Simulink® Desktop Real-Time™

Reference

MATLAB®

MathWorks®

How to Contact MathWorks

	Latest news:	www.mathworks.com
	Sales and services:	www.mathworks.com/sales_and_services
	User community:	www.mathworks.com/matlabcentral
	Technical support:	www.mathworks.com/support/contact_us
	Phone:	508-647-7000

The MathWorks, Inc.
1 Apple Hill Drive
Natick, MA 01760-2098

Simulink® Desktop Real-Time™ Reference

Revision History

Contents

Functions

rtwho

Display information about development computer status

Syntax

```
rtwho
```

Description

`rtwho` displays information about the Simulink Desktop Real-Time development computer status. For example:

- Simulink Desktop Real-Time version
- Computer type
- Timers (ID, period, and status)
- Hardware drivers (ID, device type, and address)

Examples

Get Development Computer Information

Get information about a development computer with a HUMUSOFT® MF624 board installed.

```
rtwho
```

```
Simulink Desktop Real-Time version 5.0.0 (C) The MathWorks, Inc.
1994-2014 Running on 64-bit computer.

TIMERS:  Period        Running
         0.001         Yes
         0.001         Yes

DRIVERS: Address       Name
         0             Humusoft MF624
```

See Also
`sldrtkernel`

Topics
"Install Real-Time Kernel"

Introduced before R2006a

sldrtconfigset

Attach and activate default Simulink Desktop Real-Time configuration set

Syntax

```
sldrtconfigset model
sldrtconfigset model 'ERT'

configset = sldrtconfigset
configset = sldrtconfigset('','ERT')
```

Description

`sldrtconfigset model` attaches the default Simulink Desktop Real-Time configuration set `SimulinkDesktopRealTime` to the model, and then activates the configuration set. This set specifies various simulation and code generation parameter values that are useful when working with a Simulink Desktop Real-Time model.

`sldrtconfigset model 'ERT'` attaches the default Simulink Desktop Real-Time configuration set `SimulinkDesktopRealTime_ERT` for Embedded Coder® software to the model, and then activates the configuration set. This set specifies simulation and code generation parameter values that are useful when working with a Simulink Desktop Real-Time model in the Embedded Coder environment.

`configset = sldrtconfigset` returns the default Simulink Desktop Real-Time configuration set object. Later, you can attach this configuration set to a model.

`configset = sldrtconfigset('','ERT')` returns the default Simulink Desktop Real-Time configuration set object for the Embedded Coder software. Later, you can attach this configuration set to a model.

Examples

Set and Get Default Configuration Set

Set model `sldrtex_vdp` to the default configuration set. Read the configuration set into a variable.

```
open_system sldrtex_vdp
sldrtconfigset sldrtex_vdp

configset = sldrtconfigset;
configset.Name

ans =

SimulinkDesktopRealTime
```

Set and Get Default ERT Configuration Set

Open model `sldrtex_vdp` and set `gcs` to the default Embedded Coder configuration set. Read the configuration set into a variable.

```
open_system('sldrtex_vdp')
sldrtconfigset(gcs, 'ERT')

configset = sldrtconfigset('','ERT');
configset.Name


ans =

SimulinkDesktopRealTime_ERT
```

Input Arguments

`model` — Model to associate with default configuration set
character vector

The model to which you attach the default Simulink Desktop Real-Time configuration set.

Example: `sldrtex_vdp`

Output Arguments

`configset` — Default configuration set
structure

The default Simulink Desktop Real-Time configuration set, which later you can attach to a model.

See Also

Topics
"Configure a Model for Simulink Desktop Real-Time"
"Manage Configuration Sets for a Model"
"Share a Configuration with Multiple Models"

Introduced in R2015a

sldrtkernel

Install and remove Simulink Desktop Real-Time kernel

Syntax

```
sldrtkernel -install
sldrtkernel -uninstall
sldrtkernel -version
sldrtkernel -setup
```

Description

sldrtkernel -install installs the Simulink Desktop Real-Time kernel on your system.

On Windows® systems, installing or uninstalling the kernel can require you to authorize requests from User Account Control (UAC). On Mac OS systems, installing or uninstalling the kernel can require you to enter an osascript password.

sldrtkernel -uninstall removes the Simulink Desktop Real-Time kernel from your system.

sldrtkernel -version displays the Simulink Desktop Real-Time version installed on your system.

sldrtkernel -setup installs the Simulink Desktop Real-Time kernel on your system.

Examples

Update Kernel Version

Checking the version of the kernel, uninstalling the kernel, and installing a new version

Check the kernel version.

```
sldrtkernel -version
```

```
The installed version of the Simulink Desktop Real-Time kernel
is 5.0.0.
```

Uninstall the kernel. At the command prompt, type y.

```
sldrtkernel -uninstall
```

```
You are going to uninstall the Simulink Desktop Real-Time kernel.
Do you want to proceed? [y] : y
The Simulink Desktop Real-Time kernel has been successfully
uninstalled.
```

Reinstall the kernel. At the command prompt, type y.

```
sldrtkernel -install
```

```
You are going to install the Simulink Desktop Real-Time kernel.
Do you want to proceed? [y] : y
The Simulink Desktop Real-Time kernel has been successfully
installed.
```

See Also
rtwho

Topics
"Install Real-Time Kernel"

Introduced in R2015a

Blocks

Analog Input

Read from one or multiple analog input channels
Library: Simulink Desktop Real-Time
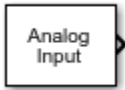
Description

Connect to and read from specific analog input channels from your Simulink model. After you add an Analog Input block to your model, you can enter the parameters for its I/O driver.

Ports

Output

`Analog Data` — **Analog-to-digital converter data read from input channels**
vector

Result of A/D conversion of an analog signal. The data type and range of the output value depend on the capabilities of the board.

Data Types: double | single | int8 | uint8 | int16 | uint16 | int32 | uint32

`Missed Ticks` — **Missed tick count (optional)**
double

In **Connected IO** mode, returns the number of timer ticks that your model lags behind the real-time kernel. When the model lags by more than **Maximum missed ticks**, the software reports an error and simulation stops.

Dependency

When you select **Show "Missed Ticks" port**, this port is visible.

Data Types: double

Parameters

`Install new board` — **Register a board**
`< no board selected >` (default)

When you click **Install new board**, the software displays a list of manufacturers of supported boards. When you select a manufacturer, the software displays a list of boards available from that manufacturer. When you select a board, the software adds the board to the list of registered boards and makes that board the current board.

By default, the initial selection in the list of registered boards is `< no board selected >`.

Dependency

When you select a board in the list of registered boards, the **Delete current board** and **Board setup** buttons are available.

Programmatic Use
Block Parameter: DrvName

Delete current board — Delete the current board
(button)

To delete the current board, click this button. The initial selection of the list of registered boards changes to < no board selected >.

Dependency

To activate this parameter, select a board in the list of registered boards.

When you delete the current board, the **Delete current board** and **Board setup** buttons are no longer available.

Programmatic Use
Block Parameter: DrvName

Board setup — Set up the board
(button)

To set up the board, click this button.

A board-specific dialog box opens to set up the board. For more information, see the board manufacturer documentation.

Dependency

To activate this parameter, select a board in the list of registered boards.

To deactivate this parameter, click **Delete current board**.

Programmatic Use
Block Parameter: DrvAddress
Block Parameter: DrvOptions

Sample time — Sample time of block
0.1 (default) | double

Enter a value, in seconds, that represents how frequently you want the block to execute and interact with the I/O hardware. The block synchronizes your model with the real-time clock at this sample rate.

If you are using a fixed-step solver, enter the value that you entered as the **Fixed step size** configuration parameter or an integer multiple of that value.

Programmatic Use
Block Parameter: SampleTime

Maximum missed ticks — Number of timer ticks by which your model can trail the kernel
10 (default) | integer

In **Connected IO** mode, enter the number of timer ticks that your model can lag behind the real-time kernel. When the model lags behind by this number or fewer timer ticks, the software assumes that the lag is temporary. It allows the model to catch up, even if the model misses some ticks. When the model lags by more than this number, the software reports an error and simulation stops.

In **Run in Kernel** mode, the software ignores this value.

Programmatic Use
Block Parameter: `MaxMissedTicks`

Show "Missed Ticks" port — Send number of missed ticks to port `Missed Ticks`
`'off'` (default) | `'on'`

In **Connected IO** mode, select this check box to send the number of missed ticks to output port `Missed Ticks`.

In **Run in Kernel** mode, the `Missed Ticks` port is zero.

Programmatic Use
Block Parameter: `ShowMissedTicks`

`Yield CPU when waiting` — Yield CPU while kernel waits for hardware
`'off'` (default) | `'on'`

In **Connected IO** mode, select this check box to grant other programs more CPU time while the kernel waits for a response from the hardware.

In **Run in Kernel** mode, the software ignores this value.

Programmatic Use
Block Parameter: `YieldWhenWaiting`

`Input channels` — Select analog input channels
1 (default) | vector

Enter a channel vector that selects the analog input channels that you are using on this board. The vector can be any valid MATLAB® vector form. For example, to select eight contiguous analog input channels, enter:

`[1,2,3,4,5,6,7,8]`

or

`[1:8]`

To select the first three analog input channels, enter:

`[1,2,3]`

Programmatic Use
Block Parameter: `Channels`

`Input range` — Input range of analog inputs
board dependent

From the list, select the input range for the analog input channels that you entered for the **Input channels** parameter. For example, if the board has an input range of ±5 V, select `-5 to 5 V`.

If you want a different input range for different analog channels, add an I/O block for each different input range. The valid values depend upon the capabilities of the board.

Programmatic Use
Block Parameter: VoltRange

Block output signal — Value type returned from block
Volts (default) | Normalized bipolar | Normalized unipolar | Raw

From the list, select from the following options:

- Volts — Returns a value equal to the analog voltage.
- Normalized bipolar — Returns a full range value of -1 to +1 regardless of the input voltage range.
- Normalized unipolar — Returns a full range value of 0 to +1 regardless of the input voltage range. For example, analog input ranges of 0 volts to +5 volts and of −5 volts to +5 volts are converted to 0 to +1.
- Raw — Returns a value from 0 through 2^n -1. For example, a 12-bit A/D converter returns values from 0 through 2^{12} -1 (0 to 4095). The advantage of this method is that the returned value is an integer with no round-off errors.

Programmatic Use
Block Parameter: RangeMode

Output data type — Data type of block output
double (default) | single | int8 | uint8 | int16 | uint16 | int32 | uint32

From the list, select the type of data that the block outputs to the model. The valid values depend on the capabilities of the board.

Programmatic Use
Block Parameter: DataType

See Also
Analog Output

Topics
"Use I/O Driver Blocks"

Introduced before R2006a

Analog Output

Write to one or multiple analog output channels
Library: Simulink Desktop Real-Time

Description

Connect to and write to specific analog output channels from your Simulink model. After you add an Analog Output block to your model, you can enter the parameters for its I/O driver.

Ports

Input

Analog Data — Digital-to-analog converter data to write to output channels
vector

Source data provided as input to D/A conversion. The data type and range of the input value depend upon the capabilities of the board.

Data Types: `single` | `double` | `int8` | `uint8` | `int16` | `uint16` | `int32` | `uint32`

Output

Missed Ticks — Missed tick count (optional)
double

In **Connected IO** mode, returns the number of timer ticks that your model lags behind the real-time kernel. When the model lags by more than **Maximum missed ticks**, the software reports an error and simulation stops.

Dependency

When you select **Show "Missed Ticks" port**, this port is visible.

Data Types: `double`

Parameters

Install new board — Register a board
`< no board selected >` (default)

When you click **Install new board**, the software displays a list of manufacturers of supported boards. When you select a manufacturer, the software displays a list of boards available from that manufacturer. When you select a board, the software adds the board to the list of registered boards and makes that board the current board.

By default, the initial selection in the list of registered boards is `< no board selected >`.

Dependency

When you select a board in the list of registered boards, the **Delete current board** and **Board setup** buttons are available.

Programmatic Use
Block Parameter: DrvName

Delete current board — Delete the current board
(button)

To delete the current board, click this button. The initial selection of the list of registered boards changes to < no board selected >.

Dependency

To activate this parameter, select a board in the list of registered boards.

When you delete the current board, the **Delete current board** and **Board setup** buttons are no longer available.

Programmatic Use
Block Parameter: DrvName

Board setup — Set up the board
(button)

To set up the board, click this button.

A board-specific dialog box opens to set up the board. For more information, see the board manufacturer documentation.

Dependency

To activate this parameter, select a board in the list of registered boards.

To deactivate this parameter, click **Delete current board**.

Programmatic Use
Block Parameter: DrvAddress
Block Parameter: DrvOptions

Sample time — Sample time of block
0.1 (default) | double

Enter a value, in seconds, that represents how frequently you want the block to execute and interact with the I/O hardware. The block synchronizes your model with the real-time clock at this sample rate.

If you are using a fixed-step solver, enter the value that you entered as the **Fixed step size** configuration parameter or an integer multiple of that value.

Programmatic Use
Block Parameter: SampleTime

Maximum missed ticks — Number of timer ticks by which your model can trail the kernel
10 (default) | integer

In **Connected IO** mode, enter the number of timer ticks that your model can lag behind the real-time kernel. When the model lags behind by this number or fewer timer ticks, the software assumes that the lag is temporary. It allows the model to catch up, even if the model misses some ticks. When the model lags by more than this number, the software reports an error and simulation stops.

In **Run in Kernel** mode, the software ignores this value.

Programmatic Use
Block Parameter: MaxMissedTicks

Show "Missed Ticks" port — Send number of missed ticks to port Missed Ticks
'off' (default) | 'on'

In **Connected IO** mode, select this check box to send the number of missed ticks to output port Missed Ticks.

In **Run in Kernel** mode, the Missed Ticks port is zero.

Programmatic Use
Block Parameter: ShowMissedTicks

Yield CPU when waiting — Yield CPU while kernel waits for hardware
'off' (default) | 'on'

In **Connected IO** mode, select this check box to grant other programs more CPU time while the kernel waits for a response from the hardware.

In **Run in Kernel** mode, the software ignores this value.

