



Σχολή Τεχνολογικών Εφαρμογών

Τμήμα Αυτοματισμού

Πτυχιακή εργασία

Θέμα:

**«ΣΥΣΚΕΥΗ ΕΝΤΟΠΙΣΜΟΥ ΘΕΣΗΣ ΚΑΙ ΑΠΟΣΤΟΛΗΣ ΔΕΔΟΜΕΝΩΝ
ΜΕΣΩ ΔΙΚΤΥΟΥ ΚΙΝΗΤΗΣ ΤΗΛΕΦΩΝΙΑΣ
ΒΑΣΙΣΜΕΝΗ ΣΤΗΝ ΠΛΑΤΦΟΡΜΑ ARDUINO»**



ΕΠΙΒΛΕΠΩΝ ΚΑΘΗΓΗΤΗΣ: **ΜΙΧΑΗΛ ΠΑΠΟΥΤΣΙΔΑΚΗΣ**

ΦΟΙΤΗΤΕΣ: **ΣΤΑΜΑΤΙΟΣ ΜΠΑΜΠΑΤΖΙΑΣ**

A.M.33803

ΚΩΝΣΤΑΝΤΙΝΟΣ ΠΟΛΙΤΟΠΟΥΛΟΣ

A.M. 33652

ΠΕΡΙΛΗΨΗ

Η παρούσα εργασία με θέμα «Συσκευή εντοπισμού θέσης και αποστολής δεδομένων μέσω δικτύου κινητής τηλεφωνίας βασισμένη στην πλατφόρμα Arduino» αφορά στην συναρμολόγηση και τον προγραμματισμό μιας συσκευής, η οποία αποτελείται από την πλατφόρμα μικροελεγκτή Arduino Leonardo και την πρόσθετη σε αυτήν πλακέτα με ενσωματωμένο δέκτη GPS και πομποδέκτη 3G/GPRS, καθώς επίσης και από το κύκλωμα τροφοδοσίας της. Η κύρια λειτουργία της κατασκευής αυτής είναι ο εντοπισμός της γεωγραφικής θέσης (γεωγραφικό μήκος και πλάτος) της συσκευής με χρήση δέκτη GPS. Αφού ενεργοποιηθεί η συσκευή, ελέγχει περιοδικά τυχόν μεταβολή της θέσης της με χρήση κατάλληλου αισθητηρίου. Σε περίπτωση που εντοπίσει αλλαγή τότε εντοπίζει το γεωγραφικό στίγμα από το GPS και το αποστέλλει σε προκαθορισμένο αριθμό κινητού τηλεφώνου και σε προκαθορισμένη διεύθυνση ηλεκτρονικού ταχυδρομείου κάνοντας χρήση του πομποδέκτη 3G/GPRS.

Η συσκευή αυτή μπορεί να χρησιμοποιηθεί σε οποιαδήποτε εφαρμογή απαιτεί έλεγχο θέσης αντικειμένου και άμεση ενημέρωση του χρήστη σε περίπτωση αλλαγής αυτής. Παραδείγματος χάριν το σύστημα είναι δυνατόν να τοποθετηθεί σε στόλο οχημάτων (αυτοκίνητο, μοτοσυκλέτα, φορτηγό, σκάφος) για την χρήση του ως μέτρο ασφάλειας. Η παραπάνω συσκευή έχει τη δυνατότητα να βελτιωθεί και να εξελιχθεί περαιτέρω έτσι ώστε να ενημερώνει τον χρήστη μέσω διαδικτύου χρησιμοποιώντας ιστοσελίδα και εφαρμογή σύγχρονου κινητού τηλεφώνου.

ABSTRACT

The present thesis with the title “Position tracking device and data transfer through mobile network based on Arduino platform” is concerned with the assembly and programming of a device that consists of the microcontroller platform Arduino Leonardo and an additional board with a built-in GPS receiver and a built-in 3G/GPRS transceiver as well as the power circuit. The main operation of the device is to check its geographical position (latitude and longitude) using the GPS receiver. Once the device is turned on, it periodically checks for changes in its geographical position by means of the proper sensor. In case there is a change, it traces the geographical position through the GPS and sends it to a default cell phone number and a default e-mail address using the 3G/GPRS transceiver.

This device can be used in any project that needs to check the geographical position of an object and inform the user immediately in case of change in its position. For example, it can be installed on a fleet of vehicle (car, motorcycle, truck, vessel) and used as a security measure. The device mentioned above can be further improved and developed to inform its user via internet using a webpage and via an application of a contemporary mobile phone.

ΕΥΧΑΡΙΣΤΙΕΣ

Θα θέλαμε να ευχαριστήσουμε ιδιαίτερος τον επιβλέποντα καθηγητή της πτυχιακής μας εργασίας κ. Παπουτσιδάκη Μιχαήλ για την καθοδήγηση και τις πολύτιμες συμβουλές του κατά την εκπόνηση της παρούσας εργασίας μας. Επίσης ευχαριστούμε τις οικογένειες μας για την στήριξη τους και την υπομονετικότητα τους καθ' όλη την διάρκεια της κατασκευής της συσκευής και της συγγραφής της εργασίας.

ΠΕΡΙΕΧΟΜΕΝΑ

ΕΙΣΑΓΩΓΗ	6
1 ΕΙΣΑΓΩΓΗ ΣΤΟΥΣ ΜΙΚΡΟΕΛΕΓΚΤΕΣ.....	8
1.1 ΨΗΦΙΑΚΑ ΟΛΟΚΛΗΡΩΜΕΝΑ ΚΥΚΛΩΜΑΤΑ	9
1.2 ΤΙ ΕΙΝΑΙ Ο ΜΙΚΡΟΕΛΕΓΚΤΗΣ.....	13
1.3 ΚΑΤΗΓΟΡΙΕΣ ΜΙΚΡΟΕΛΕΓΚΤΩΝ ΚΑΙ ΚΑΤΑΣΚΕΥΑΣΤΕΣ	17
1.4 ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΣ ΜΙΚΡΟΕΛΕΓΚΤΩΝ ΚΑΙ ΕΡΓΑΛΕΙΑ ΑΝΑΠΤΥΞΗΣ	20
2 ΠΑΡΟΥΣΙΑΣΗ ΣΤΙΣ ΤΕΧΝΟΛΟΓΙΕΣ GPS-GSM/3G	22
2.1 ΑΝΑΛΥΣΗ ΣΥΣΤΗΜΑΤΩΝ GPS.....	23
2.2 ΤΕΧΝΟΛΟΓΙΕΣ GSM ΚΑΙ 3G.....	33
3 ΠΑΡΟΥΣΙΑΣΗ ΤΗΣ ΠΛΑΤΦΟΡΜΑΣ ARDUINO.....	43
3.1 ΣΥΝΤΟΜΗ ΠΕΡΙΓΡΑΦΗ ΤΟΥ ARDUINO	44
3.2 ΔΙΑΦΟΡΕΣ ΥΛΟΠΟΙΗΣΕΙΣ ΤΟΥ ARDUINO.....	46
3.3 ΤΟ ΥΛΙΚΟ (HARDWARE) ΤΟΥ ARDUINO.....	49
3.4 Η ΤΡΟΦΟΔΟΣΙΑ ΤΗΣ ΠΛΑΚΕΤΑΣ ΤΟΥ ARDUINO.....	50
3.5 Η ΜΝΗΜΗ ΤΟΥ LEONARDO	51
3.6 ΟΙ ΑΚΡΟΔΕΚΤΕΣ ΤΟΥ ARDUINO LEONARDO.....	51
3.7 ARDUINO SHIELDS.....	54
3.8 ΤΟ ΠΕΡΙΒΑΛΛΟΝ ΑΝΑΠΤΥΞΗΣ ΛΟΓΙΣΜΙΚΟΥ ΤΟΥ ARDUINO.....	56
4 ΠΑΡΟΥΣΙΑΣΗ ΤΗΣ ΠΡΟΣΘΕΤΗΣ ΠΛΑΚΕΤΑΣ GPS-3G/GSM.....	59
4.1 ΒΑΣΙΚΑ ΧΑΡΑΚΤΗΡΙΣΤΙΚΑ ΚΑΙ ΛΕΙΤΟΥΡΓΙΕΣ	60
4.2 ΚΥΡΙΕΣ ΛΕΙΤΟΥΡΓΙΕΣ ΕΠΙΚΟΙΝΩΝΙΑΣ	61
4.3 ΛΕΙΤΟΥΡΓΙΑ ΕΝΤΟΠΙΣΜΟΥ ΓΕΩΓΡΑΦΙΚΗΣ ΘΕΣΗΣ (GPS).....	63
4.4 ΕΠΙΠΡΟΣΘΕΤΕΣ ΔΥΝΑΤΟΤΗΤΕΣ ΤΗΣ ΠΡΟΣΘΕΤΗΣ ΠΛΑΚΕΤΑΣ.....	65
5 ΥΛΟΠΟΙΗΣΗ ΤΟΥ ΥΛΙΚΟΥ ΚΑΙ ΛΟΓΙΣΜΙΚΟΥ ΤΗΣ ΣΥΣΚΕΥΗΣ.....	68
5.1 ΣΤΟΙΧΕΙΑ ΤΗΣ ΟΛΟΚΛΗΡΩΜΕΝΗΣ ΚΑΤΑΣΚΕΥΗΣ	69
5.2 ΤΡΟΦΟΔΟΣΙΑ ΤΗΣ ΚΑΤΑΣΚΕΥΗΣ.....	76
5.3 ΕΠΙΚΟΙΝΩΝΙΑ ΤΗΣ ΣΥΣΚΕΥΗΣ ΜΕ ΤΟΝ ΧΡΗΣΤΗ.....	78
5.4 ΑΝΑΛΥΤΙΚΗ ΠΑΡΟΥΣΙΑΣΗ ΤΗΣ ΚΑΤΑΣΚΕΥΗΣ.....	79
5.5 ΕΠΕΞΗΓΗΣΗ ΠΗΓΑΙΟΥ ΚΩΔΙΚΑ ΚΑΙ ΛΕΙΤΟΥΡΓΙΑΣ.....	84
6 ΣΥΜΠΕΡΑΣΜΑΤΑ.....	99



6.1	ΑΠΟΤΕΛΕΣΜΑΤΑ ΛΕΙΤΟΥΡΓΙΑΣ ΚΑΤΑΣΚΕΥΗΣ.....	100
6.2	ΠΡΟΤΕΙΝΟΜΕΝΕΣ ΒΕΛΤΙΩΣΕΙΣ.....	101
	ΒΙΒΛΙΟΓΡΑΦΙΑ.....	103
	ΠΑΡΑΡΤΗΜΑ.....	105

ΕΙΣΑΓΩΓΗ

Κατά τη διάρκεια της δεύτερης δεκαετίας του 2000 που διανύουμε, παρατηρείται όλο και πιο συχνά το φαινόμενο της έξαρσης των κλοπών. Καθημερινώς ακούμε για κλοπές αυτοκινήτων, μηχανών, ποδηλάτων και κινητών. Ειδικά κλεμμένα ποδήλατα και κινητά μπορεί κάποιος να αγοράσει από μαύρες αγορές που ανθούν στις μέρες μας. Τι θα γινόταν όμως αν κάποιος μπορούσε να ειδοποιηθεί την ίδια την στιγμή της κλοπής από αυτό που πρόκειται να κλαπεί; Αν υπήρχε κάπου τοποθετημένη μια αντικλεπτική συσκευή σε μη φανερό σημείο επάνω σε αξίας αντικείμενα όπως το αυτοκίνητο ή το ποδήλατο; Αν αυτή η συσκευή όντας τοποθετημένη πάνω σε μελλοντικό αντικείμενο κλοπής μπορούσε να ειδοποιήσει το χρήστη της για το που βρίσκεται;

Η κατασκευή μιας αντικλεπτικής συσκευής, που θα τοποθετείται σε αντικείμενα που κάποιος θέλει να προστατέψει, με στόχο την ειδοποίηση του χρήστη την ώρα της κλοπής και την επανάκτησή τους, κρίνεται απαραίτητη στις μέρες μας και γι' αυτό το λόγο αποφασίσαμε να ασχοληθούμε με την συγκεκριμένη κατασκευή.

Αρχικά πρόκειται να μελετηθούν η επιθυμητή λειτουργία της συσκευής και ποια είναι τα καταλληλότερα υλικά για την υλοποίηση της. Η εργασία θα βασιστεί κυρίως στις σύγχρονες τεχνολογίες πρόσβασης στο διαδίκτυο μέσω δικτύου κινητών τηλεπικοινωνιών (3G/GPRS), στην δημοφιλή τεχνολογία του παγκόσμιου συστήματος εντοπισμού γεωγραφικής θέσης (GPS) και στην αναπτυξιακή πλατφόρμα Arduino με χρήση μικροελεγκτή.

Θα ασχοληθούμε με την θεωρητική μελέτη των ψηφιακών ολοκληρωμένων κυκλωμάτων και την έρευνα του τρόπου λειτουργίας των τεχνολογιών που προαναφέραμε έτσι ώστε να τις αξιοποιήσουμε αποτελεσματικά στην κατασκευή μας. Θα γίνει αναφορά στα κυρίως χαρακτηριστικά των σύγχρονων ψηφιακών συστημάτων και στους μικροελεγκτές. Στη συνέχεια θα περιγραφεί η λειτουργία του συστήματος GPS και θα αναλυθεί η τεχνολογία GSM και 3G/GPRS. Στη συνέχεια θα παρουσιάσουμε αναλυτικά τα τεχνικά χαρακτηριστικά και τις δυνατότητες της πλατφόρμας Arduino Leonardo και της πρόσθετης πλακέτας 3G/GPRS + GPS που χρησιμοποιούμε στην κατασκευή. Τέλος θα περιγραφεί το υλικό μέρος της συσκευής και θα επεξηγηθούν ο κώδικας καθώς επίσης και τα αποτελέσματα της λειτουργίας.

Με την ολοκλήρωση της παρούσας εργασίας αναμένεται να κατασκευαστεί μια συσκευή η οποία θα έχει αντικλεπτική λειτουργία. Αφού ενεργοποιηθεί από το χρήστη η συσκευή θα έχει την δυνατότητα να ανιχνεύει τυχόν αλλαγή της αρχικής της θέσης, να εντοπίζει το στίγμα της και να αποστέλλει τις πληροφορίες αυτές στον χρήστη.

1 ΕΙΣΑΓΩΓΗ ΣΤΟΥΣ ΜΙΚΡΟΕΛΕΓΚΤΕΣ

1.1 ΨΗΦΙΑΚΑ ΟΛΟΚΛΗΡΩΜΕΝΑ ΚΥΚΛΩΜΑΤΑ

Τα ψηφιακά κυκλώματα κατασκευάζονται με ολοκληρωμένα κυκλώματα. Ένα ολοκληρωμένο κύκλωμα (συντομογραφία IC) είναι ένα εξάρτημα που αποτελείται από ένα μικρό υπόστρωμα πυριτίου, που ονομάζεται «chip», όπου πάνω του έχει σχηματιστεί το επιθυμητό κύκλωμα με ηλεκτρονικά στοιχεία (τρανζίστορ, δίοδοι, αντιστάσεις) από ημιαγωγό. Το chip τοποθετείται σε ένα κεραμικό ή πλαστικό περίβλημα, ενώνονται και συγκολλούνται οι επαφές του chip με εξωτερικούς ακροδέκτες «pins» για να σχηματιστεί το ολοκληρωμένο κύκλωμα. Ο αριθμός των pins μπορεί να διαφέρει από 14 σ' ένα μικρό IC περίβλημα σε 64 ή περισσότερους σε ένα μεγαλύτερο περίβλημα. Κάθε IC έχει μια αριθμητική ένδειξη τυπωμένη στην επιφάνεια του περιβλήματος, για να αναγνωρίζεται.

Τα ψηφιακά ολοκληρωμένα, συχνά, χωρίζονται σε κατηγορίες ανάλογα με την κυκλωματική τους πολυπλοκότητα, όπως αυτή μετريέται με τον αριθμό των λογικών πυλών σε ένα περίβλημα. Έτσι έχουμε:

- Τα Μικρής κλίμακας ολοκλήρωσης (Small Scale Integration SSI) κυκλώματα που περιέχουν έως 10 πύλες.
- Τα Μεσαίας κλίμακας ολοκλήρωσης (Medium – Scale Integration MSI) κυκλώματα που έχουν μια πολυπλοκότητα περίπου 10 με 100 πυλών.
- Τα Μεγάλης κλίμακας ολοκλήρωσης (Large – Scale Integration LSI) κυκλώματα που έχουν μια πολυπλοκότητα περίπου 100 με μερικές χιλιάδες πύλες σε ένα περίβλημα.
- Τα Πολύ μεγάλης κλίμακας ολοκλήρωσης (Very Large – Scale Integration VLSI) κυκλώματα που έχουν χιλιάδες πύλες σε ένα περίβλημα.

Οι ψηφιακές λογικές πύλες (AND, OR κ.τ.λ.) ταξινομούνται με την «οικογένεια» λογικού κυκλώματος στην οποία ανήκουν. Κάθε οικογένεια έχει το δικό της βασικό δομικό ηλεκτρονικό κύκλωμα από το οποίο αναπτύσσονται οι διάφορες πύλες. Ανάλογα την τεχνολογία (τρόπο) κατασκευής του βασικού κυκλώματος, έχουμε και την αντίστοιχη οικογένεια. Οι πιο πλατιά διαδεδομένες οικογένειες είναι: TTL και CMOS.

Η οικογένεια: TTL (Transistor – Transistor Logic). Η TTL είναι μια πολύ διαδεδομένη λογική οικογένεια, που υπάρχει εδώ και καιρό. Τα ολοκληρωμένα κυκλώματα TTL

κατασκευάζονται πάνω σε ένα chip ημιαγωγού με δομικό στοιχείο το διπολικό τρανζίστορ σε συνδυασμό με αντιστάσεις από ημιαγωγό.

Τα chip TTL έχουν συνήθως μια αριθμητική ονομασία των «σειρών» 5400 και 7400. Η πρώτη σειρά μπορεί να λειτουργήσει σε πλατιά ζώνη θερμοκρασιών, κατάλληλη για στρατιωτικές χρήσεις, ενώ η δεύτερη χρησιμοποιείται σε βιομηχανικές εφαρμογές. Τα chips της σειράς 7400 αριθμούνται 7400, 7401, 7402 κ.τ.λ. Έτσι για παράδειγμα, το 7400 περιέχει τέσσερις πύλες NAND 2-εισόδων, το 7404 περιέχει έξι αντιστροφείς κ.τ.λ. Τα ολοκληρωμένα κυκλώματα TTL χρειάζονται τάση τροφοδοσίας 5V για σωστή λειτουργία.

Στα ολοκληρωμένα κυκλώματα τύπου TTL, για ορθή λειτουργία, μια είσοδος χαμηλής στάθμης (λογικό «0») πρέπει να κυμαίνεται μεταξύ γείωσης και 0,8V, ενώ μια είσοδος υψηλής στάθμης (λογικό «1») πρέπει να κυμαίνεται μεταξύ 2,0V έως 5,5V. Μια έξοδος χαμηλής στάθμης μπορεί να είναι έως 0,4V ενώ μια έξοδος υψηλής στάθμης μπορεί να είναι από 2,4V έως 5,5V, η οποία εξαρτάται από το ρεύμα στην αντίσταση φορτίου της εξόδου.

Τα ολοκληρωμένα τύπου TTL χωρίζονται σε υποοικογένειες. Οι πιο συχνά χρησιμοποιούμενες είναι: η standard TTL, εκπροσωπείται με τη σειρά που αρχίζει με 74 και η Χαμηλής ισχύος Schottky που εκπροσωπείται με τη σειρά που αρχίζει με 74LS. Οι διαφορές μεταξύ διάφορων σειρών TTL είναι στα ηλεκτρικά τους χαρακτηριστικά, όπως η κατανάλωση ισχύος, η καθυστέρηση διάδοσης και η ταχύτητα εναλλαγής. Δεν διαφέρουν στις συνδέσεις των εξωτερικών ακροδεκτών ή στη λογική λειτουργία που επιτελείται από τα εσωτερικά κυκλώματα. Έτσι π.χ. τα 7486 και 74LS86 περιέχουν τέσσερις πύλες XOR με τις ίδιες συνδέσεις για τους εξωτερικούς ακροδέκτες σε κάθε περίβλημα.

Η τεχνολογία συμπληρωματικού MOS (CMOS) χρησιμοποιεί ένα PMOS και ένα NMOS τρανζίστορ συνδεδεμένα με ένα συμπληρωματικό τρόπο σε όλα τα κυκλώματα. Τα PMOS και NMOS είναι MOSFET μονοπολικά τρανζίστορ P και N καναλιού αντίστοιχα. Τα κυριότερα πλεονεκτήματα που παρουσιάζουν τα MOS έναντι των διπολικών τρανζίστορ είναι η υψηλή πυκνότητα κατασκευής των κυκλωμάτων, απλούστερη τεχνική επεξεργασίας κατά τη διάρκεια της κατασκευής, και η χαμηλότερη κατανάλωση ισχύος. Τα ολοκληρωμένα κυκλώματα CMOS μπορούν να λειτουργήσουν ορθά για τάσεις τροφοδοσίας από +3V έως +15V. Τάσεις στην είσοδο από 0 έως 30% της τάσης τροφοδοσία θεωρούνται σαν το λογικό «0», ενώ τάσεις από 70% έως 100% της τάσης τροφοδοσίας θεωρούνται σαν λογικό «1» στην είσοδο. Για παράδειγμα, με τάση τροφοδοσίας 5V, η χαμηλή στάθμη εισόδου κυμαίνεται από

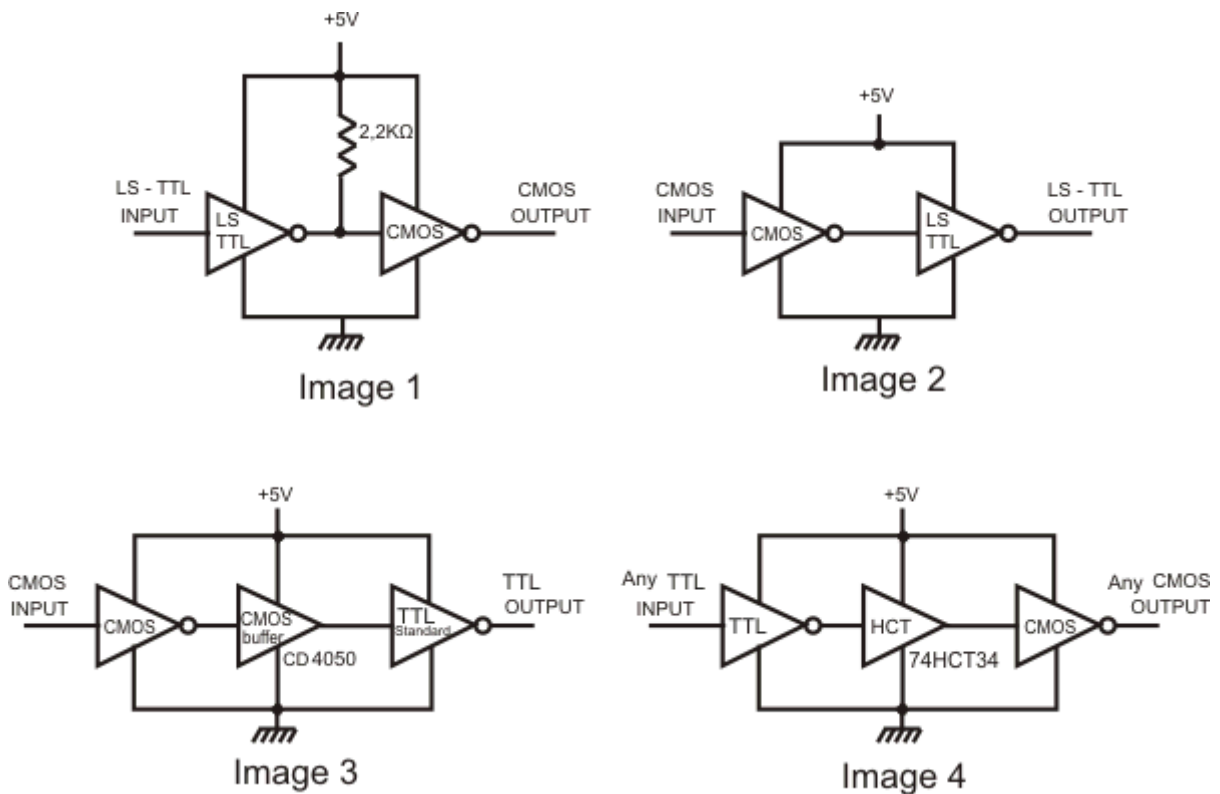
0 έως 1,5V, ενώ για τη υψηλή στάθμη εισόδου είναι από 3,5V έως 5V. Οι τάσεις εξόδου είναι σχεδόν ίσες με τις τάσεις τροφοδοσίας. Για παράδειγμα, για τάση τροφοδοσίας 5V, η έξοδος υψηλής στάθμης είναι περίπου 5V ενώ η χαμηλή στάθμη εξόδου είναι σχεδόν 0V.

Η αρχική σειρά 4000 των CMOS κυκλωμάτων σχεδιάστηκε ανεξάρτητα από τη σειρά TTL. Επειδή όμως τα TTL έγιναν βασικά κυκλώματα στη βιομηχανία οι κατασκευαστές άρχισαν να παράγουν άλλα κυκλώματα CMOS των οποίων οι εξωτερικοί ακροδέκτες είναι συμβατοί με παρόμοια ολοκληρωμένα TTL. Από αυτές, οι πιο διαδεδομένες CMOS σειρές είναι οι 74HC και 74HCT. Η σειρά 74HC λειτουργεί σε υψηλότερες ταχύτητες από την αρχική σειρά, ενώ η σειρά 74HCT είναι και ηλεκτρικά και σε επίπεδο ακροδεκτών συμβατή με την οικογένεια TTL. Αυτό σημαίνει ότι τα ολοκληρωμένα της σειράς 74HCT μπορούν να συνδεθούν απευθείας με ολοκληρωμένα TTL χωρίς τη χρήση ενδιάμεσων κυκλωμάτων.

	4000 Series	74HC Series	74HCT Series	74LS Series
Technology	CMOS	High-speed CMOS	High-speed CMOS TTL Compatible	TTL Low power Schottky
Power Supply	3 to 15V	2 to 6V	5V \pm 0,5V	5V \pm 0,25V
Maximum Frequency	about 1MHz	about 25MHz	about 25MHz	about 35MHz

Τα ψηφιακά ολοκληρωμένα κυκλώματα έχουν μια προδιαγραφή που λέγεται fan-out. Το fan-out ενός ψηφιακού IC είναι ο αριθμός των «τυπικών» εισόδων που μπορούν να οδηγηθούν από την έξοδο της πύλης. Εάν το fan-out μιας τυπικής TTL πύλης είναι 10, αυτό σημαίνει ότι η έξοδος μιας πύλης μπορεί να οδηγήσει έως και 10 εισόδους πυλών της ίδιας υπο-οικογένειας. Μια τυπική τιμή για το fan-out συνηθισμένων ολοκληρωμένων TTL είναι 10. Το fan-out για χαμηλής ισχύος Schottky TTL (LS – TTL) είναι 20 και για τη σειρά 4000 CMOS είναι περίπου 50.

Οι λογικές στάθμες (τάσεις) για είσοδο και έξοδο ψηφιακών κυκλωμάτων, είναι διαφορετικές για τα ολοκληρωμένα CMOS και TTL. Γι' αυτό αυτά τα ολοκληρωμένα συνήθως δεν μπορούν να συνδεθούν μαζί. Εξίσου σημαντικές είναι και οι διαφορετικές απαιτήσεις ρεύματος. Παρακάτω φαίνονται τρόποι διασύνδεσης TTL με CMOS και CMOS με TTL



Εικόνα 1 - Παραδείγματα συνδεσμολογίας CMOS – TTL

Στην εικόνα 1 φαίνεται το δημοφιλές LS-TTL να οδηγεί μια οποιαδήποτε πύλη CMOS. Παρατηρήστε τη χρήση αντιστάσεων pull-up 2,2kΩ (Στην περίπτωση που έχουμε Standard TTL η αντίσταση πρέπει να έχει τιμή 1kΩ). Αυτές χρησιμεύουν στο να διατηρούν τη HIGH TTL έξοδο κοντά στα 5V ώστε να είναι συμβατή με τα χαρακτηριστικά τάσης εισόδου των CMOS IC. Στην εικόνα 2 ένας αντιστροφέας CMOS (οποιαδήποτε σειράς) οδηγεί απευθείας ένα αντιστροφέα LS-TTL. Τα IC CMOS δεν μπορούν να οδηγήσουν (Standard) TTL εισόδους χωρίς ειδική προσαρμογή. Αυτό γίνεται με τον μη αντιστρέφων απομονωτή CD4050 όπως φαίνεται στην εικόνα 3. Ένας άλλος τρόπος διασύνδεσης TTL με CMOS, εκτός από την χρήση pull-up αντίστασης είναι η παρεμβολή ολοκληρωμένων κυκλωμάτων της σειράς CMOS 74HCT00 όπως φαίνεται στην εικόνα 4 που χρησιμοποιείται ο μη αναστρέφων IC 74HCT34.

Η ταχύτητα αντίδρασης σε μεταβολές εισόδου αποτελεί σημαντικό παράγοντα στις εφαρμογές υψηλής ταχύτητας των ψηφιακών ολοκληρωμένων κυκλωμάτων. Για παράδειγμα θεωρούμε ένα αντιστροφέα TTL (όπως το IC 7404). Όταν η είσοδος του μεταβαίνει από χαμηλή σε υψηλή στάθμη (από το λογικό 0 στο λογικό 1) υπάρχει καθυστέρηση διάδοσης

προς την έξοδο, που στην περίπτωση μας είναι 12ns. Όμοια στην μετάβαση από υψηλή σε χαμηλή στάθμη στην είσοδο έχουμε μια διαφορετική καθυστέρηση διάδοσης στην έξοδο, που εδώ είναι 7ns.

Το πρόβλημα γίνεται ακόμα μεγαλύτερο σε ψηφιακά κυκλώματα που έχουμε στη σειρά πολλά επίπεδα λογικών πυλών. Για να λύσουμε αυτό το πρόβλημα της καθυστέρησης διάδοσης του ψηφιακού σήματος, χρησιμοποιούμε παλμούς ορολογίου ώστε να έχουμε ταυτόχρονη μετάβαση των λογικών σταθμών σε όλο το ψηφιακό κύκλωμα που βηματοδοτείται, συνήθως με την θετική ή την αρνητική ακμή του παλμού ρολογιού. (βλέπε [8])

1.2 ΤΙ ΕΙΝΑΙ Ο ΜΙΚΡΟΕΛΕΓΚΤΗΣ

Ο μικροελεγκτής (αγγλικά, microcontroller) είναι ένας τύπος επεξεργαστή, ουσιαστικά μια παραλλαγή μικροεπεξεργαστή, ο οποίος μπορεί να λειτουργήσει με ελάχιστα εξωτερικά εξαρτήματα, λόγω των πολλών ενσωματωμένων υποσυστημάτων που διαθέτει. Χρησιμοποιείται ευρύτατα σε όλα τα ενσωματωμένα συστήματα (embedded systems) ελέγχου χαμηλού και μεσαίου κόστους, όπως αυτά που χρησιμοποιούνται σε αυτοματισμούς, ηλεκτρονικά καταναλωτικά προϊόντα (από ψηφιακές φωτογραφικές μηχανές έως παιχνίδια), ηλεκτρικές συσκευές και κάθε είδους αυτοκινούμενα τροχοφόρα οχήματα.



Εικόνα 2 - Μικροελεγκτής PIC της Microchip

Στους σύγχρονους μικροεπεξεργαστές για μη ενσωματωμένα συστήματα (πχ τους μικροεπεξεργαστές των προσωπικών υπολογιστών), δίνεται έμφαση στην υπολογιστική ισχύ. Η ευελιξία ανάπτυξης διαφορετικών εφαρμογών είναι μεγάλη, καθώς η λειτουργικότητα του τελικού συστήματος καθορίζεται από τα εξωτερικά περιφερειακά τα οποία διασυνδέονται με την κεντρική μονάδα (μικροεπεξεργαστή), η οποία δεν είναι εξειδικευμένη. Αντίθετα, στους μικροεπεξεργαστές για ενσωματωμένα συστήματα (μικροελεγκτές), οι οποίοι έχουν μικρότερες ή και μηδαμινές δυνατότητες συνεργασίας με εξωτερικά περιφερειακά, αυτού του είδους, η ευελιξία είναι περιορισμένη, καθώς και η υπολογιστική ισχύς. Οι μικροελεγκτές δίνουν έμφαση στο μικρό αριθμό ολοκληρωμένων κυκλωμάτων που απαιτείται για τη λειτουργία μιας συσκευής, το χαμηλό κόστος και την εξειδίκευση.

