AI and Edge Computing – Batch 05

CAPSTONE PROJECT

PHASE 1:

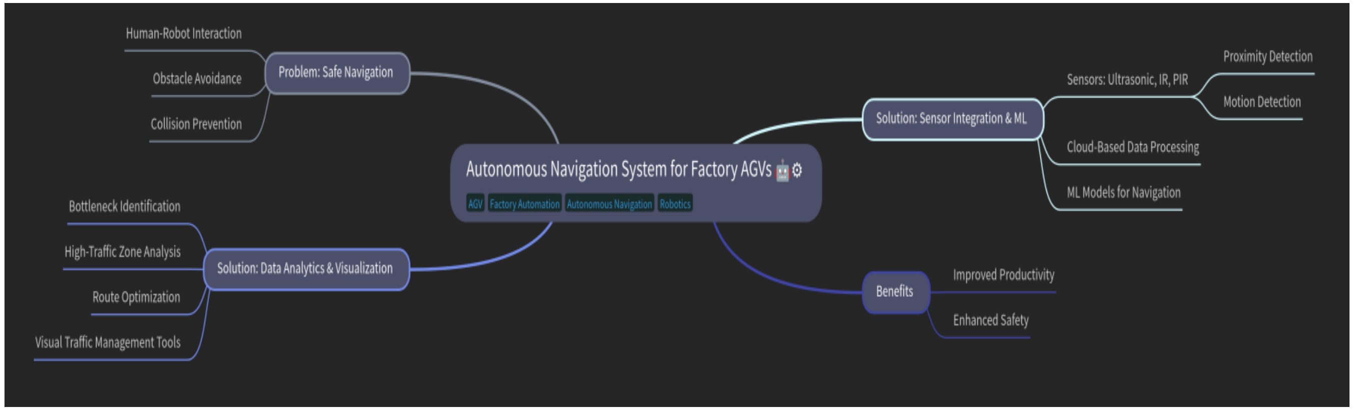
Stage I:

Brainstorming	Tool	Outcome	Metrics/Rubric
Autonomous Navigation System for Factory Floor Robots (AGVs) **Sensors:** Ultrasonic (1), IR (1), PIR (1) **Problem Description:** AGVs must navigate safely around humans and obstacles on a busy factory floor. Sensor data (Ultrasonic, IR, PIR) is uploaded to a cloud ML platform that drives navigation decisions. Data analytics identify movement bottlenecks, high-traffic zones, and route efficiency. Visual dashboards support traffic optimization, reduce collision risk, and improve	**Challenges:** Dynamic appearance of humans and forklifts on the same floor. Real-time obstacle detection when multiple AGVs converge. Low-latency path planning vs. cloud round-trip delays. **Opportunities:** Use historical traffic data to pre-emptively reroute AGVs and avoid future congestion. Leverage cloud ML to continuously improve object-recognition. Predictive maintenance: flag AGVs that deviate from optimal paths as early indicators of wheel/motor wear.	**Refined Project Statement:** A Smart AGV Navigation System that can: 1. Fuse Ultrasonic + IR + PIR sensor data on-board to detect humans, forklifts, and static obstacles in real time. 2. Execute alternate path planning locally to avoid newly detected obstacles with low latency. 3. Stream only critical events (AGV position + obstacle tag) to a cloud ML platform. 4. Use historical logs in the cloud to identify traffic bottleneck zones and update local AGV route preferences (e.g., avoid “High-density Zone A” during peak hours). 5. Apply zone-based dynamic speed limits: slow down in human-traffic corridors.	1. The problem is restated with deep insight (combining real-time detection, latency constraints, and continuous ML refinement). Outputs include precise details: which sensors, which path-planning algorithm, what data is streamed, and how cloud analytics feed back into local decisions. 2. The problem is restated with good insight, but some outputs are still slightly vague. 3. The problem is high-level, and outputs remain vague.

