Page 9

Annex B. Developer's Manual

This Annex shall list and classify all functions
Toolbox, as well as handle objects, scripts, and the function to initiate a simulation. To find out how each function is used, they have help
Typing the Matlab command "help" plus the function, script or class to use; by Example: "help OnFwd".

B.1. Robot Control Functions

This section of the annex shall list, by category, all the functions that the Toolbox possesses for robot control; these are:

- Functions of actuators:
  - Off
  - OnFwd
  - OnRev
- Start sensor functions:
  - SetSensorHtGyro
  - SetSensorLight
  - SetSensorTouch
  - SetSensorUltrasonic
- Functions of sensors (in parentheses start function of sensor
  Has to be used beforehand to define the input port to be
  Use for this):
  - MotorRotationCount
  - SENSOR_1 (SetSensorLight or SetSensorTouch)
  - SENSOR_2 (SetSensorLight or SetSensorTouch)
  - SENSOR_3 (SetSensorLight or SetSensorTouch)
  - SENSOR_4 (SetSensorLight or SetSensorTouch)
  - Sensor (SetSensorLight or SetSensorTouch)
  - SensorHtGyro (SetSensorHtGyro)

- SensorUS (SetSensorUltrasonic)
- ResetRotationCount

• GUI functions:

- ButtonPressed
- ClearScreen
- NumOut
- TextOut

• file handling functions:

- CloseFile
- CreateFile

Page 10

Attachments

- DeleteFile
- OpenFileRead
- ReadLnString
- WriteLnString

• Other functions:

- CurrentTick
- FreeMemory

Stop

Wait

• Functions that do not belong to the repertoire of functions NXC (between Parenthesis a brief description of this):

- ResetAngle (resetting the angle counting mode on the sensor gyroscopic)
- SensorAngle (returns the angle counting mode angle in the Gyroscopic sensor)
- SensorColor (returns the color code read by the color sensor)
- StatusLight (turns the status lights on the interface on or off with the Color specified and if they blink)

• Private functions:

- privateSensor
- isFileNXC

B.2. Customer management classes

These objects are responsible for the communication and execution of functions in V-REP For the robot and simulation control; Can be used if you want to develop Another toolbox with another type of function structure or expand the developed one. In the Figure 112 the structure of the classes that implement these objects and detailed Content (attributes and methods). For more information on each function, use the Command "help" or generate all the documentation of the object with "doc", for example: "Doc ev3RemoteApi".

Page 11

Figure 112. Class diagram of the toolbox's handler objects

If you want to create a "vrepRemoteApi" instance, the simulator must be initiated; On the other hand, if you want to create an instance of "ev3RemoteApi", in addition To have V-REP started, the robot has to be loaded in the scene and the Simulation started.

B.3. RunCodeNXC Function

To be able to use all this toolbox without worrying about initialization Of the constants and the connection with the simulator, a function has been created that Hides this process and, in addition, allows the use of logs to facilitate the work of depuration. It is also responsible for managing exceptions so that, in case of Error, the connection to V-REP is not left open and the workspace is cleansed. In order to use this function we have the following calls:

Page 12

Attachments

- **"ejecutarCodigoNXC (String script)".** Run the NXC program with the Name "script". Also, writing as Matlab command is allowed: "ExecuteCodeEv3 script".

- **"ejecutarCodigoNXC (String script, Bool log)".** Run the NXC program With the name "script" and, if "log" is true or 1, save in a text file the Messages written in the Matlab console with the name "script".

- **"ejecutarCodigoNXC (String script, log Bool, String file)".** Same as him Above, except that the log file is saved with the name "file".