National Textile University, Faisalabad

Department of Computer Science

Name:	Muhammad Talal Shariq
Class:	BSCS-B 5 Semester
Registration No:	23-NTU-CS-1074
ASSIGNMENT :	NO 1
Course Name:	IOT
Submitted To:	SIR NASIR
Submission Date:	18/10/2025

Q NO 1:

Why is volatile used for variables shared with ISRs?

Volatile tells the compiler to not to optimize the variable because its value can be changed unexpectedly i.e maybe by an interrupt. Volatile is used with the variables that can change .They are not fixed variables.

Q NO 2:

Compare hardware interrupts and ISRs vs. software polling:

Interrupts and ISR: Cpu is alerted by hardware when event occurs. It has immediate response.complex to implement . suitable for time critical tasks.

EX : attachInterrupt(digitalPin, ISR, mode);

Software Polling : cpu repeatedly checks for an event.response time depends on polling interval.cpu stays active during it. Simpler to implement.

EX : if (digitalRead(pin) == HIGH) { ... }

Q NO 3:

What does IRAM_ATTR do, and why is it used?

It is a function in instruction ram for faster execution.it is used for isr to ensure they run quickly and reliably specially when access is low.

Q NO 4:

Define LEDC channels, timers, and duty cycle:

- **CHANNELS:** output pins for pwm signals
- **TIMERS:** control frequency and resolution
- **DUTY CYCLE**: percentage of time the signal is high in one signal

Q NO 5:

Why avoid Serial prints in long ISR code paths?

Serial prints are slow and can block execution. In ISRs , this delays response and may cause interrupts and crashes.

Q NO 6:

What are advantages of timer based task scheduling ?

- **Precise Timing Control** Tasks can be triggered at exact intervals, ensuring consistent behavior (e.g., sensor sampling every 100 ms).
- **Non-Blocking Execution** Timers allow tasks to run independently of the main loop, avoiding delays caused by blocking functions like delay().
- **Improved CPU Efficiency** The processor can sleep or perform other tasks while waiting for a timer event, reducing idle time.
- **Real-Time Responsiveness** Critical tasks (e.g., motor control, communication) can be scheduled with minimal latency.
- **Multitasking Support** Multiple timers can manage different tasks concurrently, enabling parallel operations.
- **Event-Driven Architecture** Timers help structure code around events rather than loops, making it cleaner and more modular.
- **Power Saving** In low-power systems, timers can wake the CPU only when needed, conserving energy.

Q NO 7:

Describe I²C signals SDA and SCL:

- **SDA:** Carries data

- **SCL:** synchronize data transfer

Q NO 8:

Difference between polling and interrupt-driven input:

- **Polling**: CPU repeatedly checks input.

- **Interrupt-driven**: CPU is notified only when input changes.

Q NO 9:

What is contact bounce, and why is it an issue?

Mechanical switches may be rapidly open , close when pressed causing multiple signals.

This leads to false triggers unless debounced.

The main issue here is false triggers cause by these multiple signals know as contact bounce.

Q NO 10:

How does LEDC improve PWM precision?

LEDC uses high resolution timers.

Upto 20 bit timers.

These allow precise duty cycles and stable frequencies.

Q NO 11 :

How many hardware timers on ESP32?

ESP has 4 general purpose hardware timers.

These are grouped into 2 timer groups.

Q NO 12:

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

It reduces the speed of timers counting by a fixed factor.

If system clock is 80 mhz and prescaler is set to 80 then timer will count at 1 mhz.

It is used to extend timer range for longer delays.

To reduce power consumption.

To match timing requirements.

Q NO 13:

Define duty cycle and frequency:

- **Duty Cycle**: $\frac{\text{ON time}}{\text{Total period}} \times 100\%$

- **Frequency**: Number of cycles per second (Hz).

Q NO 14 :

How to compute duty for a given brightness level?

Duty Cycle Formula

$$\text{Duty} = \frac{\text{Brightness Level (\%)}}{100} \times (2^{\text{Resolution}} - 1)$$

If you want 70% brightness at 8-bit resolution:

$$\text{Duty} = \frac{70}{100} \times (2^8 - 1) = 0.7 \times 255 = 178.5 \approx 179$$

Q NO 15 :

Contrast non-blocking vs. blocking timing:

BLOCKING: Halts execution.

NON-BLOCKING: uses timers for millIseconds allowing multitasking

Q NO 16:

What resolution (bits) does LEDC support?

