Experimentation Facilities Specification

|  |  |  |  |
| --- | --- | --- | --- |
| **Date** | **Change description** | **Autor** | **Ver.** |
| 2021/12/03 | Added length units of responses from RoKi (mm). | Diego -  CANONICAL | 0.03 |
| 2021/12/13 | Changed document name to Experimental Facilities Specification.  Added Robot Cell description.  Added section for messages from robot cell.  Added topics for communication with robot cell LR-Mate/#. | Diego -  CANONICAL | 0.04 |
| 2021/12/13 | Fixed trajectory topic LR-Mate/receive-trajectory | Diego -  CANONICAL | 0.05 |
| 2021/12/13 | Added time format in Unix ms format.  Deleted Identifiers | Diego -  CANONICAL | 0.06 |
| 2021/12/14 | Fixed json format of quaternions. Changed external [] to {}. | Diego -  CANONICAL | 0.07 |
| 2021/12/14 | The requested data for Roki was listed  JSON data fields from robot cell was modified  Published topic was modified  Added python libraries used for image manipulation  Cell information section was modified | Mahboob Elahi TUNI | 0.08 |
| 2021/12/17 | * Simplification of the MQTT message format * Expecification of the MQTT default topic names * Expecification of the units | Alberto -  CANONCICAL | 0.09 |
| 2021/12/17 | Added information about LR-Mate 200iD base frame. | Diego - CANONICAL | 0.10 |
| 2021/12/20 | Requested for some test data from Roki for testing LR-Mate Robot cycle | Mahboob Elahi TUNI | 0.11 |
| 2022/03/21 | Fields added as per discussed in Friday’s telco | Mahboob Elahi TUNI | 0.12 |

Index

[Required Connector workflow 2](#_Toc396171472)

[Camera flow 3](#_Toc1020834489)

[Robot Flow 3](#_Toc1455584331)

[RoKi 4](#_Toc491785739)

[RoKi - S&MB Communication protocols 4](#_Toc784473729)

[Notation 4](#_Toc1677681688)

[Time 4](#_Toc1191700647)

[Quaternion 4](#_Toc1753129030)

[Joint/Angular poses 5](#_Toc790496837)

[Messages from RoKi demonstrator to Robot Cell 5](#_Toc1118833348)

[LR-Mate/receive-trajectory Json message 5](#_Toc1675278136)

[Robot Cell 6](#_Toc1910283927)

[Description 7](#_Toc1082512306)

[Messages from Robot Cell 7](#_Toc1878964154)

[Picture Data 8](#_Toc634896655)

[Cell Information 10](#_Toc1739721353)

[Notes: 10](#_Toc1728686820)

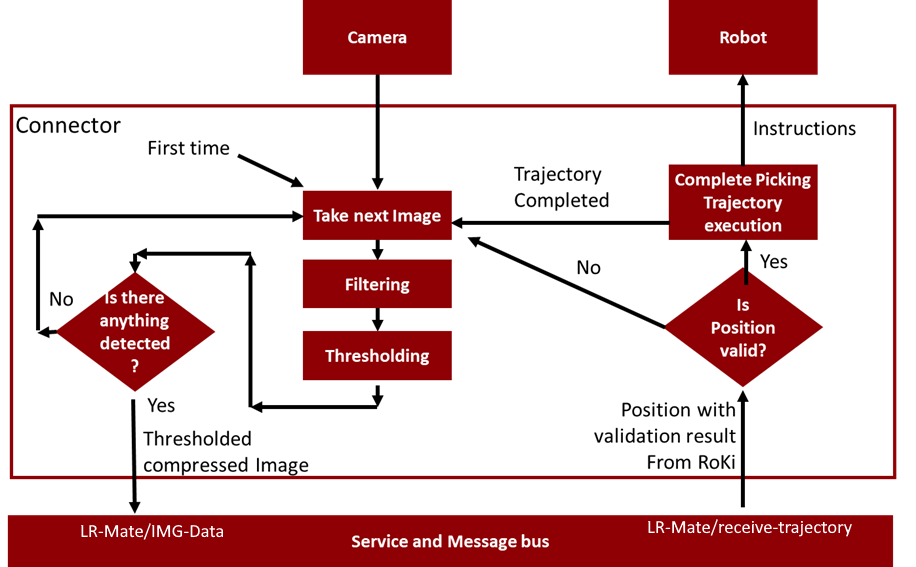
[Updated Robot Cell Information 10](#_Toc1131504144)

[Message Robot Cell 10](#_Toc1697278451)

[Serialization of Image to JSON at Middleware application 11](#_Toc1616214820)

[Part z\_dimention: 12](#_Toc1428121833)

# Required Connector workflow



There are two information flows.

1. Camera flow: The camera picture sending, triggered by the finalization of the last process cycle.
2. Root flow: The robot picking movement, triggered by the reception of a picking position.

## Camera flow

The camera flow is triggered by the finalization/cancelation of the process.

