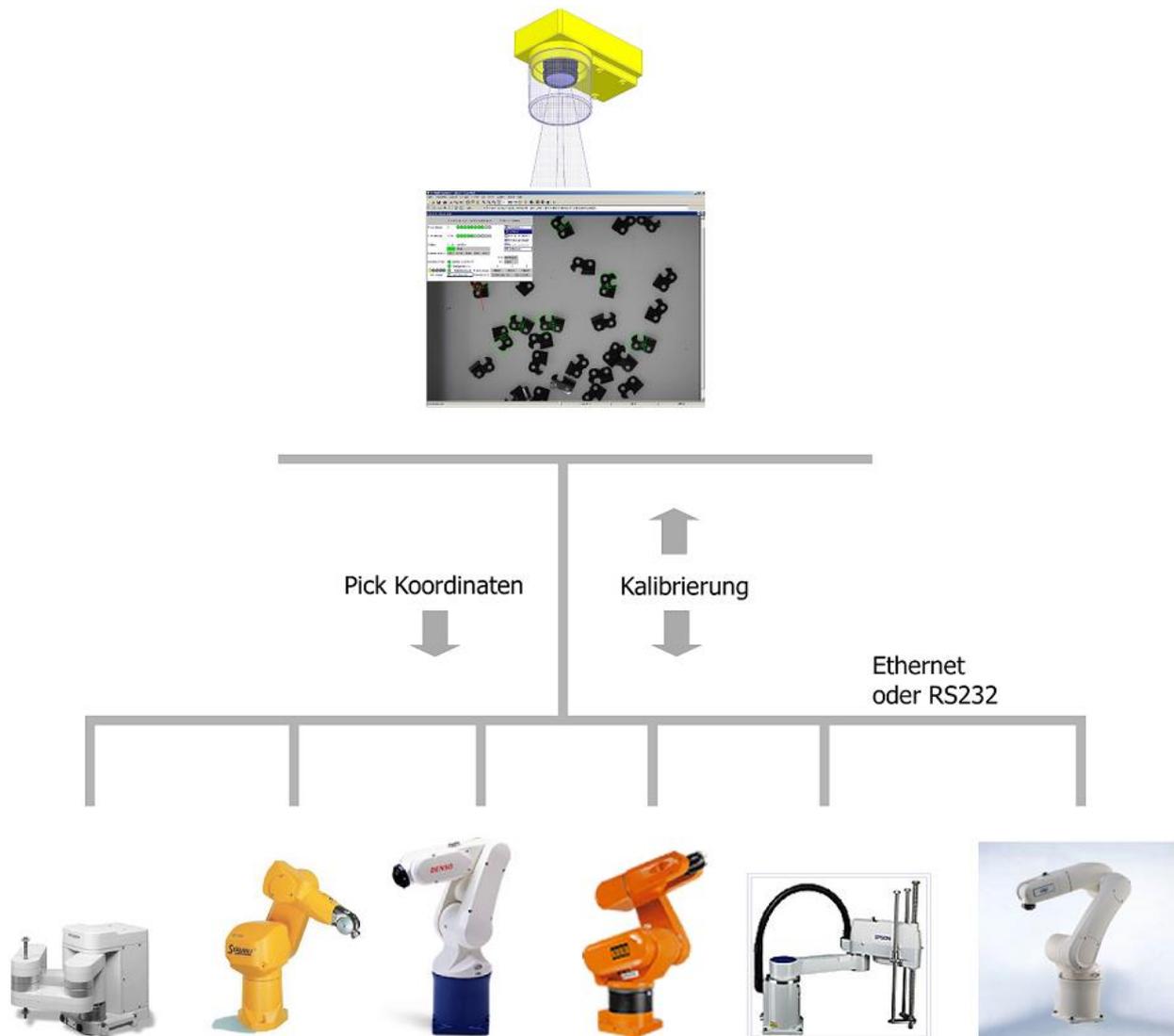


# Datenaustausch zwischen der feedWare CX Kamera-Software und Robotern unterschiedlicher Marken



flexfactory ag • Giessenstrasse 15 • CH-8953 Dietikon • Switzerland  
Tel +41 44 774 55 66 • Fax +41 44 774 55 67 • Email: [info@flexfactory.com](mailto:info@flexfactory.com)