LEDC supports 1 to 20 bits resolution.

Q NO 17:

Compare general-purpose timers vs. LEDC timers:

- **General-purpose timers** are versatile hardware modules used for counting time intervals, generating delays, and triggering interrupts. They can operate in one-shot or periodic modes and are ideal for tasks like measuring time between events or scheduling operations.
- **LEDC timers** are specialized timers designed specifically for generating PWM (Pulse Width Modulation) signals. They control the frequency and resolution of PWM output across multiple channels, making them suitable for applications like LED dimming, motor control, and audio signal generation.

Q NO 18:

Difference between Adafruit_SSD1306 and Adafruit_GFX:

- SSD 1306 is a driver for oled hardware.

- Adafruit gfx is graphics library for drawing shapes, text etc.

Q NO 19:

How to optimize text rendering on OLED screens:

- Use smaller fonts
- Minimize redraws
- Avoid full screen updates
- Use diplay.display()

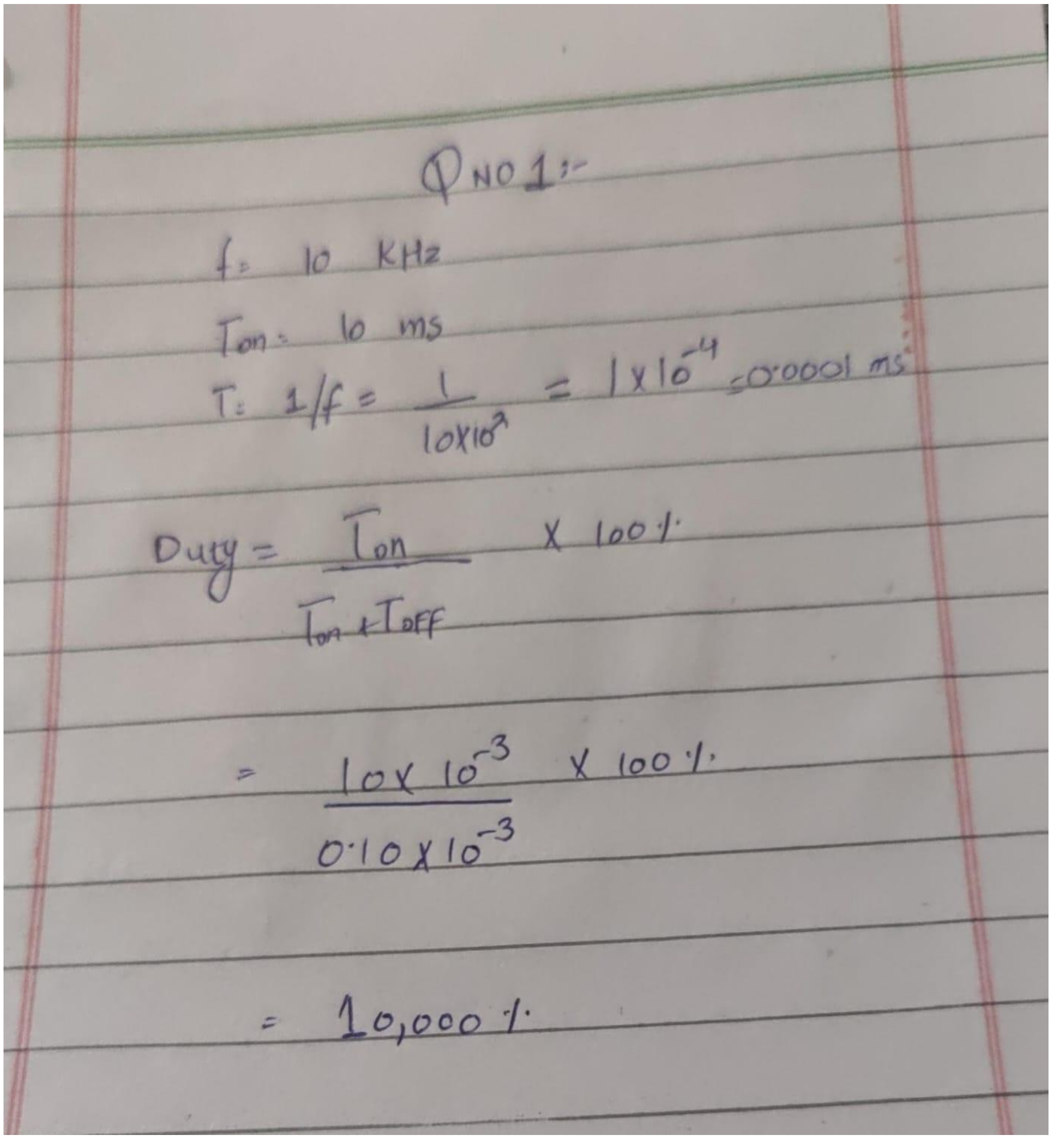
Q NO 20:

Short specs of selected display :

- **Microcontroller**: ESP32 (dual-core Xtensa LX6 processor)
- **Clock Speed**: Up to 240 MHz

- **Flash Memory**: 4 MB
- **SRAM**: 520 KB
- **Connectivity**:
- **Wi-Fi**: 802.11 b/g/n
- **Bluetooth**: v4.2 BR/EDR and BLE
- **GPIO Pins**: 30+ available, including digital, analog, PWM, I^{2}C, SPI, UART
- **Voltage**: 3.3V or 5V
- **Color**: Monochrome

Question 2 — Logical Questions



Q NO 1:-

$f = 10$ KHz

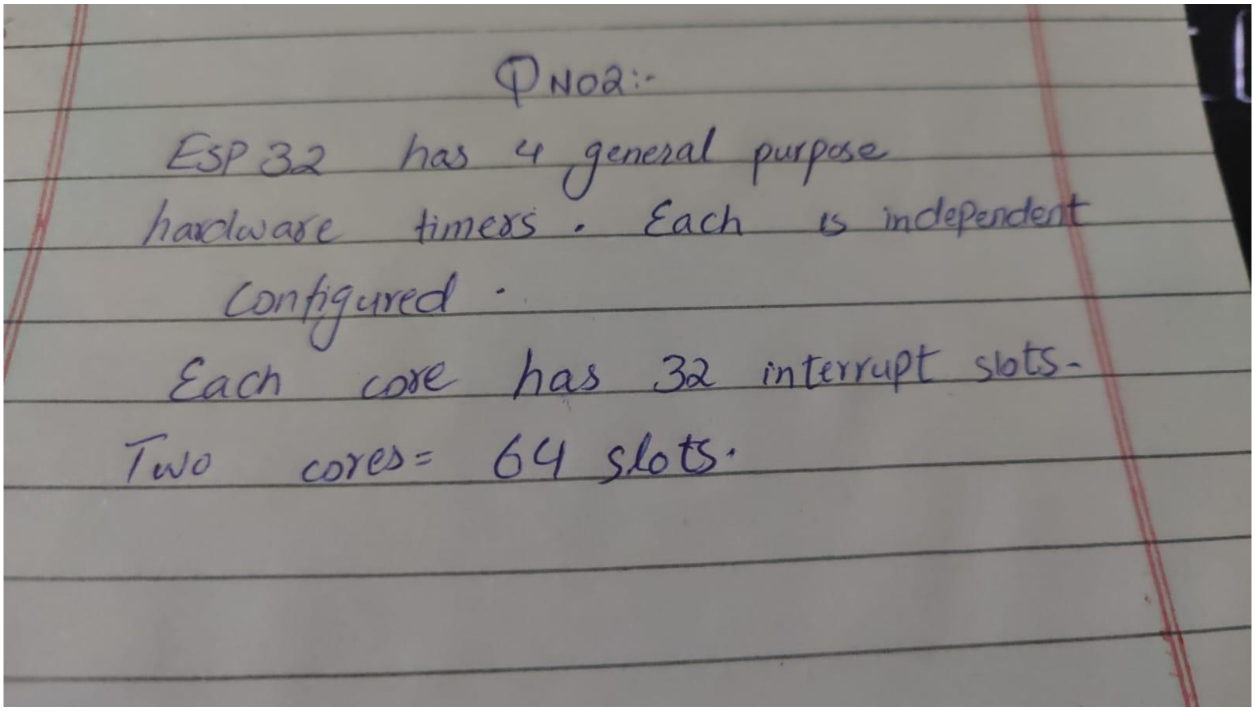
$T_{on} = 10$ ms

$T = 1/f = \frac{1}{10\times10^{3}} = 1\times10^{-4} = 0.0001$ ms

$$\text{Duty} = \frac{T_{on}}{T_{on}+T_{OFF}} \times 100\%$$

$$= \frac{10\times10^{-3}}{0.10\times10^{-3}} \times 100\%$$

$$= 10{,}000\%$$

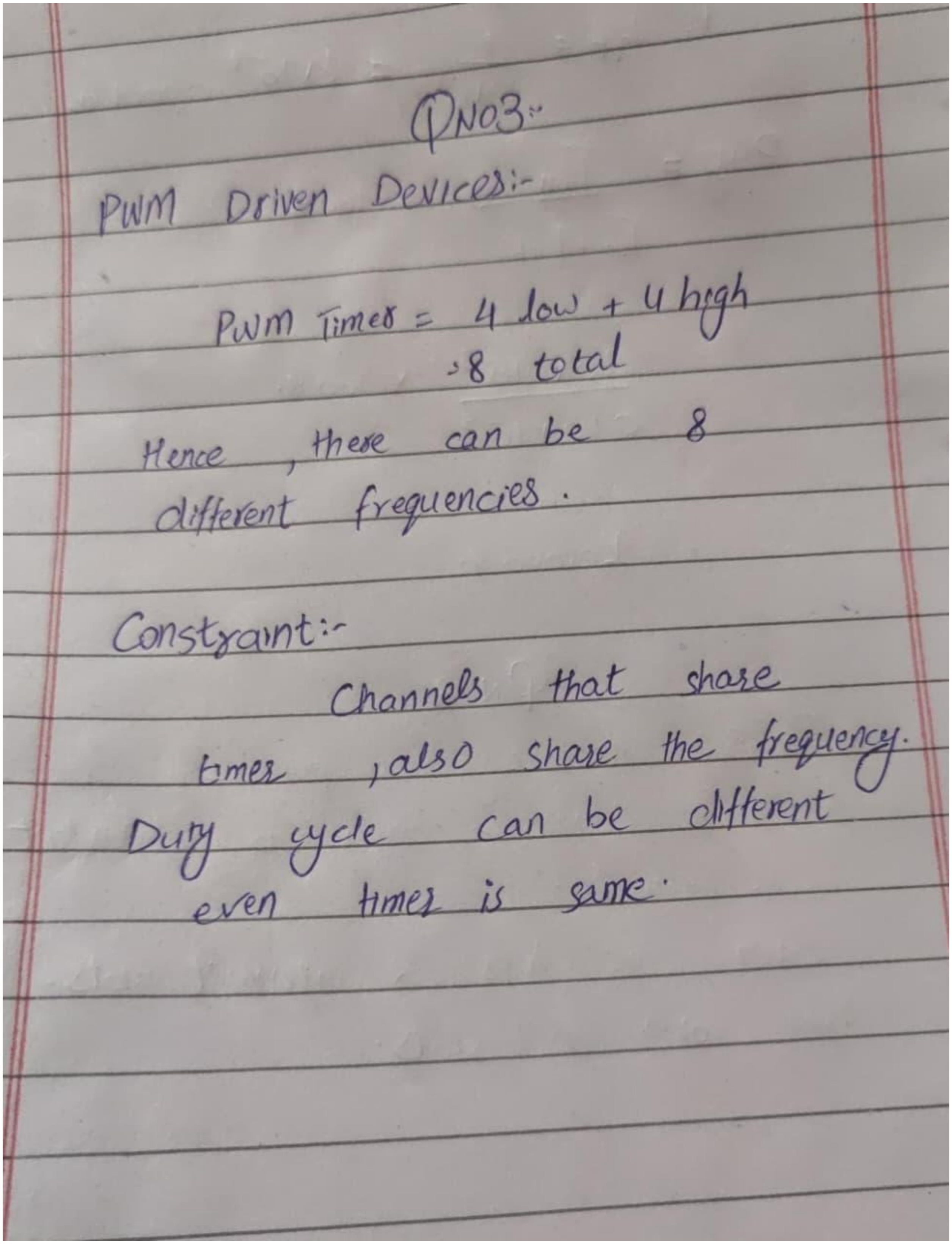


Q NO2:-

ESP 32 has 4 general purpose hardware timers. Each is independent configured.

Each core has 32 interrupt slots.

Two cores= 64 slots.



