Phcc Interface Library for .NET and COM v0.2.1.0 reference library
Phcc Namespace

Send Feedback
### Classes

<table>
<thead>
<tr>
<th>Class</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>AnalogInputChangedEventArgs</strong></td>
<td>objects hold data that the PHCC motherboard provides whenever an analog input value has changed. The AnalogInputChanged event provides AnalogInputChangedEventArgs event-args objects during the raising of each event.</td>
</tr>
<tr>
<td><strong>Device</strong></td>
<td>The Device class provides methods for communicating with the PHCC motherboard and any attached peripherals, via RS232. The PHCC USB interface also appears to Windows as a standard RS232 COM port.</td>
</tr>
<tr>
<td><strong>DigitalInputChangedEventArgs</strong></td>
<td>objects hold data that the PHCC motherboard provides whenever a digital input value changes. The DigitalInputChanged event provides DigitalInputChangedEventArgs event-args objects during the raising of each event.</td>
</tr>
<tr>
<td><strong>I2CDataReceivedEventArgs</strong></td>
<td>objects hold I2C data that is received when the PHCC motherboard signals that new I2C data has arrived. This data is provided by the I2CDataReceived event.</td>
</tr>
</tbody>
</table>
## Interfaces

<table>
<thead>
<tr>
<th>Interface</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>PhccEvents</td>
<td>COM Event Source Interface</td>
</tr>
</tbody>
</table>
# Delegates

<table>
<thead>
<tr>
<th>Delegate</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>AnalogInputChangedEventHandler</td>
<td>Event handler delegate for the AnalogInputChanged event.</td>
</tr>
<tr>
<td>I2CDataReceivedEventHandler</td>
<td>Event handler delegate for the I2CDataReceived event.</td>
</tr>
</tbody>
</table>
## Enumerations

<table>
<thead>
<tr>
<th>Enumeration</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>LcdDataModes</td>
<td>Enumeration of LCD data modes.</td>
</tr>
<tr>
<td>MotorDirections</td>
<td>Enumeration of stepper motor directions.</td>
</tr>
<tr>
<td>MotorStepTypes</td>
<td>Enumeration of possible stepper motor step types.</td>
</tr>
<tr>
<td>SevenSegmentBits</td>
<td></td>
</tr>
</tbody>
</table>

Send [feedback](mailto:feedback@microsoft.com) on this topic to Microsoft.
AnalogInputChangedEventArgs objects hold data that the PHCC motherboard provides whenever an analog input value has changed. The AnalogInputChanged event provides AnalogInputChangedEventArgs event-args objects during the raising of each event.

**Namespace:** Phcc  
**Assembly:** Phcc (in Phcc.dll)
Syntax

Visual Basic (Declaration)

<
[ClassInterfaceAttribute(ClassInterfaceType.AutoDual)]_
Public NotInheritable Class AnalogInputChangedEventArgs_
Inherits EventArgs

C#

[ClassInterfaceAttribute(ClassInterfaceType.AutoDual)]
public sealed class AnalogInputChangedEventArgs : EventArgs
Inheritance Hierarchy

System...:::Object
System...:::EventArgs
Phcc...:::AnalogInputChangedEventArgs
See Also

AnalogInputChangedEventArgs Members
Phcc Namespace

Send feedback on this topic to Microsoft.
Phcc Interface Library for .NET and COM v0.2.1.0 reference library

AnalogInputChangedEventArgs Constructor

AnalogInputChangedEventArgs Class  See Also  Send Feedback
# Overload List

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>AnalogInputChangedEventArgs()</code></td>
<td>Creates an instance of the <code>AnalogInputChangedEventArgs</code> class.</td>
</tr>
<tr>
<td><code>AnalogInputChangedEventArgs(Byte, Int16)</code></td>
<td>Creates an instance of the <code>AnalogInputChangedEventArgs</code> class.</td>
</tr>
</tbody>
</table>
See Also

AnalogInputChangedEventArgs Class
AnalogInputChangedEventArgs Members
Phcc Namespace

Send feedback on this topic to Microsoft.
Phcc Interface Library for .NET and COM v0.2.1.0 reference library

AnalogInputChangedEventArgs Constructor

Creates an instance of the `AnalogInputChangedEventArgs` class.

**Namespace:** Phcc

**Assembly:** Phcc (in Phcc.dll)
Syntax

Visual Basic (Declaration)

Public Sub New

C#

public AnalogInputChangedEventArgs()
See Also

AnalogInputChangedEventArgs Class
AnalogInputChangedEventArgs Overload
Phcc Namespace

Send feedback on this topic to Microsoft.
Phcc Interface Library for .NET and COM v0.2.1.0 reference library

AnalogInputChangedEventArgs Constructor (Byte, Int16)

AnalogInputChangedEventArgs Class  See Also  Send Feedback

Creates an instance of the AnalogInputChangedEventArgs class.

**Namespace:**  Phcc

**Assembly:**  Phcc (in Phcc.dll)
Syntax

Visual Basic (Declaration)

Public Sub New ( _
    index As Byte, _
    newValue As Short _
)

C#

public AnalogInputChangedEventArgs(
    byte index,
    short newValue
)

Parameters

index
Type: System.Byte
The index of the analog input whose value has changed. The first 3 analog input indexes represent the prioritized analog inputs (ANP1-ANP3); the remaining 32 analog input indexes represent the standard non-prioritized analog inputs (AN1-AN32).

newValue
Type: System.Int16
The new value of the indicated analog input.
See Also

AnalogInputChangedEventArgs Class
AnalogInputChangedEventArgs Overload
Phcc Namespace

Send feedback on this topic to Microsoft.
Phcc Interface Library for .NET and COM v0.2.1.0 reference library

AnalogInputChangedEventArgs Members

See Also  Send Feedback

The `AnalogInputChangedEventArgs` type exposes the following members.
### Constructors

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><a href="#">AnalogInputChangedEventArgs</a></td>
<td>Overloaded.</td>
</tr>
</tbody>
</table>
### Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Equals</td>
<td>(Inherited from <a href="#">Object</a>)</td>
</tr>
<tr>
<td>Finalize</td>
<td>(Inherited from <a href="#">Object</a>)</td>
</tr>
<tr>
<td>GetHashCode</td>
<td>(Inherited from <a href="#">Object</a>)</td>
</tr>
<tr>
<td>GetType</td>
<td>(Inherited from <a href="#">Object</a>)</td>
</tr>
<tr>
<td>MemberwiseClone</td>
<td>(Inherited from <a href="#">Object</a>)</td>
</tr>
<tr>
<td>ToString</td>
<td>(Inherited from <a href="#">Object</a>)</td>
</tr>
</tbody>
</table>
## Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Index</strong></td>
<td>Gets/sets the index of the analog input whose value has changed. The first 3 analog input indexes represent the prioritized analog inputs (ANP1-ANP3); the remaining 32 analog input indexes represent the standard analog non-prioritized inputs (AN1-AN32).</td>
</tr>
<tr>
<td><strong>NewValue</strong></td>
<td>Gets/sets the new value of the indicated analog input.</td>
</tr>
</tbody>
</table>
See Also

AnalogInputChangedEventArgs Class
Phcc Namespace

Send feedback on this topic to Microsoft.
The `AnalogInputChangedEventArgs` type exposes the following members.
## Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Equals</td>
<td>(Inherited from <a href="#">Object</a>)</td>
</tr>
<tr>
<td>Finalize</td>
<td>(Inherited from <a href="#">Object</a>)</td>
</tr>
<tr>
<td>GetHashCode</td>
<td>(Inherited from <a href="#">Object</a>)</td>
</tr>
<tr>
<td>GetType</td>
<td>(Inherited from <a href="#">Object</a>)</td>
</tr>
<tr>
<td>MemberwiseClone</td>
<td>(Inherited from <a href="#">Object</a>)</td>
</tr>
<tr>
<td>ToString</td>
<td>(Inherited from <a href="#">Object</a>)</td>
</tr>
</tbody>
</table>
See Also

AnalogInputChangedEventArgs Class
Phcc Namespace

Send feedback on this topic to Microsoft.
The `AnalogInputChangedEventArgs` type exposes the following members.
## Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Index</td>
<td>Gets/sets the index of the analog input whose value has changed. The first 3 analog input indexes represent the prioritized analog inputs (ANP1-ANP3); the remaining 32 analog input indexes represent the standard analog non-prioritized inputs (AN1-AN32).</td>
</tr>
<tr>
<td>NewValue</td>
<td>Gets/sets the new value of the indicated analog input.</td>
</tr>
</tbody>
</table>
See Also

AnalogInputChangedEventArgs Class
Phcc Namespace

Send feedback on this topic to Microsoft.
AnalogInputChangedEventArgs

Gives sets the index of the analog input whose value has changed. The first 3 analog input indexes represent the prioritized analog inputs (ANP1-ANP3); the remaining 32 analog input indexes represent the standard analog non-prioritized inputs (AN1-AN32).

