

pySM - The python state machine generator

Generated by Doxygen 1.8.13

Contents

1	Data Structure Index	1
1.1	Data Structures	1
2	File Index	3
2.1	File List	3
3	Data Structure Documentation	5
3.1	devCoffee_inputSignalsType Struct Reference	5
3.1.1	Detailed Description	5
3.1.2	Field Documentation	5
3.1.2.1	another_input_ui8	5
3.1.2.2	developer_is_ill_HA_b	5
3.2	devCoffee_outputSignalsType Struct Reference	6
3.2.1	Detailed Description	6
3.2.2	Field Documentation	6
3.2.2.1	and_another_output_ui16	6
3.2.2.2	developer_is_productive_HA_b	6
3.3	pySm_stateMachineType Struct Reference	7
3.3.1	Detailed Description	7
3.3.2	Field Documentation	8
3.3.2.1	actualState	8
3.3.2.2	entryState	8
3.3.2.3	numberOfStates	8
3.3.2.4	numberOfTransitions	8

3.3.2.5	resetVariables	8
3.3.2.6	runEntryOfInitialState_b	9
3.3.2.7	states	9
3.3.2.8	transitions	9
3.4	pySm_stateTransitionType Struct Reference	9
3.4.1	Detailed Description	10
3.4.2	Field Documentation	10
3.4.2.1	destinationState	10
3.4.2.2	sourceState	10
3.4.2.3	transitionAction	10
3.4.2.4	transitionPriority	10
3.4.2.5	transitionTest	11
3.5	pySm_stateType Struct Reference	11
3.5.1	Detailed Description	11
3.5.2	Field Documentation	11
3.5.2.1	onEntryState	11
3.5.2.2	onExitState	11
3.5.2.3	onState	11
4	File Documentation	13
4.1	Debug/LIB/PySm.d File Reference	13
4.2	Debug/LIB/pySm.d File Reference	13
4.3	Debug/main.d File Reference	13
4.4	Debug/SWC/genSM/DevCoffee.d File Reference	13
4.5	Debug/SWC/genSM/Devcoffee.d File Reference	13
4.6	Debug/SWC/genSM/devCoffee.d File Reference	13
4.7	Debug/SWC/sm1.d File Reference	13
4.8	Debug/SWC/Swc.d File Reference	13
4.9	Debug/SWC/swc.d File Reference	13
4.10	LIB/PySm.c File Reference	13
4.10.1	Detailed Description	14

4.10.2	Function Documentation	14
4.10.2.1	PySm_checkState()	15
4.10.2.2	PySm_resetStateMachine()	15
4.10.2.3	PySm_runStateMachine()	16
4.11	LIB/PySm.h File Reference	17
4.11.1	Detailed Description	18
4.11.2	Typedef Documentation	18
4.11.2.1	pySm_stateFunction	18
4.11.2.2	pySm_stateMachineResetFunction	19
4.11.2.3	pySm_transitionActionFunction	19
4.11.2.4	pySm_transitionPriorityType	19
4.11.2.5	pySm_transitionTestFunction	19
4.11.3	Enumeration Type Documentation	19
4.11.3.1	pySm_returnType	19
4.11.4	Function Documentation	20
4.11.4.1	PySm_resetStateMachine()	20
4.11.4.2	PySm_runStateMachine()	20
4.12	LIB/PySm_Cfg.h File Reference	21
4.12.1	Detailed Description	22
4.12.2	Macro Definition Documentation	22
4.12.2.1	PYSM_MAX_NO_OF_TRANSITIONS_PER_STATE	22
4.13	LIB/PySm_types.h File Reference	23
4.13.1	Detailed Description	24
4.13.2	Macro Definition Documentation	24
4.13.2.1	PYSM_FALSE	24
4.13.2.2	PYSM_NULL_PTR	24
4.13.2.3	PYSM_STD_OFF	24
4.13.2.4	PYSM_STD_ON	24
4.13.2.5	PYSM_TRUE	25
4.13.3	Typedef Documentation	25

4.13.3.1	pySm_bool	25
4.13.3.2	pySm_int16	25
4.13.3.3	pySm_int32	25
4.13.3.4	pySm_int8	25
4.13.3.5	pySm_uint16	25
4.13.3.6	pySm_uint32	25
4.13.3.7	pySm_uint8	26
4.13.4	Variable Documentation	26
4.13.4.1	pySm_int64	26
4.13.4.2	pySm_uint64	26
4.14	main.c File Reference	26
4.14.1	Detailed Description	27
4.14.2	Function Documentation	27
4.14.2.1	main()	27
4.15	SWC/genSM/DevCoffee.c File Reference	27
4.15.1	Detailed Description	29
4.15.2	Function Documentation	29
4.15.2.1	DevCoffee_getActiveState()	29
4.15.2.2	DevCoffee_mainFunction()	30
4.15.2.3	devCoffee_SF_BREAKFAST()	30
4.15.2.4	devCoffee_SF_BREAKFAST_entry()	30
4.15.2.5	devCoffee_SF_BREAKFAST_exit()	30
4.15.2.6	devCoffee_SF_DEVELOPER_IS_ILL()	30
4.15.2.7	devCoffee_SF_DEVELOPER_IS_ILL_entry()	31
4.15.2.8	devCoffee_SF_GET_COFFEE_entry()	31
4.15.2.9	devCoffee_SF_IN_OFFICE()	31
4.15.2.10	devCoffee_SF_IN_OFFICE_entry()	31
4.15.2.11	devCoffee_SF_RELAX_AND_SLEEP()	31
4.15.2.12	devCoffee_SF_RELAX_AND_SLEEP_entry()	31
4.15.2.13	devCoffee_TAF_GET_COFFEE_to_IN_OFFICE()	31

4.15.2.14 devCoffee_TAF_IN_OFFICE_to_IN_OFFICE()	32
4.15.2.15 devCoffee_TAF_IN_OFFICE_to_RELAX_AND_SLEEP()	32
4.15.2.16 devCoffee_TAF_RELAX_AND_SLEEP_to_BREAKFAST()	32
4.15.2.17 devCoffee_TTF_BREAKFAST_to_DEVELOPER_IS_ILL()	32
4.15.2.18 devCoffee_TTF_BREAKFAST_to_IN_OFFICE()	32
4.15.2.19 devCoffee_TTF_DEVELOPER_IS_ILL_to_BREAKFAST()	32
4.15.2.20 devCoffee_TTF_IN_OFFICE_to_GET_COFFEE()	32
4.15.2.21 devCoffee_TTF_IN_OFFICE_to_IN_OFFICE()	33
4.15.2.22 devCoffee_TTF_IN_OFFICE_to_RELAX_AND_SLEEP()	33
4.15.2.23 devCoffee_TTF_RELAX_AND_SLEEP_to_BREAKFAST_1()	33
4.15.2.24 devCoffee_TTF_RELAX_AND_SLEEP_to_BREAKFAST_2()	33
4.15.2.25 devCoffee_variableResetFunction()	33
4.15.3 Variable Documentation	33
4.15.3.1 coffeein_level_ui8	33
4.15.3.2 current_hour_ui8	34
4.15.3.3 devCoffee_activeState	34
4.15.3.4 devCoffee_inputSignals	34
4.15.3.5 devCoffee_outputSignals	34
4.15.3.6 devCoffee_state_BREAKFAST	34
4.15.3.7 devCoffee_state_DEVELOPER_IS_ILL	34
4.15.3.8 devCoffee_state_GET_COFFEE	35
4.15.3.9 devCoffee_state_IN_OFFICE	35
4.15.3.10 devCoffee_state_RELAX_AND_SLEEP	35
4.15.3.11 devCoffee_stateMachine_s	35
4.15.3.12 devCoffee_states_pa	36
4.15.3.13 devCoffee_transitions_sa	36
4.15.3.14 productivity_ui8	36
4.16 SWC/genSM/DevCoffee.h File Reference	36
4.16.1 Detailed Description	38
4.16.2 Macro Definition Documentation	38

4.16.2.1	GO_HOME_TIME	38
4.16.2.2	WAKE_UP_TIME_H	38
4.16.2.3	WORKTIME_PER_DAY_H	38
4.16.3	Enumeration Type Documentation	38
4.16.3.1	devCoffee_activeStateType	38
4.16.4	Function Documentation	39
4.16.4.1	DevCoffee_getActiveState()	39
4.16.4.2	DevCoffee_mainFunction()	39
4.17	SWC/Swc.c File Reference	40
4.17.1	Detailed Description	40
4.17.2	Function Documentation	41
4.17.2.1	Swc_main()	41
4.18	SWC/Swc.h File Reference	41
4.18.1	Detailed Description	41
4.18.2	Function Documentation	42
4.18.2.1	Swc_main()	42
Index		43