Αναλυτικά, τα πλεονεκτήματα των μικροελεγκτών είναι:

- Αυτονομία, μέσω της ενσωμάτωσης σύνθετων περιφερειακών υποσυστημάτων όπως μνήμες και θύρες επικοινωνίας. Έτσι πολλοί μικροελεγκτές δεν χρειάζονται κανένα άλλο ολοκληρωμένο κύκλωμα για να λειτουργήσουν.
- Η ενσωμάτωση περιφερειακών σημαίνει ευκολότερη υλοποίηση εφαρμογών λόγω των απλούστερων διασυνδέσεων. Επίσης, οδηγεί σε χαμηλότερη κατανάλωση ισχύος, μεγιστοποιώντας τη φορητότητα και ελαχιστοποιεί το κόστος της συσκευής στην οποία ενσωματώνεται ο μικροελεγκτής.
- Χαμηλό κόστος.
- Μεγαλύτερη αξιοπιστία, και πάλι λόγω των λιγότερων διασυνδέσεων.
- Μειωμένες εκπομπές ηλεκτρομαγνητικών παρεμβολών και μειωμένη ευαισθησία σε αντίστοιχες παρεμβολές από άλλες ηλεκτρικές και ηλεκτρονικές συσκευές. Το πλεονέκτημα αυτό προκύπτει από το μικρότερο αριθμό και μήκος εξωτερικών διασυνδέσεων καθώς και τις χαμηλότερες ταχύτητες λειτουργίας.
- Περισσότεροι διαθέσιμοι ακροδέκτες για ψηφιακές εισόδους-εξόδους (για δεδομένο μέγεθος ολοκληρωμένου κυκλώματος), λόγω της μη δέσμευσής τους για τη σύνδεση εξωτερικών περιφερειακών.
- Μικρό μέγεθος συνολικού υπολογιστικού συστήματος.

Η βασική αρχιτεκτονική των μικροελεγκτών δεν διαφέρει από αυτή των κοινών μικροεπεξεργαστών, αν και στους πρώτους είναι απαντάται συχνά η αρχιτεκτονική μνήμης τύπου Harvard, η οποία χρησιμοποιεί διαφορετικές αρτηρίες σύνδεσης της μνήμης

προγράμματος και της μνήμης δεδομένων (π.χ. οι σειρές AVR από την Atmel και PIC από την Microchip). Στους κοινούς μικροεπεξεργαστές συνηθίζεται η ενιαία διάταξη μνήμης τύπου von Neumann.

Στον μικροεπεξεργαστή, το ολοκληρωμένο κύκλωμα που τον αποτελεί περιέχει μόνο την Λογική και Αριθμητική Μονάδα (ALU), στοιχειώδεις καταχωρητές (registers), προσωρινή μνήμη RAM πολύ υψηλής ταχύτητας (cache memory) και, κάποιες φορές, τον ελεγκτή μνήμης (memory controller). Όμως, για τη λειτουργία ενός πλήρους ενσωματωμένου υπολογιστικού συστήματος, απαιτούνται πολλά εξωτερικά υποσυστήματα και περιφερειακά. Τέτοια είναι:

- Κύκλωμα συνδετικής λογικής (glue logic) για τη σύνδεση των εξωτερικών μνημών και άλλων περιφερειακών παράλληλης σύνδεσης στην αρτηρία δεδομένων (bus) του επεξεργαστή.
- Μνήμη προγράμματος (τύπου ROM, FLASH, EPROM κλπ) η οποία περιέχει το λογισμικό του συστήματος. Σε κάποια μοντέλα, είναι δυνατό το κλείδωμα αυτής της μνήμης, μετά την εγγραφή της, ώστε να προστατευτεί το περιεχόμενό της από αντιγραφή.
- Μεγάλο μέγεθος μνήμης RAM.
- Μόνιμη μνήμη αποθήκευσης παραμέτρων λειτουργίας (τύπου EEPROM ή NVRAM) η οποία να μπορεί να γράφεται τον πυρήνα του μικροελεγκτή. Αυτή η μνήμη έχει, έναντι της FLASH, το πλεονέκτημα της δυνατότητας διαγραφής και εγγραφής οποιουδήποτε μεμονωμένου byte.
- Κύκλωμα αρχικοποίησης (reset).
- Διαχειριστή αιτήσεων διακοπής (interrupt request controller) από τα περιφερειακά.
- Κύκλωμα επιτήρησης τροφοδοσίας (brown-out detection) το οποίο παρακολουθεί την τροφοδοσία και αρχικοποιεί ολόκληρο το σύστημα όταν αυτή πέσει κάτω από τα ανεκτά όρια, προλαμβάνοντας έτσι την αλλοίωση των δεδομένων.
- Κύκλωμα επιτήρησης λειτουργίας (watchdog timer) το οποίο αρχικοποιεί το σύστημα, αν αυτό εμφανίσει σημάδια δυσλειτουργίας λόγω κολλήματος (hang).
- Τοπικό ταλαντωτή για την παροχή παλμών χρονισμού (clock).
- Έναν ή περισσότερους χρονιστές-απαριθμητές υψηλής ταχύτητας (hardware timer-counter) για τη δημιουργία καθυστερήσεων, μέτρηση διάρκειας γεγονότων, απαρίθμηση γεγονότων και άλλων λειτουργιών ακριβούς χρονισμού.

- Ρολόι πραγματικού χρόνου (Real Time Clock, RTC) το οποίο τροφοδοτείται από ανεξάρτητη μπαταρία και γι αυτό πρέπει να έχει πολύ χαμηλή κατανάλωση ρεύματος.
- Σειρά ανεξάρτητων ψηφιακών εισόδων και εξόδων (Parallel Input-Output, PIO).

Γενικά, όλες οι οικογένειες μικροελεγκτών ενσωματώνουν τα περισσότερα από τα παραπάνω περιφερειακά, με διαφοροποιήσεις κυρίως στην ύπαρξη ή μη εσωτερικής μνήμης προγράμματος και στο είδος της. Έτσι, υπάρχουν:

- Μικροελεγκτές χωρίς μνήμη προγράμματος, οι οποίοι χαρακτηρίζονται ως ROM-less. Αυτοί παρέχουν πάντοτε μια παράλληλη αρτηρία (bus) δεδομένων, πάνω στην οποία συνδέονται εξωτερικές μνήμες προγράμματος και RAM. Τέτοιοι τύποι μικροελεγκτών προορίζονται για πιο ισχυρά υπολογιστικά συστήματα ελέγχου, με μεγαλύτερες απαιτήσεις μνήμης.
- Μικροελεγκτές με μνήμη ROM, η οποία κατασκευάζεται με το λογισμικό της (Mask ROM) ή γράφεται μόνο μια φορά (One Time Programmable, OTP). Παρέχουν τη δυνατότητα πολύ χαμηλού κόστους, όταν αγοράζονται σε πολύ μεγάλες ποσότητες.
- Μικροελεγκτές με μνήμη FLASH, οι οποία μπορούν συνήθως να προγραμματιστεί πολλές φορές. Αυτή είναι η πιο διαδεδομένη κατηγορία. Συχνά ο προγραμματισμός της μνήμης μπορεί να γίνει ακόμη και πάνω στο κύκλωμα της ίδιας της ενσωματωμένης (embedded) εφαρμογής (δυνατότητα In Circuit Programming, ISP). Αυτοί οι μικροελεγκτές έχουν ουσιαστικά αντικαταστήσει τους παλαιότερους τύπους EPROM που έσβηναν με υπεριώδη ακτινοβολία (από το ειδικό τζαμάκι).

Ανάλογα με την εφαρμογή για την οποία προορίζεται ένας μικροελεγκτής, μπορεί να περιέχει και:

- Μία ή περισσότερες ασύγχρονες σειριακές θύρες επικοινωνίας (Universal Asynchronous Receiver Transmitter, UART).
- Σύγχρονες σειριακές θύρες επικοινωνίας (πχ I²C, SPI, Ethernet).
- Ολόκληρα υποσυστήματα για την άμεση υποστήριξη από υλικολογισμικό (firmware) των πιο σύνθετων πρωτοκόλλων επικοινωνίας όπως CAN, HDLC, ISDN, ADSL.
- Μονάδα άμεσης εκτέλεσης πράξεων κινητής υποδιαστολής (Floating Point Processing Unit, FPU), η οποία είναι πάντοτε πιο γρήγορη από την ALU του επεξεργαστή. Τέτοιες μονάδες χαρακτηρίζουν τους μικροελεγκτές με δυνατότητες ψηφιακής επεξεργασίας

σήματος (Digital Signal Processing, DSP). Τα τελευταία χρόνια, με την ευρύτατη διάδοση των φορητών συσκευών ήχου και εικόνας, παρατηρείται μια τάση σύγκλισης των μικροελεγκτών με τους DSP.

- Περισσότερες από μία εισόδους για μετατροπή αναλογικού σήματος σε ψηφιακό (Analog to Digital converter, ADC).
- Μετατροπέα ψηφιακού σε αναλογικό σήμα (Digital to Analog converter, DAC).
- Ελεγκτή οθόνης υγρών κρυστάλλων (Liquid Crystal Display, LCD).
- Υποσύστημα προγραμματισμού πάνω στο κύκλωμα (τύπου ISP, βλ. παραπάνω). Χάρη σε αυτό το κύκλωμα, είναι δυνατός ο επαναπρογραμματισμός (αναβάθμιση λογισμικού) της εφαρμογής, συνδέοντας στη συσκευή μια εξωτερική συσκευή προγραμματισμού (συνήθως σε θύρα UART RS-232) ή ακόμη και από το διαδίκτυο. Αυτή η δυνατότητα απαιτεί την προϋπαρξη λογισμικού υποδοχής (bootstrap) μέσα στη μνήμη προγράμματος και επομένως δεν μπορεί να γίνει σε τελείως άδεια μνήμη προγράμματος.
- Υποσύστημα προγραμματισμού (τύπου ISP) και διάγνωσης (συνήθως είναι το καθιερωμένο πρότυπο JTAG). Χάρη σε αυτό, είναι δυνατός ο προγραμματισμός της μνήμης προγράμματος χωρίς να προαπαιτείται κάποιο πρόγραμμα υποδοχής. Γι αυτό το λόγο, είναι ιδιαίτερα χρήσιμο στον αρχικό προγραμματισμό, πχ κατά τη συναρμολόγηση, ή σε περίπτωση σφάλματος (bug) στο λογισμικό υποδοχής το οποίο να καθιστά αδύνατη την κανονική αναβάθμιση. (βλέπε [9])

1.3 ΚΑΤΗΓΟΡΙΕΣ ΜΙΚΡΟΕΛΕΓΚΤΩΝ ΚΑΙ ΚΑΤΑΣΚΕΥΑΣΤΕΣ

Λόγω του ισχυρότατου ανταγωνισμού αλλά και της τάσης ενσωμάτωσης των μικροελεγκτών σε κάθε ηλεκτρική και ηλεκτρονική συσκευή, η βιομηχανία μικροελεγκτών έχει καταλήξει στην παραγωγή ανταγωνιστικών μοντέλων μαζικής παραγωγής καθώς και μικροελεγκτών για πιο εξειδικευμένες εφαρμογές. Έτσι διακρίνονται οι εξής κυρίως κατηγορίες:

- Μικροελεγκτές (καμιά φορά 4-bit αλλά συνήθως 8-bit) πολύ χαμηλού κόστους, γενικής χρήσης, με πολύ μικρό αριθμό ακροδεκτών (ακόμη και λιγότερους από 8). Σχεδιάζονται

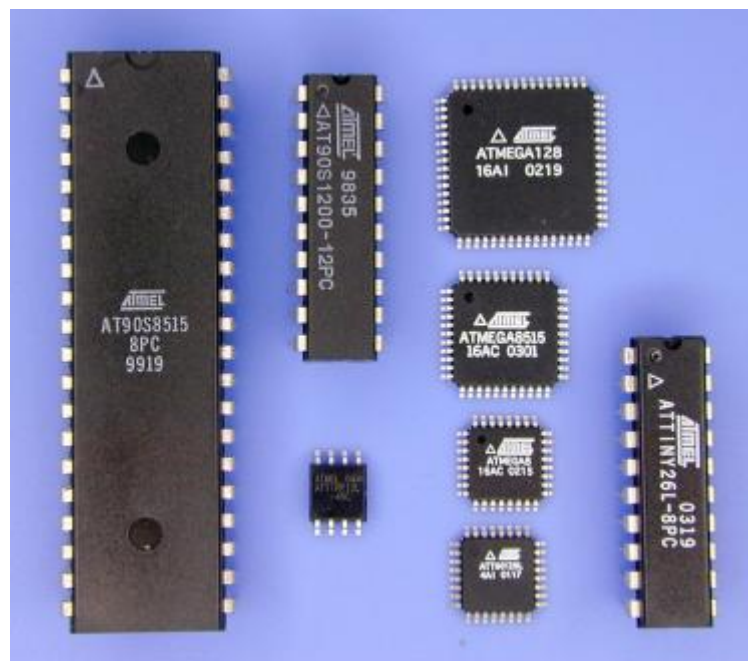
με έμφαση στη χαμηλή κατανάλωση ισχύος και την αυτάρκεια, ώστε να χρειάζονται ελάχιστα ή και καθόλου εξωτερικά εξαρτήματα και να μη μπορεί να αντιγραφεί εύκολα το εσωτερικό λογισμικό τους. Απουσιάζει η δυνατότητα επέκτασης της μνήμης τους. Μερικά μοντέλα είναι ευρέως γνωστά στους ερασιτέχνες ηλεκτρονικούς, όπως πχ οι περισσότεροι μικροελεγκτές των σειρών PIC (Microchip), AVR (Atmel) και 8051 (Intel, Atmel, Dallas κα).

- Μικροελεγκτές (συνήθως 8-bit αλλά και 16 ή 32-bit) χαμηλού κόστους, γενικής χρήσης, με μέτριο έως σχετικά μεγάλο αριθμό ακροδεκτών. Διαθέτουν μεγάλο αριθμό κοινών περιφερειακών, όπως θύρες UART, I²C, SPI ή CAN, μετατροπείς αναλογικού σε ψηφιακό και ψηφιακού σε αναλογικό. Στους κατασκευαστές της Άπω Ανατολής (Ιαπωνία, Κορέα), συνηθίζεται η ενσωμάτωση ελεγκτών οθόνης υγρών κρυστάλλων και πληκτρολογίου. Μερικές φορές παρέχουν δυνατότητα εξωτερικής επέκτασης της μνήμης τους.
- Μικροελεγκτές (κυρίως 32-bit) μέσου κόστους, γενικής χρήσης, με μεγάλο αριθμό ακροδεκτών. Χαρακτηρίζονται από έμφαση στην ταχύτητα εκτέλεσης εντολών, υψηλή αυτάρκεια περιφερειακών και μεγάλες δυνατότητες εσωτερικής ή εξωτερικής μνήμης προγράμματος (FLASH) και RAM. Στο χώρο αυτό έχουν ισχυρή παρουσία οι αρχιτεκτονικές με υψηλή μεταφερσιμότητα λογισμικού (portability) από τον ένα στον άλλο κατασκευαστή. Πχ μεταξύ των μικροελεγκτών τύπου ARM ή MIPS, το σύνολο των βασικών εντολών που αναγνωρίζει η ALU είναι ακριβώς το ίδιο, μειώνοντας έτσι τις μεγάλες αλλαγές στο λογισμικό, όταν στο μέλλον ο πελάτης υιοθετήσει ένα μικροελεγκτή άλλου κατασκευαστή (αρκεί, φυσικά, να υποστηρίξει κι αυτός το σύνολο εντολών ARM ή MIPS, αντίστοιχα).
- Μικροελεγκτές εξειδικευμένων εφαρμογών, οι οποίοι ενσωματώνουν συνήθως κάποιο εξειδικευμένο πρωτόκολλο επικοινωνίας το οποίο υλοποιείται πάντοτε σε hardware. Τέτοιοι μικροελεγκτές χρησιμοποιούνται σε τηλεπικοινωνιακές συσκευές όπως τα μόντεμ.

Η μεγάλη μερίδα πωλήσεων των μικροελεγκτών εξακολουθεί να αφορά αυτούς των 8-bit, καθώς είναι η κατηγορία με το χαμηλότερο κόστος και το μικρότερο μέγεθος λογισμικού για το ίδιο αποτέλεσμα, ιδίως επειδή οι σύγχρονες οικογένειες μικροελεγκτών 8-bit έχουν πολύ βελτιωμένες επιδόσεις σε σχέση με το παρελθόν. (βλέπε [10])

Μερικοί από τους γνωστότερους κατασκευαστές μικροελεγκτών είναι οι:

- ARM (δεν κατασκευάζει αλλά παραχωρεί δικαιώματα χρήσης του πυρήνα)
- Atmel
- Epson
- Freescale Semiconductor (πρώην Motorola)
- Hitachi
- Maxim (μετά την εξαγορά της Dallas)
- Microchip
- NEC
- Toshiba
- Texas Instruments



Εικόνα 3 - Διάφοροι Μικροελεγκτές της ATMEL



Εικόνα 4 - Μικροελεγκτής της Texas Instruments



Εικόνα 5 - Μικροελεγκτής της Toshiba

1.4 ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΣ ΜΙΚΡΟΕΛΕΓΚΤΩΝ ΚΑΙ ΕΡΓΑΛΕΙΑ ΑΝΑΠΤΥΞΗΣ

Η επιτυχία μιας οικογένειας μικροελεγκτών καθορίζεται σε μεγάλο βαθμό από τη διαθεσιμότητα και την ευχρηστία των σχετικών εργαλείων ανάπτυξης, όπως μεταφραστές από γλώσσες υψηλού επιπέδου σε γλώσσα κατανοητή από τον μικροελεγκτή (Assembly), προγραμματιστές της εσωτερικής μνήμης και εργαλεία εκσφαλμάτωσης (debuggers). Στους μικροελεγκτές, τα εργαλεία αυτά δεν αποτελούνται ποτέ μόνο λογισμικό, καθώς δεν υπάρχει τυποποιημένος τρόπος επικοινωνίας με αυτούς. Στον τομέα των εργαλείων ανάπτυξης, δραστηριοποιούνται όχι μόνο οι ίδιοι οι κατασκευαστές μικροελεγκτών αλλά και εξειδικευμένες εταιρείες.

Η πιο διαδεδομένη γλώσσα προγραμματισμού των μικροελεγκτών είναι η C, η C++ και οι παραλλαγές τους. Σε τμήματα του λογισμικού όπου απαιτείται ταχύτητα η μικρό μέγεθος χρησιμοποιούμενης μνήμης, μπορεί να χρησιμοποιείται η Assembly. Όμως οι μεγαλύτερες

απαιτήσεις σε λειτουργικότητα και η ευκολία προγραμματισμού της C έναντι της assembly, σε συνδυασμό με την επάρκεια μνήμης των σύγχρονων μικροελεγκτών, έχουν γενικά εκτοπίσει την Assembly από τις περισσότερες εφαρμογές. (βλέπε [11])

2 ΠΑΡΟΥΣΙΑΣΗ ΣΤΙΣ ΤΕΧΝΟΛΟΓΙΕΣ GPS-GSM/3G

2.1 ΑΝΑΛΥΣΗ ΣΥΣΤΗΜΑΤΩΝ GPS

Από αρχαιοτάτων χρόνων ο άνθρωπος, στην προσπάθεια του να μπορέσει να μετακινηθεί από ένα μέρος σ' ένα άλλο πάντα συναντούσε δυσκολίες, με πιο λαμπρό παράδειγμα αυτό του Οδυσσέα, ο οποίος ταλαιπωρήθηκε 10 χρόνια για να μπορέσει να φτάσει σπίτι του. Κάποια στιγμή λοιπόν ο άνθρωπος άρχισε να ψάχνει τρόπους για να μπορέσει να μετακινηθεί από ένα μέρος σ' ένα άλλο χωρίς να χρειάζεται πολύ χρόνο και κυρίως χωρίς να ταλαιπωρείται. Μέχρι τη δεκαετία 1960 οι μόνοι τρόποι για να μπορέσει να μετακινηθεί κάποιος χωρίς να χαθεί ήταν να χρησιμοποιεί φυσικά σταθερά μέσα, όπως η χρήση του ήλιου ή των αστεριών. Πλέον έρχεται στο προσκήνιο μια νέα επιστήμη που ασχολείται με την μετάβαση ενός ανθρώπου από ένα μέρος σ' ένα άλλο στο λιγότερο δυνατό χρόνο, αυτή δεν είναι άλλη από την πλοήγηση.

Κάθε ένας από εμάς στην καθημερινότητα του διεξάγει κάποιου είδους πλοήγηση. Το να οδηγεί κανείς για την δουλειά ή το να δίνει πληροφορίες για μια διαδρομή απαιτεί κάποιες βασικές γνώσεις πλοήγησης. Για τους περισσότερους από εμάς, αυτές οι ικανότητες απαιτούν την χρησιμοποίηση των ματιών, της κοινής λογικής μας και σημείων αναφοράς. Παρ' όλα αυτά, σε ορισμένες περιπτώσεις όπου απαιτείται να γνωρίζουμε την ακριβή θέση της τοποθεσίας μας, τον ακριβή προορισμό ή το χρόνο μετάβασης σε έναν συγκεκριμένο προορισμό τα βοηθήματα της πλοήγησης, οι γνωστές σε όλους μας συσκευές GPS, είναι περισσότερο χρήσιμα από τα σημεία αναφοράς.

Τα βοηθήματα πλοήγησης θα μπορούσαν να έχουν την μορφή ενός απλού ρολογιού για να υπολογίζεται η ταχύτητα για μια γνωστή διαδρομή ή τη μορφή ενός οδόμετρου στο αυτοκίνητο μας για να παρακολουθείται η απόσταση που καλύφθηκε. Κάποια άλλα μετατρέπουν ηλεκτρονικά σήματα και ως εκ τούτου είναι πιο πολύπλοκα. Τα σήματα από ένα ή περισσότερα συστήματα βοηθούν τον χρήστη να μπορέσει να υπολογίσει την τοποθεσία του. Κάποια συστήματα παρέχουν την ικανότητα του προσδιορισμού της ταχύτητας και της διάδοσης του χρόνου. Είναι σημαντικό να τονίσουμε ότι ο ραδιοδέκτης του χρήστη επεξεργάζεται τα σήματα και υπολογίζει την ακριβή θέση. Ο ραδιοδέκτης εκτελεί τους απαραίτητους υπολογισμούς (π.χ. την σχέση μεταξύ της απόστασης και του εκτιμώμενου χρόνου αφίξεως) για να μπορέσει ο χρήστης να πλοηγηθεί στον επιθυμητό προορισμό. Σε

μερικές εφαρμογές, ο δέκτης επεξεργάζεται τα λαμβανόμενα σήματα εν μέρει με τους υπολογισμούς να γίνονται καθώς η τοποθεσία αλλάζει.

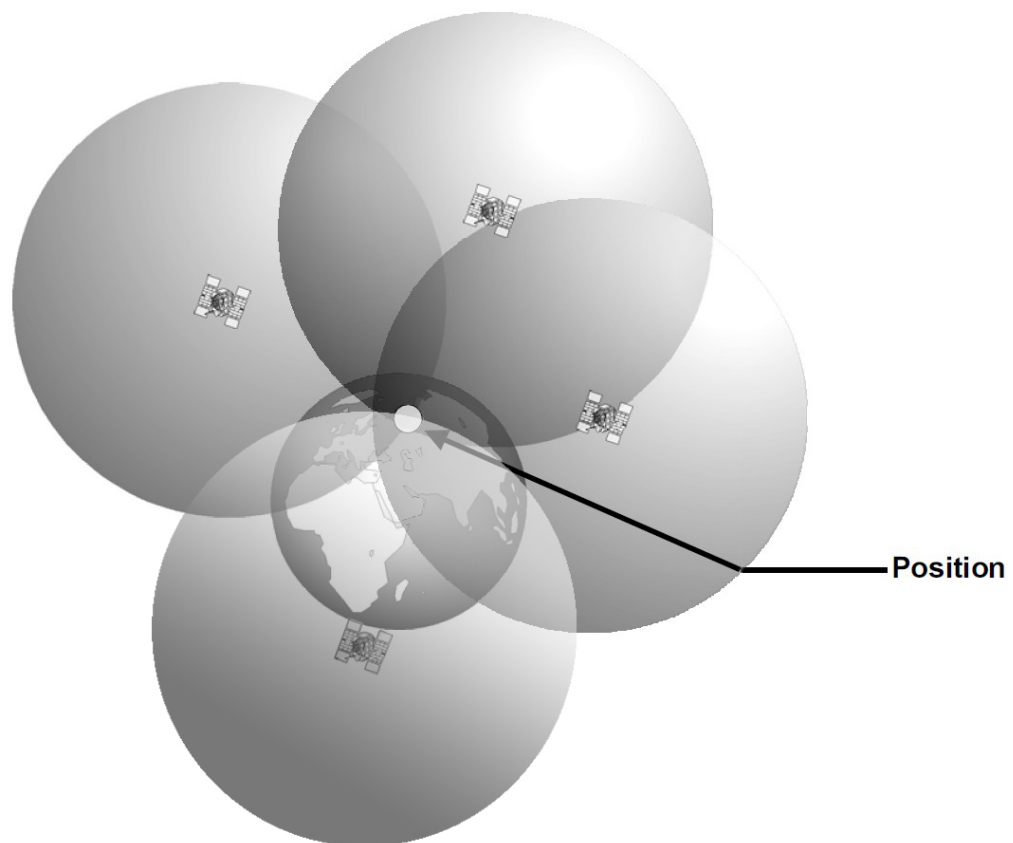
Στις αρχές της δεκαετίας του '60, κάποιες Αμερικανικές κυβερνητικές οργανώσεις συμπεριλαμβανομένου του Υπουργείου Αμύνης, της Εθνικής Υπηρεσίας Αεροναυτικής και Διαστήματος (NASA) και του Υπουργείου Μεταφορών, ενδιαφέρθηκαν να αναπτύξουν ένα δορυφορικό τρισδιάστατο σύστημα εντοπισμού θέσης. Το βέλτιστο σύστημα έπρεπε να περιλαμβάνει τα ακόλουθα χαρακτηριστικά: παγκόσμια κάλυψη, συνεχής και παντός καιρού λειτουργία και υψηλή ακρίβεια. Αυτό το σύστημα ονομάστηκε «Transit». Όταν το σύστημα μπήκε σε λειτουργία το 1964, ήταν ευρέως αποδεκτό για χρήση σε χαμηλής δυναμικής πλατφόρμες. Παρ' όλα αυτά, λόγω των έμφυτων δυνατοτήτων του, το Ναυτικό αποφάσισε να βελτιώσει το σύστημα ή να αναπτύξει ένα άλλο δορυφορικό σύστημα πλοήγησης με τα προαναφερθείσα επιθυμητά χαρακτηριστικά. Αρκετές παραλλαγές από το αρχικό σύστημα προτάθηκαν από τους προγραμματιστές του στο Πανεπιστήμιο Josh Hopkins στο εργαστήριο Εφαρμοσμένης Φυσικής. Το ίδιο χρονικό διάστημα το Ναυτικό εργαστήριο ερευνών διεξήγαγε πειράματα υψηλής ευστάθειας με διαστημικά ρολόγια για να επιτύχει ακριβή μεταφορά ώρας. Αυτό το πρόγραμμα ονομάστηκε "Timation". Έγιναν τροποποιήσεις στους δορυφόρους του "Timation" για να είναι εφικτό να παρέχουν μια ικανή μέτρηση για τον προσδιορισμό θέσης ενός δυσδιάστατου συστήματος. Κατά το ίδιο χρονικό διάστημα καθώς βελτιώσεις γίνονταν και στο αρχικό σύστημα και στο "Timation", η πολεμική αεροπορία δημιούργησε ένα δορυφορικό σύστημα θέσης που ονομάστηκε «Σύστημα 621B». Αναμενόταν ότι οι δορυφόροι του «Συστήματος 621B» θα κινούνταν σε ελλειπτικές τροχιές με γωνίες κλίσης 0° , 30° και 60° . Πολυάριθμες παραλλαγές του αριθμού των δορυφόρων και των τροχιών τους εξετάστηκαν. Προτάθηκε η χρήση διαμόρφωσης ενός ψεύδο-τυχαίου θορύβου με ψηφιακά σήματα για να επιτευχθεί η μέθοδος ranging, με την οποία μετράται η απόσταση. Το «Σύστημα 621B» φτιάχτηκε για να παρέχει τρισδιάστατη κάλυψη και συνεχή παγκόσμια κάλυψη. Η ιδέα και οι λειτουργικές τεχνικές επαληθεύτηκαν στο Yuma Proving Grounds όπου χρησιμοποιήθηκε μια αντίστροφη ακτίνα στην οποία οι ψεύδο-δορυφόροι μετέδιδαν δορυφορικό σήμα για τον εντοπισμό θέσης ενός αεροσκάφους. Επιπρόσθετα, ο στρατός στο Fort Monmouth στο New Jersey ερευνούσε πολλές τεχνικές, συμπεριλαμβανομένων των μεθόδου ranging, καθορισμού της γωνίας και χρήση μετρήσεων Doppler. Από τα αποτελέσματα των ερευνών, προτάθηκε να χρησιμοποιηθεί η μέθοδος ranging με τη χρήση διαμόρφωσης ψευδο-τυχαίου θορύβου. Το 1969, το Αμερικανικό

Υπουργείο Αμύνης δημιούργησε το Δορυφορικό Αμυντικό Σύστημα Πλοήγησης (DNSS) για να εδραιώσει τις ανεξάρτητες προσπάθειες ανάπτυξης κάθε στρατιωτικής υπηρεσίας για την δημιουργία ενός συστήματος κοινής χρήσης. Το Υπουργείο Αμύνης επίσης δημιούργησε την ομάδα εκτελεστικής διεύθυνσης δορυφορικής πλοήγησης, η οποία είχε ως στόχο τον καθορισμό της βιωσιμότητας του συστήματος DNSS και τον σχεδιασμό της ανάπτυξης του. Από αυτήν την προσπάθεια, δημιουργήθηκε η ιδέα για το NAVSTAR GPS. Το συγκεκριμένο πρόγραμμα αναπτύχθηκε από το γραφείο GPS Joint Program που βρίσκεται στην περιοχή El Segundo στην Καλιφόρνια. Με την πάροδο των ετών το προαναφερθέν γραφείο συνέχισε να επιβλέπει την ανάπτυξη και την παραγωγή νέων δορυφόρων, εξοπλισμού ελέγχου εδάφους και την πλειοψηφία αμερικανικών στρατιωτικών δεκτών χρήστη. Πλέον, το σύστημα είναι ευρέως διαδεδομένο ως GPS (Global Positioning System), Παγκόσμιο Σύστημα Προσδιορισμού Θέσης. (βλέπε [1])

Βασικός προορισμός του συστήματος από το 1978 που έγινε η εκτόξευση του πρώτου δορυφόρου, ήταν ο έλεγχος της κίνησης οχημάτων, πλοίων και αεροπλάνων σε παγκόσμια κλίμακα, για την ικανοποίηση των αναγκών του αμερικανικού στρατού. Με την πάροδο του χρόνου όμως και με την βελτίωση της ακρίβειας του συστήματος, οι εφαρμογές του επεκτάθηκαν και εξακολουθούν να επεκτείνονται και σε άλλους τομείς.

Η πλοήγηση μέσω του συστήματος GPS στηρίζεται στη μετάδοση πληροφοριών από το δορυφόρο προς τους επίγειους δέκτες. Η βασική λοιπόν ιδέα πίσω από το GPS είναι σχετικά απλή. Αν οι αποστάσεις από ένα σημείο στη γη, στην περίπτωση μας ο δέκτης του GPS, προς τρεις δορυφόρους GPS είναι γνωστές, όπως και τα σημεία των δορυφόρων, τότε η θέση του σημείου (δέκτη) μπορεί να προσδιοριστεί ακολουθώντας συγκεκριμένη μεθοδολογία. Κάθε δορυφόρος GPS εκπέμπει συνεχώς ραδιοκύματα. Όταν ένας δέκτης GPS μπαίνει σε λειτουργία λαμβάνει τα σήματα του δορυφόρου μέσω της κεραίας του και τα επεξεργάζεται. Ένα μέρος της επεξεργασίας του σήματος μας δίνει τις αποστάσεις προς τους δορυφόρους μέσω ενός ψηφιακού 7κώδικα όπως και τον χρόνο που χρειάστηκε το σήμα για να διανύσει την απόσταση αυτή (ταχύτητα μετάδοσης ήχου στον αέρα=344m/sec). Θεωρητικά λοιπόν μόνο τρεις αποστάσεις από τρεις ταυτόχρονα εντοπισμένους δορυφόρους χρειάζεται για να εντοπίσουμε ένα σημείο πάνω στη γη. Στην περίπτωση αυτή η θέση του δέκτη θα βρίσκεται στην τομή 3 σφαιρών με κέντρα τα σημεία των δορυφόρων και ακτίνες τις αντίστοιχες αποστάσεις των δορυφόρων προς τον δέκτη Από την πρακτική όμως πλευρά του θέματος

χρειάζονται τέσσερις παρατηρήσεις από τέσσερις δορυφόρους για τον προσδιορισμό της θέσης του δέκτη, αλλά και του χρόνου ως προς το σύστημα αναφοράς χρόνου του συστήματος GPS (χρόνος GPS), όπως φαίνεται και στην εικόνα παρακάτω. Αυτό συμβαίνει γιατί σκοπός της πλοήγησης είναι ο προσδιορισμός της θέσης σε πραγματικό χρόνο, ώστε να είναι δυνατή η λήψη αποφάσεων για την παραπέρα πορεία του πλοηγούμενου μέσου (πλοίου, αεροπλάνου, οπλικού συστήματος). Η ακρίβεια που επιτυγχάνεται με την μέθοδο αυτή (Standard Positioning Service) ήταν μέχρι πρόσφατα περιορισμένη στα 100 m στον οριζόντιο άξονα, 156m στον κάθετο άξονα και 340ns όσον αφορά το χρόνο, σε ένα γενικό επίπεδο ακρίβειας 95%.



