




SBC

Class List

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

 CommunicationInterface	Interface for communication operations
 EthernetInterface	Ethernet communication interface implementation
 UARTInterface	UART communication interface implementation

SBC

CommunicationInterface Class Reference abstract

[Public Member Functions](#) |

[List of all members](#)

Interface for communication operations. [More...](#)

Inheritance diagram for CommunicationInterface:



Public Member Functions

virtual	~CommunicationInterface ()=default
virtual HRESULT	Connect (const InterfaceConnection interfaceName, const BSTR comPortOrIp, const UINT32 baudRateOrPort, const BSTR protocol)=0 Establishes a connection with the specified interface.
virtual HRESULT	Disconnect ()=0 Disconnects the active communication interface.
virtual HRESULT	GetConnectionStatus (InterfaceConnection connectedInterfaceName, BOOL *connectionStatus) const =0 Retrieves the connection status of the current interface.
virtual HRESULT	GetHWVersion (HWVersion *hwVersion) const =0 Retrieves the hardware version of the communication device.
virtual HRESULT	GetVersion (const VersionInfo enumVal, UINT32 *versionInfo) const =0 Retrieves the version information of a specified version type.
virtual HRESULT	RunTest (const Group group, UINT32 subTest, BSTR *logOutput)=0 Runs a specific test on the communication interface.

Detailed Description

Interface for communication operations.

Constructor & Destructor Documentation

◆~CommunicationInterface()

virtual

CommunicationInterface::~~CommunicationInterface ()

virtual

default

Member Function Documentation

◆Connect()

virtual HRESULT

CommunicationInterface::Connect (const **InterfaceConnection** inter
const BSTR com
const UINT32 bau
const BSTR prot

Establishes a connection with the specified interface.

Parameters

interfaceName [in] The interface to connect to.
comPortOrIp [in] Additional input parameter (e.g., COM port
baudRateOrPort [in] Baud rate (if serial communication) or port
protocol [in] Communication protocol (e.g., TCP, UDP,

Returns

[out] HRESULT result of the connection attempt.

Implemented in **EthernetInterface**, and **UARTInterface**.

◆Disconnect()

virtual HRESULT

CommunicationInterface::Disconnect () **nodiscard** **pure virtual**

Disconnects the active communication interface.

Returns

[out] HRESULT result of the disconnection attempt.

Implemented in [EthernetInterface](#), and [UARTInterface](#).

◆GetConnectionStatus()

virtual HRESULT

CommunicationInterface::GetConnectionStatus ([InterfaceConnecti](#)
BOOL *

Retrieves the connection status of the current interface.

Parameters

connectedInterfaceName [In] Interface name.

connectionStatus [out] Returns the connection status.

Returns

[out] HRESULT result of the operation.

Implemented in [EthernetInterface](#), and [UARTInterface](#).

◆GetHWVersion()

virtual HRESULT

CommunicationInterface::GetHWVersion ([HWVersion](#) * [hwVersion](#))

Retrieves the hardware version of the communication device.

Parameters

hwVersion [out] Returns the Hardware Version.

Returns

[out] HRESULT result of the operation.

Implemented in **EthernetInterface**, and **UARTInterface**.

◆GetVersion()

virtual HRESULT

CommunicationInterface::GetVersion (const **VersionInfo** enumVal,
UINT32 * versionInfo)

Retrieves the version information of a specified version type.

Parameters

enumVal [in] The version type (enumeration value).

versionInfo [out] Returns version info.

Returns

[out] HRESULT result of the operation.

Implemented in **EthernetInterface**, and **UARTInterface**.

◆RunTest()

virtual HRESULT

CommunicationInterface::RunTest (const **Group** group,
UINT32 subTest,
BSTR * logOutput) nodiscard

Runs a specific test on the communication interface.

Parameters

group [in] The test group to which the test belongs.

subTest [in] The specific sub-test to run or we can send multi of the group by ORing them.

logOutput [out] Returns the log output.

Returns

[out] HRESULT result of the test run.

Implemented in **EthernetInterface**, and **UARTInterface**.

