A1

(Part 1) At the beginning, the contents of the SPI shift registers will be:

$$(\mu C \rightarrow \text{Device \#1} \rightarrow \text{Device \#2} \rightarrow \text{Device \#3} \rightarrow \text{Device \#4} \rightarrow)$$
$$(\text{0xFF} \rightarrow \text{0xFF} \rightarrow \text{0xFF} \rightarrow \text{0xFF} \rightarrow \text{0xFF} \rightarrow)$$

Now, after 8 clock cycles, the first data (corresponded to the last device, i.e., #4), will be shifted in. So,

$$(\mu C \rightarrow \text{Device \#1} \rightarrow \text{Device \#2} \rightarrow \text{Device \#3} \rightarrow \text{Device \#4} \rightarrow)$$
$$(\text{0x7C} \rightarrow \text{0x8D} \rightarrow \text{0xFF} \rightarrow \text{0xFF} \rightarrow \text{0xFF} \rightarrow)$$

Now, after 16 clock cycles, the second data (corresponded to the last device, i.e., #3), will be shifted in. So,

$$(\mu C \rightarrow \text{Device \#1} \rightarrow \text{Device \#2} \rightarrow \text{Device \#3} \rightarrow \text{Device \#4} \rightarrow)$$
$$(\text{0x6B} \rightarrow \text{0x7C} \rightarrow \text{0x8D} \rightarrow \text{0xFF} \rightarrow \text{0xFF} \rightarrow)$$

Similarly, after 32 clock cycles, all fourth data (corresponded to all devices, i.e., #3), will be shifted in. So,

$$(\mu C \rightarrow \text{Device \#1} \rightarrow \text{Device \#2} \rightarrow \text{Device \#3} \rightarrow \text{Device \#4} \rightarrow)$$
$$(\text{0xFF} \rightarrow \text{0x5A} \rightarrow \text{0x6B} \rightarrow \text{0x7C} \rightarrow \text{0x8D} \rightarrow)$$

(Part 2) Each bit takes $\frac{1}{500{,}000\ kHz} = 2\,\mu s$. So, for 32 bits (4 bytes), the total transmission time is:

$$32 \times 2\,\mu s = 64\,\mu s = 0.064\,ms.$$

(Part 3) At 250 KHz, each bit takes $4\,\mu s$. This slower clock causes a mismatch in timing expectations between the master and devices (Device #1 will sample every single bit TWICE). The timing misalignment would likely result in Device #4 receiving incomplete or corrupted data, as data depends on precise clock synchronization.

The following is a sample of data transmission for the first byte (0x8D after 8 clock cycles):

Master Clock Cycle	Master Sends	Slave Samples	Device #1 Register	Device #2 Register
1	1	11	11111111	11111111
2	0	00	11111100	11111111
3	0	00	11110000	11111111
4	0	00	11000000	11111111
5	1	11	00000011	11111111
6	1	11	00001111	11111100
7	0	00	00111100	11110000
8	1	11	11110011	11000000

So, `0xF3 (11110011)` instead of `0x8D` is now stored in Device #1.

Similarly, we can calculate the other devices values. For Device #4, after sending all four data (after 32 clock cycles of 250 KHz), the stored data will be: `0x3C (00111100)`

(Part 4) Device #2 reverses all data through the chain. So,

$$(\mu C \rightarrow \text{Device \#1} \rightarrow \text{Device \#2} \rightarrow \text{Device \#3} \rightarrow \text{Device \#4} \rightarrow)$$
$$\text{at clock cycle 0: } (\text{0x8D} \rightarrow \text{0xFF} \rightarrow \text{0xFF} \rightarrow \text{0xFF} \rightarrow \text{0xFF} \rightarrow)$$
$$\text{at clock cycle 8: } (\text{0x7C} \rightarrow \text{0x8D} \rightarrow \text{0xFF} \rightarrow \text{0x00} \rightarrow \text{0xFF} \rightarrow)$$
$$\text{at clock cycle 16: } (\text{0x6B} \rightarrow \text{0x7C} \rightarrow \text{0x72} \rightarrow \text{0x00} \rightarrow \text{0x00} \rightarrow)$$
$$\text{at clock cycle 24: } (\text{0x5A} \rightarrow \text{0x6B} \rightarrow \text{0x83} \rightarrow \text{0x72} \rightarrow \text{0x00} \rightarrow)$$
$$\text{at clock cycle 32: } (\text{0xFF} \rightarrow \text{0x5A} \rightarrow \text{0x94} \rightarrow \text{0x7C} \rightarrow \text{0x72} \rightarrow)$$

