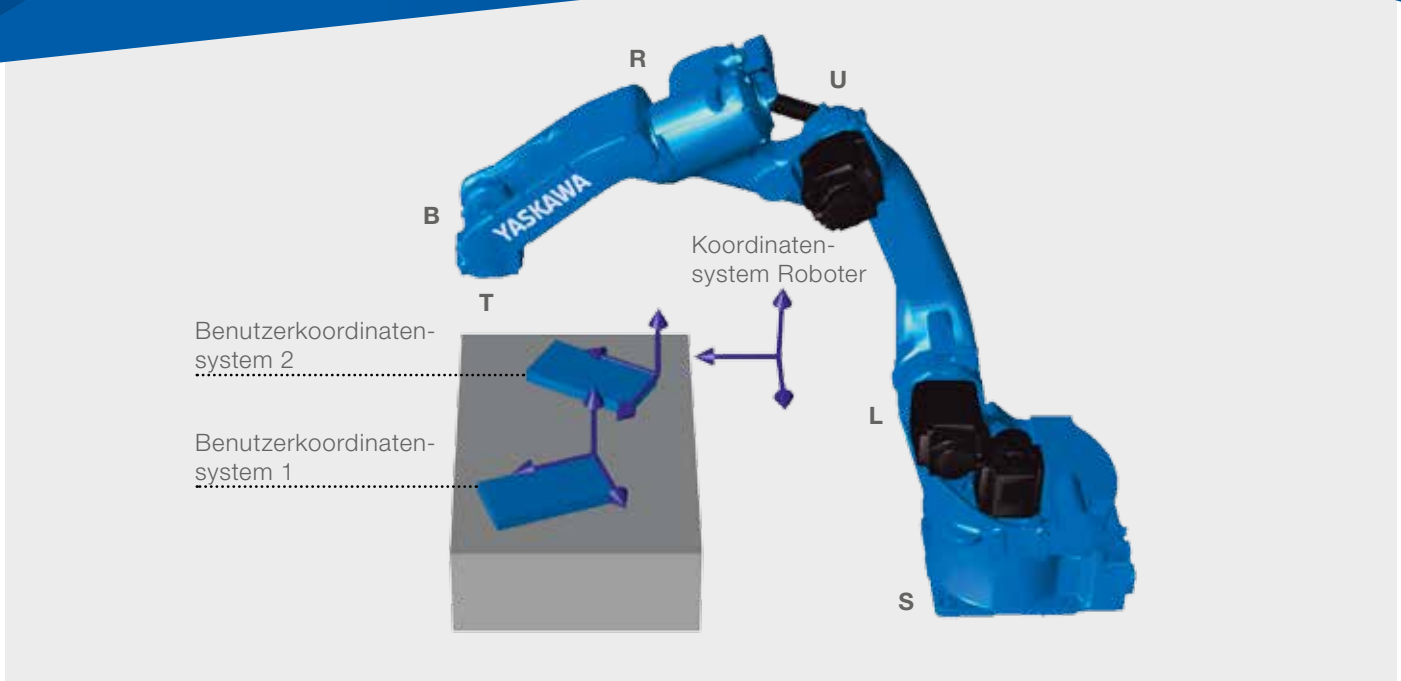


Relative Job

(195323)

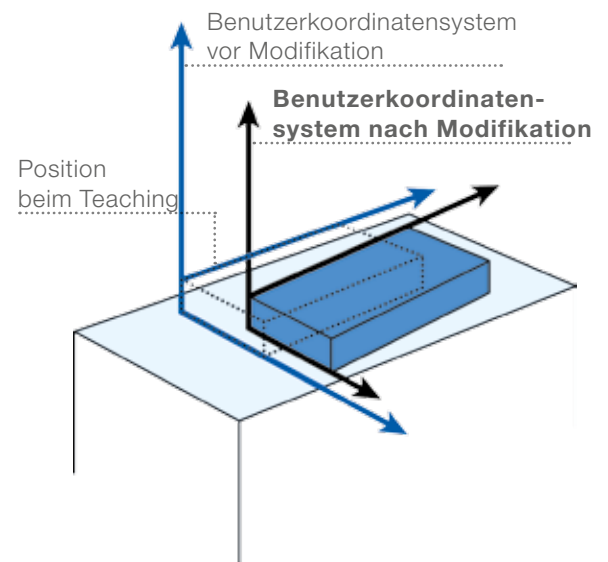


Der Roboter speichert normalerweise die Positionsdaten für den Betrieb in Form von impulsartigen Daten (Motordrehzahlimpulsbetrag von jeder Achse). Der Job (Programm), der aus den impulsartigen Daten besteht, wird „Standardjob“ genannt. Im Gegensatz zum Standardjob wird das Programm, das aus Positionsdaten in der Richtung von X, Y und Z besteht, mit Bezug auf den Ursprung in einem Koordinatensystem (wie beispielsweise Basiskoordinate und Benutzerkoordinate) „Relative Job“ genannt. Durch Konvertieren des Standardjobs wird der relative Job erstellt.

Obwohl sich die Bewegung des relativen Jobs an sich nicht von derjenigen des Standardjobs unterscheidet, verfügt der Standardjob über eine nützliche Funktion mit der Bezeichnung „Relative Job Shift“, um dieselbe Bewegung in eine andere zu verschieben. Wenn im relativen Job im Benutzerkoordinatensystem eine Benutzerkoordinate durch Ändern definierter Punkte erstellt wird, dann kann die zu der geänderten Koordinate verschobene Bewegung bei der Ausführung der Anweisung ausgeführt werden.

VORTEILE IM ÜBERBLICK

- Ändern eines impulsartigen Jobs in einen koordinierten Job
- Alle Punkte sind gespeichert in X, Y, Z
- Mit der Funktion „Relative Job Shift“ können vollständige Jobs in unterschiedliche Rahmen verschoben werden



YRC1000
micro

YASKAWA

www.yaskawa.eu.com

Macro Command

(195339)

Beispiel für Makro-Befehls-Registry: SEALON

Line Step

```
000 NOP
001 MOVJ VJ 100.0
002 WAIT IN#(1) ON
003 MOVJ VJ 50.0
004 SEALON WIDTH=8
005 MOVL V=125
006 MOVL V=95
```

Beispiel für Makro Job-Registry: SEALON

Line Step

```
000 NOP
001 GETARG LI000 IARG#(1) : Store 1st argument data '8' to LI000.
002 DOUT OT#(10) ON : Turn ON general output 10.
003 MUL LI000 10 : Multiply the number of LI000 by 10. 8x10 80
004 WAIT N#(10) ON : Wait for general input 10 to be ON.
005 AOUT AO#(1) LI000 : Output '80' to the analog output 1.
006 END
```

Die Funktion „Macro Command“ dient zur Erzeugung der Registrierung und Ausführung einer Anweisung von mehreren INFORMs, die jedem System entspricht.

Ein Argumenten-Tag kann einem Makrobefehl wie eine normale Anweisung hinzugefügt werden. Makrobefehle und ihre Argumenten-Tags sind leicht und beliebig in der Jobdatenanzeige programmierbar.

Verfahrensweise für die Erstellung von Anweisungen: Erstellen Sie einen Makrojob (im normalen Teaching-Verfahren), und registrieren Sie den Makrojob als eine Makroanweisung. (Im Einstellungsbildschirm Makroanweisungen)

Anweisung, das Argument für die Makroanweisung (GETARG) zu erhalten: Wenn die Makroanweisung ausgeführt wird, dann werden zu der Makroanweisung hinzugefügte Argumentdaten

erhalten und in die spezifizierte lokale Variable hinein gespeichert, um die Daten im Makrojob zu verwenden.

Wenn eine Nachverarbeitung erforderlich ist, wenn ein Makrobefehl unterbrochen wird, dann ist es möglich, dem Programm einen Unterbrechungs-Makrojob (SUSPEND JOB) hinzuzufügen.

Der Unterbrechungs-Makrojob wird ausgeführt, wenn ein Makrobefehl, der den Unterbrechungs-Makrojob enthält, unterbrochen wird, indem er in einen Status wie Halten, Not-Aus oder Modusumschaltung versetzt wird.

Ausführungsbefehle wie z.B. „JUMP“, „CALL“ und „PSTART“ können nicht als Makrojob registriert werden. Außerdem werden TIMER-Anweisung und WAIT-Anweisung nicht als Unterbrechungs-Makrojob ausgeführt.

VORTEILE IM ÜBERBLICK

- Schaffung eigener Funktionen und Routinen in der INFORM-Liste

YRC1000
micro

YASKAWA

www.yaskawa.eu.com

Macro Name with up to 16 Characters

(195377)

```
JOB CONTENT: MASTER
J:MACRO_JOB_WITH_MAX_16_CHARACTERS S:0000
CONTROL GROUP: R1 TOOL: **
0000 NOP
0001 .
0002 STD-NAME Arg1=5 Arg2=-4 Arg3=70000 Arg4=3.14
0003
0004 END
```

Buttons: IN/OUT, SAME, CONTROL, PRIOR, DEVICE, MOTION, ARITH, SHIFT, SENSOR, OTHER, MACRO

Buttons: STD-NAME, LONG_MACRO_NAME

LONG_MACRO_NAME Arg1=5 Arg2=-4 Arg3=70000

```
JOB CONTENT: MASTER
J:MACRO_JOB_WITH_MAX_16_CHARACTERS S:0000
CONTROL GROUP: R1 TOOL: **
0000 NOP
0001 .
0002 STD-NAME Arg1=5 Arg2=-4 Arg3=70000 Arg4=3.14
0003
0004 LONG_MACRO_NAME Arg1=5 Arg2=-4 Arg3=70000 Arg4=3.14
0005
0006 END
```

MOVJ VJ=0.78

Bei der „Standard-Makrofunktion“ können max. 8 Zeichen für den Makronamen verwendet werden. Häufig müssen Namen gekürzt werden, wodurch diese jedoch schlechter lesbar sind.

Diese Funktion ist ein Add-On für Makrofunktionen und erweitert die Zahl auf maximal 16 Zeichen.

VORTEILE IM ÜBERBLICK

- Einfaches Benennen von Makrojobs
- Bessere Lesbarkeit von Programmzeilen mit Makro

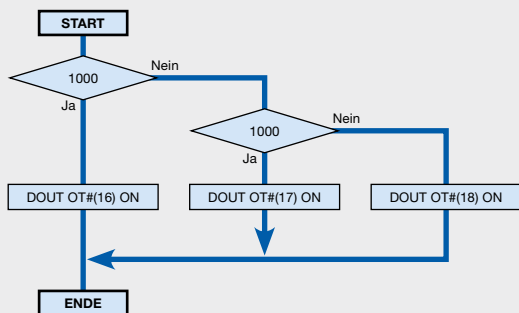
YRC1000
micro

Structured Language

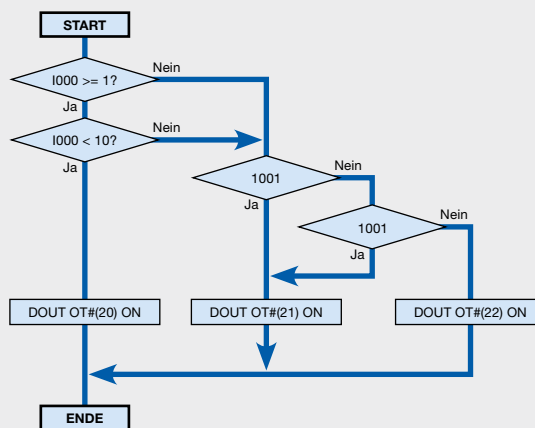
(195342)

- (1) IF I000 = '0', set OT#(16) ON
- (2) IF I000 = '1', set OT#(17) ON
- (3) IF I000 is other than '0' or '1', set OT#(18) ON

```
SWITCH I000 CASE 0
  DOUT OT#(16) ON
CASE 1
  DOUT OT#(17) ON
DEFAULT
  DOUT OT#(18) ON
ENDSWITCH
```



```
IF (I000 >= 1 AND I000 < 10) THEN
  DOUT OT#(20) ON
ELSEIF (I001 = 1 OR I001 = 2) THEN
  DOUT OT#(21) ON
ELSE
  DOUT OT#(22) ON
ENDIF
```



Mit der Funktion „Structured Language“ ist es möglich, klar arrangierte Programme auf der Basis einer hohen Sprach-ebene mit den folgenden Anweisungen zu erstellen:

- IFTHEN-ELSEIF-ELSE-ENDIF
- WHILE-ENDWHILE
- FOR-NEXT
- SWITCH-CASE-ENDSWITCH

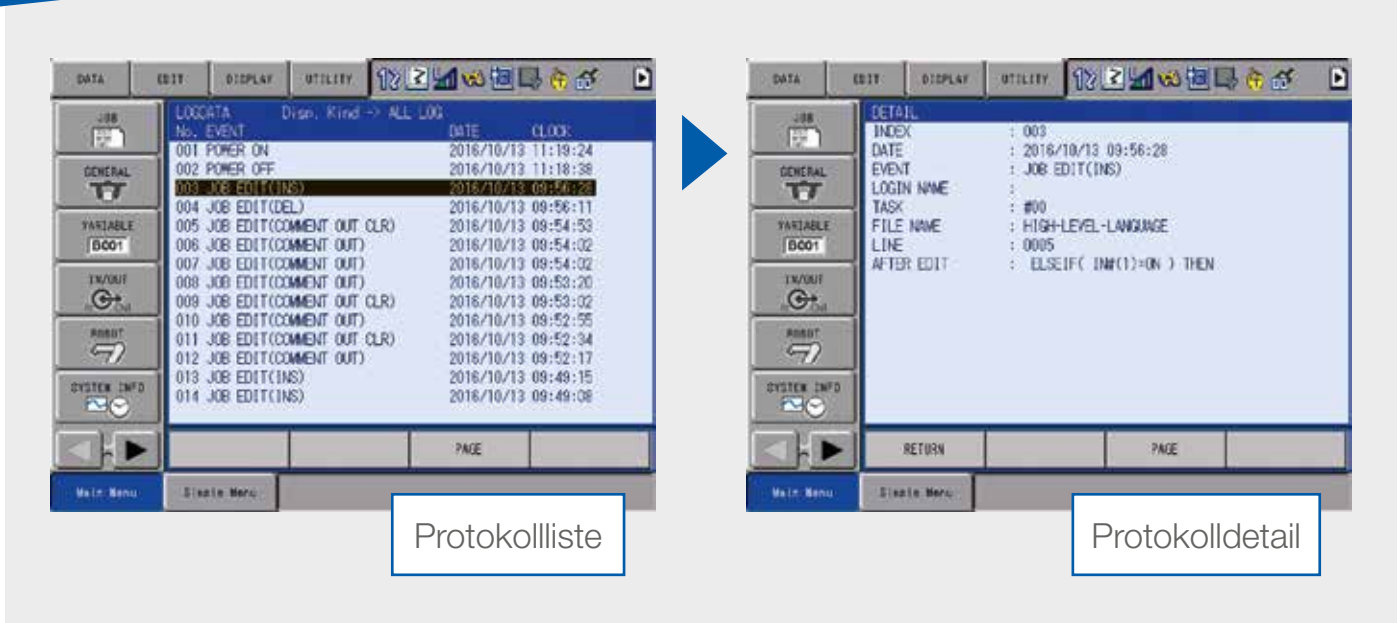
VORTEILE IM ÜBERBLICK

- Programme kürzen
- Strukturierte Programmübersicht
- Komplexe Funktionalität, leicht implementierbar

YRC1000
micro

Logging Function

(195343)



Diese Funktion speichert und zeigt die Bearbeitung wie z.B. Bearbeitungsdaten des Robotersteuerungsprogramms (Job) und die Schweißbedingung sowie die Historie (Protokoll) des Betriebs des Programmierhandgeräts wie z.B. Jobausführung an.

Diese Funktion erleichtert auch die Fehlerbeseitigung, indem sie die Datennachverfolgbarkeit sicherstellt.

Speichert 200 Datenbearbeitungsdateien (parallele Job-E/A-Verknüpfung, verschiedene Bedingungsdateien, Bearbeitung von Parametern, Variablen usw.).

Speichert 200 Handgerätevorgänge (Jobstart, Halten, Not-Aus, Laden und Speichern externer Speicher usw.).

Es ist auch möglich, nur das Protokoll der Datenbearbeitung oder des Betriebs des Handgeräts zu extrahieren und anzuzeigen.

VORTEILE IM ÜBERBLICK

- Programmierhistorie kann bei laufendem Betrieb überprüft werden
- Einfache Fehlerdiagnose

YRC1000
micro

Job Editing during Play

(195345)

CHANGE OF INTERLOCK CONDITION

(Before change)	(After change)
NOP	NOP
:	:
MOVJ VJ=50.00	MOVJ VJ=50.00
WAIT IN#(1)=ON	WAIT IN#(10)=ON
	TIMER T=0.5
MOVJ VJ=50.00	MOVJ VJ=50.00
MOVJ VJ=50.00	MOVJ VJ=50.00
:	:
END	END

Betriebszustand kann geändert werden, ohne die Arbeit des Roboters anzuhalten.

CHANGE OF WELDING CONDITION NUMBER

(Before change)	(After change)
NOP	NOP
:	:
MOVJ VJ=50.00	MOVJ VJ=50.00
ARCON ASF#(1)	ARCON ASF#(2)
MOVJ VJ=50.00	MOVJ VJ=50.00
:	:
END	END

Schweißbedingung kann geändert werden, ohne die Arbeit des Roboters anzuhalten.

Selbst bei laufendem Betrieb der Linie ist Jobbearbeitung möglich.

Ein Job kann selbst dann bearbeitet werden, wenn der Roboter produziert, oder bei laufendem Betrieb der Linie, Feineinstellung wie z.B. Änderungen von Signalnummern, Betriebsbedingungen und Arbeitsbedingungen ohne Anhalten der Arbeit des Roboters.

