SWI-PRO-NXT: SWI-PROLOG TO LEGO NXT® INTERFACE

User manual

ROBOTICS LAB,
DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING,
INDIAN INSTITUTE OF TECHNOLOGY GUWAHATI, INDIA
http://www.iitg.ernet.in/cse/robotics

Table of Contents

1. INTRODUCTION

SWI-PRO-NXT is an interface developed to facilitate the control of the Mindstorm NXT® brick from SWI-PROLOG. This interface comprises three components viz. a SWI-PROLOG interface file, an Intermediate (Relay) server and a NXT® Control file. These three components synchronize to gain the overall control of the NXT® Robot. The Bluetooth® link is for communications with the NXT®. The Bluetooth® link between the computer and the NXT® must be established a priori. In its present form this interface can be used on a machine running Windows® XP/7. Figure 1 depicts the flow of the initial setup while figure 2 depicts the flow of information between the above mentioned three components of this interface.

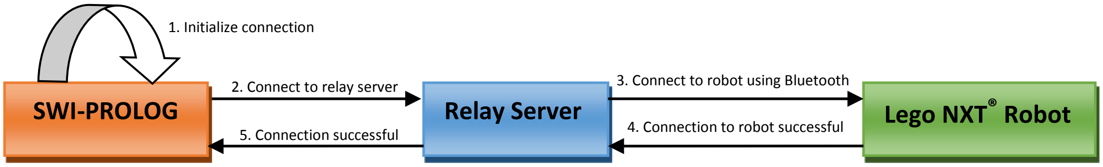
1. Initialize connection
SWI-PROLOG
2. Connect to relay server
5. Connection successful
Relay Server
3. Connect to robot using Bluetooth
4. Connection to robot successful
Lego NXT® Robot

Figure 1: The initial Connection Setup between WIN-PROLOG and the NXT®

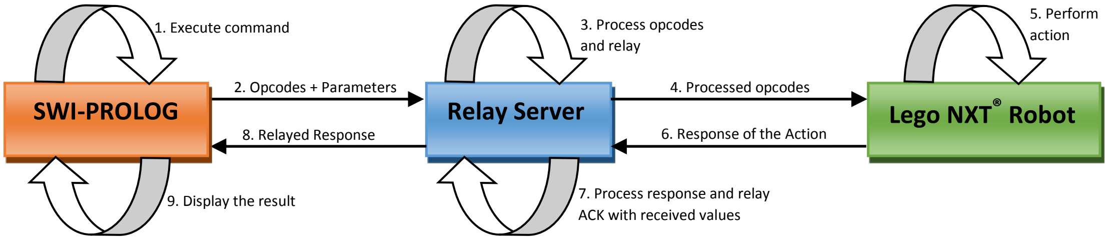

Figure 2: The information flow between SWI-PROLOG and NXT®

2. PRE-REQUISITES AND INITIAL SETUP

The pre-requisite check list for running this interface is:

1. The NXT® must be paired with the computer over the Bluetooth® link.
2. Following software needs to be installed:
   a. JDK 8 (*http://www.oracle.com/technetwork/java/javase/downloads/index.html*)
   b. LeJOS NXJ firmware 0.8.5 (*http://lejos.sourceforge.net/nxj-downloads.php*)
   c. SWI-PROLOG (http://www.swi-prolog.org/download/stable)
3. Check whether the CLASSPATH is set for LeJOS and Java. If not then set it. Just to ensure this type the following command in a console on the PC:

   nxj –h
4. This will display the **help** options for *nxj* LeJOS has been properly installed.
5. The NXT® Brick must be running on LeJOS firmware.
6. All the files of this interface must be placed in a single directory.

Execute the ***tartarus.exe*** file, it will extract interface file to C:\Tartarus folder and then follow the following steps.

Step 1:

Open the SWI-PROLOG console. Load **nxt_interface.pl** from **C:\Tartarus**. You may use one of the following methods to do so.

a. Click **File** then **Consult** and select the nxt_interface.pl file from C:\Tartarus.

OR

b. Type

consult('C:\Tartarus\nxt_interface.pl').

on the SWI-PROLOG console and then press **ENTER**.

Step 2:

a. Power ON the NXT®.

b. Ensure that Bluetooth® is ON in your computer system.

c. On the SWI-PROLOG console run the predicate using the following format:

setup_nxt('*Name of NXT®*').

The default name for NXT® Brick is **NXT** (case sensitive). This predicates takes a string argument. Before the end of this setup you will hear two beeps from the NXT® brick.

Example:

setup_nxt('NXT').

In this step ***nxtbin.nxj*** file provided in *Tartarus* folder is copied to NXT brick. You can manually copy file to NXT brick or you can use ***copy_files.bat*** file provided in *Tartarus* folder.

Step 3:

There are two predicates which define the start and end of a NXT® session.

a. initialize_nxt(Name,Link,Port,Response)

This predicates initializes the NXT® and creates the data stream over the Bluetooth® link (Your NXT® should be powered on before executing this predicate). This predicate must be executed once initially before starting an NXT® session.

Name <atom +>: Give the name of your NXT®. This will be the connection identifier for the whole NXT®session.
Link <atom +>: The host name or IP is to be typed here. If the same machine is used then use localhost.
Port <number +>: A TCP/IP port number for communication.
Response <atom - >: This is the ACK from the server. You do not have to provide a value to it. After the connection is established the ACK message will be displayed.

Example:

```
?-initialize_nxt('NXT', 'locahost', 8688,Response).

Response =started
```

b. close_nxt(Name)

Name <atom +>: Give the name of your NXT®. This will be the connection identifier for the whole NXT®session.

