



OrangeApps

Pointloader 1.0

Anwender Dokumentation

Stand: 14.06.2018, Version 1.2

© Copyright 2017

OrangeApps GmbH
Arnikaweg 1
87471 Durach
Deutschland
www.OrangeApps.de

Diese Dokumentation darf –auch auszugsweise– vervielfältigt und Dritten zugänglich gemacht werden. Bei der auszugsweisen Vervielfältigung muss jedoch ein Verweis auf den Copyright Inhaber sowie dieses Dokument vermerkt werden.

Der Inhalt der Druckschrift wurde mit der beschriebenen Software geprüft. Dennoch können Abweichungen nicht ausgeschlossen werden, so dass für die vollständige Übereinstimmung keine Gewähr übernommen werden kann.

Historie der Dokumentenversionen

Version	Datum	Autor	Änderungsgrund / Bemerkung
1.0	8.11.17	Christian Mayer	Ersterstellung für KRC2 und KRC4
1.1	18.05.2018	Christian Mayer	Kommando HALT eingefügt
1.2	16.4.2018	Christian Mayer	Kap. 4 überarbeitet, Kap. 9.1.1 eingefügt

Inhalt

1	Einleitung.....	6
1.1	Zielgruppe.....	6
1.2	Darstellung von Hinweisen.....	6
1.3	Verwendete Begriffe	6
1.4	Zeichen und Schriftarten	7
1.5	Markenzeichen	7
2	Produktbeschreibung	8
2.1	Merkmale	8
2.2	Lieferumfang	8
2.3	Einsatzgebiet / -umgebung	8
2.4	CPC	8
2.5	Voraussetzungen zum Betrieb der Software	9
3	Sicherheit.....	10
4	Installation, Deinstallation, Update	11
4.1	KRC4	11
4.1.1	Systemvoraussetzungen für die Ausführung	11
4.1.2	Pointloader installieren oder auf neue Version updaten	11
4.1.3	Pointloader deinstallieren	12
4.1.4	KSS Update	12
4.1.5	Installierte Dateien	13
4.2	KRC2	14
4.2.1	Systemvoraussetzungen für die Ausführung	14
4.2.2	PointLoader installieren oder auf neue Version updaten.....	14
4.2.3	Pointloader deinstallieren	15
4.2.4	KSS Update	15
4.2.5	Installierte Dateien	16
5	Lizenzierung	17
5.1	Lizenz-Datei verwenden	17
5.1.1	Vor der Installation von Pointloader	17
5.1.2	Nach der Installation von Pointloader	17

6	Quickstart	18
7	Funktionsweise	19
7.1	Übersicht	19
7.2	KRL Modul "RunPointLoader"	20
7.3	KRL Modul "PointLoaderLoad"	20
7.4	KRL-Modul "PointloaderMain"	20
7.5	KRL-Modul "PointloaderUser"	21
7.6	CAD / CAM Datei	21
7.6.1	Dateiname	21
7.6.2	Speicherort	21
7.6.3	Verfügbare Kommandos	21
7.6.3.1	Kommandos für Bewegungen	22
7.6.3.2	Kommandos für Bewegungsparameter	23
7.6.3.2.1	\$VEL.CP	23
7.6.3.2.2	\$ACC.CP	24
7.6.3.2.3	\$VEL_AXIS	25
7.6.3.2.4	\$ACC_AXIS	25
7.6.3.2.5	\$VEL_EXTAX	26
7.6.3.2.6	\$ACC_EXTAX	26
7.6.3.2.7	\$Vel.Ori1 and \$Vel.Ori2	27
7.6.3.2.8	\$Acc.Ori1 and \$Acc.Ori2	27
7.6.3.2.9	\$Advance	28
7.6.3.2.10	\$IPO_Mode	28
7.6.3.2.11	\$APO.CDIS	29
7.6.3.2.12	\$ORI_TYPE	29
7.6.3.2.13	\$OV_PRO	30
7.6.3.2.14	BAS	30
7.6.3.3	Kommandos für Tool und Base	31
7.6.3.3.1	ToolBase	31
7.6.3.3.2	Tool	31
7.6.3.3.3	Base	32
7.6.3.4	Kommandos für Ein- und Ausgänge	34
7.6.3.4.1	Warte auf Eingang	34
7.6.3.4.2	SetOutPut	34
7.6.3.4.3	SetOutPutB	35
7.6.3.4.4	SetOutPutW	36
7.6.3.4.5	\$ANOUT	36
7.6.3.5	Weitere Kommandos	37
7.6.3.5.1	Interrupt ON/OFF	37
7.6.3.5.2	HALT	37
7.6.3.6	Anwenderkommandos	38
7.6.3.6.1	SpindleVel	38
7.6.3.6.2	SpindleStart	39
7.6.3.6.3	SpindleStop	39
7.6.3.6.4	Cooling	40
7.6.3.6.5	Vaccum	41
7.6.3.6.6	ToolChange	41
7.6.3.6.7	UserPrg and UserPrgTr	42
8	Bedienung.....	46
8.1	Start eines Bearbeitungsprogramms	46

9	HMI	48
9.1	HMI KRC4.....	48
9.1.1	Dateiauswahl-Dialog.....	50
9.1.2	User Technologie Variablen	50
9.1.2.1	Beschriftung der Variablen in der HMI	51
9.2	HMI KRC2.....	53
10	Log-Funktion	55
11	Beispiele für Kommandos in der CAD/CAM Datei	56
12	Meldungen	58

1 Einleitung

1.1 Zielgruppe

Diese Dokumentation richtet sich an Anwender mit folgenden Kenntnissen:

- Fortgeschrittene Systemkenntnisse der Robotersteuerung
- Fortgeschrittene KRL-Programmierkenntnisse

1.2 Darstellung von Hinweisen



Diese Hinweise bedeuten, dass Tod oder schwere Körperverletzungen sicher oder sehr wahrscheinlich eintreten werden, wenn keine Vorsichtsmaßnahmen getroffen werden.



Diese Hinweise bedeuten, dass Tod oder schwere Körperverletzungen eintreten **können**, wenn keine Vorsichtsmaßnahmen getroffen werden.



Diese Hinweise bedeuten, dass leichte Körperverletzungen eintreten **können**, wenn keine Vorsichtsmaßnahmen getroffen werden.



Diese Hinweise bedeuten, dass Sachschäden eintreten **können**, wenn keine Vorsichtsmaßnahmen getroffen werden.



Diese Hinweise enthalten nützliche Tipps oder besondere Informationen für das aktuelle Thema.

1.3 Verwendete Begriffe

Begriff	Beschreibung
HMI	Die Human-Machine Interface (HMI) ist eine Schnittstelle, über die ein Mensch mit einer Maschine kommuniziert.
KSS	KUKA Systemsoftware
smartPad	Roboter Bediengerät
KRL	KUKA Robot Language

1.4 Zeichen und Schriftarten

In den Syntaxbeschreibungen werden folgende Zeichen und Schriftarten verwendet:

Syntax-Element	Darstellung
KRL-Code	- Schriftart Courier New - Großschreibung Beispiele: GLOBAL; ANIN ON; OFFSET
Elemente, die durch programmspezifische Angaben ersetzt werden müssen	- Kursiv - Groß-/Kleinschreibung Beispiele: <i>Strecke</i> ; <i>Zeit</i> ; <i>Format</i>
Optionale Elemente	In spitzen Klammern Beispiel: <STEP <i>Schrittweite</i> >
Elemente, die sich gegenseitig ausschließen	Getrennt durch das Zeichen " " Beispiel: IN OUT

1.5 Markenzeichen

. **NET Framework** ist eine Marke der Microsoft Corporation.

Windows ist eine Marke der Microsoft Corporation.

2 Produktbeschreibung

Pointloader ist ein Werkzeug um im Nachladebetrieb Bewegungspunkte und Technologiebefehle aus extern erzeugten CAD/CAM-Dateien kontinuierlich während der Programmabarbeitung in die Robotersteuerung zu laden und abzufahren. Das Softwarepaket besteht im Prinzip aus vier Teilen:

1. Plugin für das Einlesen und Laden der Kommandos aus der CAD / CAM Datei
2. Eine HMI zur Anzeige verschiedener Parameter
3. KRL-Datei zur Interaktion mit dem Plugin und zum Aufruf der Bewegungs- und Technologieroutinen
4. KRL-Datei mit Technologieroutinen

2.1 Merkmale

- Speicheroptimiertes Laden einer CAD / CAM Datei (*.src oder *.txt) in die Robotersteuerung mit mehr als 3 Mio. Punkten
- Bewegungsarten PTP, LIN, CIRC, Joint, SPTP, SLIN, SCIRC
- Angabe von externen Achsen E1-E6 in der CAD / CAM Datei wird unterstützt
- Zeitoptimiertes kontinuierliches Laden und Abfahren der Bewegungspunkte im KRL-Programm (50 Punkte < 500ms)
- Aus dem Hauptmenü und aus dem Bearbeitungsprogramm aufrufbare HMI zur Interaktion mit dem Anlagenbediener
- Technologiebefehle für Spindle-Geschwindigkeit, Kühlung, Absaugung, Ausgänge, Werkzeugwechsel
- frei parametrierbare Geschwindigkeits-Parameter (zur Laufzeit änderbar)
- HMI mit Restzeitabschätzung
- Mehrsprachigkeit der HMI (Auslieferung Deutsch und Englisch, vom Anwender erweiterbar)
- automatisches Setup zur Installation und Deinstallation
- vollständige Wiederinstallation bei KSS Update
- Lizenzmechanismus zum Betrieb der Software auf ausgewählten Robotern
- Lauffähig unter KSS 8.2 / 8.3 (KRC4) und KSS 5.5 / 5.6 (KRC2)
- Ausführbar in den Betriebsarten T1, T2, AUT, EXT

2.2 Lieferumfang

Die Lieferung erfolgt als Technologie-Paket zur einfachen Installation am Roboter. Darin sind alle zur Installation und Betrieb notwendigen Komponenten enthalten.

2.3 Einsatzgebiet / -umgebung

Für die Steuerungen KRC2 mit KSS Version 5.5/5.6 und KRC4 mit KSS Version 8.2/8.3 existiert jeweils eine Software Version.

2.4 CPC

Soll die Software auf Robotern mit CPC-Schutz eingesetzt werden, wird vor der Installation ein CPC-Zertifikat benötigt. Dies kann auf Kundenwunsch von KUKA Roboter kostenpflichtig erstellt werden.

2.5 Voraussetzungen zum Betrieb der Software

Hardware

- KUKA KRC4 Roboter
- KUKA KRC2 Roboter

Software

- KUKA KSS 8.2/8.3 (KRC4) oder KSS 5.5/5.6 (KRC2 ed05)
- Runtime-Lizenz

3 Sicherheit

Die im vorliegenden Dokument beschriebene Software dient zum Nachladen von extern erzeugten CAD / CAM Dateien in KUKA Industrieroboter der Generation KRC2 und KRC4. Sie ist nach dem Stand der Technik entwickelt.

Die Software darf nur bestimmungsgemäß, sicherheits- und gefahrenbewusst verwendet werden. Die Benutzung muss unter Beachtung dieses Dokuments und den Lizenzvereinbarungen erfolgen.

Unbefugte Verwendung kann zu Schäden am Roboter, der Umwelt und der Gesundheit von Personen führen.

4 Installation, Deinstallation, Update

Die Installation erfolgt über die Option *Zusatzsoftware*. Diese befindet sich im Hauptmenü unter *Inbetriebnahme*.



Vor der Installation eine Dateisicherung des Roboters erstellen!

