

APP

MOTOR

Push\_Button

Timer

PWM

DIO

MICROCONTROLLER

# 1) MCAL

## a) DIO

```
(i)   DIO_ERROR_state_t DIO_setPortDirection(uint8_t PORTID,
        uint8_t PortDirection);
(ii)  DIO_ERROR_state_t DIO_setPinDirection(uint8_t PortID ,
        uint8_t PinNumber, uint8_t PinDirection);
(iii) DIO_ERROR_state_t DIO_writePinValue(uint8_t PortID,
        uint8_t PinNumber, uint8_t PinDirection);
(iv)  DIO_ERROR_state_t DIO_readPort(uint8_t PortID,
        uint8_t* PortState);
(v)   DIO_ERROR_state_t DIO_readPin(uint8_t PortID, uint8_t
        PinNumber, uint8_t* PinState);
(vi)  DIO_ERROR_state_t DIO_SetPullupRes(uint8_t PortID,
        uint8_t PinNumber);
```

## b) TIMER

```
(i)   TIMER_ERROR_state_t TIMER_init(const
        STR_TIMER_config_t * configurations);
(ii)  TIMER_ERROR_state_t TIMER_start(uint8_t timer_no,
        uint16_t ticks);
(iii) TIMER_ERROR_state_t TIMER_read(uint8_t timer_no,
        uint8_t * value);
(iv)  TIMER_ERROR_state_t TIMER_set(uint8_t timer_no,
        uint8_t value);
(v)   TIMER_ERROR_state_t
        TIMER_checkStatus(uint8_t timer_no, uint8_t * status);
```

## c) PWM

```
(i)   PWM_ERROR_state_t PWM_INIT(STR_PWM_config_t
        * configurations);
(ii)  PWM_ERROR_state_t PWM_START(uint8_t pwm_ch, uint8_t
        pwm_freq, uint8_t pwm_dutyCycle);
(iii) PWM_ERROR_state_t PWM_STOP(uint8_t pwm_ch);
```

## 2) HAL

### a) **BUTTON**

```
(i)   BUTTON_ERROR_state BUTTON_INIT(uint8_t pinNo);  
(ii)  BUTTON_ERROR_state BUTTON_STATE(uint8_t pinNo, uint8_t  
      * buttonState);
```

### b) **MOTOR**

```
(i)   MOTOR_ERROR_state MOTOR_INIT(uint8_t pinNo);  
(ii)  MOTOR_ERROR_state MOTOR_STATE(uint8_t * motorState);
```

## 3) APP

### a) **CAR**

```
(i)   CAR_ERROR_state CAR_INIT(void);  
(ii)  CAR_ERROR_state CAR_DIR(uint8_t carDir);
```