Internet Of Things

Public transport optimization using IOT sensors

Phase 3:Development Part 1

Abstract:

The main role of this project is optimization of public transport by IOT Technology .such that integrating iot devices like gps sensor and esp32 in all the public vehicle in the city and connect to internet through wifi. Code esp32 with python by importing necessary libraries for calculating latitude and longitude information from the gps sensor then the esp32 sensor made a Http post request to the server to store latitude and longitude information to the real time database(i.e.,Firebase) .By this real time information in our website the user can see live location of vehicle,this 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.

gps based vehicle tracking system

Integration of Iot devices:

Connect the VCC pin of the GPS module to the ESP32 3.3V pin.

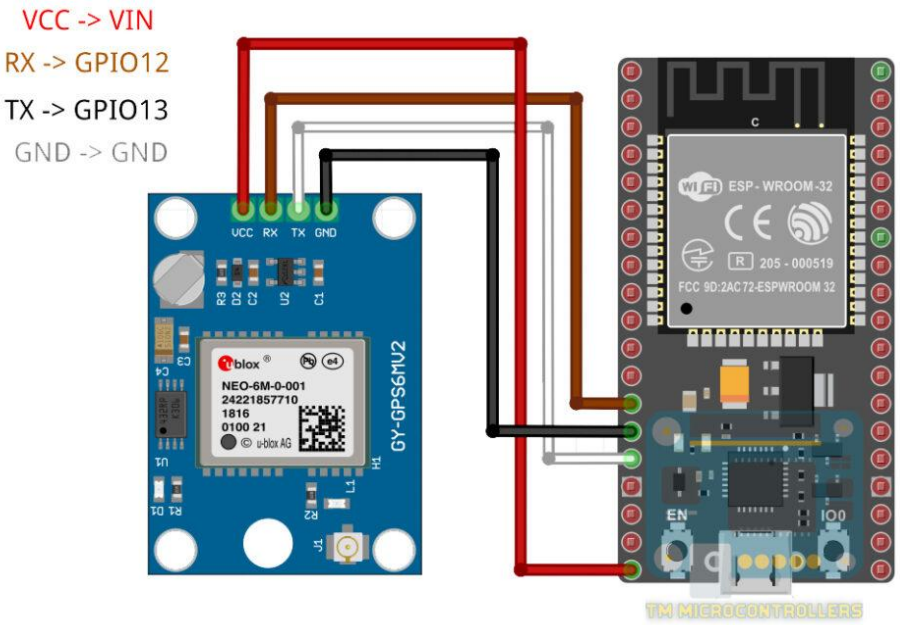
1. Connect the GPS ground pin to the ESP32 ground pin.
2. Connect the RX pin of the GPS module to the TX pin of the ESP32.
3. Connect your ESP32 to the computer through a USB cable.
4. Program the ESP32 via Arduino IDE with python to calculate latitude and longitude information from gps data .
5. Integrate this setup to public vehicle.
6. Send the GPS data, read by the ESP32 from the GPS device, to an external web server to store it on the real time database(Firebase).

Here is a sample code that uses MicroPython to interface an ESP32 with a NEO-6M GPS module and obtain GPS parameters such as latitude, longitude, altitude, date, time, speed, satellites, etc. The code also shows how to store the obtained data on Firebase.

PROGRAM:

```
# Import required libraries
import machine
import time
import network
import urequests as requests
from machine import UART

# Set up the UART interface for GPS module
uart = UART(2, baudrate=9600, tx=17, rx=16)

# Set up the Wi-Fi connection
ssid = 'your_wifi_ssid'
password = 'your_wifi_password'
station = network.WLAN(network.STA_IF)
station.active(True)
station.connect(ssid, password)

# Set up the Firebase database URL and authentication token
url =
'https://your_firebase_database_url.firebaseio.com/your_database_name.json'
auth_token = 'your_firebase_auth_token'

# Define a function to obtain GPS data from the NEO-6M module
def get_gps_data():
    while True:
        # Read the GPS data from the UART interface
        gps_data = uart.readline()
        if gps_data.startswith(b'$GPGGA'):
            # Parse the GPS data to obtain latitude and longitude
            gps_data_list = gps_data.split(b',')
            latitude = gps_data_list[2]
            longitude = gps_data_list[4]
            # Return the latitude and longitude as strings
            return latitude.decode('utf-8'), longitude.decode('utf-8')

# Define a function to store the GPS data on Firebase
def store_gps_data_on_firebase(latitude, longitude):
    # Create a dictionary with the GPS data
    gps_data_dict = {'latitude': latitude, 'longitude': longitude}
    # Send a POST request to Firebase to store the GPS data
    response = requests.post(url + '?auth=' + auth_token, json=gps_data_dict)
    # Print the response from Firebase
    print(response.text)
```

```
# Obtain GPS data and store it on Firebase every 10 seconds
while True:
    # Wait for 10 seconds before obtaining and storing GPS data again
    time.sleep(10)
    # Obtain GPS data from the NEO-6M module
    latitude, longitude = get_gps_data()
    # Store the GPS data on Firebase
    store_gps_data_on_firebase(latitude, longitude)
```



Now every 10 ESP32 sensor calculate latitude and longitude information and send post request to web server to store this information to real time database (Firebase).

