

ΤΕΧΝΟΛΟΓΙΚΟ ΕΚΠΑΙΔΥΤΙΚΟ ΙΔΡΥΜΑ ΔΥΤΙΚΗΣ
ΕΛΛΑΔΑΣ ΣΧΟΛΗ ΤΕΧΝΟΛΟΓΙΚΩΝ ΕΦΑΡΜΟΓΩΝ
ΤΜΗΜΑ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ Τ.Ε.



ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ:1683

**Υλοποίηση υπηρεσιών απομακρυσμένης
εποπτείας/ελέγχου βασισμένων στις
πλατφόρμες Arduino/Google Maps-
Charts και στις τεχνολογίες
3G/GPS/GPRS**

Συμεωνίδης Οδυσσέας (6807)

Εισηγητής: Τοπάλης Ευάγγελος

Πάτρα, Σεπτέμβριος 2018

Πρόλογος

Η ανάπτυξη της τεχνολογίας τα τελευταία έτη, έχει προκαλέσει σπουδαίες καινοτομίες στον τομέα της πληροφορικής, οι οποίες είχαν σαν αποτέλεσμα την αύξηση της επεξεργαστικής ισχύος των υπολογιστικών συστημάτων και συγχρόνως τη μείωση των διαστάσεων τους αλλά και του κόστους τους. Εφεξής η απόκτηση ενός υπολογιστικού συστήματος με ευρύ φάσμα δυνατοτήτων το οποίο μπορεί να χρησιμοποιηθεί σε ποικίλες εφαρμογές είναι απόλυτα προσιτή.

Το λογισμικό ανοικτού κώδικα(open source) βοήθησε αρκετά στην δημιουργία εργαλείων ανάπτυξης λογισμικού, δεδομένου ότι σαν φιλοσοφία, μέθοδος διανομής και επαναχρησιμοποίησης πηγαίου κώδικα , παρέχει τη δυνατότητα πρόσβασης σε εφαρμογές δωρεάν τόσο σε αυτές όσο και στον πηγαίο κώδικα. Αποτέλεσμα αυτών είναι ότι η ανάπτυξη εφαρμογών αλλά και η βελτιστοποίησή τους επιτυγχάνεται με ευκολία και ταχύτητα εξαιτίας της ελευθερίας που παρέχει το ανοικτού κώδικα λογισμικό.

Επακόλουθο των παραπάνω είναι η υλοποίηση εφαρμογών που είναι σε θέση να εκτελεστούν σε οποιοδήποτε λειτουργικό σύστημα, οι οποίες οδηγούν στη δημιουργία εργαλείων ανεξαρτήτου πλατφόρμας. Έτσι καταλήγουμε στο ότι τα εργαλεία ανάπτυξης του λειτουργικού συστήματος μας, φέρουν τις ίδιες λειτουργίες και είναι πλήρως συμβατά σε διάφορα λειτουργικά συστήματα.

Πλέον η απόκτηση ενός μικροελεγκτή με σκοπό την ανάπτυξη εφαρμογών είναι αρκετά προσιτή καθότι το κόστος τους είναι πολύ χαμηλό. Στην αγορά υπάρχει ποικιλία μικροελεγκτών που συνδυάζουν χαμηλή τιμή και συγχρόνως είναι ικανοί να παράγουν πλήθος εφαρμογών. Υπάρχει τέτοια εξέλιξη στον τομέα ανάπτυξης νέων μικροελεγκτών οι οποίοι προσφέρονται για κάθε γενική ή εξειδικευμένη εφαρμογή.

Χάρης στην ευχρηστία των εργαλείων ανάπτυξης εφαρμογών για μικροελεγκτές, η διαδικασία αυτή δεν αποτελεί έργο που μπορεί να φέρει σε πέρας μόνο ένας πεπειραμένος και εξειδικευμένος προγραμματιστής. Συνακόλουθο αυτού είναι η ανάπτυξη εφαρμογών χωρίς να χρειάζεται κάποιος να είναι ειδικός στην διαδικασία αυτή και αυτό διότι πολλές λειτουργίες έχουν καμουφλαριστεί κάτω από γραφικά περιβάλλοντα, τα οποία μέσω εύχρηστων καταλόγων διαθέσιμων επιλογών αυτοματοποιούν τη διαδικασία.

Περίληψη

Η πτυχιακή εργασία αφορά τη δημιουργία ενός συστήματος εποπτείας της πορείας ενός ηλικιωμένου ή κατοικίδιου μέσω των χαρτών της Google, το οποίο θα λειτουργεί ασύρματα με την χρήση των δεδομένων κινητής τηλεφωνίας. Πιο συγκεκριμένα για την υλοποίηση του συστήματος θα χρησιμοποιηθεί η αναπτυξιακή πλατφόρμα Arduino.

Το Arduino είναι μια υπολογιστική πλατφόρμα βασισμένη σε μια απλή μητρική πλακέτα ανοικτού κώδικα, με ενσωματωμένο μικροελεγκτή και εισόδους/εξόδους, η οποία μπορεί να προγραμματιστεί με τη γλώσσα C και ένα σύνολο από βιβλιοθήκες υλοποιημένες στην C/C++. Μία πλακέτα Arduino αποτελείται από ένα μικροελεγκτή Atmel AVR (ATmega328 και ATmega168 στις νεότερες εκδόσεις, ATmega8 στις παλαιότερες) και συμπληρωματικά εξαρτήματα για τη διευκόλυνση του χρήστη στον προγραμματισμό και την ενσωμάτωσή του σε άλλα κυκλώματα. Σε αυτή την πλατφόρμα μπορούν να συνδεθούν ενσύρματα ή ασύρματα διάφοροι αισθητήρες ή ενεργοποιητές (συσκευές δράσης).

Στην περίπτωση μας θα χρησιμοποιηθεί συνδυαστικά η μονάδα Sim808, η οποία θα είναι υπεύθυνη για τον εντοπισμό της τοποθεσίας αλλά και τη σύνδεση με το δίκτυο κινητής τηλεφωνίας. Επίσης θα εκτελεστούν υπηρεσίες όπως κουμπί πανικού, συναγερμός αν το κατοικίδιο απομακρυνθεί και ενημέρωση για τις περιβαλλοντικές συνθήκες. Ειδοποίηση για αυτά τα συμβάντα θα υπάρχει μέσω της δυνατότητας του Sim808 να πραγματοποιεί κλήσεις και να στέλνει μηνύματα.

Τέλος θα γίνει εκτεταμένη αναφορά στους μικροελεγκτές Arduino, στην πλακέτα Sim808, στις γλώσσες προγραμματισμού που χρησιμοποιήθηκαν όπως sql, php, javascript, html αλλά και στα Google Charts.

Περιεχόμενα

Πρόλογος.....	1
Περίληψη.....	2
Περιεχόμενα.....	3
Κατάλογος Εικόνων.....	5
Κατάλογος Πινάκων.....	7
Κεφάλαιο 1: Θεωρητικό υπόβαθρο.....	8
1.1 Αρχιτεκτονική Arduino.....	8
1.1.1 Είδη Arduino.....	8
1.1.2 Είδη Arduino Shields.....	9
1.1.3 Μικροελεγκτής Arduino Uno.....	11
1.2 Τροφοδοσία Arduino Uno.....	13
1.2.1 Σειριακή Επικοινωνία Arduino Uno.....	15
1.3 Προγραμματισμός Διαδικτύου.....	16
1.3.1 Javascript.....	16
1.3.2 HTML.....	17
1.3.3 CSS.....	20
1.3.4 PHP.....	20
1.3.5 My SQL.....	22
1.4 Εργαλεία Ανάπτυξης Λογισμικού.....	23
1.4.1 Notepad++.....	24
1.4.2 Xampp.....	24
1.4.3 Fritzting.....	25
1.4.4 Arduino Ide.....	25
1.4.5 Γλώσσα και δομή προγράμματος.....	27
Κεφάλαιο 2 : Υλικό μέρος συστήματος.....	30
2.1 Μονάδα Sim808.....	31
2.1.1 AT-Commands.....	34
2.2.2 AT-Commands GSM επικοινωνίας.....	35
2.2.3 AT-Commands GPRS.....	37
2.2.4 AT-Commands GPS.....	38
2.3 Αισθητήρας Θερμοκρασίας-Υγρασίας DHT-22.....	38

2.4 Κόστος Συστήματος	40
Κεφάλαιο 3 : Λογισμικό μέρος συστήματος	41
3.1.1 Υπηρεσίες συστήματος	42
3.2 Προγραμματισμός Arduino υπο	43
3.2.1 Βασικές βιβλιοθήκες	43
3.2.2 Προγραμματισμός Αισθητήρα DHT22	45
3.2.3 Υπολογισμός δεδομένων GPS	45
3.3 Προγραμματισμός υπηρεσιών	48
3.3.1 Panic Button	48
3.3.2 Παρακολούθηση απόστασης από σημείο	50
3.3.3 Συνθήκες Περιβάλλοντος	51
3.3.4 Εποπτεία τοποθεσίας	52
3.4 Βάση δεδομένων	53
3.4.1 Αρχείο Php για σύνδεση με τη βάση δεδομένων	56
3.4.2 Αρχεία Php για την γραφική απεικόνιση των μεταβλητών	57
3.4.2.1 Google Gauge	57
3.4.2.2 Statistics	61
3.4.2.3 Weekly Statistics	65
3.4.2.4 Monthly Statistics	69
3.5 Δημιουργία χάρτη	73
3.5.1 Αρχεία php και html για την παρουσίαση του χάρτη	74
3.6 Ιστοχώρος που υλοποιήθηκε και λειτουργίες του	80
3.6.1 Κεντρική σελίδα ιστοχώρου	80
3.6.2 Αρχεία CSS	87
Κεφάλαιο 4 : Συμπεράσματα-μελλοντικές επεκτάσεις	88
Βιβλιογραφία	89
Παράρτημα Α	90
Παράρτημα Β	98

Κατάλογος Εικόνων

Εικόνα 01:Arduino Uno	8
Εικόνα 02:Arduino Mega ADK.....	8
Εικόνα 03:Arduino Nano	9
Εικόνα 04:Ethernet Shield.....	10
Εικόνα 05:Wi-fi Shield.....	10
Εικόνα 06:Motor Shield	10
Εικόνα 07:LCD display	11
Εικόνα 08:Sim808	11
Εικόνα 09:Αναλυτική περιγραφή Arduino	12
Εικόνα 10:Διάγραμμα Μικροελεγκτή.....	12
Εικόνα 11:Αναλυτική περιγραφή Μικροελεγκτή	13
Εικόνα 12:Pins τροφοδοσίας Arduino UNO.....	13
Εικόνα 13:Ψηφιακά pins εισόδου/εξόδου Arduino Uno.....	14
Εικόνα 14:Αναλογικά pins εισόδου/εξόδου Arduino Uno	15
Εικόνα 15:Γλώσσες Διαδικτύου	16
Εικόνα 16:Λογότυπο Javascript	17
Εικόνα 17:Λογότυπο HTML.....	18
Εικόνα 18:Βασικά tags HTML.....	18
Εικόνα 19:Βασική σύνταξη CSS.....	20
Εικόνα 20:Λογότυπο PHP.....	21
Εικόνα 21: Λογότυπο MYSQL.....	23
Εικόνα 22: Λογότυπο Notepad++.....	24
Εικόνα 23:Καινούργιο αρχείο Notepad++	24
Εικόνα 24:Λογότυπα υποστηριζόμενα από το Xampp.....	25
Εικόνα 25:Σχέδιο Fritzing	25
Εικόνα 26:Εμφάνιση Συστήματος.....	25
Εικόνα 27:Περιβάλλον IDE	26
Εικόνα 28:Σειριακή IDE	27
Εικόνα 29:Αρχική εμφάνιση IDE.....	28
Εικόνα 30:Κύκλωμα βασικού συστήματος.....	30
Εικόνα 31:Πλαकेτα με τον Sim808	31
Εικόνα 32:Ανάλυση επιμέρους στοιχείων του Sim808.....	32
Εικόνα 33:Pinout του Sim808.....	34
Εικόνα 34:Μπροστά και πίσω όψη του Sim808.....	35
Εικόνα 35:Αισθητήρας DHT-22	39
Εικόνα 36:Διάγραμμα Λογισμικού μέρους	42
Εικόνα 37:Αρχική σελίδα phpmyadmin.....	54
Εικόνα 38:Πίνακας markers στη βάση δεδομένων	56
Εικόνα 39:Λογότυπο google charts	57
Εικόνα 40:Ο μετρητής με την τελευταία ένδειξη του γεωγραφικού πλάτους.....	61
Εικόνα 41:Η γραφική της ταχύτητας συναρτήσεως του χρόνου.....	64
Εικόνα 42:Η γραφική του υψομέτρου συναρτήσεως του χρόνου.....	65
Εικόνα 43:Η απεικόνιση της ελάχιστης, μέγιστης και μέσης τιμής της ταχύτητας	68

Εικόνα 44:Η απεικόνιση της ελάχιστης, μέγιστης και μέσης τιμής του υψομέτρου	69
Εικόνα 45:Η απεικόνιση της ελάχιστης, μέγιστης και μέσης τιμής της ταχύτητας	72
Εικόνα 46:Η απεικόνιση της ελάχιστης, μέγιστης και μέσης τιμής του υψομέτρου	73
Εικόνα 47:Το λογότυπο των χαρτών της Google.....	73
Εικόνα 48:Μορφή xml στον browser	76
Εικόνα 49: Εμφάνιση σημαδιών στο χάρτη	79
Εικόνα 50: Εμφάνιση σημαδιού με παράθυρο πληροφοριών	79
Εικόνα 51:Αρχική σελίδα ιστοχώρου	82
Εικόνα 52:Σύνδεσμος μετρητών.....	83
Εικόνα 53:Σελίδα μετρητών.....	83
Εικόνα 54:Σύνδεσμος στατιστικών	83
Εικόνα 55:Σελίδα στατιστικών	84
Εικόνα 56:Συνέχεια σελίδας στατιστικών	84
Εικόνα 57:Σύνδεσμος σελίδας εβδομαδιαίων στατιστικών.....	84
Εικόνα 58:Σελίδα εβδομαδιαίων στατιστικών.....	85
Εικόνα 59:Συνέχεια σελίδας εβδομαδιαίων στατιστικών	85
Εικόνα 60:Σύνδεσμος σελίδας μηνιαίων στατιστικών	85
Εικόνα 61:Σελίδα μηνιαίων στατιστικών	86
Εικόνα 62:Συνέχεια σελίδας μηνιαίων στατιστικών	86
Εικόνα 63:Σύνδεσμος σελίδας χάρτη σε μορφή κουμπιού.....	86
Εικόνα 64:Εμφάνιση υποσέλιδου	87

Κατάλογος Πινάκων

Πίνακας 01: Είδη Arduino.....	8
Πίνακας 02: Είδη Shields	9
Πίνακας 03: Tags HTML.....	19
Πίνακας 04: Βασικές εντολές Php.....	22
Πίνακας 05: Λειτουργίες και συναρτήσεις Arduino Ide	28
Πίνακας 06: AT-Commands SIM κάρτας	35
Πίνακας 07: AT-Commands SMS.....	36
Πίνακας 08: AT-Commands GPRS	37
Πίνακας 09: AT-Commands GPS.....	38
Πίνακας 10: Ηλεκτρικά Χαρακτηριστικά DHT-22.....	39
Πίνακας 11: Λεπτομερής πίνακας τεχνικών χαρακτηριστικών του DHT22	40
Πίνακας 12: Κόστος συστήματος.....	41
Πίνακας 13: Στήλες βάσης δεδομένων	55

Κεφάλαιο 1: Θεωρητικό υπόβαθρο


Το θέμα του συγκεκριμένου κεφαλαίου αφορά τις θεωρητικές γνώσεις που πρέπει να υπάρχουν για να πραγματοποιηθεί η πτυχιακή εργασία. Συγκεκριμένα θα αναλυθούν τα εργαλεία που χρησιμοποιήθηκαν όπως ο μικροελεγκτής arduino και η μονάδα sim808. Το κεφάλαιο αποτελείται από την αρχιτεκτονική και τις συνδέσεις του arduino ,τον προγραμματισμό του διαδικτύου και την υπηρεσία χαρτών της google.



1.1 Αρχιτεκτονική Arduino

Το arduino πήρε το όνομά του από τους ιδρυτές Massimo Banzi και David Cueartielles οι οποίοι το ονόμασαν αρχικά πρόγραμμα «arduino». Έπειτα ξεκίνησαν να παράγουν πλακέτες με βασική αρχή το δημιούργημα τους να βασίζεται στην ευκολία χρήσης τόσο του υλικού όσο και του λογισμικού. Το Arduino είναι μια υπολογιστική πλατφόρμα ανοικτού hardware/κώδικα βασισμένη σε μια απλή μητρική πλακέτα, με ενσωματωμένο μικροελεγκτή και εισόδους/εξόδους ,η οποία μπορεί να προγραμματιστεί με τη γλώσσα Wiring.Ουσιαστικά πρόκειται για τη γλώσσα προγραμματισμού C και ένα σύνολο από βιβλιοθήκες, υλοποιημένες επίσης στην C++ .Απευθύνεται κυρίως σε ερασιτέχνες που κάνουν τα πρώτα τους βήματα στον προγραμματισμό, σε λάτρεις της τεχνολογίας αλλά και σε επαγγελματίες. Στην ενότητα αυτή θα παρουσιαστεί ο τύπος arduino που χρησιμοποιήθηκε και οι βασικές μονάδες (module) που συνεργάζονται με αυτό με σκοπό την επέκταση των λειτουργιών του.

1.1.1 Είδη Arduino

Στον πίνακα που ακολουθεί παρουσιάζονται οι βασικές πλακέτες arduino που χρησιμοποιούνται στην πλειοψηφία των εφαρμογών. Ταυτόχρονα θα παραθέτονται και τα τεχνικά τους χαρακτηριστικά. Κάθε πλακέτα έχει τα πλεονεκτήματά της, αυτά αφορούν το μέγεθος, την επεξεργαστική ισχύ αλλά και το κόστος.




Είδος Arduino	Τεχνικά χαρακτηριστικά
 <p>Εικόνα 1: Arduino Uno</p>	<p>Μικροελεγκτής: ATmega328 Τάση λειτουργίας: 5V Ψηφιακές εισόδους / εξόδους: 14 (6 παρέχουν PWM έξοδο) Ρεύμα DC ανά είσοδο / έξοδο: 40 mA Αναλογικές εισόδους: 6 Μνήμη Flash: 32 KB(ATmega328) SRAM: 2 KB (ATmega328) EEPROM: 1 KB (ATmega328) Ταχύτητα ρολογιού: 16 MHz</p>

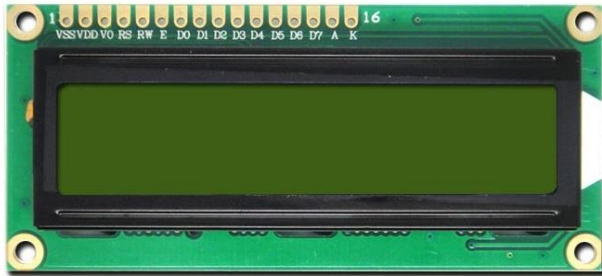
 <p>Εικόνα 2 : Arduino Mega ADK</p>	<p>Μικροελεγκτής : ATmega 2560 Τάση λειτουργίας: 5V Ψηφιακές εισοδοι / έξοδοι: 54 (15 παρέχουν PWM έξοδο) Ρεύμα DC ανά εισοδο / έξοδο: 40 mA Αναλογικές εισοδοι: 16 Μνήμη Flash: 256 KB SRAM: 8 KB EEPROM: 4 KB Ταχύτητα ρολογιού: 16 MHz</p>
 <p>Εικόνα 3 : Arduino Nano</p>	<p>Μικροελεγκτής: ATmega32u4 Τάση λειτουργίας: 5V Ψηφιακές εισοδοι / έξοδοι: 20 (7 παρέχουν PWM έξοδο) Ρεύμα DC ανά εισοδο / έξοδο: 40 mA Αναλογικές εισοδοι: 12 Μνήμη Flash: 32 KB (ATmega32u4) SRAM: 2.5 KB (ATmega32u4) EEPROM: 1 KB (ATmega32u4) Ταχύτητα ρολογιού: 16 MHz</p>

Πίνακας 1: Είδη Arduino

1.1.2 Είδη Arduino Shields

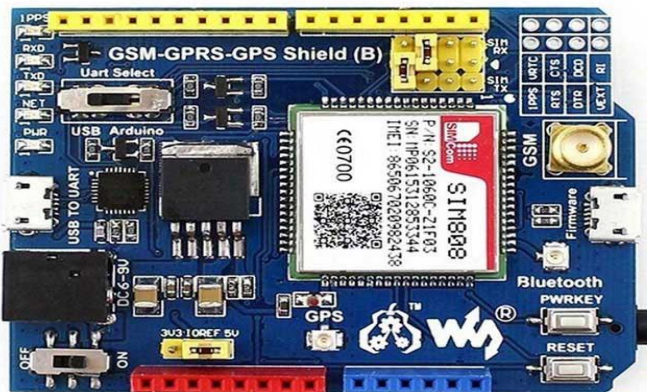
Μετά από δεκατρία χρόνια λειτουργίας του arduino καταλαβαίνει κανείς ότι έχουν δημιουργηθεί αναρίθμητα shields. Στον παρακάτω πίνακα θα παρουσιαστούν τα βασικά shields που είναι υπεύθυνα για τη σύνδεση του arduino με το διαδίκτυο.

Είδος Shield	Περιγραφή
 <p data-bbox="469 779 818 806">Εικόνα 4 : Ethernet Shield</p>	<p data-bbox="1036 548 1412 699">Επιτυγχάνει ενσύρματη επικοινωνία με το διαδίκτυο με τη χρήση καλωδίου Ethernet. Περιλαμβάνεται θύρα υποδοχής SD.</p>
 <p data-bbox="501 1316 786 1344">Εικόνα 5: Wi-fi Shield</p>	<p data-bbox="1036 1060 1396 1182">Επιτυγχάνει ασύρματη επικοινωνία με το διαδίκτυο μέσω Wi-Fi. Περιλαμβάνεται θύρα υποδοχής SD.</p>
 <p data-bbox="493 1829 794 1856">Εικόνα 6: Motor Shield</p>	<p data-bbox="1036 1556 1406 1797">Επιτρέπει την οδήγηση δύο κινητήρων συνεχούς ρεύματος, ελέγχοντας την ταχύτητα και την κατεύθυνση του καθενός ξεχωριστά. Μπορεί επίσης να μετρήσει την κατανάλωση ρεύματος κάθε κινητήρα.</p>



Εικόνα 7: LCD display

Η οθόνη αποτελείται από ένα lcd πάνελ, σε αυτή μπορούν να απεικονιστούν όλα τα αποτελέσματα εφαρμογών/αισθητήρων που περιέχονται σε προγράμματα του arduino.



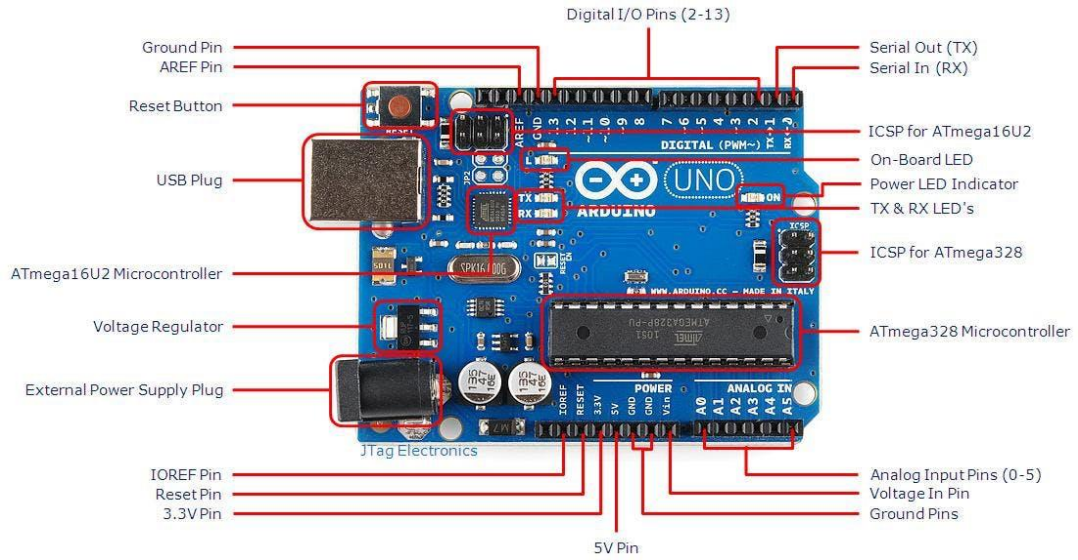
Εικόνα 8: Sim808

Η πλακέτα περιέχει τον επεξεργαστή Sim808 που συνδέεται πάνω από το κύριο Arduino uno .Ο επεξεργαστής είναι υπεύθυνος για τις λειτουργίες του GPS/GSM/GPRS.

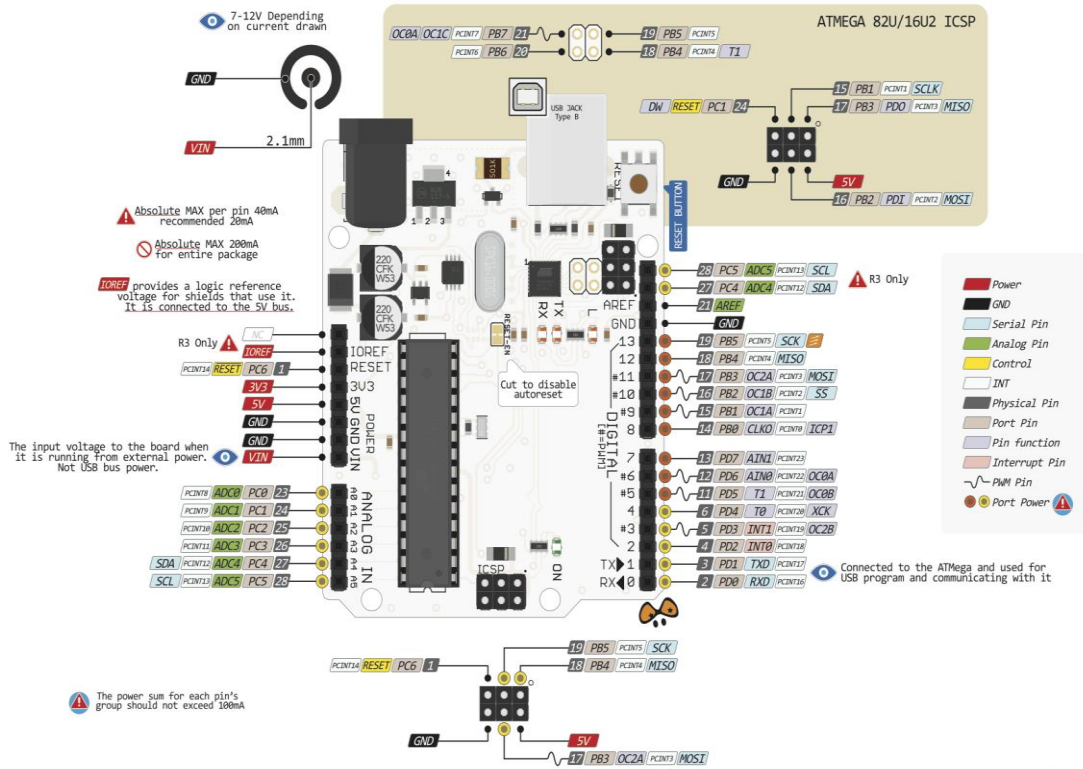
Πίνακας 2: Είδη Shields

1.1.3 Μικροελεγκτής Arduino Uno

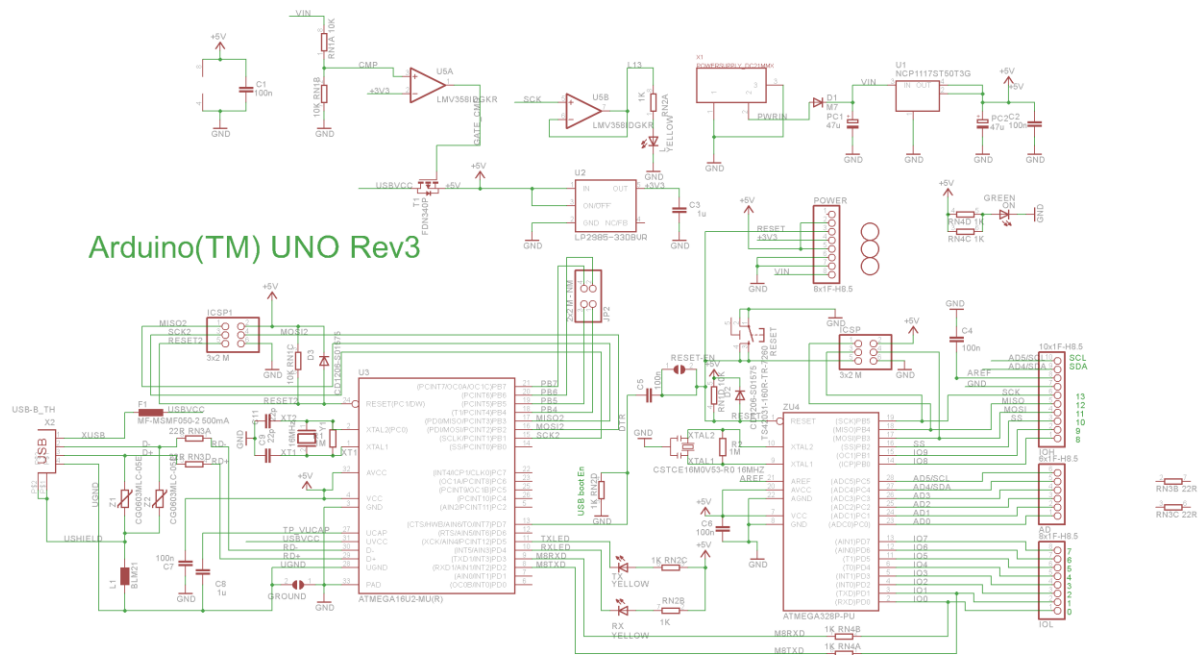
Το κύριο γρανάζι του συστήματος που υλοποιήθηκε είναι η πλακέτα του arduino uno. Η πολύ μεγάλη απήχηση που έχει από τους χρήστες arduino δικαιολογείται λόγω του μικρού κόστους του και της εξαιρετικά χαμηλής κατανάλωσης ρεύματος .Σε αυτή την ενότητα θα παρατεθούν οι ηλεκτρικές συνδέσεις που περιέχει, το σχηματικό του διάγραμμα και αναλυτικότερα τα τεχνικά του χαρακτηριστικά.



Εικόνα 9: Αναλυτική περιγραφή Arduino



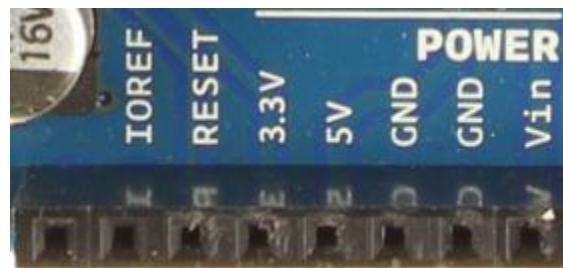
Εικόνα 10: Διάγραμμα μικροελεγκτή



Εικόνα 11: Αναλυτική γραφική αναπαράσταση μικροελεγκτή

1.2 Τροφοδοσία Arduino Uno

Όσον αφορά την τροφοδοσία, το Arduino UNO μπορεί να τροφοδοτείται μέσω της σύνδεσης USB ή με εξωτερικό τροφοδοτικό. Η πηγή ενέργειας επιλέγεται αυτόματα από την πλακέτα. Εξωτερικά μπορεί να τροφοδοτείται είτε από έναν προσαρμογέα AC σε DC 7 – 12V, είτε από μπαταρία. Ο προσαρμογέας θα πρέπει να καταλήγει σε βύσμα διαμέτρου 2,1mm με θετικό κέντρο. Για τροφοδοσία από μπαταρία, αυτή μπορεί να συνδεθεί στα Gnd και Vin pins. Στην παρακάτω εικόνα φαίνονται αυτά τα pins μαζί με τα υπόλοιπα pins τροφοδοσίας του Arduino UNO.



Εικόνα 12: Pins τροφοδοσίας Arduino UNO.

Αναλυτικότερα:

1. **VIN:** Η τάση εισόδου στην πλακέτα Arduino όταν χρησιμοποιείται μια εξωτερική πηγή ενέργειας. Είναι δυνατό είτε να εισάγουμε τροφοδοσία σ' αυτό το pin, είτε να το χρησιμοποιούμε για να έχουμε πρόσβαση στην τροφοδοσία που συνδέθηκε στο βύσμα 2.1mm.

2. **5V:** Παρέχει σταθεροποιημένη έξοδο 5V.
3. **3V3:** Παρέχει 3,3V που παράγεται από τον σταθεροποιητή. Η μέγιστη κατανάλωση ρεύματος είναι 50 mA .
4. **GND:** Pins γείωσης.
5. **IOREF:** Αυτό το pin παρέχει την τάση αναφοράς με την οποία λειτουργεί ο μικροελεγκτής. Ένα σωστά ρυθμισμένο shield μπορεί να διαβάσει την τάση στο pin IOREF και να επιλέξει την κατάλληλη πηγή ενέργειας ή να ενεργοποιήσει μεταφραστές τάσης στις εξόδους για την εργασία με 5V ή 3.3V.

Όσον αφορά τη μνήμη, ο μικροελεγκτής ATmega328, έχει τρεις ομάδες μνήμης. Διαθέτει **flash memory**, στην οποία αποθηκεύονται τα Arduino sketch, SRAM (static random access memory), στην οποία δημιουργείται το sketch και χρησιμοποιεί τις μεταβλητές όταν τρέχει, και EEPROM, η οποία χρησιμοποιείται από τους προγραμματιστές για την αποθήκευση μακροχρόνιων πληροφοριών. Αναλυτικότερα:

1. **2KB μνήμης SRAM:** Η ωφέλιμη μνήμη που μπορούν να χρησιμοποιήσουν τα προγράμματα για να αποθηκεύουν μεταβλητές, πίνακες κ.λπ. Η μνήμη χάνει τα δεδομένα της όταν η παροχή ρεύματος στο Arduino σταματήσει ή πατηθεί το κουμπί επανεκκίνησης.
2. **1KB μνήμης EEPROM:** Μπορεί να χρησιμοποιηθεί για εγγραφή ή ανάγνωση δεδομένων από τα προγράμματα. Σε αντίθεση με την SRAM, δε χάνει τα περιεχόμενά της με απώλεια τροφοδοσίας ή επανεκκίνησης.
3. **32KB μνήμης Flash:** Τα 2 KB χρησιμοποιούνται για τον bootloader, προκειμένου ο μικροελεγκτής να είναι σε θέση να προγραμματίζεται μέσω USB. Τα υπόλοιπα 30KB της μνήμης Flash χρησιμοποιούνται για την αποθήκευση των προγραμμάτων του χρήστη, αφού πρώτα μεταγλωττιστούν στον υπολογιστή. Η μνήμη Flash, δε χάνει τα περιεχόμενά της με απώλεια τροφοδοσίας ή επανεκκίνησης.

