SCUOLA DELL'INGEGNERIA INDUSTRIALE E DELL'INFORMAZIONE

COMPUTER SCIENCE AND ENGINEERING

Internet Of Things
Homework

Simone Pio Bottaro - 10774229

Gabriele Lorenzetti – 10730455

Summary

Introduction 1

Requirements 1

Assumptions 1

Hardware Installed 2

Localization 3

Transmission 4

Pseudocode 5

Backend Architecture 7

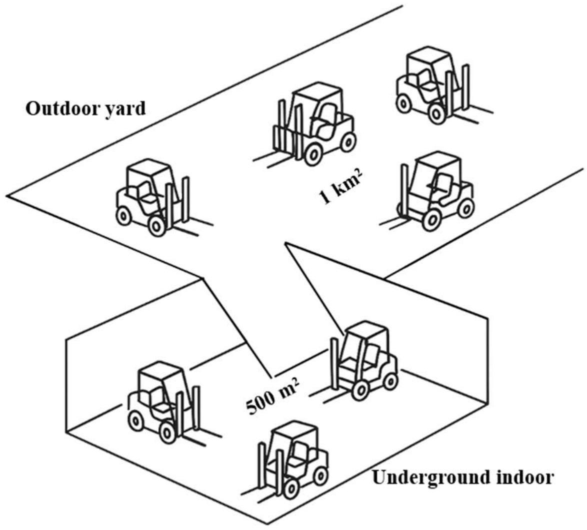
- **Introduction** 7
- **Data Ingestion** 7
- **Data Processing** 8
- **Database** 8
- **Visualization** 8

Introduction

A logistic company operates a warehouse composed of a 500 m^2 **underground** indoor area and a 1km^2 **outdoor yard**. Electric forklifts are used across both zones and return to specific docking stations to recharge.

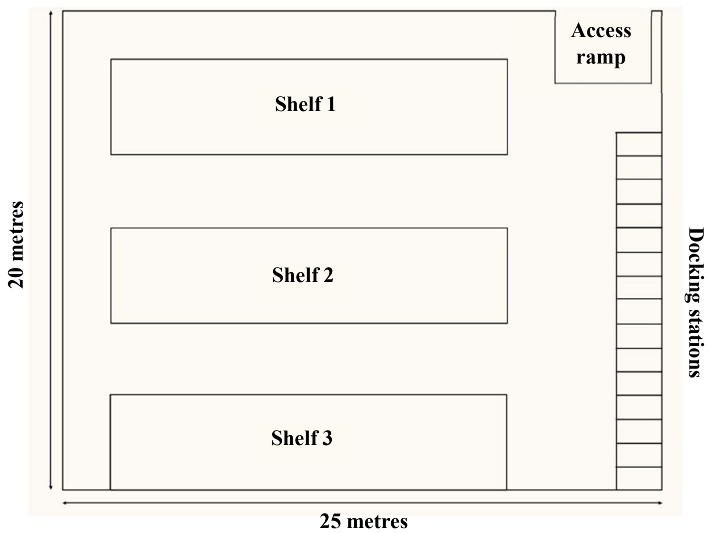
Requirements

Design a low-cost, easy and scalable IoT system to localize forklifts in real time and monitor their status including daily distance travelled, maximum and average speed and impact detection.



Assumptions

- Forklifts are human directed
- Underground area is 25 x 20 metres, level -1
- Outdoor area is 1 x 1 km square
- There are shelves in the internal warehouse
- Docking stations are inside the indoor area
- The data (maximum speed, distance, etc.) are reset at the end of the workday



