Directories in Root folder:

|  |  |
| --- | --- |
| \_\_Backup\_\_ | Contains a backup copy of Python scripts, KAREL source/pcode files and workspace images |
| \_\_pycahe\_\_ | Python cache files |
| KAREL\_PROG | Loaded KAREL p-code files to robot controller |
| p-Codes | Contains only KAREL p-code files |
| Venv | Python virtual environment |
| . idea | N/A |

Files in Root Folder:

|  |  |
| --- | --- |
| File Name | Contribution to Middleware Application |
| Configurations.py | Contains Global constants like IP address, PORT numbers, Robot ID, credentials, message bus topics, URLs for ZDMP DAQ and FANUC Robot etc. |
| app.py | 1. It runs flask server. 2. Contains routes for FLASK server. 3. Establishes connection with ZDMP message Bus. 4. Start python Socket server which listens for incoming. connections from Robot Socket client. 5. For first time “app.py” triggers Robot Camera cycle. |
| FTP\_client.py | For each camera cycle this FTP client:   1. Establishes connection to Robot FTP server. 2. Navigate to “vision” directory, search for latest workspace picture and downloads that picture as binary image data. 3. Process newly downloaded picture and serializes to JSON string. |
| SocketServer\_Zv1\_0.py | Contains logic for:   1. listing robot socket client. 2. receiving and transforming positional data from robot. 3. triggers python FTP-client which connects to Robot FTP server for downloading latest workspace picture. |
| msgBus.py | A class which contains MQTT client abstractions. This script is used for creating MQTT client objects |
| Utility\_FUNC.py | This file contains helper functions like:   1. Device registration/integration to ZDMP DAQ component. 2. Transforming Image Bytes to JSON string with base64 encoding. 3. Processing joint angles coming from zRoki. 4. Updating robot position register. 5. Starting robot camera cycle. 6. Registering data source for ASYNC data access service with multi-topic feature. |
| SM\_client.py (No Contribution to middleware App) | This script runs a python Socket client on robot. I at a later stage reversed Socket communication between robot and middleware application. |
| subTest.py | This a tester script used for testing message bus subscriptions. This script has no contribution to middleware App. |
| Test.py | This script contains testing base64 encoding of image data |
| Requirements.txt | Installed python packages in this virtual environment. |

KAREL\_PROGS

|  |  |  |
| --- | --- | --- |
| File Name | File Name in TP | Contribution to Middleware Application |
| zv1\_1\_Orchestrator | z\_Orchstrate | Orchestrates KAREL routine |
| z\_Roki\_POS | z\_getRokiPOS | Validates and updates zRoki IK solutions |
| z\_Take\_IMG | z\_take\_PIC | Snap picture for each camera cycle |
| zV1\_0\_SocketClient | z\_TCPClient3 | Robot socket client which communicates with python socket server and transfers positional data |
| TCPserver3 | z\_TCPserver3 | Not in use anymore |

p-Codes: contains complied p-files of above mentioned KAREL source files.