# Cognex Native-Mode Schnittstelle zu flexfactory feedware CX

flexfactory ag , 25.04.2016 (Rev J)

## Allgemeines zur Kommunikation im NativeMode

Die Kommunikation zur Kamera kann entweder über die RS232-Schnittstelle oder via Telnet über Ethernet erfolgen. Bei serieller Kommunikation müssen die Einstellungen beider Seiten übereinstimmen (InSight Menu >Sensor >serielle Anschlusseinstellungen). Bei Telnet-Kommunikation wird die Verbindung über die IP-Adresse und Portnummer (default: 23) hergestellt. Sobald der Port geöffnet wird schickt die Kamera folgende Willkommensmeldung und wartet auf die Eingabe des Benutzernamens (default: admin):

```
Welcome to In-Sight(tm) 5603 Session 0
User:
```

Achtung: Nach dem String „User:“ ist kein Terminator (CR LF) angehängt. Nachdem die Eingabe mit dem Terminator abgeschlossen wird, sendet die Kamera den String: „Password:“ (wieder ohne Terminator) und wartet auf Eingabe des Passworts (default: Keins, also nur den Terminator). War die Anmeldung erfolgreich, sendet die Kamera den String „User Logged In“ (diesmal mit Terminator).

Grundsätzlich antwortet die Kamera auf alle NativeMode-Kommandos mit 1<CR LF> wenn der Befehl verstanden bzw. erfolgreich ausgeführt wurde, oder mit einer negativen Zahl als Fehlercode, wenn ein Problem vorliegt.

Vor dem Absetzen eines Befehls muss also immer auf die positive Antwort des vorhergehenden Befehls gewartet werden. Zu beachten ist auch, dass alle Werte als String ausgetauscht werden. Für Zahlen muss der String also umgewandelt werden.

Im Folgenden sind einige Beispiele der NativeMode-Kommunikation aufgeführt:

### Ganzzahl in eine Zelle schreiben: z.Bsp. SIH0001 (SetInteger in Zelle H000 auf 1)

1. Roboter sendet String 'SIH0001'<CR LF>
2. Kamera antwortet mit '1'<CR LF>, wenn der Wert 1 erfolgreich in Zelle H000 übernommen worden ist.

### Kamera Status prüfen (on-/offline):

1. Roboter sendet String 'GO'<CR LF>
2. Kamera antwortet mit '1'<CR LF> wenn sie online ist oder mit '0'<CR LF> wenn sie offline ist.

### Befehlsfolge um Kamera-Job zu laden:

1. Befehl 'SO0' um Kamera offline zu schalten.
2. Befehl 'LF<filename>'<CR LF> um den gewünschten Job aus dem Memory zu laden. (Dauert ca.3s )
3. Befehl 'SO1' um Kamera wieder online zu schalten.

Achtung: Dies geht nur, wenn die Kamera zuvor schon online war! D.h. vor dem offline schalten den Status prüfen und gegebenenfalls manuell online setzen (Aufforderung ausgeben).

### String in eine Zelle schreiben: z.Bsp. SSE000x=16 (SetString in Zelle E000 auf 'x=16')

1. Roboter sendet String 'SSE000x=16'
2. Kamera antwortet bei Erfolg mit '1'<CR LF>, sonst mit ungleich '1'<CR LF>

### Auslesen einer Zelle: z.Bsp. GVL000 (GetValue von L000)

1. Roboter sendet String 'GVL000'<CR LF>
2. Kamera antwortet z.B. mit '1'<CR LF>'Pick'<CR LF>
  - Der erste Teil der Antwort, also '1'<CR LF> heisst, dass die Kamera den Befehl akzeptiert hat.
  - Der zweite Teil der Meldung enthält die eigentliche Information, hier also 'Pick'<CR LF>