Programmatic Use
Block Parameter: YieldWhenWaiting

Output channels — Channel vector that selects output channels
1 (default) | vector

Enter a channel vector that selects the analog output channels that you are using on this board. The vector can be any valid MATLAB vector form. For example, to select the first two analog output channels, enter:

[1,2]

or

[1:2]

Programmatic Use
Block Parameter: Channels

Output range — Select output range for analog outputs
board dependent

From the list, select the output range for the analog output channels that you entered for the **Output channels** parameter. For example, if the board has an output range of ±5 V, select -5 to 5 V.

If you want the output range to be different for different analog channels, add an I/O block for each different output range.

Programmatic Use
Block Parameter: VoltRange

Block input signal — Value type sent into block
Volts (default) | Normalized bipolar | Normalized unipolar | Raw

From the list, select from the following options:

- Volts — Expects a value equal to the analog output voltage.
- Normalized bipolar — Expects a value between -1 and +1 that is converted to the full range of the output voltage regardless of the output voltage range.
- Normalized unipolar — Expects a value between 0 and +1 that is converted to the full range of the output voltage regardless of the output voltage range. For example, analog output ranges of 0 volts to +5 volts and of −5 volts to +5 volts are converted from values between 0 and +1.
- Raw — Expects a value from 0 through 2^n -1. For example, a 12-bit A/D converter expects a value between 0 and 2^{12} -1 (0 to 4095). The advantage of this method is that the expected value is an integer with no round-off errors.

Programmatic Use
Block Parameter: RangeMode

Initial value — Initial value for output channels
vector

Enter the initial value for each analog output channel that you entered for the **Output channels** parameter. For example, if you entered [1,2], and you want initial values of 0 Volts, enter [0,0].

Programmatic Use
Block Parameter: InitialValue

Final value — Final value for output channels
vector

Enter a final value for each analog channel that you entered for the **Output channels** parameter. For example, if you entered [1,2], and you want final values of 0 Volts, enter [0,0].

Programmatic Use
Block Parameter: FinalValue

See Also
Analog Input

Topics
"Use I/O Driver Blocks"

Introduced before R2006a

Counter Input

Read from one or multiple counter-input channels
Library: Simulink Desktop Real-Time
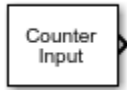
Description

Connect to and read from specific counter-input channels into your Simulink model. After you have added a Counter Input block to your model, you can enter the parameters for its I/O driver.

Ports

Output

Counter Data — Counter data read from input channels
vector

Data representing the state of a counter. The data type of the output values depends upon the capabilities of the board.

Data Types: single | double | int8 | uint8 | int16 | uint16 | int32 | uint32

Missed Ticks — Missed tick count (optional)
double

In **Connected IO** mode, returns the number of timer ticks that your model lags behind the real-time kernel. When the model lags by more than **Maximum missed ticks**, the software reports an error and simulation stops.

Dependency

When you select **Show "Missed Ticks" port**, this port is visible.

Data Types: double

Parameters

Install new board — Register a board
< no board selected > (default)

When you click **Install new board**, the software displays a list of manufacturers of supported boards. When you select a manufacturer, the software displays a list of boards available from that manufacturer. When you select a board, the software adds the board to the list of registered boards and makes that board the current board.

By default, the initial selection in the list of registered boards is < no board selected >.

Dependency

When you select a board in the list of registered boards, the **Delete current board** and **Board setup** buttons are available.

Programmatic Use
Block Parameter: DrvName

Delete current board — Delete the current board
(button)

To delete the current board, click this button. The initial selection of the list of registered boards changes to < no board selected >.

Dependency

To activate this parameter, select a board in the list of registered boards.

When you delete the current board, the **Delete current board** and **Board setup** buttons are no longer available.

Programmatic Use
Block Parameter: DrvName

Board setup — Set up the board
(button)

To set up the board, click this button.

A board-specific dialog box opens to set up the board. For more information, see the board manufacturer documentation.

Dependency

To activate this parameter, select a board in the list of registered boards.

To deactivate this parameter, click **Delete current board**.

Programmatic Use
Block Parameter: DrvAddress
Block Parameter: DrvOptions

Sample time — Sample time of block
0.1 (default) | double

Enter a value, in seconds, that represents how frequently you want the block to execute and interact with the I/O hardware. The block synchronizes your model with the real-time clock at this sample rate.

If you are using a fixed-step solver, enter the value that you entered as the **Fixed step size** configuration parameter or an integer multiple of that value.

Programmatic Use
Block Parameter: SampleTime

Maximum missed ticks — Number of timer ticks by which your model can trail the kernel
10 (default) | integer

In **Connected IO** mode, enter the number of timer ticks that your model can lag behind the real-time kernel. When the model lags behind by this number or fewer timer ticks, the software assumes that the lag is temporary. It allows the model to catch up, even if the model misses some ticks. When the model lags by more than this number, the software reports an error and simulation stops.

In **Run in Kernel** mode, the software ignores this value.

Programmatic Use
Block Parameter: MaxMissedTicks

Show "Missed Ticks" port — Send number of missed ticks to port Missed Ticks
'off' (default) | 'on'

In **Connected IO** mode, select this check box to send the number of missed ticks to output port Missed Ticks.

In **Run in Kernel** mode, the Missed Ticks port is zero.

Programmatic Use
Block Parameter: ShowMissedTicks

Yield CPU when waiting — Yield CPU while kernel waits for hardware
'off' (default) | 'on'

In **Connected IO** mode, select this check box to grant other programs more CPU time while the kernel waits for a response from the hardware.

In **Run in Kernel** mode, the software ignores this value.

Programmatic Use
Block Parameter: YieldWhenWaiting

Input channels — Select counter-input channels
1 (default) | vector

Enter a channel vector that selects the counter input channels you are using on this board. The vector can be any valid MATLAB vector form.

For example, to select the first four counter-input channels on the board, enter:

[1,2,3,4]

or

[1:4]

Programmatic Use
Block Parameter: Channels

Reset after read — Counter-reset policy
never (default) | always | level | rising edge | falling edge | either edge

Determines whether the software resets the counter to zero after reading its value. From the list, select one of the following options:

- never — Do not reset after reading.

- `always` — Always reset after reading.
- `level` — Reset after reading if the block input is nonzero. Adds an input to the Counter Input block.
- `rising edge` — Reset after reading if the block input changes from zero to nonzero between the last two successive readings. Adds an input to the Counter Input block.
- `falling edge` — Reset after reading if the block input changes from nonzero to zero between last two successive readings. Adds an input to the Counter Input block.
- `either edge` — Reset after reading if the block input changes either from zero to nonzero or from nonzero to zero between the last two successive readings. Adds an input to the Counter Input block.

Programmatic Use
Block Parameter: `ResetMode`

Clock input source — Clock input that increments counter
`input pin rising edge` (default) | `input pin falling edge` | `internal clock`

Determines the clock input source to increment the counter. From the list, select:

- `input pin rising edge` — Clock edge low-to-high transitions
- `input pin falling edge` — Clock edge high-to-low transitions
- `internal clock` — Internal time base

   If you set the **Gate input functionality** parameter to `enable when high, latch & reset on edge`, or `enable when low, latch & reset on edge`, you can measure positive or negative pulse lengths in units of the internal time base. You can use this combination with National Instruments® drivers for pulse width measurement.

Not all counter chips support selecting the input edge. In this case, only supported options appear in the list.

Programmatic Use
Block Parameter: `CounterEdge`

Gate input functionality — Action of counter-gate input pin
`none` (default) | `enable when high` | `enable when low` | `start on rising edge` | `start on falling edge` | `reset on rising edge` | `reset on falling edge` | `latch on rising edge` | `latch on falling edge` | `latch & reset on rising edge` | `latch & reset on falling edge` | `enable when high, latch & reset on edge` | `enable when low, latch & reset on edge`

Defines the action of the counter gate input pin. From the list, select:

- `none` — Enable counting unconditionally, ignoring gate input.
- `enable when high` — Disable counting when gate input is low and enable counting when gate input is high.
- `enable when low` — Disable counting when gate input is high and enable counting when gate input is low.
- `start on rising edge` — Disable counting until low-to-high gate transition occurs.
- `start on falling edge` — Disable counting until high-to-low gate transition occurs.

- `reset on rising edge` — Reset counter to zero when low-to-high gate transition occurs.
- `reset on falling edge` — Reset counter to zero when high-to-low gate transition occurs.
- `latch on rising edge` — Store count in register when low-to-high gate transition occurs. Return contents of register.
- `latch on falling edge` — Store count in register when high-to-low gate transition occurs. Return contents of register.
- `latch & reset on rising edge` — Store count in register, and then reset counter to zero when low-to-high gate transition occurs. Return contents of register.
- `latch & reset on falling edge` — Store count in register, and then reset counter to zero when high-to-low gate transition occurs. Return contents of register.
- `enable when high, latch & reset on edge` — Disable counting while gate input is low. Start counting when low-to-high gate transition occurs. Count while gate input is high. Store count in register, and then reset counter to zero when high-to-low gate transition occurs. Return contents of register.

  If you set the **Clock input source** parameter to `internal clock`, you can measure positive pulse lengths in units of the internal time base. You can use this combination with most National Instruments drivers for pulse width measurement.

- `enable when low, latch & reset on edge` — Disable counting while gate input is high. Start counting when high-to-low gate transition occurs. Count while gate input is low. Store count in register, and then reset counter to zero when low-to-high gate transition occurs. Return contents of register.

  If you set the **Clock input source** parameter to `internal clock`, you can measure negative pulse lengths in units of the internal time base. You can use this combination with National Instruments drivers for pulse width measurement.

Not all counter chips support all gate input modes. Only supported modes appear in the list.

Programmatic Use
Block Parameter: `CounterGate`

Output data type — Data type of block output
double (default) | single | int8 | uint8 | int16 | uint16 | int32 | uint32

From the list, select the type of data that the block outputs to the model. The valid values depend on the capabilities of the board.

Programmatic Use
Block Parameter: `DataType`

See Also

Topics
"Use I/O Driver Blocks"

Introduced before R2006a

Digital Input

Read from one or multiple digital input channels
Library: Simulink Desktop Real-Time

```
Digital
Input
```

Description

Connect to and read from specific digital lines or channels into your Simulink model. After you have added a Digital Input block to your model, you can enter the parameters for its I/O driver.

Ports

Output

`Digital Data` — Digital data read from input channels
vector

The data type and range of the output values depend upon the capabilities of the board.

Data Types: `single` | `double` | `int8` | `uint8` | `int16` | `uint16` | `int32` | `uint32`

`Missed Ticks` — Missed tick count (optional)
double

In **Connected IO** mode, returns the number of timer ticks that your model lags behind the real-time kernel. When the model lags by more than **Maximum missed ticks**, the software reports an error and simulation stops.

Dependency

When you select **Show "Missed Ticks" port**, this port is visible.

Data Types: `double`

Parameters

`Install new board` — Register a board
`< no board selected >` (default)

When you click **Install new board**, the software displays a list of manufacturers of supported boards. When you select a manufacturer, the software displays a list of boards available from that manufacturer. When you select a board, the software adds the board to the list of registered boards and makes that board the current board.

By default, the initial selection in the list of registered boards is `< no board selected >`.

Dependency

When you select a board in the list of registered boards, the **Delete current board** and **Board setup** buttons are available.

Programmatic Use
Block Parameter: DrvName

Delete current board — Delete the current board
(button)

To delete the current board, click this button. The initial selection of the list of registered boards changes to < no board selected >.

Dependency

To activate this parameter, select a board in the list of registered boards.

When you delete the current board, the **Delete current board** and **Board setup** buttons are no longer available.

Programmatic Use
Block Parameter: DrvName

Board setup — Set up the board
(button)

To set up the board, click this button.

A board-specific dialog box opens to set up the board. For more information, see the board manufacturer documentation.

Dependency

To activate this parameter, select a board in the list of registered boards.

To deactivate this parameter, click **Delete current board**.

Programmatic Use
Block Parameter: DrvAddress
Block Parameter: DrvOptions

Sample time — Sample time of block
0.1 (default) | double

Enter a value, in seconds, that represents how frequently you want the block to execute and interact with the I/O hardware. The block synchronizes your model with the real-time clock at this sample rate.

If you are using a fixed-step solver, enter the value that you entered as the **Fixed step size** configuration parameter or an integer multiple of that value.

Programmatic Use
Block Parameter: SampleTime

Maximum missed ticks — Number of timer ticks by which your model can trail the kernel
10 (default) | integer

In **Connected IO** mode, enter the number of timer ticks that your model can lag behind the real-time kernel. When the model lags behind by this number or fewer timer ticks, the software assumes that the lag is temporary. It allows the model to catch up, even if the model misses some ticks. When the model lags by more than this number, the software reports an error and simulation stops.

In **Run in Kernel** mode, the software ignores this value.

Programmatic Use
Block Parameter: MaxMissedTicks

Show "Missed Ticks" port — Send number of missed ticks to port Missed Ticks
'off' (default) | 'on'

In **Connected IO** mode, select this check box to send the number of missed ticks to output port Missed Ticks.

In **Run in Kernel** mode, the Missed Ticks port is zero.

Programmatic Use
Block Parameter: ShowMissedTicks

Yield CPU when waiting — Yield CPU while kernel waits for hardware
'off' (default) | 'on'

In **Connected IO** mode, select this check box to grant other programs more CPU time while the kernel waits for a response from the hardware.

In **Run in Kernel** mode, the software ignores this value.

Programmatic Use
Block Parameter: YieldWhenWaiting

Input channels — Select digital input channels
1 (default) | vector

Enter a channel vector that selects the digital input channels you are using on this board. The vector can be any valid MATLAB vector form. For example, to select the first eight digital input channels, enter:

[1,2,3,4,5,6,7,8]

or

[1:8]

If you want to use the first four digital input lines, enter:

[1,2,3,4]

If you have one 8-bit digital channel, enter [1]. If you have two 8-bit digital channels, enter [1 9], and from the Channel mode list, select Byte.

Programmatic Use
Block Parameter: Channels

Channel mode — Mode of representing the input lines
Bit (default) | Byte

From the list, select one of the following options:

- Bit — Returns a value of 0 or 1.
- Byte — Groups eight digital lines into one digital channel. Returns a value of 0–255.

Programmatic Use
Block Parameter: BitMode

Output data type — Data type of block output
double (default) | single | int8 | uint8 | int16 | uint16 | int32 | uint32

From the list, select the type of data that the block outputs to the model. The valid values depend on the capabilities of the board.

