



Πανεπιστήμιο Πειραιώς – Τμήμα Πληροφορικής

Πρόγραμμα Μεταπτυχιακών Σπουδών

«Προηγμένα Συστήματα Πληροφορικής»

Μεταπτυχιακή Διατριβή

Τίτλος Διατριβής	Ανάπτυξη εφαρμογής συλλογής δεδομένων από οχήματα μέσω θύρας OBD2, για λειτουργικό Android Android application development for vehicle data collection via OBD2 interface.
Όνοματεπώνυμο Φοιτητή	Αντώνιος Μελιγδής
Πατρώνυμο	Απόστολος
Αριθμός Μητρώου	ΜΠΣΠ13065
Επιβλέπων	Κωνσταντίνος Πατσάκης, Επίκουρος Καθηγητής

Ημερομηνία Παράδοσης **30 Οκτωβρίου 2017**

Τριμελής Εξεταστική Επιτροπή

(υπογραφή)

(υπογραφή)

(υπογραφή)

Κωνσταντίνος Πατσάκης
Επίκουρος Καθηγητής

Ευθύμιος Αλέπης
Επίκουρος Καθηγητής

Γεώργιος Τσιχριντζής
Πρόεδρος Τμήματος
Πληροφορικής
Πανεπιστημίου
Πειραιώς

Περιεχόμενα

Περίληψη	6
Εισαγωγή	8
ΚΕΦΑΛΑΙΟ 1	9
1.1 Η θύρα OBD2	9
1.2 Τα πρωτόκολλα σηματοδότησης της OBD2	12
1.2.1 SAE J1850	12
1.2.2 SAE J1939	12
1.2.3 Raw CAN (ISO 11898).....	14
1.2.4 CAN Bus (ISO 15765).....	14
1.2.5 Keyword Protocol 2000	15
1.2.6 ISO 9141-2.....	15
1.3 OBD2 Modes	16
1.3.1 Diagnostic Trouble Codes	17
1.3.2 Electronic Control Unit (ECU)	19
1.3.3 Θέματα ασφαλείας.....	21
ΚΕΦΑΛΑΙΟ 2	24
2.1 Υλοποίηση	24
2.1.1 Ξεκινώντας.....	24
2.1.2 ODB Simulator	26
2.1.3 Βασικές εντολές επικοινωνίας.	28
2.2 Κύρια σημεία πηγαίου κώδικα	29
2.2.1 Αρχιτεκτονική εφαρμογής.....	29
2.2.2 Υπολογισμός κατανάλωσης καυσίμου	31
2.3 Επεξήγηση λειτουργίας του VehicleLogger	34
2.3.1 Αρχικοποίηση σύνδεσης.....	34
2.3.2 Πλοήγηση στο GUI.....	39
2.4 Αποθήκευση σε βάση δεδομένων	41
ΚΕΦΑΛΑΙΟ 3	46
3.1 Προβλήματα Υλοποίησης	46
3.2 Επισκόπηση και μελλοντική έρευνα.	48
Παράρτημα I - Ορολογία	50
Βιβλιογραφία:	53

Περίληψη

Η παρούσα διπλωματική εργασία εκπονήθηκε στα πλαίσια του μεταπτυχιακού προγράμματος σπουδών «Προηγμένα Συστήματα Πληροφορικής» του Πανεπιστημίου Πειραιώς. Αποτελεί μία εναλλακτική λύση στις υπάρχουσες εμπορικές εφαρμογές για την ανάγνωση και λήψη δεδομένων σε πραγματικό χρόνο μέσω της θύρας OBD2 που φέρουν όλα τα σύγχρονα οχήματα.

Η εργασία έχει σκοπό την δημιουργία μίας εφαρμογής λογισμικού που θα εκτελείται σε περιβάλλον Android κινητών τηλεφώνων ή tablet. Με την βοήθεια ενός ELM dongle του εμπορίου το οποίο θα συνδέεται στην θύρα επικοινωνίας OBD2 του εγκεφάλου του ΙΧ, θα υπάρχει διεπαφή της συσκευής με την εφαρμογή μέσω του Bluetooth. Ο χρήστης, αφού εγκαταστήσει την εφαρμογή στο κινητό του, θα μπορεί να δει σε πραγματικό χρόνο τα διάφορα δεδομένα τηλεμετρίας που επιστρέφει ο εγκέφαλος του οχήματος στην οθόνη του κινητού του καθώς και τυχόν βλάβες που είναι ενεργοποιημένες στην μνήμη του εγκεφάλου. Παράλληλα, τα δεδομένα αυτά θα αποστέλλονται με upload σε μία real-time απομακρυσμένη βάση δεδομένων (Firebase) και θα μπορούν να αναγνωστούν σε πραγματικό χρόνο. Η εφαρμογή θα καταγράφει κάθε διαδρομή που θα εκτελείται, με σκοπό να συγκεντρώσει δεδομένα τόσο από την τηλεμετρία του αυτοκινήτου όσο και από τα αισθητήρια της εκάστοτε φορητής συσκευής.

Μελλοντικός σκοπός είναι η συγκέντρωση αυτών των δεδομένων σε μεγάλη κλίμακα από πολλούς και διαφορετικούς χρήστες και ΙΧ ώστε η βάση δεδομένων που θα δημιουργηθεί να δώσει την δυνατότητα να γίνουν περαιτέρω μελέτες στην συμπεριφορά των μοντέλων της εκάστοτε αυτοκινητοβιομηχανίας και όχι μόνο.

Abstract

The following master thesis was developed for the purposes of the “Advanced Informatics and Computing Systems” program for the University of Piraeus. It proposes an alternative solution to the already available commercial applications which use the OBD2 port of all vehicles and transmit live data to a user’s mobile device.

The software application presented below was developed for mobile phones or tablets that run Android OS. Assisted by a commercial OBD2 dongle that was purchased, the user’s mobile device communicates with the dongle (which is connected to the car’s interface) via Bluetooth. The user is able to have a large series of live data feed of the car’s telemetry system as well as possible error codes that show if there is hardware malfunction. The user will record each trip that he makes in order to upload the raw data from the car’s and mobile device’s sensors, to build a remote real-time database (Firebase).

The future aim of the application designed is to gather large amounts of raw data from different users and as a result from different vehicles. That will allow future researchers to conduct a series of new studies based on the raw data that are gathered, about the behavior of each vehicle make, model and many more.

Εισαγωγή

Η τεχνολογική εξέλιξη των τελευταίων 20 ετών στον τομέα της αυτοκίνησης αλλά και η ταυτόχρονη εξέλιξη στον τομέα της πληροφορικής έχουν επιτρέψει στον απλό, καθημερινό άνθρωπο να έχει πρόσβαση σε έναν όγκο πληροφοριών που άλλοτε θα θεωρείτο εξωπραγματικός. Η υποχρεωτική εγκατάσταση της διεπαφής OBD2 το 2001 σε οποιοδήποτε όχημα εργοστασιακής παραγωγής για τα κράτη μέλη της Ευρωπαϊκής Ένωσης έχει φέρει μία πραγματική επανάσταση.

Η πληροφορία που παράγει ο εγκέφαλος ενός αυτοκινήτου σχετικά με την διάγνωση βλαβών και την μελέτη της λειτουργίας και της απόδοσης ενός αυτοκινήτου σε πραγματικό χρόνο είναι πλέον προσβάσιμη όχι μόνο στα εξουσιοδοτημένα συνεργεία των εταιριών και στους μηχανικούς βελτίωσης των εργοστασίων, αλλά και στον ιδιοκτήτη ενός οχήματος. Η ταυτόχρονη ανάπτυξη των κινητών τηλεφώνων και της πληροφορικής, η οποία έχει φέρει σήμερα την δυνατότητα να έχει ο καθένας έναν υπερ-υπολογιστή στην τσέπη του, έχει δώσει μία τελείως διαφορετική οπτική στο πως μπορεί να γίνει διαχείριση ενός μέσου που χρησιμοποιείται καθημερινά από τους περισσότερους ανθρώπους. Η παρούσα διπλωματική εργασία εστιάζει στην ανάπτυξη μίας εφαρμογής λογισμικού η οποία θα δώσει διαφορετικές δυνατότητες στην συλλογή πληροφοριών από τις υπάρχουσες εμπορικές εφαρμογές.

Στο πρώτο κεφάλαιο θα γίνει αναφορά στην διεπαφή OBD2 και θα αναλυθούν τα πρωτόκολλα σηματοδότησης που χρησιμοποιεί η διεπαφή και οι καταστάσεις λειτουργίας της. Το δεύτερο κεφάλαιο είναι αφιερωμένο στην παρουσίαση του hardware που χρησιμοποιήθηκε, στην μεθοδολογία που ακολουθήθηκε για να αναπτυχθεί ο κώδικας της εφαρμογής καθώς επίσης και στην υλοποίηση. Στο τρίτο κεφάλαιο θα γίνει αναφορά στα προβλήματα που αντιμετωπίστηκαν και επίσης περιλαμβάνεται μία επισκόπηση της εργασίας παράλληλα με προτάσεις για την μελλοντική έρευνα που μπορεί να διεξαχθεί.

ΚΕΦΑΛΑΙΟ 1

1.1 Η θύρα OBD2

Ο όρος OBD2 αναφέρεται στο ακρωνύμιο της αγγλικής φράσης On-Board Diagnostics 2nd generation. Πρόκειται για ένα υποχρεωτικό σύστημα που εφαρμόζεται μέχρι και σήμερα στην αυτοκινητοβιομηχανία με σκοπό να παρακολουθείται το πρόβλημα με τις εκπομπές αερίων από τα οχήματα. Η ρίζες του πηγαινούν πίσω, πάνω από 30 χρόνια, συγκεκριμένα στο 1982, όπου στην Καλιφόρνια των Ηνωμένων Πολιτειών η California Air Resources Board (CARB) ήταν το πρώτο σύστημα που αναπτύχθηκε έτσι ώστε να μπορέσουν οι κατασκευαστές αυτοκινήτων να έχουν έναν κοινό τρόπο εντοπισμού των εκπομπών αερίων από το 1988 και μετά.

Η πρώτη έκδοση ήταν σχετικά απλή και περιελάμβανε μόνο παρακολούθηση των αισθητηρίων οξυγόνου, του συστήματος ανακύκλωσης καυσαερίων, του συστήματος παροχής καυσίμου και του module ελέγχου της μηχανής. Συνήθως ενεργοποιούσαν κάποιο ηλεκτρονικό κύκλωμα και άναβαν μία ενδεικτική λυχνία (check engine) γνωστό και ως MIL (Malfunction Indicator Lamp) στο καντράν του οδηγού όταν υπήρχε κάποιο πρόβλημα, όπως φαίνεται στην παρακάτω εικόνα.



Εικόνα 1: Λαμπτήρας check engine.

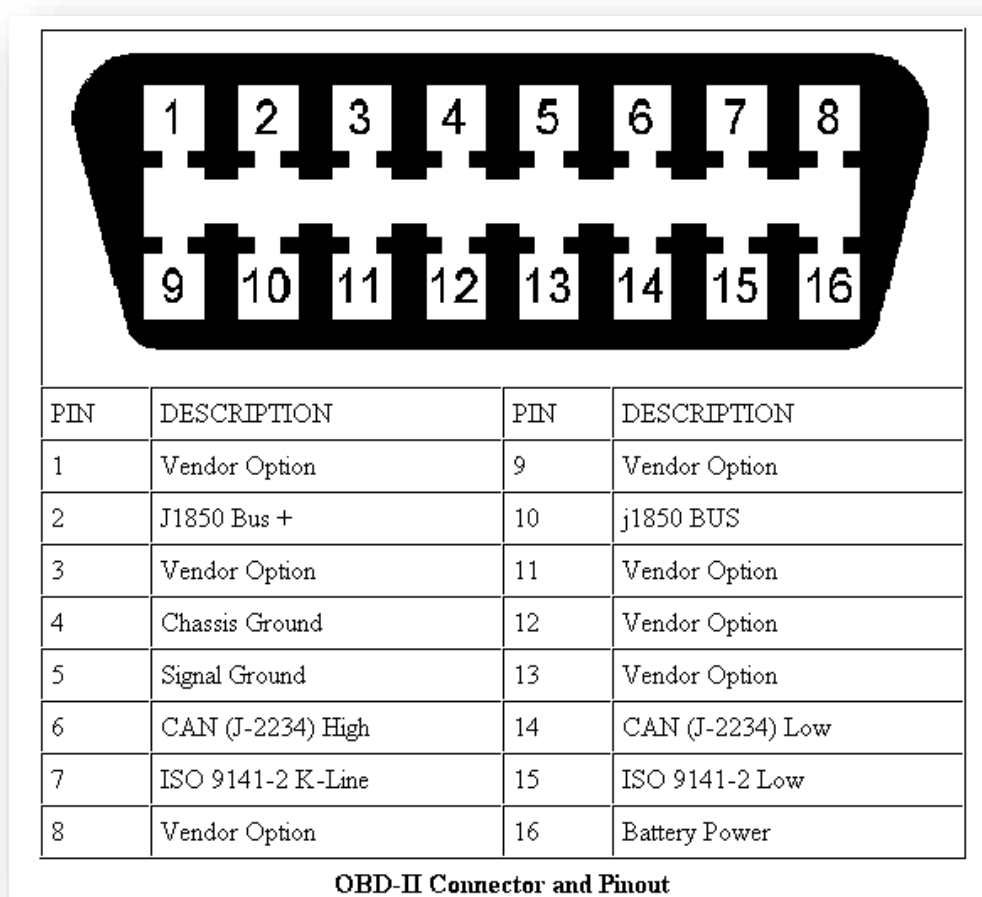
Αυτό όμως, στην ουσία, δεν παρείχε καμία επιπλέον πληροφορία ως προς την φύση του προβλήματος. Η πιθανότητα ενός ελαττωματικού καταλύτη ή η πρόβλεψη αφαίρεσης αυτού προτού συμβεί μία ανεπανόρθωτη και κοστοβόρα βλάβη δεν ήταν δυνατή. Παρόλο που η αρχική ιδέα ήταν σωστή, υπήρχε ο περιορισμός της μη τυποποίησης, με αποτέλεσμα να υπάρχουν σοβαρά προβλήματα ασυμβατότητας μεταξύ των συσκευών των διάφορων κατασκευαστών. Για παράδειγμα, θα έπρεπε να χρησιμοποιηθούν διαφορετικού τύπου αντάπτορες σε κάθε όχημα και αντιστοίχως διαφορετικά προγράμματα scanning σε κάθε μοντέλο. Έτσι, έγινε προφανές ότι μία πιο έξυπνη, καθολική και ολοκληρωμένη λύση ήταν αναγκαία.

Αυτό υλοποιήθηκε μετά το 1989 όταν η California Air Resources προχώρησε στην τυποποίηση των ISO που χρησιμοποιήθηκαν για την επόμενη γενιά των συστημάτων OBD. Η αρχή έγινε το 1994 όταν και δόθηκε στους κατασκευαστές μία διετής προθεσμία ώστε να ενσωματώσουν τα συστήματα τους στα προκαθορισμένα πρότυπα. Η ενσωμάτωση περιελάμβανε την εγκατάσταση ενός 16-pin connector σε σχήμα τραπεζίου, με κάποια από τα pin του να είναι σε προεπιλεγμένη θέση για συγκεκριμένες λειτουργίες, την συμμόρφωση ως τα ηλεκτρικά πρωτοκολλά επικοινωνίας, την δημιουργία τυποποιημένων DTCs (Diagnostic Trouble Codes) και την συμμόρφωση ως προς την τυποποιημένη ορολογία. Αντίστοιχα, το ίδιο φαινόμενο επικράτησε και στην Ευρώπη με την Clean Air Act από το 1990 και έπειτα. Το τελικό αποτέλεσμα ήταν να

Ανάπτυξη εφαρμογής συλλογής δεδομένων από οχήματα μέσω θύρας OBD2, για λειτουργικό Android

υπάρχει σχεδόν πλήρης συμμόρφωση με τα κανονιστικά πρότυπα παγκοσμίως από το 1999 και μετά.[16]

Η τελευταία έκδοση της ψηφιακής θύρας OBD2 παρέχει σήμερα ένα τυποποιημένο ψηφιακό interface το οποίο επιστρέφει δεδομένα σε πραγματικό χρόνο και ένα σετ τυποποιημένων διαγνωστικών σφαλμάτων (DTCs) το οποίο επιτρέπει τον ταχύτατο εντοπισμό βλαβών. Αυτό δίνει πρόσβαση στα διάφορα υποσυστήματα του οχήματος στον τεχνικό συντήρησης, αλλά και, κατ' επέκταση, στον ιδιοκτήτη την δυνατότητα να διαγνώσει βλάβες πριν αυτές εξελιχθούν σε ανεπανόρθωτες. Στην παρακάτω εικόνα φαίνεται η κάτοψη των 16-pin αλλά και η λειτουργία που έχει το καθένα με βάση τον παρακάτω πίνακα.[3]



Εικόνα 2: Κάτοψη και πίνακας λειτουργίας OBD2

Σε επίπεδο hardware, τα οχήματα που είναι εξοπλισμένα με τις τελευταίες εκδόσεις της OBD2 έχουν συνήθως:

- Περίπου διπλάσιους αισθητήρες οξυγόνου από τα παλαιότερα οχήματα.
- Επιπλέον αισθητήρες οξυγόνου που είναι τοποθετημένοι στο κάτω μέρος του καταλύτη για τον έλεγχο καυσαερίων.
- Μεγαλύτερες και επεξεργαστικές μονάδες 16 ή 32bit που είναι ικανές να χειριστούν μέχρι 15000 καινούργιες μεταβλητές προς παρακολούθηση.
- EEPROM (Electronically Erasable Programmable Read Only Memory) chips που επιτρέπουν στον εγκέφαλο του οχήματος να αναβαθμιστεί ή να επαναπρογραμματιστεί, μέσω τερματικού χειρός ή διαγνωστικού υπολογιστή (Εικόνα 3).



Εικόνα 3: Εξουσιοδοτημένος μηχανικός σε διαδικασία αναβάθμισης.

- Ταυτόχρονη υποστήριξη για MAP & MAF αισθητήρες ώστε να μπορεί να γίνεται παρακολούθηση της συμπεριφοράς της μηχανής στο φορτίο και την εισαγωγή αέρα.
- Δυνατότητα προγραμματισμού διαδοχικής ανάφλεξης στην μίζα και όχι χρήση της πεταλούδας του γκαζιού (injection system).

Όπως είναι προφανές, πέραν από τις θύρες που έχουν τα τυποποιημένα πρωτόκολλα επικοινωνίας (κάποια μάλιστα διαχωρίζονται και με διαφορά δυναμικού ανάμεσα στις τάσεις που λειτουργούν), οι υπόλοιπες έχουν αφηθεί στην διακριτική ευχέρεια των κατασκευαστών. Με την ανάπτυξη δικών τους ιδιωτικών πρωτοκόλλων επικοινωνίας μπορούν να υλοποιήσουν επιπρόσθετες λειτουργίες για την κατάσταση των οχημάτων. Αυτές μπορεί να είναι η λειτουργία των αερόσακων, το κεντρικό κλειδίωμα του συναγερμού, το σύστημα ελέγχου πρόσφυσης (ESP) κ.λπ. Σαφέστατα, δεν μπορεί ο οποιοσδήποτε να στείλει ή να λάβει δεδομένα σε κάποια από αυτές τις θύρες αν δεν γνωρίζει εις βάθος το εκάστοτε πρωτόκολλο επικοινωνίας καθώς αυτό θα δημιουργούσε θέματα ασφαλείας ή θα μπορούσε να προκαλέσει ανεπανόρθωτη ζημία στην ECU (Electronic Control Unit) ή τον εγκέφαλο του αυτοκινήτου. [1],[2]

1.2 Τα πρωτόκολλα σηματοδοσίας της OBD2

Με βάση λοιπόν την παραπάνω εικόνα, θα αναλυθούν τα τυποποιημένα πρωτόκολλα σηματοδοσίας που υπάρχουν αυτήν την στιγμή στην αγορά.

1.2.1 SAE J1850

Στους ακροδέκτες 2 και 10 λειτουργεί το πρωτόκολλο J1850 Bus γνωστό και ως SAE J1850 PWM και VPW. Αναπτύχθηκε το 1994 και τα ακρωνύμια PWM (Pulse Width Modulation) και VPW (Variable Pulse Width) αναφέρονται για το σήμα της τεχνικής διαμόρφωσης του εύρους του παλμού στο οποίο λειτουργεί ο ακροδέκτης. Το σύμβολο (+) σημαίνει τιμές από 4,25 Volt έως 20 Volt και το (-) οποιαδήποτε τιμή μέχρι 3,5 Volt. Στους ακροδέκτες επιστρέφουν και αποστέλλονται συμβολοσειρές 64 και 128 bit. Η λογική που χρησιμοποιείται το bus είναι weak pull-down, δηλαδή για να ενεργοποιηθεί η ροή, ο οδηγός πρέπει να κατεβάσει την υψηλή τάση χαμηλά καθώς τα υψηλά σήματα είναι το σύνηθες. Χρησιμοποιείται CSMA/CR (Carrier Sense Multiple Access / Collision Resolution) σε frame όπως είναι η Εικόνα 4.



Εικόνα 4: J1850 Transmission Signal Format.

