



ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ  
ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ  
ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ  
ΤΟΜΕΑΣ ΕΠΙΚΟΙΝΩΝΙΩΝ, ΗΛΕΚΤΡΟΝΙΚΗΣ ΚΑΙ  
ΣΥΣΤΗΜΑΤΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ

**Υλοποίηση Συστήματος Τηλεμετρίας για  
Διαγνωστικά Οχήματος**

**ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ**

Γρηγόριος Δ. Ζήρας

**Επιβλέπων :** Ευστάθιος Δ. Συκάς  
Καθηγητής Ε.Μ.Π.

Αθήνα, Ιούλιος 2016





ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ  
ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ  
ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ  
ΤΟΜΕΑΣ ΕΠΙΚΟΙΝΩΝΙΩΝ, ΗΛΕΚΤΡΟΝΙΚΗΣ ΚΑΙ  
ΣΥΣΤΗΜΑΤΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ

## Υλοποίηση Συστήματος Τηλεμετρίας για Διαγνωστικά Οχήματος

### ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

Γρηγόριος Δ. Ζήρας

**Επιβλέπων :** Ευστάθιος Δ. Συκάς  
Καθηγητής Ε.Μ.Π.

Εγκρίθηκε από την τριμελή εξεταστική επιτροπή την 1<sup>η</sup> Ιουλίου 2016.

.....  
Ε.Δ. Συκάς  
Καθηγητής Ε.Μ.Π.

.....  
Γ. Ι. Στασινόπουλος  
Καθηγητής Ε.Μ.Π.

.....  
Μ.Ε. Θεολόγου  
Καθηγητής Ε.Μ.Π.

Αθήνα, Ιούλιος 2016

.....  
Γρηγόριος Δ. Ζήρας

Διπλωματούχος Ηλεκτρολόγος Μηχανικός και Μηχανικός Υπολογιστών Ε.Μ.Π.

Copyright © Γρηγόριος Ζήρας, Έτος 2016.

Με επιφύλαξη παντός δικαιώματος. All rights reserved.

Απαγορεύεται η αντιγραφή, αποθήκευση και διανομή της παρούσας εργασίας, εξ ολοκλήρου ή τμήματος αυτής, για εμπορικό σκοπό. Επιτρέπεται η ανατύπωση, αποθήκευση και διανομή για σκοπό μη κερδοσκοπικό, εκπαιδευτικής ή ερευνητικής φύσης, υπό την προϋπόθεση να αναφέρεται η πηγή προέλευσης και να διατηρείται το παρόν μήνυμα. Ερωτήματα που αφορούν τη χρήση της εργασίας για κερδοσκοπικό σκοπό πρέπει να απευθύνονται προς τον συγγραφέα.

Οι απόψεις και τα συμπεράσματα που περιέχονται σε αυτό το έγγραφο εκφράζουν τον συγγραφέα και δεν πρέπει να ερμηνευθεί ότι αντιπροσωπεύουν τις επίσημες θέσεις του Εθνικού Μετσόβιου Πολυτεχνείου.



# Περίληψη

Τα Ευφυή Συστήματα Μεταφορών (Intelligent Transport Systems-ITS) είναι ένας συνδυασμός τεχνολογιών πληροφόρησης και επικοινωνιών εφαρμοσμένων στον τομέα των μεταφορών με στόχο την αποδοτικότερη, ασφαλέστερη και οικονομικότερη κυκλοφορία των ατόμων ή των εμπορευμάτων κάνοντας χρήση νέων τεχνολογιών.

Η παρούσα διπλωματική υπάγεται στο ευρύτερο πεδίο των Ευφυών Συστημάτων Μεταφοράς και σκοπός της είναι η ανάπτυξη ενός αυτόνομου συστήματος καταγραφής δεδομένων οχήματος. Η υλοποίηση βασίστηκε στην ενοποίηση και ολοκλήρωση των ακόλουθων υποσυστημάτων: ένα Raspberry Pi, ένα OBD-II adapter, GPS dongle, ένα επιταχυνσιόμετρο και ένα WiFi adapter ή 4G modem.

Συγκεκριμένα, στο Raspberry Pi εκτελείται εφαρμογή η οποία συλλέγει και αποθηκεύει δεδομένα οχήματος (π.χ. ταχύτητα, κατανάλωση κ.ά.), θέσης και επιτάχυνσης μέσω του OBD-II adapter, του GPS dongle και του επιταχυνσιόμετρου αντίστοιχα. Τα δεδομένα αυτά διατίθενται προς επεξεργασία σε απομακρυσμένο εξυπηρετητή, όπου αντιστοιχίζονται επακριβώς στο χάρτη (map-matching λειτουργία) μέσω του προγράμματος Barefoot.

Ακολούθως, από το σύνολο των δεδομένων που προκύπτουν από την εφαρμογή και το map-matching εξάγονται στατιστικά στοιχεία ανά οδικό σύνδεσμο τα οποία αποθηκεύονται σε βάση δεδομένων ενώ παρέχεται και η δυνατότητα για αυτοματοποιημένη απεικόνιση των εξαγόμενων δεδομένων στο χάρτη.

**Λέξεις Κλειδιά:** Ευφυή Συστήματα Μεταφορών, OBD, map-matching, GPS, Python, επιταχυνσιόμετρο, PostgreSQL, Τηλεμετρία, Raspberry Pi



# Abstract

Intelligent Transport Systems (ITS) are a combination of Information and Communications Technologies (ICT) applied to the field of transportation in order to achieve more efficient, safer and cheaper circulation of people and goods through the use of new technologies.

This dissertation falls in the broad context of ITS and aims at developing an autonomous system of recording vehicle data. The implementation plan was based upon the deployment and integration of the following on-board sub-systems: a Raspberry Pi single-board computer, an OBD-II adapter, a GPS dongle, an accelerometer and a Wi-Fi adapter or 4G modem.

In more detail, the Raspberry Pi executes an application that collects and stores vehicle data (e.g. speed, fuel consumption etc.), position and acceleration figures through the OBD-II adapter, the GPS dongle and the accelerometer, respectively. These data are then uploaded and made available for processing in a remote server, where they are accurately processed and fused with electronic map data (map-matching process) using Barefoot software.

Subsequently, from the volume of data resulting from the on-board and off-board components, statistical per-road-link data are extracted and stored in a centralized database, while at the same time the functionality of automated in-map visualization of the accumulated information is provided.

**Keywords:** Intelligent transportation systems, OBD, map-matching, GPS, accelerometer, Python, PostgreSQL, Telemetry, Raspberry Pi



# Ευχαριστίες

Αρχικά, θα ήθελα να ευχαριστήσω τον καθηγητή κ. Ευστάθιο Συκά για την εμπιστοσύνη που μου έδειξε και την ευκαιρία που μου έδωσε να ασχοληθώ με ένα τόσο ενδιαφέρον θέμα.

Στη συνέχεια, θα ήθελα να ευχαριστήσω την οικογένεια μου και τους φίλους που με υποστηρίζουν όλα αυτά τα χρόνια.

Τέλος, οφείλω ένα μεγάλο ευχαριστώ στον υποψήφιο διδάκτορα Βασίλη Ασθενόπουλο για τη συνεχή καθόγηση και την ανεκτίμητη βοήθεια που μου παρείχε κατά τη διάρκεια συγγραφής της παρούσας διπλωματικής εργασίας.



# Πίνακας Περιεχομένων

Πίνακας Διαγραμμάτων .....	4
Πίνακας Εικόνων .....	5
<b>1. Εισαγωγή .....</b>	<b>6</b>
1.1 Ευφυή Συστήματα Μεταφορών .....	6
1.2 Αντικείμενο της διπλωματικής.....	7
1.3 Οργάνωση της διπλωματικής.....	7
<b>2. Περιγραφή του προβλήματος.....</b>	<b>9</b>
2.1 Τηλεμετρία .....	9
2.2 Μικροεπεξεργαστές - Ενσωματωμένα Υπολογιστικά Συστήματα .....	9
2.3 Περιγραφή της διπλωματικής .....	11
2.4 Προδιαγραφές Απαιτήσεων .....	12
<b>3. Τεχνολογίες Hardware .....</b>	<b>14</b>
3.1 Raspberry Pi .....	15
3.2 Η διεπαφή OBD-II .....	17
3.2.1 Η θύρα OBD-II .....	18
3.2.2 ELM327 OBD Adapter.....	20
3.2.3 Bluetooth.....	21
3.3 Επιταχυνσιόμετρο .....	22
3.3.1 Microstack Accelerometer .....	22
3.3.2 I2C.....	23
3.4 GPS.....	24
3.4.1 Microstack GPS .....	25
3.4.2 SPI.....	25
3.5 WLAN adapter - Ασύρματα Τοπικά Δίκτυα.....	27
3.5.1 Λειτουργία Tethering.....	28
3.5.2 Δίκτυα 4G – LTE .....	28

<b>4. Τεχνολογίες-Software .....</b>	<b>30</b>
4.1 Python.....	31
4.2 Raspbian.....	34
4.3 OpenStreetMap.....	35
4.4 Map-Matching και Barefoot.....	36
4.4.1 Docker.....	38
4.5 Okeanos.....	38
4.6 SSH.....	40
4.6.1 Ιδιωτικά Και Δημόσια Κλειδιά .....	40
4.6.2 SCP .....	41
4.7 Git.....	42
4.8 PostgreSQL .....	42
4.9 geojson.io .....	43
<b>5. Αρχιτεκτονική .....</b>	<b>45</b>
5.1 Σχεδιαστικές Επιλογές .....	47
5.2 Πελάτης - Raspberry Pi.....	50
5.2.1 OBD adapter .....	50
5.2.2 GPS Dongle .....	51
5.2.3 Επιταχυνσιόμετρο (Accelerometer).....	52
5.3 Εξυπηρετητής – Virtual Machine@okeanos.grnet.gr .....	55
5.3.1 Map Matching Server – Barefoot .....	55
5.3.2 Database Server – PostgreSQL.....	58
5.3.3 Απεικόνιση στο χάρτη – geojson.io.....	59
<b>6. Σενάρια Χρήσης - Μελλοντικές Επεκτάσεις .....</b>	<b>62</b>
6.1 Σενάρια Χρήσης.....	63
6.1.1 Διαχείριση Στόλου – Fleet Management .....	63
6.1.2 Usage-Based Insurance .....	64

6.1.3	Συστήματα Τηλεματικής.....	65
6.2	Μελλοντικές Επεκτάσεις.....	66
6.2.1	Αναγνώριση σχέσης στο κιβώτιο ταχυτήτων .....	66
6.2.2	Προφίλ οδηγικής συμπεριφοράς (Driver profiling).....	67
6.2.3	Online (Real-time) system .....	68
6.2.4	Λοιπές μελλοντικές προεκτάσεις.....	68
<b>Παράρτημα .....</b>		<b>69</b>
<b>Βιβλιογραφία/ Παραπομπές .....</b>		<b>72</b>

## Πίνακας Διαγραμμάτων

ΔΙΑΓΡΑΜΜΑ 1: ΔΟΜΙΚΟ ΔΙΑΓΡΑΜΜΑ ΤΟΥ RASPBERRY PI 2 MODEL B .....	16
ΔΙΑΓΡΑΜΜΑ 2: ΔΟΜΙΚΟ ΔΙΑΓΡΑΜΜΑ ΤΗΣ ΘΥΡΑΣ OBD .....	18
ΔΙΑΓΡΑΜΜΑ 3: ΙΖC ΣΥΝΔΕΣΗ ΓΙΑ ΤΟ ΕΠΙΤΑΧΥΝΣΙΟΜΕΤΡΟ .....	23
ΔΙΑΓΡΑΜΜΑ 4: SPI ΣΥΝΔΕΣΗ ΓΙΑ ΤΟ GPS .....	26
ΔΙΑΓΡΑΜΜΑ 5: ΠΑΡΑΔΕΙΓΜΑ OSM ΧΑΡΤΗ .....	36
ΔΙΑΓΡΑΜΜΑ 6: ΔΙΑΓΡΑΜΜΑ ΑΡΧΙΤΕΚΤΟΝΙΚΗΣ ΤΟΥ BAREFOOT .....	37
ΔΙΑΓΡΑΜΜΑ 7: ΟΚΕΑΝΟΣ SERVER .....	39
ΔΙΑΓΡΑΜΜΑ 8: ΔΙΑΓΡΑΜΜΑ ΑΡΧΙΤΕΚΤΟΝΙΚΗΣ ΣΥΣΤΗΜΑΤΟΣ .....	46
ΔΙΑΓΡΑΜΜΑ 9: ΥΠΟΣΤΗΡΙΖΟΜΕΝΕΣ ΕΝΤΟΛΕΣ ΑΠΟ ΤΗΝ ΡΥΤΗΘΝ-OBD ΒΙΒΛΙΟΘΗΚΗ .....	47
ΔΙΑΓΡΑΜΜΑ 10: ΠΑΡΑΔΕΙΓΜΑ ΑΡΧΕΙΟΥ ΚΑΤΑΓΡΑΦΗΣ .....	54
ΔΙΑΓΡΑΜΜΑ 11: ΠΙΝΑΚΑΣ ΒΑΣΗΣ ΔΕΔΟΜΕΝΩΝ .....	59
ΔΙΑΓΡΑΜΜΑ 12: ΑΠΕΙΚΟΝΙΣΗ ΚΑΤΑΓΡΑΦΗΣ ΣΤΟ ΧΑΡΤΗ ΜΕ GEOJSON.IO .....	60
ΔΙΑΓΡΑΜΜΑ 13: ΔΙΑΓΡΑΜΜΑ ΓΙΑ ΤΟΝ ΕΝΤΟΠΙΣΜΟ ΤΗΣ ΣΧΕΣΗΣ ΣΤΟ ΚΙΒΩΤΙΟ ΤΑΧΥΤΗΤΩΝ .....	66

# Πίνακας Εικόνων

ΕΙΚΟΝΑ 1: ΕΥΦΥΗ ΣΥΣΤΗΜΑΤΑ ΜΕΤΑΦΟΡΩΝ .....	6
ΕΙΚΟΝΑ 2: ΤΟ RASPBERRY PI ΣΥΝΔΕΔΕΜΕΝΟ ΜΕ ΤΟ GPS ΚΑΙ ΤΟ ΕΠΙΤΑΧΥΝΣΙΟΜΕΤΡΟ .....	14
ΕΙΚΟΝΑ 3: ΤΟ RASPBERRY PI MODEL B .....	15
ΕΙΚΟΝΑ 4: ΤΟ RASPBERRY PI ZERO .....	15
ΕΙΚΟΝΑ 5: Η ΘΥΡΑ OBD-II ΣΕ GOLF 4 .....	17
ΕΙΚΟΝΑ 6: ELM327 OBD ADAPTER .....	20
ΕΙΚΟΝΑ 7: BLUETOOTH.....	21
ΕΙΚΟΝΑ 8: BLUETOOTH ADAPTER.....	21
ΕΙΚΟΝΑ 9: ΤΟ ΕΠΙΤΑΧΥΝΣΙΟΜΕΤΡΟ ΤΗΣ MICROSTACK .....	22
ΕΙΚΟΝΑ 10: ΤΟ GPS ΤΗΣ MICROSTACK .....	25
ΕΙΚΟΝΑ 11: WLAN ADAPTER.....	27
ΕΙΚΟΝΑ 12: Η ΕΠΙΦΑΝΕΙΑ ΕΡΓΑΣΙΑΣ ΤΟΥ RASPBIAN .....	30
ΕΙΚΟΝΑ 13: GEOJSON.IO.....	43
ΕΙΚΟΝΑ 14: ΚΟΜΒΟΣ ΣΤΟΝ OSM ΧΑΡΤΗ.....	58
ΕΙΚΟΝΑ 15: ΠΑΡΑΔΕΙΓΜΑ ΜΑΡ-MATCHING .....	61
ΕΙΚΟΝΑ 16: PI ON THE CAR.....	62

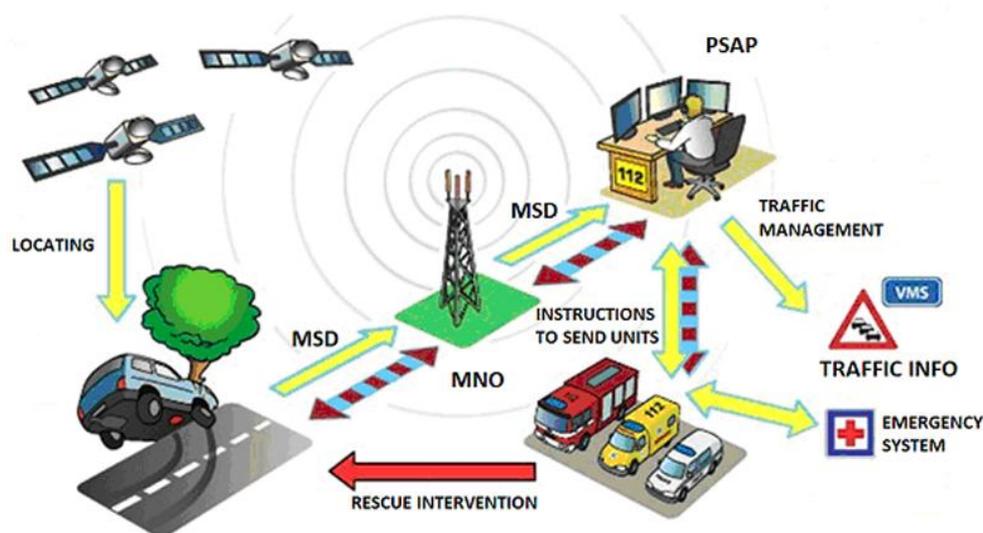
# 1. Εισαγωγή

## 1.1 Ευφυή Συστήματα Μεταφορών

Τα Ευφυή Συστήματα Μεταφορών (Intelligent Transport Systems-ITS) είναι προηγμένες εφαρμογές οι οποίες έχουν ως στόχο την παροχή καινοτόμων υπηρεσιών που σχετίζονται με διαφορετικούς τρόπους μεταφοράς και διαχείρισης της κυκλοφορίας και επιτρέπουν στους διάφορους χρήστες να ενημερώνονται καλύτερα και να κάνουν πιο ασφαλή, πιο συντονισμένη και “εξυπνότερη” χρήση των δικτύων μεταφοράς.

Τα ITS αποτελούν δυναμικό τομέα με έντονες αναπτυξιακές διαστάσεις, που συνδυάζει τις τεχνολογίες πληροφορικής και επικοινωνιών. Έχουν εφαρμογή σε οδικά, σιδηροδρομικά, θαλάσσια και εναέρια συστήματα μεταφορών και εκτός του ρόλου που έχουν στην ενίσχυση της ανάπτυξης και της εγχώριας οικονομίας, τα ITS κατέχουν επίσης σημαντικό κοινωνικό ρόλο, σώζοντας ζωές και εξοικονομώντας χρόνο και χρήμα με:

- την πρόληψη τροχαίων ατυχημάτων
- την βελτίωση της οδικής ασφάλειας
- την μείωση της συμφόρησης, της κατανάλωσης ενέργειας και των εκπομπών αερίων θερμοκηπίου
- την βελτίωση της συνολικής απόδοσης των συστημάτων μεταφορών
- και την δημιουργία φιλικών και βιώσιμων κοινοτήτων



Εικόνα 1: Ευφυή Συστήματα Μεταφορών

## 1.2 Αντικείμενο της διπλωματικής

Η παρούσα διπλωματική έχει ως αντικείμενο την ανάπτυξη ενός αυτόνομου συστήματος τηλεμετρίας το οποίο καταγράφει δεδομένα οχήματος. Η ανωτέρω διαδικασία υλοποιείται με την ενοποίηση και ολοκλήρωση των ακόλουθων υποσυστημάτων:

- Ένα Raspberry Pi που λειτουργεί ως φιλοξενούν μηχανήμα (host computer).
- Ένα GPS dongle για την καταγραφή δεδομένων θέσης.
- Ένα επιταχυνσιόμετρο για την καταγραφή δεδομένων επιτάχυνσης.
- Ένα OBD adapter για την καταγραφή μετρήσεων πραγματικού χρόνου, όπως ταχύτητα και στροφές κινητήρα από τη θύρα OBD του οχήματος.

Συγκεκριμένα, στο Raspberry Pi εκτελείται εφαρμογή η οποία συλλέγει και αποθηκεύει δεδομένα οχήματος (π.χ. ταχύτητα, κατανάλωση κ.ά.), θέσης και επιτάχυνσης μέσω του OBD-II adapter, του GPS dongle και του επιταχυνσιόμετρου αντίστοιχα. Τα δεδομένα αυτά διατίθενται προς επεξεργασία σε απομακρυσμένο εξυπηρετητή, ο οποίος επιτελεί τη λειτουργία του map-matching και αποθηκεύει στατιστικά στοιχεία ανά οδικό σύνδεσμο σε βάση δεδομένων. Τέλος, το σύνολο των δεδομένων απεικονίζεται αυτοματοποιημένα σε χάρτη.

## 1.3 Οργάνωση της διπλωματικής

Η περιγραφή καθώς και οι προδιαγραφές που θα πρέπει να πληρεί η υλοποίηση της παρούσας διπλωματική εργασίας παρουσιάζονται στο κεφάλαιο 2.

Τα κεφάλαιο 3 και 4 είναι αφιερωμένα στις τεχνολογίες που χρησιμοποιήθηκαν κατά την ανάπτυξη της εφαρμογής. Αυτές χωρίζονται σε δύο μεγάλες κατηγορίες, τις τεχνολογίες Software και τεχνολογίες Hardware, ενώ περιγράφονται και τα σχετικά με αυτές πρωτόκολλα επικοινωνίας.

Στο κεφάλαιο 5 παρουσιάζουμε την αρχιτεκτονική του συστήματος, τις σχεδιαστικές επιλογές και περιγράφουμε αναλυτικά τις λειτουργίες που επιτελεί κάθε υποσύστημα της εφαρμογής μας.

Στο κεφάλαιο 6 αναφερόμαστε στους τομείς τους οποίους μπορεί να βρει απήχηση η παρούσα διπλωματική, καθώς και πως μπορεί αυτή να επεκταθεί μελλοντικά.

Τέλος, στο Παράρτημα παρουσιάζεται ο κώδικας για την καταγραφή των μετρήσεων στο Raspberry Pi.

## 2. Περιγραφή του προβλήματος

### 2.1 Τηλεμετρία

Με τον ορο τηλεμετρία εννοείται η επιστήμη που επιτρέπει την συλλογή δεδομένων εξ αποστάσεως. Πιο αναλυτικά, στον όρο τηλεμετρία περιλαμβάνεται συνήθως η ασύρματη μετάδοση δεδομένων με χρήση πομποδεκτών μεγάλης ή μικρής εμβέλειας, τηλεπικοινωνιακών δορυφόρων κλπ, αλλά και η καλωδιακή μετάδοση δεδομένων κυρίως σήμερα μέσω δικτύων όπως το ίντερνετ ή μέσω τηλεφωνικού δικτύου.

Πρόκειται για μία αυτοματοποιημένη διαδικασία επικοινωνίας μέσω της οποίας συλλέγονται δεδομένα, με τη χρήση ειδικού εξοπλισμού, από απομακρυσμένα ή δυσπρόσιτα σημεία και μεταδίδονται στην οθόνη του χρήστη.[7]

Η Τηλεμετρία βρίσκει εφαρμογή σε ένα ευρύ σύνολο τομέων, όπως:

- Δίκτυα Κοινής Ωφέλειας
- Μετεωρολογία
- Σεισμολογία
- Ιατρική
- Μεταφορές

Στην παρούσα διπλωματική, θα ασχοληθούμε με τον τομέα των μεταφορών και για τη συλλογή δεδομένων θα χρησιμοποιήσουμε ένα ενσωματωμένο υπολογιστικό σύστημα.

### 2.2 Μικροεπεξεργαστές - Ενσωματωμένα Υπολογιστικά Συστήματα

Ένας μικροεπεξεργαστής περιλαμβάνει τις περισσότερες ή όλες τις λειτουργίες μιας κεντρικής μονάδας επεξεργασίας (ΚΜΕ) ενός ηλεκτρονικού υπολογιστή σε ένα ενιαίο ολοκληρωμένο κύκλωμα (IC). Χρησιμοποιείται διότι είναι ένας πολύ αποδοτικός τρόπος υλοποίησης ψηφιακών συστημάτων και κάνει ευκολότερη τη σχεδίαση οικογενειών

προϊόντων τα οποία μπορούν να επεκταθούν ώστε να συμβαδίσουν με τις ραγδαία μεταβαλλόμενες αγορές.[4]

Ενσωματωμένο υπολογιστικό σύστημα είναι μια οποιαδήποτε συσκευή η οποία περιλαμβάνει έναν προγραμματιζόμενο υπολογιστή, ο οποίος όμως δεν είναι υπολογιστής γενικού σκοπού. Συνεπώς, ένας προσωπικός υπολογιστής (PC) δεν είναι ένα ενσωματωμένο υπολογιστικό σύστημα αλλά ένα κινητό τηλέφωνο ή ένα «έξυπνο» ρολόι που υλοποιείται από έναν μικροεπεξεργαστή αποτελεί ένα ενσωματωμένο υπολογιστικό σύστημα. Άλλα παραδείγματα ενσωματωμένων συστημάτων που συναντάται στην καθημερινή μας ζωή είναι ένα MP3 player, ένα ADSL modem, αλλά και μεγαλύτερου μεγέθους εγκαταστάσεις όπως οι φωτεινοί σηματοδότες. Εκτεταμένη χρήση ενσωματωμένων συστημάτων γίνεται και στο αυτοκίνητο. Η μηχανή, το σύστημα ελέγχου των φρένων, το σύστημα ABS είναι μερικά μονάχα παραδείγματα εντός του αυτοκινήτου που ελέγχονται από μικροεπεξεργαστές.