This predicate ends the NXT® session and must be executed at the end of the session.

Example:

```
close_nxt('NXT').
```

After completing the initial setup, the following predicates can be used to control the NXT®.

NOTE: If the above steps fail due to some reason, an alternate method for the Initial Setup is provided in APPENDIX I.

1 INTERFACE PREDICATES

The interface caters to the actuation of the motors and sensors and also files and their execution. The table below depicts the categories of these predicates and their main functions.

Sr. No.	Type of Predicates	Functions
1.	Motor Control	Used to control the motors connected to the NXT®.
2.	Display and Beep	Used to write onto the LCD and also generate beeps.
3.	Program Execution and File	Allows initiation of program control and managing files.
4.	Sensor Predicates	Allows reading the various sensors supplied along with the NXT®.

1.1 MOTOR CONTROL PREDICATES

1.1.1 move_forward_nxt(Name,Port,Speed)

Function: To move a motor connected to Port in forward direction at a specified speed.

Description:

Name <atom +>: Name of the socket that is given while establishing the connection with NXT® robot.
Port <atom +>: Port name of motor to drive.
Speed <integer +>: Speed at which the motor shaft is to be rotated.

Example:

move_forward_nxt('NXT', 'B', 100).

This will move the motor connected to port B of the NXT® at a speed of 100 in the forward direction.

1.1.2 move_forward_nxt(Name,Port1,Port2,Speed)

Function: To move two NXT® motors in forward direction simultaneously.

Description:

Name <atom +>: Name of the socket that is given while establishing the connection with NXT® robot.
Port1 <atom +>: Port name of the first NXT® motor.
Port2 <atom +>: Port name of the second NXT® motor.
Speed <integer +>: Speed at which the NXT® motors need to be driven.

Example:

move_forward_nxt('NXT', 'A', 'B', 100).

This will move the motors connected to port A and port B of the NXT® at a speed of 100 in the forward direction.

1.1.3 move_backward_nxt(Name,Port,Speed)

Function: To move a NXT® motor in the backward direction.

Description:

Name <atom +>: Name of the socket that is given while establishing the connection with NXT® robot.
Port <atom +>: Port name of the NXT® motor.
Speed <integer +> - Speed at which the NXT® motors need to be driven.

Example:

move_backward_nxt ('NXT', 'B', 100).

This will move the motor connected to port B of the NXT® at a speed of 100 in the backward direction.

1.1.4 move_backward_nxt(Name,Port1,Port2,Speed)

Function: To move two NXT® motors in the backward direction simultaneously.

Description:

Name <atom +>: Name of the socket that is given while establishing the connection with NXT® robot.
Port1 <atom +>: Port name of the first NXT® motor.
Port2 <atom +>: Port name of the second NXT® motor.
Speed <integer +>: Speed at which the NXT® motors need to be driven.

Example:

move_backward_nxt('NXT', 'A', 'B', 100).

This will move both the motors connected to port A and port B at a speed of 100 in the backward direction.

1.1.5 flt_nxt(Name,Port)

Function: To set the motor in float mode.

Description:

Name <atom +>: Name of the socket that is given while establishing the connection with NXT® robot.
Port <atom +>: Port name of the NXT® motor where mode is to be set.

Example:

flt_nxt('NXT', 'A').

This will set the motor connected to port A of the NXT® in float mode.

1.1.6 flt_nxt(Name,Port1,Port2)

Function: To set two motors in float mode at the same time.

Description:

Name <atom +>: Name of the socket that is given while establishing the connection with NXT® robot.
Port1 <atom +>: Port name of the first NXT® motor where mode is to be set.
Port2 <atom +>: Port name of the second NXT® motor where mode is to be set.

Example:

flt_nxt('NXT', 'A', 'B') .

This will set the motors connected to port A and port B of the NXT® in float mode.

1.1.7 stop_nxt(Name,Port)

Function: To stop the motor if it is moving.

Description:

Name <atom +>: Name of the socket that is given while establishing the connection with NXT® robot.
Port <atom +>: Port name of the NXT® motor which is to be stopped.

Example:

stop_nxt(‘NXT’, ‘A’).

This will stop the motor connected to port A of the NXT®.

1.1.8 stop_nxt(Name,Port1,Port2)

Function: To stop two motors if they are moving.

Description:

Name <atom +>: Name of the socket that is given while establishing the connection with NXT® robot.
Port1 <atom +>: Port name of the first NXT® motor to be stopped.
Port2 <atom +>: Port name of the second NXT® motor to be stopped.

Example:

stop_nxt(‘NXT’, ‘A’, ‘B’).

This will stop the motors connected to port A and port B.

1.1.9 rotate_nxt(Name,Port,Angle,Direction)

Function: To rotate the motor by a given angle.

Description:

Name <atom +> Name of the socket that is given while establishing the connection with NXT® robot.
Port <atom +> Port name of the NXT® motor that is to be rotated.
Angle <integer +> Angle by which the NXT® motor is to be rotated.
Direction <atom +> Direction in which the NXT® motor is to be rotated, use ‘A’ for anticlockwise and ‘C’ for clockwise.

Example:

rotate_nxt(‘NXT’, ‘A’,90, ‘A’) .

This will rotate the motor connected at port ‘A’ in anti-clockwise direction by 90 degrees.

1.1.10 reverse_nxt(Name,Port)

Function: To reverse the direction in which a NXT® motor is moving.