Programmatic Use
Block Parameter: DataType

See Also
Digital Output

Topics
"Use I/O Driver Blocks"

Introduced before R2006a

Digital Output

Write to one or multiple digital output channels
Library: Simulink Desktop Real-Time

Description

Connect to and write to specific digital lines or channels from your Simulink model. After you have added a Digital Output block to your model, you can enter the parameters for its I/O driver.

Ports

Input

`Digital Data` — Digital data to write to output channels
vector

The data type and range of the input values depend upon the capabilities of the board.

Data Types: `single` | `double` | `int8` | `uint8` | `int16` | `uint16` | `int32` | `uint32`

Output

`Missed Ticks` — Missed tick count (optional)
double

In **Connected IO** mode, returns the number of timer ticks that your model lags behind the real-time kernel. When the model lags by more than **Maximum missed ticks**, the software reports an error and simulation stops.

Dependency

When you select **Show "Missed Ticks" port**, this port is visible.

Data Types: `double`

Parameters

`Install new board` — Register a board
`< no board selected >` (default)

When you click **Install new board**, the software displays a list of manufacturers of supported boards. When you select a manufacturer, the software displays a list of boards available from that manufacturer. When you select a board, the software adds the board to the list of registered boards and makes that board the current board.

By default, the initial selection in the list of registered boards is `< no board selected >`.

Dependency

When you select a board in the list of registered boards, the **Delete current board** and **Board setup** buttons are available.

Programmatic Use
Block Parameter: DrvName

Delete current board — Delete the current board
(button)

To delete the current board, click this button. The initial selection of the list of registered boards changes to < no board selected >.

Dependency

To activate this parameter, select a board in the list of registered boards.

When you delete the current board, the **Delete current board** and **Board setup** buttons are no longer available.

Programmatic Use
Block Parameter: DrvName

Board setup — Set up the board
(button)

To set up the board, click this button.

A board-specific dialog box opens to set up the board. For more information, see the board manufacturer documentation.

Dependency

To activate this parameter, select a board in the list of registered boards.

To deactivate this parameter, click **Delete current board**.

Programmatic Use
Block Parameter: DrvAddress
Block Parameter: DrvOptions

Sample time — Sample time of block
0.1 (default) | double

Enter a value, in seconds, that represents how frequently you want the block to execute and interact with the I/O hardware. The block synchronizes your model with the real-time clock at this sample rate.

If you are using a fixed-step solver, enter the value that you entered as the **Fixed step size** configuration parameter or an integer multiple of that value.

Programmatic Use
Block Parameter: SampleTime

Maximum missed ticks — Number of timer ticks by which your model can trail the kernel
10 (default) | integer

In **Connected IO** mode, enter the number of timer ticks that your model can lag behind the real-time kernel. When the model lags behind by this number or fewer timer ticks, the software assumes that the lag is temporary. It allows the model to catch up, even if the model misses some ticks. When the model lags by more than this number, the software reports an error and simulation stops.

In **Run in Kernel** mode, the software ignores this value.

Programmatic Use
Block Parameter: MaxMissedTicks

Show "Missed Ticks" port — Send number of missed ticks to port Missed Ticks
'off' (default) | 'on'

In **Connected IO** mode, select this check box to send the number of missed ticks to output port Missed Ticks.

In **Run in Kernel** mode, the Missed Ticks port is zero.

Programmatic Use
Block Parameter: ShowMissedTicks

Yield CPU when waiting — Yield CPU while kernel waits for hardware
'off' (default) | 'on'

In **Connected IO** mode, select this check box to grant other programs more CPU time while the kernel waits for a response from the hardware.

In **Run in Kernel** mode, the software ignores this value.

Programmatic Use
Block Parameter: YieldWhenWaiting

Output channels — Select digital output channels
vector

Enter a channel vector that selects the digital output channels you are using on this board. The vector can be any valid MATLAB vector form. For example, to select the first eight digital output channels, enter:

[1,2,3,4,5,6,7,8]

or

[1:8]

If you want to use the first four digital output lines, enter:

[1,2,3,4]

If you have one 8-bit digital channel, enter [1]. If you have two 8-bit digital channels, enter [1 9], and from the **Channel mode** list, select Byte.

Programmatic Use
Block Parameter: Channels

Channel mode — Mode of representing the output lines
Bit (default) | Byte

From the list, select from one of the following:

- `Bit` — Expects a value of `0` or `1`.
- `Byte` — Expects a value of `0`–`255` that is converted to one digital channel of eight digital lines.

Programmatic Use
Block Parameter: `BitMode`

`Initial value` — Initial value for output channels
vector

Enter the initial values for each digital output line or channel that you entered as the **Output channels** parameter. For example, if you entered `[1,2,3,4]`, and you want initial values of `0` and `1`, enter:

`[0,0,1,1]`

If you select `Byte` from the **Channel mode** list, enter a value from `0` through 255 for each digital output channel. For example, for 1 byte (eight digital lines) with an initial value of 25, enter `[25]`. For 2 bytes (16 digital lines) with initial values of 25 and 50, enter `[25 50]`.

Programmatic Use
Block Parameter: `InitialValue`

`Final value` — Final value for output channels
vector

Enter a final value for each digital output channel that you entered as the **Output channels** parameter. For example, if you entered `[1,2,3,4]`, and you want final values of `0`, enter:

`[0,0,0,0]`

If you select `Byte` from the **Channel mode** list, enter a value from `0` through 255 for each digital output channel.

Programmatic Use
Block Parameter: `FinalValue`

See Also
Digital Input

Topics
"Use I/O Driver Blocks"

Introduced before R2006a

Encoder Input

Read from one or multiple incremental encoder input channels
Library: Simulink Desktop Real-Time
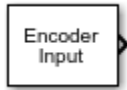
Description

Connect to and read from specific encoder input channels into your Simulink model. After you have added an Encoder Input block to your model, you can enter the parameters for its I/O driver.

Ports

Output

`Encoder Data` — Encoder data read from input channels
vector

Data representing the state of an encoder. The data type and range of the output values depend upon the capabilities of the board.

Data Types: `single` | `double` | `int8` | `uint8` | `int16` | `uint16` | `int32` | `uint32`

`Missed Ticks` — Missed tick count (optional)
double

In **Connected IO** mode, returns the number of timer ticks that your model lags behind the real-time kernel. When the model lags by more than **Maximum missed ticks**, the software reports an error and simulation stops.

Dependency

When you select **Show "Missed Ticks" port**, this port is visible.

Data Types: `double`

Parameters

`Install new board` — Register a board
`< no board selected >` (default)

When you click **Install new board**, the software displays a list of manufacturers of supported boards. When you select a manufacturer, the software displays a list of boards available from that manufacturer. When you select a board, the software adds the board to the list of registered boards and makes that board the current board.

By default, the initial selection in the list of registered boards is `< no board selected >`.

Dependency

When you select a board in the list of registered boards, the **Delete current board** and **Board setup** buttons are available.

Programmatic Use
Block Parameter: DrvName

Delete current board — Delete the current board
(button)

To delete the current board, click this button. The initial selection of the list of registered boards changes to < no board selected >.

Dependency

To activate this parameter, select a board in the list of registered boards.

When you delete the current board, the **Delete current board** and **Board setup** buttons are no longer available.

Programmatic Use
Block Parameter: DrvName

Board setup — Set up the board
(button)

To set up the board, click this button.

A board-specific dialog box opens to set up the board. For more information, see the board manufacturer documentation.

Dependency

To activate this parameter, select a board in the list of registered boards.

To deactivate this parameter, click **Delete current board**.

Programmatic Use
Block Parameter: DrvAddress
Block Parameter: DrvOptions

Sample time — Sample time of block
0.1 (default) | double

Enter a value, in seconds, that represents how frequently you want the block to execute and interact with the I/O hardware. The block synchronizes your model with the real-time clock at this sample rate.

If you are using a fixed-step solver, enter the value that you entered as the **Fixed step size** configuration parameter or an integer multiple of that value.

Programmatic Use
Block Parameter: SampleTime

Maximum missed ticks — Number of timer ticks by which your model can trail the kernel
10 (default) | integer

In **Connected IO** mode, enter the number of timer ticks that your model can lag behind the real-time kernel. When the model lags behind by this number or fewer timer ticks, the software assumes that the lag is temporary. It allows the model to catch up, even if the model misses some ticks. When the model lags by more than this number, the software reports an error and simulation stops.

In **Run in Kernel** mode, the software ignores this value.

Programmatic Use
Block Parameter: MaxMissedTicks

Show "Missed Ticks" port — Send number of missed ticks to port Missed Ticks
'off' (default) | 'on'

In **Connected IO** mode, select this check box to send the number of missed ticks to output port Missed Ticks.

In **Run in Kernel** mode, the Missed Ticks port is zero.

Programmatic Use
Block Parameter: ShowMissedTicks

Yield CPU when waiting — Yield CPU while kernel waits for hardware
'off' (default) | 'on'

In **Connected IO** mode, select this check box to grant other programs more CPU time while the kernel waits for a response from the hardware.

In **Run in Kernel** mode, the software ignores this value.

Programmatic Use
Block Parameter: YieldWhenWaiting

Input channels — Select encoder input channels
1 (default) | vector

Enter a channel vector that selects the encoder input channels that you are using on this board. The vector can be any valid MATLAB vector form. For example, to select the first four encoder input channels, enter:

[1,2,3,4]

or

[1:4]

Programmatic Use
Block Parameter: Channels

Quadrature mode — To detect optically the amplitude and direction of movement, select encoder stripe edges
quadruple (default) | double | single

Encoders use two sets of stripes, shifted in phase, to detect optically the amplitude and direction of movement. This parameter specifies which encoder stripe edges the encoder counts. From the list, select:

- quadruple (default) — Counts rising and falling edges from both stripe sets.

- `double` — Counts the rising edges from both stripe sets.
- `single` — Counts the rising edges from one stripe set.

Quadruple mode yields four times more pulses per revolution than the single mode. Therefore, quadruple is more accurate. Use quadruple mode unless other parameters dictate otherwise.

Programmatic Use
Block Parameter: QuadMode

Reset input function — Select mode of reset pin
reset (default) | gate | rising edge index | falling edge index

The encoder interface chip has a reset pin in addition to encoder inputs. Usually, you connect this pin to the index output of the encoder. However, you can connect it to any signal or not at all. This parameter specifies the function of this pin. From the list, select:

- `reset` (default) — Level reset of the encoder count.
- `gate` — Enables encoder counting.
- `rising edge index` — Resets the encoder count on the rising edge.
- `falling edge index` — Resets the encoder count on the falling edge.

Programmatic Use
Block Parameter: IndexPulse

Input filter clock frequency — Cutoff frequency (Hz) of low-pass filter
Inf (default) | scalar

The encoder interface chip includes a low-pass filter that filters out high frequencies, which the chip interprets as noise. This parameter is the cutoff frequency (Hz) of this filter. The cutoff frequency that you specify is rounded to the nearest frequency supported by the chip.

If the encoder is moving slowly and high-frequency noise is present, use the filter to eliminate the noise. With the filter in operation, the chip does not count the noise as encoder pulses. If the encoder is moving quickly, the filter can filter out the high-frequency pulses, including pulses that you want to count. In this case, consider disabling the filter by setting the cutoff frequency to Inf.

Programmatic Use
Block Parameter: InputFilter

Output data type — Data type of block output
double (default) | single | int8 | uint8 | int16 | uint16 | int32 | uint32

From the list, select the type of data that the block outputs to the model. The valid values depend on the capabilities of the board.

Programmatic Use
Block Parameter: DataType

See Also

Topics
"Use I/O Driver Blocks"

Introduced before R2006a

Execution Time

Output total execution time in Run in Kernel mode
Library: Simulink Desktop Real-Time / Target Profiling

Description

In **Run in Kernel** mode, this block outputs the execution time.

In **Connected IO** mode, this block does not perform any action. It outputs zeros.

Ports

Output

`Total Exec Time` — Total execution time for the previous base rate tick
double

Presents the total execution time as of the previous sample time.

`Task Exec Time` — Task execution times for individual sample rates (optional)
vector of double

Presents the task execution times for individual sample rates in a multitasking model as of the previous sample time.

Dependency

When you select **Measure task execution times**, this port is visible.

Parameters

`Measure task execution times` — Show task execution times for individual tasks
'off' (default) | 'on'

Makes visible the port `Task Exec Time`, which presents the task execution times for individual sample rates in a multitasking model as of the previous sample time.

Programmatic Use
Block Parameter: ShowTaskTimes

See Also
Timestamp

Topics
"Use I/O Driver Blocks"

Introduced in R2017a

Frequency Output

Write to one or multiple frequency or PWM output channels
Library: Simulink Desktop Real-Time
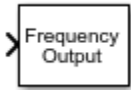
Description

Generate and output a pulse-width-modulated square wave to an analog output.

The square wave alternates between low (`0`) and high (`1`) with a specified frequency and duty cycle.

You specify the frequency in hertz (Hz).

You specify the duty cycle as a decimal fraction from `0` through `1` inclusive. The duty cycle determines the amount of time that the output signal value is high (`1`).

After you have added a Frequency Output block to your model, double-click the Frequency Output block to open the Block Parameters: Frequency Output dialog box.

Ports

Input

`Frequency` — Frequency of output waveform (optional)
vector

Frequency setting of waveform generator. The data type and range of the input values depend upon the capabilities of the board.

Dependency

When you set **Output signal frequency source** to `external`, this port is visible.

Data Types: `single` | `double` | `int8` | `uint8` | `int16` | `uint16` | `int32` | `uint32`

`Duty Cycle` — Duty cycle of output waveform (optional)
vector

Duty cycle setting of waveform generator. The data type and range of the input values depend upon the capabilities of the board.

Dependency

When you set **Output signal duty source** to `external`, this port is visible.

Data Types: `single` | `double` | `int8` | `uint8` | `int16` | `uint16` | `int32` | `uint32`

Output

`Missed Ticks` — Missed tick count (optional)
`double`

In **Connected IO** mode, returns the number of timer ticks that your model lags behind the real-time kernel. When the model lags by more than **Maximum missed ticks**, the software reports an error and simulation stops.

Dependency

When you select **Show "Missed Ticks" port**, this port is visible.

