

Remote I/O Package Specifications

IO_Interfaces

This generic package must be instantiated with some type **Property**, which can be scalar or composite. The instantiated package will define three interface types: **InputInterface**, **InputOutputInterface**, and **OutputInterface**. Each interface includes a classwide access type and **Get** and/or **Put** procedures

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

```
WITH IO_Interfaces;

PACKAGE GPIO IS

    -- Define an exception for GPIO errors

    GPIO_Error : EXCEPTION;

    -- 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 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`.

```
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;
```