Μερικά κοινά χαρακτηριστικά των ενσωματωμένων συστημάτων είναι:

- *Χαμηλό κόστος κατασκευής:* Το συνολικό κόστος κατασκευής ενός συστήματος είναι πολύ σημαντικό και προσδιορίζεται από πολλούς παράγοντες συμπεριλαμβανομένου του τύπου του μικροεπεξεργαστή που χρησιμοποιείται, της ποσότητας της απαιτούμενης μνήμης και του τύπου των συσκευών εισόδου-εξόδου. Η πλειοψηφία των ενσωματωμένων συστημάτων είναι προϊόντα μαζικής αγοράς και πρέπει να έχουν χαμηλό κόστος κατασκευής.
- *Χαμηλή κατανάλωση ισχύος:* Η κατανάλωση ισχύος επηρεάζει άμεσα το κόστος του υλικού, εφόσον μια μεγαλύτερη πηγή τροφοδοσίας ισχύος μπορεί να είναι απαραίτητη. Αλλά η κατανάλωση ισχύος επηρεάζει και τη διάρκεια ζωής της μπαταρίας, όπως και την κατανάλωση θερμότητας.
- *Μικρό Μέγεθος και Βάρος:* Πολλά ενσωματωμένα συστήματα αποτελούν μικρά κομμάτια μεγαλύτερων συσκευών γενικού σκοπού επομένως το μέγεθος τους πρέπει να προσαρμόζεται στις απαιτήσεις της κεντρικής συσκευής. Επιπρόσθετα, μια συσκευή χειρός τυπικά έχει αυστηρές απαιτήσεις τόσο για το μέγεθος όσο και για το βάρος, οι οποίες επηρεάζουν όλη τη σχεδίαση του συστήματος.

## 2.3 Περιγραφή της διπλωματικής

Στα πλαίσια της παρούσας διπλωματικής ζητείται να σχεδιαστεί ένα αυτόνομο σύστημα καταγραφής δεδομένων οχήματος, με δυνατότητα διάθεσης των δεδομένων αυτών σε απομακρυσμένο εξυπηρετητή (Server) και αποθήκευσης σε βάση δεδομένων (upload to database). Για το σκοπό αυτό χρησιμοποιείται ως φιλοξενούν μηχανήμα (host computer) ένα Raspberry Pi και ως Server ένα εικονικό μηχανήμα (virtual machine) από την υπηρεσία Okeanos@grnet.

Το Raspberry Pi είναι υπεύθυνο για την καταγραφή των δεδομένων-μετρήσεων και για αυτό το σκοπό συνδέεται και επικοινωνεί με τα ακόλουθα υποσυστήματα:

- Ένα OBD adapter. Το OBD εισέρχεται στη θύρα OBD-II του αυτοκινήτου και επικοινωνεί με το Raspberry Pi μέσω Bluetooth σύνδεσης. Καταγράφει ταχύτητα, κατανάλωση και στροφές του κινητήρα.
- Ένα GPS dongle. Το GPS dongle συνδέεται με το Raspberry Pi σειριακά και καταγράφει τη θέση του αυτοκινήτου.
- Ένα επιταχυνσιόμετρο (accelerometer). Το accelerometer επικοινωνεί με το Raspberry Pi μέσω I2C σύνδεσης και καταγράφει την επιτάχυνση του αυτοκινήτου στους 3 άξονες.

Τα παραπάνω δεδομένα αποθηκεύονται στο Raspberry Pi και αποστέλλονται στο Server μέσω του SCP πρωτοκόλλου.

Στη μεριά του Server αφού παραληφθούν τα δεδομένα υλοποιείται επάνω σε αυτά ο αλγόριθμος του map-matching μέσω του προγράμματος Barefoot. Διορθώνεται και ταυτοποιείται, δηλαδή, η ακριβής θέση του αυτοκινήτου στο χάρτη και προκύπτουν τα επεξεργασμένα και ταυτοποιημένα στο χάρτη (snapped-to-map) δεδομένα.

Τα snapped-to-map δεδομένα αποθηκεύονται σε ένα πίνακα της βάσης δεδομένων PostgreSQL, ενώ δίνεται και η δυνατότητα αυτοματοποιημένης απεικονισής τους στο χάρτη μέσω του προγράμματος geojson.io.

## 2.4 Προδιαγραφές Απαιτήσεων

Προτού ξεκινήσει η σχεδίαση και η υλοποίηση της εφαρμογής είναι απαραίτητο να αποσαφηνιστούν οι ανάγκες καθώς και να καθοριστεί η λειτουργικότητα που θα παρέχεται από την εφαρμογή. Προς αυτό το σκοπό προδιαγράφουμε τις ακόλουθες απαιτήσεις:

- Η εφαρμογή καταγράφει διαγνωστικά οχήματος, όπως ταχύτητα και κατανάλωση.
- Η εφαρμογή καταγράφει δεδομένα θέσης οχήματος.
- Μέσω της εφαρμογής θέλουμε να γνωρίζουμε πόσο ομαλά ή ανώμαλα στρίβει, φρενάρει και επιτάχυνει ο οδηγός.
- Θέλουμε οι μετρήσεις μας να είναι σύγχρονες.
- Τα δεδομένα μας θα πρέπει να ταυτοποιούνται στο χάρτη.
- Θα πρέπει να αποθηκεύονται σε ένα κεντρικό σύστημα.
- Οι καταγραφές αποστέλλονται σε απομακρυσμένο εξηρηρητή.
- Η εφαρμογή εξάγει στατιστικά στοιχεία ανά οδικό σύνδεσμο.
- Η εφαρμογή θα πρέπει να είναι απλή στη χρήση.

Στα επόμενα κεφάλαια αναλύουμε πως εκπληρώνονται οι παραπάνω απαιτήσεις. Σε αυτό το σημείο συνοπτικά σημειώνουμε ότι στη μεριά του Raspberry Pi:

- Η εφαρμογή ξεκινάει με την εκκίνηση (boot) του Raspberry Pi και λαμβάνει ανά τακτά χρονικά διαστήματα δεδομένα από το OBD adapter, το accelerometer και το GPS dongle.
- Καταγράφονται μετρήσεις για ταχύτητα, στροφές κινητήρα και κατανάλωση (calculated engine load) στο OBD adapter και επιστρέφονται στο Raspberry μέσω Bluetooth.
- Καταγράφονται μετρήσεις για θέση και επιτάχυνση στο GPS και στο accelerometer αντίστοιχα, τα οποία επικοινωνούν με το Raspberry μέσω των ακίδων (GPIO pins) του.

- Η εφαρμογή σταματάει μόλις το OBD παύει να επιστρέφει έγκυρες τιμές για την ταχύτητα, τις στροφές ή την κατανάλωση.
- Το σύνολο των ληφθέντων δεδομένων αποθηκεύεται στο Raspberry Pi και αποστέλλεται στον Server.

Στη μεριά του εξυπηρετητή:

- Στο Server τρέχει script το οποίο ελέγχει αν αφίχθει έγγυρο αρχείο καταγραφής.
- Αν το αρχείο είναι έγκυρο τότε εφαρμόζεται στα δεδομένα του ο αλγόριθμος του map-matching και εξάγονται στατιστικά στοιχεία ανά οδικό σύνδεσμο.
- Τα στοιχεία αυτά αποθηκεύονται με ασφάλεια σε βάση δεδομένων και δίνεται η δυνατότητα να απεικονιστούν αυτοματοποιημένα σε χάρτη.

Όσον αφορά το χρήστη:

- Αυτός τοποθετεί κατάλληλα το OBD adapter στη OBD-II θύρα του αυτοκινήτου και θέτει σε λειτουργία τη μηχανή του αυτοκινήτου.
- Ελέγχει την σωστή τοποθέτηση του GPS dongle και του accelerometer στις ακίδες του Raspberry Pi και το θέτει σε λειτουργία.

### 3. Τεχνολογίες Hardware

Στο παρόν κεφάλαιο, πραγματευόμαστε τις τεχνολογίες Hardware που επιλέξαμε να χρησιμοποιήσουμε για την ανάπτυξη της εφαρμογής, τους λόγους που υπαγόρευσαν τις επιλογές μας καθώς και τα κύρια χαρακτηριστικά κάθε μίας από αυτές τις τεχνολογίες. Χωρίσαμε τις υποενοτήτες που αφορούν το σύνολο του υλικού που χρησιμοποιήθηκε ως εξής:

Αναφερόμαστε αρχικά, στο ενσωματωμένο υπολογιστικό σύστημα που χρησιμοποιήσαμε το Raspberry Pi. Έπειτα, αναφερόμαστε στη διεπαφή OBD, το επιταχυνσιόμετρο και το GPS καθώς και τα σχετιζόμενα με αυτά εξαρτήματα και πρωτόκολλα που χρησιμοποιήθηκαν για την σύνδεση των υποσυστημάτων με το host PC.

Τέλος, αναλύουμε τα πρωτόκολλα που σχετίζονται με τους τρόπους με τους οποίους μπορεί να γίνει η επικοινωνία πελάτη – εξυπηρετητή μέσω ενός WLAN adapter ή ενός 4G modem.



**Εικόνα 2: Το Raspberry Pi συνδεδεμένο με το GPS και το επιταχυνσιόμετρο**

### 3.1 Raspberry Pi

Το Raspberry Pi είναι ένας πλήρης single-board υπολογιστής σε μέγεθος πιστωτικής κάρτας ο οποίος αναπτύχθηκε στο Ηνωμένο Βασίλειο με σκοπό να προωθήσει τη διδασκαλία της Επιστήμης των Υπολογιστών στα σχολεία και στις αναπτυσσόμενες χώρες. Έχουν κυκλοφορήσει αρκετές εκδόσεις για το Raspberry Pi. Η πρώτη γενιά Pi 1 κυκλοφόρησε το Φεβρουάριο του 2012 με βασικό μοντέλο το model A, ενώ το model B κυκλοφόρησε έχοντας ανώτερες προδιαγραφές. Ακολούθησαν τα μοντέλα A+ και B+ το 2013 και το Φλεβάρη του 2015 κυκλοφόρησε το Raspberry Pi 2 model B, το οποίο χρησιμοποιήθηκε για τις ανάγκες της παρούσας διπλωματικής.



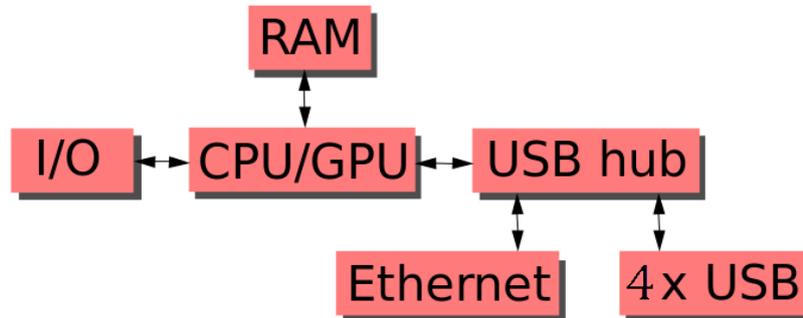
Εικόνα 3: Το Raspberry Pi Model B

Το Φλεβάρη του 2016 κυκλοφόρησε το Raspberry Pi 3 model B, ενώ το Νοέμβριο του 2015 κυκλοφόρησε και το μικροσκοπικό σε μέγεθος Raspberry Pi Zero με περιοσμένες δυνατότητες και τιμή αγοράς στα 5 μόλις δολάρια.



Εικόνα 4: Το Raspberry Pi Zero

Το hardware του Raspberry Pi έχει εξελιχθεί για τις διάφορες εκδόσεις του και διαθέτει παραλλαγές σε χωρητικότητα μνήμης και υποστήριξη περιφερειακών συσκευών. Το δομικό διάγραμμα για την έκδοση Raspberry Pi 2 Model B είναι το ακόλουθο:



Διάγραμμα 1: Δομικό Διάγραμμα του Raspberry Pi 2 Model B

Η συγκεκριμένη έκδοση περιέχει:

- Broadcom BCM2836 System-on-chip
- 900MHz quad-core ARM Cortex-A7 CPU με δυνατότητα overclocking στο 1GHz
- 1GB RAM
- 4 USB ports
- 40 GPIO pins
- Full HDMI port
- Ethernet port
- Combined 3.5mm audio jack and composite video
- Camera interface (CSI)
- Display interface (DSI)
- Micro SD card slot
- VideoCore IV 3D graphics core

Λόγω της ύπαρξης του ARMv7 επεξεργαστή, μπορεί να τρέξει ένα ευρύ σύνολο από ARM GNU/Linux διανομές, με κυριότερο λειτουργικό σύστημα το Raspbian, αλλά και διανομές όπως το Snappy Ubuntu Core, το Arch Linux ARM, το Pidora (Fedora Remix) και τα Puppy Linux. Ακόμη, το Raspberry Pi 2 υποστηρίζει Windows 10 Iot Core.

## 3.2 Η διεπαφή OBD-II

Ο όρος On-board diagnostics (Ενσωματωμένα συστήματα διάγνωσης) ή OBD, είναι ένας όρος της αυτοκινητοβιομηχανίας που αναφέρεται στην δυνατότητα των οχημάτων για αυτοδιάγνωση και αναφορά βλαβών. Τα συστήματα OBD δίνουν στον ιδιοκτήτη ενός οχήματος ή στον τεχνικό πρόσβαση σε πληροφορίες για την κατάσταση διαφόρων υποσυστημάτων του οχήματος. Το είδος και η ποσότητα των διαγνωστικών πληροφοριών που διατίθενται μέσω των OBD έχει διαφοροποιηθεί σε μεγάλο βαθμό από την εμφάνισή τους στις πρώτες τους εκδόσεις τη δεκαετία του 1980. Οι πρώτες εκδόσεις των ενσωματωμένων διαγνωστικών συστημάτων ενεργοποιούσαν κάποια ενδεικτική λυχνία βλάβης, αλλά δεν παρείχαν καμία πληροφορία ως προς τη φύση του προβλήματος. Οι σύγχρονες υλοποιήσεις OBD χρησιμοποιούν μια τυποποιημένη ψηφιακή θύρα για να παρέχουν δεδομένα σε πραγματικό χρόνο, επιπρόσθετα σε μια τυποποιημένη σειρά από Διαγνωστικούς Κωδικούς Σφαλμάτων, ή DTCs - Diagnostic Trouble Codes, οι οποίοι επιτρέπουν τον ταχύτατο εντοπισμό βλαβών.

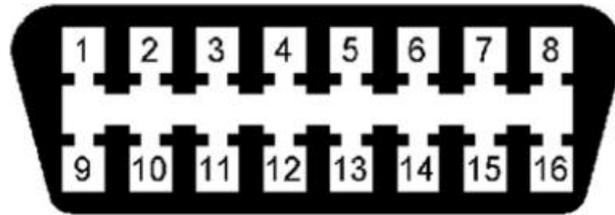
Στην παρούσα διπλωματική εργασία θα ασχοληθούμε με την έκδοση 2 των ενσωματωμένων διαγνωστικών συστημάτων (OBD-II), η οποία αποτελεί βελτίωση των εκδόσεων OBD-I και OBD-1.5. Το πρότυπο καθορίζει τον τύπο της θύρας και το pinout της, τα σήματα επικοινωνίας και τη δομή των μηνυμάτων. Επίσης καθορίζει μια υποψήφια λίστα των παραμέτρων για έλεγχο μαζί με την κωδικοποίηση των διαγνωστικών μηνυμάτων.



Εικόνα 5: Η θύρα OBD-II σε GOLF 4

### 3.2.1 Η θύρα OBD-II

Δεν υπάρχει διεθνές πρότυπο για τη θύρα OBD-II. Στην πλειοψηφία των οχημάτων έχει 16 επαφές (pins) οι οποίες διαφαίνονται στο παρακάτω διάγραμμα:



Διάγραμμα 2: Δομικό Διάγραμμα της θύρας OBD

Τα pins που η χρήση τους έχει αφηθεί στην ευχέρεια των κατασκευαστών χρησιμοποιούνται από τις εταιρίες κατασκευής αυτοκινήτων για την υλοποίηση δικών τους ιδιωτικών πρωτοκόλλων ή για πρόσβαση σε ιδιόκτητα δίκτυα των κατασκευαστών που εκτός από τις λειτουργίες που περιγράφουν τα πρωτόκολλα OBD υλοποιούν και άλλες επιπρόσθετες. Τέτοια ιδιωτικά δίκτυα δίνουν πληροφορίες για την κατάσταση συστημάτων όπως οι ζώνες ασφαλείας, οι αερόσακοι τα φώτα πορείας, το σύστημα αντιμπλοκαρίσματος φρένων (ευρέως γνωστό ως ABS), το σύστημα ελέγχου πρόσφυσης (ευρέως γνωστό ως ESP ή ESC) κλπ. Επιπλέον επιτρέπουν μέσω των κατάλληλων εντολών τον έλεγχο του συστήματος κεντρικού κλειδώματος, τον έλεγχο των ηλεκτρικών παραθύρων, του ηχοσυστήματος των φλας και των αλάρμ κλπ. Βέβαια για να μπορεί να ερμηνεύσει κανείς τα μηνύματα και να γνωρίζει τις εντολές που υποστηρίζει ένα τέτοιο ιδιωτικό δίκτυο θα πρέπει να έχει πρόσβαση στο πρωτόκολλο που έχει ορίσει ο κατασκευαστής.

Διαφορετικά πρωτόκολλα σηματοδοσίας είναι επιτρεπτά προς χρήση με την διεπαφή OBD-II. Είναι εύκολο να καταλάβει κανείς ποιο πρωτόκολλο υλοποιείται σε ένα συγκεκριμένο όχημα από την ύπαρξη ή μη συγκεκριμένων ακροδεκτών στη θύρα OBD-II. Κυριότερα είναι τα:

- ISO 15765-4
- ISO 14230-4
- ISO 9141-2
- SAE J1850 PWM και VPW
- ISO 11898
- ISO 15765
- SAE J1939

Τα OBD Modes είναι διαφορετικές υπηρεσίες που παρέχονται από τις Ηλεκτρονικές Μονάδες Ελέγχου (Electronic Control Units ή ECUs) των οχημάτων, όπως καθαρισμός των αποθηκευμένων διαγνωστικών κωδικών σφαλμάτων ή παροχή πληροφοριών σε ζωντανό χρόνο από τους αισθητήρες του οχήματος. Οι κατασκευαστές δεν είναι υποχρεωμένοι να υλοποιούν όλες τις υπηρεσίες (modes) και μπορούν να υλοποιούν δικές τους (manufacturer specific data). Κάθε υπηρεσία έχει το δικό της μοναδικό αναγνωριστικό της μορφής xx όπου το σύμβολο x αναπαριστά δεκαεξαδικά ψηφία. Υπάρχουν 10 διαφορετικές υπηρεσίες που μπορεί να υποστηρίζονται από τα συστήματα OBD. Σε νεότερα οχήματα που υπάρχει η τάση διάφορες λειτουργίες που διεξάγονταν από αμιγώς μηχανικά συστήματα να εκτελούνται πλέον από ηλεκτρονικά ή ηλεκτρονικώς ελεγχόμενα συστήματα υπάρχει πιθανότητα να υποστηρίζονται περισσότερες υπηρεσίες σε σχέση με παλαιότερα μοντέλα της αυτοκινητοβιομηχανίας. Ακολουθεί μία σύντομη περιγραφή όλων των υπηρεσιών:

- **Mode 01:** Παρέχει πληροφορίες σχετικές με το σύστημα μετάδοσης κίνησης. Ένα διαγνωστικό εργαλείο μπορεί να λάβει πληροφορίες όπως η ταχύτητα του οχήματος, οι στροφές λειτουργίας του κινητήρα, η θέση του πεντάλ ή της πεταλούδας του επιταχυντή (γκαζιού), η απόλυτη πίεση του αέρα στην εισαγωγή του κινητήρα κλπ.
- **Mode 02:** Επιστρέφει τα αποθηκευμένα freeze frames που σχετίζονται με έναν κωδικό σφάλματος. Όταν ένας κωδικός σφάλματος ενεργοποιείται μαζί με αυτόν αποθηκεύονται και πληροφορίες για τις συνθήκες που επικρατούσαν στο όχημα κατά την ενεργοποίησή του. Τέτοιες πληροφορίες μπορεί να είναι η θερμοκρασία του μείγματος αντιψυκτικού στο κύκλωμα ψύξης του αυτοκινήτου, η ταχύτητα περιστροφής του κινητήρα, η αναλογία μείγματος αέρα βενζίνης κλπ. οι οποίες μπορούν να βοηθήσουν στην ευκολότερη διάγνωση της αιτίας του προβλήματος.
- **Mode 03:** Επιστρέφει τους αποθηκευμένους κωδικούς σφαλμάτων. Οι κωδικοί μπορεί να διαγραφούν από τη μνήμη της ECU με τη βοήθεια ενός διαγνωστικού εργαλείου.
- **Mode 04:** Επιτρέπει τον καθαρισμό από τη μνήμη των ECUs των διαγνωστικών κωδικών ύστερα από κατάλληλο αίτημα της διαγνωστικής συσκευής.
- **Mode 05:** Επιστρέφει πληροφορίες από τον αισθητήρα (ή τους αισθητήρες) Οξυγόνου (γνωστό και ως αισθητήρα λάμδα) του οχήματος. Ο αισθητήρας μετρά την ποσότητα του οξυγόνου στα αέρια της εξάτμισης και από τη μέτρηση αυτή μπορεί να καθοριστεί αν η αναλογία αέρα καυσίμου είναι η ιδανική ή αν χρειάζεται η μονάδα ελέγχου του κινητήρα να κάνει κάποια προσαρμογή.

- **Mode 06:** Επιστρέφει αποτελέσματα αυτοδιάγνωσης διαφόρων ηλεκτρονικών συστημάτων του οχήματος. Για παράδειγμα μπορεί να διεξαχθεί έλεγχος για προβληματικούς αισθητήρες, για προανάφλεξη του καυσίμου κ.ά.
- **Mode 07:** Επιστρέφει του αποθηκευμένους κωδικούς σφαλμάτων οι οποίοι ενεργοποιήθηκαν στον τρέχοντα ή τον προηγούμενο κύκλο οδήγησης.
- **Mode 08:** Επιτρέπει τον έλεγχο συστημάτων του αυτοκινήτου μέσω της θύρας OBD με τη βοήθεια του διασυνδεδεμένου διαγνωστικού εργαλείου.
- **Mode 09:** Επιστρέφει πληροφορίες για το όχημα όπως η διάρκεια λειτουργίας του κινητήρα, η απόσταση που έχει διανυθεί ενώ εκκρεμούν ενεργοποιημένοι κωδικοί σφάλματος, ο αριθμός πλαισίου του αυτοκινήτου κλπ.
- **Mode 10:** Επιστρέφει του μόνιμους κωδικούς σφαλμάτων. Οι κωδικοί αυτοί παραμένουν αποθηκευμένοι σε non-volatile μνήμη της μονάδας που τους ενεργοποίησε και διαγράφονται μόνο μετά την αποκατάσταση της σχετιζόμενης βλάβης. Δεν υπάρχει δυνατότητα διαγραφής τους από τον διαγνωστικό εξοπλισμό.

