Software requirements specifications of car control system

- Problem:
  - Controlling two motors of a car through four push-buttons using a microcontroller that has (DIO, Timer & PWM) modules.
- Functional requirements:
  - our system has three speeds.
  - Button_4 configures the speed and direction (forward and backward)
  - Pressing Button_4 will change the car speed. Each button press must move to the next speed:
    - 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%.
    - Speed b: changes the car direction to backward, and speed to 30%.
  - Car won't start move until speed and direction are configured first then only one of the other three buttons is pressed after the configuration:
    - pressing 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 modules:
  - Car control application module
  - Push button module
  - Motor module
  - DIO module
  - PWM module
  - Timer module
- Modules interfaces (APIs):
  - DIO Module:
    - DIO_ERROR_t DIO_PortInit (PORT_t port, DIR_t direction);
    - DIO_ERROR_t DIO_PinInit (PORT_t port, PIN_t pin, DIR_t direction);
    - DIO_ERROR_t DIO_PinEnablePullUp (PORT_t port, PIN_t pin);
    - DIO_ERROR_t DIO_SetPortVal (PORT_t port, Byte value);
    - DIO_ERROR_t DIO_SetPinVal (PORT_t port, PIN_t pin, Byte value);
    - Byte DIO_GetPortVal (PORT_t port);
    - Byte DIO_GetPinVal (PORT_t port, PIN_t pin);
  - Timer Module:
    - TIM_ERROR_t TIM_Init (TIM_SELECT_t Timer, CLK_SELECT_t clock, TIMER_MODE_t mode);
    - TIM_ERROR_t TIM_SetTimerOVFVal (TIM_SELECT_t Timer, u8 u8TimVal);
    - TIM_ERROR_t TIM_SetTimerOCVal (TIM_SELECT_t Timer, u8 u8TimVal);
    - TIM_ERROR_t TIM_SetTimerOVFAction (TIM_SELECT_t Timer, void(*callback)(void));
    - TIM_ERROR_t TIM_SetTimerOCAction (TIM_SELECT_t Timer, void(*callback)(void));

- PWM Module:
  - PWM_ERROR_t PWM_Init (TIM_SELECT_t Timer, PWM_Mode_t Mode);
  - PWM_ERROR_t PWM_Start (duty_cycle, frequency);
- PushButton Module:
  - PSHBTTN_ERROR_t PSHBTTN_Init (PORT_t port, PIN_t pin, PULLUP_Status_t status);
  - PSHBTTN_ERROR_t PSHBTTN_EnablePullUp (PORT_t port, PIN_t pin);
  - Byte PSHBTTN_Status (PORT_t port, PIN_t pin);
- Motor Module:
  - MOTOR_ERROR_t Motor_Init (PORT_t port, PIN_t pin);
  - MOTOR_ERROR_t Motor_Start (PORT_t port, PIN_t pin, SpeedVal_t speed, MDIR_t direction);
  - MOTOR_ERROR_t Motor_Stop (PORT_t port, PIN_t pin);

- System layers (Layered architecture):
  - Application Layer
  - Electronic Unit Abstraction Layer
  - Microcontroller Abstraction Layer

<table>
<tr><td>APP.</td><td colspan="3">APPLICATION LAYER</td></tr>
<tr><td>ECUAL</td><td>PUSH BUTTON</td><td colspan="2">MOTOR</td></tr>
<tr><td>MCAL</td><td>DIO</td><td>PWM</td><td>TIMER</td></tr>
<tr><td colspan="4">MICROCONTROLLER</td></tr>
</table>