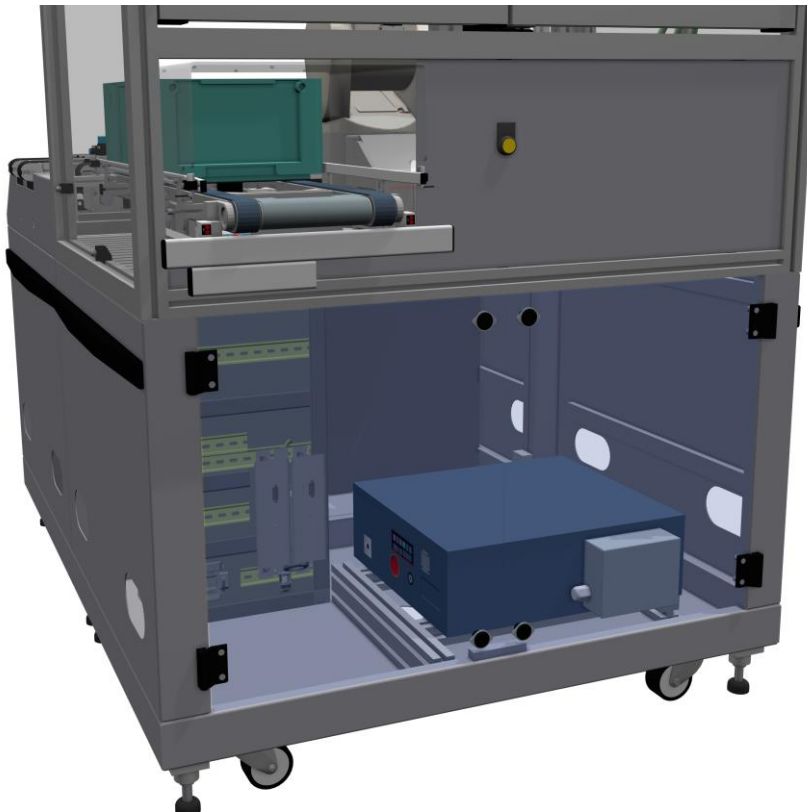
- Beschreiben Sie die Komponenten (1) - (16) in der obigen Grafik.

Beschreibung:

- (1) Kamera
- (2) Sicherheitsgehäuse
- (3) Werkzeugwechselvorrichtung
- (4) Mitsubishi Roboter RV-4FL
- (5) Zuleitungen für Sicherungen
- (6) Montageplatz
- (7) LED Flächenlicht für optische Kontrolle
- (8) Transportband mit Bypass
- (9) Hauptschalter
- (10) Multikontaktstecker
- (11) Kiste für Platinen
- (12) Transportband für den Transport der Platinenkiste
- (13) NOT-AUS Schalter am Bedienfeld
- (14) Touchpanel am Bedienfeld
- (15) E-Board für das Transportband mit Bypass
- (16) Druckschalter (optional für Energiemonitoring) und RFID

- b) Beschreiben Sie den Zugang zur Robotersteuerung und erstellen Sie eine Skizze:

Die Robotersteuerung befindet sich auf der Rückseite der Zelle hinter einer Tür, die nicht elektronisch gesichert ist.

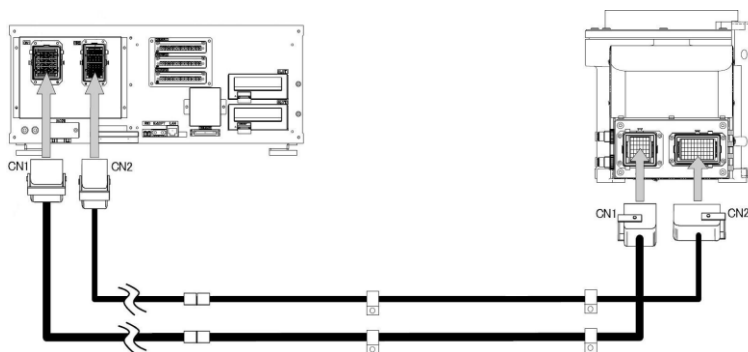


Rückseite mit Robotersteuerung CR750 mit Netzfilter

5. Erstellen Sie einen Prüfplan zur Integration des Steuerungssystems mit dem Roboter und führen Sie die Prüfung aus.

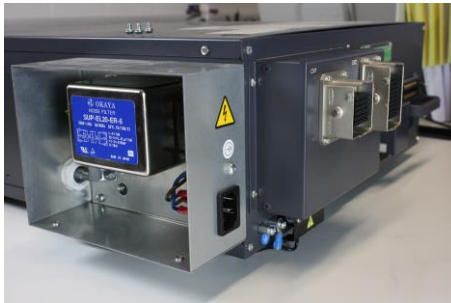
Integration Steuerungssystem

1. Roboterarm und Steuerung ist mit den beiden Kabeln entsprechend der Abbildung in Kap. 4 der Mitsubishi Installationsanleitung (DE) verbunden.



Verbindung Roboter und Steuerung

2. Netzanschluss



Montiertes Netzfiltergehäuse

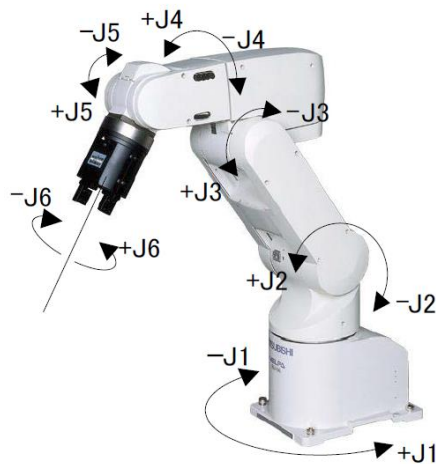


Für den Netzanschluss muss ein 230 V-Standardnetz kabel mit Kaltgerätestecker zur Spannungsversorgung am Netzfiltergehäuse angesteckt sein und mit einem Stecker auf dem E-Board auf der Rückseite verbunden sein.

3. Anschluss für NOT-HALT

Die Brücken für die NOT-HALT Schalter müssen entsprechend der Grafik Fig. 2-11 in Chapter 2 des Mitsubishi Handbuchs Controller CR750/CR751 – Instruction Manual – Controller setup, basic operations, and maintenance gesetzt sein.

6. Verschaffen Sie sich Kenntnis über die Struktur des Roboterarms und erstellen Sie eine Liste über die Achsen des Roboterarms.



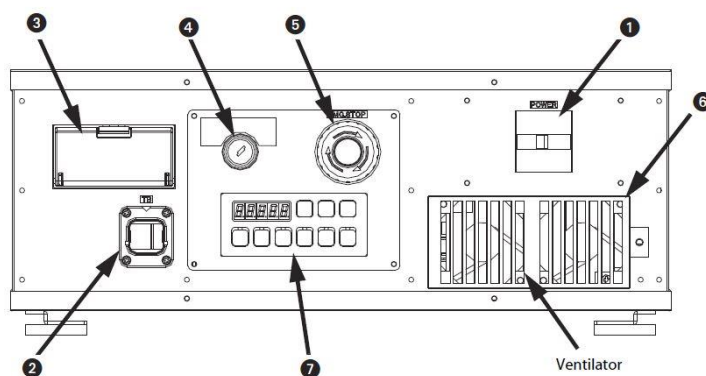
Struktur des Roboterarms RV-4FL

Liste zur Beschreibung der Achsen:

Gelenk	Beschreibung
J1	Basisachse
J2	Schulterachse
J3	Ellbogenachse

Gelenk	Beschreibung
J4	Unterarmdrehabachse
J5	Handgelenkneigungsachse
J6	Handgelenkdrehachse

7. Beschreiben Sie die Funktionen der Bedienelemente auf der Frontseite der Steuerung, siehe Kapitel 2 der Mitsubishi Installationsanleitung (DE)

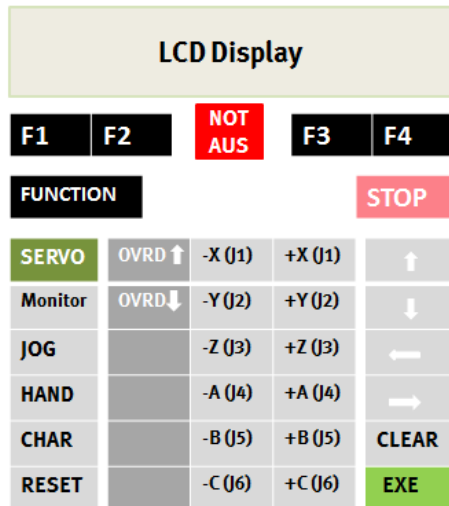


Frontseite der Steuerung

Nr.	Bezeichnung	Funktion
1	[Power]- Schalter	Ein- und Ausschalten der Steuerung
2	T/B-Anschluss	Schnittstelle für den Anschluss der PGH
3	Schnittstellenabdeckung	USB-Schnittstelle und Batterie
4	Modus-Umschalter	AUTOMATIC und MANUAL Betrieb
5	NOT-AUS - Schalter	Rastschalter für den NOT-HALT des Robotersystems
6	Filterabdeckung	
7	Bedienfeld	Statusanzeige, Bedientasten zum Ein- und Ausschalten der Servoversorgungsspannung, Starten und Stoppen der Programme

8. Basisfunktionalität des PGH kennen lernen. Erstellen Sie sich hierzu eine Grafik zur Bedienung des PGH und eine Beschreibung der Funktionen.

Die Frontseite des PGH weist folgende Bedienelemente auf:



Bedienelemente des PGH

Beschreibung der Bedienfunktionen

Bezeichnung	Funktion
NOT-HALT	NOT-HALT Schalter mit Verriegelungsfunktion
[RESET]- Taste	Quittierung einer Fehlermeldung. In Verbindung mit der [EXE]- Taste wird ein Programm zurückgesetzt.
LCD Display	Anzeige von Roboter- oder Programmstatus
[F1],...,[F4]-Taste	Ausführung der auf dem Display angezeigten Funktionen
[FUNCTION]- Taste	Umschaltung der angezeigten Funktionen
[STOP] - Taste	Unterbrechung des laufenden Programms und Anhalten der Roboterbewegung.
[OVRD]- Tasten	Änderung der Verfahrgeschwindigkeit des Roboters
12 Tasten für JOG-Betrieb: [-X(J1)] - [+C(J6)]	Im Gelenk-JOG-Betrieb können alle Gelenke einzeln bewegt werden. Im XYZ-JOG Betrieb kann der Roboterarm entlang der Koordinatenachsen bewegt werden.
[SERVO]- Taste	Betätigen Sie diese Taste bei halbdurchgedrückten Dreistufenschalter, um die Servoversorgungsspannung einzuschalten.
[Monitor]- Taste	Wechselt in den Monitor-Modus und zeigt das Monitormenü an.
[JOG]- Taste	Wechselt in den JOG-Modus und zeigt das Menü an, um den Roboter in den verschiedenen Koordinatensystemen zu bewegen.
[HAND]- Taste	Wechselt in den Hand-Modus und zeigt das Menü an, um den Greifer zu bewegen.
[CHAR]- Taste	Wechselt in den Editier-Modus zum Eingeben von Buchstaben

Pfeiltasten	Bewegt den Cursor im Display in die entsprechende Richtung
[CLEAR]- Taste	Löscht das Zeichen an der Cursor-Position
[EXE]- Taste	Bestätigung einer Dateneingabe oder Bewegung des Roboters zu einer vordefinierten Position

9. Verbinden Sie die Robotersteuerung mit dem PGH und aktivieren Sie das PGH. Dokumentieren Sie den Handlungsablauf.

Schritt 1: Untere Tür an der Rückseite der Zelle öffnen und PGH mit der Steuerung verbinden:



Schritt 2: Hauptschalter am Bedienfeld einschalten.

Schritt 3: An der Robotersteuerung den MANUAL Betrieb auswählen.

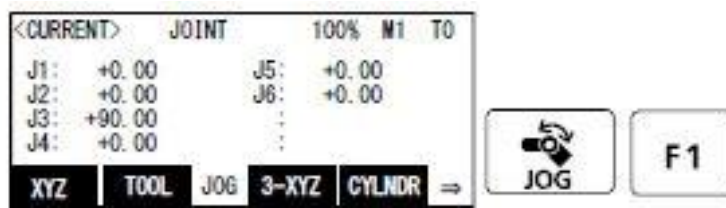
Schritt 4: Mit dem Taster [ENABLE/DISABLE] auf der Rückseite das PGH einschalten.

10. Die PGH ist aktiviert. Bewegen Sie mittels des PGH die Achsen des Roboterarms und prüfen Sie, dass sich die Achsen in die angegebene Richtung bewegen. Was passiert, wenn die Türen der Sicherheitsumhausung geöffnet werden?

Setzen Sie den Roboterarm wieder in die Ausgangstellung zurück.

Dokumentieren Sie die Ergebnisse.

2. Dreistufigen Zustimmschalter des PGH in Mittelstellung halten.
3. Betätigen der [SERVO] – Taste bis die Servomotoren eingeschaltet werden.
4. Betätigen der JOG-Taste und der F1 Taste, dann erscheint die Anzeige des JOG-Modus:



JOG-Modus

5. Mit den OVRD-Tasten kann die Geschwindigkeit reduziert oder erhöht werden.
Für den Test OVRD auf 30 % setzen.
6. Alle Achsen bewegen sich in die korrekte Richtung.
7. Die Achsen wieder in ihre Ausgangstellung zurückbewegen.
8. Das Teachen des Roboters kann auch nach Öffnung der Sicherheitstüren erfolgen.
9. PGH deaktivieren.

11. NOT-AUS Einrichtungen

Nennen Sie alle vorhandenen NOT-AUS Einrichtungen der RoboterMontagezelle und bestimmen Sie die Lage dieser NOT-AUS Einrichtungen. Beschreiben Sie den Vorgang, um das ausgelöste NOT-AUS zurückzusetzen.

NOT-AUS Einrichtungen der Roboterzelle:

- NOT-AUS Taster am Bedienfeld
- NOT-AUS durch Öffnen der Vordertür
- NOT-AUS durch Öffnen der Hintertür
- NOT-AUS Taster am PGH
- NOT-AUS Taster an der Robotersteuerung.

Zurücksetzen des NOT-AUS:

- NOT-AUS durch Öffnen der Türen:
Türen wieder schließen und am Bedienfeld den Fehler über die RESET Taste zurücksetzen.
- In den anderen Fällen, NOT-AUS Taster zurücksetzen und dann mittels der RESET Taste an der Robotersteuerung, am PGH oder Bedienfeld die Fehlermeldung zurücksetzen.

Aufgabe 2: Funktionsprüfung des Roboters RV-4FL in der Montagestation

■ Lernziele

Wenn Sie diese Aufgabe bearbeitet haben,

- können Sie einen Prüfplan für ein komplexes technisches System erstellen.
- können Sie eine Funktionsprüfung eines komplexen technischen Systems durchführen.
- können Sie die Struktur des Roboterarms eines vertikalen Knickarmroboters beschreiben.
- können Sie interne Parameter der Robotersteuerung anzeigen und ändern.
- können Sie den Roboter in seinen verschiedenen Koordinatensystemen mit dem PGH steuern.
- können Sie die NOT-HALT Einrichtungen der Station bedienen.
- kennen Sie die Grundlagen zur Bedienung des 3D-Simulationssystems CIROS.
- können Sie das virtuelle Modell der CP Factory Station Roboter montage bedienen.

■ Problemstellung

Nach dem Wareneingang und dem Aufbau einer neuen Roboterstation werden Sie beauftragt, eine Funktionsprüfung des Robotersystems durchzuführen und zu dokumentieren.

Aus Sicherheitsgründen sollen Sie diesen Vorgang zunächst an Hand eines 3D-Simulationssystems durchführen.

■ Arbeitsaufträge

1. Erstellen Sie einen Prüfplan zur Funktionsprüfung des Roboters RV-4FL.
2. Führen Sie die geplante Funktionsprüfung in der Simulationsumgebung durch und dokumentieren Sie die Ergebnisse.
3. Führen Sie die geplante Funktionsprüfung in der realen Montagestation durch und dokumentieren Sie die Ergebnisse.

■ Arbeitshilfen

- Technischen Dokumentation der CP_Factory Station Roboter montage
- Hilfe zur Software CIROS®
- Technische Dokumentation zum Mitsubishi Roboter RV-4FL

1. Erstellen Sie einen Prüfplan zur Funktionsprüfung des Roboters in der Montagezelle.

- a) Erstellen Sie einen Prüfplan für das Robotersystem, der mindestens folgendes enthalten soll:
- Sichere Ausgangsstellung des Roboters
 - Übereinstimmung der ORIGIN Daten in der PGH und auf dem Datenblatt
 - Parameterwerte für die maximal zulässigen Achswerte
 - Prüfung der NOT-HALT Schalter des Robotersystems
 - Lage des TCP
 - Verfahren des Roboters im Roboter-Koordinatensystem und im Werkzeug-Koordinatensystem

Prüfplan Robotersystem

1. Ist der Roboterarm in einer sicheren Ausgangsstellung?
2. Ist das Datenblatt mit den ORIGIN DATA verfügbar?
3. Sind die NOT-HALT Schalter entriegelt?
4. Kann das Betriebssystem der Steuerung ohne Fehlermeldung gestartet werden?
5. Wählen Sie die Anzeige der ORIGIN DATA im PGH und prüfen Sie, ob die Daten mit den Daten auf dem Datenblatt übereinstimmen.
6. NOT-HALT Schalter der Steuerung drücken. Welche Fehlermeldung wird angezeigt?
Prüfen Sie, ob die Fehlermeldung mit der Beschreibung in der Troubleshooting-Dokumentation übereinstimmt.
7. NOT-HALT entriegeln. Ist die Steuerung wieder betriebsbereit?
8. NOT-HALT Schalter der PGH drücken. Welche Fehlermeldung wird angezeigt?
Prüfen Sie, ob die Fehlermeldung mit der Beschreibung in der Troubleshooting-Dokumentation übereinstimmt.
9. NOT-HALT entriegeln. Ist die Steuerung wieder betriebsbereit?
10. NOT-HALT Schalter des Bedienfeldes drücken. Welche Fehlermeldung wird angezeigt?
Prüfen Sie, ob die Fehlermeldung mit der Beschreibung in der Troubleshooting-Dokumentation übereinstimmt.
11. NOT-HALT entriegeln. Ist die Steuerung wieder betriebsbereit?
12. Aktivieren Sie die PGH und wählen Sie den JOG-Joint Modus und prüfen Sie, ob jedes Gelenk mit der Teaching-Box bewegt werden kann.
13. Öffnen Sie das Menü PARAM(Parameterdarstellung) in der PGH und wählen Sie die Anzeige des Parameters MEJAR (maximale Gelenkwerte in negativer und positiver Richtung).
14. Prüfen Sie mittels der PGH, ob diese maximalen Werte erreicht werden.
15. Stellen Sie mittels der Teaching-Box sicher, dass der TCP die Koordinaten (0, 0, 0, 0, 0, 33.50) hat. Der TCP liegt dann im Flansch des Roboters und die x-Achse zeigt in Richtung der Werkzeugwechselandockvorrichtung.
16. Der Roboter ist in der Ausgangsstellung. Wählen Sie den JOG-XYZ Modus und prüfen Sie, ob sich der TCP ohne Änderung der Orientierung entlang der Roboter-Basiskoordinaten bewegt.
17. Der Roboter ist in der Ausgangsstellung. Ändern Sie im JOG-XYZ – Modus die Orientierung mittels der Tasten A, B, C und prüfen Sie, ob sich die Lage des TCP nicht ändert.
18. Der Roboter ist in der Ausgangsstellung. Stellen Sie den Wert des Gelenks J5 auf ca. 45°. Wählen Sie den JOG-TOOL Modus und prüfen Sie, ob sich der TCP ohne Änderung der Orientierung entlang der entsprechenden TCP-Koordinatenachsen bewegt, wenn die Tasten X, Y, Z auf der PGH gedrückt werden.
19. Der Roboter ist in der Startstellung von (20). Ändern Sie im JOG-TOOL – Modus die Orientierung mittels der Tasten A, B, C und prüfen Sie, ob sich die Lage des TCP nicht ändert und ob sich das

TCP-Koordinatensystem um die entsprechende Achse dreht.

2. Durchführung des Prüfplans in der Simulationsumgebung CIROS®

- a) Listen Sie alle Punkte des Prüfplans auf, die nicht in der Simulation durchgeführt werden können.

Liste der Punkte des Prüfplans, die nicht in der Simulation berücksichtigt werden können:
(2) – (11)

Öffnen Sie die Arbeitszelle **CP_RobotAssembly** im Verzeichnis **CIROS_CPFactory / Rob_Montage / RV-4FL** und prüfen Sie alle Punkte des Prüfplans für den Roboter in der Simulation und dokumentieren Sie diese.

Durchführung des Prüfplans für den Roboter in der Simulation:

Arbeitszelle **CP_RobotAssembly** in CIROS® öffnen.

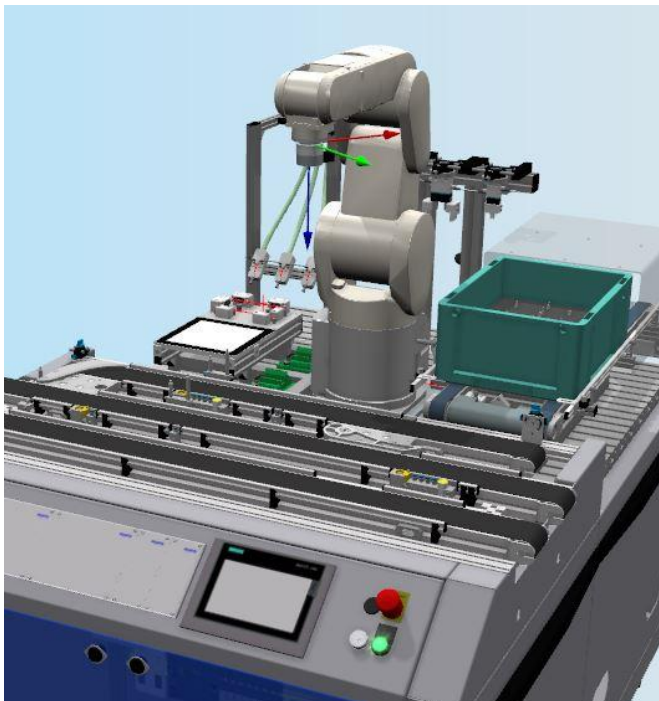
1. Roboter ist in einer sicheren Ausgangsstellung.
12. Mit der Taste **F8** die virtuelle PGH öffnen -> Das Teach-In Fenster wird angezeigt.
JOG-Joint Modus entspricht in der Simulation dem Modus **Achskoordinaten** im **Teach-In** Fenster.
Ergebnis: Durch Klicken auf die Pfeiltasten kann jedes Gelenk in positive oder negative Richtung bewegt werden. Mit **Strg+F5** den Roboter wieder in die Ausgangsstellung zurücksetzen.
13. Entsprechend der Dokumentation **Mitsubishi Standard Specification Manual**, Chapter 2.1.1, ist der Standardwert für die maximalen Gelenkwerte wie folgt gegeben:
 $J1 = (-240^\circ, +240^\circ)$; $J2 = (-120^\circ, +120^\circ)$; $J3 = (0^\circ, +164^\circ)$; $J4 = (-200^\circ, +200^\circ)$; $J5 = (-120^\circ, +120^\circ)$; $J6 = (-360^\circ, +360^\circ)$
14. **Ergebnis der Prüfuntersuchung**
J1: Geben Sie den Wert 240 in die Gelenkwertetabelle des Teach-in Fensters ein und bestätigen Sie den Wert mit der Taste **Übernehmen**. Der Roboter dreht sich um seine Basisachse bis zu dem eingestellten Wert. Klicken Sie auf die Pfeiltaste (+ Richtung), dann erscheint die Meldung **Achsgrenze erreicht**. Ein entsprechendes Resultat ergibt sich für den Wert -240°.
J2: Bringen Sie den Roboter wieder in die Grundstellung. Geben Sie den Wert -120 in die Gelenkwertetabelle des Teach-in Fensters ein und bestätigen Sie den Wert mit der Taste **Übernehmen**. Der Roboter dreht sich um sein Schultergelenk nach hinten bis zu dem eingestellten Wert. Klicken Sie auf die Pfeiltaste (- Richtung), dann erscheint die Meldung **Achsgrenze erreicht**. (Der analoge Test für den Wert +120 führt zur Kollision, die Sie in der Hardware vermeiden sollten. Drehen Sie daher zunächst den Roboter um seine Basisachse um den Wert 180°. Dann stellen Sie das Ellbogengelenk auf 0°. Jetzt können Sie ohne Kollision das Schultergelenk bis zum maximalen Wert 120° verfahren.)
J3: Bringen Sie den Roboter wieder in die Grundstellung. Geben Sie den Wert 0° in die Gelenkwertetabelle des Teach-in Fensters ein und bestätigen Sie den Wert mit der Taste **Übernehmen**. Der Roboter dreht sich um sein Schultergelenk nach hinten bis zu dem eingestellten Wert. Klicken Sie auf die Pfeiltaste (- Richtung), dann erscheint die Meldung **Achsgrenze erreicht**. Für den Test mit dem Wert +164 stellen Sie zunächst das Gelenk J4 auf 90°, um eine Kollision zu vermeiden.

J4: Bringen Sie den Roboter wieder in die Grundstellung. Geben Sie die Werte $+200^\circ$ und -200° in die Gelenkwertetabelle ein und wie zuvor können Sie testen, dass mit diesen Werten die maximalen Achsgrenzen erreicht werden.

J5: Verfahren Sie wie zuvor mit den Werten $+120^\circ$ und -120° .

J6: Verfahren Sie wie zuvor mit den Werten $+360^\circ$ und -360° .

15. Die Anzeige des eingestellten TCP erfolgt in CIROS durch eine grafische Anzeige: Öffnen Sie das Menü **EINSTELLUNGEN** -> **Modelloptionen** -> **Darstellung** -> **Koordinatensystem** und wählen dann die **Anzeige des TCP** und bestätigen dies mit **OK**, dann erhalten Sie folgendes Bild:



Hieraus folgt sofort die korrekte Einstellung.

