

AsiHpiDotNet.dll

This .NET assembly contains only a subset of the HPI functionality as this module started as a bespoke project to support GPIO functions. If you wish to use this assembly but require access to missing HPI features please contact ASI support.

AsiHpi Namespace

HpiException Class

Parent class: Exception

HpiSubsystem Class

Properties

NumAdapters (int, read only)

Version (uint, read only)

VersionString (String, read only)

Methods

GetAdapterIndexAndType(int iterator)

Returns the index and type for the specified adapter.

Note: valid values for iterator are 0 through (*NumAdapters* - 1) but the returned adapter indices may be outside that range and non-contiguous.

returns: Tuple<uint adapterIndex, String adapterType>

OpenAdapter(ushort adapterIndex)

Opens the specified adapter and returns the Adapter object

returns: Adapter object

Adapter Class

Constructor

Adapter(HpiSubsystem ss,ushort i)

Use the OpenAdapter() method of the HpiSubsystem class rather than calling this constructor directly.

Properties

Index (ushort, read only)

The adapter index.

Methods

GetInfo()

Returns the adapter information (output stream count, input stream count, hardware version, serial number and adapter type) in a Tuple.

returns: Tuple<ushort numOutStreams,ushort numInStreams,ushort Version,uint SerialNumber,ushort AdapterType>

OpenMixer()

Opens the adapter's mixer and returns the Mixer class object.

Mixer Class

Constructor

Mixer(Adapter _adapter)

Use the OpenMixer() method of the Adapter class rather than calling this constructor directly.

Properties

Handle (uint, read only)

InputLines

List of all input lines.

OutputLines

List of all output lines.

Controls

List of all controls.

Methods

GetGpio()

Returns a Gpio class object or raises HpiException if the control is not found.

FindInputLine(HPI_SOURCENODES _SrcNodeType, uint _Index, bool addIfNotFound = false)

Returns the MixerLine object for specific input type and index. If it's not found it returns **null** unless the **addIfNotFound** parameter is true, in which case it is created.

FindOutputLine(HPI_DESTNODES _DstNodeType, uint _Index, bool addIfNotFound = false)

Returns the MixerLine object for specific output type and index. If it's not found it returns **null** unless the **addIfNotFound** parameter is true, in which case it is created.

MixerLine Class

Constructor

MixerLine(HPI_SOURCENODES _SrcNodeType, HPI_DESTNODES _DstNodeType, uint _Index, Mixer _Mixer)

Create a MixerLine object. One of _SrcNodeType or _DstNodeType must be HPI_xNODE_NONE.

Properties

Controls (List<Control>, read/write)

List of all controls on this node. The same control object can appear on both a source and destination node (i.e. volume control).

mixer(Mixer)

Handle to the Mixer object that created this MixerLine.

SourceType(HPI_SOURCENODES, read only)

DestinationType(HPI_DESTNODES, read only)

Index(uint,read only)

Methods

AddControl(Control ctrl)

Adds a control to **Controls**.

GetMixer()

Returns the mixer object.

Control Class

Base class for control classes and placeholder for control types that aren't implemented.

Constructor

Control(MixerLine _SrcLine,MixerLine _DstLine,HPI_CONTROLS _Type,uint _hControl)

Create a Control object.

Properties

Handle (uint, read-only)

SourceLine (MixerLine, read-only)

DestinationLine (MixerLine, read-only)

ControlType (HPI_CONTROLS, read-only)

Multiplexer Class

Subclass of **Control** for multiplexer control.

Constructor

Multiplexer(MixerLine _SrcLine,MixerLine _DstLine,uint _hControl)

Create a Multiplexer object.

Properties

Inputs (MixerLine, read-only)

Selection (MixerLine, read/write)

Ballistics Class

Utility class used to get/set ballistics on Meter controls.

Constructor

Ballistics(UInt16 _Attack,UInt16 _Decay)

Create a Multiplexer object.

Properties

Attack (UInt16, read/write)

Decay (UInt16, read/write)

Meter Class

Subclass of **Control** for meter control.

Constructor

Meter(MixerLine _SrcLine,MixerLine _DstLine,uint _hControl)

Create a Meter object.

Properties

Peak (short[], read-only)

PeakBallistics (Ballistics, read/write)

Rms (short[], read-only)

RmsBallistics (Ballistics, read/write)

Subclass of **Control** for meter control.

Constructor

Meter(MixerLine _SrcLine,MixerLine _DstLine,uint _hControl)

Create a Meter object.

Properties

Peak (short[], read-only)

PeakBallistics (Ballistics, read/write)

Rms (short[], read-only)

RmsBallistics (Ballistics, read/write)

Tuner Class

Subclass of **Control** for tuner control.

Constructor

Tuner(MixerLine _SrcLine,MixerLine _DstLine,uint _hControl)

Create a Tuner object.

Properties

AvailableBands (List<HPI_TUNER_BAND>, read-only)

Band (HPI_TUNER_BAND, read/write)

Frequency (UInt32, read/write)

Gain (short, read/write)

SupportedGainList (List<short>, read-only)

RFLevel (short, read-only)

RawRFLevel (short, read-only)

Status (Dictionary<HPI_TUNER_STATUS_BITS,bool>, read-only)

Program (uint, read-only)

AvailablePrograms (List<uint>, read-only)

PAD Class

Subclass of **Control** for PAD (Program Associated Data) control.

Constructor

PAD(MixerLine _SrcLine,MixerLine _DstLine,uint _hControl)

Create a PAD object.

Properties

ChannelName (String, read-only)

Artist (String, read-only)

Title (String, read-only)

Comment (String, read-only)

Gpio Class

Constructor

Gpio(Mixer _mixer)

Use the GetGpio() method of the Mixer class rather than calling this constructor directly.

Properties

Outputs (List<GpioOutput>)

Inputs (List<GpioInput>)

NumOutputs (UIntPtr, read only)

NumInputs (UIntPtr, read only)

Methods

GetOutputValue(UIntPtr nOutput)

This function is used internally by the GpioOutput class's Value property.

returns: Boolean (current value)

SetOutputValue(UIntPtr nOutput,Boolean Value)

This function is used internally by the GpioOutput class's Value property.

returns: Boolean (previous value)

GetInputValue(UIntPtr nInput)

This function is used internally by the GpioInput class's Value property.

returns: Boolean (current value)

GpioOutput Class

Constructor

GpioOutput(Gpio _gpio, UIntPtr _num)

Used internally by the Gpio constructor.

Properties

Value (Boolean, read/write)

GpioInput Class

Constructor

GpioInput(Gpio _gpio, UIntPtr _num)

Used internally by the Gpio constructor.

Properties

Value (Boolean, read only)