-- Abstract data type representing the Lego ultrasonic sensor

pragma Restrictions (No_Streams);

with NXT.I2C_Sensors; use NXT.I2C_Sensors;
with Interfaces; use Interfaces;
with Ada.Real_Time; use Ada.Real_Time;
with System;

package NXT.Ultrasonic_Sensors is
  pragma Elaborate_Body;

type Ultrasonic_Sensor (<>) is new I2C_Sensor with private;
  -- An abstract data type representing the Lego ultrasonic sensor.
  -- By making the type indefinite, we force an initialization when objects
  -- are declared, thereby ensuring a call to a constructor function

type Operating_Mode is (Off, Ping, Continuous, Capture, Reset, Error);
for Operating_Mode use
  (Off      => 0,
   Ping     => 1,
   Continuous => 2,
   Capture  => 3,
   Reset    => 4,
   Error    => 5);
  -- Confirming (matches hardware codes). NB: order is critical!
pragma Discard_Names (Operating_Mode);

procedure Query_Current_Mode
  (This : in out Ultrasonic_Sensor;
   Mode : out Operating_Mode);
  -- Actually ask the hardware for the current mode
subtype Commanded_Operating_Mode is Operating_Mode range Off .. Reset;

procedure Set_Mode
    (This : in out Ultrasonic_Sensor;
     Mode : Commanded_Operating_Mode);
-- Set This sensor into the specified mode. The physical device is directly
-- affected. For example, a value of Reset will physically reset the
-- sensor, after which the sensor will be in the "off" mode. A mode value
-- of Ping puts the device into that mode and issues a single ping, and so
-- on.

function Commanded_Mode (This : Ultrasonic_Sensor) return Operating_Mode;
-- Returns the mode commanded via Set_Mode

procedure Reset (This : in out Ultrasonic_Sensor);
-- Same as Set_Mode (Reset) for convenience

procedure Ping (This : in out Ultrasonic_Sensor);
-- Issues a single ping from the device and computes a distance value.
-- Same as Set_Mode (Ping) for convenience.

procedure Off (This : in out Ultrasonic_Sensor);
-- Same as Set_Mode (Off) for convenience

procedure Get_Distance
    (This : in out Ultrasonic_Sensor;
     Reading : out Natural);
-- Gets the current computed distance into Reading. If no object is in
-- range or if an error occurred, the value is 255.

subtype Distances_Index is Integer range 1 .. 8;
type Distances is array (Distances_Index range <>) of Unsigned_8;

procedure Get_Distances
    (This : in out Ultrasonic_Sensor;
     Readings : out Distances;
     Actual : out Natural);
-- A convenience definition, equivalent to calling:
-- Get_Distances (This, Readings'Length, 0, Readings, Actual);
procedure Get_Distances
  (This : in out Ultrasonic_Sensor;
   Requested : Distances_Index;
   Offset : Natural;
   Readings : out Distances;
   Actual : out Natural);
-- Returns an array of distances, depending on the number of objects
-- detected within the range of the sensor. In continuous mode, at most one
-- distance is returned. In ping mode, up to 8 distances are returned, but
-- not more than Requested. If the distance data is not yet available the
-- method will wait for it.
-- Requested: the number of distance readings to return.
-- Offset: the offset within Readings at which new distance values should
-- start being placed.
-- Readings: the object containing the new distances returned.
-- Actual: the number of objects detected and thus the number of distances
-- assigned in Readings. Will be zero when no object is detected within
-- range.

Operating_Error : exception;
-- Raised when trying something impossible, like getting a distance when
-- the device is off

subtype Units_String is String (1 .. 8);

procedure Get_Units
  (This : in out Ultrasonic_Sensor;
   Units : out Units_String;
   Success : out Boolean);
-- Gets a string indicating the type of units in use by the sensor. The
-- default response is 10E-2m indicating use of centimeters.

subtype Data is Multiple_Bytes (1 .. 3);

procedure Get_Factory_Data
  (This : in out Ultrasonic_Sensor;
   Info : out Data;
   Success : out Boolean);
-- Gets factory calibration settings. The three bytes are as follows:
-- Info (1): always zero
procedure Get_Calibration_Data
  (This : in out Ultrasonic_Sensor;
   Info : out Data;
   Success : out Boolean);
-- Gets current calibration data. The three bytes are as follows:
-- Info (1): always zero
-- Info (2): the current scale factor
-- Info (3): the current scale divisor

procedure Set_Calibration_Data
  (This : in out Ultrasonic_Sensor;
   Info : Data;
   Success : out Boolean);
-- Sets calibration data. The three bytes are as follows:
-- Info (1): always zero
-- Info (2): the intended scale factor
-- Info (3): the intended scale divisor

procedure Get_Continuous_Interval
  (This : in out Ultrasonic_Sensor;
   Info : out Unsigned_8;
   Success : out Boolean);
-- Gets the scan interval used in continuous mode

procedure Set_Continuous_Interval
  (This : in out Ultrasonic_Sensor;
   Info : Unsigned_8;
   Success : out Boolean);
-- Sets the scan interval to be used in continuous mode

private

-- The Lego ultrasonic sensor uses a "bit-banged" I2C interface and seems
-- to require a minimum delay between commands, otherwise the commands
-- fail.
Command_Delay : constant Time_Span := Milliseconds (5);
type Ultrasonic_Sensor is new I2C_Sensor with
  record
    Next_Command_Time : Time := Clock + Command_Delay;
    Data_Available_Time : Time;
    Mode : Operating_Mode;
  end record;

overriding
procedure Send_Data
  (This : in out Ultrasonic_Sensor;
   Register : Unsigned_32;
   Buffer : System.Address;
   Length : Positive;
   Result : out Integer);
-- Override the standard version to ensure correct timing when using the
-- ultrasonic sensor. The Lego ultrasonic sensor uses a "bit-banged" I2C
-- interface and seems to require a minimum delay between commands.

overriding
procedure Get_Data
  (This : in out Ultrasonic_Sensor;
   Register : Unsigned_32;
   Buffer : System.Address;
   Length : Positive;
   Result : out Integer);
-- Override the standard version to ensure correct timing when using the
-- ultrasonic sensor. The Lego ultrasonic sensor uses a "bit-banged" I2C
-- interface and seems to require a minimum delay between commands.

-- The following are the addresses for the registers within the Lego
-- ultrasonic sensor chip
Factory_Data_Register : constant := 16#11#;
Units_Register : constant := 16#14#;
Continuous_Interval_Register : constant := 16#40#;
Mode_Register : constant := 16#41#;
Distance_Register : constant := 16#42#;
Calibration_Register : constant := 16#4A#;
-- The delay values corresponding to when the data become available in the
-- given mode.
Delay_Data_Ping : constant Time_Span := Milliseconds (50);
Delay_Data_Other : constant Time_Span := Milliseconds (30);

Default_Distance : constant := 255;
-- The value returned when no actual value is available. Also represents
-- "out of range".

procedure Initialize_Port
   (This : in out Ultrasonic_Sensor;
    Port : Sensor_Id);
   -- Sets Port Id for This sensor. Sets power and pins appropriately. Enables
   -- "Lego Mode".
   -- Intended to be called by constructors.
end NXT.Ultrasonic_Sensors;