Design an Intelligent Real-Time Public Transportation Monitoring System Based on IoT

PHASE 1: PROBLEM DEFINITION AND DESIGN THINKING

TEAM MEMBERS:

TEAM LEADER: Mopuri Lokesh Dhananjayan -112821104047

Karthikeyan .s -112821104041

Mohammed Asraf Ali S-112821104043

mukesh Varma P- 112821104048

varaprasath P- 112821104069

Abstract

In recent years, smart public transportation has become one of the most important things for developing cities and improving people's quality of life. Public transportation users face many problems, the most important of which is the long wait at the bus station. The proposed system in this paper helps users of public transportation to find public transportation, arrival times, and other information from any place and at any time using a mobile application. The main objective of the prototype is to reduce the wait time at the bus station by knowing the nearest buses to a user, the real-time location of buses on the Google map to help passengers track buses in real-time, the arrival time of buses, and speed. The system was implemented based on Internet of Things (IoT) technology, by using the Global Positioning System (GPS), a microcontroller with a built-in Wi-Fi module (ESP32), and a mobile user interface by the Blynk IoT platform. The proposed system has been implemented and tested in real-time, where all data obtained is displayed by the GPS sensors for bus locations (longitude and latitude) and speed on the smartphone application. The distance between the bus location and the passenger that will appear in the mobile app was calculated by using the Haversine formula It measured the accuracy of the distance obtained based on a study of several days at different times on multiple roads in Mosul city and compared it with the actual distance. The average difference between the calculated values computed by the Haversine equation compared to the data obtained from the actual distance is (177 meters) with a minimum error of 8 meters. The arrival time of each bus was calculated based on the distance and average speed of the bus registered along the road.

Subject Areas

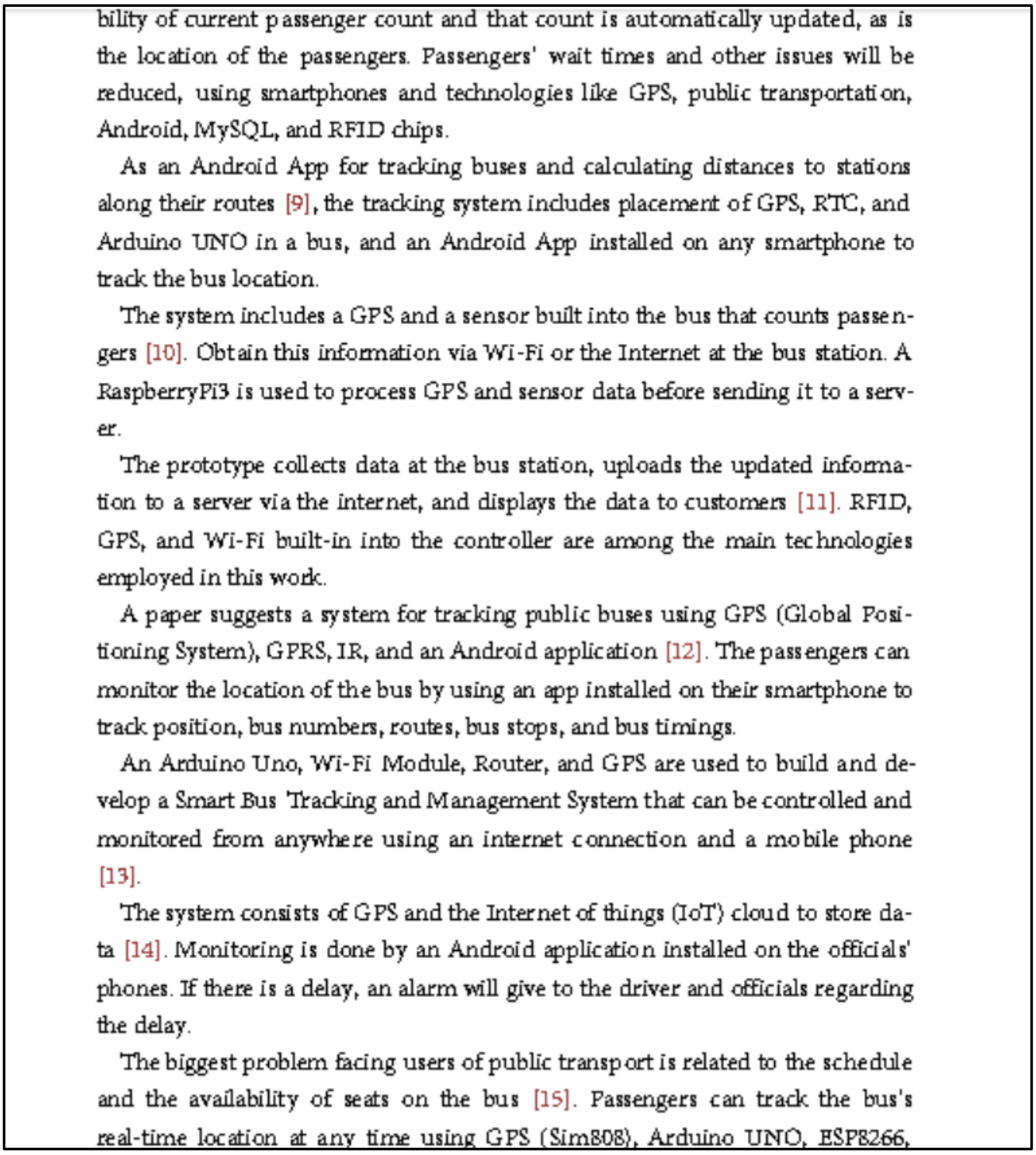
Wireless Communication, Computer Engineering