Namespace: Phcc
Assembly: Phcc (in Phcc.dll)
Syntax

Visual Basic (Declaration)

Public Property Index As Byte

C#

public byte Index { get; set; }
See Also

AnalogInputChangedEventArgs Class
Phcc Namespace

Send feedback on this topic to Microsoft.
Phcc Interface Library for .NET and COM v0.2.1.0 reference library

AnalogInputChangedEventArgs

**::**

**NewValue Property**

AnalogInputChangedEventArgs Class

See Also

Send Feedback

Gets/sets the new value of the indicated analog input.

**Namespace:** Phcc

**Assembly:** Phcc (in Phcc.dll)
Syntax

Visual Basic (Declaration)

Public Property NewValue As Short

C#

public short NewValue { get; set; }
See Also

AnalogInputChangedEventArgs Class
Phcc Namespace

Send feedback on this topic to Microsoft.
Event handler delegate for the **AnalogInputChanged** event.

**Namespace:**  Phcc  
**Assembly:**  Phcc (in Phcc.dll)
Syntax

Visual Basic (Declaration)

```vbnet
<ComVisibleAttribute(False)> _
Public Delegate Sub AnalogInputChangedEventHandler ( _
    sender As Object, _
    e As AnalogInputChangedEventArgs _
)
```

C#

```csharp
[ComVisibleAttribute(false)]
public delegate void AnalogInputChangedEventHandler( 
    Object sender, 
    AnalogInputChangedEventArgs e
)
```

Parameters

sender
Type: `System::Object`
The object raising the event.

e
Type: `Phcc::AnalogInputChangedEventArgs`
A `AnalogInputChangedEventArgs` object containing detailed information about the event.
See Also

Phcc Namespace

Send feedback on this topic to Microsoft.
The Device class provides methods for communicating with the PHCC motherboard and any attached peripherals, via RS232. The PHCC USB interface also appears to Windows as a standard RS232 COM port.

**Namespace:**  Phcc  
**Assembly:**  Phcc (in Phcc.dll)
**Syntax**

**Visual Basic (Declaration)**

```vbnet
<
[SynchronizationAttribute]
[ClassInterfaceAttribute(ClassInterfaceType.AutoDual)]
Public NotInheritable Class Device
Inherits ContextBoundObject
Implements IDisposable
```

**C#**

```csharp
[SynchronizationAttribute]
[ClassInterfaceAttribute(ClassInterfaceType.AutoDual)]
public sealed class Device : ContextBoundObject,
IDisposable
```
Inheritance Hierarchy

System::Object
System::MarshalByRefObject
System::ContextBoundObject
Phcc::Device
See Also

Device Members
Phcc Namespace

Send feedback on this topic to Microsoft.
Visual Basic  C#
Include Protected Members
Include Inherited Members
Phcc Interface Library for .NET and COM v0.2.1.0 reference library
Device Constructor
Device Class  See Also  Send Feedback
### Overload List

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>Device()</code></td>
<td>Creates an instance of the <code>Device</code> class.</td>
</tr>
<tr>
<td><code>Device(String)</code></td>
<td>Creates an instance of the <code>Device</code> class.</td>
</tr>
</tbody>
</table>
See Also

Device Class
Device Members
Phcc Namespace

Send feedback on this topic to Microsoft.
Phcc Interface Library for .NET and COM v0.2.1.0 reference library

Device Constructor

See Also  Send Feedback

Creates an instance of the Device class.

**Namespace:** Phcc

**Assembly:** Phcc (in Phcc.dll)
Syntax

Visual Basic (Declaration)

Public Sub New

C#

public Device()
See Also

Device Class
Device Overload
Phcc Namespace

Send feedback on this topic to Microsoft.
Creates an instance of the Device class.

**Namespace:** Phcc  
**Assembly:** Phcc (in Phcc.dll)
Syntax

Visual Basic (Declaration)

Public Sub New ( _
    portName As String _
)

C#

public Device(
    string portName
)

Parameters

portName

Type: System::String

The name of the COM port to use for communicating with the PHCC motherboard (i.e. "COM1", "COM2", etc.)
See Also

Device Class
Device Overload
Phcc Namespace

Send feedback on this topic to Microsoft.
The **Device** type exposes the following members.
### Events

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>AnalogInputChanged</td>
<td>The <strong>AnalogInputChanged</strong> event is raised when the PHCC motherboard detects that one of the analog inputs has changed values (i.e. whenever an analog input signal changes state).</td>
</tr>
<tr>
<td>DigitalInputChanged</td>
<td>The <strong>DigitalInputChanged</strong> event is raised when the PHCC motherboard detects that one of the digital inputs has changed (i.e. whenever a button that is wired into the digital input key matrix is pressed or released).</td>
</tr>
<tr>
<td>I2CDataReceived</td>
<td>The <strong>I2CDataReceived</strong> event is raised when the PHCC motherboard receives data from one of the attached I2C peripherals (if any).</td>
</tr>
</tbody>
</table>
See Also

Device Class
Phcc Namespace

Send feedback on this topic to Microsoft.
The AnalogInputChanged event is raised when the PHCC motherboard detects that one of the analog inputs has changed values (i.e. whenever an analog input signal changes state).

**Namespace:**  Phcc

**Assembly:**  Phcc (in Phcc.dll)
Syntax

Visual Basic (Declaration)

Public Event AnalogInputChanged As AnalogInputChangedEventHandler

C#

public event AnalogInputChangedEventHandler AnalogInputChanged
See Also

Device Class
Phcc Namespace

Send feedback on this topic to Microsoft.
The DigitalInputChanged event is raised when the PHCC motherboard detects that one of the digital inputs has changed (i.e. whenever a button that is wired into the digital input key matrix is pressed or released).

**Namespace:** Phcc

**Assembly:** Phcc (in Phcc.dll)
Syntax

Visual Basic (Declaration)

Public Event DigitalInputChanged As DigitalInputChangedEventHandler

C#

public event DigitalInputChangedEventHandler DigitalInputChanged
See Also

Device Class
Phcc Namespace

Send feedback on this topic to Microsoft.
The I2CDataReceived event is raised when the PHCC motherboard receives data from one of the attached I2C peripherals (if any).

**Namespace:** Phcc  
**Assembly:** Phcc (in Phcc.dll)
Syntax

Visual Basic (Declaration)

Public Event I2CDataReceived As I2CDataReceivedEventHandler

C#

public event I2CDataReceivedEventHandler I2CDataReceived
See Also

Device Class
Phcc Namespace

Send feedback on this topic to Microsoft.
The `Device` type exposes the following members.
# Constructors

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Device</td>
<td>Overloaded.</td>
</tr>
</tbody>
</table>
## Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CharTo7Seg</td>
<td>(Inherited from MarshalByRefObject.)</td>
</tr>
<tr>
<td>CreateObjRef</td>
<td>Public implementation of IDisposable.Dispose(). Cleans up managed and</td>
</tr>
<tr>
<td></td>
<td>unmanaged resources used by this object before allowing garbage collection</td>
</tr>
<tr>
<td>Dispose</td>
<td></td>
</tr>
<tr>
<td>DoaSend40DO</td>
<td>Sends data to a DOA_40DO digital output daughtercard.</td>
</tr>
<tr>
<td>DoaSend7Seg</td>
<td>Sends data to a DOA_7Seg daughtercard to control the 7-segment LCDs (or</td>
</tr>
<tr>
<td></td>
<td>individual LEDs) wired to the card.</td>
</tr>
<tr>
<td>DoaSend7Seg8774067</td>
<td>Sends data to a DOA_877_4067 daughtercard to control the 7-segment LCDs</td>
</tr>
<tr>
<td></td>
<td>(or individual LEDs) wired to the card.</td>
</tr>
<tr>
<td>DoaSend8ServoCalibration</td>
<td>Sends data to a DOA_servo daughtercard to control the calibration of an</td>
</tr>
<tr>
<td></td>
<td>individual servo wired to the card.</td>
</tr>
<tr>
<td>DoaSend8ServoGain</td>
<td>Sends data to a DOA_servo daughtercard to control the gain parameter of an</td>
</tr>
<tr>
<td></td>
<td>individual servo wired to the card.</td>
</tr>
<tr>
<td>DoaSend8ServoPosition</td>
<td>Sends data to a DOA_servo daughtercard to control the position of an</td>
</tr>
<tr>
<td></td>
<td>individual servo wired to the card.</td>
</tr>
<tr>
<td>DoaSendAirCoreMotor</td>
<td></td>
</tr>
<tr>
<td>Method</td>
<td>Description</td>
</tr>
<tr>
<td>-------------------------------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>DoaSendAnOut1</td>
<td>Sends data to a DOA_AnOut1 analog output daughtercard, to control the RMS voltage (using PWM) being supplied by that channel.</td>
</tr>
<tr>
<td>DoaSendAnOut1GainAllChannels</td>
<td>Sends data to an individual HD44780-compatible character control the gain parameter which is in effect for all channels simultaneously.</td>
</tr>
<tr>
<td>DoaSendCharLcd</td>
<td>Sends data to an individual HD44780-compatible character LCD display wired to a PHCC DOA_char_lcd character LCD driver daughtercard.</td>
</tr>
<tr>
<td>DoaSendRaw</td>
<td>Sends data to a Digital Output Type A (DOA) peripheral attached to the PHCC motherboard.</td>
</tr>
<tr>
<td>DoaSendStepperMotor</td>
<td>Sends data to a stepper motor daughtercard.</td>
</tr>
<tr>
<td>DobSendRaw</td>
<td>Sends data to a Digital Output Type B (DOB) peripheral attached to the PHCC motherboard.</td>
</tr>
<tr>
<td>Equals</td>
<td>(Inherited from Object.)</td>
</tr>
<tr>
<td>Finalize</td>
<td>(Inherited from Object.)</td>
</tr>
<tr>
<td>GetHashCode</td>
<td>(Inherited from Object.)</td>
</tr>
<tr>
<td>GetLifetimeService</td>
<td>(Inherited from MarshalByRefObject.)</td>
</tr>
<tr>
<td>GetType</td>
<td>(Inherited from Object.)</td>
</tr>
<tr>
<td>I2CSend</td>
<td>Sends data to an I2C peripheral attached to the PHCC motherboard. (Inherited from MarshalByRefObject.)</td>
</tr>
<tr>
<td>InitializeLifetimeService</td>
<td>Instructs the PHCC motherboard to perform a software reset on itself. (Inherited from MarshalByRefObject.)</td>
</tr>
<tr>
<td>MemberwiseClone</td>
<td></td>
</tr>
<tr>
<td>Reset</td>
<td></td>
</tr>
</tbody>
</table>
- **SetIdle**
  Instructs the PHCC motherboard to enter the IDLE state.