Τέλος, αναφορικά με τα pins εισόδου και εξόδου, χωρίζονται σε αναλογικά και ψηφιακά. Ακολουθεί η εικόνα με την μορφή και την επεξήγησή τους.



Εικόνα 13 : Ψηφιακά pins εισόδου / εξόδου Arduino UNO

Τα pins από 0 έως 13 μπορούν να χρησιμοποιηθούν ως ψηφιακές εξόδους. Το pin

AREF χρησιμοποιείται για να παρέχεται τάση αναφοράς στα αναλογικά pins που θα αναλυθούν στη συνέχεια. Εκτός από ψηφιακές έξοδοι, ορισμένα pins επιτελούν και άλλες εργασίες. Αναλυτικότερα:

1. **Pins 0 και 1:** λειτουργούν ως RX και TX της σειριακής θύρας όταν το πρόγραμμά ενεργοποιεί τη σειριακή θύρα. Έτσι, όταν το πρόγραμμά στέλνει δεδομένα στη σειριακή θύρα, αυτά προωθούνται και στη θύρα USB μέσω του ελεγκτή USB, αλλά και στον ακροδέκτη 0 για να τα διαβάσει ενδεχομένως μια άλλη συσκευή. Αυτό φυσικά σημαίνει ότι αν στο πρόγραμμά ενεργοποιήσει το σειριακό interface, χάνει 2 ψηφιακές εισόδους/εξόδους η πλατφόρμα.
2. **Pins 2 και 3:** λειτουργούν και ως εξωτερικά interrupt (interrupt 0 και 1 αντίστοιχα). Ρυθμίζονται μέσα από το πρόγραμμα, ώστε να λειτουργούν αποκλειστικά ως ψηφιακές εισοδοι στις οποίες όταν συμβαίνουν συγκεκριμένες αλλαγές, η κανονική ροή του προγράμματος σταματάει άμεσα και εκτελείται μια συγκεκριμένη συνάρτηση. Τα εξωτερικά interrupt είναι ιδιαίτερα χρήσιμα σε εφαρμογές που απαιτούν συγχρονισμό μεγάλης ακρίβειας.
3. **Pins 3, 5, 6, 9, 10 και 11:** μπορούν να λειτουργήσουν και ως ψευδό-αναλογικές έξοδοι χρησιμοποιώντας PWM (Pulse Width Modulation).



Εικόνα 14: Αναλογικά pins εισόδου Arduino UNO.

Όπως φαίνεται στην **Εικόνα 14**, υπάρχει μια σειρά από 6 pins, τα οποία αποτελούν τις αναλογικές εισόδους του Arduino UNO. Η τάση αναφοράς για τις εισόδους αυτές, μπορεί να ρυθμιστεί τροφοδοτώντας εξωτερικά με τάση το pin AREF που αναφέρθηκε παραπάνω. Για παράδειγμα, αν το pin AREF τροφοδοτηθεί με 3.3V και στη συνέχεια διαβαστεί κάποιο αναλογικό pin στο οποίο εφαρμόζεται τάση 1.65V, το Arduino θα επιστρέψει την τιμή 512.

1.2.1 Σειριακή Επικοινωνία Arduino Uno

Το Arduino διαθέτει τη δυνατότητα να επικοινωνεί με ηλεκτρονικό υπολογιστή, ένα άλλο Arduino ή άλλους μικροελεγκτές. Ο ATmega υποστηρίζει σειριακή επικοινωνία TTL στα 5 Volt τύπου UART, η οποία είναι διαθέσιμη από τους ακροδέκτες RX 0 για λήψη και TX 1 για εκπομπή. Ακόμα έχει ένα ενσωματωμένο FTDI FT232RL IC το οποίο παρέχει σειριακή επικοινωνία με ηλεκτρονικό υπολογιστή, το οποίο χρησιμοποιείται για τη μεταφορά των προγραμμάτων που σχεδιάζονται στο Arduino IDE, μέσω της θύρας USB με την βοήθεια των ανάλογων drivers. Οι drivers αυτοί περιλαμβάνονται στο software Arduino IDE και παρέχουν μία ιδεατή θύρα (virtual port) επικοινωνίας στον ηλεκτρονικό υπολογιστή για σκοπούς επικοινωνίας.

1.3 Προγραμματισμός Διαδικτύου

Η έννοια του προγραμματισμού διαδικτύου αναφέρεται στις εργασίες που απαιτούνται για την ανάπτυξη μιας ιστοσελίδας. Οι εργασίες αυτές στην παρούσα πτυχιακή περιλαμβάνουν την ανάλυση όλων των γνώσεων και συνάμα των εργαλείων που χρησιμοποιήθηκαν, για το σχεδιασμό και την υλοποίησή της. Σε αυτό το κεφάλαιο θα αναλυθούν οι γλώσσες προγραμματισμού και οι πλατφόρμες ανάπτυξη λογισμικού που χρησιμοποιήθηκαν.



Εικόνα 15: Γλώσσες διαδικτύου

1.3.1 Javascript

Η JavaScript είναι μία από τις πιο δημοφιλείς γλώσσες προγραμματισμού ηλεκτρονικών υπολογιστών στον Παγκόσμιο Ιστό (Web) και ταυτόχρονα η δημοφιλέστερη scripting γλώσσα στον κόσμο. Μια γλώσσα scripting είναι μια ελαφριά γλώσσα προγραμματισμού που υποστηρίζει τη συγγραφή σεναρίων. Σενάρια είναι γραμμές κώδικα που μπορούν να ερμηνεύονται και να εκτελούνται χωρίς μεταγλώττιση. Αρχικά όμως, πολλοί επαγγελματίες προγραμματιστές υποτίμησαν τη γλώσσα αυτή διότι το κοινό της ήταν ερασιτέχνες συγγραφείς ιστοσελίδων. Με τη χρήση της τεχνολογίας Ajax, η JavaScript γλώσσα επέστρεψε στο προσκήνιο και έφερε πιο επαγγελματική προσοχή προγραμματισμού. Το αποτέλεσμα ήταν ένα καινοτόμο αντίκτυπο στην εξάπλωση των πλαισίων και των βιβλιοθηκών, τη βελτίωση προγραμματισμού με JavaScript, καθώς και αυξημένη χρήση της JavaScript έξω από τα προγράμματα περιήγησης στο Web. Είναι

η τυπική γλώσσα που χρησιμοποιείται σε ιστοσελίδες, αλλά χρησιμοποιείται ευρέως επίσης από desktop εφαρμογές, εφαρμογές κινητής τηλεφωνίας, καθώς και διακομιστές διαδικτύου.



Εικόνα 16: Λογότυπο Javascript

Η αρχική έκδοση της Javascript βασίστηκε στη σύνταξη στη γλώσσα προγραμματισμού C, αν και έχει εξελιχθεί, ενσωματώνοντας πια χαρακτηριστικά από νεότερες γλώσσες. Αρχικά χρησιμοποιήθηκε για προγραμματισμό από την πλευρά του πελάτη (client), που ήταν ο περιηγητής (browser) του χρήστη, και χαρακτηρίστηκε σαν client-side γλώσσα προγραμματισμού. Αυτό σημαίνει ότι η επεξεργασία του κώδικα Javascript και η παραγωγή του τελικού περιεχομένου HTML δεν πραγματοποιείται στο διακομιστή, αλλά στο πρόγραμμα περιήγησης των επισκεπτών, ενώ μπορεί να ενσωματωθεί σε στατικές σελίδες HTML.

Αντίθετα, άλλες γλώσσες όπως η PHP εκτελούνται στο διακομιστή (server-side γλώσσες προγραμματισμού). Η JavaScript έχει τη δυνατότητα δυναμικής αλλαγής ετικετών και περιεχομένου της HTML, δηλαδή μπορεί να διαβάσει και να αλλάξει το περιεχόμενο των στοιχείων της. Επίσης μπορεί να χειριστεί το CSS. Μπορεί ακόμα να χρησιμοποιηθεί για την επικύρωση δεδομένων, όπως η επικύρωση μιας φόρμας εισόδου και για την αποθήκευση και ανάκτηση πληροφοριών στον υπολογιστή του χρήστη. Ακόμα είναι δυνατόν να ρυθμιστεί ώστε να εκτελείται όταν συμβαίνει κάτι, όπως όταν ένας χρήστης κάνει κλικ σε ένα στοιχείο HTML. Τέλος μπορεί να επικοινωνήσει με php, xml, json αρχεία και με άλλες ιστοσελίδες.

Ο κώδικας Javascript μιας σελίδας περικλείεται από τις ετικέτες της HTML `<script type="text/javascript">` και `</script>`. [20]

1.3.2 HTML

Η HTML (HyperText Markup Language (Γλώσσα Σήμανσης Υπερκειμένου) είναι η κύρια γλώσσα σήμανσης για τις ιστοσελίδες, και τα στοιχεία της είναι τα βασικά δομικά στοιχεία των ιστοσελίδων. Η HTML γράφεται υπό μορφή στοιχείων τα οποία αποτελούνται από ετικέτες (tags), οι οποίες περικλείονται μέσα σε σύμβολα «μεγαλύτερο από» και «μικρότερο από» (για παράδειγμα `<html>`), μέσα στο περιεχόμενο της ιστοσελίδας. [18]



Εικόνα 17: Λογότυπο HTML

Οι ετικέτες HTML συνήθως λειτουργούν ανά ζεύγη (για παράδειγμα `<h1>` και `</h1>`), με την πρώτη να ονομάζεται ετικέτα έναρξης και τη δεύτερη ετικέτα λήξης ή σε άλλες περιπτώσεις ετικέτα ανοίγματος και ετικέτα κλεισίματος αντίστοιχα. Ανάμεσα στις ετικέτες, οι σχεδιαστές ιστοσελίδων μπορούν να τοποθετήσουν κείμενο, πίνακες, εικόνες και άλλα. Ο σκοπός ενός web browser είναι να διαβάσει τα έγγραφα HTML και τα συνθέσει σε σελίδες που μπορεί κανείς να διαβάσει ή να ακούσει. Ο browser δεν εμφανίζει τις ετικέτες HTML, αλλά τις χρησιμοποιεί για να ερμηνεύσει το περιεχόμενο της σελίδας. Τα στοιχεία της HTML χρησιμοποιούνται για να κτίσουν όλους του ιστότοπους.

Η HTML επιτρέπει την ενσωμάτωση εικόνων και άλλων αντικειμένων μέσα στη σελίδα, και μπορεί να χρησιμοποιηθεί για να εμφανίσει διαδραστικές φόρμες. Παρέχει τις μεθόδους δημιουργίας δομημένων εγγράφων δηλαδή εγγράφων που αποτελούνται από το περιεχόμενο που μεταφέρουν και από τον κώδικα μορφοποίησης του περιεχομένου καθορίζοντας δομικά σημαντικά στοιχεία για το κείμενο, όπως κεφαλίδες, παραγράφους, λίστες, συνδέσμους, παραθέσεις και άλλα. Μπορούν επίσης να ενσωματώνονται σενάρια εντολών σε γλώσσες όπως η JavaScript, τα οποία επηρεάζουν τη συμπεριφορά των ιστοσελίδων HTML. Αναλόγως μπορεί να συμπεριληφθεί κώδικας php σε συνδυασμό με html και javascript.

Τα βασικά tags τα οποία είναι αναγκαία για κάθε σελίδα html παρουσιάζονται στον παρακάτω πίνακα.



Εικόνα 18: HTML βασικά tags

Tags	Επεξήγηση
<!DOCTYPE>	Η δήλωση του DOCTYPE πρέπει να είναι η πρώτη που γίνεται σε ένα έγγραφο HTML. Πρόκειται για μια εντολή στο πρόγραμμα περιήγησης που έχει να κάνει με την έκδοση της HTML στο έγγραφο.
<html>	Με την ετικέτα <html> αρχίζει πάντα ο κώδικας και με την ετικέτα </html> τον τερματίζεται. Έτσι πληροφορούμε τον browser ότι υπάρχει κώδικας γραμμένος σε γλώσσα HTML.
<head>	Αποτελούν τον πρόλογο για την HTML σελίδα. Είναι βασική ετικέτα αφού στο πεδίο της ορίζονται δεδομένα όπως το είδος των χαρακτήρων που χρησιμοποιούνται, εξωτερικές πηγές αρχείων κ.α.
<title>	Περιλαμβάνεται στην ενότητα <head> και καθορίζει τον τίτλο της σελίδας ο οποίος εμφανίζεται στο πάνω μέρος του παραθύρου του web browser.
<body>	Ορίζει το κυρίως περιεχόμενο της σελίδας μέσα στο οποίο γράφουμε το κείμενο που θέλουμε να εμφανιστεί μαζί με τις HTML ετικέτες που το μορφοποιούν. Στην ενότητα αυτή τοποθετούμε επίσης εικόνες, video κ.α.
<div>	Καθορίζει ένα τμήμα σε ένα έγγραφο HTML. Χρησιμοποιείται για την ομαδοποίηση των στοιχείων, ώστε να δοθούν κοινές μορφοποιήσεις, με τη χρήση του id.
<h1>- <h6>	Οι HTML επικεφαλίδες είναι κείμενο που εμφανίζετε με μεγάλα και έντονα γράμματα. Ορίζονται από τις ετικέτες <h1> έως <h6>. Με την <h1> ορίζουμε την μεγαλύτερη ετικέτα ενώ με την <h6> την μικρότερη.
<p>	Οι παράγραφοι ορίζονται από το ζευγάρι ετικετών <p> και </p>.

	Η ετικέτα χρησιμοποιείται όταν μια γραμμή πρέπει να τελειώσει μέσα σε ένα κείμενο και να αρχίσει μια καινούργια.
<hr>	Η οριζόντια γραμμή είναι ένα απλό γραφικό που μπορεί να χρησιμοποιηθεί στην σελίδα κυρίως σαν διαχωριστικό. Με την ετικέτα <hr> τοποθετείται μια οριζόντια γραμμή στην σελίδα.
<table>	Ορίζει έναν πίνακα. Το ζεύγος των ετικετών <tr>...</tr> ορίζει μια γραμμή του πίνακα, το <td>...</td> αρχικοποιούν και τερματίζουν αντίστοιχα ένα κελί στην γραμμή του πίνακα και το <th> ορίζει μια επικεφαλίδα σε μια στήλη του πίνακα.
<form>	Για να δημιουργηθεί μια φόρμα χρησιμοποιούνται οι ετικέτες <form> και </form>. Οι κυριότερες ιδιότητες της ετικέτας είναι οι name, method και action.
<input>	Εισάγονται τα περισσότερα στοιχεία της φόρμας. Οι κυριότερες ιδιότητες της ετικέτας είναι η type η οποία καθορίζει τον τύπο του στοιχείου της Φόρμας (Πεδίο Κειμένου ή Περιοχή Κειμένου ή Κουμπί Επιλογών).
<pre>	Ορίζει κείμενο που είναι παραμετροποιημένο διατηρεί τα κενά και τις διαφορετικές γραμμές. Χρησιμοποιείτε κυρίως για την εισαγωγή κώδικα σε κείμενο.

Πίνακας 3: Tags HTML

1.3.3 CSS

Η CSS «Cascading Style Sheets-Διαδοχικά Φύλλα Στιλ» ή «αλληλουχία φύλλων στιλ» είναι μια γλώσσα υπολογιστή που ανήκει στην κατηγορία των γλωσσών φύλλων στιλ που χρησιμοποιείται για τον έλεγχο της εμφάνισης ενός εγγράφου που έχει γραφτεί με μια γλώσσα σήμανσης. Χρησιμοποιείται δηλαδή για τον έλεγχο της εμφάνισης ενός εγγράφου που γράφτηκε στις γλώσσες HTML και XHTML για τον έλεγχο της εμφάνισης μιας ιστοσελίδας και γενικότερα ενός ιστότοπου. Η CSS είναι προορισμένη να αναπτύσσει στιλιστικά μια ιστοσελίδα δηλαδή να διαμορφώνει περισσότερα χαρακτηριστικά, χρώματα, στοίχιση και δίνει περισσότερες δυνατότητες σε σχέση με την html. Για μια όμορφη και καλοσχεδιασμένη ιστοσελίδα η χρήση της CSS κρίνεται ως απαραίτητη.

Ένας κανόνας CSS αποτελείται από δύο κύρια μέρη. Το πρώτο μέρος είναι ο επιλογέας (selector) και το δεύτερο μέρος είναι οι δηλώσεις (declaration). Ο επιλογέας αφορά το στοιχείο το οποίο είναι προς μορφοποίηση και είναι είτε μία ετικέτα html είτε κάποιο συγκεκριμένο αντικείμενο html. Οι δηλώσεις αποτελούνται από τις ιδιότητες που χαρακτηρίζουν τον επιλογέα και τις τιμές που αυτές παίρνουν. Στην ενότητα όπου επεξηγείτε ο προγραμματισμός των ιστοσελίδων παρουσιάζονται τα αρχεία css που χρησιμοποιήθηκαν στην παρούσα εργασία. Η παρακάτω εικόνα αποδίδει γραφικά τον βασικό κανόνα σύνταξης CSS.[3]



Εικόνα 19: Βασική σύνταξη Css

1.3.4 PHP

Η PHP (Hypertext PreProcessor), είναι μια γλώσσα προγραμματισμού βασισμένη σε σενάρια που χρησιμοποιείται ευρέως και μπορεί να ενσωματωθεί στον κώδικα της HTML. Πρόκειται για μια γλώσσα προγραμματισμού με σκοπό τη δημιουργία σελίδων web με δυναμικό περιεχόμενο. Μια σελίδα PHP επεξεργάζεται από ένα συμβατό διακομιστή του Παγκόσμιου Ιστού (π.χ Apache), ώστε να παραχθεί σε πραγματικό χρόνο το τελικό περιεχόμενο, που θα σταλεί στο πρόγραμμα περιήγησης των επισκεπτών σε μορφή κώδικα HTML. Ενσωματώνοντας μικρά προγράμματα σεναρίων σε έναν HTML κώδικα, αποκτούμε τη δυνατότητα να ελέγχουμε τι θα εμφανίζεται στην οθόνη του περιηγητή με τρόπο πολύ πιο ευέλικτο από ότι με τη χρήση της HTML.[19]



Εικόνα 20 : Λογότυπο Php

Η γλώσσα αυτή εκτελείται στην πλευρά του εξυπηρετητή, κάτι που την κάνει να ξεχωρίζει από άλλες γλώσσες όπως η JavaScript, που σημαίνει ότι ο εξυπηρετητής ιστού τη διερμηνεύει προτού ακόμα τη στείλει στον browser. Σχεδιάστηκε με σκοπό την ανάπτυξη διαδικτυακών εφαρμογών και τη δημιουργία δυναμικού περιεχομένου ιστοσελίδων. Το πιο σημαντικό κομμάτι της, όμως είναι το γεγονός ότι υποστηρίζει ένα ευρύ σύνολο από βάσεις δεδομένων, κάτι που κάνει πολύ πιο εύκολη και απλή τη δημιουργία μιας ιστοσελίδας η οποία επικοινωνεί με βάσεις δεδομένων. Όπως αναφέρθηκε παραπάνω η γλώσσα php περιλαμβάνει κάποιες βασικές μεταβλητές οι οποίες έχουν σκοπό να βοηθούν στην ανάπτυξη διαδικτυακών δυναμικών εφαρμογών.

Έχει το ιδιαίτερο χαρακτηριστικό ότι ο κώδικάς της πρώτα μεταγλωττίζεται στον server και μετά φορτώνεται σαν ένα κανονικό html έγγραφο, χωρίς ο χρήστης να είναι σε θέση να δει τον αρχικό κώδικα. Η PHP είναι δωρεάν για χρήση, και είναι η δημοφιλέστερη για τους hosts Unix και Linux, αν και υπάρχουν εκδόσεις διαθέσιμες για τα Windows. Τέλος η php διαδραματίζει πολύ σημαντικό ρόλο στην εργασία που έχει δημιουργηθεί καθώς συμμετέχει στην σύνδεση και την αποστολή δεδομένων στη βάση αλλά και στην δημιουργία των γραφημάτων. Στον παρακάτω πίνακα παρουσιάζονται οι βασικές εντολές της γλώσσας.

Εντολή	Λειτουργία
\$_GET	Η προκαθορισμένη μεταβλητή <code>\$_GET</code> χρησιμοποιείται για τη συλλογή τιμών σε μια φόρμα (method= "GET"). Οι πληροφορίες που αποστέλλονται από μια φόρμα με τη μέθοδο GET είναι ορατές σε όλους, εμφανίζονται στη γραμμή διευθύνσεων του browser και επίσης υπάρχει όριο στην ποσότητα των πληροφοριών προς αποστολή.
\$_POST	Η προκαθορισμένη μεταβλητή <code>\$_POST</code>

	<p>χρησιμοποιείται για τη συλλογή τιμών σε μια φόρμα (method= "POST"). Οι πληροφορίες που αποστέλλονται από μια φόρμα με την μέθοδο POST είναι αόρατες στους άλλους και δεν έχει όρια για τον όγκο των πληροφοριών κατά την αποστολή .</p>
\$_REQUEST	<p>Περιέχει όλες τις μεταβλητές που στέλνονται μέσω των HTTP GET, HTTP POST και HTTP cookies. Αποτελεί στην ουσία το ισοδύναμο του συνδυασμού των \$_GET, \$_POST και \$_COOKIE και είναι λιγότερο επικίνδυνος από τη χρήση του \$GLOBALS. Όμως, καθώς περιέχει όλες τις μεταβλητές από μη έμπιστες πηγές, όπως είναι οι επισκέπτες της ιστοσελίδας μας, θα πρέπει να χρησιμοποιείται με επιφύλαξη.</p>
\$_COOKIE	<p>Ένα cookie είναι ένα μικρό αρχείο που ενσωματώνει ο server στον υπολογιστή του χρήστη. Η συνάρτηση setcookie() χρησιμοποιείται για να ορίσει ένα cookie.</p>
\$_SESSION	<p>Μια μεταβλητή συνεδρίας χρησιμοποιείται για την αποθήκευση πληροφοριών, ή για την αλλαγή των ρυθμίσεων για μια συνεδρία χρήστη. Οι μεταβλητές συνεδρίας είναι διαθέσιμες σε όλες τις σελίδες σε μια εφαρμογή. Κατά την ενασχόληση με μια εφαρμογή μπορούν να εκτελεστούν ενέργειες και αλλαγές δεδομένων. Η κατάσταση αυτή είναι μια συνεδρία. Μια συνεδρία PHP, επιτρέπει την αποθήκευση πληροφοριών του χρήστη στο διακομιστή για μελλοντική χρήση όπως όνομα, στοιχεία αγορών, κλπ. Ωστόσο, οι πληροφορίες συνεδρίας είναι προσωρινές και θα διαγράφονται αφού ο χρήστης έχει αποχωρήσει από το δικτυακό τόπο .Για μόνιμη αποθήκευση μπορεί να χρησιμοποιηθεί μια βάση δεδομένων.</p>

Πίνακας 4: Βασικές εντολές Php

1.3.5 My SQL

Η MySQL είναι ένα πολύ γρήγορο σε απόδοση και ισχυρό σε δυνατότητες υλοποίησης, σύστημα διαχείρισης σχεσιακών βάσεων δεδομένων .Αποτελεί τον βασικό άξονα στην πτυχιακή εργασία καθώς είναι το σημείο όπου αποθηκεύονται όλα τα δεδομένα που παρέχει το arduino και βασισμένος στη βάση είναι ο τρόπος που δημιουργείται ο χάρτης της google.

Το πρόγραμμα χρησιμοποιεί έναν εξυπηρετητή (server) παρέχοντας πρόσβαση πολλών χρηστών σε ένα σύνολο βάσεων δεδομένων. Ο κωδικός του εγχειρήματος είναι διαθέσιμος μέσω της GNU General Public License , καθώς και μέσω ορισμένων ιδιόκτητων συμφωνιών. Ανήκει και χρηματοδοτείται από μία και μοναδική κερδοσκοπική εταιρία, τη σουηδική MySQL AB , η οποία σήμερα ανήκει στην Oracle.

Τα δεδομένα στη MySQL αποθηκεύονται σε αντικείμενα βάσης δεδομένων, τα οποία ονομάζονται πίνακες. Ένας πίνακας, είναι μια συλλογή από σχετικές καταχωρήσεις δεδομένων και αποτελείται από στήλες και γραμμές. Οι βάσεις δεδομένων είναι χρήσιμες για την αποθήκευση πληροφοριών σε κατηγορίες. Η διαχείριση των δεδομένων στις βάσεις, γίνονται μέσω των SQL εντολών, στη MySQL. Η πιο σημαντική ίσως κατηγορία εντολών που χρησιμοποιείται είναι τα queries. Ένα query είναι ένα ερώτημα ή ένα αίτημα. Με τη MySQL, επιτρέπεται η διερεύνηση σε μια βάση δεδομένων και η επιστροφή ζητηθέντων πληροφοριών. Η MySQL είναι η δημοφιλέστερη βάση δεδομένων για διαδικτυακά προγράμματα και ιστοσελίδες. Χρησιμοποιείται σε κάποιες από τις πιο διαδεδομένες διαδικτυακές υπηρεσίες.[21]



1.4 Εργαλεία Ανάπτυξης Λογισμικού

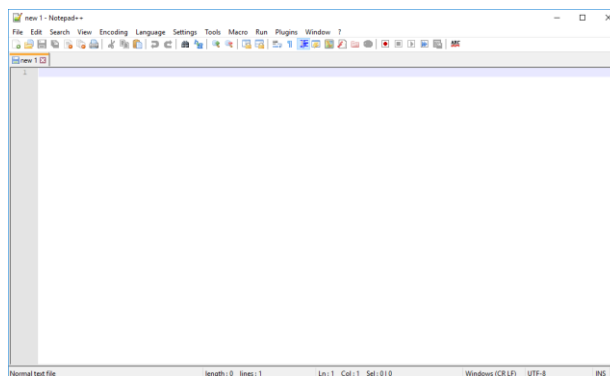
Για την υλοποίηση της εργασίας χρησιμοποιήθηκαν κάποια χρήσιμα εργαλεία, αυτά θα αναλυθούν στην παρακάτω ενότητα. Για τη συγγραφή του κώδικα που αφορά τον προγραμματισμό και την κατασκευή του ιστοχώρου χρησιμοποιήθηκε το notepad++ και συγχρόνως έγινε χρήση του τοπικού διακομιστή Xampp.Όσον αφορά τον προγραμματισμό του arduino χρησιμοποιήθηκε το λογισμικό arduino IDE για την φόρτωση του κώδικα. Η σχεδίαση κυκλωμάτων σε εικόνες πραγματοποιήθηκε με το πρόγραμμα Fritzing.

1.4.1 Notepad++

Πρόκειται για ανοικτού λογισμικού πρόγραμμα το οποίο χρησιμοποιείται για συγγραφή κώδικα Πρόκειται για τον πιο γνωστό επεξεργαστή κειμένου (text editor). Είναι απαραίτητο εργαλείο συγγραφής των προγραμματιστών, καθώς αποτελεί ένα ελαφρύ πρόγραμμα το οποίο όμως έχει πάρα πολλές δυνατότητες. Το περιβάλλον εργασίας είναι εξαιρετικά απλό και λειτουργικό. Στο επάνω μέρος του υπάρχει μία εργαλειοθήκη με τις κυριότερες λειτουργίες. Όλο το υπόλοιπο μέρος του καταλαμβάνεται από τον χώρο συγγραφής του κώδικα. Γύρω από τον χώρο αυτόν, μπορούν να εμφανιστούν διάφορα πλαίσια τα οποία παρουσιάζουν διάφορες πληροφορίες ανάλογα με τα plugins από τα οποία προέρχονται. Το σημαντικό του χαρακτηριστικό είναι το λεγόμενο syntax highlighting, το οποίο χρωματίζει τον κώδικα ανάλογα με την δομή του και διευκολύνει την ανάπτυξη του. Στην παρούσα εργασία χρησιμοποιήθηκε για τη συγγραφή όλων των script σε php, javascript, html, css, ajax και mysql.[14]



Εικόνα 22 : Λογότυπο Notepad++



Εικόνα 23 : Καινούργιο αρχείο Notepad++

1.4.2 Xampp

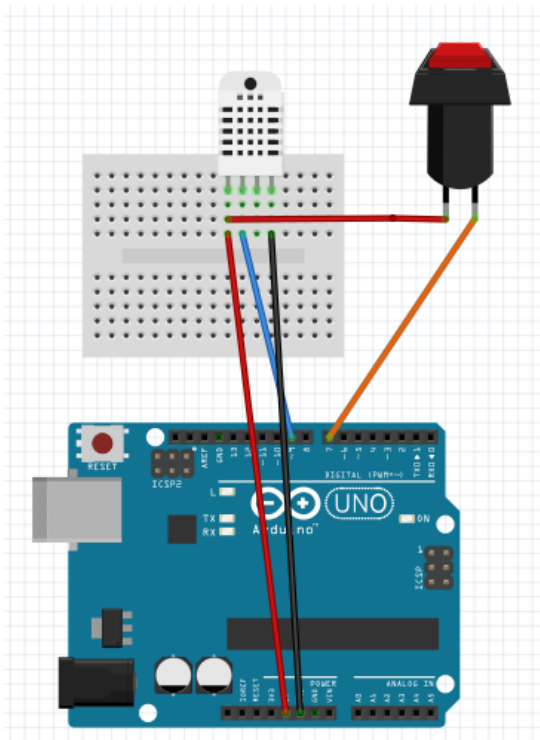
Το XAMPP είναι ένα πακέτο προγραμμάτων ελεύθερου λογισμικού το οποίο περιέχει τον εξυπηρετητή ιστοσελίδων HTTP Apache, σύστημα διαχείρισης βάσεων δεδομένων MySQL και ένα διερμηνέα για PHP και Perl. Στην παρούσα διπλωματική χρησιμοποιήθηκε για να προσομοιώσει έναν υπολογιστή εξυπηρετητή (server), ο οποίος φιλοξενεί τον ιστοχώρο της πτυχιακής.[15]



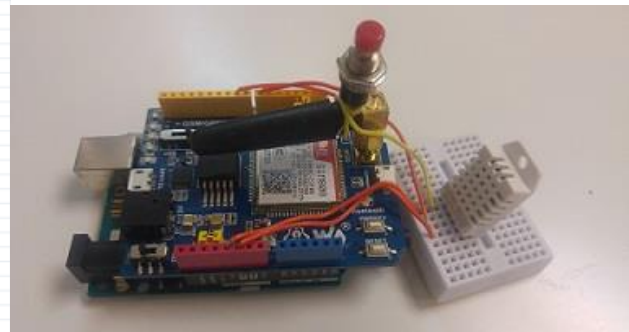
Εικόνα 24 : Λογότυπα υποστηριζόμενα από το Xampp

1.4.3 Fritzing

Το Fritzing, είναι ένα εύκολο στη χρήση λογισμικό ηλεκτρονικού σχεδιασμού ανοικτού κώδικα που έχει σχεδιαστεί για να βοηθήσει στη μετάβαση από ένα πρωτότυπο σε ένα τελικό σχέδιο. Απευθύνεται σε χρήστες που θέλουν να παράγουν έγγραφα ή να τα χωρίσουν σε κατηγορίες και να πειραματιστούν. Έχει τη δυνατότητα να δημιουργίας σχηματικών PCB αλλά και παράγωγα αρχεία PCB. Το Fritzing λοιπόν, είναι μια εφαρμογή που μπορεί να συνδυαστεί με τη τεχνολογία του μέλλοντος, το Arduino ώστε να δημιουργηθούν τ' απαραίτητα κυκλώματα που τις περισσότερες φορές είναι σχετικά απλά ώστε να ελέγχουμε ηλεκτρικές συσκευές από ένα pc. Παρακάτω θα παραθέσουμε την εικόνα του σχεδίου που έχει δημιουργηθεί για λογαριασμό της πτυχιακής εργασίας, βέβαια δίχως την πλακέτα sim808 καθώς δεν συμπεριλαμβάνεται στο λογισμικό. Τέλος η ολοκληρωμένη φυσική εικόνα του συστήματος φαίνεται παρακάτω όπου η μόνη αλλαγή σε σχέση με το σχηματικό είναι ότι η πλακέτα sim808 έχει τοποθετηθεί ακριβώς πάνω στο arduino υπο και μεταφέρει όλες τις ακίδες προς χρησιμοποίηση από τον χρήστη.



Εικόνα 25 : Σχέδιο Fritzing

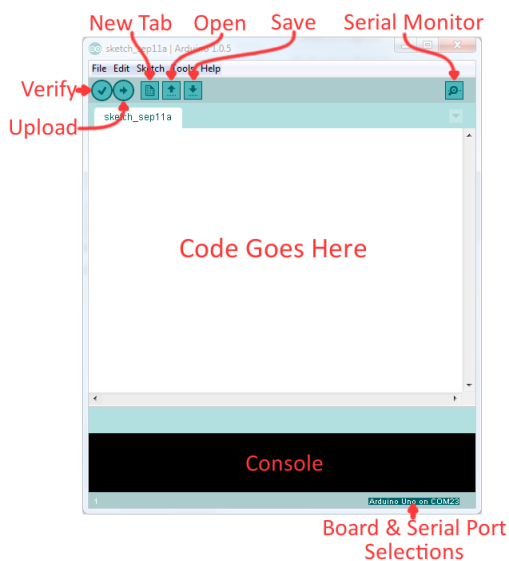


Εικόνα 26 : Εμφάνιση Συστήματος

1.4.4 Arduino Ide

Το Arduino Uno έχει τη δυνατότητα επικοινωνίας με υπολογιστή, άλλες ίδιες πλακέτες ή άλλους μικροελεγκτές. Παρέχεται η δυνατότητα σειριακής επικοινωνίας μέσω USB θύρας και τους οδηγούς (drivers). Το λογισμικό που παρέχεται επιτρέπει στον χρήστη να στέλνει εντολές ή δεδομένα γενικά από και προς την πλακέτα και να φορτώνει το πρόγραμμα στον bootloader, το οποίο θα εκτελεί αυτόματα η πλακέτα. Για τη διαχείριση του Arduino από τον υπολογιστή χρησιμοποιείται το Arduino IDE.[8]

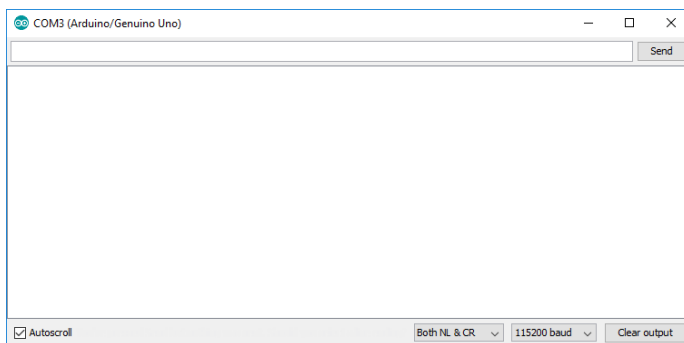
Το Arduino IDE είναι ένα περιβάλλον ανάπτυξης το οποίο περιέχει μια περιοχή επεξεργασίας κειμένου για τη συγγραφή κώδικα, μια περιοχή μηνυμάτων, ένα μενού, μια γραμμή εργαλείων με κουμπιά για κοινές λειτουργίες, καθώς και μια σειρά από μενού. Συνδέεται με το υλικό Arduino για τη φόρτωση προγραμμάτων και για να επικοινωνούν μεταξύ τους. Ένα ολοκληρωμένο πρόγραμμα συνήθως ονομάζεται sketch. Αυτό το sketch είναι γραμμένο με το πρόγραμμα επεξεργασίας κειμένου. Η κονσόλα απεικονίζει την έξοδο του κειμένου από το περιβάλλον Arduino συμπεριλαμβάνοντας πλήρη μηνύματα λάθους και άλλες πληροφορίες. Τα κουμπιά της γραμμής εργαλείων επιτρέπουν τον έλεγχο και το ανέβασμα των προγραμμάτων, τη δημιουργία νέου sketch, το άνοιγμα και την αποθήκευση των sketch και άνοιγμα της σειριακής οθόνης. Παρακάτω παρουσιάζεται το περιβάλλον του ide και επεξηγούνται οι περιοχές και τα κουμπιά του λογισμικού.



