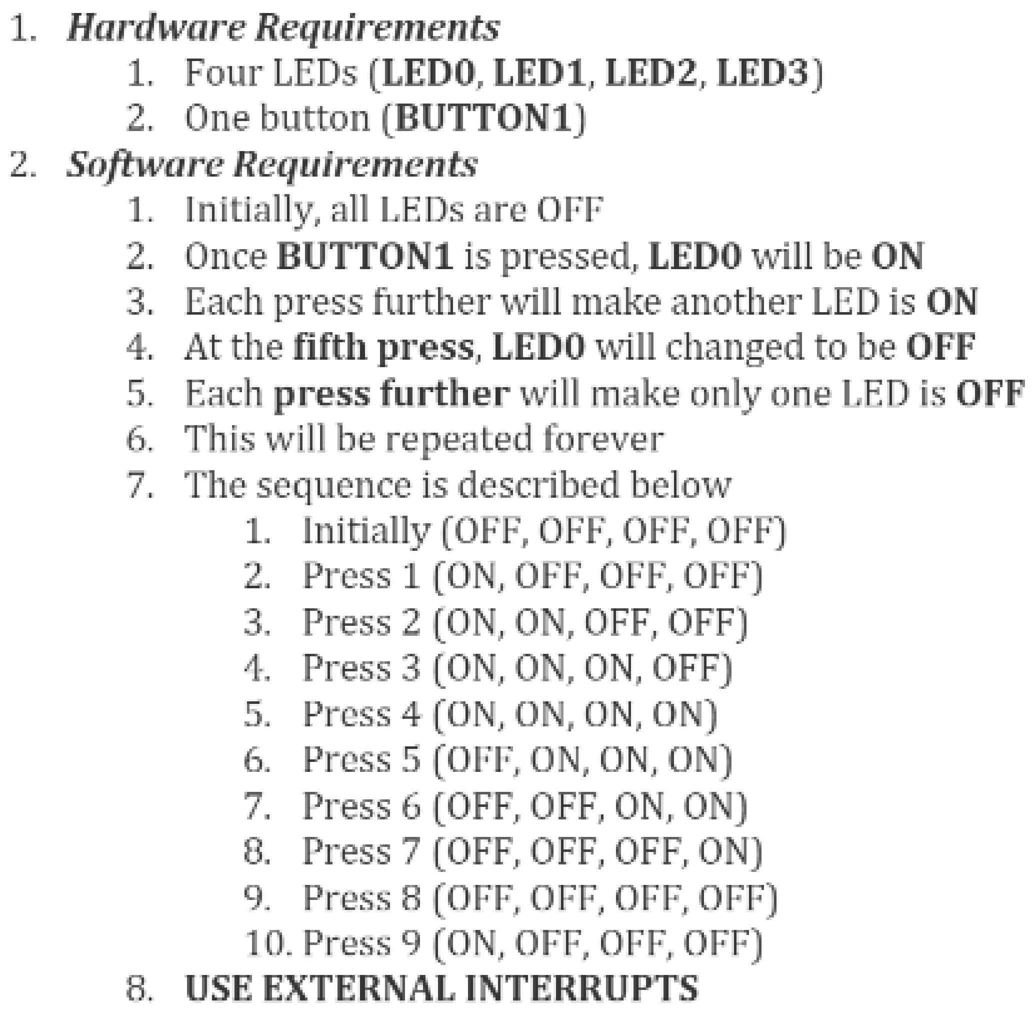
TASK: LED SEQUENCE V2.0

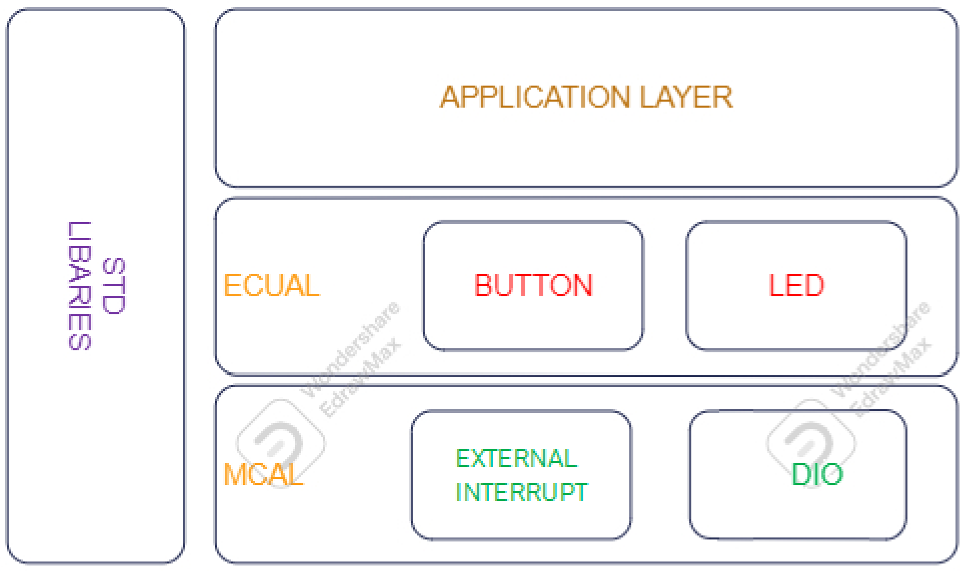
AUTHOR: MOHAMMED ABDEL-WAHAB

DESCRIPTION:

