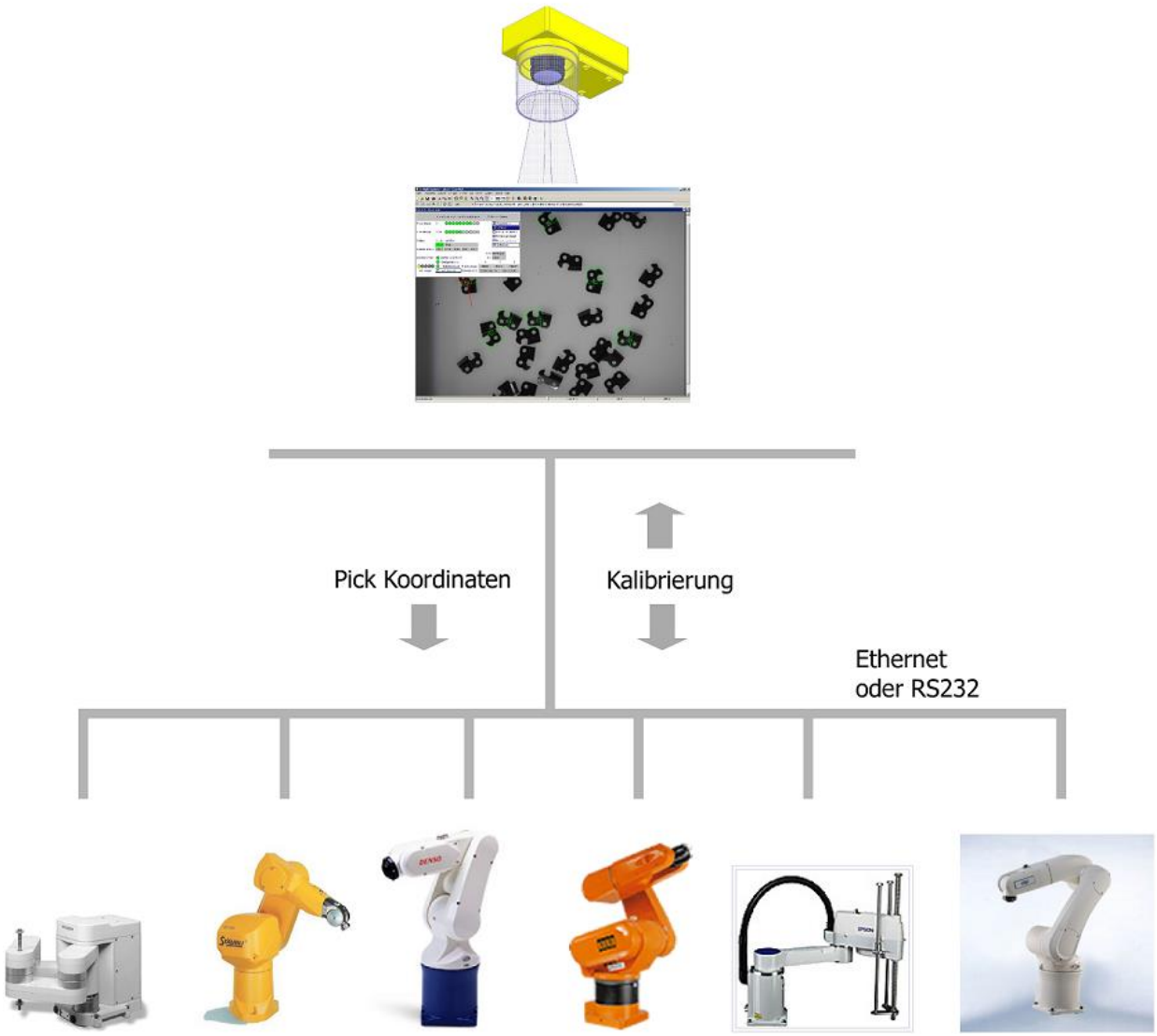
Data exchange
between feedWare CX Camera-Software
and roboters of any brand



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

flexfactory®
reinvent feeding

Cognex Native-Mode interface to flexfactory feedware CX

flexfactory ag , 10.09.2019 (Rev K)

General information about the communication via NativeMode

If you are communicating with the camera via Telnet over Ethernet you need to login before you can send any command. As soon as you open the port 23 (IP: 192.168.2.10) you receive the following string :

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

Now you have to send the username as string (admin) including the delimiter (CR LF). Then you receive:

```
Password:
```

Now send the password (default is an empty string, means only delimiter) and you receive: **`User Logged In`**

Note: The strings „User:“ and „Password:“ are sent without delimiter (no CR LF)!!