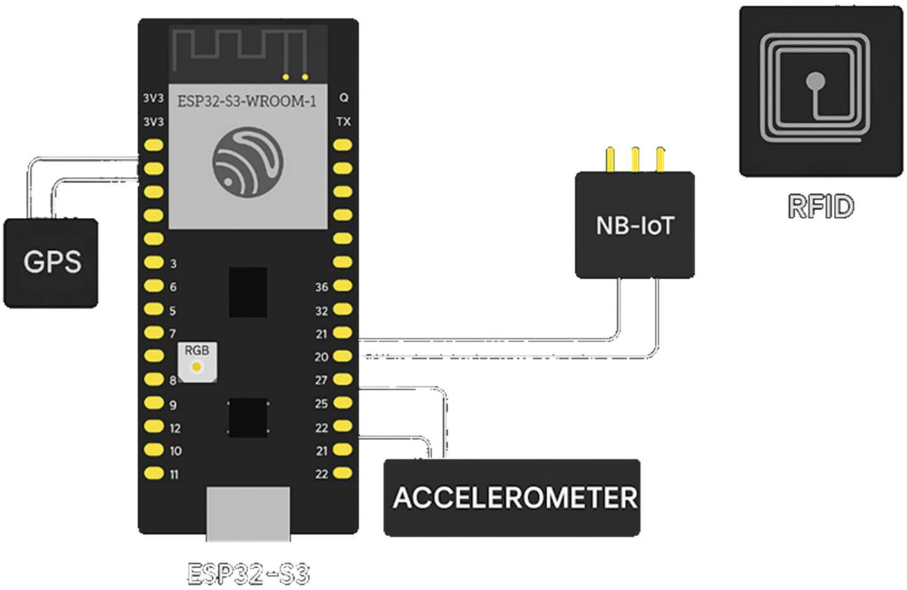
Example of a possible basement layout considered at the design stage

Hardware Installed

To design the required network, ESP-32 S3 boards have been used, connected to several additional components to process the required information. To guarantee optimal localization in both areas of the warehouse, two different tracking systems have been implemented (described below). Here is showed the list of all the components installed for each forklift.

As assumption, we considered that, using a specific circuit to reduce battery voltage level, the Esp-32 with its relative components is alimented by the forklift's battery. The RFID chip has its own battery.

- Esp-32 S3
- NB-IoT module (for communication)
- GPS module (for external localization)
- RFID chip (for internal localization)
- Accelerometer (for speed and impact tracking)



An NB-IoT antenna has been implemented for communication to permit the forklifts to communicate with the backend architecture without any gateway installed across the warehouse, with Band 8 (900 MHz) or Band 20 (800MHz) with a Bandwidth of 180-200 kHz for a better underground penetration.

The NB-IoT technology permits to communicate using special sim boards designed for IoT devices. Allows a ***bi-directional communication*** with a relative low hardware cost of about 4 euros plus a specific sim contract with an operator per unit. This technology can be very useful in the examinate case, allowing the forklifts to communicate their data even when they are in the indoor area without any problems. The bi-directional communication, as we will see below, is another important feature that permits the switch between indoor and outdoor localization, allowing the forklifts to save energy switching off the GPS module when it is not required.

Such an approach, as homogeneous as possible between inside and outside, is relatively easy to implement compared to using other technologies. Moreover, it is easily scalable if new forklifts are added.

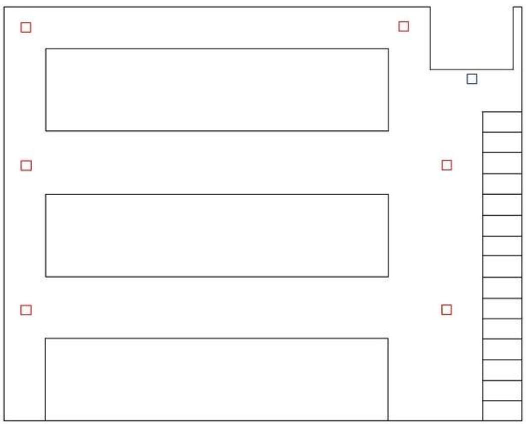
Localization

As mentioned before, the localization logic has been distinguished into two different logics. In the outdoor area, of 1km^2, the simplest way to localize the forklifts is by using GPS module. In this case ***the position of each forklift is sent to the backend from themselves.***

The problem becomes more interesting in the indoor area. Being underground, the GPS module would not be enough to localize forklifts in real time. A solution to this problem is given by **RFID** (Radio Frequency Identification), which is a wireless technology that enables automatic identification and tracking of objects using radio waves. It consists of tags, which store identifying information, and readers that communicate with the tags to retrieve data. In this project, each forklift is equipped with a ***semi-passive RFID tag*** with a ***range of approximately 20 metres***. These tags contain an internal battery that powers the circuitry but rely on external signals from readers to transmit data, allowing for extended range and reliability. **RFID readers installed on the roof** of the warehouse continuously scan the area to **detect and track the position of forklifts** in real-time. ***The position of each forklift is sent from the readers to the backend when they are inside the indoor area.***

To improve location accuracy, **6 readers**[1] are sufficient to maximize range. Moreover, the **backend system processes the RSSI (Received Signal Strength Indicator)** values received from multiple readers. By analysing the signal strength from each tag-reader interaction, the system can estimate the position of each forklift more precisely with triangulation.