Firebase is a product of Google which helps developers to build, manage, and grow their apps easily. It helps developers to build their apps faster and in a more secure way. No programming is required on the firebase side which makes it

easy to use its features more efficiently. It provides services to web applications

Create a database in firebase for our project store the latitude and longitude data into it ,every 10 seconds this data will be updated by new data come from esp32 sensors integrated with each of the vehicle in the city.Next step is display the live location of vehicle the user need to see in the website.the location of vehicle ,arrival time and the velocity of the vehicle will be accurate so that we go for loading and preprocessing the datasets stored in the firebase.

Loading and Preprocessing of datasets:

Data preprocessing is a process of preparing the raw data and making it suitable for a machine learning model. It is the first and crucial step while creating a machine learning model.

When creating a machine learning project, it is not always a case that we come across the clean and formatted data. And while doing any operation with data, it is mandatory to clean it and put in a formatted way. So for this, we use data preprocessing task.In our case we collect data from firebase and then preprocessing the location data to show accurate location of the vehicle to the user.

(i)Get the dataset:

Firstly get the dataset from the firebase database to preprocess it,

1.Connect to Firebase:

First, you need to establish a connection to your Firebase database. You can use the pyrebase package in Python for this.

```
import pyrebase

config = {
  "apiKey": "your-api-key",
  "authDomain": "your-auth-domain",
  "databaseURL": "your-database-url",
  "storageBucket": "your-storage-bucket"
}

firebase = pyrebase.initialize_app(config)
db = firebase.database()
```

2.Load the data:

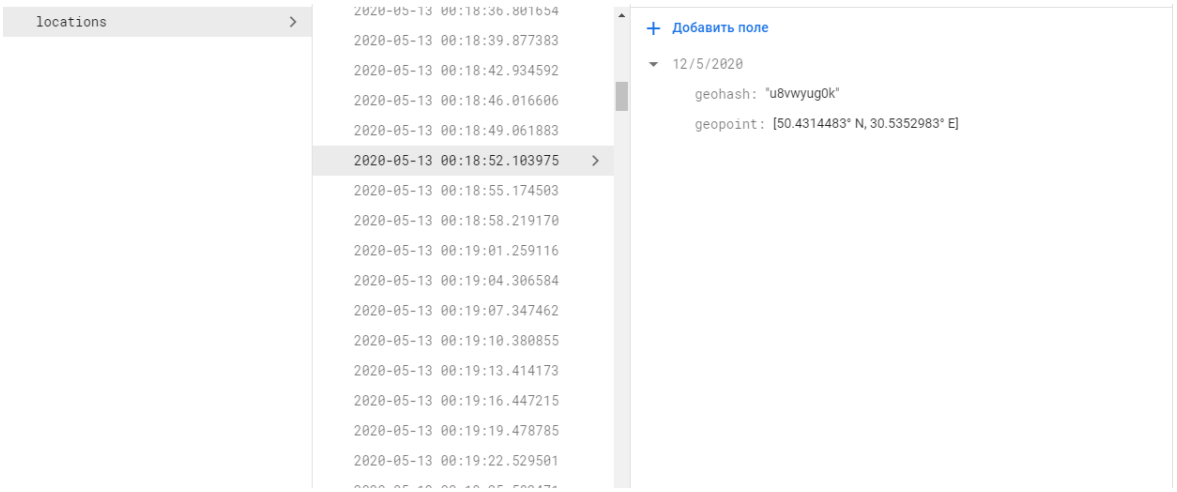
Fetch the data from the Firebase database.

```
data = db.child("your-database-child").get().val()
```

3.Convert to DataFrame:

Convert the data into a pandas DataFrame for easier manipulation.

```
import pandas as pd

df = pd.DataFrame(data)
```



This is the dataset of single node which store latitude and longitude as geopoint every four seconds it will update it.

(ii)Importiing Libraries:

Install the geopy library, which is a popular package for geocoding and reverse geocoding in Python. Geocoding is the process of converting addresses into geographical coordinates, and reverse geocoding is the opposite. You can install geopy using the command in the terminal

```
pip install geopy
```

Install the lat-lon-parser library, which is a package for parsing lat-long coordinates in various formats, and for converting between lat-long formats (e.g. decimal degrees to degrees-minutes-seconds). You can install lat-lon-parser using the command

```
pip install lat-lon-parser
```

Now you need to import the modules from these libraries in your Python script. For example, you can use the following lines of code to import the Nominatim geocoder from geopy and the parse function from lat-lon-parser:

```
from geopy.geocoders import Nominatim
from lat_lon_parser import parse
```

(iii)Preprocess dataset:

Inspect the data:Check the first few rows of your data to understand its structure. Also, check if there are any missing values in the latitude and longitude columns.

```
print(df.head())
print(df.isnull().sum())
```

Handling missing values:Preprocess the dataset is to handle missing data in the datasets. If our dataset contains some missing data, then it may create a huge problem for our machine learning model. Hence it is necessary to handle missing values present in the dataset.
There are mainly two ways to handle missing data, which are:
(i)By deleting a particular row.

If the number of rows with missing values is small, you might opt to remove these rows.

```
df = df.dropna(subset=['latitude', 'longitude'])
```

(ii)By calaculating the missing data with the help of present data.

If you can’t afford to lose data, you might decide to fill the missing values with some value. This could be the mean or median of the available values, or some other strategy.

```
df['latitude'].fillna(df['latitude'].mean(), inplace=True)
df['longitude'].fillna(df['longitude'].mean(), inplace=True)
```

The missing of data occurs due to the break of internet connection of particular node in very small duration .so that maintain unbreakable of internet is necessary.

Now with this data we can track the live location of the vehicle in our web application .