



1. MCAL

1.1. DIO APIs

```
/* This Function determine the Pin direction Whether input or output */
E_status DIO_SetPinDirection(ST_DIO_config_t * Configurations);
typedef ST_DIO_config_t{
    uint8_t port_no;
    uint8_t pin_no;
    uint8_t state;           // input or output
}ST_DIO_config_t;
E_status DIO_SetPinValue(uint8_t au8_port_no, uint8_t au8_pin_no, uint8_t au8_value);
E_status DIO_GetPinValue(uint8_t au8_port_no, uint8_t au8_pin_no, uint8_t * data);
E_status DIO_TogglePin(uint8_t au8_port_no, uint8_t au8_pin_no);
```

1.2. TIMER APIs

```
E_status TIMER_init(ST_TIMER_config_t* configurations);
E_status TIMER_Start(uint64_t au64_ticks);
E_status TIMER_Read(uint8_t *au8_value);
E_status TIMER_Set(uint8_t au8_value);
E_status TIMER_Checkstatus(uint8_t * au8_status);
```

```
typedef ST_TIMER_config_t{
    EN_timerno_t timer_no;
    EN_timerclk_t timer_clk;
    EN_timermode_t mode;
}ST_TIMER_config_t;
```

1.3. PWM APIs

```
E_status PWM_init(ST_PWM_config_t* configurations);  
E_status TIMER_Start(EN_frequency_t frequency, EN_duty_t dutycycle);  
E_status TIMER_Steering(EN_sterring_t steering);  
E_status TIMER_Stop(void);
```

2. HAL

2.1. Buttons APIs

```
E_status Button_init(EN_BTN_config_t configurations);  
E_status Button_GetState(uint8_t* au8_status);
```

2.2. MOTOR APIs

```
E_status MOTOR_init(void);  
E_status MOTOR_SetDirection(EN_DIR_t Direction);  
E_status MOTOR_SetSpeed(EN_Speed_t DutyCycle);  
E_status MOTOR_Stop(void);
```

3. APP APIs

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