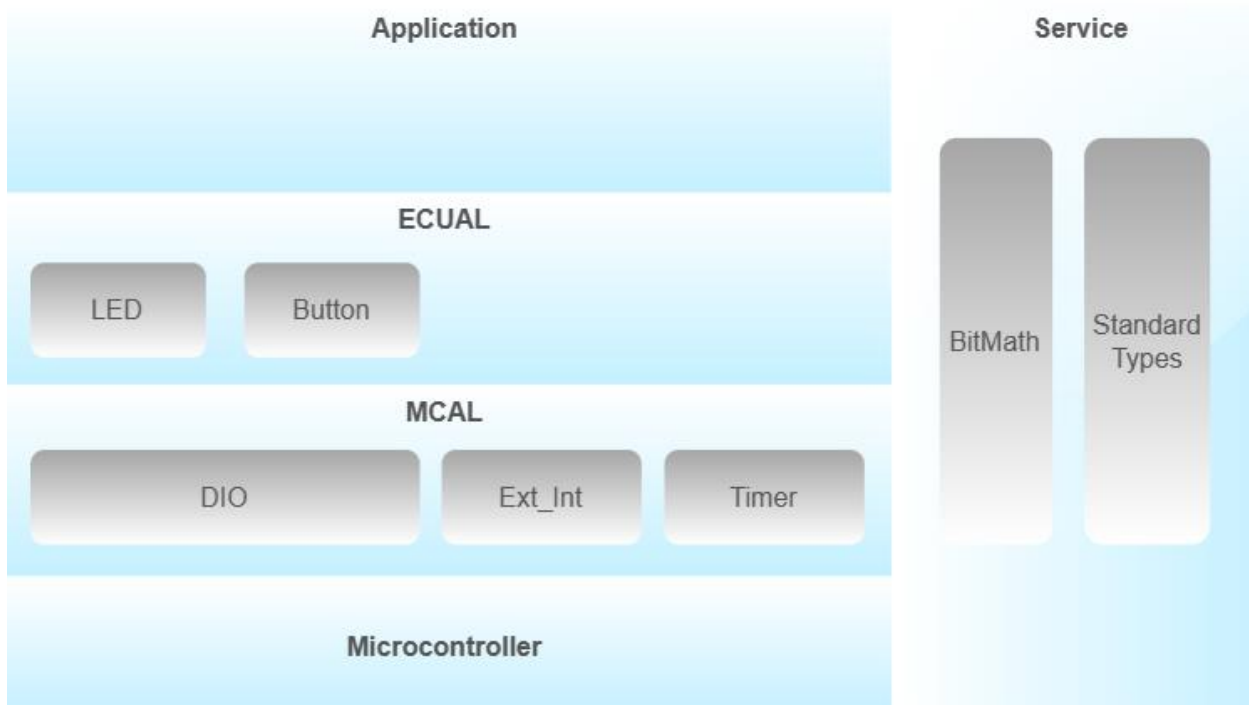