Εικόνα 27 : Περιβάλλον IDE

Σειριακή οθόνη

Εμφανίζει τα σειριακά δεδομένα που αποστέλλονται από την πλακέτα Arduino. Πιο συγκεκριμένα, η αποστολή δεδομένων στην πλακέτα γίνεται, εισάγοντας κείμενο και πατώντας το κουμπί send ή πατώντας το Enter. Επίσης, στο κάτω μέρος της σειριακής οθόνης, μπορεί να γίνει η επιλογή της κατάλληλης ταχύτητας (baud) από την λίστα που εμφανίζεται ανάλογα με την τιμή που θα επιλεγεί στο προγραμματισμό του Arduino με το Serial.begin(). Παρακάτω παρουσιάζεται η σειριακή οθόνη (serial monitor).



Εικόνα 28: Σειριακή ide

Συνοπτικά το Arduino IDE παρέχει:

- Ένα πρακτικό περιβάλλον για τη συγγραφή των προγραμμάτων, με συντακτική χρωματική σήμανση ανάλογα τον τύπο της εντολής.
- Μερικές έτοιμες βιβλιοθήκες αλλά και πολλές διαθέσιμες για λήψη από το διαδίκτυο.
- Τον compiler για τη μεταγλώττιση των sketch.
- Μία σειριακή οθόνη (serial monitor) που παρακολουθεί τις επικοινωνίες της σειριακής (USB), αναλαμβάνει να στείλει αλφαριθμητικά στο Arduino μέσω αυτής και είναι ιδιαίτερα χρήσιμο για την αποσφαλμάτωση των sketch.
- Την επιλογή για ανέβασμα των μεταγλωττισμένων sketch στο Arduino.

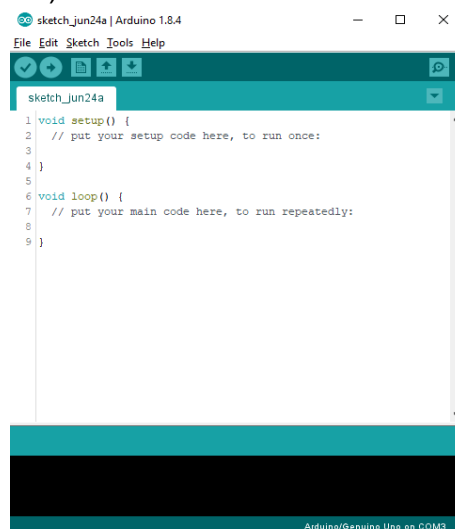
1.4.5 Γλώσσα και δομή προγράμματος

Η γλώσσα του Arduino βασίζεται στη γλώσσα Wiring μια παραλλαγή C/C++ για μικροελεγκτές αρχιτεκτονικής AVR όπως ο ATmega, και υποστηρίζει όλες τις βασικές δομές της C καθώς και μερικά χαρακτηριστικά της C++. Λόγω της καταγωγής της από τη C, στη γλώσσα του Arduino, μπορούν να χρησιμοποιηθούν ουσιαστικά οι ίδιες βασικές εντολές και συναρτήσεις, με την ίδια σύνταξη, τους ίδιους τύπων δεδομένων και τους ίδιους τελεστές όπως και στη C. . Πέρα από αυτές όμως, υπάρχουν κάποιες ειδικές εντολές, συναρτήσεις και σταθερές που βοηθούν για τη διαχείριση του ειδικού hardware του Arduino.

Τα αρχεία αυτά έχουν επέκταση INO. Τα σχόλια μιας γραμμής γράφονται μετά από διπλές καθέτους '//'. Σχόλια περισσότερων γραμμών γράφονται εντός '/'* και '*/'. (/* σχόλιο */) Το IDE αναγνωρίζει τα σχόλια και τα χρωματίζει γκριζα, για να δείξει ότι δεν είναι ενεργά στο πρόγραμμα ή στο debugging.

Τα προγράμματα του Arduino διαιρούνται σε τρία μέρη:

- Δομή (structure)
- Τιμές (values)
- Συναρτήσεις (functions)



```
sketch_jun24a | Arduino 1.8.4
File Edit Sketch Tools Help
sketch_jun24a
1 void setup() {
2   // put your setup code here, to run once:
3
4 }
5
6 void loop() {
7   // put your main code here, to run repeatedly:
8
9 }
```

Arduino/Genuino Uno on COM3

Εικόνα 29: Αρχική εμφάνιση ide

Υπάρχουν δυο ειδικές συναρτήσεις που είναι μέρος του κάθε sketch του Arduino οι οποίες είναι η setup() και η loop(). Η setup() καλείται μια φορά, όταν το sketch ξεκινά ή όποτε κάνει επαναφορά (reset) η πλατφόρμα Arduino. Κυρίως, σε αυτήν γίνονται οι αρχικοποιήσεις των μεταβλητών, η ρύθμιση της κατάστασης των ακίδων (pins) και η προετοιμασία των βιβλιοθηκών. Αντιθέτως, η συνάρτηση loop() καλείται ξανά και ξανά επιτρέποντας έτσι στο πρόγραμμα να ανταποκριθεί σε εξωτερικά ερεθίσματα συνεχόμενα μέχρι να το τερματίσει ο χρήστης.

Οι βασικές εντολές και συναρτήσεις του ide παρουσιάζονται στον παρακάτω πίνακα.

Δομή.	
setup(), loop()	
Δομές ελέγχου ροής.	
if	Δομή ελέγχου μίας συνθήκης.
if..else	Δομή ελέγχου πολλαπλών συνθηκών.
for	Δομή επαναληπτικού ελέγχου συνθήκης.
do...while	Δομή επαναληπτικού ελέγχου συνθήκης.
while	Δομή επαναληπτικού ελέγχου συνθήκης.
switch case	Δομή ελέγχου περιπτώσεων.
break	Εντολή διακοπής μιας επαναληπτικής δομής.
continue	Εντολή παράλειψης της τρέχουσας επανάληψης.
return	Εντολή επιστροφής από μία συνάρτηση.
go to	Εντολή μετάβασης σε κάποιο σημείο του κώδικα.
Αριθμητικοί τελεστές.	
=, +, -, *, /	Τελεστής εκχώρησης, πρόσθεσης, αφαίρεσης, διαίρεσης.
%	Υπόλοιπο ακέραια διαίρεσης.
Τελεστές σύγκρισης	
<, >, <=, >=	Μικρότερο, Μεγαλύτερο, Μικρότερο ή ίσο, Μεγαλύτερο ή ίσο.
==, !=	Ισότητα, Ανισότητα.
Λογικοί τελεστές.	
&&, , !	Λογική σύζευξη, διάζευξη, άρνηση.
Δυαδικοί τελεστές.	
&, 	Δυαδική σύζευξη – διάζευξη.
<<, >>	Δυαδική αριστερή – δεξιά ολίσθηση.
^	Δυαδική αποκλειστική διάζευξη.
~	Δυαδική άρνηση.
Τελεστές αύξησης και μείωσης.	
++, --	Αύξηση – μείωση κατά μία ακέραια μονάδα.

Σύνθετοι τελεστές.	
+=, -=, *=, /=, %=	Σύνθετοι αριθμητικοί τελεστές.
&=, =, ^=, ~=, <<=, >>=	Σύνθετοι δυαδικοί τελεστές.
Τελεστές δεικτών.	

*, &	Τελεστής απόκτησης περιεχομένου – διεύθυνσης.
Τύποι δεδομένων.	
boolean	Λογική δυαδική τιμή.
char	Προσημασμένος χαρακτήρας 8 ψηφίων.
(unsigned)char	Μη προσημασμένος χαρακτήρας 8 ψηφίων.
byte	Μη προσημασμένος χαρακτήρας 8 ψηφίων.
int	Προσημασμένος ακέραιος 16 ψηφίων.
(unsigned)int	Μη προσημασμένος ακέραιος 16 ψηφίων.
word	Μη προσημασμένος ακέραιος 16 ψηφίων.
long	Προσημασμένος ακέραιος 32 ψηφίων.
(unsigned)long	Μη προσημασμένος ακέραιος 32 ψηφίων.
float, double	Αριθμός κινητής υποδιαστολής απλής ακρίβειας.
string	Αντικείμενο αλφαριθμητικού με χρήσιμες μεθόδους.
Σταθερές.	
HIGH – LOW	Τιμή υψηλής - χαμηλής στάθμης για μία επαφή εισόδου.
INPUT – OUTPUT	Χρησιμοποιείται για τον ορισμό μίας επαφής ως είσοδο – έξοδο.
True-false	Λογικό επίπεδο αλήθειας - ψεύδους σε μία συνθήκη.
A0,...,A5	Συμβολοσταθερές για τις αναλογικές επαφές εισόδου.
Συναρτήσεις εισόδου και εξόδου.	
pinMode()	Ορίζει μια επαφή ως είσοδο ή έξοδο.
Συναρτήσεις αναλογικής εισόδου και εξόδου.	
analogReference()	Ορίζει την τάση αναλογικής αναφοράς.
analogRead()	Διαβάζει από μία αναλογική επαφή εισόδου.
analogWrite()	Γράφει PWM σήματα σε μία επαφή εξόδου.
Συναρτήσεις ψηφιακής εισόδου και εξόδου.	
digitalWrite()	Γράφει σε μία ψηφιακή επαφή εξόδου.
digitalRead()	Διαβάζει από μία ψηφιακή επαφή εισόδου.
Συναρτήσεις Χρόνου.	
millis(), micros()	Διάρκεια εκτέλεσης του προγράμματος σε ms, μs.
delay	Παύση προγράμματος - η διάρκεια δίδεται σε ms.
delayMicroseconds()	Παύση προγράμματος - η διάρκεια δίδεται σε μs.
Προηγμένες συναρτήσεις εισόδου και εξόδου.	
tone(),	Παράγει ένα τετραγωνικό σήμα ορισμένης συχνότητας.
noTone()	Διακόπτει την παραγωγή τετραγωνικών σημάτων.
shiftOut()	Ολισθαίνει τα ψηφία μιας τιμής σε μία επαφή εξόδου.
pulseIn()	Επιστρέφει την διάρκεια σε μs ενός παλμού HIGH ή LOW.
Μαθηματικές και τριγωνομετρικές συναρτήσεις.	
max(), min()	Βρίσκει τον μεγαλύτερο - μικρότερο ανάμεσα σε δύο αριθμούς.
abs()	Επιστρέφει την απόλυτη τιμή ενός αριθμού..
constrain()	Ελέγχει για υπερχείλιση ή υποχείλιση ορίων.
map()	Πραγματοποιεί γραμμικό μετασχηματισμό ορίων.

pow()	Επιστρέφει το αποτέλεσμα μίας δύναμης.
sqrt()	Επιστρέφει την ρίζα ενός αριθμού.
sin(), cos(), tan()	Υπολογίζει το ημίτονο, συνημίτονο , εφαπτομένη ενός αριθμού.
Συναρτήσεις επεξεργασίας δυαδικών αριθμών.	
lowByte(),highByte()	Επιστρέφει το δεξιότερο - αριστερότερο byte μίας μεταβλητής.
bitRead(),bitWrite()	Διαβάζει - γράφει ένα συγκεκριμένο ψηφίο μίας μεταβλητής.
bitSet(),bitClear()	Γράφει την τιμή 1 - 0 σε κάποιο ψηφίο μίας μεταβλητής.
bit()	Υπολογίζει μία συγκεκριμένη δύναμη με βάση το 2.
Συναρτήσεις χρήσης ρουτινών εξυπηρέτησης διακοπών.	
attachInterrupt()	Ενεργοποιεί μία ρουτίνα εξυπηρέτησης διακοπής.
detachInterrupt()	Απενεργοποιεί μία ρουτίνα εξυπηρέτησης διακοπής.
Συναρτήσεις ενεργοποίησης και απενεργοποίησης διακοπών.	
interrupts()	Ενεργοποιεί τα σήματα διακοπής.
noInterrupts()	Απενεργοποιεί τα σήματα διακοπής.
Υποστήριξη σειριακής επικοινωνίας.	
Serial	Αντικείμενο σειριακής επικοινωνίας με χρήσιμες μεθόδους.
Συναρτήσεις μετατροπής τύπων.	
char(), byte(), int(), word(), long(), float(), double()	

Πίνακας 5: Λειτουργίες και συναρτήσεις Arduino Ide

Κεφάλαιο 2 : Υλικό μέρος συστήματος

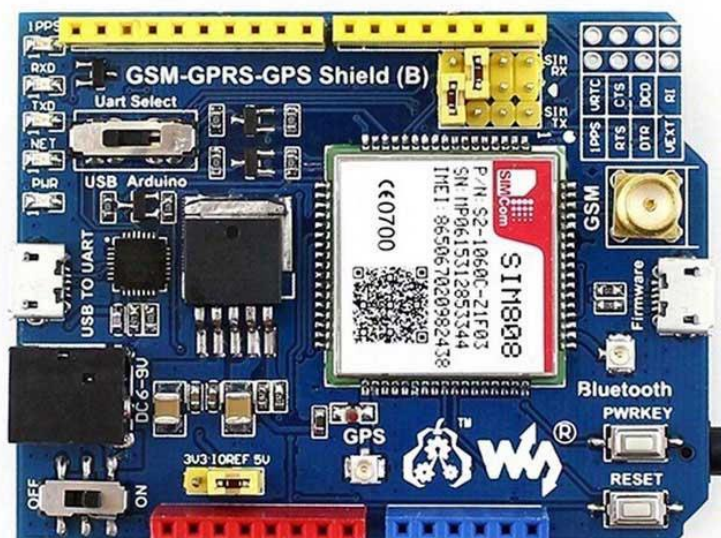
Υλοποίηση του συστήματος βασίστηκε στον μικροελεγκτή Arduino Uno, τα τεχνικά χαρακτηριστικά του οποίου αναλύθηκαν στο προηγούμενο κεφάλαιο. Πάνω στο Arduino συνδέθηκαν όλα τα περιφερειακά κυκλώματα και οι συσκευές που απαιτούνται όπως η πλακέτα sim808, ο αισθητήρας θερμοκρασίας/υγρασίας και το κουμπί «πανικού». Επίσης θα αναφερθούν σε αυτό το κεφάλαιο το ποτενσιόμετρο που χρησιμοποιήθηκε για εξομοίωση του αισθητήρα καρδιακών παλμών αλλά και το επιταχυνσιόμετρο που μετατράπηκε σε μετρητής βημάτων. Τέλος παρουσιάζεται το κύκλωμα του βασικού συστήματος χωρίς την πλακέτα sim 808 καθώς δεν υπάρχει στο περιβάλλον fritzing.



Εικόνα 30: Κύκλωμα βασικού συστήματος.

2.1 Μονάδα Sim808

Η μονάδα SIM808 σχεδιασμένη για την παγκόσμια αγορά, είναι ενσωματωμένη σε μία πλακέτα που περιέχει τις τεχνολογίες GSM / GPRS/GPS και την λειτουργία BT(Bluetooth). Οι τεχνολογίες GSM / GPRS αποτελούνται από μια μονάδα τετραπλής ζώνης που λειτουργεί σε συχνότητες GSM 850MHz, EGSM 900MHz, DCS 1800MHz και PCS 1900MHz. Το SIM808 διαθέτει GPRS πολλαπλών υποδοχών κατηγορίας 12, κλάσης 10 και υποστηρίζει τα προγράμματα κωδικοποίησης GPRS CS-1, CS-2, CS-3 και CS-4. Το μοντέλο που έχει χρησιμοποιηθεί αποτελεί διαμόρφωση της εταιρίας Waveshare.[9]



Εικόνα 31: Πλακέτα με τον Sim808

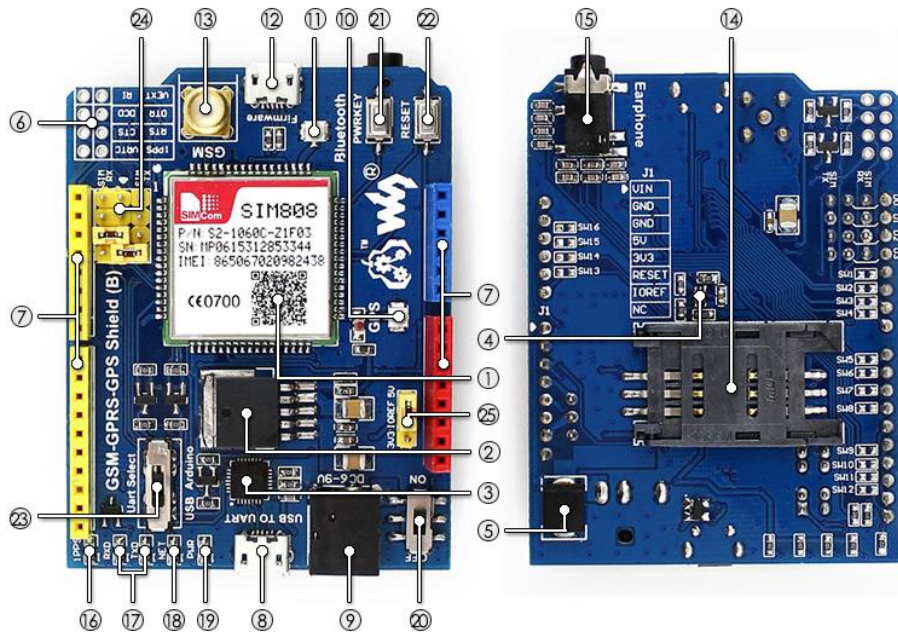
Οι διασυνδέσεις υλικού της μονάδας είναι οι εξής:

- Υποστήριξη 4 *4*2 ηλεκτρολογίων.
- Μια σειριακή θύρα (micro usb)

- Ένα USB, οι διασυνδέσεις USB μπορούν να εντοπίσουν σφάλματα, να μεταφορτώσουν λογισμικό(firmware).
- Τα κανάλια ήχου που περιλαμβάνουν είσοδο μικροφώνου και έξοδο δέκτη.
- Μια διεπαφή κάρτας SIM που υποστηρίζει κάρτα 3.3 Volts.
- Διεπαφή φόρτισης.
- Προγραμματιζόμενες εισοδοι και έξοδοι γενικού σκοπού.
- Υποστήριξη της λειτουργίας Bluetooth.
- Υποστήριξη PWM και ADC.
- Διεπαφή κάρτας PCM/SPI/SD, μόνο μία λειτουργία είναι προσβάσιμη συγχρονισμένα(Η προεπιλεγμένη λειτουργία είναι PCM).

Στο παρακάτω σχήμα θα παρουσιαστούν όλες οι βασικές λειτουργίες όπως :

- Σύστημα GSM
- Σύστημα GPS
- Σύστημα GPRS
- Ενδεικτικά LED
- Τη μονάδα ρύθμισης της συχνότητας του GSM
- Διασύνδεση κεραίας
- Άλλες διασυνδέσεις

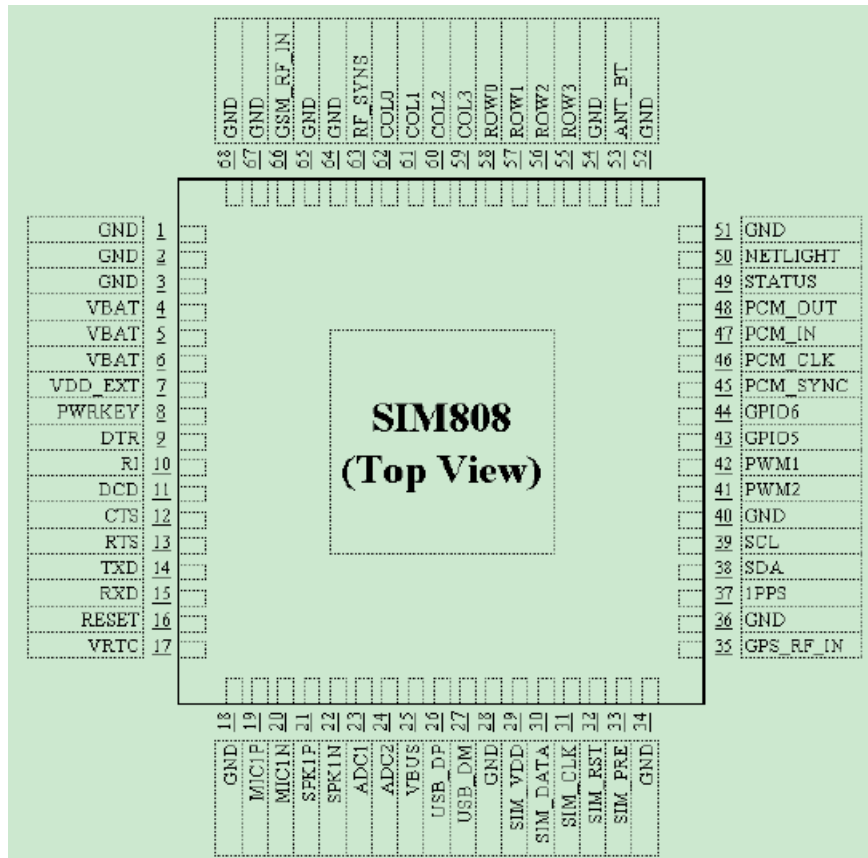


Εικόνα 32: Ανάλυση επιμέρους στοιχείων του Sim808

Τα βασικά στοιχεία της επιφάνειας της πλακέτας είναι:

1. SIM808 :Κεντρικός επεξεργαστής

2. MIC29302 power chip.
3. CP2102: USB TO UART converter.
4. SMF05C: δίοδος TVS.
5. 1N5408: ανορθωτής.
6. Υποδεικνύει την λειτουργία της κάθε ακίδας με κίτρινο χρώμα στα αριστερά του.
7. Λειτουργικές ακίδες που μεταφέρονται από την πλακέτα που είναι συνδεδεμένη κάτω από το sim808.
8. USB σε UART επικοινωνία.
9. Σημείο τροφοδοσίας συνεχούς ρεύματος.
10. Σημείο σύνδεσης gps κεραίας.
11. Σημείο σύνδεσης bluetooth κεραίας.
12. Υποδοχή αναβάθμισης firmware.
13. Σημείο σύνδεσης gsm κεραίας.
14. Σημείο εισαγωγής κάρτας SIM.
15. Σημείο εισαγωγής ακουστικών ή μικροφώνου 3.5mm.
16. Led που κάνει γνωστή τη λειτουργία του GPS.
17. Led που κάνει γνωστή την κατάσταση στα pin ανάγνωσης και μετάδοσης δεδομένων tx/rx.
18. NET led : ενεργοποιείτε όταν πατηθεί το νούμερο 8 και αναβοσβήνει γρήγορα όταν δεν έχει φορτωθεί κάποιο πρόγραμμα, ενώ αναβοσβήνει ανά 3 δεύτερα όταν εκτελεί κάποιο πρόγραμμα.
19. Power led
20. On/off διακόπτης
21. Κουμπί ενεργοποίησης
22. Κουμπί επανεκκίνησης
23. Διακόπτης που ορίζει το είδος της επικοινωνίας. Δεξιά για σύνδεση με το arduino και άλλες πλακέτες και αριστερά για σειριακή επικοινωνία .
24. Ακίδες μετάδοσης και ανάγνωσης δεδομένων.
25. Επιλογέας volt λειτουργίας σειριακής.



Εικόνα 33: Pinout διάγραμμα του Sim808

2.1.1 AT-Commands

Η μονάδα sim808 ελέγχεται από τις λεγόμενες AT-Commands. Οι εντολές αποτελούνται από μια σε μικρή σειρά κειμένου που μπορούν να συνδυαστούν για την παραγωγή εντολών και πραγματοποιούν όλες τις λειτουργίες που προαναφέρθηκαν. Η συντριπτική πλειοψηφία των πλακετών simcom υποστηρίζει τέτοιου είδους εντολές. Αφού τροφοδοτηθεί η μονάδα με παροχή 9-12 Volts και άνω των 2 amperes γίνεται δυνατό να ξεκινήσει η εκτέλεση AT-Commands. Στην παρακάτω λίστα θα αναφερθούν οι πιο χρήσιμες εντολές για την συγκεκριμένη εργασία.[12]



Εικόνα 34: Μπροστά και πίσω όψη του Sim808

2.2.2 AT-Commands GSM επικοινωνίας

Στον παρακάτω πίνακα παρουσιάζονται οι εντολές που αφορούν το δίκτυο κινητής τηλεφωνίας και λειτουργίες όπως εκτέλεση κλήσης και αποστολή μηνύματος. Για να εκτελεστούν κάποιες εντολές πρέπει να έχουν προηγηθεί κάποιες άλλες άρα διαδραματίζει μεγάλο ρόλο η σωστή αλληλουχία. Επίσης σχεδόν όλες οι εντολές δέχονται στο τέλος τους «=1» για να εκτελεστούν και «=0» για να σταματήσουν κάποια λειτουργία.[13]

Γενικές εντολές κάρτας σιμ.

Εντολή	Περιγραφή	Απάντηση	Επεξήγηση
AT+CPIN?	Ελέγχει την κατάσταση της κάρτας σιμ.	+CPIN:READY	Βρέθηκε η κάρτα σιμ
AT+CSQ	Ελέγχει την ποιότητα του σήματος	+CSQ:30,0	Μέγιστο επίπεδο είναι το 31 ικανό για τις βασικές λειτουργίες είναι το 10.
AT+COPS?	Ελέγχει τον πάροχο κινητής τηλεφωνίας	+COPS:0,0,"VODAFONE" or empty	Όνομα παρόχου
AT+CGMI	Ελέγχει τον δημιουργό της μονάδας.	SIMCOM_Ltd	Όνομα Κατασκευαστή
AT+CGMM	Ελέγχει το είδος της μονάδας	SIMCOM_SIM808	Στην περίπτωση μας SIM808
AT+CGSN	Ελέγχει τον μοναδικό αριθμό imei	858830200504041	Αριθμός κάρτας σιμ.

AT+CNUM	Ελέγχει το νούμερο της κάρτας σιμ.	+CNUM:,"6949465421"	Αριθμός κάρτας
AT+ATE1	Ελέγχει αν η μονάδα λειτουργεί ή όχι	ATE0 ή ATE1	ON/OFF
ATD+number;	Καλεί το νούμερο που έχει δηλωθεί	OK	Κλήση
ATA	Απαντά σε κλήση	OK	-
ATH	Τερματίζει την τρέχουσα κλήση	OK	-
AT+COLP	Εμφανίζει τον αριθμό που καλεί η μονάδα	+COLP:"-----"	-
AT+CLIP(=1)	Εμφανίζει τον αριθμό που καλεί τη μονάδα	+CLIP:"-----"	-
AT+VTS=*	Θέτει ένα νούμερο για μία συγκεκριμένη απάντηση όπως γίνεται για παράδειγμα στους αυτόματους τηλεφωνητές	-	-

Πίνακας 6: AT-Commands SIM κάρτας

Εντολές μηνύματος

Εντολή	Περιγραφή	Απάντηση	Επεξήγηση
AT+CNMI(2,1)	Καθορισμός θέσης αποθήκευσης μηνύματος	+CMTI:"SM",2	Όταν δεν είναι γεμάτη η θέση 2 αποθηκεύεται εκεί το μήνυμα
AT+CMGF(=1)	Ενεργοποιεί τη λειτουργία του μηνύματος	OK	-
AT+CSCS	Θέτει τον χαρακτήρα των μηνυμάτων ώστε να είναι στην αγγλική μορφή	OK	-
AT+CMGR(=1)	Διαβάζει το μήνυμα της πρώτης θέσης	-	-
AT+CMGS	Στέλνει το μήνυμα. » Πρέπει στο τέλος να προστεθεί το νούμερο 26 ώστε να καταλάβει η	">" «μήνυμα	-

	μονάδα ότι τερματίζεται το μήνυμα		
AT+CMGD	Διαγράφει το μήνυμα όποιας θέσης υποδειχτεί	OK	-
AT+CPMS?	Ελέγχει τον χώρο αποθήκευσης	+CPMS:"SM",1,50	Σημαίνει ότι υπάρχει χώρος για 50 μηνύματα και αυτή τη στιγμή είναι ένα αποθηκευμένο

Πίνακας 7: AT-Commands SMS

2.2.3 AT-Commands GPRS

Ακολουθούν οι εντολές που είναι υπεύθυνες για την σύνδεση της μονάδας με το διαδίκτυο.

Εντολή	Απάντηση	Επεξήγηση
AT+CPIN?	+CPIN:READY,OK	Ελέγχει αν ο αριθμός pin που εισήχθη δίνει πρόσβαση στην κάρτα sim
AT+CREG?	+CREG:0,1.OK	Ελέγχει αν η κάρτα είναι καταχωρημένη στο δίκτυο
AT+CGATT=1	+CGATT:1,OK	Συνδέεται με το δίκτυο κινητής τηλεφωνίας
AT+CSST='APN'	OK	Δηλώνεται το αρη(ξεχωριστό για κάθε εταιρία κινητής) αν υπάρχει κωδικός προστίθεται και αυτός.
AT+CIICR	OK	Ξεκινά τη σύνδεση
AT+CIFSR	-	IP που έχει δεσμευθεί από τη πλακέτα
AT_CIPSTART="TCP", "ιστότοπος", "80"	OK,CONNECT,OK	Δηλώνεται ποιος ιστότοπος είναι ο στόχος αλλά και η πόρτα σύνδεσης.
AT+CIPSEND > ---	SEND OK	Στέλνει δεδομένα

AT+CIPCLOSE=1	OK	Κλείνει την TCP σύνδεση
AT+CIPSTATUS	-	Ελέγχει την κατάσταση σύνδεσης της μονάδας
AT+CIPSHUT	-	Κλείνει την επικοινωνία GPRS

Πίνακας 8: AT-Commands GPRS

2.2.4 AT-Commands GPS

Ακολουθεί ο πίνακας με τις εντολές που χρειάζεται να εκτελεστούν ώστε η πλακέτα να δεχθεί τα δεδομένα της κεραίας του GPS.

Εντολή	Απάντηση	Επεξήγηση
AT+CGPSPWR=1	-	Ενεργοποιεί την κεραία του GPS
AT+CGPSRST=1	-	Επανεκκινεί την κεραία
AT+CGPSIPR=9600	-	Θέτει την ταχύτητα(baudrate) σε 9600bps
AT+CGPSINF=32	Nmea string ,OK	Διαβάζει τα δεδομένα του GPS σε μορφή nmea
AT+CGPSSTATUS?	+CGPSSATUS:LOCATION:””	Ελέγχει την κατάσταση του GPS

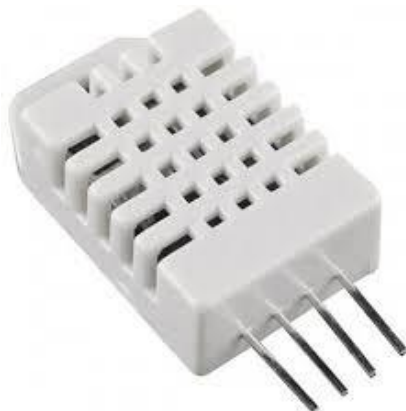
Πίνακας 9: AT-Commands GPS

2.3 Αισθητήρας Θερμοκρασίας-Υγρασίας DHT-22

Ο αισθητήρας DHT22 πρόκειται για έναν σύνθετο αισθητήρα ο οποίος έχει τη δυνατότητα μέτρησης θερμοκρασίας και υγρασίας ταυτόχρονα, εξάγοντας ένα βαθμονομημένο ψηφιακό σήμα στην έξοδό του. Με τη χρήση της αποκλειστικής τεχνικής ψηφιακού σήματος και την τεχνολογία ανίχνευσης θερμοκρασίας και υγρασίας, εξασφαλίζει υψηλή αξιοπιστία και εξαιρετικά μακροχρόνια σταθερότητα. Αυτός ο αισθητήρας περιλαμβάνει ένα ωμικού τύπου στοιχείο μέτρησης υγρασίας και ένα NTC στοιχείο μέτρησης θερμοκρασίας τα οποία συνδέονται με ένα υψηλής απόδοσης 8 bit μικροελεγκτή, προσφέροντας άριστη ποιότητα, γρήγορη απόκριση,

ικανότητα στο να μένει ανεπηρέαστος στις παρεμβολές και τέλος καλή σχέση κόστους και αποτελεσματικότητας .

Κάθε στοιχείο DHT22 βαθμονομείται με εξαιρετική ακρίβεια όσον αφορά την υγρασία. Οι συντελεστές βαθμονόμησης αποθηκεύονται ως προγράμματα στη μνήμη του OTP τα οποία χρησιμοποιούνται από την εσωτερική διαδικασία ανίχνευσης του σήματος του αισθητήρα. Η σειριακή διεπαφή με τις ακίδες καθιστά την ολοκλήρωση του συστήματος εύκολη και γρήγορη. Διακρίνεται για το μικρό μέγεθός του, την χαμηλή κατανάλωση, την πάνω από 20 μέτρα μετάδοση του σήματος, με αποτέλεσμα να είναι η καλύτερη επιλογή όχι μόνο για απλές εφαρμογές αλλά και για πιο απαιτητικές. Το εξάρτημα είναι ένα πακέτο με τέσσερις μονές ακίδες στη σειρά. Είναι βολικό στη σύνδεση και έχει τη δυνατότητα παροχής ειδικών πακέτων σύμφωνα με το αίτημα των χρηστών. Οι παρακάτω πίνακες παρουσιάζουν τις τεχνικές προδιαγραφές του αισθητήρα DHT22. Στον πρώτο πίνακα υπάρχει μία επισκόπηση των τεχνικών προδιαγραφών, στον δεύτερο τα ηλεκτρικά χαρακτηριστικά του και στον τρίτο μία λεπτομερειακή ανάλυση.



