

IO.Devices.AD5593R Namespace

AD5593 Analog/Digital I/O Device Services.

▪ Classes

	Class	Description
	Device	Encapsulates the AD5593R I ² C Analog/Digital I/O device.

▪ Enumerations

	Enumeration	Description
	PinMode	AD5593R I/O Pin Modes.
	ReferenceMode	ADC5593R ADC and DAC reference settings.

Device Class

Encapsulates the AD5593R I²C Analog/Digital I/O device.

▪ Inheritance Hierarchy

[SystemObject](#) [IO.Devices.AD5593RDevice](#)

Namespace: [IO.Devices.AD5593R](#)

Assembly: libremoteio (in libremoteio.dll) Version: 2.2020.136.3

▪ Syntax

[C#](#) [VB](#) [F#](#)

[Copy](#)

```
public class Device
```

The [Device](#) type exposes the following members.

▪ Constructors

	Name	Description
	Device	Constructor for a single AD5593R device.

[Top](#)

▪ Properties

	Name	Description
	ADC_Reference	Write-only property for setting the AD5593R ADC reference

mode.

	DAC_Reference	Write-only property for setting the AD5593R DAC reference mode.
	GPIO_Inputs	GPIO input register state. Any I/O pin that is not configured as a GPIO input will read as zero.
	GPIO_Outputs	GPIO output register state. Any I/O pin that is not configured as a GPIO output will be written as zero.

[Top](#)

◀ Methods

	Name	Description
	ADC_Create	Create an AD5593R ADC input object.
	ConfigureChannel	Configure a single AD5593R I/O pin.
	DAC_Create	Create an AD5593R DAC output object.
	GPIO_Create	Create an AD5593R GPIO pin object.
	Read_ADC	Read from an ADC channel.
	Write_DAC	Write to a DAC channel.

[Top](#)

Fields

Name	Description
  ADC_Resolution	ADC resolution in bits.
  DAC_Resolution	DAC resolution in bits.
  MaxChannel	Maximum I/O channel number.
  MinChannel	Minimum I/O channel number.

[Top](#)

See Also

Reference

[IO.Devices.AD5593R Namespace](#)

Device Constructor

Constructor for a single AD5593R device.

Namespace: [IO.Devices.AD5593R](#)

Assembly: libremoteio (in libremoteio.dll) Version: 2.2020.136.3

▪ Syntax

C# VB F#

[Copy](#)

```
public Device(  
    Bus bus,  
    int addr  
)
```

Parameters

bus

Type: [IO.Interfaces.I2CBus](#)

I²C bus controller object.

addr

Type: [SystemInt32](#)

I²C slave address.

▪ See Also

[Reference](#)

[Device Class](#)

[IO.Devices.AD5593R Namespace](#)

Device Properties

The [Device](#) type exposes the following members.

Properties

	Name	Description
	ADC_Reference	Write-only property for setting the AD5593R ADC reference mode.
	DAC_Reference	Write-only property for setting the AD5593R DAC reference mode.
	GPIO_Inputs	GPIO input register state. Any I/O pin that is not configured as a GPIO input will read as zero.
	GPIO_Outputs	GPIO output register state. Any I/O pin that is not configured as a GPIO output will be written as zero.

[Top](#)

See Also

[Reference](#)

[Device Class](#)

IO.Devices.AD5593R Namespace

DeviceADC_Reference Property

Write-only property for setting the AD5593R ADC reference mode.

Namespace: [IO.Devices.AD5593R](#)

Assembly: libremoteio (in libremoteio.dll) Version: 2.2020.136.3

◀ Syntax

C# VB F#

[Copy](#)

```
public ReferenceMode ADC_Reference { set; }
```

Property Value

Type: [ReferenceMode](#)

◀ See Also

Reference

[Device Class](#)

[IO.Devices.AD5593R Namespace](#)

DeviceDAC_Reference Property

Write-only property for setting the AD5593R DAC reference mode.

Namespace: [IO.Devices.AD5593R](#)

Assembly: libremoteio (in libremoteio.dll) Version: 2.2020.136.3

◀ Syntax

C# VB F#

[Copy](#)

```
public ReferenceMode DAC_Reference { set; }
```

Property Value

Type: [ReferenceMode](#)

◀ See Also

Reference

[Device Class](#)

[IO.Devices.AD5593R Namespace](#)

DeviceGPIO_Inputs Property

GPIO input register state. Any I/O pin that is not configured as a GPIO input will read as zero.

Namespace: [IO.Devices.AD5593R](#)

Assembly: libremoteio (in libremoteio.dll) Version: 2.2020.136.3

◀ Syntax

C# VB F#

[Copy](#)

```
public byte GPIO_Inputs { get; }
```

Property Value

Type: [Byte](#)

◀ See Also

Reference

[Device Class](#)

[IO.Devices.AD5593R Namespace](#)

DeviceGPIO_Outputs Property

GPIO output register state. Any I/O pin that is not configured as a GPIO output will be written as zero.

Namespace: [IO.Devices.AD5593R](#)

Assembly: libremoteio (in libremoteio.dll) Version: 2.2020.136.3

◀ Syntax

C# VB F#

[Copy](#)

```
public byte GPIO_Outputs { get; set; }
```

Property Value

Type: [Byte](#)

◀ See Also

Reference

[Device Class](#)

[IO.Devices.AD5593R Namespace](#)

Device Methods

The [Device](#) type exposes the following members.

▪ Methods

	Name	Description
≡	ADC_Create	Create an AD5593R ADC input object.
≡	ConfigureChannel	Configure a single AD5593R I/O pin.
≡	DAC_Create	Create an AD5593R DAC output object.
≡	GPIO_Create	Create an AD5593R GPIO pin object.
≡	Read_ADC	Read from an ADC channel.
≡	Write_DAC	Write to a DAC channel.

[Top](#)

▪ See Also

Reference

[Device Class](#)

[IO.Devices.AD5593R Namespace](#)

DeviceADC_Create Method

Create an AD5593R ADC input object.

Namespace: [IO.Devices.AD5593R](#)

Assembly: libremoteio (in libremoteio.dll) Version: 2.2020.136.3

↳ Syntax

C# VB F#

[Copy](#)

```
public Sample ADC_Create(  
    int channel  
)
```

Parameters

channel

Type: [SystemInt32](#)

AD5593R ADC channel number (0 to 7).

Return Value

Type: [Sample](#)

ADC input object.

↳ See Also

[Reference](#)

[Device Class](#)

[IO.Devices.AD5593R Namespace](#)

DeviceConfigureChannel Method

Configure a single ADC5593R I/O pin.

Namespace: [IO.Devices.AD5593R](#)

Assembly: libremoteio (in libremoteio.dll) Version: 2.2020.136.3

« Syntax

C# VB F#

[Copy](#)

```
public void ConfigureChannel(  
    int channel,  
    PinMode mode  
)
```

Parameters

channel

Type: [SystemInt32](#)

ADC5593R I/O channel number (0 to 7).

mode

Type: [IO.Devices.AD5593RPinMode](#)

ADC5593R I/O pin mode.

« See Also

[Reference](#)

[Device Class](#)

[IO.Devices.AD5593R Namespace](#)

DeviceDAC_Create Method

Create an AD5593R DAC output object.

Namespace: [IO.Devices.AD5593R](#)

Assembly: libremoteio (in libremoteio.dll) Version: 2.2020.136.3

↳ Syntax

C# VB F#

[Copy](#)

```
public Sample DAC_Create(  
    int channel,  
    int sample = 0  
)
```

Parameters

channel

Type: [SystemInt32](#)

AD5593R DAC channel number (0 to 7).

sample (Optional)

Type: [SystemInt32](#)

Initial DAC output sample.

Return Value

Type: [Sample](#)

DAC output object.

↳ See Also

Reference

[Device Class](#)

IO.Devices.AD5593R Namespace

DeviceGPIO_Create Method

Create an AD5593R GPIO pin object.

Namespace: [IO.Devices.AD5593R](#)

Assembly: libremoteio (in libremoteio.dll) Version: 2.2020.136.3

◀ Syntax

C# VB F#

[Copy](#)

```
public Pin GPIO_Create(  
    int channel,  
    Direction dir,  
    bool state = false  
)
```

Parameters

channel

Type: [SystemInt32](#)

AD5593R GPIO channel number (0 to 7).

dir

Type: [IO.Interfaces.GPIODirection](#)

GPIO pin data direction.

state (Optional)

Type: [SystemBoolean](#)

Initial GPIO output state.

Return Value

Type: [Pin](#)

GPIO pin object.

See Also

[Reference](#)

[Device Class](#)

[IO.Devices.AD5593R Namespace](#)

DeviceRead_ADC Method

Read from an ADC channel.

Namespace: [IO.Devices.AD5593R](#)

Assembly: libremoteio (in libremoteio.dll) Version: 2.2020.136.3

▪ Syntax

C# VB F#

[Copy](#)

```
public int Read_ADC(  
    int channel  
)
```

Parameters

channel

Type: [SystemInt32](#)

ADC channel number (0 to 7).

Return Value

Type: [Int32](#)

ADC input sample data (0 to 4095).

▪ See Also

[Reference](#)

[Device Class](#)

[IO.Devices.AD5593R Namespace](#)

DeviceWrite_DAC Method

Write to a DAC channel.

Namespace: [IO.Devices.AD5593R](#)

Assembly: libremoteio (in libremoteio.dll) Version: 2.2020.136.3

« Syntax

C# VB F#

[Copy](#)

```
public void Write_DAC(
    int channel,
    int data
)
```

Parameters

channel

Type: [SystemInt32](#)

DAC channel number (0 to 7).

data

Type: [SystemInt32](#)

DAC output sample data (0 to 4095).

« See Also

[Reference](#)

[Device Class](#)

[IO.Devices.AD5593R Namespace](#)

Device Fields

The [Device](#) type exposes the following members.

Fields

	Name	Description
• 	ADC_Resolution	ADC resolution in bits.
• 	DAC_Resolution	DAC resolution in bits.
• 	MaxChannel	Maximum I/O channel number.
• 	MinChannel	Minimum I/O channel number.

[Top](#)

See Also

Reference

[Device Class](#)

[IO.Devices.AD5593R Namespace](#)

DeviceADC_Resolution Field

ADC resolution in bits.

Namespace: [IO.Devices.AD5593R](#)

Assembly: libremoteio (in libremoteio.dll) Version: 2.2020.136.3

◀ Syntax

C# VB F#

[Copy](#)

```
public const int ADC_Resolution = 12
```

Field Value

Type: [Int32](#)

◀ See Also

Reference

[Device Class](#)

[IO.Devices.AD5593R Namespace](#)

DeviceDAC_Resolution Field

DAC resolution in bits.

Namespace: [IO.Devices.AD5593R](#)

Assembly: libremoteio (in libremoteio.dll) Version: 2.2020.136.3

◀ Syntax

C# VB F#

[Copy](#)

```
public const int DAC_Resolution = 12
```

Field Value

Type: [Int32](#)

◀ See Also

Reference

[Device Class](#)

[IO.Devices.AD5593R Namespace](#)

DeviceMaxChannel Field

Maximum I/O channel number.

Namespace: [IO.Devices.AD5593R](#)

Assembly: libremoteio (in libremoteio.dll) Version: 2.2020.136.3

◀ Syntax

C# VB F#

[Copy](#)

```
public const int MaxChannel = 7
```

Field Value

Type: [Int32](#)

◀ See Also

Reference

[Device Class](#)

[IO.Devices.AD5593R Namespace](#)

DeviceMinChannel Field

Minimum I/O channel number.

Namespace: [IO.Devices.AD5593R](#)

Assembly: libremoteio (in libremoteio.dll) Version: 2.2020.136.3

◀ Syntax

C# VB F#

[Copy](#)

```
public const int MinChannel = 0
```

Field Value

Type: [Int32](#)

◀ See Also

Reference

[Device Class](#)

[IO.Devices.AD5593R Namespace](#)

PinMode Enumeration

AD5593R I/O Pin Modes.

Namespace: [IO.Devices.AD5593R](#)

Assembly: libremoteio (in libremoteio.dll) Version: 2.2020.136.3

▪ Syntax

C# VB F#

[Copy](#)

```
public enum PinMode
```

▪ Members

Member name	Value	Description
ADC_Input	0	Analog input.
DAC_Output	1	Analog output.
GPIO_Input	2	GPIO input.
GPIO_Output	3	GPIO output.
GPIO_Output_OpenDrain	4	GPIO open drain output.

▪ See Also

[Reference](#)

IO.Devices.AD5593R Namespace

ReferenceMode Enumeration

ADC5593R ADC and DAC reference settings.

Namespace: [IO.Devices.AD5593R](#)

Assembly: libremoteio (in libremoteio.dll) Version: 2.2020.136.3

▪ Syntax

C# VB F#

[Copy](#)

```
public enum ReferenceMode
```

▪ Members

Member name	Value	Description
Internalx1	0	The reference voltage is 2.5V using the internal reference.
Internalx2	1	The reference voltage is 5.0V using the internal reference.
Externalx1	2	The reference voltage is 1.0*Vref, using an external reference.
Externalx2	3	The reference voltage is 2.0*Vref, using an external reference.

↳ See Also

Reference

[IO.Devices.AD5593R Namespace](#)

IO.Devices.AD5593R.ADC Namespace

AD5593 Analog/Digital I/O Device ADC Input Services.

◀ Classes

Class	Description
 Sample	Encapsulates AD5593R ADC inputs.

Sample Class

Encapsulates AD5593R ADC inputs.

▪ Inheritance Hierarchy

[SystemObject](#) [IO.Devices.AD5593R.ADCSample](#)

Namespace: [IO.Devices.AD5593R.ADC](#)

Assembly: libremoteio (in libremoteio.dll) Version: 2.2020.136.3

▪ Syntax

C# VB F#

[Copy](#)

```
public class Sample : Sample
```

The [Sample](#) type exposes the following members.

▪ Constructors

	Name	Description
	Sample	Create an AD5593R ADC input.

[Top](#)

▪ Properties

	Name	Description
	resolution	Read-only property returning the number of bits of resolution.



[sample](#)

Read-only property returning an integer analog sample value.

[Top](#)

◀ See Also

Reference

[IO.Devices.AD5593R.ADC Namespace](#)

Sample Constructor

Create an AD5593R ADC input.

Namespace: [IO.Devices.AD5593R.ADC](#)

Assembly: libremoteio (in libremoteio.dll) Version: 2.2020.136.3

« Syntax

C# VB F#

[Copy](#)

```
public Sample(  
    Device dev,  
    int channel  
)
```

Parameters

dev

Type: [IO.Devices.AD5593RDevice](#)

AD5593R device object.

channel

Type: [SystemInt32](#)

AD5593R I/O channel number (0 to 7).

« See Also

[Reference](#)

[Sample Class](#)

[IO.Devices.AD5593R.ADC Namespace](#)

Sample Properties

The [Sample](#) type exposes the following members.

Properties

	Name	Description
	resolution	Read-only property returning the number of bits of resolution.
	sample	Read-only property returning an integer analog sample value.

[Top](#)

See Also

Reference

[Sample Class](#)

[IO.Devices.AD5593R.ADC Namespace](#)

Sampleresolution Property

Read-only property returning the number of bits of resolution.

Namespace: [IO.Devices.AD5593R.ADC](#)

Assembly: libremoteio (in libremoteio.dll) Version: 2.2020.136.3

▪ Syntax

C# VB F#

[Copy](#)

```
public int resolution { get; }
```

Property Value

Type: [Int32](#)

Implements

[Sampleresolution](#)

▪ See Also

[Reference](#)

[Sample Class](#)

[IO.Devices.AD5593R.ADC Namespace](#)

Samplesample Property

Read-only property returning an integer analog sample value.

Namespace: [IO.Devices.AD5593R.ADC](#)

Assembly: libremoteio (in libremoteio.dll) Version: 2.2020.136.3

▪ Syntax

C# VB F#

[Copy](#)

```
public int sample { get; }
```

Property Value

Type: [Int32](#)

Implements

[Samplesample](#)

▪ See Also

[Reference](#)

[Sample Class](#)

[IO.Devices.AD5593R.ADC Namespace](#)

IO.Devices.AD5593R.DAC Namespace

AD5593 Analog/Digital I/O Device DAC Output Services.

◀ Classes

Class	Description
 Sample	Encapsulates AD5593R DAC outputs.

Sample Class

Encapsulates AD5593R DAC outputs.

▪ Inheritance Hierarchy

[SystemObject](#) [IO.Devices.AD5593R.DACSample](#)

Namespace: [IO.Devices.AD5593R.DAC](#)

Assembly: libremoteio (in libremoteio.dll) Version: 2.2020.136.3

▪ Syntax

C# VB F#

[Copy](#)

```
public class Sample : Sample
```

The [Sample](#) type exposes the following members.

▪ Constructors

	Name	Description
	Sample	Create an AD5593R DAC output.

[Top](#)

▪ Properties

	Name	Description
	resolution	Read-only property returning the number of bits of resolution.



sample

Write-only property for writing an integer analog sample to a DAC output.

[Top](#)

◀ See Also

Reference

[IO.Devices.AD5593R.DAC Namespace](#)

Sample Constructor

Create an AD5593R DAC output.

Namespace: [IO.Devices.AD5593R.DAC](#)

Assembly: libremoteio (in libremoteio.dll) Version: 2.2020.136.3

↳ Syntax

C# VB F#

[Copy](#)

```
public Sample(  
    Device dev,  
    int channel,  
    int sample = 0  
)
```

Parameters

dev

Type: [IO.Devices.AD5593RDevice](#)
AD5593R device object.

channel

Type: [SystemInt32](#)
AD5593R I/O channel number (0 to 7).

sample (Optional)

Type: [SystemInt32](#)
Initial DAC output sample.

↳ See Also

[Reference](#)

[Sample Class](#)

IO.Devices.AD5593R.DAC Namespace

Sample Properties

The [Sample](#) type exposes the following members.

Properties

	Name	Description
	resolution	Read-only property returning the number of bits of resolution.
	sample	Write-only property for writing an integer analog sample to a DAC output.

[Top](#)

See Also

Reference

[Sample Class](#)

[IO.Devices.AD5593R.DAC Namespace](#)

Sampleresolution Property

Read-only property returning the number of bits of resolution.

Namespace: [IO.Devices.AD5593R.DAC](#)

Assembly: libremoteio (in libremoteio.dll) Version: 2.2020.136.3

▪ Syntax

C# VB F#

[Copy](#)

```
public int resolution { get; }
```

Property Value

Type: [Int32](#)

Implements

[Sampleresolution](#)

▪ See Also

[Reference](#)

[Sample Class](#)

[IO.Devices.AD5593R.DAC Namespace](#)

Samplesample Property

Write-only property for writing an integer analog sample to a DAC output.

Namespace: [IO.Devices.AD5593R.DAC](#)

Assembly: libremoteio (in libremoteio.dll) Version: 2.2020.136.3

↳ Syntax

[C#](#) [VB](#) [F#](#)

[Copy](#)

```
public int sample { set; }
```

Property Value

Type: [Int32](#)

Implements

[Samplesample](#)

↳ See Also

Reference

[Sample Class](#)

[IO.Devices.AD5593R.DAC Namespace](#)

IO.Devices.AD5593R.GPIO Namespace

AD5593 Analog/Digital I/O Device GPIO Pin Services.

◀ Classes

Class	Description
 Pin	Encapsulates AD5593R GPIO pins.

Pin Class

Encapsulates AD5593R GPIO pins.

▪ Inheritance Hierarchy

[SystemObject](#) [IO.Devices.AD5593R.GPIOPin](#)

Namespace: [IO.Devices.AD5593R.GPIO](#)

Assembly: libremoteio (in libremoteio.dll) Version: 2.2020.136.3

▪ Syntax

C# VB F#

[Copy](#)

```
public class Pin : Pin
```

The [Pin](#) type exposes the following members.

▪ Constructors

	Name	Description
	Pin	Create an AD5593R GPIO pin.

[Top](#)

▪ Properties

	Name	Description
	state	Read/Write GPIO state property.

[Top](#)

↳ See Also

Reference

[IO.Devices.AD5593R.GPIO Namespace](#)

Pin Constructor

Create an AD5593R GPIO pin.

Namespace: [IO.Devices.AD5593R.GPIO](#)

Assembly: libremoteio (in libremoteio.dll) Version: 2.2020.136.3

◀ Syntax

C# VB F#

[Copy](#)

```
public Pin(  
    Device dev,  
    int channel,  
    Direction dir,  
    bool state = false  
)
```

Parameters

dev

Type: [IO.Devices.AD5593RDevice](#)

AD5593R device object.

channel

Type: [SystemInt32](#)

AD5593R I/O channel number (0 to 7).

dir

Type: [IO.Interfaces.GPIODirection](#)

GPIO pin data direction.

state (Optional)

Type: [SystemBoolean](#)

Initial GPIO output state.

↳ See Also

Reference

[Pin Class](#)

[IO.Devices.AD5593R.GPIO Namespace](#)

Pin Properties

The [Pin](#) type exposes the following members.

Properties

	Name	Description
	state	Read/Write GPIO state property.

[Top](#)

See Also

Reference

[Pin Class](#)

[IO.Devices.AD5593R.GPIO Namespace](#)

Pinstate Property

Read/Write GPIO state property.

Namespace: [IO.Devices.AD5593R.GPIO](#)

Assembly: libremoteio (in libremoteio.dll) Version: 2.2020.136.3

▪ Syntax

C# VB F#

[Copy](#)

```
public bool state { get; set; }
```

Property Value

Type: [Boolean](#)

Implements

[Pinstate](#)

▪ See Also

[Reference](#)

[Pin Class](#)

[IO.Devices.AD5593R.GPIO Namespace](#)

IO.Devices.ADC121C021 Namespace

ADC121C021 I²C A/D Converter Services

↳ Classes

Class	Description
 Sample	Encapsulates the ADC121C021 I ² C A/D converter.

Sample Class

Encapsulates the ADC121C021 I²C A/D converter.

▪ Inheritance Hierarchy

[SystemObject](#) [IO.Devices.ADC121C021Sample](#)

Namespace: [IO.Devices.ADC121C021](#)

Assembly: libremoteio (in libremoteio.dll) Version: 2.2020.136.3

▪ Syntax

[C#](#) [VB](#) [F#](#)

[Copy](#)

```
public class Sample : Sample
```

The [Sample](#) type exposes the following members.

▪ Constructors

	Name	Description
	Sample	Constructor for an ADC121C021 analog input.

[Top](#)

▪ Properties

	Name	Description
	resolution	Return the number of bits of A/D

resolution.



[sample](#) Returns a single 12-bit analog sample.

[Top](#)

See Also

Reference

[IO.Devices.ADC121C021 Namespace](#)

Sample Constructor

Constructor for an ADC121C021 analog input.

Namespace: [IO.Devices.ADC121C021](#)

Assembly: libremoteio (in libremoteio.dll) Version: 2.2020.136.3

▪ Syntax

C# VB F#

[Copy](#)

```
public Sample(  
    Bus bus,  
    byte addr  
)
```

Parameters

bus

Type: [IO.Interfaces.I2CBus](#)

I²C bus controller.

addr

Type: [SystemByte](#)

I²C slave address.

▪ See Also

[Reference](#)

[Sample Class](#)

[IO.Devices.ADC121C021 Namespace](#)

Sample Properties

The [Sample](#) type exposes the following members.

Properties

	Name	Description
	resolution	Return the number of bits of A/D resolution.
	sample	Returns a single 12-bit analog sample.

[Top](#)

See Also

[Reference](#)

[Sample Class](#)

[IO.Devices.ADC121C021 Namespace](#)

Sampleresolution Property

Return the number of bits of A/D resolution.

Namespace: [IO.Devices.ADC121C021](#)

Assembly: libremoteio (in libremoteio.dll) Version: 2.2020.136.3

▪ Syntax

C# VB F#

[Copy](#)

```
public int resolution { get; }
```

Property Value

Type: [Int32](#)

Implements

[Sampleresolution](#)

▪ See Also

[Reference](#)

[Sample Class](#)

[IO.Devices.ADC121C021 Namespace](#)

Samplesample Property

Returns a single 12-bit analog sample.

Namespace: [IO.Devices.ADC121C021](#)

Assembly: libremoteio (in libremoteio.dll) Version: 2.2020.136.3

▪ Syntax

C# VB F#

[Copy](#)

```
public int sample { get; }
```

Property Value

Type: [Int32](#)

Implements

[Samplesample](#)

▪ See Also

[Reference](#)

[Sample Class](#)

[IO.Devices.ADC121C021 Namespace](#)

IO.Devices.ClickBoards.RemoteIO.ADAC Namespace

Mikroelektronika ADAC Click MIKROE-2690 Services

▪ Classes

	Class	Description
	Board	Encapsulates the Mikroelektronika ADAC Click Board. MIKROE-2690 .

Board Class

Encapsulates the Mikroelektronika ADAC Click Board. [MIKROE-2690](#).

▪ Inheritance Hierarchy

[SystemObject](#) [IO.Devices.ClickBoards.RemoteIO.ADACBoard](#)

Namespace: [IO.Devices.ClickBoards.RemoteIO.ADAC](#)

Assembly: libremoteio (in libremoteio.dll) Version: 2.2020.136.3

▪ Syntax

C# VB F#

[Copy](#)

```
public class Board
```

The [Board](#) type exposes the following members.

▪ Constructors

	Name	Description
	Board	Constructor for a single ADAC click.

[Top](#)

▪ Properties

	Name	Description
	device	Returns the underlying AD5593R device object.

[Top](#)

◀ Methods

	Name	Description
≡	ADC	Factory function for creating ADC inputs.
≡	DAC	Factory function for creating DAC outputs.
≡	GPIO	Factory function for creating GPIO pins.
≡	Reset	Issue hardware reset to the AD5593R.

[Top](#)

◀ Fields

	Name	Description
◆ S	DefaultAddress	Default I ² C slave address.

[Top](#)

◀ See Also

Reference

[IO.Devices.ClickBoards.RemoteIO.ADAC Namespace](#)

Board Constructor

Constructor for a single ADAC click.

Namespace: [IO.Devices.ClickBoards.RemoteIO.ADAC](#)

Assembly: libremoteio (in libremoteio.dll) Version: 2.2020.136.3

↳ Syntax

C# VB F#

[Copy](#)

```
public Board(  
    int socknum,  
    int addr = 16,  
    Device remdev = null  
)
```

Parameters

socknum

Type: [SystemInt32](#)

mikroBUS socket number.

addr (Optional)

Type: [SystemInt32](#)

I²C slave address.

remdev (Optional)

Type: [IO.RemoteDevice](#)

Remote I/O server device object.

↳ See Also

[Reference](#)

[Board Class](#)

IO.Devices.ClickBoards.RemoteIO.ADAC Namespace

Board Properties

The [Board](#) type exposes the following members.

▪ Properties

	Name	Description
	device	Returns the underlying AD5593R device object.

[Top](#)

▪ See Also

[Reference](#)

[Board Class](#)

[IO.Devices.ClickBoards.RemoteIO.ADAC Namespace](#)

Boarddevice Property

Returns the underlying AD5593R device object.

Namespace: [IO.Devices.ClickBoards.RemoteIO.ADAC](#)

Assembly: libremoteio (in libremoteio.dll) Version: 2.2020.136.3

◀ Syntax

C# VB F#

[Copy](#)

```
public Device device { get; }
```

Property Value

Type: [Device](#)

◀ See Also

Reference

[Board Class](#)

[IO.Devices.ClickBoards.RemoteIO.ADAC Namespace](#)

Board Methods

The [Board](#) type exposes the following members.

↳ Methods

	Name	Description
≡	ADC	Factory function for creating ADC inputs.
≡	DAC	Factory function for creating DAC outputs.
≡	GPIO	Factory function for creating GPIO pins.
≡	Reset	Issue hardware reset to the AD5593R.

[Top](#)

↳ See Also

Reference

[Board Class](#)

[IO.Devices.ClickBoards.RemoteIO.ADAC Namespace](#)

BoardADC Method

Factory function for creating ADC inputs.

Namespace: [IO.Devices.ClickBoards.RemotelIO.ADAC](#)

Assembly: libremoteio (in libremoteio.dll) Version: 2.2020.136.3

◀ Syntax

C# VB F#

[Copy](#)

```
public Sample ADC(  
    int channel  
)
```

Parameters

channel

Type: [SystemInt32](#)

AD5593R I/O channel number (0 to 7).

Return Value

Type: [Sample](#)

ADC input object.

◀ See Also

[Reference](#)

[Board Class](#)

[IO.Devices.ClickBoards.RemotelIO.ADAC Namespace](#)

BoardDAC Method

Factory function for creating DAC outputs.

Namespace: [IO.Devices.ClickBoards.RemoteIO.ADAC](#)

Assembly: libremoteio (in libremoteio.dll) Version: 2.2020.136.3

↳ Syntax

C# VB F#

[Copy](#)

```
public Sample DAC(  
    int channel,  
    int sample = 0  
)
```

Parameters

channel

Type: [SystemInt32](#)

AD5593R I/O channel number (0 to 7).

sample (Optional)

Type: [SystemInt32](#)

Initial DAC output sample.

Return Value

Type: [Sample](#)

DAC output object.

↳ See Also

Reference

[Board Class](#)

IO.Devices.ClickBoards.RemoteIO.ADAC Namespace

BoardGPIO Method

Factory function for creating GPIO pins.

Namespace: [IO.Devices.ClickBoards.RemoteIO.ADAC](#)

Assembly: libremoteio (in libremoteio.dll) Version: 2.2020.136.3

◀ Syntax

C# VB F#

[Copy](#)

```
public Pin GPIO(  
    int channel,  
    Direction dir,  
    bool state = false  
)
```

Parameters

channel

Type: [SystemInt32](#)

AD5593R I/O channel number (0 to 7).

dir

Type: [IO.Interfaces.GPIODirection](#)

GPIO pin direction.

state (Optional)

Type: [SystemBoolean](#)

Initial GPIO output state.

Return Value

Type: [Pin](#)

GPIO pin object.

↳ See Also

Reference

[Board Class](#)

[IO.Devices.ClickBoards.RemoteIO.ADAC Namespace](#)

BoardReset Method

Issue hardware reset to the AD5593R.

Namespace: [IO.Devices.ClickBoards.RemoteIO.ADAC](#)

Assembly: libremoteio (in libremoteio.dll) Version: 2.2020.136.3

▪ Syntax

C# VB F#

[Copy](#)

```
public void Reset()
```

▪ See Also

Reference

[Board Class](#)

[IO.Devices.ClickBoards.RemoteIO.ADAC Namespace](#)

Board Fields

The [Board](#) type exposes the following members.

↳ Fields

	Name	Description
	DefaultAddress	Default I ² C slave address.

[Top](#)

↳ See Also

Reference

[Board Class](#)

[IO.Devices.ClickBoards.RemoteIO.ADAC Namespace](#)

BoardDefaultAddress Field

Default I²C slave address.

Namespace: [IO.Devices.ClickBoards.RemoteIO.ADAC](#)

Assembly: libremoteio (in libremoteio.dll) Version: 2.2020.136.3

↳ Syntax

[C#](#) [VB](#) [F#](#)

[Copy](#)

```
public const byte DefaultAddress = 16
```

Field Value

Type: [Byte](#)

↳ See Also

Reference

[Board Class](#)

[IO.Devices.ClickBoards.RemoteIO.ADAC Namespace](#)

IO.Devices.ClickBoards.RemoteIO.Expand Namespace

Mikroelektronika Expand Click MIKROE-951 Services

▪ Classes

Class	Description
 Board	Encapsulates the Mikroelektronika Expand Click Board.

Board Class

Encapsulates the Mikroelektronika Expand Click Board.

▪ Inheritance Hierarchy

[SystemObject](#) [IO.Devices.ClickBoards.RemoteIO.ExpandBoard](#)

Namespace: [IO.Devices.ClickBoards.RemoteIO.Expand](#)

Assembly: libremoteio (in libremoteio.dll) Version: 2.2020.136.3

▪ Syntax

C# VB F#

[Copy](#)

```
public class Board
```

The [Board](#) type exposes the following members.

▪ Constructors

	Name	Description
	Board	Constructor for a single Expand click.

[Top](#)

▪ Properties

	Name	Description
	device	Returns the underlying MCP23S17 device object.

[Top](#)

◀ Methods

	Name	Description
	GPIO	Factory function for creating GPIO pins.
	Reset	Issue hardware reset to the MCP23S17.

[Top](#)

◀ See Also

Reference

[IO.Devices.ClickBoards.RemoteIO.Expand Namespace](#)

Board Constructor

Constructor for a single Expand click.

Namespace: [IO.Devices.ClickBoards.RemotelO.Expand](#)

Assembly: libremoteio (in libremoteio.dll) Version: 2.2020.136.3

◀ Syntax

C# VB F#

[Copy](#)

```
public Board(  
    int socknum,  
    Device remdev = null  
)
```

Parameters

socknum

Type: [SystemInt32](#)

mikroBUS socket number.

remdev (Optional)

Type: [IO.RemoteDevice](#)

Remote I/O server device object.

◀ See Also

[Reference](#)

[Board Class](#)

[IO.Devices.ClickBoards.RemotelO.Expand Namespace](#)

Board Properties

The [Board](#) type exposes the following members.

▪ Properties

	Name	Description
	device	Returns the underlying MCP23S17 device object.

[Top](#)

▪ See Also

[Reference](#)

[Board Class](#)

[IO.Devices.ClickBoards.RemoteIO.Expand Namespace](#)

Boarddevice Property

Returns the underlying MCP23S17 device object.

Namespace: [IO.Devices.ClickBoards.RemoteIO.Expand](#)

Assembly: libremoteio (in libremoteio.dll) Version: 2.2020.136.3

« Syntax

C# VB F#

[Copy](#)

```
public Device device { get; }
```

Property Value

Type: [Device](#)

« See Also

Reference

[Board Class](#)

[IO.Devices.ClickBoards.RemoteIO.Expand Namespace](#)

Board Methods

The [Board](#) type exposes the following members.

▪ Methods

	Name	Description
	GPIO	Factory function for creating GPIO pins.
	Reset	Issue hardware reset to the MCP23S17.

[Top](#)

▪ See Also

[Reference](#)

[Board Class](#)

[IO.Devices.ClickBoards.RemoteIO.Expand Namespace](#)

BoardGPIO Method

Factory function for creating GPIO pins.

Namespace: [IO.Devices.ClickBoards.RemoteIO.Expand](#)

Assembly: libremoteio (in libremoteio.dll) Version: 2.2020.136.3

◀ Syntax

C# VB F#

[Copy](#)

```
public Pin GPIO(
    int channel,
    Direction dir,
    bool state = false
)
```

Parameters

channel

Type: [SystemInt32](#)

MCP23S17 channel number (0 to 15).

dir

Type: [IO.Interfaces.GPIODirection](#)

GPIO pin direction.

state (Optional)

Type: [SystemBoolean](#)

Initial GPIO output state.

Return Value

Type: [Pin](#)

GPIO pin object.

↳ See Also

Reference

[Board Class](#)

[IO.Devices.ClickBoards.RemoteIO.Expand Namespace](#)

BoardReset Method

Issue hardware reset to the MCP23S17.

Namespace: [IO.Devices.ClickBoards.RemoteIO.Expand](#)

Assembly: libremoteio (in libremoteio.dll) Version: 2.2020.136.3

▪ Syntax

C# VB F#

[Copy](#)

```
public void Reset()
```

▪ See Also

Reference

[Board Class](#)

[IO.Devices.ClickBoards.RemoteIO.Expand Namespace](#)

IO.Devices.ClickBoards.RemotelIO.Expand2 Namespace

Mikroelektronika Expand 2 Click MIKROE-1838 Services

▪ Classes

	Class	Description
	Board	Encapsulates the Mikroelektronika Expand 2 Click Board.

Board Class

Encapsulates the Mikroelektronika Expand 2 Click Board.

▪ Inheritance Hierarchy

[SystemObject](#) [IO.Devices.ClickBoards.RemoteIO.Expand2Board](#)

Namespace: [IO.Devices.ClickBoards.RemoteIO.Expand2](#)

Assembly: libremoteio (in libremoteio.dll) Version: 2.2020.136.3

▪ Syntax

C# VB F#

[Copy](#)

```
public class Board
```

The [Board](#) type exposes the following members.

▪ Constructors

	Name	Description
	Board	Constructor for a single Expand 2 click.

[Top](#)

▪ Properties

	Name	Description
	device	Returns the underlying MCP23017 device object.

[Top](#)

◀ Methods

	Name	Description
≡	GPIO	Factory function for creating GPIO pins.
≡	Reset	Issue hardware reset to the MCP23017.

[Top](#)

◀ Fields

	Name	Description
◆ S	DefaultAddress	Default I ² C slave address.

[Top](#)

◀ See Also

Reference

[IO.Devices.ClickBoards.RemoteIO.Expand2 Namespace](#)

Board Constructor

Constructor for a single Expand 2 click.

Namespace: [IO.Devices.ClickBoards.RemoteIO.Expand2](#)

Assembly: libremoteio (in libremoteio.dll) Version: 2.2020.136.3

↳ Syntax

C# VB F#

[Copy](#)

```
public Board(  
    int socknum,  
    int addr = 32,  
    Device remdev = null  
)
```

Parameters

socknum

Type: [SystemInt32](#)

mikroBUS socket number.

addr (Optional)

Type: [SystemInt32](#)

I²C slave address.

remdev (Optional)

Type: [IO.RemoteDevice](#)

Remote I/O server device object.

↳ See Also

[Reference](#)

[Board Class](#)

IO.Devices.ClickBoards.RemoteIO.Expand2 Namespace

Board Properties

The [Board](#) type exposes the following members.

▪ Properties

	Name	Description
	device	Returns the underlying MCP23017 device object.

[Top](#)

▪ See Also

[Reference](#)

[Board Class](#)

[IO.Devices.ClickBoards.RemoteIO.Expand2 Namespace](#)

Boarddevice Property

Returns the underlying MCP23017 device object.

Namespace: [IO.Devices.ClickBoards.RemoteIO.Expand2](#)

Assembly: libremoteio (in libremoteio.dll) Version: 2.2020.136.3

◀ Syntax

C# VB F#

[Copy](#)

```
public Device device { get; }
```

Property Value

Type: [Device](#)

◀ See Also

Reference

[Board Class](#)

[IO.Devices.ClickBoards.RemoteIO.Expand2 Namespace](#)

Board Methods

The [Board](#) type exposes the following members.

▪ Methods

	Name	Description
	GPIO	Factory function for creating GPIO pins.
	Reset	Issue hardware reset to the MCP23017.

