**Name: Malaika**

**RegNo : 23-NTU-CS-1291**

**AI-5th**

**Embedded IoT Systems (CSE-3080)**

**Fall 2025** (BSCS-5

A/B, BSAI-5)

# Assignment 1

## Question 1 — Short Questions

Answer concisely, in your own words.

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

**Answer:**

Volatiles used for variables because volatile **informs** **compiler** that the behavior of variables changes unexpectedly by ISR. In case of skip re-reading the variable, it is also used to **prevent optimization** and ensures that the main code always **focus** on the **latest correct value**.

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

**Answer:**

**Hardware-timer ISR:**

* Used for precise, accurate, non-blocking debounce timing.
* During handling of debounce, it allows other tasks to run smoothly.
* Ideal for real-time systems where the first priority is timing accuracy.

**Delay()-based debouncing:**

* This type of debouncing uses software delay, which blocks CPU during the wait time of process.
* It can miss the frequent or fast input changes, and prevents multitasking.
* Inefficient for real-time or multitasking applications.

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

**Purpose:**

IRAM\_ATTR uses to place ISR code in Internal RAM instead to store in flash memory.

**Need:**

IRAM\_ATTR uses on ESP32 , because flash memory may be inaccessible for some time during use .It also uses to execute interrupt routines faster and more reliable.

1. Define LEDC channels, timers, and duty cycle.

**Answer:**

**LEDC channels:**

A channel is defined as independent PWM output that control devices e.g. LEDS, buzzers, motors.

**Timers:**

A timer is defined as the hardware module that sets the frequency and resolution.

**Duty Cycle:**

It is defined as the ratio of ON time of signal to its total time period , expressed as percentage.

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

**Answer:**

**Reason:**

We avoid serial prints and long code paths to keep ISRs limited, efficient and fast. Because serial operations are slow and block interrupt handling. Long codes delay interrupts and disturb timing.

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

**Answer:**

**Advantages:**

* Timer-based task scheduling enables tasks to run automatically at precise intervals without blocking main loop.
* It allows multitasking efficiently by handling periodic events.
* It improves accuracy of time resulting in reducing CPU idle time.

1. Describe I²C signals SDA and SCL.

**Answer:**

**I²C signals:**

I²C (Inter-Integrated Circuit) is a two-wire serial communication protocol.

**SDA:** Stands for “Serial Data” used for transmitting data among devices.

**SCL:** Stands for “Serial Clock” used for synchronization**.**

Both SDA and SCL pulled by high resistors and are open drain configuration.

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

**Answer:**

**Polling:** Checks interrupts continuously in a loop, result in wasting CPU idle time.

**Interrupt-driven**: Inputs that take action only at the exact position when an event occurs without wasting CPU idle time. It improves responsiveness and save processing time.

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

**Answer:**

**Contact Bounce:** A rapid, unintended switching that occurs when a switch or button is pressed called contact bounce**.**

**Handling:** Needed to handle by debouncing because it generates wrong triggers in digital circuits. Debouncing makes sure that only one clean process executes per press.

1. How does the LEDC peripheral improve PWM precision?

**Answer:**

LEDC peripheral uses high resolution hardware-timers for PWM

generation. It provides precise and accurate control of duty cycle and frequency. In this way it improves signal accuracy and reduces flickers in LEDs or motors.

1. How many hardware timers are available on the ESP32?

**Answer:**

ESP32 has total 2 groups having 4 hardware timers per group. So, total hardware timers are 8 on ESP32 operate independently with its own settings.

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

**Answer:**

Timer prescaler, divides base clock frequency and slows down timer to achieve long time interval. Used to make precise adjustments in resolution to improve the performance nd accuracy for PWM or delays.

1. Define duty cycle and frequency in PWM.

**Answer:**

**Duty Cycle:**

It is defined as the ratio of ON time of signal to its total time period , expressed as percentage.

**Frequency:**

Total number of signal’s repetition in a second called frequency.SI unit is Hertz (Hz).In PWM ,frequency is how fast the ON-OFF cycle occurs.

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

**Answer:**

**Duty for brightness level**= Desired Brightness/Maximum Brightness \*Maximum Duty

1. Contrast non-blocking vs. blocking timing.

**Non-Blocking Time:**

It uses timers that allows other tasks to run smoothly. It improves responsiveness and multitasking.

**Blocking timing:**

It uses software delay, which blocks CPU during the wait time of process. Stops all other tasks to run smoothly.