Εικόνα 35: Αισθητήρας DHT-22

	Κατάσταση	Ελάχιστο	Τυπικό	Μέγιστο
Τροφοδοσία	DC	3.3V	5V	6V
Παροχή ρεύματος	-	1mA	-	1.5mA
	μέσος όρος	0.3mA	-	1mA
	αναμονή	40uA	-	50uA
Περίοδος δειγματοληψίας	second	1S	-	-

Πίνακας 10: Ηλεκτρικά Χαρακτηριστικά DHT-22

Παράμετροι	Καταστάσεις	Ελάχιστο	Τυπικό	Μέγιστο
Υγρασία				
Ανάλυση	-	1%RH	1%RH 8 bit	1%RH
Επαναληψιμότητα	-	-	±0.5%RH	-
Ακρίβεια	25°C	-	±2%RH	-
	0-50°C	-	-	±5%RH
Εναλλαξιμότητα	Πλήρως Εναλλάξιμα			
Εύρος Μέτρησης	-40°C	30%RH	-	90%RH
	25°C	20%RH	-	90%RH
	80°C	20%RH	-	80%RH
Χρόνος απόκρισης	1/e(63%)25°C , 1m/s Air	-	2 S	-
Υστέρηση	-	-	±0.3%RH	-
Μακροπρόθεσμη σταθερότητα	Τυπική	-	±0.5%RH/year	-
Θερμοκρασία				
Ακρίβεια	-	-	-	±0.5°C
Εύρος μέτρησης	-	-40°C	-	80°C
Χρόνος απόκρισης	1/e (63%)	1 S	-	-

Πίνακας 11: Λεπτομερής πίνακας τεχνικών χαρακτηριστικών του DHT22.

2.4 Κόστος Συστήματος

Το σύστημα που δημιουργήθηκε δεν βασίζεται τόσο στο υλικό κομμάτι όσο στο προγραμματιστικό για αυτό και δεν αγοράστηκαν πολλά εξαρτήματα για την υλοποίησή του. Όσα χρησιμοποιήθηκαν παρουσιάζονται παρακάτω όπως και η τιμή τους την στιγμή που αγοράστηκαν.

Κόστος εξαρτημάτων	
Εξάρτημα	Τιμή
Arduino Uno	22.5€
Sim808(Waveshare)	55€
Αισθητήρας Θερμοκρασίας-Υγρασίας	7€
Breadboard-Καλώδια	7€
Powerbank	30€
Συνολικό κόστος	121.5€

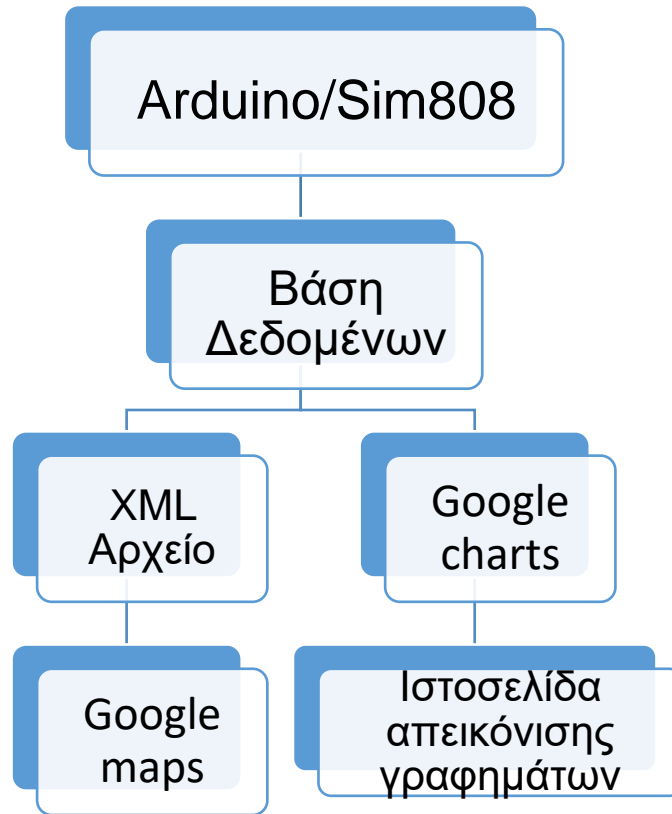
Πίνακας 12 : Κόστος συστήματος

Κεφάλαιο 3 : Λογισμικό μέρος συστήματος

Σε αυτό το κεφάλαιο θα αναλυθεί η λογική με την οποία υλοποιήθηκε και λειτουργεί το σύστημα υπηρεσιών απομακρυσμένης εποπτείας. Το σύνολο του λογισμικού μέρους μπορεί να διακριθεί σε επιμέρους τμήματα. Το πρώτο τμήμα αφορά το τεχνικό κομμάτι και ορίζεται ως «τμήμα μικροελεγκτή». Σε αυτό αναλύονται η αναγνώριση των συσκευών, η σύνδεση μεταξύ τους αλλά και η σύνδεση του κυκλώματος με τον μικροελεγκτή. Το δεύτερο τμήμα είναι υπεύθυνο για την επικοινωνία του μικροελεγκτή με τις επιμέρους υπηρεσίες. Τέλος το τρίτο τμήμα αφορά τις δυνατότητες διαχείρισης του συστήματος με το χρήστη. Ο συνδυασμός του δεύτερου και τρίτου τμήματος ορίζουν το «διαδικτυακό τμήμα».

Οι ενότητες του λογισμικού μέρους του συστήματος είναι οι εξής:

- Υπηρεσίες συστήματος
- Προγραμματισμός Arduino
- Βάση δεδομένων
- Δημιουργία XML αρχείου
- Προγραμματισμός Υπηρεσίας Google maps
- Λειτουργία ιστοχώρου



Εικόνα 36 : Διάγραμμα λογισμικού μέρους

3.1.1 Υπηρεσίες συστήματος

Το σύστημα που δημιουργήθηκε έχει την δυνατότητα να εφαρμοστεί σε ανθρώπους, κατοικίδια ακόμα και σε μηχανοκίνητα οχήματα ή ποδήλατα. Βέβαια οι υπηρεσίες που έχουν υλοποιηθεί θα εστιάσουν στον τομέα των ανθρώπων και των κατοικίδιων καθώς θα βοηθούν το χρήστη να παρακολουθεί την κατάστασή τους αλλά και τις συνθήκες του περιβάλλοντος όπου θα βρίσκονται. Σημαντική λειτουργία είναι αυτή της απομακρυσμένης εποπτείας της διαδρομής από όπου και αν βρίσκεται ο χρήστης. Τα βασικά χαρακτηριστικά του συστήματος είναι τα εξής:

- **Φιλικό απέναντι στο χρήστη:** Το σύστημα όπου εμφανίζεται η διαδρομή είναι αρκετά γνώριμο στο μέσο χρήστη καθώς αποτελεί μέρος της πλατφόρμας χαρτών της Google, η οποία είναι ευρέως διαδεδομένη σε φορητές συσκευές αλλά και ηλεκτρονικούς υπολογιστές.
- **Εύκολη Εξατομίκευση:** Οι υπηρεσίες που θα παρουσιαστούν ενεργοποιούνται μέσω κάποιων συνθηκών οι οποίες είναι πολύ εύκολο να παραμετροποιηθούν από τον εκάστοτε χρήστη για να εξυπηρετούν τις ανάγκες του.

Οι υπηρεσίες που έχουν υλοποιηθεί παρουσιάζονται παρακάτω

- **Απομακρυσμένος έλεγχος:** Ο χρήστης θα έχει την δυνατότητα να ενημερώνεται για πολλά συμβάντα μέσω ενός μηνύματος που θα δέχεται στο κινητό του τηλέφωνο με την ανάλογη ένδειξη.
- **Έλεγχος τοποθεσίας:** Η συγκεκριμένη υπηρεσία ελέγχει την τρέχουσα θέση του συστήματος και υπολογίζει την απόσταση του ηλικιωμένου ή του κατοικίδιου από μια συγκεκριμένη τοποθεσία όπως για παράδειγμα το σπίτι όπου διαμένει. Αν αυτή η απόσταση είναι μεγαλύτερη από ένα όριο που έχει θέσει ο χρήστης τότε το συμβάν ενεργοποιείται και ένα μήνυμα αποστέλλεται στον αριθμό κινητό που έχει ορίσει ο χρήστης.
- **Panic Button:** Η συγκεκριμένη υπηρεσία δίνει τη δυνατότητα σε έναν ηλικιωμένο που για κάποιον λόγο θέλει να ζητήσει βοήθεια, να ειδοποιήσει άμεσα το χρήστη με ένα γραπτό μήνυμα. Αυτό θα περιέχει το γεωγραφικό πλάτος και μήκος του ηλικιωμένου αλλά και ένα σύνδεσμο που ανακατευθύνει το χρήστη στην σελίδα τον χαρτών με το τρέχων στίγμα του ανθρώπου .
- **Καταγραφή βασικών περιβαλλοντικών συνθηκών:** Η τελευταία υπηρεσία θα παρακολουθεί τις συνθήκες της υγρασίας και της θερμοκρασίας. Αν ο συνδυασμός τους αποδεικνύει ότι οι συνθήκες είναι επιβλαβείς και καταπονούν τον άνθρωπο ή το κατοικίδιο θα αποστέλλεται ενημερωτικό μήνυμα στο χρήστη.

Στις επόμενες ενότητες γίνεται λεπτομερειακή παρουσίαση στον τρόπο δημιουργίας των παραπάνω υπηρεσιών αλλά και παραμετροποίησής τους. Στη συνέχεια ακολουθεί η παρουσίαση του προγραμματισμού του μικροελεγκτή Arduino.

3.2 Προγραμματισμός Arduino uno

Η παρούσα ενότητα αφορά την ανάλυση του κώδικα και της λειτουργίας της συσκευής. Συγκεκριμένα θα αναλυθεί ο τρόπος που η μονάδα Sim808 συγκεντρώνει τα δεδομένα του Gps αλλά και του δικτύου κινητής τηλεφωνίας τα οποία έπειτα επεξεργάζονται από το Arduino. Θα γίνει περιγραφή του τρόπου επικοινωνίας με τη βάση δεδομένων και τους χάρτες. Τέλος θα γίνει εμβάθυνση στον κώδικα με τον οποίο προγραμματίστηκε το Arduino, παρουσιάζοντας τις βασικότερες συναρτήσεις, τεχνικές και εργαλεία που χρησιμοποιήθηκαν, με σκοπό την καλύτερη κατανόηση του κώδικα. Ο κώδικας του Arduino υπάρχει ολόκληρος στο παράρτημα Α.

3.2.1 Βασικές βιβλιοθήκες

Η αρχιτεκτονική Arduino βασίζεται για τον προγραμματισμό της στη γλώσσα C. Στο φιλικό του περιβάλλον συμβάλουν οι βιβλιοθήκες και οι συναρτήσεις που βοηθούν στην υλοποίηση του κώδικα. Στη συνέχεια γίνεται ανάλυση της εκάστοτε βιβλιοθήκης και της λειτουργίας της.[5]

Βιβλιοθήκη Software Serial

Η συγκεκριμένη βιβλιοθήκη είναι προεγκατεστημένη στο λογισμικό του Arduino IDE και παρέχει σειριακή επικοινωνία με τις ακίδες 0 και 1 της πλακέτας με ταχύτητα μέχρι 115200 bps .

Αυτή επιτρέπει την εκτέλεση της επικοινωνίας παράλληλα με την εκτέλεση άλλων διεργασιών. Ακολουθεί μία λεπτομερειακή αναφορά στις εντολές και τις δομές, των οποίων έγινε χρήση:

- **SoftwareSerial mySerial(0,1):** Ονομάζει την σειριακή «mySerial» και την ξεκινά στις ψηφιακές ακίδες 0 και 1 όπου ορίζονται τα Rx και Tx αντίστοιχα.
- **mySerial.begin(9600):** Εκκινεί την σειριακή που δηλώσαμε παραπάνω με ταχύτητα 9600 bps.
- **mySerial.print ():** Τυπώνει τα δεδομένα στην ακίδα Tx.
- **mySerial.println ():** Τυπώνει τα δεδομένα στην ακίδα Tx σε μία νέα γραμμή.
- **mySerial.available():** Δέχεται τους χαρακτήρες που είναι διαθέσιμοι για ανάγνωση εκείνη τη στιγμή. Αυτοί οι χαρακτήρες θα έχουν ήδη φθάσει και αποθηκευτεί στην ενδιάμεση μνήμη (buffer).
- **MySerial.read():** Επιστρέφει τους χαρακτήρες που υπάρχουν στην ακίδα Tx. Αν η σειριακή δεν είναι διαθέσιμη απαντά με «-1».
- **MySerial.write():** Τυπώνει τα δεδομένα στην ακίδα Tx με την μορφή byte.
- **MySerial.flush():** Περιμένει να τελειώσει η μετάδοση σειριακών δεδομένων.

Βιβλιοθήκη TinyGps

Η βιβλιοθήκη που θα αναλύσουμε είναι σχεδιασμένη για να δέχεται δεδομένα της μορφής NMEA . Πρακτικά μετατρέπει τα δεδομένα που εισέρχονται από το Gps σε στοιχεία που μπορούμε να καταλάβουμε όπως: γεωγραφικό μήκος, ταχύτητα κ.α. . Ακολουθεί μία λεπτομερειακή αναφορά στις εντολές και τις δομές, των οποίων έγινε χρήση:

- **TinyGPS gps:** Δίνουμε ένα όνομα στην υπηρεσία, εδώ «gps».
- **Gps.encode():** Με αυτή την εντολή η βιβλιοθήκη καταλαβαίνει αν έχουν έρθει σωστού τύπου δεδομένα, δηλαδή αν έχει δεχτεί σωστό σήμα η κεραία του gps. Η συγκεκριμένη συνάρτηση θα δώσει αληθή απάντηση όταν έχει έγκυρα δεδομένα τα οποία στην συνέχεια θα επεξεργαστούν.
- **gps.f_get_position(&lat, &lon, &age):** Υπολογίζει την γεωγραφική θέση δηλαδή, γεωγραφικό πλάτος, μήκος και τον χρόνο τον οποίο χρειάστηκαν τα δεδομένα για να επεξεργαστούν (τάξης ms).
- **gps.f_speed_kmph():** Υπολογίζει την ταχύτητα .
- **gps.crack_datetime(&year, &month, &day, &hour, &minute, &second, &hundredths, &age):** Εξάγει την ημερομηνία και την ώρα που πήρε την τελευταία μέτρηση.
- **gps.satellites():** Υπολογίζει τους δορυφόρους με τους οποίους έχει επικοινωνία η κεραία.

- `gps.f_altitude()`: Υπολογίζει το υψόμετρο όπου βρίσκεται η κεραία.

Βιβλιοθήκη DHT

Η βιβλιοθήκη `dht` χρησιμοποιείται για την ανάγνωση των μετρήσεων μίας συγκεκριμένης σειράς αισθητήρων θερμοκρασίας και υγρασίας. Μεταφράζει το αναλογικό σήμα που δέχεται σε πραγματική τιμή θερμοκρασίας και υγρασίας. Ακολουθούν οι εντολές που χρησιμοποιήθηκαν:

- `dht.readHumidity()`: Υπολογίζει την υγρασία μέσω του αναλογικού σήματος που δέχεται.
- `dht.readTemperature()`: Υπολογίζει την θερμοκρασίας μέσω του αναλογικού σήματος που δέχεται.

3.2.2 Προγραμματισμός Αισθητήρα DHT22

Όπως αναφέραμε και παραπάνω με τη βοήθεια της βιβλιοθήκης `DHT` θα υπολογιστεί η θερμοκρασία και η υγρασία.

Αφού συμπεριληφθεί η βιβλιοθήκη με την εντολή.

```
#include "DHT.h"
```

Θα οριστεί η ακίδα της πλακέτας με την οποία έχει συνδεθεί ο αισθητήρας.

```
#define DHTPIN 9
```

Θα οριστεί ποιο μοντέλο αισθητήρα χρησιμοποιείται.

```
#define DHTTYPE DHT22
```

Επίσης με την παρακάτω εντολή εκκινεί ο αισθητήρας.

```
DHT dht (DHTPIN, DHTTYPE);
```

Ο υπολογισμός της υγρασίας και της θερμοκρασίας θα υπολογιστούν αντίστοιχα.

```
dht.readHumidity();  
dht.readTemperature();
```

3.2.3 Υπολογισμός δεδομένων GPS

Οι απαραίτητοι υπολογισμοί για τον καθορισμό της θέσης αλλά και διάφορων άλλων χρήσιμων τιμών θα γίνουν αποκλειστικά από την βιβλιοθήκη `tinypgps` σε συνδυασμό με την `Software Serial`.

Αρχικά θα συμπεριληφθεί η βιβλιοθήκη και θα οριστεί με το όνομα «`gps`».

```
#include <TinyGPS.h>  
TinyGPS gps;
```

Όπως και η βιβλιοθήκη για την σειριακή επικοινωνία αλλά και θα οριστούν οι ακίδες στις οποίες θα στέλνει και θα δέχεται δεδομένα.

```
#include <SoftwareSerial.h>
SoftwareSerial ss(2, 3);
```

Θα γίνει δήλωση των μεταβλητών που είναι απαραίτητες για τον υπολογισμό των δεδομένων που χρειάζονται(χρόνου απόκρισης, χρόνου, ημερομηνίας, ταχύτητας και πορείας).Επίσης δηλώνεται η συνάρτηση printFloat που θα αναλυθεί στη συνέχεια.

```
Unsigned long fix_age, time, date, speed, course;
Void printFloat(double f, int digits = 2);
```

Θα εκκινήσει η σειριακή με ταχύτητα 9600 bps.

```
ss.begin(9600);
```

Πλέον μέσα στην συνάρτηση της επανάληψης θα δημιουργηθεί συνθήκη που θα ελέγχει αν η πλακέτα δέχεται σωστού τύπου δεδομένα. Ξεκινά η επικοινωνία με την πλακέτα sim808 με τις AT-Commands που έχουν αναλυθεί σε προηγούμενη ενότητα όπως και η συνάρτηση millis .Τέλος όταν μεταδοθούν σωστά δεδομένα η μεταβλητή newData θα γίνει αληθής και θα αρχίσουν οι υπολογισμοί.

```
bool newData = false;
ss.println("AT");
ss.println("AT+CGNSPWR=1");
ss.println("AT+CGNSTST=1");

for (unsigned long start = millis(); millis() - start < 1000;)
{
  while (ss.available())
  {
    char c = ss.read();
    //Serial.write(c); //gps nmea string
    if (gps.encode(c))
      newData = true;
  }
}
```

Για να παρουσιάζονται όλες οι τιμές στην σειριακή ανεξαρτήτως αν είναι σωστές ή όχι θα χρειαστεί να αφαιρεθούν οι δύο «//» που ορίζουν την εντολή σαν σχόλιο.

Εφόσον η πλακέτα δεχθεί σωστά δεδομένα ορίζονται οι μεταβλητές στις οποίες θα αποθηκευτούν οι τιμές των υπολογισμών.

```
if (newData)
{
  float latitude, longitude;
  unsigned long age;
  float fkmph;
  int year;
  byte month, day, hour, minute, second, hundredths;
```

Στις επόμενες γραμμές υπολογίζονται οι τιμές που εισέρχονται από την κεραία του gps. Αυτές είναι το γεωγραφικό πλάτος και μήκος, η ταχύτητα, οι δορυφόροι, το υψόμετρο, η ώρα και η ημερομηνία. Όλες τυπώνονται στην σειριακή του ide για ευκολότερη απεικόνιση.

```
gps.f_speed_kmph();
gps.f_get_position(&latitude, &longitude, &age);
gps.crack_datetime(&year, &month, &day, &hour, &minute, &second, &hundredths, &age);

Serial.print("LAT=");
Serial.print(latitude == TinyGPS::GPS_INVALID_F_ANGLE ? 0.0 : latitude,
6);
Serial.print(" LON=");
Serial.print(longitude == TinyGPS::GPS_INVALID_F_ANGLE ? 0.0 :
longitude, 6);
Serial.print(" SAT=");
Serial.print(gps.satellites() == TinyGPS::GPS_INVALID_SATELLITES ? 0 :
gps.satellites());
Serial.print("Date: "); Serial.print(static_cast<int>(month));
Serial.print("/");
Serial.print(static_cast<int>(day)); Serial.print("/");
Serial.print(year);
Serial.print(" Time: "); Serial.print(static_cast<int>(hour+3));
Serial.print(":");
Serial.print(static_cast<int>(minute)); Serial.print(":");
Serial.print(static_cast<int>(second));
Serial.print("."); Serial.print(static_cast<int>(hundredths));
Serial.print("Alt: "); Serial.print(gps.f_altitude());
Serial.print(" (kmph): "); printFloat(gps.f_speed_kmph());
Serial.println();
```

Έπειτα θα μετατραπούν οι μεταβλητές του χρόνου και της ημερομηνίας σε τύπου string ώστε να σταλούν έπειτα με σωστό τρόπο στη βάση δεδομένων.

```
String mystring1=String(year);
String mystring2=String(month);
String mystring3=String(day);
String mystring4=String(hour+3);
String mystring5=String(minute);
String mystring6=String(second);
```

Επίσης θα συγχωνευθούν όλες οι μεταβλητές σε ένα string .

```
String datetime= mystring1 + "-" + mystring2 + "-" + mystring3 + "/" +
mystring4 + ":" + mystring5 + ":" + mystring6;
```

Τυπώνεται σαν μία μεταβλητή πλέον στην σειριακή

```
Serial.println(datetime);
```

Στην συνέχεια ορίζονται τρεις σταθερές σε μορφή string για την χρησιμοποίηση του στην υπηρεσία των χαρτών μέσω της βάσης δεδομένων.

```
String Text1 = "LAT=";
```



```
String Text2 = "LON=";  
String Text3 = "-";
```

Τέλος για την σωστή απεικόνιση των τιμών θα χρησιμοποιηθεί μία συνάρτηση που θα επεξεργάζεται τους αρνητικούς αριθμούς.

```
void printFloat(double number, int digits)  
{  
    if (number < 0.0) {  
        Serial.write('-');  
        number = -number;  
    }  
}
```

////////Διαχείριση αρνητικών αριθμών /////

```
double rounding = 0.5;  
for (uint8_t i=0; i<digits; ++i)  
    rounding /= 10.0;  
number += rounding;
```

////////Σωστή Αποικόνιση Στρογγυλοποίησης////////

```
unsigned long int_part = (unsigned long)number;  
double remainder = number - (double)int_part;  
Serial.print(int_part);
```

////////Τύπωσε το Ακέραιο Κομμάτι////////

```
if (digits > 0)  
    Serial.print(".");
```

////////Τύπωσε τα ψηφία ένα τη φορά////////

```
while (digits-- > 0) {  
    remainder *= 10.0;  
    int toPrint = int(remainder);  
    Serial.print(toPrint);  
    remainder -= toPrint;  
}  
}
```

3.3 Προγραμματισμός υπηρεσιών

Παρακάτω θα παρουσιαστεί ο κώδικας για την δημιουργία των υπηρεσιών που προαναφέρθηκαν.

3.3.1 Panic Button

Αρχικά θα προγραμματιστεί το κουμπί πανικού το οποίο όταν πατηθεί θα στείλει ένα μήνυμα sos στον χρήστη.

Θα οριστεί η ακίδα που θα συνδεθεί το κουμπί.

```
#define button 7
```

Θα οριστεί το κουμπί σαν τύπος `input_pullup` που έχει εξηγηθεί σε προηγούμενη ενότητα.

```
pinMode(button, INPUT_PULLUP);
```

Το κομμάτι της υπηρεσίας είναι το εξής:

Ορίζεται αρχικά το κουμπί με λογικό 0 (η λογική είναι ανεστραμμένη)

```
static uint8_t lastBtnState = HIGH;
```

Διαβάζεται η τωρινή τιμή.

```
uint8_t state = digitalRead(button);
```

Η συνθήκη βλέπει την λογική τιμή της μεταβλητής και αν αυτή είναι «1» θα εκτελεστεί.

```
if (state != lastBtnState) {  
    lastBtnState = state;  
    if (state == LOW) {
```

Αν πατηθεί το κουμπί το γεωγραφικό πλάτος και μήκος μετατρέπονται σε string για να μπορούν να σταλούν μέσω μηνύματος.

```
String lat=String(latitude == TinyGPS::GPS_INVALID_F_ANGLE ? 0.0 :  
latitude, 6);  
String lon=String(longtitude == TinyGPS::GPS_INVALID_F_ANGLE ? 0.0 :  
longtitude, 6);  
delay(100);
```

Στέλνονται οι AT-Commands για να εκτελεστεί η λειτουργία του μηνύματος και δηλώνεται το κινητό τηλέφωνο του χρήστη.

```
ss.println("AT+CMGF=1");  
delay(500);  
ss.println("AT+CMGS=\"+306949465421\"");  
delay(500);
```

Στέλνεται στην σειριακή το περιεχόμενο του μηνύματος.

```
ss.println("SOS");  
ss.println("Latitude= ");  
ss.print(lat);  
ss.println(" ");  
ss.println("Longitude= ");  
ss.print(lon);
```

Link που ανακατευθύνει στο google maps

```
ss.println("http://maps.google.com/maps?q=loc:");  
ss.print(lat);  
ss.print(",");  
ss.print(lon);
```

```
delay(200);
```

Αποστολή τερματικού κωδικού ώστε να καταλάβει η πλακέτα ότι τερματίστηκε το μήνυμα.

```
ss.write(26);  
Serial.println("Sms Sent !");  
}  
}
```

3.3.2 Παρακολούθηση απόστασης από σημείο

Η επόμενη υπηρεσία υπολογίζει την απόσταση του ανθρώπου ή του κατοικίδιου από μία τοποθεσία που έχει ορίσει ο χρήστης. Αν αυτή η απόσταση είναι μεγαλύτερη από αυτή που έχει θέσει ο χρήστης ενεργοποιείται το συμβάν.

Δηλώνουμε ξανά το γεωγραφικό πλάτος και μήκος.

```
latitude == TinyGPS::GPS_INVALID_F_ANGLE ? 0.0 : latitude, 6;  
longitude == TinyGPS::GPS_INVALID_F_ANGLE ? 0.0 : longitude, 6;
```

Δηλώνουμε τις απαραίτητες μεταβλητές και μέσω μαθηματικών τύπων υπολογίζεται η απόσταση από το σημείο που έχουμε θέσει με latitude=21.728059 και longitude=38.238707.

```
float X = ((21.728059 - longitude) * 24301 * cos(latitude)) / 360.0;  
float Y = ((38.238707 - latitude) * 24280) / 360.0;  
float X2 = X * X;  
float Y2 = Y * Y;  
float distance = sqrt(X2 + Y2);
```

Μετατρέπει την απόσταση από μίλια σε χιλιόμετρα.

```
float km = distance*1.609344;
```

Μετατρέπει τα χιλιόμετρα σε μέτρα και τα τυπώνει στη σειριακή.

```
float met = km*1000;  
Serial.print(met);
```

Η συνθήκη παρακολουθεί αν η απόσταση του κατοικίδιου γίνει μεγαλύτερη από τριάντα μέτρα και ενημερώνει το χρήστη με το αντίστοιχο μήνυμα. Θα μπορούσε πολύ εύκολα να αλλαχθεί το μήνυμα ώστε να ενημερώνει για την απόσταση ενός ανθρώπου, όπως και η τιμή των μέτρων που ελέγχονται.

```
if (met>30){  
String meters = String(met);  
delay(100);  
ss.println("AT+CMGF=1");  
delay(500);  
ss.println("AT+CMGS=\""+306949465421\"");  
delay(500);  
ss.println("Dog is away from home ");  
ss.print(meters);  
ss.println("meters");  
delay(200);  
ss.write(26);
```

```
}
```

Αν ο χρήστης ήθελε να ελέγχει αν ο άνθρωπος ή το κατοικίδιο απομακρυνθεί σε μεγαλύτερη απόσταση(π.χ βγει από ένα κτήμα) θα χρειαζόταν απλά να γίνει η παρακάτω τροποποίηση.

```
else if (met>100){
String meters = String(met);
delay(100);
ss.println("AT+CMGF=1");
    delay(500);
ss.println("AT+CMGS=\""+306949465421 "\"");
    delay(500);
ss.println("Dog is away from the premise ");
ss.print(meters);
ss.println("meters");
delay(200);
ss.write(26);
}
```

3.3.3 Συνθήκες Περιβάλλοντος

Η τελευταία υπηρεσία αφορά τον έλεγχο της υγρασίας και θερμοκρασίας του περιβάλλοντος. Ο σκοπός της είναι να ξέρει ο χρήστης ανά πάσα στιγμή τις συνθήκες στις οποίες βρίσκεται ο άνθρωπος ή το κατοικίδιο. Αν οι συνθήκες είναι ακατάλληλες θα ενημερωθεί με γραπτό μήνυμα με τις τρέχουσες τιμές θερμοκρασίας και υγρασίας.[10]

Η συνθήκη ενεργοποιείται εάν η υγρασία ξεπεράσει το 60% και η θερμοκρασία ταυτόχρονα ξεπεράσει τους 33 βαθμούς, κάτι που καθιστά τις συνθήκες του περιβάλλοντος άβολες.

```
if (hum>=60 & temp>=33)
{
```

Μετατρέπονται οι μεταβλητές σε string

```
String temps = String(temp);
String hums = String(hum);
    delay(100);
```

Περιεχόμενο μηνύματος.

```
ss.println("AT+CMGF=1");
    delay(500);
ss.println("AT+CMGS=\""+306949465421 "\"");
    delay(500);
ss.println("Weather Conditions: uncomfortable ");
ss.println("Temperature= ");
ss.println(temps);
ss.println("Humidity= ");
ss.print(hums);
delay(200);
ss.write(26);

}
```

3.3.4 Εποπτεία τοποθεσίας

Το σύστημα που έχει υλοποιηθεί βασίζεται στο δίκτυο κινητής τηλεφωνίας ώστε να μεταδώσει ασύρματα τα δεδομένα που έχει επεξεργαστεί ο μικροελεγκτής. Η σύνδεση της πλακέτας sim808 με την κάρτα sim πραγματοποιείται με διαδοχικές AT-Commands (έχουν αναφερθεί αναλυτικά σε προηγούμενη ενότητα) όπως και η επικοινωνία με τη βάση δεδομένων. Οι εντολές καθυστέρησης (delay) παίζουν σημαντικό ρόλο στην σωστή λειτουργία της επικοινωνίας και διαρκούν περίπου είκοσι δευτερόλεπτα, αυτός είναι ο χρόνος που θα ανανεώνεται η βάση δεδομένων.

Οι παρακάτω εντολές επανεκκινούν την σειριακή και την αδειάζουν από τα δεδομένα του gprs. Αυτό γίνεται για να εκτελεστούν σωστά οι εντολές και να μην υπάρχουν παρεμβολές από προηγούμενες.

```
ss.begin(9600);  
  Serial.begin(115200);  
  Serial.println("Sending Data...");  
ss.flush();  
  Serial.flush();
```

Διαδοχικές at-commands για την σύνδεση με το δίκτυο της κινητής τηλεφωνίας.

```
ss.println("AT+CGATT?");  
  delay(100);  
ss.println("AT+SAPBR=3,1,\"CTYPE\", \"GPRS\"");  
  delay(2000);
```

Δήλωση apn και κωδικού.

```
ss.println("AT+SAPBR=3,1,\"APN\", \"internet.vodafone.gr\"");  
  delay(2000);
```

Σύνδεση στο HTTP πρωτόκολλο.

```
ss.println("AT+SAPBR=1,1");  
  delay(2000);  
ss.println("AT+HTTTPINIT");  
  delay(2000);
```

Δήλωση της διεύθυνσης υπολογιστή στην οποία λειτουργεί η βάση δεδομένων και του αρχείου php που μεσολαβεί για να γίνει η σύνδεση.

```
ss.print("AT+HTTTPARA=\"URL\", \"http://195.251.14.234/arduino3/data.php?\"  
);
```

Διαδοχικές εκτυπώσεις στην σειριακή των μεταβλητών που στέλνονται στην βάση.

Χωρίζονται ενδιάμεσα με τον χαρακτήρα «&» ώστε το arduino να καταλαβαίνει ότι πρόκειται για διαφορετικές μεταβλητές.

```
ss.print("data1=");
```

```

ss.print(datetime);
ss.print("&data2=");
ss.print(latitude == TinyGPS::GPS_INVALID_F_ANGLE ? 0.0 : latitude, 6);
ss.print("&data3=");
ss.print(longtitude == TinyGPS::GPS_INVALID_F_ANGLE ? 0.0 : longtitude,
6);
ss.print("&data4=");
ss.print(gps.f_speed_kmph());
ss.print("&data5=");
ss.print(gps.satellites() == TinyGPS::GPS_INVALID_SATELLITES ? 0 :
gps.satellites());
ss.print("&data6=");
ss.print(gps.f_altitude());
ss.print("&data7=");
ss.print(Text1);
ss.print("&data8=");
ss.print(Text2);
ss.print("&data9=");
ss.print(Text3);
ss.print("&data10=");
ss.print(temp);
ss.print("&data11=");
ss.print(hum);
ss.print("");
ss.println("\\");
delay(2000);

```

Δήλωση στην at εντολή να κάνει Get τα δεδομένα.

```

ss.println("AT+HTTPACTION=0");
delay(5000);
ss.println("AT+HTTPREAD");
delay(1000);
ss.println("");

```

Τερματισμός σύνδεσης και ενημέρωση στη σειριακή ότι ολοκληρώθηκε η επικοινωνία.

```

ss.println("AT+HTTPTERM");
delay(300);
ss.println("");
Serial.print("Data Sent");
delay(1000);

}
}

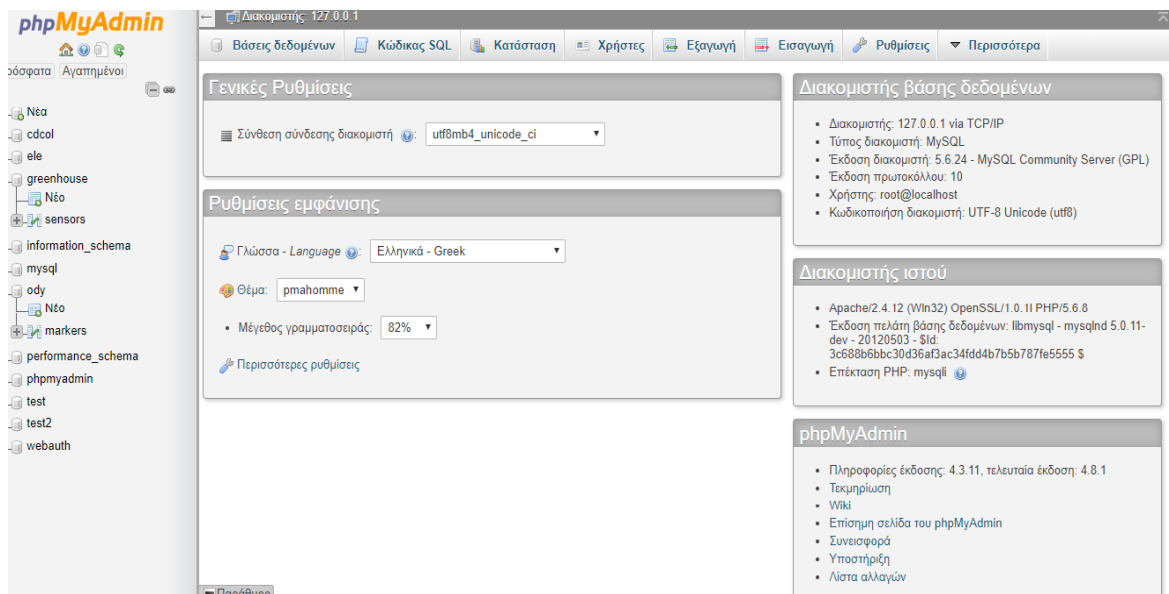
```

3.4 Βάση δεδομένων

Η βάση δεδομένων τροφοδοτεί τόσο τον ιστοχώρο που έχει υλοποιηθεί όσο και τους χάρτες όπου απεικονίζονται τα σημάδια(markers) της διαδρομής. Η βάση έχει ονομαστεί «ody» και αποτελείται από ένα πίνακα με όνομα «markers».Επίσης η δημιουργία βάσης είναι ελεύθερη και με την βοήθεια του Apache, Χαηpp και κώδικα rhr θα μεταφερθούν σε αυτή τα δεδομένα που έχουν παραχθεί. Από την πλευρά του Arduino χρειαζόμαστε προφανώς τις δύο πλακέτες Arduino uno, sim808. Θα χρησιμοποιηθούν οι at-commands της sim808 για την διαδικτυακή επικοινωνία

και την αποστολή των δεδομένων. Ο συγκεκριμένος πίνακας περιέχει δώδεκα(12) στήλες που χρησιμοποιούνται για την καταγραφή των μετρήσεων αλλά και για την παρουσίαση των δεδομένων στους χάρτες.