overall productivity and safety.	Zone-based dynamic speed adjustments. **Idea: Alternate Path Planning** If the AGV spots an obstacle, its onboard controller quickly picks a new route from the local map and sends only essential event data to the cloud. **Type:** Industrial robotics / Automation project. **Domain:** Factory automation / Smart manufacturing / Robotics. **Stakeholders:** Factory managers, AGV operators, Safety officers, Maintenance team, IT/OT integrators. **Technologies:** Ultrasonic sensors, IR sensors, PIR sensors, Cloud platform, ML algorithms, Visualization tools.	6. Provide a real-time dashboard showing AGV heatmaps, current traffic flow, and predicted congestion.	

Stage II:

Mindmap:



Abstract:

The proposed model focuses on developing an Autonomous Navigation System for Automated Guided Vehicles (AGVs) within factory environments. The core challenge addressed is ensuring safe navigation, particularly through efficient human-robot interaction, obstacle avoidance, and collision prevention. The model integrates multi-sensor technology, including ultrasonic, infrared (IR), and passive infrared (PIR) sensors, for real-time proximity and motion detection. Sensor data is processed through cloud-based infrastructure, enabling machine learning (ML) models to guide intelligent path planning and decision-making.

In parallel, the system incorporates data analytics and visualization tools to monitor and optimize AGV performance. This includes identifying bottlenecks, analyzing high-traffic zones, and implementing route optimization strategies supported by visual traffic management interfaces. Together, these solutions significantly enhance factory automation by improving operational productivity and ensuring worker safety in dynamic industrial settings.

PHASE 2:

Stage III:

Factory Floor Supervisor: Understand pain points with daily AGV operations and tracking.

Operations Manager: Explore efficiency, delay issues, and need for optimization tools.

Data Analyst: Understand data needs, visualization requirements, and reporting challenges.

Safety/Compliance Officer: Address safety concerns, AGV-human interactions, and incident prevention.

Maintenance Team: Understand diagnostic needs and difficulties in AGV repair/troubleshooting.

Questionnaire:

Instructions:
Please answer the following questions based on your experience with AGVs in your facility. All responses will remain confidential and used only for academic research purposes.

Please indicate your role in the factory environment.

- ☐ Factory Floor Supervisor
- ☐ Operations Manager
- ☐ Data Analyst
- ☐ Safety/Compliance Officer
- ☐ Maintenance Team
- ☐ Other (please specify): ________________________

Factory Floor Supervisor

1. How do you currently monitor AGV movement across the factory floor?
    - ➢ Camera feeds and manual observation are used to track AGV movement.

2. Have there been any recent incidents involving AGVs and workers?
   - Yes, minor near-misses have occurred due to blind spots.
3. How easily can you identify an AGV that's stuck or underperforming?
   - It is difficult without a central dashboard to detect stuck AGVs.
4. How often do you intervene manually in AGV operation?
   - Manual intervention is required 2–3 times daily for rerouting or resets.

Operations Manager

1. How do AGVs currently impact your overall production flow?
   - AGVs increase efficiency but occasionally cause bottlenecks.
2. Do you experience regular delays due to AGV routing inefficiencies?
   - Yes, routing inefficiencies often delay material delivery.
3. How do you assess route performance or congestion points presently?
   - Currently assessed by manual tracking and worker feedback.
4. Would predictive routing or alternate path suggestions help in your role?
   - Yes, predictive routing would greatly improve flow planning.

Data Analyst

1. What types of data do you currently collect from AGVs?
   - We collect timestamped location data, usage logs, and delivery status.
2. Which metrics are most critical to track for factory logistics?
   - Idle time, delivery time, and route overlap are most critical.
3. Do you face issues with accessing or analyzing AGV movement data?
   - Yes, real-time data is fragmented and hard to visualize.
4. What analytics tools or platforms do you currently use?
   - Excel and basic BI dashboards are currently used.
5. Would heatmaps or congestion visualizations be beneficial to you?
   - Absolutely, heatmaps would help identify congestion hotspots quickly.