### 3.2.2 ELM327 OBD Adapter

Το ELM327 είναι ένας προγραμματιζόμενος μικροελεγκτής που παράγεται από την ELM Electronics και συνδέεται με τη θύρα OBD των σύγχρονων οχημάτων. Μέσω της θύρας OBD-II επικοινωνεί με τον εγκέφαλο του αυτοκινήτου και μπορεί να παρέχει στον χρήστη διαγνωστικό έλεγχο του οχήματος αλλά και πραγματικού χρόνου μετρήσεις όπως στροφές κινητήρα, κατανάλωση, ταχύτητα κ.ά. Το συγκεκριμένο εξάρτημα συνδέεται ασύρματα μέσω Bluetooth με το smartphone, το tablet ή και το PC και αποτελεί ένα από τα πιο διαδεδομένα συστήματα επικοινωνίας PC-to-OBD. Υποστηρίζει πολλαπλά πρωτόκολλα επικοινωνίας και εντολές τύπου AT. Για παράδειγμα η εντολή: AT RV[Read the input Voltage] επιστρέφει την τάση εισόδου, ενώ η εντολή AT DP[Describe the current Protocol] περιγράφει το πρωτόκολλο που χρησιμοποιείται στο τρέχον όχημα. Τέλος, παρέχεται η δυνατότητα εκκαθάρισης των σφαλμάτων του οχήματος. [14]



Εικόνα 6: ELM327 OBD adapter

### 3.2.3 Bluetooth



Εικόνα 7: Bluetooth

Το Bluetooth είναι ένα βιομηχανικό πρότυπο για τα ασύρματα προσωπικά δίκτυα (Wireless Personal Area Networks, WPAN). Πρόκειται για μια τηλεπικοινωνιακή τεχνολογία μικρών αποστάσεων (της τάξεως των 10 μέτρων), η οποία μπορεί να μεταδώσει σήματα μέσω ραδιοκυμάτων σε ψηφιακές συσκευές. Το Bluetooth επιτρέπει την οριστική κατάργηση όλων των καλωδίων που ήταν απαραίτητα για τη διασύνδεση των διάφορων ψηφιακών συσκευών, και παρέχει προτυποποιημένη, ασύρματη επικοινωνία ανάμεσα σε PDAs, κινητά τηλέφωνα, φορητούς υπολογιστές, προσωπικούς υπολογιστές, εκτυπωτές, scanners καθώς και σε ψηφιακές φωτογραφικές μηχανές ή ψηφιακές κάμερες, μικρόφωνα, ακουστικά, ραδιόφωνα κ.α. Η επικοινωνία αυτή επιτυγχάνεται, μέσω μιας ασφαλούς, φθηνής και παγκοσμίως διαθέσιμης, χωρίς ειδική άδεια, ραδιοσυχνότητας μικρής εμβέλειας (2.4GHz).

Η βασική δομική μονάδα ενός δικτύου Bluetooth είναι το piconet, στο οποίο όλοι οι κόμβοι που μετέχουν (μέχρι 7 συσκευές Slaves) μοιράζονται τον ίδιο κώδικα διασποράς και υπόκεινται στον έλεγχο ενός κοινού Master. Ο Master είναι υπεύθυνος για τις μεταβολές της δικτυακής τοπολογίας (εισαγωγές, αποχωρήσεις κόμβων και συντονισμός τους), ενώ ένα piconet σχηματίζεται από έναν κόμβο που επιθυμεί να γίνει Master.

Στη παρούσα διπλωματική χρησιμοποιήθηκε για την επικοινωνία του Raspberry Pi με το OBD adapter το ακόλουθο εξάρτημα: Cambridge Silicon Radio, Ltd Bluetooth Dongle. Τοποθετείται στη θύρα USB και συνδυάζεται ιδανικά με το Pi λόγω του μικροσκοπικού του μεγέθους.



Εικόνα 8: Bluetooth adapter

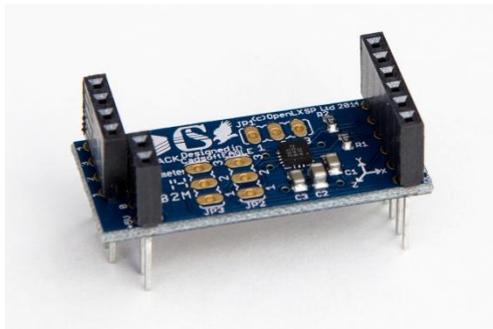
### 3.3 Επιταχυνσιόμετρο

Το Επιταχυνσιόμετρο (Accelerometer) είναι μια ηλεκτρομηχανική συσκευή που έχει την ικανότητα να μετρά δυνάμεις επιτάχυνσης. Αυτές οι δυνάμεις μπορεί να είναι στατικές, όπως είναι η επιτάχυνση της βαρύτητας, ή δυναμικές όταν προκαλούνται – προέρχονται από αλλαγές στην ταχύτητα στην διεύθυνση της κίνησης (επιταχύνσεις, επιβραδύνσεις, στροφές).

Χρησιμοποιείται κυρίως σε Αδρανειακά συστήματα πλοήγησης (Inertial Navigation System – INS), ως επίσης χρησιμοποιείται για την μέτρηση και καταγραφή των επιταχύνσεων στους τρεις άξονες των αεροσκαφών από τον καταγραφέα των στοιχείων της πτήσης, για την ενεργοποίηση του αερόσακου στα αυτοκίνητα, και σε οποιαδήποτε άλλη εφαρμογή απαιτείται η μέτρηση της επιτάχυνσης.

Για τις ανάγκες της παρούσας διπλωματικής χρησιμοποιήθηκε το Accelerometer της Microstack.

#### 3.3.1 Microstack Accelerometer



Εικόνα 9: Το επιταχυνσιόμετρο της Microstack

Για την καταγραφή της επιτάχυνσης του οχήματος χρησιμοποιήσαμε το επιταχυνσιόμετρο (Accelerometer) της Microstack[15]. Μερικά κύρια χαρακτηριστικά του είναι τα εξής:

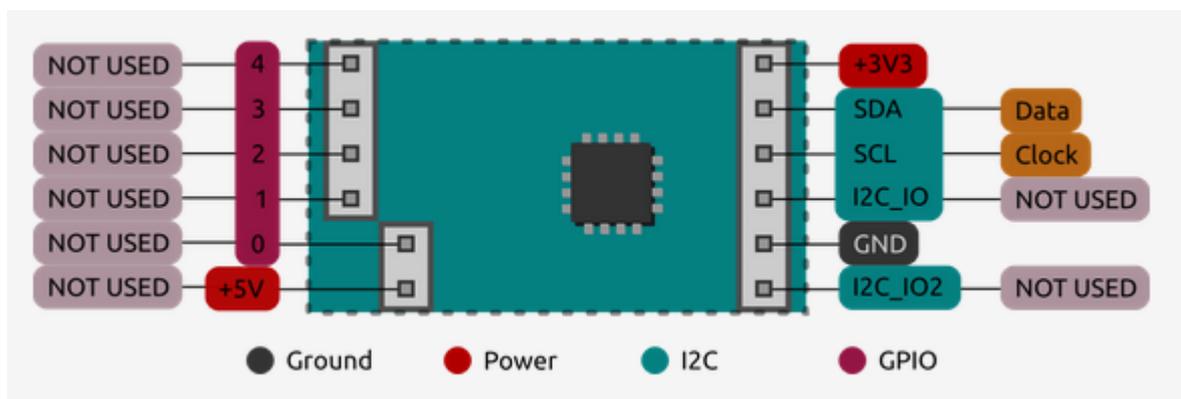
- Ιδιαίτερα χαμηλή απαιτούμενη ισχύς (400nA ανά Hz)
- Εξαιρετικά γρήγορη απόκριση (data output time ~700μs)
- Δυνατότητα μέτρησης μέχρι τα 8G
- I2C ψηφιακή διασύνδεση
- 3 άξονες - Εντοπισμός κλίσης

Και αυτό το εργαλείο λόγω του μικρού του μεγέθους συνδυάζεται ιδανικά με το Raspberry Pi.

### 3.3.2 I2C

Το πρωτόκολλο I2C (inter-integrated circuit) αναπτύχθηκε την δεκαετία του 1980 από την Philips αρχικά με σκοπό την εύκολη επικοινωνία μεταξύ μιας Κεντρικής Μονάδας Επεξεργασίας (CPU) με τα περιφερειακά κυκλώματα μιας τηλεόρασης. Αποτελεί ένα πρωτόκολλο που χρησιμοποιείται για την διασύνδεση μεταξύ διαφορετικών ολοκληρωμένων, όπως των μικροελεγκτών και των υπόλοιπων ολοκληρωμένων περιφερειακών. Με το I2C αποφεύγεται η χρησιμοποίηση ενός παράλληλου διαύλου δεδομένων που εισάγει μεγάλη πολυπλοκότητα στη σχεδίαση αλλά και μεγαλύτερο κόστος, ενώ το συγκεκριμένο πρωτόκολλο βρίσκει πολλές εφαρμογές στα σύγχρονα ηλεκτρονικά συστήματα όπως σε συσκευές εικόνας και ήχου, σε τηλεφωνικές συσκευές, modems και embedded microprocessor boards. Οι ταχύτητες τις οποίες επιτυγχάνει μπορούν να φτάσουν μέχρι και 3.4 Mbps.

Στη παρούσα διπλωματική χρησιμοποιήθηκε για τη σύνδεση του επιταχυνσιόμετρου με το Raspberry Pi.



Διάγραμμα 3: I2C σύνδεση για το επιταχυνσιόμετρο

### 3.4 GPS

Το παγκόσμιο σύστημα προσδιορισμού θέσης (GPS) είναι ένα δίκτυο δορυφόρων που κινούνται σε τροχιά γύρω από τη γη, σε σταθερά σημεία πάνω από τον πλανήτη και μεταδίδουν σήματα προς την γη. Τα σήματα αυτά, φέρουν κώδικα χρόνου και σημείο γεωγραφικών δεδομένων, που παρέχουν στους χρήστες τη δυνατότητα να εντοπίζουν την ακριβή τους θέση, την ταχύτητα και την ώρα σε οποιοδήποτε σημείο του πλανήτη.

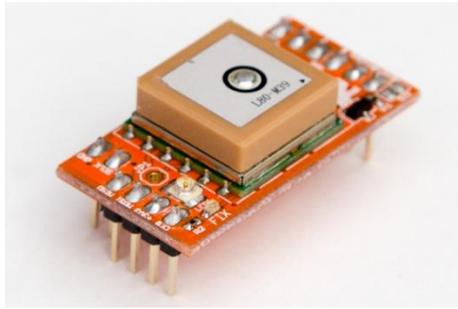
Το GPS σχεδιάστηκε για στρατιωτικές και κατασκοπευτικές εφαρμογές κατά την περίοδο της κορύφωσης του Ψυχρού Πολέμου, τη δεκαετία του 1960, ωστόσο σήμερα γνωρίζει ευρεία αποδοχή και χρησιμοποιείται σε πολλές εφαρμογές, συμπεριλαμβανομένων της πλοήγησης αεροσκαφών και πλοίων, της κατάρτισης δρομολογίων για οδηγούς, της χαρτογράφησης, της σειсмоγραφικής έρευνας και των κλιματικών μελετών.

Η μεγάλη εξάπλωση της χρήσης του GPS οφείλεται και στη διάδοση των οικονομικά προσιτών φορητών δεκτών GPS για πεζούς ή οχήματα και των γενικών υπολογιστικών συσκευών με ενσωματωμένο δέκτη GPS. Ένας φορητός δέκτης GPS αποτελείται από:

- Την εσωτερική δορυφορική κεραία, η οποία λαμβάνει το σήμα GPS από τους δορυφόρους με τους οποίους έχει οπτική επαφή.
- Τον κυρίως δέκτη GPS ο οποίος χρησιμοποιεί κυκλώματα πολύ χαμηλού θορύβου και ειδικές τεχνικές επεξεργασίας σήματος ώστε να ξεχωρίζει τα εξαιρετικά ασθενή σήματα των δορυφόρων από τον ισχυρό τηλεπικοινωνιακό θόρυβο. Ο ρυθμός με το οποίο βγαίνει νέο στίγμα στην έξοδο του δέκτη είναι συνήθως 1Hz, αν και υπάρχουν δέκτες που λαμβάνουν στίγμα με ταχύτερους ρυθμούς π.χ. 10Hz.
- Τον κυρίως μικροελεγκτή και το υπόλοιπο hardware της επικοινωνίας του χρήστη με τη συσκευή.

Στην παρούσα διπλωματική ενσωματώσαμε το GPS της Microstack στο Raspberry Pi.

### 3.4.1 Microstack GPS



Εικόνα 10: Το GPS της Microstack

Για την καταγραφή της θέσης του οχήματος χρησιμοποιήσαμε το GPS της Microstack [16]. Κάποια από τα κύρια χαρακτηριστικά του είναι:

- Ιδιαίτερα χαμηλή κατανάλωση ισχύος (20mA-25mA και 2.8V-4.3V)
- Σύντομο χρονικό διάστημα μέχρι να εντοπίσει τους δορυφόρους και να πάρει την πρώτη μέτρηση (Short time to fix)
- Ενσωματωμένη και υψηλής ακρίβειας εσωτερική κεραία patch
- Συχνότητα μετρήσεων από 1Hz έως 10Hz
- Δυνατότητα αυτόματου προγραμματισμού περιόδων wake/sleep
- Ακρίβεια στίγματος 5m

Πρόκειται για ένα GPS dongle του οποίου οι ιδιαίτερα χαμηλές απαιτήσεις σε ενέργεια καθώς και το μικρό του μέγεθος το καθιστούν ιδανικό για χρησιμοποίηση μαζί το Raspberry Pi.

### 3.4.2 SPI

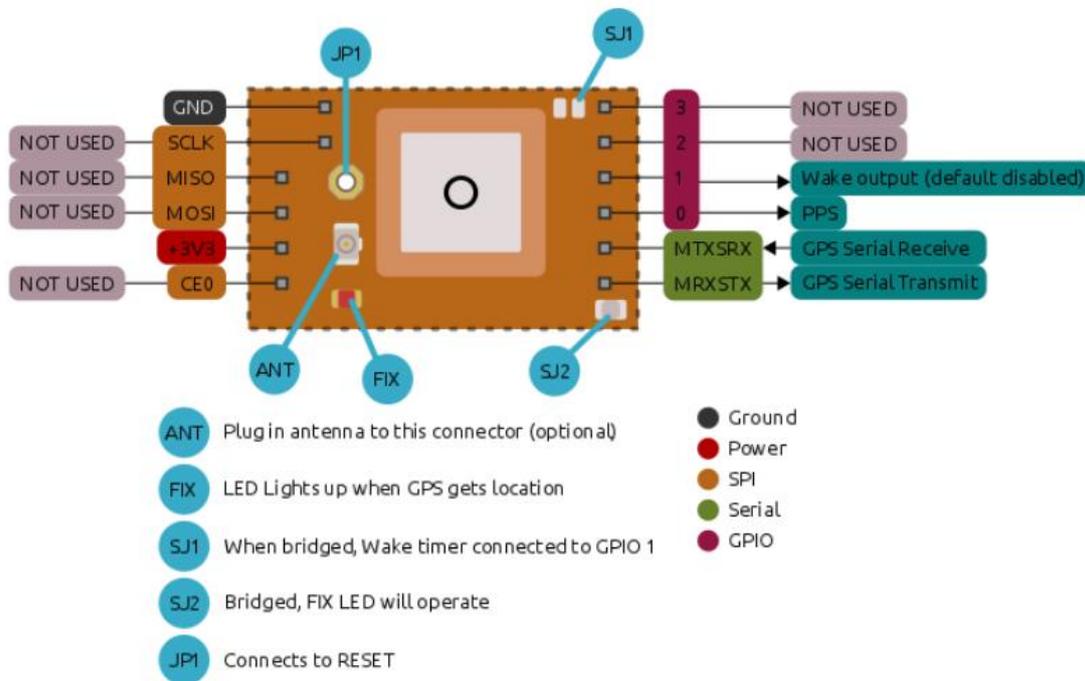
Το SPI (Serial Peripheral Interface), όπως και το I2C αναπτύχθηκε με σκοπό την εύκολη επικοινωνία μεταξύ ολοκληρωμένων και τον καλύτερο τρόπο διασύνδεσης των περιφερειακών μονάδων και των μικροελεγκτών μεταξύ τους.

Το SPI επιτρέπει σε δεδομένα των 8-bits να αποστέλλονται και να λαμβάνονται σύγχρονα με ταχύτητα που φτάνει το 1Mbps. Πρόκειται για γρήγορο, χαμηλού κόστους και παγκοσμίως αποδεκτό πρωτόκολλο το οποίο βρίσκει ευρεία εφαρμογή σε ένα σύνολο από περιφερειακά συστήματα όπως:

- αισθητήρες: θερμοκρασίες και πίεσης
- επικοινωνίες: Ethernet, USB, USART, CAN, IEEE 802.15.4, IEEE 802.11

- Μνήμες: flash και EEPROM
- Ρολόγια πραγματικού χρόνου

Στη διπλωματική χρησιμοποιήθηκε για τη σύνδεση του GPS με το Raspberry Pi.



Διάγραμμα 4: SPI σύνδεση για το GPS

### 3.5 WLAN adapter - Ασύρματα Τοπικά Δίκτυα

Ένα ασύρματο τοπικό δίκτυο (WLAN) είναι ένα ασύρματο δίκτυο υπολογιστών το οποίο συνδέει δύο ή παραπάνω συσκευές χρησιμοποιώντας μια ασύρματη μέθοδο διανομής, εντός μιας περιορισμένης σε έκταση περιοχής όπως ένα σπίτι, ένα σχολείο ή ένα εργαστήριο υπολογιστών. Επιτρέπει στους χρήστες να μετακινούνται εντός της τοπικής περιοχής κάλυψης και να παραμένουν συνδεδεμένοι στο δίκτυο, ενώ παρέχει τη δυνατότητα σύνδεσης στο ευρύτερο Internet. Τα νεότερα WLANs βασίζονται στην προδιαγραφή IEEE 802.11, και είναι ευρέως γνωστά ως Wi-Fi.

Κάθε συσκευή που χρησιμοποιεί το πρότυπο Wi-Fi μπορεί να χρησιμοποιήσει ένα σημείο πρόσβασης ασυρμάτου δικτύου (Access Point – AP) ώστε να αποκτήσει πρόσβαση σε κάποιο πόρο του δικτύου. Τέτοιες συσκευές μπορεί να είναι επιτραπέζιοι ή φορητοί υπολογιστές, έξυπνες συσκευές αλλά ακόμα και τηλεοράσεις, φωτογραφικές μηχανές και συσκευές αναπαραγωγής μουσικής. Υπάρχουν πέντε διαδεδομένοι τύποι δικτύου WLAN που έχουν σχεδιαστεί τα τελευταία χρόνια: 802.11a, 802.11b, 802.11g, 802.11n, 802.11ac, κάθε ένας από τους οποίους λειτουργεί σε διαφορετικό εύρος συχνότητας και προσφέρει διαφορετικούς ρυθμούς μετάδοσης δεδομένων. Το πρότυπο 802.11n (2009) λειτουργεί σε συχνότητες με εύρος ζώνης 2,4GHz και 5GHz και δύναται να μεταφέρει δεδομένα με ταχύτητες έως 150Mbps, ενώ το 802.11ac (2012) λειτουργεί αποκλειστικά σε συχνότητες με εύρος ζώνης 5GHz και προσφέρει ρυθμούς μετάδοσης που ξεπερνούν τα 850Mbps.

Για τη δικτύωση του Raspberry Pi χρησιμοποιούμε ένα ασύρματο USB adapter (Wireless USB Adapter). Πρόκειται για μικροσκοπικές συσκευές οι οποίες συνδέονται στη θύρα USB του φιλοξενούντος μηχανήματος και υποστηρίζουν όλα τα απαραίτητα πρωτόκολλα για την επικοινωνία του με το Internet. Στην παρούσα διπλωματική χρησιμοποιήσαμε το: Realtek Semiconductor Corp. RTL8188CUS 802.11n WLAN Adapter



Εικόνα 11: WLAN adapter

### 3.5.1 Λειτουργία Tethering

Στο πλαίσιο των κινητών τηλεφώνων και των tablet, η λειτουργία Tethering επιτρέπει την κοινή χρήση της σύνδεσης Internet του τηλεφώνου ή του tablet με άλλες συσκευές, όπως οι φορητοί υπολογιστές. Σύνδεση του τηλεφώνου ή του tablet με άλλες συσκευές μπορεί να γίνει μέσω ασύρματου LAN (Wi-Fi), μέσω Bluetooth ή με φυσική σύνδεση χρησιμοποιώντας ένα καλώδιο, για παράδειγμα μέσω USB.

Αν η σύνδεση γίνεται μέσω WLAN, η λειτουργία μπορεί να χαρακτηριστεί ως φορητό σημείο πρόσβασης (mobile hotspot), το οποίο επιτρέπει στη συσκευή με σύνδεση στο Internet να χρησιμεύσει ως ένας φορητός δρομολογητής (router).

Στη παρούσα διπλωματική χρησιμοποιείται το smartphone LG E440, το οποίο συνδέεται στο Internet μέσω των δεδομένων (data) της κινητής τηλεφωνίας και λειτουργεί ως φορητό σημείο πρόσβασης για το Raspberry Pi. Αντί για τη λειτουργία tethering μπορεί να χρησιμοποιηθεί ένα 4G modem για την επικοινωνία πελάτη-εξυπηρετητή. Αναφερόμαστε στα 4G δίκτυα στο αμέσως επόμενο κεφάλαιο.

### 3.5.2 Δίκτυα 4G – LTE

Τα δίκτυα LTE (Long Term Evolution) αποτελούν ένα πρότυπο για ασύρματη επικοινωνία, το οποίο αναπτύχθηκε από τον οργανισμό 3GPP (3<sup>rd</sup> Generation Partnership Project) και τεκμηριώνεται στις εκδόσεις 8 (3.9G) και 10 (4G) της 3GPP. Αποτελεί εξέλιξη των ασύρματων δικτύων GSM (Global System for Mobile) και UMTS (Universal Mobile Telecommunications System) προς ένα δίκτυο ευρείας ζώνης με αυξημένη χωρητικότητα και μεγαλύτερη ταχύτητα δικτύου. Ιστορικά το πρώτο δημόσιο δίκτυο LTE που εγκαταστάθηκε στο κόσμο, ήταν στο Όσλο και τη Στοκχόλμη από την εταιρεία TeliaSonera στις 14 Δεκεμβρίου 2009.

Οι δημιουργοί του LTE φιλοδοξούν να αποτελέσει το πρώτο πραγματικά παγκόσμιο πρότυπο κινητής τηλεφωνίας [12] και να ξεπεράσει το αντίστοιχο ανταγωνιστικό πρότυπο τέταρτης γενιάς WiMAX, το οποίο αποτελεί φυσική συνέχεια των δικτύων τρίτης γενιάς στις Η.Π.Α. Τα πιο σημαντικά κίνητρα για την ανάπτυξη του LTE ήταν τα εξής:

- Η ανάγκη για διατήρηση της ανταγωνιστικότητας του συστήματος 3G στο μέλλον.
- Η απαίτηση των χρηστών για υψηλότερους ρυθμούς δεδομένων και καλύτερη ποιότητα υπηρεσιών.
- Η ανάγκη για ένα βελτιστοποιημένο σύστημα με ανταλλαγή πακέτων.

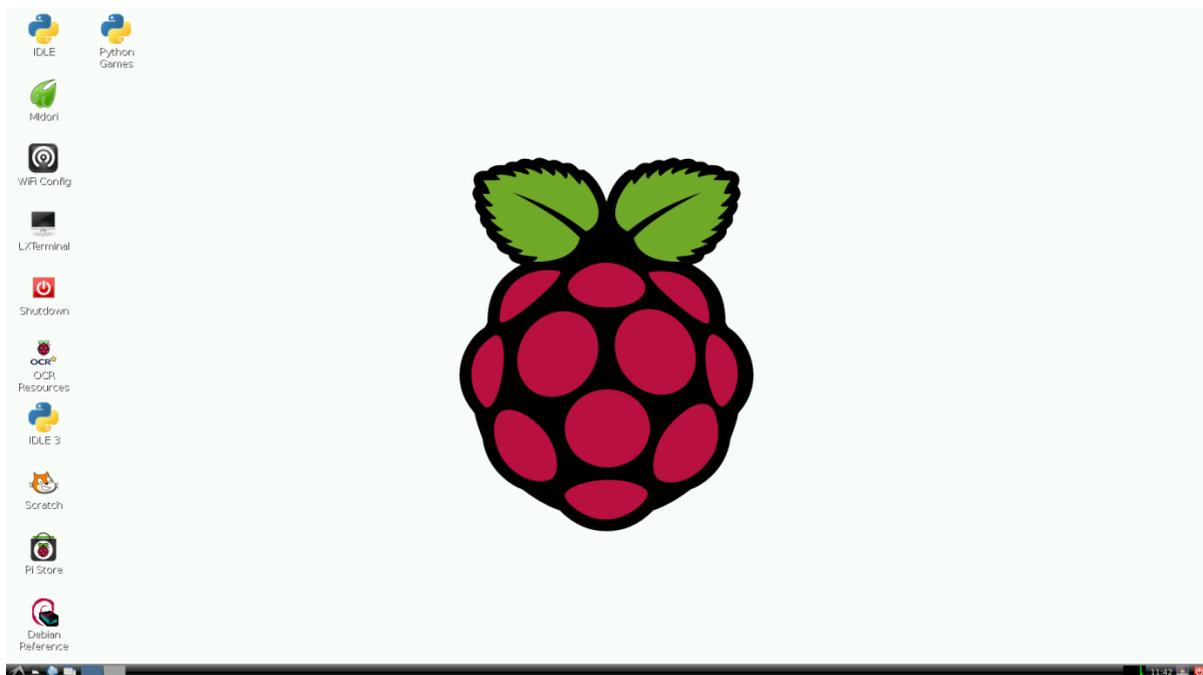
- Η συνεχής απαίτηση από τους χρήστες για χαμηλότερο κόστος.
- Η χαμηλότερη πολυπλοκότητα.
- Η αποφυγή του άσκοπου κατακερματισμού των τεχνολογιών για συνδεδεμένες και ασύνδετες ζώνες λειτουργίας.