Write an integer value into a cell, e.g. SIH0001 (SetInteger in cell H000 to value 1)

1. Robot sends string 'SIH0001'<CR LF>
2. Camera responds with '1'<CR LF>, If the value 1 is successfully stored in the cell H000.
   Camera responds with a number other than '1'<CR LF>, if the command was not successfully terminated.

Check (get) camera status (on-/offline):

1. Robot sends string 'GO'<CR LF>
2. Camera responds with '1'<CR LF> if it is online.
   Camera responds with '0'<CR LF> if it is offline.

Load job:

1. Robot sends string 'SO0' to set the camera offline.
2. Robot sends string 'LF<filename.job> to load the desired job form the cameras internal memory.
3. After receiving „1“ (=loading completed) send string 'SO1' to set the camera online again.

   Note: In order to set the camera offline via native mode it needs to be in online state beforehand! Thus, check the status before switching offline and, if necessary, switch the camera online manually or display a message to do so.

Read a value from a cell, e.g. GVL000 (GetValue from cell L000)

1. Robot sends the string 'GVL000'<CR LF>
2. Camera responds with '1'<CR LF> as acknowledgement and e.g. 'Pick'<CR LF> as cell content.
3. For an accepted command the camera always responds with '1'<CR LF>
4. If the camera doesn't understand a command or there is another issue it sends an error code instead of the '1'. Thus it is strongly recommended that you always check the answers after sending a command to verify that it is successfully processed.
5. The second part of the message contains the requested data.

   Note: The returned data is always in string format. Therefore, if you expect any value information, you need to extract it using the ASCII code.

Write a string into a cell, e.g. SSE000x=16 (SetString in cell E000 to 'x=16')

1. Robot sends string 'SSE000x=16'
2. Camera responds width '1'<CR LF> at successful completion, otherwise with <error code><CR LF>

Trigger Software Events (SE) on the camera e.g. SE7:

1. Robot sends string 'SE7'
2. Camera responds with '1'<CR LF> at successful completion, otherwise with <error code><CR LF>

   The command SE7 sends the content of cell E000 as string to the anyfeed via serial line or Ethernet, depending on the communication mode.

Robot-Camera Calibration

Preparation

1. A robot-program needs to be written that communicates with the feedWare CX calibration tool and that executes the procedure according to the specification in this document.
2. Fix the camera at the correct position and adjust sharpness and aperture of the lens.
3. Teach the calibration part in the feedware CX calibration tool.
4. Teach the robot 4 points in the field of view (FOV) where the calibration part will be placed. Make sure the camera can see the part at all points and possible angles.
5. Set up a nest where the calibration part can be picked up by the robot. In case the part is not centered and aligned automatically within the gripper (e.g. most vacuum grippers) this nest must align the part precisely in x, y and angle after each placement to avoid position errors to accumulate.

Important: The orientation of the calibration part needs to be recognized by the camera unambiguously. Don't use symmetric parts! Here are some examples of geometries suitable as calibration part:

Calibration procedure

The robot places the part on the same calibration point several times in different angels to determine the center of the rotation-axis. This procedure is repeated on all 4 calib-points. Each point is calibrated separately and independent of the others. Thus you can recalibrate a single point if needed.

