

IO.Bindings Namespace

Wrapper classes for C shared libraries.

↳ Classes

| | Class | Description |
|---|-----------------------------|---|
|  | libsimpleio | Wrapper class for the Linux Simple I/O Library libsimpleio.so |

libsimpleio Class

Wrapper class for the Linux Simple I/O Library [libsimpleio.so](#)

▪ Inheritance Hierarchy

[SystemObject](#) [IO.Bindingslibsimpleio](#)

Namespace: [IO.Bindings](#)

Assembly: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

▪ Syntax

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

[Copy](#)

```
public static class libsimpleio
```

The `libsimpleio` type exposes the following members.

▪ Methods

| | Name | Description |
|--|------------------------------|--|
| | ADC_close | Close a Linux IIO A/D converter input device. |
| | ADC_get_name | Get the subsystem name for the specified Linux IIO A/D converter device. |
| | ADC_open | Open a Linux IIO A/D converter input device. |

| | | |
|-----|-------------------------------------|--|
| | ADC_read | Read a Linux IIO A/D converter input device. |
| ≡ S | DAC_close | Close a Linux IIO D/A converter output device. |
| ≡ S | DAC_get_name | Get the subsystem name for the specified Linux IIO D/A converter device. |
| ≡ S | DAC_open | Open a Linux IIO D/A converter output device. |
| ≡ S | DAC_write | Write to a Linux IIO D/A converter output device. |
| ≡ S | EVENT_close | Close an <code>epoll</code> event dispatcher. |
| ≡ S | EVENT_modify_fd | Modify a file registration. |
| ≡ S | EVENT_open | Open an <code>epoll</code> event dispatcher. |
| ≡ S | EVENT_register_fd | Register a file descriptor with an <code>epoll</code> event dispatcher. |
| ≡ S | EVENT_unregister_fd | Unregister a file from an <code>epoll</code> dispatcher. |
| ≡ S | EVENT_wait | Wait for events from an <code>epoll</code> dispatcher. |
| ≡ S | GPIO_chip_info | Get GPIO chip information. |

| | | |
|---|---------------------------------|--|
|  S | GPIO_close | Close a Linux GPIO pin device. |
|  S | GPIO_configure | Configure a Linux GPIO pin. |
|  S | GPIO_line_close | Close a single GPIO line. |
|  S | GPIO_line_event | Read an edge trigger event from single GPIO line. |
|  S | GPIO_line_info | Get GPIO line information. |
|  S | GPIO_line_open | Open a single GPIO line. |
|  S | GPIO_line_read | Read the state of a single GPIO line. |
|  S | GPIO_line_write | Write the state of a single GPIO line. |
|  S | GPIO_open | Open a Linux GPIO pin device. |
|  S | GPIO_read | Read a Linux GPIO pin. |
|  S | GPIO_write | Write a Linux GPIO pin. |
|  S | HIDRAW_close | Close a Linux raw HID. |
|  S | HIDRAW_get_info | Get Linux raw HID bus type, vendor ID, and product ID. |

| | | |
|-----|---------------------------------|--|
| | HIDRAW_get_name | Get Linux raw HID name string. |
| ≡ S | HIDRAW_open1 | Open a Linux raw HID device by device node name. |
| ≡ S | HIDRAW_open2 | Open a Linux raw HID device by vendor ID and product ID. |
| ≡ S | HIDRAW_open3 | Open a Linux raw HID device by vendor ID and product ID and serial number. |
| ≡ S | HIDRAW_receive | Get a 64-byte report from a Linux HID. |
| ≡ S | HIDRAW_send | Send a 64-byte report to a Linux HID. |
| ≡ S | I2C_close | Close a Linux I ² C bus controller device. |
| ≡ S | I2C_open | Open a Linux I ² C bus controller device. |
| ≡ S | I2C_transaction | Send bytes to and/or receive bytes from an I ² C slave device. |
| ≡ S | IPV4_ntoa | Convert an IPv4 address to a dotted notation string (e.g. 1.2.3.4). |

| | | |
|---|---------------------------------------|--|
|  | IPV4_resolve | Resolve a domain name to an IPv4 host address. |
|  | LINUX_command | Execute a shell command string. |
|  | LINUX_detach | Detach the process and run it in the background. |
|  | LINUX_drop_privileges | Drop process privileges to those of the specified user. |
| Remarks | | |
| Only a process running at superuser privilege is allowed to drop privileges. | | |
|  | LINUX_errno | Fetch the value of <code>errno</code> . |
|  | LINUX_openlog | Open a connection to the <code>syslog</code> service. |
|  | LINUX_poll | Wait for an event on one or more files. |
|  | LINUX_strerror | Retrieve the error message for a particular <code>errno</code> error code. |
|  | LINUX_syslog | Send a message to the <code>syslog</code> service. |

| | | |
|-----|-----------------|---|
| | LINUX_usleep | Sleep for the specified number of microseconds. |
| ≡ S | PWM_close | Close a Linux PWM output device. |
| ≡ S | PWM_configure | Configure a Linux PWM output device. |
| ≡ S | PWM_open | Open a Linux PWM output device. |
| ≡ S | PWM_write | Set a Linux PWM output device duty cycle. |
| ≡ S | SERIAL_close | Close a Linux serial port device. |
| ≡ S | SERIAL_open | Open a Linux serial port device. |
| ≡ S | SERIAL_receive | Receive data from a Linux serial port device. |
| ≡ S | SERIAL_send | Send data to a Linux serial port device. |
| ≡ S | SPI_close | Close a Linux SPI device. |
| ≡ S | SPI_open | Open a Linux SPI device. |
| ≡ S | SPI_transaction | Send bytes to and/or receive bytes from a Linux SPI device. |

| | | |
|---|--------------------------------------|--|
| | STREAM_decode_frame | Decode a frame. |
| ≡  S | STREAM_encode_frame | Encode a frame. |
| ≡  S | STREAM_receive_frame | Receive an encoded frame. |
| ≡  S | STREAM_send_frame | Send an encoded frame. |
| ≡  S | TCP4_accept | Start TCP server and wait for a single connection. |
| ≡  S | TCP4_close | Close a TCP connection. |
| ≡  S | TCP4_connect | Connect to a TCP server. |
| ≡  S | TCP4_receive | Receive bytes from TCP peer. |
| ≡  S | TCP4_send | Send bytes to TCP peer. |
| ≡  S | TCP4_server | Start a TCP server and fork for each connection. |
| ≡  S | UDP4_close | Close a UDP socket. |
| ≡  S | UDP4_open | Open a UDP socket. |
| ≡  S | UDP4_receive | Receive a UDP datagram. |
| ≡  S | UDP4_send | Send a UDP datagram. |
| ≡  S | WATCHDOG_close | Close a Linux watchdog timer device. |
| ≡  S | WATCHDOG_get_timeout | Query a Linux watchdog |

timer device.



[WATCHDOG_kick](#)

Reset the watchdog timer.



[WATCHDOG_open](#)

Open a Linux watchdog timer device.



[WATCHDOG_set_timeout](#)

Change the watchdog timer period.

[Top](#)

Fields

| | Name | Description |
|---|--|--|
| • | GPIO_DIRECTION_INPUT | Input data direction. |
| • | GPIO_DIRECTION_OUTPUT | Out data direction. |
| • | GPIO_DRIVER_OPENDRAIN | Open drain (sink only) output driver. |
| • | GPIO_DRIVER_OPENSOURCE | Open source (source only) output driver |
| • | GPIO_DRIVER_PUSH_PULL | Push-pull (source and sink) output driver. |
| • | GPIO_EDGE_BOTH | Interrupt on both edges. |
| • | GPIO_EDGE_FALLING | Interrupt on falling edge. |
| • | GPIO_EDGE_NONE | Interrupts are |

disabled.

| | | |
|------------|--|---|
| • s | GPIO_EDGE_RISING | Interrupt on rising edge. |
| • s | GPIO_EVENT_REQUEST_BOTH | Enable GPIO input interrupt on both edges. |
| • s | GPIO_EVENT_REQUEST_FALLING | Enable GPIO input interrupt on falling edge. |
| • s | GPIO_EVENT_REQUEST_NONE | Disable GPIO input interrupt. |
| • s | GPIO_EVENT_REQUEST_RISING | Enable GPIO input interrupt on rising edge. |
| • s | GPIO_LINE_INFO_ACTIVE_LOW | GPIO line is configured as active low (inverted). |
| • s | GPIO_LINE_INFO_KERNEL | GPIO line is being used by the kernel. |
| • s | GPIO_LINE_INFO_OPEN_DRAIN | GPIO line is configured as open drain (current sink only). |
| • s | GPIO_LINE_INFO_OPEN_SOURCE | GPIO line is configured as open source (current source only). |

| | | |
|------------|---|--|
| • S | GPIO_LINE_INFO_OUTPUT | GPIO line is configured as an output. |
| • S | GPIO_LINE_REQUEST_ACTIVE_HIGH | Select GPIO line polarity active high (normal). |
| • S | GPIO_LINE_REQUEST_ACTIVE_LOW | Select GPIO line polarity active low (inverted). |
| • S | GPIO_LINE_REQUEST_INPUT | Select GPIO line direction input. |
| • S | GPIO_LINE_REQUEST_OPEN_DRAIN | Select GPIO line driver open drain (current sink only). |
| • S | GPIO_LINE_REQUEST_OPEN_SOURCE | Select GPIO line driver open source (current source only). |
| • S | GPIO_LINE_REQUEST_OUTPUT | Select GPIO line direction output. |
| • S | GPIO_LINE_REQUEST_PUSH_PULL | Select GPIO line driver push-pull (current source and sink). |
| • S | GPIO_POLARITY_ACTIVEHIGH | Active high (normal) polarity. |
| • S | GPIO_POLARITY_ACTIVELOW | Active low (inverted) |

polarity.

| | | |
|------------|------------------|---|
| ♦ s | INADDR_ANY | IPv4 address for binding to all network interfaces. |
| ♦ s | INADDR_BROADCAST | IPv4 broadcast address. |
| ♦ s | INADDR_LOOPBACK | IPv4 address for binding to the loopback interface (aka <code>localhost</code>). |
| ♦ s | LOG_ALERT | Action must be taken immediately. |
| ♦ s | LOG_AUTH | Security/authorization messages. |
| ♦ s | LOG_AUTHPRIV | Securit/authorization messages. |
| ♦ s | LOG_CONS | Write directly to the system console if there is an error while sending to the system logger. |
| ♦ s | LOG_CRIT | Critical condition. |
| ♦ s | LOG_CRON | <code>cron</code> daemon messages. |
| ♦ s | LOG_DAEMON | System daemons. |

| | | |
|------------|----------------------------|--------------------------------------|
| ♦ S | LOG_DEBUG | Debug message. |
| ♦ S | LOG_EMERG | System is unusable. |
| ♦ S | LOG_ERR | Error condition. |
| ♦ S | LOG_FTP | FTP daemon messages. |
| ♦ S | LOG_INFO | Informational message. |
| ♦ S | LOG_KERN | Kernel messages. |
| ♦ S | LOG_LOCAL0 | Reserved for local use. |
| ♦ S | LOG_LOCAL1 | Reserved for local use. |
| ♦ S | LOG_LOCAL2 | Reserved for local use. |
| ♦ S | LOG_LOCAL3 | Reserved for local use. |
| ♦ S | LOG_LOCAL4 | Reserved for local use. |
| ♦ S | LOG_LOCAL5 | Reserved for local use. |
| ♦ S | LOG_LOCAL6 | Reserved for local use. |
| ♦ S | LOG_LOCAL7 | Reserved for local use. |

| | | |
|------------|----------------------------|--|
| ♦ s | LOG_LPR | Line printer subsystem |
| ♦ s | LOG_MAIL | Mail system. |
| ♦ s | LOG_NDELAY | Open the connection immediately. Do not wait until syslog() is called for the first time. |
| ♦ s | LOG_NEWS | Network news subsystem |
| ♦ s | LOG_NOTICE | Normal but significant condition. |
| ♦ s | LOG_NOWAIT | Don't wait for child processes that may have been created while logging the message. (Not applicable to glibc .) |
| ♦ s | LOG_ODELAY | Do not open the connection immediately. Wait until syslog() is called for the first time. |
| ♦ s | LOG_PERROR | Also log the message to stderr . |
| ♦ s | LOG_PID | Include the caller's |

| | | |
|------------|-------------------------------|---|
| | | PID (process ID) with each message. |
| ♦ s | LOG_PROGNAME | Use the program name for the identity string. |
| ♦ s | LOG_SYSLOG | Messages generated internally by syslogd |
| ♦ s | LOG_USER | Random user-level messages. |
| ♦ s | LOG_UUCP | UUCP subsystem |
| ♦ s | LOG_WARNING | Warning condition. |
| ♦ s | MSG_DONTROUTE | Don't use a gateway to send out the packet, send to hosts only on directly connected networks. |
| ♦ s | MSG_DONTWAIT | Enables nonblocking operation; if the operation would block, EAGAIN or EWOULDBLOCK is returned. |
| ♦ s | MSG_MORE | The caller has more data to send. This flag informs the kernel to package all of the data sent in calls with |

this flag set into a single datagram which is transmitted only when a call is performed that does not specify this flag.

| | | |
|------------|-------------------------|--|
| • s | POLLERR | An error occurred. |
| • s | POLLHUP | Peer closed connection. |
| • s | POLLIN | There is data to read. |
| • s | POLLNVAL | File descriptor is invalid. |
| • s | POLLOUT | Writing is now possible. |
| • s | POLLPRI | There is urgent data to read. |
| • s | PWM_POLARITY_ACTIVEHIGH | Configure the PWM output as active high (normal). |
| • s | PWM_POLARITY_ACTIVELOW | Configure the PWM output as active low (inverted). |
| • s | SERIAL_PARITY_EVEN | Request even parity checking. |
| • s | SERIAL_PARITY_NONE | Disable parity checking. |

**S**

SERIAL_PARITY_ODD

Request odd parity checking.

**S**

SPI_AUTO_CS

Use hardware slave select.

[Top](#)

See Also

Reference

[IO.Bindings Namespace](#)

libsimpleio Methods

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

▪ Methods

| | Name | Description |
|---|------------------------------|--|
|   | ADC_close | Close a Linux IIO A/D converter input device. |
|   | ADC_get_name | Get the subsystem name for the specified Linux IIO A/D converter device. |
|   | ADC_open | Open a Linux IIO A/D converter input device. |
|   | ADC_read | Read a Linux IIO A/D converter input device. |
|   | DAC_close | Close a Linux IIO D/A converter output device. |
|   | DAC_get_name | Get the subsystem name for the specified Linux IIO D/A converter device. |
|   | DAC_open | Open a Linux IIO D/A |

| | | |
|---|---------------------|---|
| | | converter output device. |
| ≡   | DAC_write | Write to a Linux IIO D/A converter output device. |
| ≡   | EVENT_close | Close an <code>epoll</code> event dispatcher. |
| ≡   | EVENT_modify_fd | Modify a file registration. |
| ≡   | EVENT_open | Open an <code>epoll</code> event dispatcher. |
| ≡   | EVENT_register_fd | Register a file descriptor with an <code>epoll</code> event dispatcher. |
| ≡   | EVENT_unregister_fd | Unregister a file from an <code>epoll</code> dispatcher. |
| ≡   | EVENT_wait | Wait for events from an <code>epoll</code> dispatcher. |
| ≡   | GPIO_chip_info | Get GPIO chip information. |
| ≡   | GPIO_close | Close a Linux GPIO pin device. |
| ≡   | GPIO_configure | Configure a Linux GPIO pin. |
| ≡   | GPIO_line_close | Close a single GPIO line. |

| | | |
|---|-----------------|--|
| ≡ | GPIO_line_event | Read an edge trigger event from single GPIO line. |
| ≡ | GPIO_line_info | Get GPIO line information. |
| ≡ | GPIO_line_open | Open a single GPIO line. |
| ≡ | GPIO_line_read | Read the state of a single GPIO line. |
| ≡ | GPIO_line_write | Write the state of a single GPIO line. |
| ≡ | GPIO_open | Open a Linux GPIO pin device. |
| ≡ | GPIO_read | Read a Linux GPIO pin. |
| ≡ | GPIO_write | Write a Linux GPIO pin. |
| ≡ | HIDRAW_close | Close a Linux raw HID. |
| ≡ | HIDRAW_get_info | Get Linux raw HID bus type, vendor ID, and product ID. |
| ≡ | HIDRAW_get_name | Get Linux raw HID name string. |
| ≡ | HIDRAW_open1 | Open a Linux raw HID device by device node name. |

| | | |
|---|---------------------------------|--|
| | HIDRAW_open2 | Open a Linux raw HID device by vendor ID and product ID. |
| ≡   | HIDRAW_open3 | Open a Linux raw HID device by vendor ID and product ID and serial number. |
| ≡   | HIDRAW_receive | Get a 64-byte report from a Linux HID. |
| ≡   | HIDRAW_send | Send a 64-byte report to a Linux HID. |
| ≡   | I2C_close | Close a Linux I ² C bus controller device. |
| ≡   | I2C_open | Open a Linux I ² C bus controller device. |
| ≡   | I2C_transaction | Send bytes to and/or receive bytes from an I ² C slave device. |
| ≡   | IPV4_ntoa | Convert an IPv4 address to a dotted notation string (e.g. 1.2.3.4). |
| ≡   | IPV4_resolve | Resolve a domain name to an IPv4 host address. |
| ≡   | LINUX_command | Execute a shell command string. |

| | | |
|----------------|-----------------------|--|
| | LINUX_detach | Detach the process and run it in the background. |
| ≡ S | LINUX_drop_privileges | Drop process privileges to those of the specified user. |
| Remarks | | Only a process running at superuser privilege is allowed to drop privileges. |
| ≡ S | LINUX_errno | Fetch the value of <code>errno</code> . |
| ≡ S | LINUX_openlog | Open a connection to the <code>syslog</code> service. |
| ≡ S | LINUX_poll | Wait for an event on one or more files. |
| ≡ S | LINUX_strerror | Retrieve the error message for a particular <code>errno</code> error code. |
| ≡ S | LINUX_syslog | Send a message to the <code>syslog</code> service. |
| ≡ S | LINUX_usleep | Sleep for the specified number of microseconds. |

| | | |
|---|---------------------|---|
| | PWM_close | Close a Linux PWM output device. |
| ≡   | PWM_configure | Configure a Linux PWM output device. |
| ≡   | PWM_open | Open a Linux PWM output device. |
| ≡   | PWM_write | Set a Linux PWM output device duty cycle. |
| ≡   | SERIAL_close | Close a Linux serial port device. |
| ≡   | SERIAL_open | Open a Linux serial port device. |
| ≡   | SERIAL_receive | Receive data from a Linux serial port device. |
| ≡   | SERIAL_send | Send data to a Linux serial port device. |
| ≡   | SPI_close | Close a Linux SPI device. |
| ≡   | SPI_open | Open a Linux SPI device. |
| ≡   | SPI_transaction | Send bytes to and/or receive bytes from a Linux SPI device. |
| ≡   | STREAM_decode_frame | Decode a frame. |
| ≡   | STREAM_encode_frame | Encode a frame. |

| | | |
|--------|--------------------------------------|--|
| ≡ S | STREAM_receive_frame | Receive an encoded frame. |
| ≡ S | STREAM_send_frame | Send an encoded frame. |
| ≡ S | TCP4_accept | Start TCP server and wait for a single connection. |
| ≡ S | TCP4_close | Close a TCP connection. |
| ≡ S | TCP4_connect | Connect to a TCP server. |
| ≡ S | TCP4_receive | Receive bytes from TCP peer. |
| ≡ S | TCP4_send | Send bytes to TCP peer. |
| ≡ S | TCP4_server | Start a TCP server and fork for each connection. |
| ≡ S | UDP4_close | Close a UDP socket. |
| ≡ S | UDP4_open | Open a UDP socket. |
| ≡ S | UDP4_receive | Receive a UDP datagram. |
| ≡ S | UDP4_send | Send a UDP datagram. |
| ≡ S | WATCHDOG_close | Close a Linux watchdog timer device. |

| | | | |
|---|-----|--------------------------------------|--------------------------------------|
| ≡ | ⌚ S | WATCHDOG_get_timeout | Query a Linux watchdog timer device. |
| ≡ | ⌚ S | WATCHDOG_kick | Reset the watchdog timer. |
| ≡ | ⌚ S | WATCHDOG_open | Open a Linux watchdog timer device. |
| ≡ | ⌚ S | WATCHDOG_set_timeout | Change the watchdog timer period. |

[Top](#)

◀ See Also

Reference

[libsimpleio Class](#)
[IO.Bindings Namespace](#)

libsimpleioADC_close Method

Close a Linux IIO A/D converter input device.

Namespace: [IO.Bindings](#)

Assembly: libsimepleio (in libsimepleio.dll) Version: 2.2021.332.1

◀ Syntax

C# VB F#

[Copy](#)

```
public static void ADC_close(
    int fd,
    out int error
)
```

Parameters

fd

Type: [SystemInt32](#)

File descriptor.

error

Type: [SystemInt32](#)

Error code. Zero upon success or an [errno](#) value upon failure.

◀ See Also

[Reference](#)

[libsimepleio Class](#)

[IO.Bindings Namespace](#)

libsimpleioADC_get_name Method

Get the subsystem name for the specified Linux IIO A/D converter device.

Namespace: [IO.Bindings](#)

Assembly: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

↳ Syntax

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

[Copy](#)

```
public static void ADC_get_name(
    int chip,
    StringBuilder name,
    int size,
    out int error
)
```

Parameters

chip

Type: [System.Int32](#)

Linux IIO device number.

name

Type: [System.Text.StringBuilder](#)

Destination buffer.

size

Type: [System.Int32](#)

Size of destination buffer.

error

Type: [SystemInt32](#)

Error code. Zero upon success or an [errno](#) value upon failure.

◀ See Also

Reference

[libsimpleio Class](#)

[IO.Bindings Namespace](#)

libsimpleioADC_open Method

Open a Linux IIO A/D converter input device.

Namespace: [IO.Bindings](#)

Assembly: libsimepleio (in libsimepleio.dll) Version: 2.2021.332.1

◀ Syntax

C# VB F#

[Copy](#)

```
public static void ADC_open(
    int chip,
    int channel,
    out int fd,
    out int error
)
```

Parameters

chip

Type: [SystemInt32](#)

Linux IIO device number.

channel

Type: [SystemInt32](#)

Input channel number.

fd

Type: [SystemInt32](#)

File descriptor.

error

Type: [SystemInt32](#)

Error code. Zero upon success or an [errno](#) value upon failure.

↳ See Also

Reference

[libsimpleio Class](#)

[IO.Bindings Namespace](#)

libsimpleioADC_read Method

Read a Linux IIO A/D converter input device.

Namespace: [IO.Bindings](#)

Assembly: libsimepleio (in libsimepleio.dll) Version: 2.2021.332.1

↳ Syntax

C# VB F#

[Copy](#)

```
public static void ADC_read(
    int fd,
    out int sample,
    out int error
)
```

Parameters

fd

Type: [SystemInt32](#)

File descriptor.

sample

Type: [SystemInt32](#)

Analog sample data.

error

Type: [SystemInt32](#)

Error code. Zero upon success or an [errno](#) value upon failure.

↳ See Also

[Reference](#)

[libsimepleio Class](#)

IO.Bindings Namespace

libsimpleioDAC_close Method

Close a Linux IIO D/A converter output device.

Namespace: [IO.Bindings](#)

Assembly: libsimepleio (in libsimepleio.dll) Version: 2.2021.332.1

◀ Syntax

C# VB F#

[Copy](#)

```
public static void DAC_close(
    int fd,
    out int error
)
```

Parameters

fd

Type: [SystemInt32](#)

File descriptor.

error

Type: [SystemInt32](#)

Error code. Zero upon success or an [errno](#) value upon failure.

◀ See Also

[Reference](#)

[libsimepleio Class](#)

[IO.Bindings Namespace](#)

libsimpleioDAC_get_name Method

Get the subsystem name for the specified Linux IIO D/A converter device.

Namespace: [IO.Bindings](#)

Assembly: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

↳ Syntax

C# VB F#

[Copy](#)

```
public static void DAC_get_name(
    int chip,
    StringBuilder name,
    int size,
    out int error
)
```

Parameters

chip

Type: [System.Int32](#)

Linux IIO device number.

name

Type: [System.Text.StringBuilder](#)

Destination buffer.

size

Type: [System.Int32](#)

Size of destination buffer.

error

Type: [SystemInt32](#)

Error code. Zero upon success or an [errno](#) value upon failure.

◀ See Also

Reference

[libsimpleio Class](#)

[IO.Bindings Namespace](#)

libsimpleioDAC_open Method

Open a Linux IIO D/A converter output device.

Namespace: [IO.Bindings](#)

Assembly: libsimepleio (in libsimepleio.dll) Version: 2.2021.332.1

◀ Syntax

C# VB F#

[Copy](#)

```
public static void DAC_open(
    int chip,
    int channel,
    out int fd,
    out int error
)
```

Parameters

chip

Type: [SystemInt32](#)

Linux IIO device number.

channel

Type: [SystemInt32](#)

Output channel number.

fd

Type: [SystemInt32](#)

File descriptor.

error

Type: [SystemInt32](#)

Error code. Zero upon success or an [errno](#) value upon failure.

↳ See Also

Reference

[libsimpleio Class](#)

[IO.Bindings Namespace](#)

libsimpleioDAC_write Method

Write to a Linux IIO D/A converter output device.

Namespace: [IO.Bindings](#)

Assembly: libsimepleio (in libsimepleio.dll) Version: 2.2021.332.1

↳ Syntax

C# VB F#

[Copy](#)

```
public static void DAC_write(
    int fd,
    int sample,
    out int error
)
```

Parameters

fd

Type: [SystemInt32](#)

File descriptor.

sample

Type: [SystemInt32](#)

Analog sample data.

error

Type: [SystemInt32](#)

Error code. Zero upon success or an [errno](#) value upon failure.

↳ See Also

[Reference](#)

[libsimepleio Class](#)

IO.Bindings Namespace

libsimpleioEVENT_close Method

Close an `epoll` event dispatcher.

Namespace: [IO.Bindings](#)

Assembly: libsimepleio (in libsimepleio.dll) Version: 2.2021.332.1

◀ Syntax

C# VB F#

[Copy](#)

```
public static void EVENT_close(  
    int epfd,  
    out int error  
)
```

Parameters

epfd

Type: [SystemInt32](#)

File descriptor.

error

Type: [SystemInt32](#)

Error code. Zero upon success or an `errno` value upon failure.

◀ See Also

[Reference](#)

[libsimepleio Class](#)

[IO.Bindings Namespace](#)

libsimpleioEVENT_modify_fd Method

Modify a file registration.

Namespace: [IO.Bindings](#)

Assembly: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

◀ Syntax

C# VB F#

Copy

```
public static void EVENT_modify_fd(
    int epfd,
    int fd,
    int events,
    int handle,
    out int error
)
```

Parameters

epfd

Type: [SystemInt32](#)

File descriptor for the dispatcher.

fd

Type: [SystemInt32](#)

File descriptor to register for events.

events

Type: [SystemInt32](#)

Events to register for. May be a sum of the individual event flags.

handle

Type: [SystemInt32](#)

Event handle.

error

Type: [SystemInt32](#)

Error code. Zero upon success or an [errno](#) value upon failure.

See Also

Reference

[libsimpleio Class](#)

[IO.Bindings Namespace](#)

libsimpleioEVENT_open Method

Open an `epoll` event dispatcher.

Namespace: [IO.Bindings](#)

Assembly: libsimepleio (in libsimepleio.dll) Version: 2.2021.332.1

◀ Syntax

C# VB F#

[Copy](#)

```
public static void EVENT_open(
    out int epfd,
    out int error
)
```

Parameters

epfd

Type: [SystemInt32](#)

File descriptor.

error

Type: [SystemInt32](#)

Error code. Zero upon success or an `errno` value upon failure.

◀ See Also

[Reference](#)

[libsimepleio Class](#)

[IO.Bindings Namespace](#)

libsimpleioEVENT_register_fd Method

Register a file descriptor with an [epoll](#) event dispatcher.

Namespace: [IO.Bindings](#)

Assembly: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

◀ Syntax

C# VB F#

Copy

```
public static void EVENT_register_fd(  
    int epfd,  
    int fd,  
    int events,  
    int handle,  
    out int error  
)
```

Parameters

epfd

Type: [SystemInt32](#)

File descriptor for the dispatcher.

fd

Type: [SystemInt32](#)

File descriptor to register for events.

events

Type: [SystemInt32](#)

Events to register for. May be a sum of the individual event flags.

handle

Type: [SystemInt32](#)

Event handle.

error

Type: [SystemInt32](#)

Error code. Zero upon success or an [errno](#) value upon failure.

See Also

Reference

[libsimpleio Class](#)

[IO.Bindings Namespace](#)

libsimpleioEVENT_unregister_fd Method

Unregister a file from an `epoll` dispatcher.

Namespace: [IO.Bindings](#)

Assembly: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

◀ Syntax

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

[Copy](#)

```
public static void EVENT_unregister_fd(
    int epfd,
    int fd,
    out int error
)
```

Parameters

epfd

Type: [SystemInt32](#)

File descriptor for the dispatcher.>

fd

Type: [SystemInt32](#)

File descriptor.

error

Type: [SystemInt32](#)

Error code. Zero upon success or an `errno` value upon failure.

◀ See Also

Reference

[libsimpleio Class](#)
[IO.Bindings Namespace](#)

libsimpleioEVENT_wait Method

Wait for events from an `epoll` dispatcher.

Namespace: [IO.Bindings](#)

Assembly: libsimepleio (in libsimepleio.dll) Version: 2.2021.332.1

↳ Syntax

C# VB F#

[Copy](#)

```
public static void EVENT_wait(
    int epfd,
    out int fd,
    out int events,
    out int handle,
    int timeouts,
    out int error
)
```

Parameters

epfd

Type: [SystemInt32](#)

File descriptor for the dispatcher.

fd

Type: [SystemInt32](#)

File descriptor the event is applicable to.

events

Type: [SystemInt32](#)

Events that occurred. May be a sum of the individual event flags.

handle

Type: [SystemInt32](#)

Event handle provided when the file descriptor was registered.

timeoutms

Type: [SystemInt32](#)

Time in milliseconds to wait for an event.

error

Type: [SystemInt32](#)

Error code. Zero upon success or an [errno](#) value upon failure.

◀ See Also

Reference

[libsimpleio Class](#)

[IO.Bindings Namespace](#)

libsimpleioGPIO_chip_info Method

Get GPIO chip information.

Namespace: [IO.Bindings](#)

Assembly: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

◀ Syntax

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

[Copy](#)

```
public static void GPIO_chip_info(  
    int chip,  
    StringBuilder name,  
    int namesize,  
    StringBuilder Label,  
    int Labelsize,  
    out int Lines,  
    out int error  
)
```

Parameters

chip

Type: [SystemInt32](#)

GPIO chip number.

name

Type: [System.TextStringBuilder](#)

GPIO chip name.

namesize

Type: [SystemInt32](#)

Maximum size of name.

label

Type: [System.TextStringBuilder](#)

GPIO chip label.

labelsize

Type: [SystemInt32](#)

Maximum size of label.

lines

Type: [SystemInt32](#)

Number of GPIO lines.

error

Type: [SystemInt32](#)

Error code. Zero upon success or an [errno](#) value upon failure.

◀ See Also

Reference

[libsimpleio Class](#)

[IO.Bindings Namespace](#)

libsimpleioGPIO_close Method

Close a Linux GPIO pin device.

Namespace: [IO.Bindings](#)

Assembly: libsimepio (in libsimepio.dll) Version: 2.2021.332.1

◀ Syntax

C# VB F#

[Copy](#)

```
public static void GPIO_close(
    int fd,
    out int error
)
```

Parameters

fd

Type: [SystemInt32](#)

File descriptor.

error

Type: [SystemInt32](#)

Error code. Zero upon success or an [errno](#) value upon failure.

◀ See Also

[Reference](#)

[libsimepio Class](#)

[IO.Bindings Namespace](#)

libsimpleioGPIO_configure Method

Configure a Linux GPIO pin.

Namespace: [IO.Bindings](#)

Assembly: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

◀ Syntax

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

[Copy](#)

```
public static void GPIO_configure(
    int pin,
    int direction,
    int state,
    int edge,
    int polarity,
    out int error
)
```

Parameters

pin

Type: [SystemInt32](#)

Pin number.

direction

Type: [SystemInt32](#)

Data direction.

state

Type: [SystemInt32](#)

Initial GPIO output state.

edge

Type: [SystemInt32](#)

Interrupt edge for input pin.

polarity

Type: [SystemInt32](#)

Polarity

error

Type: [SystemInt32](#)

Error code. Zero upon success or an [errno](#) value upon failure.

◀ See Also

Reference

[libsimpleio Class](#)

[IO.Bindings Namespace](#)

libsimpleioGPIO_line_close Method

Close a single GPIO line.

Namespace: [IO.Bindings](#)

Assembly: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

« Syntax

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

[Copy](#)

```
public static void GPIO_line_close(
    int fd,
    out int error
)
```

Parameters

fd

Type: [SystemInt32](#)

Linux file descriptor.

error

Type: [SystemInt32](#)

Error code. Zero upon success or an [errno](#) value upon failure.

« See Also

[Reference](#)

[libsimpleio Class](#)

[IO.Bindings Namespace](#)

libsimpleioGPIO_line_event Method

Read an edge trigger event from single GPIO line.

Namespace: [IO.Bindings](#)

Assembly: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

◀ Syntax

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

[Copy](#)

```
public static void GPIO_line_event(
    int fd,
    out int state,
    out int error
)
```

Parameters

fd

Type: [SystemInt32](#)

Linux file descriptor.

state

Type: [SystemInt32](#)

State of the GPIO line after the edge trigger event.

error

Type: [SystemInt32](#)

Error code. Zero upon success or an [errno](#) value upon failure.

◀ See Also

Reference

[libsimpleio Class](#)
[IO.Bindings Namespace](#)

libsimpleioGPIO_line_info Method

Get GPIO line information.

Namespace: [IO.Bindings](#)

Assembly: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

◀ Syntax

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

[Copy](#)

```
public static void GPIO_line_info(
    int chip,
    int line,
    StringBuilder name,
    int namesize,
    StringBuilder Label,
    int labelsize,
    out int error
)
```

Parameters

chip

Type: [SystemInt32](#)

GPIO chip number.

line

Type: [SystemInt32](#)

GPIO line number.

name

Type: [System.TextStringBuilder](#)

GPIO line name.

namesize

Type: [SystemInt32](#)

Maximum size of name.

label

Type: [System.TextStringBuilder](#)

GPIO line label.

labelsize

Type: [SystemInt32](#)

Maximum size of label.

error

Type: [SystemInt32](#)

Error code. Zero upon success or an [errno](#) value upon failure.

◀ See Also

Reference

[libsimpleio Class](#)

[IO.Bindings Namespace](#)

libsimpleioGPIO_line_open Method

Open a single GPIO line.

Namespace: [IO.Bindings](#)

Assembly: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

◀ Syntax

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

[Copy](#)

```
public static void GPIO_line_open(
    int chip,
    int line,
    int flags,
    int events,
    int state,
    out int fd,
    out int error
)
```

Parameters

chip

Type: [SystemInt32](#)

GPIO chip number.

line

Type: [SystemInt32](#)

GPIO line number.

flags

Type: [SystemInt32](#)

GPIO line configuration flags.

events

Type: [SystemInt32](#)

GPIO line event flags.

state

Type: [SystemInt32](#)

Initial GPIO output state.

fd

Type: [SystemInt32](#)

Linux file descriptor.

error

Type: [SystemInt32](#)

Error code. Zero upon success or an [errno](#) value upon failure.

◀ See Also

Reference

[libsimpleio Class](#)

[IO.Bindings Namespace](#)

libsimpleioGPIO_line_read Method

Read the state of a single GPIO line.

Namespace: [IO.Bindings](#)

Assembly: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

◀ Syntax

C# VB F#

Copy

```
public static void GPIO_line_read(
    int fd,
    out int state,
    out int error
)
```

Parameters

fd

Type: [SystemInt32](#)

Linux file descriptor.

state

Type: [SystemInt32](#)

State of the GPIO line.

error

Type: [SystemInt32](#)

Error code. Zero upon success or an [errno](#) value upon failure.

◀ See Also

Reference

[libsimpleio Class](#)
[IO.Bindings Namespace](#)

libsimpleioGPIO_line_write Method

Write the state of a single GPIO line.

Namespace: [IO.Bindings](#)

Assembly: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

◀ Syntax

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

[Copy](#)

```
public static void GPIO_line_write(
    int fd,
    int state,
    out int error
)
```

Parameters

fd

Type: [SystemInt32](#)

Linux file descriptor.

state

Type: [SystemInt32](#)

State of the GPIO line.

error

Type: [SystemInt32](#)

Error code. Zero upon success or an [errno](#) value upon failure.

◀ See Also

Reference

[libsimpleio Class](#)
[IO.Bindings Namespace](#)

libsimpleioGPIO_open Method

Open a Linux GPIO pin device.

Namespace: [IO.Bindings](#)

Assembly: libsimepleio (in libsimepleio.dll) Version: 2.2021.332.1

↳ Syntax

C# VB F#

[Copy](#)

```
public static void GPIO_open(
    int pin,
    out int fd,
    out int error
)
```

Parameters

pin

Type: [SystemInt32](#)

Pin number.

fd

Type: [SystemInt32](#)

File descriptor.

error

Type: [SystemInt32](#)

Error code. Zero upon success or an [errno](#) value upon failure.

↳ See Also

[Reference](#)

[libsimepleio Class](#)

IO.Bindings Namespace

libsimpleioGPIO_read Method

Read a Linux GPIO pin.

Namespace: [IO.Bindings](#)

Assembly: libsimepleio (in libsimepleio.dll) Version: 2.2021.332.1

↳ Syntax

C# VB F#

[Copy](#)

```
public static void GPIO_read(
    int fd,
    out int state,
    out int error
)
```

Parameters

fd

Type: [SystemInt32](#)

File descriptor.

state

Type: [SystemInt32](#)

Pin state.

error

Type: [SystemInt32](#)

Error code. Zero upon success or an [errno](#) value upon failure.

↳ See Also

[Reference](#)

[libsimepleio Class](#)

IO.Bindings Namespace

libsimpleioGPIO_write Method

Write a Linux GPIO pin.

Namespace: [IO.Bindings](#)

Assembly: libsimepleio (in libsimepleio.dll) Version: 2.2021.332.1

↳ Syntax

C# VB F#

[Copy](#)

```
public static void GPIO_write(
    int fd,
    int state,
    out int error
)
```

Parameters

fd

Type: [SystemInt32](#)

File descriptor.

state

Type: [SystemInt32](#)

Pin state.

error

Type: [SystemInt32](#)

Error code. Zero upon success or an [errno](#) value upon failure.

↳ See Also

[Reference](#)

[libsimepleio Class](#)

IO.Bindings Namespace

libsimpleioHIDRAW_close Method

Close a Linux raw HID.

Namespace: [IO.Bindings](#)

Assembly: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

« Syntax

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

[Copy](#)

```
public static void HIDRAW_close(  
    int fd,  
    out int error  
)
```

Parameters

fd

Type: [SystemInt32](#)

File descriptor.

error

Type: [SystemInt32](#)

Error code. Zero upon success or an [errno](#) value upon failure.

« See Also

[Reference](#)

[libsimpleio Class](#)

[IO.Bindings Namespace](#)

libsimpleioHIDRAW_get_info Method

Get Linux raw HID bus type, vendor ID, and product ID.

Namespace: [IO.Bindings](#)

Assembly: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

◀ Syntax

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

[Copy](#)

```
public static void HIDRAW_get_info(
    int fd,
    [out] int bustype,
    [out] int vendor,
    [out] int product,
    [out] int error
)
```

Parameters

fd

Type: [SystemInt32](#)

File descriptor.

bustype

Type: [SystemInt32](#)

Bus type.

vendor

Type: [SystemInt32](#)

Vendor ID.

product

Type: [SystemInt32](#)

Product ID.

error

Type: [SystemInt32](#)

Error code. Zero upon success or an [errno](#) value upon failure.

See Also

Reference

[libsimpleio Class](#)

[IO.Bindings Namespace](#)

libsimpleioHIDRAW_get_name Method

Get Linux raw HID name string.

Namespace: [IO.Bindings](#)

Assembly: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

◀ Syntax

C# VB F#

Copy

```
public static void HIDRAW_get_name(
    int fd,
    StringBuilder name,
    int size,
    out int error
)
```

Parameters

fd

Type: [SystemInt32](#)

File descriptor.

name

Type: [System.TextStringBuilder](#)

Destination buffer.

size

Type: [SystemInt32](#)

Size of destination buffer.

error

Type: [SystemInt32](#)

Error code. Zero upon success or an [errno](#) value upon failure.

◀ See Also

Reference

[libsimpleio Class](#)

[IO.Bindings Namespace](#)

libsimpleioHIDRAW_open1

Method

Open a Linux raw HID device by device node name.

Namespace: [IO.Bindings](#)

Assembly: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

▪ Syntax

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

[Copy](#)

```
public static void HIDRAW_open1(
    string devname,
    out int fd,
    out int error
)
```

Parameters

devname

Type: [SystemString](#)

Device node name.

fd

Type: [SystemInt32](#)

Device node name.

error

Type: [SystemInt32](#)

Error code. Zero upon success or an [errno](#) value upon failure.

▪ See Also

Reference

[libsimpleio Class](#)
[IO.Bindings Namespace](#)

libsimpleioHIDRAW_open2 Method

Open a Linux raw HID device by vendor ID and product ID.

Namespace: [IO.Bindings](#)

Assembly: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

◀ Syntax

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

[Copy](#)

```
public static void HIDRAW_open2(
    int VID,
    int PID,
    out int fd,
    out int error
)
```

Parameters

VID

Type: [SystemInt32](#)

Vendor ID.

PID

Type: [SystemInt32](#)

Product ID.

fd

Type: [SystemInt32](#)

File descriptor.

error

Type: [SystemInt32](#)

Error code. Zero upon success or an [errno](#) value upon failure.

◀ See Also

Reference

[libsimpleio Class](#)

[IO.Bindings Namespace](#)

libsimpleioHIDRAW_open3 Method

Open a Linux raw HID device by vendor ID and product ID and serial number.

Namespace: [IO.Bindings](#)

Assembly: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

↳ Syntax

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

[Copy](#)

```
public static void HIDRAW_open3(
    int VID,
    int PID,
    string serial,
    out int fd,
    out int error
)
```

Parameters

VID

Type: [SystemInt32](#)

Vendor ID.

PID

Type: [SystemInt32](#)

Product ID.

serial

Type: [SystemString](#)

Serial number.

fd

Type: [SystemInt32](#)

File descriptor.

error

Type: [SystemInt32](#)

Error code. Zero upon success or an [errno](#) value upon failure.

See Also

Reference

[libsimpleio Class](#)

[IO.Bindings Namespace](#)

libsimpleioHIDRAW_receive Method

Get a 64-byte report from a Linux HID.

Namespace: [IO.Bindings](#)

Assembly: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

◀ Syntax

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

[Copy](#)

```
public static void HIDRAW_receive(
    int fd,
    byte[] buf,
    int bufsize,
    out int count,
    out int error
)
```

Parameters

fd

Type: [SystemInt32](#)

File descriptor.

buf

Type: [SystemByte](#)

Destination buffer.

bufsize

Type: [SystemInt32](#)

Destination buffer size.

count

Type: [SystemInt32](#)

Number of bytes actually received.

error

Type: [SystemInt32](#)

Error code. Zero upon success or an [errno](#) value upon failure.

◀ See Also

Reference

[libsimpleio Class](#)

[IO.Bindings Namespace](#)

libsimpleioHIDRAW_send Method

Send a 64-byte report to a Linux HID.

Namespace: [IO.Bindings](#)

Assembly: libsimepleio (in libsimepleio.dll) Version: 2.2021.332.1

◀ Syntax

C# VB F#

[Copy](#)

```
public static void HIDRAW_send(
    int fd,
    byte[] buf,
    int bufsize,
    out int count,
    out int error
)
```

Parameters

fd

Type: [SystemInt32](#)

File descriptor.

buf

Type: [SystemByte](#)

Source buffer.

bufsize

Type: [SystemInt32](#)

Source buffer size.

count

Type: [SystemInt32](#)

Number of bytes actually sent.

error

Type: [SystemInt32](#)

Error code. Zero upon success or an [errno](#) value upon failure.

◀ See Also

Reference

[libsimpleio Class](#)

[IO.Bindings Namespace](#)

libsimpleioI2C_close Method

Close a Linux I²C bus controller device.

Namespace: [IO.Bindings](#)

Assembly: libsimepleio (in libsimepleio.dll) Version: 2.2021.332.1

◀ Syntax

C# VB F#

[Copy](#)

```
public static void I2C_close(
    int fd,
    out int error
)
```

Parameters

fd

Type: [SystemInt32](#)

File descriptor.

error

Type: [SystemInt32](#)

Error code. Zero upon success or an [errno](#) value upon failure.

◀ See Also

Reference

[libsimepleio Class](#)

[IO.Bindings Namespace](#)

libsimpleioI2C_open Method

Open a Linux I²C bus controller device.

Namespace: [IO.Bindings](#)

Assembly: libsimepleio (in libsimepleio.dll) Version: 2.2021.332.1

↳ Syntax

C# VB F#

[Copy](#)

```
public static void I2C_open(
    string devname,
    out int fd,
    out int error
)
```

Parameters

devname

Type: [SystemString](#)

Device node name.

fd

Type: [SystemInt32](#)

File descriptor.

error

Type: [SystemInt32](#)

Error code. Zero upon success or an [errno](#) value upon failure.

↳ See Also

[Reference](#)

[libsimepleio Class](#)

IO.Bindings Namespace

libsimpleioI2C_transaction Method

Send bytes to and/or receive bytes from an I²C slave device.

Namespace: [IO.Bindings](#)

Assembly: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

◀ Syntax

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

[Copy](#)

```
public static void I2C_transaction(
    int fd,
    int slaveaddr,
    byte[] cmd,
    int cmdLen,
    byte[] resp,
    int resplen,
    out int error
)
```

Parameters

fd

Type: [SystemInt32](#)

File descriptor.

slaveaddr

Type: [SystemInt32](#)

Slave address.

cmd

Type: [SystemByte](#)

Source buffer.

cmdlen

Type: [SystemInt32](#)

Source buffer size.

resp

Type: [SystemByte](#)

Response buffer.

resplen

Type: [SystemInt32](#)

Response buffer size.

error

Type: [SystemInt32](#)

Error code. Zero upon success or an [errno](#) value upon failure.

◀ See Also

Reference

[libsimpleio Class](#)

[IO.Bindings Namespace](#)

libsimpleioIPv4_ntoa Method

Convert an IPv4 address to a dotted notation string (e.g. 1.2.3.4).

Namespace: [IO.Bindings](#)

Assembly: libsimepleio (in libsimepleio.dll) Version: 2.2021.332.1

◀ Syntax

C# VB F#

[Copy](#)

```
public static void IPV4_ntoa(
    int host,
    StringBuilder buf,
    int bufsize,
    out int error
)
```

Parameters

host

Type: [SystemInt32](#)

IPv4 host address

buf

Type: [System.TextStringBuilder](#)

Destination buffer.

bufsize

Type: [SystemInt32](#)

Destination buffer size.

error

Type: [SystemInt32](#)

Error code. Zero upon success or an [errno](#) value upon failure.

↳ See Also

Reference

[libsimpleio Class](#)

[IO.Bindings Namespace](#)

libsimpleioIPv4_resolve Method

Resolve a domain name to an IPv4 host address.

Namespace: [IO.Bindings](#)

Assembly: libsimepleio (in libsimepleio.dll) Version: 2.2021.332.1

◀ Syntax

C# VB F#

[Copy](#)

```
public static void IPV4_resolve(
    string hostname,
    out int host,
    out int error
)
```

Parameters

hostname

Type: [SystemString](#)

Host name to resolve.

host

Type: [SystemInt32](#)

IPv4 host address.

error

Type: [SystemInt32](#)

Error code. Zero upon success or an [errno](#) value upon failure.

◀ See Also

[Reference](#)

[libsimepleio Class](#)

IO.Bindings Namespace

libsimpleioLINUX_command Method

Execute a shell command string.