Εικόνα 6 - Τέσσερις ταυτόχρονα εντοπισμένους δορυφόρους απαιτεί το GPS

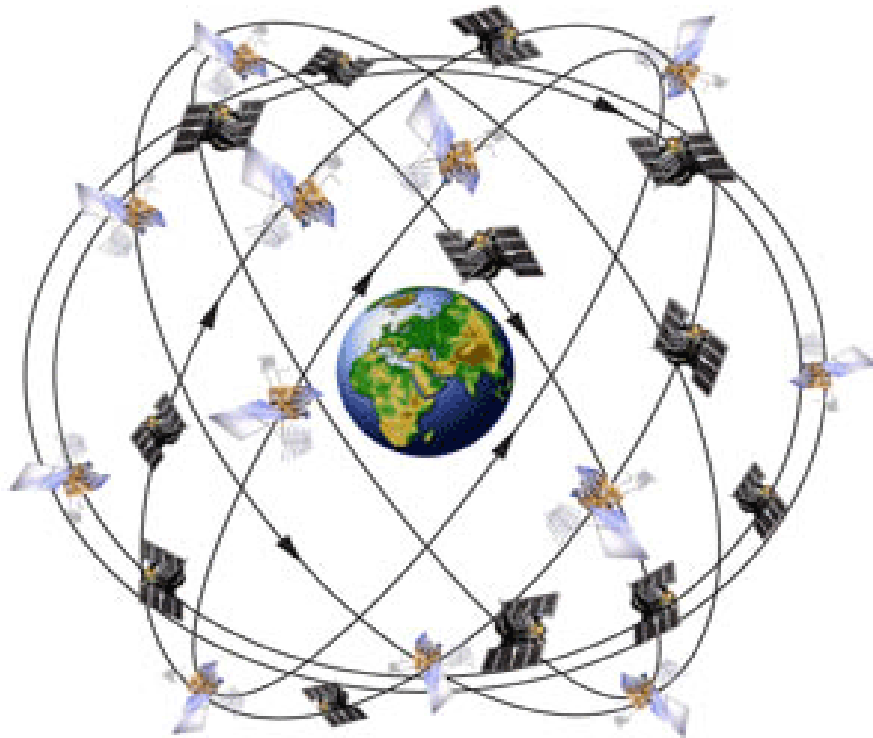
Αυτό ακριβώς το χαμηλό επίπεδο ακρίβειας είναι το αποτέλεσμα μιας τεχνικής μεθόδου σκόπιμης μείωσης της ακρίβειας του συστήματος (επιλεκτική διαθεσιμότητα). Με τις

πρόσφατες όμως αποφάσεις τερματισμού της μεθόδου αυτής, η οριζόντια ακρίβεια του σημείου βελτιώθηκε κατά 10 φορές. Ένας χρήστης του συστήματος, μπορεί πλέον να προσδιορίσει στιγμιαία τη θέση του με ακρίβεια καλύτερη από 10m και την ταχύτητά του με ακρίβεια 0,1 του κόμβου, όμως αν χρησιμοποιήσει το σύστημα ταυτόχρονα με ένα σταθμό αναφοράς, που κι αυτός μετράει προς τους ίδιους δορυφόρους, τότε μπορεί να προσδιορίσει τη θέση του ως προς το σταθμό αναφοράς με ακρίβεια 0,5 του μέτρου και την ταχύτητά του με ακρίβεια 0,05 του κόμβου. Στη δορυφορική γεωδαισία, ειδικές μετρήσεις και μέθοδοι επεξεργασίας επιτρέπουν τον εντοπισμό ενός ακίνητου σημείου ως προς ένα σταθμό αναφοράς με ακρίβεια μερικών εκατοστών σε αποστάσεις εκατοντάδων χιλιομέτρων. Τα σήματα των δορυφόρων μπορούν να διαπεράσουν τα σύννεφα, το γυαλί και το πλαστικό, αλλά όχι τα συμπαγή στερεά αντικείμενα, όπως τα κτίρια και τα βουνά. (βλέπε [2])

Το σύστημα του GPS αποτελείται από 3 τμήματα:

- Το δορυφορικό τμήμα
- Το τμήμα ελέγχου και
- Το τμήμα του χρήστη.

Το δορυφορικό τμήμα στηρίζεται σε ένα αστερισμό 24 ων δορυφόρων που για πρώτη φορά συμπληρώθηκε τον Ιούλιο του 1993. Για να εξασφαλιστεί πλήρη και παγκόσμια κάλυψη οι δορυφόροι τοποθετήθηκαν σε 6 προμελετημένες τροχιές όπου στην κάθε μια αντιστοιχούσαν 4 δορυφόροι. Στόχος του συγκεκριμένου γεωμετρικού αστερισμού ήταν κάθε στιγμή να γίνονται ορατοί από κάθε σημείο πάνω στη γη, τουλάχιστον 4 δορυφόροι, ενώ πολλές φορές ορατοί μπορούσαν να γίνουν ακόμη και 10 δορυφόροι. Στις 17 Ιουλίου 1995 το σύστημα του GPS μπαίνει επίσημα σε ένα νέο καθεστώς εξασφαλίζοντας την χρησιμότητα για τουλάχιστον 24 δορυφόρους σε τροχιά. Από εκείνη τη στιγμή και μετά ο αριθμός των δορυφόρων τους συστήματος ήταν πάντα μεγαλύτερος του 24. Ο πλήρης σχηματισμός των 24ων δορυφόρων παρουσιάζεται στην παρακάτω εικόνα.



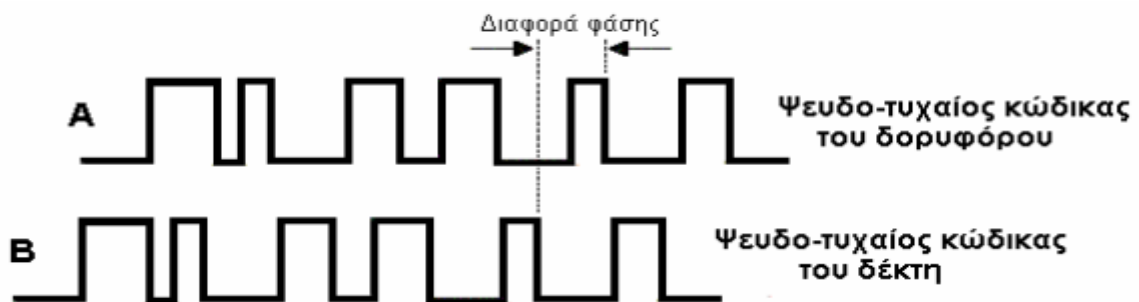
Εικόνα 7 - Τα 6 τροχιακά επίπεδα των 24ων δορυφόρων του GPS

Τα 6 τροχιακά επίπεδα έχουν περίπου 55° κλίση (σχετικά ίδια με αυτή μεταξύ της Γής και του Ισημερινού) και χωρίζονται κατά 60° δεξιά της ανόδου του ανερχόμενου κόμβου (γωνία κατά μήκος του Ισημερινού από ένα σημείο αναφοράς σε διασταύρωση της τροχιάς του). Η τροχιακή περίοδος είναι ένα μισό μιας αστρικής ημέρας, δηλαδή 11 ώρες και 58 λεπτά, έτσι ώστε οι δορυφόροι να περνούν πάνω από τις ίδιες περιοχές, ή σχεδόν από τις ίδιες θέσεις κάθε ημέρα. Οι τροχιές διατάσσονται έτσι ώστε τουλάχιστον έξι δορυφόροι να είναι πάντα σε θέση ορατή από σχεδόν παντού στην επιφάνεια της Γης. Το αποτέλεσμα του στόχου αυτού είναι ότι οι τέσσερις δορυφόροι δεν είναι ομοιόμορφα κατανεμημένοι, εκτός αν βρίσκονται στην ίδια τροχιά. Σε γενικές γραμμές, η γωνιακή διαφορά μεταξύ των δορυφόρων σε κάθε τροχιά είναι 30° , 105° και 120° .

Οι τροχιές βρίσκονται σε υψόμετρο περίπου 20.200 χιλιόμετρα, η τροχιακή ακτίνα είναι περίπου 26.600 χιλιόμετρα, κάθε δορυφόρος κάνει δύο πλήρεις τροχιές κάθε αστρική ημέρα επαναλαμβάνοντας την ίδια διαδρομή κάθε μέρα. Αυτό ήταν πολύ χρήσιμο κατά τη διάρκεια της ανάπτυξης, διότι ακόμη και με μόνο τέσσερις δορυφόρους σήμαινε ότι με σωστή

ευθυγράμμιση και οι τέσσερις είναι ορατοί από ένα σημείο για λίγες ώρες της ημέρας. Για τις στρατιωτικές επιχειρήσεις, η επανάληψη της τροχιάς του εδάφους μπορούσε να χρησιμοποιηθεί για να εξασφαλιστεί η καλή κάλυψη σε εμπόλεμες ζώνες. (βλέπε [3])

Η λειτουργία του GPS είναι η εξής. Ο GPS δέκτης παράγει έναν ψευδο-τυχαίο κώδικα, ο οποίος είναι ακριβές αντίγραφο του κώδικα που παράγει ο δορυφόρος. Ο κώδικας στον δέκτη είναι καταχωρημένος στη βάση δεδομένων του και παράγεται ταυτόχρονα με τον κώδικα του δορυφόρου. Ο δέκτης συγκρίνει τον κώδικα που παράγει ο ίδιος με τον κώδικα που λαμβάνει από το δορυφόρο και προσπαθεί να τους ταιριάξει. Επειδή όμως ο κώδικας του δορυφόρου διανύει μεγάλη απόσταση, έρχεται στο δέκτη με κάποια χρονική καθυστέρηση, η οποία φαίνεται και σα διαφορά φάσης. Για να υπολογίσει το χρόνο καθυστέρησης του σήματος από το δορυφόρο, ο δέκτης ολισθαίνει χρονικά προς τα πίσω το δικό του κώδικα, μέχρι να τον ταιριάξει με τον κώδικα του δορυφόρου. Το μέγεθος της χρονικής ολίσθησης ισούται με το χρόνο μετάδοσης του σήματος από το δορυφόρο στο δέκτη. Τότε, σύμφωνα με τον τύπο: $\text{Απόσταση} = \text{Ταχύτητα} \times \text{Χρόνος}$, πολλαπλασιάζει το χρόνο αυτό με την ταχύτητα του φωτός και υπολογίζει την απόσταση που απέχει ο δέκτης από το δορυφόρο. Παρακάτω δίνεται μια απεικόνιση των δυο ψευδοτυχαίων σημάτων. (βλέπε [2])



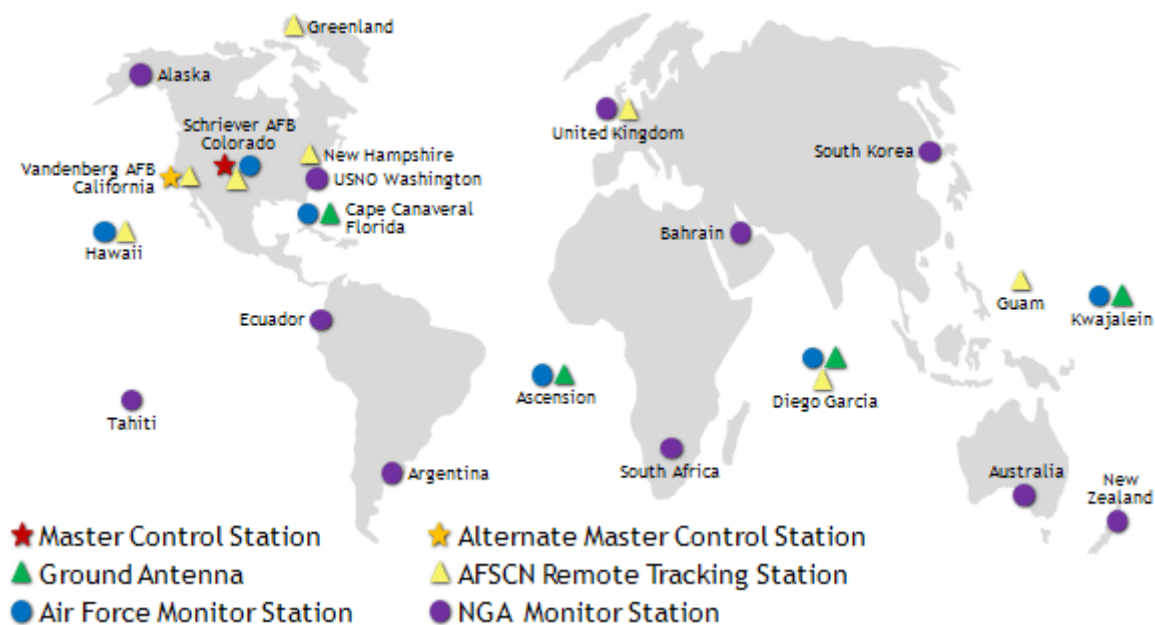
Εικόνα 8 - Απεικόνιση των 2 ψευδοτυχαίων σημάτων

Το τμήμα ελέγχου (Operational Control System - OCS) είναι υπεύθυνο για την διαχείριση του συστήματος GPS. Αποτελείται από τους σταθμούς παρακολούθησης των δορυφόρων, τους σταθμούς εκπομπής πληροφορίας προς τους δορυφόρους και το κεντρικό σταθμό ελέγχου (Master Control Station - MCS) που βρίσκεται στην Αεροπορική Βάση στο Colorado Springs των ΗΠΑ. Στις αρμοδιότητες του OCS συμπεριλαμβάνονται:

- η συνεχής παρακολούθηση των δορυφόρων,
- η υγιής συμπεριφορά τους σε θέματα όπως συγχρονισμός ρολογιών, επίπεδο αποθεμάτων πηγών ενέργειας, εκπομπή ραδιοκυμάτων και δεδομένων και τέλος
- εργάζεται για την επίλυση τυχόν προβλημάτων – ανωμαλιών.

Οι σταθμοί παρακολούθησης είναι 5. Βρίσκονται κατανεμημένοι στο Colorado Springs, στην Hawaii, στη νήσο Ascension στο Νότιο Ατλαντικό, στο Diego Garcia στον Ινδικό ωκεανό, στο Kwajalein στο Νότιο Ειρηνικό ωκεανό και στο Cape Canaveral στη Φλόριντα. Στην εικόνα που ακολουθεί φαίνονται όλοι οι σταθμοί παρακολούθησης με μπλέ κουκίδα ανά την υφήλιο. Όλοι οι σταθμοί είναι εφοδιασμένοι με δέκτες οι οποίοι συνεχώς μετρούν ψευδοαποστάσεις (γίνονται μετρήσεις κάθε 1,5 δευτερόλεπτο) των «ορατών» δορυφόρων στις συχνότητες L_1 / L_2 . Αυτές μετά από μια αρχική επεξεργασία μεταδίδονται στον κύριο σταθμό ελέγχου. Στον MCS επίσης συγκεντρώνονται οι παρατηρήσεις από 5 επιπλέον σταθμούς του στρατού των ΗΠΑ, ενώ τέλος υπολογίζονται οι διαθέσιμες οι ακριβείς πληροφορίες αποστολής δεδομένων. Οι σταθμοί παρακολούθησης είναι μη επανδρωμένες εγκαταστάσεις και παρέχουν κάλυψη των δορυφόρων περίπου 92%. Αυτό οφείλεται στο γεγονός ότι οι δορυφόροι δεν είναι συνεχώς «ορατοί» από τους σταθμούς.

Ο κεντρικός σταθμός ελέγχου (MCS) ήταν εγκατεστημένος στο Vandenberg AFB, στην California αλλά μεταφέρθηκε στην Βάση Falcon Air Force στο Colorado των ΗΠΑ. Είναι επανδρωμένος 24 ώρες το 24ωρο από ανθρώπους των Αμερικανικών Αεροπορικών Δυνάμεων ώστε να εξασφαλίζεται η σωστή λειτουργία όλων των συστημάτων. Στον MCS γίνεται επεξεργασία όλων των παρατηρήσεων που λαμβάνονται από τους σταθμούς παρακολούθησης ενώ καθορίζονται (προβλέπονται) οι δορυφορικές τροχιές καθώς και οι παράμετροι των ρολογιών τους. Στη συνέχεια οι σχετικές παράμετροι μεταδίδονται στους σταθμούς εκπομπής πληροφορίας οι οποίοι είναι τέσσερις (Ascension Island, Diego Garcia, Kwajalein, Cape Canaveral). Οι παράμετροι αποτελούν την μεταδιδόμενη πληροφορία που στηρίζεται στην ανάλυση των προηγούμενων παρατηρήσεων. Οι πληροφορίες που παρέχονται στους δορυφόρους ανανεώνονται μια ή δυο φορές την ημέρα, και αποθηκεύονται στην μνήμη των ηλεκτρονικών υπολογιστών των δορυφόρων.



Εικόνα 9 - Πλήρης χάρτης της ακριβούς θέσης των επίγειων σταθμών του συστήματος GPS

Το τρίτο και τελευταίο τμήμα του GPS, είναι το τμήμα του χρήστη. Αποτελείται από τους δέκτες του GPS οι οποίοι με την συνεχιζόμενη ανάπτυξη νέων επαναστατικών τεχνολογιών, εξελίσσονται διαρκώς. Οι αρχικοί δέκτες ήταν ογκώδης και με μεγάλο βάρος. Στις μέρες μας οι δέκτες είναι μικρές συσκευές, χαμηλού βάρους με ευκολία στην χρήση. (βλέπε [4])

Αποτελούνται από:

- Την κεραία
- Την συσκευή εισόδου / εξόδου (input / output device)
- Τον κυρίως δέκτη
- Τον προενισχυτή
- Τις μονάδες τροφοδοσίας



Εικόνα 10 - Παλαιότερης γενιάς δέκτης GPS



Εικόνα 11 - Σύγχρονος δέκτης GPS

2.2 ΤΕΧΝΟΛΟΓΙΕΣ GSM ΚΑΙ 3G

Οι κινητές επικοινωνίες έκαναν για πρώτη φορά την εμφάνιση τους τη δεκαετία του 1940, όταν τα ραδιοτηλέφωνα υπήρξαν η πρώτη μορφή διαπροσωπικής επικοινωνίας. Οι χρήστες των ραδιοτηλεφώνων πατώντας ένα κουμπί είχαν τη δυνατότητα να μιλήσουν μεταξύ τους ανταλλάσσοντας πληροφορίες. Η χρήση τους όμως ήταν περιορισμένη και εντοπισμένη σε συγκεκριμένες εφαρμογές (στρατιωτικές, επικοινωνίες πλοίων, αεροσκαφών). Η πλειονότητα των πολιτών δεν είχε τη δυνατότητα να χρησιμοποιήσει τις εφαρμογές αυτές.

Μόλις το 1961 το παράρτημα της Σουηδικής εταιρείας Ericsson με την επωνυμία Svenska Radio Aktiebolaget (SRA) παρήγαγε για πρώτη φορά διεθνώς εξοπλισμό κινητών επικοινωνιών για εμπορική χρήση, απευθυνόμενο στο ευρύ κοινό. Το πρώτο δίκτυο κινητών επικοινωνιών που καταγράφεται διεθνώς είναι αυτό που ανέπτυξε η εταιρεία Bell Systems, στην περιοχή της Πενσυλβάνια των ΗΠΑ, μεταξύ 1962 - 1964 και του έδωσε την ονομασία Improved Mobile Telephone Service (IMTS). Το δίκτυο αυτό ήταν το πρώτο στον κόσμο που επέτρεπε την αμφίδρομη επικοινωνία (full duplex) και έτσι οι συνομιλητές δεν αναγκάζονταν να πιέζουν πλήκτρο στο τερματικό τους για να μπορούν να ομιλούν. Το σύστημα αυτό αναπτύχθηκε στις ΗΠΑ και στον Καναδά, παρέχοντας τηλεπικοινωνιακές υπηρεσίες μέχρι και τα μέσα της δεκαετίας του 1980. Στην Ιαπωνία το 1967 η εταιρεία Nippon Telegraph and Telephone company (NTT) άρχισε να αναπτύσσει δίκτυο σε όλη την Ιαπωνία σε συχνότητες στην περιοχή των 800 MHz. Μόλις το 1969 η εταιρεία Bell Systems παρείχε δίκτυο κινητών επικοινωνιών στο ευρύ κοινό, το οποίο λειτουργούσε σε συχνότητες στα 450 MHz και με δυνατότητα πραγματοποίησης μεταπομπών (handovers) για πρώτη φορά. Το δίκτυο αυτό ήταν βασισμένο στο πρωτόκολλο IMTS. Στις 17 Οκτωβρίου του 1973 η εταιρεία Motorola παρουσίασε το πρώτο φορητό τηλέφωνο που μπορούσε να κρατηθεί με το ένα χέρι και να λειτουργήσει αυτοτελώς, Το 1978 τόσο η AT&T (όπως μετονομάστηκε η Bell South), όσο και η εταιρεία Bahrain Telephone Company, ανέπτυξαν και λειτούργησαν εμπορικά δίκτυα κινητών επικοινωνιών βασισμένα στο πρότυπο Advanced Mobile Phone Service (AMPS), που είναι ένα αναλογικό σύστημα κινητών επικοινωνιών και πρόδρομος του Digital Advanced Mobile Phone Service (D-AMPS) και το οποίο λειτουργεί μέχρι και σήμερα στις ΗΠΑ.

Στις αρχές της δεκαετίας του 1980 αρκετές Ευρωπαϊκές χώρες (Νορβηγία, Σουηδία, Φιλανδία, Βέλγιο, Ολλανδία, Μεγάλη Βρετανία, ΕΙΡΕ) είχαν αναπτύξει δίκτυα κινητών επικοινωνιών βασισμένα σε πρότυπα αναλογικών επικοινωνιών (NMT 450 και 900) τα οποία ήταν μη συμβατά μεταξύ τους. Την ίδια στιγμή οι ανάγκες για τηλεπικοινωνιακές υπηρεσίες αυξάνονταν ραγδαία. Εξαιτίας του γεγονότος αυτού, καθώς και της ανάγκης συμβατών δικτύων σε όλη την Ευρώπη, η Ευρωπαϊκή Επιτροπή Τηλεπικοινωνιών και Ταχυδρομείων (CEPT) ίδρυσε μια ομάδα εμπειρογνομόνων επιφορτισμένη με την έκδοση των προδιαγραφών ενός κοινού συστήματος κινητών επικοινωνιών για όλη τη Δυτική Ευρώπη. Η ομάδα αυτή ονομάστηκε Groupe Speciale Mobile, από τα αρχικά γράμματα της οποίας προέκυψε για πρώτη φορά το όνομα GSM. Η συντομογραφία αυτή έγινε όμως γνωστή με τον όρο Global System for Mobile Communications (GSM). Από το σημείο αυτό ξεκινά η εξέλιξη του πλέον διαδεδομένου και επιτυχημένου συστήματος κινητών επικοινωνιών, του γνωστού σε όλους GSM. Το 1982 η CEPT αρχικοποίησε το νέο σύστημα (GSM), ενώ το 1986 πραγματοποιήθηκαν οι πρώτες δοκιμές του συστήματος στο Παρίσι. Το 1987 καθορίστηκαν οι συχνότητες λειτουργίας του συστήματος στην περιοχή των 900 MHz και το 1988 δημιουργήθηκε το European Telecommunications Standards Institute(ETSI), το οποίο επωμίστηκε την ευθύνη της έκδοσης προδιαγραφών και συστάσεων για το GSM, που εξεδόθησαν για πρώτη φορά το 1989. Στις 1-7-1991 πραγματοποιήθηκε το πρώτο επίσημο εμπορικό τηλεφώνημα από δίκτυο GSM. Το 1992 οι Αυστραλοί πάροχοι κινητών επικοινωνιών έγιναν οι πρώτοι μη Ευρωπαίοι οι οποίοι εξεδήλωσαν επιθυμία να αναπτύξουν το νέο σύστημα, ενώ την ίδια χρονιά καθορίστηκαν και νέες συχνότητες στην περιοχή των 1800 MHz. Στην περιοχή αυτή συχνοτήτων αναπτύχθηκε και το σύστημα PCS στις ΗΠΑ, που αποτελεί σύστημα συμβατό (μερικώς) με το GSM. Έτσι ξεκίνησε η ανάπτυξη των δικτύων GSM, που είναι πλέον τα πιο διαδεδομένα δίκτυα διεθνώς διαθέτοντας εκατοντάδες εκατομμύρια συνδρομητές και αποτελώντας ταυτόχρονα την σταθερή βάση για την ανάπτυξη των δικτύων νέας γενιάς (3G/UMTS), των οποίων η εξάπλωση ξεκίνησε ήδη από το 2002.

Ένα δίκτυο τύπου GSM αποτελείται από τα ακόλουθα υποσυστήματα:

- Το MSC (Mobile Switching Center), ένα ψηφιακό τηλεπικοινωνιακό σύστημα το οποίο χρησιμοποιείται για την διαχείριση των κλήσεων φωνής των συνδρομητών στο GSM δίκτυο. Δίπλα σε κάθε MSC βρίσκεται και μια VLR (Visitor Location

Register) που είναι προσωρινή βάση δεδομένων και περιέχει τις υπηρεσίες του συνδρομητή, αλλά και την ευρύτερη περιοχή που βρίσκεται.

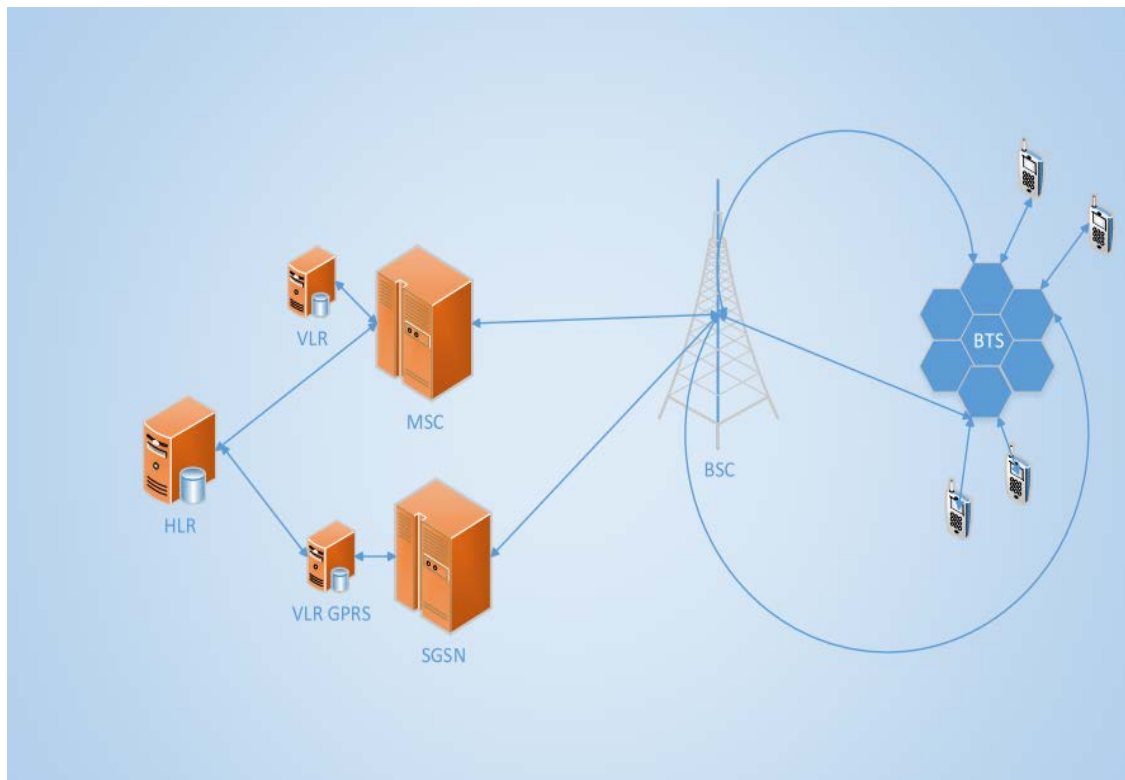
- Το SGSN (Serving GPRS Support Node) είναι ψηφιακό τηλεπικοινωνιακό σύστημα το οποίο χρησιμοποιείται για την διαχείριση των κλήσεων δεδομένων των συνδρομητών στο GSM-GPRS δίκτυο. Δίπλα σε κάθε SGSN βρίσκεται επίσης και η αντίστοιχη GPRS VLR.

- Το BSC (Base Station Controller): Είναι ένα ψηφιακό κέντρο και χρησιμοποιείται για να ελέγχει τους Σταθμούς Βάσης και την διασύνδεσή τους με το MSC.

- Το BTS (Base Transceiver Station) ή Σταθμός Βάσης (ΣΒ), χρησιμοποιείται για την διασύνδεση των κινητών τηλεφώνων με το υπόλοιπο δίκτυο. Περιέχει πομποδέκτες (συχνότητας: 900-1800 MHz). Επίσης στα BTS συγκαταλέγονται και οι μικροκυματικές κατευθυντικές ζεύξεις είτε τα μισθωμένα κυκλώματα που χρησιμοποιούνται για τη μεταφορά των δεδομένων του κάθε σταθμού βάσης στα αντίστοιχα κέντρα (BSC – MSC) και τα οποία αποτελούν το δίκτυο μετάδοσης (transmission network).

- Η HLR (Home Location Register) είναι μόνιμη βάση δεδομένων με όλες τις υπηρεσίες των συνδρομητών πχ. προωθήσεις, Data, Fax, GPRS.

Παρακάτω παρουσιάζεται η βασική δομή ενός δικτύου GSM (Εικόνα 1). (βλέπε [5])



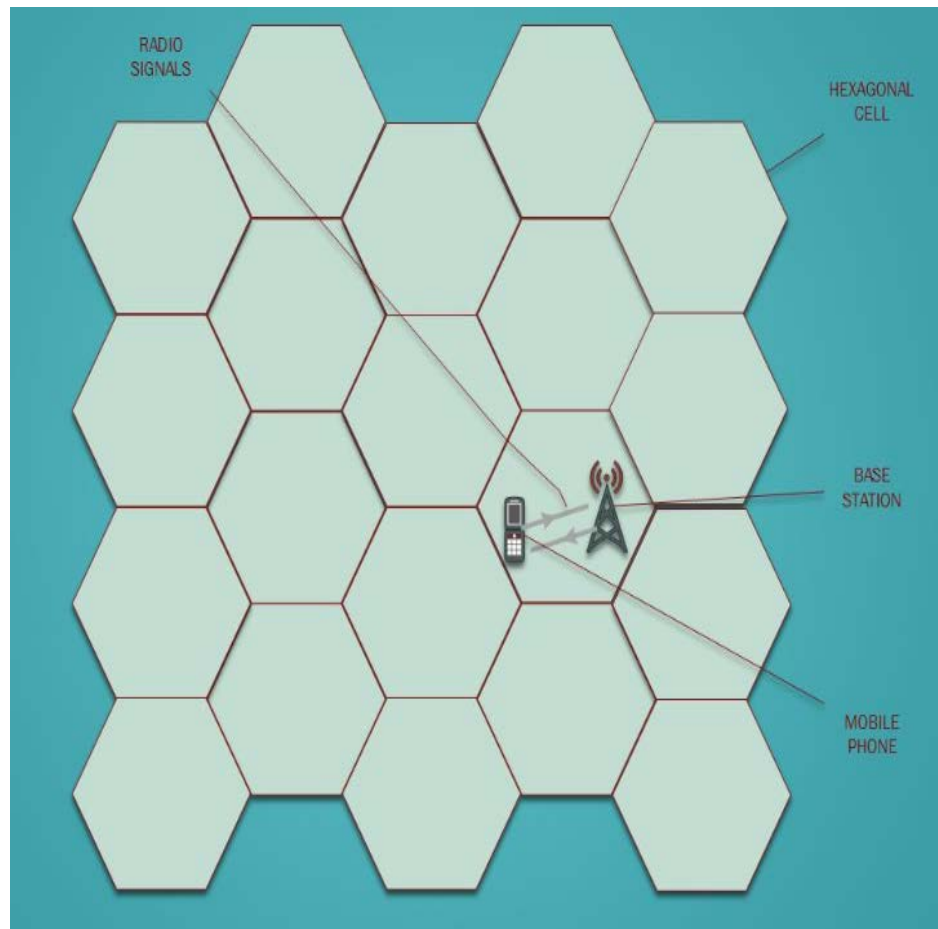
Εικόνα 12 - Βασική δομή ενός δικτύου GSM

Τα κινητά τηλέφωνα χρησιμοποιούν ραδιοκύματα, όπως και πολλές άλλες μορφές ραδιοεπικοινωνίας στο παρελθόν (π.χ. ασύρματοι στρατιωτικών εφαρμογών, ραδιοτηλέφωνα, κ.λ.π.). Σε όλη την Ευρώπη οι κυβερνήσεις έχουν χορηγήσει άδειες λειτουργίας για δίκτυα τεχνολογίας 2ης Γενιάς (GSM) στα 900 και 1800 MHz, καθώς και άδειες για δίκτυα τεχνολογίας 3ης γενιάς (UMTS) σε δεκάδες εταιρείες, ώστε να παρέχουν τηλεπικοινωνιακές υπηρεσίες στο ευρύ κοινό. Οι εταιρείες αυτές αποτελούν τους παρόχους υπηρεσιών κινητής τηλεφωνίας και ως κύριο περιοριστικό παράγοντα στην ανάπτυξη των δικτύων έχουν τον αριθμό των διαθέσιμων συχνοτήτων και καναλιών που χρησιμοποιούνται στα δίκτυα. Έτσι ο σχεδιασμός και η ανάπτυξη ενός δικτύου κινητών επικοινωνιών είναι μια εξαιρετικά περίπλοκη και δύσκολη διαδικασία. Τόσο οι ΣΒ όσο και τα κινητά τηλέφωνα λειτουργούν μετατρέποντας την φωνή, το κείμενο και τα δεδομένα σε ραδιοκύματα, τα οποία μεταδίδονται δια μέσου του αέρα όπως και σε κάθε άλλη μορφή αμφίδρομης επικοινωνίας. Οι ΣΒ λαμβάνουν τις κλήσεις – ραδιοκύματα που προέρχονται από τα κινητά τηλέφωνα και τις μεταδίδουν μέσω του δικτύου είτε σε σταθερά τηλέφωνα (ΟΤΕ), είτε σε άλλα κινητά τηλέφωνα μέσω εκπομπών των ίδιων ή άλλων ΣΒ. Για να γίνει δυνατή η παροχή υπηρεσιών

κινητής τηλεφωνίας οι πάροχοι κινητών υπηρεσιών (στην Ελλάδα είναι οι Cosmote, Vodafone, Wind) διαιρούν τις κατά τόπους γεωγραφικές περιοχές σε χιλιάδες μικρές και ανεξάρτητες περιοχές που ονομάζονται κυψέλες. Η κάθε κυψέλη από αυτές προϋποθέτει και την ύπαρξη ΣΒ στην περιοχή της.