Data Types: `double`

Parameters

`Install new board` — Register a board
`< no board selected >` (default)

When you click **Install new board**, the software displays a list of manufacturers of supported boards. When you select a manufacturer, the software displays a list of boards available from that manufacturer. When you select a board, the software adds the board to the list of registered boards and makes that board the current board.

By default, the initial selection in the list of registered boards is `< no board selected >`.

Dependency

When you select a board in the list of registered boards, the **Delete current board** and **Board setup** buttons are available.

Programmatic Use
Block Parameter: `DrvName`

`Delete current board` — Delete the current board
(button)

To delete the current board, click this button. The initial selection of the list of registered boards changes to `< no board selected >`.

Dependency

To activate this parameter, select a board in the list of registered boards.

When you delete the current board, the **Delete current board** and **Board setup** buttons are no longer available.

Programmatic Use
Block Parameter: `DrvName`

`Board setup` — Set up the board
(button)

To set up the board, click this button.

A board-specific dialog box opens to set up the board. For more information, see the board manufacturer documentation.

Dependency

To activate this parameter, select a board in the list of registered boards.

To deactivate this parameter, click **Delete current board**.

Programmatic Use
Block Parameter: DrvAddress
Block Parameter: DrvOptions

Sample time — Sample time of block
0.1 (default) | double

Enter a value, in seconds, that represents how frequently you want the block to execute and interact with the I/O hardware. The block synchronizes your model with the real-time clock at this sample rate.

If you are using a fixed-step solver, enter the value that you entered as the **Fixed step size** configuration parameter or an integer multiple of that value.

Programmatic Use
Block Parameter: SampleTime

Maximum missed ticks — Number of timer ticks by which your model can trail the kernel
10 (default) | integer

In **Connected IO** mode, enter the number of timer ticks that your model can lag behind the real-time kernel. When the model lags behind by this number or fewer timer ticks, the software assumes that the lag is temporary. It allows the model to catch up, even if the model misses some ticks. When the model lags by more than this number, the software reports an error and simulation stops.

In **Run in Kernel** mode, the software ignores this value.

Programmatic Use
Block Parameter: MaxMissedTicks

Show "Missed Ticks" port — Send number of missed ticks to port Missed Ticks
'off' (default) | 'on'

In **Connected IO** mode, select this check box to send the number of missed ticks to output port Missed Ticks.

In **Run in Kernel** mode, the Missed Ticks port is zero.

Programmatic Use
Block Parameter: ShowMissedTicks

Yield CPU when waiting — Yield CPU while kernel waits for hardware
'off' (default) | 'on'

In **Connected IO** mode, select this check box to grant other programs more CPU time while the kernel waits for a response from the hardware.

In **Run in Kernel** mode, the software ignores this value.

Programmatic Use
Block Parameter: YieldWhenWaiting

Output channels — Select frequency output channels
vector

Enter a channel vector that selects the frequency output channels you are using on this board. The vector can be any valid MATLAB vector form. For example, to select the first eight digital output channels, enter:

[1,2,3,4,5,6,7,8]

or

[1:8]

Programmatic Use
Block Parameter: Channels

Output signal frequency source — Specify source of output frequency (Hz)
external (default) | internal

Select the source that specifies the frequency to output, in Hz. The **Output signal frequency source** can be either of the following:

- internal — A tunable parameter named **Frequency** specifies the frequency.
- external (default) — A signal specifies the frequency. Input port Frequency appears on the block to accept the signal.

  You can specify additional settings for **Frequency initial value** and **Frequency final value**.

Dependency

If you specify external (the default), port Frequency and the parameters **Frequency initial value** and **Frequency final value** are visible.

If you specify internal, **Frequency** replaces **Frequency initial value** and **Frequency final value** is disabled.

Programmatic Use
Block Parameter: FrequencySource

Frequency — Specify the output frequency (Hz)
scalar

This parameter replaces **Frequency initial value**.

Dependency

To make this parameter visible, set **Output signal frequency source** to internal.

Programmatic Use
Block Parameter: Frequency

Frequency initial value — Specify the frequency that the board generates before execution starts
scalar

Optionally specify an initial frequency in hertz. When you connect to the target, the specified frequency takes effect. It persists until simulation starts, when the value of the frequency signal takes effect. You can use this parameter to specify initial conditions and give them time to stabilize. If no value appears, connecting to the target has no effect on the frequency.

Dependency

To make this parameter visible, set **Output signal frequency source** to `external`.

Programmatic Use
Block Parameter: `Frequency`

`Frequency final value` — Specify the frequency that the board generates after execution ends
scalar

Optionally specify a final frequency in hertz. When simulation is complete, the specified frequency takes effect and persists indefinitely. Disconnecting from the target does not change the frequency. You can use this parameter to put a connected device into a neutral state after simulation. If no value appears, the frequency in effect at the end of simulation persists afterward.

Dependency

To make this parameter visible, set **Output signal frequency source** to `external`.

Programmatic Use
Block Parameter: `FrequencyFinalValue`

`Output signal duty source` — Specify source of output signal duty cycle
`internal` (default) | `external`

Specify the source that determines the duty cycle. The **Output signal duty source** can be either of the following:

- `internal` (default) — A tunable parameter named **Duty** specifies the duty cycle.
- `external` — A signal specifies the duty cycle. Input port `Duty Cycle` appears on the block to accept the signal.

Dependency

If you specify `internal` (the default), **Duty** is visible. **Duty final value** is not visible.

If you specify `external`, port `Duty Cycle` appears, parameter **Duty initial value** replaces **Duty**, and parameter **Duty final value** is visible.

Programmatic Use
Block Parameter: `DutySource`

`Duty` — Specify the time during which the output signal is high
`0.5` (default) | `0—1`

Specify the duty cycle as a decimal fraction from `0` through `1` inclusive. It determines the amount of time that the output signal value is high (`1`). For example, a duty cycle of `0.7` specifies that the output is high 70% of the time and low 30% of the time.

Two duty cycle values have special significance:

- 0 — The output signal is continuously low (0) regardless of the frequency specification currently in effect.
- 1 — The output signal is continuously high (1) regardless of the frequency specification currently in effect.

Dependency

To make this parameter visible, set **Output signal duty source** to `internal`.

Programmatic Use
Block Parameter: `Duty`

Duty initial value — Specify the duty cycle of the board before execution starts
0—1

Optionally specify an initial duty cycle. When you connect to the target, the specified duty cycle takes effect. It persists until simulation starts. If no value appears, connecting to the target has no effect on the duty cycle. You can specify any duty cycle. Often the **Duty initial value** is 0 or 1, enforcing a continuously low or high initial output signal. Specifying 0 or 1 overrides any **Frequency initial value** setting.

Dependency

To make this parameter visible, set **Output signal duty source** to `external`.

Programmatic Use
Block Parameter: `Duty`

Duty final value — Specify the duty cycle of the board after execution ends
0—1

Optionally specify a final duty cycle. When simulation is complete, the specified duty cycle takes effect and persists indefinitely. Disconnecting from the target does not change the duty cycle. If no value appears, the duty cycle in effect at the end of simulation persists indefinitely afterward. You can specify any duty cycle. Often the **Duty final value** is 0 or 1, enforcing a continuously low or high initial output signal. Specifying 0 or 1 overrides any **Frequency final value** setting.

Dependency

To make this parameter visible, set **Output signal duty source** to `external`.

Programmatic Use
Block Parameter: `DutyFinalValue`

See Also

Topics
"Use I/O Driver Blocks"

Introduced in R2007b

Other Input

Read from one or more input channels not handled by specific blocks
Library:　　　　　Simulink Desktop Real-Time
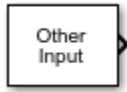
Description

Connect to and read from hardware sources requiring specialized driver support that other Simulink Desktop Real-Time input blocks do not provide. The Other Input block is rarely used and only for a few drivers. For details, see the documentation for the driver that you intend to use.

If none of the available blocks, including Other Input, provide what you need, you can create your own I/O blocks to work with Simulink Desktop Real-Time applications. For more information, see "Custom I/O Driver Basics".

Ports

Output

Data — Data read from input channels
vector

The data type and range of the output value depend on the capabilities of the board.

Data Types: `double` | `single` | `int8` | `uint8` | `int16` | `uint16` | `int32` | `uint32`

Missed Ticks — Missed tick count (optional)
double

In **Connected IO** mode, returns the number of timer ticks that your model lags behind the real-time kernel. When the model lags by more than **Maximum missed ticks**, the software reports an error and simulation stops.

Dependency

When you select **Show "Missed Ticks" port**, this port is visible.

Data Types: `double`

Parameters

Install new board — Register a board
`< no board selected >` (default)

When you click **Install new board**, the software displays a list of manufacturers of supported boards. When you select a manufacturer, the software displays a list of boards available from that manufacturer. When you select a board, the software adds the board to the list of registered boards and makes that board the current board.

By default, the initial selection in the list of registered boards is `< no board selected >`.

Dependency

When you select a board in the list of registered boards, the **Delete current board** and **Board setup** buttons are available.

Programmatic Use
Block Parameter: `DrvName`

`Delete current board` — **Delete the current board**
(button)

To delete the current board, click this button. The initial selection of the list of registered boards changes to `< no board selected >`.

Dependency

To activate this parameter, select a board in the list of registered boards.

When you delete the current board, the **Delete current board** and **Board setup** buttons are no longer available.

Programmatic Use
Block Parameter: `DrvName`

`Board setup` — **Set up the board**
(button)

To set up the board, click this button.

A board-specific dialog box opens to set up the board. For more information, see the board manufacturer documentation.

Dependency

To activate this parameter, select a board in the list of registered boards.

To deactivate this parameter, click **Delete current board**.

Programmatic Use
Block Parameter: `DrvAddress`
Block Parameter: `DrvOptions`

`Sample time` — **Sample time of block**
`0.1` (default) | double

Enter a value, in seconds, that represents how frequently you want the block to execute and interact with the I/O hardware. The block synchronizes your model with the real-time clock at this sample rate.

If you are using a fixed-step solver, enter the value that you entered as the **Fixed step size** configuration parameter or an integer multiple of that value.

Programmatic Use
Block Parameter: `SampleTime`

`Maximum missed ticks` — **Number of timer ticks by which your model can trail the kernel**
`10` (default) | integer

In **Connected IO** mode, enter the number of timer ticks that your model can lag behind the real-time kernel. When the model lags behind by this number or fewer timer ticks, the software assumes that the lag is temporary. It allows the model to catch up, even if the model misses some ticks. When the model lags by more than this number, the software reports an error and simulation stops.

In **Run in Kernel** mode, the software ignores this value.

Programmatic Use
Block Parameter: MaxMissedTicks

Show "Missed Ticks" port — Send number of missed ticks to port Missed Ticks
'off' (default) | 'on'

In **Connected IO** mode, select this check box to send the number of missed ticks to output port Missed Ticks.

In **Run in Kernel** mode, the Missed Ticks port is zero.

Programmatic Use
Block Parameter: ShowMissedTicks

Yield CPU when waiting — Yield CPU while kernel waits for hardware
'off' (default) | 'on'

In **Connected IO** mode, select this check box to grant other programs more CPU time while the kernel waits for a response from the hardware.

In **Run in Kernel** mode, the software ignores this value.

Programmatic Use
Block Parameter: YieldWhenWaiting

Input channels — Select other input channels
1 (default) | vector

Enter a channel vector that selects the other input channels that you are using on this board. The vector can be any valid MATLAB vector form. For example, to select eight contiguous analog input channels, enter:

[1,2,3,4,5,6,7,8]

or

[1:8]

To select the first three input channels, enter:

[1,2,3]

Programmatic Use
Block Parameter: Channels

Optional parameters — Specify optional parameters that the driver requires
0 (default) | vector

See the manufacturer data sheet for the required values.

Programmatic Use
Block Parameter: `OptParam`

Output data type — Data type of block output
`double` (default) | `single` | `int8` | `uint8` | `int16` | `uint16` | `int32` | `uint32`

From the list, select the type of data that the block outputs to the model. The valid values depend on the capabilities of the board.

Programmatic Use
Block Parameter: `DataType`

See Also

Other Output

Topics
"Use I/O Driver Blocks"

Introduced before R2006a

Other Output

Write to one or more output channels not handled by specific blocks
Library: Simulink Desktop Real-Time
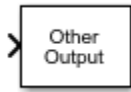
Description

Connect to and write to hardware sinks requiring specialized driver support that other Simulink Desktop Real-Time output blocks do not provide. The Other Output block is rarely used and only for a few drivers. For details, see the documentation for the driver that you intend to use.

If none of the available blocks, including Other Output, provide what you need, you can create your own I/O blocks to work with Simulink Desktop Real-Time applications. For more information, see "Custom I/O Driver Basics".

Ports

Input

Data — Data to write to output channels
vector

The data type and range of the input value depend upon the capabilities of the board.

Data Types: `single` | `double` | `int8` | `uint8` | `int16` | `uint16` | `int32` | `uint32`

Output

Missed Ticks — Missed tick count (optional)
double

In **Connected IO** mode, returns the number of timer ticks that your model lags behind the real-time kernel. When the model lags by more than **Maximum missed ticks**, the software reports an error and simulation stops.

Dependency

When you select **Show "Missed Ticks" port**, this port is visible.

Data Types: `double`

Parameters

Install new board — Register a board
`< no board selected >` (default)

When you click **Install new board**, the software displays a list of manufacturers of supported boards. When you select a manufacturer, the software displays a list of boards available from that

manufacturer. When you select a board, the software adds the board to the list of registered boards and makes that board the current board.

By default, the initial selection in the list of registered boards is `< no board selected >`.

Dependency

When you select a board in the list of registered boards, the **Delete current board** and **Board setup** buttons are available.

Programmatic Use
Block Parameter: `DrvName`

`Delete current board` — **Delete the current board**
(button)

To delete the current board, click this button. The initial selection of the list of registered boards changes to `< no board selected >`.

Dependency

To activate this parameter, select a board in the list of registered boards.

When you delete the current board, the **Delete current board** and **Board setup** buttons are no longer available.

Programmatic Use
Block Parameter: `DrvName`

`Board setup` — **Set up the board**
(button)

To set up the board, click this button.

A board-specific dialog box opens to set up the board. For more information, see the board manufacturer documentation.

Dependency

To activate this parameter, select a board in the list of registered boards.