Keywords

ESP32, Global Position System (GPS), Internet of Things(IoT), Blynk, Haversine

1. Introduction

Public transportation refers to shared passenger transportation services such as buses, trolleybuses, trains, ferries, and expresses transportation like the metro [1]. Intelligent Transportation Systems (ITS) has a subsystem called Smart Public Transportation (SPT). It can intelligently monitor public transit networks to ensure their operation and to provide clients with information on excursions and system operating conditions [2]. Smart public Design of an intelligent real-time Public Transportation Monitoring System based on IoT transport systems, depending on several technologies, allows SPTS to retrieve data from multiple sensor systems and to manage and control the transportation network. There are many innovative technologies, which promoted the development and implementation of smart public transport systems, such as Geographical Information Systems (GIS), Automatic Vehicle Location Systems (AVLS) and Traveler Information Systems (TIS) [3]. The Internet of Things (IoT) is a network of interconnected physical objects that can be accessed through the internet. As countries aim to improve their citizens' standard of living, they begin to improve the infrastructure of their cities, towns, and villages. Improving public infrastructure would also require bettering city transit services. In cities, bus services are the backbone of the public transportation system (PTS). Because of many issues such as wait times, traffic congestion, etc., the reliability of public transport is being diminished; SPTS is using IoT to avoid any of these issues [4]. Wireless communication is the transmission of data between two or more points without the use of wires or cables. There are several technologies used to manage



bility of current passenger count and that count is automatically updated, as is the location of the passengers. Passengers' wait times and other issues will be reduced, using smartphones and technologies like GPS, public transportation, Android, MySQL, and RFID chips.

As an Android App for tracking buses and calculating distances to stations along their routes [9], the tracking system includes placement of GPS, RTC, and Arduino UNO in a bus, and an Android App installed on any smartphone to track the bus location.

The system includes a GPS and a sensor built into the bus that counts passengers [10]. Obtain this information via Wi-Fi or the Internet at the bus station. A RaspberryPi3 is used to process GPS and sensor data before sending it to a server.

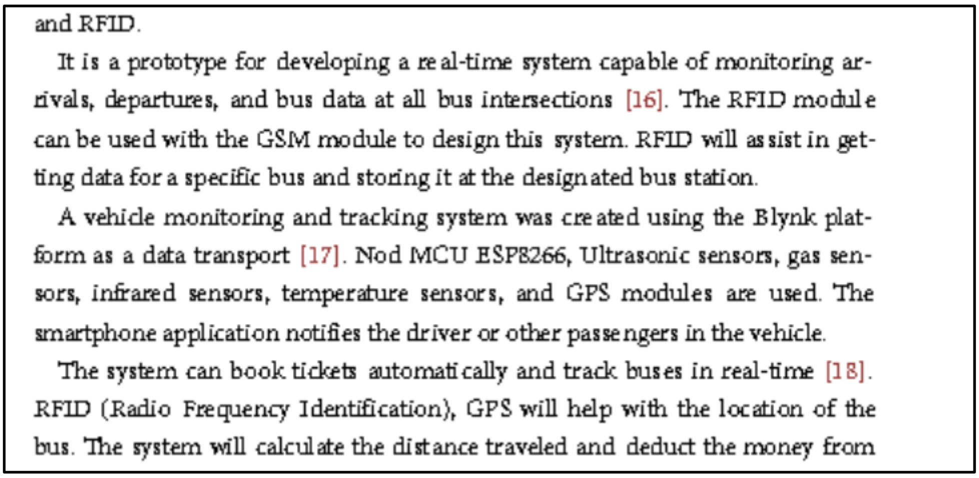
The prototype collects data at the bus station, uploads the updated information to a server via the internet, and displays the data to customers [11]. RFID, GPS, and Wi-Fi built-in into the controller are among the main technologies employed in this work.