Τα σημεία τοποθέτησης των κεραιών κινητής τηλεφωνίας είναι κρίσιμα τόσο όσον αφορά στη ραδιοκάλυψη και παροχή υπηρεσιών, όσο και σε θέματα ηλεκτρομαγνητικών εκπομπών. Κατά καιρούς έχουν διατυπωθεί διάφορες απόψεις σχετικά με τα σημεία τοποθέτησης των κεραιών. Σύμφωνα με μια άποψη υποστηρίζεται ότι οι κεραιές πρέπει να τοποθετούνται περιμετρικά πάνω σε υψώματα και μακριά από κατοικημένες περιοχές. Η άποψη αυτή είναι απολύτως λανθασμένη για δύο λόγους:

1. Οι κεραιές που βρίσκονται τοποθετημένες μακριά από κατοικημένες περιοχές για να μπορέσουν να καλύψουν μια συγκεκριμένη περιοχή, θα πρέπει να εκπέμπουν με πολύ μεγάλη ισχύ, γεγονός που είναι ανεπιθύμητο όσον αφορά στις ηλεκτρομαγνητικές εκπομπές. Τονίζουμε στο σημείο αυτό ότι οι κεραιές ραδιοσταθμών που βρίσκονται τοποθετημένες σε υψώματα περιμετρικά των πόλεων, εκπέμπουν με ισχύ εκπομπής που ξεπερνά τις μερικές χιλιάδες Watt. Αντιθέτως, οι κεραιές που βρίσκονται εντός κατοικημένων περιοχών εκπέμπουν με ισχύ εκπομπής που δεν ξεπερνά στις περισσότερες περιπτώσεις τα 20 Watt. Από τις προδιαγραφές του συστήματος GSM, ένα δίκτυο αυτού του τύπου διαθέτει κυψελοειδή εξαγωνική δομή όπως αυτή που φαίνεται στην εικόνα 2. Κάθε εξάγωνο αποτελεί και μια κυψέλη (cell), η οποία δεν μπορεί να επικαλύπτεται σημαντικά με κάποια άλλη γειτονική ή μη, του ίδιου δικτύου, διότι αυτό δημιουργεί παρεμβολή στο δίκτυο με αποτέλεσμα τη δυσλειτουργία των κινητών τηλεφώνων που βρίσκονται στην περιοχή αυτή.



Εικόνα 13 - Διάταξη δικτύου GSM

Ο λόγος που δεν μπορούν να επικαλύπτονται οι κυψέλες οφείλεται στην αδυναμία επαναχρησιμοποίησης των συχνοτήτων λειτουργίας (φερουσών συχνοτήτων) σε γειτονικές κυψέλες. Έτσι μπορεί να λεχθεί ότι σε ένα δίκτυο GSM, οι συχνότητες αποτελούν ένα «ακριβό αγαθό», σε περιορισμένη ποσότητα. Ένα κινητό τηλέφωνο επικοινωνεί μόνο με το ΣΒ που βρίσκεται στην περιοχή του, μέσω του οποίου συνδέεται στο υπόλοιπο τηλεφωνικό δίκτυο (κινητό ή σταθερό). Η μέγιστη ισχύς εκπομπής του κινητού τηλεφώνου είναι 2 Watt για κινητό τηλέφωνο που λειτουργεί στα 900MHz και 1 Watt για κινητό τηλέφωνο που λειτουργεί στα 1800MHz. Η ισχύς λειτουργίας του κινητού τηλεφώνου ρυθμίζεται από το δίκτυο και εξαρτάται από τη θέση αυτού σε σχέση με τη θέση της κεραίας. Έτσι ένα κινητό τηλέφωνο που είναι απομακρυσμένο από την κεραία με την οποία συνδέεται, εκπέμπει με πολύ μεγαλύτερη ισχύ από ένα κινητό που βρίσκεται κοντά στην κεραία – ΣΒ. Επομένως κινητό που βρίσκεται μακριά από τον ΣΒ του

εκπέμπει πολύ μεγαλύτερα ποσά ΗΜ – ακτινοβολίας στον χρήστη του, λόγω του ότι απαιτείται μεγαλύτερη ισχύς εκπομπής για να επικοινωνήσει με τον απομακρυσμένο ΣΒ. Αντιθέτως όσο πιο πυκνό είναι το δίκτυο σε ΣΒ τόσο μικρότερη ισχύς εκπομπής απαιτείται και από το ΣΒ και από το κινητό τηλέφωνο, επομένως τόσο το περιβάλλον όσο και οι χρήστες των τηλεφώνων δέχονται πολύ μικρότερα ποσά ΗΜ-ακτινοβολίας. Ενδεικτικά αναφέρουμε ότι σε σύγχρονα πυκνά δίκτυα GSM, όπως αυτά που έχουν αναπτυχθεί στην Ελλάδα, το ποσό της ΗΜ – ακτινοβολίας που εκπέμπει ένα κινητό τηλέφωνο των 900 MHz είναι 250mW και των 1800 MHz είναι 125mW.

2. Εξαιτίας του περιορισμένου αριθμού συχνοτήτων που έχει στην κατοχή της η κάθε εταιρεία πάροχος κινητών επικοινωνιών, η κάθε κυψέλη – ΣΒ του δικτύου μπορεί να εξυπηρετήσει μικρό και συγκεκριμένο αριθμό ταυτόχρονων κλήσεων από συνδρομητές, που ανέρχονται κατά μέσο όρο από 10 έως 50 ανά κυψέλη, ανάλογα με τη διαμόρφωση του ΣΒ. Έτσι για να εξυπηρετηθούν οι εκατομμύρια συνδρομητές που είναι χρήστες κινητών τηλεφώνων απαιτούνται πάρα πολλοί ΣΒ (από εκατοντάδες μέχρι και χιλιάδες ανά περιοχή), οι οποίοι εγκαθίστανται σε πυκνή διάταξη και έχουν χαμηλή ισχύ εκπομπής, έτσι ώστε να μην προκαλούν παρεμβολές στο δίκτυο.

Στην Ελλάδα ειδικά, όπου η μορφολογία του εδάφους δεν είναι ομαλή (υπάρχουν πολλά βουνά, λόφοι, κ.λ.π.) και οι πόλεις έχουν πυκνή δόμηση με κτίρια από τσιμέντο, απαιτούνται πολλοί περισσότεροι ΣΒ εντός των αστικών κατοικημένων περιοχών, όπου βρίσκεται η μεγάλη πλειοψηφία του πληθυσμού που επιζητεί να χρησιμοποιεί κινητά τηλέφωνα. Έτσι κάθε κυψέλη απαιτεί την ύπαρξη ενός ΣΒ ο οποίος παρέχει κάλυψη και χωρητικότητα σε μια μικρή περιοχή γύρω του, εκπέμποντας ακτινοβολία σχεδόν ίσης ποσότητας με αυτή ενός κινητού τηλεφώνου που λειτουργεί στη μέγιστη ισχύ του. Καθώς ένας χρήστης κινητού τηλεφώνου κινείται μέσα στο δίκτυο κινητής τηλεφωνίας, η κλήση του πρέπει να μεταφέρεται από ένα ΣΒ στον γειτονικό του, προϋπόθεση που επίσης πρέπει να λαμβάνεται υπόψη κατά το σχεδιασμό και διαστασιολόγηση του δικτύου. (βλέπε [6])

Μετά το 2001 άρχισε η ανάπτυξη δικτύων κινητής τηλεφωνίας 3ης γενιάς (UMTS – 3G), που παρέχουν νέες και βελτιωμένες υπηρεσίες με ταχύτερη πρόσβαση, στις οποίες συγκαταλέγονται η βιντεοκλήση (videocall), η πρόσβαση σε υπηρεσίες δεδομένων (384 kbps

σήμερα και μέχρι 14 Mbps με χρήση του ειδικού χαρακτηριστικού HSDPA), εφαρμογές αποθήκευσης βίντεο (video streaming), κ.λ.π. Τα δίκτυα αυτά προσφέρουν νέες δυνατότητες στους χρήστες κινητών τηλεφώνων.

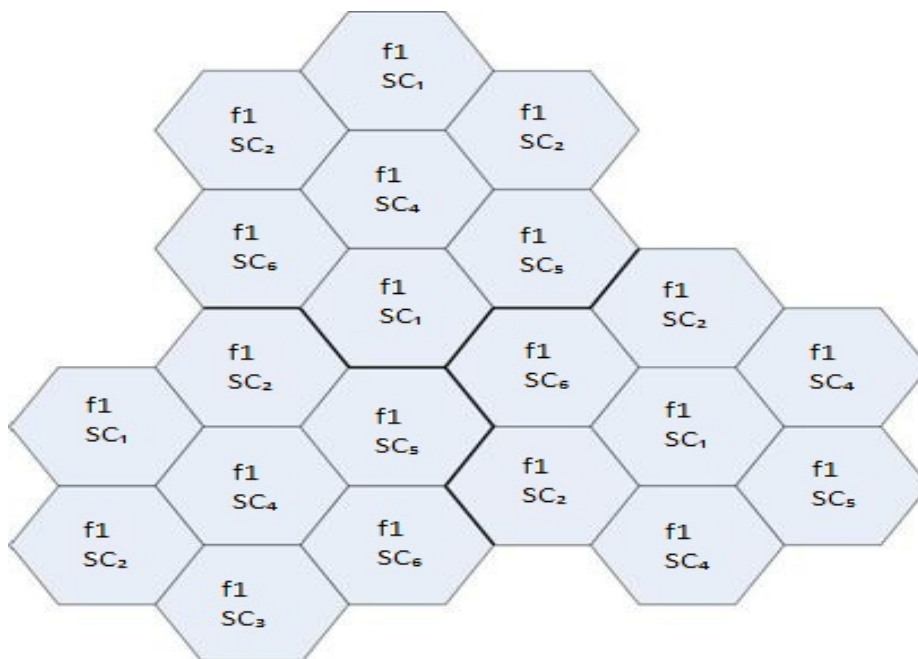
Τα δίκτυα 3ης γενιάς διαφέρουν σημαντικά στη φιλοσοφία ανάπτυξης τους συγκριτικά με τα δίκτυα 2ης γενιάς (GSM). Τα δίκτυα 3ης γενιάς βασίζονται σε τεχνολογία Wideband Code Division Multiple Access (WCDMA) και ο περιοριστικός παράγοντας παροχής υπηρεσίας στα δίκτυα αυτά είναι η στάθμη της παρεμβολής στο δίκτυο, που προέρχεται τόσο από τους χρήστες (κινητά τηλέφωνα) όσο και από τους ΣΒ που βρίσκονται στην κάθε περιοχή. Στα δίκτυα 3ης γενιάς οι γειτονικές κυψέλες χρησιμοποιούν την ίδια φέρουσα συχνότητα (carrier), με παράγοντα επαναχρησιμοποίησης ένα (1) και η διαφοροποίηση τους πραγματοποιείται με τη χρήση διαφορετικών κωδικών ανά κυψέλη, οι οποίοι ονομάζονται κώδικες scrambling (Scrambling Codes). Από την πλευρά των χρηστών στα δίκτυα 3G/UMTS ο κάθε χρήστης χρησιμοποιεί τον δικό του κωδικό και έτσι το δίκτυο τον διακρίνει από τους γειτονικούς χρήστες. Η συγκεκριμένη διάταξη παρουσιάζεται στην εικόνα 2. Η ισχύς εκπομπής του κάθε σταθμού βάσης είναι σημαντικά χαμηλότερη από του αντίστοιχου GSM και είναι της τάξης του 1 Watt για το κανάλι πιλότο (Pilot Channel) το οποίο εκπέμπει συνεχώς, ενώ η μέση ισχύ ενός ΣΒ 3G βρίσκεται στα 3 Watt. Το δε κινητό τηλέφωνο 3ης γενιάς έχει μέγιστη ισχύ εκπομπής 125 mW, το οποίο σημαίνει ότι εκπέμπει (λειτουργώντας με τη μέγιστη ισχύ του) οκτώ φορές λιγότερη ακτινοβολία από το αντίστοιχο κινητό τηλέφωνο των 1800 MHz και 16 φορές λιγότερη ακτινοβολία από το αντίστοιχο κινητό τηλέφωνο των 900 MHz.

Τα δίκτυα 3G αποτελούνται από τα παρακάτω υποσυστήματα:

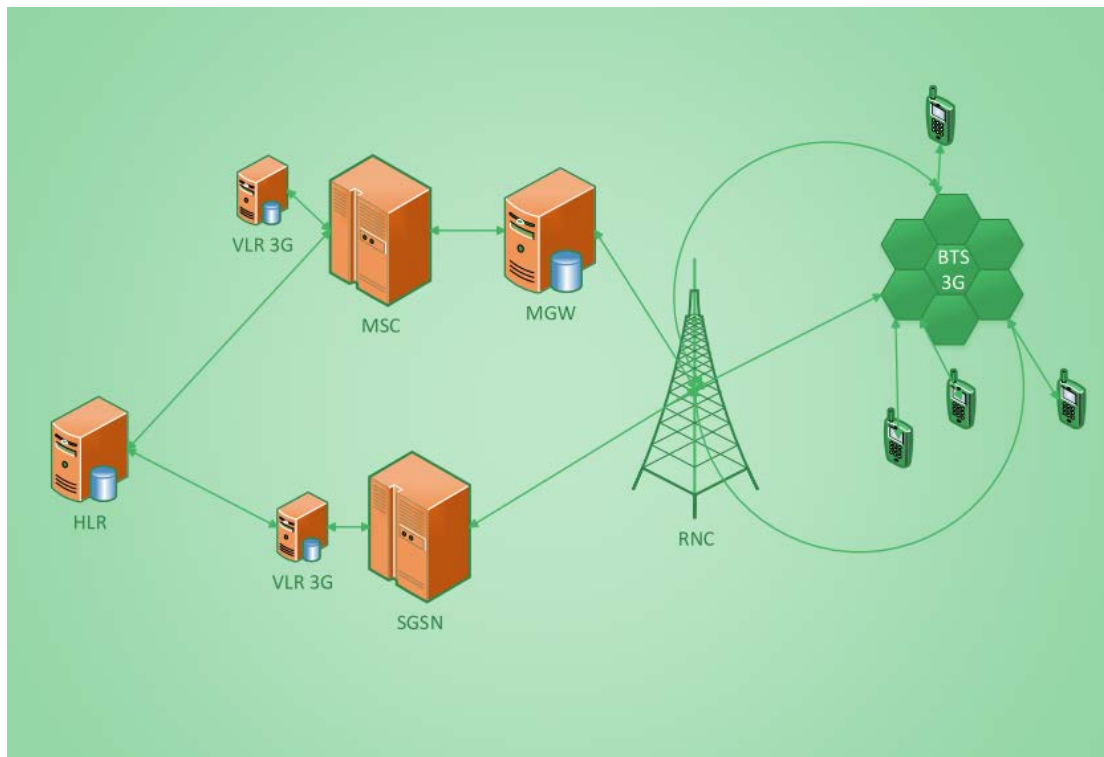
- Το MSC (Mobile Switching Controller), είναι το ίδιο σύστημα με αυτό του δικτύου GSM και επιτελεί τις ίδιες λειτουργίες. Η διαφορά είναι ότι εδώ προστίθεται δίπλα στο MSC η VLR 3G (Visitor Location Register 3G), βάση δεδομένων που χρησιμοποιεί η τεχνολογία 3G για τις υπηρεσίες του συνδρομητή.
- Το SGSN (Serving GPRS Support Node), όπως και το MSC έτσι και το SGSN χρησιμοποιείται και από το δίκτυο 3G για τον ίδιο ακριβώς λόγο που το χρησιμοποιεί το δίκτυο GSM. Δίπλα στο SGSN υπάρχει και πάλι μια βάση δεδομένων 3G VLR.

- Το MGW (Media Gateway). Είναι ουσιαστικά μια συσκευή που έχει το ρόλο μιας πύλης και μετατρέπει τη μορφή των δεδομένων ενός δικτύου σε μια άλλη μορφή, για την χρησιμοποίησή τους από ένα άλλο δίκτυο. Στην περίπτωση μας κάνει εφικτή την επικοινωνία ανάμεσα στο MSC και το RNC (αναλύεται παρακάτω).
- Το RNC (Radio Network Controller), το οποίο είναι υπεύθυνο για την διαχείριση των BTS 3G (αναλύονται παρακάτω) και για την επικοινωνία τους με το MGW.
- Το BTS 3G (Base Transceiver Station) ή Σταθμός Βάσης (ΣΒ). Επιτελεί την ίδια ακριβώς λειτουργία όπως και στο δίκτυο GSM με τη μόνη διαφορά ότι χρησιμοποιεί μεγαλύτερες συχνότητες της τάξεως των 2100 MHz.
- Την HLR (Home Location Register), την ίδια μόνιμη βάση δεδομένων με το δίκτυο GSM.

Στην εικόνα 4 παρουσιάζεται πλήρως η αρχιτεκτονική ενός δικτύου 3G.



Εικόνα 14 - Διάταξη κυψελοειδούς μορφής ενός δικτύου 3G



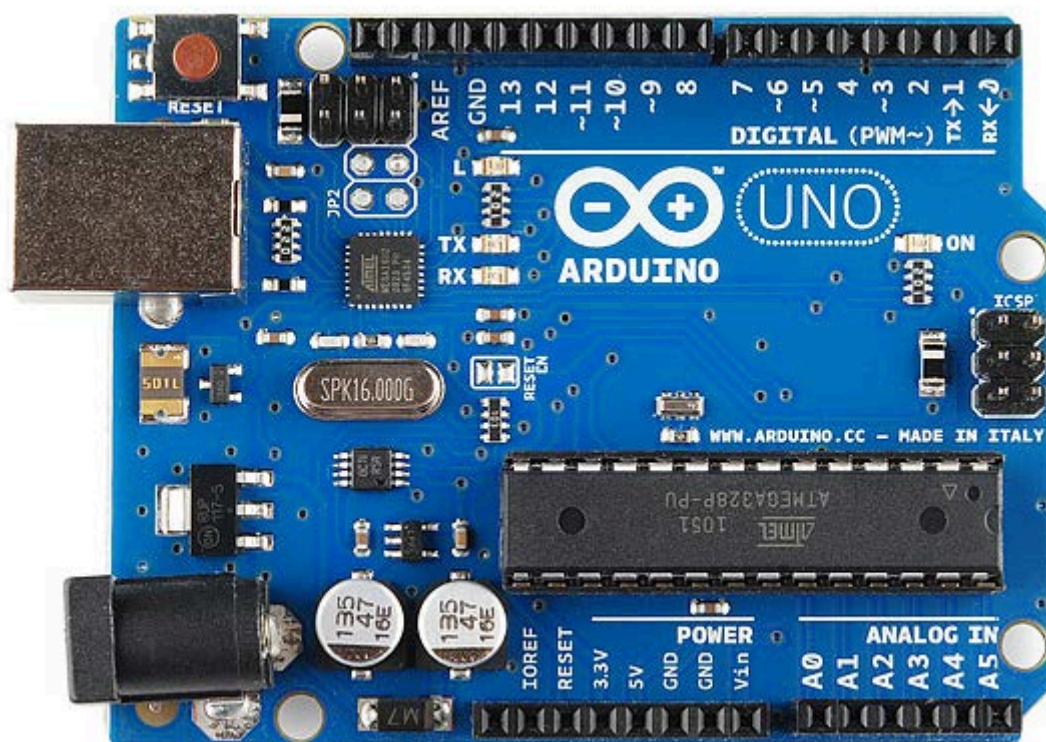
Εικόνα 15 - Βασική δομή του δικτύου 3G

Τα δίκτυα 3ης γενιάς χρειάζονται σημαντικά περισσότερους ΣΒ για να επιτύχουν κάλυψη αντίστοιχη με αυτή του δικτύου GSM που έχει ήδη αναπτυχθεί στην ίδια περιοχή. Αυτό οφείλεται στο ότι το δίκτυο 3ης γενιάς λειτουργεί σε υψηλότερες συχνότητες από τα δίκτυα GSM (2100 MHz συγκριτικά με τα 900 / 1800 MHz του GSM) και έτσι τα ραδιοκύματα που εκπέμπει καλύπτουν συγκριτικά μικρότερες αποστάσεις στον χώρο και έχουν μικρότερο βάθος διείσδυσης τόσο στα κτίρια, αλλά και στα έμβια όντα. Επίσης η κάλυψη του δικτύου ελαττώνεται καθώς η κίνηση αυξάνει, γεγονός που δυσχεραίνει περαιτέρω τον αναλυτικό σχεδιασμό ενός τέτοιου δικτύου. Τέλος, θα πρέπει να τονισθεί ότι οι ΣΒ του δικτύου 3ης γενιάς ελέγχουν διαρκώς την ισχύ εκπομπής του κάθε κινητού τηλεφώνου, ωθώντας το να λειτουργεί στη χαμηλότερη δυνατή ισχύ εκπομπής που απαιτείται για την παροχή συγκεκριμένης υπηρεσίας. Με τον τρόπο αυτό ελέγχεται και ελαχιστοποιείται η παρεμβολή που προκαλούν τα κινητά στο δίκτυο, ενώ ταυτόχρονα αυξάνεται η διάρκεια της μπαταρίας του τηλεφώνου (αφού η κατανάλωση είναι σημαντικά χαμηλότερη). (βλέπε [7])

3 ΠΑΡΟΥΣΙΑΣΗ ΤΗΣ ΠΛΑΤΦΟΡΜΑΣ ARDUINO

3.1 ΣΥΝΤΟΜΗ ΠΕΡΙΓΡΑΦΗ ΤΟΥ ARDUINO

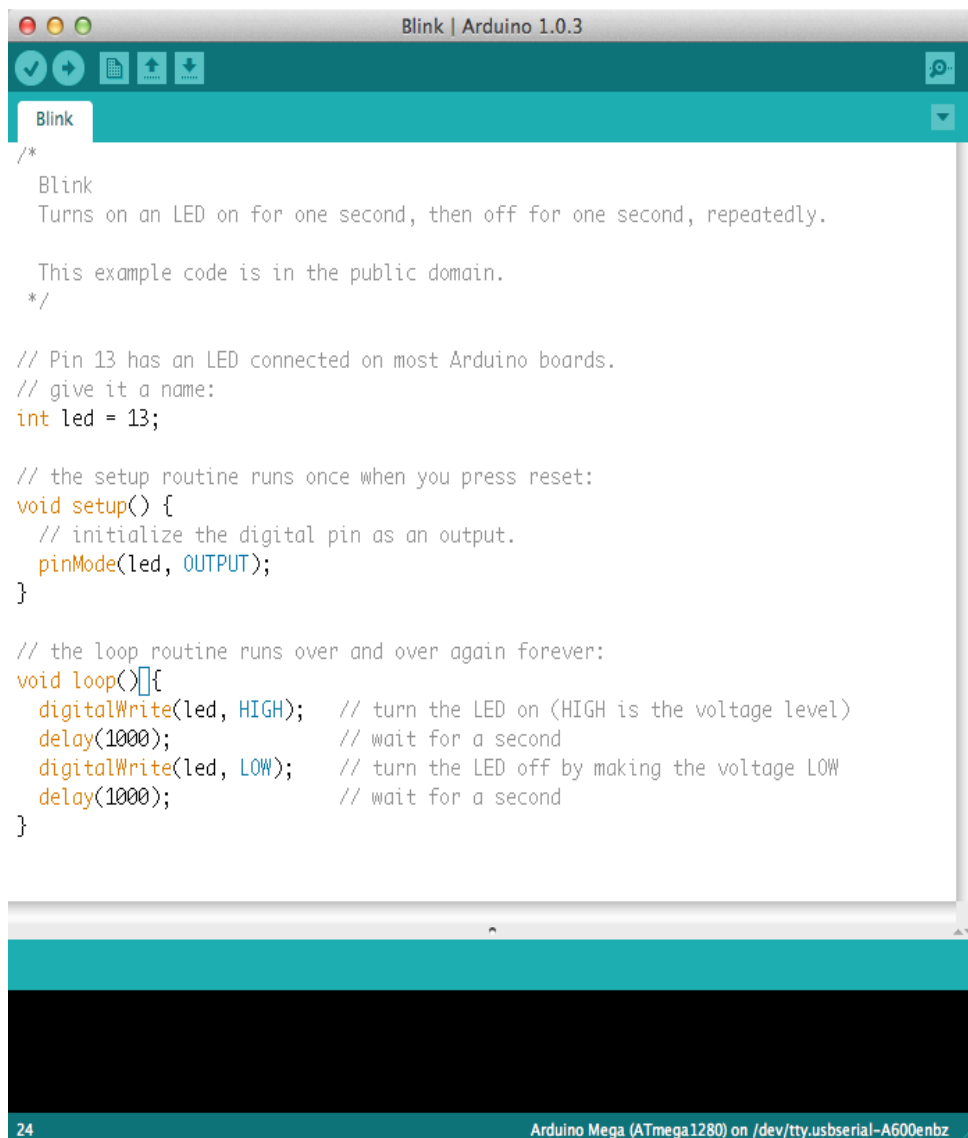
Το Arduino είναι μια πλατφόρμα ανοιχτού κώδικα που χρησιμοποιείται για την ανάπτυξη ηλεκτρονικών εφαρμογών. Αποτελείται από το υλικό (hardware), το οποίο είναι μία ηλεκτρονική πλακέτα με προγραμματιζόμενο μικροελεγκτή (microcontroller). Το άλλο κύριο μέρος από το οποίο αποτελείται είναι το κομμάτι του λογισμικού (software) ή IDE(Ολοκληρωμένο Περιβάλλον Ανάπτυξης), το οποίο τρέχει στον Η/Υ και χρησιμοποιείται για να γράψουμε και να στείλουμε τον κώδικα μας στην πλακέτα του Arduino.



Εικόνα 16 - Arduino Uno(R3)

Η πλατφόρμα του Arduino έχει γίνει αρκετά δημοφιλής στους ανθρώπους που κάνουν τα πρώτα τους βήματα στα ηλεκτρονικά κυκλώματα και τον προγραμματισμό τους για τους παρακάτω λόγους. Αντίθετα με προηγούμενες υλοποιήσεις προγραμματιζόμενων

ηλεκτρονικών κυκλωμάτων, το Arduino δεν απαιτεί ξεχωριστό κομμάτι υλικού για την φόρτωση του κώδικα στην πλακέτα αλλά χρησιμοποιώντας μόνο ένα καλώδιο σειριακής επικοινωνίας με τον Η/Υ (USB). Επιπλέον το IDE της πλατφόρμας του Arduino χρησιμοποιεί μια απλοποιημένη έκδοση της C++, βασισμένη στην γλώσσα προγραμματισμού Wiring, πράγμα το οποίο διευκολύνει την εκμάθηση του προγραμματισμού της πλακέτας του Arduino. Τέλος το Arduino προσφέρει τις δυνατότητες ενός μικροελεγκτή σε ένα ολοκληρωμένο και πολύ προσιτό πακέτο με πάρα πολλές δυνατότητες επέκτασης. (βλέπε [12])



```
/*
 * Blink
 * Turns on an LED on for one second, then off for one second, repeatedly.
 *
 * This example code is in the public domain.
 */
// Pin 13 has an LED connected on most Arduino boards.
// give it a name:
int led = 13;

// the setup routine runs once when you press reset:
void setup() {
  // initialize the digital pin as an output.
  pinMode(led, OUTPUT);
}

// the loop routine runs over and over again forever:
void loop(){
  digitalWrite(led, HIGH); // turn the LED on (HIGH is the voltage level)
  delay(1000); // wait for a second
  digitalWrite(led, LOW); // turn the LED off by making the voltage LOW
  delay(1000); // wait for a second
}
```

Εικόνα 17 - Arduino IDE

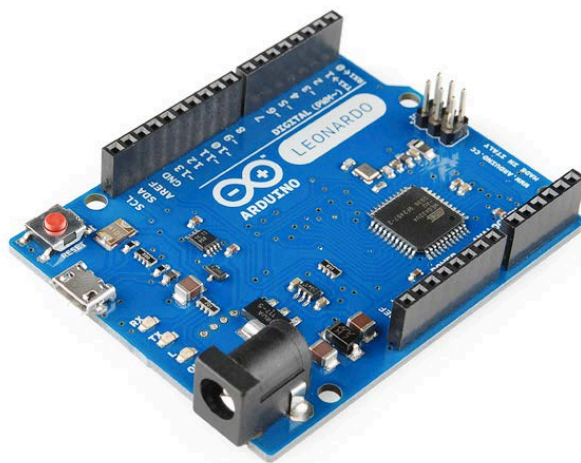
3.2 ΔΙΑΦΟΡΕΣ ΥΛΟΠΟΙΗΣΕΙΣ ΤΟΥ ARDUINO

Η κατασκευάστρια εταιρεία του Arduino προσφέρει πολλές διαφορετικές υλοποιήσεις της πλατφόρμας ανάλογα με τις διαφορετικές απαιτήσεις των χρηστών. Οι διάφορες πλακέτες Arduino χρησιμοποιούν διάφορους τύπους μικροελεγκτών AVR της ATMEL, διαφέρουν κυρίως ως προς τον αριθμό των αναλογικών/ψηφιακών εξόδων και εισόδων, τις απαιτήσεις τροφοδοσίας τους, καθώς επίσης και ως προς το μέγεθός τους. Ακολουθεί πίνακας με σύγκριση των βασικών χαρακτηριστικών των διαφορετικών υλοποιήσεων του Arduino. (βλέπε [13])

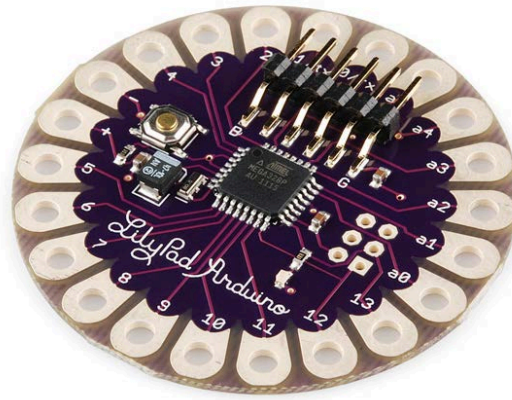
Πίνακας 4.1 - Πίνακας σύντομης σύγκρισης μεταξύ των διαφόρων πλακετών Arduino

Name	Processor	Operating Voltage/Input Voltage	CPU Speed	Analog In/Out	Digital IO/PWM	EEPROM [KB]	SRAM [KB]	Flash [KB]	USB	UART
Uno	ATmega328	5 V/7-12 V	16 Mhz	6/0	14/6	1	2	32	Regular	1
Due	AT91SAM3X8E	3.3 V/7-12 V	84 Mhz	12/2	54/12	-	96	512	2 Micro	4
Leonardo	ATmega32u4	5 V/7-12 V	16 Mhz	12/0	20/7	1	2.5	32	Micro	1
Mega 2560	ATmega2560	5 V/7-12 V	16 Mhz	16/0	54/15	4	8	256	Regular	4
Mega ADK	ATmega2560	5 V/7-12 V	16 Mhz	16/0	54/15	4	8	256	Regular	4
Micro	ATmega32u4	5 V/7-12 V	16 Mhz	12/0	20/7	1	2.5	32	Micro	1
Mini	ATmega328	5 V/7-9 V	16 Mhz	8/0	14/6	1	2	32	-	-
Nano	ATmega168 ATmega328	5 V/7-9 V	16 Mhz	8/0	14/6	0.512 1	1 2	16 32	Mini-B	1
Ethernet	ATmega328	5 V/7-12 V	16 Mhz	6/0	14/4	1	2	32	Regular	-
Esplora	ATmega32u4	5 V/7-12 V	16 Mhz	-	-	1	2.5	32	Micro	-
ArduinoBT	ATmega328	5 V/2.5-12 V	16 Mhz	6/0	14/6	1	2	32	-	1

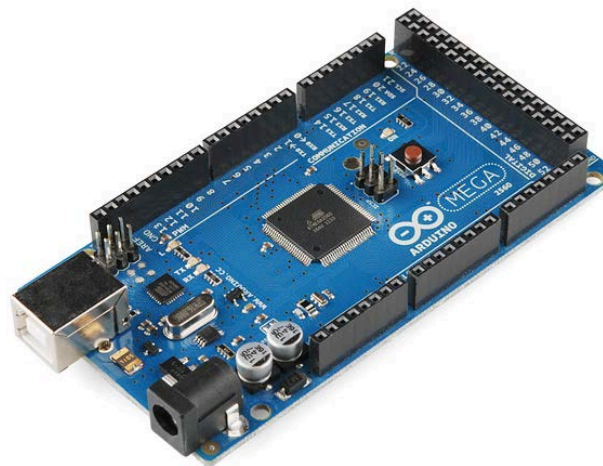
Fio	ATmega328P	3.3 V/3.7-7 V	8 Mhz	8/0	14/6	1	2	32	Mini	1
Pro (168)	ATmega168	3.3 V/3.35-12 V	8 Mhz	6/0	14/6	0.51	1	16	-	1
Pro (328)	ATmega328	5 V/5-12 V	16 Mhz	6/0	14/6	1	2	32	-	1
Pro Mini	ATmega168	3.3 V/3.35-12V 5 V/5-12 V	8 Mhz 16 Mhz	6/0	14/6	0.512	1	16	-	1
LilyPad	ATmega168V ATmega328V	2.7-5.5 V/2.7- 5.5 V	8 Mhz	6/0	14/6	0.512	1	16	-	-
LilyPad USB	ATmega32u4	3.3 V/3.8-5V	8 Mhz	4/0	9/4	1	2.5	32	Micro	-



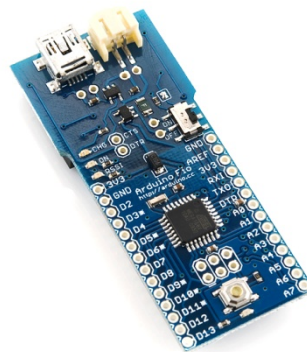
Εικόνα 18 - Arduino Leonardo



Εικόνα 19 - Arduino Lilypad



Εικόνα 20 - Arduino Mega 2560



Εικόνα 21 - Arduino Fio

3.3 ΤΟ ΥΛΙΚΟ (HARDWARE) ΤΟΥ ARDUINO

Στην εργασία αυτή κάνουμε χρήση του Arduino Leonardo, το οποίο αποτελεί μία από τις τελευταίες εκδόσεις της πλατφόρμας του Arduino. Το Leonardo βασίζεται στον μικροελεγκτή ATmega32u4 της ATMEL. Διαθέτει 37 ακροδέκτες, εκ των οποίων 20 μπορούν να λειτουργήσουν ως ψηφιακές έξοδοι, 7 ως έξοδοι PWM (Pulse Width Modulation), 12 ως αναλογικοί είσοδοι και οι υπόλοιποι ακροδέκτες είναι για τροφοδοσία και άλλες ειδικές λειτουργίες που θα αναλυθούν στη συνέχεια. Η κύρια διαφορά του Leonardo από άλλες εκδόσεις Arduino είναι ότι ο μικροελεγκτής (ATmega32u4) που χρησιμοποιεί ενσωματώνει το πρωτόκολλο επικοινωνίας USB, εξαλείφοντας την ανάγκη για τοποθέτηση δευτερεύοντος ολοκληρωμένου κυκλώματος για αυτήν την δυνατότητα. Αυτό το χαρακτηριστικό επιτρέπει στο Leonardo να αναγνωρίζεται από τον Η/Υ ως συνδεδεμένο ποντίκι ή πληκτρολόγιο πέρα από μια απλή εικονική σειριακή θύρα(CDC/COM), χαρίζοντάς του επιπρόσθετες λειτουργίες για πλήθος εφαρμογών. Παρακάτω ακολουθεί παρουσίαση της συγκεκριμένης πλακέτας και ανάλυση των διαφόρων χαρακτηριστικών της. (βλέπε [14])



Εικόνα 22 - Atmel ATmega32u4

3.4 Η ΤΡΟΦΟΔΟΣΙΑ ΤΗΣ ΠΛΑΚΕΤΑΣ ΤΟΥ ARDUINO

Κάθε πλακέτα του Arduino απαιτεί εξωτερική πηγή τροφοδοσίας. Στις ευρέως διαδεδομένες εκδόσεις των Duemillanove/UNO/Leonardo υπάρχουν δύο τρόποι για την τροφοδοσία της πλακέτας. Οι παραπάνω γνωστές πλακέτες Arduino μπορούν να τροφοδοτηθούν είτε από την θύρα USB που διαθέτουν, είτε από υποδοχή σύνδεσης με εξωτερικό τροφοδοτικό τύπου 2.1mm center-positive DC power connector jack. Η πηγή της τροφοδοσίας επιλέγεται αυτόματα από την πλακέτα του Arduino. Επίσης υπάρχουν είσοδοι με την ένδειξη “Vin” για την τάση τροφοδοσίας και “GND” για την γείωση που διατίθενται για την απευθείας σύνδεση τροφοδοσίας με καλώδιο χωρίς την ανάγκη χρησιμοποίησης ειδικού ακροδέκτη.

