Puzzle 1: The LED Blink

Scenario: You have a microcontroller with a GPIO pin connected to an LED. The goal is to write a program that makes the LED blink with a specific pattern.

Challenge: Write the code for an embedded system (assuming a generic microcontroller with C/C++ syntax) that will make the LED blink in the following pattern:

- **On** for 500 milliseconds
- **Off** for 500 milliseconds
- **On** for 200 milliseconds
- **Off** for 200 milliseconds
- **On** for 500 milliseconds
- **Off** for 500 milliseconds
- Repeat the pattern indefinitely

Solution Hint: You will need to use delay functions and a loop to create the timing and pattern.

Puzzle 2: Temperature Monitoring

Scenario: You have an embedded system with a temperature sensor and a display. The system should display the current temperature and trigger an alert if the temperature goes above a threshold.

Challenge: Design a simple algorithm to monitor the temperature and control an alert system. The requirements are:

1. Continuously read the temperature from the sensor.
2. Display the temperature on the screen.
3. If the temperature exceeds 75°C, activate an alert (e.g., turn on an LED or sound a buzzer).

Solution Hint: Use a loop to read the temperature, check if it exceeds the threshold, and then update the display and activate the alert as needed.

Puzzle 3: UART Communication

Scenario: You are designing a system that uses UART for serial communication between two devices. You need to implement a protocol to ensure data integrity.

Challenge: Create a simple protocol that includes:

1. A start byte (0xAA).
2. A checksum byte that ensures data integrity.
3. A stop byte (0xFF).

Design an algorithm that:

1. Encodes a message with a start byte, data bytes, checksum, and stop byte.
2. Decodes the message, verifies the checksum, and processes the data.

Solution Hint: Use basic checksum algorithms like XOR or simple addition, and implement error checking to ensure valid messages are processed.

Puzzle 4: Timer-Based Events

Scenario: You have a microcontroller with a timer that can be set to generate interrupts at regular intervals.

Challenge: Implement a system where the timer generates an interrupt every 100 milliseconds. On each interrupt, the system should:

1. Increment a counter.
2. Toggle an LED state (from ON to OFF or OFF to ON).

Solution Hint: Configure the timer to generate interrupts, write an interrupt service routine (ISR) that increments the counter and toggles the LED, and ensure that the timer is correctly initialized.

Puzzle 5: Power Management

Scenario: You have an embedded system running on a battery, and you need to optimize power consumption.

Challenge: Design a power management strategy that includes:

1. Entering a low-power sleep mode when the system is idle.
2. Waking up the system on a specific event (e.g., a button press).

Solution Hint: Use sleep modes provided by the microcontroller and configure wake-up sources. Ensure that the transition between active and sleep modes is handled efficiently.