### Auslösen eines Software Events (SE7) auf der Kamera:

1. Roboter sendet String 'SE7'<CR LF>
2. Kamera antwortet bei Erfolg mit '1'<CR LF>

Was im Anschluss an SE7 ausgelöst wird, ist vom Job abhängig. Bei der feedWare CX, wird der String in Zelle E000 an den anyfeed gesendet (mit dem String 'x=16' wird der Feeder initialisiert, 'x=7' bedeutet: Feeder entleeren)

# Roboter-Kamera Kalibrierung

## Vorbereitung

1. Roboterseitig muss ein Programm geschrieben werden, welches mit dem feedWare CX Calibration-Tool kommuniziert und den Ablauf gemäss Vorgabe ausführt.
2. Kamera in korrekter Position fixieren, Blende und Schärfe einstellen.
3. Kalibrierteil in feedWare CX Calibration-Tool einlernen.
4. Dem Roboter 4 Punkte im Bildbereich lernen, an denen das Kalibrierteil abgelegt werden kann. Die Positionen müssen so festgelegt werden, dass die Kamera das Teil immer vollständig sieht.
5. Einrichten einer Aufnahme ('Nest') für das Kalibrierteil, wo es vom Roboter gegriffen werden kann. Falls das Teil nicht im Greifer selbst zentriert wird, muss diese Aufnahme das Kalibrierteil nach jedem Ablegen wieder in x, y und Winkel exakt ausrichten, damit Positionierfehler nicht akkumuliert werden!

**Wichtig:** Die Orientierung des Kalibrierteils muss eindeutig bestimmt werden können. Verwenden Sie keine symmetrischen Geometrien! Hier einige Beispiele von möglichen Formen, die sich als Kalibrierteil eignen:



## Ablauf der Kalibrierung

Der Roboter legt das Teil jeweils auf einem der 4 Kalibrierpunkte in einem vorgegebenen Winkel ab. Um das Drehzentrum zu ermitteln wird das Teil mehrmals unter verschiedenen Winkeln an die gleiche Position abgelegt.

1. Sicherstellen, dass das Kalibrierteil in der Aufnahme liegt.
2. Das File 'Calibration32-P.job' laden.
3. Kamera online schalten.
4. Die Nummer des gewünschten Kalibrierpunktes wählen und 'Start procedure' klicken.
5. Der Roboter liest permanent das Aufgabenfeld L0 aus.
6. Wenn „NoPart“ dann erneut Aufgabenfeld L0 lesen.
7. Wenn „Calib...“ dann zuerst Status auf Eingriff (H0 auf 1) setzen.
  - Kalibrierteil vom Nest (oder später aktueller Kalibrierposition) holen und mit angegebenem Winkel auf den entsprechenden Kalibrierpunkt ablegen. ('Calib1...' für Punkt 1, 'Calib2...' für Punkt 2 usw.)
  - Roboterkoordinaten (x und y) des aktuellen Kalibrierpunktes in Zelle P0 und Q0 schreiben.
  - Arm zurückziehen und sobald er ausserhalb des Bildbereiches ist, Status auf Ausgeführt (H0 auf 0) setzen und wieder Aufgabenfeld L0 auslesen.
8. Wenn „CalibZur“ dann Teil holen und in die Aufnahme zurück legen.

