

Design Document

Project Name:

Moving car Design

By:

Mohamed Sayed

Project Description:-

- **Car Components:**

- 1) Four motors (M1, M2, M3, M4)
- 2) One button to start (PB1)
- 3) One button for stop (PB2)
- 4) Four LEDs (LED1, LED2, LED3, LED4)

- **System Requirements:**

- 1) The car starts initially from 0 speed
- 2) When PB1 is pressed, the car will move forward after 1 second
- 3) The car will move forward to create the longest side of the rectangle for 3 seconds with 50% of its maximum speed
- 4) After finishing the first longest side the car will stop for 0.5 seconds, rotate 90 degrees to the right, and stop for 0.5 second
- 5) The car will move to create the short side of the rectangle at 30% of its speed for 2 seconds
- 6) After finishing the shortest side the car will stop for 0.5 seconds, rotate 90 degrees to the right, and stop for 0.5 second
- 7) Steps 3 to 6 will be repeated infinitely until you press the stop button (PB2)
- 8) PB2 acts as a sudden break, and it has the highest priority

Layered architecture

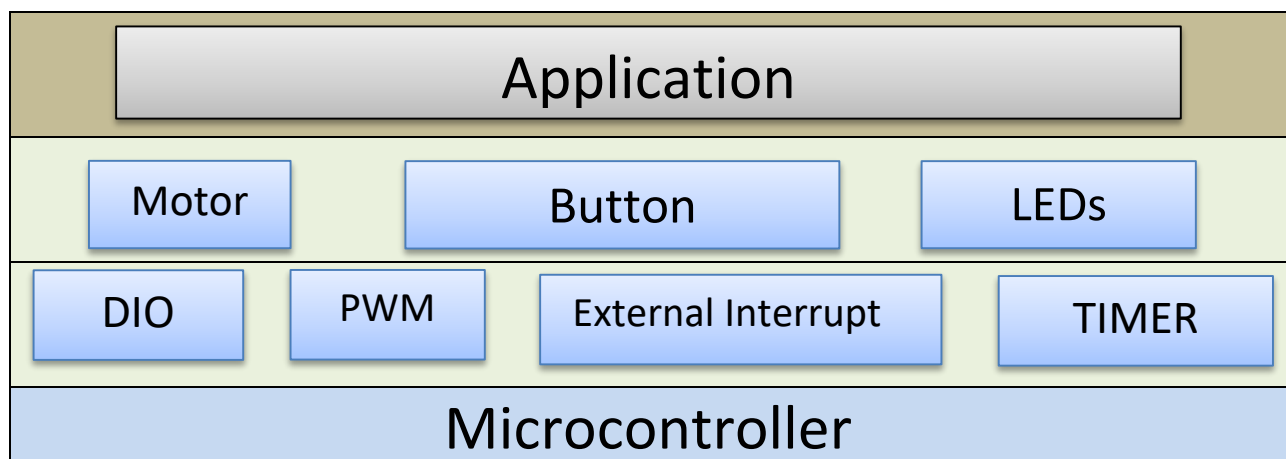
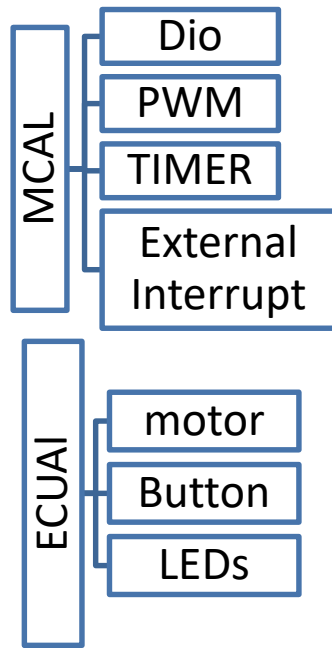
The system may be divided to 4 layers:-

- Microcontroller
- MCAL
- ECUAI
- Application

Application
ECUAL
MCAL
Microcontroller

System modules

The system may be divided into drivers:-



The APIs for each module that will provide specific functionalities for the upper layers

DIO APIs

```
void DIO_init(st_Dio_config_t* configurations);  
void DIO_read (uint8_t port, EN pins pin, uint8_t data);  
void DIO_toggle (uint8_t port, EN pins pin);
```

PWM APIs

```
void PWM_init(ST_PWM_config_t* configuration);  
void PWM_start(EN_frequency_t frequency, EN_duty_t dutyCycle);  
void PWM_stop(void);
```

TIMER APIs

```
void TIMER_init(ST_TIMER_config_t* configuration);  
void TIMER_start(unit64_t ticks);  
void TIMER_read(unit8_t* value);  
void TIMER_set (unit8_t value);  
void TIMER_checkStatus(unit8_t *status);
```

External interrupt APIs

```
void EXI_Enable (ExInterruptSource_type Interrupt);  
void EXI_Disable (ExInterruptSource_type Interrupt);  
void EXI_Trigger(ExInterruptSource_type Interrupt,TriggerEdge_type trigger);  
void EXI_SetCallBack(ExInterruptSource_type Interrupt,void(*pf)(void));
```

Button APIs

```
Button_status Button_check(u8 ButtonNo);
```

LEDs APIs

```
void LED_ON(u8 LEDno);  
void LED_Off(u8 LEDno);
```

Motor APIs

```
void Forward(void);  
void Stop (void);  
void Right (void);
```

Application APIs

```
void APP_Init(void);  
void APP_Start(void);
```