## Contents

1 File Index  
   1.1 File List ................................................................. 1

2 File Documentation  
   2.1 seasim.h File Reference .............................................. 3  
   2.1.1 Detailed Description .................................................. 11
Chapter 1

File Index

1.1 File List

Here is a list of all documented files with brief descriptions:

- **seasim.h**
  API for simulators and test code. Provides functions to simulate Arduino pin input and means to check Arduino output pins .......................... 3
Chapter 2

File Documentation

2.1 seasim.h File Reference

API for simulators and test code. Provides functions to simulate Arduino pin input and means to check Arduino output pins.

```c
#include "searduino.h"
#include "searduino_pin.h"
#include "setup.h"
#include "arduino/ext_io.h"
#include <stdio.h>
#include <string.h>
#include <unistd.h>
#include <signal.h>
```

Functions

**seasim_fake_digital_input**

Fakes input on an Arduino digital pin

This function should be used if you want to simulate input on the Arduino boards. The corresponds to actually setting the input manually (e.g using power supply) an input pin on the Arduino.

Parameters

- `uint8_t pin` Digital pin to fake input on
- `uint8_t value` Value (0 or 1) to fake

Return values

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>on success</td>
</tr>
<tr>
<td>!0</td>
<td>on failure</td>
</tr>
</tbody>
</table>

Example Usage:

```c
seasim_fake_digital_input (7, 0);
```

- `uint8_t seasim_fake_digital_input (uint8_t pin, uint8_t val)`

**seasim_fake_analog_input**

Fakes input on an Arduino digital pin

This function should be used if you want to simulate input on the Arduino boards. The corresponds to actually setting the input manually (e.g using power supply) an input pin on the Arduino.
**seasim_fake_digital_input**

*Fakes input on a general Arduino pin*

This function should be used if you want to simulate input on the Arduino boards. The corresponds to actually setting the input manually (e.g using power supply) an input pin on the Arduino.

**Parameters**

- **[uint8_t]** pin Analog pin to fake input on
- **[uint8_t]** value Value (0 - 1024) to fake

**Return values**

- **0** on success
- **0x0** on failure

**Example Usage:**

```c
seasim_fake_digital_input (8, 786);
```

- `uint8_t seasim_fake_digital_input (uint8_t pin, unsigned int val)`

**seasim_fake_input**

*Fakes input on a general Arduino pin*

This function should be used if you want to simulate input on the Arduino boards. The corresponds to actually setting the input manually (e.g using power supply) an input pin on the Arduino.

**Parameters**

- **[uint8_t]** pin Analog or Digital pin to fake input on
- **[uint8_t]** value Value (0 - 1024) to fake
- **[uint8_t]** type type of pin (SEARDUINO_PIN_TYPE_DIGITAL, SEARDUINO_PIN_TYPE_ANALOG)

**Return values**

- **0** on success
- **0x0** on failure

**Example Usage:**

```c
seasim_fake_input (8, 786, SEARDUINO_PIN_TYPE_ANALOG);
seasim_fake_input (13, 1, SEARDUINO_PIN_TYPE_DIGITAL);
```

- `uint8_t seasim_fake_input (uint8_t pin, unsigned int val, uint8_t type)`

**seasim_get_output**

*Returns output from a general Arduino pin*

This function should be used if you want to check the current output on the an Arduino board pin. The output is set by the application running in the simulated environment (the code you’ll put on the Arduino board).

**Parameters**

- **[uint8_t]** pin Analog or Digital pin to fake input on
- **[uint8_t]** type type of pin (SEARDUINO_PIN_TYPE_DIGITAL, SEARDUINO_PIN_TYPE_ANALOG)

**Return values**

- **[int]** the value of the pin

**Example Usage:**

```c
seasim_get_output (8, SEARDUINO_PIN_TYPE_ANALOG);
seasim_get_output (13, SEARDUINO_PIN_TYPE_DIGITAL);
```

- `int seasim_get_output (uint8_t pin, uint8_t pin_type)`

**seasim_is_pausersal**

*Check if you can pause the Arduino code*

Searduino provides some macros allowing the simulator to pause the code.

