Static Design for car Control System

**MCAL**

**ECUAL**

**Microcontroller**

**APP**

**TIMER**

**PWM**

**DIO**

**MOTOR**

**Buttons**

**APP**

MCAL APIs

* DIO APIs

**E\_status DIO\_init(ST\_DIO\_config\_t \* Configurations);**

**E\_status DIO\_Write(uint8\_t port, EN\_pins pin, uint8\_t data);**

**E\_status DIO\_Read(uint8\_t port, EN\_pins pin, uint8\_t \*data);**

**E\_status DIO\_Toggle(uint8\_t port, EN \_pins pin);**

* TIMER APIs

**E\_status TIMER\_init(ST\_TIMER\_config\_t\* configurations);**

**E\_status TIMER\_Start(uint64\_t ticks);**

**E\_status TIMER\_Read(uint8\_t \*value);**

**E\_status TIMER\_Set(uint8\_t value);**

**E\_status TIMER\_Checkstatus(uint8\_t \*status);**

* 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);**

ECUAL APIs

* Buttons APIs

**E\_status Button\_init(EN\_BTN\_config\_t configurations);**

**E\_status Button\_GetState(uint8\_t\* status);**

* 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);**

APP APIs

**E\_status CAR\_init(void);**

**E\_status CAR\_update(void);**