Η τάση τροφοδοσίας της πλακέτας μπορεί να είναι από 6 έως 20 Volts. Ωστόσο όταν τροφοδοτείται με λιγότερα από 7V η έξοδος των 5V μπορεί να παρέχει λιγότερα από 5V και το κύκλωμα μπορεί να μην είναι σταθερό. Ενώ αν τροφοδοτείται με περισσότερα από 12V ο σταθεροποιητής τάσης μπορεί να υπερθερμανθεί και να προκαλέσει βλάβη στην πλακέτα. Επομένως η προτεινόμενη τάση τροφοδοσίας είναι από 7 έως 12V για την σωστή λειτουργία της πλακέτας. (βλέπε [15])



Εικόνα 23 - Είσοδοι / Έξοδοι τροφοδοσίας Arduino Leonardo

3.5 Η ΜΝΗΜΗ ΤΟΥ LEONARDO

Η πλακέτα του Leonardo χρησιμοποιεί τον μικροελεγκτή της ATMEL ATmega32u4 όπως έχουμε ήδη αναφέρει. Ο συγκεκριμένος μικροελεγκτής διαθέτει τρεις κατηγορίες μνήμης οι οποίες είναι απαραίτητες για την λειτουργία του.

Ενσωματώνει flash memory μεγέθους 32 KB, εκ των οποίων τα 4 KB είναι δεσμευμένα από τον bootloader του Leonardo και χρησιμοποιούνται για την αποθήκευση του firmware του. Το firmware χρησιμεύει για την φόρτωση του προγράμματος στον μικροελεγκτή διαμέσου του Arduino IDE κάνοντας χρήση της USB θύρας. Τα προγράμματα που αναπτύσσονται στο Arduino IDE αφού πρώτα μεταγλωττιστούν στον Η/Υ αποστέλλονται στην πλακέτα του Arduino όπου και αποθηκεύονται στην ελεύθερη μνήμη flash των 28 KB. Κύριο χαρακτηριστικό της μνήμης flash είναι ότι δεν χάνει τα περιεχόμενά της μετά από απώλεια τροφοδοσίας ή reset της πλακέτας.

Το δεύτερο είδος μνήμης που διαθέτει ο ATmega32u4 είναι η SRAM μεγέθους 2.5 KB. Η μνήμη αυτή χρησιμοποιείται από το πρόγραμμα για να αποθηκεύονται οι τιμές των μεταβλητών, πινάκων και άλλων δεδομένων. Η μνήμη αυτή είναι προσωρινή και χάνει τα δεδομένα της όταν διακοπεί η τροφοδοσία της πλακέτας καθώς επίσης και με reset.

Τέλος ο συγκεκριμένος μικροελεγκτής έχει 1 KB μνήμη τύπου EEPROM. Η συγκεκριμένη μνήμη μπορεί να χρησιμοποιηθεί για εγγραφή και ανάγνωση δεδομένων από το πρόγραμμα που έχει φορτωθεί στον ATmega32u4. Αντίθετα με την SRAM η EEPROM δεν χάνει τα περιεχόμενά της σε απώλεια τροφοδοσίας ή επανεκκίνησης. (βλέπε [15])

3.6 ΟΙ ΑΚΡΟΔΕΚΤΕΣ ΤΟΥ ARDUINO LEONARDO

Η επικοινωνία της πλατφόρμας του Arduino Leonardo με αισθητήρια, κινητήρες, led, κλπ. γίνεται μέσω των ακροδεκτών που διαθέτει. Οι ακροδέκτες του Leonardo έχουν πολλές διαφορετικές λειτουργίες οι οποίες παρουσιάζονται στην συνέχεια. Το πλήθος των ακροδεκτών αυτών είναι 37.

Οι 20 ακροδέκτες ψηφιακής εισόδου/εξόδου μπορούν να χρησιμοποιηθούν ως ψηφιακές εισοδοί ή έξοδοι και κάποιοι ως αναλογικές εισοδοί ή έξοδοι(PWM). Μερικοί ακροδέκτες υποστηρίζουν πολλαπλές λειτουργίες. Όλοι οι ακροδέκτες λειτουργούν με τάση 5V και κάθε ακροδέκτης μέγιστη ένταση ρεύματος ίση με 40mA. (βλέπε [15])

Οι 12 ακροδέκτες (A0, A1, A2, A3, A4, A5, 4, 6, 8, 9, 10, 12) από αυτούς μπορούν να χρησιμοποιηθούν ως αναλογικές εισοδοί για την χρήση σημάτων από αναλογικά αισθητήρες. Οι ακροδέκτες αυτοί μπορούν να λειτουργήσουν σε συνδυασμό με την τάση εξόδου του ακροδέκτη AREF ο οποίος μας δίνει μια τάση αναφοράς η οποία μπορεί να ρυθμιστεί από 2V έως 5V και να χρησιμοποιηθεί ως τάση αναφοράς για τις αναλογικές εισόδους του Leonardo.

Οι 7 ακροδέκτες (3, 5, 6, 9, 10, 11, 13) υποστηρίζουν δυνατότητα ψευδοαναλογικής εξόδου με χρήση του συστήματος PWM(Pulse Width Modulation) 8 bit.

Όλοι οι ακροδέκτες (0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13) υποστηρίζουν λειτουργία ψηφιακής εισόδου και εξόδου καθώς επίσης και οι ακροδέκτες των αναλογικών εισόδων (A0, A1, A2, A3, A4, A5) μπορούν να λειτουργήσουν επίσης ως ψηφιακές εισοδοί ή έξοδοι.

Επίσης οι ακροδέκτες (0, 1) υποστηρίζουν σειριακή επικοινωνία. Ο ακροδέκτης 0 χρησιμοποιείται για την λήψη (RX) δεδομένων και ο ακροδέκτης 1 για την αποστολή (TX) δεδομένων της σειριακής επικοινωνίας. Όταν ενεργοποιείται η σειριακή επικοινωνία από το πρόγραμμα εκτός από τους ακροδέκτες (0, 1) τα δεδομένα αποστέλλονται και στην micro USB θύρα της πλακέτας. Σημαντικό είναι να αναφέρουμε ότι όταν ενεργοποιείται η σειριακή επικοινωνία στους ακροδέκτες (0, 1) χάνει η πλατφόρμα την δυνατότητα να τους χρησιμοποιήσει για ψηφιακές εισόδους ή εξόδους.

Οι ακροδέκτες (2, 3, SCL, SDA) μπορούν να υποστηρίξουν επικοινωνία TWI κάνοντας χρήση της βιβλιοθήκης Wire.

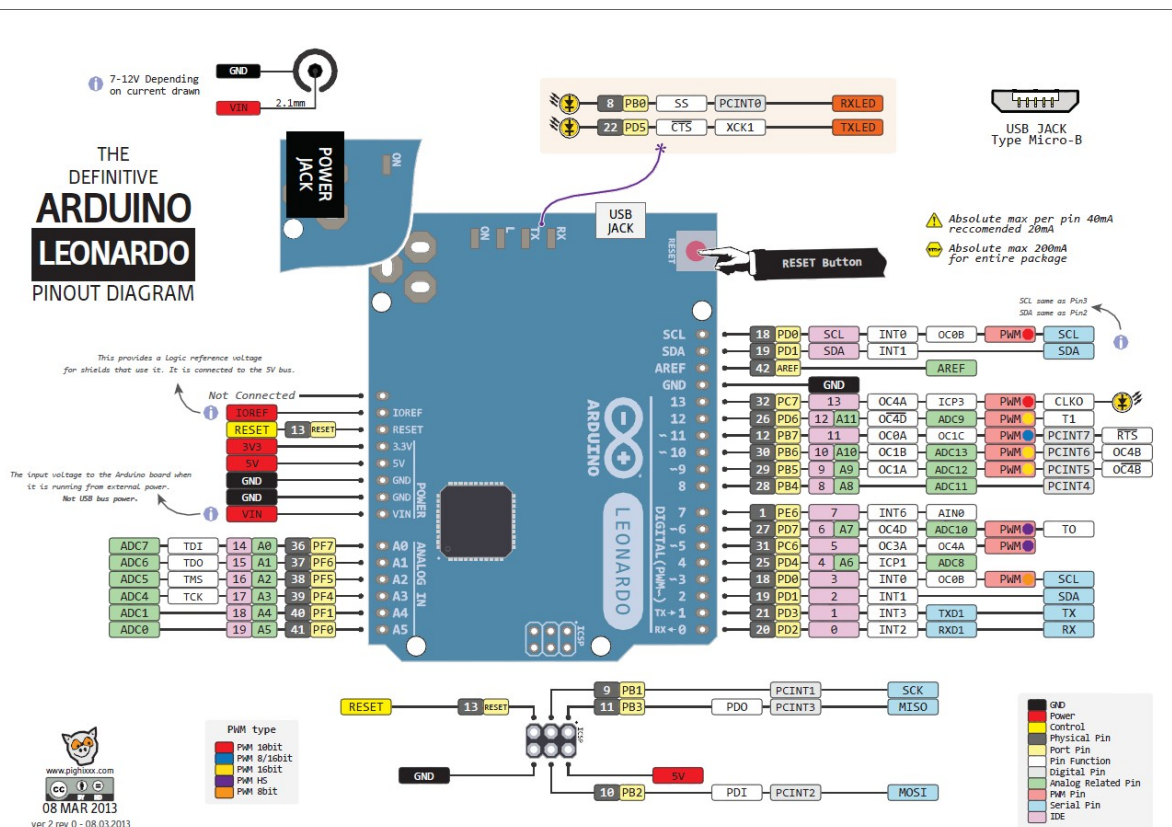
Ακόμη οι 6 ακροδέκτες που βρίσκονται στον κονέκτορα ICSP χρησιμοποιούνται αποκλειστικά για επικοινωνία SPI μέσω της βιβλιοθήκης SPI. Να σημειωθεί ότι οι ακροδέκτες του ICSP δεν είναι συνδεδεμένοι με άλλους ακροδέκτες ψηφιακών εισόδων/εξόδων της πλατφόρμας του Leonardo.

Πολύ σημαντικό είναι ότι 4 ακροδέκτες (3, 2, 0, 1, 7) υποστηρίζουν αντίστοιχα τις 5 εξωτερικές διακοπές (interrupt 0, interrupt 1, interrupt 2, interrupt 3, interrupt 4) που μπορεί

να δεχτεί ο μικροελεγκτής ATmega32u4 και να τροποποιήσει την ροή του προγράμματος που εκτελείται ανάλογα με τις συναρτήσεις που συνδέονται σε κάθε εξωτερική διακοπή.

Επιπλέον υπάρχει ο ακροδέκτης RESET ο οποίος όταν του δίνουμε LOW κάνει επανεκκίνηση τον μικροελεγκτή. Συνήθως χρησιμοποιείται για να μεταφέρει το button RESET σε shield που τοποθετείται πάνω στο Arduino και εμποδίζεται το πάτημα του.

Στην παρακάτω εικόνα φαίνονται αναλυτικά όλες οι λειτουργίες των ακροδεκτών, των υποδοχών και των LEDS του Arduino Leonardo. (βλέπε [16])

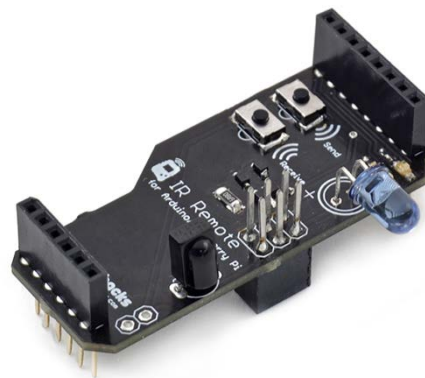


Εικόνα 24 - Αναλυτική παρουσίαση των ακροδεκτών του Leonardo

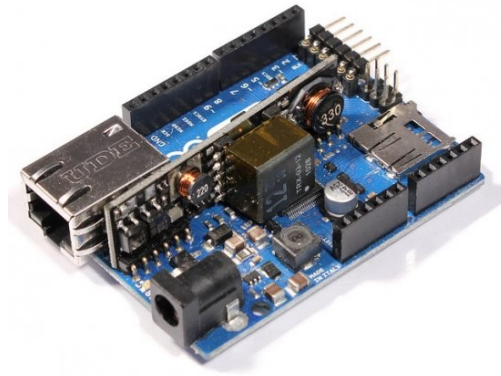
3.7 ARDUINO SHIELDS

Η έννοια του shield (ασπίδα) στην αρχιτεκτονική του Arduino, είναι η ενσωμάτωση επιπλέον υλικού (hardware) στον μικροελεγκτή που του προσδίδει μια νέα ιδιότητα, η οποία συνήθως αφορά σε θέματα επικοινωνίας, πολυμέσων, αποθηκευτικών μέσων. Με την προσαρμογή του επιπρόσθετου αυτού υλικού η εκάστοτε πλατφόρμα Arduino μέσω της σειριακής επικοινωνίας συνήθως κάνει χρήση της τεχνολογίας που διαθέτει το κάθε shield. Στην συγκεκριμένη πτυχιακή εργασία έγινε χρήση ενός shield το οποίο διαθέτει διεπαφή 3G/GPRS και δέκτη GPS καθώς επίσης και τεχνολογίες πολυμέσων όπως κάμερα, ηχείο και μικρόφωνο.

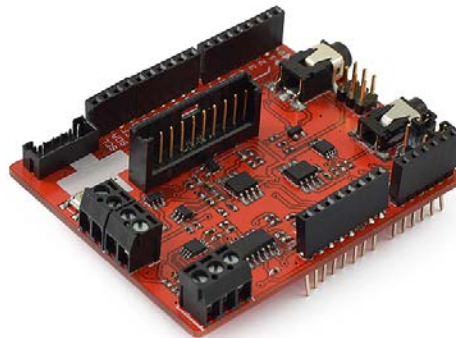
Στις παρακάτω εικόνες παρουσιάζονται ενδεικτικά shields με διαφορετικές λειτουργίες το καθένα. (βλέπε [17])



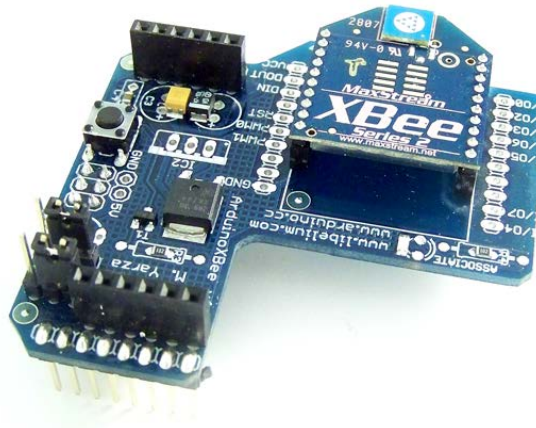
Εικόνα 25 - Shield για τηλεχειρισμό με υπέρυθη επικοινωνία



Εικόνα 26 - Shield για επικοινωνία Ethernet



Εικόνα 27 - Shield για σύνδεση ειδικών αισθητήρων για χρήση σε ιατρικές εφαρμογές



Εικόνα 28 - Shield για επικοινωνία πρωτοκόλλου Xbee

3.8 ΤΟ ΠΕΡΙΒΑΛΛΟΝ ΑΝΑΠΤΥΞΗΣ ΛΟΓΙΣΜΙΚΟΥ ΤΟΥ ARDUINO

Το ολοκληρωμένο περιβάλλον ανάπτυξης (IDE) του Arduino είναι μία εφαρμογή γραμμένη σε Java, που λειτουργεί σε πολλές πλατφόρμες, και προέρχεται από το IDE για τη γλώσσα προγραμματισμού Processing και το σχέδιο Wiring. Έχει σχεδιαστεί για να εισαγάγει τον προγραμματισμό στους καλλιτέχνες και τους νέους που δεν είναι εξοικειωμένοι με την ανάπτυξη λογισμικού. Περιλαμβάνει ένα πρόγραμμα επεξεργασίας κώδικα με χαρακτηριστικά όπως είναι η επισήμανση σύνταξης και ο συνδυασμός αγκύλων και είναι επίσης σε θέση να μεταγλωττίζει και να φορτώνει προγράμματα στην πλακέτα με ένα μόνο κλικ. Δεν υπάρχει συνήθως καμία ανάγκη να επεξεργαστείτε αρχεία make ή να τρέξετε προγράμματα σε ένα περιβάλλον γραμμής εντολών. Ένα πρόγραμμα ή κώδικας που γράφτηκε για Arduino ονομάζεται σκίτσο (sketch).

Τα Arduino προγράμματα είναι γραμμένα σε C ή C++. Το Arduino IDE έρχεται με μια βιβλιοθήκη λογισμικού που ονομάζεται "Wiring" από το πρωτότυπο σχέδιο Wiring γεγονός που καθιστά πολλές κοινές λειτουργίες εισόδου/εξόδου πολύ πιο εύκολες. Οι χρήστες πρέπει μόνο να ορίσουν δύο λειτουργίες για να κάνουν ένα πρόγραμμα κυκλικής εκτέλεσης:



```
void setup() {  
  // put your setup code here, to run once:  
  Κώδικας που εκτελείται μία φορά κατά την εκκίνηση της συσκευής  
}  
  
void loop() {  
  // put your main code here, to run repeatedly:  
  Κώδικας που εκτελείται συνεχώς  
}
```

Εικόνα 29 - Βασική Δομή Κώδικα Προγράμματος Arduino

-setup():μία συνάρτηση που τρέχει μία φορά στην αρχή του προγράμματος η οποία αρχικοποιεί τις ρυθμίσεις

-loop():μία συνάρτηση η οποία καλείται συνέχεια μέχρι η πλακέτα να απενεργοποιηθεί

Ένα τυπικό πρώτο πρόγραμμα για έναν μικροελεγκτή αναβοσβήνει απλά ένα LED.

Στο περιβάλλον του Arduino, ο χρήστης μπορεί να γράψει ένα πρόγραμμα σαν αυτό:

```
#define LED_PIN 13  
  
void setup () {  
  pinMode (LED_PIN, OUTPUT); // enable pin 13 for digital output  
}  
  
void loop () {  
  digitalWrite (LED_PIN, HIGH); // turn on the LED  
  delay (1000); // wait one second (1000 milliseconds)  
  digitalWrite (LED_PIN, LOW); // turn off the LED  
  delay (1000); // wait one second  
}
```

Είναι ένα χαρακτηριστικό των περισσότερων πλακετών Arduino ότι έχουν ένα LED και μία αντίσταση φορτίου που συνδέονται μεταξύ του pin 13 και του εδάφους, ένα βολικό χαρακτηριστικό για πολλά απλά τεστ. Ο προηγούμενος κώδικας δεν θα αναγνωριστεί από ένα κανονικό μεταγλωττιστή C++ ως έγκυρο πρόγραμμα, έτσι ώστε όταν ο χρήστης κάνει κλικ

στο κουμπί "Upload to I / O board" στο IDE, ένα αντίγραφο του κώδικα θα γραφτεί σε ένα προσωρινό αρχείο με ένα παραπάνω include στην κορυφή και μία πολύ απλή συνάρτηση main() στο τέλος, για να φτιάξει ένα έγκυρο C++ πρόγραμμα.

Το IDE του Arduino χρησιμοποιεί το GNU toolchain και το AVR Libc για να μεταγλωττίζει προγράμματα και το avrdude για να φορτώνει προγράμματα στην πλακέτα.

Δεδομένου ότι η πλατφόρμα Arduino χρησιμοποιεί Atmel μικροελεγκτές, το περιβάλλον ανάπτυξης της Atmel, το AVR Studio ή νεότερη έκδοση του Atmel Studio, μπορεί επίσης να χρησιμοποιηθεί για την ανάπτυξη λογισμικού για το Arduino. (βλέπε [12])

4 ΠΑΡΟΥΣΙΑΣΗ ΤΗΣ ΠΡΟΣΘΕΤΗΣ ΠΛΑΚΕΤΑΣ GPS-3G/GSM

4.1 ΒΑΣΙΚΑ ΧΑΡΑΚΤΗΡΙΣΤΙΚΑ ΚΑΙ ΛΕΙΤΟΥΡΓΙΕΣ

Η πρόσθετη πλακέτα (shield) 3G/GPRS + GPS για Arduino δίνει την δυνατότητα συνδεσιμότητας σε δίκτυα WCDMA και HSPA κινητών τηλεπικοινωνιών υψηλών ταχυτήτων. Με αυτό τον τρόπο μπορεί να επιτευχθεί η δημιουργία μιας νέας γενιάς διαδραστικών εφαρμογών στην νέα εποχή του Internet.

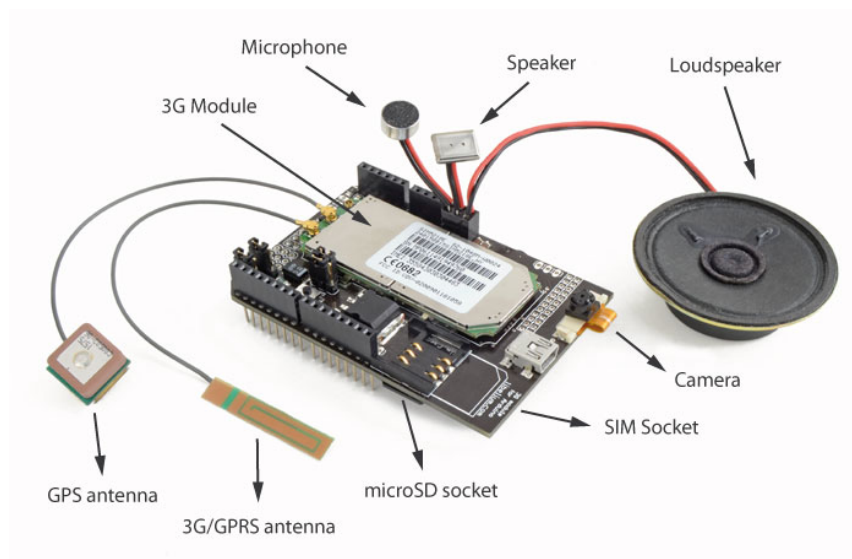
Το shield αυτό διαθέτει επίσης μονάδα με την δυνατότητα λήψης δεδομένων GPS, λειτουργία η οποία ενεργοποιεί τον εντοπισμό της θέσης της συσκευής είτε σε εξωτερικό χώρο είτε σε εσωτερικό συνδυάζοντας τις πληροφορίες από τα πλαίσια NMEA που δέχεται από τους δορυφόρους του συστήματος GPS με τις πληροφορίες της ταυτότητας κυψέλης από το δίκτυο κινητής τηλεφωνίας. Ιδιαίτερα χρήσιμα χαρακτηριστικά της πλακέτας αυτής είναι οι πολυμεσικές δυνατότητές της όπως λήψη φωτογραφιών και καταγραφή βίντεο, αναπαραγωγή και καταγραφή ήχου, καθώς και οι δυνατότητες αποθήκευσης σε κάρτα μνήμης που διαθέτει. Επιπροσθέτως το shield παρέχει την δυνατότητα αναπαραγωγής αρχείων mp3 μετά από αναβάθμιση του firmware του.

Επίσης το συγκεκριμένο shield διαθέτει δυνατότητες σειριακής επικοινωνίας μέσω υποδοχής miniUSB για επικοινωνία με τον Η/Υ και ακροδέκτες για επικοινωνία με την πλακέτα του Arduino.

Όλες οι δυνατότητες που διαθέτει η πρόσθετη αυτή πλακέτα υλοποιούνται κάνοντας χρήση της μονάδας (module) SIM5218 της εταιρείας SIMCom. Ο χειρισμός του module γίνεται κάνοντας χρήση εντολών «AT». Το προαναφερθέν module διατίθεται σε 4 εκδόσεις δικτύου 3G κινητής τηλεφωνίας (βλέπε [18]):

- SIM5218A: Tri-Band UMTS/HSPA850/1900/2100 MHz
- SIM5218E: Tri-Band UMTS/HSPA 900/1900/2100MHz
- SIM5218C: Single-Band UMTS 2100MHz
- SIM5218J: Tri-Band UMTS/HSPA800/1900/2100 MHz

Τέλος αξίζει να αναφέρουμε ότι το συγκεκριμένο shield μπορεί να χρησιμοποιηθεί και αυτόνομα χωρίς την σύνδεση του με πλακέτα Arduino. Σε αυτή την περίπτωση μπορεί να χρησιμοποιηθεί ως 3G/GPRS modem και να παρέχει πρόσβαση στο Internet στον Η/Υ με τον οποίο συνδέεται.



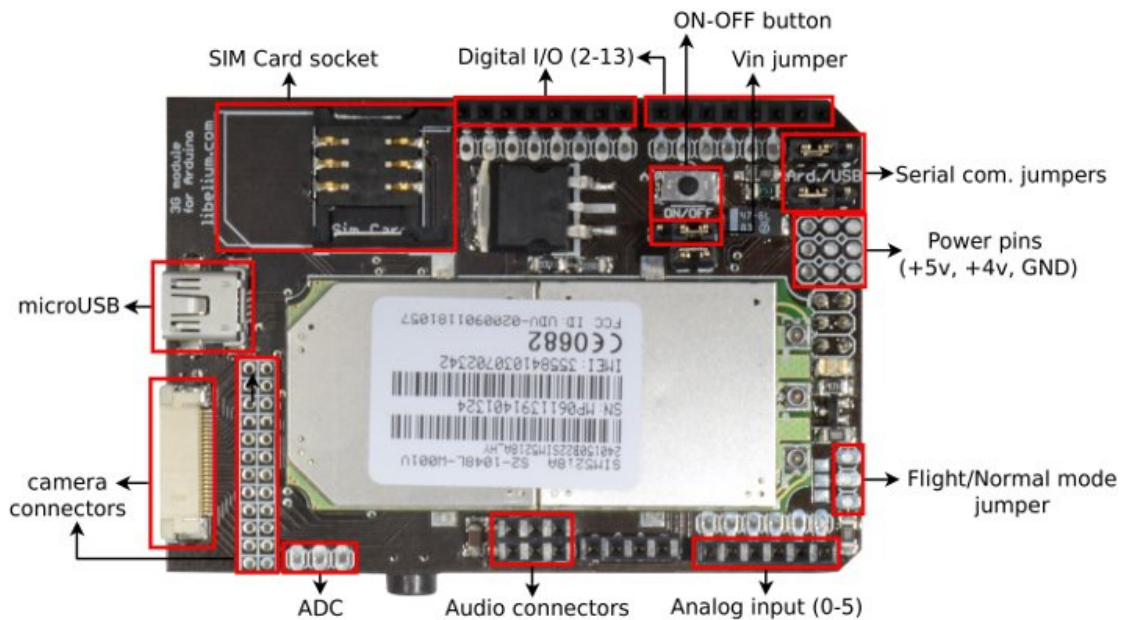
Εικόνα 30 - Αναλυτική Παρουσίαση του 3G/GPRS + GPS shield

4.2 ΚΥΡΙΕΣ ΛΕΙΤΟΥΡΓΙΕΣ ΕΠΙΚΟΙΝΩΝΙΑΣ

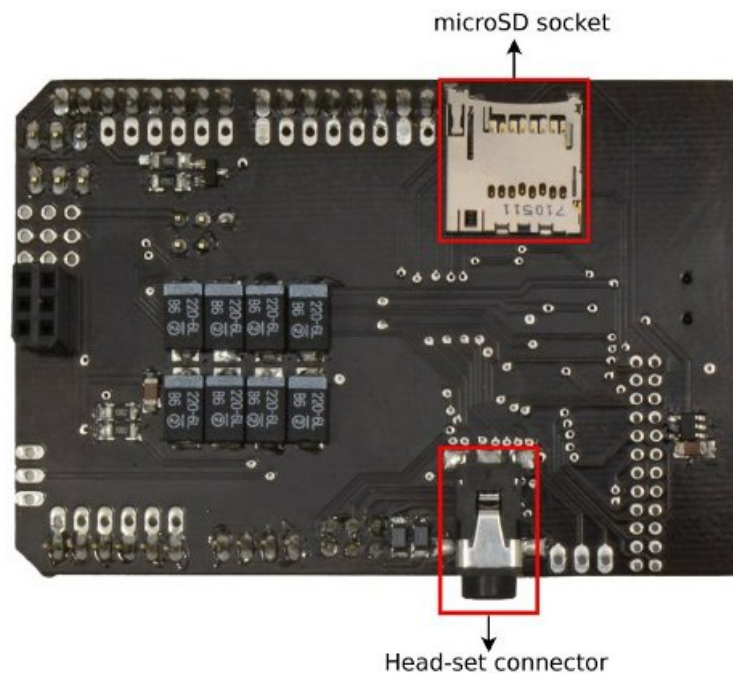
Η κύρια λειτουργία της πρόσθετης πλακέτας είναι η παροχή σύνδεσης σε δίκτυα τρίτης γενιάς κινητής τηλεφωνίας τύπου WCDMA και HSPA. Πέρα από την δυνατότητα λειτουργίας σε δίκτυα 3G υποστηρίζει και σύνδεση σε δίκτυα προηγούμενης γενιάς. Η συσκευή μας δίνει την δυνατότητα να πραγματοποιούμε λειτουργίες κλήσεων εισερχόμενων και εξερχόμενων όσο και πρόσβασης στο Internet μέσω δικτύου κινητής τηλεφωνίας 3G αλλά και 2G υποστηρίζοντας μέγιστες ταχύτητες έως 7.2 Mbps για κατέβασμα δεδομένων προς την συσκευή και 5.5 Mbps για αποστολή δεδομένων από την συσκευή. Όταν η συσκευή συνδέεται πάνω σε πλακέτα Arduino περιορίζεται η ταχύτητα επικοινωνίας του σε περίπου 115 Kbps για κατέβασμα δεδομένων προς την συσκευή και σε 35 Kbps για αποστολή δεδομένων από την συσκευή. Η μειωμένες ταχύτητες παρουσιάζονται λόγω της χαμηλής ταχύτητας (115200 bauds) της σειριακής επικοινωνίας μεταξύ Arduino και shield. Παρόλα αυτά ακόμη και όταν το shield είναι συνδεδεμένο σε Arduino κατά την διάρκεια μεταφοράς δεδομένων τα οποία δεν προέρχονται από το Arduino όπως πχ. κατά την μεταφορά αρχείου από την κάρτα μνήμης του shield προς έναν διακομιστή ftp ή http τότε επιτυγχάνονται υψηλότερες ταχύτητες.