[Top](#)

▪ See Also

[Reference](#)

[Board Class](#)

[IO.Devices.ClickBoards.RemoteIO.Expand2 Namespace](#)

BoardGPIO Method

Factory function for creating GPIO pins.

Namespace: [IO.Devices.ClickBoards.RemoteIO.Expand2](#)

Assembly: libremoteio (in libremoteio.dll) Version: 2.2020.136.3

◀ Syntax

C# VB F#

[Copy](#)

```
public Pin GPIO(
    int channel,
    Direction dir,
    bool state = false
)
```

Parameters

channel

Type: [SystemInt32](#)

MCP23017 channel number (0 to 15).

dir

Type: [IO.Interfaces.GPIODirection](#)

GPIO pin direction.

state (Optional)

Type: [SystemBoolean](#)

Initial GPIO output state.

Return Value

Type: [Pin](#)

GPIO pin object.

↳ See Also

Reference

[Board Class](#)

[IO.Devices.ClickBoards.RemoTeIO.Expand2 Namespace](#)

BoardReset Method

Issue hardware reset to the MCP23017.

Namespace: [IO.Devices.ClickBoards.RemoteIO.Expand2](#)

Assembly: libremoteio (in libremoteio.dll) Version: 2.2020.136.3

▪ Syntax

C# VB F#

[Copy](#)

```
public void Reset()
```

▪ See Also

Reference

[Board Class](#)

[IO.Devices.ClickBoards.RemoteIO.Expand2 Namespace](#)

Board Fields

The [Board](#) type exposes the following members.

↳ Fields

	Name	Description
	DefaultAddress	Default I ² C slave address.

[Top](#)

↳ See Also

Reference

[Board Class](#)

[IO.Devices.ClickBoards.RemoteIO.Expand2 Namespace](#)

BoardDefaultAddress Field

Default I²C slave address.

Namespace: [IO.Devices.ClickBoards.RemoteIO.Expand2](#)

Assembly: libremoteio (in libremoteio.dll) Version: 2.2020.136.3

↳ Syntax

[C#](#) [VB](#) [F#](#)

[Copy](#)

```
public const int DefaultAddress = 32
```

Field Value

Type: [Int32](#)

↳ See Also

Reference

[Board Class](#)

[IO.Devices.ClickBoards.RemoteIO.Expand2 Namespace](#)

IO.Devices.ClickBoards.RemoteIO.PWM Namespace

Mikroelektronika PWM Click MIKROE-1898 Services

▪ Classes

	Class	Description
	Board	Encapsulates the Mikroelektronika PWM Click Board. MIKROE-1898 .

Board Class

Encapsulates the Mikroelektronika PWM Click Board. [MIKROE-1898](#).

▪ Inheritance Hierarchy

[SystemObject](#) [IO.Devices.ClickBoards.RemoteIO.PWMBoard](#)

Namespace: [IO.Devices.ClickBoards.RemoteIO.PWM](#)

Assembly: libremoteio (in libremoteio.dll) Version: 2.2020.136.3

▪ Syntax

C# VB F#

[Copy](#)

```
public class Board
```

The [Board](#) type exposes the following members.

▪ Constructors

	Name	Description
	Board	Constructor for a single PWM click.

[Top](#)

▪ Properties

	Name	Description
	dev	Returns the underlying PCA9685 device object.

[Top](#)

◀ Methods

	Name	Description
≡	GPIO	Factory function for creating GPIO output pins.
≡	PWM	Factory function for creating PWM outputs.
≡	Servo	Factory function for creating servo outputs.

[Top](#)

◀ Fields

	Name	Description
• s	DefaultAddress	Default I ² C slave address.

[Top](#)

◀ See Also

Reference

[IO.Devices.ClickBoards.RemoteIO.PWM Namespace](#)

Board Constructor

Constructor for a single PWM click.

Namespace: [IO.Devices.ClickBoards.RemoteIO.PWM](#)

Assembly: libremoteio (in libremoteio.dll) Version: 2.2020.136.3

↳ Syntax

C# VB F#

[Copy](#)

```
public Board(
    int socknum,
    int freq,
    int addr = 64,
    Device remdev = null
)
```

Parameters

socknum

Type: [SystemInt32](#)

mikroBUS socket number.

freq

Type: [SystemInt32](#)

PWM pulse frequency in Hz.

addr (Optional)

Type: [SystemInt32](#)

I²C slave address.

remdev (Optional)

Type: [IO.RemoteDevice](#)

Remote I/O device object.

↳ See Also

Reference

[Board Class](#)

[IO.Devices.ClickBoards.RemoteIO.PWM Namespace](#)

Board Properties

The [Board](#) type exposes the following members.

▪ Properties

	Name	Description
	dev	Returns the underlying PCA9685 device object.

[Top](#)

▪ See Also

[Reference](#)

[Board Class](#)

[IO.Devices.ClickBoards.RemoteIO.PWM Namespace](#)

Boarddev Property

Returns the underlying PCA9685 device object.

Namespace: [IO.Devices.ClickBoards.RemoteIO.PWM](#)

Assembly: libremoteio (in libremoteio.dll) Version: 2.2020.136.3

↳ Syntax

C# VB F#

[Copy](#)

```
public Device dev { get; }
```

Property Value

Type: [Device](#)

↳ See Also

Reference

[Board Class](#)

[IO.Devices.ClickBoards.RemoteIO.PWM Namespace](#)

Board Methods

The [Board](#) type exposes the following members.

↳ Methods

	Name	Description
≡	GPIO	Factory function for creating GPIO output pins.
≡	PWM	Factory function for creating PWM outputs.
≡	Servo	Factory function for creating servo outputs.

[Top](#)

↳ See Also

Reference

[Board Class](#)

[IO.Devices.ClickBoards.RemoteIO.PWM Namespace](#)

BoardGPIO Method

Factory function for creating GPIO output pins.

Namespace: [IO.Devices.ClickBoards.RemotelIO.PWM](#)

Assembly: libremoteio (in libremoteio.dll) Version: 2.2020.136.3

↳ Syntax

C# VB F#

[Copy](#)

```
public Pin GPIO(  
    int channel,  
    bool state = false  
)
```

Parameters

channel

Type: [SystemInt32](#)

PCA9685 output channel number.

state (Optional)

Type: [SystemBoolean](#)

Initial GPIO output state.

Return Value

Type: [Pin](#)

GPIO output pin object.

↳ See Also

Reference

[Board Class](#)

IO.Devices.ClickBoards.RemoteIO.PWM Namespace

BoardPWM Method

Factory function for creating PWM outputs.

Namespace: [IO.Devices.ClickBoards.RemoteIO.PWM](#)

Assembly: libremoteio (in libremoteio.dll) Version: 2.2020.136.3

↳ Syntax

C# VB F#

[Copy](#)

```
public Output PWM(  
    int channel,  
    double dutycycle = 0  
)
```

Parameters

channel

Type: [SystemInt32](#)

PCA9685 output channel number.

dutycycle (Optional)

Type: [SystemDouble](#)

Initial PWM output duty cycle.

Return Value

Type: [Output](#)

PWM output object.

↳ See Also

Reference

[Board Class](#)

IO.Devices.ClickBoards.RemoteIO.PWM Namespace

BoardServo Method

Factory function for creating servo outputs.

Namespace: [IO.Devices.ClickBoards.RemoteIO.PWM](#)

Assembly: libremoteio (in libremoteio.dll) Version: 2.2020.136.3

↳ Syntax

C# VB F#

[Copy](#)

```
public Output Servo(  
    int channel,  
    double position = 0  
)
```

Parameters

channel

Type: [SystemInt32](#)

PCA9685 output channel number.

position (Optional)

Type: [SystemDouble](#)

Initial servo position.>

Return Value

Type: [Output](#)

Servo output object.

↳ See Also

Reference

[Board Class](#)

IO.Devices.ClickBoards.RemoteIO.PWM Namespace

Board Fields

The [Board](#) type exposes the following members.

↳ Fields

	Name	Description
 	DefaultAddress	Default I ² C slave address.

[Top](#)

↳ See Also

Reference

[Board Class](#)

[IO.Devices.ClickBoards.RemoteIO.PWM Namespace](#)

BoardDefaultAddress Field

Default I²C slave address.

Namespace: [IO.Devices.ClickBoards.RemoteIO.PWM](#)

Assembly: libremoteio (in libremoteio.dll) Version: 2.2020.136.3

↳ Syntax

[C#](#) [VB](#) [F#](#)

[Copy](#)

```
public const byte DefaultAddress = 64
```

Field Value

Type: [Byte](#)

↳ See Also

Reference

[Board Class](#)

[IO.Devices.ClickBoards.RemoteIO.PWM Namespace](#)

IO.Devices.ClickBoards.RemoteIO.SevenSegment Namespace

Mikroelektronika 7Seg Click MIKROE-1201 Services

Classes

Class	Description
 Board	Encapsulates the Mikroelektronika 7Seg Click Board. MIKROE-1201 .

Enumerations

Enumeration	Description
 BoardBase	Numeral systems.
 BoardZeroBlanking	Zero blanking modes.

Board Class

Encapsulates the Mikroelektronika 7Seg Click Board. [MIKROE-1201](#).

▪ Inheritance Hierarchy

[SystemObject](#) [IO.Devices.ClickBoards.RemoteIO.SevenSegmentBoard](#)

Namespace: [IO.Devices.ClickBoards.RemoteIO.SevenSegment](#)

Assembly: libremoteio (in libremoteio.dll) Version: 2.2020.136.3

▪ Syntax

C# VB F#

[Copy](#)

```
public class Board
```

The [Board](#) type exposes the following members.

▪ Constructors

	Name	Description
	Board	Constructor for a single 7seg click.

[Top](#)

▪ Properties

	Name	Description
	blanking	Zero blanking mode. Allowed values are None , Leading , and Full .

	brightness	Write-only property for setting the brightness of the display. Allowed values are 0.0 to 100.0 percent.
	leftdp	Write-only property for setting the left digit decimal point.
	radix	Numerical base or radix. Allowed values are Decimal and Hexadecimal .
	rightdp	Write-only property for setting the right digit decimal point.
	state	Write-only property for setting the state of the display. Allowed values are 0 to 99 for decimal mode and 0 to 255 for hexadecimal mode.

[Top](#)

Methods

	Name	Description
	Clear	Clear the display.
	Reset	Issue hardware reset to the 74HC595 shift register chain.

[Top](#)

Remarks

The **MISOakaSDI** pin should be removed from the 7seg click, because it is not tri-state and will interfere with other devices on the same SPI

bus.

▲ See Also

Reference

[IO.Devices.ClickBoards.RemoteIO.SevenSegment Namespace](#)

Board Constructor

Constructor for a single 7seg click.

Namespace: [IO.Devices.ClickBoards.RemotelO.SevenSegment](#)

Assembly: libremoteio (in libremoteio.dll) Version: 2.2020.136.3

◀ Syntax

C# VB F#

Copy

```
public Board(
    int socket,
    BoardBase radix = BoardBase.Decimal,
    BoardZeroBlanking blanking = BoardZeroBlanking,
    int pwmfreq = 100,
    Device remdev = null
)
```

Parameters

socket

Type: [SystemInt32](#)

mikroBUS socket number.

radix (Optional)

Type: [IO.Devices.ClickBoards.RemotelO.SevenSegmentBoardBase](#)

Numerical base or radix. Allowed values are [Decimal](#) and [Hexadecimal](#).

blanking (Optional)

Type: [IO.Devices.ClickBoards.RemotelO.SevenSegmentBoardZeroBlanking](#)

Zero blanking. Allowed values are [None](#), [Leading](#), and [Full](#).

pwmfreq (Optional)

Type: [SystemInt32](#)

PWM frequency. Set to zero to use GPIO instead of PWM.

remdev (Optional)

Type: [IO.RemoteDevice](#)

Remote I/O server device object.

See Also

[Reference](#)

[Board Class](#)

[IO.Devices.ClickBoards.RemoteIO.SevenSegment Namespace](#)

Board Properties

The [Board](#) type exposes the following members.

Properties

	Name	Description
	blanking	Zero blanking mode. Allowed values are None , Leading , and Full .
	brightness	Write-only property for setting the brightness of the display. Allowed values are 0.0 to 100.0 percent.
	leftdp	Write-only property for setting the left digit decimal point.
	radix	Numerical base or radix. Allowed values are Decimal and Hexadecimal .
	rightdp	Write-only property for setting the right digit decimal point.
	state	Write-only property for setting the state of the display. Allowed values are 0 to 99 for decimal mode and 0 to 255 for hexadecimal mode.

[Top](#)

↳ See Also

Reference

[Board Class](#)

[IO.Devices.ClickBoards.RemoTeIO.SevenSegment Namespace](#)

Boardblanking Property

Zero blanking mode. Allowed values are `None`, `Leading`, and `Full`.

Namespace: [IO.Devices.ClickBoards.RemoteIO.SevenSegment](#)

Assembly: libremoteio (in libremoteio.dll) Version: 2.2020.136.3

↳ Syntax

C# VB F#

[Copy](#)

```
public BoardZeroBlanking blanking { get; set; }
```

Property Value

Type: [BoardZeroBlanking](#)

↳ See Also

Reference

[Board Class](#)

[IO.Devices.ClickBoards.RemoteIO.SevenSegment Namespace](#)

Boardbrightness Property

Write-only property for setting the brightness of the display. Allowed values are 0.0 to 100.0 percent.

Namespace: [IO.Devices.ClickBoards.RemotelO.SevenSegment](#)

Assembly: libremoteio (in libremoteio.dll) Version: 2.2020.136.3

↳ Syntax

C# VB F#

[Copy](#)

```
public double brightness { set; }
```

Property Value

Type: [Double](#)

↳ See Also

Reference

[Board Class](#)

[IO.Devices.ClickBoards.RemotelO.SevenSegment Namespace](#)

Boardleftdp Property

Write-only property for setting the left digit decimal point.

Namespace: [IO.Devices.ClickBoards.RemoteIO.SevenSegment](#)

Assembly: libremoteio (in libremoteio.dll) Version: 2.2020.136.3

↳ Syntax

C# VB F#

[Copy](#)

```
public bool leftdp { set; }
```

Property Value

Type: [Boolean](#)

↳ See Also

Reference

[Board Class](#)

[IO.Devices.ClickBoards.RemoteIO.SevenSegment Namespace](#)

Boardradix Property

Numerical base or radix. Allowed values are [Decimal](#) and [Hexadecimal](#).

Namespace: [IO.Devices.ClickBoards.RemoteIO.SevenSegment](#)

Assembly: libremoteio (in libremoteio.dll) Version: 2.2020.136.3

« Syntax

C# VB F#

[Copy](#)

```
public BoardBase radix { get; set; }
```

Property Value

Type: [BoardBase](#)

« See Also

Reference

[Board Class](#)

[IO.Devices.ClickBoards.RemoteIO.SevenSegment Namespace](#)

Boardrightdp Property

Write-only property for setting the right digit decimal point.

Namespace: [IO.Devices.ClickBoards.RemoteIO.SevenSegment](#)

Assembly: libremoteio (in libremoteio.dll) Version: 2.2020.136.3

↳ Syntax

C# VB F#

[Copy](#)

```
public bool rightdp { set; }
```

Property Value

Type: [Boolean](#)

↳ See Also

Reference

[Board Class](#)

[IO.Devices.ClickBoards.RemoteIO.SevenSegment Namespace](#)

Boardstate Property

Write-only property for setting the state of the display. Allowed values are 0 to 99 for decimal mode and 0 to 255 for hexadecimal mode.

Namespace: [IO.Devices.ClickBoards.RemotelO.SevenSegment](#)

Assembly: libremoteio (in libremoteio.dll) Version: 2.2020.136.3

↳ Syntax

[C#](#) [VB](#) [F#](#)

[Copy](#)

```
public int state { set; }
```

Property Value

Type: [Int32](#)

↳ See Also

Reference

[Board Class](#)

[IO.Devices.ClickBoards.RemotelO.SevenSegment Namespace](#)

Board Methods

The [Board](#) type exposes the following members.

↳ Methods

	Name	Description
	Clear	Clear the display.
	Reset	Issue hardware reset to the 74HC595 shift register chain.

[Top](#)

↳ See Also

[Reference](#)

[Board Class](#)

[IO.Devices.ClickBoards.RemoteIO.SevenSegment Namespace](#)

BoardClear Method

Clear the display.

Namespace: [IO.Devices.ClickBoards.RemotelO.SevenSegment](#)

Assembly: libremoteio (in libremoteio.dll) Version: 2.2020.136.3

◀ Syntax

C# VB F#

[Copy](#)

```
public void Clear()
```

◀ See Also

Reference

[Board Class](#)

[IO.Devices.ClickBoards.RemotelO.SevenSegment Namespace](#)

BoardReset Method

Issue hardware reset to the 74HC595 shift register chain.

Namespace: [IO.Devices.ClickBoards.RemoteIO.SevenSegment](#)

Assembly: libremoteio (in libremoteio.dll) Version: 2.2020.136.3

▪ Syntax

C# VB F#

[Copy](#)

```
public void Reset()
```

▪ See Also

Reference

[Board Class](#)

[IO.Devices.ClickBoards.RemoteIO.SevenSegment Namespace](#)

BoardBase Enumeration

Numerical systems.

Namespace: [IO.Devices.ClickBoards.RemotelO.SevenSegment](#)

Assembly: libremoteio (in libremoteio.dll) Version: 2.2020.136.3

▪ Syntax

C# VB F#

[Copy](#)

```
public enum Base
```

▪ Members

Member name	Value	Description
Decimal	0	Base 10.
Hexadecimal	1	Base 16.

▪ See Also

Reference

[IO.Devices.ClickBoards.RemotelO.SevenSegment Namespace](#)

BoardZeroBlanking Enumeration

Zero blanking modes.

Namespace: [IO.Devices.ClickBoards.RemotelO.SevenSegment](#)

Assembly: libremoteio (in libremoteio.dll) Version: 2.2020.136.3

▪ Syntax

C# VB F#

[Copy](#)

```
public enum ZeroBlanking
```

▪ Members

Member name	Value	Description
None	0	No zero blanking.
Leading	1	Leading zero blanking.
Full	2	Full zero blanking.

▪ See Also

Reference

[IO.Devices.ClickBoards.RemotelO.SevenSegment Namespace](#)

IO.Devices.Grove.ADC Namespace

Seeed Studio Grove I²C ADC (ADC121C021) Services

◀ Classes

	Class	Description
	Device	Encapsulates the Seeed Studio Grove I ² C ADC (ADC121C021).

Device Class

Encapsulates the Seeed Studio Grove I²C ADC (ADC121C021).

▪ Inheritance Hierarchy

[SystemObject](#) [IO.Devices.Grove.ADCDevice](#)

Namespace: [IO.Devices.Grove.ADC](#)

Assembly: libremoteio (in libremoteio.dll) Version: 2.2020.136.3

▪ Syntax

[C#](#) [VB](#) [F#](#)

[Copy](#)

```
public class Device
```

The [Device](#) type exposes the following members.

▪ Constructors

	Name	Description
	Device	Constructor for a Seeed Studio Grove I ² C ADC (ADC121C021).

[Top](#)

▪ Properties

	Name	Description
	voltage	Read-only property returning an analog

input voltage measurement.

[Top](#)

◀ See Also

Reference

[IO.Devices.Grove.ADC Namespace](#)

Device Constructor

Constructor for a Seeed Studio Grove I²C ADC (ADC121C021).

Namespace: [IO.Devices.Grove.ADC](#)

Assembly: libremoteio (in libremoteio.dll) Version: 2.2020.136.3

▪ Syntax

C# VB F#

[Copy](#)

```
public Device(  
    Bus bus,  
    byte addr = 80  
)
```

Parameters

bus

Type: [IO.Interfaces.I2CBus](#)

I²C bus object.

addr (Optional)

Type: [SystemByte](#)

I²C device address.

▪ See Also

[Reference](#)

[Device Class](#)

[IO.Devices.Grove.ADC Namespace](#)

Device Properties

The [Device](#) type exposes the following members.

▪ Properties

	Name	Description
	voltage	Read-only property returning an analog input voltage measurement.

[Top](#)

▪ See Also

[Reference](#)

[Device Class](#)

[IO.Devices.Grove.ADC Namespace](#)

Devicevoltage Property

Read-only property returning an analog input voltage measurement.

Namespace: [IO.Devices.Grove.ADC](#)

Assembly: libremoteio (in libremoteio.dll) Version: 2.2020.136.3

◀ Syntax

C# VB F#

[Copy](#)

```
public double voltage { get; }
```

Property Value

Type: [Double](#)

◀ See Also

Reference

[Device Class](#)

[IO.Devices.Grove.ADC Namespace](#)

IO.Devices.Grove.Temperature Namespace

Seeed Studio Grove Temperature Sensor (thermistor) Services

↳ Classes

Class	Description
 Device	Encapsulates the Seeed Studio Grove Temperature Sensor (thermistor).

Device Class

Encapsulates the Seeed Studio Grove Temperature Sensor (thermistor).

▪ Inheritance Hierarchy

[SystemObject](#) [IO.Devices.Grove.TemperatureDevice](#)

Namespace: [IO.Devices.Grove.Temperature](#)

Assembly: libremoteio (in libremoteio.dll) Version: 2.2020.136.3

▪ Syntax

C# VB F#

[Copy](#)

```
public class Device : Sensor
```

The [Device](#) type exposes the following members.

▪ Constructors

	Name	Description
	Device	Constructor for a Seeed Studio Grove Temperature Sensor (thermistor).

[Top](#)

▪ Properties

	Name	Description
	Celsius	Read-only property returning the

temperature in degrees Celsius.



Fahrenheit Read-only property returning the temperature in degrees Fahrenheit.



Kelvins Read-only property returning the temperature in Kelvins.

[Top](#)

◀ See Also

Reference

[IO.Devices.Grove.Temperature Namespace](#)

Device Constructor

Constructor for a Seeed Studio Grove Temperature Sensor (thermistor).

Namespace: [IO.Devices.Grove.Temperature](#)

Assembly: libremoteio (in libremoteio.dll) Version: 2.2020.136.3

► Syntax

C# VB F#

[Copy](#)

```
public Device(  
    Voltage Vin,  
    double Vcc = 3.3  
)
```

Parameters

Vin

Type: [IO.Interfaces.ADCVoltage](#)

Voltage input object.

Vcc (Optional)

Type: [SystemDouble](#)

Reference voltage.

► See Also

[Reference](#)

[Device Class](#)

[IO.Devices.Grove.Temperature Namespace](#)

Device Properties

The [Device](#) type exposes the following members.

Properties

	Name	Description
	Celsius	Read-only property returning the temperature in degrees Celsius.
	Fahrenheit	Read-only property returning the temperature in degrees Fahrenheit.
	Kelvins	Read-only property returning the temperature in Kelvins.

[Top](#)

See Also

Reference

[Device Class](#)

[IO.Devices.Grove.Temperature Namespace](#)

DeviceCelsius Property

Read-only property returning the temperature in degrees Celsius.

Namespace: [IO.Devices.Grove.Temperature](#)

Assembly: libremoteio (in libremoteio.dll) Version: 2.2020.136.3

▪ Syntax

C# VB F#

[Copy](#)

```
public double Celsius { get; }
```

Property Value

Type: [Double](#)

Implements

[SensorCelsius](#)

▪ See Also

[Reference](#)

[Device Class](#)

[IO.Devices.Grove.Temperature Namespace](#)

DeviceFahrenheit Property

Read-only property returning the temperature in degrees Fahrenheit.

Namespace: [IO.Devices.Grove.Temperature](#)

Assembly: libremoteio (in libremoteio.dll) Version: 2.2020.136.3

▪ Syntax

C# VB F#

[Copy](#)

```
public double Fahrenheit { get; }
```

Property Value

Type: [Double](#)

Implements

[SensorFahrenheit](#)

▪ See Also

[Reference](#)

[Device Class](#)

[IO.Devices.Grove.Temperature Namespace](#)

DeviceKelvins Property

Read-only property returning the temperature in Kelvins.

Namespace: [IO.Devices.Grove.Temperature](#)

Assembly: libremoteio (in libremoteio.dll) Version: 2.2020.136.3

▪ Syntax

C# VB F#

[Copy](#)

```
public double Kelvins { get; }
```

Property Value

Type: [Double](#)

Implements

[SensorKelvins](#)

▪ See Also

[Reference](#)

[Device Class](#)

[IO.Devices.Grove.Temperature Namespace](#)

IO.Devices.Grove.Temperature_Humidity Namespace

Seeed Studio Grove Temperature and Humdity Sensor (TH02) Services.

▪ Classes

	Class	Description
	Device	Encapsulate the Seeed Studio Grove Temperature and Humidity Sensor (TH02).

Device Class

Encapsulate the Seeed Studio Grove Temperature and Humidity Sensor (TH02).

▪ Inheritance Hierarchy

```
SystemObject IO.Devices.TH02Device
    IO.Devices.Grove.Temperature_HumidityDevice
```

Namespace: [IO.Devices.Grove.Temperature_Humidity](#)

Assembly: libremoteio (in libremoteio.dll) Version: 2.2020.136.3

▪ Syntax

C# VB F#

[Copy](#)

```
public class Device : Device
```

The [Device](#) type exposes the following members.

▪ Constructors

Name	Description
 Device	Constructor for a Seeed Studio Grove Temperature and Humidity Sensor (TH02).

[Top](#)

▪ Properties

	Name	Description
	Celsius	Read-only property returning the temperature in degrees Celsius. (Inherited from Device .)
	DeviceID	Read-only property returning the device ID. (Inherited from Device .)
	Fahrenheit	Read-only property returning the temperature in degrees Fahrenheit. (Inherited from Device .)
	Humidity	Read-only property returning the percentage relative humidity. (Inherited from Device .)
	Kelvins	Read-only property returning the temperature in Kelvins. (Inherited from Device .)

[Top](#)

See Also

Reference

[IO.Devices.Grove.Temperature_Humidity Namespace](#)

Device Constructor

Constructor for a Seeed Studio Grove Temperature and Humidity Sensor (TH02).

Namespace: [IO.Devices.Grove.Temperature_Humidity](#)

Assembly: libremoteio (in libremoteio.dll) Version: 2.2020.136.3

◀ Syntax

C# VB F#

[Copy](#)

```
public Device(  
    Bus bus  
)
```

Parameters

bus

Type: [IO.Interfaces.I2CBus](#)

I² bus object.

◀ See Also

Reference

[Device Class](#)

[IO.Devices.Grove.Temperature_Humidity Namespace](#)

Device Properties

The [Device](#) type exposes the following members.

Properties

	Name	Description
	Celsius	Read-only property returning the temperature in degrees Celsius. (Inherited from Device .)
	DeviceID	Read-only property returning the device ID. (Inherited from Device .)
	Fahrenheit	Read-only property returning the temperature in degrees Fahrenheit. (Inherited from Device .)
	Humidity	Read-only property returning the percentage relative humidity. (Inherited from Device .)
	Kelvins	Read-only property returning the temperature in Kelvins. (Inherited from Device .)

[Top](#)

See Also

Reference

Device Class

[IO.Devices.Grove.Temperature_Humidity Namespace](#)

IO.Devices.HDC1080 Namespace

HDC1080 I²C Temperature/Humidity Sensor Services

» Classes

	Class	Description
	Device	Encapsulates the HDC1080 temperature and humidity sensor.

Device Class

Encapsulates the HDC1080 temperature and humidity sensor.

▪ Inheritance Hierarchy

```
SystemObject IO.Devices.HDC1080Device
IO.Devices.Pmod.HYGRODevice
```

Namespace: [IO.Devices.HDC1080](#)

Assembly: libremoteio (in libremoteio.dll) Version: 2.2020.136.3

▪ Syntax

C# VB F#

[Copy](#)

```
public class Device : Sensor, Sensor
```

The [Device](#) type exposes the following members.

▪ Constructors

	Name	Description
•	Device	Constructor for an HDC1080 temperature and humidity sensor object.

[Top](#)

▪ Properties

	Name	Description
--	------	-------------

	Celsius	Read-only property returning the temperature in degrees Celsius.
	DeviceID	Read-only property returning the device ID.
	Fahrenheit	Read-only property returning the temperature in degrees Fahrenheit.
	Humidity	Read-only property returning the percentage relative humidity.
	Kelvins	Read-only property returning the temperature in Kelvins.
	ManufacturerID	Read-only property returning the manufacturer ID.

[Top](#)

◀ Methods

	Name	Description
	Read	Read from an HDC1080 device register.
	Write	Write to an HDC1080 device register.

[Top](#)

◀ Fields

	Name	Description
	RegConfiguration	Configuration Register

address.

• S	RegDeviceID	Device ID Register address.
• S	RegHumidity	Humidity Register address.
• S	RegManufacturerID	Manufacturer ID Register address.
• S	RegSerialNumberFirst	Serial Number First Bits Register address.
• S	RegSerialNumberLast	Serial Number Last Bits Register address.
• S	RegSerialNumberMid	Serial Number Middle Bits Register address.
• S	RegTemperature	Temperature Register address.

[Top](#)

See Also

Reference

[IO.Devices.HDC1080 Namespace](#)

Device Constructor

Constructor for an HDC1080 temperature and humidity sensor object.

Namespace: [IO.Devices.HDC1080](#)

Assembly: libremoteio (in libremoteio.dll) Version: 2.2020.136.3

▪ Syntax

C# VB F#

[Copy](#)

```
public Device(  
    Bus bus  
)
```

Parameters

bus

Type: [IO.Interfaces.I2CBus](#)

I²C bus controller.

▪ See Also

Reference

[Device Class](#)

[IO.Devices.HDC1080 Namespace](#)

Device Properties

The [Device](#) type exposes the following members.

Properties

	Name	Description
	Celsius	Read-only property returning the temperature in degrees Celsius.
	DeviceID	Read-only property returning the device ID.
	Fahrenheit	Read-only property returning the temperature in degrees Fahrenheit.
	Humidity	Read-only property returning the percentage relative humidity.
	Kelvins	Read-only property returning the temperature in Kelvins.
	ManufacturerID	Read-only property returning the manufacturer ID.

[Top](#)

See Also

Reference

Device Class IO.Devices.HDC1080 Namespace

DeviceCelsius Property

Read-only property returning the temperature in degrees Celsius.

Namespace: [IO.Devices.HDC1080](#)

Assembly: libremoteio (in libremoteio.dll) Version: 2.2020.136.3

▪ Syntax

C# VB F#

[Copy](#)

```
public double Celsius { get; }
```

Property Value

Type: [Double](#)

Implements

[SensorCelsius](#)

▪ See Also

[Reference](#)

[Device Class](#)

[IO.Devices.HDC1080 Namespace](#)

DeviceDeviceID Property

Read-only property returning the device ID.

Namespace: [IO.Devices.HDC1080](#)

Assembly: libremoteio (in libremoteio.dll) Version: 2.2020.136.3

◀ Syntax

C# VB F#

[Copy](#)

```
public ushort DeviceID { get; }
```

Property Value

Type: [UInt16](#)

◀ See Also

Reference

[Device Class](#)

[IO.Devices.HDC1080 Namespace](#)

DeviceFahrenheit Property

Read-only property returning the temperature in degrees Fahrenheit.

Namespace: [IO.Devices.HDC1080](#)

Assembly: libremoteio (in libremoteio.dll) Version: 2.2020.136.3

▪ Syntax

C# VB F#

[Copy](#)

```
public double Fahrenheit { get; }
```

Property Value

Type: [Double](#)

Implements

[SensorFahrenheit](#)

▪ See Also

[Reference](#)

[Device Class](#)

[IO.Devices.HDC1080 Namespace](#)

DeviceHumidity Property

Read-only property returning the percentage relative humidity.

Namespace: [IO.Devices.HDC1080](#)

Assembly: libremoteio (in libremoteio.dll) Version: 2.2020.136.3

▪ Syntax

C# VB F#

[Copy](#)

```
public double Humidity { get; }
```

Property Value

Type: [Double](#)

Implements

[SensorHumidity](#)

▪ See Also

[Reference](#)

[Device Class](#)

[IO.Devices.HDC1080 Namespace](#)

DeviceKelvins Property

Read-only property returning the temperature in Kelvins.

Namespace: [IO.Devices.HDC1080](#)

Assembly: libremoteio (in libremoteio.dll) Version: 2.2020.136.3

▪ Syntax

C# VB F#

[Copy](#)

```
public double Kelvins { get; }
```

Property Value

Type: [Double](#)

Implements

[SensorKelvins](#)

▪ See Also

[Reference](#)

[Device Class](#)

[IO.Devices.HDC1080 Namespace](#)

DeviceManufacturerID Property

Read-only property returning the manufacturer ID.

Namespace: [IO.Devices.HDC1080](#)

Assembly: libremoteio (in libremoteio.dll) Version: 2.2020.136.3

◀ Syntax

C# VB F#

[Copy](#)

```
public ushort ManufacturerID { get; }
```

Property Value

Type: [UInt16](#)

◀ See Also

Reference

[Device Class](#)

[IO.Devices.HDC1080 Namespace](#)

Device Methods

The [Device](#) type exposes the following members.

▪ Methods

	Name	Description
	Read	Read from an HDC1080 device register.
	Write	Write to an HDC1080 device register.

[Top](#)

▪ See Also

[Reference](#)

[Device Class](#)

[IO.Devices.HDC1080 Namespace](#)

DeviceRead Method

Read from an HDC1080 device register.

Namespace: [IO.Devices.HDC1080](#)

Assembly: libremoteio (in libremoteio.dll) Version: 2.2020.136.3

► Syntax

C# VB F#

[Copy](#)

```
public ushort Read(  
    byte reg  
)
```

Parameters

reg

Type: [SystemByte](#)

8-bit register address.

Return Value

Type: [UInt16](#)

16-bit register data

► See Also

[Reference](#)

[Device Class](#)

[IO.Devices.HDC1080 Namespace](#)

DeviceWrite Method

Write to an HDC1080 device register.

Namespace: [IO.Devices.HDC1080](#)

Assembly: libremoteio (in libremoteio.dll) Version: 2.2020.136.3

« Syntax

C# VB F#

[Copy](#)

```
public void Write(  
    byte reg,  
    ushort data  
)
```

Parameters

reg

Type: [SystemByte](#)

8-bit register address.

data

Type: [SystemUInt16](#)

16-bit register data.

« See Also

[Reference](#)

[Device Class](#)

[IO.Devices.HDC1080 Namespace](#)

Device Fields

The [Device](#) type exposes the following members.

Fields

Name	Description
RegConfiguration	Configuration Register address.
RegDeviceID	Device ID Register address.
RegHumidity	Humidity Register address.
RegManufacturerID	Manufacturer ID Register address.
RegSerialNumberFirst	Serial Number First Bits Register address.
RegSerialNumberLast	Serial Number Last Bits Register address.
RegSerialNumberMid	Serial Number Middle Bits Register address.
RegTemperature	Temperature Register address.

[Top](#)

See Also

[Reference](#)

[Device Class](#)

[IO.Devices.HDC1080 Namespace](#)

DeviceRegConfiguration Field

Configuration Register address.

Namespace: [IO.Devices.HDC1080](#)

Assembly: libremoteio (in libremoteio.dll) Version: 2.2020.136.3

◀ Syntax

C# VB F#

[Copy](#)

```
public const byte RegConfiguration = 2
```

Field Value

Type: [Byte](#)

◀ See Also

Reference

[Device Class](#)

[IO.Devices.HDC1080 Namespace](#)

DeviceRegDeviceID Field

Device ID Register address.

Namespace: [IO.Devices.HDC1080](#)

Assembly: libremoteio (in libremoteio.dll) Version: 2.2020.136.3

◀ Syntax

C# VB F#

[Copy](#)

```
public const byte RegDeviceID = 255
```

Field Value

Type: [Byte](#)

◀ See Also

Reference

[Device Class](#)

[IO.Devices.HDC1080 Namespace](#)

DeviceRegHumidity Field

Humidity Register address.

Namespace: [IO.Devices.HDC1080](#)

Assembly: libremoteio (in libremoteio.dll) Version: 2.2020.136.3

◀ Syntax

C# VB F#

[Copy](#)

```
public const byte RegHumidity = 1
```

Field Value

Type: [Byte](#)

◀ See Also

Reference

[Device Class](#)

[IO.Devices.HDC1080 Namespace](#)

DeviceRegManufacturerID Field

Manufacturer ID Register address.

Namespace: [IO.Devices.HDC1080](#)

Assembly: libremoteio (in libremoteio.dll) Version: 2.2020.136.3

◀ Syntax

C# VB F#

[Copy](#)

```
public const byte RegManufacturerID = 254
```

Field Value

Type: [Byte](#)

◀ See Also

Reference

[Device Class](#)

[IO.Devices.HDC1080 Namespace](#)

DeviceRegSerialNumberFirst Field

Serial Number First Bits Register address.

Namespace: [IO.Devices.HDC1080](#)

Assembly: libremoteio (in libremoteio.dll) Version: 2.2020.136.3

◀ Syntax

C# VB F#

[Copy](#)

```
public const byte RegSerialNumberFirst = 251
```

Field Value

Type: [Byte](#)

◀ See Also

Reference

[Device Class](#)

[IO.Devices.HDC1080 Namespace](#)

DeviceRegSerialNumberLast Field

Serial Number Last Bits Register address.

Namespace: [IO.Devices.HDC1080](#)

Assembly: libremoteio (in libremoteio.dll) Version: 2.2020.136.3

◀ Syntax

C# VB F#

[Copy](#)

```
public const byte RegSerialNumberLast = 253
```

Field Value

Type: [Byte](#)

◀ See Also

Reference

[Device Class](#)

[IO.Devices.HDC1080 Namespace](#)

DeviceRegSerialNumberMid Field

Serial Number Middle Bits Register address.

Namespace: [IO.Devices.HDC1080](#)

Assembly: libremoteio (in libremoteio.dll) Version: 2.2020.136.3

◀ Syntax

C# VB F#

[Copy](#)

```
public const byte RegSerialNumberMid = 252
```

Field Value

Type: [Byte](#)

◀ See Also

Reference

[Device Class](#)

[IO.Devices.HDC1080 Namespace](#)

DeviceRegTemperature Field

Temperature Register address.

Namespace: [IO.Devices.HDC1080](#)

Assembly: libremoteio (in libremoteio.dll) Version: 2.2020.136.3

« Syntax

C# VB F#

[Copy](#)

```
public const byte RegTemperature = 0
```

Field Value

Type: [Byte](#)

« See Also

Reference

[Device Class](#)

[IO.Devices.HDC1080 Namespace](#)

IO.Devices.MCP23017 Namespace

MCP23017 I²C GPIO Expander Device Services.

↳ Classes

	Class	Description
	Device	Encapsulates the MCP23017 I ² C I/O GPIO Expander.