Για τη δημιουργία της βάσης κατευθυνόμαστε στην ιστοσελίδα phpmyadmin φτιάχνουμε μία νέα βάση με ένα δικό μας όνομα.



Εικόνα 37 : Αρχική σελίδα phpmyadmin

Έπειτα εκτελούμε τον παρακάτω sql κώδικα:

```
CREATE TABLE `markers` (
  `id` int(11) NOT NULL,
  `Datetime` datetime NOT NULL,
  `Lat` decimal(10,6) NOT NULL,
  `Lon` decimal(10,6) NOT NULL,
  `Speed` decimal(8,2) NOT NULL,
  `Satellites` int(10) NOT NULL,
  `Altitude` decimal(5,2) NOT NULL,
  `Text1` char(10) NOT NULL,
  `Text2` char(10) NOT NULL,
  `Text3` char(10) NOT NULL,
  `Temperature` decimal(8,2) NOT NULL,
  `Humidity` decimal(8,2) NOT NULL
)
ENGINE=MyISAM DEFAULT CHARSET=latin1;
```

Η περιγραφή των μεταβλητών και των τύπων του φαίνεται στον παρακάτω πίνακα.

Όνομα	Τύπος μεταβλητής	Σύνθεση	Κενές τιμές	Προεπιλογή
id	int(11)	Καμία	Όχι	Αυτόματη προσαύξηση κατά 1

Datetime	datetime	Καμία	Όχι	Καμία
Lat	decimal(10,6)	Καμία	Όχι	Καμία
Lon	decimal(10,6)	Καμία	Όχι	Καμία
Speed	decimal(8,2)	Καμία	Όχι	Καμία
Satellites	int(10)	Καμία	Όχι	Καμία
Altitude	decimal(5,2)	Καμία	Όχι	Καμία
Text1	char(10)	latin1_swedish_ci	Όχι	Καμία
Text2	char(10)	latin1_swedish_ci	Όχι	Καμία
Text3	char(10)	latin1_swedish_ci	Όχι	Καμία
Temperature	decimal(8,2)	Καμία	Όχι	Καμία
Humidity	decimal(8,2)	Καμία	Όχι	Καμία

Πίνακας 13: Στήλες βάσης δεδομένων

Αναλυτικότερα:

- **id:** Αποτελεί μία στήλη του πίνακα και η τιμή της μεταβλητής παράγεται αυτόματα από τη βάση κάθε φορά που εισέρχεται μία γραμμή με καινούργια δεδομένα. Σε αυτή τη περίπτωση έχουμε επιλέξει να αυξάνεται κατά 1 με κάθε νέα εγγραφή.
- **Datetime:** Καταγράφει τον ακριβή χρόνο που εισέρχεται η νέα γραμμή δεδομένων. Στην περίπτωση μας τροφοδοτείτε από τον μικροελεγκτή. Συνήθως επιλέγεται η αυτόματη εγγραφή τιμών από τη βάση.
- **Lat:** Αποθηκεύει την τιμή του γεωγραφικού πλάτους.
- **Lon:** Αποθηκεύει την τιμή του γεωγραφικού μήκους.
- **Speed:** Αποθηκεύει την τιμή της ταχύτητας.
- **Satellites:** Αποθηκεύει την τιμή των δορυφόρων.
- **Altitude:** Αποθηκεύει την τιμή του υψόμετρου.
- **Text1:** Αποθηκεύει τους χαρακτήρες «LAT=» που χρειάζονται για την απεικόνιση στους χάρτες
- **Text2:** Αποθηκεύει τους χαρακτήρες «LON=» που χρειάζονται για την απεικόνιση στους χάρτες
- **Text3:** Αποθηκεύει τον χαρακτήρα «-» που χρειάζεται για την απεικόνιση στους χάρτες
- **Temperature:** Αποθηκεύει την τιμή της θερμοκρασίας.
- **Humidity:** Αποθηκεύει την τιμή της υγρασίας.

			id	Datetime	Lat	Lon	Speed	Satellites	Altitude	Text1	Text2	Text3	Temperature	Humidity
Επεξεργασία	Αντιγραφή	Διαγραφή	135	2018-05-02 17:26:32	38.222061	21.750104	47.60		7 21.20	LAT=	LON=	-	0.00	0.00
Επεξεργασία	Αντιγραφή	Διαγραφή	136	2018-05-02 17:26:52	38.224288	21.749303	35.17		7 15.40	LAT=	LON=	-	0.00	0.00
Επεξεργασία	Αντιγραφή	Διαγραφή	137	2018-05-02 17:27:12	38.225036	21.747968	35.28		7 4.20	LAT=	LON=	-	0.00	0.00
Επεξεργασία	Αντιγραφή	Διαγραφή	138	2018-05-02 17:27:32	38.225761	21.747692	0.13		7 -2.10	LAT=	LON=	-	0.00	0.00
Επεξεργασία	Αντιγραφή	Διαγραφή	139	2018-05-02 17:27:53	38.226062	21.747627	27.35		7 -6.50	LAT=	LON=	-	0.00	0.00
Επεξεργασία	Αντιγραφή	Διαγραφή	140	2018-05-02 17:28:13	38.227714	21.744068	27.35		7 -10.50	LAT=	LON=	-	0.00	0.00
Επεξεργασία	Αντιγραφή	Διαγραφή	141	2018-05-02 17:28:33	38.229103	21.741477	16.65		8 -18.30	LAT=	LON=	-	0.00	0.00
Επεξεργασία	Αντιγραφή	Διαγραφή	142	2018-05-02 17:28:53	38.229209	21.741458	0.00		8 -22.60	LAT=	LON=	-	0.00	0.00
Επεξεργασία	Αντιγραφή	Διαγραφή	143	2018-05-02 17:29:14	38.230411	21.739931	36.26		8 -20.30	LAT=	LON=	-	0.00	0.00
Επεξεργασία	Αντιγραφή	Διαγραφή	144	2018-05-02 17:29:34	38.231529	21.740688	17.65		8 -15.60	LAT=	LON=	-	0.00	0.00
Επεξεργασία	Αντιγραφή	Διαγραφή	145	2018-05-02 17:29:54	38.233642	21.740348	43.84		8 -2.90	LAT=	LON=	-	0.00	0.00
Επεξεργασία	Αντιγραφή	Διαγραφή	146	2018-05-02 17:30:14	38.235832	21.739967	43.15		8 11.60	LAT=	LON=	-	0.00	0.00
Επεξεργασία	Αντιγραφή	Διαγραφή	147	2018-05-02 17:30:34	38.237792	21.739732	43.15		8 28.30	LAT=	LON=	-	0.00	0.00
Επεξεργασία	Αντιγραφή	Διαγραφή	148	2018-05-02 17:30:52	38.238231	21.739019	11.65		8 32.40	LAT=	LON=	-	0.00	0.00
Επεξεργασία	Αντιγραφή	Διαγραφή	560	2018-06-16 16:14:16	38.238544	21.727943	0.26		7 18.10	LAT=	LON=	-	26.90	50.10
Επεξεργασία	Αντιγραφή	Διαγραφή	561	2018-06-16 16:14:36	38.238494	21.727947	0.31		7 16.30	LAT=	LON=	-	26.90	50.90

Εικόνα 38 : Πίνακας markers βάσης δεδομένων

3.4.1 Αρχείο Php για σύνδεση με τη βάση δεδομένων

Τα αρχεία php που θα παρουσιαστούν παρακάτω λειτουργούν ως μεσολαβητές για την επικοινωνία μεταξύ Arduino και sql βάσης. Τα συγκεκριμένα αρχεία έχουν αποθηκευτεί στον φάκελο του τοπικού server όπου λειτουργεί τοxampp.Ολόκληροι οι κώδικες υπάρχουν στο παράρτημα Β.

Αρχείο data.php

Περιέχει τα απαραίτητα στοιχεία σύνδεσης με τη βάση όπως όνομα, κωδικό και τοπικό server. Έπειτα κάνει «insert» τις μεταβλητές και σταθερές που προέρχονται από το arduino στον πίνακα της βάσης που έχει δηλωθεί.

Δήλωση στοιχείων της βάση.

```
<?php
$dbusername = "root";
$dbpassword = "goopy3121!!";
$server = "localhost";
$My_db = "ody";

// Connect to your database
$dbconnect = mysql_pconnect($server, $dbusername, $dbpassword);
$dbselect = mysql_select_db("ody", $dbconnect);
```

Εισαγωγή μεταβλητών, σταθερών και τερματισμός του «query» .

```
$sql = "INSERT INTO markers (Datetime, Lat, Lon, Speed, Satellites, Altitude,
Text1, Text2, Text3, Temperature, Humidity) VALUES
```

```
('"$_GET["data1"] "','"$_GET["data2"] "','"$_GET["data3"] "','"$_GET["data4"] "','"$_GET["data5"] "','"$_GET["data6"] "','"$_GET["data7"] "','"$_GET["data8"] "','"$_GET["data9"] "','"$_GET["data10"] "','"$_GET["data11"] "','"');
// Execute SQL statement
mysql_query($sql);
?>
```

3.4.2 Αρχεία Php για την γραφική απεικόνιση των μεταβλητών

Για την άμεση και πιο ευανάγνωστη μορφή των μεταβλητών θα δημιουργηθούν γραφήματα και γραφικά στοιχεία με την βοήθεια των Google Charts, που έχουν αναλυθεί σε προηγούμενη ενότητα. Η google παρέχει δωρεάν κώδικα για μια μεγάλη ποικιλία διαφορετικών τύπων διαγραμμάτων. Θα παρουσιαστεί ο κώδικας με τον οποίο δημιουργήθηκαν και θα δοθούν απλές οδηγίες για την τροποποίηση τους με βάση τις ανάγκες του χρήστη.[17]



Εικόνα 39 : Λογότυπο google charts

3.4.2.1 Google Gauge

Google Gauge

Αρχικά θα σχεδιαστεί το γραφικό μορφής μετρητή(gauge).Ο κώδικας που θα παρουσιαστεί αφορά ένα μετρητή, ο κώδικας για όλους τους μετρητές εμφανίζεται στο παράρτημα Β.

«gauge.php»

Δηλώνονται τα στοιχεία της βάσης

```
<?php
// Prepare variables for database connection

$dbusername = "root";
$dbpassword = "goopy3121!!";
```

```
$server = "localhost";
```

Γίνεται σύνδεση στη βάση.

```
// Connect to your database
$mysqli = new mysqli($server, $dbusername, $dbpassword, "ody");
if($mysqli->connect_error)
    die('Connect Error (' . mysqli_connect_errno() . ') ' .
mysqli_connect_error());
```

Δημιουργούμε το query στη βάση όπου ζητείται η τιμή του γεωγραφικού πλάτους.

```
// Execute SQL statement

$result = $mysqli->query("SELECT Lat FROM ody.markers ORDER BY id DESC
LIMIT 1");

if ($result->num_rows > 0) {

    // output data of each row
    while($row = $result->fetch_assoc()) {
        $LValue = $row["Lat"];

    }
}
else {
    $LValue = 0;
}
?>
```

Παρακάτω υπάρχει το κομμάτι html κώδικα που είναι απαραίτητο για την παρουσίαση του γραφικού στην ιστοσελίδα που έχει δημιουργηθεί και θα εξηγηθεί σε επόμενη ενότητα.

```
<!doctype html>
<html>
<head><meta http-equiv="Content-Type" content="text/html; charset=iso8859-7">

<title>Gauge </title>
<link rel="stylesheet" href="../fonts/fonts.css">
<link rel="stylesheet" href="buttons.css">
<script src="gauge.js"></script>
```

Δήλωση χρόνου ανά τον οποίο ανανεώνεται το γραφικό.

```
<meta http-equiv="refresh" content="30" />
```

Κύριο κομμάτι ιστοσελίδας.

```
</head>
<body style="background: # 001433">
<div style="width:70%;font-size:28pt;text-align:center;color:#ffffff;text-
align:left;">Gps Real-Time Tracking </div>
```

```



<div style="width:100%;font-size:19pt;color:#cca300;text-
align:left;"><strong>Real-Time Gauge
</div>
</div>
</div>

```

Παρακάτω παρουσιάζεται η σχεδίαση του μετρητή όπου δηλώνεται το πλάτος το ύψος του περιγράμματος, ο τίτλος, το χρώμα αλλά και τις κλίμακες της βελόνας στο εσωτερικό του.

```

<canvas data-type="radial-gauge"
data-width="222"
data-height="222"
data-units=""
data-title="Latitude"
data-value=<?php echo "\"\". $LValue. "\"\n"?>
data-animate-on-init="false"
data-animated-value="false"
data-min-value="0"
data-max-value="45"
data-major-ticks="0,5,10,15,20,25,30,35,45"
data-minor-ticks="1"
data-stroke-ticks="false"
data-highlights=' [
    { "from": 0, "to": 22.5, "color": "rgba(0, 153, 51,1)" },
    { "from": 22.5, "to":45, "color": "rgba(204, 153, 0,1)" }
]'
data-color-plate="transparent"
data-color-major-ticks="#f5f5f5"
data-color-minor-ticks="#ddd"
data-color-title="#fff"
data-color-units="#ccc"
data-color-numbers="#eee"
data-color-needle-start="rgba(240, 128, 128, 1)"
data-color-needle-end="rgba(255, 160, 122, .9)"
data-value-box="true"
data-animation-rule="bounce"
data-animation-duration="500"
data-border-outer-width="3"
data-border-middle-width="3"
data-border-inner-width="3"
>>/canvas>

```

Το απαραίτητο script για την δημιουργία του γραφικού είναι το εξής:

```

<script>
if (!Array.prototype.forEach) {
    Array.prototype.forEach = function(cb) {
        var i = 0, s = this.length;
        for (; i < s; i++) {
            cb && cb(this[i], i, this);
        }
    }
}

```

```

    }
}

document.fonts && document.fonts.forEach(function(font) {
    font.loaded.then(function() {
        if (font.family.match(/Led/)) {
            document.gauges.forEach(function(gauge) {
                gauge.update();
                gauge.options.renderTo.style.visibility = 'visible';
            });
        }
    });
});

var timers = [];

function animateGauges() {
    document.gauges.forEach(function(gauge) {
        timers.push(setInterval(function() {
            gauge.value = Math.random() *
                (gauge.options.maxValue - gauge.options.minValue) +
                gauge.options.minValue;
        }, gauge.animation.duration + 50));
    });
}

function stopGaugesAnimation() {
    timers.forEach(function(timer) {
        clearInterval(timer);
    });
}

function resize() {
    var size = parseFloat(document.getElementById('gauge-size').value) ||
400;

    document.gauges.forEach(function(gauge) {
        gauge.update({ width: size, height: size });
    });
}

function setText() {
    var text = document.getElementById('gauge-text').value;

    document.gauges.forEach(function(gauge) {
        gauge.update({ valueText: text });
    });
}
</script>
</body>
</html>

```



Εικόνα 40 : Ο μετρητής με την τελευταία μέτρηση του γεωγραφικού πλάτους

3.4.2.2 Statistics

«statistics.php»

Στο επόμενο γράφημα που θα δημιουργηθεί παρουσιάζονται οι τελευταίες πέντε τιμές της μεταβλητής που θα επιλέξει ο χρήστης, συναρτήσει του χρόνου. Θα παρουσιαστούν ενδεικτικά δύο γραφήματα αυτό της ταχύτητας και του υψομέτρου . Ο κώδικας για τις υπόλοιπες μεταβλητές της βάσης παρουσιάζεται στο παράρτημα Β.

Το κομμάτι του html κώδικα που υποδεικνύει τα τριάντα δευτερόλεπτα που ανανεώνεται το γράφημα αλλά και το απαραίτητο script που καλείται από το server της Google.

```
<html>
<head>
  <meta http-equiv="refresh" content="30" /> <!--refresh selidas-->
  <script type="text/javascript"
src="https://www.gstatic.com/charts/loader.js"></script>
  <script type="text/javascript">
    google.charts.load('current', {'packages':['line']});
    google.charts.setOnLoadCallback(drawChart);
```

Η συνάρτηση `drawChart()` δημιουργεί το γράφημα και δηλώνει ποιες στήλες καλεί από τη βάση.

```
function drawChart() {
  var data = new google.visualization.DataTable();
  data.addColumn('datetime', 'Datetime');
  data.addColumn('number', 'Speed');
  data.addColumn('number', 'Altitude');
```

Το κομμάτι του php κώδικα ορίζει τα στοιχεία της βάσης και πραγματοποιεί τη σύνδεση.

```

<?php
// Prepare variables for database connection

$dbusername = "root";
$dbpassword = "goopy3121!!";    $server = "localhost";

// Connect to your database
$mysqli = new mysqli($server, $dbusername, $dbpassword, "ody");
if($mysqli->connect_error)
    die('Connect Error (' . mysqli_connect_errno() . ') ' .
mysqli_connect_error());

```

Αφού γίνει το query στη βάση «σπάει» την μεταβλητή datetime σε έτος, μήνα, μέρα, ώρα και δευτερόλεπτα για να δηλωθεί κατάλληλα στο script παρακάτω. Με την εντολή DESC LIMIT 5 ορίζεται πόσες τιμές θα υπάρχουν στο διάγραμμα .

```

$result = $mysqli->query("SELECT YEAR(Datetime) AS year,
MONTH(Datetime) AS month, DAY(Datetime) AS day, HOUR(Datetime) AS hour,
MINUTE(Datetime) AS minute, SECOND(Datetime) AS sec, Speed, Altitude FROM
ody.markers ORDER BY Datetime DESC LIMIT 5");

if ($result->num_rows > 0) {

// output data of each row
while($row = $result->fetch_assoc()) {
    echo "data.addRow([new Date(".$row["year"].",".$row["month"]."-
1,$row["day"].",".$row["hour"].",".$row["minute"].",".$row["sec"]."),
".$row["Speed"].",".$row["Altitude"]);
}
}
else {
}
}
?>

```

Η κάθε μεταβλητή περιέχει τις επιλογές απεικόνισής της .Όπως ο τίτλος, η μονάδα μέτρησης και ο τρόπος που θα παρουσιάζεται.

```

var Speed_options = {
    chart: {

        title: 'Speed vs Time',
        subtitle: 'measured in km/h'
    },
    hAxis: { format: 'HH:mm:ss a' },
    width: 600,
    height: 300
};

```

```

var Altitude_options = {
    chart: {
        title: 'Altitude vs Time',
        subtitle: 'measured in meters'
    },
    hAxis: { format: 'HH:mm:ss a' },
};

```

```

        width: 600,
        height: 300
    };

```

Δηλώνεται σε ποιες στήλες θα αντιστοιχεί η κάθε μεταβλητή.

```

    var Speed_chart = new
google.charts.Line(document.getElementById('Speed_div'));

    var Speed_view = new google.visualization.DataView(data);
    Speed_view.setColumns([0,1]);
    Speed_chart.draw(Speed_view,
google.charts.Line.convertOptions(Speed_options));

//-----
    var Altitude_chart = new
google.charts.Line(document.getElementById('Altitude_div'));

    var Altitude_view = new google.visualization.DataView(data);
    Altitude_view.setColumns([0,2]);
    Altitude_chart.draw(Altitude_view,
google.charts.Line.convertOptions(Altitude_options));
//-----

}
</script>

```

Μετά το τέλος του javascript υπάρχει το υπόλοιπο κομμάτι κώδικα html που εκτός από το να δίνει ένα τίτλο στη σελίδα παρουσιάζει και τα γραφικά μέσα σε έναν πίνακα για καλύτερη στοίχιση.

```

</head>
<title>Statistics </title>
<body style="background: # 001433">
    <div style="width:100%">
        <div style="width:90%;font-size:30pt;text-
align:center;color:#ffffff;text-align:left;">Gps real-time tracking</div>
    </div>


    <div style="width:100%;font-size:19pt;color:#cca300;text-
align:left;"><strong>Real-Time Διαγράμματα
    </div>
    <br>
    <div align="right">

</div>

<table style="border-spacing: 15px">
<tr>
    <td style="border: 1px solid white">
        <div id="Speed_div" style="width: 600px; height:
300px"></div>

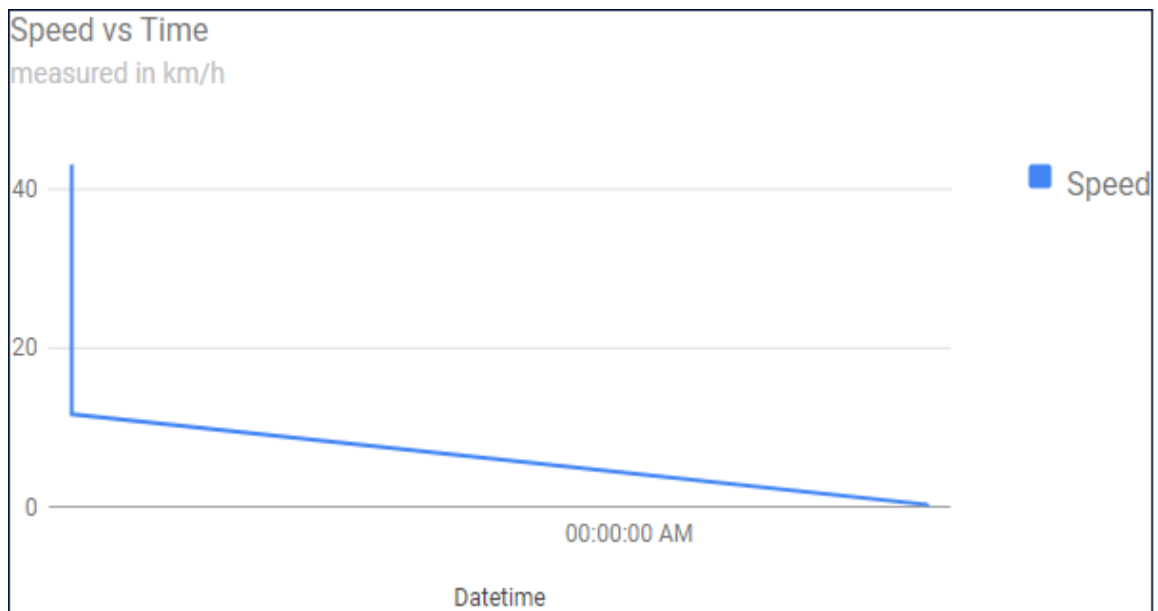
```



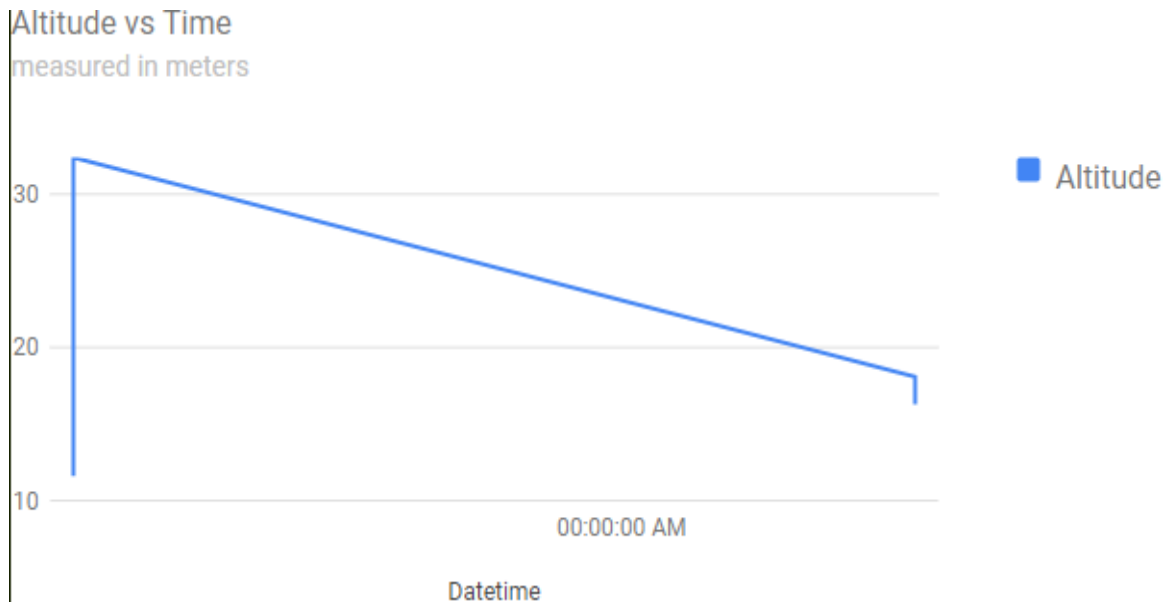
```

        </td>
        <td style="border: 1px solid white">
            <div id="Altitude_div" style="width: 600px; height:
300px"></div>
        </td>
    </tr>
</table>
</body>
</html>

```



Εικόνα 41 : Η γραφική της ταχύτητας συναρτήσεϊ του χρόνου



Εικόνα 42 : Η γραφική του υψομέτρου συναρτήσει του χρόνου

3.4.2.3 Weekly Statistics

«weekly_statistics.php»

Στα επόμενα γραφήματα που θα δημιουργηθούν παρουσιάζονται η μέση, ελάχιστη και μέγιστη τιμή της μεταβλητής για κάθε ημέρα της τελευταίας εβδομάδας. Θα παρουσιαστούν ενδεικτικά δύο γραφήματα αυτό της ταχύτητας και του υψομέτρου . Αυτές οι γραφικές αναπαραστάσεις μας δίνουν ένα άμεσο τρόπο παρακολούθησης των μεταβλητών που περιέχει η βάση δεδομένων. Ο κώδικας για τα υπόλοιπα γραφήματα παρουσιάζεται στο παράρτημα Β.Η διαδικασία που θα ακολουθηθεί θα είναι πανομοιότυπη με την προηγούμενη.

Το κομμάτι του html κώδικα που υποδεικνύει τα τριάντα δευτερόλεπτα που ανανεώνεται το γράφημα αλλά και το απαραίτητο script που καλείται από το server της Google.

```
<html>
  <head>
    <meta http-equiv="refresh" content="30" />
    <script type="text/javascript"
src="https://www.gstatic.com/charts/loader.js"></script>
    <script type="text/javascript">
      google.charts.load('current', {'packages':['corechart']});
      google.charts.setOnLoadCallback(drawChart);
```

Η συνάρτηση `drawChart()` δημιουργεί το γράφημα και δηλώνει ποιες στήλες καλεί από τη βάση. Σε αυτή τη περίπτωση δηλώνονται η ελάχιστη, μέγιστη και μέση τιμή της μεταβλητής ξεχωριστά ώστε να υπολογιστεί με sql κώδικα .

```
function drawChart() {
    var data = new google.visualization.DataTable();
    data.addColumn('datetime', 'Datetime');
    data.addColumn('number', 'Average Speed');
    data.addColumn('number', 'Max Speed');
    data.addColumn('number', 'Min Speed');
    data.addColumn('number', 'Average Altitude');
    data.addColumn('number', 'Max Altitude');
    data.addColumn('number', 'Min Altitude');
}
```

Έπεται το κομμάτι του κώδικα στο οποίο ορίζονται τα στοιχεία της βάσης και γίνεται η σύνδεση.

```
<?php
// Prepare variables for database connection

$dbusername = "root";
$dbpassword = "goopy3121";
$server = "localhost";

// Connect to your database
$mysqli = new mysqli($server, $dbusername, $dbpassword, "ody");
if($mysqli->connect_error)
    die('Connect Error (' . mysqli_connect_errno() . ') ' .
mysqli_connect_error());
```

Αφού γίνει το query στη βάση «σπάει» την μεταβλητή datetime σε έτος, μήνα, μέρα, ώρα και δευτερόλεπτα για να δηλωθεί κατάλληλα στο script παρακάτω. Με την εντολή `DESC LIMIT 7` ορίζεται πόσες τιμές θα υπάρχουν στο διάγραμμα ,στην συγκεκριμένη περίπτωση 7.

```
$result = $mysqli->query("SELECT YEAR(Datetime) AS year, MONTH(Datetime)
AS month, DAY(Datetime) AS day, AVG(Speed),MAX(Speed),MIN(Speed),
AVG(Altitude),MAX(Altitude),MIN(Altitude) from markers group by DAY(Datetime)
ORDER BY Datetime DESC limit 7");

if ($result->num_rows > 0) {

// output data of each row
while($row = $result->fetch_assoc()) {
    echo "data.addRow([new Date(".$row["year"].".".$row["month"]."-
1, ".$row["day"]."),
    ".$row["AVG(Speed)"].".".$row["MAX(Speed)"].".".$row["MIN(Speed)"].".".$row["
AVG(Altitude)"].".".$row["MAX(Altitude)"].".".$row["MIN(Altitude)"]."]); \n";
}
}
else {
}
?>
```

Η κάθε μεταβλητή περιέχει τις επιλογές απεικόνισής της .Όπως ο τίτλος, η μονάδα μέτρησης και ο τρόπος που θα παρουσιάζεται.

```
var Speed_options = {
    title: 'Speed Last Week',
```

```

        subtitle: 'measured in km/h',
        hAxis: { format: 'MMM dd, yyyy' },
        seriesType: 'bars',
        width: 600,
        height: 300
    };

```

```

    var Altitude_options = {
        title: 'Altitude Last Week',
        subtitle: 'measured in meters',
        hAxis: { format: 'MMM dd, yyyy' },
        seriesType: 'bars',
        width: 600,
        height: 300
    };

```

Δηλώνεται σε ποιες στήλες θα αντιστοιχεί η κάθε μεταβλητή. Σε αυτή την περίπτωση δηλώνονται σε κάθε μεταβλητή τέσσερις στήλες που περιέχουν την ημερομηνία, την ελάχιστη, μέγιστη και μέση τιμή.

```

    var Speed_chart = new
    google.visualization.ComboChart(document.getElementById('Speed_div'));

```

```

    var Speed_view = new google.visualization.DataView(data);
    Speed_view.setColumns([0,1,2,3]);
    Speed_chart.draw(Speed_view, Speed_options);

```

```

//-----
    var Altitude_chart = new
    google.visualization.ComboChart(document.getElementById('Altitude_div'));

```

```

    var Altitude_view = new google.visualization.DataView(data);
    Altitude_view.setColumns([0,4,5,6]);
    Altitude_chart.draw(Altitude_view, Altitude_options);

```

```

//-----

```

```

}

```

Μετά το τέλος του javascript υπάρχει το υπόλοιπο κομμάτι κώδικα html που εκτός από το να δίνει ένα τίτλο στη σελίδα παρουσιάζει και τα γραφικά μέσα σε έναν πίνακα για καλύτερη στοίχιση.

```

</script>
</head>
<title>Weekly </title>
<body style="background: # 001433">
    <div style="width:100%">

        <div style="width:90%;font-size:30pt;text-align:center;color:#
cca300;text-align:left;"><strong>Gps Real-Time Tracking</div>

```

```

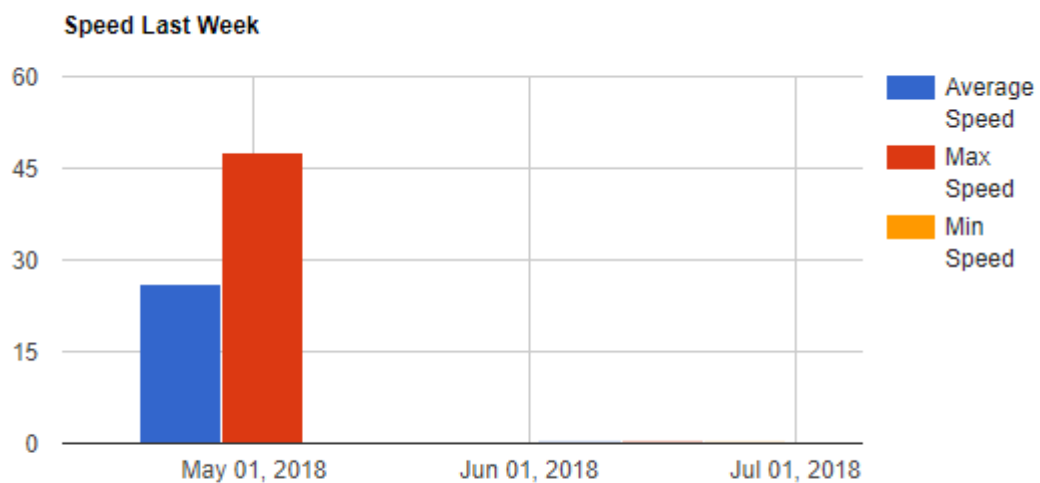
    </div>


    <div style="width:100%;font-size:19pt;color:#ffffff;text-
align:left;">Εβδομαδιαίες Μετρήσεις
    </div>
    <br>
    <div align="right">

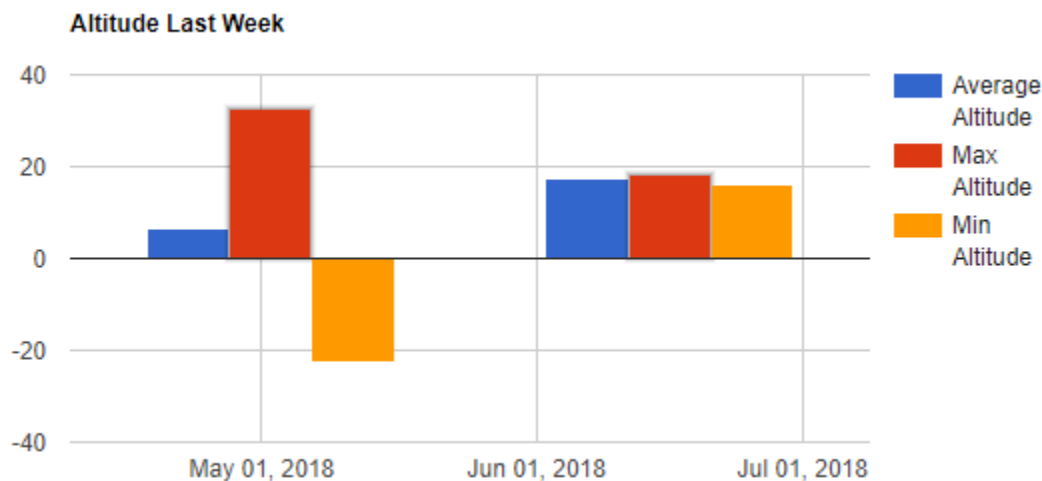
</div>

<table style="border-spacing: 15px">
    <tr>
        <td style="border: 1px solid white">
            <div id="Speed_div" style="width: 600px; height:
300px"></div>
        </td>
        <td style="border: 1px solid white">
            <div id="Altitude_div" style="width: 600px; height:
300px"></div>
        </td>
    </tr>
</table>
</body>
</html>

```



Εικόνα 43: Απεικόνιση της ελάχιστης, μέγιστης και μέσης τιμής της ταχύτητας.



