ELEC339 - Sensors
Cold Chain Logistics Monitoring System

Buğra Tufan
Department of Electrical and Electronics Engineering
Koç University
Istanbul, Turkey
btufan21@ku.edu.tr

***Abstract*— With the continuous development of the logistics industry, the relevant technology is becoming more mature, the system is increasingly perfect, and the accuracy requirements for the logistics monitoring system are becoming higher and higher. Traditional logistics monitoring is usually manually recorded, which wastes human resources and is inefficient, and cannot meet the logistics industry's current scale and intelligent development requirements. This project aims at providing an example application scenario for cold-chain logistics monitoring system that uses various IoT and sensor technologies.**

Keywords* — *logistic, cold-chain, monitoring, IoT

I. Introduction

Temperature-sensitive products like chemicals, meat, plants, drugs, and vaccines require temperature-controlled environments from production to consumption and every step in between. A cold chain is a temperature-controlled supply chain established to provide an uninterrupted series of refrigerated production, storage and distribution processes. Equipment and logistics are also critical in maintaining a desired range of low temperatures. [1]

Traditional logistics monitoring is usually manually recorded, which wastes human resources and is inefficient, and cannot meet the logistics industry's current scale and intelligent development requirements.

II. Proposed Method

A. Hardware Architecture

The hardware system has three sensors for temperature sensing, acceleration measurement, and localization. Bosch **BNO055** sensor is used for acceleration measurement. This sensor has an I2C interface for data reading and configuration purposes. **NEO-6MV** GPS module is used for localization. This module checks its location on earth and provides output data: the longitude and latitude of its position. For the user interface of hardware, **SSD1306** OLED display is used. U8glib library is used for driving the OLED display. [2]

These sensors and the display are connected to an Arduino Pro Mini board. This board has an **ATMEGA328** microcontroller. Sensor reading and display driving operations are performed on this processor, thanks to the software created for this project.

The collected data by the Arduino board are transmitted to the Raspberry Pi module using the UART protocol in JSON format. On the Raspberry Pi side, received data are decoded and sent to the server by HTTP protocol. The server system records the telemetry data to the database individually for each user.

The architectural diagram of the hardware is given in Figure 1.

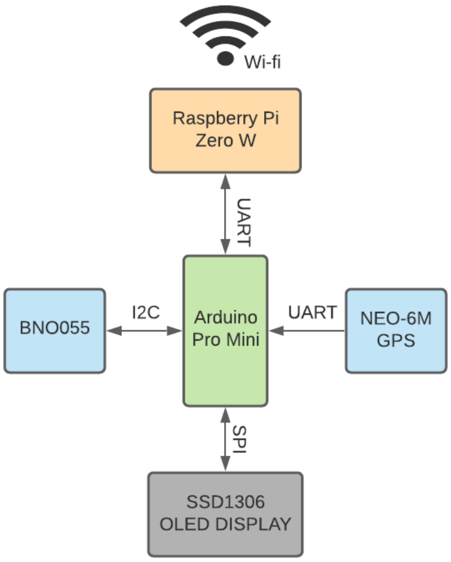

Figure 1: Architecture of Hardware

B. Database Management and Visualization - IoT Server

To support millions of IoT devices, the efficiency of server systems is crucial. IoT cloud architecture that supports multiple devices is created. This IoT server is capable of database management and data visualization.

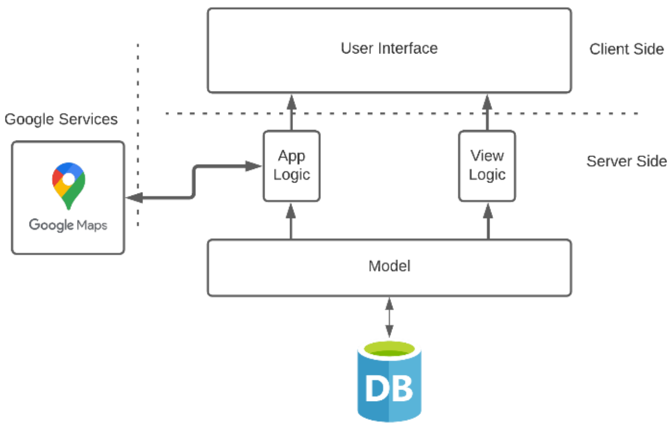

Figure 2: Architecture of IoT Server

To achieve efficient IoT server architecture, **Django** framework is used the frontend and backend systems of the IoT server. This framework allows creating utilized server infrastructure resulting in an efficient server system.

For the visualization, a map view interface is created. Users can access recorded data as a map view. Google Maps API is used to create a map view for this interface. The collected data is processed, and processed information is transmitted to Google's server by using Google API. The API response is sent to the user interface. The users can view the record points as placemarks and have telemetries. So, users can also see telemetries by clicking these placemarks. The map view of user interface is given in Figure 3.

