

## My Project

Generated by Doxygen 1.8.14



# Contents

<b>1</b>	<b>Namespace Index</b>	<b>1</b>
1.1	Namespace List . . . . .	1
<b>2</b>	<b>Hierarchical Index</b>	<b>3</b>
2.1	Class Hierarchy . . . . .	3
<b>3</b>	<b>Class Index</b>	<b>5</b>
3.1	Class List . . . . .	5
<b>4</b>	<b>File Index</b>	<b>7</b>
4.1	File List . . . . .	7
<b>5</b>	<b>Namespace Documentation</b>	<b>9</b>
5.1	simlib Namespace Reference . . . . .	9
5.2	simlib.action Namespace Reference . . . . .	9
5.2.1	Detailed Description . . . . .	10
5.2.2	Function Documentation . . . . .	10
5.2.2.1	fn() . . . . .	10
5.2.2.2	fn2() . . . . .	10
5.2.3	Variable Documentation . . . . .	10
5.2.3.1	errors . . . . .	10
5.2.3.2	test_obj . . . . .	10
5.3	simlib.anchor Namespace Reference . . . . .	10
5.4	simlib.archspec Namespace Reference . . . . .	11
5.4.1	Detailed Description . . . . .	11

5.4.2	Variable Documentation	11
5.4.2.1	archspec_obj	11
5.4.2.2	errors	11
5.5	simlib.FSM Namespace Reference	11
5.5.1	Function Documentation	12
5.5.1.1	getBattery()	12
5.5.1.2	getDeepSleepCurrent()	13
5.5.1.3	getIdleCurrent()	13
5.5.1.4	getInitCurrent()	13
5.5.1.5	getOffCurrent()	13
5.5.1.6	getRXCurrent()	13
5.5.1.7	getSleepCurrent()	13
5.5.1.8	getTime()	14
5.5.1.9	getTXCurrent()	14
5.5.2	Variable Documentation	14
5.5.2.1	AVAILABLE_STATES	14
5.5.2.2	battlife	14
5.5.2.3	DEBUG	14
5.5.2.4	DEEPSLEEP_STATE	14
5.5.2.5	device	15
5.5.2.6	DEVICE_DATA	15
5.5.2.7	errors	15
5.5.2.8	IDLE_STATE	15
5.5.2.9	INIT_STATE	15
5.5.2.10	INITIAL_STATE	15
5.5.2.11	my_current	15
5.5.2.12	OFF_STATE	16
5.5.2.13	OSC_STARTUP_SEND_CURRENT	16
5.5.2.14	OSC_STARTUP_SEND_TIME	16
5.5.2.15	PLL_STARTUP_SEND_CURRENT	16

5.5.2.16	PLL_STARTUP_SEND_TIME . . . . .	16
5.5.2.17	RX_STATE . . . . .	16
5.5.2.18	simlist . . . . .	16
5.5.2.19	SLEEP_STATE . . . . .	17
5.5.2.20	TX_PHR_SEND_CURRENT . . . . .	17
5.5.2.21	TX_PHR_SEND_TIME . . . . .	17
5.5.2.22	TX_SHR_SEND_CURRENT . . . . .	17
5.5.2.23	TX_SHR_SEND_TIME . . . . .	17
5.5.2.24	TX_STATE . . . . .	17
5.5.2.25	WR_TX_DATA_SEND_CURRENT . . . . .	17
5.5.2.26	WR_TX_DATA_SEND_TIME . . . . .	18
5.6	simlib.hub Namespace Reference . . . . .	18
5.6.1	Variable Documentation . . . . .	18
5.6.1.1	hub . . . . .	18
5.7	simlib.node Namespace Reference . . . . .	18
5.8	simlib.simulated Namespace Reference . . . . .	18
5.8.1	Variable Documentation . . . . .	19
5.8.1.1	action1 . . . . .	19
5.8.1.2	action2 . . . . .	19
5.8.1.3	action3 . . . . .	19
5.8.1.4	aq . . . . .	19
5.8.1.5	DEBUG . . . . .	19
5.8.1.6	errors . . . . .	19
5.8.1.7	i . . . . .	19
5.9	simlib.simulationenvironment Namespace Reference . . . . .	20
5.9.1	Function Documentation . . . . .	20
5.9.1.1	superSmartTrilaterationAlgorithm() . . . . .	20
5.9.2	Variable Documentation . . . . .	20
5.9.2.1	anchor . . . . .	20
5.9.2.2	archspeg . . . . .	20
5.9.2.3	errors . . . . .	21
5.9.2.4	hub . . . . .	21
5.9.2.5	node . . . . .	21
5.9.2.6	simEnv . . . . .	21

<b>6</b>	<b>Class Documentation</b>	<b>23</b>
6.1	simlib.action.Action Class Reference	23
6.1.1	Detailed Description	23
6.1.2	Constructor & Destructor Documentation	23
6.1.2.1	__init__()	23
6.1.3	Member Function Documentation	24
6.1.3.1	decrement()	24
6.1.3.2	get_args()	24
6.1.3.3	get_ctr()	24
6.1.3.4	get_fn()	24
6.1.3.5	set_ctr()	25
6.1.3.6	set_fn()	25
6.2	simlib.simulated.ActionQueue Class Reference	25
6.2.1	Detailed Description	25
6.2.2	Constructor & Destructor Documentation	25
6.2.2.1	__init__()	25
6.2.3	Member Function Documentation	26
6.2.3.1	addToQueue()	26
6.2.3.2	popAction()	26
6.2.3.3	test()	26
6.2.3.4	update()	26
6.2.4	Member Data Documentation	26
6.2.4.1	queue	27
6.3	simlib.anchor.Anchor Class Reference	27
6.3.1	Constructor & Destructor Documentation	27
6.3.1.1	__init__()	28
6.3.2	Member Function Documentation	28
6.3.2.1	addAction()	28
6.3.2.2	getID()	28
6.3.2.3	listenForSignal()	28

6.3.2.4	<a href="#">pingNode()</a>	28
6.3.2.5	<a href="#">prependAction()</a>	29
6.3.2.6	<a href="#">requestPing()</a>	29
6.3.2.7	<a href="#">setTime()</a>	29
6.3.2.8	<a href="#">waitForReply()</a>	29
6.3.3	<a href="#">Member Data Documentation</a>	29
6.3.3.1	<a href="#">distance</a>	29
6.3.3.2	<a href="#">ID</a>	29
6.3.3.3	<a href="#">measurement</a>	30
6.3.3.4	<a href="#">pingedNodeID</a>	30
6.3.3.5	<a href="#">signalList</a>	30
6.3.3.6	<a href="#">time</a>	30
6.3.3.7	<a href="#">xPos</a>	30
6.3.3.8	<a href="#">yPos</a>	30
6.3.3.9	<a href="#">zPos</a>	30
6.4	<a href="#">simlib.archspec.ArchSpec Class Reference</a>	31
6.4.1	<a href="#">Detailed Description</a>	31
6.4.2	<a href="#">Constructor &amp; Destructor Documentation</a>	31
6.4.2.1	<a href="#">__init__()</a>	31
6.4.3	<a href="#">Member Function Documentation</a>	31
6.4.3.1	<a href="#">get_anchorclass()</a>	31
6.4.3.2	<a href="#">get_hubclass()</a>	32
6.4.3.3	<a href="#">get_nodeclass()</a>	32
6.4.3.4	<a href="#">set_anchorclass()</a>	32
6.4.3.5	<a href="#">set_hubclass()</a>	32
6.4.3.6	<a href="#">set_nodeclass()</a>	32
6.5	<a href="#">simlib.FSM.Device Class Reference</a>	33
6.5.1	<a href="#">Detailed Description</a>	33
6.5.2	<a href="#">Constructor &amp; Destructor Documentation</a>	33
6.5.2.1	<a href="#">__init__()</a>	33

6.5.3	Member Function Documentation . . . . .	34
6.5.3.1	getParam() . . . . .	34
6.5.3.2	getState() . . . . .	34
6.5.3.3	setNextState() . . . . .	34
6.5.4	Member Data Documentation . . . . .	34
6.5.4.1	available_states . . . . .	34
6.5.4.2	dev_state . . . . .	34
6.5.4.3	initial_state . . . . .	35
6.5.4.4	next_states . . . . .	35
6.5.4.5	physical_data . . . . .	35
6.6	simlib.FSM.DW1000 Class Reference . . . . .	35
6.6.1	Constructor & Destructor Documentation . . . . .	35
6.6.1.1	__init__() . . . . .	36
6.6.2	Member Function Documentation . . . . .	36
6.6.2.1	mainloop() . . . . .	36
6.7	simlib.hub.Hub Class Reference . . . . .	36
6.7.1	Constructor & Destructor Documentation . . . . .	37
6.7.1.1	__init__() . . . . .	37
6.7.2	Member Function Documentation . . . . .	37
6.7.2.1	addAction() . . . . .	37
6.7.2.2	addAnchor() . . . . .	37
6.7.2.3	addNode() . . . . .	37
6.7.2.4	containsAnchor() . . . . .	38
6.7.2.5	containsNode() . . . . .	38
6.7.2.6	generateCompleteMap() . . . . .	38
6.7.2.7	getNodePosition() . . . . .	38
6.7.2.8	mainloop() . . . . .	38
6.7.2.9	mapAnchorAndNode() . . . . .	38
6.7.2.10	mapDistance() . . . . .	39
6.7.2.11	prependAction() . . . . .	39