4.1 KRC4

4.1.1 Systemvoraussetzungen für die Ausführung

Mindestanforderungen Software

- KUKA System Software 8.2, 8.3



Falls auf dem Roboter KUKA.CPC verwendet wird, wird zur Installation des Plugin ein Softwarezertifikat benötigt. Bitte halten Sie in diesem Fall, vor dem Kauf der Software, Rücksprache mit unserem Kundenservice (Email an info@orangeapps.de)

4.1.2 Pointloader installieren oder auf neue Version updaten

Voraussetzung

- Benutzergruppe Experte
- Betriebsart T1, T2

Zur Installation auf den drei Systemen Realer Roboter, OfficePC und OfficeLite wie folgt vorgehen:

Vorgehensweise

1. Die .zip Datei entpacken
2. Den Ordner OrangeApps.Pointloader mit den enthaltenen Dateien entweder auf einen USB-Stick oder direkt auf ein Laufwerk des Zielsystems kopieren (z.B. d:\).
3. Bei der Installation von einem USB-Stick, diesen an den Steuerungs-PC oder das smartPad anschließen.
4. Im Hauptmenü unter *Inbetriebnahme* → *Zusatzsoftware* auswählen.
5. Auf den Button **Neue Software** klicken.
6. Es erscheint eine Liste für die zur Installation zur Verfügung stehender Software. Sollte in der Liste kein Eintrag mit **OrangeApps.Pointloader** aufgeführt sein, auf **Aktualisieren** klicken. Wird nun der Eintrag angezeigt, mit Schritt 9 fortfahren.
7. Sollte der Eintrag nicht angezeigt werden, muss das Laufwerk, von dem installiert werden soll, erst konfiguriert werden. Dazu **Konfiguration** wählen. In einem neuen Fenster besteht nun die Möglichkeit den Pfad auswählen unter dem der Ordner **OrangeApps.Pointloader** zu finden ist.
8. Im Bereich **Installationspfade für Optionen** eine leere Zelle markieren und **Pfadauswahl** wählen. Die vorhandenen Laufwerke werden angezeigt. Das Laufwerk, an dem der Ordner **OrangeApps.Pointloader** zur Verfügung steht, markieren und die

Auswahl mit **Speichern** speichern. Das Fenster schließt sich wieder. In der Liste sollte nun ein Eintrag **OrangeApps.Pointloader** erscheinen. Ist dies nicht der Fall, auf **Aktualisieren** drücken und/oder wiederholen die Schritte 7 und 8 wiederholen.

9. Den Eintrag **OrangeApps.Pointloader** markieren und auf **Installieren** drücken. Die Sicherheitsabfrage mit **Ja** bestätigen.
10. Die eigentliche Installation wird nun vorbereitet. Um die endgültige Installation auszuführen, muss der Steuerungs-PC neu gestartet werden. Dies kann sofort durch Anklicken von **Steuerungs-PC jetzt neu starten** oder später durch Anklicken von **Später** erfolgen.
11. Falls **Später** gewählt, wird das Fenster geschlossen und es muss zur endgültigen Installation mit Schritt 12 fortgefahren werden. Falls **Steuerungs-PC jetzt neu starten** gewählt wird, schließt das Fenster und der Steuerungs-PC wird automatisch neu gestartet. Schritt 13 wird ausgeführt.
12. Mit **Herunterfahren** im Hauptmenü einen Neustart des Steuerungs-PC durchführen.
13. Beim erneuten Hochlauf der Steuerung wird die Software **Pointloader** auf dem Rechner installiert.
14. Gegebenenfalls den USB-Stick abziehen.

4.1.3 Pointloader deinstallieren

Voraussetzung

- Benutzergruppe Experte
- Betriebsart T1, T2

Vorgehensweise

1. Im Hauptmenü unter **Inbetriebnahme** → **Zusatzsoftware** auswählen.
2. Den Eintrag **OrangeApps.Pointloader** markieren und auf **Deinstallieren** drücken. Die Sicherheitsabfrage mit **Ja** beantworten. Die Deinstallation wird vorbereitet. Um die endgültige Deinstallation auszuführen muss der Steuerungs-PC neu gestartet werden. Dies kann sofort durch Anklicken von **Steuerungs-PC jetzt neu starten** oder später durch Anklicken von **Später** erfolgen.
3. Falls **Später** gewählt, wird das Fenster geschlossen und es muss zur endgültigen Deinstallation mit Schritt 4 fortgefahren werden. Falls **Steuerungs-PC jetzt neu starten** gewählt wird, schließt das Fenster und der Steuerungs-PC wird automatisch neu gestartet. Schritt 5 wird ausgeführt.
4. Mit **Herunterfahren** im Hauptmenü einen Neustart des Steuerungs-PC durchführen.
5. Beim erneuten Hochlauf der Steuerung wird die Software **OrangeApps.Pointloader** auf dem Rechner deinstalliert.

4.1.4 KSS Update

Bei einem Update der KSS Software innerhalb der KSS Versionen 8.2/8.3 erfolgt eine automatische Wiederinstallation von Pointloader. Der aktuelle Stand der KRL-Module wird gespeichert und nach dem KSS Update automatisch wieder hergestellt.

4.1.5 Installierte Dateien

Zum Betrieb der Software werden folgende Dateien installiert:

Ordner	Dateien	Funktion
C:\KRC\SmartHMI	SmartHMI.exe.Pointloader.config Pointloader.dll	Plugin
C:\KRC\DATA	Pointloader.kxr PointloaderUser.kxr	Sprachdatenbank für Plugin Sprachdatenbank für User-Parameter
C:\KRC\ROBOTER\KRC\R1\TP\PointLoader	PointLoaderLoad (src und dat) PointLoaderMain (src und dat) PointLoaderData.dat	KRL-Bewegungsmodul
C:\KRC\ROBOTER\KRC\R1\TP\PointLoader	PointloaderUser (src und dat)	"User-Modul" Interface für Anwenderparameter und Aktionen
C:\KRC\ROBOTER\KRC\R1\Program	RunPointLoader (src und dat)	Anwenderprogramm zum Start der Abarbeitung
D:\	Example.src Example.txt	CAD/CAM Datei mit Beispielen der verfügbaren Kommandos

4.2 KRC2

4.2.1 Systemvoraussetzungen für die Ausführung

Mindestanforderungen Hardware/Software

- KRC2: KUKA System Software 5.5/5.6



Falls auf dem Roboter KUKA.CPC verwendet wird, wird zur Installation des Plugin ein Softwarezertifikat benötigt. Bitte halten Sie in diesem Fall, vor dem Kauf der Software, Rücksprache mit unserem Kundenservice (Email an info@orangeapps.de)

4.2.2 PointLoader installieren oder auf neue Version updaten

Voraussetzung

- Benutzergruppe Experte
- Betriebsart T1, T2

Zur Installation auf den drei Systemen Realer Roboter, OfficePC und OfficeLite gehen Sie wie folgt vor:

Vorgehensweise KRC2

Die Installation erfolgt über **Inbetriebnahme → Zusatzsoftware installieren** im Hauptmenü.

1. Entpacken Sie die .zip Datei
2. Kopieren sie den erhaltenen Installationsordner **OrangeApps.Pointloader** mit den enthaltenen Dateien entweder auf einen USB-Stick oder direkt auf ein Laufwerk des Zielsystems (z.B. d:\).
3. Falls Sie bereits im Besitz einer gültigen Lizenzdatei sind, kopieren Sie diese zu den Dateien im Installationsordner. Die Lizenzdatei wird beim Setup automatisch erkannt und installiert. Alternativ haben Sie die Möglichkeit die Lizenzdatei nach erfolgter Installation manuell zu installieren.
6. Bei der Installation von einem USB-Stick, schließen sie diesen an den Steuerungs-PC an.
7. Wählen Sie im Hauptmenü unter **Inbetriebnahme → Zusatzsoftware installieren** aus.
8. Klicken Sie auf die Schaltfläche **Neue SW**.
9. Sie erhalten eine Liste für die Installation zur Verfügung stehender Software. Sollte in der Liste kein Eintrag mit **OrangeApps.Pointloader** aufgeführt sein, klicken Sie auf **Aktualisieren**. Wird nun der Eintrag angezeigt, machen Sie weiter mit Schritt 10.
10. Sollte der Eintrag nicht angezeigt werden, muss das Laufwerk von dem installiert werden soll, erst konfiguriert werden. Wählen Sie dazu **Konfig**. In einem neuen Fenster haben Sie nun die Möglichkeit den Pfad auswählen unter dem der Ordner **OrangeApps.Pointloader** zu finden ist.
11. Markieren Sie im Bereich **InstallTech – Installationspfade konfigurieren** eine leere Zelle und wählen Sie **Pfadauswahl**. Die vorhandenen Laufwerke werden angezeigt. Markieren Sie das Laufwerk an dem der Ordner **OrangeApps.Pointloader** zur Verfügung steht und speichern Sie Ihre Auswahl mit **Übernehmen**. Das Fenster schließt sich wieder. In der Liste sollte nun ein Eintrag **OrangeApps.Pointloader**

erscheinen. Ist dies nicht der Fall, drücken Sie auf **Aktualisieren** und/oder wiederholen Sie die Schritte 7 und 8. Schließen Sie das Fenster mit **Übernehmen**.

12. Markieren Sie in der Liste **InstallTech - Auswahl** den Eintrag **OrangeApps.Pointloader** und drücken Sie auf **Installieren**. Bestätigen Sie die Sicherheitsabfrage mit **Ja**.
13. Lesen Sie die Lizenzbestimmungen aufmerksam durch. Erklären Sie ihr Einverständnis mit den Lizenzbedingungen durch Anklicken von **Akzeptieren** und setzen sie die Installation mit **Weiter** fort. Sind Sie mit den Lizenzbestimmungen nicht einverstanden, brechen Sie die Installation mit **Abbrechen** ab. **Verwenden Sie zur Bedienung die Software-Funktionstasten unten am KCP (3.Taste=Akzeptieren, 4.Taste=Weiter, 5.Taste=Abbrechen)**
14. Die eigentliche Installation wird vorbereitet. Um die endgültige Installation auszuführen muss der Steuerungs-PC neu gestartet werden. Dies kann sofort durch Anklicken von **Steuerungs-PC jetzt neu starten** oder später durch Anklicken von **Später** erfolgen.
15. Wenn Sie **Später** wählen, wird das Fenster geschlossen und Sie müssen zur endgültigen Installation mit Schritt 14 fortfahren. Falls Sie **Steuerungs-PC jetzt neu starten** wählen, wird das Fenster geschlossen und der Steuerungs-PC automatisch neu gestartet. Schritt 15. wird ausgeführt.
16. Führen Sie unter **Herunterfahren** im Hauptmenü einen Neustart des Steuerungs-PC durch.
17. Beim erneuten Hochlauf der Steuerung wird die Software **UserLogonIO** auf dem Rechner installiert.
18. Ziehen Sie gegebenenfalls den USB-Stick ab.