Description:

Name <atom +>: Name of the socket that is given while establishing the connection with NXT® robot.
Port <atom +>: Port name of the NXT® motor of which the direction is to be reversed.

Example:

reverse_nxt(‘NXT’, ‘B’).

This will move the NXT® motor connected to port B in the forward direction if it is moving in backward direction or vice versa. The speed remains unaffected.

1.1.11 reverse_nxt(Name,Port1,Port2)

Function: To reverse the direction of motion of two NXT® motors.

Description:

Name <atom +>: Name of the socket that is given while establishing the connection with NXT® robot.
Port1 <atom +>: Port name of the first NXT® motor of which the direction is to be reversed.
Port2 <atom +>: Port name of the second NXT® motor of which the direction is to be reversed.

Example:

reverse_nxt('NXT', 'A', 'B').

This will reverse the direction of movement of the two NXT® motors connected to port A and port B. The speed remains unaffected.

1.1.12 ismoving_nxt(Name,Port,Response)

Function: To check whether the motor on a given port is moving or not.

Description:

Name <atom +>: Name of the socket that is given while establishing the connection with NXT® robot.
Port <atom +>: Name of the port on which the NXT® motor is connected.
Response <integer ->: This will return Response=1 if the motor connected to the specified port and 0 otherwise.

Example:

ismoving_nxt('NXT', 'A',R) .
R=0

1.1.13 getmode_nxt(Name,Port,Response)

Function: To check the mode of the motor on the given port.

Description:

Name <atom +>: Name of the socket that is given while establishing the connection with NXT® robot.
Port <atom +>: Name of port on which the NXT® motor is connected.
Response <integer ->: The return value of Response will provide information on the state of the NXT® motor.

Response	**Mode**
1	Motor is moving forward (anticlockwise)
2	Motor is moving backward (clockwise)
3	Motor is not moving (stop)
4	Motor is in float mode

Example:

getmode_nx('NXT', 'A',R).
R=1

1.2 Display and Beep Control Predicates

1.2.1 write_lcd_nxt(Name,Corx,Cory,String)

Function: To write a string on the LCD of the NXT brick at given co-ordinates.

Description:

Name <atom +>: Name of the socket that is given while establishing the connection NXT® robot.
Corx <integer +>: X co-ordinate of the position.
Cory <integer +>: Y co-ordinate of the position.
String <atom +>: String to be written on the NXT® LCD screen.

Example:

write_lcd_nxt('NXT',2,2, 'hello').

This will write "hello" on the LCD of NXT® brick starting at position (2, 2).

1.2.2 clear_lcd_nxt(Name)

Function: To clear the LCD screen of the NXT® brick.

Description:

Name <atom +>: Name of the socket that is given while establishing the connection with NXT® robot.

Example:

clear_lcd_nxt('NXT').

This will clear the LCD of NXT brick.

1.2.3 beep_nxt(Name,Type)

Function: To make a beep sound.

Description:

Name <atom +>: Name of the socket that is given while establishing the connection with NXT® robot.
Type <integer +>: Type of beep.

Type	Beep
1	Simple Beep
2	Two Beeps
3	Beep Sequence
4	Buzz

Example:

```
beep_nxt('NXT', 1) .
```

This will cause the NXT® to generate a simple beep sound.

1.3 Program Execution Predicates

1.3.1 exec_nxt(Name,Program)

Function: To execute a file that is saved on the NXT® brick.

Description:

Name <atom +>: Name of the socket that is given while establishing the connection with NXT® robot.
Program <atom +>: Name of the file to be executed. File should be available in the NXT® brick.

Example:

```
exec_nxt('NXT', 'sample').
```

This will execute the file named *sample* that is saved on the NXT® brick.

1.4 File Control Predicates

1.4.1 del_nxt(Name,Program)

Function: To delete a file that is saved on the NXT® brick.

Description:

Name <atom +>: Name of the socket that is given while establishing the connection with NXT® robot.
Program <atom +>: Name of the file to be deleted.

Example:

```
exec_nxt('NXT', 'sample').
```

This will delete the file named "sample" that is saved on the NXT® brick.

1.4.2 exists_nxt(Name,Program,Response)

Function: To check if a file exists on the NXT® brick.

Description:

Name <atom +>: Name of the socket that is given while establishing the connection with NXT® robot.
Program <atom +>: Name of the file whose existence is to be checked.
Response <atom ->: Response message.

Example:

```
exists_nxt('NXT', 'sample',Response).
```

This will return Response=1 if the file named "sample" exists on the NXT® brick and 0 otherwise.

1.4.3 download_nxt(Name,File,Path)

Function: To download a .java file from the controlling computer to the NXT® brick in an executable format.

Description:

Name <atom +>: Name of the socket that is given while establishing the connection with NXT® robot.
File <atom +>: Name of the file to be downloaded.
Path <atom +>: Path to the file.

Example:

download_nxt('NXT', 'sample', 'C:\Users\Navin\Desktop\ISI').

This will download the file named "sample.java" in the given path to the NXT® brick in the executable format.

1.4.4 get_files_nxt(Name,Response)

Function: To get the list of files on the NXT®.

Description:

Name <atom +>: Name of the socket that is given while establishing the connection with NXT® robot.
Response <atom ->: The list of files.

Example:

get_files_nxt('NXT', R).

This will display:
Sample1 Sample2 Sample3 Sample4
Sample1, Sample2, Sample3 and Sample4 are the files resident on the NXT®.

1.5 Sensor Predicates

1.5.1 Light Sensor

1.5.1.1 init_light_nxt(Name,Port)

Function: To initialize a NXT® light sensor connected at a given port.