6.7.2.12	<a href="#">removeAnchor()</a>	39
6.7.2.13	<a href="#">removeNode()</a>	39
6.7.2.14	<a href="#">resetMap()</a>	39
6.7.2.15	<a href="#">setTime()</a>	39
6.7.2.16	<a href="#">triliterateNode()</a>	40
6.7.3	<a href="#">Member Data Documentation</a>	40
6.7.3.1	<a href="#">algorithm</a>	40
6.7.3.2	<a href="#">anchors</a>	40
6.7.3.3	<a href="#">map</a>	40
6.7.3.4	<a href="#">nodePositions</a>	40
6.7.3.5	<a href="#">nodes</a>	40
6.7.3.6	<a href="#">time</a>	40
6.8	<a href="#">simlib.archspec.MySubclass Class Reference</a>	41
6.9	<a href="#">simlib.node.Node Class Reference</a>	41
6.9.1	<a href="#">Constructor &amp; Destructor Documentation</a>	42
6.9.1.1	<a href="#">__init__()</a>	42
6.9.2	<a href="#">Member Function Documentation</a>	42
6.9.2.1	<a href="#">addAction()</a>	42
6.9.2.2	<a href="#">getID()</a>	42
6.9.2.3	<a href="#">listenForSignal()</a>	42
6.9.2.4	<a href="#">prependAction()</a>	42
6.9.2.5	<a href="#">setTime()</a>	43
6.9.3	<a href="#">Member Data Documentation</a>	43
6.9.3.1	<a href="#">ID</a>	43
6.9.3.2	<a href="#">signalList</a>	43
6.9.3.3	<a href="#">time</a>	43
6.9.3.4	<a href="#">xPos</a>	43
6.9.3.5	<a href="#">xVel</a>	43
6.9.3.6	<a href="#">yPos</a>	43
6.9.3.7	<a href="#">yVel</a>	44

6.9.3.8	zPos	44
6.9.3.9	zVel	44
6.10	simlib.simulated.Simulated Class Reference	44
6.10.1	Detailed Description	45
6.10.2	Constructor & Destructor Documentation	45
6.10.2.1	__init__()	45
6.10.3	Member Function Documentation	45
6.10.3.1	mainloop()	45
6.10.3.2	run_timestep()	45
6.10.4	Member Data Documentation	45
6.10.4.1	actionQueue	45
6.11	simlib.simulationenvironment.SimulationEnvironment Class Reference	46
6.11.1	Constructor & Destructor Documentation	46
6.11.1.1	__init__()	46
6.11.2	Member Function Documentation	46
6.11.2.1	associateAnchor()	46
6.11.2.2	associateNode()	47
6.11.2.3	createAnchor()	47
6.11.2.4	createHub()	47
6.11.2.5	createNode()	47
6.11.2.6	deleteAnchor()	47
6.11.2.7	deleteNode()	47
6.11.2.8	dissassociateAnchor()	48
6.11.2.9	dissassociateNode()	48
6.11.2.10	getAnchorByID()	48
6.11.2.11	getNodeByID()	48
6.11.2.12	mainloop()	48
6.11.3	Member Data Documentation	48
6.11.3.1	anchors	48
6.11.3.2	hubs	49
6.11.3.3	nextID	49
6.11.3.4	nodes	49
6.11.3.5	signalList	49
6.11.3.6	time	49
6.12	simlib.FSM.State Class Reference	49
6.12.1	Detailed Description	50
6.12.2	Constructor & Destructor Documentation	50
6.12.2.1	__init__()	50
6.12.3	Member Function Documentation	50
6.12.3.1	getParam()	50
6.12.4	Member Data Documentation	50
6.12.4.1	physical_data	50

<b>7 File Documentation</b>	<b>51</b>
7.1 <code>__init__.py</code> File Reference . . . . .	51
7.2 <code>action.py</code> File Reference . . . . .	51
7.3 <code>anchor.py</code> File Reference . . . . .	51
7.4 <code>archspec.py</code> File Reference . . . . .	52
7.5 <code>FSM.py</code> File Reference . . . . .	52
7.6 <code>hub.py</code> File Reference . . . . .	53
7.7 <code>node.py</code> File Reference . . . . .	53
7.8 <code>simulated.py</code> File Reference . . . . .	54
7.9 <code>simulationenvironment.py</code> File Reference . . . . .	54
<b>Index</b>	<b>55</b>



# Chapter 1

## Namespace Index

### 1.1 Namespace List

Here is a list of all namespaces with brief descriptions:

<a href="#">simlib</a>	9
<a href="#">simlib.action</a>	9
<a href="#">simlib.anchor</a>	10
<a href="#">simlib.archspec</a>	11
<a href="#">simlib.FSM</a>	11
<a href="#">simlib.hub</a>	18
<a href="#">simlib.node</a>	18
<a href="#">simlib.simulated</a>	18
<a href="#">simlib.simulationenvironment</a>	20



## Chapter 2

# Hierarchical Index

### 2.1 Class Hierarchy

This inheritance list is sorted roughly, but not completely, alphabetically:

simlib.action.Action . . . . .	23
simlib.simulated.ActionQueue . . . . .	25
simlib.archspec.ArchSpec . . . . .	31
simlib.simulationenvironment.SimulationEnvironment . . . . .	46
simlib.FSM.State . . . . .	49
ABC	
simlib.FSM.Device . . . . .	33
simlib.anchor.Anchor . . . . .	27
simlib.FSM.DW1000 . . . . .	35
simlib.node.Node . . . . .	41
simlib.simulated.Simulated . . . . .	44
simlib.archspec.MySubclass . . . . .	41
simlib.FSM.Device . . . . .	33
simlib.hub.Hub . . . . .	36





## Chapter 3

# Class Index

### 3.1 Class List

Here are the classes, structs, unions and interfaces with brief descriptions:

<a href="#">simlib.action.Action</a>	
Classes ##	23
<a href="#">simlib.simulated.ActionQueue</a>	25
<a href="#">simlib.anchor.Anchor</a>	27
<a href="#">simlib.archspec.ArchSpec</a>	
Classes ##	31
<a href="#">simlib.FSM.Device</a>	33
<a href="#">simlib.FSM.DW1000</a>	35
<a href="#">simlib.hub.Hub</a>	36
<a href="#">simlib.archspec.MySubclass</a>	41
<a href="#">simlib.node.Node</a>	41
<a href="#">simlib.simulated.Simulated</a>	
Classes ##	44
<a href="#">simlib.simulationenvironment.SimulationEnvironment</a>	46
<a href="#">simlib.FSM.State</a>	49



## Chapter 4

# File Index

### 4.1 File List

Here is a list of all files with brief descriptions:

<a href="#">__init__.py</a>	51
<a href="#">action.py</a>	51
<a href="#">anchor.py</a>	51
<a href="#">archspec.py</a>	52
<a href="#">FSM.py</a>	52
<a href="#">hub.py</a>	53
<a href="#">node.py</a>	53
<a href="#">simulated.py</a>	54
<a href="#">simulationenvironment.py</a>	54



## Chapter 5

# Namespace Documentation

### 5.1 simlib Namespace Reference

#### Namespaces

- [action](#)
- [anchor](#)
- [archspec](#)
- [FSM](#)
- [hub](#)
- [node](#)
- [simulated](#)
- [simulationenvironment](#)

### 5.2 simlib.action Namespace Reference

#### Classes

- class [Action](#)  
*Classes ##.*

#### Functions

- def [fn](#) (n)
- def [fn2](#) (x, y)

#### Variables

- int [errors](#) = 0
- [test\\_obj](#) = None

### 5.2.1 Detailed Description

@package Action

This module defines the Action object, which is used, together with the ActionQueue class, to implement delayed function calls.

### 5.2.2 Function Documentation

#### 5.2.2.1 fn()

```
def simlib.action.fn (  
    n )
```

#### 5.2.2.2 fn2()

```
def simlib.action.fn2 (  
    x,  
    y )
```

### 5.2.3 Variable Documentation

#### 5.2.3.1 errors

```
int simlib.action.errors = 0
```

#### 5.2.3.2 test\_obj

```
simlib.action.test_obj = None
```

## 5.3 simlib.anchor Namespace Reference

### Classes

- class [Anchor](#)

## 5.4 simlib.archspec Namespace Reference

### Classes

- class [ArchSpec](#)  
*Classes ##.*
- class [MySubclass](#)

### Variables

- int [errors](#) = 0
- [archspec\\_obj](#) = [ArchSpec](#)( int , int , int )

#### 5.4.1 Detailed Description

@package Architecture Specification

This module presents the simulation environment with the types of the hub, nodes, and anchors.

#### 5.4.2 Variable Documentation

##### 5.4.2.1 archspec\_obj

```
simlib.archspec.archspec_obj = ArchSpec( int , int , int )
```

##### 5.4.2.2 errors

```
int simlib.archspec.errors = 0
```

## 5.5 simlib.FSM Namespace Reference

### Classes

- class [Device](#)
- class [DW1000](#)
- class [State](#)

## Functions

- def `getOffCurrent` (physical\_data)  
*This is dumb but i cant think of a way to do this while making it # general to each device.*
- def `getDeepSleepCurrent` (physical\_data)
- def `getSleepCurrent` (physical\_data)
- def `getInitCurrent` (physical\_data)
- def `getIdleCurrent` (physical\_data)
- def `getRXCurrent` (physical\_data)
- def `getTXCurrent` (physical\_data)
- def `getTime` (physical\_data)
- def `getBattery` (dev\_data, state\_data)