Εικόνα 44 : Απεικόνιση της ελάχιστης, μέγιστης και μέσης τιμής του υψομέτρου.

Παρατηρείτε ότι η ελάχιστη τιμή της ταχύτητας είναι μηδενική για αυτό και δεν απεικονίζεται με την κίτρινη μπάρα στο γράφημα. Επίσης στην απεικόνιση του υψόμετρου παρατηρούνται αρνητικές τιμές, ο λόγος είναι ότι ο άνθρωπος ή το κατοικίδιο βρίσκονται σε σημείο που το υψόμετρό του είναι κάτω από το επίπεδο της θάλασσας.

3.4.2.4 Monthly Statistics

«monthly_statistics.php»

Στα επόμενα γραφήματα που θα δημιουργηθούν παρουσιάζονται η μέση, ελάχιστη και μέγιστη τιμή της μεταβλητής για κάθε μήνα του τελευταίου χρόνου. Θα παρουσιαστούν ενδεικτικά δύο γραφήματα αυτό της ταχύτητας και του υψόμετρου . Αυτές οι γραφικές αναπαραστάσεις μας δίνουν ένα άμεσο τρόπο παρακολούθησης των μεταβλητών που περιέχει η βάση δεδομένων. Ο κώδικας για τα υπόλοιπα γραφήματα παρουσιάζεται στο παράρτημα Β.Η διαδικασία που θα ακολουθηθεί θα είναι πανομοιότυπη με την προηγούμενη.

Το κομμάτι του html κώδικα που υποδεικνύει τα τριάντα δευτερόλεπτα που ανανεώνεται το γράφημα αλλά και το απαραίτητο script που καλείται από το server της Google.

```
<html>
  <head>
    <meta http-equiv="refresh" content="30" />
    <script type="text/javascript"
src="https://www.gstatic.com/charts/loader.js"></script>
    <script type="text/javascript">
      google.charts.load('current', {'packages':['corechart']});
      google.charts.setOnLoadCallback(drawChart);
```

Η συνάρτηση `drawChart()` δημιουργεί το γράφημα και δηλώνει ποιες στήλες καλούνται από τη βάση. Σε αυτή τη περίπτωση δηλώνονται η ελάχιστη, μέγιστη και μέση τιμή της μεταβλητής ξεχωριστά ώστε να υπολογιστεί με sql κώδικα .

```
function drawChart() {  
  
    var data = new google.visualization.DataTable();  
    data.addColumn('datetime', 'Datetime');  
    data.addColumn('number', 'Average Speed');  
    data.addColumn('number', 'Max Speed');  
    data.addColumn('number', 'Min Speed');  
    data.addColumn('number', 'Average Altitude');  
    data.addColumn('number', 'Max Altitude');  
    data.addColumn('number', 'Min Altitude');
```

Έπεται το κομμάτι του κώδικα στο οποίο ορίζονται τα στοιχεία της βάσης και γίνεται η σύνδεση.

```
<?php  
    // Prepare variables for database connection  
  
    $dbusername = "root";  
    $dbpassword = "goopy3121";  
    $server = "localhost";  
  
    // Connect to your database  
    $mysqli = new mysqli($server, $dbusername, $dbpassword, "ody");  
    if($mysqli->connect_error)  
        die('Connect Error (' . mysqli_connect_errno() . ') ' .  
mysqli_connect_error());
```

Αφού γίνει το query στη βάση «σπάει» την μεταβλητή `datetime` σε έτος, μήνα, μέρα, ώρα και δευτερόλεπτα για να δηλωθεί κατάλληλα στο script παρακάτω. Με την εντολή `DESC LIMIT 12` ορίζεται πόσες τιμές θα υπάρχουν στο διάγραμμα ,στην συγκεκριμένη περίπτωση 12 δηλαδή των 12 τελευταίων μηνών.

```
    $result = $mysqli->query("SELECT YEAR(Datetime) AS year, MONTH(Datetime)  
AS month, DAY(Datetime) AS day, AVG(Speed),MAX(Speed),MIN(Speed),  
AVG(Altitude),MAX(Altitude),MIN(Altitude) from markers group by  
MONTH(Datetime) ORDER BY Datetime DESC limit 12");  
  
    if ($result->num_rows > 0) {  
  
        // output data of each row  
        while($row = $result->fetch_assoc()) {  
            echo "data.addRow([new Date(".$row["year"].",".$row["month"]."-  
1, ".$row["day"]."),  
            ".$row["AVG(Speed)"].",".$row["MAX(Speed)"].",".$row["MIN(Speed)"].",".$row["  
AVG(Altitude)"].",".$row["MAX(Altitude)"].",".$row["MIN(Altitude)"]."]); \n";  
        }  
    }  
    else {  
    }  
?>
```

Η κάθε μεταβλητή περιέχει τις επιλογές απεικόνισής της .Όπως ο τίτλος, η μονάδα μέτρησης και ο τρόπος που θα παρουσιάζεται.

```
var Speed_options = {
  title: 'Speed Last Month',
  subtitle: 'measured in km/h',
  hAxis: { format: 'MMM dd, yyyy' },
  seriesType: 'bars',
  width: 600,
  height: 300
};
```

```
var Altitude_options = {
  title: 'Altitude Last Month',
  subtitle: 'measured in meters',
  hAxis: { format: 'MMM dd, yyyy' },
  seriesType: 'bars',
  width: 600,
  height: 300
};
```

Δηλώνεται σε ποιες στήλες θα αντιστοιχεί η κάθε μεταβλητή. Σε αυτή την περίπτωση δηλώνονται σε κάθε μεταβλητή τέσσερις στήλες που περιέχουν την ημερομηνία, την ελάχιστη, μέγιστη και μέση τιμή.

```
var Speed_chart = new
google.visualization.ComboChart(document.getElementById('Speed_div'));
```

```
var Speed_view = new google.visualization.DataView(data);
Speed_view.setColumns([0,1,2,3]);
Speed_chart.draw(Speed_view, Speed_options);
```

```
//-----
var Altitude_chart = new
google.visualization.ComboChart(document.getElementById('Altitude_div'));
```

```
var Altitude_view = new google.visualization.DataView(data);
Altitude_view.setColumns([0,4,5,6]);
Altitude_chart.draw(Altitude_view, Altitude_options);
```

```
//-----
```

```
}
```

Μετά το τέλος του javascript υπάρχει το υπόλοιπο κομμάτι κώδικα html που εκτός από το να δίνει ένα τίτλο στη σελίδα παρουσιάζει και τα γραφικά μέσα σε έναν πίνακα για καλύτερη στοίχιση.

```
</script>
</head>
```



```

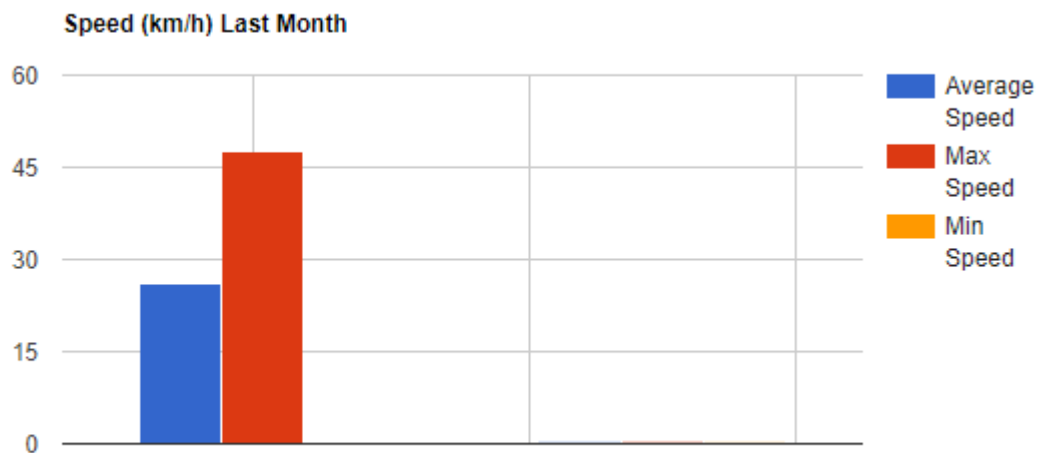
<title>Monthly </title>
<body style="background: # 001433">
  <div style="width:100%">

    <div style="width:90%;font-size:30pt;text-align:center;color:#
cca300;text-align:left;"><strong>Gps Real-Time Tracking</div>
    </div>

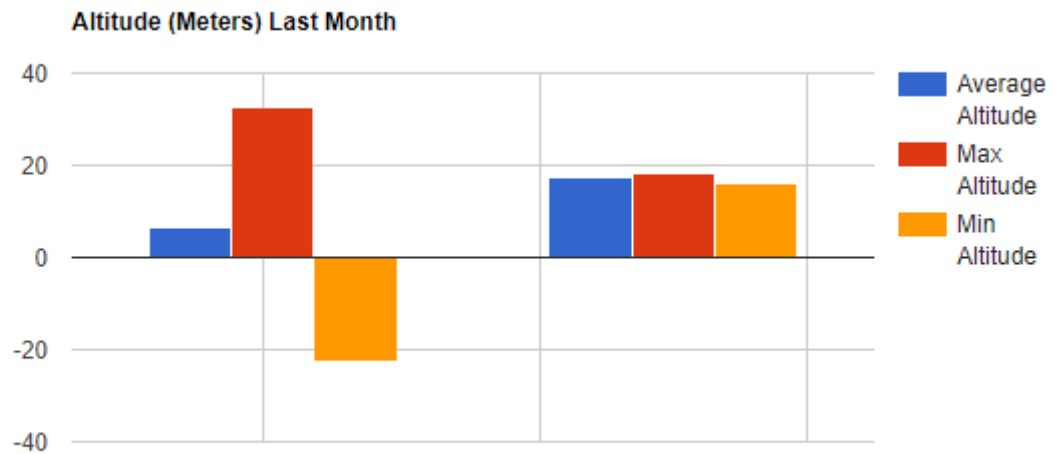

  <div style="width:100%;font-size:19pt;color:#ffffff;text-
align:left;">Μηνιαίες Μετρήσεις
  </div>
  <div align="right">
  <br>
  </div>

  <table style="border-spacing: 15px">
    <tr>
      <td style="border: 1px solid white">
        <div id="Speed_div" style="width: 600px; height:
300px"></div>
      </td>
      <td style="border: 1px solid white">
        <div id="Altitude_div" style="width: 600px; height:
300px"></div>
      </td>
    </tr>
  </table>
</body>
</html>

```



Εικόνα 45 : Απεικόνιση της ελάχιστης, μέγιστης και μέσης τιμής της ταχύτητας.



Εικόνα 46 : Απεικόνιση της ελάχιστης, μέγιστης και μέσης τιμής του υψομέτρου.

Θα πρέπει να σημειωθεί ότι οι τιμές εβδομάδας και μήνα είναι ίδιες καθώς όλες οι μετρήσεις έχουν «γραφτεί» τον ίδιο μήνα. Παρατηρείτε ότι η ελάχιστη τιμή της ταχύτητας είναι μηδενική για αυτό και δεν απεικονίζεται με την κίτρινη μπάρα στο γράφημα. Επίσης στην απεικόνιση του υψόμετρου παρατηρούνται αρνητικές τιμές, ο λόγος είναι ότι ο άνθρωπος ή το κατοικίδιο βρίσκονται σε σημείο που το υψόμετρό του είναι κάτω από το επίπεδο της θάλασσας.

3.5 Δημιουργία χάρτη

Η παρούσα ενότητα θα ασχοληθεί με την σχεδίαση του χάρτη (google map) και την παρουσίαση των σημαδιών (markers) που θα οπτικοποιούν τις μετρήσεις που δέχεται η βάση δεδομένων. Αρχικά θα συνδεθεί η υπηρεσία χαρτογράφησης με την βάση δεδομένων χρησιμοποιώντας τη χρήση ενός αρχείου php. Έπειτα θα αναλυθεί το αρχείο html που δημιουργεί τον χάρτη και ανανεώνεται δυναμικά αναλόγως τον χρόνο που θα ρυθμιστεί. Η υπηρεσία χαρτογράφησης έχει την δυνατότητα αναγνώρισης javascript κάτι που είναι εξαιρετικά χρήσιμο για την εύκολη παραμετροποίηση των χαρτών βάση των αναγκών του χρήστη.[16]



Εικόνα 47 : Το λογότυπο των χαρτών της Google

3.5.1 Αρχεία php και html για την παρουσίαση του χάρτη

«connection.php»

Όπως προϋποθέτει το όνομά του, το παρακάτω αρχείο δηλώνει τα στοιχεία της βάσης δεδομένων. Στην συνέχεια θα καλεστεί από τα αρχεία php για την σύνδεση με το rhrmyadmin.

```
<?php
$username="root";
$password="goopy3121!!";
$database="ody";
?>
```

«xml.php»

Η μεταφορά των δεδομένων της βάσης στην σελίδα των χαρτών δεν μπορεί να γίνει άμεσα. Θα πρέπει τα στοιχεία της βάσης δεδομένων να μετατραπούν σε ένα αρχείο με Xml περιεχόμενο. Η συγκεκριμένη κωδικοποίηση έχει αναλυθεί σε προηγούμενη ενότητα.

Αρχικά εισάγονται τα στοιχεία του παραπάνω php αρχείου.

```
<?php
require("connection.php");
```

Δημιουργείται η συνάρτηση «parseToXML» που φτιάχνει τα απαραίτητα string.

```
function parseToXML($htmlStr)
{
$xmlStr=str_replace('<','&lt;',$htmlStr);
$xmlStr=str_replace('>','&gt;',$xmlStr);
$xmlStr=str_replace('"','&quot;',$xmlStr);
$xmlStr=str_replace("'",'&#39;',$xmlStr);
$xmlStr=str_replace('&','&amp;',$xmlStr);
return $xmlStr;
}
```

Πραγματοποιείται σύνδεση με το rhrmyadmin και επιλογή της κατάλληλης βάσης.

```
$connection=mysql_connect ('localhost', $username, $password);
if (!$connection) {
    die('Not connected : ' . mysql_error());
}

$db_selected = mysql_select_db($database, $connection);
if (!$db_selected) {
    die ('Can\'t use db : ' . mysql_error());
}
```

Επιλέγονται όλες οι στήλες του πίνακα «markers» όπου έχουν αποθηκευτεί τα δεδομένα με την εντολή «`mysql_query`».

```
$query = "SELECT * FROM markers WHERE 1";
$result = mysql_query($query);
if (!$result) {
    die('Invalid query: ' . mysql_error());
}
```

Δημιουργείται η κεφαλίδα του xml και εκτελείτε η εντολή `echo` για να μεταφερθούν τα δεδομένα του πίνακα.

```
header("Content-type: text/xml");

echo "<?xml version='1.0' ?>";
echo '<markers>';
$ind=0;
```

Δηλώνονται οι στήλες της βάσης που μετατρέπονται σε αρχείο xml. Να σημειωθεί ότι δεν είναι υποχρεωτικό να δηλωθούν όλες οι στήλες, στην περίπτωση μας χρειάζονται μόνο το γεωγραφικό πλάτος και μήκος.

```
while ($row = @mysql_fetch_assoc($result)){

    echo '<marker ' ;
    echo 'id="' . $row['id'] . ' ' ;
    echo 'Datetime="' . parseToXML($row['Datetime']) . ' ' ;
    echo 'Lat="' . $row['Lat'] . ' ' ;
    echo 'Lon="' . $row['Lon'] . ' ' ;
    echo 'Speed="' . $row['Speed'] . ' ' ;
    echo 'Satellites="' . $row['Satellites'] . ' ' ;
    echo 'Altitude="' . $row['Altitude'] . ' ' ;
    echo 'Text1="' . $row['Text1'] . ' ' ;
    echo 'Text2="' . $row['Text2'] . ' ' ;
    echo 'Text3="' . $row['Text3'] . ' ' ;
    echo 'Temperature="' . $row['Temperature'] . ' ' ;
    echo 'Humidity="' . $row['Humidity'] . ' ' ;
    echo '>';
    $ind = $ind + 1;
}
```

Τερματισμός επικοινωνίας.

```
echo '</markers>';

?>
```

```

▼<markers>
<marker id="142" Datetime="2018-05-02 17:28:53" Lat="38.229209" Lon="21.741458" Speed="0.00" Satellites="8" Altitude="-22.60" Text1="LAT=" Text2="LON=" Text3="-" Temperature="0.00" Humidity="0.00"/>
<marker id="141" Datetime="2018-05-02 17:28:33" Lat="38.229103" Lon="21.741477" Speed="16.65" Satellites="8" Altitude="-18.30" Text1="LAT=" Text2="LON=" Text3="-" Temperature="0.00" Humidity="0.00"/>
<marker id="140" Datetime="2018-05-02 17:28:13" Lat="38.227714" Lon="21.744068" Speed="27.35" Satellites="7" Altitude="-10.50" Text1="LAT=" Text2="LON=" Text3="-" Temperature="0.00" Humidity="0.00"/>
<marker id="139" Datetime="2018-05-02 17:27:53" Lat="38.226062" Lon="21.747627" Speed="27.35" Satellites="7" Altitude="-6.50" Text1="LAT=" Text2="LON=" Text3="-" Temperature="0.00" Humidity="0.00"/>
<marker id="138" Datetime="2018-05-02 17:27:32" Lat="38.225761" Lon="21.747692" Speed="0.13" Satellites="7" Altitude="-2.10" Text1="LAT=" Text2="LON=" Text3="-" Temperature="0.00" Humidity="0.00"/>
<marker id="137" Datetime="2018-05-02 17:27:12" Lat="38.225036" Lon="21.747968" Speed="35.28" Satellites="7" Altitude="4.20" Text1="LAT=" Text2="LON=" Text3="-" Temperature="0.00" Humidity="0.00"/>
<marker id="136" Datetime="2018-05-02 17:26:52" Lat="38.224288" Lon="21.749303" Speed="35.17" Satellites="7" Altitude="15.40" Text1="LAT=" Text2="LON=" Text3="-" Temperature="0.00" Humidity="0.00"/>
<marker id="135" Datetime="2018-05-02 17:26:32" Lat="38.222061" Lon="21.750104" Speed="47.60" Satellites="7" Altitude="21.20" Text1="LAT=" Text2="LON=" Text3="-" Temperature="0.00" Humidity="0.00"/>
<marker id="134" Datetime="2018-05-02 17:26:12" Lat="38.220474" Lon="21.750339" Speed="0.02" Satellites="7" Altitude="22.20" Text1="LAT=" Text2="LON=" Text3="-" Temperature="0.00" Humidity="0.00"/>
<marker id="133" Datetime="2018-05-02 17:25:52" Lat="38.220241" Lon="21.750371" Speed="37.15" Satellites="7" Altitude="24.20" Text1="LAT=" Text2="LON=" Text3="-" Temperature="0.00" Humidity="0.00"/>
<marker id="132" Datetime="2018-05-02 17:25:32" Lat="38.219505" Lon="21.750116" Speed="1.57" Satellites="7" Altitude="24.50" Text1="LAT=" Text2="LON=" Text3="-" Temperature="0.00" Humidity="0.00"/>

```

Εικόνα 48 : Μορφή Xml στον browser

«index.html»

Πλέον τα δεδομένα που χρειάζονται έχουν μετατραπεί σε μορφή xml άρα είναι σε κατάλληλη μορφή για να σχεδιαστεί ο χάρτης με τα σημάδια που υπάρχουν στη βάση δεδομένων. Με το αντίστοιχο javascript που παρέχει η google και με μερικές μετατροπές/προσθήκες ο χρήστης μπορεί να δημιουργήσει το χάρτη που επιθυμεί. Να σημειωθεί ότι ο περιηγητής(browser) θα ανανεώνεται δυναμικά ανά τριάντα δευτερόλεπτα με τα νέα σημάδια στην διαδρομή. Παρακάτω παρουσιάζεται το αρχείο που σχεδιάζει το χάρτη.

Δήλωση βασικών στοιχείων html, μορφής περιεχομένου(UTF-8), χρόνου ανανέωσης τριάντα δευτερολέπτων αλλά και τίτλου.

```

<!DOCTYPE html >
<head>
<meta name="viewport" content="initial-scale=1.0, user-scalable=no" />
<meta http-equiv="content-type" content="text/html; charset=UTF-8"/>
<meta http-equiv="refresh" content="30" />
<title>Map </title>

```

Σχεδιασμός μορφής του χάρτη, όπως : ύψος, πλάτος.

```

<style>

#map {
height: 100%;
}

html, body {
height: 100%;
margin: 0;
padding: 0;
}
</style>
</head>
<body>

```

Παρακάτω γίνεται η δημιουργία του χάρτη, τα απαραίτητα στοιχεία που περιέχει είναι :

- Η αρχική μεγέθυνση που θα επιβάλλει ο περιηγητής στο χάρτη.

- Σε ποιο σημείο θα κεντράρει ο χάρτης με το αρχικό άνοιγμα ή μια ανανέωση της σελίδας.

```
<div id="map"></div>
```

```
<script>
```

```
function initMap() {  
var map = new google.maps.Map(document.getElementById('map'), {  
center: new google.maps.LatLng(38.23, 21.74),  
zoom: 14  
});
```

Δημιουργείται το μονοπάτι από το οποίο καλείται το αρχείο που περιέχει μορφή xml(xml.php).

```
var path = []; // global variable to hold all the past locations  
var infoWindow = new google.maps.InfoWindow;
```

```
downloadUrl('xml.php', function(data) {  
var xml = data.responseXML;
```

Παρακάτω δηλώνονται οι μεταβλητές που περιέχονται στην βάση δεδομένων, τα απαραίτητα στοιχεία είναι το γεωγραφικό πλάτος και μήκος.

```
var markers = xml.documentElement.getElementsByTagName('marker');  
Array.prototype.forEach.call(markers, function(markerElem) {  
var id = markerElem.getAttribute('id');  
var Datetime = markerElem.getAttribute('Datetime');  
var Speed = markerElem.getAttribute('Speed');  
var Satellites = markerElem.getAttribute('Satellites');  
var Altitude = markerElem.getAttribute('Altitude');  
var data1 = markerElem.getAttribute('Text1');  
var data2 = markerElem.getAttribute('Lat');  
var data3 = markerElem.getAttribute('Text3');  
var data4 = markerElem.getAttribute('Text2');  
var data5 = markerElem.getAttribute('Lon');
```

Η μεταβλητή «point» δημιουργεί το σημάδι με βάση το Lat και Lon.

```
var point = new google.maps.LatLng(  
parseFloat(markerElem.getAttribute('Lat')),  
parseFloat(markerElem.getAttribute('Lon')));
```

Δηλώνονται τα στοιχεία που επιθυμεί χρήστης να περιέχονται στο κουτί λεπτομερειών του σημαδιού.

```
var infowincontent = document.createElement('div');  
var strong = document.createElement('strong');  
strong.textContent = Datetime  
infowincontent.appendChild(strong);  
infowincontent.appendChild(document.createElement('br'));
```

Συγκεκριμένα θα συμπεριληφθούν:

- Η ημερομηνία και η ώρα.

- Το γεωγραφικό μήκος.
- Το γεωγραφικό πλάτος.
- Οι απαραίτητοι χαρακτήρες για τον διαχωρισμό των τιμών.

```

var text = document.createElement('text');
text.textContent = data1
infowincontent.appendChild(text);
var text2 = document.createElement('text');
text2.textContent = data2
infowincontent.appendChild(text2);
var text3 = document.createElement('text');
text3.textContent = data3
infowincontent.appendChild(text3);
var text4 = document.createElement('text');
text4.textContent = data4
infowincontent.appendChild(text4);
var text5 = document.createElement('text');
text5.textContent = data5
infowincontent.appendChild(text5);

```

Το σημάδι δημιουργείται και προστίθεται η λειτουργία να ανοίγει ένα παράθυρο με πληροφορίες κάθε φορά που ο χρήστης επιλέγει να πατήσει σε ένα σημάδι. Οι πληροφορίες αυτές είναι διαφορετικές για κάθε σημάδι και προέρχονται από την βάση δεδομένων.

```

var marker = new google.maps.Marker({
  map: map,
  position: point,
  url: "charts/index.php"
});
marker.addListener('click', function() {
  infoWindow.setContent(infowincontent);
  infoWindow.open(map, marker);
  window.open(marker.url);
});
});
});
}

```

Συμπληρωματική συνάρτηση που χρειάζεται για την σωστή απεικόνιση του χάρτη στον περιηγητή.

```

function downloadUrl(url, callback) {
  var request = window.ActiveXObject ?
    new ActiveXObject('Microsoft.XMLHTTP') :
    new XMLHttpRequest;

  request.onreadystatechange = function() {
    if (request.readyState == 4) {
      request.onreadystatechange = doNothing;
      callback(request, request.status);
    }
  };
}

```

```

    request.open('GET', url, true);
    request.send(null);
}
function doNothing() {}
</script>

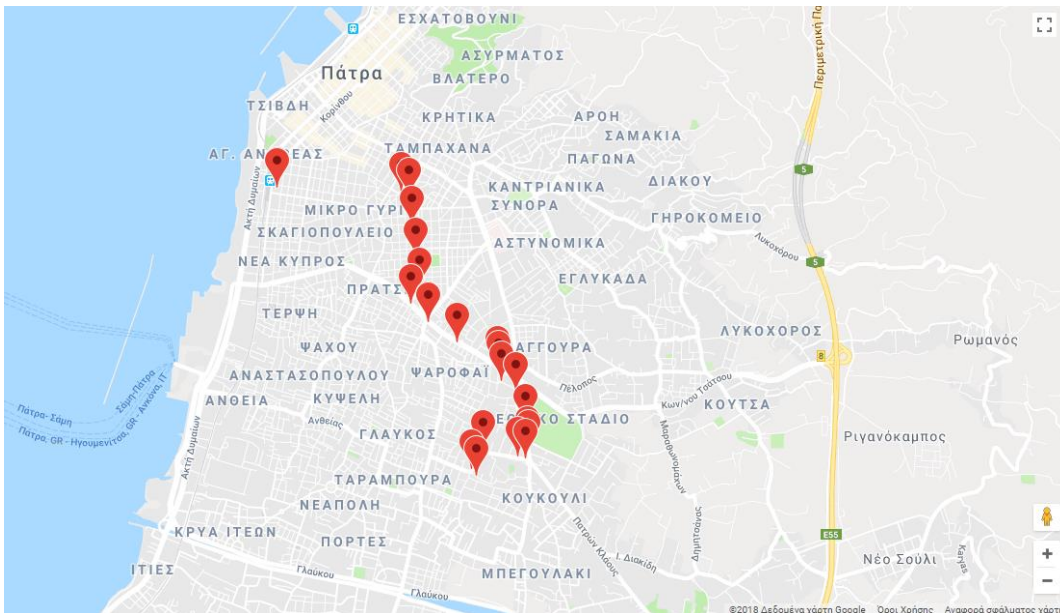
```

Τέλος χρησιμοποιείται το javascript της google για να δηλωθεί το προσωπικό api-key που μπορεί να αποκτηθεί δωρεάν κάτω από συγκεκριμένες προϋποθέσεις μέσω του προσωπικού email της google.

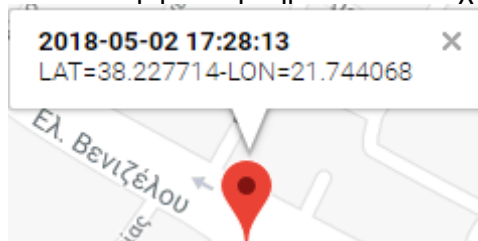
```

<script async defer
src="https://maps.googleapis.com/maps/api/js?key=AIzaSyADHR5AbFtlf_OiODUT8I6I
p-cpOyPVeIw&callback=initMap">
</script>
</body>
</html>

```



Εικόνα 49: Εμφάνιση σημαδιών στον χάρτη



Εικόνα 50: Εμφάνιση σημαδιού με παράθυρο πληροφοριών

3.6 Ιστοχώρος που υλοποιήθηκε και λειτουργίες του

Στην ενότητα που ακολουθεί θα παρουσιαστεί η κεντρική σελίδα του ιστοχώρου που έχει δημιουργηθεί. Μέσα σε αυτή τη σελίδα περιέχονται σύνδεσμοι που ανακατευθύνουν το χρήστη σε πολλών ειδών στατιστικά και γραφήματα. Οι σύνδεσμοι αυτοί βρίσκονται πίσω από τις αντίστοιχες εικόνες για μια πιο άμεση απεικόνιση. Συγχρόνως υπάρχει ξεχωριστό κουμπί που μεταφέρει το χρήστη στην σελίδα του χάρτη. Ο κώδικας που δημιουργεί αυτά τα αρχεία έχει συμπεριληφθεί και αναλυθεί σε προηγούμενη ενότητα. Ακολουθεί ο κώδικας της σελίδας.

3.6.1 Κεντρική σελίδα ιστοχώρου

«index.php»

Δημιουργία html κώδικα και δήλωση πρόσθετων αρχείων css για χρήση επιπλέον λειτουργιών. Τα αρχεία css έχουν μελετηθεί σε προηγούμενη ενότητα.

```
<!doctype html>
<html>
<head><meta http-equiv="Content-Type" content="text/html; charset=utf-8">

<title>Gps Real-Time Tracking</title>
<link rel="stylesheet" href="fonts.css">
<link rel="stylesheet" href="buttons.css">
<link rel="stylesheet" href="footer.css">
<meta http-equiv="refresh" content="30" />
</head>
```

Προστίθενται χρώμα παρασκήνιου, μέγεθος πλαισίου, τίτλος αλλά και λοιπά στοιχεία εμφάνισης των τμημάτων της σελίδας.

```
<body style="background: #001433">
<div style="width:100%">
<div style="width:90%;font-size:25pt;text-align:center;color:#FFFFFF;text-align:left;"><strong>ΤΕΙ Δυτικής Ελλάδας Τμήμα Ηλεκτρολόγων Μηχανικών</div>
</div>
<div>
<div style="width:100%;font-size:19pt;color:#cca300;text-align:left;">Gps
Realtime Tracking Home Page
```

Προγραμματισμός των δύο φωτογραφιών που βρίσκονται στην δεξιά γωνία της σελίδας.

```


</div>
<br>
<hr>
```



```
<br>
<br>
<br>
<br>
```

Το κουμπί που δημιουργείται παρακάτω περιέχεται μέσα σε ένα τμήμα και μόλις πατηθεί μεταφέρει το χρήστη στην σελίδα «index.html» όπου υπάρχει ο χάρτης.

```
<div style="cursor:pointer; width:95.5%;font-size:23pt;background-
color:#cca300;color:#000000;align:middle; margin:0 fixed;"
<button class="button"
onclick="window.open ('../index.html', '_blank') ">Map&nbsp;&nbsp;&nbsp;</button>
</div>
```

Τέλος με την βοήθεια του «footer.css» δημιουργείται ένα υποσέλιδο το οποίο περιλαμβάνει εικόνες από όλες τις υπηρεσίες που έχουν χρησιμοποιηθεί για την κατασκευή αυτής της εργασίας.

```
<div id="footer">
<div id="footerContainer">





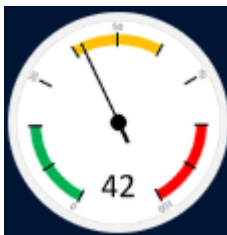



</div>
</div>
</body>
</html>
```

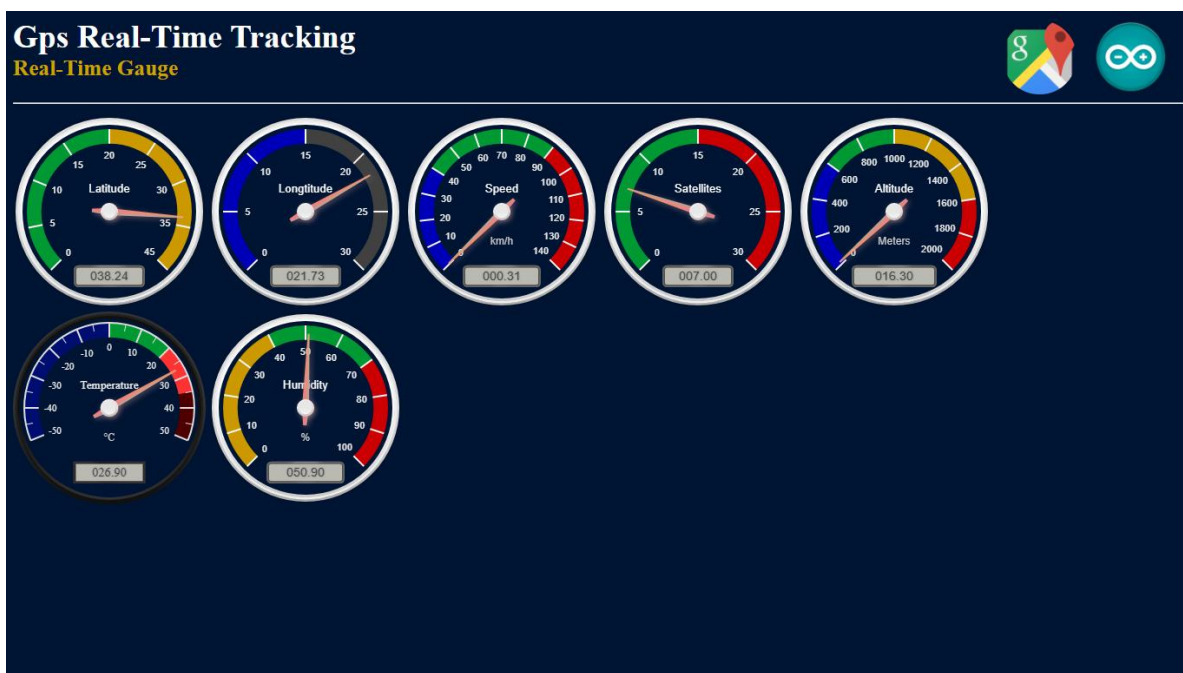


Εικόνα 51: Αρχική σελίδα ιστοχώρου

Πατώντας την παρακάτω εικόνα ο χρήστης μεταφέρεται στην σελίδα των μετρητών.