VORTEILE IM ÜBERBLICK

- Zykluszeit kann verringert werden
- Betriebsbedingungen können ohne Anhalten während seiner Arbeitszeit geändert werden

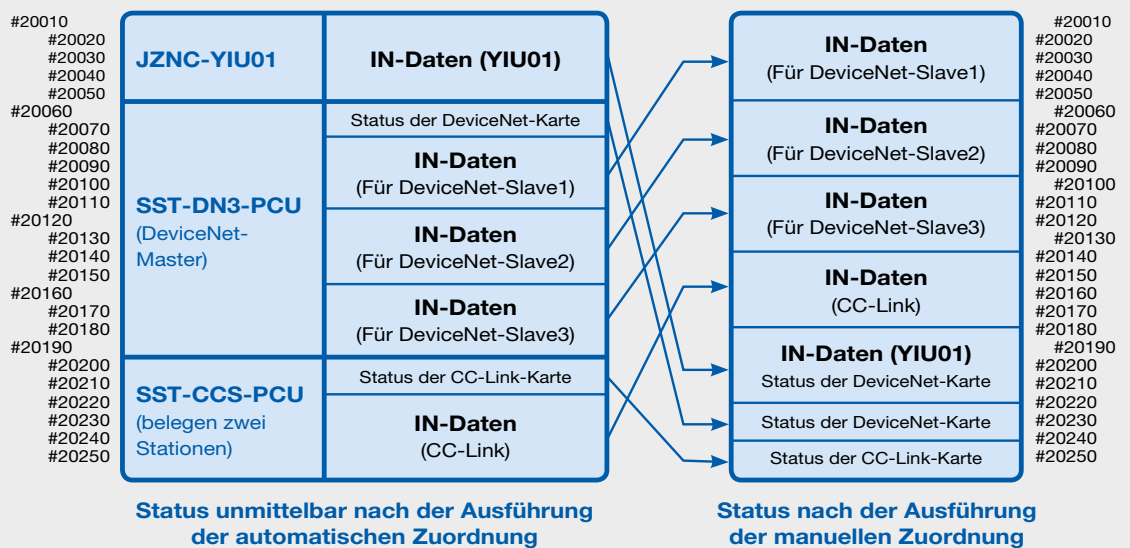
YRC1000
micro

External IO Allocation

(195346)

EXTERNAL IO ALLOCATION (INPUT)						
	STB	CH	MAG ID	ADDR	BYTE	NAME
#20010	14	0	0	0	5	YIU01
#20060	16	0	254	0	1	DN3-PCU-1
#20070	16	0	1	1	5	DN3-PCU-1
#20120	16	0	2	6	4	DN3-PCU-1
#20160	16	0	3	10	3	DN3-PCU-1
#20190	17	0	254	0	1	CCS-PCU
#20200	17	0	0	1	6	CCS-PCU

E/A-Abbildung für jede E/A- oder Feldbuskarte möglich:



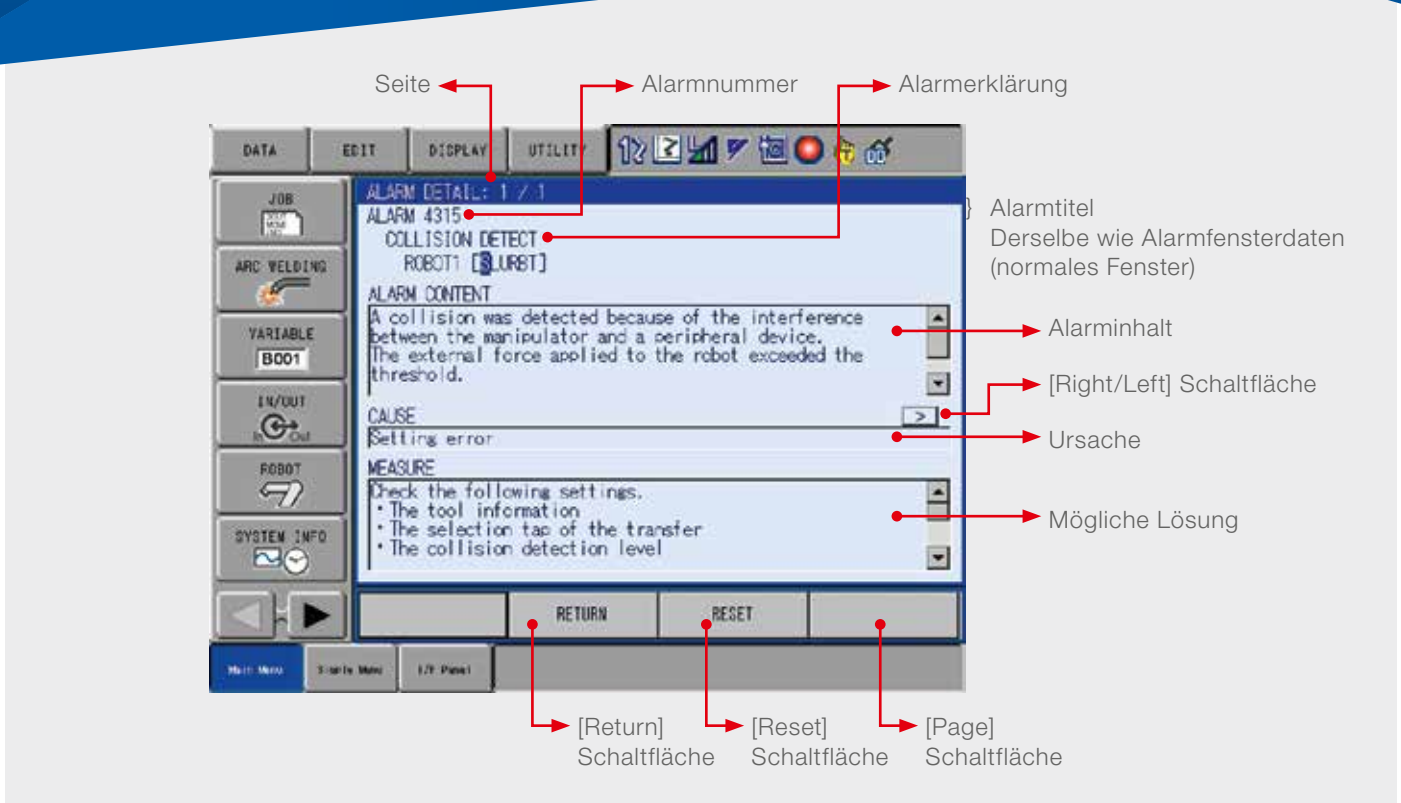
VORTEILE IM ÜBERBLICK

- E/A-Bereich für jede E/A-Karte oder Feldbus (ProfiNet, DeviceNet usw.) ist anwenderspezifisch anpassbar
- Leicht erstellbare E/A-Schnittstelle je nach Anwendung oder Kundenanfragen

YRC1000
micro

Alarm Detail Displaying

(195349)



Benutzer können eine Einstellung vornehmen, damit das Fenster bei Auftreten eines Alarms angezeigt wird:

- ALARM-Fenster (Standard)
=> Drücken Sie [SELECT], um das Fenster ALARM DETAILS anzuzeigen
- ALARM DETAIL (Parametersatz)

Zeigt mögliche Gründe für das Auftreten des Alarms in absteigender Häufigkeitsreihenfolge an um Benutzern zu helfen, die Gegenmaßnahme für jede Ursache sofort zu finden.

Wenn es einige mögliche Gründe und Gegenmaßnahmen für einen Alarm gibt, drücken Sie die Schaltfläche [Left/Right] oder die Tasten [Left/Right] um die nächste „Ursache“ und die nächste „Maßnahme“ zu sehen. „Ursache“ und „Maßnahme“ werden in der auszuführenden Reihenfolge angezeigt.

Wenn mehrere Alarmer gleichzeitig auftreten, drücken Sie die Schaltfläche [Page] oder die Taste, um zum nächsten Alarm zu wechseln.

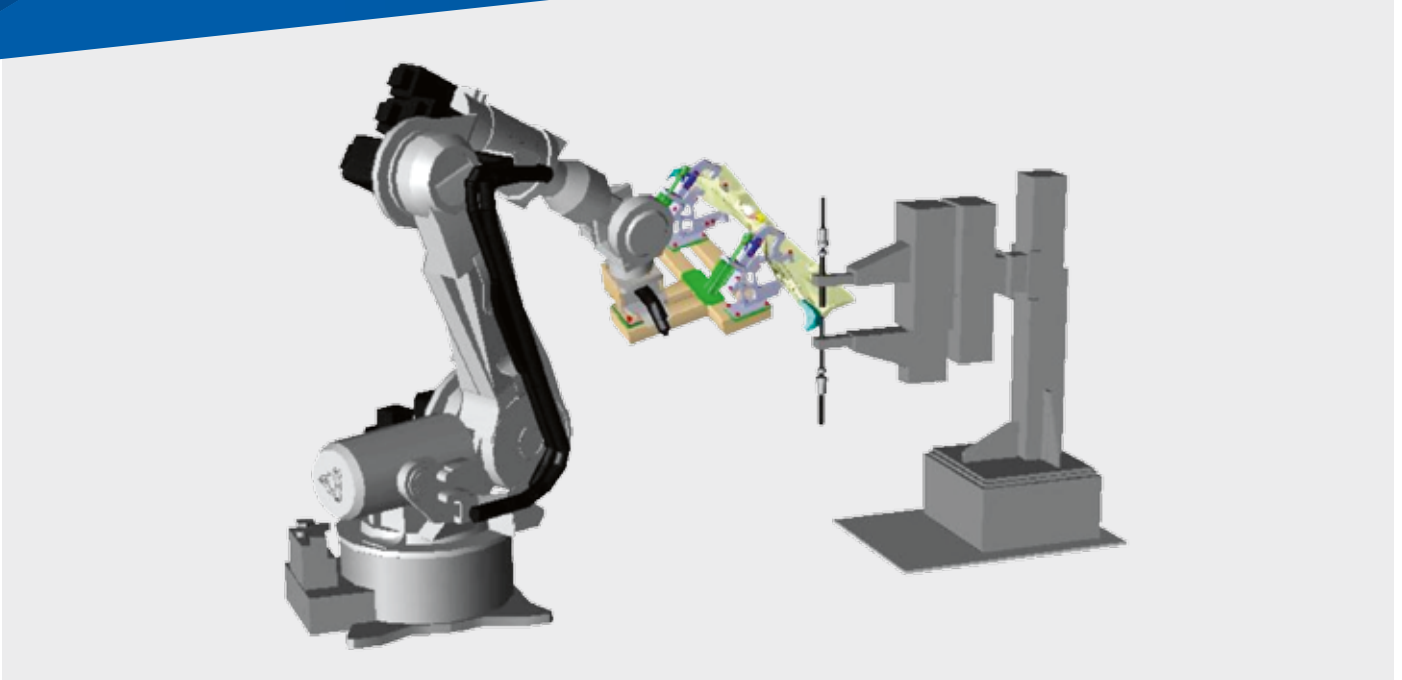
VORTEILE IM ÜBERBLICK

- Zeigt mögliche Ursachen für auftretende Alarmer an
- Hilft dem Benutzer, die Ursache von Fehlern sofort zu finden

YRC1000
micro

External Reference Point

(195324)



Die Steuerungsfunktion „External Reference Point“ führt Teaching und Playback aus, wobei ein Punkt im Raum als Steuerungspunkt des Roboters dient. Dieser eine Punkt im Raum wird als External reference point bezeichnet. Dieser Punkt wird insbesondere in den Fällen verwendet, in denen die Abdichtung im Werkstück zur Unterstützung von Arbeits- oder Punktschweißen mit dem ortsfesten Werkzeug ausgeführt wird. In diesem Fall kann ein Vorgang wie z.B. Änderung der Stellung eines Werkstücks leicht durch Einstellen der Werkzeugspitze, Düse oder Zange auf den Referenzpunkt ausgeführt werden. Natürlich wird während des Playbacks eine Interpolation durch Steuerung der relativen Geschwindigkeit und der relativen Position zwischen dem externen Referenzpunkt und dem Werkstück ausgeführt.

VORTEILE IM ÜBERBLICK

- Die Qualität wird selbst für Werkstücke verbessert, die die Arbeit mit einem festen Werkzeug unterstützen
- Teaching-Zeit kann verringert werden.
- Anwendbar für mehr als ein Werkzeug, eine Zange oder Düse (bis zu 63 externe Referenzpunkte)

YRC1000
micro

System Job

(195384)

Startet automatisch nach dem Einschalten der Stromversorgung des Schaltschranks und läuft im Hintergrund.

System JOB

- *Start
- DELAY 100
- DIN LB000 SOUT#(441)
- DIN LB001 SOUT#(37)
- AND LB000 LB001
- JUMP *Start IF LB000=1
- SETUALM 8000 „Alarm“
- JUMB *Start

Work JOB

- MOVJ VJ = 100
- MOVJ VJ = 100
- MOVJ VJ = 100
- MOVJ VJ = 100

Startet, wenn Sie in den Play/Remote Modus gehen und Start drücken.

Eine Systemaufgabe wird unmittelbar und automatisch nach dem Start gestartet, wenn die Steuerung gestartet ist. Im Vergleich zu einer normalen Aufgabe (Benutzeraufgabe) ist weder Start noch ein Stopp erforderlich, um eine solche Aufgabe zu steuern. So ist sie unabhängig vom Betriebsmodus (Play/Teach/Operating) und vom Status Servo ein. Während normalerweise Bewegungs- oder Vorgangssequenzen (Workflow) in normalen Aufgaben programmiert werden, wird eine Systemaufgabe verwendet, um zyklische Aufgaben im Hintergrund zu verarbeiten. Eine Systemaufgabe kann registriert werden und gleichzeitig zusätzlich zu normalen Jobaufgaben durchgeführt werden.

In einer Systemaufgabe laufende Programme werden Systemjobs genannt. Ein Systemjob ist einem normalen Job (Benutzerjob) sehr ähnlich und verwendet dieselbe Inform-Programmiersprache, jedoch mit einem begrenzten Anweisungssatz. So sind z.B. Bewegungsbefehle nicht verfügbar. Andererseits wird sie aufgrund ihrer ähnlichen Merkmale manchmal als Ersatz für das interne Verknüpfungsprogramm

verwendet. Während der Zugriff auf ein Verknüpfungsprogramm auf E/As und Register begrenzt ist, können Systemjobs alle verfügbaren Ressourcen wie Variablen, Systemvariablen, E/As und Register verarbeiten. Andererseits ist die Zykluszeit einer Systemaufgabe im Gegensatz zu einem Verknüpfungsprogramm nicht fest. Somit erhöht sich die Verarbeitungszeit mit jeder zusätzlichen Codelinie, und ist von der aktuellen CPU-Last abhängig.

Beispiel:

- Sie können einen Systemjob mit einer permanenten Überprüfung des Kollisionserkennungsfunktionsstatus programmieren.
- Wenn Sie sich im Teaching-Modus befinden und die Kollisionserkennung AUS ist, passiert nichts. Wenn Sie sich aber im Play-Modus befinden und die Kollisionserkennung AUS ist, dann resultiert dies in einem Alarm, z.B. mit einer Erinnerung „Please turn ON collision detection while PLAY mode“.

VORTEILE IM ÜBERBLICK

- Zyklische Hintergrundaufgaben in der Inform-Sprache
- Autostart und unabhängig vom Betriebsstatus
- Zugriff auf alle Ressourcen
- Zykluszeit ist abhängig von den Zahlencodelinien und der CPU-Last

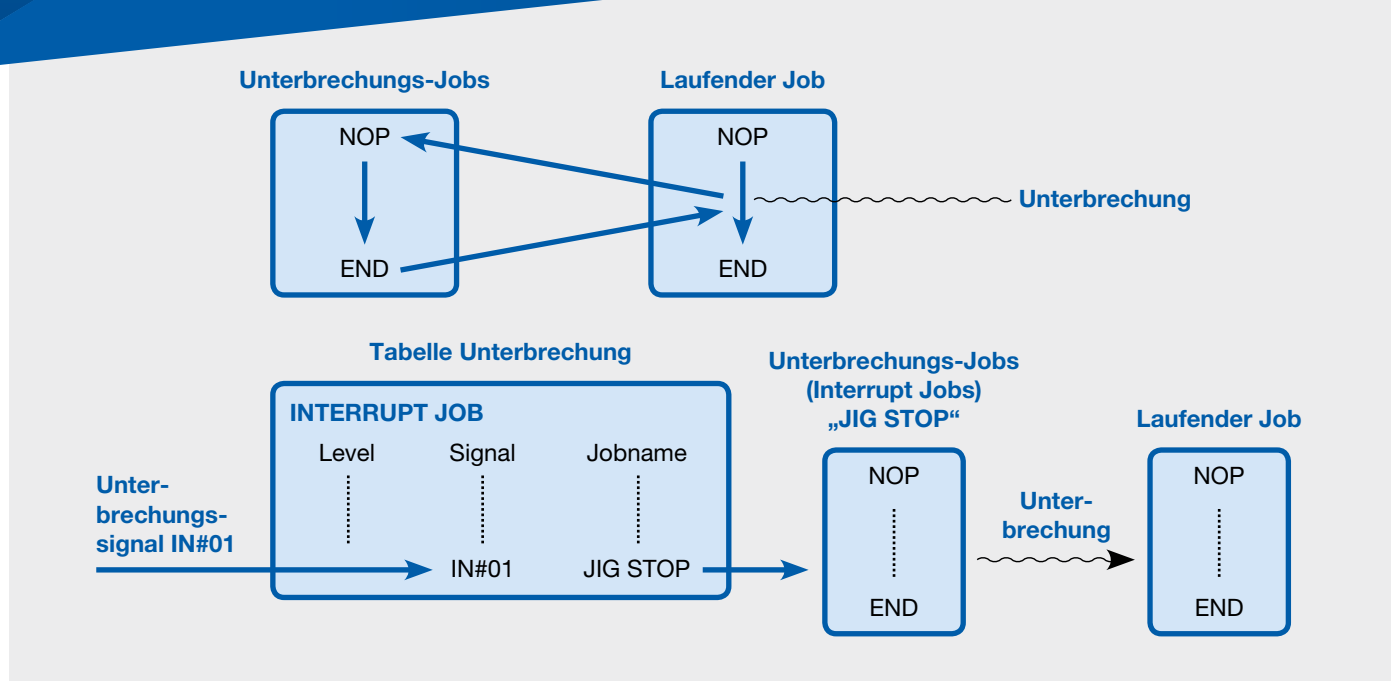
YRC1000
micro

YASKAWA

www.yaskawa.eu.com

Interrupt Job

(195330)



Die Funktion „Interrupt Job“ ist eine Art von Aufrufjob. Wenn ein Signal zur Unterbrechung des Jobs von einem Zusatzgerät oder einem anderen System gesendet wird, dann setzt diese Funktion momentan einen in Bearbeitung befindlichen Job aus, und führt den Job entsprechend dem Signal aus.

