

# Description:

You are required to design a small OS with a priority based preemptive scheduler based on time-triggered.

## Detailed Requirements

### 1. Read System Requirement Specifications

1. This OS is using a preemptive priority-based scheduler
2. Design the following functions/APIs:
  1. **sos\_init** function, this function will initialize the SOS database

Function Name	sos_init
Syntax	enu_system_status_t sos_init (void);
Sync/Async	Synchronous
Reentrancy	Non-Reentrant
Parameters (in):	None
Parameters (out):	None
Parameters (in, out):	None
Return:	SOS_STATUS_SUCCESS: In case of Successful Operation SOS_STATUS_INVALID_STATE: In case The SOS is already Initialized

2. **sos\_deinit** function, this function will reset the SOS database to invalid values

Function Name	sos_deinit
Syntax	enu_system_status_t sos_deinit (void);
Sync/Async	Synchronous
Reentrancy	Non-Reentrant
Parameters (in):	None
Parameters (out):	None
Parameters (in, out):	None
Return:	TMU_STATUS_SUCCESS: In case of Successful Operation SOS_STATUS_INVALID_STATE: In case The SOS is already De-Initialized or was not initialized previously.

3. **sos\_create\_task** API, this API will create a new task and add it to the SOS database
4. **sos\_delete\_task** API, this API will delete an existing task from the SOS database.
5. **sos\_modify\_task** API, this API will modify existing task parameters in the SOS database

6. **sos\_run** API, this API will run the small scheduler
7. **sos\_disable** API, this API will stop the scheduler

## 2. Main Application Flow

1. Implement an application that calls the SOS module and use 2 tasks
  1. Task 1: Toggle LED\_0 (Every 3 Milli-Seconds)
  2. Task 2: Toggle LED\_1 (Every 5 Milli-Seconds)
2. Make sure that these tasks occur periodically and forever
3. When pressing **PBUTTON0**, the SOS will stop
4. When Pressing **PBUTTON1**, the SOS will run

## 3. Prepare your design

1. Create a PDF file with the name **Small OS Design**
2. The design document should contain the below fields
  1. Cover Page
  2. Table of content
  3. Project introduction
  4. High Level Design
    1. Layered architecture
    2. Modules Descriptions
    3. Drivers' documentation
    4. UML
    5. Sequence diagram and/or flowchart
  5. Low Level Design
    1. Provide the flowchart for each function in each module
    2. Pre-compiling configurations for each module
    3. Linking configurations for each module

# Delivery

1. Deliver the Design Document
2. English Video recording 5 minutes maximum discuss your design