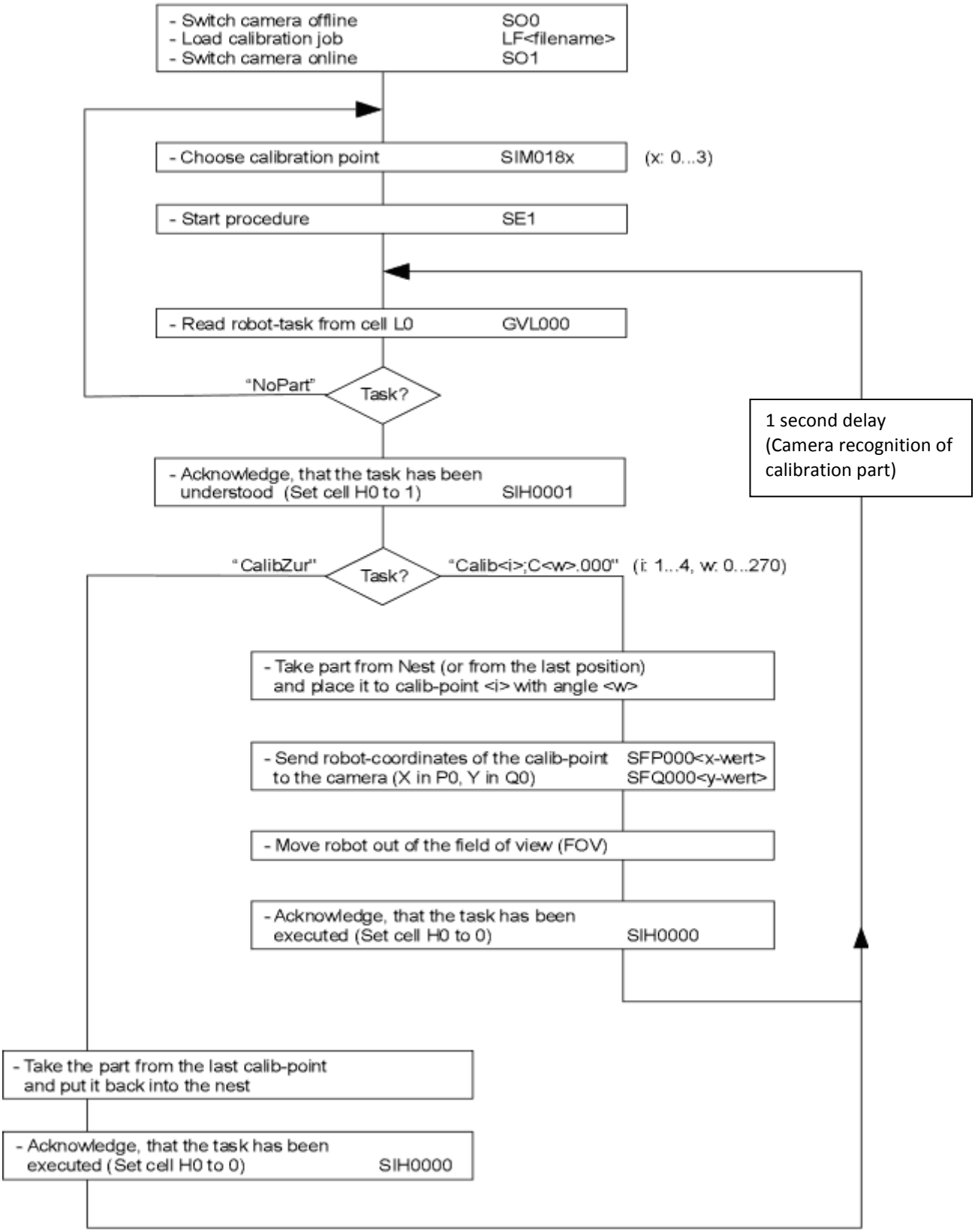
1. Make sure the calibration part sits in the prepared nest.
2. Load the file 'Calibration32-P.job' (manually or via NativeMode. See Flow-chart on next page).
3. Set the camera online.
4. Choose the number of the desired calibration point and click 'Start procedure'.
5. The robot meanwhile reads permanently the task from cell L0.
6. If „NoPart“ the robot continues reading cell L0.
7. If „Calib...“ then first set the status to 'picking' (H0 to value 1). This tells the camera that the task was understood and the robot moves inside the FOV → image acquisition stops.
8. Pick the calib-part from its nest and place it to the given calib-point in the appropriate orientation (angle) („Calib1...“ for Point 1, „Calib2...“ for Point 2 etc.).
9. Write robot coordinates (x and y) of the current calibration point into cells P0 and Q0.
10. Move robot arm outside FOV and then set the status to 'task completed' (set integer in cell H0 to 0)
11. Read again the task in the cell L0.
12. If „CalibZur“ then get part from the last calib-position and put it back into the nest.
13. Repeat procedure starting at step 6.