Το frame αποτελείται από ένα SOF (start of frame) και EOF (end of frame), το header, τα δεδομένα, δύο CRC για έλεγχο λαθών και 2 flags για τερματισμό μετάδοσης. Το πρότυπο καθορίζει όλες τις απαιτήσεις και τα χαρακτηριστικά ώστε να υπάρξει ένα πρωτόκολλο επικοινωνίας το οποίο θα είναι αρκετά φθινό για απλές εφαρμογές και αρκετά ευέλικτο για τις πιο πολύπλοκες. Μέχρι σήμερα υπάρχουν 2 υλοποιήσεις αυτού και διαφέρουν κυρίως στην ταχύτητα της μετάδοσης των δεδομένων όπου η πρώτη λειτουργεί στα 10,4 kbps και η δεύτερη στα 41,6 kbps. Σε κάθε περίπτωση, οι κατασκευαστές έχουν προνοήσει ώστε να υπάρχει δυνατότητα υιοθέτησης μίας παραλλαγής του πρωτοκόλλου ώστε να είναι δυνατή η χρήση του ανάλογα με την διάδοση που υπάρχει στο μέσον, τον σκοπό και τον τύπο της εφαρμογής. Συναντάται συνήθως σε αυτοκίνητα της General Motors. [2],[3]

1.2.2 SAE J1939

Η περαιτέρω εξέλιξη του παραπάνω πρωτοκόλλου είναι το SAE J1939. Αποτελεί την υλοποίηση για την επικοινωνία των συσκευών και αισθητηρίων με την ECU για μεγάλα οχήματα. Ουσιαστικά, είναι τρόπος υλοποίησης του διαύλου που έχει να κάνει με τράκτορες και ρυμουλκούμενα οχήματα όπως νταλίκες. Στην λίστα που ακολουθεί παρουσιάζονται με χρονολογική σειρά τα τέσσερα πρότυπα ISO:

- **ISO 11992-1 του 2002**
- **ISO 11992-2 του 2003**
- **ISO 11992-3 του 2003**
- **ISO 11992-4 του 2005**

Πολύ σημαντική εξέλιξη είναι αυτή του 2005, όπου πέραν των βαρέων οχημάτων, άρχισε να χρησιμοποιείται και σε ΙΧ με πετρελαιοκινητήρες (Diesel). Βασική διαφορά που έχει με το 1850 είναι και ο ανάπτορας που χρειάζεται για να λειτουργήσει η OBD2 καθώς η απόληξη που επιστρέφει η ECU είναι σε κυκλικό σχήμα αντί για τραπέζιο όπως απεικονίζεται στην Εικόνα 5. [4]



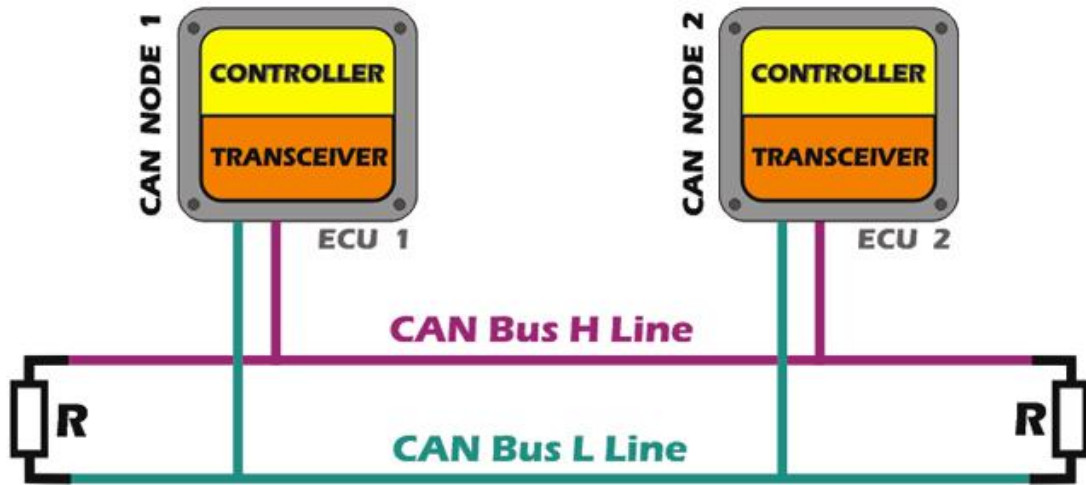
Εικόνα 5: J1939 OBD2 Adapter.

Συνοπτικά, παρακάτω εμφανίζεται η λίστα με τις κυριότερες αλλαγές που έχουν γίνει τα τελευταία πέντε χρόνια στο πρότυπο OBD:

- Οδηγός υλοποίησης για OBD2 σε Εσωλέμβιους και Εξωλέμβιους κινητήρες ναυτικών σκαφών με ανάφλεξη σπινθήρα (04/2012).
- Προτεινόμενες πρακτικές για δίκτυα σειριακής επικοινωνίας οχημάτων μεγάλων αποστάσεων (11/2012).
- Προτεινόμενες πρακτικές για δίκτυα επικοινωνίας και ελέγχου γεωργικών και δασικών οχημάτων με μηχανικό εξοπλισμό εκτός δρόμου (03/2013).
- Δίκτυα σειριακής επικοινωνίας βαρέων οχημάτων (08/2013).
- Αλλαγές στο network layer του πρωτοκόλλου (04/2014).
- Οδηγός κανονιστικής συμμόρφωσης (06/2015).
- Αλλαγές στο physical layer με χρήση UTP καλωδίου και ταχύτητα 250 Kbits/s (08/2015).
- Αλλαγές στο επίπεδο της εφαρμογής με χρήση ρυθμιζόμενων μηνμάτων (09/2015).
- Προσθήκη αυτόματης ανίχνευσης ταχύτητας μεταφοράς δεδομένων (10/2015).
- Οδηγός υλοποίησης OBD2 (11/2015).
- Προσθήκη υποστήριξης για γεννήτριες και βιομηχανικό εξοπλισμό στο επίπεδο της εφαρμογής (11/2015).
- Προσθήκη δυνατότητας διαγνωστικών (01/2016).
- Οδηγός συμβατότητας επικοινωνίας της OBD2 για τους κινητήρες σε οχήματα υψηλής αντοχής(01/2016).
- Αλλαγές στην διαχείριση δικτύου (02/2016).
- Αλλαγές στο data link layer (03/2016).
- Αλλαγές στο επίπεδο της εφαρμογής του οχήματος (10/2016).
- Αλλαγές στην μη ενσωματωμένη διαγνωστική διεπαφή (10/2016).
- Αλλαγές στο physical layer με ταχύτητα 500 Kbits/s (12/2016).
- Αλλαγές στο physical layer με χρήση θωρακισμένου UTP καλωδίου και ταχύτητα 250 Kbits/s (12/2016).
- Αλλαγές στην διαχείριση δικτύου (03/2017).
- Αλλαγές στο επίπεδο της εφαρμογής του οχήματος (05/2017).
- Αλλαγές σε Εσωλέμβιους και Εξωλέμβιους κινητήρες ναυτικών σκαφών με ανάφλεξη σπινθήρα (04/2017).[4]

1.2.3 Raw CAN (ISO 11898)

Στους ακροδέκτες 6 και 14 βασίζεται το πρωτόκολλο CAN Bus (Connected Area Network). Είναι ένα πρωτόκολλο το οποίο επιτρέπει μέσω του διαύλου επικοινωνίας να επικοινωνούν οι συσκευές μεταξύ τους χωρίς να χρειάζεται κεντρική επεξεργαστική μονάδα. Αυτό του δίνει το βασικό πλεονέκτημα στην πολυπλεξία των δεδομένων και δυνατότητα για καταναμημένο έλεγχο σε πραγματικό χρόνο. Βασική προϋπόθεση είναι ότι όλοι οι ακροδέκτες του εκάστοτε node πολώνονται από την ίδια πλευρά π.χ. όλοι οι controller στην γραμμή high (pin6) και όλοι οι transceiver στην γραμμή low (pin 14) όπως φαίνεται ακριβώς στην Εικόνα 6. [7]



Εικόνα 6: Διασύνδεση των High-Low ακροδεκτών σε CAN Bus

Τα πρότυπα που το καθορίζουν είναι 5. Τα δύο πρώτα είναι το 2003 ISO 11898-1 & ISO 11898-2. Στο πρώτο υπάρχει περιγραφή του τρόπου ζεύξης, της υποδομής για ανταλλαγή μηνυμάτων καθώς και την χρήση των MAC address στους διάφορους nodes. Στο δεύτερο υπάρχει καθορισμός της ταχύτητας στο 1 Mbit/s με την χρήση των MAC. Στο 2004 υπάρχει το ISO 11898-3 όπου γίνεται τυποποίηση των χαρακτηριστικών των ηλεκτρονικών μονάδων των ECU ώστε αυτές να μπορούν να ανταλλάζουν ψηφιακά δεδομένα με τις συσκευές και να υποστηρίζουν ταχύτητες μετάδοσης από 40 έως και 125 Kbits/s. Το ISO 11898-4 το 2006 καθορίζει τα trigger που ξεκινούν την επικοινωνία μεταξύ των ECU και των nodes. Τέλος, το ISO 11898-5 το 2007 είναι μία μετεξέλιξη του 11898-2 με αλλαγές κυρίως στα πρότυπα όσον αφορά στο φυσικό μέσο. [5],[6],[7],[8]

1.2.4 CAN Bus (ISO 15765)

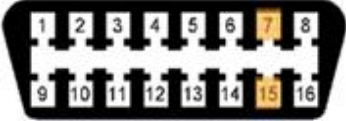
Ουσιαστικά, είναι η μετεξέλιξη του raw CAN και υπάρχουν 4 πρότυπα που το καθορίζουν και συνοψίζονται στην παρακάτω λίστα:

- **ISO 15765-1** (2011): Καθορισμός της ορολογίας που θα χρησιμοποιηθεί για όλες τις εκδόσεις και έπειτα. Γίνεται αναλυτική παρουσίαση της σχέσης των μερών που το αποτελούν αλλά και της δομής του πρότυπου. [9]
- **ISO 15765-2** (2011): Περιγράφεται ο τρόπος αποστολής των πακέτων από node σε node μέσα σε ένα δίκτυο CAN. Ακόμα καθορίζεται η μέθοδος όπου γίνεται κατακερματισμός σε πολλαπλά πακέτα των μηνυμάτων (επιτρέποντας να είναι το μέγεθος του άνω των 8 bytes) και πως γίνεται η επανένωση τους από τον δέκτη με την βοήθεια μεταδεδομένων. [10]
- **ISO 15765-3** (2012): Είναι η αναθεώρηση του πρότυπου για τις απαιτήσεις ενός σετ διαγνωστικών υπηρεσιών για επιβατικά αυτοκίνητα. [12]
- **ISO 15765-4** (2013): Παρέχει όλες τις απαραίτητες πληροφορίες για την εγκαθίδρυση ενός δικτύου με εξωτερικό εξοπλισμό για την διάγνωση και επιδιόρθωση βλαβών. Δίνει την δυνατότητα plug and play επικοινωνίας και οι ρυθμοί μετάδοσης των δεδομένων είναι από 250 έως 500 Kbit/s. [13],[15]

1.2.5 Keyword Protocol 2000

Είναι ένα άλλο από τα πρωτόκολλα σηματοδότησης το οποίο καλύπτει το φάσμα των εφαρμογών σε επίπεδο session, διασφαλίζοντας την σωστή επικοινωνία για την έναρξη, διατήρηση και τερματισμό του κάθε socket. Όπως φαίνεται και στην Εικόνα 7, λειτουργεί μέσω των ακροδεκτών 7 και 15, έχει μέγιστο μέγεθος συμβολοσειράς τα 255 bytes, αλλά χαμηλό ρυθμό μετάδοσης που κυμαίνεται μόλις από 1,2 έως 10,4 kbps. Η τυποποίηση του είναι το ISO 14230-4. [14]

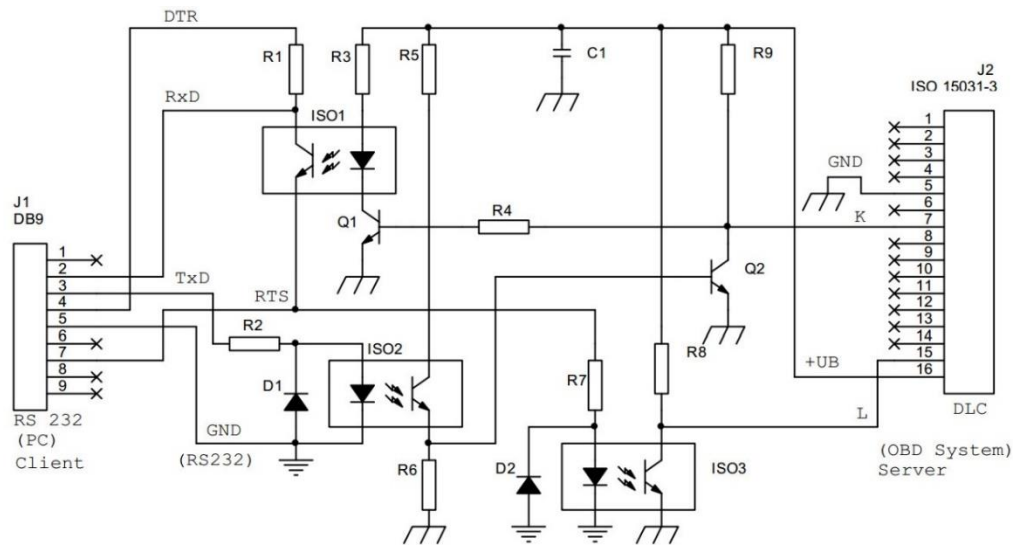
ISO 14230 KWP2000

- ISO 14230 KWP2000 (Keyword Protocol 2000)
 - pin 7: K-line
 - pin 15: L-line (optional)
- 
- Physical layer identical to ISO 9141-2
 - Data rate 1.2 to 10.4 kbaud
 - Message may contain up to 255 bytes in the data field

Εικόνα 7: Τα χαρακτηριστικά KWP 2000

1.2.6 ISO 9141-2

Όπως και το Keyword protocol 2000 που προαναφέρθηκε, το 9141-2 λειτουργεί και αυτό μέσω των pins 7 και 15. Η διαφορά είναι ότι εδώ το μέγιστο μέγεθος της συμβολοσειράς του είναι 12 bytes και έχει ασύγχρονο ρυθμό μετάδοσης στα 10,4 kbps. Αυτό το κάνει να λειτουργεί σαν το γνωστό πρωτόκολλο RS-232 που υπάρχει στις σειριακές θύρες των PC. Δεν χρησιμοποιεί την διαδικασία του handshake παρόλο που υπάρχει μονή αμφίδρομη επικοινωνία στο μέσω διασύνδεσης. Με την ανάλογη παραμετροποίηση που περιγράφεται στο πρότυπο είναι ικανό να μεταφέρει την πληροφορία με τον ίδιο τρόπο όπως φαίνεται στην Εικόνα 8 σε έναν client. Συναντάται κυρίως σε ευρωπαϊκά και ασιατικά οχήματα καθώς και την αυτοκινητοβιομηχανία της Chrysler. [15],[16]



Εικόνα 8: Διάταξη με PC-Client για το ISO 9141-2

1.3 OBD2 Modes

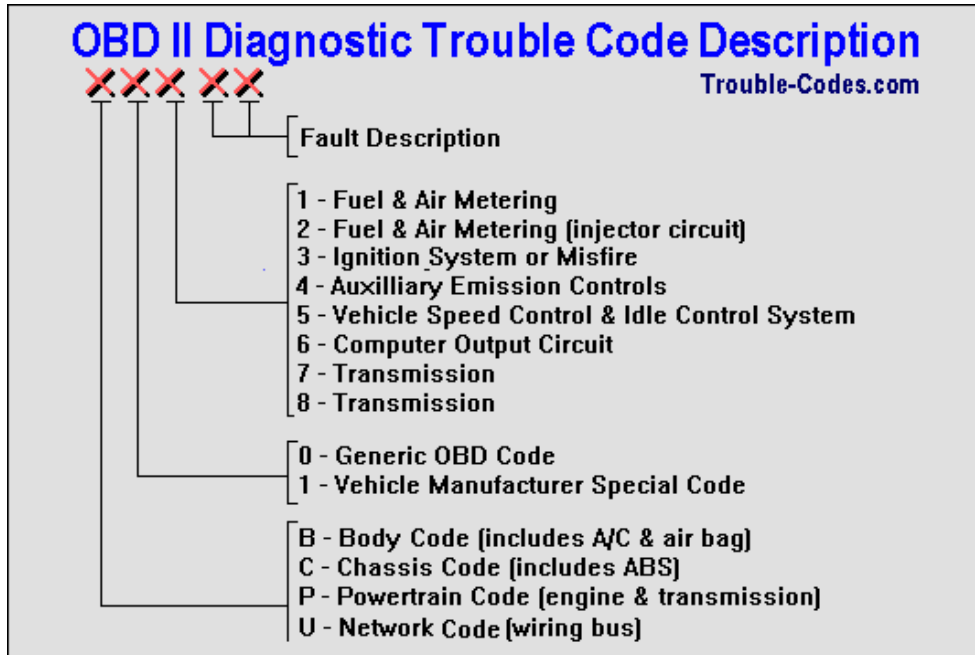
Με τον όρο OBD2 modes αποσαφηνίζονται οι υπηρεσίες που μπορεί να προσφέρει η διεπαφή OBD2 και τον τρόπο επικοινωνίας με αυτές. Κάθε υπηρεσία αναπαρίσταται με την μορφή \$XX. Το κάθε X αναπαριστά ένα δεκαεξαδικό ψηφίο (από 0 έως F). Αυτήν την στιγμή είναι υλοποιημένες και λειτουργούν παγκοσμίως 10 τέτοιες υπηρεσίες όπως φαίνεται στον παρακάτω πίνακα.

MODE (Hex)	Περιγραφή λειτουργίας
01	Εμφάνιση παρόντων δεδομένων.
02	Εμφάνιση των freeze frame δεδομένων.
03	Εμφάνιση των αποθηκευμένων Diagnostic Trouble Codes.
04	Καθαρισμός τοπικής μνήμης αποθηκευμένων Diagnostic Trouble Codes.
05	Εμφάνιση τιμών αισθητήρα οξυγόνου (όχι για CAN Bus).
06	Εμφάνιση τιμών αισθητήρα οξυγόνου (για CAN Bus).
07	Εμφάνιση DTC τα οποία όμως έχουν ανιχνευθεί στον τελευταίο κύκλο οδήγησης.
08	Διεπαφή για την διεργασία ελέγχου της λειτουργίας της OBD2.
09	Διεπαφή για περαιτέρω πληροφορίες του οχήματος.
0A	Εμφάνιση μόνιμων DTC τα οποία σβήνονται μόνο με μόνιμη αποκατάσταση της βλάβης

Συνεπώς, κάθε service το οποίο καλείται προς εκτέλεση είτε επιστρέφει πίσω κάποια τιμή είτε εκτελεί κάποια λειτουργία, όπως για παράδειγμα τον καθαρισμό της προσωρινής μνήμης. Οι κατασκευαστές στην αυτοκινητοβιομηχανία δεν είναι υποχρεωμένοι να υλοποιούν όλες τις παραπάνω υπηρεσίες αλλά μπορούν να δίνουν τις ίδιες πληροφορίες μέσω και δικών τους υλοποιήσεων.

1.3.1 Diagnostic Trouble Codes

Η τυποποίηση που αναφέρθηκε στα προηγούμενα κεφάλαια έχει παίξει σημαντικό ρόλο στη δυνατότητα συμβατότητας σε όλες τις διαγνωστικές εφαρμογές που χρησιμοποιούν το OBD2. Κάθε DTC το οποίο επιστρέφεται προς τα πίσω σαν τιμή έχει την μορφή X XX XX. Κάθε χαρακτήρας έχει μία διαφορετική σημασία. Στην Εικόνα 9 παρουσιάζεται μία σύντομη επεξήγηση αυτών.



Εικόνα 9: Συνοπτική ανάλυση των χαρακτήρων DTC

Από αριστερά προς τα δεξιά θα αναλυθούν οι παραπάνω χαρακτήρες. Ο πρώτος χαρακτήρας αναφέρεται στο σημείο του οχήματος που εντοπίζεται η βλάβη. Μπορεί να πάρει τις τιμές B,C,P,U. Το γράμμα B αναφέρεται στο αμάξωμα του οχήματος και συμπεριλαμβάνει την μονάδα κλιματισμού και το σύστημα ενεργοποίησης των αερόσακων. Το γράμμα C αναφέρεται στο σασί του οχήματος και περιλαμβάνει και το αυτόματο σύστημα ελέγχου πέδησης (ABS). Το γράμμα P αναφέρεται στο σύστημα μετάδοσης της κίνησης και στον κινητήρα. Το γράμμα U αφορά το δίκτυο καλωδίωσης του οχήματος και το σύστημα εσωτερικής επικοινωνίας των συσκευών αυτού.

Ο δεύτερος χαρακτήρας είναι δυαδικού τύπου και μπορεί να πάρει μόνο τις τιμές 0 και 1. Ο ρόλος του είναι να διαχωρίσει τον τύπο του DTC, δηλαδή να καθορίσει αν είναι τυποποιημένος κωδικός ή αποτελεί ξεχωριστή υλοποίηση του κατασκευαστή.

Ο τρίτος χαρακτήρας είναι αριθμητικός και έχει την δυνατότητα να πάρει τιμές από το 1 έως το 8. Κάθε αριθμός έχει διαφορετική σημασία για το σύστημα όπως φαίνεται παρακάτω:

- No 1: Παράμετρος καυσίμων και ροής αέρα για βενζινοκινητήρες.
- No 2: Παράμετρος καυσίμων και ροής αέρα για πετρελαιοκινητήρες.
- No 3: Πρόβλημα στην μίζα (σπινθηριστής) ή πρόβλημα στον χρονιστή ανάφλεξης.
- No 4: Πρόβλημα στο βοηθητικό - εφεδρικό σύστημα ελέγχου των ρύπων.
- No 5: Πρόβλημα στο κύκλωμα ελέγχου ταχύτητας του οχήματος και στον ρυθμιστή ελαχίστων στροφών μηχανής (ρελαντί).
- No 6: Πρόβλημα στην έξοδο του συστήματος των ηλεκτρονικών κυκλωμάτων.
- No 7: Πρόβλημα στο σύστημα μετάδοσης.
- No 8: Πρόβλημα στα δευτερεύοντα συστήματα μετάδοσης.

Η μίξη των δύο παραπάνω κατηγοριών οδηγεί στην δημιουργία του παρακάτω πίνακα ώστε να γίνει πιο προφανής η μορφή του εκάστοτε DTC. Οι δύο τελευταίοι χαρακτήρες είναι αλφαριθμητικοί και χρησιμεύουν στη πιο αναλυτική περιγραφή της βλάβης.

Powertrain	Body Codes
P0XXX – Γενικό σφάλμα	B0XXX – Γενικό σφάλμα
P1XXX – Ειδικό σφάλμα κατασκευαστή	B1XXX – Ειδικό σφάλμα κατασκευαστή
P2XXX – Γενικό σφάλμα	B2XXX – Ειδικό σφάλμα κατασκευαστή
P30XX έως P33XX – Ειδικό σφάλμα κατασκευαστή	B3XXX – Γενικό σφάλμα
P34XX έως P39XX – Γενικό σφάλμα	
Chassis Codes	Network Communication Codes
C0XXX – Γενικό σφάλμα	U0XXX – Γενικό σφάλμα
C1XXX – Ειδικό σφάλμα κατασκευαστή	U1XXX – Ειδικό σφάλμα κατασκευαστή
C2XXX – Ειδικό σφάλμα κατασκευαστή	U2XXX – Ειδικό σφάλμα κατασκευαστή
C3XXX – Γενικό σφάλμα	U3XXX – Γενικό σφάλμα

Για να γίνει όμως περισσότερο κατανοητό πρέπει να αντιληφθεί κανείς ότι ακόμα και με την παραπάνω κατηγοριοποίηση η λυχνία του check engine δεν θα ανάψει απαραίτητα σε όλες τις περιπτώσεις. Αυτό συμβαίνει διότι τα DTC χωρίζονται σε τέσσερις βασικές κατηγορίες. Αυτές είναι τα type A,B,C και D. Η βασική διαφοροποίηση έχει να κάνει με την εξής παραδοχή. Έστω ότι ο μηχανικός συντήρησης λαμβάνει ένα DTC στην διαγνωστική του συσκευή με κωδικό P0300. Αυτό είναι ένα τυχαίο misfire code το οποίο θα μπορούσε να πάρει τις εξής ερμηνείες:

- Πρόβλημα στην εισαγωγή αέρα.
- Χαμηλή πίεση βενζίνης.
- Πρόβλημα στους σπινθηριστές (μπουζί).
- Πρόβλημα ανάφλεξης λόγω φθαρμένων καλωδίων των μπουζί.
- Καμέντοι πυκνωτές ή πηνία στο κύκλωμα ανάφλεξης.

