Operating Systems Design for Embedded Environments

**Practica 1 0.0 OS Tick**

|  |  |
| --- | --- |
| **Project:** | **OS Tick** |
| **Team:** | **Team 4** |
| **Date:** | **10/02/2014** |
| **Comments:** | **This is just a initial draft, can be changed if needed.** |

**Table of Contents**

[Operating Systems Design for Embedded Environments 1](#_Toc379621831)

[1. SW Conceptual design 3](#_Toc379621832)

[2 Function Description and Dynamic Behavior 4](#_Toc379621833)

[*2.1* Function *void Mcu\_Init (void)* 4](#_Toc379621834)

[*2.2* Function *void Mcu\_InitClock (void)* 4](#_Toc379621835)

[*2.3* Function *void Mcu\_DistributePllClock (void)* 4](#_Toc379621836)

# SW Conceptual design



# Function Description and Dynamic Behavior

## Function void Mcu\_Init (void)

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

**Dynamic Behavior**

Mcu\_Init API initializes the PLL or XTAL frequency configuration.

## Function void Mcu\_InitClock (void)

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

**Dynamic Behavior**

Mcu\_InitClock API enables XTAL Clock as the main clock source.

## Function void Mcu\_DistributePllClock (void)

|  |  |
| --- | --- |
| **Description** | *Void Mcu\_DistributePllClock(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 Gpt\_Init (void)

|  |  |
| --- | --- |
| **Description** | *Void Gpt\_Init(const Gpt\_ConfigType\* ConfigPtr);* |
| **Return Value** | *None* |
| **Parameters** | *ConfigPtr : pointer to the initial configuration structure* |
| **Post condition** | *None* |
| **Error Conditions** |  |

**Dynamic Behavior**

Gpt\_Init API initializes PIT according *ConfigPtr* configuration.

## Function void Gpt\_StartTimer( Gpt\_ChannelType Channel, Gpt\_ValueType Value )

|  |  |
| --- | --- |
| **Description** | *void Gpt\_StartTimer( Gpt\_ChannelType Channel, Gpt\_ValueType Value );* |
| **Return Value** | *None* |
| **Parameters** | Channel*: pointer to the initial configuration structure* |
| **Post condition** | *None* |
| **Error Conditions** |  |

**Dynamic Behavior**

Starst/Run the GPT timer desired in the parameters