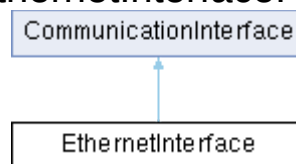
SBC

EthernetInterface Class Reference

[Public Member Functions](#) |
[List of all members](#)

Ethernet communication interface implementation. [More...](#)

Inheritance diagram for EthernetInterface:



Public Member Functions

	EthernetInterface () Default constructor for EthernetInterface .
virtual	~EthernetInterface () Destructor for EthernetInterface .
virtual HRESULT	Connect (const InterfaceConnection interfaceName, const BSTR ip, const UINT32 port, const BSTR protocol) override Establishes a connection with the specified Ethernet interface.
virtual HRESULT	Disconnect () override Disconnects the active Ethernet communication interface.
virtual HRESULT	GetConnectionStatus (InterfaceConnection connectedInterfaceName, BOOL *connectionStatus) const override Retrieves the connection status of the current Ethernet interface.
virtual HRESULT	GetHWVersion (HWVersion *hwVersion) const override Retrieves the hardware version of the Ethernet device.
virtual HRESULT	GetVersion (const VersionInfo enumVal, UINT32 *versionInfo) const override Retrieves the version information of a specified version type.
virtual HRESULT	RunTest (const Group group, UINT32 subTest, BSTR *logOutput) override Runs a specific test on the Ethernet communication interface.
► Public Member Functions inherited from CommunicationInterface	
	virtual ~CommunicationInterface () =default

Detailed Description

Ethernet communication interface implementation.

Constructor & Destructor Documentation

◆EthernetInterface()

EthernetInterface::EthernetInterface ()

Default constructor for **EthernetInterface**.

◆~EthernetInterface()

EthernetInterface::~~EthernetInterface ()

virtual

Destructor for **EthernetInterface**.

Member Function Documentation

◆Connect()

HRESULT

```
EthernetInterface::Connect ( const InterfaceConnection interfaceName,  
                             const BSTR ip,  
                             const UINT32 port,  
                             const BSTR protocol )
```

Establishes a connection with the specified Ethernet interface.

Parameters

interfaceName	[in] The interface to connect to.
ip	[in] IP address of the Ethernet device.
port	[in] Port number of the Ethernet device.
protocol	[in] Communication protocol (e.g., TCP, UDP).

Returns

[out] HRESULT result of the connection attempt.

Implements **CommunicationInterface**.

◆Disconnect()

```
HRESULT EthernetInterface::Disconnect ( )
```

nodiscard

override

virtual

Disconnects the active Ethernet communication interface.

Returns

[out] HRESULT result of the disconnection attempt.

Implements **CommunicationInterface**.

◆GetConnectionStatus()

HRESULT

EthernetInterface::GetConnectionStatus (**InterfaceConnection** **conn**)
BOOL *

Retrieves the connection status of the current Ethernet interface.

Parameters

connectedInterfaceName [In] Interface name.

connectionStatus [out] Returns the connection status.

Returns

[out] HRESULT result of the operation.

Implements **CommunicationInterface**.

◆GetHWVersion()

HRESULT

EthernetInterface::GetHWVersion (**HWVersion** * **hwVersion**) const

Retrieves the hardware version of the Ethernet device.

Parameters

hwVersion [out] Returns the Hardware Version.

Returns

[out] HRESULT result of the operation.

Implements **CommunicationInterface**.

◆GetVersion()

HRESULT

```
EthernetInterface::GetVersion ( const VersionInfo enumVal,  
                                UINT32 *          versionInfo ) const
```

Retrieves the version information of a specified version type.

Parameters

enumVal [in] The version type (enumeration value).
versionInfo [out] Returns version info.

Returns

[out] HRESULT result of the operation.

Implements **CommunicationInterface**.

◆RunTest()

HRESULT

```
EthernetInterface::RunTest ( const Group group,  
                             UINT32      subTest,  
                             BSTR *      logOutput ) nodiscard override
```

Runs a specific test on the Ethernet communication interface.

Parameters

group [in] The test group to which the test belongs.

subTest [in] The specific sub-test to run or we can send multi tests of the group by ORing them.

logOutput [out] Returns the log output.

Returns

[out] HRESULT result of the test run.

Implements **CommunicationInterface**.