Namespace: [IO.Bindings](#)

Assembly: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

▪ Syntax

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

[Copy](#)

```
public static void LINUX_command(  
    string cmd,  
    out int error  
)
```

Parameters

cmd

Type: [SystemString](#)

Command string.

error

Type: [SystemInt32](#)

Error code.

▪ See Also

[Reference](#)

[libsimpleio Class](#)

[IO.Bindings Namespace](#)

libsimpleioLINUX_detach Method

Detach the process and run it in the background.

Namespace: [IO.Bindings](#)

Assembly: libsimepleio (in libsimepleio.dll) Version: 2.2021.332.1

◀ Syntax

C# VB F#

[Copy](#)

```
public static void LINUX_detach(  
    out int error  
)
```

Parameters

error

Type: [SystemInt32](#)

Error code. Zero upon success or an [errno](#) value upon failure.

◀ See Also

Reference

[libsimepleio Class](#)

[IO.Bindings Namespace](#)

libsimpleioLINUX_drop_privileges Method

Drop process privileges to those of the specified user.

Remarks

Only a process running at superuser privilege is allowed to drop privileges.

Namespace: [IO.Bindings](#)

Assembly: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

Syntax

C# VB F#

[Copy](#)

```
public static void LINUX_drop_privileges(
    string username,
    out int error
)
```

Parameters

username

Type: [SystemString](#)

User privileges to assume.

error

Type: [SystemInt32](#)

Error code. Zero upon success or an [errno](#) value upon failure.

See Also

Reference

[libsimpleio Class](#)
[IO.Bindings Namespace](#)

libsimpleioLINUX_errno Method

Fetch the value of `errno`.

Namespace: [IO.Bindings](#)

Assembly: libsimepleio (in libsimepleio.dll) Version: 2.2021.332.1

▪ Syntax

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

[Copy](#)

```
public static int LINUX_errno()
```

Return Value

Type: [Int32](#)

Current value of `errno`.

▪ See Also

Reference

[libsimepleio Class](#)

[IO.Bindings Namespace](#)

libsimpleioLINUX_openlog Method

Open a connection to the [syslog](#) service.

Namespace: [IO.Bindings](#)

Assembly: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

◀ Syntax

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

[Copy](#)

```
public static void LINUX_openlog(
    string id,
    int options,
    int facility,
    out int error
)
```

Parameters

id

Type: [SystemString](#)

Program identifier.

options

Type: [SystemInt32](#)

Logging options.

facility

Type: [SystemInt32](#)

Logging facility identifier.

error

Type: [SystemInt32](#)

Error code. Zero upon success or an [errno](#) value upon failure.

◀ See Also

Reference

[libsimpleio Class](#)

[IO.Bindings Namespace](#)

libsimpleioLINUX_poll Method

Wait for an event on one or more files.

Namespace: [IO.Bindings](#)

Assembly: libsimepleio (in libsimepleio.dll) Version: 2.2021.332.1

◀ Syntax

C# VB F#

[Copy](#)

```
public static void LINUX_poll(
    int numfiles,
    int[] files,
    int[] events,
    int[] results,
    int timeouts,
    out int error
)
```

Parameters

numfiles

Type: [SystemInt32](#)

Number elements in each of the following arrays.

files

Type: [SystemInt32](#)

File descriptors.

events

Type: [SystemInt32](#)

Events to wait for on each file descriptor.

results

Type: [SystemInt32](#)

Events that occurred on each file descriptor.

timeoutms

Type: [SystemInt32](#)

Milliseconds to wait for an event to occur. A value of -1 means wait forever and a value of 0 means do not wait at all.

error

Type: [SystemInt32](#)

Error code.

◀ See Also

Reference

[libsimpleio Class](#)

[IO.Bindings Namespace](#)

libsimpleioLINUX_strerror Method

Retrieve the error message for a particular `errno` error code.

Namespace: [IO.Bindings](#)

Assembly: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

◀ Syntax

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

[Copy](#)

```
public static void LINUX_strerror(  
    int error,  
    StringBuilder buf,  
    int bufsize  
)
```

Parameters

error

Type: [SystemInt32](#)

Error code.

buf

Type: [System.TextStringBuilder](#)

Destination buffer.

bufsize

Type: [SystemInt32](#)

Destination buffer size.

◀ See Also

Reference

[libsimpleio Class](#)
[IO.Bindings Namespace](#)

libsimpleioLINUX_syslog Method

Send a message to the `syslog` service.

Namespace: [IO.Bindings](#)

Assembly: libsimepleio (in libsimepleio.dll) Version: 2.2021.332.1

↳ Syntax

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

[Copy](#)

```
public static void LINUX_syslog(
    int priority,
    string msg,
    out int error
)
```

Parameters

priority

Type: [SystemInt32](#)

Message priority

msg

Type: [SystemString](#)

Message to send.

error

Type: [SystemInt32](#)

Error code. Zero upon success or an `errno` value upon failure.

↳ See Also

[Reference](#)

[libsimepleio Class](#)

IO.Bindings Namespace

libsimpleioLINUX_usleep Method

Sleep for the specified number of microseconds.

Namespace: [IO.Bindings](#)

Assembly: libsimepleio (in libsimepleio.dll) Version: 2.2021.332.1

◀ Syntax

C# VB F#

[Copy](#)

```
public static void LINUX_usleep(
    int microsecs,
    out int error
)
```

Parameters

microsecs

Type: [SystemInt32](#)

Number of microseconds to sleep.

error

Type: [SystemInt32](#)

Error code.

◀ See Also

[Reference](#)

[libsimepleio Class](#)

[IO.Bindings Namespace](#)

libsimpleioPWM_close Method

Close a Linux PWM output device.

Namespace: [IO.Bindings](#)

Assembly: libsimepleio (in libsimepleio.dll) Version: 2.2021.332.1

◀ Syntax

C# VB F#

[Copy](#)

```
public static void PWM_close(
    int fd,
    [out] int error
)
```

Parameters

fd

Type: [SystemInt32](#)

File descriptor.

error

Type: [SystemInt32](#)

Error code. Zero upon success or an [errno](#) value upon failure.

◀ See Also

[Reference](#)

[libsimepleio Class](#)

[IO.Bindings Namespace](#)

libsimpleioPWM_configure Method

Configure a Linux PWM output device.

Namespace: [IO.Bindings](#)

Assembly: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

◀ Syntax

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

[Copy](#)

```
public static void PWM_configure(
    int chip,
    int channel,
    int period,
    int ontime,
    int polarity,
    out int error
)
```

Parameters

chip

Type: [SystemInt32](#)

Chip number.

channel

Type: [SystemInt32](#)

Channel number.

period

Type: [SystemInt32](#)

Pulse period in microseconds.

ontime

Type: [SystemInt32](#)

Initial on time in microseconds.

polarity

Type: [SystemInt32](#)

PWM output polarity (0 for active low/inverted or 1 for active high/normal).

error

Type: [SystemInt32](#)

Error code. Zero upon success or an [errno](#) value upon failure.

▪ Remarks

On many platforms two or more PWM outputs may share the same clock generator, so configuring different PWM pulse periods may not be possible.

▪ Remarks

Not all platforms support active low (inverted) PWM outputs.

▪ See Also

Reference

[libsimpleio Class](#)

[IO.Bindings Namespace](#)

libsimpleioPWM_open Method

Open a Linux PWM output device.

Namespace: [IO.Bindings](#)

Assembly: libsimepleio (in libsimepleio.dll) Version: 2.2021.332.1

◀ Syntax

C# VB F#

[Copy](#)

```
public static void PWM_open(
    int chip,
    int channel,
    out int fd,
    out int error
)
```

Parameters

chip

Type: [SystemInt32](#)

Chip number.

channel

Type: [SystemInt32](#)

Channel number.

fd

Type: [SystemInt32](#)

File descriptor.

error

Type: [SystemInt32](#)

Error code. Zero upon success or an [errno](#) value upon failure.

↳ See Also

Reference

[libsimpleio Class](#)

[IO.Bindings Namespace](#)

libsimpleioPWM_write Method

Set a Linux PWM output device duty cycle.

Namespace: [IO.Bindings](#)

Assembly: libsimepleio (in libsimepleio.dll) Version: 2.2021.332.1

« Syntax

C# VB F#

[Copy](#)

```
public static void PWM_write(
    int fd,
    int ontime,
    out int error
)
```

Parameters

fd

Type: [SystemInt32](#)

File descriptor.

ontime

Type: [SystemInt32](#)

On time in microseconds.

error

Type: [SystemInt32](#)

Error code. Zero upon success or an [errno](#) value upon failure.

« See Also

[Reference](#)

[libsimepleio Class](#)

IO.Bindings Namespace

libsimpleioSERIAL_close Method

Close a Linux serial port device.

Namespace: [IO.Bindings](#)

Assembly: libsimepleio (in libsimepleio.dll) Version: 2.2021.332.1

◀ Syntax

C# VB F#

[Copy](#)

```
public static void SERIAL_close(
    int fd,
    out int error
)
```

Parameters

fd

Type: [SystemInt32](#)

File descriptor.

error

Type: [SystemInt32](#)

Error code. Zero upon success or an [errno](#) value upon failure.

◀ See Also

[Reference](#)

[libsimepleio Class](#)

[IO.Bindings Namespace](#)

libsimpleioSERIAL_open Method

Open a Linux serial port device.

Namespace: [IO.Bindings](#)

Assembly: libsimepleio (in libsimepleio.dll) Version: 2.2021.332.1

◀ Syntax

C# VB F#

[Copy](#)

```
public static void SERIAL_open(
    string devname,
    int baudrate,
    int parity,
    int databits,
    int stopbits,
    out int fd,
    out int error
)
```

Parameters

devname

Type: [SystemString](#)

Device node name.

baudrate

Type: [SystemInt32](#)

Baud rate.

parity

Type: [SystemInt32](#)

Parity setting (0 to 2).

databits

Type: [SystemInt32](#)

Word size setting (5 to 8).

stopbits

Type: [SystemInt32](#)

Number of stop bits (1 or 2).

fd

Type: [SystemInt32](#)

File descriptor.

error

Type: [SystemInt32](#)

Error code. Zero upon success or an [errno](#) value upon failure.

◀ See Also

Reference

[libsimpleio Class](#)

[IO.Bindings Namespace](#)

libsimpleioSERIAL_receive Method

Receive data from a Linux serial port device.

Namespace: [IO.Bindings](#)

Assembly: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

◀ Syntax

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

[Copy](#)

```
public static void SERIAL_receive(
    int fd,
    byte[] buf,
    int bufsize,
    out int count,
    out int error
)
```

Parameters

fd

Type: [SystemInt32](#)

File descriptor.

buf

Type: [SystemByte](#)

Destination buffer.

bufsize

Type: [SystemInt32](#)

Destination buffer size.

count

Type: [SystemInt32](#)

Number of bytes actually received.

error

Type: [SystemInt32](#)

Error code. Zero upon success or an [errno](#) value upon failure.

See Also

Reference

[libsimpleio Class](#)

[IO.Bindings Namespace](#)

libsimpleioSERIAL_send Method

Send data to a Linux serial port device.

Namespace: [IO.Bindings](#)

Assembly: libsimepleio (in libsimepleio.dll) Version: 2.2021.332.1

◀ Syntax

C# VB F#

[Copy](#)

```
public static void SERIAL_send(  
    int fd,  
    byte[] buf,  
    int bufsize,  
    out int count,  
    out int error  
)
```

Parameters

fd

Type: [SystemInt32](#)

File descriptor.

buf

Type: [SystemByte](#)

Source buffer.

bufsize

Type: [SystemInt32](#)

Source buffer size.

count

Type: [SystemInt32](#)

Number of bytes actually sent.

error

Type: [SystemInt32](#)

Error code. Zero upon success or an [errno](#) value upon failure.

◀ See Also

Reference

[libsimpleio Class](#)

[IO.Bindings Namespace](#)

libsimpleioSPI_close Method

Close a Linux SPI device.

Namespace: [IO.Bindings](#)

Assembly: libsimepleio (in libsimepleio.dll) Version: 2.2021.332.1

◀ Syntax

C# VB F#

[Copy](#)

```
public static void SPI_close(
    int fd,
    out int error
)
```

Parameters

fd

Type: [SystemInt32](#)

File descriptor.

error

Type: [SystemInt32](#)

Error code. Zero upon success or an [errno](#) value upon failure.

◀ See Also

[Reference](#)

[libsimepleio Class](#)

[IO.Bindings Namespace](#)

libsimpleioSPI_open Method

Open a Linux SPI device.

Namespace: [IO.Bindings](#)

Assembly: libsimepleio (in libsimepleio.dll) Version: 2.2021.332.1

◀ Syntax

C# VB F#

[Copy](#)

```
public static void SPI_open(
    string devname,
    int mode,
    int wordsize,
    int speed,
    out int fd,
    out int error
)
```

Parameters

devname

Type: [SystemString](#)

Device node name.

mode

Type: [SystemInt32](#)

SPI transfer mode (0 .. 3)

wordsize

Type: [SystemInt32](#)

SPI transfer word size (8, 16, or 32).

speed

Type: [SystemInt32](#)

SPI transfer speed in Hz.

fd

Type: [SystemInt32](#)

File descriptor.

error

Type: [SystemInt32](#)

Error code. Zero upon success or an [errno](#) value upon failure.

Remarks

The Linux kernel creates device nodes for each SPI slave device, of the form `/dev/spidevX.Y` where `X` is the SPI bus controller number and `Y` is the SPI slave select number.

See Also

Reference

[libsimpleio Class](#)

[IO.Bindings Namespace](#)

libsimpleioSPI_transaction Method

Send bytes to and/or receive bytes from a Linux SPI device.

Namespace: [IO.Bindings](#)

Assembly: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

◀ Syntax

C# VB F#

Copy

```
public static void SPI_transaction(
    int fd,
    int csfd,
    byte[] cmd,
    int cmdLen,
    int delayus,
    byte[] resp,
    int resplen,
    out int error
)
```

Parameters

fd

Type: [SystemInt32](#)

File descriptor.

csfd

Type: [SystemInt32](#)

Chip select file descriptor.

cmd

Type: [SystemByte](#)

Source buffer.

cmdlen

Type: [SystemInt32](#)

Source buffer size.

delayus

Type: [SystemInt32](#)

Delay in microseconds between the write and read operations.

resp

Type: [SystemByte](#)

Destination buffer.

resplen

Type: [SystemInt32](#)

Destination buffer size.

error

Type: [SystemInt32](#)

Error code. Zero upon success or an [errno](#) value upon failure.

See Also

[Reference](#)

[libsimpleio Class](#)

[IO.Bindings Namespace](#)

libsimpleioSTREAM_decode_frame Method

Decode a frame.

Namespace: [IO.Bindings](#)

Assembly: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

◀ Syntax

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

[Copy](#)

```
public static void STREAM_decode_frame(
    byte[] src,
    int srcLen,
    byte[] dst,
    int dstSize,
    out int dstLen,
    out int error
)
```

Parameters

src

Type: [SystemByte](#)

Source buffer.

srcLen

Type: [SystemInt32](#)

Source buffer size.

dst

Type: [SystemByte](#)

Destination buffer.

dstsize

Type: [SystemInt32](#)

Destination buffer size.

dstlen

Type: [SystemInt32](#)

Size of decoded frame.

error

Type: [SystemInt32](#)

Error code. Zero upon success or an [errno](#) value upon failure.

◀ See Also

Reference

[libsimpleio Class](#)

[IO.Bindings Namespace](#)

libsimpleioSTREAM_encode_frame Method

Encode a frame.

Namespace: [IO.Bindings](#)

Assembly: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

◀ Syntax

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

[Copy](#)

```
public static void STREAM_encode_frame(
    byte[] src,
    int srcLen,
    byte[] dst,
    int dstSize,
    out int dstLen,
    out int error
)
```

Parameters

src

Type: [SystemByte](#)

Source buffer.

srcLen

Type: [SystemInt32](#)

Source buffer size.

dst

Type: [SystemByte](#)

Destination buffer.

dstsize

Type: [SystemInt32](#)

Destination buffer size.

dstlen

Type: [SystemInt32](#)

Size of encoded frame.

error

Type: [SystemInt32](#)

Error code. Zero upon success or an [errno](#) value upon failure.

◀ See Also

Reference

[libsimpleio Class](#)

[IO.Bindings Namespace](#)

libsimpleioSTREAM_receive_frame Method

Receive an encoded frame.

Namespace: [IO.Bindings](#)

Assembly: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

◀ Syntax

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

[Copy](#)

```
public static void STREAM_receive_frame(
    int fd,
    byte[] buf,
    int bufsize,
    out int count,
    out int error
)
```

Parameters

fd

Type: [SystemInt32](#)

File descriptior.

buf

Type: [SystemByte](#)

Destination buffer.

bufsize

Type: [SystemInt32](#)

Destination buffer size.

count

Type: [SystemInt32](#)

Number of bytes actually received.

error

Type: [SystemInt32](#)

Error code. Zero upon success or an [errno](#) value upon failure.

◀ See Also

Reference

[libsimpleio Class](#)

[IO.Bindings Namespace](#)

libsimpleioSTREAM_send_frame Method

Send an encoded frame.

Namespace: [IO.Bindings](#)

Assembly: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

◀ Syntax

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

[Copy](#)

```
public static void STREAM_send_frame(
    int fd,
    byte[] buf,
    int bufsize,
    out int count,
    out int error
)
```

Parameters

fd

Type: [SystemInt32](#)

File descriptor.

buf

Type: [SystemByte](#)

Source buffer.

bufsize

Type: [SystemInt32](#)

Source buffer size.

count

Type: [SystemInt32](#)

Number of bytes actually sent.

error

Type: [SystemInt32](#)

Error code. Zero upon success or an [errno](#) value upon failure.

See Also

Reference

[libsimpleio Class](#)

[IO.Bindings Namespace](#)

libsimpleioTCP4_accept Method

Start TCP server and wait for a single connection.

Namespace: [IO.Bindings](#)

Assembly: libsimepleio (in libsimepleio.dll) Version: 2.2021.332.1

◀ Syntax

C# VB F#

[Copy](#)

```
public static void TCP4_accept(
    int host,
    int port,
    out int fd,
    out int error
)
```

Parameters

host

Type: [SystemInt32](#)

IPv4 address, of the interface to listen on. Use 0.0.0.0 to listen on all interfaces.

port

Type: [SystemInt32](#)

TCP port number.

fd

Type: [SystemInt32](#)

File descriptor.

error

Type: [SystemInt32](#)

Error code. Zero upon success or an [errno](#) value upon failure.

◀ See Also

Reference

[libsimpleio Class](#)

[IO.Bindings Namespace](#)

libsimpleioTCP4_close Method

Close a TCP connection.

Namespace: [IO.Bindings](#)

Assembly: libsimepleio (in libsimepleio.dll) Version: 2.2021.332.1

◀ Syntax

C# VB F#

[Copy](#)

```
public static void TCP4_close(
    int fd,
    out int error
)
```

Parameters

fd

Type: [SystemInt32](#)

File descriptor.

error

Type: [SystemInt32](#)

Error code. Zero upon success or an [errno](#) value upon failure.

◀ See Also

[Reference](#)

[libsimepleio Class](#)

[IO.Bindings Namespace](#)

libsimpleioTCP4_connect Method

Connect to a TCP server.

Namespace: [IO.Bindings](#)

Assembly: libsimepleio (in libsimepleio.dll) Version: 2.2021.332.1

◀ Syntax

C# VB F#

[Copy](#)

```
public static void TCP4_connect(
    int host,
    int port,
    out int fd,
    out int error
)
```

Parameters

host

Type: [SystemInt32](#)

IPv4 host address.

port

Type: [SystemInt32](#)

TCP port number.

fd

Type: [SystemInt32](#)

File descriptor.

error

Type: [SystemInt32](#)

Error code. Zero upon success or an [errno](#) value upon failure.

↳ See Also

Reference

[libsimpleio Class](#)

[IO.Bindings Namespace](#)

libsimpleioTCP4_receive Method

Receive bytes from TCP peer.

Namespace: [IO.Bindings](#)

Assembly: libsimepleio (in libsimepleio.dll) Version: 2.2021.332.1

◀ Syntax

C# VB F#

[Copy](#)

```
public static void TCP4_receive(
    int fd,
    byte[] buf,
    int bufsize,
    out int count,
    out int error
)
```

Parameters

fd

Type: [SystemInt32](#)

File descriptor.

buf

Type: [SystemByte](#)

Destination buffer.

bufsize

Type: [SystemInt32](#)

Destination buffer size.

count

Type: [SystemInt32](#)

Number of bytes actually received.

error

Type: [SystemInt32](#)

Error code. Zero upon success or an [errno](#) value upon failure.

◀ See Also

Reference

[libsimpleio Class](#)

[IO.Bindings Namespace](#)

libsimpleioTCP4_send Method

Send bytes to TCP peer.

Namespace: [IO.Bindings](#)

Assembly: libsimepleio (in libsimepleio.dll) Version: 2.2021.332.1

◀ Syntax

C# VB F#

[Copy](#)

```
public static void TCP4_send(
    int fd,
    byte[] buf,
    int bufsize,
    out int count,
    out int error
)
```

Parameters

fd

Type: [SystemInt32](#)

File descriptor.

buf

Type: [SystemByte](#)

Source buffer.

bufsize

Type: [SystemInt32](#)

Source buffer size.

count

Type: [SystemInt32](#)

Number of bytes actually sent.

error

Type: [SystemInt32](#)

Error code. Zero upon success or an [errno](#) value upon failure.

◀ See Also

Reference

[libsimpleio Class](#)

[IO.Bindings Namespace](#)

libsimpleioTCP4_server Method

Start a TCP server and fork for each connection.

Namespace: [IO.Bindings](#)

Assembly: libsimepleio (in libsimepleio.dll) Version: 2.2021.332.1

◀ Syntax

C# VB F#

[Copy](#)

```
public static void TCP4_server(  
    int host,  
    int port,  
    out int fd,  
    out int error  
)
```

Parameters

host

Type: [SystemInt32](#)

IPv4 address, of the interface to listen on. Use 0.0.0.0 to listen on all interfaces.

port

Type: [SystemInt32](#)

TCP port number.

fd

Type: [SystemInt32](#)

File descriptor.

error

Type: [SystemInt32](#)

Error code. Zero upon success or an [errno](#) value upon failure.

◀ See Also

Reference

[libsimpleio Class](#)

[IO.Bindings Namespace](#)

libsimpleioUDP4_close Method

Close a UDP socket.

Namespace: [IO.Bindings](#)

Assembly: libsimepleio (in libsimepleio.dll) Version: 2.2021.332.1

◀ Syntax

C# VB F#

[Copy](#)

```
public static void UDP4_close(
    int fd,
    out int error
)
```

Parameters

fd

Type: [SystemInt32](#)

File descriptor.

error

Type: [SystemInt32](#)

Error code. Zero upon success or an [errno](#) value upon failure.

◀ See Also

[Reference](#)

[libsimepleio Class](#)

[IO.Bindings Namespace](#)

libsimpleioUDP4_open Method

Open a UDP socket.

Namespace: [IO.Bindings](#)

Assembly: libsimepleio (in libsimepleio.dll) Version: 2.2021.332.1

◀ Syntax

C# VB F#

[Copy](#)

```
public static void UDP4_open(
    int host,
    int port,
    out int fd,
    out int error
)
```

Parameters

host

Type: [SystemInt32](#)

IPv4 host address.

port

Type: [SystemInt32](#)

UDP port number.

fd

Type: [SystemInt32](#)

File descriptor.

error

Type: [SystemInt32](#)

Error code. Zero upon success or an [errno](#) value upon failure.

↳ See Also

Reference

[libsimpleio Class](#)

[IO.Bindings Namespace](#)

libsimpleioUDP4_receive Method

Receive a UDP datagram.

Namespace: [IO.Bindings](#)

Assembly: libsimepleio (in libsimepleio.dll) Version: 2.2021.332.1

◀ Syntax

C# VB F#

[Copy](#)

```
public static void UDP4_receive(
    int fd,
    out int host,
    out int port,
    byte[] buf,
    int bufsize,
    int flags,
    out int count,
    out int error
)
```

Parameters

fd

Type: [SystemInt32](#)

File descriptor.

host

Type: [SystemInt32](#)

Source IPv4 host address.

port

Type: [SystemInt32](#)

Source UDP port number.

buf

Type: [SystemByte](#)

Destination buffer.

bufsize

Type: [SystemInt32](#)

Destination buffer size.

flags

Type: [SystemInt32](#)

Flags for the Linux `recvfrom()` system call.

count

Type: [SystemInt32](#)

Number of bytes actually received.

error

Type: [SystemInt32](#)

Error code. Zero upon success or an `errno` value upon failure.

See Also

Reference

[libsimpleio Class](#)

[IO.Bindings Namespace](#)

libsimpleioUDP4_send Method

Send a UDP datagram.

Namespace: [IO.Bindings](#)

Assembly: libsimepleio (in libsimepleio.dll) Version: 2.2021.332.1

◀ Syntax

C# VB F#

[Copy](#)

```
public static void UDP4_send(
    int fd,
    int host,
    int port,
    byte[] buf,
    int bufsize,
    int flags,
    out int count,
    out int error
)
```

Parameters

fd

Type: [SystemInt32](#)

File descriptor.

host

Type: [SystemInt32](#)

Destination IPv4 host address.

port

Type: [SystemInt32](#)

Destination UDP port number.

buf

Type: [SystemByte](#)

Source buffer.

bufsize

Type: [SystemInt32](#)

Source buffer size.

flags

Type: [SystemInt32](#)

Flags for the Linux `sendto()` system call.

count

Type: [SystemInt32](#)

Number of bytes actually sent.

error

Type: [SystemInt32](#)

Error code. Zero upon success or an `errno` value upon failure.

See Also

Reference

[libsimpleio Class](#)

[IO.Bindings Namespace](#)

libsimpleioWATCHDOG_close Method

Close a Linux watchdog timer device.

Namespace: [IO.Bindings](#)

Assembly: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

◀ Syntax

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

[Copy](#)

```
public static void WATCHDOG_close(
    int fd,
    out int error
)
```

Parameters

fd

Type: [SystemInt32](#)

File descriptor.

error

Type: [SystemInt32](#)

Error code. Zero upon success or an [errno](#) value upon failure.

◀ See Also

[Reference](#)

[libsimpleio Class](#)

[IO.Bindings Namespace](#)

libsimpleioWATCHDOG_get_timeout Method

Query a Linux watchdog timer device.

Namespace: [IO.Bindings](#)

Assembly: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

◀ Syntax

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

[Copy](#)

```
public static void WATCHDOG_get_timeout(
    int fd,
    out int timeout,
    out int error
)
```

Parameters

fd

Type: [SystemInt32](#)

File descriptor.

timeout

Type: [SystemInt32](#)

Timeout period in seconds.

error

Type: [SystemInt32](#)

Error code. Zero upon success or an [errno](#) value upon failure.

◀ See Also

Reference

[libsimpleio Class](#)
[IO.Bindings Namespace](#)

libsimpleioWATCHDOG_kick Method

Reset the watchdog timer.

Namespace: [IO.Bindings](#)

Assembly: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

« Syntax

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

[Copy](#)

```
public static void WATCHDOG_kick(
    int fd,
    out int error
)
```

Parameters

fd

Type: [SystemInt32](#)

File descriptor.

error

Type: [SystemInt32](#)

Error code. Zero upon success or an [errno](#) value upon failure.

« See Also

[Reference](#)

[libsimpleio Class](#)

[IO.Bindings Namespace](#)

libsimpleioWATCHDOG_open Method

Open a Linux watchdog timer device.

Namespace: [IO.Bindings](#)

Assembly: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

◀ Syntax

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

[Copy](#)

```
public static void WATCHDOG_open(
    string devname,
    out int fd,
    out int error
)
```

Parameters

devname

Type: [SystemString](#)

Device node name.

fd

Type: [SystemInt32](#)

File descriptor.

error

Type: [SystemInt32](#)

Error code. Zero upon success or an [errno](#) value upon failure.

◀ See Also

Reference

[libsimpleio Class](#)
[IO.Bindings Namespace](#)

libsimpleioWATCHDOG_set_timeout Method

Change the watchdog timer period.

Namespace: [IO.Bindings](#)

Assembly: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

◀ Syntax

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

[Copy](#)

```
public static void WATCHDOG_set_timeout(
    int fd,
    int newtimeout,
    out int timeout,
    out int error
)
```

Parameters

fd

Type: [SystemInt32](#)

File descriptor.

newtimeout

Type: [SystemInt32](#)

Requested timeout period in seconds.

timeout

Type: [SystemInt32](#)

Actual timeout period in seconds.

error

Type: [SystemInt32](#)

Error code. Zero upon success or an [errno](#) value upon failure.

Remarks

Not all platforms allow changing the timeout period. Some platforms may not allow *increasing* the period.

See Also

Reference

[libsimpleio Class](#)

[IO.Bindings Namespace](#)

libsimpleio Fields

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

Fields

| Name | Description |
|---|--|
| GPIO_DIRECTION_INPUT | Input data direction. |
| GPIO_DIRECTION_OUTPUT | Out data direction. |
| GPIO_DRIVER_OPENDRAIN | Open drain (sink only) output driver. |
| GPIO_DRIVER_OPENSOURCE | Open source (source only) output driver |
| GPIO_DRIVER_PUSHPULL | Push-pull (source and sink) output driver. |
| GPIO_EDGE_BOTH | Interrupt on both edges. |
| GPIO_EDGE_FALLING | Interrupt on falling edge. |
| GPIO_EDGE_NONE | Interrupts are disabled. |
| GPIO_EDGE_RISING | Interrupt on rising edge. |

| | | |
|------------|---|---|
| • s | <code>GPIO_EVENT_REQUEST_BOTH</code> | Enable GPIO input interrupt on both edges. |
| • s | <code>GPIO_EVENT_REQUEST_FALLING</code> | Enable GPIO input interrupt on falling edge. |
| • s | <code>GPIO_EVENT_REQUEST_NONE</code> | Disable GPIO input interrupt. |
| • s | <code>GPIO_EVENT_REQUEST_RISING</code> | Enable GPIO input interrupt on rising edge. |
| • s | <code>GPIO_LINE_INFO_ACTIVE_LOW</code> | GPIO line is configured as active low (inverted). |
| • s | <code>GPIO_LINE_INFO_KERNEL</code> | GPIO line is being used by the kernel. |
| • s | <code>GPIO_LINE_INFO_OPEN_DRAIN</code> | GPIO line is configured as open drain (current sink only). |
| • s | <code>GPIO_LINE_INFO_OPEN_SOURCE</code> | GPIO line is configured as open source (current source only). |
| • s | <code>GPIO_LINE_INFO_OUTPUT</code> | GPIO line is configured as an output. |

| | | |
|------------|--|--|
| • s | <code>GPIO_LINE_REQUEST_ACTIVE_HIGH</code> | Select GPIO line polarity active high (normal). |
| • s | <code>GPIO_LINE_REQUEST_ACTIVE_LOW</code> | Select GPIO line polarity active low (inverted). |
| • s | <code>GPIO_LINE_REQUEST_INPUT</code> | Select GPIO line direction input. |
| • s | <code>GPIO_LINE_REQUEST_OPEN_DRAIN</code> | Select GPIO line driver open drain (current sink only). |
| • s | <code>GPIO_LINE_REQUEST_OPEN_SOURCE</code> | Select GPIO line driver open source (current source only). |
| • s | <code>GPIO_LINE_REQUEST_OUTPUT</code> | Select GPIO line direction output. |
| • s | <code>GPIO_LINE_REQUEST_PUSH_PULL</code> | Select GPIO line driver push-pull (current source and sink). |
| • s | <code>GPIO_POLARITY_ACTIVEHIGH</code> | Active high (normal) polarity. |
| • s | <code>GPIO_POLARITY_ACTIVELOW</code> | Active low (inverted) polarity. |
| • s | <code>INADDR_ANY</code> | IPv4 address for binding to all network interfaces. |

| | | |
|------------|------------------|---|
| ♦ s | INADDR_BROADCAST | IPv4 broadcast address. |
| ♦ s | INADDR_LOOPBACK | IPv4 address for binding to the loopback interface (aka <code>localhost</code>). |
| ♦ s | LOG_ALERT | Action must be taken immediately. |
| ♦ s | LOG_AUTH | Security/authorization messages. |
| ♦ s | LOG_AUTHPRIV | Securit/authorization messages. |
| ♦ s | LOG_CONS | Write directly to the system console if there is an error while sending to the system logger. |
| ♦ s | LOG_CRIT | Critical condition. |
| ♦ s | LOG_CRON | <code>cron</code> daemon messages. |
| ♦ s | LOG_DAEMON | System daemons. |
| ♦ s | LOG_DEBUG | Debug message. |
| ♦ s | LOG_EMERG | System is unusable. |
| ♦ s | LOG_ERR | Error condition. |

| | | |
|-----|------------|-------------------------|
| ♦ S | LOG_FTP | FTP daemon messages. |
| ♦ S | LOG_INFO | Informational message. |
| ♦ S | LOG_KERN | Kernel messages. |
| ♦ S | LOG_LOCAL0 | Reserved for local use. |
| ♦ S | LOG_LOCAL1 | Reserved for local use. |
| ♦ S | LOG_LOCAL2 | Reserved for local use. |
| ♦ S | LOG_LOCAL3 | Reserved for local use. |
| ♦ S | LOG_LOCAL4 | Reserved for local use. |
| ♦ S | LOG_LOCAL5 | Reserved for local use. |
| ♦ S | LOG_LOCAL6 | Reserved for local use. |
| ♦ S | LOG_LOCAL7 | Reserved for local use. |
| ♦ S | LOG_LPR | Line printer subsystem |
| ♦ S | LOG_MAIL | Mail system. |

| | | |
|------------|------------------------------|---|
| ♦ s | LOG_NDELAY | Open the connection immediately. Do not wait until <code>syslog()</code> is called for the first time. |
| ♦ s | LOG_NEWS | Network news subsystem |
| ♦ s | LOG_NOTICE | Normal but significant condition. |
| ♦ s | LOG_NOWAIT | Don't wait for child processes that may have been created while logging the message. (Not applicable to <code>glibc</code> .) |
| ♦ s | LOG_ODELAY | Do not open the connection immediately. Wait until <code>syslog()</code> is called for the first time. |
| ♦ s | LOG_PERROR | Also log the message to <code>stderr</code> . |
| ♦ s | LOG_PID | Include the caller's PID (process ID) with each message. |
| ♦ s | LOG_PROGNAME | Use the program name for the identity |

string.

| | | |
|------------|-------------------------------|--|
| ♦ s | LOG_SYSLOG | Messages generated internally by syslogd |
| ♦ s | LOG_USER | Random user-level messages. |
| ♦ s | LOG_UUCP | UUCP subsystem |
| ♦ s | LOG_WARNING | Warning condition. |
| ♦ s | MSG_DONTROUTE | Don't use a gateway to send out the packet, send to hosts only on directly connected networks. |
| ♦ s | MSG_DONTWAIT | Enables nonblocking operation; if the operation would block, EAGAIN or EWOULDBLOCK is returned. |
| ♦ s | MSG_MORE | The caller has more data to send. This flag informs the kernel to package all of the data sent in calls with this flag set into a single datagram which is transmitted only when a call is |

performed that does not specify this flag.

| | | |
|------------|-------------------------|--|
| ♦ s | POLLERR | An error occurred. |
| ♦ s | POLLHUP | Peer closed connection. |
| ♦ s | POLLIN | There is data to read. |
| ♦ s | POLLNVAL | File descriptor is invalid. |
| ♦ s | POLLOUT | Writing is now possible. |
| ♦ s | POLLPRI | There is urgent data to read. |
| ♦ s | PWM_POLARITY_ACTIVEHIGH | Configure the PWM output as active high (normal). |
| ♦ s | PWM_POLARITY_ACTIVELOW | Configure the PWM output as active low (inverted). |
| ♦ s | SERIAL_PARITY_EVEN | Request even parity checking. |
| ♦ s | SERIAL_PARITY_NONE | Disable parity checking. |
| ♦ s | SERIAL_PARITY_ODD | Request odd parity checking. |

[SPI_AUTO_CS](#)

Use hardware slave select.

[Top](#)

◀ See Also

Reference

[libsimpleio Class](#)

[IO.Bindings Namespace](#)

libsimpleioGPIO_DIRECTION_INPUT Field

Input data direction.

Namespace: [IO.Bindings](#)

Assembly: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

◀ Syntax

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

[Copy](#)

```
public const int GPIO_DIRECTION_INPUT = 0
```

Field Value

Type: [Int32](#)

◀ See Also

Reference

[libsimpleio Class](#)

[IO.Bindings Namespace](#)

libsimpleioGPIO_DIRECTION_OUTPUT Field

Out data direction.

Namespace: [IO.Bindings](#)

Assembly: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

▲ Syntax

C# VB F#

[Copy](#)

```
public const int GPIO_DIRECTION_OUTPUT = 1
```

Field Value

Type: [Int32](#)

▲ See Also

Reference

[libsimpleio Class](#)

[IO.Bindings Namespace](#)

libsimpleioGPIO_DRIVER_OPENDRAIN Field

Open drain (sink only) output driver.

Namespace: [IO.Bindings](#)

Assembly: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

▪ Syntax

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

[Copy](#)

```
public const int GPIO_DRIVER_OPENDRAIN = 1
```

Field Value

Type: [Int32](#)

▪ See Also

Reference

[libsimpleio Class](#)

[IO.Bindings Namespace](#)

libsimpleioGPIO_DRIVER_OPENSOURCE Field

Open source (source only) output driver

Namespace: [IO.Bindings](#)

Assembly: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

▪ Syntax

C# VB F#

[Copy](#)

```
public const int GPIO_DRIVER_OPENSOURCE = 2
```

Field Value

Type: [Int32](#)

▪ See Also

Reference

[libsimpleio Class](#)
[IO.Bindings Namespace](#)

libsimpleioGPIO_DRIVER_PUSH_PULL Field

Push-pull (source and sink) output driver.

Namespace: [IO.Bindings](#)

Assembly: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

◀ Syntax

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

[Copy](#)

```
public const int GPIO_DRIVER_PUSH_PULL = 0
```

Field Value

Type: [Int32](#)

◀ See Also

Reference

[libsimpleio Class](#)

[IO.Bindings Namespace](#)

libsimpleioGPIO_EDGE_BOTH Field

Interrupt on both edges.

Namespace: [IO.Bindings](#)

Assembly: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

▪ Syntax

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

[Copy](#)

```
public const int GPIO_EDGE_BOTH = 3
```

Field Value

Type: [Int32](#)

▪ See Also

Reference

[libsimpleio Class](#)

[IO.Bindings Namespace](#)

libsimpleioGPIO_EDGE_FALLING Field

Interrupt on falling edge.

Namespace: [IO.Bindings](#)

Assembly: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

◀ Syntax

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

[Copy](#)

```
public const int GPIO_EDGE_FALLING = 2
```

Field Value

Type: [Int32](#)

◀ See Also

Reference

[libsimpleio Class](#)

[IO.Bindings Namespace](#)

libsimpleioGPIO_EDGE_NONE Field

Interrupts are disabled.

Namespace: [IO.Bindings](#)

Assembly: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

◀ Syntax

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

[Copy](#)

```
public const int GPIO_EDGE_NONE = 0
```

Field Value

Type: [Int32](#)

◀ See Also

Reference

[libsimpleio Class](#)
[IO.Bindings Namespace](#)

libsimpleioGPIO_EDGE_RISING Field

Interrupt on rising edge.

Namespace: [IO.Bindings](#)

Assembly: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

▪ Syntax

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

[Copy](#)

```
public const int GPIO_EDGE_RISING = 1
```

Field Value

Type: [Int32](#)

▪ See Also

Reference

[libsimpleio Class](#)

[IO.Bindings Namespace](#)

libsimpleioGPIO_EVENT_REQUEST_BOTH Field

Enable GPIO input interrupt on both edges.

Namespace: [IO.Bindings](#)

Assembly: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

▪ Syntax

C# VB F#

[Copy](#)

```
public const int GPIO_EVENT_REQUEST_BOTH = 3
```

Field Value

Type: [Int32](#)

▪ See Also

Reference

[libsimpleio Class](#)

[IO.Bindings Namespace](#)

libsimpleioGPIO_EVENT_REQUEST_FALLING Field

Enable GPIO input interrupt on falling edge.

Namespace: [IO.Bindings](#)

Assembly: libsimepio (in libsimepio.dll) Version: 2.2021.332.1

▪ Syntax

C# VB F#

[Copy](#)

```
public const int GPIO_EVENT_REQUEST_FALLING = 2
```

Field Value

Type: [Int32](#)

▪ See Also

Reference

[libsimepio Class](#)

[IO.Bindings Namespace](#)

libsimpleioGPIO_EVENT_REQUEST_NONE Field

Disable GPIO input interrupt.

Namespace: [IO.Bindings](#)

Assembly: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

▪ Syntax

C# VB F#

[Copy](#)

```
public const int GPIO_EVENT_REQUEST_NONE = 0
```

Field Value

Type: [Int32](#)

▪ See Also

Reference

[libsimpleio Class](#)

[IO.Bindings Namespace](#)

libsimpleioGPIO_EVENT_REQUEST_RISING Field

Enable GPIO input interrupt on rising edge.

Namespace: [IO.Bindings](#)

Assembly: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

▲ Syntax

C# VB F#

[Copy](#)

```
public const int GPIO_EVENT_REQUEST_RISING = 1
```

Field Value

Type: [Int32](#)

▲ See Also

Reference

[libsimpleio Class](#)

[IO.Bindings Namespace](#)

libsimpleioGPIO_LINE_INFO_ACTIVE_LOW Field

GPIO line is configured as active low (inverted).

Namespace: [IO.Bindings](#)

Assembly: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

▪ Syntax

C# VB F#

[Copy](#)

```
public const int GPIO_LINE_INFO_ACTIVE_LOW = 4
```

Field Value

Type: [Int32](#)

▪ See Also

Reference

[libsimpleio Class](#)

[IO.Bindings Namespace](#)

libsimpleioGPIO_LINE_INFO_KERNEL Field

GPIO line is being used by the kernel.

Namespace: [IO.Bindings](#)

Assembly: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

▪ Syntax

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

[Copy](#)

```
public const int GPIO_LINE_INFO_KERNEL = 1
```

Field Value

Type: [Int32](#)

▪ See Also

Reference

[libsimpleio Class](#)

[IO.Bindings Namespace](#)

libsimpleioGPIO_LINE_INFO_OPEN_DRAIN Field

GPIO line is configured as open drain (current sink only).

Namespace: [IO.Bindings](#)

Assembly: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

▲ Syntax

C# VB F#

[Copy](#)

```
public const int GPIO_LINE_INFO_OPEN_DRAIN = 8
```

Field Value

Type: [Int32](#)

▲ See Also

Reference

[libsimpleio Class](#)

[IO.Bindings Namespace](#)

libsimpleioGPIO_LINE_INFO_OPEN_SOURCE Field

GPIO line is configured as open source (current source only).

Namespace: [IO.Bindings](#)

Assembly: libsimepio (in libsimepio.dll) Version: 2.2021.332.1

▪ Syntax

C# VB F#

Copy

```
public const int GPIO_LINE_INFO_OPEN_SOURCE = 16
```

Field Value

Type: [Int32](#)

▪ See Also

Reference

[libsimepio Class](#)

[IO.Bindings Namespace](#)

libsimpleioGPIO_LINE_INFO_OUTPUT Field

GPIO line is configured as an output.

Namespace: [IO.Bindings](#)

Assembly: libsimepio (in libsimepio.dll) Version: 2.2021.332.1

« Syntax

C# VB F#

[Copy](#)

```
public const int GPIO_LINE_INFO_OUTPUT = 2
```

Field Value

Type: [Int32](#)

« See Also

Reference

[libsimepio Class](#)

[IO.Bindings Namespace](#)

libsimpleioGPIO_LINE_REQUEST_ACTIVE_HIGH Field

Select GPIO line polarity active high (normal).

Namespace: [IO.Bindings](#)

Assembly: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

▲ Syntax

C# VB F#

[Copy](#)

```
public const int GPIO_LINE_REQUEST_ACTIVE_HIGH = 0
```

Field Value

Type: [Int32](#)

▲ See Also

Reference

[libsimpleio Class](#)

[IO.Bindings Namespace](#)

libsimpleioGPIO_LINE_REQUEST_ACTIVE_LOW Field

Select GPIO line polarity active low (inverted).

Namespace: [IO.Bindings](#)

Assembly: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

◀ Syntax

C# VB F#

[Copy](#)

```
public const int GPIO_LINE_REQUEST_ACTIVE_LOW = 4
```

Field Value

Type: [Int32](#)

◀ See Also

Reference

[libsimpleio Class](#)

[IO.Bindings Namespace](#)

libsimpleioGPIO_LINE_REQUEST_INPUT Field

Select GPIO line direction input.

Namespace: [IO.Bindings](#)

Assembly: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

▲ Syntax

C# VB F#

[Copy](#)

```
public const int GPIO_LINE_REQUEST_INPUT = 1
```

Field Value

Type: [Int32](#)

▲ See Also

Reference

[libsimpleio Class](#)

[IO.Bindings Namespace](#)

libsimpleioGPIO_LINE_REQUEST_OPEN_DRAIN Field

Select GPIO line driver open drain (current sink only).

Namespace: [IO.Bindings](#)

Assembly: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

▲ Syntax

C# VB F#

Copy

```
public const int GPIO_LINE_REQUEST_OPEN_DRAIN = 8
```

Field Value

Type: [Int32](#)

▲ See Also

Reference

[libsimpleio Class](#)

[IO.Bindings Namespace](#)

libsimpleioGPIO_LINE_REQUEST_OPEN_SOURCE Field

Select GPIO line driver open source (current source only).

Namespace: [IO.Bindings](#)

Assembly: libsimepio (in libsimepio.dll) Version: 2.2021.332.1

◀ Syntax

C# VB F#

Copy

```
public const int GPIO_LINE_REQUEST_OPEN_SOURCE = 16
```

Field Value

Type: [Int32](#)

◀ See Also

Reference

[libsimepio Class](#)

[IO.Bindings Namespace](#)

libsimpleioGPIO_LINE_REQUEST_OUTPUT Field

Select GPIO line direction output.

Namespace: [IO.Bindings](#)

Assembly: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

▪ Syntax

C# VB F#

[Copy](#)

```
public const int GPIO_LINE_REQUEST_OUTPUT = 2
```

Field Value

Type: [Int32](#)

▪ See Also

Reference

[libsimpleio Class](#)

[IO.Bindings Namespace](#)

libsimpleioGPIO_LINE_REQUEST_PUSH_PULL Field

Select GPIO line driver push-pull (current source and sink).

Namespace: [IO.Bindings](#)

Assembly: libsimepio (in libsimepio.dll) Version: 2.2021.332.1

◀ Syntax

C# VB F#

Copy

```
public const int GPIO_LINE_REQUEST_PUSH_PULL = 0
```

Field Value

Type: [Int32](#)

◀ See Also

[Reference](#)

[libsimepio Class](#)

[IO.Bindings Namespace](#)

libsimpleioGPIO_POLARITY_ACTIVEHIGH Field

Active high (normal) polarity.

Namespace: [IO.Bindings](#)

Assembly: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

▪ Syntax

C# VB F#

[Copy](#)

```
public const int GPIO_POLARITY_ACTIVEHIGH = 1
```

Field Value

Type: [Int32](#)

▪ See Also

Reference

[libsimpleio Class](#)

[IO.Bindings Namespace](#)

libsimpleioGPIO_POLARITY_ACTIVELOW Field

Active low (inverted) polarity.

Namespace: [IO.Bindings](#)

Assembly: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

▪ Syntax

C# VB F#

Copy

```
public const int GPIO_POLARITY_ACTIVELOW = 0
```

Field Value

Type: [Int32](#)

▪ See Also

Reference

[libsimpleio Class](#)

[IO.Bindings Namespace](#)

libsimpleioINADDR_ANY Field

IPv4 address for binding to all network interfaces.

Namespace: [IO.Bindings](#)

Assembly: libsimepleio (in libsimepleio.dll) Version: 2.2021.332.1

◀ Syntax

C# VB F#

[Copy](#)

```
public const int INADDR_ANY = 0
```

Field Value

Type: [Int32](#)

◀ See Also

Reference

[libsimepleio Class](#)

[IO.Bindings Namespace](#)

libsimpleioINADDR_BROADCAST Field

IPv4 broadcast address.