16. Öffnen Sie das Teach-In Fenster. Der Modus **JOG-XYZ** entspricht im Teach-In Fenster dem Modus **Kartesische Koordinaten** mit dem Bezugssystem Roboterbasis. Blenden Sie das Roboterbasiskoordinatensystem ein. Hierzu wählen Sie das Menü **EINSTELLUNGEN** -> **Modelloptionen** -> **Darstellung** -> **Koordinatensysteme** -> **Roboter** -> **Basiskoordinatensystem**. Klicken Sie auf die linearen Pfeiltasten zu **X**, **Y** und **Z**, dann können Sie optisch verfolgen, dass sich der TCP ohne Änderung der Orientierung entlang der entsprechenden Koordinatenachsen bewegt. Sie können dies auch numerisch verfolgen. Im rechten Fenster des Teach-In ändern sich die Werte der Orientierungswinkel nicht, sondern nur der entsprechende Koordinatenwert.
17. Bringen Sie den Roboter in die Ausgangsstellung. Den Orientierungswinkeln **A**, **B**, **C** auf der PGH entsprechen im Teach-In Fenster die gekrümmten Pfeile zur **X**, **Y**, **Z** Achse. Die linke(rechte) Pfeiltaste entspricht einer negativen (positiven) Drehung um die entsprechende Koordinatenachse. Sie können visuell oder durch Beobachtung der Koordinatenwerte im rechten Fenster verfolgen, dass sich die Position des TCP bei einer beliebigen Orientierungsänderung

nicht verändert.

18. Bringen Sie den Roboter in die Ausgangsstellung. Wählen Sie den Modus **Achskoordinaten** und stellen Sie den **J5-Gelenkwert** auf **45°**.

Der Modus **JOG-TOOL** entspricht im Teach-In Fenster dem Modus **Kartesische Koordinaten** mit dem Bezugssystem TCP. Klicken Sie auf die linearen Pfeiltasten zu **X**, **Y** und **Z**, dann können Sie optisch verfolgen, dass sich der TCP ohne Änderung der Orientierung entlang der entsprechenden TCP-Koordinatenachsen bewegt.

Beachten Sie, dass z. B. bei Änderung des X-Wertes im TCP-Koordinatensystem sich die (x, y, z) Koordinaten relativ zum Roboterkoordinatensystem ändern können.

19. Bringen Sie den Roboter in die Ausgangsstellung. Klicken Sie auf die gekrümmten Pfeile neben den Achsen X, Y, Z dann können Sie visuell oder durch Beobachtung der Koordinatenwerte im rechten Fenster verfolgen, dass sich die Lage des TCP bei einer beliebigen Orientierungsänderung nicht verändert und dass sich das Koordinatensystem bei der linken (rechten) Pfeiltaste in negativer (positiver) Richtung um die entsprechende TCP-Koordinatenachse dreht.

3. Funktionsprüfung für die reale CP_Factory Station Roboter montage

Hinweis:

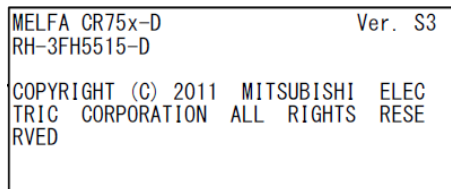
Wenn Sie sich mit Ihrem PC mit der Robotersteuerung verbinden möchten, dann stecken Sie das Ethernetkabel vom PC in die Anschlussbuchse am Bedienfeld (siehe Grafik).

Die korrekte IP-Adresse von der Robotersteuerung entnehmen Sie dem Parameter **NETIP** der Robotersteuerung.

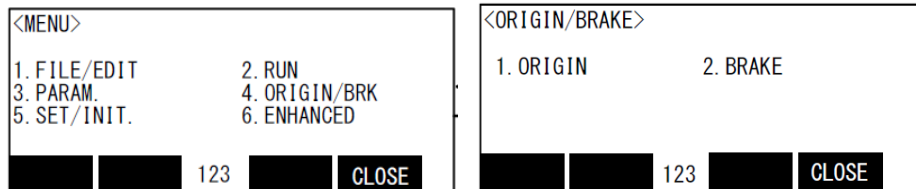
- a) Funktionsprüfung des Robotersystems an Hand des erstellten Prüfplans

Funktionsprüfung des Robotersystems

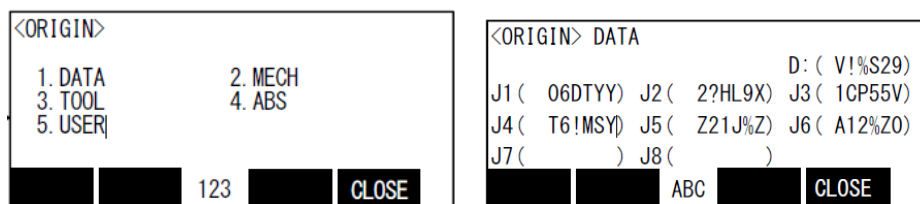
1. Ist in gewünschter Ausgangsstellung.
2. **Hinweis**
Falls kein Datenblatt vorhanden, dann Lieferant kontaktieren.
3. Die **NOT-HALT** Schalter an der Robotersteuerung, an der PGH und am Bedienfeld prüfen, ob die Schalter entriegelt sind und gegebenenfalls diese entriegeln.
4. Prüfen Sie, dass die Spannung an der Robotersteuerung eingeschaltet und der **Automatik Modus** gewählt ist. Dann schalten Sie die Spannung am Bedienfeld ein. Prüfen Sie, dass dann die Robotersteuerung einwandfrei startet.
Stellen Sie sicher, dass
 - die PGH mit der Steuerung verbunden ist
 - die Steuerung betriebsbereit und im Modus **MANUAL** geschaltet ist.Aktivieren Sie das PGH wie in Aufgabe 1. Es erscheint folgende Meldung auf der Anzeige des PGH:



Drücken Sie die **EXE**-Taste des PGH und es erscheint das Hauptmenü:



Wählen Sie mittels der Cursor Tasten das Menü **ORIGIN/BRK** und drücken Sie die **EXE**-Taste. Es erscheint die nebenstehende Anzeige. Öffnen Sie das Menü **ORIGIN**. Sie erhalten 5 verschiedene Untermenüs. Öffnen Sie das Menü **DATA**, das folgende Anzeige liefert, wobei die Daten hier nur ein Beispiel sind:



Prüfen Sie, ob diese Daten mit den Werten auf dem Datenblatt zur Einstellung des **ORIGIN** übereinstimmen. Falls dies nicht der Fall ist, dann tragen Sie die Daten in Ihr Datenblatt mit einer Datumsanzeige ein. Mit der Taste F4 gelangen Sie zurück ins Hauptmenü.

Stellen Sie sicher, dass

- die Robotersteuerung mit dem Roboter verbunden ist,
- das Spannungskabel der Steuerung angeschlossen ist und
- die PGH nicht aktiviert ist.

Schalten Sie die Spannung an der Steuerung ein. Sobald das Betriebssystem der Steuerung funktionsbereit ist, erscheint auf dem Display des **Operation Panels** eine Statusanzeige, z. B. **o 70**, die besagt, dass die Geschwindigkeit auf 70% der Maximalgeschwindigkeit eingestellt ist.

5. Drücken Sie den **NOT-HALT** Schalter an der Steuerung.

Funktionsverhalten ist korrekt, wenn

- Warnton ertönt und
- Fehlermeldung **H0060** auf dem Display angezeigt wird.

6. Entriegeln Sie den **NOT-HALT** Schalter an der Steuerung und bestätigen Sie die Fehlermeldung mit der **RESET**-Taste. Die Steuerung sollte dann wieder betriebsbereit sein.

7. Drücken Sie den **NOT-HALT** Schalter an der PGH.

Funktionsverhalten ist korrekt, wenn

- Warnton ertönt und
- Fehlermeldung **H0070** auf dem Display angezeigt wird.

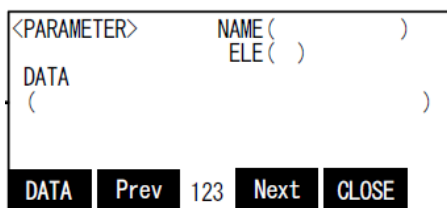
Entriegeln Sie den **NOT-HALT** Schalter an der PGH und bestätigen Sie die Fehlermeldung mit der **RESET**-Taste. Die Steuerung sollte dann wieder betriebsbereit sein.

8. Drücken Sie den **NOT-HALT** Schalter am Bedienfeld.
Funktionsverhalten ist korrekt, wenn
 - Warnton ertönt und
 - Fehlermeldung **H0050** auf dem Display angezeigt wird.
9. Entriegeln Sie den **NOT-HALT** Schalter am Bedienfeld und bestätigen Sie die Fehlermeldung mit der **RESET**-Taste. Die Steuerung sollte dann wieder betriebsbereit sein.
10. Die PGH ist aktiviert und funktionsbereit. Der Roboterarm ist in der Ausgangsstellung.
Drücken Sie die **JOG**-Taste der Teaching-Box und wählen Sie den **Joint**-Modus. Im Display erscheint die Winkelangabe für die einzelnen Achspositionen. Halten Sie den **ENABLE**-Schalter in der Mittelstellung, drücken Sie die **SERVO**-Taste und warten bis sich Servo-Motoren einschalten. Reduzieren Sie die Geschwindigkeit auf 30%.
Drücken Sie die **+J1 / -J1** Taste.
Funktionsverhalten ist korrekt, wenn
 - sich der Roboter in positive Richtung bzw. negative Richtung um die Basisachse dreht.
 Prüfen Sie, ob dies auch für die Gelenke **J2, ..., J6** gilt.

Hinweis

Nach erfolgreichem Test bringen Sie die Achsen wieder in ihre Ausgangsstellung.

11. Die PGH ist aktiviert und das Hauptmenü wird angezeigt. Öffnen Sie das Menü **PARAM**.



Tragen Sie bei **Name (...)** die Bezeichnung **MEJAR** ein. Zur Eingabe von Buchstaben müssen Sie zuvor die Taste **CHAR** drücken. Bestätigen Sie Ihre Eingabe mit der **EXE**-Taste. Es erscheinen unter **DATA** die minimalen und maximalen Werte für die Gelenkachsen, z. B. -240, +240, -120, +120, 0, +164, -200, +200, -120, +120, -360, +360).
Notieren Sie sich die angezeigten Werte. Mit der Taste **F4** gelangen Sie wieder zurück ins Hauptmenü.

12. Zur Einstellung der minimalen und maximalen Gelenkwerte gehen Sie vom Ablauf wie in der Simulation vor. Falls Sie in die Nähe der Endlage kommen müssen Sie die Geschwindigkeit reduzieren. Wählen Sie z.B. 5%. Ist die Endlage erreicht, so ertönt ein Warnton.

Prüfen Sie zunächst mit der PGH die Einstellung des **TCP**. Öffnen Sie das Menü **PARAM** im Hauptmenü der PGH. Geben Sie **MEXTL** als Name ein und bestätigen Sie ihre Eingabe mit der **EXE**-Taste. Sie erhalten z.B. folgende Anzeige: (0.0, 0.0, 0.0, 0.0, 0.0, 0.0). Zur Änderung der Daten müssen Sie den Modus **DATA** aktivieren. Gehen Sie mit dem Cursor auf den letzten Wert und ändern Sie diesen in **33.5**. Mit der **EXE**-Taste bestätigen Sie wieder Ihre Eingabe. Sie werden gefragt, ob Sie diese Änderung übernehmen wollen und wählen **YES**. Schließen Sie das Menü und gehen wieder zurück ins Hauptmenü.

Hinweis

Der neue Parameterwert wird erst wieder übernommen, wenn die Steuerung neu gestartet wird. Schalten Sie die Steuerung aus und starten die Steuerung erneut. Aktivieren Sie die PGH und öffnen im Hauptmenü wieder das Menü PARAM. und prüfen, ob der Parameter MEXTL den neuen Wert angenommen hat.

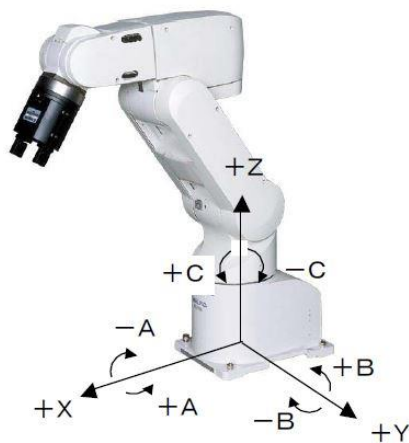
13. Die Robotersteuerung ist eingeschaltet und die PGH ist funktionsbereit. Der Roboterarm ist in der Ausgangsstellung. Falls dies nicht der Fall ist, dann wählen Sie den **JOG-JOINT** Modus und bringen die Achsen in die gewünschte Stellung.

Sie wählen den **JOG-XYZ** Modus. Im Display der PGH werden die aktuelle Position des **TCP** und die Orientierungswinkel angezeigt. Schalten Sie die Servo-Motoren ein. Drücken Sie die Tasten -X oder +X.

Funktionsverhalten ist korrekt, wenn

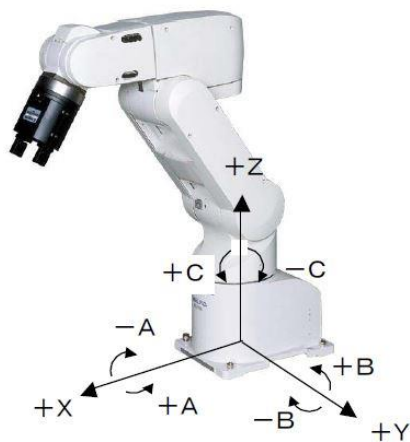
- der **TCP** (Mittelpunkt der Unterseite des Greifers) sich entlang der X-Achse in negativer oder positiver Richtung bewegt,
- sich nur die x-Koordinate des TCP ändert und
- keine Winkeländerung erfolgt.

Untersuchen Sie, ob dies auch entsprechend bei Drücken der Tasten (-Y, +Y) und (-Z, +Z) gilt, vgl. folgende Grafik.



JOG XYZ Modus, Tasten X, Y und Z

14. Die Robotersteuerung ist eingeschaltet, und die PGH ist funktionsbereit. Der Roboterarm ist in seiner Ausgangsstellung. Sie wählen wie zuvor in (16) den **JOG-XYZ** Modus. Schalten Sie die Servo-Motoren ein und drücken eine der Tasten (-A, +A), (-B, +B), (-C, +C). Das Funktionsverhalten ist korrekt, wenn
- keine Positionsveränderung des **TCP** erfolgt und
 - sich die Ausrichtung des Greifers entsprechend der Drehung um die gewählte Achse sich ändert (siehe folgende Grafik).

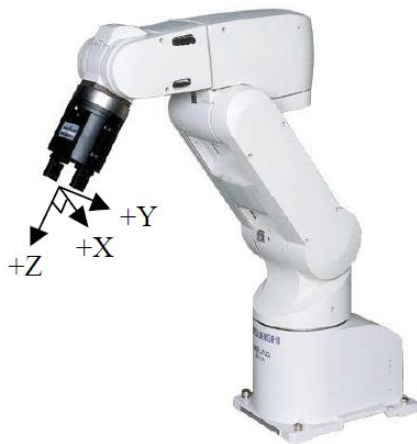


JOG XYZ Modus, Tasten A, B und C

15. Die Robotersteuerung ist eingeschaltet, und die PGH ist funktionsbereit. Der Roboterarm befindet sich in seiner Ausgangsstellung. Zur besseren Visualisierung der folgenden Untersuchung wählen Sie zunächst den **JOG-Joint** Modus und stellen das Gelenk auf ca. 45°. Wählen Sie den **JOG-TOOL** Modus. Bewegen Sie den Roboter durch Drücken der Tasten (-X, +X), (-Y, +Y), (-Z, +Z).

Funktionsverhalten ist korrekt, wenn

- sich der **TCP** in der korrekten Richtung entlang der entsprechenden **TCP**-Koordinatenachse bewegt und
- sich die Orientierungswinkel A, B, C nicht ändern.

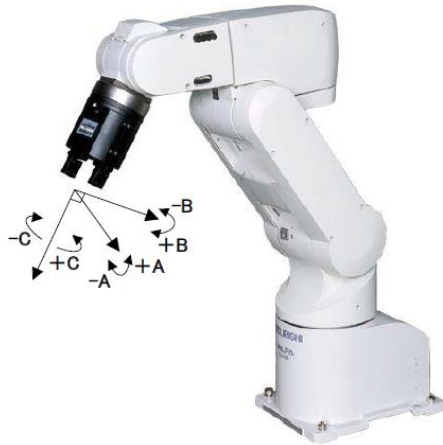


JOG TOOL Modus, Tasten X, Y und Z

16. Die Robotersteuerung ist eingeschaltet, und die PGH ist funktionsbereit. Der Roboterarm befindet sich in der Ausgangsstellung wie in (16). Schalten Sie die Servo-Motoren ein und drücken Sie eine der Tasten (-A, +A), (-B, +B), (-C, +C).

Funktionsverhalten ist korrekt, wenn

- sich Position des **TCP** nicht verändert,
- sich die Ausrichtung des Greifers entsprechend folgender Grafik ändert:



JOG TOOL Modus, Tasten A, B und C

Aufgabe 3: Werkzeugwechselsystem für einfache Pick&Place Anwendung nutzen

■ Lernziele

Wenn Sie diese Aufgabe bearbeitet haben,

- kennen Sie die Grundfunktionen der Mitsubishi Roboterprogrammiersprache Melfa Basic
- können Sie den Ablaufplan für eine einfache Roboteranwendung erstellen.
- können Sie Positionen in CIROS für eine Roboteranwendung einlernen (teachen) und in eine Positionsliste eintragen.
- können Sie eine Kollisionsprüfung in der Simulation durchführen.
- kennen Sie den unterschiedlichen Einsatz von PTP und linearen Bewegungen eines Robotersystems.
- kennen Sie die Funktionalität eines Greiferwechselsystems und können Sie in einer einfachen Roboteranwendung in CIROS anwenden, testen und optimieren.
- können Sie die Kommunikationsschnittstelle zwischen CIROS und der realen Robotersteuerung konfigurieren und den Download von Programmen und Positionslisten in die reale Steuerung durchführen.
- kennen Sie die Sicherheitsregeln bei der Kommunikation zwischen PC und der realen Steuerung.
- können Sie das Programm im Debug-Modus analysieren und optimieren.

■ Problemstellung

Die Funktionsprüfung der neuen Roboterstation ist erfolgreich durchgeführt worden. Die neue Station soll der Produktionsleitung vorgestellt werden. Zur Präsentation werden Sie von Ihrem Gruppenleiter beauftragt eine einfache Anwendung zu erstellen. Aus Sicherheitsgründen sollen Sie diesen Vorgang zunächst an Hand des 3D-Simulationssystems CIROS® durchführen und dann die Anwendung nach Abnahme durch den Gruppenleiter in die reale Hardware übertragen.

■ Arbeitsaufträge

An der Stopper-Position 1 im Bypass steht ein Palettenträger mit Palette und einer Frontschale bereit. Der Roboter besitzt keine Greifvorrichtung. Die Aufgabe für den Roboter ist es, die Frontschale zu greifen, sie in der Montagestation abzulegen und wieder auf den Palettenträger zurückzulegen. Danach soll der Roboter das notwendige Greifersystem wieder in die Wechsellvorrichtung zurücklagern.

1. Erstellen Sie einen Lageplan zur Aufnahme des Werkstückgreifers und für die Zurücklagerung des Werkstückgreifers.
2. Erstellen Sie ein Melfa Basic V Programm , so dass der Roboter
 - den Werkstückgreifer aus der Wechsellvorrichtung entnimmt,
 - den Greifer öffnet und zur Startposition fährt
 - den Greifer wieder schließt und zur Wechsellvorrichtung fährt,
 - den Greifer wieder zurücklagert,
 - aus der Wechsellvorrichtung herausfährt
 - und in die Startposition zurückfährt.

Testen Sie ihr Programm zunächst in der Simulation.

3. Erstellen Sie ein Melfa Basic V Programm für den gesamten Prozessablauf, testen und optimieren Sie das Programm in der Simulation. Dokumentieren Sie Ihr Programm.
4. Führen Sie eine Inbetriebnahme des erstellten Programms an der realen Roboterstation durch.

■ Arbeitshilfen

- Mitsubishi Controller CR750/CR751 – Instruction Manual – Detailed Explanation of Functions and Operations
- CIROS® Hilfe -> Programmierung -> Melfa Basic V
- Kapitel 5 in den Grundlagen
- Technische Dokumentation CP_Factory Roboterstation, Kapitel 2 und Kapitel 4.6
- Einleitung zu den Aufgaben

Hinweis:

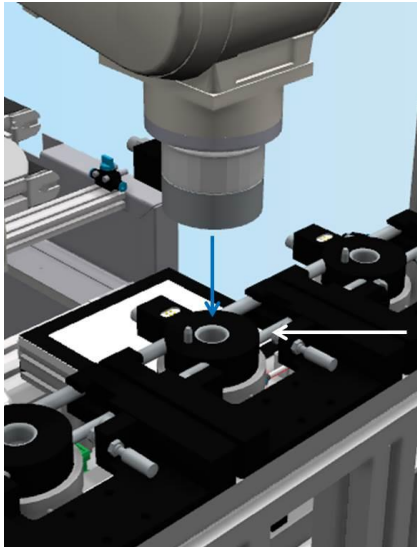
Am Flansch des Roboters befindet sich die Vorrichtung zur Übernahme eines Greifersystems aus der Wechselvorrichtung. Sie ist verbunden mit einer pneumatischen Steuerung, mit der sowohl das gewünschte Greifersystem aufgenommen und abgegeben, als auch der Greifer geöffnet und geschlossen wird. Hierzu gibt es vorbereitete Programme, die in ihr Projekt übernommen werden können:

- “GRPRelease“: Greifer aus der Wechselvorrichtung entnehmen
- “GRPLock“: Greifer an die Wechselvorrichtung übergeben
- “GRPOpen“: Greifer öffnen
- “GRPClose“: Greifer schließen.

1. Lageplan zur Aufnahme und Zurücklagerung des Werkzeuggreifers

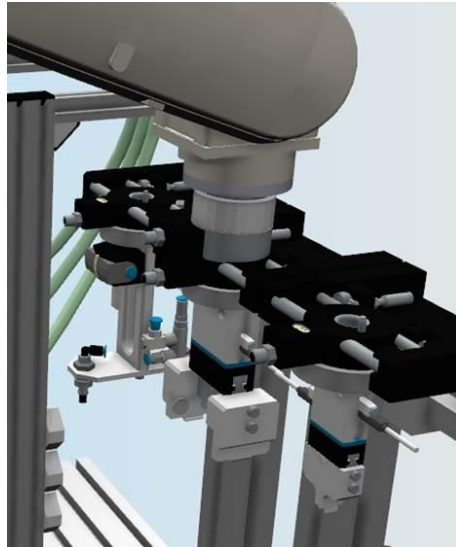
Lösung:

Schritt 1: Anfahren zur Aufnahme des
Werkzeuggreifers



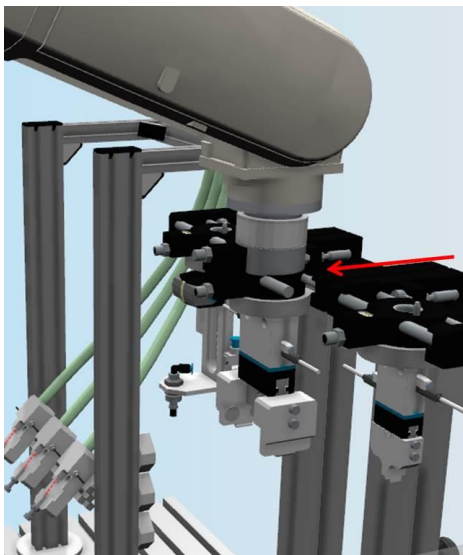
Roboter fährt zur Position oberhalb des
Werkzeuggreifers

Schritt 2: Aufnahme des Werkzeuggreifers



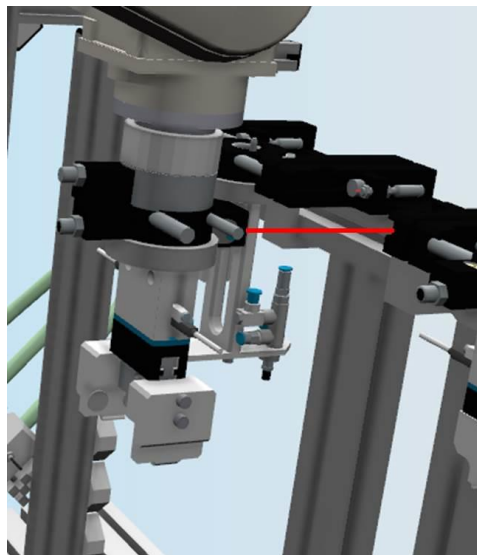
Roboter fährt linear zur Aufnahmeposition.

Schritt 3: Anfahren zur Aufnahme des
Werkzeuggreifers

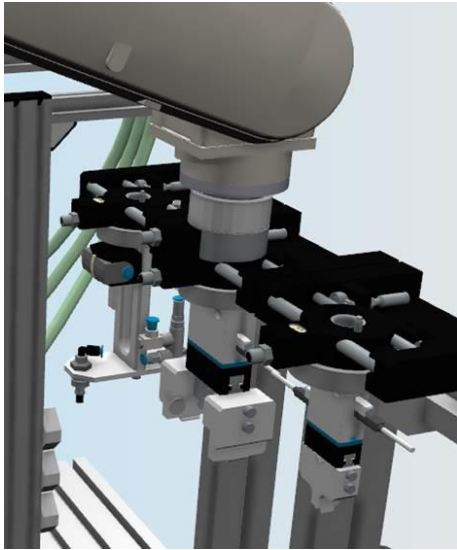


Nach Aufnahme des Greifers fährt der
Roboter linear in positive X-Richtung ohne
Änderung der Aufnahmehöhe.

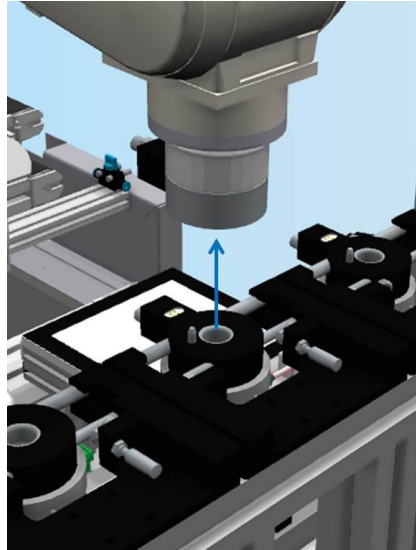
Schritt 4: Anfahren der Wechsellvorrichtung , um
den Greifer zurück zu lagern.



