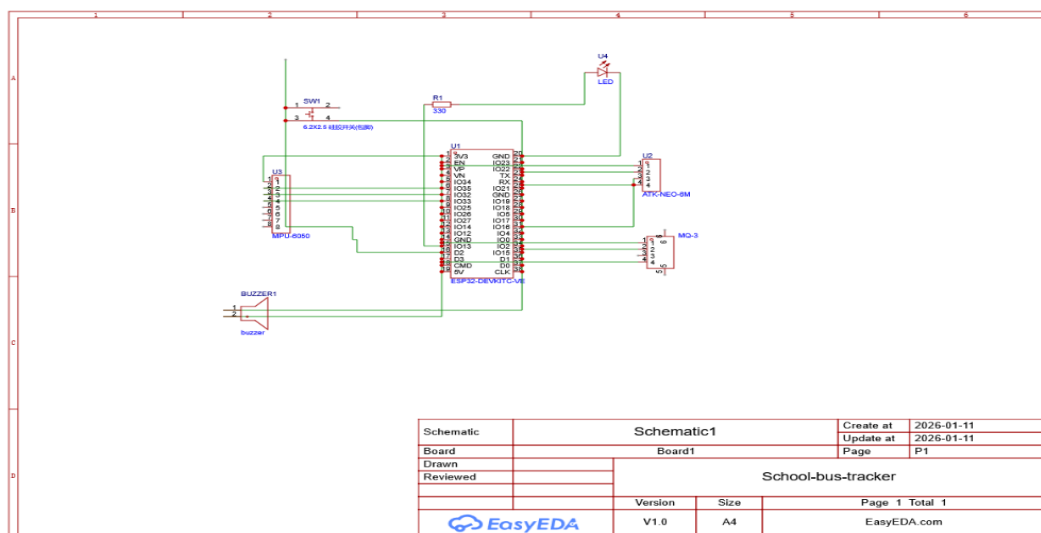


1. Αρχιτεκτονική Συστήματος

Η προτεινόμενη αρχιτεκτονική ακολουθεί το μοντέλο των κυβερνοφυσικών συστημάτων, συνδυάζοντας το φυσικό υλικό με το λογισμικό cloud.

Το σύστημα αποτελείται από τρία βασικά στάδια:

- Επίπεδο συλλογής δεδομένων: Χρησιμοποιείται ο μικροελεγκτής ESP 32, ο οποίος συνδέεται με αισθητήρες GPS, MPU 60-50 (επιταχυνσιόμετρο) και MQ-3 (ανίχνευση αλκοόλ).
- Επίπεδο επικοινωνίας: Η μεταφορά των δεδομένων γίνεται σε πραγματικό χρόνο μέσω του πρωτοκόλλου MQTT, το οποίο εξασφαλίζει χαμηλή κατανάλωση πόρων και ταχύτητα.
- Επίπεδο εφαρμογής: Το Backend λαμβάνει τα δεδομένα και τα διοχετεύει σε web εφαρμογή βασισμένη σε Next.js, όπου οι γονείς και η διεύθυνση βλέπουν τη θέση του λεωφορείου σε πραγματικό χρόνο.



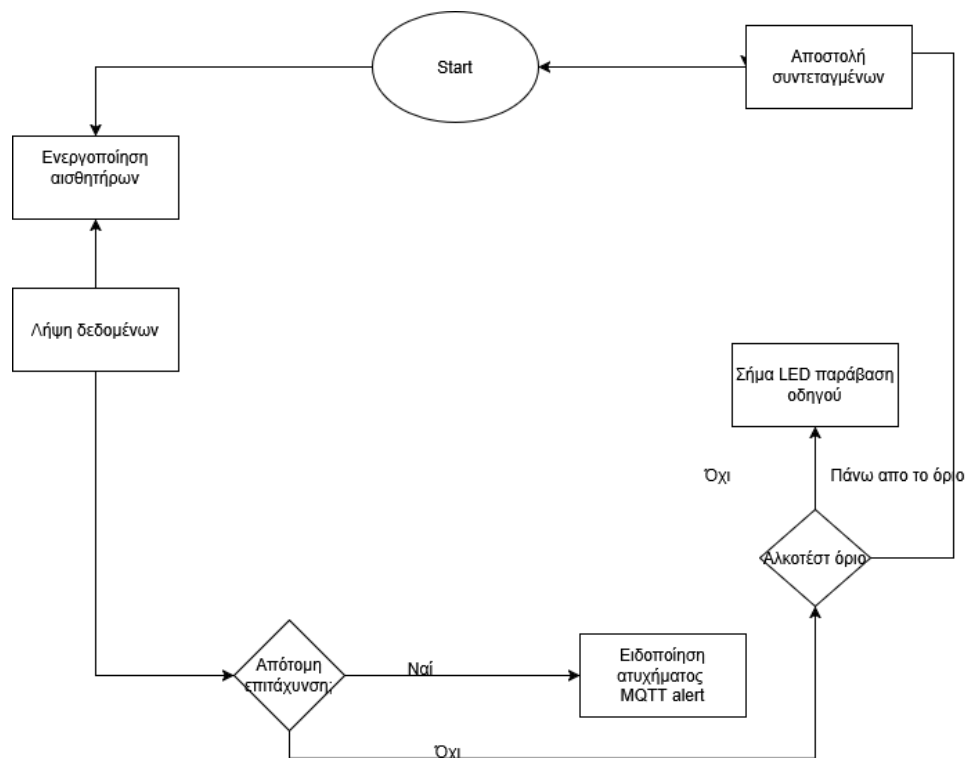
Σχήμα 1: Στο παραπάνω σχήμα απεικονίζεται η συνδεσμολογία των εξαρτημάτων.