1. When **program starts**, the camera flow is launched, taking a **first photograph**.
2. The camera flow always starts **taking a photograph** of the table.
3. The photograph is **filtered**.
4. The picture is **binarized**.
5. Pixel counting or other method is used to **detect** if there is any **object in the table**.
6. If there is any **object detected**, the **image is sent** to the defined image topic (**LR-Mate/IMG-Data)** in the Message Bus, and the camera flow ends.
7. If there is **no object detected**, the camera flow is **restarted**, taking a **new picture**.

## Robot Flow

The robot flow starts when a position is received after camera flow sends a picture.

1. RoKi publishes a position with its validation result (The message is described in this document, it corresponds with Inverse Kinematics Request message) on **LR-Mate/ receive-trajectory**Topic.
2. The **connector receives** the **position** with the validation result.
3. The **message is parsed**, and the position and the validation result are extracted.
4. If the validation **result** is “**false**”, the **robot Workflow is canceled** and the Camera Workflow’s **picture taking** is triggered.
5. If the validation **result** is “**true**”, the **position is accepted**.
6. A **complete picking trajectory** is composed by the connector for the picking position (Using cartesian or joints, do not care).
7. The commands for the trajectory and gripper opening/closing are sent to the robot and **trajectory is performed**.
8. When the **trajectory ends**, the robot workflow ends and the **taking picture** action of the camera workflow is triggered.

# RoKi

## RoKi - S&MB Communication protocols

Several tasks can be performed in RoKi through S&MB:

* Cartesian to Angular pose conversion + pose verification request:
* Angular pose to cartesian pose request
* Error and event logging through a dedicated topic

## Notation

### Time

Unix Time format will be used. In this format ms from 1970-01-01 00:00.

### Quaternion

In this notation, the translation of the position is expressed with a real vector ( “p” = (<x>, <y> , <z>), in mm ) and its orientation with a quaternion ( “rijk” = <r> + <i>i + <j>j + <k>k ):

"quaternion": { “xyz” : [<x>, <y> , <z>], “rijk” : [<r>, <i>, <j>, <j>] }

The Units for lengths on positions are in **mm**, with a point for decimal separator.

### Joint/Angular poses

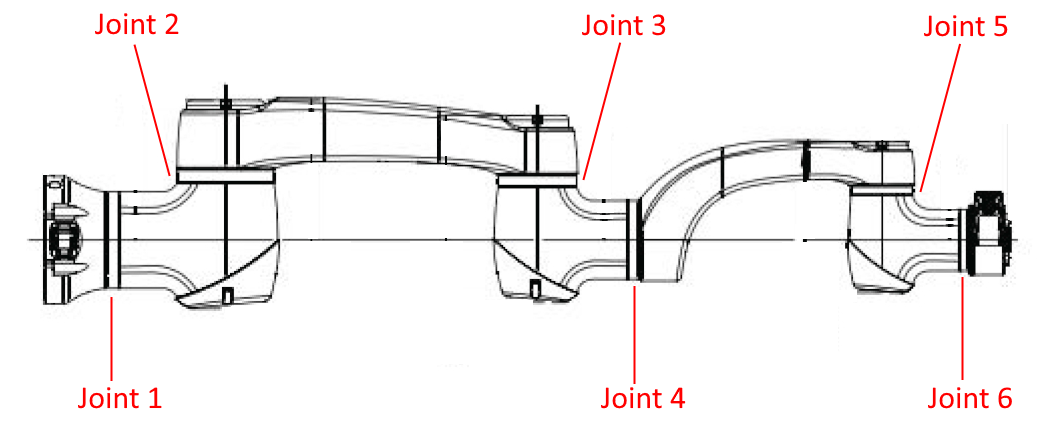
For representing angular poses of a robot, their values are listed following the joints of the robot, from the joint 1, that fix the robot to the ground, to the n-joint that corresponds to the end effector (joint 6 for 6-DOF robots).

Fig. Joints of a 6-DOF robot.

|  |
| --- |
| “rad”: [ <Joint 1>, <Joint 2>, <Joint 3>, <Joint 4>, <Joint 5>, <Joint 6> ] |

Notation for joint angles on radians.

|  |
| --- |
| “deg”: [ <Joint 1>, <Joint 2>, <Joint 3>, <Joint 4>, <Joint 5>, <Joint 6> ] |

Notation for joint angles on degrees.

## Messages from RoKi demonstrator to Robot Cell

These positions are the response to Inverse Kinematics requests. These responses are published in **LR-Mate/receive-trajectory.**

### LR-Mate/receive-trajectory Json message

Cartesian pose to Angular pose. Includes verification of the pose.

A message is sent to RoKi through the set topic. This message includes the pose expressed as Quaternion. The format of the cartesian position (with rotation on quaternion) must be agreed with “poseUnits” statement. . The field **<pose name>** will be filled with “pose1”.