Chapter 1

Data Structure Index

1.1 Data Structures

Here are the data structures with brief descriptions:

devCoffee_inputSignalsType	
Structure defining input signals for state machine devCoffee	5
devCoffee_outputSignalsType	
Structure defining output signals for state machine devCoffee	6
pySm_stateMachineType	
Structure defining a state machine	7
pySm_stateTransitionType	
Structure defining a state transition	9
pySm_stateType	
Structure defining a state	11

Chapter 2

File Index

2.1 File List

Here is a list of all files with brief descriptions:

main.c	Test main function, calling the test-SWC	26
Debug/ main.d		13
Debug/LIB/ PySm.d		13
Debug/LIB/ pySm.d		13
Debug/SWC/ sm1.d		13
Debug/SWC/ swc.d		13
Debug/SWC/ Swc.d		13
Debug/SWC/genSM/ DevCoffee.d		13
Debug/SWC/genSM/ devCoffee.d		13
Debug/SWC/genSM/ Devcoffee.d		13
LIB/ PySm.c	File containing the main implementation of the pySM API and internal functions	13
LIB/ PySm.h	File containing typedefs and extern callable function declarations	17
LIB/ PySm_Cfg.h	File containing configuration of the pySM library	21
LIB/ PySm_types.h	File containing basic data types	23
SWC/ Swc.c	Test-SWC to demonstrate the use of the generated state machine	40
SWC/ Swc.h	Header file of Test-SWC to demonstrate the use of the generated state machine	41
SWC/genSM/ DevCoffee.c	Header for generated state machine devCoffee Generated 2017-09-27 19:41:10 by PySM - The python state machine generator	27
SWC/genSM/ DevCoffee.h	Header for generated state machine devCoffee Generated 2017-09-27 19:41:10 by PySM - The python state machine generator	36

Chapter 3

Data Structure Documentation

3.1 devCoffee_inputSignalsType Struct Reference

Structure defining input signals for state machine devCoffee.

```
#include <DevCoffee.h>
```

Data Fields

- [pySm_bool developer_is_ill_HA_b](#)
- [pySm_uint8 another_input_ui8](#)

3.1.1 Detailed Description

Structure defining input signals for state machine devCoffee.

3.1.2 Field Documentation

3.1.2.1 another_input_ui8

```
pySm_uint8 devCoffee_inputSignalsType::another_input_ui8
```

3.1.2.2 developer_is_ill_HA_b

```
pySm_bool devCoffee_inputSignalsType::developer_is_ill_HA_b
```

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

- SWC/genSM/[DevCoffee.h](#)

3.2 devCoffee_outputSignalsType Struct Reference

Structure defining output signals for state machine devCoffee.

```
#include <DevCoffee.h>
```

Data Fields

- [pySm_bool developer_is_productive_HA_b](#)
- [pySm_uint16 and_another_output_ui16](#)

3.2.1 Detailed Description

Structure defining output signals for state machine devCoffee.

3.2.2 Field Documentation

3.2.2.1 and_another_output_ui16

```
pySm_uint16 devCoffee_outputSignalsType::and_another_output_ui16
```

3.2.2.2 developer_is_productive_HA_b

```
pySm_bool devCoffee_outputSignalsType::developer_is_productive_HA_b
```

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

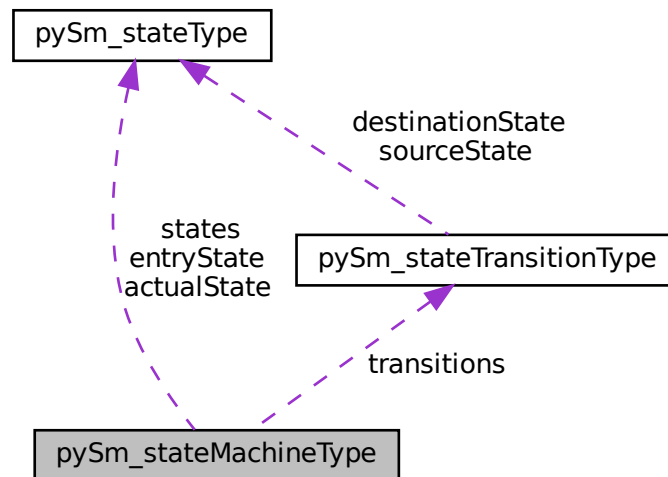
- [SWC/genSM/DevCoffee.h](#)

3.3 pySm_stateMachineType Struct Reference

Structure defining a state machine.

```
#include <PySm.h>
```

Collaboration diagram for pySm_stateMachineType:



Data Fields

- const `pySm_stateType` * `entryState`
- `pySm_stateType` * `actualState`
- const `pySm_stateType` ** `states`
- const `pySm_uint8` `numberOfStates`
- const `pySm_stateTransitionType` * `transitions`
- const `pySm_uint8` `numberOfTransitions`
- `pySm_bool` `runEntryOfInitialState_b`
- const `pySm_stateMachineResetFunction` `resetVariables`

3.3.1 Detailed Description

Structure defining a state machine.

A state machine is defined by it's

- entry state `pySm_stateMachineType::entryState`
- current active state `pySm_stateMachineType::actualState`

- a list of all states, `pySm_stateMachineType::states` given by an pointer array, pointing to all states of the generated state machine
- the overall number of all states `pySm_stateMachineType::numberOfTransitions`
- all existing transitions of the state machine `pySm_stateMachineType::transitions`, given by an pointer to an array, containing the transitions
- the overall number of transitions `pySm_stateMachineType::numberOfTransitions`
- a flag `pySm_stateMachineType::runEntryOfInitialState_b`, enabling the execution of the onEntry-function of the first, initial state (entryState). This flag get's generated as TRUE when the entryState has an onEntry statement
- a function `pySm_stateMachineResetFunction::resetVariables` for resetting the state machine's local variables

3.3.2 Field Documentation

3.3.2.1 actualState

```
pySm_stateType* pySm_stateMachineType::actualState
```

3.3.2.2 entryState

```
const pySm_stateType* pySm_stateMachineType::entryState
```

3.3.2.3 numberOfStates

```
const pySm_uint8 pySm_stateMachineType::numberOfStates
```

3.3.2.4 numberOfTransitions

```
const pySm_uint8 pySm_stateMachineType::numberOfTransitions
```

3.3.2.5 resetVariables

```
const pySm_stateMachineResetFunction pySm_stateMachineType::resetVariables
```


3.3.2.6 runEntryOfInitialState_b

```
pySm_bool pySm_stateMachineType::runEntryOfInitialState_b
```

3.3.2.7 states

```
const pySm_stateType** pySm_stateMachineType::states
```

3.3.2.8 transitions

```
const pySm_stateTransitionType* pySm_stateMachineType::transitions
```

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

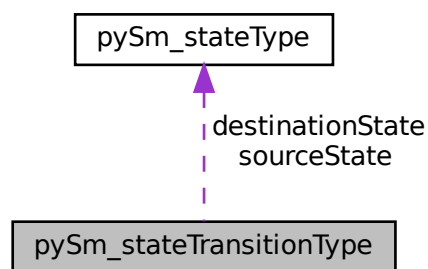
- [LIB/PySm.h](#)

3.4 pySm_stateTransitionType Struct Reference

Structure defining a state transition.

```
#include <PySm.h>
```

Collaboration diagram for pySm_stateTransitionType:



Data Fields

- const [pySm_stateType](#) * [sourceState](#)
- const [pySm_stateType](#) * [destinationState](#)
- const [pySm_transitionTestFunction](#) [transitionTest](#)
- const [pySm_transitionPriorityType](#) [transitionPriority](#)
- const [pySm_transitionActionFunction](#) [transitionAction](#)

3.4.1 Detailed Description

Structure defining a state transition.

A state transition is defined by it's

- source `pySm_stateTransitionType::sourceState` and
- destination state `pySm_stateTransitionType::destinationState` ,
- it's transition condition `pySm_stateTransitionType::transitionTest` (given by the transition test function) ,
- the transitions priority `pySm_stateTransitionType::transitionPriority`
- and actions to be performed, if an transition has been triggered, given by the transition action function `py↔
Sm_stateTransitionType::transitionAction`

3.4.2 Field Documentation

3.4.2.1 destinationState

```
const pySm_stateType* pySm_stateTransitionType::destinationState
```

3.4.2.2 sourceState

```
const pySm_stateType* pySm_stateTransitionType::sourceState
```

3.4.2.3 transitionAction

```
const pySm_transitionActionFunction pySm_stateTransitionType::transitionAction
```

3.4.2.4 transitionPriority

```
const pySm_transitionPriorityType pySm_stateTransitionType::transitionPriority
```

3.4.2.5 transitionTest

```
const pySm\_transitionTestFunction pySm_stateTransitionType::transitionTest
```

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

- [LIB/PySm.h](#)

3.5 pySm_stateType Struct Reference

Structure defining a state.

```
#include <PySm.h>
```

Data Fields

- const [pySm_stateFunction](#) onEntryState
- const [pySm_stateFunction](#) onState
- const [pySm_stateFunction](#) onExitState

3.5.1 Detailed Description

Structure defining a state.

