

ΣΧΟΛΗ ΜΗΧΑΝΙΚΩΝ  
ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ  
ΚΑΙ ΗΛΕΚΤΡΟΝΙΚΩΝ ΣΥΣΤΗΜΑΤΩΝ

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ  
ΣΥΣΚΕΥΗ ΕΝΤΟΠΙΣΜΟΥ ΘΕΣΗΣ RADAR  
ΤΑΧΥΤΗΤΑΣ ΜΕ ΤΗΝ ΒΟΗΘΕΙΑ GPS ΚΑΙ  
ΜΙΚΡΟΕΛΕΓΚΤΗ.



Της φοιτήτριας  
Τσάτσο Κριστέλντα  
Αρ. Μητρώου: 516140

Επιβλέπων  
Όνοματεπώνυμο: Γιακουμής Άγγελος  
Βαθμίδα: Επίκουρος καθηγητής

Ημερομηνία 10/09/2024

Τίτλος Δ.Ε.: Συσκευή εντοπισμού θέσης Radar ταχύτητας με την βοήθεια GPS και μικροελεγκτή.

Κωδικός Δ.Ε.: 24131

Όνοματεπώνυμο φοιτητή: Τσάτσο Κρισέλντα

Όνοματεπώνυμο εισηγητή: Γιακουμής Άγγελος

Ημερομηνία ανάληψης Δ.Ε. 02/03/2024

Ημερομηνία περάτωσης Δ.Ε. 10/09/2024

*Βεβαιώνω ότι είμαι ο συγγραφέας αυτής της εργασίας και ότι κάθε βοήθεια την οποία είχα για την προετοιμασία της είναι πλήρως αναγνωρισμένη και αναφέρεται στην εργασία. Επίσης, έχω καταγράψει τις όποιες πηγές από τις οποίες έκανα χρήση δεδομένων, ιδεών, εικόνων και κειμένου, είτε αυτές αναφέρονται ακριβώς είτε παραφρασμένες. Επιπλέον, βεβαιώνω ότι αυτή η εργασία προετοιμάστηκε από εμένα προσωπικά, ειδικά ως διπλωματική εργασία, στο Τμήμα Μηχανικών Πληροφορικής και Ηλεκτρονικών Συστημάτων του ΔΙ.ΠΑ.Ε.*

*Η παρούσα εργασία αποτελεί πνευματική ιδιοκτησία της φοιτήτριας Τσάτσο Κρισέλντα που την εκπόνησε. Στο πλαίσιο της πολιτικής ανοικτής πρόσβασης, ο συγγραφέας/δημιουργός εκχωρεί στο Διεθνές Πανεπιστήμιο της Ελλάδος άδεια χρήσης του δικαιώματος αναπαραγωγής, δανεισμού, παρουσίασης στο κοινό και ψηφιακής διάχυσης της εργασίας διεθνώς, σε ηλεκτρονική μορφή και σε οποιοδήποτε μέσο, για διδακτικούς και ερευνητικούς σκοπούς, άνευ ανταλλάγματος. Η ανοικτή πρόσβαση στο πλήρες κείμενο της εργασίας, δεν σημαίνει καθ' οιονδήποτε τρόπο παραχώρηση δικαιωμάτων διανοητικής ιδιοκτησίας του συγγραφέα/δημιουργού, ούτε επιτρέπει την αναπαραγωγή, αναδημοσίευση, αντιγραφή, πώληση, εμπορική χρήση, διανομή, έκδοση, μεταφόρτωση (downloading), ανάρτηση (uploading), μετάφραση, τροποποίηση με οποιονδήποτε τρόπο, τμηματικά ή περιληπτικά της εργασίας, χωρίς τη ρητή προηγούμενη έγγραφη συναίνεση του συγγραφέα/δημιουργού.*

Η έγκριση της διπλωματικής εργασίας από το Τμήμα Μηχανικών Πληροφορικής και Ηλεκτρονικών Συστημάτων του Διεθνούς Πανεπιστημίου της Ελλάδος, δεν υποδηλώνει απαραίτητως και αποδοχή των απόψεων του συγγραφέα, εκ μέρους του Τμήματος.

## Πρόλογος

Τα Radar ελέγχου ταχύτητας αποτελούν ένα κοινό θέαμα στους δρόμους όλων των χωρών παγκοσμίως. Πρόκειται για ένα πολύ σημαντικό και χρήσιμο εργαλείο το οποίο έχει οδηγήσει αφενός στην αποτροπή της επικίνδυνης οδήγησης και αφετέρου στη μείωση των ατυχημάτων. Από την άλλη πλευρά, μια μικρή έστω υπέρβαση του ορίου ταχύτητας θα μπορούσε να οδηγήσει στην επιβολή προστίμου χωρίς αυτό να θεωρείται απαραίτητο. Ο λόγος λοιπόν που επέλεξα να κάνω αυτή την εργασία είναι όχι για να δημιουργήσω μια συσκευή που θα δώσει τη δυνατότητα σε ορισμένους οδηγούς να αποφύγουν την κλήση και να συνεχίσουν μια κακή και επικίνδυνη οδηγική συμπεριφορά αλλά για να αποφευχθούν πρόστιμα για παραβάσεις οι οποίες δεν αποτελούν κανέναν απόλυτως κίνδυνο. Επιπλέον, με την εργασία αυτή προσφέρεται η δυνατότητα ηχητικής ειδοποίησης για κάποιες προκαθορισμένες ταχύτητες. Η επιλογή αυτή έγινε ώστε ο εκάστοτε οδηγός να μη χρειάζεται να κοιτάει διαρκώς το ταμπλό ώστε να διατηρήσει τη σωστή ταχύτητα και επομένως να μπορεί να παρακολουθεί και να συγκεντρωθεί στο δρόμο.

## Περίληψη

Η παρούσα διπλωματική ασχολείται με την σχεδίαση και υλοποίηση μιας συσκευής εντοπισμού radar ταχύτητας. Η ολοκλήρωσή της επιτυγχάνεται με τη βοήθεια του μικροελεγκτή PIC18F2420 και του GPS Ublox Neo 7M. Η συσκευή τοποθετείται στο ταμπλό του αυτοκινήτου ώστε να έχει όσο το δυνατόν περισσότερη οπτική επαφή με τον ουρανό με σκοπό να αλληλοεπιδρά κάθε χρονική στιγμή με δορυφόρους της γης. Ο μικροελεγκτής επεξεργάζεται τα σήματα που λαμβάνονται και σε συνδυασμό με το GPS η συσκευή αναγνωρίζει την ακριβή θέση του αυτοκινήτου. Παράλληλα, μέσω της βάσης δεδομένων που δημιουργεί ο χρήστης εντοπίζονται οι θέσεις των radar με σκοπό να προειδοποιήσουν τον οδηγό.

Στο 1<sup>ο</sup> Κεφάλαιο περιγράφεται η λειτουργία του GPS καθώς αποτελεί ένα σημαντικό κομμάτι του κυκλώματος και απαιτούνται βασικές γνώσεις για την κατανόηση της διαδικασίας εκπομπών και λήψεων δορυφορικών σημάτων.

Στο 2<sup>ο</sup> κεφάλαιο παρουσιάζεται ο σχεδιασμός του κυκλώματος, αναλύεται η λειτουργία του και παρουσιάζονται όλα τα εξαρτήματα που το συνθέτουν.

Στο 3<sup>ο</sup> κεφάλαιο αναλύεται η σύνθεση κυκλωμάτων μέσω λογισμικού προσομοίωσης και παρουσιάζονται τόσο η διαδικασία δημιουργίας τυπωμένου κυκλώματος όσο και η κατασκευή και σύνθεση της πλακέτας.

Τέλος, αναφέρονται τα συμπεράσματα που προέκυψαν από την ολοκλήρωση της εργασίας.

# Speed Radar positioning device with the help of GPS and microcontroller.

Tsatso Kriselda

## **Abstract**

This thesis deals with the design and implementation of a speed radar tracking device. Its integration is achieved with the help of the PIC18F2420 microcontroller and the Ublox Neo 7M GPS. The device is placed on the dashboard of the car in order to have as much visual contact with the sky as possible in order to interact with satellites of the earth at any time. The microcontroller processes the signals received and in conjunction with the GPS the device identifies the exact position of the car. At the same time, through the database created by the user, the positions of the radars are identified in order to warn the driver.

Chapter 1 describes the operation of the GPS as it is an important part of the circuit and basic knowledge is required to understand the process of transmitting and receiving satellite signals.

Chapter 2 presents the design of the circuit, analyses its operation and presents all the components that make up the circuit.

In chapter 3, the circuit synthesis through simulation software is discussed and presents both the process of creating a printed circuit and the construction and synthesis of the board.

Finally, the conclusions drawn from the completion of the thesis are mentioned.

# Περιεχόμενα

Πρόλογος .....	1
Περίληψη .....	2
Abstract .....	3
Περιεχόμενα.....	4
Κατάλογος Σχημάτων .....	5
Κατάλογος Πινάκων.....	6
Κεφάλαιο 1ο: Global Positioning System.....	7
1.1 Εισαγωγή.....	7
1.2 Δομή του συστήματος GPS .....	8
1.3 Δομή του σήματος GPS.....	11
1.4 Αρχή λειτουργίας GPS .....	13
1.5 Ublox Neo 7M GPS.....	14
Κεφάλαιο 2ο: Δημιουργία κυκλώματος.....	15
2.1 Λειτουργία κυκλώματος.....	15
2.2 Σχηματικό διάγραμμα κυκλώματος.....	15
Κεφάλαιο 3ο: Λογισμικό προσομοίωσης.....	18
3.1 Παρουσίαση λογισμικού.....	18
3.2 Δημιουργία Project .....	18
3.3 Δημιουργία PCB.....	27
3.4 Τύπωση πλακέτας.....	28
Κεφάλαιο 4ο: Συμπεράσματα ή/και προτάσεις βελτίωσης .....	31
ΒΙΒΛΙΟΓΡΑΦΙΑ .....	32
ΠΑΡΑΡΤΗΜΑ Α : ΚΩΔΙΚΑΣ .....	33

## Κατάλογος Σχημάτων

Εικόνα 1.1: Τμήματα του συστήματος GPS .....	8
Εικόνα 1.2: Δορυφόρος BLOCK IIR .....	9
Εικόνα 1.3: Τροχιά δορυφόρων γύρω από τη Γη .....	10
Εικόνα 1.4: Επίγειοι σταθμοί ελέγχου .....	10
Εικόνα 1.5: Παγκόσμιο δίκτυο παρακολούθησης .....	11
Εικόνα 1.6: Φέροντες συχνότητες και διαμόρφωση .....	12
Εικόνα 1.7: Ublox Neo 7M .....	14
Εικόνα 2.1: Σχηματικό διάγραμμα κυκλώματος .....	16
Εικόνα 3.1: Δημιουργία Project .....	18
Εικόνα 3.2: Ονομασία Project .....	19
Εικόνα 3.3: Περιβάλλον σχηματικού .....	19
Εικόνα 3.4: Commonly Library .....	20
Εικόνα 3.5: Προεπιλεγμένες τιμές του Commonly Library .....	20
Εικόνα 3.6: Library .....	21
Εικόνα 3.7: Εμφανιζόμενα στοιχεία εξαρτήματος .....	21
Εικόνα 3.8: Wiring Tool .....	22
Εικόνα 3.9: Αποθήκευση Project .....	22
Εικόνα 3.10: Simulation .....	23
Εικόνα 3.11: Μετατροπή schematic σε PCB .....	23
Εικόνα 3.12: Ρυθμίσεις παραμέτρων PCB .....	24
Εικόνα 3.13: Παράδειγμα δημιουργίας σχηματικού .....	25
Εικόνα 3.14: Παράδειγμα δημιουργίας PCB .....	25
Εικόνα 3.15: Σύνδεση στοιχείων στο PCB .....	26
Εικόνα 3.16: Δημιουργία Gerber αρχείου .....	26
Εικόνα 3.17: PCB συσκευής εντοπισμού θέσης radar .....	27
Εικόνα 3.18: Τυπωμένο κύκλωμα .....	28
Εικόνα 3.19: Τυπωμένη πλακέτα .....	28
Εικόνα 3.20: Κολλήσεις .....	29
Εικόνα 3.21: Τελικές συνδέσεις .....	29

## **Κατάλογος Πινάκων**

Πίνακας 3.1: Κατάλογος υλικών .....	30
-------------------------------------	----

## Κεφάλαιο 1ο: Global Positioning System

### 1.1 Εισαγωγή

Αυτό είναι το πρώτο κεφάλαιο του προτύπου συγγραφής διπλωματικών εργασιών του Τμήματος Μηχανικών Πληροφορικής Το GPS (Global Positioning System) αποτελεί ένα δορυφορικό σύστημα μέσω του οποίου γίνεται ο προσδιορισμός της θέσης ενός σημείου παρατήρησης συντεταγμένων X,Y,Z ως προς ένα σύστημα αναφοράς που είναι κατάλληλο.

Η ανάπτυξη του GPS την δεκαετία του '70 και μετέπειτα η αξιοποίησή του στις αρχές της δεκαετίας του '80 έγινε από το υπουργείο άμυνας των ΗΠΑ. Ο αρχικός σκοπός σχεδίασής του ήταν να καλύψει τις ανάγκες της ναυσιπλοΐας αλλά και για στρατιωτικούς σκοπούς. Αυτό που ήθελαν να επιτύχουν είναι ο προσδιορισμός θέσης σε πραγματικό χρόνο να εμφανίζει ακρίβεια  $\pm 10-15m$ . Αργότερα όμως έγινε κατανοητό πως η χρήση του GPS σε υπολογισμούς μεγαλύτερης ακρίβειας είναι δυνατό να επιτευχθεί και να επεκταθεί και έτσι σήμερα καλύπτει ένα μεγάλο κενό όσον αφορά τον προσδιορισμό θέσης.

Ο προσδιορισμός της θέσης γίνεται μετρώντας τέσσερις ψευδοαποστάσεις μεταξύ παρατηρητή και δορυφόρου. Αυτός ήταν και ο λόγος για τον οποίο οι τροχιές των δορυφόρων σχεδιάστηκαν έτσι ώστε να είναι εφικτή η παρατήρηση τουλάχιστον τεσσάρων δορυφόρων από οποιοδήποτε σημείο της γης και οποιαδήποτε χρονική στιγμή. Αξιοσημείωτο είναι πως για να προσδιοριστεί η θέση ενός σημείου του οποίου οι συντεταγμένες στο χώρο είναι άγνωστες αρκεί να μετρηθούν οι αποστάσεις από τρία σημεία των οποίων οι συντεταγμένες είναι γνωστές. Παρ' όλο λοιπόν που θα ήταν αρκετοί τρεις δορυφόροι για να προσδιοριστεί μια θέση, απαιτούνται στην πραγματικότητα τέσσερις. Αυτό συμβαίνει διότι πρέπει να προσδιοριστεί η διαφορά ανάμεσα στην ένδειξη του χρονόμετρου του χρήστη και την ένδειξη που εμφανίζει το χρονόμετρο του δορυφόρου. Όπως προκύπτει, ο όρος ψευδοαπόσταση δημιουργήθηκε για να περιγράψει αυτό το σφάλμα που εμφανίζεται.

