RNS C# Port

0.0.1pa

Generated by Doxygen 1.9.6

Chapter 1

Hierarchical Index

1.1 Class Hierarchy

This inheritance list is sorted roughly, but not completely, alphabetically:

Chapter 2

Class Index

2.1 Class List

Here are the classes, structs, unions and interfaces with brief descriptions:

Chapter 3

Namespace Documentation

3.1 RNS Namespace Reference

Namespaces

- namespace **Interfaces**

Classes

- class **Interface**

  Initial Interface class.

3.2 RNS.Interfaces Namespace Reference

Classes

- class **RNodeInterface**

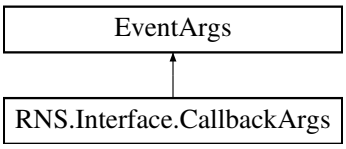
Chapter 4

Class Documentation

4.1 RNS.Interface.CallbackArgs Class Reference

Arguments for callback event handler.

Inheritance diagram for RNS.Interface.CallbackArgs:



Public Member Functions

- **CallbackArgs** (byte[ ] _message, **RNS.Interface** _interface)

Properties

- byte[ ] **Message** `[get]`
- **RNS.Interface Interface** `[get]`

4.1.1 Detailed Description

Arguments for callback event handler.

Definition at line **80** of file **Interface.cs**.

4.1.2 Constructor & Destructor Documentation

4.1.2.1 CallbackArgs()

```
RNS.Interface.CallbackArgs.CallbackArgs (
            byte[] _message,
             RNS::Interface _interface )  [inline]
```

Definition at line **84** of file **Interface.cs**.

4.1.3 Property Documentation

4.1.3.1 Interface

```
RNS.Interface RNS.Interface.CallbackArgs.Interface  [get]
```

Definition at line **83** of file **Interface.cs**.

4.1.3.2 Message

```
byte [] RNS.Interface.CallbackArgs.Message  [get]
```

Definition at line **82** of file **Interface.cs**.

4.2 RNS.Interface.CallbackClass Class Reference

Class for callback event handler.

Public Member Functions

- void **Process_Inbound** (byte[] _message, **RNS.Interface** _interface)

Protected Member Functions

- virtual void **OnCallback** (**CallbackArgs** e)

Events

- EventHandler< **CallbackArgs** >? **CallbackEventHandler**

4.2.1 Detailed Description

Class for callback event handler.

Definition at line **94** of file **Interface.cs**.

4.2.2 Member Function Documentation

4.2.2.1 OnCallback()

```
virtual void RNS.Interface.CallbackClass.OnCallback (
              CallbackArgs e )  [inline], [protected], [virtual]
```

Definition at line **102** of file **Interface.cs**.

4.2.2.2 Process_Inbound()

```
void RNS.Interface.CallbackClass.Process_Inbound (
             byte[] _message,
              RNS::Interface _interface )  [inline]
```

Definition at line **98** of file **Interface.cs**.

4.2.3 Event Documentation

4.2.3.1 CallbackEventHandler

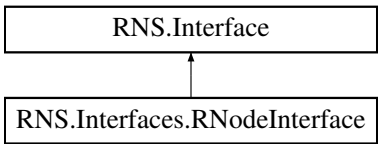
```
EventHandler< CallbackArgs>?  RNS.Interface.CallbackClass.CallbackEventHandler
```

Definition at line **97** of file **Interface.cs**.

4.3 RNS.Interface Class Reference

Initial Interface class.

Inheritance diagram for RNS.Interface:



Classes

- class **CallbackArgs**

  Arguments for callback event handler.
- class **CallbackClass**

  Class for callback event handler.

Public Member Functions

- **Interface** ()

  Basic initialization.

Public Attributes

- bool **IN** = false
- bool **OUT** = false
- bool **FWD** = false
- bool **RPT** = false
- string **name** = ""
- **CallbackClass Callbacks**
- int **ifac_size**

4.3.1 Detailed Description

Initial Interface class.

Definition at line **31** of file **Interface.cs**.

4.3.2 Constructor & Destructor Documentation

4.3.2.1 Interface()

```
RNS.Interface.Interface ( )  [inline]
```

Basic initialization.

Definition at line **65** of file **Interface.cs**.

4.3.3 Member Data Documentation

4.3.3.1 Callbacks

```
CallbackClass RNS.Interface.Callbacks
```

Definition at line **56** of file **Interface.cs**.

4.3.3.2 FWD

```
bool RNS.Interface.FWD = false
```

Definition at line **35** of file **Interface.cs**.

4.3.3.3 ifac_size

```
int RNS.Interface.ifac_size
```

Definition at line **59** of file **Interface.cs**.