## Variables

- int `OSC_STARTUP_SEND_CURRENT` = 3
- int `OSC_STARTUP_SEND_TIME` = 2000
- int `PLL_STARTUP_SEND_CURRENT` = 12
- int `PLL_STARTUP_SEND_TIME` = 7
- int `WR_TX_DATA_SEND_CURRENT` = 15
- int `WR_TX_DATA_SEND_TIME` = 10
- int `TX_SHR_SEND_CURRENT` = 65
- int `TX_SHR_SEND_TIME` = 135
- int `TX_PHR_SEND_CURRENT` = 48
- float `TX_PHR_SEND_TIME` = 1.33
- bool `DEBUG` = True
- `OFF_STATE` = `State({'current' : getOffCurrent, 'time' : getTime, 'state' : 'off_state'})`  
*FSM for the DW1000, includes states, thier delays, and the current for each state.*
- `DEEPSLEEP_STATE` = `State({'current' : getDeepSleepCurrent, 'time' : getTime, 'state' : 'deepsleep_state'})`
- `SLEEP_STATE` = `State({'current' : getSleepCurrent, 'time' : getTime, 'state' : 'sleep_state'})`
- `INIT_STATE` = `State({'current' : getInitCurrent, 'time' : getTime, 'state' : 'init_state'})`
- `IDLE_STATE` = `State({'current' : getIdleCurrent, 'time' : getTime, 'state' : 'idle_state'})`
- `RX_STATE` = `State({'current' : getRXCurrent, 'time' : getTime, 'state' : 'rx_state'})`
- `TX_STATE` = `State({'current' : getTXCurrent, 'time' : getTime, 'state' : 'tx_state'})`
- `INITIAL_STATE` = `OFF_STATE`
- dictionary `DEVICE_DATA` = {'batt\_life' : `getBattery`}
- dictionary `AVAILABLE_STATES`
- `battlife`
- `device` = `DW1000(AVAILABLE_STATES, INITIAL_STATE, DEVICE_DATA)`
- int `errors` = 0
- list `simlist` = [ ]
- `my_current` = `device.getState().getParam('current')`

### 5.5.1 Function Documentation

#### 5.5.1.1 `getBattery()`

```
def simlib.FSM.getBattery (
    dev_data,
    state_data )
```



#### 5.5.1.2 getDeepSleepCurrent()

```
def simlib.FSM.getDeepSleepCurrent (
    physical_data )
```

#### 5.5.1.3 getIdleCurrent()

```
def simlib.FSM.getIdleCurrent (
    physical_data )
```

#### 5.5.1.4 getInitCurrent()

```
def simlib.FSM.getInitCurrent (
    physical_data )
```

#### 5.5.1.5 getOffCurrent()

```
def simlib.FSM.getOffCurrent (
    physical_data )
```

This is dumb but i cant think of a way to do this while making it # general to each device.

I won't be adding comments to these # functions because they will be replaced when a better solution # is found. #

#### 5.5.1.6 getRXCurrent()

```
def simlib.FSM.getRXCurrent (
    physical_data )
```

#### 5.5.1.7 getSleepCurrent()

```
def simlib.FSM.getSleepCurrent (
    physical_data )
```

#### 5.5.1.8 getTime()

```
def simlib.FSM.getTime (
    physical_data )
```

#### 5.5.1.9 getTXCurrent()

```
def simlib.FSM.getTXCurrent (
    physical_data )
```

### 5.5.2 Variable Documentation

#### 5.5.2.1 AVAILABLE\_STATES

```
dictionary simlib.FSM.AVAILABLE_STATES
```

##### Initial value:

```
1 = {OFF_STATE : [(INIT_STATE, 3000)],
2     DEEPSLEEP_STATE : [(INIT_STATE, 3000)],
3     SLEEP_STATE : [(INIT_STATE, 3000)],
4     INIT_STATE : [(IDLE_STATE, 5)],
5     IDLE_STATE : [(RX_STATE, 0) , (TX_STATE, 0)],
6     RX_STATE : [(IDLE_STATE, 0) , (SLEEP_STATE, 0) , (DEEPSLEEP_STATE, 0)],
7     TX_STATE : [(IDLE_STATE, 0) , (SLEEP_STATE, 0) , (DEEPSLEEP_STATE, 0)]}
```

#### 5.5.2.2 battlife

```
simlib.FSM.battlife
```

#### 5.5.2.3 DEBUG

```
bool simlib.FSM.DEBUG = True
```

#### 5.5.2.4 DEEPSLEEP\_STATE

```
simlib.FSM.DEEPSLEEP_STATE = State({'current' : getDeepSleepCurrent, 'time' : getTime, 'state'
: 'deepsleep_state'})
```

#### 5.5.2.5 device

```
simlib.FSM.device = DW1000(AVAILABLE_STATES, INITIAL_STATE, DEVICE_DATA)
```

#### 5.5.2.6 DEVICE\_DATA

```
dictionary simlib.FSM.DEVICE_DATA = {'batt_life' : getBattery}
```

#### 5.5.2.7 errors

```
int simlib.FSM.errors = 0
```

#### 5.5.2.8 IDLE\_STATE

```
simlib.FSM.IDLE_STATE = State({'current' : getIdleCurrent, 'time' : getTime, 'state' ←  
: 'idle_state'})
```

#### 5.5.2.9 INIT\_STATE

```
simlib.FSM.INIT_STATE = State({'current' : getInitCurrent, 'time' : getTime, 'state' ←  
: 'init_state'})
```

#### 5.5.2.10 INITIAL\_STATE

```
simlib.FSM.INITIAL_STATE = OFF_STATE
```

#### 5.5.2.11 my\_current

```
simlib.FSM.my_current = device.getState().getParam('current')
```

#### 5.5.2.12 OFF\_STATE

```
simlib.FSM.OFF_STATE = State({'current' : getOffCurrent, 'time' : getTime, 'state' : 'off←  
_state'})
```

FSM for the DW1000, includes states, thier delays, and the current for each state.

#### 5.5.2.13 OSC\_STARTUP\_SEND\_CURRENT

```
int simlib.FSM.OSC_STARTUP_SEND_CURRENT = 3
```

#### 5.5.2.14 OSC\_STARTUP\_SEND\_TIME

```
int simlib.FSM.OSC_STARTUP_SEND_TIME = 2000
```

#### 5.5.2.15 PLL\_STARTUP\_SEND\_CURRENT

```
int simlib.FSM.PLL_STARTUP_SEND_CURRENT = 12
```

#### 5.5.2.16 PLL\_STARTUP\_SEND\_TIME

```
int simlib.FSM.PLL_STARTUP_SEND_TIME = 7
```

#### 5.5.2.17 RX\_STATE

```
simlib.FSM.RX_STATE = State({'current' : getRXCurrent, 'time' : getTime, 'state' : 'rx←  
state'})
```

#### 5.5.2.18 simlist

```
list simlib.FSM.simlist = [ ]
```

#### 5.5.2.19 SLEEP\_STATE

```
simlib.FSM.SLEEP_STATE = State({'current' : getSleepCurrent, 'time' : getTime, 'state' :  
: 'sleep_state'})
```

#### 5.5.2.20 TX\_PHR\_SEND\_CURRENT

```
int simlib.FSM.TX_PHR_SEND_CURRENT = 48
```

#### 5.5.2.21 TX\_PHR\_SEND\_TIME

```
float simlib.FSM.TX_PHR_SEND_TIME = 1.33
```

#### 5.5.2.22 TX\_SHR\_SEND\_CURRENT

```
int simlib.FSM.TX_SHR_SEND_CURRENT = 65
```

#### 5.5.2.23 TX\_SHR\_SEND\_TIME

```
int simlib.FSM.TX_SHR_SEND_TIME = 135
```

#### 5.5.2.24 TX\_STATE

```
simlib.FSM.TX_STATE = State({'current' : getTXCurrent, 'time' : getTime, 'state' : 'tx_  
state'})
```

#### 5.5.2.25 WR\_TX\_DATA\_SEND\_CURRENT

```
int simlib.FSM.WR_TX_DATA_SEND_CURRENT = 15
```

### 5.5.2.26 WR\_TX\_DATA\_SEND\_TIME

```
int simlib.FSM.WR_TX_DATA_SEND_TIME = 10
```

## 5.6 simlib.hub Namespace Reference

### Classes

- class [Hub](#)

### Variables

- [hub](#) = [Hub](#)(0)

### 5.6.1 Variable Documentation

#### 5.6.1.1 hub

```
simlib.hub.hub = Hub(0)
```

## 5.7 simlib.node Namespace Reference

### Classes

- class [Node](#)

## 5.8 simlib.simulated Namespace Reference

### Classes

- class [ActionQueue](#)
- class [Simulated](#)

*Classes ##.*

### Variables

- bool [DEBUG](#) = True
- int [errors](#) = 0
- [aq](#) = [ActionQueue](#)( )
- [action1](#) = [Action](#)( aq.test, ["I was the first action added"] )
- [action2](#) = [Action](#)( aq.test, ["then me (:)"] )
- [action3](#) = [Action](#)( aq.test, ["lastly me (;"] )
- int [i](#) = 0

## 5.8.1 Variable Documentation

### 5.8.1.1 action1

```
simlib.simulated.action1 = Action( aq.test, ["I was the first action added"] )
```

### 5.8.1.2 action2

```
simlib.simulated.action2 = Action( aq.test, ["then me (:" ] )
```

### 5.8.1.3 action3

```
simlib.simulated.action3 = Action( aq.test, ["lastly me (;" ] )
```

### 5.8.1.4 aq

```
simlib.simulated.aq = ActionQueue( )
```

### 5.8.1.5 DEBUG

```
bool simlib.simulated.DEBUG = True
```

### 5.8.1.6 errors

```
int simlib.simulated.errors = 0
```

### 5.8.1.7 i

```
int simlib.simulated.i = 0
```

## 5.9 simlib.simulationenvironment Namespace Reference

### Classes

- class [SimulationEnvironment](#)

### Functions

- def [superSmartTrilaterationAlgorithm](#)

### Variables

- int [errors](#) = 0
- [archspec](#) = [ArchSpec](#)([Hub](#), [Anchor](#), [Node](#))
- [simEnv](#) = [SimulationEnvironment](#)()
- [hub](#) = [simEnv.createHub](#)([superSmartTrilaterationAlgorithm](#))
- [anchor](#) = [simEnv.createAnchor](#)(0, 0, i)
- [node](#) = [simEnv.createNode](#)(3, 3, 3, 0, 0, 0)

### 5.9.1 Function Documentation

#### 5.9.1.1 [superSmartTrilaterationAlgorithm\(\)](#)

```
def simlib.simulationenvironment.superSmartTrilaterationAlgorithm (  
    mapDict )
```

### 5.9.2 Variable Documentation

#### 5.9.2.1 [anchor](#)

```
simlib.simulationenvironment.anchor = simEnv.createAnchor(0, 0, i)
```

#### 5.9.2.2 [archspec](#)

```
simlib.simulationenvironment.archspec = ArchSpec(Hub, Anchor, Node)
```



### 5.9.2.3 errors

```
int simlib.simulationenvironment.errors = 0
```

### 5.9.2.4 hub

```
simlib.simulationenvironment.hub = simEnv.createHub(superSmartTrilaterationAlgorithm)
```

### 5.9.2.5 node

```
simlib.simulationenvironment.node = simEnv.createNode(3, 3, 3, 0, 0, 0)
```

### 5.9.2.6 simEnv

```
simlib.simulationenvironment.simEnv = SimulationEnvironment()
```



## Chapter 6

# Class Documentation

### 6.1 simlib.action.Action Class Reference

Classes ##.