- **StartTalking**
  Informs the PHCC motherboard to start sending automatic change notification events.

- **StopTalking**
  Informs the PHCC motherboard to stop sending change notification events

- **ToString**
  (Inherited from **Object**)
## Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>AnalogInputs</td>
<td>Gets an array of 16-bit signed integers containing the current values of all analog inputs. Only the low 10 bits contain information; the high 6 bits are always zero because the precision of the analog inputs is currently limited to 10 bits.</td>
</tr>
<tr>
<td>DigitalInputs</td>
<td>Gets a bool array containing the current values of all digital inputs.</td>
</tr>
<tr>
<td>FirmwareVersion</td>
<td>Gets a string containing the PHCC motherboard's firmware version.</td>
</tr>
<tr>
<td>PortName</td>
<td>Gets/sets the name of the COM port to use for communicating with the PHCC motherboard (i.e. &quot;COM1&quot;, &quot;COM2&quot;, etc.)</td>
</tr>
<tr>
<td>SerialPort</td>
<td>Gets the underlying SerialPort object, which allows direct communication with the PHCC motherboard via RS232.</td>
</tr>
</tbody>
</table>
# Events

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>AnalogInputChanged</strong></td>
<td>The AnalogInputChanged event is raised when the PHCC motherboard detects that one of the analog inputs has changed values (i.e. whenever an analog input signal changes state).</td>
</tr>
<tr>
<td><strong>DigitalInputChanged</strong></td>
<td>The DigitalInputChanged event is raised when the PHCC motherboard detects that one of the digital inputs has changed (i.e. whenever a button that is wired into the digital input key matrix is pressed or released).</td>
</tr>
<tr>
<td><strong>I2CDataReceived</strong></td>
<td>The I2CDataReceived event is raised when the PHCC motherboard receives data from one of the attached I2C peripherals (if any).</td>
</tr>
</tbody>
</table>
See Also

Device Class
Phcc Namespace

Send feedback on this topic to Microsoft.
The **Device** type exposes the following members.
# Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CharTo7Seg</td>
<td>(Inherited from MarshalByRefObject.)</td>
</tr>
<tr>
<td>CreateObjRef</td>
<td>Public implementation of IDisposable.Dispose(). Cleans up managed and unmanaged resources used by this object before allowing garbage collection</td>
</tr>
<tr>
<td>Dispose</td>
<td>Sends data to a DOA_40DO digital output daughtercard. Sends data to a DOA_7Seg daughtercard to control the 7-segment LCDs (or individual LEDs) wired to the card. Sends data to a DOA_877_4067 daughtercard to control the 7-segment LCDs (or individual LEDs) wired to the card. Sends data to a DOA_servo daughtercard to control the calibration of an individual servo wired to the card. Sends data to a DOA_servo daughtercard to control the gain parameter of an individual servo wired to the card. Sends data to a DOA_servo daughtercard to control the position of an individual servo wired to the card.</td>
</tr>
<tr>
<td>DoaSend40DO</td>
<td></td>
</tr>
<tr>
<td>DoaSend7Seg</td>
<td></td>
</tr>
<tr>
<td>DoaSend7Seg8774067</td>
<td></td>
</tr>
<tr>
<td>DoaSend8ServoCalibration</td>
<td></td>
</tr>
<tr>
<td>DoaSend8ServoGain</td>
<td></td>
</tr>
<tr>
<td>DoaSend8ServoPosition</td>
<td></td>
</tr>
<tr>
<td>DoaSendAirCoreMotor</td>
<td></td>
</tr>
</tbody>
</table>
- **DoaSendAnOut1** Sends data to a DOA_AnOut1 analog output daughtercard, to control the RMS voltage (using PWM) being supplied by that channel.

- **DoaSendAnOut1GainAllChannels** Sends data to a DOA_AnOut1 analog output daughtercard, to control the gain parameter which is in effect for all channels simultaneously.

- **DoaSendCharLcd** Sends data to an individual HD44780-compatible character LCD display wired to a PHCC DOA_char_lcd character LCD driver daughtercard.

- **DoaSendRaw** Sends data to a Digital Output Type A (DOA) peripheral attached to the PHCC motherboard.

- **DoaSendStepperMotor** Sends data to a stepper motor daughtercard.

- **DobSendRaw** Sends data to a Digital Output Type B (DOB) peripheral attached to the PHCC motherboard.

- **Equals** (Inherited from **Object**.)

- **Finalize** (Inherited from **Object**.)

- **GetHashCode** (Inherited from **Object**.)

- **GetLifetimeService** (Inherited from **MarshalByRefObject**.)

- **GetType** (Inherited from **Object**.)

- **I2CSend** Sends data to an I2C peripheral attached to the PHCC motherboard.

- **InitializeLifetimeService** (Inherited from **MarshalByRefObject**.)

- **MemberwiseClone** Overloaded.

- **Reset** Instructs the PHCC motherboard to perform a software reset on itself.
- **SetIdle**
  Instructs the PHCC motherboard to enter the IDLE state.

- **StartTalking**
  Informs the PHCC motherboard to start sending automatic change notification events.

- **StopTalking**
  Informs the PHCC motherboard to stop sending change notification events.

- **ToString**
  (Inherited from **Object**.)
See Also

Device Class
Phcc Namespace

Send feedback on this topic to Microsoft.
Phcc Interface Library for .NET and COM v0.2.1.0 reference library

Device:::.CharTo7Seg Method

Namespace:  Phcc
Assembly:  Phcc (in Phcc.dll)
Syntax

Visual Basic (Declaration)

Public Function CharTo7Seg ( _
    charToConvert As Char _
) As Byte

C#

public byte CharTo7Seg(
    char charToConvert
)

Parameters

charToConvert
    Type: System..:::Char
See Also

Device Class
Phcc Namespace

Send feedback on this topic to Microsoft.
Visual Basic  C#
Phcc Interface Library for .NET and COM v0.2.1.0 reference library
Device... Dispose Method

Public implementation of IDisposable.Dispose(). Cleans up managed and unmanaged resources used by this object before allowing garbage collection

Namespace: Phcc
Assembly: Phcc (in Phcc.dll)
Syntax

Visual Basic (Declaration)
Public Sub Dispose

C#
public void Dispose()

Implements

IDisposable...:::Dispose()()
See Also

Device Class
Phcc Namespace

Send feedback on this topic to Microsoft.
Sends data to a DOA_40DO digital output daughtercard.