Γενικά τα GPS εμφανίζουν ορισμένα πλεονεκτήματα σε σχέση με παλαιότερες επίγειες και δορυφορικές μεθόδους. Αυτά που ενδιαφέρουν και σχετίζονται με την παρούσα εργασία είναι:

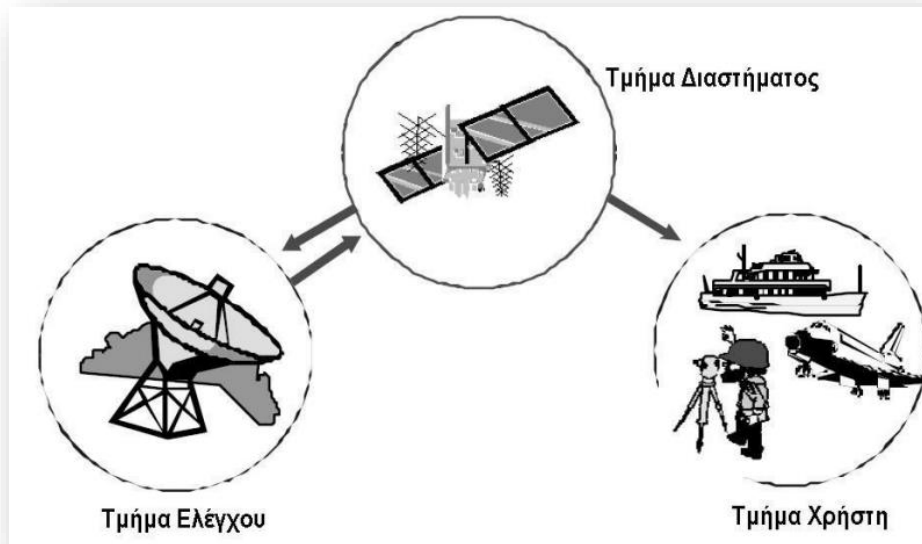
- Η απευθείας μετάδοση της θέσης ενός σημείου και επομένως είναι εφικτό να γνωρίζουμε κάθε στιγμή ποια είναι η θέση μας.
- Πρόκειται για ένα σύστημα το οποίο χρησιμοποιείται κάτω από οποιασδήποτε καιρικές συνθήκες.
- Για να μπορέσει να γίνει ο προσδιορισμός μιας θέσης αρκεί να υπάρχει ορατότητα προς έναν ικανοποιητικό αριθμό δορυφόρων και όχι να υπάρχει ορατότητα μεταξύ των σημείων παρατήρησης.
- Οι μετρήσεις απαιτούν μικρό χρόνο και γίνονται όλο το 24ωρο.

Από την άλλη πλευρά το GPS εμφανίζει κι ένα σημαντικό μειονέκτημα που είναι η απαίτηση ανοιχτού ορίζοντα. Η συσκευή εντοπισμού πρέπει να έχει οπτική επαφή με τους δορυφόρους κι αυτό αποτελεί πρόβλημα σε πυκνοκατοικημένες περιοχές.

## 1.2 Δομή του συστήματος GPS

Το GPS όπως αναφέρθηκε νωρίτερα αποτελεί ένα δορυφορικό σύστημα το οποίο περιγράφεται από συνεχή λειτουργία, παρακολούθηση, έλεγχο και συντήρηση των δορυφόρων και πιο συγκεκριμένα αποτελείται από τα παρακάτω κύρια τμήματα:

- Δορυφορικό τμήμα
- Τμήμα ελέγχου
- Τμήμα χρηστών



Εικόνα 1.1: Τμήματα του συστήματος GPS

### Δορυφορικό τμήμα

Το δορυφορικό τμήμα αποτελείται από δορυφόρους και από τα σήματα τα οποία οι ίδιοι εκπέμπουν. Ο αριθμός των δορυφόρων που περιστρέφονται σήμερα γύρω από τη γη είναι 28. Οι σειρές των δορυφόρων που είναι σε λειτουργία είναι οι BLOCK II, IIA και IIR.

Η ταξινόμησή τους γίνεται με ποικίλους τρόπους:

- Ανάλογα με τη σειρά εκτόξευσης
- Ανάλογα με την θέση στην τροχιά
- Σύμφωνα με δεδομένο κώδικα της NASA
- Σύμφωνα με ένα διεθνούς επιπέδου κώδικα
- Με βάση τον αριθμό PRN, ο οποίος δείχνει τη εβδομάδα του P-Κώδικα κατά την οποία εκπέμπει ο δορυφόρος

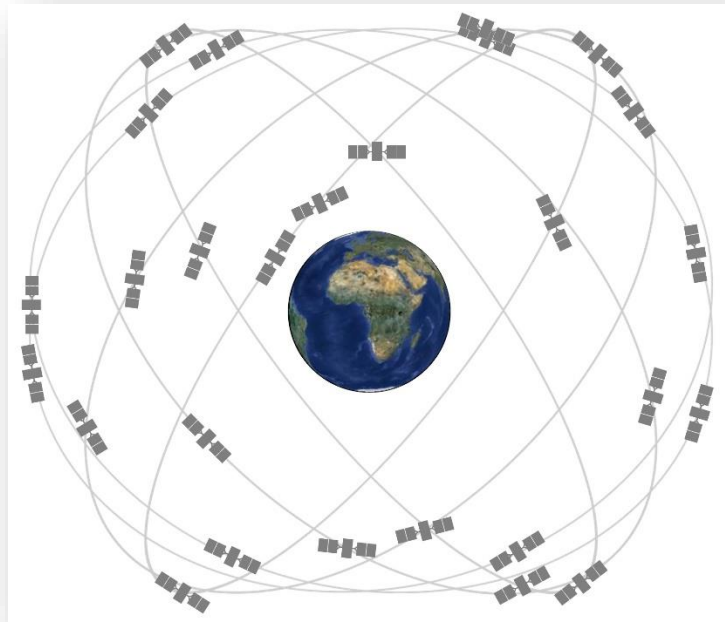
Ο τελευταίος τρόπος αποτελεί τον πιο συνηθισμένο τρόπο καταχώρισης.



Εικόνα 1.2: Δορυφόρος BLOCK IIR

Οι δορυφόροι έχουν κατανεμηθεί ισόποσα σε έξι τροχιακά επίπεδα. Τα επίπεδα αυτά σχηματίζουν μεταξύ τους γωνία  $60^\circ$  ενώ κάθε επίπεδο σχηματίζει γωνία κλίσης  $55^\circ$  με τον ισημερινό της Γης. Δεν χρησιμοποιούνται άμεσα όμως και οι 28 δορυφόροι. Αντ' αυτού χρησιμοποιούνται οι 25 και οι υπόλοιποι 3 χρησιμοποιούνται ως εφεδρικοί με σκοπό να αντικαταστήσουν κάποιον από τους βασικούς δορυφόρους σε περίπτωση βλάβης. Οι δορυφόροι περιστρέφονται με περίοδο 12 αστρικές ώρες.

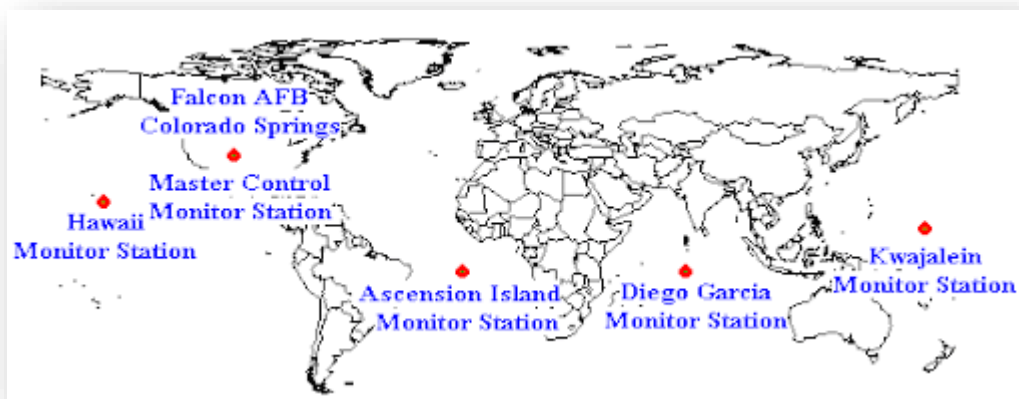
Οι δορυφόροι βρίσκονται σε ελλειπτικές τροχιές με μέγιστη εκκεντρότητα  $e=0.015$ . Το ύψος στο οποίο περιστρέφονται είναι περίπου 20200km ενώ κάθε τροχιακό επίπεδο από τα υπάρχοντα έξι, περιλαμβάνει τέσσερις δορυφόρους. Κάθε δορυφόρος έχει σχεδιαστεί ώστε να «ζει» περίπου 7.5 χρόνια. Μέσω του GPS ο χρήστης έχει τη δυνατότητα να «βλέπει» από οποιοδήποτε σημείο στη γη έναν αριθμό δορυφόρων ο οποίος είναι μεταξύ 5 και 8.



Εικόνα 1.3: Τροχιά δορυφόρων γύρω από τη Γη

#### Τμήμα ελέγχου

Στο τμήμα ελέγχου βρίσκονται πέντε επίγειοι σταθμοί ελέγχου και παρακολούθησης δορυφορικών σημάτων. Ο κύριος σταθμός ελέγχου βρίσκεται στην τοποθεσία Colorado Springs (ΗΠΑ) ενώ οι υπόλοιποι τέσσερις βρίσκονται στις περιοχές: Ascension Island, Diego Garcia, Kwajalein και Hawaii. Αυτό που κάνουν ουσιαστικά οι σταθμοί ελέγχου είναι να προβλέπουν τη δορυφορική θέση και τα παραμέτρους των δορυφορικών χρονομέτρων. Οι δορυφόροι τροφοδοτούνται από τους σταθμούς ελέγχου κάθε 8 ώρες και έπειτα τα δεδομένα αυτά εκπέμπονται προς τους χρήστες του GPS.



Εικόνα 1.4: Επίγειοι σταθμοί ελέγχου

Ο μικρός αριθμός επίγειων σταθμών παρακολούθησης καθιστά την ακρίβεια του δικτύου ανεπαρκή για ορισμένες εφαρμογές. Έτσι προέκυψε η ανάγκη για δημιουργία επιπλέον ανεξάρτητων δικτύων παρακολούθησης ώστε να εμπλουτιστεί το δίκτυο GPS. Το 1994 η Διεθνής Υπηρεσία GPS ανέλαβε τη διαχείριση και παρακολούθηση οπότε και προέκυψε ένα εκτεταμένο παγκόσμιο δίκτυο ενώ δημιουργήθηκαν και υπηρεσίες μέσω των οποίων τα δεδομένα διατίθενται μέσω του διαδικτύου.

### Τμήμα χρηστών

Το τμήμα αυτό αποτελείται από τους δέκτες GPS. Οι δέκτες είναι υπεύθυνοι για τη λήψη, την επεξεργασία των σημάτων και την καταγραφή των μετρήσεων. Ένας δέκτης αποτελείται από τα εξής τρία στοιχεία: την κεραία, τον βασικό δέκτη και τον υπολογιστή. Μέσω της κεραίας είναι δυνατό να κεντρώνεται σε σημεία για τον προσδιορισμό της θέσης. Όπως αναφέρθηκε και νωρίτερα, οι τροχιές είναι σχεδιασμένες με τέτοιο τρόπο ώστε κάθε χρονική στιγμή και σε οποιοδήποτε σημείο να υπάρχει ορατότητα με τέσσερις δορυφόρους.



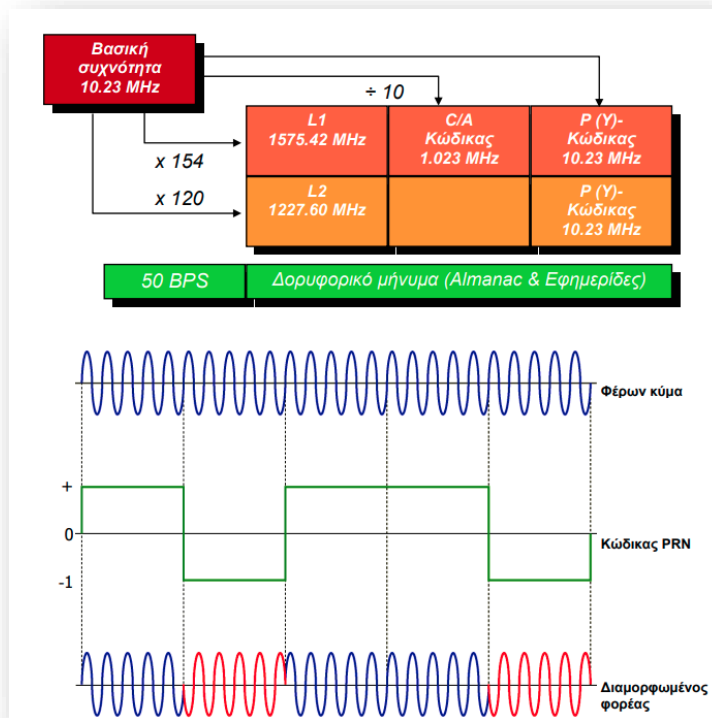
Εικόνα 1.5: Παγκόσμιο δίκτυο παρακολούθησης

### 1.3 Δομή του σήματος GPS

Οι δορυφόροι του συστήματος GPS εκπέμπουν σε δύο συχνότητες L1 και L2, οι οποίες καλούνται φέρουσες και παράγονται μέσω της θεμελιώδους συχνότητας των 10.23MHz. Έτσι προκύπτουν:

$$L1 = 154 \times 10.23 \text{ MHz} = 1575.42 \text{ MHz} \quad (1.1)$$

$$L2 = 120 \times 10.23 \text{ MHz} = 1227.6 \text{ MHz} \quad (1.2)$$



Εικόνα 1.6: Φέρουσες συχνότητες και διαμόρφωση

Ο κώδικας που χρησιμοποιείται για τη διαμόρφωση των φερουσών συχνοτήτων ονομάζεται κώδικας PRN (Pseudo Random Noise Code). Ο κώδικας αυτός αποτελείται από μια φαινομενικά τυχαία σειρά που περιλαμβάνει +1 και -1. Αναλυτικότερα, υπάρχουν οι παρακάτω 3 κώδικες:

- P
- C/A
- D

Ο P κώδικας ή αλλιώς ακριβής κώδικας (Precision Code), έχει συχνότητα  $f=10.23\text{MHz}$ , ίση δηλαδή με τη θεμελιώδη συχνότητα, μήκος παλμού ίσο περίπου με 30m και ρυθμός επανάληψης περίπου 267 ημέρες. Ο συγκεκριμένος κώδικας λαμβάνεται και στις 2 συχνότητες. Ένα τμήμα του κώδικα το οποίο έχει διάρκεια επτά ημέρες, παραμένει αποθηκευμένο στη μνήμη των δορυφόρων, το οποίο ανανεώνεται τα μεσάνυχτα του Σαββάτου. Υπάρχει η δυνατότητα εσκεμμένης παρέμβασης όπου δημιουργείται ένας κώδικας στον οποίο έχουν πρόσβαση μόνο χρήστες με την απαραίτητη εξουσιοδότηση.

