# Communications

## 1.1 COMMAND TYPES AND DETAILS

Their command legend is denoted by the bracketed letter

* Status Check (C)
* Status Change (H)
* Set DIO (D)
* Set Pose (P)
* Jog Joint (J)
* Jog Linear (L)
* Jog Stop (S)

### 1.1 a) Movement (input Movement Data detailed in 1.2)

#### Pose

1. specify a position in both con/table frames to move to (P)
2. specify a position of joint angles (P)
3. specify an end effector orientation (P)
4. click on a position in both feeds to move the end effector there (P)
5. speed set

#### Motion

1. specify a position in both con/table frames to move to (P)
2. specify a pose of joint angles (P)
3. specify an end effector orientation (P)
4. reorient

#### Jogging

1. manually move the robot in linear mode respect to base frame (L)
2. manually move the robot in linear mode respect to end effect (L)
3. reorient the end effector (L)
4. jog the joints (J)

### 1.1 b) Change status (input Status Data detailed in 1.2

1. shutdown - move robot to home position, avoid collision, set all DOs to 0
2. pause - stop current command but remember what its doing
3. resume - resume command
4. cancel - stop current command, remove commands from the queue

### 1.1 c) DIO change (input DIO Data detailed in 1.2)

1. VacSol - Solenoid starts the suck
2. VacPump - Engages pre-suck
3. ConRun - Makes conveyer go
4. ConDir - Conveyer Direction

### 1.1 d) Poll for status (No extra input)

1. safety systems
2. joint positions
3. DIO

## 1.2 COMMAND INPUTS

This section details all the inputs required to give to the MATLAB functions, which will then be sent to RobotStudio. All inputs are listed in order, with their type specified and units.

When inputting, please put in the order specified below. Example string to send would look like:

EXAMPLE: Sending pose changing from MATLAB -> Robot (P) with velocity 100:

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| BYTE | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 |
|  | ‘F’ | ‘P’ | ‘ d’ | Pi/4 | | Pi/4 | | .. | | .. | | .. | | .. | |
|  | HEAD | TYPE | VEL | J1 |  | J2 |  | J3 |  | J4 |  | J5 |  | J6 |  |

## 1.2 a) Movement Inputs and Formats/Units

### POSE CHANGING (P)

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| INPUTS | TYPE | SIZE | SEND TYPE | SIZE | Comments |
| Velocity (v100) | int | 4 | char | 1 |  |
| Joint Angles (radians) | float | 4 | 2 chars | 2 | rad \* /(2\*pi)\*(2^16-1)) |

### JOINT JOGGING (J)

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| INPUTS | TYPE | SIZE | SEND TYPE | SIZE | Comments |
| Velocity (v100) | int | 4 | char | 1 |  |
| Joint Number (radians) | int | 4 | char | 1 |  |

### LINEAR JOGGING (L)

No Inputs Required

### JOG STOP (S)

No Inputs Required

## 1.2 b) Change Status Inputs

## CHANGE STATUS (G)

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| INPUTS | TYPE | SIZE | SEND TYPE | SIZE | Comments |
| StatusType | string | x | string | 4 | Specify input  ‘Confirmation’  ‘Connected’ ... |

e.g. cmd = changeStatus(‘Confirmation’)

## c) Change DIO inputs (D)

## CHANGE DIO (D)

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| INPUTS | TYPE | SIZE | SEND TYPE | SIZE | Comments |
| DIOType | string | x | string | 4 | Specify input  ‘ConRun’  ‘ConDir’ ... |

e.g. cmd = changeDIO(‘VacSol’)

## 1.2 d ) Status Check

No Inputs

# 1.3 PACKET PROTOCOL

-------------------

1.3 a) MATLAB -> Robot

-------------------

For more information on the various types of inputs and

how to structure this, refer to section 1.2

BYTESUM

byte 1: HEADER 1

'F' orward

byte 2: COMMAND TYPE 2

Status Check (C)

Status Change (H)

Set Pose (P)

Jog Joint (J)

Jog Linear (L)

Jog Stop (S)

Set DIO (D)

byte 3 - x: DATA

See section 1.2 for things being sent from each type

--------------------

1.3 b) Robot -> MATLAB

--------------------

BYTESUM COMMENTS

HEADER 1

'B' ckward

STATUS BYTE

- confirmation 2 check this bit

- connected 3

- paused 4

- awaiting command 5 0 = false (busy), set to 1 at the very start

PREV COMMAND 6 i.e. last recieved Pose change (P)

