



SOS

By.

Ahmed Atef

Sprints.ai

High Level Design	2
Layered Architecture	2
→ APP Layer	2
→ SERVICE Layer	2
→ HAL Layer	2
→ MCAL Layer	3
→ COMMON Layer	3
Module Description	3
→ APP Layer	3
→ SERVICE Layer	3
→ HAL Layer	3
→ MCAL Layer	3
→ COMMON Layer	3
Sequence Diagram	4
State Machine Diagram	5
Class Diagram	Error! Bookmark not defined.
Drivers' Documentation (APIs)	6

Project Introduction

This project is aiming to deliver a SOS -Small Operating System- which will manage the scheduling of some tasks. The project will be resembling RTOS and for the delivery it will test on some output/input modules which will toggle LEDs at different periodicities and will be checking on the buttons' states too. 1.1. Project

Components

- ATmega32 microcontroller
- 2 LEDS
- 2 Buttons

So, I will be using ATmega32 microcontroller for generating timer interrupts which will be used as system tick, will create two different tasks for the LEDs, 1 for STOP button, and START button will be checking on it in the super loop of the application.

Main Application Flow

Implement an application that calls the SOS module and use 2 tasks -

Task 1: Toggle LED_0 (Every 300 Milli-Seconds)

Task 2: Toggle LED_1 (Every 500 Milli-Seconds)