Ένα 4G δίκτυο λειτουργεί στις συχνότητες 2-8 GHz και επιτρέπει τη μεταφορά δεδομένων με ταχύτητες πολλαπλάσιες σε σχέση με τον προκάτοχο του 3G. Στη παρούσα διπλωματική χρησιμοποιήθηκε για την επικοινωνία του πελάτη Raspberry Pi με τον εξυπηρετητή που στήθηκε στην υπηρεσία Okeanos μέσω ενός 4G modem.

## 4. Τεχνολογίες-Software

Στο συγκεκριμένο κεφάλαιο, πραγματευόμαστε τις Τεχνολογίες Software που χρησιμοποιούμε στο παρόν πόνημα και αναλύουμε τα βασικά χαρακτηριστικά τους. Οι υποενότητες όσον αφορά το λογισμικό έχουν χωριστεί ως εξής:

Αρχικά αναφερόμαστε εκτενώς στη γλώσσα προγραμματισμού που χρησιμοποιήθηκε κατά κόρον στην παρούσα διπλωματική, την Python. Ακολούθως, αναφερόμαστε στο λειτουργικό σύστημα Raspbian που τρέχει το Raspberry Pi και στους χάρτες OpenStreetMap. Σε ξεχωριστή ενότητα γίνεται αναφορά στον Map-Matching αλγόριθμο του Barefoot και στην υπηρεσία Okeanos. Έπειτα αναλύονται το πρωτόκολλο SSH και το σύστημα ελέγχου εκδόσεων Git. Τέλος, γίνεται μνεία στη βάση δεδομένων PostgreSQL και στο πρόγραμμα gejson.io.



Εικόνα 12: Η επιφάνεια εργασίας του Raspbian

## 4.1 Python



Η Python είναι μια διερμηνευόμενη, αντικειμενοστρεφής, υψηλού επιπέδου γλώσσα προγραμματισμού με δυναμική σημασιολογία. Η υψηλού επιπέδου κατασκευή της σε δομές δεδομένων, σε συνδυασμό με τη δυναμική τυποποίηση και την δυναμική διασύνδεση, την καθιστούν αρκετά ελκυστική για Ταχεία Ανάπτυξη Εφαρμογών, όπως για τη χρήση της ως μια προγραμματιστική ή ενσωματώσιμη γλώσσα για να συνδέει υπάρχοντα κομμάτια μαζί. Η Python είναι απλή, εύκολη στην εκμάθηση του τρόπου που συντάσσεται ενώ δίνει έμφαση στην αναγνωσιμότητα και επομένως μειώνει το κόστος της συντήρησης των προγραμμάτων. Η Python υποστηρίζει δομές και πακέτα, τα οποία βοηθούν στη διαμόρφωση του προγράμματος και στην επαναχρησιμοποίηση του κώδικα. Ο διερμηνέας Python και η εκτεταμένη κύρια βιβλιοθήκη είναι διαθέσιμα σε κώδικα ή σε δυαδική μορφή χωρίς χρέωση για όλες τις μεγάλες πλατφόρμες, και μπορούν να διανέμονται ελεύθερα.

Ο Guido van Rossum συνέλαβε την ιδέα της Python στα τέλη της δεκαετίας του 1980 ενώ η εφαρμογή της ξεκίνησε το Δεκέμβριο του 1989 στο CWI στην Ολλανδία, ως διάδοχος της γλώσσας προγραμματισμού ABC. Αρχικά, η Python ήταν γλώσσα scripting που χρησιμοποιούνταν στο λειτουργικό σύστημα Amoeba, ικανή και για κλήσεις συστήματος. Η Python 2.0 κυκλοφόρησε στις 16 Οκτωβρίου του 2000. Στις 3 Δεκεμβρίου 2008 κυκλοφόρησε η έκδοση 3.0. Πολλά από τα καινούργια χαρακτηριστικά αυτής της έκδοσης έχουν μεταφερθεί στις εκδόσεις 2.6 και 2.7 που είναι προς τα πίσω συμβατές.

Η Python 3 είναι ιστορικά η πρώτη γλώσσα προγραμματισμού που σπάει την προς τα πίσω συμβατότητα με προηγούμενες εκδόσεις ώστε να διορθωθούν κάποια λάθη που υπήρχαν σε προγενέστερες εκδόσεις και να καταστεί ακόμα πιο σαφής ο απλός τρόπος με τον οποίο μπορούν να γίνουν κάποια πράγματα.

Αξίζει να σημειωθεί ότι το όνομα της γλώσσας προέρχεται από την ομάδα άγγλων κωμικών Μόντυ Πάιθον.

Η Python είναι μια από εκείνες τις σπάνιες γλώσσες που ισχυρίζονται ότι είναι και απλές και ισχυρές. Ένα στοιχείο της συγκεκριμένης γλώσσας που την κάνει αρκετά ελκυστική είναι

πως επιτρέπει στον προγραμματιστή να συγκεντρώνεται στη λύση ενός προβλήματος παρά στο συντακτικό και τη δομή της.

Τα βασικά χαρακτηριστικά της Python που την καθιστούν ως ένα εξαιρετικό εργαλείο προγραμματισμού, είναι τα εξής:

- **Απλή:** Η Python είναι μια απλή και μινιμαλιστική γλώσσα. Η ομοιότητα της Python με ψευδοκώδικα είναι ένα από τα πιο ισχυρά σημεία της. Επιτρέπει στον προγραμματιστή να συγκεντρώνεται στη λύση του προβλήματος αντί στην ίδια τη γλώσσα.
- **Εύκολη στην εκμάθηση:** Είναι εξαιρετικά απλό να ξεκινήσει κάποιος με την Python. Η Python έχει μια ασυνήθιστα απλή σύνταξη, όπως έχει ήδη αναφερθεί.
- **Ελεύθερη και ανοικτού κώδικα:** Η Python είναι ένα παράδειγμα *ΕΛΛΑΚ* (Ελεύθερο Λογισμικό και Λογισμικό Ανοικτού Κώδικα). Με απλά λόγια, μπορούν να διανέμονται αντίγραφα αυτού του λογισμικού, να διαβάζεται ο πηγαίος κώδικάς του, να γίνονται αλλαγές σ' αυτόν και να χρησιμοποιούνται κομμάτια του σε νέα ελεύθερα προγράμματα. Το ΕΛΛΑΚ βασίζεται στην ιδέα μιας κοινότητας που μοιράζεται τη γνώση. Αυτός είναι ένας από τους λόγους για τους οποίους η Python είναι τόσο καλή -δημιουργήθηκε και βελτιώνεται συνεχώς από μια κοινότητα που το μόνο που θέλει είναι μια καλύτερη Python.
- **Γλώσσα υψηλού επιπέδου:** Όταν γράφει κάποιος προγράμματα στην Python, δε χρειάζεται να νοιάζεται για τις χαμηλού επιπέδου λεπτομέρειες όπως η διαχείριση της μνήμης που χρησιμοποιείται από τα προγράμματά, κλπ.
- **Φορητή:** Λόγω του ανοικτού της κώδικα, η Python έχει υλοποιηθεί (δηλαδή αλλάχθηκε για να λειτουργεί) σε πολλές πλατφόρμες. Όλα τα Python προγράμματά μπορούν να δουλέψουν σε οποιαδήποτε από αυτές τις πλατφόρμες χωρίς να χρειάζονται καθόλου αλλαγές αρκεί να έχει αποφευχθεί η χρησιμοποίηση χαρακτηριστικών που εξαρτώνται από κάθε σύστημα.
- **Διερμηνευόμενη:** Ένα πρόγραμμα που γράφεται σε μια μεταγλωττιζόμενη γλώσσα όπως η C ή η C++ μετατρέπεται από την πηγαία γλώσσα σε γλώσσα μηχανής, χρησιμοποιώντας ένα μεταγλωττιστή με διάφορες σημαίες και επιλογές. Έπειτα ο συνδέτης αντιγράφει το πρόγραμμα στη μνήμη και έπειτα το εκτελεί. Από την άλλη η Python δεν χρειάζεται μεταγλώττιση σε δυαδικό αρχείο. Εσωτερικά, μετατρέπει τον

πηγαίο κώδικα σε μια ενδιάμεση μορφή που ονομάζεται bytecode έπειτα το μεταφράζει στη γλώσσα του υπολογιστή και μετά το εκτελεί.

- **Αντικειμενοστρεφής:** Η Python υποστηρίζει τόσο το διαδικασιοστρεφή προγραμματισμό (procedure-oriented) όσο και τον αντικειμενοστρεφή προγραμματισμό (object-oriented). Στο διαδικασιοστρεφή προγραμματισμό, το πρόγραμμα δομείται πάνω σε διαδικασίες ή συναρτήσεις οι οποίες δεν είναι τίποτε άλλο από επαναχρησιμοποιήσιμα κομμάτια από προγράμματα. Στις αντικειμενοστρεφείς γλώσσες, τα προγράμματα δομούνται πάνω σε αντικείμενα τα οποία συνδυάζουν δεδομένα και λειτουργικότητα. Η Python έχει έναν πολύ ισχυρό αλλά πολύ απλό τρόπο για αντικειμενοστρεφή προγραμματισμό, ειδικά όταν συγκρίνεται με μεγάλες γλώσσες όπως η C++ ή η Java.
- **Ενσωματώσιμη-Επεκτάσιμη:** Η Python μπορεί να ενσωματωθεί μέσα στα προγράμματα σε C/C++ για να δώσει δυνατότητες 'scripting' για τους χρήστες των προγραμμάτων, ενώ ένα κρίσιμο κομμάτι κώδικα υπάρχει η δυνατότητα να υλοποιηθεί σε C/C++ και έπειτα να χρησιμοποιηθεί από το Python πρόγραμμα.
- **Εκτεταμένες Βιβλιοθήκες:** Η Πρότυπη βιβλιοθήκη της Python είναι πραγματικά τεράστια. Μπορεί να βοηθήσει στη πραγματοποίηση διαφόρων πραγμάτων σχετικά με κανονικές εκφράσεις, δημιουργία τεκμηρίωσης, δοκιμές μονάδων, νημάτωση, βάσεις δεδομένων, περιηγητές ιστού, CGI, FTP, email, XML, XML-RPC, HTML, κρυπτογράφηση, γραφικές διεπαφές χρήστη (GUI -graphical user interfaces) και άλλα πράγματα που εξαρτώνται από το σύστημα. Όλα αυτά είναι διαθέσιμα όποτε είναι εγκατεστημένη η Python. Επιπλέον από την πρότυπη βιβλιοθήκη, υπάρχουν διάφορες άλλες βιβλιοθήκες υψηλής ποιότητας οι οποίες χρησιμοποιήθηκαν στην παρούσα διπλωματική και στις οποίες αναφερόμαστε παρακάτω.

Όλα τα ανωτέρω χαρακτηριστικά καθιστούν τη γλώσσα Python ένα εξαιρετικό εργαλείο για τη για την υλοποίηση όλων των προγραμμάτων που αφορούν την παρούσα διπλωματική εργασία.

Ακολούθως αναφερόμαστε στις βιβλιοθήκες και modules που χρησιμοποιήθηκαν. Αρχικά όσον αφορά τα προεγκατεστημένα modules χρησιμοποιούμε ευρέως τα:

- **psycopg2:** Πρόκειται για την πιο δημοφιλή βιβλιοθήκη σε Python για προσαρμογή βάσεων δεδομένων PostgreSQL. Υλοποιεί το DB API 2.0 specification της Python και

παρέχει την απαραίτητη ασφάλεια έτσι ώστε διαφορετικά νήματα-threads να διαμοιράζονται την ίδια σύνδεση.

- **csv:** Το συγκεκριμένο module υλοποιεί τις κλάσεις για read και write operations σε csv format.
- **json:** Επιτρέπει το encoding και το decoding σε json τύπο αρχείων.

Καθώς και πλήθος βιβλιοθηκών που εγκαταστάθηκαν χειροκίνητα με κυριότερες τις:

- **Python-OBD:** Η συγκεκριμένη βιβλιοθήκη διαθέτει τα απαραίτητα εργαλεία για διαχείριση των δεδομένων που επιστρέφει η θύρα OBD. Επιστρέφει real-time μετρήσεις όπως φαίνεται στον παρακάτω πίνακα, κάνει διαγνωστικό έλεγχο του αυτοκινήτου και παρέχει συμβατότητα με το Raspberry Pi καθώς και με τους ELM327 OBD-II adapters.
- **python3-mickrostacknode:** Πρόκειται για τις απαραίτητες βιβλιοθήκες σε Python 3 για την επικοινωνία με το L80 GPS και το MMA8452 Accelerometer της Mickrostack.

## 4.2 Raspbian



Το Raspbian είναι ένα ελεύθερο Λειτουργικό Σύστημα βασισμένο στο Debian βελτιστοποιημένο έτσι ώστε να προσαρμόζεται με το υλικό του Raspberry Pi. Είναι διαθέσιμο με πάνω από 35.000 πακέτα λογισμικού ή προ-μεταγλωττισμένα προγράμματα σε κατάλληλη μορφή για εύκολη εγκατάσταση στο Raspberry Pi. Η αρχική έκδοση ολοκληρώθηκε τον Ιούνιο του 2012, αλλά η διανομή συνεχίζει να αναπτύσσεται ενεργά με έμφαση στη βελτίωση της σταθερότητας και της αποδόσης σε όσο το δυνατόν περισσότερα πακέτα λογισμικών του Debian. Έχει προεγκατεστημένες τις Python, Java, Mathematica κ.ά. Στη παρούσα διπλωματική χρησιμοποιήσαμε το Raspbian Jessie (έκδοση 8.0-βασισμένο στο Debian Jessie) και έκδοση πυρήνα 4.1.

### 4.3 OpenStreetMap

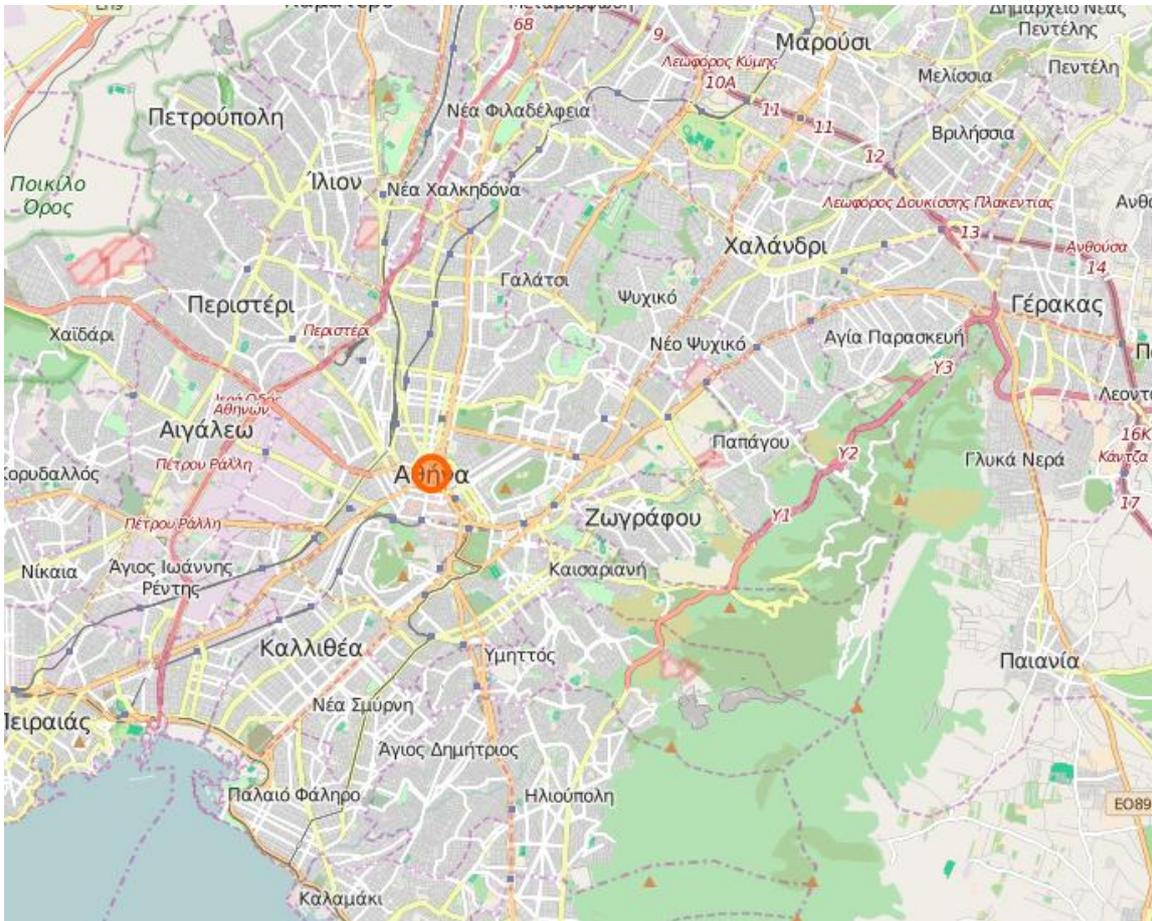


Το **OpenStreetMap** (OSM) είναι ένας χάρτης με ελεύθερη άδεια ο οποίος αναπτύσσεται από μια κοινότητα εθελοντών που συνεισφέρουν και διατηρούν δεδομένα σχετικά με δρόμους, μονοπάτια, καφετέριες, σιδηροδρομικούς σταθμούς, και πολλά περισσότερα, σε όλον τον κόσμο. Δημιουργήθηκε από τον Steve Coast στη Μεγάλη Βρετανία το 2004 εμπνευσμένο από την μεγάλη επιτυχία της Wikipedia και πλέον έχει πάνω από 2 εκατομύρια εγγεγραμμένους χρήστες. Οι συνεισφέροντες χρησιμοποιούν αεροφωτογραφίες, συσκευές GPS, και τοπικούς χάρτες χαμηλής τεχνολογίας για να σιγουρευτούν πως το OSM είναι ακριβής και ενημερωμένο στο μικρότερο δυνατό επίπεδο. [32]

Το OSM χρησιμοποιεί μια τοπολογική δομή δεδομένων, με τέσσερα βασικά στοιχεία γνωστά και ως αρχέγονα δεδομένα (data primitives).

- **Κόμβοι (nodes):** Είναι τα σημεία με μία γεωγραφική θέση και αποθηκεύονται ως συντεταγμένες (ζεύγη ενός γεωγραφικού μήκους και πλάτους). Εκτός από τη χρήση τους στους δρόμους, χρησιμοποιούνται για να αναπαραστήσουν στοιχεία του χάρτη δίχως μέγεθος, όπως κορυφές βουνών είτε σημεία ενδιαφέροντος.
- **Αρτηρίες (ways):** Οι αρτηρίες είναι διατεταγμένες λίστες από κόμβους που διαμορφώνουν μια πολυγωνική γραμμή. Χρησιμοποιούνται τόσο για να αναπαραστήσουν γραμμικά στοιχεία, όπως δρόμους και ποτάμια όσο και περιοχές, όπως δάση, λίμνες, πάρκα και θέσεις στάθμευσης.
- **Σύνδεσμοι (relations):** Πρόκειται για διατεταγμένες λίστες από αρτηρίες και κόμβους που χρησιμοποιούνται για να αναπαραστήσουν σχέσεις μεταξύ τους. Μερικά παραδείγματα είναι οι απαγορεύσεις στροφής σε αρτηρίες καθώς και διαδρομές οι οποίες περιλαμβάνουν διάφορους δρόμους.
- **Ετικέτες (tags):** Είναι ζεύγη κλειδιού-τιμής και χρησιμοποιούνται για την αποθήκευση μεταδεδομένων (metadata) του χάρτη, όπως ο τύπος, το όνομα και οι φυσικές ιδιοτητές των οντοτήτων που τον απαρτίζουν δηλαδή των κόμβων, των αρτηριών ή των συνδέσμων.

Στην παρούσα διπλωματική εργασία χρησιμοποιούμε και τον όρο τμήμα δρόμου (segment ή καλύτερα way-segment) για να αναπαραστήσουμε δύο κόμβους που ενώνονται από μία ακριβώς αρτηρία.



Διάγραμμα 5: Παράδειγμα OSM χάρτη

#### 4.4 Map-Matching και Barefoot

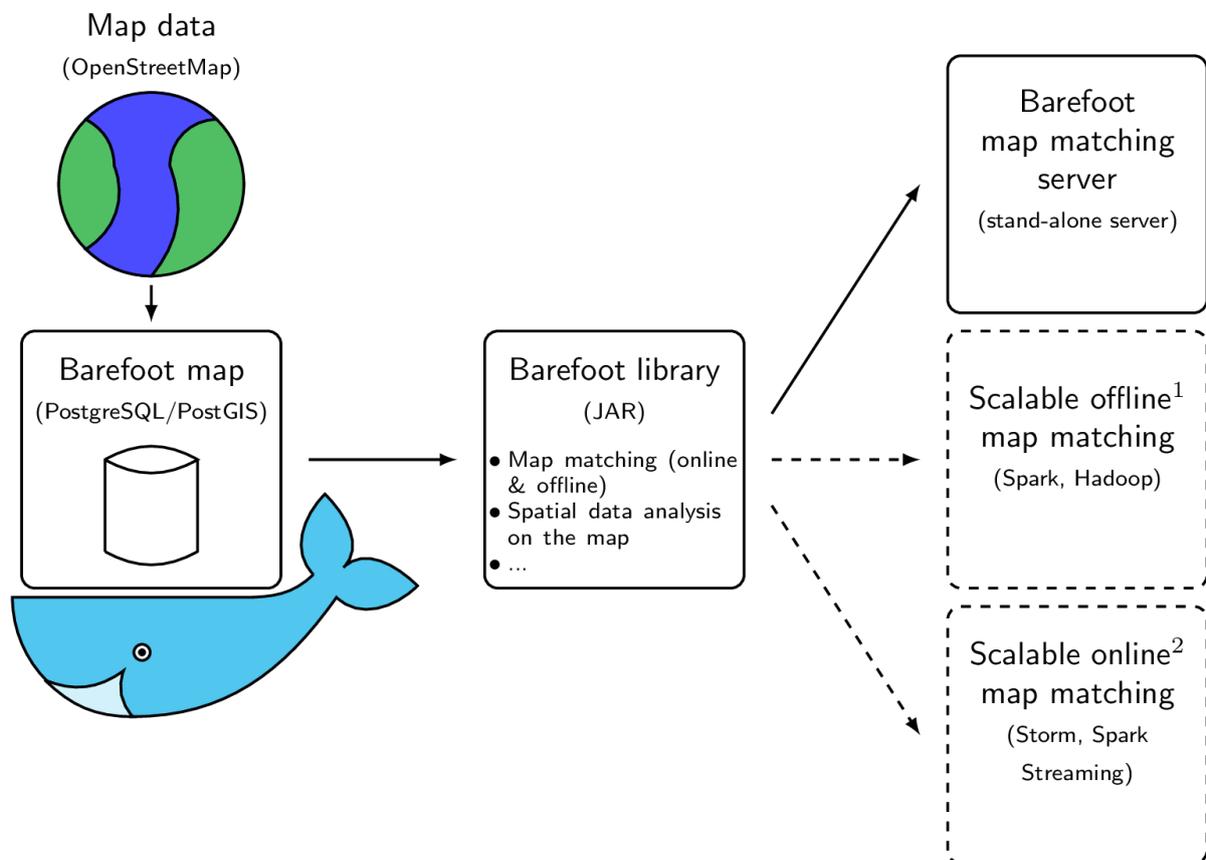
Με τον όρο Map-Matching αναφερόμαστε σε εκείνη τη διαδικασία κατά την οποία ένα σύνολο γεωγραφικών σημείων (τα οποία έχουν ληφθεί για παράδειγμα από μια GPS καταγραφή) αντιστοιχίζονται στο χάρτη. Πρόκειται για μία πολυσύνθετη διαδικασία κυρίως λόγω των σφαλμάτων του GPS αλλά και του τρόπου αναπαράστασης του χάρτη.

Η λειτουργία του map-matching είναι κομβικής σημασίας για την εφαρμογή μας, καθώς πρέπει να γνωρίζουμε το ακριβές τμήμα δρόμου (segment) που ανήκει ένα καταγραφθέν

σημείο για να προβούμε στην εξαγωγή στατιστικών στοιχείων ανά οδικό σύνδεσμο (per-segment analysis).

Δοκιμάστηκαν διάφοροι αλγόριθμοι για την ταυτοποίηση της θέσης στο χάρτη με κυριότερους: το Graphhopper[22], το Barefoot και τη Snap to Roads[23] μέθοδο του Roads API της Google. Επιλέξαμε το Barefoot και οι λόγοι που συνετέλεσαν σε αυτήν μας την επιλογή αναλύονται στο κεφάλαιο των σχεδιαστικών επιλογών (5.1.v).

