Remote I/O Package Specifications

Introduction

This document contains selected package specifications from **`libsimpleio/ada/`** that are pertinent to the Ada-Europe 2019 Remote I/O Tutorial.

Table of Contents

IO_Interfaces

This generic package must be instantiated with some type **`Property`**, which can be scalar or composite. The instantiated package will define three abstract interface types: **`InputInterface`**, **`InputOutputInterface`**, and **`OutputInterface`**. Each abstract interface includes a classwide access type and **`Get`** and/or **`Put`** procedures. **`IO_Interfaces`** is used extensively internally within **`libsimpleio`** but will seldom if ever be need to be referenced from an application program.

```
GENERIC

  TYPE Property IS PRIVATE;

PACKAGE IO_Interfaces IS

  -- Define an abstract input only interface

  TYPE InputInterface IS INTERFACE;

  -- Define an access type compatible with any subclass implementing
  -- InputInterface

  TYPE Input IS ACCESS ALL InputInterface'Class;

  -- Define a method for reading from an input

  FUNCTION Get(Self : IN OUT InputInterface) RETURN Property IS ABSTRACT;

  ------------------------------------------------------------------------

  -- Define an abstract input/output interface

  TYPE InputOutputInterface IS INTERFACE;

  -- Define an access type compatible with any subclass implementing
  -- InputOutputInterface

  TYPE InputOutput IS ACCESS ALL InputOutputInterface'Class;

  -- Define a method for reading from an input

  FUNCTION Get(Self : IN OUT InputOutputInterface) RETURN Property IS ABSTRACT;

  -- Define a method for writing to an output

  PROCEDURE Put(Self : IN OUT InputOutputInterface; value : Property) IS ABSTRACT;
```

```
   ------------------------------------------------------------------------

   -- Define an abstract output only interface

   TYPE OutputInterface IS INTERFACE;

   -- Define an access type compatible with any subclass implementing
   -- OutputInterface

   TYPE Output IS ACCESS ALL OutputInterface'Class;

   -- Define a method for writing to an output

   PROCEDURE Put(Self : IN OUT OutputInterface; value : Property) IS ABSTRACT;

END IO_Interfaces;
```

GPIO

This package defines an abstract interface for all GPIO (General Purpose Input/Output) pins. It defines an exception **`GPIO_Error`**, an abstract interface type **`PinInterface`**, and a classwide access type **`Pin`**.

```
WITH Ada.Text_IO;
WITH IO_Interfaces;

PACKAGE GPIO IS

  -- Define an exception for GPIO errors

  GPIO_Error : EXCEPTION;

  -- Instantiate text I/O package

  PACKAGE Boolean_IO IS NEW Ada.Text_IO.Enumeration_IO(Boolean);

  -- Type definitions

  TYPE Direction IS (Input, Output);

  -- Instantiate I/O interfaces package for digital I/O

  PACKAGE Interfaces IS NEW IO_Interfaces(Boolean);

  -- Define an abstract interface for GPIO pins, derived from
  -- Interfaces.InputOutputInterface

  TYPE PinInterface IS INTERFACE AND Interfaces.InputOutputInterface;

  -- Define an access type compatible with any subclass implementing
  -- PinInterface

  TYPE Pin IS ACCESS ALL PinInterface'Class;

END GPIO;
```

GPIO.RemoteIO

This package provides GPIO pin services using the Remote I/O protocol. It defines a concrete subclass of **`GPIO.PinInterface`** called **`GPIO.RemoteIO.PinSubclass`**.

Note that the **`Create`** function returns a value of classwide access type **`GPIO.Pin`** and is ***not*** a primitive operation of **`GPIO.RemoteIO.PinSubclass`**. This is a pattern followed throughout **`libsimpleio`**.

```
WITH RemoteIO.Client;

PACKAGE GPIO.RemoteIO IS

  TYPE PinSubclass IS NEW PinInterface WITH PRIVATE;

  -- GPIO pin object constructor

  FUNCTION Create
   (dev   : Standard.RemoteIO.Client.Device;
    num   : Standard.RemoteIO.ChannelNumber;
    dir   : Direction;
    state : Boolean := False) RETURN Pin;

  -- Read GPIO pin state

  FUNCTION Get(Self : IN OUT PinSubclass) RETURN Boolean;

  -- Write GPIO pin state

  PROCEDURE Put(Self : IN OUT PinSubclass; state : Boolean);

PRIVATE
  -- Implementation defined
END GPIO.RemoteIO;
```

Analog

This package defines abstract interfaces for analog sampled data inputs, outputs, and input/outputs. Use **`InputInterface`** and **`Input`** for ADC (Analog to Digital Converter) inputs and **`OutputInterface`** and **`Output`** for DAC (Digital to Analog Converter) outputs.