**Namespace:** Phcc  
**Assembly:** Phcc (in Phcc.dll)
Syntax

Visual Basic (Declaration)

Public Sub DoaSend40DO ( _
    deviceAddr As Byte, _
    connectorNum As Byte, _
    bits As Byte _
)

C#

public void DoaSend40DO(
    byte deviceAddr,
    byte connectorNum,
    byte bits
)

Parameters

deviceAddr
    Type: System:::Byte
    The device address of the specific DOA_40DO daughtercard to send data to.

connectorNum
    Type: System:::Byte
    The output connector number to send data to (3=CON3, 4=CON4, 5=CON5, 6=CON6, 7=CON7)

bits
    Type: System:::Byte
    A byte, whose bits correspond to the pins on the specified connector. Each bit in this byte that is set to TRUE will result in a logic HIGH on the corresponding output pin on the specified connector.
See Also

Device Class
Phcc Namespace

Send feedback on this topic to Microsoft.
Phcc Interface Library for .NET and COM v0.2.1.0 reference library
Device...::DoaSend7Seg Method

Device Class  See Also  Send Feedback

Sends data to a DOA_7Seg daughtercard to control the 7-segment LCDs (or individual LEDs) wired to the card.

Namespace:  Phcc
Assembly:  Phcc (in Phcc.dll)
Syntax

Visual Basic (Declaration)

Public Sub DoaSend7Seg (_
    deviceAddr As Byte, _
    displayNum As Byte, _
    bits As Byte _
)

C#

public void DoaSend7Seg(
    byte deviceAddr,
    byte displayNum,
    byte bits
)

Parameters

deviceAddr
    Type: System::Byte
    The device address of the specific DOA_7Seg daughtercard to send data to.

displayNum
    Type: System::Byte
    The number of the 7-segment display (1-32), on the specified daughtercard, that the bits parameter will control.

bits
    Type: System::Byte
    A byte, whose bits correspond to individual segments of the specified 7-segment display (including the decimal point). Each bit in this byte that is set to will result in a logic HIGH being sent to the corresponding segment (or LED).
See Also

Device Class
Phcc Namespace

Send feedback on this topic to Microsoft.
Phcc Interface Library for .NET and COM v0.2.1.0 reference library

Device...:.DoaSend7Seg8774067 Method

Device Class  See Also  Send Feedback

Sends data to a DOA_877_4067 daughtercard to control the 7-segment LCDs (or individual LEDs) wired to the card.

Namespace:  Phcc
Assembly:  Phcc (in Phcc.dll)
Syntax

**Visual Basic (Declaration)**

```vbnet
Public Sub DoaSend7Seg8774067 ( 
    deviceAddr As Byte, _
    displayNum As Byte, _
    bits As Byte _
) 
```

**C#**

```csharp
public void DoaSend7Seg8774067(
    byte deviceAddr,
    byte displayNum,
    byte bits
)
```

**Parameters**

deviceAddr
Type: `System::Byte`
The device address of the specific DOA_877_4067 daughtercard to send data to.

displayNum
Type: `System::Byte`
The number of the 7-segment display (1-48), on the specified daughtercard, that the bits parameter will control.

bits
Type: `System::Byte`
A byte, whose bits correspond to individual segments of the specified 7-segment display (including the decimal point). Each bit in this byte that is set to will result in a logic HIGH being sent to the corresponding segment (or LED).
See Also

Device Class
Phcc Namespace

Send feedback on this topic to Microsoft.
Phcc Interface Library for .NET and COM v0.2.1.0 reference library

Device...:.DoaSend8ServoCalibration Method

**Device Class**  **See Also**  **Send Feedback**

Sends data to a DOA_servo daughtercard to control the calibration of an individual servo wired to the card.

**Namespace:**  Phcc

**Assembly:**  Phcc (in Phcc.dll)
Syntax

Visual Basic (Declaration)

Public Sub DoaSend8ServoCalibration(_
    deviceAddr As Byte, _
    servoNum As Byte, _
    calibrationOffset As Short _
)

C#

public void DoaSend8ServoCalibration(
    byte deviceAddr,
    byte servoNum,
    short calibrationOffset
)

Parameters

deviceAddr
    Type: System::Byte
    The device address of the specific DOA_servo daughtercard to send data to.

servoNum
    Type: System::Byte
    The number of the specific servo on the DOA_servo daughtercard, to which this calibration data will apply.

calibrationOffset
    Type: System::Int16
    The new 16-bit calibration offset value to use with this specific servo.
See Also

Device Class
Phcc Namespace

Send feedback on this topic to Microsoft.
Phcc Interface Library for .NET and COM v0.2.1.0 reference library

Device...: DoaSend8ServoGain Method

Device Class  See Also  Send Feedback

Sends data to a DOA_servo daughtercard to control the gain parameter of an individual servo wired to the card.

Namespace: Phcc
Assembly: Phcc (in Phcc.dll)
Syntax

Visual Basic (Declaration)

Public Sub DoaSend8ServoGain( _
    deviceAddr As Byte, _
    servoNum As Byte, _
    gain As Byte _
)

C#

public void DoaSend8ServoGain(
    byte deviceAddr,
    byte servoNum,
    byte gain
)

Parameters

deviceAddr
Type: System::Byte
The device address of the specific DOA_servo daughtercard to send data to.

servoNum
Type: System::Byte
The number of the specific servo on the DOA_servo daughtercard, to which this gain parameter value will be applied.

gain
Type: System::Byte
The new gain value to use for the specified servo.
See Also

Device Class
Phcc Namespace

Send feedback on this topic to Microsoft.
Device::DoaSend8ServoPosition Method

Sends data to a DOA_servo daughtercard to control the position of an individual servo wired to the card.

**Namespace:** Phcc
**Assembly:** Phcc (in Phcc.dll)
Syntax

Visual Basic (Declaration)

```vbnet
Public Sub DoaSend8ServoPosition(_
    deviceAddr As Byte, _
    servoNum As Byte, _
    position As Byte)
```

C#

```csharp
public void DoaSend8ServoPosition(
    byte deviceAddr,
    byte servoNum,
    byte position
)
```

Parameters

deviceAddr
Type: System..::.Byte
The device address of the specific DOA_servo daughtercard to send data to.

servoNum
Type: System..::.Byte
The number of the specific servo on the DOA_servo daughtercard, to which this position update will be sent.

position
Type: System..::.Byte
The position value to set this servo to.
See Also

Device Class
Phcc Namespace

Send feedback on this topic to Microsoft.
Phcc Interface Library for .NET and COM v0.2.1.0 reference library

Device...::DoaSendAirCoreMotor Method

Device Class  See Also  Send Feedback

Namespace:  Phcc
Assembly:  Phcc (in Phcc.dll)
Syntax

Visual Basic (Declaration)

Public Sub DoaSendAirCoreMotor ( _
    deviceAddr As Byte, _
    motorNum As Byte, _
    position As Integer _
)

C#

public void DoaSendAirCoreMotor(
    byte deviceAddr,
    byte motorNum,
    int position
)

Parameters

deviceAddr
    Type: System::Byte

motorNum
    Type: System::Byte

position
    Type: System::Int32
See Also

Device Class
Phcc Namespace

Send feedback on this topic to Microsoft.
Sends data to a DOA_AnOut1 analog output daughtercard, to control the RMS voltage (using PWM) being supplied by that channel.

**Namespace:** Phcc

**Assembly:** Phcc (in Phcc.dll)
**Syntax**

**Visual Basic (Declaration)**

```vbnet
Public Sub DoaSendAnOut1 ( _
    deviceAddr As Byte, _
    channelNum As Byte, _
    value As Byte _
)
```

**C#**

```csharp
public void DoaSendAnOut1(
    byte deviceAddr, 
    byte channelNum, 
    byte value
)
```

**Parameters**

deviceAddr
   Type: System::Byte
   The device address of the specific DOA_AnOut1 daughtercard to send data to.

channelNum
   Type: System::Byte
   The channel number of the channel on the DOA_AnOut1 daughtercard whose PWM output voltage is to be set.

value
   Type: System::Byte
   A byte whose integer value controls the PWM pulse width (delay time), which, in turn, dictates the RMS (average) voltage between the control pins on the specified channel.
See Also

Device Class
Phcc Namespace

Send feedback on this topic to Microsoft.
Phcc Interface Library for .NET and COM v0.2.1.0 reference library
Device::.DoaSendAnOut1GainAllChannels Method

Device Class  See Also  Send Feedback

Sends data to a DOA_AnOut1 analog output daughtercard, to control the gain parameter which is in effect for all channels simultaneously.

Namespace:  Phcc
Assembly:  Phcc (in Phcc.dll)
Syntax

Visual Basic (Declaration)

Public Sub DoaSendAnOut1GainAllChannels ( _
    deviceAddr As Byte, _
    gain As Byte _
)

C#

public void DoaSendAnOut1GainAllChannels(
    byte deviceAddr,
    byte gain
)

Parameters

deviceAddr
    Type: System::Byte
    The device address of the specific DOA_AnOut1 daughtercard to send data to.

gain
    Type: System::Byte
    A byte whose integer value specifies the new value to set for the all-channels gain parameter.
See Also

Device Class
Phcc Namespace

Send feedback on this topic to Microsoft.
Sends data to an individual HD44780-compatible character LCD display wired to a PHCC DOA_char_lcd character LCD driver daughtercard.