4.2.3 Pointloader deinstallieren

Voraussetzung

- Benutzergruppe Experte
- Betriebsart T1, T2

Vorgehensweise

1. Im Hauptmenü unter **Inbetriebnahme** → **Zusatzsoftware installieren** auswählen.
2. Den Eintrag **OrangeApps.Pointloader** markieren und auf **Deinstallieren** drücken. Die Sicherheitsabfrage mit **Ja** beantworten. Die Deinstallation wird vorbereitet. Um die endgültige Deinstallation auszuführen muss der Steuerungs-PC neu gestartet werden. Dies kann sofort durch Anklicken von **Steuerungs-PC jetzt neu starten** oder später durch Anklicken von **Später** erfolgen.
3. Falls **Später** gewählt, wird das Fenster geschlossen und es muss zur endgültigen Deinstallation mit Schritt 4 fortgefahren werden. Falls **Steuerungs-PC jetzt neu starten** gewählt wird, schließt das Fenster und der Steuerungs-PC wird automatisch neu gestartet. Schritt 5 wird ausgeführt.
4. Mit **Herunterfahren** im Hauptmenü einen Neustart des Steuerungs-PC durchführen.
5. Beim erneuten Hochlauf der Steuerung wird die Software **OrangeApps.Pointloader** auf dem Rechner deinstalliert.

4.2.4 KSS Update

Bei einem Update der KSS Software innerhalb der KSS Versionen 5.5/5.6 erfolgt eine automatische Wiederinstallation von Pointloader. Der aktuelle Stand der KRL-Module wird gespeichert und nach dem KSS Update automatisch wieder hergestellt.

4.2.5 Installierte Dateien

Zum Betrieb der Software werden folgende Dateien installiert:

Ordner	Dateien	Funktion
C:\KRC\TP\Pointloader	PointLoader.dll PointLoader.mdb	Plugin Sprachdatenbank
C:\KRC\ROBOTER\KRC\R1\TP\PointLoader	PointLoaderLoad (src and dat) PointLoaderMain (src and dat) PointLoaderData.dat	Technologiemodule
C:\KRC\ROBOTER\KRC\R1\TP\PointLoader	PointLoaderUser (src and dat)	"User-Modul" Interface für Anwenderparameter und Aktionen
C:\KRC\R1\TP\PointLoader\Lic	PointLoader_*.lic	Lizenz
C:\KRC\ROBOTER\KRC\R1\Program	RunPointLoader (src und dat)	Anwenderprogramm zum Start der Abarbeitung
D:\	Example.src Example.txt	CAD/CAM Datei mit Beispielen zu allen verfügbaren Kommandos

5 Lizenzierung

Um Pointloader nutzen zu können, muss für jede Roboter-Seriennummer eine Lizenz-Datei erstellt werden. Für OfficePC und OfficeLite ist ebenfalls eine Lizenz-Datei erforderlich.

Merkmale

- Der Lizenzmechanismus enthält eine zeitbegrenzte Notlaufeigenschaft für den Fall eines Robotertausches. Innerhalb von 14 Tagen muss zur weiteren Nutzung eine neue Lizenz-Datei erstellt werden

5.1 Lizenz-Datei verwenden

5.1.1 Vor der Installation von Pointloader

Steht bereits vor der Installation eine Lizenzdatei zur Verfügung, kann Sie diese dem Setup beigelegt werden. Die Lizenz wird dann automatisch installiert.

5.1.2 Nach der Installation von Pointloader

Steht die Lizenzdatei erst nach der Installation zur Verfügung, wie folgt vorgehen:

- Die Lizenz-Datei in folgendes Verzeichnis auf dem Roboter-System kopieren:
„C:\KRC\TP\PointLoader\Lic“

6 Quickstart

Das Program „RunPointLoader“ ist das Anwenderprogramm. Dieses Programm lädt weitere Module um die CAD/CAM Datei zu laden und abzuarbeiten.

Vorgehensweise KRC4:

- CAD / CAM Datei entsprechend den verfügbaren Kommandos erstellen und auf ein vom Roboter erreichbares Laufwerk kopieren.
- Die HMI (Hauptmenü “PointLoader → halb- oder ganzseitig”)



- Mit der Schaltfläche den Datei-Auswahl-Dialog öffnen um die CAD/CAM Datei auszuwählen

➔ Das Anwenderprogram „RunPointLoader“ wird automatisch angewählt

- Die Betriebsart auswählen und den Ablauf starten

Vorgehensweise KRC2:

- CAD / CAM Datei entsprechend den verfügbaren Kommandos erstellen und auf ein vom Roboter erreichbares Laufwerk kopieren.
- Die HMI öffnen (Menü “Anzeige → PointLoader)
- Mit der Schaltfläche „Datei öffnen“ den Datei-Auswahl-Dialog öffnen um die CAD/CAM Datei auszuwählen

➔ Das Anwenderprogram „RunPointLoader“ wird automatisch angewählt

- Die Betriebsart auswählen und den Ablauf starten



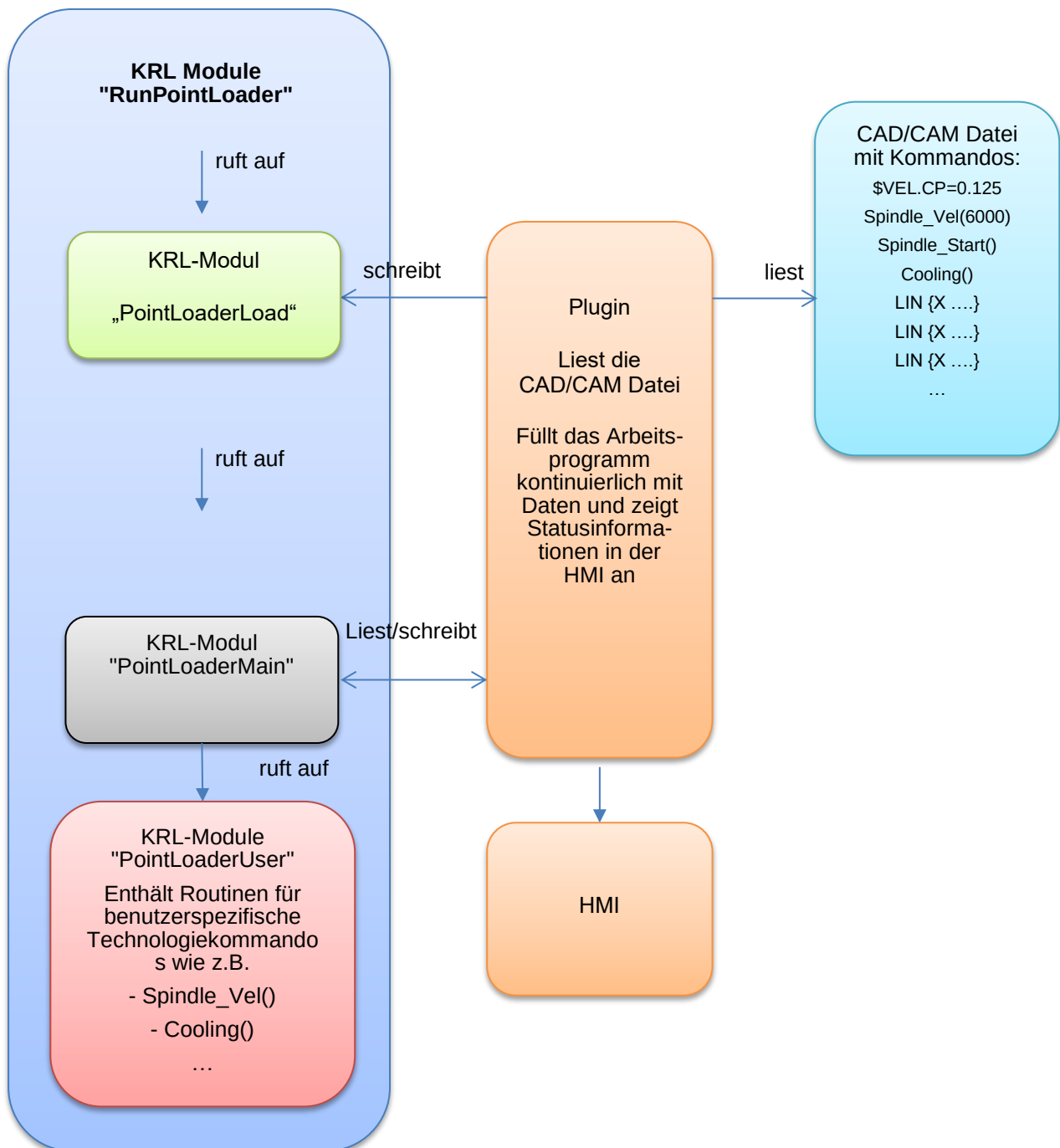
Nach jedem Laden einer CAD/CAM Datei werden alle nicht ausführbaren Kommandos in die Datei D:\PLSkippedLines.txt geschrieben.

7 Funktionsweise

7.1 Übersicht

Die Software besteht aus KRL-Modulen und einem Plugin (dll) welche miteinander interagieren. Die Bewegungen und Befehle erfolgen im KRL-Modul welches vom Plugin kontinuierlich mit Kommandos gefüllt wird. Die Kommandos werden vom Plugin vor Beginn der Bewegungen zeilenweise aus einer CAD / CAM Datei eingelesen und in einer internen Liste gespeichert. Nicht unterstützte Kommandos werden übersprungen.

Schematische Darstellung



7.2 KRL Modul "RunPointLoader"

Dies ist das Anwenderprogramm. Von dort werden die Module "PointLoaderLoad" und "PointLoaderMain" aufgerufen.

7.3 KRL Modul "PointLoaderLoad"

Dieses Modul lädt die CAD / CAM Datei in den Arbeitsspeicher.

Bei der KRC4 steht zusätzlich ein zusätzlicher Parameter zur Verfügung. Mit diesem Parameter kann angegeben werden ob und in welcher Größe die HMI automatisch geöffnet werden soll.

Die Variable "FilePath[]" spezifiziert den Pfad zur CAD/CAM Datei.

Kommando KRC2

```
PointLoaderLoad(FilePath[])
```

Kommando KRC4

```
PointLoaderLoad(FilePath[],View)
```

Werte von View: #Fullscreen,#Halfscreen,#None

#FullScreen → HMI wird automatisch als ganzes Fenster geöffnet

#HalfScreen → HMI wird automatisch als halbes Fenster geöffnet

#None → HMI wird nicht automatisch geöffnet

7.4 KRL-Modul "PointloaderMain"

Dieses Modul ist das eigentliche Arbeitsprogram. Es wird kontinuierlich vom Plugin mit Daten gefüllt.

7.5 KRL-Modul "PointloaderUser"

Dieses Modul beinhaltet verschiedene Routinen zu einigen Kommandos der CAD / CAM Datei. Sie werden aus dem Modul "PointloaderMain" aufgerufen. Die Routinen können / müssen vom Anwender entsprechend der Peripherie und der Anwendung angepasst werden.

Enthaltene Routinen

Routinen	Beschreibung
SpindleVel()	Wird durch das Kommando "Spindle_Vel" aufgerufen
SpindleStart()	Wird durch das Kommando "Spindle_Start" aufgerufen
SpindleStop()	Wird durch das Kommando "Spindle_Stop" aufgerufen
ToolChange()	Wird durch das Kommando "Toolchange" aufgerufen
Cooling()	Wird durch das Kommando "Cooling" aufgerufen
Vacuum()	Wird durch das Kommando "Vacuum" aufgerufen
UserPrg()	Wird durch das Kommando "UserPrg" aufgerufen
UserPrgTr()	Wird durch das Kommando "UserPrgTr" aufgerufen

7.6 CAD / CAM Datei

Die komplette CAD / CAM Datei wird vom Plugin sequentiell eingelesen. Die ermittelten Kommandos und Parameter wie Bewegungsart, Punktkoordinaten, Geschwindigkeit etc. werden dem KRL-Modul "PointloaderMain" über zwei Arrays kontinuierlich zur Verfügung gestellt.

Die CAD / CAM Datei muss eine festgelegte Struktur aufweisen und kann neben den Bewegungen verschiedene Kommandos beinhalten

7.6.1 Dateiname

Der Dateiname ist beliebig wählbar. Die Dateierweiterung muss *.src oder *.txt sein. Auf die KRL-spezifischen Angaben im Header kann verzichtet werden.