**Example Usage:**

```c
seasim_is_pausersal (13, SEARDUINO_PIN_TYPE_ANALOG);
```

- `int seasim_is_pausersal (uint8_t pin, uint8_t pin_type)`
Return values

uint8_t seasim_is_pausable (void)

seasim_is_paused
Check if you Arduino code is halted (stopped)

Return values

uint8_t seasim_is_paused (void)
uint8_t seasim_is_halted (void)

seasim_is_running
Check if you Arduino code is running

Return values

uint8_t seasim_is_running (void)

seasim_set_paused
Pauses the Arduino code
Searduino provides some macros allowing the simulator to pause the code. If you're not using these this function has no impact on the running Arduino code.

Return values

void seasim_set_paused (void)

seasim_set_running
Starts the Arduino code
Searduino provides some macros allowing the simulator to pause the code. If you're not using these this function has no impact on the running Arduino code.

Return values

void seasim_set_running (void)

seasim_set_halted
Halts the Arduino code
Searduino provides some macros allowing the simulator to pause the code. If you're not using these this function has no impact on the running Arduino code.

Return values

void seasim_set_halted (void)

seasim_get_dig_mode
Returns the mode from a digital Arduino pin
Parameters

[uint8_t]  pin  Analog or Digital pin to fake input on

Return values

[int]  the mode

Example Usage:

seasim_get_dig_mode (13);

- uint8_t seasim_get_dig_mode (uint8_t pin)

seasim_get_arduino_code_name

Returns the name of the Arduino code (file name) loaded

The returned string must not be freed. If you need to modify it you must copy the string.

Return values

[char∗]  the name of the Arduino code running

Example Usage:

char *arduino_code;
arduino_code = seasim_get_arduino_code_name();

- char * seasim_get_arduino_code_name (void)

seasim_set_arduino_code_name

Sets the name of the Arduino code (file name) loaded

Parameters

[const char∗]  libname  The file name of the shared library (including path) you wish to load for later execution.

Return values

[char∗]  the name of the Arduino code running

Example Usage:

int ret;
ret = seasim_set_arduino_code_name("./my-ard-code.so");

- int seasim_set_arduino_code_name (const char *libname)

seasim_register_dig_mode_simCb

Register a callback for changes in digital modes of pin

Whenever a pin changes mode on the Arduino you'll get noticed via your registered callback.

Parameters

[dm_to_sim - callback_ptr]  cb  The callback

Return values

[uint8_t]  0 on success. Non zero otherwise.

Example Usage:
```cpp
void
my_dm_sim_callback(uint8_t pin, uint8_t mode)
{
    fprintf(stdout,"Pin %d changed mode to %d\n",pin, mode);
}
.....
int ret;
ret = seasim_register_dig_mode_sim_cb(my_dm_sim_callback);

• uint8_t seasim_register_dig_mode_sim_cb (dm_to_sim_callback_ptr cb)

seasim_register_out_sim_cb
Register a callback for updates on the pin outputs
Whenever an output pin changes value on the Arduino you’ll get noticed via your registered callback.

Parameters
[out_to_sim_callback_ptr] cb The callback

Return values
[uint8_t] 0 on success. Non zero otherwise.

Example Usage:
```
• uint8_t seasim_register_log_cb (log_to_sim_callback_ptr cb)

seasim_register_type_cb
Register a callback for type changes on a pin
Setup all functionality of the Searduino simulator part. You can call this function many times - the setup is only done once.

Return values
  [uint8_t] 0 on success. Non zero otherwise.

• uint8_t seasim_register_type_cb (pintype_to_sim_callback_ptr cb)
• int seasim_setup (void)

seasim_disable_streamed_output
Turn off printouts on terminal
Apart from using callbacks to notify simulators/tests of updates Searduino also prints to stdout to inform of changes. This function disables those printouts.
The printout is enabled by default.

• void seasim_disable_streamed_output (void)

seasim_is_enable_streamed_output
Cheks if streamed output is enabled or disabled
Apart from using callbacks to notify simulators/tests of updates Searduino also prints to stdout to inform of changes. This function checks if such printout is enabled or disabled.

Return values
  [int] 1 if enabled. 0 if not.

