



Inhaltsverzeichnis /Content

Inhalt

1. Funktionsweise	3
2. Voraussetzung	3
3. Aktivierung	3
4. Einrichtung.....	4
4.2 Variablen Einrichtung	4
4.2.1 out sequence (B)	4
4.2.2 in sequence (B)	4
4.2.3 in status (B)	4
4.2.4 out command (S)	4
4.2.5 out jobname (S)	4
4.2.6 in message (S).....	4
5. Roboadmin eDCI	5
5.1 Übertragungsfunktionen	5
5.2 Download / Upload zum eDCI Verzeichnis	5
6. Übertragungsbefehle roboterseitig	6
6.1 Einzelbefehle.....	6
6.1.1 LOADJ	6
6.1.2 SAVEJ	6
6.1.3 DELETEJ	6
6.1.4 LISTJJ	6
6.2 Patternfunktionsbeschreibung LOADJP, SAVEJP, DELETEJP	6
6.2.1 Pattern Syntax	6
6.2.2 Wichtige Pattern Zeichen.....	6
10. English	7
11. functionality.....	7
12. requirement.....	7
13. activation	7
14. Einrichtung.....	8
14.2 Variable settings.....	8
14.2.1 out sequence (B).....	8

yasXTouch

eDCI Function



14.2.2 in sequence (B)	8
14.2.3 in status (B)	8
14.2.4 out command (S)	8
14.2.5 out jobname (S)	8
14.2.6 in message (S).....	8
15. Roboadmin eDCI	9
15.1 Transfer functions	9
15.2 Download / Upload into eDCI folder	9
16. Transmission commands on the robot side	10
16.1 Single commands	10
16.1.1 LOADJ	10
16.1.2 SAVEJ.....	10
16.1.3 DELETEJ	10
16.1.4 LISTJJ	10
16.2 Pattern functions description LOADJP, SAVEJP, DELETEJP	10
16.2.1 Pattern syntax	10
16.2.2 Important pattern signs	10
20. Programm Beispiele/Sample	11
20.1 Hilfs Job einzel / Tool Jobs Single	11
20.1.1 SAVEJ.JBI	11
20.1.2 LOADJ.JBI.....	11
20.1.3 DELETEJ.JBI	11
20.1.4 EDCISAMPLE.JBI.....	11
20.2 Hilfs Jobs zusammen / Tool Jobs batch	12
20.2.1 SAVEJP.JBI	12
20.2.2 LOADJP.JBI.....	12
20.2.3 DELETEJP.JBI	12
20.2.4 EDCISAMPLEP.JBI.....	12
20.3 Kommunikation / Communication	12
20.3.1 EDCI_ES.JBI.....	12

yasXTouch

eDCI Function



1. Funktionsweise

Durch die eDCI Funktion ist es möglich, vom Roboter Controller durch Befehle Jobs vom YasXTouch zu laden, zu speichern und zu löschen

2. Voraussetzung

- YasXTouch Unit

3. Aktivierung

Die eDCI Funktion ist eine Option, die über eine Lizenz aktiviert wird. Zusätzlich kann ein Update an der YasXTouch Unit erforderlich sein



4. Einrichtung

4.1 Serveradmin Functions

Unter Serveradmin

eDCI

kann für jeden Controller die Variablen Zuordnung für die eDCI Communication eingerichtet werden.

eDCI

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
<input checked="" type="checkbox"/> active														
out sequence (B)	<input type="text" value="26"/>				in sequence (B)	<input type="text" value="27"/>				in status (B)	<input type="text" value="28"/>			
out command (S)	<input type="text" value="26"/>				out jobname (S)	<input type="text" value="27"/>				in message (S)	<input type="text" value="28"/>			
														<input checked="" type="checkbox"/>

4.2 Variablen Einrichtung

4.2.1 out sequence (B)

Durch Ändern des Variablenwertes in der out sequence Variable wird eine Kommunikation zwischen Controller und YasXTouch angestoßen, getriggert. Der Wert 0 schaltet den Sleepmodus ein.

4.2.2 in sequence (B)

Nach Beendigung des Befehls wird der in 4.2.1 eingegebene Wert zurückgegeben

4.2.3 in status (B)

Statusmeldungen von der eDCI Funktion

- 0: OK
- 1: sleep
- 10: get command
- 11: execute SAVEJ
- 12: execute LOADJ
- 13: execute DELETEJ
- 14: execute SAVEJP
- 15: execute LOADJP
- 16: execute DELETEJP
- 100: error
- 111: error SAVEJ
- 112: error LOADJ
- 113: error DELETEJ
- 114: error SAVEJP
- 115: error LOADJP
- 116: error DELETEJP

4.2.4 out command (S)

Hier wird der Befehl eingegeben, Einzel oder Pattern

4.2.5 out jobname (S)

Der zu verarbeitende Jobname, ohne Endung

4.2.6 in message (S)