Device Class

Encapsulates the MCP23017 I²C I/O GPIO Expander.

▪ Inheritance Hierarchy

[SystemObject](#) [IO.Devices.MCP23017Device](#)

Namespace: [IO.Devices.MCP23017](#)

Assembly: libremoteio (in libremoteio.dll) Version: 2.2020.136.3

▪ Syntax

[C#](#) [VB](#) [F#](#)

[Copy](#)

```
public class Device
```

The [Device](#) type exposes the following members.

▪ Constructors

	Name	Description
	Device	Constructor for a single MCP23017 device.

[Top](#)

▪ Properties

	Name	Description
	Direction	Data Direction Property (16 bits). Bits 0

to 7 correspond to PORT A and bits 8 to 15 correspond to PORT B. For each bit, 0=input and 1=output.

	DirectionA	Port A Data Direction Property (8 bits). For each bit, 0=input and 1=output.
	DirectionB	Port B Data Direction Property (8 bits). For each bit, 0=input and 1=output.
	Polarity	Data Polarity Property (16 bits). Bits 0 to 7 correspond to PORT A and bits 8 to 15 correspond to PORT B. For each bit, 0=input and 1=output.
	PolarityA	Port A Data Polarity Property (8 bits). For each bit, 0=input and 1=output.
	PolarityB	Port B Data Polarity Property (8 bits). For each bit, 0=input and 1=output.
	Port	Port Data Property (16 bites). Bits 0 to 7 correspond to PORT A and bits 8 to 15 correspond to PORT B.
	PortA	Port A Data Property (8 bits).
	PortB	Port B Data Property (8 bits).
	Pullups	Input Pullup Property (16 bits). Bits 0 to 7 correspond to PORT A and bits 8 to 15 correspond to PORT B. For each bit, 0=high impedance and 1=100k pullup.

	PullupsA	Port A Input Pullup Property (16 bits). For each bit, 0=high impedance and 1=100k pullup.
	PullupsB	Port B Input Pullup Property (16 bits). For each bit, 0=high impedance and 1=100k pullup.

[Top](#)

◀ Methods

	Name	Description
	GPIO_Create	Create an MCP23017 GPIO pin object.

[Top](#)

◀ Fields

	Name	Description
	MaxChannel	Maximum I/O channel number.
	MinChannel	Minimum I/O channel number.

[Top](#)

◀ See Also

Reference

[IO.Devices.MCP23017 Namespace](#)

Device Constructor

Constructor for a single MCP23017 device.

Namespace: [IO.Devices.MCP23017](#)

Assembly: libremoteio (in libremoteio.dll) Version: 2.2020.136.3

▪ Syntax

C# VB F#

[Copy](#)

```
public Device(  
    Bus bus,  
    int addr  
)
```

Parameters

bus

Type: [IO.Interfaces.I2CBus](#)

I²C bus controller object.

addr

Type: [SystemInt32](#)

I²C slave address.

▪ See Also

[Reference](#)

[Device Class](#)

[IO.Devices.MCP23017 Namespace](#)

Device Properties

The [Device](#) type exposes the following members.

Properties

	Name	Description
	Direction	Data Direction Property (16 bits). Bits 0 to 7 correspond to PORT A and bits 8 to 15 correspond to PORT B. For each bit, 0=input and 1=output.
	DirectionA	Port A Data Direction Property (8 bits). For each bit, 0=input and 1=output.
	DirectionB	Port B Data Direction Property (8 bits). For each bit, 0=input and 1=output.
	Polarity	Data Polarity Property (16 bits). Bits 0 to 7 correspond to PORT A and bits 8 to 15 correspond to PORT B. For each bit, 0=input and 1=output.
	PolarityA	Port A Data Polarity Property (8 bits). For each bit, 0=input and 1=output.
	PolarityB	Port B Data Polarity Property (8 bits). For each bit, 0=input and 1=output.
	Port	Port Data Property (16 bites). Bits 0 to 7 correspond to PORT A and bits 8 to

15 correspond to PORT B.

	PortA	Port A Data Property (8 bits).
	PortB	Port B Data Property (8 bits).
	Pullups	Input Pullup Property (16 bits). Bits 0 to 7 correspond to PORT A and bits 8 to 15 correspond to PORT B. For each bit, 0=high impedance and 1=100k pullup.
	PullupsA	Port A Input Pullup Property (16 bits). For each bit, 0=high impedance and 1=100k pullup.
	PullupsB	Port B Input Pullup Property (16 bits). For each bit, 0=high impedance and 1=100k pullup.

[Top](#)

See Also

[Reference](#)

[Device Class](#)

[IO.Devices.MCP23017 Namespace](#)

DeviceDirection Property

Data Direction Property (16 bits). Bits 0 to 7 correspond to PORT A and bits 8 to 15 correspond to PORT B. For each bit, 0=input and 1=output.

Namespace: [IO.Devices.MCP23017](#)

Assembly: libremoteio (in libremoteio.dll) Version: 2.2020.136.3

► Syntax

C# **VB** **F#**

[Copy](#)

```
public uint Direction { get; set; }
```

Property Value

Type: [UInt32](#)

► Remarks

This property follows the industry standard convention for data direction bit polarity (1=output) rather than the MCP23017 [IODIR](#) register polarity (0=output).

► See Also

Reference

[Device Class](#)

[IO.Devices.MCP23017 Namespace](#)

DeviceDirectionA Property

Port A Data Direction Property (8 bits). For each bit, 0=input and 1=output.

Namespace: [IO.Devices.MCP23017](#)

Assembly: libremoteio (in libremoteio.dll) Version: 2.2020.136.3

▪ Syntax

C# VB F#

[Copy](#)

```
public byte DirectionA { get; set; }
```

Property Value

Type: [Byte](#)

▪ Remarks

This property follows the industry standard convention for data direction bit polarity (1=output) rather than the MCP23017 [IODIRA](#) register polarity (0=output).

▪ See Also

Reference

[Device Class](#)

[IO.Devices.MCP23017 Namespace](#)

DeviceDirectionB Property

Port B Data Direction Property (8 bits). For each bit, 0=input and 1=output.

Namespace: [IO.Devices.MCP23017](#)

Assembly: libremoteio (in libremoteio.dll) Version: 2.2020.136.3

▪ Syntax

C# VB F#

[Copy](#)

```
public byte DirectionB { get; set; }
```

Property Value

Type: [Byte](#)

▪ Remarks

This property follows the industry standard convention for data direction bit polarity (1=output) rather than the MCP23017 [IODIRA](#) register polarity (0=output).

▪ See Also

Reference

[Device Class](#)

[IO.Devices.MCP23017 Namespace](#)

DevicePolarity Property

Data Polarity Property (16 bits). Bits 0 to 7 correspond to PORT A and bits 8 to 15 correspond to PORT B. For each bit, 0=input and 1=output.

Namespace: [IO.Devices.MCP23017](#)

Assembly: libremoteio (in libremoteio.dll) Version: 2.2020.136.3

► Syntax

[C#](#) [VB](#) [F#](#)

[Copy](#)

```
public uint Polarity { get; set; }
```

Property Value

Type: [UInt32](#)

► See Also

Reference

[Device Class](#)

[IO.Devices.MCP23017 Namespace](#)

DevicePolarityA Property

Port A Data Polarity Property (8 bits). For each bit, 0=input and 1=output.

Namespace: [IO.Devices.MCP23017](#)

Assembly: libremoteio (in libremoteio.dll) Version: 2.2020.136.3

◀ Syntax

[C#](#) [VB](#) [F#](#)

[Copy](#)

```
public byte PolarityA { get; set; }
```

Property Value

Type: [Byte](#)

◀ See Also

Reference

[Device Class](#)

[IO.Devices.MCP23017 Namespace](#)

DevicePolarityB Property

Port B Data Polarity Property (8 bits). For each bit, 0=input and 1=output.

Namespace: [IO.Devices.MCP23017](#)

Assembly: libremoteio (in libremoteio.dll) Version: 2.2020.136.3

◀ Syntax

C# VB F#

[Copy](#)

```
public byte PolarityB { get; set; }
```

Property Value

Type: [Byte](#)

◀ See Also

Reference

[Device Class](#)

[IO.Devices.MCP23017 Namespace](#)

DevicePort Property

Port Data Property (16 bites). Bits 0 to 7 correspond to PORT A and bits 8 to 15 correspond to PORT B.

Namespace: [IO.Devices.MCP23017](#)

Assembly: libremoteio (in libremoteio.dll) Version: 2.2020.136.3

◀ Syntax

C# VB F#

[Copy](#)

```
public uint Port { get; set; }
```

Property Value

Type: [UInt32](#)

◀ See Also

Reference

[Device Class](#)

[IO.Devices.MCP23017 Namespace](#)

DevicePortA Property

Port A Data Property (8 bits).

Namespace: [IO.Devices.MCP23017](#)

Assembly: libremoteio (in libremoteio.dll) Version: 2.2020.136.3

« Syntax

C# VB F#

[Copy](#)

```
public byte PortA { get; set; }
```

Property Value

Type: [Byte](#)

« See Also

Reference

[Device Class](#)

[IO.Devices.MCP23017 Namespace](#)

DevicePortB Property

Port B Data Property (8 bits).

Namespace: [IO.Devices.MCP23017](#)

Assembly: libremoteio (in libremoteio.dll) Version: 2.2020.136.3

« Syntax

C# VB F#

[Copy](#)

```
public byte PortB { get; set; }
```

Property Value

Type: [Byte](#)

« See Also

Reference

[Device Class](#)

[IO.Devices.MCP23017 Namespace](#)

DevicePullups Property

Input Pullup Property (16 bits). Bits 0 to 7 correspond to PORT A and bits 8 to 15 correspond to PORT B. For each bit, 0=high impedance and 1=100k pullup.

Namespace: [IO.Devices.MCP23017](#)

Assembly: libremoteio (in libremoteio.dll) Version: 2.2020.136.3

↳ Syntax

C# VB F#

[Copy](#)

```
public uint Pullups { get; set; }
```

Property Value

Type: [UInt32](#)

↳ See Also

[Reference](#)

[Device Class](#)

[IO.Devices.MCP23017 Namespace](#)

DevicePullupsA Property

Port A Input Pullup Property (16 bits). For each bit, 0=high impedance and 1=100k pullup.

Namespace: [IO.Devices.MCP23017](#)

Assembly: libremoteio (in libremoteio.dll) Version: 2.2020.136.3

◀ Syntax

C# VB F#

[Copy](#)

```
public byte PullupsA { get; set; }
```

Property Value

Type: [Byte](#)

◀ See Also

Reference

[Device Class](#)

[IO.Devices.MCP23017 Namespace](#)

DevicePullupsB Property

Port B Input Pullup Property (16 bits). For each bit, 0=high impedance and 1=100k pullup.

Namespace: [IO.Devices.MCP23017](#)

Assembly: libremoteio (in libremoteio.dll) Version: 2.2020.136.3

◀ Syntax

C# VB F#

[Copy](#)

```
public byte PullupsB { get; set; }
```

Property Value

Type: [Byte](#)

◀ See Also

Reference

[Device Class](#)

[IO.Devices.MCP23017 Namespace](#)

Device Methods

The [Device](#) type exposes the following members.

▪ Methods

	Name	Description
	GPIO_Create	Create an MCP23017 GPIO pin object.

[Top](#)

▪ See Also

[Reference](#)

[Device Class](#)

[IO.Devices.MCP23017 Namespace](#)

DeviceGPIO_Create Method

Create an MCP23017 GPIO pin object.

Namespace: [IO.Devices.MCP23017](#)

Assembly: libremoteio (in libremoteio.dll) Version: 2.2020.136.3

◀ Syntax

C# VB F#

[Copy](#)

```
public Pin GPIO_Create(  
    int channel,  
    Direction dir,  
    bool state = false  
)
```

Parameters

channel

Type: [SystemInt32](#)

MCP23017 channel number (0 to 15).

dir

Type: [IO.Interfaces.GPIODirection](#)

GPIO pin data direction.

state (Optional)

Type: [SystemBoolean](#)

Initial GPIO output state.

Return Value

Type: [Pin](#)

GPIO pin object.

↳ See Also

[Reference](#)

[Device Class](#)

[IO.Devices.MCP23017 Namespace](#)

Device Fields

The [Device](#) type exposes the following members.

Fields

	Name	Description
• 	MaxChannel	Maximum I/O channel number.
• 	MinChannel	Minimum I/O channel number.

[Top](#)

See Also

[Reference](#)

[Device Class](#)

[IO.Devices.MCP23017 Namespace](#)

DeviceMaxChannel Field

Maximum I/O channel number.

Namespace: [IO.Devices.MCP23017](#)

Assembly: libremoteio (in libremoteio.dll) Version: 2.2020.136.3

◀ Syntax

C# VB F#

[Copy](#)

```
public const int MaxChannel = 15
```

Field Value

Type: [Int32](#)

◀ See Also

Reference

[Device Class](#)

[IO.Devices.MCP23017 Namespace](#)

DeviceMinChannel Field

Minimum I/O channel number.

Namespace: [IO.Devices.MCP23017](#)

Assembly: libremoteio (in libremoteio.dll) Version: 2.2020.136.3

◀ Syntax

C# VB F#

[Copy](#)

```
public const int MinChannel = 0
```

Field Value

Type: [Int32](#)

◀ See Also

Reference

[Device Class](#)

[IO.Devices.MCP23017 Namespace](#)

IO.Devices.MCP23017.GPIO Namespace

MCP23017 I²C GPIO Expander GPIO Pin Services.

↳ Classes

	Class	Description
	Pin	Encapsulates MCP23017 GPIO pins.

Pin Class

Encapsulates MCP23017 GPIO pins.

▪ Inheritance Hierarchy

[SystemObject](#) [IO.Devices.MCP23017.GPIOPin](#)

Namespace: [IO.Devices.MCP23017.GPIO](#)

Assembly: libremoteio (in libremoteio.dll) Version: 2.2020.136.3

▪ Syntax

C# VB F#

[Copy](#)

```
public class Pin : Pin
```

The [Pin](#) type exposes the following members.

▪ Constructors

	Name	Description
	Pin	Create a single MCP23017 GPIO pin.

[Top](#)

▪ Properties

	Name	Description
	state	Read/Write GPIO state property.

[Top](#)

◀ See Also

Reference

[IO.Devices.MCP23017.GPIO Namespace](#)

Pin Constructor

Create a single MCP23017 GPIO pin.

Namespace: [IO.Devices.MCP23017.GPIO](#)

Assembly: libremoteio (in libremoteio.dll) Version: 2.2020.136.3

◀ Syntax

C# VB F#

[Copy](#)

```
public Pin(  
    Device dev,  
    int channel,  
    Direction dir,  
    bool state = false  
)
```

Parameters

dev

Type: [IO.Devices.MCP23017Device](#)

MCP23017 device object.

channel

Type: [SystemInt32](#)

MCP23017 I/O channel number.

dir

Type: [IO.Interfaces.GPIODirection](#)

GPIO pin data direction.

state (Optional)

Type: [SystemBoolean](#)

Initial GPIO output state.

↳ See Also

Reference

[Pin Class](#)

[IO.Devices.MCP23017.GPIO Namespace](#)

Pin Properties

The [Pin](#) type exposes the following members.

Properties

	Name	Description
	state	Read/Write GPIO state property.

[Top](#)

See Also

Reference

[Pin Class](#)

[IO.Devices.MCP23017.GPIO Namespace](#)

Pinstate Property

Read/Write GPIO state property.

Namespace: [IO.Devices.MCP23017.GPIO](#)

Assembly: libremoteio (in libremoteio.dll) Version: 2.2020.136.3

↳ Syntax

C# VB F#

[Copy](#)

```
public bool state { get; set; }
```

Property Value

Type: [Boolean](#)

Implements

[Pinstate](#)

↳ See Also

[Reference](#)

[Pin Class](#)

[IO.Devices.MCP23017.GPIO Namespace](#)

IO.Devices.MCP23S17 Namespace

MCP23S17 SPI GPIO Expander Device Services.

↳ Classes

	Class	Description
	Device	Encapsulates the MCP23S17 SPI I/O GPIO Expander.

Device Class

Encapsulates the MCP23S17 SPI I/O GPIO Expander.

▪ Inheritance Hierarchy

[SystemObject](#) [IO.Devices.MCP23S17Device](#)

Namespace: [IO.Devices.MCP23S17](#)

Assembly: libremoteio (in libremoteio.dll) Version: 2.2020.136.3

▪ Syntax

C# VB F#

[Copy](#)

```
public class Device
```

The [Device](#) type exposes the following members.

▪ Constructors

	Name	Description
	Device	Constructor for a single MCP23S17 device.

[Top](#)

▪ Properties

	Name	Description
	Direction	Data Direction Property (16 bits). Bits 0

to 7 correspond to PORT A and bits 8 to 15 correspond to PORT B. For each bit, 0=input and 1=output.

	DirectionA	Port A Data Direction Property (8 bits). For each bit, 0=input and 1=output.
	DirectionB	Port B Data Direction Property (8 bits). For each bit, 0=input and 1=output.
	Polarity	Data Polarity Property (16 bits). Bits 0 to 7 correspond to PORT A and bits 8 to 15 correspond to PORT B. For each bit, 0=input and 1=output.
	PolarityA	Port A Data Polarity Property (8 bits). For each bit, 0=input and 1=output.
	PolarityB	Port B Data Polarity Property (8 bits). For each bit, 0=input and 1=output.
	Port	Port Data Property (16 bites). Bits 0 to 7 correspond to PORT A and bits 8 to 15 correspond to PORT B.
	PortA	Port A Data Property (8 bits).
	PortB	Port B Data Property (8 bits).
	Pullups	Input Pullup Property (16 bits). Bits 0 to 7 correspond to PORT A and bits 8 to 15 correspond to PORT B. For each bit, 0=high impedance and 1=100k pullup.

 PullupsA	Port A Input Pullup Property (16 bits). For each bit, 0=high impedance and 1=100k pullup.
 PullupsB	Port B Input Pullup Property (16 bits). For each bit, 0=high impedance and 1=100k pullup.

[Top](#)

◀ Methods

Name	Description
 GPIO_Create	Create an MCP23S17 GPIO pin object.

[Top](#)

◀ Fields

Name	Description
  MaxChannel	Maximum I/O channel number.
  MinChannel	Minimum I/O channel number.
  SPI_Frequency	SPI maximum clock frequency in Hz.
  SPI_Mode	SPI transfer mode.
  SPI_WordSize	SPI transaction word size.

[Top](#)

↳ See Also

Reference

[IO.Devices.MCP23S17 Namespace](#)

Device Constructor

Constructor for a single MCP23S17 device.

Namespace: [IO.Devices.MCP23S17](#)

Assembly: libremoteio (in libremoteio.dll) Version: 2.2020.136.3

▪ Syntax

C# VB F#

[Copy](#)

```
public Device(  
    Device dev  
)
```

Parameters

dev

Type: [IO.Interfaces.SPIDevice](#)

SPI slave device object.

▪ See Also

Reference

[Device Class](#)

[IO.Devices.MCP23S17 Namespace](#)

Device Properties

The [Device](#) type exposes the following members.

Properties

	Name	Description
	Direction	Data Direction Property (16 bits). Bits 0 to 7 correspond to PORT A and bits 8 to 15 correspond to PORT B. For each bit, 0=input and 1=output.
	DirectionA	Port A Data Direction Property (8 bits). For each bit, 0=input and 1=output.
	DirectionB	Port B Data Direction Property (8 bits). For each bit, 0=input and 1=output.
	Polarity	Data Polarity Property (16 bits). Bits 0 to 7 correspond to PORT A and bits 8 to 15 correspond to PORT B. For each bit, 0=input and 1=output.
	PolarityA	Port A Data Polarity Property (8 bits). For each bit, 0=input and 1=output.
	PolarityB	Port B Data Polarity Property (8 bits). For each bit, 0=input and 1=output.
	Port	Port Data Property (16 bites). Bits 0 to 7 correspond to PORT A and bits 8 to

15 correspond to PORT B.

	PortA	Port A Data Property (8 bits).
	PortB	Port B Data Property (8 bits).
	Pullups	Input Pullup Property (16 bits). Bits 0 to 7 correspond to PORT A and bits 8 to 15 correspond to PORT B. For each bit, 0=high impedance and 1=100k pullup.
	PullupsA	Port A Input Pullup Property (16 bits). For each bit, 0=high impedance and 1=100k pullup.
	PullupsB	Port B Input Pullup Property (16 bits). For each bit, 0=high impedance and 1=100k pullup.

[Top](#)

See Also

[Reference](#)

[Device Class](#)

[IO.Devices.MCP23S17 Namespace](#)

DeviceDirection Property

Data Direction Property (16 bits). Bits 0 to 7 correspond to PORT A and bits 8 to 15 correspond to PORT B. For each bit, 0=input and 1=output.

Namespace: [IO.Devices.MCP23S17](#)

Assembly: libremoteio (in libremoteio.dll) Version: 2.2020.136.3

▪ Syntax

C# VB F#

[Copy](#)

```
public uint Direction { get; set; }
```

Property Value

Type: [UInt32](#)

▪ Remarks

This property follows the industry standard convention for data direction bit polarity (1=output) rather than the MCP23S17 [IODIR](#) register polarity (0=output).

▪ See Also

Reference

[Device Class](#)

[IO.Devices.MCP23S17 Namespace](#)

DeviceDirectionA Property

Port A Data Direction Property (8 bits). For each bit, 0=input and 1=output.

Namespace: [IO.Devices.MCP23S17](#)

Assembly: libremoteio (in libremoteio.dll) Version: 2.2020.136.3

▪ Syntax

C# VB F#

[Copy](#)

```
public byte DirectionA { get; set; }
```

Property Value

Type: [Byte](#)

▪ Remarks

This property follows the industry standard convention for data direction bit polarity (1=output) rather than the MCP23S17 [IODIRA](#) register polarity (0=output).

▪ See Also

Reference

[Device Class](#)

[IO.Devices.MCP23S17 Namespace](#)

DeviceDirectionB Property

Port B Data Direction Property (8 bits). For each bit, 0=input and 1=output.

Namespace: [IO.Devices.MCP23S17](#)

Assembly: libremoteio (in libremoteio.dll) Version: 2.2020.136.3

▪ Syntax

C# VB F#

[Copy](#)

```
public byte DirectionB { get; set; }
```

Property Value

Type: [Byte](#)

▪ Remarks

This property follows the industry standard convention for data direction bit polarity (1=output) rather than the MCP23S17 [IODIRA](#) register polarity (0=output).

▪ See Also

Reference

[Device Class](#)

[IO.Devices.MCP23S17 Namespace](#)

DevicePolarity Property

Data Polarity Property (16 bits). Bits 0 to 7 correspond to PORT A and bits 8 to 15 correspond to PORT B. For each bit, 0=input and 1=output.

Namespace: [IO.Devices.MCP23S17](#)

Assembly: libremoteio (in libremoteio.dll) Version: 2.2020.136.3

« Syntax

[C#](#) [VB](#) [F#](#)

[Copy](#)

```
public uint Polarity { get; set; }
```

Property Value

Type: [UInt32](#)

« See Also

Reference

[Device Class](#)

[IO.Devices.MCP23S17 Namespace](#)

DevicePolarityA Property

Port A Data Polarity Property (8 bits). For each bit, 0=input and 1=output.

Namespace: [IO.Devices.MCP23S17](#)

Assembly: libremoteio (in libremoteio.dll) Version: 2.2020.136.3

◀ Syntax

C# VB F#

[Copy](#)

```
public byte PolarityA { get; set; }
```

Property Value

Type: [Byte](#)

◀ See Also

Reference

[Device Class](#)

[IO.Devices.MCP23S17 Namespace](#)

DevicePolarityB Property

Port B Data Polarity Property (8 bits). For each bit, 0=input and 1=output.

Namespace: [IO.Devices.MCP23S17](#)

Assembly: libremoteio (in libremoteio.dll) Version: 2.2020.136.3

◀ Syntax

C# VB F#

[Copy](#)

```
public byte PolarityB { get; set; }
```

Property Value

Type: [Byte](#)

◀ See Also

Reference

[Device Class](#)

[IO.Devices.MCP23S17 Namespace](#)

DevicePort Property

Port Data Property (16 bites). Bits 0 to 7 correspond to PORT A and bits 8 to 15 correspond to PORT B.

Namespace: [IO.Devices.MCP23S17](#)

Assembly: libremoteio (in libremoteio.dll) Version: 2.2020.136.3

▲ Syntax

[C#](#) [VB](#) [F#](#)

[Copy](#)

```
public uint Port { get; set; }
```

Property Value

Type: [UInt32](#)

▲ See Also

Reference

[Device Class](#)

[IO.Devices.MCP23S17 Namespace](#)

DevicePortA Property

Port A Data Property (8 bits).

Namespace: [IO.Devices.MCP23S17](#)

Assembly: libremoteio (in libremoteio.dll) Version: 2.2020.136.3

« Syntax

C# VB F#

[Copy](#)

```
public byte PortA { get; set; }
```

Property Value

Type: [Byte](#)

« See Also

Reference

[Device Class](#)

[IO.Devices.MCP23S17 Namespace](#)

DevicePortB Property

Port B Data Property (8 bits).

Namespace: [IO.Devices.MCP23S17](#)

Assembly: libremoteio (in libremoteio.dll) Version: 2.2020.136.3

« Syntax

C# VB F#

[Copy](#)

```
public byte PortB { get; set; }
```

Property Value

Type: [Byte](#)

« See Also

Reference

[Device Class](#)

[IO.Devices.MCP23S17 Namespace](#)

DevicePullups Property

Input Pullup Property (16 bits). Bits 0 to 7 correspond to PORT A and bits 8 to 15 correspond to PORT B. For each bit, 0=high impedance and 1=100k pullup.

Namespace: [IO.Devices.MCP23S17](#)

Assembly: libremoteio (in libremoteio.dll) Version: 2.2020.136.3

↳ Syntax

C# VB F#

[Copy](#)

```
public uint Pullups { get; set; }
```

Property Value

Type: [UInt32](#)

↳ See Also

[Reference](#)

[Device Class](#)

[IO.Devices.MCP23S17 Namespace](#)

DevicePullupsA Property

Port A Input Pullup Property (16 bits). For each bit, 0=high impedance and 1=100k pullup.

Namespace: [IO.Devices.MCP23S17](#)

Assembly: libremoteio (in libremoteio.dll) Version: 2.2020.136.3

◀ Syntax

[C#](#) [VB](#) [F#](#)

[Copy](#)

```
public byte PullupsA { get; set; }
```

Property Value

Type: [Byte](#)

◀ See Also

Reference

[Device Class](#)

[IO.Devices.MCP23S17 Namespace](#)

DevicePullupsB Property

Port B Input Pullup Property (16 bits). For each bit, 0=high impedance and 1=100k pullup.

Namespace: [IO.Devices.MCP23S17](#)

Assembly: libremoteio (in libremoteio.dll) Version: 2.2020.136.3

◀ Syntax

C# VB F#

[Copy](#)

```
public byte PullupsB { get; set; }
```

Property Value

Type: [Byte](#)

◀ See Also

Reference

[Device Class](#)

[IO.Devices.MCP23S17 Namespace](#)

Device Methods

The [Device](#) type exposes the following members.

▪ Methods

	Name	Description
	GPIO_Create	Create an MCP23S17 GPIO pin object.

[Top](#)

▪ See Also

[Reference](#)

[Device Class](#)

[IO.Devices.MCP23S17 Namespace](#)

DeviceGPIO_Create Method

Create an MCP23S17 GPIO pin object.

Namespace: [IO.Devices.MCP23S17](#)

Assembly: libremoteio (in libremoteio.dll) Version: 2.2020.136.3

◀ Syntax

C# VB F#

[Copy](#)

```
public Pin GPIO_Create(  
    int channel,  
    Direction dir,  
    bool state = false  
)
```

Parameters

channel

Type: [SystemInt32](#)

MCP23S17 channel number (0 to 15).

dir

Type: [IO.Interfaces.GPIODirection](#)

GPIO pin data direction.

state (Optional)

Type: [SystemBoolean](#)

Initial GPIO output state.

Return Value

Type: [Pin](#)

GPIO pin object.

↳ See Also

[Reference](#)

[Device Class](#)

[IO.Devices.MCP23S17 Namespace](#)

Device Fields

The [Device](#) type exposes the following members.

Fields

	Name	Description
• 	MaxChannel	Maximum I/O channel number.
• 	MinChannel	Minimum I/O channel number.
• 	SPI_Frequency	SPI maximum clock frequency in Hz.
• 	SPI_Mode	SPI transfer mode.
• 	SPI_WordSize	SPI transaction word size.

[Top](#)

See Also

Reference

[Device Class](#)

[IO.Devices.MCP23S17 Namespace](#)

DeviceMaxChannel Field

Maximum I/O channel number.

Namespace: [IO.Devices.MCP23S17](#)

Assembly: libremoteio (in libremoteio.dll) Version: 2.2020.136.3

◀ Syntax

C# VB F#

[Copy](#)

```
public const int MaxChannel = 15
```

Field Value

Type: [Int32](#)

◀ See Also

Reference

[Device Class](#)

[IO.Devices.MCP23S17 Namespace](#)

DeviceMinChannel Field

Minimum I/O channel number.

Namespace: [IO.Devices.MCP23S17](#)

Assembly: libremoteio (in libremoteio.dll) Version: 2.2020.136.3

◀ Syntax

C# VB F#

[Copy](#)

```
public const int MinChannel = 0
```

Field Value

Type: [Int32](#)

◀ See Also

Reference

[Device Class](#)

[IO.Devices.MCP23S17 Namespace](#)

DeviceSPI_Frequency Field

SPI maximum clock frequency in Hz.

Namespace: [IO.Devices.MCP23S17](#)

Assembly: libremoteio (in libremoteio.dll) Version: 2.2020.136.3

▪ Syntax

C# VB F#

[Copy](#)

```
public const int SPI_Frequency = 10000000
```

Field Value

Type: [Int32](#)

▪ Remarks

Guaranteed only for 2.7V to 5.5V and -40°C to +85°C.

▪ See Also

Reference

[Device Class](#)

[IO.Devices.MCP23S17 Namespace](#)

DeviceSPI_Mode Field

SPI transfer mode.

Namespace: [IO.Devices.MCP23S17](#)

Assembly: libremoteio (in libremoteio.dll) Version: 2.2020.136.3

« Syntax

C# VB F#

[Copy](#)

```
public const int SPI_Mode = 0
```

Field Value

Type: [Int32](#)

« See Also

Reference

[Device Class](#)

[IO.Devices.MCP23S17 Namespace](#)

DeviceSPI_WordSize Field

SPI transaction word size.

Namespace: [IO.Devices.MCP23S17](#)

Assembly: libremoteio (in libremoteio.dll) Version: 2.2020.136.3

◀ Syntax

C# VB F#

[Copy](#)

```
public const int SPI_WordSize = 8
```

Field Value

Type: [Int32](#)

◀ See Also

Reference

[Device Class](#)

[IO.Devices.MCP23S17 Namespace](#)

IO.Devices.MCP23S17.GPIO Namespace

MCP23S17 SPI GPIO Expander GPIO Pin Services.

◀ Classes

Class	Description
 Pin	Encapsulates MCP23S17 GPIO pins.

Pin Class

Encapsulates MCP23S17 GPIO pins.

▪ Inheritance Hierarchy

[SystemObject](#) [IO.Devices.MCP23S17.GPIOPin](#)

Namespace: [IO.Devices.MCP23S17.GPIO](#)

Assembly: libremoteio (in libremoteio.dll) Version: 2.2020.136.3

▪ Syntax

C# VB F#

[Copy](#)

```
public class Pin : Pin
```

The [Pin](#) type exposes the following members.

▪ Constructors

	Name	Description
	Pin	Create a single MCP23S17 GPIO pin.

[Top](#)

▪ Properties

	Name	Description
	state	Read/Write GPIO state property.

[Top](#)

◀ See Also

Reference

[IO.Devices.MCP23S17.GPIO Namespace](#)

Pin Constructor

Create a single MCP23S17 GPIO pin.

Namespace: [IO.Devices.MCP23S17.GPIO](#)

Assembly: libremoteio (in libremoteio.dll) Version: 2.2020.136.3

◀ Syntax

C# VB F#

[Copy](#)

```
public Pin(  
    Device dev,  
    int channel,  
    Direction dir,  
    bool state = false  
)
```

Parameters

dev

Type: [IO.Devices.MCP23S17Device](#)

MCP23S17 device object.

channel

Type: [SystemInt32](#)

MCP23S17 I/O channel number.

dir

Type: [IO.Interfaces.GPIODirection](#)

GPIO pin data direction.

state (Optional)

Type: [SystemBoolean](#)

Initial GPIO output state.

↳ See Also

Reference

[Pin Class](#)

[IO.Devices.MCP23S17.GPIO Namespace](#)

Pin Properties

The [Pin](#) type exposes the following members.

Properties

	Name	Description
	state	Read/Write GPIO state property.

[Top](#)

See Also

Reference

[Pin Class](#)

[IO.Devices.MCP23S17.GPIO Namespace](#)

Pinstate Property

Read/Write GPIO state property.

Namespace: [IO.Devices.MCP23S17.GPIO](#)

Assembly: libremoteio (in libremoteio.dll) Version: 2.2020.136.3

▪ Syntax

C# VB F#

[Copy](#)

```
public bool state { get; set; }
```

Property Value

Type: [Boolean](#)

Implements

[Pinstate](#)

▪ See Also

[Reference](#)

[Pin Class](#)

[IO.Devices.MCP23S17.GPIO Namespace](#)

IO.Devices.PCA8574 Namespace

PCA8574 (and similar) I²C GPIO Expander Device Services

↳ Classes

Class	Description
 Device	Encapsulates PCA8574 (and similar) I ² C GPIO Expanders.

Device Class

Encapsulates PCA8574 (and similar) I²C GPIO Expanders.

▪ Inheritance Hierarchy

[SystemObject](#) [IO.Devices.PCA8574Device](#)

Namespace: [IO.Devices.PCA8574](#)

Assembly: libremoteio (in libremoteio.dll) Version: 2.2020.136.3

▪ Syntax

[C#](#) [VB](#) [F#](#)

[Copy](#)

```
public class Device
```

The [Device](#) type exposes the following members.

▪ Constructors

	Name	Description
	Device	Constructor for a PCA8574 (or similar) GPIO Expander.

[Top](#)

▪ Properties

	Name	Description
	Latch	This read-only property returns the last

value written to the output latch.

[Top](#)

◀ Methods

	Name	Description
	Read	Return actual state of the GPIO pins.
	Write	Write all GPIO pins.

[Top](#)

◀ Fields

	Name	Description
 	MAX_PINS	The number of available GPIO pins per chip.

[Top](#)

◀ See Also

Reference

[IO.Devices.PCA8574 Namespace](#)

Device Constructor

Constructor for a PCA8574 (or similar) GPIO Expander.

Namespace: [IO.Devices.PCA8574](#)

Assembly: libremoteio (in libremoteio.dll) Version: 2.2020.136.3

► Syntax

C# VB F#

[Copy](#)

```
public Device(  
    Bus bus,  
    int addr,  
    byte states = 255  
)
```

Parameters

bus

Type: [IO.Interfaces.I2CBus](#)

I²C bus controller.

addr

Type: [SystemInt32](#)

I²C slave address.

states (Optional)

Type: [SystemByte](#)

Initial output states.

► See Also

Reference

Device Class
IO.Devices.PCA8574 Namespace

Device Properties

The [Device](#) type exposes the following members.

▪ Properties

	Name	Description
	Latch	This read-only property returns the last value written to the output latch.

[Top](#)

▪ See Also

[Reference](#)

[Device Class](#)

[IO.Devices.PCA8574 Namespace](#)

DeviceLatch Property

This read-only property returns the last value written to the output latch.

Namespace: [IO.Devices.PCA8574](#)

Assembly: libremoteio (in libremoteio.dll) Version: 2.2020.136.3

↳ Syntax

[C#](#) [VB](#) [F#](#)

[Copy](#)

```
public byte Latch { get; }
```

Property Value

Type: [Byte](#)

↳ See Also

Reference

[Device Class](#)

[IO.Devices.PCA8574 Namespace](#)

Device Methods

The [Device](#) type exposes the following members.

▪ Methods

	Name	Description
	Read	Return actual state of the GPIO pins.
	Write	Write all GPIO pins.

[Top](#)

▪ See Also

[Reference](#)

[Device Class](#)

[IO.Devices.PCA8574 Namespace](#)

DeviceRead Method

Return actual state of the GPIO pins.

Namespace: [IO.Devices.PCA8574](#)

Assembly: libremoteio (in libremoteio.dll) Version: 2.2020.136.3

▪ Syntax

C# VB F#

[Copy](#)

```
public byte Read()
```

Return Value

Type: [Byte](#)

Pin states (MSB = GPIO7).

▪ See Also

Reference

[Device Class](#)

[IO.Devices.PCA8574 Namespace](#)

DeviceWrite Method

Write all GPIO pins.

Namespace: [IO.Devices.PCA8574](#)

Assembly: libremoteio (in libremoteio.dll) Version: 2.2020.136.3

◀ Syntax

C# VB F#

[Copy](#)

```
public void Write(  
    byte data  
)
```

Parameters

data

Type: [SystemByte](#)

Data to write to pins (MSB = GPIO7).

◀ See Also

Reference

[Device Class](#)

[IO.Devices.PCA8574 Namespace](#)

Device Fields

The [Device](#) type exposes the following members.

↳ Fields

Name	Description
  MAX_PINS	The number of available GPIO pins per chip.

[Top](#)

↳ See Also

[Reference](#)

[Device Class](#)

[IO.Devices.PCA8574 Namespace](#)

DeviceMAX_PINS Field

The number of available GPIO pins per chip.

Namespace: [IO.Devices.PCA8574](#)

Assembly: libremoteio (in libremoteio.dll) Version: 2.2020.136.3

« Syntax

C# VB F#

[Copy](#)

```
public const int MAX_PINS = 8
```

Field Value

Type: [Int32](#)

« See Also

Reference

[Device Class](#)

[IO.Devices.PCA8574 Namespace](#)

IO.Devices.PCA8574.GPIO Namespace

PCA8574 (and similar) I²C GPIO Expander GPIO Pin Services

↳ Classes

	Class	Description
	Pin	Encapsulates PCA8574 (and similar) I ² C GPIO Expander pins.

Pin Class

Encapsulates PCA8574 (and similar) I²C GPIO Expander pins.