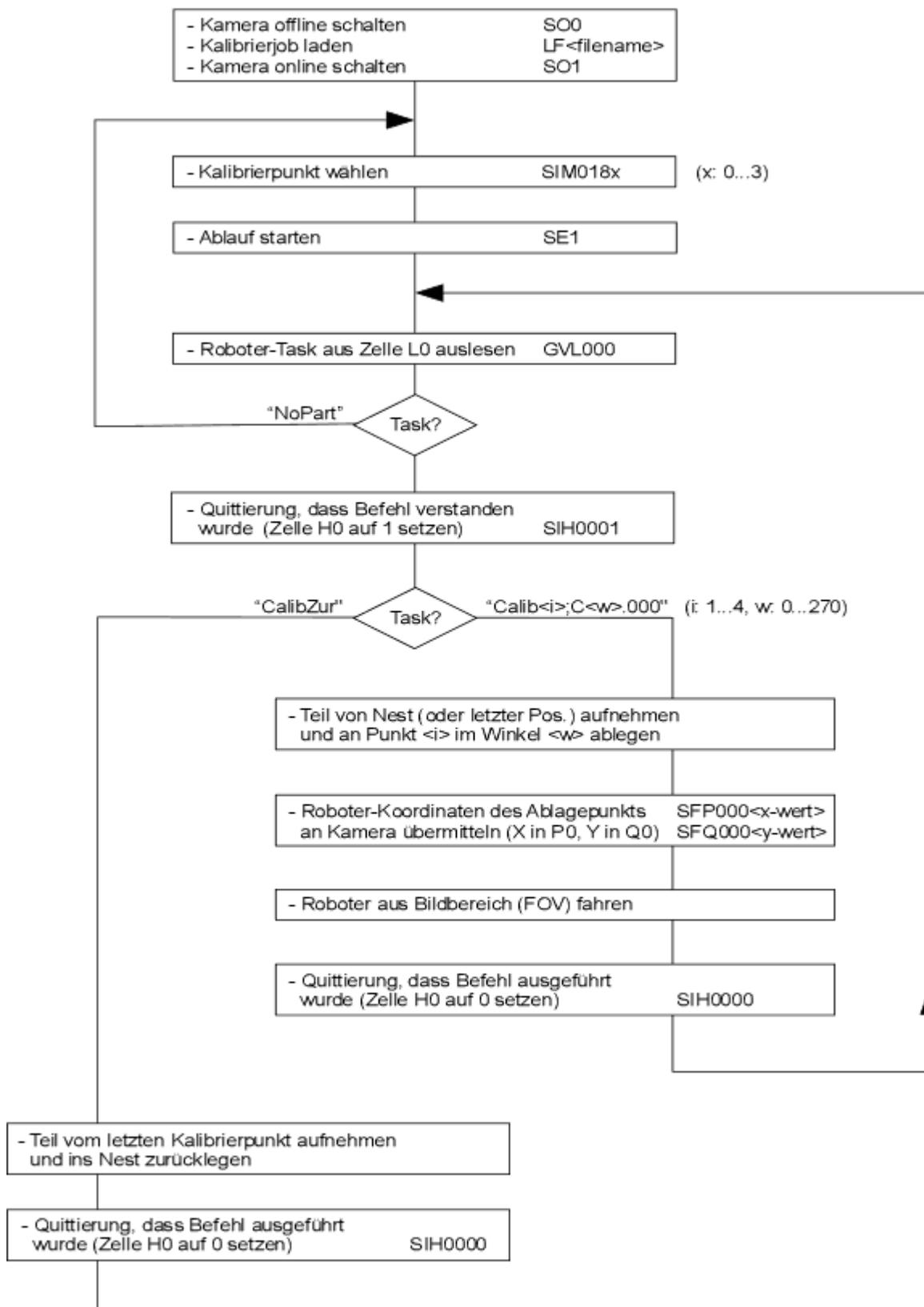
**WICHTIG:** Das Kalibrierteil langsam auf die Positionen absetzen und sicherstellen, dass es nicht verrutscht.

## Beispiele von Set und Get Kommandos:

Was	Wo	Wie	Erklärung
Roboter Aufgabe beim Kalibrieren	Zelle L0	GVL000 (GetValue)	Calib1;C000.000: Teil auf Punkt 1 mit 0° ablegen Calib1;C090.000: Teil auf Punkt 1 mit 90° ablegen CalibZur: Kalibrierteil auf Lagerstelle zurücklegen Calib2;C000.000: Teil auf Punkt 2 mit 0° ablegen (Die 0°-Orientierung ist beliebig) Wenn ein Punkt fertig aufgenommen ist, steht die Aufgabe 'NoPart' an.
Roboter-Koord. an Kamera senden	Zellen P0,Q0	SFP000nn.n SFQ000nn.n (SetFloat)	Wenn der Roboter das Teil abgelegt hat, muss er die Roboterkoordinaten (x und y) in diese beiden Zellen schreiben bevor er den Status in H0 wieder auf 0 (Befehl abgeschlossen) setzt.

