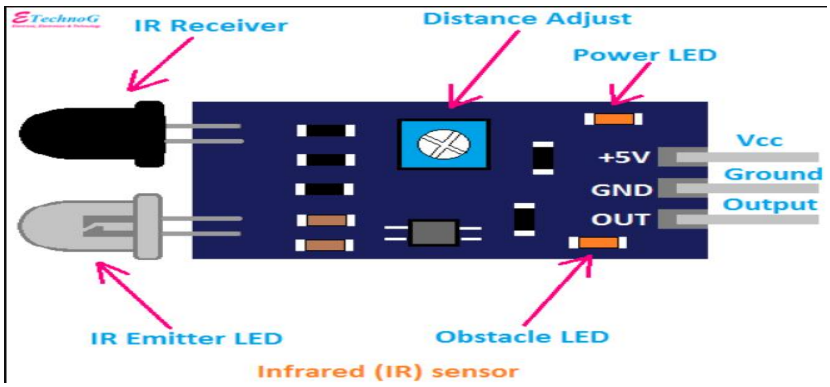
IoT: Theoretical Assignment Answers

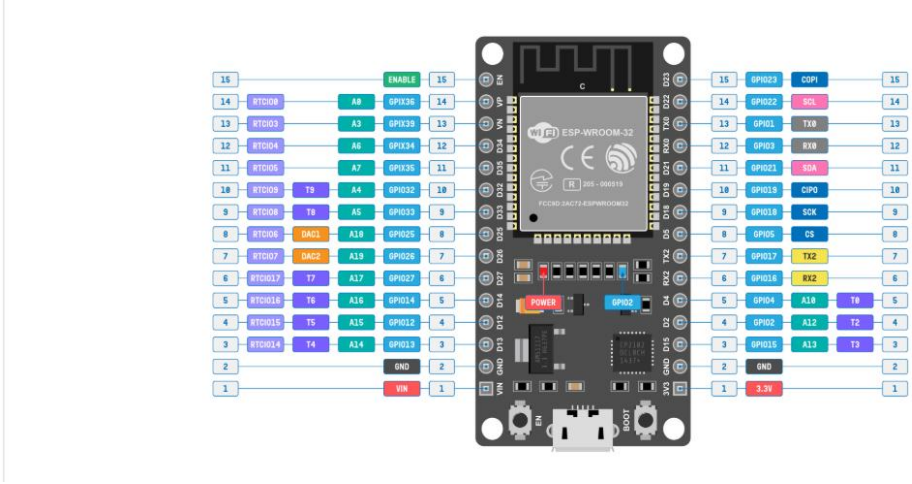
Question 1: IR Sensor Basics

An analog IR sensor outputs a continuous voltage signal that varies with the distance of the object from the sensor. A digital IR sensor outputs a binary signal (either HIGH or LOW) based on whether an object is detected or not. The ESP32 reads analog IR sensors using its ADC (Analog-to-Digital Converter) pins, which convert voltage to digital values (0–4095). Digital IR sensors are read directly using GPIO pins configured as digital inputs.



Question 2: ADC in ESP32

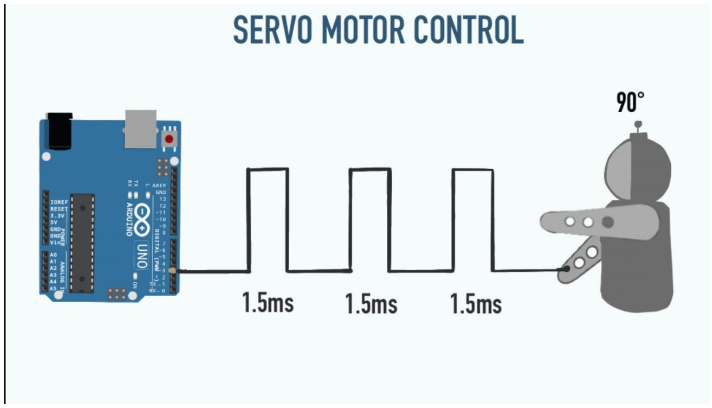
The ESP32 requires an ADC to read analog signals from devices like IR sensors, LDRs, or potentiometers. Analog signals are continuous and cannot be processed directly by digital systems. The ADC samples the voltage from the analog sensor and converts it into a digital value that the ESP32 can understand and process.