Ο κώδικας C/A (Coarse/acquisition Code) εμφανίζει πολύ χαμηλότερη ακρίβεια από τον P κώδικα και συγκεκριμένα 10 φορές χαμηλότερη. Η συχνότητα που εμφανίζει είναι  $f=1.023\text{MHz}$  και ο ρυθμός επανάληψης είναι 1msec. Συμπερασματικά, η διαμόρφωση και η μετάδοση γίνεται μόνο στη συχνότητα L1.

Ο κώδικας D (Data Code) ή αλλιώς μήνυμα ναυσιπλοΐας, αποτελείται από μια σειρά από bits και η συχνότητά του είναι 50Hz. Ο κώδικας αυτός είναι υπεύθυνος για την παροχή πληροφοριών σχετικά με το χρόνο εκπομπής του σήματος από το δορυφόρο κάθε χρονική στιγμή, σχετικά με τις διορθώσεις

των καθυστερήσεων ατμόσφαιρας για τα δορυφορικά χρονόμετρα, σχετικά με τα στοιχεία τροχιάς των δορυφόρων κ.α. Η πληροφορία που περιέχεται σε κάθε ναυτιλιακό μήνυμα είναι 1500 bits με συνολική διάρκεια 30 δευτερόλεπτα και ρυθμό μετάδοσης 50 bps. Η περαιτέρω ανάλυση των τμημάτων του κώδικα αυτού δεν αποτελεί κομμάτι της παρούσας εργασίας.

Τα μεσάνυχτα κάθε Σαββάτου γίνεται η επανάληψη όλων των κωδίκων από την αρχή. Κάθε δορυφόρος επίσης αντιστοιχίζεται με μια μοναδική δομή του κώδικα P. Αξίζει να σημειωθεί πως από την αρχή που ξεκίνησε να λειτουργεί το GPS ο κώδικας P χρησιμοποιήθηκε για στρατιωτικούς σκοπούς και δεν μπορεί να χρησιμοποιηθεί ελεύθερα για πολιτικούς σκοπούς.

#### 1.4 Αρχή λειτουργίας GPS

Στον δέκτη GPS λαμβάνεται και αναλύεται το λαμβανόμενο σήμα και με μετρήσεις αποστάσεις που γίνονται ανάμεσα σε δορυφόρο και δέκτη, γίνεται ο προσδιορισμός της θέσης του δέκτη. Ενώ οι δορυφόροι του συστήματος έχουν ατομικά χρονόμετρα ρουβιδίου ή καισίου, οι δέκτες GPS έχουν χρονόμετρα χαμηλής ή μέσης ακρίβειας. Αυτό έχει ως αποτέλεσμα πέραν των ατμοσφαιρικών χρονικών καθυστερήσεων να εμφανίζονται και χρονικές καθυστερήσεις από το χρονόμετρο του δέκτη αλλά και του δορυφόρου. Επομένως, όταν προσδιορίζεται η θέση ενός δέκτη, εκτός από τις συντεταγμένες X,Y,Z του δέκτη υπάρχει και η μεταβλητή dT η οποία συμβολίζει τον χρόνο που καθυστερεί το χρονόμετρο του δέκτη συγκριτικά με το χρόνο αναφοράς του GPS. Ο χρόνος αυτός έχει έναρξη την ώρα 0 της 05/01/1980. Η αναφορά της προσδιοριζόμενης θέσης, συντεταγμένων X,Y,Z γίνεται στο Παγκόσμιο Σύστημα Αναφοράς του έτους 1984 το οποίο είναι γνωστό και ως WGS84.

Στη συνέχεια αναφέρονται ορισμένες κατηγοριοποιήσεις των GPS.

Υπάρχουν 2 κατηγορίες διάκρισης των μετρήσεων:

- Μετρήσεις με τη χρήση ψευδοαποστάσεων
- Μετρήσεις με τη χρήση φάσεων

με πιο διαδεδομένες ως προς την ακρίβεια, τις μετρήσεις με τη χρήση φάσεων.

Επίσης, υφίστανται 2 μέθοδοι για τον προσδιορισμό της θέσης:

- Η στατική
- Όπου ο δέκτης διακρίνεται από ακινησία και οι παρατηρήσεις έχουν διάρκεια λίγα λεπτά έως μερικές ώρες.
- Η κινηματική
- Όπου ο δέκτης κινείται και λαμβάνει συνεχώς σήμα από δορυφόρο.

Ο τρόπος να προσδιοριστεί η θέση με τη χρήση GPS μπορεί να είναι:

- Απόλυτος
- Όπου ο υπολογισμός της θέσης γίνεται σύμφωνα με το γεωκεντρικό σύστημα αναφοράς
- Σχετικός

Όπου ο καθορισμός της θέσης του δέκτη προκύπτει συγκριτικά με κάποιον άλλο δέκτη.

Στις περιπτώσεις που απαιτείται μεγάλη ακρίβεια δεδομένων, γίνεται χρήση των τεχνικών του σχετικού προσδιορισμού.

## 1.5 Ublox Neo 7M GPS

Το GPS που χρησιμοποιήθηκε στην κατασκευή της συσκευής εντοπισμού είναι το Ublox Neo 7m. Τα βασικά στοιχεία του μοντέλου αυτού είναι:

- Τροφοδοτικό 3V-5V
- Μονάδα με κεραμική κεραία, σήμα super
- Απενεργοποίηση EEPROM για την αποθήκευση των δεδομένων παραμέτρων διαμόρφωσης
- Με εφεδρική μπαταρία δεδομένων
- Υπάρχουν φώτα LED
- Υψηλής ποιότητας PCB χρυσής εμβάπτισης
- Μέγεθος κεραίας: 25\*25mm/0.98\*0.98»
- Μέγεθος μονάδας: 25\*35mm/0.98\*1.37»
- Διάμετρος εγκατάστασης: 3mm
- Τάση τροφοδοσίας: 3V-5V
- Ο προεπιλεγμένος ρυθμός baud: 9600



Εικόνα 1.7: Ublox Neo 7M

## Κεφάλαιο 2ο: Δημιουργία κυκλώματος

### 2.1 Λειτουργία κυκλώματος

Αρχικά, όταν η συσκευή ενεργοποιηθεί για πρώτη φορά, το GPS θα ξεκινήσει να ψάχνει δεδομένα δορυφόρων. Ο χρόνος που θα χρειαστεί για να κάνει ανάκτηση των δεδομένων εξαρτάται από το χρονικό διάστημα που πέρασε από την τελευταία ενεργοποίηση της συσκευής. Έτσι, ο χρόνος αυτός μπορεί να είναι από μερικά δευτερόλεπτα μέχρι περίπου 1 λεπτό. Μία ακόμη παράμετρος που επηρεάζει το χρόνο ανάκτησης δεδομένων είναι η κίνηση του αυτοκινήτου. Οι χρόνοι που αναφέρθηκαν νωρίτερα ισχύουν για ένα ακίνητο όχημα. Εάν το όχημα βρίσκεται σε κίνηση τότε οι χρόνοι αυτοί αυξάνονται αφού το GPS προσπαθεί ταυτόχρονα να αντισταθμίσει την κίνηση του οχήματος.

Κάθε δευτερόλεπτο το σύστημα υπολογίζει την απόσταση από την πλησιέστερο radar ταχύτητας, την ταχύτητα με την οποία κινείται το όχημα, την κατεύθυνση του οχήματος καθώς και το υψόμετρο σε σχέση με την μέση επιφάνεια της θάλασσας.

Ο χρήστης έχει τη δυνατότητα να επιλέξει εάν θέλει να ενημερωθεί όταν πλησιάζει σε ένα radar ή εάν έχει υπερβεί κάποιο όριο ταχύτητας.

Παράλληλα με τον εντοπισμό μιας θέσης radar ταχύτητας, παρέχεται στο χρήστη-οδηγό και η δυνατότητα ηχητικής προειδοποίησης κατά την υπέρβαση προκαθορισμένων ταχυτήτων. Με παρόμοιο τρόπο όπως και με τις θέσεις των radar δίνεται η δυνατότητα δημιουργίας μιας βάσης δεδομένων όπου θα περιλαμβάνονται όρια ταχυτήτων για συγκεκριμένους δρόμους. Κάθε φορά που ο οδηγός θα μπαίνει σε δρόμο με υψηλότερη ταχύτητα από την προκαθορισμένη, το σύστημα θα τον ενημερώνει με έναν ήχο.

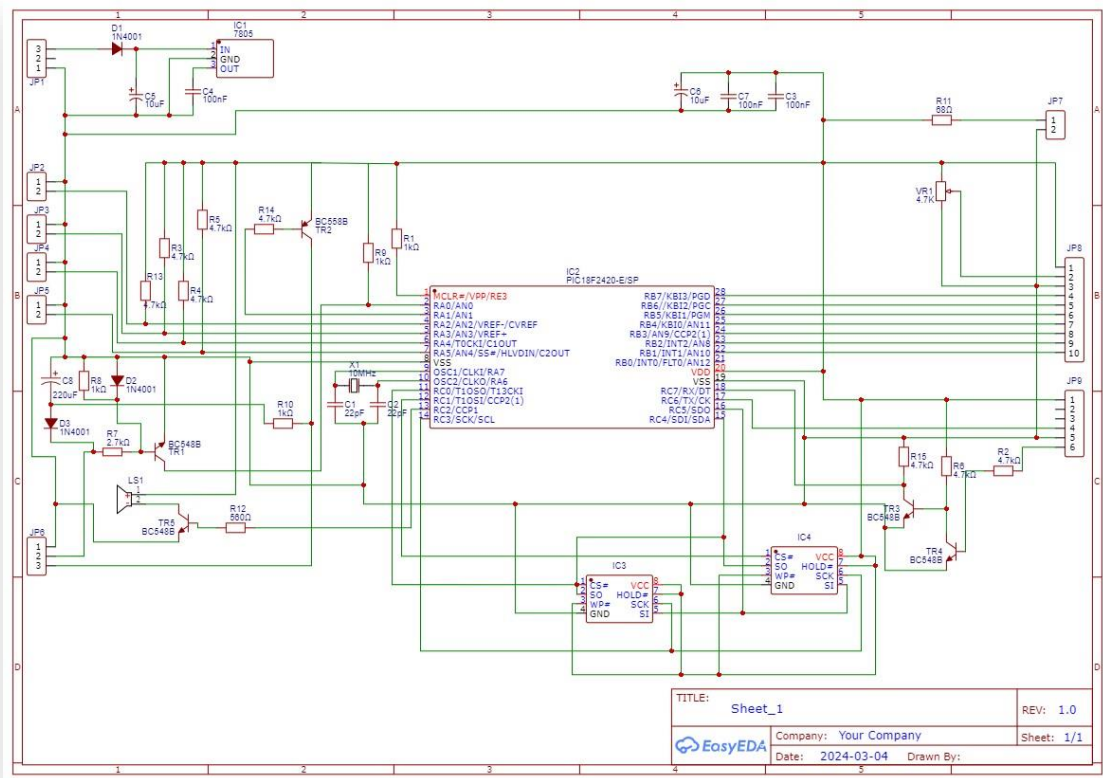
Με τη χρήση της διεπαφής RS232 δίνεται η δυνατότητα εξαγωγής της βάσης δεδομένων που έχει δημιουργήσει ένας χρήστης και μεταφόρτωσής της στο διαδίκτυο με σκοπό να τη μοιραστεί με άλλους χρήστες. Έτσι, μπορούν οι χρήστες να ανταλλάσσουν μεταξύ τους δεδομένα και να εμπλουτίζουν την βάση δεδομένων τους πολύ εύκολα. Αυτό μπορεί και έχει επιτευχθεί μέσω ιστοτόπων που δημιουργήθηκαν στο διαδίκτυο.

Η συσκευή δίνει τη δυνατότητα αποθήκευσης 10000 θέσεων radar και με τη βοήθεια του μικροελεγκτή μπορεί να το επιτύχει μέσα σε ένα μόλις δευτερόλεπτο.

### 2.2 Σχηματικό διάγραμμα κυκλώματος

Στη συνέχεια παρατίθεται το σχηματικό διάγραμμα του κυκλώματος.

## Κεφάλαιο 2



Εικόνα 2.1: Σχηματικό διάγραμμα κυκλώματος

Τα σήματα που λαμβάνονται από τον δέκτη GPS εισέρχονται στο κύκλωμα μέσω της υποδοχής JP9. Η ακίδα εξόδου των δεδομένων GPS παρέχει μόνο 3V με αποτέλεσμα να μην είναι αρκετό για τον PIC. Έτσι, με τη βοήθεια των τρανζίστορ TR3 και TR4 η τάση αυτή ενισχύεται και φτάνει στα 5V τα οποία μεταφέρονται στη θύρα UART του επεξεργαστή μέσω του ακροδέκτη RC7 του μικροελεγκτή. Ο ακροδέκτης RC6 που αποτελεί την έξοδο UART συνδέεται στη μονάδα GPS. Στην υποδοχή GP6 βρίσκεται μια επιπλέον σύνδεση σειριακής θύρας η οποία λειτουργεί σαν διεπαφή υπολογιστή. Μέσω των τρανζίστορ TR1 και TR2 και την διεπαφή RS232 μετατοπίζεται η στάθμη των σημάτων σε επίπεδα TTL. Ο πυκνωτής C8 καθώς και η διόδος D3 κρατούν αρνητική τάση από το σήμα εκπομπής του υπολογιστή με σκοπό να παρέχεται ταλάντευση τάσης εξόδου RS232. Τα IC3 και IC4 είναι σειριακές μνήμες EEPROM λειτουργίας SPI που παρέχουν τη μη πτητική μνήμη για τη βάση δεδομένων της συσκευής. Κάθε θέση κάμερας απαιτεί 6 byte και συγκεκριμένα 24 bit για τη διατήρηση των τιμών γεωγραφικού πλάτους και μήκους.