Description:

Name <atom +>: Name of the socket that is given while establishing the connection with NXT® robot.
Port <integer +>: Name of the sensor port at which the NXT® light sensor is connected (Possible values are 1, 2, 3 and 4).

Example:

init_light_nxt('NXT', 1) .

This will initialize the NXT® light sensor connected at port 1.

1.5.1.2 *init_light_nxt(Name,Port1,Port2)*

Function: To initialize two NXT® light sensors connected to given ports.

Description:

Name <atom +>: Name of the socket that is given while establishing the connection with NXT® robot.
Port1 <integer +>: Name of the sensor port at which the first NXT® light sensor is connected.
Port2 <integer +>: Name of the sensor port at which the second NXT® light sensor is connected.

Example:

init_light_nxt('NXT', 1,2) .

This will initialize the NXT® light sensors connected to port 1 and 2 of the NXT®.

1.5.1.3 *init_light_nxt(Name,Port1,Port2,Port3)*

Function: To initialize three NXT® light sensors connected to given ports.

Description:

Name <atom +>: Name of the socket that is given while establishing the connection with NXT® robot.
Port1 <integer +>: Name of the sensor port at which the NXT® first light sensor is connected.
Port2 <integer +>: Name of the sensor port at which the NXT® second light sensor is connected.
Port3 <integer +>: Name of the sensor port at which the NXT® third light sensor is connected.

Example:

init_light_nxt('NXT', 1,2,3) .

This will initialize the NXT® light sensors connected to ports 1, 2 and 3.

1.5.1.4 *init_light_nxt(Name,Port1,Port2,Port3,Port4)*

Function: To initialize four NXT® light sensors connected at a given ports.

Description:

Name <atom +>: Name of the socket that is given while establishing the connection with NXT® robot.
Port1 <integer +>: Name of the sensor port at which the first NXT® light sensor is connected.
Port2 <integer +>: Name of the sensor port at which the second NXT® light sensor is connected.
Port3 <integer +>: Name of the sensor port at which the third NXT® light sensor is connected.
Port4 <integer +>: Name of the sensor port at which the fourth NXT® light sensor is connected.

Example:

init_light_nxt('NXT', 1,2,3,4).

This will initialize the NXT® light sensors connected at ports 1, 2, 3 and 4.

1.5.1.5 *read_light_nxt(Name,Port,Response)*

Function: To read the value of a light sensor.

Description:

Name <atom +>: Name of the socket that is given while establishing the connection with NXT® robot.
Port <integer +>: Name of port at which the NXT® light sensor is connected.

Response <integer ->: Value read by the NXT® light sensor.

Example:

read_light_nxt('NXT',1,R).

This will display R= (some integer value). R is the value read by the light sensor connected to port 1.

1.5.1.6 read_light_nxt(Name,Port1,Port2,Response1,Response2)

Function: To read the value of two NXT® light sensors.

Description:

Name <atom +>: Name of the socket that is given while establishing the connection with NXT® robot.
Port1 <integer +>: Name of port at which the first NXT® light sensor is connected.
Port2 <integer +>: Name of port at which the second NXT® light sensor is connected.
Response1 <integer ->: Value read by the first NXT® light sensor.
Response2 <integer ->: Value read by the second NXT® light sensor.

Example:

read_light_nxt('NXT',1,2,R1,R2) .

This will display R1 = (some integer value), R2= (some integer value). R1 and R2 are the value read by the light sensors connected to port 1 and port 2 respectively.

1.5.1.7 floodlight_nxt(Name,Port,State)

Function: To turn on/off the flood light of a light sensor.

Description:

Name <atom +>: Name of the socket that is given while establishing the connection with NXT® robot.
Port <integer +>: Name of port at which the NXT® light sensor is connected.
State <atom +>: State of the flood light to be set. Value can be either "on" or "off".

Example:

floodlight_nxt('NXT',1, 'off') .

This will turn off the flood light of the light sensor connected at port 1.

1.5.2 Sound Sensor

1.5.2.1 init_sound_nxt(Name,Port)

Function: To initialize a sound sensor connected at a given port.

Description:

Name <atom +>: Name of the socket that is given while establishing the connection with NXT® robot.
Port <integer +>: Name of the sensor port at which the NXT® sound sensor is connected
(Possible values are 1, 2, 3 and 4).

Example:

init_sound_nxt('NXT', 1).

This will initialize the NXT® sound sensor connected at port 1.

1.5.2.2 *init_sound_nxt(Name,Port1,Port2)*

Function: To initialize two NXT® sound sensors connected to given ports.

Description:

Name <atom +>: Name of the socket that is given while establishing the connection with NXT® robot.
Port1 <integer +>: Name of the sensor port at which the first NXT® sound sensor is connected.
Port2 <integer +>: Name of the sensor port at which the second NXT® sound sensor is connected.

Example:

init_sound_nxt('NXT', 1,2) .

This will initialize the sound sensors connected to port 1 and 2 of the NXT®.

1.5.2.3 *init_sound_nxt(Name,Port1,Port2,Port3)*

Function: To initialize three NXT® sound sensors connected to given ports.

Description:

Name <atom +>: Name of the socket that is given while establishing the connection with NXT® robot.
Port1 <integer +>: Name of the sensor port at which the first NXT® sound sensor is connected.
Port2 <integer +>: Name of the sensor port at which the second NXT® sound sensor is connected.
Port3 <integer +>: Name of the sensor port at which the third NXT® sound sensor is connected.

