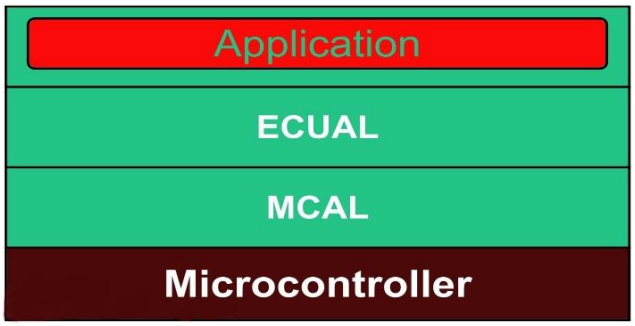
Sprints

Task Moving Car Design

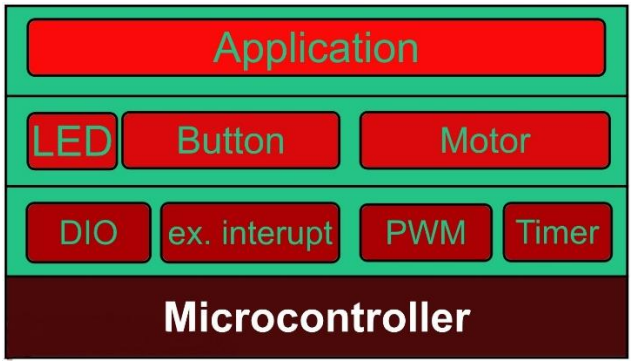
Firstly: Layered architecture:

1- Microcontroller
2- MCAL
3- ECUAL
4-
5- Application



Secondly: System modules:

1- **Specify system modules/drivers:**
   - DIO, PWM, TIMER, BUTTON, MOTOR
2- **Assign each module to its related layer:**
   - By drawing



Thirdly: APIs:

1- 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);
```

2- External interrupt APIs:

```
void INT_VECT(void) __attribute__ ((signal,used));
void SIE(void);
void CLI(void);
void INT_SENSE(void);
void EX_INT_zero(void);
```

Sprints

3- PWM APIs:

```
void PWM_init (ST_PWM_config_t* configurations);
void PWM_start (EN_frequency_t frequency, EN_duty_t duty);
void PWM_stop (void);
```

4- TIMER APIs:

```
void TIMER_init (uint8_t Mode,uint8_t intial_value);
void TIMER_start (uint8_t prescaler_value);
void TIMER_set(uint8_t intial_value);
void TIMER_getStatus(uint8_t *value);
void TIMER_Stop (void);
Void Delay(uint32_t milliseconds);
```

5- LED APIs:

```
void LED_init (uint8_t port, uint8_t pin);
void LED_on (uint8_t port, uint8_t pin);
void LED_off (uint8_t port, uint8_t pin);
void LED_toggle (uint8_t port, uint8_t pin);
```

Sprints

6- BUTTON APIs:

```
void BUTTON_init (uint8_t buttonport, uint8_t buttonpin);
void BUTTON_read (uint8_t buttonport, uint8_t buttonpin, uint8_t *value);
```

7- MOTOR APIs:

```
void MOTOR_init (ST_MOTOR_config_t* configurations);
void MOTORS _forward (uint8_t delay1, uint8_t delay2);
void MOTORS _right (uint8_t angle);
void MOTORS _stop (uint8_t delay);
void M _clockWise (ST_MOTOR_config_t* configurations, uint8_t value);
void M _CclockWise (ST_MOTOR_config_t* configurations, uint8_t value);
```

8- APPLICATION APIs:

```
void App_init (void);
void App_start (void);
```