2. Εξοπλισμός

- ESP 32-DevKitC-VE (Μικροελεγκτής): Μικροελεγκτής υψηλών επιδόσεων με ενσωματωμένο WI-FI και Bluetooth, υπεύθυνος για την επεξεργασία των δεδομένων και την επικοινωνία μέσω MQTT.
- ATK-NEO-6M (GPS) : Αισθητήρας εντοπισμού θέσης, παρέχει γεωγραφικές συντεταγμένες και ταχύτητα σε πραγματικό χρόνο.
- MPU-6050 (Accelerometer): Αισθητήρας κίνησης και επιτάχυνσης έξι αξόνων για την ανίχνευση απότομων μεταβολών και πιθανού ατυχήματος.
- MQ-3 (Alcohol sensor): Αναλογικός αισθητήρας για την ανίχνευση συγκέντρωσης αλκοόλ στον αέρα της καμπίνας του οδηγού.

- Buzzer (Ηχητική μονάδα) : Άμεση προειδοποίηση του οδηγού σε περίπτωση ατυχήματος ή έκτακτης ανάγκης.
- LED & RESISTOR (330Ω) : Μονάδες οπτικής ένδειξης για την κατάσταση λειτουργίας και την επισήμανση παραβάσεων.
- Push Button (Διακόπτης) : Διακόπτης χειροκίνητης παρέμβασης, για ακύρωση συναγερμού σε περίπτωση εσφαλμένης ενεργοποίησης.

3. Διαδικτυακή εφαρμογή (FASTAPI & Next.js)



Σχήμα 2: Λογικό διάγραμμα ροής

Κύριες δομές και βιβλιοθήκες :

Λογισμικό μονάδας :

- Paho-MQTT (Client) : Χρησιμοποιείται για τη σύνδεση με το Broker (test.mosquitto.org) και τη δημοσίευση των δεδομένων σε μορφή αρχείου JSON.
- Βιβλιοθήκη JSON : Μετατρέπει τα δεδομένα σε τυποποιημένη μορφή για εύκολη ανάγνωση από το Backend.

Διαδουκτιακή εφαρμογή (FASTAPI & Next.js) :

- FastAPI framework : Διαχειρίζεται τις εισερχόμενες κλήσεις API και διατηρεί στη μνήμη την τελευταία γνωστή κατάσταση του λεωφορείου.
- CORSMiddleware : Επιτρέπει την ασφαλή επικοινωνία μεταξύ του Frontend (Next.js) και του Backend (FastAPI).

- **React Leaflet** : Βιβλιοθήκη για την απόδοση του διαδραστικού χάρτη OpenStreetMap και την ενημέρωση του marker στο χάρτη.
- **Fetch API** : Λειτουργία που ανανεώνει τα δεδομένα στην οθόνη του χρήστη κάθε ένα δευτερόλεπτο, χωρίς επαναφόρτωση της σελίδας.

4.Πρόγραμμα

Η υλοποίηση του λογισμικού χωρίζεται σε τρία διακριτά μέρη:

- Προσομοιωτή δεδομένων
- Το παρασκήνιο (Backend)
- Διεπαφή χρήστη (Frontend)

GitHub repository: <https://github.com/AlexNT-maker/school-bus-tracker>

Simulator.py:

```
# Προσομοίωση κίνησης, ταχύτητας και ατυχήματος
while (True):
    bus_data["latitude"] += random.uniform(-0.0001, 0.0001)
    bus_data["longitude"] += random.uniform(-0.0001, 0.0001)
    bus_data["speed"] = round(random.uniform(0 , 95), 1)
    # 2% πιθανότητα ανίχνευσης ατυχήματος (MPU-6050 Logic)
    if random.randint(0,100) > 98 : bus_data["accident"] = True
    else: bus_data["accident"] = False
    # Μετατροπή σε JSON και αποστολή μέσω MQTT Broker
    payload = json.dumps(bus_data)
    client.publish(TOPIC, payload)
    time.sleep(1)
```

main.py:

```
# Η συνάρτηση που εκτελείται κάθε φορά που λαμβάνεται νέο στίγμα
def on_message(client, userdata, msg):
    global latest_data
    try:
```

```
# Αποκωδικοποίηση μηνύματος και αποθήκευση στη μεταβλητή latest_data
payload = msg.payload.decode("utf-8")
latest_data = json.loads(payload)
except Exception as e:
    print(f'Error decoding message: {e}')
# API Endpoint που καλείται από την εφαρμογή Next.js
@app.get("/api/bus-location")
def get_bus_location():
    """Επιστρέφει το τελευταίο γνωστό στίγμα στην εφαρμογή"""
    return latest_data
```

*Αποτυπώνονται τα κρίσιμα κομμάτια του κώδικα python που αφορούν τη λογική του συστήματος.
Ολοκληρωμένη εφαρμογή με οδηγίες και README στο αποθετήριο.

GitHub repository: <https://github.com/AlexNT-maker/school-bus-tracker>