The message send through the **LR-Mate/receive-trajectory** will have the following format:

|  |
| --- |
| {  "RequestId":"0a69389d",  "TimeStamp":"2021-12-16 13:48:53Z",  "Tcp":{"xyz":[0,0,0],"rijk":[1,0,0,0]},  "Pose":{"xyz":[0,0,0],"rijk":[1,0,0,0]},  "InverseKinematicSolutions":  [  {"j":[0,0,0,0,0,0]},  {"j":[0,0,0,0,0,0]}  ]  } |

Where:

* RequestID is an unique identifier of the message
* Pose is the picking position of the next item expressed in the base of the robot.
* InverseKinematicSolutions are all the possible joint configurations of the robot for the picking position. If this list is empty it means that the the piking position is not reachable by the robot.
* All the send and received using MQTT have a TimeStamp in Zulu format, that is defined by the ISO 8601 format, that is YYYY-MM-DDTHH:mm:ssZ.
* Lengths ("xyz") are given always in millimeters
* The joint angles of the robot ("j") are expressed in degrees
* Quaternions components ("rijk") are dimensionless

Size of the message for one pose with 8 solutions: near to 2048 Bytes.

# Robot Cell

## Description

Robot cell based on FANUC LR-Mate 200iD robot and a Sony XC56 camera.



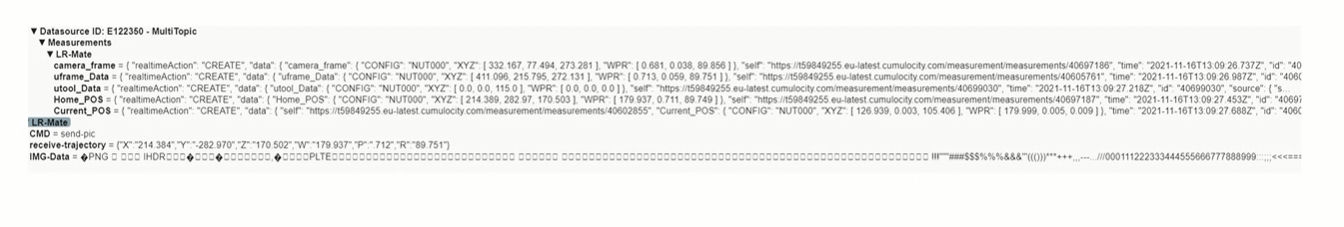
Fig. Robotic cell with the black plate for piece picking.

## Messages from Robot Cell

The robot cell is communicated directly to ZDMP’s Service and Message Bus at 192.168.100.100:30204

### Picture Data

The data from the camera is filtered, binarized, compressed, and published on **LR-Mate/IMG-Data** Topic.



The format of this data must be accorded.

### Cell Information

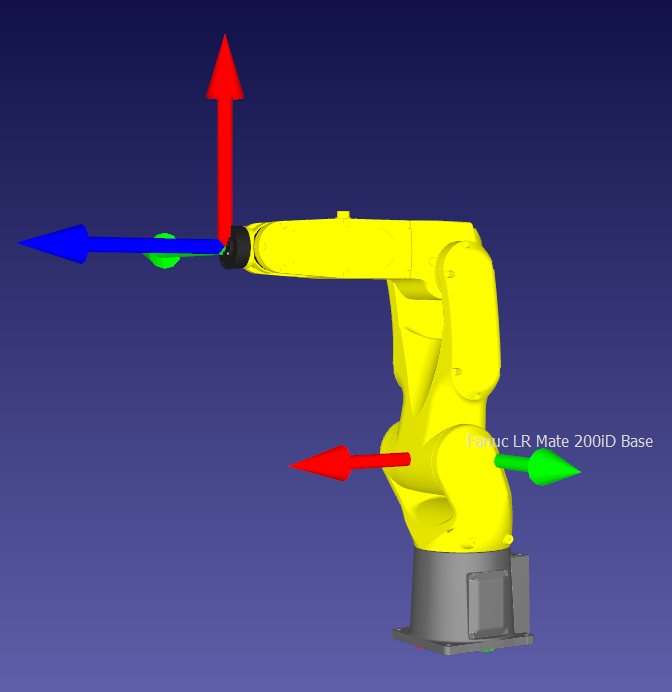
~~There is an additional topic that is used to provide information about the robot cell. This topic is~~

**~~Now~~** ~~there is only one topic~~ **~~“LR-Mate/data”~~** ~~which provides following data as JSON string over MQTT~~

* 1. ~~pix/mm~~
  2. ~~robot's TCP data~~
  3. ~~Origin of Image frame with respect to robot base~~
  4. ~~Image size (width and height)~~
  5. ~~Image format~~
  6. ~~Z\_dimention of part to be picked~~
  7. ~~Image data as string~~ *~~(in utf-8)~~*

## Robot base frame information

The LR-Mate 100iD robot has its “world” frame upper J1, instead of its base.