Roboter fährt zur Endposition von Schritt 3.

Schritt 5: Ablage des Werkzeuggreifers

Roboter fährt linear in X-Richtung zur Ablageposition.

Schritt 6: Werkzeuggreifer zurücklagern

Nach Loslösen des Greifers fährt der Roboter linear nach oben aus der Haltevorrichtung heraus.

2. Programm zur Aufnahme und Zurücklagerung des Werkzeuggreifers

- a) Öffnen Sie die Arbeitszelle **CP_RobotAssembly** im Verzeichnis **CIROS_CPFactory /Rob_Montage /RV-4FL**. Der Roboter hat keinen Greifer angeschlossen und steht in der Nähe der Startposition. Erstellen Sie ein Programm, so dass der Roboter den Werkzeuggreifer anschließt, zur Startposition fährt und dann den Greifer wieder zurücklagert. Achten Sie darauf, dass die Anschlussvorrichtung am Roboter nicht verriegelt ist.

Hinweis

Das Projekt A3.prj ist schon angelegt. Es besteht aus dem Hauptprogramm A3.mb5 mit der Positionsliste A3.pos und den oben erwähnten externen Programmen. Alle notwendigen Positionen sind schon in der Positionsliste eingetragen. Im Programm sind schon die Basiseinstellungen für Sie zur Vereinfachung vorgenommen worden.

Lösung:

- Öffnen Sie die Arbeitszelle **CP_RobotAssembly.modx**. Es wird das Modellfenster, das Programmfenster mit dem vordefinierten Programmnamen A3.mb5 und die Positionsliste A3.pos angezeigt.
- Zur Prüfung des TCP öffnen Sie das **Teach-In Fenster** mit der Option **Kartesische Koordinaten** und wählen bei TCP aus der Liste **TCP_None**.
- Programm und Positionsliste:

[Montage_RV-4FL] \Programme\A3.pos *

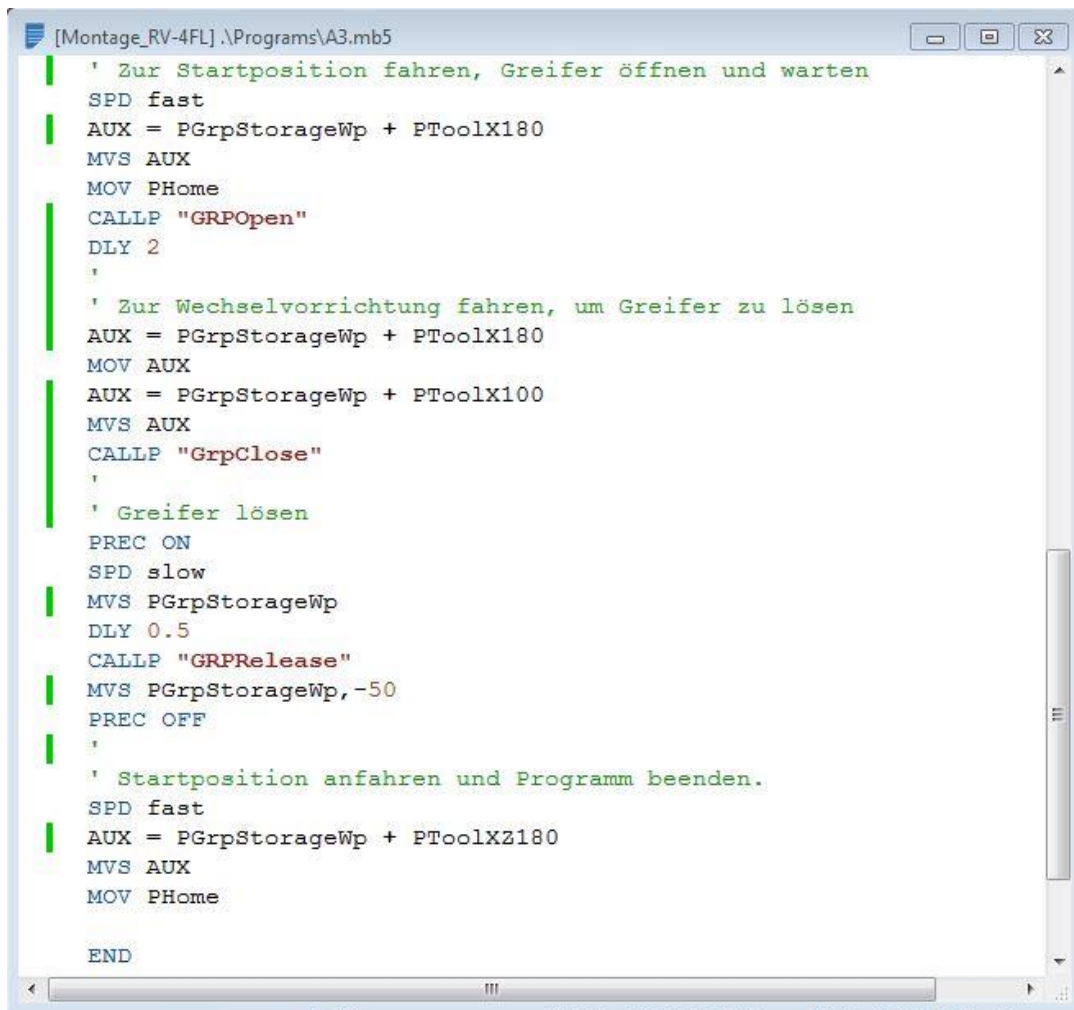
Nr.	Position	Orientierung	Kommentar
PHome	250.3,1.4,544.3	180,0,-90,R,A,N	Startposition
PGrpStorageWp	-395.6,0.0,484.9	-180,0,0,R,A,N	Lagerplatz f
PToolX180	180.0,0.0,0.0	0,0,0	Hilfsposition
PToolX100	100.0,0.0,0.0	0,0,0	Hilfsposition
PToolXZ180	180.0,0.0,50.0	0,0,0	Hilfsposition

[Montage_RV-4FL] \Programs\A3.mb5

```

' Aufgabe 3
' Externe Programme:
' "GRPLock" - Greifer an der Wechsellvorrichtung übernehmen
' "GRPrelease" - Greifer an die Wechsellvorrichtung übergeben
' "GRPClose" - Greifer öffnen
' "GRPClose" - Greifer schließen
' -----
' Positionen
DEF POS PtGripperNone ' Position für TCP am Flansch
DEF POS PtGripperWP ' Position für TCP des Werkzeuggreifers
DEF POS AUX ' Hilfsposition
' -----
DEF INTE slow ' Variable für Lineargeschwindigkeit
DEF INTE fast
' -----
' Parameter setzen
PtGripperWP =(0,0,170,0,0,33.50)
PtGripperNone =(0,0,0,0,0,33.50)
slow = 20
fast = 150
' -----
' Programmstart
TOOL PtGripperNone ' TCP setzen
JOVRD 30 ' Geschwindigkeitsparameter für Achsen
SPD fast ' Lineargeschwindigkeit
MOV PHome
'
' Zur Wechsellvorrichtung fahren
CALLP "GRPrelease" ' Anschlussvorrichtung entriegeln
AUX = PGrpStorageWp + PToolXZ180
MOV AUX
MVS PGrpStorageWp,-50
'
' Werkzeuggreifer aufnehmen
SPD slow
PREC ON ' Sehr genaues Bahnfahren
MVS PGrpStorageWp
DLY 0.5
CALLP "GRPLock"
AUX = PGrpStorageWp + PToolX100
MVS AUX
PREC OFF

```

```

[Montage_RV-4FL] .\Programs\A3.mb5
' Zur Startposition fahren, Greifer öffnen und warten
SPD fast
AUX = PGrpStorageWp + PToolX180
MVS AUX
MOV PHome
CALLP "GRPOpen"
DLY 2
'
' Zur Wechsellvorrichtung fahren, um Greifer zu lösen
AUX = PGrpStorageWp + PToolX180
MOV AUX
AUX = PGrpStorageWp + PToolX100
MVS AUX
CALLP "GrpClose"
'
' Greifer lösen
PREC ON
SPD slow
MVS PGrpStorageWp
DLY 0.5
CALLP "GRPRelease"
MVS PGrpStorageWp, -50
PREC OFF
'
' Startposition anfahren und Programm beenden.
SPD fast
AUX = PGrpStorageWp + PToolX2180
MVS AUX
MOV PHome
END

```

- b) Datensicherung: Speichern Sie Programm und Positionsliste unter dem Namen **A3**.
 Zur Vorbereitung der Inbetriebnahme an der Hardware speichern Sie das Programm und die Positionsliste unter dem Namen **A3R** im Verzeichnis **CIROS_CPFactory / Rob_Montage /RV-4FL /Programs** ab. Können Sie die Positionsdaten aus der Simulation in Ihrer realen Anwendung nutzen?

Lösung:

Die Positionsdaten aus der Simulation sind sehr gute Annäherungswerte, aber aufgrund von Montagetoleranzen können Abweichungen auftreten, die korrigiert werden müssen.

Öffnen Sie die universelle Positionsliste **UBP.pos** im Verzeichnis **Pos_Rob**, siehe **§6 der Einleitung** zu den Aufgaben. Diese enthält alle Positionsdaten zu Ihrer Hardware-Station.

Position PHome in A3R.pos:

Doppelklick auf die Position **P_Home** in der universellen Positionsliste. Der Roboter springt an diese Position. Übernehmen Sie diese Position für PHome.

Position PGrpStorageWp:

Doppelklick auf die Position **P_GrpStorageWp** in der universellen Positionsliste. Der Roboter springt an diese Position. Übernehmen Sie diese Position für PGrpStorageWp.

- c) Schalten Sie die Spannungsversorgung an dem Bedienfeld ein. Die Arbeitszelle **CP_RobotAssembly** ist geöffnet und der PC ist mit der Robotersteuerung verbunden, siehe **§ 2** in der Einleitung zu den Aufgaben.

Führen Sie einen Download Ihres Programms **A3R** und der zugehörigen Positionsliste in die Robotersteuerung durch. Prüfen Sie, ob das Programm im RCI Explorer angezeigt wird.

Hinweis:

Aktivieren Sie Ihr Programmfenster und wählen das Menü EXTRAS -> Online-Kopplung -> Download PC -> realer Roboter

Bestätigen Sie das Dialogfenster mit OK. Verfahren Sie analog für die Positionsliste.

Die externen Programme GrpLock, GrpRelease, GrpClose, GrpOpen sind schon in der Steuerung und müssen nicht mehr geladen werden.

Lösung:

- Schalten Sie die Spannungsversorgung am Bedienfeld ein..
- Führen Sie einen Download Ihres Programms **A3R.mb5** und der Positionsliste **A3R.pos** entsprechend der Anleitung im Hinweis durch. Sie erhalten jeweils eine Meldung, ob der Download erfolgreich durchgeführt werden konnte.
- Wählen Sie die Gruppe **Programme** im **RCI-Explorer**. Im rechten Fenster wird der Programmname **A3R.mb5** angezeigt.

- d) Die Arbeitszelle **CP_RobotAssembly** ist geöffnet und der PC ist mit der Robotersteuerung verbunden. Programm und Positionsliste wurden in die Robotersteuerung geladen. Prüfen Sie die Position PGrpStorageWp in der realen Station, beschreiben Sie die Optionen, um Ihr Programm über den RCI-Explorer zu starten und zu stoppen und testen Sie Ihr Programm.

Hinweis:

Lesen Sie sorgfältig die Hilfe zum RCI-Explorer in der CIROS® Hilfe.

Lösung:

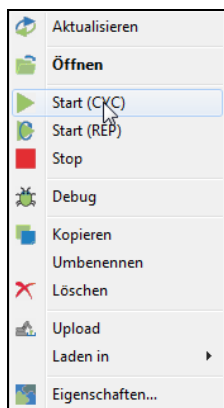
Position PGrpStorageWp:

- Wählen Sie den Modus **MANUAL** an der Robotersteuerung und aktivieren Sie das PGH.
- Öffnen Sie das Menü **1. FILE/EDIT** im Hauptmenü Fenster. Es werden alle Programme angezeigt.
- Wählen Sie Ihr Programm **A3R** und drücken Sie die Taste **F2 [POS.]**. Es werden die Positionen der Positionsliste angezeigt. Mittels der Tasten **[NEXT]** und **[PREV]** können Sie zwischen den Positionen wechseln.
- Prüfen Sie, dass der TCP im Flansch des Roboters eingestellt ist.
- Wählen Sie die Anzeige der Position PGrpStorageWp zu Programm **A3R**.
- Wählen Sie den **JOG-Modus** und bewegen den Roboter so, dass die Anschlussvorrichtung vom Roboter oberhalb der Aufnahme positioniert ist.
- Wählen Sie den **XYZ-Modus** und reduzieren Sie den **OVRD auf 10%**.
- Achten Sie darauf, dass die Anschlussvorrichtung nicht verriegelt ist, d.h. die Stifte im Anschluss sind nicht ausgefahren. Andernfalls wählen Sie den **HAND-Modus** und drücken **(+X)**.
- Schalten Sie wieder in die Anzeige der Position **PGrpStorageWp** und bewegen den Roboter mit **MOVE** in die Zielposition. Im Ausnahmefall müssen Sie kleine Korrekturen vornehmen.

- Wählen Sie den **HAND-Modus** und schließen Sie den Greifer mittels der Handtaste **(-X)** an den Roboter, siehe **3.5** der Einleitung.
- Wechseln Sie zurück in den **JOG-Modus** und bewegen den Roboter ca. 60 mm in **+X**-Richtung. Der Test ist positiv, wenn der Greifer sich nahezu geräuschlos aus der Halterung löst.
- Wechseln Sie wieder zurück in die Positionsanzeige und bewegen den Roboter im XYZ-Modus mit der **MOVE**-Taste zurück in die Zielposition und lösen den Greifer.
- Wechseln Sie in den JOG-Modus und bewegen den Roboter im XYZ-Modus ca. 50 mm in **+Z**-Richtung und dann ca. 100mm in **+X**- Richtung.
- Wechseln Sie zurück in die Positionslistenanzeige und wählen die Position **PHome**. Bewegen Sie den Roboter mit **MOVE** zurück in die Startposition.

Programmablauf:

- Deaktivieren Sie das **PGH** und wählen Sie den Modus **AUTO** an der Robotersteuerung.
- Selektieren Sie das Programm **A3R.mb5** in der rechten Fensteranzeige des RCI-Explorers. Mit rechtem Mausklick öffnet sich ein Dialogfenster:



- Mit den beiden Befehlen **Start(CYC)** und **Start(REP)** wird das Programm gestartet, wobei **CYC** bedeutet, dass das Programm einen Zyklus durchläuft, und **REP** bedeutet, dass der Programmablauf nach dem Ende stets neu gestartet wird.
- Mit dem Befehl **Stop** kann das Programm jederzeit angehalten werden.
- Mit dem Befehl **Debug** können Sie sich das Programm online anzeigen lassen und den Programmablauf schrittweise analysieren. Sie können Haltepunkte setzen und/oder den Ablauf im Einzelschrittmodus fortsetzen.
- Starten Sie Ihr Programm zunächst im Einzelschrittmodus. Falls der Ablauf fehlerfrei ist, verlassen Sie den **Debug**-Modus und starten das Programm mit der Option **Start(CYC)**.

3. Programm in der Simulationsumgebung CIROS® erstellen

- Öffnen Sie die Arbeitszelle **CP_RobotAssembly** im Verzeichnis **CIROS_CPFactory /Rob_Montage /RV-4FL** mit dem Programmfenster **A3.mb5**. Fügen Sie Sie ihren vollständigen Programmablaufplan als Kommentarzeilen in das vorhandene Programmfenster ein.

Lösung: Programmablaufplan in Form von Kommentarzeilen im Programmfenster:

```

[Montage_RV-4FL] \Programme\A3.mb5 *
-----
' Programmstart
TOOL PtGripperNone
JOVRD 30
SPD fast
MOV PHome
'
' Warten bis Frontschale an Stopperposition bereit steht
'|
' Zur Wechsellvorrichtung fahren
AUX = PGrpStorageWp + PToolXZ180
MOV AUX
MVS PGrpStorageWp, -50
'
' Werkzeuggreifer aufnehmen
SPD slow
PREC ON          ' Sehr genaues Bahnfahren
MVS PGrpStorageWp
DLY 0.5
CALLP "GRPLock"
AUX = PGrpStorageWp + PToolX100
MVS AUX
PREC OFF
'
' Zur Startposition fahren ,Greifer öffnen und warten
SPD fast
CALLP "GRPOpen"
MOV PHome
DLY 2
'
' Zur Stopperposition fahren und Frontschale greifen
' Frontschale in Montageposition ablegen
' Lageprüfung vornehmen und Roboter fährt zurück
' Werkstück einspannen und Roboter fährt in Startposition
' Nach kurzer Wartezeit holt Roboter Frontschale aus Montageposition
' Frontschale wird auf Palettenträger abgelgt
'
' Zur Wechsellvorrichtung fahren, um Greifer zurück zu lagern.
AUX = PGrpStorageWp + PToolX180
MOV AUX
AUX = PGrpStorageWp + PToolX100

```

- b) Mit Hilfe der Beschreibung der Ein- und Ausgangssignale in §1 der Einleitung zu den Aufgaben fügen Sie die Deklaration der Ein- und Ausgänge in das Programm ein.

Lösung:

' Eingänge

```

DEF IO DI_WPAvail      = Bit,251  ' Signal von der SPS
DEF IO DI_WPAvail      = Bit,17   ' Werkstück in Montagestation
DEF IO DI_WPClamped    = Bit,16   ' Werkstück eingespannt
DEF IO DI_WPOrientOk   = Bit,18   ' Orientierung des Werkstücks ist OK
DEF IO DI_WPHoleOK     = Bit,19   ' Bohrung ist an der richtigen Position

```

' Ausgänge

```

DEF IO DO_WPMountLock  = Bit,16   ' Werkstück einspannen

```

- c) Die Positionsliste **A3.pos** ist noch nicht vollständig. Für die Aufgabenstellung müssen Sie noch die Werte für die Stopper-Position = **PCarrierStop1** und für die Montageposition = **PAssembleWp** ermitteln.

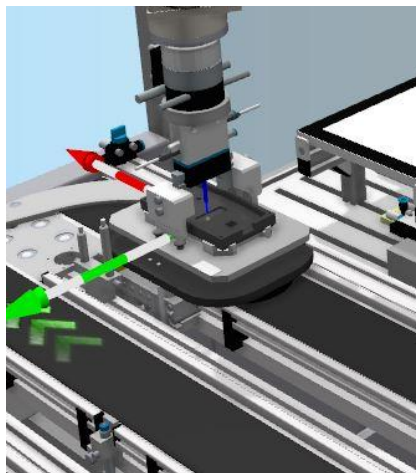
Hinweis

Zum Teachen dieser Positionen ist es notwendig, eine Frontschale bereit zu stellen. Hierzu fügen Sie einfach am Anfang des Programmes die folgende Warteschleife hinzu:

```
' Programmstart
TOOL PtGripperNone
JOVRD 30
SPD fast
MOV PHome
WAIT DI_WPAAtStopAvail = 1
.....
END
```

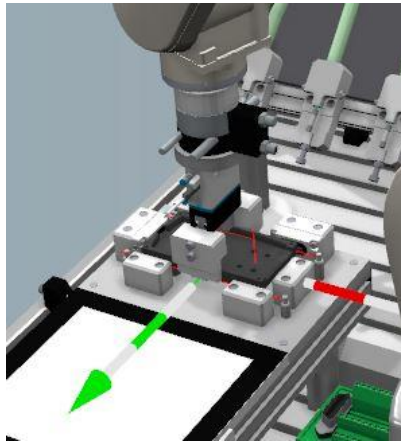
Lösung:

- Kompilieren Sie das erweiterte Programm, starten Sie es und stoppen es, wenn der Roboter mit geöffneten Werkzeuggreifer sich in der Startposition **PHome** befindet.
- Teachen der Position **PCarrierStop1**:
 - Ändern Sie den **TCP**: Wählen Sie im **Teach-In** Fenster **TCP_WP** als neuen TCP.
 - Öffnen Sie den Modell-Explorer (**Strg+T**) und markieren den **Greifpunkt GP_Frontschale_unten** der Basis von der Komponente **Werkstücke->CP_Frontschale**. Wählen Sie im Menü **Modellierung/3D Marker** die Option **3D Marker -> Selektion**. Visualisieren Sie den **3D-Marker**.
 - Der Greifvorgang wird in der Simulation dadurch realisiert, dass ein **Greiferpunkt** sich in der Nähe des **Greifpunktes** befindet. Im vorliegenden Fall ist der neue **TCP** identisch mit dem **Greiferpunkt** des Werkzeuggreifers.
 - Wählen Sie im Menü **Modellierung /3D Marker** die Option **TCP -> 3D Marker**. Dann springt der TCP des Roboters in die Position des 3D Markers und damit der Greiferpunkt des Werkzeuggreifers in den Greifpunkt der Frontschale.



- Übernehmen Sie die Position in Ihre Positionsliste (**Strg+F2**) und bezeichnen die Position mit **PCarrierStop1**. Zum Testen der Position schließen Sie den Greifer und machen einen Doppelklick auf die Position **PHome**. Der Test ist erfolgreich, wenn die Frontschale sich mitbewegt.

- Teachen der Position **PAssembleWp**:
 - Öffnen Sie den Modell-Explorer (**Strg+T**) und markieren den **Greiferpunkt GPP_Montage** der Basis von der Komponente **MontagePosition**. Wählen Sie im Menü **Modellierung/3D Marker** die Option **3D Marker -> Selektion**.
 - Setzen Sie den TCP in den 3D Marker, dann wird die Frontschale einwandfrei mittig in die Montageposition abgelegt.



- Position in die Positionsliste übernehmen und mit **PAssembleWp** bezeichnen.
- Setzen Sie die Zelle wieder zurück in den Ausgangszustand (**Strg+F5**).

- d) Erstellen Sie den vollständigen Programmablauf.
- Ersetzen Sie schrittweise Ihre Kommentarzeilen durch Programmsequenzen.
 - Überlegen Sie sich, wie Sie die Hilfspositionen in der Positionsliste nutzen können, um die Wechsellvorrichtung kollisionsfrei anfahren und verlassen zu können.
 - Testen Sie den Ablauf und dokumentieren Sie Ihre Testergebnisse

Lösung: Programm

```

[Montage_RV-4FL] \Programme\A3.mb5
' Aufgabe 3
' Externe Programme:
' "GRPlock" - Greifer an der Wechsellvorrichtung übernehmen
' "GRPRelease" - Greifer an die Wechsellvorrichtung übergeben
' "GRPClose" - Greifer öffnen
' "GRPClose" - Greifer schließen
' -----
' Eingänge
DEF IO DI_WPAStopAvail = Bit,251 ' Signal von der SPS
DEF IO DI_WPAvail      = Bit,17  ' Werkstück ist in
                              ' Montagestation vorhanden
DEF IO DI_WPCLamped    = Bit,16
DEF IO DI_WPOrientOk   = Bit,18  ' Orientierung des Werkstücks ist OK
DEF IO DI_WPHoleOK     = Bit,19  ' Bohrung ist an der richtigen Position
'
' Ausgänge
DEF IO DO_WPMountLock  = Bit,16  ' Werkstück einspannen
'
' Positionen
DEF POS PtGripperNone  ' Position für TCP am Flansch
DEF POS PtGripperWP    ' Position für TCP im Zentrum des
                      ' Werkzeuggreifers
DEF POS AUX            ' Hilfsposition
' -----
DEF INTE slow          ' Variable für Lineargeschwindigkeit
DEF INTE fast
' -----
' Parameter setzen
PtGripperWP = (0,0,170,0,0,33.50)
PtGripperNone = (0,0,0,0,0,33.50)
slow = 20
fast = 150
' -----
' Programmstart
TOOL PtGripperNone
JOVRD 30
SPD fast
MOV PHome
'
' Warten bis Palettenträger an Stopperposition bereit steht
WAIT DI_WPAStopAvail = 1
'

```

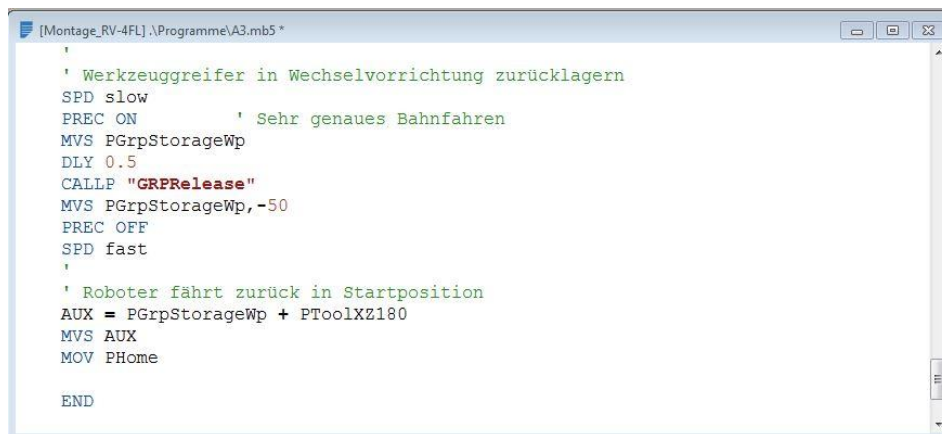


```

[Montage_RV-4FL] \Programme\A3.mb5
' Zur Wechselvorrichtung fahren
AUX = PGrpStorageWp + PToolXZ180
MOV AUX
MVS PGrpStorageWp,-50
'
' Werkzeuggreifer aufnehmen
SPD slow
PREC ON          ' Sehr genaues Bahnfahren
MVS PGrpStorageWp
DLY 0.5
CALLP "GRPLock"
AUX = PGrpStorageWp + PToolX100
MVS AUX
PREC OFF
'
' Zur Startposition fahren ,Greifer öffnen und warten
SPD fast
CALLP "GRPOpen"
MOV PHome
DLY 2
'
' Zur Stopperposition fahren und Frontschale greifen
TOOL PtGripperWP
MOV PCarrierStop1,-30
MVS PCarrierStop1
DLY 0.5
CALLP "GRPClose"
DLY 0.5
MVS PCarrierStop1,-50
'
' Frontschale in Montageposition ablegen
MOV PAssembleWP,-100
MVS PAssembleWP,-30
SPD slow
MVS PAssembleWP
DLY 0.5
CALLP "GRPOpen"
'
' Lageprüfung vornehmen und Roboter fährt zurück
SPD fast
WHILE (DI_WPAvail = 0) OR (DI_WPOrientOk = 0) OR (DI_WPHoleOK = 0)
WEND
MVS PAssembleWP,-100
'

[Montage_RV-4FL] \Programme\A3.mb5
' Werkstück einspannen und Roboter fährt in Startposition
DO_WPMountLock = 1
WAIT DI_WPClamped = 1
Tool PtGripperNone
MOV PHome
'
' Nach kurzer Wartezeit Frontschale aus Montage holen
DLY 2
Tool PtGripperWP
MOV PAssembleWP,-100
MVS PAssembleWP,-30
SPD slow
MVS PAssembleWP
DLY 0.5
CALLP "GRPClose"
DO_WPMountLock = 0
DLY 0.5
SPD fast
MVS PAssembleWP,-100
'
' Frontschale auf Palettenträger zurücklegen
MOV PCarrierStop1,-100
MVS PCarrierStop1,-30
SPD slow
MVS PCarrierStop1
DLY 0.5
CALLP "GRPOpen"
SPD fast
MVS PCarrierStop1,-80
Tool PtGripperNone
MOV PHome
'
' Zur Wechselvorrichtung fahren, um Greifer zurück zu lagern.
AUX = PGrpStorageWp + PToolX180
MOV AUX
AUX = PGrpStorageWp + PToolX100
MVS AUX
CALLP "GRPClose"
'

```



```

[Montage_RV-4FL] \Programme\A3.mb5
'
' Werkzeuggreifer in Wechsellvorrichtung zurücklagern
SPD slow
PREC ON          ' Sehr genaues Bahnfahren
MVS PGrpStorageWp
DLY 0.5
CALLP "GRPrelease"
MVS PGrpStorageWp,-50
PREC OFF
SPD fast
'
' Roboter fährt zurück in Startposition
AUX = PGrpStorageWp + PToolXZ180
MVS AUX
MOV PHome
END

```

Testergebnis

- Programmablauf einwandfrei.

