THE UNIVERSITY OF DANANG

UNIVERSITY OF SCIENCE AND TECHNOLOGY

DANANG INTERNATIONAL INSTITUTE OF TECHNOLOGY

Pellegrini Romain

Zanin Nicolas

Group 5.1

Real-time Time Display LED System at Bus Stations

(sensor part)

Table of content

1.1 Overview **3**
a. The motivation of the project 3
b. Existing techniques solutions 3
1.2. Preliminary solution **4**
a. Technical solution with block diagram 4
b. Design process 5
Stage 1 : Determine the goals and the requirements 5
Stage 2 : Research the technologies used in the system 6
Stage 3: Research on microcontroller programming and types of sensors used in the system 6
Stage 4 : System design 6
Stage 5 : Test and check 7
c. Requirements for the system 8
Requirement 1 : For wearable devices 8
Requirement 2 : For LoRa 8
Requirement 3: For monitoring software 9
d. Main components for implementation 9
1.1 List of the devices 9
1.2 Details of the components 10
1.3 Expected Results 11
1.4 Evaluation Method 11
1.5. Plan of implementation 12
References **16**

1.1 Overview

a. The motivation of the project

To enhance public transportation systems, LED information displays are installed at bus stations to offer real-time information to passengers, improving overall user experience.

These displays provide a range of information, including bus arrival times, route details, service alerts, real-time tracking, weather updates, advisory notices, emergency notifications, advertisements, and promotions.

Multilingual support ensures accessibility for all passengers, while a smartphone app allows users to access information for each LED-equipped bus station.

This update will allow commuters to know the state of their bus, will calm users during the wait and help renew their confidence in the bus network.

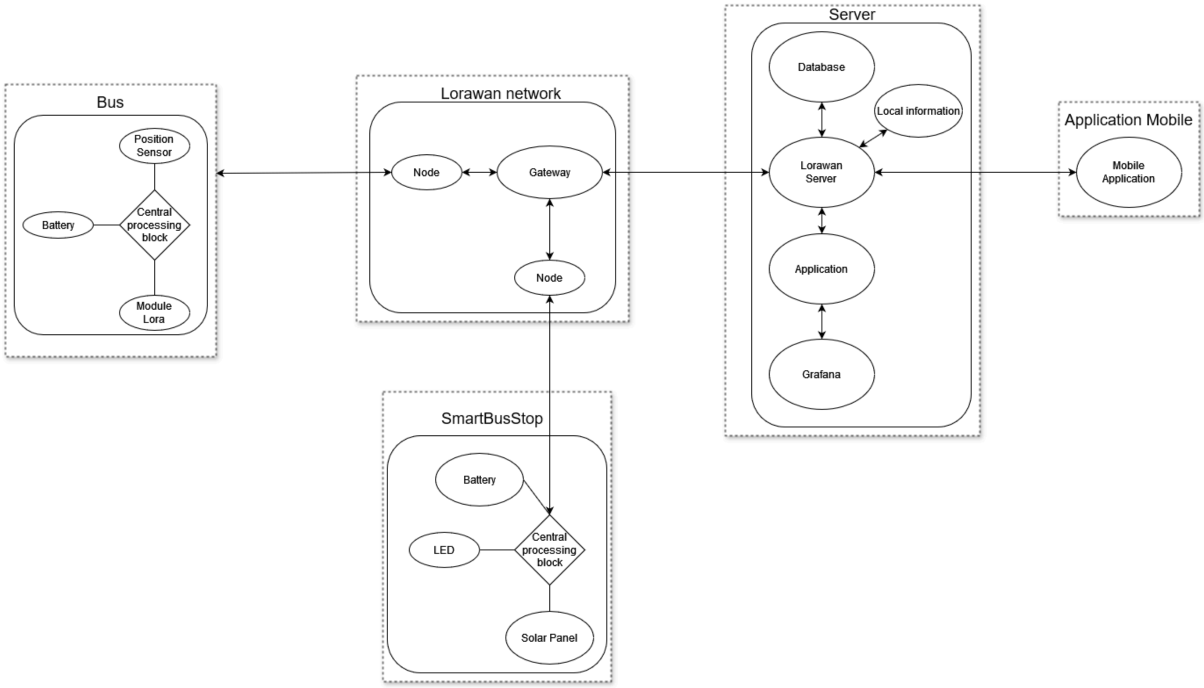
Our purpose is to retrieve the data on a bus with our board. We need to send it to the LoRaWan server and print it on a LED display or the mobile application.

b. Existing techniques solutions

- Real Time Bus Monitoring System
- Real Time Bus Position and Time Monitoring System

1.2. Preliminary solution

a. Technical solution with block diagram



b. Design process

Stage 1 : Determine the goals and the requirements

- Clarify the goals of wearable devices on the buses.

    - The GPS goal is to get the current position to transmit to the LoRaWan server by passing the LoRa Module.
    - The LoRa module goal is to send a position to the LoRaWan network by passing LoRaWan Network.
    - The battery goal is to avoid the need for buses to be plugged in
    - The objective of wearable devices on buses is to transmit position data to the LoRaWAN network.
      The transmitted position will be used to calculate the real-time duration to reach the next station.