7.6.2 Speicherort

Die Datei kann sich direkt auf dem Roboter oder auf einem Netzlaufwerk befinden.

7.6.3 Verfügbare Kommandos

In der CAD / CAM Datei können verschiedene Kommandos angegeben werden. Diese werden vom Plugin interpretiert, gespeichert und während des Programmablaufs kontinuierlich in das Arbeitsprogramm geschrieben. Nicht unterstützte Kommandos und Zeilen werden nicht eingelesen und stehen somit dem Arbeitsprogramm nicht zur Verfügung.

Alle nicht eingelesenen Zeilen werden in der Datei „D:\KRC\Roboter\Log\PLSkippedLines.txt“ gespeichert.

Die Anzahl nicht gespeicherter Kommandos wird in der KRL Variable „PLSkippedLines“ gespeichert.

7.6.3.1 Kommandos für Bewegungen

Es stehen folgende Kommandos zur Verfügung:

- PTP \$POS_ACT
- PTP xHOME
- PTP {X ...,Y,C ...,E1.....E6} C_DIS
- PTP {A1 ...,A2,A6 ...,E1.....E6} C_DIS
- LIN {X ...,Y,C ...,E1.....E6} C_DIS
- CIRC {X ...,Y,C ...,E1.....E6},{X ...,Y,C ...,E1.....E6} C_DIS
- SPTP {X ...,Y,C ...,E1.....E6} C_DIS
- SLIN {X ...,Y,C ...,E1.....E6} C_DIS
- SCIRC {X ...,Y,C ...,E1.....E6},{X ...,Y,C ...,E1.....E6} C_DIS

Die Angabe der externen Achsen und des Überschleifparameters C_DIS ist optional

Erläuterung der Kommandos

Element	Beschreibung
PTP \$POS_ACT	ruft die Routine MovePTPPosAct im Module PointLoaderMain.src auf
PTP xHome	ruft die Routine MoveH im Module PointLoaderMain.src auf
PTP	Die Koordinaten werden in kartesischer Form angegeben: Ruft die Routine MoveP im Modul PointLoaderMain.src auf Die Koordinaten werden in achsspezifischer Form angegeben: Ruft die Routine MoveA im Modul PointLoaderMain.src auf Status und Turn können als dezimale oder binäre Zahlen angegeben werden
LIN	Ruft die Routine MoveL im Modul PointLoaderMain.src auf
CIRC	Ruft die Routine MoveC im Modul PointLoaderMain.src auf
SPTP	Ruft die Routine MoveSP im Modul PointLoaderMain.src auf
SLIN	Ruft die Routine MoveSL im Modul PointLoaderMain.src auf
SCIRC	Ruft die Routine MoveSC im Modul PointLoaderMain.src auf

7.6.3.2 Kommandos für Bewegungsparameter

7.6.3.2.1 \$VEL.CP

Beschreibung Bahngeschwindigkeit wird gesetzt.

Die Angabe der Geschwindigkeit kann dabei in direkter Form oder indirekter Form unter Nutzung vordefinierter Geschwindigkeiten erfolgen

Syntax \$VEL.CP = Wert

Erläuterung der Syntax

Element	Beschreibung	Vorlaufstopp
\$VEL.CP	Ruft die Routine VelCP im Modul Pointloader.src auf	nein
Wert	Einheit: m/s Direkte Angabe: Real Wert, Indirekte Angabe:	

	-1 : für die Geschwindigkeit wird der Wert der KRL-Variablen pl_RapidFeed verwendet -2 : für die Geschwindigkeit wird der Wert der KRL-Variablen pl_ContactFeed verwendet -2 : für die Geschwindigkeit wird der Wert der KRL-Variablen pl_WorkFeed verwendet	
--	--	--

Beispiel 1

```
$Vel.CP=0.125
```

Beispiel 2

```
$Vel.CP=-2
```

7.6.3.2.2 \$ACC.CP

Beschreibung Bahnbeschleunigung wird gesetzt

Syntax \$ACC.CP = *Wert*

Erläuterung der Syntax

Element	Beschreibung	Vorlaufstopp
\$ACC.CP	Ruft die Routine AccCP im Modul Pointloader.src auf	nein
Wert	Real Wert, Einheit m/s ²	

Beispiel

```
$ACC.CP=0.125
```


7.6.3.2.3 \$VEL_AXIS

Beschreibung Geschwindigkeit der angegebenen Achse wird gesetzt.

Syntax \$VEL_AXIS[Nummer] = Wert

Erläuterung der Syntax

Element	Beschreibung	Vorlaufstopp
\$VEL_AXIS	Ruft die Routine VelAxis im Modul PointLoaderMain.src auf	ja
No	Nummer der Achse	
Wert	Einheit: % Real Wert, Wertebereich 0-100%	

Beispiel 1

```
$Vel_Axis[1]=80
```

7.6.3.2.4 \$ACC_AXIS

Beschreibung Beschleunigung der angegebenen Achse wird gesetzt.

Syntax \$ACC_AXIS[Nummer] = Wert

Erläuterung der Syntax

Element	Beschreibung	Vorlaufstopp
\$ACC_AXIS	Ruft die Routine AccAxis im Modul PointLoaderMain.src auf	ja
No	Nummer der Achse	
Wert	Einheit: % Real Wert, Wertebereich 0-100%	

Beispiel 1

```
$Acc_Axis[1]=100
```

7.6.3.2.5 \$VEL_EXTAX

Beschreibung Geschwindigkeit der angegebenen externen Achse wird gesetzt.

Syntax \$VEL_EXTAX[Nummer] = Wert

Erläuterung der Syntax

Element	Beschreibung	Vorlaufstopp
\$VEL_EXTAX	Ruft die Routine VelExtAxis im Modul PointLoaderMain.src auf	ja
No	Number of the external axis (1-6)	
Wert	Einheit: % Real Wert, Wertebereich 0-100%	

Beispiel 1

```
$Vel_EXTAX[1]=80
```

7.6.3.2.6 \$ACC_EXTAX

Beschreibung Beschleunigung der angegebenen externen Achse wird gesetzt.

Syntax \$ACC_EXTAX[Nummer] = Wert

Erläuterung der Syntax

Element	Beschreibung	Vorlaufstopp
\$ACC_EXTAX	Ruft die Routine AccExtAxis im Modul PointLoaderMain.src auf	ja
No	Nummer der Achse (1-6)	
Wert	Einheit: % Real Wert, Wertebereich 0-100%	

Beispiel 1

```
$Acc_ExtAx[1]=100
```

7.6.3.2.7 \$Vel.Ori1 and \$Vel.Ori2

Beschreibung Orientierungsgeschwindigkeit von Bahn-Bewegungen wird gesetzt.

Syntax \$VEL.ORI1 = Wert , \$VEL.ORI2 = Wert

Erläuterung der Syntax

Element	Beschreibung	Vorlaufstopp
\$VEL.ORI1 \$VEL.ORI2	Ruft die Routine VelOri im Modul PointLoaderMain.src auf	ja
Wert	Einheit: Grad/s Real Wert	

Beispiel 1

```
$Vel.Ori1=200
```

Beispiel 2

```
$Vel.Ori2=200
```

7.6.3.2.8 \$Acc.Ori1 and \$Acc.Ori2

Beschreibung Orientierungsbeschleunigung von Bahn-Bewegungen wird gesetzt.

Syntax \$ACC.ORI1 = Wert , \$ACC.ORI2 = Wert

Erläuterung der Syntax

Element	Beschreibung	Vorlaufstopp
\$ACC.ORI1 \$ACC.ORI2	Ruft die Routine VelOri im Modul PointLoaderMain.src auf	ja
Wert	Einheit: Grad/s Real Wert	

Beispiel 1

```
$Acc.Ori1=200
```

Beispiel 2

```
$Acc.Ori2=200
```

7.6.3.2.9 \$Advance

Beschreibung Bestimmt die Anzahl der vom Vorlaufzeiger eingelesen Kommandos.

Syntax \$Advance = *Wert*

Erläuterung der Syntax

Element	Beschreibung	Vorlaufstopp
\$Advance	Ruft die Routine Advance im Modul PointLoaderMain.src auf	no
Wert	Integer Wert Wertebereich: 0-5	

Beispiel 1

```
$Advance=3
```

7.6.3.2.10 \$IPO_Mode

Beschreibung Modus der Interpolation wird gesetzt.

Syntax \$IPO_MODE = *Wert*

Erläuterung der Syntax

Element	Beschreibung	Vorlaufstopp
\$IPO_MODE	Ruft die Routine SetIpoMode im Modul PointLoaderMain.src auf	Ja
Wert	#BASE or #TCP	

Beispiel 1

```
$IPO_MODE=#BASE
```

7.6.3.2.11 \$APO.CDIS

Beschreibung Überschleifdistanz wird gesetzt.

Syntax \$APO.CDIS = *Wert*

Erläuterung der Syntax

Element	Beschreibung	Vorlaufstopp
\$APO.CDIS	Ruft die Routine ApoCdis im Modul PointLoaderMain.src auf	Ja
Wert	real Wert [mm]	

Beispiel 1

```
$APO.CDIS=10
```

7.6.3.2.12 \$ORI_TYPE

Beschreibung Orientierungsführung des TCP wird gesetzt

Syntax \$ORI_TYPE = *Wert*

Erläuterung der Syntax

Element	Beschreibung	Vorlaufstopp
\$ORI_TYPE	Ruft die Routine SetOriType im Modul PointLoaderMain.src auf	Ja
Wert	ENUM #VAR, #Constant, #Joint, #Ignore	

Beispiel 1

```
$Ori_Type=#VAR
```

7.6.3.2.13 \$OV_PRO

Beschreibung Programm Geschwindigkeit wird gesetzt.

Syntax \$OV_PRO = Wert

Erläuterung der Syntax

Element	Beschreibung	Vorlaufstopp
\$OV_PRO	Ruft die Routine OV_PRO im Modul PointLoaderMain.src auf	nein
Wert	Integer Wert Wertebereich: 0-100	

Beispiel 1

```
$OV_PRO=50
```

7.6.3.2.14 BAS

Beschreibung Funktionalitäten des Moduls BAS.SRC

Syntax BAS (COMMAND,REAL_PAR)

Erläuterung der Syntax

Element	Beschreibung	Vorlaufstopp
BAS	Ruft die Routine SetBas im Modul PointLoaderMain.src auf	Ja
COMMAND	BAS_COMMAND	
REAL_PAR	REAL	

Verfügbare BAS Kommandos:

```
#INITMOV, #ACC_CP, #ACC_PTP, #VEL_CP, #VEL_PTP, #ACC_GLUE, #TOOL, #BASE, #EX_BASE, #PTP_DAT, #CP_DAT, #OUT_SYNC, #OUT_ASYNC, #GROUP, #FRAMES, #PTP_PARAMS, #CP_PARAMS
```

Beispiel 1

```
BAS (#INITMOV, 0)
```

Beispiel 2

```
BAS (#VEL_CP, 0.05)
```

7.6.3.3 Kommandos für Tool und Base**7.6.3.3.1 ToolBase**

Beschreibung Tool und Base werden auf die angegebenen Werte gesetzt.

Syntax ToolBase(*Tool*, *Base*)

Erläuterung der Syntax

Element	Beschreibung	Vorlaufstopp
ToolBase	Ruft die Routine ToolBase im Modul PointloaderUser.src auf	Ja
Tool	Integer Wert	
Base	Integer Wert	

Beispiel

```
ToolBase(1, 2)
```

7.6.3.3.2 Tool**7.6.3.3.2.1 Option 1: \$Tool=Tool_Data[No]**

Beschreibung \$Tool wird auf den Wert von Tool_Data[No] gesetzt.