Το Barefoot είναι μία βιβλιοθήκη ανοιχτού κώδικα γραμμένη σε Java η οποία υλοποιεί map matching καθώς και χωρική ανάλυση δεδομένων σε χάρτες OpenStreetMap. Παρέχει ένα πλήθος γεωμετρικών και χωρικών συναρτήσεων ενώ είναι σχεδιασμένη για χρήση σε παράλληλα και κατανεμημένα συστήματα. Το Barefoot χρησιμοποιεί έναν αλγόριθμο HMM (Hidden Markov Model) για το map matching μαζί με OpenStreetMap δεδομένα, περισσότερες πληροφορίες για τον οποίο διατίθενται στα [8] και [20]. Υποστηρίζει τόσο online map matching, δηλαδή αντιστοίχιση σε πραγματικό χρόνο, όσο και offline map matching, δηλαδή αντιστοίχιση μιας καταγραφής GPS δεδομένων στο χάρτη σε μη πραγματικό χρόνο. Το δομικό διάγραμμα του Barefoot φαίνεται στο ακόλουθο σχήμα:



Διάγραμμα 6: Διάγραμμα Αρχιτεκτονικής του Barefoot

Η παραμετροποίηση και η εγκατάσταση του map-matching εξυπηρετητή γίνεται σε container (Docker), ενώ το Barefoot μπορεί να στηθεί είτε ως ένας αυτόνομος εξυπηρετητής, είτε ως ένας κεντρικός χάρτης-αποθετήριο (repository) για κατανεμημένα συστήματα.

#### 4.4.1 Docker



Το Docker (Ντόκερ) είναι μια πλατφόρμα λογισμικού ανοιχτού κώδικα που υλοποιεί Εικονικοποίηση (Virtualization) σε επίπεδο Λειτουργικού Συστήματος. Ουσιαστικά το Docker προσφέρει αυτοματοποιημένες διαδικασίες για την ανάπτυξη εφαρμογών σε απομονωμένες Περιοχές Χρήστη (User Spaces) που ονομάζονται Software Containers. Το λογισμικό χρησιμοποιεί τεχνολογίες του πυρήνα του Linux όπως τα cgroups και οι χώροι ονομάτων πυρήνα (kernel namespaces), για να επιτρέπει σε ανεξάρτητα software containers να εκτελούνται στο ίδιο λειτουργικό σύστημα. Έτσι αποφεύγεται η χρήση επιπλέον υπολογιστικών πόρων που θα απαιτούσε μια εικονική μηχανή (virtual machine). [31]

#### 4.5 Okeanos



Το ~okeanos είναι μια cloud υπηρεσία του ΕΔΕΤ για όλα τα μέλη της εκπαιδευτικής και ερευνητικής κοινότητας.[10] Πρόκειται για μία νέα IaaS (Infrastructure as a Service) υπηρεσία που δίνει τη δυνατότητα στους χρήστες, φοιτητές, καθηγητές και ερευνητές, να δημιουργούν τους δικούς τους υπολογιστές, πάντα συνδεδεμένους στο Internet, χωρίς την ανησυχία για προβλήματα υλικού, θέματα συνδεσιμότητας και προβλήματα λογισμικού.

Το ~okeanos αποτελείται από τις υπηρεσίες "cyclades" και "pithos+".

Η υπηρεσία "cyclades" δίνει τη δυνατότητα στους χρήστες να δημιουργήσουν εικονικές μηχανές (virtual machines) με σύνδεση στο Internet και συγκεκριμένες τεχνικές προδιαγραφές. Μέσα σε λίγα λεπτά ο χρήστης μπορεί να επιλέξει λειτουργικό σύστημα,

επεξεργαστική ισχύ, και μέγεθος μνήμης RAM που θα διαθέτει η εικονική μηχανή. Οι χρήστες μπορούν επίσης να διασυνδέσουν τις εικονικές μηχανές τους μέσω τοπικών εικονικών δικτύων (virtual networks).

Η υπηρεσία "rithos+" παρέχει δυνατότητα αποθήκευσης στο σύννεφο (cloud) με άμεση πρόσβαση στα αρχεία από οπουδήποτε και οποτεδήποτε. Εξυπηρετεί επίσης τις ανάγκες αποθήκευσης των εικονικών μηχανών που έχουν δημιουργηθεί.

Στο παρακάτω διάγραμμα φαίνεται το Εικονικό Μηχάνημα που στήσαμε στην υπηρεσία Okeanos:



Διάγραμμα 7: Okeanos Server

## 4.6 SSH

Το SSH είναι ένα πρωτόκολλο που παρέχει ασφαλή απομακρυσμένη σύνδεση σε υπολογιστές πάνω από μη ασφαλές δίκτυο. Αποτελείται από τρία βασικά στοιχεία:

- Το Transport layer protocol παρέχει πιστοποίηση της ταυτότητας του server, ακεραιότητα των δεδομένων και εξασφάλιση του απόρρητου της συναλλαγής. Προαιρετικά μπορεί να εφαρμόσει και συμπίεση δεδομένων. Τυπικά τρέχει πάνω από μία TCP/IP σύνδεση.
- Το User Authentication protocol πιστοποιεί την ταυτότητα του πελάτη-χρήστη στον server. Τρέχει πάνω από το Transport layer protocol.
- Το Connection protocol πολυπλέκει το κρυπτογραφημένο φυσικό κανάλι σε αρκετά λογικά κανάλια και τρέχει πάνω από το User Authentication protocol.

### 4.6.1 Ιδιωτικά Και Δημόσια Κλειδιά

Κάθε server και client πρέπει να έχει ένα ζευγάρι ιδιωτικής – δημόσιας κλειδας για να μπορεί να επαληθεύσει την ταυτότητα του στο άλλο άκρο. Επιτρέπεται η κατοχή περισσότερων του ενός ζευγάρια κλειδιών, όταν χρησιμοποιούνται με διαφορετικούς αλγόριθμους, ενώ η από κοινού χρήση ενός ζεύγους από πολλούς server δεν απαγορεύεται.

Για να μπορεί ο client με ευκολία να επαληθεύσει την ταυτότητα του server είναι απαραίτητο να γνωρίζει την δημόσια κλειδα που αντιστοιχεί στον server που θέλει να συνδεθεί. Υπάρχουν δυο διαφορετικά μοντέλα που εξασφαλίζουν την προηγούμενη προϋπόθεση:

- Πρώτον, ο client έχει αποθηκευμένα σε μια τοπική βάση δεδομένων τα ονόματα των server και τις σχετιζόμενες με αυτά δημόσιες κλειδες. Αυτή η μέθοδος δεν απαιτεί μια κεντρική διαχείριση των κλειδιών από τρίτους. Το μειονέκτημα είναι ότι το μέγεθος μιας τέτοιας βάσης δεδομένων μπορεί να εξελιχθεί σημαντικά και συνεπώς η συντήρηση της να γίνει δύσκολη.
- Στην δεύτερη περίπτωση, σχέση μεταξύ του ονόματος του server και του κλειδιού του πιστοποιείται από μια άξια εμπιστοσύνης Certification Authority. Το πρόγραμμα του πελάτη γνωρίζει μόνο την δημόσια κλειδα της Certification Authority και μπορεί να επιβεβαιώσει την εγκυρότητα των κλειδών που έχουν πιστοποιηθεί από την CA. Εδώ

δεν υπάρχει το πρόβλημα της διατήρησης μεγάλων βάσεων δεδομένων από τα τοπικά συστήματα, αφού μόνο ένα κλειδί χρειάζεται να αποθηκεύει ο client. Από την άλλη μεριά, όμως, δεν είναι δυνατή η απόλυτη εμπιστοσύνη στις διαδικασίες της Certification Authority. Επίσης, πιστοποίηση κάθε κλειδιού μπορεί να είναι μια χρονοβόρα και περίπλοκη διαδικασία.

Οι εφαρμογές του SSH μπορούν να παρέχουν επιπρόσθετες μεθόδους επικύρωσης των δημόσιων κλειδιών, όπως για παράδειγμα την παραγωγή ενός δεκαεξαδικού "αποτυπώματος" της κλειδας και από τα δύο άκρα και την σύγκριση τους μέσω εξωτερικών καναλιών επικοινωνίας (π.χ. τηλέφωνο). Κλειδες που δεν επαληθεύονται, κανονικά δεν πρέπει να γίνονται δεκτές.

Έγινε εκτεταμένη χρήση του πρωτοκόλλου ssh τόσο για τη σύνδεση στον πελάτη Raspberry, όσο και για την σύνδεση στον εξυπηρετητή στην υπηρεσία okeanos. Παράλληλα, χρησιμοποιήθηκε κρυπτογράφηση δημοσίου κλειδιού για την επικοινωνία client και server.

#### 4.6.2 SCP

Το SCP πρωτόκολλο (Secure Copy Protocol) είναι ένα διαδικτυακό πρωτόκολλο το οποίο υποστηρίζει μεταφορά αρχείων μεταξύ υπολογιστών σε ένα δίκτυο. Το SCP χρησιμοποιεί Secure Shell (SSH) για τη μεταφορά δεδομένων, ως επίσης χρησιμοποιεί τους ίδιους μηχανισμούς πιστοποίησης με αποτέλεσμα να εξασφαλίζεται η αυθεντικότητα και η εμπιστευτικότητα κατά τη μεταφορά δεδομένων. Ένας πελάτης μπορεί να ανεβάσει (upload) αρχεία σε ένα διακομιστή, συμπεριλαμβανομένων προαιρετικά των βασικών χαρακτηριστικών τους (άδειες, χρονικές σημάνσεις). Οι πελάτες μπορούν επίσης να ζητήσουν αρχεία ή καταλόγους από ένα διακομιστή (download). Το SCP τρέχει πάνω από TCP και η προεπιλεγμένη θύρα που ακούει είναι η 22. Το Linux έχει προεγκατεστημένο το πρόγραμμα scp και η βασική του χρήση δίδεται από την ακόλουθη εντολή:

```
scp τοπικό_αρχείο όνομα_χρήστη@απομακρυσμένος_υπολογιστής:τοποθεσία
```

Μέσω SCP αποστέλλεται η καταγραφή που επιτελείται στον πελάτη – Raspberry Pi στον εξυπηρετητή που στήθηκε στην υπηρεσία Okeanos.

## 4.7 Git



Το Git είναι ένα σύστημα ελέγχου εκδόσεων (λέγεται και σύστημα ελέγχου αναθεωρήσεων ή σύστημα ελέγχου πηγαίου κώδικα) με έμφαση στην ταχύτητα, στην ακεραιότητα των δεδομένων και στην υποστήριξη για κατανεμημένες μη γραμμικές ροές εργασίας. Το Git σχεδιάστηκε και αναπτύχθηκε αρχικά από τον Λίνους Τόρβαλντς για τη ανάπτυξη του πυρήνα Linux το 2005 και έχει γίνει από τότε το πιο διαδεδομένο σύστημα ελέγχου εκδόσεων για ανάπτυξη λογισμικού.

Όπως τα περισσότερα άλλα κατανεμημένα συστήματα ελέγχου εκδόσεων/αναθεωρήσεων και αντίθετα με τα περισσότερα συστήματα πελάτη-διακομιστή, κάθε κατάλογος εργασίας του Git είναι ένα ολοκληρωμένο αποθετήριο λογισμικού με πλήρες ιστορικό και δυνατότητες πλήρους παρακολούθησης της έκδοσης, ανεξάρτητα από την πρόσβαση δικτύου ή ενός κεντρικού διακομιστή. Όπως ο πυρήνας Λίνουξ, το Git είναι Ελεύθερο λογισμικό που διανέμεται κάτω από τους όρους της έκδοσης 2 της Γενικής Άδειας Δημόσιας Χρήσης GNU.

## 4.8 PostgreSQL



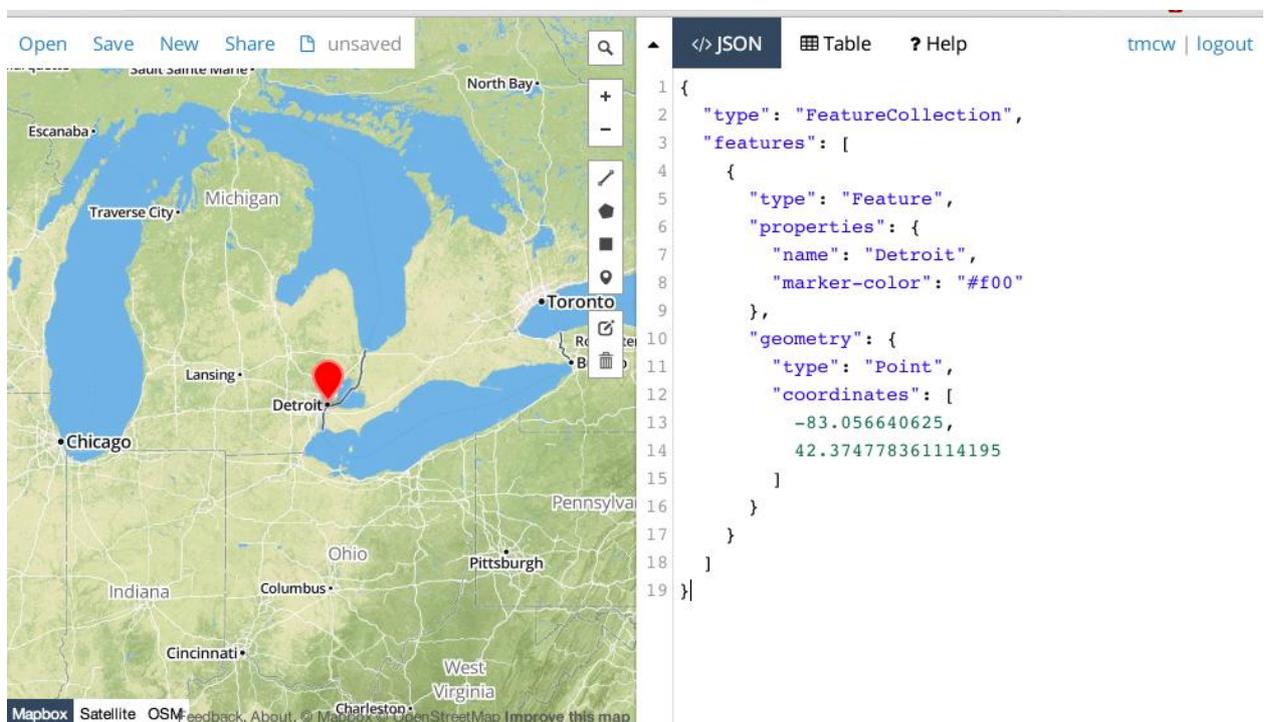
Η PostgreSQL είναι ένα σχεσιακό σύστημα διαχείρισης βάσεων δεδομένων. Η ανάπτυξη της διαρκεί πάνω από δύο δεκαετίες και βασίζεται σε μια αποδεδειγμένα καλή αρχιτεκτονική η οποία έχει δημιουργήσει μια ισχυρή αντίληψη των χρηστών γύρω από την αξιοπιστία, την ακεραιότητα και την ορθή λειτουργία. Υποστηρίζει ένα μεγάλο μέρος του πρότυπου SQL και προσφέρει πολλά σύγχρονα χαρακτηριστικά όπως: σύνθετα ερωτήματα, ξένα κλειδιά, triggers, views και transactional integrity. Η PostgreSQL βασίζεται στο μοντέλο πελάτη/εξυπηρετητή ενώ μπορεί να διαχειριστεί μεγάλο αριθμό ταυτόχρονων χρηστών καθώς και μεγάλο όγκο δεδομένων. Διαθέτει περιβάλλοντα προγραμματισμού για C, C++,

Java, Perl, Python, Ruby και τρέχει σε όλα τα λειτουργικά συστήματα Linux, Unix, MacOS X, Windows. Έχει σχεδιαστεί για να ικανοποιήσει τις ανάγκες των χρηστών, από απλό γράψιμο SQL έως την ανάπτυξη πολύπλοκων βάσεων δεδομένων.

Παράλληλα, πάνω στην PostgreSQL (απαντάται και ως Postgres) έχουν υλοποιηθεί μια σειρά από επεκτάσεις ικανές για την διαχείριση και την αποθήκευση γεωγραφικών δεδομένων. Οι επεκτάσεις αυτές είναι γνωστές με το όνομα PostGIS. Η PostGIS δίδει τη δυνατότητα στην Postgres να υποστηρίζει γεωγραφικά αντικείμενα. Επί της ουσίας, χρησιμοποιεί τον Server της Postgres επιτρέποντας του να λειτουργεί σαν μία back-end χωρική βάση δεδομένων για γεωγραφικά πληροφοριακά συστήματα (Geographic Information Systems), υποστηρίζοντας μια σειρά από γεωγραφικούς τελεστές για επερωτήσεις γεωγραφικών δεδομένων.

## 4.9 geojson.io

Το geojson.io είναι ένα γρήγορο και απλό εργαλείο για δημιουργία, προβολή και κοινή χρήση χαρτών. Πήρε το όνομα του από το GeoJSON, μια μορφή δεδομένων ανοιχτού κώδικα σχεδιασμένη για την αναπαράσταση γεωγραφικών δεδομένων, και υποστηρίζει πέρα από GeoJSON και δεδομένα τύπου KML, GPX, CSV κ.ά.



Εικόνα 13: geojson.io

Προγραμματιστικά μπορεί κανείς να αλληλεπιδράσει με το Geojson.io με δύο τρόπους:

- Μέσω παραμέτρων στη διεύθυνση της ιστοσελίδας (URL)
- Μέσω της κονσόλας του περιηγητή (Browser Console)

Στην παρούσα διπλωματική χρησιμοποιούμε το εργαλείο `geojsonio.py` γραμμένο σε Python, το ανοίγει αυτόματα περιηγητή και απεικονίζει τα GeoJSON δεδομένα στο `geojson.io`.

Η χρήση του είναι αρκετά απλή:

```
from geojsonio import display

with open('map.geojson') as f:
    contents = f.read()
    display(contents)
```

## 5. Αρχιτεκτονική

Στο κεφάλαιο αυτό, αναφερόμαστε στην αρχιτεκτονική του συστήματος που υλοποιήσαμε. Παρουσιάζουμε και αναλύουμε τις σχεδιαστικές επιλογές και περιγράφουμε αναλυτικά τις λειτουργίες του κάθε υποσυστήματος. Αναφερόμαστε αρχικά στον πελάτη Raspberry Pi και έπειτα στον εξυπηρετητή που στήθηκε στην υπηρεσία okeanos.

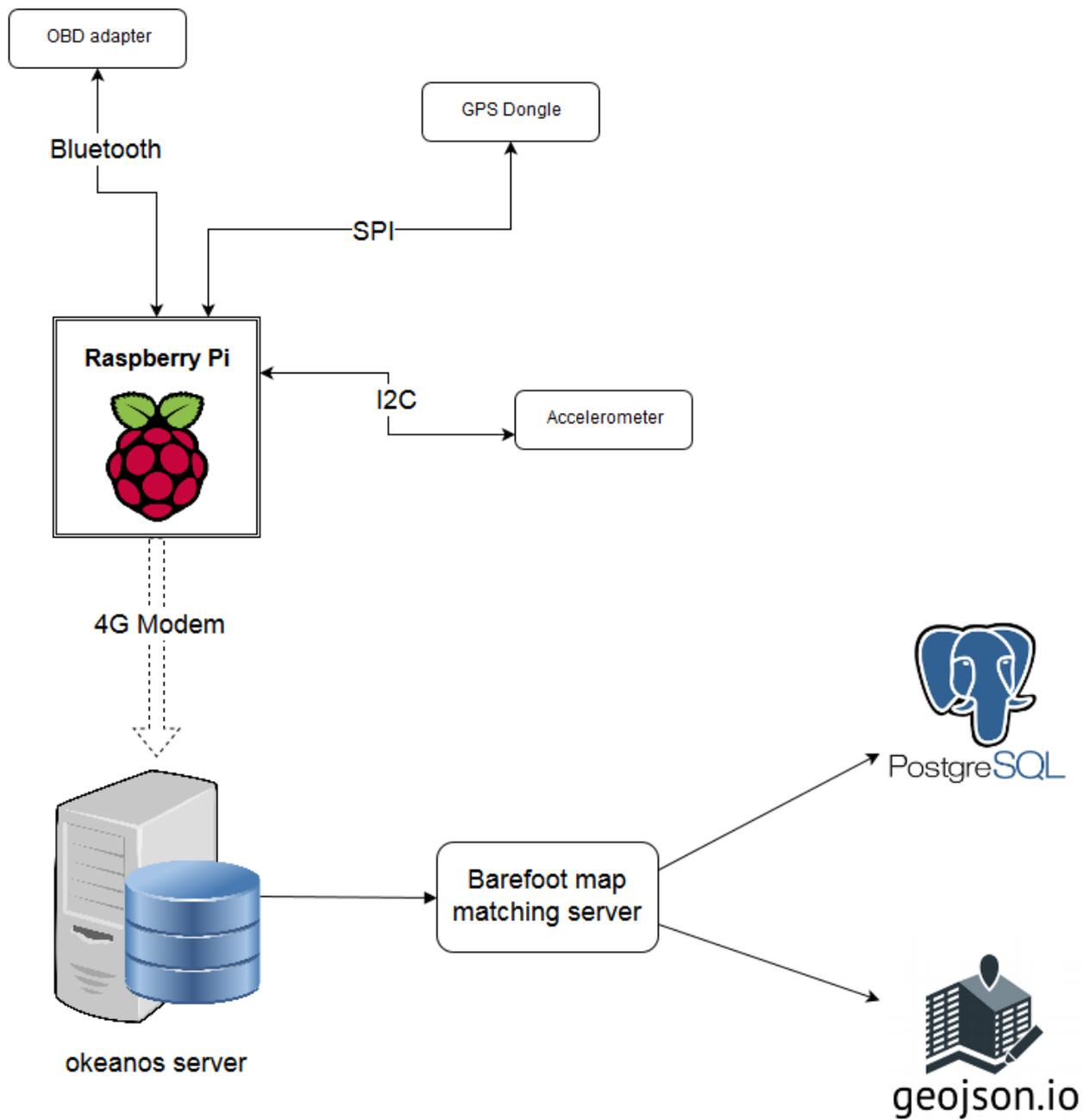
Το Raspberry Pi συνδέεται με τρία υποσυστήματα που επιτελούν τις ακόλουθες λειτουργίες:

- OBD adapter (καταγράφει πραγματικού χρόνου μετρήσεις διαγνωστικών οχήματος)
- GPS dongle (καταγράφει μετρήσεις θέσης του οχήματος)
- Accelerometer (καταγράφει μετρήσεις επιτάχυνσης για το όχημα)

Ενώ, ο εξυπηρετητής είναι υπεύθυνος για τη λειτουργία των:

- Barefoot map-matching Server (πρόγραμμα που κάνει το map-matching)
- PostgreSQL Server (εξυπηρετητής βάσης δεδομένων)
- Geojson.io (πρόγραμμα απεικόνισης γεωγραφικών δεδομένων στο χάρτη)

Στην επόμενη σελίδα παρατίθεται το διάγραμμα αρχιτεκτονικής του συστήματος όπου διαφαίνονται τα ανωτέρω συστατικά στοιχεία της εφαρμογής:



Διάγραμμα 8: Διάγραμμα Αρχιτεκτονικής Συστήματος

## 5.1 Σχεδιαστικές Επιλογές

Κατά τη διάρκεια υλοποίησης της εφαρμογής προβήκαμε σε συγκεκριμένες σχεδιαστικές επιλογές τις οποίες πραγματευόμαστε στο κεφάλαιο αυτό. Ακόμη, παρουσιάζουμε τους λόγους που συνετέλεσαν στη διαμόρφωση των επιλογών μας.

- i. Η εφαρμογή μας καταγράφει συνολικά τα ακόλουθα στοιχεία: ημερομηνία και ώρα καταγραφής από το Pi, κατανάλωση (calculated engine load), στροφές κινητήρα (rpm) και ταχύτητα αυτοκινήτου (speed) από το OBD adapter, γεωγραφικό μήκος (longitude), πλάτος (latitude), ύψος πάνω από το μέσο επίπεδο της θάλασσας (AMSL), οριζόντια ακρίβεια (HDOP) και ταχύτητα GPS (speed) από το GPS, και τέλος επιτάχυνση κατά τους 3 άξονες από το επιταχυνσιόμετρο. Η εφαρμογή είναι εύκολα κλιμακώσιμη ώστε να καταγράφονται, ιδιαίτερα από το OBD adapter όπως δείχνουμε και στον παρακάτω πίνακα, ακόμα περισσότερα δεδομένα. Για τις ανάγκες της υλοποίησής αρκούν τα παραπάνω δεδομένα.