▪ Inheritance Hierarchy

[SystemObject](#) [IO.Devices.PCA8574.GPIOPin](#)

Namespace: [IO.Devices.PCA8574.GPIO](#)

Assembly: libremoteio (in libremoteio.dll) Version: 2.2020.136.3

▪ Syntax

[C#](#) [VB](#) [F#](#)

[Copy](#)

```
public class Pin : Pin
```

The [Pin](#) type exposes the following members.

▪ Constructors

	Name	Description
	Pin	Constructor for a single GPIO pin.

[Top](#)

▪ Properties

	Name	Description
	state	Read/Write GPIO state property.

[Top](#)

« See Also

Reference

[IO.Devices.PCA8574.GPIO Namespace](#)

Pin Constructor

Constructor for a single GPIO pin.

Namespace: [IO.Devices.PCA8574.GPIO](#)

Assembly: libremoteio (in libremoteio.dll) Version: 2.2020.136.3

◀ Syntax

C# VB F#

[Copy](#)

```
public Pin(  
    Device dev,  
    int num,  
    Direction dir,  
    bool state = false  
)
```

Parameters

dev

Type: [IO.Devices.PCA8574Device](#)
PCA8574 (or similar) device.

num

Type: [SystemInt32](#)
GPIO pin number.

dir

Type: [IO.Interfaces.GPIODirection](#)
Data direction.

state (Optional)

Type: [SystemBoolean](#)
Initial GPIO output state.

See Also

Reference

[Pin Class](#)

[IO.Devices.PCA8574.GPIO Namespace](#)

Pin Properties

The [Pin](#) type exposes the following members.

Properties

	Name	Description
	state	Read/Write GPIO state property.

[Top](#)

See Also

Reference

[Pin Class](#)

[IO.Devices.PCA8574.GPIO Namespace](#)

Pinstate Property

Read/Write GPIO state property.

Namespace: [IO.Devices.PCA8574.GPIO](#)

Assembly: libremoteio (in libremoteio.dll) Version: 2.2020.136.3

▪ Syntax

C# VB F#

[Copy](#)

```
public bool state { get; set; }
```

Property Value

Type: [Boolean](#)

Implements

[Pinstate](#)

▪ See Also

[Reference](#)

[Pin Class](#)

[IO.Devices.PCA8574.GPIO Namespace](#)

IO.Devices.PCA9534 Namespace

PCA9534 (and similar) I²C GPIO Expander Device Services

↳ Classes

Class	Description
 Device	Encapsulates PCA9534 (and similar) I ² C GPIO Expanders.

Device Class

Encapsulates PCA9534 (and similar) I²C GPIO Expanders.

▪ Inheritance Hierarchy

[SystemObject](#) [IO.Devices.PCA9534Device](#)

Namespace: [IO.Devices.PCA9534](#)

Assembly: libremoteio (in libremoteio.dll) Version: 2.2020.136.3

▪ Syntax

[C#](#) [VB](#) [F#](#)

[Copy](#)

```
public class Device
```

The [Device](#) type exposes the following members.

▪ Constructors

	Name	Description
	Device	Constructor for a PCA9534 (or similar) GPIO Expander.

[Top](#)

▪ Properties

	Name	Description
	Config	This read-only property returns the last

value written to the configuration register.



Latch This read-only property returns the last value written to the output port register.

[Top](#)

Methods

	Name	Description
≡	Read	Return actual state of the GPIO pins.
≡	Read(Byte)	Read from the specified PCA9534 device register.
≡	Write(Byte)	Write all GPIO pins.
≡	Write(Byte, Byte)	Write to the specified PCA9534 device register.

[Top](#)

Fields

	Name	Description
• s	AllInputs	Configure all pins as inputs.
• s	AllNormal	Configure all inputs as normal polarity.
• s	AllOff	Turn all outputs off.
• s	AllOutputs	Configure all pins as outputs.

• S	ConfigurationReg	Configuration Register address.
• S	InputPolarityReg	Input Port Polarity Register address.
• S	InputPortReg	Input Port Register address.
• S	MAX_PINS	The number of available GPIO pins per chip.
• S	OutputPortReg	Output Port Register address.

[Top](#)

◀ See Also

Reference

[IO.Devices.PCA9534 Namespace](#)

Device Constructor

Constructor for a PCA9534 (or similar) GPIO Expander.

Namespace: [IO.Devices.PCA9534](#)

Assembly: libremoteio (in libremoteio.dll) Version: 2.2020.136.3

◀ Syntax

C# VB F#

[Copy](#)

```
public Device(  
    Bus bus,  
    int addr,  
    byte config = 255,  
    byte states = 0  
)
```

Parameters

bus

Type: [IO.Interfaces.I2CBus](#)

I²C bus controller.

addr

Type: [SystemInt32](#)

I²C slave address.

config (Optional)

Type: [SystemByte](#)

GPIO pin configuration.

states (Optional)

Type: [SystemByte](#)

Initial output states.

See Also

[Reference](#)

[Device Class](#)

[IO.Devices.PCA9534 Namespace](#)

Device Properties

The [Device](#) type exposes the following members.

Properties

	Name	Description
	Config	This read-only property returns the last value written to the configuration register.
	Latch	This read-only property returns the last value written to the output port register.

[Top](#)

See Also

Reference

[Device Class](#)

[IO.Devices.PCA9534 Namespace](#)

DeviceConfig Property

This read-only property returns the last value written to the configuration register.

Namespace: [IO.Devices.PCA9534](#)

Assembly: libremoteio (in libremoteio.dll) Version: 2.2020.136.3

◀ Syntax

C# VB F#

[Copy](#)

```
public byte Config { get; }
```

Property Value

Type: [Byte](#)

◀ See Also

Reference

[Device Class](#)

[IO.Devices.PCA9534 Namespace](#)

DeviceLatch Property

This read-only property returns the last value written to the output port register.

Namespace: [IO.Devices.PCA9534](#)

Assembly: libremoteio (in libremoteio.dll) Version: 2.2020.136.3

↳ Syntax

C# VB F#

[Copy](#)

```
public byte Latch { get; }
```

Property Value

Type: [Byte](#)

↳ See Also

Reference

[Device Class](#)

[IO.Devices.PCA9534 Namespace](#)

Device Methods

▪ Methods

	Name	Description
≡	Read	Return actual state of the GPIO pins.
≡	Read(Byte)	Read from the specified PCA9534 device register.
≡	Write(Byte)	Write all GPIO pins.
≡	Write(Byte, Byte)	Write to the specified PCA9534 device register.

[Top](#)

▪ See Also

Reference

[Device Class](#)

[IO.Devices.PCA9534 Namespace](#)

DeviceRead Method

▪ Overload List

	Name	Description
	Read	Return actual state of the GPIO pins.
	Read(Byte)	Read from the specified PCA9534 device register.

[Top](#)

▪ See Also

[Reference](#)

[Device Class](#)

[IO.Devices.PCA9534 Namespace](#)

DeviceRead Method

Return actual state of the GPIO pins.

Namespace: [IO.Devices.PCA9534](#)

Assembly: libremoteio (in libremoteio.dll) Version: 2.2020.136.3

▪ Syntax

C# VB F#

[Copy](#)

```
public byte Read()
```

Return Value

Type: [Byte](#)

Pin states (MSB = GPIO7).

▪ See Also

Reference

[Device Class](#)

[Read Overload](#)

[IO.Devices.PCA9534 Namespace](#)

DeviceRead Method (Byte)

Read from the specified PCA9534 device register.

Namespace: [IO.Devices.PCA9534](#)

Assembly: libremoteio (in libremoteio.dll) Version: 2.2020.136.3

◀ Syntax

C# VB F#

[Copy](#)

```
public byte Read(  
    byte addr  
)
```

Parameters

addr

Type: [SystemByte](#)

Register address.

Return Value

Type: [Byte](#)

Register contents.

◀ See Also

Reference

[Device Class](#)

[Read Overload](#)

[IO.Devices.PCA9534 Namespace](#)

DeviceWrite Method

↳ Overload List

	Name	Description
≡	Write(Byte)	Write all GPIO pins.
≡	Write(Byte, Byte)	Write to the specified PCA9534 device register.

[Top](#)

↳ See Also

[Reference](#)

[Device Class](#)

[IO.Devices.PCA9534 Namespace](#)

DeviceWrite Method (Byte)

Write all GPIO pins.

Namespace: [IO.Devices.PCA9534](#)

Assembly: libremoteio (in libremoteio.dll) Version: 2.2020.136.3

◀ Syntax

C# VB F#

[Copy](#)

```
public void Write(  
    byte data  
)
```

Parameters

data

Type: [SystemByte](#)

Data to write to pins (MSB = GPIO7).

◀ See Also

Reference

[Device Class](#)

[Write Overload](#)

[IO.Devices.PCA9534 Namespace](#)

DeviceWrite Method (Byte, Byte)

Write to the specified PCA9534 device register.

Namespace: [IO.Devices.PCA9534](#)

Assembly: libremoteio (in libremoteio.dll) Version: 2.2020.136.3

◀ Syntax

C# VB F#

[Copy](#)

```
public void Write(  
    byte addr,  
    byte data  
)
```

Parameters

addr

Type: [SystemByte](#)

Register address.

data

Type: [SystemByte](#)

Data to written.

◀ See Also

[Reference](#)

[Device Class](#)

[Write Overload](#)

[IO.Devices.PCA9534 Namespace](#)

Device Fields

The [Device](#) type exposes the following members.

Fields

	Name	Description
• s	AllInputs	Configure all pins as inputs.
• s	AllNormal	Configure all inputs as normal polarity.
• s	AllOff	Turn all outputs off.
• s	AllOutputs	Configure all pins as outputs.
• s	ConfigurationReg	Configuration Register address.
• s	InputPolarityReg	Input Port Polarity Register address.
• s	InputPortReg	Input Port Register address.
• s	MAX_PINS	The number of available GPIO pins per chip.
• s	OutputPortReg	Output Port Register address.

[Top](#)

See Also

Reference

Device Class

IO.Devices.PCA9534 Namespace

DeviceAllInputs Field

Configure all pins as inputs.

Namespace: [IO.Devices.PCA9534](#)

Assembly: libremoteio (in libremoteio.dll) Version: 2.2020.136.3

↳ Syntax

C# VB F#

[Copy](#)

```
public const byte AllInputs = 255
```

Field Value

Type: [Byte](#)

↳ See Also

Reference

[Device Class](#)

[IO.Devices.PCA9534 Namespace](#)

DeviceAllNormal Field

Configure all inputs as normal polarity.

Namespace: [IO.Devices.PCA9534](#)

Assembly: libremoteio (in libremoteio.dll) Version: 2.2020.136.3

↳ Syntax

C# VB F#

[Copy](#)

```
public const byte AllNormal = 0
```

Field Value

Type: [Byte](#)

↳ See Also

Reference

[Device Class](#)

[IO.Devices.PCA9534 Namespace](#)

DeviceAllOff Field

Turn all outputs off.

Namespace: [IO.Devices.PCA9534](#)

Assembly: libremoteio (in libremoteio.dll) Version: 2.2020.136.3

◀ Syntax

C# VB F#

[Copy](#)

```
public const byte AllOff = 0
```

Field Value

Type: [Byte](#)

◀ See Also

Reference

[Device Class](#)

[IO.Devices.PCA9534 Namespace](#)

DeviceAllOutputs Field

Configure all pins as outputs.

Namespace: [IO.Devices.PCA9534](#)

Assembly: libremoteio (in libremoteio.dll) Version: 2.2020.136.3

↳ Syntax

C# VB F#

[Copy](#)

```
public const byte AllOutputs = 0
```

Field Value

Type: [Byte](#)

↳ See Also

Reference

[Device Class](#)

[IO.Devices.PCA9534 Namespace](#)

DeviceConfigurationReg Field

Configuration Register address.

Namespace: [IO.Devices.PCA9534](#)

Assembly: libremoteio (in libremoteio.dll) Version: 2.2020.136.3

◀ Syntax

C# VB F#

[Copy](#)

```
public const byte ConfigurationReg = 3
```

Field Value

Type: [Byte](#)

◀ See Also

Reference

[Device Class](#)

[IO.Devices.PCA9534 Namespace](#)

DeviceInputPolarityReg Field

Input Port Polarity Register address.

Namespace: [IO.Devices.PCA9534](#)

Assembly: libremoteio (in libremoteio.dll) Version: 2.2020.136.3

◀ Syntax

C# VB F#

[Copy](#)

```
public const byte InputPolarityReg = 2
```

Field Value

Type: [Byte](#)

◀ See Also

Reference

[Device Class](#)

[IO.Devices.PCA9534 Namespace](#)

DeviceInputPortReg Field

Input Port Register address.

Namespace: [IO.Devices.PCA9534](#)

Assembly: libremoteio (in libremoteio.dll) Version: 2.2020.136.3

◀ Syntax

C# VB F#

[Copy](#)

```
public const byte InputPortReg = 0
```

Field Value

Type: [Byte](#)

◀ See Also

Reference

[Device Class](#)

[IO.Devices.PCA9534 Namespace](#)

DeviceMAX_PINS Field

The number of available GPIO pins per chip.

Namespace: [IO.Devices.PCA9534](#)

Assembly: libremoteio (in libremoteio.dll) Version: 2.2020.136.3

◀ Syntax

C# VB F#

[Copy](#)

```
public const int MAX_PINS = 8
```

Field Value

Type: [Int32](#)

◀ See Also

Reference

[Device Class](#)

[IO.Devices.PCA9534 Namespace](#)

DeviceOutputPortReg Field

Output Port Register address.

Namespace: [IO.Devices.PCA9534](#)

Assembly: libremoteio (in libremoteio.dll) Version: 2.2020.136.3

◀ Syntax

C# VB F#

[Copy](#)

```
public const byte OutputPortReg = 1
```

Field Value

Type: [Byte](#)

◀ See Also

Reference

[Device Class](#)

[IO.Devices.PCA9534 Namespace](#)

IO.Devices.PCA9534.GPIO Namespace

PCA9534 (and similar) I²C GPIO Expander GPIO Pin Services

↳ Classes

	Class	Description
	Pin	Encapsulates PCA9534 (and similar) I ² C GPIO Expander pins.

Pin Class

Encapsulates PCA9534 (and similar) I²C GPIO Expander pins.

▪ Inheritance Hierarchy

[SystemObject](#) [IO.Devices.PCA9534.GPIOPin](#)

Namespace: [IO.Devices.PCA9534.GPIO](#)

Assembly: libremoteio (in libremoteio.dll) Version: 2.2020.136.3

▪ Syntax

[C#](#) [VB](#) [F#](#)

[Copy](#)

```
public class Pin : Pin
```

The [Pin](#) type exposes the following members.

▪ Constructors

	Name	Description
	Pin	Constructor for a single GPIO pin.

[Top](#)

▪ Properties

	Name	Description
	state	Read/Write GPIO state property.

[Top](#)

↳ See Also

Reference

[IO.Devices.PCA9534.GPIO Namespace](#)

Pin Constructor

Constructor for a single GPIO pin.

Namespace: [IO.Devices.PCA9534.GPIO](#)

Assembly: libremoteio (in libremoteio.dll) Version: 2.2020.136.3

◀ Syntax

C# VB F#

[Copy](#)

```
public Pin(  
    Device dev,  
    int num,  
    Direction dir,  
    bool state = false  
)
```

Parameters

dev

Type: [IO.Devices.PCA9534Device](#)
PCA9534 (or similar) device.

num

Type: [SystemInt32](#)
GPIO pin number.

dir

Type: [IO.Interfaces.GPIODirection](#)
Data direction.

state (Optional)

Type: [SystemBoolean](#)
Initial GPIO output state.

See Also

Reference

[Pin Class](#)

[IO.Devices.PCA9534.GPIO Namespace](#)

Pin Properties

The [Pin](#) type exposes the following members.

Properties

	Name	Description
	state	Read/Write GPIO state property.

[Top](#)

See Also

Reference

[Pin Class](#)

[IO.Devices.PCA9534.GPIO Namespace](#)

Pinstate Property

Read/Write GPIO state property.

Namespace: [IO.Devices.PCA9534.GPIO](#)

Assembly: libremoteio (in libremoteio.dll) Version: 2.2020.136.3

▪ Syntax

C# VB F#

[Copy](#)

```
public bool state { get; set; }
```

Property Value

Type: [Boolean](#)

Implements

[Pinstate](#)

▪ See Also

[Reference](#)

[Pin Class](#)

[IO.Devices.PCA9534.GPIO Namespace](#)

IO.Devices.PCA9685 Namespace

PCA9685 I²C PWM Controller Device Services

» Classes

Class	Description
 Device	Encapsulates the PCA9685 I ² C PWM Controller.

Device Class

Encapsulates the PCA9685 I²C PWM Controller.

▪ Inheritance Hierarchy

[SystemObject](#) [IO.Devices.PCA9685Device](#)

Namespace: [IO.Devices.PCA9685](#)

Assembly: libremoteio (in libremoteio.dll) Version: 2.2020.136.3

▪ Syntax

[C#](#) [VB](#) [F#](#)

[Copy](#)

```
public class Device
```

The [Device](#) type exposes the following members.

▪ Constructors

	Name	Description
	Device	Constructor for a single PCA9685 device.

[Top](#)

▪ Properties

	Name	Description
	Frequency	Read-only property returning the configured PWM pulse frequency.

[Top](#)

◀ Methods

	Name	Description
≡	ReadChannel	Read PCA9685 output channel data.
≡	WriteChannel	Write PCA9685 output channel data.

[Top](#)

◀ Fields

	Name	Description
♦ s	INTERNAL_CLOCK	Select internal 25 MHz clock oscillator.
♦ s	MAX_CHANNEL	Maximum PCA9685 output channel number.
♦ s	MAX_CLOCK	Maximum clock frequency.
♦ s	MIN_CHANNEL	Minimum PCA9685 output channel number.
♦ s	MIN_CLOCK	Minimum clock frequency.

[Top](#)

◀ See Also

Reference

[IO.Devices.PCA9685 Namespace](#)

Device Constructor

Constructor for a single PCA9685 device.

Namespace: [IO.Devices.PCA9685](#)

Assembly: libremoteio (in libremoteio.dll) Version: 2.2020.136.3

◀ Syntax

C# VB F#

[Copy](#)

```
public Device(  
    Bus bus,  
    int addr,  
    int freq = 50,  
    int clock = 0  
)
```

Parameters

bus

Type: [IO.Interfaces.I2CBus](#)

I²C bus controller object.

addr

Type: [SystemInt32](#)

I²C slave address.

freq (Optional)

Type: [SystemInt32](#)

PWM pulse frequency. Default is 50 Hz.

clock (Optional)

Type: [SystemInt32](#)

PCA9685 clock source. Use [INTERNAL_CLOCK](#) to select the

internal 25 MHz clock generator.

◀ See Also

Reference

[Device Class](#)

[IO.Devices.PCA9685 Namespace](#)

Device Properties

The [Device](#) type exposes the following members.

▪ Properties

	Name	Description
	Frequency	Read-only property returning the configured PWM pulse frequency.

[Top](#)

▪ See Also

[Reference](#)

[Device Class](#)

[IO.Devices.PCA9685 Namespace](#)

DeviceFrequency Property

Read-only property returning the configured PWM pulse frequency.

Namespace: [IO.Devices.PCA9685](#)

Assembly: libremoteio (in libremoteio.dll) Version: 2.2020.136.3

« Syntax

C# VB F#

[Copy](#)

```
public int Frequency { get; }
```

Property Value

Type: [Int32](#)

« See Also

Reference

[Device Class](#)

[IO.Devices.PCA9685 Namespace](#)

Device Methods

The [Device](#) type exposes the following members.

▪ Methods

	Name	Description
	ReadChannel	Read PCA9685 output channel data.
	WriteChannel	Write PCA9685 output channel data.

[Top](#)

▪ See Also

[Reference](#)

[Device Class](#)

[IO.Devices.PCA9685 Namespace](#)

DeviceReadChannel Method

Read PCA9685 output channel data.

Namespace: [IO.Devices.PCA9685](#)

Assembly: libremoteio (in libremoteio.dll) Version: 2.2020.136.3

◀ Syntax

C# VB F#

[Copy](#)

```
public void ReadChannel(  
    byte channel,  
    ref byte[] data  
)
```

Parameters

channel

Type: [SystemByte](#)

Output channel number.

data

Type: [SystemByte](#)

Output channel data (4 bytes).

◀ See Also

[Reference](#)

[Device Class](#)

[IO.Devices.PCA9685 Namespace](#)

DeviceWriteChannel Method

Write PCA9685 output channel data.

Namespace: [IO.Devices.PCA9685](#)

Assembly: libremoteio (in libremoteio.dll) Version: 2.2020.136.3

◀ Syntax

C# VB F#

[Copy](#)

```
public void WriteChannel(  
    byte channel,  
    byte[] data  
)
```

Parameters

channel

Type: [SystemByte](#)

Output channel number.

data

Type: [SystemByte](#)

Output channel data.

◀ See Also

[Reference](#)

[Device Class](#)

[IO.Devices.PCA9685 Namespace](#)

Device Fields

The [Device](#) type exposes the following members.

Fields

	Name	Description
• S	INTERNAL_CLOCK	Select internal 25 MHz clock oscillator.
• S	MAX_CHANNEL	Maximum PCA9685 output channel number.
• S	MAX_CLOCK	Maximum clock frequency.
• S	MIN_CHANNEL	Minimum PCA9685 output channel number.
• S	MIN_CLOCK	Minimum clock frequency.

[Top](#)

See Also

Reference

[Device Class](#)

[IO.Devices.PCA9685 Namespace](#)

DeviceINTERNAL_CLOCK Field

Select internal 25 MHz clock oscillator.

Namespace: [IO.Devices.PCA9685](#)

Assembly: libremoteio (in libremoteio.dll) Version: 2.2020.136.3

◀ Syntax

C# VB F#

[Copy](#)

```
public const int INTERNAL_CLOCK = 0
```

Field Value

Type: [Int32](#)

◀ See Also

Reference

[Device Class](#)

[IO.Devices.PCA9685 Namespace](#)

DeviceMAX_CHANNEL Field

Maximum PCA9685 output channel number.

Namespace: [IO.Devices.PCA9685](#)

Assembly: libremoteio (in libremoteio.dll) Version: 2.2020.136.3

« Syntax

C# VB F#

[Copy](#)

```
public const int MAX_CHANNEL = 15
```

Field Value

Type: [Int32](#)

« See Also

Reference

[Device Class](#)

[IO.Devices.PCA9685 Namespace](#)

DeviceMAX_CLOCK Field

Maximum clock frequency.

Namespace: [IO.Devices.PCA9685](#)

Assembly: libremoteio (in libremoteio.dll) Version: 2.2020.136.3

« Syntax

C# VB F#

[Copy](#)

```
public const int MAX_CLOCK = 50000000
```

Field Value

Type: [Int32](#)

« See Also

Reference

[Device Class](#)

[IO.Devices.PCA9685 Namespace](#)

DeviceMIN_CHANNEL Field

Minimum PCA9685 output channel number.

Namespace: [IO.Devices.PCA9685](#)

Assembly: libremoteio (in libremoteio.dll) Version: 2.2020.136.3

◀ Syntax

C# VB F#

[Copy](#)

```
public const int MIN_CHANNEL = 0
```

Field Value

Type: [Int32](#)

◀ See Also

Reference

[Device Class](#)

[IO.Devices.PCA9685 Namespace](#)

DeviceMIN_CLOCK Field

Minimum clock frequency.

Namespace: [IO.Devices.PCA9685](#)

Assembly: libremoteio (in libremoteio.dll) Version: 2.2020.136.3

◀ Syntax

C# VB F#

[Copy](#)

```
public const int MIN_CLOCK = 1
```

Field Value

Type: [Int32](#)

◀ See Also

Reference

[Device Class](#)

[IO.Devices.PCA9685 Namespace](#)

IO.Devices.PCA9685.GPIO Namespace

PCA9685 I²C PWM Controller GPIO Pin Services

↳ Classes

Class	Description
 Pin	Encapsulates PCA9685 GPIO outputs.

Pin Class

Encapsulates PCA9685 GPIO outputs.

▪ Inheritance Hierarchy

[SystemObject](#) [IO.Devices.PCA9685.GPIOPin](#)

Namespace: [IO.Devices.PCA9685.GPIO](#)

Assembly: libremoteio (in libremoteio.dll) Version: 2.2020.136.3

▪ Syntax

C# VB F#

[Copy](#)

```
public class Pin : Pin
```

The [Pin](#) type exposes the following members.

▪ Constructors

	Name	Description
	Pin	Constructor for a single GPIO output pin.

[Top](#)

▪ Properties

	Name	Description
	state	Read/Write GPIO output state property.

[Top](#)

« See Also

Reference

[IO.Devices.PCA9685.GPIO Namespace](#)

Pin Constructor

Constructor for a single GPIO output pin.

Namespace: [IO.Devices.PCA9685.GPIO](#)

Assembly: libremoteio (in libremoteio.dll) Version: 2.2020.136.3

◀ Syntax

C# VB F#

[Copy](#)

```
public Pin(  
    Device dev,  
    int channel,  
    bool state = false  
)
```

Parameters

dev

Type: [IO.Devices.PCA9685Device](#)
PCA9685 device object.

channel

Type: [SystemInt32](#)
Output channel number.

state (Optional)

Type: [SystemBoolean](#)
Initial GPIO output state.

◀ See Also

Reference

[Pin Class](#)

IO.Devices.PCA9685.GPIO Namespace

Pin Properties

The [Pin](#) type exposes the following members.

Properties

	Name	Description
	state	Read/Write GPIO output state property.

[Top](#)

See Also

Reference

[Pin Class](#)

[IO.Devices.PCA9685.GPIO Namespace](#)

Pinstate Property

Read/Write GPIO output state property.

Namespace: [IO.Devices.PCA9685.GPIO](#)

Assembly: libremoteio (in libremoteio.dll) Version: 2.2020.136.3

▪ Syntax

C# VB F#

[Copy](#)

```
public bool state { get; set; }
```

Property Value

Type: [Boolean](#)

Implements

[Pinstate](#)

▪ See Also

[Reference](#)

[Pin Class](#)

[IO.Devices.PCA9685.GPIO Namespace](#)

IO.Devices.PCA9685.PWM Namespace

PCA9685 I²C PWM Controller PWM Output Services

↳ Classes

Class	Description
 Output	Encapsulates PCA9685 PWM outputs.

Output Class

Encapsulates PCA9685 PWM outputs.

▪ Inheritance Hierarchy

[SystemObject](#) [IO.Devices.PCA9685.PWMOutput](#)

Namespace: [IO.Devices.PCA9685.PWM](#)

Assembly: libremoteio (in libremoteio.dll) Version: 2.2020.136.3

▪ Syntax

C# VB F#

[Copy](#)

```
public class Output : Output
```

The [Output](#) type exposes the following members.

▪ Constructors

	Name	Description
	Output	Constructor for a single PWM output.

[Top](#)

▪ Properties

	Name	Description
	dutycycle	Write-only property for setting the PWM output duty cycle. Allowed values

are 0.0 to 100.0 percent.

[Top](#)

◀ See Also

Reference

[IO.Devices.PCA9685.PWM Namespace](#)

Output Constructor

Constructor for a single PWM output.

Namespace: [IO.Devices.PCA9685.PWM](#)

Assembly: libremoteio (in libremoteio.dll) Version: 2.2020.136.3

◀ Syntax

C# VB F#

[Copy](#)

```
public Output(  
    Device dev,  
    int channel,  
    double dutycycle = 0  
)
```

Parameters

dev

Type: [IO.Devices.PCA9685Device](#)
PCA9685 device object.

channel

Type: [SystemInt32](#)
Output channel number.

dutycycle (Optional)

Type: [SystemDouble](#)
Initial PWM output duty cycle.

◀ See Also

[Reference](#)

[Output Class](#)

IO.Devices.PCA9685.PWM Namespace

Output Properties

The [Output](#) type exposes the following members.

▪ Properties

	Name	Description
	dutycycle	Write-only property for setting the PWM output duty cycle. Allowed values are 0.0 to 100.0 percent.

[Top](#)

▪ See Also

[Reference](#)

[Output Class](#)

[IO.Devices.PCA9685.PWM Namespace](#)

Outputdutycycle Property

Write-only property for setting the PWM output duty cycle. Allowed values are 0.0 to 100.0 percent.

Namespace: [IO.Devices.PCA9685.PWM](#)

Assembly: libremoteio (in libremoteio.dll) Version: 2.2020.136.3

↳ Syntax

C# VB F#

[Copy](#)

```
public double dutycycle { set; }
```

Property Value

Type: [Double](#)

Implements

[Outputdutycycle](#)

↳ See Also

Reference

[Output Class](#)

[IO.Devices.PCA9685.PWM Namespace](#)

IO.Devices.PCA9685.Servo Namespace

PCA9685 I²C PWM Controller Servo Output Services

↳ Classes

Class	Description
 Output	Encapsulates PCA9685 servo outputs.

Output Class

Encapsulates PCA9685 servo outputs.

▪ Inheritance Hierarchy

[SystemObject](#) [IO.Devices.PCA9685.ServoOutput](#)

Namespace: [IO.Devices.PCA9685.Servo](#)

Assembly: libremoteio (in libremoteio.dll) Version: 2.2020.136.3

▪ Syntax

C# VB F#

[Copy](#)

```
public class Output : Output
```

The [Output](#) type exposes the following members.

▪ Constructors

	Name	Description
	Output	Constructor for a single servo output.

[Top](#)

▪ Properties

	Name	Description
	position	Write-only property for setting the normalized servo position. Allowed

values are -0.0 to +1.0.

[Top](#)

◀ See Also

Reference

[IO.Devices.PCA9685.Servo Namespace](#)

Output Constructor

Constructor for a single servo output.

Namespace: [IO.Devices.PCA9685.Servo](#)

Assembly: libremoteio (in libremoteio.dll) Version: 2.2020.136.3

◀ Syntax

C# VB F#

[Copy](#)

```
public Output(  
    Device dev,  
    int channel,  
    double position = 0  
)
```

Parameters

dev

Type: [IO.Devices.PCA9685Device](#)
PCA9685 device object.

channel

Type: [SystemInt32](#)
Output channel number.

position (Optional)

Type: [SystemDouble](#)
Initial servo position.

◀ See Also

[Reference](#)

[Output Class](#)

IO.Devices.PCA9685.Servo Namespace

Output Properties

The [Output](#) type exposes the following members.

▪ Properties

	Name	Description
	position	Write-only property for setting the normalized servo position. Allowed values are -0.0 to +1.0.

[Top](#)

▪ See Also

[Reference](#)

[Output Class](#)

[IO.Devices.PCA9685.Servo Namespace](#)

Outputposition Property

Write-only property for setting the normalized servo position. Allowed values are -0.0 to +1.0.

Namespace: [IO.Devices.PCA9685.Servo](#)

Assembly: libremoteio (in libremoteio.dll) Version: 2.2020.136.3

↳ Syntax

[C#](#) [VB](#) [F#](#)

[Copy](#)

```
public double position { set; }
```

Property Value

Type: [Double](#)

Implements

[Outputposition](#)

↳ See Also

[Reference](#)

[Output Class](#)

[IO.Devices.PCA9685.Servo Namespace](#)

IO.Devices.Pmod.HYGRO Namespace

Digilent Pmod HYGRO Temperature and Humdity Sensor (HDC1080) Services.

◀ Classes

	Class	Description
	Device	Encapsulate the Digilent Pmod HYGRO Temperature and Humidity Sensor (HDC1080).

Device Class

Encapsulate the Digilent Pmod HYGRO Temperature and Humidity Sensor (HDC1080).

▪ Inheritance Hierarchy

```
SystemObject IO.Devices.HDC1080Device
    IO.Devices.Pmod.HYGRODevice
```

Namespace: [IO.Devices.Pmod.HYGRO](#)

Assembly: libremoteio (in libremoteio.dll) Version: 2.2020.136.3

▪ Syntax

C# VB F#

[Copy](#)

```
public class Device : Device
```

The [Device](#) type exposes the following members.

▪ Constructors

Name	Description
 Device	Constructor for a Digilent Pmod HYGRO Temperature and Humidity Sensor (HDC1080).

[Top](#)

▪ Properties

	Name	Description
	Celsius	Read-only property returning the temperature in degrees Celsius. (Inherited from Device .)
	DeviceID	Read-only property returning the device ID. (Inherited from Device .)
	Fahrenheit	Read-only property returning the temperature in degrees Fahrenheit. (Inherited from Device .)
	Humidity	Read-only property returning the percentage relative humidity. (Inherited from Device .)
	Kelvins	Read-only property returning the temperature in Kelvins. (Inherited from Device .)
	ManufacturerID	Read-only property returning the manufacturer ID. (Inherited from Device .)

[Top](#)

◀ Methods

	Name	Description
	Read	Read from an HDC1080 device register. (Inherited from Device .)



Write

Write to an HDC1080 device register.
(Inherited from [Device](#).)

[Top](#)

See Also

Reference

[IO.Devices.Pmod.HYGRO Namespace](#)

Device Constructor

Constructor for a Digilent Pmod HYGRO Temperature and Humidity Sensor (HDC1080).

Namespace: [IO.Devices.Pmod.HYGRO](#)

Assembly: libremoteio (in libremoteio.dll) Version: 2.2020.136.3

◀ Syntax

C# VB F#

[Copy](#)

```
public Device(  
    Bus bus  
)
```

Parameters

bus

Type: [IO.Interfaces.I2CBus](#)

I² bus object.

◀ See Also

Reference

[Device Class](#)

[IO.Devices.Pmod.HYGRO Namespace](#)

Device Properties

The [Device](#) type exposes the following members.

Properties

	Name	Description
	Celsius	Read-only property returning the temperature in degrees Celsius. (Inherited from Device .)
	DeviceID	Read-only property returning the device ID. (Inherited from Device .)
	Fahrenheit	Read-only property returning the temperature in degrees Fahrenheit. (Inherited from Device .)
	Humidity	Read-only property returning the percentage relative humidity. (Inherited from Device .)
	Kelvins	Read-only property returning the temperature in Kelvins. (Inherited from Device .)
	ManufacturerID	Read-only property returning the manufacturer ID.

(Inherited from [Device](#).)

[Top](#)

◀ See Also

Reference

[Device Class](#)

[IO.Devices.Pmod.HYGRO Namespace](#)

Device Methods

The [Device](#) type exposes the following members.

◀ Methods

	Name	Description
≡	Read	Read from an HDC1080 device register. (Inherited from Device .)
≡	Write	Write to an HDC1080 device register. (Inherited from Device .)

[Top](#)

◀ See Also

Reference

[Device Class](#)

[IO.Devices.Pmod.HYGRO Namespace](#)

IO.Devices.SN74HC595 Namespace

SN74HC595 8-Bit Shift Register Device Services

► Classes

Class	Description
 Device	Encapsulates a chain of one or more SN74HC595 8-bit shift registers.

Device Class

Encapsulates a chain of one or more SN74HC595 8-bit shift registers.

▪ Inheritance Hierarchy

[SystemObject](#) [IO.Devices.SN74HC595Device](#)

Namespace: [IO.Devices.SN74HC595](#)

Assembly: libremoteio (in libremoteio.dll) Version: 2.2020.136.3

▪ Syntax

C# VB F#

[Copy](#)

```
public class Device
```

The [Device](#) type exposes the following members.

▪ Constructors

	Name	Description
	Device	Constructor for a chain of one or more SN74HC595 shift registers.

[Top](#)

▪ Properties

	Name	Description
	Length	Read-only property returning the

number of stages in the chain.



state Read/Write shift register chain state property.

[Top](#)

◀ Methods

	Name	Description
≡	ClrBit	Clear a single bit in the shift register chain.
≡	ReadBit	Read a single bit in the shift register chain.
≡	SetBit	Set a single bit in the shift register chain.

[Top](#)

◀ Fields

	Name	Description
♦ S	SPI_MaxFreq	SPI maximum clock frequency for the SNHC74HC595 shift register. (Most pessimistic datasheet limit at 2V.)
♦ S	SPI_Mode	SPI clock mode for the SNHC74HC595 shift register.

[Top](#)

◀ See Also

Reference

IO.Devices.SN74HC595 Namespace

Device Constructor

Constructor for a chain of one or more SN74HC595 shift registers.

Namespace: [IO.Devices.SN74HC595](#)

Assembly: libremoteio (in libremoteio.dll) Version: 2.2020.136.3

↳ Syntax

C# VB F#

[Copy](#)

```
public Device(  
    Device dev,  
    int stages = 1,  
    byte[] initialstate = null  
)
```

Parameters

dev

Type: [IO.Interfaces.SPIDevice](#)

SPI device object.

stages (Optional)

Type: [SystemInt32](#)

Number of stages in the chain.

initialstate (Optional)

Type: [SystemByte](#)

Initial shift register chain state.

↳ See Also

[Reference](#)

[Device Class](#)

IO.Devices.SN74HC595 Namespace

Device Properties

The [Device](#) type exposes the following members.

Properties

	Name	Description
	Length	Read-only property returning the number of stages in the chain.
	state	Read/Write shift register chain state property.

[Top](#)

See Also

Reference

[Device Class](#)

[IO.Devices.SN74HC595 Namespace](#)

DeviceLength Property

Read-only property returning the number of stages in the chain.

Namespace: [IO.Devices.SN74HC595](#)

Assembly: libremoteio (in libremoteio.dll) Version: 2.2020.136.3

↳ Syntax

C# VB F#

[Copy](#)

```
public int Length { get; }
```

Property Value

Type: [Int32](#)

↳ See Also

Reference

[Device Class](#)

[IO.Devices.SN74HC595 Namespace](#)

Devicestate Property

Read/Write shift register chain state property.

Namespace: [IO.Devices.SN74HC595](#)

Assembly: libremoteio (in libremoteio.dll) Version: 2.2020.136.3

↳ Syntax

C# VB F#

[Copy](#)

```
public byte[] state { get; set; }
```

Property Value

Type: [Byte](#)

↳ See Also

Reference

[Device Class](#)

[IO.Devices.SN74HC595 Namespace](#)

Device Methods

The [Device](#) type exposes the following members.

▪ Methods

	Name	Description
≡	ClrBit	Clear a single bit in the shift register chain.
≡	ReadBit	Read a single bit in the shift register chain.
≡	SetBit	Set a single bit in the shift register chain.

[Top](#)

▪ See Also

Reference

[Device Class](#)

[IO.Devices.SN74HC595 Namespace](#)

DeviceClrBit Method

Clear a single bit in the shift register chain.

Namespace: [IO.Devices.SN74HC595](#)

Assembly: libremoteio (in libremoteio.dll) Version: 2.2020.136.3

« Syntax

C# VB F#

[Copy](#)

```
public void ClrBit(  
    int index,  
    byte mask  
)
```

Parameters

index

Type: [SystemInt32](#)

Shift register stage number. Zero indicates the first register stage.

mask

Type: [SystemByte](#)

Shift register bit mask.

