**National Textile University, Faisalabad**



**Department of Computer Science**

|  |  |
| --- | --- |
| **Name:** | Muhammad Haseeb |
| **Class:** | BSCS 5th (Section – B) |
| **Registration No:** | 23-NTU-CS-1069 |
| **Lab Report:** | Assignment |
| **Course Name:** | Embedded IOT Systems |
| **Submitted To:** | Sir. Nasir Mahmood |
| **Submission Date:** | 19th Oct – 2025 |

***Question:01***

**1. Why is volatile used for variables shared with ISRs?**

**Ans:**

Volatile is the opposite of const. As const makes sure the value of a variable remains the same throughout its execution. Therefore, a Volatile variable means the value of the variable can be changed; it does not remain the same.

**2. Compare hardware-timer ISR debouncing vs. delay()-based debouncing.**

**Ans:**

|  |  |
| --- | --- |
| **Hardware Timers:**   * It only pauses a specific part of the code, as desired by the user. * The rest of the code keeps on working. | **Delay() - Band:**   * It pauses everything. |

**3. What does IRAM\_ATTR do, and why is it needed?**

**Ans:**

* Stores the function in internal RAM, not along the main code.
* So it runs only when required, else stays dormant.
* Used to handle interrupts.

**4. Define LEDC channels, timers, and duty cycle.**

**Ans:**

Channel: It controls output via a specific pin.  
Timer : It sets the frequency.  
Duty Cycle: It is the percentage of the ON signal.

**5. Why should you avoid Serial prints or long code paths inside ISRs?**

**Ans:**

Our ISR (Interrupt Service Routine) needs to be fast.

* Using Serial.print() means giving info about the current condition, then executing the ISR.
* Also, long code would need more CPU time to execute.

Hence, we keep our ISR as simple as possible.

**6. What are the advantages of timer-based task scheduling?**

**Ans:**

It helps in running the code smoothly. It uses the concept of multiprogramming, working on multiple parts at once - unlike delay() that pauses everything.

**7. Describe I²C signals SDA and SCL.**

**Ans:**

**SDA:** Carries the data between devices.  
**SCL:** Carries the clock signal, which controls the timing.

**8. What is the difference between polling and interrupt-driven input?**

**Ans:**

|  |  |
| --- | --- |
| **Polling:**   * Continuously checking the hardware if it needs attention. | **Interrupt:**   * A signal generated by the hardware, telling it needs attention. |

**9. What is contact bounce, and why must it be handled?**

**Ans:**

When a button is pressed, the metal plates do not make contact once. The current bounces for a few milliseconds. It rapidly turns on and off for a short time before stabilizing. Therefore, we use debouncing.

**10. How does the LEDC peripheral improve PWM precision?**

**Ans:**

LEDC (LED Control) uses the hardware timers. Hence, it works on its own without the need for the CPU.

**11. How many hardware timers are available on the ESP32?**

**Ans:**

Two timer groups:

1. Group 0: Timer0 & Timer1
2. Group 1: Timer2 & Timer3

Each has two timers. Hence, total 4 Hardware Timers (64-bit).

**12. What is a timer prescaler, and why is it used?**

**Ans:**

It is a clock divider. It divides the main clock speed to create a slower time. Why? To get a desired time for timing/waiting.

**13. Define duty cycle and frequency in PWM.**

**Ans:**

Duty Cycle: Percentage of time the signal is ON during a cycle.  
Frequency: Number of times a signal cycles (on and off) in a second.

**14. How do you compute duty for a given brightness level?**

**Ans:**

**15. Contrast non-blocking vs. blocking timing.**

**Ans:**

|  |  |
| --- | --- |
| **Non-Blocking:**   * Runs everything in parallel. * Uses millis(). | **Blocking:**   * Stops everything until time passes. * Uses delay(). |

**16. What resolution (bits) does LEDC support?**

**Ans:**

Supports up to 20-bit.

**17. Compare general-purpose hardware timers and LEDC (PWM) timers.**

**Ans:**

|  |  |
| --- | --- |
| **Hardware Timers:**  Used for scheduling. | **PWM Timers:**  Used for creating frequency. |

**18. What is the difference between Adafruit\_SSD1306 and Adafruit\_GFX?**

**Ans:**

|  |  |
| --- | --- |
| **SSD1306:**   * Controls the OLED display hardware. * It is the "canvas" | **GFX (Adafruit\_GFX Library):**   * Provides the drawing and text functions. * It is the "art kit." |

**19. How can you optimize text rendering performance on an OLED?**

**Ans:**

* Update only what is required.
* Use simple fonts.
* Avoid full screen clears.

**20. Give short specifications of your selected ESP32 board (NodeMCU-32S).**

**Ans:**

**Core:** Dual-Core processors  
**Speed:** Up to 240 MHz  
**Memory:** 4MB Flash Storage, 520 KB RAM  
**Connectivity:** Wi-Fi and Bluetooth  
**Pins:** 30  
**Voltage:** 3.3V

***Question:02***

**1. A 10 kHz signal has an ON time of 10 ms. What is the duty cycle? Justify with the formula.**

**Ans:**

**Data:**

f = 10 KHz = 10000 Hz

= 10 ms = 0.01 ms

T =

**Sol:**

**Conclusion:**

10,000% not possible. Therefore; we take 100%.

**2. How many hardware interrupts and timers can be used concurrently? Justify.**

**Ans:**

|  |  |
| --- | --- |
| **Hardware Timers:**  Because the ESP-32 itself has only 4 timers. Only 4. | **Hardware Interrupts:** All GPIO pins. Each pin can be configured individually; the limit is the number of pins (GPIO). |

**3. How many PWM-driven devices can run at distinct frequencies at the same time on ESP32? Explain constraints.**

**Ans:**

**PWM Timers:**

Hence, **8** different frequencies.

**Constraint:**

* 1. Channels that share a timer also share the frequency condition.
  2. Duty cycle can be different when the timer is the same.

**4. Compare a 30% duty cycle at 8-bit resolution and 1 kHz to a 30% duty cycle at 10-bit resolution (all else equal).**

**Ans:**

|  |  |
| --- | --- |
| **8-bit Resolution** Max Value =  At 30% Duty Cycle (8-bit):  Find Duty % from Value 77 (8-bit): | **10-bit Resolution** Max Value =  At 30% Duty Cycle (10-bit):  Find Duty % from Value 307 (10-bit): |

**5. How many characters can be displayed on a 128×64 OLED at once with the minimum font size vs. the maximum font size? State assumptions.**

**Ans:**

|  |  |
| --- | --- |
| **Minimum Font**  Font Size: 5x8 pixels  **Total Bytes for Screen:**  **Total Bytes per Character:**  **Final:**  Total Characters (assuming 5 bytes per character): | **Maximum Font**  Font Size**:** 16x32 pixels  **Total Bytes for Screen:**  **Total Bytes per Character:**  **Final:**  Total Characters (assuming 64 bytes per character): |