**Designing Capstone Projects: Group-1**

**Remote Flow Monitoring in Chemical Processing Plants**

|  |  |
| --- | --- |
| Stage -1 : Brainstorming | In chemical and oil processing plants, maintaining accurate and continuous monitoring of fluid flow is vital for ensuring safety, operational efficiency, and preventing costly equipment failures. Traditional monitoring systems often rely on manual checks or isolated instrumentation, which can delay anomaly detection and maintenance responses. This project proposes a **remote flow monitoring system** integrating IoT sensors, cloud communication, and machine intelligence for real-time anomaly detection and predictive maintenance.  The **idea layout** presented here is a comprehensive blueprint for developing this system, detailing the hardware, communication, software modules, data flow, analytics, user interfaces, and expected outcomes.  Accurate fluid flow monitoring in chemical/oil processing is essential for safety and efficiency. Flow sensor data is collected and sent to the cloud via MQTT. MI models detect anomalies such as clogs or pump malfunctions. Data analytics enables trend analysis, flow rate forecasting, and real-time alerting for deviations. Interactive dashboards help plant managers visualize performance metrics and make informed decisions for preventive maintenance and process optimization. |
| Stage -2 :  Idea Posting |  |
| Stage -3 :  Customer Mapping | **1. Primary Customer:**  **Plant Managers / Operations Engineers**   * **Goals:**   + Ensure optimal flow across all pipelines   + Detect clogs or pump issues early   + Maintain high production uptime * **Pain Points:**   + Lack of real-time visibility into flow anomalies   + Delayed detection of equipment failure   + Reactive maintenance leads to production loss * **What They Get:**   + Real-time alerts for abnormal flow   + Dashboard visualization for easy decision-making   + Historical data for audit, forecasting, and planning   **2. Maintenance Teams / Technicians**   * **Goals:**   + Fix flow-related issues before failures occur   + Perform scheduled preventive maintenance   + Reduce downtime and emergency repairs * **Pain Points:**   + No early warning system for component degradation   + Relying on manual inspections * **What They Get:**   + ML-based anomaly predictions   + Alerts for suspected clogs or pump performance drops   + Access to trend data for targeted maintenance   **3. Data Analysts / Process Engineers**   * **Goals:**   + Analyze flow behavior under varying conditions   + Forecast demand/load on fluid systems   + Optimize chemical mixing, pumping schedules * **Pain Points:**   + Disconnected data from sensors   + No unified dataset for ML or statistics * **What They Get:**   + Structured flow rate data via MQTT   + Clean time-series logs for ML model training   + Ability to derive insights for process improvement   **4. Plant Executives / Management**   * **Goals:**   + Increase ROI through automation and efficiency   + Minimize unplanned shutdowns   + Justify investment in smart technologies * **Pain Points:**   + High maintenance cost   + Unpredictable process failures * **What They Get:**   + Dashboard KPIs on flow performance   + Reports showing reduced downtime   + ML-driven preventive maintenance planning   **5. System Integrators / Automation Consultants**   * **Goals:**   + Provide plug-and-play monitoring solutions   + Help clients with digital transformation   + Scale systems across plants/sites * **Pain Points:**   + Complex sensor setups   + Incompatibility across platforms * **What They Get:**   + Lightweight, MQTT-based flow sensor system   + Easy integration with Node-RED, Grafana, or cloud IoT   + A starting point for larger IIoT (Industrial IoT) rollouts |
| Stage – 4: Idea Layout | **Start**   1. **Initialize Hardware**    * Setup flow sensor    * Setup ESP32 microcontroller    * Connect to MQTT broker 2. **Read Flow Sensor Data**    * ESP32 reads pulse signals    * Convert pulses to flow rate 3. **Publish Flow Data to MQTT Broker** 4. **MQTT Broker Receives Data** 5. **Backend Python Subscriber Receives Data** 6. **Store Data in Database** 7. **Run Anomaly Detection Algorithm**    * Check for sudden drops/spikes    * Use statistical or ML model 8. **Is Anomaly Detected?**    * Yes → Go to step 10    * No → Go to step 12 9. **Generate Alert**    * Send SMS/Email/Push notification    * Flag anomaly on dashboard 10. **Log Anomaly Event** 11. **Run Forecasting Model**     * Predict future flow rate trends 12. **Update Dashboard**     * Real-time data visualization     * Display anomalies and forecasts 13. **Check for Network Issues**     * If network lost, buffer data and retry 14. **Loop Back to Step 3 (Continuous Monitoring)**   **End** |
| Stage – 5: Reflection |  |
| Stage – 6: Designs of Modules |  |
| Stage – 7: Resource Identification | Software :    Hardware Cost: |
| Stage – 8: Planning |  |
| Stage -9: Re-Design | * Conducted a detailed review of the initial project checklist after the first implementation phase. * Identified inconsistencies in simulating the flow sensor readings. * Noted the absence of MQTT protocol integration in the simulation environment. * Recognized the need for improved alert mechanisms for high flow detection. * Revised the system design to refine ADC value processing for more accurate sensor simulation. * Incorporated LED-based visual alerts to indicate abnormal flow conditions. * Substituted MQTT communication with HTTP-based ThingSpeak data transmission for better compatibility with Wokwi simulation. * Enhanced overall system reliability and alignment with project objectives through these modifications. |
| Stage -10: Execution Framework | Inserting image... |
| Stage -11: Mircomodules |  |