Syntax \$Tool=Tool_Data[*No*]

Erläuterung der Syntax

Element	Beschreibung	Vorlaufstop
\$Tool=Tool_Data[<i>No</i>]	Calls the routine Tool in the module PointLoaderMain.src	Ja
No	Integer Value	

Beispiel

```
$Tool=Tool_Data[1]
```

7.6.3.3.2.2 Option 2: \$Tool={X ...,Y ..., Z ..., A ..., B ..., C ...}

Beschreibung \$Tool wird auf die angegebenen Werte gesetzt.

Syntax \$Tool={X ...,Y ..., Z ..., A ..., B ..., C ...}

Beispiel

```
$Tool={X 100,Y 200,Z 500, A 90, B 0, C 0}
```

7.6.3.3.3 Base

7.6.3.3.3.1 Option 1: \$Base=Base_Data[No]

Beschreibung \$Base wird auf den Wert von Base_Data[No] gesetzt.

Syntax \$Base=Base_Data[No]

Erläuterung der Syntax

Element	Beschreibung	Vorlaufstop
\$Base=Base_Data[No]	Calls the routine Tool in the module PointLoaderMain.src	Ja
No	Integer Value	

Beispiel

```
$Base=Base_Data[1]
```

7.6.3.3.3.2 Option 2: \$Base={X ...,Y ..., Z ..., A ..., B ..., C ...}

Beschreibung \$Base wird auf die angegebenen Werte gesetzt.

Syntax \$Base={X ...,Y ..., Z ..., A ..., B ..., C ...}

Beispiel

```
$Base={X 200,Y 2000,Z 1000, A 00, B 0, C 0}
```


7.6.3.3.3 Externe Kinematic: \$Base=EK(Machine_Def[Mach_Idx].Root.....)

Vier verschiedene Kommandos werden unterstützt:

- \$BASE=EK(MACHINE_DEF[MACH_IDX].ROOT,MACHINE_DEF[MACH_IDX].MECH_TYPE,Frame1:BASE_DATA[BASE_NO]:Frame2)
- \$BASE=EK(MACHINE_DEF[MACH_IDX].ROOT,MACHINE_DEF[MACH_IDX].MECH_TYPE,BASE_DATA[BASE_NO]: Frame1)
- \$BASE=EK(MACHINE_DEF[MACH_IDX].ROOT,MACHINE_DEF[MACH_IDX].MECH_TYPE, Frame1:BASE_DATA[BASE_NO])
- \$BASE=EK(MACHINE_DEF[MACH_IDX].ROOT,MACHINE_DEF[MACH_IDX].MECH_TYPE,BASE_DATA[BASE_NO])

Beschreibung \$Base wird durch die Systemfunktion EK(...) gesetzt.

Erläuterung der Syntax

Element	Beschreibung	Vorlaufstop
MACH_IDX	Integer, Index of used Machine_Def	
BASE_NO	Integer, Index of the used Base_Data	
Frame1,Frame	Frame, muss als Struktur angegeben werden ({x ...,y ...,z ...,a ...,b ...,c ...})	

Beispiel

```
$BASE = EK(MACHINE_DEF[2].ROOT,MACHINE_DEF[2].MECH_TYPE,{x 10,y 10,z 10,a 0,b 0,c 0}:BASE_DATA[1]:{x 20,y 30,z 50,a 10,b 0,c 0})
```

7.6.3.4 Kommandos für Ein- und Ausgänge

7.6.3.4.1 Warte auf Eingang

Beschreibung Wartet auf den angegebenen Zustand eines Eingangs

Syntax WAIT FOR \$IN[Bit]==Value

**Anpassung der
aufgerufenen
Routine notwendig** nein

Erläuterung der Syntax

Element	Beschreibung	Vorlaufstop
WAIT FOR \$IN	Calls the routine WaitIn in the module PointLoaderUser.src	No
Bit	Number of input	
Value	0 or 1 TRUE or FALSE	

Beispiel: wait for \$IN[10] is TRUE

```
WAIT FOR $IN[10]==TRUE
```

7.6.3.4.2 SetOutPut

Beschreibung Setzt einen Ausgang auf TRUE oder False

Syntax SetOutPut(Bit,Wert)

**Anpassung der
aufgerufenen
Routine notwendig** nein

Erläuterung der Syntax

Element	Beschreibung	Vorlaufstopp
SetOutPut	Ruft die Routine SetOutPut im Modul PointloaderUser.src auf	Nein
Bit	Nummer des Ausgangsbits	
Wert	0 oder 1 0 → Ausgang wird auf False gesetzt	

	1 → Ausgang wird auf True gesetzt	
--	-----------------------------------	--

Beispiel: Ausgang 10 wird auf True gesetzt

```
SetOutput(10,1)
```

7.6.3.4.3 SetOutPutB

Beschreibung Setzt ein Ausgangsbyte auf den angegebenen Wert

Syntax SetOutPutB(*Startbit*,*Wert*)

Anpassung der aufgerufenen Routine notwendig nein

Erläuterung der Syntax

Element	Beschreibung	Vorlaufstopp
SetOutPutB	Ruft die Routine SetOutPutB im Modul PointloaderUser.src auf	Nein
Startbit	Nummer des ersten Bits des Ausgangsbytes	
Wert	0 bis 255	

Beispiel: Ausgang 28 bis 35 wird auf Wert 100 gesetzt

```
OutputB(28,100)
```

7.6.3.4.4 SetOutPutW

Beschreibung Setzt ein Ausgangswort auf den angegebenen Wert

Syntax SetOutPutW(*Startbit*,*Wert*)

**Anpassung der
aufgerufenen
Routine notwendig** nein

Erläuterung der Syntax

Element	Beschreibung	Vorlaufstopp
SetOutPutW	Ruft die Routine SetOutPutW im Modul PointloaderUser.src auf	Nein
Startbit	Nummer des ersten Bits des Ausgangswortes	
Wert	0 bis 65535	

Beispiel: Ausgang 33 bis 48 wird auf Wert 100 gesetzt

```
OutputW(33,100)
```

7.6.3.4.5 \$ANOUT

Beschreibung Setzt einen analogen Ausgang auf den angegebenen Wert

Syntax \$ANOUT[*Bit*]==(*Value*)

**Anpassung der
aufgerufenen
Routine notwendig** nein

Erläuterung der Syntax

Element	Beschreibung	Vorlaufstopp
\$Anout	Ruft die Routine SetAnout im Modul PointLoaderUser.src	Nein
Bit	Nr des Analogausgangs	
Value	Real Wert	

Beispiel: Analog Ausgang 2 soll auf 18V gesetzt werden

```
$Anout[2]=1.8
```

7.6.3.5 Weitere Kommandos

7.6.3.5.1 Interrupt ON/OFF

Beschreibung Der angegebene Interrupt wird ein- oder ausgeschaltet

Syntax Interrupt ON/OFF *No*

Erläuterung der Syntax

Element	Beschreibung	Vorlaufstop
Interrupt	Ruft die Routine InterruptOnOff im Modul PointLoaderMain.src auf	Ja
No	Nummer des Interrupt	

Beispiel 1

```
Interrupt On 5
```

Die Deklaration des betreffenden Interrupts muss in einem übergeordneten KRL-Modul vor dem Aufruf des Kommandos erfolgt sein. Eine Deklaration in der CAD/CAM Datei ist nicht zulässig.

7.6.3.5.2 HALT

Beschreibung Hält die Programmabarbeitung an

Syntax HALT

Erläuterung der Syntax

Element	Beschreibung	Vorlaufstop
HALT	Hält die Programmabarbeitung an	Ja

Beispiel 1

```
HALT
```

7.6.3.6 Anwenderkommandos

Die Anwenderkommandos rufen korrespondierende Funktionen im KRL-Modul „PointLoaderUser.src“ auf. Diese Funktionen können/müssen vom Anwender mit Inhalt entsprechend seiner Anforderungen gefüllt werden.

7.6.3.6.1 SpindleVel

Beschreibung Ruft die Routine SpindleVel im Modul PointloaderUser.src auf. Die Routine ist nur als Rumpfprogramm programmiert und muss vom Anwender angepasst werden.

Syntax Spindle_Vel(*Vel*, <*Check*>)

**Anpassung der
aufgerufenen
Routine notwendig** ja

Erläuterung der Syntax

Element	Beschreibung	Vorlaufstopp
Spindle_Vel	Ruft die Routine SpindleVel im Modul PointloaderUser.src auf	Nein
Vel	Geschwindigkeit, Integer Wert	
Check	0 oder 1	

Beispiel1: Geschwindigkeit=6000, Check=0

```
Spindle_Vel(6000)
```

Beispiel2: Geschwindigkeit=1000, Check=1

```
Spindle_Vel(1000,0)
```

Beispiel3: Geschwindigkeit=-2000, Check=1

```
Spindle_Vel(-2000,1)
```

7.6.3.6.2 SpindleStart

Beschreibung Ruft die Routine SpindleStart im Modul PointloaderUser.src auf. Die Routine ist nur als Rumpfprogramm programmiert und muss vom Anwender angepasst werden.

Syntax Spindle_Start(<Check>)

Anpassung der aufgerufenen Routine notwendig ja

Erläuterung der Syntax

Element	Beschreibung	Vorlaufstopp
Spindle_Start	Ruft die Routine SpindleStart im Modul PointloaderUser.src auf	Nein
Check	0 oder 1	

Beispiel1: Check=0

```
Spindle_Start()
```

Beispiel2: Check=1

```
Spindle_Start(1)
```

7.6.3.6.3 SpindleStop

Beschreibung Ruft die Routine SpindleStop im Modul PointloaderUser.src auf. Die Routine ist nur als Rumpfprogramm programmiert und muss vom Anwender angepasst werden.

Syntax Spindle_Stop(<Check>)

Anpassung der aufgerufenen Routine notwendig ja

Erläuterung der Syntax

Element	Beschreibung	Vorlaufstopp
Spindle_Stop	Ruft die Routine SpindleStop im Modul PointloaderUser.src auf	Nein
Check	0 oder 1	

Beispiel1: Check=0

```
Spindle_Stop()
```

Beispiel2: Check=1

```
Spindle_Stop(1)
```

7.6.3.6.4 Cooling

Beschreibung Ruft die Routine Cooling im Modul PointloaderUser.src auf. Die Routine ist nur als Rumpfprogramm programmiert und muss vom Anwender angepasst werden.

Syntax Cooling(*State*,<*Check*>)

Anpassung der aufgerufenen Routine notwendig ja

Erläuterung der Syntax

Element	Beschreibung	Vorlaufstopp
Cooling	Ruft die Routine Cooling im Modul PointloaderUser.src auf	ja
State	0 oder 1	
Check	0 oder 1	

Beispiel1: State=1, Check=0

```
Cooling(1,0)
```

Beispiel2: State=0, Check=1

```
Cooling(0,1)
```


7.6.3.6.5 Vaccum

Beschreibung Ruft die Routine Vaccum im Modul PointloaderUser.src auf. Die Routine ist nur als Rumpfprogramm programmiert und muss vom Anwender angepasst werden.