Mode 01		
PID	Name	Description
00	PIDS_A	Supported PIDs [01-20]
01	STATUS	Status since DTCs cleared
02	<i>unsupported</i>	<i>unsupported</i>
03	FUEL_STATUS	Fuel System Status
04	ENGINE_LOAD	Calculated Engine Load
05	COOLANT_TEMP	Engine Coolant Temperature
06	SHORT_FUEL_TRIM_1	Short Term Fuel Trim - Bank 1
07	LONG_FUEL_TRIM_1	Long Term Fuel Trim - Bank 1
08	SHORT_FUEL_TRIM_2	Short Term Fuel Trim - Bank 2
09	LONG_FUEL_TRIM_2	Long Term Fuel Trim - Bank 2
0A	FUEL_PRESSURE	Fuel Pressure
0B	INTAKE_PRESSURE	Intake Manifold Pressure
0C	RPM	Engine RPM
0D	SPEED	Vehicle Speed
0E	TIMING_ADVANCE	Timing Advance
0F	INTAKE_TEMP	Intake Air Temp
10	MAF	Air Flow Rate (MAF)
11	THROTTLE_POS	Throttle Position

Διάγραμμα 9: Υποστηρίζομενες εντολές από την Python-OBd βιβλιοθήκη

- ii. Η εκκίνηση της διαδικασίας καθώς και η συχνότητα των μετρήσεων καθορίζεται από το GPS dongle. Πρόκειται για το πιο ευαίσθητο σύστημα από τα τρία υποσυστήματα που συνδέονται στο Raspberry Pi. Επιλέγουμε, επομένως, η διαδικασία να ξεκινήσει 2 λεπτά αφού εκκινήσει το Raspberry Pi, χρόνος (υπερ)αρκετός για να πάρει την πρώτη μέτρηση (first fix) το GPS αν τοποθετηθεί το Raspberry Pi σε κατάλληλο σημείο (κοντά σε παράθυρο). Ακόμα, η συχνότητα των μετρήσεων επηρεάζεται από το GPS. Θέλουμε την ίδια χρονική στιγμή να καταγράφονται δεδομένα από το OBD, το GPS και το accelerometer και υπάρχουν περιπτώσεις που το GPS χάνει το σήμα του (πχ. μέσα σε μια υπόγεια σήραγγα). Ως αποτέλεσμα, επιλέγουμε το GPS να ορίζει κάθε πότε θα καταγράφονται μετρήσεις στο Raspberry Pi, με προεπιλογή τα 2 δευτερόλεπτα συχνότητα.
- iii. Οι βιβλιοθήκες για το OBD adapter είναι γραμμένες σε Python 2, ενώ οι βιβλιοθήκες της Microstack για το GPS dongle και το accelerometer είναι γραμμένες σε Python 3. Το πρόβλημα που ανακύπτει προέρχεται από το γεγονός ότι η Python 3 δεν είναι προς τα πίσω συμβατή (backwards compatibility problem) με την Python 2, με αποτέλεσμα να χρειαζόμαστε ένα ειδικό μηχανισμό για το συγχρονισμό των μετρήσεων από τα διαφορετικά υποσυστήματα. Επιλέγουμε να χρησιμοποιήσουμε σήματα του Linux (signals) για το συγχρονισμό των προγραμμάτων λόγω της απλότητας στην χρήση τους, σε σχέση με κάποιον άλλον μηχανισμό όπως κλειδώματα, σημαφόροι ή διασωλήνωση. Χρησιμοποιούμε τα modules subprocess, signal και os της Python. Συγκεκριμένα, υλοποιούμε μια διαδικασία συγχρονισμού μεταξύ των προγραμμάτων obd.py και gps\_acc.py την οποία αναλύουμε στο επόμενο κεφάλαιο.
- iv. Χρησιμοποιούμε πόρους του okeanos για να υλοποιήσουμε τον απομακρυσμένο εξυπηρετητή. Το στήσιμο εικονικού μηχανήματος είναι ιδιαίτερα απλή διαδικασία στην υπηρεσία okeanos@grnet, ενώ μας δίνεται και η δυνατότητα εύκολης καταστροφής και επαναδημιουργίας του μηχανήματος σε περίπτωση σφάλματος. Παράλληλα, οι πόροι που μας παρέχονται δωρεάν από την υπηρεσία είναι παραπάνω από αρκετοί για το στήσιμο ενός αρκετά γρήγορου εξυπηρετητή.
- v. Για την υλοποίηση του map-matching δοκιμάσαμε διάφορα εργαλεία με κυριότερα το ανοικτού-κώδικα graphhopper, τη Snap to Roads μέθοδο του Roads API της Google καθώς και το Barefoot. Το Graphhopper υλοποιεί γρήγορα τη διαδικασία του map-

matching αλλά παρουσιάζει λιγότερο ακριβή αποτελέσματα σε σχέση με το Barefoot. Μια συγκριτική μελέτη φαίνεται στον παρακάτω χάρτη. Το API της Google εμφανίζει αρκετά ακριβή αποτέλεσμα αλλά θέτει όρια στη χρήση του και πρόκειται για κλειστού-κώδικα λογισμικό. Προτιμάται το Barefoot έναντι των άλλων εργαλείων, το οποίο υλοποιεί State-of-the-Art αλγόριθμο map-matching, είναι open-source βιβλιοθήκη και παρουσιάζει ιδιαίτερα ακριβή αποτελέσματα.

- vi. Χρησιμοποιούμε το εικονικό μηχάνημα (Virtual Machine) στον Okeanos για την υλοποίηση του map-matching. Το VM μας έχει 6GB RAM και επεξεργαστή QEMU Virtual CPU version 2.1.2 @2.1GHz, έναντι του 1GB RAM και 900MHz quad-core ARM Cortex-A7 CPU που έχει το Raspberry Pi. Ενδεικτικά, αναφέρουμε ότι ένα αρχείο με περίπου 500 σε πλήθος καταγραφές έκανε 24 περίπου λεπτά να ολοκληρωθεί η διαδικασία του map-matching με Barefoot, ενώ στο εικονικό μηχάνημα στον okeanos έκανε περίπου 1 λεπτό.
- vii. Χρησιμοποιούμε την ανοικτού κώδικα βάση δεδομένων PostgreSQL καθώς πρόκειται για βάση δεδομένων με ισχυρή υποστήριξη τύπων γεωγραφικών δεδομένων. Προτιμήθηκε έναντι άλλων βάσεων δεδομένων καθότι χρησιμοποιείται κατά κόρον σε Συστήματα Διαχείρισης Βάσεων Δεδομένων που έχουν δυνατότητες υποστήριξης χωρικών τύπων.
- viii. Για την αναπαράσταση δεδομένων στο χάρτη χρησιμοποιούμε το εργαλείο Geojson.io καθότι είναι ιδιαίτερα εύκολο στη χρήση του, χρησιμοποιεί OpenStreetMaps, είναι open project και σε συνδυασμό με το geojsonio.py δίδει δυνατότητα αυτοματοποιημένης απεικόνισης γεωγραφικών δεδομένων και στατιστικών στοιχείων στο χάρτη.
- ix. Χρησιμοποιούμε τη λειτουργία tethering των "έξυπνων" κινητών για την επικοινωνία του Pi με τον Server. Η πλειοψηφία των χρηστών του δρόμου έχουν στη διάθεσή τους ένα smartphone το οποίο μπορεί να χρησιμοποιηθεί ως φορητό σημείο πρόσβασης. Αποφεύγουμε τη χρήση ενός 4G modem λόγω κόστους και πιθανώς επιπλέον υλικού που χρειάζεται να εγκατασταθεί (powered usb hub) για να λειτουργήσει.

## 5.2 Πελάτης - Raspberry Pi

Στο παρόν κεφάλαιο παρουσιάζουμε αναλυτικά τη διαδικασία που ακολουθείται ώστε να συνδεθεί το Raspberry Pi με το OBD adapter, το GPS Dongle και το επιταχυνσιόμετρο. Όπου κρίνεται απαραίτητο παραθέτουμε κομμάτια κώδικα για την καλύτερη κατανόηση της διαδικασίας.

### 5.2.1 OBD adapter

Απαραίτητη προϋπόθεση για την επικοινωνία του Raspberry Pi με το OBD adapter είναι η σύνδεση τους μέσω Bluetooth. Η διαδικασία αυτή επιτυγχάνεται με τη βοήθεια του παρακάτω script:

```
#!/bin/sh
sudo hciconfig hci0 piscan
sudo rfcomm bind /dev/rfcomm0 XX:XX:XX:XX:XX:XX 1
```

Η πρώτη εντολή: `hciconfig hci0 piscan` θέτει τις διεπαφές του Bluetooth της συσκευής εντοπίσιμες από άλλες συσκευές. Η δεύτερη εντολή δεσμεύει την πόρτα `/dev/rfcomm0` για τη συσκευή με MAC address `XX:XX:XX:XX:XX:XX`. Η σύνδεση και επικοινωνία επιτυγχάνεται από το πρόγραμμα `obd.py`.

Τοποθετούμε το OBD adapter στη θύρα OBD-II του αυτοκινήτου. Το πρόγραμμα `obd.py` γραμμένο σε **Python** είναι υπεύθυνο για τη σύνδεση του Raspberry Pi με το OBD adapter. Ενδεικτικά, παραθέτουμε τις ακόλουθες γραμμές κώδικα που επεξηγούν πως λειτουργεί η βιβλιοθήκη Python-OBD:

```
import obd #Εισάγεται η βιβλιοθήκη Python-OBD

#Επιστρέφει μια λίστα από τις διαθέσιμες σειριακές πόρτες
ports = obd.scanSerial()

#Συνδέεται στην πρώτη διαθέσιμη πόρτα
connection = obd.OBD(ports[0])

#Επιστρέφει τις τρέχουσες στροφές του κινητήρα
rpm = connection.query(obd.commands[1][12])

#Επιστρέφει την τρέχουσα ταχύτητα του αυτοκινήτου
speed = connection.query(obd.commands[1][13])
```

## 5.2.2 GPS Dongle

Το GPS dongle συνδέεται με το Raspberry Pi μέσω SPI σύνδεσης. Το πρόγραμμα `gps_acc.py` γραμμένο σε **Python 3** είναι υπεύθυνο για τη σύνδεση του Raspberry Pi με το GPS Dongle. Το ακόλουθο κομμάτι κώδικα καταδεικνύει πως λειτουργεί η βιβλιοθήκη της Microstack για το GPS L80:

```
import microstacknode.hardware.gps.l80gps

'''
Εισαγωγή της βιβλιοθήκης της Microstack για το L80 GPS
Η βιβλιοθήκη είναι γραμμένη σε Python 3
'''

#Δημιουργείται αντικείμενο GPS
gps = microstacknode.hardware.gps.l80gps.L80GPS()

#Επιστρέφονται δεδομένα gprgga σε μορφή dictionary
gprgga = gps.get_gprgga()

#Επιστρέφονται δεδομένα gprmc σε μορφή dictionary
gprmc = gps.get_gprmc()

#Επιστρέφεται η ταχύτητα του GPS - και μετατροπή της από κόμβους σε km/h
km_h = float(gprmc['speed'])*1.852 #1knot=1.852km/h

#Εκτυπώνεται στην οθόνη το γεωμετρικό πλάτος με ακρίβεια 5 δεκαδικών ψηφίων
print('{:.5f}'.format(gprgga['latitude']))

#Εκτυπώνεται το γεωγραφικό μήκος
print('{:.5f}'.format(gprgga['longitude']))

#Εκτυπώνεται η ταχύτητα του GPS
print('{:.1f}'.format(km_h))
```

### 5.2.3 Επιταχυνσιόμετρο (Accelerometer)

Το Accelerometer συνδέεται με το Raspberry Pi μέσω I2C σύνδεσης. Το πρόγραμμα `gps_acc.py` είναι υπεύθυνο και για την επικοινωνία του Raspberry Pi με το Accelerometer (πέρα από την επικοινωνία με το GPS όπως είδαμε προηγουμένως). Παραθέτουμε και εδώ κώδικα που εξηγεί τη λειτουργία της βιβλιοθήκης της Microstack για το MMA8452Q accelerometer.

```
from microstacknode.hardware.accelerometer.mma8452q import MMA8452Q
import time
'''
Εισαγωγή των απαραίτητων βιβλιοθηκών
Η βιβλιοθήκη για το accelerometer είναι γραμμένη σε Python 3
'''

#δημιουργία αντικειμένου accelerometer
with MMA8452Q() as accelerometer:
    accelerometer.standby() #standby για την αποθήκευση των αλλαγών
    accelerometer.set_g_range(2) #2G ή 4G ή 8G
    accelerometer.activate() #ενεργοποίηση του accelerometer
    time.sleep(2) #απαραίτητο χρονικό διάστημα για να πάρει μετρήσεις

    #Επιστρέφει τις τιμές της επιτάχυνσης σε m/s^2
    ms = accelerometer.get_xyz_ms2()

'''
Εκτυπώνεται με ακρίβεια 2 δεκαδικών ψηφίων η επιτάχυνση στους x, y, z
άξονες
'''

print('{:.2f}'.format(ms['x']))
print('{:.2f}'.format(ms['y']))
print('{:.2f}'.format(ms['z']))
```

Χρειαζόμαστε ακόμα ένα μηχανισμό, ώστε οι μετρήσεις που εξάγονται από τα τρία υποσυστήματα να πραγματοποιούνται την ίδια χρονική στιγμή (timestamp), με περίοδο έστω 2 δευτερολέπτων. Αυτό θα μπορούσε κάλλιστα να επιτευχθεί αν είχαμε ένα script στο οποίο διαδοχικά οι εντολές θα γράφονταν σε ψευδοκώδικα κάπως έτσι:

```
while 1:
```

```
get_data_from_OBD()
get_data_from_GPS()
get_data_from_ACC()
sleep 2
```

Το πρόβλημα που ανακύπτει είναι ότι οι βιβλιοθήκες για το OBD adapter είναι γραμμένες σε Python 2, ενώ οι βιβλιοθήκες της Microstack είναι γραμμένες σε Python 3, γεγονός που μας οδηγεί σε πιο σύνθετη υλοποίηση για τον συγχρονισμό των μετρήσεων. Επιλέγουμε να χρησιμοποιήσουμε σήματα του Linux για την ολοκλήρωση της διαδικασίας. Θέτουμε το πρόγραμμα `obd.py` ως διεργασία-πατέρα η οποία πέρα από την άντληση δεδομένων από τη θύρα OBD του αυτοκινήτου είναι πλέον υπεύθυνη για το συγχρονισμό των μετρήσεων. Υλοποιούμε σχήμα συγχρονισμού κατά την οποία το `obd.py` επιτελεί τις ακόλουθες λειτουργίες:

- Καλεί την διεργασία-παιδί `gps_acc.py` και κρατάει το PID της με:

```
proc = subprocess.Popen(["python3", "gps_acc.py", FILE])
c_pid = proc.pid
```

όπου `FILE`, το αρχείο που καταγράφονται οι μετρήσεις και δίδεται ως όρισμα όταν καλείται το `obd.py`.

- Ελέγχει αν η διεργασία-παιδί έχει σταματήσει με:

```
os.waitpid(c_pid, os.WUNTRACED)
```

- Στέλνει στη διεργασία-παιδί σήμα για να συνεχίσει τη λειτουργία της με:

```
os.kill(c_pid, signal.SIGCONT)
```

Η διεργασία-παιδί `gps_acc.py` από την άλλη επιτελεί τις ακόλουθες λειτουργίες πέρα από την καταγραφή των δεδομένων θέσης και επιτάχυνσης:

- Σταματάει τον εαυτό της μόλις λάβει τις μετρήσεις με:

```
os.kill(os.getpid(), signal.SIGSTOP)
```

Με αποτέλεσμα να επαναφέρεται πλέον ο έλεγχος στην διεργασία-πατέρα, η οποία «ξεμπλοκάρει» και παίρνει μετρήσεις τις οποίες καταγράφει στο αρχείο `FILE`.

- Δέχεται σήμα SIGCONT από τον πατέρα και καταγράφει στο αρχείο FILE τις μετρήσεις που είχε λάβει πριν το SIGSTOP που είχε στείλει η ίδια στον εαυτό της.

Η διαδικασία αυτή εξασφαλίζει τη σύγχρονη άντληση μετρήσεων από τα 3 υποσυστήματα. Οι αναλυτικοί κώδικες φαίνονται στο Παράρτημα.

Με το που ολοκληρωθεί η διαδικασία των μετρήσεων, τα δεδομένα αποστέλλονται στον εξυπηρετητή με:

```
scp $raw_data user@<server_IP_public_address>:/home/user/Templates/records
```

όπου raw\_data, το σύνολο των δεδομένων που έχει καταγραφεί και server\_IP\_public\_address, είναι η δημόσια IP που μας παρέχεται από την υπηρεσία Okeanos για το εικονικό μας μηχάνημα.

```
date,time,cel,rpm,speed,acc_ped_pos,eng_fuel_rate,lat,long,alt,hdop,gps_speed,a_x,a_y,a_z
2016-03-20,21:23:51.781,3.52941176471,1670.0,40, None, None, 38.01750,23.79680,156.4,4.55,40.3,2.10,-2.04,8.31
2016-03-20,21:23:53.277,0.0,1559.0,37, None, None, 38.01745,23.79691,156.8,3.73,40.8,1.75,-1.65,8.46
2016-03-20,21:23:54.671,19.6078431373,1517.0,36, None, None, 38.01741,23.79703,157.3,3.73,39.6,2.08,-2.82,9.75
2016-03-20,21:23:56.064,5.09803921569,1478.0,35, None, None, 38.01738,23.79715,157.7,3.73,37.1,2.82,-2.38,9.86
2016-03-20,21:23:57.497,0.392156862745,1391.0,33, None, None, 38.01736,23.79727,157.9,4.55,37.2,0.83,-2.54,11.06
2016-03-20,21:23:58.803,5.49019607843,1057.0,28, None, None, 38.01734,23.79739,158.1,4.55,36.8,0.48,-3.06,9.30
2016-03-20,21:24:00.132,8.62745098039,817.0,21, None, None, 38.01732,23.79750,158.3,3.73,36.1,0.57,-0.49,9.79
2016-03-20,21:24:01.444,9.01960784314,802.0,15, None, None, 38.01730,23.79762,158.6,3.75,34.4,1.29,-1.87,9.31
2016-03-20,21:24:02.785,8.62745098039,803.0,10, None, None, 38.01729,23.79773,158.9,3.73,30.7,1.36,-1.58,9.53
2016-03-20,21:24:04.134,9.41176470588,789.0,4, None, None, 38.01728,23.79782,159.2,3.73,27.2,0.46,-1.78,9.82
2016-03-20,21:24:05.404,8.23529411765,812.0,1, None, None, 38.01727,23.79791,159.5,3.73,19.3,1.49,-2.61,9.32
2016-03-20,21:24:06.683,36.862745098,1316.0,0, None, None, 38.01726,23.79798,159.8,3.73,14.1,2.21,-2.75,8.52
2016-03-20,21:24:07.941,37.2549019608,862.0,1, None, None, 38.01727,23.79803,160.1,3.73,10.8,1.70,-3.22,9.34
2016-03-20,21:24:09.191,32.5490196078,1433.0,6, None, None, 38.01727,23.79807,160.3,3.73,6.4,2.93,-4.01,8.14
2016-03-20,21:24:10.461,29.0196078431,1495.0,12, None, None, 38.01728,23.79809,160.4,3.75,0.2,2.27,-3.67,8.00
2016-03-20,21:24:11.733,34.5098039216,2297.0,19, None, None, 38.01729,23.79810,160.4,3.73,0.1,2.15,-3.60,9.28
2016-03-20,21:24:12.989,0.0,1939.0,21, None, None, 38.01730,23.79811,160.5,3.75,0.0,1.18,-2.05,9.73
2016-03-20,21:24:14.247,7.05882352941,956.0,20, None, None, 38.01730,23.79810,160.2,3.73,0.0,1.49,-3.22,9.49
2016-03-20,21:24:15.510,24.3137254902,1246.0,20, None, None, 38.01730,23.79810,160.1,3.73,0.0,1.81,-3.36,9.99
2016-03-20,21:24:16.767,50.5882352941,1558.0,25, None, None, 38.01730,23.79810,160.2,3.75,0.3,1.82,-4.23,9.60
2016-03-20,21:24:18.020,30.1960784314,1841.0,29, None, None, 38.01730,23.79810,160.2,3.75,0.5,1.44,-3.27,8.45
2016-03-20,21:24:19.325,34.1176470588,2027.0,32, None, None, 38.01730,23.79811,160.3,3.75,14.8,1.47,-2.37,8.26
2016-03-20,21:24:20.585,37.6470588235,2278.0,35, None, None, 38.01730,23.79815,160.4,3.75,20.0,1.75,-4.27,8.92
2016-03-20,21:24:21.860,7.45098039216,2378.0,37, None, None, 38.01730,23.79822,160.5,3.75,21.6,0.52,-2.64,10.92
2016-03-20,21:24:23.161,10.1960784314,2266.0,35, None, None, 38.01730,23.79829,160.7,3.75,21.0,1.24,-2.35,8.68
2016-03-20,21:24:24.466,48.2352941176,2444.0,38, None, None, 38.01731,23.79836,160.8,3.75,20.3,0.90,-3.18,7.41
2016-03-20,21:24:25.786,0.0,1613.0,38, None, None, 38.01731,23.79844,160.9,3.75,22.5,0.62,-3.45,8.75
2016-03-20,21:24:27.078,37.2549019608,1598.0,38, None, None, 38.01732,23.79851,161.1,3.75,26.0,2.39,-3.40,9.87
2016-03-20,21:24:28.377,44.7058823529,1717.0,41, None, None, 38.01733,23.79860,161.3,3.75,28.8,1.80,-2.40,7.27
2016-03-20,21:24:29.631,34.1176470588,1788.0,43, None, None, 38.01734,23.79869,161.4,3.75,30.6,2.44,-3.37,10.13
2016-03-20,21:24:30.895,36.4705882353,1859.0,44, None, None, 38.01736,23.79879,161.6,3.75,33.4,1.72,-2.66,7.71
2016-03-20,21:24:32.197,7.8431372549,1820.0,43, None, None, 38.01737,23.79890,161.7,3.75,34.9,1.15,-3.21,9.59
```

**Διάγραμμα 10: Παράδειγμα αρχείου καταγραφής**

## 5.3 Εξυπηρετητής – Virtual Machine@oceanos.grnet.gr

Στον Server μας τρέχει συνεχώς script στο παρασκήνιο το οποίο είναι υπεύθυνο για τις παρακάτω λειτουργίες:

- Αναγνωρίζει αν αφίχθει έγκυρο αρχείο καταγραφής στον κατάλογο:

```
/home/user/Templates/records
```

Η λειτουργία αυτή επιτυγχάνεται με τη χρήση της inotifywait, εργαλείο των linux για τον εντοπισμό αλλαγών σε αρχεία που χρησιμοποιεί την inotify.

Σε περίπτωση που το αρχείο που παραλήφθει είναι έγκυρο:

- Μετατρέπει τα ακατέργαστα δεδομένα (raw data) που είναι σε μορφή csv, σε μορφή json μέσω του προγράμματος csv\_to\_json.py, διαδικασία απαραίτητη καθότι το πρόγραμμα Barefoot παίρνει ως είσοδο json-formatted δεδομένα.
- Αποστέλλει το json αρχείο στον map-matching server, ώστε να παραχθούν τα ταυτοποιημένα στο χάρτη δεδομένα (snapped-to-map data). Τα εξαγόμενα δεδομένα από το Barefoot είναι και αυτά σε μορφή json.
- Τα ταυτοποιημένα δεδομένα, σε συνδυασμό με τα αρχικά ακατέργαστα δεδομένα ενοποιούνται κατάλληλα με το slimjson\_to\_csv\_unified.py και δημιουργείται αρχείο csv το οποίο χρησιμοποιείται για να: Αποθηκευτούν στη βάση δεδομένων στατιστικά στοιχεία ανά οδικό σύνδεσμο μέσω του προγράμματος connect\_to\_data.py και να απεικονιστεί αυτοματοποιημένα στο χάρτη η διαδρομή με τις ιδιότητες τις, μέσω των προγραμμάτων geojson.io και geojsonio.py.

### 5.3.1 Map Matching Server – Barefoot

#### 5.3.1.1 Στήσιμο του map matching Server

Για το στήσιμο του Barefoot map Server χρησιμοποιήσαμε τη μέθοδο του Docker container, η οποία εγκαθιστά τον PostgreSQL/PostGIS database server στο απομονωμένο περιβάλλον του container.

