

# 1. MCAL APIs

## 1.1 DIO APIs

```
void DIO_init(ST_DIO_config_t * configurations);  
void DIO_write(uint8_t port, EN_pins pin, uint8_t data);  
void DIO_read(uint8_t port, EN_pins pin, uint8_t * data);  
void DIO_toggle(uint8_t port, EN_pins pin);
```

```
struct ST_DIO_config_t {  
    uint8_t port_no;  
    uint8_t pin_no;  
    uint8_t direction;  
    uint8_t resistor;  
    uint8_t initial_value;  
};
```

## 1.2 PWM APIs

```
void PWM_init(ST_PWM_config_t * configurations);  
void PWM_start(EN_frequency_t frequency, EN_duty_t dutyCycle);  
void PWM_stop(void);
```

```
struct ST_PWM_config_t {  
    uint8_t port_no;  
    uint8_t pin_no;  
    uint8_t clock_source;  
    uint8_t duty_cycle;  
};
```

## 1.3 TIMER APIs

```
void TIMER_init(ST_TIMER_config_t * configurations);  
void TIMER_start(uint64_t ticks);  
void TIMER_read(uint8_t * value);  
void TIMER_set(uint8_t value);  
void TIMER_checkStatus(uint8_t * status);
```

```
struct ST_TIMER_config_t {  
    uint8_t timer_no;  
    uint8_t clk_frequency;  
    uint8_t mode;  
};
```

## 2. ECUAL APIs

### 2.1 Control Buttons APIs

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

### 2.2 Motor APIs

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

## 3. APP APIs

### 3.1 Car Control Unit APIs

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