- e) Datensicherung: Speichern Sie Programm und Positionsliste unter dem Namen **A3**.
Zur Vorbereitung der Inbetriebnahme an der Hardware speichern Sie das Programm und die Positionsliste unter dem Namen **A3R** im Verzeichnis **CIROS_CPFactory / Rob_Montage /RV-4FL /Programs** ab.

4. Inbetriebnahme des erstellten Programms an der realen Roboterstation

- a) Ersetzen Sie in **A3R.pos** die eingelesenen Positionen **PGrpStorageWp** , **PHome** , **PCarrierStop1** und **PAssembleWp** durch die entsprechenden Positionen der universellen Positionsliste **UBP.pos** in Ihrem Verzeichnis **Pos_Rob**, siehe §2 der Einleitung.

Lösung:

Die Positionsdaten aus der Simulation sind sehr gute Annäherungswerte, aber aufgrund von Montagetoleranzen können Abweichungen auftreten, die korrigiert werden müssen.
Öffnen Sie die universelle Positionsliste **UBP.pos** im Verzeichnis **Pos_Rob**, siehe §6 der Einleitung zu den Aufgaben. Diese enthält alle Positionsdaten zu Ihrer Hardware-Station.

Position PHome in A3R.pos:

Doppelklick auf die Position **P_Home** in der universellen Positionsliste. Der Roboter springt an diese Position. Übernehmen Sie diese Position für PHome.

Position PGrpStorageWp:

Doppelklick auf die Position **P_GrpStorageWp** in der universellen Positionsliste. Der Roboter springt an diese Position. Übernehmen Sie diese Position für PGrpStorageWp.

Position PCarrierStop1:

Doppelklick auf die Position **P_CayrrierStop1** in der universellen Positionsliste. Der Roboter springt an diese Position. Übernehmen Sie diese Position für PCarrierStop1.

Position PAssembleWp:

Doppelklick auf die Position **P_AssmbleWp** in der universellen Positionsliste. Der Roboter springt an diese Position. Übernehmen Sie diese Position für PAssembleWp.

- b) Schalten Sie die Spannungsversorgung an dem Bedienfeld ein und setzen Sie die Station in die Ausgangsstellung wie in **1.5** der Einleitung beschrieben.
- Die Arbeitszelle **CP_RobotAssembly** ist geöffnet und der PC ist mit der Robotersteuerung verbunden, siehe §2 in der Einleitung zu den Aufgaben.
- Führen Sie einen Download Ihres Programms **A3R** und der zugehörigen geänderten Positionsliste in die Robotersteuerung durch. Prüfen Sie, ob das Programm im RCI Explorer angezeigt wird.

Lösung:

- Führen Sie die in **1.5** der Einleitung beschriebenen Schritte durch, um die Ausgangsstellung zu erreichen.
- Führen Sie einen Download Ihres Programms **A3R.mb5** und der Positionsliste **A3R.pos** entsprechend der Anleitung im Hinweis durch. Bestätigen Sie das Überschreiben des schon vorhandenen Programmes A3R. Sie erhalten jeweils eine Meldung, ob der Download erfolgreich durchgeführt werden konnte.
- Wählen Sie die Gruppe **Programme** im **RCI-Explorer**. Im rechten Fenster wird der Programmname **A3R.mb5** angezeigt.
- **Hinweis:** Die externen Programme **GRPLock, GRPRelease, GRPClose, GRPOpen** sind schon in der Steuerung und müssen nicht mehr geladen werden.

- c) Programm und Positionsliste wurden in die Robotersteuerung geladen. Können Sie bedenkenlos Ihr Programm starten? Wenn nein, welche Daten müssen Sie vor dem Programmstart noch prüfen?

Nein, es müssen folgende Daten geprüft werden:

- Es müssen die Verdrahtung der Eingänge **DI_WPAtStopAvail, DI_WPAvail (Bit 17), DI_WPOrientOk (Bit18), DI_WPHoleOK (Bit19)** und des Ausgangs **DO_WPMountLock (Bit16)** geprüft werden. Es ist zu beachten, dass entsprechend der technischen Dokumentation, siehe auch **1.2** in der Einleitung, das Signal vom Sensor **DI_WPAtStopAvail** mit dem Eingangsbit 2151 verdrahtet sein soll.
- Im Programm **A3R** ist daher folgende Änderung im Deklarationsteil notwendig:
DEF IO DI_WPAtStopAvail = Bit, 2151
 Diese Änderung können Sie mit dem PGH vornehmen oder Sie führen die Änderung auf dem PC im Programm A3R durch und laden es erneut nach der Änderung in die Steuerung.

- d) Prüfen Sie die Verdrahtung der Eingänge und des Ausgangs im Programm **A3R**.

Lösung:

- Die Station ist in der Ausgangsstellung, siehe **1.5** der Einleitung.
- Eingänge **DI_WPAvail, DI_WPOrientOk, DI_WPHoleOK:**
 Öffnen Sie den **RCI-Explorer** und starten das Menü **Monitore**. Wählen Sie die Option **Eingänge** und klicken dann auf **Importieren**. Geben Sie **15** als **Startwert** ein und aktivieren die Option **EIN**. Es werden die Eingänge zu den entsprechenden Bit-Nummern angezeigt. Aktivieren und deaktivieren Sie manuell die Eingänge und stellen Sie sicher, dass die Reaktionen korrekt sind.
- Ausgang **DO_WPMountLock:**
 Wählen Sie im Menü **Monitore** die Option **Ausgänge**. Importieren Sie diese und wählen wieder 15

als Startwert. Aktivieren Sie die Online-Anzeige. Legen Sie ein Werkstück in den Montageplatz. Setzen Sie den Ausgang und prüfen Sie, ob das Werkstück eingespannt wird.

- Eingang **DI_WPAStopAvail:**

Setzen Sie den Startwert für die Anzeige der Eingänge auf 2150. Dann sollte der Eingang angezeigt werden. Entnehmen Sie das Werkstück vom Transportträger, dann muss der Eingang auf 0 zurückschalten.

- e) Programm und Positionsliste wurden in die Robotersteuerung geladen. Prüfen Sie die vier Positionen **PGrpStorageWp**, **PHome**, **PCarrierStop1** und **PAssembleWp** in der realen Umgebung.

Hinweis:



Sie finden eine detaillierte Beschreibung zur Handhabung des PGH zum Einrichten von Positionen in Kapitel 3 des Mitsubishi Handbuchs **Mitsubishi Controller CR750/CR751/ Instruction Manual – Detailed Explanation of Functions and Operations**.

Lösung:

- Wählen Sie den Modus **MANUAL** an der Robotersteuerung und aktivieren Sie das PGH.
- Öffnen Sie das Menü **1. FILE/EDIT** im Hauptmenü Fenster. Es werden alle Programme angezeigt.
- Wählen Sie Ihr Programm **A3R** und drücken Sie die Taste **F2 [POS.]**. Es werden die Positionen der Positionsliste angezeigt. Mittels der Tasten **[NEXT]** und **[PREV]** können Sie zwischen den Positionen wechseln.
- Die Positionen **PHome** und **PGrpStorageWp** wurden schon zuvor geprüft.
- Position **PCarrierStop1:**
 - Prüfen Sie, dass der TCP auf **TCP_None** eingestellt ist.
 - Wählen Sie den JOG Modus und holen Sie mit dem PGH wie in 2(d) beschrieben den Werkzeuggreifer und fahren den Roboter in die Startposition **PHome**.
 - Öffnen Sie den Greifer.
 - Wählen Sie mittels der Hand-Taste die TCP Einstellung **TOOL([2]) = TCP_WP**.
 - Wählen die Position **PCarrierStop1** in der Positionslistenanzeige.
 - Aktivieren Sie den **JOG**-Modus und bewegen den Roboter im **XYZ**-Modus möglichst senkrecht und mittig oberhalb des Werkstücks.
 - Mit dem **MOVE**-Kommando bewegen Sie den Roboter in die Position **PCarrierStop1**.
 - Die Greifposition ist korrekt, wenn beim Öffnen und Schließen des Greifers das Werkstück nicht sichtbar die Lage verändert.
 - Falls die Position nicht korrekt sein sollte, dann müssen Sie im **XYZ**-Modus den Roboter neu justieren, so dass der Greifvorgang einwandfrei funktioniert.
 - Schließen Sie dann den **JOG**-Modus und übernehmen Sie mit der **TEACH**-Tastenfunktion gegebenenfalls die neue Position.
- Position **PAssembleWp:**
 - Sie haben das Werkstück gegriffen.
 - Wählen Sie die Position **PAssembleWp** in der Positionsliste **A3R**.
 - Aktivieren Sie den **JOG**-Modus und bewegen den Roboter im **XYZ**-Modus oberhalb und möglichst mittig vom Montageplatz.
 - Mit der **MOVE**-Tastenfunktion bewegen Sie den Roboter in die Position **PAssembleWp**.
 - Die Position ist korrekt, wenn folgende Bedingungen erfüllt sind:

- Die Sensoren **DI_WPAvail**, **DI_WPOrientOk**, **DI_WPHoleOK** sind aktiviert.

f) Programm und Positionsliste sind in der Robotersteuerung aktualisiert worden. Testen Sie mittels des RCI-Explorers den Programmablauf und optimieren Sie die Geschwindigkeit. Dokumentieren Sie Ihre Ergebnisse für eventuelle Änderungen im Programmablauf.

- Wählen Sie zum Test Ihres Programms im RCI-Explorer zunächst den **Debug**-Modus. In der Online-Anzeige Ihres Programms sind die Programmzeilen nummeriert und die Kommentare sind gelöscht. Mit  können Sie **Haltepunkte** in den entsprechenden Programmzeilen Ihres Programms setzen.
- Das Programm wird jeweils vor Ausführung der entsprechenden Programmzeile gestoppt.
- Mit dem Befehl  können Sie Ihr Programm mit Haltepunkte-Überwachung starten.

g) Testen Sie Ihr Programm im Automatikmodus

- Starten Sie Ihr Programm über das Bedienfeld der Robotersteuerung.
 - Wählen Sie das Programm **A3R** aus.
 - Die **OVRD** Auswahl bewirkt eine prozentuale Reduzierung der im Programm definierten maximalen Geschwindigkeit.
 - Schalten Sie die Servo Motoren ein und starten Sie das Programm.

h) Datensicherung: Führen Sie einen Upload des aktuellen Roboterprogramms und der aktuellen Positionsliste in der Robotersteuerung auf den PC durch.

Die Arbeitszelle **CP_RobotAssembly** ist geöffnet und der PC ist mit der Robotersteuerung verbunden. Öffnen und aktivieren Sie das Programmfenster mit dem Programm **A3.mb5**. Wählen Sie das Menü **EXTRAS -> Online-Kopplung -> Upload realer Roboter -> PC**. Das Programm wird auf den PC geladen und überschreibt das vorhandene Programm **A3R**. Führen Sie das ebenfalls für die Positionsliste **A3R.pos** durch.

Hinweis:

Sie können die neue Positionsliste A3R.pos nicht mehr für die Simulation nutzen!

- i) Dokumentieren Sie Ihren Programmcode.

Hinweis

Zu einer vollständigen Dokumentation sind folgende Angaben notwendig:

1. Kopfbereich:

- Programmname
- Kurze Beschreibung
- Name des Autors
- Freigabedatum mit Versionsnummer
- Änderungen mit Datumsangabe
- Wichtige Parametereinstellungen (z. B. Positionsliste, TCP Einstellung)

2. Quellcodebereich:

- Kurze Beschreibung von Programmsequenzen

Ergänzen Sie entsprechend Ihre Dokumentation für **A3R**.

Programm A3 für CP Factory Station Roboterontage mit Mitsubishi Roboter RV-4FL.
Autor: Hans Mustermann Freigabe: 20.11.2015 / Version 1.0 Änderungen: IP Adresse Robotersteuerung: 192.168.0.20 IP Adresse PC: 192.168.0.120 Port Nr. Robotersteuerung: 10001
Hauptprogramm A3R : Palettenträger mit Frontschale wartet in Stopper Position 1. Roboter entnimmt Frontschale und legt sie in dem Montagebereich ab. Roboter legt die Frontschale wieder auf den Palettenträger und lagert Werkzeuggreifer in die Wechselvorrichtung ein. Externe Programme: GRPLock - Greifer an der Wechselvorrichtung übernehmen GRPRelease - Greifer an die Wechselvorrichtung übergeben GRPClose - Greifer öffnen GRPClose - Greifer schließen

Aufgabe 4: Platine auf Werkstück montieren und Prüfung des Werkzeugwechselsystems

■ Lernziele

Wenn Sie diese Aufgabe bearbeitet haben,

- können Sie Projekte mit mehreren Programmen erstellen.
- kennen Sie den Einsatz globaler Variablen.
- können sie die Positionen in CIROS für eine Roboteranwendung einlernen (teachen) und in eine Positionsliste eintragen.
- können Sie Testszenarien für Roboterfunktionen erstellen.
- können Sie die Kommunikation mit einer externen SPS anwenden.
- können Sie die Kommunikationsschnittstelle zwischen CIROS und der realen Robotersteuerung konfigurieren und den Download von Programmen und Positionslisten in die reale Steuerung durchführen.
- können Sie Programm im Debug-Modus im Abgleich mit der Hardware analysieren und optimieren.

■ Problemstellung

Die Präsentation der neuen Station war erfolgreich und Sie werden beauftragt sich darauf vorzubereiten, eine professionelle Anwendung für die neue Station in Betrieb zu nehmen. Ihr Gruppenleiter beauftragt Sie eine Testumgebung für die Prüfung des Wechselgreifersystems durch den Roboter im Rahmen einer Bestückung eines Werkstücks mit einer Platine zu erstellen. Aus Sicherheitsgründen sollen Sie Ihr Konzept zunächst an Hand des 3D-Simulationssystems CIROS® durchführen und dann die Anwendung nach Abnahme durch den Gruppenleiter in die reale Hardware übertragen.

■ Arbeitsauftrag

An der Stopper-Position 1 im Bypass steht ein Palettenträger mit Palette und einer Frontschale bereit. Der Roboter besitzt keine Greifvorrichtung. Die Aufgabe für den Roboter ist es, die Frontschale zu greifen, sie in der Montagestation abzulegen und mit einer Platine zu bestücken. Nach der Bestückung soll das montierte Werkstück auf den Palettenträger zurückgelegt werden.

1. Erstellen Sie in der Simulation ein Testprogramm, so dass der Roboter prüft, ob er einen vorgegebenen Greifer angeschlossen hat und im negativen Fall den vorhandenen Greifer zurücklagert und den korrekten Greifer anschließt.
2. Erstellen Sie in der Simulation ein Programm, das
 - a. einen beliebigen Greiferstatus für den Roboter generiert,
 - b. den Roboter den Greifertyp überprüfen lässt,
 - c. die Frontschale in die Montageposition einspannt
 - d. auf Anforderung eine bestimmte Platine aus der Transportbox entnimmt und auf die Frontschale montiert,
 - e. das montierte Werkstück auf den Palettenträger zurücklegt
 - f. und den vorhandenen Greifer in die Wechselvorrichtung zurücklagert.

3. Führen Sie eine Inbetriebnahme des in (2) erstellten Programms für die reale RoboterMontagestation durch.

■ Arbeitshilfen

- Mitsubishi Controller CR750/CR751 – Instruction Manual – Detailed Explanation of Functions and Operations
- CIROS® Hilfe -> Programmierung -> Melfa Basic V
- Kapitel 5 in den Grundlagen
- Technische Dokumentation CP_Factory RoboterMontagezelle, Kapitel 2.

Hinweis:

Am Flansch des Roboters befindet sich die Vorrichtung zur Übernahme eines Greifersystems aus der Wechselvorrichtung. Sie ist verbunden mit einer pneumatischen Steuerung, mit der sowohl das gewünschte Greifersystem aufgenommen und abgegeben, als auch der Greifer geöffnet und geschlossen wird. Hierzu gibt es vorbereitete Programme, die in Ihr Projekt übernommen werden können:

- „GRPRelease“: Greifer aus der Wechselvorrichtung entnehmen
- „GRPLock“: Greifer an die Wechselvorrichtung übergeben
- „GRPOpen“: Greifer öffnen
- „GRPClose“: Greifer schließen.
- „GRPVacOff“: Vakuum am Greifer abschalten
- „GRPVacOn“: Vakuum am Greifer öffnen

Über die SPS können externe Informationen eingelesen werden.

1. Erstellen Sie in der Simulation ein Testprogramm, so dass der Roboter prüft, ob er einen vorgegebenen Greifer angeschlossen hat und im negativen Fall den vorhandenen Greifer zurücklagert und den korrekten Greifer anschließt.

- a) Öffnen Sie die Arbeitszelle **CP_RobotAssembly** im Verzeichnis **CIROS_CPFactory /Rob_Montage /RV-4FL**. Öffnen Sie das Projekt **A4** in der Projektverwaltung. Es zeigt die Dateien **PickGrp.mb5** und **PickGrp.pos** an. Vervollständigen Sie das Programm, so dass der Roboter je nach Parametervorgabe einen bestimmten Greifertyp anschließt und mit geöffneten Greifer in der Startposition anhält. Falls die Parametereingabe nicht im zulässigen Bereich liegt, dann soll das Programm sofort beendet werden.

Hinweis:

- Zur Generierung eines Greifertyps legen Sie folgende Zuordnung fest:
 - 1 <-> Vakuumgreifer
 - 2 <-> Werkzeuggreifer
 - 3 <-> Sicherungsgreifer
 - 4 <-> Kein Greifer

- Das Programm enthält alle Deklarationen und Parametereinträge. Die Positionsliste enthält schon alle notwendigen Einträge und das Projekt hat auch schon folgende externen Programme geladen: **GRPLock, GrpRelease, GRPOpen, GRPVacOff**

Lösung:

- Öffnen Sie das Projekt **A4** der Arbeitszelle **CP_RobotAssembly** im Verzeichnis **CIROS_CPFactory /Rob_Montage /RV-4FL**. Ordnen Sie dem Projekt A4 die Steuerung **Montage_RV-4FL** zu.
- Testen Sie die Positionen **PGrpStorageVac, PGrpStorageWP, PGrpStorageFuse**:
 - Prüfen Sie, dass im **Teach_In** Fenster **TCP_None** als **TCP** eingestellt ist.
 - Mit einem Doppelklick auf die Positionen, soll der Roboter in die gewünschten Positionen springen.
- Programmbeispiel:

```
[Montage_RV-4FL] \Programs\PickGrp.mb5 *
'Aufgabe 4 / Auf Anforderung einen Greifer anschließen
' Externe Programme:
' "GRPLock" - Greifer an der Wechsellvorrichtung übernehmen
' "GRPOpen"
' "GRPVacOff"
' -----
' Positionen
DEF POS PtGripperNone ' Position für TCP am Flansch
DEF POS AUX           ' Hilfsposition
DEF POS PAUX
' -----
DEF INTE ToolNo       ' Greifertyp
DEF INTE fast
DEF INTE slow
' -----
' Parameter setzen
PtGripperNone = (0,0,0,0,0,33.50)
slow = 20
fast = 150
' -----
' Programmstart
TOOL PtGripperNone ' TCP setzen
JOVRD 30
SPD fast
ToolNo = 3          ' Parametewert = 1,2,3,4
'
' Roboter in Startposition fahren
MOV PHome
'
' Greiferauswahl
IF ToolNo = 1 THEN
AUX = PGrpStorageVac
ENDIF
IF ToolNo = 2 THEN
AUX = PGrpStorageWp
ENDIF
IF ToolNo = 3 THEN
AUX = PGrpStorageFuse
ENDIF
IF (ToolNo < 1 OR ToolNo > 3) THEN GOTO *ende
'
```

```

[Montage_RV-4FL] \Programs\PickGrp.mb5
' Ausgewählten Greifer anschließen
PAUX = AUX + PToolXZ180
MOV PAUX
MVS AUX, -50
SPD slow
PREC ON
MVS AUX
DLY 0.5
CALLP "GRPlock"
'
' Greifer aus Halterung ziehen
PAUX = AUX + PToolX100
MVS PAUX
PREC OFF
SPD fast
PAUX = AUX + PToolX180
MVS PAUX
MOV PHome
'
' Greifer öffnen
IF ToolNo = 1 THEN
CALLP "GRPVacOff"
ENDIF
IF ToolNo > 1 THEN
CALLP "GRPOpen"
ENDIF
*ende
END
(neu)

```

- Testergebnis: Wählen Sie verschiedene Eingaben und prüfen Sie, ob stets der richtige Greifer gewählt wird. Prüfen Sie, dass der Greifer einwandfrei aus der Halterung entnommen wird.
- Wird für ToolNo ein Wert < 1 oder ≥ 4 eingegeben, dann wird das Programm beendet.

b) Erstellen Sie für das Projekt A4 ein neues Hauptprogramm **A4.mb5**, das **PickGrp** als externes Programm aufruft und an **PickGrp** den Greifertyp als Parameter übergibt.

Hinweis: Zur Übergabe definieren Sie in **A4** für den Parameter eine globale Variable, z.B.

M_GripperType, die Sie dann in **PickGrp** ohne weitere Deklaration verwenden können.

Lösung:

• Programmbeispiel für A4.mb5:

```

' Aufgabe 4
' Externe Programme:
' "PickGrp" - Einen Greifer auswählen und anschließen
' "GRPlock" - Greifer an der Wechsellvorrichtung anschließen
' "GRPOpen"
' "GRPVacOff"
' .....

```

```

DEF INTE M_GripperType      ' Greifertyp ( 1 - 4)
' .....

```

```

' Programmstart
M_GripperType = 3
CALLP "PickGrp"
END

```

• Änderung in PickGrp.mb5:

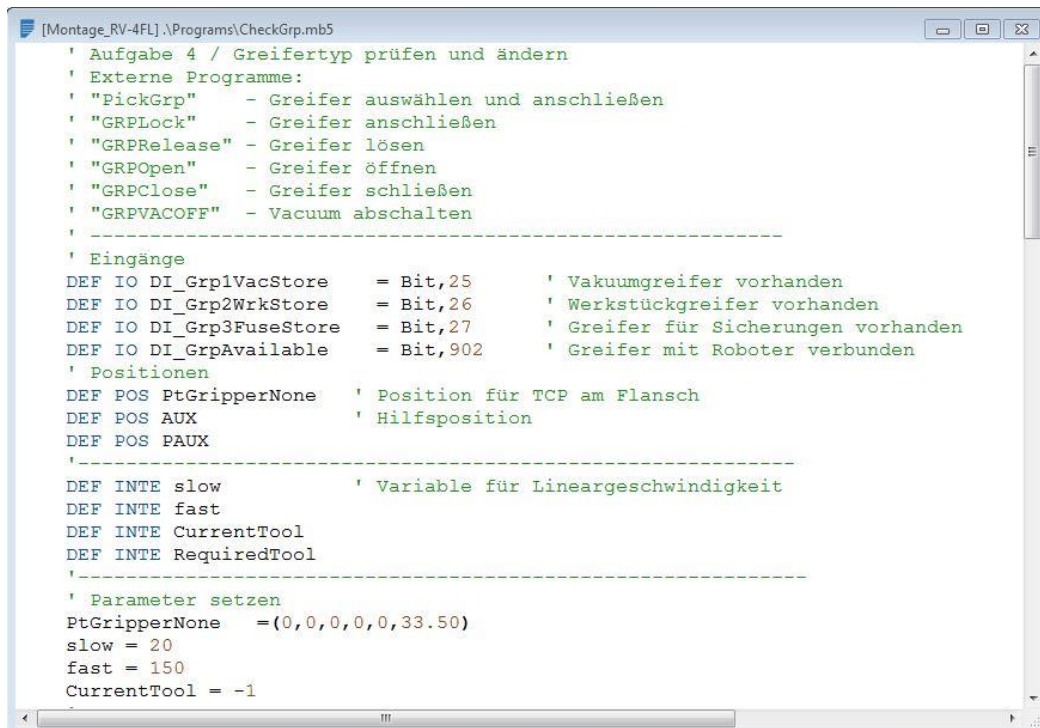
Ersetzen Sie die Zeile **ToolNo = 1,2,3,4** durch
ToolNo = M_GripperType.

- Entfernen Sie die Datei **PickGrp.mb5** aus dem Projekt A4 und fügen **A4.mb5** als neues Hauptprogramm hinzu. Danach fügen Sie das geänderte Programm **PickGrp.mb5** wieder dem Projekt hinzu.
 - Kompilieren Sie das Projekt und testen Sie es für die verschiedenen Werte.
- c) Erweitern Sie Ihr Projekt durch ein externes Programm **CheckGrp.mb5**, dass
- den angeschlossenen Greifertyp überprüft
 - den Greifer zurücklagert, wenn er nicht dem angeforderten Typ entspricht,
 - den angeforderten Greifertyp anschließt.