Example:

init_sound_nxt('NXT', 1,2,3) .

This will initialize the NXT® sound sensors connected to ports 1, 2 and 3.

1.5.2.4 *init_sound_nxt(Name,Port1,Port2,Port3,Port4)*

Function: To initialize four NXT® sound sensors connected to given ports.

Description:

Name <atom +>: Name of the socket that is given while establishing the connection with NXT robot.
Port1 <integer +>: Name of the sensor port at which the first NXT® sound sensor is connected.
Port2 <integer +>: Name of the sensor port at which the second NXT® sound sensor is connected.
Port3 <integer +>: Name of the sensor port at which the third NXT® sound sensor is connected.
Port4 <integer +>: Name of the sensor port at which the fourth NXT® sound sensor is connected.

Example:

init_sound_nxt('NXT', 1,2,3,4) .

This will initialize the NXT® sound sensors connected at ports 1, 2, 3 and 4.

1.5.2.5 *read_sound_nxt(Name,Port,Response)*

Function: To read the value of a NXT® sound sensor.

Description:

Name <atom +>: Name of the socket that is given while establishing the connection with NXT® robot.
Port <integer +>: Name of port at which the NXT® sound sensor is connected.
Response <integer ->: Value read by the NXT® sound sensor.

Example:

read_sound_nxt('NXT',1,R) .

This will display R= (some integer value). R is the value read by the NXT® sound sensor connected to port 1.

1.5.2.6 *read_sound_nxt(Name,Port1,Port2,Response1,Response2)*

Function: To read the value of two NXT® sound sensors.

Description:

Name <atom +>: Name of the socket that is given while establishing the connection with NXT® robot.
Port1 <integer +>: Name of port at which the first NXT® sound sensor is connected.
Port2 <integer +>: Name of port at which the second NXT® sound sensor is connected.
Response1 <integer ->: Value read by the first NXT® sound sensor.
Response2 <integer ->: Value read by the second NXT® sound sensor.

Example:

read_sound_nxt('NXT',1,2,R1,R2) .

This will display R1= (some integer value), R2= (some integer value). R1 and R2 are the value read by the sound sensors connected to port 1 and port 2 respectively.

1.5.3 Ultrasonic Sensor

1.5.3.1 *init_ultrasonic_nxt(Name,Port)*

Function: To initialize an ultrasonic sensor connected at a given port.

Description:

Name <atom +>: Name of the socket that is given while establishing the connection with NXT® robot.
Port <integer +>: Name of the sensor port at which the NXT® ultrasonic sensor is connected (Possible values are 1, 2, 3 and 4).

Example:

init_ultrasonic_nxt('NXT',1).

This will initialize the NXT® ultrasonic sensor connected at port 1.

1.5.3.2 *init_ultrasonic_nxt(Name,Port1,Port2)*

Function: To initialize two NXT® ultrasonic sensors connected at a given ports.

Description:

Name <atom +>: Name of the socket that is given while establishing the connection with NXT® robot.
Port1 <integer +>: Name of the sensor port at which the first NXT® ultrasonic sensor is connected.
Port2 <integer +>: Name of the sensor port at which the second NXT® ultrasonic sensor is connected.

Example:

init_ultrasonic_nxt('NXT', 1,2) .

This will initialize the ultrasonic sensors connected to port 1 and 2 of the NXT®.

1.5.3.3 *init_ultrasonic_nxt(Name,Port1,Port2,Port3)*

Function: To initialize three NXT® ultrasonic sensors connected to given ports.

Description:

Name <atom +>: Name of the socket that is given while establishing the connection with NXT® robot.
Port1 <integer +>: Name of the sensor port at which the first NXT® ultrasonic sensor is connected.
Port2 <integer +>: Name of the sensor port at which the second NXT® ultrasonic sensor is connected.
Port3 <integer +>: Name of the sensor port at which the third NXT® ultrasonic sensor is connected.

Example:

```
init_ultrasonic_nxt(‘NXT’, 1,2,3) .
```

This will initialize the NXT® ultrasonic sensors connected at ports 1, 2 and 3.

1.5.3.4 *init_ultrasonic_nxt(Name,Port1,Port2,Port3,Port4)*

Function: To initialize four NXT® ultrasonic sensors connected to given ports.

Description:

Name <atom +>: Name of the socket that is given while establishing the connection with NXT® robot.
Port1 <integer +>: Name of the sensor port at which the first NXT® ultrasonic sensor is connected.
Port2 <integer +>: Name of the sensor port at which the second NXT® ultrasonic sensor is connected.
Port3 <integer +>: Name of the sensor port at which the third NXT® ultrasonic sensor is connected.
Port4 <integer +>: Name of the sensor port at which the NXT® fourth ultrasonic sensor is connected.

Example:

```
init_ultrasonic_nxt(‘NXT’, 1,2,3,4) .
```

This will initialize the ultrasonic sensors connected at ports 1, 2, 3 and 4.

1.5.3.5 *read_ultrasonic_nxt(Name,Port,Response)*

Function: To read the value of a NXT® ultrasonic sensor.

Description:

Name <atom +>: Name of the socket that is given while establishing the connection with NXT® robot.
Port <integer +>: Name of port at which the NXT® ultrasonic sensor is connected.
Response <integer ->: Value read by the NXT® ultrasonic sensor.

Example:

```
read_ultrasonic_nxt(‘NXT’,1,R).
```

This will display R= (some integer value). R is the value read by the NXT® ultrasonic sensor connected to port 1.