A state is defined by it's

- entry function `PySm_StateType::onEntryState`
- during (state main function) `PySm_StateType::onState`
- exit function `PySm_StateType::onExitState`. Each of these functions can be generated as `NULL_PTR`, if not needed

3.5.2 Field Documentation

3.5.2.1 onEntryState

```
const pySm\_stateFunction pySm_stateType::onEntryState
```

OnEntryState-function, if needed.

3.5.2.2 onExitState

```
const pySm\_stateFunction pySm_stateType::onExitState
```

OnExitState-function, if needed.

3.5.2.3 onState

```
const pySm\_stateFunction pySm_stateType::onState
```

OnDuringState-function, if needed.

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

- [LIB/PySm.h](#)

Chapter 4

File Documentation

4.1 Debug/LIB/PySm.d File Reference

4.2 Debug/LIB/pySm.d File Reference

4.3 Debug/main.d File Reference

4.4 Debug/SWC/genSM/DevCoffee.d File Reference

4.5 Debug/SWC/genSM/Devcoffee.d File Reference

4.6 Debug/SWC/genSM/devCoffee.d File Reference

4.7 Debug/SWC/sm1.d File Reference

4.8 Debug/SWC/Swc.d File Reference

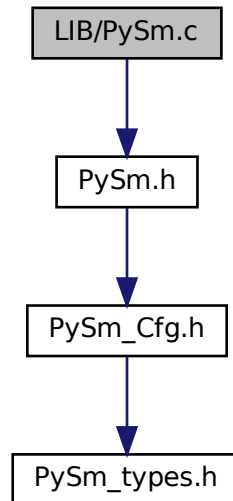
4.9 Debug/SWC/swc.d File Reference

4.10 LIB/PySm.c File Reference

File containing the main implementation of the pySM API and internal functions.

```
#include "PySm.h"
```

Include dependency graph for PySm.c:



Functions

- static `pySm_returnType PySm_checkState` (const `pySm_stateMachineType` *stateMachine, const `pySm_stateType` *stateToCheck)
Checks a state of a given state machine.
- `pySm_returnType PySm_runStateMachine` (`pySm_stateMachineType` *stateMachine)
Runs the given state machine.
- `pySm_returnType PySm_resetStateMachine` (`pySm_stateMachineType` *stateMachine)
Resets the given state machine.

4.10.1 Detailed Description

File containing the main implementation of the pySM API and internal functions.

Author

Markus Burger

Date

2017-09-11

4.10.2 Function Documentation

4.10.2.1 PySm_checkState()

```
static pySm_returnType PySm_checkState (
    const pySm_stateMachineType * stateMachine,
    const pySm_stateType * stateToCheck ) [static]
```

Checks a state of a given state machine.

Checks, if a given state exists in a given state machine

```
pySm_returnType out = pySm_checkState(&stateMachine, &stateToCheck);
```

Parameters

<i>stateMachine</i>	State machine object
<i>stateToCheck</i>	State to check if existing in <i>stateMachine</i>

Returns

Returns either PYSM_E_UNKNOWN_STATE when the given state doesn't exist in *stateMachine* or PYSM_E_OK

4.10.2.2 PySm_resetStateMachine()

```
pySm_returnType PySm_resetStateMachine (
    pySm_stateMachineType * stateMachine )
```

Resets the given state machine.

Resets the given state machine back to initial state and resets the state machine's local variables back to their initial values

```
pySm_returnType out = pySm_resetStateMachine(&stateMachine);
```

Parameters

<i>stateMachine</i>	State machine object to reset
---------------------	-------------------------------

Returns

Returns either PYSM_INVALID_MACHINE when an invalid stateMachine has been given or PYSM_E_OK

Here is the call graph for this function:

**4.10.2.3 PySm_runStateMachine()**

```
pySm_returnType PySm_runStateMachine (
    pySm_stateMachineType * stateMachine )
```

Runs the given state machine.

Executes the given state machine

```
pySm_returnType out = pySm_runStateMachine(&stateMachine);
```

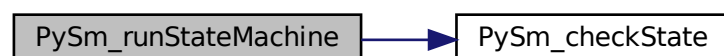
Parameters

<i>stateMachine</i>	State machine object to run
---------------------	-----------------------------

Returns

Returns either PYSM_INVALID_MACHINE when an invalid statemachine has been given or PYSM_E_OK

Here is the call graph for this function:

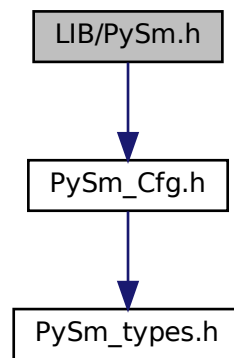


4.11 LIB/PySm.h File Reference

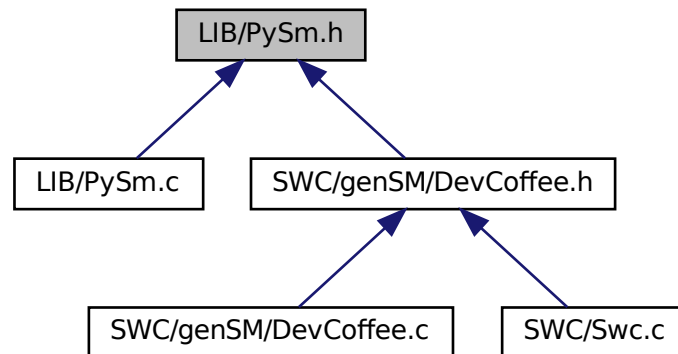
File containing typedefs and extern callable function declarations.

```
#include "PySm_Cfg.h"
```

Include dependency graph for PySm.h:



This graph shows which files directly or indirectly include this file:



Data Structures

- struct [pySm_stateType](#)
Structure defining a state.
- struct [pySm_stateTransitionType](#)
Structure defining a state transition.
- struct [pySm_stateMachineType](#)
Structure defining a state machine.

Typedefs

- typedef void(* [pySm_stateFunction](#)) (void)
Function pointer definition for state functions.
- typedef void(* [pySm_stateMachineResetFunction](#)) (void)
Function pointer definition for state machine reset functions.
- typedef void(* [pySm_transitionActionFunction](#)) (void)
Function pointer definition for state transition functions.
- typedef [pySm_bool](#)(* [pySm_transitionTestFunction](#)) (void)
Function pointer definition for state transition test functions.
- typedef [pySm_uint8](#) [pySm_transitionPriorityType](#)
Type for transition priorities.

Enumerations

- enum [pySm_returnType](#) { [PYSM_E_OK](#) = 0u, [PYSM_E_UNKNOWN_STATE](#), [PYSM_E_UNKNOWN_TRANSITION](#), [PYSM_INVALID_MACHINE](#) }
Return type for the pySM librarie's API functions.

Functions

- [pySm_returnType](#) [PySm_runStateMachine](#) ([pySm_stateMachineType](#) *stateMachine)
Runs the given state machine.
- [pySm_returnType](#) [PySm_resetStateMachine](#) ([pySm_stateMachineType](#) *stateMachine)
Resets the given state machine.

4.11.1 Detailed Description

File containing typedefs and extern callable function declarations.

Author

Markus Burger

Date

2017-09-11

4.11.2 Typedef Documentation

4.11.2.1 [pySm_stateFunction](#)

```
typedef void(* pySm_stateFunction) (void)
```

Function pointer definition for state functions.

4.11.2.2 pySm_stateMachineResetFunction

```
typedef void(* pySm_stateMachineResetFunction) (void)
```

Function pointer definition for state machine reset functions.

This function get's generated and is used to reset state machine local internal static variables. This function normally get's called only when resetting a whole state machine by the according API function.

4.11.2.3 pySm_transitionActionFunction

```
typedef void(* pySm_transitionActionFunction) (void)
```

Function pointer definition for state transition functions.

These functions get called (if generated/needed) to perform given actions when a transition gets triggered and executed.

4.11.2.4 pySm_transitionPriorityType

```
typedef pySm_uint8 pySm_transitionPriorityType
```

Type for transition priorities.

4.11.2.5 pySm_transitionTestFunction

```
typedef pySm_bool(* pySm_transitionTestFunction) (void)
```

Function pointer definition for state transition test functions.

These functions get called to perform the evaluation of the transition condition.