Hinweis: Die Greiferwechselvorrichtung hat an jeder Übergabeposition einen Sensor, der prüft, ob der Greifer vorhanden ist.

Lösung:

- Überlegen Sie sich zunächst, dass die Positionsliste von PickGrp übernommen werden kann.
- Erstellen Sie für das Programm die notwendigen Deklarationen von Eingängen und Parametern



```

[Montage_RV-4FL] \Programs\CheckGrp.mb5

' Aufgabe 4 / Greifertyp prüfen und ändern
' Externe Programme:
' "PickGrp" - Greifer auswählen und anschließen
' "GRPLock" - Greifer anschließen
' "GRPRelease" - Greifer lösen
' "GRPOpen" - Greifer öffnen
' "GRPClose" - Greifer schließen
' "GRPVACOFF" - Vacuum abschalten
' -----
' Eingänge
DEF IO DI_Grp1VacStore = Bit,25 ' Vakuumgreifer vorhanden
DEF IO DI_Grp2WrkStore = Bit,26 ' Werkstückgreifer vorhanden
DEF IO DI_Grp3FuseStore = Bit,27 ' Greifer für Sicherungen vorhanden
DEF IO DI_GrpAvailable = Bit,902 ' Greifer mit Roboter verbunden
' Positionen
DEF POS PtGripperNone ' Position für TCP am Flansch
DEF POS AUX ' Hilfsposition
DEF POS PAUX
' -----
' Variable für Lineargeschwindigkeit
DEF INTE slow
DEF INTE fast
DEF INTE CurrentTool
DEF INTE RequiredTool
' -----
' Parameter setzen
PtGripperNone = (0,0,0,0,0,33.50)
slow = 20
fast = 150
CurrentTool = -1
  
```

- Prüfen Sie zunächst, welcher Greifertyp angeschlossen ist und entscheiden dann, ob der Greifer zurücklagert werden muss:

```

[Montage_RV-4FL] \Programs\CheckGrp.mb5 *
' Programmstart
TOOL FtGripperNone      ' TCP setzen
JOVRD 30
SPD fast
'
' Roboter in Startposition fahren
MOV PHome
RequiredTool = M_GripperType
' Prüfen ob der Roboter einen Greifer angeschlossen hat
'
' Fall 1 = Kein Greifer angeschlossen
If ((DI_GrpAvailable = 0) And (DI_Grp1VacStore = 1) And (DI_Grp2WrkStore = 1) And (DI_Grp3FuseStore = 1)) Then
    CurrentTool = 4
EndIf
' Fall 2 = Vacuum Greifer ist angeschlossen
If ((DI_GrpAvailable = 1) And (DI_Grp1VacStore = 0) And (DI_Grp2WrkStore = 1) And (DI_Grp3FuseStore = 1)) Then
    CurrentTool = 1
EndIf
' Fall 3 = Werkzeuggreifer ist angeschlossen
If ((DI_GrpAvailable = 1) And (DI_Grp1VacStore = 1) And (DI_Grp2WrkStore = 0) And (DI_Grp3FuseStore = 1)) Then
    CurrentTool = 2
EndIf
' Fall 4 = Sicherungsgreifer ist angeschlossen
If ((DI_GrpAvailable = 1) And (DI_Grp1VacStore = 1) And (DI_Grp2WrkStore = 1) And (DI_Grp3FuseStore = 0)) Then
    CurrentTool = 3
EndIf
' Falls keiner der Fälle vorliegt, dann liegt ein Fehler vor
' und das Programm wird beendet.
If CurrentTool < 0 Then GOTO *ende
'

```

- Versuchen Sie wie in **PickGrp** mit den Hilfsvariablen PAUX und AUX den Zurücklagerungsprozess für alle Fälle zusammenzufassen:

```

[Montage_RV-4FL] \Programs\CheckGrp.mb5 *
'
' Greifer zurücklagern und korrekten Greifertyp anschließen
IF CurrentTool = 4 THEN
CALLP "PickGrp"
GOTO *ende
ENDIF
IF CurrentTool = 3 THEN
AUX = PGrpStorageFuse
ENDIF
IF CurrentTool = 2 THEN
AUX = PGrpStorageWp
ENDIF
IF CurrentTool = 1 THEN
AUX = PGrpStorageVac
ENDIF
'
' Greifer zurücklagern
PAUX = AUX + PToolX180
MOV PAUX
PAUX = PAUX - PToolX100
MVS PAUX
IF CurrentTool = 3 OR CurrentTool = 2 THEN
CALLP "GrpClose"
ENDIF
SPD slow
PREC ON
MVS AUX
DLY 0.5
CALLP "GRPrelease"
MVS AUX, -50
SPD fast
'
' Korrekten Greifer anschließen
CALLP "PickGrp"
'
*ende
END

```

- d) Ergänzen Sie Ihr Programm A4, so dass Sie das geforderte vollständige Testprogramm erhalten.

Lösung:

Ergänzen Sie Ihr Programm A4 wie folgt:

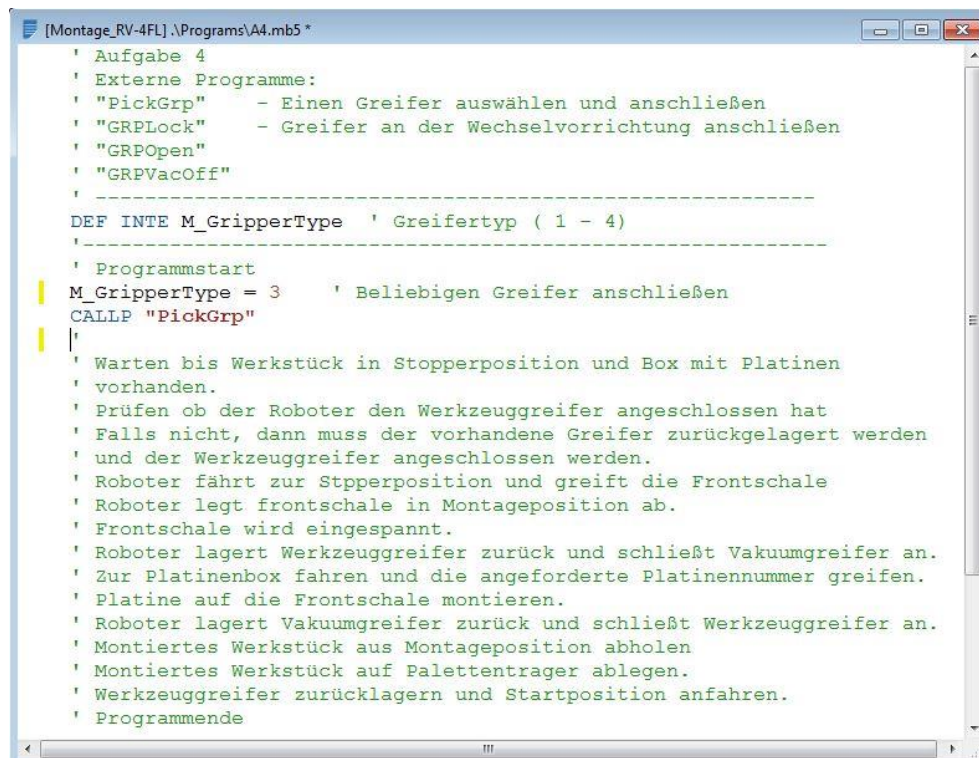
```
' Programmstart
M_GripperType = 3    ' Beliebigen Greifer anschließen
CALLP "PickGrp"
M_GripperType = 1    ' Angeforderter Greifertyp
CALLP "CheckGrp"
END
```

2. Erstellen Sie in der Simulation ein Programm, das

- einen beliebigen Greiferstatus für den Roboter generiert,
- überprüft, ob der Werkzeuggreifer angeschlossen ist und in negativen Fall den Greifer wechselt,
- die Frontschale in die Montageposition einspannt
- auf Anforderung eine bestimmte Platine aus der Transportbox entnimmt und auf die Frontschale montiert,
- das montierte Werkstück auf den Palettenträger zurücklegt
- und den vorhandenen Greifer in die Wechselvorrichtung zurücklagert.

- a) Öffnen sie die Anzeige des Programms **A4** und erstellen Sie im Programmfenster zunächst mit Hilfe von Kommentarzeilen einen Ablaufplan.

Lösung:



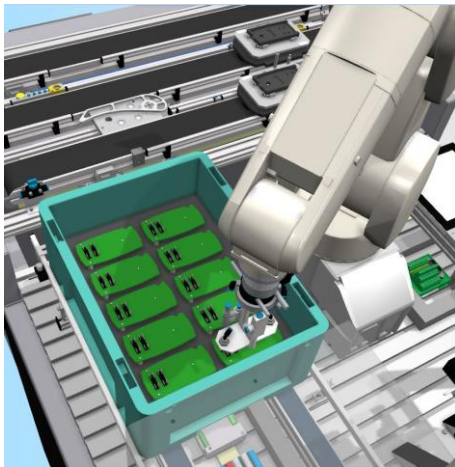
```
[Montage_RV-4FL] \Programs\A4.mb5 *
' Aufgabe 4
' Externe Programme:
' "PickGrp"      - Einen Greifer auswählen und anschließen
' "GRPLock"     - Greifer an der Wechselvorrichtung anschließen
' "GRPOpen"
' "GRPVacOff"
' -----
DEF INTE M_GripperType  ' Greifertyp ( 1 - 4)
' -----
' Programmstart
M_GripperType = 3    ' Beliebigen Greifer anschließen
CALLP "PickGrp"
|
' Warten bis Werkstück in Stopperposition und Box mit Platinen
' vorhanden.
' Prüfen ob der Roboter den Werkzeuggreifer angeschlossen hat
' Falls nicht, dann muss der vorhandene Greifer zurückgelagert werden
' und der Werkzeuggreifer angeschlossen werden.
' Roboter fährt zur Stpperposition und greift die Frontschale
' Roboter legt frontschale in Montageposition ab.
' Frontschale wird eingespannt.
' Roboter lagert Werkzeuggreifer zurück und schließt Vakuumgreifer an.
' Zur Platinenbox fahren und die angeforderte Platinennummer greifen.
' Platine auf die Frontschale montieren.
' Roboter lagert Vakuumgreifer zurück und schließt Werkzeuggreifer an.
' Montiertes Werkstück aus Montageposition abholen
' Montiertes Werkstück auf Palettenträger ablegen.
' Werkzeuggreifer zurücklagern und Startposition anfahren.
' Programmende
```

- b) Prüfen Sie, dass die Positionsliste **A4.pos** aus den Positionen von **A3.pos**, **PickGrip.pos**, den vier Palettenpositionen (Pal1,...,Pal4) und der Ablageposition einer Platine in die Montageposition (PCBAssembly) besteht, die schon in der Liste vorgegeben ist. Die Palettenpositionen müssen Sie noch teachen.

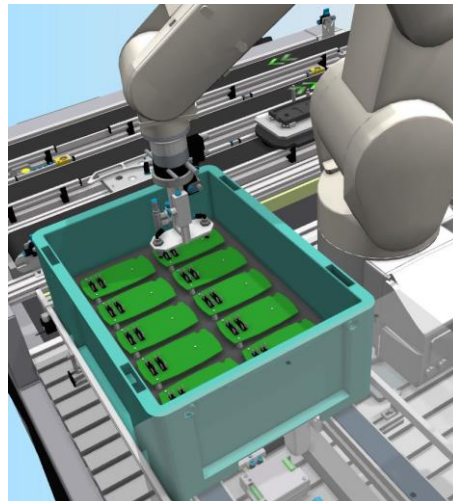
Lösung:

- Starten Sie hierzu das Programm A4 mit der Option, dass der Vakuumgreifer angeschlossen wird.
- Öffnen Sie das Teach-In Fenster und wählen **TCP_Vac** als TCP im Modus **Kartesische Koordinaten**. Bewegen Sie den Roboter zur Position **Pal1** (siehe Grafik). Der vordere Saugnapf muss sehr eng an der ersten Sicherungsvorrichtung sitzen. Zum Testen der Positionen setzen Sie im Teach-In Fenster die Greiferausgänge **HOPEN_1**, **HCLOSE_1**, **HOPEN_2**, **HCLOSE_2**, **HOPEN_3** und **HCLOSE_3** entsprechend wie in 3.1 der Einleitung, um das Vakuum einzuschalten. Bewegen Sie den Roboter in positive z-Richtung. Falls die Platine sich mitbewegt, dann ist die Position korrekt eingelesen.

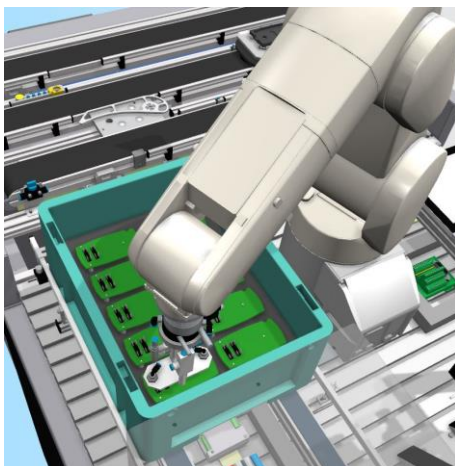
Position Pal1:



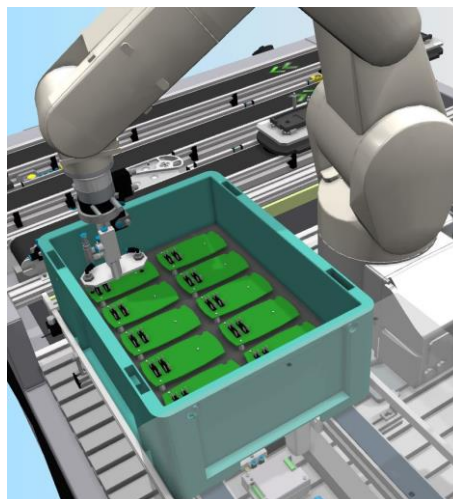
Position Pal2



Position Pal3



Position Pal4



- c) Erstellen Sie zunächst den Deklarationsteil für die benötigten Ein- und Ausgänge und die Parameter. Überlegen Sie sich insbesondere die Definition der Palette für die Aufnahme der Platinen.

Lösung:

```
[Montage_RV-4FL] \Programs\A4.mb5

' Aufgabe 4
' Externe Programme:
' "PickGrp" - einen Greifer auswählen und anschließen
' "CheckGrp" - Greiferstatus prüfen und gegebenenfalls austauschen
' "GRPLock" - Greifer an der Wechselvorrichtung übernehmen
' "GRPRelease" - Greifer an die Wechselvorrichtung übergeben
' "GRPClose" - Greifer öffnen
' "GRPClose" - Greifer schließen
' "GRPVACOFF" - Vacuum abschalten
' "GRPVACON" - Vacuum einschalten

' -----
' Eingänge
DEF IO DI_PalletInFront = Bit,244 ' Signal von der SPS
DEF IO DI_WPAAtStopAvail = Bit,251 ' Signal von der SPS
DEF IO DI_WPAvail = Bit,17 ' Werkstück in Montagestation
DEF IO DI_WPOrientOk = Bit,18 ' Orientierung des Werkstücks ist OK
DEF IO DI_WPHoleOk = Bit,19 ' Bohrung ist an der richtigen Position
DEF IO DI_Grp1VacStore = Bit,25 ' Vakuumgreifer vorhanden
DEF IO DI_Grp2WrkStore = Bit,26 ' Werkstückgreifer vorhanden
DEF IO DI_Grp3FuseStore = Bit,27 ' Greifer für Sicherungen vorhanden
DEF IO DI_GrpAvailable = Bit,902 ' Greifer mit Roboter verbunden
DEF IO DI_PCBPalletNo = Byte,172 ' Palettensnummer
'
' Ausgänge
DEF IO DO_WPMountLock = Bit,16 ' Werkstück einspannen
'
' Positionen
DEF POS PtGripperNone ' Position für TCP am Flansch
DEF POS PtGripperWp ' Position für TCP Werkzeuggreifer
DEF POS PtGripperVac ' Position für TCP im Zentrum des
' Vacuumgreifers
DEF POS AUX ' Hilfsposition
' -----
DEF INTE slow ' Variable für Lineargeschwindigkeit
DEF INTE fast
DEF INTE M_GripperType ' Greifertyp
DEF INTE PalletNo
DEF INTE CurrentTool
' -----
DEF PLT 1, Pal1,Pal2,Pal3,Pal4,5,2,1 ' Platinenpalette
' -----
' Parameter setzen
PtGripperVac = (0,0,205,0,0,33.50)
PtGripperWp = (0,0,170,0,0,33.50)
PtGripperNone = (0,0,0,0,0,33.50)
JOVRD 30
slow = 20
fast = 150
CurrentTool = -1
' -----
```

- d) Entwickeln Sie schrittweise für Ihre Ablaufschritte Programmlösungen und testen Sie diese.

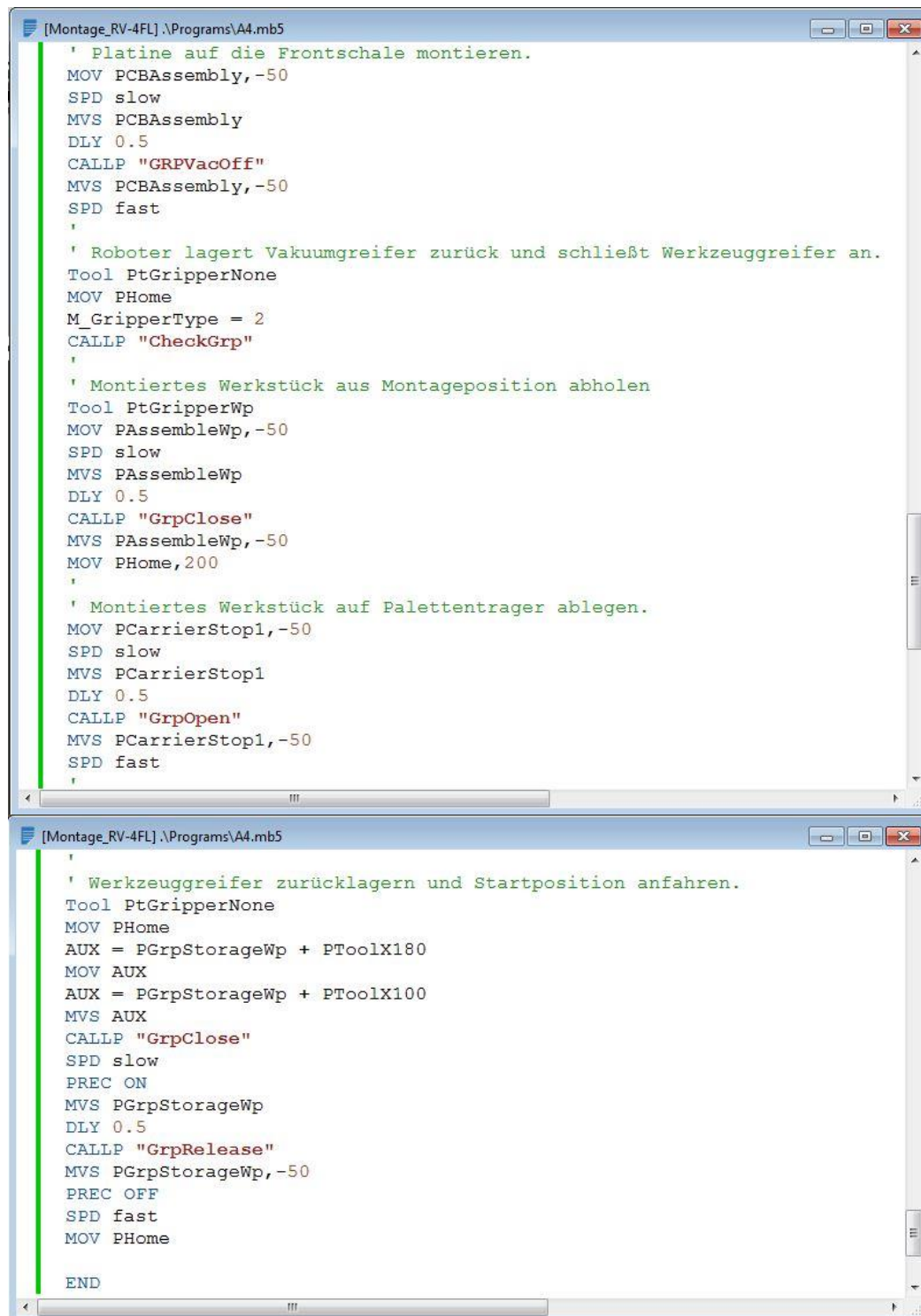
Lösung:

```

[Montage_RV-4FL] \Programs\A4.mb5 *
' Programmstart
PalletNo = DI_PCBPalletNo
M_GripperType = 4      ' Beliebigen Greifer anschließen
CALLP "PickGrp"
'
' Warten bis Werkstück und Box mit Platinen in Stopperposition
WHILE (DI_PalletInFront = 0) OR (DI_WPAAtStopAvail = 0)
WEND
'
' Prüfen ob der Roboter den Werkzeuggreifer angeschlossen hat
' Falls nicht, dann muss der vorhandene Greifer zurückgelagert werden
' und der Werkzeuggreifer angeschlossen werden.
M_GripperType = 2
CALLP "CheckGrp"
'
' Roboter fährt zur Stopperposition und greift die Frontschale
TOOL PtGripperWp
MOV PCarrierStop1,-50
SPD slow
MVS PCarrierStop1
DLY 0.5
CALLP "GRPClose"
MVS PCarrierStop1,-50
'

[Montage_RV-4FL] \Programs\A4.mb5 *
' Roboter legt frontschale in Montageposition ab.
MOV PAssembleWP,-100
MVS PAssembleWP,-30
SPD slow
MVS PAssembleWP
DLY 0.5
CALLP "GRPOpen"
SPD fast
MVS PAssembleWP,-100
' Frontschale wird eingespannt.
WHILE ((DI_WPAvail = 0) OR (DI_WPOrientOk = 0) OR (DI_WPHoleOK = 0))
WEND
DO_WPMountLock = 1
Tool PtGripperNone
MOV PHome
' Roboter lagert Werkzeuggreifer zurück und schließt Vakuumgreifer an.
M_GripperType = 1
CALLP "CheckGrp"
' Zur Platinenbox fahren und die angeforderte Platinennummer greifen.
TOOL PtGripperVac
AUX = Plt 1,PalletNo
MOV AUX,-100
SPD slow
MVS AUX
DLY 0.5
CALLP "GRPVAcon"
DLY 0.5
SPD fast
MVS AUX,-100
MOV PHome
DLY 0.5
'

```



```

[Montage_RV-4FL] \Programs\A4.mb5
' Platine auf die Frontschale montieren.
MOV PCBAssembly,-50
SPD slow
MVS PCBAssembly
DLY 0.5
CALLP "GRPVacOff"
MVS PCBAssembly,-50
SPD fast
'
' Roboter lagert Vakuumgreifer zurück und schließt Werkzeuggreifer an.
Tool PtGripperNone
MOV PHome
M_GripperType = 2
CALLP "CheckGrp"
'
' Montiertes Werkstück aus Montageposition abholen
Tool PtGripperWp
MOV PAssembleWp,-50
SPD slow
MVS PAssembleWp
DLY 0.5
CALLP "GrpClose"
MVS PAssembleWp,-50
MOV PHome,200
'
' Montiertes Werkstück auf Palettenträger ablegen.
MOV PCarrierStop1,-50
SPD slow
MVS PCarrierStop1
DLY 0.5
CALLP "GrpOpen"
MVS PCarrierStop1,-50
SPD fast
'

[Montage_RV-4FL] \Programs\A4.mb5
'
' Werkzeuggreifer zurücklagern und Startposition anfahren.
Tool PtGripperNone
MOV PHome
AUX = PGrpStorageWp + PToolX180
MOV AUX
AUX = PGrpStorageWp + PToolX100
MVS AUX
CALLP "GrpClose"
SPD slow
PREC ON
MVS PGrpStorageWp
DLY 0.5
CALLP "GrpRelease"
MVS PGrpStorageWp,-50
PREC OFF
SPD fast
MOV PHome
END

```

- **Testergebnis:**
Zur Änderung der Eingabe der Palettennummer öffnen Sie den Modell-Explorer und öffnen Sie die Unterkomponenten der Roboterkomponente **Montage_RV-4FL**. Wählen Sie die Eingänge zu **AD_Converter_8Bit_PCBPalletNo**. Es wird nur ein Eingang **AnalogIn** angezeigt. Markieren Sie den Eingang und öffnen Sie das Kontextmenü. Wählen Sie die Option **Bearbeiten -> Wert erzwingen** und tragen dann einen beliebigen Wert zwischen 1 und 10 ein.
- Überprüfen Sie für alle Werte, dass die richtige Platine gegriffen wird.

- e) Datensicherung: Speichern Sie Programme und Positionslisten unter den Namen **A4R**, **PickGrpR** und **CheckGrpR**.

3. Führen Sie reale Inbetriebnahme Ihres Projektes A4 durch.

- a) Die Arbeitszelle **CP_RobotAssembly** ist geöffnet und der PC ist mit der Robotersteuerung verbunden. Führen Sie einen Download Ihrer Programme **A4R**, **PickGrpR** und **CheckGrpR** und der zugehörigen Positionslisten in die Robotersteuerung durch. Prüfen Sie, welche Positionen sie aus A3R.pos übernehmen können und welche Programmänderungen Sie durchführen müssen.

Lösung:

- Es wird vorausgesetzt, dass Sie die Aufgabe 3 schon erfolgreich bearbeitet haben und dass Sie mittels des Einrichtemodus am Bedienpanel einen Palettenträger mit Werkstück an die Stopper-Position im Bypass und die mit Platinen beladene Transportbox an die Stopposition gefahren haben.
- Übernehmen Sie die Positionen **PHome**, **PCarrierStop1**, **PGrpStorageWP** und **PAssembleWp** aus **A3R.pos** in die Positionslisten **A4R.pos**, **PickGrpR.pos** und **CheckGrpR.pos**
- Ändern Sie in den Programmen **A4R**, **PickGrpR** und **CheckGrpR** die Aufrufe der externen Programme **PickGrpR** und **CheckGrpR**.
- Führen Sie einen Download Ihrer Programme **A4R.mb5**, **PickGrpR.mb5**, **CheckGrpR.mb5** und der entsprechenden Positionslisten durch. Sie erhalten jeweils eine Meldung, ob der Download erfolgreich durchgeführt werden konnte.
- Wählen Sie die Gruppe **Programme** im **RCI-Explorer**. Im rechten Fenster werden die Programme **A4R**, **PickGrpR** und **CheckGrpR** angezeigt.