1.5.3.6 *read_ultrasonic_nxt(Name,Port1,Port2,Response1,Response2)*

Function: To read the value of two NXT® ultrasonic sensors.

Description:

Name <atom +>: Name of the socket that is given while establishing the connection with NXT® robot.
Port1 <integer +>: Name of port at which the first NXT® ultrasonic sensor is connected.
Port2 <integer +>: Name of port at which the second NXT® ultrasonic sensor is connected.
Response1 <integer ->: Value read by the first NXT® ultrasonic sensor.

Response2 <integer ->: Value read by the second NXT® ultrasonic sensor.

Example:

read_ultrasonic_nxt('NXT',1,2,R1,R2) .

This will display R1= (some integer value), R2= (some integer value). R1 and R2 are the value read by the NXT® ultrasonic sensors connected to port 1 and port 2 respectively.

1.5.4 Touch Sensor

1.5.4.1 init_touch_nxt (Name,Port)

Function: To initialize a touch sensor connected at a given port.

Description:

Name <atom +>: Name of the socket that is given while establishing the connection with NXT® robot.
Port <integer +>: Name of the sensor port at which the NXT® touch sensor is connected
(Possible values are 1, 2, 3 and 4).

Example:

init_touch_nxt('NXT', 1) .

This will initialize the NXT® touch sensor connected at port 1.

1.5.4.2 init_touch_nxt(Name,Port1,Port2)

Function: To initialize two NXT® touch sensors connected at a given ports.

Description:

Name <atom +>: Name of the socket that is given while establishing the connection with NXT® robot.
Port1 <integer +>: Name of the sensor port at which the first NXT® touch sensor is connected.
Port2 <integer +>: Name of the sensor port at which the second NXT® touch sensor is connected.

Example:

init_touch_nxt('NXT', 1,2) .

This will initialize the NXT® touch sensors connected at port 1 and 2 of the NXT®.

1.5.4.3 init_touch_nxt(Name,Port1,Port2,Port3)

Function: To initialize three NXT® touch sensors connected to given ports.

Description:

Name <atom +>: Name of the socket that is given while establishing the connection with NXT robot.
Port1 <Integer +>: Name of the sensor port at which the first touch sensor is connected.
Port2 <Integer +>: Name of the sensor port at which the second touch sensor is connected.
Port3 <Integer +>: Name of the sensor port at which the third touch sensor is connected.

Example:

init_touch_nxt('NXT', 1,2,3).

This will initialize the NXT® touch sensors connected at ports 1, 2 and 3.

1.5.4.4 *init_touch_nxt(Name,Port1,Port2,Port3,Port4)*

Function: To initialize four NXT® touch sensors connected to given ports.

Description:

Name <atom +>: Name of the socket that is given while establishing the connection with NXT® robot.
Port1 <integer +>: Name of the sensor port at which the first NXT® touch sensor is connected.
Port2 <integer +>: Name of the sensor port at which the second NXT® touch sensor is connected.
Port3 <integer +>: Name of the sensor port at which the third NXT® touch sensor is connected.
Port4 <integer +>: Name of the sensor port at which the fourth NXT® touch sensor is connected.

Example:

```
init_touch_nxt('NXT', 1,2,3,4) .
```

This will initialize the NXT® touch sensors connected at ports 1, 2, 3 and 4.

1.5.4.5 *read_touch_nxt(Name,Port,Response)*

Function: To read the value of a touch sensor.

Description:

Name <atom +>: Name of the socket that is given while establishing the connection with NXT® robot.
Port <integer +>: Name of port at which the NXT® touch sensor is connected.
Response <integer ->: Value read by the NXT® touch sensor.

Example:

```
read_touch_nxt('NXT',1,R).
```

This will display R=1/0. R is the value read by the NXT® touch sensor connected to port 1.

1.5.4.6 *read_touch_nxt(Name,Port1,Port2,Response1,Response2)*

Function: To read the values of two NXT® touch sensors.

Description:

Name <atom +>: Name of the socket that is given while establishing the connection with NXT® robot.
Port1 <integer +>: Name of port at which the first NXT® touch sensor is connected.
Port2 <integer +>: Name of port at which the second NXT® touch sensor is connected.
Response1 <integer ->: Value read by the first NXT® touch sensor.
Response2 <integer ->: Value read by the second NXT® touch sensor.

Example:

```
read_touch_nxt('NXT',1,2,R1,R2) .
```

This will display R1=1 or 0, R2=1 or 0. R1 and R2 are the value read by the touch sensors connected to port 1 and port 2 respectively.

1.5.5 Colour Sensor

1.5.5.1 init_Color_nxt(Name,Port)

Function: To initialize a HiTechnic (HT) colour sensor connected at a given port.

Description:

Name <atom +>: Name of the socket that is given while establishing the connection with NXT® robot.
Port <integer +>: Name of the sensor port at which the HT® colour sensor is connected
(Possible values are 1, 2, 3 and 4).

Example:

init_color_nxt(‘NXT’, 1) .

This will initialize the HT colour sensor connected at port 1.

1.5.5.2 init_color_nxt(Name,Port1,Port2)

Function: To initialize two HT® colour sensors connected at given ports.

Description:

Name <atom +>: Name of the socket that is given while establishing the connection with NXT® robot.
Port1 <integer +>: Name of the sensor port at which the first HT® colour sensor is connected.
Port2 <integer +>: Name of the sensor port at which the second HT® colour sensor is connected.

Example:

init_color_nxt(‘NXT’, 1,2) .

