```
------------------------------------------------------------------------------
--                                                                          --
--                            GNAT RAVENSCAR for NXT                        --
--                         Copyright (C) 2011, AdaCore                      --
--                                                                          --
------------------------------------------------------------------------------

--  Abstract state machine representing NXT ports using the I2C protocol

with Interfaces;  use Interfaces;
with System;

package NXT.I2C_Ports is
   pragma Elaborate_Body;

   procedure Initialize;
   --  Initialize this package's state.
   --  Called automatically during package body elaboration, but can be called
   --  manually as well.

   subtype IO_Modes is Unsigned_16;
   --  The modes used for I2C communication.  See the specific values below.
   --  Not an enumeration type since can be used as a set, ie with multiple
   --  bits set to enable more than one mode aspect at a time.

   Standard_Mode : constant IO_Modes := 0;
   LEGO_Mode     : constant IO_Modes := 1;
   Always_Active : constant IO_Modes := 2;
   No_Release    : constant IO_Modes := 4;

   procedure Configure_I2C_Port (Port : Sensor_Id;  Mode : IO_Modes);
   --  Enable I2C on Port using the I/O mode specified by Mode.
   --  Does all hardware configuration required.
   --  Only the mode values above are recognized. All others are silently
   --  ignored.

   function Enabled (Port : Sensor_Id) return Boolean;
   --  Returns whether the port identified by Port is enabled.
   --  Note that Set_IO_Mode enables the port.

```

```
41    procedure Disable (Port : Sensor_Id);
42    --  Disable I2C communications on port identified by Port. Resets that port.
43    --  No further I2C transactions on that port will be processed, all pending
44    --  transactions are lost.
45
46    procedure Disable_All_Ports;
47    --  Invokes Disable on each port
48
49    function Busy (Port : Sensor_Id) return Boolean;
50    --  Returns whether the I2C connection for port identified by Port is busy
51
52    type Transfer_Type is (Read, Write);
53    pragma Discard_Names (Transfer_Type);
54
55    procedure Start_Transaction
56      (Port                   : Sensor_Id;
57       Device_Address         : Unsigned_32;
58       Register_Address       : Unsigned_32;
59       Register_Address_Bytes : Unsigned_16;
60       Write_Buffer           : System.Address;
61       Bytes_To_Transfer      : Positive;
62       Operation              : Transfer_Type;
63       Result                 : out Integer);
64    --  Starts an I2C I/O transaction.
65    --  Port: the id of the sensor port to use for communication.
66    --  Device_Address: the address of the chip within the sensor.
67    --  Register_Address: the address of the internal register within the chip.
68    --  Register_Address_Bytes: the number of bytes in the register address.
69    --  Write_Buffer: the address of an array of bytes containing data to send.
70    --  Bytes_To_Transfer: the number of bytes to send.
71    --  Operation: indicates whether this is a read or write transaction.
72    --  Result is < 0 if there is an error.
73
74    procedure Complete_Transaction
75      (Port            : Sensor_Id;
76       Incoming_Buffer : System.Address;
77       Bytes_To_Read   : Positive;
78       Result          : out Integer);
79    --  Complete an I2C transaction and retrieve any data read.
80    --  Port: the id of the sensor port to use for communication.
```

```
81    --  Incoming_Buffer: address of the array of bytes to contain the data read.
82    --  Bytes_to_Read: then number of bytes expected to be received.
83    --  Result is < 0 if there is an error, otherwise is the number of bytes
84    --  transferred. Specifically, the error codes are as follows:
85    --  when the port is not enabled: -1
86    --  when Busy (Port) : -2
87    --  when the port has a fault: -3
88    --  when Bytes_To_Read > max buffer size (32): -4
89
90 end NXT.I2C_Ports;
91
```