#### Public Member Functions

- def `__init__`
- def `decrement`
- def `set_fn`
- def `set_ctr`
- def `get_fn` (self)
- def `get_args` (self)
- def `get_ctr` (self)

#### 6.1.1 Detailed Description

Classes ##.

Element of an action queue

#### 6.1.2 Constructor & Destructor Documentation

##### 6.1.2.1 `__init__()`

```
def simlib.action.Action.__init__ (
    self,
    fn_to_be_called )
```

## 6.1.3 Member Function Documentation

### 6.1.3.1 decrement()

```
def simlib.action.Action.decrement (
    self,
    n )
```

### 6.1.3.2 get\_args()

```
def simlib.action.Action.get_args (
    self,
    list )
```

Returns the function args associated with this action.

### 6.1.3.3 get\_ctr()

```
def simlib.action.Action.get_ctr (
    self,
    int )
```

Returns the counter value associated with this action.

### 6.1.3.4 get\_fn()

```
def simlib.action.Action.get_fn (
    self,
    types,
    FunctionType )
```

Returns the function associated with this action.

### 6.1.3.5 set\_ctr()

```
def simlib.action.Action.set_ctr (
    self,
    ctr_val )
```

### 6.1.3.6 set\_fn()

```
def simlib.action.Action.set_fn (
    self,
    fn_to_be_called )
```

The documentation for this class was generated from the following file:

- [action.py](#)

## 6.2 simlib.simulated.ActionQueue Class Reference

### Public Member Functions

- def [\\_\\_init\\_\\_](#) (self)
- def [addToQueue](#)
- def [popAction](#) (self)
- def [update](#) (self)
- def [test](#) (self, test)

### Public Attributes

- [queue](#)

### 6.2.1 Detailed Description

Creates instance of Action Queue

### 6.2.2 Constructor & Destructor Documentation

#### 6.2.2.1 \_\_init\_\_()

```
def simlib.simulated.ActionQueue.__init__ (
    self )
```

## 6.2.3 Member Function Documentation

### 6.2.3.1 addToQueue()

```
def simlib.simulated.ActionQueue.addToQueue (
    self,
    newAction )
```

### 6.2.3.2 popAction()

```
def simlib.simulated.ActionQueue.popAction (
    self )
```

removes the oldest Action from the list

### 6.2.3.3 test()

```
def simlib.simulated.ActionQueue.test (
    self,
    test )
```

arbitrary function added for testing purposes

### 6.2.3.4 update()

```
def simlib.simulated.ActionQueue.update (
    self )
```

Every time the update function is called, decrement the counter of the first action in the action queue. If the the counter is zero, pop the action off the action queue

## 6.2.4 Member Data Documentation

#### 6.2.4.1 queue

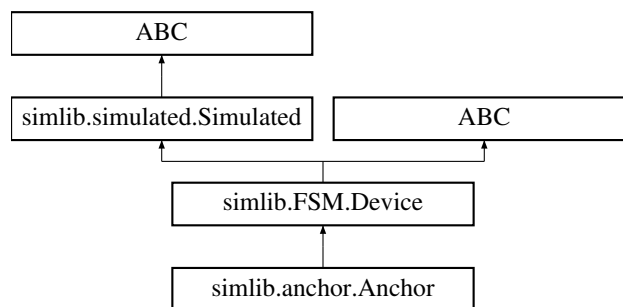
`simlib.simulated.ActionQueue.queue`

The documentation for this class was generated from the following file:

- [simulated.py](#)

## 6.3 simlib.anchor.Anchor Class Reference

Inheritance diagram for `simlib.anchor.Anchor`:



### Public Member Functions

- `def __init__`
- `def setTime`
- `def getID ()`
- `def listenForSignal ()`
- `def requestPing`
- `def pingNode`
- `def waitForReply`
- `def addAction`
- `def prependAction`

### Public Attributes

- `time`
- `ID`
- `xPos`
- `yPos`
- `zPos`
- `pingedNodeID`
- `measurement`
- `signalList`
- `distance`

#### 6.3.1 Constructor & Destructor Documentation

#### 6.3.1.1 `__init__()`

```
def simlib.anchor.Anchor.__init__ (
    self,
    ID )
```

### 6.3.2 Member Function Documentation

#### 6.3.2.1 `addAction()`

```
def simlib.anchor.Anchor.addAction (
    self,
    function )
```

#### 6.3.2.2 `getID()`

```
def simlib.anchor.Anchor.getID (
    int )
```

#### 6.3.2.3 `listenForSignal()`

```
def simlib.anchor.Anchor.listenForSignal (
    int )
```

Searches to see if it is the target of any signals. Returns the sending ID if a signal is found or 0 if no signal is found.

#### 6.3.2.4 `pingNode()`

```
def simlib.anchor.Anchor.pingNode (
    self,
    args )
```



#### 6.3.2.5 prependAction()

```
def simlib.anchor.Anchor.prependAction (
    self,
    function )
```

#### 6.3.2.6 requestPing()

```
def simlib.anchor.Anchor.requestPing (
    self,
    node )
```

#### 6.3.2.7 setTime()

```
def simlib.anchor.Anchor.setTime (
    time )
```

#### 6.3.2.8 waitForReply()

```
def simlib.anchor.Anchor.waitForReply (
    self,
    args )
```

### 6.3.3 Member Data Documentation

#### 6.3.3.1 distance

simlib.anchor.Anchor.distance

#### 6.3.3.2 ID

simlib.anchor.Anchor.ID

#### 6.3.3.3 measurement

`simlib.anchor.Anchor.measurement`

#### 6.3.3.4 pingedNodeID

`simlib.anchor.Anchor.pingedNodeID`

#### 6.3.3.5 signalList

`simlib.anchor.Anchor.signalList`

#### 6.3.3.6 time

`simlib.anchor.Anchor.time`

#### 6.3.3.7 xPos

`simlib.anchor.Anchor.xPos`

#### 6.3.3.8 yPos

`simlib.anchor.Anchor.yPos`

#### 6.3.3.9 zPos

`simlib.anchor.Anchor.zPos`

The documentation for this class was generated from the following file:

- [anchor.py](#)

## 6.4 simlib.archspec.ArchSpec Class Reference

Classes ##.

### Public Member Functions

- def [\\_\\_init\\_\\_](#)
- def [get\\_hubclass](#) (self)
- def [get\\_anchorclass](#) (self)
- def [get\\_nodeclass](#) (self)
- def [set\\_hubclass](#)
- def [set\\_anchorclass](#)
- def [set\\_nodeclass](#)

### 6.4.1 Detailed Description

Classes ##.

Architecture specification  
Defines what classes to use for hub, anchors, and nodes.

The default constructor  
@param hubclass The type of the central hub  
@param anchorclass The type of the anchor  
@param nodeclass The type of the node

### 6.4.2 Constructor & Destructor Documentation

#### 6.4.2.1 [\\_\\_init\\_\\_\(\)](#)

```
def simlib.archspec.ArchSpec.__init__ (
    self,
    hubclass )
```

### 6.4.3 Member Function Documentation

#### 6.4.3.1 [get\\_anchorclass\(\)](#)

```
def simlib.archspec.ArchSpec.get_anchorclass (
    self,
    type )
```

#### 6.4.3.2 `get_hubclass()`

```
def simlib.archspec.ArchSpec.get_hubclass (
    self,
    type )
```

#### 6.4.3.3 `get_nodeclass()`

```
def simlib.archspec.ArchSpec.get_nodeclass (
    self,
    type )
```

#### 6.4.3.4 `set_anchorclass()`

```
def simlib.archspec.ArchSpec.set_anchorclass (
    self,
    anchorclass )
```

#### 6.4.3.5 `set_hubclass()`

```
def simlib.archspec.ArchSpec.set_hubclass (
    self,
    hubclass )
```

#### 6.4.3.6 `set_nodeclass()`

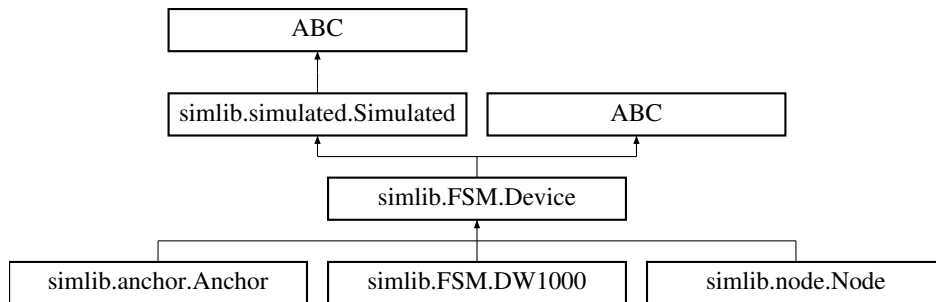
```
def simlib.archspec.ArchSpec.set_nodeclass (
    self,
    nodeclass )
```

The documentation for this class was generated from the following file:

- [archspec.py](#)

## 6.5 simlib.FSM.Device Class Reference

Inheritance diagram for simlib.FSM.Device:



### Public Member Functions

- def [\\_\\_init\\_\\_](#)
- def [getState](#) (self)
- def [setNextState](#)
- def [getParam](#)

### Public Attributes

- [available\\_states](#)
- [initial\\_state](#)
- [dev\\_state](#)
- [physical\\_data](#)
- [next\\_states](#)

### 6.5.1 Detailed Description

Creates an instance of a device dependent on the FSM, physical data, and available states used to accurately model the DW1000 or other like devices.