« See Also

[Reference](#)

[Device Class](#)

[IO.Devices.SN74HC595 Namespace](#)

DeviceReadBit Method

Read a single bit in the shift register chain.

Namespace: [IO.Devices.SN74HC595](#)

Assembly: libremoteio (in libremoteio.dll) Version: 2.2020.136.3

◀ Syntax

C# VB F#

[Copy](#)

```
public bool ReadBit(  
    int index,  
    byte mask  
)
```

Parameters

index

Type: [SystemInt32](#)

Shift register stage number. Zero indicates the first register stage.

mask

Type: [SystemByte](#)

Shift register bit mask.

Return Value

Type: [Boolean](#)

Boolean bit value.

◀ See Also

Reference

[Device Class](#)

IO.Devices.SN74HC595 Namespace

DeviceSetBit Method

Set a single bit in the shift register chain.

Namespace: [IO.Devices.SN74HC595](#)

Assembly: libremoteio (in libremoteio.dll) Version: 2.2020.136.3

« Syntax

C# VB F#

[Copy](#)

```
public void SetBit(  
    int index,  
    byte mask  
)
```

Parameters

index

Type: [SystemInt32](#)

Shift register stage number. Zero indicates the first register stage.

mask

Type: [SystemByte](#)

Shift register bit mask.

« See Also

[Reference](#)

[Device Class](#)

[IO.Devices.SN74HC595 Namespace](#)

Device Fields

The [Device](#) type exposes the following members.

Fields

	Name	Description
◆ 	SPI_MaxFreq	SPI maximum clock frequency for the SNHC74HC595 shift register. (Most pessimistic datasheet limit at 2V.)
◆ 	SPI_Mode	SPI clock mode for the SNHC74HC595 shift register.

[Top](#)

See Also

Reference

[Device Class](#)

[IO.Devices.SN74HC595 Namespace](#)

DeviceSPI_MaxFreq Field

SPI maximum clock frequency for the SNHC74HC595 shift register.
(Most pessimistic datasheet limit at 2V.)

Namespace: [IO.Devices.SN74HC595](#)

Assembly: libremoteio (in libremoteio.dll) Version: 2.2020.136.3

◀ Syntax

C# VB F#

[Copy](#)

```
public const int SPI_MaxFreq = 4000000
```

Field Value

Type: [Int32](#)

◀ See Also

Reference

[Device Class](#)

[IO.Devices.SN74HC595 Namespace](#)

DeviceSPI_Mode Field

SPI clock mode for the SNHC74HC595 shift register.

Namespace: [IO.Devices.SN74HC595](#)

Assembly: libremoteio (in libremoteio.dll) Version: 2.2020.136.3

« Syntax

C# VB F#

[Copy](#)

```
public const int SPI_Mode = 0
```

Field Value

Type: [Int32](#)

« See Also

Reference

[Device Class](#)

[IO.Devices.SN74HC595 Namespace](#)

IO.Devices.SN74HC595.GPIO Namespace

SN74HC595 8-Bit Shift Register GPIO Pin Services

◀ Classes

	Class	Description
	Pin	Encapsulates SN74HC595 GPIO outputs.

Pin Class

Encapsulates SN74HC595 GPIO outputs.

▪ Inheritance Hierarchy

[SystemObject](#) [IO.Devices.SN74HC595.GPIOPin](#)

Namespace: [IO.Devices.SN74HC595.GPIO](#)

Assembly: libremoteio (in libremoteio.dll) Version: 2.2020.136.3

▪ Syntax

C# VB F#

[Copy](#)

```
public class Pin : Pin
```

The [Pin](#) type exposes the following members.

▪ Constructors

	Name	Description
	Pin	Constructor for a single GPIO output pin.

[Top](#)

▪ Properties

	Name	Description
	state	Read/Write GPIO pin state property.

[Top](#)

↳ See Also

Reference

[IO.Devices.SN74HC595.GPIO Namespace](#)

Pin Constructor

Constructor for a single GPIO output pin.

Namespace: [IO.Devices.SN74HC595.GPIO](#)

Assembly: libremoteio (in libremoteio.dll) Version: 2.2020.136.3

◀ Syntax

C# VB F#

[Copy](#)

```
public Pin(  
    Device dev,  
    int pos,  
    bool state = false  
)
```

Parameters

dev

Type: [IO.Devices.SN74HC595Device](#)
SN74HC595 device object.

pos

Type: [SystemInt32](#)
Bit position, numbered left to right. Zero indicates the most significant bit of the first shift register stage.

state (Optional)

Type: [SystemBoolean](#)
Initial GPIO output state.

◀ See Also

Reference

Pin Class
IO.Devices.SN74HC595.GPIO Namespace

Pin Properties

The [Pin](#) type exposes the following members.

Properties

	Name	Description
	state	Read/Write GPIO pin state property.

[Top](#)

See Also

Reference

[Pin Class](#)

[IO.Devices.SN74HC595.GPIO Namespace](#)

Pinstate Property

Read/Write GPIO pin state property.

Namespace: [IO.Devices.SN74HC595.GPIO](#)

Assembly: libremoteio (in libremoteio.dll) Version: 2.2020.136.3

↳ Syntax

C# VB F#

[Copy](#)

```
public bool state { get; set; }
```

Property Value

Type: [Boolean](#)

Implements

[Pinstate](#)

↳ See Also

[Reference](#)

[Pin Class](#)

[IO.Devices.SN74HC595.GPIO Namespace](#)

IO.Devices.TH02 Namespace

TH02 I²C Temperature/Humidity Sensor Services

↳ Classes

	Class	Description
	Device	Encapsulates the TH02 temperature and humidity sensor.

Device Class

Encapsulates the TH02 temperature and humidity sensor.

▪ Inheritance Hierarchy

[SystemObject](#) [IO.Devices.TH02Device](#)
[IO.Devices.Grove.Temperature_HumidityDevice](#)

Namespace: [IO.Devices.TH02](#)

Assembly: libremoteio (in libremoteio.dll) Version: 2.2020.136.3

▪ Syntax

[C#](#) [VB](#) [F#](#)

[Copy](#)

```
public class Device : Sensor, Sensor
```

The [Device](#) type exposes the following members.

▪ Constructors

	Name	Description
≡	Device	Constructor for an TH02 temperature and humidity sensor object.

[Top](#)

▪ Properties

	Name	Description
--	------	-------------

	Celsius	Read-only property returning the temperature in degrees Celsius.
	DeviceID	Read-only property returning the device ID.
	Fahrenheit	Read-only property returning the temperature in degrees Fahrenheit.
	Humidity	Read-only property returning the percentage relative humidity.
	Kelvins	Read-only property returning the temperature in Kelvins.

[Top](#)

See Also

Reference

[IO.Devices.TH02 Namespace](#)

Device Constructor

Constructor for an TH02 temperature and humidity sensor object.

Namespace: [IO.Devices.TH02](#)

Assembly: libremoteio (in libremoteio.dll) Version: 2.2020.136.3

▀ Syntax

C# VB F#

[Copy](#)

```
public Device(  
    Bus bus  
)
```

Parameters

bus

Type: [IO.Interfaces.I2CBus](#)

I²C bus controller.

▀ See Also

Reference

[Device Class](#)

[IO.Devices.TH02 Namespace](#)

Device Properties

The [Device](#) type exposes the following members.

Properties

	Name	Description
	Celsius	Read-only property returning the temperature in degrees Celsius.
	DeviceID	Read-only property returning the device ID.
	Fahrenheit	Read-only property returning the temperature in degrees Fahrenheit.
	Humidity	Read-only property returning the percentage relative humidity.
	Kelvins	Read-only property returning the temperature in Kelvins.

[Top](#)

See Also

Reference

[Device Class](#)

[IO.Devices.TH02 Namespace](#)

DeviceCelsius Property

Read-only property returning the temperature in degrees Celsius.

Namespace: [IO.Devices.TH02](#)

Assembly: libremoteio (in libremoteio.dll) Version: 2.2020.136.3

▪ Syntax

C# VB F#

[Copy](#)

```
public double Celsius { get; }
```

Property Value

Type: [Double](#)

Implements

[SensorCelsius](#)

▪ See Also

[Reference](#)

[Device Class](#)

[IO.Devices.TH02 Namespace](#)

DeviceDeviceID Property

Read-only property returning the device ID.

Namespace: [IO.Devices.TH02](#)

Assembly: libremoteio (in libremoteio.dll) Version: 2.2020.136.3

◀ Syntax

C# VB F#

[Copy](#)

```
public byte DeviceID { get; }
```

Property Value

Type: [Byte](#)

◀ See Also

Reference

[Device Class](#)

[IO.Devices.TH02 Namespace](#)

DeviceFahrenheit Property

Read-only property returning the temperature in degrees Fahrenheit.

Namespace: [IO.Devices.TH02](#)

Assembly: libremoteio (in libremoteio.dll) Version: 2.2020.136.3

▪ Syntax

C# VB F#

[Copy](#)

```
public double Fahrenheit { get; }
```

Property Value

Type: [Double](#)

Implements

[SensorFahrenheit](#)

▪ See Also

[Reference](#)

[Device Class](#)

[IO.Devices.TH02 Namespace](#)

DeviceHumidity Property

Read-only property returning the percentage relative humidity.

Namespace: [IO.Devices.TH02](#)

Assembly: libremoteio (in libremoteio.dll) Version: 2.2020.136.3

▪ Syntax

C# VB F#

[Copy](#)

```
public double Humidity { get; }
```

Property Value

Type: [Double](#)

Implements

[SensorHumidity](#)

▪ See Also

[Reference](#)

[Device Class](#)

[IO.Devices.TH02 Namespace](#)

DeviceKelvins Property

Read-only property returning the temperature in Kelvins.

Namespace: [IO.Devices.TH02](#)

Assembly: libremoteio (in libremoteio.dll) Version: 2.2020.136.3

▪ Syntax

C# VB F#

[Copy](#)

```
public double Kelvins { get; }
```

Property Value

Type: [Double](#)

Implements

[SensorKelvins](#)

▪ See Also

[Reference](#)

[Device Class](#)

[IO.Devices.TH02 Namespace](#)

IO.Devices.Thermistor Namespace

Thermistor Modeling Services

↳ Classes

	Class	Description
	NTC_B	Encapsulate an NTC thermistor.

NTC_B Class

Encapsulate an NTC thermistor.

▪ Inheritance Hierarchy

[SystemObject](#) [IO.Devices.Thermistor](#)**NTC_B**

Namespace: [IO.Devices.Thermistor](#)

Assembly: libremoteio (in libremoteio.dll) Version: 2.2020.136.3

▪ Syntax

C# VB F#

[Copy](#)

```
public class NTC_B
```

The [NTC_B](#) type exposes the following members.

▪ Constructors

	Name	Description
≡	NTC_B	Constructor for a single NTC thermistor object instance.

[Top](#)

▪ Methods

	Name	Description
≡	Kelvins	Kelvin temperature as a function of the

thermistor resistance.

[Top](#)

◀ See Also

Reference

[IO.Devices.Thermistor Namespace](#)

NTC_B Constructor

Constructor for a single NTC thermistor object instance.

Namespace: [IO.Devices.Thermistor](#)

Assembly: libremoteio (in libremoteio.dll) Version: 2.2020.136.3

◀ Syntax

C# VB F#

[Copy](#)

```
public NTC_B(  
    double B,  
    double R0,  
    double T0 = 298.15  
)
```

Parameters

B

Type: [SystemDouble](#)

Thermistor B parameter.

R0

Type: [SystemDouble](#)

Thermistor resistance in ohms at the specified reference temperature.

T0 (Optional)

Type: [SystemDouble](#)

Thermistor reference temperature in Kelvins.

◀ See Also

Reference

NTC_B Class
IO.Devices.Thermistor Namespace

NTC_B Methods

The [NTC_B](#) type exposes the following members.

▪ Methods

	Name	Description
	Kelvins	Kelvin temperature as a function of the thermistor resistance.

[Top](#)

▪ See Also

[Reference](#)

[NTC_B Class](#)

[IO.Devices.Thermistor Namespace](#)

NTC_BKelvins Method

Kelvin temperature as a function of the thermistor resistance.

Namespace: [IO.Devices.Thermistor](#)

Assembly: libremoteio (in libremoteio.dll) Version: 2.2020.136.3

◀ Syntax

C# VB F#

[Copy](#)

```
public double Kelvins(  
    double R  
)
```

Parameters

R

Type: [SystemDouble](#)

Thermistor resistance in ohms.

Return Value

Type: [Double](#)

Temperature in Kelvins.

◀ See Also

Reference

[NTC_B Class](#)

[IO.Devices.Thermistor Namespace](#)

IO.Devices.USB.Munts Namespace

Vendor and Product Identifiers for Munts Technologies

(<http://tech.munts.com>) USB devices

◀ Classes

	Class	Description
	HID	USB device constants for Munts Technologies USB HID devices.
	Serial	USB device constants for Munts Technologies USB serial port devices.

HID Class

USB device constants for Munts Technologies USB HID devices.

▪ Inheritance Hierarchy

[SystemObject](#) [IO.Devices.USB.MuntsHID](#)

Namespace: [IO.Devices.USB.Munts](#)

Assembly: libremoteio (in libremoteio.dll) Version: 2.2020.136.3

▪ Syntax

C# VB F#

[Copy](#)

```
public static class HID
```

The [HID](#) type exposes the following members.

▪ Fields

	Name	Description
◆ S	Product	Product ID for Munts Technologies USB hid devices.
◆ S	Vendor	Vendor ID for Munts Technologies

[Top](#)

▪ See Also

[Reference](#)

IO.Devices.USB.Munts Namespace

HID Fields

The [HID](#) type exposes the following members.

↳ Fields

Name	Description
  Product	Product ID for Munts Technologies USB hid devices.
  Vendor	Vendor ID for Munts Technologies

[Top](#)

↳ See Also

[Reference](#)

[HID Class](#)

[IO.Devices.USB.Munts Namespace](#)

HIDProduct Field

Product ID for Munts Technologies USB hid devices.

Namespace: [IO.Devices.USB.Munts](#)

Assembly: libremoteio (in libremoteio.dll) Version: 2.2020.136.3

◀ Syntax

C# VB F#

[Copy](#)

```
public const int Product = 2810
```

Field Value

Type: [Int32](#)

◀ See Also

Reference

[HID Class](#)

[IO.Devices.USB.Munts Namespace](#)

HIDVendor Field

Vendor ID for Munts Technologies

Namespace: [IO.Devices.USB.Munts](#)

Assembly: libremoteio (in libremoteio.dll) Version: 2.2020.136.3

◀ Syntax

C# VB F#

[Copy](#)

```
public const int Vendor = 5840
```

Field Value

Type: [Int32](#)

◀ See Also

Reference

[HID Class](#)

[IO.Devices.USB.Munts Namespace](#)

Serial Class

USB device constants for Munts Technologies USB serial port devices.

▪ Inheritance Hierarchy

[SystemObject](#) [IO.Devices.USB.MuntsSerial](#)

Namespace: [IO.Devices.USB.Munts](#)

Assembly: libremoteio (in libremoteio.dll) Version: 2.2020.136.3

▪ Syntax

C# VB F#

[Copy](#)

```
public static class Serial
```

The [Serial](#) type exposes the following members.

▪ Fields

	Name	Description
❖ S	Product	Product ID for Munts Technologies USB serial port devices.
❖ S	Vendor	Vendor ID for Munts Technologies

[Top](#)

▪ See Also

[Reference](#)

IO.Devices.USB.Munts Namespace

Serial Fields

The [Serial](#) type exposes the following members.

↳ Fields

	Name	Description
• 	Product	Product ID for Munts Technologies USB serial port devices.
• 	Vendor	Vendor ID for Munts Technologies

[Top](#)

↳ See Also

[Reference](#)

[Serial Class](#)

[IO.Devices.USB.Munts Namespace](#)

SerialProduct Field

Product ID for Munts Technologies USB serial port devices.

Namespace: [IO.Devices.USB.Munts](#)

Assembly: libremoteio (in libremoteio.dll) Version: 2.2020.136.3

◀ Syntax

C# VB F#

[Copy](#)

```
public const int Product = 2811
```

Field Value

Type: [Int32](#)

◀ See Also

Reference

[Serial Class](#)

[IO.Devices.USB.Munts Namespace](#)

SerialVendor Field

Vendor ID for Munts Technologies

Namespace: [IO.Devices.USB.Munts](#)

Assembly: libremoteio (in libremoteio.dll) Version: 2.2020.136.3

◀ Syntax

C# VB F#

[Copy](#)

```
public const int Vendor = 5840
```

Field Value

Type: [Int32](#)

◀ See Also

Reference

[Serial Class](#)

[IO.Devices.USB.Munts Namespace](#)

IO.Interfaces.ADC Namespace

Abstract Interface for ADC (Analog to Digital Converter) Inputs

▪ Classes

	Class	Description
	Input	Encapsulates ADC voltage inputs.

▪ Interfaces

	Interface	Description
	Sample	Abstract interface for ADC inputs returning an integer sample value.
	Voltage	Abstract interface for ADC inputs returning a floating point voltage value.

Input Class

Encapsulates ADC voltage inputs.

▪ Inheritance Hierarchy

[SystemObject](#) [IO.Interfaces.ADCHandle](#)

Namespace: [IO.Interfaces.ADC](#)

Assembly: libremoteio (in libremoteio.dll) Version: 2.2020.136.3

▪ Syntax

C# VB F#

[Copy](#)

```
public class Input : Voltage
```

The [Input](#) type exposes the following members.

▪ Constructors

	Name	Description
	Input	Create an ADC voltage input.

[Top](#)

▪ Properties

	Name	Description
	voltage	Read-only property returning the analog input voltage.

[Top](#)

↳ See Also

Reference

[IO.Interfaces.ADC Namespace](#)

Input Constructor

Create an ADC voltage input.

Namespace: [IO.Interfaces.ADC](#)

Assembly: libremoteio (in libremoteio.dll) Version: 2.2020.136.3

▪ Syntax

C# VB F#

[Copy](#)

```
public Input(  
    Sample input,  
    double reference,  
    double gain = 1  
)
```

Parameters

input

Type: [IO.Interfaces.ADCSample](#)
ADC sample object.

reference

Type: [SystemDouble](#)
ADC reference in volts.

gain (Optional)

Type: [SystemDouble](#)
ADC input gain in volts per volt.

▪ See Also

[Reference](#)
[Input Class](#)

IO.Interfaces.ADC Namespace

Input Properties

The [Input](#) type exposes the following members.

▪ Properties

	Name	Description
	voltage	Read-only property returning the analog input voltage.

[Top](#)

▪ See Also

[Reference](#)

[Input Class](#)

[IO.Interfaces.ADC Namespace](#)

InputVoltage Property

Read-only property returning the analog input voltage.

Namespace: [IO.Interfaces.ADC](#)

Assembly: libremoteio (in libremoteio.dll) Version: 2.2020.136.3

▪ Syntax

[C#](#) [VB](#) [F#](#)

[Copy](#)

```
public double voltage { get; }
```

Property Value

Type: [Double](#)

Implements

[VoltageVoltage](#)

▪ See Also

[Reference](#)

[Input Class](#)

[IO.Interfaces.ADC Namespace](#)

Sample Interface

Abstract interface for ADC inputs returning an integer sample value.

Namespace: [IO.Interfaces.ADC](#)

Assembly: libremoteio (in libremoteio.dll) Version: 2.2020.136.3

▪ Syntax

[C#](#) [VB](#) [F#](#)

[Copy](#)

```
public interface Sample
```

The [Sample](#) type exposes the following members.

▪ Properties

	Name	Description
	resolution	Read-only property returning the number of bits of resolution.
	sample	Read-only property returning an integer analog sample value.

[Top](#)

▪ See Also

[Reference](#)

[IO.Interfaces.ADC Namespace](#)

Sample Properties

The [Sample](#) type exposes the following members.

Properties

	Name	Description
	resolution	Read-only property returning the number of bits of resolution.
	sample	Read-only property returning an integer analog sample value.

[Top](#)

See Also

Reference

[Sample Interface](#)

[IO.Interfaces.ADC Namespace](#)

Sampleresolution Property

Read-only property returning the number of bits of resolution.

Namespace: [IO.Interfaces.ADC](#)

Assembly: libremoteio (in libremoteio.dll) Version: 2.2020.136.3

« Syntax

C# VB F#

[Copy](#)

```
int resolution { get; }
```

Property Value

Type: [Int32](#)

« See Also

Reference

[Sample Interface](#)

[IO.Interfaces.ADC Namespace](#)

Samplesample Property

Read-only property returning an integer analog sample value.

Namespace: [IO.Interfaces.ADC](#)

Assembly: libremoteio (in libremoteio.dll) Version: 2.2020.136.3

↳ Syntax

C# VB F#

[Copy](#)

```
int sample { get; }
```

Property Value

Type: [Int32](#)

↳ See Also

Reference

[Sample Interface](#)

[IO.Interfaces.ADC Namespace](#)

Voltage Interface

Abstract interface for ADC inputs returning a floating point voltage value.

Namespace: [IO.Interfaces.ADC](#)

Assembly: libremoteio (in libremoteio.dll) Version: 2.2020.136.3

▪ Syntax

[C#](#) [VB](#) [F#](#)

[Copy](#)

```
public interface Voltage
```

The [Voltage](#) type exposes the following members.

▪ Properties

	Name	Description
	voltage	Read-only property returning a floating point analog voltage value.

[Top](#)

▪ See Also

[Reference](#)

[IO.Interfaces.ADC Namespace](#)

Voltage Properties

The [Voltage](#) type exposes the following members.

▪ Properties

	Name	Description
	voltage	Read-only property returning a floating point analog voltage value.

[Top](#)

▪ See Also

[Reference](#)

[Voltage Interface](#)

[IO.Interfaces.ADC Namespace](#)

Voltagevoltage Property

Read-only property returning a floating point analog voltage value.

Namespace: [IO.Interfaces.ADC](#)

Assembly: libremoteio (in libremoteio.dll) Version: 2.2020.136.3

↳ Syntax

C# VB F#

[Copy](#)

```
double voltage { get; }
```

Property Value

Type: [Double](#)

↳ See Also

Reference

[Voltage Interface](#)

[IO.Interfaces.ADC Namespace](#)

IO.Interfaces.DAC Namespace

Abstract Interface for DAC (Digital to Analog Converter) Outputs

▪ Classes

	Class	Description
	Output	Encapsulates DAC voltage outputs.

▪ Interfaces

	Interface	Description
	Sample	Abstract interface for DAC outputs accepting an integer output sample value.
	Voltage	Abstract interface for DAC outputs accepting a floating point output voltage value.

Output Class

Encapsulates DAC voltage outputs.

▪ Inheritance Hierarchy

[SystemObject](#) [IO.Interfaces.DACOutput](#)

Namespace: [IO.Interfaces.DAC](#)

Assembly: libremoteio (in libremoteio.dll) Version: 2.2020.136.3

▪ Syntax

C# VB F#

[Copy](#)

```
public class Output : Voltage
```

The [Output](#) type exposes the following members.

▪ Constructors

	Name	Description
	Output	Create an DAC voltage output.

[Top](#)

▪ Properties

	Name	Description
	voltage	Write-only for setting the DAC output voltage.

[Top](#)

↳ See Also

Reference

[IO.Interfaces.DAC Namespace](#)

Output Constructor

Create an DAC voltage output.

Namespace: [IO.Interfaces.DAC](#)

Assembly: libremoteio (in libremoteio.dll) Version: 2.2020.136.3

↳ Syntax

C# VB F#

[Copy](#)

```
public Output(  
    Sample output,  
    double reference,  
    double gain = 1  
)
```

Parameters

output

Type: [IO.Interfaces.DACSample](#)

DAC output object.

reference

Type: [SystemDouble](#)

DAC output reference in volts.

gain (Optional)

Type: [SystemDouble](#)

DAC output gain in volts per volt.

↳ See Also

[Reference](#)

[Output Class](#)

IO.Interfaces.DAC Namespace

Output Properties

The [Output](#) type exposes the following members.

▪ Properties

	Name	Description
	voltage	Write-only for setting the DAC output voltage.

[Top](#)

▪ See Also

[Reference](#)

[Output Class](#)

[IO.Interfaces.DAC Namespace](#)

Outputvoltage Property

Write-only for setting the DAC output voltage.

Namespace: [IO.Interfaces.DAC](#)

Assembly: libremoteio (in libremoteio.dll) Version: 2.2020.136.3

« Syntax

C# VB F#

[Copy](#)

```
public double voltage { set; }
```

Property Value

Type: [Double](#)

Implements

[Voltagevoltage](#)

« See Also

[Reference](#)

[Output Class](#)

[IO.Interfaces.DAC Namespace](#)

Sample Interface

Abstract interface for DAC outputs accepting an integer output sample value.

Namespace: [IO.Interfaces.DAC](#)

Assembly: libremoteio (in libremoteio.dll) Version: 2.2020.136.3

▪ Syntax

[C#](#) [VB](#) [F#](#)

[Copy](#)

```
public interface Sample
```

The [Sample](#) type exposes the following members.

▪ Properties

	Name	Description
	resolution	Read-only property returning the number of bits of resolution.
	sample	Write-only property for setting the DAC output level.

[Top](#)

▪ See Also

Reference

[IO.Interfaces.DAC Namespace](#)

Sample Properties

The [Sample](#) type exposes the following members.

Properties

	Name	Description
	resolution	Read-only property returning the number of bits of resolution.
	sample	Write-only property for setting the DAC output level.

[Top](#)

See Also

Reference

[Sample Interface](#)

[IO.Interfaces.DAC Namespace](#)

Sampleresolution Property

Read-only property returning the number of bits of resolution.

Namespace: [IO.Interfaces.DAC](#)

Assembly: libremoteio (in libremoteio.dll) Version: 2.2020.136.3

↳ Syntax

C# VB F#

[Copy](#)

```
int resolution { get; }
```

Property Value

Type: [Int32](#)

↳ See Also

Reference

[Sample Interface](#)

[IO.Interfaces.DAC Namespace](#)

Samplesample Property

Write-only property for setting the DAC output level.

Namespace: [IO.Interfaces.DAC](#)

Assembly: libremoteio (in libremoteio.dll) Version: 2.2020.136.3

↳ Syntax

C# VB F#

[Copy](#)

```
int sample { set; }
```

Property Value

Type: [Int32](#)

↳ See Also

Reference

[Sample Interface](#)

[IO.Interfaces.DAC Namespace](#)

Voltage Interface

Abstract interface for DAC outputs accepting a floating point output voltage value.

Namespace: [IO.Interfaces.DAC](#)

Assembly: libremoteio (in libremoteio.dll) Version: 2.2020.136.3

▪ Syntax

[C#](#) [VB](#) [F#](#)

[Copy](#)

```
public interface Voltage
```

The [Voltage](#) type exposes the following members.

▪ Properties

	Name	Description
	voltage	Write-only property for setting the DAC output voltage.

[Top](#)

▪ See Also

[Reference](#)

[IO.Interfaces.DAC Namespace](#)

Voltage Properties

The [Voltage](#) type exposes the following members.

▪ Properties

	Name	Description
	voltage	Write-only property for setting the DAC output voltage.

[Top](#)

▪ See Also

Reference

[Voltage Interface](#)

[IO.Interfaces.DAC Namespace](#)

Voltagevoltage Property

Write-only property for setting the DAC output voltage.

Namespace: [IO.Interfaces.DAC](#)

Assembly: libremoteio (in libremoteio.dll) Version: 2.2020.136.3

« Syntax

C# VB F#

[Copy](#)

```
double voltage { set; }
```

Property Value

Type: [Double](#)

« See Also

Reference

[Voltage Interface](#)

[IO.Interfaces.DAC Namespace](#)

IO.Interfaces.GPIO Namespace

Abstract Interface for GPIO (General Purpose Input/Output) Pins

▪ Interfaces

	Interface	Description
	Pin	Abstract interface for GPIO pins.

▪ Enumerations

	Enumeration	Description
	Direction	GPIO pin data direction settings.

Direction Enumeration

GPIO pin data direction settings.

Namespace: [IO.Interfaces.GPIO](#)

Assembly: libremoteio (in libremoteio.dll) Version: 2.2020.136.3

▪ Syntax

C# VB F#

[Copy](#)

```
public enum Direction
```

▪ Members

Member name	Value	Description
Input	0	Input pin (read only)
Output	1	Output pin (read or write)

▪ See Also

Reference

[IO.Interfaces.GPIO Namespace](#)

Pin Interface

Abstract interface for GPIO pins.

Namespace: [IO.Interfaces.GPIO](#)

Assembly: libremoteio (in libremoteio.dll) Version: 2.2020.136.3

▪ Syntax

C# VB F#

[Copy](#)

```
public interface Pin
```

The [Pin](#) type exposes the following members.

▪ Properties

Name	Description
 state	Read/Write GPIO state property.

[Top](#)

▪ See Also

[Reference](#)

[IO.Interfaces.GPIO Namespace](#)

Pin Properties

The [Pin](#) type exposes the following members.

Properties

	Name	Description
	state	Read/Write GPIO state property.

[Top](#)

See Also

[Reference](#)

[Pin Interface](#)

[IO.Interfaces.GPIO Namespace](#)

Pinstate Property

Read/Write GPIO state property.

Namespace: [IO.Interfaces.GPIO](#)

Assembly: libremoteio (in libremoteio.dll) Version: 2.2020.136.3

« Syntax

C# VB F#

[Copy](#)

```
bool state { get; set; }
```

Property Value

Type: [Boolean](#)

« See Also

Reference

[Pin Interface](#)

[IO.Interfaces.GPIO Namespace](#)

IO.Interfaces.Humidity Namespace

Abstract Interface for Humidity Sensors

► Interfaces

Interface	Description
 Sensor	Abstract interface for humidity sensors.

Sensor Interface

Abstract interface for humidity sensors.

Namespace: [IO.Interfaces.Humidity](#)

Assembly: libremoteio (in libremoteio.dll) Version: 2.2020.136.3

▪ Syntax

C# VB F#

[Copy](#)

```
public interface Sensor
```

The [Sensor](#) type exposes the following members.

▪ Properties

Name	Description
	Humidity Read-only property returning the percentage relative humidity.

[Top](#)

▪ See Also

Reference

[IO.Interfaces.Humidity Namespace](#)

Sensor Properties

The [Sensor](#) type exposes the following members.

▪ Properties

	Name	Description
	Humidity	Read-only property returning the percentage relative humidity.

[Top](#)

▪ See Also

[Reference](#)

[Sensor Interface](#)

[IO.Interfaces.Humidity Namespace](#)

SensorHumidity Property

Read-only property returning the percentage relative humidity.

Namespace: [IO.Interfaces.Humidity](#)

Assembly: libremoteio (in libremoteio.dll) Version: 2.2020.136.3

↳ Syntax

C# VB F#

[Copy](#)

```
double Humidity { get; }
```

Property Value

Type: [Double](#)

↳ See Also

Reference

[Sensor Interface](#)

[IO.Interfaces.Humidity Namespace](#)

IO.Interfaces.I2C Namespace

Abstract Interface for I²C (Inter-Integrated Circuit) Bus Controllers

► Classes

	Class	Description
	Device	Encapsulates a single I ² C slave device.
	Speeds	I ² C bus speed constants.

► Interfaces

	Interface	Description
	Bus	Abstract interface for I ² C bus controllers.

Bus Interface

Abstract interface for I²C bus controllers.

Namespace: [IO.Interfaces.I2C](#)

Assembly: libremoteio (in libremoteio.dll) Version: 2.2020.136.3

▪ Syntax

[C#](#) [VB](#) [F#](#)

[Copy](#)

```
public interface Bus
```

The [Bus](#) type exposes the following members.

▪ Methods

	Name	Description
≡	Read	Read bytes from an I ² C slave device.
≡	Transaction	Write and read bytes to and from an I ² C slave device.
≡	Write	Write bytes to an I ² C slave device.

[Top](#)

▪ See Also

[Reference](#)

[IO.Interfaces.I2C Namespace](#)

Bus Methods

The [Bus](#) type exposes the following members.

▪ Methods

	Name	Description
	Read	Read bytes from an I ² C slave device.
	Transaction	Write and read bytes to and from an I ² C slave device.
	Write	Write bytes to an I ² C slave device.

[Top](#)

▪ See Also

Reference

[Bus Interface](#)

[IO.Interfaces.I2C Namespace](#)

BusRead Method

Read bytes from an I²C slave device.

Namespace: [IO.Interfaces.I2C](#)

Assembly: libremoteio (in libremoteio.dll) Version: 2.2020.136.3

↳ Syntax

C# VB F#

[Copy](#)

```
void Read(  
    int slaveaddr,  
    byte[] resp,  
    int resplen  
)
```

Parameters

slaveaddr

Type: [SystemInt32](#)

I²C slave address.

resp

Type: [SystemByte](#)

Response buffer.

resplen

Type: [SystemInt32](#)

Number of bytes to read.

↳ See Also

Reference

Bus Interface

IO.Interfaces.I2C Namespace

BusTransaction Method

Write and read bytes to and from an I²C slave device.

Namespace: [IO.Interfaces.I2C](#)

Assembly: libremoteio (in libremoteio.dll) Version: 2.2020.136.3

↳ Syntax

C# VB F#

[Copy](#)

```
void Transaction(  
    int slaveaddr,  
    byte[] cmd,  
    int cmdlen,  
    byte[] resp,  
    int resplen,  
    int delayus = 0  
)
```

Parameters

slaveaddr

Type: [SystemInt32](#)

I²C slave address.

cmd

Type: [SystemByte](#)

Command buffer.

cmdlen

Type: [SystemInt32](#)

Number of bytes to write.

resp

Type: [SystemByte](#)

Response buffer.

resplen

Type: [SystemInt32](#)

Number of bytes to read.

delayus (Optional)

Type: [SystemInt32](#)

Delay in microseconds between the I²C write and read cycles.

Allowed values are 0 to 65535 microseconds.

See Also

Reference

[Bus Interface](#)

[IO.Interfaces.I2C Namespace](#)

BusWrite Method

Write bytes to an I²C slave device.

Namespace: [IO.Interfaces.I2C](#)

Assembly: libremoteio (in libremoteio.dll) Version: 2.2020.136.3

« Syntax

C# VB F#

[Copy](#)

```
void Write(  
    int slaveaddr,  
    byte[] cmd,  
    int cmdlen  
)
```

Parameters

slaveaddr

Type: [SystemInt32](#)

I²C slave address.

cmd

Type: [SystemByte](#)

Command buffer.

cmdlen

Type: [SystemInt32](#)

Number of bytes to write.

« See Also

Reference

Bus Interface

IO.Interfaces.I2C Namespace

Device Class

Encapsulates a single I²C slave device.

▪ Inheritance Hierarchy

[SystemObject](#) [IO.Interfaces.I2CDevice](#)

Namespace: [IO.Interfaces.I2C](#)

Assembly: libremoteio (in libremoteio.dll) Version: 2.2020.136.3

▪ Syntax

[C#](#) [VB](#) [F#](#)

[Copy](#)

```
public class Device
```

The [Device](#) type exposes the following members.

▪ Constructors

	Name	Description
≡	Device	Create an I ² C slave device.

[Top](#)

▪ Methods

	Name	Description
≡	Read	Read bytes from an I ² C slave device.
≡		

Transaction Write and read bytes to and from an I²C slave device.



Write Write bytes to an I²C slave device.

[Top](#)

◀ See Also

Reference

[IO.Interfaces.I2C Namespace](#)

Device Constructor

Create an I²C slave device.

Namespace: [IO.Interfaces.I2C](#)

Assembly: libremoteio (in libremoteio.dll) Version: 2.2020.136.3

▪ Syntax

C# VB F#

[Copy](#)

```
public Device(  
    Bus bus,  
    int slaveaddr  
)
```

Parameters

bus

Type: [IO.Interfaces.I2CBus](#)

I²C bus controller object.

slaveaddr

Type: [SystemInt32](#)

I²C slave address.

▪ See Also

[Reference](#)

[Device Class](#)

[IO.Interfaces.I2C Namespace](#)

Device Methods

The [Device](#) type exposes the following members.

▪ Methods

	Name	Description
	Read	Read bytes from an I ² C slave device.
	Transaction	Write and read bytes to and from an I ² C slave device.
	Write	Write bytes to an I ² C slave device.

[Top](#)

▪ See Also

Reference

[Device Class](#)

[IO.Interfaces.I2C Namespace](#)

DeviceRead Method

Read bytes from an I²C slave device.

Namespace: [IO.Interfaces.I2C](#)

Assembly: libremoteio (in libremoteio.dll) Version: 2.2020.136.3

▪ Syntax

C# VB F#

[Copy](#)

```
public void Read(  
    byte[] resp,  
    int resplen  
)
```

Parameters

resp

Type: [SystemByte](#)

Response buffer.

resplen

Type: [SystemInt32](#)

Number of bytes to read.

▪ See Also

[Reference](#)

[Device Class](#)

[IO.Interfaces.I2C Namespace](#)

DeviceTransaction Method

Write and read bytes to and from an I²C slave device.

Namespace: [IO.Interfaces.I2C](#)

Assembly: libremoteio (in libremoteio.dll) Version: 2.2020.136.3

↳ Syntax

C# VB F#

[Copy](#)

```
public void Transaction(  
    byte[] cmd,  
    int cmdLen,  
    byte[] resp,  
    int resplen,  
    int delayus = 0  
)
```

Parameters

cmd

Type: [SystemByte](#)

Command buffer.

cmdlen

Type: [SystemInt32](#)

Number of bytes to write.

resp

Type: [SystemByte](#)

Response buffer.

resplen

Type: [SystemInt32](#)

Number of bytes to read.

delayus (Optional)

Type: [SystemInt32](#)

Delay in microseconds between the I²C write and read cycles.

Allowed values are 0 to 65535 microseconds.

See Also

Reference

[Device Class](#)

[IO.Interfaces.I2C Namespace](#)

DeviceWrite Method

Write bytes to an I²C slave device.

Namespace: [IO.Interfaces.I2C](#)

Assembly: libremoteio (in libremoteio.dll) Version: 2.2020.136.3

▪ Syntax

C# VB F#

[Copy](#)

```
public void Write(  
    byte[] cmd,  
    int cmdLen  
)
```

Parameters

cmd

Type: [SystemByte](#)

Command buffer.

cmdlen

Type: [SystemInt32](#)

Number of bytes to write.

▪ See Also

[Reference](#)

[Device Class](#)

[IO.Interfaces.I2C Namespace](#)

Speeds Class

I²C bus speed constants.

