# MCAL APIs

## 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;

};

## 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;

};

## 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;

};

# ECUAL APIs

## Control Buttons APIs

void BUTTON\_init(void);

void BUTTON\_checkState(uint8\_t \* status);

void BUTTON\_update(void);

## Motor APIs

void MOTOR\_init(void);

void MOTOR\_setState(void);

void MOTOR\_getState(uint8\_t \* state);

void MOTOR\_update(void);

# APP APIs

## Car Control Unit APIs

void CAR\_init(void);

void CAR\_update(void);