Namespace: [IO.Bindings](#)

Assembly: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

▪ Syntax

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

[Copy](#)

```
public const int INADDR_BROADCAST = -1
```

Field Value

Type: [Int32](#)

▪ See Also

Reference

[libsimpleio Class](#)

[IO.Bindings Namespace](#)

libsimpleioINADDR_LOOPBACK Field

IPv4 address for binding to the loopback interface (aka `localhost`).

Namespace: [IO.Bindings](#)

Assembly: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

▪ Syntax

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

[Copy](#)

```
public const int INADDR_LOOPBACK = 2130706433
```

Field Value

Type: [Int32](#)

▪ See Also

Reference

[libsimpleio Class](#)

[IO.Bindings Namespace](#)

libsimpleioLOG_ALERT Field

Action must be taken immediately.

Namespace: [IO.Bindings](#)

Assembly: libsimepleio (in libsimepleio.dll) Version: 2.2021.332.1

◀ Syntax

C# VB F#

[Copy](#)

```
public const int LOG_ALERT = 1
```

Field Value

Type: [Int32](#)

◀ See Also

Reference

[libsimepleio Class](#)

[IO.Bindings Namespace](#)

libsimpleioLOG_AUTH Field

Security/authorization messages.

Namespace: [IO.Bindings](#)

Assembly: libsimepleio (in libsimepleio.dll) Version: 2.2021.332.1

◀ Syntax

C# VB F#

[Copy](#)

```
public const int LOG_AUTH = 32
```

Field Value

Type: [Int32](#)

◀ See Also

Reference

[libsimepleio Class](#)

[IO.Bindings Namespace](#)

libsimpleioLOG_AUTHPRIV Field

Securit/authorization messages.

Namespace: [IO.Bindings](#)

Assembly: libsimepleio (in libsimepleio.dll) Version: 2.2021.332.1

◀ Syntax

C# VB F#

[Copy](#)

```
public const int LOG_AUTHPRIV = 80
```

Field Value

Type: [Int32](#)

◀ See Also

Reference

[libsimepleio Class](#)

[IO.Bindings Namespace](#)

libsimpleioLOG_CONS Field

Write directly to the system console if there is an error while sending to the system logger.

Namespace: [IO.Bindings](#)

Assembly: libsimepleio (in libsimepleio.dll) Version: 2.2021.332.1

◀ Syntax

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

[Copy](#)

```
public const int LOG_CONS = 2
```

Field Value

Type: [Int32](#)

◀ See Also

Reference

[libsimepleio Class](#)

[IO.Bindings Namespace](#)

libsimpleioLOG_CRIT Field

Critical condition.

Namespace: [IO.Bindings](#)

Assembly: libsimepleio (in libsimepleio.dll) Version: 2.2021.332.1

◀ Syntax

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

[Copy](#)

```
public const int LOG_CRIT = 2
```

Field Value

Type: [Int32](#)

◀ See Also

Reference

[libsimepleio Class](#)

[IO.Bindings Namespace](#)

libsimpleioLOG_CRON Field

[cron](#) daemon messages.

Namespace: [IO.Bindings](#)

Assembly: libsimepleio (in libsimepleio.dll) Version: 2.2021.332.1

◀ Syntax

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

[Copy](#)

```
public const int LOG_CRON = 72
```

Field Value

Type: [Int32](#)

◀ See Also

Reference

[libsimepleio Class](#)

[IO.Bindings Namespace](#)

libsimpleioLOG_DAEMON Field

System daemons.

Namespace: [IO.Bindings](#)

Assembly: libsimepleio (in libsimepleio.dll) Version: 2.2021.332.1

◀ Syntax

C# VB F#

[Copy](#)

```
public const int LOG_DAEMON = 24
```

Field Value

Type: [Int32](#)

◀ See Also

Reference

[libsimepleio Class](#)

[IO.Bindings Namespace](#)

libsimpleioLOG_DEBUG Field

Debug message.

Namespace: [IO.Bindings](#)

Assembly: libsimepleio (in libsimepleio.dll) Version: 2.2021.332.1

◀ Syntax

C# VB F#

[Copy](#)

```
public const int LOG_DEBUG = 7
```

Field Value

Type: [Int32](#)

◀ See Also

Reference

[libsimepleio Class](#)

[IO.Bindings Namespace](#)

libsimpleioLOG_EMERG Field

System is unusable.

Namespace: [IO.Bindings](#)

Assembly: libsimepleio (in libsimepleio.dll) Version: 2.2021.332.1

◀ Syntax

C# VB F#

[Copy](#)

```
public const int LOG_EMERG = 0
```

Field Value

Type: [Int32](#)

◀ See Also

Reference

[libsimepleio Class](#)

[IO.Bindings Namespace](#)

libsimpleioLOG_ERR Field

Error condition.

Namespace: [IO.Bindings](#)

Assembly: libsimepleio (in libsimepleio.dll) Version: 2.2021.332.1

◀ Syntax

C# VB F#

[Copy](#)

```
public const int LOG_ERR = 3
```

Field Value

Type: [Int32](#)

◀ See Also

Reference

[libsimepleio Class](#)

[IO.Bindings Namespace](#)

libsimpleioLOG_FTP Field

[FTP](#) daemon messages.

Namespace: [IO.Bindings](#)

Assembly: libsimepleio (in libsimepleio.dll) Version: 2.2021.332.1

◀ Syntax

C# VB F#

[Copy](#)

```
public const int LOG_FTP = 88
```

Field Value

Type: [Int32](#)

◀ See Also

Reference

[libsimepleio Class](#)

[IO.Bindings Namespace](#)

libsimpleioLOG_INFO Field

Informational message.

Namespace: [IO.Bindings](#)

Assembly: libsimepleio (in libsimepleio.dll) Version: 2.2021.332.1

◀ Syntax

C# VB F#

[Copy](#)

```
public const int LOG_INFO = 6
```

Field Value

Type: [Int32](#)

◀ See Also

Reference

[libsimepleio Class](#)

[IO.Bindings Namespace](#)

libsimpleioLOG_KERN Field

Kernel messages.

Namespace: [IO.Bindings](#)

Assembly: libsimepleio (in libsimepleio.dll) Version: 2.2021.332.1

◀ Syntax

C# VB F#

[Copy](#)

```
public const int LOG_KERN = 0
```

Field Value

Type: [Int32](#)

◀ See Also

Reference

[libsimepleio Class](#)

[IO.Bindings Namespace](#)

libsimpleioLOG_LOCAL0 Field

Reserved for local use.

Namespace: [IO.Bindings](#)

Assembly: libsimepleio (in libsimepleio.dll) Version: 2.2021.332.1

◀ Syntax

C# VB F#

[Copy](#)

```
public const int LOG_LOCAL0 = 128
```

Field Value

Type: [Int32](#)

◀ See Also

Reference

[libsimepleio Class](#)

[IO.Bindings Namespace](#)

libsimpleioLOG_LOCAL1 Field

Reserved for local use.

Namespace: [IO.Bindings](#)

Assembly: libsimepleio (in libsimepleio.dll) Version: 2.2021.332.1

◀ Syntax

C# VB F#

[Copy](#)

```
public const int LOG_LOCAL1 = 136
```

Field Value

Type: [Int32](#)

◀ See Also

Reference

[libsimepleio Class](#)

[IO.Bindings Namespace](#)

libsimpleioLOG_LOCAL2 Field

Reserved for local use.

Namespace: [IO.Bindings](#)

Assembly: libsimepleio (in libsimepleio.dll) Version: 2.2021.332.1

◀ Syntax

C# VB F#

[Copy](#)

```
public const int LOG_LOCAL2 = 144
```

Field Value

Type: [Int32](#)

◀ See Also

Reference

[libsimepleio Class](#)

[IO.Bindings Namespace](#)

libsimpleioLOG_LOCAL3 Field

Reserved for local use.

Namespace: [IO.Bindings](#)

Assembly: libsimepleio (in libsimepleio.dll) Version: 2.2021.332.1

« Syntax

C# VB F#

[Copy](#)

```
public const int LOG_LOCAL3 = 152
```

Field Value

Type: [Int32](#)

« See Also

Reference

[libsimepleio Class](#)

[IO.Bindings Namespace](#)

libsimpleioLOG_LOCAL4 Field

Reserved for local use.

Namespace: [IO.Bindings](#)

Assembly: libsimepleio (in libsimepleio.dll) Version: 2.2021.332.1

◀ Syntax

C# VB F#

[Copy](#)

```
public const int LOG_LOCAL4 = 160
```

Field Value

Type: [Int32](#)

◀ See Also

Reference

[libsimepleio Class](#)

[IO.Bindings Namespace](#)

libsimpleioLOG_LOCAL5 Field

Reserved for local use.

Namespace: [IO.Bindings](#)

Assembly: libsimepleio (in libsimepleio.dll) Version: 2.2021.332.1

◀ Syntax

C# VB F#

[Copy](#)

```
public const int LOG_LOCAL5 = 168
```

Field Value

Type: [Int32](#)

◀ See Also

Reference

[libsimepleio Class](#)

[IO.Bindings Namespace](#)

libsimpleioLOG_LOCAL6 Field

Reserved for local use.

Namespace: [IO.Bindings](#)

Assembly: libsimepleio (in libsimepleio.dll) Version: 2.2021.332.1

◀ Syntax

C# VB F#

[Copy](#)

```
public const int LOG_LOCAL6 = 176
```

Field Value

Type: [Int32](#)

◀ See Also

Reference

[libsimepleio Class](#)

[IO.Bindings Namespace](#)

libsimpleioLOG_LOCAL7 Field

Reserved for local use.

Namespace: [IO.Bindings](#)

Assembly: libsimepleio (in libsimepleio.dll) Version: 2.2021.332.1

◀ Syntax

C# VB F#

[Copy](#)

```
public const int LOG_LOCAL7 = 184
```

Field Value

Type: [Int32](#)

◀ See Also

Reference

[libsimepleio Class](#)

[IO.Bindings Namespace](#)

libsimpleioLOG_LPR Field

Line printer subsystem

Namespace: [IO.Bindings](#)

Assembly: libsimepleio (in libsimepleio.dll) Version: 2.2021.332.1

◀ Syntax

C# VB F#

[Copy](#)

```
public const int LOG_LPR = 48
```

Field Value

Type: [Int32](#)

◀ See Also

Reference

[libsimepleio Class](#)

[IO.Bindings Namespace](#)

libsimpleioLOG_MAIL Field

Mail system.

Namespace: [IO.Bindings](#)

Assembly: libsimepleio (in libsimepleio.dll) Version: 2.2021.332.1

◀ Syntax

C# VB F#

[Copy](#)

```
public const int LOG_MAIL = 16
```

Field Value

Type: [Int32](#)

◀ See Also

Reference

[libsimepleio Class](#)

[IO.Bindings Namespace](#)

libsimpleioLOG_NDELAY Field

Open the connection immediately. Do not wait until [syslog\(\)](#) is called for the first time.

Namespace: [IO.Bindings](#)

Assembly: libsimepleio (in libsimepleio.dll) Version: 2.2021.332.1

◀ Syntax

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

[Copy](#)

```
public const int LOG_NDELAY = 8
```

Field Value

Type: [Int32](#)

◀ See Also

Reference

[libsimepleio Class](#)

[IO.Bindings Namespace](#)

libsimpleioLOG_NEWS Field

Network news subsystem

Namespace: [IO.Bindings](#)

Assembly: libsimepleio (in libsimepleio.dll) Version: 2.2021.332.1

◀ Syntax

C# VB F#

[Copy](#)

```
public const int LOG_NEWS = 56
```

Field Value

Type: [Int32](#)

◀ See Also

Reference

[libsimepleio Class](#)

[IO.Bindings Namespace](#)

libsimpleioLOG_NOTICE Field

Normal but significant condition.

Namespace: [IO.Bindings](#)

Assembly: libsimepleio (in libsimepleio.dll) Version: 2.2021.332.1

◀ Syntax

C# VB F#

[Copy](#)

```
public const int LOG_NOTICE = 5
```

Field Value

Type: [Int32](#)

◀ See Also

Reference

[libsimepleio Class](#)

[IO.Bindings Namespace](#)

libsimpleioLOG_NOWAIT Field

Don't wait for child processes that may have been created while logging the message. (Not applicable to [glibc](#).)

Namespace: [IO.Bindings](#)

Assembly: libsimepleio (in libsimepleio.dll) Version: 2.2021.332.1

◀ Syntax

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

[Copy](#)

```
public const int LOG_NOWAIT = 16
```

Field Value

Type: [Int32](#)

◀ See Also

Reference

[libsimepleio Class](#)

[IO.Bindings Namespace](#)

libsimpleioLOG_ODELAY Field

Do not open the connection immediately. Wait until [syslog\(\)](#) is called for the first time.

Namespace: [IO.Bindings](#)

Assembly: libsimepleio (in libsimepleio.dll) Version: 2.2021.332.1

◀ Syntax

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

[Copy](#)

```
public const int LOG_ODELAY = 4
```

Field Value

Type: [Int32](#)

◀ See Also

Reference

[libsimepleio Class](#)

[IO.Bindings Namespace](#)

libsimpleioLOG_PERROR Field

Also log the message to `stderr`.

Namespace: [IO.Bindings](#)

Assembly: libsimepleio (in libsimepleio.dll) Version: 2.2021.332.1

↳ Syntax

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

[Copy](#)

```
public const int LOG_PERROR = 32
```

Field Value

Type: [Int32](#)

↳ See Also

Reference

[libsimepleio Class](#)

[IO.Bindings Namespace](#)

libsimpleioLOG_PID Field

Include the caller's PID (process ID) with each message.

Namespace: [IO.Bindings](#)

Assembly: libsimepleio (in libsimepleio.dll) Version: 2.2021.332.1

↳ Syntax

C# VB F#

[Copy](#)

```
public const int LOG_PID = 1
```

Field Value

Type: [Int32](#)

↳ See Also

Reference

[libsimepleio Class](#)

[IO.Bindings Namespace](#)

libsimpleioLOG_PROGNAME Field

Use the program name for the identity string.

Namespace: [IO.Bindings](#)

Assembly: libsimepleio (in libsimepleio.dll) Version: 2.2021.332.1

◀ Syntax

C# VB F#

[Copy](#)

```
public const string LOG_PROGNAME = ""
```

Field Value

Type: [String](#)

◀ See Also

Reference

[libsimepleio Class](#)

[IO.Bindings Namespace](#)

libsimpleioLOG_SYSLOG Field

Messages generated internally by syslogd

Namespace: [IO.Bindings](#)

Assembly: libsimepleio (in libsimepleio.dll) Version: 2.2021.332.1

◀ Syntax

C# VB F#

[Copy](#)

```
public const int LOG_SYSLOG = 40
```

Field Value

Type: [Int32](#)

◀ See Also

Reference

[libsimepleio Class](#)

[IO.Bindings Namespace](#)

libsimpleioLOG_USER Field

Random user-level messages.

Namespace: [IO.Bindings](#)

Assembly: libsimepleio (in libsimepleio.dll) Version: 2.2021.332.1

◀ Syntax

C# VB F#

[Copy](#)

```
public const int LOG_USER = 8
```

Field Value

Type: [Int32](#)

◀ See Also

Reference

[libsimepleio Class](#)

[IO.Bindings Namespace](#)

libsimpleioLOG_UUCP Field

UUCP subsystem

Namespace: [IO.Bindings](#)

Assembly: libsimepleio (in libsimepleio.dll) Version: 2.2021.332.1

◀ Syntax

C# VB F#

[Copy](#)

```
public const int LOG_UUCP = 64
```

Field Value

Type: [Int32](#)

◀ See Also

Reference

[libsimepleio Class](#)

[IO.Bindings Namespace](#)

libsimpleioLOG_WARNING Field

Warning condition.

Namespace: [IO.Bindings](#)

Assembly: libsimepleio (in libsimepleio.dll) Version: 2.2021.332.1

◀ Syntax

C# VB F#

[Copy](#)

```
public const int LOG_WARNING = 4
```

Field Value

Type: [Int32](#)

◀ See Also

Reference

[libsimepleio Class](#)

[IO.Bindings Namespace](#)

libsimpleioMSG_DONTROUTE Field

Don't use a gateway to send out the packet, send to hosts only on directly connected networks.

Namespace: [IO.Bindings](#)

Assembly: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

↳ Syntax

C# VB F#

Copy

```
public const int MSG_DONTROUTE = 4
```

Field Value

Type: [Int32](#)

↳ See Also

Reference

[libsimpleio Class](#)

[IO.Bindings Namespace](#)

libsimpleioMSG_DONTWAIT Field

Enables nonblocking operation; if the operation would block, [EAGAIN](#) or [EWOULDBLOCK](#) is returned.

Namespace: [IO.Bindings](#)

Assembly: libsimepleio (in libsimepleio.dll) Version: 2.2021.332.1

◀ Syntax

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

[Copy](#)

```
public const int MSG_DONTWAIT = 64
```

Field Value

Type: [Int32](#)

◀ See Also

Reference

[libsimepleio Class](#)

[IO.Bindings Namespace](#)

libsimpleioMSG_MORE Field

The caller has more data to send. This flag informs the kernel to package all of the data sent in calls with this flag set into a single datagram which is transmitted only when a call is performed that does not specify this flag.

Namespace: [IO.Bindings](#)

Assembly: libsimepleio (in libsimepleio.dll) Version: 2.2021.332.1

↳ Syntax

C# VB F#

[Copy](#)

```
public const int MSG_MORE = 32768
```

Field Value

Type: [Int32](#)

↳ See Also

Reference

[libsimepleio Class](#)

[IO.Bindings Namespace](#)

libsimpleioPOLLERR Field

An error occurred.

Namespace: [IO.Bindings](#)

Assembly: libsimepleio (in libsimepleio.dll) Version: 2.2021.332.1

◀ Syntax

C# VB F#

[Copy](#)

```
public const int POLLERR = 8
```

Field Value

Type: [Int32](#)

◀ See Also

Reference

[libsimepleio Class](#)

[IO.Bindings Namespace](#)

libsimpleioPOLLHUP Field

Peer closed connection.

Namespace: [IO.Bindings](#)

Assembly: libsimepleio (in libsimepleio.dll) Version: 2.2021.332.1

◀ Syntax

C# VB F#

[Copy](#)

```
public const int POLLHUP = 16
```

Field Value

Type: [Int32](#)

◀ See Also

Reference

[libsimepleio Class](#)

[IO.Bindings Namespace](#)

libsimpleioPOLLIN Field

There is data to read.

Namespace: [IO.Bindings](#)

Assembly: libsimepleio (in libsimepleio.dll) Version: 2.2021.332.1

◀ Syntax

C# VB F#

[Copy](#)

```
public const int POLLIN = 1
```

Field Value

Type: [Int32](#)

◀ See Also

Reference

[libsimepleio Class](#)

[IO.Bindings Namespace](#)

libsimpleioPOLLNVAL Field

File descriptor is invalid.

Namespace: [IO.Bindings](#)

Assembly: libsimepleio (in libsimepleio.dll) Version: 2.2021.332.1

◀ Syntax

C# VB F#

[Copy](#)

```
public const int POLLNVAL = 32
```

Field Value

Type: [Int32](#)

◀ See Also

Reference

[libsimepleio Class](#)

[IO.Bindings Namespace](#)

libsimpleioPOLLOUT Field

Writing is now possible.

Namespace: [IO.Bindings](#)

Assembly: libsimepleio (in libsimepleio.dll) Version: 2.2021.332.1

↳ Syntax

C# VB F#

[Copy](#)

```
public const int POLLOUT = 4
```

Field Value

Type: [Int32](#)

↳ See Also

Reference

[libsimepleio Class](#)

[IO.Bindings Namespace](#)

libsimpleioPOLLPRI Field

There is urgent data to read.

Namespace: [IO.Bindings](#)

Assembly: libsimepleio (in libsimepleio.dll) Version: 2.2021.332.1

◀ Syntax

C# VB F#

[Copy](#)

```
public const int POLLPRI = 2
```

Field Value

Type: [Int32](#)

◀ See Also

Reference

[libsimepleio Class](#)

[IO.Bindings Namespace](#)

libsimpleioPWM_POLARITY_ACTIVEHIGH Field

Configure the PWM output as active high (normal).

Namespace: [IO.Bindings](#)

Assembly: libsimepleio (in libsimepleio.dll) Version: 2.2021.332.1

▪ Syntax

C# VB F#

Copy

```
public const int PWM_POLARITY_ACTIVEHIGH = 1
```

Field Value

Type: [Int32](#)

▪ See Also

Reference

[libsimepleio Class](#)

[IO.Bindings Namespace](#)

libsimpleioPWM_POLARITY_ACTIVELOW Field

Configure the PWM output as active low (inverted).

Namespace: [IO.Bindings](#)

Assembly: libsimepleio (in libsimepleio.dll) Version: 2.2021.332.1

▪ Syntax

C# VB F#

Copy

```
public const int PWM_POLARITY_ACTIVELOW = 0
```

Field Value

Type: [Int32](#)

▪ Remarks

Not all platforms support active low (inverted) PWM outputs.

▪ See Also

Reference

[libsimepleio Class](#)

[IO.Bindings Namespace](#)

libsimpleioSERIAL_PARITY EVEN Field

Request even parity checking.

Namespace: [IO.Bindings](#)

Assembly: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

▪ Syntax

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

[Copy](#)

```
public const int SERIAL_PARITY EVEN = 1
```

Field Value

Type: [Int32](#)

▪ See Also

Reference

[libsimpleio Class](#)

[IO.Bindings Namespace](#)

libsimpleioSERIAL_PARITY_NONE Field

Disable parity checking.

Namespace: [IO.Bindings](#)

Assembly: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

▪ Syntax

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

[Copy](#)

```
public const int SERIAL_PARITY_NONE = 0
```

Field Value

Type: [Int32](#)

▪ See Also

Reference

[libsimpleio Class](#)

[IO.Bindings Namespace](#)

libsimpleioSERIAL_PARITY_ODD Field

Request odd parity checking.

Namespace: [IO.Bindings](#)

Assembly: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

▪ Syntax

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

[Copy](#)

```
public const int SERIAL_PARITY_ODD = 2
```

Field Value

Type: [Int32](#)

▪ See Also

Reference

[libsimpleio Class](#)

[IO.Bindings Namespace](#)

libsimpleioSPI_AUTO_CS Field

Use hardware slave select.

Namespace: [IO.Bindings](#)

Assembly: libsimepleio (in libsimepleio.dll) Version: 2.2021.332.1

◀ Syntax

C# VB F#

[Copy](#)

```
public const int SPI_AUTO_CS = -1
```

Field Value

Type: [Int32](#)

◀ See Also

Reference

[libsimepleio Class](#)

[IO.Bindings Namespace](#)

IO.Devices.A4988 Namespace

A4988 Stepper Motor Controller Services.

◀ Classes

| | Class | Description |
|---|------------------------|--|
|  | Device | Encapsulates the A4988 Stepper Motor Controller. |

Device Class

Encapsulates the A4988 Stepper Motor Controller.

▪ Inheritance Hierarchy

[SystemObject](#) [IO.Devices.A4988Device](#)

Namespace: [IO.Devices.A4988](#)

Assembly: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

▪ Syntax

C# VB F#

[Copy](#)

```
public class Device : Output
```

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

▪ Constructors

| | Name | Description |
|---|------------------------|--|
|  | Device | Constructor for a single A4988 device. |

[Top](#)

▪ Properties

| | Name | Description |
|---|----------------------------------|--|
|  | StepsPerRotation | Read-only property returning the number of steps a stepper |

motor has.

[Top](#)

◀ Methods

| | Name | Description |
|---|-------------------------|---|
| ≡ | Disable | Disable the A4988 device. |
| ≡ | Enable | Enable the A4988 device. |
| ≡ | Move | Move the stepper motor a specified number of steps at a specified rate. |
| ≡ | Reset | Reset the A4988 device. |
| ≡ | Sleep | Put the A4988 device to sleep. |
| ≡ | Spin | Spin (i.e. continuous rotation) the stepper motor at a specified rate. |
| ≡ | Wakeup | Wake up the A4988 device. |

[Top](#)

◀ See Also

[Reference](#)

[IO.Devices.A4988 Namespace](#)

Device Constructor

Constructor for a single A4988 device.

Namespace: [IO.Devices.A4988](#)

Assembly: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

◀ Syntax

C# VB F#

[Copy](#)

```
public Device(  
    int StepsPerRotation,  
    Pin Step,  
    Pin Dir,  
    Pin Enable = null,  
    Pin Reset = null,  
    Pin Sleep = null  
)
```

Parameters

StepsPerRotation

Type: [SystemInt32](#)

The number of steps per rotation. This is a physical characteristic of the particular stepper motor being driven.

Step

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

GPIO pin object for the **STEP** signal.

Dir

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

GPIO pin object for the **DIR** signal.

Enable (Optional)

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

GPIO pin object for the `-ENABLE` signal.

Reset (Optional)

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

GPIO pin object for the `-RESET` signal.

Sleep (Optional)

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

GPIO pin object for the `-SLEEP` signal.

◀ See Also

Reference

[Device Class](#)

[IO.Devices.A4988 Namespace](#)

Device Properties

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

▪ Properties

| | Name | Description |
|--|----------------------------------|---|
| | StepsPerRotation | Read-only property returning the number of steps a stepper motor has. |

[Top](#)

▪ See Also

Reference

[Device Class](#)

[IO.Devices.A4988 Namespace](#)

DeviceStepsPerRotation Property

Read-only property returning the number of steps a stepper motor has.

Namespace: [IO.Devices.A4988](#)

Assembly: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

▪ Syntax

C# VB F#

[Copy](#)

```
public int StepsPerRotation { get; }
```

Property Value

Type: [Int32](#)

Implements

[OutputStepsPerRotation](#)

▪ See Also

[Reference](#)

[Device Class](#)

[IO.Devices.A4988 Namespace](#)

Device Methods

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

▪ Methods

| | Name | Description |
|--|-------------------------|---|
| | Disable | Disable the A4988 device. |
| | Enable | Enable the A4988 device. |
| | Move | Move the stepper motor a specified number of steps at a specified rate. |
| | Reset | Reset the A4988 device. |
| | Sleep | Put the A4988 device to sleep. |
| | Spin | Spin (i.e. continuous rotation) the stepper motor at a specified rate. |
| | Wakeup | Wake up the A4988 device. |

[Top](#)

▪ See Also

[Reference](#)

[Device Class](#)

[IO.Devices.A4988 Namespace](#)

DeviceDisable Method

Disable the A4988 device.

Namespace: [IO.Devices.A4988](#)

Assembly: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

◀ Syntax

C# VB F#

[Copy](#)

```
public void Disable()
```

◀ See Also

Reference

[Device Class](#)

[IO.Devices.A4988 Namespace](#)

DeviceEnable Method

Enable the A4988 device.

Namespace: [IO.Devices.A4988](#)

Assembly: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

▪ Syntax

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

[Copy](#)

```
public void Enable()
```

▪ See Also

[Reference](#)

[Device Class](#)

[IO.Devices.A4988 Namespace](#)

DeviceMove Method

Move the stepper motor a specified number of steps at a specified rate.

Namespace: [IO.Devices.A4988](#)

Assembly: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

◀ Syntax

C# VB F#

[Copy](#)

```
public void Move(  
    int steps,  
    float rate  
)
```

Parameters

steps

Type: [SystemInt32](#)

Number of steps to move. Negative values indicate reverse motion and positive values indicate forward motion. (The directions are nominal and depend on how the stepper motor coils are wired). Zero indicates the stepper motor should be stopped.

rate

Type: [SystemSingle](#)

The rate of motion, in steps per second. Negative values indicate reverse motion and positive values indicate forward motion. (The directions are nominal and depend on how the stepper motor coils are wired). Zero indicates the stepper motor should be stopped.

Implements
[OutputMove\(Int32, Single\)](#)

◀ Remarks

This implementation supports a maximum rate of 500 steps per second.

◀ See Also

Reference

[Device Class](#)

[IO.Devices.A4988 Namespace](#)

DeviceReset Method

Reset the A4988 device.

Namespace: [IO.Devices.A4988](#)

Assembly: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

◀ Syntax

C# VB F#

[Copy](#)

```
public void Reset()
```

◀ See Also

Reference

[Device Class](#)

[IO.Devices.A4988 Namespace](#)

DeviceSleep Method

Put the A4988 device to sleep.

Namespace: [IO.Devices.A4988](#)

Assembly: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

▪ Syntax

C# VB F#

[Copy](#)

```
public void Sleep()
```

▪ See Also

Reference

[Device Class](#)

[IO.Devices.A4988 Namespace](#)

DeviceSpin Method

Spin (i.e. continuous rotation) the stepper motor at a specified rate.

Namespace: [IO.Devices.A4988](#)

Assembly: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

► Syntax

C# VB F#

[Copy](#)

```
public void Spin(  
    float rate  
)
```

Parameters

rate

Type: [SystemSingle](#)

The rate of motion, in steps per second. Negative values indicate reverse motion and positive values indicate forward motion. (The directions are nominal and depend on how the stepper motor coils are wired). Zero indicates the stepper motor should be stopped.

Implements

[OutputSpin\(Single\)](#)

► Remarks

The A4988 stepper motor driver does not support continuous rotation and this method will always throw an exception.

See Also

[Reference](#)

[Device Class](#)

[IO.Devices.A4988 Namespace](#)

DeviceWakeup Method

Wake up the A4988 device.

Namespace: [IO.Devices.A4988](#)

Assembly: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

▀ Syntax

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

[Copy](#)

```
public void Wakeup()
```

▀ See Also

[Reference](#)

[Device Class](#)

[IO.Devices.A4988 Namespace](#)

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: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

▪ 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 |
|------------|--------------------------------|-----------------------------|
| ◆ S | ADC_Resolution | ADC resolution in bits. |
| ◆ S | DAC_Resolution | DAC resolution in bits. |
| ◆ S | MaxChannel | Maximum I/O channel number. |
| ◆ S | 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: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

▪ 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: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

◀ 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: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

◀ 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: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

◀ 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: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

◀ 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: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

◀ 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: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

« 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: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

↳ 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: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

◀ 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: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

◀ 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: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

◀ 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: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

◀ 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: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

◀ 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: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

◀ 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: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

◀ 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: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

▪ 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: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

▪ 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: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

▪ 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: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

◀ 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: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

▪ 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: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

▪ 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: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

▪ 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: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

◀ 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: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

« 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: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

▲ 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: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

▪ 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: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

◀ 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: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

▪ 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: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

▪ 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: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

▪ 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: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

▪ 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: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

▪ 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: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

▪ 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: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

↳ 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: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

◀ 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: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

◀ 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: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

↳ 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: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

◀ 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: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

▪ 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: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

↳ 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: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

▪ 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: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

◀ 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: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

◀ 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: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

◀ 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: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

▪ Syntax

C# VB F#

[Copy](#)

```
public void Reset()
```

▪ See Also

Reference

[Board Class](#)

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

IO.Devices.ClickBoards.RemoteIO.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: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

▪ 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.RemotelO.Expand2](#)

Assembly: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

↳ 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: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

◀ 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: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

◀ 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: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

▪ 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: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

↳ 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: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

▪ 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.RemotelIO.PWM](#)

Assembly: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

◀ 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: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

↳ 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: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

↳ 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.RemotelIO.PWM](#)

Assembly: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

↳ 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.RemotelIO.PWM](#)

Assembly: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

↳ 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: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

↳ 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: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

▪ 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 |
|---|--------------------------|--|
|  | blinking | 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: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

◀ 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 |
|--|----------------------------|--|
| | blinking | 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: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

↳ 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.RemoteIO.SevenSegment](#)

Assembly: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

↳ Syntax

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

[Copy](#)

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

Property Value

Type: [Double](#)

↳ See Also

Reference

[Board Class](#)

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

Boardleftdp Property

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

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

Assembly: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

↳ 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: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

↳ 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: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

↳ 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.RemoteIO.SevenSegment](#)

Assembly: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

↳ Syntax

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

[Copy](#)

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

Property Value

Type: [Int32](#)

↳ See Also

Reference

[Board Class](#)

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

Assembly: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

◀ Syntax

C# VB F#

[Copy](#)

```
public void Clear()
```

◀ See Also

Reference

[Board Class](#)

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

BoardReset Method

Issue hardware reset to the 74HC595 shift register chain.

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

Assembly: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

▪ 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: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

▪ 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: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

▪ 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.ClickBoards.SimpleIO.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.SimpleIO.ADACBoard](#)

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

Assembly: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

▪ 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.SimpleIO.ADAC Namespace](#)

Board Constructor

Constructor for a single ADAC click.

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

Assembly: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

◀ Syntax

C# VB F#

[Copy](#)

```
public Board(  
    int socknum,  
    int addr = 16  
)
```

Parameters

socknum

Type: [SystemInt32](#)

mikroBUS socket number.

addr (Optional)

Type: [SystemInt32](#)

I²C slave address.

◀ See Also

[Reference](#)

[Board Class](#)

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

Boarddevice Property

Returns the underlying AD5593R device object.

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

Assembly: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

◀ Syntax

C# VB F#

[Copy](#)

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

Property Value

Type: [Device](#)

◀ See Also

Reference

[Board Class](#)

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

BoardADC Method

Factory function for creating ADC inputs.

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

Assembly: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

◀ 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.SimpleIO.ADAC Namespace](#)

BoardDAC Method

Factory function for creating DAC outputs.

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

Assembly: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

↳ 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.SimpleIO.ADAC Namespace

BoardGPIO Method

Factory function for creating GPIO pins.

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

Assembly: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

◀ 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.SimpleIO.ADAC Namespace](#)

BoardReset Method

Issue hardware reset to the AD5593R.

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

Assembly: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

▪ Syntax

C# VB F#

[Copy](#)

```
public void Reset()
```

▪ See Also

Reference

[Board Class](#)

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

BoardDefaultAddress Field

Default I²C slave address.

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

Assembly: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

↳ Syntax

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

[Copy](#)

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

Field Value

Type: [Byte](#)

↳ See Also

Reference

[Board Class](#)

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

IO.Devices.ClickBoards.SimpleIO.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.SimpleIO.ExpandBoard](#)

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

Assembly: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

▪ 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 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.SimpleIO.Expand Namespace](#)

Board Constructor

Constructor for a single Expand 2 click.

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

Assembly: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

◀ Syntax

C# VB F#

[Copy](#)

```
public Board(  
            int socknum  
)
```

Parameters

socknum

Type: [SystemInt32](#)

mikroBUS socket number.

◀ See Also

Reference

[Board Class](#)

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

Boarddevice Property

Returns the underlying MCP23S17 device object.

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

Assembly: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

◀ Syntax

C# VB F#

[Copy](#)

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

Property Value

Type: [Device](#)

◀ See Also

Reference

[Board Class](#)

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

BoardGPIO Method

Factory function for creating GPIO pins.

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

Assembly: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

◀ 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.SimpleIO.Expand Namespace](#)

BoardReset Method

Issue hardware reset to the MCP23S17.

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

Assembly: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

▪ Syntax

C# VB F#

[Copy](#)

```
public void Reset()
```

▪ See Also

Reference

[Board Class](#)

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

IO.Devices.ClickBoards.SimpleIO.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.SimpleIO.Expand2Board](#)

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

Assembly: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

▪ 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.SimpleIO.Expand2 Namespace](#)

Board Constructor

Constructor for a single Expand 2 click.

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

Assembly: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

◀ Syntax

C# VB F#

[Copy](#)

```
public Board(  
    int socknum,  
    int addr = 32  
)
```

Parameters

socknum

Type: [SystemInt32](#)

mikroBUS socket number.

addr (Optional)

Type: [SystemInt32](#)

I²C slave address.

◀ See Also

[Reference](#)

[Board Class](#)

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

Boarddevice Property

Returns the underlying MCP23017 device object.

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

Assembly: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

◀ Syntax

C# VB F#

[Copy](#)

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

Property Value

Type: [Device](#)

◀ See Also

Reference

[Board Class](#)

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

BoardGPIO Method

Factory function for creating GPIO pins.

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

Assembly: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

◀ 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.SimpleIO.Expand2 Namespace](#)

BoardReset Method

Issue hardware reset to the MCP23017.

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

Assembly: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

▪ Syntax

C# VB F#

[Copy](#)

```
public void Reset()
```

▪ See Also

Reference

[Board Class](#)

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

BoardDefaultAddress Field

Default I²C slave address.

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

Assembly: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

↳ Syntax

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

[Copy](#)

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

Field Value

Type: [Int32](#)

↳ See Also

Reference

[Board Class](#)

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

IO.Devices.ClickBoards.SimpleIO.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.SimpleIO.PWMBoard](#)

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

Assembly: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

▪ 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.SimpleIO.PWM Namespace](#)

Board Constructor

Constructor for a single PWM click.

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

Assembly: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

↳ Syntax

C# VB F#

[Copy](#)

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

Parameters

socknum

Type: [SystemInt32](#)

mikroBUS socket number.

freq

Type: [SystemInt32](#)

PWM pulse frequency in Hz.

addr (Optional)

Type: [SystemInt32](#)

I²C slave address.

↳ See Also

[Reference](#)

[Board Class](#)

IO.Devices.ClickBoards.SimpleIO.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.SimpleIO.PWM Namespace](#)

Boarddev Property

Returns the underlying PCA9685 device object.

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

Assembly: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

↳ Syntax

C# VB F#

[Copy](#)

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

Property Value

Type: [Device](#)

↳ See Also

Reference

[Board Class](#)

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

BoardGPIO Method

Factory function for creating GPIO output pins.

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

Assembly: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

↳ 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.SimpleIO.PWM Namespace

BoardPWM Method

Factory function for creating PWM outputs.

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

Assembly: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

↳ 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.SimpleIO.PWM Namespace

BoardServo Method

Factory function for creating servo outputs.

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

Assembly: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

↳ 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.SimpleIO.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.SimpleIO.PWM Namespace](#)

BoardDefaultAddress Field

Default I²C slave address.

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

Assembly: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

↳ Syntax

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

[Copy](#)

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

Field Value

Type: [Byte](#)

↳ See Also

Reference

[Board Class](#)

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

IO.Devices.ClickBoards.SimpleIO.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.SimpleIO.SevenSegmentBoard](#)

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

Assembly: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

▪ 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 |
|---|--------------------------|--|
|  | blinking | 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. |

[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.SimpleIO.SevenSegment Namespace](#)

Board Constructor

Constructor for a single 7seg click.

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

Assembly: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

◀ Syntax

C# VB F#

[Copy](#)

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

Parameters

socket

Type: [SystemInt32](#)

mikroBUS socket number.

radix (Optional)

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

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

[Hexadecimal](#).

blanking (Optional)

Type: [IO.Devices.ClickBoards.SimpleIO.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.

◀ See Also

Reference

[Board Class](#)

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

Boardblanking Property

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

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

Assembly: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

↳ Syntax

C# VB F#

[Copy](#)

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

Property Value

Type: [BoardZeroBlanking](#)

↳ See Also

Reference

[Board Class](#)

[IO.Devices.ClickBoards.SimpleIO.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.SimpleIO.SevenSegment](#)

Assembly: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

↳ Syntax

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

[Copy](#)

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

Property Value

Type: [Double](#)

↳ See Also

Reference

[Board Class](#)

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

Boardleftdp Property

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

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

Assembly: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

↳ Syntax

C# VB F#

[Copy](#)

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

Property Value

Type: [Boolean](#)

↳ See Also

Reference

[Board Class](#)

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

Boardradix Property

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

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

Assembly: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

↳ Syntax

C# VB F#

[Copy](#)

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

Property Value

Type: [BoardBase](#)

↳ See Also

Reference

[Board Class](#)

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

Boardrightdp Property

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

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

Assembly: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

↳ Syntax

C# VB F#

[Copy](#)

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

Property Value

Type: [Boolean](#)

↳ See Also

Reference

[Board Class](#)

[IO.Devices.ClickBoards.SimpleIO.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.SimpleIO.SevenSegment](#)

Assembly: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

↳ Syntax

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

[Copy](#)

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

Property Value

Type: [Int32](#)

↳ See Also

Reference

[Board Class](#)

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

Board Methods

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

↳ Methods

| | Name | Description |
|---|-----------------------|--------------------|
|  | Clear | Clear the display. |

[Top](#)

↳ See Also

Reference

[Board Class](#)

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

BoardClear Method

Clear the display.

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

Assembly: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

◀ Syntax

C# VB F#

[Copy](#)

```
public void Clear()
```

◀ See Also

Reference

[Board Class](#)

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

BoardBase Enumeration

Numerical systems.

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

Assembly: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

▪ 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.SimpleIO.SevenSegment Namespace](#)

BoardZeroBlanking Enumeration

Zero blanking modes.

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

Assembly: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

▪ 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.SimpleIO.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: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

▪ 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: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

▪ 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: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

« 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: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

▪ 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: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

◀ 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: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

▪ 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: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

▪ 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: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

▪ 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: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

▪ 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: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

◀ 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: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

▪ 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: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

▪ 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: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

▪ 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: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

◀ 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: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

▪ 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: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

▪ 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: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

▪ 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: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

◀ 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: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

◀ 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: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

« 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: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

◀ 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: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

◀ 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: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

◀ 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: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

◀ 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: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

◀ 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: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

◀ 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: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

◀ 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: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

« 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: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

▪ 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: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

▪ 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: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

▪ 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: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

▪ 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: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

▪ 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: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

◀ 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: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

◀ 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: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

◀ 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: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

◀ 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: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

« 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: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

« 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: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

↳ 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: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

◀ 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: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

◀ 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: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

◀ 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: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

◀ 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: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

◀ 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: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

▪ 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: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

◀ 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: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

↳ 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: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

▪ 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: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

▀ 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: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

▪ 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: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

▪ 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: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

▪ 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: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

◀ 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: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

◀ 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: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

◀ 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: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

◀ 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: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

« 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: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

« 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: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

↳ 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: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

◀ 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: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

◀ 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: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

◀ 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: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

◀ 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: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

◀ 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: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

▪ 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: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

◀ 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: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

◀ 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: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

▪ 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: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

◀ 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: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

↳ 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: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

▪ 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: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

◀ 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: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

◀ 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: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

▪ 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: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

▀ 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: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

◀ 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: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

▪ 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: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

◀ 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: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

▪ 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: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

▪ 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: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

◀ 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: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

◀ 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: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

◀ 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: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

▪ 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: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

◀ 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: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

◀ 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: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

◀ 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: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

◀ 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: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

◀ 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: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

◀ 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: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

« 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: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

◀ 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: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

◀ 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: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

◀ 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: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

◀ 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: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

◀ 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: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

▪ 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: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

◀ 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: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

↳ 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: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

▪ 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: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

◀ 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: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

◀ 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: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

◀ 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: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

◀ 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: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

↳ 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: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

◀ 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: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

« 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: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

◀ 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: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

◀ 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: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

▪ 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: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

◀ 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: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

▪ 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: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

▪ 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: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

◀ 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: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

↳ 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: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

▪ 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: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

◀ 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: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

↳ 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: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

▪ 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: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

◀ 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: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

▪ 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: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

↳ 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: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

« 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: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

↳ 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: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

« 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: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

◀ 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: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

« 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: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

◀ 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: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

« 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: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

▪ 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: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

◀ 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: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

▪ 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: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

▪ 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: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

▀ 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: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

▪ 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: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

◀ 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: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

▪ 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: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

↳ 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: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

▪ 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.ThermistorNTC_B](#)

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

Assembly: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

▪ 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: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

◀ 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: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

◀ 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: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

▪ 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: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

« 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: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

↳ 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: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

▪ 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: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

« 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: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

◀ 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.ADCInput](#)

Namespace: [IO.Interfaces.ADC](#)

Assembly: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

▪ 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: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

↳ 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: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

▪ 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: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

▪ 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: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

◀ 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: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

◀ 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: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

↳ 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: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

◀ 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: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

▪ 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: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

↳ 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: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

↳ 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: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

▲ 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: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

◀ 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: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

◀ 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: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

▪ 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: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

◀ 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: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

▪ 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: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

▪ 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: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

↳ 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: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

▪ 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: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

◀ 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: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

▪ 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: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

« 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: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

↳ 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: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

« 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: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

▪ 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: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

▪ 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: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

▪ 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: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

↳ 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: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

▪ 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: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

▪ 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: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

◀ 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: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

◀ 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: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

↳ 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.Log Namespace

Abstract Interface for Error Logging Services

▪ Interfaces

| Interface | Description |
|-----------|-------------|
|-----------|-------------|



| | |
|------------------------|---|
| Logger | Abstract interface for error logging systems. |
|------------------------|---|

Logger Interface

Abstract interface for error logging systems.

Namespace: [IO.Interfaces.Log](#)

Assembly: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

▪ Syntax

C# VB F#

[Copy](#)

```
public interface Logger
```

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

▪ Methods

| | Name | Description |
|---|--------------------------------------|---|
| ≡ | Error(String) | Log an error message. |
| ≡ | Error(String, Int32) | Log an error message, including an <code>errno</code> error string. |
| ≡ | Note | Log a notification message. |
| ≡ | Warning | Log a warning message. |

[Top](#)

▪ See Also

[Reference](#)

IO.Interfaces.Log Namespace

Logger Methods

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

▪ Methods

| | Name | Description |
|---|--------------------------------------|---|
| ≡ | Error(String) | Log an error message. |
| ≡ | Error(String, Int32) | Log an error message, including an <code>errno</code> error string. |
| ≡ | Note | Log a notification message. |
| ≡ | Warning | Log a warning message. |

[Top](#)

▪ See Also

Reference

[Logger Interface](#)

[IO.Interfaces.Log Namespace](#)

LoggerError Method

↳ Overload List

| | Name | Description |
|---|--------------------------------------|---|
| ≡ | Error(String) | Log an error message. |
| ≡ | Error(String, Int32) | Log an error message, including an <code>errno</code> error string. |

[Top](#)

↳ See Also

Reference

[Logger Interface](#)

[IO.Interfaces.Log Namespace](#)

LoggerError Method (String)

Log an error message.

Namespace: [IO.Interfaces.Log](#)

Assembly: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

◀ Syntax

C# VB F#

[Copy](#)

```
void Error(  
    string message  
)
```

Parameters

message

Type: [System.String](#)

Error message.

◀ See Also

Reference

[Logger Interface](#)

[Error Overload](#)

[IO.Interfaces.Log Namespace](#)

LoggerError Method (String, Int32)

Log an error message, including an `errno` error string.

Namespace: [IO.Interfaces.Log](#)

Assembly: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

◀ Syntax

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

[Copy](#)

```
void Error(  
    string message,  
    int errnum  
)
```

Parameters

message

Type: [SystemString](#)

Error Message.

errnum

Type: [SystemInt32](#)

`errno` error number.

◀ See Also

Reference

[Logger Interface](#)

[Error Overload](#)

[IO.Interfaces.Log Namespace](#)

LoggerNote Method

Log a notification message.

Namespace: [IO.Interfaces.Log](#)

Assembly: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

◀ Syntax

C# VB F#

[Copy](#)

```
void Note(  
          string message  
)
```

Parameters

message

Type: [System.String](#)

Notification message.

◀ See Also

Reference

[Logger Interface](#)

[IO.Interfaces.Log Namespace](#)

LoggerWarning Method

Log a warning message.

Namespace: [IO.Interfaces.Log](#)

Assembly: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

◀ Syntax

C# VB F#

[Copy](#)

```
void Warning(  
            string message  
)
```

Parameters

message

Type: [System.String](#)

Warning message.

◀ See Also

Reference

[Logger Interface](#)

[IO.Interfaces.Log Namespace](#)

IO.Interfaces.Message.Text Namespace

Abstract Interface for Text Message Relays

▪ Interfaces

| Interface | Description |
|---|--|
|  Relay | Abstract interface for a text message relay. |

Relay Interface

Abstract interface for a text message relay.

Namespace: [IO.Interfaces.Message.Text](#)

Assembly: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

▪ Syntax

C# VB F#

[Copy](#)

```
public interface Relay
```

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

▪ Methods

| | Name | Description |
|---|----------------------|------------------------------------|
|  | Send | Method for sending a text message. |

[Top](#)

▪ See Also

Reference

[IO.Interfaces.Message.Text Namespace](#)

Relay Methods

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

▪ Methods

| | Name | Description |
|---|----------------------|------------------------------------|
|  | Send | Method for sending a text message. |

[Top](#)

▪ See Also

Reference

[Relay Interface](#)

[IO.Interfaces.Message.Text Namespace](#)

RelaySend Method

Method for sending a text message.