Αρχικά κατεβάζουμε το Barefoot με:

```
git clone https://github.com/bmwcarit/barefoot.git
```

Έπειτα στήνουμε την εικόνα του Docker (build Docker image) με:

```
cd barefoot  
sudo docker build --rm=true -t barefoot ./docker
```

και δημιουργούμε τον Docker container με:

```
sudo docker run -t -i -p 127.0.0.1:5432:5432 --name="barefoot-greece" -v  
${PWD}/bfmap/:/mnt/bfmap -v ${PWD}/docker/osm/:/mnt/osm barefoot
```

Κατεβάζουμε τον OSM χάρτη της Ελλάδας από την ιστοσελίδα [download.geofabrik.de](http://download.geofabrik.de) και τον τοποθετούμε στον κατάλογο `barefoot/docker/osm/` εξωτερικά του container. Ο κατάλογος αυτός είναι προσβάσιμος εντός του container από τη θέση `/mnt/osm/`.

```
curl http://download.geofabrik.de/europe/greece-latest.osm.pbf -o  
barefoot/docker/osm/greece.osm.pbf
```

Εκτελούμε το παρακάτω script (εντός του container) για να εισάγουμε το OSM στο container χρησιμοποιώντας τις προεπιλεγμένες παραμέτρους. Για διαφορετικές παραμετροποιήσεις παραπέμπουμε στο άρθρο εγχειρίδιο χρήσης του Barefoot στο [20].

```
bash /mnt/osm/import.sh
```

Μεταγλώττιση των Barefoot JAR και εκκίνηση του map matching server:

```
cd barefoot  
mvn compile assembly:single  
java -jar target/barefoot-0.0.2-jar-with-dependencies.jar [--slimjson|--  
debug|--geojson] /path/to/server/properties /path/to/mapserver/properties
```

Μόλις στηθεί ο map-matching Server μπορούμε να τρέξουμε:

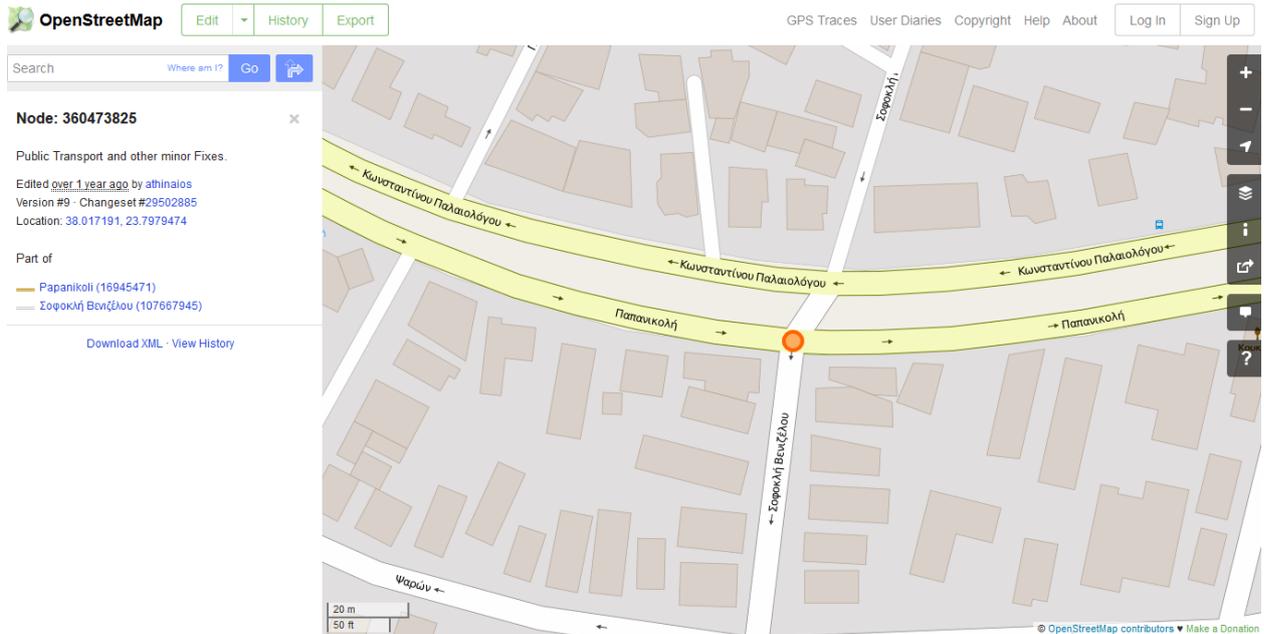
```
cat <input_json_file> | netcat 127.0.0.1 1234
```

ώστε να πάρουμε τα ταυτοποιημένα με το χάρτη αποτελέσματα στη μορφή που επιλέξαμε να στήσουμε τον Server. (slimjson, debug ή geojson)

### 5.3.1.2 Λειτουργίες που επιτελεί ο map matching server

Όπως προαναφέραμε τα ακατέργαστα δεδομένα μετατρέπονται από μορφή csv σε κατάλληλη μορφή json, ούτως ώστε να δοθούν ως είσοδο στον Barefoot Server. Εκεί επιτελείται η λειτουργία του map matching την οποία αναλύσαμε σε προηγούμενο κεφάλαιο. Κάθε ζεύγος γεωγραφικού μήκους και πλάτους που καταγράφηκε από το GPS, αντιστοιχίζεται στο χάρτη. Η έξοδος του Barefoot για κάθε αρχικό σημείο (γεωγραφικό μήκος, πλάτος) εμπεριέχει:

- **road:** Το δρόμο στον οποίο απεικονίζεται το σημείο. Πρόκειται για ακέραιο αριθμό που προσδιορίζει μονοσήμαντα σε ποιο δρόμο βρίσκεται το κινητό και έχει να κάνει με το πως αποθηκεύει το Barefoot τους δρόμους στη βάση δεδομένων του. Σημειώνουμε ότι αν ο αριθμός του δρόμου είναι άρτιος π.χ. 563902 το αυτοκίνητο κινείται προς μία κατεύθυνση, ενώ αν ο αριθμός του δρόμου είναι περιττός π.χ. 563903 τότε το αυτοκίνητο κινείται προς την αντίθετη κατεύθυνση.
- **route:** Το σημείο ή καλύτερα το ζεύγος σημείων στο οποίο αντιστοιχίζεται η αρχική καταγραφή. Για παράδειγμα:  
Τα διαδοχικά σημεία: (23.79691, 38.01745), (23.79703, 38.01741) αντιστοιχίζονται μέσω του Barefoot στο ζεύγος σημείων:  
"route": "LINESTRING(23.7968513 38.0174087, 23.7968844726 38.01739842)"  
"route": "LINESTRING (23.7968844726 38.01739842, 23.7970053 38.017360875)"  
Παρατηρούμε ότι για διαδοχικά αρχικά σημεία, το δεύτερο ζεύγος συντεταγμένων του πρώτου σημείου ταυτίζεται με το πρώτο ζεύγος συντεταγμένων του δεύτερου σημείου. Αυτό συμβαίνει για το σύνολο της καταγραφής, με αποτέλεσμα να δημιουργείται μια διαδρομή από σημεία.
- **fraction:** Το κλάσμα του δρόμου στο οποίο βρίσκεται το σημείο. Πρόκειται για αριθμό που παίρνει πραγματικές τιμές από 0 έως 1. Όσο μικρότερη η τιμή του, τόσο κοντύτερα βρισκόμαστε στην αρχή του δρόμου (road), ενώ όσο η τιμή του κλάσματος προσεγγίζει τη μονάδα τόσο εγγύτερα βρισκόμαστε στο τέλος του δρόμου.
- **source:** Ο κόμβος από τον οποίο εκκινεί ο δρόμος. Πρόκειται για ακέραιο αριθμό ο οποίος συμπίπτει με τον αριθμό κόμβου στον OSM χάρτη. (OSM node)
- **target:** Ο κόμβος στον οποίο τερματίζεται ο δρόμος. Αντιστοιχίζεται στον OSM node. Για παράδειγμα αν έχουμε "target":360473825, μπορούμε να πληκτρολογήσουμε σε περιηγητή: [www.openstreetmap.org/node/360473825](http://www.openstreetmap.org/node/360473825) και να πάρουμε αποτελέσματα σαν το ακόλουθο:



Εικόνα 14: Κόμβος στον OSM χάρτη

Προκύπτουν επομένως, τα ταυτοποιημένα στο χάρτη δεδομένα (snapped-to-map data) τα οποία μέσω του `slimjson_to_csv_unified.py` ενοποιούνται με τα αρχικά ακατέργαστα δεδομένα με σκοπό την εξαγωγή στατιστικών στοιχείων ανά οδικό σύνδεσμο και την απεικόνιση τους στο χάρτη.

### 5.3.2 Database Server – PostgreSQL

Η διαδικασία εξαγωγής στατιστικών στοιχείων ανά οδικό σύνδεσμο (per segment analysis) επιτελείται στο Database Server μέσω του προγράμματος `connect_to_data.py`. Για κάθε οδικό σύνδεσμο (ο οποίος ορίζεται μονοσήμαντα από το `road`, το `source` και το `target node`) το πρόγραμμα αυτό αποθηκεύει και αποστέλλει στη βάση δεδομένων τα ακόλουθα στοιχεία:

- **Χρόνο εισόδου:** Ο χρόνος σε μορφή HH:MM:SS.mmm στον οποίο το κινητό εισέρχεται για πρώτη φορά στον οδικό σύνδεσμο.
- **Χρόνο παραμονής:** Ο χρόνος σε ms όπου το κινητό παραμένει στον ίδιο οδικό σύνδεσμο.
- **Μέσες τιμές για δεδομένα οχήματος:** Κρατάει τις μέσες τιμές για στροφές κινητήρα ανά λεπτό (rpm), ταχύτητα σε km/h (speed), επί της εκατό κατανάλωση (cel – calculated engine load) και επιταχύνσεις σε  $m/s^2$  στους άξονες x, y (τους άξονες δηλαδή που δεν είναι κάθετοι στο επίπεδο κίνησης). Στη βάση φαίνονται ως `a_x_avg`, `a_y_avg`.

- **Μέγιστες και ελάχιστες τιμές για επιταχύνσεις:** Φαίνονται ως max\_a\_x, max\_a\_y, min\_a\_x, min\_a\_y στη βάση.

Παρακάτω δείχνουμε τον πίνακα testing από τη βάση δεδομένων per-segment.

Πληκτρολογούμε σε τερματικό: `SELECT * FROM TESTING;`

Και παίρνουμε το ακόλουθο αποτέλεσμα:

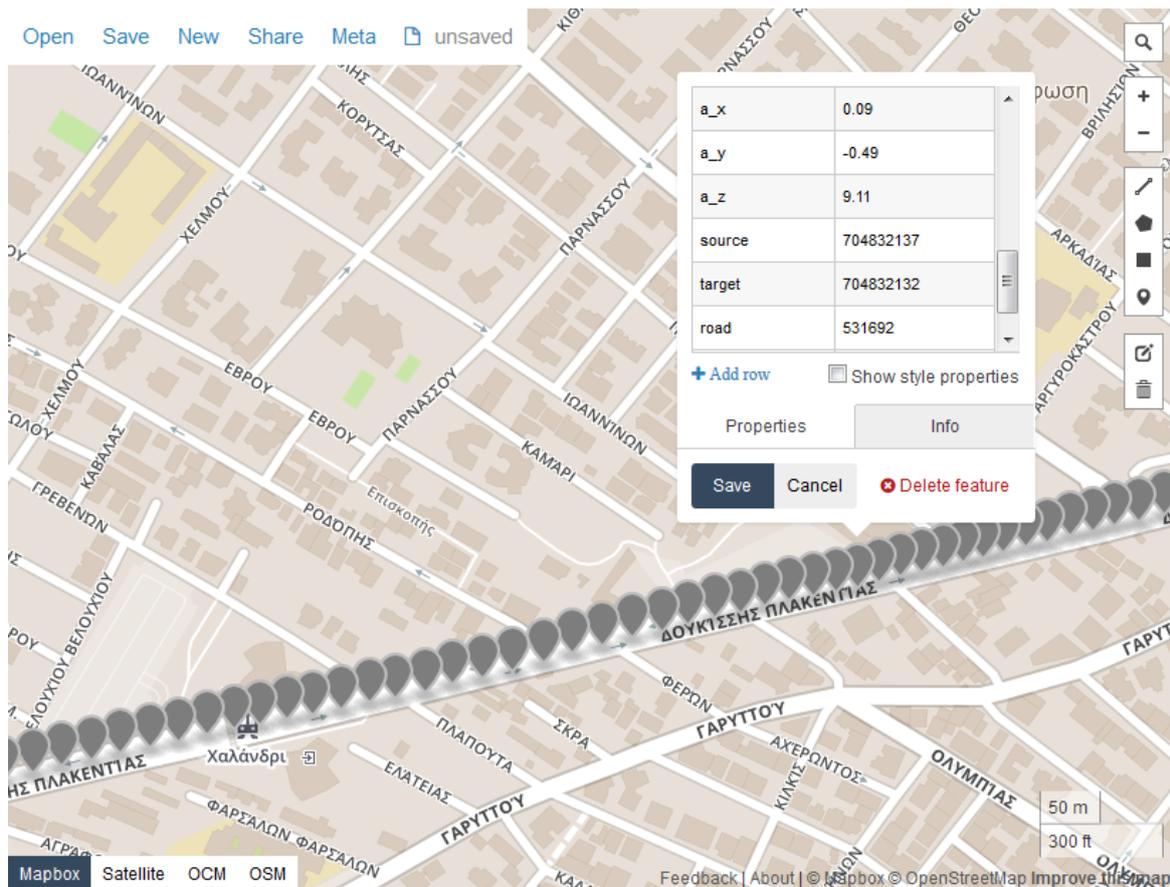
id	road	source_node	target_node	entry_time	travel_time	rpm	cel	speed	a_x_avg	a_y_avg	max_a_x	max_a_y	min_a_x	min_a_y
1	1080837	353244833	360473623	21:23:51.781	0	1670	3.53	40	2.1	-2.04	2.1	-2.04	2.1	-2.04
2	563902	360473623	360473825	21:23:53.277	12127	1102.5	7.45	22	1.31	-2.08	2.82	-3.06	1.75	-1.65
3	563904	360473825	360472856	21:24:06.683	30640	1756.84	27.65	29.88	1.7	-3.06	2.93	-4.27	2.21	-2.75
4	1616544	360472856	705990478	21:24:38.602	6406	1973.5	30.13	46.83	1.65	-2.6	2.44	-2.92	1.63	-1.87
5	1616546	705990478	360473834	21:24:46.271	5160	2211.4	27.14	52.8	1.69	-2.63	2.44	-2.77	1.75	-2.71
6	1616548	360473834	360473933	21:24:52.690	10212	2134.78	11.02	50.67	1.64	-2.81	2.52	-4.17	2.07	-2.37
7	1616550	360473933	360473974	21:25:04.171	9025	1431	12.11	29.88	0.53	-2.02	1.67	-2.99	1.42	-2.72
8	1616552	360473974	705979338	21:25:14.468	6380	1616.83	23.59	10.83	2.31	-3.65	3.06	-4.49	1.67	-3.47
9	1616554	705979338	705979340	21:25:22.112	8914	1869.5	40.93	37.75	2.34	-3.74	3.32	-4.74	2.41	-3.89
10	1616556	705979340	175678407	21:25:32.309	29419	1167.88	8.55	21.04	1.62	-2.05	3.42	-3.7	1.79	-2.97
11	1392944	175678407	3811346974	21:26:03.006	0	1220	36.86	10	3.64	-4.27	3.64	-4.27	3.64	-4.27
12	1392946	3811346974	360474589	21:26:04.303	15406	1860.62	26.67	34.62	1.77	-2.84	3.47	-3.83	2.42	-4.02
13	1392948	360474589	360474301	21:26:20.971	37091	1958.73	27.12	55.77	1.75	-2.78	4.47	0.22	1.6	-1.03
14	1392950	360474301	3400280295	21:26:59.351	11546	2084.9	19.92	68.3	1.63	-2.65	2.59	-3.81	2.11	-2.45
15	1392952	3400280295	360730493	21:27:12.189	0	1951	16.08	64	1.56	-2.76	1.56	-2.76	1.56	-2.76
16	1392954	360730493	175678410	21:27:13.455	14066	1793.08	26.01	67.67	1.6	-2.5	2.88	-3.29	2.88	-2.61
17	953862	175678410	705073963	21:27:28.834	2540	796.33	8.89	58	1.13	-1.63	1.98	-1.25	1.63	-2.4
18	953864	705073963	705073942	21:27:32.630	16672	1422.86	27.31	53.93	1.47	-2.89	2.58	-5.06	1.05	-2.18
19	953866	705073942	705073943	21:27:50.574	2534	2048	38.43	67.33	1.7	-2.6	2.06	-2.99	1.49	-2.57
20	953868	705073943	705073948	21:27:54.390	2597	2150.33	28.37	70.67	1.44	-2.89	1.71	-3.07	1.52	-3.13
21	953870	705073948	2186462701	21:27:58.299	7637	2081.29	12.66	68.43	1.3	-2.67	2.06	-2.93	0.23	-3.49
22	953872	2186462701	3403618793	21:28:07.219	0	1846	0	61	2.78	-1.86	2.78	-1.86	2.78	-1.86
23	1029738	3403618793	2186462704	21:28:08.486	0	1767	0	58	3.31	-1.75	3.31	-1.75	3.31	-1.75
24	330124	2186462704	2186462733	21:28:09.778	7678	1668.57	26.55	54.71	2.43	-2.41	2.84	-2.8	3.25	-2.6
25	531686	2186462733	705073998	21:28:18.718	0	1673	2.35	55	1.18	-1.77	1.18	-1.77	1.18	-1.77
26	531688	705073998	705074001	21:28:20.010	7654	1585	17.87	52	0.69	-2.85	1.47	-3.51	1.02	-2.7
27	531690	705074001	704832137	21:28:28.956	5138	2000.6	58.75	65.8	0.91	-4.13	1.92	-5	-0.28	-3.27
28	531692	704832137	704832132	21:28:35.368	6380	1414.17	11.57	68.33	0.95	-1.53	2.06	-2.08	1.98	-3.85
29	531694	704832132	246588491	21:28:43.030	11496	1293.1	22	29.2	1.48	-2.39	2.45	-4.25	0.74	-1.31
30	1890656	246588491	704832204	21:28:55.794	9055	1808.25	34.9	42.63	1.71	-3.27	2.55	-4.2	1.33	-3.85
31	1890658	704832204	581288189	21:29:06.118	19150	1593.81	15.86	47.38	2.1	-2.18	3.96	-3.62	2.18	-4.17
32	1890660	581288189	581288189	21:29:26.536	2577	1555.67	33.33	37.33	1.78	-2.23	2.44	-3.1	0.79	-2.7
33	1890662	581288189	579805587	21:29:30.379	8954	2054.75	36.52	49	2.9	-3.03	4.9	-4.7	3.74	-1.84
34	1213090	579805587	361441549	21:29:40.622	3856	1287.25	9.12	50.5	2.44	-2.31	3.37	-2.89	1.73	-2.05
35	1213092	361441549	361441551	21:29:45.766	3918	799.75	8.92	34.25	1.62	-1.94	2.81	-2.44	0.61	-1.34
36	1410112	361441551	361441552	21:29:50.947	3790	787.5	10.59	13.75	0.79	-1.32	1.83	-2.07	0.32	-1.28
37	1410114	361441552	361441554	21:29:56.026	1248	1063.5	25.49	4	2.52	-3.53	3.23	-4.57	1.8	-2.48
38	1781804	361441554	361444098	21:29:58.601	1300	2045.5	28.43	17	2.73	-3.6	2.67	-3.47	2.79	-3.73
39	982484	361444098	792071790	21:30:01.151	6431	1903.5	33.14	32	2.15	-2.88	3.21	-3.84	0	-2.17
40	982486	792071790	579794578	21:30:08.869	11452	2007.3	34.08	53.6	2.16	-2.98	3.18	-5.04	2.48	-2.87
41	982488	579794578	361441560	21:30:21.572	6389	1741	41.37	57.17	1.62	-2.99	2.3	-4.09	1.01	-2.25

Διάγραμμα 11: Πίνακας Βάσης Δεδομένων

### 5.3.3 Απεικόνιση στο χάρτη – geojson.io

Μέσω του προγράμματος slimjson\_to\_csv\_unified.py, όπως προαναφέραμε, έχουμε ένα ενοποιημένο αρχείο σε μορφή csv το οποίο για κάθε σημείο της αρχικής καταγραφής που έγινε με το Raspberry Pi εμπεριέχει τη διορθωμένη και ταυτοποιημένη με το χάρτη συντεταγμένη καθώς και τα δεδομένα οχήματος που είχαν αρχικώς καταγραφεί.

Με drag-n-drop ένα τέτοιο αρχείο παρέχεται στη σελίδα geojson.io και προκύπτει το ακόλουθο αποτέλεσμα. Σημειώνουμε ότι πέρα από τη διαδρομή, διαφαίνονται και για κάθε σημείο τα καταγραφθέντα δεδομένα.



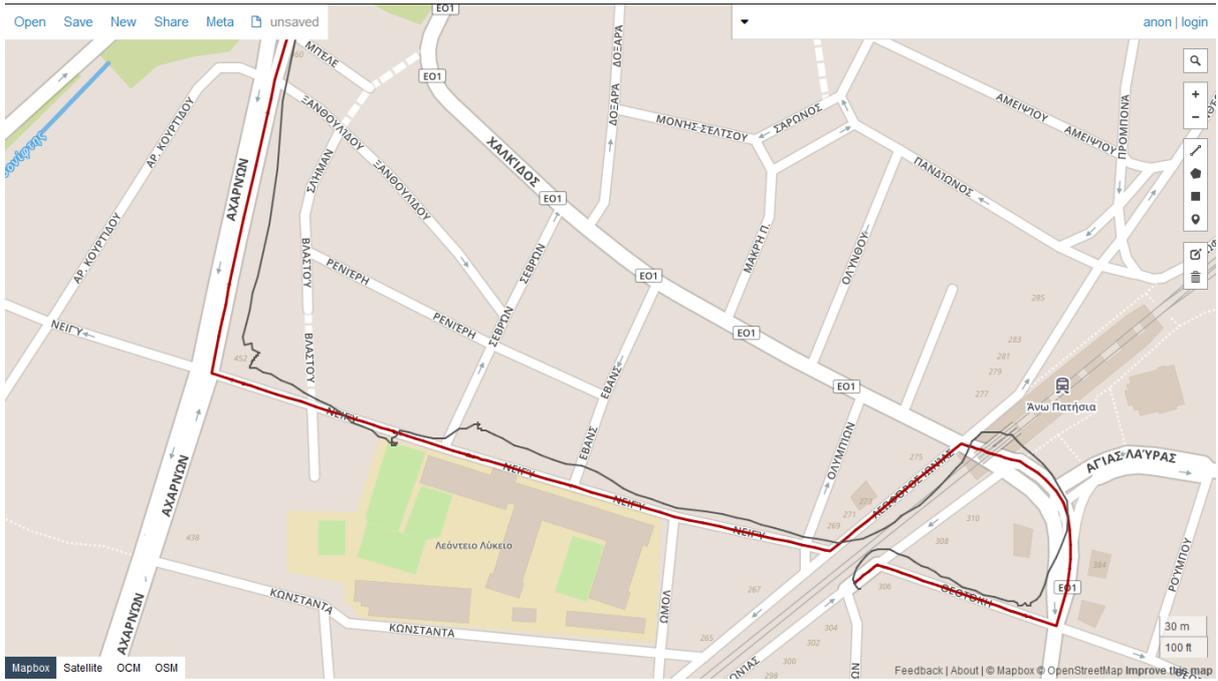
**Διάγραμμα 12: Απεικόνιση καταγραφής στο χάρτη με geojson.io**

Στη παρούσα διπλωματική μετατρέπουμε τα ενοποιημένα δεδομένα από csv σε (geo)json και τα απεικονίζουμε αυτοματοποιημένα στο χάρτη μέσω του geojsonio.py. Το πρόγραμμα αυτό ανοίγει αυτόματα στον περιηγητή την ιστοσελίδα geojson.io και απεικονίζει τα δεδομένα που του αποστέλλουμε στο χάρτη.

Συνοψίζοντας:

- Τα raw data καταφθάνουν στο Server
- Μετατρέπονται από csv μορφή σε json
- Εισάγονται στο Barefoot map-matching Server
- Εξάγονται τα snapped-to-map data σε μορφή json
- Γίνεται επεξεργασία των raw και snapped-to-map data και εξάγονται στατιστικά στοιχεία ανά οδικό σύνδεσμο τα οποία αποθηκεύονται σε βάση PostgreSQL
- Παράλληλα raw και snapped-to-map data απεικονίζονται στο χάρτη.

Ενδεικτική έξοδος της ανωτέρω διαδικασίας απεικονίζεται στο χάρτη στην επόμενη σελίδα: Με κόκκινο σημειώνεται η snapped-to-map διαδρομή και με μαύρο οι GPS καταγραφές.



**Εικόνα 15: Παράδειγμα Map-Matching**

## 6. Σενάρια Χρήσης - Μελλοντικές Επεκτάσεις



Εικόνα 16: Pi on the car

## 6.1 Σενάρια Χρήσης

Σε αυτό το κεφάλαιο αναφερόμαστε στα συστήματα Τηλεμετρίας στα οποία θα μπορούσε να βρει απήχηση η παρούσα διπλωματική.

### 6.1.1 Διαχείριση Στόλου – Fleet Management

Οι υπηρεσίες διαχείρισης στόλου οχημάτων δίνουν τη δυνατότητα στους “χρήστες” τους με τη χρήση ειδικής τεχνολογίας να ενημερώνονται σε πραγματικό χρόνο για τη γεωγραφική θέση του εκάστοτε οχήματος “φέροντος” την συσκευή γεωεντοπισμού.