A paper suggests a system for tracking public buses using GPS (Global Positioning System), GPRS, IR, and an Android application [12]. The passengers can monitor the location of the bus by using an app installed on their smartphone to track position, bus numbers, routes, bus stops, and bus timings.

An Arduino Uno, Wi-Fi Module, Router, and GPS are used to build and develop a Smart Bus Tracking and Management System that can be controlled and monitored from anywhere using an internet connection and a mobile phone [13].

The system consists of GPS and the Internet of things (IoT) cloud to store data [14]. Monitoring is done by an Android application installed on the officials' phones. If there is a delay, an alarm will give to the driver and officials regarding the delay.

The biggest problem facing users of public transport is related to the schedule and the availability of seats on the bus [15]. Passengers can track the bus's real-time location at any time using GPS (Sim808), Arduino UNO, ESP8266,

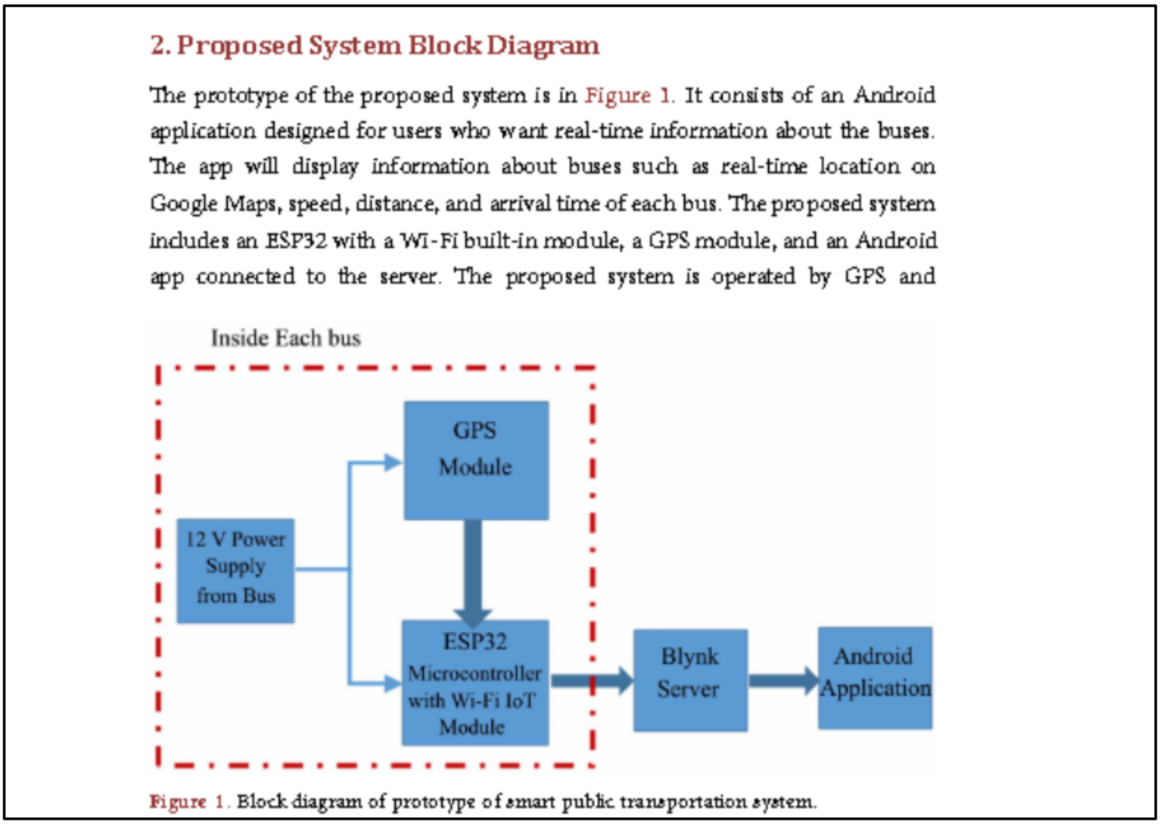


and RFID.