Namespace: [IO.Interfaces.Message.Text](#)

Assembly: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

◀ Syntax

C# VB F#

[Copy](#)

```
void Send(  
    string sender,  
    string recipient,  
    string subject,  
    string message  
)
```

Parameters

sender

Type: [SystemString](#)

Message originator.

recipient

Type: [SystemString](#)

Message recipient.

subject

Type: [SystemString](#)

Message subject.

message

Type: [SystemString](#)

Message text.

↳ See Also

Reference

[Relay Interface](#)

[IO.Interfaces.Message.Text 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: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

▪ 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: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

▪ 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: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

▪ 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: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

« 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: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

◀ 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: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

▪ 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: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

◀ 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: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

◀ 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: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

◀ 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: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

▪ 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: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

« 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: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

▪ 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: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

« 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: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

« 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: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

◀ 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: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

▪ 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: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

◀ 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: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

« 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: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

▪ 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: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

« 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: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

▪ 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: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

« 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: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

▪ 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: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

« 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: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

◀ 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: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

« 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: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

▪ 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: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

◀ 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: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

◀ 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: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

« 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.Stepper Namespace

Abstract Interface for Stepper Motor Outputs

↳ Interfaces

| Interface | Description |
|-----------|-------------|
|-----------|-------------|



| | |
|------------------------|---|
| Output | Abstract interface for stepper motor outputs. |
|------------------------|---|

Output Interface

Abstract interface for stepper motor outputs.

Namespace: [IO.Interfaces.Stepper](#)

Assembly: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

▪ Syntax

C# VB F#

[Copy](#)

```
public interface Output
```

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

▪ Properties

| | Name | Description |
|---|----------------------------------|---|
|  | StepsPerRotation | Read-only property returning the number of steps a stepper motor has. |

[Top](#)

▪ Methods

| | Name | Description |
|---|----------------------|---|
|  | Move | Move the stepper motor a specified number of steps at a specified rate. |
|  | Spin | Spin (i.e. continuous rotation) the stepper |

moter at a specified rate.

[Top](#)

◀ See Also

Reference

[IO.Interfaces.Stepper Namespace](#)

Output Properties

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

▪ Properties

| | Name | Description |
|---|----------------------------------|---|
|  | StepsPerRotation | Read-only property returning the number of steps a stepper motor has. |

[Top](#)

▪ See Also

Reference

[Output Interface](#)

[IO.Interfaces.Stepper Namespace](#)

OutputStepsPerRotation Property

Read-only property returning the number of steps a stepper motor has.

Namespace: [IO.Interfaces.Stepper](#)

Assembly: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

◀ Syntax

C# VB F#

[Copy](#)

```
int StepsPerRotation { get; }
```

Property Value

Type: [Int32](#)

◀ See Also

Reference

[Output Interface](#)

[IO.Interfaces.Stepper Namespace](#)

Output Methods

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

▪ Methods

| | Name | Description |
|---|----------------------|---|
|  | Move | Move the stepper motor a specified number of steps at a specified rate. |
|  | Spin | Spin (i.e. continuous rotation) the stepper motor at a specified rate. |

[Top](#)

▪ See Also

Reference

[Output Interface](#)

[IO.Interfaces.Stepper Namespace](#)

OutputMove Method

Move the stepper motor a specified number of steps at a specified rate.

Namespace: [IO.Interfaces.Stepper](#)

Assembly: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

◀ Syntax

C# VB F#

[Copy](#)

```
void Move(  
    int steps,  
    float rate  
)
```

Parameters

steps

Type: [SystemInt32](#)

Number of steps to move. Negative values indicate reverse motion and positive values indicate forward motion. (The directions are nominal and depend on how the stepper motor coils are wired). Zero indicates the stepper motor should be stopped.

rate

Type: [SystemSingle](#)

The rate of motion, in steps per second. Negative values indicate reverse motion and positive values indicate forward motion. (The directions are nominal and depend on how the stepper motor coils are wired). Zero indicates the stepper motor should be stopped.

↳ See Also

Reference

[Output Interface](#)

[IO.Interfaces.S stepper Namespace](#)

OutputSpin Method

Spin (i.e. continuous rotation) the stepper motor at a specified rate.

Namespace: [IO.Interfaces.Stepper](#)

Assembly: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

◀ Syntax

C# VB F#

[Copy](#)

```
void Spin(  
    float rate  
)
```

Parameters

rate

Type: [SystemSingle](#)

The rate of motion, in steps per second. Negative values indicate reverse motion and positive values indicate forward motion. (The directions are nominal and depend on how the stepper motor coils are wired). Zero indicates the stepper motor should be stopped.

◀ Remarks

Not all stepper motor drivers support continuous rotation.

◀ See Also

Reference

[Output Interface](#)

IO.Interfaces.Stepper 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: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

▪ 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.

| | | |
|---|-------------------------------------|--|
|  | 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

[IO.Interfaces.Temperature Namespace](#)

Conversions Constructor

Initializes a new instance of the [Conversions](#) class

Namespace: [IO.Interfaces.Temperature](#)

Assembly: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

▪ 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: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

◀ 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: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

◀ 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: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

◀ 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: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

◀ 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: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

◀ 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: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

◀ Syntax

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

[Copy](#)

```
public static double KelvinsToFahrenheit(  
    double kelvins  
)
```

Parameters

kelvins

Type: [SystemDouble](#)

Temperature in Kelivns.

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: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

▪ 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: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

◀ 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: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

◀ 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: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

◀ 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: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

▀ 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: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

◀ 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: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

▪ Syntax

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

[Copy](#)

```
void Kick()
```

▪ See Also

Reference

[Timer Interface](#)

[IO.Interfaces.Watchdog Namespace](#)

IO.Objects.Email.Mail Namespace

Email Sending Services using

↳ Classes

| | Class | Description |
|---|-----------------------|---|
|  | Relay | This class implements the IO.Interfaces.Messaging.Text.Relay interface, for sending an email message via the Unix program /usr/bin/mail . |

Relay Class

This class implements the [IO.Interfaces.Messaging.Text.Relay](#) interface, for sending an email message via the Unix program [/usr/bin/mail](#).

► Inheritance Hierarchy

[SystemObject](#) [IO.Objects.Email.MailRelay](#)

Namespace: [IO.Objects.Email](#)

Assembly: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

► Syntax

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

[Copy](#)

```
public class Relay : Relay
```

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

► Constructors

| | Name | Description |
|---|-----------------------|--------------------------------------|
| ≡ | Relay | Constructor for a single mail relay. |

[Top](#)

► Methods

| | Name | Description |
|---|------|-------------|
| ≡ | | |

| | | |
|---|--|---|
| | Send(String, String, String) | Method for sending an email message. |
| » | Send(String, String, String, String) | Method for sending an email message. |
| » | Send(String, String, String, String, String) | Method for sending an email message with an attachment. |

[Top](#)

◀ See Also

Reference

[IO.Objects.Email.Mail Namespace](#)

Relay Constructor

Constructor for a single mail relay.

Namespace: [IO.Objects.Email.Mail](#)

Assembly: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

▀ Syntax

C# VB F#

[Copy](#)

```
public Relay()
```

▀ See Also

Reference

[Relay Class](#)

[IO.Objects.Email.Mail Namespace](#)

Relay Methods

◀ Methods

| | Name | Description |
|---|--|---|
| ≡ | Send(String, String, String) | Method for sending an email message. |
| ≡ | Send(String, String, String, String) | Method for sending an email message. |
| ≡ | Send(String, String, String, String, String) | Method for sending an email message with an attachment. |

[Top](#)

◀ See Also

Reference

[Relay Class](#)

[IO.Objects.Email.Mail Namespace](#)

RelaySend Method

↳ Overload List

| | Name | Description |
|---|--|---|
| ≡ | Send(String, String, String) | Method for sending an email message. |
| ≡ | Send(String, String, String, String) | Method for sending an email message. |
| ≡ | Send(String, String, String, String, String) | Method for sending an email message with an attachment. |

[Top](#)

↳ See Also

Reference

[Relay Class](#)

[IO.Objects.Email.Mail Namespace](#)

RelaySend Method (String, String, String)

Method for sending an email message.

Namespace: [IO.Objects.Email.Mail](#)

Assembly: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

◀ Syntax

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

[Copy](#)

```
public void Send(  
    string recipient,  
    string subject,  
    string message  
)
```

Parameters

recipient

Type: [SystemString](#)

Originator email address.

subject

Type: [SystemString](#)

Recipient email address.

message

Type: [SystemString](#)

Email message body.

◀ See Also

Reference

[Relay Class](#)

[Send Overload](#)

[IO.Objects.Email.Mail Namespace](#)

RelaySend Method (String, String, String, String)

Method for sending an email message.

Namespace: [IO.Objects.Email.Mail](#)

Assembly: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

◀ Syntax

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

[Copy](#)

```
public void Send(  
    string sender,  
    string recipient,  
    string subject,  
    string message  
)
```

Parameters

sender

Type: [SystemString](#)

Originator email address.

recipient

Type: [SystemString](#)

Recipient email address.

subject

Type: [SystemString](#)

Subject of the email message.

message

Type: [SystemString](#)
Email message body.

Implements

[RelaySend\(String, String, String, String\)](#)

See Also

Reference

[Relay Class](#)

[Send Overload](#)

[IO.Objects.Email.Mail Namespace](#)

RelaySend Method (String, String, String, String, String)

Method for sending an email message with an attachment.

Namespace: [IO.Objects.Email.Mail](#)

Assembly: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

◀ Syntax

C# VB F#

Copy

```
public void Send(  
    string sender,  
    string recipient,  
    string subject,  
    string message,  
    string attachment  
)
```

Parameters

sender

Type: [SystemString](#)

Originator email address.

recipient

Type: [SystemString](#)

Recipient email address.

subject

Type: [SystemString](#)

Subject of the email message.

message

Type: [SystemString](#)

Email message body.

attachment

Type: [SystemString](#)

Attachment file name.

↳ See Also

Reference

[Relay Class](#)

[Send Overload](#)

[IO.Objects.Email.Mail Namespace](#)

IO.Objects.Email.SMTP Namespace

Email Sending Services using SMTP to

↳ Classes

| | Class | Description |
|---|-----------------------|--|
|  | Relay | This class implements the IO.Interfaces.Messaging.Text.Relay interface, for sending an email message via SMTP (Simple Mail Transfer Protocol). |

Relay Class

This class implements the [IO.Interfaces.Messaging.Text.Relay](#) interface, for sending an email message via SMTP (Simple Mail Transfer Protocol).

► Inheritance Hierarchy

[SystemObject](#) [IO.Objects.Email.SMTPRelay](#)

Namespace: [IO.Objects.Email.SMTP](#)

Assembly: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

► Syntax

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

[Copy](#)

```
public class Relay : Relay
```

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

► Constructors

| | Name | Description |
|---|-----------------------|---|
| ≡ | Relay | Constructor for a single SMTP mail relay. |

[Top](#)

► Methods

| | Name | Description |
|---|------|-------------|
| ≡ | | |

| | | |
|---|--|---|
| | Send(String, String, String) | Method for sending an email message. |
| » | Send(String, String, String, String) | Method for sending an email message. |
| » | Send(String, String, String, String, String) | Method for sending an email message with an attachment. |

[Top](#)

◀ See Also

Reference

[IO.Objects.Email.SMTP Namespace](#)

Relay Constructor

Constructor for a single SMTP mail relay.

Namespace: [IO.Objects.Email.SMTP](#)

Assembly: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

◀ Syntax

C# VB F#

[Copy](#)

```
public Relay(  
    string server = "localhost",  
    int port = 25  
)
```

Parameters

server (Optional)

Type: [SystemString](#)

SMTP server domain name or IP addaress.

port (Optional)

Type: [SystemInt32](#)

SMTP server TCP port number.

◀ See Also

[Reference](#)

[Relay Class](#)

[IO.Objects.Email.SMTP Namespace](#)

Relay Methods

◀ Methods

| | Name | Description |
|---|--|---|
| ≡ | Send(String, String, String) | Method for sending an email message. |
| ≡ | Send(String, String, String, String) | Method for sending an email message. |
| ≡ | Send(String, String, String, String, String) | Method for sending an email message with an attachment. |

[Top](#)

◀ See Also

Reference

[Relay Class](#)

[IO.Objects.Email.SMTP Namespace](#)

RelaySend Method

↳ Overload List

| | Name | Description |
|---|--|---|
| ≡ | Send(String, String, String) | Method for sending an email message. |
| ≡ | Send(String, String, String, String) | Method for sending an email message. |
| ≡ | Send(String, String, String, String, String) | Method for sending an email message with an attachment. |

[Top](#)

↳ See Also

Reference

[Relay Class](#)

[IO.Objects.Email.SMTP Namespace](#)

RelaySend Method (String, String, String)

Method for sending an email message.

Namespace: [IO.Objects.Email.SMTP](#)

Assembly: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

◀ Syntax

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

[Copy](#)

```
public void Send(  
    string recipient,  
    string subject,  
    string message  
)
```

Parameters

recipient

Type: [SystemString](#)

Originator email address.

subject

Type: [SystemString](#)

Recipient email address.

message

Type: [SystemString](#)

Email message body.

◀ See Also

Reference

[Relay Class](#)

[Send Overload](#)

[IO.Objects.Email.SMTP Namespace](#)

RelaySend Method (String, String, String, String)

Method for sending an email message.

Namespace: [IO.Objects.Email.SMTP](#)

Assembly: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

◀ Syntax

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

[Copy](#)

```
public void Send(  
    string sender,  
    string recipient,  
    string subject,  
    string message  
)
```

Parameters

sender

Type: [SystemString](#)

Originator email address.

recipient

Type: [SystemString](#)

Recipient email address.

subject

Type: [SystemString](#)

Subject of the email message.

message

Type: [SystemString](#)
Email message body.

Implements

[RelaySend\(String, String, String, String\)](#)

See Also

Reference

[Relay Class](#)

[Send Overload](#)

[IO.Objects.Email.SMTP Namespace](#)

RelaySend Method (String, String, String, String, String)

Method for sending an email message with an attachment.

Namespace: [IO.Objects.Email.SMTP](#)

Assembly: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

◀ Syntax

C# VB F#

Copy

```
public void Send(  
    string sender,  
    string recipient,  
    string subject,  
    string message,  
    string attachment  
)
```

Parameters

sender

Type: [SystemString](#)

Originator email address.

recipient

Type: [SystemString](#)

Recipient email address.

subject

Type: [SystemString](#)

Subject of the email message.

message

Type: [SystemString](#)

Email message body.

attachment

Type: [SystemString](#)

Attachment file name.

↳ See Also

Reference

[Relay Class](#)

[Send Overload](#)

[IO.Objects.Email.SMTP Namespace](#)

IO.Objects.GPIO.PWM Namespace

PWM Controlled GPIO Output Services

↳ Classes

| Class | Description |
|---|---|
|  Pin | Encapsulates GPIO output pins implemented with PWM outputs. |

Pin Class

Encapsulates GPIO output pins implemented with PWM outputs.

▪ Inheritance Hierarchy

[SystemObject](#) [IO.Objects.GPIO.PWMPin](#)

Namespace: [IO.Objects.GPIO](#)

Assembly: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

▪ 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](#)

▪ Remarks

Use cases for this class include ON/OFF things like an LED or a solenoid valve driven from a PWM output.

▪ Remarks

Depending on the PWM hardware implementation, the OFF duty cycle may be slightly greater than 0 % and/or the ON duty cycle may be slightly less than 100 %.

▪ See Also

Reference

[IO.Objects.GPIO.PWM Namespace](#)

Pin Constructor

Constructor for a single GPIO pin.

Namespace: [IO.Objects.GPIO.PWM](#)

Assembly: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

◀ Syntax

C# VB F#

[Copy](#)

```
public Pin(  
    Output outp,  
    bool state = false,  
    double dutycycle = 100  
)
```

Parameters

outp

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

PWM output instance.

state (Optional)

Type: [SystemBoolean](#)

Initial GPIO output state.

dutycycle (Optional)

Type: [SystemDouble](#)

Initial PWM output duty cycle. Allowed values are 0.0 to 100.0 percent.

◀ See Also

[Reference](#)

Pin Class
IO.Objects.GPIO.PWM 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.Objects.GPIO.PWM Namespace](#)

Pinstate Property

Read/Write GPIO state property.

Namespace: [IO.Objects.GPIO.PWM](#)

Assembly: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

↳ Syntax

C# VB F#

[Copy](#)

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

Property Value

Type: [Boolean](#)

Implements

[Pinstate](#)

↳ See Also

[Reference](#)

[Pin Class](#)

[IO.Objects.GPIO.PWM Namespace](#)

IO.Objects.libsimpleio.ADC Namespace

ADC (Analog to Digital Converter) Input Services

◀ Classes

| Class | Description |
|--|--|
|  Sample | Encapsulates Linux Industrial I/O Subsystem ADC inputs using libsimpleio . |

Sample Class

Encapsulates Linux Industrial I/O Subsystem ADC inputs using [libsimpleio](#).

▪ Inheritance Hierarchy

[SystemObject](#) [IO.Objects.libsimpleio.ADCSample](#)

Namespace: [IO.Objects.libsimpleio.ADC](#)

Assembly: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

▪ Syntax

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

[Copy](#)

```
public class Sample : Sample
```

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

▪ Constructors

| | Name | Description |
|---|------------------------|-------------------------------------|
| ≡ | Sample | Constructor for a single ADC input. |

[Top](#)

▪ Properties

| | Name | Description |
|---|--------------------|--|
| ! | fd | Read-only property returning the Linux |

file descriptor for the ADC input.



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



sample Read-only property returning an integer analog sample from an ADC input.

[Top](#)

◀ Methods

| | Name | Description |
|-------|-------------|---|
| S | name | Retrieve the subsystem name string for a Linux Industrial I/O Subsystem ADC device. |

[Top](#)

◀ See Also

Reference

[IO.Objects.libsimpleio.ADC Namespace](#)

Sample Constructor

Constructor for a single ADC input.

Namespace: [IO.Objects.libsimpleio.ADC](#)

Assembly: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

◀ Syntax

C# VB F#

[Copy](#)

```
public Sample(  
    Designator desg,  
    int resolution  
)
```

Parameters

desg

Type: [IO.Objects.libsimpleio.DeviceDesignator](#)

ADC input designator.

resolution

Type: [SystemInt32](#)

Bits of resolution.

◀ See Also

[Reference](#)

[Sample Class](#)

[IO.Objects.libsimpleio.ADC Namespace](#)

Sample Properties

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

Properties

| | Name | Description |
|--|----------------------------|---|
| | fd | Read-only property returning the Linux file descriptor for the ADC input. |
| | resolution | Read-only property returning the number of bits of resolution. |
| | sample | Read-only property returning an integer analog sample from an ADC input. |

[Top](#)

See Also

Reference

[Sample Class](#)

[IO.Objects.libsimpleio.ADC Namespace](#)

Samplefd Property

Read-only property returning the Linux file descriptor for the ADC input.

Namespace: [IO.Objects.libsimpleio.ADC](#)

Assembly: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

« Syntax

C# VB F#

[Copy](#)

```
public int fd { get; }
```

Property Value

Type: [Int32](#)

« See Also

Reference

[Sample Class](#)

[IO.Objects.libsimpleio.ADC Namespace](#)

Sampleresolution Property

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

Namespace: [IO.Objects.libsimpleio.ADC](#)

Assembly: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

▪ Syntax

C# VB F#

[Copy](#)

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

Property Value

Type: [Int32](#)

Implements

[Sampleresolution](#)

▪ See Also

[Reference](#)

[Sample Class](#)

[IO.Objects.libsimpleio.ADC Namespace](#)

Samplesample Property

Read-only property returning an integer analog sample from an ADC input.

Namespace: [IO.Objects.libsimpleio.ADC](#)

Assembly: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

▲ Syntax

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

[Copy](#)

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

Property Value

Type: [Int32](#)

Implements

[Samplesample](#)

▲ See Also

Reference

[Sample Class](#)

[IO.Objects.libsimpleio.ADC Namespace](#)

Sample Methods

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

▪ Methods

| | Name | Description |
|---|----------------------|---|
|   | name | Retrieve the subsystem name string for a Linux Industrial I/O Subsystem ADC device. |

[Top](#)

▪ See Also

[Reference](#)

[Sample Class](#)

[IO.Objects.libsimpleio.ADC Namespace](#)

Samplename Method

Retrieve the subsystem name string for a Linux Industrial I/O Subsystem ADC device.

Namespace: [IO.Objects.libsimpleio.ADC](#)

Assembly: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

◀ Syntax

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

[Copy](#)

```
public static string name(  
    int chip  
)
```

Parameters

chip

Type: [SystemInt32](#)

ADC chip number.

Return Value

Type: [String](#)

Subsystem name.

◀ See Also

[Reference](#)

[Sample Class](#)

[IO.Objects.libsimpleio.ADC Namespace](#)

IO.Objects.libsimpleio.DAC Namespace

DAC (Digital to Analog Converter) Output Services

↳ Classes

| Class | Description |
|--|---|
|  Sample | Encapsulates Linux Industrial I/O Subsystem DAC outputs using libsimpleio . |

Sample Class

Encapsulates Linux Industrial I/O Subsystem DAC outputs using [libsimpleio](#).

▪ Inheritance Hierarchy

[SystemObject](#) [IO.Objects.libsimpleio.DACSample](#)

Namespace: [IO.Objects.libsimpleio.DAC](#)

Assembly: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

▪ Syntax

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

[Copy](#)

```
public class Sample : Sample
```

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

▪ Constructors

| | Name | Description |
|---|------------------------|--------------------------------------|
| ≡ | Sample | Constructor for a single DAC output. |

[Top](#)

▪ Properties

| | Name | Description |
|---|--------------------|--|
| ! | fd | Read-only property returning the Linux |

file descriptor for the DAC output.



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](#)

◀ Methods

| | Name | Description |
|--|-------------|---|
| | name | Retrieve the subsystem name string for a Linux Industrial I/O Subsystem DAC device. |

[Top](#)

◀ See Also

Reference

[IO.Objects.libsimpleio.DAC Namespace](#)

Sample Constructor

Constructor for a single DAC output.

Namespace: [IO.Objects.libsimpleio.DAC](#)

Assembly: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

↳ Syntax

C# VB F#

[Copy](#)

```
public Sample(  
    Designator desg,  
    int resolution,  
    int sample = 0  
)
```

Parameters

desg

Type: [IO.Objects.libsimpleio.DeviceDesignator](#)
DAC output designator.

resolution

Type: [SystemInt32](#)
Bits of resolution.

sample (Optional)

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

↳ See Also

[Reference](#)

[Sample Class](#)

IO.Objects.libsimpleio.DAC Namespace

Sample Properties

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

Properties

| | Name | Description |
|--|----------------------------|--|
| | fd | Read-only property returning the Linux file descriptor for the DAC output. |
| | 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.Objects.libsimpleio.DAC Namespace](#)

Samplefd Property

Read-only property returning the Linux file descriptor for the DAC output.

Namespace: [IO.Objects.libsimpleio.DAC](#)

Assembly: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

▲ Syntax

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

[Copy](#)

```
public int fd { get; }
```

Property Value

Type: [Int32](#)

▲ See Also

[Reference](#)

[Sample Class](#)

[IO.Objects.libsimpleio.DAC Namespace](#)

Sampleresolution Property

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

Namespace: [IO.Objects.libsimpleio.DAC](#)

Assembly: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

▪ Syntax

C# VB F#

[Copy](#)

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

Property Value

Type: [Int32](#)

Implements

[Sampleresolution](#)

▪ See Also

[Reference](#)

[Sample Class](#)

[IO.Objects.libsimpleio.DAC Namespace](#)

Samplesample Property

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

Namespace: [IO.Objects.libsimpleio.DAC](#)

Assembly: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

▲ Syntax

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

[Copy](#)

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

Property Value

Type: [Int32](#)

Implements

[Samplesample](#)

▲ See Also

Reference

[Sample Class](#)

[IO.Objects.libsimpleio.DAC Namespace](#)

Sample Methods

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

▪ Methods

| | Name | Description |
|---|----------------------|---|
|   | name | Retrieve the subsystem name string for a Linux Industrial I/O Subsystem DAC device. |

[Top](#)

▪ See Also

[Reference](#)

[Sample Class](#)

[IO.Objects.libsimpleio.DAC Namespace](#)

Samplename Method

Retrieve the subsystem name string for a Linux Industrial I/O Subsystem DAC device.

Namespace: [IO.Objects.libsimpleio.DAC](#)

Assembly: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

◀ Syntax

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

[Copy](#)

```
public static string name(  
    int chip  
)
```

Parameters

chip

Type: [SystemInt32](#)

DAC chip number.

Return Value

Type: [String](#)

Subsystem name.

◀ See Also

[Reference](#)

[Sample Class](#)

[IO.Objects.libsimpleio.DAC Namespace](#)

IO.Objects.libsimpleio.Device Namespace

Common device declarations

▪ Structures

| | Structure | Description |
|---|----------------------------|--------------------------------------|
|  | Designator | Linux kernel I/O device designator.. |

Designator Structure

Linux kernel I/O device designator..

Namespace: [IO.Objects.libsimpleio.Device](#)

Assembly: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

▪ Syntax

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

[Copy](#)

```
public struct Designator
```

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

▪ Constructors

| | Name | Description |
|---|----------------------------|------------------------------------|
|  | Designator | device pin designator constructor. |

[Top](#)

▪ Properties

| | Name | Description |
|---|---------------------------|---|
|  | available | Returns <code>true</code> if this device designator is not equal to Unavailable . |

[Top](#)

▪ Fields

| | Name | Description |
|------------|-----------------------------|---|
| ◆ | chan | Linux kernel I/O device channel number. |
| ◆ | chip | Linux kernel I/O device chip number. |
| ◆ S | Unavailable | Linux kernel I/O device designator for an unavailable device. |

[Top](#)

◀ Remarks

Many Linux kernel I/O devices, including ADC inputs, DAC outputs, GPIO pins, I2C buses, PWM outputs, and SPI devices, are selected by a tuple of integers: chip and channel.

◀ See Also

Reference

[IO.Objects.libsimpleio.Device Namespace](#)

Designator Constructor

device pin designator constructor.

Namespace: [IO.Objects.libsimpleio.Device](#)

Assembly: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

◀ Syntax

C# VB F#

[Copy](#)

```
public Designator(  
    uint chip,  
    uint chan  
)
```

Parameters

chip

Type: [SystemUInt32](#)

Linux kernel I/O device chip number.

chan

Type: [SystemUInt32](#)

Linux kernel I/O device channel number.

◀ See Also

Reference

[Designator Structure](#)

[IO.Objects.libsimpleio.Device Namespace](#)

Designator Properties

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

Properties

| | Name | Description |
|---|---------------------------|---|
|  | available | Returns <code>true</code> if this device designator is not equal to Unavailable . |

[Top](#)

See Also

Reference

[Designator Structure](#)

[IO.Objects.libsimpleio.Device Namespace](#)

Designatoravailable Property

Returns `true` if this device designator is not equal to `Unavailable`.

Namespace: [IO.Objects.libsimpleio.Device](#)

Assembly: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

« Syntax

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

[Copy](#)

```
public bool available { get; }
```

Property Value

Type: [Boolean](#)

« See Also

Reference

[Designator Structure](#)

[IO.Objects.libsimpleio.Device Namespace](#)

Designator Fields

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

Fields

| | Name | Description |
|------------|-----------------------------|---|
| ◆ | chan | Linux kernel I/O device channel number. |
| ◆ | chip | Linux kernel I/O device chip number. |
| ◆ S | Unavailable | Linux kernel I/O device designator for an unavailable device. |

[Top](#)

See Also

Reference

[Designator Structure](#)

[IO.Objects.libsimpleio.Device Namespace](#)

Designatorchan Field

Linux kernel I/O device channel number.

Namespace: [IO.Objects.libsimpleio.Device](#)

Assembly: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

◀ Syntax

C# VB F#

[Copy](#)

```
public uint chan
```

Field Value

Type: [UInt32](#)

◀ See Also

Reference

[Designator Structure](#)

[IO.Objects.libsimpleio.Device Namespace](#)

Designatorchip Field

Linux kernel I/O device chip number.

Namespace: [IO.Objects.libsimpleio.Device](#)

Assembly: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

◀ Syntax

C# VB F#

[Copy](#)

```
public uint chip
```

Field Value

Type: [UInt32](#)

◀ See Also

Reference

[Designator Structure](#)

[IO.Objects.libsimpleio.Device Namespace](#)

DesignatorUnavailable Field

Linux kernel I/O device designator for an unavailable device.

Namespace: [IO.Objects.libsimpleio.Device](#)

Assembly: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

◀ Syntax

C# VB F#

[Copy](#)

```
public static readonly Designator Unavailable
```

Field Value

Type: [Designator](#)

◀ See Also

Reference

[Designator Structure](#)

[IO.Objects.libsimpleio.Device Namespace](#)

IO.Objects.libsimpleio.GPIO Namespace

GPIO (General Purpose Input/Output) Pin Services

▪ Classes

| | Class | Description |
|---|---------------------|--|
|  | Pin | Encapsulates Linux GPIO pins using libsimpleio . |

▪ Enumerations

| | Enumeration | Description |
|---|-----------------------------|-------------------------------------|
|  | PinDriver | GPIO output driver settings. |
|  | PinEdge | GPIO input interrupt edge settings. |
|  | PinPolarity | GPIO polarity settings |

Pin Class

Encapsulates Linux GPIO pins using [libsimpleio](#).

▪ Inheritance Hierarchy

[SystemObject](#) [IO.Objects.libsimpleio.GPIOPin](#)

Namespace: [IO.Objects.libsimpleio.GPIO](#)

Assembly: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

▪ 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 |
|--|--|
|  fd | Read-only property returning the Linux file descriptor for the GPIO pin. |



[state](#)

Read/Write GPIO state property.

[Top](#)

◀ See Also

Reference

[IO.Objects.libsimpleio.GPIO Namespace](#)

Pin Constructor

Constructor for a single GPIO pin.

Namespace: [IO.Objects.libsimpleio.GPIO](#)

Assembly: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

◀ Syntax

C# VB F#

[Copy](#)

```
public Pin(  
    Designator desg,  
    Direction dir,  
    bool state = false,  
    PinDriver driver = PinDriver.PushPull,  
    PinEdge edge = PinEdge.None,  
    PinPolarity polarity = PinPolarity.ActiveHigh  
)
```

Parameters

desg

Type: [IO.Objects.libsimpleio.DeviceDesignator](#)
GPIO pin designator.

dir

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

state (Optional)

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

driver (Optional)

Type: [IO.Objects.libsimpleio.GPIOPinDriver](#)
Output driver setting.

***edge* (Optional)**

Type: [IO.Objects.libsimpleio.GPIOPinEdge](#)
Interrupt edge setting.

***polarity* (Optional)**

Type: [IO.Objects.libsimpleio.GPIOPinPolarity](#)
Polarity setting.

◀ See Also

Reference

[Pin Class](#)

[IO.Objects.libsimpleio.GPIO Namespace](#)

Pin Properties

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

Properties

| | Name | Description |
|--|-----------------------|--|
| | fd | Read-only property returning the Linux file descriptor for the GPIO pin. |
| | state | Read/Write GPIO state property. |

[Top](#)

See Also

[Reference](#)

[Pin Class](#)

[IO.Objects.libsimpleio.GPIO Namespace](#)

Pinfd Property

Read-only property returning the Linux file descriptor for the GPIO pin.

Namespace: [IO.Objects.libsimpleio.GPIO](#)

Assembly: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

↳ Syntax

C# VB F#

[Copy](#)

```
public int fd { get; }
```

Property Value

Type: [Int32](#)

↳ See Also

Reference

[Pin Class](#)

[IO.Objects.libsimpleio.GPIO Namespace](#)

Pinstate Property

Read/Write GPIO state property.

Namespace: [IO.Objects.libsimpleio.GPIO](#)

Assembly: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

▪ Syntax

C# VB F#

[Copy](#)

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

Property Value

Type: [Boolean](#)

Implements

[Pinstate](#)

▪ See Also

[Reference](#)

[Pin Class](#)

[IO.Objects.libsimpleio.GPIO Namespace](#)

PinDriver Enumeration

GPIO output driver settings.

Namespace: [IO.Objects.libsimpleio.GPIO](#)

Assembly: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

▪ Syntax

C# VB F#

[Copy](#)

```
public enum Driver
```

▪ Members

| Member name | Value | Description |
|-------------|-------|--|
| PushPull | 0 | Push Pull (current source/sink) output driver. |
| OpenDrain | 1 | Open Drain (current sink) output driver. |
| OpenSource | 2 | Open Source (current source) output driver. |

▪ See Also

Reference

[IO.Objects.libsimpleio.GPIO Namespace](#)

PinEdge Enumeration

GPIO input interrupt edge settings.

Namespace: [IO.Objects.libsimpleio.GPIO](#)

Assembly: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

▪ Syntax

C# VB F#

[Copy](#)

```
public enum Edge
```

▪ Members

| Member name | Value | Description |
|-------------|-------|---|
| None | 0 | Configure GPIO input pin with interrupt disabled. |
| Rising | 1 | Configure GPIO input pin to interrupt on rising edge. |
| Falling | 2 | Configure GPIO pin to interrupt on falling edge. |
| Both | 3 | Configure GPIO pin to interrupt on both edges. |

▪ See Also

Reference

[IO.Objects.libsimpleio.GPIO Namespace](#)

PinPolarity Enumeration

GPIO polarity settings

Namespace: [IO.Objects.libsimpleio.GPIO](#)

Assembly: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

▪ Syntax

C# VB F#

[Copy](#)

```
public enum Polarity
```

▪ Members

| Member name | Value | Description |
|-------------|-------|--|
| ActiveLow | 0 | Configure GPIO pin as active low (inverted logic). |
| ActiveHigh | 1 | Configure GPIO pin as active high (normal logic). |

▪ See Also

Reference

[IO.Objects.libsimpleio.GPIO Namespace](#)

IO.Objects.libsimpleio.HID Namespace

Raw HID (Human Interface Device) Services

↳ Classes

| Class | Description |
|---|--|
|  Messenger | Encapsulates Linux raw HID devices using libsimpleio . |

Messenger Class

Encapsulates Linux raw HID devices using [libsimpleio](#).

▪ Inheritance Hierarchy

[SystemObject](#) [IO.Objects.libsimpleio.HIDMessenger](#)

Namespace: [IO.Objects.libsimpleio.HID](#)

Assembly: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

▪ Syntax

C# VB F#

[Copy](#)

```
public class Messenger : Messenger
```

The Messenger type exposes the following members.

▪ Constructors

| | Name | Description |
|---|--|--|
| ≡ | Messenger(String, Int32) | Constructor for a single raw HID device. |
| ≡ | Messenger(Int32, Int32, String, Int32) | Constructor for a single raw HID device. |

[Top](#)

▪ Properties

| | Name | Description |
|---|-------------------------|--|
|  | bustype | Read-only property returning the bus type identifierfor a raw HID device. |
|  | fd | Read-only property returning the Linux file descriptor for a raw HID device. |
|  | name | Read-only property returning the device information string for a raw HID device. |
|  | product | Read-only property returning the vendor identifierfor a raw HID device. |
|  | vendor | Read-only property returning the bus type identifierfor a raw HID device. |

[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.libsimpleio.HID Namespace](#)

Messenger Constructor

↳ Overload List

| | Name | Description |
|---|--|--|
| ≡ | Messenger(String, Int32) | Constructor for a single raw HID device. |
| ≡ | Messenger(Int32, Int32, String, Int32) | Constructor for a single raw HID device. |

[Top](#)

↳ See Also

Reference

[Messenger Class](#)

[IO.Objects.libsimpleio.HID Namespace](#)

Messenger Constructor (String, Int32)

Constructor for a single raw HID device.

Namespace: [IO.Objects.libsimpleio.HID](#)

Assembly: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

▪ Syntax

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

[Copy](#)

```
public Messenger(  
    string devname,  
    int timeoutms = 1000  
)
```

Parameters

devname

Type: [SystemString](#)

Device node name.

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](#)

[Messenger Overload](#)

IO.Objects.libsimpleio.HID Namespace

Messenger Constructor (Int32, Int32, String, Int32)

Constructor for a single raw HID device.

Namespace: [IO.Objects.libsimpleio.HID](#)

Assembly: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

◀ Syntax

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

[Copy](#)

```
public Messenger(  
    int VID = 5840,  
    int PID = 2810,  
    string serial = null,  
    int timeoutms = 1000  
)
```

Parameters

VID (Optional)

Type: [SystemInt32](#)

Vendor ID.

PID (Optional)

Type: [SystemInt32](#)

Product ID.

serial (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](#)

[Messenger Overload](#)

[IO.Objects.libsimpleio.HID Namespace](#)

Messenger Properties

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

Properties

| | Name | Description |
|--|-------------------------|--|
| | bustype | Read-only property returning the bus type identifierfor a raw HID device. |
| | fd | Read-only property returning the Linux file descriptor for a raw HID device. |
| | name | Read-only property returning the device information string for a raw HID device. |
| | product | Read-only property returning the vendor identifierfor a raw HID device. |
| | vendor | Read-only property returning the bus type identifierfor a raw HID device. |

[Top](#)

See Also

Reference

[Messenger Class](#)

[IO.Objects.libsimpleio.HID Namespace](#)

Messengerbustype Property

Read-only property returning the bus type identifier for a raw HID device.

Namespace: [IO.Objects.libsimpleio.HID](#)

Assembly: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

◀ Syntax

C# VB F#

[Copy](#)

```
public int bustype { get; }
```

Property Value

Type: [Int32](#)

◀ See Also

Reference

[Messenger Class](#)

[IO.Objects.libsimpleio.HID Namespace](#)

Messengerfd Property

Read-only property returning the Linux file descriptor for a raw HID device.

Namespace: [IO.Objects.libsimpleio.HID](#)

Assembly: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

◀ Syntax

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

[Copy](#)

```
public int fd { get; }
```

Property Value

Type: [Int32](#)

◀ See Also

Reference

[Messenger Class](#)

[IO.Objects.libsimpleio.HID Namespace](#)

Messengername Property

Read-only property returning the device information string for a raw HID device.

Namespace: [IO.Objects.libsimpleio.HID](#)

Assembly: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

◀ Syntax

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

[Copy](#)

```
public string name { get; }
```

Property Value

Type: [String](#)

◀ See Also

Reference

[Messenger Class](#)

[IO.Objects.libsimpleio.HID Namespace](#)

Messengerproduct Property

Read-only property returning the vendor identifierfor a raw HID device.

Namespace: [IO.Objects.libsimpleio.HID](#)

Assembly: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

◀ Syntax

C# VB F#

[Copy](#)

```
public int product { get; }
```

Property Value

Type: [Int32](#)

◀ See Also

Reference

[Messenger Class](#)

[IO.Objects.libsimpleio.HID Namespace](#)

Messengervendor Property

Read-only property returning the bus type identifierfor a raw HID device.

Namespace: [IO.Objects.libsimpleio.HID](#)

Assembly: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

◀ Syntax

C# VB F#

[Copy](#)

```
public int vendor { get; }
```

Property Value

Type: [Int32](#)

◀ See Also

Reference

[Messenger Class](#)

[IO.Objects.libsimpleio.HID 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.libsimpleio.HID Namespace](#)

MessengerReceive Method

Receive a 64-byte response message from a raw HID device.

Namespace: [IO.Objects.libsimpleio.HID](#)

Assembly: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

◀ 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.libsimpleio.HID Namespace](#)

MessengerSend Method

Send a 64-byte command message to a raw HID device.

Namespace: [IO.Objects.libsimpleio.HID](#)

Assembly: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

◀ 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.libsimpleio.HID Namespace](#)

MessengerTransaction Method

Send a 64-byte command message and receive a 64-byte response message.

Namespace: [IO.Objects.libsimpleio.HID](#)

Assembly: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

◀ 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.libsimpleio.HID Namespace

IO.Objects.libsimpleio.I2C Namespace

I²C (Inter-Integrated Circuit) Bus Controller Services

↳ Classes

| | Class | Description |
|---|---------------------|---|
|  | Bus | Encapsulates Linux I ² C bus controllers using libsimpleio . |

Bus Class

Encapsulates Linux I²C bus controllers using [libsimpleio](#).

▪ Inheritance Hierarchy

[SystemObject](#) [IO.Objects.libsimpleio.I2CBus](#)

Namespace: [IO.Objects.libsimpleio.I2C](#)

Assembly: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

▪ Syntax

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

[Copy](#)

```
public class Bus : Bus
```

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

▪ Constructors

| | Name | Description |
|---|---------------------------------|---|
| ≡ | Bus(Designator) | Constructor for a single I ² C bus controller. |
| ≡ | Bus(String) | Constructor for a single I ² C bus controller. |

[Top](#)

▪ Properties

| | Name | Description |
|---|--------------------|---|
|  | fd | Read-only property returning the Linux file descriptor for the I ² C bus controller. |

[Top](#)

◀ Methods

| | Name | Description |
|---|-----------------------------|---|
|  | Read | Read bytes from an I ² C device. |
|  | Transaction | Write and receive bytes to/from an I ² C device. |
|  | Write | Write bytes to an I ² C device. |

[Top](#)

◀ See Also

Reference

[IO.Objects.libsimpleio.I2C Namespace](#)

Bus Constructor

↳ Overload List

| | Name | Description |
|---|---------------------------------|---|
| ≡ | Bus(Designator) | Constructor for a single I ² C bus controller. |
| ≡ | Bus(String) | Constructor for a single I ² C bus controller. |

[Top](#)

↳ See Also

Reference

[Bus Class](#)

[IO.Objects.libsimpleio.I2C Namespace](#)

Bus Constructor (Designator)

Constructor for a single I²C bus controller.

Namespace: [IO.Objects.libsimpleio.I2C](#)

Assembly: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

▪ Syntax

C# VB F#

[Copy](#)

```
public Bus(  
    Designator desg  
)
```

Parameters

desg

Type: [IO.Objects.libsimpleio.DeviceDesignator](#)

I² bus designator.

▪ See Also

Reference

[Bus Class](#)

[Bus Overload](#)

[IO.Objects.libsimpleio.I2C Namespace](#)

Bus Constructor (String)

Constructor for a single I²C bus controller.

Namespace: [IO.Objects.libsimpleio.I2C](#)

Assembly: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

▪ Syntax

C# VB F#

[Copy](#)

```
public Bus(  
    string devname  
)
```

Parameters

devname

Type: [SystemString](#)

Device node name.

▪ See Also

Reference

[Bus Class](#)

[Bus Overload](#)

[IO.Objects.libsimpleio.I2C Namespace](#)

Bus Properties

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

▪ Properties

| | Name | Description |
|---|------|---|
|  | fd | Read-only property returning the Linux file descriptor for the I ² C bus controller. |

[Top](#)

▪ See Also

[Reference](#)

[Bus Class](#)

[IO.Objects.libsimpleio.I2C Namespace](#)

Busfd Property

Read-only property returning the Linux file descriptor for the I²C bus controller.

Namespace: [IO.Objects.libsimpleio.I2C](#)

Assembly: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

◀ Syntax

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

[Copy](#)

```
public int fd { get; }
```

Property Value

Type: [Int32](#)

◀ See Also

Reference

[Bus Class](#)

[IO.Objects.libsimpleio.I2C Namespace](#)

Bus Methods

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

▪ Methods

| | Name | Description |
|--|-----------------------------|---|
|  | Read | Read bytes from an I ² C device. |
|  | Transaction | Write and receive bytes to/from an I ² C device. |
|  | Write | Write bytes to an I ² C device. |

[Top](#)

▪ See Also

Reference

[Bus Class](#)

[IO.Objects.libsimpleio.I2C Namespace](#)

BusRead Method

Read bytes from an I²C device.

Namespace: [IO.Objects.libsimpleio.I2C](#)

Assembly: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

↳ Syntax

C# VB F#

[Copy](#)

```
public void Read(  
    int slaveaddr,  
    byte[] resp,  
    int resplen  
)
```

Parameters

slaveaddr

Type: [SystemInt32](#)

Slave device address.

resp

Type: [SystemByte](#)

Response buffer.

resplen

Type: [SystemInt32](#)

Number of bytes to read.

Implements

[BusRead\(Int32, Byte, Int32\)](#)

↳ See Also

Reference

[Bus Class](#)

[IO.Objects.libsimpleio.I2C Namespace](#)

BusTransaction Method

Write and receive bytes to/from an I²C device.

Namespace: [IO.Objects.libsimpleio.I2C](#)

Assembly: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

↳ Syntax

C# VB F#

[Copy](#)

```
public void Transaction(  
    int slaveaddr,  
    byte[] cmd,  
    int cmdlen,  
    byte[] resp,  
    int resplen,  
    int delayus = 0  
)
```

Parameters

slaveaddr

Type: [SystemInt32](#)

Device 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.

Implements

[BusTransaction\(Int32, Byte, Int32, Byte, Int32, Int32\)](#)

See Also

Reference

[Bus Class](#)

[IO.Objects.libsimpleio.I2C Namespace](#)

BusWrite Method

Write bytes to an I²C device.

Namespace: [IO.Objects.libsimpleio.I2C](#)

Assembly: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

↳ Syntax

C# VB F#

[Copy](#)

```
public void Write(  
    int slaveaddr,  
    byte[] cmd,  
    int cmdlen  
)
```

Parameters

slaveaddr

Type: [SystemInt32](#)

Slave device address.

cmd

Type: [SystemByte](#)

Command buffer.

cmdlen

Type: [SystemInt32](#)

Number of bytes to write.

Implements

[BusWrite\(Int32, Byte, Int32\)](#)

↳ See Also

Reference

[Bus Class](#)

[IO.Objects.libsimpleio.I2C Namespace](#)

IO.Objects.libsimpleio.mikroBUS Namespace

Mikroelektronika mikroBUS (<https://www.mikroe.com/mikrobus>) Shield and Socket Services

▪ Classes

| | Class | Description |
|---|------------------------|---|
|  | Shield | Encapsulates mikroBUS shields (add-on boards providing mikroBUS sockets). |
|  | Socket | Encapsulates mikroBUS sockets. |

▪ Enumerations

| | Enumeration | Description |
|---|-----------------------------|-----------------------------|
|  | ShieldKinds | Supported mikroBUS shields. |

Shield Class

Encapsulates mikroBUS shields (add-on boards providing [mikroBUS](#) sockets).

▪ Inheritance Hierarchy

[SystemObject](#) [IO.Objects.libsimpleio.mikroBUSShield](#)

Namespace: [IO.Objects.libsimpleio.mikroBUS](#)

Assembly: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

▪ 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 target board, as obtained from the SHIELDNAME environment variable or guessed from the BOARDNAME environment variable. The guessed value for BeagleBone family target boards (BOARDNAME == "BeagleBone*") is BeagleBoneClick4 . |

The guessed value for Raspberry Pi family target boards (`BOARDNAME == "RaspberryPi*"`) is [PiClick3](#).

[Top](#)

↳ Fields

| | Name | Description |
|---|------------------------|---|
| •   | I2CBus | Shared I ² C bus that is common to all sockets on this shield. |

[Top](#)

↳ See Also

Reference

[IO.Objects.libsimpleio.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 target board, as obtained from the <code>SHIELDNAME</code> environment variable or guessed from the <code>BOARDNAME</code> environment variable. The guessed value for BeagleBone family target boards (<code>BOARDNAME == "BeagleBone*"</code>) is <code>BeagleBoneClick4</code> . The guessed value for Raspberry Pi family target boards (<code>BOARDNAME == "RaspberryPi*"</code>) is <code>PiClick3</code> . |

[Top](#)

See Also

Reference

[Shield Class](#)

[IO.Objects.libsimpleio.mikroBUS Namespace](#)

Shieldkind Property

Returns the kind of mikroBUS shield that is installed on the target board, as obtained from the `SHIELDNAME` environment variable or guessed from the `BOARDNAME` environment variable. The guessed value for BeagleBone family target boards (`BOARDNAME == "BeagleBone*"`) is `BeagleBoneClick4`. The guessed value for Raspberry Pi family target boards (`BOARDNAME == "RaspberryPi*"`) is `PiClick3`.