4.11.3 Enumeration Type Documentation

4.11.3.1 pySm_returnType

```
enum pySm_returnType
```

Return type for the pySM librarie's API functions.

Enumerator

PYSM_E_OK	
PYSM_E_UNKNOWN_STATE	
PYSM_E_UNKNOWN_TRANSITION	
PYSM_INVALID_MACHINE	

4.11.4 Function Documentation

4.11.4.1 PySm_resetStateMachine()

```
pySm_returnType PySm_resetStateMachine (
    pySm_stateMachineType * stateMachine )
```

Resets the given state machine.

Resets the given state machine back to initial state and resets the state machine's local variables back to their initial values

```
pySm_returnType out = pySm_resetStateMachine(&stateMachine);
```

Parameters

<i>stateMachine</i>	State machine object to reset
---------------------	-------------------------------

Returns

Returns either PYSM_INVALID_MACHINE when an invalid stateMachine has been given or PYSM_E_OK

Here is the call graph for this function:



4.11.4.2 PySm_runStateMachine()

```
pySm_returnType PySm_runStateMachine (
    pySm_stateMachineType * stateMachine )
```

Runs the given state machine.

Executes the given state machine

```
pySm_returnType out = pySm_runStateMachine(&stateMachine);
```

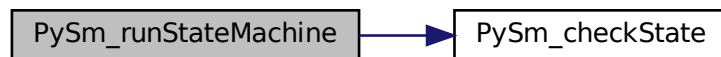
Parameters

<i>stateMachine</i>	State machine object to run
---------------------	-----------------------------

Returns

Returns either PYSM_INVALID_MACHINE when an invalid statemachine has been given or PYSM_E_OK

Here is the call graph for this function:

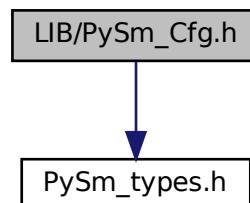


4.12 LIB/PySm_Cfg.h File Reference

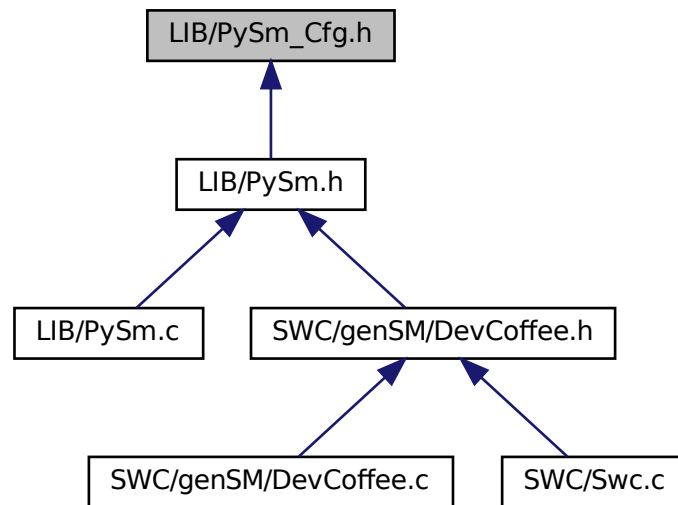
File containing configuration of the pySM library.

```
#include "PySm_types.h"
```

Include dependency graph for PySm_Cfg.h:



This graph shows which files directly or indirectly include this file:



Macros

- `#define PYSM_MAX_NO_OF_TRANSITIONS_PER_STATE 10u`
Number of maximum allowed in-/outgoing transitions per state.

4.12.1 Detailed Description

File containing configuration of the pySM library.

Author

Markus Burger

Date

2017-09-11

4.12.2 Macro Definition Documentation

4.12.2.1 PYSM_MAX_NO_OF_TRANSITIONS_PER_STATE

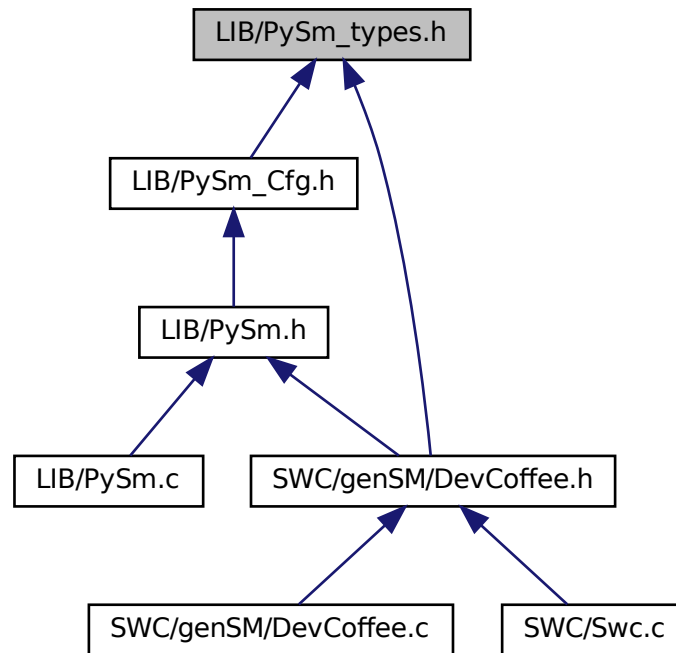
```
#define PYSM_MAX_NO_OF_TRANSITIONS_PER_STATE 10u
```

Number of maximum allowed in-/outgoing transitions per state.

4.13 LIB/PySm_types.h File Reference

File containing basic data types.

This graph shows which files directly or indirectly include this file:



Macros

- `#define PYSM_TRUE 1u`
- `#define PYSM_FALSE 0u`
- `#define PYSM_STD_ON 1u`
- `#define PYSM_STD_OFF 0u`
- `#define PYSM_NULL_PTR (void *)0`

Typedefs

- `typedef signed char pySm_int8`
- `typedef short int pySm_int16`
- `typedef int pySm_int32`
- `typedef unsigned char pySm_uint8`
- `typedef unsigned short int pySm_uint16`
- `typedef unsigned int pySm_uint32`
- `typedef unsigned char pySm_bool`

Variables

- `__extension__` typedef long long int [pySm_int64](#)
- `__extension__` typedef unsigned long long int [pySm_uint64](#)

4.13.1 Detailed Description

File containing basic data types.

Author

Markus Burger

Date

2017-09-11

4.13.2 Macro Definition Documentation

4.13.2.1 PYSM_FALSE

```
#define PYSM_FALSE 0u
```

4.13.2.2 PYSM_NULL_PTR

```
#define PYSM_NULL_PTR (void *)0
```

4.13.2.3 PYSM_STD_OFF

```
#define PYSM_STD_OFF 0u
```

4.13.2.4 PYSM_STD_ON

```
#define PYSM_STD_ON 1u
```


4.13.2.5 PYSM_TRUE

```
#define PYSM_TRUE 1u
```

4.13.3 Typedef Documentation

4.13.3.1 pySm_bool

```
typedef unsigned char pySm\_bool
```

4.13.3.2 pySm_int16

```
typedef short int pySm\_int16
```

4.13.3.3 pySm_int32

```
typedef int pySm\_int32
```

4.13.3.4 pySm_int8

```
typedef signed char pySm\_int8
```

4.13.3.5 pySm_uint16

```
typedef unsigned short int pySm\_uint16
```

4.13.3.6 pySm_uint32

```
typedef unsigned int pySm\_uint32
```

4.13.3.7 pySm_uint8

```
typedef unsigned char pySm_uint8
```

4.13.4 Variable Documentation

4.13.4.1 pySm_int64

```
__extension__ typedef long long int pySm_int64
```

4.13.4.2 pySm_uint64

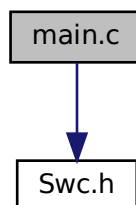
```
__extension__ typedef unsigned long long int pySm_uint64
```

4.14 main.c File Reference

Test main function, calling the test-SWC.

```
#include "Swc.h"
```

Include dependency graph for main.c:



Functions

- int [main](#) (void)

4.14.1 Detailed Description

Test main function, calling the test-SWC.