To deactivate this parameter, click **Delete current board**.

Programmatic Use
Block Parameter: `DrvAddress`
Block Parameter: `DrvOptions`

`Sample time` — **Sample time of block**
`0.1` (default) | double

Enter a value, in seconds, that represents how frequently you want the block to execute and interact with the I/O hardware. The block synchronizes your model with the real-time clock at this sample rate.

If you are using a fixed-step solver, enter the value that you entered as the **Fixed step size** configuration parameter or an integer multiple of that value.

Programmatic Use
Block Parameter: SampleTime

Maximum missed ticks — Number of timer ticks by which your model can trail the kernel
10 (default) | integer

In **Connected IO** mode, enter the number of timer ticks that your model can lag behind the real-time kernel. When the model lags behind by this number or fewer timer ticks, the software assumes that the lag is temporary. It allows the model to catch up, even if the model misses some ticks. When the model lags by more than this number, the software reports an error and simulation stops.

In **Run in Kernel** mode, the software ignores this value.

Programmatic Use
Block Parameter: MaxMissedTicks

Show "Missed Ticks" port — Send number of missed ticks to port Missed Ticks
'off' (default) | 'on'

In **Connected IO** mode, select this check box to send the number of missed ticks to output port Missed Ticks.

In **Run in Kernel** mode, the Missed Ticks port is zero.

Programmatic Use
Block Parameter: ShowMissedTicks

Yield CPU when waiting — Yield CPU while kernel waits for hardware
'off' (default) | 'on'

In **Connected IO** mode, select this check box to grant other programs more CPU time while the kernel waits for a response from the hardware.

In **Run in Kernel** mode, the software ignores this value.

Programmatic Use
Block Parameter: YieldWhenWaiting

Output channels — Channel vector that selects other output channels
1 (default) | vector

Enter a channel vector that selects the other output channels that you are using on this board. The vector can be any valid MATLAB vector form. For example, to select the first two other output channels, enter:

[1,2]

or

[1:2]

Programmatic Use
Block Parameter: Channels

Optional parameters — Specify optional parameters that the driver requires
0 (default) | vector

See the manufacturer data sheet for the required values.

Programmatic Use
Block Parameter: `OptParam`

`Initial value` — Specify the value that the channel generates before execution starts
vector

Enter the initial value for each output channel that you entered in the **Output channels** box.

Programmatic Use
Block Parameter: `InitialValue`

`Final value` — Specify the value that the channel generates after execution ends
vector

Enter the final value for each output channel that you entered in the **Output channels** box.

Programmatic Use
Block Parameter: `FinalValue`

See Also

Other Input

Topics
"Use I/O Driver Blocks"

Introduced before R2006a

Packet Input

Read binary data or a CAN message from a communication channel
Library: Simulink Desktop Real-Time
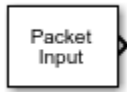
Description

Read unformatted binary data from a port or file. After you add a Packet Input block to your model, double-click the block to set its parameters.

The driver must be one of:

- **Standard Devices > File**
- **Standard Devices > Serial Port**
- **Standard Devices > TCP Protocol**
- **Standard Devices > UDP Protocol**
- **Kvaser > CAN Device**
- **PEAK-System > CAN Device**
- **Vector > CAN Device**
- **MathWorks > virtual CAN** (requires Vehicle Network Toolbox™)

When you install a UDP device, enter port addresses in decimal format in the Standard Devices UDP Protocol dialog box.

Ports

Output

Output Data — Data received from communication channel
vector

Data values that were received from the communication channel as unformatted binary data. For CAN or Virtual CAN devices, this port supports:

- CAN_MESSAGE datatype
- CAN_MESSAGE_BUS datatype
- CAN_FD_MESSAGE_BUS datatype in CAN FD mode

Data Ready — Data ready port (optional)
1 | 0

This port signals 1 if the block has new data available and 0 otherwise.

Dependency

When you select **Show "Data Ready" port**, this port is visible.

`Data Error` — Data error port (optional)
1 | 0

This port signals 1 if a data error occurs and 0 otherwise.

Dependency

When you select **Show "Data Error" port**, this port is visible.

`Timestamp` — Packet timestamp port (optional)
double

This port shows the timestamp for the CAN message packets.

Dependency

When you select **Show packet timestamp port**, this port is visible.

`Missed Ticks` — Missed tick count (optional)
double

In **Connected IO** mode, returns the number of timer ticks that your model lags behind the real-time kernel. When the model lags by more than **Maximum missed ticks**, the software reports an error and simulation stops.

Dependency

When you select **Show "Missed Ticks" port**, this port is visible.

Data Types: `double`

Parameters

`Install new board` — Register a board
< no board selected > (default)

When you click **Install new board**, the software displays a list of manufacturers of supported boards. When you select a manufacturer, the software displays a list of boards available from that manufacturer. When you select a board, the software adds the board to the list of registered boards and makes that board the current board.

By default, the initial selection in the list of registered boards is < no board selected >.

Dependency

When you select a board in the list of registered boards, the **Delete current board** and **Board setup** buttons are available.

Programmatic Use
Block Parameter: DrvName

`Delete current board` — Delete the current board
(button)

To delete the current board, click this button. The initial selection of the list of registered boards changes to `< no board selected >`.

Dependency

To activate this parameter, select a board in the list of registered boards.

When you delete the current board, the **Delete current board** and **Board setup** buttons are no longer available.

Programmatic Use
Block Parameter: `DrvName`

Board setup — Set up the board
(button)

To set up the board, click this button.

A board-specific dialog box opens to set up the board. For more information, see the board manufacturer documentation.

Dependency

To activate this parameter, select a board in the list of registered boards.

To deactivate this parameter, click **Delete current board**.

Programmatic Use
Block Parameter: `DrvAddress`
Block Parameter: `DrvOptions`

Sample time — Sample time of block
`0.1` (default) | double

Enter a value, in seconds, that represents how frequently you want the block to execute and interact with the I/O hardware. The block synchronizes your model with the real-time clock at this sample rate.

If you are using a fixed-step solver, enter the value that you entered as the **Fixed step size** configuration parameter or an integer multiple of that value.

Programmatic Use
Block Parameter: `SampleTime`

Maximum missed ticks — Number of timer ticks by which your model can trail the kernel
`10` (default) | integer

In **Connected IO** mode, enter the number of timer ticks that your model can lag behind the real-time kernel. When the model lags behind by this number or fewer timer ticks, the software assumes that the lag is temporary. It allows the model to catch up, even if the model misses some ticks. When the model lags by more than this number, the software reports an error and simulation stops.

In **Run in Kernel** mode, the software ignores this value.

Programmatic Use
Block Parameter: `MaxMissedTicks`

Show "Missed Ticks" port — Send number of missed ticks to port `Missed Ticks`
`'off'` (default) | `'on'`

In **Connected IO** mode, select this check box to send the number of missed ticks to output port
`Missed Ticks`.

In **Run in Kernel** mode, the `Missed Ticks` port is zero.

Programmatic Use
Block Parameter: `ShowMissedTicks`

Yield CPU when waiting — Yield CPU while kernel waits for hardware
`'off'` (default) | `'on'`

In **Connected IO** mode, select this check box to grant other programs more CPU time while the
kernel waits for a response from the hardware.

In **Run in Kernel** mode, the software ignores this value.

Programmatic Use
Block Parameter: `YieldWhenWaiting`

Packet identifier — Identifier of the packet
1 (default) | vector

Enter the ID of the packet to process. If your protocol does not have packet IDs, this parameter is
disabled.

Programmatic Use
Block Parameter: `PacketID`

Extended identifier — Extended identifier of the packet (CAN boards only)
`'off'` (default) | `'on'`

Select this check box if **Packet identifier** is an extended identifier.

Programmatic Use
Block Parameter: `PacketID`

Input packet size — Number of bytes that each input packet contains
32 (default) | integer

Enter the number of bytes expected in each input packet. This number must match the number of
bytes required by the type specifications in **Block output data types**.

Programmatic Use
Block Parameter: `PacketSize`

Block output data types — Output type and grouping of data from the packet
`'4*double'` (default) | character vector | cell array of character vector

One or more character vectors that specify how the data in each packet is to be typed and grouped
for input to the application. The Packet Input block has an output port corresponding to each
character vector in **Block output data types**. Changing the number of character vectors changes
the number of output ports.

Each character vector has the format [*n**]*datatype*. The data that the character vector describes
has the type specified by `datatype` and the width specified by *n*. If *n* is not specified, the width is 1.

For example, `'double'` means one `double` value, and `'4*int8'` means a vector of four `int8` values. You can also type `'CAN_MESSAGE'` to specify a single `CAN_MESSAGE` value.

By providing a cell array of such character vectors, you can convert an input packet into the types required, packaged into vectors. For example, specifying `{'int16','int16','double'}` creates three ports. The first and second each output an `int16` value, and the third outputs a `double`. Specifying `{'2*int16','double'}` creates two ports: a vector of two `int16` values, and a scalar `double`.

The block forces an out-of-range value to the maximum value for that data type.

Programmatic Use
Block Parameter: DataTypes

Input packet field byte order — Byte order of packet acquired from device
Little Endian (default) | Big Endian

From the list, select:

- Little Endian (default) — Send multibyte values in little-endian format.
- Big Endian — Send multibyte values in big-endian format.

Programmatic Use
Block Parameter: ByteOrder

Show "Data Ready" port — Add port Data Ready that signals 1 when data is available
'off' (default) | 'on'

When you select this check box, the block makes visible the output port Data Ready. This port signals 1 when the block has new data available and 0 otherwise.

Programmatic Use
Block Parameter: ShowReady

Show "Data Error" port — Add port Data Error that signals 1 if error occurs
'off' (default) | 'on'

When you select this check box, the block makes visible the output port Data Error. This port signals 1 when a data error occurs and 0 otherwise.

Programmatic Use
Block Parameter: ShowError

Show packet timestamp port — Add port Timestamp that outputs CAN message timestamps
'off' (default) | 'on'

When you select this check box, the block makes visible the output port Timestamp. This port shows the timestamps for the CAN message packets. If your protocol does not have packet timestamps, the block ignores this value.

Programmatic Use
Block Parameter: ShowTimestamp

See Also
Packet Output

Topics
"Use I/O Driver Blocks"

Introduced in R2007b

Packet Output

Write binary data or a CAN message to a communication channel
Library: Simulink Desktop Real-Time

Packet
Output

Description

Connect to and write unformatted binary data to a port, file, or CAN device. After you add a Packet Output block to your model, double-click the block to set its parameters.

The driver must be one of:

*   **Standard Devices > File**
*   **Standard Devices > Serial Port**
*   **Standard Devices > TCP Protocol**
*   **Standard Devices > UDP Protocol**
*   **Kvaser > CAN Device**
*   **PEAK-System > CAN Device**
*   **Vector > CAN Device**
*   **MathWorks > virtual CAN** (requires Vehicle Network Toolbox)

When you install a UDP device, enter port addresses in decimal format in the Standard Devices UDP Protocol dialog box.

Any serial output port can send data with sample rates up to 500 Hz. To enable a faster sample rate of up to 10 kHz for some output serial ports, click **Install new board**, select **Standard Devices > Serial Port**, and select **Direct port access**.

The block does not support this option with all types of serial output hardware.

Ports

Input

Input Data — Data to write to communication channel
vector

Data values that are sent to communication channel as unformatted binary data. For CAN or Virtual CAN devices, this port supports:

*   CAN_MESSAGE datatype
*   CAN_MESSAGE_BUS datatype
*   CAN_FD_MESSAGE_BUS datatype in CAN FD mode

Data Types: single | double | int8 | uint8 | int16 | uint16 | int32 | uint32

Output

`Data Ready` — Data ready port (optional)
`1` | `0`

This port signals `1` if the block is ready to accept new data and `0` otherwise.

Dependency

When you select **Show "Data Ready" port**, this port is visible.

`Data Error` — Data error port (optional)
`1` | `0`

This port signals `1` if a data error occurs and `0` otherwise.

Dependency

When you select **Show "Data Error" port**, this port is visible.

`Missed Ticks` — Missed tick count (optional)
double

In **Connected IO** mode, returns the number of timer ticks that your model lags behind the real-time kernel. When the model lags by more than **Maximum missed ticks**, the software reports an error and simulation stops.

Dependency

When you select **Show "Missed Ticks" port**, this port is visible.

Data Types: `double`

Parameters

`Install new board` — Register a board
`< no board selected >` (default)

When you click **Install new board**, the software displays a list of manufacturers of supported boards. When you select a manufacturer, the software displays a list of boards available from that manufacturer. When you select a board, the software adds the board to the list of registered boards and makes that board the current board.

By default, the initial selection in the list of registered boards is `< no board selected >`.

Dependency

When you select a board in the list of registered boards, the **Delete current board** and **Board setup** buttons are available.

Programmatic Use
Block Parameter: `DrvName`

`Delete current board` — Delete the current board
(button)

To delete the current board, click this button. The initial selection of the list of registered boards changes to `< no board selected >`.

Dependency

To activate this parameter, select a board in the list of registered boards.

When you delete the current board, the **Delete current board** and **Board setup** buttons are no longer available.

Programmatic Use
Block Parameter: `DrvName`

Board setup — Set up the board
(button)

To set up the board, click this button.

A board-specific dialog box opens to set up the board. For more information, see the board manufacturer documentation.

Dependency

To activate this parameter, select a board in the list of registered boards.

To deactivate this parameter, click **Delete current board**.

Programmatic Use
Block Parameter: `DrvAddress`
Block Parameter: `DrvOptions`

Sample time — Sample time of block
`0.1` (default) | double

Enter a value, in seconds, that represents how frequently you want the block to execute and interact with the I/O hardware. The block synchronizes your model with the real-time clock at this sample rate.

If you are using a fixed-step solver, enter the value that you entered as the **Fixed step size** configuration parameter or an integer multiple of that value.

Programmatic Use
Block Parameter: `SampleTime`

Maximum missed ticks — Number of timer ticks by which your model can trail the kernel
`10` (default) | integer

In **Connected IO** mode, enter the number of timer ticks that your model can lag behind the real-time kernel. When the model lags behind by this number or fewer timer ticks, the software assumes that the lag is temporary. It allows the model to catch up, even if the model misses some ticks. When the model lags by more than this number, the software reports an error and simulation stops.