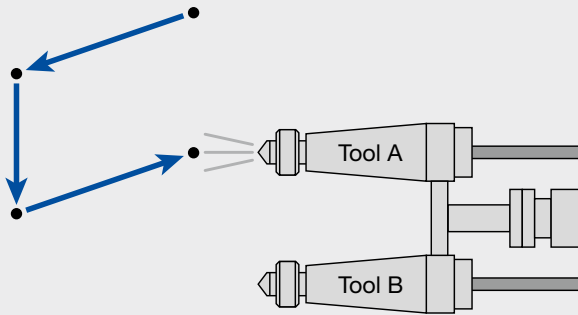
Diese Funktion ist nützlich, wenn ein Fehler in einem Zusatzgerät oder in einem anderen System auftritt, oder wenn der Roboter in einer Notsituation abgezogen werden sollte.

Das Senden eines in der Unterbrechungstabelle spezifizierten Eingangssignals ruft einen Job entsprechend dem Signal (IN#1 => job A; IN#2 => job B... usw.) auf. Wenn der Interrupt Job abgeschlossen ist, dann wird der ausgesetzte Job erneut von der Anweisungsline gestartete, auf der sich der Cursor zum Zeitpunkt der Unterbrechung befand. Die Anweisungen EI (Interrupt enabled) und DI (Interrupt disabled) sind notwendig, um die Unterbrechungsüberwachung zu aktivieren.

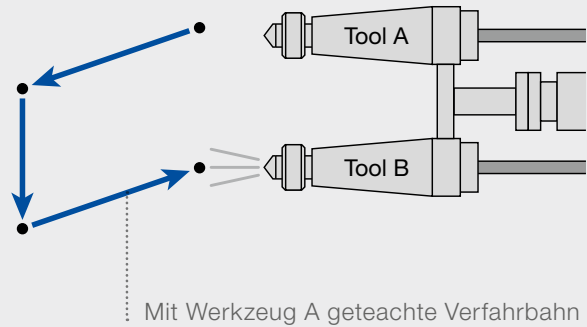
VORTEILE IM ÜBERBLICK

- Unterbrechungen sind möglich während Bewegungsanweisungen sowie Timer-Anweisungen
- Taktzeit kann verringert werden
- Sie können einen Bereich von Ihrem Job beobachten
- Für jeden Roboter können Sie 8 Interrupt Jobs erzeugen

YRC1000
micro



1. Nach dem Teaching mit Werkzeug A tragen Sie die Grundierung mit Werkzeug A auf



2. Mit der TCP Funktion tragen Sie die Deckschicht mit Werkzeug B auf

Die Funktion „TCP“ (TCP: Tool Center Point) bewegt das Werkzeug entlang der Bahn, die mit einem anderen Werkzeug geteacht wurde.

Schnelle Programmierung von Jobs (Beispiel):

- Der von dem mit Werkzeug A geteachten Job kopierte Job, der als „Job A“ definiert ist, kann als „Job B“ definiert werden. Fügen Sie die TCP-Anweisungen vor und nach den Abschnitten hinzu, in denen Werkzeug B in Job B zu verwenden ist. Setzen Sie eine Werkzeugdateinummer für Werkzeug B in der „TCPON“-Anweisung.
- Ein Werkzeug wird für den Grundierungsanstrichvorgang verwendet, das andere wird für den Deckschichtenanstrichvorgang verwendet. Teachen Sie einem der zwei Werkzeuge eine Verfahrbahn, dann ist das Teachen für das andere Werkzeug nicht notwendig.

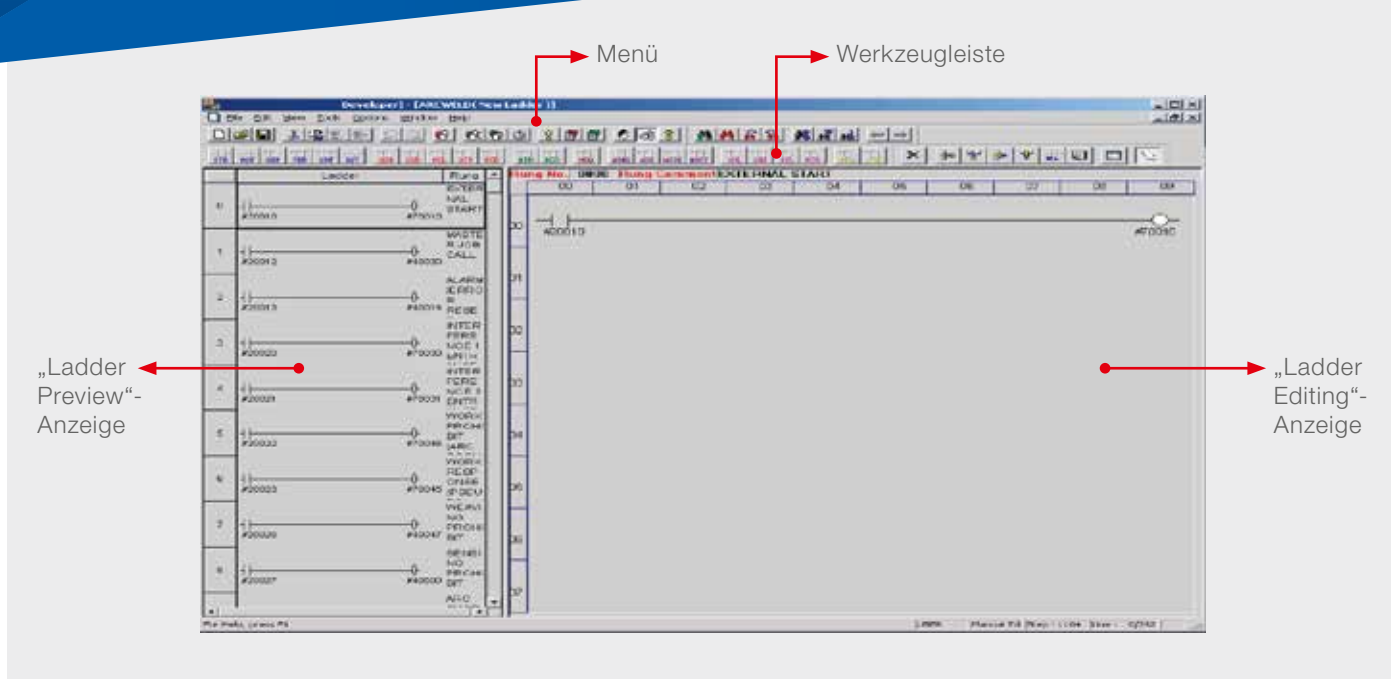
VORTEILE IM ÜBERBLICK

- Verringerung der Programmierungszeit:
Beim Bewegen von Doppelwerkzeugen entlang derselben Bahn wird nur einmaliges Teaching bei jedem Werkzeug angewandt, was eine Verringerung der Joberzeugungszeit ermöglicht
- Präzise Spur beider Werkzeuge

YRC1000
micro

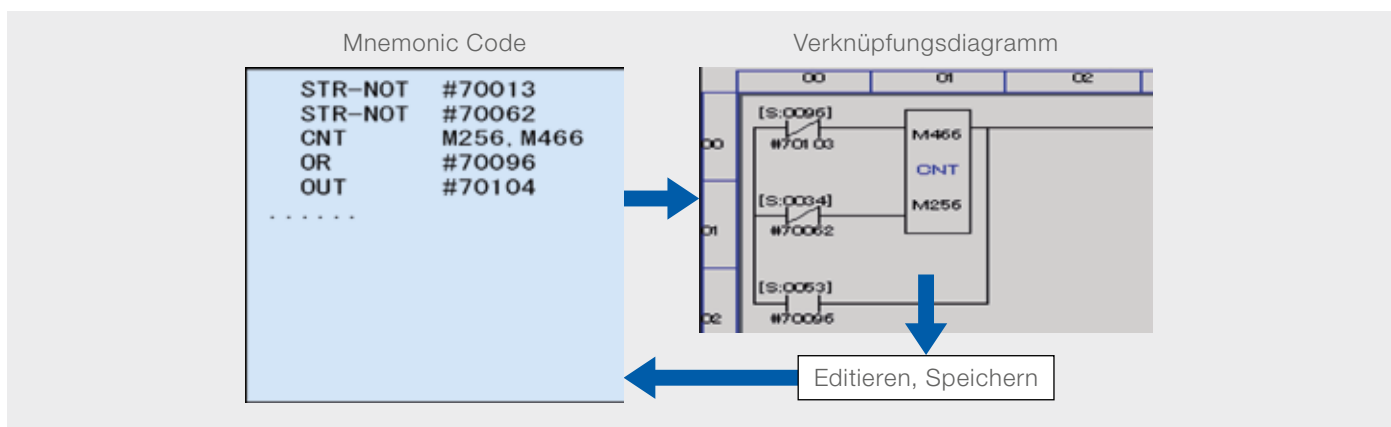
Ladder Editor

(195365)



Die Robotersteuerung verfügt über eine parallele E/A-Funktion, die eine E/A-bezogene Steuerung unabhängig von dem Roboter, und parallel mit dem Roboterbetrieb verarbeitet. Verknüpfungssoftware wird verwendet, um Verknüpfungs-

programme mit der obigen Funktion als ein Signalverbindungsdiagramm anzuzeigen oder um Verknüpfungsprogramme leicht zu bearbeiten, wie z.B. Kopieren von Kommandoteilen durch einen Dragund Drop-Vorgang mit der Maus.



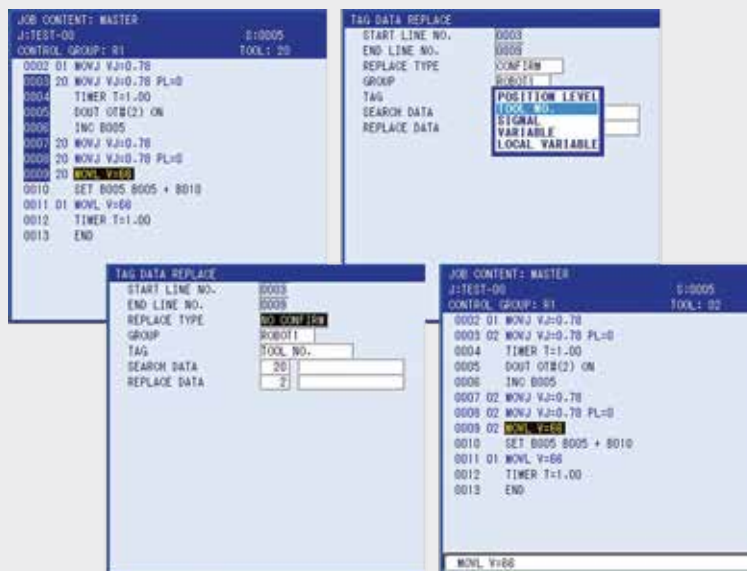
VORTEILE IM ÜBERBLICK

- Einfacher Drag- und Drop-Vorgang
- Funktion Cross reference wird angereichert
- Funktion History jump ist verfügbar
- Druckfunktion wird angereichert

YRC1000
micro

Tag Data Replace

(195354)



Diese Editierungsfunktion ermöglicht Ihnen die Änderung von Daten-Tags in einer oder mehreren Linien eines Jobs zur gleichen Zeit.

Änderbare Linien-Tags:

- Positionslevel-Nummern
- Werkzeugnummern
- Signalnummern
- Variable Nummern
- Lokale variable Nummern

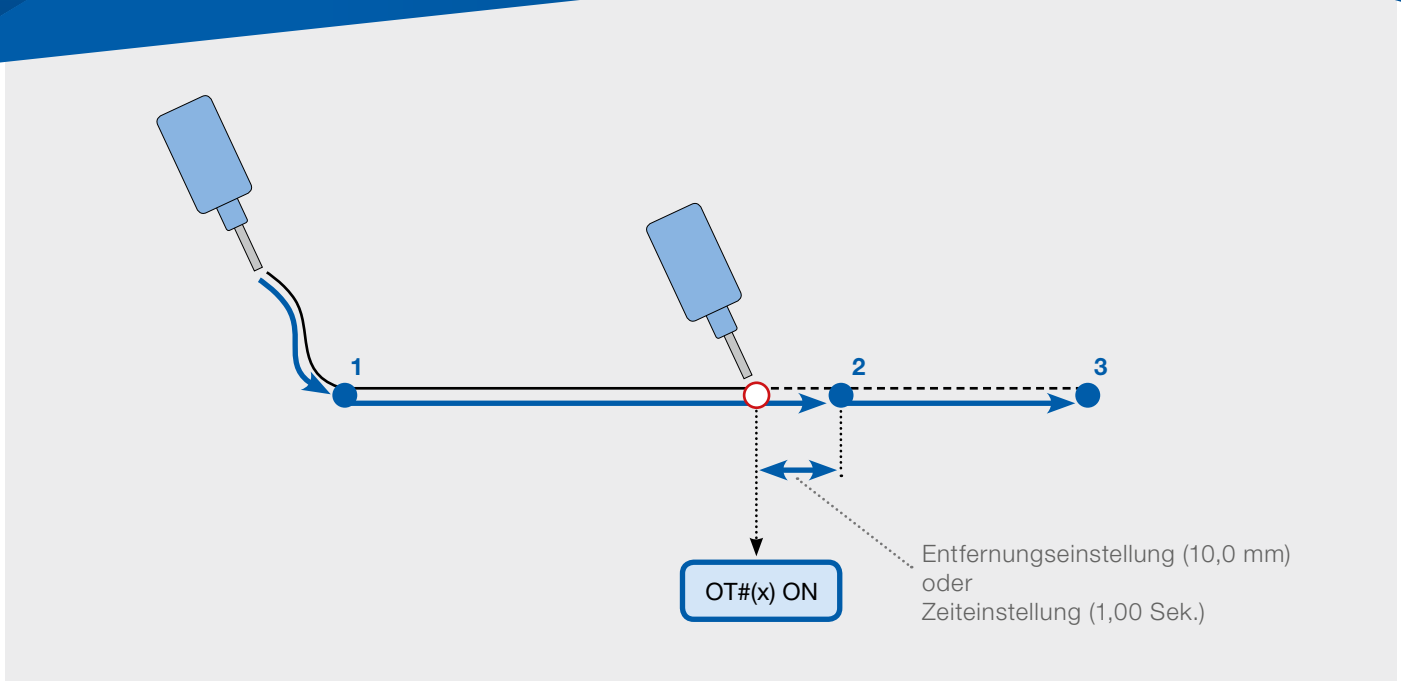
VORTEILE IM ÜBERBLICK

- Die Teaching-Modifikationszeit wird bedeutend verringert
- Tag-Daten können in dem Fenster auf dem Programmierhandgerät leicht korrigiert werden

YRC1000
micro

IO Output Timing Control

(195352)



Die Funktion „IO Output Timing Control“ ermöglicht dem Benutzer das Schalten eines Ausgangs in Abhängigkeit von der Entfernung oder Zeit im Verhältnis zu einem Punkt.

Diese Anweisung wird direkt hinter dem zugehörigen Bewegungs-Tag hinzugefügt.

Die Anweisung ist geschwindigkeitsabhängig.

VORTEILE IM ÜBERBLICK

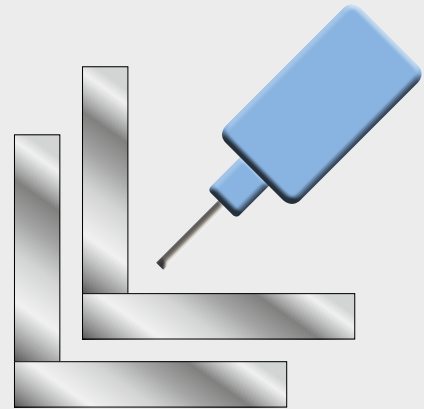
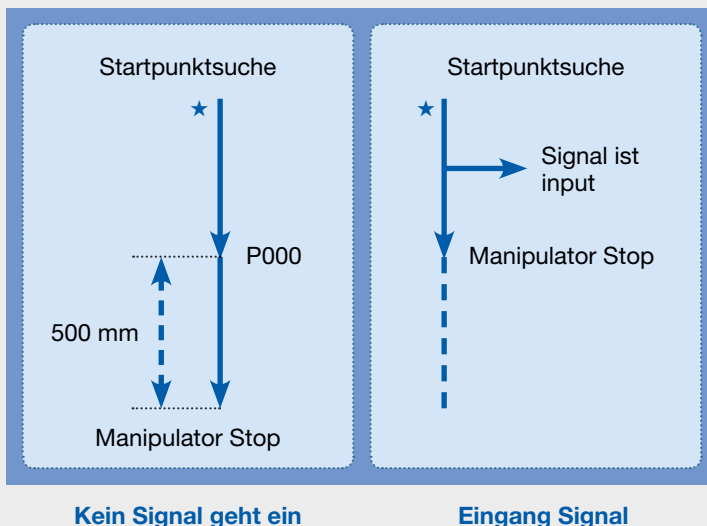
- Präzise E/A-Zeitsteuerung je nach Roboterposition
- Geschwindigkeitsunabhängig

YRC1000
micro

Search Function

(195356)

<Robot Operation>



Diese Suchfunktion verwendet allgemeine Erkennungssensoren, um den Roboter durch das Erkennungssignal zu stoppen und den Roboter die nächste Arbeit ausführen zu lassen. Das heißt, die Funktion sucht das zu bearbeitende Ziel.

Von der Startposition (★ Marke) bis zu der von der Variablen P000 definierten Position arbeitet der Roboter mittels linearer Interpolation mit einer Geschwindigkeit von 10,0 mm/Sek. Gleichzeitig beginnt die Funktion, nachdem der Roboterbetrieb startet, mit der Überwachung, ob ein Signal in DIRECT IN Nr. 1 eingegeben wird.

Nach dem Beginn des Betriebs stoppt der Roboter, sobald es einen Signaleingang gibt. Wenn es keinen Signaleingang gibt, dann stoppt der Roboter in einer Entfernung von 500 mm vor der von P000 definierten Position. In diesem Fall kann durch den Wert des System-Byte-Status \$B02 bestimmt werden, ob der Roboter mit oder ohne einen Signaleingang stoppt.