- Set specific requirements :
    - Set the components needed for implementation.
        - The GPS module determines the current position and transmits it to the LoRa module, which can then send the position to the server. The server uses this information to calculate the real-time duration required to reach the next station.
        - The two batteries, each with a capacity of 2000mAh and a voltage of 3.7V, allow for autonomy without the need for a cable connection, thus eliminating the necessity for frequent recharging.
        - The LoRa module is responsible for transmitting data (position) to the LoRaWAN network. Within the LoRa network, this data is forwarded to the LoRaWAN server.
        - The STM32 card needs to send position data to the LoRaWAN Server and will be used to calculate the real-time duration to reach the next station.
    - List the different constraints of each device
        - The battery constraint requires occasional battery replacements.
        - The constraint with the LoRa module lies in its energy consumption, as sending data over the network consumes energy
        - The constraint of the STM32 card is the battery, as we can only use two batteries with a capacity of 2000 mA and 3.7V, so we sometimes need to change the battery.

Stage 2 : Research the technologies used in the system

- Overview of LoRaWan technology.
    - LoRaWAN is a Media Access Control (MAC) layer protocol built on top of LoRa modulation. LoRaWAN is suitable for transmitting small size payloads over long distances. It is a software layer which defines how devices use the LoRa hardware.
- Learn about communication protocols in the system particularly the upload of the data on LoRaWan server.
    - It's a long-range, low-power wireless protocol designed to connect objects to the Internet. We will use it to send the sensor data to the database in the cloud.

Stage 3: Research on microcontroller programming and types of sensors used in the system

- Learn about microcontroller programming.
    - The programming language is C/ C++ for microcontroller and we put the code in the card with the arduino IDE.
- Learn about the sensor modules used in the system and how to read values from the sensors.
    - The GPS module is the ublox CAM-M8Q-0-10 and for reading values from the sensor we will use the library "SoftwareSerial.h".
    - The LoRa module is RAK3172 and we will follow this tutorial to send data : Link Github Lora

Stage 4 : System design

- Design components in the system.
    - Gps sensor to find the position of the bus.
    - LoRaWAN chip to communicate with the LoRaWAN server.
    - Arduino board to control the sensors
- Send the data to the LoRaWan server.
    - We can use the library LoRa.h to transfer data via LoRa.
- Define the emergency notification
    - We need to display some emergency notifications. We had to set which notification we consider in an emergency, and if the sensor will be able to capture it or not.

Stage 5 : Test and check

- Conduct performance testing of the wearable device and the associated application.
    - Execution time measurement :
      We will use time library as like “time.h” to view execution time

    - Energy consumption:
      We will analyze the energy consumption thanks at a library INA219, that can analyze the current voltage and energy uses

    - Memory Test:
      We will use for test memory, a module already here in arduino IDE as like Memory profiler.
      For mobile applications, we use the same module

    - Communication Test:
      For testing communication speed between the microcontroller and the server, we record the time when the packet is sent and when it is received by the server. We then compare these times, thanks to the timestamps provided by the network (or console print at each location)

      For application performance, We will check if all data packets have been received and verify the receipt time of these packets depending on the network type used by the customer

      In general rule, We use Grafana to view various statistics such as time, memory, etc., to get a performance overview

- Ensure that the system operates stably and responds properly to the requirements.

  To ensure the stable operation of the system, we employ integration tests to verify that it responds appropriately to requirements, and unit tests to ensure its stability.
  For unit testing on microcontrollers, we use the Unity library, while for mobile applications, we rely on the JUnit library.
  Additionally, we implement tests according to the project timeline to ensure adherence to project deadlines.

c. Requirements for the system

Requirement 1 : For wearable devices

- **GPS positioning :**
  - Measuring parameters: Determine the bus exact location to provide geographical information and calculate the travel time.
  - Accuracy: GPS needs to be accurate within 20 meters or better to provide accurate positioning information.
  - Measuring range: Depends on environmental conditions, but typically between 0-10,000 meters.

- **Connect to LoRa gateway :**
  - Requirements: The device needs to be able to connect to the LoRa gateway over long distances, especially with obstacles like the buildings.
  - Farthest distance: Depending on environmental conditions and areas such as urban or suburban, the farthest distance can be 15km under ideal conditions.
  - Note: In real environments, the connection distance may be reduced due to various obstacles and environmental conditions.
- **Battery and usage time :**
  - Requirements: The battery needs to have enough capacity to maintain device operation for the expected time.
  - Usage time: Depending on the device's power consumption, the 2000mAh, 3.7V Lithium Polymer Battery can provide power for a few hours to a few days, depending on usage.
  - Size and power consumption: The device needs to be compact in size, save space, and have the lowest possible energy consumption to ensure long-term operation of the device.