▪ Inheritance Hierarchy

[SystemObject](#) [IO.Interfaces.I2CSpeeds](#)

Namespace: [IO.Interfaces.I2C](#)

Assembly: libremoteio (in libremoteio.dll) Version: 2.2020.136.3

▪ Syntax

[C#](#) [VB](#) [F#](#)

[Copy](#)

```
public static class Speeds
```

The [Speeds](#) type exposes the following members.

▪ Fields

	Name	Description
• S	FastMode	Fast Mode
• S	FastModePlus	Fast Mode Plus
• S	StandardMode	Standard Mode

[Top](#)

▪ See Also

Reference

[IO.Interfaces.I2C Namespace](#)

Speeds Fields

The [Speeds](#) type exposes the following members.

↳ Fields

	Name	Description
• 	FastMode	Fast Mode
• 	FastModePlus	Fast Mode Plus
• 	StandardMode	Standard Mode

[Top](#)

↳ See Also

[Reference](#)

[Speeds Class](#)

[IO.Interfaces.I2C Namespace](#)

SpeedsFastMode Field

Fast Mode

Namespace: [IO.Interfaces.I2C](#)

Assembly: libremoteio (in libremoteio.dll) Version: 2.2020.136.3

◀ Syntax

C# VB F#

[Copy](#)

```
public const int FastMode = 400000
```

Field Value

Type: [Int32](#)

◀ See Also

Reference

[Speeds Class](#)

[IO.Interfaces.I2C Namespace](#)

SpeedsFastModePlus Field

Fast Mode Plus

Namespace: [IO.Interfaces.I2C](#)

Assembly: libremoteio (in libremoteio.dll) Version: 2.2020.136.3

◀ Syntax

C# VB F#

[Copy](#)

```
public const int FastModePlus = 1000000
```

Field Value

Type: [Int32](#)

◀ See Also

Reference

[Speeds Class](#)

[IO.Interfaces.I2C Namespace](#)

SpeedsStandardMode Field

Standard Mode

Namespace: [IO.Interfaces.I2C](#)

Assembly: libremoteio (in libremoteio.dll) Version: 2.2020.136.3

↳ Syntax

C# VB F#

[Copy](#)

```
public const int StandardMode = 100000
```

Field Value

Type: [Int32](#)

↳ See Also

Reference

[Speeds Class](#)

[IO.Interfaces.I2C Namespace](#)

IO.Interfaces.Message64 Namespace

Abstract Interface for 64-Byte Message Services

▪ Classes

	Class	Description
	Message	Encapsulates 64-byte messages.

▪ Interfaces

	Interface	Description
	Messenger	Abstract interface for sending and receiving 64-byte messages.

Message Class

Encapsulates 64-byte messages.

▪ Inheritance Hierarchy

[SystemObject](#) [IO.Interfaces.Message64Message](#)

Namespace: [IO.Interfaces.Message64](#)

Assembly: libremoteio (in libremoteio.dll) Version: 2.2020.136.3

▪ Syntax

C# VB F#

[Copy](#)

```
public class Message
```

The [Message](#) type exposes the following members.

▪ Constructors

	Name	Description
≡	Message	Create a message object without initializing the payload.
≡	Message(Byte)	Create a message object with an initialized payload.

[Top](#)

▪ Fields

Name	Description
• payload	Message payload.
• S Size	Message payload size.

[Top](#)

See Also

Reference

[IO.Interfaces.Message64 Namespace](#)

Message Constructor

↳ Overload List

	Name	Description
≡	Message	Create a message object without initializing the payload.
≡	Message(Byte)	Create a message object with an initialized payload.

[Top](#)

↳ See Also

Reference

[Message Class](#)

[IO.Interfaces.Message64 Namespace](#)

Message Constructor

Create a message object without initializing the payload.

Namespace: [IO.Interfaces.Message64](#)

Assembly: libremoteio (in libremoteio.dll) Version: 2.2020.136.3

▪ Syntax

[C#](#) [VB](#) [F#](#)

[Copy](#)

```
public Message()
```

▪ See Also

Reference

[Message Class](#)

[Message Overload](#)

[IO.Interfaces.Message64 Namespace](#)

Message Constructor (Byte)

Create a message object with an initialized payload.

Namespace: [IO.Interfaces.Message64](#)

Assembly: libremoteio (in libremoteio.dll) Version: 2.2020.136.3

▀ Syntax

C# VB F#

[Copy](#)

```
public Message(  
    byte fill  
)
```

Parameters

fill

Type: [System.Byte](#)

Value to initialize the payload with.

▀ See Also

Reference

[Message Class](#)

[Message Overload](#)

[IO.Interfaces.Message64 Namespace](#)

Message Fields

The [Message](#) type exposes the following members.

Fields

	Name	Description
◆	payload	Message payload.
◆ S	Size	Message payload size.

[Top](#)

See Also

Reference

[Message Class](#)

[IO.Interfaces.Message64 Namespace](#)

Messagepayload Field

Message payload.

Namespace: [IO.Interfaces.Message64](#)

Assembly: libremoteio (in libremoteio.dll) Version: 2.2020.136.3

« Syntax

C# VB F#

[Copy](#)

```
public byte[] payload
```

Field Value

Type: [Byte](#)

« See Also

Reference

[Message Class](#)

[IO.Interfaces.Message64 Namespace](#)

MessageSize Field

Message payload size.

Namespace: [IO.Interfaces.Message64](#)

Assembly: libremoteio (in libremoteio.dll) Version: 2.2020.136.3

« Syntax

C# VB F#

[Copy](#)

```
public const int Size = 64
```

Field Value

Type: [Int32](#)

« See Also

Reference

[Message Class](#)

[IO.Interfaces.Message64 Namespace](#)

Messenger Interface

Abstract interface for sending and receiving 64-byte messages.

Namespace: [IO.Interfaces.Message64](#)

Assembly: libremoteio (in libremoteio.dll) Version: 2.2020.136.3

▪ Syntax

C# VB F#

[Copy](#)

```
public interface Messenger
```

The [Messenger](#) type exposes the following members.

▪ Methods

	Name	Description
≡	Receive	Receive a 64-byte message.
≡	Send	Send a 64-byte message.
≡	Transaction	Send a 64-byte command and receive a 64-byte response.

[Top](#)

▪ See Also

Reference

[IO.Interfaces.Message64 Namespace](#)

Messenger Methods

The [Messenger](#) type exposes the following members.

▪ Methods

	Name	Description
	Receive	Receive a 64-byte message.
	Send	Send a 64-byte message.
	Transaction	Send a 64-byte command and receive a 64-byte response.

[Top](#)

▪ See Also

Reference

[Messenger Interface](#)

[IO.Interfaces.Message64 Namespace](#)

MessengerReceive Method

Receive a 64-byte message.

Namespace: [IO.Interfaces.Message64](#)

Assembly: libremoteio (in libremoteio.dll) Version: 2.2020.136.3

▪ Syntax

C# VB F#

[Copy](#)

```
void Receive(  
    Message resp  
)
```

Parameters

resp

Type: [IO.Interfaces.Message64Message](#)

Message received.

▪ See Also

Reference

[Messenger Interface](#)

[IO.Interfaces.Message64 Namespace](#)

MessengerSend Method

Send a 64-byte message.

Namespace: [IO.Interfaces.Message64](#)

Assembly: libremoteio (in libremoteio.dll) Version: 2.2020.136.3

◀ Syntax

C# VB F#

[Copy](#)

```
void Send(  
    Message cmd  
)
```

Parameters

cmd

Type: [IO.Interfaces.Message64Message](#)

Message to be sent.

◀ See Also

Reference

[Messenger Interface](#)

[IO.Interfaces.Message64 Namespace](#)

MessengerTransaction Method

Send a 64-byte command and receive a 64-byte response.

Namespace: [IO.Interfaces.Message64](#)

Assembly: libremoteio (in libremoteio.dll) Version: 2.2020.136.3

◀ Syntax

C# VB F#

[Copy](#)

```
void Transaction(  
    Message cmd,  
    Message resp  
)
```

Parameters

cmd

Type: [IO.Interfaces.Message64Message](#)

Command to be sent.

resp

Type: [IO.Interfaces.Message64Message](#)

Response to be received.

◀ See Also

Reference

[Messenger Interface](#)

[IO.Interfaces.Message64 Namespace](#)

IO.Interfaces.Motor Namespace

Abstract Interface For Variable Speed Motor Outputs

▪ Classes

	Class	Description
	Velocities	Motor velocity constants.

▪ Interfaces

	Interface	Description
	Output	Abstract interface for variable speed motor outputs.

Output Interface

Abstract interface for variable speed motor outputs.

Namespace: [IO.Interfaces.Motor](#)

Assembly: libremoteio (in libremoteio.dll) Version: 2.2020.136.3

▪ Syntax

[C#](#) [VB](#) [F#](#)

[Copy](#)

```
public interface Output
```

The [Output](#) type exposes the following members.

▪ Properties

Name	Description
 velocity	Write-only motor velocity property.

[Top](#)

▪ See Also

[Reference](#)

[IO.Interfaces.Motor Namespace](#)

Output Properties

The [Output](#) type exposes the following members.

Properties

	Name	Description
	velocity	Write-only motor velocity property.

[Top](#)

See Also

Reference

[Output Interface](#)

[IO.Interfaces.Motor Namespace](#)

Outputvelocity Property

Write-only motor velocity property.

Namespace: [IO.Interfaces.Motor](#)

Assembly: libremoteio (in libremoteio.dll) Version: 2.2020.136.3

↳ Syntax

C# VB F#

[Copy](#)

```
double velocity { set; }
```

Property Value

Type: [Double](#)

↳ See Also

Reference

[Output Interface](#)

[IO.Interfaces.Motor Namespace](#)

Velocities Class

Motor velocity constants.

▪ Inheritance Hierarchy

[SystemObject](#) [IO.Interfaces.MotorVelocities](#)

Namespace: [IO.Interfaces.Motor](#)

Assembly: libremoteio (in libremoteio.dll) Version: 2.2020.136.3

▪ Syntax

C# VB F#

[Copy](#)

```
public static class Velocities
```

The [Velocities](#) type exposes the following members.

▪ Fields

	Name	Description
◆ S	Maximum	Maximum velocity (full speed forward).
◆ S	Minimum	Minimum velocity (full speed reverse).
◆ S	Stop	Zero velocity (motor stopped).

[Top](#)

▪ See Also

Reference

IO.Interfaces.Motor Namespace

Velocities Fields

The [Velocities](#) type exposes the following members.

↳ Fields

	Name	Description
•  	Maximum	Maximum velocity (full speed forward).
•  	Minimum	Minimum velocity (full speed reverse).
•  	Stop	Zero velocity (motor stopped).

[Top](#)

↳ See Also

Reference

[Velocities Class](#)

[IO.Interfaces.Motor Namespace](#)

VelocitiesMaximum Field

Maximum velocity (full speed forward).

Namespace: [IO.Interfaces.Motor](#)

Assembly: libremoteio (in libremoteio.dll) Version: 2.2020.136.3

« Syntax

C# VB F#

[Copy](#)

```
public const double Maximum = 1
```

Field Value

Type: [Double](#)

« See Also

Reference

[Velocities Class](#)

[IO.Interfaces.Motor Namespace](#)

VelocitiesMinimum Field

Minimum velocity (full speed reverse).

Namespace: [IO.Interfaces.Motor](#)

Assembly: libremoteio (in libremoteio.dll) Version: 2.2020.136.3

« Syntax

C# VB F#

[Copy](#)

```
public const double Minimum = -1
```

Field Value

Type: [Double](#)

« See Also

Reference

[Velocities Class](#)

[IO.Interfaces.Motor Namespace](#)

VelocitiesStop Field

Zero velocity (motor stopped).

Namespace: [IO.Interfaces.Motor](#)

Assembly: libremoteio (in libremoteio.dll) Version: 2.2020.136.3

◀ Syntax

C# VB F#

[Copy](#)

```
public const double Stop = 0
```

Field Value

Type: [Double](#)

◀ See Also

Reference

[Velocities Class](#)

[IO.Interfaces.Motor Namespace](#)

IO.Interfaces.PWM Namespace

Abstract Interface for PWM (Pulse Width Modulated) Outputs

▪ Classes

	Class	Description
	DutyCycles	PWM dutycycle constants.

▪ Interfaces

	Interface	Description
	Output	Abstract interface for PWM outputs.

DutyCycles Class

PWM dutycycle constants.

▪ Inheritance Hierarchy

[SystemObject](#) [IO.Interfaces.PWMDutyCycles](#)

Namespace: [IO.Interfaces.PWM](#)

Assembly: libremoteio (in libremoteio.dll) Version: 2.2020.136.3

▪ Syntax

C# VB F#

[Copy](#)

```
public static class DutyCycles
```

The [DutyCycles](#) type exposes the following members.

▪ Fields

	Name	Description
◆ S	Maximum	Maximum duty cycle (percent).
◆ S	Minimum	Minimum duty cycle (percent).

[Top](#)

▪ See Also

Reference

[IO.Interfaces.PWM Namespace](#)

DutyCycles Fields

The [DutyCycles](#) type exposes the following members.

↳ Fields

	Name	Description
• 	Maximum	Maximum duty cycle (percent).
• 	Minimum	Minimum duty cycle (percent).

[Top](#)

↳ See Also

Reference

[DutyCycles Class](#)

[IO.Interfaces.PWM Namespace](#)

DutyCyclesMaximum Field

Maximum duty cycle (percent).

Namespace: [IO.Interfaces.PWM](#)

Assembly: libremoteio (in libremoteio.dll) Version: 2.2020.136.3

◀ Syntax

C# VB F#

[Copy](#)

```
public const double Maximum = 100
```

Field Value

Type: [Double](#)

◀ See Also

Reference

[DutyCycles Class](#)

[IO.Interfaces.PWM Namespace](#)

DutyCyclesMinimum Field

Minimum duty cycle (percent).

Namespace: [IO.Interfaces.PWM](#)

Assembly: libremoteio (in libremoteio.dll) Version: 2.2020.136.3

◀ Syntax

C# VB F#

[Copy](#)

```
public const double Minimum = 0
```

Field Value

Type: [Double](#)

◀ See Also

Reference

[DutyCycles Class](#)

[IO.Interfaces.PWM Namespace](#)

Output Interface

Abstract interface for PWM outputs.

Namespace: [IO.Interfaces.PWM](#)

Assembly: libremoteio (in libremoteio.dll) Version: 2.2020.136.3

▪ Syntax

[C#](#) [VB](#) [F#](#)

[Copy](#)

```
public interface Output
```

The [Output](#) type exposes the following members.

▪ Properties

Name	Description
 dutycycle	Write-only PWM duty cycle property.

[Top](#)

▪ See Also

[Reference](#)

[IO.Interfaces.PWM Namespace](#)

Output Properties

The [Output](#) type exposes the following members.

Properties

	Name	Description
	dutycycle	Write-only PWM duty cycle property.

[Top](#)

See Also

Reference

[Output Interface](#)

[IO.Interfaces.PWM Namespace](#)

Outputdutycycle Property

Write-only PWM duty cycle property.

Namespace: [IO.Interfaces.PWM](#)

Assembly: libremoteio (in libremoteio.dll) Version: 2.2020.136.3

« Syntax

C# VB F#

[Copy](#)

```
double dutycycle { set; }
```

Property Value

Type: [Double](#)

« See Also

Reference

[Output Interface](#)

[IO.Interfaces.PWM Namespace](#)

IO.Interfaces.Servo Namespace

Abstract Interface for Servo Outputs

▪ Classes

	Class	Description
	Positions	Servo position constants.

▪ Interfaces

	Interface	Description
	Output	Abstract interface for servo outputs.

Output Interface

Abstract interface for servo outputs.

Namespace: [IO.Interfaces.Servo](#)

Assembly: libremoteio (in libremoteio.dll) Version: 2.2020.136.3

▪ Syntax

[C#](#) [VB](#) [F#](#)

[Copy](#)

```
public interface Output
```

The [Output](#) type exposes the following members.

▪ Properties

Name	Description
 position	Write-only servo position property.

[Top](#)

▪ See Also

[Reference](#)

[IO.Interfaces.Servo Namespace](#)

Output Properties

The [Output](#) type exposes the following members.

Properties

	Name	Description
	position	Write-only servo position property.

[Top](#)

See Also

Reference

[Output Interface](#)

[IO.Interfaces.Servo Namespace](#)

Outputposition Property

Write-only servo position property.

Namespace: [IO.Interfaces.Servo](#)

Assembly: libremoteio (in libremoteio.dll) Version: 2.2020.136.3

« Syntax

C# VB F#

[Copy](#)

```
double position { set; }
```

Property Value

Type: [Double](#)

« See Also

Reference

[Output Interface](#)

[IO.Interfaces.Servo Namespace](#)

Positions Class

Servo position constants.

▪ Inheritance Hierarchy

[SystemObject](#) [IO.Interfaces.ServoPositions](#)

Namespace: [IO.Interfaces.Servo](#)

Assembly: libremoteio (in libremoteio.dll) Version: 2.2020.136.3

▪ Syntax

C# VB F#

[Copy](#)

```
public static class Positions
```

The [Positions](#) type exposes the following members.

▪ Fields

	Name	Description
❖ S	Maximum	Maximum displacement position.
❖ S	Minimum	Minimum displacement position.
❖ S	Neutral	Zero displacement (neutral) position.

[Top](#)

▪ See Also

Reference

[IO.Interfaces.Servo Namespace](#)

Positions Fields

The [Positions](#) type exposes the following members.

↳ Fields

	Name	Description
•  	Maximum	Maximum displacement position.
•  	Minimum	Minimum displacement position.
•  	Neutral	Zero displacement (neutral) position.

[Top](#)

↳ See Also

Reference

[Positions Class](#)

[IO.Interfaces.Servo Namespace](#)

PositionsMaximum Field

Maximum displacement position.

Namespace: [IO.Interfaces.Servo](#)

Assembly: libremoteio (in libremoteio.dll) Version: 2.2020.136.3

« Syntax

C# VB F#

[Copy](#)

```
public const double Maximum = 1
```

Field Value

Type: [Double](#)

« See Also

Reference

[Positions Class](#)

[IO.Interfaces.Servo Namespace](#)

PositionsMinimum Field

Minimum displacement position.

Namespace: [IO.Interfaces.Servo](#)

Assembly: libremoteio (in libremoteio.dll) Version: 2.2020.136.3

↳ Syntax

C# VB F#

[Copy](#)

```
public const double Minimum = -1
```

Field Value

Type: [Double](#)

↳ See Also

Reference

[Positions Class](#)

[IO.Interfaces.Servo Namespace](#)

PositionsNeutral Field

Zero displacement (neutral) position.

Namespace: [IO.Interfaces.Servo](#)

Assembly: libremoteio (in libremoteio.dll) Version: 2.2020.136.3

↳ Syntax

C# VB F#

[Copy](#)

```
public const double Neutral = 0
```

Field Value

Type: [Double](#)

↳ See Also

Reference

[Positions Class](#)

[IO.Interfaces.Servo Namespace](#)

IO.Interfaces.SPI Namespace

Abstract Interface for SPI (Serial Peripheral Interconnect) Slave Devices

▪ Interfaces

Interface	Description
Device	Abstract interface for SPI slave devices.

Device Interface

Abstract interface for SPI slave devices.

Namespace: [IO.Interfaces.SPI](#)

Assembly: libremoteio (in libremoteio.dll) Version: 2.2020.136.3

▪ Syntax

C# VB F#

[Copy](#)

```
public interface Device
```

The [Device](#) type exposes the following members.

▪ Methods

	Name	Description
≡	Read	Read bytes from an SPI slave device.
≡	Transaction	Write bytes to and read bytes from an SPI slave device.
≡	Write	Write bytes to an SPI slave device.

[Top](#)

▪ See Also

Reference

[IO.Interfaces.SPI Namespace](#)

Device Methods

The [Device](#) type exposes the following members.

▪ Methods

	Name	Description
	Read	Read bytes from an SPI slave device.
	Transaction	Write bytes to and read bytes from an SPI slave device.
	Write	Write bytes to an SPI slave device.

[Top](#)

▪ See Also

Reference

[Device Interface](#)

[IO.Interfaces.SPI Namespace](#)

DeviceRead Method

Read bytes from an SPI slave device.

Namespace: [IO.Interfaces.SPI](#)

Assembly: libremoteio (in libremoteio.dll) Version: 2.2020.136.3

◀ Syntax

C# VB F#

[Copy](#)

```
void Read(  
    byte[] resp,  
    int resplen  
)
```

Parameters

resp

Type: [SystemByte](#)

Response buffer.

resplen

Type: [SystemInt32](#)

Number of bytes to read.

◀ See Also

[Reference](#)

[Device Interface](#)

[IO.Interfaces.SPI Namespace](#)

DeviceTransaction Method

Write bytes to and read bytes from an SPI slave device.

Namespace: [IO.Interfaces.SPI](#)

Assembly: libremoteio (in libremoteio.dll) Version: 2.2020.136.3

◀ Syntax

C# VB F#

[Copy](#)

```
void Transaction(  
    byte[] cmd,  
    int cmdLen,  
    byte[] resp,  
    int resplen,  
    int delayus = 0  
)
```

Parameters

cmd

Type: [SystemByte](#)

Command buffer.

cmdlen

Type: [SystemInt32](#)

Number of bytes to write.

resp

Type: [SystemByte](#)

Response buffer.

resplen

Type: [SystemInt32](#)

Number of bytes to read.

delayus (Optional)

Type: [SystemInt32](#)

Delay in microseconds between write and read operations.

See Also

Reference

[Device Interface](#)

[IO.Interfaces.SPI Namespace](#)

DeviceWrite Method

Write bytes to an SPI slave device.

Namespace: [IO.Interfaces.SPI](#)

Assembly: libremoteio (in libremoteio.dll) Version: 2.2020.136.3

◀ Syntax

C# VB F#

[Copy](#)

```
void Write(  
    byte[] cmd,  
    int cmdLen  
)
```

Parameters

cmd

Type: [SystemByte](#)

Command buffer.

cmdlen

Type: [SystemInt32](#)

Number of bytes to write.

◀ See Also

Reference

[Device Interface](#)

[IO.Interfaces.SPI Namespace](#)

IO.Interfaces.Temperature Namespace

Abstract Interface for Temperature Sensors

▪ Classes

Class	Description
 Conversions	Temperature conversion functions.

▪ Interfaces

Interface	Description
 Sensor	Abstract interface for temperature sensors.

Conversions Class

Temperature conversion functions.

▪ Inheritance Hierarchy

[SystemObject](#) [IO.Interfaces.TemperatureConversions](#)

Namespace: [IO.Interfaces.Temperature](#)

Assembly: libremoteio (in libremoteio.dll) Version: 2.2020.136.3

▪ Syntax

C# VB F#

[Copy](#)

```
public class Conversions
```

The [Conversions](#) type exposes the following members.

▪ Constructors

	Name	Description
≡	Conversions	Initializes a new instance of the Conversions class

[Top](#)

▪ Methods

	Name	Description
≡ S	CelsiusToFahrenheit	Convert degrees Celsius to

degrees Fahrenheit.

 S	CelsiusToKelvins	Convert degrees Celsius to Kelvins.
 S	FahrenheitToCelsius	Convert degrees Fahrenheit to degrees Celsius.
 S	FahrenheitToKelvins	Convert degrees Fahrenheit to Kelvins.
 S	KelvinsToCelsius	Convert Kelvins to degrees Celsius.
 S	KelvinsToFahrenheit	Convert Kelvins to degrees Fahrenheit.

[Top](#)

See Also

Reference

[IO.Interfaces.Temperature Namespace](#)

Conversions Constructor

Initializes a new instance of the [Conversions](#) class

Namespace: [IO.Interfaces.Temperature](#)

Assembly: libremoteio (in libremoteio.dll) Version: 2.2020.136.3

▪ Syntax

C# VB F#

[Copy](#)

```
public Conversions()
```

▪ See Also

Reference

[Conversions Class](#)

[IO.Interfaces.Temperature Namespace](#)

Conversions Methods

The [Conversions](#) type exposes the following members.

▪ Methods

	Name	Description
 	CelsiusToFahrenheit	Convert degrees Celsius to degrees Fahrenheit.
 	CelsiusToKelvins	Convert degrees Celsius to Kelvins.
 	FahrenheitToCelsius	Convert degrees Fahrenheit to degrees Celsius.
 	FahrenheitToKelvins	Convert degrees Fahrenheit to Kelvins.
 	KelvinsToCelsius	Convert Kelvins to degrees Celsius.
 	KelvinsToFahrenheit	Convert Kelvins to degrees Fahrenheit.

[Top](#)

▪ See Also

[Reference](#)

[Conversions Class](#)

IO.Interfaces.Temperature Namespace

ConversionsCelsiusToFahrenheit Method

Convert degrees Celsius to degrees Fahrenheit.

Namespace: [IO.Interfaces.Temperature](#)

Assembly: libremoteio (in libremoteio.dll) Version: 2.2020.136.3

◀ Syntax

[C#](#) [VB](#) [F#](#)

[Copy](#)

```
public static double CelsiusToFahrenheit(  
    double celsius  
)
```

Parameters

celsius

Type: [SystemDouble](#)

Temperature in degrees Celsius.

Return Value

Type: [Double](#)

Temperature in degrees Fahrenheit.

◀ See Also

Reference

[Conversions Class](#)

[IO.Interfaces.Temperature Namespace](#)

ConversionsCelsiusToKelvins Method

Convert degrees Celsius to Kelvins.

Namespace: [IO.Interfaces.Temperature](#)

Assembly: libremoteio (in libremoteio.dll) Version: 2.2020.136.3

◀ Syntax

[C#](#) [VB](#) [F#](#)

[Copy](#)

```
public static double CelsiusToKelvins(  
    double celsius  
)
```

Parameters

celsius

Type: [SystemDouble](#)

Temperature in degrees Celsius.

Return Value

Type: [Double](#)

Temperature in Kelvins.

◀ See Also

Reference

[Conversions Class](#)

[IO.Interfaces.Temperature Namespace](#)

ConversionsFahrenheitToCelsius Method

Convert degrees Fahrenheit to degrees Celsius.

Namespace: [IO.Interfaces.Temperature](#)

Assembly: libremoteio (in libremoteio.dll) Version: 2.2020.136.3

► Syntax

[C#](#) [VB](#) [F#](#)

[Copy](#)

```
public static double FahrenheitToCelsius(  
    double fahrenheit  
)
```

Parameters

fahrenheit

Type: [SystemDouble](#)

Temperature in degrees Fahrenheit.

Return Value

Type: [Double](#)

Temperature in degrees Celsius.

► See Also

Reference

[Conversions Class](#)

[IO.Interfaces.Temperature Namespace](#)

ConversionsFahrenheitToKelvins Method

Convert degrees Fahrenheit to Kelvins.

Namespace: [IO.Interfaces.Temperature](#)

Assembly: libremoteio (in libremoteio.dll) Version: 2.2020.136.3

◀ Syntax

[C#](#) [VB](#) [F#](#)

[Copy](#)

```
public static double FahrenheitToKelvins(  
    double fahrenheit  
)
```

Parameters

fahrenheit

Type: [SystemDouble](#)

Temperature in degrees Fahrenheit.

Return Value

Type: [Double](#)

Temperature in Kelvins.

◀ See Also

Reference

[Conversions Class](#)

[IO.Interfaces.Temperature Namespace](#)

ConversionsKelvinsToCelsius Method

Convert Kelvins to degrees Celsius.

Namespace: [IO.Interfaces.Temperature](#)

Assembly: libremoteio (in libremoteio.dll) Version: 2.2020.136.3

◀ Syntax

[C#](#) [VB](#) [F#](#)

[Copy](#)

```
public static double KelvinsToCelsius(  
    double kelvins  
)
```

Parameters

kelvins

Type: [SystemDouble](#)

Temperature in Kelvins.

Return Value

Type: [Double](#)

Temperature in degrees Celsius.

◀ See Also

Reference

[Conversions Class](#)

[IO.Interfaces.Temperature Namespace](#)

ConversionsKelvinsToFahrenheit Method

Convert Kelvins to degrees Fahrenheit.

Namespace: [IO.Interfaces.Temperature](#)

Assembly: libremoteio (in libremoteio.dll) Version: 2.2020.136.3

◀ Syntax

[C#](#) [VB](#) [F#](#)

[Copy](#)

```
public static double KelvinsToFahrenheit(  
    double kelvins  
)
```

Parameters

kelvins

Type: [SystemDouble](#)

Temperature in Kelvns.

Return Value

Type: [Double](#)

Temperature in degrees Fahrenheit.

◀ See Also

Reference

[Conversions Class](#)

[IO.Interfaces.Temperature Namespace](#)

Sensor Interface

Abstract interface for temperature sensors.

Namespace: [IO.Interfaces.Temperature](#)

Assembly: libremoteio (in libremoteio.dll) Version: 2.2020.136.3

▪ Syntax

C# VB F#

[Copy](#)

```
public interface Sensor
```

The [Sensor](#) type exposes the following members.

▪ Properties

	Name	Description
	Celsius	Read-only property returning the temperature in degrees Celsius.
	Fahrenheit	Read-only property returning the temperature in degrees Fahrenheit.
	Kelvins	Read-only property returning the temperature in Kelvins.

[Top](#)

▪ See Also

[Reference](#)

IO.Interfaces.Temperature Namespace

Sensor Properties

The [Sensor](#) type exposes the following members.

▪ Properties

	Name	Description
	Celsius	Read-only property returning the temperature in degrees Celsius.
	Fahrenheit	Read-only property returning the temperature in degrees Fahrenheit.
	Kelvins	Read-only property returning the temperature in Kelvins.

[Top](#)

▪ See Also

Reference

[Sensor Interface](#)

[IO.Interfaces.Temperature Namespace](#)

SensorCelsius Property

Read-only property returning the temperature in degrees Celsius.

Namespace: [IO.Interfaces.Temperature](#)

Assembly: libremoteio (in libremoteio.dll) Version: 2.2020.136.3

↳ Syntax

C# VB F#

[Copy](#)

```
double Celsius { get; }
```

Property Value

Type: [Double](#)

↳ See Also

Reference

[Sensor Interface](#)

[IO.Interfaces.Temperature Namespace](#)

SensorFahrenheit Property

Read-only property returning the temperature in degrees Fahrenheit.

Namespace: [IO.Interfaces.Temperature](#)

Assembly: libremoteio (in libremoteio.dll) Version: 2.2020.136.3

↳ Syntax

C# VB F#

[Copy](#)

```
double Fahrenheit { get; }
```

Property Value

Type: [Double](#)

↳ See Also

Reference

[Sensor Interface](#)

[IO.Interfaces.Temperature Namespace](#)

SensorKelvins Property

Read-only property returning the temperature in Kelvins.

Namespace: [IO.Interfaces.Temperature](#)

Assembly: libremoteio (in libremoteio.dll) Version: 2.2020.136.3

↳ Syntax

C# VB F#

[Copy](#)

```
double Kelvins { get; }
```

Property Value

Type: [Double](#)

↳ See Also

Reference

[Sensor Interface](#)

[IO.Interfaces.Temperature Namespace](#)

IO.Interfaces.Watchdog Namespace

Abstract Interface for Watchdog Timers

► Interfaces

Interface	Description
 Timer	Abstract interface for watchdog timers.

Timer Interface

Abstract interface for watchdog timers.

Namespace: [IO.Interfaces.Watchdog](#)

Assembly: libremoteio (in libremoteio.dll) Version: 2.2020.136.3

▀ Syntax

[C#](#) [VB](#) [F#](#)

[Copy](#)

```
public interface Timer
```

The [Timer](#) type exposes the following members.

▀ Properties

	Name	Description
	timeout	Read/Write watchdog timer period property.

[Top](#)

▀ Methods

	Name	Description
	Kick	Reset the watchdog timer.

[Top](#)

▀ See Also

Reference

IO.Interfaces.Watchdog Namespace

Timer Properties

The [Timer](#) type exposes the following members.

Properties

	Name	Description
	timeout	Read/Write watchdog timer period property.

[Top](#)

See Also

Reference

[Timer Interface](#)

[IO.Interfaces.Watchdog Namespace](#)

Timertimeout Property

Read/Write watchdog timer period property.

Namespace: [IO.Interfaces.Watchdog](#)

Assembly: libremoteio (in libremoteio.dll) Version: 2.2020.136.3

« Syntax

C# VB F#

[Copy](#)

```
int timeout { get; set; }
```

Property Value

Type: [Int32](#)

« See Also

Reference

[Timer Interface](#)

[IO.Interfaces.Watchdog Namespace](#)

Timer Methods

The [Timer](#) type exposes the following members.

▪ Methods

	Name	Description
	Kick	Reset the watchdog timer.

[Top](#)

▪ See Also

Reference

[Timer Interface](#)

[IO.Interfaces.Watchdog Namespace](#)

TimerKick Method

Reset the watchdog timer.

Namespace: [IO.Interfaces.Watchdog](#)

Assembly: libremoteio (in libremoteio.dll) Version: 2.2020.136.3

▪ Syntax

[C#](#) [VB](#) [F#](#)

[Copy](#)

```
void Kick()
```

▪ See Also

Reference

[Timer Interface](#)

[IO.Interfaces.Watchdog Namespace](#)

IO.Objects.Message64.UDP Namespace

64-Byte Message Services over UDP

↳ Classes

Class	Description
 Messenger	64-Byte Message Transport Client Services using UDP (User Datagram Protocol).

Messenger Class

64-Byte Message Transport Client Services using UDP (User Datagram Protocol).

▪ Inheritance Hierarchy

[SystemObject](#) [IO.Objects.Message64.UDPMessenger](#)

Namespace: [IO.Objects.Message64.UDP](#)

Assembly: libremoteio (in libremoteio.dll) Version: 2.2020.136.3

▪ Syntax

[C#](#) [VB](#) [F#](#)

[Copy](#)

```
public class Messenger : Messenger
```

The `Messenger` type exposes the following members.

▪ Constructors

	Name	Description
≡	Messenger	Constructor for a 64-byte Messenger instance using UDP.

[Top](#)

▪ Methods

	Name	Description
≡		

	Receive	Receive a 64-byte response message from a raw HID device.
	Send	Send a 64-byte command message to a raw HID device.
	Transaction	Send a 64-byte command message and receive a 64-byte response message.

[Top](#)

◀ See Also

Reference

[IO.Objects.Message64.UDP Namespace](#)

Messenger Constructor

Constructor for a 64-byte Messenger instance using UDP.

Namespace: [IO.Objects.Message64.UDP](#)

Assembly: libremoteio (in libremoteio.dll) Version: 2.2020.136.3

◀ Syntax

C# VB F#

[Copy](#)

```
public Messenger(  
    string host,  
    int port,  
    int timeoutms = 1000  
)
```

Parameters

host

Type: [SystemString](#)

UDP server domain name or IP address.

port

Type: [SystemInt32](#)

UDP server port number.

timeoutms (**Optional**)

Type: [SystemInt32](#)

Receive timeout in milliseconds. Zero indicates wait forever.

◀ See Also

[Reference](#)

[Messenger Class](#)

IO.Objects.Message64.UDP Namespace

Messenger Methods

The [Messenger](#) type exposes the following members.

▪ Methods

	Name	Description
≡	Receive	Receive a 64-byte response message from a raw HID device.
≡	Send	Send a 64-byte command message to a raw HID device.
≡	Transaction	Send a 64-byte command message and receive a 64-byte response message.

[Top](#)

▪ See Also

Reference

[Messenger Class](#)

[IO.Objects.Message64.UDP Namespace](#)

MessengerReceive Method

Receive a 64-byte response message from a raw HID device.

Namespace: [IO.Objects.Message64.UDP](#)

Assembly: libremoteio (in libremoteio.dll) Version: 2.2020.136.3

◀ Syntax

C# VB F#

[Copy](#)

```
public void Receive(  
    Message resp  
)
```

Parameters

resp

Type: [IO.Interfaces.Message64Message](#)

64-byte response message.

Implements

[MessengerReceive\(Message\)](#)

◀ See Also

Reference

[Messenger Class](#)

[IO.Objects.Message64.UDP Namespace](#)

MessengerSend Method

Send a 64-byte command message to a raw HID device.

Namespace: [IO.Objects.Message64.UDP](#)

Assembly: libremoteio (in libremoteio.dll) Version: 2.2020.136.3

◀ Syntax

C# VB F#

[Copy](#)

```
public void Send(  
    Message cmd  
)
```

Parameters

cmd

Type: [IO.Interfaces.Message64Message](#)

64-byte command message.

Implements

[MessengerSend\(Message\)](#)

◀ See Also

Reference

[Messenger Class](#)

[IO.Objects.Message64.UDP Namespace](#)

MessengerTransaction Method

Send a 64-byte command message and receive a 64-byte response message.

Namespace: [IO.Objects.Message64.UDP](#)

Assembly: libremoteio (in libremoteio.dll) Version: 2.2020.136.3

◀ Syntax

C# VB F#

[Copy](#)

```
public void Transaction(  
    Message cmd,  
    Message resp  
)
```

Parameters

cmd

Type: [IO.Interfaces.Message64Message](#)
64-byte command message.

resp

Type: [IO.Interfaces.Message64Message](#)
64-byte response message.

Implements

[MessengerTransaction\(Message, Message\)](#)

◀ See Also

Reference

[Messenger Class](#)

IO.Objects.Message64.UDP Namespace

IO.Objects.Motor.PWM Namespace

PWM Controlled Motor Services

↳ Classes

Class	Description
 Output	Encapsulates motors controlled by PWM and GPIO outputs.

Output Class

Encapsulates motors controlled by PWM and GPIO outputs.

▪ Inheritance Hierarchy

[SystemObject](#) [IO.Objects.Motor.PWMOutput](#)

Namespace: [IO.Objects.Motor.PWM](#)

Assembly: libremoteio (in libremoteio.dll) Version: 2.2020.136.3

▪ Syntax

C# VB F#

[Copy](#)

```
public class Output : Output
```

The [Output](#) type exposes the following members.

▪ Constructors

	Name	Description
≡	Output(Output, Output, Double)	Constructor for a single motor, using two PWM outputs for clockwise and counterclockwise rotation control.
≡	Output(Pin, Output, Double)	Constructor for a single motor, using one GPIO pin for direction control, and one PWM output for speed control.

[Top](#)

◀ Properties

	Name	Description
	velocity	Write-only property for setting the normalized motor velocity. Allowed values are -1.0 (full speed reverse) to +1.0 (full speed forward).

[Top](#)

◀ See Also

Reference

[IO.Objects.Motor.PWM Namespace](#)

Output Constructor

↳ Overload List

	Name	Description
≡	Output(Output, Output, Double)	Constructor for a single motor, using two PWM outputs for clockwise and counterclockwise rotation control.
≡	Output(Pin, Output, Double)	Constructor for a single motor, using one GPIO pin for direction control, and one PWM output for speed control.