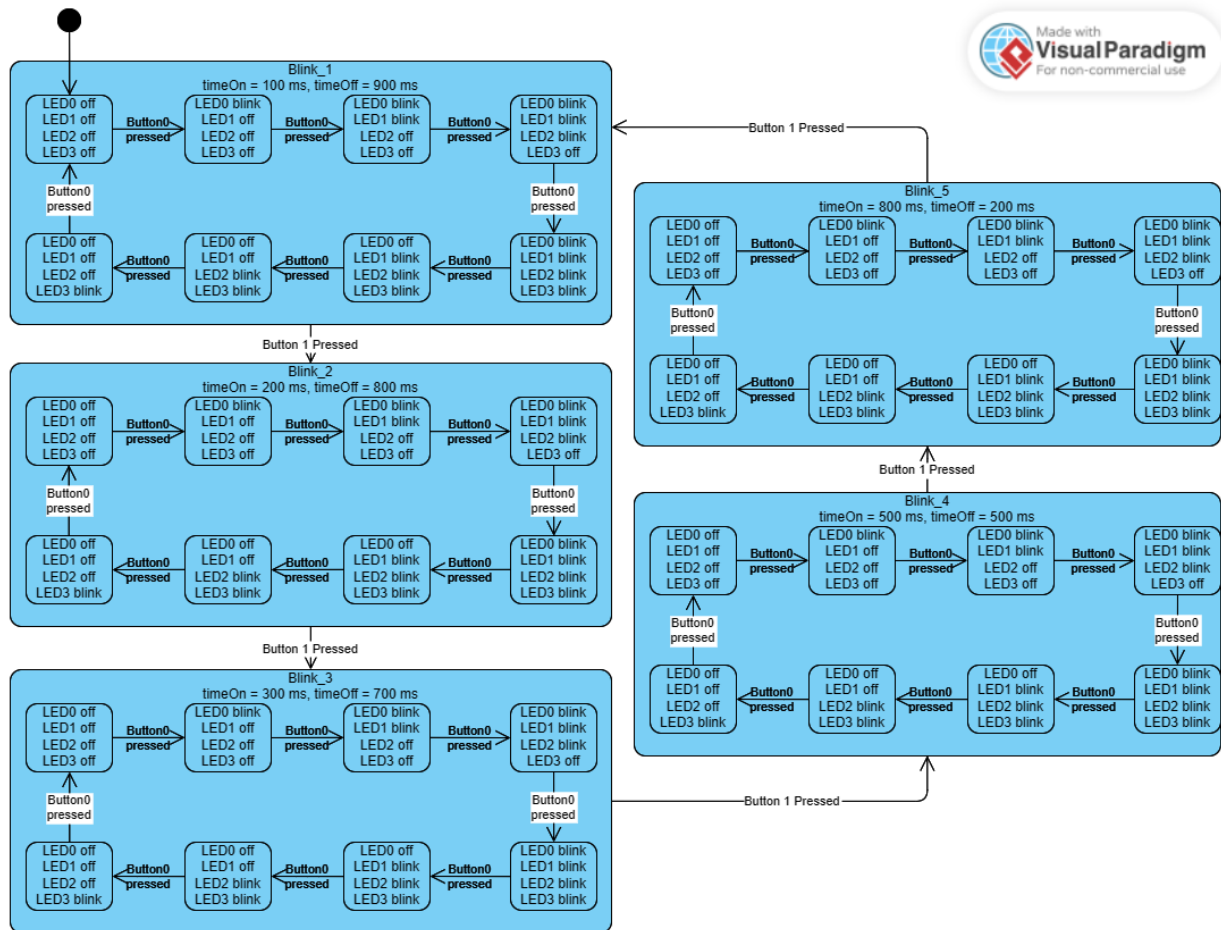