Hinweis:

Die externen Programme **GRPLock**, **GRPRelease**, **GRPOpen**, **GRPClose**, **GRPVacOff** und **GRPVacOn** sind schon in der Steuerung und müssen nicht mehr geladen werden.

- b) Überprüfen Sie die Verdrahtung der Eingangssignale.

Lösung:

- Es muss die Verdrahtung der Eingänge **DI_Grp1VacStore (Bit 25)**, **DI_Grp2WrkStore (Bit26)**, **DI_Grp3FuseStore (Bit27)**, **DI_GrpAvailable (Bit 902)** und **DI_PalletinFront (Bit 2144)** geprüft werden. Es ist zu beachten, dass der Sensor **DI_WPAtStopAvail** schon in Aufgabe 3 geprüft worden ist.
- Ist das Signal vom Sensor **DI_Grp1VacStore** mit dem Eingangsbit 25 der realen Robotersteuerung verdrahtet?
Öffnen Sie Anzeige **Monitore** im **RCI Explorer** und wählen Sie die Anzeige **Eingänge** und klicken auf **Importieren**. Als Startadresse wählen Sie 16 und klicken auf **Online**. Prüfen Sie die Anzeige an Bit-Adresse 25.
- Verfahren Sie analog für die Sensoren **DI_Grp2WrkStore** und **DI_Grp3FuseStore**.

- Zur Prüfung von **DI_GrpAvailable** wählen Sie die Startadresse 900.
- Zur Prüfung von **DI_PalletinFront** wählen Sie die Startadresse 2140. Prüfen Sie, dass die Kommunikation mit der SPS aktiv ist.
- Die Palettennummer **DI_PCBPalletNo** können Sie durch Angabe des **KistenNr.**, siehe **1.5.3** in Einleitung festlegen.
- Es müssen im Deklarationsteil der Programme **A4R**, **A41R** und **PickGrpR** folgende Änderungen vorgenommen werden:

```

DEF IO DI_WPAAtStopAvail    =    Bit, 2151
DEF IO DI_PalletinFront     =    Bit, 2144
DEF IO DI_PCBPalletNo       =    Bit, 2072

```

- c) Programm und Positionsliste wurden in die Robotersteuerung geladen. Es müssen nur die Positionen in Positionsliste **A4R.pos** geprüft werden. Die Positionen in den Positionslisten **PickGrpR.pos** und **CheckGrp.pos** können aus **A4R.pos** übernommen werden.

Hinweis:

Sie finden eine detaillierte Beschreibung zur Handhabung des PGH zum Einrichten von Positionen in Kapitel 3 des Mitsubishi Handbuchs **Mitsubishi Controller CR750/CR751 – Instruction Manual – Detailed Explanation of Functions and Operations**.

Zur Positionsprüfung müssen Sie den TCP ändern, siehe **§4** in der **Einleitung**.



Lösung:

- Wählen Sie den Modus **MANUAL** an der Robotersteuerung und aktivieren Sie das PGH.
- Öffnen Sie das Menü **1. FILE/EDIT** im Hauptmenü Fenster. Es werden alle Programme angezeigt.
- Wählen Sie Ihr Programm **A4R** und drücken Sie die Taste **F2 [POSI.]**. Es werden die Positionen der Positionsliste angezeigt. Mittels der Tasten **[NEXT]** und **[PREV]** können Sie zwischen den Positionen wechseln.
- Prüfen Sie, dass der TCP im Flansch des Roboters eingestellt ist.
- Position **PGrpStorageFuse**:
 - Übernehmen Sie die Position **P_GrpStorageFuse** aus der universellen Positionsliste **UBP.pos**, siehe **§6** in der **Einleitung**.
 - Wählen Sie im PGH die Anzeige der Position **PGrpStorageFuse** zu Programm **A4R**.
 - Wählen Sie den **JOG-Modus** und bewegen den Roboter so, dass die Anschlussvorrichtung vom Roboter oberhalb der Aufnahmeposition positioniert ist.
 - Wählen Sie den **XYZ-Modus** und reduzieren Sie den **OVRD auf 10%**.
 - Achten Sie darauf, dass die Anschlussvorrichtung nicht verriegelt ist, d.h. die Stifte im Anschluss sind nicht ausgefahren. Andernfalls wählen Sie den **HAND-Modus** und drücken **(+X)**.
 - Schalten Sie wieder in die Anzeige der Position **PGrpStorageFuse** und bewegen den Roboter mit **MOVE** in die Zielposition. Im Ausnahmefall müssen Sie kleine Korrekturen vornehmen.
 - Wählen Sie den **HAND-Modus** und schließen Sie den Greifer mittels der Handtaste **(-X)** an den Roboter, siehe **3.5** der Einleitung.
 - Wechseln Sie zurück in den **JOG-Modus** und bewegen den Roboter ca. 60 mm in **+X**-Richtung. Der Test ist positiv, wenn der Greifer sich nahezu geräuschlos aus der

- Halterung löst.
- Wechseln Sie wieder zurück in die Positionsanzeige und bewegen den Roboter im XYZ-Modus mit der **MOVE**-Taste zurück in die Zielposition und lösen den Greifer.
- Wechseln Sie in den JOG-Modus und bewegen den Roboter im XYZ-Modus ca. 50 mm in **+Z**-Richtung und dann ca. 100mm in **+X**- Richtung.
- Wechseln Sie zurück in die Positionslistenanzeige und wählen die Position **PHome**. Bewegen Sie den Roboter mit **MOVE** zurück in die Startposition.
- Position **PGrpStorageVac**:
 - Übernehmen Sie die Position **P_GrpStorageVac** aus der universellen Positionsliste **UBP.pos**.
 - Zum Testen der Position aktivieren Sie das **PGH** und wählen die Anzeige der Position **PGrpStorageVac** zu Programm **A4R**.
 - Wählen Sie den **JOG-Modus XYZ** und bewegen Sie den Roboter senkrecht oberhalb der Zielposition.
 - Verfahren Sie dann ähnlich wie bei der Position **PGrpStorageFuse** .
- Positionen **Pal1 - Pal4**:
 - Der Roboter hat den Vakuumgreifer angeschlossen.
 - Setzen Sie den **TCP = TCP_Vac**.
 - Übernehmen Sie für **Pal1** den Positionswert **P_PCBPaletXDir**, für **Pal2** den Positionswert **P_PCBPaletOrigin**, für **Pal3** den Positionswert **P_PCBPaletXYDir** und für **Pal4** den Positionswert **P_PCBPaletYDir** aus der universellen Positionsliste **UBP.pos**.
 - Zum Testen der aktualisierten Position **Pal1** aktivieren Sie das **PGH** und wählen die Anzeige der Position **Pal1** zu Programm **A4R**.
 - Wählen Sie den **JOG-Modus XYZ** und bewegen den Roboter ungefähr senkrecht oberhalb der Zielposition.
 - Reduzieren Sie den **OVRD auf 30%** und bewegen Sie den Roboter mit **MOVE** in die Position **Pal1**.
 - Wählen Sie den **HAND-Modus** und schalten Sie mit den Tasten **(-B) + (+C)** das Vakuum ein, siehe **3.1** der Einleitung.
 - Wechseln Sie zurück in den **JOG-Modus** und bewegen den Roboter ca. 60 mm in **+Z**-Richtung. Der Test ist positiv, wenn die Platine aus der Box angehoben wird.
 - Zum Test der weiteren Positionen bewegen Sie den Roboter mit der **MOVE**-Taste wieder zurück in **Pal1**.
 - Wiederholen Sie den Prozess für die Positionen **Pal2, Pal3** und **Pal4**.
- Position **PCBAssembly**:
 - Der Roboter hat den Vakuumgreifer angeschlossen.
 - Setzen Sie den **TCP = TCP_Vac**.
 - Übernehmen Sie für **PCBAssembly** den Positionswert **P_AssemblePCB** aus der universellen Positionsliste **UBP.Pos**.
 - Zum Testen der aktualisierten Position **PCBAssembly** aktivieren Sie das **PGH** und wählen zunächst die Anzeige der Position **Pal1** zu Programm **A4R**.
 - Bewegen Sie den Roboter mit der **MOVE** Taste zur Position **Pal1** und saugen eine Platine mit dem Vakuumgreifer an.
 - Wählen Sie jetzt im PGH die Anzeige der Position **PCBAssembly**.

- Schalten Sie in den **JOG** Modus und bewegen den Roboter schrittweise ungefähr senkrecht oberhalb der Position **PCBAssembly**.
 - Reduzieren Sie den **OVRD** auf 30% und bewegen den Roboter mit der **MOVE** Taste zur Zielposition **PCBAssembly**.
 - Die Position ist korrekt, wenn die Platine nach Abschalten des Vakuums einwandfrei auf dem Werkstück aufliegt.
 - Zum Ein-und Ausschalten des Vakuums mit dem PGH beachten Sie §3.1 in der Einleitung.
- Übernehmen Sie die Positionen aus **A4R.pos** in die Positionslisten **PickGrpR.pos** und **CheckGrpR.pos**.
- d) Programm und Positionsliste sind in der Robotersteuerung aktualisiert worden. Testen Sie mittels des RCI-Explorers den Programmablauf und optimieren Sie die Geschwindigkeit. Dokumentieren Sie Ihre Ergebnisse für eventuelle Änderungen im Programmablauf.

Lösung:

- Wählen Sie zum Test Ihres Programms im RCI-Explorer zunächst den **Debug**-Modus. In der Online-Anzeige Ihres Programms sind die Programmzeilen nummeriert und die Kommentare sind gelöscht. Mit  können Sie Breakpoints in den entsprechenden Programmzeilen Ihres Programms setzen.
 - Das Programm wird jeweils vor Ausführung der entsprechenden Programmzeile gestoppt.
 - Mit dem Befehl  können Sie Ihr Programm mit Breakpoint-Überwachung starten.
 - Achten Sie darauf, dass der Wert von **DI_PCBalletNo < 11** sein muss.
- e) Testen Sie Ihr Programm im Automatikmodus

Lösung:

- Starten Sie Ihr Programm über das Bedienfeld der Robotersteuerung.
 - Wählen Sie das Programm **A4R** aus.
 - Die **OVRD** Auswahl bewirkt eine prozentuale Reduzierung der im Programm definierten maximalen Geschwindigkeit.
 - Schalten Sie die Servo Motoren ein und starten Sie das Programm.
- f) Datensicherung: Führen Sie einen Upload des aktuellen Roboterprogramms und der aktuellen Positionsliste in der Robotersteuerung auf den PC durch.

Lösung:

Die Arbeitszelle **CP_RobotAssembly** ist geöffnet und der PC ist mit der Robotersteuerung verbunden. Öffnen und aktivieren Sie das Programmfenster mit dem Programm **A4R.mb5**. Wählen Sie das Menü **EXTRAS -> Online-Kopplung -> Upload realer Roboter -> PC**. Das Programm wird auf den PC geladen und überschreibt das vorhandene Programm **A4R**. Führen Sie dies ebenfalls für die Positionsliste **A4R.pos** und die Dateien **PickGrpR** und **CheckgrpR** durch. Speichern Sie die Files im Verzeichnis **CiROS_CPFactory / Rob_Montage / RV-4FL / Programs** ab.

Hinweis:

Sie können die neuen Positionslisten A4R.pos , PickGrpR.pos und CheckGrpR.pos nicht mehr für die Simulation nutzen!

g) Dokumentieren Sie Ihren Programmcode.

Hinweis

Zu einer vollständigen Dokumentation sind folgende Angaben notwendig:

1. Kopfbereich:

- Programmname
- Kurze Beschreibung
- Name des Autors
- Freigabedatum mit Versionsnummer
- Änderungen mit Datumsangabe
- Wichtige Parametereinstellungen (z. B. Positionsliste, TCP Einstellung)

2. Quellcodebereich:

- Kurze Beschreibung von Programmsequenzen

Ergänzen Sie entsprechend Ihre Dokumentation für **A4**.

Programm A4R für CP Factory Station Roboter montage mit Mitsubishi Roboter RV-4FL.
Autor: Hans Mustermann Freigabe: 20.11.2015 / Version 1.0 Änderungen: IP Adresse Robotersteuerung: 192.168.0.20 IP Adresse PC: 192.168.0.120 Port Nr. Robotersteuerung: 10001
Hauptprogramm A4R : Roboter prüft den Greiferstatus und lagert dann den Greifer entsprechend der Prüfung ins Werkzeugwechselsystem zurück und schließt den Vakuumgreifer an. Roboter entnimmt der Transportbox eine wählbare Platine und bestückt einen Palettenträger an der Stopper-Position im Bypass. Externe Programme: GRPLock - Greifer an der Wechselvorrichtung anschließen GRPRelease - Greifer an die Wechselvorrichtung übergeben GRPClose - Greifer öffnen GRPClose - Greifer schließen GRPVacOn - Vakuum einschalten GRPVacOff - Vakuum ausschalten PickGrpR - Greifertyp auswählen und anschließen CheckGrpR - Greiferstatus überprüfen und gegebenenfalls Greiferwechsel

Aufgabe 5: Optische Positionsprüfung eines Werkstücks

■ Lernziele

Wenn Sie diese Aufgabe bearbeitet haben,

- können Sie den Ablaufplan für eine komplexe Anwendung erstellen.
- können Sie den Einsatz der Kamera für eine Anwendung konzipieren.
- können Sie die Kameraauswertung ins Programm übernehmen.
- können sie die Sensorik bei der Montage identifizieren und in den Ablauf integrieren.
- können sie die Kollisionsprüfung in der Simulation durchführen.
- Können Sie die modulare Programmerstellung durchführen.
- können Sie die Kommunikationsschnittstelle zwischen CIROS und der realen Robotersteuerung konfigurieren und den Download von Programmen und Positionslisten in die reale Steuerung durchführen.
- können Sie umfangreiche Positionsliste in der Hardware testen und optimieren.
- können Sie ein einfaches Programm im Debug-Modus im Abgleich mit der Hardware analysieren und optimieren.

■ Problemstellung

In einem Betrieb soll eine neue Roboterstation in den Montageprozess integriert werden. Die Funktionalität des Roboters ist geprüft. Sie erhalten als Facharbeiter den Auftrag, die Leistungsfähigkeit der Kamera zu überprüfen. Obwohl die Funktionalität der Kamera nicht in der Simulationsumgebung getestet werden kann, soll aus Sicherheitsgründen der Testablauf zunächst in der Simulation geprüft werden.

■ Arbeitsaufträge

1. Erstellen und dokumentieren Sie in der Simulation Ihr Programm A5, das folgende Aufgaben erfüllt:
 - Der Roboter wartet bis das Werkstück an der Stopper-Position bereitsteht.
 - Roboter entnimmt das Werkstück vom Palettenträger und legt es auf dem LED-Feld der Kamera ab.
 - Simulation der Kameraauswertung (Lage ist ok).
 - Roboter entnimmt Werkstück und legt es im Montageplatz ab.
 - Falls die Sensorwerte ok sind, dann wird das Werkstück eingespannt.
 - Roboter entnimmt das Werkstück aus der Montage und legt es zurück auf den Palettenträger.
2. Führen Sie eine Inbetriebnahme des in (1) erstellten Programms für die reale Roboterstation durch.

■ Arbeitshilfen

- Mitsubishi Controller CR750/CR751 – Instruction Manual – Detailed Explanation of Functions and Operations

- CIROS® Hilfe -> Programmierung -> Melfa Basic V
- Kapitel 5 in den Grundlagen
- Technische Dokumentation CP_Factory Robotermonatagezelle, Kapitel 2.

1. Erstellen und dokumentieren Sie in der Simulation Ihr Programm A5, das folgende Aufgaben erfüllt:

- Der Roboter wartet bis das Werkstück an der Stopper-Position bereitsteht.
- Roboter entnimmt das Werkstück vom Palettenträger und legt es auf dem LED-Feld der Kamera ab.
- Simulation der Kameraauswertung (Lage ist ok).
- Roboter entnimmt Werkstück und legt es im Montageplatz ab.
- Falls die Sensorwerte ok sind, dann wird das Werkstück eingespannt.
- Roboter entnimmt das Werkstück aus der Montage und legt es zurück auf den Palettenträger.

Es wird vorausgesetzt, dass Sie schon die Aufgaben 3 und 4 erfolgreich gelöst haben.

- a) Zur Entwicklung des Programms A5 überlegen Sie sich zunächst, welche Eingänge und Ausgänge und welche Positionen und Variablen Sie deklarieren müssen.

Lösung:

- Öffnen Sie den **Modell-Explorer** und öffnen Sie zur Komponente **Montage_RV-4FL** die Liste der Eingänge. Die Namen der Eingänge finden Sie auch in der Technischen Dokumentation zur RoboterMontagestation mit den entsprechenden Funktionsbeschreibungen:

DEF IO DI_PalletInFront	= Bit,244	' Signal von der SPS
DEF IO DI_WPAAtStopAvail	= Bit,251	' Signal von der SPS
DEF IO DI_GrpAvailable	= Bit,902	' Greifer mit Roboter verbunden
DEF IO DI_WPAvail	= Bit,17	' Werkstück ist in Montagestation vorhanden
- Der Ausgang zum Einspannen des Werkstücks ist wie folgt gegeben:

DEF IO DO_WPMountLock	= Bit,16	' Werkstück einspannen
-----------------------	----------	------------------------
- Als Position benötigen Sie die Positionen für die TCPs, eine Hilfsposition und die neue „simulierte“ Position nach der Kameraauswertung:

' Positionen	
DEF POS PtGripperNone	' TCP am Flansch
DEF POS AUX	' Hilfsposition
DEF POS PVisionNew	' Position nach Kameraauswertung
- Als Variablen benötigen Sie die Variablen für die Lineargeschwindigkeit:

DEF INTE slow	' Variable für Lineargeschwindigkeit
DEF INTE fast	' Variable für Lineargeschwindigkeit
- Die Positionsliste **A5.pos** besteht aus den Positionen der Positionsliste **A3.pos** und einer zusätzlichen Ausgangsposition **PVision**, die im Zentrum des LED-Feldes der Kamera liegen soll.

- b) Erstellen Sie den Programmcode entsprechend des vordefinierten Ablaufs und dokumentieren Sie diesen.

Lösung: Ein mögliches Programm könnte wie folgt aussehen.

```

[Montage_RV-4FL] \Programs\A5.mb5 *

' Aufgabe 5
' Externe Programme:
' "GRPLock" - Greifer an der Wechsellvorrichtung übernehmen
' "GRPRelease" - Greifer an die Wechsellvorrichtung übergeben
' "GRPClose" - Greifer öffnen
' "GRPClose" - Greifer schließen
' "CheckGrp" - Greiferstatus prüfen und gegebenenfalls Greifer austauschen
' "PickGrp" - beliebigen Greifer anschließen
' -----
' Eingänge
DEF IO DI_PalletInFront = Bit,244 ' Signal von der SPS
DEF IO DI_WPatStopAvail = Bit,251 ' Signal von der SPS
DEF IO DI_GrpAvailable = Bit,902 ' Greifer mit Roboter verbunden
DEF IO DI_PCBPalletNo = Byte,172 ' Palettensnummer
DEF IO DI_WPAvail = Bit,17 ' Werkstück in Montagestation
DEF IO DI_WPOrientOk = Bit,18 ' Orientierung des Werkstücks ist OK
DEF IO DI_WPHoleOK = Bit,19 ' Bohrung ist an der richtigen Position

' Ausgänge
DEF IO DO_WPMountLock = Bit,16 ' Werkstück einspannen
' -----
' Positionen
DEF POS PtGripperNone ' TCP am Flansch
DEF POS PtGripperWp ' TCP Werkstückgreifer
DEF POS AUX ' Hilfsposition
DEF POS PVisionNew ' Position nach Kameraauswertung
DEF POS PVision ' Position vor Kameraauswertung
' -----
DEF INTE slow ' Variable für Lineargeschwindigkeit
DEF INTE fast ' Variable für Lineargeschwindigkeit
DEF INTE M_GripperType ' Greifertyp
' -----
' Parameter setzen
PtGripperWp = (0,0,170,0,0,33.50) ' TCP Werkzeuggreifer
PtGripperNone = (0,0,0,0,0,33.50) ' TCP im Flansch
slow = 20
fast = 150
' -----

[Montage_RV-4FL] \Programs\A5.mb5

' Programmstart

' Warten bis Werkstück in Stopperposition
WAIT DI_WPatStopAvail = 1

' TCP setzen und Geschwindigkeiten definieren.
TOOL PtGripperNone ' TCP setzen
JOVRD 30 ' Geschwindigkeitsparameter für Achsen
SPD fast ' Lineargeschwindigkeit
M_GripperType = 2
CALLP "CheckGrp" ' ausgewählten Greifer anschließen

' Zur Stopperposition fahren und Frontschale greifen
TOOL PtGripperWP
MOV PCarrierStop1,-50
MVS PCarrierStop1
DLY 0.5
CALLP "GRPClose"
DLY 0.5
MVS PCarrierStop1,-50

' Frontschale auf dem Kamerafeld ablegen
MOV PVision,-30
SPD slow
MVS PVision
DLY 0.5
CALLP "GRPOpen"
MVS PVision,-30
Tool PtGripperNone
MOV PHome

' Kameraauswertung
DLY 3
PVisionNew = PVision ' In der Simulation
'

```



```

[Montage_RV-4FL] \Programs\A5.mb5 *
' Frontschale vom Kamerafeld abholen und in Montage ablegen
MOV PVisionNew,-30
SPD slow
MVS PVisionNew
DLY 0.5
CALLP "GRPClose"
MVS PVision,-50
MOV PASsembleWP,-50
MVS PASsembleWP
DLY 0.5
'
'Werkstück im Montageplatz einspannen
CALLP "GRPOpen"
MVS PASsembleWP,-50
WHILE (DI_WPAvail = 0) OR (DI_WPOrientOk = 0) OR (DI_WPHoleOK = 0)
WEND
DO_WPMountLock = 1
'
Tool PtGripperNone
MOV PHome
DLY 1
'
' Montiertes Werkstück auf Palettenträger zurücklegen
TOOL PtGripperWP
MOV PASsembleWP,-50
MVS PASsembleWP
CALLP "GrpClose"
MVS PASsembleWP,-50
MOV PCarrierStop1,-50
MVS PCarrierStop1
DLY 0.5
CALLP "GRPOpen"
MVS PCarrierStop1,-50
MOV PHome
'
' Greifer zurücklagern
M_GripperType = 4
CALLP "CheckGrp"

END

```

- c) Kompilieren Sie das Projekt A5 und starten Sie die Simulation, falls keine Syntaxfehler gemeldet werden. Prüfen Sie ob der Ablauf korrekt ist und keine Kollisionen sichtbar auftreten.

Lösung:

- Der Ablauf ist korrekt und es sind keine Kollisionen beobachtet worden.
Beachten Sie, dass die Kameraauswertung hier in der Simulation nicht durchgeführt wird.

- d) Datensicherung: Speichern Sie das Programm **A5** und die Positionsliste zusätzlich unter den Namen **A5R**. Führen Sie die notwendigen Änderungen durch.

Lösung:

- Speichern Sie das Programm und die Positionsliste zusätzlich unter dem Namen **A5R.mb5** im Verzeichnis **CIROS_CPFactory / Rob_Montage /RV-4FL /Programs** ab.
- Kopieren Sie die Positionen **PGrpStorageWp**, **PHome**, **PCarrierStop1** und **PASsembleWp** von **A3R** in die Positionsliste **A5R**, falls Sie Aufgabe 3 schon erfolgreich gelöst haben.
- Ändern Sie beim Aufruf die Namen der externen Programme PickGrp und CheckGrp.
- Ändern Sie die Daten der Eingangssignale von der SPS:
DEF IO DI_PalletInFront = Bit, 2144; DEF IO DI_WPAStopAvail = Bit, 2151

2. Führen Sie reale Inbetriebnahme Ihres Projektes A5 durch.

- a) Die Arbeitszelle **CP_RobotAssembly** ist geöffnet und der PC ist mit der Robotersteuerung verbunden. Führen Sie einen Download Ihres Programme **A5R** sowie der zugehörigen Positionsliste in die Robotersteuerung durch und prüfen Sie, ob die Programme im RCI Explorer angezeigt werden.

Lösung:

- Es wird vorausgesetzt, dass Sie die Aufgaben 3 und 4 schon erfolgreich bearbeitet haben und dass Sie mittels des Einrichtemodus am Bedienpanel einen Palettenträger mit Werkstück an die Stopper-Position im Bypass gefahren haben.
- Führen Sie einen Download Ihres Programmes **A5R.mb5** und der entsprechenden Positionsliste durch. Sie erhalten jeweils eine Meldung, ob der Download erfolgreich durchgeführt werden konnte.
- Wählen Sie die Gruppe **Programme** im **RCI-Explorer**. Im rechten Fenster wird das Programm **A5R.mb5** angezeigt.

Hinweis:

Die externen Programme sind schon der Steuerung und müssen nicht mehr geladen werden.

- b) Überprüfen Sie die Verdrahtung der Eingangssignale.

Lösung:

- Die Verdrahtung der Sensoren **DI_PalletInFront**, **DI_WPAAtStopAvail**, **DI_WPAvail**, **DI_WPOrientOK** und **DI_WPHoleOK** wurde schon in den Aufgaben 3 und 4 überprüft. Dies gilt auch für den Ausgang **DO_WPMountWP**.

- c) Überprüfen Sie Positionen der Positionsliste **A5R**.

Hinweis:

Sie finden eine detaillierte Beschreibung zur Handhabung des PGH zum Einrichten von Positionen in Kapitel 3 des Mitsubishi Handbuchs **Mitsubishi Controller CR750/CR751 – Instruction Manual – Detailed Explanation of Functions and Operations**.

Zur Positionsprüfung müssen Sie den TCP ändern. Sie können dies mit dem PGH im Hand-Modus durchführen, siehe **§4** in der **Einleitung**.

Lösung:.

- es muss nur noch die Position **PVision** in **A5R.pos** geprüft werden.
- Übernehmen Sie zunächst den entsprechenden Wert von **P_Vision** aus der globalen Positionsliste **UBP.pos**.
- Position **PVision**:
 - Starten Sie das Programm **A5R** und stoppen es, sobald der Roboter das Werkstück vom Palettenträger abgeholt hat.
 - Schalten Sie an der Robotersteuerung den Modus **MANUAL** ein und aktivieren Sie das **PGH**.