### 6.5.2 Constructor & Destructor Documentation

#### 6.5.2.1 `__init__()`

```
def simlib.FSM.Device.__init__ (
    self,
    available_states )
```

### 6.5.3 Member Function Documentation

#### 6.5.3.1 `getParam()`

```
def simlib.FSM.Device.getParam (
    self,
    param )
```

#### 6.5.3.2 `getState()`

```
def simlib.FSM.Device.getState (
    self,
    State )
```

Returns the current state of the device.

#### 6.5.3.3 `setNextState()`

```
def simlib.FSM.Device.setNextState (
    self,
    state )
```

### 6.5.4 Member Data Documentation

#### 6.5.4.1 `available_states`

```
simlib.FSM.Device.available_states
```

#### 6.5.4.2 `dev_state`

```
simlib.FSM.Device.dev_state
```

## 6.5.4.3 initial\_state

```
simlib.FSM.Device.initial_state
```

## 6.5.4.4 next\_states

```
simlib.FSM.Device.next_states
```

## 6.5.4.5 physical\_data

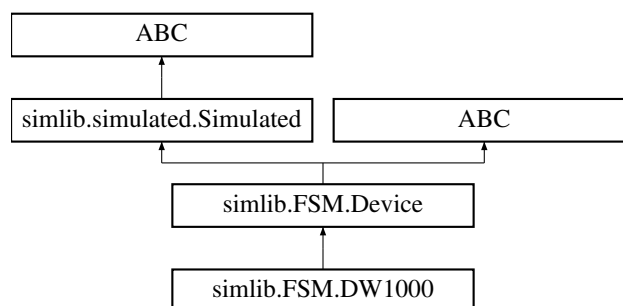
```
simlib.FSM.Device.physical_data
```

The documentation for this class was generated from the following file:

- [FSM.py](#)

## 6.6 simlib.FSM.DW1000 Class Reference

Inheritance diagram for simlib.FSM.DW1000:



## Public Member Functions

- `def \_\_init\_\_`
- `def mainloop`

## Additional Inherited Members

## 6.6.1 Constructor &amp; Destructor Documentation

### 6.6.1.1 `__init__()`

```
def simlib.FSM.DW1000.__init__ (
    self,
    available_states )
```

## 6.6.2 Member Function Documentation

### 6.6.2.1 `mainloop()`

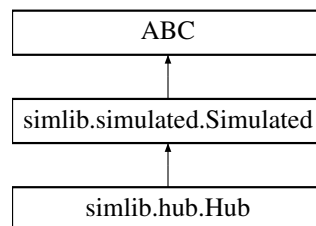
```
def simlib.FSM.DW1000.mainloop (
    self,
    simlist )
```

The documentation for this class was generated from the following file:

- [FSM.py](#)

## 6.7 `simlib.hub.Hub` Class Reference

Inheritance diagram for `simlib.hub.Hub`:



### Public Member Functions

- def `__init__` (self, [algorithm](#))
- def [setTime](#)
- def [containsAnchor](#)
- def [addAnchor](#)
- def [removeAnchor](#)
- def [containsNode](#)
- def [addNode](#)
- def [removeNode](#)
- def [resetMap](#) (self)
- def [mapDistance](#)
- def [getNodePosition](#)
- def [mapAnchorAndNode](#)
- def [triliterateNode](#)
- def [generateCompleteMap](#) (self)
- def [addAction](#)
- def [prependAction](#)
- def [mainloop](#) ()



## Public Attributes

- [time](#)
- [anchors](#)
- [nodes](#)
- [map](#)
- [nodePositions](#)
- [algorithm](#)

## 6.7.1 Constructor & Destructor Documentation

### 6.7.1.1 `__init__()`

```
def simlib.hub.Hub.__init__ (
    self,
    algorithm )
```

## 6.7.2 Member Function Documentation

### 6.7.2.1 `addAction()`

```
def simlib.hub.Hub.addAction (
    self,
    function )
```

### 6.7.2.2 `addAnchor()`

```
def simlib.hub.Hub.addAnchor (
    self,
    anchor )
```

### 6.7.2.3 `addNode()`

```
def simlib.hub.Hub.addNode (
    self,
    node )
```

#### 6.7.2.4 containsAnchor()

```
def simlib.hub.Hub.containsAnchor (
    self,
    anchor )
```

#### 6.7.2.5 containsNode()

```
def simlib.hub.Hub.containsNode (
    self,
    node )
```

#### 6.7.2.6 generateCompleteMap()

```
def simlib.hub.Hub.generateCompleteMap (
    self )
```

Generates a complete map of anchors and nodes

#### 6.7.2.7 getNodePosition()

```
def simlib.hub.Hub.getNodePosition (
    self,
    node )
```

#### 6.7.2.8 mainloop()

```
def simlib.hub.Hub.mainloop ( )
```

#### 6.7.2.9 mapAnchorAndNode()

```
def simlib.hub.Hub.mapAnchorAndNode (
    self,
    args )
```

#### 6.7.2.10 mapDistance()

```
def simlib.hub.Hub.mapDistance (
    self,
    xVal )
```

#### 6.7.2.11 prependAction()

```
def simlib.hub.Hub.prependAction (
    self,
    function )
```

#### 6.7.2.12 removeAnchor()

```
def simlib.hub.Hub.removeAnchor (
    self,
    anchor )
```

#### 6.7.2.13 removeNode()

```
def simlib.hub.Hub.removeNode (
    self,
    node )
```

#### 6.7.2.14 resetMap()

```
def simlib.hub.Hub.resetMap (
    self )
```

Resets the map to [[]]

#### 6.7.2.15 setTime()

```
def simlib.hub.Hub.setTime (
    time )
```

#### 6.7.2.16 trilliterateNode()

```
def simlib.hub.Hub.trilliterateNode (
    node )
```

### 6.7.3 Member Data Documentation

#### 6.7.3.1 algorithm

```
simlib.hub.Hub.algorithm
```

#### 6.7.3.2 anchors

```
simlib.hub.Hub.anchors
```

#### 6.7.3.3 map

```
simlib.hub.Hub.map
```

#### 6.7.3.4 nodePositions

```
simlib.hub.Hub.nodePositions
```

#### 6.7.3.5 nodes

```
simlib.hub.Hub.nodes
```

#### 6.7.3.6 time

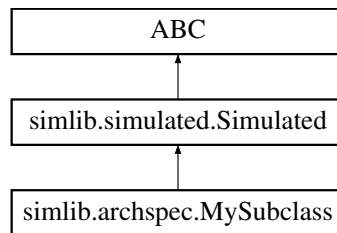
```
simlib.hub.Hub.time
```

The documentation for this class was generated from the following file:

- [hub.py](#)

## 6.8 simlib.archspec.MySubclass Class Reference

Inheritance diagram for simlib.archspec.MySubclass:



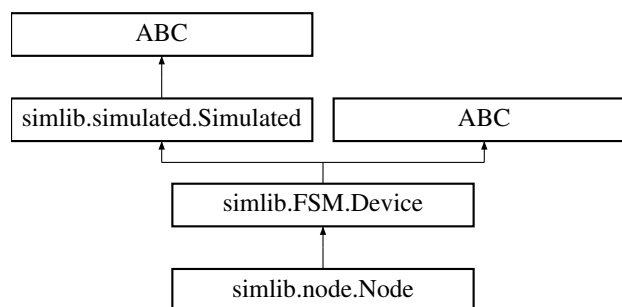
### Additional Inherited Members

The documentation for this class was generated from the following file:

- [archspec.py](#)

## 6.9 simlib.node.Node Class Reference

Inheritance diagram for simlib.node.Node:



### Public Member Functions

- `def \_\_init\_\_`
- `def setTime`
- `def getID ()`
- `def listenForSignal ()`
- `def addAction`
- `def prependAction`

### Public Attributes

- `time`
- `ID`
- `xPos`
- `yPos`
- `zPos`
- `xVel`
- `yVel`
- `zVel`
- `signalList`

## 6.9.1 Constructor & Destructor Documentation

### 6.9.1.1 `__init__()`

```
def simlib.node.Node.__init__ (
    self,
    ID )
```

## 6.9.2 Member Function Documentation

### 6.9.2.1 `addAction()`

```
def simlib.node.Node.addAction (
    self,
    function )
```

### 6.9.2.2 `getID()`

```
def simlib.node.Node.getID (
    int )
```

### 6.9.2.3 `listenForSignal()`

```
def simlib.node.Node.listenForSignal (
    int )
```

Searches to see if it is the target of any signals. Returns the sending ID if a signal is found or 0 if no signal is found.

### 6.9.2.4 `prependAction()`

```
def simlib.node.Node.prependAction (
    self,
    function )
```

### 6.9.2.5 setTime()

```
def simlib.node.Node.setTime (
    time )
```

## 6.9.3 Member Data Documentation

### 6.9.3.1 ID

`simlib.node.Node.ID`

### 6.9.3.2 signalList

`simlib.node.Node.signalList`

### 6.9.3.3 time

`simlib.node.Node.time`

### 6.9.3.4 xPos

`simlib.node.Node.xPos`

### 6.9.3.5 xVel

`simlib.node.Node.xVel`

### 6.9.3.6 yPos

`simlib.node.Node.yPos`

#### 6.9.3.7 yVel

`simlib.node.Node.yVel`

#### 6.9.3.8 zPos

`simlib.node.Node.zPos`

#### 6.9.3.9 zVel

`simlib.node.Node.zVel`

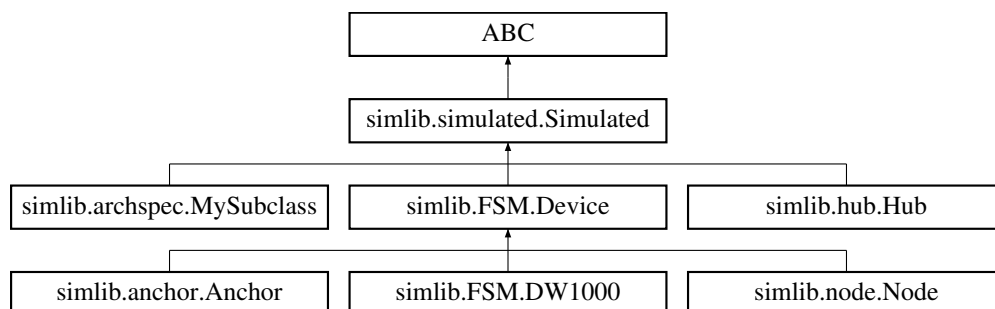
The documentation for this class was generated from the following file:

- [node.py](#)

## 6.10 simlib.simulated.Simulated Class Reference

Classes ##.