Εικόνα 52: Σύνδεσμος μετρητών

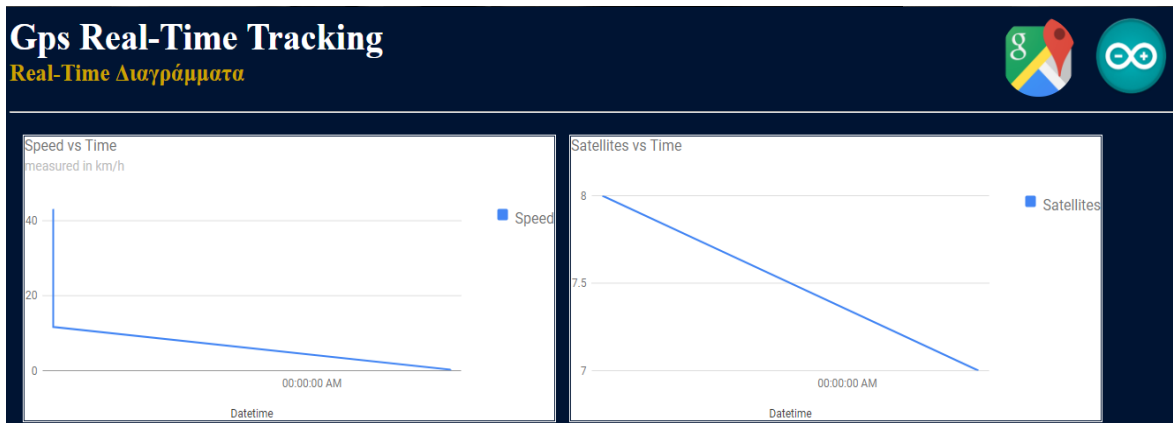


Εικόνα 53: Σελίδα μετρητών

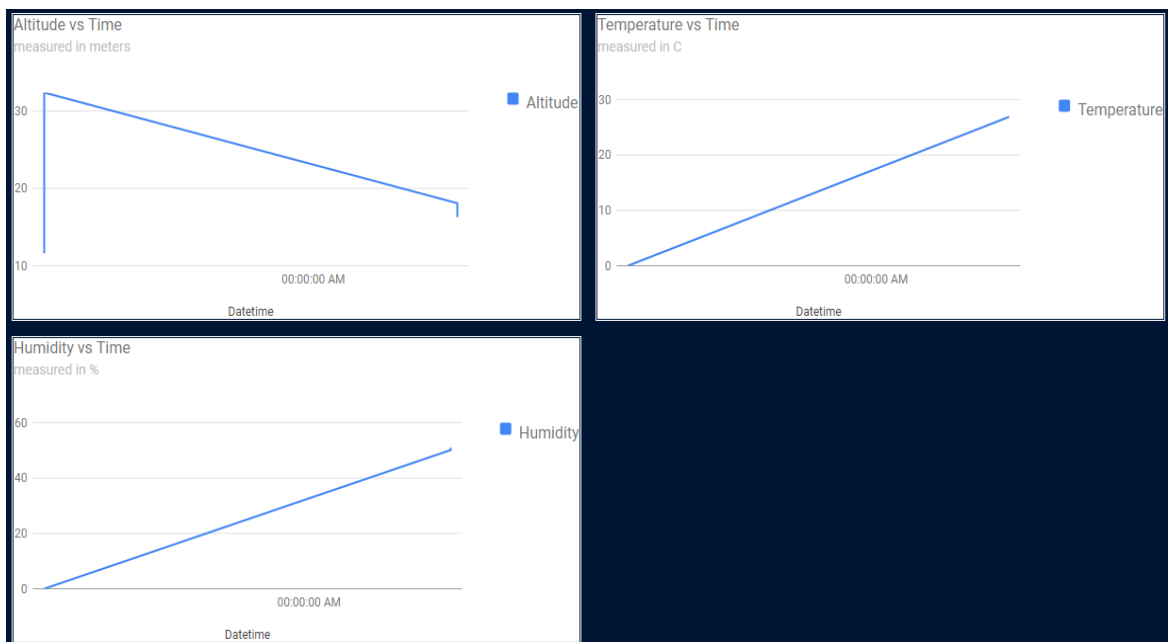
Πατώντας την παρακάτω εικόνα ο χρήστης μεταφέρεται στην σελίδα των στατιστικών.



Εικόνα 54: Σύνδεσμος στατιστικών



Εικόνα 55: Σελίδα στατιστικών

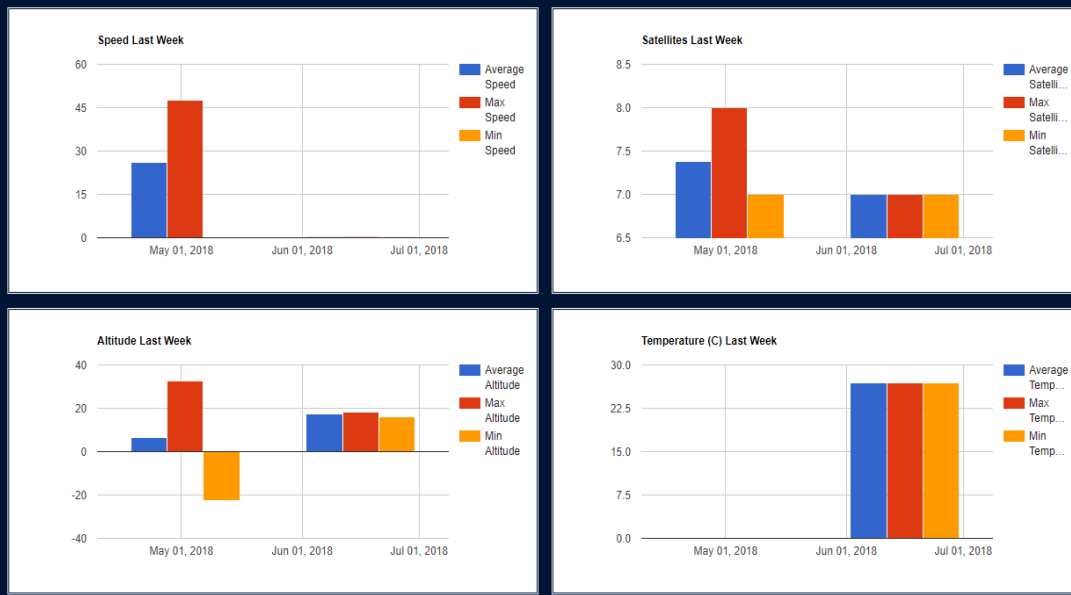


Εικόνα 56: Συνέχεια σελίδας στατιστικών

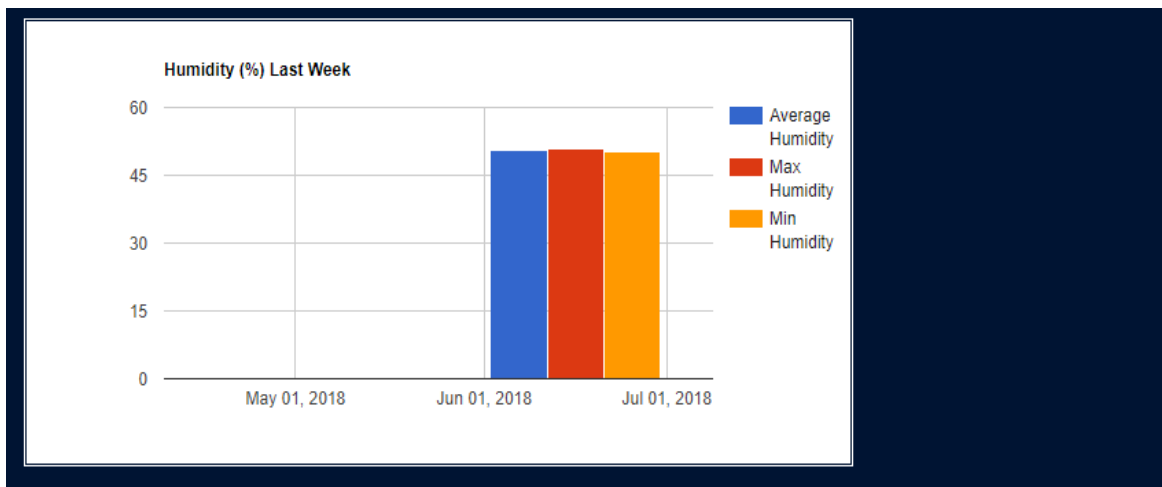
Πατώντας την παρακάτω εικόνα ο χρήστης μεταφέρεται στην σελίδα των εβδομαδιαίων στατιστικών.



Εικόνα 57: Σύνδεσμος σελίδας εβδομαδιαίων στατιστικών

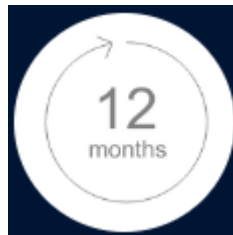


Εικόνα 58: Σελίδα εβδομαδιαίων στατιστικών



Εικόνα 59: Συνέχεια σελίδας εβδομαδιαίων στατιστικών

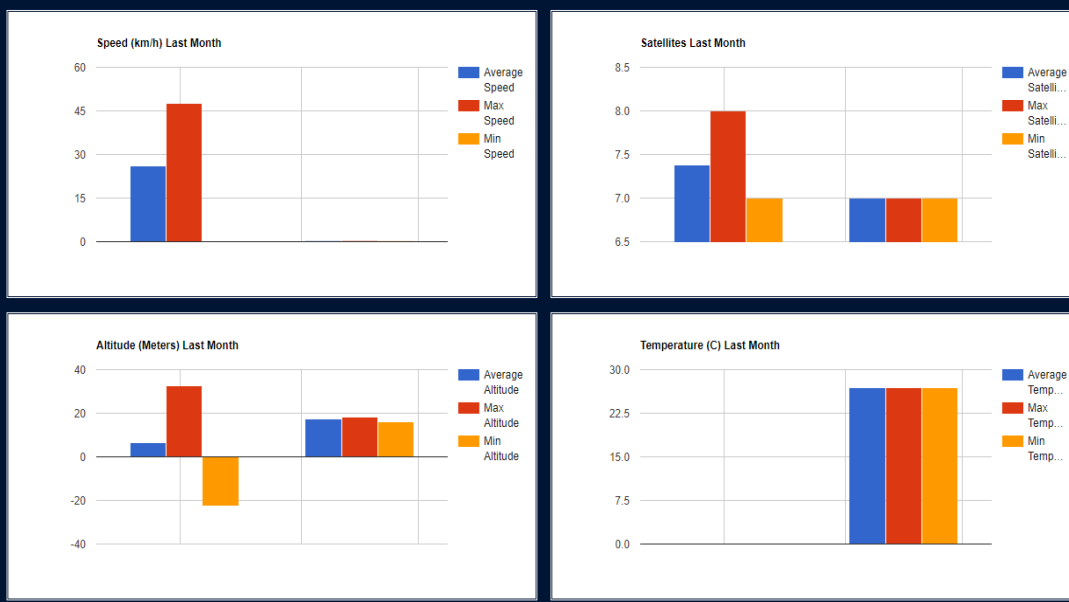
Πατώντας την παρακάτω εικόνα ο χρήστης μεταφέρεται στην σελίδα των μηνιαίων στατιστικών.



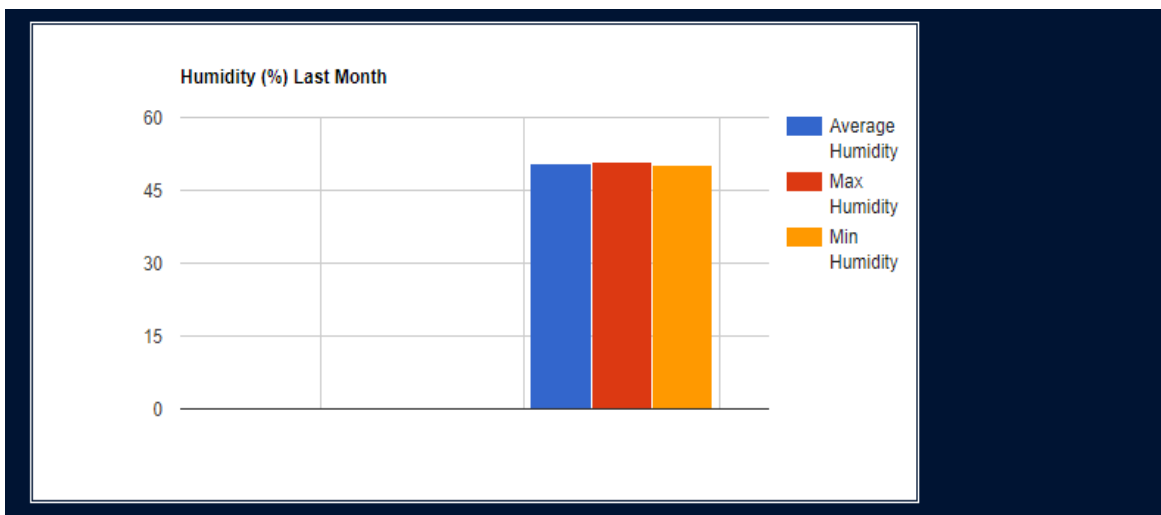
Εικόνα 60: Σύνδεσμος σελίδας μηνιαίων στατιστικών

Gps Real-Time Tracking

Μηνιαίες Μετρήσεις



Εικόνα 61: Σελίδα μηνιαίων στατιστικών



Εικόνα 62: Συνέχεια σελίδας μηνιαίων στατιστικών

Τέλος πατώντας την παρακάτω εικόνα ο χρήστης μεταφέρεται στην σελίδα των χαρτών που παρουσιάστηκε σε παραπάνω ενότητα.

Map

Εικόνα 63 : Σύνδεσμος σελίδας χάρτη σε μορφή κουμπιού

3.6.2 Αρχεία CSS

Τα παρακάτω αρχεία έχουν συμπληρωματικό χαρακτήρα, καθορίζουν την εμφάνιση και την λειτουργία ορισμένων στοιχείων.[3]

«button.css»

Το κουμπί ονομάζεται(button), δηλώνεται το χρώμα, το σημείο που θα προστεθεί, το μέγεθος της γραμματοσειράς που περιέχει και το είδος του δείκτη όταν αυτός θα τοποθετείται από πάνω του.

```
.button{
  background-color: #cca300;
  padding: 15px 32px;
  text-align: center;
  text-decoration: none;
  display: inline-block;
  font-size: 16px;
  cursor:pointer;
}
```

«footer.css»

Το υποσέλιδο δηλώνει σε ποια θέση θα βρίσκεται και ρυθμίζει το ύψος και πλάτος του.

```
#footer {
  position:fixed;
  bottom:0;
  left:0;
  right:0;
  background-color: white;
  width:100%;
  height:80px;
}
```

```
#footerContainer {
  position:fixed;
  width:100%;
  height:70px;
}
```



Εικόνα 64: Εμφάνιση υποσέλιδου

Κεφάλαιο 4 : Συμπεράσματα-μελλοντικές επεκτάσεις

Με την υλοποίηση της πτυχιακής αποκτήθηκαν σημαντικές γνώσεις επάνω στην πλατφόρμα του arduino των περιφερειακών , αλλά και των αισθητήρων του. Ήταν η αφετηρία για την εισαγωγή στον κόσμο του arduino και γενικότερα στην λογική προγραμματισμού του. Η δημιουργία του συστήματος που αναλύθηκε στις παραπάνω ενότητες υπήρξε αφορμή για εισαγωγή στις γλώσσες προγραμματισμού php, sql,c,c++. Τέλος εξαιρετικά χρήσιμη ήταν η εμπειρία που αποκτήθηκε όσον αφορά την δημιουργία και επεξεργασία ιστοσελίδων.

Το σύστημα που κατασκευάστηκε έχει μικρή κατανάλωση ρεύματος, βέβαια προϋποθέτει την χρήση δύο πηγών ρεύματος με διαφορετικά volt για την πλακέτα του arduino και 12 την αντίστοιχη του sim808.Το γεγονός αυτό αναγκάζει τον χρήστη να διαθέτει δύο ξεχωριστές μπαταρίες ή ένα φορητό φορτιστή μπαταριών κοινώς powerbank.Όμως το χαμηλό κόστος ολόκληρου του συστήματος σε συνδυασμό με την σύνδεση στο διαδίκτυο το καθιστούν μια καλή επιλογή για την ασύρματη παρακολούθηση της διαδρομής ενός ανθρώπου ή κατοικίδιου .Μεγάλο πλεονέκτημα αποτελεί επίσης το γεγονός ότι μπορεί να γίνει πολύ εύκολα ανάλογη παραμετροποίηση του κώδικα και των υπηρεσιών βάσει των αναγκών του χρήστη.

Η υλοποίηση έχει βασιστεί στις τεχνολογίες 3G/GPS/GPRS οι οποίες είναι ευρέως διαδεδομένες και χρησιμοποιούνται όλες ή κάποιες από αυτές σε πάρα πολλές τεχνολογικές κατασκευές. Με αυτή τη λογική χρησιμοποιήθηκε και η γλώσσα php που αποτελεί βασικό εργαλείο στις περισσότερες ενότητες της πτυχιακής εργασίας.

Στον αντίποδα το σύστημα και οι υπηρεσίες που δημιουργήθηκαν σίγουρα δέχεται βελτιώσεις και επεκτάσεις. Στη συνέχεια γίνεται αναφορά σε επιπλέον δυνατότητες, λειτουργίες και επεκτάσεις που θα μπορούσαν να συμπεριλαμβάνονται στο σύστημα όσον αφορά πολλούς τομείς.

- Στον τομέα επικοινωνίας και δεδομένων κινητής τηλεφωνίας θα μπορούσε να επιλεγεί μια κατά πολύ ακριβότερη μονάδα όπως η sim908 που υποστηρίζει κάρτες sim τεχνολογίας 4G. Αυτή η αναβάθμιση θα καθιστούσε την μεταφορά δεδομένων ακόμα πιο γρήγορη.
- Στον τομέα της τροφοδοσίας θα μπορούσαν να αγοραστούν οι κατάλληλες μπαταρίες για τις δύο μονάδες με σκοπό την μείωση του όγκου και βάρους του συστήματος. Αυτές θα διευκόλυναν την μετακίνησή και την εφαρμογή του σε μικρότερου μεγέθους κατοικίδια σε σχέση με το βαρύτερο powerbank που χρησιμοποιήθηκε καθώς προϋπήρχε.
- Στον τομέα του προγραμματισμού πάντα υπάρχει περιθώριο για βελτίωση του κώδικα με σκοπό την χρήση λιγότερων πόρων.

- Επίσης θα μπορούσε να δημιουργηθεί μία εφαρμογή ελέγχου για κινητά με λογισμικό Android ή IOS.
- Ο αριθμός των αισθητήρων είναι σχετικά περιορισμένος κάτι που θα μπορούσε να αποφευχθεί με διαφορετική δομή του προγράμματος του arduino. Συγκεκριμένα υπήρξε προσπάθεια για προσθήκη αισθητήρα καρδιακών παλμών αλλά και μετρητή βημάτων. Οι κύριοι λόγοι που αυτό δεν επετεύχθη είναι: η μη ύπαρξη βιβλιοθήκης για τον υπολογισμό της τοποθεσίας και την αποστολή των δεδομένων μέσω ίντερνετ και η αδυναμία σύνδεσης του κυρίως arduino με ένα δευτερεύον για τον παράλληλο υπολογισμό των τιμών των αισθητήρων.
- Τέλος το σύστημα που έχει υλοποιηθεί θα μπορούσε πολύ εύκολα να αποτελεί μέρος μιας ευρύτερης κατασκευής ώστε να της προσθέσει λειτουργίες όπως 3G/GPRS/GPS. Κάποιες τέτοιες εφαρμογές θα μπορούσαν να είναι, ένα drone, μια έξυπνη γλάστρα και γενικότερα όποια εφαρμογή απαιτεί τις παραπάνω τεχνολογίες ή ο χρήστης της θέλει να συμπεριλάβει την ασύρματη σύνδεση με το διαδίκτυο στην υλοποίησή του.

Βιβλιογραφία

Βιβλία:

Ενσωματωμένα Συστήματα, οι μικροελεγκτές AVR και Arduino, Πογαρίδης Δημήτριος , Εκδόσεις Δίσιγμα.

Σ από την θεωρία στην εφαρμογή Γ έκδοση, Δρ Γ. Σ. Τσελίκης | Δρ Ν. Δ. Τσελίκας.

Ηλεκτρονικά Ι-Εισαγωγή στα Ηλεκτρονικά, Ι. Χαριτάντης, Πανεπιστημιακές Εκδόσεις Αράκυνθος, 2006.

Arduino Projects Book, Arduino

Διαδίκτυο:

[1] HTML, <https://www.w3schools.com/Html/>

[2] PHP, <https://www.w3schools.com/pHP/default.asp>

[3] CSS, <https://www.w3schools.com/CSS/>

[4] Arduino, <https://www.cooking-hacks.com/documentation/tutorials/geolocation-tracker-gprs-gps-geoposition-sim908-arduino-raspberry-pi/>

[5] Arduino, <http://arduiniana.org/libraries/tinygps/>

- [6] Arduino, <https://forum.arduino.cc/index.php?topic=116867.0>
- [7] Arduino, <http://www.circuitstoday.com/pulse-sensor-arduino>
- [8] Arduino IDE, <https://www.arduino.cc/en/Main/Software?>
- [9] Arduino,sim808, www.stackoverflow.com
- [10] Dht22, <https://create.arduino.cc/projecthub/attari/temperature-monitoring-with-dht22-arduino-15b013>
- [11] Mysql , <https://www.mysql.com/>
- [12] Sim808, [https://www.waveshare.com/wiki/GSM/GPRS/GPS_Shield_\(B\)](https://www.waveshare.com/wiki/GSM/GPRS/GPS_Shield_(B))
- [13] Sim808, <http://www.raviyp.com/embedded/194-sim900-gprs-http-at-commands3>
- [14] Notepad++, <https://notepad-plus-plus.org/download/v7.5.6.html>
- [15] Xampp, <https://www.apachefriends.org/index.html>
- [16] Google maps, <https://developers.google.com/maps/documentation/javascript/mysql-to-maps>
- [17] Google charts, <https://developers.google.com/chart/>
- [18] Html, <https://en.wikipedia.org/wiki/HTML>
- [19] Php, <https://el.wikipedia.org/wiki/PHP>
- [20] Javascript, <https://en.wikipedia.org/wiki/JavaScript>
- [21] Mysql, <https://en.wikipedia.org/wiki/MySQL>

Παράρτημα Α

Κώδικας προγραμματισμού μικροελεγκτή

«gps.ino»

```
#include <SoftwareSerial.h>
#include <TinyGPS.h>
#include "DHT.h"
#define DHTPIN 9
#define DHTTYPE DHT22
DHT dht (DHTPIN,DHTTYPE) ;
#define button 7
TinyGPS gps;
SoftwareSerial ss(2, 3);
unsigned long fix_age, time, date, speed, course;
```

```

void printFloat(double f, int digits = 2);

void setup()
{
    pinMode(button, INPUT_PULLUP);
    Serial.begin(115200);

    ss.begin(9600);
}
void loop()
{
    bool newData = false;
    ss.println("AT");
    ss.println("AT+CGNSPWR=1");
    ss.println("AT+CGNSTST=1");

    for (unsigned long start = millis(); millis() - start < 1000;)
    {
        while (ss.available())
        {
            char c = ss.read();
            //Serial.write(c); //gps nmea string
            if (gps.encode(c))
                newData = true;
        }
    }
    ////////////////GPS//////////////////////////////////////
    if (newData)
    {
        float latitude, longitude;
        unsigned long age;
        float fkmph;
        int year;
        byte month, day, hour, minute, second, hundredths;
        gps.f_speed_kmph();
        gps.f_get_position(&latitude, &longitude, &age);
        gps.crack_datetime(&year, &month, &day, &hour, &minute, &second, &hundredths,
&age);
        Serial.print("LAT=");
        Serial.print(latitude == TinyGPS::GPS_INVALID_F_ANGLE ? 0.0 : latitude, 6);
        Serial.print(" LON=");
        Serial.print(longitude == TinyGPS::GPS_INVALID_F_ANGLE ? 0.0 : longitude, 6);
        Serial.print(" SAT=");
        Serial.print(gps.satellites() == TinyGPS::GPS_INVALID_SATELLITES ? 0 :
gps.satellites());
        Serial.print("Date: "); Serial.print(static_cast<int>(month)); Serial.print("/");
        Serial.print(static_cast<int>(day)); Serial.print("/"); Serial.print(year);
        Serial.print(" Time: "); Serial.print(static_cast<int>(hour+3));
Serial.print(":");
        Serial.print(static_cast<int>(minute)); Serial.print(":");
Serial.print(static_cast<int>(second));
        Serial.print("."); Serial.print(static_cast<int>(hundredths));
        Serial.print("Alt: "); Serial.print(gps.f_altitude());
        Serial.print(" (kmph): "); printFloat(gps.f_speed_kmph());
        Serial.println();

        ////////////////Ημερομηνία και ώρα σε String//////////////////////////////////////

        String mystring1=String(year);
        String mystring2=String(month);

```

```

String mystring3=String(day);
String mystring4=String(hour+3);
String mystring5=String(minute);
String mystring6=String(second);
String datetime= mystring1 + "-" + mystring2 + "-" + mystring3 + "/" + mystring4 +
":" + mystring5 + ":" + mystring6;
Serial.println(datetime);

////////Για Απεικόνιση στους Χάρτες //////////

String Text1 = "LAT=";
String Text2 = "LON=";
String Text3 = "-";

//Μήνυμα τις Συντεταγμένες Αν Πατηθεί Το Panic Button//////////

static uint8_t lastBtnState = HIGH;
uint8_t state = digitalRead(button);
if (state != lastBtnState) {
    lastBtnState = state;
    if (state == LOW) {
////////Μεταβλητές σε String//////////
        String lat=String(latitude == TinyGPS::GPS_INVALID_F_ANGLE ? 0.0 : latitude, 6);
        String lon=String(longtitude == TinyGPS::GPS_INVALID_F_ANGLE ? 0.0 : longtitude,
6);
        delay(100);
ss.println("AT+CMGF=1");
        delay(500);
ss.println("AT+CMGS=\"+306949465421\"");
        delay(500);
ss.println("SOS");
ss.println("Latitude= ");
ss.print(lat);
ss.println(" ");
ss.println("Longitude= ");
ss.print(lon);
ss.println("http://maps.google.com/maps?q=loc:");
ss.print(lat);
ss.print(",");
ss.print(lon);
delay(200);
ss.write(26);
Serial.println("Sms Sent !");
    }
}
////////Μέτρηση Απόστασης και μήνυμα//////////
latitude == TinyGPS::GPS_INVALID_F_ANGLE ? 0.0 : latitude, 6;
longtitude == TinyGPS::GPS_INVALID_F_ANGLE ? 0.0 : longtitude, 6;
float X = ((21.728059 - longtitude) * 24301 * cos(latitude)) / 360.0;
float Y = ((38.238707 - latitude) * 24280) / 360.0;
float X2 = X * X;
float Y2 = Y * Y;
float distance = sqrt(X2 + Y2);
float km = distance*1.609344;
float met = km*1000;
Serial.print(met);
if (met>3000){
String meters = String(met);
delay(100);
ss.println("AT+CMGF=1");
        delay(500);
ss.println("AT+CMGS=\"+306949465421\"");
        delay(500);
}

```

```

ss.println("Dog is away from home ");
ss.print(meters);
ss.println("meters");
delay(200);
ss.write(26);
}

////////Θερμοκρασία-Υγρασία////////
float hum=dht.readHumidity();
float temp=dht.readTemperature();
if (hum>=60 & temp>=33)
{
    String temps = String(temp);
    String hums = String(hum);
    delay(100);
ss.println("AT+CMGF=1");
    delay(500);
ss.println("AT+CMGS=\"+306949465421\"");
    delay(500);
ss.println("Weather Conditions: uncomfotable ");
ss.println("Temperature= ");
ss.println(temps);
ss.println("Humidity= ");
ss.print(hums);
delay(200);
ss.write(26);

}

//////////GSM//////////
delay(1000);
ss.begin(9600);
    Serial.begin(115200);
    Serial.println("Sending Data...");
ss.flush();
    Serial.flush();
ss.println("AT+CGATT?");
    delay(100);
ss.println("AT+SAPBR=3,1,\"CONTYPE\",\"GPRS\"");
    delay(2000);
ss.println("AT+SAPBR=3,1,\"APN\",\"internet.vodafone.gr\"");
    delay(2000);
ss.println("AT+SAPBR=1,1");
    delay(2000);
ss.println("AT+HTTTPINIT");
    delay(2000);
ss.print("AT+HTTTPARA=\"URL\", \"http://195.251.14.234/arduino3/data.php?");
ss.print("data1=");
ss.print(datetime);
ss.print("&data2=");
ss.print(latitude == TinyGPS::GPS_INVALID_F_ANGLE ? 0.0 : latitude, 6);
ss.print("&data3=");
ss.print(longtitude == TinyGPS::GPS_INVALID_F_ANGLE ? 0.0 : longtitude, 6);
ss.print("&data4=");
ss.print(gps.f_speed_kmph());
ss.print("&data5=");
ss.print(gps.satellites() == TinyGPS::GPS_INVALID_SATELLITES ? 0 : gps.satellites());
ss.print("&data6=");
ss.print(gps.f_altitude());
ss.print("&data7=");
ss.print(Text1);
ss.print("&data8=");

```

```

ss.print(Text2);
ss.print("&data9=");
ss.print(Text3);
ss.print("&data10=");
ss.print(temp);
ss.print("&data11=");
ss.print(hum);
ss.print("");
ss.println("\n");
delay(2000);
ss.println("AT+HTTPACTION=0");
    delay(5000);
ss.println("AT+HTTPREAD");
    delay(1000);
ss.println("");
ss.println("AT+HTTPTERM");
    delay(300);
ss.println("");
Serial.print("Data Sent");
    delay(1000);

}
}
////////Διαχείριση Αρνητικών αριθμών////////
void printFloat(double number, int digits)
{
    if (number < 0.0) {
        Serial.write('-');
        number = -number;
    }
    //////////Σωστή Απεικόνιση Στρογγυλοποίησης/////
    double rounding = 0.5;
    for (uint8_t i=0; i<digits; ++i)
        rounding /= 10.0;
    number += rounding;
    //////////Τύπωσε το Ακέραιο Κομμάτι////////
    unsigned long int_part = (unsigned long)number;
    double remainder = number - (double)int_part;
    Serial.print(int_part);
    //////////Τύπωσε Δεκαδικά Ψηφία Αν υπάρχουν////////
    if (digits > 0)
        Serial.print(".");
    //////////Τύπωσε Τα ψηφία Ένα Τη Φορά////////
    while (digits-- > 0) {
        remainder *= 10.0;
        int toPrint = int(remainder);
        Serial.print(toPrint);
        remainder -= toPrint;
    }
}
}

```

Αρχεία Βάσης-Χάρτη

«data.php»

```

<?php
$dbusername = "root";
$dbpassword = "goopy3121!!";
$server = "localhost";
$My_db = "ody";
// Connect to your database

```

```

$dbconnect = mysql_pconnect($server, $dbusername, $dbpassword);
$dbselect = mysql_select_db("ody", $dbconnect);
$sql = "INSERT INTO markers (Datetime, Lat, Lon, Speed, Satellites, Altitude,
Text1, Text2, Text3, Temperature, Humidity) VALUES
('".$_GET["data1"]."', '".$_GET["data2"]."', '".$_GET["data3"]."', '".$_GET["data4"]."', '
".$_GET["data5"]."', '".$_GET["data6"]."', '".$_GET["data7"]."', '".$_GET["data8"]."', '
".$_GET["data9"]."', '".$_GET["data10"]."', '".$_GET["data11"]."'');";
// Execute SQL statement
mysql_query($sql);
?>

```

«connection.php»

```

<?php
$username="root";
$password="goopy3121!!";
$database="ody";
?>

```

«xml.php»

```

<?php
require("connection.php");

function parseToXML($htmlStr)
{
$xmlStr=str_replace('<', '&lt;', $htmlStr);
$xmlStr=str_replace('>', '&gt;', $xmlStr);
$xmlStr=str_replace('"', '&quot;', $xmlStr);
$xmlStr=str_replace("'", '&#39;', $xmlStr);
$xmlStr=str_replace('&', '&amp;', $xmlStr);
return $xmlStr;
}

// Opens a connection to a MySQL server
$connection=mysql_connect ('localhost', $username, $password);
if (!$connection) {
die('Not connected : ' . mysql_error());
}

// Set the active MySQL database
$db_selected = mysql_select_db($database, $connection);
if (!$db_selected) {
die ('Can\'t use db : ' . mysql_error());
}

// Select all the rows in the markers table
$query = "SELECT * FROM markers WHERE 1";
$result = mysql_query($query);
if (!$result) {
die('Invalid query: ' . mysql_error());
}

header("Content-type: text/xml");

// Start XML file, echo parent node
echo "<?xml version='1.0' ?>";
echo '<markers>';
$ind=0;
// Iterate through the rows, printing XML nodes for each
while ($row = @mysql_fetch_assoc($result)){

```



```

// Add to XML document node
echo '<marker ' ;
echo 'id="' . $row['id'] . '" ' ;
echo 'Datetime="' . parseToXML($row['Datetime']) . '" ' ;
echo 'Lat="' . $row['Lat'] . '" ' ;
echo 'Lon="' . $row['Lon'] . '" ' ;
echo 'Speed="' . $row['Speed'] . '" ' ;
echo 'Satellites="' . $row['Satellites'] . '" ' ;
echo 'Altitude="' . $row['Altitude'] . '" ' ;
echo 'Text1="' . $row['Text1'] . '" ' ;
echo 'Text2="' . $row['Text2'] . '" ' ;
echo 'Text3="' . $row['Text3'] . '" ' ;
echo 'Temperature="' . $row['Temperature'] . '" ' ;
echo 'Humidity="' . $row['Humidity'] . '" ' ;
echo '>';
$ind = $ind + 1;
}

// End XML file
echo '</markers>';

?>