Syntax `Vaccum(State,<Check>)`

Anpassung der aufgerufenen Routine notwendig ja

Erläuterung der Syntax

Element	Beschreibung	Vorlaufstopp
Vacuum	Ruft die Routine Vacuum im Modul PointloaderUser.src auf	ja
State	0 oder 1	
Check	0 oder 1	

Beispiel1: State=1, Check=0

```
Vaccum(1,0)
```

Beispiel2: State=0, Check=1

```
Vaccum(0,1)
```

7.6.3.6.6 ToolChange

Beschreibung Ruft die Routine ToolChange im Modul PointloaderUser.src auf. Die Routine ist nur als Rumpfprogramm programmiert und muss vom Anwender angepasst werden.

Syntax `ToolChange(NewTool,OldTool)`

Anpassung der aufgerufenen Routine notwendig ja

Erläuterung der Syntax

Element	Beschreibung	Vorlaufstopp
ToolChange	Ruft die Routine ToolChange im Modul PointloaderUser.src auf	ja
NewTool	Nummer des neuen Werkzeugs	
OldTool	Nummer des alten Werkzeugs	

Beispiel1: Neues Werkzeug=1, Altes Werkzeug=2

```
ToolChange (1, 2)
```

7.6.3.6.7 UserPrg and UserPrgTr

Diese beiden Kommandos rufen entsprechende KRL-Funktionen auf und können somit zur individuellen Programmierung verwendet werden. Die Funktionen können bis zu sieben Parameter vom Typ INT und REAL verarbeiten.

Syntax `UserPrg(Par1,Par2,Par3,Par4,Par5,Par6,Par7)`

Oder

`UserPrg(Par1,Par2,Par3,Par4,Par5,Par6,Par7)`

**Anpassung der
aufgerufenen
Routine notwendig** Ja

Erläuterung der Syntax

Element	Beschreibung
UserPrg	Ruft die Routine UserPrg im Modul PointloaderUser.src auf
UserPrgTr	Ruft über einen Bahnschaltbefehl die Routine UserPrgTr im Modul PointloaderUser.src auf

Parameter

Element	Beschreibung
Par1	Integer
Par2...Par7	Real

Par1 wird in aufgerufenen KRL-Funktion als Parameter für eine Switch-Anweisung verwendet

Par2-Par7 können als Parameter für Variablen und Funktionen verwendet werden.

7.6.3.6.7.1 UserPrg

KRL Code:

```
GLOBAL DEF
UserPrg (PAR1:IN, PAR2:IN, PAR3:IN, PAR4:IN, PAR5:IN, PAR6:IN, PAR7:IN)
;*****
;Function: user commands
;Params: PAR1, PAR2, PAR3, PAR4, PAR5, PAR6, PAR7
;*****
DECL INT PAR1 ;function identifier
DECL REAL PAR2, PAR3, PAR4, PAR5, PAR6, PAR7 ;parameters for function
calls

DECL INT PAR20, PAR30, PAR40, PAR50, PAR60, PAR70 ;parameters for
function calls
;your code here
SWITCH PAR1 ;function identifier
CASE 10
CASE 20
CASE 30
CASE 40
CASE 50
DEFAULT
WAIT FOR FALSE
ENDSWITCH
END
```

Beispiel 1 CAD/CAM File: Die Variable "UserVariable_1" soll auf den Wert 10.8 gesetzt werden

CAD/CAM:

```
UserPrg(10,10.8,0,0,0,0,0)
```

➔ Anpassung der KRL-Funktion "UserPrg":

```
GLOBAL DEF
UserPrg (PAR1:IN, PAR2:IN, PAR3:IN, PAR4:IN, PAR5:IN, PAR6:IN, PAR7:IN)
;*****
;Function: user commands
;Params: PAR1, PAR2, PAR3, PAR4, PAR5, PAR6, PAR7
;*****
DECL INT PAR1
DECL REAL PAR2, PAR3, PAR4, PAR5, PAR6, PAR7

SWITCH PAR1
CASE 10
    UserVariable_1=Par2
CASE 20
    ...
CASE...
DEFAULT
ENDSWITCH
END
```

Beispiel 2 CAD/CAM File: Die Routine MyProg soll aufgerufen werden. Der Routine sollen drei Parameter mit den Werten 1.5, 600 und TRUE übergeben werden

CAD/CAM:

```
UserPrg(20,1.5,600,1,0,0,0)
```

→ **Anpassung der KRL-Funktion UserPrg:**

```
GLOBAL DEF
UserPrg(PAR1:IN,PAR2:IN,PAR3:IN,PAR4:IN,PAR5:IN,PAR6:IN,PAR7:IN)
;*****
;Function: user commands
;Params: PAR1,PAR2,PAR3,PAR4,PAR5,PAR6,PAR7
;*****
DECL INT PAR1
DECL REAL PAR2,PAR3,PAR4,PAR5,PAR6,PAR7

SWITCH PAR1
CASE 10
    UserVariable_1=Par2
CASE 20
    MyProg(Par2,Par3,Par4)
CASE...
DEFAULT
ENDSWITCH
HALT
END
```



Die Variable UserVariable_1 und die Funktion MyProg müssen auf dem System existieren. Anderseits tritt ein Kompilierungsfehler auf.

7.6.3.6.7.2 UserPrgTr

Die Funktionalität ist gleich zur Funktion "UserPrg"

Einziger Unterschied: Der Funktionsaufruf erfolgt mittels einem Banhschaltbefehl (Trigger).

KRL-Code

```
GLOBAL DEF
UserPrgTr (PAR1:IN, PAR2:IN, PAR3:IN, PAR4:IN, PAR5:IN, PAR6:IN, PAR7:IN)
;*****
;Function: user commands called by Trigger
;Params: PAR1, PAR2, PAR3, PAR4, PAR5, PAR6, PAR7
;*****
DECL INT PAR1 ;function identifier
DECL REAL PAR2, PAR3, PAR4, PAR5, PAR6, PAR7 ;parameters for function
calls

DECL INT PAR20, PAR30, PAR40, PAR50, PAR60, PAR70 ;parameters for
function calls
;your code here
SWITCH PAR1 ;function identifier
CASE 10
CASE 20
CASE 30
CASE 40
CASE 50
DEFAULT
WAIT FOR FALSE
ENDSWITCH
END
```

8 Bedienung

8.1 Start eines Bearbeitungsprogramms

Das Anwenderprogramm "RunPointloader" startet die Programmabarbeitung.

In diesem Programm werden weitere Unterprogramme zum Laden und Abarbeiten der CAD/CAM Datei aufgerufen. Welche CAD/CAM geladen wird, wird bestimmt durch die Variable „FilePath[]“. Diese Variable beinhaltet den Pfad und den Namen der CAD/CAM Datei. Die Zuweisung kann entweder direkt über ein KRL-Kommando oder über einen Datei-Auswahl-Dialog erfolgen.

Der Datei-Auswahl-Dialog kann über die HMI geöffnet werden.

Der Status des Ladens und der Abarbeitung wird in der HMI angezeigt.

**Kommando zum
Laden der CAD/CAM
Datei:** PointloaderLoad(*FilePath[]*)

**Kommando zum
Aufruf des
Abarbeitungs-
programms:** PointloaderMain(*Tool,Base*)

Erläuterung der Syntax:

Parameter	Datentyp	Erlaubte Werte	Beschreibung
FilePath[]	CHAR	512 Zeichen (KRC 4) 255 Zeichen (KRC 2)	Vollständiger Pfad zur CAD / CAM Datei
Tool	INT	>= 0	Nummer des für die Bewegungsausführung zu verwendenden Werkzeugs-Koordinatensystems
Base	INT	>= 0	Nummer des für die Bewegungsausführung zu verwendenden Basis-Koordinatensystems



Die Angabe von Tool und Base ist nur notwendig falls in der CAD/CAM Datei keine Tool- und Basedaten gesetzt werden.

Beispiel 1:

Die CAD / CAM Datei "MyPosFile.src" befindet sich auf Laufwerk D: im Ordner "PosFiles". Da in der CAD/CAM Datei keine Tool- und Basedaten angegeben werden, sollen die Daten von Tool 10 und Base 2 verwendet werden.

```
PointloaderLoad("D:\PosFiles\MyPosFile.src")
...
PointloaderMain(10,2)
```

Beispiel 2:

Die CAD / CAM Datei "MyPosFile.src" befindet sich auf einem Netzlaufwerk "Net" im Ordner "PosFiles". Die zu verwendeten Tool- und Basedaten werden in der CAD/CAM Datei angegeben.

```
PointloaderLoad("\\Net\PosFiles\MyPosFile.src")
...
PointLoaderMain(0,0)
```

Beispiel 3:

Die CAD / CAM Datei "MyPosFile.src" befindet sich auf einem Netzlaufwerk "Net" im Ordner "PosFiles". Die zu verwendeten Tool- und Basedaten werden in der CAD/CAM Datei angegeben. Die Auswahl der CAD/CAM Datei erfolgt über den Datei-Auswahl-Dialog in der HMI

```
PointloaderLoad(FilePath[])
...
PointLoaderMain(0,0)
```

9 HMI

Die HMI dient zur Darstellung verschiedener Roboterparameter. Zusätzlich wird der Fortschritt des aktuellen Bearbeitungsprogramms angezeigt.

In der HMI kann mittels einem Datei-Auswahl-Dialog die zu öffnende CAD/CAM Datei ausgewählt werden.

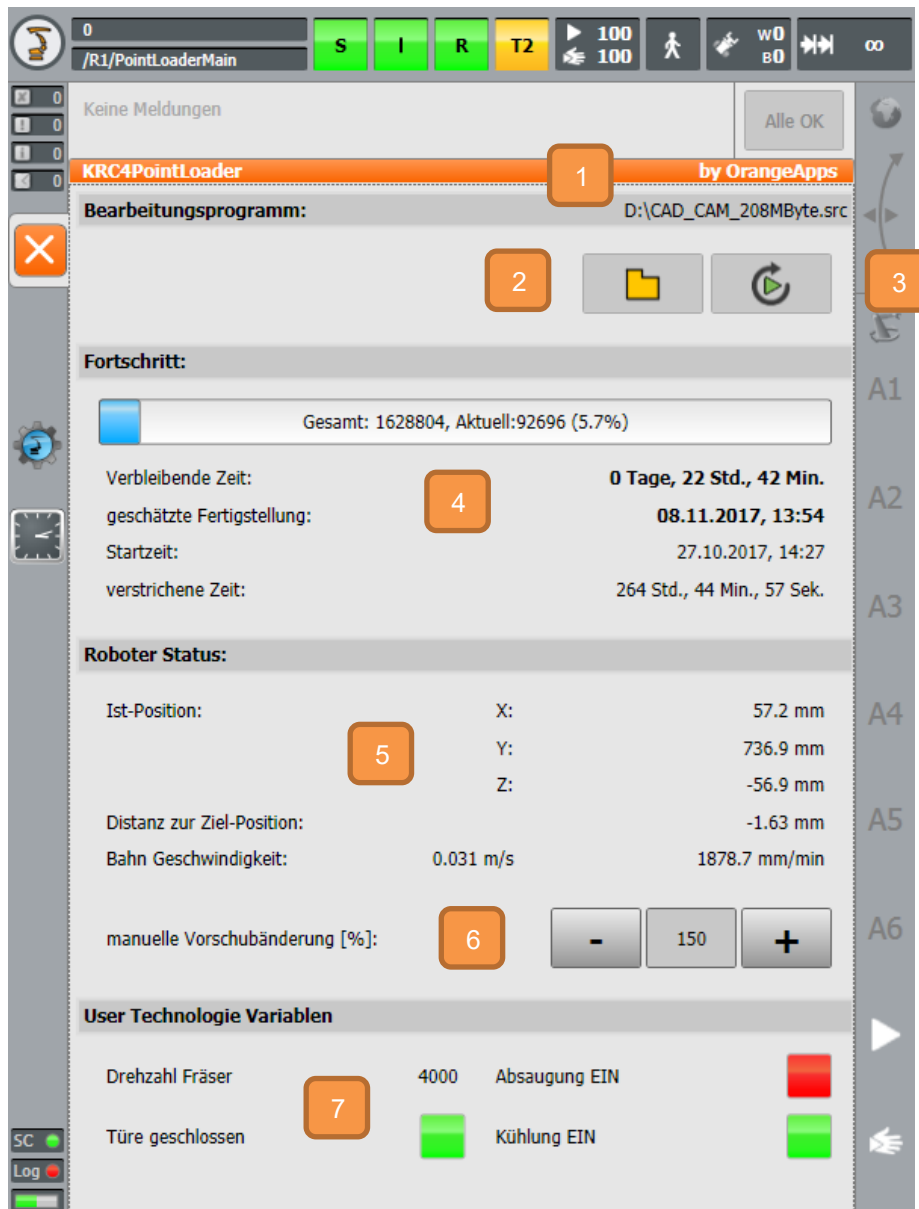
9.1 HMI KRC4

Die HMI kann aus dem Hauptmenü unter dem Eintrag "PointLoader" als halb- oder ganzseitiges Fenster geöffnet werden oder automatisch durch den Aufruf zum Laden der CAD/CAM Datei

