FOR BATCH2

Problem Definition: Path Finder Vehicle Using ARM Processor

1. Project Overview:

Design and develop a path-finder vehicle controlled by an ARM processor. The vehicle should be capable of navigating through a predefined path using sensor inputs and real-time decision making. The system should employ interrupt-based processing for efficient handling of sensor data and obstacle detection. Additionally, the vehicle should feature a display unit to show real-time direction, speed, and elapsed time since the start of the journey.

2. Objectives:

- **Navigation and Pathfinding:** Implement an algorithm to enable the vehicle to follow a predefined path, avoiding obstacles and making necessary turns.
- **Sensor Integration:** Use sensors (e.g., IR, ultrasonic) to detect obstacles and gather environmental data.
- **Interrupt Handling:** Employ interrupt-based processing to handle sensor inputs and control vehicle movements promptly.
- **Display Unit:** Integrate an LCD or LED display to show real-time navigation data (e.g., direction, speed) and elapsed time.
- **Timing Mechanism:** Implement a timing mechanism to record and display the elapsed time since the start of the navigation.

3. Functional Requirements:

- **Vehicle Control:**
  - The vehicle should move forward, backward, and make left/right turns.
  - It should stop or change direction upon detecting obstacles.
- **Sensor System:**
  - The vehicle should use multiple sensors to detect obstacles and edges of the path.
  - Sensor data should trigger interrupts for immediate processing.
- **Interrupt Handling:**
  - Use external interrupts for sensor data processing.
  - Ensure timely and prioritized handling of sensor inputs to avoid collisions.
- **Display Unit:**
  - The display should show current direction, speed, and elapsed time.
  - The time should be updated in real-time.
- **Timing Mechanism:**
  - Implement a timer to record the elapsed time since the vehicle started moving.
  - Display the elapsed time on the display unit.

4. Technical Requirements:

- **Microcontroller/Processor:**
  - ARM Processor (e.g., ARM Cortex-M series).
- **Sensors:**
  - Infrared (IR) sensors, Ultrasonic sensors.

- **Display:**
  - LCD or LED display unit.
- **Power Supply:**
  - Battery pack or an external power source suitable for the chosen microcontroller and sensors.
- **Software:**
  - Bare metal C/C++ [***Usage of in-built libraries is strictly not allowed***]
  - Or, Assembly for ARM processor.

5. Constraints:

- The vehicle must operate within a predefined path and avoid collisions.
- The system should handle interrupts efficiently to ensure real-time processing.
- The display unit should provide clear and accurate information.

6. Evaluation Criteria:

- **Functionality:** Project's ability to perform successfully.
- **Hardware such as Processor architecture, features, and technical know-how.**
- **Software programming skills**
- **Regular attendance in the project lab and interaction with the faculty.**
- **Project diary**
- **Synopsis**
- **Final plagiarism-checked report**

7. Potential Challenges:

- Ensuring timely interrupt handling without missing critical sensor data.
- Calibrating sensors for accurate obstacle detection.
- Synchronizing the display updates with the vehicle's movement and sensor readings.

By meeting these requirements and overcoming the potential challenges, the project will result in a functional path finder vehicle that efficiently uses an ARM processor, leverages interrupt-based processing and provides real-time navigation data and elapsed time on a display unit.

Probable questions that can be asked during the synopsis presentation.

1. **Which microcontroller of the ARM family will you choose for the home automation system, and why? What are the key factors influencing your decision?**
2. **How will interrupts be used to handle sensor data and user inputs efficiently? Provide examples of specific interrupts you plan to implement.**
3. **What communication modules will you integrate into the system, and how will you handle communication using interrupts?**
4. **Describe your approach to implementing timer interrupts. What tasks will they handle, and how will they improve system performance?**
5. **How will you ensure that the system remains responsive and reliable, especially in handling real-time events and maintaining user interaction?**
6. **What are your strategies for testing and debugging the system, particularly focusing on interrupt-driven features?**

7. **What documentation will be provided to support the installation and use of the home automation system? How will it aid users and future maintenance?**