- Öffnen Sie im **PGH** die Positionsliste zum Programm **A5R** und wählen Sie die Position **PVision**.
 - Klicken Sie auf die Taste **JOG** und wählen den Modus **XYZ**.
 - Bewegen Sie den Roboter oberhalb des Kamerafeldes.
 - Klicken Sie erneut auf die Taste **JOG**, dann gelangen Sie wieder in die Anzeige der Position.
 - Bewegen Sie den Roboter mit der **MOVE** Taste in die Position **PVision**.
 - Die Position ist korrekt, falls das Werkstück einwandfrei auf dem Kamerafeld aufliegt.
 - Aktivieren Sie den JOG-Modus und bewegen den Roboter ca. 100mm in **+Z** Richtung im **XYZ**-Modus.
- d) Erstellen Sie ein Testprogramm für die Funktionalität der Kamera. Legen Sie ein Werkstück in die Nähe der Position **PVision**. Der Roboter ist in der Startposition mit dem Werkzeuggreifer. Die Kameraprüfung wird gestartet und meldet dem Roboter die Lage. Der Roboter greift das Werkstück und fährt zurück in die Startposition. Falls Fehler in der Kameraauswertung sichtbar werden, dann erfolgt keine Aktion vom Roboter.

Hinweis:

- a) Die Auswertung der Kamera kann nur in Verbindung mit der Hardware erfolgen.
- b) Die Parametereinstellungen für die Kommunikation zur Kamera sind schon erfolgt.
- c) Die Kommunikation mit der Kamera erfolgt über folgende Befehle:
 - Kommunikation eröffnen: Open "COM3:" As #1
 - Rückmeldung lesen und verwerfen: Input #1, CRETURN\$
 - Programmnummer (z.B. 1) in der Kamera auswählen: Print #1, "CHANGEPRG 1"
 - Rückmeldung lesen und verwerfen
 - Kamerabild aufnehmen: Print #1, "IMAGE"
 - Rückmeldung lesen und verwerfen
 - Wert für X-Koordinate anfordern: Print #1, "RNV [X]"
 - Lesen des Wertes: Input #1, X-Position
 - Entsprechend für Y-Koordinate und Drehwinkel
 - Verbindung zur Kamera schließen: CLOSE #1
- d) Der Maximalbereich für die Kamerawerte ist wie folgt gegeben:
 - $100 < X < 300 \text{ mm}$, $-600 < Y < -200 \text{ mm}$, $-360^\circ < \text{Drehwinkel} < 360^\circ$
- e) Der genaue Wert für die Programmnummer in der Kamera muss vor Ort abgeklärt werden.
- f) Die Genauigkeitsangabe der Kamera liegt im mm-Bereich.

Lösung:

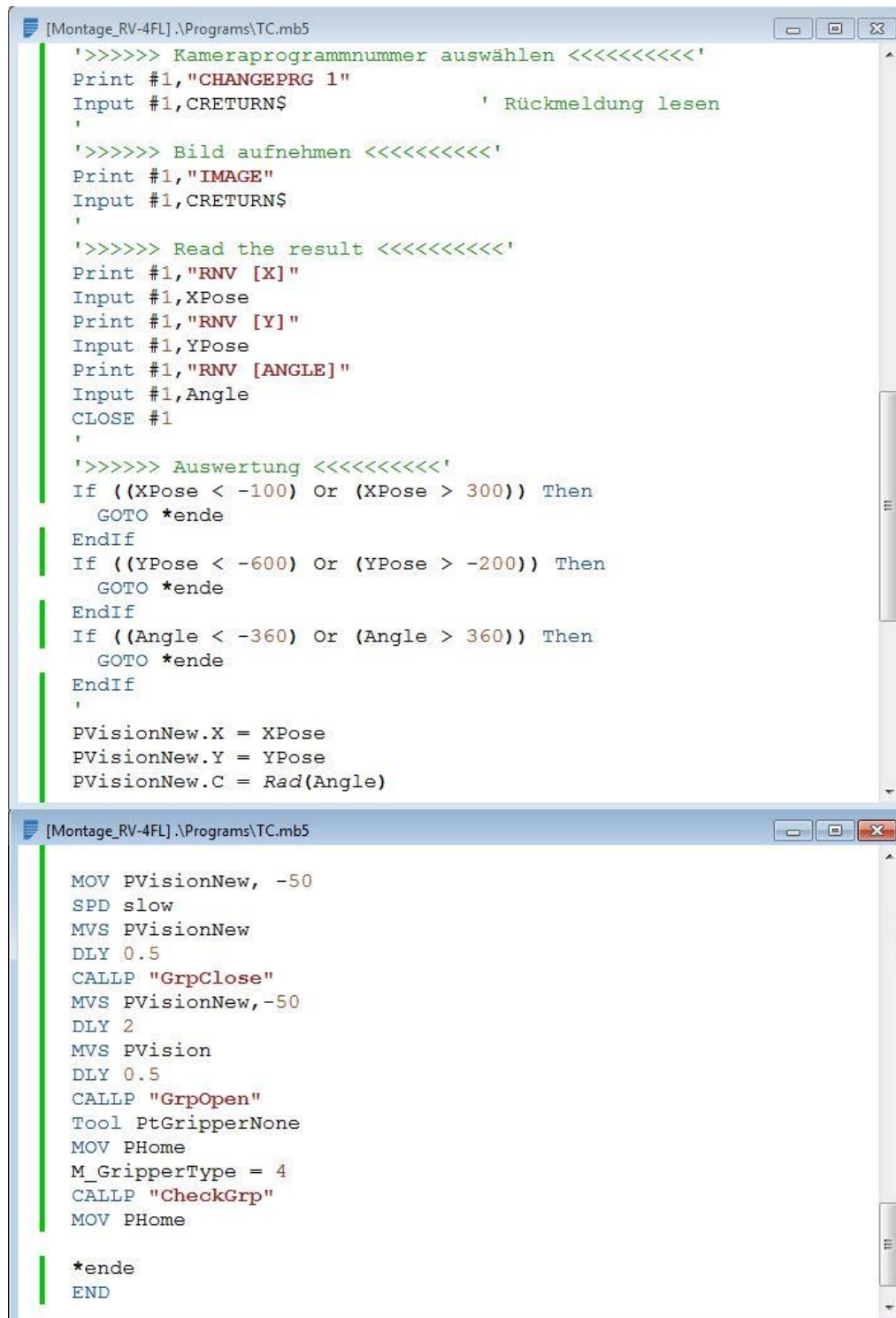
Ein mögliches Programm kann wie folgt aussehen:

```

[Montage_RV-4FL] \Programs\TC.mb5

'#####'
'### Programm TC: Kamera testen
'### Werkstück (Frontschale) in die Nähe des Zentrums des
'### Lichtfeldes der Kamera legen.
'### Externe Programme:
'### "GrpClose", "GrpOpen"
'### "GrpLock", "GrpRelease"
'### "CheckGrp", "PickGrp"
'#####'
'>>>>> Variablendeklaration<<<<<<<<'
DEF POS PVisionNew
DEF POS PtGripperNone ' TCP am Flansch
DEF POS PtGripperWp ' TCP Werkstückgreifer
'
DEF INTE M_GripperType
DEF INTE XPose
DEF INTE YPose
DEF INTE Angle
'>>>>> Parameterwerte <<<<<<<<'
PtGripperWp = (0,0,170,0,0,33.50)
PtGripperNone = (0,0,0,0,0,33.50)
XPose = -1000
YPose = -1000
Angle = -400
'
JOVRD 30
SPD 150
' Startwert für die Kameraauswertung
PVisionNew = PVision
'-----
' Programmstart
Tool PtGripperNone
MOV PHome
M_GripperType = 2
CALLP "CheckGrp"
Tool PtGripperWp
'
'>>>>> Kommunikation zur Kamera öffnen <<<<<<<<'
Close #1
Dly 2
Open "COM3:" As #1
Input #1, CRETURN$ ' Rückmeldung lesen
'

```



```

[Montage_RV-4FL] \Programs\TC.mb5

'>>>>> Kameraprogrammnummer auswählen <<<<<<<<<'
Print #1,"CHANGEPRG 1"
Input #1,CRETURN$          ' Rückmeldung lesen
'

'>>>>> Bild aufnehmen <<<<<<<<<'
Print #1,"IMAGE"
Input #1,CRETURN$
'

'>>>>> Read the result <<<<<<<<<'
Print #1,"RNV [X]"
Input #1,XPose
Print #1,"RNV [Y]"
Input #1,YPose
Print #1,"RNV [ANGLE]"
Input #1,Angle
CLOSE #1
'

'>>>>> Auswertung <<<<<<<<<'
If ((XPose < -100) Or (XPose > 300)) Then
    GOTO *ende
EndIf
If ((YPose < -600) Or (YPose > -200)) Then
    GOTO *ende
EndIf
If ((Angle < -360) Or (Angle > 360)) Then
    GOTO *ende
EndIf
'

PVisionNew.X = XPose
PVisionNew.Y = YPose
PVisionNew.C = Rad(Angle)

MOV PVisionNew, -50
SPD slow
MVS PVisionNew
DLY 0.5
CALLP "GrpClose"
MVS PVisionNew, -50
DLY 2
MVS PVision
DLY 0.5
CALLP "GrpOpen"
Tool PtGripperNone
MOV PHome
M_GripperType = 4
CALLP "CheckGrp"
MOV PHome

*ende
END
  
```

- Die Positionsliste **TC.pos** besteht aus den beiden Positionen **PVision** und **PHome**, die beide aus der Positionsliste **A5R.pos** übernommen werden können.
- Zur Wiederholung des Tests verändern Sie die Lage des Werkstücks auf dem LED-Feld der Kamera und starten das Programm erneut.
- Der Test ist erfolgreich, wenn der Roboter bei drei Versuchen stets das Werkstück einwandfrei greifen kann und es jeweils in die Startposition transportiert.

e) Integration der Kamera in das Programm **A5R**:

Erstellen Sie jetzt ein externes Programm, dass die Kameraprüfung nach Ablage des Werkstücks auf dem Kamerafeld durchführt. Nennen Sie dieses Programm **PrgCamera**. Falls die Kameraprüfung fehlerhaft ist, soll das Programm beendet werden. Testen Sie mittels des **RCI-Explorers** den Programmablauf und optimieren Sie die Geschwindigkeit. Dokumentieren Sie Ihre Ergebnisse für eventuelle Änderungen im Programmablauf.

Hinweis:

Führen Sie im Programm **A5R** zwei globale Positionsvariablen **P_VisionNew** und **P_Vision** ein und setzen **P_Vision = P_VisionNew**. Der Vorteil ist, dass Sie dann diese Variablen direkt in Ihrem externen Programm **PrgCamera** nutzen können.

Lösung:

- Ergänzen Sie Ihre Deklaration in **A5R** wie folgt:
 DEF INTE m_error
 DEF POS P_VisionNew
 DEF POS P_Vision
- Ersetzen Sie die Programmzeilen zur Kameraauswertung wie folgt:
- ‘ Kameraauswertung
 DLY 5 ‘ Hier können Sie die Lage des Werkstücks verändern
 m_error = 0
 P_Vision = P_VisionNew
 CALLP “PrgCamera“
 IF m_error > 0 THEN GOTO *ende
 P_VisionNew = P_VisionNew
 (nächste Zeile gestrichen)
- Eine Lösung für das Programm **PrgCamera** lässt sich sehr leicht aus dem Programm **TC.mb5** ableiten:


```

[Montage_RV-4FL] \Programs\PrgCamera.mb5

'#####'
'### Programm: PrgCamera'
'###'
'### Programmbeschreibung:'
'### Kommunikation zur Kamera öffnen'
'### Kameraprogramm auswählen'
'### Kamerabild generieren'
'### Auswertung des Kamerabildes'
'###'
'#####'
'      Program'
'#####'
'>>>>> Variablendeklaration<<<<<<<<<'
DEF INTE XPose
DEF INTE YPose
DEF INTE Angle
'>>>>> Werte festlegen <<<<<<<<<'
XPose = -1000
YPose = -1000
Angle = -400
'
'>>>>> Kommunikation zur Kamera öffnen <<<<<<<<<'
Close #1
Dly 2
Open "COM3:" As #1
Input #1, CRETURN$      ' Rückmeldung lesen
'



[Montage_RV-4FL] \Programs\PrgCamera.mb5

Open "COM3:" As #1
Input #1, CRETURN$      ' Rückmeldung lesen
'
'>>>>> Kameraprogrammnummer auswählen <<<<<<<<<'
Print #1, "CHANGEPRG 1"
Input #1, CRETURN$      ' Rückmeldung lesen
'
'>>>>> Bild aufnehmen <<<<<<<<<'
Print #1, "IMAGE"
Input #1, CRETURN$
'
'>>>>> Read the result <<<<<<<<<'
Print #1, "RNV [X]"
Input #1, XPose
Print #1, "RNV [Y]"
Input #1, YPose
Print #1, "RNV [ANGLE]"
Input #1, Angle
CLOSE #1
'
'>>>>> Auswertung <<<<<<<<<'
If ((XPose < -100) Or (XPose > 300)) Then m_error = 1
If ((YPose < -600) Or (YPose > -200)) Then m_error = m_error + 1
If ((Angle < -360) Or (Angle > 360)) Then m_error = m_error + 3

IF m_error = 0 Then
    P_VisionNew.X = XPose
    P_VisionNew.Y = YPose
    P_VisionNew.C = Rad(Angle)
ENDIF

END

```

- Laden Sie das modifizierte Programm **A5R** und das Programm **PrgCamera** in die Robotersteuerung.
- Die Kameraeinstellung wurde überprüft und die Kamera wurde eingeschaltet. Die Programmnummer ist hier im Beispiel = **1**.
- Wählen Sie zum Test Ihres Programms im **RCI-Explorer** zunächst den **Debug**-Modus. In der online-Anzeige Ihres Programms sind die Programmzeilen nummeriert und die Kommentare sind gelöscht. Mit  können Sie Breakpoints in den entsprechenden Programmzeilen Ihres Programms setzen.
- Das Programm wird jeweils vor Ausführung der entsprechenden Programmzeile gestoppt.
- Mit dem Befehl  können Sie Ihr Programm mit Breakpoint-Überwachung starten.

f) Testen Sie Ihr Programm im Automatikmodus

- Starten Sie Ihr Programm über das Bedienfeld der Robotersteuerung.
 - Wählen Sie das Programm **A5R** aus.
 - Die **OVRD** Auswahl bewirkt eine prozentuale Reduzierung der im Programm definierten maximalen Geschwindigkeit.
 - Schalten Sie die Servo Motoren ein und starten Sie das Programm.

g) Datensicherung: Führen Sie einen Upload des aktuellen Roboterprogramms und der aktuellen Positionsliste in der Robotersteuerung auf den PC durch.

Lösung:

Die Arbeitszelle **CP_RobotAssembly** ist geöffnet und der PC ist mit der Robotersteuerung verbunden. Öffnen und aktivieren Sie das Programmfenster mit dem Programm **A5R.mb5**. Wählen Sie das Menü **EXTRAS -> Online-Kopplung -> Upload realer Roboter -> PC**. Das Programm wird auf den PC geladen und überschreibt das vorhandene Programm **A5R**. Führen Sie dies ebenfalls für die Positionsliste **A5R.pos** und die Datei **PrgCamera** durch. Speichern Sie die Files im Verzeichnis **CiROS_CPFactory / Rob_Montage / RV-4FL / Programs** ab.

Aufgabe 6: Anwendung der externen Funktionsbibliothek

■ Lernziele

Wenn Sie diese Aufgabe bearbeitet haben,

- können Sie eine komplexe Funktionsbibliothek analysieren und die Anwendungsmöglichkeiten dokumentieren.
- können Sie die Funktionsbibliothek zur optischen Kontrolle der Montage eines Werkstücks anwenden.
- können Sie verschiedene Fehlersituationen in der Robotik identifizieren und im Programmablauf berücksichtigen.
- können Sie das Konzept einer externen universellen Liste von Positionsvariablen anwenden und in der realen Roboteranwendung einlernen.
- beherrschen Sie die Methode der modularen Programmerstellung.
- können Sie die Kommunikationsschnittstelle zwischen CIROS und der realen Robotersteuerung konfigurieren und den Download von Programmen und Positionslisten in die reale Steuerung durchführen.
- können Sie das Programm im Debug-Modus im Abgleich mit der Hardware analysieren und optimieren.

■ Problemstellung

In einem Betrieb soll eine neue Roboterstation in den Montageprozess integriert werden. Die Firmenleitung hat ein externes Softwareunternehmen mit der Erstellung einer Funktionsbibliothek beauftragt, die auch die Kommunikation mit dem externen Steuerungssystem berücksichtigt. Sie erhalten als Facharbeiter den Auftrag, die Funktionsbibliothek zu dokumentieren und diese an Hand einer einfachen Anwendung sowohl in der Simulationsumgebung als auch an der Hardware zu testen.

■ Arbeitsaufträge

1. Geben Sie eine kurze inhaltliche Dokumentation der Funktionen in der Bibliothek.
2. Erstellen Sie in der Simulation ein Programm **A6** mit Hilfe der Funktionsbibliothek, das folgende Aufgaben erfüllt:
 - Der Roboter wartet bis das Werkstück an der Stopper-Position und die Transportbox mit Platinen bereitstehen.
 - Der Roboter entnimmt das Werkstück vom Palettenträger und legt es auf dem LED-Feld der Kamera ab.
 - Kameraprüfung der Werkstücklage.
 - Der Roboter entnimmt das Werkstück und legt es im Montageplatz ab.
 - Der Roboter entnimmt die angeforderte Platine aus der Transportbox und fügt sie in das Werkstück ein.
 - Der Roboter entnimmt das montierte Werkstück aus der Montage und legt es zurück auf den Palettenträger.
3. Führen Sie eine Inbetriebnahme des in (2) erstellten Programms für die reale Roboterstation durch.

■ Arbeitshilfen

- Mitsubishi Controller CR750/CR751 – Instruction Manual – Detailed Explanation of Functions and Operations
- CIROS® Hilfe -> Programmierung -> Melfa Basic V
- Kapitel 5 in den Grundlagen
- Technische Dokumentation CP_Factory RoboterMontagezelle, Kapitel 2.
- Konzept der Funktionsbibliothek.
 - Die Funktionsbibliothek finden Sie im Ordner **CIROS_CPFactory / Rob_Montage / RV4FL / Programs/Bibliothek**. Alle Funktionen enthalten eine ausführliche Fehlerbehandlung. Alle Programme sind im Quellcode vorhanden, Sie sollten sehr ausführlich die Programme studieren und ihren Anwendungsbereich analysieren. Einige Programme sind von besonderer Bedeutung:
 - **UBP – User Based Program:**
Mit einem solchen Programm kann der Anwender Variablen, Flags und Positionen, die er in verschiedenen Programmen nutzen möchte anlegen. Damit kann der Anwender die Verwendung von Positionen in verschiedenen Programmen zusammenfassen. Dieses Programm muss in der Simulation zu Beginn gestartet werden. Für die Robotersteuerung ist dies nicht notwendig. Hier muss das Programm nur als PRGUSER Parameter registriert werden. Feste Positionswerte, die eingelernt werden müssen, sind in der Positionsliste **UBP.pos** hinterlegt. Diese globalen Positionen müssen im Programm **UPB** auskommentiert sein. Die Positionsliste ist schon in der Robotersteuerung vorhanden. Sie können die Werte mit ihren Werten in der Positionsliste von **A5** vergleichen.
 - **Programm 999:**
Dieses Programm initialisiert zahlreiche globale Variablen und Flags vom **UBP** Programm. Dieses Programm muss stets zu Beginn gestartet werden.
 - **Programme INITIALIZE und SENSORCHECK:**
Das erste Programm initialisiert zusätzliche globale Variablen und das zweite Programm prüft den Status der Sensoren in der Station.
 - **Profinet Eingänge DI_RetryLastStep und DI_ExitCurrCycle:**
Diese Eingänge sind nur von Bedeutung, wenn die Anwendung von extern gesteuert wird. Ist der erste Eingang gesetzt, dann besagt dies, dass der Vorgang nochmals wiederholt werden soll. Ist der zweite Eingang gesetzt, so besagt dies, dass der Vorgang beendet werden soll. Wird die Anwendung nicht von extern gesteuert, so sind die beide Eingänge stets auf 0 gesetzt.

1. Geben Sie eine kurze inhaltliche Dokumentation der Funktionen in der Bibliothek.**Lösung:**

- **ENRGSAVEVACON:**
Falls das Signal zur Vakuumsteuerung ausfällt, dann wird das Vakuum wieder eingeschaltet.
- **GETCAMRESULT:**
Der Aufruf benötigt einen Eingabeparameter zur Programmnummer. Es wird die Positionslage des Werkstücks als Ausgabewert geliefert.
- **GETCURTOOLNO:**
Die Funktion prüft den aktuellen Greiferstatus am Roboter. (1 = Vakuumgreifer, 2 = Werkzeuggreifer, 3 = Greifer für Sicherungen, 4 = kein Greifer)
- **GETFUSEMAGNO:**
Funktion prüft, welche Magazine Sicherungen bereitstellen. Das Prüfergebnis wird in eine globale Variable **m_AvailFuseMagNo** geschrieben. (=3, falls in allen Magazinen Sicherungen vorhanden, =2, falls nur in den ersten beiden Magazine Sicherungen vorhanden, = 1, falls nur im ersten Magazin Sicherungen vorhanden sind).
- **GRPOPEN, GRPCLOSE, GRPVACON, GRPVACOFF, GRPRELEASE, GRPLOCK:**
Funktionen sind schon in den Aufgaben 3-5 erläutert worden.
- **INITIALIZE:**
Öffnet die Einspannvorrichtung, setzt **m_IsTryUsedByRob** = 0 (d.h. Platinenbox wird nicht vom Roboter genutzt) und setzt **m_IsVacEnrgSvEna** = 0 (d.h. der spezielle Energiesparmodus beim Vakuum wird nicht genutzt.)
- **MONITORHOME:**
Funktion prüft, ob der Roboter sich in der Nähe der Startposition befindet.
- **MonitorPalWS:**
Funktion prüft, ob der Roboter sich im Bereich der Platinenbox befindet.
- **MOUNTBOTFUSE:**
Roboter montiert eine Sicherung an der unteren Sicherungshalterung der Platine.
- **MOUNTPCB:**
Roboter platziert Platine in die Montagestation.
- **MOUNTTOPFUSE:**
Roboter montiert eine Sicherung an der oberen Sicherungshalterung der Platine.
- **PCBTRAYCNTRL:**
Funktion prüft, ob die Stopper für die Platinenbox ausgefahren sind und setzt die globale Variable **m_IsTrayLocked** = 1, falls dies der Fall ist.
- **PICKFRMSTOPR:**
Funktionsaufruf benötigt als Übergabeparameter die Stoppernummer. Der Roboter entnimmt vom Palettenträger das Werkstück.
- **PICKFRMVISON:**
Funktionsaufruf benötigt als Übergabeparameter die Lage des Werkstücks. Roboter entnimmt vom Kamerafeld das Werkstück.
- **PICKFUSFRMAG:**
Funktionsaufruf benötigt als Übergabeparameter die Nummer des Magazins. Roboter entnimmt Sicherung aus dem Magazin.
- **PICKNEWTOL:**
Funktionsaufruf benötigt als Übergabeparameter die Nummer des Greifers. Es wird geprüft, mit

welchem Greifer der Roboter aktuell verbunden ist. Falls dies nicht der gewünschte Greifer ist, dann wird der Greifer zurückgelagert und der Roboter entnimmt der angeforderten Greifer.

- **PICKPCBFRPAL:**

Funktion prüft, ob die Platinenbox vorhanden ist. Wenn dies der Fall ist, dann entnimmt der Roboter die angeforderte Platine.

- **PICKWPFRMASS:**

Roboter entnimmt das Werkstück aus der Montageposition.

- **PLACETOSTOPR:**

Funktionsaufruf benötigt als Übergabeparameter die Nummer des Stoppers. Es wird geprüft, ob der Palettenträger dort verfügbar ist. Wenn dies der Falls ist, dann legt Roboter das Werkstück auf den Palettenträger.

- **PLACETOVISION:**

Roboter legt das Werkstück auf das Kamerafeld.

- **PLACEWPTOASS:**

Roboter legt das Werkstück in die Montagestation und das Werkstück wird eingespannt.

2. Erstellen Sie in der Simulation ein Programm A6 mit Hilfe der Funktionsbibliothek, das folgende Aufgaben erfüllt:

- Der Roboter wartet bis das Werkstück an der Stopper-Position und die Transportbox mit Platinen bereitstehen.
- Der Roboter entnimmt das Werkstück vom Palettenträger und legt es auf dem LED-Feld der Kamera ab.
- Kameraprüfung der Werkstücklage.
- Der Roboter entnimmt das Werkstück und legt es im Montageplatz ab.
- Der Roboter entnimmt die angeforderte Platine aus der Transportbox und fügt sie in das Werkstück ein.
- Der Roboter entnimmt das montierte Werkstück aus der Montage und legt es zurück auf den Palettenträger.