The switch between the two tracking logics, indoor and outdoor, is made possible by another ***dedicated RFID reader (blue one)*** placed along the ramp to the outdoor area. When it detects a forklift on the ramp, the reader sends the forklift's ID to the backend, indicating that it is entering or exiting. The backend, thanks to the NB-IoT technology's ability to communicate in both directions, communicates to the specific forklift to change its state by turning the GPS module off or on as needed.



[1] The number of readers may vary depending on the number and position of shelves, which could interfere with the reading of the tags installed on the forklifts. To reduce their number and increase their efficiency, it is very important to orient them in the best possible way.

Transmission

Both the forklifts and the readers communicate with the backend using NB-IoT technology. Packets are sent using MQTT protocol. Since the tracked objects are forklifts, the position information can be transmitted every 5 seconds for both indoor and outdoor tracking. As for the transmission of status data, which includes the average and maximum speed reached, updates can be sent less frequently, even every 15 minutes. Special attention must be given to collision events: in such cases, the ESP detects the impact via the accelerometer and immediately sends an error status to the backend without waiting for the next scheduled transmission interval.

Here are reported some examples of payload transmitted from the different parts of the system:

- **Indoor localization (RFID reader to backend)**


```
{
    "type": "rfid_read",
    "readerId": "WH_RMP_01",       // reader ID
    "tagId": "ABC123456789",        // forklift ID
    "timestamp": "2025-05-22T17:25:00.123Z", // Timestamp
    "rssi": -55,                    // RSSI in dBm.
}
```


- **Outdoor localization (forklift to backend)**


```
{
  "type": "gps_loc",
  "carId": "FORKLIFT_XYZ_123",
  "timestamp": "2025-05-22T17:25:05.456Z",
  "location": {
      "type": "GPS",
      "latitude": 45.4642,
      "longitude": 9.1900,
      "altitude": 120.5,
    }
}
```


- **Status feedback (forklift to backend)**


```
{
    "carId": "FORKLIFT_XYZ_123",
    "timestamp": "2025-05-22T17:25:05.456Z",
    "metrics": {
        "currentSpeedKmH": 15.2,
        "totalDistanceKm": 25.7,
        "maxSpeedKmH": 22.1,
        "averageSpeedKmH": 12.8,
        "batteryLevelPercent": 85
    },
    "event": {
        "impactDetected": false,
        "impactSeverity": null
    },
    "status": {
        "mode": "GPS_ACTIVE"
    }
}
```


- **Status setting (backend to forklift)**


```
{
    "command": "SET_LOCATION_MODE",
    "targetCarId": "FORKLIFT_XYZ_123",
    "mode": "RFID_MODE",               // "GPS_MODE" or "RFID_MODE"
    "timestamp": "2025-05-22T17:26:00.000Z",
    "commandId": "CMD_SW_001_12345"
}
```