4.3.3.4 IN

```
bool RNS.Interface.IN = false
```

Definition at line **33** of file **Interface.cs**.

4.3.3.5 name

```
string RNS.Interface.name = ""
```

Definition at line **37** of file **Interface.cs**.

4.3.3.6 OUT

```
bool RNS.Interface.OUT = false
```

Definition at line **34** of file **Interface.cs**.

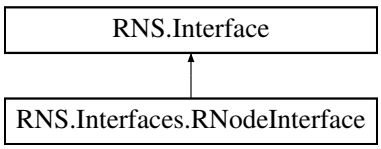
4.3.3.7 RPT

```
bool RNS.Interface.RPT = false
```

Definition at line **36** of file **Interface.cs**.

4.4 RNS.Interfaces.RNodeInterface Class Reference

Inheritance diagram for RNS.Interfaces.RNodeInterface:



Public Member Functions

- **RNodeInterface** (RNS.Transport _owner, string _name, string _port, uint _frequency=0, uint _bandwidth=0, byte _txpower=0, byte _sf=0, byte _cr=0, bool _flow_control=false, int _id_interval=0, string _id_callsign="")

  RNode initialization routine.
- void **Send** (byte[ ] data)

  Simple hook for manual packet transmission. Non-API.
- void **DisableBacklight** ()

  Disables RNode backlight. Non-API, but planned for next Python release. Will be brought into line after release.
- void **EnableBacklight** ()

  Enables RNode backlight. Non-API, but planned for next Python release. Will be brought into line after release.
- void **CloseRadio** ()

Closes the radio port. Non-API. Pending depreciation.

- void **Queue** (byte[ ] Payload)

  Add message to transmit queue.
- void **Configure_Device** ()

  Attemps to send configuration data to the radio. Aborts configuration if radio does not return the configuration.
- void **ReceiveLoop** ()

  Main receive loop. Processes incoming control and data packets, writing to registers or passing off to an event handler as appropriate.
- void **Detect** ()

  Detects the RNode and its firmware version.
- bool **ValidateRadioState** ()

  Detects if the radio's reported state matches, within tolerance, the desired configuration. Frequency is the only variable with a tolerance: +/- 500 Hz. Sets isValidConfig as well as returns a boolean.
- void **InitRadio** ()

  Sends radio initialization commands.
- void **Detach** ()

  Detaches radio.
- void **Disable_External_Framebuffer** ()

  Disables the external framebuffer, returnig the RNode display graphic to internal control.
- void **Enable_External_Framebuffer** ()

  Enables external frame buffer, taking control of RNode display graphic.
- void **Display_Image** (byte[ ] ImageData)

  Sends image data to RNode display.

Public Member Functions inherited from RNS.Interface

- **Interface** ()

  Basic initialization.

Properties

- string **Name** = "Undefined Interface" `[get]`

Additional Inherited Members

Public Attributes inherited from RNS.Interface

- bool **IN** = false
- bool **OUT** = false
- bool **FWD** = false
- bool **RPT** = false
- string **name** = ""
- **CallbackClass Callbacks**
- int **ifac_size**

4.4.1 Detailed Description

Definition at line **33** of file **RNodeInterface.cs**.

4.4.2 Constructor & Destructor Documentation

4.4.2.1 RNodeInterface()

```
RNS.Interfaces.RNodeInterface.RNodeInterface (
            RNS::Transport _owner,
            string _name,
            string _port,
            uint _frequency = 0,
            uint _bandwidth = 0,
            byte _txpower = 0,
            byte _sf = 0,
            byte _cr = 0,
            bool _flow_control = false,
            int _id_interval = 0,
            string _id_callsign = "" )  [inline]
```

RNode initialization routine.

Parameters

_owner	Reticulum specific. NYI.
_name	Interface name. Example: RNode on Server 3
_port	Port name. Example: COM6 or /dev/ttyAMC0
_frequency	Frequency in Hz. Example: 915000000 for a 915MHz signal. Note: You are responsible for selecting a band legal in your nation and municipality.
_bandwidth	Bandwidth in Hz: Example: 125000 for 125kHz bandwidth. Acceptable values from 7800 to 500000.
_txpower	Transmission power in dBm. Acceptable values range from 0 - 17.
_sf	Spreading factor. Acceptable values range from 7 - 12.
_cr	Coding rate, 4 data bits per N transmitted bits. Acceptable values range from 5 - 8.
_flow_control	I'm actually unsure. Will consult with stack designer.
_id_interval	Time, in seconds, between callsign broadcasts. 0 disables.
_id_callsign	Station callsign. When a packet is sent, begins a cooldown of _id_interval seconds, then broadcasts station ID. Meant for amateur radio compliance. string = "" disables.