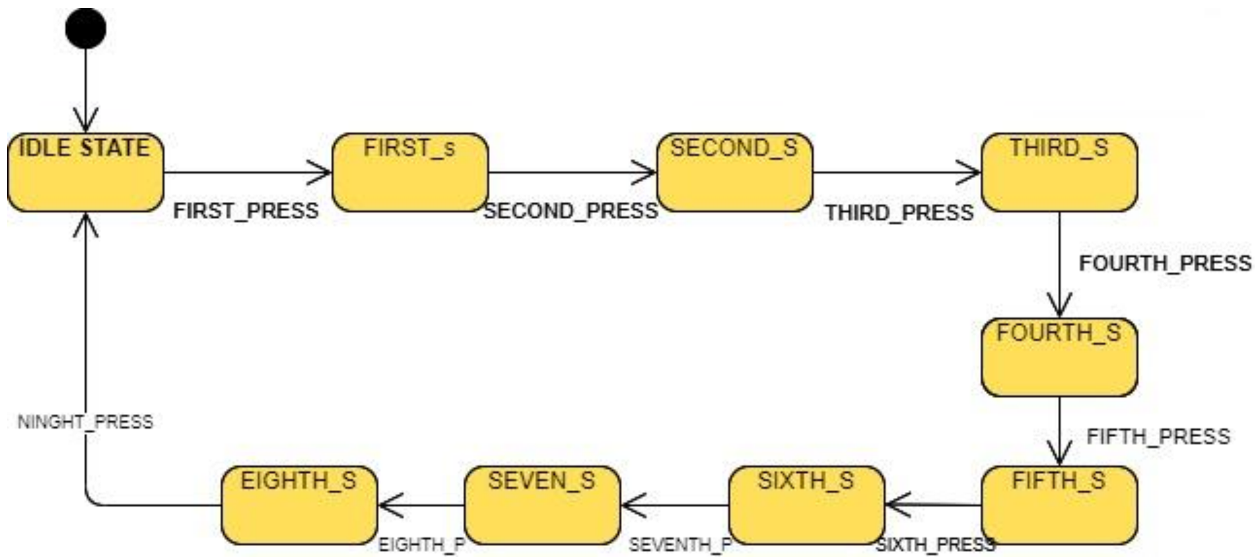


1. ***Hardware Requirements***
    1. Four LEDs (**LED0**, **LED1**, **LED2**, **LED3**)
    2. One button (**BUTTON1**)
2. ***Software Requirements***
    1. Initially, all LEDs are OFF
    2. Once **BUTTON1** is pressed, **LED0** will be **ON**
    3. Each press further will make another LED is **ON**
    4. At the **fifth press**, **LED0** will changed to be **OFF**
    5. Each **press further** will make only one LED is **OFF**
    6. This will be repeated forever
    7. The sequence is described below
        1. Initially (OFF, OFF, OFF, OFF)
        2. Press 1 (ON, OFF, OFF, OFF)
        3. Press 2 (ON, ON, OFF, OFF)
        4. Press 3 (ON, ON, ON, OFF)
        5. Press 4 (ON, ON, ON, ON)
        6. Press 5 (OFF, ON, ON, ON)
        7. Press 6 (OFF, OFF, ON, ON)
        8. Press 7 (OFF, OFF, OFF, ON)
        9. Press 8 (OFF, OFF, OFF, OFF)
        10. Press 9 (ON, OFF, OFF, OFF)
    8. **USE EXTERNAL INTERRUPTS**

Layered Architecture:



State machine diagram for the main flow of the Application:



- IDLE STATE: ALL LEDS ARE OFF
- FIRST STATE: LED 0 IS ONLY ON
- SECOND STATE: LED 0 & LED 1 ARE ON

- THIRD STATE : LED 0 & LED 1 & LED 2 ARE ON
- FOURTH STATE: ALL LEDS ARE ON
- FIFTH STATE: LED 0 IS ONLY OFF
- SIXTH STATE: LED 0 & LED 1 ARE OFF
- SEVENTH STATE: LED 0 & LED 1 & LED 2 ARE OFF
- EIGHTH STATE: ALL LEDS ARE OFF

A project APIs:

DIO DRIVER:

```
/**
* @brief Initialize the direction of specific pin @ref direction_t
* @param _pin_config A Reference of the pin configuration @pin_config_t
* @return status of the function
* E_OK :the function done successfully
* E_NOT_OK :the function has issues performing the function
*/
Std_ReturnType DIO_pin_direction_intialize(const pin_config_t *pin_config_ptr,direction_t
a_direction);
/**
* @brief Write the logic of specific pin @ref logic_t
* @param _pin_config A Reference of the pin configuration @pin_config_t
* @param logic
* @return status of the function
* E_OK :the function done successfully
* E_NOT_OK :the function has issues performing the function
*/
Std_ReturnType DIO_pin_write_logic(const pin_config_t *pin_config_ptr,const logic_t