Author

Markus Burger

Date

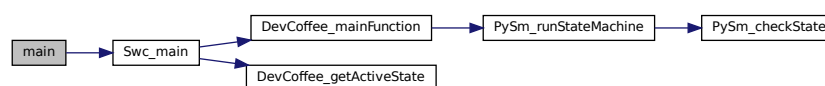
2017-09-11

4.14.2 Function Documentation

4.14.2.1 main()

```
int main (  
    void )
```

Here is the call graph for this function:

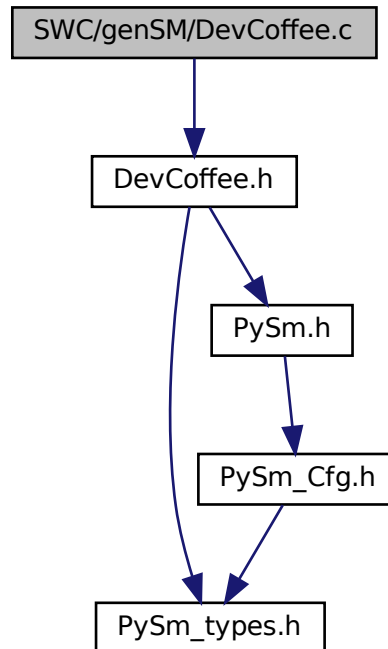


4.15 SWC/genSM/DevCoffee.c File Reference

Header for generated state machine devCoffee Generated 2017-09-27 19:41:10 by PySM - The python state machine generator.

```
#include "DevCoffee.h"
```

Include dependency graph for DevCoffee.c:



Functions

- static void [devCoffee_SF_BREAKFAST_entry](#) (void)
- static void [devCoffee_SF_BREAKFAST](#) (void)
- static void [devCoffee_SF_BREAKFAST_exit](#) (void)
- static void [devCoffee_SF_IN_OFFICE_entry](#) (void)
- static void [devCoffee_SF_IN_OFFICE](#) (void)
- static void [devCoffee_SF_GET_COFFEE_entry](#) (void)
- static void [devCoffee_SF_RELAX_AND_SLEEP_entry](#) (void)
- static void [devCoffee_SF_RELAX_AND_SLEEP](#) (void)
- static void [devCoffee_SF_DEVELOPER_IS_ILL_entry](#) (void)
- static void [devCoffee_SF_DEVELOPER_IS_ILL](#) (void)
- static void [devCoffee_variableResetFunction](#) (void)
- static [pySm_bool](#) [devCoffee_TTF_IN_OFFICE_to_IN_OFFICE](#) (void)
- static [pySm_bool](#) [devCoffee_TTF_BREAKFAST_to_IN_OFFICE](#) (void)
- static [pySm_bool](#) [devCoffee_TTF_IN_OFFICE_to_GET_COFFEE](#) (void)
- static [pySm_bool](#) [devCoffee_TTF_IN_OFFICE_to_RELAX_AND_SLEEP](#) (void)
- static [pySm_bool](#) [devCoffee_TTF_RELAX_AND_SLEEP_to_BREAKFAST_1](#) (void)
- static [pySm_bool](#) [devCoffee_TTF_BREAKFAST_to_DEVELOPER_IS_ILL](#) (void)
- static [pySm_bool](#) [devCoffee_TTF_DEVELOPER_IS_ILL_to_BREAKFAST](#) (void)
- static [pySm_bool](#) [devCoffee_TTF_RELAX_AND_SLEEP_to_BREAKFAST_2](#) (void)
- static void [devCoffee_TAF_IN_OFFICE_to_IN_OFFICE](#) (void)
- static void [devCoffee_TAF_GET_COFFEE_to_IN_OFFICE](#) (void)

- static void `devCoffee_TAF_IN_OFFICE_to_RELAX_AND_SLEEP` (void)
- static void `devCoffee_TAF_RELAX_AND_SLEEP_to_BREAKFAST` (void)
- `pySm_returnType DevCoffee_mainFunction` (`devCoffee_inputSignalsType *swc_inputSignals`, `devCoffee_outputSignalsType *swc_outputSignals`)
Main function of the state machine devCoffee.
- void `DevCoffee_getActiveState` (`devCoffee_activeStateType *swc_activeState`)
Main function of the state machine devCoffee.

Variables

- static const `pySm_stateType devCoffee_state_BREAKFAST`
- static const `pySm_stateType devCoffee_state_IN_OFFICE`
- static const `pySm_stateType devCoffee_state_GET_COFFEE`
- static const `pySm_stateType devCoffee_state_RELAX_AND_SLEEP`
- static const `pySm_stateType devCoffee_state_DEVELOPER_IS_ILL`
- static const `pySm_stateType * devCoffee_states_pa` [5]
- static `pySm_stateTransitionType devCoffee_transitions_sa` [9]
- `pySm_stateMachineType devCoffee_stateMachine_s`
- static `devCoffee_inputSignalsType * devCoffee_inputSignals`
- static `devCoffee_outputSignalsType * devCoffee_outputSignals`
- static `devCoffee_activeStateType devCoffee_activeState = devCoffee_UNINITIALIZED_STATE_MACHINE`
- `pySm_uint8 coffeein_level_ui8 = 0u`
- `pySm_uint8 productivity_ui8 = 0u`
- `pySm_uint8 current_hour_ui8 = 0u`

4.15.1 Detailed Description

Header for generated state machine devCoffee Generated 2017-09-27 19:41:10 by PySM - The python state machine generator.

Author

Markus

Date

2017-09-27

4.15.2 Function Documentation

4.15.2.1 DevCoffee_getActiveState()

```
void DevCoffee_getActiveState (
    devCoffee_activeStateType * swc_activeState )
```

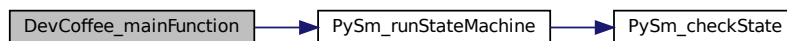
Main function of the state machine devCoffee.

4.15.2.2 DevCoffee_mainFunction()

```
pySm_returnType DevCoffee_mainFunction (
    devCoffee_inputSignalsType * swc_inputSignals,
    devCoffee_outputSignalsType * swc_outputSignals )
```

Main function of the state machine devCoffee.

Here is the call graph for this function:



4.15.2.3 devCoffee_SF_BREAKFAST()

```
static void devCoffee_SF_BREAKFAST (
    void ) [static]
```

4.15.2.4 devCoffee_SF_BREAKFAST_entry()

```
static void devCoffee_SF_BREAKFAST_entry (
    void ) [static]
```

4.15.2.5 devCoffee_SF_BREAKFAST_exit()

```
static void devCoffee_SF_BREAKFAST_exit (
    void ) [static]
```

4.15.2.6 devCoffee_SF_DEVELOPER_IS_ILL()

```
static void devCoffee_SF_DEVELOPER_IS_ILL (
    void ) [static]
```

4.15.2.7 devCoffee_SF_DEVELOPER_IS_ILL_entry()

```
static void devCoffee_SF_DEVELOPER_IS_ILL_entry (  
    void ) [static]
```

4.15.2.8 devCoffee_SF_GET_COFFEE_entry()

```
static void devCoffee_SF_GET_COFFEE_entry (  
    void ) [static]
```

4.15.2.9 devCoffee_SF_IN_OFFICE()

```
static void devCoffee_SF_IN_OFFICE (  
    void ) [static]
```

4.15.2.10 devCoffee_SF_IN_OFFICE_entry()

```
static void devCoffee_SF_IN_OFFICE_entry (  
    void ) [static]
```

4.15.2.11 devCoffee_SF_RELAX_AND_SLEEP()

```
static void devCoffee_SF_RELAX_AND_SLEEP (  
    void ) [static]
```

4.15.2.12 devCoffee_SF_RELAX_AND_SLEEP_entry()

```
static void devCoffee_SF_RELAX_AND_SLEEP_entry (  
    void ) [static]
```

4.15.2.13 devCoffee_TAF_GET_COFFEE_to_IN_OFFICE()

```
static void devCoffee_TAF_GET_COFFEE_to_IN_OFFICE (  
    void ) [static]
```

4.15.2.14 devCoffee_TAF_IN_OFFICE_to_IN_OFFICE()

```
static void devCoffee_TAF_IN_OFFICE_to_IN_OFFICE (  
    void ) [static]
```

4.15.2.15 devCoffee_TAF_IN_OFFICE_to_RELAX_AND_SLEEP()

```
static void devCoffee_TAF_IN_OFFICE_to_RELAX_AND_SLEEP (  
    void ) [static]
```

4.15.2.16 devCoffee_TAF_RELAX_AND_SLEEP_to_BREAKFAST()

```
static void devCoffee_TAF_RELAX_AND_SLEEP_to_BREAKFAST (  
    void ) [static]
```

4.15.2.17 devCoffee_TTF_BREAKFAST_to_DEVELOPER_IS_ILL()

```
static pySm_bool devCoffee_TTF_BREAKFAST_to_DEVELOPER_IS_ILL (  
    void ) [static]
```

4.15.2.18 devCoffee_TTF_BREAKFAST_to_IN_OFFICE()

```
static pySm_bool devCoffee_TTF_BREAKFAST_to_IN_OFFICE (  
    void ) [static]
```

4.15.2.19 devCoffee_TTF_DEVELOPER_IS_ILL_to_BREAKFAST()

```
static pySm_bool devCoffee_TTF_DEVELOPER_IS_ILL_to_BREAKFAST (  
    void ) [static]
```

