**Q) ESP32: Difference between motor-control PWM and LED-control PWM**

*Ans)*

* **LEDC PWM:**
* **The LED control (LEDC) peripheral is primarily designed to control the intensity of LEDs, although it can also be used to generate PWM signals for other purposes as well. It has 16 channels which can generate independent waveforms that can be used, for example, to drive RGB LED devices.**
* **LEDC channels are divided into two groups of 8 channels each. One group of LEDC channels operates in high speed mode. This mode is implemented in hardware and offers automatic and glitch-free changing of the PWM duty cycle. The other group of channels operate in low speed mode, the PWM duty cycle must be changed by the driver in software. Each group of channels is also able to use different clock sources.**
* **The PWM controller can automatically increase or decrease the duty cycle gradually, allowing for fades without any processor interference**
* **MCPWM**
* **ESP32 has two MCPWM units which can be used to control different types of motors. Each unit has three pairs of PWM outputs.**
* **The MCPWM has limited speed but wide pulse-width adjustment. Motors are mechanical devices so feeding them ultra-fast or ultra-short pulses would do no good, unless low pass filters were added, which would waste power.**
* **Motors will work with somewhat fast PWM due to their inertia and high inductance, which acts as a low pass filter.**
* **Normally a motor speed control stays within the range that the motor can filter naturally, due to its inductance. The capture inputs for the motor generate ISRs to judge if motor is overloaded, or an absence of load.**
* **The same ISRs can be used to judge motor current vs. expected current and detect cogging due to a dynamic load. The motors can be synchronous to internal optional clocks.**