Namespace: [IO.Objects.libsimpleio.mikroBUS](#)

Assembly: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

↳ Syntax

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

[Copy](#)

```
public static ShieldKinds kind { get; }
```

Property Value

Type: [ShieldKinds](#)

↳ See Also

[Reference](#)

[Shield Class](#)

[IO.Objects.libsimpleio.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.Objects.libsimpleio.mikroBUS Namespace](#)

ShieldI2CBus Field

Shared I²C bus that is common to all sockets on this shield.

Namespace: [IO.Objects.libsimpleio.mikroBUS](#)

Assembly: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

↳ Syntax

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

[Copy](#)

```
public static Bus I2CBus
```

Field Value

Type: [Bus](#)

↳ See Also

[Reference](#)

[Shield Class](#)

[IO.Objects.libsimpleio.mikroBUS Namespace](#)

ShieldKinds Enumeration

Supported mikroBUS shields.

Namespace: [IO.Objects.libsimpleio.mikroBUS](#)

Assembly: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

▪ Syntax

C# VB F#

[Copy](#)

```
public enum Kinds
```

▪ Members

| Member name | Value | Description |
|------------------|-------|---|
| BeagleBoneClick2 | 0 | Mikroelektronika BeagleBone Click Shield MIKROE-1596 , with two mikroBUS sockets. (Obsolete, but still useful.) |
| BeagleBoneClick4 | 1 | Mikroelektronika mikroBUS Cape MIKROE-1857 with four mikroBUS sockets. |

| | | |
|--------------|---|--|
| PiClick1 | 2 | Mikroelektronika Pi Click Shield MIKROE-1512/1513 for 26-pin expansion header, with one mikroBUS socket (Obsolete.) |
| PiClick2 | 3 | Mikroelektronika Pi 2 Click Shield MIKROE-1879 for 40-pin expansion header, with two mikroBUS sockets. |
| PiClick3 | 4 | Mikroelektronika Pi 3 Click Shield MIKROE-2756 for 40-pin expansion header, with selectable on- board A/D converter and two mikroBUS sockets. |
| PocketBeagle | 5 | PocketBeagle with female headers on top, with two |

mikroBUS
sockets.

Unknown

2147483647

No known
mikroBUS shield
installed.

↳ See Also

Reference

[IO.Objects.libsimpleio.mikroBUS Namespace](#)

Socket Class

Encapsulates mikroBUS sockets.

▪ Inheritance Hierarchy

[SystemObject](#) [IO.Objects.libsimpleio.mikroBUSSocket](#)

Namespace: [IO.Objects.libsimpleio.mikroBUS](#)

Assembly: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

▪ 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 this socket. |
| | 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.



[UART](#)

Returns the UART device name for this socket.

[Top](#)

« See Also

Reference

[IO.Objects.libsimpleio.mikroBUS Namespace](#)

Socket Constructor

Constructor for a single mikroBUS socket.

Namespace: [IO.Objects.libsimpleio.mikroBUS](#)

Assembly: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

▪ Syntax

C# VB F#

[Copy](#)

```
public Socket(  
    int num,  
    ShieldKinds shield = ShieldKinds.Unknown  
)
```

Parameters

num

Type: [SystemInt32](#)

Socket number.

shield (Optional)

Type: [IO.Objects.libsimpleio.mikroBUSShieldKinds](#)

mikroBUS shield kind. `Shield.Kinds.Unknown` indicates automatic detection using the `Shield.kind` property.

▪ See Also

[Reference](#)

[Socket Class](#)

[IO.Objects.libsimpleio.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 this socket. |

| | | |
|---|--------|--|
| | 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. |
|  | UART | Returns the UART device name for this socket. |

[Top](#)

See Also

Reference

[Socket Class](#)

[IO.Objects.libsimpleio.mikroBUS Namespace](#)

SocketAIN Property

Returns the ADC input designator for AN.

Namespace: [IO.Objects.libsimpleio.mikroBUS](#)

Assembly: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

« Syntax

C# VB F#

[Copy](#)

```
public Designator AIN { get; }
```

Property Value

Type: [Designator](#)

« See Also

Reference

[Socket Class](#)

[IO.Objects.libsimpleio.mikroBUS Namespace](#)

SocketAN Property

Returns the GPIO pin designator for AN.

Namespace: [IO.Objects.libsimpleio.mikroBUS](#)

Assembly: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

↳ Syntax

C# VB F#

[Copy](#)

```
public Designator AN { get; }
```

Property Value

Type: [Designator](#)

↳ See Also

Reference

[Socket Class](#)

[IO.Objects.libsimpleio.mikroBUS Namespace](#)

SocketCS Property

Returns the GPIO pin designator for CS.

Namespace: [IO.Objects.libsimpleio.mikroBUS](#)

Assembly: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

« Syntax

C# VB F#

[Copy](#)

```
public Designator CS { get; }
```

Property Value

Type: [Designator](#)

« See Also

Reference

[Socket Class](#)

[IO.Objects.libsimpleio.mikroBUS Namespace](#)

SocketI2CBus Property

Returns the I²C bus designator for this socket.

Namespace: [IO.Objects.libsimpleio.mikroBUS](#)

Assembly: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

« Syntax

C# VB F#

[Copy](#)

```
public Designator I2CBus { get; }
```

Property Value

Type: [Designator](#)

« See Also

Reference

[Socket Class](#)

[IO.Objects.libsimpleio.mikroBUS Namespace](#)

SocketINT Property

Returns the GPIO pin designator for INT.

Namespace: [IO.Objects.libsimpleio.mikroBUS](#)

Assembly: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

« Syntax

C# VB F#

[Copy](#)

```
public Designator INT { get; }
```

Property Value

Type: [Designator](#)

« See Also

Reference

[Socket Class](#)

[IO.Objects.libsimpleio.mikroBUS Namespace](#)

SocketMISO Property

Returns the GPIO pin designator for MISO.

Namespace: [IO.Objects.libsimpleio.mikroBUS](#)

Assembly: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

« Syntax

C# VB F#

[Copy](#)

```
public Designator MISO { get; }
```

Property Value

Type: [Designator](#)

« See Also

Reference

[Socket Class](#)

[IO.Objects.libsimpleio.mikroBUS Namespace](#)

SocketMOSI Property

Returns the GPIO pin designator for MOSI.

Namespace: [IO.Objects.libsimpleio.mikroBUS](#)

Assembly: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

◀ Syntax

C# VB F#

[Copy](#)

```
public Designator MOSI { get; }
```

Property Value

Type: [Designator](#)

◀ See Also

Reference

[Socket Class](#)

[IO.Objects.libsimpleio.mikroBUS Namespace](#)

SocketPWM Property

Returns the GPIO pin designator for PWM.

Namespace: [IO.Objects.libsimpleio.mikroBUS](#)

Assembly: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

« Syntax

C# VB F#

[Copy](#)

```
public Designator PWM { get; }
```

Property Value

Type: [Designator](#)

« See Also

Reference

[Socket Class](#)

[IO.Objects.libsimpleio.mikroBUS Namespace](#)

SocketPWMOut Property

Returns the PWM output designator for this socket.

Namespace: [IO.Objects.libsimpleio.mikroBUS](#)

Assembly: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

◀ Syntax

C# VB F#

[Copy](#)

```
public Designator PWMOut { get; }
```

Property Value

Type: [Designator](#)

◀ See Also

Reference

[Socket Class](#)

[IO.Objects.libsimpleio.mikroBUS Namespace](#)

SocketRST Property

Returns the GPIO pin designator for RST.

Namespace: [IO.Objects.libsimpleio.mikroBUS](#)

Assembly: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

« Syntax

C# VB F#

[Copy](#)

```
public Designator RST { get; }
```

Property Value

Type: [Designator](#)

« See Also

Reference

[Socket Class](#)

[IO.Objects.libsimpleio.mikroBUS Namespace](#)

SocketRX Property

Returns the GPIO pin designator for RX.

Namespace: [IO.Objects.libsimpleio.mikroBUS](#)

Assembly: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

« Syntax

C# VB F#

[Copy](#)

```
public Designator RX { get; }
```

Property Value

Type: [Designator](#)

« See Also

Reference

[Socket Class](#)

[IO.Objects.libsimpleio.mikroBUS Namespace](#)

SocketSCK Property

Returns the GPIO pin designator for SCK.

Namespace: [IO.Objects.libsimpleio.mikroBUS](#)

Assembly: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

« Syntax

C# VB F#

[Copy](#)

```
public Designator SCK { get; }
```

Property Value

Type: [Designator](#)

« See Also

Reference

[Socket Class](#)

[IO.Objects.libsimpleio.mikroBUS Namespace](#)

SocketSCL Property

Returns the GPIO pin designator for SCL.

Namespace: [IO.Objects.libsimpleio.mikroBUS](#)

Assembly: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

« Syntax

C# VB F#

[Copy](#)

```
public Designator SCL { get; }
```

Property Value

Type: [Designator](#)

« See Also

Reference

[Socket Class](#)

[IO.Objects.libsimpleio.mikroBUS Namespace](#)

SocketSDA Property

Returns the GPIO pin designator for SDA.

Namespace: [IO.Objects.libsimpleio.mikroBUS](#)

Assembly: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

« Syntax

C# VB F#

[Copy](#)

```
public Designator SDA { get; }
```

Property Value

Type: [Designator](#)

« See Also

Reference

[Socket Class](#)

[IO.Objects.libsimpleio.mikroBUS Namespace](#)

SocketSPIDev Property

Returns the SPI device designator for this socket.

Namespace: [IO.Objects.libsimpleio.mikroBUS](#)

Assembly: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

« Syntax

C# VB F#

[Copy](#)

```
public Designator SPIDev { get; }
```

Property Value

Type: [Designator](#)

« See Also

Reference

[Socket Class](#)

[IO.Objects.libsimpleio.mikroBUS Namespace](#)

SocketTX Property

Returns the GPIO pin designator for TX.

Namespace: [IO.Objects.libsimpleio.mikroBUS](#)

Assembly: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

« Syntax

C# VB F#

[Copy](#)

```
public Designator TX { get; }
```

Property Value

Type: [Designator](#)

« See Also

Reference

[Socket Class](#)

[IO.Objects.libsimpleio.mikroBUS Namespace](#)

SocketUART Property

Returns the UART device name for this socket.

Namespace: [IO.Objects.libsimpleio.mikroBUS](#)

Assembly: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

◀ Syntax

C# VB F#

[Copy](#)

```
public string UART { get; }
```

Property Value

Type: [String](#)

◀ See Also

Reference

[Socket Class](#)

[IO.Objects.libsimpleio.mikroBUS Namespace](#)

IO.Objects.libsimpleio.Platforms Namespace

Platform Definition Classes

↳ Classes

| Class | Description |
|--|--|
|  BeagleBone | This class defines identifiers for the devices provided by the BeagleBone hardware platform. |
|  PocketBeagle | This class defines identifiers for the devices provided by the PocketBeagle hardware platform. |
|  RaspberryPi | This class defines identifiers for the devices provided by the Raspberry Pi hardware platform. |

BeagleBone Class

This class defines identifiers for the devices provided by the BeagleBone hardware platform.

▪ Inheritance Hierarchy

[SystemObject](#) [IO.Objects.libsimpleio.Platforms](#)[BeagleBone](#)

Namespace: [IO.Objects.libsimpleio.Platforms](#)

Assembly: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

▪ Syntax

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

[Copy](#)

```
public static class BeagleBone
```

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

▪ Fields

| | Name | Description |
|----------------------------|----------------------|---------------------------------------|
| • S | AIN0 | ADC input designator for P9.39 (1.8V) |
| • S | AIN1 | ADC input designator for P9.40 (1.8V) |
| • S | AIN2 | ADC input designator for P9.37 (1.8V) |
| • S | AIN3 | ADC input designator for P9.38 (1.8V) |
| • S | AIN4 | ADC input designator for P9.33 (1.8V) |

| | | |
|-----|---------|---------------------------------------|
| ♦ S | AIN5 | ADC input designator for P9.36 (1.8V) |
| ♦ S | AIN6 | ADC input designator for P9.35 (1.8V) |
| ♦ S | GPIO10 | Legacy GPIO pin designator for P8.31 |
| ♦ S | GPIO11 | Legacy GPIO pin designator for P8.32 |
| ♦ S | GPIO110 | Legacy GPIO pin designator for P9.31 |
| ♦ S | GPIO111 | Legacy GPIO pin designator for P9.29 |
| ♦ S | GPIO112 | Legacy GPIO pin designator for P9.30 |
| ♦ S | GPIO113 | Legacy GPIO pin designator for P9.28 |
| ♦ S | GPIO115 | Legacy GPIO pin designator for P9.27 |
| ♦ S | GPIO117 | Legacy GPIO pin designator for P9.25 |
| ♦ S | GPIO12 | Legacy GPIO pin designator for P9.20 |
| ♦ S | GPIO13 | Legacy GPIO pin designator for P9.19 |
| ♦ S | GPIO14 | Legacy GPIO pin designator for P9.26 |
| ♦ S | GPIO15 | Legacy GPIO pin designator for P9.24 |
| ♦ S | GPIO2 | Legacy GPIO pin designator for P9.22 |
| ♦ S | GPIO20 | Legacy GPIO pin designator for P9.41 |
| ♦ S | GPIO22 | Legacy GPIO pin designator for P8.19 |
| ♦ S | GPIO23 | Legacy GPIO pin designator for P8.13 |
| ♦ S | GPIO26 | Legacy GPIO pin designator for P8.14 |

| | | |
|---|--------|--------------------------------------|
| ♦ | GPIO27 | Legacy GPIO pin designator for P8.17 |
| ♦ | GPIO3 | Legacy GPIO pin designator for P9.21 |
| ♦ | GPIO30 | Legacy GPIO pin designator for P9.11 |
| ♦ | GPIO31 | Legacy GPIO pin designator for P9.13 |
| ♦ | GPIO32 | Legacy GPIO pin designator for P8.25 |
| ♦ | GPIO33 | Legacy GPIO pin designator for P8.24 |
| ♦ | GPIO34 | Legacy GPIO pin designator for P8.5 |
| ♦ | GPIO35 | Legacy GPIO pin designator for P8.6 |
| ♦ | GPIO36 | Legacy GPIO pin designator for P8.23 |
| ♦ | GPIO37 | Legacy GPIO pin designator for P8.22 |
| ♦ | GPIO38 | Legacy GPIO pin designator for P8.3 |
| ♦ | GPIO39 | Legacy GPIO pin designator for P8.4 |
| ♦ | GPIO4 | Legacy GPIO pin designator for P9.18 |
| ♦ | GPIO44 | Legacy GPIO pin designator for P8.12 |
| ♦ | GPIO45 | Legacy GPIO pin designator for P8.11 |
| ♦ | GPIO46 | Legacy GPIO pin designator for P8.16 |
| ♦ | GPIO47 | Legacy GPIO pin designator for P8.15 |
| ♦ | GPIO48 | Legacy GPIO pin designator for P9.15 |
| ♦ | GPIO49 | Legacy GPIO pin designator for P9.23 |

| | | |
|-----|--------|--------------------------------------|
| • S | GPIO5 | Legacy GPIO pin designator for P9.17 |
| • S | GPIO50 | Legacy GPIO pin designator for P9.14 |
| • S | GPIO51 | Legacy GPIO pin designator for P9.16 |
| • S | GPIO60 | Legacy GPIO pin designator for P9.12 |
| • S | GPIO61 | Legacy GPIO pin designator for P8.26 |
| • S | GPIO62 | Legacy GPIO pin designator for P8.21 |
| • S | GPIO63 | Legacy GPIO pin designator for P8.20 |
| • S | GPIO65 | Legacy GPIO pin designator for P8.18 |
| • S | GPIO66 | Legacy GPIO pin designator for P8.7 |
| • S | GPIO67 | Legacy GPIO pin designator for P8.8 |
| • S | GPIO68 | Legacy GPIO pin designator for P8.10 |
| • S | GPIO69 | Legacy GPIO pin designator for P8.9 |
| • S | GPIO7 | Legacy GPIO pin designator for P9.42 |
| • S | GPIO70 | Legacy GPIO pin designator for P8.45 |
| • S | GPIO71 | Legacy GPIO pin designator for P8.46 |
| • S | GPIO72 | Legacy GPIO pin designator for P8.43 |
| • S | GPIO73 | Legacy GPIO pin designator for P8.44 |
| • S | GPIO74 | Legacy GPIO pin designator for P8.41 |
| • S | GPIO75 | Legacy GPIO pin designator for P8.42 |

| | | |
|------------|------------------------|--|
| • S | GPIO76 | Legacy GPIO pin designator for P8.39 |
| • S | GPIO77 | Legacy GPIO pin designator for P8.40 |
| • S | GPIO78 | Legacy GPIO pin designator for P8.37 |
| • S | GPIO79 | Legacy GPIO pin designator for P8.38 |
| • S | GPIO8 | Legacy GPIO pin designator for P8.35 |
| • S | GPIO80 | Legacy GPIO pin designator for P8.36 |
| • S | GPIO81 | Legacy GPIO pin designator for P8.34 |
| • S | GPIO86 | Legacy GPIO pin designator for P8.27 |
| • S | GPIO87 | Legacy GPIO pin designator for P8.29 |
| • S | GPIO88 | Legacy GPIO pin designator for P8.28 |
| • S | GPIO89 | Legacy GPIO pin designator for P8.30 |
| • S | GPIO9 | Legacy GPIO pin designator for P8.33 |
| • S | I2C2 | I2C bus designator for P9.19 and P9.20 |
| • S | SPI2_0 | SPI slave select designator for P9.28 |
| • S | SPI2_1 | SPI slave select designator for P9.42 |

[Top](#)

See Also

Reference

[IO.Objects.libsimpleio.Platforms Namespace](#)

BeagleBone Fields

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

Fields

| | Name | Description |
|---|---------|---------------------------------------|
| •   | AIN0 | ADC input designator for P9.39 (1.8V) |
| •   | AIN1 | ADC input designator for P9.40 (1.8V) |
| •   | AIN2 | ADC input designator for P9.37 (1.8V) |
| •   | AIN3 | ADC input designator for P9.38 (1.8V) |
| •   | AIN4 | ADC input designator for P9.33 (1.8V) |
| •   | AIN5 | ADC input designator for P9.36 (1.8V) |
| •   | AIN6 | ADC input designator for P9.35 (1.8V) |
| •   | GPIO10 | Legacy GPIO pin designator for P8.31 |
| •   | GPIO11 | Legacy GPIO pin designator for P8.32 |
| •   | GPIO110 | Legacy GPIO pin designator for P9.31 |
| •   | GPIO111 | Legacy GPIO pin designator for P9.29 |
| •   | GPIO112 | Legacy GPIO pin designator for P9.30 |
| •   | GPIO113 | Legacy GPIO pin designator for P9.28 |

| | | |
|-----|---------|--------------------------------------|
| ♦ S | GPIO115 | Legacy GPIO pin designator for P9.27 |
| ♦ S | GPIO117 | Legacy GPIO pin designator for P9.25 |
| ♦ S | GPIO12 | Legacy GPIO pin designator for P9.20 |
| ♦ S | GPIO13 | Legacy GPIO pin designator for P9.19 |
| ♦ S | GPIO14 | Legacy GPIO pin designator for P9.26 |
| ♦ S | GPIO15 | Legacy GPIO pin designator for P9.24 |
| ♦ S | GPIO2 | Legacy GPIO pin designator for P9.22 |
| ♦ S | GPIO20 | Legacy GPIO pin designator for P9.41 |
| ♦ S | GPIO22 | Legacy GPIO pin designator for P8.19 |
| ♦ S | GPIO23 | Legacy GPIO pin designator for P8.13 |
| ♦ S | GPIO26 | Legacy GPIO pin designator for P8.14 |
| ♦ S | GPIO27 | Legacy GPIO pin designator for P8.17 |
| ♦ S | GPIO3 | Legacy GPIO pin designator for P9.21 |
| ♦ S | GPIO30 | Legacy GPIO pin designator for P9.11 |
| ♦ S | GPIO31 | Legacy GPIO pin designator for P9.13 |
| ♦ S | GPIO32 | Legacy GPIO pin designator for P8.25 |
| ♦ S | GPIO33 | Legacy GPIO pin designator for P8.24 |
| ♦ S | GPIO34 | Legacy GPIO pin designator for P8.5 |
| ♦ S | GPIO35 | Legacy GPIO pin designator for P8.6 |

| | | |
|-----|--------|--------------------------------------|
| ♦ S | GPIO36 | Legacy GPIO pin designator for P8.23 |
| ♦ S | GPIO37 | Legacy GPIO pin designator for P8.22 |
| ♦ S | GPIO38 | Legacy GPIO pin designator for P8.3 |
| ♦ S | GPIO39 | Legacy GPIO pin designator for P8.4 |
| ♦ S | GPIO4 | Legacy GPIO pin designator for P9.18 |
| ♦ S | GPIO44 | Legacy GPIO pin designator for P8.12 |
| ♦ S | GPIO45 | Legacy GPIO pin designator for P8.11 |
| ♦ S | GPIO46 | Legacy GPIO pin designator for P8.16 |
| ♦ S | GPIO47 | Legacy GPIO pin designator for P8.15 |
| ♦ S | GPIO48 | Legacy GPIO pin designator for P9.15 |
| ♦ S | GPIO49 | Legacy GPIO pin designator for P9.23 |
| ♦ S | GPIO5 | Legacy GPIO pin designator for P9.17 |
| ♦ S | GPIO50 | Legacy GPIO pin designator for P9.14 |
| ♦ S | GPIO51 | Legacy GPIO pin designator for P9.16 |
| ♦ S | GPIO60 | Legacy GPIO pin designator for P9.12 |
| ♦ S | GPIO61 | Legacy GPIO pin designator for P8.26 |
| ♦ S | GPIO62 | Legacy GPIO pin designator for P8.21 |
| ♦ S | GPIO63 | Legacy GPIO pin designator for P8.20 |
| ♦ S | GPIO65 | Legacy GPIO pin designator for P8.18 |

| | | |
|-----|--------|--------------------------------------|
| ♦ S | GPIO66 | Legacy GPIO pin designator for P8.7 |
| ♦ S | GPIO67 | Legacy GPIO pin designator for P8.8 |
| ♦ S | GPIO68 | Legacy GPIO pin designator for P8.10 |
| ♦ S | GPIO69 | Legacy GPIO pin designator for P8.9 |
| ♦ S | GPIO7 | Legacy GPIO pin designator for P9.42 |
| ♦ S | GPIO70 | Legacy GPIO pin designator for P8.45 |
| ♦ S | GPIO71 | Legacy GPIO pin designator for P8.46 |
| ♦ S | GPIO72 | Legacy GPIO pin designator for P8.43 |
| ♦ S | GPIO73 | Legacy GPIO pin designator for P8.44 |
| ♦ S | GPIO74 | Legacy GPIO pin designator for P8.41 |
| ♦ S | GPIO75 | Legacy GPIO pin designator for P8.42 |
| ♦ S | GPIO76 | Legacy GPIO pin designator for P8.39 |
| ♦ S | GPIO77 | Legacy GPIO pin designator for P8.40 |
| ♦ S | GPIO78 | Legacy GPIO pin designator for P8.37 |
| ♦ S | GPIO79 | Legacy GPIO pin designator for P8.38 |
| ♦ S | GPIO8 | Legacy GPIO pin designator for P8.35 |
| ♦ S | GPIO80 | Legacy GPIO pin designator for P8.36 |
| ♦ S | GPIO81 | Legacy GPIO pin designator for P8.34 |
| ♦ S | GPIO86 | Legacy GPIO pin designator for P8.27 |

| | | |
|------------|------------------------|--|
| • S | GPIO87 | Legacy GPIO pin designator for P8.29 |
| • S | GPIO88 | Legacy GPIO pin designator for P8.28 |
| • S | GPIO89 | Legacy GPIO pin designator for P8.30 |
| • S | GPIO9 | Legacy GPIO pin designator for P8.33 |
| • S | I2C2 | I2C bus designator for P9.19 and P9.20 |
| • S | SPI2_0 | SPI slave select designator for P9.28 |
| • S | SPI2_1 | SPI slave select designator for P9.42 |

[Top](#)

↳ See Also

Reference

[BeagleBone Class](#)

[IO.Objects.libsimpleio.Platforms Namespace](#)

BeagleBoneAIN0 Field

ADC input designator for P9.39 (1.8V)

Namespace: [IO.Objects.libsimpleio.Platforms](#)

Assembly: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

▪ Syntax

C# VB F#

[Copy](#)

```
public static readonly Designator AIN0
```

Field Value

Type: [Designator](#)

▪ Remarks

Requires the [BB-ADC](#) device tree overlay.

▪ See Also

Reference

[BeagleBone Class](#)

[IO.Objects.libsimpleio.Platforms Namespace](#)

BeagleBoneAIN1 Field

ADC input designator for P9.40 (1.8V)

Namespace: [IO.Objects.libsimpleio.Platforms](#)

Assembly: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

▪ Syntax

C# VB F#

[Copy](#)

```
public static readonly Designator AIN1
```

Field Value

Type: [Designator](#)

▪ Remarks

Requires the [BB-ADC](#) device tree overlay.

▪ See Also

Reference

[BeagleBone Class](#)

[IO.Objects.libsimpleio.Platforms Namespace](#)

BeagleBoneAIN2 Field

ADC input designator for P9.37 (1.8V)

Namespace: [IO.Objects.libsimpleio.Platforms](#)

Assembly: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

▪ Syntax

C# VB F#

[Copy](#)

```
public static readonly Designator AIN2
```

Field Value

Type: [Designator](#)

▪ Remarks

Requires the [BB-ADC](#) device tree overlay.

▪ See Also

Reference

[BeagleBone Class](#)

[IO.Objects.libsimpleio.Platforms Namespace](#)

BeagleBoneAIN3 Field

ADC input designator for P9.38 (1.8V)

Namespace: [IO.Objects.libsimpleio.Platforms](#)

Assembly: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

▪ Syntax

C# VB F#

[Copy](#)

```
public static readonly Designator AIN3
```

Field Value

Type: [Designator](#)

▪ Remarks

Requires the [BB-ADC](#) device tree overlay.

▪ See Also

Reference

[BeagleBone Class](#)

[IO.Objects.libsimpleio.Platforms Namespace](#)

BeagleBoneAIN4 Field

ADC input designator for P9.33 (1.8V)

Namespace: [IO.Objects.libsimpleio.Platforms](#)

Assembly: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

▪ Syntax

C# VB F#

[Copy](#)

```
public static readonly Designator AIN4
```

Field Value

Type: [Designator](#)

▪ Remarks

Requires the [BB-ADC](#) device tree overlay.

▪ See Also

Reference

[BeagleBone Class](#)

[IO.Objects.libsimpleio.Platforms Namespace](#)

BeagleBoneAIN5 Field

ADC input designator for P9.36 (1.8V)

Namespace: [IO.Objects.libsimpleio.Platforms](#)

Assembly: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

▪ Syntax

C# VB F#

[Copy](#)

```
public static readonly Designator AIN5
```

Field Value

Type: [Designator](#)

▪ Remarks

Requires the [BB-ADC](#) device tree overlay.

▪ See Also

Reference

[BeagleBone Class](#)

[IO.Objects.libsimpleio.Platforms Namespace](#)

BeagleBoneAIN6 Field

ADC input designator for P9.35 (1.8V)

Namespace: [IO.Objects.libsimpleio.Platforms](#)

Assembly: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

▪ Syntax

C# VB F#

[Copy](#)

```
public static readonly Designator AIN6
```

Field Value

Type: [Designator](#)

▪ Remarks

Requires the [BB-ADC](#) device tree overlay.

▪ See Also

Reference

[BeagleBone Class](#)

[IO.Objects.libsimpleio.Platforms Namespace](#)

BeagleBoneGPIO10 Field

Legacy GPIO pin designator for P8.31

Namespace: [IO.Objects.libsimpleio.Platforms](#)

Assembly: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

▪ Syntax

C# VB F#

[Copy](#)

```
public static readonly Designator GPIO10
```

Field Value

Type: [Designator](#)

▪ Remarks

Requires the [BB-GPIO](#) device tree overlay.

▪ See Also

Reference

[BeagleBone Class](#)

[IO.Objects.libsimpleio.Platforms Namespace](#)

BeagleBoneGPIO11 Field

Legacy GPIO pin designator for P8.32

Namespace: [IO.Objects.libsimpleio.Platforms](#)

Assembly: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

▪ Syntax

C# VB F#

[Copy](#)

```
public static readonly Designator GPIO11
```

Field Value

Type: [Designator](#)

▪ Remarks

Requires the [BB-GPIO](#) device tree overlay.

▪ See Also

Reference

[BeagleBone Class](#)

[IO.Objects.libsimpleio.Platforms Namespace](#)

BeagleBoneGPIO110 Field

Legacy GPIO pin designator for P9.31

Namespace: [IO.Objects.libsimpleio.Platforms](#)

Assembly: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

▪ Syntax

C# VB F#

[Copy](#)

```
public static readonly Designator GPIO110
```

Field Value

Type: [Designator](#)

▪ Remarks

Requires the [BB-GPIO](#) device tree overlay.

▪ See Also

Reference

[BeagleBone Class](#)

[IO.Objects.libsimpleio.Platforms Namespace](#)

BeagleBoneGPIO111 Field

Legacy GPIO pin designator for P9.29

Namespace: [IO.Objects.libsimpleio.Platforms](#)

Assembly: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

▪ Syntax

C# VB F#

[Copy](#)

```
public static readonly Designator GPIO111
```

Field Value

Type: [Designator](#)

▪ Remarks

Requires the [BB-GPIO](#) device tree overlay.

▪ See Also

Reference

[BeagleBone Class](#)

[IO.Objects.libsimpleio.Platforms Namespace](#)

BeagleBoneGPIO112 Field

Legacy GPIO pin designator for P9.30

Namespace: [IO.Objects.libsimpleio.Platforms](#)

Assembly: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

▪ Syntax

C# VB F#

[Copy](#)

```
public static readonly Designator GPIO112
```

Field Value

Type: [Designator](#)

▪ Remarks

Requires the [BB-GPIO](#) device tree overlay.

▪ See Also

Reference

[BeagleBone Class](#)

[IO.Objects.libsimpleio.Platforms Namespace](#)

BeagleBoneGPIO113 Field

Legacy GPIO pin designator for P9.28

Namespace: [IO.Objects.libsimpleio.Platforms](#)

Assembly: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

▪ Syntax

C# VB F#

[Copy](#)

```
public static readonly Designator GPIO113
```

Field Value

Type: [Designator](#)

▪ Remarks

Requires the [BB-GPIO](#) device tree overlay.

▪ See Also

Reference

[BeagleBone Class](#)

[IO.Objects.libsimpleio.Platforms Namespace](#)

BeagleBoneGPIO115 Field

Legacy GPIO pin designator for P9.27

Namespace: [IO.Objects.libsimpleio.Platforms](#)

Assembly: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

▪ Syntax

C# VB F#

[Copy](#)

```
public static readonly Designator GPIO115
```

Field Value

Type: [Designator](#)

▪ Remarks

Requires the [BB-GPIO](#) device tree overlay.

▪ See Also

Reference

[BeagleBone Class](#)

[IO.Objects.libsimpleio.Platforms Namespace](#)

BeagleBoneGPIO117 Field

Legacy GPIO pin designator for P9.25

Namespace: [IO.Objects.libsimpleio.Platforms](#)

Assembly: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

▪ Syntax

C# VB F#

[Copy](#)

```
public static readonly Designator GPIO117
```

Field Value

Type: [Designator](#)

▪ Remarks

Requires the [BB-GPIO](#) device tree overlay.

▪ See Also

Reference

[BeagleBone Class](#)

[IO.Objects.libsimpleio.Platforms Namespace](#)

BeagleBoneGPIO12 Field

Legacy GPIO pin designator for P9.20

Namespace: [IO.Objects.libsimpleio.Platforms](#)

Assembly: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

▪ Syntax

C# VB F#

[Copy](#)

```
public static readonly Designator GPIO12
```

Field Value

Type: [Designator](#)

▪ Remarks

Requires the [BB-GPIO](#) device tree overlay.

▪ See Also

Reference

[BeagleBone Class](#)

[IO.Objects.libsimpleio.Platforms Namespace](#)

BeagleBoneGPIO13 Field

Legacy GPIO pin designator for P9.19

Namespace: [IO.Objects.libsimpleio.Platforms](#)

Assembly: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

▪ Syntax

C# VB F#

[Copy](#)

```
public static readonly Designator GPIO13
```

Field Value

Type: [Designator](#)

▪ Remarks

Requires the [BB-GPIO](#) device tree overlay.

▪ See Also

Reference

[BeagleBone Class](#)

[IO.Objects.libsimpleio.Platforms Namespace](#)

BeagleBoneGPIO14 Field

Legacy GPIO pin designator for P9.26

Namespace: [IO.Objects.libsimpleio.Platforms](#)

Assembly: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

▪ Syntax

C# VB F#

[Copy](#)

```
public static readonly Designator GPIO14
```

Field Value

Type: [Designator](#)

▪ Remarks

Requires the [BB-GPIO](#) device tree overlay.

▪ See Also

Reference

[BeagleBone Class](#)

[IO.Objects.libsimpleio.Platforms Namespace](#)

BeagleBoneGPIO15 Field

Legacy GPIO pin designator for P9.24

Namespace: [IO.Objects.libsimpleio.Platforms](#)

Assembly: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

▪ Syntax

C# VB F#

[Copy](#)

```
public static readonly Designator GPIO15
```

Field Value

Type: [Designator](#)

▪ Remarks

Requires the [BB-GPIO](#) device tree overlay.

▪ See Also

Reference

[BeagleBone Class](#)

[IO.Objects.libsimpleio.Platforms Namespace](#)

BeagleBoneGPIO2 Field

Legacy GPIO pin designator for P9.22

Namespace: [IO.Objects.libsimpleio.Platforms](#)

Assembly: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

▪ Syntax

C# VB F#

[Copy](#)

```
public static readonly Designator GPIO2
```

Field Value

Type: [Designator](#)

▪ Remarks

Requires the [BB-GPIO](#) device tree overlay.

▪ See Also

Reference

[BeagleBone Class](#)

[IO.Objects.libsimpleio.Platforms Namespace](#)

BeagleBoneGPIO20 Field

Legacy GPIO pin designator for P9.41

Namespace: [IO.Objects.libsimpleio.Platforms](#)

Assembly: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

▪ Syntax

C# VB F#

[Copy](#)

```
public static readonly Designator GPIO20
```

Field Value

Type: [Designator](#)

▪ Remarks

Requires the [BB-GPIO](#) device tree overlay.

▪ See Also

Reference

[BeagleBone Class](#)

[IO.Objects.libsimpleio.Platforms Namespace](#)

BeagleBoneGPIO22 Field

Legacy GPIO pin designator for P8.19

Namespace: [IO.Objects.libsimpleio.Platforms](#)

Assembly: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

▪ Syntax

C# VB F#

[Copy](#)

```
public static readonly Designator GPIO22
```

Field Value

Type: [Designator](#)

▪ Remarks

Requires the [BB-GPIO](#) device tree overlay.

▪ See Also

Reference

[BeagleBone Class](#)

[IO.Objects.libsimpleio.Platforms Namespace](#)

BeagleBoneGPIO23 Field

Legacy GPIO pin designator for P8.13

Namespace: [IO.Objects.libsimpleio.Platforms](#)

Assembly: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

▪ Syntax

C# VB F#

[Copy](#)

```
public static readonly Designator GPIO23
```

Field Value

Type: [Designator](#)

▪ Remarks

Requires the [BB-GPIO](#) device tree overlay.

▪ See Also

Reference

[BeagleBone Class](#)

[IO.Objects.libsimpleio.Platforms Namespace](#)

BeagleBoneGPIO26 Field

Legacy GPIO pin designator for P8.14

Namespace: [IO.Objects.libsimpleio.Platforms](#)

Assembly: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

▪ Syntax

C# VB F#

[Copy](#)

```
public static readonly Designator GPIO26
```

Field Value

Type: [Designator](#)

▪ Remarks

Requires the [BB-GPIO](#) device tree overlay.

▪ See Also

Reference

[BeagleBone Class](#)

[IO.Objects.libsimpleio.Platforms Namespace](#)

BeagleBoneGPIO27 Field

Legacy GPIO pin designator for P8.17

Namespace: [IO.Objects.libsimpleio.Platforms](#)

Assembly: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

▪ Syntax

C# VB F#

[Copy](#)

```
public static readonly Designator GPIO27
```

Field Value

Type: [Designator](#)

▪ Remarks

Requires the [BB-GPIO](#) device tree overlay.

▪ See Also

Reference

[BeagleBone Class](#)

[IO.Objects.libsimpleio.Platforms Namespace](#)

BeagleBoneGPIO3 Field

Legacy GPIO pin designator for P9.21

Namespace: [IO.Objects.libsimpleio.Platforms](#)

Assembly: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

▪ Syntax

C# VB F#

[Copy](#)

```
public static readonly Designator GPIO3
```

Field Value

Type: [Designator](#)

▪ Remarks

Requires the [BB-GPIO](#) device tree overlay.

▪ See Also

Reference

[BeagleBone Class](#)

[IO.Objects.libsimpleio.Platforms Namespace](#)

BeagleBoneGPIO30 Field

Legacy GPIO pin designator for P9.11

Namespace: [IO.Objects.libsimpleio.Platforms](#)

Assembly: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

▪ Syntax

C# VB F#

[Copy](#)

```
public static readonly Designator GPIO30
```

Field Value

Type: [Designator](#)

▪ Remarks

Requires the [BB-GPIO](#) device tree overlay.

▪ See Also

Reference

[BeagleBone Class](#)

[IO.Objects.libsimpleio.Platforms Namespace](#)

BeagleBoneGPIO31 Field

Legacy GPIO pin designator for P9.13

Namespace: [IO.Objects.libsimpleio.Platforms](#)

Assembly: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

▪ Syntax

C# VB F#

[Copy](#)

```
public static readonly Designator GPIO31
```

Field Value

Type: [Designator](#)

▪ Remarks

Requires the [BB-GPIO](#) device tree overlay.

▪ See Also

Reference

[BeagleBone Class](#)

[IO.Objects.libsimpleio.Platforms Namespace](#)

BeagleBoneGPIO32 Field

Legacy GPIO pin designator for P8.25

Namespace: [IO.Objects.libsimpleio.Platforms](#)

Assembly: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

▪ Syntax

C# VB F#

[Copy](#)

```
public static readonly Designator GPIO32
```

Field Value

Type: [Designator](#)

▪ Remarks

Requires the BeagleBone White and the [BB-NOEMMC](#) device tree overlay.

▪ See Also

Reference

[BeagleBone Class](#)

[IO.Objects.libsimpleio.Platforms Namespace](#)

BeagleBoneGPIO33 Field

Legacy GPIO pin designator for P8.24

Namespace: [IO.Objects.libsimpleio.Platforms](#)

Assembly: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

▪ Syntax

C# VB F#

[Copy](#)

```
public static readonly Designator GPIO33
```

Field Value

Type: [Designator](#)

▪ Remarks

Requires the BeagleBone White and the [BB-NOEMMC](#) device tree overlay.

▪ See Also

Reference

[BeagleBone Class](#)

[IO.Objects.libsimpleio.Platforms Namespace](#)

BeagleBoneGPIO34 Field

Legacy GPIO pin designator for P8.5

Namespace: [IO.Objects.libsimpleio.Platforms](#)

Assembly: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

▪ Syntax

C# VB F#

[Copy](#)

```
public static readonly Designator GPIO34
```

Field Value

Type: [Designator](#)

▪ Remarks

Requires the BeagleBone White and the [BB-NOEMMC](#) device tree overlay.

▪ See Also

Reference

[BeagleBone Class](#)

[IO.Objects.libsimpleio.Platforms Namespace](#)

BeagleBoneGPIO35 Field

Legacy GPIO pin designator for P8.6

Namespace: [IO.Objects.libsimpleio.Platforms](#)

Assembly: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

▪ Syntax

C# VB F#

[Copy](#)

```
public static readonly Designator GPIO35
```

Field Value

Type: [Designator](#)

▪ Remarks

Requires the BeagleBone White and the [BB-NOEMMC](#) device tree overlay.

▪ See Also

Reference

[BeagleBone Class](#)

[IO.Objects.libsimpleio.Platforms Namespace](#)

BeagleBoneGPIO36 Field

Legacy GPIO pin designator for P8.23

Namespace: [IO.Objects.libsimpleio.Platforms](#)

Assembly: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

▪ Syntax

C# VB F#

[Copy](#)

```
public static readonly Designator GPIO36
```

Field Value

Type: [Designator](#)

▪ Remarks

Requires the BeagleBone White and the [BB-NOEMMC](#) device tree overlay.

▪ See Also

Reference

[BeagleBone Class](#)

[IO.Objects.libsimpleio.Platforms Namespace](#)

BeagleBoneGPIO37 Field

Legacy GPIO pin designator for P8.22

Namespace: [IO.Objects.libsimpleio.Platforms](#)

Assembly: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

▪ Syntax

C# VB F#

[Copy](#)

```
public static readonly Designator GPIO37
```

Field Value

Type: [Designator](#)

▪ Remarks

Requires the BeagleBone White and the [BB-NOEMMC](#) device tree overlay.

▪ See Also

Reference

[BeagleBone Class](#)

[IO.Objects.libsimpleio.Platforms Namespace](#)

BeagleBoneGPIO38 Field

Legacy GPIO pin designator for P8.3

Namespace: [IO.Objects.libsimpleio.Platforms](#)

Assembly: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

▪ Syntax

C# VB F#

[Copy](#)

```
public static readonly Designator GPIO38
```

Field Value

Type: [Designator](#)

▪ Remarks

Requires the BeagleBone White and the [BB-NOEMMC](#) device tree overlay.

▪ See Also

Reference

[BeagleBone Class](#)

[IO.Objects.libsimpleio.Platforms Namespace](#)

BeagleBoneGPIO39 Field

Legacy GPIO pin designator for P8.4

Namespace: [IO.Objects.libsimpleio.Platforms](#)

Assembly: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

▪ Syntax

C# VB F#

[Copy](#)

```
public static readonly Designator GPIO39
```

Field Value

Type: [Designator](#)

▪ Remarks

Requires the BeagleBone White and the [BB-NOEMMC](#) device tree overlay.

▪ See Also

Reference

[BeagleBone Class](#)

[IO.Objects.libsimpleio.Platforms Namespace](#)

BeagleBoneGPIO4 Field

Legacy GPIO pin designator for P9.18

Namespace: [IO.Objects.libsimpleio.Platforms](#)

Assembly: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

▪ Syntax

C# VB F#

[Copy](#)

```
public static readonly Designator GPIO4
```

Field Value

Type: [Designator](#)

▪ Remarks

Requires the [BB-GPIO](#) device tree overlay.

▪ See Also

Reference

[BeagleBone Class](#)

[IO.Objects.libsimpleio.Platforms Namespace](#)

BeagleBoneGPIO44 Field

Legacy GPIO pin designator for P8.12

Namespace: [IO.Objects.libsimpleio.Platforms](#)

Assembly: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

▪ Syntax

C# VB F#

[Copy](#)

```
public static readonly Designator GPIO44
```

Field Value

Type: [Designator](#)

▪ Remarks

Requires the [BB-GPIO](#) device tree overlay.

▪ See Also

Reference

[BeagleBone Class](#)

[IO.Objects.libsimpleio.Platforms Namespace](#)

BeagleBoneGPIO45 Field

Legacy GPIO pin designator for P8.11

Namespace: [IO.Objects.libsimpleio.Platforms](#)

Assembly: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

▪ Syntax

C# VB F#

[Copy](#)

```
public static readonly Designator GPIO45
```

Field Value

Type: [Designator](#)

▪ Remarks

Requires the [BB-GPIO](#) device tree overlay.

▪ See Also

Reference

[BeagleBone Class](#)

[IO.Objects.libsimpleio.Platforms Namespace](#)

BeagleBoneGPIO46 Field

Legacy GPIO pin designator for P8.16

Namespace: [IO.Objects.libsimpleio.Platforms](#)

Assembly: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

▪ Syntax

C# VB F#

[Copy](#)

```
public static readonly Designator GPIO46
```

Field Value

Type: [Designator](#)

▪ Remarks

Requires the [BB-GPIO](#) device tree overlay.

▪ See Also

Reference

[BeagleBone Class](#)

[IO.Objects.libsimpleio.Platforms Namespace](#)

BeagleBoneGPIO47 Field

Legacy GPIO pin designator for P8.15

Namespace: [IO.Objects.libsimpleio.Platforms](#)

Assembly: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

▪ Syntax

C# VB F#

[Copy](#)

```
public static readonly Designator GPIO47
```

Field Value

Type: [Designator](#)

▪ Remarks

Requires the [BB-GPIO](#) device tree overlay.

▪ See Also

Reference

[BeagleBone Class](#)

[IO.Objects.libsimpleio.Platforms Namespace](#)

BeagleBoneGPIO48 Field

Legacy GPIO pin designator for P9.15

Namespace: [IO.Objects.libsimpleio.Platforms](#)

Assembly: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

▪ Syntax

C# VB F#

[Copy](#)

```
public static readonly Designator GPIO48
```

Field Value

Type: [Designator](#)

▪ Remarks

Requires the [BB-GPIO](#) device tree overlay.

▪ See Also

Reference

[BeagleBone Class](#)

[IO.Objects.libsimpleio.Platforms Namespace](#)

BeagleBoneGPIO49 Field

Legacy GPIO pin designator for P9.23

Namespace: [IO.Objects.libsimpleio.Platforms](#)

Assembly: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

▪ Syntax

C# VB F#

[Copy](#)

```
public static readonly Designator GPIO49
```

Field Value

Type: [Designator](#)

▪ Remarks

Requires the [BB-GPIO](#) device tree overlay.

▪ See Also

Reference

[BeagleBone Class](#)

[IO.Objects.libsimpleio.Platforms Namespace](#)

BeagleBoneGPIO5 Field

Legacy GPIO pin designator for P9.17

Namespace: [IO.Objects.libsimpleio.Platforms](#)

Assembly: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

▪ Syntax

C# VB F#

[Copy](#)

```
public static readonly Designator GPIO5
```

Field Value

Type: [Designator](#)

▪ Remarks

Requires the [BB-GPIO](#) device tree overlay.

▪ See Also

Reference

[BeagleBone Class](#)

[IO.Objects.libsimpleio.Platforms Namespace](#)

BeagleBoneGPIO50 Field

Legacy GPIO pin designator for P9.14

Namespace: [IO.Objects.libsimpleio.Platforms](#)

Assembly: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

▪ Syntax

C# VB F#

[Copy](#)

```
public static readonly Designator GPIO50
```

Field Value

Type: [Designator](#)

▪ Remarks

Requires the [BB-GPIO](#) device tree overlay.

▪ See Also

Reference

[BeagleBone Class](#)

[IO.Objects.libsimpleio.Platforms Namespace](#)

BeagleBoneGPIO51 Field

Legacy GPIO pin designator for P9.16

Namespace: [IO.Objects.libsimpleio.Platforms](#)

Assembly: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

▪ Syntax

C# VB F#

[Copy](#)

```
public static readonly Designator GPIO51
```

Field Value

Type: [Designator](#)

▪ Remarks

Requires the [BB-GPIO](#) device tree overlay.

▪ See Also

Reference

[BeagleBone Class](#)

[IO.Objects.libsimpleio.Platforms Namespace](#)

BeagleBoneGPIO60 Field

Legacy GPIO pin designator for P9.12

Namespace: [IO.Objects.libsimpleio.Platforms](#)

Assembly: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

▪ Syntax

C# VB F#

[Copy](#)

```
public static readonly Designator GPIO60
```

Field Value

Type: [Designator](#)

▪ Remarks

Requires the [BB-GPIO](#) device tree overlay.

▪ See Also

Reference

[BeagleBone Class](#)

[IO.Objects.libsimpleio.Platforms Namespace](#)

BeagleBoneGPIO61 Field

Legacy GPIO pin designator for P8.26

Namespace: [IO.Objects.libsimpleio.Platforms](#)

Assembly: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

▪ Syntax

C# VB F#

[Copy](#)

```
public static readonly Designator GPIO61
```

Field Value

Type: [Designator](#)

▪ Remarks

Requires the [BB-GPIO](#) device tree overlay.