Requirement 2 : For LoRa

- **Spreading Factor (SF)** is a factor that affects the speed and energy consumption of devices, usually ranging from 7-12.
- **BW (Bandwidth):** there are three main types of bandwidth: 125kHz, 250kHz, and 500kHz - the deciding factor in the data rate and transmission distance of the devices. The larger the bandwidth, the shorter the transmission distance, so to achieve the longest transmission distance, choose the smallest bandwidth - 125kHz.
- **CR (Coding Rate):** the factor that affects the time delay when the device transmits data to the LoRa gateway and monitoring software, usually the delay can range from a few seconds to a few minutes.

Requirement 3: For monitoring software

- **Parameter display :**
  - GPS location: Displays the bus exact location on the map, updated continuously.
  - Print the bus schedule

- **Map integration :**
  - Need to use a map service or map API (Google Maps) to display the bus location on a map.
  - The map needs to support GPS location display.

- **Problem warning :**
  - An emergency appended
  - If the bus makes a detour the sensor has to understand it and it will notify the detour.
- **How to warn :**
  - The LED display will notify the commuters at the bus station
  - A notification could be sent through the app to notify the commuters anywhere.

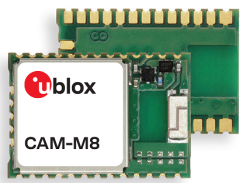
d. Main components for implementation

1.1 List of the devices

Based on the system requirements mentioned in section C, we have decided to choose the following components to design the system :

- **MCU :** uses LoRaWan RAK3172 module.
- **GPS positioning module :** uses CAM-M8Q-0-10 GPS module.
- **Battery :** choose Rechargeable Battery Lithium 2000mAh, 3.7V

1.2 Details of the components

Components	Device name	Specification
MCU (LoRa)	RAK3172 	- Dimension: 15 x 15 x 3.5mm - Operating temperature: -40 degrees to 85 degrees - Supply voltage: 2.0-3.6V - Frequency range: 150MHz to 960MHz - Datasheet:[1]
GPS	CAM-M8Q-0-10 	- Dimension of GPS board: 9.6 x 14 x 1.95 mm - Dimension of antenna: 9.6 x 14.0 x 1.95 mm - Operating temperature: -40 degrees to 85 degrees - Supply voltage: 1.65-3.6V - Accuracy: 2.5m - Number of channel: 72 - Datasheet: Link ublox
Battery	Rechargeable Battery Lithium 	- Capacity: 2000mAh - Voltage: 3.7V - Dimension: 18 * 65mm - Weight: 60g - Cycle life: 500 times

1.3 Expected Results

- Find the localisation of the bus with good accuracy.
- Get all route information on mobile application
- Get weather information on mobile application and LED displays
- Calculate the travel time of the bus and display his road on mobile application.
- Verify delays, detours or disruption and display on mobile application and LED display
- Have a real-time update for the commuters

1.4 Evaluation Method

- · **Evaluation of effectiveness:**
  - Always have correct data sent at the chosen frequency
- · **Cost assessment:**
  - Standards: attract investors to earn returns and accelerate the payback period.
- · **Complexity rating:**
  - Add more bus lines and change the bus station of each line easily. So good maintenance and good for developing new features.
- · **Accuracy and performance rating:**
  - Evaluation methods : product testing on real environment (bus from the line 5)

    + Evaluate the accuracy of the data

    + Check that the system response time is within the allowable range.
  - Standards: the results obtained are within the allowed threshold, and the data obtained can be accepted as reliable.

1.5. Plan of implementation

Timeline	2024												2024
	May				June				July				August
Member	W1	W2	W3	W4	W1	W2	W3	W4	W1	W2	W3	W4	W1
Romain Pellegrini													
Nicolas Zanin													

Romain Pellegrini	**Nicolas Zanin**	
Report		
Setup API server LoRaWan		
Calculate and display the path	Send data and position	
Checking with delay or detour	Add meteo API & Grafana	
Display bus map	Display bus information	
Notify customer		
Finalization and adjustment project		

Timeline		Objectives	Expected results	Confirmation from instructor
01/10-31/10/2024	Weeks 1-2	Report complete	Report complete and validate by instructor	
	Weeks 3-4	Setup API Server so the server can get and store position	Store on the server the information of the schedule and the path of the bus	
01/11-30/11/2024	Weeks 5-6	Calculate the path with a position to the next station	Send the position with the board and store it on the server	
	Weeks 7-8	Checking position and evaluating if the bus had a delay or did a detour.	Check performance application with Grafana, and meteo with an api	

01/12-31/12/2024	Weeks 9-10	Display route information, so display the bus number, destination, and stops on mobile application	The customer with mobile application can see the destination, and stops depending on bus number	
	Weeks 11-12	Display detour, or disruptions due to accidents or construction and add Emergency Notification on mobile application	The customer with mobile application can see the service alert (detour, disruption, etc.) and Emergency Notification	
01/01-04/01/2024	Week 13	Finalization and adjustment of project	Correct the possible bugs or design problem	

References

[1] K. Eurviriyanukul, K. Phiewluang, S. Yawichai, S. Chaichana, “Evaluation of Recognition of Water-meter Digits with Application Programs, APIs, and Machine Learning Algorithms”, 8th International Electrical Engineering Congress (iEECON), pp. 1-4, 2020.