Εκτός της επικοινωνίας μέσω δικτύου κινητής τηλεφωνίας η συσκευή διαθέτει και υποδοχή mini USB για σειριακή επικοινωνία με ηλεκτρονικό υπολογιστή. Η θύρα USB

χρησιμοποιείται για την μεταφορά δεδομένων από την διεπαφή 3G/GPRS από και προς τον Η/Υ για να χρησιμοποιηθούν από τον χρήστη. Στις παρακάτω φωτογραφίες παρουσιάζονται λεπτομερώς οι διάφορες υποδοχές, διακόπτες και ακροδέκτες του shield.

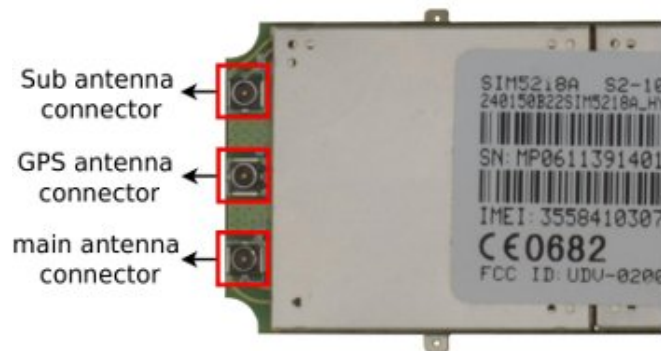


Εικόνα 31 - Παρουσίαση των εξαρτημάτων/υποδοχών της άνω πλευράς του shield

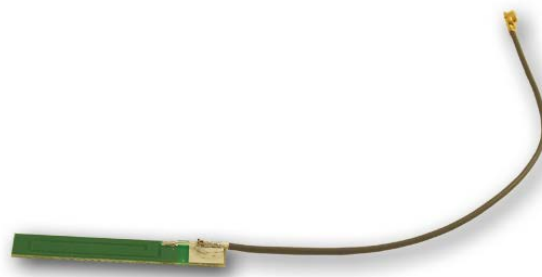


Εικόνα 32 - Παρουσίαση των εξαρτημάτων/υποδοχών της κάτω πλευράς του shield

Η μονάδα SIM5218 διαθέτει δύο υποδοχές τύπου UFL στις οποίες μπορούν να συνδεθούν 2 κεραίες (μία κύρια και μία δευτερεύουσα) για την λήψη σήματος από το δίκτυο κινητής τηλεπικοινωνίας. Μαζί με τις δύο αυτές υποδοχές βρίσκεται και η υποδοχή τύπου UFL επίσης της κεραίας του GPS. Οι υποδοχές αυτές παρουσιάζονται στην παρακάτω εικόνα. Επίσης παρουσιάζεται και η εσωτερική κεραία GSM που χρησιμοποιούμε στην κατασκευή μας (βλέπε [19]).



Εικόνα 33 - Υποδοχές όπου συνδέονται οι κεραίες των GSM και GPS στο module SIM5218



Εικόνα 34 - Εσωτερική κεραία GSM

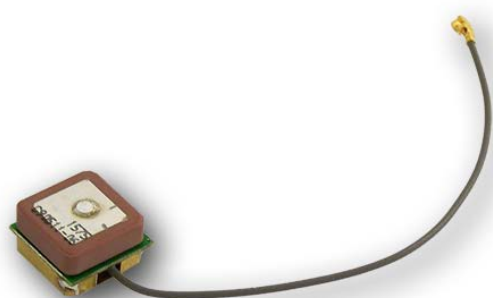
4.3 ΛΕΙΤΟΥΡΓΙΑ ΕΝΤΟΠΙΣΜΟΥ ΓΕΩΓΡΑΦΙΚΗΣ ΘΕΣΗΣ (GPS)

Πέρα από τις δυνατότητες επικοινωνίας της συσκευής υποστηρίζει και λειτουργία εντοπισμού. Η προαναφερθείσα λειτουργία επιτυγχάνεται με χρήση των τεχνολογιών assisted mobile (A-GPS) και mobile based (S-GPS) οι οποίες αξιοποιούν δεδομένα από το δίκτυο κινητής τηλεφωνίας για καλύτερο εντοπισμό της γεωγραφικής θέσης της συσκευής σε κλειστούς χώρους. Σε περίπτωση μη χρήσης δικτύου κινητής τηλεφωνίας υπάρχει η

δυνατότητα λειτουργίας της συσκευής και σε κατάσταση αυτόνομη (standalone) αξιοποιώντας μόνο τα δεδομένα από τους δορυφόρους για τον προσδιορισμό της θέσης της συσκευής. Στον παρακάτω πίνακα παρουσιάζονται λεπτομέρειες σχετικά με τις τρεις δυνατές λειτουργίες του GPS. Ακόμη στην παρακάτω εικόνα παρουσιάζεται η εσωτερική κεραία που χρησιμοποιούμε για την λήψη του σήματος από τους δορυφόρους του συστήματος GPS.

Πίνακας 2 - Πίνακας επεξήγησης των 3 καταστάσεων λειτουργίας του GPS

MS-assisted	Server	Module
Location server sends aiding data that is valid for the current fix	Sending data	
Module sends code phases		Code phases
Server calculates position	Calculate position	
MS-based	Server	Module
Location server sends aiding data that is valid for the current fix	Send aiding data	
Module calculates		Calculate position
Stand alone	Server	Module
Module demodulates data form GPS satellite		Demodulates GPS satellite data
Module calculates position		Calculate position



Εικόνα 35 - Εσωτερική κεραία GPS

Στέλνοντας στην μονάδα την εντολή AT+CGPSINFO επιστρέφει τις πληροφορίες από το GPS μέσω του παρακάτω string: +CGPSINFO: [<latitude>], [<N/S>], [<longitude>],

[<E/W>], [<date>], [<UTC_time>], [<altitude>], [<speedOG>], [<course>]. Τα δεδομένα που περιλαμβάνονται στο παραπάνω string αφορούν το γεωγραφικό πλάτος, το γεωγραφικό μήκος, την ημερομηνία, την ώρα UTC (Coordinated Universal Time), το υψόμετρο, την ταχύτητα με την οποία κινείται η συσκευή μας και τον προσανατολισμό της. Όλα τα παραπάνω δεδομένα μπορούν να χρησιμοποιηθούν και να επεξεργαστούν από τον κώδικα που έχουμε αναπτύξει στην κύρια πλακέτα της πλατφόρμας του Arduino (βλέπε [19]).

4.4 ΕΠΙΠΡΟΣΘΕΤΕΣ ΔΥΝΑΤΟΤΗΤΕΣ ΤΗΣ ΠΡΟΣΘΕΤΗΣ ΠΛΑΚΕΤΑΣ

Η πλακέτα που χρησιμοποιούμε για τον εντοπισμό των συντεταγμένων και την αποστολή πληροφοριών μέσω δικτύου κινητής τηλεφωνίας πέρα από αυτές τις δύο κύριες λειτουργίες διαθέτει και αρκετές επιπρόσθετες δυνατότητες οι οποίες μπορούν να αξιοποιηθούν ανάλογα με την εφαρμογή που επιθυμούμε να υλοποιήσουμε.

Οι πρόσθετες αυτές δυνατότητες αφορούν στην διαχείριση πολυμέσων όπως ήχος, φωτογραφίες και βίντεο. Τα πολυμεσικά αυτά χαρακτηριστικά είναι τα παρακάτω:

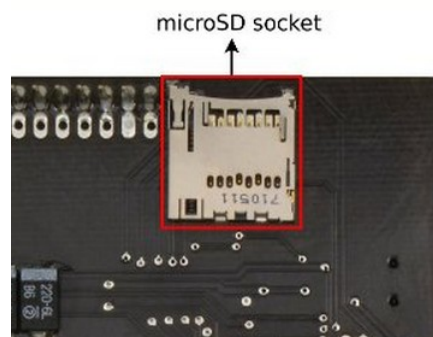
- Βιντεοκάμερα ανάλυσης VGA(640x480)pixels για καταγραφή βίντεο και λήψη φωτογραφιών
- Αναπαραγωγή συμπιεσμένων(mp3) αρχείων ήχου
- Υποδοχή σύνδεσης μικροφώνου για καταγραφή ήχου και χρήση σε τηλεφωνική κλήση
- Υποδοχή σύνδεσης ηχείου για αναπαραγωγή ήχου
- Υποδοχή σύνδεσης ακουστικών ή hands free για πραγματοποίηση τηλεφωνικών κλήσεων ή αναπαραγωγή και καταγραφή ήχου



Εικόνα 36 - Hands free συνδεδεμένο στο shield

Επιπροσθέτως η πλακέτα διαθέτει και αποθηκευτικές δυνατότητες για αποθήκευση π.χ. αρχείων πολυμέσων (.jpg, .mp3). Η δυνατότητα αυτή προσφέρεται από το παρακάτω χαρακτηριστικό:

- Υποδοχή για κάρτα μνήμης τύπου microSD μεγέθους έως 32 Gigabytes και συστήματος αρχείων τύπου FAT16



Εικόνα 37 - Υποδοχή κάρτας μνήμης microSD

Τέλος το συγκεκριμένο shield υποστηρίζει πληθώρα πρωτοκόλλων για την επίτευξη επιπλέον λειτουργιών. Τα πρωτόκολλα αυτά είναι τα παρακάτω (βλέπε [19]):

- HTTP (Πρωτόκολλο Μεταφοράς Υπερκειμένου)
- HTTPS (Πρωτόκολλο Μεταφοράς Υπερκειμένου Κρυπτογραφημένο)
- TCP (Πρωτόκολλο Ελέγχου Μεταφοράς, για αμφίδρομη επικοινωνία μέσω δικτύου)
- UDP (Εφαρμογές Audio και video streaming)

- FTP (Αποστολή/Λήψη αρχείων)
- FTPS (Αποστολή/Λήψη αρχείων σε κρυπτογραφημένη σύνδεση)
- POP3 (Λήψη email)
- SMTP (Αποστολή email)
- SMS (Αποστολή σύντομου μηνύματος κειμένου σε κινητό τηλέφωνο)
- I²C (Σειριακή επικοινωνία με περιφερειακές συσκευές)

Όλες οι παραπάνω δυνατότητες της πλακέτας μπορούν να χρησιμοποιηθούν για την ανάπτυξη σύνθετων εφαρμογών που θα καλύπτουν τις υψηλές απαιτήσεις των χρηστών του σύγχρονου διαδικτύου και τηλεπικοινωνιών.

5 ΥΛΟΠΟΙΗΣΗ ΤΟΥ ΥΛΙΚΟΥ ΚΑΙ ΛΟΓΙΣΜΙΚΟΥ ΤΗΣ ΣΥΣΚΕΥΗΣ

5.1 ΣΤΟΙΧΕΙΑ ΤΗΣ ΟΛΟΚΛΗΡΩΜΕΝΗΣ ΚΑΤΑΣΚΕΥΗΣ

Στην ενότητα αυτή θα αναφερθούμε στο υλικό κομμάτι της εργασίας μας. Θα παρουσιαστούν αναλυτικά όλα τα στοιχεία/υλικά που απαιτήθηκαν για την κατασκευή μας είτε πρόκειται για μηχανολογικής είτε για ηλεκτρονικής φύσεως. Θα ξεκινήσουμε από το υλικά που χρησιμοποιήθηκαν για το εξωτερικό μέρος της συσκευής και θα προχωρήσουμε σε αυτά που την απαρτίζουν εσωτερικά.

1. Εξωτερική θήκη συσκευής

Σημαντικό ρόλο στην υλοποίηση της κατασκευής μας έπαιξε η εξωτερική θήκη. Λόγω του ότι η κατασκευή αποτελείται από ηλεκτρονικά στοιχεία, τα οποία είναι ευαίσθητα σε εξωτερικές παρεμβολές, όπως π.χ. πτώσεις ή άμεση επαφή με νερό ή σκόνη, χρησιμοποιήθηκε μια εξωτερική θήκη κυρίως για την προστασία της. Έπρεπε επίσης να βρεθεί ένας τρόπος να επιτρέπει την εύκολη μεταφορά της συσκευής με όλα τα υλικά να συνδέονται μεταξύ τους και με την αποφυγή βέβαια ανεπιθύμητων αποσυνδέσεων. Αυτή την λύση την έδωσε η εξωτερική θήκη η οποία φαίνεται στην παρακάτω φωτογραφία.



Εικόνα 38 - Εξωτερική θήκη συσκευής

2. Αδιάβροχη υποδοχή USB με καπάκι

Η χρησιμοποίηση της εξωτερικής θήκης έκανε όμως δύσκολη την σύνδεση του εσωτερικού της συσκευής με τον «έξω» κόσμο. Για τον λόγο αυτόν κρίθηκε απαραίτητη η χρησιμοποίηση μιας υποδοχής USB η οποία έχει διττό ρόλο. Επιτρέπει την τροφοδοσία της συσκευής και παράλληλα την σύνδεση αυτής με ηλεκτρονικό υπολογιστή. Προτιμήθηκε υποδοχή USB αδιάβροχη, η οποία διαθέτει εσωτερικά λάστιχο για καλύτερη στεγανοποίηση. Η εν λόγω υποδοχή φαίνεται παρακάτω.



Εικόνα 39 - Αδιάβροχη υποδοχή USB με καπάκι

3. Υποδοχή εξωτερικής τροφοδοσίας πλακέτας

Η κυρίως λειτουργία της συσκευής πραγματοποιείται χωρίς την ύπαρξη ηλεκτρονικού υπολογιστή. Για τον λόγο αυτό χρησιμοποιήθηκε μια υποδοχή εξωτερικής τροφοδοσίας της συσκευής. Στην συγκεκριμένη υποδοχή μπορεί να συνδεθεί καλώδιο τροφοδοσίας 7-12V, το οποίο προέρχεται από εξωτερική πηγή, είτε από μπαταρία είτε από τροφοδοτικό. Στην εικόνα που ακολουθεί φαίνεται η συγκεκριμένη υποδοχή.



Εικόνα 40 - Υποδοχή εξωτερικής τροφοδοσίας πλακέτας

4. Διακόπτης ON/OFF

Ένας διακόπτης ON/OFF τοποθετήθηκε στο εξωτερικό της θήκης της συσκευής για να δίνεται η δυνατότητα στο χρήστη να μπορεί να ενεργοποιεί ή να απενεργοποιεί την συσκευή όταν αυτή τροφοδοτείται από εξωτερική πηγή τροφοδοσίας. Ο διακόπτης, που φαίνεται στην εικόνα που ακολουθεί, συνδέεται ανάμεσα στην εξωτερική υποδοχή τροφοδοσίας και στον μικροεπεξεργαστή.



Εικόνα 41 - Διακόπτης ON/OFF

5. LED και βάση στήριξης LED

Ο μικροεπεξεργαστής διαθέτει κάποια LED ενσωματωμένα, τα οποία παρέχουν στο χρήστη ένα τρόπο αναγνώρισης της εκάστοτε λειτουργίας της συσκευής. Λόγω του ότι ο μικροεπεξεργαστής τοποθετήθηκε εσωτερικά έπρεπε να μεταφερθούν και στο εξωτερικό μέρος τα «μηνύματα» που στέλνει μέσω των LED. Για τον λόγο αυτό χρησιμοποιήθηκαν 3 LED, τα οποία ενσωματώθηκαν στην εξωτερική θήκη μέσω των βάσεων στήριξης. Στην παρακάτω εικόνα διακρίνονται τα LED και οι βάσεις στήριξης τους.



Εικόνα 42 - LED και βάση στήριξης LED

6. Πλακέτα Arduino Leonardo

Αποτελεί το κεντρικό μέρος της κατασκευής. Είναι υπεύθυνος για την μεταφορά των εντολών στην πρόσθετη πλακέτα 3G/GPRS + GPS και μέσω αυτού τροφοδοτείται ολόκληρο το κύκλωμα. Επίσης αποτελεί το μέσο σύνδεσης ανάμεσα στον χρήστη και την συσκευή. Η πλήρης ανάλυση του γίνεται σε προηγούμενο κεφάλαιο και παρακάτω υπάρχει μια εικόνα του.



Εικόνα 43 – Πλακέτα Arduino Leonardo

7. Πρόσθετη πλακέτα 3G/GPRS + GPS

Η πλήρης λειτουργία και παρουσίαση της γίνεται σε προηγούμενο κεφάλαιο. Η πρόσθετη πλακέτα είναι υπεύθυνη για την εύρεση των συντεταγμένων της συσκευής και για την αποστολή τους. Απεικονίζεται στην συνέχεια μαζί με τις δυο κεραίες του GPS και του GSM.



Εικόνα 44 - Πρόσθετη πλακέτα 3G/GPRS + GPS

8. Pins σύνδεσης Arduino με Shield

Τα pins είναι τοποθετημένα ανάμεσα στον μικροεπεξεργαστή Arduino και την πρόσθετη πλακέτα Shield. Είναι υπεύθυνα για την άμεση σύνδεση των δυο. Στην παρακάτω εικόνα διακρίνονται τα διάφορα μεγέθη τους.



Εικόνα 45 - Pins σύνδεσης Arduino με Shield

9. Αρσενικό βύσμα *micro-usb*

Ο μικροεπεξεργαστής διαθέτει θηλυκό βύσμα USB για σύνδεση του με τον ηλεκτρονικό υπολογιστή. Επειδή όμως το συγκεκριμένο βύσμα δεν είναι τόσο διαδεδομένο προτιμήθηκε στην εξωτερική θήκη να χρησιμοποιηθεί υποδοχή κανονικής USB. Το εξάρτημα το οποίο συνδέει την παραπάνω υποδοχή με τον μικροεπεξεργαστή και μετατρέπει το *micro-USB* σε USB είναι ένα αρσενικό βύσμα *micro-USB* και φαίνεται στην εικόνα που ακολουθεί. Βρίσκεται μόνιμα τοποθετημένο στο εξωτερικό της κατασκευής.



Εικόνα 46 - Αρσενικό βύσμα micro-usb

10. Καλώδιο σύνδεσης βύσματος USB με εξωτερική υποδοχή USB

Το καλώδιο σύνδεσης βύσματος USB με εξωτερική υποδοχή USB βρίσκεται και αυτό στο εσωτερικό της θήκης και είναι υπεύθυνο για την μεταξύ τους σύνδεση. Στην παρακάτω εικόνα φαίνεται το πώς μοιάζει.



Εικόνα 47 - Καλώδιο σύνδεσης βύσματος USB με εξωτερική υποδοχή USB

11. Διακόπτης tilt

Ο συγκεκριμένος διακόπτης βρίσκεται τοποθετημένος στο εσωτερικό της κατασκευής και είναι υπεύθυνος για την ανίχνευση της κίνησης της συσκευής έτσι ώστε να ξεκινήσει η συσκευή την λειτουργία της. Ο διακόπτης φαίνεται στην παρακάτω εικόνα.



Εικόνα 48 - Διακόπτης tilt

5.2 ΤΡΟΦΟΔΟΣΙΑ ΤΗΣ ΚΑΤΑΣΚΕΥΗΣ

Στην συγκεκριμένη ενότητα θα παρουσιαστούν οι τρόποι τροφοδοσίας της κατασκευής μας. Όπως αναφέρθηκε και σε προηγούμενη ενότητα υπάρχουν 2 τρόποι για να μπορέσει κάποιος να τροφοδοτήσει την συσκευή.

- Μέσω της εξωτερικής υποδοχής USB
- Μέσω της εξωτερικής υποδοχής τροφοδοσίας

Οι υποδοχές USB που χρησιμοποιούν όλοι οι ηλεκτρονικοί υπολογιστές για να επικοινωνούν με περιφερειακά συστήματα, έχουν την δυνατότητα να παρέχουν τάση της

τάξεως των 5V σε αυτά. Επομένως με την σύνδεση της εξωτερικής υποδοχής USB με την υποδοχή USB ενός ηλεκτρονικού υπολογιστή επιτυγχάνεται και η τροφοδοσία της συσκευής μας. Αυτός είναι ο ένας τρόπος για να τροφοδοτηθεί η κατασκευή μας.

Ένας άλλος τρόπος για τροφοδότηση της συσκευής είναι δια μέσου της εξωτερικής υποδοχής τροφοδοσίας. Στην συγκεκριμένη υποδοχή μπορεί να συνδεθεί βύσμα 2,5 χιλιοστών, το οποίο προέρχεται από εξωτερική πηγή, όπως έχει ήδη αναφερθεί προηγουμένως. Εξωτερική πηγή μπορεί να θεωρηθεί ένα τροφοδοτικό, το οποίο διαθέτει το εν λόγω βύσμα. Επίσης εξωτερική πηγή μπορεί να θεωρηθεί μια μπαταρία, η οποία έχει τοποθετηθεί σε μια υποδοχή μπαταρίας όπως αυτή που φαίνεται στην εικόνα 41. Λόγω του ότι η συσκευή πρέπει να λειτουργεί χωρίς τροφοδοσία έπρεπε να χρησιμοποιηθεί μια πηγή τροφοδοσίας που να επιτρέπει στην συσκευή να λειτουργεί ενώ βρίσκεται εν κινήσει. Για τα πειράματα που διεξήγαμε σε εξωτερικούς χώρους χρησιμοποιήθηκε η συγκεκριμένη θήκη.



Εικόνα 49 - Θήκη υποδοχής 9V εξωτερικής μπαταρίας

5.3 ΕΠΙΚΟΙΝΩΝΙΑ ΤΗΣ ΣΥΣΚΕΥΗΣ ΜΕ ΤΟΝ ΧΡΗΣΤΗ

Όπως σε κάθε ηλεκτρονική συσκευή έτσι και στην περίπτωσή μας παρέχεται στον χρήστη η δυνατότητα να βρίσκεται σε συνεχή επικοινωνία με την συσκευή. Αυτό σημαίνει ότι θα μπορεί ο χρήστης να παρέμβει εάν το θέλει στη λειτουργία της συσκευής και επίσης ότι θα μπορεί να παρακολουθεί τις λειτουργίες της συσκευής.

Στην πρώτη περίπτωση, για να επιτευχθεί η σύνδεση με την συσκευή θα πρέπει ο χρήστης να κάνει χρήση ενός ηλεκτρονικού υπολογιστή με τον οποίο θα συνδεθεί με την συσκευή μέσω της εξωτερικής υποδοχής USB. Μ' αυτόν τρόπο καταφέρνουμε να επιτρέψουμε στον χρήστη να έχει πλήρη πρόσβαση στον κώδικα της συσκευής και να μπορεί να τον επεξεργαστεί. Επίσης μέσω της σειριακής οθόνης που παρέχει το πρόγραμμα του Arduino (IDE), μπορεί ο χρήστης να παρακολουθεί βασικές λειτουργίες της συσκευής, όπως π.χ. να παρακολουθεί τις εντολές που εκτελούνται στο Arduino και αποστέλλονται μέσω της σειριακής επικοινωνίας.

Επειδή όμως η συσκευή σχεδιάστηκε για να χρησιμοποιείται χωρίς την χρήση ηλεκτρονικού υπολογιστή, να είναι δηλαδή αυτόνομη, ο χρήστης μπορεί να παρακολουθεί την συσκευή και με άλλους τρόπους.

Ο ένας είναι μέσω ενδεικτικών λυχνιών (LED). Υπάρχουν 3 LED διαφορετικού χρώματος τοποθετημένα επάνω στην εξωτερική θήκη, για να μπορεί ο χρήστης κάθε φορά να ξεχωρίσει την διαδικασία που εκτελείται.

- Το κόκκινο LED. Όταν είναι αναμμένο δείχνει ότι η συσκευή τροφοδοτείται με τάση και ο διακόπτης ON/OFF είναι ενεργοποιημένος.
- Το μπλε LED, το δεύτερο κατά σειρά LED. Όταν αναβοσβήνει δείχνει ότι η συσκευή βρίσκεται σε λειτουργία standby, ότι δηλαδή δεν έχει ανιχνεύσει κίνηση ακόμα, ενώ όταν ανάβει σταθερά δείχνει ότι ο διακόπτης tilt switch έχει στείλει σήμα.
- Το πράσινο LED, το τρίτο και τελευταίο κατά σειρά LED, το οποίο ανάβει σταθερά όταν έχει ανιχνευθεί κίνηση και δείχνει ότι η συσκευή έχει ενεργοποιηθεί και εκτελείται ο κώδικας με τον οποίο είναι προγραμματισμένη.

Ένας άλλος τρόπος παρακολούθησης της συσκευής είναι μέσω της διεπαφής 3G/GPRS κάνοντας χρήση του δικτύου κινητής τηλεπικοινωνίας. Μέσω της διεπαφής αυτής, μπορούν να αποστέλλονται δεδομένα από τη συσκευή, κάνοντας χρήση των υπηρεσιών του δικτύου. Οι υπηρεσίες που εκμεταλλευόμαστε είναι αυτές του *e-mail* (με χρήση των πρωτοκόλλων SMTP και POP3) και του *sms* στα οποία περιλαμβάνονται πληροφορίες που αφορούν την θέση της συσκευής (γεωγραφικό μήκος και πλάτος), την ώρα και την ημερομηνία.

5.4 ΑΝΑΛΥΤΙΚΗ ΠΑΡΟΥΣΙΑΣΗ ΤΗΣ ΚΑΤΑΣΚΕΥΗΣ

Στην ενότητα αυτή θα παρουσιαστεί αναλυτικά η διαδικασία συναρμολόγησης του υλικού μέρους της κατασκευής. Το πρώτο βήμα για να ξεκινήσει η συναρμολόγηση ήταν η συλλογή των απαιτούμενων υλικών, τα οποία έχουν προαναφερθεί στην ενότητα 5.1. Η επιλογή των υλικών που χρησιμοποιήθηκαν έγινε μετά από επιβεβαίωση της συμβατότητας τους, στα ηλεκτρονικά και μηχανολογικά χαρακτηριστικά τους.

Αρχικά επιλέχθηκε ως πλατφόρμα λειτουργίας της συσκευής το Arduino Leonardo, διότι διαθέτει σύγχρονο μικροελεγκτή AVR, αρκετές ψηφιακές εισόδους και εξόδους για τις ανάγκες της κατασκευής και εύκολο στη χρήση περιβάλλον ανάπτυξης λογισμικού. Όσον αφορά στην επιλογή της συγκεκριμένης πλακέτας σημαντικό ρόλο έπαιξε και το μικρό μέγεθος της.

Εν συνεχεία επιλέχθηκε η πρόσθετη πλακέτα 3G/GPRS + GPS λόγω πλήρους συμβατότητας με το Leonardo (συμβατοί ακροδέκτες ψηφιακών και αναλογικών εισόδων/εξόδων) και επειδή διαθέτει ενσωματωμένους δέκτες GPS και GSM.

Το κυρίως ηλεκτρονικό κομμάτι της κατασκευής αποτελείται από τις δυο παραπάνω πλακέτες στις οποίες έχουν προστεθεί δευτερεύοντα ηλεκτρονικά στοιχεία όπως τα LED και ο διακόπτης tilt.

Μετά τη συλλογή των υλικών έγινε η μελέτη για την κατάλληλη διάταξή τους στην εξωτερική θήκη όπου και θα τοποθετηθούν. Όπως φαίνεται στην εικόνα 50 στο άνω μέρος της θήκης ανοίχθηκαν 5 οπές. Μία για κάθε βάση στήριξης των τριών LED, μια για την στήριξη της εξωτερικής υποδοχής USB και μία για την διέλευση των καλωδίων των δυο

κεραιών, GPS και GSM. Στην εμπρός πλαϊνή όψη έγιναν δύο οπές, εκ των οποίων η μεγαλύτερη χρησιμοποιείται για την τοποθέτηση του διακόπτη ON/OFF και η μικρότερη για την τοποθέτηση της υποδοχής εξωτερικής τροφοδοσίας της συσκευής. (εικόνα 51)



Εικόνα 50 - Οπές στο άνω μέρος της θήκης



Εικόνα 51 - Οπές στην εμπρός πλαϊνή όψη

Έπειτα τοποθετήθηκαν στην βάση της θήκης το Leonardo, το shield και ο διακόπτης tilt όπως διακρίνεται στην παρακάτω εικόνα. Λόγω του ιδιαίτερα περιορισμένου χώρου του εσωτερικού της θήκης παρουσιάστηκαν αρκετές δυσκολίες στην τοποθέτηση των παραπάνω υλικών.



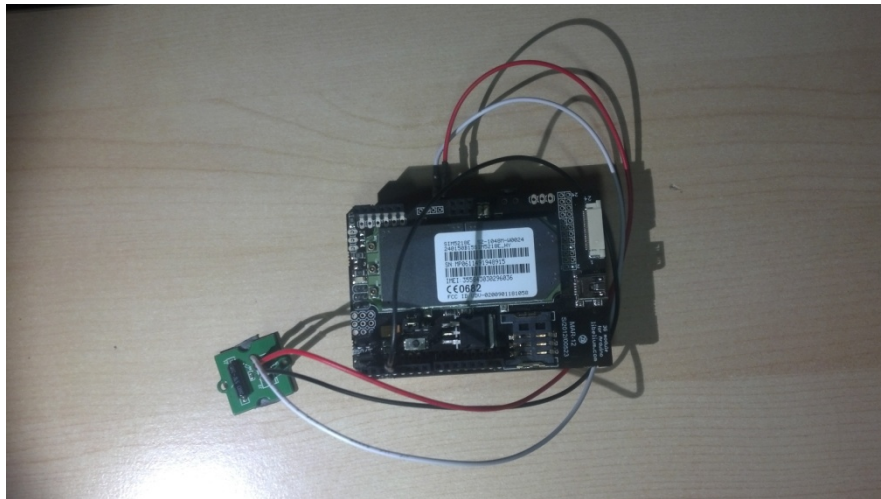
Εικόνα 52 – Τοποθέτηση Leonardo, shield και tilt switch εντός της θήκης

Στην συνέχεια πραγματοποιήθηκε η σύνδεση μέσω καλωδίων της εξωτερικής αδιάβροχης υποδοχής USB με το αρσενικό βύσμα micro-USB, το οποίο έχει τοποθετηθεί πάνω στο αντίστοιχο βύσμα της πλακέτας του Leonardo. (εικόνα 53)

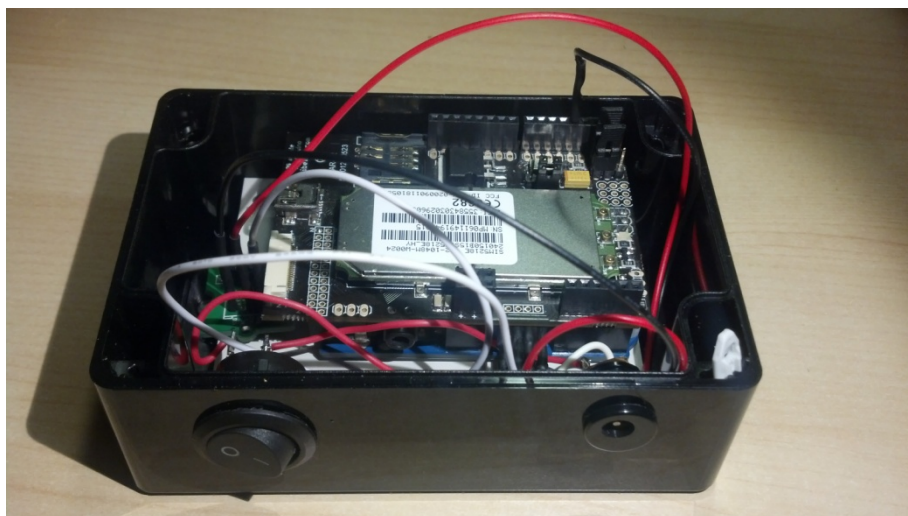


Εικόνα 53 - Σύνδεση υποδοχής USB με micro-USB

Επίσης συνδέθηκαν μεταξύ τους ο διακόπτης tilt με τους κατάλληλους ακροδέκτες του Leonardo και του shield (εικόνα 54). Ακόμη συνδέθηκε πάνω στο Leonardo ο διακόπτης ON/OFF και η εξωτερική υποδοχή τροφοδοσίας. Τα δυο τελευταία εξαρτήματα συνδέονται και μεταξύ τους σε σειρά (εικόνα 55). Έπειτα συνδέθηκαν τα τρία LED ένδειξης λειτουργίας με τις αντίστοιχες ψηφιακές εξόδους του Leonardo. Όλα τα παραπάνω εξαρτήματα τοποθετήθηκαν στις προκαθορισμένες θέσεις πάνω στην εξωτερική θήκη της κατασκευής. Τέλος στερεώθηκαν στα δεξιά της άνω πλευράς της θήκης οι δύο κεραίες και συνδέθηκαν πάνω στο module του shield. Η τελική εξωτερική όψη της συσκευής φαίνεται στην εικόνα 56.