Das Diagramm auf der folgenden Seite soll den gesamten Kalibrier-Ablauf verdeutlichen:

## Ablauf-Diagramm für Roboter-Kamera Kalibrierung mit anyfeedWare CX



**Wichtig:** Ist die Kalibrierung abgeschlossen, d.h. sind alle Punkte durchlaufen, müssen die Resultate noch in das spezifizierte File (.xcd) gespeichert werden. (SE2)

## Task-Abfolge für alle 4 Kalibrierpunkte (mit jeweils 4 Winkelpositionen)

### Kalibrierteil nach jedem Schritt zurück ins Nest zur Zentrierung:

Calib1;C000.000  
CalibZur  
Calib1;C090.000  
CalibZur  
Calib1;C180.000  
CalibZur  
Calib1;C270.000  
CalibZur  
NoPart  
  
Calib2;C000.000  
CalibZur  
Calib2;C090.000  
CalibZur  
Calib2;C180.000  
CalibZur  
Calib2;C270.000  
CalibZur  
NoPart  
  
Calib3;C000.000  
CalibZur  
Calib3;C090.000  
CalibZur  
Calib3;C180.000  
CalibZur  
Calib3;C270.000  
CalibZur  
NoPart  
  
Calib4;C000.000  
CalibZur  
Calib4;C090.000  
CalibZur  
Calib4;C180.000  
CalibZur  
Calib4;C270.000  
CalibZur  
NoPart

### Kalibrierteil wird im Greifer zentriert:

Calib1;C000.000  
Calib1;C090.000  
Calib1;C180.000  
Calib1;C270.000  
CalibZur  
NoPart  
  
Calib2;C000.000  
Calib2;C090.000  
Calib2;C180.000  
Calib2;C270.000  
CalibZur  
NoPart  
  
Calib3;C000.000  
Calib3;C090.000  
Calib3;C180.000  
Calib3;C270.000  
CalibZur  
NoPart  
  
Calib4;C000.000  
Calib4;C090.000  
Calib4;C180.000  
Calib4;C270.000  
CalibZur  
NoPart

## Ablauf Pick&Place-Betrieb: Kamera im Streaming Mode

Mit dem Befehl SM"Start"0 wird der Streaming-Mode zwischen Kamera und Robotersteuerung initialisiert. Es spielt keine Rolle, ob Telnet über Ethernet oder RS232 als Kommunikationsart eingesetzt wird.

Einmal initialisiert, sendet die Kamera der Robotersteuerung nach jeder Bildaufnahme die Abgreifkoordinaten im folgenden Format oder 'NoPart' wenn kein Teil zum Abgreifen bereit ist.

**Standard (ein Teil pro Bild abgreifen):**

**'Pick;Xvxxx.xxx;Yvyyy.yyy;Cvccc.ccc;'**

v steht stellvertretend für Vorzeichen und kann die Werte '+' oder '-' annehmen

(Total 35 Zeichen, ohne <CR LF>)

**Multi-Pick aktiviert:**

**'Pick;Ni;Xvxxx.xxx;Yvyyy.yyy;Cvccc.ccc;Xvxxx.xxx;Yvyyy.yyy;Cvccc.ccc;'**

v steht stellvertretend für Vorzeichen und kann die Werte '+', '-' annehmen

i kann die Werte '1' oder '2' haben.

'1' : es liegt 1 Teil zum Abholen bereit, d.h. nur der erste Punkt im obigen Format ist gültig ist. Beim 2. Punkt sind alle drei Werte auf +000.000 gesetzt.

'2' : es liegen 2 Teile zum Abholen bereit, d.h. Beide gesendeten Punkte repräsentieren gültige Abgreifkoordinaten.

