**Car Control Unit**

**MCAL** **APIs**

* **DIO APIs**

E\_STATUS DIO\_Init(ST\_DIO\_config\* configurations);

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

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

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

* **PWM APIs**

E\_STATUS PWM\_Init(ST\_PWM\_config\_t\* configurations);

E\_STATUS PWM\_Start(EN\_frequency\_t frequency, EN\_duty\_t dutyCycle);

E\_STATUS PWM\_Steering(EN\_sterring\_t steering);

E\_STATUS PWM\_Stop(void);

* **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);

**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);