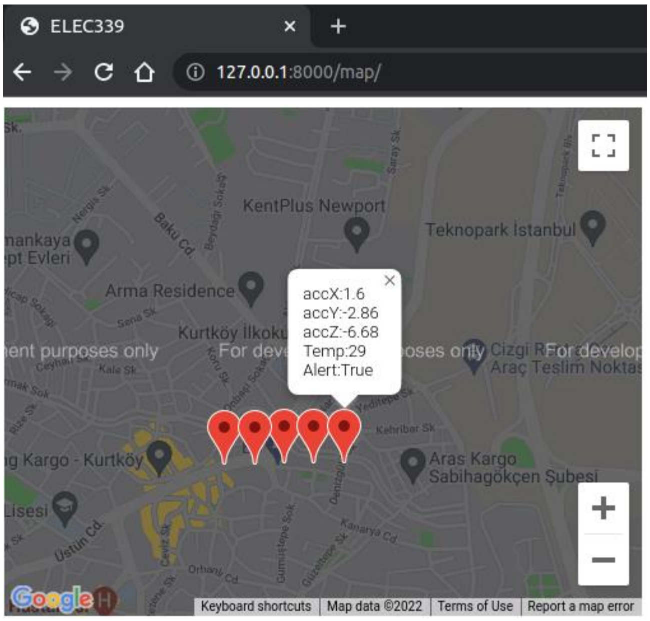

Figure 3: User Interface of Monitoring System

Consequently, HTTP server is introduced as the application servers of the IoT cloud. The HTTP servers can provide services for end-users and ensure many device connections and real-time communication among devices.

C. Mechanical Design

Mechanical rigidity and durability are essential features for end-user products. All parts were drawn with the CAD tool, and an assembly image was created. **SolidWorks** was used for this project. The mechanical design of hardware is given in Figure 4.

Figure 4: Mechanical Design of Hardware

III. FINAL HARDWARE AND SOFTWARE DESIGN

After a long soldering and assembling process, the designed hardware is finished as shown in Figure 5.

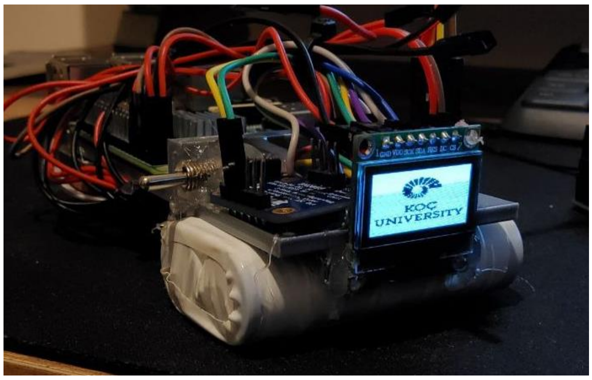

Figure 5: Final Hardware

Various pages were created for each sensor and these pages were printed on the display. So that live data can be viewed by users. Improvements have been made to the embedded software so that the sensor readings are minimally affected by the screen update processes. The view of acceleration values page is given in Figure 6 as an example.

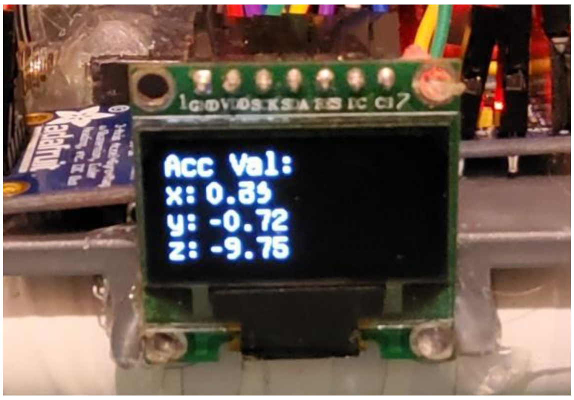

Figure 6: Accelerometer Values Displayed on SSD1306.

Also, the hardware system has an integrated battery module that contains two 18650 battery modules and high efficient buck converter to supply uninterrupted power. The battery pack has individual connections to battery cells. These connections give the ability for separate charging to achieve long battery life. The battery module of the hardware is shown in Figure 7.

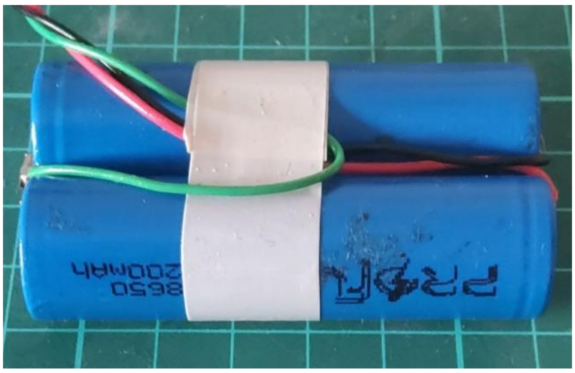

Figure 7: Battery Module of Hardware

IV. Conclusion

The cold chain monitoring solutions help businesses streamline processes in the supply chain and enhance decision-making. These systems help businesses achieve a temperature-controlled supply chain that provides operational efficiency and improves quality control.

In practice, as proposed above, this way of monitoring systems provides the ability to access information from anywhere at any time on any device.

Consequently, automating tasks helps improve the quality of a business's services and reduces the need for human intervention, saving time and money. Also, this kind of system enhances employee productivity.

References

[1] Cold Chain Monitoring - The Complete Guide, Aug. 2020. Accessed on: Jan. 10, 2022. [Online]. Available: https://www.logmore.com/post/cold-chain-monitoring

[2] U8glib. Accessed on: Jan. 10, 2022. [Online]. Available: https://www.arduino.cc/reference/en/libraries/u8glib/