The location of the world frame of the robot from its physical base is:



Source: Fanuc LR-Mate 200iD model of RoboDK.

# Updated Robot Cell Information

**Now** there is only one topic **“LR-Mate/data”** which provides following data as JSON string over MQTT

1. pix/mm
2. robot's TCP data
3. Origin of Image frame with respect to robot base
4. Image size (width and height)
5. Image format
6. Z\_dimention of part to be picked
7. Image Type
8. Image data as string *(in utf-8)*

## Message Robot Cell

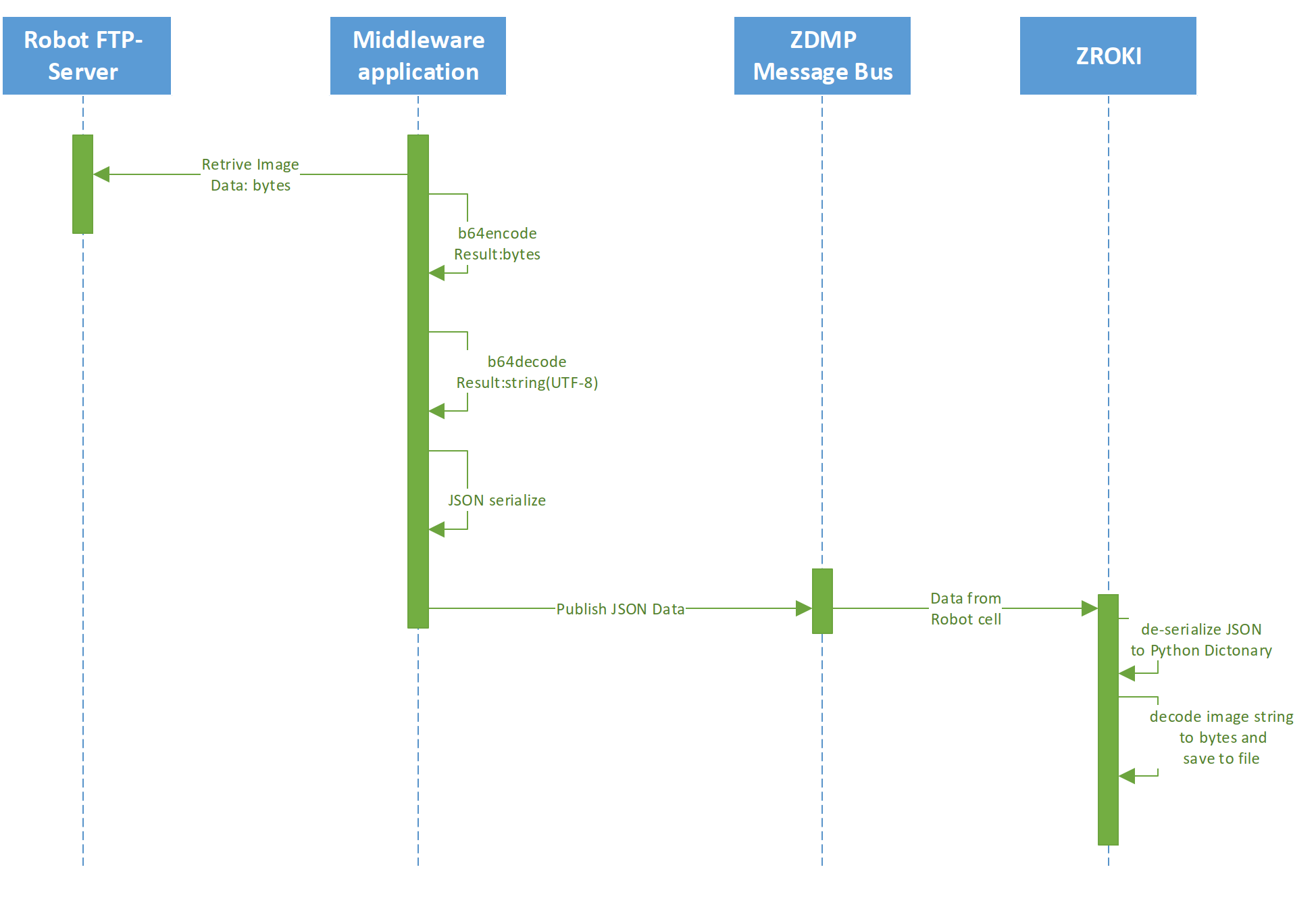
Robot Cell will publish the following JSON message to ZDMP service and message bus on topic **“LR-Mate/data”**

|  |
| --- |
| {  "ImageData": [  {  "pix/mm": 0.623,  "format": "PNG",  "Width": 640,  "Height": 480,  "Part\_Z\_dimention": -217.278  "Picture": Image Decoded as String (UTF-8)  }  ],  "RobotData": [  {  "utool\_Data": {  "XYZ": [0.0,0.0,115.0],  "WPR": [0.0,0.0,0.0],  "CONFIG": "NUT000"  }  },  {  "IMG\_ORIGN\_Pt": {  "XYZ": [269.177,-255.776,-254.474],  "WPR": [ -179.489,-0.789,-88.246],  "CONFIG": "NUT000"  }  }  ],  "timeStamp": "2021-12-15T14:27:37"  } |