Lösung:

- Öffnen Sie die Projektverwaltung und definieren Sie ein neues Projekt **A6**. Fügen Sie dem Projekt das vorbereitete leere Programm **A6.mb5** hinzu. Zusätzlich fügen Sie alle externen Programme aus der Bibliothek und die Positionsliste **UBP.pos** hinzu.
- Eine Lösung könnte wie folgt aussehen:

```

[Montage_RV-4FL] \Programs\A6.mb5
#####
'### Aufgabe 6:
'###
'### Programmbeschreibung:
'### Werkstück vom Stopper abholen
'### Werkstück auf Kamerafeld absetzen
'### Kameraprüfung ( in Simulation auskommentieren)
'### Werkstück vom Kamerafeld abholen
'### Werkstück in Montageplatz ablegen
'### Platine aus Transportbox abholen
'### Platine montieren
'### Montiertes Werkstück an Stopperposition auf
'### Palettenträger zurücklegen
'###
'###
#####
'

```

```

[Montage_RV-4FL] \Programs\A6.mb5
'#####'
'           Lokale Positionen
'#####'
Def Pos pVisionResult
'#####'
'           Eingänge und Ausgänge
'#####'
'>>>>> Eingänge - ProfiNet <<<<<<<'
Def Io DI_StopperNo      = Byte,164
DEF IO DI_WPatStopAvail  = Bit,251
DEF IO DI_PalletInFront  = Bit,244
'#####'
'>>>>> Ausgänge - ProfiNet <<<<<<<'
Def Io ReturnErrorValue = Byte,164 'Fehlermeldung an SPS

'#####'
'           Lokale Variablen
'#####'
Def Inte StopperNumber
Def Inte CamPrgNumber
'#####'
'           Programm
'#####'
CALLP "UBP"
CALLP "999"

' Warten bis Werkstück in Stopperposition und Box mit Platinen
' vorhanden.
WHILE (DI_PalletInFront = 0) OR (DI_WPatStopAvail = 0)
WEND
CALLP "PCBTrayCntrl"
'>>>>> Werte für lokale Variablen <<<<<<<'
StopperNumber = DI_StopperNo
CamPrgNumber = 1
m_ReturnedError = 0
CallP "Initialize"      ' Initialisierung von globalen Variablen
CallP "SensorCheck"     ' Überprüfung der Sensorwerte in der Anlage
If (m_ReturnedError > 0) GoTo *EndOfPrg
'#####'

'>>>>> Werkstück vom Stopper abholen <<<<<<<'
CallP "PickNewTool", m_GripperWP
If (m_ReturnedError > 0) GoTo *EndOfPrg
CallP "PickFrmStopR", StopperNumber
If (m_ReturnedError > 0) GoTo *EndOfPrg
'#####'
'>>>>> Werkstück auf Kamerafeld ablegen <<<<<<<'
CallP "PlaceToVison"
'#####'
'>>>>> Kameraprüfung <<<<<<<'
CallP "GetCamResult", CamPrgNumber
If (m_ReturnedError > 0) GoTo *EndOfPrg
pVisionResult = P_VisionNew
'#####'
'>>>>> Werkstück in Montageplatz einfügen <<<<<<<'
CallP "PickFrmVison", pVisionResult
CallP "PlaceWpToAss"
If (m_ReturnedError > 0) GoTo *EndOfPrg
'#####'
'>>>>> Platine aus Transportbox entnehmen <<<<<<<'
CallP "PickNewTool", m_GripperVac
If (m_ReturnedError > 0) GoTo *EndOfPrg
CallP "PickPCBFrPal"
If (m_ReturnedError > 0) GoTo *EndOfPrg
'#####'
'>>>>> Platine montieren <<<<<<<'
CallP "MountPCB"
'#####'
'>>>>> Montiertes Werkstück auf Palettenträger zurücklagern <<<<<<<'
CallP "PickNewTool", m_GripperWP
If (m_ReturnedError > 0) GoTo *EndOfPrg
CallP "PickWpFrmAss"
'#####'
CallP "PlaceToStopR", StopperNumber%
If (m_ReturnedError > 0) GoTo *EndOfPrg
'#####'
'>>>>> Fehler bei der Ausführung <<<<<<<'
*EndOfPrg

END

```

- a) Datensicherung: Speichern Sie das Programm **A6** und die Positionsliste zusätzlich unter den Namen **A6R**.

Lösung:

- Speichern Sie das Programm und die Positionsliste zusätzlich unter dem Namen **A6R.mb5** im Verzeichnis **CIROS_CPFactory / Rob_Montage /RV-4FL /Programs** ab.
- Ändern Sie die Deklaration der SPS-Signale und entfernen Sie die Kommentierung der Programmzeile
CallP "GetCamResult", CamPrgNumber
- Achten Sie darauf, dass sie **NICHT** die Positionsliste **UBP.pos** durch die entsprechende Datei in der Robotersteuerung überschreiben. Die Datei **UBP.pos** ist im Verzeichnis **CIROS_CPFactory / Rob_Montage /RV4FL /Programs/Bibliothek** nur für die Simulation anwendbar.

3. Führen Sie eine Inbetriebnahme des in (2) erstellten Programms für die reale Robotermontagestation durch.

- a) Die Arbeitszelle **CP_RobotAssembly** ist geöffnet und der PC ist mit der Robotersteuerung verbunden. Führen Sie einen Download Ihres Programms **A6R** in die Robotersteuerung durch und prüfen Sie, ob das Programm im RCI Explorer angezeigt wird.

Lösung:

- Es wird vorausgesetzt, dass die Positionen von **UBP.pos** in der Robotersteuerung alle schon geprüft worden sind. Weiterhin, dass Sie mittels des Einrichtemodus am Bedienpanel einen Palettenträger mit Werkstück an die Stopper-Position im Bypass und die mit Platinen beladene Transportbox an die Stoppposition gefahren haben.
- Führen Sie ein Download des in (2a) geänderten Programms **A6R** durch.
- Testen Sie es zunächst im **Debug**-Modus des RCI-Explorers.

Aufgabe 7: Anwendung der externen Funktionsbibliothek zur Montage einer Platine mit Sicherungen

■ Lernziele

Wenn Sie diese Aufgabe bearbeitet haben,

- können Sie den Ablaufplan unter Anwendung der Funktionsbibliothek erstellen.
- können Sie die Funktionsbibliothek zur optischen Kontrolle der Montage eines Werkstücks effizient einsetzen und ihren Nutzen beurteilen.
- Können Sie verschiedene Fehlersituationen in der Robotik identifizieren und im Programmablauf berücksichtigen.
- Können Sie eine umfangreiche Kollisionserkennung in der Simulation konfigurieren und durchführen.
- beherrschen Sie die Methode der modularen Programmerstellung.
- Beherrschen Sie den Einsatz der externen Funktionsbibliothek.
- Können Sie die Kommunikationsschnittstelle zwischen CIROS und der realen Robotersteuerung konfigurieren und den Download von Programmen und Positionslisten in die reale Steuerung durchführen.
- können sie das Programm im Debug-Modus im Abgleich mit der Hardware analysieren und optimieren.

■ Problemstellung

In einem Betrieb soll eine neue Roboterstation in den Montageprozess integriert werden. Die Firmenleitung hat ein externes Softwareunternehmen mit der Erstellung einer Funktionsbibliothek beauftragt, die auch die Kommunikation mit dem externen Steuerungssystem berücksichtigt. Sie erhalten als Facharbeiter den Auftrag, mit Hilfe der Funktionsbibliothek die Programmierung des geplanten Montageprozesses durchzuführen und zu testen. Aus Sicherheitsgründen sollen Sie das Programm zunächst in der Simulationsumgebung testen.

■ Arbeitsaufträge

1. Erweitern Sie die Funktionsbibliothek um ein Programm, das
 - den aktuellen Greiferstatus des Roboters prüft,
 - und den Greifer in die Wechsellagerung zurücklagert.
 - Falls der Roboter sich nicht im Bereich der Platinenbox befindet, soll er vor dem Zurücklagern zunächst die Startposition **P_Home** anfahren.Nennen Sie das Programm **StoreCurTool**.
2. Testen Sie Ihr Programm in der Simulation. Generieren Sie alle möglichen Greifertypen für den Roboter und prüfen Sie den Fall, wenn sich der Roboter mit dem Vakuumgreifer im Bereich der Platinenbox befindet.
3. Erstellen Sie in der Simulation ein Programm A7 mit Hilfe der Funktionsbibliothek, das folgende Aufgaben erfüllt:
 - Der Roboter wartet bis das Werkstück an der Stopper-Position und die Transportbox mit Platinen bereitstehen.
 - Der Roboter entnimmt das Werkstück und legt es im Montageplatz ab.
 - Der Roboter entnimmt angeforderte Platine aus der Transportbox und fügt sie in das

Werkstück ein.

- Der Roboter entnimmt zwei Sicherungen aus den Magazinen und setzt sie auf der Platine in der jeweiligen Sicherungsvorrichtung ein.
- Der Roboter entnimmt das montierte Werkstück aus der Montage und legt es zurück auf den Palettenträger.
- Der Roboter lagert den Greifer zurück ins Wechselsystem.

4. Führen Sie eine Inbetriebnahme des in (3) erstellten Programms für die reale RoboterMontagestation durch.

■ Arbeitshilfen

- Mitsubishi Controller CR750/CR751 – Instruction Manual – Detailed Explanation of Functions and Operations
- CIROS® Hilfe -> Programmierung -> Melfa Basic V
- Kapitel 5 in den Grundlagen
- Technische Dokumentation CP_Factory RoboterMontagezelle, Kapitel 2.
- Konzept der Funktionsbibliothek.

1. Erweitern Sie die Funktionsbibliothek um ein Programm, das

- den aktuellen Greiferstatus des Roboters prüft,
- und den Greifer in die Wechselvorrichtung zurücklagert.
- Falls der Roboter sich nicht im Bereich der Platinenbox befindet, soll er vor dem Zurücklagern zunächst die Startposition **P_Home** anfahren.

Nennen Sie das Programm **StoreCurTool**.

Hinweis:

Orientieren Sie sich an der Struktur des Programms **PickNewTool** und nutzen Sie das Programm **MonitorPalWS**.

Lösung:

- Öffnen Sie das vorbereitete leere Programm **StoreCurTool.mb5**.
- Eine Lösung könnte zum Beispiel wie folgt aussehen:

```
[Montage_RV-4FL] \Programs\STORECURTOOL.mb5

'#####
'### Programmname:
'### - StoreCurTool
'###
'### Programmbeschreibung:
'### Roboter hat kein Teil gegriffen
'### Den aktuellen Greiferstatus prüfen
'### Den aktuellen Greifer austauschen, falls er nicht mit
'### dem angeforderten Greifer übereinstimmt
'### Den angeforderten Greifer anschließen
'###
'###
'#####

'#####
'##### Deklaration lokaler Position
'#####
DEF POS pStorageTarget
DEF POS pStorageFront
DEF POS pStorageAux
DEF POS AUX
'

'#####
'##### Deklaration Eingänge und Ausgänge
'#####
'>>>>> Ausgänge - Robotersteuerung <<<<<<<'
Def Io DO_MountLock = Bit,16
'

'>>>>> Eingänge - ProfiNet <<<<<<<'
Def Io DI_RetryLastStep = Bit,132
Def Io DI_ExitCurrCycle = Bit,133
'

'>>>>> Ausgänge - ProfiNet <<<<<<<'
Def Io ReturnErrorValue = Byte,164 'Fehlermeldung an die SPS
'

'#####
'##### Deklaration lokaler Variablen
'#####
DEF INTE FunErrorStatus
DEF INTE CurrentToolNo
'
```

```

[Montage_RV-4FL] \Programs\STORECURTOOL.mb5
'
' Programm
' #####
*RetryStep:
'>>>>> Werte initialisieren <<<<<<<<<<<<'
FunErrorStatus = 0
ReturnErrorValue = 0
JOvrd 20
Ovrd m_ProgOvrd
Accel 15,15
If ((DI_RetryLastStep = 1) Or (DI_ExitCurrCycle = 1)) GoTo *RetryStep
'
'>>>>> Aktuellen Greifertyp bestimmen <<<<<<<<<<<<'
CallP "GetCurToolNo"
CurrentToolNo = m_CurrentGripper
'
'>>>>> Prüfung des aktuellen Greifertyps <<<<<<<<<<<<'
If ((CurrentToolNo < 0) Or (CurrentToolNo > 4)) Then
    FunErrorStatus% = m_RobotToolError
EndIf
If (FunErrorStatus > 0) GoTo *ErrorLable
'
'>>>>> Aktuellen Greifer zurücklagern <<<<<<<<<<<<'
IF CurrentToolNo = m_GripperVac THEN
    pStorageTarget = P_GrpStorageVac
    pStorageFront = P_GrpStorageVac + P_ToolXZ180
    pStorageAux = P_GrpStorageVac + P_ToolX180
ENDIF
IF CurrentToolNo = m_GripperWP THEN
    pStorageTarget = P_GrpStorageWp
    pStorageFront = P_GrpStorageWp + P_ToolX180
    pStorageAux = P_GrpStorageWp + P_ToolX180
ENDIF
IF CurrentToolNo = m_GripperFuse THEN
    pStorageTarget = P_GrpStorageFuse
    pStorageFront = P_GrpStorageFuse + P_ToolX180
    pStorageAux = P_GrpStorageFuse + P_ToolX180
ENDIF
IF CurrentToolNo = m_GripperNone THEN
    pStorageTarget = P_GrpStorageWp + P_ToolX180
    pStorageFront = P_GrpStorageWp + P_ToolX180
    pStorageAux = P_GrpStorageWp + P_ToolX180
ENDIF
ENDIF

If (CurrentToolNo = m_GripperVac) Then
    CALLP "MonitorFalWS"
    If (m_IsRobotArmTray = 1) Then
        Spd m_Fast
        AUX = P_CURR ' P_CURR = aktuelle Roboterposition
        mvs AUX,-300

    Else
        Spd m_Fast
        TOOL P_tGripperNone
        Mov P_Home
    EndIf
ELSE
    Spd m_Fast
    TOOL P_tGripperNone
    Mov P_Home
    If (CurrentToolNo <> m_GripperNone) Then
        CallP "GrpOpen"
    EndIf
EndIf
Spd m_Fast
TOOL P_tGripperNone
Mov pStorageFront
Prec On
Spd m_Slow
Mvs pStorageAux
Mvs pStorageTarget
Dly 0.5
CallP "GrpRelease"
Mvs pStorageTarget, - 30
Prec Off
MOV pStorageFront
MOV P_Home
GoTo *EndOfProg
'

*ErrorLable:
ReturnErrorValue = FunErrorStatus%
If ((DI_RetryLastStep = 0) And (DI_ExitCurrCycle = 0)) GoTo *ErrorLable
If DI_RetryLastStep = 1 GoTo *RetryStep
If DI_ExitCurrCycle = 1 Then
    ReturnErrorValue = 0
    DO_MountLock = 0
EndIf

*EndOfProg:
m_ReturnedError = FunErrorStatus%

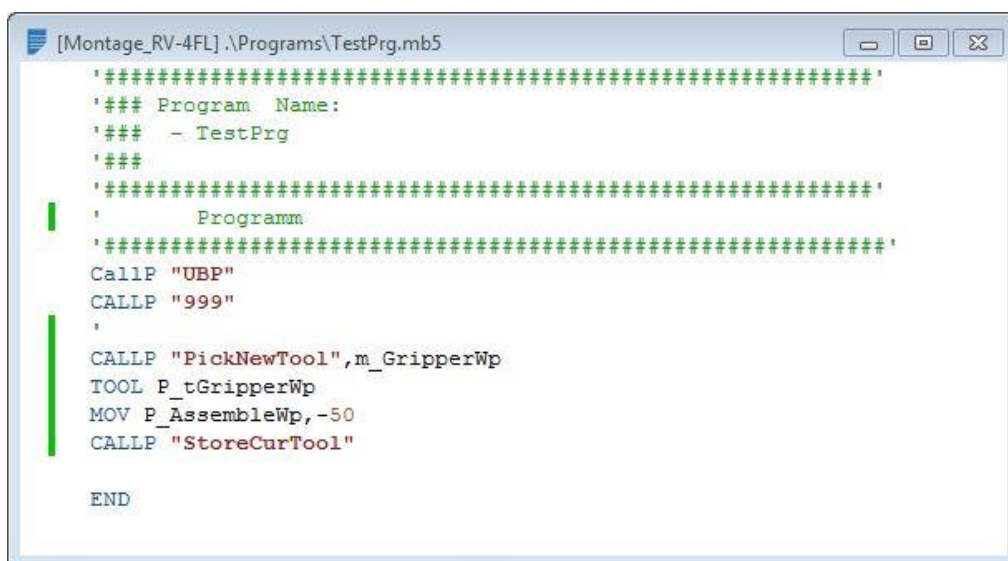
```

2. Testen Sie Ihr Programm in der Simulation. Generieren Sie alle möglichen Greifertypen für den Roboter und prüfen Sie den Fall, wenn sich der Roboter mit dem Vakuumgreifer im Bereich der Platinenbox befindet.

- a) Legen Sie ein neues Projekt A7 an und erstellen Sie zunächst ein Testprogramm (**TestPrg**) für den Fall eines Werkzeuggreifers.

Lösung:

- Öffnen Sie die Projektverwaltung und legen Sie ein neues Projekt **A7** an. Fügen Sie dem Projekt das vorbereitete leere Programm **TestPrg** und alle Programme aus der Bibliothek und die Positionsliste **UBP.pos**. Zusätzlich fügen Sie Ihr neu erstelltes Programm **StoreCurTool** hinzu.
- Ein Beispiel für das Programm **TestPrg** ist folgendes:



```

[Montage_RV-4FL] \Programs\TestPrg.mb5

'#####'
'### Program Name:
'### - TestPrg
'###
'#####'
'      Programm
'#####'
CALLP "UBP"
CALLP "999"
'
CALLP "PickNewTool",m_GripperWp
TOOL P_tGripperWp
MOV P_AssembleWp,-50
CALLP "StoreCurTool"

END
  
```

- b) Ordnen Sie dem Projekt **A7** die Steuerung **Montage_RV-4FL** zu. Kompilieren Sie das Projekt. Falls keine Syntaxfehler gemeldet werden, starten Sie die Simulation. Prüfen Sie den Ablauf und achten Sie auf mögliche Kollisionen beim Zurücklagern.

Lösung:

Der Ablauf ist einwandfrei und es treten keine Kollisionen auf.

- c) Ändern Sie das Testprogramm wie folgt:
- Ersetzen Sie die globale Variable **m_GripperWP** durch **m_GripperVac** bzw. **m_GripperFuse**.
 - Ändern Sie entsprechend den **TCP**
 - Bewegen Sie den Roboter in Abhängigkeit vom gewählten Greifer zu einem Punkt im Bereich der Platinenbox (**m_GripperVac**) bzw. zu einem Punkt in der Nähe eines Magazins mit Sicherungen.

Lösung:

- Vakuumgreifer:

.....

```
CALLP "PickNewTool", m_GripperVac
TOOL P_tGripperVac
```

```

MOV P_PCBPaletXYDir,-200
MVS P_PCBPaletXDir,-30
CALLP "StoreCurTool"
END

```

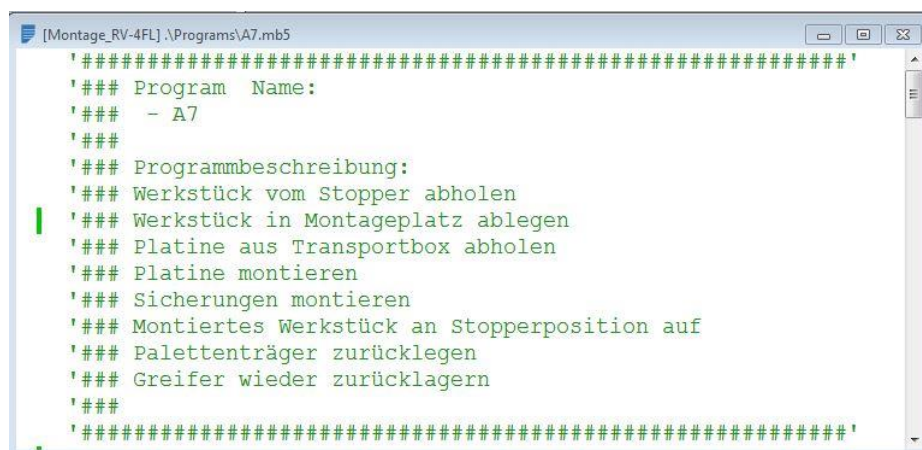
- Greifer für Sicherungen:
.....
- CALLP "PickNewTool", m_GripperFuse
TOOL P_tGripperFuse
MOV P_FuseMagazine2,-200
MVS P_FuseMagazine2,-30
CALLP "StoreCurTool"
END
- In allen Fällen ist der Ablauf korrekt und es treten keine Kollisionsfehler auf.

3. Erstellen Sie in der Simulation ein Programm A7 mit Hilfe der Funktionsbibliothek, das folgende Aufgaben erfüllt:

- Der Roboter wartet bis das Werkstück an der Stopper-Position und die Transportbox mit Platinen bereitstehen.
- Der Roboter entnimmt das Werkstück vom Palettenträger und legt es im Montageplatz ab.
- Der Roboter entnimmt die angeforderte Platine aus der Transportbox und fügt sie in das Werkstück ein.
- Der Roboter entnimmt zwei Sicherungen aus den Magazinen und setzt sie auf der Platine in der jeweiligen Sicherungsvorrichtung ein.
- Der Roboter entnimmt das montierte Werkstück aus der Montage und legt es zurück auf den Palettenträger.
- Der Roboter lagert den Greifer zurück ins Wechselsystem.

Lösung:

- Öffnen Sie die Projektverwaltung und entfernen Sie das Programm **TestPrg** aus dem Projekt **A7** und ersetzen Sie es durch das schon vorbereitete leere Programm **A7**.
- Ein Lösungsbeispiel für die obige Aufgabenstellung ist folgendes Programm:



```

[Montage_RV-4FL] \Programs\A7.mb5
'#####'
'### Program Name:
'### - A7
'###
'### Programmbeschreibung:
'### Werkstück vom Stopper abholen
'### Werkstück in Montageplatz ablegen
'### Platine aus Transportbox abholen
'### Platine montieren
'### Sicherungen montieren
'### Montiertes Werkstück an Stopperposition auf
'### Palettenträger zurücklegen
'### Greifer wieder zurücklagern
'###
'#####'

```



```

[Montage_RV-4FL] \Programs\A7.mb5

'##### Eingänge und Ausgänge #####'
'>>>>> Eingänge - ProfiNet <<<<<<<'
Def Io DI_StopperNo      = Byte,164
Def IO DI_WPatStopAvail  = Bit,251
Def IO DI_PalletInFront  = Bit,244
'
'>>>>> Ausgänge - ProfiNet <<<<<<<'
Def Io ReturnErrorValue = Byte,164 'Fehlermeldung an SPS
'
'##### Lokale Variablen #####'
'
Def Inte StopperNumber
DEF INTE FuseMagNumber
'
'##### Programm #####'
'
CALLP "UBP"
CALLP "999"

' Warten bis Werkstück in Stopperposition und Box mit Platinen
' vorhanden.
WHILE (DI_PalletInFront = 0) OR (DI_WPatStopAvail = 0)
WEND
CALLP "PCBTrayCntrl"
'
'>>>>> Werte für lokale Variablen <<<<<<<'
StopperNumber = DI_StopperNo
m_ReturnedError = 0
CallP "Initialize"      ' Initialisierung von globalen Variablen
CallP "SensorCheck"     ' Überprüfung der Sensorwerte in der Anlage
If (m_ReturnedError > 0) GoTo *EndOfPrg
'
'>>>>> Werkstück vom Stopper abholen <<<<<<<'
CallP "PickNewTool", m_GripperWP
If (m_ReturnedError > 0) GoTo *EndOfPrg
CallP "PickFrmStopR", StopperNumber
If (m_ReturnedError > 0) GoTo *EndOfPrg
'
'>>>>> Werkstück in Montageplatz einfügen <<<<<<<'
CallP "PlaceWpToAss"
If (m_ReturnedError > 0) GoTo *EndOfPrg
'

```

```

[Montage_RV-4FL] \Programs\A7.mb5
'>>>>>> Platine aus Transportbox entnehmen <<<<<<<<<'
CallP "PickNewTool", m_GripperVac
If (m_ReturnedError > 0) GoTo *EndOfPrg
CallP "PickPCBFrPal"
If (m_ReturnedError > 0) GoTo *EndOfPrg
'
'>>>>>> Platine montieren <<<<<<<<<'
CallP "MountPCB"
'
| '>>>>>> Sicherung aus Magazin entnehmen <<<<<<<<<'
FuseMagNumber = 0
CallP "GetFuseMagNo"
If (m_ReturnedError > 3) GoTo *EndOfPrg
FuseMagNumber = m_AvailFuseMagNo
CallP "PickNewTool", m_GripperFuse
If (m_ReturnedError > 0) GoTo *EndOfPrg
CallP "PickFusFrMag", FuseMagNumber
'
| '>>>>>> Sicherung in obere Halterung eindrücken <<<<<<<<<'
CallP "MountTopFuse"
'
| '>>>>>> Zweite Sicherung entnehmen <<<<<<<<<'
FuseMagNumber = 0
CallP "GetFuseMagNo"
If (m_ReturnedError > 3) GoTo *EndOfPrg
FuseMagNumber = m_AvailFuseMagNo
CallP "PickFusFrMag", FuseMagNumber
'
| '>>>>>> Sicherung in untere Halterung eindrücken <<<<<<<<<'
CallP "MountBotFuse"
'
'>>>>>> Montiertes Werkstück auf Palettenträger zurücklagern <<<<<<<<<'
CallP "PickNewTool", m_GripperWP
If (m_ReturnedError > 0) GoTo *EndOfPrg
CallP "PickWpFrmAss"
'
CallP "PlaceToStopR", StopperNumber%
If (m_ReturnedError > 0) GoTo *EndOfPrg
'
CALLP "StoreCurTool"
If (m_ReturnedError > 0) GoTo *EndOfPrg
'
'>>>>>> Fehler bei der Ausführung <<<<<<<<<'
'
*EndOfPrg

```

- Nach dem Kompilieren des Projekts **A7** kann das Programm gestartet werden und der Ablauf erfüllt alle Anforderungen.
- Speichern Sie die Programme **A7** und **StoreCurTool** zusätzlich unter den Namen **A7R** und **StoreCurToolR**. Ändern Sie die Deklarationen der Profinet-Ein- und Ausgänge in den Programmen und ändern Sie den Aufruf von **StoreCurTool** in **A7R**.

4. Führen Sie eine Inbetriebnahme des in (3) erstellten Programms für die reale Roboterstation durch.

- a) Die Arbeitszelle **CP_RobotAssembly** ist geöffnet und der PC ist mit der Robotersteuerung verbunden. Führen Sie einen Download Ihrer Programme **A7R** und **StoreCurToolR** in die Robotersteuerung durch und prüfen Sie, ob die Programme im RCI Explorer angezeigt werden.

Lösung:

- Es wird vorausgesetzt, dass die Positionen von **UBP.pos** alle schon eingelernt sind, und dass Sie mittels des Einrichtemodus am Bedienpanel einen Palettenträger mit Werkstück an die Stopper-Position im Bypass sowie die mit Platinen beladene Transportbox an die Stopposition gefahren haben.
- Führen Sie ein Download der geänderten Programme **A7R** und **StoreCurToolR** durch.

- Testen Sie **A7R** zunächst im **Debug**-Modus des RCI-Explorers.
- Starten Sie dann das Programm **A7** von der Steuerung.

Festo Didactic SE

Rechbergstraße 3
73770 Denkendorf
Allemagne



+49 711 3467-0

+49 711 34754-88500



www.festo-didactic.com

did@festo.com