LED SEQUENCE v 3.0

LAYERED ARCHITECTURE

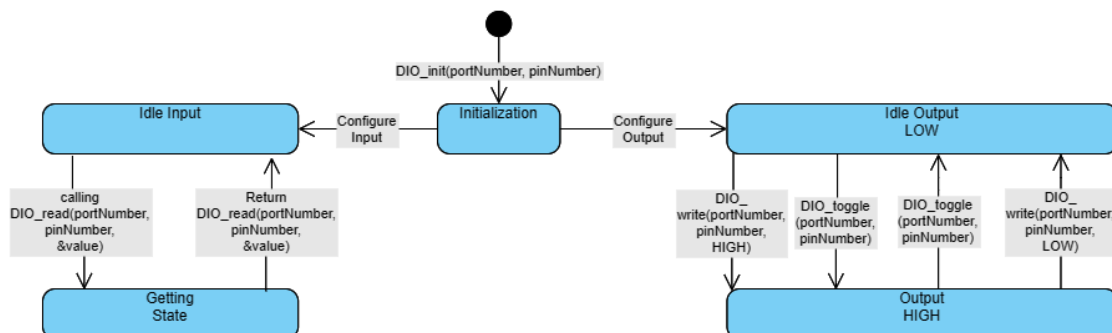


STATE MACHINE DIAGRAM

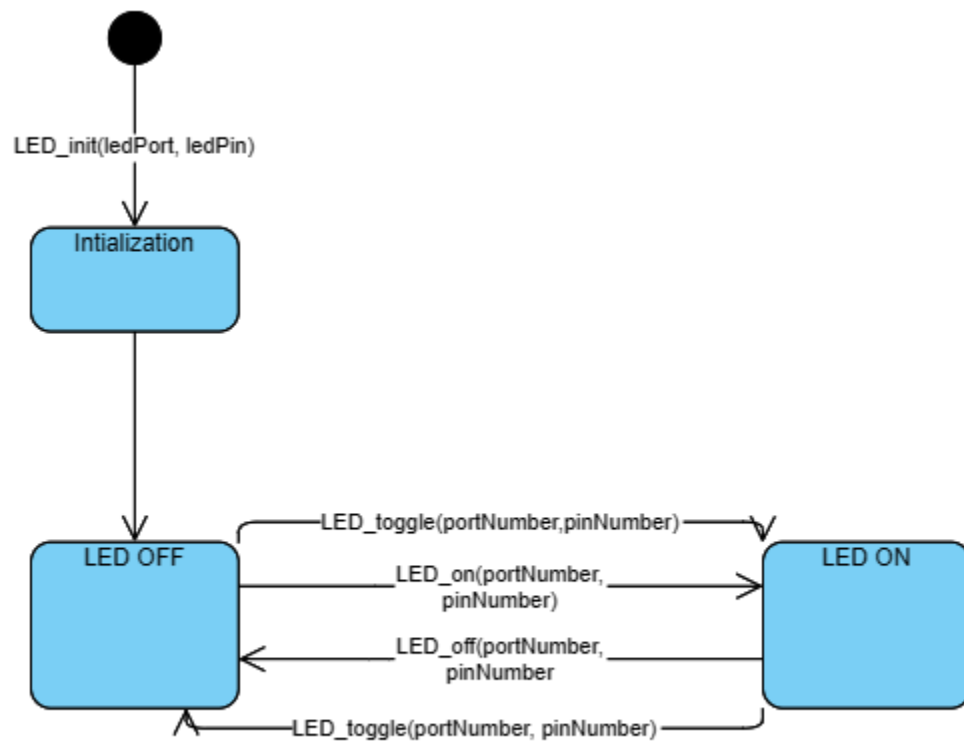


DRIVERS APIs STATE DIAGRAMS

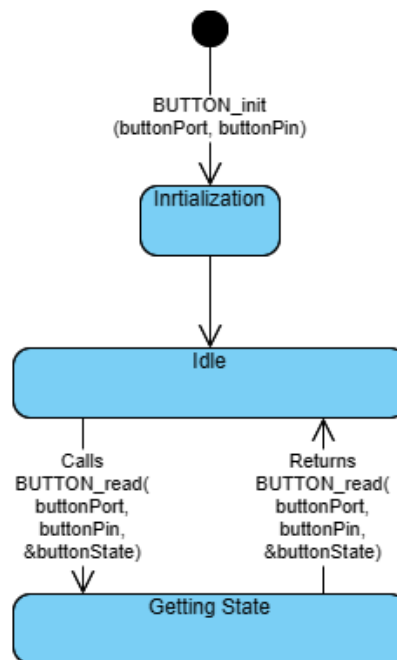
DIO



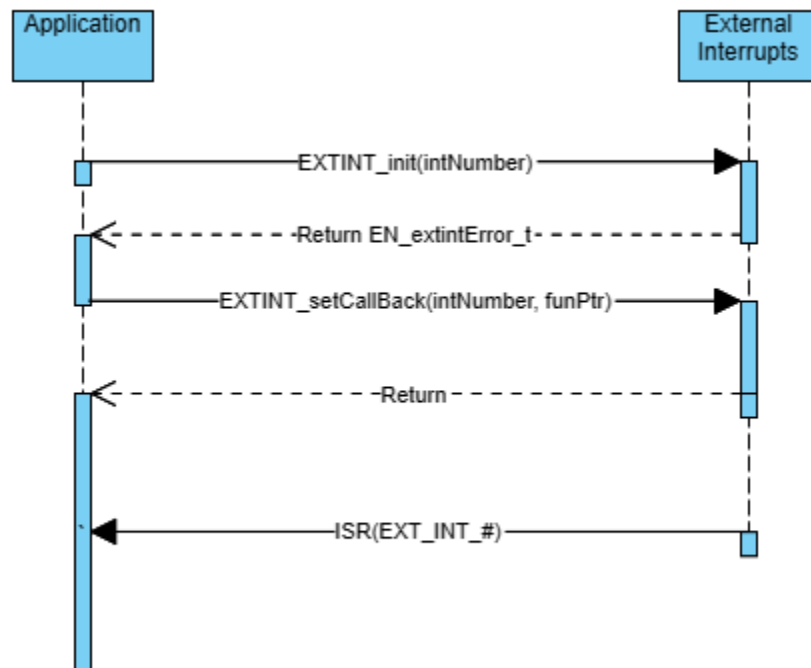
LED



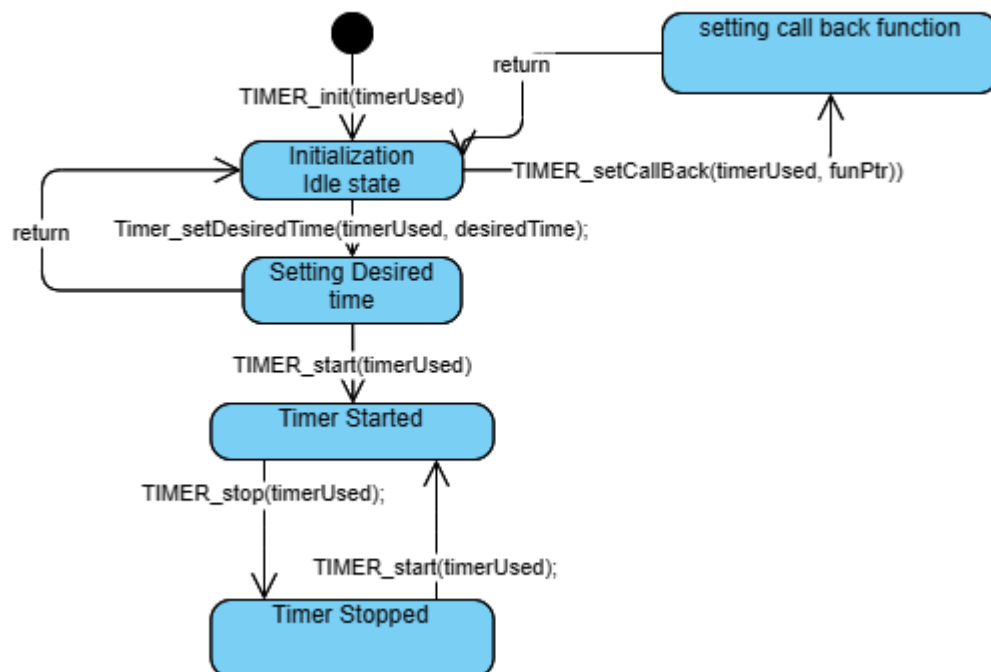
BUTTON



EXTERNAL INTERRUPTS



Timers



API

1. DIO

EN_dioError_t **DIO_init**(uint8_t portNumber, uint8_t pinNumber, uint8_t direction);

EN_dioError_t **DIO_write**(uint8_t portNumber, uint8_t pinNumber, uint8_t value);

EN_dioError_t **DIO_toggle**(uint8_t portNumber, uint8_t pinNumber);

EN_dioError_t **DIO_read**(uint8_t portNumber, uint8_t pinNumber, uint8_t *value);

2. EXTERNAL INTERRUPTS

EN_extintError_t **EXTINT_Init** (uint8_t intNumber);

void **EXTINT_setCallBackInt** (uint8_t intNumber, void (*funPtr) (void));

3. LED

EN_ledError_t **LED_init**(uint8_t ledPort, uint8_t ledPin);

EN_ledError_t **LED_on**(uint8_t ledPort, uint8_t ledPin);

EN_ledError_t **LED_off**(uint8_t ledPort, uint8_t ledPin);

EN_ledError_t **LED_toggle**(uint8_t ledPort, uint8_t ledPin);

4. BUTTON

EN_buttonError_t **BUTTON_init**(uint8_t buttonPort, uint8_t buttonPin);

EN_buttonError_t **BUTTON_read**(uint8_t buttonPort, uint8_t buttonPin, uint8_t *buttonState);

5. APP

void **APP_initModules**(void);

void **APP_ledSequenceV_1** (void);

void **button0Task** (uint8_t* counter, uint8_t state);

void **button1Task** (void);

void **sysTickTask**(void);

6. TIMERS

EN_timerError_t **TIMER_init**(uint8_t timerUsed);

EN_timerError_t **TIMER_setTime**(uint8_t timerUsed, uint32_t desiredTime);

EN_timerError_t **TIMER_start**(uint8_t timerUsed);

EN_timerError_t **TIMER_stop**(uint8_t timerUsed);

void **TIMER_setCallBack**(uint8_t timerUsed, void (*funPtr)(void));