Για να μην υπάρχει μεγάλη πιθανότητα το OBD2 να επιστρέφει συχνά check engine λόγω misfire, έχει γίνει προγραμματισμός να ανάβει μόνον όταν εντοπίζει την ίδια βλάβη για δεύτερη φορά σε ίδιες συνθήκες ταξιδιού. Από την στιγμή που η αυτόματη διάγνωση ανά κύλινδρο αντιληφθεί κάτι τέτοιο, δίνει μέχρι 2 % πιθανότητα να μην ενεργοποιήσει το MIL. Έτσι λοιπόν, τα type A error codes που θεωρούνται τα πιο σοβαρά θα ενεργοποιήσουν αμέσως την ενδεικτική λυχνία. Επιπλέον, θα αποθηκευτούν στην μνήμη ως error code και ταυτόχρονα θα δημιουργηθεί ένα failure record και ένα freeze frame ώστε να μπορεί να γίνει περαιτέρω ανάλυση μέσω του reporting.

Αντίστοιχα, τα DTC type B είναι μικρότερης σημασίας και θα πρέπει να συμβούν τουλάχιστον δύο φορές για να ενεργοποιηθεί η ένδειξη αλλά και για αυτά ισχύουν οι ίδιες συνθήκες αποθήκευσης με τα A. Τα DTC type C συνήθως αφορούν δευτερεύοντα συστήματα όχι και τόσο σημαντικά για την ορθή λειτουργία του οχήματος. Οι κατασκευαστές τείνουν τα τελευταία χρόνια

να ενεργοποιούν άλλους λαμπτήρες για να δείξουν το σφάλμα. Χαρακτηριστικό παράδειγμα αποτελεί η στάθμη του καθαριστικού υγρού των τζαμιών στο σύμβολο 4. Μια επτοπτική υλοποίηση φαίνεται στην παρακάτω εικόνα.



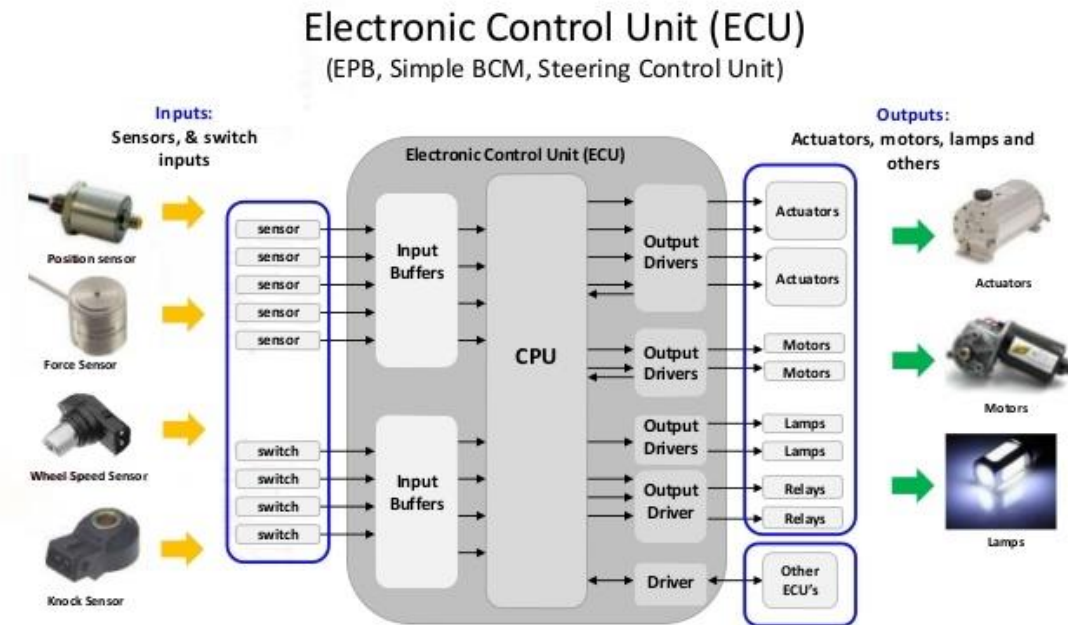
Εικόνα 10: Λυχνίες DTC

Τα DTC type D εμφανίζονται μόνο μέσω διαγνωστικής εφαρμογής και δεν ενεργοποιούν λαμπτήρες. Αξίζει σε αυτό το σημείο να αναφερθεί ότι, ασχέτως των ενεργειών ενός διαγνωστικού προγράμματος, το ίδιο το όχημα εκτελεί από μόνο του self-diagnostics κάθε ένα driver cycle. Αυτό πρακτικά σημαίνει πως για να ολοκληρωθεί ένα τέτοιο test, ο κινητήρας θα πρέπει να παραμείνει ενεργός έως το ψυκτικό υγρό του φτάσει την θερμοκρασία των 40 βαθμών κελσίου αν θεωρηθεί ότι γίνεται εκκίνηση από εξωτερική θερμοκρασία που δεν ξεπερνάει τους 25 βαθμούς. Είναι δε προφανές ότι ακόμα και αν κάποιος χειροκίνητα παρέμβει με το διαγνωστικό και σβήσει ένα error code, αυτό μετά από λίγο καιρό θα επανέλθει αν η βλάβη δεν έχει λυθεί πραγματικά.[35]

1.3.2 Electronic Control Unit (ECU)

Πριν την ανάλυση της εφαρμογής που πραγματεύεται η εργασία, είναι βασικό να αναλυθούν τα επιμέρους κομμάτια τα οποία θα αποτελέσουν τα σημεία ελέγχου που θα εξεταστούν επάνω στο όχημα. Το κύριο σημείο ένωσης όλων των ηλεκτρικών και μηχανικών υποσυστημάτων είναι η ECU. Κάθε σύστημα που επικοινωνεί με αυτή, στις εισόδους της έχει αντίστοιχα και την ανάλογη έξοδο που λαμβάνει σήματα ελέγχου.[36] Στην παρακάτω εικόνα παρουσιάζεται μία τυπική διάταξη.

Όπως είναι προφανές από την φωτογραφία, το κάθε pin μπορεί να συνδεθεί με την ανάλογη μονάδα και να παρέχει ή να δεχθεί δεδομένα για μία διεργασία. Παρακάτω παρατίθεται μία συνοπτική λίστα τέτοιων μονάδων.



Εικόνα 11: Μερικές από τις εισόδους και εξόδους που έχουν οι σύγχρονες ECU

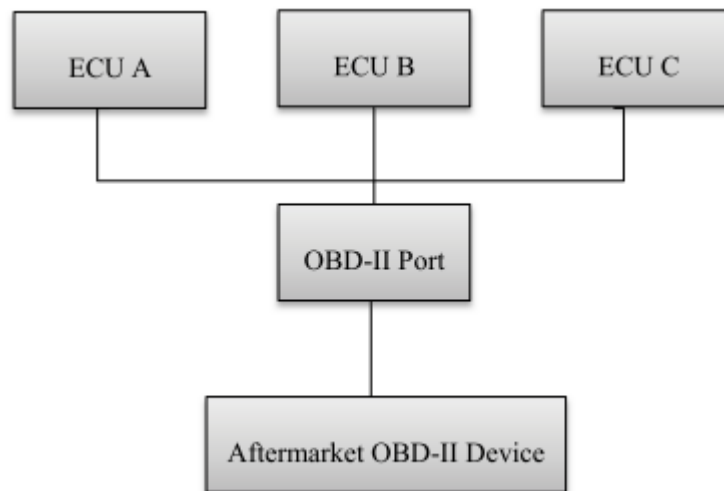
- Engine Control Unit:** Αποτελεί την βασική λειτουργία ελέγχου του κινητήρα και ίσως είναι η μόνη που δέχεται πολλαπλά ερεθίσματα και από άλλες εισόδους της ECU. Βασικές λειτουργίες που εκτελεί είναι να ελέγχει τους χρονιστές για την ανάφλεξη του καυσίμου στον θάλαμο καύσης. Αν τυχόν αυτό υστερήσει έστω και λίγο, ελλοχεύει ο κίνδυνος μόνιμης καταστροφής του κινητήρα. Επίσης ελέγχει τις στροφές του ρελαντί, ώστε ο κινητήρας να λειτουργεί αδιάληπτα ακόμα και όταν η θέση του πεντάλ στο γκάζι είναι μηδενική. Επίσης, αναλαμβάνει τον έλεγχο της στοιχειομετρικής αναλογίας του μίγματος (π.χ. 14,7 για τους βενζινοκινητήρες) που καίγεται και επηρεάζεται άμεσα από τον αισθητήρα MAF που δίνει την ποσότητα παροχής οξυγόνου για τέλεια καύση. Επιπλέον, ελέγχεται η λειτουργία του turbocharger αν το όχημα τον υποστηρίζει και ο χρονισμός των βαλβίδων, που αφορά την ταχύτητα εναλλαγής στον άνοιγμα και κλείσιμο του καπακιού ενός κυλίνδρου με αποτέλεσμα την βελτίωση της ιπποδύναμης.
- Transmission Control Unit:** Η δεύτερη πιο σημαντική μονάδα είναι αυτή της μετάδοσης της κίνησης. Έχει άμεση σχέση με την αλλαγή σχέσης στο κιβώτιο ταχυτήτων για την επιλογή της σωστής σχέσης κυρίως στα αυτόματα οχήματα. Πέραν τις οικονομίας καυσίμου, εκτελεί αρκετές ακόμα εργασίες καθώς δέχεται πάρα πολλές μεταβλητές από διαφορετικούς αισθητήρες. Μερικοί από αυτούς είναι ο αισθητήρας της ταχύτητας, της θέσης του γκαζιού, της θερμοκρασίας του λιπαντικού μετάδοσης (για βαρέα οχήματα με ρυμουλκό) στις υδραυλικές υποδοχές, του διακόπτη των φαναριών πέδησης λόγω φρεναρίσματος, κ.λπ. Σε πιο νέα οχήματα αξίζει να αναφερθεί ότι έχει ενσωματωθεί στην μονάδα αυτή στο σύστημα του Cruise Control για την πορεία ταξιδιού καθώς και το Traction Control ώστε να αποφεύγεται να εκτραπεί το όχημα από την πορεία του λόγω υπέρβασης της ροπής σε μία στροφή.
- Brake Control Unit:** Αποτελεί την καρδιά του συστήματος πέδησης που διατηρεί τον έλεγχο της ταχύτητας του οχήματος ακόμα και σε συνθήκες πανικού. Εδώ περιέχεται η υλοποίηση του ABS για τον έλεγχο του αντιμπλοκαρίσματος των τροχών καθώς και αυτή

του ESP για την ανίχνευση εκτροπών λόγω ολισθηρότητας στο οδόστρωμα παράλληλα με τον έλεγχο των καιρικών συνθηκών εκτός του οχήματος (υγρασία, θερμοκρασία κ.λπ.).

- **Body Control Unit:** Εδώ πραγματοποιείται ο εποπτικός έλεγχος του αμαξώματος. Κεντρικό κλειδίωμα, ηλεκτρικά παράθυρα, φώτα καμπίνας, κλιματισμός, immobilizer αποτελούν τα σημεία ελέγχου και ειδοποίησης.
- **Airbag Control Unit:** Εδώ περιέχονται μηχανισμοί πυροδότησης προστατευτικών αερόσακων σε περίπτωση πρόσκρουσης και ατυχήματος, Βασικά σημεία που το σύστημα επικοινωνεί είναι τα επιταχυνσιόμετρα, τυχόν γυροσκόπια κ.α.
- **Telematics Control Unit:** Σε πιο νέα οχήματα, δίνεται η δυνατότητα, με χρήση αισθητήριων GPS και δεδομένα κινητής τηλεφωνίας, να παρέχονται υπηρεσίες απομακρυσμένης διαχείρισης του οχήματος. Για παράδειγμα, σε περίπτωση ατυχήματος δηλώνεται ο ακριβής εντοπισμός της θέσεως του οχήματος ενώ σε άλλη περίπτωση ο χρήστης μπορεί να ενεργοποιήσει λειτουργίες του οχήματος (φώτα, κλιματισμό, την μίζα της μηχανής, κ.α.) απομακρυσμένα.

1.3.3 Θέματα ασφαλείας

Χωρίς η παρούσα εργασία να πραγματεύεται τα ζητήματα ασφαλείας κατά την επικοινωνίας μέσω της OBD2, θα ήταν σκόπιμο να γίνει και μία συνοπτική αναφορά στα θέματα που τίθενται όταν κάποιος αλληλοεπιδρά με την θύρα OBD2 και με την ECU του εγκεφάλου κατ' επέκταση. Έχοντας πρόσβαση στο εσωτερικό της καμπίνας του αυτοκινήτου (ακόμα και μετά από κακόβουλη ενέργεια με σκοπό αποκλειστικά την κλοπή), κάποιος που έχει την γνώση να εκμεταλλευθεί τα όποια κενά ασφαλείας υπάρχουν στο πρωτόκολλο μπορεί να προκαλέσει μία σειρά από ζημιές αλλά ακόμα και την απώλεια ανθρώπινης ζωής. Το μοντέλο απειλής που παρουσιάζεται στην εικόνα φανερώνει τον τρόπο απειλής όπου κάποιος θα προσπαθήσει να αλληλοεπιδράσει με τις επιμέρους μονάδες του οχήματος.



Εικόνα 12: Μοντέλο απειλής μέσω OBD2 θύρας

Ένα από τα πρώτα περιστατικά που έχουν καταγραφεί με αποκλειστικό σκοπό την κλοπή οχήματος, ήταν το 2012 στο Λονδίνο, όπου καταγράφηκε από κάμερα ασφαλείας η παραβίαση και κλοπή ΙΧ επιβατικού αυτοκινήτου στο οποίο οι κακοποιοί έκαναν jumpstart τον κινητήρα χρησιμοποιώντας OBD2 dongle και ανάλογο λογισμικό.

Από τους βασικότερους λόγους που υπάρχει τέτοιο vulnerability είναι γιατί ότι το ίδιο το πρωτόκολλο CAN λειτουργεί με broadcast μετάδοση. Ως αποτέλεσμα, κάθε frame το οποίο αποστέλλεται σε κάποιο node, απλά χρησιμοποιεί την source και destination IP Address ώστε να αναγνωρίσει εάν το πακέτο προορίζεται για αυτό. Αυτό συμβαίνει για δύο βασικούς λόγους, ο πρώτος έχει να κάνει με την λογική ότι κάποιος για να παρέμβει σε ένα κλειστό δίκτυο ενός οχήματος θα πρέπει να έχει φυσική παρουσία το οποίο, εν γένει, είναι αρκετά δύσκολο χωρίς να

γίνει αντιληπτός. Ο δεύτερος λόγος είναι ότι η κατασκευάστριες εταιρίες χρησιμοποιώντας το πρωτόκολλο CAN Bus προσπαθούν να εξαλείψουν την καθυστέρηση λόγω της μετάδοσης της πληροφορίας σε δέκατα του δέκατου του δευτερολέπτου. Για την ακρίβεια, αρκεί να αναλογιστεί κανείς την διαφορά που θα έκανε ένα έξτρα millisecond, αν για παράδειγμα υπήρχε αυθεντικοποίηση του εκάστοτε πακέτου προς στην συσκευή για την οποία προορίζεται, όταν αυτό το millisecond, μπορεί να κάνει την διαφορά κατά το άνοιγμα ενός αερόσακου σε περίπτωση ατυχήματος. Συγκρίνοντας τα υπέρ και τα κατά σε ένα τέτοιο δίλημμά, σαφέστατά μία μονάδα συστήματος σε ένα όχημα θα πρέπει να λειτουργεί κάτω από συγκεκριμένες ανοχές και να αντιδρά σχεδόν ακαριαία σε περίπτωση που χρειαστεί.

Ένα όχημα όμως με θέματα ασφαλείας στο εσωτερικό του δίκτυο μπορεί να γίνει μη ασφαλές όταν βρεθεί απέναντι σε κάποιον, ο οποίος είναι ικανός να εκμεταλλευθεί την φυσική λογική των ελεγκτών και των μηχανισμών αυτών προς δικό του όφελος. Ένα aftermarket OBD2 dongle μπορεί να δεχτεί κακόβουλο κώδικα ακόμα και από απόσταση (Bluetooth ή Wireless pairing) και να τον μεταδώσει στο εσωτερικό δίκτυο του CAN κυριολεκτικά αφιltrάριστο. Ακολουθεί μία λίστα με πιθανά σενάρια που μπορεί να προκαλέσει ένα τέτοιο malicious-code injection:

Σενάρια σοβαρής επίπτωσης:

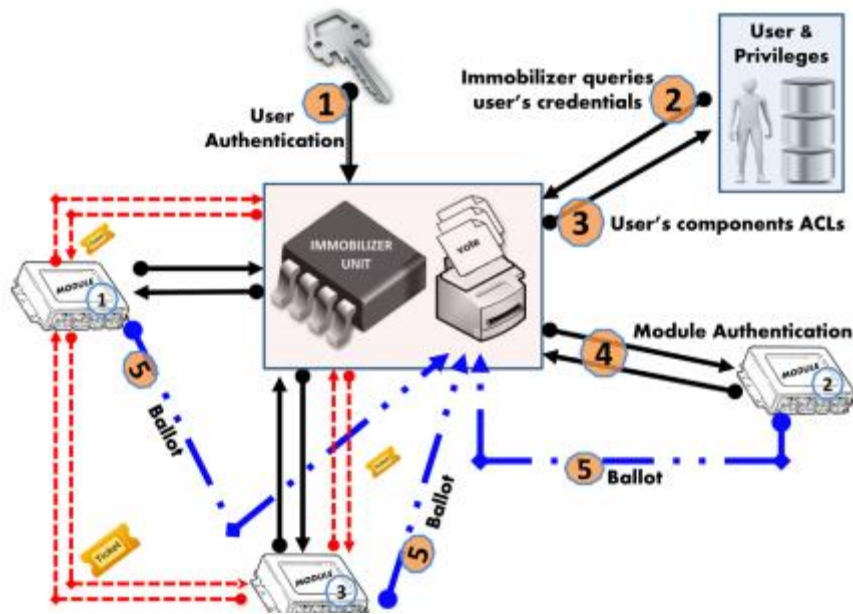
- Εξ' αποστάσεως σβήσιμο κινητήρα σε λειτουργία ή υποβάθμιση απόδοσης αυτού.
- Αλλαγή κατεύθυνσης τιμονιού (μόνο για οχήματα με καλωδιακό σύστημα πλοήγησης).
- Αποπροσανατολισμός οδηγού (απότομη αλλαγή έντασης φωνής, ενεργοποίηση ηλεκτρικών συστημάτων π.χ. υαλοκαθαριστήρες, παράθυρα κ.λπ.)

Σενάρια λιγότερο σοβαρής επίπτωσης:

- Απομακρυσμένη παρακολούθηση πορείας οχήματος μέσω του ενσωματωμένου GPS.
- Κλοπή οχήματος ή περιεχομένων αυτού λόγω απενεργοποίησης immobilizer και συναγερμού.
- Προσπάθεια απάτης σε ενοικίαση ή σε βάρος ασφαλιστικής εταιρείας.
- Κίνδυνος για επίθεση σε κινητές συσκευές που συνδέονται με το όχημα.
- Κλοπή προσωπικών πληροφοριών με βάση το προηγούμενο.
- Υποκλοπή ακόμα και των συνομιλιών που εκτελούνται μέσα μια καμπίνα.

Κάθε περίπτωση από τις παραπάνω μπορεί να φαίνεται σε κάποιους ακόμα και σενάριο φαντασίας. Αρκεί όμως μόνο να αναλογιστεί κανείς ότι όσο όλες οι συσκευές που χρησιμοποιούνται, συνδέονται στο διαδίκτυο ή σε συσκευές που συνδέονται στο διαδίκτυο, όλα έχουν τρόπο να παραβιαστούν. Αντιστοίχως, η πλατφόρμα που μπορεί να επηρεάσει κάτι τέτοιο μπορεί να είναι ένα κινητό τηλέφωνο. Οπότε το επιχείρημα ότι μία επίθεση για να πραγματοποιηθεί χρειάζεται απαραίτητα φυσική παρουσία, καταρρίπτεται.

Ευτυχώς τα τελευταία χρόνια αρκετοί αναλυτές έχουν ασχοληθεί με το θέμα και αυτό έχει αρχίσει να εφαρμόζεται και στις υλοποιήσεις των κατασκευαστών στην βιομηχανία. Η απομόνωση του εκάστοτε module από την ECU, η τεχνική της διπλής αυθεντικοποίησης με την μονάδα του immobilizer σε συνδυασμό με το ticketing protocol (Εικόνα 13), αλλά και η δυνατότητα να κλειδώνει μία μονάδα όταν αρχίζει να δέχεται σήματα τέτοιου τύπου είναι μόνο μερικές από αυτές.



Εικόνα 13: Τεχνική προστασίας στο immobilizer

Επιπλέον, πέρα από την ασφάλιση του οχήματος, θα πρέπει να υλοποιηθεί και ασφάλιση από πλευράς OBD2 dongle. Η πιθανότητα κάποιος χρήστης να αφήσει τα default credentials του raicing είναι πάρα πολύ μεγάλη, (π.χ. στην περίπτωση της παρούσας υλοποίησης, το password είναι το default: 1234). Επίσης, η τυχόν αναβάθμιση του firmware του dongle θα πρέπει να γίνεται μόνο με ψηφιακά κρυπτογραφημένα πακέτα ώστε να εμποδιστεί και με αυτόν τον τρόπο η τυχόν εισροή κακόβουλου λογισμικού.[17],[18],[19]

ΚΕΦΑΛΑΙΟ 2

2.1 Υλοποίηση

2.1.1 Ξεκινώντας

Πριν την έναρξη ανάπτυξης της εφαρμογής, έπρεπε να ρυθμιστεί το ζήτημα του περιβάλλοντος ανάπτυξης και δοκιμών. Η εφαρμογή αναπτύχθηκε χρησιμοποιώντας το Android SDK με χρήση του Android Studio IDE και του Java Development Kit (JDK) τα οποία θα πρέπει να θεωρούνται δεδομένα για την ανάπτυξη native εφαρμογών για Android OS και είναι αντικείμενα τα οποία δεν πραγματεύεται η παρούσα εργασία.

Συνεπώς, δημιουργήθηκε μια εφαρμογή ανεπτυγμένη για να εκτελείται σε συσκευές με λειτουργικό σύστημα Android, η οποία έχει ως κύριο σκοπό την συλλογή real-time δεδομένων από τον εγκέφαλο του οχήματος και από το περιβάλλον. Η εφαρμογή αποτελείται από ένα απλό user interface, καθώς η κύρια λειτουργία της είναι το data collection και όχι η γραφική απεικόνιση. Η διασύνδεση με τον εγκέφαλο του οχήματος επιτυγχάνεται μέσω της θύρας OBD και, πιο συγκεκριμένα, μέσω ενός OBD2 Bluetooth Adapter του οποίου η προμήθεια έγινε από το εμπόριο και απεικονίζεται παρακάτω. Το κόστος ανήλθε περίπου στα 20 €.