▪ See Also

Reference

[BeagleBone Class](#)

[IO.Objects.libsimpleio.Platforms Namespace](#)

BeagleBoneGPIO62 Field

Legacy GPIO pin designator for P8.21

Namespace: [IO.Objects.libsimpleio.Platforms](#)

Assembly: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

▪ Syntax

C# VB F#

[Copy](#)

```
public static readonly Designator GPIO62
```

Field Value

Type: [Designator](#)

▪ Remarks

Requires the BeagleBone White and the [BB-NOEMMC](#) device tree overlay.

▪ See Also

Reference

[BeagleBone Class](#)

[IO.Objects.libsimpleio.Platforms Namespace](#)

BeagleBoneGPIO63 Field

Legacy GPIO pin designator for P8.20

Namespace: [IO.Objects.libsimpleio.Platforms](#)

Assembly: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

▪ Syntax

C# VB F#

[Copy](#)

```
public static readonly Designator GPIO63
```

Field Value

Type: [Designator](#)

▪ Remarks

Requires the BeagleBone White and the [BB-NOEMMC](#) device tree overlay.

▪ See Also

Reference

[BeagleBone Class](#)

[IO.Objects.libsimpleio.Platforms Namespace](#)

BeagleBoneGPIO65 Field

Legacy GPIO pin designator for P8.18

Namespace: [IO.Objects.libsimpleio.Platforms](#)

Assembly: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

▪ Syntax

C# VB F#

[Copy](#)

```
public static readonly Designator GPIO65
```

Field Value

Type: [Designator](#)

▪ Remarks

Requires the [BB-GPIO](#) device tree overlay.

▪ See Also

Reference

[BeagleBone Class](#)

[IO.Objects.libsimpleio.Platforms Namespace](#)

BeagleBoneGPIO66 Field

Legacy GPIO pin designator for P8.7

Namespace: [IO.Objects.libsimpleio.Platforms](#)

Assembly: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

▪ Syntax

C# VB F#

[Copy](#)

```
public static readonly Designator GPIO66
```

Field Value

Type: [Designator](#)

▪ Remarks

Requires the [BB-GPIO](#) device tree overlay.

▪ See Also

Reference

[BeagleBone Class](#)

[IO.Objects.libsimpleio.Platforms Namespace](#)

BeagleBoneGPIO67 Field

Legacy GPIO pin designator for P8.8

Namespace: [IO.Objects.libsimpleio.Platforms](#)

Assembly: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

▪ Syntax

C# VB F#

[Copy](#)

```
public static readonly Designator GPIO67
```

Field Value

Type: [Designator](#)

▪ Remarks

Requires the [BB-GPIO](#) device tree overlay.

▪ See Also

Reference

[BeagleBone Class](#)

[IO.Objects.libsimpleio.Platforms Namespace](#)

BeagleBoneGPIO68 Field

Legacy GPIO pin designator for P8.10

Namespace: [IO.Objects.libsimpleio.Platforms](#)

Assembly: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

▪ Syntax

C# VB F#

[Copy](#)

```
public static readonly Designator GPIO68
```

Field Value

Type: [Designator](#)

▪ Remarks

Requires the [BB-GPIO](#) device tree overlay.

▪ See Also

Reference

[BeagleBone Class](#)

[IO.Objects.libsimpleio.Platforms Namespace](#)

BeagleBoneGPIO69 Field

Legacy GPIO pin designator for P8.9

Namespace: [IO.Objects.libsimpleio.Platforms](#)

Assembly: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

▪ Syntax

C# VB F#

[Copy](#)

```
public static readonly Designator GPIO69
```

Field Value

Type: [Designator](#)

▪ Remarks

Requires the [BB-GPIO](#) device tree overlay.

▪ See Also

Reference

[BeagleBone Class](#)

[IO.Objects.libsimpleio.Platforms Namespace](#)

BeagleBoneGPIO7 Field

Legacy GPIO pin designator for P9.42

Namespace: [IO.Objects.libsimpleio.Platforms](#)

Assembly: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

▪ Syntax

C# VB F#

[Copy](#)

```
public static readonly Designator GPIO7
```

Field Value

Type: [Designator](#)

▪ Remarks

Requires the [BB-GPIO](#) device tree overlay.

▪ See Also

Reference

[BeagleBone Class](#)

[IO.Objects.libsimpleio.Platforms Namespace](#)

BeagleBoneGPIO70 Field

Legacy GPIO pin designator for P8.45

Namespace: [IO.Objects.libsimpleio.Platforms](#)

Assembly: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

▪ Syntax

C# VB F#

[Copy](#)

```
public static readonly Designator GPIO70
```

Field Value

Type: [Designator](#)

▪ Remarks

Requires the [BB-GPIO](#) device tree overlay.

▪ See Also

Reference

[BeagleBone Class](#)

[IO.Objects.libsimpleio.Platforms Namespace](#)

BeagleBoneGPIO71 Field

Legacy GPIO pin designator for P8.46

Namespace: [IO.Objects.libsimpleio.Platforms](#)

Assembly: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

▪ Syntax

C# VB F#

[Copy](#)

```
public static readonly Designator GPIO71
```

Field Value

Type: [Designator](#)

▪ Remarks

Requires the [BB-GPIO](#) device tree overlay.

▪ See Also

Reference

[BeagleBone Class](#)

[IO.Objects.libsimpleio.Platforms Namespace](#)

BeagleBoneGPIO72 Field

Legacy GPIO pin designator for P8.43

Namespace: [IO.Objects.libsimpleio.Platforms](#)

Assembly: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

▪ Syntax

C# VB F#

[Copy](#)

```
public static readonly Designator GPIO72
```

Field Value

Type: [Designator](#)

▪ Remarks

Requires the [BB-GPIO](#) device tree overlay.

▪ See Also

Reference

[BeagleBone Class](#)

[IO.Objects.libsimpleio.Platforms Namespace](#)

BeagleBoneGPIO73 Field

Legacy GPIO pin designator for P8.44

Namespace: [IO.Objects.libsimpleio.Platforms](#)

Assembly: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

▪ Syntax

C# VB F#

[Copy](#)

```
public static readonly Designator GPIO73
```

Field Value

Type: [Designator](#)

▪ Remarks

Requires the [BB-GPIO](#) device tree overlay.

▪ See Also

Reference

[BeagleBone Class](#)

[IO.Objects.libsimpleio.Platforms Namespace](#)

BeagleBoneGPIO74 Field

Legacy GPIO pin designator for P8.41

Namespace: [IO.Objects.libsimpleio.Platforms](#)

Assembly: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

▪ Syntax

C# VB F#

[Copy](#)

```
public static readonly Designator GPIO74
```

Field Value

Type: [Designator](#)

▪ Remarks

Requires the [BB-GPIO](#) device tree overlay.

▪ See Also

Reference

[BeagleBone Class](#)

[IO.Objects.libsimpleio.Platforms Namespace](#)

BeagleBoneGPIO75 Field

Legacy GPIO pin designator for P8.42

Namespace: [IO.Objects.libsimpleio.Platforms](#)

Assembly: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

▪ Syntax

C# VB F#

[Copy](#)

```
public static readonly Designator GPIO75
```

Field Value

Type: [Designator](#)

▪ Remarks

Requires the [BB-GPIO](#) device tree overlay.

▪ See Also

Reference

[BeagleBone Class](#)

[IO.Objects.libsimpleio.Platforms Namespace](#)

BeagleBoneGPIO76 Field

Legacy GPIO pin designator for P8.39

Namespace: [IO.Objects.libsimpleio.Platforms](#)

Assembly: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

▪ Syntax

C# VB F#

[Copy](#)

```
public static readonly Designator GPIO76
```

Field Value

Type: [Designator](#)

▪ Remarks

Requires the [BB-GPIO](#) device tree overlay.

▪ See Also

Reference

[BeagleBone Class](#)

[IO.Objects.libsimpleio.Platforms Namespace](#)

BeagleBoneGPIO77 Field

Legacy GPIO pin designator for P8.40

Namespace: [IO.Objects.libsimpleio.Platforms](#)

Assembly: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

▪ Syntax

C# VB F#

[Copy](#)

```
public static readonly Designator GPIO77
```

Field Value

Type: [Designator](#)

▪ Remarks

Requires the [BB-GPIO](#) device tree overlay.

▪ See Also

Reference

[BeagleBone Class](#)

[IO.Objects.libsimpleio.Platforms Namespace](#)

BeagleBoneGPIO78 Field

Legacy GPIO pin designator for P8.37

Namespace: [IO.Objects.libsimpleio.Platforms](#)

Assembly: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

▪ Syntax

C# VB F#

[Copy](#)

```
public static readonly Designator GPIO78
```

Field Value

Type: [Designator](#)

▪ Remarks

Requires the [BB-GPIO](#) device tree overlay.

▪ See Also

Reference

[BeagleBone Class](#)

[IO.Objects.libsimpleio.Platforms Namespace](#)

BeagleBoneGPIO79 Field

Legacy GPIO pin designator for P8.38

Namespace: [IO.Objects.libsimpleio.Platforms](#)

Assembly: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

▪ Syntax

C# VB F#

[Copy](#)

```
public static readonly Designator GPIO79
```

Field Value

Type: [Designator](#)

▪ Remarks

Requires the [BB-GPIO](#) device tree overlay.

▪ See Also

Reference

[BeagleBone Class](#)

[IO.Objects.libsimpleio.Platforms Namespace](#)

BeagleBoneGPIO8 Field

Legacy GPIO pin designator for P8.35

Namespace: [IO.Objects.libsimpleio.Platforms](#)

Assembly: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

▪ Syntax

C# VB F#

[Copy](#)

```
public static readonly Designator GPIO8
```

Field Value

Type: [Designator](#)

▪ Remarks

Requires the [BB-GPIO](#) device tree overlay.

▪ See Also

Reference

[BeagleBone Class](#)

[IO.Objects.libsimpleio.Platforms Namespace](#)

BeagleBoneGPIO80 Field

Legacy GPIO pin designator for P8.36

Namespace: [IO.Objects.libsimpleio.Platforms](#)

Assembly: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

▪ Syntax

C# VB F#

[Copy](#)

```
public static readonly Designator GPIO80
```

Field Value

Type: [Designator](#)

▪ Remarks

Requires the [BB-GPIO](#) device tree overlay.

▪ See Also

Reference

[BeagleBone Class](#)

[IO.Objects.libsimpleio.Platforms Namespace](#)

BeagleBoneGPIO81 Field

Legacy GPIO pin designator for P8.34

Namespace: [IO.Objects.libsimpleio.Platforms](#)

Assembly: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

▪ Syntax

C# VB F#

[Copy](#)

```
public static readonly Designator GPIO81
```

Field Value

Type: [Designator](#)

▪ Remarks

Requires the [BB-GPIO](#) device tree overlay.

▪ See Also

Reference

[BeagleBone Class](#)

[IO.Objects.libsimpleio.Platforms Namespace](#)

BeagleBoneGPIO86 Field

Legacy GPIO pin designator for P8.27

Namespace: [IO.Objects.libsimpleio.Platforms](#)

Assembly: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

▪ Syntax

C# VB F#

[Copy](#)

```
public static readonly Designator GPIO86
```

Field Value

Type: [Designator](#)

▪ Remarks

Requires the [BB-GPIO](#) device tree overlay.

▪ See Also

Reference

[BeagleBone Class](#)

[IO.Objects.libsimpleio.Platforms Namespace](#)

BeagleBoneGPIO87 Field

Legacy GPIO pin designator for P8.29

Namespace: [IO.Objects.libsimpleio.Platforms](#)

Assembly: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

▪ Syntax

C# VB F#

[Copy](#)

```
public static readonly Designator GPIO87
```

Field Value

Type: [Designator](#)

▪ Remarks

Requires the [BB-GPIO](#) device tree overlay.

▪ See Also

Reference

[BeagleBone Class](#)

[IO.Objects.libsimpleio.Platforms Namespace](#)

BeagleBoneGPIO88 Field

Legacy GPIO pin designator for P8.28

Namespace: [IO.Objects.libsimpleio.Platforms](#)

Assembly: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

▪ Syntax

C# VB F#

[Copy](#)

```
public static readonly Designator GPIO88
```

Field Value

Type: [Designator](#)

▪ Remarks

Requires the [BB-GPIO](#) device tree overlay.

▪ See Also

Reference

[BeagleBone Class](#)

[IO.Objects.libsimpleio.Platforms Namespace](#)

BeagleBoneGPIO89 Field

Legacy GPIO pin designator for P8.30

Namespace: [IO.Objects.libsimpleio.Platforms](#)

Assembly: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

▪ Syntax

C# VB F#

[Copy](#)

```
public static readonly Designator GPIO89
```

Field Value

Type: [Designator](#)

▪ Remarks

Requires the [BB-GPIO](#) device tree overlay.

▪ See Also

Reference

[BeagleBone Class](#)

[IO.Objects.libsimpleio.Platforms Namespace](#)

BeagleBoneGPIO9 Field

Legacy GPIO pin designator for P8.33

Namespace: [IO.Objects.libsimpleio.Platforms](#)

Assembly: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

▪ Syntax

C# VB F#

[Copy](#)

```
public static readonly Designator GPIO9
```

Field Value

Type: [Designator](#)

▪ Remarks

Requires the [BB-GPIO](#) device tree overlay.

▪ See Also

Reference

[BeagleBone Class](#)

[IO.Objects.libsimpleio.Platforms Namespace](#)

BeagleBoneI2C2 Field

I2C bus designator for P9.19 and P9.20

Namespace: [IO.Objects.libsimpleioPlatforms](#)

Assembly: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

↳ Syntax

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

[Copy](#)

```
public static readonly Designator I2C2
```

Field Value

Type: [Designator](#)

↳ See Also

Reference

[BeagleBone Class](#)

[IO.Objects.libsimpleioPlatforms Namespace](#)

BeagleBoneSPI2_0 Field

SPI slave select designator for P9.28

Namespace: [IO.Objects.libsimpleioPlatforms](#)

Assembly: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

↳ Syntax

C# VB F#

[Copy](#)

```
public static readonly Designator SPI2_0
```

Field Value

Type: [Designator](#)

↳ See Also

Reference

[BeagleBone Class](#)

[IO.Objects.libsimpleioPlatforms Namespace](#)

BeagleBoneSPI2_1 Field

SPI slave select designator for P9.42

Namespace: [IO.Objects.libsimpleioPlatforms](#)

Assembly: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

↳ Syntax

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

[Copy](#)

```
public static readonly Designator SPI2_1
```

Field Value

Type: [Designator](#)

↳ See Also

Reference

[BeagleBone Class](#)

[IO.Objects.libsimpleioPlatforms Namespace](#)

PocketBeagle Class

This class defines identifiers for the devices provided by the PocketBeagle hardware platform.

▪ Inheritance Hierarchy

[SystemObject](#) [IO.Objects.libsimpleio.Platforms](#)[PocketBeagle](#)

Namespace: [IO.Objects.libsimpleio.Platforms](#)

Assembly: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

▪ Syntax

C# VB F#

[Copy](#)

```
public static class PocketBeagle
```

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

▪ Fields

| | Name | Description |
|----------------------------|------|---------------------------------------|
| • S | AIN0 | ADC input designator for P1.19 (1.8V) |
| • S | AIN1 | ADC input designator for P1.21 (1.8V) |
| • S | AIN2 | ADC input designator for P1.23 (1.8V) |
| • S | AIN3 | ADC input designator for P1.25 (1.8V) |
| • S | AIN4 | ADC input designator for P1.27 (1.8V) |

| | | |
|-----|---------|---------------------------------------|
| ♦ S | AIN5 | ADC input designator for P2.35 (3.6V) |
| ♦ S | AIN6 | ADC input designator for P1.2 (3.6V) |
| ♦ S | AIN7 | ADC input designator for P2.36 (1.8V) |
| ♦ S | GPIO110 | Legacy GPIO pin designator for P1.36 |
| ♦ S | GPIO111 | Legacy GPIO pin designator for P1.33 |
| ♦ S | GPIO112 | Legacy GPIO pin designator for P2.32 |
| ♦ S | GPIO113 | Legacy GPIO pin designator for P2.30 |
| ♦ S | GPIO114 | Legacy GPIO pin designator for P1.31 |
| ♦ S | GPIO115 | Legacy GPIO pin designator for P2.34 |
| ♦ S | GPIO116 | Legacy GPIO pin designator for P2.28 |
| ♦ S | GPIO117 | Legacy GPIO pin designator for P1.29 |
| ♦ S | GPIO12 | Legacy GPIO pin designator for P1.26 |
| ♦ S | GPIO13 | Legacy GPIO pin designator for P1.28 |
| ♦ S | GPIO14 | Legacy GPIO pin designator for P2.11 |
| ♦ S | GPIO15 | Legacy GPIO pin designator for P2.9 |
| ♦ S | GPIO19 | Legacy GPIO pin designator for P2.31 |
| ♦ S | GPIO2 | Legacy GPIO pin designator for P1.8 |
| ♦ S | GPIO20 | Legacy GPIO pin designator for P1.20 |
| ♦ S | GPIO23 | Legacy GPIO pin designator for P2.3 |

| | | |
|---|--------|--------------------------------------|
| ♦ | GPIO26 | Legacy GPIO pin designator for P1.34 |
| ♦ | GPIO27 | Legacy GPIO pin designator for P2.19 |
| ♦ | GPIO3 | Legacy GPIO pin designator for P1.10 |
| ♦ | GPIO30 | Legacy GPIO pin designator for P2.5 |
| ♦ | GPIO31 | Legacy GPIO pin designator for P2.7 |
| ♦ | GPIO4 | Legacy GPIO pin designator for P1.12 |
| ♦ | GPIO40 | Legacy GPIO pin designator for P2.27 |
| ♦ | GPIO41 | Legacy GPIO pin designator for P2.25 |
| ♦ | GPIO42 | Legacy GPIO pin designator for P1.32 |
| ♦ | GPIO43 | Legacy GPIO pin designator for P1.30 |
| ♦ | GPIO44 | Legacy GPIO pin designator for P2.24 |
| ♦ | GPIO45 | Legacy GPIO pin designator for P2.33 |
| ♦ | GPIO46 | Legacy GPIO pin designator for P2.22 |
| ♦ | GPIO47 | Legacy GPIO pin designator for P2.18 |
| ♦ | GPIO5 | Legacy GPIO pin designator for P1.6 |
| ♦ | GPIO50 | Legacy GPIO pin designator for P2.1 |
| ♦ | GPIO52 | Legacy GPIO pin designator for P2.10 |
| ♦ | GPIO57 | Legacy GPIO pin designator for P2.6 |
| ♦ | GPIO58 | Legacy GPIO pin designator for P2.4 |

| | | |
|------------|------------------------|--|
| • S | GPIO59 | Legacy GPIO pin designator for P2.2 |
| • S | GPIO60 | Legacy GPIO pin designator for P2.8 |
| • S | GPIO64 | Legacy GPIO pin designator for P2.20 |
| • S | GPIO65 | Legacy GPIO pin designator for P2.17 |
| • S | GPIO7 | Legacy GPIO pin designator for P2.29 |
| • S | GPIO86 | Legacy GPIO pin designator for P2.35 |
| • S | GPIO87 | Legacy GPIO pin designator for P1.2 |
| • S | GPIO88 | Legacy GPIO pin designator for P1.35 |
| • S | GPIO89 | Legacy GPIO pin designator for P1.4 |
| • S | I2C1 | I2C bus designator for P1.26 and P1.28 |
| • S | I2C2 | I2C bus designator for P2.9 and P2.11 |
| • S | PWM0_0 | PWM output designator for P1.36 |
| • S | PWM2_0 | PWM output designator for P2.1 |
| • S | SPI0_0 | SPI slave select designator for P1.6 |
| • S | SPI1_1 | SPI slave select designator for P2.31 |

[Top](#)

See Also

Reference

[IO.Objects.libsimpleio.Platforms Namespace](#)

PocketBeagle Fields

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

Fields

| | Name | Description |
|------------|---------|---------------------------------------|
| • s | AIN0 | ADC input designator for P1.19 (1.8V) |
| • s | AIN1 | ADC input designator for P1.21 (1.8V) |
| • s | AIN2 | ADC input designator for P1.23 (1.8V) |
| • s | AIN3 | ADC input designator for P1.25 (1.8V) |
| • s | AIN4 | ADC input designator for P1.27 (1.8V) |
| • s | AIN5 | ADC input designator for P2.35 (3.6V) |
| • s | AIN6 | ADC input designator for P1.2 (3.6V) |
| • s | AIN7 | ADC input designator for P2.36 (1.8V) |
| • s | GPIO110 | Legacy GPIO pin designator for P1.36 |
| • s | GPIO111 | Legacy GPIO pin designator for P1.33 |
| • s | GPIO112 | Legacy GPIO pin designator for P2.32 |
| • s | GPIO113 | Legacy GPIO pin designator for P2.30 |
| • s | GPIO114 | Legacy GPIO pin designator for P1.31 |

| | | |
|-----|---------|--------------------------------------|
| ♦ S | GPIO115 | Legacy GPIO pin designator for P2.34 |
| ♦ S | GPIO116 | Legacy GPIO pin designator for P2.28 |
| ♦ S | GPIO117 | Legacy GPIO pin designator for P1.29 |
| ♦ S | GPIO12 | Legacy GPIO pin designator for P1.26 |
| ♦ S | GPIO13 | Legacy GPIO pin designator for P1.28 |
| ♦ S | GPIO14 | Legacy GPIO pin designator for P2.11 |
| ♦ S | GPIO15 | Legacy GPIO pin designator for P2.9 |
| ♦ S | GPIO19 | Legacy GPIO pin designator for P2.31 |
| ♦ S | GPIO2 | Legacy GPIO pin designator for P1.8 |
| ♦ S | GPIO20 | Legacy GPIO pin designator for P1.20 |
| ♦ S | GPIO23 | Legacy GPIO pin designator for P2.3 |
| ♦ S | GPIO26 | Legacy GPIO pin designator for P1.34 |
| ♦ S | GPIO27 | Legacy GPIO pin designator for P2.19 |
| ♦ S | GPIO3 | Legacy GPIO pin designator for P1.10 |
| ♦ S | GPIO30 | Legacy GPIO pin designator for P2.5 |
| ♦ S | GPIO31 | Legacy GPIO pin designator for P2.7 |
| ♦ S | GPIO4 | Legacy GPIO pin designator for P1.12 |
| ♦ S | GPIO40 | Legacy GPIO pin designator for P2.27 |
| ♦ S | GPIO41 | Legacy GPIO pin designator for P2.25 |

| | | |
|-----|--------|--------------------------------------|
| ♦ S | GPIO42 | Legacy GPIO pin designator for P1.32 |
| ♦ S | GPIO43 | Legacy GPIO pin designator for P1.30 |
| ♦ S | GPIO44 | Legacy GPIO pin designator for P2.24 |
| ♦ S | GPIO45 | Legacy GPIO pin designator for P2.33 |
| ♦ S | GPIO46 | Legacy GPIO pin designator for P2.22 |
| ♦ S | GPIO47 | Legacy GPIO pin designator for P2.18 |
| ♦ S | GPIO5 | Legacy GPIO pin designator for P1.6 |
| ♦ S | GPIO50 | Legacy GPIO pin designator for P2.1 |
| ♦ S | GPIO52 | Legacy GPIO pin designator for P2.10 |
| ♦ S | GPIO57 | Legacy GPIO pin designator for P2.6 |
| ♦ S | GPIO58 | Legacy GPIO pin designator for P2.4 |
| ♦ S | GPIO59 | Legacy GPIO pin designator for P2.2 |
| ♦ S | GPIO60 | Legacy GPIO pin designator for P2.8 |
| ♦ S | GPIO64 | Legacy GPIO pin designator for P2.20 |
| ♦ S | GPIO65 | Legacy GPIO pin designator for P2.17 |
| ♦ S | GPIO7 | Legacy GPIO pin designator for P2.29 |
| ♦ S | GPIO86 | Legacy GPIO pin designator for P2.35 |
| ♦ S | GPIO87 | Legacy GPIO pin designator for P1.2 |
| ♦ S | GPIO88 | Legacy GPIO pin designator for P1.35 |

| | | |
|------------|------------------------|--|
| • S | GPIO89 | Legacy GPIO pin designator for P1.4 |
| • S | I2C1 | I2C bus designator for P1.26 and P1.28 |
| • S | I2C2 | I2C bus designator for P2.9 and P2.11 |
| • S | PWM0_0 | PWM output designator for P1.36 |
| • S | PWM2_0 | PWM output designator for P2.1 |
| • S | SPI0_0 | SPI slave select designator for P1.6 |
| • S | SPI1_1 | SPI slave select designator for P2.31 |

[Top](#)

See Also

Reference

[PocketBeagle Class](#)

[IO.Objects.libsimpleio.Platforms Namespace](#)

PocketBeagleAIN0 Field

ADC input designator for P1.19 (1.8V)

Namespace: [IO.Objects.libsimpleio.Platforms](#)

Assembly: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

▪ Syntax

C# VB F#

[Copy](#)

```
public static readonly Designator AIN0
```

Field Value

Type: [Designator](#)

▪ Remarks

Requires the [PB-ADC](#) device tree overlay.

▪ See Also

Reference

[PocketBeagle Class](#)

[IO.Objects.libsimpleio.Platforms Namespace](#)

PocketBeagleAIN1 Field

ADC input designator for P1.21 (1.8V)

Namespace: [IO.Objects.libsimpleio.Platforms](#)

Assembly: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

▪ Syntax

C# VB F#

[Copy](#)

```
public static readonly Designator AIN1
```

Field Value

Type: [Designator](#)

▪ Remarks

Requires the [PB-ADC](#) device tree overlay.

▪ See Also

Reference

[PocketBeagle Class](#)

[IO.Objects.libsimpleio.Platforms Namespace](#)

PocketBeagleAIN2 Field

ADC input designator for P1.23 (1.8V)

Namespace: [IO.Objects.libsimpleio.Platforms](#)

Assembly: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

▪ Syntax

C# VB F#

[Copy](#)

```
public static readonly Designator AIN2
```

Field Value

Type: [Designator](#)

▪ Remarks

Requires the [PB-ADC](#) device tree overlay.

▪ See Also

Reference

[PocketBeagle Class](#)

[IO.Objects.libsimpleio.Platforms Namespace](#)

PocketBeagleAIN3 Field

ADC input designator for P1.25 (1.8V)

Namespace: [IO.Objects.libsimpleio.Platforms](#)

Assembly: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

▪ Syntax

C# VB F#

[Copy](#)

```
public static readonly Designator AIN3
```

Field Value

Type: [Designator](#)

▪ Remarks

Requires the [PB-ADC](#) device tree overlay.

▪ See Also

Reference

[PocketBeagle Class](#)

[IO.Objects.libsimpleio.Platforms Namespace](#)

PocketBeagleAIN4 Field

ADC input designator for P1.27 (1.8V)

Namespace: [IO.Objects.libsimpleio.Platforms](#)

Assembly: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

▪ Syntax

C# VB F#

[Copy](#)

```
public static readonly Designator AIN4
```

Field Value

Type: [Designator](#)

▪ Remarks

Requires the [PB-ADC](#) device tree overlay.

▪ See Also

Reference

[PocketBeagle Class](#)

[IO.Objects.libsimpleio.Platforms Namespace](#)

PocketBeagleAIN5 Field

ADC input designator for P2.35 (3.6V)

Namespace: [IO.Objects.libsimpleio.Platforms](#)

Assembly: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

▪ Syntax

C# VB F#

[Copy](#)

```
public static readonly Designator AIN5
```

Field Value

Type: [Designator](#)

▪ Remarks

Requires the [PB-ADC](#) device tree overlay.

▪ See Also

Reference

[PocketBeagle Class](#)

[IO.Objects.libsimpleio.Platforms Namespace](#)

PocketBeagleAIN6 Field

ADC input designator for P1.2 (3.6V)

Namespace: [IO.Objects.libsimpleio.Platforms](#)

Assembly: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

▪ Syntax

C# VB F#

[Copy](#)

```
public static readonly Designator AIN6
```

Field Value

Type: [Designator](#)

▪ Remarks

Requires the [PB-ADC](#) device tree overlay.

▪ See Also

Reference

[PocketBeagle Class](#)

[IO.Objects.libsimpleio.Platforms Namespace](#)

PocketBeagleAIN7 Field

ADC input designator for P2.36 (1.8V)

Namespace: [IO.Objects.libsimpleio.Platforms](#)

Assembly: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

▪ Syntax

C# VB F#

[Copy](#)

```
public static readonly Designator AIN7
```

Field Value

Type: [Designator](#)

▪ Remarks

Requires the [PB-ADC](#) device tree overlay.

▪ See Also

Reference

[PocketBeagle Class](#)

[IO.Objects.libsimpleio.Platforms Namespace](#)

PocketBeagleGPIO110 Field

Legacy GPIO pin designator for P1.36

Namespace: [IO.Objects.libsimpleio.Platforms](#)

Assembly: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

↳ Syntax

C# VB F#

[Copy](#)

```
public static readonly Designator GPIO110
```

Field Value

Type: [Designator](#)

↳ See Also

Reference

[PocketBeagle Class](#)

[IO.Objects.libsimpleio.Platforms Namespace](#)

PocketBeagleGPIO111 Field

Legacy GPIO pin designator for P1.33

Namespace: [IO.Objects.libsimpleio.Platforms](#)

Assembly: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

↳ Syntax

C# VB F#

[Copy](#)

```
public static readonly Designator GPIO111
```

Field Value

Type: [Designator](#)

↳ See Also

Reference

[PocketBeagle Class](#)

[IO.Objects.libsimpleio.Platforms Namespace](#)

PocketBeagleGPIO112 Field

Legacy GPIO pin designator for P2.32

Namespace: [IO.Objects.libsimpleio.Platforms](#)

Assembly: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

↳ Syntax

C# VB F#

[Copy](#)

```
public static readonly Designator GPIO112
```

Field Value

Type: [Designator](#)

↳ See Also

Reference

[PocketBeagle Class](#)

[IO.Objects.libsimpleio.Platforms Namespace](#)

PocketBeagleGPIO113 Field

Legacy GPIO pin designator for P2.30

Namespace: [IO.Objects.libsimpleio.Platforms](#)

Assembly: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

↳ Syntax

C# VB F#

[Copy](#)

```
public static readonly Designator GPIO113
```

Field Value

Type: [Designator](#)

↳ See Also

Reference

[PocketBeagle Class](#)

[IO.Objects.libsimpleio.Platforms Namespace](#)

PocketBeagleGPIO114 Field

Legacy GPIO pin designator for P1.31

Namespace: [IO.Objects.libsimpleio.Platforms](#)

Assembly: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

↳ Syntax

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

[Copy](#)

```
public static readonly Designator GPIO114
```

Field Value

Type: [Designator](#)

↳ See Also

Reference

[PocketBeagle Class](#)

[IO.Objects.libsimpleio.Platforms Namespace](#)

PocketBeagleGPIO115 Field

Legacy GPIO pin designator for P2.34

Namespace: [IO.Objects.libsimpleio.Platforms](#)

Assembly: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

↳ Syntax

C# VB F#

[Copy](#)

```
public static readonly Designator GPIO115
```

Field Value

Type: [Designator](#)

↳ See Also

Reference

[PocketBeagle Class](#)

[IO.Objects.libsimpleio.Platforms Namespace](#)

PocketBeagleGPIO116 Field

Legacy GPIO pin designator for P2.28

Namespace: [IO.Objects.libsimpleio.Platforms](#)

Assembly: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

↳ Syntax

C# VB F#

[Copy](#)

```
public static readonly Designator GPIO116
```

Field Value

Type: [Designator](#)

↳ See Also

Reference

[PocketBeagle Class](#)

[IO.Objects.libsimpleio.Platforms Namespace](#)

PocketBeagleGPIO117 Field

Legacy GPIO pin designator for P1.29

Namespace: [IO.Objects.libsimpleio.Platforms](#)

Assembly: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

↳ Syntax

C# VB F#

[Copy](#)

```
public static readonly Designator GPIO117
```

Field Value

Type: [Designator](#)

↳ See Also

Reference

[PocketBeagle Class](#)

[IO.Objects.libsimpleio.Platforms Namespace](#)

PocketBeagleGPIO12 Field

Legacy GPIO pin designator for P1.26

Namespace: [IO.Objects.libsimpleio.Platforms](#)

Assembly: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

↳ Syntax

C# VB F#

[Copy](#)

```
public static readonly Designator GPIO12
```

Field Value

Type: [Designator](#)

↳ See Also

Reference

[PocketBeagle Class](#)

[IO.Objects.libsimpleio.Platforms Namespace](#)

PocketBeagleGPIO13 Field

Legacy GPIO pin designator for P1.28

Namespace: [IO.Objects.libsimpleio.Platforms](#)

Assembly: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

↳ Syntax

C# VB F#

[Copy](#)

```
public static readonly Designator GPIO13
```

Field Value

Type: [Designator](#)

↳ See Also

Reference

[PocketBeagle Class](#)

[IO.Objects.libsimpleio.Platforms Namespace](#)

PocketBeagleGPIO14 Field

Legacy GPIO pin designator for P2.11

Namespace: [IO.Objects.libsimpleio.Platforms](#)

Assembly: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

↳ Syntax

C# VB F#

[Copy](#)

```
public static readonly Designator GPIO14
```

Field Value

Type: [Designator](#)

↳ See Also

Reference

[PocketBeagle Class](#)

[IO.Objects.libsimpleio.Platforms Namespace](#)

PocketBeagleGPIO15 Field

Legacy GPIO pin designator for P2.9

Namespace: [IO.Objects.libsimpleio.Platforms](#)

Assembly: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

↳ Syntax

C# VB F#

[Copy](#)

```
public static readonly Designator GPIO15
```

Field Value

Type: [Designator](#)

↳ See Also

Reference

[PocketBeagle Class](#)

[IO.Objects.libsimpleio.Platforms Namespace](#)

PocketBeagleGPIO19 Field

Legacy GPIO pin designator for P2.31

Namespace: [IO.Objects.libsimpleio.Platforms](#)

Assembly: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

↳ Syntax

C# VB F#

[Copy](#)

```
public static readonly Designator GPIO19
```

Field Value

Type: [Designator](#)

↳ See Also

Reference

[PocketBeagle Class](#)

[IO.Objects.libsimpleio.Platforms Namespace](#)

PocketBeagleGPIO2 Field

Legacy GPIO pin designator for P1.8

Namespace: [IO.Objects.libsimpleio.Platforms](#)

Assembly: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

↳ Syntax

C# VB F#

[Copy](#)

```
public static readonly Designator GPIO2
```

Field Value

Type: [Designator](#)

↳ See Also

Reference

[PocketBeagle Class](#)

[IO.Objects.libsimpleio.Platforms Namespace](#)

PocketBeagleGPIO20 Field

Legacy GPIO pin designator for P1.20

Namespace: [IO.Objects.libsimpleio.Platforms](#)

Assembly: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

↳ Syntax

C# VB F#

[Copy](#)

```
public static readonly Designator GPIO20
```

Field Value

Type: [Designator](#)

↳ See Also

Reference

[PocketBeagle Class](#)

[IO.Objects.libsimpleio.Platforms Namespace](#)

PocketBeagleGPIO23 Field

Legacy GPIO pin designator for P2.3

Namespace: [IO.Objects.libsimpleio.Platforms](#)

Assembly: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

↳ Syntax

C# VB F#

[Copy](#)

```
public static readonly Designator GPIO23
```

Field Value

Type: [Designator](#)

↳ See Also

Reference

[PocketBeagle Class](#)

[IO.Objects.libsimpleio.Platforms Namespace](#)

PocketBeagleGPIO26 Field

Legacy GPIO pin designator for P1.34

Namespace: [IO.Objects.libsimpleio.Platforms](#)

Assembly: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

↳ Syntax

C# VB F#

[Copy](#)

```
public static readonly Designator GPIO26
```

Field Value

Type: [Designator](#)

↳ See Also

Reference

[PocketBeagle Class](#)

[IO.Objects.libsimpleio.Platforms Namespace](#)

PocketBeagleGPIO27 Field

Legacy GPIO pin designator for P2.19

Namespace: [IO.Objects.libsimpleio.Platforms](#)

Assembly: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

↳ Syntax

C# VB F#

[Copy](#)

```
public static readonly Designator GPIO27
```

Field Value

Type: [Designator](#)

↳ See Also

Reference

[PocketBeagle Class](#)

[IO.Objects.libsimpleio.Platforms Namespace](#)

PocketBeagleGPIO3 Field

Legacy GPIO pin designator for P1.10

Namespace: [IO.Objects.libsimpleio.Platforms](#)

Assembly: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

↳ Syntax

C# VB F#

[Copy](#)

```
public static readonly Designator GPIO3
```

Field Value

Type: [Designator](#)

↳ See Also

Reference

[PocketBeagle Class](#)

[IO.Objects.libsimpleio.Platforms Namespace](#)

PocketBeagleGPIO30 Field

Legacy GPIO pin designator for P2.5

Namespace: [IO.Objects.libsimpleio.Platforms](#)

Assembly: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

↳ Syntax

C# VB F#

[Copy](#)

```
public static readonly Designator GPIO30
```

Field Value

Type: [Designator](#)

↳ See Also

Reference

[PocketBeagle Class](#)

[IO.Objects.libsimpleio.Platforms Namespace](#)

PocketBeagleGPIO31 Field

Legacy GPIO pin designator for P2.7

Namespace: [IO.Objects.libsimpleio.Platforms](#)

Assembly: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

↳ Syntax

C# VB F#

[Copy](#)

```
public static readonly Designator GPIO31
```

Field Value

Type: [Designator](#)

↳ See Also

Reference

[PocketBeagle Class](#)

[IO.Objects.libsimpleio.Platforms Namespace](#)

PocketBeagleGPIO4 Field

Legacy GPIO pin designator for P1.12

Namespace: [IO.Objects.libsimpleio.Platforms](#)

Assembly: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

↳ Syntax

C# VB F#

[Copy](#)

```
public static readonly Designator GPIO4
```

Field Value

Type: [Designator](#)

↳ See Also

Reference

[PocketBeagle Class](#)

[IO.Objects.libsimpleio.Platforms Namespace](#)

PocketBeagleGPIO40 Field

Legacy GPIO pin designator for P2.27

Namespace: [IO.Objects.libsimpleio.Platforms](#)

Assembly: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

↳ Syntax

C# VB F#

[Copy](#)

```
public static readonly Designator GPIO40
```

Field Value

Type: [Designator](#)

↳ See Also

Reference

[PocketBeagle Class](#)

[IO.Objects.libsimpleio.Platforms Namespace](#)

PocketBeagleGPIO41 Field

Legacy GPIO pin designator for P2.25

Namespace: [IO.Objects.libsimpleio.Platforms](#)

Assembly: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

↳ Syntax

C# VB F#

[Copy](#)

```
public static readonly Designator GPIO41
```

Field Value

Type: [Designator](#)

↳ See Also

Reference

[PocketBeagle Class](#)

[IO.Objects.libsimpleio.Platforms Namespace](#)

PocketBeagleGPIO42 Field

Legacy GPIO pin designator for P1.32

Namespace: [IO.Objects.libsimpleio.Platforms](#)

Assembly: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

↳ Syntax

C# VB F#

[Copy](#)

```
public static readonly Designator GPIO42
```

Field Value

Type: [Designator](#)

↳ See Also

Reference

[PocketBeagle Class](#)

[IO.Objects.libsimpleio.Platforms Namespace](#)

PocketBeagleGPIO43 Field

Legacy GPIO pin designator for P1.30

Namespace: [IO.Objects.libsimpleio.Platforms](#)

Assembly: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

↳ Syntax

C# VB F#

[Copy](#)

```
public static readonly Designator GPIO43
```

Field Value

Type: [Designator](#)

↳ See Also

Reference

[PocketBeagle Class](#)

[IO.Objects.libsimpleio.Platforms Namespace](#)

PocketBeagleGPIO44 Field

Legacy GPIO pin designator for P2.24

Namespace: [IO.Objects.libsimpleio.Platforms](#)

Assembly: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

↳ Syntax

C# VB F#

[Copy](#)

```
public static readonly Designator GPIO44
```

Field Value

Type: [Designator](#)

↳ See Also

Reference

[PocketBeagle Class](#)

[IO.Objects.libsimpleio.Platforms Namespace](#)

PocketBeagleGPIO45 Field

Legacy GPIO pin designator for P2.33

Namespace: [IO.Objects.libsimpleio.Platforms](#)

Assembly: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

↳ Syntax

C# VB F#

[Copy](#)

```
public static readonly Designator GPIO45
```

Field Value

Type: [Designator](#)

↳ See Also

Reference

[PocketBeagle Class](#)

[IO.Objects.libsimpleio.Platforms Namespace](#)

PocketBeagleGPIO46 Field

Legacy GPIO pin designator for P2.22

Namespace: [IO.Objects.libsimpleio.Platforms](#)

Assembly: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

↳ Syntax

C# VB F#

[Copy](#)

```
public static readonly Designator GPIO46
```

Field Value

Type: [Designator](#)

↳ See Also

Reference

[PocketBeagle Class](#)

[IO.Objects.libsimpleio.Platforms Namespace](#)

PocketBeagleGPIO47 Field

Legacy GPIO pin designator for P2.18

Namespace: [IO.Objects.libsimpleio.Platforms](#)

Assembly: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

↳ Syntax

C# VB F#

[Copy](#)

```
public static readonly Designator GPIO47
```

Field Value

Type: [Designator](#)

↳ See Also

Reference

[PocketBeagle Class](#)

[IO.Objects.libsimpleio.Platforms Namespace](#)

PocketBeagleGPIO5 Field

Legacy GPIO pin designator for P1.6

Namespace: [IO.Objects.libsimpleio.Platforms](#)

Assembly: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

↳ Syntax

C# VB F#

[Copy](#)

```
public static readonly Designator GPIO5
```

Field Value

Type: [Designator](#)

↳ See Also

Reference

[PocketBeagle Class](#)

[IO.Objects.libsimpleio.Platforms Namespace](#)

PocketBeagleGPIO50 Field

Legacy GPIO pin designator for P2.1

Namespace: [IO.Objects.libsimpleio.Platforms](#)

Assembly: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

↳ Syntax

C# VB F#

[Copy](#)

```
public static readonly Designator GPIO50
```

Field Value

Type: [Designator](#)

↳ See Also

Reference

[PocketBeagle Class](#)

[IO.Objects.libsimpleio.Platforms Namespace](#)

PocketBeagleGPIO52 Field

Legacy GPIO pin designator for P2.10

Namespace: [IO.Objects.libsimpleio.Platforms](#)

Assembly: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

↳ Syntax

C# VB F#

[Copy](#)

```
public static readonly Designator GPIO52
```

Field Value

Type: [Designator](#)

↳ See Also

Reference

[PocketBeagle Class](#)

[IO.Objects.libsimpleio.Platforms Namespace](#)

PocketBeagleGPIO57 Field

Legacy GPIO pin designator for P2.6

Namespace: [IO.Objects.libsimpleio.Platforms](#)

Assembly: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

« Syntax

C# VB F#

[Copy](#)

```
public static readonly Designator GPIO57
```

Field Value

Type: [Designator](#)

« See Also

Reference

[PocketBeagle Class](#)

[IO.Objects.libsimpleio.Platforms Namespace](#)

PocketBeagleGPIO58 Field

Legacy GPIO pin designator for P2.4

Namespace: [IO.Objects.libsimpleio.Platforms](#)

Assembly: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

↳ Syntax

C# VB F#

[Copy](#)

```
public static readonly Designator GPIO58
```

Field Value

Type: [Designator](#)

↳ See Also

Reference

[PocketBeagle Class](#)

[IO.Objects.libsimpleio.Platforms Namespace](#)

PocketBeagleGPIO59 Field

Legacy GPIO pin designator for P2.2

Namespace: [IO.Objects.libsimpleio.Platforms](#)

Assembly: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

↳ Syntax

C# VB F#

[Copy](#)

```
public static readonly Designator GPIO59
```

Field Value

Type: [Designator](#)

↳ See Also

Reference

[PocketBeagle Class](#)

[IO.Objects.libsimpleio.Platforms Namespace](#)

PocketBeagleGPIO60 Field

Legacy GPIO pin designator for P2.8

Namespace: [IO.Objects.libsimpleio.Platforms](#)

Assembly: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

↳ Syntax

C# VB F#

[Copy](#)

```
public static readonly Designator GPIO60
```

Field Value

Type: [Designator](#)

↳ See Also

Reference

[PocketBeagle Class](#)

[IO.Objects.libsimpleio.Platforms Namespace](#)

PocketBeagleGPIO64 Field

Legacy GPIO pin designator for P2.20

Namespace: [IO.Objects.libsimpleio.Platforms](#)

Assembly: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

↳ Syntax

C# VB F#

[Copy](#)

```
public static readonly Designator GPIO64
```

Field Value

Type: [Designator](#)

↳ See Also

Reference

[PocketBeagle Class](#)

[IO.Objects.libsimpleio.Platforms Namespace](#)

PocketBeagleGPIO65 Field

Legacy GPIO pin designator for P2.17

Namespace: [IO.Objects.libsimpleio.Platforms](#)

Assembly: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

↳ Syntax

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

[Copy](#)

```
public static readonly Designator GPIO65
```

Field Value

Type: [Designator](#)

↳ See Also

Reference

[PocketBeagle Class](#)

[IO.Objects.libsimpleio.Platforms Namespace](#)

PocketBeagleGPIO7 Field

Legacy GPIO pin designator for P2.29

Namespace: [IO.Objects.libsimpleio.Platforms](#)

Assembly: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

↳ Syntax

C# VB F#

[Copy](#)

```
public static readonly Designator GPIO7
```

Field Value

Type: [Designator](#)

↳ See Also

Reference

[PocketBeagle Class](#)

[IO.Objects.libsimpleio.Platforms Namespace](#)

PocketBeagleGPIO86 Field

Legacy GPIO pin designator for P2.35

Namespace: [IO.Objects.libsimpleio.Platforms](#)

Assembly: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

↳ Syntax

C# VB F#

[Copy](#)

```
public static readonly Designator GPIO86
```

Field Value

Type: [Designator](#)

↳ See Also

Reference

[PocketBeagle Class](#)

[IO.Objects.libsimpleio.Platforms Namespace](#)

PocketBeagleGPIO87 Field

Legacy GPIO pin designator for P1.2

Namespace: [IO.Objects.libsimpleio.Platforms](#)

Assembly: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

↳ Syntax

C# VB F#

[Copy](#)

```
public static readonly Designator GPIO87
```

Field Value

Type: [Designator](#)

↳ See Also

Reference

[PocketBeagle Class](#)

[IO.Objects.libsimpleio.Platforms Namespace](#)

PocketBeagleGPIO88 Field

Legacy GPIO pin designator for P1.35

Namespace: [IO.Objects.libsimpleio.Platforms](#)

Assembly: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

↳ Syntax

C# VB F#

[Copy](#)

```
public static readonly Designator GPIO88
```

Field Value

Type: [Designator](#)

↳ See Also

Reference

[PocketBeagle Class](#)

[IO.Objects.libsimpleio.Platforms Namespace](#)

PocketBeagleGPIO89 Field

Legacy GPIO pin designator for P1.4

Namespace: [IO.Objects.libsimpleio.Platforms](#)

Assembly: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

↳ Syntax

C# VB F#

[Copy](#)

```
public static readonly Designator GPIO89
```

Field Value

Type: [Designator](#)

↳ See Also

Reference

[PocketBeagle Class](#)

[IO.Objects.libsimpleio.Platforms Namespace](#)

PocketBeagleI2C1 Field

I2C bus designator for P1.26 and P1.28

Namespace: [IO.Objects.libsimpleio.Platforms](#)

Assembly: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

↳ Syntax

C# VB F#

[Copy](#)

```
public static readonly Designator I2C1
```

Field Value

Type: [Designator](#)

↳ See Also

Reference

[PocketBeagle Class](#)

[IO.Objects.libsimpleio.Platforms Namespace](#)

PocketBeagleI2C2 Field

I2C bus designator for P2.9 and P2.11

Namespace: [IO.Objects.libsimpleio.Platforms](#)

Assembly: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

« Syntax

C# VB F#

[Copy](#)

```
public static readonly Designator I2C2
```

Field Value

Type: [Designator](#)

« See Also

Reference

[PocketBeagle Class](#)

[IO.Objects.libsimpleio.Platforms Namespace](#)

PocketBeaglePWM0_0 Field

PWM output designator for P1.36

Namespace: [IO.Objects.libsimpleioPlatforms](#)

Assembly: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

↳ Syntax

C# VB F#

[Copy](#)

```
public static readonly Designator PWM0_0
```

Field Value

Type: [Designator](#)

↳ See Also

Reference

[PocketBeagle Class](#)

[IO.Objects.libsimpleioPlatforms Namespace](#)

PocketBeaglePWM2_0 Field

PWM output designator for P2.1

Namespace: [IO.Objects.libsimpleio.Platforms](#)

Assembly: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

« Syntax

C# VB F#

[Copy](#)

```
public static readonly Designator PWM2_0
```

Field Value

Type: [Designator](#)

« See Also

Reference

[PocketBeagle Class](#)

[IO.Objects.libsimpleio.Platforms Namespace](#)

PocketBeagleSPI0_0 Field

SPI slave select designator for P1.6

Namespace: [IO.Objects.libsimpleioPlatforms](#)

Assembly: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

↳ Syntax

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

[Copy](#)

```
public static readonly Designator SPI0_0
```

Field Value

Type: [Designator](#)

↳ See Also

Reference

[PocketBeagle Class](#)

[IO.Objects.libsimpleioPlatforms Namespace](#)

PocketBeagleSPI1_1 Field

SPI slave select designator for P2.31

Namespace: [IO.Objects.libsimpleioPlatforms](#)

Assembly: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

↳ Syntax

C# VB F#

[Copy](#)

```
public static readonly Designator SPI1_1
```

Field Value

Type: [Designator](#)

↳ See Also

Reference

[PocketBeagle Class](#)

[IO.Objects.libsimpleioPlatforms Namespace](#)

RaspberryPi Class

This class defines identifiers for the devices provided by the Raspberry Pi hardware platform.