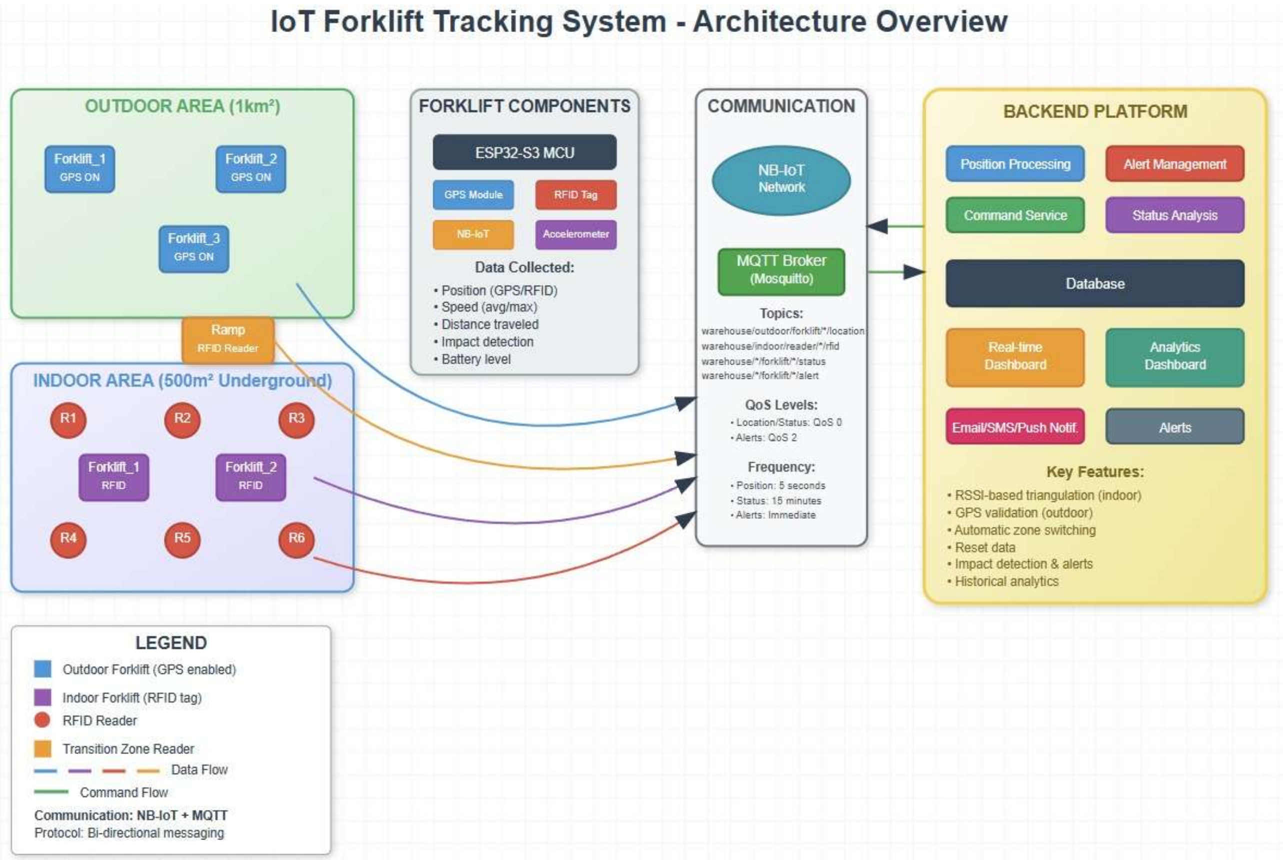
Pseudocode

Here is showed a simple pseudocode that shows the core logic installed in each ESP-32 board installed.

```
declare LocationMode as String
declare forkliftData as Object // Contains values as speed, distance, impact, battery
declare isGpsModuleActive as Boolean
declare impactDetected as Integer
declare AvgSpeed as Float
declare MaximumSpeed as Float

function setup_forklift():
    initialize_system()
    loop_forever()

function initialize_system():
    initialize_NB_IoT_connection()
    initialize_forklift_sensors()
    deactivate_GPS_module()

    LocationMode = "RFID"
    isGpsModuleActive = false
    impactDetected = 0
    AvgSpeed = 0
    MaximumSpeed = 0

function loop_forever():
    loop forever:
        read_sensors()
        detect_impact()
        handle_backend_commands()

        if time_to_send_location():  //every 5sec
            send_location_payload()

        if time_to_send_status_payload():  //every 15min
            send_status_payload()

// --- Sensor & Impact Management ---

function read_sensors():
    read_speed_data()
    read_battery_level()
    calculate_speed_and_distance()

function detect_impact():
    if check_impact():
        set_impact_flag_in_forklift_data(true)
        log("Impact detected!")
        send_location_payload() //Sending location in case of impact
        send_status_payload() //Sending status in case of impact
    else:
        set_impact_flag_in_forklift_data(false)

// --- Payload Handling ---

function send_location_payload():
    if isGpsModuleActive:
        read_GPS_data()
        update_GPS_position_in_forklift_data()
        payload = prepare_gps_payload()
        send_data_to_backend(payload)

function send_status_payload():
    payload = prepare_status_payload()
    send_data_to_backend(payload)

// --- Backend Communication ---

function handle_backend_commands():
    if receive_NB_IoT_message_from_backend():
        log("Message received from backend.")
        command = parse_message()
```

```
            if command_is_change_location_mode(command):
                handle_location_mode_change(command)

    else if end_of_workday(command):
                handle_reset_command()

function handle_location_mode_change(command):
    new_mode = extract_new_mode(command)

    if new_mode == "GPS":
        activate_GPS_module()
        isGpsModuleActive = true
        LocationMode = "GPS"
        log("Switched to GPS mode.")

    else if new_mode == "RFID":
        deactivate_GPS_module()
        isGpsModuleActive = false
        LocationMode = "RFID"
        log("Switched to RFID mode.")

    confirm_command_receipt_to_backend()

function handle_reset_command():
    log("Reset command received. Restoring default system state.")
    impactDetected = 0
    AvgSpeed = 0
    MaximumSpeed = 0
    forkliftData = create_empty_forklift_data_object()
    confirm_command_receipt_to_backend()

// --- GPS Control ---

function activate_GPS_module():
    log("GPS module is powering ON.")

function deactivate_GPS_module():
    log("GPS module is powering OFF.")
```

