Small Operating System

Prepared by:Mohab Ahmed

Table of Contents

Project Description

This is an application that calls the SOS module and use 2 tasks:

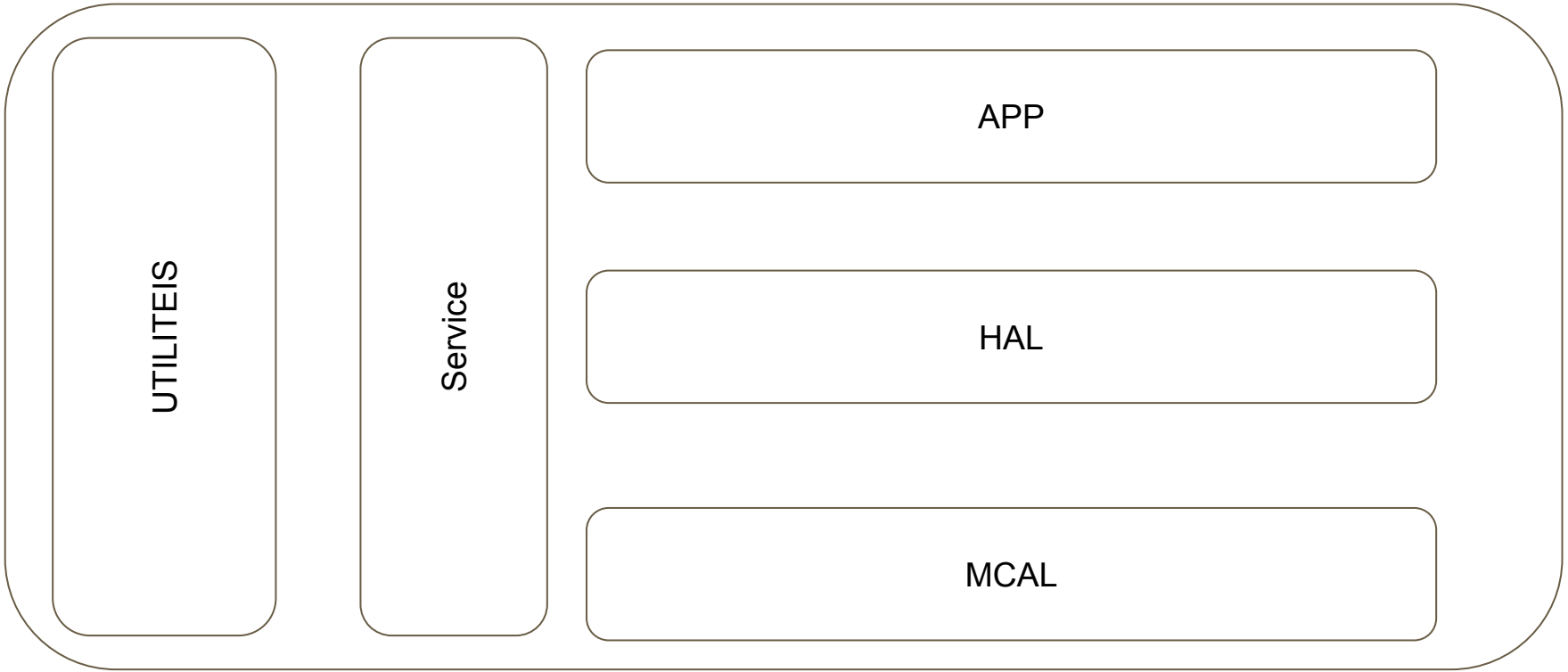
1. Task 1:Toggle LED_0 (Every 3 Milli-seconds)
2. Task 2:Toggle LED_1 (Every 5 Milli-seconds)

These tasks will occur periodically and forever.