Εικόνα 14: Το OBD2 Dongle

Το dongle αυτό έχει εγκατεστημένο τον μικροελεγκτή PIC 18F2480 (ELM327) της ELM Electronics και είναι από τους πιο συνηθισμένους στην αγορά, όσον αφορά τα συστήματα On Board Diagnostics. Το ELM327 συνοψίζει τις δυνατότητες των πρωτόκολλων που αναφέρθηκαν στο προηγούμενο κεφάλαιο και δίνει ένα σχετικά απλό interface UART (Universal Asynchronous Receiver - Transmitter), δηλαδή έναν ασύγχρονο πομποδέκτη, ο οποίος μπορεί να έχει δυνατότητα σύνδεσης με την εφαρμογή και μέσω USB, RS-232, Wi-Fi. Αυτός επικοινωνεί με την ECU βάση τα παρακάτω πρωτόκολλα:

- SAE J1850 PWM (41.6 kbit/s)
- SAE J1850 VPW (10.4 kbit/s)
- ISO 9141-2 (5 baud init, 10.4 kbit/s)
- ISO 14230-4 KWP (5 baud init, 10.4 kbit/s)
- ISO 14230-4 KWP (fast init, 10.4 kbit/s)
- ISO 15765-4 CAN (11 bit ID, 500 kbit/s)

- ISO 15765-4 CAN (29 bit ID, 500 kbit/s)
- ISO 15765-4 CAN (11 bit ID, 250 kbit/s)
- ISO 15765-4 CAN (29 bit ID, 250 kbit/s)
- SAE J1939 (250kbps)
- SAE J1939 (500kbps)

Κατά την σύνδεση στον adapter, αρχικοποιούνται οι ρυθμίσεις και το πρωτόκολλο επικοινωνίας με τον εγκέφαλο του οχήματος και στην συνέχεια ξεκινά η καταγραφή δεδομένων. Τα κύρια συστατικά της εφαρμογής είναι ένα background service το οποίο πραγματοποιεί την σύνδεση με τον adapter μέσω Bluetooth. Το service αυτό έχει ως εργασία να κρατά την σύνδεση αυτή σταθερή και ζωντανή καθ' όλη την εκτέλεση της καταγραφής, ασχέτως εάν η Android συσκευή εκτελεί την εφαρμογή στο προσκήνιο ή στο παρασκήνιο ενώ εκτελεί και άλλες παράλληλες λειτουργίες όπως π.χ. τηλεφωνική κλήση, πλοήγηση μέσω GPS, αδράνεια, κ.α.

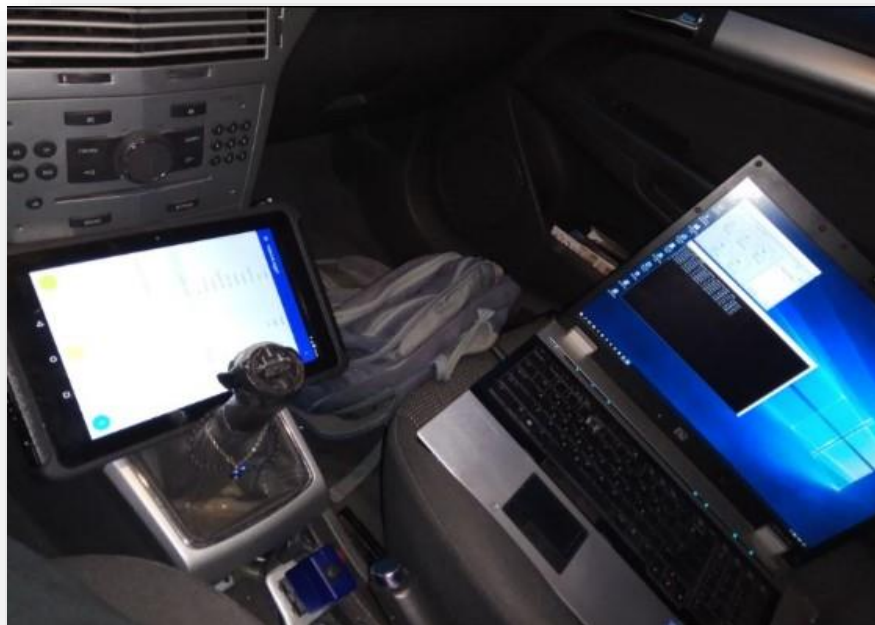
Το background service αυτό, εφόσον υπάρχει ενεργή σύνδεση Bluetooth, επικοινωνεί με τον adapter χρησιμοποιώντας σειριακή επικοινωνία κατά την οποία μέσω συνεχόμενου polling λαμβάνει απαντήσεις σε bytes τα οποία μορφοποιούνται και μεταφράζονται σε human readable data στην κύρια εφαρμογή. Το κύριο activity της εφαρμογής αναλαμβάνει την συγκέντρωση και την απεικόνιση των δεδομένων αυτών, καθώς και την δημιουργία των κατάλληλων δομών οι οποίες αποθηκεύονται και αποστέλλονται σε πραγματικό χρόνο (εφόσον υπάρχει ενεργή σύνδεση δεδομένων) σε μία Online real-time database.

Παράλληλα, μέσω του activity αυτού διατηρείται αναλυτικό log των συμβάντων της εφαρμογής, αποθηκεύονται τα δεδομένα και σε μορφή αρχείου .CSV, γίνεται διαχείριση στα στοιχεία του συνδεδεμένου οχήματος, ρυθμίζονται οι κύριες λειτουργίες της εφαρμογής, κ.α. Ταυτόχρονα με το data logging που αναφέρεται παραπάνω, η εφαρμογή καταγράφει διαρκώς και το στίγμα της τοποθεσίας που λαμβάνει από τον δορυφόρο ή από την κυψέλη που είναι συνδεδεμένη η host συσκευή όπως επίσης και δεδομένα από τα αισθητήρια όργανα του accelerometer και του gyroscope για περαιτέρω αποθήκευση και επεξεργασία με σκοπό την εξαγωγή συμπερασμάτων όπως π.χ. το ποσοστό ανωμαλίας των δρόμων, κ.λπ.[34]

Συνεπώς, αποθηκεύονται περιοδικά σε μία online database χιλιάδες δεδομένα οχήματος τα οποία μπορούν να χρησιμοποιηθούν για να εξαχθούν χρήσιμα συμπεράσματα, έχοντας διαθέσιμες τιμές ταχύτητας, στροφών ανά λεπτό, κατανάλωσης, θερμοκρασίας, γεωγραφικού στίγματος, κ.λπ. Καλό είναι να αναφερθεί πως μπορεί να εφαρμοστεί και η διαδικασία road snapping, όπου στο τέλος κάθε καταγεγραμμένη διαδρομής τα raw geo-data της εφαρμογής μπορούν να μετατραπούν σε πραγματικά γεωγραφικά δεδομένα οδών και δρόμων με αποτέλεσμα την εύκολη και σωστή αναζήτηση και απεικόνιση των δεδομένων αυτών.

Πέρα από το περιβάλλον ανάπτυξης, ένα από τα πιο σημαντικά κομμάτια της ανάπτυξης ήταν οι δοκιμές και η αποσφαλμάτωση της εφαρμογής, όπου λόγω της φύσης της συγκεκριμένης υλοποίησης, αποτέλεσε ένα δύσκολο εγχείρημα. Πιο συγκεκριμένα, χρειάστηκε ένας simulator έτσι ώστε να καταστεί δυνατό να δοκιμαστεί η αντίδραση της εφαρμογής σε διάφορες βασικές εντολές, χωρίς να είναι απαραίτητη η χρήση του Bluetooth dongle το οποίο επιτρέπει την σειριακή σύνδεση μέσω Bluetooth ανάμεσα στον υπολογιστή που εκτελούσε τον simulator και στην δοκιμαστική κινητή συσκευή. Παρόλα αυτά, ο simulator που χρησιμοποιήθηκε έχει πολύ περιορισμένες δυνατότητες και δεν μπόρεσε να ανταποκριθεί επαρκώς στις ανάγκες της εφαρμογής, ενώ η συμπεριφορά του αποδείχθηκε πως διαφέρει λίγο σε σχέση με τις πραγματικές συνθήκες που επικρατούν κατά την διάρκεια ενός δρομολογίου με πραγματικό όχημα. Συνεπώς, θα πρέπει να θεωρείται βέβαιη η ανάγκη ανάπτυξης, αποσφαλμάτωσης και δοκιμών μέσα στον χώρο ενός αυτοκινήτου όπως φαίνεται και στην παρακάτω πραγματική φωτογραφία.

Κατά την διάρκεια της υλοποίησης, παρουσιάστηκαν αρκετές προκλήσεις και δυσκολίες, οι οποίες θα πρέπει να ληφθούν υπόψη από όποιον θελήσει να επιχειρήσει κάτι ανάλογο στο μέλλον, έτσι ώστε να στηθεί το κατάλληλο περιβάλλον για ανάπτυξη και debugging του λογισμικού.



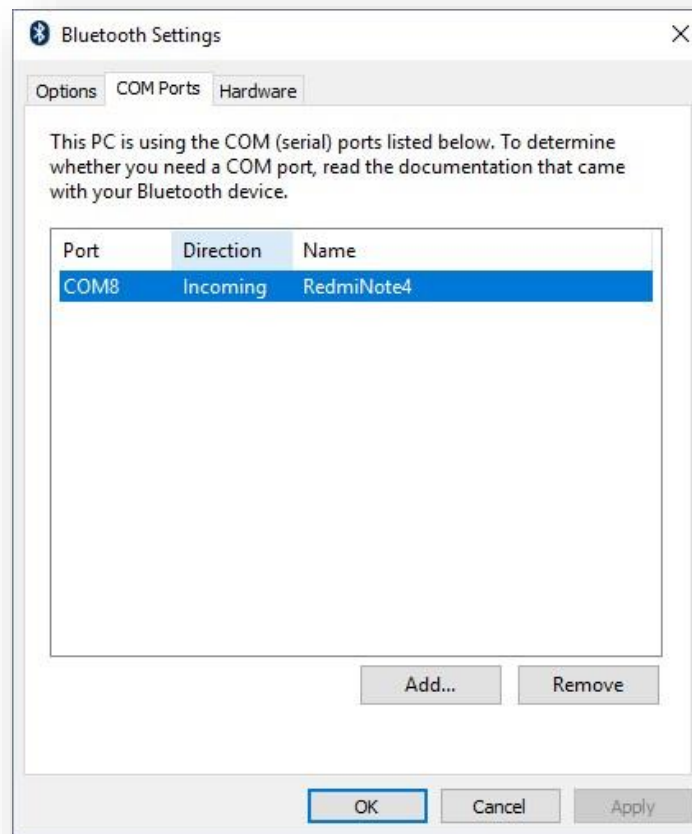
Εικόνα 15: Εικόνα καμπίνας του αυτοκινήτου την ώρα των δοκιμών

2.1.2 ODB Simulator

Η εφαρμογή OBDSim αποτελεί ίσως την μοναδική δωρεάν αξιόπιστη software λύση για την προσομοίωση του εγκεφάλου ενός οχήματος έτσι ώστε να καταστεί δυνατή η δοκιμή του λογισμικού που αναπτύσσεται. Η εφαρμογή βρέθηκε μέσα από απλή έρευνα στο Διαδίκτυο στην διεύθυνση <https://icculus.org/obdgpslogger/obdsim.html> όπου και περιγράφεται πολύ συνοπτικά η λειτουργία της. Μέσω του simulator, είναι δυνατό να επιστραφούν οι τιμές της ταχύτητας, των στροφών ανά λεπτό, του ποσοστού της χρήσης του γκαζιού, της θερμοκρασίας του κινητήρα και του ποσοστού της εισαγωγής αέρα. Ο simulator απαντά ταχύτατα και ανταποκρίνεται στις βασικές εντολές αρχικοποίησης που απαιτούνται για την ομαλή σύνδεση με τον εγκέφαλο του οχήματος. Αρχικά, θα πρέπει το υπολογιστικό σύστημα στο οποίο θα εκτελείται ο simulator να είναι εφοδιασμένο με Bluetooth adapter είτε on-board είτε πρόσθετο USB. Ο adapter αυτός θα χρησιμοποιηθεί ως σειριακή θύρα για την διασύνδεση μεταξύ της εφαρμογής που θα εκτελείται στην κινητή συσκευή και του simulator. Συνεπώς, θα πρέπει να έχει ρυθμιστεί η εισερχόμενη θύρα επικοινωνιών στον Bluetooth adapter όπως παρουσιάζεται στην Εικόνα 16.[32]

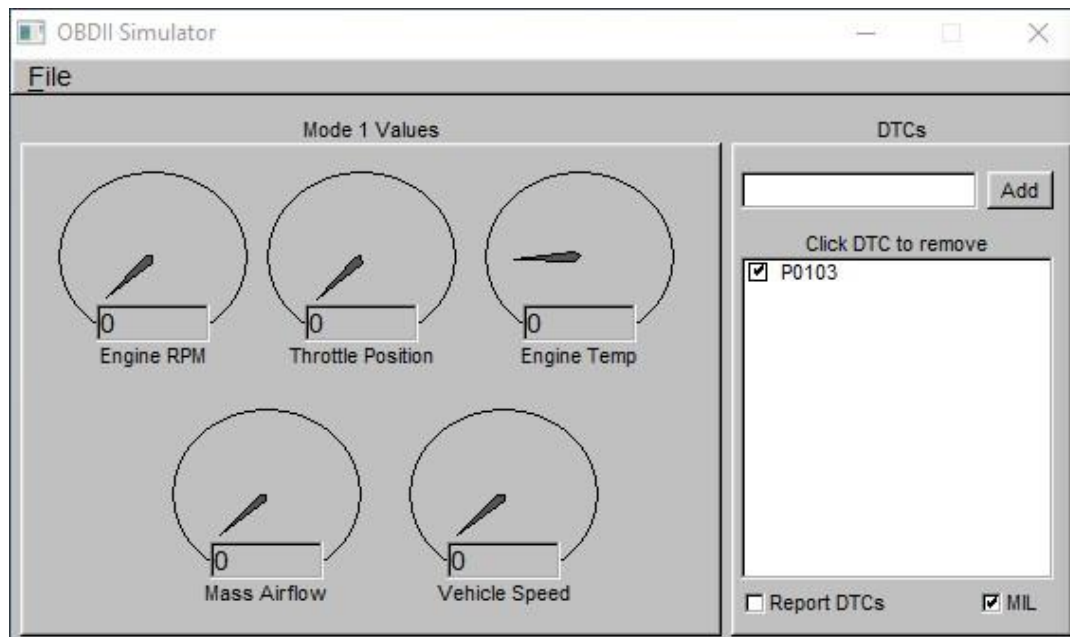
Έπειτα, κατά την εκτέλεση του simulator θα πρέπει να δοθεί ως παράμετρος η θύρα η οποία θα χρησιμοποιηθεί για τις I/O διεργασίες και στην οποία θα συνδεθεί και η κινητή συσκευή της εφαρμογής που αναπτύχθηκε. Υποθέτοντας πως ο simulator είναι εγκατεστημένος στο path C:\OBDSim\ και η εισερχόμενη θύρα Bluetooth επικοινωνιών είναι π.χ. η COM8, θα πρέπει να εκτελεστεί η ακόλουθη εντολή:

- C:\OBDSim\obdsim.exe -w COM8



Εικόνα 16: Bluetooth Connection Dialog

Έτσι θα εμφανιστούν στο desktop τα παράθυρα που εμφανίζονται στις παρακάτω εικόνες:



Εικόνα 17: Graphic User Interface simulator

Στην παραπάνω εικόνα παρουσιάζεται το GUI του simulator όπου σε εμφανή θέση διακρίνονται τα 5 αισθητήρια όργανα μέτρησης (Engine RPM, Throttle Position, Engine Temp, Ανάπτυξη εφαρμογής συλλογής δεδομένων από οχήματα μέσω θύρας OBD2, για λειτουργικό Android

Mass Airflow, Vehicle Speed) και πιο δεξιά εμφανίζεται η λίστα με τα Error Codes που μπορούν να αναπαραχθούν. Παρόλα αυτά, ο simulator δεν έχει την δυνατότητα να απαντήσει σε περισσότερες μετρήσεις, ούτε σε εντολές που επιστρέφουν πληροφορίες οχήματος όπως π.χ. το Vehicle Identification Number. Αντιστοίχως, στην εικόνα εμφανίζεται το output του simulator το οποίο ενημερώνει για τις εντολές αρχικοποίησης και για το πόσα queries εκτελούνται και απαντώνται ανά δευτερόλεπτο.

2.1.3 Βασικές εντολές επικοινωνίας.

Για να δημιουργηθεί η σύνδεση μεταξύ της εφαρμογής και του simulator (και κατά συνέπεια και του ECU του οχήματος) πρέπει να εκτελεστεί μία σειρά εντολών έτσι ώστε να αρχικοποιηθεί ο adapter. Ξεκινώντας, η εφαρμογή αποστέλλει σειριακά τις παρακάτω εντολές προς τον adapter:

- **ATZ** : Εντολή με την οποία ο adapter επιχειρεί reset και επιστρέφει ως απάντηση την ταυτότητα του ELM-USB dongle.
- **ATE0** : Εντολή με την οποία ο adapter ρυθμίζεται να μην επιστρέφει ως μέρος της απάντησης την εντολή την οποία δέχθηκε.
- **ATL0** : Εντολή με την οποία ο adapter ρυθμίζεται να μην επιστρέφει ως μέρος της απάντησης μία επιπλέον γραμμή ως Line Feed.
- **ATH0** : Εντολή με την οποία ο adapter ρυθμίζεται να μην επιστρέφει ως μέρος της απάντησης τα headers του frame π.χ. το ACKNOWLEDGE byte.
- **AT SP 0:** Εντολή με την οποία ο adapter επιλέγει αυτόματα το πρωτόκολλο επικοινωνίας με το όχημα.

Εφόσον ο adapter επιστρέφει τα σωστά αποτελέσματα (OK) σε κάθε εντολή, σημαίνει πως η αρχική ρύθμιση της σύνδεσης έχει γίνει με επιτυχία. Παρόλα αυτά, ακόμα δεν έχει ξεκινήσει η συνεχόμενη ανταλλαγή δεδομένων και δεν έχει επιτευχθεί η σύνδεση με τα αισθητήρια όργανα του οχήματος. Έπειτα από τις εντολές αρχικοποίησης και πριν από τις εντολές δεδομένων, θα πρέπει να υπάρξουν οι εντολές ανάκτησης γενικών πληροφοριών για το όχημα. Αυτές είναι οι ακόλουθες:

- **03** : Εντολή ανάκτησης Diagnostic Trouble Codes (DTC).
- **09 02** : Εντολή ανάκτησης Vehicle Identification Number (VIN).
- **01 21** : Εντολή ανάκτησης των χιλιομέτρων που έχουν διανυθεί με αναμμένη την λυχνία δυσλειτουργίας (Distance traveled with malfunction indicator lamp - MIL).

Τέλος, το βασικό κομμάτι της επικοινωνίας μεταξύ της εφαρμογής και του οχήματος αποτελείται από την επαναλαμβανόμενη εκτέλεση των παρακάτω εντολών:

- **01 0C** : Εντολή ανάκτησης των Στροφών ανά Λεπτό (Σ.α.Λ ή RPM) του κινητήρα
- **01 0D** : Εντολή ανάκτησης της ταχύτητας του οχήματος
- **01 05** : Εντολή ανάκτησης της θερμοκρασίας του κινητήρα
- **01 04** : Εντολή ανάκτησης του ποσοστού πλήρωσης του κινητήρα (Engine Load)
- **01 10** : Εντολή ανάκτησης του αισθητήρα μετρητή μάζας αέρα (MAF)
- **01 11** : Εντολή ανάκτησης του ποσοστού χρήσης του ποδομοχλού επιταχύνσεως (γκάζι)
- **01 06** : Εντολή ανάκτησης του αισθητήρα βενζίνης STFT1
- **01 07** : Εντολή ανάκτησης του αισθητήρα βενζίνης LTFT1
- **01 0F** : Εντολή ανάκτησης της θερμοκρασίας του εισαγόμενου αέρα (IAT)
- **01 0B** : Εντολή ανάκτησης του αισθητήρα απόλυτης πίεσης (MAP).

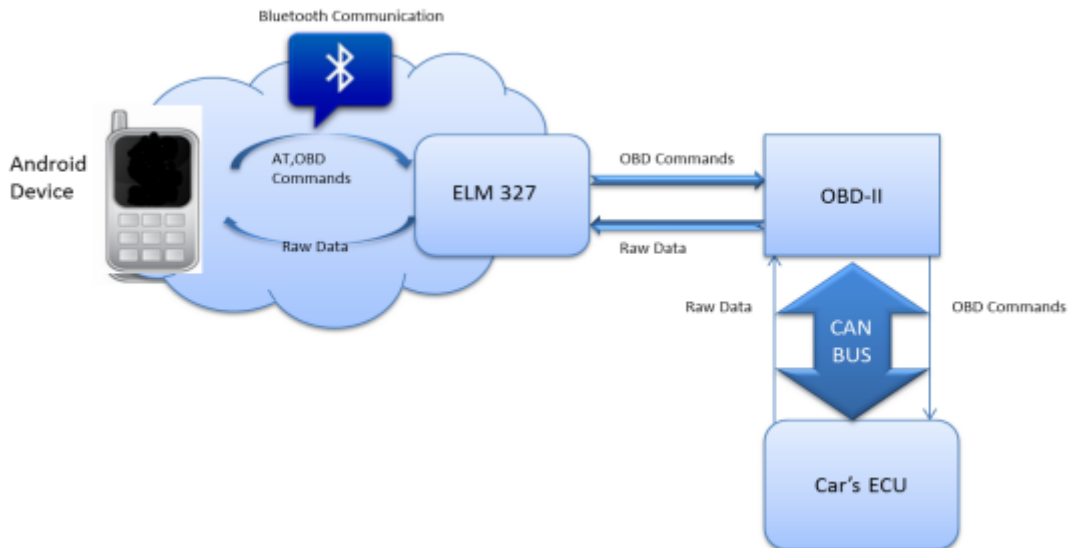
Κάθε όχημα δεν υποστηρίζει όλους τους τύπους εντολών ενώ μπορεί να μην έχει κάποιον από τους τύπους των αισθητήρων που απαιτούνται για να παράγουν απάντηση σε κάποια εντολή. Π.χ. από τα οχήματα που δοκιμάστηκαν, όσα οχήματα είχαν MAF sensor, δεν είχαν υποστήριξη για IAT και MAP sensors και το αντίστροφο.[20],[23],[24],[29]

2.2 Κύρια σημεία πηγαίου κώδικα

Σε αυτό το σημείο θα αναλυθούν τα κυριότερα σημεία του κώδικα της εφαρμογής καθώς και οι υπολογισμοί που πραγματοποιήθηκαν για να προκύψουν κάποια συγκεκριμένα αποτελέσματα.