Safety/Compliance Officer

1. Have you recorded any AGV-related safety incidents or near-misses?

- Yes, we’ve recorded a few close encounters in shared zones.

2. How do you currently monitor AGV proximity to human workers?
    - Monitoring is done visually and through PIR sensors in high-traffic areas.
3. Would automated alerts for human detection or collision risk be useful?
    - Yes, real-time alerts would reduce response time and improve safety.
4. What features would help you feel confident about AGV safety compliance?
    - Clear zones, proximity sensors, and automated braking features help ensure compliance.

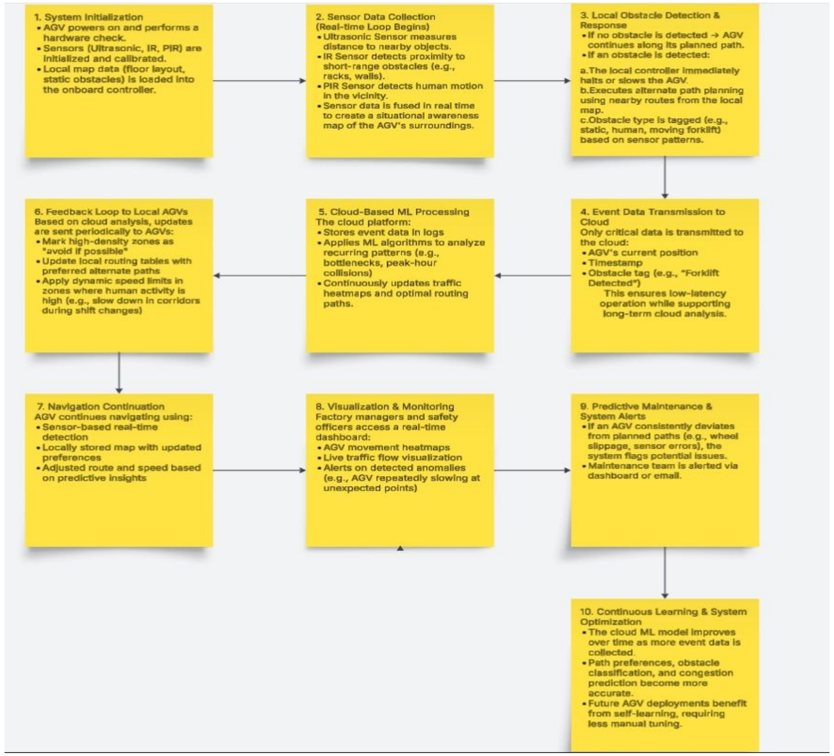
Maintenance Team

1. What are the most common AGV issues you deal with?
    - Battery issues and sensor failures are the most common.
2. How do you currently diagnose AGV faults or downtimes?
    - Diagnosis is done through manual inspection and system logs.
3. How often do issues go unnoticed until they cause bigger delays?
    - Yes, **minor issues often escalate due to delayed detection**.

PHASE 3:

Stage IV:

Sticky Notes:



PHASE IV:

Stage V:

Project Overview:

The Smart AGV (Automated Guided Vehicle) Navigation System is an ongoing project aimed at improving autonomous navigation within dynamic factory environments. The system is designed to enhance safety, efficiency, and decision-making for AGVs operating in shared spaces with humans, forklifts, and other moving elements.

The core idea revolves around integrating multiple onboard sensors—Ultrasonic, Infrared (IR), and PIR—to detect obstacles in real time. These detections enable the AGV to make quick local decisions and reroute itself without relying entirely on cloud commands.

Simultaneously, critical data such as AGV location, obstacle events, and route history is transmitted to a cloud-based machine learning platform.