**Namespace:** Phcc  
**Assembly:** Phcc (in Phcc.dll)
Syntax

Visual Basic (Declaration)

Public Sub DoaSendCharLcd ( _
    deviceAddr As Byte, _
    displayNum As Byte, _
    mode As LcdDataModes, _
    data As Byte _
)

C#

public void DoaSendCharLcd(
    byte deviceAddr,
    byte displayNum,
    LcdDataModes mode,
    byte data
)

Parameters

deviceAddr
Type: System:: Byte
The device address of the specific DOA_char_lcd character LCD driver daughtercard to send data to.

displayNum
Type: System:: Byte
The display number of the individual HD44780-compatible character LCD wired to the indicated DOA_char_lcd character LCD driver daughtercard, to which, the specified data will be sent.

mode
Type: Phcc::LcdDataModes
A value from the LcdDataModes enumeration, specifying whether the value contained in the data parameter is to be considered Display Data or Control Data.
data

Type: System::Byte

The actual data value to send to the indicated HD44780-compatible character LCD display.
See Also

Device Class
Phcc Namespace

Send feedback on this topic to Microsoft.
Phcc Interface Library for .NET and COM v0.2.1.0 reference library

Device...:.DoaSendRaw Method

Sends data to a Digital Output Type A (DOA) peripheral attached to the PHCC motherboard.

**Namespace:** Phcc

**Assembly:** Phcc (in Phcc.dll)
**Syntax**

**Visual Basic (Declaration)**

```vbnet
Public Sub DoaSendRaw ( _
    addr As Byte, _
    subAddr As Byte, _
    data As Byte _
)
```

**C#**

```csharp
public void DoaSendRaw(
    byte addr,
    byte subAddr,
    byte data
)
```

**Parameters**

**addr**
Type: `System::Byte`
The device address of the specific Digital Output Type A (DOA) peripheral to send data to.

**subAddr**
Type: `System::Byte`
The sub-address of the Digital Output Type A (DOA) peripheral to send data to.

**data**
Type: `System::Byte`
The data to send to the specified Output Type A (DOA) peripheral.
See Also

Device Class
Phcc Namespace

Send feedback on this topic to Microsoft.
Sends data to a stepper motor daughtercard.

**Namespace:**  Phcc  
**Assembly:**  Phcc (in Phcc.dll)
Syntax

Visual Basic (Declaration)

Public Sub DoaSendStepperMotor ( _
    deviceAddr As Byte, _
    motorNum As Byte, _
    direction As MotorDirections, _
    numSteps As Byte, _
    stepType As MotorStepTypes _
)

C#

public void DoaSendStepperMotor(
    byte deviceAddr,
    byte motorNum,
    MotorDirections direction,
    byte numSteps,
    MotorStepTypes stepType
)

Parameters

deviceAddr
    Type: System..::.Byte
    The device address of the specific stepper motor daughtercard to send data to.

motorNum
    Type: System..::.Byte
    The motor number (1-4) of the motor to control.

direction
    Type: Phcc..::.MotorDirections
    A value from the MotorDirections enumeration, indicating the direction (clockwise or counterclockwise) to move the motor (this ultimately depends on how the motor is wired to the card).
numSteps
   Type:  System::Byte
   A byte, whose value (0-127) represents the number of discrete steps to
   command the stepper motor to move, in the indicated direction.

stepType
   Type:  Phcc::MotorStepTypes
   A value from the MotorStepTypes enumeration, indicating whether to move
   the motor in full-steps or in half-steps.
See Also

Device Class
Phcc Namespace

Send feedback on this topic to Microsoft.
Phcc Interface Library for .NET and COM v0.2.1.0 reference library

Device...::DobSendRaw Method

Device Class  See Also  Send Feedback

Sends data to a Digital Output Type B (DOB) peripheral attached to the PHCC motherboard.

Namespace:  Phcc
Assembly:  Phcc (in Phcc.dll)
Syntax

Visual Basic (Declaration)

Public Sub DobSendRaw ( _
    addr As Byte, _
    data As Byte _
)

C#

public void DobSendRaw(
    byte addr,
    byte data
)

Parameters

addr
Type: System::Byte
The address of the Digital Output Type B (DOB) peripheral to send data to.

data
Type: System::Byte
The data to send to the specified Digital Output Type B (DOB) peripheral.
See Also

Device Class
Phcc Namespace

Send feedback on this topic to Microsoft.
Phcc Interface Library for .NET and COM v0.2.1.0 reference library
Device::.I2CSend Method

Sends data to an I2C peripheral attached to the PHCC motherboard.

**Namespace:** Phcc
**Assembly:** Phcc (in Phcc.dll)
Syntax

Visual Basic (Declaration)

Public Sub I2CSend ( _
    addr As Byte, _
    subAddr As Byte, _
    data As Byte _
)

C#

public void I2CSend(
    byte addr,
    byte subAddr,
    byte data
)

Parameters

addr
Type: System..::.Byte
The address of the I2C peripheral to send data to.

subAddr
Type: System..::.Byte
The sub-address of the I2C peripheral to send data to.

data
Type: System..::.Byte
The data to send to the specified I2C peripheral.
See Also

Device Class
Phcc Namespace

Send feedback on this topic to Microsoft.
Visual Basic  □  C#

□ Include Protected Members
□ Include Inherited Members

Phcc Interface Library for .NET and COM v0.2.1.0 reference library

Device...::.MemberwiseClone Method

Device Class  See Also  Send Feedback
## Overload List

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>MemberwiseClone()</code></td>
<td>(Inherited from <a href="#">Object</a>)</td>
</tr>
<tr>
<td><code>MemberwiseClone(Boolean)</code></td>
<td>(Inherited from <a href="#">MarshalByRefObject</a>)</td>
</tr>
</tbody>
</table>
See Also

Device Class
Device Members
Phcc Namespace

Send feedback on this topic to Microsoft.
Device Reset Method

Instructs the PHCC motherboard to perform a software reset on itself.

Namespace: Phcc
Assembly: Phcc (in Phcc.dll)
Syntax

Visual Basic (Declaration)
Public Sub Reset

C#

public void Reset()
See Also

Device Class
Phcc Namespace

Send feedback on this topic to Microsoft.
Phcc Interface Library for .NET and COM v0.2.1.0 reference library
Device...: SetIdle Method

Device Class  See Also  Send Feedback

Instructs the PHCC motherboard to enter the IDLE state.

Namespace:  Phcc
Assembly:  Phcc (in Phcc.dll)
Syntax

Visual Basic (Declaration)

Public Sub SetIdle

C#

public void SetIdle()
See Also

Device Class
Phcc Namespace

Send feedback on this topic to Microsoft.
Phcc Interface Library for .NET and COM v0.2.1.0 reference library
Device....StartTalking Method

**Device Class**  **See Also**  **Send Feedback**

Informs the PHCC motherboard to start sending automatic change notification events.

**Namespace:**  Phcc
**Assembly:**  Phcc (in Phcc.dll)
Syntax

Visual Basic (Declaration)

Public Sub StartTalking

C#

public void StartTalking()
See Also

Device Class
Phcc Namespace

Send feedback on this topic to Microsoft.
Phcc Interface Library for .NET and COM v0.2.1.0 reference library
Device....StopTalking Method

*Device Class*  See Also  Send Feedback

Informs the PHCC motherboard to stop sending change notification events

**Namespace:**  Phcc
**Assembly:**  Phcc (in Phcc.dll)
Syntax

Visual Basic (Declaration)

Public Sub StopTalking

C#

public void StopTalking()
See Also

Device Class
Phcc Namespace

Send feedback on this topic to Microsoft.
Visual Basic  C#
Include Protected Members
Include Inherited Members

Phcc Interface Library for .NET and COM v0.2.1.0 reference library
Device Properties

Device Class  See Also  Send Feedback

The Device type exposes the following members.
<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>AnalogInputs</td>
<td>Gets an array of 16-bit signed integers containing the current values of all analog inputs. Only the low 10 bits contain information; the high 6 bits are always zero because the precision of the analog inputs is currently limited to 10 bits.</td>
</tr>
<tr>
<td>DigitalInputs</td>
<td>Gets a bool array containing the current values of all digital inputs.</td>
</tr>
<tr>
<td>FirmwareVersion</td>
<td>Gets a string containing the PHCC motherboard's firmware version.</td>
</tr>
<tr>
<td>PortName</td>
<td>Gets/sets the name of the COM port to use for communicating with the PHCC motherboard (i.e. &quot;COM1&quot;, &quot;COM2&quot;, etc.)</td>
</tr>
<tr>
<td>SerialPort</td>
<td>Gets the underlying SerialPort object, which allows direct communication with the PHCC motherboard via RS232.</td>
</tr>
</tbody>
</table>
See Also

Device Class
Phcc Namespace

Send feedback on this topic to Microsoft.
Gets an array of 16-bit signed integers containing the current values of all analog inputs. Only the low 10 bits contain information; the high 6 bits are always zero because the precision of the analog inputs is currently limited to 10 bits.

