System Static Design

The system:

A car control.

Hardware to use:

- Two motors.
- Four push buttons:
  - Button_1: Forward direction.
  - Button_2: Turn Right.
  - Button_3: Turn Left.
  - Button_4: Change speed and direction.
- Microcontroller that has the following modules:
  - DIO.
  - Timer.
  - PWM.

System in action:

- Your system has three speeds.
- Pressing Button_4 will change the car speed:
  - **Speed b:** changes the car direction to backward, and speed to 30%.
  - **Speed 0:** changes the car speed to 0%.
  - **Speed 1:** changes the car direction to forward, and speed to 30%.
  - **Speed 2:** changes the car direction to forward, and speed to 60%.
  - **Speed 3:** changes the car direction to forward, and speed to 90%.
  - Each button press must move to the next speed.
- After setting your preferred speed press Button_1 will make the car moves in the decided direction and speed as long as you press the button.
- Pressing Button_2 the car will turn right, and speed to 30%. as long as you press the button.
- Pressing Button_3 the car will turn left, and speed to 30%. as long as you press the button.

Layered Architecture:

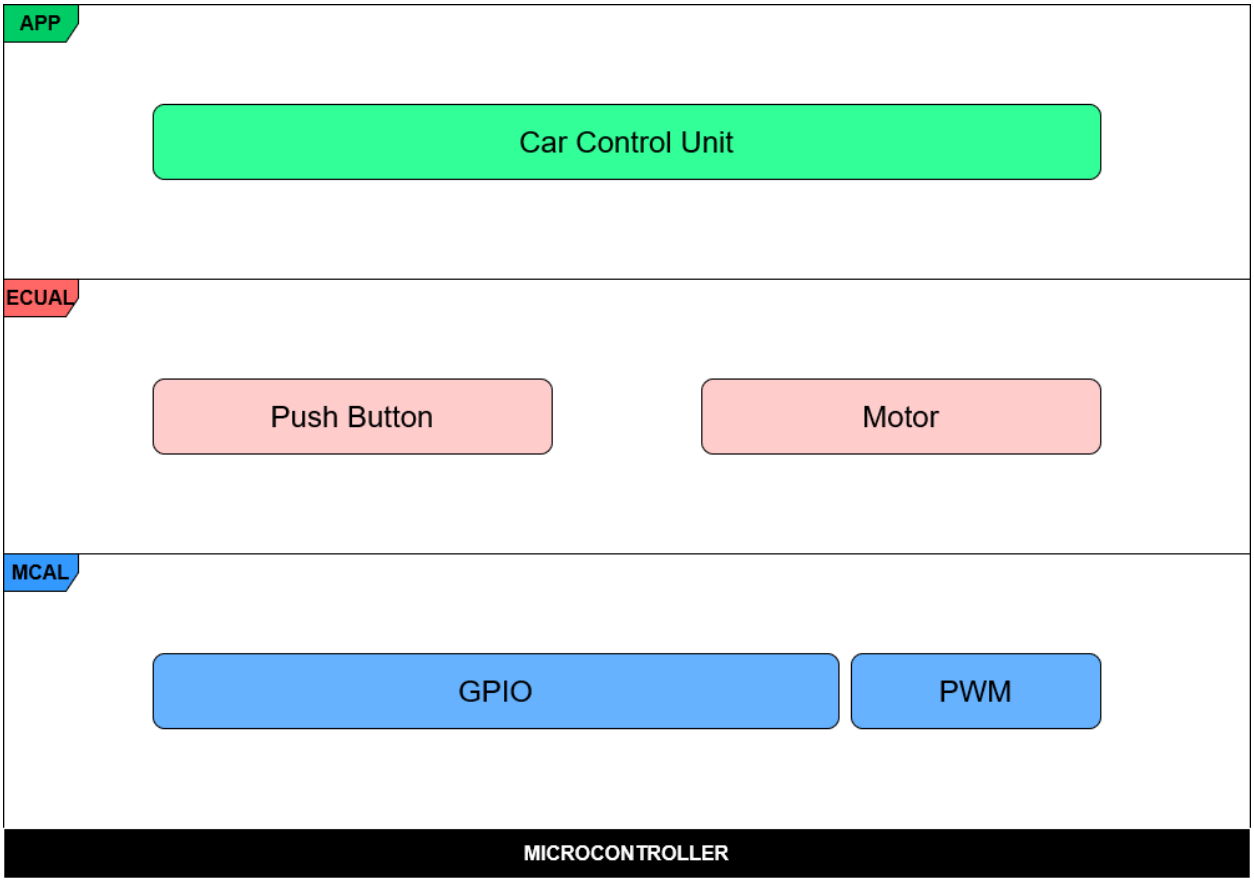

Figure 1 Car Layered Architecture Design

- MCAL Layer Modules:
  - GPIO Module.
  - PWM Module.
- HWAL Layer Modules:
  - Push Button Module.
  - Motor Module.
- Application Layer Modules:
  - Car Control Module.