This cloud system will analyze historical traffic patterns to identify high-congestion zones, enable predictive maintenance, and optimize route planning. An administrative dashboard will provide real-time visibility into AGV performance, heatmaps, and alerts, supporting data-driven operational decisions.

The project is currently in the development phase, with key components such as sensor integration, route mapping, and dashboard planning in progress. Once complete, this system is expected to significantly reduce collision risks, improve workflow efficiency, and offer a scalable solution for smart factory automation.

Cost estimation of the project:

Item	Quantity	Unit Price (Approx.)	Total Cost (INR)
Ultrasonic Sensor (HC-SR04)	1	₹60	₹60
IR Sensor Module	1	₹25	₹25
PIR Motion Sensor (HC-SR501)	1	₹50	₹50
Microcontroller (Raspberry Pi Pico)	1	₹400	₹400
Motor Driver Module (L298N)	1	₹150	₹150
DC Motors with wheels	2	₹150	₹300
Chassis (basic 2-wheel robot frame)	1	₹200	₹200
Battery (Li-ion pack or 12V battery)	1	₹300	₹300
Wires, Connectors, Breadboard, etc. ESP32 (wifi module)	- 1	₹150 ₹100	₹150 ₹100
Dashboard/Server (cloud-based)	-	Free / ₹200 (if needed)	₹0–₹200

Total estimate cost will be approximately ₹1800 to ₹2000

Path detection mechamism of the project:

Mechanism Overview:

1. Sensor Fusion (Input Collection):
    a. The AGV is equipped with:
        i. Ultrasonic Sensor – Detects nearby obstacles (static or moving).
        ii. Infrared (IR) Sensor – Detects object edges and short-range obstacles.
        iii. PIR Sensor – Detects human motion and thermal presence.
2. Obstacle Detection Logic:
    a. If any sensor detects an object within a predefined safety radius, the system flags a path block event.
    b. The AGV pauses immediately and triggers the local path planning module.
3. Alternate Path Planning (Onboard Logic):
    a. The AGV uses a local map (pre-loaded or updated periodically from the cloud) to identify available alternate paths.
    b. A pathfinding algorithm (e.g., A*, Dijkstra, or BFS) computes the next shortest viable route avoiding the blocked node/zone.
    c. Only essential event data (e.g., "Obstacle at Node C4") is sent to the cloud to update global traffic models.
4. Cloud Feedback (Learning & Optimization):
    a. The cloud logs all such events.
    b. Over time, ML models analyze bottleneck zones, predict congestion, and push updated routing preferences to AGVs.

Step by step flow:

AGV is in motion → Obstacle detected by sensors → Trigger emergency halt + event logging → Run onboard path finding → Select new route → Resume motion → Update cloud with event and selected route

Stage VI:

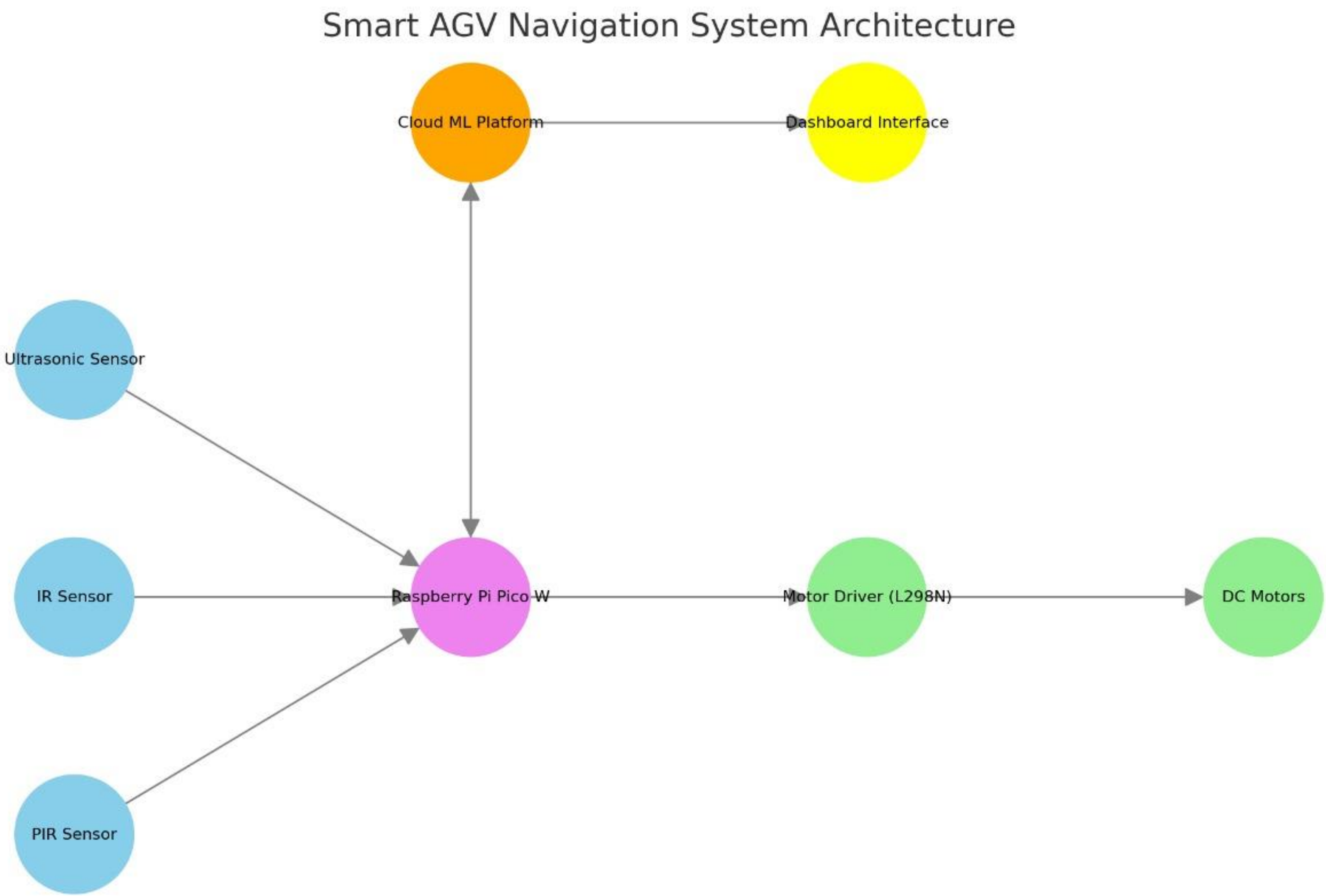
Conceptual Design:

The AGV moves autonomously across the factory floor using inputs from onboard sensors. Obstacle and human detection is done using Ultrasonic, IR, and PIR sensors. The Raspberry Pi Pico W processes sensor data in real-time, makes routing decisions locally, and communicates critical events to a cloud server. The cloud platform performs advanced analytics and improves navigation over time. A dashboard visualizes AGV behavior and factory traffic.

Component-Level Architecture:

Component	Function
Ultrasonic Sensor	Detects proximity to static/moving obstacles
IR Sensor	Detects object edges and nearby obstructions
PIR Sensor	Detects human movement/thermal signature
Raspberry Pi Pico	Sensor fusion, local path planning, event handling
Motor Driver (L298N)	Drives the motors according to controller output
DC Motors	Provide mobility to the AGV
Wi-Fi Module	Sends critical event data to the cloud
Cloud Platform	ML analytics for traffic prediction & rerouting logic
Dashboard UI	Visual representation of AGV heatmaps and alerts
Battery Pack	Powers entire system

Block diagram description:


Smart AGV Navigation System Architecture
Cloud ML Platform
Dashboard Interface
Ultrasonic Sensor
IR Sensor
PIR Sensor
Raspberry Pi Pico W
Motor Driver (L298N)
DC Motors

Interfaces and standards:

1. Sensors:

Sensor	Interface	Communication Standard/Type
Ultrasonic (HC-SR04)	Digital GPIO + Pulse Timing	Custom Timing-based Protocol
IR Sensor	Digital GPIO	TTL Logic
PIR Sensor	Digital GPIO	TTL Logic

2. Microcontroller:

Feature	Interface	Standard/Protocol
Wi-Fi Communication	Built-in Wi-Fi	IEEE 802.11 b/g/n (2.4 GHz)
Sensor Input	GPIO	3.3V TTL compatible
Motor Driver Control	GPIO (PWM)	PWM Signal Control (TTL)
Cloud Communication	TCP/IP stack	HTTPS / MQTT (standard IoT protocols)
Programming Interface	USB (or UART)	UF2 Firmware + MicroPython/C SDK

3. Motion control interfaces:

Component	Interface	Standard
L298N Motor Driver	GPIO (IN1–IN4)	TTL logic, PWM speed control
DC Motors	Driven by L298N	Motor control standards (DC, brushed)

4. Power interfaces:

Component	Power Supply	Standard
Raspberry Pi Pico W	5V via USB or Vin	5V logic input (with onboard LDO)
Sensors (IR, PIR, Ultrasonic)	3.3V–5V	Regulated power from Pico/Battery
Motor Driver + Motors	6V–12V Battery	DC motor power standards

5. Cloud connectivity and data exchange:

Communication	Protocol	Standard/Technology
Pico W to Cloud	HTTPS / MQTT	IoT protocols via TCP/IP
Data Format	JSON / CSV	Interoperable data exchange formats

Stage VII:

Revised Budget:

Item	Quantity	Unit Price (Approx.)	Total Cost (INR)
Ultrasonic Sensor (HC-SR04)	1	₹60	₹60
IR Sensor Module	1	₹25	₹25
PIR Motion Sensor (HC-SR501)	1	₹50	₹50
Microcontroller (Raspberry Pi Pico W)	1	₹500	₹500
Battery	1	₹100	₹100
Wires, Connectors, Breadboard, etc.	-	₹150	₹150
Dashboard/Server (cloud-based)	-	-	-

Revised Cost of Resources- ₹885

Revised Software Needs:

Software	Purpose
Thonny IDE	Programming the Raspberry Pi Pico W
ThingSpeak	Data logging and cloud storage
MQTT/HTTP	Protocol for sending sensor data
Python Librarie	For Wi-Fi, sensor interfacing & HTTP requests
Visualization Tools	ThingSpeak and custom dashboard for heatmaps, event logging

Cloud ML Engine	For future implementation of traffic prediction, route optimization

Stage VIII:

Planning:

1. Project Objective:

To design and implement a smart, sensor-integrated AGV system capable of real-time obstacle and human detection, dynamic local path planning, and cloud-based event analytics for predictive optimization.

2. Development Phases

1) Phase 1: Sensor interfacing and onboard path planning (using Raspberry Pi Pico W).
2) Phase 2: Critical event transmission and cloud storage using ThingSpeak.
3) Phase 3: Visualization and heatmap generation using cloud dashboard tools.
4) Phase 4: Future integration of machine learning models for traffic prediction and route optimization.

3. Software Stack

1) Thonny IDE – for coding and flashing the Pico W.
2) Python libraries – for sensor input, Wi-Fi connectivity, and cloud communication.
3) ThingSpeak – for data logging, cloud storage, and visual dashboards.
4) MQTT/HTTP – for lightweight, real-time communication between the AGV and cloud.
5) Cloud ML Engine (future phase) – for predictive routing and bottleneck detection.