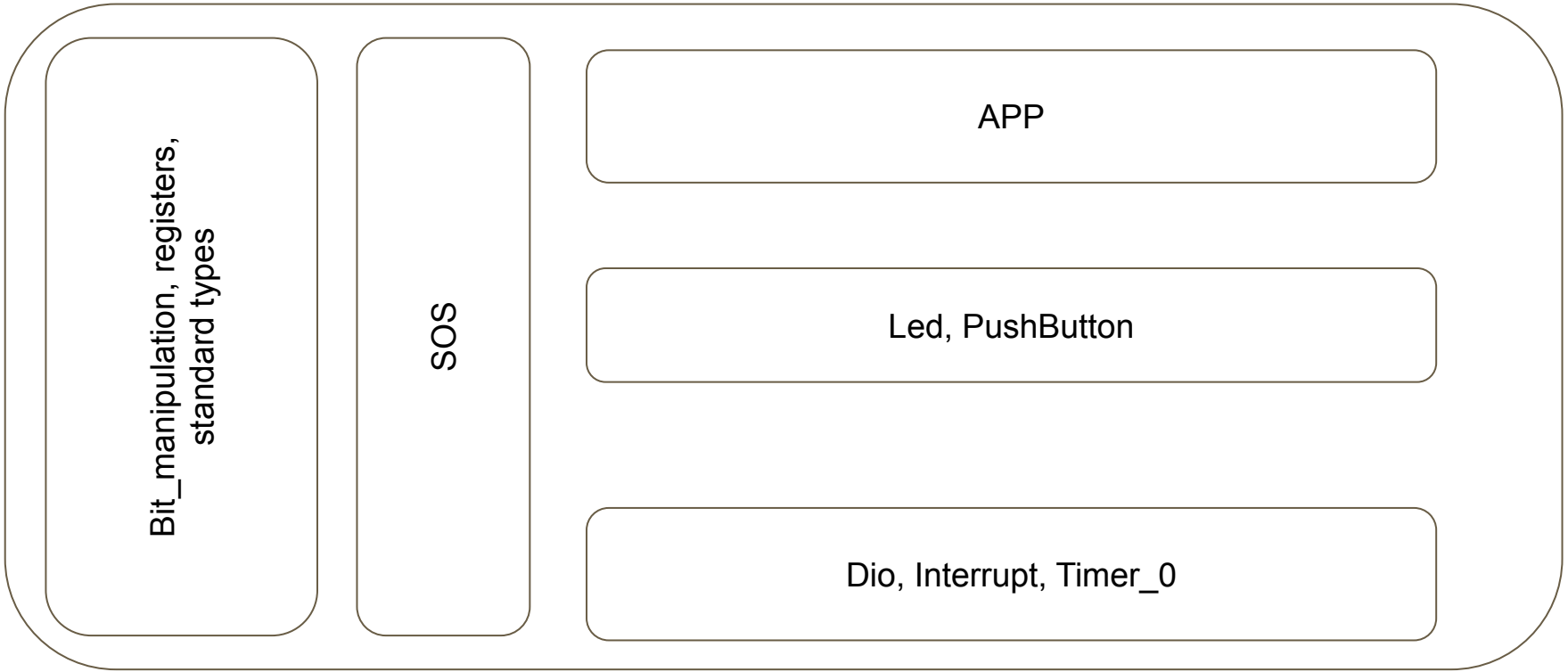
When pressing PBUTTON_0, the SOS will stop.

When pressing PBUTTON_1, the SOS will run.

Layered architecture



Modules/drivers



Modules Description

- DIO: It is Digital Input/Output module and it is responsible for dealing with the input and output digital pins.
- Timer_0: It is the timer module which deals with timers in the microcontroller and enables us to make delays.
- LED: It is the module that controls the LEDs through the Microcontroller.
- PushButton: It is responsible for knowing the state of the pushbutton.
- SOS: This is a service layer that manages the tasks occurrence.
- APP: It is the module containing the application itself.

Drivers Documentation

- **DIO**

1. PinDirection_t DIO_setpindir(uint8_t u8_a_portid , uint8_t u8_a_pinid , uint8_t u8_a_pindir);

   Description: sets the direction of the GPIO to either Input or Output.

   Arguments: takes the port and pin numbers and pin direction either Input or output.

   Return: Return Error statement from VALID_DIRECTION or NOT_VALID_DIRECTION.