QNo3:-

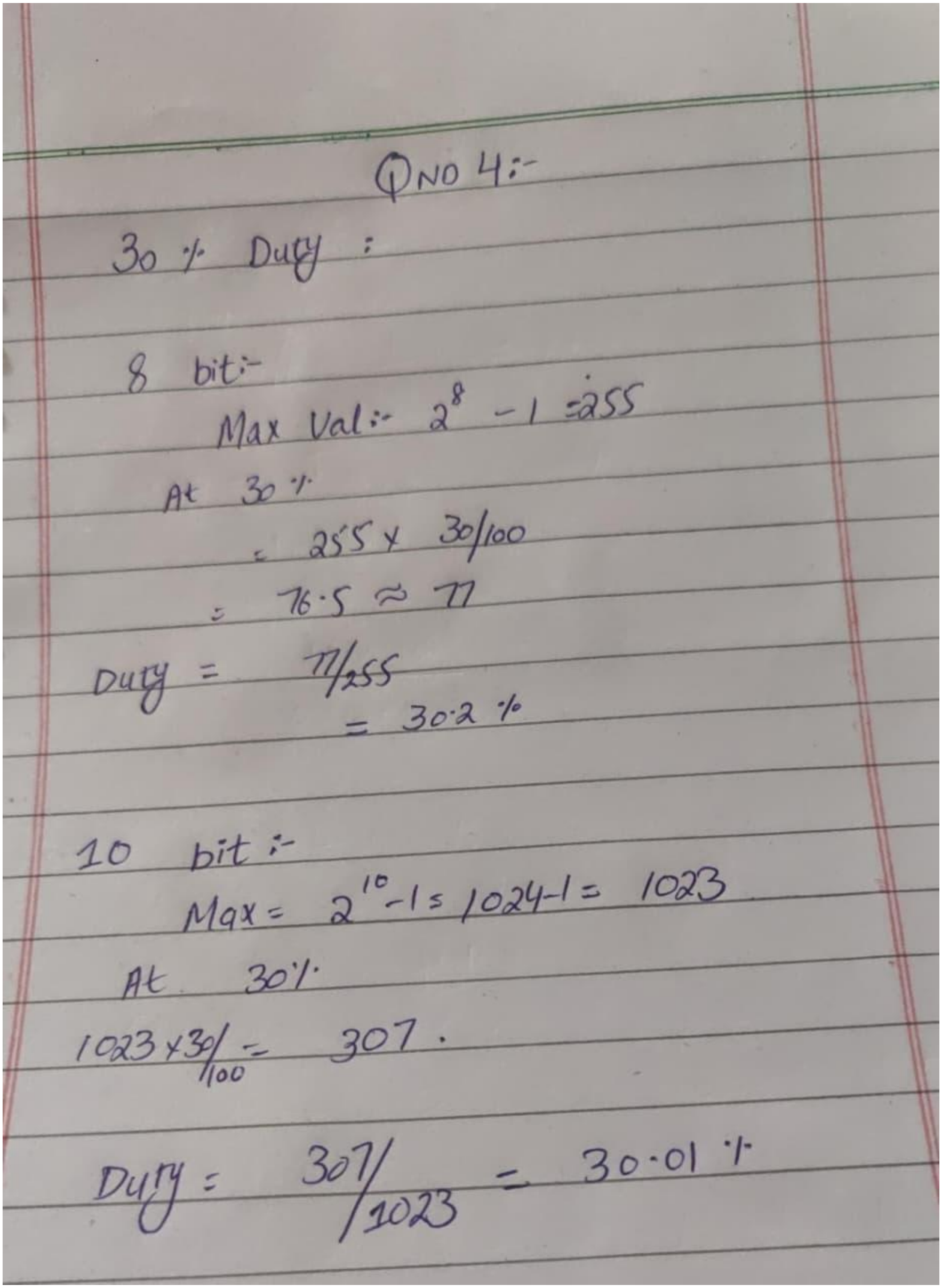
PWM Driven Devices:-

PWM Timer = 4 low + 4 high
= 8 total

Hence, there can be 8 different frequencies.

Constraint:-

Channels that share timer, also share the frequency. Duty cycle can be different even timer is same.



