Operating Systems Design for Embedded Environments

**Practice 3 2.0 OSTask\_Manager**

|  |  |
| --- | --- |
| **Project:** | **OSTask\_Manager** |
| **Team:** | **Team 4** |
| **Date:** | **30/03/2014** |
| **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 *Status\_Type ActivateTask(TaskType taskID)* 3](#_Toc381448028)

[*2.2* Function *Status\_Type TerminateTask (void)* 3](#_Toc381448029)

[*2.3* Function *Status\_Type GetTaskID(TaskRefType taskIDRef)* 3](#_Toc381448030)

[*2.4* Function *Status\_Type GetTaskState(TaskType taskID, TaskStateRefType stateRef)* 4](#_Toc381448031)

[*2.5* Function *void Dispatcher(void)* 4](#_Toc381448032)

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

# SW Conceptual design











# Function Description and Dynamic Behavior

## Function Status\_Type ActivateTask(TaskType taskID)

|  |  |  |
| --- | --- | --- |
| **Description** | *Status\_Type ActivateTask(TaskType taskID)* | |
| **Return Value** | *Status* | |
| **Precondition** | *None* | |
| **Parameters** | *taskID* | *TaskType* |
| **Error Conditions** |  | |

**Dynamic Behavior**

Set a task in Ready inside the Dispatcher Array.

## Function Status\_Type TerminateTask (void)

|  |  |
| --- | --- |
| **Description** | *Status\_Type TerminateTask (void)* |
| **Return Value** | *None* |
| **Precondition** | *None* |
| **Post condition** | *None* |
| **Error Conditions** |  |

**Dynamic Behavior**

Set a task in Suspended inside the Dispatcher Array

## Function Status\_Type GetTaskID(TaskRefType taskIDRef)

|  |  |
| --- | --- |
| **Description** | *Status\_Type GetTaskID(TaskRefType taskIDRef)* |
| **Return Value** | *Status\_Type* |
| **Precondition** | *None* |
| **Post condition** | *None* |
| **Error Conditions** |  |

**Dynamic Behavior**

Return the ID of the Task Running, in case no task is running return 0xFFFF.

## Function Status\_Type GetTaskState(TaskType taskID, TaskStateRefType stateRef)

|  |  |
| --- | --- |
| **Description** | *Status\_Type GetTaskState(TaskType taskID, TaskStateRefType stateRef)* |
| **Return Value** | *Status\_Type* |
| **Precondition** | *None* |
| **Post condition** | *None* |
| **Error Conditions** |  |

**Dynamic Behavior**

Returns the State of the ID specified in the parameter taskID

## Function void Dispatcher(void)

|  |  |
| --- | --- |
| **Description** | *void Dispatcher(void)* |
| **Return Value** | *None* |
| **Parameters** | None |
| **Post condition** | *None* |
| **Error Conditions** |  |

**Dynamic Behavior**

Run the hightest priority task in ready.

## 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 requested.