4.15.2.20 devCoffee_TTF_IN_OFFICE_to_GET_COFFEE()

```
static pySm_bool devCoffee_TTF_IN_OFFICE_to_GET_COFFEE (  
    void ) [static]
```


4.15.2.21 devCoffee_TTF_IN_OFFICE_to_IN_OFFICE()

```
static pySm_bool devCoffee_TTF_IN_OFFICE_to_IN_OFFICE (  
    void ) [static]
```

4.15.2.22 devCoffee_TTF_IN_OFFICE_to_RELAX_AND_SLEEP()

```
static pySm_bool devCoffee_TTF_IN_OFFICE_to_RELAX_AND_SLEEP (  
    void ) [static]
```

4.15.2.23 devCoffee_TTF_RELAX_AND_SLEEP_to_BREAKFAST_1()

```
static pySm_bool devCoffee_TTF_RELAX_AND_SLEEP_to_BREAKFAST_1 (  
    void ) [static]
```

4.15.2.24 devCoffee_TTF_RELAX_AND_SLEEP_to_BREAKFAST_2()

```
static pySm_bool devCoffee_TTF_RELAX_AND_SLEEP_to_BREAKFAST_2 (  
    void ) [static]
```

4.15.2.25 devCoffee_variableResetFunction()

```
static void devCoffee_variableResetFunction (  
    void ) [static]
```

4.15.3 Variable Documentation

4.15.3.1 coffeein_level_ui8

```
pySm_uint8 coffeein_level_ui8 = 0u
```

4.15.3.2 current_hour_ui8

```
pySm_uint8 current_hour_ui8 = 0u
```

4.15.3.3 devCoffee_activeState

```
devCoffee_activeStateType devCoffee_activeState = devCoffee_UNINITIALIZED_STATE_MACHINE [static]
```

4.15.3.4 devCoffee_inputSignals

```
devCoffee_inputSignalsType* devCoffee_inputSignals [static]
```

4.15.3.5 devCoffee_outputSignals

```
devCoffee_outputSignalsType* devCoffee_outputSignals [static]
```

4.15.3.6 devCoffee_state_BREAKFAST

```
const pySm_stateType devCoffee_state_BREAKFAST [static]
```

Initial value:

```
=
{
    .onEntryState = devCoffee_SF_BREAKFAST_entry,
    .onState = devCoffee_SF_BREAKFAST,
    .onExitState = devCoffee_SF_BREAKFAST_exit
}
```

4.15.3.7 devCoffee_state_DEVELOPER_IS_ILL

```
const pySm_stateType devCoffee_state_DEVELOPER_IS_ILL [static]
```

Initial value:

```
=
{
    .onEntryState = devCoffee_SF_DEVELOPER_IS_ILL_entry,
    .onState = devCoffee_SF_DEVELOPER_IS_ILL,
    .onExitState = PYSM_NULL_PTR
}
```

4.15.3.8 devCoffee_state_GET_COFFEE

```
const pySm_stateType devCoffee_state_GET_COFFEE [static]
```

Initial value:

```
=
{
    .onEntryState = devCoffee_SF_GET_COFFEE_entry,
    .onState = PYSM_NULL_PTR,
    .onExitState = PYSM_NULL_PTR
}
```

4.15.3.9 devCoffee_state_IN_OFFICE

```
const pySm_stateType devCoffee_state_IN_OFFICE [static]
```

Initial value:

```
=
{
    .onEntryState = devCoffee_SF_IN_OFFICE_entry,
    .onState = devCoffee_SF_IN_OFFICE,
    .onExitState = PYSM_NULL_PTR
}
```

4.15.3.10 devCoffee_state_RELAX_AND_SLEEP

```
const pySm_stateType devCoffee_state_RELAX_AND_SLEEP [static]
```

Initial value:

```
=
{
    .onEntryState = devCoffee_SF_RELAX_AND_SLEEP_entry,
    .onState = devCoffee_SF_RELAX_AND_SLEEP,
    .onExitState = PYSM_NULL_PTR
}
```

4.15.3.11 devCoffee_stateMachine_s

```
pySm_stateMachineType devCoffee_stateMachine_s
```

Initial value:

```
=
{
    &devCoffee_state_RELAX_AND_SLEEP,
    &devCoffee_state_RELAX_AND_SLEEP,
    devCoffee_states_pa,
    5u,
    devCoffee_transitions_sa,
    9u,
    PYSM_TRUE,
    devCoffee_variableResetFunction
}
```

4.15.3.12 devCoffee_states_pa

```
const pySm_stateType* devCoffee_states_pa[5] [static]
```

Initial value:

```
=
{
    &devCoffee_state_BREAKFAST,
    &devCoffee_state_IN_OFFICE,
    &devCoffee_state_GET_COFFEE,
    &devCoffee_state_RELAX_AND_SLEEP,
    &devCoffee_state_DEVELOPER_IS_ILL
}
```

4.15.3.13 devCoffee_transitions_sa

```
pySm_stateTransitionType devCoffee_transitions_sa[9] [static]
```

4.15.3.14 productivity_ui8

```
pySm_uint8 productivity_ui8 = 0u
```

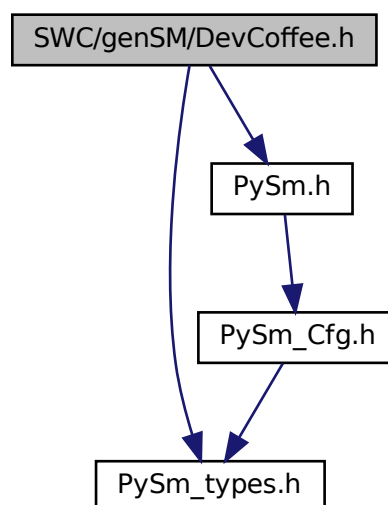
4.16 SWC/genSM/DevCoffee.h File Reference

Header for generated state machine devCoffee Generated 2017-09-27 19:41:10 by PySM - The python state machine generator.

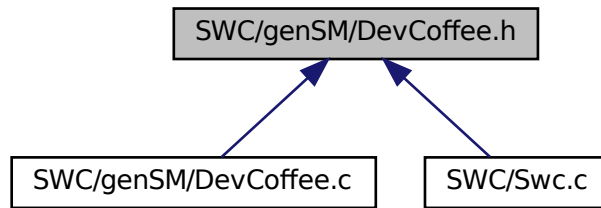
```
#include "PySm_types.h"
```

```
#include "PySm.h"
```

Include dependency graph for DevCoffee.h:



This graph shows which files directly or indirectly include this file:



Data Structures

- struct [devCoffee_inputSignalsType](#)
Structure defining input signals for state machine devCoffee.
- struct [devCoffee_outputSignalsType](#)
Structure defining output signals for state machine devCoffee.

Macros

- `#define` [WORKTIME_PER_DAY_H](#) 8
- `#define` [WAKE_UP_TIME_H](#) 6 /* stand up at 6 o'clock */
- `#define` [GO_HOME_TIME](#) ([WORKTIME_PER_DAY_H](#)+[WAKE_UP_TIME_H](#))

Enumerations

- enum [devCoffee_activeStateType](#) {
[devCoffee_UNINITIALIZED_STATE_MACHINE](#), [DEVCOFFEE_BREAKFAST](#), [DEVCOFFEE_IN_OFFICE](#),
[DEVCOFFEE_GET_COFFEE](#),
[DEVCOFFEE_RELAX_AND_SLEEP](#), [DEVCOFFEE_DEVELOPER_IS_ILL](#) }
Enum for exporting current active state of state machine devCoffee.

Functions

- [pySm_returnType](#) [DevCoffee_mainFunction](#) ([devCoffee_inputSignalsType](#) *, [devCoffee_outputSignalsType](#) *)
Main function of the state machine devCoffee.
- void [DevCoffee_getActiveState](#) ([devCoffee_activeStateType](#) *)
Main function of the state machine devCoffee.

4.16.1 Detailed Description

Header for generated state machine devCoffee Generated 2017-09-27 19:41:10 by PySM - The python state machine generator.

Author

Markus

Date

2017-09-27

4.16.2 Macro Definition Documentation

4.16.2.1 GO_HOME_TIME

```
#define GO_HOME_TIME (WORKTIME_PER_DAY_H+WAKE_UP_TIME_H)
```

4.16.2.2 WAKE_UP_TIME_H

```
#define WAKE_UP_TIME_H 6 /* stand up at 6 o'clock */
```

4.16.2.3 WORKTIME_PER_DAY_H

```
#define WORKTIME_PER_DAY_H 8
```

4.16.3 Enumeration Type Documentation

4.16.3.1 devCoffee_activeStateType

```
enum devCoffee_activeStateType
```

Enum for exporting current active state of state machine devCoffee.