Question 3: PWM for Servo Control

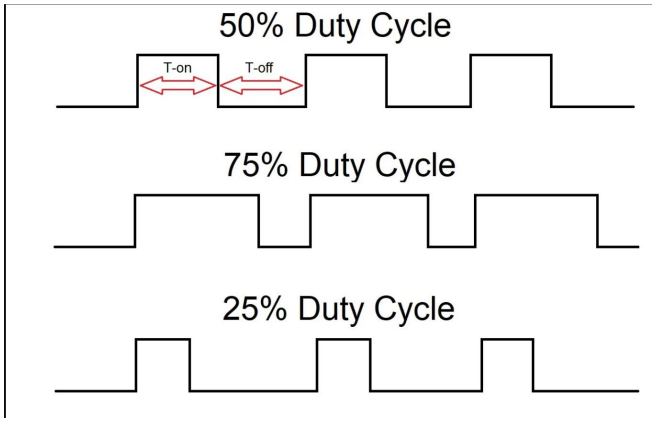
PWM (Pulse Width Modulation) is a technique used to simulate analog signals using digital means. It involves switching the signal between HIGH and LOW rapidly with a certain duty

cycle. Servos require varying signals to position their shaft at different angles. A simple HIGH or LOW cannot represent multiple positions, but varying the PWM signal allows fine control over the servo's angle.



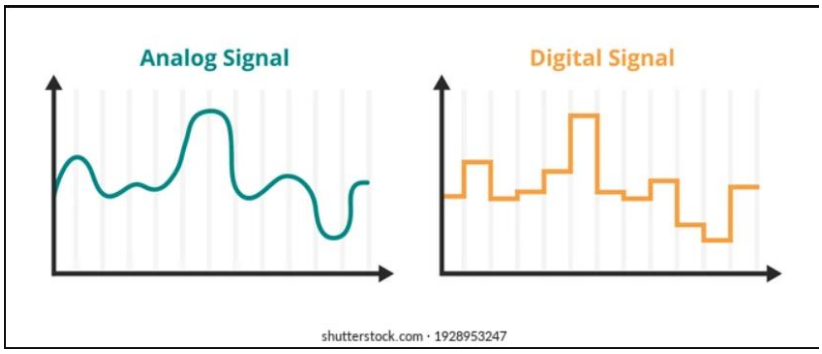
Question 4: Duty Cycle Meaning

The duty cycle in PWM is the percentage of one period in which the signal is HIGH. For example, a 25% duty cycle means the signal is HIGH for 25% of the time and LOW for the remaining 75%. Changing the duty cycle changes the average voltage delivered to the servo, which in turn adjusts its position.



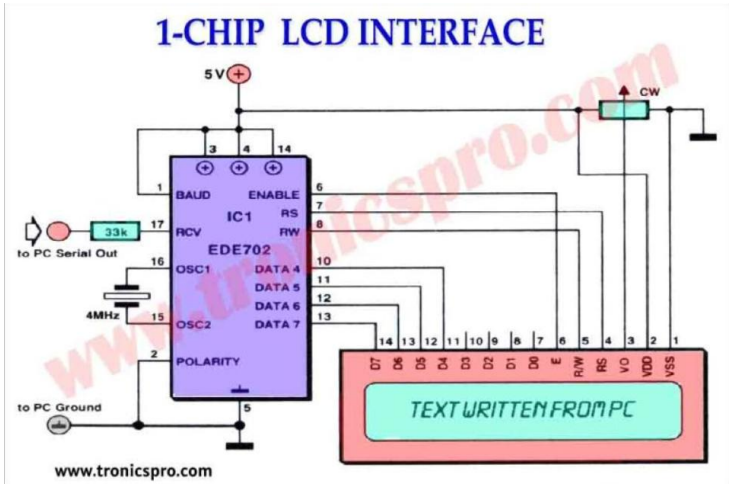
Question 5: Analog vs. Digital Signals

Analog signals are continuous and can take on any value within a range, while digital signals are binary (HIGH or LOW). Sensors like LDRs and IR sensors often produce analog outputs because light and distance vary continuously. In contrast, push buttons have only two states (pressed or not), so they are digital.



Question 6: LCD Communication

The ESP32 can communicate with LCDs using either I2C (two-wire interface) or parallel communication. I2C is often preferred because it uses only two wires (SDA and SCL), reducing pin usage and simplifying wiring. Parallel communication uses more wires and is more complex, though it can be faster.



Question 7: Debouncing Concept

When a push button is pressed or released, the contacts can bounce, causing multiple brief signals. Debouncing eliminates these unwanted signals by adding a delay or software filtering. Without debouncing, a single press may be detected as multiple presses, leading to unreliable behavior when controlling an LED or servo.