▪ Inheritance Hierarchy

[SystemObject](#) [IO.Objects.libsimpleio.Platforms](#)[RaspberryPi](#)

Namespace: [IO.Objects.libsimpleio.Platforms](#)

Assembly: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

▪ Syntax

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

[Copy](#)

```
public static class RaspberryPi
```

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

▪ Fields

| | Name | Description |
|----------------------------|------------------------|----------------------------|
| • S | AIN0 | Analog input designator |
| • S | AIN1 | Analog input designator |
| • S | GPIO10 | Legacy GPIO pin designator |
| • S | GPIO11 | Legacy GPIO pin designator |
| • S | GPIO12 | Legacy GPIO pin designator |

| | | |
|-----|--------|----------------------------|
| • S | GPIO13 | Legacy GPIO pin designator |
| • S | GPIO14 | Legacy GPIO pin designator |
| • S | GPIO15 | Legacy GPIO pin designator |
| • S | GPIO16 | Legacy GPIO pin designator |
| • S | GPIO17 | Legacy GPIO pin designator |
| • S | GPIO18 | Legacy GPIO pin designator |
| • S | GPIO19 | Legacy GPIO pin designator |
| • S | GPIO2 | Legacy GPIO pin designator |
| • S | GPIO20 | Legacy GPIO pin designator |
| • S | GPIO21 | Legacy GPIO pin designator |
| • S | GPIO22 | Legacy GPIO pin designator |
| • S | GPIO23 | Legacy GPIO pin designator |
| • S | GPIO24 | Legacy GPIO pin designator |
| • S | GPIO25 | Legacy GPIO pin designator |
| • S | GPIO26 | Legacy GPIO pin designator |
| • S | GPIO27 | Legacy GPIO pin designator |
| • S | GPIO3 | Legacy GPIO pin designator |
| • S | GPIO4 | Legacy GPIO pin designator |
| • S | GPIO5 | Legacy GPIO pin designator |

| | | |
|------------|------------------------|--|
| • S | GPIO6 | Legacy GPIO pin designator |
| • S | GPIO7 | Legacy GPIO pin designator |
| • S | GPIO8 | Legacy GPIO pin designator |
| • S | GPIO9 | Legacy GPIO pin designator |
| • S | I2C1 | I2C bus designator for GPIO2 and GPIO3 |
| • S | PWM0_0 | PWM output designator for GPIO18 |
| • S | PWM0_1 | PWM output designator for GPIO19 |
| • S | SPI0_0 | SPI slave select designator for GPIO8 |
| • S | SPI0_1 | SPI slave select designator for GPIO7 |

[Top](#)

See Also

Reference

[IO.Objects.libsimpleio.Platforms Namespace](#)

RaspberryPi Fields

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

Fields

| | Name | Description |
|---|--------|----------------------------|
| •  | AIN0 | Analog input designator |
| •  | AIN1 | Analog input designator |
| •  | GPIO10 | Legacy GPIO pin designator |
| •  | GPIO11 | Legacy GPIO pin designator |
| •  | GPIO12 | Legacy GPIO pin designator |
| •  | GPIO13 | Legacy GPIO pin designator |
| •  | GPIO14 | Legacy GPIO pin designator |
| •  | GPIO15 | Legacy GPIO pin designator |
| •  | GPIO16 | Legacy GPIO pin designator |
| •  | GPIO17 | Legacy GPIO pin designator |
| •  | GPIO18 | Legacy GPIO pin designator |
| •  | GPIO19 | Legacy GPIO pin designator |
| •  | GPIO2 | Legacy GPIO pin designator |

| | | |
|------------|------------------------|--|
| • S | GPIO20 | Legacy GPIO pin designator |
| • S | GPIO21 | Legacy GPIO pin designator |
| • S | GPIO22 | Legacy GPIO pin designator |
| • S | GPIO23 | Legacy GPIO pin designator |
| • S | GPIO24 | Legacy GPIO pin designator |
| • S | GPIO25 | Legacy GPIO pin designator |
| • S | GPIO26 | Legacy GPIO pin designator |
| • S | GPIO27 | Legacy GPIO pin designator |
| • S | GPIO3 | Legacy GPIO pin designator |
| • S | GPIO4 | Legacy GPIO pin designator |
| • S | GPIO5 | Legacy GPIO pin designator |
| • S | GPIO6 | Legacy GPIO pin designator |
| • S | GPIO7 | Legacy GPIO pin designator |
| • S | GPIO8 | Legacy GPIO pin designator |
| • S | GPIO9 | Legacy GPIO pin designator |
| • S | I2C1 | I2C bus designator for GPIO2 and GPIO3 |
| • S | PWM0_0 | PWM output designator for GPIO18 |
| • S | PWM0_1 | PWM output designator for GPIO19 |
| • S | SPI0_0 | SPI slave select designator for GPIO8 |



[SPI0_1](#)

SPI slave select designator for GPIO7

[Top](#)

« See Also

Reference

[RaspberryPi Class](#)

[IO.Objects.libsimpleio.Platforms Namespace](#)

RaspberryPiAIN0 Field

Analog input designator

Namespace: [IO.Objects.libsimpleio.Platforms](#)

Assembly: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

▪ Syntax

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

[Copy](#)

```
public static readonly Designator AIN0
```

Field Value

Type: [Designator](#)

▪ Remarks

Requires the Mikroelektronika Pi 3 Click Shield and the [Pi3ClickShield](#) device tree overlay.

▪ See Also

Reference

[RaspberryPi Class](#)

[IO.Objects.libsimpleio.Platforms Namespace](#)

RaspberryPiAIN1 Field

Analog input designator

Namespace: [IO.Objects.libsimpleio.Platforms](#)

Assembly: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

▪ Syntax

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

[Copy](#)

```
public static readonly Designator AIN1
```

Field Value

Type: [Designator](#)

▪ Remarks

Requires the Mikroelektronika Pi 3 Click Shield and the [Pi3ClickShield](#) device tree overlay.

▪ See Also

Reference

[RaspberryPi Class](#)

[IO.Objects.libsimpleio.Platforms Namespace](#)

RaspberryPiGPIO10 Field

Legacy GPIO pin designator

Namespace: [IO.Objects.libsimpleio.Platforms](#)

Assembly: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

« Syntax

C# VB F#

[Copy](#)

```
public static readonly Designator GPIO10
```

Field Value

Type: [Designator](#)

« See Also

Reference

[RaspberryPi Class](#)

[IO.Objects.libsimpleio.Platforms Namespace](#)

RaspberryPiGPIO11 Field

Legacy GPIO pin designator

Namespace: [IO.Objects.libsimpleioPlatforms](#)

Assembly: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

« Syntax

C# VB F#

[Copy](#)

```
public static readonly Designator GPIO11
```

Field Value

Type: [Designator](#)

« See Also

Reference

[RaspberryPi Class](#)

[IO.Objects.libsimpleioPlatforms Namespace](#)

RaspberryPiGPIO12 Field

Legacy GPIO pin designator

Namespace: [IO.Objects.libsimpleioPlatforms](#)

Assembly: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

↳ Syntax

C# VB F#

[Copy](#)

```
public static readonly Designator GPIO12
```

Field Value

Type: [Designator](#)

↳ See Also

Reference

[RaspberryPi Class](#)

[IO.Objects.libsimpleioPlatforms Namespace](#)

RaspberryPiGPIO13 Field

Legacy GPIO pin designator

Namespace: [IO.Objects.libsimpleio.Platforms](#)

Assembly: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

« Syntax

C# VB F#

[Copy](#)

```
public static readonly Designator GPIO13
```

Field Value

Type: [Designator](#)

« See Also

Reference

[RaspberryPi Class](#)

[IO.Objects.libsimpleio.Platforms Namespace](#)

RaspberryPiGPIO14 Field

Legacy GPIO pin designator

Namespace: [IO.Objects.libsimpleio.Platforms](#)

Assembly: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

« Syntax

C# VB F#

[Copy](#)

```
public static readonly Designator GPIO14
```

Field Value

Type: [Designator](#)

« See Also

Reference

[RaspberryPi Class](#)

[IO.Objects.libsimpleio.Platforms Namespace](#)

RaspberryPiGPIO15 Field

Legacy GPIO pin designator

Namespace: [IO.Objects.libsimpleio.Platforms](#)

Assembly: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

« Syntax

C# VB F#

[Copy](#)

```
public static readonly Designator GPIO15
```

Field Value

Type: [Designator](#)

« See Also

Reference

[RaspberryPi Class](#)

[IO.Objects.libsimpleio.Platforms Namespace](#)

RaspberryPiGPIO16 Field

Legacy GPIO pin designator

Namespace: [IO.Objects.libsimpleio.Platforms](#)

Assembly: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

« Syntax

C# VB F#

[Copy](#)

```
public static readonly Designator GPIO16
```

Field Value

Type: [Designator](#)

« See Also

Reference

[RaspberryPi Class](#)

[IO.Objects.libsimpleio.Platforms Namespace](#)

RaspberryPiGPIO17 Field

Legacy GPIO pin designator

Namespace: [IO.Objects.libsimpleio.Platforms](#)

Assembly: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

« Syntax

C# VB F#

[Copy](#)

```
public static readonly Designator GPIO17
```

Field Value

Type: [Designator](#)

« See Also

Reference

[RaspberryPi Class](#)

[IO.Objects.libsimpleio.Platforms Namespace](#)

RaspberryPiGPIO18 Field

Legacy GPIO pin designator

Namespace: [IO.Objects.libsimpleioPlatforms](#)

Assembly: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

↳ Syntax

C# VB F#

[Copy](#)

```
public static readonly Designator GPIO18
```

Field Value

Type: [Designator](#)

↳ See Also

Reference

[RaspberryPi Class](#)

[IO.Objects.libsimpleioPlatforms Namespace](#)

RaspberryPiGPIO19 Field

Legacy GPIO pin designator

Namespace: [IO.Objects.libsimpleio.Platforms](#)

Assembly: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

« Syntax

C# VB F#

[Copy](#)

```
public static readonly Designator GPIO19
```

Field Value

Type: [Designator](#)

« See Also

Reference

[RaspberryPi Class](#)

[IO.Objects.libsimpleio.Platforms Namespace](#)

RaspberryPiGPIO2 Field

Legacy GPIO pin designator

Namespace: [IO.Objects.libsimpleio.Platforms](#)

Assembly: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

« Syntax

C# VB F#

[Copy](#)

```
public static readonly Designator GPIO2
```

Field Value

Type: [Designator](#)

« See Also

Reference

[RaspberryPi Class](#)

[IO.Objects.libsimpleio.Platforms Namespace](#)

RaspberryPiGPIO20 Field

Legacy GPIO pin designator

Namespace: [IO.Objects.libsimpleio.Platforms](#)

Assembly: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

« Syntax

C# VB F#

[Copy](#)

```
public static readonly Designator GPIO20
```

Field Value

Type: [Designator](#)

« See Also

Reference

[RaspberryPi Class](#)

[IO.Objects.libsimpleio.Platforms Namespace](#)

RaspberryPiGPIO21 Field

Legacy GPIO pin designator

Namespace: [IO.Objects.libsimpleioPlatforms](#)

Assembly: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

« Syntax

C# VB F#

[Copy](#)

```
public static readonly Designator GPIO21
```

Field Value

Type: [Designator](#)

« See Also

Reference

[RaspberryPi Class](#)

[IO.Objects.libsimpleioPlatforms Namespace](#)

RaspberryPiGPIO22 Field

Legacy GPIO pin designator

Namespace: [IO.Objects.libsimpleio.Platforms](#)

Assembly: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

« Syntax

C# VB F#

[Copy](#)

```
public static readonly Designator GPIO22
```

Field Value

Type: [Designator](#)

« See Also

Reference

[RaspberryPi Class](#)

[IO.Objects.libsimpleio.Platforms Namespace](#)

RaspberryPiGPIO23 Field

Legacy GPIO pin designator

Namespace: [IO.Objects.libsimpleio.Platforms](#)

Assembly: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

« Syntax

C# VB F#

[Copy](#)

```
public static readonly Designator GPIO23
```

Field Value

Type: [Designator](#)

« See Also

Reference

[RaspberryPi Class](#)

[IO.Objects.libsimpleio.Platforms Namespace](#)

RaspberryPiGPIO24 Field

Legacy GPIO pin designator

Namespace: [IO.Objects.libsimpleioPlatforms](#)

Assembly: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

↳ Syntax

C# VB F#

[Copy](#)

```
public static readonly Designator GPIO24
```

Field Value

Type: [Designator](#)

↳ See Also

Reference

[RaspberryPi Class](#)

[IO.Objects.libsimpleioPlatforms Namespace](#)

RaspberryPiGPIO25 Field

Legacy GPIO pin designator

Namespace: [IO.Objects.libsimpleio.Platforms](#)

Assembly: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

« Syntax

C# VB F#

[Copy](#)

```
public static readonly Designator GPIO25
```

Field Value

Type: [Designator](#)

« See Also

Reference

[RaspberryPi Class](#)

[IO.Objects.libsimpleio.Platforms Namespace](#)

RaspberryPiGPIO26 Field

Legacy GPIO pin designator

Namespace: [IO.Objects.libsimpleio.Platforms](#)

Assembly: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

« Syntax

C# VB F#

[Copy](#)

```
public static readonly Designator GPIO26
```

Field Value

Type: [Designator](#)

« See Also

Reference

[RaspberryPi Class](#)

[IO.Objects.libsimpleio.Platforms Namespace](#)

RaspberryPiGPIO27 Field

Legacy GPIO pin designator

Namespace: [IO.Objects.libsimpleio.Platforms](#)

Assembly: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

« Syntax

C# VB F#

[Copy](#)

```
public static readonly Designator GPIO27
```

Field Value

Type: [Designator](#)

« See Also

Reference

[RaspberryPi Class](#)

[IO.Objects.libsimpleio.Platforms Namespace](#)

RaspberryPiGPIO3 Field

Legacy GPIO pin designator

Namespace: [IO.Objects.libsimpleio.Platforms](#)

Assembly: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

« Syntax

C# VB F#

[Copy](#)

```
public static readonly Designator GPIO3
```

Field Value

Type: [Designator](#)

« See Also

Reference

[RaspberryPi Class](#)

[IO.Objects.libsimpleio.Platforms Namespace](#)

RaspberryPiGPIO4 Field

Legacy GPIO pin designator

Namespace: [IO.Objects.libsimpleio.Platforms](#)

Assembly: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

« Syntax

C# VB F#

[Copy](#)

```
public static readonly Designator GPIO4
```

Field Value

Type: [Designator](#)

« See Also

Reference

[RaspberryPi Class](#)

[IO.Objects.libsimpleio.Platforms Namespace](#)

RaspberryPiGPIO5 Field

Legacy GPIO pin designator

Namespace: [IO.Objects.libsimpleio.Platforms](#)

Assembly: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

« Syntax

C# VB F#

[Copy](#)

```
public static readonly Designator GPIO5
```

Field Value

Type: [Designator](#)

« See Also

Reference

[RaspberryPi Class](#)

[IO.Objects.libsimpleio.Platforms Namespace](#)

RaspberryPiGPIO6 Field

Legacy GPIO pin designator

Namespace: [IO.Objects.libsimpleio.Platforms](#)

Assembly: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

« Syntax

C# VB F#

[Copy](#)

```
public static readonly Designator GPIO6
```

Field Value

Type: [Designator](#)

« See Also

Reference

[RaspberryPi Class](#)

[IO.Objects.libsimpleio.Platforms Namespace](#)

RaspberryPiGPIO7 Field

Legacy GPIO pin designator

Namespace: [IO.Objects.libsimpleio.Platforms](#)

Assembly: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

↳ Syntax

C# VB F#

[Copy](#)

```
public static readonly Designator GPIO7
```

Field Value

Type: [Designator](#)

↳ See Also

Reference

[RaspberryPi Class](#)

[IO.Objects.libsimpleio.Platforms Namespace](#)

RaspberryPiGPIO8 Field

Legacy GPIO pin designator

Namespace: [IO.Objects.libsimpleio.Platforms](#)

Assembly: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

« Syntax

C# VB F#

[Copy](#)

```
public static readonly Designator GPIO8
```

Field Value

Type: [Designator](#)

« See Also

Reference

[RaspberryPi Class](#)

[IO.Objects.libsimpleio.Platforms Namespace](#)

RaspberryPiGPIO9 Field

Legacy GPIO pin designator

Namespace: [IO.Objects.libsimpleio.Platforms](#)

Assembly: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

« Syntax

C# VB F#

[Copy](#)

```
public static readonly Designator GPIO9
```

Field Value

Type: [Designator](#)

« See Also

Reference

[RaspberryPi Class](#)

[IO.Objects.libsimpleio.Platforms Namespace](#)

RaspberryPiI2C1 Field

I2C bus designator for GPIO2 and GPIO3

Namespace: [IO.Objects.libsimpleio.Platforms](#)

Assembly: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

↳ Syntax

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

[Copy](#)

```
public static readonly Designator I2C1
```

Field Value

Type: [Designator](#)

↳ See Also

Reference

[RaspberryPi Class](#)

[IO.Objects.libsimpleio.Platforms Namespace](#)

RaspberryPiPWM0_0 Field

PWM output designator for GPIO18

Namespace: [IO.Objects.libsimpleio.Platforms](#)

Assembly: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

▪ Syntax

C# VB F#

[Copy](#)

```
public static readonly Designator PWM0_0
```

Field Value

Type: [Designator](#)

▪ Remarks

Requires the [pwm](#) device tree overlay.

▪ See Also

Reference

[RaspberryPi Class](#)

[IO.Objects.libsimpleio.Platforms Namespace](#)

RaspberryPiPWM0_1 Field

PWM output designator for GPIO19

Namespace: [IO.Objects.libsimpleio.Platforms](#)

Assembly: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

▪ Syntax

C# VB F#

[Copy](#)

```
public static readonly Designator PWM0_1
```

Field Value

Type: [Designator](#)

▪ Remarks

Requires the [pwm](#) device tree overlay.

▪ See Also

Reference

[RaspberryPi Class](#)

[IO.Objects.libsimpleio.Platforms Namespace](#)

RaspberryPiSPI0_0 Field

SPI slave select designator for GPIO8

Namespace: [IO.Objects.libsimpleio.Platforms](#)

Assembly: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

« Syntax

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

[Copy](#)

```
public static readonly Designator SPI0_0
```

Field Value

Type: [Designator](#)

« See Also

Reference

[RaspberryPi Class](#)

[IO.Objects.libsimpleio.Platforms Namespace](#)

RaspberryPiSPI0_1 Field

SPI slave select designator for GPIO7

Namespace: [IO.Objects.libsimpleio.Platforms](#)

Assembly: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

« Syntax

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

[Copy](#)

```
public static readonly Designator SPI0_1
```

Field Value

Type: [Designator](#)

« See Also

Reference

[RaspberryPi Class](#)

[IO.Objects.libsimpleio.Platforms Namespace](#)

IO.Objects.libsimpleio.PWM Namespace

PWM (Pulse Width Modulated) Output Services

↳ Classes

| Class | Description |
|--|--|
|  Output | Encapsulates Linux PWM outputs using libsimpleio . |

Output Class

Encapsulates Linux PWM outputs using [libsimpleio](#).

▪ Inheritance Hierarchy

[SystemObject](#) [IO.Objects.libsimpleio.PWMOutput](#)

Namespace: [IO.Objects.libsimpleio.PWM](#)

Assembly: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

▪ 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.



fd

Read-only property returning the Linux file descriptor for the PWM output.

[Top](#)

◀ See Also

Reference

[IO.Objects.libsimpleio.PWM Namespace](#)

Output Constructor

Constructor for a single PWM output.

Namespace: [IO.Objects.libsimpleio.PWM](#)

Assembly: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

◀ Syntax

C# VB F#

[Copy](#)

```
public Output(  
    Designator desg,  
    int frequency,  
    double dutycycle = 0,  
    int polarity = 1  
)
```

Parameters

desg

Type: [IO.Objects.libsimpleio.DeviceDesignator](#)

PWM output designator.

frequency

Type: [SystemInt32](#)

PWM pulse frequency.

dutycycle (Optional)

Type: [SystemDouble](#)

Initial PWM output duty cycle. Allowed values are 0.0 to 100.0 percent.

polarity (Optional)

Type: [SystemInt32](#)
PWM output polarity.

◀ See Also

[Reference](#)

[Output Class](#)

[IO.Objects.libsimpleio.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. |
|  | fd | Read-only property returning the Linux file descriptor for the PWM output. |

[Top](#)

See Also

Reference

[Output Class](#)

[IO.Objects.libsimpleio.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.Objects.libsimpleio.PWM](#)

Assembly: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

↳ Syntax

C# VB F#

[Copy](#)

```
public double dutycycle { set; }
```

Property Value

Type: [Double](#)

Implements

[Outputdutycycle](#)

↳ See Also

Reference

[Output Class](#)

[IO.Objects.libsimpleio.PWM Namespace](#)

Outputfd Property

Read-only property returning the Linux file descriptor for the PWM output.

Namespace: [IO.Objects.libsimpleio.PWM](#)

Assembly: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

▲ Syntax

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

[Copy](#)

```
public int fd { get; }
```

Property Value

Type: [Int32](#)

▲ See Also

Reference

[Output Class](#)

[IO.Objects.libsimpleio.PWM Namespace](#)

IO.Objects.libsimpleio.Servo Namespace

Servo Output Services

↳ Classes

| Class | Description |
|--|--|
|  Output | Encapsulates Linux servo outputs using libsimpleio . |

Output Class

Encapsulates Linux servo outputs using [libsimpleio](#).

▪ Inheritance Hierarchy

[SystemObject](#) [IO.Objects.libsimpleio.ServoOutput](#)

Namespace: [IO.Objects.libsimpleio.Servo](#)

Assembly: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

▪ 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 |
|---|--------------------|--|
|  | fd | Read-only property returning the Linux file descriptor for the servo output. |



[position](#) Write-only property for setting the servo position. Allowed values are -1.0 to +1.0.

[Top](#)

◀ See Also

Reference

[IO.Objects.libsimpleio.Servo Namespace](#)

Output Constructor

Constructor for a single servo output.

Namespace: [IO.Objects.libsimpleio.Servo](#)

Assembly: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

▪ Syntax

C# VB F#

[Copy](#)

```
public Output(  
    Designator desg,  
    int frequency = 50,  
    double position = 0  
)
```

Parameters

desg

Type: [IO.Objects.libsimpleio.DeviceDesignator](#)

PWM output designator.

frequency (Optional)

Type: [SystemInt32](#)

PWM pulse frequency.

position (Optional)

Type: [SystemDouble](#)

Initial servo position.

▪ See Also

[Reference](#)

[Output Class](#)

IO.Objects.libsimpleio.Servo Namespace

Output Properties

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

Properties

| | Name | Description |
|---|--------------------------|--|
|  | fd | Read-only property returning the Linux file descriptor for the servo output. |
|  | 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.libsimpleio.Servo Namespace](#)

Outputfd Property

Read-only property returning the Linux file descriptor for the servo output.

Namespace: [IO.Objects.libsimpleio.Servo](#)

Assembly: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

▲ Syntax

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

[Copy](#)

```
public int fd { get; }
```

Property Value

Type: [Int32](#)

▲ See Also

Reference

[Output Class](#)

[IO.Objects.libsimpleio.Servo Namespace](#)

Outputposition Property

Write-only property for setting the servo position. Allowed values are -1.0 to +1.0.

Namespace: [IO.Objects.libsimpleio.Servo](#)

Assembly: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

▲ Syntax

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

[Copy](#)

```
public double position { set; }
```

Property Value

Type: [Double](#)

Implements

[Outputposition](#)

▲ See Also

[Reference](#)

[Output Class](#)

[IO.Objects.libsimpleio.Servo Namespace](#)

IO.Objects.libsimpleio.SPI Namespace

SPI (Serial Peripheral Interconnect) Device Services

◀ Classes

| Class | Description |
|--|--|
|  Device | Encapsulates Linux SPI devices using libsimpleio . |

Device Class

Encapsulates Linux SPI devices using [libsimpleio](#).

▪ Inheritance Hierarchy

[SystemObject](#) [IO.Objects.libsimpleio.SPIDevice](#)

Namespace: [IO.Objects.libsimpleio.SPI](#)

Assembly: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

▪ Syntax

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

[Copy](#)

```
public class Device : Device
```

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

▪ Constructors

| | Name | Description |
|---|--|--------------------------------------|
| ≡ | Device(Designator, Int32, Int32, Int32, Pin) | Constructor for a single SPI device. |
| ≡ | Device(String, Int32, Int32, Int32, Pin) | Constructor for a single SPI device. |

[Top](#)

▪ Properties

| | Name | Description |
|---|--------------------|--|
|  | fd | Read-only property returning the Linux file descriptor for the 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](#)

◀ Fields

| | Name | Description |
|---|--------------------------------|----------------------------|
|   | AUTOCHIPSELECT | Use hardware slave select. |

[Top](#)

◀ See Also

Reference

[IO.Objects.libsimpleio.SPI Namespace](#)

Device Constructor

↳ Overload List

| | Name | Description |
|---|--|--------------------------------------|
| ≡ | Device(Designator, Int32, Int32, Int32, Pin) | Constructor for a single SPI device. |
| ≡ | Device(String, Int32, Int32, Int32, Pin) | Constructor for a single SPI device. |

[Top](#)

↳ See Also

[Reference](#)

[Device Class](#)

[IO.Objects.libsimpleio.SPI Namespace](#)

Device Constructor (Designator, Int32, Int32, Int32, Pin)

Constructor for a single SPI device.

Namespace: [IO.Objects.libsimpleio.SPI](#)

Assembly: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

◀ Syntax

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

[Copy](#)

```
public Device(  
    Designator desg,  
    int mode,  
    int wordsize,  
    int speed,  
    Pin cspin = null  
)
```

Parameters

desg

Type: [IO.Objects.libsimpleio.DeviceDesignator](#)

SPI device designator.

mode

Type: [SystemInt32](#)

SPI clock mode.

wordsize

Type: [SystemInt32](#)

SPI transfer word size.

speed

Type: [SystemInt32](#)

SPI transfer speed.

cspin (Optional)

Type: [IO.Objects.libsimpleio.GPIOPin](#)

SPI slave select GPIO pin number, or [AUTOCHIPSELECT](#).

See Also

Reference

[Device Class](#)

[Device Overload](#)

[IO.Objects.libsimpleio.SPI Namespace](#)

Device Constructor (String, Int32, Int32, Int32, Pin)

Constructor for a single SPI device.

Namespace: [IO.Objects.libsimpleio.SPI](#)

Assembly: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

◀ Syntax

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

[Copy](#)

```
public Device(  
    string devname,  
    int mode,  
    int wordsize,  
    int speed,  
    Pin cspin = null  
)
```

Parameters

devname

Type: [SystemString](#)

SPI device node name.

mode

Type: [SystemInt32](#)

SPI clock mode.

wordsize

Type: [SystemInt32](#)

SPI transfer word size.

speed

Type: [SystemInt32](#)

SPI transfer speed.

cspin (Optional)

Type: [IO.Objects.libsimpleio.GPIOPin](#)

SPI slave select GPIO pin number, or [AUTOCHIPSELECT](#).

See Also

Reference

[Device Class](#)

[Device Overload](#)

[IO.Objects.libsimpleio.SPI Namespace](#)

Device Properties

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

▪ Properties

| | Name | Description |
|---|------|--|
|  | fd | Read-only property returning the Linux file descriptor for the SPI slave device. |

[Top](#)

▪ See Also

[Reference](#)

[Device Class](#)

[IO.Objects.libsimpleio.SPI Namespace](#)

Devicefd Property

Read-only property returning the Linux file descriptor for the SPI slave device.

Namespace: [IO.Objects.libsimpleio.SPI](#)

Assembly: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

◀ Syntax

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

[Copy](#)

```
public int fd { get; }
```

Property Value

Type: [Int32](#)

◀ See Also

Reference

[Device Class](#)

[IO.Objects.libsimpleio.SPI Namespace](#)

Device Methods

The [Device](#) 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

[Device Class](#)

[IO.Objects.libsimpleio.SPI Namespace](#)

DeviceRead Method

Read bytes from an SPI slave device.

Namespace: [IO.Objects.libsimpleio.SPI](#)

Assembly: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

◀ 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.

Implements

[DeviceRead\(Byte, Int32\)](#)

◀ See Also

Reference

[Device Class](#)

[IO.Objects.libsimpleio.SPI Namespace](#)

DeviceTransaction Method

Write and read bytes to and from an SPI slave device.

Namespace: [IO.Objects.libsimpleio.SPI](#)

Assembly: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

◀ 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 write and read operations.

Implements

[DeviceTransaction\(Byte, Int32, Byte, Int32, Int32\)](#)

See Also

Reference

[Device Class](#)

[IO.Objects.libsimpleio.SPI Namespace](#)

DeviceWrite Method

Write bytes to an SPI slave device.

Namespace: [IO.Objects.libsimpleio.SPI](#)

Assembly: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

↳ 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.

Implements

[DeviceWrite\(Byte, Int32\)](#)

↳ See Also

Reference

[Device Class](#)

[IO.Objects.libsimpleio.SPI Namespace](#)

Device Fields

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

Fields

| | Name | Description |
|---|--------------------------------|----------------------------|
|   | AUTOCHIPSELECT | Use hardware slave select. |

[Top](#)

See Also

Reference

[Device Class](#)

[IO.Objects.libsimpleio.SPI Namespace](#)

DeviceAUTOCHIPSELECT Field

Use hardware slave select.

Namespace: [IO.Objects.libsimpleio.SPI](#)

Assembly: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

◀ Syntax

C# VB F#

[Copy](#)

```
public const Pin AUTOCHIPSELECT = null
```

Field Value

Type: [Pin](#)

◀ See Also

Reference

[Device Class](#)

[IO.Objects.libsimpleio.SPI Namespace](#)

IO.Objects.libsimpleio.syslog Namespace

Error Logging Services Using the Linux [syslog](#) Subsystem

↳ Classes

| Class | Description |
|--|--|
|  Logger | Encapsulates system logging services using libsimpleio . |

Logger Class

Encapsulates system logging services using [libsimpleio](#).

▪ Inheritance Hierarchy

[SystemObject](#) [IO.Objects.libsimpleio.syslogLogger](#)

Namespace: [IO.Objects.libsimpleio.syslog](#)

Assembly: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

▪ Syntax

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

[Copy](#)

```
public class Logger : Logger
```

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

▪ Constructors

| | Name | Description |
|---|------------------------|--|
| ≡ | Logger | Constructor for a logging object that uses the Linux syslog subsystem. |

[Top](#)

▪ Methods

| | Name | Description |
|---|-------------------------------|-----------------------|
| ≡ | Error(String) | Log an error message. |

| | | |
|---|----------------------|---|
| ≡ | Error(String, Int32) | Log an error message, including an <code>errno</code> error string. |
| ≡ | Note | Log a notification message. |
| ≡ | Warning | Log a warning message. |

[Top](#)

◀ Fields

| | Name | Description |
|------------|----------------------------|--|
| ♦ s | LOG_AUTH | Authentication facility. |
| ♦ s | LOG_DAEMON | System daemon/background process facility. |
| ♦ s | LOG_LOCAL0 | Locally defined facility. |
| ♦ s | LOG_LOCAL1 | Locally defined facility. |
| ♦ s | LOG_LOCAL2 | Locally defined facility. |
| ♦ s | LOG_LOCAL3 | Locally defined facility. |
| ♦ s | LOG_LOCAL4 | Locally defined facility. |
| ♦ s | LOG_LOCAL5 | Locally defined facility. |
| ♦ s | LOG_LOCAL6 | Locally defined facility. |
| ♦ s | LOG_LOCAL7 | Locally defined facility. |
| ♦ s | LOG_MAIL | Mail subsystem facility. |
| ♦ s | LOG_NDELAY | Open the connection to the |

`syslog` subsystem immediately.
Recommended.

-
- **s** `LOG_ODELAY` Do not open the connection to the `syslog` subsystem before logging the first message. Not recommended.
 - **s** `LOG_PERROR` Write message to both `syslog` subsystem AND `stderr`.
 - **s** `LOG_PID` Prepend the caller's process ID to the message.
 - **s** `LOG_USER` User program facility. Use `LOG_LOCALx` instead.
-

[Top](#)

See Also

Reference

[IO.Objects.libsimpleio.syslog Namespace](#)

Logger Constructor

Constructor for a logging object that uses the Linux `syslog` subsystem.

Namespace: [IO.Objects.libsimpleio.syslog](#)

Assembly: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

◀ Syntax

C# VB F#

[Copy](#)

```
public Logger(  
    string id,  
    int facility,  
    int options = 33  
)
```

Parameters

id

Type: [SystemString](#)

Program identifier string.

facility

Type: [SystemInt32](#)

`syslog` facility.

options (Optional)

Type: [SystemInt32](#)

`syslog` options.

◀ See Also

[Reference](#)

[Logger Class](#)

IO.Objects.libsimpleio.syslog Namespace

Logger Methods

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

▪ Methods

| | Name | Description |
|---|--------------------------------------|---|
| ≡ | Error(String) | Log an error message. |
| ≡ | Error(String, Int32) | Log an error message, including an <code>errno</code> error string. |
| ≡ | Note | Log a notification message. |
| ≡ | Warning | Log a warning message. |

[Top](#)

▪ See Also

Reference

[Logger Class](#)

[IO.Objects.libsimpleio.syslog Namespace](#)

LoggerError Method

↳ Overload List

| | Name | Description |
|---|--------------------------------------|---|
| ≡ | Error(String) | Log an error message. |
| ≡ | Error(String, Int32) | Log an error message, including an <code>errno</code> error string. |

[Top](#)

↳ See Also

[Reference](#)

[Logger Class](#)

[IO.Objects.libsimpleio.syslog Namespace](#)

LoggerError Method (String)

Log an error message.

Namespace: [IO.Objects.libsimpleio.syslog](#)

Assembly: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

◀ Syntax

C# VB F#

[Copy](#)

```
public void Error(  
    string message  
)
```

Parameters

message

Type: [System.String](#)

Error message.

Implements

[LoggerError\(String\)](#)

◀ See Also

Reference

[Logger Class](#)

Error Overload

[IO.Objects.libsimpleio.syslog Namespace](#)

LoggerError Method (String, Int32)

Log an error message, including an `errno` error string.

Namespace: [IO.Objects.libsimpleio.syslog](#)

Assembly: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

◀ Syntax

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

Copy

```
public void Error(  
    string message,  
    int errnum  
)
```

Parameters

message

Type: [SystemString](#)

Error Message.

errnum

Type: [SystemInt32](#)

`errno` error number.

Implements

[LoggerError\(String, Int32\)](#)

◀ See Also

Reference

Logger Class
Error Overload
IO.Objects.libsimpleio.syslog Namespace

LoggerNote Method

Log a notification message.

Namespace: [IO.Objects.libsimpleio.syslog](#)

Assembly: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

◀ Syntax

C# VB F#

[Copy](#)

```
public void Note(  
    string message  
)
```

Parameters

message

Type: [SystemString](#)

Notification message.

Implements

[LoggerNote\(String\)](#)

◀ See Also

Reference

[Logger Class](#)

[IO.Objects.libsimpleio.syslog Namespace](#)

LoggerWarning Method

Log a warning message.

Namespace: [IO.Objects.libsimpleio.syslog](#)

Assembly: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

◀ Syntax

C# VB F#

[Copy](#)

```
public void Warning(  
    string message  
)
```

Parameters

message

Type: [System.String](#)

Warning message.

Implements

[LoggerWarning\(String\)](#)

◀ See Also

Reference

[Logger Class](#)

[IO.Objects.libsimpleio.syslog Namespace](#)

Logger Fields

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

Fields

| | Name | Description |
|---------------------|----------------------------|--|
| • S | LOG_AUTH | Authentication facility. |
| • S | LOG_DAEMON | System daemon/background process facility. |
| • S | LOG_LOCAL0 | Locally defined facility. |
| • S | LOG_LOCAL1 | Locally defined facility. |
| • S | LOG_LOCAL2 | Locally defined facility. |
| • S | LOG_LOCAL3 | Locally defined facility. |
| • S | LOG_LOCAL4 | Locally defined facility. |
| • S | LOG_LOCAL5 | Locally defined facility. |
| • S | LOG_LOCAL6 | Locally defined facility. |
| • S | LOG_LOCAL7 | Locally defined facility. |
| • S | LOG_MAIL | Mail subsystem facility. |
| • S | LOG_NDELAY | Open the connection to the syslog subsystem immediately. |

Recommended.

-
- ❖ **S** [LOG_ODELAY](#) Do not open the connection to the `syslog` subsystem before logging the first message. Not recommended.
 - ❖ **S** [LOG_PERROR](#) Write message to both `syslog` subsystem AND `stderr`.
 - ❖ **S** [LOG_PID](#) Prepend the caller's process ID to the message.
 - ❖ **S** [LOG_USER](#) User program facility. Use `LOG_LOCALx` instead.
-

[Top](#)

See Also

[Reference](#)

[Logger Class](#)

[IO.Objects.libsimpleio.syslog Namespace](#)

LoggerLOG_AUTH Field

Authentication facility.

Namespace: [IO.Objects.libsimpleio.syslog](#)

Assembly: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

◀ Syntax

C# VB F#

[Copy](#)

```
public const int LOG_AUTH = 32
```

Field Value

Type: [Int32](#)

◀ See Also

Reference

[Logger Class](#)

[IO.Objects.libsimpleio.syslog Namespace](#)

LoggerLOG_DAEMON Field

System daemon/background process facility.

Namespace: [IO.Objects.libsimpleio.syslog](#)

Assembly: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

◀ Syntax

C# VB F#

[Copy](#)

```
public const int LOG_DAEMON = 24
```

Field Value

Type: [Int32](#)

◀ See Also

Reference

[Logger Class](#)

[IO.Objects.libsimpleio.syslog Namespace](#)

LoggerLOG_LOCAL0 Field

Locally defined facility.

Namespace: [IO.Objects.libsimpleio.syslog](#)

Assembly: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

◀ Syntax

C# VB F#

[Copy](#)

```
public const int LOG_LOCAL0 = 128
```

Field Value

Type: [Int32](#)

◀ See Also

Reference

[Logger Class](#)

[IO.Objects.libsimpleio.syslog Namespace](#)

LoggerLOG_LOCAL1 Field

Locally defined facility.

Namespace: [IO.Objects.libsimpleio.syslog](#)

Assembly: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

◀ Syntax

C# VB F#

[Copy](#)

```
public const int LOG_LOCAL1 = 136
```

Field Value

Type: [Int32](#)

◀ See Also

Reference

[Logger Class](#)

[IO.Objects.libsimpleio.syslog Namespace](#)

LoggerLOG_LOCAL2 Field

Locally defined facility.

Namespace: [IO.Objects.libsimpleio.syslog](#)

Assembly: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

◀ Syntax

C# VB F#

[Copy](#)

```
public const int LOG_LOCAL2 = 144
```

Field Value

Type: [Int32](#)

◀ See Also

Reference

[Logger Class](#)

[IO.Objects.libsimpleio.syslog Namespace](#)

LoggerLOG_LOCAL3 Field

Locally defined facility.

Namespace: [IO.Objects.libsimpleio.syslog](#)

Assembly: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

◀ Syntax

C# VB F#

[Copy](#)

```
public const int LOG_LOCAL3 = 152
```

Field Value

Type: [Int32](#)

◀ See Also

Reference

[Logger Class](#)

[IO.Objects.libsimpleio.syslog Namespace](#)

LoggerLOG_LOCAL4 Field

Locally defined facility.

Namespace: [IO.Objects.libsimpleio.syslog](#)

Assembly: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

◀ Syntax

C# VB F#

[Copy](#)

```
public const int LOG_LOCAL4 = 160
```

Field Value

Type: [Int32](#)

◀ See Also

Reference

[Logger Class](#)

[IO.Objects.libsimpleio.syslog Namespace](#)

LoggerLOG_LOCAL5 Field

Locally defined facility.

Namespace: [IO.Objects.libsimpleio.syslog](#)

Assembly: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

◀ Syntax

C# VB F#

[Copy](#)

```
public const int LOG_LOCAL5 = 168
```

Field Value

Type: [Int32](#)

◀ See Also

Reference

[Logger Class](#)

[IO.Objects.libsimpleio.syslog Namespace](#)

LoggerLOG_LOCAL6 Field

Locally defined facility.

Namespace: [IO.Objects.libsimpleio.syslog](#)

Assembly: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

◀ Syntax

C# VB F#

[Copy](#)

```
public const int LOG_LOCAL6 = 176
```

Field Value

Type: [Int32](#)

◀ See Also

Reference

[Logger Class](#)

[IO.Objects.libsimpleio.syslog Namespace](#)

LoggerLOG_LOCAL7 Field

Locally defined facility.

Namespace: [IO.Objects.libsimpleio.syslog](#)

Assembly: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

◀ Syntax

C# VB F#

[Copy](#)

```
public const int LOG_LOCAL7 = 184
```

Field Value

Type: [Int32](#)

◀ See Also

Reference

[Logger Class](#)

[IO.Objects.libsimpleio.syslog Namespace](#)

LoggerLOG_MAIL Field

Mail subsystem facility.

Namespace: [IO.Objects.libsimpleio.syslog](#)

Assembly: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

◀ Syntax

C# VB F#

[Copy](#)

```
public const int LOG_MAIL = 16
```

Field Value

Type: [Int32](#)

◀ See Also

Reference

[Logger Class](#)

[IO.Objects.libsimpleio.syslog Namespace](#)

LoggerLOG_NDELAY Field

Open the connection to the `syslog` subsystem immediately.
Recommended.

Namespace: [IO.Objects.libsimpleio.syslog](#)

Assembly: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

◀ Syntax

C# VB F#

[Copy](#)

```
public const int LOG_NDELAY = 8
```

Field Value

Type: [Int32](#)

◀ See Also

Reference

[Logger Class](#)

[IO.Objects.libsimpleio.syslog Namespace](#)

LoggerLOG_ODELAY Field

Do not open the connection to the `syslog` subsystem before logging the first message. Not recommended.

Namespace: [IO.Objects.libsimpleio.syslog](#)

Assembly: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

◀ Syntax

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

[Copy](#)

```
public const int LOG_ODELAY = 8
```

Field Value

Type: [Int32](#)

◀ See Also

Reference

[Logger Class](#)

[IO.Objects.libsimpleio.syslog Namespace](#)

LoggerLOG_PERROR Field

Write message to both `syslog` subsystem AND `stderr`.

Namespace: [IO.Objects.libsimpleio.syslog](#)

Assembly: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

◀ Syntax

C# VB F#

[Copy](#)

```
public const int LOG_PERROR = 32
```

Field Value

Type: [Int32](#)

◀ See Also

Reference

[Logger Class](#)

[IO.Objects.libsimpleio.syslog Namespace](#)

LoggerLOG_PID Field

Prepend the caller's process ID to the message.

Namespace: [IO.Objects.libsimpleio.syslog](#)

Assembly: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

◀ Syntax

C# VB F#

[Copy](#)

```
public const int LOG_PID = 1
```

Field Value

Type: [Int32](#)

◀ See Also

Reference

[Logger Class](#)

[IO.Objects.libsimpleio.syslog Namespace](#)

LoggerLOG_USER Field

User program facility. Use `LOG_LOCALx` instead.

Namespace: [IO.Objects.libsimpleio.syslog](#)

Assembly: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

◀ Syntax

C# VB F#

[Copy](#)

```
public const int LOG_USER = 8
```

Field Value

Type: [Int32](#)

◀ See Also

Reference

[Logger Class](#)

[IO.Objects.libsimpleio.syslog Namespace](#)

IO.Objects.libsimpleio.Watchdog Namespace

Watchdog Timer Services

↳ Classes

| Class | Description |
|---|--|
|  Timer | Encapsulates Linux watchdog timers using libsimpleio . |

Timer Class

Encapsulates Linux watchdog timers using [libsimpleio](#).

▪ Inheritance Hierarchy

[SystemObject](#) [IO.Objects.libsimpleio.WatchdogTimer](#)

Namespace: [IO.Objects.libsimpleio.Watchdog](#)

Assembly: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

▪ Syntax

C# VB F#

[Copy](#)

```
public class Timer : Timer
```

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

▪ Constructors

| | Name | Description |
|---|-----------------------|--|
|  | Timer | Constructor for a single watchdog timer. |

[Top](#)

▪ Properties

| | Name | Description |
|---|--------------------|--|
|  | fd | Read-only property returning the Linux file descriptor for the watchdog timer. |



`timeout` Get or set the watchdog timeout. Not all platforms may support this. Even if supported, there may be constraints. For example, some platforms allow shortening the timeout but not lengthening it.

[Top](#)

◀ Methods

| | Name | Description |
|---|----------------------|---------------------------|
| ≡ | Kick | Reset the watchdog timer. |

[Top](#)

◀ Fields

| | Name | Description |
|-----|--------------------------------|--|
| ❖ S | DefaultDevice | Default watchdog timer device name. |
| ❖ S | DefaultTimeout | Default watchdog timer timeout value (disabled). |

[Top](#)

◀ See Also

Reference

[IO.Objects.libsimpleio.Watchdog Namespace](#)

Timer Constructor

Constructor for a single watchdog timer.

Namespace: [IO.Objects.libsimpleio.Watchdog](#)

Assembly: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

◀ Syntax

C# VB F#

[Copy](#)

```
public Timer(  
    string devname = "/dev/watchdog",  
    int timeout = 0  
)
```

Parameters

devname (Optional)

Type: [SystemString](#)

Device node name.

timeout (Optional)

Type: [SystemInt32](#)

Watchdog timeout setting in seconds, or [DefaultTimeout](#).

◀ See Also

[Reference](#)

[Timer Class](#)

[IO.Objects.libsimpleio.Watchdog Namespace](#)

Timer Properties

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

Properties

| | Name | Description |
|--|----------------------|--|
| | <code>fd</code> | Read-only property returning the Linux file descriptor for the watchdog timer. |
| | <code>timeout</code> | Get or set the watchdog timeout. Not all platforms may support this. Even if supported, there may be constraints. For example, some platforms allow shortening the timeout but not lengthening it. |

[Top](#)

See Also

Reference

[Timer Class](#)

[IO.Objects.libsimpleio.Watchdog Namespace](#)

Timerfd Property

Read-only property returning the Linux file descriptor for the watchdog timer.