In **Run in Kernel** mode, the software ignores this value.

Programmatic Use
Block Parameter: `MaxMissedTicks`

Show "Missed Ticks" port — Send number of missed ticks to port Missed Ticks
`'off'` (default) | `'on'`

In **Connected IO** mode, select this check box to send the number of missed ticks to output port `Missed Ticks`.

In **Run in Kernel** mode, the `Missed Ticks` port is zero.

Programmatic Use
Block Parameter: `ShowMissedTicks`

Yield CPU when waiting — Yield CPU while kernel waits for hardware
`'off'` (default) | `'on'`

In **Connected IO** mode, select this check box to grant other programs more CPU time while the kernel waits for a response from the hardware.

In **Run in Kernel** mode, the software ignores this value.

Programmatic Use
Block Parameter: `YieldWhenWaiting`

Packet identifier — Identifier of the packet
1 (default) | vector

Enter the ID of the packet to process. If your protocol does not have packet IDs, this parameter is disabled.

Programmatic Use
Block Parameter: `PacketID`

Extended identifier — Extended identifier of the packet (CAN boards only)
`'off'` (default) | `'on'`

Select this check box if **Packet identifier** is an extended identifier.

Programmatic Use
Block Parameter: `PacketID`

Output packet size — Number of bytes that each input packet contains
32 (default) | integer

Enter the number of bytes expected in each input packet. This number must match the number of bytes required by the type specifications in **Output packet data types**.

Programmatic Use
Block Parameter: `PacketSize`

Output packet field data types — Output type and grouping of data in the packet sent to the device
`'4*double'` (default) | character vector | cell array of character vector

One or more character vectors that specify how data provided by the application is formatted into a packet. The Packet Output block has an input port corresponding to each character vector in **Output packet field data types**. Changing the number of character vectors changes the number of ports.

Each character vector has the format [n*]*datatype*. The data that the character vector describes has the type specified by `datatype` and the width specified by *n*. If `n` is not specified, the width is 1. For example, `'double'` means one `double` value, and `'4*int8'` means a vector of four `int8` values. You can also type `'CAN_MESSAGE'` to specify a single `CAN_MESSAGE` value.

The signal input to each port of the Packet Output block can be a scalar or vector of the data type. The character vector for each port specifies the type to be used when its signal is output to the device. If the character vector for a port matches the type of the signal input to that port, the signal value appears verbatim in the output packet.

You can also perform type conversion on output. For example, assume that the input is a four-element `int16` vector and the character vector is `'4*int8'`. The four integers are converted to `int8` before being written to the packet. The resulting data occupies 4 bytes in the output packet.

The block forces an out-of-range value to the maximum value for that data type.

Programmatic Use
Block Parameter: `DataTypes`

Output packet field byte order — Byte order of packet being sent to device
Little Endian (default) | Big Endian

From the list, select:

- Little Endian (default) — Receive multibyte values in little-endian format.
- Big Endian — Receive multibyte values in big-endian format.

Programmatic Use
Block Parameter: `ByteOrder`

Show "Data Ready" port — Add port Data Ready that signals 1 when the block can accept new data
'off' (default) | 'on'

When you select this check box, the block makes visible the output port Data Ready. This output port signals 1 when the block is ready to accept new data and otherwise signals 0.

Programmatic Use
Block Parameter: `ShowReady`

Show "Data Error" port — Add port Data Error that signals 1 if error occurs
'off' (default) | 'on'

When you select this check box, the block makes visible the output port Data Error. This port signals 1 when a data error occurs and 0 otherwise.

Programmatic Use
Block Parameter: `ShowError`

Initial value — Data to send when simulation begins
vector

Optional. Enter a vector that has the same number of elements as the sum of the widths of the input signals across all ports. When simulation begins, send the specified data before the other data that is sent during simulation.

Programmatic Use
Block Parameter: `InitialValue`

Final value — Data to send when simulation ends
vector

Optional. Enter a vector that has the same number of elements as the sum of the widths of the input signals across all ports. When simulation ends, after the other data that is sent during simulation, send the specified data.

Programmatic Use
Block Parameter: `FinalValue`

See Also
Packet Input

Topics
"Use I/O Driver Blocks"

Introduced in R2007b

Real-Time Sync

Synchronize model execution to real time in Connected IO mode
Library: Simulink Desktop Real-Time

Description

Synchronize your Simulink model with the real-time kernel clock at Real-Time Sync block sample hits. The parameters specify only sample time and missed tick values. There are no input ports.

The Real-Time Sync block operates only in **Connected IO** mode. It has no function in **Run in Kernel** mode.

If the **Connected IO** mode model contains another Simulink Desktop Real-Time block, the other block synchronizes the Simulink model with the real-time clock. Do not include the Real-Time Sync block in such a model.

Ports

Output

`Missed Ticks` — Missed tick count (optional)
double

In **Connected IO** mode, returns the number of timer ticks that your model lags behind the real-time kernel. When the model lags by more than **Maximum missed ticks**, the software reports an error and simulation stops.

Dependency

When you select **Show "Missed Ticks" port**, this port is visible.

Data Types: `double`

Parameters

`Sample time` — Sample time of block
`0.1` (default) | double

Enter a value, in seconds, that represents how frequently you want the block to execute and interact with the I/O hardware. The block synchronizes your model with the real-time clock at this sample rate.

If you are using a fixed-step solver, enter the value that you entered as the **Fixed step size** configuration parameter or an integer multiple of that value.

Programmatic Use
Block Parameter: SampleTime

Maximum missed ticks — Number of timer ticks by which your model can trail the kernel
10 (default) | integer

In **Connected IO** mode, enter the number of timer ticks that your model can lag behind the real-time kernel. When the model lags behind by this number or fewer timer ticks, the software assumes that the lag is temporary. It allows the model to catch up, even if the model misses some ticks. When the model lags by more than this number, the software reports an error and simulation stops.

In **Run in Kernel** mode, the software ignores this value.

Programmatic Use
Block Parameter: MaxMissedTicks

Show "Missed Ticks" port — Send number of missed ticks to port Missed Ticks
'off' (default) | 'on'

In **Connected IO** mode, select this check box to send the number of missed ticks to output port Missed Ticks.

In **Run in Kernel** mode, the Missed Ticks port is zero.

Programmatic Use
Block Parameter: ShowMissedTicks

Yield CPU when waiting — Yield CPU while kernel waits for hardware
'off' (default) | 'on'

In **Connected IO** mode, select this check box to grant other programs more CPU time while the kernel waits for a response from the hardware.

In **Run in Kernel** mode, the software ignores this value.

Programmatic Use
Block Parameter: YieldWhenWaiting

See Also

Topics
"Use I/O Driver Blocks"

Introduced in R2012a

Servo Output

Write to standard servo output channels
Library: Simulink Desktop Real-Time

Description

Set the shaft position of a standard servo motor.

To rotate the servo shaft, send values to the block input.

Sending out-of-range values to the block input has the same effect as sending the minimum or maximum input values.

The block input inherits the data type of the upstream block.

Ports

Input

Servo Data — Servo data to write to output channels
double | single | int8 | uint8 | int16 | uint16 | int32 | uint32 | boolean

Servo position (angle). The minimum and maximum values for shaft position vary with the **Block input signal** (units) argument value:

Block input signal	Minimum Servo Data	Maximum Servo Data
Degrees (default)	0	180
Radians	0	pi
Normalized bipolar	-1	1
Normalized unipolar	0	1

Example: 90

Data Types: double | single | int8 | uint8 | int16 | uint16 | int32 | uint32

Output

Missed Ticks — Missed tick count (optional)
double

In **Connected IO** mode, returns the number of timer ticks that your model lags behind the real-time kernel. When the model lags by more than **Maximum missed ticks**, the software reports an error and simulation stops.

Dependency

When you select **Show "Missed Ticks" port**, this port is visible.

Data Types: `double`

Parameters

`Install new board` **— Register a board**
`< no board selected >` (default)

When you click **Install new board**, the software displays a list of manufacturers of supported boards. When you select a manufacturer, the software displays a list of boards available from that manufacturer. When you select a board, the software adds the board to the list of registered boards and makes that board the current board.

By default, the initial selection in the list of registered boards is `< no board selected >`.

Dependency

When you select a board in the list of registered boards, the **Delete current board** and **Board setup** buttons are available.

Programmatic Use
Block Parameter: `DrvName`

`Delete current board` **— Delete the current board**
(button)

To delete the current board, click this button. The initial selection of the list of registered boards changes to `< no board selected >`.

Dependency

To activate this parameter, select a board in the list of registered boards.

When you delete the current board, the **Delete current board** and **Board setup** buttons are no longer available.

Programmatic Use
Block Parameter: `DrvName`

`Board setup` **— Set up the board**
(button)

To set up the board, click this button.

A board-specific dialog box opens to set up the board. For more information, see the board manufacturer documentation.

Dependency

To activate this parameter, select a board in the list of registered boards.

To deactivate this parameter, click **Delete current board**.

Programmatic Use
Block Parameter: DrvAddress
Block Parameter: DrvOptions

Sample time — Sample time of block
0.1 (default) | double

Enter a value, in seconds, that represents how frequently you want the block to execute and interact with the I/O hardware. The block synchronizes your model with the real-time clock at this sample rate.

If you are using a fixed-step solver, enter the value that you entered as the **Fixed step size** configuration parameter or an integer multiple of that value.

Programmatic Use
Block Parameter: SampleTime

Maximum missed ticks — Number of timer ticks by which your model can trail the kernel
10 (default) | integer

In **Connected IO** mode, enter the number of timer ticks that your model can lag behind the real-time kernel. When the model lags behind by this number or fewer timer ticks, the software assumes that the lag is temporary. It allows the model to catch up, even if the model misses some ticks. When the model lags by more than this number, the software reports an error and simulation stops.

In **Run in Kernel** mode, the software ignores this value.

Programmatic Use
Block Parameter: MaxMissedTicks

Show "Missed Ticks" port — Send number of missed ticks to port Missed Ticks
'off' (default) | 'on'

In **Connected IO** mode, select this check box to send the number of missed ticks to output port Missed Ticks.

In **Run in Kernel** mode, the Missed Ticks port is zero.

Programmatic Use
Block Parameter: ShowMissedTicks

Yield CPU when waiting — Yield CPU while kernel waits for hardware
'off' (default) | 'on'

In **Connected IO** mode, select this check box to grant other programs more CPU time while the kernel waits for a response from the hardware.

In **Run in Kernel** mode, the software ignores this value.

Programmatic Use
Block Parameter: YieldWhenWaiting

Output channels — Channel vector that selects other output channels
1 (default) | vector

Enter a channel vector that selects the other output channels that you are using on this board. The vector can be any valid MATLAB vector form. For example, to select the first two other output channels, enter:

`[1,2]`

or

`[1:2]`

Programmatic Use
Block Parameter: `Channels`

`Block input signal` — Units of the block input signal
`Degrees (default)` | `Radians` | `Normalized bipolar` | `Normalized unipolar`

Select the units of the block input signal that specifies servo position.

Programmatic Use
Block Parameter: `RangeMode`

`Minimum pulse duration` — Duration of the control pulse
`500e-6 (default)`

Select the duration of the control pulse that corresponds to the minimum servo angle. Because not all servos are the same, this parameter lets you calibrate the input signal to the actual servo angle.

Programmatic Use
Block Parameter: `MinPulseDuration`

`Maximum pulse duration` — Duration of the control pulse
`2500e-6 (default)`

Select the duration of the control pulse that corresponds to the maximum servo angle. Because not all servos are the same, this parameter lets you calibrate the input signal to the actual servo angle.

Programmatic Use
Block Parameter: `MaxPulseDuration`

`Initial value` — Specify the value that the channel generates before execution starts
`vector`

Enter the initial value for each output channel that you entered in the **Output channels** box.

Programmatic Use
Block Parameter: `InitialValue`

`Final value` — Specify the value that the channel generates after execution ends
`vector`

Enter the final value for each output channel that you entered in the **Output channels** box.

Programmatic Use
Block Parameter: `FinalValue`

2-61

See Also
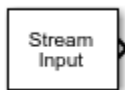
Topics
"Use I/O Driver Blocks"

Introduced in R2018b

Stream Input

Read formatted ASCII data from communication channel
Library: Simulink Desktop Real-Time

Description

Read formatted ASCII data from a port or file. After you have added a Stream Input block to your model, double-click the Stream Input block to open the Block Parameters: Stream Input dialog box.

The driver must be one of:

- **Standard Devices > File**
- **Standard Devices > Serial Port**
- **Standard Devices > TCP Protocol**
- **Standard Devices > UDP Protocol**

Specify parameter values as required, then click **OK** or **Apply**. When you install a UDP device, enter port addresses in decimal format in the Standard Devices UDP Protocol dialog box.

Ports

Output

`Output Data` — Data received from communication channel
numeric vector | string

Data values that were received from the communication channel as formatted ASCII data and decoded according to **Format string**.

`Data Ready` — Data ready port (optional)
1 | 0

This port signals 1 if the block has new data available and 0 otherwise.

Dependency

When you select **Show "Data Ready" port**, this port is visible.

`Data Error` — Data error port (optional)
1 | 0

This port signals 1 if a data error occurs and 0 otherwise.

Dependency

When you select **Show "Data Error" port**, this port is visible.

Missed Ticks — Missed tick count (optional)
double

In **Connected IO** mode, returns the number of timer ticks that your model lags behind the real-time kernel. When the model lags by more than **Maximum missed ticks**, the software reports an error and simulation stops.

Dependency

When you select **Show "Missed Ticks" port**, this port is visible.

Data Types: `double`

Parameters

Install new board — Register a board
`< no board selected >` (default)

When you click **Install new board**, the software displays a list of manufacturers of supported boards. When you select a manufacturer, the software displays a list of boards available from that manufacturer. When you select a board, the software adds the board to the list of registered boards and makes that board the current board.

By default, the initial selection in the list of registered boards is `< no board selected >`.

Dependency

When you select a board in the list of registered boards, the **Delete current board** and **Board setup** buttons are available.

Programmatic Use
Block Parameter: `DrvName`

Delete current board — Delete the current board
(button)

To delete the current board, click this button. The initial selection of the list of registered boards changes to `< no board selected >`.

Dependency

To activate this parameter, select a board in the list of registered boards.