Continue Drivers Documentation

2. PinValue_t DIO_setpinvalue(uint8_t u8_a_portid , uint8_t u8_a_pinid , uint8_t u8_a_pinval);

   Description: This function sets a specified pin to high or low.

   Arguments: takes the port and pin numbers and the value of the Pin.

   Return: returns the error statement from VALID_VALUE or NOT_VALID_VALUE.

Continue Drivers Documentation

3. PinRead_t DIO_readpin(uint8_t u8_a_portid , uint8_t u8_a_pinid , uint8_t* u8_a_val);

   Description: This function reads the state of the pin in case of input either high or low.

   Arguments: takes the port and pin numbers and a variable to put the value inside.

   Return: returns the error statement from VALID_READ or NOT_VALID_READ.

Continue Drivers Documentation

- **Timer_0**

1. TMR0_init_error TMR0_init(void);

   Description: sets the direction of the GPIO to either Input or Output.

   Arguments: nothing to take.

   Return: Return Error statement from VALID_INIT or NOT_VALID_INIT.

Continue Drivers Documentation

2. TMR0_start_error TMR0_start(void);

   Description: starts the timer to count.

   Arguments: nothing to take.

   Return: Return Error statement from VALID_START or NOT_VALID_START.

Continue Drivers Documentation

3. TMR0_stop_error TMR0_stop(void);

   Description: stops the timer to count.

   Arguments: nothing to take.

   Return: Return Error statement from VALID_STOP or NOT_VALID_STOP.

Continue Drivers Documentation

4. TMR0_delay_error TMR0_delayms(uint32_t u32_a_delayms);

   Description: delays the system for a period of time.

   Arguments: takes the time to be delayed for in milliseconds.

   Return: Return Error statement from VALID_DELAY or NOT_VALID_DELAY.

Continue Drivers Documentation

- **LED**

1. uint8_t LED_init(uint8_t Port_number, uint8_t Pin_number);

    Description: initializes LED port and pin.

    Arguments: takes the port and pin numbers.

    Return: Return Error statement.

Continue Drivers Documentation

2. uint8_t LED_ON(uint8_t Port_number, uint8_t Pin_number);

Description: turn the Led pin high.

Arguments: takes the port number and the pin number.

Return: Return Error statement.

Continue Drivers Documentation

3. uint8_t LED_OFF(uint8_t Port_number, uint8_t Pin_number);

   Description: turn the Led pin off.

   Arguments: takes the port number and the pin number.

   Return: Return Error statement.

Continue Drivers Documentation

4. uint8_t LED_toggle(uint8_t Port_number, uint8_t Pin_number);

   Description: toggle the Led pin.

   Arguments: takes the port number and the pin number.

   Return: Return Error statement.

Continue Drivers Documentation

- **SOS**

1. enu_system_status_t sos_init(void);

   Description: initializes The SOS by initializing the LED, PB, Timer and External interrupt.

   Arguments: takes nothing.

   Return: Return Error statement.

Continue Drivers Documentation

2. enu_creat_status_t sos_create_task(uint8_t ar_task_ID, uint8_t ar_task_priorty, uint8_t ar_task_priodicity);

Description: creates a new task and adds it to the SOS database.

Arguments: takes the task ID, priority and periodicity.

Return: Return Error statement.

Continue Drivers Documentation

3. enu_delete_status_t sos_delete_task(uint8_t ar_task_ID);

   Description: Deletes certain task from the SOS database.

   Arguments: takes the task ID.

   Return: Return Error statement.

Continue Drivers Documentation

4. enu_modify_status_t sos_modify_task(uint8_t ar_task_ID, uint8_t ar_task_priorty, uint8_t ar_task_priodicity);

Description: Modifies existing task parameters in the SOS database.

Arguments: takes the task ID, priority and periodicity.

Return: Return Error statement.

Continue Drivers Documentation

5. enu_run_status_t sos_run(void);

   Description: run the scheduler.

   Arguments: takes nothing.

   Return: Return Error statement.

sequence diagram