



1. MCAL APIs

1.1 DIO APIs

```
ERROR_state_t DIO_init(const STR_DIO_config_t * configurations);  
ERROR_state_t DIO_write(uint8_t port, ENU_pins pin, uint8_t data);  
ERROR_state_t DIO_read(uint8_t port, ENU_pins pin, uint8_t * data);  
ERROR_state_t DIO_toggle(uint8_t port, ENU_pins pin);
```

```
struct STR_DIO_config_t {  
    uint8_t port_no;  
    ENU_pin pin_no;  
    uint8_t direction;  
    uint8_t resistor;  
    uint8_t initial_value;  
};
```

1.2 PWM APIs

```
ERROR_state_t PWM_init(const STR_PWM_config_t * configurations);  
ERROR_state_t PWM_start(  
    uint8_t pwm_no, ENU_frequency_t frequency,  
    ENU_duty_t dutyCycle  
);  
ERROR_state_t PWM_stop(uint8_t pwm_no);
```

```
struct STR_PWM_config_t {  
    uint8_t pwm_no;  
    ENU_frequency_t frequency;  
    ENU_duty_t duty_cycle;  
};
```

1.3 TIMER APIs

```
ERROR_state_t TIMER_init(const STR_TIMER_config_t * configurations);  
ERROR_state_t TIMER_start(uint8_t timer_no, uint16_t ticks);  
ERROR_state_t TIMER_read(uint8_t timer_no, uint8_t * value);  
ERROR_state_t TIMER_set(uint8_t timer_no, uint8_t value);  
ERROR_state_t TIMER_checkStatus(uint8_t timer_no, uint8_t * status);
```

```
struct STR_TIMER_config_t {  
    uint8_t timer_no;  
    uint8_t clock_source;  
    uint8_t mode;  
    uint8_t interrupt_mode;  
};
```

2. ECUAL APIs

2.1 Control Buttons APIs

```
ERROR_state_t BUTTON_init(void);  
ERROR_state_t BUTTON_checkState(uint8_t * status);  
ERROR_state_t BUTTON_update(void);
```

2.2 Motor APIs

```
ERROR_state_t MOTOR_init(void);  
ERROR_state_t MOTOR_setState(void);  
ERROR_state_t MOTOR_getState(uint8_t * state);  
ERROR_state_t MOTOR_update(void);
```

3. APP APIs

3.1 Car Control Unit APIs

```
ERROR_state_t CAR_init(void);  
ERROR_state_t CAR_update(void);
```