Ο απαραίτητος εξοπλισμός για την παροχή των υπηρεσιών διαχείρισης στόλου συνήθως συνίσταται αφενός σε μία συσκευή διαχείρισης στόλου, που περιλαμβάνει κάρτα SIM για τη σύνδεση με δίκτυο κινητής τηλεφωνίας ή δυνατότητα σύνδεσης στο διαδίκτυο μέσω Wi-Fi, σύστημα γεωγραφικού προσδιορισμού θέσης (GPS), και αφετέρου σε ένα λογισμικό σύστημα, που εγκαθίσταται στον διακομιστή (server) του παρόχου της υπηρεσίας και καταγράφει το γεωγραφικό στίγμα των “φερόντων” οχημάτων σε πραγματικό χρόνο. Εκτός από την παροχή του στίγματος του “φέροντος” οχήματος, τέτοιες υπηρεσίες μπορεί να προσφέρουν και πρόσθετα χαρακτηριστικά, όπως ενδεικτικά :

- Κατάρτιση αναλυτικών στατιστικών ως προς την θέση και κίνηση του “φέροντος” οχήματος (λ.χ. κατανάλωση καυσίμων, ταχύτητα και μοτίβα κίνησης / συμπεριφοράς).
- Δυνατότητα ειδοποίησης του “χρήστη”, όταν συντρέχουν συγκεκριμένες συνθήκες (λ.χ. με την έξοδο από ορισμένη γεωγραφική ζώνη ή δρομολόγιο ή την υπέρβαση ταχύτητας).

Απευθύνονται κυρίως σε επιχειρήσεις ανεξάρτητα από το μέγεθός τους και παρέχουν σημαντικά οφέλη όπως:

- Μείωση του κόστους λειτουργίας του εταιρικού στόλου όπως έξοδα επικοινωνίας, συντήρησης, υπερωριών, κατανάλωσης καυσίμων
- Βελτίωση του επιπέδου εξυπηρέτησης των πελατών
- Καλύτερη οργάνωση καθημερινών δρομολογίων και αποφάσεων διαχείρισης
- Προστασία του μεταφερόμενου φορτίου από πιθανές αλλοιώσεις

- Αυτοματοποίηση και επιτάχυνση διαδικασιών παραγγελιοληψίας και παράδοσης
- Καλύτερη εκμετάλλευση των οχημάτων και αύξηση παραγωγικότητας
- Αποτελεσματική αντιμετώπιση εκτάκτων καταστάσεων
- Διαφοροποίηση από τον ανταγωνισμό.

### **6.1.2 Usage-Based Insurance**

Παράλληλα, υπάρχει πιθανότητα η παρούσα διπλωματική εργασία να φανεί χρήσιμο εργαλείο και στον τομέα του usage-based insurance, υπό το πρίσμα της εξάρτησης του κόστους οδήγησης με τον μηχανικό τύπο του αυτοκινήτου, και έπειτα από μέτρησεις σχετικές με τον χρόνο, την απόσταση, τη συμπεριφορά και την τοποθεσία. Αυτό το είδος ασφάλισης, διαφέρει από τον παραδοσιακό τύπο της "ασφαλούς" οδήγησης και σχετίζεται με την έννοια του κόστους ανάλογα με τον τρόπο οδήγησης. Για αυτό το λόγο το κόστος ασφάλισης εξαρτάται σε μεγάλο βαθμό όχι μόνο από το πόσο πολύ οδηγεί κάποιος, αλλά κιόλας από τον τρόπο, το μέρος και την ώρα οδήγησης.

Υπάρχουν τρία είδη για την εν λόγω ασφάλιση, η κάλυψη της οποίας βασίζεται είτε στον οδομετρητή του αυτοκινήτου, στη μέτρηση των χιλιομέτρων από τα δεδομένα του GPS, είτε σε άλλα δεδομένα όπως η επικινδυνότητα της διαδρομής, ο χρόνος ταξιδιού, η ταχύτητα κ.ο.κ.

Η usage-based insurance παρουσιάζει τόσο πλεονεκτήματα όσο και μειονεκτήματα. Στα οφέλη μπορούμε να επικεντρωθούμε στις θετικές κοινωνικές και περιβαλλοντικές συνέπειες ως απόρροια υπεύθυνης οδήγησης, στο περιορισμό του κόστους για τους καταναλωτές και στην ποικιλία επιλογών για την αγορά πακέτου ασφάλισης, εγκαθίδρυση μιας κουλτούρας περισσότερης ασφάλειας στους δρόμους μια και η επικίνδυνη οδήγηση ταυτίζεται με υψηλό κόστος. Οι πιθανές αρνητικές επιπτώσεις, όμως, περιλαμβάνουν την μη ακρίβεια στην πρόβλεψη του ρίσκου της οδήγησης από μέρους ενός, καθώς και αντίκτυπο στον τομέα της ιδιωτικότητας και των προσωπικών δεδομένων.

### **6.1.3 Συστήματα Τηλεματικής**

Πεδίο εφαρμογής θα μπορούσε να είναι ακόμα σε συστήματα Τηλεματικής, όπως το νέο σύστημα για τις αστικές συγκοινωνίες του ΟΑΣΑ (<http://telematics.oasa.gr>).

Οι επιβάτες πληροφορούνται τον ακριβή χρόνο διέλευσης των οχημάτων, από τις οθόνες που θα είναι τοποθετημένες στις στάσεις, καθώς επίσης και από το κινητό τους τηλέφωνο και τον ηλεκτρονικό υπολογιστή τους.

Από κέντρα ελέγχου παρακολουθείται σε πραγματικό χρόνο η λειτουργική κατάσταση του συστήματος συγκοινωνιών και γίνονται οι αναγκαίες παρεμβάσεις για τον καλύτερο συντονισμό και βελτιστοποίηση του συγκοινωνιακού έργου.

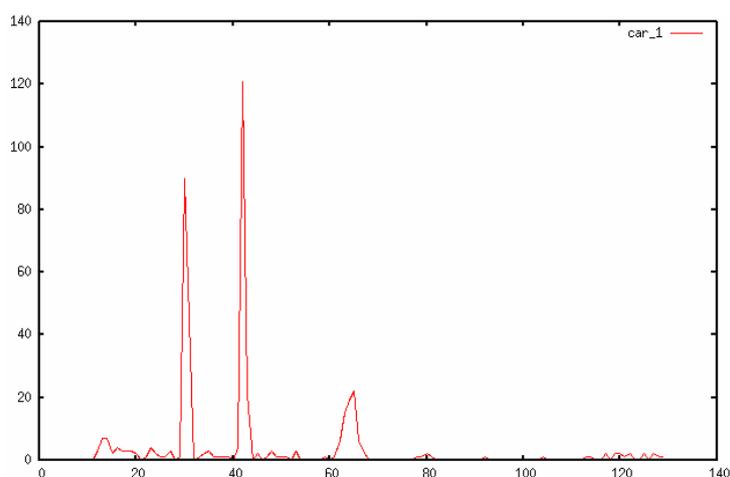
Το εν λόγω σύστημα συνδυάζει τις δυνατότητες που προσφέρουν οι Τεχνολογίες της Πληροφορικής και των Τηλεπικοινωνιών για την υλοποίηση ενός ολοκληρωμένου συστήματος πληροφόρησης επιβατών και διαχείρισης συγκοινωνιακού στόλου. Πρόκειται για το πρώτο βήμα εφαρμογής ενός ολοκληρωμένου σχεδίου για τη δημιουργία ενός πρότυπου συστήματος αστικών συγκοινωνιών, που θα συμβάλει στη βελτίωση της καθημερινότητας του επιβατικού κοινού και στην αναβάθμιση του επιπέδου συντονισμού και παραγωγικότητας του συγκοινωνιακού έργου.

## 6.2 Μελλοντικές Επεκτάσεις

Η εφαρμογή που υλοποιήσαμε στα πλαίσια της παρούσας διπλωματικής είναι εύκολα επεκτάσιμη και κλιμακώσιμη. Στο παρόν κεφάλαιο αναφερόμαστε στο πώς μπορεί να προεκταθεί.

### 6.2.1 Αναγνώριση σχέσης στο κιβώτιο ταχυτήτων

Αποδεικνύεται εύκολα ότι η σχέση που επιλέγουμε στο κιβώτιο ταχυτήτων είναι συνάρτηση του λόγου στροφών κινητήρα προς την ταχύτητα του αυτοκινήτου. Είναι δηλαδή:  $\lambda = f\left(\frac{rpm}{v}\right)$ , όπου  $\lambda$  η σχέση που επιλέγουμε στο κιβώτιο ταχυτήτων (1<sup>η</sup>, 2<sup>α</sup>, 3<sup>η</sup>, 4<sup>η</sup> και 5<sup>η</sup> – η όπισθεν και η 6<sup>η</sup> που υπάρχει σε κάποια αυτοκίνητα δεν θα μας απασχολήσει),  $rpm$  οι τρέχουσες στροφές του κινητήρα και  $v$  η τρέχουσα ταχύτητα του αυτοκινήτου. Σημειώνοντας επομένως τον λόγο  $\frac{rpm}{v}$  στις μετρήσεις μας κατά τη διάρκεια μιας διαδρομής μπορούμε να προσδιορίσουμε τη σχέση που έχει επιλέξει ο οδηγός στο κιβώτιο ταχυτήτων. Θα επιδείξουμε τον παραπάνω ισχυρισμό με διαγραμματικό τρόπο. Το παρακάτω διάγραμμα αφορά διαδρομή κατά την οποία το αυτοκίνητο κινήθηκε τόσο σε αρτηρία ταχείας κυκλοφορίας όσο και σε αστική περιοχή και έχει στον οριζόντιο άξονα το λόγο  $\frac{rpm}{v}$  και στο κάθετο το πλήθος των επαναλήψεων που εμφανίστηκε ο συγκεκριμένος λόγος στις μετρήσεις μας.



Διάγραμμα 13: Διάγραμμα για τον εντοπισμό της σχέσης στο κιβώτιο ταχυτήτων

Στο παραπάνω διάγραμμα παρατηρούμε κάποιες κορυφές (τοπικά μέγιστα). Οι αιχμηρότερες κορυφές αφορούν λόγους  $\frac{rpm}{v}$  που αποτελούν ξεχωριστή σχέση στο κιβώτιο ταχυτήτων.

Επομένως, το πρόβλημα εντοπισμού της σχέσης στο κιβώτιο ταχυτήτων ανάγεται σε πρόβλημα εντοπισμού κορυφών (peak-detection) σε ένα τέτοιο διάγραμμα.

Γνωρίζοντας τη φύση της διαδρομής μπορούμε να συμπεράνουμε ότι η κορυφή που βρίσκεται στο λόγο  $\frac{rpm}{v} = 80$  αφορά την 1<sup>η</sup> σχέση στο κιβώτιο, η κορυφή που βρίσκεται λίγο πάνω από το λόγο 60 αφορά την 2<sup>η</sup> σχέση στο κιβώτιο, η κορυφή που βρίσκεται κοντά στο 40 αντιστοιχίζεται στην 3<sup>η</sup>, ο λόγος 30 αντιστοιχίζεται στην 4<sup>η</sup> και λόγος λίγο κάτω από 20 αντιστοιχίζεται στην 5<sup>η</sup> ταχύτητα. Μερικές σημαντικές παρατηρήσεις:

- Οι κορυφές που εμφανίζονται κοντά στο λόγο 120 οφείλονται στη φύση της 1<sup>ης</sup> ταχύτητας.
- Οι ενδιάμεσοι λόγοι αφορούν στάδια κατά τα οποία γίνονται εναλλαγές στη σχέση στο κιβώτιο και σε σφάλματα των μετρήσεων.
- Η 3<sup>η</sup> σχέση είναι αυτή που χρησιμοποιήθηκε ως επί το πλείστον στη διαδρομή και ακολουθεί η 4<sup>η</sup> κ.ο.κ.

Η αναγνώριση της σχέσης στο κιβώτιο ταχυτήτων μπορεί να χρησιμοποιηθεί ως μια παράμετρος για το δημιουργία προφίλ οδηγικής συμπεριφοράς για τον οδηγό (driver profiling).

### 6.2.2 Προφίλ οδηγικής συμπεριφοράς (Driver profiling)

Αντικείμενο πολλών ερευνητών είναι η εξαγωγή δεικτών που υπολογίζουν τον αντίκτυπο της οδηγικής συμπεριφοράς στην οδική ασφάλεια και στο περιβάλλον.

Τα αποτελέσματα και οι μετρήσεις που καταγράφηκαν στα πλαίσια της παρούσας διπλωματικής, θα μπορούσαν να χρησιμοποιηθούν προς τον σκοπό αυτό. Οι μέσες, μέγιστες και ελάχιστες επιταχύνσεις ανά οδικό σύνδεσμο είναι παράγοντες οι οποίοι προσδιορίζουν το πόσο αμυντικά ή επιθετικά κινείται ο οδηγός στον συγκεκριμένο σύνδεσμο. Η μέση τιμή της ταχύτητας, των στροφών κινητήρα καθώς και η σχέση στο κιβώτιο ταχυτήτων που επιλέγεται από τον οδηγό μπορεί να καθορίσει κατά πόσο είναι οικονομική η οδήγηση του. Ο υπολογισμός του φορτίου που αποδίδει ο κινητήρας αναδεικνύει αν η οδήγηση είναι ενεργειακά αποδοτική. Βάσει των παραπάνω παραμέτρων δίνεται η δυνατότητα να προκύψει μία βαθμολόγηση (driver score) για τον οδηγό η οποία θα αναλύει το οδηγικό στυλ του. Με

αυτόν τον τρόπο επιτυγχάνεται μείωση της κατανάλωσης καυσίμου καθώς και αύξηση του κινήτρου του οδηγού για ασφαλέστερη και οικονομικότερη οδήγηση.

### 6.2.3 Online (Real-time) system

Στην παρούσα διπλωματική υλοποιούμε Offline map matching μέσω του προγράμματος Barefoot. Το Barefoot με κάποιες παραμετροποιήσεις δίδει τη δυνατότητα στο χρήστη για online real-time map matching. Δηλαδή, σε πραγματικό χρόνο η θέση του αυτοκινήτου αντιστοιχίζεται επακριβώς στο χάρτη. Έτσι, αν στο Raspberry Pi συνδέσουμε ένα δέκτη (οθόνη) μπορούμε να υλοποιήσουμε σύστημα που να απεικονίζει σε κάθε χρονική στιγμή την ακριβή θέση του αυτοκινήτου στον OpenStreetMap χάρτη. Με μερικές ακόμα παραμετροποιήσεις στα προγράμματα μας μπορούμε να εξάγουμε και μετρήσεις πραγματικού χρόνου, δίδοντας τη δυνατότητα στο χρήστη να βλέπει στη οθόνη διαγράμματα για καταναλώσεις, επιταχύνσεις κτλ κατά τη διάρκεια της κίνησης του. Τέλος, χρησιμοποιώντας για παράδειγμα τον αλγόριθμο Routing του Graphhopper έχουμε τη δυνατότητα να υλοποιήσουμε ένα σύγχρονο υπολογιστή ταξιδιού κάνοντας χρήση μόνο open-source εργαλείων.

### 6.2.4 Λοιπές μελλοντικές προεκτάσεις

Πέρα από τα παραπάνω, τα εξαγόμενα αποτελέσματα της παρούσας διπλωματικής μπορούν να χρησιμοποιηθούν με πολλούς ακόμα τρόπους. Για παράδειγμα:

- Μπορεί να ανιχνευθεί αν έχει υπάρξει φθορά στον συμπλέκτη του αυτοκινήτου (clutch wear detection) ή αν το ελαστικό είναι ξεφούσκωτο (flat tire detection) μέσω παρατήρησης της μεταβολής του λόγου  $\frac{rpm}{v}$ .
- Μπορούμε ακόμα να εντοπίσουμε μέσω του επιταχυνσιόμετρου την κλίση του αυτοκινήτου, προσδιορίζοντας την κάθετη συνιστώσα της επιτάχυνσης στο επίπεδο της κίνησης.
- Μπορεί να λειτουργήσει ως συσκευή για distributed sensing βάσει αισθητήρων που υπάρχουν στο όχημα (πχ καιρός). Ένα συμβατό εξάρτημα με το Raspberry είναι το SHT21 Temperature and Humidity Sensor, το οποίο εξάγει αποτελέσματα για θερμοκρασία και υγρασία κάνοντας χρήση των βιβλιοθηκών της Microstack.[28]

## Παράρτημα

Στο παράρτημα παραθέτουμε τους απαραίτητους κώδικες για να γίνει η καταγραφή στο Raspberry Pi. Οι υπόλοιποι απαραίτητοι κώδικες για να υλοποιηθεί η εφαρμογή τόσο στον client όσο και στον Server βρίσκονται στο CD.

Το πρόγραμμα `obd1.py` είναι υπεύθυνο να καταγράψει μετρήσεις από τη θύρα OBD του αυτοκινήτου στο αρχείο `FILE` καθώς και να εκκινήσει το πρόγραμμα `gps_acc.py`

```
import obd
import time
import subprocess
import signal
import sys
import os
import datetime

#obd.debug.console = True
FILE = sys.argv[1]
p_pid = os.getpid()
ports = obd.scanSerial()
connection = obd.OBD(ports[0])
proc = subprocess.Popen(["python3", "gps_acc.py", FILE])
c_pid = proc.pid
os.waitpid(c_pid, os.WUNTRACED)

while 1:
    date_time = datetime.datetime.now().strftime('%Y-%m-%d,%H:%M:%S.%f')[:-3]
    cel = connection.query(obd.commands[1][4])
    rpm = connection.query(obd.commands[1][12]) #print (rpm.value)
    speed = connection.query(obd.commands[1][13])
    #acc_ped_pos = connection.query(obd.commands[1][90])
    #eng_fuel_rate = connection.query(obd.commands[1][94])
    f = open(FILE, 'a')
    f.write(date_time + ',')
    f.write(str(cel.value)+',')
    f.write(str(rpm.value)+',')
```

```

f.write(str(speed.value)+',')
f.write(str(acc_ped_pos.value)+',')
f.write(str(eng_fuel_rate.value)+',')
f.close()
os.kill(c_pid, signal.SIGCONT)
os.waitpid(c_pid, os.WUNTRACED)

```

Το πρόγραμμα `gps_acc.py` εκκινείται από το `obd.py` και καταγράφει μετρήσεις θέσης και επιτάχυνσης συγχρονισμένα με το `obd.py`.

```

import time
import datetime
import microstacknode.hardware.gps.l80gps
from microstacknode.hardware.accelerometer.mma8452q import MMA8452Q
import signal
import os
import sys

def gps_acc():
    gps = microstacknode.hardware.gps.l80gps.L80GPS()
    with MMA8452Q() as accelerometer:
        accelerometer.standby()
        accelerometer.set_g_range(2)
        accelerometer.activate()
        time.sleep(2)
        while 1:
            gpgga = gps.get_gpgga()
            gprmc = gps.get_gprmc()
            ms = accelerometer.get_xyz_ms2()
            os.kill(my_pid, signal.SIGSTOP) #signal stop
            to_file(gpgga, gprmc, ms)
            time.sleep(1)

def to_file(gpgga, gprmc, ms):
    km_h = float(gprmc['speed'])*1.852 #1knot=1.852km/h
    hdop = float(gpgga['hdop'])*2.5 #accuracy = 2.5m
    f = open(FILE, 'a')
    f.write('{:.5f},'.format(gpgga['latitude']))
    f.write('{:.5f},'.format(gpgga['longitude']))

```

```

f.write('{}', '.format(gpgga['altitude']))
f.write(' {:.2f}', '.format(hdop))
f.write(' {:.1f}', '.format(km_h))
f.write(' {:.2f}', '.format(ms['x']))
f.write(' {:.2f}', '.format(ms['y']))
f.write(' {:.2f}\n'.format(ms['z']))
f.close()

if __name__ == '__main__':
    my_pid = os.getpid()
    p_pid = os.getppid()
    print(p_pid)
    print(my_pid)
    FILE = sys.argv[1]
    f = open(FILE, 'x')
    f.write ("date_time, cel, rpm, speed, acc_ped_pos, eng_fuel_rate, lat,
long, alt, hdop, gps_speed, a_x, a_y, a_z\n")
    f.close()
    gps_acc()

```

## Βιβλιογραφία/ Παραπομπές

- [1] Ασύρματες Επικοινωνίες και Δίκτυα, Stallings William, Εκδ. Τζιόλα (ISBN: 9604181238)
- [2] Δίκτυα υπολογιστών, Tanenbaum Andrew S., Εκδ. Κλειδάριθμος (ISBN: 960-461-447-9)
- [3] Δικτύωση Υπολογιστών: Προσέγγιση από Πάνω προς τα Κάτω, James F. Kurose, Keith W. Ross, Εκδ. Μ. Γκιούρδας (ISBN: 978-960-512-657-5)
- [4] Οι Υπολογιστές ως Συστατικά Στοιχεία – Αρχές Σχεδίασης Ενσωματωμένων Υπολογιστικών Συστημάτων, Wolf W., Εκδ. Νέων Τεχνολογιών (ISBN: 978-960-6759-18-5)
- [5] Συστήματα Βάσεων Δεδομένων: Η Πλήρης Θεωρία των Βάσεων Δεδομένων, Abraham Silberschatz, Henry F. Korth, S. Sudarshan, Εκδ. Μ. Γκιούρδας (ISBN: 978-960-512-623-0)
- [6] Λειτουργικά Συστήματα, Abraham Silberschatz, Peter Baer Galvin, Greg Gagne, Εκδ. ΙΩΝ (ISBN: 978-960-411-692-8)
- [7] Telemetry Systems Engineering, Frank Carden, Robert Henry, Russell Jedlicka (ISBN: 1580532578)
- [8] Sebastian Mattheis, Kazi Khaled Al-Zahid, Birgit Engelmann, Andreas Hildisch, Stefan Holder, Olexiy Lazarevych, Daniel Mohr, Felix Sedlmeier, Richard Zinck, "Putting the car on the map: A scalable map matching system for the Open Source Community", GI-Jahrestagung (2014).
- [9] Francisco Camara Pereira, Hugo Costa, Nuno Martinho Pereira, "An off-line map-matching algorithm for incomplete map databases", European Transport Research Review (2009).

- [10] E. Koukis, P. Louridas, "~oceanos Iaas", EGI Community Forume (2012)
- [11] Miyajima, Chiyomi, Hiroki Ukai, Atsumi Naito, Hideomi Amata, Norihide Kitaoka and Kazuya Takeda, "Driver risk evaluation based on acceleration, deceleration, and steering behavior.", ICASSP (2011).
- [12] C. Ball, "LTE and WiMax Technology and Performance Comparison", Nokia Siemens Networks (2007).
- [13] Real-time Hidden markov model based map-matching,  
<http://ds.cs.ut.ee/courses/course-files/Kibal-seminar-report.pdf>
- [14] ELM327 OBD adapter, <http://elmelectronics.com/DSheets/ELM327DS.pdf>
- [15] Microstack Accelerometer,  
<http://www.microstack.org.uk/assets/accel/FormattedAccelgettingstarted.pdf>
- [16] Microstack GPS, <http://www.farnell.com/datasheets/1860443.pdf?COM=July>
- [17] Documentation για την PostgreSQL, <https://www.postgresql.org/docs/>
- [18] Documentation για την γλώσσα Προγραμματισμού Python, <https://docs.python.org>
- [19] Η ιστοθεσία του gejson.io στο Github, <https://github.com/mapbox/gejson.io>
- [20] Η ιστοθεσία του Barefoot στο Github, <https://github.com/bmwcarit/barefoot>
- [21] Gejsonio.py, <https://github.com/jwass/gejsonio.py>
- [22] Graphhopper, <https://github.com/graphhopper>
- [23] Google Roads API – Snap to Roads  
<https://developers.google.com/maps/documentation/roads/snap>

- [24] Η κεντρική ιστοσελίδα για το Raspberry Pi, <https://www.raspberrypi.org>
  
- [25] Η κεντρική ιστοθεσία του Raspbian, <https://www.raspbian.org/>
  
- [26] Βιβλιοθήκες της Microstack για GPS και επιταχυνσιόμετρο γραμμένες σε Python 3, <http://python3-microstacknode.readthedocs.io/>
  
- [27] Βιβλιοθήκη Python-OBd, <https://github.com/brendan-w/python-OBd>
  
- [28] Documentation για τον αισθητήρα θερμοκρασίας και υγρασίας της Microstack, <http://python3-microstacknode.readthedocs.io/en/latest/example.html#sht21-temperature-and-humidity-sensor>
  
- [29] OBd-II, <http://en.wikipedia.org/wiki/Obd-ii>
  
- [30] On-board diagnostics, [http://en.wikipedia.org/wiki/On-board\\_diagnostics](http://en.wikipedia.org/wiki/On-board_diagnostics)
  
- [31] Docker, <https://el.wikipedia.org/wiki/Docker>
  
- [32] OSM χάρτες, <http://www.openstreetmap.org>