• uint8_t seasim_is_enable_streamed_output (void)

seasim_enable_streamed_output
Turn on printouts on terminal
Apart from using callbacks to notify simulators/tests of updates Searduino also prints to stdout to inform of changes. This function enables those printouts.
The printout is enabled by default.

• void seasim_enable_streamed_output (void)

seasim_set_write_timelimit
Get the maximum update frequency for a pin
If pins get updated to fast you can limit the speed. This function returns the current update limit.

Return values
  [uint8_t] lim The current limit.

• void seasim_set_write_timelimit (unsigned int lim)
• unsigned int seasim_get_write_timelimit (void)
• int seasim_i2c_add_device (unsigned int device_nr, const char *setup_fun)

seasim_get_searduino_version
Returns the version of Searduino (as a string)
seasim.h File Reference

Return values

<table>
<thead>
<tr>
<th>char*</th>
<th>the version of Searduino</th>
</tr>
</thead>
</table>

- const char * **seasim_get_searduino_version** (void)

**seasim_get_searduino_name**

*Returns the name of Searduino (as a string)*

Return values

<table>
<thead>
<tr>
<th>char*</th>
<th>the name of Searduino</th>
</tr>
</thead>
</table>

- const char * **seasim_get_searduino_name** (void)

**seasim_set_board_name**

*Sets the name of board to use*

*If you’re setting the name to a board not supported by Searduino the last set board will be kept.*

Parameters

| char* | the name of board to use. If NULL Searduino sets no board in use. |

Return values

| 0 | on success |
| 10 | on failure |

- int **seasim_set_board_name** (char *board)

**seasim_get_board_name**

*Returns the name of currently used board*

Return values

| char* | the name of board in use. NULL if no board has been set. |

- char * **seasim_get_board_name** (void)

**seasim_get_current_pin_type**

*Returns the pin type of a pin*

Parameters

| uint8_t | pin Pin to check |

Return values

| int | the type of the pin |

- int **seasim_get_current_pin_type** (uint8_t pin)

**seasim_has_generic_pin_type**

*Checks if a pin can be of a certain type*

Parameters

| uint8_t | pin Pin to check |


Return values

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>if it can not</td>
</tr>
<tr>
<td>1</td>
<td>if it can</td>
</tr>
</tbody>
</table>

* int `seasim_has_generic_pin_type` (uint8_t pin, uint8_t type)

**seasim_hid_disable_feedback**
* Turn off faked hid
* Searduino can on some platforms move the mouse and keyboard just as Arduino code connected to a computer would do. This functions disables that feature.

Return values

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>on success</td>
</tr>
<tr>
<td>1</td>
<td>on failure</td>
</tr>
</tbody>
</table>

* uint8_t `seasim_hid_disable_feedback` (void)

**seasim_hid_enable_feedback**
* Turn off faked hid
* Searduino can on some platforms move the mouse and keyboard just as Arduino code connected to a computer would do. This functions enables that feature.

Return values

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>on success</td>
</tr>
<tr>
<td>1</td>
<td>on failure</td>
</tr>
</tbody>
</table>

* uint8_t `seasim_hid_enable_feedback` (void)

**seasim_get_supported_boards**
* Returns a string with all boards supported by Searduino

Return values

<p>| |</p>
<table>
<thead>
<tr>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>[char ∗] Returns an empty string if no boards are supported. Otherwise it returns a comma separated list of boards</td>
</tr>
</tbody>
</table>

* char ∗ `seasim_get_supported_boards` (void)

**seasim_get_nr_of_pins**
* Returns the number of pins of the current board

Return values

<p>| |</p>
<table>
<thead>
<tr>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>[uint8_t] The number of pins on the currently chose board.</td>
</tr>
</tbody>
</table>

* uint8_t `seasim_get_nr_of_pins` (void)

**Variables**

* searduino_main_ptr_ptr `searduino_main_entry`
2.1.1 Detailed Description

API for simulators and test code. Provides functions to simulate Arduino pin input and means to check Arduino output pins. Henrik Sandklef

Date
January 2013

Definition in file seasim.h.
Index

seasim.h, 3