This will initialize the HT® colour sensors connected at port 1 and 2 of the NXT®.

1.5.5.3 init_color_nxt(Name,Port1,Port2,Port3)

Function: To initialize three HT® colour sensors connected to given ports.

Description:

Name <atom +>: Name of the socket that is given while establishing the connection with NXT robot.
Port 1 <Integer +>: Name of the sensor port at which the first colour sensor is connected.
Port 2 <Integer +>: Name of the sensor port at which the second colour sensor is connected.
Port 3 <Integer +>: Name of the sensor port at which the third colour sensor is connected.

Example:

init_touch_nxt(‘NXT’, 1,2,3) .

This will initialize the HT® colour sensors connected at ports 1, 2 and 3.

1.5.5.4 init_color_nxt(Name,Port1,Port2,Port3,Port4)

Function: To initialize four HT® colour sensors connected to given ports.

Description:

Name <atom +>: Name of the socket that is given while establishing the connection with NXT® robot.
Port1 <integer +>: Name of the sensor port at which the first HT® colour sensor is connected.
Port2 <integer +>: Name of the sensor port at which the second HT® colour sensor is connected.
Port3 <integer +>: Name of the sensor port at which the third HT® colour sensor is connected.

Port4 <integer +>: Name of the sensor port at which the fourth HT® colour sensor is connected.

Example:

init_color_nxt('NXT', 1, 2, 3, 4) .

This will initialize the HT® colour sensors connected at ports 1, 2, 3 and 4.

1.5.5.5 *read_color_nxt(Name,Port,Res)*

Function: To read the value of a colour sensor.

Description:

Name <atom +>: Name of the socket that is given while establishing the connection with NXT® robot.
Port <integer +>: Name of port at which the HT® colour sensor is connected.
Res <integer -> Value read by the HT® colour sensor.

Example:

read_color_nxt('NXT',1,R) .

This will display the value read by the HT® colour sensor connected to port 1.

1.5.5.6 *read_color_nxt(Name,Port1,Port2,Res1,Res2)*

Function: To read the values of two HT® colour sensors.

Description:

Name <atom +>: Name of the socket that is given while establishing the connection with NXT® robot.
Port1 <integer +>: Name of port at which the first HT® colour sensor is connected.
Port2 <integer +>: Name of port at which the second HT® colour sensor is connected.
Res1 <integer ->: Value read by the first HT® colour sensor.
Res2 <integer ->: Value read by the second HT® colour sensor.

Example:

read_color_nxt('NXT',1,2,R1,R2) .

This will display R1 and R2 - the value read by the colour sensors connected to port 1 and port 2 respectively.

1.5.6 Compass Sensor

1.5.6.1 *init_compass_nxt (Name,Port)*

Function: To initialize a HT compass sensor connected at a given port.

Description:

Name <atom +>: Name of the socket that is given while establishing the connection with NXT® robot.
Port <integer +>: Name of the sensor port at which the HT® compass sensor is connected (Possible values are 1, 2, 3 and 4).

Example:

init_compass_nxt('NXT', 1) .

This will initialize the HT® compass sensor connected at port 1.

1.5.6.2 *init_compass_nxt(Name,Port1,Port2)*

Function: To initialize two HT® compass sensors connected at given ports.

Description:

Name <atom +>: Name of the socket that is given while establishing the connection with NXT® robot.
Port1 <integer +>: Name of the sensor port at which the first HT® compass sensor is connected.
Port2 <integer +>: Name of the sensor port at which the second HT® compass sensor is connected.

Example:

init_compass_nxt('NXT', 1, 2) .

This will initialize the HT® compass sensors connected at port 1 and 2 of the NXT®.

1.5.6.3 *read_compass_nxt(Name,Port,Res)*

Function: To read the value of a HT compass sensor.

Description:

Name <atom +>: Name of the socket that is given while establishing the connection with NXT® robot.
Port <integer +>: Name of port at which the HT® compass sensor is connected.
Res <integer ->: Value read by the HT® compass sensor.

Example:

read_compass_nxt('NXT',1,R) .

This will display the value read by the HT® compass sensor connected to port 1.

1.5.6.4 *read_compass_nxt(Name,Port1,Port2,Res1,Res2)*

Function: To read the values of two HT® compass sensors.

Description:

Name <atom +>: Name of the socket that is given while establishing the connection with NXT® robot.
Port1 <integer +>: Name of port at which the first HT® compass sensor is connected.
Port2 <integer +>: Name of port at which the second HT® compass sensor is connected.
Res1 <integer ->: Value read by the first HT® compass sensor.
Res2 <integer ->: Value read by the second HT® compass sensor.

Example:

read_compass_nxt('NXT',1, 2, R1, R2) .

This will display R1 and R2 - the value read by the compass sensors connected to port 1 and port 2 respectively.

1.5.7 Gyro Sensor

1.5.7.1 *init_gyro_nxt (Name,Port)*

Function: To initialize a HT Gyro sensor connected at a given port.

Description:

Name <atom +>: Name of the socket that is given while establishing the connection with NXT® robot.

Port <integer +>: Name of the sensor port at which the HT® gyro sensor is connected (Possible values are 1, 2, 3 and 4).

Example:

init_gyro_nxt('NXT', 1) .

This will initialize the HT® gyro sensor connected at port 1.

1.5.7.2 init_gyro_nxt(Name,Port1,Port2)

Function: To initialize two HT® gyro sensors connected at given ports.