Menüeintrag

Datei	▶	HMI - halbseitig
Konfiguration	▶	HMI - ganzseitig
Anzeige	▶	
Diagnose	▶	
Inbetriebnahme	▶	
Herunterfahren		
Hilfe	▶	
KRC4PointLoader	▶	

Darstellung



1	Name und Pfad zur aktuellen CAD/CAM Datei
2	Schaltfläche zum Öffnen des Datei-Öffnen-Dialog. Nur aktiv falls kein Programm läuft.
3	Schaltfläche um aktuelles Programm zurück zu setzen. Nur aktiv falls kein Programm läuft.
4	Der Fortschrittsbalken zeigt die Anzahl an Kommandos der CAD/CAM Datei und die Nummer des aktuellen Kommandos. Während des Ladens einer CAD/CAM Datei wird der Ladefortschritt angezeigt. Weitere Anzeigen: - Startzeit - Verstrichene Zeit seit dem Start

	Verbleibende Zeit und daraus resultierende Zeit der Fertigstellung
5	Anzeige von: - Aktuelle Ist Position des TCP - Distanz zum nächsten Zielpunkt - Bahngeschwindigkeit des TCP in m/s und mm/min (bei PTP Punkten wird die Bahngeschwindigkeit 0)
6	Die aktuell eingestellte Vorschubgeschwindigkeit kann mit der Plus-Minus Taste im Bereich 50....150% verändert werden. Die Übernahme des Wertes erfolgt nach Erreichen des nächsten Punktes.
7	Wert der Variablen HMIUserVarBool1, HMIUserVarBool2, HMIUserVarBool3, HMIUserVarReal1 Bei boolschen Variablen: rot = die Variable hat den Wert FALSE grün = Die Variable hat den Wert TRUE Die Beschriftung der Variablen kann in der Datei PointLoaderUser.kxr im Verzeichnis C:\KRC\DATA angepasst werden. Nach einer Anpassung ist ein Neustart des Systems erforderlich.

9.1.1 Dateiauswahl-Dialog

Nach dem Klick auf das Ordnersymbol öffnet sich der Dateiauswahl-Dialog. Dort kann die zu ladende CAD/CAM-Datei ausgewählt werden. Über die KRL-Variable „DefaultPath“ im Modul „PointLoaderData.dat“ kann ein Pfad voreingestellt werden. Im Auslieferungszustand ist Laufwerk D: voreingestellt. Um keinen Pfad voreinzustellen und alle vom Roboter erreichbaren Laufwerke anzuzeigen kann die Zeile

DefaultPath[]="D:\“ entweder auskommentiert werden oder auf

DefaultPath[]="" gesetzt werden. Beachten Sie dabei das Leerzeichen zwischen den Anführungszeichen. Andernfalls kommt es zu einem Kompilierungsfehler.

9.1.2 User Technologie Variablen

Die vier Technologievariablen können in jedem beliebigen KRL-Modul verwendet werden und deren Werte werden auf der HMI dargestellt.

Beschreibung

Element	Typ
HMIUserVarReal1	Real
HMIUserVarBool1	bool
HMIUserVarBool2	bool

HMIUserVarBool3	bool
-----------------	------

Die Variablen sind definiert in C:\KRC\Roboter\R1\TP\PointLoader\PointLoaderUser.dat

9.1.2.1 Beschriftung der Variablen in der HMI

Die Beschriftung dieser Variablen in der HMI ist mehrsprachig. Beim Ändern der Sprache der HMI wird der Text der Variablen in die tatsächliche Sprache des HMI übersetzt, vorausgesetzt dass ein entsprechender Schlüssel in der Sprachdatenbank gefunden wird. Eine Beschriftung kann in der Sprachdatenbank "C:\KRC\DATA\PointLoaderUser.kxr" geändert oder hinzugefügt werden.

Voreingestellte Sprachen sind Englisch und Deutsch.

Changes in this file take effect after a restart of the robot.

Entry in C:\KRC\DATA\PointLoaderUser.kxr

```
<uiText key="strHMIUserVarReal1">
  <text xml:lang="de-DE">HMIUserVarReal1</text>
  <text xml:lang="en-US">HMIUserVarReal1</text>
</uiText>
<uiText key="strHMIUserVarBool1">
  <text xml:lang="de-DE">HMIUserVarBool1</text>
  <text xml:lang="en-US">HMIUserVarBool1</text>
</uiText>
<uiText key="strHMIUserVarBool2">
  <text xml:lang="de-DE">HMIUserVarBool2</text>
  <text xml:lang="en-US">HMIUserVarBool2</text>
</uiText>
<uiText key="strHMIUserVarBool3">
  <text xml:lang="de-DE">HMIUserVarBool3</text>
  <text xml:lang="en-US">HMIUserVarBool3</text>
</uiText>
```

de-DE: deutsche Übersetzung

en-US: englische Übersetzung

Um weitere Sprachen anzuzeigen müssen entsprechend der folgenden Länderzeichentabelle weitere Einträge hinzugefügt werden.

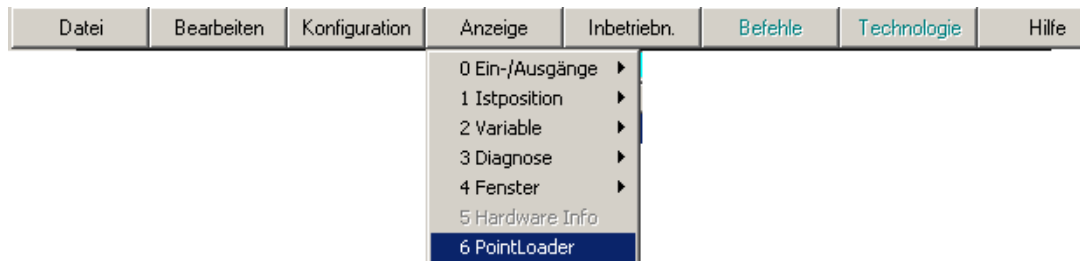
Länderzeichen

Element	Sprache	Element	Sprache
cs	Tschechisch	sk	Slowakisch
because	Dänisch	sl	Slowenisch
de	Deutsch	fi	Finnisch
en	Englisch	tr	Türkisch
it	Spanisch	el	Griechisch
fr	Französisch	ru	Russisch
it	Italienisch	ko	Koreanisch
ugh	Ungarisch	sk	chinesisch
nl	Holländisch	sl	japanisch
pl	Polnisch	fi	Türkisch
pt	Portugiesisch	sv	Griechisch
ro	Rumänisch	tr	Türkisch

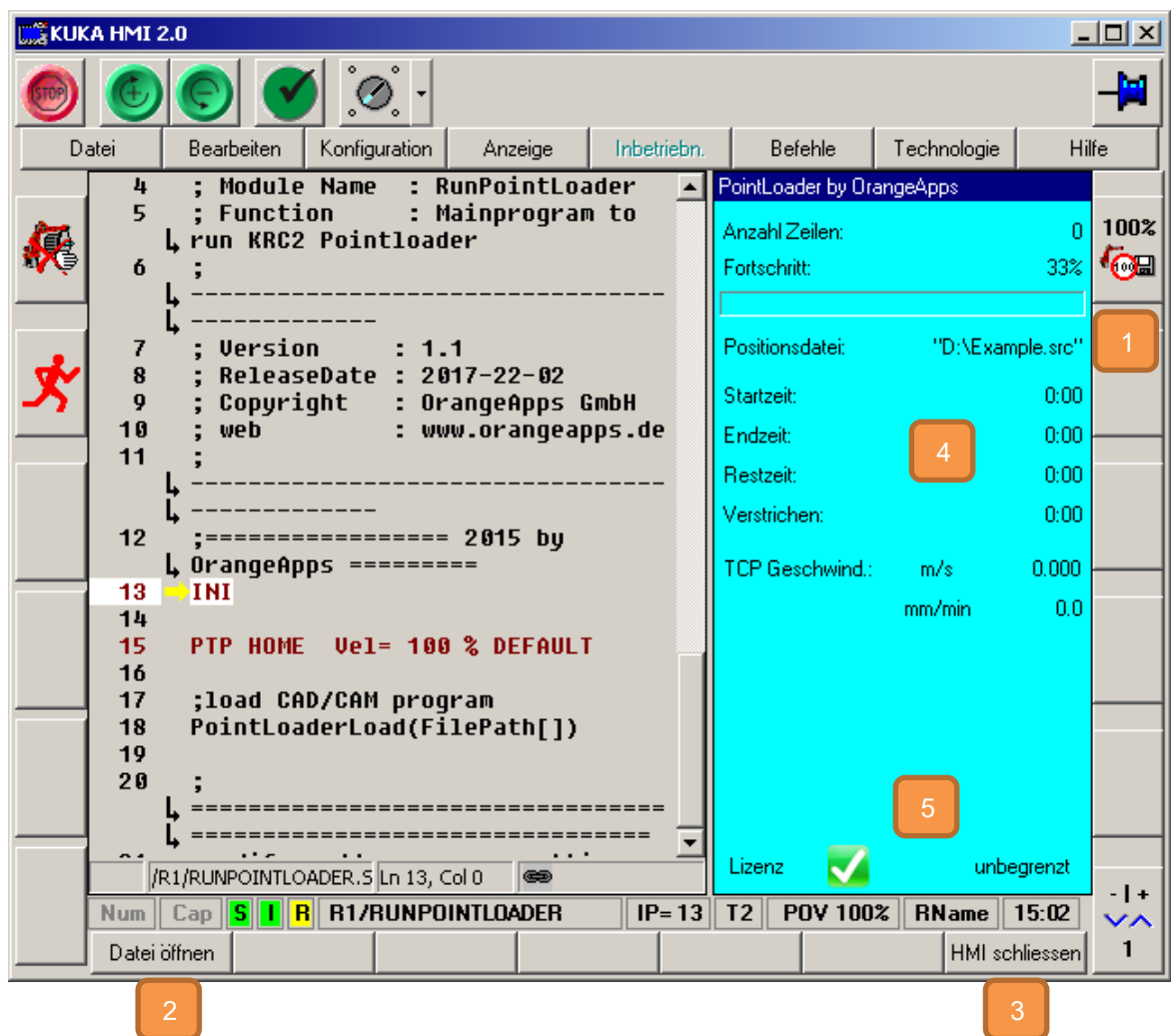
9.2 HMI KRC2

Die HMI kann aus dem Hauptmenü im Menüpunkt „Anzeige“ unter dem Eintrag "PointLoader“ geöffnet werden.