**Namespace:** Phcc

**Assembly:** Phcc (in Phcc.dll)
Syntax

Visual Basic (Declaration)

Public ReadOnly Property AnalogInputs As Short()

C#

public short[] AnalogInputs { get; }

Return Value

An array of 16-bit signed integers containing the current values of all analog inputs. Only the low 10 bits in each array element contain useful information; the high 6 bits are always zero because the precision of the PHCC analog inputs is currently limited to 10 bits.
See Also

Device Class
Phcc Namespace

Send feedback on this topic to Microsoft.
Gets a bool array containing the current values of all digital inputs.

Namespace: Phcc
Assembly: Phcc (in Phcc.dll)
Syntax

Visual Basic (Declaration)

Public ReadOnly Property DigitalInputs As Boolean()

C#

public bool[] DigitalInputs { get; }

Return Value

A bool array containing the current values of all digital inputs. Each value in the array represents a single discrete digital input, out of a total of 1024 inputs.
See Also

Device Class
Phcc Namespace

Send feedback on this topic to Microsoft.
Gets a string containing the PHCC motherboard's firmware version.

**Namespace:** Phcc  
**Assembly:** Phcc (in Phcc.dll)
Syntax

Visual Basic (Declaration)
Public ReadOnly Property FirmwareVersion As String

C#

public string FirmwareVersion { get; }

Return Value

A String containing the PHCC motherboard's firmware version.
See Also

Device Class
Phcc Namespace

Send feedback on this topic to Microsoft.
Phcc Interface Library for .NET and COM v0.2.1.0 reference library

Device..::.PortName Property

Namespace: Phcc  
Assembly: Phcc (in Phcc.dll)

Gets/sets the name of the COM port to use for communicating with the PHCC motherboard (i.e. "COM1", "COM2", etc.)
Syntax

Visual Basic (Declaration)

Public Property PortName As String

C#

public string PortName { get; set; }
See Also

Device Class
Phcc Namespace

Send feedback on this topic to Microsoft.
Phcc Interface Library for .NET and COM v0.2.1.0 reference library

Device....SerialPort Property

Gets the underlying **SerialPort** object, which allows direct communication with the PHCC motherboard via RS232.

**Namespace:** Phcc

**Assembly:** Phcc (in Phcc.dll)
Syntax

Visual Basic (Declaration)

Public ReadOnly Property SerialPort As SerialPort

C#

public SerialPort SerialPort { get; }
See Also

Device Class
Phcc Namespace

Send feedback on this topic to Microsoft.
DigitalInputChangedEventArgs objects hold data that the PHCC motherboard provides whenever a digital input value changes. The `DigitalInputChanged` event provides `DigitalInputChangedEventArgs` event-args objects during the raising of each event.

**Namespace:** Phcc  
**Assembly:** Phcc (in Phcc.dll)
Syntax

Visual Basic (Declaration)

```vbnet
<ClassInterfaceAttribute(ClassInterfaceType.AutoDual)> _
Public NotInheritable Class DigitalInputChangedEventArgs _
    Inherits EventArgs
```

C#

```csharp
[ClassInterfaceAttribute(ClassInterfaceType.AutoDual)]
public sealed class DigitalInputChangedEventArgs : EventArgs
```
Inheritance Hierarchy

- `System::Object`
- `System::EventArgs`
  - Phcc::DigitalInputChangedEventArgs
See Also

DigitalInputChangedEventArgs Members
Phcc Namespace

Send feedback on this topic to Microsoft.
Visual Basic  C#
Include Protected Members
Include Inherited Members
Phcc Interface Library for .NET and COM v0.2.1.0 reference library
DigitalInputChangedEventArgs Constructor
DigitalInputChangedEventArgs Class
See Also
Send Feedback
## Overload List

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DigitalInputChangedEventArgs()()</td>
<td>Creates an instance of the <strong>DigitalInputChangedEventArgs</strong> class.</td>
</tr>
<tr>
<td>DigitalInputChangedEventArgs(Int16, Boolean)</td>
<td>Creates an instance of the <strong>DigitalInputChangedEventArgs</strong> class.</td>
</tr>
</tbody>
</table>
See Also

DigitalInputChangedEventArgs Class
DigitalInputChangedEventArgs Members
Phcc Namespace

Send feedback on this topic to Microsoft.
Phcc Interface Library for .NET and COM v0.2.1.0 reference library

DigitalInputChangedEventArgs Constructor

DigitalInputChangedEventArgs Class  See Also  Send Feedback

Creates an instance of the DigitalInputChangedEventArgs class.

Namespace:  Phcc
Assembly:  Phcc (in Phcc.dll)
Syntax

Visual Basic (Declaration)

Public Sub New

C#

public DigitalInputChangedEventArgs()
See Also

DigitalInputChangedEventArgs Class
DigitalInputChangedEventArgs Overload
Phcc Namespace

Send feedback on this topic to Microsoft.
Creates an instance of the `DigitalInputChangedEventArgs` class.

**Namespace:** Phcc  
**Assembly:** Phcc (in Phcc.dll)
Syntax

**Visual Basic (Declaration)**

```vbnet
Public Sub New ( _
    index As Short, _
    newValue As Boolean _
)
```

**C#**

```csharp
public DigitalInputChangedEventArgs(
    short index,
    bool newValue
)
```

**Parameters**

**index**

Type: `System::Int16`

The index of the digital input whose value has changed.

**newValue**

Type: `System::Boolean`

The new value of the digital input indicated by the Address parameter.
See Also

DigitalInputChangedEventArgs Class
DigitalInputChangedEventArgs Overload
Phcc Namespace

Send feedback on this topic to Microsoft.
The `DigitalInputChangedEventArgs` type exposes the following members.
## Constructors

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DigitalInputChangedEventArgs</td>
<td>Overloaded.</td>
</tr>
</tbody>
</table>
## Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Equals</td>
<td>(Inherited from <a href="#">Object</a>)</td>
</tr>
<tr>
<td>Finalize</td>
<td>(Inherited from <a href="#">Object</a>)</td>
</tr>
<tr>
<td>GetHashCode</td>
<td>(Inherited from <a href="#">Object</a>)</td>
</tr>
<tr>
<td>GetType</td>
<td>(Inherited from <a href="#">Object</a>)</td>
</tr>
<tr>
<td>MemberwiseClone</td>
<td>(Inherited from <a href="#">Object</a>)</td>
</tr>
<tr>
<td>ToString</td>
<td>(Inherited from <a href="#">Object</a>)</td>
</tr>
</tbody>
</table>
## Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Index</td>
<td>Gets/sets the index of the digital input whose value has changed.</td>
</tr>
<tr>
<td>NewValue</td>
<td>Gets/sets the new value of the indicated digital input.</td>
</tr>
</tbody>
</table>
See Also

DigitalInputChangedEventArgs Class
Phcc Namespace

Send feedback on this topic to Microsoft.
Phcc Interface Library for .NET and COM v0.2.1.0 reference library

DigitalInputChangedEventArgs Methods

DigitalInputChangedEventArgs Class  See Also  Send Feedback

The DigitalInputChangedEventArgs type exposes the following members.
## Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Equals</strong></td>
<td>(Inherited from Object.)</td>
</tr>
<tr>
<td><strong>Finalize</strong></td>
<td>(Inherited from Object.)</td>
</tr>
<tr>
<td><strong>GetHashCode</strong></td>
<td>(Inherited from Object.)</td>
</tr>
<tr>
<td><strong>GetType</strong></td>
<td>(Inherited from Object.)</td>
</tr>
<tr>
<td><strong>MemberwiseClone</strong></td>
<td>(Inherited from Object.)</td>
</tr>
<tr>
<td><strong>ToString</strong></td>
<td>(Inherited from Object.)</td>
</tr>
</tbody>
</table>
See Also

DigitalInputChangedEventArgs Class
Phcc Namespace

Send feedback on this topic to Microsoft.
The DigitalInputChangedEventArgs type exposes the following members.
## Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Index</td>
<td>Gets/sets the index of the digital input whose value has changed.</td>
</tr>
<tr>
<td>NewValue</td>
<td>Gets/sets the new value of the indicated digital input.</td>
</tr>
</tbody>
</table>
See Also

DigitalInputChangedEventArgs Class
Phcc Namespace

Send feedback on this topic to Microsoft.
Gets/sets the index of the digital input whose value has changed.

**Namespace:** Phcc

**Assembly:** Phcc (in Phcc.dll)
Syntax

Visual Basic (Declaration)

Public Property Index As Short

C#

public short Index { get; set; }
See Also

DigitalInputChangedEventArgs Class
Phcc Namespace

Send feedback on this topic to Microsoft.
Phcc Interface Library for .NET and COM v0.2.1.0 reference library

DigitalInputChangedEventArgs::NewValue Property

`DigitalInputChangedEventArgs` Class  See Also  Send Feedback

Gets/sets the new value of the indicated digital input.