Description:

Name <atom +>: Name of the socket that is given while establishing the connection with NXT® robot.
Port1 <integer +>: Name of the sensor port at which the first HT® gyro sensor is connected.
Port2 <integer +>: Name of the sensor port at which the second HT® gyro sensor is connected.

Example:

init_gyro_nxt('NXT', 1, 2) .

This will initialize the HT® gyro sensors connected at port 1 and 2 of the NXT®.

1.5.7.3 read_gyro_nxt(Name,Port,Res)

Function: To read the value of a HT gyro sensor.

Description:

Name <atom +>: Name of the socket that is given while establishing the connection with NXT® robot.
Port <integer +>: Name of port at which the HT® gyro sensor is connected.
Res <integer ->: Value read by the HT® gyro sensor.

Example:

read_gyro_nxt('NXT',1,R) .

This will display the value read by the HT® gyro sensor connected to port 1.

1.5.7.4 read_gyro_nxt(Name,Port1,Port2,Res1,Res2)

Function: To read the values of two HT® gyro sensors.

Description:

Name <atom +>: Name of the socket that is given while establishing the connection with NXT® robot.
Port1 <integer +>: Name of port at which the first HT® gyro sensor is connected.
Port2 <integer +>: Name of port at which the second HT® gyro sensor is connected.
Res1 <integer ->: Value read by the first HT® gyro sensor.
Res1 <integer ->: Value read by the second HT® gyro sensor.

Example:

read_gyro_nxt('NXT', 1, 2, R1, R2) .

This will display R1 and R2 - the values read by the gyro sensors connected to port 1 and port 2 respectively.

1.5.8 IR Seeker

1.5.8.1 init_irseeker_nxt (Name,Port)

Function: To initialize a HT ir seeker connected at a given port.

Description:

Name <atom +>: Name of the socket that is given while establishing the connection with NXT® robot.
Port <integer +>: Name of the sensor port at which the HT® ir seeker is connected (Possible values are 1, 2, 3 and 4).

Example:

init_irseeker_nxt('NXT', 1) .

This will initialize the HT® ir seeker sensor connected at port 1.

1.5.8.2 init_irseeker_nxt(Name,Port1,Port2)

Function: To initialize two HT® IR seekers connected at given ports.

Description:

Name <atom +>: Name of the socket that is given while establishing the connection with NXT® robot.
Port1 <integer +>: Name of the sensor port at which the first HT® ir seeker is connected.
Port2 <integer +>: Name of the sensor port at which the second HT® ir seeker is connected.

Example:

init_irseeker_nxt('NXT', 1, 2) .

This will initialize the HT® ir seekers connected at port 1 and 2 of the NXT®.

1.5.8.3 read_irseeker_nxt(Name,Port,Res)

Function: To read the value of a HT ir seeker.

Description:

Name <atom +>: Name of the socket that is given while establishing the connection with NXT® robot.
Port <integer +>: Name of port at which the HT® ir seeker is connected.
Res <integer ->: Value read by the HT® ir seeker.

Example:

read_irseeker_nxt('NXT', 1, R) .

This will display the value read by the HT® ir seeker connected to port 1.

1.5.8.4 read_irseeker_nxt(Name,Port1,Port2,Res1,Res2)

Function: To read the values of two HT® ir seekers.

Description:

Name <atom +>: Name of the socket that is given while establishing the connection with NXT® robot.
Port1 <integer +>: Name of port at which the first HT® ir seeker is connected.
Port2 <integer +>: Name of port at which the second HT® ir seeker is connected.
Res1 <integer ->: Value read by the first HT® ir seeker.
Res1 <integer ->: Value read by the second HT® ir seeker.

Example:

read_gyro_nxt('NXT', 1, 2, R1, R2) .

This will display R1 and R2 - the values read by the ir seekers connected to port 1 and port 2 respectively.

APPENDIX I

Alternate Method for Initial Setup of the SWI-PRO-NXT

Step 1:

Download the "*nxtbin.class*" from the home directory of the LPA-PRO-NXT interface (C:\Tartarus) onto the NXT® brick.

a. Power on the NXT®.

b. Open console (command prompt) on your PC.

c. Change the directory to the home directory of the SWI-PRO-NXT interface (C:\Tartarus).

d. Check if the system's Bluetooth® is ON.

e. Execute the following command:

   nxj -b -n "name of your NXT® (default is NXT)" nxtbin

   You should hear a beep from the NXT® brick when this download is complete. Ensure that the file name in the command has only the primary name.

f. Execute the *nxtbin.nxj* file on the NXT®,

The NXT® brick will display **"Waiting…"** on the LCD screen.

Step 2:

Open another console on the same PC and change the directory to 'C:\Tartarus'. Execute the following command:

nxjpc –jar interface.jar 8688

(8688 is a port number. You may use other port number provided it is not already in use.)

The console will now display "**Waiting for connection…"**

Step 3:

Open SWI-PROLOG console and load the *nxt_interface.pl* using the predicate *consult/1* or the **consult** option in the **File** Menu. This file will be available in the local directory (C:\Tartarus) of the interface.

Upon successful loading of the file, execute the following predicate:

connect2_nxt(Name, Link, Port, Response).

Name <atom +>: Stands for the name of your NXT®. This will be the connection identifier for the whole NXT® session.
Link <atom +>: The host name or IP is to be typed here. If the same machine is used then use *localhost.*
Port <number +>: The port number is needed by the *relay_server*. It should be noted that the port number should be the same as that of the one used while executing the command in Step 2.
Response <atom ->: This is the **ACK** from the relay server. You do not have to provide a value to it. After the connection is established the **ACK** message '*started*' will be displayed.