It is a prototype for developing a real-time system capable of monitoring arrivals, departures, and bus data at all bus intersections [16]. The RFID module can be used with the GSM module to design this system. RFID will assist in getting data for a specific bus and storing it at the designated bus station.

A vehicle monitoring and tracking system was created using the Blynk platform as a data transport [17]. Nod MCU ESP8266, Ultrasonic sensors, gas sensors, infrared sensors, temperature sensors, and GPS modules are used. The smartphone application notifies the driver or other passengers in the vehicle.

The system can book tickets automatically and track buses in real-time [18]. RFID (Radio Frequency Identification), GPS will help with the location of the bus. The system will calculate the distance traveled and deduct the money from

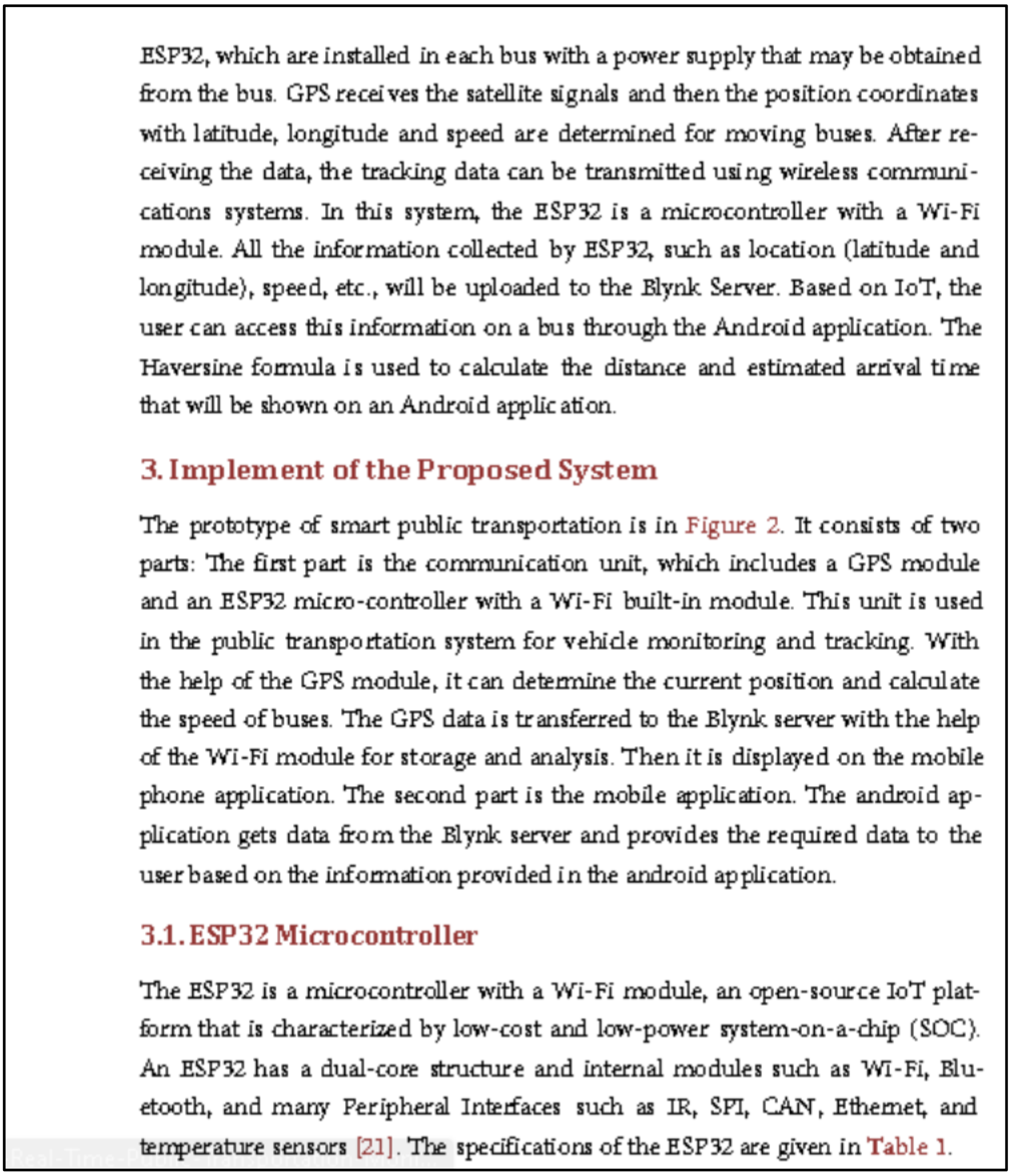


2. Proposed System Block Diagram

The prototype of the proposed system is in Figure 1. It consists of an Android application designed for users who want real-time information about the buses. The app will display information about buses such as real-time location on Google Maps, speed, distance, and arrival time of each bus. The proposed system includes an ESP32 with a Wi-Fi built-in module, a GPS module, and an Android app connected to the server. The proposed system is operated by GPS and