SBC

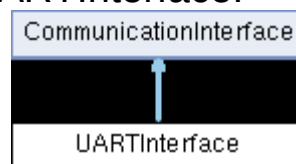
UARTInterface Class Reference

[Public Member Functions](#) |

[List of all members](#)

UART communication interface implementation. [More...](#)

Inheritance diagram for UARTInterface:



Public Member Functions

	UARTInterface () Default constructor for the UART interface.
	~UARTInterface () Destructor for the UART interface. Cleans up allocated resources.
virtual HRESULT	Connect (const InterfaceConnection interfaceName, const BSTR comPortOrIp, const UINT32 baudRateOrPort, const BSTR protocol) override Establishes a connection with the specified UART interface.
virtual HRESULT	Disconnect () override Disconnects the active UART communication interface.
virtual HRESULT	GetConnectionStatus (InterfaceConnection connectedInterfaceName, BOOL *connectionStatus) const override Retrieves the connection status of the current UART interface.
virtual HRESULT	GetHWVersion (HWVersion *hwVersion) const override Retrieves the hardware version of the UART device.
virtual HRESULT	GetVersion (const VersionInfo enumVal, UINT32 *versionInfo) const override Retrieves version information of the UART interface.
virtual HRESULT	RunTest (const Group group, UINT32 subTest, BSTR *logOutput) override Runs a specific test on the UART communication interface.

► **Public Member Functions inherited from [CommunicationInterface](#)**

```
virtual ~CommunicationInterface ()=default
```

Detailed Description

UART communication interface implementation.

This class provides the methods for connecting, disconnecting, and querying the status of a UART interface. The connection uses COM ports and baud rates to establish communication.

Constructor & Destructor Documentation

◆UARTInterface()

UARTInterface::UARTInterface ()

Default constructor for the UART interface.

◆~UARTInterface()

UARTInterface::~~UARTInterface ()

Destructor for the UART interface. Cleans up allocated resources.

Member Function Documentation

◆Connect()

HRESULT

UARTInterface::Connect (const **InterfaceConnection** interfaceName,
const BSTR comPortOrIp,
const UINT32 baudRateOrPort,
const BSTR protocol)

Establishes a connection with the specified UART interface.

Parameters

interfaceName [in] The interface to connect to.
comPortOrIp [in] The COM port for UART communication (e.g., COM1 or 192.168.1.1).
baudRateOrPort [in] The baud rate for UART communication (e.g., 115200 or 1).
protocol [in] Communication protocol, which may not be needed for all communication.

Returns

HRESULT indicating the result of the connection attempt (S_OK for success).

Implements **CommunicationInterface**.

◆Disconnect()

HRESULT UARTInterface::Disconnect ()

nodiscard

override

virtual

Disconnects the active UART communication interface.

Resets all connection details and frees resources.

Returns

HRESULT indicating the result of the disconnection attempt (S_OK for success).

Implements **CommunicationInterface**.

◆GetConnectionStatus()

HRESULT

UARTInterface::GetConnectionStatus (**InterfaceConnection** *connection* ,
BOOL * *connectionStatus*)

Retrieves the connection status of the current UART interface.

Parameters

connectedInterfaceName [In] Interface name.

connectionStatus [out] Returns the connection status.

Returns

[out] HRESULT result of the operation.

Implements **CommunicationInterface**.

◆GetHWVersion()

HRESULT

UARTInterface::GetHWVersion (**HWVersion** * *hwVersion*) const

Retrieves the hardware version of the UART device.

Parameters

hwVersion [out] A Returns the Hardware Version.

Returns

HRESULT indicating the result of the operation (S_OK for success).

Implements **CommunicationInterface**.

◆GetVersion()

HRESULT

```
UARTInterface::GetVersion ( const VersionInfo enumVal,  
                             UINT32 *          versionInfo ) const
```

Retrieves version information of the UART interface.

Parameters

enumVal [in] The version type.

versionInfo [out] Returns version info.

Returns

HRESULT indicating the result of the operation (S_OK for success).

Implements **CommunicationInterface**.

◆RunTest()

HRESULT

```
UARTInterface::RunTest ( const Group group,  
                          UINT32      subTest,  
                          BSTR *      logOutput ) nodiscard override
```

Runs a specific test on the UART communication interface.

Parameters

- group** [in] The test group to which the test belongs.
- subTest** [in] The specific sub-test to run or we can send multi tests of the group by ORing them.
- logOutput** [out] Returns the log output.

Returns

HRESULT indicating the result of the test run (S_OK for succes




Implements **CommunicationInterface**.