Make sure that these tasks occur periodically and forever

When pressing PBUTTON0, the SOS will stop

When Pressing PBUTTON1, the SOS will run

High Level Design

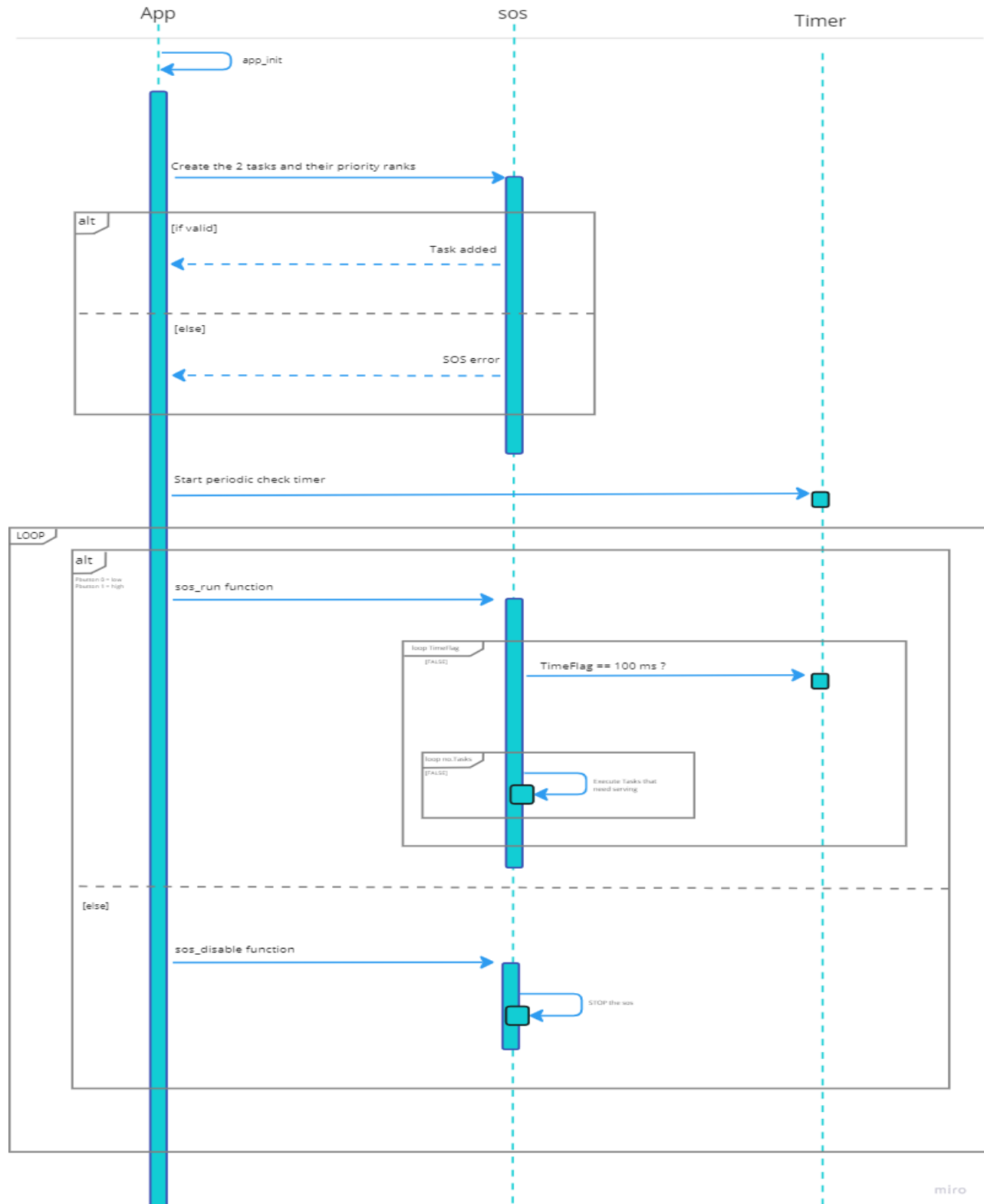
Layered Architecture

- APP Layer
- SERVICE Layer
- HAL Layer

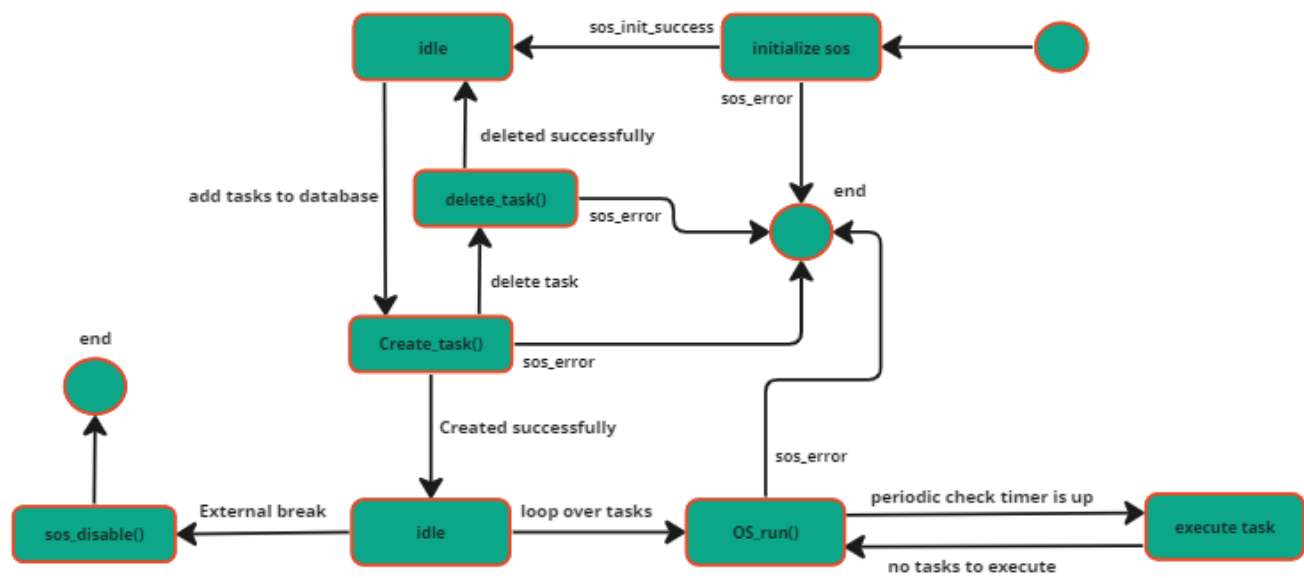
- MCAL Layer
- COMMON Layer

Module Description

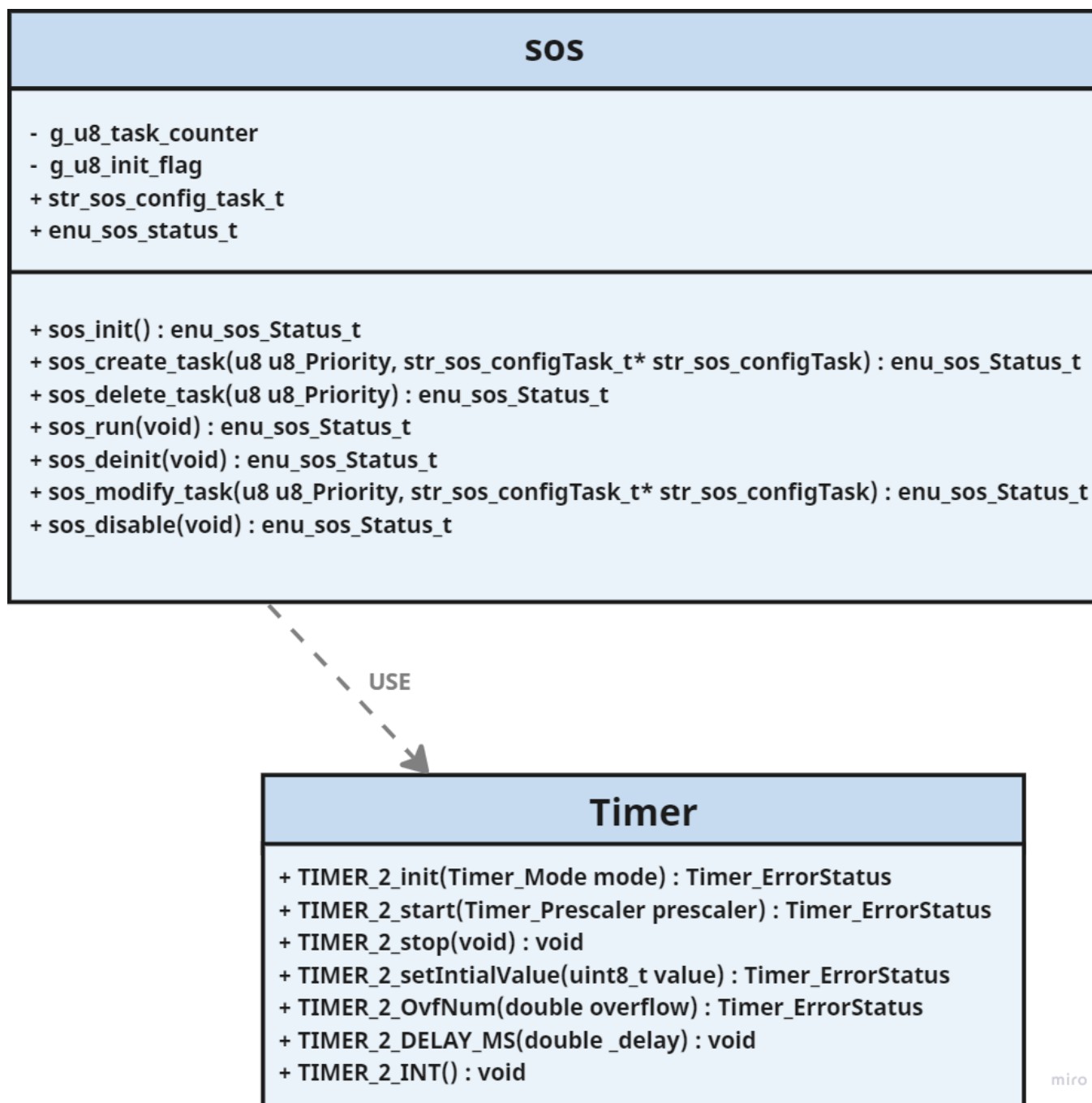
- APP Layer
- SERVICE Layer
- HAL Layer
- MCAL Layer
- COMMON Layer