IMPORTANT: Place the calib-part at calibration points very carefully and check that it doesn't shift while it is released from the gripper! Generally move the robot slowly to prevent from disturbing vibrations.

Examples of Set and Get Commands:

What	Where	How	Description
Robot task while calibration (Camera driven)	Cell L0	GVL000 (GetValue)	Calib1;C000.000: Place at calibration point 1, 0° Calib1;C090.000: Place at calibration point 1, 90° CalibZur: Put calib-part back into nest. Calib2;C000.000: Place at calibration point 2, 0° Note: The 0°-orientation is arbitrary. Only the angle steps are important. After all orientations of one calib-point have been recorded the task will be 'NoPart'
Robot coordinates of the current calibration point	Cells P0,Q0	SFP000nn.n SFQ000nn.n (SetFloat)	After the robot has placed the calib-part, the robot's X- and Y-coordinates of that location must be written into the dedicated cells on camera. IMPORTANT: This has to be done before the status in cell H0 is set to 0 (Task executed).

Flow-chart for robot-to-vision calibration with anyfeedWare CX



Important: After the above procedure has been applied to all 4 calib-points, the gathered data must be stored in the specified file (.cxd) using the Native-Mode command: SE2.

Task-sequence for all 4 calibration points (each with 4 orientation steps)

Calib part is put back into the nest after each step for centering:

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

Calib part is centered within the gripper:

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

Pick&Place-Sequence: Camera in Streaming Mode

Start Streaming-Mode with the command **SM“Start“0** This mode runs whether Telnet or RS232 is used. Streaming-Mode means that the camera actively sends the robot task after each trigger.

To stop the streaming-Mode use the command **SM“Stop“0** .

Note: in most programming languages the apostrophe is used to define a string expression. Therefore it could be necessary to build the complete string by adding strings and characters using the ASCII code for double quote: “SM“ + Chr(34) + “Start“ + Chr(34) + “0“

If there is a part ready to pick the coordinates are sent together with the task in the following format:

'Pick;Xvxxx.xxx;Yvyyy.yyy;Cvccc.ccc;'<CR LF>

- **v** stands representative for the algebraic sign and can therefore be '+' or '-'

  Each coordinate value consists of an algebraic sign ('+' or '-') , 3 digits before the decimal point, the point '.' itself and 3 fractional digits.
  (Totally 35 characters, without <CR LF>

If no part is ready to pick, means the feeder is in the process of preparing new parts, the string **'NoPart'** is sent. Other possible tasks are: 'STOP' and 'ERROR'

In case you are using Multi-Pick (selectable from the Pickpoint settings-Menu) the Pick-string is as follows:

'Pick;Ni;Xvxxx.xxx;Yvyyy.yyy;Cvccc.ccc;Xvxxx.xxx;Yvyyy.yyy;Cvccc.ccc;'<CR LF>

- **v** stands representative for the algebraic sign and can therefore be '+' or '-'
- **i** can be '1' or '2'.

  '1' : There is only one part ready to pick, meaning only the first set of coordinates is a valid point. The values of the second point are all set to +000.000.

  '2' : There are two parts ready to pick, meaning both sets of coordinates are valid pick points.

  Note: With Multi-Pick-active, always the whole string, as shown above, is transmitted.
  (Totally 68 characters, without <CR LF>

The robot program has to read the data flow from the camera continuously and if the task is 'Pick;…', the coordinates X, Y and rotation C need to be extracted from the string and put together in a format the robot understands.
We recommend, that every extracted set of coordinates is checked to be within valid xy-boundaries.