Inheritance diagram for `simlib.simulated.Simulated`:



### Public Member Functions

- `def \_\_init\_\_ (self)`
- `def mainloop`
- `def run\_timestep`

### Public Attributes

- `actionQueue`



### 6.10.1 Detailed Description

Classes ##.

### 6.10.2 Constructor & Destructor Documentation

#### 6.10.2.1 `__init__()`

```
def simlib.simulated.Simulated.__init__ (
    self )
```

### 6.10.3 Member Function Documentation

#### 6.10.3.1 `mainloop()`

```
def simlib.simulated.Simulated.mainloop (
    self,
    simlist )
```

#### 6.10.3.2 `run_timestep()`

```
def simlib.simulated.Simulated.run_timestep (
    self,
    simlist )
```

### 6.10.4 Member Data Documentation

#### 6.10.4.1 `actionQueue`

`simlib.simulated.Simulated.actionQueue`

The documentation for this class was generated from the following file:

- [simulated.py](#)

## 6.11 simlib.simulationenvironment.SimulationEnvironment Class Reference

### Public Member Functions

- def `__init__` (self)
- def `createHub`
- def `createNode`
- def `getNodeByID`
- def `associateNode`
- def `dissassociateNode`
- def `deleteNode`
- def `createAnchor`
- def `getAnchorByID`
- def `associateAnchor`
- def `dissassociateAnchor`
- def `deleteAnchor`
- def `mainloop` ()

### Public Attributes

- `time`
- `hubs`
- `anchors`
- `nodes`
- `nextID`
- `signalList`

### 6.11.1 Constructor & Destructor Documentation

#### 6.11.1.1 `__init__()`

```
def simlib.simulationenvironment.SimulationEnvironment.__init__ (  
    self )
```

### 6.11.2 Member Function Documentation

#### 6.11.2.1 `associateAnchor()`

```
def simlib.simulationenvironment.SimulationEnvironment.associateAnchor (  
    self,  
    hub )
```

### 6.11.2.2 associateNode()

```
def simlib.simulationenvironment.SimulationEnvironment.associateNode (
    self,
    hub )
```

### 6.11.2.3 createAnchor()

```
def simlib.simulationenvironment.SimulationEnvironment.createAnchor (
    self,
    xPos )
```

### 6.11.2.4 createHub()

```
def simlib.simulationenvironment.SimulationEnvironment.createHub (
    self,
    algorithm )
```

### 6.11.2.5 createNode()

```
def simlib.simulationenvironment.SimulationEnvironment.createNode (
    self,
    xPos )
```

### 6.11.2.6 deleteAnchor()

```
def simlib.simulationenvironment.SimulationEnvironment.deleteAnchor (
    self,
    anchor )
```

### 6.11.2.7 deleteNode()

```
def simlib.simulationenvironment.SimulationEnvironment.deleteNode (
    self,
    node )
```

#### 6.11.2.8 `dissassociateAnchor()`

```
def simlib.simulationenvironment.SimulationEnvironment.dissassociateAnchor (
    hub )
```

#### 6.11.2.9 `dissassociateNode()`

```
def simlib.simulationenvironment.SimulationEnvironment.dissassociateNode (
    self,
    hub )
```

#### 6.11.2.10 `getAnchorByID()`

```
def simlib.simulationenvironment.SimulationEnvironment.getAnchorByID (
    self,
    ID )
```

#### 6.11.2.11 `getNodeByID()`

```
def simlib.simulationenvironment.SimulationEnvironment.getNodeByID (
    self,
    ID )
```

#### 6.11.2.12 `mainloop()`

```
def simlib.simulationenvironment.SimulationEnvironment.mainloop ( )
```

### 6.11.3 Member Data Documentation

#### 6.11.3.1 `anchors`

```
simlib.simulationenvironment.SimulationEnvironment.anchors
```

### 6.11.3.2 hubs

`simlib.simulationenvironment.SimulationEnvironment.hubs`

### 6.11.3.3 nextID

`simlib.simulationenvironment.SimulationEnvironment.nextID`

### 6.11.3.4 nodes

`simlib.simulationenvironment.SimulationEnvironment.nodes`

### 6.11.3.5 signalList

`simlib.simulationenvironment.SimulationEnvironment.signalList`

### 6.11.3.6 time

`simlib.simulationenvironment.SimulationEnvironment.time`

The documentation for this class was generated from the following file:

- [simulationenvironment.py](#)

## 6.12 simlib.FSM.State Class Reference

### Public Member Functions

- [def `\_\_init\_\_`](#)
- [def `getParam`](#)

### Public Attributes

- [physical\\_data](#)

### 6.12.1 Detailed Description

Creates instance of a State for the device.

## 6.12.2 Constructor & Destructor Documentation

### 6.12.2.1 `__init__()`

```
def simlib.FSM.State.__init__ (
    self,
    physical_data )
```

## 6.12.3 Member Function Documentation

### 6.12.3.1 `getParam()`

```
def simlib.FSM.State.getParam (
    self,
    param )
```

## 6.12.4 Member Data Documentation

### 6.12.4.1 `physical_data`

```
simlib.FSM.State.physical_data
```

The documentation for this class was generated from the following file:

- [FSM.py](#)

## Chapter 7

# File Documentation

### 7.1 `__init__.py` File Reference

#### Namespaces

- [simlib](#)

### 7.2 `action.py` File Reference

#### Classes

- class [simlib.action.Action](#)  
*Classes ##.*

#### Namespaces

- [simlib.action](#)

#### Functions

- def [simlib.action.fn](#) (n)
- def [simlib.action.fn2](#) (x, y)

#### Variables

- int [simlib.action.errors](#) = 0
- [simlib.action.test\\_obj](#) = None

### 7.3 `anchor.py` File Reference

#### Classes

- class [simlib.anchor.Anchor](#)

## Namespaces

- [simlib.anchor](#)

## 7.4 archspec.py File Reference

### Classes

- class [simlib.archspec.ArchSpec](#)  
*Classes ##.*
- class [simlib.archspec.MySubclass](#)

### Namespaces

- [simlib.archspec](#)

### Variables

- int [simlib.archspec.errors](#) = 0
- [simlib.archspec.archspec\\_obj](#) = ArchSpec( int , int , int )

## 7.5 FSM.py File Reference

### Classes

- class [simlib.FSM.State](#)
- class [simlib.FSM.Device](#)
- class [simlib.FSM.DW1000](#)

### Namespaces

- [simlib.FSM](#)

### Functions

- def [simlib.FSM.getOffCurrent](#) (physical\_data)  
*This is dumb but i cant think of a way to do this while making it # general to each device.*
- def [simlib.FSM.getDeepSleepCurrent](#) (physical\_data)
- def [simlib.FSM.getSleepCurrent](#) (physical\_data)
- def [simlib.FSM.getInitCurrent](#) (physical\_data)
- def [simlib.FSM.getIdleCurrent](#) (physical\_data)
- def [simlib.FSM.getRXCurrent](#) (physical\_data)
- def [simlib.FSM.getTXCurrent](#) (physical\_data)
- def [simlib.FSM.getTime](#) (physical\_data)
- def [simlib.FSM.getBattery](#) (dev\_data, state\_data)



## Variables

- `int simlib.FSM.OSC_STARTUP_SEND_CURRENT = 3`
- `int simlib.FSM.OSC_STARTUP_SEND_TIME = 2000`
- `int simlib.FSM.PLL_STARTUP_SEND_CURRENT = 12`
- `int simlib.FSM.PLL_STARTUP_SEND_TIME = 7`
- `int simlib.FSM.WR_TX_DATA_SEND_CURRENT = 15`
- `int simlib.FSM.WR_TX_DATA_SEND_TIME = 10`
- `int simlib.FSM.TX_SHR_SEND_CURRENT = 65`
- `int simlib.FSM.TX_SHR_SEND_TIME = 135`
- `int simlib.FSM.TX_PHR_SEND_CURRENT = 48`
- `float simlib.FSM.TX_PHR_SEND_TIME = 1.33`
- `bool simlib.FSM.DEBUG = True`
- `simlib.FSM.OFF_STATE = State({'current' : getOffCurrent, 'time' : getTime, 'state' : 'off_state'})`  
*FSM for the DW1000, includes states, thier delays, and the current for each state.*
- `simlib.FSM.DEEPSLEEP_STATE = State({'current' : getDeepSleepCurrent, 'time' : getTime, 'state' : 'deepsleep_state'})`
- `simlib.FSM.SLEEP_STATE = State({'current' : getSleepCurrent, 'time' : getTime, 'state' : 'sleep_state'})`
- `simlib.FSM.INIT_STATE = State({'current' : getInitCurrent, 'time' : getTime, 'state' : 'init_state'})`
- `simlib.FSM.IDLE_STATE = State({'current' : getIdleCurrent, 'time' : getTime, 'state' : 'idle_state'})`
- `simlib.FSM.RX_STATE = State({'current' : getRXCurrent, 'time' : getTime, 'state' : 'rx_state'})`
- `simlib.FSM.TX_STATE = State({'current' : getTXCurrent, 'time' : getTime, 'state' : 'tx_state'})`
- `simlib.FSM.INITIAL_STATE = OFF_STATE`
- `dictionary simlib.FSM.DEVICE_DATA = {'batt_life' : getBattery}`
- `dictionary simlib.FSM.AVAILABLE_STATES`
- `simlib.FSM.battlife`
- `simlib.FSM.device = DW1000(AVAILABLE_STATES, INITIAL_STATE, DEVICE_DATA)`
- `int simlib.FSM.errors = 0`
- `list simlib.FSM.simlist = [ ]`
- `simlib.FSM.my_current = device.getState().getParam('current')`