Diese Variable zeigt im Klartext Meldungen wie Fehler oder Fehlfunktionen an



5. Roboadmin eDCI

Jobname	Controller	Server
!KPIASD.JBI	[Icons]	[Icons]
!PRODUKT1.JBI	[Icons]	[Icons]
!PRODUKT2.JBI	[Icons]	[Icons]
!PRODUKT3.JBI	[Icons]	[Icons]
!TIMER23.JBI	[Icons]	[Icons]
!UNTERJOB.JBI	[Icons]	[Icons]
*TEST.JBI	[Icons]	[Icons]
5.JBI	[Icons]	[Icons]
DCI.JBI	[Icons]	[Icons]
DCIPARALLEL.JBI	[Icons]	[Icons]

5.1 Übertragungsfunktionen



Über die Button auf der rechten Seite können die Jobs angeschaut, gelöscht und übertragen werden. Beim Löschen und Überschreiben erscheint zusätzlich eine Abfrage

Bestätigung x

!KPIASD.JBI auf dem Server überschreiben?

ja nein

ob man sicher ist.

5.2 Download / Upload zum eDCI Verzeichnis



Über diese 3 Buttons können Jobs in das eDCI Verzeichnis hochgeladen, heruntergeladen und archiviert werden. Der Disketten Symbol Button erstellt ein Zip File und speichert es auf dem lokalen Computer



6. Übertragungsbefehle roboterseitig

6.1 Einzelbefehle

6.1.1 LOADJ

Durch den Befehl LOADJ wird der in der out jobname Variable eingetragene Job von der im eDCI Ordner der YasXTouch Unit in den Roboter Controller geladen.

6.1.2 SAVEJ

Durch den Befehl SAVEJ wird der in der out jobname Variable eingetragene Job vom Roboter Controller in den eDCI Ordner der YasXTouch Unit gespeichert.

6.1.3 DELETEJ

Durch den Befehl DELETEJ wird der in der out jobname Variable eingetragene Job vom Roboter Controller gelöscht.

6.1.4 LISTJ

Der Befehl LISTJ erstellt einen Job mit der Inhaltsauflistung vom eDCI Ordner des Server. In der out jobname wird der Jobname für die Liste benannt. Die Zeilen sind dann in folgender Syntax geschrieben:

```
SET S027 "JOB123"  
SET S027 "JOB345"  
SET S027 "MAINJOB"
```

6.2 Patternfunktionsbeschreibung LOADJP, SAVEJP, DELETEJP

Durch die Patternfunktion kann die Mehrfach-Jobübertragung stark vereinfacht werden.

Es werden bestimmte, standardisierte Zeichen, für das zusammenfassen verwendet. Diese Jobs werden dann gesammelt verarbeitet.

6.2.1 Pattern Syntax

Der Syntax basiert auf <https://www.php.net/manual/de/reference.pcre.pattern.syntax.php>.

Der Pattern Befehl wird in die out jobname Variable eingetragen. In der In message Variable wird angezeigt wie viele Jobs behandelt wurden.

6.2.2 Wichtige Pattern Zeichen

- ^ - Erkennung ab Namensanfang
- \$ - Erkennung bis zu diesem Zeichen
- ^PR1 - betrifft alle Jobs mit dem Anfang PR1, z.B. PR12345678, PR1333378 und PR1AB78
- P{78|33} - betrifft alle Jobs die P78 oder P33 im Namen enthalten
- .
- 456\$ - betrifft die Jobs die mit 456 enden

yasXTouch

eDCI Function



10. English

11. functionality

The eDCI function enables the robot controller to load, save and delete jobs from the YasXTouch using commands

12. requirement

- YasXTouch Unit

13. activation

The eDCI function is an option that is activated via a license. An update to the YasXTouch unit may also be required



14. Einrichtung

14.1 Serveradmin Functions

At Serveradmin

eDCI

the variable assignment for the eDCI communication can be set up for each controller.

eDCI

1 2 3 4 5 6 7 8 9 10 11 12 13 14 15

active

out sequence (B)	<input type="text" value="26"/>	in sequence (B)	<input type="text" value="27"/>	in status (B)	<input type="text" value="28"/>
out command (S)	<input type="text" value="26"/>	out jobname (S)	<input type="text" value="27"/>	in message (S)	<input type="text" value="28"/>

14.2 Variable settings

14.2.1 out sequence (B)

By changing the variable value in the out sequence variable, communication between the controller and YasXTouch is initiated and triggered. The value 0 switches to sleep mode.

14.2.2 in sequence (B)

After the command has ended, the value entered in 4.2.1 is returned

14.2.3 in status (B)

Status messages from the eDCI Function

- 0: OK
- 1: sleep
- 10: get command
- 11: execute SAVEJ
- 12: execute LOADJ
- 13: execute DELETEJ
- 14: execute SAVEJP
- 15: execute LOADJP
- 16: execute DELETEJP
- 100: error
- 111: error SAVEJ
- 112: error LOADJ
- 113: error DELETEJ
- 114: error SAVEJP
- 115: error LOADJP
- 116: error DELETEJP

14.2.4 out command (S)

The command is entered here, single or pattern command

14.2.5 out jobname (S)

The job name to be processed, without an extension

14.2.6 in message (S)

This variable shows messages such as errors or malfunctions in plain text

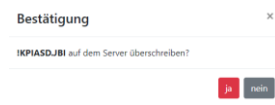


15. Roboadmin eDCI

15.1 Transfer functions



The jobs can be viewed, deleted, and transferred using the buttons on the right. A query also appears when deleting and overwriting.