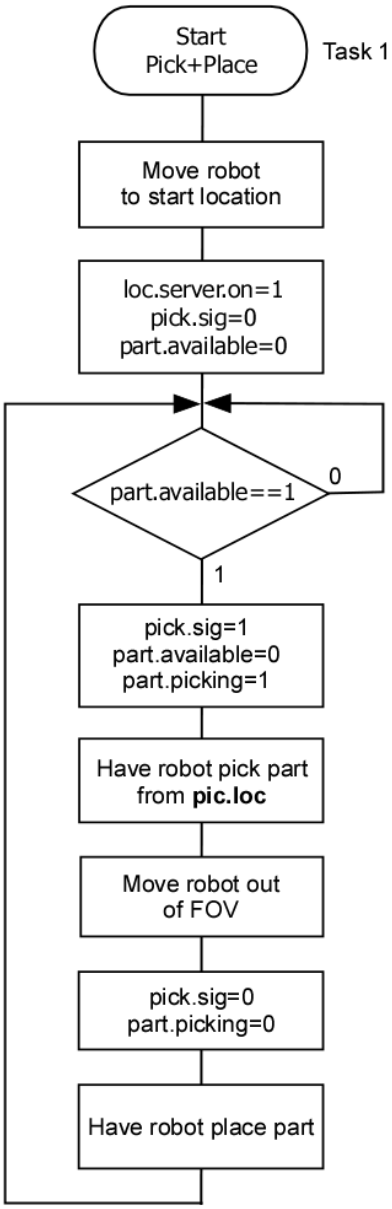
Pick-signal to the anyfeed

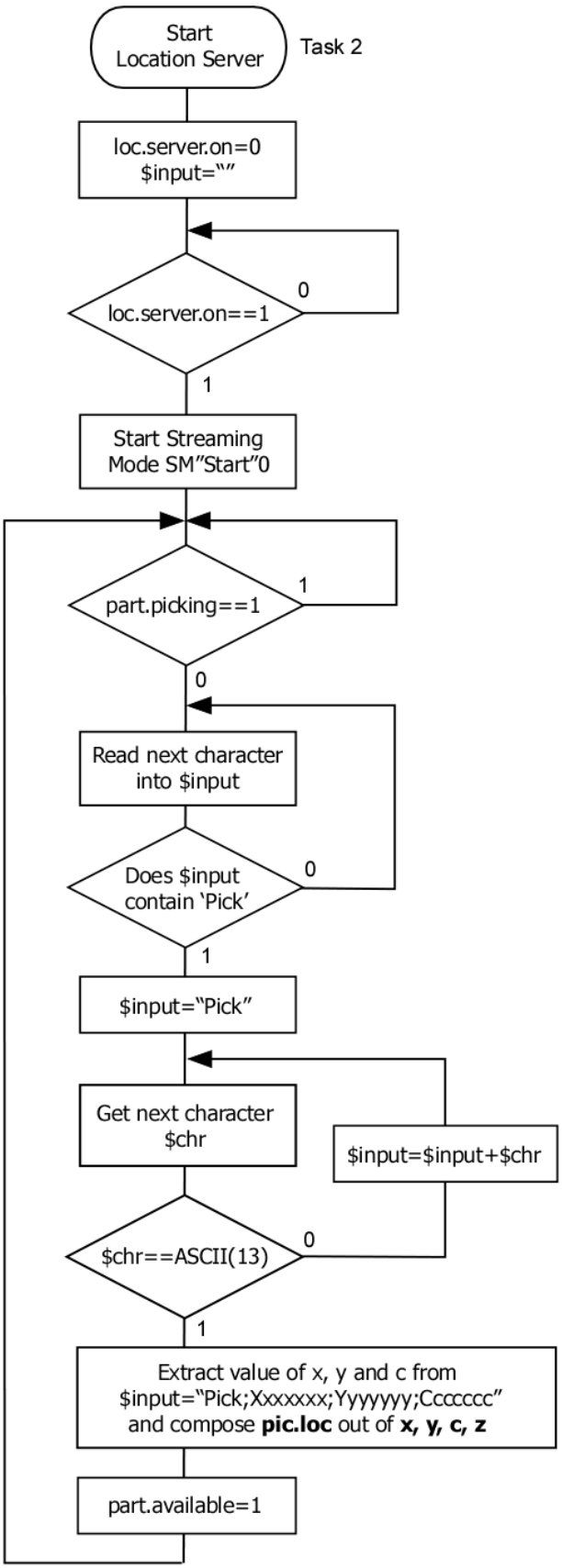
When the robot moves inside the feeding area to pick a part neither a feeder motion nor a camera trigger should be initiated. To prevent this the robot controller must set a digital output (24VDC), called Pick-signal, as soon as it received a pick coordinate. With this signal set, the feeder won't accept any (motion-) commands and stops triggering the camera.
After picking, the Pick-signal should be reset to 0V, as soon as the robot moves outside the FOV. At this moment a camera-trigger is initiated to take the next image.