## 7.6 hub.py File Reference

### Classes

- `class simlib.hub.Hub`

### Namespaces

- `simlib.hub`

### Variables

- `simlib.hub.hub = Hub(0)`

## 7.7 node.py File Reference

### Classes

- `class simlib.node.Node`

## Namespaces

- [simlib.node](#)

## 7.8 simulated.py File Reference

### Classes

- class [simlib.simulated.Simulated](#)  
*Classes ##.*
- class [simlib.simulated.ActionQueue](#)

### Namespaces

- [simlib.simulated](#)

### Variables

- bool [simlib.simulated.DEBUG](#) = True
- int [simlib.simulated.errors](#) = 0
- [simlib.simulated.aq](#) = ActionQueue( )
- [simlib.simulated.action1](#) = Action( aq.test, ["I was the first action added"] )
- [simlib.simulated.action2](#) = Action( aq.test, ["then me (:)"] )
- [simlib.simulated.action3](#) = Action( aq.test, ["lastly me (;") ]
- int [simlib.simulated.i](#) = 0

## 7.9 simulationenvironment.py File Reference

### Classes

- class [simlib.simulationenvironment.SimulationEnvironment](#)

### Namespaces

- [simlib.simulationenvironment](#)

### Functions

- def [simlib.simulationenvironment.superSmartTrilaterationAlgorithm](#)

### Variables

- int [simlib.simulationenvironment.errors](#) = 0
- [simlib.simulationenvironment.archspec](#) = ArchSpec(Hub, Anchor, Node)
- [simlib.simulationenvironment.simEnv](#) = SimulationEnvironment()
- [simlib.simulationenvironment.hub](#) = simEnv.createHub(superSmartTrilaterationAlgorithm)
- [simlib.simulationenvironment.anchor](#) = simEnv.createAnchor(0, 0, i)
- [simlib.simulationenvironment.node](#) = simEnv.createNode(3, 3, 3, 0, 0, 0)

# Index

- `__init__`
  - `simlib::FSM::DW1000`, 35
  - `simlib::FSM::Device`, 33
  - `simlib::FSM::State`, 50
  - `simlib::action::Action`, 23
  - `simlib::anchor::Anchor`, 27
  - `simlib::archspec::ArchSpec`, 31
  - `simlib::hub::Hub`, 37
  - `simlib::node::Node`, 42
  - `simlib::simulated::ActionQueue`, 25
  - `simlib::simulated::Simulated`, 45
  - `simlib::simulationenvironment::Simulation↔`
    - Environment, 46
- `__init__.py`, 51
- AVAILABLE\_STATES
  - `simlib::FSM`, 14
- `action.py`, 51
- `action1`
  - `simlib::simulated`, 19
- `action2`
  - `simlib::simulated`, 19
- `action3`
  - `simlib::simulated`, 19
- `actionQueue`
  - `simlib::simulated::Simulated`, 45
- `addAction`
  - `simlib::anchor::Anchor`, 28
  - `simlib::hub::Hub`, 37
  - `simlib::node::Node`, 42
- `addAnchor`
  - `simlib::hub::Hub`, 37
- `addNode`
  - `simlib::hub::Hub`, 37
- `addToQueue`
  - `simlib::simulated::ActionQueue`, 26
- `algorithm`
  - `simlib::hub::Hub`, 40
- `anchor`
  - `simlib::simulationenvironment`, 20
- `anchor.py`, 51
- `anchors`
  - `simlib::hub::Hub`, 40
  - `simlib::simulationenvironment::Simulation↔`
    - Environment, 48
- `aq`
  - `simlib::simulated`, 19
- `archspec`
  - `simlib::simulationenvironment`, 20
- `archspec.py`, 52
- `archspec_obj`
  - `simlib::archspec`, 11
- `associateAnchor`
  - `simlib::simulationenvironment::Simulation↔`
    - Environment, 46
- `associateNode`
  - `simlib::simulationenvironment::Simulation↔`
    - Environment, 46
- `available_states`
  - `simlib::FSM::Device`, 34
- `battlife`
  - `simlib::FSM`, 14
- `containsAnchor`
  - `simlib::hub::Hub`, 37
- `containsNode`
  - `simlib::hub::Hub`, 38
- `createAnchor`
  - `simlib::simulationenvironment::Simulation↔`
    - Environment, 47
- `createHub`
  - `simlib::simulationenvironment::Simulation↔`
    - Environment, 47
- `createNode`
  - `simlib::simulationenvironment::Simulation↔`
    - Environment, 47
- DEBUG
  - `simlib::FSM`, 14
  - `simlib::simulated`, 19
- DEEPSLEEP\_STATE
  - `simlib::FSM`, 14
- DEVICE\_DATA
  - `simlib::FSM`, 15
- `decrement`
  - `simlib::action::Action`, 24
- `deleteAnchor`
  - `simlib::simulationenvironment::Simulation↔`
    - Environment, 47
- `deleteNode`
  - `simlib::simulationenvironment::Simulation↔`
    - Environment, 47
- `dev_state`
  - `simlib::FSM::Device`, 34
- `device`
  - `simlib::FSM`, 14
- `dissassociateAnchor`
  - `simlib::simulationenvironment::Simulation↔`
    - Environment, 47

- dissassociateNode
  - simlib::simulationenvironment::Simulation↔  
Environment, 48
- distance
  - simlib::anchor::Anchor, 29
- errors
  - simlib::FSM, 15
  - simlib::action, 10
  - simlib::archspec, 11
  - simlib::simulated, 19
  - simlib::simulationenvironment, 20
- FSM.py, 52
- fn
  - simlib::action, 10
- fn2
  - simlib::action, 10
- generateCompleteMap
  - simlib::hub::Hub, 38
- get\_anchorclass
  - simlib::archspec::ArchSpec, 31
- get\_args
  - simlib::action::Action, 24
- get\_ctr
  - simlib::action::Action, 24
- get\_fn
  - simlib::action::Action, 24
- get\_hubclass
  - simlib::archspec::ArchSpec, 31
- get\_nodeclass
  - simlib::archspec::ArchSpec, 32
- getAnchorByID
  - simlib::simulationenvironment::Simulation↔  
Environment, 48
- getBattery
  - simlib::FSM, 12
- getDeepSleepCurrent
  - simlib::FSM, 12
- getID
  - simlib::anchor::Anchor, 28
  - simlib::node::Node, 42
- getIdleCurrent
  - simlib::FSM, 13
- getInitCurrent
  - simlib::FSM, 13
- getNodeByID
  - simlib::simulationenvironment::Simulation↔  
Environment, 48
- getNodePosition
  - simlib::hub::Hub, 38
- getOffCurrent
  - simlib::FSM, 13
- getParam
  - simlib::FSM::Device, 34
  - simlib::FSM::State, 50
- getRXCurrent
  - simlib::FSM, 13
- getSleepCurrent
  - simlib::FSM, 13
- getState
  - simlib::FSM::Device, 34
- getTXCurrent
  - simlib::FSM, 14
- getTime
  - simlib::FSM, 13
- hub
  - simlib::hub, 18
  - simlib::simulationenvironment, 21
- hub.py, 53
- hubs
  - simlib::simulationenvironment::Simulation↔  
Environment, 48
- i
  - simlib::simulated, 19
- IDLE\_STATE
  - simlib::FSM, 15
- INIT\_STATE
  - simlib::FSM, 15
- INITIAL\_STATE
  - simlib::FSM, 15
- ID
  - simlib::anchor::Anchor, 29
  - simlib::node::Node, 43
- initial\_state
  - simlib::FSM::Device, 34
- listenForSignal
  - simlib::anchor::Anchor, 28
  - simlib::node::Node, 42
- mainloop
  - simlib::FSM::DW1000, 36
  - simlib::hub::Hub, 38
  - simlib::simulated::Simulated, 45
  - simlib::simulationenvironment::Simulation↔  
Environment, 48
- map
  - simlib::hub::Hub, 40
- mapAnchorAndNode
  - simlib::hub::Hub, 38
- mapDistance
  - simlib::hub::Hub, 38
- measurement
  - simlib::anchor::Anchor, 29
- my\_current
  - simlib::FSM, 15
- next\_states
  - simlib::FSM::Device, 35
- nextID
  - simlib::simulationenvironment::Simulation↔  
Environment, 49
- node
  - simlib::simulationenvironment, 21