Εικόνα 54 - Σύνδεση Leonardo, shield και tilt switch



Εικόνα 55 - Σύνδεση διακόπτη ON/OFF, εξωτερικής τροφοδοσίας με τα κύρια μέρη



Εικόνα 56 – Εξωτερική όψη της συσκευής

5.5 ΕΠΕΞΗΓΗΣΗ ΠΗΓΑΙΟΥ ΚΩΔΙΚΑ ΚΑΙ ΛΕΙΤΟΥΡΓΙΑΣ

```
int led = 13; //shield led
int ledb = 12; //blue led
int ledg = 11; //green led

int onModulePin = 2; //power on the modulepin
int x=0;
int w=0;

char dataIn[255]; //GPS array variables
char answer[255];
char Lat[12];
char Latdecimal[9];
char Latsecond[9];
char Long[13];
char Longdecimal[9];
char Longsecond[9];
char Latsymbol[2];
char Longsymbol[2];
char Datearray[7];
char Gmtarray[8];
char phone_number[]="+xxxxxxxxxx"; //SMS number
char server[ ]="smtp.mail.yahoo.com"; //EMAIL array variables
char user_name[ ]="username";
char password[ ]="password";
char port[ ]="25";
char sender_add[ ]="username@yahoo.gr"; //information about sender, directions and names
char sender_name[ ]="GPS device"; //name of sender
char to_add[ ]="user@gmail.com"; //receiver email address
char to_name[ ]="name"; //receiver name
char subject[ ]="Geographical position from tracking device"; //email subject

float latsec;
float longsec;

String NorthSouth; //GPS string variables
```

String EastWest;
String Latitude1;
String Decimlat;
String Longitude1;
String Decimlong;
String Date;
String Gmt;

Στο παραπάνω κομμάτι του κώδικα γίνεται ο ορισμός του τύπου των μεταβλητών και η αρχικοποίηση των τιμών τους.

```
void switchModule() { //power on shield function  
    digitalWrite(onModulePin,HIGH);  
    delay(3000);  
    digitalWrite(onModulePin,LOW);  
}
```

Στο παραπάνω μέρος του κώδικα δημιουργείται η συνάρτηση switchModule η οποία ενεργοποιεί την πρόσθετη πλακέτα shield,

```
void pregps() { //GPS initialization function  
    Serial.println("Pregps started");  
    Serial1.println("AT+CGSOCKCONT=1,\"IP\", \"internet\");  
    Serial1.flush();  
    x=0;  
    do{  
        while(Serial1.available()==0);  
        answer[x]=Serial1.read();  
        Serial.print(answer[x]);  
        x++;  
    }  
    while(!(answer[x-1]=='K'&&answer[x-2]=='O'));  
    delay(500);  
    Serial1.println("AT+CGPSURL=\"supl.google.com:7276\");  
    Serial1.flush();  
    x=0;  
    do{
```

```
while(Serial1.available()==0);
answer[x]=Serial1.read();
x++;
}
while(!(answer[x-1]=='K'&&answer[x-2]=='O'));
delay(500);
Serial1.println("AT+CGPSSSL=0");
Serial1.flush();
x=0;
do{
  while(Serial1.available()==0);
  answer[x]=Serial1.read();
  x++;
}
while(!(answer[x-1]=='K'&&answer[x-2]=='O'));
delay(500);
Serial1.println("AT+CGPS=1,2");
Serial1.flush();
x=0;
do{
  while(Serial1.available()==0);
  answer[x]=Serial1.read();
  x++;
}
while(!(answer[x-1]=='K'&&answer[x-2]=='O'));
}
```

Στο παραπάνω κομμάτι του κώδικα δημιουργείται η συνάρτηση pregps η οποία είναι υπεύθυνη την αποστολή των κατάλληλων εντολών στο shield, έτσι ώστε να ξεκινήσει η διαδικασία λήψης πληροφοριών μέσω του δέκτη GPS.

```
void getgps(){ //receive GPS data function
  int found_coo=0;
  int thesi=0;
  int lastStringLength=0;
  String gpsresponse;
```

```
while(found_coo < 1){
  delay(10000);
  while (found_coo < 1)
  {
    x=0;
    for (x=0;x<255;x++){
      dataIn[x]='\0';
    }
    delay(100);
    Serial1.println("AT+CGPSINFO");
    Serial1.flush();
    x=0;
    do{
      while(Serial1.available()==0);
      dataIn[x]=Serial1.read();
      x++;
    }
    while(!(dataIn[x-1]=='K'&&dataIn[x-2]=='O'));
    delay(100);
    Serial.println("Here is the response from gps-->");
    Serial.println(dataIn);
    Serial.println("End of the response.");
    gpsresponse = dataIn;
    // I use a string to extract substring that are GPS value
    Serial.print("gpsresponse: ");
    Serial.println(gpsresponse);
    lastStringLength = gpsresponse.length();
    Serial.print("Length:");
    Serial.println(lastStringLength);
    thesi = gpsresponse.lastIndexOf('+CGPSINFO:');
    Serial.print("Thesi:");
    Serial.println(thesi);
    Latitude1 = gpsresponse.substring((thesi+1),(thesi+12));
    Latitude1.toCharArray(Lat,12);
    Decimlat = gpsresponse.substring((thesi+6),(thesi+12));
    Decimlat.toCharArray(Latdecimal,9);
    float latsec= atof(Latdecimal);
    latsec=(float)latsec/1000000;
```



```
latsec=(float)latsec*60;
dtostrf(latsec, 5, 2, Latsecond);
NorthSouth = gpsresponse.substring((thesi+13),(thesi+14));
NorthSouth.toCharArray(Latsymbol,2);
Longitude1 = gpsresponse.substring((thesi+15),(thesi+27));
Longitude1.toCharArray(Long,13);
Decimlong = gpsresponse.substring((thesi+21),(thesi+27));
Decimlong.toCharArray(Longdecimal,10);
float longsec= atof(Longdecimal);
longsec=(float)longsec/1000000;
longsec=(float)longsec*60;
dtostrf(longsec, 5, 2, Longsecond);
EastWest = gpsresponse.substring((thesi+28),(thesi+29));
EastWest.toCharArray(Longsymbol,2);
Date= gpsresponse.substring((thesi+30),(thesi+36));
Date.toCharArray(Datearray,7);
Gmt = gpsresponse.substring((thesi+37),(thesi+43));
Gmt.toCharArray(Gmtarray,7);

if (lastStringLength > 84) {
    found_coo++;
    break;
}
}
delay(5000);
}
found_coo=0;
}
```

Παραπάνω δημιουργείται η συνάρτηση `getgps`, η οποία μέσω κατάλληλων εντολών του `shield`, λαμβάνει τα δεδομένα από το δέκτη GPS, τα επεξεργάζεται και τα αποθηκεύει σε συγκεκριμένες μεταβλητές.

```
void hour(){ //time correction function
if((Gmtarray[0]!='0') && (Gmtarray[1]!='7')){
    Gmtarray[0]='0';
    Gmtarray[1]='9';
}
```

```
}  
else if((Gmtarray[0]=='0') && (Gmtarray[1]=='8')){  
    Gmtarray[0]='1';  
    Gmtarray[1]='0';  
}  
else if((Gmtarray[0]=='0') && (Gmtarray[1]=='9')){  
    Gmtarray[0]='1';  
    Gmtarray[1]='1';  
}  
else if((Gmtarray[0]=='1') && (Gmtarray[1]=='7')){  
    Gmtarray[0]='1';  
    Gmtarray[1]='9';  
}  
else if((Gmtarray[0]=='1') && (Gmtarray[1]=='8')){  
    Gmtarray[0]='2';  
    Gmtarray[1]='0';  
}  
else if((Gmtarray[0]=='1') && (Gmtarray[1]=='9')){  
    Gmtarray[0]='2';  
    Gmtarray[1]='1';  
}  
else if((Gmtarray[0]=='2') && (Gmtarray[1]=='1')){  
    Gmtarray[0]='2';  
    Gmtarray[1]='3';  
}  
else if((Gmtarray[0]=='2') && (Gmtarray[1]=='2')){  
    Gmtarray[0]='0';  
    Gmtarray[1]='0';  
}  
else if((Gmtarray[0]=='2') && (Gmtarray[1]=='3')){  
    Gmtarray[0]='0';  
    Gmtarray[1]='1';  
}  
else{  
    Gmtarray[1]=Gmtarray[1]+2;  
}  
}
```

Στο παραπάνω κομμάτι κώδικα αναπτύσσεται η συνάρτηση hour, μέσα στην οποία μετατρέπεται η ώρα από UTC σε UTC+2. (Ωρα Ελλάδος)

```
void email_text() { //send email function
  Serial1.print("AT+SMTPSRV=\"");
  Serial1.print(server);
  Serial1.print("\",");
  Serial1.println(port);
  Serial1.flush();
  x=0;
  do{
    while(Serial1.available()==0);
    answer[x]=Serial1.read();
    x++;
  }
  while(!(answer[x-1]=='K' && answer[x-2]=='O'));
  delay(1000);
  Serial1.print("AT+SMTPAUTH=1,\"");
  Serial1.print(user_name);
  Serial1.print("\",\"");
  Serial1.print(password);
  Serial1.println("\"");
  Serial1.flush();
  x=0;
  do{
    while(Serial1.available()==0);
    answer[x]=Serial1.read();
    x++;
  }
  while(!(answer[x-1]=='K' && answer[x-2]=='O'));
  delay(1000);
  Serial1.print("AT+SMTPFROM=\"");
  Serial1.print(sender_add);
  Serial1.print("\",\"");
  Serial1.print(sender_name);
  Serial1.println("\"");
  Serial1.flush();
```

```
x=0;
do{
  while(Serial1.available()==0);
  answer[x]=Serial1.read();
  x++;
}
while(!(answer[x-1]=='K'&&answer[x-2]=='O'));
delay(1000);
Serial1.print("AT+SMTPRCPT=0,0,");
Serial1.print(to_add);
Serial1.print("\",");
Serial1.print(to_name);
Serial1.println("");
Serial1.flush();
x=0;
do{
  while(Serial1.available()==0);
  answer[x]=Serial1.read();
  x++;
}
while(!(answer[x-1]=='K'&&answer[x-2]=='O'));
delay(1000);
Serial1.print("AT+SMTPSUB=");
Serial1.print(subject);
Serial1.println("");
Serial1.flush();
x=0;
do{
  while(Serial1.available()==0);
  answer[x]=Serial1.read();
  x++;
}
while(!(answer[x-1]=='K'&&answer[x-2]=='O'));
delay(1000);
Serial1.flush();
Serial1.println("AT+SMTPBODY");
Serial1.flush();
x=0;
```

```
do{
    while(Serial1.available()>0);
    answer[x]=Serial1.read();
    x++;
}
while(!(answer[x-1]=='>&&answer[x-2]=='>'));
Serial1.write("Position of Device:");
Serial1.write(0x0A);
Serial1.write(Latitude1[0]);
Serial1.write(Latitude1[1]);
Serial1.write("°");
Serial1.write(Latitude1[2]);
Serial1.write(Latitude1[3]);
Serial1.write("");
Serial1.write(Latsecond);
Serial1.write("");
Serial1.write(Latsymbol);
Serial1.write(",");
Serial1.write(Longitude1[0]);
Serial1.write(Longitude1[1]);
Serial1.write(Longitude1[2]);
Serial1.write("°");
Serial1.write(Longitude1[3]);
Serial1.write(Longitude1[4]);
Serial1.write("");
Serial1.write(Longsecond);
Serial1.write("");
Serial1.write(Longsymbol);
Serial1.write(0x0A);
Serial1.write("Date:");
Serial1.write(Datearray[0]);
Serial1.write(Datearray[1]);
Serial1.write("/");
Serial1.write(Datearray[2]);
Serial1.write(Datearray[3]);
Serial1.write("/");
Serial1.write(Datearray[4]);
Serial1.write(Datearray[5]);
```

```
Serial1.write(0x0A);
Serial1.write("UTC+2:00 Time: ");
Serial1.write(Gmtarray[0]);
Serial1.write(Gmtarray[1]);
Serial1.write(":");
Serial1.write(Gmtarray[2]);
Serial1.write(Gmtarray[3]);
Serial1.write(":");
Serial1.write(Gmtarray[4]);
Serial1.write(Gmtarray[5]);
Serial1.write(0x0A);
Serial1.write(0x1A);
Serial1.println("AT+CGSOCKCONT=1,\"IP\", \"internet\");
Serial1.flush();
x=0;
do{
    while(Serial1.available()==0);
    answer[x]=Serial1.read();
    Serial.print(answer[x]);
    x++;
}
while(!(answer[x-1]=='K'&&answer[x-2]=='O'));
Serial1.println("AT+SMTPSEND");
Serial1.flush();
x=0;
do{
    while(Serial1.available()==0);
    answer[x]=Serial1.read();
    x++;
}
while(!(answer[x-1]=='K'&&answer[x-2]=='O'));
Serial.println("Sending email");
x=0;
do{
    do{
        digitalWrite(led,HIGH);
    }
    while(Serial1.available()==0);
```

```
digitalWrite(led,LOW);
answer[x]=Serial1.read();
x++;
}
while(!(answer[x-1]=='S'&&answer[x-2]=='S'));
Serial.println("Email sended successfully");
}
```

Στο παραπάνω μέρος του κώδικα δημιουργείται η συνάρτηση `email_text`. Η συνάρτηση δημιουργήθηκε για να αποστέλλει τα δεδομένα μέσω email, στην ηλεκτρονική διεύθυνση του χρήστη, σε μορφή μοιρών, λεπτών και δευτερολέπτων, χρησιμοποιώντας τις κατάλληλες εντολές του `shield`. Προσθέτει επίσης ώρα και ημερομηνία.

```
void sms(){                                     //send sms function
  Serial.println("Sending sms");
  Serial1.println("AT+CMGF=1");
  delay(1000);
  Serial1.print("AT+CMGS=\"");
  Serial1.print(phone_number);
  Serial1.println("\");
  delay(1000);
  Serial1.println("Position of Device:");
  Serial1.print(Latitude1[0]);
  Serial1.print(Latitude1[1]);
  Serial1.print("o");
  Serial1.print(Latitude1[2]);
  Serial1.print(Latitude1[3]);
  Serial1.print("");
  Serial1.print(Latsecond);
  Serial1.print("");
  Serial1.print(Latsymbol);
  Serial1.print(",");
  Serial1.print(Longitude1[0]);
  Serial1.print(Longitude1[1]);
  Serial1.print(Longitude1[2]);
  Serial1.print("o");
  Serial1.print(Longitude1[3]);
```

```
Serial1.print(Longitude1[4]);
Serial1.print("");
Serial1.print(Longsecond);
Serial1.print("");
Serial1.println(Longsymbol);
Serial1.print("Date: ");
Serial1.print(Datearray[0]);
Serial1.print(Datearray[1]);
Serial1.print("/");
Serial1.print(Datearray[2]);
Serial1.print(Datearray[3]);
Serial1.print("/");
Serial1.print(Datearray[4]);
Serial1.println(Datearray[5]);
Serial1.print("UTC+2:00 Time: ");
Serial1.print(Gmtarray[0]);
Serial1.print(Gmtarray[1]);
Serial1.print(":");
Serial1.print(Gmtarray[2]);
Serial1.print(Gmtarray[3]);
Serial1.print(":");
Serial1.print(Gmtarray[4]);
Serial1.print(Gmtarray[5]);
Serial1.write(0x1A);
Serial1.write(0x0D);
Serial1.write(0x0A);
delay(5000);
Serial.println("Sms sended successfully");
}
```

Στο παραπάνω κομμάτι του κώδικα αναπτύσσεται η συνάρτηση sms η οποία ενσωματώνει όλα τα δεδομένα (γεωγραφικό μήκος και πλάτος, ώρα, ημερομηνία) σε ένα κείμενο και μέσω της κατάλληλης εντολής και το αποστέλλει σε αριθμό κινητού, το οποίο έχει οριστεί από τον χρήστη.

```
void primary() { //operation function
  Serial.println("Primary Started");
```



```
digitalWrite(ledg, HIGH);  
getgps();  
hour();  
delay(100);  
sms();  
delay(200);  
email_text();  
digitalWrite(ledg, LOW);  
delay(6000000);  
}
```

Παραπάνω δημιουργείται η συνάρτηση `primary`, στην οποία έχουν ενσωματωθεί και καλούνται με τη σειρά οι προαναφερθείσες βασικές συναρτήσεις. Στο τέλος της `primary` υπάρχει μια καθυστέρηση 10 λεπτών μέχρι να ξεκινήσει και πάλι η λειτουργία της συσκευής.

```
void breakint(){  
  w++;  
}
```

Παραπάνω δημιουργείται η συνάρτηση `breakint` η οποία περιέχει έναν μετρητή. Με την αύξηση του μετρητή συνεχίζεται η ροή του προγράμματος αφού δεχθεί εξωτερική διακοπή.

```
void setup() { //Arduino setup function  
  pinMode(led, OUTPUT);  
  pinMode(ledg, OUTPUT);  
  pinMode(ledb, OUTPUT);  
  pinMode(onModulePin, OUTPUT);  
  delay(4000);  
  attachInterrupt(0, breakint, CHANGE);  
  while(w == 0){  
    digitalWrite(ledb, HIGH);  
    delay(1000);  
    digitalWrite(ledb, LOW);  
    delay(1000);  
    Serial.println("Wait for interrupt");  
  }  
}
```

```
digitalWrite(ledb, HIGH);  
Serial1.begin(115200);  
switchModule();  
delay(25000);  
pregps();  
Serial.println("Setup ended");  
}
```

Στο παραπάνω κομμάτι κώδικα καλείται η βασική συνάρτηση του Arduino, `setup`, η οποία ενσωματώνει την διακοπή που χρησιμοποιείται για να ξεκινήσει η λειτουργία της συσκευής. Λειτουργεί δηλαδή σαν ανιχνευτής κίνησης.

```
void loop(){ //Arduino loop function  
  primary();  
}
```

Στο παραπάνω μέρος κώδικα παρουσιάζεται η συνάρτηση `loop` του Arduino, η οποία επαναλαμβάνεται συνεχώς. Στην `loop` ενσωματώνεται μόνο η συνάρτηση `primary`.



6 ΣΥΜΠΕΡΑΣΜΑΤΑ

6.1 ΑΠΟΤΕΛΕΣΜΑΤΑ ΛΕΙΤΟΥΡΓΙΑΣ ΚΑΤΑΣΚΕΥΗΣ

Το τελικό αποτέλεσμα της λειτουργίας της κατασκευής μας θεωρούμε ότι είναι ιδιαίτερα επιτυχημένο καθώς οι επιθυμητοί μας στόχοι υλοποιήθηκαν. Καταφέραμε να αναπτύξουμε τον κατάλληλο κώδικα έτσι ώστε να επιτύχουμε την επιθυμητή λειτουργία της κατασκευής μας, η οποία είναι:

- i. Ανίχνευση τυχόν μεταβολή της θέσης της
- ii. Εντοπισμός του γεωγραφικού στίγματός της κάνοντας χρήση του συστήματος GPS
- iii. Συλλογή και επεξεργασία πληροφοριών (γεωγραφικό πλάτος και μήκος, ώρα, ημερομηνία) και
- iv. Αποστολή τους προς τον χρήστη χρησιμοποιώντας τις τεχνολογίες του email και του sms

Όσον αφορά στο κατασκευαστικό κομμάτι της εργασίας μας, καταφέραμε να ενσωματώσουμε όλα τα απαραίτητα ηλεκτρονικά και μηχανολογικά εξαρτήματα που απαρτίζουν την συσκευή μας σε μια αρκετά μικρή σε όγκο συσκευασία. Το παραπάνω χαρακτηριστικό της δίνει την δυνατότητα να τοποθετηθεί σε οποιοδήποτε αντικείμενο επιθυμεί ο χρήστης για να εκμεταλλευθεί των λειτουργιών της.

Κατά την διάρκεια των πειραμάτων που πραγματοποιήθηκαν για τον έλεγχο της σωστής λειτουργίας της συσκευής παρατηρήθηκαν ελάχιστες δυσλειτουργίες. Η περισσότερο σημαντική δυσλειτουργία είναι ότι ο διακόπτης tilt, ο οποίος ανιχνεύει την αλλαγή της αρχικής θέσης της συσκευής, εμφανίζει κάποιες φορές αυξημένη ευαισθησία σε κραδασμούς που δύναται να δεχθεί η συσκευή. Το παραπάνω γεγονός μπορεί να οδηγήσει σε λανθασμένη ενεργοποίηση της συσκευής και εσφαλμένη ενημέρωση του χρήστη της.

Μια ακόμα δυσλειτουργία της συσκευής μπορεί να θεωρηθεί η μειωμένη λειτουργικότητα της σε κλειστούς χώρους. Στην παραπάνω περίπτωση η λήψη σήματος από το σύστημα GPS ενδέχεται να είναι μειωμένη με αποτέλεσμα την αδυναμία εντοπισμού της γεωγραφικής θέσης της συσκευής. Επίσης υπάρχει το ενδεχόμενο η συσκευή να βρεθεί σε χώρο όπου δεν υπάρχει κάλυψη από το δίκτυο κινητής τηλεφωνίας που χρησιμοποιεί. Οι δυο παραπάνω καταστάσεις οδηγούν στην πλήρη απώλεια λειτουργικότητας της συσκευής μας, η οποία όμως οφείλεται σε μη ελέγξιμες παραμέτρους της λειτουργίας της.

6.2 ΠΡΟΤΕΙΝΟΜΕΝΕΣ ΒΕΛΤΙΩΣΕΙΣ

Στην ενότητα αυτή θα αναφερθούμε σε πιθανές προτάσεις βελτίωσης και επέκτασης των δυνατοτήτων της συσκευής που εντοπίσαμε στη διάρκεια των προσπαθειών υλοποίησης της συγκεκριμένης εργασίας. Οι παρακάτω προτάσεις αφορούν στην πιο αξιόπιστη λειτουργία της συσκευής και στην πληρέστερη πληροφόρηση του χρήστη.

Όπως αναφέρθηκε στην προηγούμενη ενότητα ο διακόπτης tilt που χρησιμοποιούμε για τον εντοπισμό της κίνησης συσκευής μπορεί να παρουσιάσει μειωμένη απόδοση. Ως βελτίωση σε αυτό προτείνουμε την αντικατάστασή του με ένα ψηφιακό επιταχυνσιόμετρο. Το εν λόγω εξάρτημα θα εξαλείψει το πρόβλημα της αυξημένης ευαισθησίας σε κραδασμούς κατά τον έλεγχο της αλλαγής θέσης που εμφανίζεται με την χρήση του διακόπτη tilt. Επομένως θα επιτευχθεί πιο αξιόπιστος έλεγχος στην ανίχνευση της κίνησης με αποτέλεσμα την ελαχιστοποίηση της εσφαλμένης ενεργοποίησης της συσκευής.

Η παρούσα κατασκευή αναπτύσσεται κάνοντας χρήση αποκλειστικά μονόδρομης επικοινωνίας από την συσκευή προς τον χρήστη. Η προσθήκη αμφίδρομης επικοινωνίας μεταξύ χρήστη και συσκευής θα προσδώσει καλύτερο έλεγχο της λειτουργίας της και ευκολότερο και πληρέστερο χειρισμό της από τον χρήστη. Κάνοντας χρήση της υπηρεσίας του sms που ήδη υποστηρίζει η συσκευή ο χρήστης έχει τη δυνατότητα να λάβει κατ' απαίτηση ενημέρωση από την συσκευή καθώς επίσης και να ελέγξει τις επιμέρους λειτουργίες της. Ένα απλό παράδειγμα χρήσης αμφίδρομης επικοινωνίας είναι η αποστολή σύντομου γραπτού μηνύματος (sms) από τον χρήστη προς την συσκευή ζητώντας ενημέρωση σχετικά με τη θέση της ανεξαρτήτως της προκαθορισμένης λειτουργίας της.

Επιπροσθέτως χρησιμοποιώντας την τεχνολογία του internet θα μπορούσε ο χρήστης να παρακολουθεί το στίγμα της συσκευής, κάθε φορά που λαμβάνει ενημέρωση, σε ιστοσελίδα η οποία θα το προβάλλει γραφικά αξιοποιώντας την υπηρεσία του google maps. Η παραπάνω πρόταση μπορεί να επιτευχθεί στέλνοντας η συσκευή δεδομένα μέσω των πρωτοκόλλων του internet (http, tcp, ftp) στον server που χρησιμοποιεί η ιστοσελίδα. Με αυτό τον τρόπο επιτυγχάνουμε οπτικοποίηση της θέσης της συσκευής πάνω σε χάρτη για ευκολότερη κατανόησή της.

Για οπτικοποίηση όμως της θέσης δεν κρίνεται απαραίτητη μια οθόνη υπολογιστή. Θα μπορούσε κάλλιστα να χρησιμοποιηθεί μια οθόνη κινητού τελευταίας τεχνολογίας που θα προβάλλει το στίγμα της συσκευής μέσω μιας εφαρμογής. Η συγκεκριμένη εφαρμογή θα αναπτυχθεί για χρήση στις πιο δημοφιλείς πλατφόρμες λογισμικού «έξυπνων» κινητών τηλεφώνων. Θα υπάρχει άμεση επικοινωνία του κινητού με την συσκευή έτσι ώστε να μπορεί ανά πάσα στιγμή ο χρήστης να ενημερωθεί για την κατάσταση της συσκευής δια μέσου της εφαρμογής. Επίσης θα είναι δυνατόν ο χρήστης να μπορεί να χειριστεί τις λειτουργίες της συσκευής.

Βάσει όλων αυτών των προτάσεων μπορεί να βελτιωθεί κατά πολύ η ευκολία χρήσης της συσκευής και η επικοινωνία του χρήστη με τη συσκευή. Οι περισσότερες από τις παραπάνω προτάσεις μπορούν να επιτευχθούν με σχετικά μικρό κόστος αφού δεν απαιτούν την αγορά επιπλέον υλικού αλλά κυρίως την ανάπτυξη λογισμικού.

ΒΙΒΛΙΟΓΡΑΦΙΑ

ΒΙΒΛΙΑ

- [1]: Ahmed El-Rabbany «Introduction to GPS The Global Positioning System» Artech House 2002
- [2]: Elliott D. Kaplan, Christopher Hegarty «Understanding GPS principles and applications Second Edition» Artech House 2005
- [3]: Langley, R. B., «The Orbits of GPS Satellites» GPS World Vol. 2, No. 3 March 1991
- [4]: <http://www.gps.gov/systems/gps/control/>
- [5]: Asha Mehrotra «GSM System Engineering» Artech House 1997
- [6]: Siegmund M. Redl, Matthias K. Weber, Malcolm W. Oliphant «GSM and Personal Communications Handbook» Artech House 1998
- [7]: Juha Korhonen «Introduction to 3G Mobile Communications, Second Edition» Artech House 2003

ΙΣΤΟΣΕΛΙΔΕΣ

- [8]: <http://electronicslab.eu/el/psifiaka-hlektronika-2/149-oloklhromena-kyklomata.html>
- [9]: <http://el.wikipedia.org/wiki/%CE%9C%CE%B9%CE%BA%CF%81%CE%BF%CE%B5%CE%BB%CE%B5%CE%B3%CE%BA%CF%84%CE%AE%CF%82>
- [10]: <http://el.wikipedia.org/wiki/%CE%9C%CE%B9%CE%BA%CF%81%CE%BF%CE%B5%CE%BB%CE%B5%CE%B3%CE%BA%CF%84%CE%AE%CF%82>

[11]:

<http://el.wikipedia.org/wiki/%CE%9C%CE%B9%CE%BA%CF%81%CE%BF%CE%B5%CE%BB%CE%B5%CE%B3%CE%BA%CF%84%CE%AE%CF%82>

[12]: <http://el.wikipedia.org/wiki/Arduino>

[13]: <https://learn.sparkfun.com/tutorials/what-is-an-arduino/the-arduino-family>

[14]: <https://learn.sparkfun.com/tutorials/what-is-an-arduino/whats-on-the-board>

[15]: <http://arduino.cc/en/Main/ArduinoBoardLeonardo>

[16]: <http://www.pighixxx.com/2013/03/arduino-leonardo-pinout-v2/>

[17]: <http://www.cooking-hacks.com/shop/arduino/shields>

[18]: <http://wm.sim.com/producten.aspx?id=1026>

[19]: <http://www.cooking-hacks.com/documentation/tutorials/arduino-3g-gprs-gsm-gps#step14>

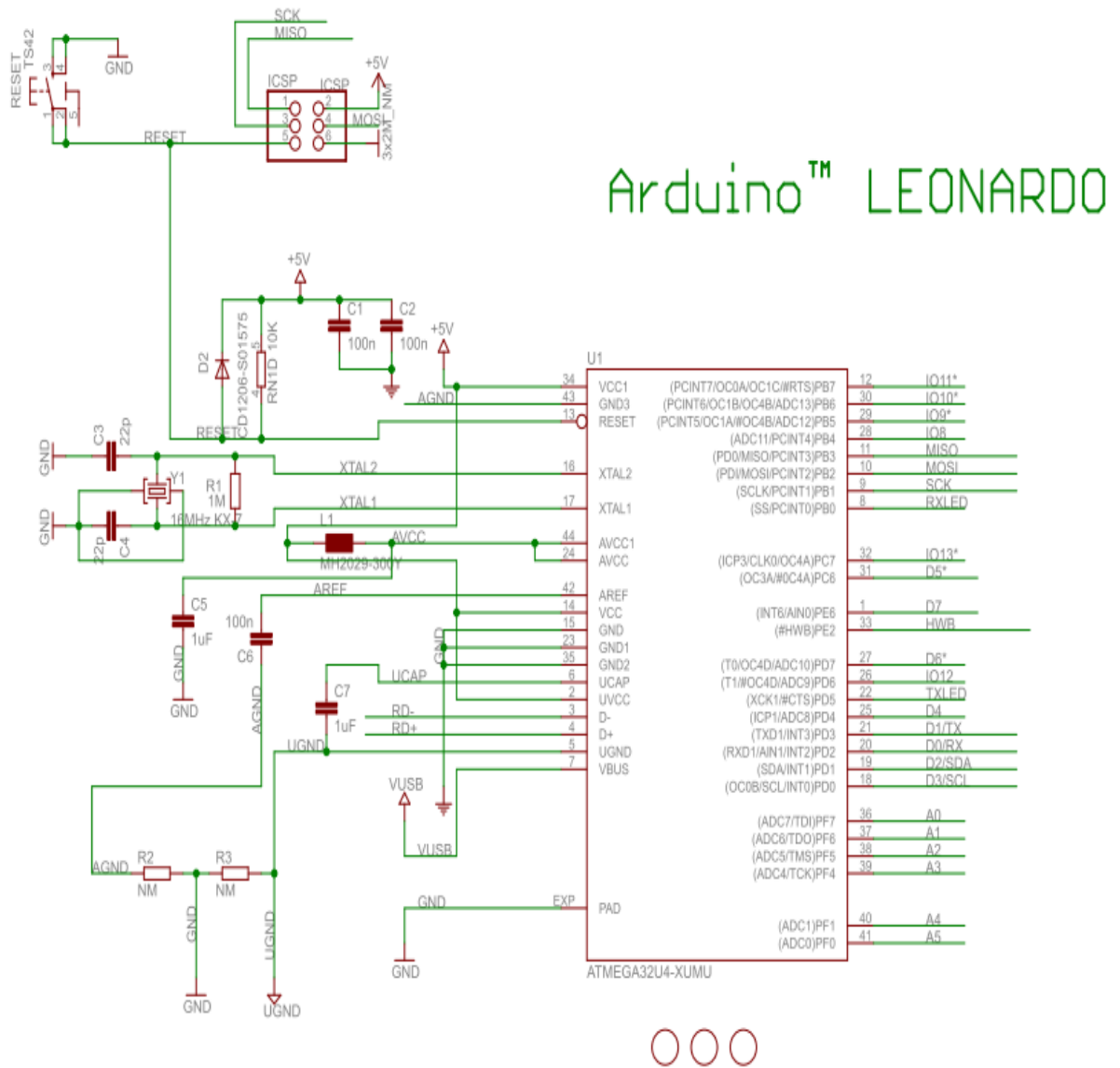
ΠΑΡΑΡΤΗΜΑ

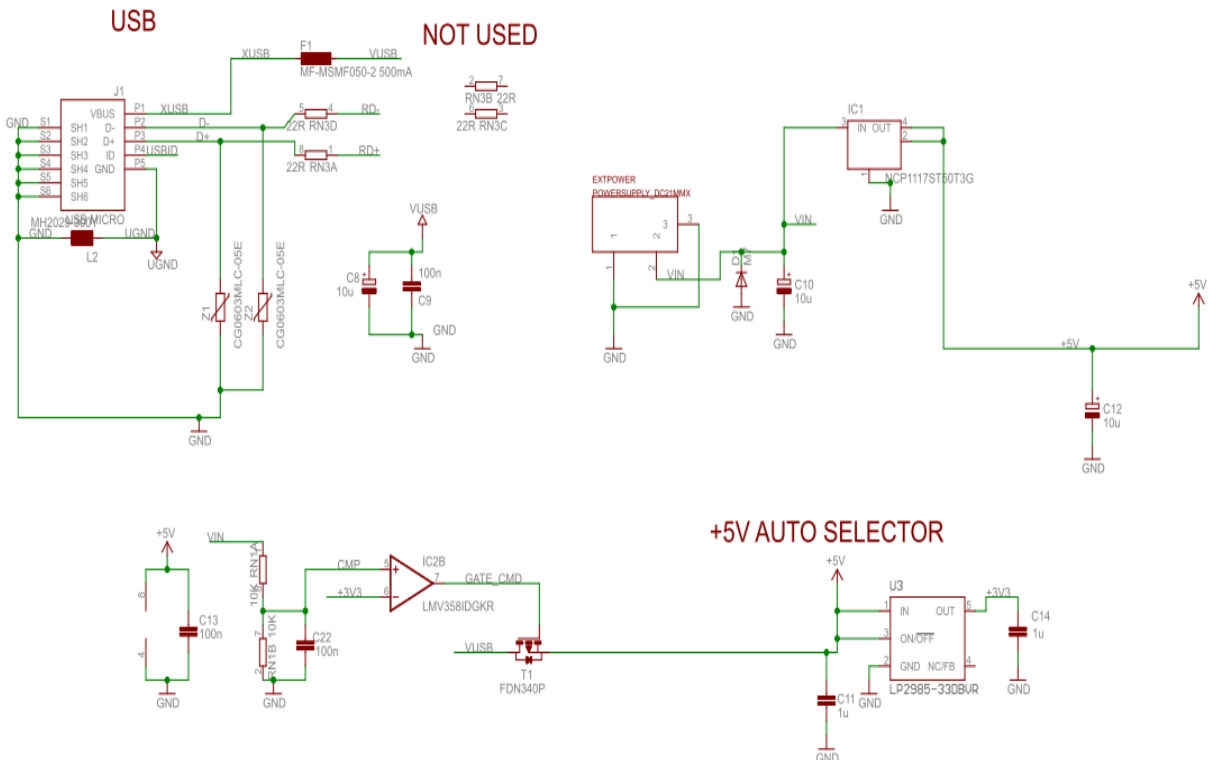
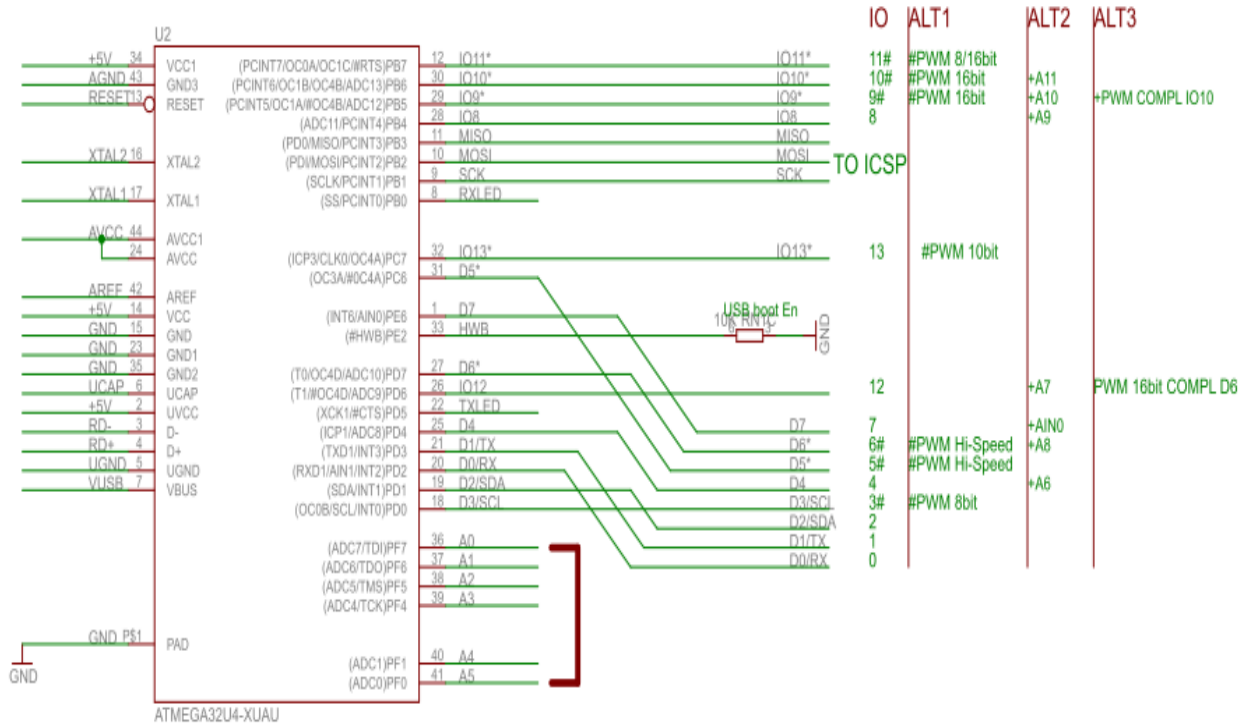
ι). ΛΙΣΤΑ ΥΛΙΚΩΝ ΚΑΤΑΣΚΕΥΗΣ

