

System Static Design

▪ The system:

A car control.

▪ Hardware to use:

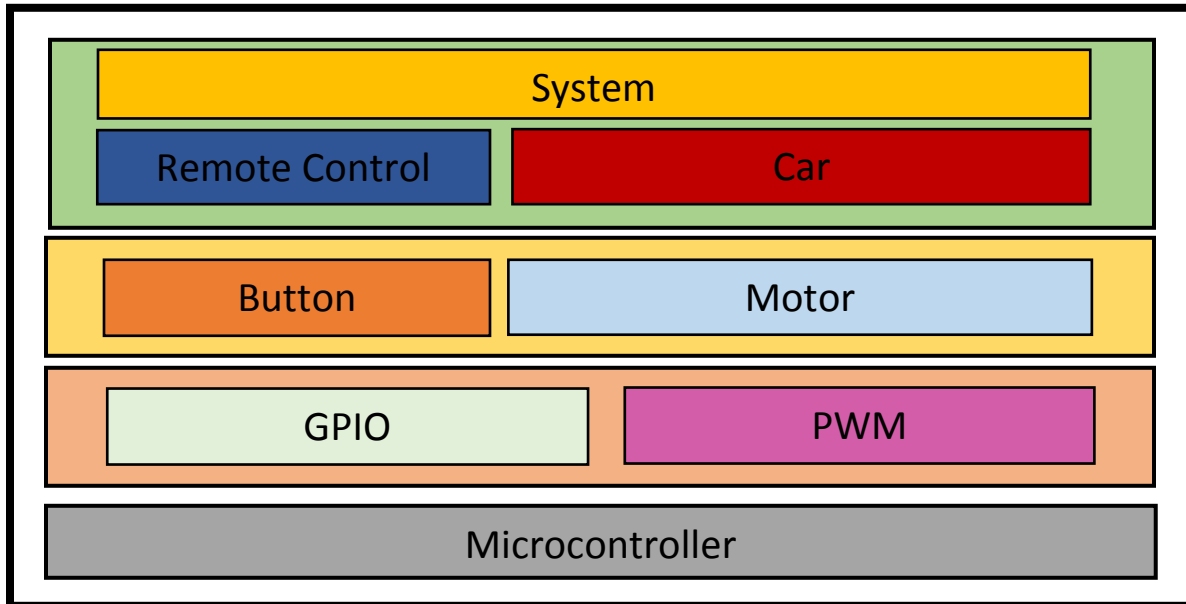
- Two motors.
- Four push buttons:
 - Button_1: Forward direction.
 - Button_2: Turn Right.
 - Button_3: Turn Left.
 - Button_4: Change speed and direction.
- Microcontroller that has the following modules:
 - DIO.
 - Timer.
 - PWM.

▪ 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 60%.
 - 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 as long as you press the button.
- Pressing Button_3 the car will turn left as long as you press the button.

▪ System Design:

• Car System Layer Architecture:



- MCAL Layer Modules:
 - ✓ GPIO Module.
 - ✓ PWM Module.
- HWAL Layer Modules:
 - ✓ Button Module.
 - ✓ Motor Module.
- Application Layer Modules:
 - ✓ Car Module.
 - ✓ Remote Control Module.
 - ✓ System Module.

- Car System Modules APIs:

- **GPIO Module:**

```
//FUNCTIONS PROTOTYPES
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 Module:**

```
//FUNCTIONS PROTOTYPES
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);
```

- **Button Module:**

```
//FUNCTIONS PROTOTYPES
uint8_t ButtonInit(void);
uint8_t ButtonRead(uint8_t ButtonNumber,ptr_uint8_t ButtonState);
uint8_t ButtonStillPressed(uint8_t ButtonNumber);
```

- **Motor Module:**

```
//FUNCTIONS PROTOTYPES
uint8_t MotorInit(void);
uint8_t MotorStart(uint8_t MotorNumber);
uint8_t MotorStop(uint8_t MotorNumber);
uint8_t MotorSetDirection(uint8_t MotorNumber,uint8_t MotorDirection);
uint8_t MotorSetSpeed(uint8_t MotorNumber,uint8_t MotorDuty);
```

- **Car Module:**

```
//FUNCTIONS PROTOTYPES
void CarInit(void);
void CarSet(void);
void CarUpdate(void);
```

- Remote Control Module:

```
//FUNCTIONS PROTOTYPES  
void RemoteControlInit(void);  
void RemoteControlGet(void);  
void RemoteControlUpdate(void);
```

- System Module:

```
//FUNCTIONS PROTOTYPES  
void SystemInit(void);  
void SystemUpdate(void);
```