a_logic);
/**
* @brief Read the logic of specific pin @ref logic_t
* @param _pin_config A Reference of the pin configuration @pin_config_t
* @param logic
* @return status of the function
* E_OK :the function done successfully
* E_NOT_OK :the function has issues performing the function
*/
Std_ReturnType DIO_pin_read_logic(const pin_config_t *pin_config_ptr, logic_t
*logic_ptr);

/**
* @brief Toggle the logic of specific pin @ref logic_t
* @param _pin_config A Reference of the pin configuration @pin_config_t
* @return status of the function
* E_OK :the function done successfully
* E_NOT_OK :the function has issues performing the function
*/
Std_ReturnType DIO_pin_toggle_logic(const pin_config_t *pin_config_ptr);
/**
* @brief Initialize the direction of specific pin and Initialize its logic
* @param _pin_config A Reference of the pin configuration @pin_config_t
* @return status of the function
* E_OK :the function done successfully
* E_NOT_OK :the function has issues performing the function
*/
/*
Std_ReturnType DIO_pin_intialize(const pin_config_t *pin_config_ptr);
*/
/**
*
* @param port_index
* @param direction
* @return status of the function
* E_OK :the function done successfully
* E_NOT_OK :the function has issues performing the function
*/
Std_ReturnType DIO_port_direction_intialize(const port_index_t a_port_index, uint8_t
a_direction);
/**
* @param port_index
* @param logic
* @return status of the function
* E_OK :the function done successfully
* E_NOT_OK :the function has issues performing the function
*/
Std_ReturnType DIO_port_write_logic(const port_index_t a_port_index , uint8_t a_logic);
/**
* @param port_index
* @param logic
* @return status of the function
* E_OK :the function done successfully
* E_NOT_OK :the function has issues performing the function
*/
Std_ReturnType DIO_port_read_logic(const port_index_t a_port_index , uint8_t *const
a_logic_ptr);
/**
* @param port_index
* @return status of the function
* E_OK :the function done successfully
* E_NOT_OK :the function has issues performing the function
*/
Std_ReturnType DIO_port_toggle_logic(const port_index_t a_port_index);
```