Η lcd οθόνη είναι συνδεδεμένη στον μικροελεγκτή σε λειτουργία σύνδεσης 4bit. Το VR1 χρησιμεύει στον έλεγχο της αντίθεσης. Για τον οπίσθιο φωτισμό που χρησιμοποιείται παρέχεται ισχύς από την υποδοχή JP7. Παράλληλα για τη μείωση της φωτεινότητας του οπίσθιου φωτισμού γίνεται χρήση της αντίστασης R11. Οι αντιστάσεις R3, R4, R5 και R13 πολώνουν τους ακροδέκτες του μικροελεγκτή. Η τροφοδοσία ορίζεται από 9V-14V και εισάγεται μέσω της υποδοχής JP1. Με σκοπό να μειωθεί η τάση αυτή στα 5V και να εισέλθει στο κύκλωμα, χρησιμοποιείται ο ρυθμιστής τάσης IC1.

Η ηχητικές ειδοποιήσεις παρέχονται μέσω του ακροδέκτη RC2 του μικρολεγκτή. Μέσω της αντίστασης R12 το σήμα αποθηκεύεται στην προσωρινή μνήμη και η ενίσχυσή του επιτυγχάνεται μέσω του τρανζίστορ TR5 το οποίο εν συνεχεία οδηγείται στο μεγάφωνο LS1.

Ο διακόπτης S1 χρησιμεύει ώστε να απενεργοποιεί το σύστημα σε περίπτωση που το όχημα παρέχει ισχύ ενώ ο κινητήρας είναι σβηστός.

Ο διακόπτης S4 (next) χρησιμοποιείται για να μετακινηθεί ο χρήστης στο μενού.

Ο διακόπτης S5 (select) χρησιμοποιείται ώστε η οθόνη που προβάλλεται μια δεδομένη στιγμή να γίνεται η προεπιλεγμένη. Σημειώνεται πως μετά από 5 δευτερόλεπτα αδράνειας της συσκευής, η οθόνη επιστρέφει στην προεπιλεγμένη οθόνη.

Ο διακόπτης S3 (clear) χρησιμοποιείται είτε για να γίνει σίγαση κάποιας ειδοποίησης είτε για να διαγραφεί η πλησιέστερη θέση κάμερας από τη βάση δεδομένων. Αυτό συμβαίνει εάν πατήσουμε το clear ενώ δεν ακούγεται κάποια ηχητική προειδοποίηση.

Ο διακόπτης S2 (record) χρησιμοποιείται για την καταγραφή μιας θέσης κάμερας στη βάση δεδομένων. Τη στιγμή που κάποιος χρήστης περνάει από μία θέση, θα πρέπει να πατήσει τον διακόπτη αυτό εφόσον αυτό είναι εφικτό και ασφαλές.

Καθώς πλησιάζει ο χρήστης σε ένα radar, η συσκευή αρχίζει να εκπέμπει ηχητικά σήματα η συχνότητα των οποίων αυξάνεται όσο η απόσταση μικραίνει.

Παρόμοια, σε περίπτωση που υπάρξει υπέρβαση του ορίου ταχύτητας, εκπέμπονται ηχητικά σήματα τα οποία σταματούν κατά την επιβράδυνση του οχήματος.

## Κεφάλαιο 3ο: Λογισμικό προσομοίωσης

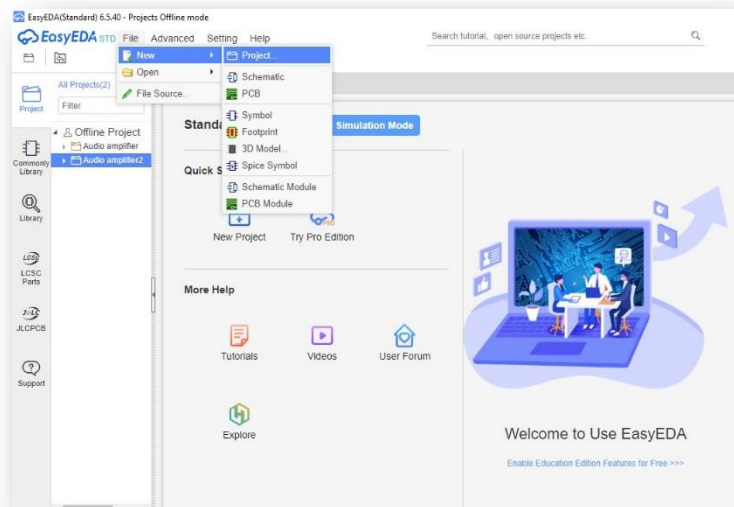
### 3.1 Παρουσίαση λογισμικού

Για τη σχεδίαση αλλά και την υλοποίηση του κυκλώματος χρησιμοποιήθηκε το λογισμικό EasyEDA. Πρόκειται για ένα λογισμικό σχεδιασμού PCB και προσομοίωσης κυκλώματος. Ο λόγος που επιλέχθηκε το συγκεκριμένο λογισμικό είναι πως πρόκειται για ένα αρκετά εύχρηστο πρόγραμμα που προσφέρει και τη λειτουργία της σχεδίασης του κυκλώματος και της προσομοίωσης αλλά και της δημιουργίας τυπωμένου κυκλώματος. Παράλληλα, είναι δωρεάν επομένως μπορεί οποιοσδήποτε να το εγκαταστήσει και να το χρησιμοποιήσει στον υπολογιστή του χωρίς κάποια επιβάρυνση.

### 3.2 Δημιουργία Project

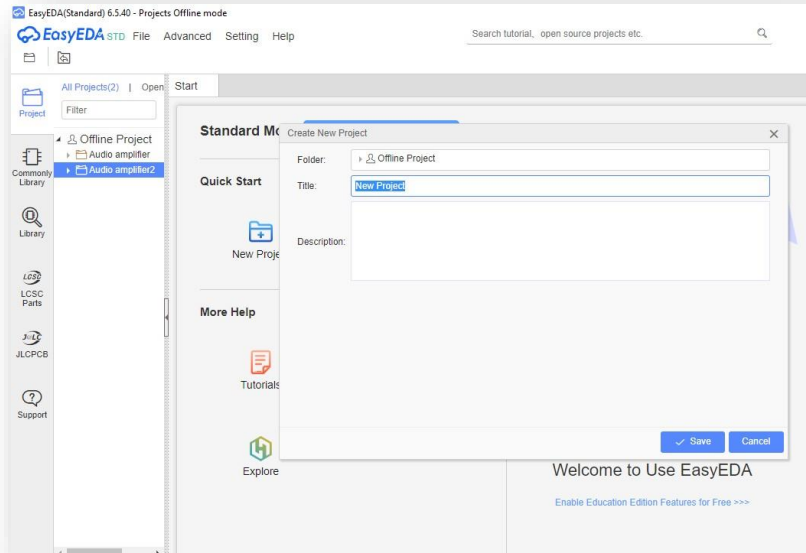
Στη συνέχεια περιγράφεται ο τρόπος χρήσης του προγράμματος και παρουσιάζονται τα κυκλώματα που προέκυψαν με τη βοήθειά του.

Αρχικά από το μενού επιλέγουμε File→ New→ Project.



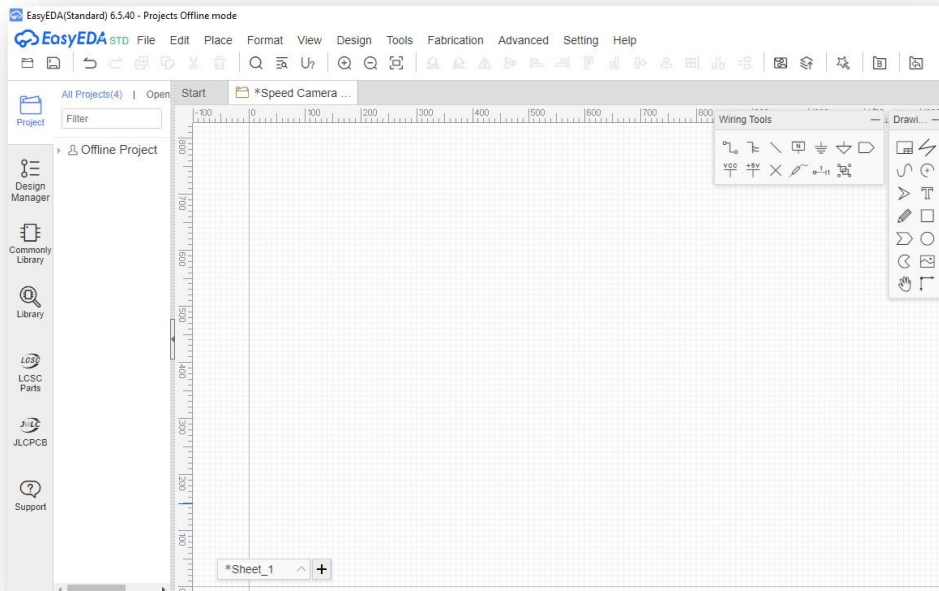
Εικόνα 3.1: Δημιουργία Project

Στη συνέχεια στο πεδίο «Title» δίνουμε το όνομα του αρχείου και επιλέγουμε Save.



Εικόνα 3.2: Ονομασία Project

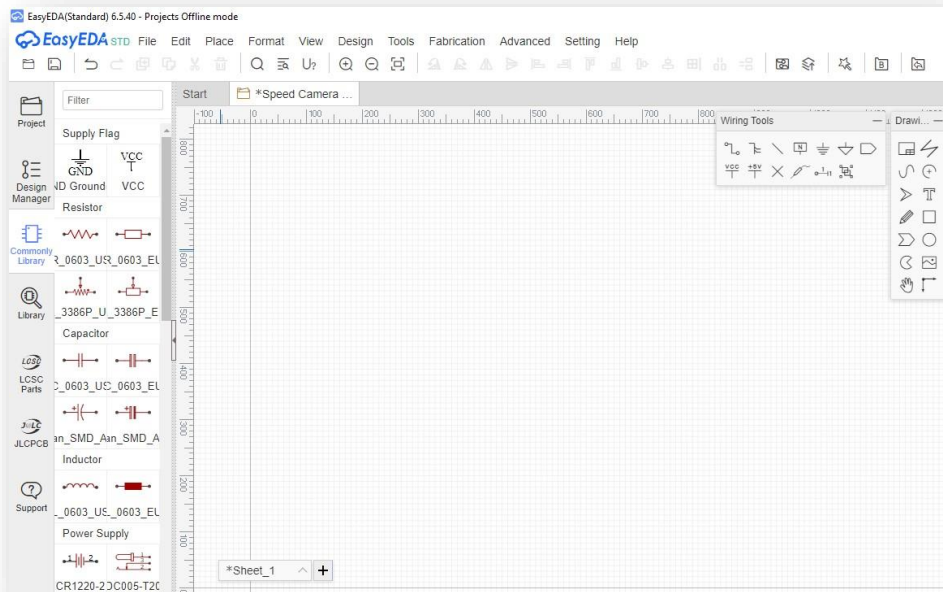
Σε αυτό το σημείο ανοίγει ένα κενό φύλλο εργασίας στο οποίο θα τοποθετηθούν τα εξαρτήματα που θα δημιουργήσουν το σχηματικό του κυκλώματος.



Εικόνα 3.3: Περιβάλλον σχηματικού

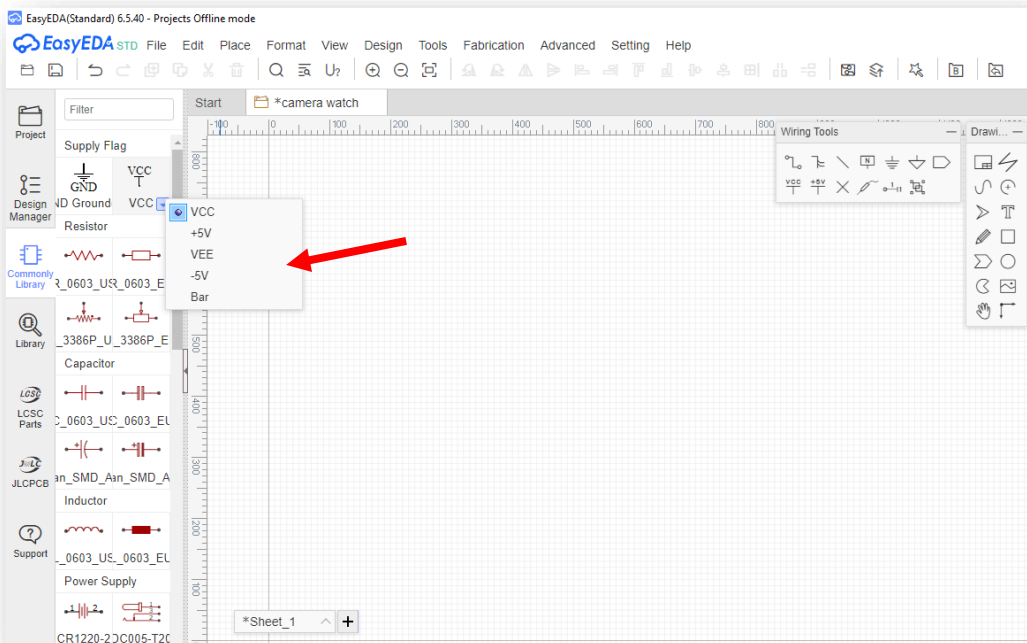
## Κεφάλαιο 3

Αριστερά στο παράθυρο του προγράμματος εμφανίζεται μια στήλη με ορισμένες επιλογές. Μία από αυτές είναι η «Commonly Library» από την οποία μπορούμε να προσθέσουμε στο σχηματικό μας κάποια κοινώς χρησιμοποιούμενα εξαρτήματα.



