**Protocols for Communicating between MATLAB and Robot Studio**

For controlling the robot arm with a graphical user interface made in MATLAB, a client/server based protocol style, using a pair of request and response message, is adopted for communication between MATLAB and Robot Studio. The client, i.e. MATLAB, sends a request message (usually an action) to the server, i.e. Robot Studio, and the server respond back with a message after processing the request.

The style of the request and response messages is detailed in the following sections.

**MATLAB ‘Client’ Requests**

The request message is a string message categorised into two main actions: to set an aspect of the robot workstation or to get information about an aspect of the robot workstation. The aspects of the robot is organized under seven ranks, the description of the ranks is shown in table 1.

The string message is formed by using an action code which corresponds to an aspect of the robot workstation, followed by a single space, then the value to be set or a single character ‘X’ as a placeholder for a get action. The first character of the code is a text and it determines the type of action. Please refer to the EXCEL file ‘Request Protocol Mapping’ for all the action code. The minimum number of characters that can be messaged is 5 and the maximum number of characters that can be messaged is 80.

**String Message Style**

String = [Code]/Space[Value]

**Request Examples**

1) Getting Emergency Stop Status

String = ‘G03 X’

2) Getting the conveyer direction status

String = ‘G34 X’

3) Setting End Effector Jogging X-Axis Position by -150.00 mm

String = ‘S17 -150.00’

4) Setting the robot to run installed module ‘MTRN4230\_Move\_Sample’

String = ‘S07 MTRN4230\_Move\_Sample’

5) Setting Joint 5 Jogging Orientation by 45.5 degrees

String = ‘S12 45.5’

Table 1 - Eight Ranks of the Request

|  |  |  |
| --- | --- | --- |
| Rank Number | Section Title | Description |
| I | ACTION | Indicates a retrieval of information from server or requesting a change in the workstation |
| II | FEATURE | Indicates a specific workstation subsystem |
| III | OPERATION TYPE | Only used for the manipulator subsystem. Indicates arm control via real time or programmed instructions |
| IV | JOGGING TYPE | Only used for jogging mode. Indicates the type of manipulator motion |
| V | MOVEMENT ARGUMENTS | Identifies the variables for motion control of manipulator arm |
| VI | DATA COMPONENTS | Identifies components of section V variables |
| VII | ASPECT | Can be workstation subsystem status or further categorization of the section VI components |
| VIII | VALUE | Used to set specific changes in the workstation |

**RAPID (‘Server’) Responses**

The response message is a string message categorised into three classes: success, client error and server error. Please refer to table 2 for the description of the class types.

The string message is usually a code composed of three number characters (see table 3 to table 5). The exception is when Robot Studio is responding to a request for an aspect of the robot workstation (i.e. ‘get’ action). For a successful get action, the string sent by Robot Studio should be ‘100/Space/ NUM STRING’ where NUM STRING represents the corresponding value or name to the requested.

Response Example for a ‘get’ Request

a) The client request the status of the jogging mode joint3 orientation and Robot Studio has successfully read the joint3 orientation value of 46.7 degree. Then the response is a string = ‘100 46.7’

Table 2 - Classes of Responses

|  |  |  |
| --- | --- | --- |
| **Code** | **Class** | **Description** |
| 1xx | Success | It means the action was successfully received, understood, and accepted. |
| 2xx | Client Error | It means the request contains incorrect syntax or cannot be fulfilled. |
| 3xx | Server Error | It means the server failed to fulfil an apparently valid request. |

Table 3 – Codes for Success Class

|  |  |  |
| --- | --- | --- |
| **Code** | **Text Message** | **Description** |
| 100 | Completed | The request is accepted for processing and the processing is completed. |
| 101 | Accepted | The request is accepted for processing, but the processing is not complete. |

Table 4 – Codes for Client Error Class

|  |  |  |
| --- | --- | --- |
| **Code** | **Text Message** | **Description** |
| 200 | Bad Request | The server did not understand the request. |
| 201 | Not Acknowledged | Light Curtain Obstruction not removed or General Mode Error is not acknowledged. |
| 202 | Forbidden | Access is forbidden. |
| 203 | Request Timeout | The request took longer than the server was prepared to wait. |
| 204 | Action Not Allowed | The action specified in the request is not allowed. |
| 205 | Conflict | The request could not be completed because of a conflict. |

Table 5 – Codes for Server Error Class

|  |  |  |
| --- | --- | --- |
| **Code** | **Text Message** | **Description** |
| 300 | Internal Server Error | The request was not completed. The server met an unexpected condition. |
| 301 | Not Implemented | Access is forbidden. |
| 302 | Service Unavailable | The request took longer than the server was prepared to wait. |
| 303 | Connection Timeout | The action specified in the request is not allowed. |