**Namespace:**  Phcc

**Assembly:**  Phcc (in Phcc.dll)
Syntax

Visual Basic (Declaration)

Public Property NewValue As Boolean

C#

public bool NewValue { get; set; }
See Also

DigitalInputChangedEventArgs Class
Phcc Namespace

Send feedback on this topic to Microsoft.
Visual Basic  C#
Phcc Interface Library for .NET and COM v0.2.1.0 reference library
DigitalInputChangedEventHandler Delegate
See Also  Send Feedback

Event handler delegate for the DigitalInputChanged event.

Namespace:  Phcc
Assembly:  Phcc (in Phcc.dll)
Syntax

Visual Basic (Declaration)

<ComVisibleAttribute(False)> _
Public Delegate Sub DigitalInputChangedEventHandler ( _
    sender As Object, _
    e As DigitalInputChangedEventArgs _
)

C#

[ComVisibleAttribute(false)]
public delegate void DigitalInputChangedEventHandler(
    Object sender,
    DigitalInputChangedEventArgs e)

Parameters

sender
Type: System::Object
the object raising the event.

e
Type: Phcc::DigitalInputChangedEventArgs
a DigitalInputChangedEventArgs object containing detailed information about the event.
See Also

Phcc Namespace

Send feedback on this topic to Microsoft.
I2CDataReceivedEventArgs objects hold I2C data that is received when the PHCC motherboard signals that new I2C data has arrived. This data is provided by the I2CDataReceived event.

**Namespace:**  Phcc  
**Assembly:** Phcc (in Phcc.dll)
Syntax

Visual Basic (Declaration)

```
<ClassInterfaceAttribute(ClassInterfaceType.AutoDual)> _
Public NotInheritable Class I2CDataReceivedEventArgs _
    Inherits EventArgs
```

C#

```
[ClassInterfaceAttribute(ClassInterfaceType.AutoDual)]
public sealed class I2CDataReceivedEventArgs : EventArgs
```
Inheritance Hierarchy

System...:::Object
System...:::EventArgs
Phcc...:::I2CDataReceivedEventArgs
See Also

I2CDataReceivedEventArgs Members
Phcc Namespace

Send feedback on this topic to Microsoft.
Visual Basic  C#
Include Protected Members
Include Inherited Members
Phcc Interface Library for .NET and COM v0.2.1.0 reference library
I2CDataReceivedEventArgs Constructor
I2CDataReceivedEventArgs Class  See Also  Send Feedback
<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>I2CDataReceivedEventArgs()()()</td>
<td>Creates an instance of the <code>I2CDataReceivedEventArgs</code> class.</td>
</tr>
<tr>
<td>I2CDataReceivedEventArgs(Int16, Byte)</td>
<td>Creates an instance of the <code>I2CDataReceivedEventArgs</code> class.</td>
</tr>
</tbody>
</table>
See Also

I2CDataReceivedEventArgs Class
I2CDataReceivedEventArgs Members
Phcc Namespace

Send feedback on this topic to Microsoft.
Phcc Interface Library for .NET and COM v0.2.1.0 reference library
I2CDataReceivedEventArgs Constructor

I2CDataReceivedEventArgs Class  See Also  Send Feedback

Creates an instance of the I2CDataReceivedEventArgs class.

Namespace: Phcc
Assembly: Phcc (in Phcc.dll)
Syntax

Visual Basic (Declaration)
Public Sub New

C#

public I2CDataReceivedEventArgs()
See Also

I2CDataReceivedEventArgs Class
I2CDataReceivedEventArgs Overload
Phcc Namespace

Send feedback on this topic to Microsoft.
Phcc Interface Library for .NET and COM v0.2.1.0 reference library

I2CDataReceivedEventArgs Constructor (Int16, Byte)

I2CDataReceivedEventArgs Class  See Also  Send Feedback

Creates an instance of the I2CDataReceivedEventArgs class.

Namespace:  Phcc
Assembly:  Phcc (in Phcc.dll)
### Syntax

**Visual Basic (Declaration)**

```vbnet
Public Sub New ( 
    address As Short, _
    data As Byte _
)
```

**C#**

```csharp
public I2CDataReceivedEventArgs(
    short address,
    byte data
)
```

**Parameters**

- **address**
  - Type: `System:::Int16`
  - The address of the I2C device that is providing the data during this event.

- **data**
  - Type: `System:::Byte`
  - The data being provided by the I2C device during this event.
See Also

I2CDataReceivedEventArgs Class
I2CDataReceivedEventArgs Overload
Phcc Namespace

Send feedback on this topic to Microsoft.
The **I2CDataReceivedEventArgs** type exposes the following members.
## Constructors

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>I2CDataReceivedEventArgs</td>
<td>Overloaded.</td>
</tr>
</tbody>
</table>
## Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Equals</td>
<td>(Inherited from Object.)</td>
</tr>
<tr>
<td>Finalize</td>
<td>(Inherited from Object.)</td>
</tr>
<tr>
<td>GetHashCode</td>
<td>(Inherited from Object.)</td>
</tr>
<tr>
<td>GetType</td>
<td>(Inherited from Object.)</td>
</tr>
<tr>
<td>MemberwiseClone</td>
<td>(Inherited from Object.)</td>
</tr>
<tr>
<td>ToString</td>
<td>(Inherited from Object.)</td>
</tr>
</tbody>
</table>
### Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Address</td>
<td>Gets/sets the address of the I2C device that is providing the data during this event.</td>
</tr>
<tr>
<td>Data</td>
<td>Gets/sets the data being provided by the I2C device during this event.</td>
</tr>
</tbody>
</table>
See Also

I2CDataReceivedEventArgs Class
Phcc Namespace

Send feedback on this topic to Microsoft.
The `I2CDataReceivedEventArgs` type exposes the following members.
# Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Equals</td>
<td>(Inherited from <a href="#">Object</a>)</td>
</tr>
<tr>
<td>Finalize</td>
<td>(Inherited from <a href="#">Object</a>)</td>
</tr>
<tr>
<td>GetHashCode</td>
<td>(Inherited from <a href="#">Object</a>)</td>
</tr>
<tr>
<td>GetType</td>
<td>(Inherited from <a href="#">Object</a>)</td>
</tr>
<tr>
<td>MemberwiseClone</td>
<td>(Inherited from <a href="#">Object</a>)</td>
</tr>
<tr>
<td>ToString</td>
<td>(Inherited from <a href="#">Object</a>)</td>
</tr>
</tbody>
</table>
See Also

I2CDataReceivedEventArgs Class
Phcc Namespace

Send feedback on this topic to Microsoft.
Phcc Interface Library for .NET and COM v0.2.1.0 reference library

I2CDataReceivedEventArgs Properties

The **I2CDataReceivedEventArgs** type exposes the following members.
## Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Address</td>
<td>Gets/sets the address of the I2C device that is providing the data during this event.</td>
</tr>
<tr>
<td>Data</td>
<td>Gets/sets the data being provided by the I2C device during this event.</td>
</tr>
</tbody>
</table>
See Also

I2CDataReceivedEventArgs Class
Phcc Namespace

Send feedback on this topic to Microsoft.
I2CDataReceivedEventArgs.Address Property

Gets/sets the address of the I2C device that is providing the data during this event.

Namespace: Phcc
Assembly: Phcc (in Phcc.dll)
Syntax

Visual Basic (Declaration)
Public Property Address As Short

C#
public short Address { get; set; }
See Also

I2CDataReceivedEventArgs Class
Phcc Namespace

Send feedback on this topic to Microsoft.
Gets/sets the data being provided by the I2C device during this event.

**Namespace:** Phcc  
**Assembly:** Phcc (in Phcc.dll)
Syntax

Visual Basic (Declaration)

Public Property Data As Byte

C#

public byte Data { get; set; }
See Also

I2CDataReceivedEventArgs Class
Phcc Namespace

Send feedback on this topic to Microsoft.
Event handler delegate for the `I2CDataReceived` event.

**Namespace:** Phcc  
**Assembly:** Phcc (in Phcc.dll)
Syntax

Visual Basic (Declaration)

```vbnet
<ComVisibleAttribute(False)> _
Public Delegate Sub I2CDataReceivedEventHandler ( _
    sender As Object, _
    e As I2CDataReceivedEventArgs _
)
```

C#

```csharp
[ComVisibleAttribute(false)]
public delegate void I2CDataReceivedEventHandler(
    Object sender,
    I2CDataReceivedEventArgs e
)
```

Parameters

sender

Type: `System.Object`
the object raising the event.

e

Type: `Phcc.I2CDataReceivedEventArgs`
an `I2CDataReceivedEventArgs` object containing detailed information about the event.
See Also

Phcc Namespace

Send feedback on this topic to Microsoft.
Phcc Interface Library for .NET and COM v0.2.1.0 reference library

LcdDataModes Enumeration

See Also  Send Feedback

Enumeration of LCD data modes.