Inside Each bus
GPS Module
12 V Power Supply from Bus
ESP32 Microcontroller with Wi-Fi IoT Module
Blynk Server
Android Application

Figure 1. Block diagram of prototype of smart public transportation system.



ESP32, which are installed in each bus with a power supply that may be obtained from the bus. GPS receives the satellite signals and then the position coordinates with latitude, longitude and speed are determined for moving buses. After receiving the data, the tracking data can be transmitted using wireless communications systems. In this system, the ESP32 is a microcontroller with a Wi-Fi module. All the information collected by ESP32, such as location (latitude and longitude), speed, etc., will be uploaded to the Blynk Server. Based on IoT, the user can access this information on a bus through the Android application. The Haversine formula is used to calculate the distance and estimated arrival time that will be shown on an Android application.

3. Implement of the Proposed System

The prototype of smart public transportation is in Figure 2. It consists of two parts: The first part is the communication unit, which includes a GPS module and an ESP32 micro-controller with a Wi-Fi built-in module. This unit is used in the public transportation system for vehicle monitoring and tracking. With the help of the GPS module, it can determine the current position and calculate the speed of buses. The GPS data is transferred to the Blynk server with the help of the Wi-Fi module for storage and analysis. Then it is displayed on the mobile phone application. The second part is the mobile application. The android application gets data from the Blynk server and provides the required data to the user based on the information provided in the android application.

3.1. ESP32 Microcontroller

The ESP32 is a microcontroller with a Wi-Fi module, an open-source IoT platform that is characterized by low-cost and low-power system-on-a-chip (SOC). An ESP32 has a dual-core structure and internal modules such as Wi-Fi, Bluetooth, and many Peripheral Interfaces such as IR, SPI, CAN, Ethernet, and temperature sensors [21]. The specifications of the ESP32 are given in Table 1.

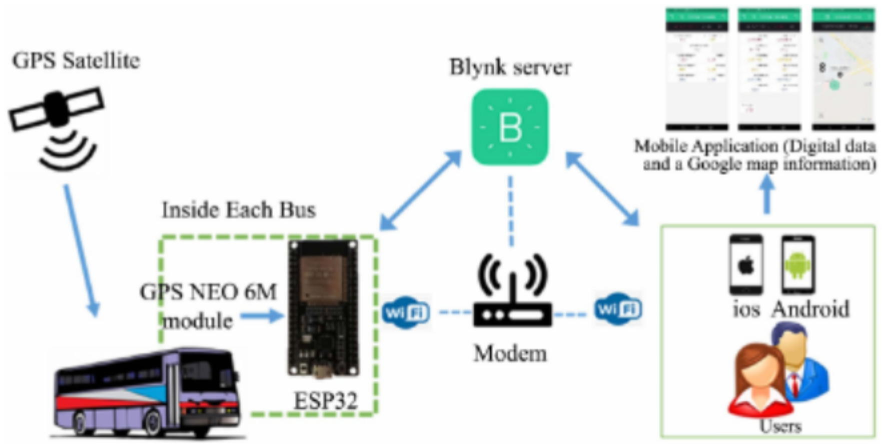


Figure 2. Prototype of smart public transportation architecture.

