

ΑΛΕΞΑΝΔΡΕΙΟ ΤΕΧΝΟΛΟΓΙΚΟ ΕΚΠΑΙΔΕΥΤΙΚΟ ΙΔΡΥΜΑ

ΘΕΣΣΑΛΟΝΙΚΗΣ

ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΑΥΤΟΜΑΤΙΣΜΟΥ



DEPT. OF AUTOMATION ENGINEERING T.E.
ATEI OF THESSALONIKI

ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ

ΣΥΣΤΗΜΑ ΕΝΤΟΠΙΣΜΟΥ ΘΕΣΗΣ ΡΑΔΙΟΠΟΜΠΟΥ
ΑΠΟ ΜΗ ΕΠΑΝΔΡΩΜΕΝΟ ΕΠΙΓΕΙΟ ΟΧΗΜΑ (ΑΡΓΟΣ)

DETECTION OF RADIO TRANSMITTER LOCATION BY
UNMANNED TERRESTRIAL VEHICLE (ARGOS)



Σπουδαστής:

ΣΤΟΪΚΟΣ ΧΡΙΣΤΟΔΟΥΛΟΣ

Επιβλέπων καθηγητής:

ΜΙΧΑΛΗΣ ΚΙΖΗΡΟΓΛΟΥ

(το θέμα προτάθηκε από τον Δρ. Αλέξανδρο Αστάρα και θα υλοποιηθεί σε συνεργασία με αυτόν)

ΔΗΛΩΣΗ ΦΟΙΤΗΤΗ στο εσώφυλλο (copyright notice)

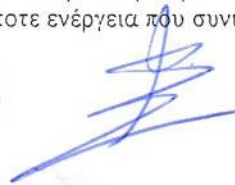
Η παρούσα Πτυχιακή Εργασία και τα συμπεράσματά της, σε οποιαδήποτε μορφή, αποτελούν συνιδιοκτησία του Τμήματος Μηχανικών Αυτοματισμού Τ.Ε. του Αλεξάνδρειου ΤΕΙ Θεσσαλονίκης και του φοιτητή. Οι προαναφερόμενοι διατηρούν το δικαίωμα ανεξάρτητης χρήσης και αναπαραγωγής (τμηματικά ή συνολικά) για διδακτικούς και ερευνητικούς σκοπούς. Σε κάθε περίπτωση πρέπει να αναφέρεται ο τίτλος, ο συγγραφέας, ο επιβλέπων και το τμήμα του ΑΤΕΙΘ.

Η έγκριση της παρούσας Πτυχιακής Εργασίας από το Τμήμα Μηχανικών Αυτοματισμού Τ.Ε. δεν υποδηλώνει απαραίτητως και αποδοχή των απόψεων του συγγραφέα εκ μέρους του Τμήματος.

Ο υπογεγραμμένος δηλώνω υπεύθυνα ότι η παρούσα Πτυχιακή Εργασία είναι εξ' ολοκλήρου δικό μου έργο και συγγράφηκε ειδικά για τις απαιτήσεις του προγράμματος σπουδών του Τμήματος Μηχανικών Αυτοματισμού Τ.Ε.

Δηλώνω υπεύθυνα ότι κατά τη συγγραφή ακολούθησα την πρόπουσα ακαδημαϊκή δεοντολογία αποφυγής λογοκλοπής και έχω απούγει οποιαδήποτε ενέργεια που συνιστά παράπτωμα λογοκλοπής.

Στόικος Χριστόδουλος



27/01/2018

(Όνομα, Υπογραφή, Ημερομηνία)