Namespace:  Phcc
Assembly:  Phcc (in Phcc.dll)
Syntax

Visual Basic (Declaration)

Public Enumeration LcdDataModes

C#

public enum LcdDataModes
# Members

<table>
<thead>
<tr>
<th>Member name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DisplayData</td>
<td>Specifies that the data being sent to the LCD is to be considered as Display Data</td>
</tr>
<tr>
<td>ControlData</td>
<td>Specifies that the data being sent to the LCD is to be considered as Control Data</td>
</tr>
</tbody>
</table>
See Also

Phcc Namespace

Send feedback on this topic to Microsoft.
Phcc Interface Library for .NET and COM v0.2.1.0 reference library
MotorDirections Enumeration

See Also  Send Feedback

Enumeration of stepper motor directions.

**Namespace:**  Phcc
**Assembly:**  Phcc (in Phcc.dll)
Syntax

Visual Basic (Declaration)

Public Enumeration MotorDirections

C#

public enum MotorDirections
## Members

<table>
<thead>
<tr>
<th>Member name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Clockwise</td>
<td>Specifies clockwise stepper motor movement.</td>
</tr>
<tr>
<td>Counterclockwise</td>
<td>Specifies counterclockwise stepper motor movement.</td>
</tr>
</tbody>
</table>
See Also

Phcc Namespace

Send feedback on this topic to Microsoft.
Enumeration of possible stepper motor step types.

**Namespace:**  Phcc

**Assembly:**  Phcc (in Phcc.dll)
Syntax

Visual Basic (Declaration)

Public Enumeration MotorStepTypes

C#

public enum MotorStepTypes
## Members

<table>
<thead>
<tr>
<th>Member name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>FullStep</td>
<td>Indicates that the step count refers to the number of full steps that the stepper motor should move.</td>
</tr>
<tr>
<td>HalfStep</td>
<td>Indicates that the step count refers to the number of half steps that the stepper motor should move.</td>
</tr>
</tbody>
</table>
See Also

Phcc Namespace

Send feedback on this topic to Microsoft.
COM Event Source Interface

**Namespace:** Phcc

**Assembly:** Phcc (in Phcc.dll)
Syntax

Visual Basic (Declaration)

```vbnet
<ComVisibleAttribute(True)> _
<GuidAttribute("8709CA5D-79FA-4a63-ACF4-C99475990BC3")> _
<InterfaceTypeAttribute(ComInterfaceType.InterfaceIsIDispatch)> _
Public Interface PhccEvents
```

C#

```csharp
[ComVisibleAttribute(true)]
[GuidAttribute("8709CA5D-79FA-4a63-ACF4-C99475990BC3")]
[InterfaceTypeAttribute(ComInterfaceType.InterfaceIsIDispatch)]
public interface PhccEvents
```
See Also

PhccEvents Members
Phcc Namespace

Send feedback on this topic to Microsoft.
The PhccEvents type exposes the following members.
## Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>AnalogInputChanged</td>
<td>The <code>AnalogInputChanged(Object, AnalogInputChangedEventArgs)</code> event is raised when the PHCC motherboard detects that one of the analog inputs has changed values (i.e. whenever an analog input signal changes state).</td>
</tr>
<tr>
<td>DigitalInputChanged</td>
<td>The <code>DigitalInputChanged(Object, DigitalInputChangedEventArgs)</code> event is raised when the PHCC motherboard detects that one of the digital inputs has changed (i.e. whenever a button that is wired into the digital input key matrix is pressed or released).</td>
</tr>
<tr>
<td>I2CDataReceived</td>
<td>The <code>I2CDataReceived(Object, I2CDataReceivedEventArgs)</code> event is raised when the PHCC motherboard receives data from one of the attached I2C peripherals (if any).</td>
</tr>
</tbody>
</table>

See Also

PhccEvents Interface
Phcc Namespace

Send feedback on this topic to Microsoft.
The PhccEvents type exposes the following members.
## Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>AnalogInputChanged</strong></td>
<td>The <a href="AnalogInputChangedEventArgs">AnalogInputChanged(Object, AnalogInputChangedEventArgs)</a> event is raised when the PHCC motherboard detects that one of the analog inputs has changed values (i.e. whenever an analog input signal changes state).</td>
</tr>
<tr>
<td><strong>DigitalInputChanged</strong></td>
<td>The <a href="DigitalInputChangedEventArgs">DigitalInputChanged(Object, DigitalInputChangedEventArgs)</a> event is raised when the PHCC motherboard detects that one of the digital inputs has changed (i.e. whenever a button that is wired into the digital input key matrix is pressed or released).</td>
</tr>
<tr>
<td><strong>I2CDataReceived</strong></td>
<td>The <a href="I2CDataReceivedEventArgs">I2CDataReceived(Object, I2CDataReceivedEventArgs)</a> event is raised when the PHCC motherboard receives data from one of the attached I2C peripherals (if any).</td>
</tr>
</tbody>
</table>
See Also

PhccEvents Interface
Phcc Namespace

Send feedback on this topic to Microsoft.
The AnalogInputChanged(Object, AnalogInputChangedEventArgs) event is raised when the PHCC motherboard detects that one of the analog inputs has changed values (i.e. whenever an analog input signal changes state).

**Namespace:** Phcc  
**Assembly:** Phcc (in Phcc.dll)
Syntax

Visual Basic (Declaration)

Sub AnalogInputChanged ( _
    sender As Object, _
    e As AnalogInputChangedEventArgs _
) _

C#

void AnalogInputChanged(
    Object sender,
    AnalogInputChangedEventArgs e
)

Parameters

sender
    Type: System:::Object

e
    Type: Phcc:::AnalogInputChangedEventArgs
See Also

PhccEvents Interface
Phcc Namespace

Send feedback on this topic to Microsoft.
Visual Basic  C#
Phcc Interface Library for .NET and COM v0.2.1.0 reference library
PhccEvents..::.DigitalInputChanged Method
PhccEvents Interface  See Also  Send Feedback

The DigitalInputChanged(Object, DigitalInputChangedEventArgs) event is raised when the PHCC motherboard detects that one of the digital inputs has changed (i.e. whenever a button that is wired into the digital input key matrix is pressed or released).

Namespace:  Phcc
Assembly:  Phcc (in Phcc.dll)
Syntax

Visual Basic (Declaration)

Sub DigitalInputChanged ( _
    sender As Object, _
    e As DigitalInputChangedEventArgs _
)

C#

void DigitalInputChanged(
    Object sender,
    DigitalInputChangedEventArgs e
)

Parameters

sender
    Type: System.Object

e
    Type: Phcc.DigitalInputChangedEventArgs
See Also

PhccEvents Interface
Phcc Namespace

Send feedback on this topic to Microsoft.
The `I2CDataReceived(Object, I2CDataReceivedEventArgs)` event is raised when the PHCC motherboard receives data from one of the attached I2C peripherals (if any).

**Namespace:** Phcc

**Assembly:** Phcc (in Phcc.dll)
Syntax

Visual Basic (Declaration)

Sub I2CDataReceived ( _
    sender As Object, _
    e As I2CDataReceivedEventArgs _
)

C#

void I2CDataReceived(
    Object sender,
    I2CDataReceivedEventArgs e
)

Parameters

sender
    Type: System:::Object

e
    Type: Phcc:::I2CDataReceivedEventArgs
See Also

PhccEvents Interface
Phcc Namespace

Send feedback on this topic to Microsoft.
Phcc Interface Library for .NET and COM v0.2.1.0 reference library

SevenSegmentBits Enumeration

See Also  Send Feedback

**Namespace:**  Phcc

**Assembly:**  Phcc (in Phcc.dll)
Syntax

Visual Basic (Declaration)

```vbnet
<FlagsAttribute> _
<ComVisibleAttribute>(True)> _
Public Enumeration SevenSegmentBits
```

C#

```csharp
[FlagsAttribute]
[ComVisibleAttribute(true)]
public enum SevenSegmentBits
```
## Members

<table>
<thead>
<tr>
<th>Member name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>None</td>
<td></td>
</tr>
<tr>
<td>SegmentA</td>
<td></td>
</tr>
<tr>
<td>SegmentB</td>
<td></td>
</tr>
<tr>
<td>SegmentC</td>
<td></td>
</tr>
<tr>
<td>SegmentD</td>
<td></td>
</tr>
<tr>
<td>SegmentE</td>
<td></td>
</tr>
<tr>
<td>SegmentF</td>
<td></td>
</tr>
<tr>
<td>SegmentG</td>
<td></td>
</tr>
<tr>
<td>SegmentDP</td>
<td></td>
</tr>
<tr>
<td>All</td>
<td></td>
</tr>
</tbody>
</table>
See Also

Phcc Namespace

Send feedback on this topic to Microsoft.