Namespace: [IO.Objects.libsimpleio.Watchdog](#)

Assembly: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

◀ Syntax

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

[Copy](#)

```
public int fd { get; }
```

Property Value

Type: [Int32](#)

◀ See Also

Reference

[Timer Class](#)

[IO.Objects.libsimpleio.Watchdog Namespace](#)

Timertimeout Property

Get or set the watchdog timeout. Not all platforms may support this. Even if supported, there may be constraints. For example, some platforms allow shortening the timeout but not lengthening it.

Namespace: [IO.Objects.libsimpleio.Watchdog](#)

Assembly: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

↳ Syntax

C# VB F#

[Copy](#)

```
public int timeout { get; set; }
```

Property Value

Type: [Int32](#)

Implements

[Timertimeout](#)

↳ See Also

Reference

[Timer Class](#)

[IO.Objects.libsimpleio.Watchdog Namespace](#)

Timer Methods

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

▪ Methods

| | Name | Description |
|---|----------------------|---------------------------|
|  | Kick | Reset the watchdog timer. |

[Top](#)

▪ See Also

Reference

[Timer Class](#)

[IO.Objects.libsimpleio.Watchdog Namespace](#)

TimerKick Method

Reset the watchdog timer.

Namespace: [IO.Objects.libsimpleio.Watchdog](#)

Assembly: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

« Syntax

C# VB F#

[Copy](#)

```
public void Kick()
```

Implements

[TimerKick](#)

« See Also

Reference

[Timer Class](#)

[IO.Objects.libsimpleio.Watchdog Namespace](#)

Timer Fields

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

Fields

| | Name | Description |
|---|--------------------------------|--|
| •  | DefaultDevice | Default watchdog timer device name. |
| •  | DefaultTimeout | Default watchdog timer timeout value (disabled). |

[Top](#)

See Also

Reference

[Timer Class](#)

[IO.Objects.libsimpleio.Watchdog Namespace](#)

TimerDefaultDevice Field

Default watchdog timer device name.

Namespace: [IO.Objects.libsimpleio.Watchdog](#)

Assembly: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

◀ Syntax

C# VB F#

[Copy](#)

```
public const string DefaultDevice = "/dev/watchdog"
```

Field Value

Type: [String](#)

◀ See Also

Reference

[Timer Class](#)

[IO.Objects.libsimpleio.Watchdog Namespace](#)

TimerDefaultTimeout Field

Default watchdog timer timeout value (disabled).

Namespace: [IO.Objects.libsimpleio.Watchdog](#)

Assembly: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

◀ Syntax

C# VB F#

[Copy](#)

```
public const int DefaultTimeout = 0
```

Field Value

Type: [Int32](#)

◀ See Also

Reference

[Timer Class](#)

[IO.Objects.libsimpleio.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: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

▪ 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: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

◀ 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: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

◀ 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: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

◀ 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: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

◀ 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: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

▪ 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: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

↳ 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: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

↳ 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: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

↳ 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: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

▪ 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: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

◀ 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: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

↳ 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](#)

Assembly: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

▪ 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: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

↳ 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: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

↳ 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.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: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

▪ 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: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

« 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](#)

IO.Remote Namespace

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: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

▪ 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: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

▪ 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: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

▪ 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: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

◀ 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: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

▪ 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: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

↳ 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: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

▪ 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: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

◀ 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: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

▀ 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: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

↳ 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: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

« 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: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

▪ 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: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

↳ 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: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

▪ 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: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

↳ 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: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

◀ 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: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

▪ 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: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

◀ 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: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

▪ 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: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

↳ 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: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

▪ 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: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

◀ 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: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

▪ 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: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

◀ 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: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

« 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: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

↳ 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: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

▪ 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: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

◀ 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: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

▪ 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.Remotel2C](#)

Namespace: [IO.Remote](#)

Assembly: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

▪ 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: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

▪ 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: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

↳ 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: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

↳ 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: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

↳ 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: `libsimpleio` (in `libsimpleio.dll`) Version: 2.2021.332.1

▪ 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: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

▪ 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: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

▪ 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: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

◀ 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: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

▪ 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: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

▪ 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: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

◀ 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: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

↳ 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](#)

SPI Transaction Method

Write and read bytes to and from an SPI slave device.

Namespace: [IO.Remote](#)

Assembly: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

↳ 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: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

↳ 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: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

▪ 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 SHIELDNAME 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: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

↳ 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: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

↳ 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: `libsimpleio` (in `libsimpleio.dll`) Version: 2.2021.332.1

▪ Syntax

C# VB F#

[Copy](#)

```
public enum Kinds
```

▪ Members

| Member name | Value | Description |
|-------------|-------|---|
| Clicker | 0 | Mikroelektronika STM32F4 Clicker with MUNTS-0011 Remote I/O Server firmware, with one mikroBUS socket. |
| PiClick1 | 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: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

▪ 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: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

↳ 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: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

« 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: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

« 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: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

↳ 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: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

↳ 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: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

« 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: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

« 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: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

↳ 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: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

« 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: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

◀ 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: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

◀ 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: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

« 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: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

« 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: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

« 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: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

« 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: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

« 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: libsimpleio (in libsimpleio.dll) Version: 2.2021.332.1

« Syntax

C# VB F#

[Copy](#)

```
public int TX { get; }
```

Property Value

Type: [Int32](#)

« See Also

Reference

[Socket Class](#)

[IO.Remote.mikroBUS Namespace](#)

Index

| | |
|----------------------------|----|
| IO.Bindings | 1 |
| libsimpleio Class | 2 |
| libsimpleio Methods | 18 |
| ADC_close Method | 26 |
| ADC_get_name Method | 27 |
| ADC_open Method | 29 |
| ADC_read Method | 31 |
| DAC_close Method | 33 |
| DAC_get_name Method | 34 |
| DAC_open Method | 36 |
| DAC_write Method | 38 |
| EVENT_close Method | 40 |
| EVENT_modify_fd Method | 41 |
| EVENT_open Method | 43 |
| EVENT_register_fd Method | 44 |
| EVENT_unregister_fd Method | 46 |
| EVENT_wait Method | 48 |
| GPIO_chip_info Method | 50 |
| GPIO_close Method | 52 |
| GPIO_configure Method | 53 |
| GPIO_line_close Method | 55 |
| GPIO_line_event Method | 57 |
| GPIO_line_info Method | 59 |
| GPIO_line_open Method | 61 |
| GPIO_line_read Method | 63 |
| GPIO_line_write Method | 65 |
| GPIO_open Method | 67 |

| | |
|------------------------------|-----|
| GPIO_read Method | 69 |
| GPIO_write Method | 71 |
| HIDRAW_close Method | 73 |
| HIDRAW_get_info Method | 75 |
| HIDRAW_get_name Method | 77 |
| HIDRAW_open1 Method | 79 |
| HIDRAW_open2 Method | 81 |
| HIDRAW_open3 Method | 83 |
| HIDRAW_receive Method | 85 |
| HIDRAW_send Method | 87 |
| I2C_close Method | 89 |
| I2C_open Method | 90 |
| I2C_transaction Method | 92 |
| IPV4_ntoa Method | 94 |
| IPV4_resolve Method | 96 |
| LINUX_command Method | 98 |
| LINUX_detach Method | 100 |
| LINUX_drop_privileges Method | 101 |
| LINUX_errno Method | 103 |
| LINUX_openlog Method | 104 |
| LINUX_poll Method | 106 |
| LINUX_strerror Method | 108 |
| LINUX_syslog Method | 110 |
| LINUX_usleep Method | 112 |
| PWM_close Method | 113 |
| PWM_configure Method | 114 |
| PWM_open Method | 116 |
| PWM_write Method | 118 |
| SERIAL_close Method | 120 |

| | |
|-----------------------------|-----|
| SERIAL_open Method | 121 |
| SERIAL_receive Method | 123 |
| SERIAL_send Method | 125 |
| SPI_close Method | 127 |
| SPI_open Method | 128 |
| SPI_transaction Method | 130 |
| STREAM_decode_frame Method | 132 |
| STREAM_encode_frame Method | 134 |
| STREAM_receive_frame Method | 136 |
| STREAM_send_frame Method | 138 |
| TCP4_accept Method | 140 |
| TCP4_close Method | 142 |
| TCP4_connect Method | 143 |
| TCP4_receive Method | 145 |
| TCP4_send Method | 147 |
| TCP4_server Method | 149 |
| UDP4_close Method | 151 |
| UDP4_open Method | 152 |
| UDP4_receive Method | 154 |
| UDP4_send Method | 156 |
| WATCHDOG_close Method | 158 |
| WATCHDOG_get_timeout Method | 160 |
| WATCHDOG_kick Method | 162 |
| WATCHDOG_open Method | 164 |
| WATCHDOG_set_timeout Method | 166 |
| libsimpleio Fields | 168 |
| GPIO_DIRECTION_INPUT Field | 177 |
| GPIO_DIRECTION_OUTPUT Field | 178 |
| GPIO_DRIVER_OPENDRAIN Field | 179 |

| | |
|-------------------------------------|-----|
| GPIO_DRIVER_OPENSOURCE Field | 180 |
| GPIO_DRIVER_PUSH_PULL Field | 181 |
| GPIO_EDGE_BOTH Field | 182 |
| GPIO_EDGE_FALLING Field | 183 |
| GPIO_EDGE_NONE Field | 184 |
| GPIO_EDGE_RISING Field | 185 |
| GPIO_EVENT_REQUEST_BOTH Field | 186 |
| GPIO_EVENT_REQUEST_FALLING Field | 187 |
| GPIO_EVENT_REQUEST_NONE Field | 188 |
| GPIO_EVENT_REQUEST_RISING Field | 189 |
| GPIO_LINE_INFO_ACTIVE_LOW Field | 190 |
| GPIO_LINE_INFO_KERNEL Field | 191 |
| GPIO_LINE_INFO_OPEN_DRAIN Field | 192 |
| GPIO_LINE_INFO_OPEN_SOURCE Field | 193 |
| GPIO_LINE_INFO_OUTPUT Field | 194 |
| GPIO_LINE_REQUEST_ACTIVE_HIGH Field | 195 |
| GPIO_LINE_REQUEST_ACTIVE_LOW Field | 196 |
| GPIO_LINE_REQUEST_INPUT Field | 197 |
| GPIO_LINE_REQUEST_OPEN_DRAIN Field | 198 |
| GPIO_LINE_REQUEST_OPEN_SOURCE Field | 199 |
| GPIO_LINE_REQUEST_OUTPUT Field | 200 |
| GPIO_LINE_REQUEST_PUSH_PULL Field | 201 |
| GPIO_POLARITY_ACTIVEHIGH Field | 202 |
| GPIO_POLARITY_ACTIVELOW Field | 203 |
| INADDR_ANY Field | 204 |
| INADDR_BROADCAST Field | 205 |
| INADDR_LOOPBACK Field | 206 |
| LOG_ALERT Field | 207 |
| LOG_AUTH Field | 208 |

| | |
|--------------------|-----|
| LOG_AUTHPRIV Field | 209 |
| LOG_CONS Field | 210 |
| LOG_CRIT Field | 211 |
| LOG_CRON Field | 212 |
| LOG_DAEMON Field | 213 |
| LOG_DEBUG Field | 214 |
| LOG_EMERG Field | 215 |
| LOG_ERR Field | 216 |
| LOG_FTP Field | 217 |
| LOG_INFO Field | 218 |
| LOG_KERN Field | 219 |
| LOG_LOCAL0 Field | 220 |
| LOG_LOCAL1 Field | 221 |
| LOG_LOCAL2 Field | 222 |
| LOG_LOCAL3 Field | 223 |
| LOG_LOCAL4 Field | 224 |
| LOG_LOCAL5 Field | 225 |
| LOG_LOCAL6 Field | 226 |
| LOG_LOCAL7 Field | 227 |
| LOG_LPR Field | 228 |
| LOG_MAIL Field | 229 |
| LOG_NDELAY Field | 230 |
| LOG_NEWS Field | 231 |
| LOG_NOTICE Field | 232 |
| LOG_NOWAIT Field | 233 |
| LOG_ODELAY Field | 234 |
| LOG_PERROR Field | 235 |
| LOG_PID Field | 236 |
| LOG_PROGNAME Field | 237 |

| | |
|-------------------------------|-----|
| LOG_SYSLOG Field | 238 |
| LOG_USER Field | 239 |
| LOG_UUCP Field | 240 |
| LOG_WARNING Field | 241 |
| MSG_DONTROUTE Field | 242 |
| MSG_DONTWAIT Field | 243 |
| MSG_MORE Field | 244 |
| POLLERR Field | 245 |
| POLLHUP Field | 246 |
| POLLIN Field | 247 |
| POLLNVAL Field | 248 |
| POLLOUT Field | 249 |
| POLLPRI Field | 250 |
| PWM_POLARITY_ACTIVEHIGH Field | 251 |
| PWM_POLARITY_ACTIVELOW Field | 252 |
| SERIAL_PARITY_EVEN Field | 253 |
| SERIAL_PARITY_NONE Field | 254 |
| SERIAL_PARITY_ODD Field | 255 |
| SPI_AUTO_CS Field | 256 |
| IO.Devices.A4988 | 257 |
| Device Class | 258 |
| Device Constructor | 260 |
| Device Properties | 262 |
| StepsPerRotation Property | 263 |
| Device Methods | 264 |
| Disable Method | 265 |
| Enable Method | 266 |
| Move Method | 267 |
| Reset Method | 269 |

| | |
|---------------------------|-----|
| Sleep Method | 270 |
| Spin Method | 271 |
| Wakeup Method | 273 |
| IO.Devices.AD5593R | 274 |
| Device Class | 275 |
| Device Constructor | 278 |
| Device Properties | 279 |
| ADC_Reference Property | 281 |
| DAC_Reference Property | 282 |
| GPIO_Inputs Property | 283 |
| GPIO_Outputs Property | 284 |
| Device Methods | 285 |
| ADC_Create Method | 286 |
| ConfigureChannel Method | 287 |
| DAC_Create Method | 288 |
| GPIO_Create Method | 290 |
| Read_ADC Method | 292 |
| Write_DAC Method | 293 |
| Device Fields | 294 |
| ADC_Resolution Field | 295 |
| DAC_Resolution Field | 296 |
| MaxChannel Field | 297 |
| MinChannel Field | 298 |
| PinMode Enumeration | 299 |
| ReferenceMode Enumeration | 301 |
| IO.Devices.AD5593R.ADC | 303 |
| Sample Class | 304 |
| Sample Constructor | 306 |
| Sample Properties | 307 |

| | |
|--------------------------------------|-----|
| resolution Property | 308 |
| sample Property | 309 |
| IO.Devices.AD5593R.DAC | 310 |
| Sample Class | 311 |
| Sample Constructor | 313 |
| Sample Properties | 315 |
| resolution Property | 316 |
| sample Property | 317 |
| IO.Devices.AD5593R.GPIO | 318 |
| Pin Class | 319 |
| Pin Constructor | 321 |
| Pin Properties | 323 |
| state Property | 324 |
| IO.Devices.ADC121C021 | 325 |
| Sample Class | 326 |
| Sample Constructor | 328 |
| Sample Properties | 329 |
| resolution Property | 330 |
| sample Property | 331 |
| IO.Devices.ClickBoards.RemoteIO.ADAC | 332 |
| Board Class | 333 |
| Board Constructor | 335 |
| Board Properties | 337 |
| device Property | 338 |
| Board Methods | 339 |
| ADC Method | 340 |
| DAC Method | 341 |
| GPIO Method | 343 |
| Reset Method | 345 |

| | |
|---|-----|
| Board Fields | 346 |
| DefaultAddress Field | 347 |
| IO.Devices.ClickBoards.RemoteIO.Expand | 348 |
| Board Class | 349 |
| Board Constructor | 351 |
| Board Properties | 352 |
| device Property | 353 |
| Board Methods | 354 |
| GPIO Method | 355 |
| Reset Method | 357 |
| IO.Devices.ClickBoards.RemoteIO.Expand2 | 358 |
| Board Class | 359 |
| Board Constructor | 361 |
| Board Properties | 363 |
| device Property | 364 |
| Board Methods | 365 |
| GPIO Method | 366 |
| Reset Method | 368 |
| Board Fields | 369 |
| DefaultAddress Field | 370 |
| IO.Devices.ClickBoards.RemoteIO.PWM | 371 |
| Board Class | 372 |
| Board Constructor | 374 |
| Board Properties | 376 |
| dev Property | 377 |
| Board Methods | 378 |
| GPIO Method | 379 |
| PWM Method | 381 |
| Servo Method | 383 |

| | |
|--|-----|
| Board Fields | 385 |
| DefaultAddress Field | 386 |
| IO.Devices.ClickBoards.RemoteIO.SevenSegment | 387 |
| Board Class | 388 |
| Board Constructor | 391 |
| Board Properties | 393 |
| blanking Property | 395 |
| brightness Property | 396 |
| leftdp Property | 397 |
| radix Property | 398 |
| rightdp Property | 399 |
| state Property | 400 |
| Board Methods | 401 |
| Clear Method | 402 |
| Reset Method | 403 |
| Board.Base Enumeration | 404 |
| Board.ZeroBlanking Enumeration | 405 |
| IO.Devices.ClickBoards.SimpleIO.ADAC | 406 |
| Board Class | 407 |
| Board Constructor | 409 |
| Board Properties | 410 |
| device Property | 411 |
| Board Methods | 412 |
| ADC Method | 413 |
| DAC Method | 414 |
| GPIO Method | 416 |
| Reset Method | 418 |
| Board Fields | 419 |
| DefaultAddress Field | 420 |

| | |
|---|-----|
| IO.Devices.ClickBoards.SimpleIO.Expand | 421 |
| Board Class | 422 |
| Board Constructor | 424 |
| Board Properties | 425 |
| device Property | 426 |
| Board Methods | 427 |
| GPIO Method | 428 |
| Reset Method | 430 |
| IO.Devices.ClickBoards.SimpleIO.Expand2 | 431 |
| Board Class | 432 |
| Board Constructor | 434 |
| Board Properties | 435 |
| device Property | 436 |
| Board Methods | 437 |
| GPIO Method | 438 |
| Reset Method | 440 |
| Board Fields | 441 |
| DefaultAddress Field | 442 |
| IO.Devices.ClickBoards.SimpleIO.PWM | 443 |
| Board Class | 444 |
| Board Constructor | 446 |
| Board Properties | 448 |
| dev Property | 449 |
| Board Methods | 450 |
| GPIO Method | 451 |
| PWM Method | 453 |
| Servo Method | 455 |
| Board Fields | 457 |
| DefaultAddress Field | 458 |

| | |
|--|-----|
| IO.Devices.ClickBoards.SimpleIO.SevenSegment | 459 |
| Board Class | 460 |
| Board Constructor | 463 |
| Board Properties | 465 |
| blanking Property | 467 |
| brightness Property | 468 |
| leftdp Property | 469 |
| radix Property | 470 |
| rightdp Property | 471 |
| state Property | 472 |
| Board Methods | 473 |
| Clear Method | 474 |
| Board.Base Enumeration | 475 |
| Board.ZeroBlanking Enumeration | 476 |
| IO.Devices.Grove.ADC | 477 |
| Device Class | 478 |
| Device Constructor | 480 |
| Device Properties | 481 |
| voltage Property | 482 |
| IO.Devices.Grove.Temperature | 483 |
| Device Class | 484 |
| Device Constructor | 486 |
| Device Properties | 487 |
| Celsius Property | 488 |
| Fahrenheit Property | 489 |
| Kelvins Property | 490 |
| IO.Devices.Grove.Temperature_Humidity | 491 |
| Device Class | 492 |
| Device Constructor | 494 |

| | |
|----------------------------|-----|
| Device Properties | 495 |
| IO.Devices.HDC1080 | 497 |
| Device Class | 498 |
| Device Constructor | 501 |
| Device Properties | 502 |
| Celsius Property | 504 |
| DeviceID Property | 505 |
| Fahrenheit Property | 506 |
| Humidity Property | 507 |
| Kelvins Property | 508 |
| ManufacturerID Property | 509 |
| Device Methods | 510 |
| Read Method | 511 |
| Write Method | 512 |
| Device Fields | 513 |
| RegConfiguration Field | 515 |
| RegDeviceID Field | 516 |
| RegHumidity Field | 517 |
| RegManufacturerID Field | 518 |
| RegSerialNumberFirst Field | 519 |
| RegSerialNumberLast Field | 520 |
| RegSerialNumberMid Field | 521 |
| RegTemperature Field | 522 |
| IO.Devices.MCP23017 | 523 |
| Device Class | 524 |
| Device Constructor | 527 |
| Device Properties | 528 |
| Direction Property | 530 |
| DirectionA Property | 531 |

| | |
|---------------------------------|-----|
| DirectionB Property | 532 |
| Polarity Property | 533 |
| PolarityA Property | 534 |
| PolarityB Property | 535 |
| Port Property | 536 |
| PortA Property | 537 |
| PortB Property | 538 |
| Pullups Property | 539 |
| PullupsA Property | 540 |
| PullupsB Property | 541 |
| Device Methods | 542 |
| GPIO_Create Method | 543 |
| Device Fields | 545 |
| MaxChannel Field | 546 |
| MinChannel Field | 547 |
| IO.Devices.MCP23017.GPIO | 548 |
| Pin Class | 549 |
| Pin Constructor | 551 |
| Pin Properties | 553 |
| state Property | 554 |
| IO.Devices.MCP23S17 | 555 |
| Device Class | 556 |
| Device Constructor | 560 |
| Device Properties | 561 |
| Direction Property | 563 |
| DirectionA Property | 564 |
| DirectionB Property | 565 |
| Polarity Property | 566 |
| PolarityA Property | 567 |

| | |
|--------------------------|-----|
| PolarityB Property | 568 |
| Port Property | 569 |
| PortA Property | 570 |
| PortB Property | 571 |
| Pullups Property | 572 |
| PullupsA Property | 573 |
| PullupsB Property | 574 |
| Device Methods | 575 |
| GPIO_Create Method | 576 |
| Device Fields | 578 |
| MaxChannel Field | 579 |
| MinChannel Field | 580 |
| SPI_Frequency Field | 581 |
| SPI_Mode Field | 582 |
| SPI_WordSize Field | 583 |
| IO.Devices.MCP23S17.GPIO | 584 |
| Pin Class | 585 |
| Pin Constructor | 587 |
| Pin Properties | 589 |
| state Property | 590 |
| IO.Devices.PCA8574 | 591 |
| Device Class | 592 |
| Device Constructor | 594 |
| Device Properties | 596 |
| Latch Property | 597 |
| Device Methods | 598 |
| Read Method | 599 |
| Write Method | 600 |
| Device Fields | 601 |

| | |
|---------------------------|-----|
| MAX_PINS Field | 602 |
| IO.Devices.PCA8574.GPIO | 603 |
| Pin Class | 604 |
| Pin Constructor | 606 |
| Pin Properties | 608 |
| state Property | 609 |
| IO.Devices.PCA9534 | 610 |
| Device Class | 611 |
| Device Constructor | 614 |
| Device Properties | 616 |
| Config Property | 617 |
| Latch Property | 618 |
| Device Methods | 619 |
| Read Method | 620 |
| Read Method | 621 |
| Read Method (Byte) | 622 |
| Write Method | 623 |
| Write Method (Byte) | 624 |
| Write Method (Byte, Byte) | 625 |
| Device Fields | 626 |
| AllInputs Field | 628 |
| AllNormal Field | 629 |
| AllOff Field | 630 |
| AllOutputs Field | 631 |
| ConfigurationReg Field | 632 |
| InputPolarityReg Field | 633 |
| InputPortReg Field | 634 |
| MAX_PINS Field | 635 |
| OutputPortReg Field | 636 |

| | |
|-------------------------|-----|
| IO.Devices.PCA9534.GPIO | 637 |
| Pin Class | 638 |
| Pin Constructor | 640 |
| Pin Properties | 642 |
| state Property | 643 |
| IO.Devices.PCA9685 | 644 |
| Device Class | 645 |
| Device Constructor | 648 |
| Device Properties | 650 |
| Frequency Property | 651 |
| Device Methods | 652 |
| ReadChannel Method | 653 |
| WriteChannel Method | 654 |
| Device Fields | 655 |
| INTERNAL_CLOCK Field | 656 |
| MAX_CHANNEL Field | 657 |
| MAX_CLOCK Field | 658 |
| MIN_CHANNEL Field | 659 |
| MIN_CLOCK Field | 660 |
| IO.Devices.PCA9685.GPIO | 661 |
| Pin Class | 662 |
| Pin Constructor | 664 |
| Pin Properties | 666 |
| state Property | 667 |
| IO.Devices.PCA9685.PWM | 668 |
| Output Class | 669 |
| Output Constructor | 671 |
| Output Properties | 673 |
| dutycycle Property | 674 |

| | |
|---------------------------|-----|
| IO.Devices.PCA9685.Servo | 675 |
| Output Class | 676 |
| Output Constructor | 678 |
| Output Properties | 680 |
| position Property | 681 |
| IO.Devices.Pmod.HYGRO | 682 |
| Device Class | 683 |
| Device Constructor | 686 |
| Device Properties | 687 |
| Device Methods | 689 |
| IO.Devices.SN74HC595 | 690 |
| Device Class | 691 |
| Device Constructor | 694 |
| Device Properties | 696 |
| Length Property | 697 |
| state Property | 698 |
| Device Methods | 699 |
| ClrBit Method | 700 |
| ReadBit Method | 701 |
| SetBit Method | 703 |
| Device Fields | 704 |
| SPI_MaxFreq Field | 705 |
| SPI_Mode Field | 706 |
| IO.Devices.SN74HC595.GPIO | 707 |
| Pin Class | 708 |
| Pin Constructor | 710 |
| Pin Properties | 712 |
| state Property | 713 |
| IO.Devices.TH02 | 714 |

| | |
|-----------------------|-----|
| Device Class | 715 |
| Device Constructor | 717 |
| Device Properties | 718 |
| Celsius Property | 719 |
| DeviceID Property | 720 |
| Fahrenheit Property | 721 |
| Humidity Property | 722 |
| Kelvins Property | 723 |
| IO.Devices.Thermistor | 724 |
| NTC_B Class | 725 |
| NTC_B Constructor | 727 |
| NTC_B Methods | 729 |
| Kelvins Method | 730 |
| IO.Devices.USB.Munts | 731 |
| HID Class | 732 |
| HID Fields | 734 |
| Product Field | 735 |
| Vendor Field | 736 |
| Serial Class | 737 |
| Serial Fields | 739 |
| Product Field | 740 |
| Vendor Field | 741 |
| IO.Interfaces.ADC | 742 |
| Input Class | 743 |
| Input Constructor | 745 |
| Input Properties | 747 |
| voltage Property | 748 |
| Sample Interface | 749 |
| Sample Properties | 750 |

| | |
|------------------------|-----|
| resolution Property | 751 |
| sample Property | 752 |
| Voltage Interface | 753 |
| Voltage Properties | 754 |
| voltage Property | 755 |
| IO.Interfaces.DAC | 756 |
| Output Class | 757 |
| Output Constructor | 759 |
| Output Properties | 761 |
| voltage Property | 762 |
| Sample Interface | 763 |
| Sample Properties | 765 |
| resolution Property | 766 |
| sample Property | 767 |
| Voltage Interface | 768 |
| Voltage Properties | 769 |
| voltage Property | 770 |
| IO.Interfaces.GPIO | 771 |
| Direction Enumeration | 772 |
| Pin Interface | 773 |
| Pin Properties | 774 |
| state Property | 775 |
| IO.Interfaces.Humidity | 776 |
| Sensor Interface | 777 |
| Sensor Properties | 778 |
| Humidity Property | 779 |
| IO.Interfaces.I2C | 780 |
| Bus Interface | 781 |
| Bus Methods | 783 |

| | |
|------------------------------|-----|
| Read Method | 784 |
| Transaction Method | 786 |
| Write Method | 788 |
| Device Class | 790 |
| Device Constructor | 792 |
| Device Methods | 793 |
| Read Method | 794 |
| Transaction Method | 795 |
| Write Method | 797 |
| Speeds Class | 798 |
| Speeds Fields | 800 |
| FastMode Field | 801 |
| FastModePlus Field | 802 |
| StandardMode Field | 803 |
| IO.Interfaces.Log | 804 |
| Logger Interface | 805 |
| Logger Methods | 807 |
| Error Method | 808 |
| Error Method (String) | 809 |
| Error Method (String, Int32) | 810 |
| Note Method | 812 |
| Warning Method | 813 |
| IO.Interfaces.Message.Text | 814 |
| Relay Interface | 815 |
| Relay Methods | 816 |
| Send Method | 817 |
| IO.Interfaces.Message64 | 819 |
| Message Class | 820 |
| Message Constructor | 822 |

| | |
|----------------------------|-----|
| Message Constructor | 823 |
| Message Constructor (Byte) | 824 |
| Message Fields | 825 |
| payload Field | 826 |
| Size Field | 827 |
| Messenger Interface | 828 |
| Messenger Methods | 830 |
| Receive Method | 831 |
| Send Method | 832 |
| Transaction Method | 833 |
| IO.Interfaces.Motor | 834 |
| Output Interface | 835 |
| Output Properties | 836 |
| velocity Property | 837 |
| Velocities Class | 838 |
| Velocities Fields | 840 |
| Maximum Field | 841 |
| Minimum Field | 842 |
| Stop Field | 843 |
| IO.Interfaces.PWM | 844 |
| DutyCycles Class | 845 |
| DutyCycles Fields | 847 |
| Maximum Field | 848 |
| Minimum Field | 849 |
| Output Interface | 850 |
| Output Properties | 851 |
| dutycycle Property | 852 |
| IO.Interfaces.Servo | 853 |
| Output Interface | 854 |

| | |
|----------------------------|-----|
| Output Properties | 855 |
| position Property | 856 |
| Positions Class | 857 |
| Positions Fields | 859 |
| Maximum Field | 860 |
| Minimum Field | 861 |
| Neutral Field | 862 |
| IO.Interfaces.SPI | 863 |
| Device Interface | 864 |
| Device Methods | 866 |
| Read Method | 867 |
| Transaction Method | 868 |
| Write Method | 870 |
| IO.Interfaces.Stepper | 871 |
| Output Interface | 872 |
| Output Properties | 874 |
| StepsPerRotation Property | 875 |
| Output Methods | 876 |
| Move Method | 877 |
| Spin Method | 879 |
| IO.Interfaces.Temperature | 881 |
| Conversions Class | 882 |
| Conversions Constructor | 884 |
| Conversions Methods | 885 |
| CelsiusToFahrenheit Method | 887 |
| CelsiusToKelvins Method | 888 |
| FahrenheitToCelsius Method | 889 |
| FahrenheitToKelvins Method | 890 |
| KelvinsToCelsius Method | 891 |

| | |
|--|-----|
| KelvinsToFahrenheit Method | 892 |
| Sensor Interface | 893 |
| Sensor Properties | 895 |
| Celsius Property | 896 |
| Fahrenheit Property | 897 |
| Kelvins Property | 898 |
| IO.Interfaces.Watchdog | 899 |
| Timer Interface | 900 |
| Timer Properties | 902 |
| timeout Property | 903 |
| Timer Methods | 904 |
| Kick Method | 905 |
| IO.Objects.Email.Mail | 906 |
| Relay Class | 907 |
| Relay Constructor | 909 |
| Relay Methods | 910 |
| Send Method | 911 |
| Send Method (String, String, String) | 912 |
| Send Method (String, String, String, String) | 914 |
| Send Method (String, String, String, String, String) | 916 |
| IO.Objects.Email.SMTP | 918 |
| Relay Class | 919 |
| Relay Constructor | 921 |
| Relay Methods | 922 |
| Send Method | 923 |
| Send Method (String, String, String) | 924 |
| Send Method (String, String, String, String) | 926 |
| Send Method (String, String, String, String, String) | 928 |
| IO.Objects.GPIO.PWM | 930 |

| | |
|-------------------------------|-----|
| Pin Class | 931 |
| Pin Constructor | 933 |
| Pin Properties | 935 |
| state Property | 936 |
| IO.Objects.libsimpleio.ADC | 937 |
| Sample Class | 938 |
| Sample Constructor | 940 |
| Sample Properties | 941 |
| fd Property | 942 |
| resolution Property | 943 |
| sample Property | 944 |
| Sample Methods | 945 |
| name Method | 946 |
| IO.Objects.libsimpleio.DAC | 947 |
| Sample Class | 948 |
| Sample Constructor | 950 |
| Sample Properties | 952 |
| fd Property | 953 |
| resolution Property | 954 |
| sample Property | 955 |
| Sample Methods | 956 |
| name Method | 957 |
| IO.Objects.libsimpleio.Device | 958 |
| Designator Structure | 959 |
| Designator Constructor | 961 |
| Designator Properties | 962 |
| available Property | 963 |
| Designator Fields | 964 |
| chan Field | 965 |

| | |
|---|------|
| chip Field | 966 |
| Unavailable Field | 967 |
| IO.Objects.libsimpleio.GPIO | 968 |
| Pin Class | 969 |
| Pin Constructor | 971 |
| Pin Properties | 973 |
| fd Property | 974 |
| state Property | 975 |
| Pin.Driver Enumeration | 976 |
| Pin.Edge Enumeration | 977 |
| Pin.Polarity Enumeration | 979 |
| IO.Objects.libsimpleio.HID | 980 |
| Messenger Class | 981 |
| Messenger Constructor | 984 |
| Messenger Constructor (String, Int32) | 985 |
| Messenger Constructor (Int32, Int32, String, Int32) | 987 |
| Messenger Properties | 989 |
| bustype Property | 990 |
| fd Property | 991 |
| name Property | 992 |
| product Property | 993 |
| vendor Property | 994 |
| Messenger Methods | 995 |
| Receive Method | 996 |
| Send Method | 997 |
| Transaction Method | 998 |
| IO.Objects.libsimpleio.I2C | 1000 |
| Bus Class | 1001 |
| Bus Constructor | 1003 |

| | |
|---------------------------------|------|
| Bus Constructor (Designator) | 1004 |
| Bus Constructor (String) | 1005 |
| Bus Properties | 1006 |
| fd Property | 1007 |
| Bus Methods | 1008 |
| Read Method | 1009 |
| Transaction Method | 1011 |
| Write Method | 1013 |
| IO.Objects.libsimpleio.mikroBUS | 1015 |
| Shield Class | 1016 |
| Shield Properties | 1018 |
| kind Property | 1019 |
| Shield Fields | 1020 |
| I2CBus Field | 1021 |
| Shield.Kinds Enumeration | 1022 |
| Socket Class | 1025 |
| Socket Constructor | 1028 |
| Socket Properties | 1029 |
| AIN Property | 1031 |
| AN Property | 1032 |
| CS Property | 1033 |
| I2CBus Property | 1034 |
| INT Property | 1035 |
| MISO Property | 1036 |
| MOSI Property | 1037 |
| PWM Property | 1038 |
| PWMOut Property | 1039 |
| RST Property | 1040 |
| RX Property | 1041 |

| | |
|----------------------------------|------|
| SCK Property | 1042 |
| SCL Property | 1043 |
| SDA Property | 1044 |
| SPIDev Property | 1045 |
| TX Property | 1046 |
| UART Property | 1047 |
| IO.Objects.libsimpleio.Platforms | 1048 |
| BeagleBone Class | 1049 |
| BeagleBone Fields | 1055 |
| AIN0 Field | 1060 |
| AIN1 Field | 1061 |
| AIN2 Field | 1062 |
| AIN3 Field | 1063 |
| AIN4 Field | 1064 |
| AIN5 Field | 1065 |
| AIN6 Field | 1066 |
| GPIO10 Field | 1067 |
| GPIO11 Field | 1068 |
| GPIO110 Field | 1069 |
| GPIO111 Field | 1070 |
| GPIO112 Field | 1071 |
| GPIO113 Field | 1072 |
| GPIO115 Field | 1073 |
| GPIO117 Field | 1074 |
| GPIO12 Field | 1075 |
| GPIO13 Field | 1076 |
| GPIO14 Field | 1077 |
| GPIO15 Field | 1078 |
| GPIO2 Field | 1079 |

| | |
|--------------|------|
| GPIO20 Field | 1080 |
| GPIO22 Field | 1081 |
| GPIO23 Field | 1082 |
| GPIO26 Field | 1083 |
| GPIO27 Field | 1084 |
| GPIO3 Field | 1085 |
| GPIO30 Field | 1086 |
| GPIO31 Field | 1087 |
| GPIO32 Field | 1088 |
| GPIO33 Field | 1089 |
| GPIO34 Field | 1090 |
| GPIO35 Field | 1091 |
| GPIO36 Field | 1092 |
| GPIO37 Field | 1093 |
| GPIO38 Field | 1094 |
| GPIO39 Field | 1095 |
| GPIO4 Field | 1096 |
| GPIO44 Field | 1097 |
| GPIO45 Field | 1098 |
| GPIO46 Field | 1099 |
| GPIO47 Field | 1100 |
| GPIO48 Field | 1101 |
| GPIO49 Field | 1102 |
| GPIO5 Field | 1103 |
| GPIO50 Field | 1104 |
| GPIO51 Field | 1105 |
| GPIO60 Field | 1106 |
| GPIO61 Field | 1107 |
| GPIO62 Field | 1108 |

| | |
|--------------------|------|
| GPIO63 Field | 1109 |
| GPIO65 Field | 1110 |
| GPIO66 Field | 1111 |
| GPIO67 Field | 1112 |
| GPIO68 Field | 1113 |
| GPIO69 Field | 1114 |
| GPIO7 Field | 1115 |
| GPIO70 Field | 1116 |
| GPIO71 Field | 1117 |
| GPIO72 Field | 1118 |
| GPIO73 Field | 1119 |
| GPIO74 Field | 1120 |
| GPIO75 Field | 1121 |
| GPIO76 Field | 1122 |
| GPIO77 Field | 1123 |
| GPIO78 Field | 1124 |
| GPIO79 Field | 1125 |
| GPIO8 Field | 1126 |
| GPIO80 Field | 1127 |
| GPIO81 Field | 1128 |
| GPIO86 Field | 1129 |
| GPIO87 Field | 1130 |
| GPIO88 Field | 1131 |
| GPIO89 Field | 1132 |
| GPIO9 Field | 1133 |
| I2C2 Field | 1134 |
| SPI2_0 Field | 1135 |
| SPI2_1 Field | 1136 |
| PocketBeagle Class | 1137 |

| | |
|---------------------|------|
| PocketBeagle Fields | 1142 |
| AIN0 Field | 1146 |
| AIN1 Field | 1147 |
| AIN2 Field | 1148 |
| AIN3 Field | 1149 |
| AIN4 Field | 1150 |
| AIN5 Field | 1151 |
| AIN6 Field | 1152 |
| AIN7 Field | 1153 |
| GPIO110 Field | 1154 |
| GPIO111 Field | 1155 |
| GPIO112 Field | 1156 |
| GPIO113 Field | 1157 |
| GPIO114 Field | 1158 |
| GPIO115 Field | 1159 |
| GPIO116 Field | 1160 |
| GPIO117 Field | 1161 |
| GPIO12 Field | 1162 |
| GPIO13 Field | 1163 |
| GPIO14 Field | 1164 |
| GPIO15 Field | 1165 |
| GPIO19 Field | 1166 |
| GPIO2 Field | 1167 |
| GPIO20 Field | 1168 |
| GPIO23 Field | 1169 |
| GPIO26 Field | 1170 |
| GPIO27 Field | 1171 |
| GPIO3 Field | 1172 |
| GPIO30 Field | 1173 |

| | |
|--------------|------|
| GPIO31 Field | 1174 |
| GPIO4 Field | 1175 |
| GPIO40 Field | 1176 |
| GPIO41 Field | 1177 |
| GPIO42 Field | 1178 |
| GPIO43 Field | 1179 |
| GPIO44 Field | 1180 |
| GPIO45 Field | 1181 |
| GPIO46 Field | 1182 |
| GPIO47 Field | 1183 |
| GPIO5 Field | 1184 |
| GPIO50 Field | 1185 |
| GPIO52 Field | 1186 |
| GPIO57 Field | 1187 |
| GPIO58 Field | 1188 |
| GPIO59 Field | 1189 |
| GPIO60 Field | 1190 |
| GPIO64 Field | 1191 |
| GPIO65 Field | 1192 |
| GPIO7 Field | 1193 |
| GPIO86 Field | 1194 |
| GPIO87 Field | 1195 |
| GPIO88 Field | 1196 |
| GPIO89 Field | 1197 |
| I2C1 Field | 1198 |
| I2C2 Field | 1199 |
| PWM0_0 Field | 1200 |
| PWM2_0 Field | 1201 |
| SPI0_0 Field | 1202 |

| | |
|--------------------|------|
| SPI1_1 Field | 1203 |
| RaspberryPi Class | 1204 |
| RaspberryPi Fields | 1207 |
| AIN0 Field | 1210 |
| AIN1 Field | 1211 |
| GPIO10 Field | 1212 |
| GPIO11 Field | 1213 |
| GPIO12 Field | 1214 |
| GPIO13 Field | 1215 |
| GPIO14 Field | 1216 |
| GPIO15 Field | 1217 |
| GPIO16 Field | 1218 |
| GPIO17 Field | 1219 |
| GPIO18 Field | 1220 |
| GPIO19 Field | 1221 |
| GPIO2 Field | 1222 |
| GPIO20 Field | 1223 |
| GPIO21 Field | 1224 |
| GPIO22 Field | 1225 |
| GPIO23 Field | 1226 |
| GPIO24 Field | 1227 |
| GPIO25 Field | 1228 |
| GPIO26 Field | 1229 |
| GPIO27 Field | 1230 |
| GPIO3 Field | 1231 |
| GPIO4 Field | 1232 |
| GPIO5 Field | 1233 |
| GPIO6 Field | 1234 |
| GPIO7 Field | 1235 |

| | |
|---|------|
| GPIO8 Field | 1236 |
| GPIO9 Field | 1237 |
| I2C1 Field | 1238 |
| PWM0_0 Field | 1239 |
| PWM0_1 Field | 1240 |
| SPI0_0 Field | 1241 |
| SPI0_1 Field | 1242 |
| IO.Objects.libsimpleio.PWM | 1243 |
| Output Class | 1244 |
| Output Constructor | 1246 |
| Output Properties | 1248 |
| dutycycle Property | 1249 |
| fd Property | 1250 |
| IO.Objects.libsimpleio.Servo | 1251 |
| Output Class | 1252 |
| Output Constructor | 1254 |
| Output Properties | 1256 |
| fd Property | 1257 |
| position Property | 1258 |
| IO.Objects.libsimpleio.SPI | 1259 |
| Device Class | 1260 |
| Device Constructor | 1262 |
| Device Constructor (Designator, Int32, Int32, Int32, Pin) | 1263 |
| Device Constructor (String, Int32, Int32, Int32, Pin) | 1265 |
| Device Properties | 1267 |
| fd Property | 1268 |
| Device Methods | 1269 |
| Read Method | 1270 |
| Transaction Method | 1272 |

| | |
|-------------------------------|------|
| Write Method | 1274 |
| Device Fields | 1276 |
| AUTOCHIPSELECT Field | 1277 |
| IO.Objects.libsimpleio.syslog | 1278 |
| Logger Class | 1279 |
| Logger Constructor | 1282 |
| Logger Methods | 1284 |
| Error Method | 1285 |
| Error Method (String) | 1286 |
| Error Method (String, Int32) | 1287 |
| Note Method | 1289 |
| Warning Method | 1290 |
| Logger Fields | 1291 |
| LOG_AUTH Field | 1293 |
| LOG_DAEMON Field | 1294 |
| LOG_LOCAL0 Field | 1295 |
| LOG_LOCAL1 Field | 1296 |
| LOG_LOCAL2 Field | 1297 |
| LOG_LOCAL3 Field | 1298 |
| LOG_LOCAL4 Field | 1299 |
| LOG_LOCAL5 Field | 1300 |
| LOG_LOCAL6 Field | 1301 |
| LOG_LOCAL7 Field | 1302 |
| LOG_MAIL Field | 1303 |
| LOG_NDELAY Field | 1304 |
| LOG_ODELAY Field | 1305 |
| LOG_PERROR Field | 1306 |
| LOG_PID Field | 1307 |
| LOG_USER Field | 1308 |

| | |
|---|------|
| IO.Objects.libsimpleio.Watchdog | 1309 |
| Timer Class | 1310 |
| Timer Constructor | 1312 |
| Timer Properties | 1313 |
| fd Property | 1314 |
| timeout Property | 1315 |
| Timer Methods | 1316 |
| Kick Method | 1317 |
| Timer Fields | 1318 |
| DefaultDevice Field | 1319 |
| DefaultTimeout Field | 1320 |
| IO.Objects.Message64.UDP | 1321 |
| Messenger Class | 1322 |
| Messenger Constructor | 1324 |
| Messenger Methods | 1326 |
| Receive Method | 1327 |
| Send Method | 1328 |
| Transaction Method | 1329 |
| IO.Objects.Motor.PWM | 1331 |
| Output Class | 1332 |
| Output Constructor | 1334 |
| Output Constructor (Output, Output, Double) | 1335 |
| Output Constructor (Pin, Output, Double) | 1337 |
| Output Properties | 1339 |
| velocity Property | 1340 |
| IO.Objects.Motor.Servo | 1341 |
| Output Class | 1342 |
| Output Constructor | 1344 |
| Output Properties | 1345 |

| | |
|--------------------------------|------|
| velocity Property | 1346 |
| IO.Objects.Servo.PWM | 1347 |
| Output Class | 1348 |
| Output Constructor | 1350 |
| Output Properties | 1352 |
| position Property | 1353 |
| IO.Remote | 1354 |
| ADC Class | 1356 |
| ADC Constructor | 1358 |
| ADC Properties | 1360 |
| resolution Property | 1361 |
| sample Property | 1362 |
| DAC Class | 1363 |
| DAC Constructor | 1365 |
| DAC Properties | 1367 |
| resolution Property | 1368 |
| sample Property | 1369 |
| Device Class | 1370 |
| Device Constructor | 1373 |
| Device Constructor | 1374 |
| Device Constructor (Messenger) | 1375 |
| Device Properties | 1376 |
| Capabilities Property | 1377 |
| Version Property | 1378 |
| Device Methods | 1379 |
| ADC_Available Method | 1381 |
| ADC_Create Method | 1382 |
| DAC_Available Method | 1383 |
| DAC_Create Method | 1384 |

| | |
|-----------------------------|------|
| Dispatcher Method | 1386 |
| GPIO_Available Method | 1387 |
| GPIO_Create Method | 1388 |
| I2C_Available Method | 1390 |
| I2C_Create Method | 1391 |
| PWM_Available Method | 1393 |
| PWM_Create Method | 1394 |
| SPI_Available Method | 1396 |
| SPI_Create Method | 1397 |
| Device Fields | 1399 |
| MAX_CHANNELS Field | 1400 |
| Unavailable Field | 1401 |
| GPIO Class | 1402 |
| GPIO Constructor | 1404 |
| GPIO Properties | 1406 |
| state Property | 1407 |
| I2C Class | 1408 |
| I2C Constructor | 1410 |
| I2C Methods | 1412 |
| Read Method | 1413 |
| Transaction Method | 1415 |
| Write Method | 1417 |
| MessageTypes Enumeration | 1419 |
| PeripheralTypes Enumeration | 1425 |
| PWM Class | 1427 |
| PWM Constructor | 1429 |
| PWM Properties | 1431 |
| dutycycle Property | 1432 |
| SPI Class | 1433 |

| | |
|--------------------------|------|
| SPI Constructor | 1435 |
| SPI Methods | 1437 |
| Read Method | 1438 |
| Transaction Method | 1440 |
| Write Method | 1442 |
| IO.Remote.mikroBUS | 1444 |
| Shield Class | 1445 |
| Shield Properties | 1447 |
| kind Property | 1448 |
| Shield Fields | 1449 |
| I2CBus Field | 1450 |
| Shield.Kinds Enumeration | 1451 |
| Socket Class | 1453 |
| Socket Constructor | 1456 |
| Socket Properties | 1457 |
| AIN Property | 1459 |
| AN Property | 1460 |
| CS Property | 1461 |
| I2CBus Property | 1462 |
| INT Property | 1463 |
| MISO Property | 1464 |
| MOSI Property | 1465 |
| PWM Property | 1466 |
| PWMOut Property | 1467 |
| RST Property | 1468 |
| RX Property | 1469 |
| SCK Property | 1470 |
| SCL Property | 1471 |
| SDA Property | 1472 |

SPIDev Property
TX Property

1473
1474