1. What resolution (bits) does LEDC support?

**Answer:**

LEDC supports 20 bits of resolution. Duty cycle support by LEDC equivalent to 2^20=1048576 discrete levels**.**

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

**Answer:**

**General-purpose hardware timers**: Used for timing delays and interrupts in order to handle multitasking and to improve responsiveness.

**LEDC timers**: Specialized timers used for the generation of PWM signals by adjusting the frequency and duty cycle timing. Used to handle multiple PWM channels**.**

1. What is the difference between Adafruit\_SSD1306andAdafruit\_GFX?

**Answer:**

**Adafruit\_SSD1306:** It is used to control Organic Light Emitting Diode (OLED) display hardware like communication pixels with(128x64).

**Adafruit\_GFX:** It is used for graphics in OLED in order to draw functions like text , shapes etc.

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

**Answer:**

We can optimize text rendering

* By using small fonts,
* By avoid using complicated graphics or frequent screen clears,
* By minimizing screen updates to increase speed in Adafruit\_GFX library.

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

**Answer:**

We us **NodeMCU-32S**, because it uses the **ESP32** dual-core 32-bit processor at up to **240 MHz**.  
It provides **Wi-Fi + Bluetooth**, and **30 GPIO pins**.  
Operates on **3.3 V logic** with **micro-USB** programming and power.

## Question 2 — Logical Questions

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

**Answer:**

**Duty cycle=**

**Total time = ==0.0001s**

**Duty cycle=0.01/0.0001**

**Duty cycle=100 x 100=10,000% shows inconsistency**

**Duty cycle in microsecond =?**

**T=100microsecond**

**T (on) =10 microsecond**

**Duty cycle =**

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

**Answer:**

ESP32 has total 2 groups having 4 hardware timers per group. So, total hardware timers are 8 on ESP32 operate independently with its own settings. Various other interrupts like GPIO, also run concurrently, managed by CPU interrupt handler.

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

**Answer:**

**Eight** PWM-driven devices can run at distinct frequencies at the same time on ESP32.

**Constraints:** PWM channels uses same LEDC timer having same frequency but their duty cycles are different. 8 number of

LEDC timer used, so only eight frequencies can exist.

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

**Answer:**

**Duty cycle**=30%

**T= 1/f** =1/1000 =1ms =100 micro sec

**For 8-bit Resolution:**

Total levels = = 256

Duty count=30% of 256=0.3x256 =76.8

ONN steps=76.8

**For 10-bit resolution:**

Total levels=

=1024

Duty count= 30% of 1024 = 0.3x1024=307.2

**Final Result:**

Both Resolutions give same ON –OFF time. Both give O.3ms (Ton) and 0.7ms (T off).But 10 bit-resolution give 4 times faster smoother brightness as compared to 8-bit because of 1024 levels.

1. 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.

**Answer:**

**Minimum Font size (4x8 pixels):**

Characters per line = = 32

Lines= =8

Total characters= =256

**Maximum Font size (24x32pixels):**

Characters per line = = 5

Lines= =2

Total characters= =10

## Guidelines

### Repository & Submission

* Repository: Use your existing course repository; add a folder: assignment1-<regno> • Include folders:

/src, /include, /docs, /wokwi (JSON exported from Wokwi)

/screenshots (photos or short clip showing different outputs)

* Push to GitHub with a README.md containing the Wokwi link, pin map, and screenshots.
* Submit a compiled PDF on MS Teams before the deadline.

### Coding & Documentation

* Use your own circuit design with chosen pins and clearly colored wiring/components.
* Write code in your own style with clear, technical variable names and meaningful comments.
* Begin each program with a comment block containing the title, your name, and registration number.
* Keep answers to short and logical questions concise and original.

### Academic Integrity

* All material must be defensible and understandable to you.
* Avoid direct copy-paste from any source (students, websites, etc.).
* You may discuss ideas, but the final submitted work must be your own.
* The viva will be held in Week 7.

## Marking Rubric (50 points)

*Evaluation Criteria Marks*

|  |  |
| --- | --- |
| *Circuit design (originality)*  *Interrupt stability*  *Debounce accuracy*  *OLED updates*  *Documentation (code + circuit demo)*  *Short questions (clarity of understanding)*  *Logical questions (reasoning and justification) Total* | 5 |
| 5 |
| 5 |
| 5 |
| 10 |
| 10 |
| 10 |
| 50 (All marks are viva-based) |

## Deadline

Submission Deadline(Q1,2): Sunday, 19 October 2025 (PDF on MS Teams).

Submission Deadline(Complete): Sunday, 26 October 2025 (code on GitHub).

As the deadline of code is still several days away, make good use of the time to fully understand the concepts and develop your own original work.