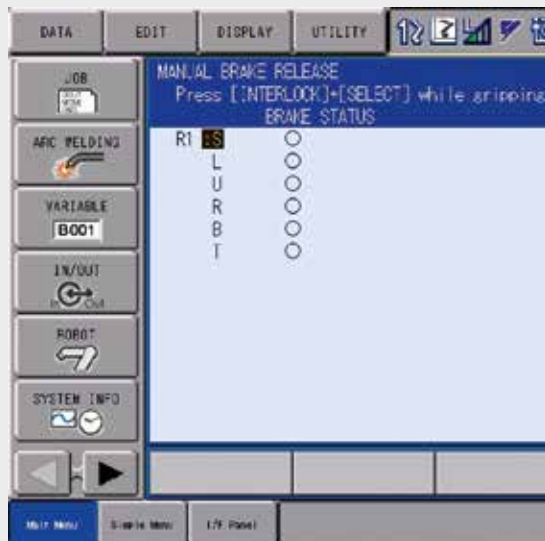
VORTEILE IM ÜBERBLICK

- Obwohl normale Programme ein Teaching-Verfahren für jedes Werkstück erfordern, ermöglicht diese Funktion dem Roboter, sich auf der Suche nach Werkstücken zu bewegen, was das Programm vereinfacht

YRC1000
micro

Manual Brake Release

(195348)



Die Funktion „Manual Brake Release“ ermöglicht die Zwangsöffnung jeder Motorbremse des Manipulators und der externen Achsen mithilfe des Programmiergeräts.

Roboter und externe Achsen können ohne Stromanschluss bewegt werden.

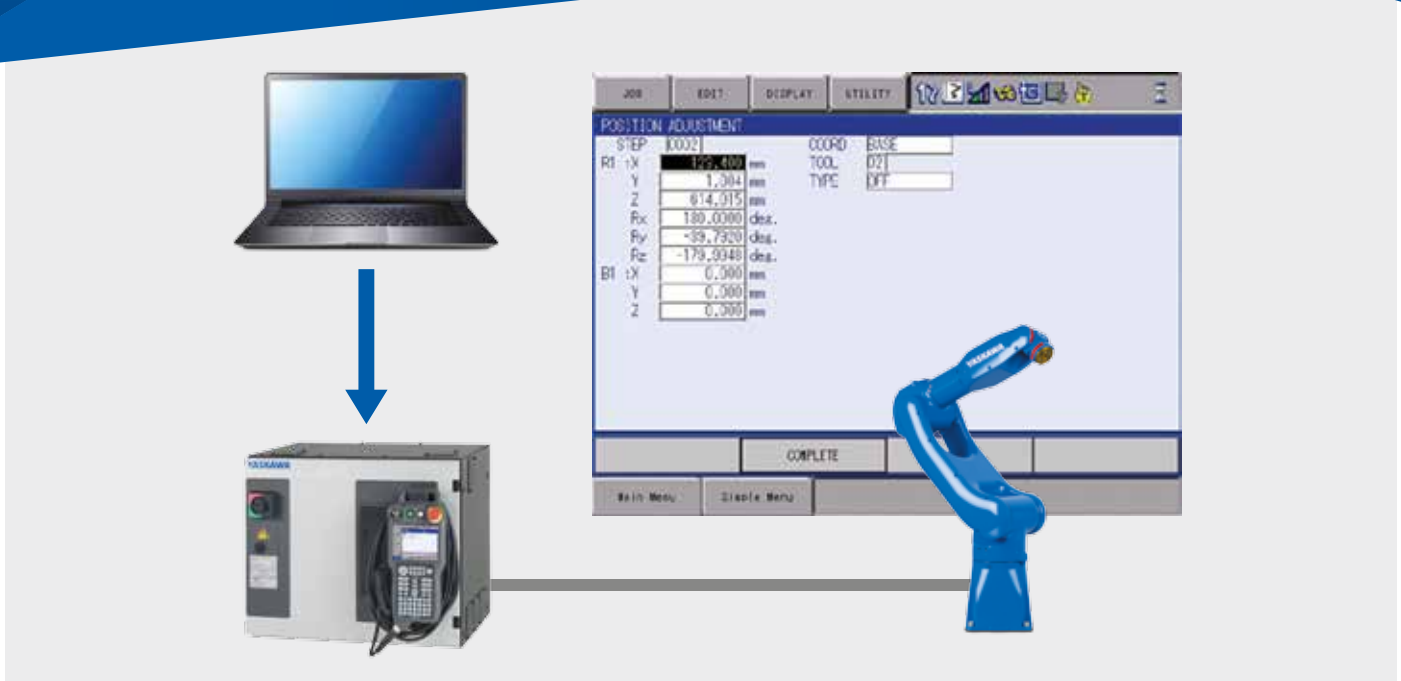
VORTEILE IM ÜBERBLICK

- Die Roboter und die externe Achse können ohne Stromanschluss bewegt werden

YRC1000
micro

Teaching Point Adjustment

(195334)



Die Funktion „Teaching Point Adjustmen“ ändert die ge-teachten Funktionsdaten durch Eingabe der numerischen Zahl auf dem Programmierhandgerät ohne Betrieb des Roboters.

Diese Funktion ermöglicht vereinfachtes Offline-Teaching mit CAD-Daten oder anderen, und ermöglicht eine Fein-einstellung von Positionsdaten in jedem beliebigen Koordinatensystem ohne Betrieb des Roboters.

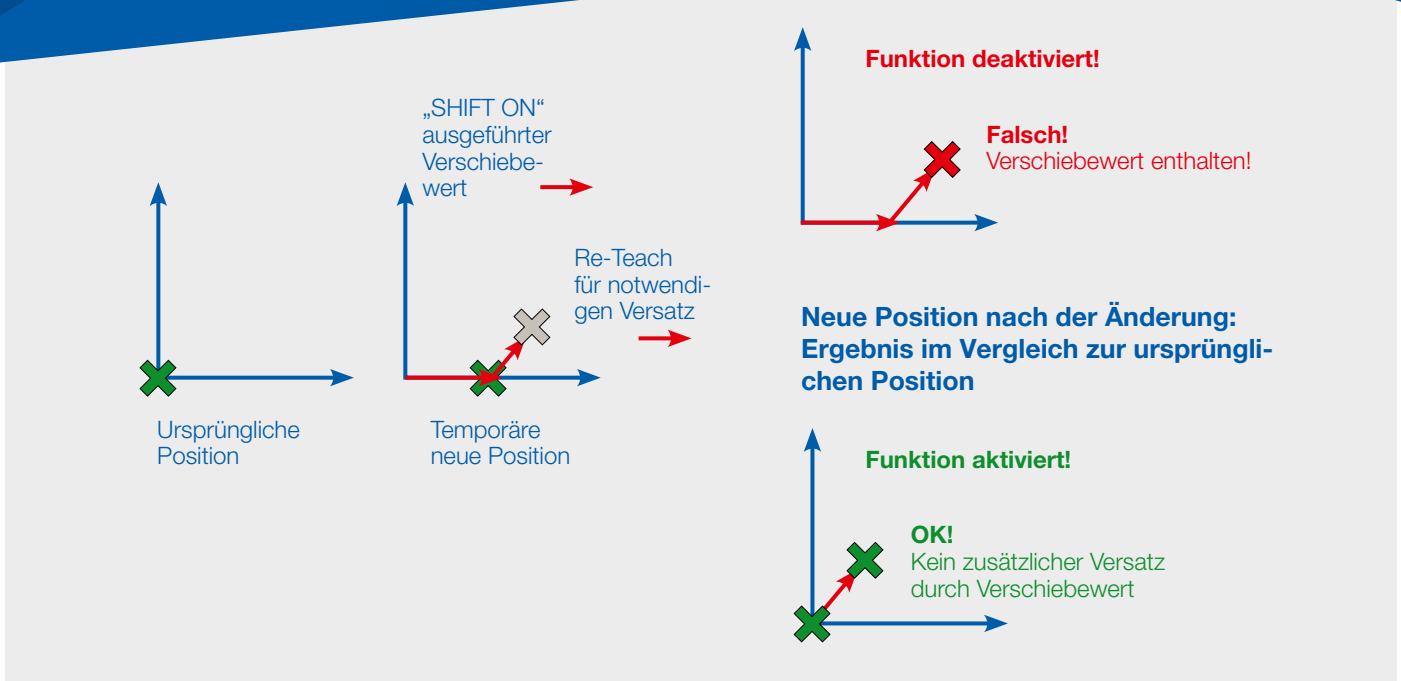
VORTEILE IM ÜBERBLICK

- Die Teaching-Modifikationszeit wird bedeutend verringert
- Die Position kann in dem Fenster auf dem Programmierhandgerät einfach korrigiert werden

YRC1000
micro

Shift Condition Cancel

(195338)



Diese Funktion sollte immer aktiviert sein, wenn die „Shift On“-Anweisung zur Programmierung verwendet wird!

Sie lässt das Überschreiben (Re-teach) von Roboterpositionen unter Berücksichtigung aktueller Versatzwerte zu.

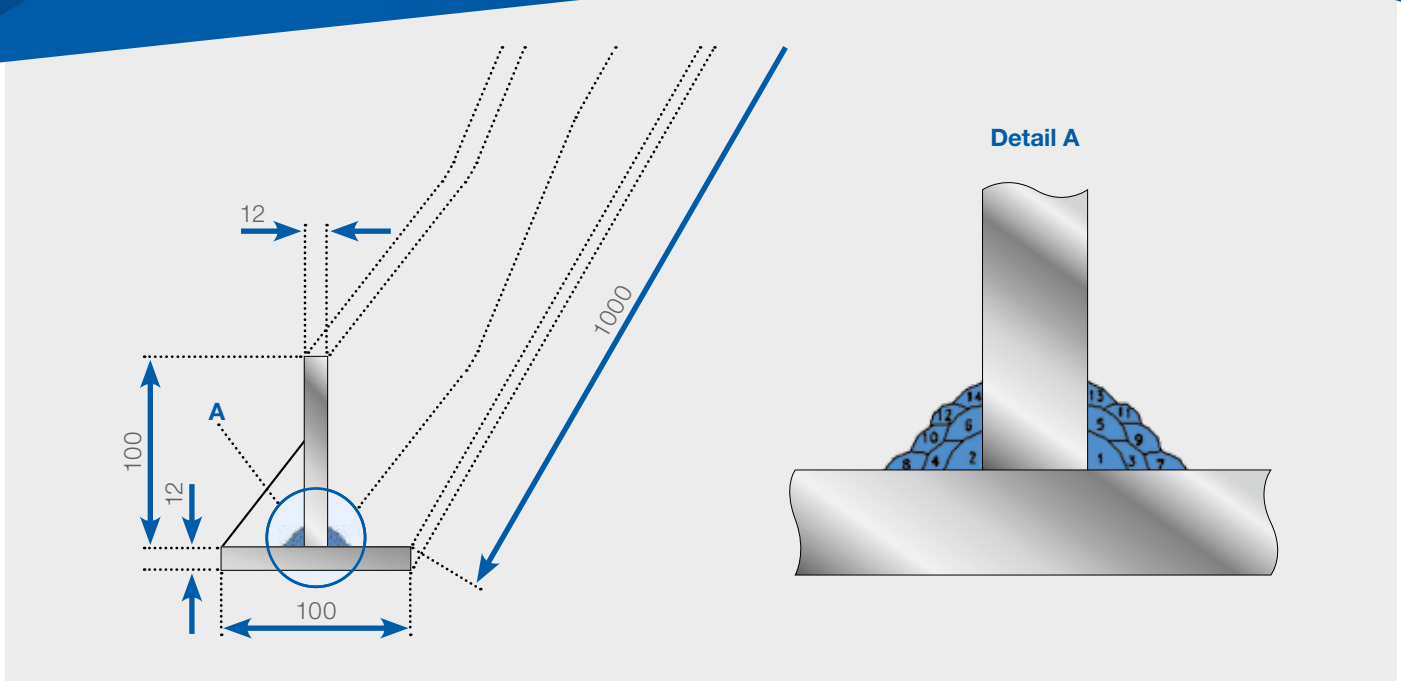
VORTEILE IM ÜBERBLICK

- Kein „Master-Werkstück“ mit Originalprogrammierung notwendig
- Nachbesserungen an vorhandenen „SHIFT Programmen“ sind einfach und komfortabel
- Vorübergehend verschobene Punkte können unabhängig von dem aktuellen Verschiebewert überschrieben (modifiziert) werden

YRC1000
micro

Point Variable (T-Variable)

(195366)



Diese Funktion ist fester Teil eines Mehrlagen-Funktionspakets. Diese spezielle Art von Positionsvariablen wurde geschaffen, um die Programmierung von mehrlagigen Schweißaufgaben (Figuren oben) zu vereinfachen. Alle Schweißvorgänge haben im Grunde dieselbe Roboterposition. Der einzige Unterschied besteht in dem (variablen) Verschiebewert von der Wurzel zur aktuellen Lage. Nur die erste Lage muss durch Bewegen des Roboters geteacht werden. Diese Positionen werden als T-Variable registriert, und in Ihrem JOB können Sie dieselben Positionen so oft wie notwendig (Lagenanzahl) verwenden, ohne den Roboter erneut zu bewegen.

Der Hauptunterschied zur Standardpositionsvariablen (P-Variable) ist der „lokale“ Charakter: Das heißt, dass alle Informationen im aktuellen JOB gespeichert sind, und jede definierte T-Variable nur zu dem JOB gehört, in dem sie ge-

setzt worden ist. Die gleiche T-Variable (Zahl) kann in unterschiedlichen JOBS verwendet werden, und kann unterschiedliche Positionsinformationen beinhalten! Die P-Variable andererseits hat „globalen“ Charakter, und wenn sie einmal gesetzt ist, dann ist sie allgemein in allen Steuerungsbereichen gültig. Die Verwendung der T-Variablen ist nicht obligatorisch, doch je mehr Schichten Sie teachen müssen, umso nützlicher werden diese T-Variablen.

VORTEILE IM ÜBERBLICK

- Vereinfacht die Programmierung von Mehrlagen-Schweißungen
- Hilft bei der Einsparung von Speicherplatz
- Ersetzt begrenzte P-Variablen
- Bis zu 9999 T-Variablen für jeden JOB
- Offline-Editieren möglich

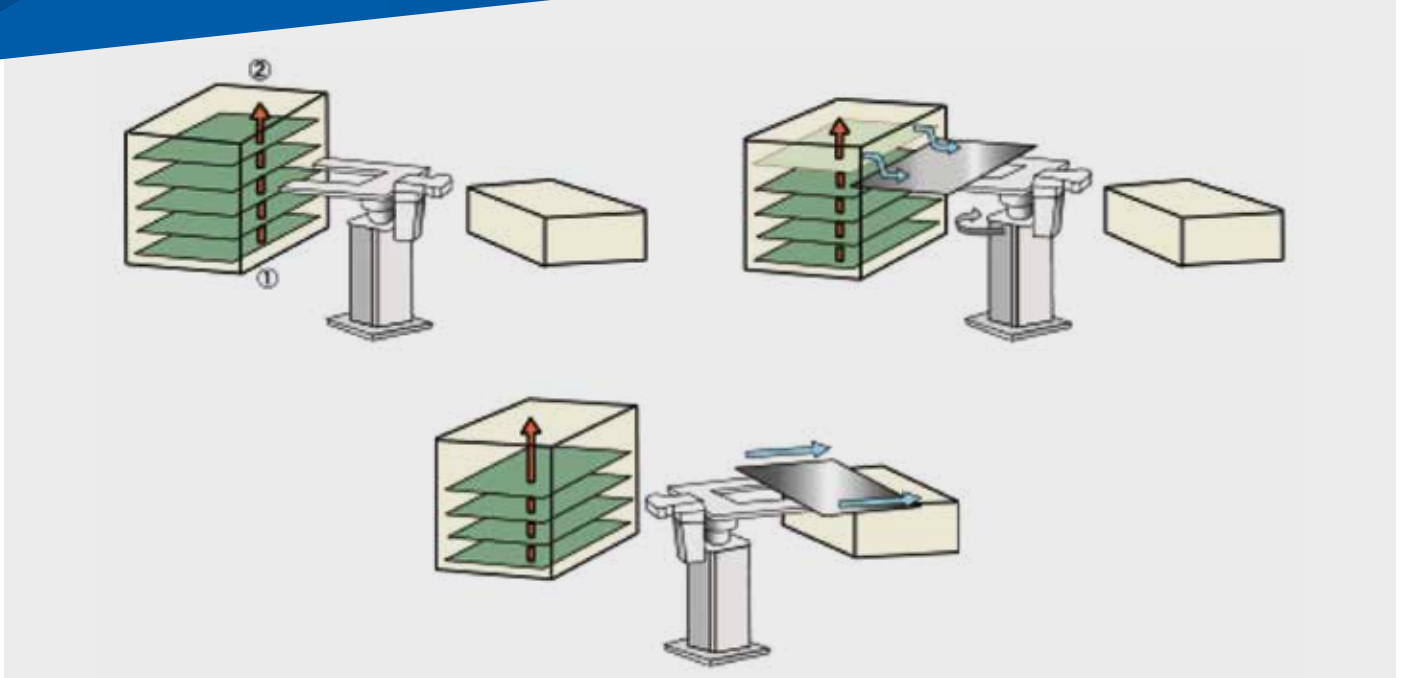
POSITION ADJUSTMENT			
STEP	0006	T0004	
R1 :X	820.000	mm	COORD
Y	0.000	mm	TOOL
Z	614.000	mm	TYPE
A	0.0000	deg.	
B	0.0000	deg.	
C	180.0000	deg.	
Re	0.0000	deg.	
B1 :X	0.000	mm	
Y	0.000	mm	
Z	0.000	mm	

POINT VARIABLE				
●: UNUSED ○: RESERVED				
0000	0001	0002	0003	0004
0005	●0006	0020	0030	0040
0050	0060	●0100	○0110	

YRC1000
micro

Search Continuous Motion

(195340)



Die Funktion „Search Continuous Motion“ enthält die Daten von bis zu 50 Positionen, in denen der YRC1000-Digital-signale von allgemeinen Sensoren während der Ausführung eines Jobs erkennt, während er sich ohne Anhalten der Bewegung über einige Teile in einer Linie bewegt.

Erkennt außerdem bis zu 50 Positionen von kreisförmig positionierten Teilen mit einer kreisförmigen Bewegung.