LED DRIVER:

```
/**
 * @breif  Initialize The led by configuring the pin as output and write low
 * @param Led  The reference of the led module configuration
 * @return status of the function
 *          E_OK :the function done successfully
 *          E_NOT_OK :the function has issues performing the function
 */
Std_ReturnType LED_initialize(const led_t *led_ptr);

/**
 * @breif Turn the led on
 * @param led  The reference of the led module configuration
 * @return status of the function
 *          E_OK :the function done successfully
 *          E_NOT_OK :the function has issues performing the function
 */

Std_ReturnType LED_turn_on(const led_t *led_ptr);

/**
 * @breif Turn the led off
 * @param led  The reference of the led module configuration
 * @return status of the function
 *          E_OK :the function done successfully
 *          E_NOT_OK :the function has issues performing the function
 */

Std_ReturnType LED_turn_off (const led_t *led_ptr);

/**
 * @breif  Toggle the led
 * @param led  The reference of the led module configuration
 * @return status of the function
 *          E_OK :the function done successfully
 *          E_NOT_OK :the function has issues performing the function
 */
Std_ReturnType LED_turn_toggle (const led_t *led_ptr);
```

BUTTON DRIVER:

```
/**
 * @breif Initialize The assigned pin to be input
 * @param btn he reference of the button module configuration
 * @return status of the function
 *          E_OK :the function done successfully
 *          E_NOT_OK :the function has issues performing the function
 */
Std_ReturnType BTN_init(const button_t *btn_ptr);

/**
 * @breif Read the push button if is it pressed or released
 * @param btn    The reference of the button module configuration
 * @param btn_state  The reference of the variable that store the button status @ref
button_status_t
 * @return status of the function
 *          E_OK :the function done successfully
 *          E_NOT_OK :the function has issues performing the function
 */
Std_ReturnType BTN_read_state(const button_t *btn_ptr, button_status_t *btn_states_ptr);
```

EXTERNAL INTERRUPT DRIVER:

```
/*
* Description : Call the Call Back function in the application after the edge is detected
* @param A pointer to function & the external interrupt id
* @return status of the function
* E_OK :the function done successfully
* E_NOT_OK :the function has issues performing the function
*/
Std_ReturnType EXT_INTx_setCallBack(volatile void(*a_fptr)(void), const Interrupt_ID_t
a_interrupt_number );
/*
* Description : initialize the the dio pin to be an external interrupt
* @param A Reference of the external interrupt configuration
* @return status of the function
* E_OK :the function done successfully
* E_NOT_OK :the function has issues performing the function
*/
Std_ReturnType EXT_INTx_Init(const Interrupt_Config_t *Interrupt_Config_Ptr );
/*
* Description : set the edge in which the external interrupt will be triggered
* @param edge type & the external interrupt id
* @return status of the function
* E_OK :the function done successfully
* E_NOT_OK :the function has issues performing the function
*/
Std_ReturnType EXT_INTx_setEdgeType(Interrupt_Edge_type_t a_edgeType ,  Interrupt_ID_t
a_interrupt_Id);
/**
* @brief DeInitialize the interrupt module
* @param the external interrupt id
* @return Status of the a_interrupt_Id
* (E_OK) : The function done successfully
* (E_NOT_OK) : The function has issue to perform this action
*/
Std_ReturnType EXT_INTx_DeInit(const Interrupt_ID_t a_interrupt_Id);
```