package NXT.I2C_Ports is

  abstract state machine representing NXT ports using the I2C protocol

with Interfaces; use Interfaces;
with System;

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, i.e. 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.
procedure Disable (Port : Sensor_Id);
-- Disable I2C communications on port identified by Port. Resets that port.
-- No further I2C transactions on that port will be processed, all pending
-- transactions are lost.

procedure Disable_All_Ports;
-- Invokes Disable on each port

function Busy (Port : Sensor_Id) return Boolean;
-- Returns whether the I2C connection for port identified by Port is busy

type Transfer_Type is (Read, Write);
pragma Discard_Names (Transfer_Type);

procedure Start_Transaction
(Port : Sensor_Id;
Device_Address : Unsigned_32;
Register_Address : Unsigned_32;
Register_Address_Bytes : Unsigned_16;
Write_Buffer : System.Address;
Bytes_To_Transfer : Positive;
Operation : Transfer_Type;
Result : out Integer);
-- Starts an I2C I/O transaction.
-- Port: the id of the sensor port to use for communication.
-- Device_Address: the address of the chip within the sensor.
-- Register_Address: the address of the internal register within the chip.
-- Register_Address_Bytes: the number of bytes in the register address.
-- Write_Buffer: the address of an array of bytes containing data to send.
-- Bytes_To_Transfer: the number of bytes to send.
-- Operation: indicates whether this is a read or write transaction.
-- Result is < 0 if there is an error.

procedure Complete_Transaction
(Port : Sensor_Id;
Incoming_Buffer : System.Address;
Bytes_To_Read : Positive;
Result : out Integer);
-- Complete an I2C transaction and retrieve any data read.
-- Port: the id of the sensor port to use for communication.
-- Incoming_Buffer: address of the array of bytes to contain the data read.
-- Bytes_to_Read: then number of bytes expected to be received.
-- Result is < 0 if there is an error, otherwise is the number of bytes
-- transferred. Specifically, the error codes are as follows:
-- when the port is not enabled: -1
-- when Busy (Port): -2
-- when the port has a fault: -3
-- when Bytes_To_Read > max buffer size (32): -4

end NXT.I2C_Ports;