Sequence Diagram (FOR HQ -> [S.D](#))

State Machine Diagram



Class Diagram



<pre>typedef struct { /* Selecting the periodicity of the selected task */ u8 u8Periodicity; /* Assigning the pointer to the task function */ void (*pfTask) (void); } strOSConfigTask_t;</pre>	<pre>typedef Enum{ OS_OK, OS_ERROR, PRIORITY_EMPTY, PRIORITY_FULL, OS_INIT, OS_TASK_ADDED, OS_TASK_DELETED } enuOSErrorStatus_t;</pre>
---	---

Syntax	enuOSErrorStatus_t OS_Init ()
Description	A function to initialize the timer used in OS
Sync\Async	Synchronous
Reentrancy	Reentrant
Parameters (in)	void
Parameters (out)	None
Return value	<pre>typedef Enum{ OS_OK, OS_ERROR, PRIORITY_EMPTY, PRIORITY_FULL, OS_INIT, OS_TASK_ADDED, OS_TASK_DELETED } enuOSErrorStatus_t;</pre>

Syntax	enuOSErrorStatus_t OS_CreateTask (u8 u8Priority, strOSConfigTask_t* strOSConfigTask_t)
Description	A function to create a certain task
Sync\Async	Synchronous
Reentrancy	Reentrant
Parameters (in)	*strOSConfigTask_t
Parameters (out)	None
Return value	<pre>typedef Enum{ OS_OK, OS_ERROR, PRIORITY_EMPTY, PRIORITY_FULL, OS_INIT, OS_TASK_ADDED, OS_TASK_DELETED } enuOSErrorStatus_t;</pre>

Syntax	enuOSErrorStatus_t OS_DeleteTask (u8 u8Priority)
Description	A function to delete a certain task
Sync\Async	Synchronous
Reentrancy	Reentrant

Parameters (in)	<code>*strOSConfigTask_t</code>
Parameters (out)	None
Return value	<pre>typedef Enum{ OS_OK, OS_ERROR, PRIORITY_EMPTY, PRIORITY_FULL, OS_INIT, OS_TASK_ADDED, OS_TASK_DELETED } enuOSErrorStatus_t;</pre>

Syntax	<code>enuOSErrorStatus_t OS_modify_task (u8 u8Priority, strOSConfigTask_t*</code> <code>strOSConfigTask_t)</code>
Description	A function to modify a certain task
Sync\Async	Synchronous
Reentrancy	Reentrant
Parameters (in)	<code>*strOSConfigTask_t</code>
Parameters (out)	None
Return value	<pre>typedef Enum{ OS_OK, OS_ERROR, PRIORITY_EMPTY, PRIORITY_FULL, OS_INIT, OS_TASK_ADDED, OS_TASK_DELETED } enuOSErrorStatus_t;</pre>

Syntax	<code>void OS_Run(void)</code>
Description	A function to run the OS
Sync\Async	Synchronous
Reentrancy	Reentrant
Parameters (in)	void
Parameters (out)	None
Return value	void

Syntax	<code>void OS_disable ()</code>
Description	A function to disable running of OS
Sync\Async	Synchronous
Reentrancy	Reentrant
Parameters (in)	void
Parameters (out)	None
Return value	void

SOS