2.2.1 Αρχιτεκτονική εφαρμογής

Όπως αναφέρθηκε και παραπάνω, η εφαρμογή αποτελείται από το front-end το οποίο αποτελεί το UI και από ένα background service το οποίο διαχειρίζεται τη σύνδεση μέσω Bluetooth με τον αντάπτορα. Το background service αρχικοποιεί τη σύνδεση με τη θύρα Bluetooth και έτσι ξεκινάει η σειριακή επικοινωνία. Για τη σειριακή επικοινωνία αποστέλλεται κάθε φορά η κατάλληλη εντολή η οποία είναι στιγμιότυπο ενός γενικού object το οποίο έχει ορισμένα βασικά χαρακτηριστικά και το οποίο κληρονομούν όλες οι εντολές που αποστέλλονται στον αντάπτορα. Παρακάτω παρουσιάζεται το block διάγραμμα στην παρακάτω εικόνα:



Εικόνα 18: Διάγραμμα κυκλώματος εφαρμογής

Η μητρική κλάση της εντολής η οποία ονομάζεται OBDCommand περιέχει τους κωδικούς όλων των mode του OBD πρωτοκόλλου αλλά και όλους τους κωδικούς των εντολών που ζητούνται από την εφαρμογή. Κάθε φορά που η εφαρμογή θέλει να στείλει μία εντολή, αρχικοποιεί ένα object τύπου OBDCommand μέσα στο οποίο περνάει ως παραμέτρους το input και το output buffer της σειριακής θύρας, τον χρόνο αναμονής μετά την αποστολή της κάθε εντολής, τον χρόνο που θα αποτελέσει το time-out, τον τύπο της εντολής και το πόσα bytes η εντολή περιμένει να λάβει ως απάντηση. Επιπλέον, ορίζεται ένα flag το οποίο σηματοδοτεί την εντολή εάν είναι εντολή αρχικοποίησης ή εντολή data.

```
public OBDCommand(OutputStream out, InputStream in, Long responseDelay, Long readTimeout, String data, int dataBytes, String responseMode, boolean isInitCommand)
```

Η κλάση της εντολής περιέχει την function που αποστέλλει τα data στη σειριακή αλλά και την function όπου διαβάζει το buffer της σειριακής θύρας και μέσα από κάποιες διεργασίες σχηματίζει την απάντηση για κάθε εντολή που αποστέλλεται. Για κάθε εντολή, ο αντάπτορας αποκρίνεται με δεδομένα τα οποία περιέχουν και τον τύπο της εντολής στην οποία αυτός απαντά,

έτσι είναι εύκολο για την εφαρμογή να ξεχωρίσει ποια απόκριση αντιστοιχεί σε ποια εντολή. Μετά από κάθε αποστολή εντολής, η εφαρμογή γνωρίζει τον τύπο και τον αριθμό των data σε bytes που θα πρέπει να διαβάσει και να σχηματίσει την απόκριση, οπότε γίνεται εύκολα αντιληπτό πως εάν δεν ληφθούν τα κατάλληλα δεδομένα, η function θα αποτύχει εσκεμμένα επιστρέφοντας NO DATA. Παράλληλα, καθώς έχει οριστεί το time-out για το διάβασμα της σειριακής μετά από κάθε εντολή, εάν αυτός ο χρόνος παρέλθει τότε πάλι η function επιστρέψει NO DATA.

```
new Thread(new Runnable() {
    @Override
    public void run() {
        try {
            out.write(commanddata.getBytes());
            out.flush();
            if (responseDelayInMs != null && responseDelayInMs > 0) {
                Thread.sleep(responseDelayInMs);
            }
            readSerialPort(readTimeout);
        } catch (IOException e) {
            Log.e(TAG, "IOException occurred while connecting to socket", e);
        } catch (InterruptedException e) {
            Log.e(TAG, "InterruptedException occurred while sleeping with responseDelayInMs", e);
        }
    }
}).start();
```

Όπως είναι φανερό από τον παραπάνω κώδικα, η εντολή γράφει τα data της στο buffer της bluetooth θύρας και αμέσως μετά περιμένει για όσο διάστημα της έχει οριστεί στη μεταβλητή responseDelay. Αμέσως μετά την ολοκλήρωσή του sleep, εκτελείται η εντολή διαβάσματος της σειριακής θύρας.

Κατά την έναρξη του διαβάσματος της θύρας ορίζεται ένα timestamp το οποίο θα έχει διάρκεια όσο το time-out που έχει οριστεί για κάθε εντολή. Έτσι ξεκινάει ένα while loop το οποίο εκτελείται μέχρι να παρέλθει ο χρόνος του time-out ή έως ότου η θύρα δεν έχει διαθέσιμα άλλα data για να διαβαστούν. Όσο η θύρα έχει data για ανάγνωση, αυτά διαβάζονται ανά χαρακτήρα και τοποθετούνται σε ένα τοπικό buffer, έως ότου εντοπιστεί ο χαρακτήρας που σηματοδοτεί την λήξη των δεδομένων. Αυτός ο χαρακτήρας για την επικοινωνία μέσω OBD είναι το σύμβολο > ή αλλιώς ο ASCII χαρακτήρας 62.

Μόλις ολοκληρωθεί η συλλογή των δεδομένων από τη θύρα, ελέγχεται ο αριθμός των bytes που ελήφθησαν και ο τύπος τους. Εάν το πλήθος των bytes είναι ίσο με το πλήθος που αναμένονται για κάθε εντολή, τότε η εφαρμογή προχωρά στο validation της απάντησης και ελέγχει εάν αυτή περιέχει στα δύο πρώτα bytes της το mode και την εντολή που ζητήθηκε. Εφόσον το validation πετύχει, τότε τα υπόλοιπα bytes μετατρέπονται από χαρακτήρες σε human readable data, αναλόγως με τον τύπο της κάθε εντολής.

Για παράδειγμα, μετά την αποστολή της εντολής 010D για το διάβασμα της ταχύτητας, ο αντίκτυπος απαντά με τα data 410D19>, όπου τα 2 πρώτα bytes αποτελούν την επιβεβαίωση πως ο εγκέφαλος αποκρίθηκε επιτυχώς (4) στο mode 1 και στην εντολή 0D, ενώ τα υπόλοιπα data μεταφράζονται σε αριθμό, ο οποίος σε αυτή την περίπτωση φανερώνει την ταχύτητα σε χιλιόμετρα / ώρα. Με τον ίδιο τρόπο λειτουργούν και οι άλλοι τύποι εντολών έτσι όπως έχουν αναφερθεί σε προηγούμενα κεφάλαια.

Κατά την αρχικοποίηση του αντικείμενου της εντολής, ορίζεται ως παράμετρος και ένα event το οποίο θα πυροδοτηθεί από την εντολή μόλις ολοκληρωθεί το διάβασμα της σειριακής. Αυτό σημαίνει πως σε περίπτωση επιτυχίας θα πυροδοτηθεί το event έχοντας ως παράμετρο τα data που διαβάστηκαν ενώ σε περίπτωση σφάλματος ή παρέλευσης του time-out, θα επιστραφεί το λεκτικό NO DATA.

```
public interface OnOBDCommandEventListener {
    void OBDResultEvent(boolean initCommandFlag, String commandType, String
data);
}
```

Ο διαχειριστής αυτού του event ανήκει στο background service και είναι αυτός που συντονίζει την περιοδική αποστολή των εντολών. Κατά την έναρξη της διαδρομής αρχικοποιείται μία FIFO λίστα εντολών, μέσα στην οποία προσθέτονται 20 περίπου εντολές. Από αυτές, οι 10 είναι εντολές αρχικοποίησης και οι υπόλοιπες είναι εντολές δεδομένων. Η εφαρμογή κρατά έναν index και ξεκινά από το πρώτο στοιχείο της λίστας εντολών, εκτελώντας την εντολή που βρίσκεται σε αυτή τη θέση. Μόλις η εντολή ολοκληρώσει το γράψιμο των data και το διάβασμα της σειριακής, πυροδοτεί το event που προαναφέρθηκε περνώντας ως παραμέτρους μέσα σε αυτό, τα data που διαβάστηκαν από τη σειριακή ενώ ορίζεται και ως παράμετρος ο τύπος της εντολής που εστάλη.

Μέσα στο event, εφόσον ληφθούν τα κατάλληλα data, η εφαρμογή απομακρύνει την εντολή που εστάλη και μειώνει το μέγεθος της λίστας κατά ένα, εκτελώντας πάντα την εντολή που βρίσκεται στην πρώτη θέση. Καθώς εκτελούνται οι εντολές αρχικοποίησης και πυροδοτείται πάντα το συγκεκριμένο event, αφαιρούνται μία-μία όλες οι εντολές αρχικοποίησης. Μόλις ο index φτάσει στην τελευταία εντολή αρχικοποίησης και καταλάβει ότι η επόμενη εντολή είναι εντολή data, σταματάει η διαγραφή των εντολών και ξεκινά μία επαναληπτική διαδικασία μέσα στην οποία εκτελούνται επαναλαμβανόμενα οι εντολές δεδομένων. Όσο πυροδοτείται το event και όσο η διαδρομή συνεχίζεται, οι εντολές εκτελούνται η μία μετά την άλλη. Έτσι μέσω ενός event-driven μοντέλου, οι εντολές εκτελούνται, διαβάζονται τα αποτελέσματά τους και ολοκληρώνεται ο κύκλος.

```
if (cmdList.get(0).isInitCommand()) {
    cmdList.remove(0);
}
cmdList.get(0).runOBDCommand()
```

Κατά την πυροδότηση του event και εφόσον έχουν εκτελεστεί όλες οι εντολές αρχικοποίησης, οι απαντήσεις των εντολών δεδομένων καταχωρούνται μία-μία σε ένα global object τύπου VehicleTripDetail. Εφόσον η πυροδότηση του event δεν ήρθε από την τελευταία εντολή της λίστας, η επαναληπτική εκτέλεση συνεχίζεται. Εφόσον ληφθεί απάντηση και για την τελευταία εντολή της λίστας, τότε ολοκληρώνονται οι εγγραφές στο παραπάνω object VehicleTripDetail και υπολογίζονται διάφορες ιδιότητες για όλους τους τύπους μετρήσεων που περιλαμβάνονται μέσα σε αυτή την περίοδο, ιδιότητες όπως η στιγμιαία κατανάλωση, το x, y, z από το accelerometer, το γεωγραφικό στίγμα της συσκευής, τις τιμές του γυροσκοπίου της συσκευής κ.λπ.

Μόλις ολοκληρωθεί ο υπολογισμός των ιδιοτήτων αυτών και εφόσον το object έχει valid τιμές, η εγγραφή τύπου VehicleTripDetail αποστέλλεται στη real-time database της Firebase. Παράλληλα, το record γράφεται και σε ένα αρχείο .CSV εφόσον βέβαια αυτή η επιλογή έχει ενεργοποιηθεί από τις ρυθμίσεις.

2.2.2 Υπολογισμός κατανάλωσης καυσίμου

Ο ακριβής υπολογισμός της κατανάλωσης του καυσίμου ενός οχήματος δεν είναι εύκολος. Κάθε όχημα είναι διαφορετικό, χρησιμοποιεί διάφορους τύπους καυσίμου ενώ διαθέτει διαφορετικούς αισθητήρες. Ο OBD adapter διαθέτει υποστήριξη για την εντολή που επιστρέφει την κατανάλωση του καυσίμου, όμως τα οχήματα που μπορούν να αποκριθούν σε αυτή την εντολή είναι ελάχιστα. Σχεδόν κανένα όχημα δεν μπορεί να αποδώσει απευθείας απάντηση που να σχετίζεται με την κατανάλωση, οπότε θα πρέπει αυτή να υπολογιστεί με την χρήση άλλων μεθόδων και δεδομένων.[21], [28]

Τα περισσότερα οχήματα τα οποία συνδέθηκαν με την εφαρμογή της συγκεκριμένης εργασίας (Opel, Smart) διέθεταν αισθητήρα MAF. Μέσω του αισθητήρα MAF και κάποιων άλλων παραδοχών, μπορεί να γίνει υπολογισμός για τον δείκτη **Km / Liter** καυσίμου. Έπειτα, έχοντας αυτή την πληροφορία για κάθε εγγραφή, μπορεί να γίνει ο υπολογισμός της τιμής **Liters / 100 Km** και, συνεπώς, να υπολογιστεί ο μ.ό. αυτής της τιμής στο πλήθος όλων των εγγραφών.

Για να πραγματοποιηθεί ο υπολογισμός της τιμής **Km / Liter** απαιτούνται τιμές που προέρχονται από δύο αισθητήρια όργανα του οχήματος: του MAF και της ταχύτητας που κινείται το όχημα. Έχοντας αυτές τις δύο τιμές, μπορεί να γίνει κατά προσέγγιση ο υπολογισμός της στιγμιαίας κατανάλωσης του καυσίμου. Γι' αυτό απαιτούνται μερικές τιμές ακόμα, όπως η πυκνότητα του καυσίμου και η σταθερά της αναλογίας Αέρα / Καυσίμου η οποία για κάθε τύπο καυσίμου θεωρείται σταθερή.

Κατά την λειτουργία του κινητήρα, ο εγκέφαλος (Engine Control Unit) του οχήματος μέσω των αισθητήριων οργάνων του, προσπαθεί να κρατήσει την αναλογία αυτή κοντά στον ιδανικό αριθμό των 14,7 γραμμάρια αέρα για κάθε γραμμάριο καυσίμου (για τους κινητήρες βενζίνης) οπότε αυτή η τιμή θεωρείται δεδομένη. Δεδομένη επίσης μπορεί να θεωρείται και η πυκνότητα του κάθε τύπου καυσίμου η οποία για την αμόλυβδη βενζίνη υπολογίζεται περίπου στα 725 kg / m³ (0,725-0,780 kg/l). Παρόλα αυτά, αναλόγως με την ποιότητα και την θερμοκρασία του καυσίμου μαζί με άλλους εξωγενείς παράγοντες, η πυκνότητα αυτή ενδεχομένως να αλλάζει.

Έχοντας ως δεδομένα όλα τα παραπάνω, η στιγμιαία τιμή της σχέσης **Km / Liter** μπορεί να υπολογιστεί από την παρακάτω σχέση:

- **Km / Liter** = Speed / (MAF * 3600 / Air Fuel Ratio / Fuel Density).

Έχοντας αυτή την τιμή, η εφαρμογή μπορεί να υπολογίσει την τιμή **Liter / 100 Km** εφαρμόζοντας τον παρακάτω τύπο:

- **Liters / 100 km** = 1.0 / km / Liter * 1000

Εφόσον κάθε μέτρηση περιέχει την παραπάνω πληροφορία, είναι εύκολο να γίνει ο υπολογισμός της μέσης κατανάλωσης σε όλο το πλήθος των εγγραφών κάθε διαδρομής. Υπήρξαν όμως και οχήματα που χρησιμοποίησαν την εφαρμογή (Fiat Punto, Fiat Panta) τα οποία, είτε δεν ανταποκρίνονται στην αντίστοιχη εντολή, είτε δεν διαθέτουν αισθητήρα MAF. Σε αυτή την περίπτωση, θα πρέπει να υπολογιστεί η τιμή του MAF κατά προσέγγιση μέσα από άλλα αισθητήρια όργανα και μετέπειτα να ακολουθηθεί η παραπάνω διαδικασία.

Συνήθως, όσα οχήματα δεν διαθέτουν MAF, ανταποκρίνονται στις εντολές MAP (Manifold Absolute Pressure) και IAT (Intake Air Temperature σε kelvin), μέσα από τις οποίες μπορεί να υπολογιστεί η τιμή IMAP, με δεδομένο πως υπάρχει διαθέσιμη πάντα η τιμή των στροφών ανά λεπτό ή αλλιώς RPM:

- **IMAP** = RPM * MAP / (IAT + 273)

Έπειτα, μπορεί να υπολογιστεί κατά προσέγγιση η τιμή του MAF έχοντας ως δεδομένα την Ideal Gas Constant R (8.314472 J / mol.K - Συγκεκριμένη Σταθερά του καυσίμου), τη MM (28.9644 g/mol - Μέση μοριακή μάζα αέρα), τη VE (Ογκομετρική Απόδοση κινητήρα ≈ 80%) και την χωρητικότητα του κινητήρα σε λίτρα (π.χ. 1,4):

- **MAF** = (IMAP/120)*(VE/100)*(ED)*(MM)/(R)

Για μεγαλύτερη ακρίβεια, θα πρέπει η ογκομετρική απόδοση του κινητήρα (VE) να οριστεί με ειδικά τεστ του οχήματος κάτω από συγκεκριμένες συνθήκες και στις ίδιες διαδρομές. Δίχως αυτά, η απόδοση του κινητήρα σε φυσιολογικές συνθήκες υπολογίζεται πως κυμαίνεται μεταξύ του 70%-90%, οπότε μια απόδοση της τάξης του 80% μπορεί να θεωρηθεί δεδομένη. Μετά και τον υπολογισμό του MAF μέσω αυτής της μεθόδου, μπορεί να ακολουθηθεί η μέθοδος που περιγράφεται παραπάνω για τον υπολογισμό της στιγμιαίας κατανάλωσης και κατά συνέπεια της μέσης κατανάλωσης για όλη την διαδρομή.[27]

Η εφαρμογή κατά την διάρκεια καταγραφής της διαδρομής υπολογίζει όλες τις τιμές που πρέπει, έτσι ώστε στο τέλος να εξαχθούν οι μέσοι όροι οι οποίοι μπορούν να βοηθήσουν σε στατιστικές αναλύσεις. Πιο συγκεκριμένα, σε κάθε καταγραφή υπολογίζεται η στιγμιαία κατανάλωση σε L / 100 km και σε L / Hour με τους τρόπους που αναφέρθηκαν παραπάνω. Ταυτόχρονα, υπολογίζεται η μέγιστη τιμή των μετρήσεων ταχύτητας και RPM ενώ κρατείται και το άθροισμα των τιμών τους για την εξαγωγή του Μ.Ο. στο τέλος της διαδρομής. Παράλληλα, κρατείται πάντα η τελευταία εγγραφή, έτσι ώστε σε κάθε επόμενη να μπορεί να γίνει ο υπολογισμός της απόστασης που διανύθηκε. Όταν η συχνότητα των καταγραφών είναι μεγάλη, η εξαγωγή της απόστασης μέσα από την μέση ταχύτητα δυο διαδοχικών καταγραφών μέσα στο διάστημα του χρόνου μπορεί να γίνει με μεγάλη ακρίβεια όπως φαίνεται παρακάτω:

```

float avgSpeed = (instantSpeed + Float.parseFloat(lastTripDetail.getSpeed())) /
                2;
float timeElapsed = Math.abs(detail.getTimestamp().getTime() -
                             lastTripDetail.getTimestamp().getTime()) / 1000f;
float instantDistance = (avgSpeed * timeElapsed) / 3.6f;
currentTrip.setTotalDistance(currentTrip.getTotalDistance() + instantDistance);

```

Έτσι υπολογίζοντας την μέση ταχύτητα δύο διαδοχικών καταγραφών $S = (Speed_n + Speed_{n-1})$ σε km / hour και τον χρόνο που διανύθηκε $T = ABS (Timestamp_n - Timestamp_{n-1})$ σε seconds, προκύπτει η απόσταση που διανύθηκε μεταξύ αυτών των δύο καταγραφών με την πράξη $D = (S * T) / 3,6$. Τέλος, προσθέτοντας όλες αυτές τις καταγραφές, προκύπτει η συνολική απόσταση της διαδρομής.

Όταν η διαδρομή ολοκληρώνεται και κατά συνέπεια τερματίζεται και η καταγραφή από την εφαρμογή, υπολογίζονται και αποθηκεύονται όλες οι μέσες τιμές στην εγγραφή της διαδρομής, πριν αυτή σταλεί ως ολοκληρωμένη στη real-time βάση δεδομένων. Πιο συγκεκριμένα, οι μέσες τιμές της ταχύτητας, των RPM, της κατανάλωσης Liters ανά 100 km και ανά hour υπολογίζονται με βάση το πλήθος όλων των καταγραφών. Έπειτα, βάσει της απόστασης που διανύθηκε και της μέσης κατανάλωσης, υπολογίζονται τα λίτρα που χρησιμοποιήθηκαν για την συγκεκριμένη διαδρομή.



```

20171021164744
----- avgLitPer100: 10.4071060095192
----- avgLitPerHour: 1.587614356087262
----- avgRPM: 1190.329956054687
----- avgSpeed: 12.47000026702886
----- deviceId: "355233088050362"
----- fuelConsumedPer100: 0.225884556770324
----- fuelConsumedPerHour: 0.1195501163601872
----- isTripSnapped: 1
----- maxRPM: 3235
----- maxSpeed: 65
----- milDistance: "4073"
----- totalDistance: 2569.790972222222
----- totalDuration: 10.67000007629392
----- tripEnd: "2017-10-21T16:58:2"
----- tripID: "20171021164744"
----- tripStart: "2017-10-21T16:47:4"
----- vehicle
----- engineCapacity: 1400
----- fuelType: 1
----- vehicleMake: 200
----- vehicleMakeDescr: "Opel"
----- vehicleModel: "ASTRA H"
----- vehicleModelType: "GTC"
----- vehicleRegYear: 2007
----- vehicleIN: "W0L0AHL087515200"
----- vehicleTC: "P0136;P0141;P0141

```

Εικόνα 19: Συγκριτικά αποτελέσματα μεταξύ του Board Computer του Opel Astra H και των αντίστοιχων καταγραφών μέσω της εφαρμογής

Κρίνοντας από τα αποτελέσματα, η εφαρμογή υπολογίζει με μεγάλη ακρίβεια τις μετρήσεις της διαδρομής, συγκρίνοντας τις καταγραφές της με τις καταγραφές που εμφανίζει το board computer του οχήματος δοκιμών, όπως φαίνεται στην παραπάνω εικόνα.

Όπως γίνεται αντιληπτό, παρατηρείται πως ο υπολογιστής του οχήματος κατέγραψε για μια δεδομένη διαδρομή πως διανύθηκαν 2,6 χιλιόμετρα, με μέση ταχύτητα 12 km. / hour, καταναλώνοντας 0,2 λίτρα καυσίμου και με μέση κατανάλωση 10,3 liters / 100 km.

Αντίστοιχα, στις καταγραφές της εφαρμογής και συγκεκριμένα όπως φαίνεται μέσα από την διαχείριση της Firebase, υπολογίστηκε για την συγκεκριμένη διαδρομή η απόσταση που διανύθηκε στα 2569 μέτρα, με μέση ταχύτητα 12,47 km. / hour, καταναλώνοντας περίπου 0,23 λίτρα καυσίμου με μέση κατανάλωση περίπου στα 10,41 liters / 100 km.

2.3 Επεξήγηση λειτουργίας του VehicleLogger

Έχοντας ξεκαθαρίσει το κομμάτι που αφορά τον τρόπο υλοποίησης του κώδικα, η συνέχεια περιέχει μία συνολική αναφορά στον τρόπο που λειτουργεί η εφαρμογή από την μεριά του χρήστη και τον τρόπο που πρέπει να ρυθμιστεί στην συσκευή του ώστε να λειτουργήσει σωστά.

2.3.1 Αρχικοποίηση σύνδεσης

Όπως αναφέρθηκε παραπάνω, το πρώτο βήμα που θα πρέπει να πραγματοποιηθεί έτσι ώστε να ξεκινήσει η καταγραφή μέσω της εφαρμογής, είναι η επιλογή του Bluetooth adapter. Για να γίνει αυτό, θα πρέπει πρώτα να έχει γίνει σύζευξη της συσκευής με τον adapter μέσω του λειτουργικού συστήματος.

Αρχικά, θα πρέπει ο adapter να έχει συνδεθεί επάνω στην OBD θύρα του οχήματος. Σε κάθε όχημα η θύρα εντοπίζεται σε διαφορετική τοποθεσία, συνήθως γύρω από το τιμόνι ή κάπου στο ταμπλό. Η θύρα αυτή δεν είναι ποτέ σε εμφανές σημείο και κρύβεται τις περισσότερες φορές κάτω από κάποιο πλαστικό ή κάλυμμα το οποίο θα πρέπει να ξεβιδωθεί ή να ξεκουμπωθεί.