Constraints and assumptions:

1. Hardware Limitations
    a. Raspberry Pi Pico W has limited RAM/processing power compared to full Linux boards.
    b. Sensor readings can be affected by factory noise, IR reflections, or lighting.
2. Real-Time Processing
    a. Latency must be minimal for path redirection; hence, only critical events can be sent to the cloud.
    b. Dependence on Wi-Fi stability may cause intermittent cloud sync failures.
3. Power Supply
    a. All components must run efficiently from a shared Li-ion or 12V battery.
    b. Voltage regulation must support both 3.3V logic and motor driver demands (6V–12V).
4. Scalability
    a. Current setup is for a single AGV prototype; scaling to multiple units would require dynamic ID tracking and decentralized conflict resolution.
5. Cost
    a. Budget capped at ₹2000; more advanced features like additional sensors, higher-spec cloud services, or AI accelerators are out of scope for this phase.
6. Environment
    a. Factory floor is pre-mapped and has stable Wi-Fi coverage.
    b. Obstacles and human activity follow regular patterns that can be learned over time.
7. Hardware Availability
    a. All components (sensors, microcontroller, battery) are available off-the-shelf and compatible without custom fabrication.
8. Data Transmission
    b. Event-based data transmission via MQTT or HTTP is sufficient to represent the AGV's status and doesn't overwhelm network bandwidth.

Stage IX:

Redesign:

The system integrates three onboard sensors with a Raspberry Pi Pico W microcontroller to detect obstacles, human presence, and object edges. It performs local decision-making and selectively sends important events to a cloud dashboard (ThingSpeak). This approach ensures low-latency reactions and uses the cloud for analytics and visualization. For the revised design, the functional modules are as follows -

1. Onboard Sensor Fusion & Obstacle Detection

- All 3 sensors are polled via GPIO.
- Threshold-based detection triggers emergency stop and path planning.

2. Local Path Planning

- When an obstacle is detected:
    - AGV halts.
    - Onboard logic runs A* or BFS algorithm on local map.
    - Chooses new shortest route.
    - Resumes movement.

3. Cloud Integration

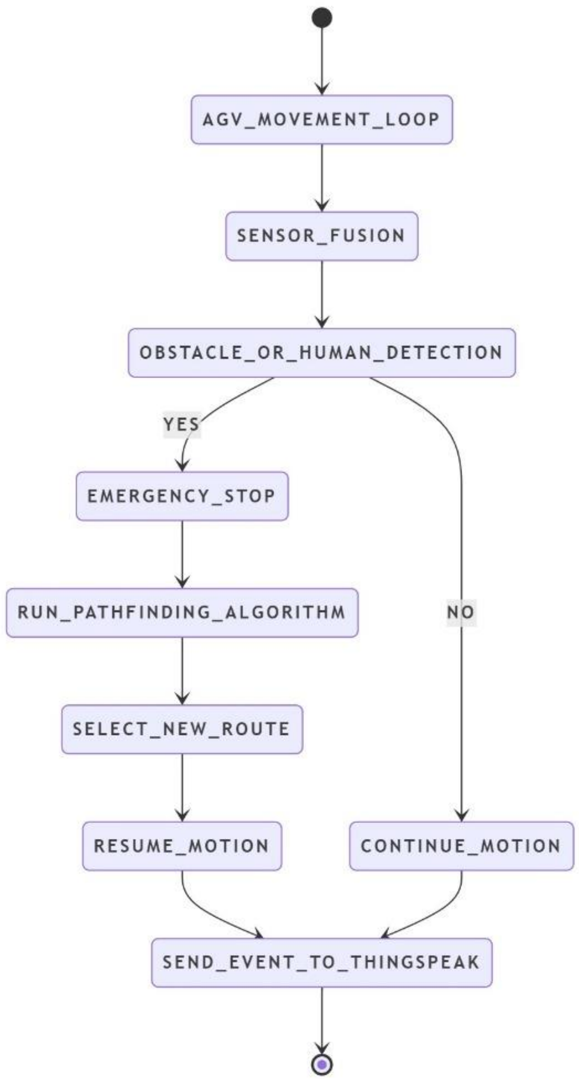
- Only critical events like:
    - "Obstacle Detected at X"
    - "AGV rerouted to Y"
- are sent to ThingSpeak via MQTT/HTTP.
- Events are visualized as heatmaps & logs.

