# MCAL APIs

## 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;

    uint8\_t pin\_no;

    uint8\_t direction;

    uint8\_t resistor;

    uint8\_t initial\_value;

};

## PWM APIs

ERROR\_state\_t PWM\_init(const STR\_PWM\_config\_t \* configurations);

ERROR\_state\_t PWM\_start(

uint8\_t port,  ENU\_pins pin,

ENU\_frequency\_t frequency, ENU\_duty\_t dutyCycle

);

ERROR\_state\_t PWM\_stop((uint8\_t port,  ENU\_pins pin);

struct STR\_PWM\_config\_t{

    uint8\_t port\_no;

    uint8\_t pin\_no;

    uint8\_t clock\_source;

    uint8\_t duty\_cycle;

};

## TIMER APIs

ERROR\_state\_t TIMER\_init(const STR\_TIMER\_config\_t \*  configurations);

ERROR\_state\_t TIMER\_start(uint64\_t ticks);

ERROR\_state\_t TIMER\_read(uint8\_t \* value);

ERROR\_state\_t TIMER\_set(uint8\_t value);

ERROR\_state\_t TIMER\_checkStatus(uint8\_t \* status);

struct STR\_TIMER\_config\_t{

    uint8\_t timer\_no;

    uint8\_t clk\_frequency;

    uint8\_t mode;

};

# ECUAL APIs

## Control Buttons APIs

ERROR\_state\_t BUTTON\_init(void);

ERROR\_state\_t BUTTON\_checkState(uint8\_t \* status);

ERROR\_state\_t BUTTON\_update(void);

## 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);

# APP APIs

## Car Control Unit APIs

ERROR\_state\_t CAR\_init(void);

ERROR\_state\_t CAR\_update(void);