***Im Multi-Pick-Betrieb wird immer der ganze String wie oben gezeigt übermittelt, jeder Koordinaten-Wert besteht aus einem Vorzeichen ('+' oder '-'), 3 Vor- und 3 Nachkommastellen, getrennt durch einen '.'***

(Total 68 Zeichen, ohne <CR LF>)

Roboterseitig muss ein Programm geschrieben werden, welches die Daten im Eingangspuffer fortlaufend ausliest, die Koordinaten inkl. Rotation (X, Y, C) aus dem String extrahiert und daraus den Abgreifpunkt für den Roboter zusammenbaut. Im Datenstrom der Kamera werden auch noch andere Information gesendet, die zur Zeit noch nicht dokumentiert sind. Es ist deshalb wichtig, dass das Roboterprogramm nur nach Empfang von 'Pick;...' die nachfolgenden Zeichen als Koordinaten interpretiert und umrechnet.

Wir empfehlen überdies, dass jedesmal geprüft wird, ob der berechnete Abgreifpunkt innerhalb eines erlaubten xy-Bereichs der Applikation liegt.

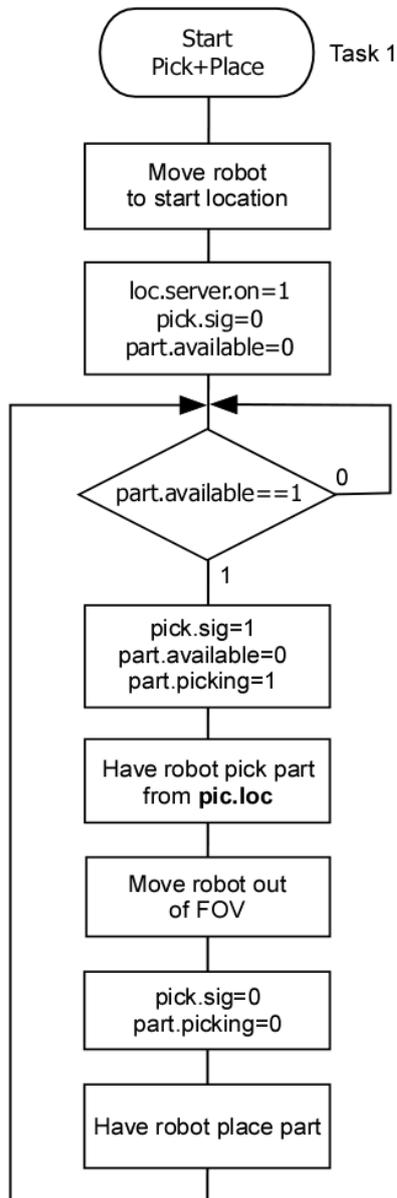
Der Zugriff auf den Feeder muss natürlich nach wie vor koordiniert werden:

- Roboter im Bildbereich der Kamera:  
PICK-Signal zu Feeder einschalten (24V) und die Variable part.picking=1 setzen
- Roboter ausserhalb des Kamerablickfeldes:  
PICK-Signal zu Feeder ausschalten (0V) und die Variable part.picking=0 setzen

Es ist zu beachten, dass beim Input-Buffer der seriellen Schnittstelle kein Overflow auftritt und dabei Zeichen verloren gehen.

Um den Streaming Mode zu stoppen kann der String SM"Stop"0 gesendet werden.