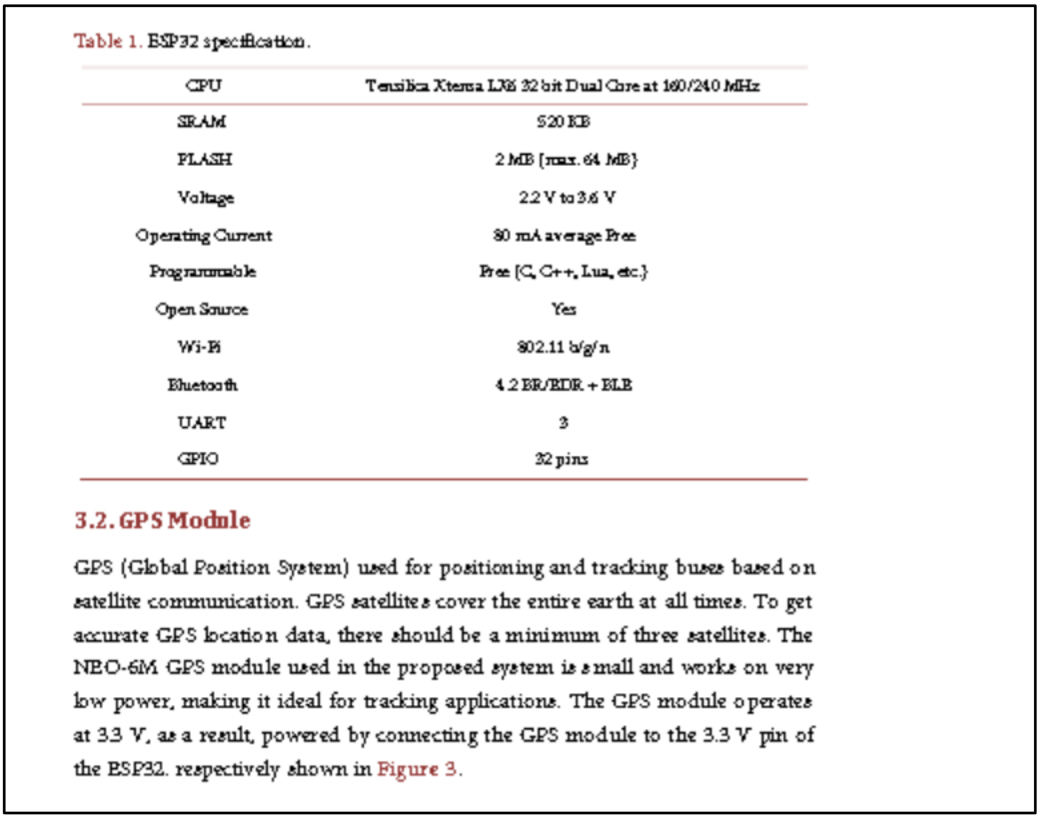


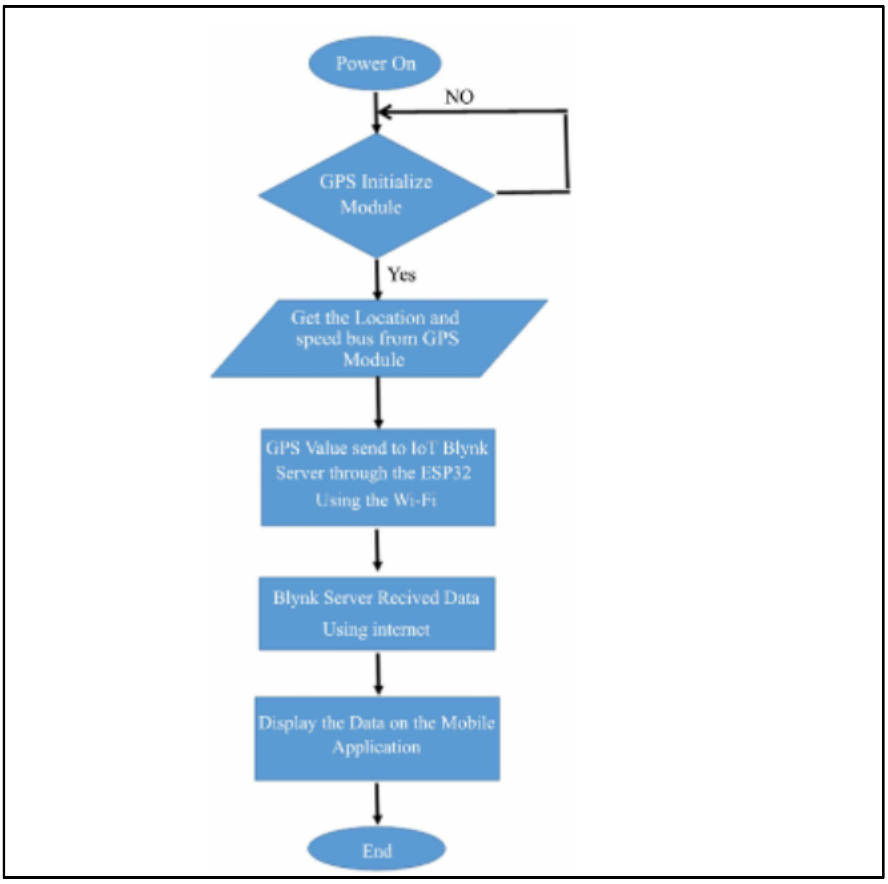
Table 1. ESP32 specification.