SBC

Class Hierarchy

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

[detail level [1](#) [2](#)]

▼  CommunicationInterface	Interface for communication operations
 EthernetInterface	Ethernet communication interface implementation
 UARTInterface	UART communication interface implementation

Generated by  1.13.2

SBC

Here is a list of all class members with links to the classes they belong to:

- Connect() : [CommunicationInterface](#), [EthernetInterface](#), [UARTInterface](#)
- Disconnect() : [CommunicationInterface](#), [EthernetInterface](#), [UARTInterface](#)
- EthernetInterface() : [EthernetInterface](#)
- GetConnectionStatus() : [CommunicationInterface](#), [EthernetInterface](#), [UARTInterface](#)
- GetHWVersion() : [CommunicationInterface](#), [EthernetInterface](#), [UARTInterface](#)
- GetVersion() : [CommunicationInterface](#), [EthernetInterface](#), [UARTInterface](#)
- RunTest() : [CommunicationInterface](#), [EthernetInterface](#), [UARTInterface](#)
- UARTInterface() : [UARTInterface](#)
- ~CommunicationInterface() : [CommunicationInterface](#)
- ~EthernetInterface() : [EthernetInterface](#)
- ~UARTInterface() : [UARTInterface](#)

SBC

Here is a list of all functions with links to the classes they belong to:

- Connect() : [CommunicationInterface](#), [EthernetInterface](#), [UARTInterface](#)
- Disconnect() : [CommunicationInterface](#), [EthernetInterface](#), [UARTInterface](#)
- EthernetInterface() : [EthernetInterface](#)
- GetConnectionStatus() : [CommunicationInterface](#), [EthernetInterface](#), [UARTInterface](#)
- GetHWVersion() : [CommunicationInterface](#), [EthernetInterface](#), [UARTInterface](#)
- GetVersion() : [CommunicationInterface](#), [EthernetInterface](#), [UARTInterface](#)
- RunTest() : [CommunicationInterface](#), [EthernetInterface](#), [UARTInterface](#)
- UARTInterface() : [UARTInterface](#)
- ~CommunicationInterface() : [CommunicationInterface](#)
- ~EthernetInterface() : [EthernetInterface](#)
- ~UARTInterface() : [UARTInterface](#)

SBC

File List

Here is a list of all files with brief descriptions:

CommunicationInterface.cpp	
CommunicationInterface.h	
devdefs.h	
EthernetInterface.cpp	
EthernetInterface.h	
UARTInterface.cpp	
UARTInterface.h	

Generated by doxygen 1.13.2

SBC

C:/Users/Mourya/source/repos/DLL_SBC/CommunicationInterface.cpp File Reference

Generated by  1.13.2

SBC

[Classes](#) | [Macros](#)

C:/Users/Mourya/source/repos/DLL_SBC/CommunicationInterface.h File Reference

Classes

class **CommunicationInterface**
Interface for communication operations. [More...](#)

Macros

```
#define EXPORT_API __declspec(dllexport)
```

Macro Definition Documentation

◆ EXPORT_API

```
#define EXPORT_API __declspec(dllexport)
```

Generated by [doxygen](https://doxygen.org) 1.13.2

SBC

Enumerations

C:/Users/Mourya/source/repos/DLL_SBC/devdefs.h File Reference

Enumerations

enum class	HWVersion { SBC_6U }	Enumerates supported hardware versions. More...
enum class	Group { Interface , PBIT , IPMC }	Enumerates available test or functionality groups. More...
enum class	InterfaceTests { PCIE_GEN4_X8_VPX = 1 << 0 , PCIE_SWITCH_X8_GEN3 = 1 << 1 , PCIE_SWITCH_VPX_X4_GEN3_CH0 = 1 << 2 , PCIE_SWITCH_VPX_X4_GEN3_CH1 = 1 << 3 , PCIE_SWITCH_XMC_X8_GEN3_CH0 = 1 << 4 , PCIE_SWITCH_XMC_X8_GEN3_CH1 = 1 << 5 , PCIE_FPGA_GEN3_X4 = 1 << 6 , PCIE_FPGA_GEN3_X8 = 1 << 7 , PCIE_ETH_CONTROLLER_1GBPS = 1 << 8 , XEON_40G_BASE = 1 << 9 , XEON_100G_BASE = 1 << 10 , SATA_NAND_FLASH = 1 << 11 , SATA_VPX_CH0 = 1 << 12 , SATA_VPX_CH1 = 1 << 13 , SATA_VPX_CH2 = 1 << 14 , SATA_VPX_M2 = 1 << 15 , LPC_TEST = 1 << 16 , XEON_IPMC_UART = 1 << 17 , XEON_IPMC_I2C = 1 << 18 , XEON_EMMC_TEST = 1 << 19 , XEON_VPX_UART = 1 << 20 , XEON_USB = 1 << 21 , XEON_VPX_USB_CH1 = 1 << 22 , XEON_VPX_USB_CH2 = 1 << 23 }	Enumerates top-level group commands. More...
enum class	VersionInfo { FPGA_VERSION , API_VERSION , GUI_VERSION , FIRMWARE_VERSION }	
enum class	PBITTests { PBIT_DATA }	Bitmask flags representing tests for PBIT (Power-on Built-In Test) operations. More...

```
enum class InterfaceConnection { INTERFACE_UART ,  
    INTERFACE_ETHERNET , INTERFACE_UNKOWN }  
Enum representing different communication interfaces.  
More...
```

Enumeration Type Documentation

◆HWVersion

enum class **HWVersion**

strong

Enumerates supported hardware versions.