Q NO 4:-

30 % Duty :

8 bit:-

Max Val:- $2^8 - 1 = 255$

At 30 %

$= 255 \times 30/100$

$= 76.5 \approx 77$

Duty $= 77/255$

$= 30.2$ %

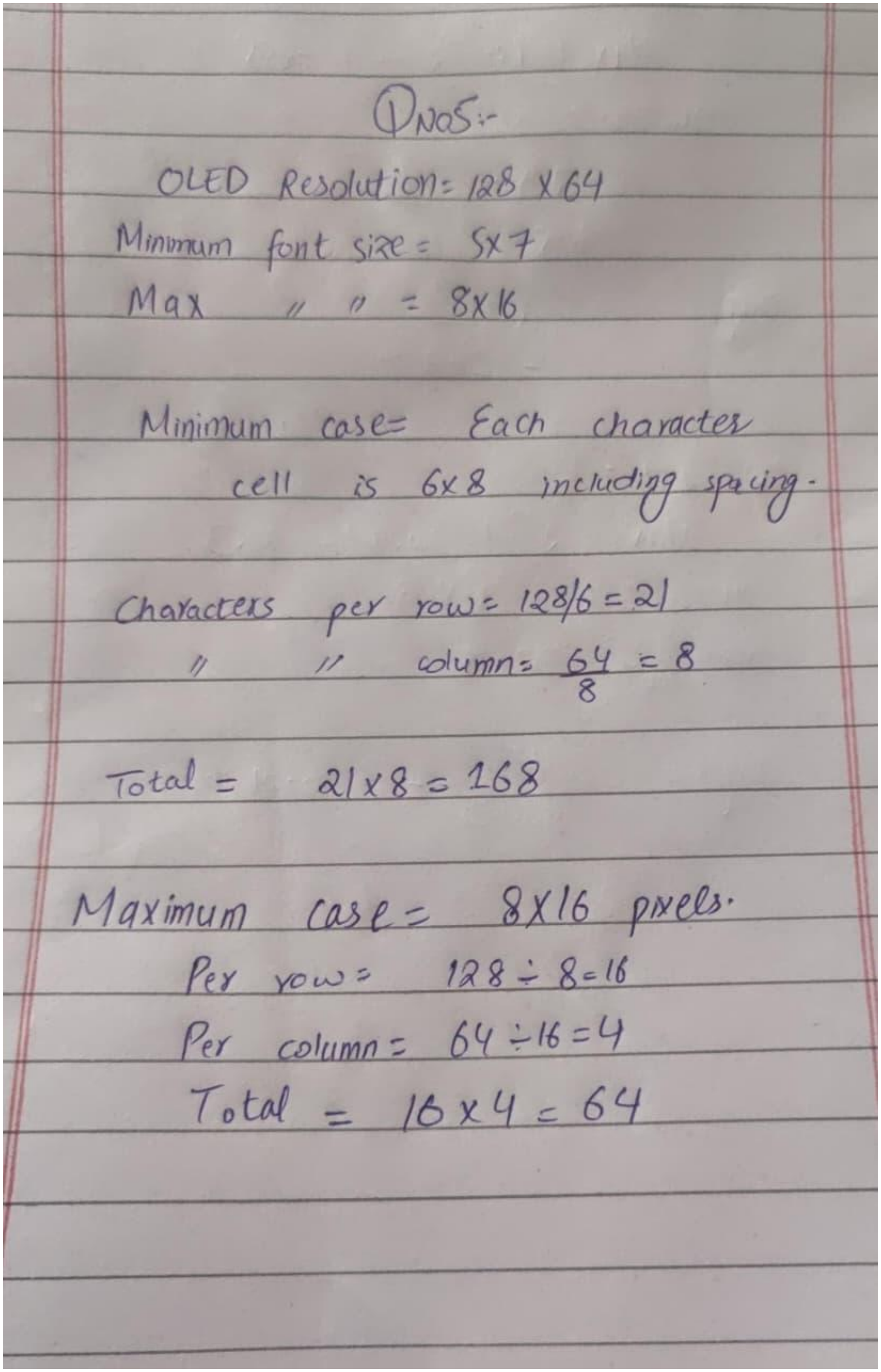
10 bit :-

Max $= 2^{10} - 1 = 1024 - 1 = 1023$

At 30%

$1023 \times 30/100 = 307$.

Duty $= \frac{307}{1023} = 30.01$ %



Q No5:-

OLED Resolution= 128 X 64

Minimum font size = 5x7

Max " " = 8x16

Minimum case= Each character cell is 6x8 including spacing.

Characters per row= 128/6 = 21

" " column= $\frac{64}{8}$ = 8

Total = 21x8 = 168

Maximum case = 8x16 pixels.

Per row= 128 ÷ 8=16

Per column = 64÷16=4

Total = 16x4 = 64