

```
1 -----
2 --
3 --          GNAT RAVENSCAR for NXT          --
4 --          Copyright (C) 2011, AdaCore     --
5 --
6 -----
7
8 -- Abstract data type representing the Lego ultrasonic sensor
9
10 pragma Restrictions (No_Streams);
11
12 with NXT.I2C_Sensors; use NXT.I2C_Sensors;
13 with Interfaces;      use Interfaces;
14 with Ada.Real_Time;  use Ada.Real_Time;
15 with System;
16
17 package NXT.Ultrasonic_Sensors is
18     pragma Elaborate_Body;
19
20     type Ultrasonic_Sensor (<>) is new I2C_Sensor with private;
21     -- An abstract data type representing the Lego ultrasonic sensor.
22     -- By making the type indefinite, we force an initialization when objects
23     -- are declared, thereby ensuring a call to a constructor function
24
25     type Operating_Mode is (Off, Ping, Continuous, Capture, Reset, Error);
26     for Operating_Mode use
27         (Off          => 0,
28          Ping         => 1,
29          Continuous   => 2,
30          Capture      => 3,
31          Reset        => 4,
32          Error        => 5);
33     -- Confirming (matches hardware codes). NB: order is critical!
34     pragma Discard_Names (Operating_Mode);
35
36     procedure Query_Current_Mode
37         (This : in out Ultrasonic_Sensor;
38          Mode : out Operating_Mode);
39     -- Actually ask the hardware for the current mode
40
```

```
41  subtype Commanded_Operating_Mode is Operating_Mode range Off .. Reset;
42
43  procedure Set_Mode
44      (This : in out Ultrasonic_Sensor;
45       Mode : Commanded_Operating_Mode);
46  -- Set This sensor into the specified mode. The physical device is directly
47  -- affected. For example, a value of Reset will physically reset the
48  -- sensor, after which the sensor will be in the "off" mode. A mode value
49  -- of Ping puts the device into that mode and issues a single ping, and so
50  -- on.
51
52  function Commanded_Mode (This : Ultrasonic_Sensor) return Operating_Mode;
53  -- Returns the mode commanded via Set_Mode
54
55  procedure Reset (This : in out Ultrasonic_Sensor);
56  -- Same as Set_Mode (Reset) for convenience
57
58  procedure Ping (This : in out Ultrasonic_Sensor);
59  -- Issues a single ping from the device and computes a distance value.
60  -- Same as Set_Mode (Ping) for convenience.
61
62  procedure Off (This : in out Ultrasonic_Sensor);
63  -- Same as Set_Mode (Off) for convenience
64
65  procedure Get_Distance
66      (This      : in out Ultrasonic_Sensor;
67       Reading  : out Natural);
68  -- Gets the current computed distance into Reading. If no object is in
69  -- range or if an error occurred, the value is 255.
70
71  subtype Distances_Index is Integer range 1 .. 8;
72  type Distances is array (Distances_Index range <>) of Unsigned_8;
73
74  procedure Get_Distances
75      (This      : in out Ultrasonic_Sensor;
76       Readings  : out Distances;
77       Actual    : out Natural);
78  -- A convenience definition, equivalent to calling:
79  -- Get_Distances (This, Readings'Length, 0, Readings, Actual);
80
```

```
81   procedure Get_Distances
82     (This      : in out Ultrasonic_Sensor;
83      Requested : Distances_Index;
84      Offset    : Natural;
85      Readings  : out Distances;
86      Actual    : out Natural);
87   -- Returns an array of distances, depending on the number of objects
88   -- detected within the range of the sensor. In continuous mode, at most one
89   -- distance is returned. In ping mode, up to 8 distances are returned, but
90   -- not more than Requested. If the distance data is not yet available the
91   -- method will wait for it.
92   -- Requested: the number of distance readings to return.
93   -- Offset: the offset within Readings at which new distance values should
94   -- start being placed.
95   -- Readings: the object containing the new distances returned.
96   -- Actual: the number of objects detected and thus the number of distances
97   -- assigned in Readings. Will be zero when no object is detected within
98   -- range.
99
100  Operating_Error : exception;
101  -- Raised when trying something impossible, like getting a distance when
102  -- the device is off
103
104  subtype Units_String is String (1 .. 8);
105
106  procedure Get_Units
107    (This      : in out Ultrasonic_Sensor;
108     Units     : out Units_String;
109     Success   : out Boolean);
110  -- Gets a string indicating the type of units in use by the sensor. The
111  -- default response is 10E-2m indicating use of centimeters.