[Top](#)

↳ See Also

Reference

[Output Class](#)

[IO.Objects.Motor.PWM Namespace](#)

Output Constructor (Output, Output, Double)

Constructor for a single motor, using two PWM outputs for clockwise and counterclockwise rotation control.

Namespace: [IO.Objects.Motor.PWM](#)

Assembly: libremoteio (in libremoteio.dll) Version: 2.2020.136.3

↳ Syntax

C# VB F#

Copy

```
public Output(  
    Output clockwise,  
    Output counterclockwise,  
    double velocity = 0  
)
```

Parameters

clockwise

Type: [IO.Interfaces.PWMOutput](#)

PWM output instance (for clockwise rotation control).

counterclockwise

Type: [IO.Interfaces.PWMOutput](#)

PWM output instance (for counterclockwise rotation control).

velocity (Optional)

Type: [SystemDouble](#)

Initial motor velocity.

↳ See Also

Reference

[Output Class](#)

[Output Overload](#)

[IO.Objects.Motor.PWM Namespace](#)

Output Constructor (Pin, Output, Double)

Constructor for a single motor, using one GPIO pin for direction control, and one PWM output for speed control.

Namespace: [IO.Objects.Motor.PWM](#)

Assembly: libremoteio (in libremoteio.dll) Version: 2.2020.136.3

↳ Syntax

C# VB F#

Copy

```
public Output(  
    Pin direction,  
    Output speed,  
    double velocity = 0  
)
```

Parameters

direction

Type: [IO.Interfaces.GPIOPin](#)

GPIO pin instance (for direction control).

speed

Type: [IO.Interfaces.PWMOutput](#)

PWM output instance (for speed control).

velocity (Optional)

Type: [SystemDouble](#)

Initial motor velocity.

↳ See Also

Reference

[Output Class](#)

[Output Overload](#)

[IO.Objects.Motor.PWM Namespace](#)

Output Properties

The [Output](#) type exposes the following members.

▪ Properties

	Name	Description
	velocity	Write-only property for setting the normalized motor velocity. Allowed values are -1.0 (full speed reverse) to +1.0 (full speed forward).

[Top](#)

▪ See Also

[Reference](#)

[Output Class](#)

[IO.Objects.Motor.PWM Namespace](#)

Outputvelocity Property

Write-only property for setting the normalized motor velocity. Allowed values are -1.0 (full speed reverse) to +1.0 (full speed forward).

Namespace: [IO.Objects.Motor.PWM](#)

Assembly: libremoteio (in libremoteio.dll) Version: 2.2020.136.3

↳ Syntax

[C#](#) [VB](#) [F#](#)

[Copy](#)

```
public double velocity { set; }
```

Property Value

Type: [Double](#)

Implements

[Outputvelocity](#)

↳ See Also

[Reference](#)

[Output Class](#)

[IO.Objects.Motor.PWM Namespace](#)

IO.Objects.Motor.Servo Namespace

Servo Controlled Motor (e.g. continuous rotation servo) Services

↳ Classes

Class	Description
 Output	Encapsulates motors controlled by servo outputs (e.g. continuous rotation servos).

Output Class

Encapsulates motors controlled by servo outputs (e.g. continuous rotation servos).

▪ Inheritance Hierarchy

[SystemObject](#) [IO.Objects.Motor.ServoOutput](#)

Namespace: [IO.Objects.Motor.Servo](#)

Assembly: libremoteio (in libremoteio.dll) Version: 2.2020.136.3

▪ Syntax

[C#](#) [VB](#) [F#](#)

[Copy](#)

```
public class Output : Output
```

The [Output](#) type exposes the following members.

▪ Constructors

	Name	Description
	Output	Constructor for a single motor output.

[Top](#)

▪ Properties

	Name	Description
	velocity	Write-only property for setting the

normalized motor velocity. Allowed values are -1.0 (full speed reverse) to +1.0 (full speed forward).

[Top](#)

◀ See Also

Reference

[IO.Objects.Motor.Servo Namespace](#)

Output Constructor

Constructor for a single motor output.

Namespace: [IO.Objects.Motor.Servo](#)

Assembly: libremoteio (in libremoteio.dll) Version: 2.2020.136.3

◀ Syntax

C# VB F#

[Copy](#)

```
public Output(  
    Output servo,  
    double velocity = 0  
)
```

Parameters

servo

Type: [IO.Interfaces.ServoOutput](#)

Servo output instance.

velocity (Optional)

Type: [SystemDouble](#)

Initial motor velocity.

◀ See Also

[Reference](#)

[Output Class](#)

[IO.Objects.Motor.Servo Namespace](#)

Output Properties

The [Output](#) type exposes the following members.

▪ Properties

	Name	Description
	velocity	Write-only property for setting the normalized motor velocity. Allowed values are -1.0 (full speed reverse) to +1.0 (full speed forward).

[Top](#)

▪ See Also

[Reference](#)

[Output Class](#)

[IO.Objects.Motor.Servo Namespace](#)

Outputvelocity Property

Write-only property for setting the normalized motor velocity. Allowed values are -1.0 (full speed reverse) to +1.0 (full speed forward).

Namespace: [IO.Objects.Motor.Servo](#)

Assembly: libremoteio (in libremoteio.dll) Version: 2.2020.136.3

↳ Syntax

[C#](#) [VB](#) [F#](#)

[Copy](#)

```
public double velocity { set; }
```

Property Value

Type: [Double](#)

Implements

[Outputvelocity](#)

↳ See Also

[Reference](#)

[Output Class](#)

[IO.Objects.Motor.Servo Namespace](#)

IO.Objects.Servo.PWM Namespace

PWM Controlled Servo Services

↳ Classes

Class	Description
 Output	Encapsulates servo outputs using PWM outputs.

Output Class

Encapsulates servo outputs using PWM outputs.

▪ Inheritance Hierarchy

[SystemObject](#) [IO.Objects.Servo.PWMOutput](#)

Namespace: [IO.Objects.Servo](#).PWM

Assembly: libremoteio (in libremoteio.dll) Version: 2.2020.136.3

▪ Syntax

C# VB F#

[Copy](#)

```
public class Output : Output
```

The [Output](#) type exposes the following members.

▪ Constructors

	Name	Description
	Output	Constructor for a single servo output.

[Top](#)

▪ Properties

	Name	Description
	position	Write-only property for setting the servo position. Allowed values are -1.0 to +1.0.

[Top](#)

↳ See Also

Reference

[IO.Objects.Servo.PWM Namespace](#)

Output Constructor

Constructor for a single servo output.

Namespace: [IO.Objects.Servo.PWM](#)

Assembly: libremoteio (in libremoteio.dll) Version: 2.2020.136.3

◀ Syntax

C# VB F#

[Copy](#)

```
public Output(  
    Output pwm,  
    int freq = 50,  
    double position = 0  
)
```

Parameters

pwm

Type: [IO.Interfaces.PWMOutput](#)

PWM output instance.

freq (Optional)

Type: [SystemInt32](#)

PWM pulse frequency.

position (Optional)

Type: [SystemDouble](#)

Initial servo position.

◀ See Also

[Reference](#)

[Output Class](#)

IO.Objects.Servo.PWM Namespace

Output Properties

The [Output](#) type exposes the following members.

Properties

	Name	Description
	position	Write-only property for setting the servo position. Allowed values are -1.0 to +1.0.

[Top](#)

See Also

[Reference](#)

[Output Class](#)

[IO.Objects.Servo.PWM Namespace](#)

Outputposition Property

Write-only property for setting the servo position. Allowed values are -1.0 to +1.0.

Namespace: [IO.Objects.Servo.PWM](#)

Assembly: libremoteio (in libremoteio.dll) Version: 2.2020.136.3

↳ Syntax

[C#](#) [VB](#) [F#](#)

[Copy](#)

```
public double position { set; }
```

Property Value

Type: [Double](#)

Implements

[Outputposition](#)

↳ See Also

[Reference](#)

[Output Class](#)

[IO.Objects.Servo.PWM Namespace](#)

IO.Objects.USB.HID Namespace

USB Raw HID (Human Interface Device) Transport Services

◀ Classes

	Class	Description
	Messenger	Encapsulates USB raw HID devices, using the HidSharp library or libsimpleio.

Messenger Class

Encapsulates USB raw HID devices, using the HidSharp library or libsimpleio.

▪ Inheritance Hierarchy

[SystemObject](#) [IO.Objects.USB.HIDMessenger](#)

Namespace: [IO.Objects.USB.HID](#)

Assembly: libremoteio (in libremoteio.dll) Version: 2.2020.136.3

▪ Syntax

[C#](#) [VB](#) [F#](#)

[Copy](#)

```
public class Messenger : Messenger
```

The `Messenger` type exposes the following members.

▪ Constructors

	Name	Description
≡	Messenger	Create a 64-byte messenger object bound to a USB HID device.

[Top](#)

▪ Properties

	Name	Description
--	------	-------------



[Info](#) Information string from the USB HID device.

[Top](#)

◀ Methods

	Name	Description
≡	Receive	Receive a 64-byte message from a USB HID device.
≡	Send	Send a 64-byte message to a USB HID device.
≡	Transaction	Send a 64-byte command message to and receive a 64-byte response message from a USB HID device.

[Top](#)

◀ See Also

Reference

[IO.Objects.USB.HID Namespace](#)

Messenger Constructor

Create a 64-byte messenger object bound to a USB HID device.

Namespace: [IO.Objects.USB.HID](#)

Assembly: libremoteio (in libremoteio.dll) Version: 2.2020.136.3

◀ Syntax

C# VB F#

[Copy](#)

```
public Messenger(  
    int vid = 5840,  
    int pid = 2810,  
    string serialnumber = null,  
    int timeoutms = 1000  
)
```

Parameters

vid (Optional)

Type: [SystemInt32](#)

Vendor ID

pid (Optional)

Type: [SystemInt32](#)

Product ID

serialnumber (Optional)

Type: [SystemString](#)

Serial number

timeoutms (Optional)

Type: [SystemInt32](#)

Time in milliseconds to wait for read and write operations to

complete. Zero means wait forever.

◀ See Also

Reference

[Messenger Class](#)

[IO.Objects.USB.HID Namespace](#)

Messenger Properties

The [Messenger](#) type exposes the following members.

▪ Properties

	Name	Description
	Info	Information string from the USB HID device.

[Top](#)

▪ See Also

Reference

[Messenger Class](#)

[IO.Objects.USB.HID Namespace](#)

MessengerInfo Property

Information string from the USB HID device.

Namespace: [IO.Objects.USB.HID](#)

Assembly: libremoteio (in libremoteio.dll) Version: 2.2020.136.3

◀ Syntax

C# VB F#

[Copy](#)

```
public string Info { get; }
```

Property Value

Type: [String](#)

◀ See Also

Reference

[Messenger Class](#)

[IO.Objects.USB.HID Namespace](#)

Messenger Methods

The [Messenger](#) type exposes the following members.

▪ Methods

	Name	Description
≡	Receive	Receive a 64-byte message from a USB HID device.
≡	Send	Send a 64-byte message to a USB HID device.
≡	Transaction	Send a 64-byte command message to and receive a 64-byte response message from a USB HID device.

[Top](#)

▪ See Also

Reference

[Messenger Class](#)

[IO.Objects.USB.HID Namespace](#)

MessengerReceive Method

Receive a 64-byte message from a USB HID device.

Namespace: [IO.Objects.USB.HID](#)

Assembly: libremoteio (in libremoteio.dll) Version: 2.2020.136.3

◀ Syntax

C# VB F#

[Copy](#)

```
public void Receive(  
    Message resp  
)
```

Parameters

resp

Type: [IO.Interfaces.Message64Message](#)

Message received.

Implements

[MessengerReceive\(Message\)](#)

◀ See Also

Reference

[Messenger Class](#)

[IO.Objects.USB.HID Namespace](#)

MessengerSend Method

Send a 64-byte message to a USB HID device.

Namespace: [IO.Objects.USB.HID](#)

Assembly: libremoteio (in libremoteio.dll) Version: 2.2020.136.3

◀ Syntax

C# VB F#

[Copy](#)

```
public void Send(  
    Message cmd  
)
```

Parameters

cmd

Type: [IO.Interfaces.Message64Message](#)

Message to be sent.

Implements

[MessengerSend\(Message\)](#)

◀ See Also

Reference

[Messenger Class](#)

[IO.Objects.USB.HID Namespace](#)

MessengerTransaction Method

Send a 64-byte command message to and receive a 64-byte response message from a USB HID device.

Namespace: [IO.Objects.USB.HID](#)

Assembly: libremoteio (in libremoteio.dll) Version: 2.2020.136.3

◀ Syntax

C# VB F#

[Copy](#)

```
public void Transaction(  
    Message cmd,  
    Message resp  
)
```

Parameters

cmd

Type: [IO.Interfaces.Message64Message](#)

Message to be sent.

resp

Type: [IO.Interfaces.Message64Message](#)

Message received.

Implements

[MessengerTransaction\(Message, Message\)](#)

◀ See Also

Reference

[Messenger Class](#)

IO.Objects.USB.HID Namespace

IO.Remote Namespace

Remote I/O Device Framework, for sending commands and receiving response to/from [Remote I/O Protocol](#) devices.

▪ Classes

	Class	Description
	ADC	Encapsulates remote A/D inputs.
	DAC	Encapsulates remote D/A outputs.
	Device	Encapsulates a remote I/O device.
	GPIO	Encapsulates remote GPIO pins.
	I2C	Encapsulates remote I ² C buses.
	PWM	Encapsulates remote PWM outputs.
	SPI	Encapsulates remote SPI slave devices.

▪ Enumerations

	Enumeration	Description
	MessageTypes	Remote I/O protocol message types
	PeripheralTypes	Types of remote peripherals

ADC Class

Encapsulates remote A/D inputs.

▪ Inheritance Hierarchy

[SystemObject](#) [IO.RemoteADC](#)

Namespace: [IO.Remote](#)

Assembly: libremoteio (in libremoteio.dll) Version: 2.2020.136.3

▪ Syntax

C# VB F#

[Copy](#)

```
public class ADC : Sample
```

The [ADC](#) type exposes the following members.

▪ Constructors

	Name	Description
	ADC	Create a remote A/D input.

[Top](#)

▪ Properties

	Name	Description
	resolution	Read-only property returning the number of bits of resolution.



[sample](#)

Read-only property returning an integer analog input sample.

[Top](#)

◀ See Also

Reference

[IO.Remote Namespace](#)

ADC Constructor

Create a remote A/D input.

Namespace: [IO.Remote](#)

Assembly: libremoteio (in libremoteio.dll) Version: 2.2020.136.3

▪ Syntax

C# VB F#

[Copy](#)

```
public ADC(  
    Device dev,  
    int num  
)
```

Parameters

dev

Type: [IO.RemoteDevice](#)

Remote I/O device object.

num

Type: [SystemInt32](#)

A/D input number: 0 to 127.

▪ Remarks

Use [Device.ADC_Create\(\)](#) instead of this constructor.

▪ See Also

Reference

[ADC Class](#)

ADC Properties

The [ADC](#) type exposes the following members.

Properties

	Name	Description
	resolution	Read-only property returning the number of bits of resolution.
	sample	Read-only property returning an integer analog input sample.

[Top](#)

See Also

Reference

[ADC Class](#)

[IO.Remote Namespace](#)

ADCresolution Property

Read-only property returning the number of bits of resolution.

Namespace: [IO.Remote](#)

Assembly: libremoteio (in libremoteio.dll) Version: 2.2020.136.3

« Syntax

C# VB F#

[Copy](#)

```
public int resolution { get; }
```

Property Value

Type: [Int32](#)

Implements

[Sampleresolution](#)

« See Also

[Reference](#)

[ADC Class](#)

[IO.Remote Namespace](#)

ADCsample Property

Read-only property returning an integer analog input sample.

Namespace: [IO.Remote](#)

Assembly: libremoteio (in libremoteio.dll) Version: 2.2020.136.3

▪ Syntax

C# VB F#

[Copy](#)

```
public int sample { get; }
```

Property Value

Type: [Int32](#)

Implements

[Samplesample](#)

▪ See Also

[Reference](#)

[ADC Class](#)

[IO.Remote Namespace](#)

DAC Class

Encapsulates remote D/A outputs.

▪ Inheritance Hierarchy

[SystemObject](#) [IO.RemoteDAC](#)

Namespace: [IO.Remote](#)

Assembly: libremoteio (in libremoteio.dll) Version: 2.2020.136.3

▪ Syntax

C# VB F#

[Copy](#)

```
public class DAC : Sample
```

The [DAC](#) type exposes the following members.

▪ Constructors

	Name	Description
	DAC	Create a remote D/A output.

[Top](#)

▪ Properties

	Name	Description
	resolution	Read-only property returning the number of bits of resolution.



sample

Write-only property for writing an integer analog sample to a DAC output.

[Top](#)

◀ See Also

Reference

[IO.Remote Namespace](#)

DAC Constructor

Create a remote D/A output.

Namespace: [IO.Remote](#)

Assembly: libremoteio (in libremoteio.dll) Version: 2.2020.136.3

◀ Syntax

C# VB F#

[Copy](#)

```
public DAC(  
    Device dev,  
    int num,  
    int sample = 0  
)
```

Parameters

dev

Type: [IO.RemoteDevice](#)

Remote I/O device object.

num

Type: [SystemInt32](#)

D/A output number: 0 to 127.

sample (Optional)

Type: [SystemInt32](#)

Initial DAC output sample.

◀ Remarks

Use [Device.DAC_Create\(\)](#) instead of this constructor.

See Also

[Reference](#)

[DAC Class](#)

[IO.Remote Namespace](#)

DAC Properties

The [DAC](#) type exposes the following members.

Properties

	Name	Description
	resolution	Read-only property returning the number of bits of resolution.
	sample	Write-only property for writing an integer analog sample to a DAC output.

[Top](#)

See Also

Reference

[DAC Class](#)

[IO.Remote Namespace](#)

DACresolution Property

Read-only property returning the number of bits of resolution.

Namespace: [IO.Remote](#)

Assembly: libremoteio (in libremoteio.dll) Version: 2.2020.136.3

▪ Syntax

C# VB F#

[Copy](#)

```
public int resolution { get; }
```

Property Value

Type: [Int32](#)

Implements

[Sampleresolution](#)

▪ See Also

[Reference](#)

[DAC Class](#)

[IO.Remote Namespace](#)

DACsample Property

Write-only property for writing an integer analog sample to a DAC output.

Namespace: [IO.Remote](#)

Assembly: libremoteio (in libremoteio.dll) Version: 2.2020.136.3

↳ Syntax

[C#](#) [VB](#) [F#](#)

[Copy](#)

```
public int sample { set; }
```

Property Value

Type: [Int32](#)

Implements

[Samplesample](#)

↳ See Also

Reference

[DAC Class](#)

[IO.Remote Namespace](#)

Device Class

Encasulates a remote I/O device.

Encasulates a remote I/O device.

▪ Inheritance Hierarchy

[SystemObject](#) [IO.RemoteDevice](#)

Namespace: [IO.Remote](#)

Assembly: libremoteio (in libremoteio.dll) Version: 2.2020.136.3

▪ Syntax

[C#](#) [VB](#) [F#](#)

[Copy](#)

```
public class Device
```

The [Device](#) type exposes the following members.

▪ Constructors

Name	Description
 Device	Create a Remote I/O device object for a Munts Technologies USB raw HID (VID=0x16D0, PID=0x0AFA) device Remote I/O Server.
 Device(Messenger)	Create a Remote I/O device object.

[Top](#)

◀ Properties

	Name	Description
	Capabilities	Capability string from the Remote I/O device.
	Version	Version string from the Remote I/O device.

[Top](#)

◀ Methods

	Name	Description
	ADC_Available	Query available A/D inputs.
	ADC_Create	Create a remote A/D input.
	DAC_Available	Query available D/A outputs.
	DAC_Create	Create a remote D/A output.
	Dispatcher	Command dispatcher.
	GPIO_Available	Query available GPIO pins.
	GPIO_Create	Create a remote GPIO pin object.
	I2C_Available	Query available I ² C buses.
	I2C_Create	Create a remote I ² C bus controller.

≡	PWM_Available	Query available PWM outputs.
≡	PWM_Create	Create a remote PWM output.
≡	SPI_Available	Query available SPI slave devices.
≡	SPI_Create	Create a remote SPI slave device.

[Top](#)

◀ Fields

	Name	Description
♦ S	MAX_CHANNELS	Maximum number of channels each subsystem can support.
♦ S	Unavailable	Designator for an unavailable channel.

[Top](#)

◀ See Also

Reference

[IO.Remote Namespace](#)

Device Constructor

▪ Overload List

	Name	Description
	Device	Create a Remote I/O device object for a Munts Technologies USB raw HID (VID=0x16D0, PID=0x0AFA) device Remote I/O Server.
	Device(Messenger)	Create a Remote I/O device object.

[Top](#)

▪ See Also

Reference

[Device Class](#)

[IO.Remote Namespace](#)

Device Constructor

Create a Remote I/O device object for a Munts Technologies USB raw HID (VID=0x16D0, PID=0x0AFA) device Remote I/O Server.

Namespace: [IO.Remote](#)

Assembly: libremoteio (in libremoteio.dll) Version: 2.2020.136.3

◀ Syntax

[C#](#) [VB](#) [F#](#)

[Copy](#)

```
public Device()
```

◀ See Also

Reference

[Device Class](#)

[Device Overload](#)

[IO.Remote Namespace](#)

Device Constructor (Messenger)

Create a Remote I/O device object.

Namespace: [IO.Remote](#)

Assembly: libremoteio (in libremoteio.dll) Version: 2.2020.136.3

▀ Syntax

C# VB F#

[Copy](#)

```
public Device(  
    Messenger m  
)
```

Parameters

m

Type: [IO.Interfaces.Message64Messenger](#)
Message transport object

▀ See Also

Reference

[Device Class](#)

[Device Overload](#)

[IO.Remote Namespace](#)

Device Properties

The [Device](#) type exposes the following members.

Properties

	Name	Description
	Capabilities	Capability string from the Remote I/O device.
	Version	Version string from the Remote I/O device.

[Top](#)

See Also

Reference

[Device Class](#)

[IO.Remote Namespace](#)

DeviceCapabilities Property

Capability string from the Remote I/O device.

Namespace: [IO.Remote](#)

Assembly: libremoteio (in libremoteio.dll) Version: 2.2020.136.3

« Syntax

C# VB F#

[Copy](#)

```
public string Capabilities { get; }
```

Property Value

Type: [String](#)

« See Also

Reference

[Device Class](#)

[IO.Remote Namespace](#)

DeviceVersion Property

Version string from the Remote I/O device.

Namespace: [IO.Remote](#)

Assembly: libremoteio (in libremoteio.dll) Version: 2.2020.136.3

« Syntax

C# VB F#

[Copy](#)

```
public string Version { get; }
```

Property Value

Type: [String](#)

« See Also

Reference

[Device Class](#)

[IO.Remote Namespace](#)

Device Methods

The [Device](#) type exposes the following members.

▪ Methods

	Name	Description
≡■	ADC_Available	Query available A/D inputs.
≡■	ADC_Create	Create a remote A/D input.
≡■	DAC_Available	Query available D/A outputs.
≡■	DAC_Create	Create a remote D/A output.
≡■	Dispatcher	Command dispatcher.
≡■	GPIO_Available	Query available GPIO pins.
≡■	GPIO_Create	Create a remote GPIO pin object.
≡■	I2C_Available	Query available I ² C buses.
≡■	I2C_Create	Create a remote I ² C bus controller.
≡■	PWM_Available	Query available PWM outputs.
≡■	PWM_Create	Create a remote PWM output.
≡■	SPI_Available	Query available SPI slave devices.

[SPI_Create](#)

Create a remote SPI slave device.

[Top](#)

◀ See Also

Reference

[Device Class](#)

[IO.Remote Namespace](#)

DeviceADC_Available Method

Query available A/D inputs.

Namespace: [IO.Remote](#)

Assembly: libremoteio (in libremoteio.dll) Version: 2.2020.136.3

▪ Syntax

C# VB F#

[Copy](#)

```
public List<int> ADC_Available()
```

Return Value

Type: [ListInt32](#)

List of available A/D input numbers.

▪ See Also

Reference

[Device Class](#)

[IO.Remote Namespace](#)

DeviceADC_Create Method

Create a remote A/D input.

Namespace: [IO.Remote](#)

Assembly: libremoteio (in libremoteio.dll) Version: 2.2020.136.3

↳ Syntax

C# VB F#

[Copy](#)

```
public Sample ADC_Create(  
    int num  
)
```

Parameters

num

Type: [SystemInt32](#)

A/D input number: 0 to 127.

Return Value

Type: [Sample](#)

A/D input object.

↳ See Also

Reference

[Device Class](#)

[IO.Remote Namespace](#)

DeviceDAC_Available Method

Query available D/A outputs.

Namespace: [IO.Remote](#)

Assembly: libremoteio (in libremoteio.dll) Version: 2.2020.136.3

▪ Syntax

C# VB F#

[Copy](#)

```
public List<int> DAC_Available()
```

Return Value

Type: [ListInt32](#)

List of available D/A output numbers.

▪ See Also

Reference

[Device Class](#)

[IO.Remote Namespace](#)

DeviceDAC_Create Method

Create a remote D/A output.

Namespace: [IO.Remote](#)

Assembly: libremoteio (in libremoteio.dll) Version: 2.2020.136.3

↳ Syntax

C# VB F#

[Copy](#)

```
public Sample DAC_Create(  
    int num,  
    int sample = 0  
)
```

Parameters

num

Type: [SystemInt32](#)

D/A output number: 0 to 127.

sample (Optional)

Type: [SystemInt32](#)

Initial DAC output sample.

Return Value

Type: [Sample](#)

D/A output object.

↳ See Also

Reference

[Device Class](#)

IO.Remote Namespace

DeviceDispatcher Method

Command dispatcher.

Namespace: [IO.Remote](#)

Assembly: libremoteio (in libremoteio.dll) Version: 2.2020.136.3

◀ Syntax

C# VB F#

[Copy](#)

```
public void Dispatcher(  
    Message cmd,  
    Message resp  
)
```

Parameters

cmd

Type: [IO.Interfaces.Message64Message](#)

Command to be sent.

resp

Type: [IO.Interfaces.Message64Message](#)

Response to be received.

◀ See Also

[Reference](#)

[Device Class](#)

[IO.Remote Namespace](#)

DeviceGPIO_Available Method

Query available GPIO pins.

Namespace: [IO.Remote](#)

Assembly: libremoteio (in libremoteio.dll) Version: 2.2020.136.3

▪ Syntax

C# VB F#

[Copy](#)

```
public List<int> GPIO_Available()
```

Return Value

Type: [ListInt32](#)

List of available GPIO pin numbers.

▪ See Also

Reference

[Device Class](#)

[IO.Remote Namespace](#)

DeviceGPIO_Create Method

Create a remote GPIO pin object.

Namespace: [IO.Remote](#)

Assembly: libremoteio (in libremoteio.dll) Version: 2.2020.136.3

◀ Syntax

C# VB F#

[Copy](#)

```
public Pin GPIO_Create(  
    int num,  
    Direction dir,  
    bool state = false  
)
```

Parameters

num

Type: [SystemInt32](#)

GPIO pin number: 0 to 127.

dir

Type: [IO.Interfaces.GPIODirection](#)

GPIO pin data direction: Input or Output.

state (Optional)

Type: [SystemBoolean](#)

Initial GPIO output state.

Return Value

Type: [Pin](#)

GPIO pin object.

↳ See Also

[Reference](#)

[Device Class](#)

[IO.Remote Namespace](#)

DeviceI2C_Available Method

Query available I²C buses.

Namespace: [IO.Remote](#)

Assembly: libremoteio (in libremoteio.dll) Version: 2.2020.136.3

▪ Syntax

C# VB F#

[Copy](#)

```
public List<int> I2C_Available()
```

Return Value

Type: [ListInt32](#)

List of available I²C bus numbers.

▪ See Also

Reference

[Device Class](#)

[IO.Remote Namespace](#)

DeviceI2C_Create Method

Create a remote I²C bus controller.

Namespace: [IO.Remote](#)

Assembly: libremoteio (in libremoteio.dll) Version: 2.2020.136.3

▪ Syntax

C# VB F#

[Copy](#)

```
public Bus I2C_Create(  
    int num,  
    int speed = 100000  
)
```

Parameters

num

Type: [SystemInt32](#)

I²C bus number: 0 to 127.

speed (Optional)

Type: [SystemInt32](#)

I²C bus clock frequency in Hz

Return Value

Type: [Bus](#)

I²C bus controller object.

▪ See Also

[Reference](#)

[Device Class](#)

IO.Remote Namespace

DevicePWM_Available Method

Query available PWM outputs.

Namespace: [IO.Remote](#)

Assembly: libremoteio (in libremoteio.dll) Version: 2.2020.136.3

▪ Syntax

C# VB F#

[Copy](#)

```
public List<int> PWM_Available()
```

Return Value

Type: [ListInt32](#)

List of available PWM output numbers.

▪ See Also

Reference

[Device Class](#)

[IO.Remote Namespace](#)

DevicePWM_Create Method

Create a remote PWM output.

Namespace: [IO.Remote](#)

Assembly: libremoteio (in libremoteio.dll) Version: 2.2020.136.3

◀ Syntax

C# VB F#

[Copy](#)

```
public Output PWM_Create(  
    int num,  
    int freq,  
    double duty = 0  
)
```

Parameters

num

Type: [SystemInt32](#)

PWM output number: 0 to 127.

freq

Type: [SystemInt32](#)

PWM pulse frequency in Hz.

duty (Optional)

Type: [SystemDouble](#)

Initial PWM output duty cycle.

Return Value

Type: [Output](#)

PWM output object.

↳ See Also

[Reference](#)

[Device Class](#)

[IO.Remote Namespace](#)

DeviceSPI_Available Method

Query available SPI slave devices.

Namespace: [IO.Remote](#)

Assembly: libremoteio (in libremoteio.dll) Version: 2.2020.136.3

▪ Syntax

C# VB F#

[Copy](#)

```
public List<int> SPI_Available()
```

Return Value

Type: [ListInt32](#)

List of available SPI slave device numbers.

▪ See Also

Reference

[Device Class](#)

[IO.Remote Namespace](#)

DeviceSPI_Create Method

Create a remote SPI slave device.

Namespace: [IO.Remote](#)

Assembly: libremoteio (in libremoteio.dll) Version: 2.2020.136.3

◀ Syntax

C# VB F#

[Copy](#)

```
public Device SPI_Create(  
    int num,  
    int mode,  
    int wordsize,  
    int speed  
)
```

Parameters

num

Type: [SystemInt32](#)

SPI slave device number: 0 to 127.

mode

Type: [SystemInt32](#)

SPI transfer mode: 0 to 3.

wordsize

Type: [SystemInt32](#)

SPI transfer word size: 8, 16, or 32.

speed

Type: [SystemInt32](#)

SPI transfer speed in bits per second.

Return Value

Type: [Device](#)

SPI slave device object.

Remarks

The actual SPI transfer rate will be the highest realizable rate that does not exceed the value specified in [speed](#).

See Also

Reference

[Device Class](#)

[IO.Remote Namespace](#)

Device Fields

The [Device](#) type exposes the following members.

↳ Fields

	Name	Description
• 	MAX_CHANNELS	Maximum number of channels each subsystem can support.
• 	Unavailable	Designator for an unavailable channel.

[Top](#)

↳ See Also

Reference

[Device Class](#)

[IO.Remote Namespace](#)

DeviceMAX_CHANNELS Field

Maximum number of channels each subsystem can support.

Namespace: [IO.Remote](#)

Assembly: libremoteio (in libremoteio.dll) Version: 2.2020.136.3

« Syntax

C# VB F#

[Copy](#)

```
public const int MAX_CHANNELS = 128
```

Field Value

Type: [Int32](#)

« See Also

Reference

[Device Class](#)

[IO.Remote Namespace](#)

DeviceUnavailable Field

Designator for an unavailable channel.

Namespace: [IO.Remote](#)

Assembly: libremoteio (in libremoteio.dll) Version: 2.2020.136.3

◀ Syntax

C# VB F#

[Copy](#)

```
public const int Unavailable = -1
```

Field Value

Type: [Int32](#)

◀ See Also

Reference

[Device Class](#)

[IO.Remote Namespace](#)

GPIO Class

Encapsulates remote GPIO pins.

▪ Inheritance Hierarchy

[SystemObject](#) [IO.RemoteGPIO](#)

Namespace: [IO.Remote](#)

Assembly: libremoteio (in libremoteio.dll) Version: 2.2020.136.3

▪ Syntax

C# VB F#

[Copy](#)

```
public class GPIO : Pin
```

The [GPIO](#) type exposes the following members.

▪ Constructors

	Name	Description
	GPIO	Create a remote GPIO pin.

[Top](#)

▪ Properties

	Name	Description
	state	Read/Write GPIO state property.

[Top](#)

◀ See Also

Reference

[IO.Remote Namespace](#)

GPIO Constructor

Create a remote GPIO pin.

Namespace: [IO.Remote](#)

Assembly: libremoteio (in libremoteio.dll) Version: 2.2020.136.3

◀ Syntax

C# VB F#

[Copy](#)

```
public GPIO(  
    Device dev,  
    int num,  
    Direction dir,  
    bool state = false  
)
```

Parameters

dev

Type: [IO.RemoteDevice](#)

Remote I/O device object.

num

Type: [SystemInt32](#)

GPIO pin number: 0 to 127.

dir

Type: [IO.Interfaces.GPIODirection](#)

GPIO pin data direction: Input or Output.

state (Optional)

Type: [SystemBoolean](#)

Initial GPIO output state.

▪ Remarks

Use [Device.GPIO_Create\(\)](#) instead of this constructor.

▪ See Also

Reference

[GPIO Class](#)

[IO.Remote Namespace](#)

GPIO Properties

The [GPIO](#) type exposes the following members.

Properties

	Name	Description
	state	Read/Write GPIO state property.

[Top](#)

See Also

[Reference](#)

[GPIO Class](#)

[IO.Remote Namespace](#)

GPIOstate Property

Read/Write GPIO state property.

Namespace: [IO.Remote](#)

Assembly: libremoteio (in libremoteio.dll) Version: 2.2020.136.3

↳ Syntax

C# VB F#

[Copy](#)

```
public bool state { get; set; }
```

Property Value

Type: [Boolean](#)

Implements

[Pinstate](#)

↳ See Also

[Reference](#)

[GPIO Class](#)

[IO.Remote Namespace](#)

I2C Class

Encapsulates remote I²C buses.

▪ Inheritance Hierarchy

[SystemObject](#) [IO.Remote.I2C](#)

Namespace: [IO.Remote](#)

Assembly: libremoteio (in libremoteio.dll) Version: 2.2020.136.3

▪ Syntax

[C#](#) [VB](#) [F#](#)

[Copy](#)

```
public class I2C : Bus
```

The [I2C](#) type exposes the following members.

▪ Constructors

	Name	Description
≡	I2C	Create a remote I ² C bus controller.

[Top](#)

▪ Methods

	Name	Description
≡	Read	Read bytes from an I ² C slave device.
≡		

Transaction Write and read bytes to and from an I²C slave device.



Write

Write bytes to an I²C slave device.

[Top](#)

◀ See Also

Reference

[IO.Remote Namespace](#)

I2C Constructor

Create a remote I²C bus controller.

Namespace: [IO.Remote](#)

Assembly: libremoteio (in libremoteio.dll) Version: 2.2020.136.3

▪ Syntax

C# VB F#

[Copy](#)

```
public I2C(  
    Device dev,  
    int num,  
    int speed = 100000  
)
```

Parameters

dev

Type: [IO.RemoteDevice](#)

Remote I/O device object.

num

Type: [SystemInt32](#)

I²C bus number: 0 to 127.

speed (Optional)

Type: [SystemInt32](#)

I²C bus clock frequency in Hz

▪ Remarks

Use [Device.I2C_Create\(\)](#) instead of this constructor.

↳ See Also

Reference

[I2C Class](#)

[IO.Remote Namespace](#)

I²C Methods

The [I²C](#) type exposes the following members.

▪ Methods

	Name	Description
	Read	Read bytes from an I ² C slave device.
	Transaction	Write and read bytes to and from an I ² C slave device.
	Write	Write bytes to an I ² C slave device.

[Top](#)

▪ See Also

[Reference](#)

[I²C Class](#)

[IO.Remote Namespace](#)

I2CRead Method

Read bytes from an I²C slave device.

Namespace: [IO.Remote](#)

Assembly: libremoteio (in libremoteio.dll) Version: 2.2020.136.3

↳ Syntax

C# VB F#

[Copy](#)

```
public void Read(  
    int slaveaddr,  
    byte[] resp,  
    int resplen  
)
```

Parameters

slaveaddr

Type: [SystemInt32](#)

I²C slave address.

resp

Type: [SystemByte](#)

Response buffer.

resplen

Type: [SystemInt32](#)

Number of bytes to read.

Implements

[BusRead\(Int32, Byte, Int32\)](#)

↳ See Also

Reference

[I2C Class](#)

[IO.Remote Namespace](#)

I2CTransaction Method

Write and read bytes to and from an I²C slave device.

Namespace: `IO.Remote`

Assembly: libremoteio (in libremoteio.dll) Version: 2.2020.136.3

↳ Syntax

`C# VB F#`

[Copy](#)

```
public void Transaction(  
    int slaveaddr,  
    byte[] cmd,  
    int cmdlen,  
    byte[] resp,  
    int resplen,  
    int delayus  
)
```

Parameters

slaveaddr

Type: `SystemInt32`

I²C slave address.

cmd

Type: `SystemByte`

Command buffer.

cmdlen

Type: `SystemInt32`

Number of bytes to write.

resp

Type: [SystemByte](#)

Response buffer.

resplen

Type: [SystemInt32](#)

Number of bytes to read.

delayus

Type: [SystemInt32](#)

Delay in microseconds between the I²C write and read cycles.

Allowed values are 0 to 65535 microseconds.

Implements

[BusTransaction\(Int32, Byte, Int32, Byte, Int32, Int32\)](#)

See Also

Reference

[I2C Class](#)

[IO.Remote Namespace](#)

I2CWrite Method

Write bytes to an I²C slave device.

