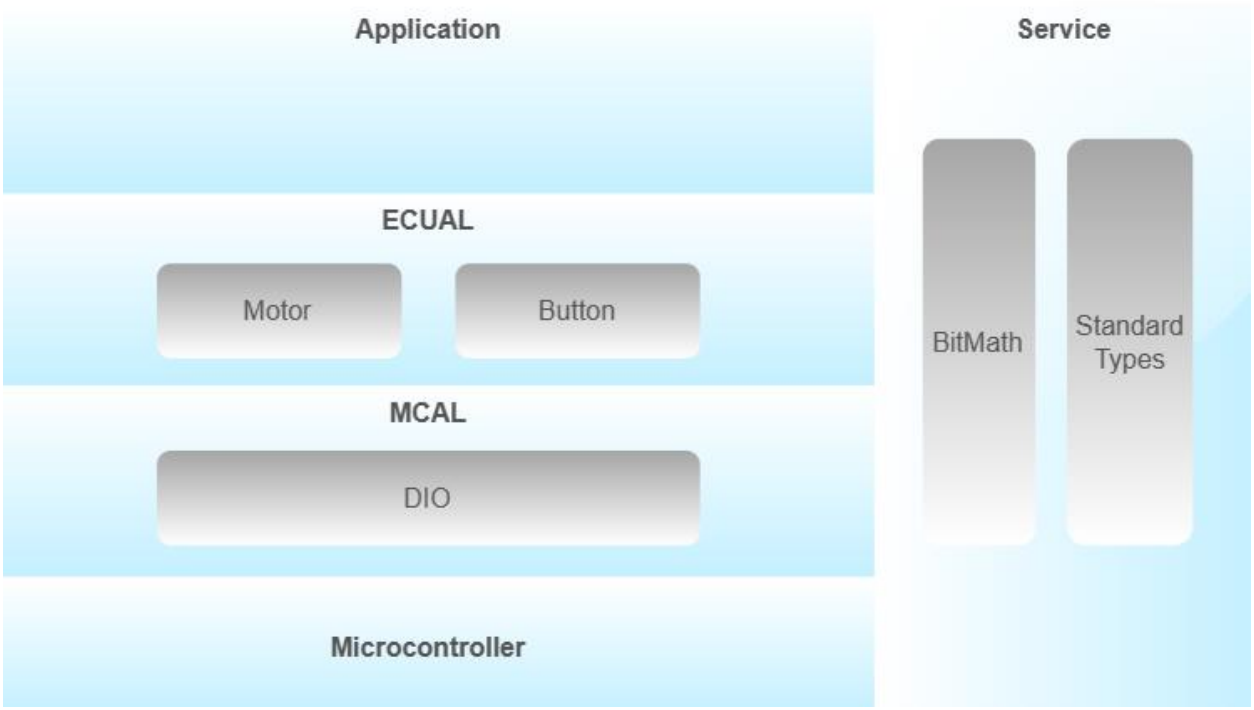
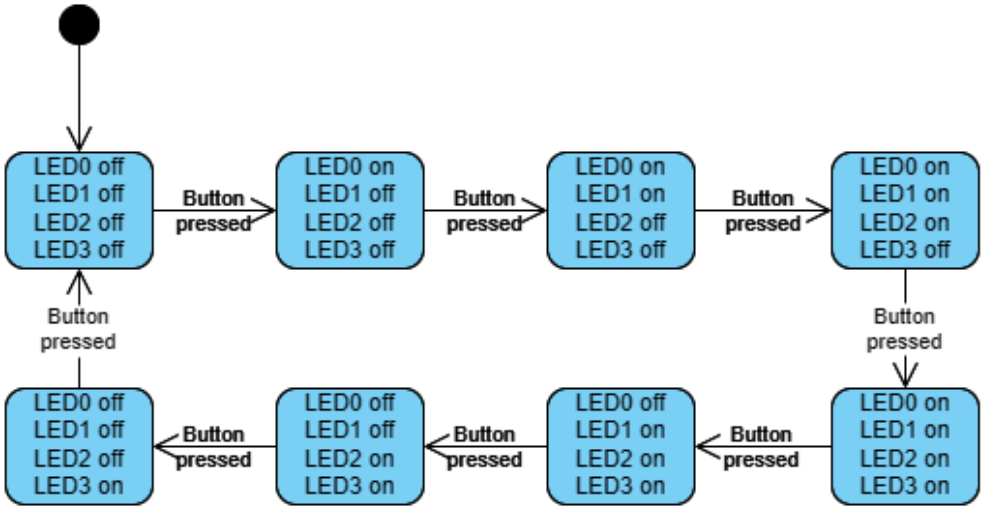
LED SEQUENCE v 1.0