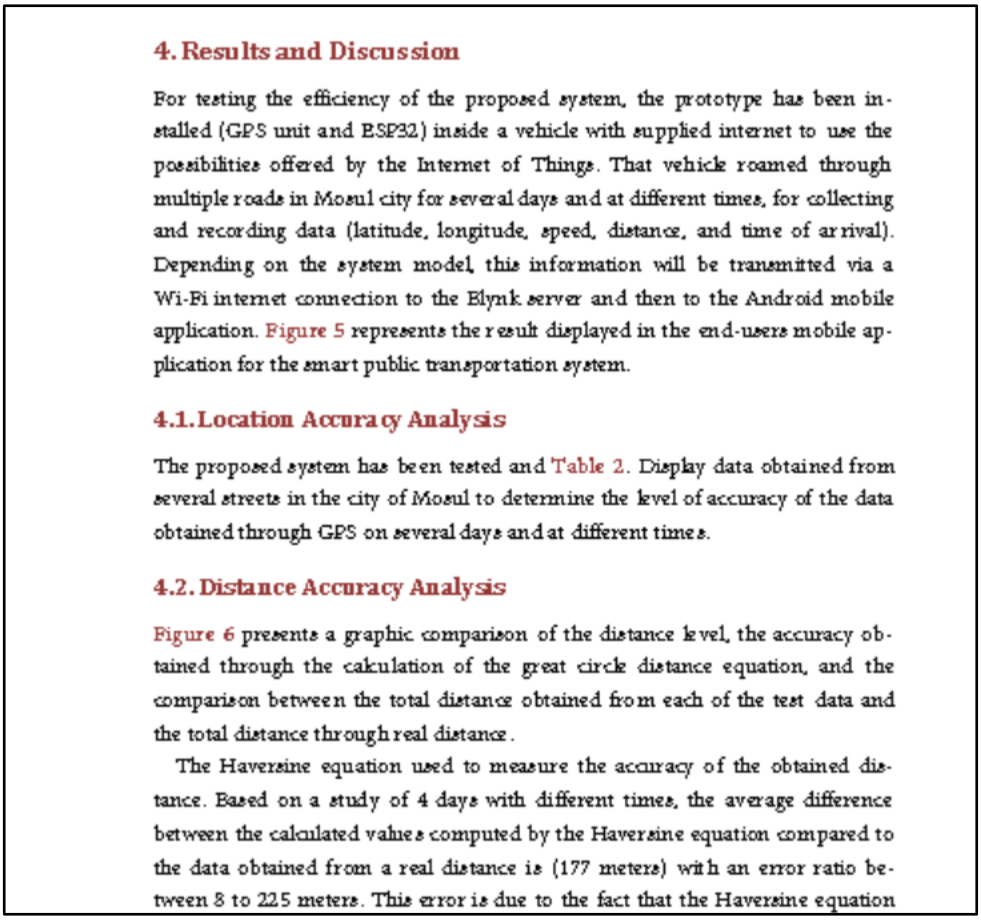
CPU	Tensilica Xtensa LX6 32 bit Dual Core at 160/240 MHz
SRAM	520 KB
FLASH	2 MB (max. 64 MB)
Voltage	2.2 V to 3.6 V
Operating Current	80 mA average Free
Programmable	Free (C, C++, Lua, etc.)
Open Source	Yes
Wi-Fi	802.11 b/g/n
Bluetooth	4.2 BR/EDR + BLE
UART	3
GPIO	32 pins

3.2. GPS Module

GPS (Global Position System) used for positioning and tracking buses based on satellite communication. GPS satellites cover the entire earth at all times. To get accurate GPS location data, there should be a minimum of three satellites. The NEO-6M GPS module used in the proposed system is small and works on very low power, making it ideal for tracking applications. The GPS module operates at 3.3 V, as a result, powered by connecting the GPS module to the 3.3 V pin of the ESP32. respectively shown in Figure 3.