Εικόνα 20: Θέση θύρας OBD σε Opel Astra H GTC

Για να βρεθεί το σημείο στο οποίο βρίσκεται η θύρα στα διάφορα οχήματα, ο ιστότοπος <https://www.outilsobdfacile.com/location-plug-connector-obd.php> προσφέρει χρήσιμη βοήθεια, προβάλλοντας φωτογραφίες για κάθε κατασκευαστή και τύπο οχήματος. Στην περίπτωση της

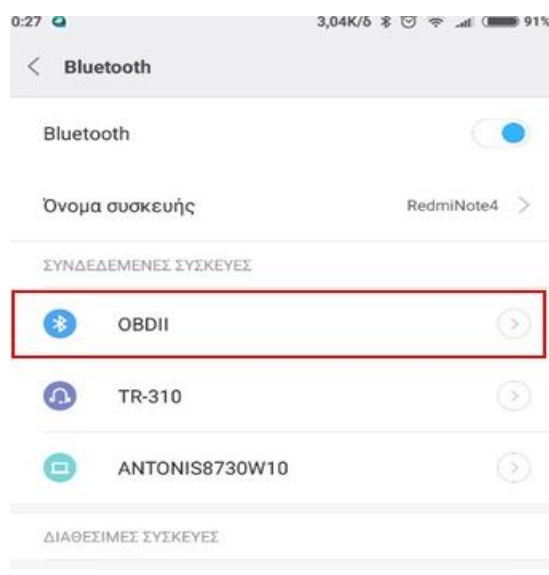
παρούσας εργασίας, το βασικό όχημα δοκιμών και ανάπτυξης της εφαρμογής είναι ένα Opel Astra H GTC (2007) 1400cc το οποίο έχει τοποθετημένη την OBD θύρα σε ένα κάλυμμα κάτω από το χειρόφρενο όπως φαίνεται στην εικόνα που βρέθηκε από τον παραπάνω ιστότοπο.

Ο adapter εφαρμόζει με συγκεκριμένη κατεύθυνση πολύ εύκολα στην θύρα. Κατά την σύνδεσή του, ανάβει ένα κόκκινο Led πάνω σε αυτόν, το οποίο σηματοδοτεί πως υπάρχει τροφοδοσία ρεύματος. Παρόλα αυτά, καμία συσκευή δεν μπορεί να ξεκινήσει να τον χρησιμοποιεί ως καταγραφικό εάν το όχημα δεν τεθεί σε λειτουργία ή έστω το κλειδί δεν τοποθετηθεί στην μίζα στην κατάσταση check engine. Παρακάτω στην εικόνα παρουσιάζεται ο OBD2 αντάπτορας σε διαδικασία λειτουργίας.



Εικόνα 21: Ο OBD2 adapter σε λειτουργία

Εφόσον υλοποιηθούν τα παραπάνω, η συσκευή πρέπει να πραγματοποιήσει σύζευξη με τον adapter σε επίπεδο λειτουργικού συστήματος. Συνεπώς, μέσα από τις ρυθμίσεις Bluetooth του λειτουργικού, θα πρέπει να γίνει ανίχνευση της ELM συσκευής η οποία συνήθως εμφανίζεται με την ονομασία OBDII. Μόλις επιλεγθεί η σύζευξη, θα πρέπει να εισαχθεί το default password της συσκευής (συνήθως 1234) και να ολοκληρωθεί η διαδικασία όπως φαίνεται στην παρακάτω εικόνα.[25]



Εικόνα 23: Bluetooth device selection menu

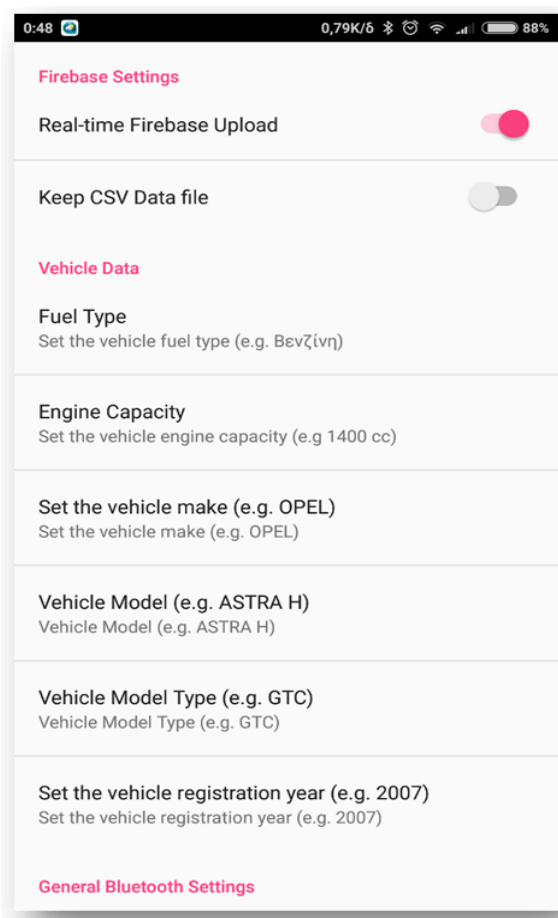
Επιστρέφοντας στην εφαρμογή της παρούσας εργασίας, πατώντας το πλήκτρο με το σήμα του Bluetooth εμφανίζονται όλες οι συσκευές που βρίσκονται σε σύζευξη με την συσκευή. Εκεί ο χρήστης επιλέγει την συσκευή OBDII και πλέον είναι έτοιμος να συνδεθεί με τον adapter και να ξεκινήσει την διαδρομή του πατώντας το πλήκτρο start trip.



Εικόνα 22: Μενού επιλογής συσκευής στην εφαρμογή

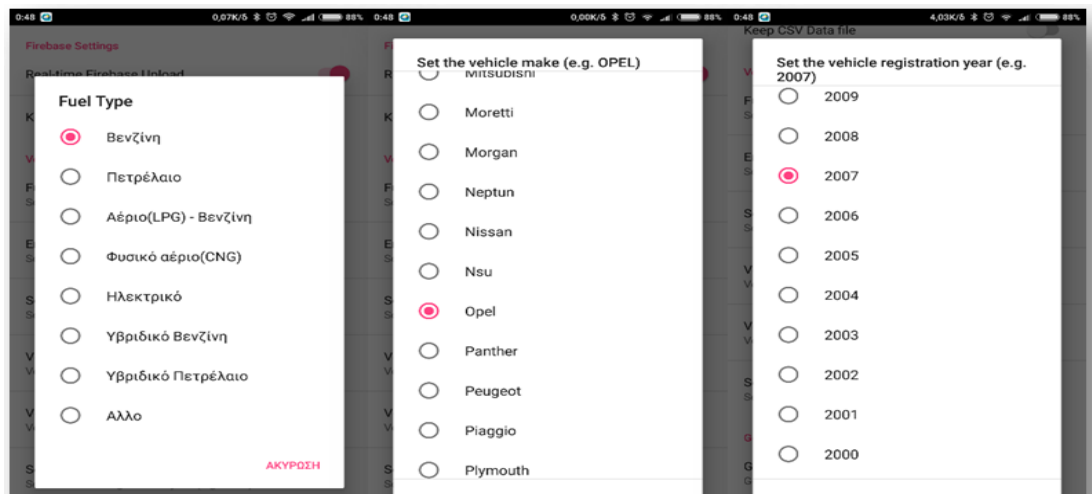
Πριν πατηθεί το πλήκτρο start trip, θα πρέπει να έχουν ρυθμιστεί οι κατάλληλες παράμετροι της εφαρμογής ενώ προαπαιτούμενη είναι και η δήλωση του οχήματος στην εφαρμογή. Οπότε θα πρέπει να πατηθεί το πλήκτρο των ρυθμίσεων στο επάνω δεξιά μενού για να ξεκινήσει σωστά η διαδικασία.

Ξεκινώντας, διακρίνεται ο τομέας Firebase Settings. Με την 1η επιλογή σε αυτό τον τομέα ο χρήστης επιλέγει εάν θέλει οι εγγραφές του να αποστέλλονται στην real-time database του Firebase framework της Google. Κάτι ανάλογο επιχειρεί και η 2η επιλογή αυτού του τομέα, αφού προσφέρει την δυνατότητα στον χρήστη να αποθηκεύσει και να εξάγει τα data του σε αρχεία CSV όπως φαίνεται στην εικόνα.



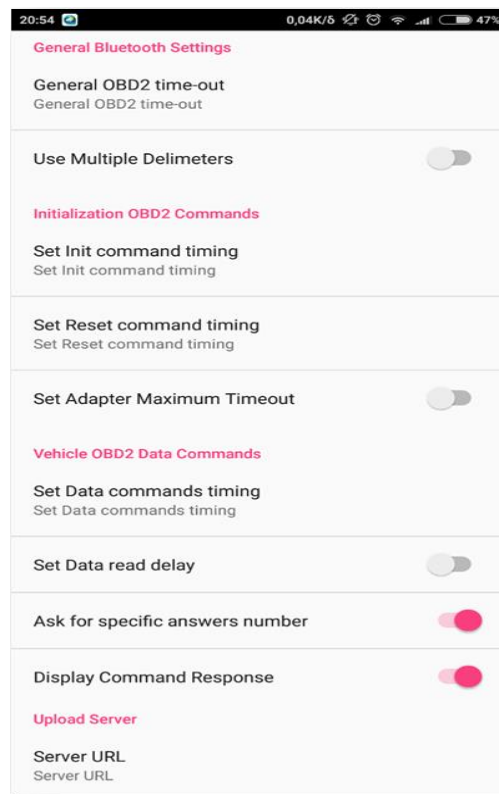
Εικόνα 24: Preferences Menu – Top Side

Στον τομέα Vehicle Data ορίζεται ο τύπος του οχήματος με το οποίο θα συνδεθεί η εφαρμογή. Στο πεδίο Fuel Type ορίζεται ο τύπος του καυσίμου που χρησιμοποιεί το όχημα ενώ στο Engine Capacity ορίζεται η χωρητικότητα του κινητήρα σε κυβικά εκατοστά π.χ. 1.400. Στο πεδίο Vehicle Make ο χρήστης επιλέγει την μάρκα του οχήματός του μέσα από μία προκαθορισμένη λίστα με κατασκευαστές ενώ στο πεδίο Vehicle Model πληκτρολογεί το μοντέλο του οχήματός του (π.χ. ASTRA H). Τέλος, στο πεδίο Vehicle Model Type, ο χρήστης μπορεί να ορίσει ένα ιδιαίτερο χαρακτηριστικό του τύπου του οχήματος του π.χ. GTC. Τέλος, στο πεδίο Vehicle Registration Year ορίζεται το έτος της πρώτης κυκλοφορίας του οχήματος. Όλα τα παραπάνω ορίζονται για στατιστικούς λόγους αλλά και για να μπορούν να πραγματοποιηθούν σωστά οι υπολογισμοί της κατανάλωσης του καυσίμου για κάθε διαδρομή. Πιο συγκεκριμένα, ο τύπος του καυσίμου παίζει ρόλο για τον υπολογισμό της κατανάλωσης σε μία συγκεκριμένη πυκνότητα ενώ τα κυβικά του οχήματος επηρεάζουν τον υπολογισμό της κατανάλωσης σε συγκεκριμένα οχήματα τα οποία δεν διαθέτουν τους κατάλληλους αισθητήρες (MAF). Παρακάτω ακολουθεί τριπλό screenshot των μενού.



Εικόνα 26: Screenshot για το μενού δήλωσης πληροφοριών οχήματος

Συνεχίζοντας, οι επόμενες ρυθμίσεις στην εικόνα είναι πολύ σημαντικές για τη διασύνδεση μεταξύ της συσκευής και του OBD2 Dongle.



Εικόνα 25: Preferences Menu Down-Side

Με την επιλογή General OBD2 time-out ορίζεται ο χρόνος σε ms, μετά την πάροδο του οποίου η εφαρμογή θα επιστρέψει time-out. Αυτό πρακτικά σημαίνει πως αν οριστεί ο time-out χρόνος π.χ. στα 500ms, εάν ο αντάπτορας δεν απαντήσει μέσα σε αυτό το χρονικό διάστημα η εφαρμογή θα επιστρέψει για την συγκεκριμένη εντολή την τιμή NaN. Προεπιλεγμένος χρόνος time-out για την εφαρμογή έχουν ορισθεί τα 250ms.

Παρακάτω, η ρύθμιση Set Init command timing ορίζει τον χρόνο που θα χρειαστεί ο OBD2 Adapter να απαντήσει μετά από μία εντολή αρχικοποίησης. Πιο συγκεκριμένα, μετά από κάθε εντολή η εφαρμογή πρέπει να δίνει λίγο χρόνο στον adapter έτσι ώστε αυτός να καταφέρει να απαντήσει. Αυτό πρακτικά μπορεί να επιτευχθεί μέσω ενός sleep-timer αμέσως μετά την εντολή που θα σταλεί προς την θύρα Bluetooth. Η εντολή αρχικοποίησης είναι μία από τις εντολές που έχουν την πιο αργή απόκριση, οπότε για αυτήν ορίζεται ένα ξεχωριστό delay το οποίο έχει προεπιλεγμένο χρόνο για την εφαρμογή στα 2500ms. Το ίδιο ισχύει και για την εντολή επανεκκίνησης (Reset command) του adapter. Προεπιλεγμένος χρόνος αναμονής μετά την αποστολή αυτής της εντολής είναι τα 1000ms.

Με τη ρύθμιση Set Adapter Maximum time-out ορίζεται το μέγιστο time-out για την κάθε απόκριση του adapter σε επίπεδο hardware. Αυτό σημαίνει πως δύναται να αποσταλεί εντολή στον adapter να περιμένει μέχρι και 1000ms μέχρι να επιστρέψει μία απάντηση. Η παρακάτω ρύθμιση Set Data commands timing αφορά τον χρόνο που θα παρέρχεται μετά την αποστολή κάθε εντολής η οποία περιμένει ως απόκριση raw data, δηλαδή δεδομένα μετρήσεων. Ο προκαθορισμένος χρόνος της εφαρμογής για αυτή τη ρύθμιση είναι τα 50ms, χρόνος όμως ο οποίος μπορεί να μειωθεί αρκετά και να φτάσει μέχρι και τα 20ms.

Ο διακόπτης της ρύθμισης Set Data read delay χρησιμοποιείται κυρίως σε παλαιότερες συσκευές οι οποίες δεν μπορούν να ανταποκριθούν σε τόσο χαμηλό χρονισμό εντολών. Με αυτό τον τρόπο η εφαρμογή προσθέτει μία έξτρα καθυστέρηση κατά την ανάγνωση έτσι ώστε να καταφέρει να συνθέσει σωστά μία ολοκληρωμένη απόκριση.

Έπειτα, ο διακόπτης της ρύθμισης Ask for specific answers number μπορεί να ορίσει το εάν ο αντάπτορας θα απαντήσει απευθείας μετά την άφιξη της πρώτης απάντησης για κάθε εντολή. Πιο συγκεκριμένα, κάποιες εντολές που ζητούνται από τον αντάπτορα μπορεί να συνθέτονται μέσα από πολλά αισθητήρια όργανα. Αν αυτή η επιλογή είναι απενεργοποιημένη, τότε ο αντάπτορας περιμένει να καταφτάσουν όλες οι απαντήσεις από τον εγκέφαλο του οχήματος και μετά να αποστείλει την τελική απάντηση προς την εφαρμογή. Σε περίπτωση όμως που επιλογή αυτή είναι ενεργοποιημένη, ο αντάπτορας επιστρέφει την απάντηση απευθείας μετά την απόκριση του πρώτου αισθητήριου, γεγονός που μπορεί να μειώσει κατά πολύ το χρόνο απόκρισης του αντάπτορα.

Τέλος, με τη ρύθμιση Server URL ορίζεται ο ιστότοπος στον οποίο η εφαρμογή θα ανεβάζει τα δεδομένα σε περίπτωση που έχει επιλεγθεί η δυνατότητα καταγραφής δεδομένων και σε .CSV αρχεία. Αφού έχουν γίνει οι απαραίτητες ρυθμίσεις, η εφαρμογή είναι έτοιμη να συνδεθεί στον αντάπτορα και να ξεκινήσει να αποστέλλει τις εντολές.

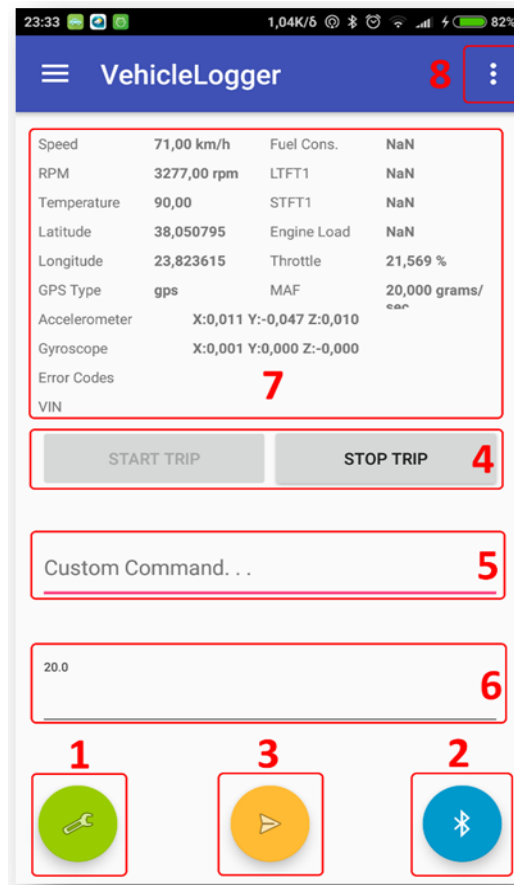
2.3.2 Πλοήγηση στο GUI

Η εφαρμογή που αναπτύχθηκε έχει ως σκοπό την real-time συλλογή δεδομένων από τον εγκέφαλο του οχήματος στο οποίο θα συνδέεται μέσω της OBD θύρας, σε συνδυασμό με τους αισθητήρες της κινητής συσκευής στην οποία θα εκτελείται, όπως το Accelerometer, το Gyroscope και τον αισθητήρα του GPS. Παράλληλα, η εφαρμογή παρουσιάζει τα δεδομένα που ανακτά σε ένα UI έτσι ώστε να ενημερώνει τον χρήστη για την πορεία της καταγραφής, ενώ παράλληλα αποθηκεύει τις καταγραφές σε μία online real-time database. Επιπλέον, η εφαρμογή κρατά πλήρες log των καταγραφών και των συμβάντων ενώ αποθηκεύει όλες τις καταγραφές και σε επιπλέον αρχεία CSV σε περίπτωση που αυτά χρειαστούν.

Ο σχεδιασμός της εφαρμογής είναι σχετικά απλός. Η κύρια εφαρμογή περιέχει το UI και τα controls τα οποία ελέγχουν την καταγραφή, παρουσιάζει τα δεδομένα που διαβάζονται ενώ περιέχει και τις φόρμες από όπου ο χρήστης μπορεί να επεξεργαστεί τις διάφορες ρυθμίσεις και να εισάγει τις λεπτομέρειες του οχήματος επάνω στο οποίο συνδέεται η εφαρμογή.

Επιπλέον, η εφαρμογή δημιουργεί και συνδέεται με μία υπηρεσία η οποία εκτελείται ως background task και είναι υπεύθυνη για την σύνδεση με τον OBD adapter μέσω της θύρας Bluetooth. Μέσω αυτής της υπηρεσίας η εφαρμογή καταφέρνει να επικοινωνεί με τον αντάπτορα ακόμα και όταν η εφαρμογή λειτουργεί στο παρασκήνιο ή όταν το τηλέφωνο είναι κλειδωμένο ή σε sleep mode.

Στην εικόνα που ακολουθεί έχουν επιλεγθεί τα κύρια σημεία του μενού με κόκκινο χρώμα και αρίθμηση. Παρακάτω περιγράφεται μια συνοπτική επεξήγηση αυτών.



Εικόνα 27: Κύριο μενού εφαρμογής

- Πλήκτρο 1: Με αυτό το πλήκτρο ο χρήστης μπορεί να επιλέξει τον OBD Adapter από μια λίστα με paired Bluetooth συσκευές.
- Πλήκτρο 2: Με το πλήκτρο αυτό ο χρήστης μπορεί να ενεργοποιήσει/απενεργοποιήσει μια νέα Bluetooth σύνδεση με τον adapter που επιλέχθηκε στο 1ο βήμα.
- Πλήκτρο 3: Με το πλήκτρο αυτό ο χρήστης μπορεί να αποστείλει μια custom εντολή προς τον adapter.
- Πλήκτρα 4: Τα πλήκτρα Start / Stop Trip ενεργοποιούνται εφόσον υπάρχει μια ενεργή σύνδεση Bluetooth με κάποια συσκευή. Με το πλήκτρο Start Trip, ξεκινάει μια νέα διαδρομή κατά την οποία αρχικοποιούνται όλες οι εγγραφές και ξεκινάει ένας νέος κύκλος αρχικοποίησης του ELM Adapter. Με το πλήκτρο End Trip, σταματά η καταγραφή εντολών και τερματίζεται η διαδρομή.
- Πεδίο 5: Στο πεδίο αυτό ο χρήστης μπορεί να πληκτρολογήσει μια custom εντολή (π.χ. ATI) και να την αποστείλει πατώντας το πλήκτρο στο 3ο βήμα.
- Πεδίο 6: Στο πεδίο αυτό αναγράφονται όλα τα Raw responses του adapter κάθε φορά που του απαντά σε μια προκαθορισμένη ή custom εντολή.
- Πεδίο 7: Σε όλα τα πεδία αυτής της ομάδας στοιχείων αναγράφονται οι πραγματικές τιμές που διαβάζονται περιοδικά από τον εγκέφαλο του οχήματος μέσω του adapter.

2.4 Αποθήκευση σε βάση δεδομένων.

Για την αποστολή των μετρήσεων επιλέχθηκε η real time database του Firebase framework της Google. Η βάση δεδομένων αυτή είναι μια NoSQL βάση η οποία δέχεται αντικείμενα σε JSON format και τα αποθηκεύει στο cloud. Προσφέρει υπηρεσίες συγχρονισμού των δεδομένων σε όλους τους συνδεδεμένους χρήστες, built-in RESTful API για όλα τα data που αποθηκεύονται, authentication και πολλά άλλα.[26],[30]

Στο πλαίσιο αυτής της εργασίας η Firebase εξυπηρετεί το γεγονός πως παρέχει real-time υποστήριξη, όπως επίσης και σε περίπτωση που ο client είναι offline τα δεδομένα κρατούνται τοπικά. Συνεπώς, ακόμα και όταν η συσκευή στη οποία εκτελείται η εφαρμογή βρεθεί χωρίς παροχή Διαδικτύου, τα data κρατούνται τοπικά και μόλις βρεθεί διαθέσιμη σύνδεση Internet τότε ανεβαίνουν όλα στην Firebase. Το integration με το framework είναι αρκετά απλό και δεν απαιτούνται περίπλοκα μπλοκ εντολών για να επικοινωνήσει η εφαρμογή με την Firebase.