When you delete the current board, the **Delete current board** and **Board setup** buttons are no longer available.

Programmatic Use
Block Parameter: `DrvName`

Board setup — Set up the board
(button)

To set up the board, click this button.

A board-specific dialog box opens to set up the board. For more information, see the board manufacturer documentation.

Dependency

To activate this parameter, select a board in the list of registered boards.

To deactivate this parameter, click **Delete current board**.

Programmatic Use
Block Parameter: DrvAddress
Block Parameter: DrvOptions

Sample time — Sample time of block
0.1 (default) | double

Enter a value, in seconds, that represents how frequently you want the block to execute and interact with the I/O hardware. The block synchronizes your model with the real-time clock at this sample rate.

If you are using a fixed-step solver, enter the value that you entered as the **Fixed step size** configuration parameter or an integer multiple of that value.

Programmatic Use
Block Parameter: SampleTime

Maximum missed ticks — Number of timer ticks by which your model can trail the kernel
10 (default) | integer

In **Connected IO** mode, enter the number of timer ticks that your model can lag behind the real-time kernel. When the model lags behind by this number or fewer timer ticks, the software assumes that the lag is temporary. It allows the model to catch up, even if the model misses some ticks. When the model lags by more than this number, the software reports an error and simulation stops.

In **Run in Kernel** mode, the software ignores this value.

Programmatic Use
Block Parameter: MaxMissedTicks

Show "Missed Ticks" port — Send number of missed ticks to port Missed Ticks
'off' (default) | 'on'

In **Connected IO** mode, select this check box to send the number of missed ticks to output port Missed Ticks.

In **Run in Kernel** mode, the Missed Ticks port is zero.

Programmatic Use
Block Parameter: ShowMissedTicks

Yield CPU when waiting — Yield CPU while kernel waits for hardware
'off' (default) | 'on'

In **Connected IO** mode, select this check box to grant other programs more CPU time while the kernel waits for a response from the hardware.

In **Run in Kernel** mode, the software ignores this value.

Programmatic Use
Block Parameter: YieldWhenWaiting

Block output data types — Output type and grouping of data sent to the block outputs
'double' (default) | character vector | cell array of character vector

A character vector or a cell array of character vectors. The block has as many output ports as the number of character vectors. Each vector specifies a data type by name, and optionally the number of items of that data type in an output vector. For example, 'double' means one double value, and '4*int8' means four int8 values.

The values made available on output ports are grouped and typed according to the **Block output data types** character vectors. For example, if **Block output data types** is {'2*int8','3*double'}, the block outputs:

- On output port 1 — an int8 vector of width 2
- On output port 2 — a double vector of width 3

Programmatic Use
Block Parameter: DataTypes

Format string — Format of data items being read from the stream
'%f' (default) | character vector

A specification in the same format used by C library I/O routines like scanf. The format string describes the data to be received. The number of elements in the format string must equal the number of data items specified in **Block output data types**. If the specifier is %s or %c for string, the created output port of type string only produces string signals. The Block output data types parameter value for this output port must be "1", because strings can only be scalars.

For example, assume that **Block output data types** is {'2*int8','3*double'} and **Format string** is '%d %d %f %f %f'. The block reads an ASCII representation of two integers and three doubles. The block makes the resulting values available to the application on:

- Output port 1 — an int8 vector of width 2
- Output port 2 — a double vector of width 3

If the data type specified for a value in **Block output data types** differs from the type of the corresponding element in **Format string**, type conversion occurs. The block reads data as specified by **Format string**, converts the data to match the **Block output data types**, and provides the data to the application on the block output ports.

Programmatic Use
Block Parameter: FormatString

Message termination — Specify end of message
{'\r', '\n'} (default) | character vector | cell array of character vector | integer

A character vector, cell array of character vectors, or a number. If the value is a character vector, receiving this sequence of characters terminates data input. If the value is a cell array, any character vector in the cell array terminates data input. If the value is a number, data input terminates after reading the specified number of characters.

Programmatic Use
Block Parameter: Terminator

Show "Data Ready" port — Add port Data Ready that signals 1 when data is available
'off' (default) | 'on'

When you select this check box, the block makes visible the output port `Data Ready`. This port signals `1` when the block has new data available and `0` otherwise.

Programmatic Use
Block Parameter: ShowReady

Show "Data Error" port — Add port `Data Error` that signals 1 if error occurs
'off' (default) | 'on'

When you select this check box, the block makes visible the output port `Data Error`. This port signals `1` when a data error occurs and `0` otherwise.

Programmatic Use
Block Parameter: ShowError

See Also
Stream Output

Topics
"Use I/O Driver Blocks"

Introduced in R2007b

Stream Output

Write formatted ASCII data to communication channel
Library: Simulink Desktop Real-Time

Description

Write formatted ASCII data to a port or file. After you have added a Stream Output block to your model, double-click the Stream Output block to open the Block Parameters: Stream Output dialog box.

The driver must be one of:

- **Standard Devices > File**
- **Standard Devices > Serial Port**
- **Standard Devices > TCP Protocol**
- **Standard Devices > UDP Protocol**

When you install a UDP device, enter port addresses in decimal format in the Standard Devices UDP Protocol dialog box.

Any serial output port can send data with sample rates up to 500 Hz. To enable a faster sample rate of up to 10 kHz for some output serial ports, click **Install new board**, select **Standard Devices > Serial Port**, and select **Direct port access**.

The block does not support this option with all types of serial output hardware.

Ports

Input

Input Data — Data to write to communication channel
numeric vector | string

Data values that are sent to the communication channel as ASCII data, encoded according to **Format string**.

Output

Data Ready — Data ready port (optional)
1 | 0

This port signals 1 if the block is ready to accept new data and 0 otherwise.

Dependency

When you select **Show "Data Ready" port**, this port is visible.

Data Error — Data error port (optional)
1 | 0

This port signals 1 if a data error occurs and 0 otherwise.

Dependency

When you select **Show "Data Error" port**, this port is visible.

Missed Ticks — Missed tick count (optional)
double

In **Connected IO** mode, returns the number of timer ticks that your model lags behind the real-time kernel. When the model lags by more than **Maximum missed ticks**, the software reports an error and simulation stops.

Dependency

When you select **Show "Missed Ticks" port**, this port is visible.

Data Types: double

Parameters

Install new board — Register a board
< no board selected > (default)

When you click **Install new board**, the software displays a list of manufacturers of supported boards. When you select a manufacturer, the software displays a list of boards available from that manufacturer. When you select a board, the software adds the board to the list of registered boards and makes that board the current board.

By default, the initial selection in the list of registered boards is < no board selected >.

Dependency

When you select a board in the list of registered boards, the **Delete current board** and **Board setup** buttons are available.

Programmatic Use
Block Parameter: DrvName

Delete current board — Delete the current board
(button)

To delete the current board, click this button. The initial selection of the list of registered boards changes to < no board selected >.

Dependency

To activate this parameter, select a board in the list of registered boards.

When you delete the current board, the **Delete current board** and **Board setup** buttons are no longer available.

Programmatic Use
Block Parameter: DrvName

Board setup — Set up the board
(button)

To set up the board, click this button.

A board-specific dialog box opens to set up the board. For more information, see the board manufacturer documentation.

Dependency

To activate this parameter, select a board in the list of registered boards.

To deactivate this parameter, click **Delete current board**.

Programmatic Use
Block Parameter: DrvAddress
Block Parameter: DrvOptions

Sample time — Sample time of block
0.1 (default) | double

Enter a value, in seconds, that represents how frequently you want the block to execute and interact with the I/O hardware. The block synchronizes your model with the real-time clock at this sample rate.

If you are using a fixed-step solver, enter the value that you entered as the **Fixed step size** configuration parameter or an integer multiple of that value.

Programmatic Use
Block Parameter: SampleTime

Maximum missed ticks — Number of timer ticks by which your model can trail the kernel
10 (default) | integer

In **Connected IO** mode, enter the number of timer ticks that your model can lag behind the real-time kernel. When the model lags behind by this number or fewer timer ticks, the software assumes that the lag is temporary. It allows the model to catch up, even if the model misses some ticks. When the model lags by more than this number, the software reports an error and simulation stops.

In **Run in Kernel** mode, the software ignores this value.

Programmatic Use
Block Parameter: MaxMissedTicks

Show "Missed Ticks" port — Send number of missed ticks to port Missed Ticks
'off' (default) | 'on'

In **Connected IO** mode, select this check box to send the number of missed ticks to output port Missed Ticks.

In **Run in Kernel** mode, the Missed Ticks port is zero.

Programmatic Use
Block Parameter: ShowMissedTicks

Yield CPU when waiting — Yield CPU while kernel waits for hardware
'off' (default) | 'on'

In **Connected IO** mode, select this check box to grant other programs more CPU time while the kernel waits for a response from the hardware.

In **Run in Kernel** mode, the software ignores this value.

Programmatic Use
Block Parameter: `YieldWhenWaiting`

`Input port sizes` — Specify number and width of ports
1 (default) | vector

Enter a port width, or vector of port widths. The number of elements determines the number of input ports. Each port has the width specified by the corresponding element. For example, specifying `[3,1,2]` indicates that the block has three input ports: a vector of width 3, a scalar, and a vector of width 2, for a total of six elements. You specify only the widths.

Programmatic Use
Block Parameter: `PortSizes`

`Format string` — Format of data items being written to the stream
'%g' (default) | character vector

Enter a specification in the same format used by C library I/O routines like `printf`. The format string describes the data to be sent. The format string must have the same number of specifiers as the sum of the elements of **Input port sizes**. If the specifier is `%s` or `%c` for string, the created input port of type string only accepts string signals. The `Input port sizes` parameter value for this input port must be `"1"`, because strings can only be scalars.

For example, assume that **Input port sizes** is `[3,1,2]`. Then a **Format string** of '%d %d %d %f %d %d' outputs ASCII representing:

- Three integers from the vector on the first input port
- One double from the scalar on the input second port
- Two integers from the vector on the third input port

If the data type specified in the **Format string** for a value differs from the type of the actual value, type conversion occurs. Converted data that conforms to the **Format string** is output as ASCII to the device.

Programmatic Use
Block Parameter: `FormatString`

Show "Data Ready" port — Add port Data Ready that signals 1 when the block can accept new data
'off' (default) | 'on'

When you select this check box, the block makes visible the output port `Data Ready`. This output port signals 1 when the block is ready to accept new data and otherwise signals 0.

Programmatic Use
Block Parameter: `ShowReady`

Show "Data Error" port — Add port Data Error that signals 1 if error occurs
'off' (default) | 'on'

When you select this check box, the block makes visible the output port `Data Error`. This port signals `1` when a data error occurs and `0` otherwise.

Programmatic Use
Block Parameter: `ShowError`

`Initial string` — Data to send when simulation begins
character vector

Optional. Enter a character vector. When simulation begins, send the specified data before the other data that is sent during simulation.

The character vector can contain anything. You can use it to initialize a device. It is sent literally as specified, without reference to **Format string**.

Programmatic Use
Block Parameter: `InitialValue`

`Final string` — Data to send when simulation ends
character vector

Optional. Enter a character vector. When simulation ends, after the other data that is sent during simulation, send the specified data.

The character vector can contain anything. You can use it to shut down a device. It is sent literally as specified, without reference to **Format string**

Programmatic Use
Block Parameter: `FinalValue`

See Also

Stream Input

Topics
"Use I/O Driver Blocks"

Introduced in R2007b

ThingSpeak Input

Read input fields from ThingSpeak channel
Library: Simulink Desktop Real-Time / Internet of Things

```
ThingSpeak Input
channel 12397
```

Description

The ThingSpeak Input block reads numeric data from ThingSpeak™ channels.

By default, this block uses the public ThingSpeak server, `api.thingspeak.com`. To configure the ThingSpeak channel, click **Configure channel parameters**.

To access the web page for **Channel number**, click **View channel web page**.

This block does not synchronize the ThingSpeak timestamp with the simulation time. For example, assume that the block is set to a sample time of `0.1` s and that the channel update period is `20` s. The sensor acquires new data at time *T* and transmits it to the ThingSpeak server, which gives the data a timestamp. If new data arrives before the `20` s update period has passed, the new data is lost. The block runs every `0.1` s and gets new data only every `20` s. The rest of the time, the block gets old data.

Ports

Output

`Data` — **Data read from channel**
numeric

Scalar numeric data read from the ThingSpeak channel.

Example: `3.14159`

Dependency

The setting of the **Output data type** parameter determines the port data type.

Data Types: `single` | `double` | `int8` | `int16` | `int32` | `uint8` | `uint16` | `uint32`

Parameters

`Channel number` — **ThingSpeak channel being accessed**
`0` (default) | numeric

This property is read-only.

To select a channel, click the **Configure channel parameters** button. In the Network Services ThingSpeak dialog box, set the **Channel ID** parameter.

Programmatic Use
Block Parameter: DrvAddress

Configure channel parameters — Select and configure channel to access
(button)

To configure the channel, click this button. A dialog box displays the channel parameter settings.

To accept your changes, click **OK**. To test your changes, click **Test**. To revert your settings to their previous values, click **Revert**. To exit the dialog box without changing the settings, click **Cancel**.

Programmatic Use
Block Parameter: DrvName
Block Parameter: DrvOptions

ThingSpeak server — Server for data transfer
api.thingspeak.com (default) | character vector

The default value is the standard address of the public ThingSpeak server.

Programmatic Use
Block Parameter: DrvName
Block Parameter: DrvOptions

Channel ID — Unique channel
12397 (default) | numeric

Select a unique channel among the available channels.

You cannot read and write over the same ThingSpeak channel in the same model.

Programmatic Use
Block Parameter: DrvOptions

Read API key — Access key for reading from channel
character vector

By default, this key is the API key for the channel on the public server, api.thingspeak.com.

Programmatic Use
Block Parameter: DrvOptions

Update period (secs) — Read update period, in seconds
20 (default) | numeric

Time between successive reads over the channel.

Programmatic Use
Block Parameter: DrvOptions

View channel web page — Access web page for channel
(button)

To access the web page for **Channel number**, click **View channel web page**.

Programmatic Use
Block Parameter: DrvOptions

Sample time — Sample time of block
`0.1` (default) | double

Enter a value, in seconds, that represents how frequently you want the block to execute and interact with the I/O hardware. The block synchronizes your model with the real-time clock at this sample rate.