Enumerator

devCoffee_UNINITIALIZED_STATE_MACHINE	
DEVCOFFEE_BREAKFAST	
DEVCOFFEE_IN_OFFICE	
DEVCOFFEE_GET_COFFEE	
DEVCOFFEE_RELAX_AND_SLEEP	
DEVCOFFEE_DEVELOPER_IS_ILL	

4.16.4 Function Documentation

4.16.4.1 DevCoffee_getActiveState()

```
void DevCoffee_getActiveState (
    devCoffee_activeStateType * )
```

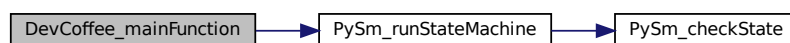
Main function of the state machine devCoffee.

4.16.4.2 DevCoffee_mainFunction()

```
pySm_returnType DevCoffee_mainFunction (
    devCoffee_inputSignalsType * ,
    devCoffee_outputSignalsType * )
```

Main function of the state machine devCoffee.

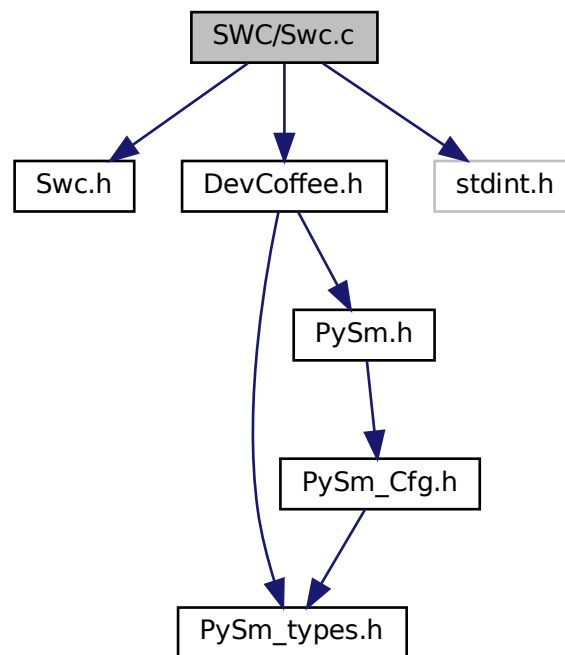
Here is the call graph for this function:



4.17 SWC/Swc.c File Reference

Test-SWC to demonstrate the use of the generated state machine.

```
#include "Swc.h"
#include "DevCoffee.h"
#include <stdint.h>
Include dependency graph for Swc.c:
```



Functions

- void [Swc_main](#) (void)

4.17.1 Detailed Description

Test-SWC to demonstrate the use of the generated state machine.

Author

Markus Burger

Date

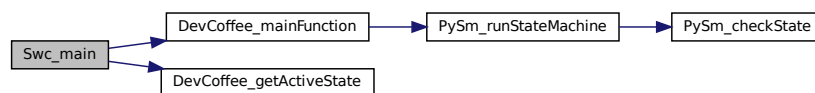
2017-09-11

4.17.2 Function Documentation

4.17.2.1 Swc_main()

```
void Swc_main (  
    void )
```

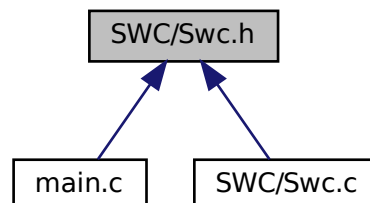
Here is the call graph for this function:



4.18 SWC/Swc.h File Reference

Header file of Test-SWC to demonstrate the use of the generated state machine.

This graph shows which files directly or indirectly include this file:



Functions

- void [Swc_main](#) (void)

4.18.1 Detailed Description

Header file of Test-SWC to demonstrate the use of the generated state machine.

Author

Markus Burger

Date

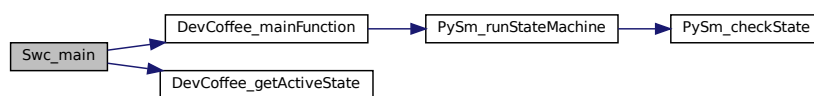
2017-09-11

4.18.2 Function Documentation

4.18.2.1 Swc_main()

```
void Swc_main (
    void )
```

Here is the call graph for this function:



Index

- actualState
 - pySm_stateMachineType, [8](#)
- and_another_output_ui16
 - devCoffee_outputSignalsType, [6](#)
- another_input_ui8
 - devCoffee_inputSignalsType, [5](#)
- coffeein_level_ui8
 - DevCoffee.c, [33](#)
- current_hour_ui8
 - DevCoffee.c, [33](#)
- Debug/LIB/PySm.d, [13](#)
- Debug/LIB/pySm.d, [13](#)
- Debug/SWC/Swc.d, [13](#)
- Debug/SWC/genSM/DevCoffee.d, [13](#)
- Debug/SWC/genSM/Devcoffee.d, [13](#)
- Debug/SWC/genSM/devCoffee.d, [13](#)
- Debug/SWC/sm1.d, [13](#)
- Debug/SWC/swc.d, [13](#)
- Debug/main.d, [13](#)
- destinationState
 - pySm_stateTransitionType, [10](#)
- DevCoffee.c
 - coffeein_level_ui8, [33](#)
 - current_hour_ui8, [33](#)
 - devCoffee_SF_BREAKFAST_entry, [30](#)
 - devCoffee_SF_BREAKFAST_exit, [30](#)
 - devCoffee_SF_BREAKFAST, [30](#)
 - devCoffee_SF_DEVELOPER_IS_ILL_entry, [30](#)
 - devCoffee_SF_DEVELOPER_IS_ILL, [30](#)
 - devCoffee_SF_GET_COFFEE_entry, [31](#)
 - devCoffee_SF_IN_OFFICE_entry, [31](#)
 - devCoffee_SF_IN_OFFICE, [31](#)
 - devCoffee_SF_RELAX_AND_SLEEP_entry, [31](#)
 - devCoffee_SF_RELAX_AND_SLEEP, [31](#)
 - devCoffee_TAF_GET_COFFEE_to_IN_OFFICE, [31](#)
 - devCoffee_TAF_IN_OFFICE_to_IN_OFFICE, [31](#)
 - devCoffee_TAF_IN_OFFICE_to_RELAX_AND_SLEEP, [32](#)
 - devCoffee_TAF_RELAX_AND_SLEEP_to_BREAKFAST, [32](#)
 - devCoffee_TTF_BREAKFAST_to_DEVELOPER_IS_ILL, [32](#)
 - devCoffee_TTF_BREAKFAST_to_IN_OFFICE, [32](#)
 - devCoffee_TTF_DEVELOPER_IS_ILL_to_BREAKFAST, [32](#)
 - devCoffee_TTF_IN_OFFICE_to_GET_COFFEE, [32](#)
- devCoffee_TTF_IN_OFFICE_to_IN_OFFICE, [32](#)
- devCoffee_TTF_IN_OFFICE_to_RELAX_AND_SLEEP, [33](#)
- devCoffee_TTF_RELAX_AND_SLEEP_to_BREAKFAST_1, [33](#)
- devCoffee_TTF_RELAX_AND_SLEEP_to_BREAKFAST_2, [33](#)
- devCoffee_activeState, [34](#)
- DevCoffee_getActiveState, [29](#)
- devCoffee_inputSignals, [34](#)
- DevCoffee_mainFunction, [29](#)
- devCoffee_outputSignals, [34](#)
- devCoffee_state_BREAKFAST, [34](#)
- devCoffee_state_DEVELOPER_IS_ILL, [34](#)
- devCoffee_state_GET_COFFEE, [34](#)
- devCoffee_state_IN_OFFICE, [35](#)
- devCoffee_state_RELAX_AND_SLEEP, [35](#)
- devCoffee_stateMachine_s, [35](#)
- devCoffee_states_pa, [35](#)
- devCoffee_transitions_sa, [36](#)
- devCoffee_variableResetFunction, [33](#)
- productivity_ui8, [36](#)
- DevCoffee.h
 - devCoffee_activeStateType, [38](#)
 - DevCoffee_getActiveState, [39](#)
 - DevCoffee_mainFunction, [39](#)
 - GO_HOME_TIME, [38](#)
 - WAKE_UP_TIME_H, [38](#)
 - WORKTIME_PER_DAY_H, [38](#)
- devCoffee_SF_BREAKFAST_entry
 - DevCoffee.c, [30](#)
- devCoffee_SF_BREAKFAST_exit
 - DevCoffee.c, [30](#)
- devCoffee_SF_BREAKFAST
 - DevCoffee.c, [30](#)
- devCoffee_SF_DEVELOPER_IS_ILL_entry
 - DevCoffee.c, [30](#)
- devCoffee_SF_DEVELOPER_IS_ILL
 - DevCoffee.c, [30](#)
- devCoffee_SF_GET_COFFEE_entry
 - DevCoffee.c, [31](#)
- devCoffee_SF_IN_OFFICE_entry
 - DevCoffee.c, [31](#)
- devCoffee_SF_IN_OFFICE
 - DevCoffee.c, [31](#)
- devCoffee_SF_RELAX_AND_SLEEP_entry
 - DevCoffee.c, [31](#)
- devCoffee_SF_RELAX_AND_SLEEP
 - DevCoffee.c, [31](#)