15.2 Download / Upload into eDCI folder



Jobs can be uploaded, downloaded, and archived in the eDCI directory using these 3 buttons. The diskette symbol button creates a zip file and saves it on the local computer.



16. Transmission commands on the robot side

16.1 Single commands

16.1.1 LOADJ

With the LOADJ command, the job entered in the out jobname variable is loaded into the robot controller from the eDCI folder of the YasXTouch unit.

16.1.2 SAVEJ

The SAVEJ command saves the job entered in the out jobname variable from the robot controller in the eDCI folder of the YasXTouch unit.

16.1.3 DELETEJ

The job entered in the variable out jobname is deleted from the robot controller by the DELETEJ command.

16.1.4 LISTJ

The LISTJ command creates a job with the content listing from the server's eDCI folder. The job name for the list is specified in the out jobname. The lines are then written in the following syntax:

```
SET S027 "JOB123"  
SET S027 "JOB345"  
SET S027 "MAINJOB"
```

16.2 Pattern functions description LOADJP, SAVEJP, DELETEJP

The pattern function can greatly simplify multiple job transmission.

Certain, standardized characters are used to summarize. These jobs are then processed collectively.

16.2.1 Pattern syntax

The syntax is based on <https://www.php.net/manual/de/reference.pcre.pattern.syntax.php>.

The pattern command is entered in the out jobname variable. The In message variable shows how many jobs have been handled.

16.2.2 Important pattern signs

- ^ - Recognition from the beginning of the name
- \$ - Recognition up to this sign
- ^PR1 - affects all jobs with the beginning PR1, z.B. PR12345678, PR1333378 und PR1AB78
- P(78|33) - affects all jobs that contain P78 or P33 in their names
- .
- 456\$ - affects the jobs that end with 456



20. Programm Beispiele/Sample

20.1 Hilfs Job einzel / Tool Jobs Single

Mit Call im übergeordneten Job aufrufen, wenn ein Job in den Server gesichert werden soll.
Call from a higher-level job if a job has to be saved onto the server.

20.1.1 SAVEJ.JBI

```
NOP
SET S023 "SAVEJ"
CALL JOB: EDCI_ES
END
```

20.1.2 LOADJ.JBI

```
NOP
SET S023 "LOADJ"
CALL JOB: EDCI_ES
END
```

20.1.3 DELETEJ.JBI

```
NOP
SET S023 "DELETEJ"
CALL JOB: EDCI_ES
END
```

20.1.4 EDCISAMPLE.JBI

Beispiel Ablauf zum Speichern, Löschen und Laden eines Jobs
Example sequence for saving, deleting, and loading a job

```
NOP
'INIT and Com. check
SET B023 0
WAIT B024 = 0
WAIT B025 = 1
'
SET S024 "TYPE_A"
CALL JOB: SAVEJ
'
SET S024 " TYPE_A"
CALL JOB: DELETEJ
'
SET S024 " TYPE_B"
CALL JOB: LOADJ
'
SET B023 0
END
```



20.2 Hilfs Jobs zusammen / Tool Jobs batch

Mit Call im übergeordneten Job aufrufen, wenn ein Job in den Server gesichert werden soll.
Call from a higher-level job if a job has to be saved onto the server.

20.2.1 SAVEJP.JBI

```
NOP
SET S023 "SAVEJP"
CALL JOB: EDCI_ES
END
```

20.2.2 LOADJP.JBI

```
NOP
SET S023 "LOADJP"
CALL JOB: EDCI_ES
END
```

20.2.3 DELETEJP.JBI

```
NOP
SET S023 "DELETEJP"
CALL JOB: EDCI_ES
END
```

20.2.4 EDCISAMPLEP.JBI

Beispiel Ablauf zum Speichern, Löschen und Laden einer Jobsgruppe, alle mit dem Zeichen PR1 werden gesichert und gelöscht, alle mit PR2 werden geladen. In der Variable out Jobname wird die Menge der bearbeiteten Jobs angezeigt
Example sequence for saving, deleting, and loading a job group, all with the character PR1 will be saved and deleted, all with PR2 will be loaded. The Value of the involved Jobs will displayed in the out Jobname Variable

```
NOP
'INIT and com. check
SET B023 0
WAIT B024 = 0
WAIT B025 = 1
'
SET S024 "PR1"
CALL JOB: SAVEJP
'
SET S024 " PR1"
CALL JOB: DELETEJP
'
SET S024 " PR2"
CALL JOB: LOADJP
'
SET B023 0
END
```

20.3 Kommunikation / Communication

Übertragungslogik
Communication Logic

20.3.1 EDCI_ES.JBI

```
NOP
INC B023
WAIT B023=B024 T=300.00
JUMP *COM_OK IF B023=B024
SETUALM 8050 "TIME OUT" B025
*COM_OK
JUMP *OK IF B025=0
SETUALM 8050 "Message in S25" B025
*OK
RET
END
```