4. Visualization Dashboard

- Real-time data shown:

    - AGV location history
    - Obstacle zones
    - Congestion zones (future implementation)
- Basic GUI from ThingSpeak or a custom web-based heatmap interface.

Flowchart:


AGV_MOVEMENT_LOOP
SENSOR_FUSION
OBSTACLE_OR_HUMAN_DETECTION
YES
EMERGENCY_STOP
RUN_PATHFINDING_ALGORITHM
SELECT_NEW_ROUTE
RESUME_MOTION
NO
CONTINUE_MOTION
SEND_EVENT_TO_THINGSPEAK

PHASE 5:

Stage X:

Executional framework:

Module	Purpose	Tools & Technologies	Key Metrics
Edge Sensing & Alerts	Read Ultrasonic, IR, PIR sensors Local collision/part-mismatch buzzer & LED alerts	Pico W (GP12–GP15, GP16 MicroPython firmware Buzzer + LED	Alert latency (ms) False-positive rate
Data Ingestion & Health	Publish sensor heartbeats & events to cloud Monitor sensor uptime & errors	MQTT (e.g. Mosquitto) or HTTPS → ThingSpeak/IoT Hub Prometheus + Grafana or BI streaming datasets	Sensor uptime (%) Error-read rate (%)
Real-Time Dashboard	Live view: AGV locations, obstacles, sensor status, part matches	Power BI (or Tableau) WebSockets/MQTT connector ThingSpeak JSON API	Dashboard refresh time % AGVs reporting
Defect Analytics	Historical root-cause analysis of near-misses & mismatches	Power BI time-series & clustering	Near-misses per shift/zone Trend accuracy
Remote Notifications	SMS/app push on critical events	ThingSpeak → Power Automate/IFTTT Email/SMS gateway	Alerts/day Mean time to acknowledge

Stage XI:

Micromodules:

Micromodule Name	Description

1. Sensor Fusion & Detection Module	Gathers and processes input from Ultrasonic, IR, and PIR sensors via GPIO on the Raspberry Pi Pico W. Performs real-time obstacle/human detection using threshold logic.
2. Local Path Planning Module	On detecting an obstacle, this module halts AGV motion and computes a new path using A* or BFS algorithm on a pre-loaded local map. Enables low-latency rerouting onboard.
3. Motor & Motion Control Module	Translates navigation commands into PWM signals to drive motors through the L298N motor driver. Enables smooth start/stop/turn actions and resumes motion after rerouting.
4. Safety & Emergency Override Module	Monitors for collision risks or close-range human activity using PIR/Ultrasonic. Triggers an emergency halt independently of path logic. Acts as a safety failsafe.
5. Cloud Sync & Uplink Module	Publishes only critical events (e.g., "Obstacle at X", "Rerouted to Y") to ThingSpeak using MQTT or HTTP. Operates under bandwidth constraints for minimal data exchange.
6. Event Logging & Visualization Module	Formats and logs AGV events (position, detections, route changes) into ThingSpeak channels. Supports cloud dashboard integration for real-time heatmaps and performance logs.
7. Dashboard & Monitoring Interface	Enables operators to monitor AGV status, zone alerts, and movement trails through ThingSpeak visualizations or a custom web-based dashboard. Aids diagnostics and traffic optimization.
8. ML Optimization & Maintenance Module *(Future Phase)*	Analyzes historical cloud data to identify traffic bottlenecks, optimize routes, and flag predictive maintenance (e.g., repeated slowdowns at same node = motor wear).