## Serialization of Image to JSON at Middleware application

This process consists of the following steps

1. Middleware application retrieves workspace image from Robot’s FTP server in binaries. Python’s **“File Handling”** module is used to save the image binaries to a file.
2. Image bytes from step 1 are encoded to base64 bytes using pythons **“base64”** library.
3. The bytes obtain from step 2 are then decoded to UTF-8 string using pythons **“base64”.**
4. The image string is then serialize to JSON using python “**JSON”** library.
5. Following UML sequence diagram shows how image bytes are serialize to JSON as string (UTF-8)



1. zRoki just need to de-serialized the image data and then decode image string back to base64 bytes.

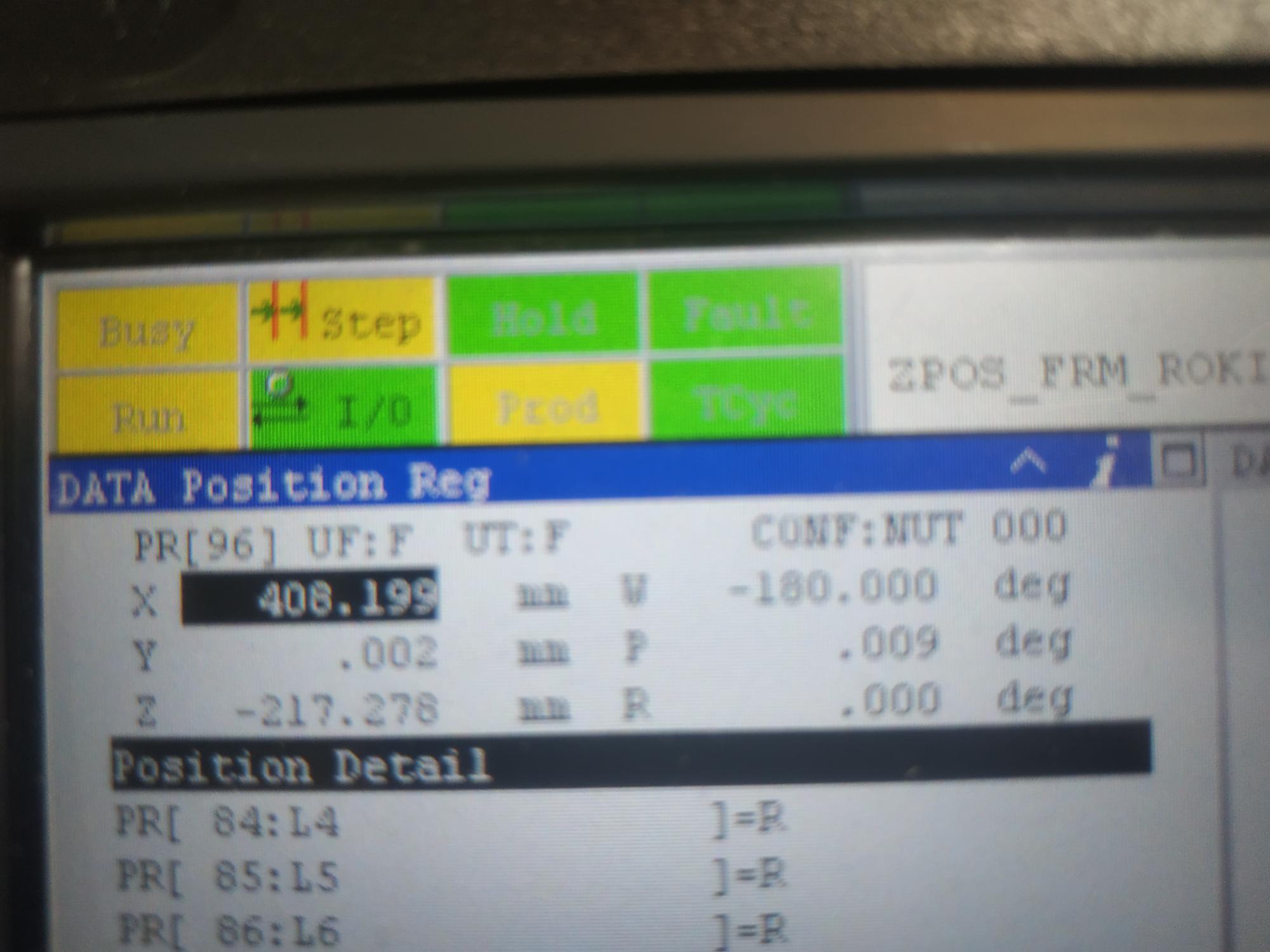
Useful links:

<https://www.codegrepper.com/code-examples/javascript/python+base64+string+json>

<https://www.youtube.com/watch?v=dW-b5S7cOTw>

## Part z\_dimention:

Following pictures shows the part to be picked and its z-dimention



# Requested Data:

At this time of development process, can CANONICAL provide some data for testing the “Robot cycle of LR-Mate”? I would be good to have some data from Roki to conduct local tests before going for real-time test.

