Sensirion Module API Specification

This section describes the API from the Task Scheduler module Function

**Type Definition**

Init Status for STH11 sensor

Enumeration

SHT\_InitState

* SHT\_Init
* SHT\_NoInit

Acquisition States from STH11 sensor

Enumeration

SHT\_TaskState

* SHT\_Task\_Busy
* SHT\_Task \_OK
* SHT\_Task\_Not\_OK

TWI module Status

Enumeration

TWI\_ModuleStatus

* TWIHS\_SUCCESS
* TWIHS\_INVALID\_ARGUMENT
* TWIHS\_ARBITRATION\_LOST
* TWIHS\_NO\_CHIP\_FOUND
* TWIHS\_RECEIVE\_OVERRUN
* TWIHS\_RECEIVE\_NACK
* TWIHS\_SEND\_OVERRUN
* TWIHS\_SEND\_NACK
* TWIHS\_BUSY
* TWIHS\_ERROR\_TIMEOUT

#define TWIHS\_NO\_CHIP\_FOUND 3

#define TWIHS\_ERROR\_TIMEOUT 9

**Main Function Definitions**

|  |  |
| --- | --- |
| ***Service Name*** | SHT\_Init |
| ***Syntax*** | SHT\_InitState SHT\_Init(void) |
| ***Sync/Async*** | Asynchronous |
| ***Reentrancy*** | Non-reentrant |
| ***Param (in)*** | None |
| ***Param (out)*** | None |
| ***Return Value*** | SHT\_InitState Status |
| ***Description*** | This function performs the basic initialization of the I2C module configured to use SHT11 sensirion sensor |

|  |  |
| --- | --- |
| ***Service Name*** | SHT\_DeInit |
| ***Syntax*** | SHT\_InitState SHT\_DeInit(void) |
| ***Sync/Async*** | Asynchronous |
| ***Reentrancy*** | Non-reentrant |
| ***Param (in)*** | None |
| ***Param (out)*** | None |
| ***Return Value*** | SHT\_InitState Status |
| ***Description*** | This function performs deinitialization of the I2C module |

|  |  |
| --- | --- |
| ***Service Name*** | SHT\_GetTemperatureRaw |
| ***Syntax*** | SHT\_TaskState SHT\_GetTemperatureRaw (uint16\*pTemp) |
| ***Sync/Async*** | Asynchronous |
| ***Reentrancy*** | Non-reentrant |
| ***Param (in)*** | uint16\* pTemp |
| ***Param (out)*** | uint16\* pTemp |
| ***Return Value*** | SHT\_TaskState AcqStatus |
| ***Description*** | This function performs load the Temp message for SHT11 module and send it throw I2C module |

|  |  |
| --- | --- |
| ***Service Name*** | SHT\_ReadHumidityRaw |
| ***Syntax*** | SHT\_TaskState SHT\_ ReadHumidityRaw (uint16\* pHumidity) |
| ***Sync/Async*** | Asynchronous |
| ***Reentrancy*** | Non-reentrant |
| ***Param (in)*** | uint16\* pHumidity |
| ***Param (out)*** | uint16\* pHumidity |
| ***Return Value*** | SHT\_TaskState AcqStatus |
| ***Description*** | This function performs load the Humidity message for SHT11 module and send it throw I2C module |

|  |  |
| --- | --- |
| ***Service Name*** | twihs\_master\_init |
| ***Syntax*** | TWI\_ModuleStatus twihs\_master\_init (void) |
| ***Sync/Async*** | Asynchronous |
| ***Reentrancy*** | Non-reentrant |
| ***Param (in)*** | None |
| ***Param (out)*** | None |
| ***Return Value*** | Module Init Module Status |
| ***Description*** | This function is called by SHT\_Init to load driver module configuration |

|  |  |
| --- | --- |
| ***Service Name*** | twihs\_master\_init |
| ***Syntax*** | SHT\_ModuleStatus I2C\_ twihs\_master\_init (void) |
| ***Sync/Async*** | Asynchronous |
| ***Reentrancy*** | Non-reentrant |
| ***Param (in)*** | None |
| ***Param (out)*** | None |
| ***Return Value*** | Module DeInit Module Status |
| ***Description*** | This function is called by SHT\_DeInit to load driver module configuration |

|  |  |
| --- | --- |
| ***Service Name*** | twihs\_master\_write |
| ***Syntax*** | SHT\_ModuleStatus twihs\_master\_write (uint8\* pMessage,uint8 Size) |
| ***Sync/Async*** | Asynchronous |
| ***Reentrancy*** | Non-reentrant |
| ***Param (in)*** | Uint8 size |
| ***Param (out)*** | uint16\* pMessage |
| ***Return Value*** | SHT\_ModuleStatus SHT\_ModuleStatus |
| ***Description*** | This function is called during the data request |

|  |  |
| --- | --- |
| ***Service Name*** | twihs\_master\_read |
| ***Syntax*** | SHT\_ModuleStatus twihs\_master\_read (uint8\* pMessage,uint8 Size) |
| ***Sync/Async*** | Asynchronous |
| ***Reentrancy*** | Non-reentrant |
| ***Param (in)*** | Uint8 size |
| ***Param (out)*** | uint16\* pMessage |
| ***Return Value*** | SHT\_\_ModuleStatus SHT\_ModuleStatus |
| ***Description*** | This function is called during the Rx interruption and process the information for the module |