Menüeintrag



Darstellung



1	Name und Pfad zur aktuellen CAD/CAM Datei
2	Schaltfläche zum Öffnen des Datei-Öffnen-Dialog. Nur aktiv falls kein Programm ausgewählt oder Programm zurückgesetzt. Nach Auswahl einer Datei wird automatisch das Benutzerprogramm „RunPointLoader“ ausgewählt.
3	Schaltfläche um HMI zu schließen.
4	Weitere Anzeigen: ▪ Fortschrittsbalken ▪ Startzeit ▪ Verstrichene Zeit seit dem Start ▪ Verbleibende Zeit und daraus resultierende Zeit der Fertigstellung ▪ Bahngeschwindigkeit des TCP in m/s und mm/min (bei PTP Punkten wird die Bahngeschwindigkeit 0)
5	Information zur Lizenz

10 Log-Funktion

Nach jedem Laden eines CAD/CAM Files werden die Ergebnisse dieses Ladevorgangs in zwei Textdateien im Ordner C:\KRC\Roboter\Log geloggt.

Die Datei "PLSkippedLines.txt" dokumentiert alle Zeilen für die in PointLoader kein Kommando zur Verfügung steht.

Über die KRL-Variable "PLSkippedLines" kann die Anzahl dieser Zeilen abgefragt werden.

Die Datei "PILastReadLine.txt" zeigt die zuletzt eingelesene Zeile der CAD/CAM Datei.

11 Beispiele für Kommandos in der CAD/CAM Datei

```

DEF Example()
$VEL.CP=-1 ;if $vel.CP=-1 , -2, or -3 then $vel.cp is set to the
value of the KRL-variable pl_RapidFeed, pl_ContactFeed, pl_WorkFeed
$VEL.CP=0.002
$ACC.CP=2.0 ;calls the routine AccCP and set $Acc.CP to 2.0
$VEL_AXIS[1]=20 ;$Vel_AXIS[Number]=Value
$ACC_AXIS[1]=100 ;$ACC_AXIS[Number]=Value
$VEL_EXTAX[1]=20 ;$Vel_EXTAX[Number]=Value
$ACC_EXTAX[1]=100 ;$ACC_EXTAX[Number]=Value
;SET LIN AND ARC MOTION VARIABLES
$VEL.ORI1=200
$VEL.ORI2=200
$ACC.ORI1=100
$ACC.ORI2=100
$IPO_Mode=#Base ;#TCP, calls the routine SetIpoMode and sets
$IPO_Mode
$OV_PRO=50 ;sets program override
HALT ;stops working program
$ORI_TYPE=#VAR ;Orientierungsuehrung
Spindle_Stop(1) ;calls the routine spindle_stop
ToolChange(1,2) ;calls the routine ToolChange
ToolBase(1,2) ;calls the routine ToolBase and sets the $tool to
tool_data[1] and $base to base_data[2]
Spindle_Vel(4000,1) ;calls the routine Spindle_Vel
Cooling(1,0) ;calls the routine Cooling
Vacuum(1,1) ;calls the routine Vacuum
Spindle_Stop(1) ;calls the routine Spindle_Stop
Spindle_Start(1) ;calls the routine Spindle_Start

;user commands, calls the routines UserPrg and UserPrgTr (UserPrgTr
is called by trigger)
;first parameter is for switch command in the routines
UserPrg(10,1.0,2.0,3.0,4.0,5.0,6.0)
UserPrgTr(10,1.0,2.0,3.0,4.0,5.0,6.0)
;external kinematic : $BASE =
EK(MACHINE_DEF[No].ROOT,MACHINE_DEF[No].MECH_TYPE,BASE_DATA[No])
;Option 1:
$BASE = EK(MACHINE_DEF[2].ROOT,MACHINE_DEF[2].MECH_TYPE,{x 0,y 0,z
0,a 0,b 0,c 0}:BASE_DATA[1]:{x 0,y 0,z 0,a 0,b 0,c 0})
;Option 2:
$BASE =
EK(MACHINE_DEF[2].ROOT,MACHINE_DEF[2].MECH_TYPE,BASE_DATA[1]:{x 0,y
0,z 0,a 0,b 0,c 0})
;Option 3:
$BASE = EK(MACHINE_DEF[2].ROOT,MACHINE_DEF[2].MECH_TYPE,{x 0,y 0,z
0,a 0,b 0,c 0}:BASE_DATA[1])
;Option 4:
$BASE =
EK(MACHINE_DEF[2].ROOT,MACHINE_DEF[2].MECH_TYPE,BASE_DATA[1])
; Frames must be given as aggregate like {x ...,y ...,z ...,a ...,b
...,c ...}!!! No variables allowed.

$BASE=BASE_DATA[15] ;calls the routine Base and sets the $base to
base_data[15]
$TOOL=TOOL_DATA[15] ;calls the routine Tool and sets the $tool to
tool_data[15]
$TOOL={X 10, Y 20, Z 30, A 0, B -90, c 0} ;sets $tool to specific
data

```



```

$BASE={X 1000, Y 20, Z 30, A 0, B 0, c 0} ;sets $base to specific
data

$ANOUT[1]=1.8 ;$Anout[Nr]=Value

PTP xHOME
PTP $POS_ACT
PTP {A1 0, A2 -90, A3 90, A4 0, A5 -45, A6 0, E1 0}
PTP {X 1450, Y 110, Z 2000, A 9.952, B 0, c 0, E1 1000, E2 0, E3 0,
E4 0, E5 0, E6 0} C_DIS ;movements with upto 6 external axis
PTP {X 1450, Y 110, Z 2000, A 9.952, B 0, c 0, E1 0, E2 0, S
'B110', T 'B110011'} C_DIS ;PTP with Status and Turn binary
PTP {X 1450, Y 110, Z 2000, A 9.952, B 0, c 0, E1 0, E2 0, S 6, T 51}
;PTP with Status and Turn decimal

$VEL.CP=0.167 ;calls the routine VelCp and sets $vel_cp to 0.167
OutPutB(1,255) ;calls the routine SetOutPutB and sets a byte
beginning at bit 1 to 255
OutPutW(16,65535) ;calls the routine SetOutPutW and sets a word
beginning at bit 16 to 65535
OutPut(40,1) ;calls the routine SetOutPut and sets output 40 to True
;HINT: too much technology commands in a row can cause overflow of
interrupts

INTERRUPT OFF 3 ;calls the routine InterruptOnOff and switches off
the interrupt with given number
INTERRUPT ON 3 ;calls the routine InterruptOnOff and switches on the
interrupt with given number

WAIT FOR $IN[1]==FALSE ;calls the routine WaitIN and waits for the
given value off an input
$OUT[1]=TRUE ;calls the routine SetOutPut and sets the output 1 to
True

LIN {X 1501, Y 110, Z 2000, A 9.952, B 0, c 0} C_DIS
CIRC {X 1507.5, Y 105, Z 2000, A 9.952, B 0, c 0}, {X 1508, Y 108, Z
2000, A 9.952, B 0, c 0} C_DIS
CIRC {X 1508.5, Y 109, Z 2005, A 9.952, B 0, c 0}, {X 1509, Y 111, Z
2005, A 9.952, B 0, c 0} C_DIS
LIN {X 1508.8, Y 110, Z 2000, A 9.952, B 0, c 0}
$OUT[2]=TRUE
$advance=3
WAIT FOR $IN[1]==False
$ACC.CP=2
Spindle_Vel(6000,0)
Spindle_Start(1)

```

12 Meldungen

Folgende Meldungen können bei der Verwendung der Software angezeigt werden.

Lizenz

Meldung	Beschreibung
Dialog-Meldung: Lizenz für das Produkt Pointloader ungültig oder abgelaufen. Kontaktieren sie ihren Systemintegrator.	Die Lizenz-Datei ist ungültig z.B. falsche Seriennummer, oder eine laufzeitbegrenzte Lizenz ist abgelaufen. Eine neue Lizenz-Datei behebt das Problem.
Dialog-Meldung: Keine Lizenz für das Produkt Pointloader vorhanden. Kontaktieren sie ihren Systemintegrator.	Es ist keine Lizenz-Datei auf dem System. Eine neue Lizenz-Datei behebt das Problem.
Status-Meldung: X Tage verbleibend bis Lizenz abläuft	Wird bei laufzeitbegrenzten Lizenzen angezeigt wenn verbleibender Zeitraum < 15 Tage ist.
Status-Meldung: Keine Lizenz-Datei für Roboter X vorhanden	Keine Lizenz-Datei für den Roboter mit der Serien-Nummer X (X=Seriennummer). Eine neue Lizenz-Datei behebt das Problem.
Status-Meldung: Lizenz für Roboter X ungültig oder abgelaufen	Keine gültige Lizenz für den Roboter mit der Serien-Nummer X (X=Seriennummer). Eine neue Lizenz-Datei behebt das Problem.
Info-Meldung: Datum wurde manipuliert, Lizenz wurde zurückgesetzt!	Bei Verwendung einer laufzeitbegrenzten Lizenz wurde erkannt, dass das Datum des Roboter-Systems verändert wurde. Die Lizenz wird ungültig. Eine neue Lizenz-Datei behebt das Problem.

Programmabarbeitung

Meldung	Beschreibung
Info-Meldung: Unbekanntes Dateiformat <i>Dateiname</i>	Das Dateiformat der CAD / CAM Datei ist nicht „src“
Info-Meldung: Keine CAD / CAM Datei ausgewählt	Das Laden einer CAD / CAM Datei wurde ausgelöst, jedoch wurde keine Datei ausgewählt. Die Variable FilePath[] ist leer.
Dialog-Meldung: Datei <i>Dateiname</i> nicht gefunden	Die CAD / CAM Datei <i>Dateiname</i> konnte nicht gefunden werden. Pfadangabe überprüfen.
Info-Meldung: Fehler beim Lesen der CAD / CAM Datei <i>Dateiname</i>	Beim Lesen der CAD / CAM Datei ist ein Fehler aufgetreten. Inhalt der Datei prüfen.
Info-Meldung: Fehler beim Laden der Positionen	Die Positionen konnten nicht in die Arrays geladen werden
Info-Meldung: Ladevorgang Datei <i>Dateiname</i> gestartet.	Das Laden der CAD / CAM Datei <i>Dateiname</i> wurde gestartet.
Info-Meldung: Ladevorgang: ...% gelesen	Fortschrittsanzeige Ladevorgang der CAD / CAM Datei
Info-Meldung: Ladevorgang Datei <i>Dateiname</i> abgeschlossen	Der Ladevorgang der CAD / CAM Datei wurde abgeschlossen
Info-Meldung: Ladevorgang Datei <i>Dateiname</i> abgebrochen	Der Ladevorgang der CAD / CAM Datei wurde z.B. durch Drücken der Stopptaste abgebrochen
Info-Meldung: Fehler beim Berechnen der Zeitanzeige.	Beim Berechnen der Zeitanzeige ist ein Fehler aufgetreten. Screenshot der Zeitanzeige anfertigen, Archiv erstellen, Fehler notieren und an OrangeApps melden.
Info-Meldung: Fehler beim Schreiben der KRL-Daten.	Die vom Plugin eingelesen Kommandos konnten nicht in die Arrays im Modul Pointloader.src geschrieben werden. Eventuell ist das Kommando in der CAD / CAM Datei fehlerhaft.