Following is the data Roki for computing IK solution and pickup pose for the all cylinders in the attached picture.

{

"ImageData": [

{

"pix/mm": 0.623,

"format": "PNG",

"Width": 640,

"Height": 480,

"Part\_Z\_dimention": -269.974

**"Picture": above picture can be used**

}

],

"RobotData": [

{

"utool\_Data": {

"XYZ": [0.0,0.0,170],

"WPR": [0.0,0.0,0.0],

"CONFIG": "NUT000"

}

},

{

"IMG\_ORIGN\_Pt": {

"XYZ": [269.690,-256.519,-309.467],

"WPR": [ -179.489,-0.789,-88.246],

"CONFIG": "NUT000"

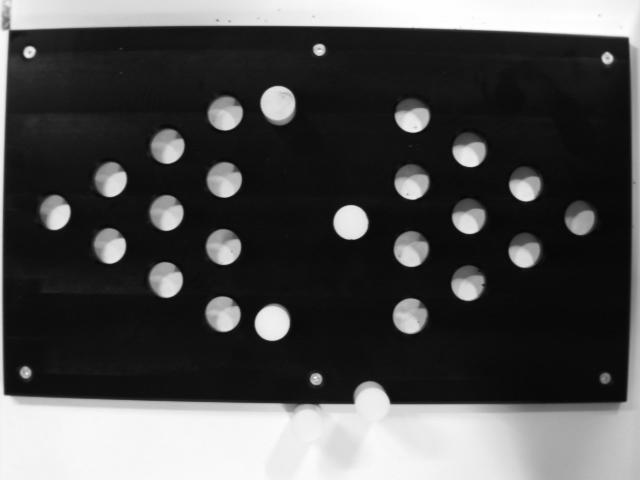
}

}

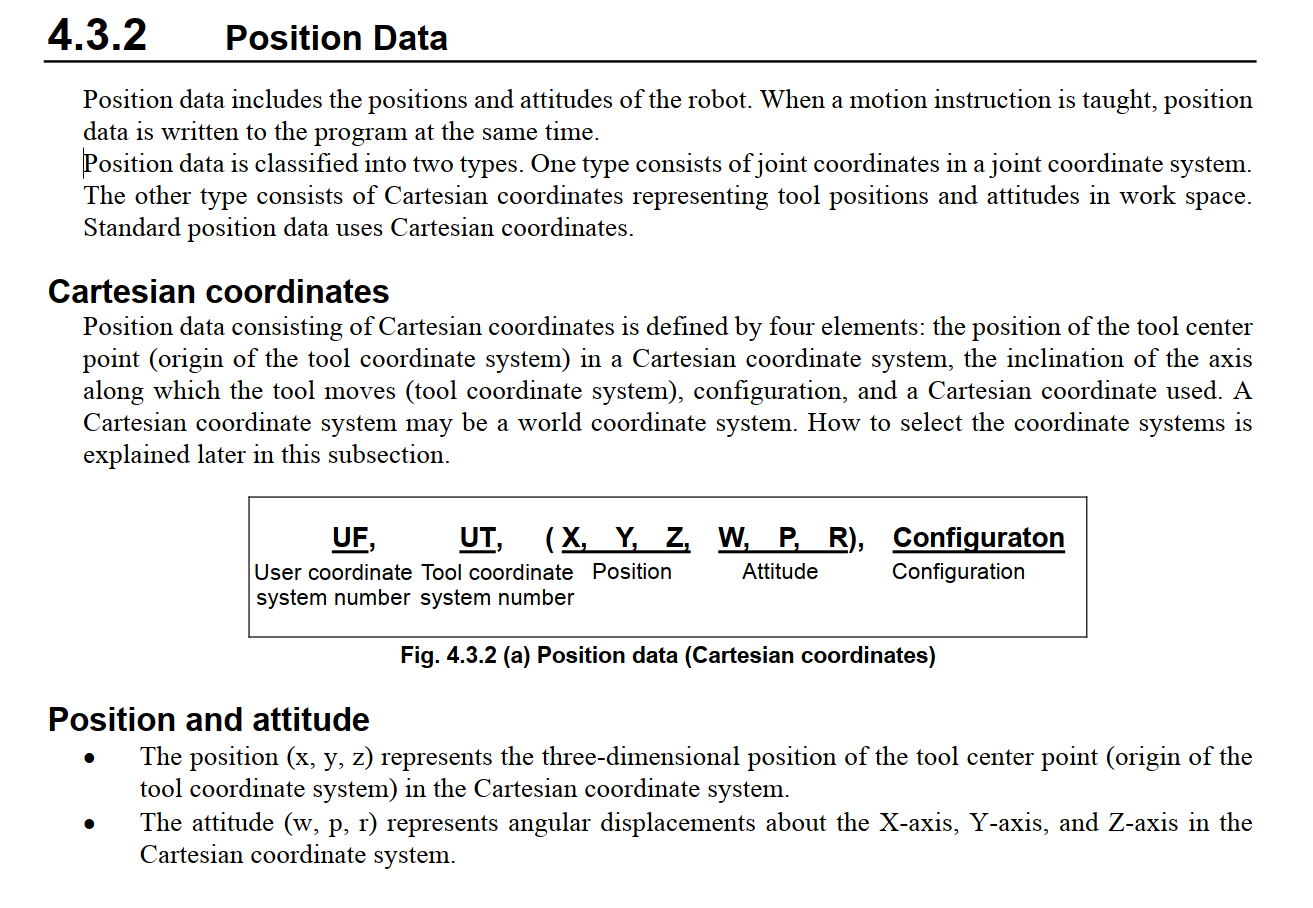
],

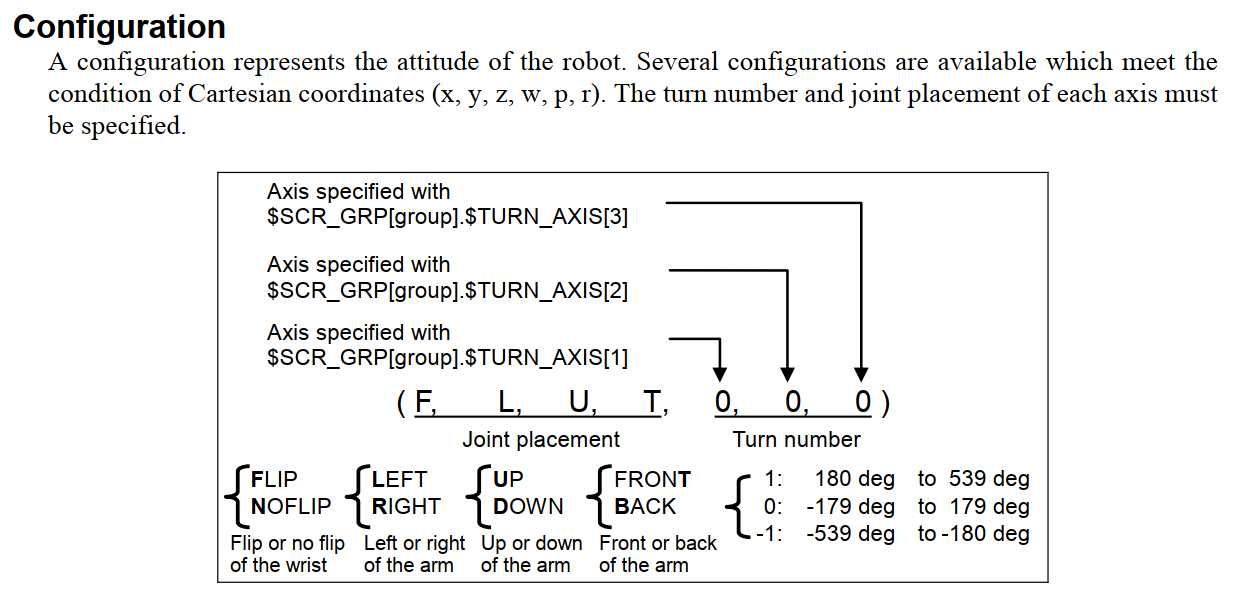
"timeStamp": "2021-12-15T14:27:37"

}



# FANUC Position Format:

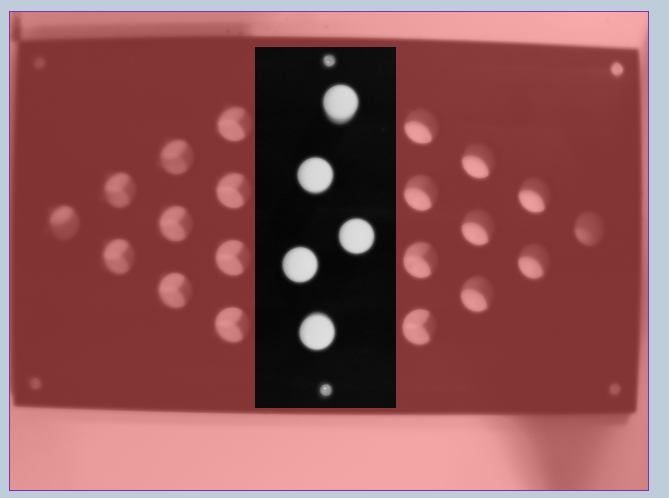




|  |  |
| --- | --- |
| **Field** | **Unit** |
| Position | mm |
| Attitude | deg |
| Eular-angle | XYZ convention used |

# Pickup place:

The pickup place for cylinders is the middle rectangular area between two trianguls as shown below.