- devCoffee_TAF_GET_COFFEE_to_IN_OFFICE
 - DevCoffee.c, [31](#)
- devCoffee_TAF_IN_OFFICE_to_IN_OFFICE
 - DevCoffee.c, [31](#)
- devCoffee_TAF_IN_OFFICE_to_RELAX_AND_SLEEP
 - DevCoffee.c, [32](#)
- devCoffee_TAF_RELAX_AND_SLEEP_to_BREAKFAST
 - AST
 - DevCoffee.c, [32](#)
- devCoffee_TTF_BREAKFAST_to_DEVELOPER_IS_ILL
 - ILL
 - DevCoffee.c, [32](#)
- devCoffee_TTF_BREAKFAST_to_IN_OFFICE
 - DevCoffee.c, [32](#)
- devCoffee_TTF_DEVELOPER_IS_ILL_to_BREAKFAST
 - ST
 - DevCoffee.c, [32](#)
- devCoffee_TTF_IN_OFFICE_to_GET_COFFEE
 - DevCoffee.c, [32](#)
- devCoffee_TTF_IN_OFFICE_to_IN_OFFICE
 - DevCoffee.c, [32](#)
- devCoffee_TTF_IN_OFFICE_to_RELAX_AND_SLEEP
 - DevCoffee.c, [33](#)
- devCoffee_TTF_RELAX_AND_SLEEP_to_BREAKFAST
 - AST_1
 - DevCoffee.c, [33](#)
- devCoffee_TTF_RELAX_AND_SLEEP_to_BREAKFAST
 - AST_2
 - DevCoffee.c, [33](#)
- devCoffee_activeState
 - DevCoffee.c, [34](#)
- devCoffee_activeStateType
 - DevCoffee.h, [38](#)
- DevCoffee_getActiveState
 - DevCoffee.c, [29](#)
 - DevCoffee.h, [39](#)
- devCoffee_inputSignals
 - DevCoffee.c, [34](#)
- devCoffee_inputSignalsType, [5](#)
 - another_input_ui8, [5](#)
 - developer_is_ill_HA_b, [5](#)
- DevCoffee_mainFunction
 - DevCoffee.c, [29](#)
 - DevCoffee.h, [39](#)
- devCoffee_outputSignals
 - DevCoffee.c, [34](#)
- devCoffee_outputSignalsType, [6](#)
 - and_another_output_ui16, [6](#)
 - developer_is_productive_HA_b, [6](#)
- devCoffee_state_BREAKFAST
 - DevCoffee.c, [34](#)
- devCoffee_state_DEVELOPER_IS_ILL
 - DevCoffee.c, [34](#)
- devCoffee_state_GET_COFFEE
 - DevCoffee.c, [34](#)
- devCoffee_state_IN_OFFICE
 - DevCoffee.c, [35](#)
- devCoffee_state_RELAX_AND_SLEEP
 - DevCoffee.c, [35](#)
- devCoffee_stateMachine_s
 - DevCoffee.c, [35](#)
- devCoffee_states_pa
 - DevCoffee.c, [35](#)
- devCoffee_transitions_sa
 - DevCoffee.c, [36](#)
- devCoffee_variableResetFunction
 - DevCoffee.c, [33](#)
- developer_is_ill_HA_b
 - devCoffee_inputSignalsType, [5](#)
- developer_is_productive_HA_b
 - devCoffee_outputSignalsType, [6](#)
- entryState
 - pySm_stateMachineType, [8](#)
- GO_HOME_TIME
 - DevCoffee.h, [38](#)
- LIB/PySm.c, [13](#)
- LIB/PySm.h, [17](#)
- LIB/PySm_Cfg.h, [21](#)
- LIB/PySm_types.h, [23](#)
- main
 - main.c, [27](#)
- main.c, [26](#)
 - main, [27](#)
- numberOfStates
 - pySm_stateMachineType, [8](#)
- numberOfTransitions
 - pySm_stateMachineType, [8](#)
- onEntryState
 - pySm_stateType, [11](#)
- onExitState
 - pySm_stateType, [11](#)
- onState
 - pySm_stateType, [11](#)
- PYSM_FALSE
 - PySm_types.h, [24](#)
- PYSM_MAX_NO_OF_TRANSITIONS_PER_STATE
 - PySm_Cfg.h, [22](#)
- PYSM_NULL_PTR
 - PySm_types.h, [24](#)
- PYSM_STD_OFF
 - PySm_types.h, [24](#)
- PYSM_STD_ON
 - PySm_types.h, [24](#)
- PYSM_TRUE
 - PySm_types.h, [24](#)
- productivity_ui8
 - DevCoffee.c, [36](#)
- PySm.c
 - PySm_checkState, [14](#)
 - PySm_resetStateMachine, [15](#)
 - PySm_runStateMachine, [16](#)

PySm.h
 PySm_resetStateMachine, 20
 pySm_returnType, 19
 PySm_runStateMachine, 20
 pySm_stateFunction, 18
 pySm_stateMachineResetFunction, 18
 pySm_transitionActionFunction, 19
 pySm_transitionPriorityType, 19
 pySm_transitionTestFunction, 19
 PySm_Cfg.h
 PYSM_MAX_NO_OF_TRANSITIONS_PER_STATE, 22
 pySm_bool
 PySm_types.h, 25
 PySm_checkState
 PySm.c, 14
 pySm_int16
 PySm_types.h, 25
 pySm_int32
 PySm_types.h, 25
 pySm_int64
 PySm_types.h, 26
 pySm_int8
 PySm_types.h, 25
 PySm_resetStateMachine
 PySm.c, 15
 PySm.h, 20
 pySm_returnType
 PySm.h, 19
 PySm_runStateMachine
 PySm.c, 16
 PySm.h, 20
 pySm_stateFunction
 PySm.h, 18
 pySm_stateMachineResetFunction
 PySm.h, 18
 pySm_stateMachineType, 7
 actualState, 8
 entryState, 8
 numberOfStates, 8
 numberOfTransitions, 8
 resetVariables, 8
 runEntryOfInitialState_b, 8
 states, 9
 transitions, 9
 pySm_stateTransitionType, 9
 destinationState, 10
 sourceState, 10
 transitionAction, 10
 transitionPriority, 10
 transitionTest, 10
 pySm_stateType, 11
 onEntryState, 11
 onExitState, 11
 onState, 11
 pySm_transitionActionFunction
 PySm.h, 19
 pySm_transitionPriorityType
 PySm.h, 19
 pySm_transitionTestFunction
 PySm.h, 19
 PySm_types.h
 PYSM_FALSE, 24
 PYSM_NULL_PTR, 24
 PYSM_STD_OFF, 24
 PYSM_STD_ON, 24
 PYSM_TRUE, 24
 pySm_bool, 25
 pySm_int16, 25
 pySm_int32, 25
 pySm_int64, 26
 pySm_int8, 25
 pySm_uint16, 25
 pySm_uint32, 25
 pySm_uint64, 26
 pySm_uint8, 25
 pySm_uint16
 PySm_types.h, 25
 pySm_uint32
 PySm_types.h, 25
 pySm_uint64
 PySm_types.h, 26
 pySm_uint8
 PySm_types.h, 25
 resetVariables
 pySm_stateMachineType, 8
 runEntryOfInitialState_b
 pySm_stateMachineType, 8
 SWC/Swc.c, 40
 SWC/Swc.h, 41
 SWC/genSM/DevCoffee.c, 27
 SWC/genSM/DevCoffee.h, 36
 sourceState
 pySm_stateTransitionType, 10
 states
 pySm_stateMachineType, 9
 Swc.c
 Swc_main, 41
 Swc.h
 Swc_main, 42
 Swc_main
 Swc.c, 41
 Swc.h, 42
 transitionAction
 pySm_stateTransitionType, 10
 transitionPriority
 pySm_stateTransitionType, 10
 transitionTest
 pySm_stateTransitionType, 10
 transitions
 pySm_stateMachineType, 9
 WAKE_UP_TIME_H
 DevCoffee.h, 38

WORKTIME_PER_DAY_H
DevCoffee.h, [38](#)