Programme können vereinfacht werden. Da normale Programme ein Suchverfahren für jedes Werkstück erfordern, ermöglicht diese Funktion dem Roboter, alle Werkstücke in einer Linie mit nur einer Bewegung und einem NSRCHON-Befehl zu suchen, was das Programm vereinfacht.

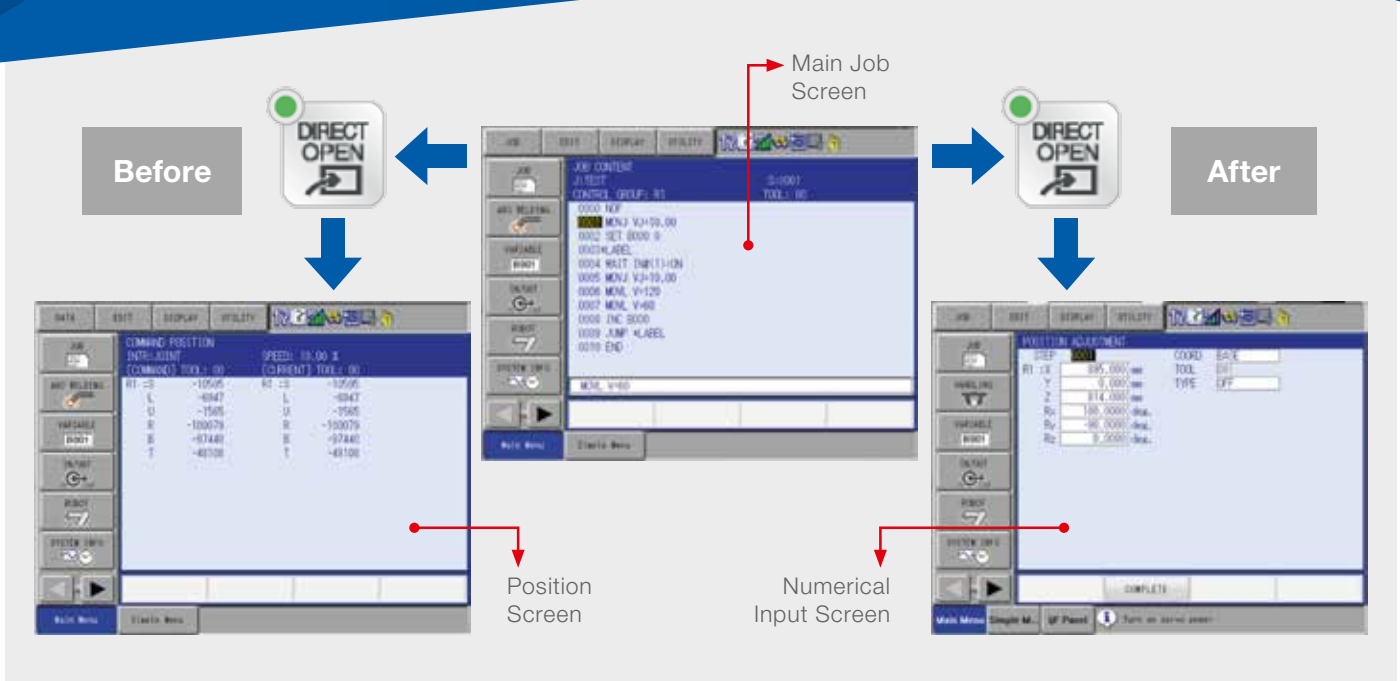
VORTEILE IM ÜBERBLICK

- Schnelle Möglichkeit zum Auffinden jeder Position von mehr Teilen in einer Linie mit nur einer Bewegung
- Einsparung von Zykluszeit
- Vereinfachte Roboterjobs, Einsparung von Programmierungszeit

YRC1000
micro

Numerical Input Screen Display with Direct Open

(195335)



Diese Funktion zeigt umgehend die registrierten Positionsdaten des Korrektur-Displays an. (Hinweis: Funktioniert nur, wenn die Funktion „Teaching Point Adjustment“ aktiviert ist).

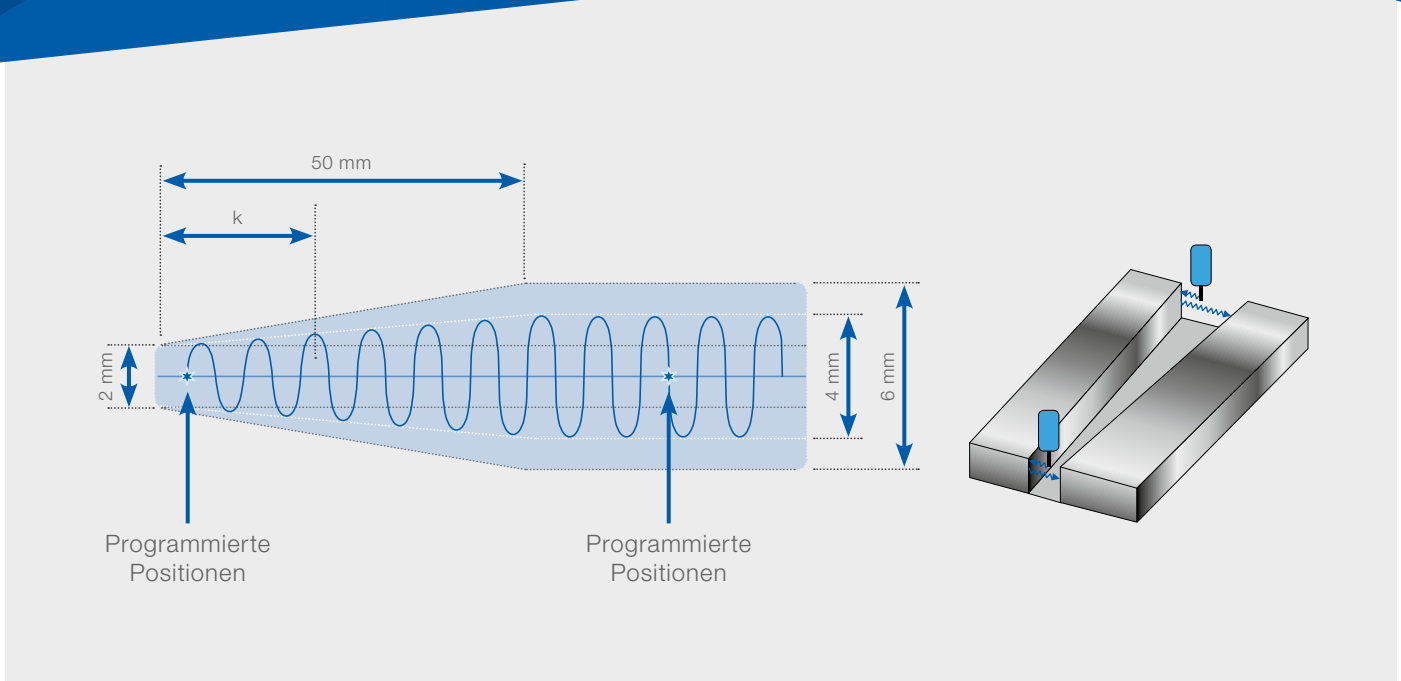
VORTEILE IM ÜBERBLICK

- Einfacheres und schnelleres Online-Teaching

YRC1000
micro

Weaving Adjustment

(195367)



Diese Funktion kann verwendet werden, um variable Bedingungen in einer Schweißnaht durch kontinuierliche Anpassung der Pendelparameter und Geschwindigkeit auszugleichen.

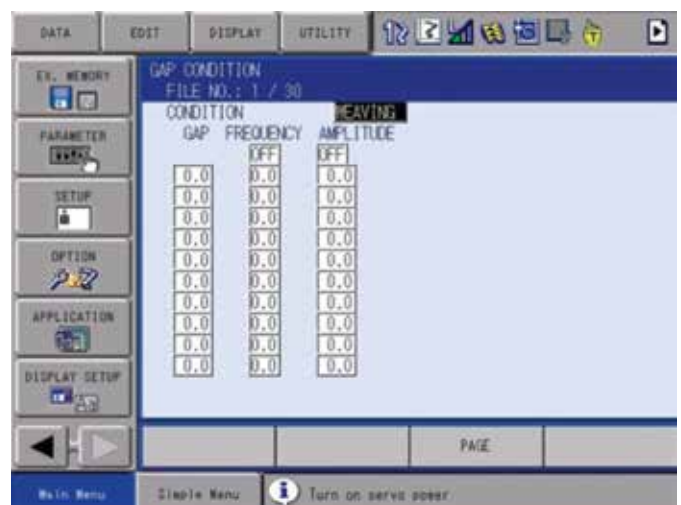
Die Informationen über Fugenbedingungen können manuell als konstante Werte gesetzt werden oder können automatisch durch Messgeräte wie Touch-Sensoren oder eine andere Startpunkterkennungseinheit ermittelt werden: Bitte klären Sie die Machbarkeit im Detail mit der Technikabteilung.

FUNKTIONSÜBERSICHT

- Pendelamplitude wird kontinuierlich erhöht/erniedrigt
- Robotergeschwindigkeit und Pendelfrequenz können bei Bedarf ebenfalls angepasst werden

VORTEILE IM ÜBERBLICK

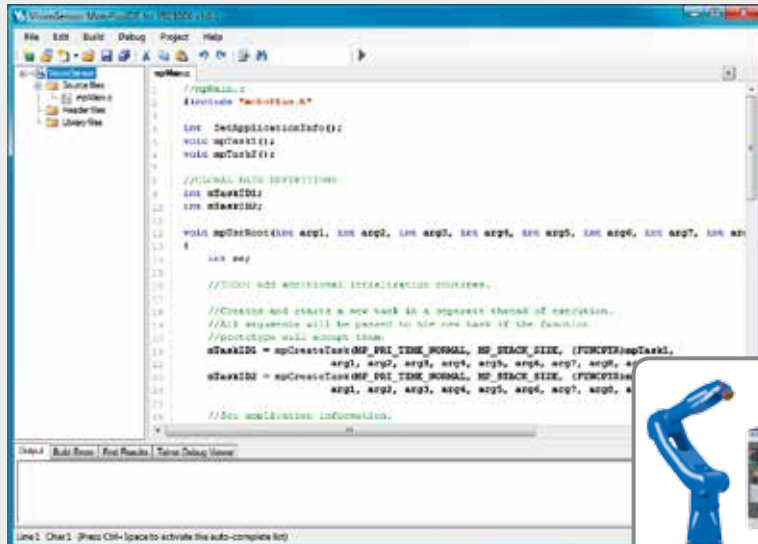
- Erweiterung der Standard-Pendelfunktion
- Verfügbar für die Anwendung „General“ und „Arc Welding“
- Die Funktion ist optional auch Teil der adaptiven Schweißroutinen in der MotoEyeLT-Software (siehe Bild oben: Spaltbedingungsdatei), wenn Laser-Schweißnahtverfolgung verwendet wird



YRC1000
micro

MotoPlus Runtime

(195358)



```
//apMain.c
#include "MotoPlus.h"

int SetApplicationInfo()
void apTaskInit()
void apTaskStart()

//FUNCTIONS DEFINITIONS
int mTaskID1;
int mTaskID2;

void apTaskStart(int arg1, int arg2, int arg3, int arg4, int arg5, int arg6, int arg7, int arg8)
{
    int arg;

    //TODO: Add additional initialization routines.

    //Create and create a new task in a separate thread of execution.
    //All arguments will be passed to the new task if the function.
    //prototype will accept them:
    mTaskID1 = apCreateTask(MP_PRI_TIME_NORMAL, MP_STACK_SIZE, (FUNC_PTR)apTask,
        arg1, arg2, arg3, arg4, arg5, arg6, arg7, arg8, 0);
    mTaskID2 = apCreateTask(MP_PRI_TIME_NORMAL, MP_STACK_SIZE, (FUNC_PTR)apTask,
        arg1, arg2, arg3, arg4, arg5, arg6, arg7, arg8, 0);
}

//See application information.
```



Die „MotoPlus Runtime“ ist für die Ausführung von Anwendungen erforderlich, die mit dem MotoPlus SDK erstellt sind. Das SDK bietet eine vollständige Programmierungsumgebung zur Erzeugung von Anwendungen auf der Basis der Programmiersprache C, die in der Steuerung ausgeführt wird. Das SDK bietet Zugang zu den folgenden APIs:

- Aufgabensteuerungs-API
- Netzwerk-API
- Serielles Kommunikations-API
- Systemüberwachungs-/Steuerungs-API
- Bewegungsüberwachungs-/Steuerungs-API
- Sensor-API
- Datei-API

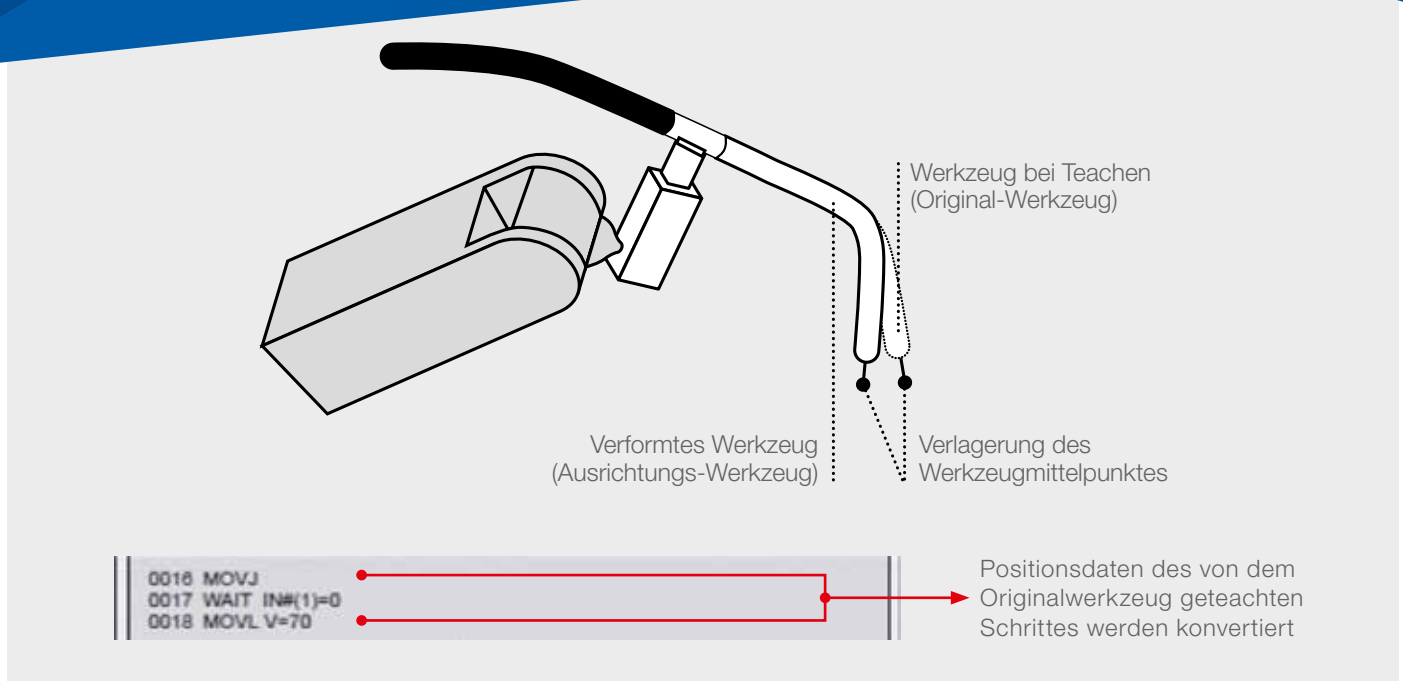
VORTEILE IM ÜBERBLICK

- Ist für die Ausführung von Anwendungen erforderlich, die mit dem MotoPlus SDK erstellt sind
- Ausführung mehrerer MotoPlus-Anwendungen pro Steuerung
- Laufzeitlizenz für jede Steuerung erforderlich
- MotoPlus SDK zur Entwicklung von Anwendungen

OPTIONALE SOFTWARE

- MotoPlus SDK – Entwicklungsumgebung zur Erzeugung von MotoPlus-Anwendungen
- 181529_F-Pkt YRC1000 MotoGSI –MotoGSI ist ein Entwicklungspaket zur Erzeugung von Kommunikationsschnittstellen auf Jobbasis

YRC1000
micro



Die Funktion „PMT“ ändert einfach und genau Positionsdaten, wenn das Werkzeug verformt ist. (PMT: Positionsänderung für Werkzeugverformung: Positionsdaten-Änderungsfunktion für Werkzeugverformung).

Wenn das Werkzeug zufällig mit einem Jig oder einer Wand in der Umgebung kollidiert und verformt wird, werden die Kontrollpunkte verschoben. Folglich werden im Job geteachte Teaching-Positionen ebenfalls verschoben.

Die Funktion „PMT“ ermöglicht eine Verringerung von Zeit und Arbeit zur Änderung der Verschiebung. Wenn ein Job spezifiziert wird, dann werden die Positionsdaten der Werkzeugabmessungen vor der Verformung (während dem Teachen) automatisch in die Positionsdaten der Werkzeugabmessungen nach der Verformung konvertiert.

Das Werkzeug vor der Verformung wird das alte Werkzeug genannt, und das Werkzeug nach der Verformung wird das neue Werkzeug in der PMT-Funktion genannt.