SAFETY

- emergency stop 7 0 = deactivated, i.e. emergency stop isn't on

- light curtain 8 1 = activated, i.e. it's on

- motors 9

- hold to enable 10

- motion task error 11

DIO

- VacSol 12

- VacPump 13

- ConRun 14

- ConDir 15

- ConEnabled 16

JOINTS

- J1 17/18

- J2 19/20

- J3 21/22

- J4 23/24

- J5 25/26

- J6 27/28

---------------------------------------------------------------

1.4 COMMUNICATIONS METHOD PROTOCOL

---------------------------------------------------------------

1. MATLAB -> Send Command

- If theres a problem, run internal timer to time out

2. RobotStudio -> Reply

- Send confirmation bit checked

- Execute

3. MATLAB

- Wait for command finish

4. RobotStudio

- Movement confirmation when done (awaiting command = true)

--------------------------------------------------------------------------

2. Robot

---------------------------------------------------------------------------

process info communications

---------------------------

- incoming/outgoing byte

- command type

- subcommand type

- queue number

return info packets

---------------------

- waiting for command

- status

- pose

- joints

- end effector

- awaiting command

safety systems

---------------------

- emergency stop

- light curtain

- motors

- hold to enable

- motion task error

update status

---------------------

- robot studio

update CAD

---------------------

- robot studio

--------------------------------------------------------------------------

3.MATLAB

--------------------------------------------------------------------------

--------------------------------------------------------------------

3.1 GUI Elements and Stuff

--------------------------------------------------------------------

Startup and shutdown

---------------------

- startup (on command)

[SWPWindow | text] SWP

[SWPWindow | button] ask for SWP confirmation

[guideWindow | text] guide how to operate robot

- shutdown button (on command)

[main window | button] Shutdown

Robot Status

---------------------

- live update (live)

[mainWindow | label 1] DIO Status

[mainWindow | label 2] Joint angles

[mainWindow | label 3] End effector Pos/Orient

[mainWindow | label 4] awiting command

[mainWindow | text box 1] Sent Message Packet List

- don't print idle

[mainWindow | text box 2] Recv Message Packet List

- don't print idle

Diagnostics (on error)

---------------------

- Stops

[diagWindow | label 1] Emergency Stop

[diagWindow | label 2] Light Curtain

[diagWindow | label 3] Motor

[diagWindow | label 4] Flex Pendant Enable

[diagWindow | label 5] Execution Error/Motion task

--------------------------------------------------------------------

3.2 MATLAB Functions List

--------------------------------------------------------------------

Movement Functions

-------------------

string cmd = setJointPose(int J1, J2, J3, J4, J5, J6)

(times Joints angles by 1000)

string cmd = startJointJogging(int joint, int vel)

string cmd = stopJointJogging()

string cmd = startLinJogging()

Pose Functions

-------------------

[x ,y ,z] = findJAnglesToPos()

DIO Functions

------------------

string cmd = changeDIO(string DIOType)

Sends DIO in question to change, creates a toggle command

% List of DIO:

% VacSol (1)

% VacPump (2)

% ConRun (3)

% ConDir (4)

e.g. usage: changeDIO('VacSol')

setConRun()

setConDir()

setVacSol()

setVacPump()

Status Functions

-------------------

string cmd = changeStatus(char statusType, newState)

Sends status to change and new state of that status

shutdownRobot()

startupRobot()

pauseRobot()

resumeRobot()

Queue Functions

-------------------

queue = sendToQueue(string[] queue, string cmd) [done]

appends a command (string cmd) to the queue (string[] queue)

i.e "WOW"

queue = clearQueue(string[] queue) [done]

cancels all commands in the queue

queue = removeFromQueue(string[] queue, int cmdInd) [done]

Communications Functions

-----------------------

startConnection(IP)

opens a TCPIP socket to robot located at IP address 'IP'

sendToRobot(string cmd)

closeConnection(socket)

closes TCPIP socket

sendFromQueue(string[] queue, socket)

decodeMessage(string reply)

Computer Vision

-----------------------

blocks = detectBlocks(img)

Misc Helper Functions

-----------------------

string cmd = createCmdString(char cmdType, string data) (for creating communications strings)

--------------------------------------------------------------------

3.3 MATLAB Protocol

--------------------------------------------------------------------

cmd Strings - everything gets converted to hexadecimal, but unfortunately hex

is just a string, so we have to use 2 bytes per byte we're sending

--------------------------------------------------------------------

3.4 Misc Notes

--------------------------------------------------------------------

-------------------------

3.4.1 Taken letters for comms

-------------------------

B - for back sending

F - for foward send

-------------------------------------------------------------------------

4.NOTES

-------------------------------------------------------------------------

Notes:

Make sure theres a flag to make sure none of the stops are active