LAYERED ARCHITECTURE

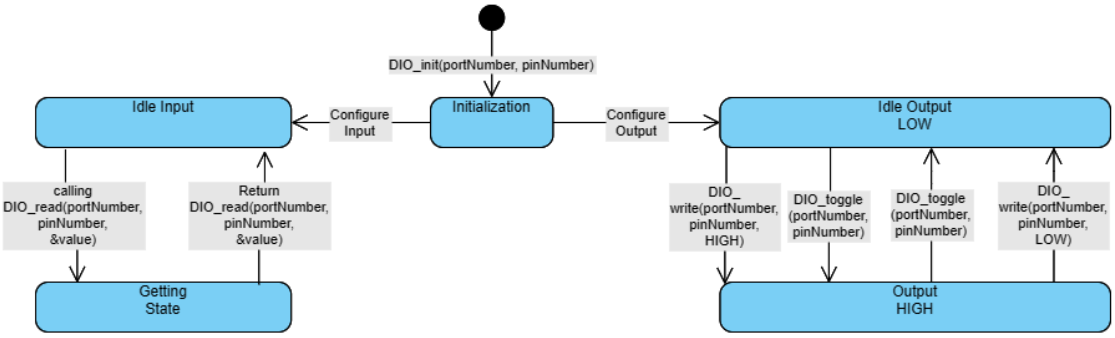


STATE MACHINE DIAGRAM

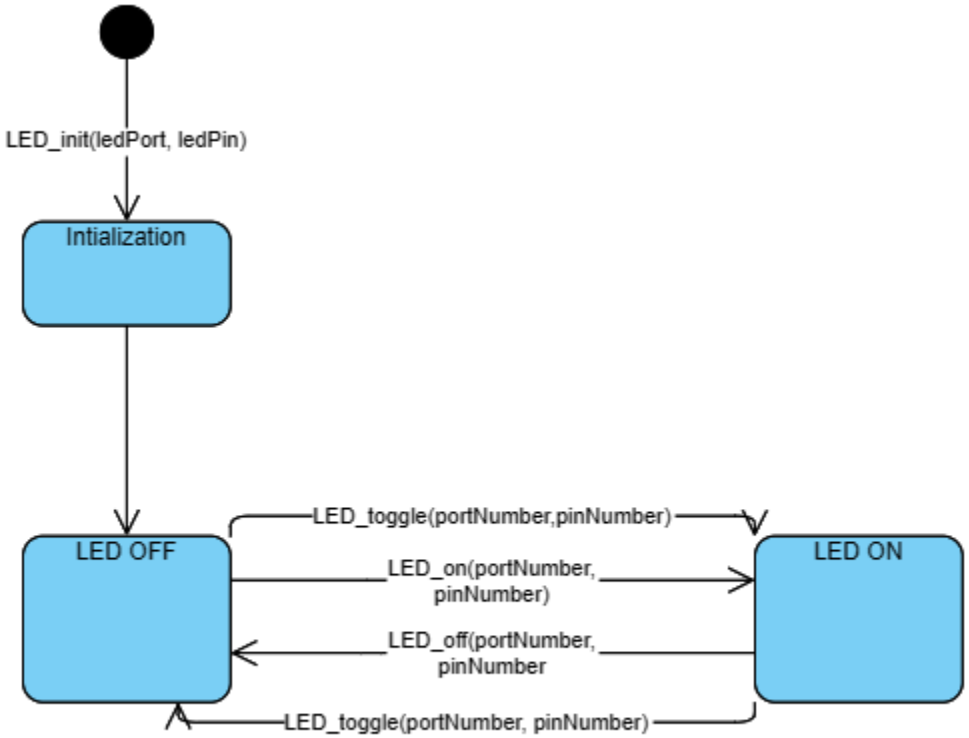


DRIVERS APIS STATE DIAGRAMS

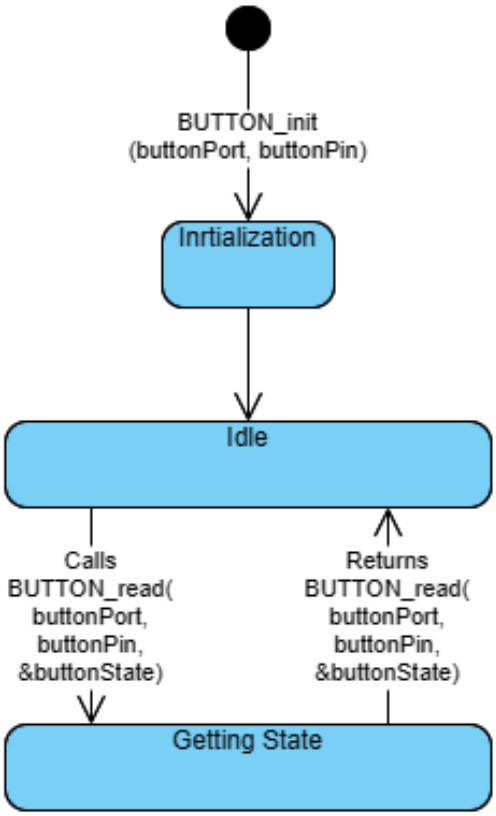
DIO



LED



BUTTON


BUTTON_init
(buttonPort, buttonPin)
Inrtialization
Idle
Calls
BUTTON_read(
buttonPort,
buttonPin,
&buttonState)
Returns
BUTTON_read(
buttonPort,
buttonPin,
&buttonState)
Getting State

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. 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);
```

3. 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);
```

4. APP

```
void APP_initModules(void);

void APP_ledSequenceV_1 (void);
```