**Test A.1: Basic Actuator Creation**

1. The rationale behind the test; i.e., what is it testing and why we care.

This test shows that the actuator is successfully created on the network.

2. A general English description of the initial conditions of the test.

The initial conditions required for this test will be an empty network.

3. The commands for (2), which must appear in a standalone form that could be directly copied into a text file to reproduce the test without manual intervention. Do not cross-reference other tests.

@CONFIGURE LOG \"a.txt\" DOT SEQUENCE \"b.txt\" NETWORK \"c.txt\" XML \"d.txt\"  
CREATE SENSOR POSITION mySensor1  
CREATE ACTUATOR LINEAR myActuator1 GROUPS g1 g2 g3 g4 SENSOR mySensor1 ACCELERATION LEADIN 0.1 LEADOUT -0.2 RELAX 0.3 VELOCITY LIMIT 5 VALUE MIN 1 MAX 20 INITIAL 2 JERK LIMIT 3  
BUILD NETWORK WITH COMPONENTS myActuator1  
@exit

4. A brief English narrative of the expected results of executing the test. (Proper testing discipline expects that you do this before running the test.)

This test will show a proper creation of an actuator component. The actuator will be created and placed into groups g1-4 and have mySensor1 attached.

Text

Description automatically generated5. At least one representation of the actual results. The form is your choice.

6. A brief discussion on how the actual results differ from the expected results.

These results fit the expected results. The actuator can be seen on the network with the assigned attached components and in the correct groups.

7. A suggestion for how to extend this test to cover related aspects not required here

This test could be expanded to include more components watching the actuator and checking the data they report.

**Task A.2: Basic Actuator Manipulation**

1. The rationale behind the test; i.e., what is it testing and why we care.

This test shows the actuator’s ability to move from one position to another.

2. A general English description of the initial conditions of the test.

The initial conditions for this test will be an empty network. The test will create an actuator and show it moves.

3. The commands for (2), which must appear in a standalone form that could be directly copied into a text file to reproduce the test without manual intervention. Do not cross-reference other tests.

@CONFIGURE LOG \"a.txt\" DOT SEQUENCE \"b.txt\" NETWORK \"c.txt\" XML \"d.txt\"

CREATE ACTUATOR LINEAR myActuator1 GROUPS g1 g2 g3 g4 ACCELERATION LEADIN 0.1 LEADOUT -0.2 RELAX 0.3 VELOCITY LIMIT 5 VALUE MIN 1 MAX 20 INITIAL 2 JERK LIMIT 3

BUILD NETWORK WITH COMPONENTS myActuator1

@CLOCK WAIT UNTIL 0.5

@exit

4. A brief English narrative of the expected results of executing the test. (Proper testing discipline expects that you do this before running the test.)

The results of this test show the actuator’s position at each time step. The expected results are the actuator to begin moving and to reach the final expected position of 15 and then stop.

5. At least one representation of the actual results. The form is your choice.

|  |  |  |  |
| --- | --- | --- | --- |
| time | position | velocity | comment |
| 0.04 | 2 | 0 | StateAscendingLeadin |
| 0.05 | 2.1 | 0.1 | StateAscendingLeadin |
| 0.06 | 2.3 | 0.2 | StateAscendingLeadin |
| 0.07 | 2.6 | 0.3 | StateAscendingLeadin |
| 0.08 | 3 | 0.4 | StateAscendingLeadin |
| 0.09 | 3.5 | 0.5 | StateAscendingLeadin |
| 0.1 | 4.1 | 0.6 | StateAscendingLeadin |
| 0.11 | 4.8 | 0.7 | StateAscendingLeadin |
| 0.12 | 5.6 | 0.8 | StateAscendingLeadin |
| 0.13 | 6.5 | 0.9 | StateAscendingLeadin |
| 0.14 | 7.5 | 1 | StateAscendingLeadin |
| 0.15 | 8.6 | 1.1 | StateAscendingLeadin |
| 0.16 | 9.8 | 1.2 | StateAscendingLeadin |
| 0.17 | 10.8 | -1.2 | StateAscendingLeadin |
| 0.18 | 12 | -1 | StateAscendingLeadout |
| 0.19 | 13 | -0.8 | StateAscendingLeadout |
| 0.2 | 13.8 | -0.6 | StateAscendingLeadout |
| 0.21 | 14.4 | -0.4 | StateAscendingLeadout |
| 0.22 | 14.8 | -0.2 | StateAscendingLeadout |
| 0.23 | 15 | 0 | StateAscendingLeadout |

6. A brief discussion on how the actual results differ from the expected results.

The actuator’s motion meets expectation of increasing its position and velocity and then slowing to a stop at the target value of 15.

7. A suggestion for how to extend this test to cover related aspects not required here

Extending this test may include moving the actuator in both directions and changing directions while moving towards a different target destination.

**Task F.1: Ping Message to Actuator**

1. The rationale behind the test; i.e., what is it testing and why we care.

This test shows the message command can ping the master controller, and the forwarding master controller will send the ping to an attached actuator.

2. A general English description of the initial conditions of the test.

The initial conditions for this test are an empty network.

3. The commands for (2), which must appear in a standalone form that could be directly copied into a text file to reproduce the test without manual intervention. Do not cross-reference other tests.

@CONFIGURE LOG \"a.txt\" DOT SEQUENCE \"b.txt\" NETWORK \"c.txt\" XML \"d.txt\"

CREATE ACTUATOR LINEAR myActuator1 GROUPS g1 g2 g3 g4 ACCELERATION LEADIN 0.1 LEADOUT -0.2 RELAX 0.3 VELOCITY LIMIT 5 VALUE MIN 1 MAX 20 INITIAL 2 JERK LIMIT 3

BUILD NETWORK WITH COMPONENTS c1 myActuator1

SEND MESSAGE PING

@exit

4. A brief English narrative of the expected results of executing the test. (Proper testing discipline expects that you do this before running the test.)

The test results should show a ping message received by the controller and then forwarded to the attached components which should reply to the ping.

5. At least one representation of the actual results. The form is your choice.



6. A brief discussion on how the actual results differ from the expected results.

These results meet the expected results of the test.

7. A suggestion for how to extend this test to cover related aspects not required here.

The test could be extended by including more items on the controller, and even including subcomponents on the controller child items as well.