Κατά την λειτουργία της εφαρμογής, όταν ο χρήστης πατάει το πλήκτρο START TRIP, αυτόματα μαζί με όλες τις άλλες εργασίες, δημιουργείται και μια global οντότητα τύπου VehicleTrip η οποία συγκρατεί γενικές πληροφορίες για την διαδρομή, όπως την ώρα έναρξης, την οντότητα του οχήματος η οποία περιέχει όλες τις πληροφορίες που προαναφέρθηκαν όπως π.χ. Τύπος Καυσίμου, Κυβικά, Κατασκευαστής, Τύπος, κ.α., το IMEI της συσκευής στην οποία εκτελείται η εφαρμογή, κ.α. Κατά την αποστολή εντολών αρχικοποίησης στον OBD adapter, η οντότητα VehicleTrip ανανεώνεται με τις απαντήσεις του εγκεφάλου του οχήματος και αποθηκεύει πληροφορίες όπως το Vehicle Identification Number (VIN), τα Trouble Codes, την απόσταση που έχει διανυθεί με αναμμένη την λυχνία Check-Engine της μηχανής, κ.α.

```
private void uploadTripToFirebase(VehicleTrip trip) {  
    Map<String, Object> postValues = trip.toMap();  
    FirebaseDatabase().child("VehicleTrips").child(new  
        SimpleDateFormat("yyyyMMddHHmmss").format(trip.getTripStart())) .updateChildren(postValues);  
}
```

Με την παραπάνω εντολή, η εφαρμογή αποστέλλει την οντότητα τύπου VehicleTrip στην Firebase η οποία την αποθηκεύει κάτω από την parent οντότητα VehicleTrips με κλειδί το timestamp της δεδομένης χρονικής στιγμής σε μορφή YYYYMMDDHHmmss. Μόλις ο χρήστης πατήσει το πλήκτρο END TRIP, η εφαρμογή αποθηκεύει το timestamp της λήξης και την υπολογισμένη κατανάλωση καυσίμου.

Στην εικόνα διακρίνεται η οντότητα του οχήματος (OPEL, κ.λπ.), η κατανάλωση του καυσίμου της διαδρομής (11,86L / 100 km), η απόσταση που έχει διανυθεί με αναμμένη την λυχνία σφάλματος (3883 χλμ.), η έναρξη και η λήξη της διαδρομής, το VIN του οχήματος (W0L0AHL0875152002) και τις βλάβες που είναι ενεργές την στιγμή της έναρξης της διαδρομής (P0136, P0141).

[vehiclelogger-19018](#) > [VehicleTrips](#) > [20171011174733](#)

```

20171011174733
-----
avgLitPer100: 11.865855244863772
deviceID: "862844033425586"
milDistance: "3883"
totalDistance: 14614.4
totalDuration: 3347.4
tripEnd: "2017-10-11T18:12:16"
tripID: "20171011174733"
tripStart: "2017-10-11T17:47:33"
- vehicle
-----
engineCapacity: 1400
fuelType: 1
vehicleMake: 208
vehicleMakeDescr: "Opel"
vehicleModel: "ASTRA H"
vehicleModelType: "GTC"
vehicleRegYear: 2007
vehicleIN: "W0L0AHL087515200"
vehicleTC: "P0136;P0141;P0141

```

Εικόνα 28: Firebase object

Όπως προαναφέρθηκε, η Firebase προσφέρει αυτόματα RESTful API για τα data που αποθηκεύονται σε αυτή, οπότε προσθέτοντας την κατάληξη .json π.χ. στο URL <https://vehiclelogger-19018.firebaseio.com/VehicleTrips/20171011174733.json> η Firebase θα επιστρέψει σε μορφή JSON string το object της διαδρομής απεικονίζει η προηγούμενη εικόνα:

```

{
  "avgLitPer100": 11.865855244863772,
  "deviceID": "862844033425586",
  "isTripSnapped": 1,
  "milDistance": "3883",
  "totalDistance": 14614.4,
  "totalDuration": 3347.4,
  "tripEnd": "2017-10-11T18:12:16",
  "tripID": "20171011174733",
  "tripStart": "2017-10-11T17:47:33",
  "vehicle": {
    "engineCapacity": 1400,
    "fuelType": 1,
    "vehicleMake": 208,
    "vehicleMakeDescr": "Opel",
    "vehicleModel": "ASTRA H",
    "vehicleModelType": "GTC",
    "vehicleRegYear": 2007
  }
},

```

```
"vehicleIN": "W0L0AHL0875152002",  
"vehicleTC": "P0136;P0141;P0141;"  
}
```

Μετά από τις εντολές αρχικοποίησης του OBD adapter και κατά την διάρκεια της περιοδικής αποστολής των εντολών δεδομένων, δημιουργείται μια εγγραφή τύπου VehicleTripDetail η οποία και αποστέλλεται στην Firebase κάθε φορά που ολοκληρώνεται ο κύκλος.

```
{  
  "accTot": "0.5118554638352477",  
  "accX": "-0.4724893569946289",  
  "accY": "-0.16260585188865662",  
  "accZ": "-0.1109466552734375",  
  "alt": "220.1",  
  "engineLoad": "2.745098114013672 %",  
  "gyrX": "-0.11618660390377045",  
  "gyrY": "-0.03197402134537697",  
  "gyrZ": "-0.11618660390377045",  
  "intakeAirTemp": 36.0,  
  "lat": "38.04671833333334",  
  "lng": "23.815441666666667",  
  "lper100": 4.2223786066150595,  
  "ltft1": "4.6875",  
  "maf": "1.0",  
  "provider": "gps",  
  "rpm": "796.0",  
  "speed": "8",  
  "stft1": "1.5625",  
  "temp": "93 C",  
  "throttle": "3.529411792755127",  
  "timestamp": "11/10/2017 18:11:51",  
  "tripID": "20171011174733"  
}
```

Αντιστοίχως, παρουσιάζεται η εικόνα όπου φαίνεται το παραπάνω detail object.

```

1134
-----
accTot: "0.511855463835247"
accX: "-0.472489356994628"
accY: "-0.1626058518886566"
accZ: "-0.110946655273437"
alt: "220.1"
engineLoad: "2.745098114013672"
gyrX: "-0.1161866039037704"
gyrY: "-0.0319740213453769"
gyrZ: "-0.1161866039037704"
intakeAirTemp: 36
lat: "38.04671833333333"
lng: "23.81544166666666"
lper100: 4.222378606615059
ltft1: "4.6875"
maf: "1.0"
provider: "gps"
rpm: "796.0"
speed: "8"
stft1: "1.5625"
temp: "93 C"
throttle: "3.52941179275512"
timestamp: "11/10/2017 18:11:5"
tripID: "20171011174733"

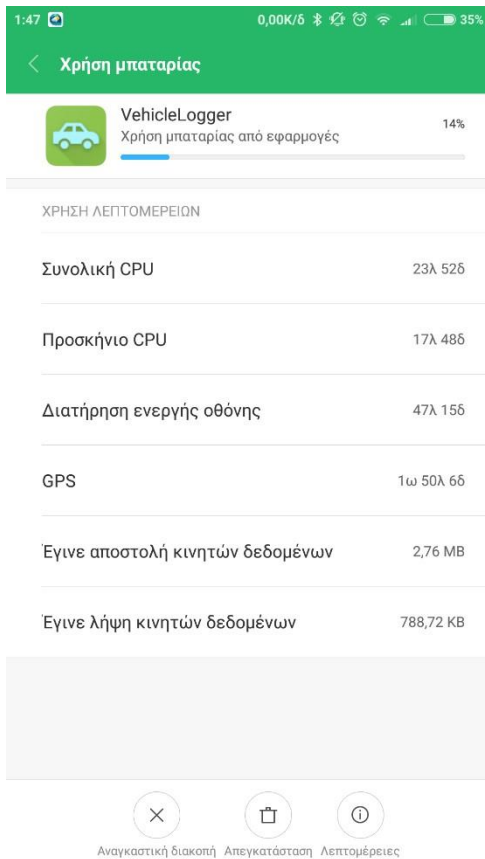
```

Εικόνα 29: Firebase object 2

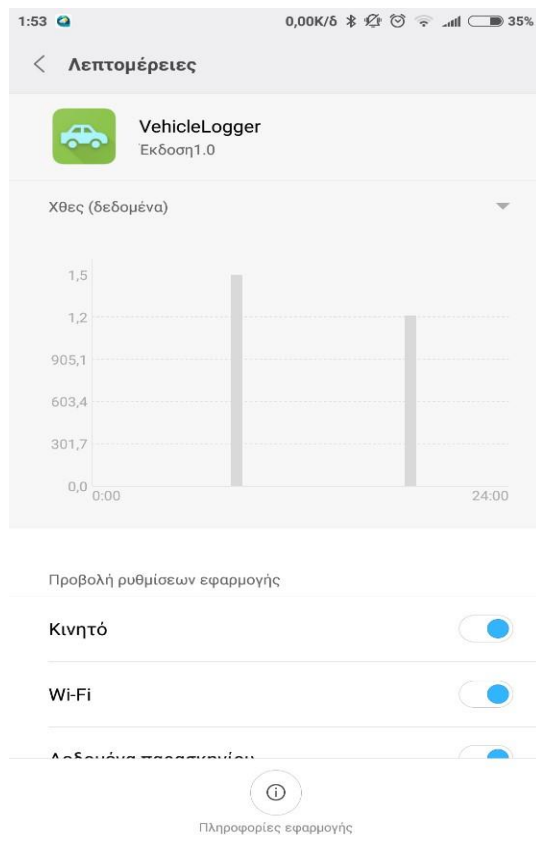
Στην παραπάνω οντότητα, διακρίνονται οι τιμές X,Y,Z του accelerometer και του gyroscope, οι τιμές latitude, longitude, altitude του GPS της συσκευής και τα δεδομένα των μετρήσεων του οχήματος για το δεδομένο timestamp όπως το πόσο γεμάτος είναι ο κινητήρας (2.745%), η θερμοκρασία του εισαγόμενου αέρα (36 C), ο δείκτης MAF (1.0), οι στροφές του κινητήρα (796.0 RPM), η ταχύτητα του οχήματος (8 km / h), η θερμοκρασία του κινητήρα (93 C), το ποσοστό χρήσης του πεντάλ επιτάχυνσης (3.529%) και οι δείκτες Short / Long Term Fuel Trim (1.5625% / 4.6875%).

Υποθέτοντας πως διαβάζονται 9 εντολές κάθε 800ms, τότε σε μία διαδρομή 20 λεπτών θα αποθηκευτούν περίπου 20 min. * 60 sec. * 1000 ms = 1200000ms / 800ms = 1500 records. Εάν κάθε εγγραφή είναι ≈ 650 bytes τότε για μια διαδρομή 20 λεπτών θα χρειαστεί upload περίπου 975Kb δεδομένων (χωρίς βέβαια να υπολογίζεται και το download των πακέτων για SYN / ACK). Συνεπώς, τα δεδομένα έχουν αρκετά μικρό footprint.

Η εφαρμογή σε μια ημέρα πραγματοποίησε καταγραφή δύο ίδιων διαδρομών (άφιξη / επιστροφή) περιλαμβάνοντας τα ίδια τελικά σημεία. Η πρωινή διαδρομή ήταν διάρκειας 26 λεπτών ενώ η απογευματινή ήταν διάρκειας 20 λεπτών. Όπως φαίνεται από την εικόνα, η συνολική κατανάλωση δεδομένων κινητής τηλεφωνίας ήταν περίπου 2,7mb.



Εικόνα 30: Screenshot Application Battery Usage



Εικόνα 31: Screenshot trip data traffic

Αντιθέτως με την παραπάνω περίπτωση, η κατανάλωση της μπαταρίας της συσκευής δεν μπορεί να χαρακτηριστεί μικρή. Η εφαρμογή VehicleLogger χαρακτηρίστηκε από το σύστημα ως η πιο ενεργοβόρα, καθώς κατανάλωσε το 14% της μπαταρίας. Βέβαια, αν αναλογιστεί κανείς πως η εφαρμογή διατήρησε την οθόνη σε λειτουργία για 47 λεπτά, ενώ χρησιμοποίησε το GPS για 1 ώρα και 50 λεπτά, παράλληλα με την αποστολή δεδομένων μέσω 4G και την χρήση της θύρας Bluetooth, τότε θα καταλήξει στο συμπέρασμα πως τα αποτελέσματα που φαίνονται στις παραπάνω εικόνες θα μπορούσε να ήταν αρκετά χειρότερα.

ΚΕΦΑΛΑΙΟ 3

3.1 Προβλήματα Υλοποίησης

Κατά την ανάπτυξη και τη δοκιμή της εφαρμογής παρουσιάστηκαν κάποιες δυσλειτουργίες, η αιτία των οποίων δεν ήταν πολύ εύκολο να γίνει αντιληπτή. Οι δυσλειτουργίες αυτές είχαν ως αποτέλεσμα την καθυστέρηση της εκτέλεσης των εντολών που είχε με την σειρά της ως συνέπεια την περιπλοκή μεταξύ των εντολών και των αποκρίσεων τους. Ο αρχικός προκαθορισμένος χρόνος αναμονής μετά την αποστολή μιας εντολής ήταν ορισμένος στα 250ms, αλλά και πάλι, μετά από αρκετή ώρα χρήσης της εφαρμογής, ο αντίκτυπος άρχισε να απαντά όλο και πιο αργά.

Καθώς είχαν δοκιμαστεί αντίστοιχες ή παραπλήσιες εμπορικές εφαρμογές του είδους, σε κάποιες από αυτές παρατηρήθηκε ότι ο χρόνος αναμονής μεταξύ των εντολών ήταν ελάχιστος με αποτέλεσμα οι τιμές από τον εγκέφαλο του οχήματος να ανανεώνονται σχεδόν real-time. Κάτι τέτοιο δεν συνέβαινε με την εφαρμογή αυτής της εργασίας και η λύση δεν ήταν προφανής ενώ το documentation του OBD δεν έκανε ξεκάθαρο για ποιο λόγο οι ερωταπαντήσεις δεν εκτελούνται άμεσα. Έτσι, εκτελέστηκε ο προσομοιωτής OBDSim με συγκεκριμένες παραμέτρους.[31]

- **C:\OBDSim\obdsim.exe -w com8 -q obdlogfile.txt**

Στο manual του προσομοιωτή αναφέρει πως με την παράμετρο **-q** όλες οι επικοινωνίες προς και από τον simulator θα καταγράφονται σε ένα log αρχείο στον υπολογιστή που εκτελείται ο προσομοιωτής. Έπειτα, εγκαθιστώντας μία εμπορική εφαρμογή παραπλήσια με αυτή της παρούσας εργασίας στην Android συσκευή και συνδέοντάς την μέσω της incoming Bluetooth θύρας στον υπολογιστή και κατά συνέπεια στον simulator, παρατηρήθηκε πως αποστέλλεται μια επιπλέον εντολή η οποία δεν αναφερόταν εμφανώς στο documentation. Αυτή η εντολή είναι η εντολή αρχικοποίησης **ATTA2** η οποία μεταφράζεται ως **Timing Advantage 2** και η οποία ενεργοποιεί την πολύ μεγάλη ταχύτητα απόκρισης του OBD adapter.[22]

Μετά την ενσωμάτωση αυτής της εντολής, ο adapter και ο simulator δέχονται εντολές και απαντούν σε αυτές μέσα σε μερικές δεκάδες milliseconds καθιστώντας έτσι την επικοινωνία πάρα πολύ γρήγορη και σταθερή.

Ένα πρόσθετο ζήτημα που παρουσιάστηκε, ήταν το πώς θα είναι διαχειρίσιμες οι μη υποστηριζόμενες εντολές από τα διάφορα οχήματα. Κάθε όχημα δεν υποστηρίζει όλες τις εντολές, π.χ. το Opel Astra H 2007 υποστηρίζει τον αισθητήρα MAF και όχι τους αισθητήρες MAP, IAT ενώ στο Fiat Punto ELX 2004 συμβαίνει το αντίστροφο. Έτσι, για όσες εντολές δεν υποστηρίζονταν, η εφαρμογή περίμενε υποχρεωτικά να προκύψει το time-out, γεγονός το οποίο οδηγούσε σε καθυστερήσεις μεταξύ των εντολών και χαμένου χρόνου που μπορούσε να φτάσει ακόμα και τα 1500ms μέσα σε μια περίοδο.

Καθώς δεν είναι κομψό ο χρήστης να ενεργοποιεί / απενεργοποιεί εντολές κατά βούληση, χρησιμοποιήθηκε η εντολή αρχικοποίησης **0100** η οποία επιστρέφει 4 bytes τα οποία απεικονίζουν σε hex μορφή όλα τα bitwise encoded PIDs που υποστηρίζει το όχημα στο οποίο είναι συνδεδεμένος ο OBD adapter. Μετατρέποντας αυτά τα bytes σε δυαδική μορφή, το αποτέλεσμα είναι ένας πίνακας με 0 (μη υποστηριζόμενες) και 1 (υποστηριζόμενες) εντολές. Για παράδειγμα, εάν η απόκριση του οχήματος στην παραπάνω εντολή είναι η **BE1FA813**, τότε αυτή μεταφράζεται ως εξής από την εικόνα:

Hexadecimal	B				E				1				F				A				8				1				3					
Binary	1	0	1	1	1	1	1	0	0	0	0	1	1	1	1	1	1	0	1	0	1	0	0	0	0	0	0	1	0	0	0	1	1	1
Supported?	Yes	No	Yes	Yes	Yes	Yes	Yes	No	No	No	No	Yes	Yes	Yes	Yes	Yes	Yes	No	Yes	No	Yes	No	No	No	No	No	No	Yes	No	No	Yes	Yes		
PID number	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F	10	11	12	13	14	15	16	17	18	19	1A	1B	1C	1D	1E	1F	20		

Εικόνα 32: Πίνακας PID με υποστηριζόμενες εντολές οχήματος

Γεγονός που σημαίνει πως το όχημα υποστηρίζει μόνο τις εντολές 01, 03, 04, 05, 06, 07, 0C, 0D, 0E, 0F, 10, 11, 13, 15, 1C, 1F και 20. Έτσι, κατά την αρχικοποίηση ανακτώνται οι υποστηριζόμενες εντολές και, μέσα από μια διαδικασία decoding, αφαιρούνται από την λίστα εντολών που αναφέρθηκε παραπάνω, όλες οι μη υποστηριζόμενες εντολές.

```

if (!TextUtils.isEmpty(data)) {
    String[] pids = data.split(";");
    if (pids.length == 8) {
        for (int i = 0; i < pids.length; i++) {
            char[] frame = pids[i].toCharArray();
            for (int j = 0; j < frame.length; j++) {
                if (frame[j] == '0') {
                    String pid = OBDCommand.PIDs[(i * 4) + j];

                    // remove unsupported pids
                    for (int k = cmdList.size()-1; k >= 0; k--) {
                        if (cmdList.get(k).getResponseMode() ==
                            pid) {
                            writeToLog("Disabling " +
                                pid + " command...");
                            cmdList.remove(k);
                        }
                    }
                }
            }
        }
    }
}

```

Αυτό έχει ως αποτέλεσμα:

```

13:21:33.010 Disabling 04 command...
13:21:33.011 Disabling 06 command...
13:21:33.011 Disabling 07 command...
13:21:33.011 Disabling 0B command...
13:21:33.012 Disabling 0F command...

```

Συνεπώς, περιοδικά αποστέλλονται μόνο οι εντολές που υποστηρίζονται, χωρίς να προκύπτουν καθυστερήσεις ή κολλήματα με αποτέλεσμα η εφαρμογή να καταγράφει 9 εντολές μέσα σε μια περίοδο διάρκειας σχεδόν 700ms όπως φαίνεται και στο παρακάτω απόσπασμα του log.

```

08:40:51.197 sendDataToBluetooth -> commandType: 0C - sendData: 010C1
08:40:51.262 readSerialPort -> commandType: 0C - actualData: 410C0C4C
08:40:51.267 sendDataToBluetooth -> commandType: 0D - sendData: 010D1
08:40:51.331 readSerialPort -> commandType: 0D - actualData: 410D00
08:40:51.337 sendDataToBluetooth -> commandType: 05 - sendData: 01051
08:40:51.402 readSerialPort -> commandType: 05 - actualData: 410586
08:40:51.408 sendDataToBluetooth -> commandType: 04 - sendData: 01041
08:40:51.473 readSerialPort -> commandType: 04 - actualData: 410406
08:40:51.479 sendDataToBluetooth -> commandType: 10 - sendData: 01101
08:40:51.544 readSerialPort -> commandType: 10 - actualData: 41100096
08:40:51.559 sendDataToBluetooth -> commandType: 11 - sendData: 01111
08:40:51.619 readSerialPort -> commandType: 11 - actualData: 411109
08:40:51.627 sendDataToBluetooth -> commandType: 06 - sendData: 01061
08:40:51.692 readSerialPort -> commandType: 06 - actualData: 41067F
08:40:51.700 sendDataToBluetooth -> commandType: 07 - sendData: 01071
08:40:51.787 readSerialPort -> commandType: 07 - actualData: 410785
08:40:51.797 sendDataToBluetooth -> commandType: 0F - sendData: 010F1
08:40:51.894 readSerialPort -> commandType: 0F - actualData: 410F41

```

Εκτελώντας την εφαρμογή με τον OBDSim (υποστήριξη έως 5 εντολές):

```
SimPort name: com8
Successfully initialised obdsim, entering main loop
Reset
Echo disabled
Linefeed disabled
Headers disabled
Adaptive Timing: 2
New Protocol: Automatic
10.021355 seconds. 63 samples, 63 queries. 6.29 s/s, 6.29 q/s
10.079632 seconds. 74 samples, 74 queries. 7.34 s/s, 7.34 q/s
10.021327 seconds. 68 samples, 68 queries. 6.79 s/s, 6.79 q/s
10.010944 seconds. 74 samples, 74 queries. 7.39 s/s, 7.39 q/s
10.022628 seconds. 82 samples, 82 queries. 8.18 s/s, 8.18 q/s
```

Παρατηρείται πως εκτελούνται σχεδόν 7 query/sec, γεγονός που καθιστά την εφαρμογή αρκετά κοντά στην real-time απεικόνιση των πραγματικών δεδομένων.

Ένας επιπλέον προβληματισμός παρουσιάστηκε όταν η εφαρμογή δοκιμάστηκε σε παλιότερη συσκευή τηλεφώνου με έκδοση Android 2.3. Σε αυτή την περίπτωση, το λειτουργικό σύστημα απενεργοποιούσε μετά από κάποια ώρα καταγραφής (≈ 15 min.) το Bluetooth socket και έτσι έσπαγε ο κύκλος της αποστολής των εντολών και η διαδρομή έμενε μετέωρη. Έπειτα από την λεπτομερή μελέτη των log και μέσα από έρευνα στα documents της Google αλλά και σε forums, αποδείχτηκε πως το πρόβλημα οφειλόταν σε κάποια buggy λογική του λειτουργικού συστήματος. Χωρίς να μπορεί να γίνει κάποια δραστική ενέργεια στην υλοποίηση για να αποφευχθεί το παραπάνω γεγονός, προστέθηκε μια λογική μέσω της οποίας ανιχνεύεται το ανωτέρω συμβάν κατά την διάρκεια του κύκλου και εφόσον προκύψει βίαιη απενεργοποίηση του socket, η εφαρμογή κλείνει την δομή του VehicleTrip και ξεκινά να δονεί την συσκευή χρησιμοποιώντας ένα ιδιαίτερο μοτίβο, έτσι ώστε ο χρήστης να αντιληφθεί πως κάτι δεν εξελίσσεται ομαλά. Καθώς η εφαρμογή δεν έχει δικαίωμα να εκκινήσει την Bluetooth σύνδεση αυτόματα χωρίς αλληλεπίδραση με τον χρήστη, αυτός μπορεί να εκκινήσει ξανά την καταγραφή της διαδρομής του μόλις αντιληφθεί την αποσύνδεση, χωρίς να χάσει πολύτιμα data.