**Work the Proposed System


Power On
NO
GPS Initialize Module
Yes
Get the Location and speed bus from GPS Module
GPS Value send to IoT Blynk Server through the ESP32 Using the Wi-Fi
Blynk Server Recived Data Using internet
Display the Data on the Mobile Application
End



4. Results and Discussion

For testing the efficiency of the proposed system, the prototype has been installed (GPS unit and ESP32) inside a vehicle with supplied internet to use the possibilities offered by the Internet of Things. That vehicle roamed through multiple roads in Mosul city for several days and at different times, for collecting and recording data (latitude, longitude, speed, distance, and time of arrival). Depending on the system model, this information will be transmitted via a Wi-Fi internet connection to the Blynk server and then to the Android mobile application. Figure 5 represents the result displayed in the end-users mobile application for the smart public transportation system.

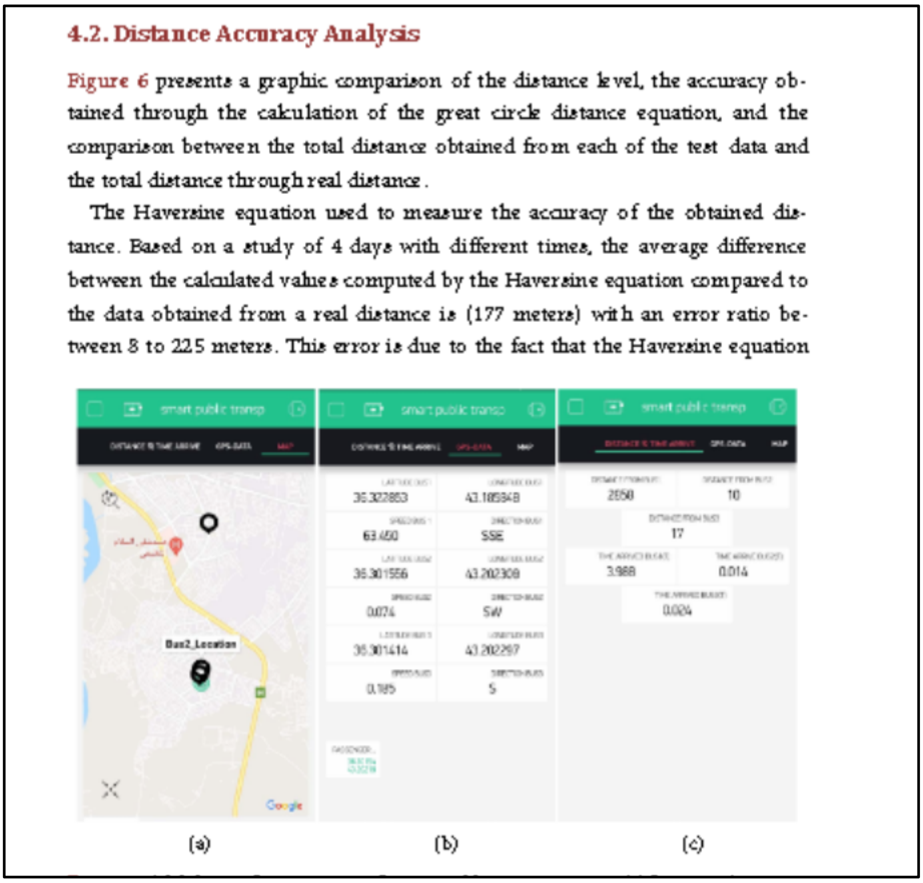
4.1. Location Accuracy Analysis

The proposed system has been tested and Table 2. Display data obtained from several streets in the city of Mosul to determine the level of accuracy of the data obtained through GPS on several days and at different times.

4.2. Distance Accuracy Analysis

Figure 6 presents a graphic comparison of the distance level, the accuracy obtained through the calculation of the great circle distance equation, and the comparison between the total distance obtained from each of the test data and the total distance through real distance.

The Haversine equation used to measure the accuracy of the obtained distance. Based on a study of 4 days with different times, the average difference between the calculated values computed by the Haversine equation compared to the data obtained from a real distance is (177 meters) with an error ratio between 8 to 225 meters. This error is due to the fact that the Haversine equation



4.2. Distance Accuracy Analysis

Figure 6 presents a graphic comparison of the distance level, the accuracy obtained through the calculation of the great circle distance equation, and the comparison between the total distance obtained from each of the test data and the total distance through real distance.

The Haversine equation used to measure the accuracy of the obtained distance. Based on a study of 4 days with different times, the average difference between the calculated values computed by the Haversine equation compared to the data obtained from a real distance is (177 meters) with an error ratio between 8 to 225 meters. This error is due to the fact that the Haversine equation