- Robot receives a camera coordinate:
  → instantly set PICK-Signal to anyfeed high (24V) and set the flag 'part.picking=1' (see following flowchart)
- Robot is outside the cameras FOV after picking:
  → Set PICK-Signal to anyfeed low (0V) and set the flag 'part.picking=0' (see following flowchart)

Remember that the camera sends strings continuously. Make sure to prevent an input buffer overflow.

Top Level Flow Chart for Pick-nd-Place Operation with anyfeed SX and anyware/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

Procedure via Native-Mode

Info:
Every command must be sent with an CR LF at the end. Accepted commands will be responsed with an '1' after executing. Always wait for the cameras respond '1' before sending a new command.

Communication to the Camera (via Ethernet)

1. Connect the camera		
Telnet connection to camera IP:	192.168.2.10 (Port: 23)	//Default
Wait until response form the camera	Welcome to In-Sight(tm) xxxx Session 0 User:	
2. Send username	**admin**	//Default
Wait until response form the camera	Passwort:	
3. send password	***Empty-String*** → **only CR LF**	//Default
Wait until response form the camera	User Logged In	

Load and start a production/recognition job

1. Set camera offline	**Command:**	**SO0**
Wait until response form the camera		1
2. Load production job	**Command:**	**LF<filename>**
Wait until response form the camera		1
3. Start Feeder automatic	**Command:**	**SIT0121**
Wait until response form the camera		1
4. Set pick-signal to low (0V)		
5. Set camera online	**Command:**	**SO1**
Wait until response form the camera		1
6. Wait 2 seconds (Initializing in progress…)		
7. Request State 1	**Command:**	**GVM017**
Wait until 1st response form the camera		1
Wait until 2nd response form the camera		1.000
8. Request State 2	**Command:**	**GVM016**
Wait until 1st response form the camera		1
Wait until 2nd response form the camera		1.000
9. Request State 3	**Command:**	**GVM018**
Wait until 1st response form the camera		1
Wait until 2nd response form the camera		1.000
10. Start Feeder automatic	**Command:**	**SIT0120**
Wait until response form the camera		1
11. Start streaming mode	**Command:**	**SM"Start"0**
Wait until response form the camera		1

Stop Feeder:

1. Stop streaming mode	**Command:**	**SM"Stop"0**
Wait until response form the camera		1
2. Stop Feeder automatic	**Command:**	**SIT0121**
Wait until response form the camera		1
3. Set pick signal to low (0V)		

Empty/purge Feeder:

1. Stop streaming mode	**Command:**	**SM"Stop"0**
Wait until response form the camera		1
2. Stop Feeder automatic	**Command:**	**SIT0121**
Wait until response form the camera		1
3. Set pick signal to low (0V)		
4. Empty Feeder (2 commands)	**1. Command:**	**SSE000x=7**
	Response form the camera	1
	2. Command:	**SE7**
	Response form the camera	1

Feeder purging is now running

5. Request state of emptying	**Command:**	**GVM016**
Wait until 1st response form the camera		1
Wait until 2nd response form the camera		1.000*

*if the return value is 0.000, it means the purge cycle is still active. Request state of emptying until the second response line of the camera is “1.000”. This return value equals the “done” state of the purge cycle.

