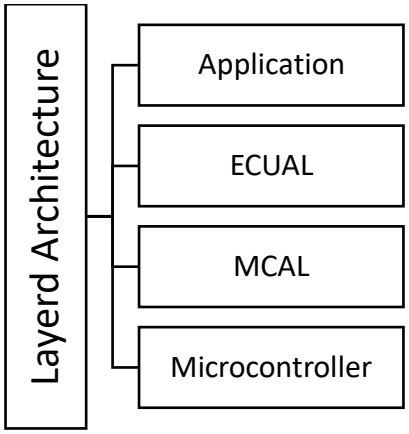
Moving Car System Design

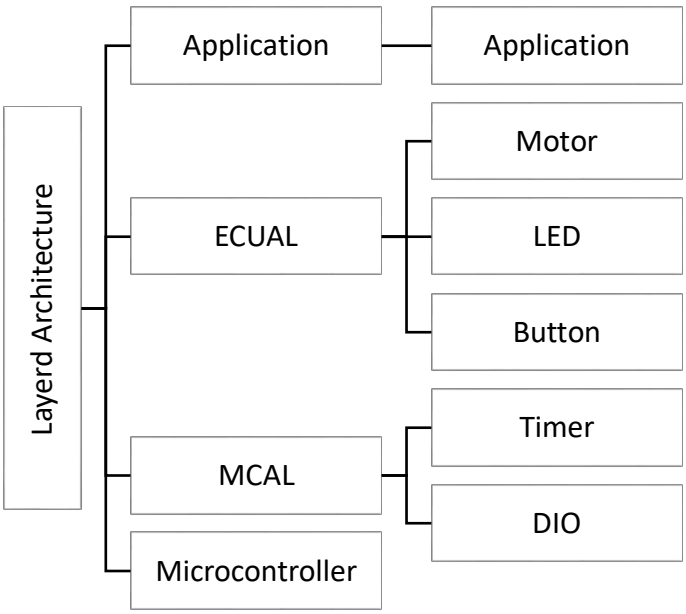
System layered architecture:

- Application
- ECUAL: Electronic Unit Application Layer, For Drivers
- MCAL: Microcontroller Application Layer
- Microcontroller



Specify system modules/drivers

- Motor (4 Motors) - ECUAL
- Button (2 Buttons) - ECUAL
- LED (4 LEDs) - ECUAL
- Timer - MCAL
- DIO - MCAL
- PWM - MCAL



API’s

DIO

```
EN_DIO_ERROR_t DIO_init(ST_DIO_config_t * configurations);
EN_DIO_ERROR_t DIO_write(ST_DIO_config_t *configurations , uint8_t data);
EN_DIO_ERROR_t DIO_read(ST_DIO_config_t *configurations , uint8_t *data);
EN_DIO_ERROR_t DIO_toggle(ST_DIO_config_t *configurations);
```

Timer

```
EN_timer_error_t timer_init(timer_configuration_t *timer_configuration);
EN_timer_error_t timer_init_callback_OVF(void (*callback)(void));
EN_timer_error_t timer_init_callback_COMP(void (*callback)(void));
```

Motors

```
EN_DC_motor_error_t motor_initialize(const dc_motor_config_t *dc_motor);
EN_DC_motor_error_t motor_turn_on(dc_motor_config_t\
*dc_motor,EN_DC_motor_direction_t dc_motor_direction );
EN_DC_motor_error_t motor_stop(dc_motor_config_t *dc_motor);
EN_DC_motor_error_t motor_set_speed(dc_motor_config_t *dc_motor, uint8_t
dc_motor_speed);
```

LED

```
EN_LED_status_t led_init(ST_LED_config_t *led);
EN_LED_status_t led_turn_on(ST_LED_config_t *led);
EN_LED_status_t led_turn_off(ST_LED_config_t *led);
EN_LED_status_t led_toggle(ST_LED_config_t *led);
```

Button

```
EN_button_error_t button_init(const EN_button_t* btn);
EN_button_error_t button_read_state( EN_button_t *btn, EN_button_state_t
*btn_state);
```