3.2 Επισκόπηση και μελλοντική έρευνα.

Με την ολοκλήρωση της παρούσας διπλωματικής εργασίας έγιναν αντιληπτές οι δυνατότητες που μπορούν να προσφέρουν τέτοιου είδους εφαρμογές. Οι βελτιώσεις που μπορούν να γίνουν έχουν τρεις κύριους άξονες:

1. Το hardware που χρησιμοποιήθηκε για την υλοποίηση της εφαρμογής.
2. Το software που χρησιμοποιήθηκε για την υλοποίηση της εφαρμογής.
3. Τα data τα οποία συλλέχθηκαν από τα test.

Αναφέροντας το hardware, εννοείται πως με την περαιτέρω εξέλιξη της τεχνολογίας τα τυποποιημένα πρωτόκολλα επικοινωνίας θα βελτιώνονται με τον καιρό, με αποτέλεσμα όλο και περισσότερα επιμέρους συστήματα που υπακούουν στην ECU να είναι διαθέσιμα στον εκάστοτε ιδιοκτήτη - τεχνικό συντήρησης για προβολή ή επικοινωνία. Η υλοποίηση από πλευράς πτυχιακής με hardware του εμπορίου δεν είναι δεσμευτική για τον τελικό χρήστη καθώς σαφέστατα κάποιος μπορεί να υλοποιήσει την διαγνωστική συσκευή κυριολεκτικά εκ του μηδενός, κατασκευάζοντας την δική του πλακέτα με τα ανάλογα αισθητήρια. Ένας πολύ μεγάλος κλάδος που ανοίγει με το συγκεκριμένο θέμα είναι και αυτός της ασφάλειας, καθώς εφόσον η διεπαφή στο όχημα συνήθως βρίσκεται σε απόσταση το πολύ 2 μέτρων από το τιμόνι, θα πρέπει να βρεθεί και ο αντίστοιχος τρόπος ασφάλισης της εφαρμογής. Είναι αυτονόητο ότι ο οποιοσδήποτε κακόβουλος χρήστης που θα μπορούσε να είχε πρόσβαση στον χώρο της καμπίνας, μπορεί να προκαλέσει είτε ανεπανόρθωτη ζημιά στον εγκέφαλο του οχήματος είτε να προβεί σε κλοπή αυτού.

Στο κομμάτι που έχει να κάνει με τα IX, δυστυχώς, εξετάστηκαν πολύ λίγα μοντέλα (κυρίως του δημιουργού) και η συμπεριφορά αυτών. Πέραν της παλαιότητας των μοντέλων και τον περιορισμό στο ότι όλα χρησιμοποιούσαν καύσιμο βενζίνης, υπήρξε και ο περιορισμός του μεγέθους της πληροφορίας που παρείχε η κάθε ECU. Θα ήταν άξιο αναφοράς το λογισμικό να

προσαρμοστεί ανάλογα ώστε να μπορέσει να διαβάσει και νεότερους πετρελαιοκινητήρες ή ακόμα και υβριδικά οχήματα (αέριο ή ηλεκτρικά). Επιπλέον, το υπάρχον λογισμικό μπορεί να εμπλουτιστεί με διάφορες έξυπνες τεχνικές για να γίνει πιο προσιτό στον τελικό χρήστη. Μερικές από αυτές είναι η αυτόματη ανίχνευση (εφόσον είναι ενεργοποιημένος ο Bluetooth πομποδέκτης) του OBD dongle και η αυτόματη σύνδεση με αυτόν χωρίς την επίδραση του χρήστη καθώς και η αυτόματη έναρξη / λήξη της καταγραφής μιας διαδρομής εφόσον η εφαρμογή αντιληφθεί θετικές / μηδενικές τιμές RPM αντίστοιχα. Επιπλέον, αξιοποιώντας μία από τις πολλές επί πληρωμή online υπηρεσίες VIN decoding, η εφαρμογή μπορεί, βάσει του αναγνωρισμένου VIN του οχήματος στο οποίο συνδέεται, να ανακτήσει δεκάδες πληροφορίες οι οποίες αφορούν γενικές και τεχνικές προδιαγραφές του συγκεκριμένου οχήματος, να αναζητήσει σε βάσεις δεδομένων με οχήματα τα οποία έχουν αναφερθεί σε περιστατικά κλοπής και να ανακτήσει την τελευταία αναφορά του χιλιομετρικού οδομετρητή με βάσει τους τακτικούς ελέγχους ΚΤΕΟ, κ.λπ.

Πολύ σημαντικό θα ήταν επίσης κάποιος να μπορέσει να ασχοληθεί και με την μεταφορά του πηγαίου κώδικα σε επίπεδο άλλης πλατφόρμας. Η ανάπτυξη του σε λειτουργικό iOS για συσκευές Apple αλλά και Windows Phone είναι από μόνο του μία πρόκληση καθώς τόσο το κόστος του αναπτυξιακού αλλά και η μεταγλώττιση θα είναι μία πολύπλοκη διαδικασία. Οι αλγόριθμοι επικοινωνίας αλλά και μεταφοράς των δεδομένων θα έπρεπε πιθανόν να γραφούν και σε μεταγενέστερες γλώσσες προγραμματισμού. Ακόμα και η μεταφορά της σε σχεσιακή βάση δεδομένων σε κάποια άλλη πλατφόρμα όπως MySQL θα χρειαστεί περαιτέρω μελέτη έτσι ώστε τα δεδομένα να παραμείνουν ανέπαφα και να μην χαθούν τιμές.

Στο κομμάτι της βελτίωσης των συλλεγόμενων δεδομένων μπορεί να εφαρμοσθεί περαιτέρω testing και debugging ενώ το live data feed μπορεί να εξισωθεί με αυτό που παρέχουν στα καντράν (gauges) οι νεότεροι κατασκευαστές αυτοκινήτων. Υπάρχουν πάρα πολλά αισθητήρια που μπορεί να μετρήσει κανείς και να εμφανίσει τις τιμές τους στο GUI της εφαρμογής.

Όσον αφορά στο κομμάτι που σχετίζεται με την χρήση των δεδομένων, αρκεί να σκεφτεί κανείς πόσο μεγάλη διαφορά θα έκανε αν κάποιος ασχολείτο σε μεγάλη κλίμακα με την συλλογή αυτών και προχωρούσε σε περαιτέρω μελέτη τους. Πραγματικά, η εξαγωγή συμπερασμάτων και η ενασχόληση με τα Big Data είναι το επόμενο μεγάλο στοίχημα στον χώρο της πληροφορικής τα επόμενα χρόνια αν αναλογιστεί κανείς ότι αντί για μόνο ένα όχημα, η εφαρμογή αυτή μπορεί να εκτελεστεί από έναν στόλο οχημάτων.

Άλλωστε, σκοπός δεν θα ήταν να βοηθηθεί από την παραγόμενη πληροφορία μόνο ο απλός ιδιώτης αλλά και ο επαγγελματίας που έχει το όχημα ως μέσο επιβίωσης. Οι κατασκευαστές που μπορεί να έχουν μέρος αυτών των data δεν έχουν και τις αντίστοιχες μετρήσεις από τις κινητές συσκευές των καταναλωτών τους.

Παράρτημα I - Ορολογία

ABS	(Anti-lock Braking System) , Το σύστημα αντιμπλοκαρίσματος τροχών είναι ηλεκτρονικό και μηχανικό σύστημα ελέγχου της κίνησης των τροχών κατά τη διαδικασία φρεναρίσματος ενός οχήματος.
ACU	(Airbag Control Unit) , Η μονάδα του εγκεφάλου ενός οχήματος που είναι υπεύθυνη για την έγκαιρη αναγνώριση της σοβαρότητας μιας πρόσκρουσης και για την εφαρμογή μέτρων παθητικής ασφάλειας (ενεργοποίηση αερόσακων)
Adapter	(Προσαρμογέας) , Γενική ονομασία για μια ηλεκτρονική συσκευή η οποία μετατρέπει ιδιότητες και δεδομένα από ένα σύστημα σε ένα άλλο σύστημα έτσι ώστε να υπάρξει συμβατότητα στην επικοινωνία
Android	Το Android είναι ένα λειτουργικό σύστημα το οποίο βασίζεται στον πυρήνα του λειτουργικού linux, υποστηρίζει συσκευές κινητών τηλεφώνων, tablets ενώ προσφάτως ανακοινώθηκε και η υποστήριξη συσκευών αρχιτεκτονικής x86. Αναπτύχθηκε από την Google
API	(Application Programming Interface) , όρος με τον οποίο συνήθως περιγράφεται μια έτοιμη βιβλιοθήκη η οποία περιέχει ρουτίνες, δομές δεδομένων, κλάσεις και μεταβλητές και μπορεί να χρησιμοποιηθεί ως πρόσθετο σε κάποιο λογισμικό.
BCU	(Brake Control Unit) , Η μονάδα του εγκεφάλου ενός οχήματος η οποία αναλαμβάνει την διαχείριση του συστήματος πέδησης και φρεναρίσματος
Bluetooth	Βιομηχανικό πρότυπο για την ασύρματη σύνδεση συσκευών για μικρές αποστάσεις, συνήθως μεταξύ κινητών συσκευών.
CAN Bus	(Controller Area Network) , Πρότυπο για επικοινωνία μεταξύ μικροελεγκτών και συσκευών στα ηλεκτρονικά μέρη οχημάτων χωρίς την παρουσία υπολογιστή
CARB	(California Air Resources Board) , Πρότυπο μέσα από το οποίο απαιτήθηκε όλα τα οχήματα που πουλήθηκαν στην Καλιφόρνια από το 1991 να εφαρμόζουν κάποια OBD standards.
CRC	(Cyclic Redundancy Check) , Αλγόριθμος ανίχνευσης λαθών μέσα σε raw data κατά την επικοινωνία μεταξύ δικτύων, συσκευών, μικροελεγκτών, κ.λπ.
CSV	(Comma-Separated Values) , αποτελεί τύπο αρχείου όπου αποθηκεύει τα δεδομένα σε μορφή text, χωρισμένα με κόμμα μεταξύ τους, το οποίο σηματοδοτεί μια ξεχωριστή στήλη δεδομένων. Χρησιμοποιείται για export και import δεδομένων από και προς διάφορα λογισμικά και βάσεις δεδομένων.
DTC	(Diagnostic Trouble Codes) , Καθολικό πρότυπο αναφοράς σε βλάβες οχημάτων
ECU	(Engine Control Unit) , Ηλεκτρονική μονάδα ελέγχου των διαφόρων μηχανικών μερών, συσκευών και αισθητηρίων ενός οχήματος
EEPROM	(Electrically Erasable Programmable Read-Only Memory) , είναι μία νεότερη εξέλιξη των μνήμων ROM. Οι μνήμες, σε αντίθεση με τις μνήμες RAM, διατηρούν τα περιεχόμενά τους και μετά την διακοπή της τροφοδοσίας τους. Μπορούν όμως να διαγραφούν και να επαναπρογραμματιστούν με νέες, ακόμη και πάνω στο κύκλωμα στο οποίο είναι τοποθετημένες
ELM327	Είναι η ονομασία ενός προγραμματιζόμενου μικροελεγκτή ο οποίος είναι ο πιο διαδεδομένος για την PC-to-OBD επικοινωνία μέσω του OBD2 προτύπου των οχημάτων και κατασκευάστηκε από την ELM Electronics .

EOF	(End Of File) , χαρακτηράς-ένδειξη η οποία υποδεικνύει το τέλος της μετάδοσης δεδομένων μέσα από έναν δίαυλο
ESP	(Electronic Stability Programme) , Είναι ένα ηλεκτρονικό σύστημα ενεργητικής ασφάλειας των σύγχρονων αυτοκινήτων, που βελτιώνει την ευστάθειά τους κατά την οδήγηση
Firebase	Σχετικά νέα πλατφόρμα της Google η οποία προσφέρει ένα ολοκληρωμένο πακέτο υπηρεσιών (database, hosting, storage, notifications, cloud, analytics, etc.) για την ανάπτυξη εφαρμογών, ιδιαίτερα για εφαρμογές κινητών συσκευών αλλά και web.
GPS	(Global Positioning System) , Αναφέρεται στο Παγκόσμιο Σύστημα Στιγματοθέτησης μέσα από το οποίο μπορεί να εντοπιστεί η γεωγραφική θέση ενός κινητού ή ακίνητου σημείου.
GUI	(Graphical User Interface) , Αποτελεί το γραφικό περιβάλλον μια εφαρμογής που χρησιμοποιείται για να αλληλοεπιδράσει με τον χρήστη
IAT	(Intake Air Temperature) , Αναφέρεται στον αισθητήρα ενός οχήματος που ανιχνεύει την θερμοκρασία του εισαγόμενου αέρα στον κινητήρα.
IDE	(Integrated Development Environment) , είναι μία σουίτα λογισμικού που βοηθάει στην ανάπτυξη προγραμμάτων υπολογιστή. Συνήθως ένα IDE περιλαμβάνει κάποιον επεξεργαστή πηγαίου κώδικα, έναν μεταγλωττιστή, εργαλεία αυτόματης παραγωγής κώδικα, αποσφαλματωτή, συνδέτη, σύστημα ελέγχου εκδόσεων και εργαλεία κατασκευής γραφικών διασυνδέσεων χρήστη για τις υπό ανάπτυξη εφαρμογές.
Interface	(Διεπαφή) , το σύνορο επικοινωνίας μιας οντότητας (π.χ. το κομμάτι ενός λογισμικού, μια συσκευή υλικού, ένας χρήστης, κτλ.) με το περιβάλλον της.
JDK	(Java Development Kit) , Ολοκληρωμένο πακέτο εντολών, κλάσεων και αλγορίθμων που βοηθά στην ανάπτυξη εφαρμογών χρησιμοποιώντας την γλώσσα προγραμματισμού Java.
JSON	(JavaScript Object Notation) , είναι η μορφή αναπαράστασης δεδομένων η οποία παρουσιάζει απλές δομές σε αναγνώσιμη από τον άνθρωπο μορφή. Χρησιμοποιείται κυρίως για μετάδοση δεδομένων μέσω πρωτοκόλλων διαδικτύου.
JumpStart	Έναρξη λειτουργίας του κινητήρα ενός οχήματος χωρίς την φυσιολογική χρήση της μίζας
MAF	(Mass Air Flow) , αναφέρεται στον αισθητήρα ενός οχήματος ο οποίος ανιχνεύει τον όγκο και την πυκνότητα του αέρα που γίνεται εισαγωγή στον κινητήρα.
MAP	(Manifold Absolute Pressure) , Αναφέρεται στον αισθητήρα ενός οχήματος ο οποίος ανιχνεύει την πίεση στην πολλαπλή εισαγωγή του κινητήρα
MIL	(Malfunction Indicator Light) , είναι η λυχνία που εμφανίζεται στο καντράν ενός οχήματος όταν υπάρχει κάποιο συνήθως σοβαρό πρόβλημα. Λέμε έτσι αλλιώς και το Check Engine.
NoSQL	(Non SQL) Βάση δεδομένων η οποία δεν εφαρμόζει σχεσιακούς κανόνες αποθήκευσης δεδομένων
OBD2	(On-board Diagnostics) , Είναι ένα πρότυπο το οποίο είναι καθολικό για όλα τα οχήματα και προσφέρει επικοινωνία μεταξύ ενός οχήματος και ενός υπολογιστικού συστήματος.
PIC	(Peripheral Interface Controller) , είναι μια οικογένεια μικροελεγκτών η οποία στην ουσία αποτελεί ένα ολοκληρωμένο που χρησιμοποιείται προγραμματιζόμενο σε άλλα ηλεκτρονικά κυκλώματα

REST	(Representational State Transfer) , αποτελεί μία μορφή αρχιτεκτονικής λογισμικού για καταναμημένα συστήματα όπως ο παγκόσμιος ιστός και βασίζεται κατά πολύ στο πρωτόκολλο HTTP.
RS-232	(Recommended Standard 232) , είναι ένα πρότυπο για σειριακή μετάδοση δυαδικών σημάτων δεδομένων μεταξύ συσκευών και υπολογιστικών συστημάτων.
SDK	(Software Development Kit) , είναι ένα πακέτο Ανάπτυξης Λογισμικού το οποίο περιέχει ένα σύνολο εργαλείων ανάπτυξης που επιτρέπουν σε έναν προγραμματιστή να δημιουργήσει λογισμικό εφαρμογών για ένα συγκεκριμένο πακέτο λογισμικού, πλατφόρμα, παιχνιδιομηχανή, λειτουργικά συστήματα, κ.λπ.
SOT	(Start Of Text) , χαρακτήρας-ένδειξη η οποία υποδεικνύει την έναρξη της μετάδοσης δεδομένων μέσα από έναν δίαυλο
TCU	(Transmission Control Unit) , Ηλεκτρική μονάδα ελέγχου της αυτόματης μετάδοσης κίνησης η οποία μέσω αισθητήρων υπολογίζει πότε θα πρέπει να αλλάζουν οι ταχύτητες με γνώμονα τις καλύτερες επιδόσεις αλλά και την ελάχιστη κατανάλωση
UART	(Universal Asynchronous Receiver / Transmitter) είναι ένα κύκλωμα των υπολογιστών το οποίο διαμεσολαβεί στην σειριακή επικοινωνία υπολογιστών ή υπολογιστών με συσκευές μέσω των στάνταρντ θυρών RS-232, RS-422 ή RS-485
VIN	(Vehicle Identification Number) , είναι ένας μοναδικός κωδικός πλαισίου ο οποίος χρησιμοποιείται για την αναγνώριση οχημάτων

Βιβλιογραφία:

1. OBDII Bus Description, <http://www.interfacebus.com/OBDII-pinout-signal-assignment.html>
2. SAE J1850 Description http://www.interfacebus.com/Automotive_SAE_J1850_Bus.html
3. OBD-II signal protocols, Wikipedia, https://en.wikipedia.org/wiki/On-board_diagnostics#OBD-II_signal_protocols
4. SAE J1939 Standards Collection on the Web, <http://store.sae.org/j1939/contents/>
5. All About CAN Bus (November 20, 2012) <https://www.kanda.com/blog/microcontrollers/bus/>
6. International Organization for Standardization, ISO 11898-3:2006 (<https://www.iso.org/standard/36055.html>)
7. International Organization for Standardization, ISO 11898-4:2004 <https://www.iso.org/standard/36306.html>
8. International Organization for Standardization, ISO 11898-5:2007 <https://www.iso.org/standard/41284.html>
9. International Organization for Standardization, ISO 15765-1:2011 <https://www.iso.org/standard/54498.html>
10. International Organization for Standardization, ISO 15765-2:2016 <https://www.iso.org/standard/66574.html>
11. International Organization for Standardization, ISO 14229-3:2012 <https://www.iso.org/standard/55284.html>
12. International Organization for Standardization, ISO 15765-3:2004 <https://www.iso.org/standard/33618.html>
13. International Organization for Standardization, ISO 15765-4:2011 <https://www.iso.org/standard/46045.html>
14. American National Standards Institute (ANSI), ISO 14230-4:2000 <https://webstore.ansi.org/RecordDetail.aspx?sku=ISO+14230-4%3a2000&source=google&adgroup=iso3&keyword=iso%2014230-4&clid=CLSB0PaJorgCFYmS3god9hsAfA>
15. International Organization for Standardization, ISO 9141-2:1994 <https://www.iso.org/obp/ui/#iso:std:iso:9141:-2:ed-1:v1:en>
16. World Wide Harmonization of On-Board Diagnostics, (23 October, 2012) http://www.ukintpress-conferences.com/uploads/SPTXNA12/Day1_2_Greg_Potter.pdf
17. Constantinos Patsakis, Kleanthis Dellios, Mélanie Bourroche, (November 2013), Towards a distributed secure in-vehicle communication architecture for modern vehicles, <http://www.sciencedirect.com/science/article/pii/S016740481300165X>
18. Dan Klinedinst, Christopher King, (March 2016), On Board Diagnostics: Risks and Vulnerabilities of the Connected Vehicle https://resources.sei.cmu.edu/asset_files/WhitePaper/2016_019_001_453877.pdf
19. Karl Koscher, Alexei Czeskis, Franziska Roesner, Shwetak Patel, Tadayoshi Kohno, Experimental Security Analysis of a Modern Automobile, <http://www.autosec.org/pubs/cars-oakland2010.pdf>
20. ELM-USB OBD2 Interface commands description, <http://www.obdtester.com/elm-usb-commands>
21. EnviroCar & Citizen Science, <https://wiki.52north.org/Projects/EnviroCar>
22. OBD Problem Vehicles, <http://www.obdsol.com/knowledgebase/obd-software-development/obd-problem-vehicles/>
23. Reading Real-time Data,

- <http://www.obdsol.com/knowledgebase/obd-software-development/reading-real-time-data/>
24. Andrew Fong, (June 2013), OBD2 PID Reader, <http://digitalcommons.calpoly.edu/cgi/viewcontent.cgi?article=1089&context=cpesp>
 25. How-to Guide for OBDII Reader App Development, (21 May 2014), <http://blog.lemberg.co.uk/how-guide-obdii-reader-app-development>
 26. Firebase Real-time Database, Google, <https://firebase.google.com/docs/database/>
 27. Volumetric Efficiency Testing - Engine Performance, BOSCH, <http://www.natef.org/NATEF/media/NATEFMedia/2015%20ASEIEA%20Conference/BO SCH-VE-Test-Diagnostics-SHORT-7-27-15.pdf>
 28. Paul Higgins, Graham Williams, (10 June 2010), US-Patent - Vehicle fuel consumption calculator, <https://www.google.ch/patents/US8340925>
 29. Alex (Im) En., (February 2013), Reading Performance Information Data (PID), <http://www.totalcardiagnostics.com/support/Knowledgebase/Article/View/43/6/reading-performance-information-data-pid>
 30. Attila Hegedüs, (September 19, 2016), Firebase Tutorial: Getting Started, <https://www.raywenderlich.com/139322/firebase-tutorial-getting-started-2>
 31. OBDSim - Simulate an ELM327 device - Manual, <http://icculus.org/obdgpslogger/manpages/render/obdsim.txt>
 32. Connect OBDSim to Torque on Windows through Bluetooth, (February 2014), Stackoverflow, <https://stackoverflow.com/questions/21957505/connect-obdsim-to-torque-on-windows-through-bluetooth>
 34. SensorEvent, Android Developer's Reference, <https://developer.android.com/reference/android/hardware/SensorEvent.html#values>
 35. Trouble Codes Turn On Your Check Engine Light, <http://trouble-codes.com/>
 36. ECU Explained, <http://www.ecutesting.com/ecu.html>