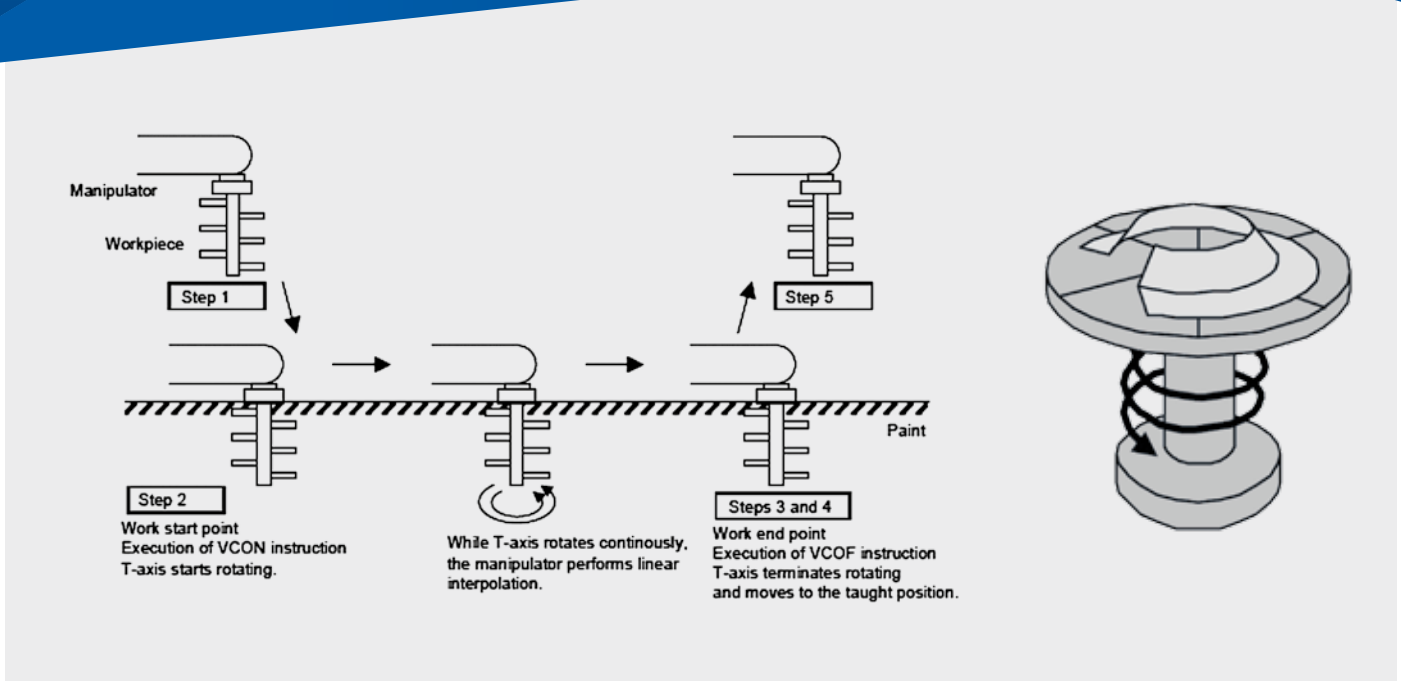
VORTEILE IM ÜBERBLICK

- Zeit und Arbeit für die Teaching-Veränderung können verringert werden. Die Teaching-Änderungsarbeit kann effizient ausgeführt werden, da die Positionsdaten automatisch von dem alten Werkzeug zum neuen konvertiert werden können.
- Durch die Speicherung von Werkzeugdatenaufzeichnungen können die ursprünglichen Werkzeugdaten einfach wieder hergestellt werden. Die ursprünglichen Werkzeugdaten können einfach wieder hergestellt werden, da die Änderungsaufzeichnung der Werkzeugkonstanten im Fenster „Tool Backup“ geprüft werden können.

YRC1000
micro

Speed Control

(195361)



Die Funktion „Speed Control“ ermöglicht, dass die T-Achse, die Spitzenachse des Manipulators und die externe Achse (nachfolgend bezeichnet als Drehzahlregelachse) kontinuierlich mit konstanter Drehzahl rotieren können.

Während der kontinuierlichen Rotation wird die Drehzahl der Drehzahlregelachse unabhängig geregelt. Während des Betriebs des Manipulators mit Drehzahlregelfunktion rotiert die Drehzahlregelachse unabhängig vom Teaching mit der festgelegten Drehzahl, während andere Achsen entsprechend den Teaching- Vorgaben rotieren.

Line	Step	INFORM Instruction	Explanation
0000		NOP	
0001	0001	MOVJ VJ=12.50	Moves to the waiting point.
0002	0002	MOVJ VJ=12.50	Moves to the work start point.
0003		VCON ROBOT=1 JOINT=6 RPM=1000	Starts rotation of the T-axis of the manipulator 1. Rotation speed: 10.00 [rotation/min]
0004		TIMER T=0.50	Waits for rotation to start.
0005	0003	MOVL V=100	Moves to work end point by linear interpolation at 100.0 [mm/sec].
0006		VCOF ROBOT=1 JOINT=6	Terminates rotation.
0007	0004	MOVJ VJ=12.50	Moves T-axis to the taught position.
0008	0005	MOVJ VJ=12.50	Moves to the waiting point.
0009		END	

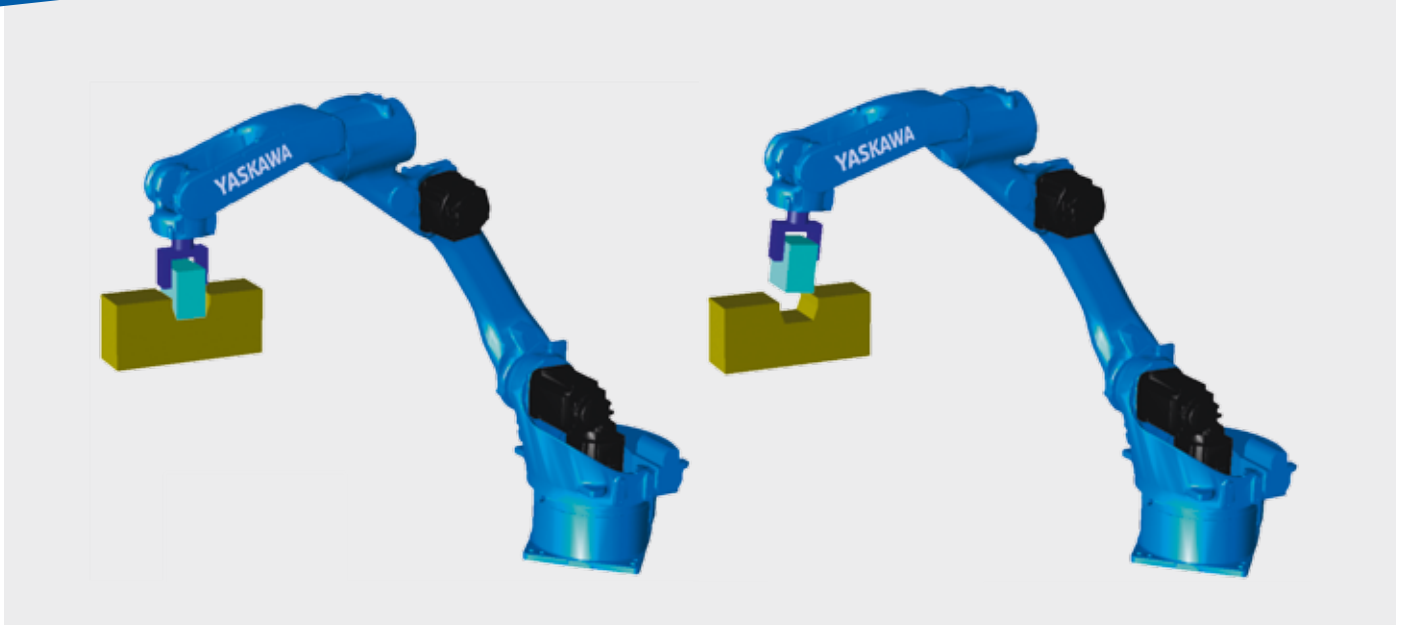
VORTEILE IM ÜBERBLICK

- Die Teaching-Zeit kann verringert werden
- Die Arbeitseffizienz kann verbessert werden
- Die Drehzahlregelachse rotiert unabhängig vom Teaching mit der festgelegten Drehzahl

YRC1000
micro

Link Servo Float

(195332)



Die Funktion „Servo Float“ steuert nicht nur die Position des Roboters, sondern die Position und die Kraft des Roboters. Normalerweise, selbst wenn eine Kraft von außen auf den Roboter aufgebracht wird, versucht der Roboter, die aktuelle Position beizubehalten und bewegt sich nicht, da nur die Roboterposition gesteuert wird. In diesem Fall bietet die Funktion „Servo Float“ eine flexible Steuerung der Position und Haltung des Roboters als Reaktion auf die von außen aufgebrachte Kraft. Sie lässt das Überschreiben (Reteach) von Roboterpositionen unter Berücksichtigung aktueller Versatzwerte zu.

Link Servo Float-Funktion

Diese Funktion führt das Servo-Gleiten für jede Achse des Roboters aus. Diese Funktion wird verwendet, wenn eine Kraft nur auf eine spezifische Achse aufgebracht wird, oder wenn Servo Float auf alle Achsen des Roboters aufgebracht wird, da die Richtung, in der die Kraft aufgebracht wird, nicht identifizierbar ist.

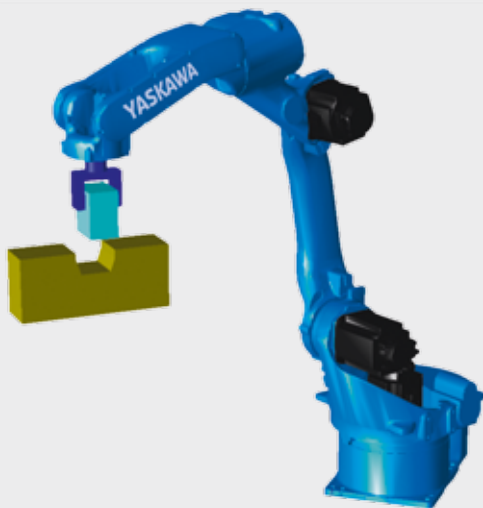
VORTEILE IM ÜBERBLICK

- Keine zusätzliche Hardware erforderlich
- **Link Servo Float-Funktion**
Kann von dem Programmierer für jede Achse oder insgesamt aktiviert werden

YRC1000
micro

Link and Linear Servo Float

(195333)



Die Funktion „Servo Float“ steuert nicht nur die Position des Roboters, sondern die Position und die Kraft des Roboters. Normalerweise, selbst wenn eine Kraft von außen auf den Roboter aufgebracht wird, versucht der Roboter, die aktuelle Position beizubehalten und bewegt sich nicht, da nur die Roboterposition gesteuert wird. In diesem Fall bietet die Funktion „Servo Float“ eine flexible Steuerung der Position und Haltung des Roboters als Reaktion auf die von außen aufgebrachte Kraft. Sie lässt das Überschreiben (Reteach) von Roboterpositionen unter Berücksichtigung aktueller Versatzwerte zu.

Link Servo Float-Funktion

Diese Funktion führt das Servo-Gleiten für jede Achse des Roboters aus. Diese Funktion wird verwendet, wenn eine Kraft nur auf eine spezifische Achse aufgebracht wird, oder wenn Servo Float auf alle Achsen des Roboters aufgebracht wird, da die Richtung, in der die Kraft aufgebracht wird, nicht identifizierbar ist.

Linear Servo Float-Funktion

Diese Funktion führt das Servo-Gleiten für jede Koordinatenachse des Koordinatensystems wie die Roboterkoordinate, Basiskoordinate, Benutzerkoordinate und Werkzeugkoordinate aus. Diese Funktion wird verwendet, wenn die Kraft nur auf die spezifische Richtung von jedem Koordinatensystem aufgebracht wird.

VORTEILE IM ÜBERBLICK

- Keine zusätzliche Hardware erforderlich
- **Link Servo Float-Funktion**
Kann von dem Programmierer für jede Achse oder insgesamt aktiviert werden
- **Linear Servo Float-Funktion**
Kann von dem Programmierer für jedes Koordinatensystem in jeder Richtung aktiviert werden

YRC1000
micro

T-axis Endless

(195325)



Diese Funktion dreht ununterbrochen die T-Achse für eine Mehrzahl von Umdrehungen. Obwohl der Drehwinkel der T-Achse im Allgemeinen auf innerhalb von ± 360 Grad begrenzt ist, ermöglicht die Funktion deren unendliche Drehung. Beim Ausführen der Bewegungsanweisung MOVJ mit spezifizierter „Anzahl von Umdrehungen der externen Achse“, dreht sich die Drehachse um den „spezifizierten Drehbetrag + Teaching-Positionsimpulse“ während der Bewegung in die Zielposition. Ein Bewegungsbefehl kann bis zu ± 100 Umdrehungen spezifizieren.

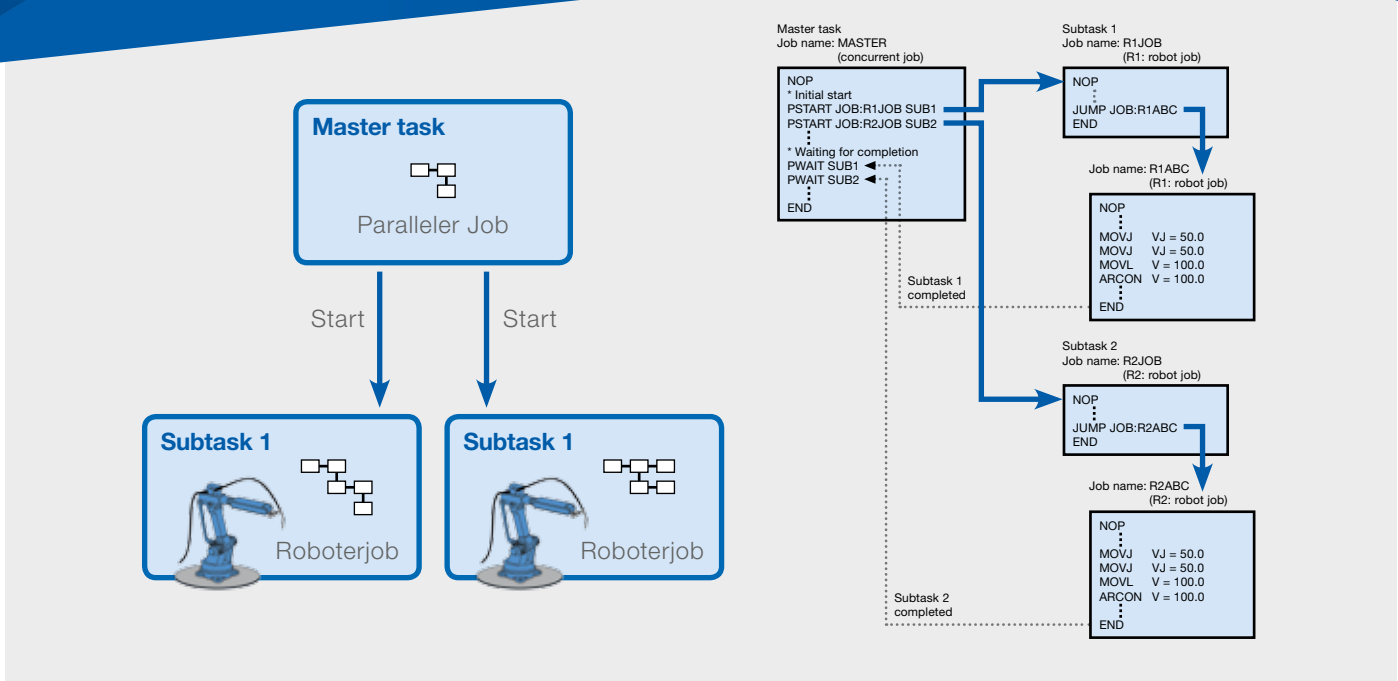
VORTEILE IM ÜBERBLICK

- Kein Zurückdrehen der T-Achse notwendig
- Teaching-Zeit kann verringert werden
- Taktzeit kann verringert werden

YRC1000
micro

Independent Control

(195326)



Die Steuerung kann maximal 8 Roboter und mehrere Stationen (Jig) mit einer Steuerung steuern. Die Funktion „Independent Control“ ermöglicht den unabhängigen Betrieb von zwei Robotern oder des Roboters und des Jigs unabhängig voneinander mit unterschiedlichen Programmen.

Die Steuerung verfügt über eine Funktion zum Decodieren und Ausführen von maximal 16 Jobs einzeln. Die in dieser Funktion ausgeführte Multitask-Steuerung wird Independent Control genannt.

So kann z.B. ein Zwei-Roboter-System gebaut werden, das aus einem Schweißroboter besteht, der das Werkstück schweißt und aus dem anderen Roboter, der ungeschweißte Werkstücke sowie geschweißte Werkstücke überträgt.

Das System ist frei mit Anwendungen nicht nur zum Schweißen + Handhaben, sondern auch Punktschweißen + Handhaben, Handhaben + Handhaben und anderen Dingen kombinierbar. (Die Tastenfolien sind auf der Basis der Beststellungsproduktion verfügbar.)

Außerdem ist ein unabhängiger Betrieb von Roboter und Station (externes Jig) möglich. Als externer Achsenmotor muss jedoch der Motor für den YASKAWA-Roboter verwendet werden.

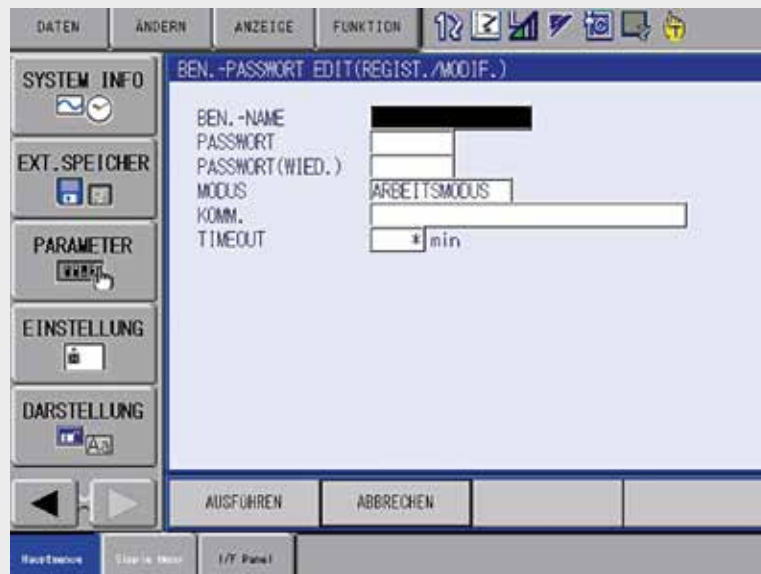
VORTEILE IM ÜBERBLICK

- Multitasking von 16 Jobs einzeln
- Maximal 8 Roboter und mehrere Stationen mit einer Steuerung
- Frei kombiniert mit Anwendungen

YRC1000
micro

Password Protection

(195347)



Die Funktion „Password Protection“ hilft, die Systemsicherheit zu garantieren, indem sie von jedem Benutzer verlangt, dass er über eine persönliche registrierte Berechtigung verfügt, um auf die Steuerung zuzugreifen um zu steuern, welche Vorgänge von den Benutzern ausgeführt werden können.