Enumerator

SBC_6U	6U Single Board Computer
--------	--------------------------

◆Group

enum class **Group**

strong

Enumerates available test or functionality groups.

Enumerator

Interface	Interface-related group.
PBIT	Power-on Built-In Test group.
IPMC	Intelligent Platform Management Controller group.

◆InterfaceTests

enum class **InterfaceTests**

strong

Enumerates top-level group commands.

Defines bitmask flags for various interface tests.

Enumerator	
PCIE_GEN4_X8_VPX	PCle Gen4 x8 VPX interface.
PCIE_SWITCH_X8_GEN3	PCle switch x8 Gen3.
PCIE_SWITCH_VPX_X4_GEN3_CH0	PCle switch VPX x4 Gen3 channel 0.
PCIE_SWITCH_VPX_X4_GEN3_CH1	PCle switch VPX x4 Gen3 channel 1.
PCIE_SWITCH_XMC_X8_GEN3_CH0	PCle switch XMC x8 Gen3 channel 0.
PCIE_SWITCH_XMC_X8_GEN3_CH1	PCle switch XMC x8 Gen3 channel 1.
PCIE_FPGA_GEN3_X4	PCle FPGA Gen3 x4.
PCIE_FPGA_GEN3_X8	PCle FPGA Gen3 x8.
PCIE_ETH_CONTROLLER_1GBPS	PCle Ethernet controller 1 Gbps.
XEON_40G_BASE	Xeon 40GBase interface.
XEON_100G_BASE	Xeon 100GBase interface.
SATA_NAND_FLASH	SATA NAND flash storage.
SATA_VPX_CH0	SATA VPX channel 0.
SATA_VPX_CH1	SATA VPX channel 1.
SATA_VPX_CH2	SATA VPX channel 2.
SATA_VPX_M2	SATA M.2 VPX interface.

LPC_TEST	LPC interface test.
XEON_IPMC_UART	UART interface between Xeon and IPMC.
XEON_IPMC_I2C	I2C interface between Xeon and IPMC.
XEON_EMMC_TEST	Xeon eMMC interface test.
XEON_VPX_UART	UART over VPX on Xeon.
XEON_USB	USB interface on Xeon.
XEON_VPX_USB_CH1	USB VPX channel 1 on Xeon.
XEON_VPX_USB_CH2	USB VPX channel 2 on Xeon.

◆VersionInfo

enum class **VersionInfo**

strong

Enumerator

FPGA_VERSION	Version of the FPGA (Field-Programmable Gate Array) component.
API_VERSION	Version of the system's Application Programming Interface (API).
GUI_VERSION	Version of the Graphical User Interface (GUI) component.
FIRMWARE_VERSION	Version of the firmware running on the hardware.

◆PBITTests

enum class **PBITTests**

strong

Bitmask flags representing tests for PBIT (Power-on Built-In Test) operations.

Enumerator

PBIT_DATA	PBIT data test flag.
-----------	----------------------

◆InterfaceConnection

enum class **InterfaceConnection**

strong

Enum representing different communication interfaces.

This enum class is used to specify various types of communication interfaces supported by the system. Each value corresponds to a specific communication protocol, such as UART or Ethernet. It helps in identifying and selecting the appropriate interface for communication operations.

Enumerator

INTERFACE_UART	UART communication interface. This value represents a UART (Universal Asynchronous Receiver-Transmitter) interface used for serial communication. It is commonly used in embedded systems for low-speed communication with peripherals.
INTERFACE_ETHERNET	Ethernet communication interface.