If you are using a fixed-step solver, enter the value that you entered as the **Fixed step size** configuration parameter or an integer multiple of that value.

Programmatic Use
Block Parameter: `SampleTime`

Maximum missed ticks — Number of timer ticks by which your model can trail the kernel
`10` (default) | integer

In **Connected IO** mode, enter the number of timer ticks that your model can lag behind the real-time kernel. When the model lags behind by this number or fewer timer ticks, the software assumes that the lag is temporary. It allows the model to catch up, even if the model misses some ticks. When the model lags by more than this number, the software reports an error and simulation stops.

In **Run in Kernel** mode, the software ignores this value.

Programmatic Use
Block Parameter: `MaxMissedTicks`

Show "Missed Ticks" port — Send number of missed ticks to port `Missed Ticks`
`'off'` (default) | `'on'`

In **Connected IO** mode, select this check box to send the number of missed ticks to output port `Missed Ticks`.

In **Run in Kernel** mode, the `Missed Ticks` port is zero.

Programmatic Use
Block Parameter: `ShowMissedTicks`

Yield CPU when waiting — Yield CPU while kernel waits for hardware
`'off'` (default) | `'on'`

In **Connected IO** mode, select this check box to grant other programs more CPU time while the kernel waits for a response from the hardware.

In **Run in Kernel** mode, the software ignores this value.

Programmatic Use
Block Parameter: `YieldWhenWaiting`

Channel fields to read — Define which parts of channel to read
`1` (default) | [1–8]

You can read 1–8 fields from each channel. Use this parameter to ignore implementation-specific information and receive only the fields that your application requires.

Programmatic Use
Block Parameter: `ChannelFields`

Output data type — Define data type of output port
double (default) | single | int8 | uint8 | int16 | uint16 | int32 | uint32 | <data type expression>

Define the data type of the output port.

To open the Data Type Assistant, click the **>>** button. To close the Data Type Assistant, click the **<<** button.

Programmatic Use
Block Parameter: DataTypeStr

Mode — Method of defining type
Built in (default) | Expression

Use a built-in type, such as double or one defined by a data type expression.

Programmatic Use
Block Parameter: OptParam

Type — Data type selector
double (default) | single | int8 | uint8 | int16 | uint16 | int32 | uint32

Set of built-in types that are available for constructing data type expressions.

Programmatic Use
Block Parameter: DataTypeStr

Data type override — Policy on a signal type mismatch
inherit (default) | off

- inherit — Port inherits type of signal that it is connected to.
- off — Port sets signal data type to the type defined by this block.

Programmatic Use
Block Parameter: DataTypeStr

See Also
ThingSpeak Output

Topics
"Use I/O Driver Blocks"

Introduced in R2017b

ThingSpeak Output

Write output fields to ThingSpeak channel
Library: Simulink Desktop Real-Time / Internet of Things

> ThingSpeak Output
> channel 0

Description

The ThingSpeak Output block writes numeric data to 1–8 ThingSpeak channels.

By default, this block uses the public ThingSpeak server, `api.thingspeak.com`. To configure the ThingSpeak channel, click **Configure channel parameters**.

To access the web page for **Channel number**, click **View channel web page**.

This block does not synchronize the ThingSpeak timestamp with the simulation time. For example, assume that the block is set to a sample time of `0.1` s and that the channel update period is `20` s. The block transmits the data at a particular simulation time, *T*, but ThingSpeak writes the data and gives it a timestamp only at the `20` update period.

Ports

Input

Data — Data to write to channel
numeric

Scalar numeric data to write to the ThingSpeak channel.

Example: `3.14159`

Data Types: `single` | `double` | `int8` | `int16` | `int32` | `uint8` | `uint16` | `uint32`

Parameters

Channel number — ThingSpeak channel being accessed
`0` (default) | numeric

This property is read-only.

To select a channel, click the **Configure channel parameters** button. In the Network Services ThingSpeak dialog box, set the **Channel ID** parameter.

Programmatic Use
Block Parameter: `DrvAddress`

Configure channel parameters — Select and configure channel to access
(button)

To configure the channel, click this button. A dialog box displays the channel parameter settings.

To accept your changes, click **OK**. To test your changes, click **Test**. To revert your settings to their previous values, click **Revert**. To exit the dialog box without changing the settings, click **Cancel**.

Programmatic Use
Block Parameter: DrvName
Block Parameter: DrvOptions

ThingSpeak server — Server for data transfer
api.thingspeak.com (default) | character vector

The default value is the standard address of the public ThingSpeak server.

Programmatic Use
Block Parameter: DrvName
Block Parameter: DrvOptions

Channel ID — Unique channel
0 (default) | numeric

Select a unique channel among the available channels.

You cannot read and write over the same ThingSpeak channel in the same model.

Programmatic Use
Block Parameter: DrvOptions

Write API key — Access key for writing to channel
character vector

By default, this key is the API key for the channel on the public server, api.thingspeak.com.

Programmatic Use
Block Parameter: DrvOptions

Update period (secs) — Write update period, in seconds
20 (default) | numeric

Time between successive writes to the channel.

Programmatic Use
Block Parameter: DrvOptions

View channel web page — Access web page for channel
(button)

To access the web page for **Channel number**, click **View channel web page**.

Programmatic Use
Block Parameter: DrvOptions

Sample time — Sample time of block
0.1 (default) | double

Enter a value, in seconds, that represents how frequently you want the block to execute and interact with the I/O hardware. The block synchronizes your model with the real-time clock at this sample rate.

If you are using a fixed-step solver, enter the value that you entered as the **Fixed step size** configuration parameter or an integer multiple of that value.

Programmatic Use
Block Parameter: SampleTime

Maximum missed ticks — Number of timer ticks by which your model can trail the kernel
10 (default) | integer

In **Connected IO** mode, enter the number of timer ticks that your model can lag behind the real-time kernel. When the model lags behind by this number or fewer timer ticks, the software assumes that the lag is temporary. It allows the model to catch up, even if the model misses some ticks. When the model lags by more than this number, the software reports an error and simulation stops.

In **Run in Kernel** mode, the software ignores this value.

Programmatic Use
Block Parameter: MaxMissedTicks

Show "Missed Ticks" port — Send number of missed ticks to port Missed Ticks
'off' (default) | 'on'

In **Connected IO** mode, select this check box to send the number of missed ticks to output port Missed Ticks.

In **Run in Kernel** mode, the Missed Ticks port is zero.

Programmatic Use
Block Parameter: ShowMissedTicks

Yield CPU when waiting — Yield CPU while kernel waits for hardware
'off' (default) | 'on'

In **Connected IO** mode, select this check box to grant other programs more CPU time while the kernel waits for a response from the hardware.

In **Run in Kernel** mode, the software ignores this value.

Programmatic Use
Block Parameter: YieldWhenWaiting

Output fields — Define which parts of data to write
1 (default) | [1—8]

You can write 1—8 fields to each channel. Use this parameter to ignore implementation-specific information and transmit only the fields that your application requires.

Programmatic Use
Block Parameter: ChannelFields

Initial value — First value transmitted
[] (default) | [integer]

Transmit an initial value, for example an initial state value or an initialization command.

Programmatic Use
Block Parameter: InitialValue

Final value — Last value transmitted
[] (default) | [integer]

Transmit a final value, for example a final state value or a wrap-up command.

Programmatic Use
Block Parameter: FinalValue

See Also
ThingSpeak Input

Topics
"Use I/O Driver Blocks"

Introduced in R2017b

Timestamp

Output signal timestamp in Run in Kernel mode
Library: Simulink Desktop Real-Time / Target Profiling

Description

In **Run in Kernel** mode, this block outputs the timestamp for its input signal. The timestamp is relative to model start time. The input signal can be any signal of any data type.

In **Connected IO** mode, this block does not perform any action. It outputs zeros.

Ports

Input

`Signals` — Signals for which to capture timestamps
vector

The actual data values are ignored.

Output

`Timestamps` — Timestamps of signals
vector of double

For each signal connected to `Signals`, the block returns the signal timestamp measured from the beginning of model execution.

See Also
Execution Time

Topics
"Use I/O Driver Blocks"

Introduced in R2017a

3

Configuration Parameters

Simulink Desktop Real-Time Pane

Parameter	Description
"Code replacement library" on page 3-4	Specify a target-specific floating-point math library extension
"Shared code placement" on page 3-5	Specify the location for generating shared utility code
"Generate assembly listings" on page 3-6	Instruct the C compiler to generate assembly listings for the generated code.
"Rebuild all" on page 3-7	Force the object files to be rebuilt regardless of their timestamps.
"External mode" on page 3-8	Enable client/server communication between Simulink software and an application.
"Transport layer" on page 3-9	Specify the transport protocol for external mode communication.
"MEX-file arguments" on page 3-10	Specify external mode MEX arguments.

Control the code created by Simulink Coder™ code generation software for a Simulink Desktop Real-Time application.

Configuration

The **Simulink Desktop Real-Time** node in the Configuration Parameters dialog box allows you to specify how the software generates the real-time application. To reveal the **Simulink Desktop Real-Time** node, in the **Code Generation** pane, in the **System target file** list, select one of these settings:

- sldrt.tlc

  Generate system target code for Simulink Desktop Real-Time.

- sldrtert.tlc

  Generate system target code for a Simulink Desktop Real-Time using the required Embedded Coder software.

Note If you open a model that was originally saved with **System target file** set to rtwin.tlc, the software updates the setting to sldrt.tlc, and likewise with rtwinert.tlc and sldrtert.tlc. To retain the updated setting, save the updated model.

To get help on an option

1   Right-click the option text label.
2   From the context menu, select **What's This**.
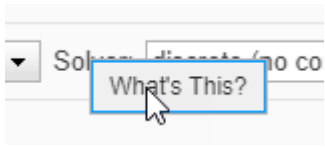
Code replacement library

Specify a target-specific floating-point math library extension

Settings

Default: None

None

   Does not use a code replacement library.

GNU C99 extensions

   Generates calls to the GNU® gcc math library, which provides C99 extensions as defined by compiler option -std=gnu99.

Tips

Before setting this parameter, check that your compiler supports the library you want to use. If you select a parameter value that your compiler does not support, compiler errors can occur.

Command-Line Information

Parameter: CodeReplacementLibrary
Type: character vector
Value: 'None' | 'GNU99 (GNU)'
Default: 'None'

See Also

More About

- "Simulink Desktop Real-Time Pane" on page 3-2

Shared code placement

Specify the location for generating shared utility code.

Settings

Default: Auto

Auto

Operates as follows:

- When the model contains Model blocks, place utility code within the slprj/target/ _sharedutils folder.

- When the model does not contain Model blocks, place utility code in the build folder (generally, in *model*.c or *model*.cpp).

Shared location

Directs code for utilities to be placed within the slprj folder in your working folder.

Command-Line Information

Parameter: UtilityFuncGeneration
Type: character vector
Value: 'Auto' | 'Shared location'
Default: 'Auto'

See Also

More About

- "Simulink Desktop Real-Time Pane" on page 3-2

Generate assembly listings

Instruct the C compiler to generate assembly listings for the generated code.

Settings

Default: off

☑ On

Generates assembly listings.

☐ Off

Suppresses assembly listings.

Command-Line Information
Parameter: `CCListing`
Type: Boolean
Value: `on` | `off`
Default: `off`

See Also

More About
•   "Simulink Desktop Real-Time Pane" on page 3-2

Rebuild all

Force the object files to be rebuilt regardless of their timestamps.

Settings

Default: off

☑ On

Rebuilds the object files at every build.

☐ Off

Rebuilds object files whose time stamps show they are outdated.

Command-Line Information
Parameter: RebuildAll
Type: Boolean
Value: on | off
Default: off

See Also

More About
• "Simulink Desktop Real-Time Pane" on page 3-2

External mode

Enable client/server communication between Simulink software and an application.

Settings

Default: on

☑ On

Enables external mode.

☐ Off

Disables external mode

Tip

- This check box is selected by default and grayed out.
- Code generation is only supported in external mode.

Dependencies

Selecting **External mode** enables:

- **Transport layer**
- **MEX-file arguments**
- **Static memory allocation**

Command-Line Information
Parameter: ExtMode
Type: character vector
Value: 'on' | 'off'
Default: 'on'

See Also

More About
- "Simulink Desktop Real-Time Pane" on page 3-2
- "Create a Transport Layer for TCP/IP or Serial External Mode Communication" (Simulink Coder)

Transport layer

Specify the transport protocol for external mode communication.

Settings

Default: sharedmem

sharedmem

Uses a shared memory transport mechanism. The MEX-file name is sldrtext.

Tips

*   The MEX-file name displayed next to **Transport layer** cannot be edited in the Configuration Parameters dialog box.
*   For targets provided by MathWorks®, like Simulink Desktop Real-Time, the MEX-file name is specified in:

    matlabroot/toolbox/simulink/simulink/extmode_transports.m

Dependency

This parameter is enabled by checking **External mode**.

Command-Line Information
Parameter: ExtModeTransport
Type: integer
Value: 0
Default: 0

See Also

More About
*   "Simulink Desktop Real-Time Pane" on page 3-2
*   "External Mode Simulation with TCP/IP or Serial Communication" (Simulink Coder)
*   "Create a Transport Layer for TCP/IP or Serial External Mode Communication" (Simulink Coder)

MEX-file arguments

Specify external mode MEX arguments.

Settings

Default: ''

For a shared memory transport, `sldrtext` allows two optional positional arguments. If the second argument appears, the first argument must appear.

- An argument that is unused. Specify '' for this argument.
- An argument that specifies verbosity. Specify `0` for Nonverbose, or `1` for Verbose.

The default is Nonverbose mode (`0`). To specify Verbose mode, set **MEX-file arguments** to:

`'' 1`

where `''` is the empty character vector and `1` specifies Verbose mode.

Dependency

This parameter is enabled by checking **External mode**.

Command-Line Information
Parameter: `ExtModeMexArgs`
Type: character vector followed by integer
Value: `''` | `'' 0` | `'' 1`
Default: `''`

See Also

More About
- "Simulink Desktop Real-Time Pane" on page 3-2
- "External Mode Simulation with TCP/IP or Serial Communication" (Simulink Coder)
- "Choose Communication Protocol for Client and Server" (Simulink Coder)