Der Systemadministrator autorisiert jeden Benutzer durch Zuordnung eines Login-Namens und eines Passworts, einer Sicherheitsniveau- und einer Timeout-Einstellung, wodurch ein spezifisches Niveau des Zugriffs auf die Steuerung ermöglicht wird.

Der Administrator kann bis zu 100 Benutzerkonten registrieren. Die Benutzerkonteninformationen können in einer Datei (USRINFO.DAT) gespeichert werden.

VORTEILE IM ÜBERBLICK

- 100 Benutzerkonten
- Alarme mit Benutzernamen
- Logbuchfunktion mit Benutzernamen

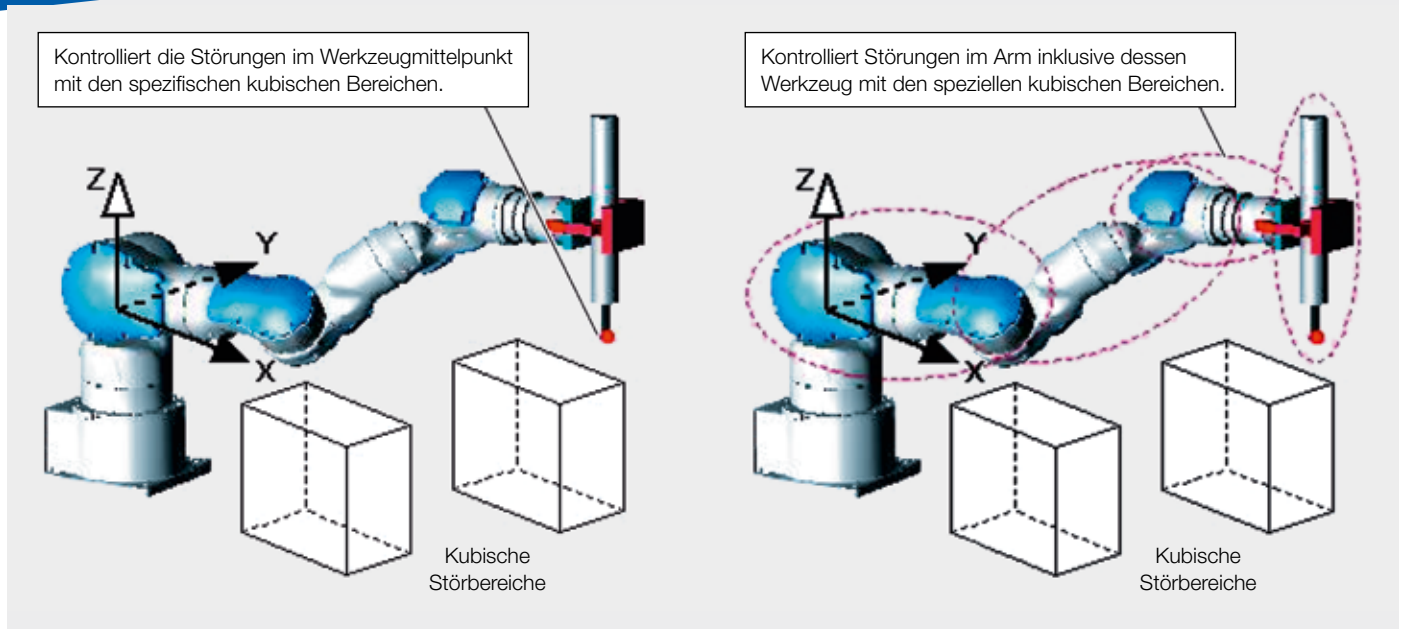
Nur der Systemadministrator kann registrierte Benutzerkonteninformationen ändern.

Die Funktion „Password Protection“ ermöglicht auch durch Nachverfolgung der Alarmhistorie herauszufinden, welcher Benutzer in der Zeit eines bestimmten Alarms eingeloggt war.

YRC1000
micro

Arm Interference with Specified Cubic Area Check

(195350)



Durch diese Funktion werden die entsprechenden Systemausgangssignale „Signal KUBISCHER STÖRBEREICH“ aktiviert, wenn der Werkzeugmittelpunkt mit spezifizierten kubischen Bereichen kollidierte.

Andererseits werden durch diese Funktion die entsprechenden Systemausgangssignale „Signal KUBISCHER STÖRBEREICH“ aktiviert, wenn der Arm des Roboters mit seinem Werkzeugmittelpunkt mit spezifizierten kubischen Bereichen kollidierte.

Es können bis zu 8 Überschneidungsbereiche für den Arm als Störbereiche registriert werden.

Wenn der Arm des Roboters mit seinem Werkzeugmittelpunkt mit bereits spezifizierten kubischen Bereichen kollidiert, tritt ein Alarm ein und der Roboter hält sofort an.

Die Form des Werkzeugteils des Roboters muss vom Kunden registriert werden, da die Werkzeugform von der Arbeit abhängig ist, die der Roboter ausführt. Die Form kann in der Datei TOOL INTERFERE registriert werden. Die Funktion „Arm interference check“ muss aktiviert sein.

VORTEILE IM ÜBERBLICK

- Durch diese Funktion können Kollisionen zwischen Robotern/Werkzeugen und kubischen Elementen vermieden werden

YRC1000
micro

Pendant Oscilloscope

(195351)



„Pendant Oscilloscope“ ist eine Überwachungsfunktion zur Visualisierung der Geschwindigkeit und des Drehmoments von jeder Roboterachse, und des Status von gleichzeitigen E/ASignalen auf dem Programmierhandgerät. Das Handgeräteoszilloskop wird über eine Benutzeroberfläche konfiguriert, für die keine zusätzlichen Hardwarekomponenten erforderlich sind. Die mit einem integralen Anzeigebildschirm mit Wellenformanzeigefenster und der Bedingungeinstellungskonsole ausgerüstete Programmierhandgeräte-Oszilloskop-Anwendung ermöglicht die Ausführung mehrerer Prozesse von Bedingungeinstellung bis zu gleichzeitigen Messvorgängen.

DIE FOLGENDEN MERKMALE SIND VORHANDEN

- Selektive Kanalauslösung und Auslöseniveauanpassung
- Cursor-Messfunktion zum Messen bestimmter Abstände in Wellenformen oder zwischen Spitzen nach der Datenerfassung
- Zoomfunktion zum Anzeigen der Daten in vergrößerter Ansicht
- Manueller und automatischer Speichermodus zum Speichern von Daten

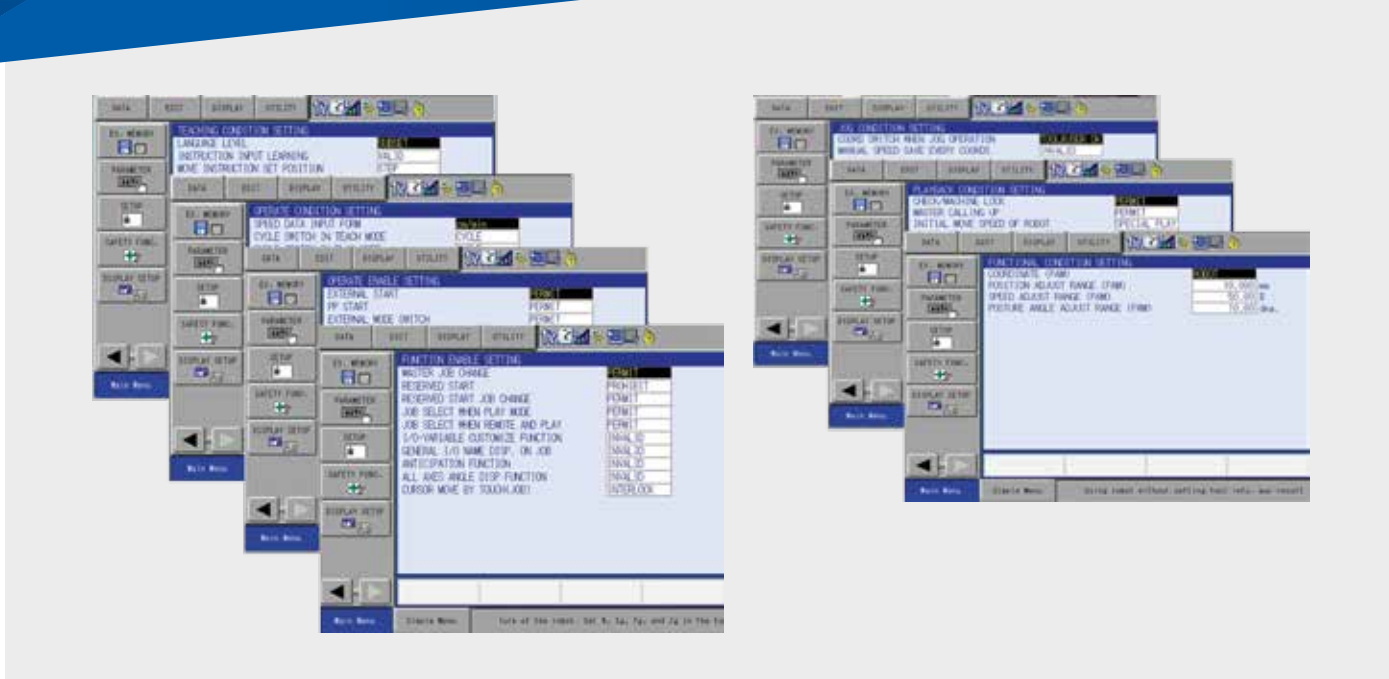
VORTEILE IM ÜBERBLICK

- Kein externes Oszilloskop notwendig
- Die Servobedingungen können leicht überwacht und aufgezeichnet werden
- Einfache Konfiguration aller relevanten Messparameter
- Daten können zwecks einfacher Analyse auf einem PC in einer CSV-Datei protokolliert werden

YRC1000
micro

Configuration Parameter

(195353)



- Empfohlene Grundeinstellungen
- YASKAWA-Standardsystemkonfiguration
z.B. das Verhalten des Programmierhandgerätes,
Setup-Menüereinstellungen, Komforteinstellungen usw.
- Diese Einstellungen können vom Kunden in der
Steuerung selbst eingestellt /verändert werden

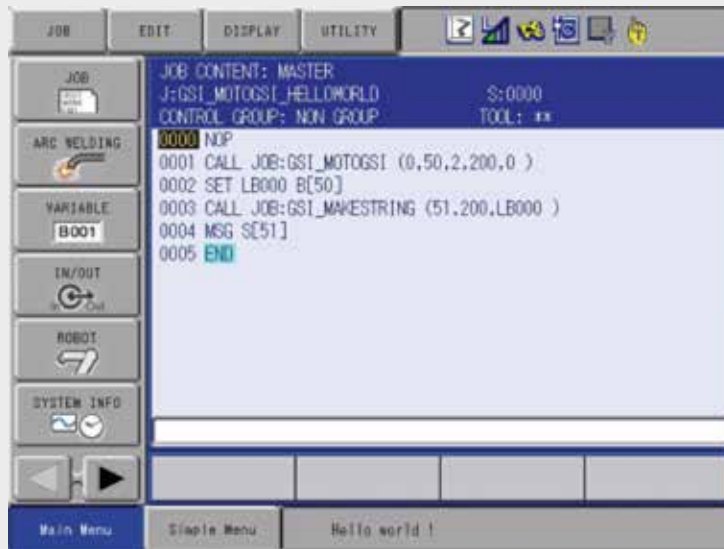
VORTEILE IM ÜBERBLICK

- Nutzung empfohlener Grundeinstellungen
- Individuelle Konfiguration

YRC1000
micro

Ethernet Standard GSI

(195379)



MotoGSI erweitert die Kommunikationsmöglichkeit der YRC1000-Steuerung, indem die allgemeine Stecker-Kommunikationsanweisung zur INFORM-Sprache hinzugefügt wird. Dadurch kann ein üblicher Roboterprogrammierer ein breites Spektrum von Kommunikationsaufgaben lösen.

HAUPTFUNKTIONEN

- Erweiterung der INFORM Sprache durch Socket-kommunikationsfunktionen basierend auf Ethernet/TCP, Ethernet/UDP oder RS232
- Keine zusätzliche Hardware benötigt. Kommunikation via Standard Ethernet oder RS232 Anschluss an der Steuerung
- Kommunikation mit fast jeder externen Schnittstelle die Ethernet/TCP, Ethernet/UDP oder RS232 unterstützt
- Zusätzliche PHG App für Fehlerbehebung von GSI Jobs

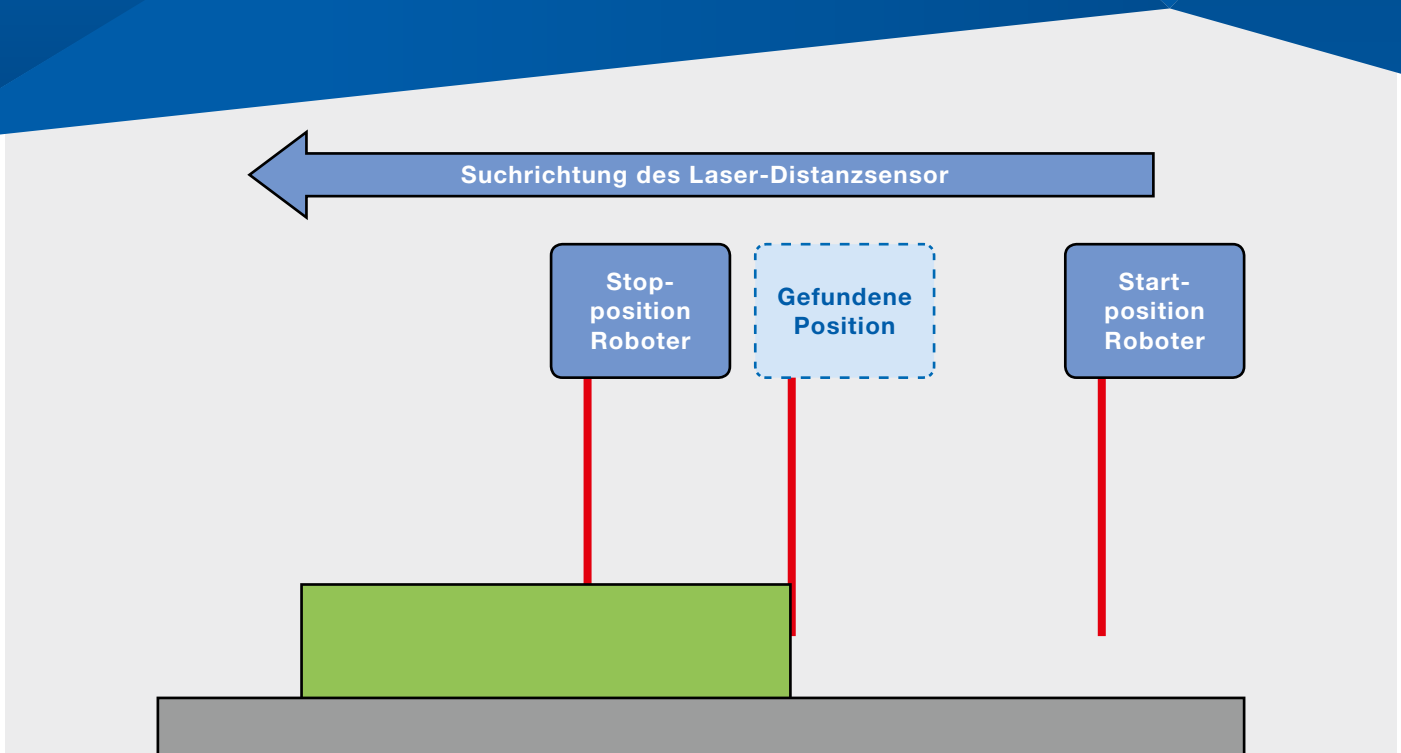
VORTEILE IM ÜBERBLICK

- Erweiterung der INFORM Sprache
- Implementierung von speziellen Protokollen basierend auf Ethernet/TCP, Ethernet/UDP oder RS232
- Einstellung von Hilfe-Funktionen um Datenströme zu verarbeiten oder zu konvertieren
- Auf Pattern basierende Suche im Eingangs-Datenstrom
- Einfache Integration in eigene INFORM Jobs
- Basierend auf MotoPlus

YRC1000
micro

Search Function – Soft Stop

(195328)



Die „Search Function – Soft Stop“ nutzt diverse allgemeine Erkennungssensoren um den Roboter über das Erkennungssignal des Sensors zu stoppen. Der Roboter kann mit dem nächsten Arbeitsschritt fortfahren, da diese Funktion das zu bearbeitende Werkstück sucht.

Wenn der Roboter ein Sensor-Signal entdeckt, speichert die Steuerung die gefundene Position und der Roboter stoppt die Bewegung mit Erklärung (Soft Stop).

Mit berührungslosen Sensoren ist eine extrem schnelle Suche möglich.

VORTEILE IM ÜBERBLICK

- Roboter stoppt Bewegung mit Erklärung
- Hochgeschwindigkeits-Suche
- Höhere Laufzeit des Untersetzungsgetriebes
- Anwendbar um die Stapelhöhe zu ermitteln

YRC1000
micro

I/O JOG Operation in Play-mode

(195337)



Die Funktion „I/O JOG Operation in Play-Mode“ führt die Achsbewegungen für Roboter und externe Achsen aus und verwendet dabei universelle Eingabesignale anstelle des Programmiergeräts.

Die Achsbedienung (JOG Betrieb) eines Roboters oder einer Stationssteuerungsgruppe, die nicht in dem Job im Play-back-Betrieb registriert ist, kann mithilfe der zugewiesenen universellen Eingangssignale ausgeführt werden.

Universelle Eingänge werden auch für die Auswahl der Bewegungsgeschwindigkeit verwendet. 5 Stufen oder individuelle Geschwindigkeiten sind möglich.

Im Falle eines Aufrufs oder eines Sprungs zu einem Job, der die Station beinhaltet, die im laufenden I/O JOG-Betrieb aktiv ist, wird ein Alarm ausgelöst, um den Roboter- oder Stationsbetrieb anzuhalten.

Wir empfehlen, jeden Roboter oder jede Station an eine Power-On-Einheit anzuschließen. Zudem sollte für die Steuerungsgruppe, die nicht in Betrieb ist, die Servo abgeschaltet werden.