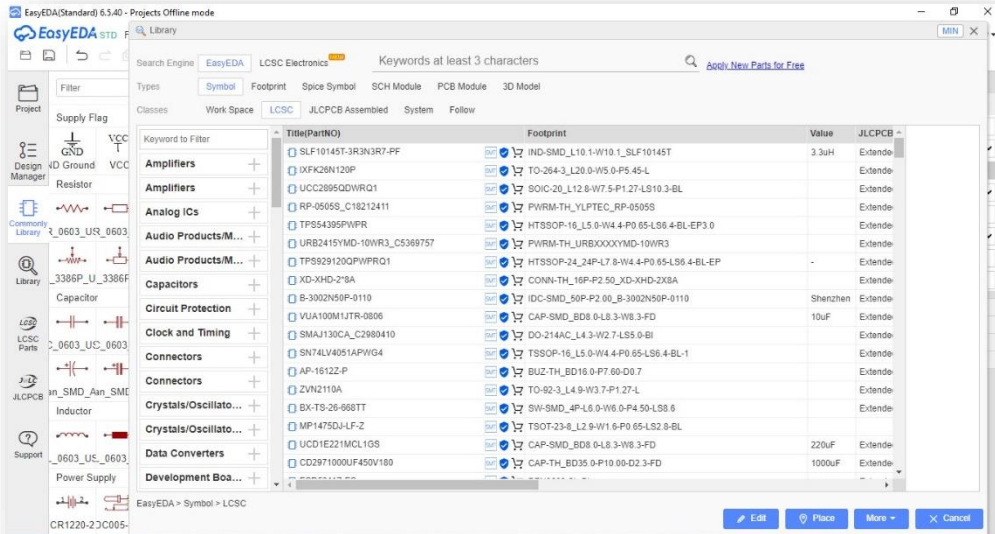
Εικόνα 3.4: Commonly Library

Πατώντας πάνω στο εξάρτημα μπορούμε να επιλέξουμε ανάμεσα σε κάποιες προεπιλεγμένες τιμές.



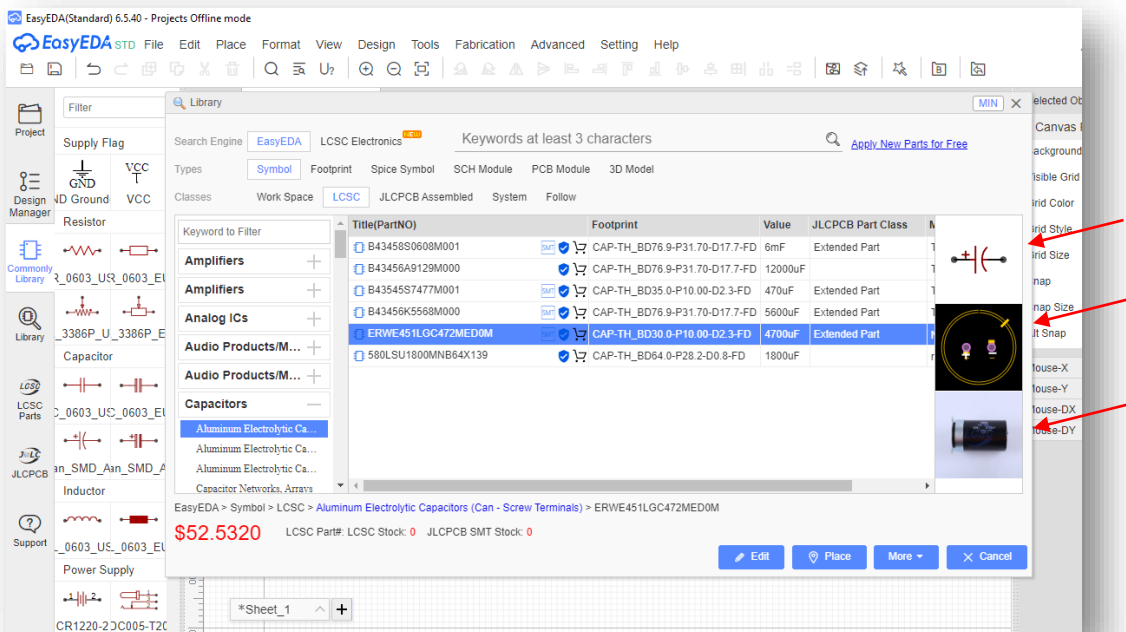
Εικόνα 3.5: Προεπιλεγμένες τιμές του Commonly Library

Επιλέγοντας το πεδίο “Library” που βρίσκεται ακριβώς από κάτω, δίνεται η επιλογή για την αναζήτηση οποιουδήποτε εξαρτήματος. Σε αυτό το σημείο να σημειωθεί πως το συγκεκριμένο λογισμικό δίνει τη δυνατότητα επιλογής πραγματικών εξαρτημάτων τα οποία βρίσκει κανείς στο εμπόριο και μάλιστα υπάρχει η δυνατότητα παραγγελίας τους.



Εικόνα 3.6: Library

Επιλέγοντας κάποιο εξάρτημα εμφανίζονται στα δεξιά το σύμβολο, το footprint και η φωτογραφία του εξαρτήματος.



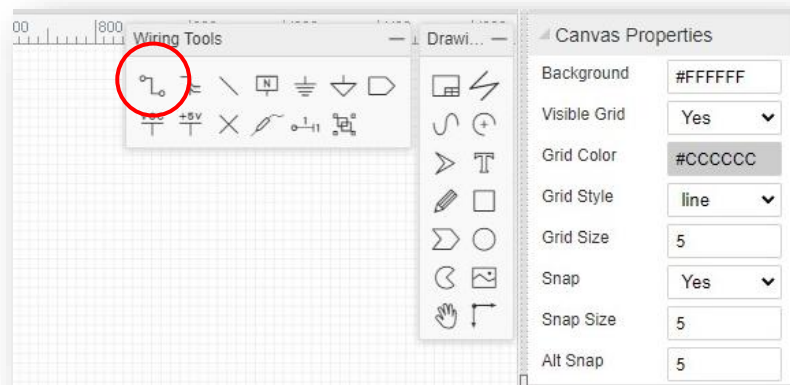
Εικόνα 3.7: Εμφανιζόμενα στοιχεία εξαρτήματος

### Κεφάλαιο 3

Η τοποθέτηση των εξαρτημάτων γίνεται επιλέγοντας από τον πίνακα το επιθυμητό εξάρτημα και επιλέγοντας Place στο κάτω μέρος του παραθύρου.

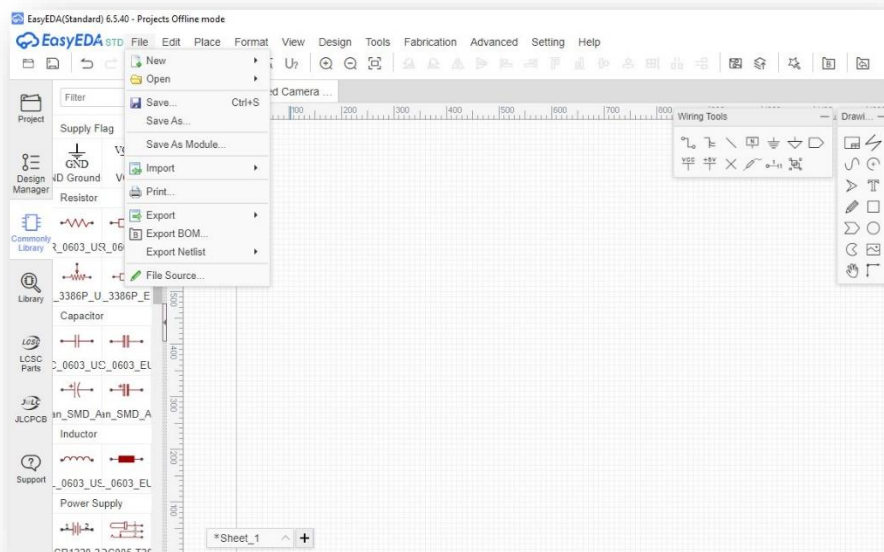
Κατά την επιλογή των εξαρτημάτων χρειάζεται ιδιαίτερη προσοχή καθώς πολλά από αυτά δεν μπορούν να μεταφερθούν στο PCB που θα δημιουργηθεί αλλά χρησιμοποιούνται μόνο για την προσομοίωση του κυκλώματος.

Αφού τοποθετηθούν τα επιθυμητά εξαρτήματα στο φύλλο, επιλέγουμε από τον πίνακα “Wiring Tools” το καλώδιο και κάνουμε τις απαραίτητες ενώσεις.



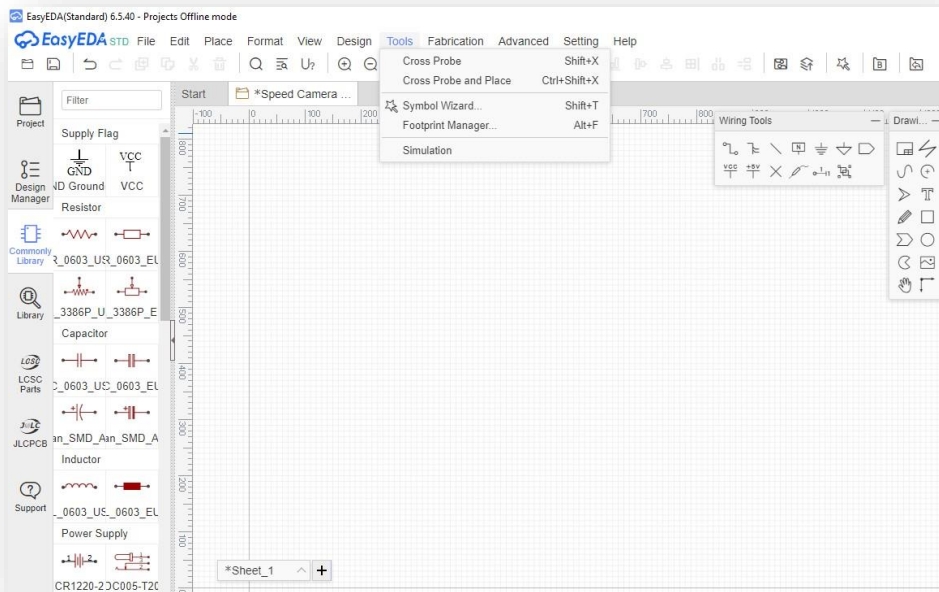
Εικόνα 3.8: Wiring Tool

Για να μπορέσει να ολοκληρωθεί η προσομοίωση θα πρέπει να αποθηκεύσουμε το σχηματικό πηγαίνοντας από το μενού File → Save.



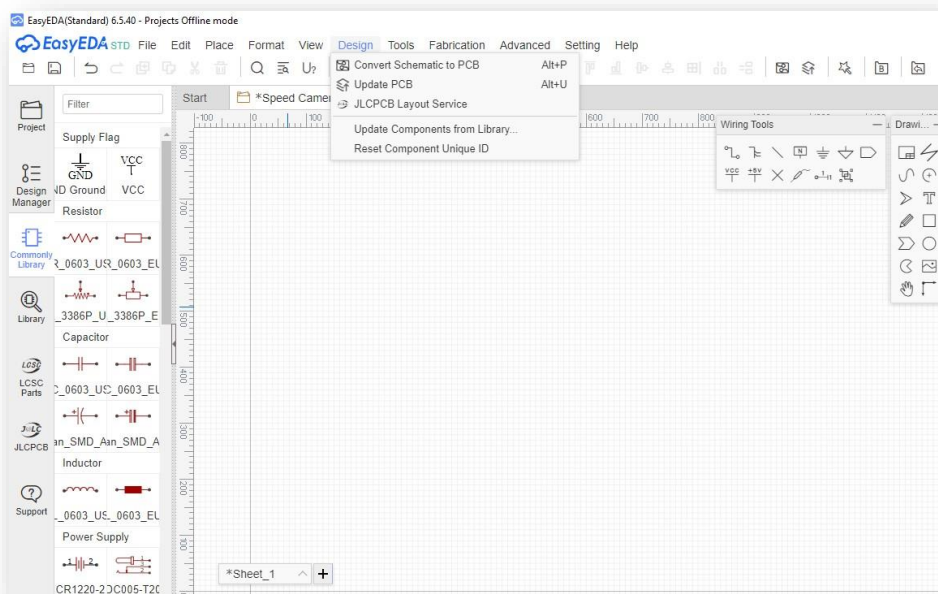
Εικόνα 3.9: Αποθήκευση Project

Στη συνέχεια μπορούμε να μεταβούμε στην κατάσταση προσομοίωσης επιλέγοντας από το μενού Tools → Simulation.



Εικόνα 3.10: Simulation

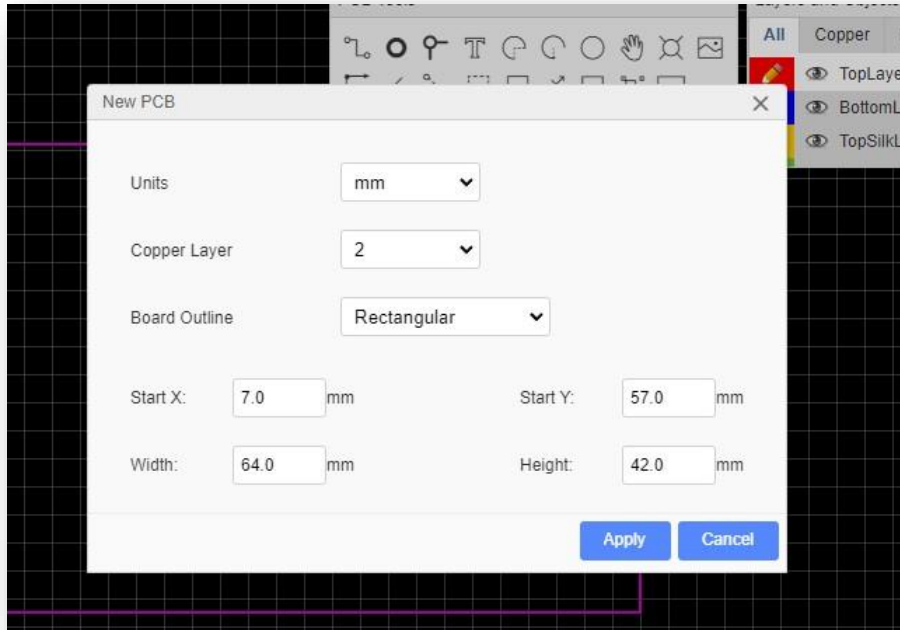
Αφού επιβεβαιώσουμε πως το κύκλωμα δουλεύει σωστά μπορούμε να το μετατρέψουμε σε PCB. Επιλέγουμε από το μενού Design → Convert Schematic to PCB



Εικόνα 3.11: Μετατροπή schematic σε PCB

### Κεφάλαιο 3

Πλέον μεταφερόμαστε από το περιβάλλον του σχηματικού στο περιβάλλον του PCB. Στο παράθυρο που ανοίγει επιλέγουμε στο πεδίο “Copper Layer” τον αριθμό των επιπέδων που θέλουμε να έχει το PCB και στη συνέχεια τις διαστάσεις του τυπωμένου κυκλώματος και επιλέγουμε Apply.