Exceptions

Exception	Thrown if platform is unsupported.
ArgumentException	Thrown when interface contains errors.

Exceptions

IOException	Thrown if serial port is unavailable.

Definition at line **140** of file **RNodeInterface.cs**.

4.4.3 Member Function Documentation

4.4.3.1 CloseRadio()

```
void RNS.Interfaces.RNodeInterface.CloseRadio ( )  [inline]
```

Closes the radio port. Non-API. Pending depreciation.

Definition at line **796** of file **RNodeInterface.cs**.

4.4.3.2 Configure_Device()

```
void RNS.Interfaces.RNodeInterface.Configure_Device ( )  [inline]
```

Attemps to send configuration data to the radio. Aborts configuration if radio does not return the configuration.

Definition at line **822** of file **RNodeInterface.cs**.

4.4.3.3 Detach()

```
void RNS.Interfaces.RNodeInterface.Detach ( )  [inline]
```

Detaches radio.

Definition at line **1337** of file **RNodeInterface.cs**.

4.4.3.4 Detect()

```
void RNS.Interfaces.RNodeInterface.Detect ( )  [inline]
```

Detects the RNode and its firmware version.

Exceptions

IOException	Thrown on serial error

Definition at line **1239** of file **RNodeInterface.cs**.

4.4.3.5 Disable_External_Framebuffer()

```
void RNS.Interfaces.RNodeInterface.Disable_External_Framebuffer ( )  [inline]
```

Disables the external framebuffer, returnig the RNode display graphic to internal control.

Exceptions

IOException	Thrown on serial error

Definition at line **1350** of file **RNodeInterface.cs**.

4.4.3.6 DisableBacklight()

```
void RNS.Interfaces.RNodeInterface.DisableBacklight ( )  [inline]
```

Disables RNode backlight. Non-API, but planned for next Python release. Will be brought into line after release.

Exceptions

IOException	Thrown on serial error

Definition at line **556** of file **RNodeInterface.cs**.

4.4.3.7 Display_Image()

```
void RNS.Interfaces.RNodeInterface.Display_Image (
            byte[] ImageData )  [inline]
```

Sends image data to RNode display.

Parameters

ImageData	Array of bytes containing image information

Definition at line **1390** of file **RNodeInterface.cs**.

4.4.3.8 Enable_External_Framebuffer()

```
void RNS.Interfaces.RNodeInterface.Enable_External_Framebuffer ( )  [inline]
```

Enables external frame buffer, taking control of RNode display graphic.

Exceptions

IOException	Thrown on serial error

Definition at line **1370** of file **RNodeInterface.cs**.

4.4.3.9 EnableBacklight()

```
void RNS.Interfaces.RNodeInterface.EnableBacklight ( )  [inline]
```

Enables RNode backlight. Non-API, but planned for next Python release. Will be brought into line after release.

Exceptions

IOException	Thrown on serial error

Definition at line **575** of file **RNodeInterface.cs**.

4.4.3.10 InitRadio()

```
void RNS.Interfaces.RNodeInterface.InitRadio ( )  [inline]
```

Sends radio initialization commands.

Definition at line **1314** of file **RNodeInterface.cs**.

4.4.3.11 Queue()

```
void RNS.Interfaces.RNodeInterface.Queue (
            byte[] Payload )  [inline]
```

Add message to transmit queue.

Parameters

Payload	Message data in an array of bytes

Definition at line **809** of file **RNodeInterface.cs**.

4.4.3.12 ReceiveLoop()

```
void RNS.Interfaces.RNodeInterface.ReceiveLoop ( )  [inline]
```

Main receive loop. Processes incoming control and data packets, writing to registers or passing off to an event handler as appropriate.

Exceptions

IOException	Thrown on serial error

Definition at line **960** of file **RNodeInterface.cs**.

4.4.3.13 Send()

```
void RNS.Interfaces.RNodeInterface.Send (
            byte[] data )  [inline]
```

Simple hook for manual packet transmission. Non-API.

Parameters

data	Raw message in an array of bytes.

Definition at line **499** of file **RNodeInterface.cs**.

4.4.3.14 ValidateRadioState()

```
bool RNS.Interfaces.RNodeInterface.ValidateRadioState ( )  [inline]
```

Detects if the radio's reported state matches, within tolerance, the desired configuration. Frequency is the only variable with a tolerance: +/- 500 Hz. Sets isValidConfig as well as returns a boolean.

Returns

True if state is valid, else false

Definition at line **1274** of file **RNodeInterface.cs**.

4.4.4 Property Documentation

4.4.4.1 Name

```
string RNS.Interfaces.RNodeInterface.Name = "Undefined Interface"  [get]
```

Definition at line **80** of file **RNodeInterface.cs**.

Index