

```
1 -----
2 --
3 --          GNAT RAVENSCAR for NXT          --
4 --          Copyright (C) 2011, AdaCore     --
5 --
6 -----
7
8 -- Low level controls for sensor port hardware
9
10 with Interfaces;      use Interfaces;
11 with NXT.Registers;  use NXT.Registers;
12
13 package NXT.Sensor_Ports is
14   pragma Elaborate_Body;
15
16   procedure Reset (Port : Sensor_Id);
17
18   procedure Reset_All_Ports;
19
20   type Pin_Id is (Digital_0, Digital_1);
21   pragma Discard_Names (Pin_Id);
22
23   type Modes is (Off, Input, Output, ADC);
24   pragma Discard_Names (Modes);
25
26   procedure Set_Pin_Mode (Port : Sensor_Id; Pin : Pin_Id; Mode : Modes);
27
28   type Pin_States is (Low, High);
29   pragma Discard_Names (Pin_States);
30
31   function Current_State (Port : Sensor_Id; Pin : Pin_Id) return Pin_States;
32
33   procedure Set_Pin_State
34     (Port  : Sensor_Id;
35      Pin   : Pin_Id;
36      Value : Pin_States);
37
38   procedure Set_Input_Power (Port : Sensor_Id; Kind : Sensor_Power);
39
40   type Digital_Pins is array (Pin_Id) of Unsigned_32;
```

```
41
42     type Port_Pins is
43         record
44             Pins                : Digital_Pins;
45             ADC_Channel_Number : Unsigned_32;
46             ADC_Data_Reg_Number : Unsigned_32;
47         end record;
48
49     Sensor_Pins : constant array (Sensor_Id) of Port_Pins :=
50         (((PIO_PA23, PIO_PA18), ADC_CH1, ADC_CDR1),
51          ((PIO_PA28, PIO_PA19), ADC_CH2, ADC_CDR2),
52          ((PIO_PA29, PIO_PA20), ADC_CH3, ADC_CDR3),
53          ((PIO_PA30, PIO_PA2),  ADC_CH7, ADC_CDR7));
54
55 end NXT.Sensor_Ports;
56
```