with Interfaces; use Interfaces;
with System;

package NXT.I2C_Sensors is
  pragma Elaborate_Body;

type I2C_Sensor is abstract tagged limited private;
  -- The I2C_Sensor abstract data type is the root for all I2C sensors types,
  -- such as the Lego ultrasonic sensor, as well as all third-party sensors.
  -- I2C_Sensor objects contain a port identifier and a device address. The
  -- port is the id for the sensor hardware port used by the sensor object.
  -- The device address is the address of the chip within the sensor to
  -- communicate with using the I2C protocol.

procedure Set_Port_Id (This : in out I2C_Sensor; To : Sensor_Id);
  -- Sets the port id for for This sensor. The software will then interact
  -- with the sensor using that physical port.

procedure Set_Device_Address (This : in out I2C_Sensor; To : Unsigned_32);
  -- Set the device address for the chip being communicated with.
  -- If not called, the address will have the value 2.

function Port (This : I2C_Sensor) return Sensor_Id;
  -- Returns the port id for This sensor

function Device (This : I2C_Sensor) return Unsigned_32;
-- Returns the device address for This sensor's chip

procedure Send_Data
  (This   : in out I2C_Sensor;
   Register : Unsigned_32;
   Buffer   : System.Address;
   Length   : Positive;
   Result   : out Integer);
-- Executes an I2C write transaction.
-- Register: the internal address of the target register within the chip
-- Buffer: the address of a byte buffer containing the data to send
-- Length: the length in bytes of the data to send
-- Result: 0 for success, negative otherwise

procedure Get_Data
  (This   : in out I2C_Sensor;
   Register : Unsigned_32;
   Buffer   : System.Address;
   Length   : Positive;
   Result   : out Integer);
-- Executes an I2C read transaction and waits for the result.
-- Register: the internal address of the target register within the chip
-- Buffer: the address of a byte buffer to contain the returning data
-- Length: the expected length in bytes of the return data
-- Result: 0 for success, negative otherwise

subtype Query_Reply is String (1 .. 8);

procedure Get_Sensor_Type
  (This   : in out I2C_Sensor;
   Result : out Query_Reply;
   Success : out Boolean);
-- Returns the sensor type, e.g., "Sonar"

procedure Get_Product_Id
  (This   : in out I2C_Sensor;
   Result : out Query_Reply;
   Success : out Boolean);
-- Returns the sensor product identifier, e.g., "LEGO"
procedure Get_Version
  (This : in out I2C_Sensor;
   Result : out Query_Reply;
   Success : out Boolean);
-- Returns the sensor version number

procedure Fetch_String
  (This : in out I2C_Sensor;
   Register : Unsigned_32;
   Length : Positive;
   Fetched : out Query_Reply;
   Success : out Boolean);
-- Get a string from the specified register address within the device
-- associated with This sensor

type Multiple_Bytes is array (Positive range <>) of Unsigned_8;

procedure Fetch_Multiple_Bytes
  (This : in out I2C_Sensor;
   Base_Register : Unsigned_32;
   Output : out Multiple_Bytes;
   Success : out Boolean);
-- Get Output'Length bytes of data, one byte each, from the registers
-- having contiguous address values starting at the address specified by
-- Base_Register.

procedure Set_Multiple_Bytes
  (This : in out I2C_Sensor;
   Base_Register : Unsigned_32;
   Input : Multiple_Bytes;
   Success : out Boolean);
-- Set Input'Length bytes of data, one byte each, into the registers having
-- contiguous address values starting at the address specified by
-- Base_Register.

private

type I2C_Sensor is abstract tagged limited
  record
    Port_Id : Sensor_Id;
Device_Address : Unsigned_32 := 2; -- by default
end record;

Version_Register   : constant := 16#00#;
Sensor_Type_Register : constant := 16#10#;
Product_Id_Register    : constant := 16#08#;

end NXT.I2C_Sensors;