- node.py, [53](#)
- nodePositions
  - simlib::hub::Hub, [40](#)
- nodes
  - simlib::hub::Hub, [40](#)
  - simlib::simulationenvironment::Simulation↔Environment, [49](#)
- OFF\_STATE
  - simlib::FSM, [15](#)
- OSC\_STARTUP\_SEND\_CURRENT
  - simlib::FSM, [16](#)
- OSC\_STARTUP\_SEND\_TIME
  - simlib::FSM, [16](#)
- PLL\_STARTUP\_SEND\_CURRENT
  - simlib::FSM, [16](#)
- PLL\_STARTUP\_SEND\_TIME
  - simlib::FSM, [16](#)
- physical\_data
  - simlib::FSM::Device, [35](#)
  - simlib::FSM::State, [50](#)
- pingNode
  - simlib::anchor::Anchor, [28](#)
- pingedNodeID
  - simlib::anchor::Anchor, [30](#)
- popAction
  - simlib::simulated::ActionQueue, [26](#)
- prependAction
  - simlib::anchor::Anchor, [28](#)
  - simlib::hub::Hub, [39](#)
  - simlib::node::Node, [42](#)
- queue
  - simlib::simulated::ActionQueue, [26](#)
- RX\_STATE
  - simlib::FSM, [16](#)
- removeAnchor
  - simlib::hub::Hub, [39](#)
- removeNode
  - simlib::hub::Hub, [39](#)
- requestPing
  - simlib::anchor::Anchor, [29](#)
- resetMap
  - simlib::hub::Hub, [39](#)
- run\_timestep
  - simlib::simulated::Simulated, [45](#)
- SLEEP\_STATE
  - simlib::FSM, [16](#)
- set\_anchorclass
  - simlib::archspec::ArchSpec, [32](#)
- set\_ctr
  - simlib::action::Action, [24](#)
- set\_fn
  - simlib::action::Action, [25](#)
- set\_hubclass
  - simlib::archspec::ArchSpec, [32](#)
- set\_nodeclass
  - simlib::archspec::ArchSpec, [32](#)
- setNextState
  - simlib::FSM::Device, [34](#)
- setTime
  - simlib::anchor::Anchor, [29](#)
  - simlib::hub::Hub, [39](#)
  - simlib::node::Node, [42](#)
- signalList
  - simlib::anchor::Anchor, [30](#)
  - simlib::node::Node, [43](#)
  - simlib::simulationenvironment::Simulation↔Environment, [49](#)
- simEnv
  - simlib::simulationenvironment, [21](#)
- simlib, [9](#)
- simlib.action, [9](#)
- simlib.action.Action, [23](#)
- simlib.anchor, [10](#)
- simlib.anchor.Anchor, [27](#)
- simlib.archspec, [11](#)
- simlib.archspec.ArchSpec, [31](#)
- simlib.archspec.MySubclass, [41](#)
- simlib.FSM.DW1000, [35](#)
- simlib.FSM.Device, [33](#)
- simlib.FSM.State, [49](#)
- simlib.FSM, [11](#)
- simlib.hub, [18](#)
- simlib.hub.Hub, [36](#)
- simlib.node, [18](#)
- simlib.node.Node, [41](#)
- simlib.simulated, [18](#)
- simlib.simulated.ActionQueue, [25](#)
- simlib.simulated.Simulated, [44](#)
- simlib.simulationenvironment, [20](#)
- simlib.simulationenvironment.SimulationEnvironment, [46](#)
- simlib::FSM::DW1000
  - \_\_init\_\_, [35](#)
  - mainloop, [36](#)
- simlib::FSM::Device
  - \_\_init\_\_, [33](#)
  - available\_states, [34](#)
  - dev\_state, [34](#)
  - getParam, [34](#)
  - getState, [34](#)
  - initial\_state, [34](#)
  - next\_states, [35](#)
  - physical\_data, [35](#)
  - setNextState, [34](#)
- simlib::FSM::State
  - \_\_init\_\_, [50](#)
  - getParam, [50](#)
  - physical\_data, [50](#)
- simlib::FSM
  - AVAILABLE\_STATES, [14](#)
  - battlife, [14](#)
  - DEBUG, [14](#)

- DEEPSLEEP\_STATE, 14
- DEVICE\_DATA, 15
- device, 14
- errors, 15
- getBattery, 12
- getDeepSleepCurrent, 12
- getIdleCurrent, 13
- getInitCurrent, 13
- getOffCurrent, 13
- getRXCurrent, 13
- getSleepCurrent, 13
- getTXCurrent, 14
- getTime, 13
- IDLE\_STATE, 15
- INIT\_STATE, 15
- INITIAL\_STATE, 15
- my\_current, 15
- OFF\_STATE, 15
- OSC\_STARTUP\_SEND\_CURRENT, 16
- OSC\_STARTUP\_SEND\_TIME, 16
- PLL\_STARTUP\_SEND\_CURRENT, 16
- PLL\_STARTUP\_SEND\_TIME, 16
- RX\_STATE, 16
- SLEEP\_STATE, 16
- simlist, 16
- TX\_PHR\_SEND\_CURRENT, 17
- TX\_PHR\_SEND\_TIME, 17
- TX\_SHR\_SEND\_CURRENT, 17
- TX\_SHR\_SEND\_TIME, 17
- TX\_STATE, 17
- WR\_TX\_DATA\_SEND\_CURRENT, 17
- WR\_TX\_DATA\_SEND\_TIME, 17
- simlib::action
  - errors, 10
  - fn, 10
  - fn2, 10
  - test\_obj, 10
- simlib::action::Action
  - \_\_init\_\_, 23
  - decrement, 24
  - get\_args, 24
  - get\_ctr, 24
  - get\_fn, 24
  - set\_ctr, 24
  - set\_fn, 25
- simlib::anchor::Anchor
  - \_\_init\_\_, 27
  - addAction, 28
  - distance, 29
  - getID, 28
  - ID, 29
  - listenForSignal, 28
  - measurement, 29
  - pingNode, 28
  - pingedNodeID, 30
  - prependAction, 28
  - requestPing, 29
  - setTime, 29
  - signalList, 30
  - time, 30
  - waitForReply, 29
  - xPos, 30
  - yPos, 30
  - zPos, 30
- simlib::archspec
  - archspec\_obj, 11
  - errors, 11
- simlib::archspec::ArchSpec
  - \_\_init\_\_, 31
  - get\_anchorclass, 31
  - get\_hubclass, 31
  - get\_nodeclass, 32
  - set\_anchorclass, 32
  - set\_hubclass, 32
  - set\_nodeclass, 32
- simlib::hub
  - hub, 18
- simlib::hub::Hub
  - \_\_init\_\_, 37
  - addAction, 37
  - addAnchor, 37
  - addNode, 37
  - algorithm, 40
  - anchors, 40
  - containsAnchor, 37
  - containsNode, 38
  - generateCompleteMap, 38
  - getNodePosition, 38
  - mainloop, 38
  - map, 40
  - mapAnchorAndNode, 38
  - mapDistance, 38
  - nodePositions, 40
  - nodes, 40
  - prependAction, 39
  - removeAnchor, 39
  - removeNode, 39
  - resetMap, 39
  - setTime, 39
  - time, 40
  - triliterateNode, 39
- simlib::node::Node
  - \_\_init\_\_, 42
  - addAction, 42
  - getID, 42
  - ID, 43
  - listenForSignal, 42
  - prependAction, 42
  - setTime, 42
  - signalList, 43
  - time, 43
  - xPos, 43
  - xVel, 43
  - yPos, 43
  - yVel, 43
  - zPos, 44

- zVel, [44](#)
- simlib::simulated
  - action1, [19](#)
  - action2, [19](#)
  - action3, [19](#)
  - aq, [19](#)
  - DEBUG, [19](#)
  - errors, [19](#)
  - i, [19](#)
- simlib::simulated::ActionQueue
  - \_\_init\_\_, [25](#)
  - addToQueue, [26](#)
  - popAction, [26](#)
  - queue, [26](#)
  - test, [26](#)
  - update, [26](#)
- simlib::simulated::Simulated
  - \_\_init\_\_, [45](#)
  - actionQueue, [45](#)
  - mainloop, [45](#)
  - run\_timestep, [45](#)
- simlib::simulationenvironment
  - anchor, [20](#)
  - archspeg, [20](#)
  - errors, [20](#)
  - hub, [21](#)
  - node, [21](#)
  - simEnv, [21](#)
  - superSmartTrilaterationAlgorithm, [20](#)
- simlib::simulationenvironment::SimulationEnvironment
  - \_\_init\_\_, [46](#)
  - anchors, [48](#)
  - associateAnchor, [46](#)
  - associateNode, [46](#)
  - createAnchor, [47](#)
  - createHub, [47](#)
  - createNode, [47](#)
  - deleteAnchor, [47](#)
  - deleteNode, [47](#)
  - dissassociateAnchor, [47](#)
  - dissassociateNode, [48](#)
  - getAnchorByID, [48](#)
  - getNodeByID, [48](#)
  - hubs, [48](#)
  - mainloop, [48](#)
  - nextID, [49](#)
  - nodes, [49](#)
  - signalList, [49](#)
  - time, [49](#)
- simlist
  - simlib::FSM, [16](#)
- simulated.py, [54](#)
- simulationenvironment.py, [54](#)
- superSmartTrilaterationAlgorithm
  - simlib::simulationenvironment, [20](#)
- TX\_PHR\_SEND\_CURRENT
  - simlib::FSM, [17](#)
- TX\_PHR\_SEND\_TIME
  - simlib::FSM, [17](#)
- TX\_SHR\_SEND\_CURRENT
  - simlib::FSM, [17](#)
- TX\_SHR\_SEND\_TIME
  - simlib::FSM, [17](#)
- TX\_STATE
  - simlib::FSM, [17](#)
- test
  - simlib::simulated::ActionQueue, [26](#)
- test\_obj
  - simlib::action, [10](#)
- time
  - simlib::anchor::Anchor, [30](#)
  - simlib::hub::Hub, [40](#)
  - simlib::node::Node, [43](#)
  - simlib::simulationenvironment::SimulationEnvironment, [49](#)
- triliterateNode
  - simlib::hub::Hub, [39](#)
- update
  - simlib::simulated::ActionQueue, [26](#)
- WR\_TX\_DATA\_SEND\_CURRENT
  - simlib::FSM, [17](#)
- WR\_TX\_DATA\_SEND\_TIME
  - simlib::FSM, [17](#)
- waitForReply
  - simlib::anchor::Anchor, [29](#)
- xPos
  - simlib::anchor::Anchor, [30](#)
  - simlib::node::Node, [43](#)
- xVel
  - simlib::node::Node, [43](#)
- yPos
  - simlib::anchor::Anchor, [30](#)
  - simlib::node::Node, [43](#)
- yVel
  - simlib::node::Node, [43](#)
- zPos
  - simlib::anchor::Anchor, [30](#)
  - simlib::node::Node, [44](#)
- zVel
  - simlib::node::Node, [44](#)