## Top Level Flow Chart for Pick-and-Place Operation with anyfeed SX and anywhere/Insight 5000 (streaming mode)



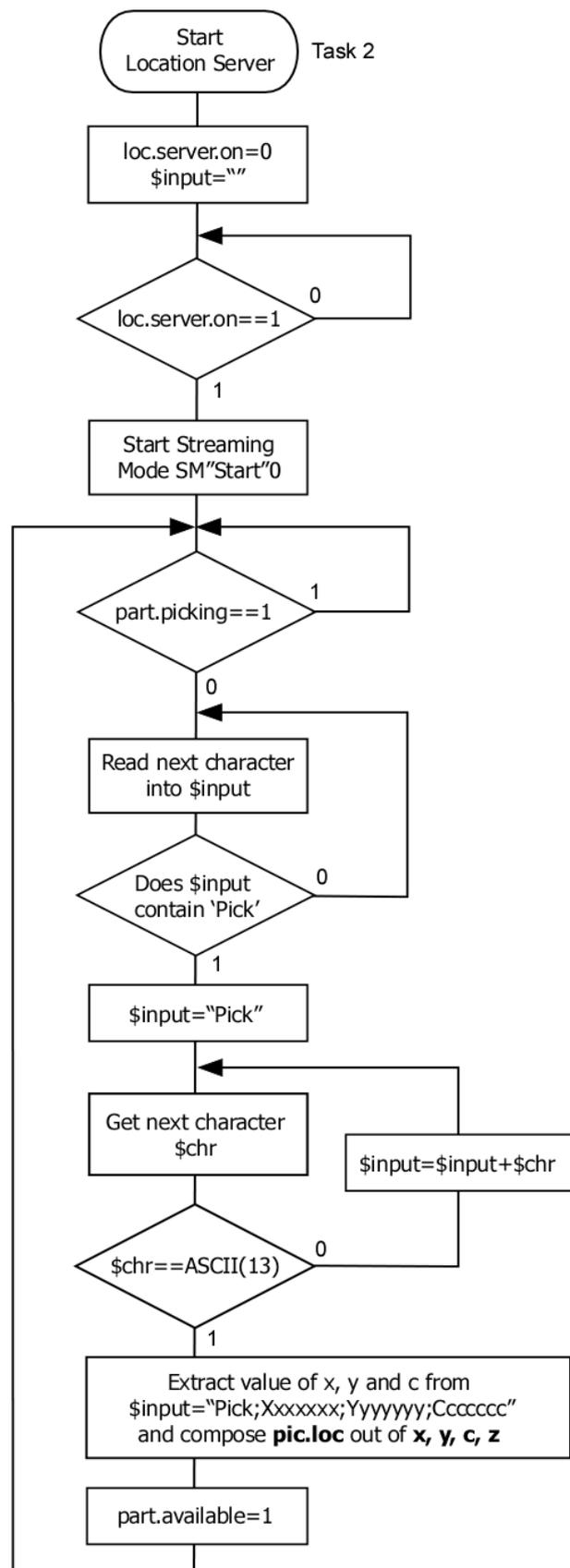
### Note:

Task 1 and Task 2 run in parallel

Shared variables:

loc.server.on starts Location Server (T2)  
 part.available set by T2 when a part is ready to pick  
 reset by T1 right before picking  
 part.picking tells T2 that robot is picking a part  
 pick.sig digital output to stop feeder motion  
 and camera trigger  
 pic.loc contains pick location if part.available==1

fov = field of view



## Abläufe via Native-Mode steuern:

### Job laden und Zuführung starten:

- |                                |              |
|--------------------------------|--------------|
| 1. Kamera offline              | SO0          |
| 2. Job laden                   | LF<filename> |
| 3. Feeder stoppen (präventiv)  | SIT0121      |
| 4. Pick-Signal low (0V) setzen |              |
| 5. Online gehen                | SO1          |
| 6. 4 Sekunden warten           |              |
| 7. Feeder starten              | SIT0120      |
| 8. Streaming-Mode starten      | SM"Start"0   |

Dieser String muss meist aus einzelnen Teilen zusammengesetzt werden: "SM"+Chr(34)+"Start"+Chr(34)+"0"  
Anschließend sendet die Kamera nach jedem Bild aktiv Informationen (siehe Seite 6) über die Schnittstelle.  
(Der komplette String, d.h. Task und Koordinaten, liegt in Zelle L369)

### Feeder entleeren:

- |                                |                  |
|--------------------------------|------------------|
| 1. Feeder stoppen              | SIT0121          |
| 2. Pick-Signal low (0V) setzen |                  |
| 3. Feeder entleeren            | SSE000x=7<br>SE7 |