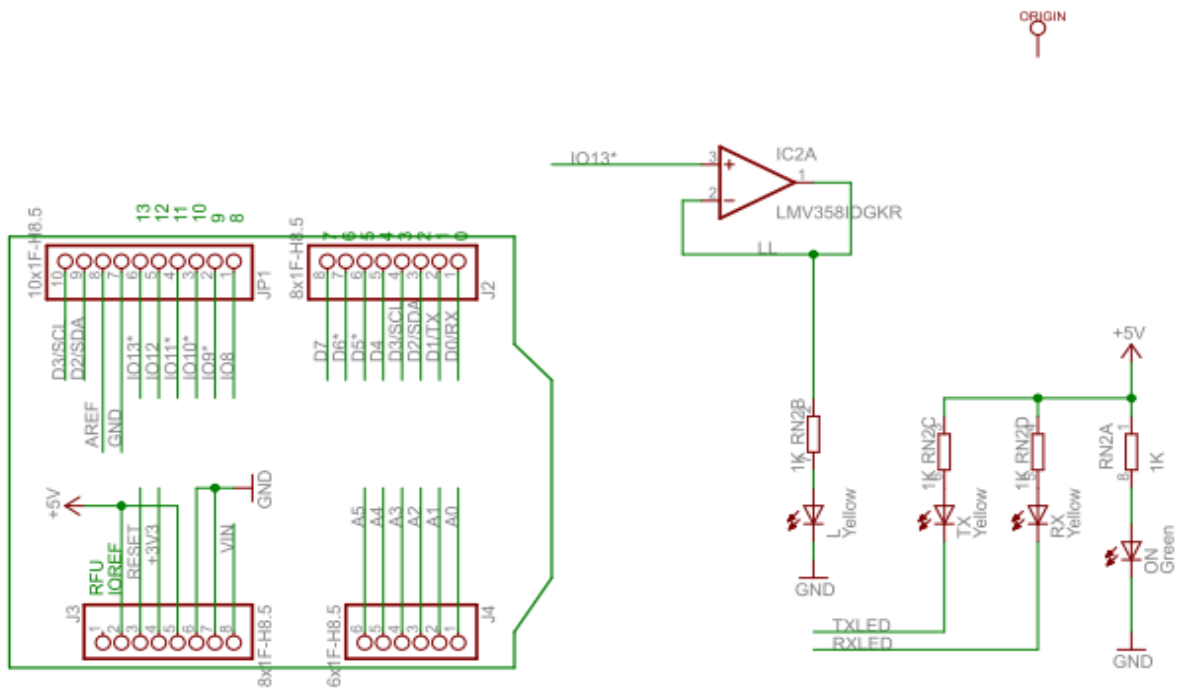
ΥΛΙΚΟ	ΤΕΜΑΧΙΑ	ΚΟΣΤΟΣ (€)
Εξωτερική θήκη	1	19,7
Αδιάβροχη υποδοχή USB με καπάκι	1	13
Υποδοχή εξωτερικής τροφοδοσίας πλακέτας	1	3,2
Διακόπτης ON/OFF	1	0,4
LED κόκκινο	1	0,7
LED πράσινο	1	0,7
LED μπλε	1	0,7
Βάση στήριξης LED	3	1,2
Πλακέτα Arduino Leonardo	1	22,2
Πλακέτα 3G/GPRS + GPS	1	221,4
Pins σύνδεσης Arduino με Shield	9	0,45
Αρσενικό βύσμα micro- usb	1	3
Καλώδιο σύνδεσης βύσματος USB με εξωτερική υποδοχή USB	1	1,4
Διακόπτης tilt	1	3
Εσωτερική κεραία GPS	1	9,8
Εσωτερική κεραία GSM	1	7,4
Σύνολο	17	308,25

ii). ΣΧΗΜΑΤΙΚΑ ΔΙΑΓΡΑΜΜΑΤΑ ΗΛΕΚΤΡΟΝΙΚΩΝ ΣΤΟΙΧΕΙΩΝ

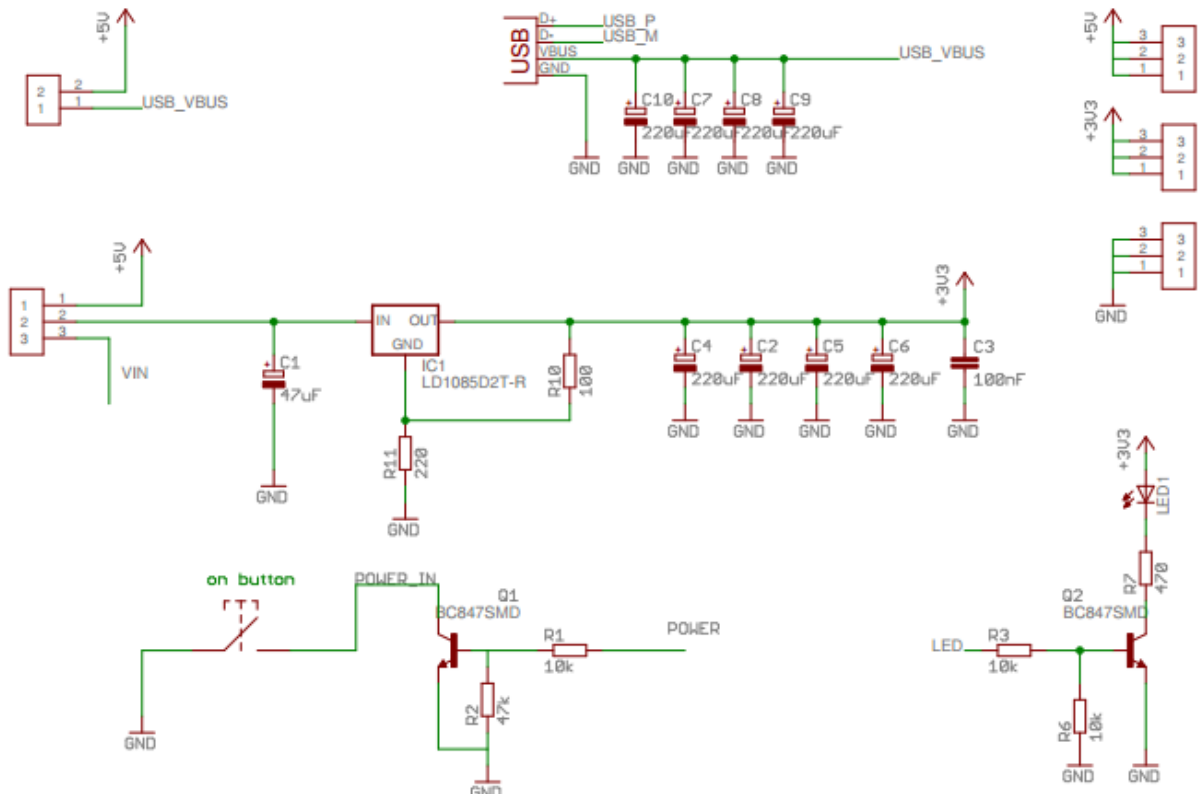
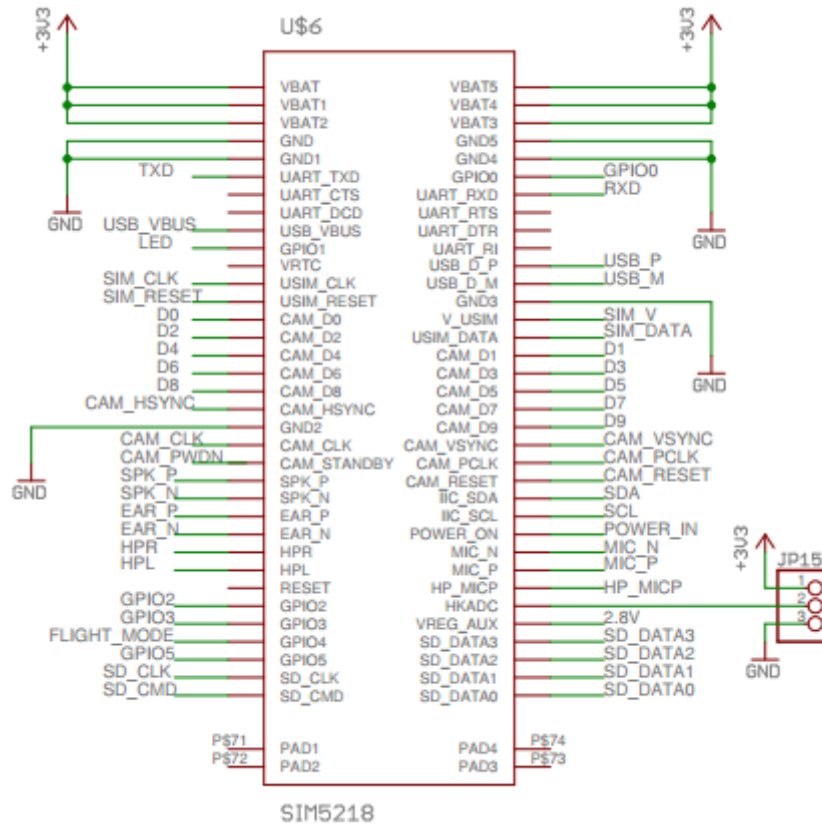
Παρουσίαση σχηματικών της πλακέτας Arduino Leonardo

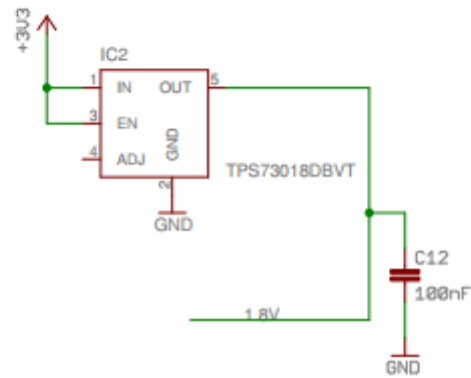
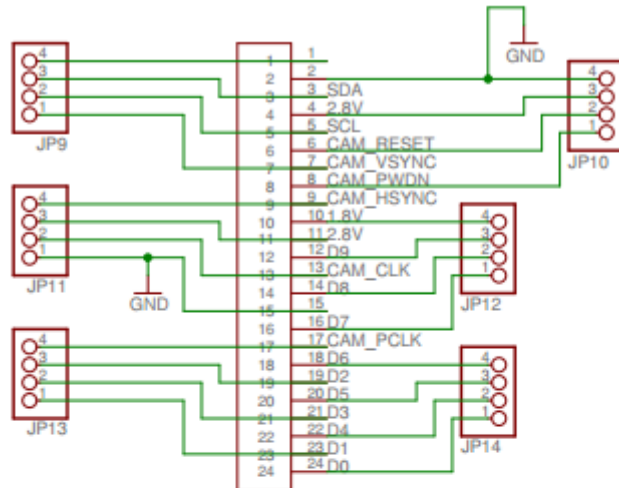
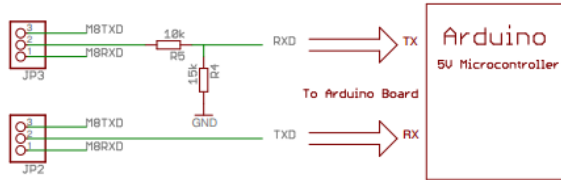
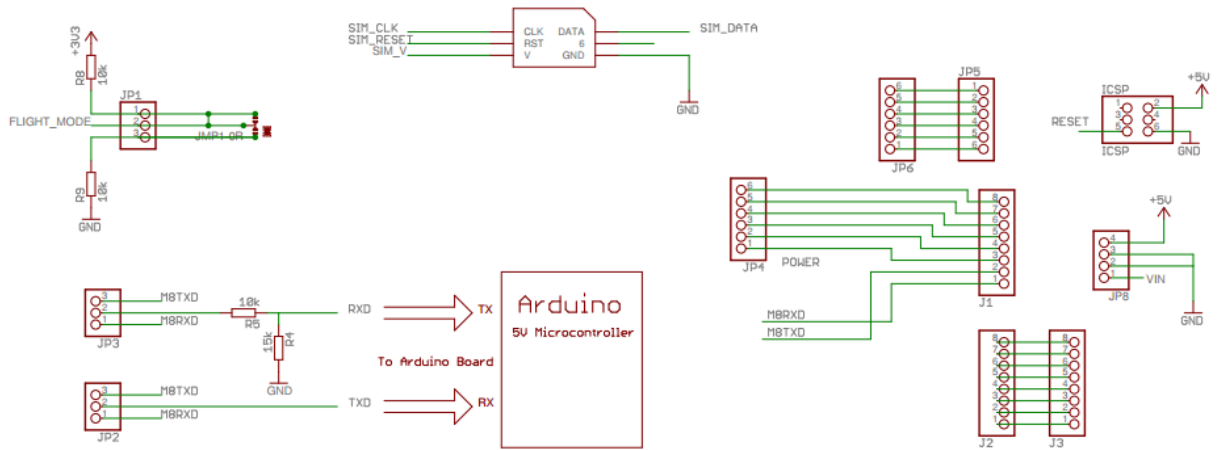


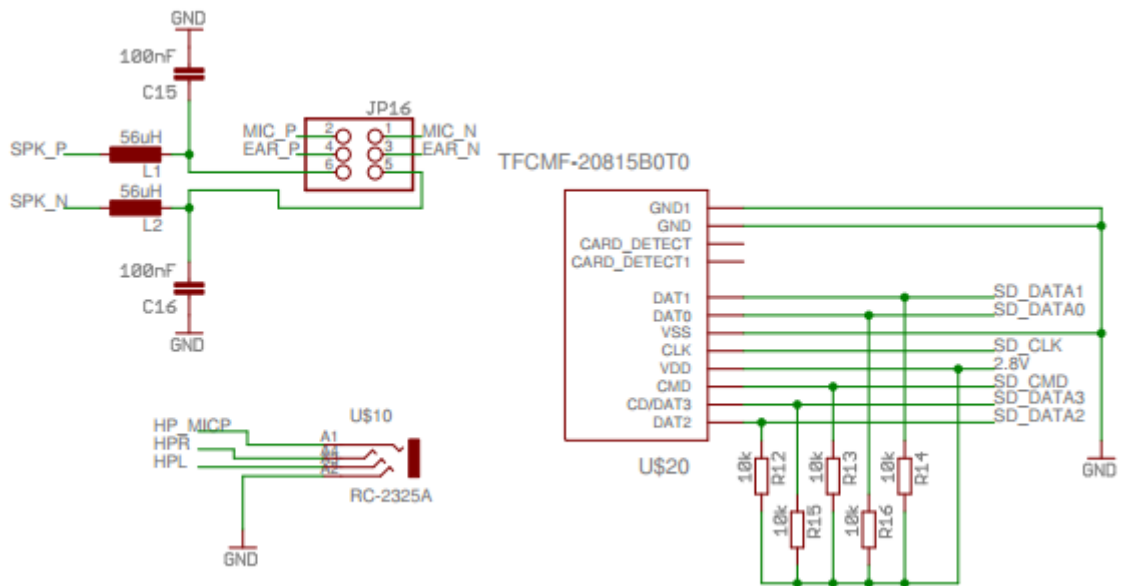




Παρουσίαση σχηματικών της πρόσθετης πλακέτας 3G/GPRS + GPS







iii). ΚΩΔΙΚΑΣ ΥΛΟΠΟΙΗΣΗΣ ΠΡΟΓΡΑΜΜΑΤΟΣ ARDUINO

```

int led = 13;           //shield led
int ledb = 12;         //blue led
int ledg = 11;         //green led

int onModulePin = 2;   //power on the modulepin
int x=0;
int w=0;

char dataIn[255];      //GPS array variables
char answer[255];
char Lat[12];
char Latdecimal[9];
char Latsecond[9];
char Long[13];
char Longdecimal[9];
char Longsecond[9];
char Latsymbol[2];
char Longsymbol[2];
char Datearray[7];
char Gmtarray[8];
    
```



```
char phone_number[]="+xxxxxxxxx"; //SMS number
char server[]="smtp.mail.yahoo.com"; //EMAIL array variables
char user_name[]="username";
char password[]="password";
char port[]="25";
char sender_add[]="username@yahoo.gr"; //information about sender, directions and names
char sender_name[]="GPS device"; //name of sender
char to_add[]="user@gmail.com"; //receiver email address
char to_name[]="name"; //receiver name
char subject[]="Geographical position from tracking device"; //email subject

float latsec;
float longsec;

String NorthSouth; //GPS string variables
String EastWest;
String Latitude1;
String Decimlat;
String Longitude1;
String Decimlong;
String Date;
String Gmt;

void switchModule() //power on shield function
{
    digitalWrite(onModulePin,HIGH);
    delay(3000);
    digitalWrite(onModulePin,LOW);
}

void pregps() //GPS initialization function
{
    Serial.println("Pregps started");
    Serial1.println("AT+CGSOCKCONT=1,\"IP\", \"internet\");
    Serial1.flush();
    x=0;
    do{
        while(Serial1.available()<=0);
        answer[x]=Serial1.read();
        Serial.print(answer[x]);
    }
```

```
x++;
}
while(!(answer[x-1]=='K'&&answer[x-2]=='O'));
delay(500);
Serial1.println("AT+CGPSURL=\"supl.google.com:7276\"");
Serial1.flush();
x=0;
do{
  while(Serial1.available()==0);
  answer[x]=Serial1.read();
  x++;
}
while(!(answer[x-1]=='K'&&answer[x-2]=='O'));
delay(500);
Serial1.println("AT+CGPSSSL=0");
Serial1.flush();
x=0;
do{
  while(Serial1.available()==0);
  answer[x]=Serial1.read();
  x++;
}
while(!(answer[x-1]=='K'&&answer[x-2]=='O'));
delay(500);
Serial1.println("AT+CGPS=1,2");
Serial1.flush();
x=0;
do{
  while(Serial1.available()==0);
  answer[x]=Serial1.read();
  x++;
}
while(!(answer[x-1]=='K'&&answer[x-2]=='O'));
}

void getgps(){                                     //receive GPS data function
  int found_coo=0;
```

```
int thesi=0;
int lastStringLength=0;
String gpsresponse;

while(found_coo < 1){
    delay(10000);
    while (found_coo < 1)
    {
        x=0;
        for (x=0;x<255;x++){
            dataIn[x]='\0';
        }
        delay(100);
        Serial1.println("AT+CGPSINFO");
        Serial1.flush();
        x=0;
        do{
            while(Serial1.available()==0);
            dataIn[x]=Serial1.read();
            x++;
        }
        while(!(dataIn[x-1]=='K'&&dataIn[x-2]=='O'));
        delay(100);
        Serial.println("Here is the response from gps-->");
        Serial.println(dataIn);
        Serial.println("End of the response.");
        gpsresponse = dataIn;
        // I use a string to extract substring that are GPS value
        Serial.print("gpsresponse: ");
        Serial.println(gpsresponse);
        lastStringLength = gpsresponse.length();
        Serial.print("Length:");
        Serial.println(lastStringLength);
        thesi = gpsresponse.lastIndexOf('+CGPSINFO:');
        Serial.print("Thesi:");
        Serial.println(thesi);
        Latitude1 = gpsresponse.substring((thesi+1),(thesi+12));
        Latitude1.toCharArray(Lat,12);
    }
}
```

```
Decimlat = gpsresponse.substring((thesi+6),(thesi+12));
Decimlat.toCharArray(Latdecimal,9);
float latsec= atof(Latdecimal);
latsec=(float)latsec/1000000;
latsec=(float)latsec*60;
dtostrf(latsec, 5, 2, Latsecond);
NorthSouth = gpsresponse.substring((thesi+13),(thesi+14));
NorthSouth.toCharArray(Latsymbol,2);
Longitude1 = gpsresponse.substring((thesi+15),(thesi+27));
Longitude1.toCharArray(Long,13);
Decimlong = gpsresponse.substring((thesi+21),(thesi+27));
Decimlong.toCharArray(Longdecimal,10);
float longsec= atof(Longdecimal);
longsec=(float)longsec/1000000;
longsec=(float)longsec*60;
dtostrf(longsec, 5, 2, Longsecond);
EastWest = gpsresponse.substring((thesi+28),(thesi+29));
EastWest.toCharArray(Longsymbol,2);
Date= gpsresponse.substring((thesi+30),(thesi+36));
Date.toCharArray(Datearray,7);
Gmt = gpsresponse.substring((thesi+37),(thesi+43));
Gmt.toCharArray(Gmtarray,7);

if (lastStringLength > 84) {
    found_coo++;
    break;
}
}
delay(5000);
}
found_coo=0;
}

void hour(){
//time correction function
if((Gmtarray[0]!='0') && (Gmtarray[1]!='7')){
    Gmtarray[0]='0';
    Gmtarray[1]='9';
}
}
```

```
else if((Gmtarray[0]=='0') && (Gmtarray[1]=='8')){
    Gmtarray[0]='1';
    Gmtarray[1]='0';
}
else if((Gmtarray[0]=='0') && (Gmtarray[1]=='9')){
    Gmtarray[0]='1';
    Gmtarray[1]='1';
}
else if((Gmtarray[0]=='1') && (Gmtarray[1]=='7')){
    Gmtarray[0]='1';
    Gmtarray[1]='9';
}
else if((Gmtarray[0]=='1') && (Gmtarray[1]=='8')){
    Gmtarray[0]='2';
    Gmtarray[1]='0';
}
else if((Gmtarray[0]=='1') && (Gmtarray[1]=='9')){
    Gmtarray[0]='2';
    Gmtarray[1]='1';
}
else if((Gmtarray[0]=='2') && (Gmtarray[1]=='1')){
    Gmtarray[0]='2';
    Gmtarray[1]='3';
}
else if((Gmtarray[0]=='2') && (Gmtarray[1]=='2')){
    Gmtarray[0]='0';
    Gmtarray[1]='0';
}
else if((Gmtarray[0]=='2') && (Gmtarray[1]=='3')){
    Gmtarray[0]='0';
    Gmtarray[1]='1';
}
else{
    Gmtarray[1]=Gmtarray[1]+2;
}
}

void email_text() {                                     //send email function
```

```
Serial1.print("AT+SMTPSRV=\");
Serial1.print(server);
Serial1.print("\",");
Serial1.println(port);
Serial1.flush();
x=0;
do{
  while(Serial1.available()==0);
  answer[x]=Serial1.read();
  x++;
}
while(!(answer[x-1]=='K'&&answer[x-2]=='O'));
delay(1000);
Serial1.print("AT+SMTPAUTH=1,\");
Serial1.print(user_name);
Serial1.print("\",\");
Serial1.print(password);
Serial1.println("\");
Serial1.flush();
x=0;
do{
  while(Serial1.available()==0);
  answer[x]=Serial1.read();
  x++;
}
while(!(answer[x-1]=='K'&&answer[x-2]=='O'));
delay(1000);
Serial1.print("AT+SMTPFROM=\");
Serial1.print(sender_add);
Serial1.print("\",\");
Serial1.print(sender_name);
Serial1.println("\");
Serial1.flush();
x=0;
do{
  while(Serial1.available()==0);
  answer[x]=Serial1.read();
  x++;
```

```
}
while(!(answer[x-1]=='K'&&answer[x-2]=='O'));
delay(1000);
Serial1.print("AT+SMTPRCPT=0,0,\");
Serial1.print(to_add);
Serial1.print("\,");
Serial1.print(to_name);
Serial1.println("\");
Serial1.flush();
x=0;
do{
  while(Serial1.available()==0);
  answer[x]=Serial1.read();
  x++;
}
while(!(answer[x-1]=='K'&&answer[x-2]=='O'));
delay(1000);
Serial1.print("AT+SMTPSUB=\");
Serial1.print(subject);
Serial1.println("\");
Serial1.flush();
x=0;
do{
  while(Serial1.available()==0);
  answer[x]=Serial1.read();
  x++;
}
while(!(answer[x-1]=='K'&&answer[x-2]=='O'));
delay(1000);
Serial1.flush();
Serial1.println("AT+SMTPBODY");
Serial1.flush();
x=0;
do{
  while(Serial1.available()==0);
  answer[x]=Serial1.read();
  x++;
}
```

```
while(!(answer[x-1]=='>'&&answer[x-2]=='>'));
Serial1.write("Position of Device:");
Serial1.write(0x0A);
Serial1.write(Latitude1[0]);
Serial1.write(Latitude1[1]);
Serial1.write("");
Serial1.write(Latitude1[2]);
Serial1.write(Latitude1[3]);
Serial1.write("");
Serial1.write(Latsecond);
Serial1.write("");
Serial1.write(Latsymbol);
Serial1.write(",");
Serial1.write(Longitude1[0]);
Serial1.write(Longitude1[1]);
Serial1.write(Longitude1[2]);
Serial1.write("");
Serial1.write(Longitude1[3]);
Serial1.write(Longitude1[4]);
Serial1.write("");
Serial1.write(Longsecond);
Serial1.write("");
Serial1.write(Longsymbol);
Serial1.write(0x0A);
Serial1.write("Date:");
Serial1.write(Datearray[0]);
Serial1.write(Datearray[1]);
Serial1.write("/");
Serial1.write(Datearray[2]);
Serial1.write(Datearray[3]);
Serial1.write("/");
Serial1.write(Datearray[4]);
Serial1.write(Datearray[5]);
Serial1.write(0x0A);
Serial1.write("UTC+2:00 Time: ");
Serial1.write(Gmtarray[0]);
Serial1.write(Gmtarray[1]);
Serial1.write(":");
```



```
Serial1.write(Gmtarray[2]);
Serial1.write(Gmtarray[3]);
Serial1.write(":");
Serial1.write(Gmtarray[4]);
Serial1.write(Gmtarray[5]);
Serial1.write(0x0A);
Serial1.write(0x1A);
Serial1.println("AT+CGSOCKCONT=1,\"IP\", \"internet\");
Serial1.flush();
x=0;
do{
    while(Serial1.available()==0);
    answer[x]=Serial1.read();
    Serial.print(answer[x]);
    x++;
}
while(!(answer[x-1]=='K'&&answer[x-2]=='O'));
Serial1.println("AT+SMTPSEND");
Serial1.flush();
x=0;
do{
    while(Serial1.available()==0);
    answer[x]=Serial1.read();
    x++;
}
while(!(answer[x-1]=='K'&&answer[x-2]=='O'));
Serial.println("Sending email");
x=0;
do{
    do{
        digitalWrite(led,HIGH);
    }
    while(Serial1.available()==0);
    digitalWrite(led,LOW);
    answer[x]=Serial1.read();
    x++;
}
while(!(answer[x-1]=='S'&&answer[x-2]=='S'));
```

```
Serial.println("Email sended successfully");
}

void sms(){                                     //send sms function
Serial.println("Sending sms");
Serial1.println("AT+CMGF=1");
delay(1000);
Serial1.print("AT+CMGS=\"");
Serial1.print(phone_number);
Serial1.println("\");
delay(1000);
Serial1.println("Position of Device:");
Serial1.print(Latitude1[0]);
Serial1.print(Latitude1[1]);
Serial1.print("o");
Serial1.print(Latitude1[2]);
Serial1.print(Latitude1[3]);
Serial1.print("");
Serial1.print(Latsecond);
Serial1.print("");
Serial1.print(Latsymbol);
Serial1.print(",");
Serial1.print(Longitude1[0]);
Serial1.print(Longitude1[1]);
Serial1.print(Longitude1[2]);
Serial1.print("o");
Serial1.print(Longitude1[3]);
Serial1.print(Longitude1[4]);
Serial1.print("");
Serial1.print(Longsecond);
Serial1.print("");
Serial1.println(Longsymbol);
Serial1.print("Date: ");
Serial1.print(Datearray[0]);
Serial1.print(Datearray[1]);
Serial1.print("/");
Serial1.print(Datearray[2]);
Serial1.print(Datearray[3]);
```

```
Serial1.print("/");
Serial1.print(Datearray[4]);
Serial1.println(Datearray[5]);
Serial1.print("UTC+2:00 Time: ");
Serial1.print(Gmtarray[0]);
Serial1.print(Gmtarray[1]);
Serial1.print(":");
Serial1.print(Gmtarray[2]);
Serial1.print(Gmtarray[3]);
Serial1.print(":");
Serial1.print(Gmtarray[4]);
Serial1.print(Gmtarray[5]);
Serial1.write(0x1A);
Serial1.write(0x0D);
Serial1.write(0x0A);
delay(5000);
Serial.println("Sms sended successfully");
}

void primary() { //operation function
  Serial.println("Primary Started");
  digitalWrite(ledg, HIGH);
  getgps();
  hour();
  delay(100);
  sms();
  delay(200);
  email_text();
  digitalWrite(ledg, LOW);
  delay(6000000);
}

void breakint() { //interrupt function
  w++;
}

void setup() { //Arduino setup function
  pinMode(led, OUTPUT);
```

```
pinMode(ledg, OUTPUT);
pinMode(ledb, OUTPUT);
pinMode(onModulePin, OUTPUT);
delay(4000);
attachInterrupt(0, breakint, CHANGE);
while(w == 0){
digitalWrite(ledb, HIGH);
delay(1000);
digitalWrite(ledb, LOW);
delay(1000);
Serial.println("Wait for interrupt");
}
digitalWrite(ledb, HIGH);
Serial1.begin(115200);
switchModule();
delay(25000);
pregps();
Serial.println("Setup ended");
}

void loop(){                                     //Arduino loop function
primary();
}
```