Namespace: [IO.Remote](#)

Assembly: libremoteio (in libremoteio.dll) Version: 2.2020.136.3

↳ Syntax

C# VB F#

[Copy](#)

```
public void Write(  
    int slaveaddr,  
    byte[] cmd,  
    int cmdlen  
)
```

Parameters

slaveaddr

Type: [SystemInt32](#)

I²C slave address.

cmd

Type: [SystemByte](#)

Command buffer.

cmdlen

Type: [SystemInt32](#)

Number of bytes to write.

Implements

[BusWrite\(Int32, Byte, Int32\)](#)

↳ See Also

Reference

[I2C Class](#)

[IO.Remote Namespace](#)

MessageTypes Enumeration

Remote I/O protocol message types

Namespace: [IO.Remote](#)

Assembly: libremoteio (in libremoteio.dll) Version: 2.2020.136.3

▪ Syntax

C# VB F#

[Copy](#)

```
public enum MessageTypes
```

▪ Members

Member name	Value	Description
LOOPBACK_REQUEST	0	Loopback request
LOOPBACK_RESPONSE	1	Loopback response
VERSION_REQUEST	2	Version string request
VERSION_RESPONSE	3	Version string response
CAPABILITY_REQUEST	4	Capability

			string request
CAPABILITY_RESPONSE	5	Capability string response	
GPIO_PRESENT_REQUEST	6	GPIO pins available request	
GPIO_PRESENT_RESPONSE	7	GPIO pins available response	
GPIO_CONFIGURE_REQUEST	8	GPIO pins configure request	
GPIO_CONFIGURE_RESPONSE	9	GPIO pins configure response	
GPIO_READ_REQUEST	10	GPIO pins read request	
GPIO_READ_RESPONSE	11	GPIO pins read response	
GPIO_WRITE_REQUEST	12	GPIO pins write request	

GPIO_WRITE_RESPONSE	13	GPIO pins write response
I2C_PRESENT_REQUEST	14	I ² C buses available request
I2C_PRESENT_RESPONSE	15	I ² C buses available response
I2C_CONFIGURE_REQUEST	16	I ² C bus configure request
I2C_CONFIGURE_RESPONSE	17	I ² C bus configure response
I2C_TRANSACTION_REQUEST	18	I ² C bus transaction request
I2C_TRANSACTION_RESPONSE	19	I ² C bus transaction response
SPI_PRESENT_REQUEST	20	SPI slave devices available request
SPI_PRESENT_RESPONSE	21	SPI slave devices

			available response
SPI_CONFIGURE_REQUEST	22	SPI slave device configure request	
SPI_CONFIGURE_RESPONSE	23	SPI slave device configure response	
SPI_TRANSACTION_REQUEST	24	SPI bus transaction request	
SPI_TRANSACTION_RESPONSE	25	SPI bus transaction response	
ADC_PRESENT_REQUEST	26	ADC inputs available request	
ADC_PRESENT_RESPONSE	27	ADC inputs available response	
ADC_CONFIGURE_REQUEST	28	ADC input configure request	
ADC_CONFIGURE_RESPONSE	29	ADC input configure	

			response
ADC_READ_REQUEST	30		ADC input read request
ADC_READ_RESPONSE	31		ADC input read response
DAC_PRESENT_REQUEST	32		DAC outputs available request
DAC_PRESENT_RESPONSE	33		DAC outputs available response
DAC_CONFIGURE_REQUEST	34		DAC input configure request
DAC_CONFIGURE_RESPONSE	35		DAC input configure response
DAC_WRITE_REQUEST	36		DAC output write request
DAC_WRITE_RESPONSE	37		DAC output write response

PWM_PRESENT_REQUEST	38	PWM outputs available request
PWM_PRESENT_RESPONSE	39	PWM outputs available response
PWM_CONFIGURE_REQUEST	40	PWM input configure request
PWM_CONFIGURE_RESPONSE	41	PWM input configure response
PWM_WRITE_REQUEST	42	PWM output write request
PWM_WRITE_RESPONSE	43	PWM output write response

See Also

Reference

[IO.Remote Namespace](#)

PeripheralTypes Enumeration

Types of remote peripherals

Namespace: [IO.Remote](#)

Assembly: libremoteio (in libremoteio.dll) Version: 2.2020.136.3

▪ Syntax

C# VB F#

[Copy](#)

```
public enum PeripheralTypes
```

▪ Members

Member name	Value	Description
ADC	0	A/D inputs
DAC	1	D/A outputs
GPIO	2	GPIO pins
I2C	3	I ² C bus controllers
PWM	4	SPI slave devices
SPI	5	PWM outputs

▪ See Also

[Reference](#)

[IO.Remote Namespace](#)

PWM Class

Encapsulates remote PWM outputs.

▪ Inheritance Hierarchy

[SystemObject](#) [IO.RemotePWM](#)

Namespace: [IO.Remote](#)

Assembly: libremoteio (in libremoteio.dll) Version: 2.2020.136.3

▪ Syntax

C# VB F#

[Copy](#)

```
public class PWM : Output
```

The [PWM](#) type exposes the following members.

▪ Constructors

	Name	Description
	PWM	Create a remote PWM output.

[Top](#)

▪ Properties

	Name	Description
	dutycycle	Write-only property for setting the PWM output duty cycle.

[Top](#)

◀ See Also

Reference

[IO.Remote Namespace](#)

PWM Constructor

Create a remote PWM output.

Namespace: [IO.Remote](#)

Assembly: libremoteio (in libremoteio.dll) Version: 2.2020.136.3

◀ Syntax

C# VB F#

[Copy](#)

```
public PWM(  
    Device dev,  
    int num,  
    int freq,  
    double duty = 0  
)
```

Parameters

dev

Type: [IO.RemoteDevice](#)

Remote I/O device object.

num

Type: [SystemInt32](#)

PWM output number: 0 to 127.

freq

Type: [SystemInt32](#)

PWM pulse frequency in Hz.

duty (Optional)

Type: [SystemDouble](#)

Initial PWM output duty cycle.

▪ Remarks

Use `Device.PWM_Create()` instead of this constructor.

▪ See Also

Reference

[PWM Class](#)

[IO.Remote Namespace](#)

PWM Properties

The [PWM](#) type exposes the following members.

▪ Properties

	Name	Description
	dutycycle	Write-only property for setting the PWM output duty cycle.

[Top](#)

▪ See Also

[Reference](#)

[PWM Class](#)

[IO.Remote Namespace](#)

PWMdutycycle Property

Write-only property for setting the PWM output duty cycle.

Namespace: [IO.Remote](#)

Assembly: libremoteio (in libremoteio.dll) Version: 2.2020.136.3

▪ Syntax

C# VB F#

[Copy](#)

```
public double dutycycle { set; }
```

Property Value

Type: [Double](#)

Implements

[Outputdutycycle](#)

▪ See Also

[Reference](#)

[PWM Class](#)

[IO.Remote Namespace](#)

SPI Class

Encapsulates remote SPI slave devices.

▪ Inheritance Hierarchy

[SystemObject](#) [IO.RemoteSPI](#)

Namespace: [IO.Remote](#)

Assembly: libremoteio (in libremoteio.dll) Version: 2.2020.136.3

▪ Syntax

C# VB F#

[Copy](#)

```
public class SPI : Device
```

The [SPI](#) type exposes the following members.

▪ Constructors

	Name	Description
≡	SPI	Create a remote SPI slave device.

[Top](#)

▪ Methods

	Name	Description
≡	Read	Read bytes from an SPI slave device.
≡		

Transaction Write and read bytes to and from an SPI slave device.



Write Write bytes to an SPI slave device.

[Top](#)

◀ See Also

Reference

[IO.Remote Namespace](#)

SPI Constructor

Create a remote SPI slave device.

Namespace: [IO.Remote](#)

Assembly: libremoteio (in libremoteio.dll) Version: 2.2020.136.3

◀ Syntax

C# VB F#

[Copy](#)

```
public SPI(  
    Device dev,  
    int num,  
    int mode,  
    int wordsize,  
    int speed  
)
```

Parameters

dev

Type: [IO.RemoteDevice](#)

Remote I/O device object.

num

Type: [SystemInt32](#)

SPI slave device number: 0 to 127.

mode

Type: [SystemInt32](#)

SPI transfer mode: 0 to 3.

wordsize

Type: [SystemInt32](#)

SPI transfer word size: 8, 16, or 32.

speed

Type: [SystemInt32](#)

SPI transfer speed in bits per second.

▪ Remarks

Use [Device.SPI_Create\(\)](#) instead of this constructor.

The actual SPI transfer rate will be the highest realizable rate that does not exceed the value specified in *speed*.

▪ See Also

Reference

[SPI Class](#)

[IO.Remote Namespace](#)

SPI Methods

The [SPI](#) type exposes the following members.

↳ Methods

	Name	Description
	Read	Read bytes from an SPI slave device.
	Transaction	Write and read bytes to and from an SPI slave device.
	Write	Write bytes to an SPI slave device.

[Top](#)

↳ See Also

Reference

[SPI Class](#)

[IO.Remote Namespace](#)

SPIRead Method

Read bytes from an SPI slave device.

Namespace: [IO.Remote](#)

Assembly: libremoteio (in libremoteio.dll) Version: 2.2020.136.3

↳ Syntax

C# VB F#

[Copy](#)

```
public void Read(  
    byte[] resp,  
    int resplen  
)
```

Parameters

resp

Type: [SystemByte](#)

Response buffer.

resplen

Type: [SystemInt32](#)

Number of bytes to read: 1 to 60.

Implements

[DeviceRead\(Byte, Int32\)](#)

↳ See Also

Reference

[SPI Class](#)

[IO.Remote Namespace](#)

SPITransaction Method

Write and read bytes to and from an SPI slave device.

Namespace: [IO.Remote](#)

Assembly: libremoteio (in libremoteio.dll) Version: 2.2020.136.3

◀ Syntax

C# VB F#

[Copy](#)

```
public void Transaction(  
    byte[] cmd,  
    int cmdLen,  
    byte[] resp,  
    int resplen,  
    int delayus = 0  
)
```

Parameters

cmd

Type: [SystemByte](#)

Command buffer.

cmdlen

Type: [SystemInt32](#)

Number of bytes to write: 0 to 57.

resp

Type: [SystemByte](#)

Response buffer.

resplen

Type: [SystemInt32](#)

Number of bytes to read: 0 to 60.

delayus (Optional)

Type: [SystemInt32](#)

Delay in microseconds between write and read operations: 0 to 65535.

Implements

[DeviceTransaction\(Byte, Int32, Byte, Int32, Int32\)](#)

See Also

Reference

[SPI Class](#)

[IO.Remote Namespace](#)

SPIWrite Method

Write bytes to an SPI slave device.

Namespace: [IO.Remote](#)

Assembly: libremoteio (in libremoteio.dll) Version: 2.2020.136.3

↳ Syntax

C# VB F#

[Copy](#)

```
public void Write(  
    byte[] cmd,  
    int cmdLen  
)
```

Parameters

cmd

Type: [SystemByte](#)

Command buffer.

cmdlen

Type: [SystemInt32](#)

Number of bytes to write: 1 to 57.

Implements

[DeviceWrite\(Byte, Int32\)](#)

↳ See Also

Reference

[SPI Class](#)

[IO.Remote Namespace](#)

IO.Remote.mikroBUS Namespace

Mikroelektronika mikroBUS (<https://www.mikroe.com/mikrobus>)
Remote I/O protocol Server and Socket Services

▪ Classes

	Class	Description
	Shield	Encapsulates mikroBUS shields on Remote I/O Protocol servers providing mikroBUS sockets).
	Socket	Encapsulates mikroBUS sockets.

▪ Enumerations

	Enumeration	Description
	ShieldKinds	Supported mikroBUS shields.

Shield Class

Encapsulates mikroBUS shields on Remote I/O Protocol servers providing [mikroBUS](#) sockets).

▪ Inheritance Hierarchy

[SystemObject](#) [IO.Remote.mikroBUSShield](#)

Namespace: [IO.Remote.mikroBUS](#)

Assembly: libremoteio (in libremoteio.dll) Version: 2.2020.136.3

▪ Syntax

[C#](#) [VB](#) [F#](#)

[Copy](#)

```
public static class Shield
```

The [Shield](#) type exposes the following members.

▪ Properties

	Name	Description
 S	kind	Returns the kind of mikroBUS shield that is installed on the Remote I/O Protocol server, as obtained from the SHIELDNAME environment variable.

[Top](#)

▪ Fields

	Name	Description
 	I2CBus	Shared I ² C bus that is common to all sockets on this shield.

[Top](#)

◀ See Also

Reference

[IO.Remote.mikroBUS Namespace](#)

Shield Properties

The [Shield](#) type exposes the following members.

▪ Properties

	Name	Description
 S	kind	Returns the kind of mikroBUS shield that is installed on the Remote I/O Protocol server, as obtained from the <code>SHIELDNAME</code> environment variable.

[Top](#)

▪ See Also

[Reference](#)

[Shield Class](#)

[IO.Remote.mikroBUS Namespace](#)

Shieldkind Property

Returns the kind of mikroBUS shield that is installed on the Remote I/O Protocol server, as obtained from the `SHIELDNAME` environment variable.

Namespace: [IO.Remote.mikroBUS](#)

Assembly: libremoteio (in libremoteio.dll) Version: 2.2020.136.3

↳ Syntax

C# VB F#

[Copy](#)

```
public static ShieldKinds kind { get; }
```

Property Value

Type: [ShieldKinds](#)

↳ See Also

[Reference](#)

[Shield Class](#)

[IO.Remote.mikroBUS Namespace](#)

Shield Fields

The [Shield](#) type exposes the following members.

↳ Fields

	Name	Description
 	I2CBus	Shared I ² C bus that is common to all sockets on this shield.

[Top](#)

↳ See Also

[Reference](#)

[Shield Class](#)

[IO.Remote.mikroBUS Namespace](#)

ShieldI2CBus Field

Shared I²C bus that is common to all sockets on this shield.

Namespace: [IO.Remote.mikroBUS](#)

Assembly: libremoteio (in libremoteio.dll) Version: 2.2020.136.3

↳ Syntax

[C#](#) [VB](#) [F#](#)

[Copy](#)

```
public static Bus I2CBus
```

Field Value

Type: [Bus](#)

↳ See Also

[Reference](#)

[Shield Class](#)

[IO.Remote.mikroBUS Namespace](#)

ShieldKinds Enumeration

Supported mikroBUS shields.

Namespace: `IO.Remote.mikroBUS`

Assembly: `libremoteio` (in `libremoteio.dll`) Version: 2.2020.136.3

▪ Syntax

C# VB F#

[Copy](#)

```
public enum Kinds
```

▪ Members

Member name	Value	Description
<code>Clicker</code>	0	Mikroelektronika STM32F4 Clicker with MUNTS-0011 Remote I/O Server firmware, with one mikroBUS socket.
<code>PiClick1</code>	1	Raspberry Pi with Mikroelektronika Pi Click Shield MIKROE-1512/1513 for 26-pin expansion header, with one mikroBUS socket (Obsolete.)

PiClick2	2	Raspberry Pi with Mikroelektronika Pi 2 Click Shield MIKROE-1879 for 40-pin expansion header, with two mikroBUS sockets.
PiClick3	3	Mikroelektronika Pi 3 Click Shield MIKROE-2756 for 40-pin expansion header, with selectable on-board A/D converter and two mikroBUS sockets.
PocketBeagle	4	PocketBeagle with female headers on top, with two mikroBUS sockets.
Unknown	2147483647	No known mikroBUS shield installed.

See Also

Reference

[IO.Remote.mikroBUS Namespace](#)

Socket Class

Encapsulates mikroBUS sockets.

▪ Inheritance Hierarchy

[SystemObject](#) [IO.Remote.mikroBUSSocket](#)

Namespace: [IO.Remote.mikroBUS](#)

Assembly: libremoteio (in libremoteio.dll) Version: 2.2020.136.3

▪ Syntax

C# VB F#

[Copy](#)

```
public class Socket
```

The [Socket](#) type exposes the following members.

▪ Constructors

	Name	Description
	Socket	Constructor for a single mikroBUS socket.

[Top](#)

▪ Properties

	Name	Description
	AIN	Returns the ADC input designator for AN.

	AN	Returns the GPIO pin designator for AN.
	CS	Returns the GPIO pin designator for CS.
	I2CBus	Returns the I ² C bus designator for this socket.
	INT	Returns the GPIO pin designator for INT.
	MISO	Returns the GPIO pin designator for MISO.
	MOSI	Returns the GPIO pin designator for MOSI.
	PWM	Returns the GPIO pin designator for PWM.
	PWMOut	Returns the PWM output designator for PWM.
	RST	Returns the GPIO pin designator for RST.
	RX	Returns the GPIO pin designator for RX.
	SCK	Returns the GPIO pin designator for SCK.
	SCL	Returns the GPIO pin designator for SCL.
	SDA	Returns the GPIO pin designator for SDA.
	SPIDev	Returns the SPI device designator for

this socket.



[TX](#)

Returns the GPIO pin designator for TX.

[Top](#)

◀ See Also

Reference

[IO.Remote.mikroBUS Namespace](#)

Socket Constructor

Constructor for a single mikroBUS socket.

Namespace: [IO.Remote.mikroBUS](#)

Assembly: libremoteio (in libremoteio.dll) Version: 2.2020.136.3

↳ Syntax

C# VB F#

[Copy](#)

```
public Socket(  
    int num,  
    ShieldKinds shield = ShieldKinds.Unknown  
)
```

Parameters

num

Type: [SystemInt32](#)

Socket number.

shield (Optional)

Type: [IO.Remote.mikroBUSShieldKinds](#)

mikroBUS shield kind. Zero indicates automatic detection using the [Shield.kind](#) property.

↳ See Also

[Reference](#)

[Socket Class](#)

[IO.Remote.mikroBUS Namespace](#)

Socket Properties

The [Socket](#) type exposes the following members.

Properties

	Name	Description
	AIN	Returns the ADC input designator for AN.
	AN	Returns the GPIO pin designator for AN.
	CS	Returns the GPIO pin designator for CS.
	I2CBus	Returns the I ² C bus designator for this socket.
	INT	Returns the GPIO pin designator for INT.
	MISO	Returns the GPIO pin designator for MISO.
	MOSI	Returns the GPIO pin designator for MOSI.
	PWM	Returns the GPIO pin designator for PWM.
	PWMOut	Returns the PWM output designator for PWM.

	RST	Returns the GPIO pin designator for RST.
	RX	Returns the GPIO pin designator for RX.
	SCK	Returns the GPIO pin designator for SCK.
	SCL	Returns the GPIO pin designator for SCL.
	SDA	Returns the GPIO pin designator for SDA.
	SPIDev	Returns the SPI device designator for this socket.
	TX	Returns the GPIO pin designator for TX.

[Top](#)

◀ See Also

[Reference](#)

[Socket Class](#)

[IO.Remote.mikroBUS Namespace](#)

SocketAIN Property

Returns the ADC input designator for AN.

Namespace: [IO.Remote.mikroBUS](#)

Assembly: libremoteio (in libremoteio.dll) Version: 2.2020.136.3

« Syntax

C# VB F#

[Copy](#)

```
public int AIN { get; }
```

Property Value

Type: [Int32](#)

« See Also

Reference

[Socket Class](#)

[IO.Remote.mikroBUS Namespace](#)

SocketAN Property

Returns the GPIO pin designator for AN.

Namespace: [IO.Remote.mikroBUS](#)

Assembly: libremoteio (in libremoteio.dll) Version: 2.2020.136.3

« Syntax

C# VB F#

[Copy](#)

```
public int AN { get; }
```

Property Value

Type: [Int32](#)

« See Also

Reference

[Socket Class](#)

[IO.Remote.mikroBUS Namespace](#)

SocketCS Property

Returns the GPIO pin designator for CS.

Namespace: [IO.Remote.mikroBUS](#)

Assembly: libremoteio (in libremoteio.dll) Version: 2.2020.136.3

« Syntax

C# VB F#

[Copy](#)

```
public int CS { get; }
```

Property Value

Type: [Int32](#)

« See Also

Reference

[Socket Class](#)

[IO.Remote.mikroBUS Namespace](#)

SocketI2CBus Property

Returns the I²C bus designator for this socket.

Namespace: [IO.Remote.mikroBUS](#)

Assembly: libremoteio (in libremoteio.dll) Version: 2.2020.136.3

« Syntax

C# VB F#

[Copy](#)

```
public int I2CBus { get; }
```

Property Value

Type: [Int32](#)

« See Also

Reference

[Socket Class](#)

[IO.Remote.mikroBUS Namespace](#)

SocketINT Property

Returns the GPIO pin designator for INT.

Namespace: [IO.Remote.mikroBUS](#)

Assembly: libremoteio (in libremoteio.dll) Version: 2.2020.136.3

« Syntax

C# VB F#

[Copy](#)

```
public int INT { get; }
```

Property Value

Type: [Int32](#)

« See Also

Reference

[Socket Class](#)

[IO.Remote.mikroBUS Namespace](#)

SocketMISO Property

Returns the GPIO pin designator for MISO.

Namespace: [IO.Remote.mikroBUS](#)

Assembly: libremoteio (in libremoteio.dll) Version: 2.2020.136.3

« Syntax

C# VB F#

[Copy](#)

```
public int MISO { get; }
```

Property Value

Type: [Int32](#)

« See Also

Reference

[Socket Class](#)

[IO.Remote.mikroBUS Namespace](#)

SocketMOSI Property

Returns the GPIO pin designator for MOSI.

Namespace: [IO.Remote.mikroBUS](#)

Assembly: libremoteio (in libremoteio.dll) Version: 2.2020.136.3

↳ Syntax

C# VB F#

[Copy](#)

```
public int MOSI { get; }
```

Property Value

Type: [Int32](#)

↳ See Also

Reference

[Socket Class](#)

[IO.Remote.mikroBUS Namespace](#)

SocketPWM Property

Returns the GPIO pin designator for PWM.

Namespace: [IO.Remote.mikroBUS](#)

Assembly: libremoteio (in libremoteio.dll) Version: 2.2020.136.3

↳ Syntax

C# VB F#

[Copy](#)

```
public int PWM { get; }
```

Property Value

Type: [Int32](#)

↳ See Also

Reference

[Socket Class](#)

[IO.Remote.mikroBUS Namespace](#)

SocketPWMOut Property

Returns the PWM output designator for PWM.

Namespace: [IO.Remote.mikroBUS](#)

Assembly: libremoteio (in libremoteio.dll) Version: 2.2020.136.3

↳ Syntax

C# VB F#

[Copy](#)

```
public int PWMOut { get; }
```

Property Value

Type: [Int32](#)

↳ See Also

Reference

[Socket Class](#)

[IO.Remote.mikroBUS Namespace](#)

SocketRST Property

Returns the GPIO pin designator for RST.

Namespace: [IO.Remote.mikroBUS](#)

Assembly: libremoteio (in libremoteio.dll) Version: 2.2020.136.3

↳ Syntax

C# VB F#

[Copy](#)

```
public int RST { get; }
```

Property Value

Type: [Int32](#)

↳ See Also

Reference

[Socket Class](#)

[IO.Remote.mikroBUS Namespace](#)

SocketRX Property

Returns the GPIO pin designator for RX.

Namespace: [IO.Remote.mikroBUS](#)

Assembly: libremoteio (in libremoteio.dll) Version: 2.2020.136.3

↳ Syntax

C# VB F#

[Copy](#)

```
public int RX { get; }
```

Property Value

Type: [Int32](#)

↳ See Also

Reference

[Socket Class](#)

[IO.Remote.mikroBUS Namespace](#)

SocketSCK Property

Returns the GPIO pin designator for SCK.

Namespace: [IO.Remote.mikroBUS](#)

Assembly: libremoteio (in libremoteio.dll) Version: 2.2020.136.3

↳ Syntax

C# VB F#

[Copy](#)

```
public int SCK { get; }
```

Property Value

Type: [Int32](#)

↳ See Also

Reference

[Socket Class](#)

[IO.Remote.mikroBUS Namespace](#)

SocketSCL Property

Returns the GPIO pin designator for SCL.

Namespace: [IO.Remote.mikroBUS](#)

Assembly: libremoteio (in libremoteio.dll) Version: 2.2020.136.3

« Syntax

C# VB F#

[Copy](#)

```
public int SCL { get; }
```

Property Value

Type: [Int32](#)

« See Also

Reference

[Socket Class](#)

[IO.Remote.mikroBUS Namespace](#)

SocketSDA Property

Returns the GPIO pin designator for SDA.

Namespace: [IO.Remote.mikroBUS](#)

Assembly: libremoteio (in libremoteio.dll) Version: 2.2020.136.3

« Syntax

C# VB F#

[Copy](#)

```
public int SDA { get; }
```

Property Value

Type: [Int32](#)

« See Also

Reference

[Socket Class](#)

[IO.Remote.mikroBUS Namespace](#)

SocketSPIDev Property

Returns the SPI device designator for this socket.

Namespace: [IO.Remote.mikroBUS](#)

Assembly: libremoteio (in libremoteio.dll) Version: 2.2020.136.3

↳ Syntax

C# VB F#

[Copy](#)

```
public int SPIDev { get; }
```

Property Value

Type: [Int32](#)

↳ See Also

Reference

[Socket Class](#)

[IO.Remote.mikroBUS Namespace](#)

SocketTX Property

Returns the GPIO pin designator for TX.

Namespace: [IO.Remote.mikroBUS](#)

Assembly: libremoteio (in libremoteio.dll) Version: 2.2020.136.3

« Syntax

C# VB F#

[Copy](#)

```
public int TX { get; }
```

Property Value

Type: [Int32](#)

« See Also

Reference

[Socket Class](#)

[IO.Remote.mikroBUS Namespace](#)

Index

IO.Devices.AD5593R	1
Device Class	2
Device Constructor	5
Device Properties	6
ADC_Reference Property	8
DAC_Reference Property	9
GPIO_Inputs Property	10
GPIO_Outputs Property	11
Device Methods	12
ADC_Create Method	13
ConfigureChannel Method	14
DAC_Create Method	15
GPIO_Create Method	17
Read_ADC Method	19
Write_DAC Method	20
Device Fields	21
ADC_Resolution Field	22
DAC_Resolution Field	23
MaxChannel Field	24
MinChannel Field	25
PinMode Enumeration	26
ReferenceMode Enumeration	28
IO.Devices.AD5593R.ADC	30
Sample Class	31
Sample Constructor	33
Sample Properties	34
resolution Property	35

sample Property	36
IO.Devices.AD5593R.DAC	37
Sample Class	38
Sample Constructor	40
Sample Properties	42
resolution Property	43
sample Property	44
IO.Devices.AD5593R.GPIO	45
Pin Class	46
Pin Constructor	48
Pin Properties	50
state Property	51
IO.Devices.ADC121C021	52
Sample Class	53
Sample Constructor	55
Sample Properties	56
resolution Property	57
sample Property	58
IO.Devices.ClickBoards.RemoteIO.ADAC	59
Board Class	60
Board Constructor	62
Board Properties	64
device Property	65
Board Methods	66
ADC Method	67
DAC Method	68
GPIO Method	70
Reset Method	72
Board Fields	73

DefaultAddress Field	74
IO.Devices.ClickBoards.RemoteIO.Expand	75
Board Class	76
Board Constructor	78
Board Properties	79
device Property	80
Board Methods	81
GPIO Method	82
Reset Method	84
IO.Devices.ClickBoards.RemoteIO.Expand2	85
Board Class	86
Board Constructor	88
Board Properties	90
device Property	91
Board Methods	92
GPIO Method	93
Reset Method	95
Board Fields	96
DefaultAddress Field	97
IO.Devices.ClickBoards.RemoteIO.PWM	98
Board Class	99
Board Constructor	101
Board Properties	103
dev Property	104
Board Methods	105
GPIO Method	106
PWM Method	108
Servo Method	110
Board Fields	112

DefaultAddress Field	113
IO.Devices.ClickBoards.RemoteIO.SevenSegment	114
Board Class	115
Board Constructor	118
Board Properties	120
blanking Property	122
brightness Property	123
leftdp Property	124
radix Property	125
rightdp Property	126
state Property	127
Board Methods	128
Clear Method	129
Reset Method	130
Board.Base Enumeration	131
Board.ZeroBlanking Enumeration	132
IO.Devices.Grove.ADC	133
Device Class	134
Device Constructor	136
Device Properties	137
voltage Property	138
IO.Devices.Grove.Temperature	139
Device Class	140
Device Constructor	142
Device Properties	143
Celsius Property	144
Fahrenheit Property	145
Kelvins Property	146
IO.Devices.Grove.Temperature_Humidity	147

Device Class	148
Device Constructor	150
Device Properties	151
IO.Devices.HDC1080	153
Device Class	154
Device Constructor	157
Device Properties	158
Celsius Property	160
DeviceID Property	161
Fahrenheit Property	162
Humidity Property	163
Kelvins Property	164
ManufacturerID Property	165
Device Methods	166
Read Method	167
Write Method	168
Device Fields	169
RegConfiguration Field	171
RegDeviceID Field	172
RegHumidity Field	173
RegManufacturerID Field	174
RegSerialNumberFirst Field	175
RegSerialNumberLast Field	176
RegSerialNumberMid Field	177
RegTemperature Field	178
IO.Devices.MCP23017	179
Device Class	180
Device Constructor	183
Device Properties	184

Direction Property	186
DirectionA Property	187
DirectionB Property	188
Polarity Property	189
PolarityA Property	190
PolarityB Property	191
Port Property	192
PortA Property	193
PortB Property	194
Pullups Property	195
PullupsA Property	196
PullupsB Property	197
Device Methods	198
GPIO_Create Method	199
Device Fields	201
MaxChannel Field	202
MinChannel Field	203
IO.Devices.MCP23017.GPIO	204
Pin Class	205
Pin Constructor	207
Pin Properties	209
state Property	210
IO.Devices.MCP23S17	211
Device Class	212
Device Constructor	216
Device Properties	217
Direction Property	219
DirectionA Property	220
DirectionB Property	221

Polarity Property	222
PolarityA Property	223
PolarityB Property	224
Port Property	225
PortA Property	226
PortB Property	227
Pullups Property	228
PullupsA Property	229
PullupsB Property	230
Device Methods	231
GPIO_Create Method	232
Device Fields	234
MaxChannel Field	235
MinChannel Field	236
SPI_Frequency Field	237
SPI_Mode Field	238
SPI_WordSize Field	239
IO.Devices.MCP23S17.GPIO	240
Pin Class	241
Pin Constructor	243
Pin Properties	245
state Property	246
IO.Devices.PCA8574	247
Device Class	248
Device Constructor	250
Device Properties	252
Latch Property	253
Device Methods	254
Read Method	255

Write Method	256
Device Fields	257
MAX_PINS Field	258
IO.Devices.PCA8574.GPIO	259
Pin Class	260
Pin Constructor	262
Pin Properties	264
state Property	265
IO.Devices.PCA9534	266
Device Class	267
Device Constructor	270
Device Properties	272
Config Property	273
Latch Property	274
Device Methods	275
Read Method	276
Read Method	277
Read Method (Byte)	278
Write Method	279
Write Method (Byte)	280
Write Method (Byte, Byte)	281
Device Fields	282
AllInputs Field	284
AllNormal Field	285
AllOff Field	286
AllOutputs Field	287
ConfigurationReg Field	288
InputPolarityReg Field	289
InputPortReg Field	290

MAX_PINS Field	291
OutputPortReg Field	292
IO.Devices.PCA9534.GPIO	293
Pin Class	294
Pin Constructor	296
Pin Properties	298
state Property	299
IO.Devices.PCA9685	300
Device Class	301
Device Constructor	304
Device Properties	306
Frequency Property	307
Device Methods	308
ReadChannel Method	309
WriteChannel Method	310
Device Fields	311
INTERNAL_CLOCK Field	312
MAX_CHANNEL Field	313
MAX_CLOCK Field	314
MIN_CHANNEL Field	315
MIN_CLOCK Field	316
IO.Devices.PCA9685.GPIO	317
Pin Class	318
Pin Constructor	320
Pin Properties	322
state Property	323
IO.Devices.PCA9685.PWM	324
Output Class	325
Output Constructor	327

Output Properties	329
dutycycle Property	330
IO.Devices.PCA9685.Servo	331
Output Class	332
Output Constructor	334
Output Properties	336
position Property	337
IO.Devices.Pmod.HYGRO	338
Device Class	339
Device Constructor	342
Device Properties	343
Device Methods	345
IO.Devices.SN74HC595	346
Device Class	347
Device Constructor	350
Device Properties	352
Length Property	353
state Property	354
Device Methods	355
ClrBit Method	356
ReadBit Method	357
SetBit Method	359
Device Fields	360
SPI_MaxFreq Field	361
SPI_Mode Field	362
IO.Devices.SN74HC595.GPIO	363
Pin Class	364
Pin Constructor	366
Pin Properties	368

state Property	369
IO.Devices.TH02	370
Device Class	371
Device Constructor	373
Device Properties	374
Celsius Property	375
DeviceID Property	376
Fahrenheit Property	377
Humidity Property	378
Kelvins Property	379
IO.Devices.Thermistor	380
NTC_B Class	381
NTC_B Constructor	383
NTC_B Methods	385
Kelvins Method	386
IO.Devices.USB.Munts	387
HID Class	388
HID Fields	390
Product Field	391
Vendor Field	392
Serial Class	393
Serial Fields	395
Product Field	396
Vendor Field	397
IO.Interfaces.ADC	398
Input Class	399
Input Constructor	401
Input Properties	403
voltage Property	404

Sample Interface	405
Sample Properties	406
resolution Property	407
sample Property	408
Voltage Interface	409
Voltage Properties	410
voltage Property	411
IO.Interfaces.DAC	412
Output Class	413
Output Constructor	415
Output Properties	417
voltage Property	418
Sample Interface	419
Sample Properties	421
resolution Property	422
sample Property	423
Voltage Interface	424
Voltage Properties	425
voltage Property	426
IO.Interfaces.GPIO	427
Direction Enumeration	428
Pin Interface	429
Pin Properties	430
state Property	431
IO.Interfaces.Humidity	432
Sensor Interface	433
Sensor Properties	434
Humidity Property	435
IO.Interfaces.I2C	436

Bus Interface	437
Bus Methods	439
Read Method	440
Transaction Method	442
Write Method	444
Device Class	446
Device Constructor	448
Device Methods	449
Read Method	450
Transaction Method	451
Write Method	453
Speeds Class	454
Speeds Fields	456
FastMode Field	457
FastModePlus Field	458
StandardMode Field	459
IO.Interfaces.Message64	460
Message Class	461
Message Constructor	463
Message Constructor	464
Message Constructor (Byte)	465
Message Fields	466
payload Field	467
Size Field	468
Messenger Interface	469
Messenger Methods	471
Receive Method	472
Send Method	473
Transaction Method	474

IO.Interfaces.Motor	475
Output Interface	476
Output Properties	477
velocity Property	478
Velocities Class	479
Velocities Fields	481
Maximum Field	482
Minimum Field	483
Stop Field	484
IO.Interfaces.PWM	485
DutyCycles Class	486
DutyCycles Fields	488
Maximum Field	489
Minimum Field	490
Output Interface	491
Output Properties	492
dutycycle Property	493
IO.Interfaces.Servo	494
Output Interface	495
Output Properties	496
position Property	497
Positions Class	498
Positions Fields	500
Maximum Field	501
Minimum Field	502
Neutral Field	503
IO.Interfaces.SPI	504
Device Interface	505
Device Methods	507

Read Method	508
Transaction Method	509
Write Method	511
IO.Interfaces.Temperature	512
Conversions Class	513
Conversions Constructor	515
Conversions Methods	516
CelsiusToFahrenheit Method	518
CelsiusToKelvins Method	519
FahrenheitToCelsius Method	520
FahrenheitToKelvins Method	521
KelvinsToCelsius Method	522
KelvinsToFahrenheit Method	523
Sensor Interface	524
Sensor Properties	526
Celsius Property	527
Fahrenheit Property	528
Kelvins Property	529
IO.Interfaces.Watchdog	530
Timer Interface	531
Timer Properties	533
timeout Property	534
Timer Methods	535
Kick Method	536
IO.Objects.Message64.UDP	537
Messenger Class	538
Messenger Constructor	540
Messenger Methods	542
Receive Method	543

Send Method	544
Transaction Method	545
IO.Objects.Motor.PWM	547
Output Class	548
Output Constructor	550
Output Constructor (Output, Output, Double)	551
Output Constructor (Pin, Output, Double)	553
Output Properties	555
velocity Property	556
IO.Objects.Motor.Servo	557
Output Class	558
Output Constructor	560
Output Properties	561
velocity Property	562
IO.Objects.Servo.PWM	563
Output Class	564
Output Constructor	566
Output Properties	568
position Property	569
IO.Objects.USB.HID	570
Messenger Class	571
Messenger Constructor	573
Messenger Properties	575
Info Property	576
Messenger Methods	577
Receive Method	578
Send Method	579
Transaction Method	580
IO.Remote	582

ADC Class	584
ADC Constructor	586
ADC Properties	588
resolution Property	589
sample Property	590
DAC Class	591
DAC Constructor	593
DAC Properties	595
resolution Property	596
sample Property	597
Device Class	598
Device Constructor	601
Device Constructor	602
Device Constructor (Messenger)	603
Device Properties	604
Capabilities Property	605
Version Property	606
Device Methods	607
ADC_Available Method	609
ADC_Create Method	610
DAC_Available Method	611
DAC_Create Method	612
Dispatcher Method	614
GPIO_Available Method	615
GPIO_Create Method	616
I2C_Available Method	618
I2C_Create Method	619
PWM_Available Method	621
PWM_Create Method	622

SPI_Available Method	624
SPI_Create Method	625
Device Fields	627
MAX_CHANNELS Field	628
Unavailable Field	629
GPIO Class	630
GPIO Constructor	632
GPIO Properties	634
state Property	635
I2C Class	636
I2C Constructor	638
I2C Methods	640
Read Method	641
Transaction Method	643
Write Method	645
MessageTypes Enumeration	647
PeripheralTypes Enumeration	653
PWM Class	655
PWM Constructor	657
PWM Properties	659
dutycycle Property	660
SPI Class	661
SPI Constructor	663
SPI Methods	665
Read Method	666
Transaction Method	668
Write Method	670
IO.Remote.mikroBUS	672
Shield Class	673

Shield Properties	675
kind Property	676
Shield Fields	677
I2CBus Field	678
Shield.Kinds Enumeration	679
Socket Class	681
Socket Constructor	684
Socket Properties	685
AIN Property	687
AN Property	688
CS Property	689
I2CBus Property	690
INT Property	691
MISO Property	692
MOSI Property	693
PWM Property	694
PWMOut Property	695
RST Property	696
RX Property	697
SCK Property	698
SCL Property	699
SDA Property	700
SPIDev Property	701
TX Property	702