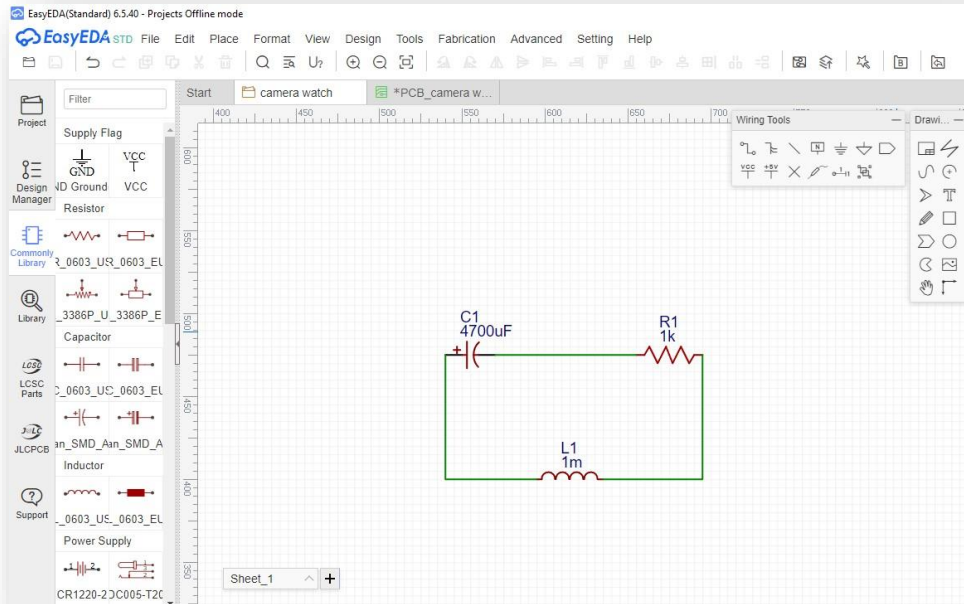


Εικόνα 3.12: Ρυθμίσεις παραμέτρων PCB

Μπορούμε τώρα να επιλέξουμε το χρώμα των γραμμών που θα συνδέουν τα εξαρτήματα ανάλογα με το επίπεδο στο οποίο βρίσκονται από τον πίνακα “Layers and Objects”.

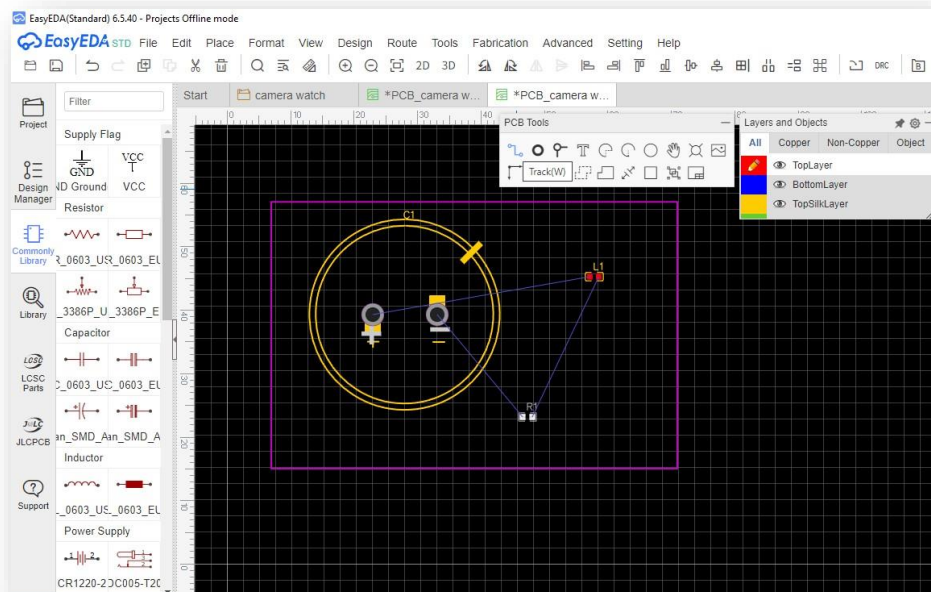
Σημειώνεται πως τα εξαρτήματα είναι ήδη συνδεδεμένα καθώς λαμβάνονται υπόψη οι συνδέσεις που έχουν γίνει στη σχηματικό.

Για την εργασία έχουν επιλεγεί διαστάσεις 101.6x76, ένα επίπεδο και κόκκινη γραμμή επιπέδου.



Εικόνα 3.13: Παράδειγμα δημιουργίας σχηματικού

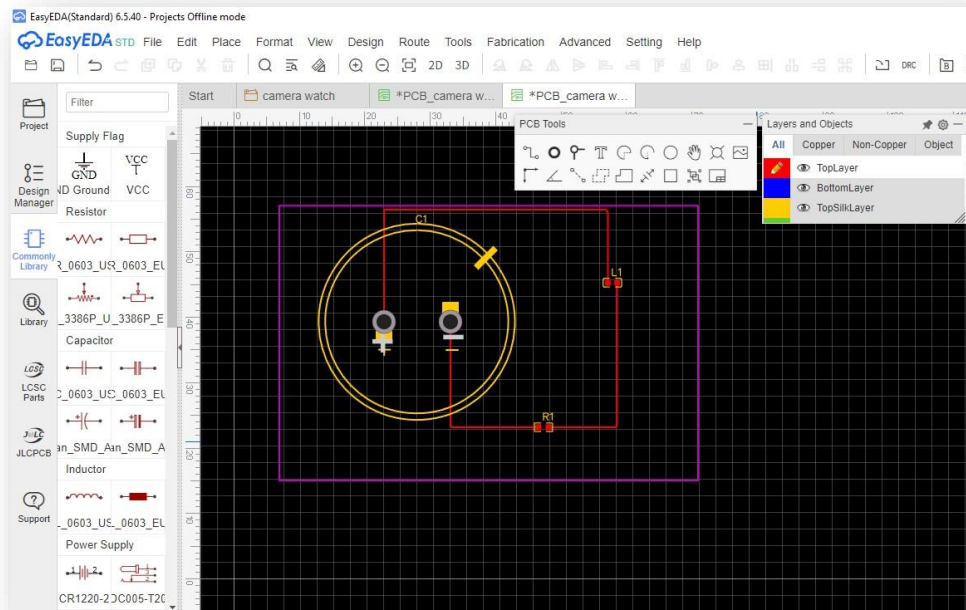
Σε αυτό το σημείο επιλέγουμε ουσιαστικά για τις ήδη υπάρχουσες συνδέσεις, τον τρόπο με τον οποίο θα τυπωθούν στην πλακέτα, «χαράζουμε» δηλαδή τον δρόμο τους. Με σκοπό να μπορέσουμε να ελέγξουμε ποια εξαρτήματα συνδέονται μεταξύ τους, εμφανίζεται η διακριτική μπλε γραμμή η οποία θα εξαφανιστεί μόλις ολοκληρώσουμε τη σύνδεση.



Εικόνα 3.14: Παράδειγμα δημιουργίας PCB

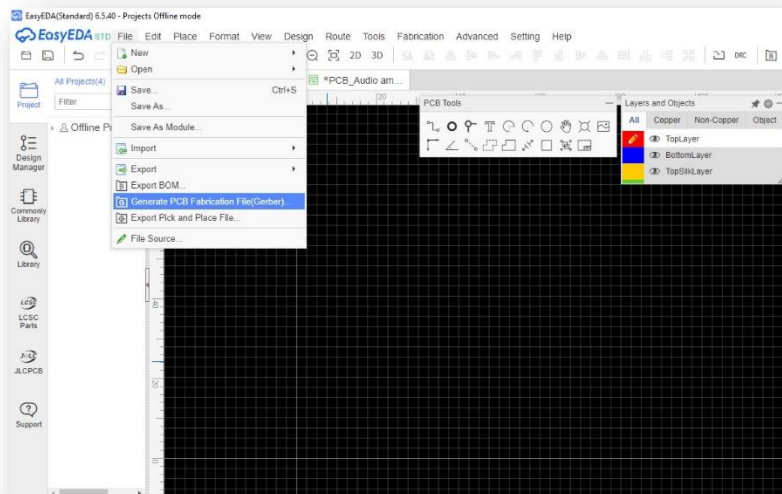
## Κεφάλαιο 3

Για τη σύνδεση επιλέγουμε από το παράθυρο “PCB Tools” το εργαλείο “Track”.



Εικόνα 3.15: Σύνδεση στοιχείων στο PCB

Με σκοπό να κρατήσουμε τον έλεγχο των συνδεδεμένων και μη εξαρτημάτων ελέγχουμε εάν έχουν καλυφθεί όλες οι διακριτικές μπλε γραμμές. Να σημειωθεί πως σε περίπτωση που ορισμένα εξαρτήματα δεν έχουν ενωθεί με τη χρήση του “Track”, το λογισμικό μας βγάζει μήνυμα σφάλματος κατά την αποθήκευσή του και την εξαγωγή του Gerber το οποίο αναφέρεται παρακάτω. Τέλος, αποθηκεύουμε το αρχείο και από το μενού επιλέγουμε File → Generate PCB Fabrication File (Gerber)

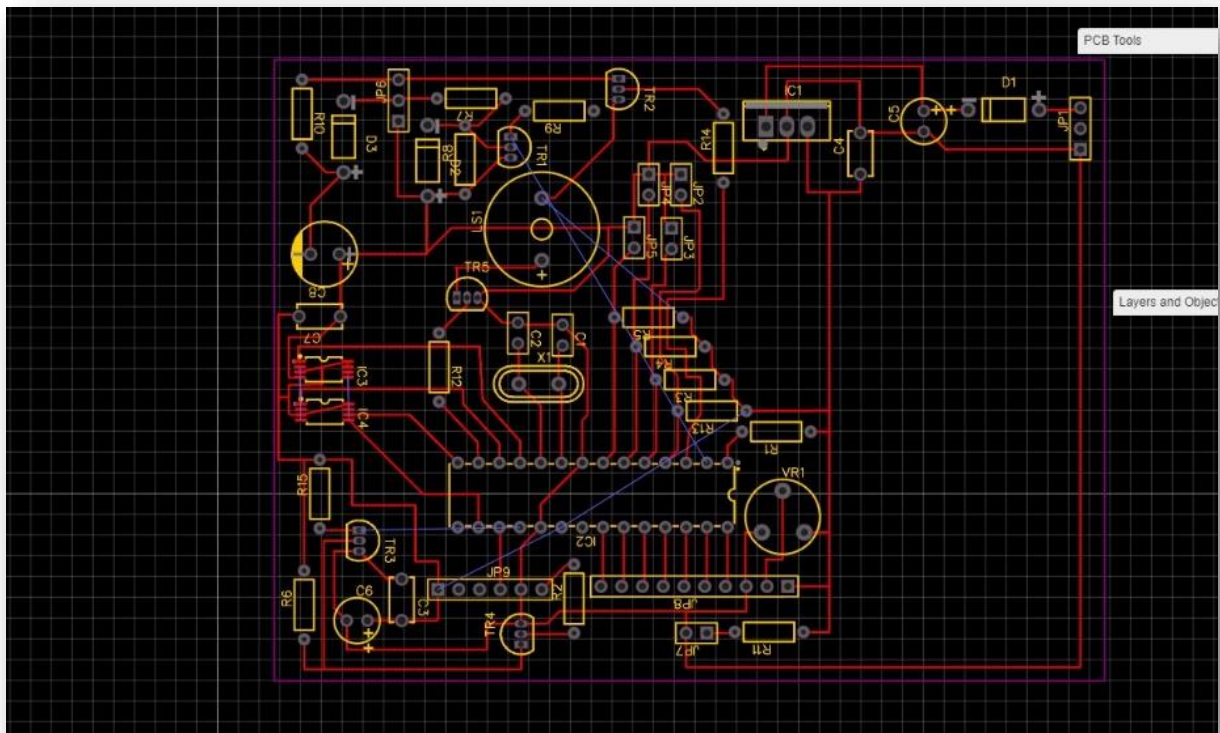


Εικόνα 3.16: Δημιουργία Gerber αρχείου

Το αρχείο αυτό θα χρησιμοποιηθεί αργότερα για την τύπωση του κυκλώματος στην επιλεγμένη πλακέτα.

### 3.3 Δημιουργία PCB

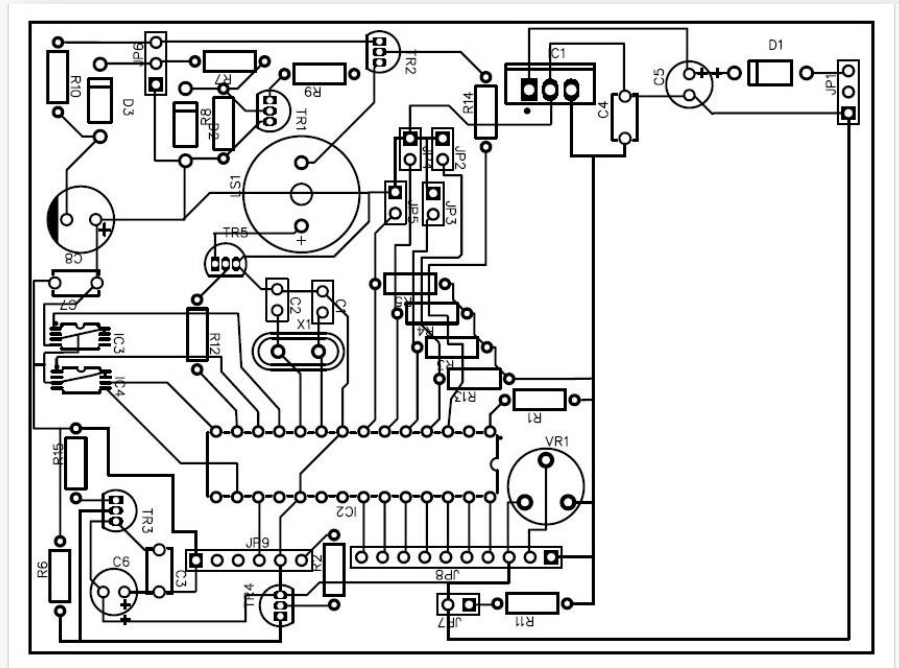
Το PCB που δημιουργήθηκε σύμφωνα με την διαδικασία που περιγράφηκε παραπάνω και μέσω του σχηματικού που παρουσιάστηκε στο κεφάλαιο 2, φαίνεται παρακάτω:



Εικόνα 3.17: PCB συσκευής εντοπισμού θέσης radar

Όπως αναφέρθηκε και νωρίτερα, οι διακριτικές μπλε γραμμές υποδηλώνουν ασύνδετα στοιχεία. Στην προκειμένη και με σκοπό να αποφευχθεί η χρήση πλακέτας διπλής όψης, επιλέχθηκε οι συγκεκριμένες συνδέσεις να γίνουν με εξωτερική κόλληση καλωδίου.