```

«index.html»

```

<!DOCTYPE html >
<head>
  <meta name="viewport" content="initial-scale=1.0, user-scalable=no" />
  <meta http-equiv="content-type" content="text/html; charset=UTF-8"/>
  <meta http-equiv="refresh" content="30" />
  <title>Map</title>
  <style>

    #map {
      height: 100%;
    }

    html, body {
      height: 100%;
      margin: 0;
      padding: 0;
    }
  </style>
</head>
<body>
  <div id="map"></div>

  <script>
    function initMap() {
      var map = new google.maps.Map(document.getElementById('map'), {
        center: new google.maps.LatLng(38.23, 21.74),
        zoom: 14
      });
      var path = []; // global variable to hold all the past locations
      var infoWindow = new google.maps.InfoWindow;

      downloadUrl('xml.php', function(data) {
        var xml = data.responseXML;
        var markers = xml.documentElement.getElementsByTagName('marker');
        Array.prototype.forEach.call(markers, function(markerElem) {
          var id = markerElem.getAttribute('id');

```

```

        var Datetime = markerElem.getAttribute('Datetime');
        var Speed = markerElem.getAttribute('Speed');
        var Satellites = markerElem.getAttribute('Satellites');
        var Altitude = markerElem.getAttribute('Altitude');
        var data1 = markerElem.getAttribute('Text1');
        var data2 = markerElem.getAttribute('Lat');
        var data3 = markerElem.getAttribute('Text3');
        var data4 = markerElem.getAttribute('Text2');
        var data5 = markerElem.getAttribute('Lon');
        var point = new google.maps.LatLng(
            parseFloat(markerElem.getAttribute('Lat')),
            parseFloat(markerElem.getAttribute('Lon')));
        var infowincontent = document.createElement('div');
        var strong = document.createElement('strong');
        strong.textContent = Datetime
        infowincontent.appendChild(strong);
        infowincontent.appendChild(document.createElement('br'));
        ////////////////Marker info ////////////////////////////
        var text = document.createElement('text');
        text.textContent = data1
        infowincontent.appendChild(text);
        var text2 = document.createElement('text');
        text2.textContent = data2
        infowincontent.appendChild(text2);
        var text3 = document.createElement('text');
        text3.textContent = data3
        infowincontent.appendChild(text3);
        var text4 = document.createElement('text');
        text4.textContent = data4
        infowincontent.appendChild(text4);
        var text5 = document.createElement('text');
        text5.textContent = data5
        infowincontent.appendChild(text5);

        var marker = new google.maps.Marker({
            map: map,
            position: point,
        });
        marker.addListener('click', function() {
            infoWindow.setContent(infowincontent);
            infoWindow.open(map, marker);
        });
    });
});
}

function downloadUrl(url, callback) {
    var request = window.ActiveXObject ?
        new ActiveXObject('Microsoft.XMLHTTP') :
        new XMLHttpRequest;

    request.onreadystatechange = function() {
        if (request.readyState == 4) {
            request.onreadystatechange = doNothing;
            callback(request, request.status);
        }
    };

    request.open('GET', url, true);
    request.send(null);
}

```

```

        function doNothing() {}
    </script>
    <script async defer
        src="https://maps.googleapis.com/maps/api/js?key=AIzaSyADHR5AbFt1f_OiODUT8I6Ip-
cpOyPVeIw&callback=initMap">
    </script>
</body>
</html>

```

Παράρτημα Β

Αρχεία Ιστοχώρου

«gauge.php»

```

<?php
    // Prepare variables for database connection

    $dbusername = "root";
    $dbpassword = "goopy3121!!";
    $server = "localhost";

    // Connect to your database
    $mysqli = new mysqli($server, $dbusername, $dbpassword, "ody");
    if($mysqli->connect_error)
        die('Connect Error (' . mysqli_connect_errno() . ') ' . mysqli_connect_error());

    // Execute SQL statement

    $result = $mysqli->query("SELECT
Lat,Lon,Speed,Temperature,Humidity,Satellites,Altitude FROM ody.markers ORDER BY id
DESC LIMIT 1");

    if ($result->num_rows > 0) {

        // output data of each row
        while($row = $result->fetch_assoc()) {
            $LValue = $row["Lat"];
            $LOValue = $row["Lon"];
            $SValue = $row["Speed"];
            $TValue = $row["Temperature"];
            $HValue = $row["Humidity"];
            $SAValue = $row["Satellites"];
            $AValue = $row["Altitude"];
        }
    }
    else {
        $LValue = 0;
        $LOValue = 0;
        $SValue = 0;
        $TValue = 0;
        $HValue = 0;
        $SAValue = 0;
        $AValue = 0;
    }
?>
<!doctype html>

```

```

<html>
<head><meta http-equiv="Content-Type" content="text/html; charset=iso8859-7">

    <title>Gauge </title>
    <link rel="stylesheet" href="../fonts/fonts.css">
    <link rel="stylesheet" href="buttons.css">
    <link rel="stylesheet" href="footer.css">
    <script src="gauge.js"></script>
    <meta http-equiv="refresh" content="30" />
</head>
<body style="background: #001433">
<div style="width:70%;font-size:28pt;text-align:center;color:#ffffff;text-
align:left;"><strong>Gps Real-Time Tracking </div>


<div style="width:100%;font-size:19pt;color:#cca300;text-align:left;">Real-Time Gauge
<p>
<hr>

    </div>
</div>
</div>

<canvas data-type="radial-gauge"
data-width="222"
data-height="222"
data-units=""
data-title="Latitude"
data-value=<?php echo "\"" . $LValue . "\"\n"?>
data-animate-on-init="false"
data-animated-value="false"
data-min-value="0"
data-max-value="45"
data-major-ticks="0,5,10,15,20,25,30,35,45"
data-minor-ticks="1"
data-stroke-ticks="false"
data-highlights=' [
    { "from": 0, "to": 22.5, "color": "rgba(0, 153, 51,1)" },
    { "from": 22.5, "to":45, "color": "rgba(204, 153, 0,1)" }
]'
data-color-plate="transparent"
data-color-major-ticks="#f5f5f5"
data-color-minor-ticks="#ddd"
data-color-title="#fff"
data-color-units="#ccc"
data-color-numbers="#eee"
data-color-needle-start="rgba(240, 128, 128, 1)"
data-color-needle-end="rgba(255, 160, 122, .9)"
data-value-box="true"
data-animation-rule="bounce"
data-animation-duration="500"
data-border-outer-width="3"
data-border-middle-width="3"
data-border-inner-width="3"
>>/canvas>

<canvas data-type="radial-gauge"
data-width="222"
data-height="222"
data-units=""
data-title="Longitude"

```

```

data-value=<?php echo "\"".$LOValue."\""\n"?>
data-animate-on-init="false"
data-animated-value="false"
data-min-value="0"
data-max-value="30"
data-major-ticks="0,5,10,15,20,25,30"
data-minor-ticks="1"
data-stroke-ticks="false"
data-highlights=' [
    { "from": 0, "to": 15, "color": "rgba(0, 0, 183)" },
    { "from": 15, "to": 30, "color": "rgba(65, 65, 65)" }
]'
data-color-plate="transparent"
data-color-major-ticks="#f5f5f5"
data-color-minor-ticks="#ddd"
data-color-title="#fff"
data-color-units="#ccc"
data-color-numbers="#eee"
data-color-needle-start="rgba(240, 128, 128, 1)"
data-color-needle-end="rgba(255, 160, 122, .9)"
data-value-box="true"
data-animation-rule="bounce"
data-animation-duration="500"
data-border-outer-width="3"
data-border-middle-width="3"
data-border-inner-width="3"
</canvas>

<canvas data-type="radial-gauge"
data-width="222"
data-height="222"
data-units="km/h"
data-title="Speed"
data-value=<?php echo "\"".$SValue."\""\n"?>
data-animate-on-init="false"
data-animated-value="false"
data-min-value="0"
data-max-value="140"
data-major-ticks="0,10,20,30,40,50,60,70,80,90,100,110,120,130,140"
data-minor-ticks="0.1"
data-stroke-ticks="false"
data-highlights=' [
    { "from": 0, "to": 40, "color": "rgba(0, 0, 183)" },
    { "from": 40, "to": 90, "color": "rgba(0, 153, 51,1)" },
    { "from": 90, "to": 140, "color": "rgba(204, 0, 0,1)" }
]'
data-color-plate="transparent"
data-color-major-ticks="#f5f5f5"
data-color-minor-ticks="#ddd"
data-color-title="#fff"
data-color-units="#ccc"
data-color-numbers="#eee"
data-color-needle-start="rgba(240, 128, 128, 1)"
data-color-needle-end="rgba(255, 160, 122, .9)"
data-value-box="true"
data-animation-rule="bounce"
data-animation-duration="500"
data-border-outer-width="3"
data-border-middle-width="3"
data-border-inner-width="3"
</canvas>

<canvas data-type="radial-gauge"
data-width="222"

```

```

data-height="222"
data-units=""
data-title="Satellites"
data-value=<?php echo "\"".$SAValue."\""\n"?>
data-animate-on-init="false"
data-animated-value="false"
data-min-value="0"
data-max-value="30"
data-major-ticks="0,5,10,15,20,25,30"
data-minor-ticks="0.1"
data-stroke-ticks="false"
data-highlights='[
    { "from": 0, "to": 15, "color": "rgba(0, 153, 51,1)" },
    { "from": 15, "to": 30, "color": "rgba(204, 0, 0,1)" }
]'
data-color-plate="transparent"
data-color-major-ticks="#f5f5f5"
data-color-minor-ticks="#ddd"
data-color-title="#fff"
data-color-units="#ccc"
data-color-numbers="#eee"
data-color-needle-start="rgba(240, 128, 128, 1)"
data-color-needle-end="rgba(255, 160, 122, .9)"
data-value-box="true"
data-animation-rule="bounce"
data-animation-duration="500"
data-border-outer-width="3"
data-border-middle-width="3"
data-border-inner-width="3"
<</canvas>
<canvas data-type="radial-gauge"
data-width="222"
data-height="222"
data-units="Meters"
data-title="Altitude"
data-value=<?php echo "\"".$AValue."\""\n"?>
data-animate-on-init="false"
data-animated-value="false"
data-min-value="0"
data-max-value="2000"
data-major-ticks="0,200,400,600,800,1000,1200,1400,1600,1800,2000"
data-minor-ticks="0.1"
data-stroke-ticks="false"
data-highlights='[
    { "from": 0, "to": 600, "color": "rgba(0, 0, 183)" },
    { "from": 600, "to": 1000, "color": "rgba(0, 153, 51,1)" },
    { "from": 1000, "to": 1600, "color": "rgba(204, 153, 0,1)" },
    { "from": 1600, "to": 2000, "color": "rgba(204, 0, 0,1)" }
]'
data-color-plate="transparent"
data-color-major-ticks="#f5f5f5"
data-color-minor-ticks="#ddd"
data-color-title="#fff"
data-color-units="#ccc"
data-color-numbers="#eee"
data-color-needle-start="rgba(240, 128, 128, 1)"
data-color-needle-end="rgba(255, 160, 122, .9)"
data-value-box="true"
data-animation-rule="bounce"
data-animation-duration="500"
data-border-outer-width="3"
data-border-middle-width="3"

```

```

        data-border-inner-width="3"
<</canvas>
<canvas data-type="radial-gauge"
  data-width="222"
  data-height="222"
  data-units="&deg;C"
  data-title="Temperature"
  data-value=<?php echo "\".$TValue.\""\n"?>
  data-min-value="-50"
  data-max-value="50"
  data-major-ticks="[-50,-40,-30,-20,-10,0,10,20,30,40,50]"
  data-minor-ticks="2"
  data-stroke-ticks="true"
  data-highlights=' [
    {"from": -50, "to": 0, "color": "rgba(0,0, 255, .3)"},
    {"from": 0, "to": 20, "color": "rgba(0, 153, 51,1)"},
    {"from": 35, "to": 50, "color": "rgba(77, 0, 0,1)"},
    {"from": 20, "to": 35, "color": "rgba(255, 51, 51,1)"}
  ]'
  data-color-plate="transparent"
  data-ticks-angle="225"
  data-start-angle="67.5"
  data-color-major-ticks="#ddd"
  data-color-minor-ticks="#ddd"
  data-color-title="#eee"
  data-color-units="#ccc"
  data-color-numbers="#eee"
  data-color-plate="#222"
  data-border-shadow-width="0"
  data-borders="true"
  data-color-needle-start="rgba(240, 128, 128, 1)"
  data-color-needle-end="rgba(255, 160, 122, .9)"
  data-animation-duration="1500"
  data-animation-rule="linear"
  data-color-border-outer="#333"
  data-color-border-outer-end="#111"
  data-color-border-middle="#222"
  data-color-border-middle-end="#111"
  data-color-border-inner="#111"
  data-color-border-inner-end="#333"
  data-value-box-border-radius="0"
  data-color-value-box-rect="#222"
  data-color-value-box-rect-end="#333"
  data-font-value="Led"
  data-font-numbers="Led"
  data-font-title="Led"
  data-font-units="Led"
<</canvas>

<canvas data-type="radial-gauge"
  data-width="222"
  data-height="222"
  data-units="%"
  data-title="Humidity"
  data-value=<?php echo "\".$HValue.\""\n"?>
  data-animate-on-init="false"
  data-animated-value="false"
  data-min-value="0"
  data-max-value="100"
  data-major-ticks="0,10,20,30,40,50,60,70,80,90,100"
  data-minor-ticks="0.1"
  data-stroke-ticks="false"
  data-highlights=' [

```

```

        { "from": 0, "to": 40, "color": "rgba(204, 153, 0,1)" },
        { "from": 40, "to": 70, "color": "rgba(0, 153, 51,1)" },
        { "from": 70, "to": 100, "color": "rgba(204, 0, 0,1)" }
    ]'
    data-color-plate="transparent"
    data-color-major-ticks="#f5f5f5"
    data-color-minor-ticks="#ddd"
    data-color-title="#fff"
    data-color-units="#ccc"
    data-color-numbers="#eee"
    data-color-needle-start="rgba(240, 128, 128, 1)"
    data-color-needle-end="rgba(255, 160, 122, .9)"
    data-value-box="true"
    data-animation-rule="bounce"
    data-animation-duration="500"
    data-border-outer-width="3"
    data-border-middle-width="3"
    data-border-inner-width="3"
<</canvas>
<script>
if (!Array.prototype.forEach) {
    Array.prototype.forEach = function(cb) {
        var i = 0, s = this.length;
        for (; i < s; i++) {
            cb && cb(this[i], i, this);
        }
    }
}

document.fonts && document.fonts.forEach(function(font) {
    font.loaded.then(function() {
        if (font.family.match(/Led/)) {
            document.gauges.forEach(function(gauge) {
                gauge.update();
                gauge.options.renderTo.style.visibility = 'visible';
            });
        }
    });
});

var timers = [];

function animateGauges() {
    document.gauges.forEach(function(gauge) {
        timers.push(setInterval(function() {
            gauge.value = Math.random() *
                (gauge.options.maxValue - gauge.options.minValue) +
                gauge.options.minValue;
        }, gauge.animation.duration + 50));
    });
}

function stopGaugesAnimation() {
    timers.forEach(function(timer) {
        clearInterval(timer);
    });
}

function resize() {
    var size = parseFloat(document.getElementById('gauge-size').value) || 400;

    document.gauges.forEach(function(gauge) {
        gauge.update({ width: size, height: size });
    });
}

```



```

    });
}

function setText() {
    var text = document.getElementById('gauge-text').value;

    document.gauges.forEach(function (gauge) {
        gauge.update({ valueText: text });
    });
}
</script>

</body>
</html>

```

«statistics.php»

```

<html>
  <head>
    <meta http-equiv="refresh" content="30" /> <!--refresh selidas-->
    <script type="text/javascript"
src="https://www.gstatic.com/charts/loader.js"></script>
    <script type="text/javascript">
      google.charts.load('current', {'packages':['line']});
      google.charts.setOnLoadCallback(drawChart);

      function drawChart() {

        var data = new google.visualization.DataTable();
        data.addColumn('datetime', 'Datetime');
        data.addColumn('number', 'Speed');
        data.addColumn('number', 'Satellites');
        data.addColumn('number', 'Altitude');
        data.addColumn('number', 'Temperature');
        data.addColumn('number', 'Humidity');

<?php
    // Prepare variables for database connection

    $dbusername = "root"
    $dbpassword = "goopy3121"

    $server = "localhost";

    // Connect to your database
    $mysqli = new mysqli($server, $dbusername, $dbpassword, "ody");
    if($mysqli->connect_error)
        die('Connect Error (' . mysqli_connect_errno() . ') ' . mysqli_connect_error());

    $result = $mysqli->query("SELECT YEAR(Datetime) AS year, MONTH(Datetime) AS month,
DAY(Datetime) AS day, HOUR(Datetime) AS hour, MINUTE(Datetime) AS minute,
SECOND(Datetime) AS sec, Speed,Satellites,Altitude,Temperature,Humidity FROM
ody.markers ORDER BY Datetime DESC LIMIT 5");

    if ($result->num_rows > 0) {

        // output data of each row
        while($row = $result->fetch_assoc()) {
            echo "data.addRow([new Date(".$row["year"].".".$row["month"]."."-
1, ".$row["day"].".".$row["hour"].".".$row["minute"].".".$row["sec"]."),

```

```

".$row["Speed"].",".$row["Satellites"].",".$row["Altitude"].",".$row["Temperature"].",
".$row["Humidity"].");\n";
}
}
else {
}
?>
var Speed_options = {
  chart: {
    title: 'Speed vs Time',
    subtitle: 'measured in km/h'
  },
  hAxis: { format: 'HH:mm:ss a' },
  width: 600,
  height: 300
};

var Satellites_options = {
  chart: {
    title: 'Satellites vs Time',
  },
  hAxis: { format: 'HH:mm:ss a' },
  width: 600,
  height: 300
};

var Altitude_options = {
  chart: {
    title: 'Altitude vs Time',
    subtitle: 'measured in meters'
  },
  hAxis: { format: 'HH:mm:ss a' },
  width: 600,
  height: 300
};

var Temperature_options = {
  chart: {
    title: 'Temperature vs Time',
    subtitle: 'measured in C'
  },
  hAxis: { format: 'HH:mm:ss a' },
  width: 600,
  height: 300
};

var Humidity_options = {
  chart: {
    title: 'Humidity vs Time',
    subtitle: 'measured in %'
  },
  hAxis: { format: 'HH:mm:ss a' },
  width: 600,
  height: 300
};

var Speed_chart = new google.charts.Line(document.getElementById('Speed_div'));

var Speed_view = new google.visualization.DataView(data);
Speed_view.setColumns([0,1]);

```

```

        Speed_chart.draw(Speed_view, google.charts.Line.convertOptions(Speed_options));
//-----
        var Satellites_chart = new
google.charts.Line(document.getElementById('Satellites_div'));

        var Satellites_view = new google.visualization.DataView(data);
        Satellites_view.setColumns([0,2]);
        Satellites_chart.draw(Satellites_view,
google.charts.Line.convertOptions(Satellites_options));
//-----
        var Altitude_chart = new
google.charts.Line(document.getElementById('Altitude_div'));

        var Altitude_view = new google.visualization.DataView(data);
        Altitude_view.setColumns([0,3]);
        Altitude_chart.draw(Altitude_view,
google.charts.Line.convertOptions(Altitude_options));
//-----
        var Temperature_chart = new
google.charts.Line(document.getElementById('Temperature_div'));

        var Temperature_view = new google.visualization.DataView(data);
        Temperature_view.setColumns([0,4]);
        Temperature_chart.draw(Temperature_view,
google.charts.Line.convertOptions(Temperature_options));
//-----
        var Humidity_chart = new
google.charts.Line(document.getElementById('Humidity_div'));

        var Humidity_view = new google.visualization.DataView(data);
        Humidity_view.setColumns([0,5]);
        Humidity_chart.draw(Humidity_view,
google.charts.Line.convertOptions(Humidity_options));

//-----
    }
</script>
</head>
<title>Statistics </title>
<body style="background: #001433">
    <div style="width:100%">
        <div style="width:90%;font-size:30pt;text-align:center;color:#ffffff;text-
align:left;"><strong>Gps Real-Time Tracking</div>
        </div>
        
        
        <div style="width:100%;font-size:19pt;color:#cca300;text-align:left;">Real-
Time Διαγράμματα
        </div>
        <br>
        <hr>
        <div align="right">

</div>

<table style="border-spacing: 15px">
    <tr>
        <td style="border: 1px solid white">
            <div id="Speed_div" style="width: 600px; height: 300px"></div>
        </td>
    </tr>
</table>

```

```

                <td style="border: 1px solid white">
                    <div id="Satellites_div" style="width: 600px; height:
300px"></div>
                </td>
            </tr>
            <tr>
                <td style="border: 1px solid white">
                    <div id="Altitude_div" style="width: 600px; height: 300px"></div>
                </td>
                <td style="border: 1px solid white">
                    <div id="Temperature_div" style="width: 600px; height:
300px"></div>
                </td>
            </tr>
            <tr>
                <td style="border: 1px solid white">
                    <div id="Humidity_div" style="width: 600px; height: 300px"></div>
                </td>
            </tr>
        </table>
    </body>
</html>

```

«weekly_statistics.php»

```

<html>
  <head>
    <meta http-equiv="refresh" content="30" />
    <script type="text/javascript"
src="https://www.gstatic.com/charts/loader.js"></script>
    <script type="text/javascript">
        google.charts.load('current', {'packages':['corechart']});
        google.charts.setOnLoadCallback(drawChart);

        function drawChart() {

            var data = new google.visualization.DataTable();
            data.addColumn('datetime', 'Datetime');
            data.addColumn('number', 'Average Speed');
            data.addColumn('number', 'Max Speed');
            data.addColumn('number', 'Min Speed');
            data.addColumn('number', 'Average Satellites');
            data.addColumn('number', 'Max Satellites');
            data.addColumn('number', 'Min Satellites');
            data.addColumn('number', 'Average Altitude');
            data.addColumn('number', 'Max Altitude');
            data.addColumn('number', 'Min Altitude');
            data.addColumn('number', 'Average Temperature');
            data.addColumn('number', 'Max Temperature');
            data.addColumn('number', 'Min Temperature');
            data.addColumn('number', 'Average Humidity');
            data.addColumn('number', 'Max Humidity');
            data.addColumn('number', 'Min Humidity');

```

```

<?php
// Prepare variables for database connection

$dbusername = "root";
$dbpassword = "goopy3121!!";
$server = "localhost";

// Connect to your database
$mysqli = new mysqli($server, $dbusername, $dbpassword, "ody");
if($mysqli->connect_error)
    die('Connect Error (' . mysqli_connect_errno() . ') ' . mysqli_connect_error());

$result = $mysqli->query("SELECT YEAR(Datetime) AS year, MONTH(Datetime) AS month,
DAY(Datetime) AS day, AVG(Speed),MAX(Speed),MIN(Speed),
AVG(Satellites),MAX(Satellites),MIN(Satellites),
AVG(Altitude),MAX(Altitude),MIN(Altitude),
AVG(Temperature),MAX(Temperature),MIN(Temperature),
AVG(Humidity),MAX(Humidity),MIN(Humidity) from markers group by DAY(Datetime) ORDER BY
Datetime DESC limit 7");

if ($result->num_rows > 0) {

// output data of each row
while($row = $result->fetch_assoc()) {
    echo "data.addRow([new Date(".$row["year"].",".$row["month"]."-
1, ".$row["day"]."),
".$row["AVG(Speed)"].",".$row["MAX(Speed)"].",".$row["MIN(Speed)"].",".$row["AVG(Satel
lites)"].",".$row["MAX(Satellites)"].",".$row["MIN(Satellites)"].",".$row["AVG(Altitud
e)"].",".$row["MAX(Altitude)"].",".$row["MIN(Altitude)"].",".$row["AVG(Temperature)"].
",".$row["MAX(Temperature)"].",".$row["MIN(Temperature)"].",".$row["AVG(Humidity)"].",
".$row["MAX(Humidity)"].",".$row["MIN(Humidity)"]);\n";
}
}
else {
}
?>

var Speed_options = {
    title: 'Speed Last Week',
    subtitle: 'measured in km/h',
    hAxis: { format: 'MMM dd, yyyy' },
    seriesType: 'bars',
    width: 600,
    height: 300
};

var Satellites_options = {
    title: 'Satellites Last Week',
    hAxis: { format: 'MMM dd, yyyy' },
    seriesType: 'bars',
    width: 600,
    height: 300
};

var Altitude_options = {
    title: 'Altitude Last Week',
    subtitle: 'measured in meters',
    hAxis: { format: 'MMM dd, yyyy' },
    seriesType: 'bars',
    width: 600,
    height: 300
};

```

```

    var Temperature_options = {
      title: 'Temperature (C) Last Week',
      subtitle: 'measured in C',
      hAxis: { format: 'MMM dd, yyyy' },
      seriesType: 'bars',
      width: 600,
      height: 300
    };

    var Humidity_options = {
      title: 'Humidity (%) Last Week',
      subtitle: 'measured in %',
      hAxis: { format: 'MMM dd, yyyy' },
      seriesType: 'bars',
      width: 600,
      height: 300
    };

    var Speed_chart = new
google.visualization.ComboChart(document.getElementById('Speed_div'));

    var Speed_view = new google.visualization.DataView(data);
    Speed_view.setColumns([0,1,2,3]);
    Speed_chart.draw(Speed_view, Speed_options);

//-----
    var Satellites_chart = new
google.visualization.ComboChart(document.getElementById('Satellites_div'));

    var Satellites_view = new google.visualization.DataView(data);
    Satellites_view.setColumns([0,4,5,6]);
    Satellites_chart.draw(Satellites_view, Satellites_options);

//-----
    var Altitude_chart = new
google.visualization.ComboChart(document.getElementById('Altitude_div'));

    var Altitude_view = new google.visualization.DataView(data);
    Altitude_view.setColumns([0,7,8,9]);
    Altitude_chart.draw(Altitude_view, Altitude_options);

//-----
    var Temperature_chart = new
google.visualization.ComboChart(document.getElementById('Temperature_div'));

    var Temperature_view = new google.visualization.DataView(data);
    Temperature_view.setColumns([0,10,11,12]);
    Temperature_chart.draw(Temperature_view, Temperature_options);

//-----
    var Humidity_chart = new
google.visualization.ComboChart(document.getElementById('Humidity_div'));

    var Humidity_view = new google.visualization.DataView(data);
    Humidity_view.setColumns([0,13,14,15]);
    Humidity_chart.draw(Humidity_view, Humidity_options);

}
</script>
</head>
<title>Weekly </title>

```

```

<body style="background: #001433">
  <div style="width:100%">

    <div style="width:90%;font-size:30pt;text-align:center;color:#ffffff;text-
align:left;"><strong>Gps Real-Time Tracking</div>
    </div>
    

    <div style="width:100%;font-size:19pt;color:#cca300;text-
align:left;">Εβδομαδιαίες Μετρήσεις
    </div>
    <br>
    <hr>
    <div align="right">

</div>

    <table style="border-spacing: 15px">
      <tr>
        <td style="border: 1px solid white">
          <div id="Speed_div" style="width: 600px; height: 300px"></div>
        </td>
        <td style="border: 1px solid white">
          <div id="Satellites_div" style="width: 600px; height:
300px"></div>
        </td>
      </tr>
      <tr>
        <td style="border: 1px solid white">
          <div id="Altitude_div" style="width: 600px; height: 300px"></div>
        </td>
        <td style="border: 1px solid white">
          <div id="Temperature_div" style="width: 600px; height:
300px"></div>
        </td>
      </tr>
      <tr>
        <td style="border: 1px solid white">
          <div id="Humidity_div" style="width: 600px; height: 300px"></div>
        </td>
      </tr>
    </table>
  </body>
</html>

```

«monthly_statistics.php»

```

<html>
  <head>
    <meta http-equiv="refresh" content="30" />
    <script type="text/javascript"
src="https://www.gstatic.com/charts/loader.js"></script>
    <script type="text/javascript">

      google.charts.load('current', {'packages':['corechart']});
      google.charts.setOnLoadCallback(drawChart);

      function drawChart() {

```

```

var data = new google.visualization.DataTable();
data.addColumn('datetime', 'Datetime');
data.addColumn('number', 'Average Speed');
data.addColumn('number', 'Max Speed');
data.addColumn('number', 'Min Speed');
data.addColumn('number', 'Average Satellites');
data.addColumn('number', 'Max Satellites');
data.addColumn('number', 'Min Satellites');
data.addColumn('number', 'Average Altitude');
data.addColumn('number', 'Max Altitude');
data.addColumn('number', 'Min Altitude');
data.addColumn('number', 'Average Temperature');
data.addColumn('number', 'Max Temperature');
data.addColumn('number', 'Min Temperature');
data.addColumn('number', 'Average Humidity');
data.addColumn('number', 'Max Humidity');
data.addColumn('number', 'Min Humidity');

<?php
// Prepare variables for database connection

$dbusername = "root";
$dbpassword = "goopy3121!!";
$server = "localhost";

// Connect to your database
$mysqli = new mysqli($server, $dbusername, $dbpassword, "ody");
if($mysqli->connect_error)
    die('Connect Error (' . mysqli_connect_errno() . ') ' . mysqli_connect_error());

$result = $mysqli->query("SELECT YEAR(Datetime) AS year, MONTH(Datetime) AS month,
DAY(Datetime) AS day, AVG(Speed),MAX(Speed),MIN(Speed),
AVG(Satellites),MAX(Satellites),MIN(Satellites),
AVG(Altitude),MAX(Altitude),MIN(Altitude),
AVG(Temperature),MAX(Temperature),MIN(Temperature),
AVG(Humidity),MAX(Humidity),MIN(Humidity) from markers group by MONTH(Datetime) ORDER
BY Datetime DESC limit 12");

if ($result->num_rows > 0) {

// output data of each row
while($row = $result->fetch_assoc()) {
    echo "data.addRow([new Date(".$row["year"].".".$row["month"]."."-
1, ".$row["day"]."),
".$row["AVG(Speed)"].".".$row["MAX(Speed)"].".".$row["MIN(Speed)"].".".$row["AVG(Satel
lites)"].".".$row["MAX(Satellites)"].".".$row["MIN(Satellites)"].".".$row["AVG(Altitud
e)"].".".$row["MAX(Altitude)"].".".$row["MIN(Altitude)"].".".$row["AVG(Temperature)"].
".".$row["MAX(Temperature)"].".".$row["MIN(Temperature)"].".".$row["AVG(Humidity)"]."."
".$row["MAX(Humidity)"].".".$row["MIN(Humidity)"]." ]);\n";
}
}
else {
}
?>

var Speed_options = {
    title: 'Speed (km/h) Last Month',
    subtitle: 'measured in km/h',
    hAxis: { format: ' '},
    seriesType: 'bars',
    width: 600,
    height: 300
}

```



```

};

var Satellites_options = {
  title: 'Satellites Last Month',
  hAxis: { format: ' ' },
  seriesType: 'bars',
  width: 600,
  height: 300
};

var Altitude_options = {
  title: 'Altitude (Meters) Last Month',
  subtitle: 'measured in meters',
  hAxis: { format: ' ' },
  seriesType: 'bars',
  width: 600,
  height: 300
};

var Temperature_options = {
  title: 'Temperature (C) Last Month ',
  subtitle: 'measured in C',
  hAxis: { format: ' ' },
  seriesType: 'bars',
  width: 600,
  height: 300
};

var Humidity_options = {
  title: 'Humidity (%) Last Month ',
  subtitle: 'measured in %',
  hAxis: { format: ' ' },
  seriesType: 'bars',
  width: 600,
  height: 300
};

var Speed_chart = new
google.visualization.ComboChart(document.getElementById('Speed_div'));

var Speed_view = new google.visualization.DataView(data);
Speed_view.setColumns([0,1,2,3]);
Speed_chart.draw(Speed_view, Speed_options);

//-----
var Satellites_chart = new
google.visualization.ComboChart(document.getElementById('Satellites_div'));

var Satellites_view = new google.visualization.DataView(data);
Satellites_view.setColumns([0,4,5,6]);
Satellites_chart.draw(Satellites_view, Satellites_options);

//-----
var Altitude_chart = new
google.visualization.ComboChart(document.getElementById('Altitude_div'));

var Altitude_view = new google.visualization.DataView(data);
Altitude_view.setColumns([0,7,8,9]);
Altitude_chart.draw(Altitude_view, Altitude_options);

//-----

```

```

    var Temperature_chart = new
google.visualization.ComboChart(document.getElementById('Temperature_div'));

    var Temperature_view = new google.visualization.DataView(data);
    Temperature_view.setColumns([0,10,11,12]);
    Temperature_chart.draw(Temperature_view, Temperature_options);
//-----
    var Humidity_chart = new
google.visualization.ComboChart(document.getElementById('Humidity_div'));

    var Humidity_view = new google.visualization.DataView(data);
    Humidity_view.setColumns([0,13,14,15]);
    Humidity_chart.draw(Humidity_view, Humidity_options);
//-----
}
</script>
</head>
<title>Monthly </title>
<body style="background: #001433">
    <div style="width:100%">
        <div style="width:90%;font-size:30pt;text-align:center;color:#ffffff;text-
align:left;"><strong>Gps Real-Time Tracking</div>
        </div>
        

        <div style="width:100%;font-size:19pt;color:#cca300;text-align:left;">Μηνιαίες
Μετρήσεις
        </div>
        <br>
        <hr>
        <div align="right">

        </div>

        <table style="border-spacing: 15px">
            <tr>
                <td style="border: 1px solid white">
                    <div id="Speed_div" style="width: 600px; height: 300px"></div>
                </td>
                <td style="border: 1px solid white">
                    <div id="Satellites_div" style="width: 600px; height:
300px"></div>
                </td>
            </tr>
            <tr>
                <td style="border: 1px solid white">
                    <div id="Altitude_div" style="width: 600px; height: 300px"></div>
                </td>
                <td style="border: 1px solid white">
                    <div id="Temperature_div" style="width: 600px; height:
300px"></div>
                </td>
            </tr>
            <tr>
                <td style="border: 1px solid white">
                    <div id="Humidity_div" style="width: 600px; height: 300px"></div>
                </td>
            </tr>
        </table>

```



```

<td><a href="weekly_statistics.php">
</a></td>

<td><a href="monthly_statistics.php">
</a></td>

    </table>

</div>
    <br>
    <br>
    <br>
    <br>
    <br>
    <br>

    <div style="cursor:pointer; width:95.5%;font-size:23pt;background-
color:#cca300;color:#000000;align:middle; margin:0 fixed;"
    <button class="button"
onclick="window.open('../index.html','_blank')">Map&nbsp;&nbsp;&nbsp;</button>
    </div>
<div id="footer">
    <div id="footerContainer">
        
        
        
        
        
        
        
        

    </div>
</div>
</body>
</html>

```

«button.css»

```

.button{
    background-color: #cca300;
    padding: 15px 32px;
    text-align: center;
    text-decoration: none;
    display: inline-block;
    font-size: 16px;
    cursor:pointer;
}

```

«footer.css»

```
#footer {  
  position:fixed;  
  bottom:0;  
  left:0;  
  right:0;  
  background-color: white;  
  width:100%;  
  height:80px;  
}
```

```
#footerContainer {  
  position:fixed;  
  width:100%;  
  height:70px;  
}
```