Module APIs:

MCAL APIs

GPIO APIs

```
uint8_t GpioSetPinDirection(
      uint8_t PortName , uint8_t PinNo ,uint8_t PinDirection
);
uint8_t GpioWritePin(
      uint8_t PortName , uint8_t PinNo ,uint8_t PinValue
);
uint8_t GpioTogglePin(uint8_t PortName,uint8_t PinNo);
uint8_t GpioReadPin(
      uint8_t PortName,uint8_t PinNo,ptr_uint8_t PinData
);
uint8_t GpioEnablePinPullup(uint8_t PortName,uint8_t PinNo);
```

PWM APIs

```
uint8_t PwmInit(void);
uint8_t PwmStart(uint8_t PwmChannelNumber);
uint8_t PwmStop(uint8_t PwmChannelNumber);
uint8_t PwmConnect(uint8_t PwmChannelNumber);
uint8_t PwmDisconnect(uint8_t PwmChannelNumber);
uint8_t PwmSetDuty(uint8_t PwmChannelNumber,uint8_t PwmDuty);
```

ECUAL APIs

Push Button APIs

```
PSHBTTN_ERROR_RETVAL_t PSHBTTN_Init (
      DIO_PORT_ID_t port, DIO_PIN_ID_t pin,
      PSHBTTN_PULLUP_Status_t status
);
PSHBTTN_ERROR_RETVAL_t PSHBTTN_EnablePullUp (
      DIO_PORT_ID_t port, DIO_PIN_ID_t pin
```

```
);
uint8_t PSHBTTN_Status (DIO_PORT_ID_t port, DIO_PIN_ID_t pin);
```

Motor APIs

```
void MOTOR_init(void);
void MOTOR_stop(uint8_t motor_no);
void MOTOR_start(uint8_t motor_no,uint8_t speed,uint8_t dir);
void MOTOR_speed(uint8_t motor_no,uint8_t speed);
void MOTOR_dir(uint8_t motor_no,uint8_t dir);
```

APP APIs

Car Control Unit APIs

```
CAR_ERROR_state_t CAR_Init(void);
CAR_ERROR_state_t CAR_Update(void);
```

Manual:

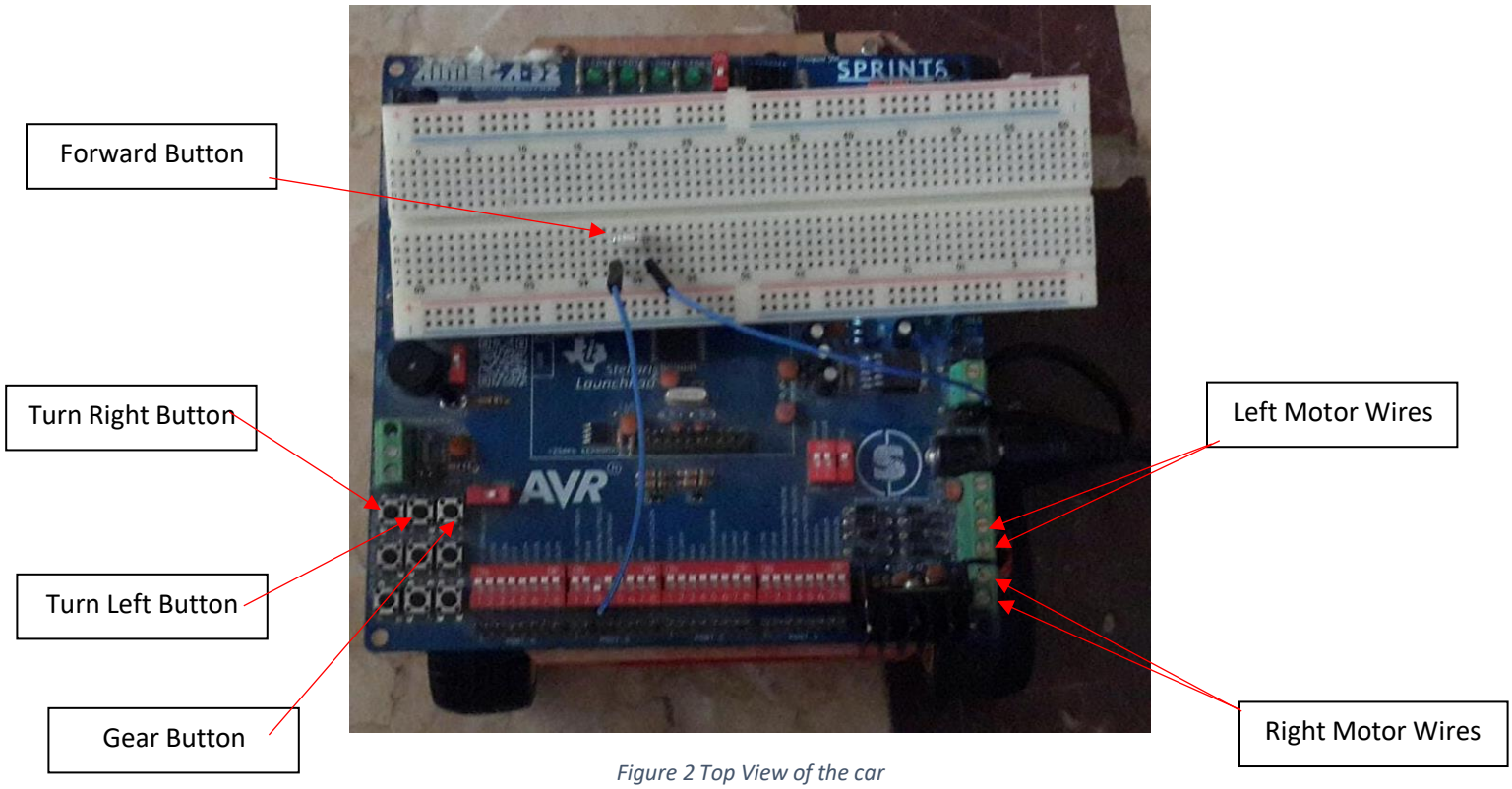

Figure 2 Top View of the car

Simulation:

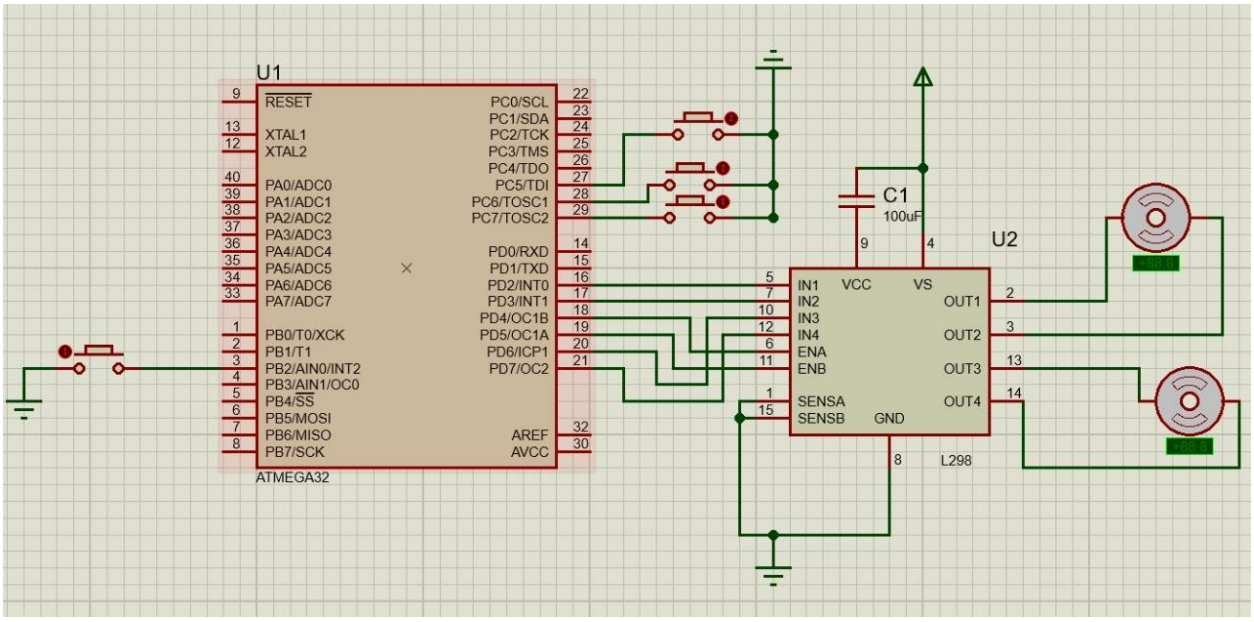

Figure 3 Proteus Simulation Circuit