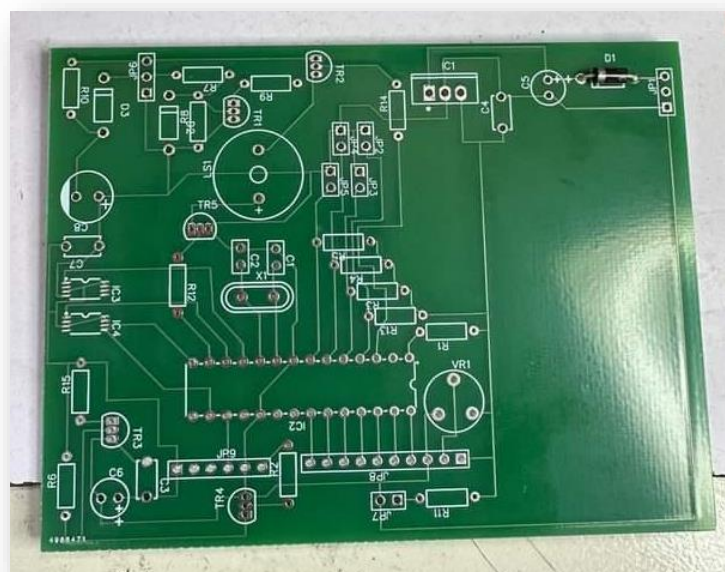
Τέλος, προέκυψε το τυπωμένο κύκλωμα σύμφωνα με το οποίο ολοκληρώθηκε και η δημιουργία της πλακέτας.



Εικόνα 3.18: Τυπωμένο κύκλωμα

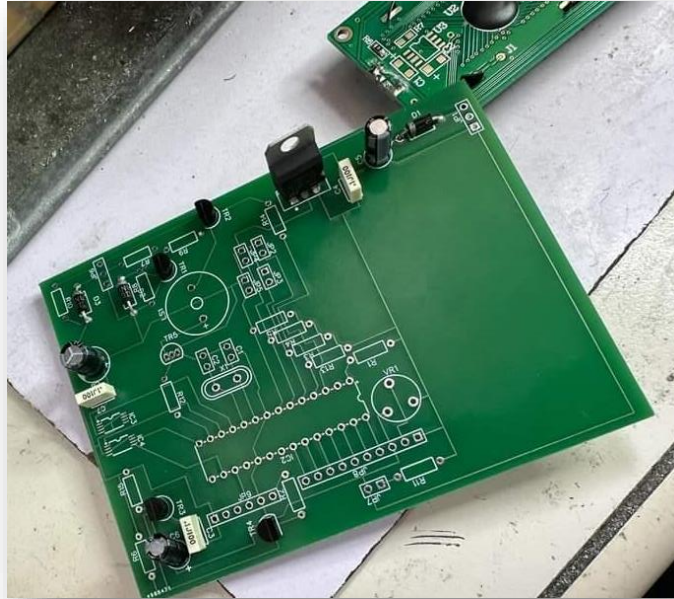
### 3.4 Τύπωση πλακέτας

Με τη βοήθεια του τυπωμένου κυκλώματος έγινε η εκτύπωση πάνω σε μια πλακέτα.



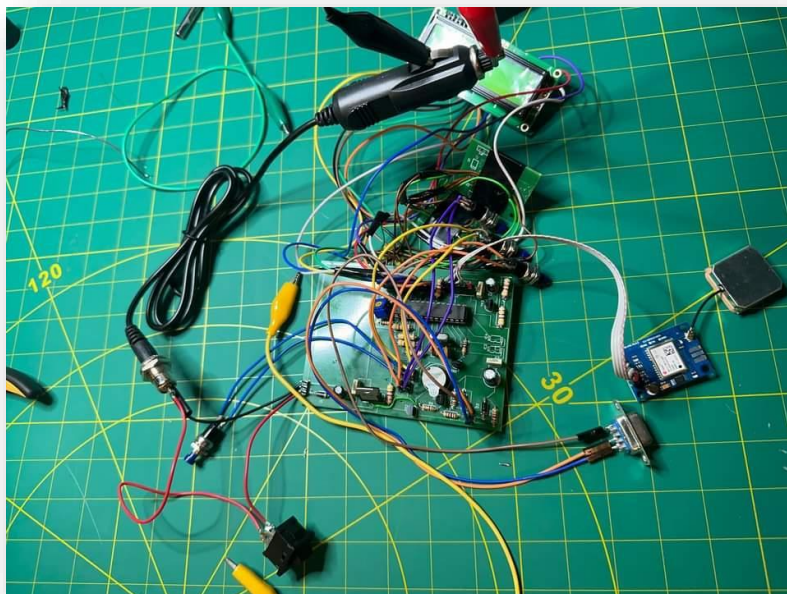
Εικόνα 3.19: Τυπωμένη πλακέτα

Έπειτα έγιναν οι κολλήσεις των εξαρτημάτων επάνω στην πλακέτα καθώς και οι επιπλέον συνδέσεις που χρειάστηκε με εξωτερικά καλώδια.



Εικόνα 3.20: Κολλήσεις

Το τελικό κύκλωμα φαίνεται παρακάτω:



Εικόνα 3.21: Τελικές συνδέσεις

### Κεφάλαιο 3

Ο κατάλογος με τα υλικά που χρησιμοποιήθηκαν παρουσιάζεται στον παρακάτω πίνακα:

Πίνακας 3.1: Κατάλογος υλικών

ΕΙΔΟΣ	ΤΙΜΗ	ΠΟΣΟΤΗΤΑ
Αντίσταση	1kΩ	4
Αντίσταση	4.7kΩ	8
Αντίσταση	2.7kΩ	1
Αντίσταση	68Ω	1
Αντίσταση	560Ω	1
Ποτενσιόμετρο	4.7kΩ	1
Κεραμικός πυκνωτής	22p	2
Πυκνωτής πολυεστέρα	100n	3
Πυκνωτής ταντάλου	10μF/16V	2
Πυκνωτής ταντάλου	220μF/16V	1
Δίοδος	1N4001	3
Σταθεροποιητής	7805	1
Μικροελεγκτής	PIC18F2420	1
Σειριακή μνήμη	25C256	2
NPN Τρανζίστορ	BC548	4
PNP Τρανζίστορ	BC558	1
Connector	Header pin-3 way	2
Connector	Header pin-2 way	5
Connector	Header pin-10 way	1
Connector	Header pin-6 way	1
Connector	9-Pin, D type, female	1
Ηχείο	T70P015H	1
Κρύσταλλος	10MHz	1
GPS	Ublox Neo 7M	1
Οθόνη LCD	2 γραμμών/16 χαρακτήρων	1
Διακόπτης	s.p.s.t. slider	1
Διακόπτης	Push button	4

## **Κεφάλαιο 4ο: Συμπεράσματα ή/και προτάσεις βελτίωσης**

Πολλές φορές η προσπάθεια επιβολής του νόμου στο οδικό δίκτυο και ο έλεγχος της ταχύτητάς των οδηγών οδηγεί σε όχι και τόσο αποτελεσματικές λύσεις. Η χρήση radar για τον έλεγχο της ταχύτητας δεν αποτελεί πάντα ένα χρήσιμο εργαλείο καθώς δεν μπορεί να διακρίνει τις επικίνδυνες και μη υπερβάσεις. Με το σύστημα εντοπισμού θέσης radar ταχύτητας ο οδηγός μπορεί να αποφύγει άσκοπες κλήσεις αλλά παράλληλα να ενημερωθεί για κάποια υπέρβαση ταχύτητας χωρίς να χρειαστεί να έχει τα μάτια του μονίμως στο ταμπλό.

Η σχεδίαση και κατασκευή της συσκευής αυτής συμπεριέλαβε μελέτη γύρω από τη λειτουργία του GPS, μελέτη γύρω από τον προγραμματισμό του μικροελεγκτή καθώς και γνώσεις γύρω από τη χρήση λογισμικού και σύνθεση μιας πλακέτας.

Τα αποτελέσματα της εργασίας είναι πολύ θετικά. Η συσκευή είναι πλήρως λειτουργική και αυτό μπορεί να επιβεβαιωθεί από την αντίστοιχη καταγραφή που έχει γίνει και μπορεί να παρουσιαστεί.

## ΒΙΒΛΙΟΓΡΑΦΙΑ

- [1] M. Bousquet και G. Maral, *ΔΟΡΥΦΟΡΙΚΕΣ ΕΠΙΚΟΙΝΩΝΙΕΣ*, Αθήνα: Εκδόσεις ΤΖΙΟΛΑ, 2014.
- [2] Στ. Μπουλαδάκης, Ν. Ασημόπουλος και Γ. Πατουλίδης, *ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΣ ΜΙΚΡΟΕΛΕΓΚΤΩΝ ΓΙΑ ΜΗΧΑΝΙΚΟΥΣ*, Αθήνα: Εκδόσεις ΤΖΙΟΛΑ, 2011.
- [3] J. Blewitt, *Basics of the GPS Technique*. Sweden: Εκδόσεις Swedish Land Survey, 1997.
- [4] R. B. Langley, “A primer on GPS Antennas,” *GPS world*, pp. 50-54, Jul. 1998.
- [5] R. Rifandi, S. F. Assagaf and Y. Q. Q. K. Ningtyas, “An insight about GPS,” *Reasoning, Proof, and Communication*, 2013.
- [6] X Wang and W. Guo, “The design of RS232 and CAN Protocol converter based on PIC MCU,” *Computer and Information Science*, vol. 2, no. 3, Aug. 2009.
- [7] Microchip, “PIC18F2420/2520/4420/4520 DataSheet,” 2008.

## ΠΑΡΑΡΤΗΜΑ Α : ΚΩΔΙΚΑΣ

```
; Macro : BLDISPLAY
;     Helper macro to write a string to the display
;
; Input:  Address of String to display
;
; Output: W modified, BLDspStr called
;
;*****
BLDISPLAY macro addr
movlw HIGH addr
movwf TBLPTRH
movlw LOW addr
movwf TBLPTL
call BLDspStr
endm
;*****
;
; Function : bootloader
;
;     This routine is always in low memory immediately after the
;     reset and interrupt vectors (along with the remapped vectors.)
;
;     It provides initial hardware setup, detection of bootloader
;     activation request. It will never exceed 0x600-0x1C hex bytes
;
; Input:  None.
;
; Output: Returns if bootloader activation is not required.
;
;*****
Bootloader
clrf PORTB
```

```

clrf LATB

; First, delay for 1/2s for things to settle
; We are running at about 40MHz
movlw D'50'
call UIWait10ms

; do initial bootloader hardware setup ( port directions and levels )

movlw COMP_OFF_VAL
movwf CMCON
movlw CVRCON_OFF_VAL
movwf CVRCON
movlw ADC_OFF_VAL
movwf ADCON1

; Port directions and levels
movlw PORTA_LEVELS
movwf PORTA
movlw PORTA_TRIS_VAL
movwf TRISA
movlw PORTB_LEVELS
movwf PORTB
movlw PORTB_TRIS_VAL
movwf TRISB
movlw PORTC_LEVELS
movwf PORTC
movlw PORTC_TRIS_VAL
movwf TRISC

; Now setup the LCD into 4-bit mode, and clear the display
call DspPutCmd4Bit
movlw D'4'
call UIWait10ms

```

```

movlw 0x28
call DspPutCmd ; 4 bit, 2 line, small font
movlw D'4'
call UIWait10ms
movlw 0x28
call DspPutCmd ; repeat, since first time doesn't always work
movlw 0x00C
call DspPutCmd ; disp on, no cursor
call DspHome
call DspClear

```

```

BLDISPLAY WELCOME_L1

```

```

call DspLine2

```

```

BLDISPLAY WELCOME_L2

```

```

; Display logo for 1s at least

```

```

movlw D'100'

```

```

call UIWait10ms

```

```

; Now look to see if a button is pressed

```

```

movf PORTA, W

```

```

andlw 0x3C

```

```

sublw 0x3C

```

```

btfsc STATUS, Z

```

```

return

```

```

; Debounce period

```

```

movlw D'5'

```

```

call UIWait10ms

```

```

; Now look again to see if a button is still pressed

```

```

movf PORTA, W

```

```

andlw 0x3C

```

```

sublw 0x3C

```

```
btfsc STATUS, Z
return
```

```
; A key was pressed. Update display...
```

```
call DspHome
call DspClear
```

```
BLDISPLAY_DOWNLOAD_L1
```

```
call DspLine2
```

```
BLDISPLAY_DOWNLOAD_L2
```

```
; ... Wait for key to be released
```

```
BL001
```

```
movf PORTA, W
andlw 0x3C
sublw 0x3C
btfss STATUS, Z
bra BL001
```

```
; OK, execute the RS232 Download routine.
```

```
; We never return from this code.
```

```
; Sync with the PC by exchanging 'X' bytes, followed by a G for Go!
```

```
BL002
```

```
movlw 0x01
call UIWait10ms
movlw 'X'
call RS232TxByte
call RS232RxByte
movlw 'X'
subwf hostByte, W
btfss STATUS, Z
bra BL002
```

```

movlw 0x01
call UIWait10ms
movlw 'G'
call RS232TxByte

; Then wait for a flash programming packet. This is
; AAD.....DC
; AA == 16 bit address to write to
; D..D 64 data bytes
; C check byte
; PIC returns Y for ok, C for checkdigit error, F for flash error
; Z for exiting ok
; Send address of 0x0000 ( and no data or check byte ) to end transfer
; Only addresses above bootloader are accepted. TODO - validate addr

```

BLProg

```

call RxFlashBlock
; Termination packet received?
movlw 'Z'
subwf progRxResult, W
btfsc STATUS, Z
bra BLEnd

```

; Error in received packet?

```

movlw 'C'
subwf progRxResult, W
btfsc STATUS, Z
bra BLError

```

call ProgramFlashBlock

; Error in received packet?

```

movlw 'F'