Backend Architecture



Introduction

The IoT system backend for forklift tracking is designed to manage real-time location, status monitoring and data analysis from both the internal (RFID-based) and external (GPS-based) areas of the warehouse. The architecture must support two-way communication via NB-IoT, real-time data processing and provide intuitive interfaces for visualising and managing the system.

Data Ingestion

The entry point for all data in the system is an MQTT broker, implemented using e.g. Mosquitto. This component handles all communication from the forklifts and RFID readers installed in the warehouse. The broker configuration provides a topic structure organised hierarchically according to the pattern: 'warehouse/{area}/{device_type}/{device_id}/{data_type}'.

Messages are classified using different Quality of Service levels: QoS 0 for location and status data, and QoS2 for critical alerts that need guaranteed delivery.

Data Processing

An important part is real-time position processing, managing the location of the forklift in both areas of the warehouse. For the indoor area, it implements triangulation algorithms based on RSSI values received from multiple RFID readers, calculating the position through triangulation techniques that consider the estimated distance from each reader. For the outdoor area, it validates and filters the GPS coordinates received from the forklift, verifying their accuracy and temporal consistency.

A command service would manage two-way communication to the forklift, implementing an MQTT client for sending commands like the crucial aspect of the management of zone transitions, where the system automatically detects the passage of forklift between indoor and outdoor areas using the dedicated RFID reader located on the ramp.

When a transition is detected, the system sends commands to the forklift to activate or deactivate the GPS module according to the new zone of operation, thereby optimising energy consumption.

Status analysis service is responsible for receiving and storing data on maximum and average speeds and distances travelled.

Another important feature is the reset of data of all the ESPs at the end of the workday. Sending a specific public command to all the forklifts, in fact, is possible to set all the variables stored to default values.

Database

To ensure scalability, reliability and accessibility of the data collected by the IoT system, a cloud-based database is implemented. The solution adopted involves the use of Amazon DynamoDB or MongoDB Atlas, NoSQL databases that are particularly suitable for managing the semi-structured, high-volume data typical of IoT applications.

The database is structured to optimise the different types of queries required by the system. The main tables include: a collection for real-time location data with partitioning by device_id and timestamp, one for historical performance data aggregated daily, and one dedicated to critical events such as impacts.

The indexing strategy is designed to effectively support the most frequent queries: search by forklift ID and time interval, geospatial queries for heatmap analysis, and filters by event type and warehouse area. The database also implements automatic retention policies that archive data older than 2 years in long-term storage to reduce costs, keeping only the most recent data available online for daily operational analysis.

Cloud integration offers significant advantages in terms of automatic backup, disaster recovery and automatic scaling based on workload, ensuring high system availability even during peaks in usage or forklift fleet expansions.

Visualization

The real-time dashboard provides an instant view of the system status. The interactive warehouse map shows the current locations of all forklifts with coloured status indicators, real-time location updates, instant notifications for critical alerts and widgets for key metrics such as number of active forklifts and aggregate performance.

The analytics dashboard offers historical views and trend analysis. It includes heatmaps of historical locations to identify patterns of usage, time graphs of speed and distance for performance analysis, and efficiency comparisons between different shopping carts or time periods.