	<p>This value represents an Ethernet interface used for network communication. It is typically used for high-speed, reliable communication over IP networks.</p>
INTERFACE_UNKOWN	<p>Unknown interface.</p> <p>This value represents an unknown or undefined communication interface. It is typically used as a placeholder or default value when the interface type cannot be determined or is not yet set.</p>

SBC

C:/Users/Mourya/source/repos/DLL_SBC/EthernetInterface.cpp File Reference

Generated by  1.13.2

SBC

[Classes](#) | [Macros](#)

C:/Users/Mourya/source/repos/DLL_SBC/EthernetInterface.h File Reference

Classes

class **EthernetInterface**

Ethernet communication interface implementation. [More...](#)

Macros

```
#define EXPORT_API __declspec(dllexport)
```

Macro Definition Documentation

◆ EXPORT_API

```
#define EXPORT_API __declspec(dllexport)
```

Generated by [doxygen](https://doxygen.org) 1.13.2

SBC

C:/Users/Mourya/source/repos/DLL_SBC/UARTInterface.cpp File Reference

Generated by  1.13.2

SBC

[Classes](#) | [Macros](#)

C:/Users/Mourya/source/repos/DLL_SBC/UARTInterface.h File Reference

Classes

class **UARTInterface**

UART communication interface implementation. [More...](#)

Macros

```
#define EXPORT_API __declspec(dllexport)
```

Macro Definition Documentation

◆ EXPORT_API

```
#define EXPORT_API __declspec(dllexport)
```

Generated by [doxygen](https://doxygen.org) 1.13.2

SBC

Here is a list of all file members with links to the files they belong to:

- EXPORT_API : [CommunicationInterface.h](#), [EthernetInterface.h](#), [UARTInterface.h](#)
- Group : [devdefs.h](#)
- HWVersion : [devdefs.h](#)
- InterfaceConnection : [devdefs.h](#)
- InterfaceTests : [devdefs.h](#)
- PBITTests : [devdefs.h](#)
- VersionInfo : [devdefs.h](#)

SBC

Here is a list of all enums with links to the files they belong to:

- Group : [devdefs.h](#)
- HWVersion : [devdefs.h](#)
- InterfaceConnection : [devdefs.h](#)
- InterfaceTests : [devdefs.h](#)
- PBITTests : [devdefs.h](#)
- VersionInfo : [devdefs.h](#)

SBC

Here is a list of all macros with links to the files they belong to:

- EXPORT_API : [CommunicationInterface.h](#),
[EthernetInterface.h](#), [UARTInterface.h](#)

SBC

CommunicationInterface Member List

This is the complete list of members for **CommunicationInterface**, including all inherited members.

Connect(const InterfaceConnection interfaceName, const BSTR con
Disconnect()=0

GetConnectionStatus(InterfaceConnection connectedInterfaceNam

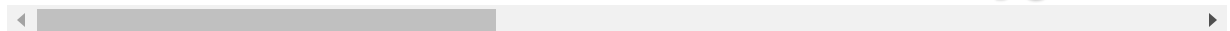
GetHWVersion(HWVersion *hwVersion) const =0

GetVersion(const VersionInfo enumVal, UINT32 *versionInfo) const :

RunTest(const Group group, UINT32 subTest, BSTR *logOutput)=0

~CommunicationInterface()=default

Generated by [doxygen](#) 1.13.2



SBC

EthernetInterface Member List

This is the complete list of members for **EthernetInterface**, including all inherited members.

Connect(const InterfaceConnection interfaceName, const BSTR ip, c

Disconnect() override

EthernetInterface()

GetConnectionStatus(InterfaceConnection connectedInterfaceNam

GetHWVersion(HWVersion *hwVersion) const override

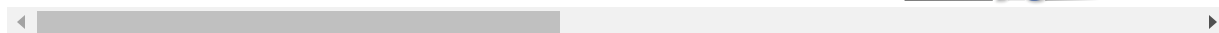
GetVersion(const VersionInfo enumVal, UINT32 *versionInfo) const c

RunTest(const Group group, UINT32 subTest, BSTR *logOutput) ove

~CommunicationInterface()=default

~EthernetInterface()

Generated by [doxygen](http://doxygen.org) 1.13.2



SBC

UARTInterface Member List

This is the complete list of members for **UARTInterface**, including all inherited members.

Connect(const InterfaceConnection interfaceName, const BSTR con
Disconnect() override

GetConnectionStatus(InterfaceConnection connectedInterfaceNam

GetHWVersion(HWVersion *hwVersion) const override

GetVersion(const VersionInfo enumVal, UINT32 *versionInfo) const (

RunTest(const Group group, UINT32 subTest, BSTR *logOutput) ove

UARTInterface()

~CommunicationInterface()=default

~UARTInterface()