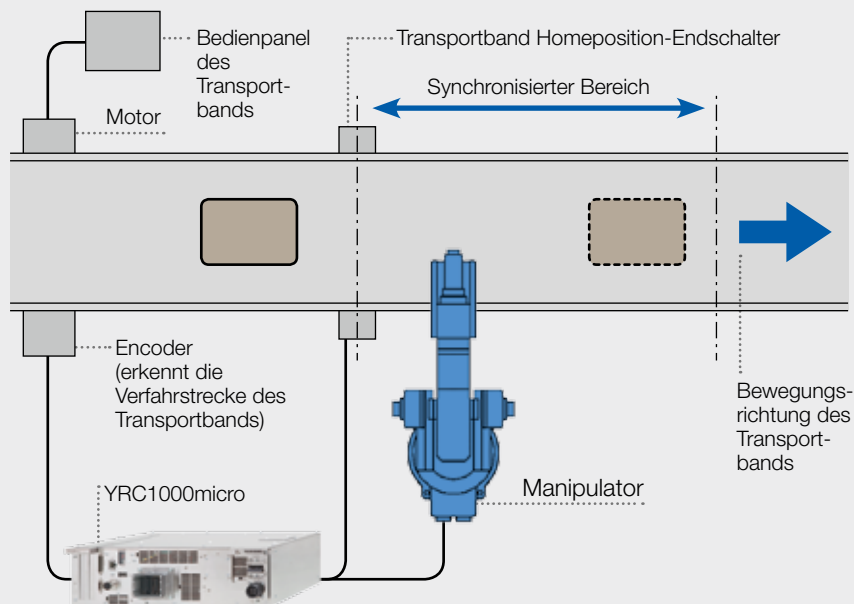
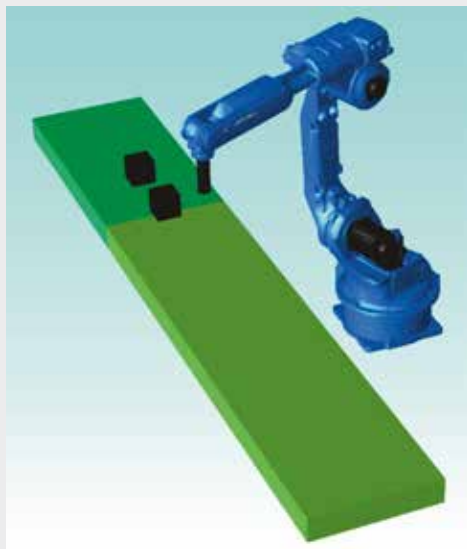
VORTEILE IM ÜBERBLICK

- Roboter oder externe Achsen können ohne Anhalten der Fertigungslinie bewegt werden
- Mit dieser Funktion können momentan ungenutzte Roboter und externe Achsen betrieben werden, die Qualität von Werkstücken geprüft oder zusätzliche vorbereitende Tätigkeiten an diesen ausgeführt werden
- Software für „I/O JOG-Operation in Play-Mode“ ist notwendig
- Diese Funktion kann nicht bei Twin-Drive-Stationen verwendet werden

YRC1000
micro

Conveyor Synchronized

(195375)



„Conveyor-synchronized“ bedeutet, dass die Bewegungen des Manipulators mit dem Transportband synchron ablaufen.

Der Manipulator bearbeitet das Werkstück während sich das Transportband bewegt.

Es gibt drei Bewegungsarten für die Transportbandnachverfolgung: Roboterachsen-Nachverfolgung, Basisachsen-Nachverfolgung und zirkuläre Nachverfolgung.

Bei der Transportbandnachverfolgung wird die Verfahrdistanz des Transportbands verwendet. Dies bedeutet, dass die Geschwindigkeit des Manipulators in Bezug auf das Werkstück immer konstant bleibt.

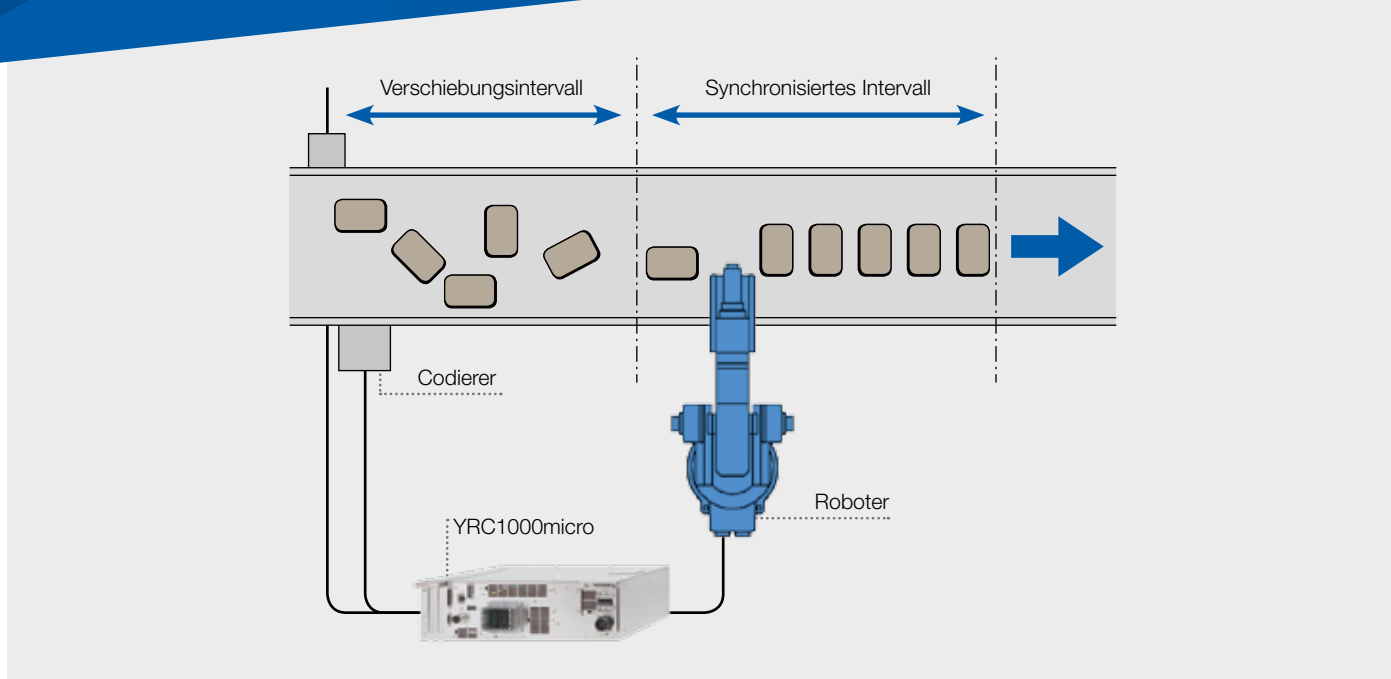
VORTEILE IM ÜBERBLICK

- Roboterbewegungen sind mit dem Transportband synchron
- Die Geschwindigkeit des Roboters in Bezug auf das Werkstück bleibt immer konstant
- Die Funktion „Conveyor Tracking“ kann ein einzelnes Werkstück vom Eingang des Signals des Endschalters an bis zum Roboternachverfolgungsvorgang bearbeiten

YRC1000
micro

Conveyor Synchronized with Shift

(195376)

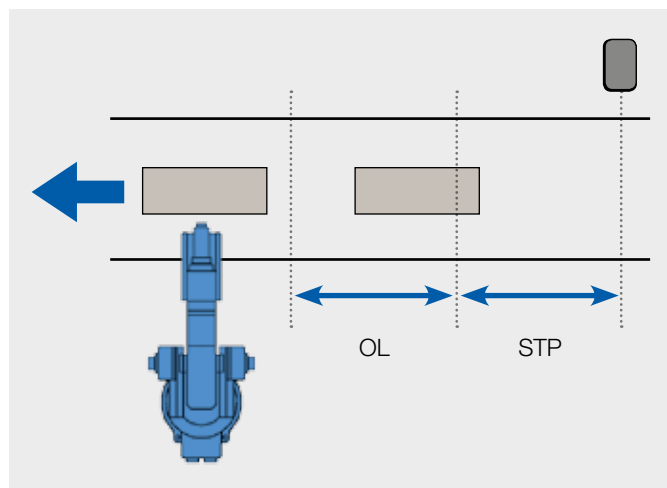


Die Funktion „Conveyor Synchronized with Shift“ speichert vorübergehend Informationen über mehrere Werkstücke, um den Nachlaufprozess in Abhängigkeit von jedem Werkstück zu ermöglichen.

Die Funktion wird verwendet, wenn eine große Anzahl von Werkstücken gleichzeitig zwischen dem Grenzscharter und dem Roboter fließt.

Beispiel:

Wenn ein Teil die STP (Startposition) erreicht, dann beginnt der Roboter mit der Synchronisierung. Wenn ein anderer Teil die STP erreicht, dann schließt der Roboter zuerst den laufenden Job am aktuellen Teil ab. Dann kehrt der Roboter in die Ausgangsstellung zurück, und beginnt die Synchronisierung des zweiten Teils, während er sich innerhalb des OL (Over Limit)-Bereichs befindet.



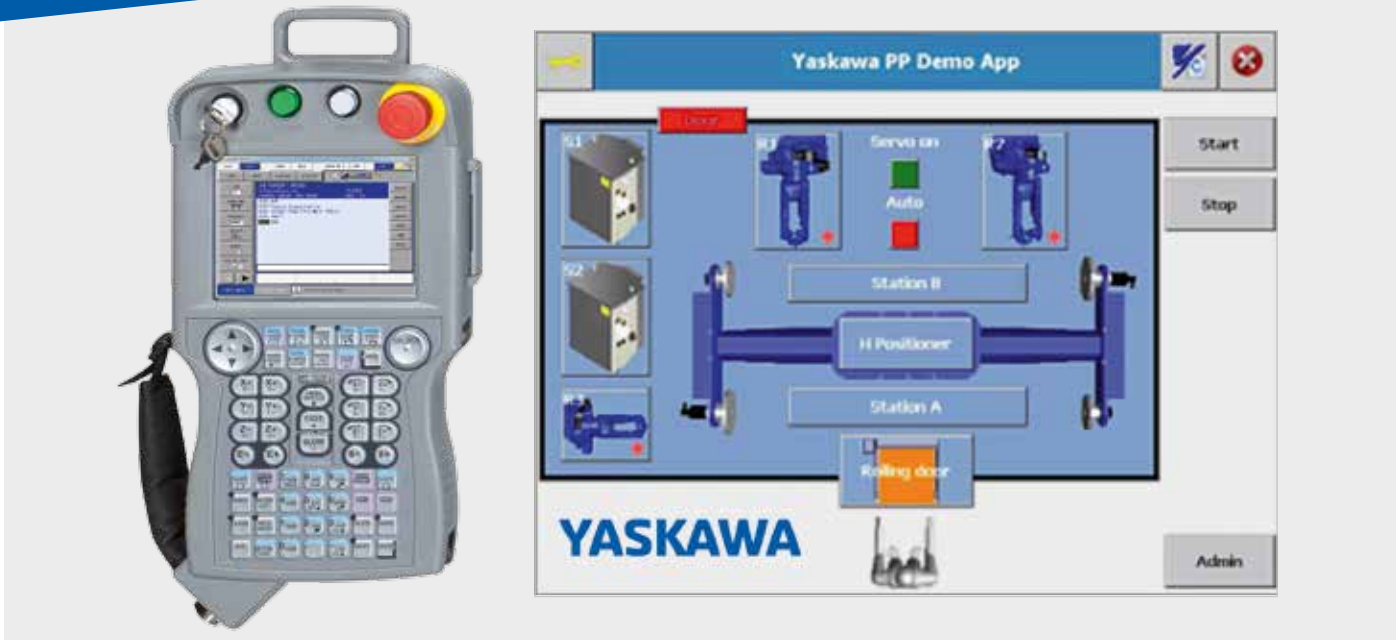
VORTEILE IM ÜBERBLICK

- Robotervorgänge werden mit dem Förderer synchronisiert
- Die Funktion kann auf Linearförderer, Rundförderer und Rundtaktisch angewandt werden
- Die Funktion kann auch auf den Fall angewandt werden, wenn mehrere Werkstücke innerhalb des Verschiebungsintervalls (maximal 99 Stücke) fließen

YRC1000
micro

Advanced PP Customization Runtime

(195357)



Die „Advanced PP Customization Runtime“ ist für die Ausführung von Anwendungen erforderlich, die mit „Advanced PP Customization SDK“ erstellt sind. Das SDK ist eine Sammlung von Modulen zur Erzeugung anwenderspezifisch angepasster Benutzeroberflächen für das Programmierhandgerät. Die anwenderspezifisch angepasste Benutzeroberfläche liegt über der standardmäßigen Benutzeroberfläche, sodass es möglich ist, bedeutende Steuerungen durchzuführen und leichter an Informationen zu gelangen und ein einzigartiges Oberflächenkonzept über mehrere Vorrichtungen zu erzeugen.

VORTEILE IM ÜBERBLICK

- Anwenderspezifisch angepasste Benutzeroberflächenelemente
- Unterstützung mehrerer Sprachen
- Ereignisbasierte Variablenhandhabung. Erhalten einer Mitteilung, wenn sich eine Variable ändert
- Visualisierung des Status von Variablen durch ihre Verbindung mit anwenderspezifisch angepassten Benutzeroberflächenelementen
- Unterstützung für unterschiedliche Benutzerebenen
- Protokollierung
- Vorlage zur Erzeugung neuer Anwendungen
- Anwendung basiert auf .NET 3.5 Compact Framework
- Visual Studio 2008 ist für die Entwicklung erforderlich
- Musteranwendung, die alle verfügbaren Merkmale implementiert

Das SDK wird mit unterschiedlichen Funktionsmodulen geliefert, die leicht an anwenderspezifisch angepasste Anwendungen angepasst werden können. Diese Module bieten Funktionalitäten wie ereignisbasierten Variablenzugang, Protokollierung, Unterstützung mehrerer Sprachen, anwenderspezifisch angepasste Benutzeroberflächenelemente usw. an.

ERFORDERLICHE SOFTWARE SDK

- „Advanced PP Customization Runtime“ – Die Laufzeit ist für jede Steuerung erforderlich, die eine anwenderspezifisch angepasste Benutzeroberfläche ausführt. 1 Laufzeitlizenz ist bereits in der „Advanced PP Customization SDK“.

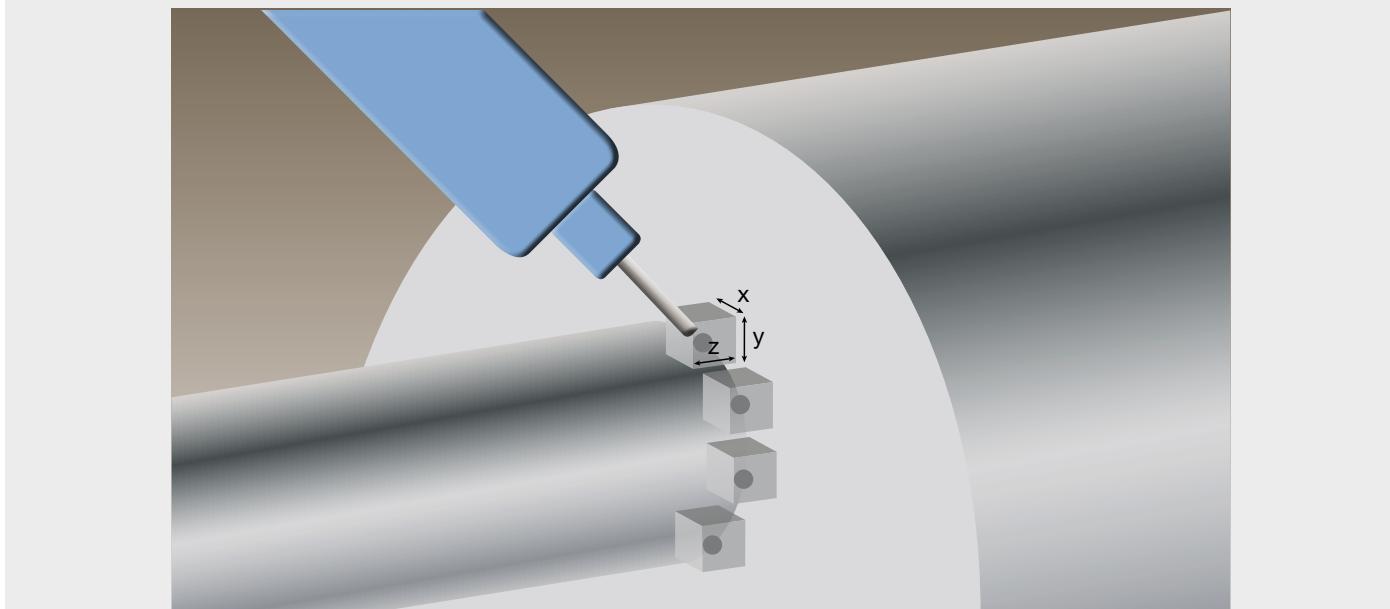
YRC1000
micro

YASKAWA

www.yaskawa.eu.com

Safety Re-teach

(195378)



Mit dieser Funktion kann der Programmierer die Anlage steuern, den Bereich, in dem der Systembetreiber einige Einstellungen des Programms vornehmen kann.

VORTEILE IM ÜBERBLICK

- Qualitätsverbesserung durch schnelle Anpassung
- Re-Teaching ist bereits durch den Bediener möglich

YRC1000
micro

Automatic Backup

**YRC1000micro Ethernet Standard
(195355)**



Die Systemdaten können kollektiv gesichert werden, wenn die intern gespeicherten Daten in einer einzelnen Datei gespeichert sind, sodass sie sofort bei einem unerwarteten Fehler (Datenverlust) geladen, und somit wieder hergestellt werden können.

Die Funktion „Automatic Backup“ bietet eine Vielzahl an Modi für den Support verschiedener Backup-Strategien.

VORTEILE IM ÜBERBLICK

- Sichert komplette Systemdaten in einer Datei
- Backup-Modi: Terminierte Backups, Backup an Modus-Schalter, Backup an Steuerungs-Start, Backup per IO
- Unterstützt diverse Backup-Speicherorte (CF-Karte, Programmierhandgerät, interne CF-Karte, interne RAM, interner USB)
- Speichern von freier Anzahl an vorherigen Versionen der gespeicherten Backup-Daten
- Fernzugriff zu Backup über FTP oder High Speed Ethernet Server
- Benachrichtigung wenn neue Backups verfügbar sind

**YRC1000
micro**