`InputOutputInterface` and **`InputOutput`**) are provided for completeness. They might be useful for a DAC with readback capability, or for unusual devices that are configurable as either analog input or output.

Sampled analog data values (of type **`Sample`**) are 32-bit unsigned and right justified.

```
WITH Ada.Text_IO;
WITH IO_Interfaces;

PACKAGE Analog IS

  -- Define a type for sampled analog data

  MaxResolution : CONSTANT := 32; -- Bits

  TYPE Sample IS MOD 2**MaxResolution;

  -- Instantiate text I/O package

  PACKAGE Sample_IO IS NEW Ada.Text_IO.Modular_IO(Sample);

  -- Instantiate I/O interfaces package for digital I/O

  PACKAGE Interfaces IS NEW IO_Interfaces(Sample);

  -- Interfaces

  TYPE InputInterface IS INTERFACE AND Interfaces.InputInterface;

  TYPE OutputInterface IS INTERFACE AND Interfaces.OutputInterface;

  TYPE InputOutputInterface IS INTERFACE AND Interfaces.InputOutputInterface;

  -- Access types

  TYPE Input IS ACCESS ALL InputInterface'Class;

  TYPE Output IS ACCESS ALL OutputInterface'Class;

  TYPE InputOutput IS ACCESS ALL InputOutputInterface'Class;

  -- Additional methods

  FUNCTION GetResolution(Self : IN OUT InputInterface) RETURN Positive IS ABSTRACT;

  FUNCTION GetResolution(Self : IN OUT InputOutputInterface) RETURN Positive IS
    ABSTRACT;

  FUNCTION GetResolution(Self : IN OUT OutputInterface) RETURN Positive IS ABSTRACT;

END Analog;
```

Voltage

This package defines a type **`Volts`** to represent continuously variable voltage measurements. It is representative of all of the physical quantity packages.

```
WITH Ada.Text_IO;
WITH IO_Interfaces;

PACKAGE Voltage IS

  TYPE Volts IS NEW Float;

  -- Instantiate text I/O package

  PACKAGE Volts_IO IS NEW Ada.Text_IO.Float_IO(Volts);

  -- Instantiate abstract interfaces package

  PACKAGE Interfaces IS NEW IO_Interfaces(Volts);

END Voltage;
```

ADC

This package provides services for reading the scaled input voltage from ADC (Analog to Digital Converter) inputs. It defines a concrete subclass of **`Volts.Interfaces.InputInterface`** called **`ADC.InputSubclass`**.

The **`Create`** function accepts an analog input object instance (of type **`Analog.Input`**), a reference voltage value (of type **`Voltage.Volts`**), and a voltage gain value (also of type **`Voltage.Volts`**) and returns a voltage input object instance of type **`Voltage.Interfaces.Input`**.

As is usual throughout **`libsimpleio`**, **`Create`** is **not** a primitive operation of **`ADC.InputSubclass`**.

```
WITH Analog;
WITH Voltage;

PACKAGE ADC IS

  ADC_Error : EXCEPTION;

  TYPE InputSubclass IS NEW Voltage.Interfaces.InputInterface WITH PRIVATE;

  -- Constructor

  FUNCTION Create
   (input     : Analog.Input;
    reference : Voltage.Volts;
    gain      : Voltage.Volts := 1.0) RETURN Voltage.Interfaces.Input;

  -- Methods

  FUNCTION Get(Self : IN OUT InputSubclass) RETURN Voltage.Volts;

PRIVATE
  -- Implementation defined
END ADC;
```

ADC.RemoteIO

This package provides analog input services using the Remote I/O protocol. It defines a concrete subclass of **`Analog.InputInterface`** called **`ADC.RemoteIO.InputSubclass`**.

The **`Create`** function returns an analog input object instance of type **`Analog.Input`**.

```
WITH Analog;
WITH RemoteIO.Client;

PACKAGE ADC.RemoteIO IS

  TYPE InputSubclass IS NEW Analog.InputInterface WITH PRIVATE;

  -- A/D input pin object constructor

  FUNCTION Create
   (dev : Standard.RemoteIO.Client.Device;
    num : Standard.RemoteIO.ChannelNumber) RETURN Analog.Input;

  -- Read A/D input pin

  FUNCTION Get(Self : IN OUT InputSubclass) RETURN Analog.Sample;

  -- Retrieve A/D input resolution

  FUNCTION GetResolution(Self : IN OUT InputSubclass) RETURN Positive;

PRIVATE
  -- Implementation defined
END ADC.RemoteIO;
```