Operating Systems Design for Embedded Environments

**Practice 2 0.0 OS Scheduler**

|  |  |
| --- | --- |
| **Project:** | **OS Scheduler** |
| **Team:** | **Team 4** |
| **Date:** | **01/03/2014** |
| **Comments:** | **This is just an initial draft, can be changed if needed.** |

**Table of Contents**

[Operating Systems Design for Embedded Environments 1](#_Toc381448025)

[1. SW Conceptual design 3](#_Toc381448026)

[2 Function Description and Dynamic Behavior 3](#_Toc381448027)

[*2.1* Function *void SchM\_Init (SchM\_TaskConfigType \*SchM\_Config)* 3](#_Toc381448028)

[*2.2* Function *void SchM\_Deinit (void)* 3](#_Toc381448029)

[*2.3* Function *void SchM\_Start (void)* 3](#_Toc381448030)

[*2.4* Function *void SchM\_OsTick (void)* 4](#_Toc381448031)

[*2.5* Function *void SchM\_Background(void )* 4](#_Toc381448032)

[*2.6* Scheduler task callback functions 4](#_Toc381448033)

# SW Conceptual design

# Function Description and Dynamic Behavior

## Function void SchM\_Init (SchM\_TaskConfigType \*SchM\_Config)

|  |  |  |
| --- | --- | --- |
| **Description** | *Void SchM\_Init(SchM\_TaskConfigType \*SchM\_Config);* | |
| **Return Value** | *None* | |
| **Precondition** | *None* | |
| **Parameters** | *SchM\_Config* | *Pointer to the module’s configuration structure* |
| **Error Conditions** |  | |

**Dynamic Behavior**

SchM\_Init function shall allocate and initialize the resources to be used by the Scheduler Module.

## Function void SchM\_Deinit (void)

|  |  |
| --- | --- |
| **Description** | *Void SchM\_Deinit(void);* |
| **Return Value** | *None* |
| **Precondition** | *None* |
| **Post condition** | *None* |
| **Error Conditions** |  |

**Dynamic Behavior**

SchM\_Deinit function shall finalize and free the resources used by the Scheduler Module.

## Function void SchM\_Start (void)

|  |  |
| --- | --- |
| **Description** | *Void SchM\_Start(void);* |
| **Return Value** | *None* |
| **Precondition** | *None* |
| **Post condition** | *None* |
| **Error Conditions** |  |

**Dynamic Behavior**

Mcu\_DistributePllClock API enables PLL as the main clock source.

## Function void SchM\_OsTick (void)

|  |  |
| --- | --- |
| **Description** | *Void SchM\_OsTick(void);* |
| **Return Value** | *None* |
| **Precondition** | *None* |
| **Post condition** | *None* |
| **Error Conditions** |  |

**Dynamic Behavior**

SchM\_OsTick is a callback function for the base timer Tick.

## Function void SchM\_Background(void )

|  |  |
| --- | --- |
| **Description** | *Void SchM\_Background(void );* |
| **Return Value** | *None* |
| **Parameters** | None |
| **Post condition** | *None* |
| **Error Conditions** |  |

**Dynamic Behavior**

SchM\_Background function runs when no tasks are scheduled.

## Scheduler task callback functions

|  |  |
| --- | --- |
| **Task** | **Description & Period** |
| *(void) SchM\_Tsk\_1ms (void);* | *Callback function for 1ms* |
| *(void) SchM\_Tsk\_4ms (void);* | *Callback function for 4ms* |
| *(void) SchM\_Tsk\_8ms (void);* | *Callback function for 8ms* |
| *(void) SchM\_Tsk\_16ms (void);* | *Callback function for 16ms* |
| *(void) SchM\_Tsk\_32ms (void);* | *Callback function for 32ms* |
| *(void) SchM\_Tsk\_64ms (void);* | *Callback function for 64ms* |

**Dynamic Behavior**

Callback functions for periods required.