Request states in feedWare job

Feeder state	**GVM016**	
Response 1 von camera	1	
Response 2 von camera	1.000	Instruction completed
	0.000	Processing in command
	-1.000	Feeder Error
Parameter state	**GVM017**	
Response 1 von camera	1	
Response 2 von camera	1.000	Configuration OK
	0.000	Configuration not OK
	-1.000	Configuration not sent
Initialization state	**GVM018**	
Response 1 von camera	1	
Response 2 von camera	1.000	Initialization OK
	0.000	Not Initialized
	-1.000	Feeder not initialized

Vision-Roboter calibration (referring to diagram on page 4)

1. Set camera offline — **Command:** **SO0**
Wait until response form the camera — 1

2. Load calibration job — **Command:** **LF<filename>**
Wait until response form the camera — 1

3. Set camera online — **Command:** **SO1**
Wait until response form the camera — 1

4. Wait 2 seconds (initializing in progress)

5. Request initializing state — **Command:** **GVM023**
Wait until 1st response form the camera — 1
Wait until 2nd response form the camera — 1.000

---

6. Set calibration point — **Command:** **SIM018**(A)
Wait until response form the camera — 1

7. Start calibration procedure — **Command:** **SE1**
Wait until response form the camera — 1

---

8. Read Roboter-Task — **Command:** **GVL000**
Wait until 1st response form the camera: — 1
Wait until 2nd response form the camera: — Calib(A);C(B).000

9. Lock image trigger — **Command:** **SIH0001**
Wait until response form the camera — 1

10. Pick up the calibration part and place it on point (A) in angle (B)

11. Send Roboter coordinates from point (A) to camera
1st Command (x-Value) — **SFP000**<x-Value>
Wait until response form the camera — 1
2nd Command (y-Value) — **SFQ000**<y-Value>
Wait until response form the camera — 1

3x

12. Move the Roboter out of the cameras field of view (FOV)

13. Unlock image trigger — **Command:** **SIH0000**
Wait until response form the camera — 1

14. Wait 1 second (Recognition is beeing executed)

15. Go ahead until all 4 angle (B) in point (A) have been placed and recognized (repeat 3x)

3x

---

16. Read Roboter-Task — **Command:** **GVL000**
Wait until 1st response form the camera — 1
Wait until 2nd response form the camera — CalibZur

17. Lock image trigger — **Command:** **SIH0001**
Wait until response form the camera — 1

18. Pick up the calibration part and move it out of the cameras field of view

19. Unlock image trigger — **Command** — **SIH0000**
Wait until response form the camera

20. Wait 2 seconds (Recognition is beeing executed – No parts in the camera picture)

21. Read Roboter-Task — **Command:** **GVL000**
Wait until 1st response form the camera — 1
Wait until 2nd response form the camera — NoPart

22.) Repeat whole procedure with all points (A) (repeat 3x)

A = Calibration point, possible value 0, 1, 2, 3; B = Angle, possible value: 000, 090, 180, 270)

After finishing the calibration, all results from de procedure must be saved in the **.cxd** File on the camera.

1. Generate .cxd file	**Command:**	**SE2**
Wait until response form the camera		1

Furthermore, we recommend saving the calibration job after a successful run-through

1. Set camera offline	**Command:**	**SO0**
Wait until response form the camera		1
2. Save calibration job	**Command:**	**TF**<filename>**.job**
Wait until response form the camera		1
3. Set camera online	**Command:**	**SO1**
Wait until response form the camera		1

Reset Error state in calibration job

1. Set camera online	**Command:**	**SO1**
Wait until response form the camera		1
2. Lock image trigger	**Command:**	**SIH0001**
Wait until response form the camera		1
3. start calibration procedure	**Command:**	**SE1**
Wait until response form the camera		1
4. Unlock image trigger	**Command:**	**SIH0000**
Wait until response form the camera		1

Request states in feedWare calibration job

Initialize Status	**GVM023**	
Response 1 von camera	1	
Response 2 von camera	1.000	Initialization OK
	0.000	Not Initialized
	-1.000	Feeder not initialized

Shorts

GO	= Get Online state
GV	= Get Value
GF	= Get File
GJ	= Get Job
LF	= Load File
SE	= Set Event
SE1	= Start calibration procedure (equals cell P018)
SE2	= Save calibration data (equals cell P020)
SE6	= send Feeder Parameter
SE7	= Send manually written command strings from cell E000 to Feeder.
SF	= Set Float
SI	= Set Integer
SM	= Send Message
SO	= Set Online (0=Offline; 1=Online)
SS	= Set String
TF	= Save/Store File