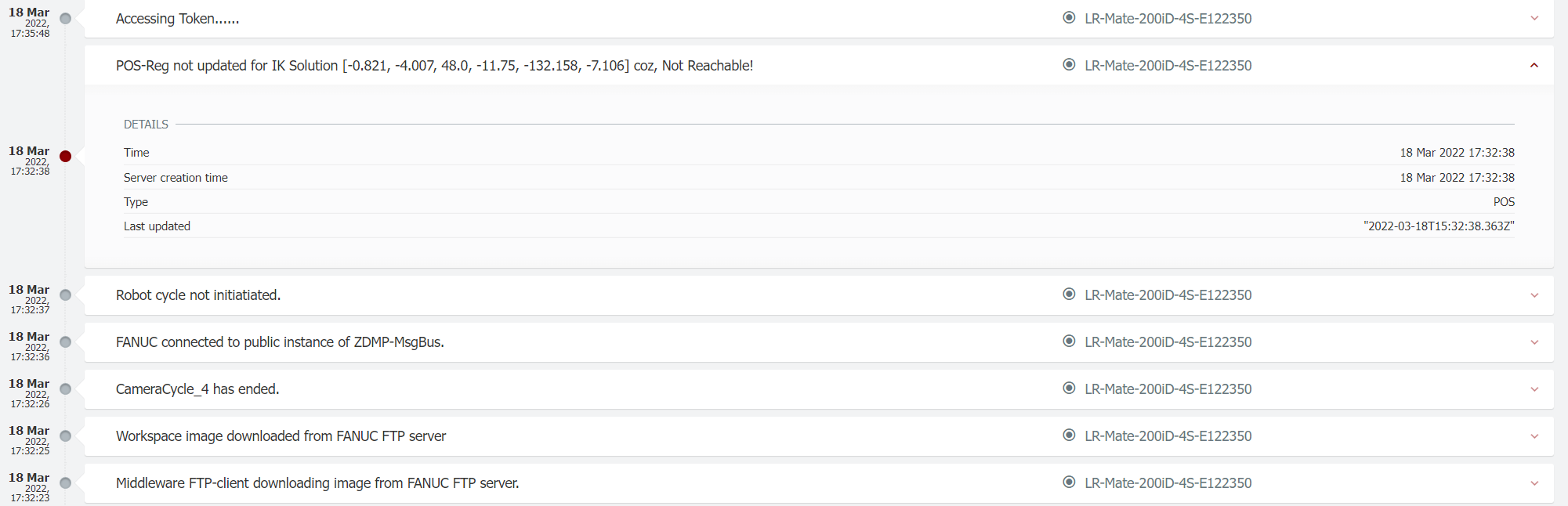


Pickup Area

PLACE Area

# Topic convention in MsgBus and DAQ

As I mention just for information, I added how data will be send using DAQ. Also, DAQ can maintain alarms/events/operations log for a device.



|  |  |
| --- | --- |
| DAQ | For this case data from FANUC sent to DAQ using “sendCustomMeasurment” end point and DAQ will publish data to Msg bus on following topic   * Send data to: T5\_1-Data-Acquisition/Datasource ID: **<DAQ-external-ID >** - MultiTopic/LR-Mate/ fromFANUC   If zRoki wants to send data using DAQ then it uses following Api: http://apigw-zdmp.platform.zdmp.eu/gateway/data-acquisition-synch-service/v0/sendCustomMeasurement?externalId=**E122350**&fragment=< LR-Mate/fromRoki > and DAQ will publish data on following topic   * Receive data at: T5\_1-Data-Acquisition/Datasource ID: **<DAQ-external-ID >** - MultiTopic/LR-Mate/ fromRoki |
| MsgBus (Agreed Communication Chanel) | * FANUC publish data to: LR-Mate/fromFANUC * FANUC listen to: LR-Mate/fromRoki |
| Data format (from FANUC) | {  "ImageData": [  {  "pix/mm": 0.623,  "format": "PNG",  "Width": 640,  "Height": 480,  "Part\_Z\_dimention": -269.974  "Picture": picture  }  ],  "RobotData": [  {  "utool\_Data": {  "XYZ": [0.0,0.0,170], //mm  "WPR": [0.0,0.0,0.0], //deg (roll/pitch/yaw)  "CONFIG": "NUT000"  }  },  {  "IMG\_ORIGN\_Pt": {  "XYZ": [269.690,-256.519,-309.467], //mm  "WPR": [ -179.489,-0.789,-88.246], //deg (roll/pitch/yaw)  "CONFIG": "NUT000"  }  }  ],  "timeStamp": "2021-12-15T14:27:37"  } |
|  |  |