Any data passing through Device #2 has been toggled. For instance, NOT(0x8D) is 0x72 (at clock 16), NOT(0x7C) is 0x83 (at clock 24), and NOT(0X6B) = 0x94 (at clock 32).

A2

(Part 1)

1. The period of a PWM signal is given by:

$$\text{Period} = \frac{1}{\text{Frequency}} = \frac{1}{1000} = 1000\,\mu s$$

2. The value to be loaded into the timer's period register (TA0CCR0 value) is calculated as:

$$\text{TA0CCR0} = \frac{\text{Clock Frequency}}{\text{Prescaler} \times \text{PWM Frequency}} - 1$$

$$\text{TA0CCR0} = \frac{16,000,000}{64 \times 1,000} = 250$$

(Part 2) Given the TA0CCR0 value of 249:

$$\text{TA0CCR1} = \text{Duty Cycle} \times (\text{TA0CCR0})$$

1. For 50% duty cycle:

$$\text{Value} = 0.50 \times (249) \approx 125$$

2. For 75% duty cycle:

$$\text{Value} = 0.75 \times (249) \approx 188$$

(Part 3)

1. For an 8-bit timer (TA0CCR0 = 255), the minimum and maximum PWM frequencies are:

$$\text{Minimum Frequency} = \frac{\text{Clock Frequency}}{\text{Prescaler} \times (\text{TA0CCR0} + 1)}$$

$$\text{Minimum Frequency} = \frac{16,000,000}{64 \times 256} \approx 976.56\,\text{Hz}$$

$$\text{Maximum Frequency} = \frac{\text{Clock Frequency}}{\text{Prescaler} \times 2} = \frac{16,000,000}{64 \times 2} = 125,000\,\text{Hz}$$

2. The resolution of the duty cycle is:

$$\text{Resolution} = \frac{1}{\text{TA0CCR0} + 1} \times 100 = \frac{1}{256} \times 100 = 0.39\%$$

(Part 4)

1. The maximum TA0CCR0 value for a 16-bit timer is:

$$2^{16} - 1 = 65,535$$

2. The minimum achievable PWM frequency is:

$$\text{Minimum Frequency} = \frac{\text{Clock Frequency}}{\text{Prescaler} \times (\text{TA0CCR0} + 1)}$$

$$\text{Minimum Frequency} = \frac{16,000,000}{64 \times 65,536} \approx 3.81\,\text{Hz}$$

A3

(Part 1)

- **Press 1:** At Time = 0 seconds, LED turns ON from 0 to 5 seconds.
- **Press 2:** At Time = 3 seconds, the timer resets. LED stays ON from 3 to 8 seconds.
- **Press 3:** At Time = 8 seconds, LED turns ON from 8 to 13 seconds.
- **Press 4:** At Time = 12 seconds, the timer resets. LED stays ON from 12 to 17 seconds.

Total Time LED is ON: From 0 to 17 seconds = **17 seconds**.

(Part 2) The only time a button press has no effect is when it occurs *outside the 5-second window*. In the given sequence, all button presses reset the timer, so no presses are ignored.

(Part 3) If the button is pressed every 4 seconds starting at Time = 0 seconds:

- Each press resets the 5-second timer before it completes.
- Therefore, the LED remains ON indefinitely, as the timer never reaches 5 seconds.

Answer: The LED stays ON **continuously**.