```

```
subwf progRxResult, W
btfsc STATUS, Z
bra BLError2
```

```
; Echo OK to Host
movlw 0x01
call UIWait10ms
movlw 'Y'
call RS232TxByte
bra BLProg
BLEnd
```

```
; Echo exiting to Host
movlw 0x01
call UIWait10ms
movlw 'Z'
call RS232TxByte
call DspHome
call DspClear
```

```
BLDISPLAY REBOOT_L1
call DspLine2
BLDISPLAY REBOOT_L2
movlw D'200'
call UIWait10ms
reset
```

```
BLError
; Echo Error to Host
movlw 0x01
call UIWait10ms
movlw 'C'
call RS232TxByte
```

```
bra BLProg
```

```
BLError2
```

```
; Echo Error to Host
```

```
movlw 0x01
```

```
call UIWait10ms
```

```
movlw 'F'
```

```
call RS232TxByte
```

```
bra BLProg
```

```
;*****
```

```
;
```

```
; Function : LCDWriteDelay
```

```
; Provides a small delay when driving the LCD pins
```

```
;
```

```
; Input: None
```

```
;
```

```
; Output: None
```

```
;
```

```
;*****
```

```
LCDWriteDelay
```

```
nop
```

```
nop
```

```
return
```

```
;*****
```

```
;
```

```
; Function : dspPutCmd4Bit
```

```
; Puts the LCD into 4 bit mode from 8 bit mode
```

```
;
```

```
; Input: Command to write is in W
```

```
;
```

```
; Output: None
```

```

;
;*****
DspPutCmd4Bit
bcf  CONTROL, LCD_RW
call LCDWriteDelay
bcf  CONTROL, LCD_RS
call LCDWriteDelay
bsf  CONTROL, LCD_E
movlw 0x0F
andwf CONTROL, F
movlw 0x20
iorwf CONTROL, F
call LCDWriteDelay
bcf  CONTROL, LCD_E
call LCDWriteDelay
return

;*****
;
; Function : LCDBusy
;      This call blocks until the busy flag in the LCD is clear
;
; Input:  None
;
; Output: None
;
;*****
LCDBusy
movlw DATA_BUS_INPUT
movwf DATA_BUS_TRIS
bcf  CONTROL, LCD_RS ; Set LCD for command mode
call LCDWriteDelay
bsf  CONTROL, LCD_RW ; Setup to read busy flag

```

```

call LCDWriteDelay
bsf CONTROL, LCD_E ; LCD E-line High
call LCDWriteDelay
movf PORTB, W ; Read busy flag + DDram address
andlw 0xF0
movwf lcdTmp2
bcf CONTROL, LCD_E ; LCD E-line Low
call LCDWriteDelay
bsf CONTROL, LCD_E ; LCD E-line High
call LCDWriteDelay
swapf PORTB, W ; Read busy flag + DDram address
andlw 0x0F
addwf lcdTmp2, F
bcf CONTROL, LCD_E ; LCD E-line Low
movf lcdTmp2, W
andlw 0x80 ; Check Busy flag, High = Busy
btfss STATUS, Z
bra LCDBusy
bcf CONTROL, LCD_RW
call LCDWriteDelay
movlw DATA_BUS_OUTPUT
movwf DATA_BUS_TRIS
return

```

```

;*****
;
; Function : DspPutChar
; This call writes a character to the current cursor position.
;
; Input: Character in W
;
; Output: None
;

```

```

,*****
DspPutChar
movwf lcdTmp      ; Character to be sent is in W
call LCDBusy      ; Wait for LCD to be ready
bcf CONTROL, LCD_RW ; Set LCD in read mode
call LCDWriteDelay
bsf CONTROL, LCD_RS ; Set LCD in data mode
call LCDWriteDelay
bsf CONTROL, LCD_E ; LCD E-line High
call LCDWriteDelay
movlw 0x0F
andwf CONTROL, F
movf lcdTmp, W
andlw 0xF0
iorwf CONTROL, F ; Send data to LCD
call LCDWriteDelay
bcf CONTROL, LCD_E ; LCD E-line Low
call LCDWriteDelay
bsf CONTROL, LCD_E ; LCD E-line High
call LCDWriteDelay
movlw 0x0F
andwf CONTROL, F
swapf lcdTmp, W
andlw 0xF0
iorwf CONTROL, F ; Send data to LCD
call LCDWriteDelay
bcf CONTROL, LCD_E ; LCD E-line Low
call LCDWriteDelay
return

,*****
;
; Function : DspPutCmd

```

; Sends a command to the LCD.

;

; Input: Command to write is in W

;

; Output: None

;

\*\*\*\*\*

DspPutCmd

movwf lcdTmp

call LCDBusy

bcf CONTROL, LCD\_RW

call LCDWriteDelay

bcf CONTROL, LCD\_RS

call LCDWriteDelay

bsf CONTROL, LCD\_E

movlw 0x0F

andwf CONTROL, F

movf lcdTmp, W ; Retrieve command byte, send first nibble

andlw 0xF0

iorwf CONTROL, F

call LCDWriteDelay

bcf CONTROL, LCD\_E

movlw 0x0F

call LCDWriteDelay

andwf CONTROL, F

bsf CONTROL, LCD\_E

swpf lcdTmp, W ; Retrieve command byte, send second nibble

andlw 0xF0

iorwf CONTROL, F

call LCDWriteDelay

bcf CONTROL, LCD\_E

call LCDWriteDelay

return

```

;*****
;
; Function : DspClear
;      This call clears the display and returns the cursor position
;      to top left.
;
; Input:  None
;
; Output: None
;

```

```

;*****
DspClear
movlw 0x001
call DspPutCmd
return

```

```

;*****
;
; Function : DspHome
;      This call returns the cursor position to top left.
;
; Input:  None
;
; Output: None
;

```

```

;*****
DspHome
movlw 0x002
call DspPutCmd
return

```

```

;*****

```

```

;
; Function : DspLine2
;       This call positions the cursor at the beginning of the second
;       line
;
; Input:  None
;
; Output: None
;
;*****
DspLine2
movlw  0x0c0      ; pos 40h = 01000000
call  DspPutCmd
return

;*****
;
; Function : UIWait10ms
;       delays for a multiple of 10ms
;
; Input:  multiple in W
;
;*****
UIWait10ms
movwf  delay3
d1ms001
movlw  D'200'
call  UIWait50us
decfsz delay3, F
bra   d1ms001
return

```

```

;*****
;
; Function : waitBitTime
;         Waits for 1 bit time
;         Baud rate is 115200 @ 40MHZ. 1 bit time = 8.68us, say 87 cycles
;
; Input:   None
;
; Output:  None
;
;
;*****

```

```
waitBitTime
```

```
; 2 + 2 + 2 + (d-1)*(3) + 2
```

```
movlw D'27'    ; 1
```

```
movwf bitDelay ; 1
```

```
wbt001
```

```
decfsz bitDelay,F ; 1/2
```

```
bra wbt001 ; 2
```

```
nop
```

```
return
```

```

;*****
;
; Function : waitTxBitTime
;         Waits for 1 bit time, less a constant time of 8 cycles
;         because the loop overhead is 8 cycles
;         Baud rate is 115200 @ 40MHZ 1 bit time = 8.68us, say 87 cycles
;         Therefore we want, including call/return overhead, 79 cycles

```

```

;
; Input:  None
;
; Output: None
;
;
;*****
waitTxBitTime
; 2 + 2 + 2 + (d-1)*(3) + 2
movlw  D'24'      ; 1
movwf  bitDelay   ; 1
wtbt001
decfsz bitDelay,F ; 1/2
bra    wtbt001    ; 2
nop
nop
return

;*****
;
; Function : waitHalfBitTime
;           Waits for 1 bit time
;           Baud rate is 38400 @ 20MHz
;
; Input:  None
;
; Output: None
;
;
;*****
waitHalfBitTime
;           Waits for 1/2 bit time
;           Baud rate is 115200 @ 40MHZ. 1 bit time = 8.68us, say 87 cycles

```

```

;      Therefoe we want, including call/return overhead, 43 cycles
; 2 + 2 + 2 + (d-1)*(3) + 2

```

```

movlw  D'12'      ; 1
movwf  bitDelay   ; 1
whbt001
decfsz bitDelay,F ; 1/2
bra    whbt001    ; 2
nop
nop
return

```

```

;*****

```

```

;

```

```

; Function : UIWait50us

```

```

;      delays for a multiple of 50us

```

```

;

```

```

; Input:  multiple in W

```

```

;      cycles are 2 + 1 + (delay2)*( 1+1 + ((d-1)*3 + 2) + 3) + 3

```

```

;      Need 500 cycles at 40MHz

```

```

;

```

```

;

```

```

;*****

```

```

UIWait50us      ; 2 ( the call itself )

```

```

movwf  delay2   ; 1

```

```

d1us002

```

```

movlw  D'165'   ; 1

```

```

movwf  delay1   ; 1

```

```

d1us001

```

```

decfsz delay1,F ; 1 or 2

```

```

bra    d1us001  ; 2

```

```

decfsz delay2,F ; 1 or 2

```

```

bra    d1us002  ; 2

```

```

return      ; 2

```

```

;*****
;
; Function : RS232TxByte
;         Sends byte in w over the rs32 interface
;         Baud rate is 115200 @ 40MHz
; Input:   byte in W
;
; Output:  None
;
;*****

```

```

RS232TxByte
movwf  hostByte
movlw  0x08
movwf  bitCount
; start bit
bcf   LATA, TX_DATA
rt001
call  waitTxBitTime      ; wait 8.7us - 8 cycles
rrcf  hostByte,F
btfss STATUS, C
bcf   LATA, TX_DATA
btfsc STATUS,C
bsf   LATA, TX_DATA
decfsz bitCount,F
bra   rt001
call  waitTxBitTime      ; wait 8.7us - 8 cycles
nop
nop
nop
nop
nop

```

```

nop
; stop bit
bsf  LATA, TX_DATA
call waitTxBitTime      ; wait 8.7us - 8 cycles
nop
nop
nop
nop
nop
nop
nop
return

```

```

;*****
;
; Function : RS232RxByte
;         Receives a byte over the rs232 interface
;         Baud rate is 115200 @ 40MHz
; Input:   None
;
; Output:  byte in hostByte
;
;         NOTE: This is a blocking call
;

```

```

;*****
RS232RxByte
movlw 0x08
movwf bitCount
clrf  hostByte
re002
btfsc PORTA, RX_DATA
bra  re002
call waitBitTime      ; Past the start bit
call waitHalfBitTime ; to middle of first bit

```

```

re003
bsf  STATUS,C
btfss PORTA, RX_DATA
bcf  STATUS,C
rrcf hostByte, F
call waitTxBitTime    ; wait 1 bit time - 8 cycles
decfsz bitCount,F
bra  re003
return

```

```

;*****
;

```

```

; Function : RxFlashBlock

```

```

;     Receives a 64 byte data block over RS232

```

```

;     Baud rate is 115200 @ 40MHz

```

```

;

```

```

; Input:  None.

```

```

;

```

```

; Output: progAddHigh, progAddLow, progRxResult, prog buffer modified

```

```

;

```

```

;     NOTE: Returns result in progRxResult:

```

```

;     'Y' Block received ok

```

```

;     'C' Check byte error

```

```

;     'Z' Termination requested

```

```

;

```

```

;     Received data is not echoed ( a result byte is echoed later)

```

```

;

```

```

;*****

```

```

RxFlashBlock

```

```

call RS232RxByte

```

```

movf hostByte, W

```

```

movwf progRxCheck ; Init Check digit

```

```

movwf progAddHigh
call RS232RxByte
movf hostByte, W
iorwf progRxCheck, W ; Zero address sent?
btfsc STATUS, Z
bra RXF001 ; Yes - exit now, no more data is sent
movf hostByte, W
movwf progAddLow
addwf progRxCheck, F

; setup pointer to where received data will go
movlw 0x40
movwf FSR0L
clrf FSR0H
RXF000
call RS232RxByte
movf hostByte, W
movwf POSTINC0
addwf progRxCheck, F
btfss FSR0L, 7
bra RXF000

; All bytes received. Now get Check byte, and compare with ours
call RS232RxByte
movf hostByte, W
subwf progRxCheck, W
btfss STATUS, Z
bra RXF002 ; CheckSum Error

; No errors
movlw 'Y'
movwf progRxResult
return

```

```
; Checksum Error
```

```
RXF002
```

```
movlw 'C'
```

```
movwf progRxResult
```

```
return
```

```
RXF001
```

```
movlw 'Z'
```

```
movwf progRxResult
```

```
return
```

```
;*****
```

```
;
```

```
; Function : ProgramFlashBlock
```

```
; Programs a 64 byte data block into the flash
```

```
;
```

```
; Input: None
```

```
;
```

```
; Output: progAddHigh, progAddLow, progRxResult, prog buffer modified
```

```
;
```

```
; NOTE: Returns result in progRxResult:
```

```
; 'Y' Programmed OK
```

```
; 'F' Flash programming error
```

```
;
```

```
; A result byte is echoed to the host later on.
```

```
;
```

```
;*****
```

```
ProgramFlashBlock
```

```
; First, erase the block requested
```

```
clrf TBLPTRU
```

```
movf progAddHigh, W
```

```

movwf TBLPTRH
movf progAddLow, W
movwf TBLPTL

bsf EECON1, EEPGD
bcf EECON1, CFGS
bsf EECON1, WREN
bsf EECON1, FREE
bcf INTCON, GIE

movlw 0x55
movwf EECON2
movlw 0xAA
movwf EECON2
bsf EECON1, WR ; Start erase; CPU stalls until done

; Now program the erased block. We do it in 2, 32byte writes

tblrd*- ; Dummy read decrement

movlw 0x40
movwf FSR0L
clrf FSR0H
movlw D'32'
movwf progCounter

PFB001
movf POSTINC0, W
movwf TABLAT
tblwt+* ; inc address then write
decfsz progCounter, F
bra PFB001
bsf EECON1, EEPGD

```

```

bcf  EECON1, CFGS
bsf  EECON1, WREN

movlw 0x55
movwf EECON2
movlw 0xAA
movwf EECON2
bsf  EECON1, WR      ; Start erase; CPU stalls until done
bcf  EECON1, WREN

```

; Now do next 32 bytes

```

movlw 0x60
movwf FSR0L
clrf  FSR0H

```

```

movlw D'32'
movwf progCounter

```

PFB002

```

movf  POSTINC0, W
movwf TABLAT
tblwt+*      ; inc address then write
decfsz progCounter, F
bra  PFB002

```

```

bsf  EECON1, EEPGD
bcf  EECON1, CFGS
bsf  EECON1, WREN

```

```

movlw 0x55
movwf EECON2
movlw 0xAA
movwf EECON2

```

```

bsf  EECON1, WR      ; Start erase; CPU stalls until done
bcf  EECON1, WREN

```

```

movlw 'Y'
movwf progRxResult

```

```

return

```

```

;*****
;
; Function : BLDspStr
;          Write a string to the display at the current cursor position.
;          This does not handle line wrapping
;
; Input:   TBLPTRH/L setup with string address.
;
; Output:  None
;
;*****

```

```

BLDspStr

```

```

clrf TBLPTRU

```

```

BLD001

```

```

tblrd*+

```

```

movf TABLAT, W

```

```

bz   BLD000

```

```

call DspPutChar

```

```

bra  BLD001

```

```

BLD000

```

```

return

```

WELCOME\_L1

DB " Camera Watch ",0

WELCOME\_L2

DB "Kriselnta Tsatso",0

DOWNLOAD\_L1

DB "Bootloader is",0

DOWNLOAD\_L2

DB "ready for PC",0

REBOOT\_L1

DB "Finished loading",0

REBOOT\_L2

DB "SW resetting...",0