112
113  subtype Data is Multiple_Bytes (1 .. 3);
114
115  procedure Get_Factory_Data
116    (This : in out Ultrasonic_Sensor;
117     Info  : out Data;
118     Success : out Boolean);
119  -- Gets factory calibration settings. The three bytes are as follows:
120  -- Info (1): always zero
```

```
121  -- Info (2): the current scale factor
122  -- Info (3): the current scale divisor
123
124  procedure Get_Calibration_Data
125    (This      : in out Ultrasonic_Sensor;
126     Info      : out Data;
127     Success   : out Boolean);
128  -- Gets current calibration data. The three bytes are as follows:
129  -- Info (1): always zero
130  -- Info (2): the current scale factor
131  -- Info (3): the current scale divisor
132
133  procedure Set_Calibration_Data
134    (This      : in out Ultrasonic_Sensor;
135     Info      : Data;
136     Success   : out Boolean);
137  -- Sets calibration data. The three bytes are as follows:
138  -- Info (1): always zero
139  -- Info (2): the intended scale factor
140  -- Info (3): the intended scale divisor
141
142  procedure Get_Continuous_Interval
143    (This      : in out Ultrasonic_Sensor;
144     Info      : out Unsigned_8;
145     Success   : out Boolean);
146  -- Gets the scan interval used in continuous mode
147
148  procedure Set_Continuous_Interval
149    (This      : in out Ultrasonic_Sensor;
150     Info      : Unsigned_8;
151     Success   : out Boolean);
152  -- Sets the scan interval to be used in continuous mode
153
154  private
155
156  -- The Lego ultrasonic sensor uses a "bit-banged" I2C interface and seems
157  -- to require a minimum delay between commands, otherwise the commands
158  -- fail.
159  Command_Delay : constant Time_Span := Milliseconds (5);
160
```

```
161     type Ultrasonic_Sensor is new I2C_Sensor with
162         record
163             Next_Command_Time    : Time := Clock + Command_Delay;
164             Data_Available_Time  : Time;
165             Mode                  : Operating_Mode;
166         end record;
167
168     overriding
169     procedure Send_Data
170         (This      : in out Ultrasonic_Sensor;
171          Register  : Unsigned_32;
172          Buffer     : System.Address;
173          Length    : Positive;
174          Result    : out Integer);
175     -- Override the standard version to ensure correct timing when using the
176     -- ultrasonic sensor. The Lego ultrasonic sensor uses a "bit-banged" I2C
177     -- interface and seems to require a minimum delay between commands.
178
179     overriding
180     procedure Get_Data
181         (This      : in out Ultrasonic_Sensor;
182          Register  : Unsigned_32;
183          Buffer     : System.Address;
184          Length    : Positive;
185          Result    : out Integer);
186     -- Override the standard version to ensure correct timing when using the
187     -- ultrasonic sensor. The Lego ultrasonic sensor uses a "bit-banged" I2C
188     -- interface and seems to require a minimum delay between commands.
189
190     -- The following are the addresses for the registers within the Lego
191     -- ultrasonic sensor chip
192     Factory_Data_Register      : constant := 16#11#;
193     Units_Register             : constant := 16#14#;
194     Continuous_Interval_Register : constant := 16#40#;
195     Mode_Register              : constant := 16#41#;
196     Distance_Register          : constant := 16#42#;
197     Calibration_Register       : constant := 16#4A#;
198
199     -- The delay values corresponding to when the data become available in the
200     -- given mode.
```

```
201   Delay_Data_Ping   : constant Time_Span := Milliseconds (50);
202   Delay_Data_Other : constant Time_Span := Milliseconds (30);
203
204   Default_Distance : constant := 255;
205   -- The value returned when no actual value is available. Also represents
206   -- "out of range".
207
208   procedure Initialize_Port
209     (This : in out Ultrasonic_Sensor;
210      Port : Sensor_Id);
211   -- Sets Port Id for This sensor. Sets power and pins appropriately. Enables
212   -- "Lego Mode".
213   -- Intended to be called by constructors.
214
215 end NXT.Ultrasonic_Sensors;
216
```