ΠΙΝΑΚΑΣ ΠΕΡΙΕΧΟΜΕΝΩΝ

ΠΕΡΙΛΗΨΗ	4
ABSTRACT	4
ΕΠΕΞΗΓΗΣΗ ΑΚΡΩΝΥΜΙΩΝ	5
1. ΕΙΣΑΓΩΓΗ	6
ΙΣΤΟΡΙΚΗ ΑΝΑΔΡΟΜΗ.....	6
2. ΜΗ ΕΠΑΝΔΡΩΜΕΝΑ ΑΥΤΟΝΟΜΑ ΟΧΗΜΑΤΑ	8
2.1. ΠΕΡΙΓΡΑΦΗ ΜΗ ΕΠΑΝΔΡΩΜΕΝΩΝ ΟΧΗΜΑΤΩΝ.....	8
2.2. ΕΙΔΗ ΠΠΤΑΜΕΝΩΝ ΜΗΧΑΝΩΝ.....	8
2.3. ΜΕΛΛΟΝΤΙΚΕΣ ΧΡΗΣΕΙΣ	8
2.4. ΧΡΗΣΗ ΤΟΥ ΟΡΟΥ “DRONE”.....	8
3. ΠΕΡΙΓΡΑΦΗ ΥΛΙΚΟΥ	9
3.1. ARDUINO ΠΛΑΤΦΟΡΜΑ	9
3.1.1. Ο ΜΙΚΡΟΕΛΕΓΚΤΗΣ ΤΟΥ ARDUINO.....	9
3.1.2. ARDUINO MEGA (rev3)	10
3.1.3. ΤΡΟΦΟΔΟΣΙΑ	11
3.1.4. ΓΛΩΣΣΑ ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΥ	12
3.2. ΑΥΤΟΝΟΜΟ ΕΠΙΓΕΙΟ ΟΧΗΜΑ – ΠΛΑΤΦΟΡΜΑ ΚΙΝΗΣΕΩΣ	13
3.2.1. ΣΚΑΦΟΣ ΟΧΗΜΑΤΟΣ QUANUM VANDAL	13
3.2.2. ΚΙΝΗΤΗΡΑΣ ΟΧΗΜΑΤΟΣ	13
3.2.3. ΗΛΕΚΤΡΟΝΙΚΟΣ ΕΛΕΓΚΤΗΣ ΤΑΧΥΤΗΤΑΣ ΚΙΝΗΤΗΡΑ	14
3.2.4. ΤΡΟΦΟΔΟΣΙΑ ΟΧΗΜΑΤΟΣ	15
3.3. ΣΥΣΤΗΜΑ ΑΥΤΟΜΑΤΗΣ ΠΛΟΗΓΗΣΗΣ	15
3.3.1. ΕΛΕΓΚΤΗΣ ΑΥΤΟΜΑΤΗΣ ΠΛΟΗΓΗΣΗΣ – ΑΥΤΟΜΑΤΟΣ ΠΙΛΟΤΟΣ	15
3.3.2. ΑΠΑΡΑΙΤΗΤΑ ΠΕΡΙΦΕΡΕΙΑΚΑ ΑΥΤΟΜΑΤΗΣ ΠΛΟΗΓΗΣΗΣ	16
4. ΠΕΡΙΓΡΑΦΗ ΛΟΓΙΣΜΙΚΟΥ	18

4.1. ΜΕΘΟΔΟΣ ΥΠΟΛΟΓΙΣΜΟΥ ΚΕΝΤΡΟΥ ΒΑΡΟΥΣ ΓΕΩΓΡΑΦΙΚΩΝ ΣΥΝΤΕΤΑΓΜΕΝΩΝ	18
4.2. ΠΡΟΓΡΑΜΜΑ ΕΙΚΟΝΙΚΟΥ ΤΡΙΓΩΝΙΣΜΟΥ	19
4.3. ΠΡΟΓΡΑΜΜΑ ΣΥΣΤΗΜΑΤΟΣ ΕΚΠΟΜΠΗΣ ΜΗΝΥΜΑΤΟΣ ΔΙΑΣΩΣΗΣ	65
ΣΥΜΠΕΡΑΣΜΑΤΑ – ΠΑΡΑΤΗΡΗΣΕΙΣ	68
ΕΞΕΛΙΞΗ	69
ΚΟΣΤΟΛΟΓΙΟ	69
ΕΥΧΑΡΙΣΤΙΕΣ	70
ΒΙΒΛΙΟΓΡΑΦΙΑ – ΠΗΓΕΣ.....	70

ΠΕΡΙΛΗΨΗ

Στην πτυχιακή αυτή εργασία υλοποίησα ένα φορητό σύστημα συλλογής δεδομένων πορείας για χρήση μέσα σε μοντέλο οχήματος μικρής κλίμακας. Ο συντονισμός του ηλεκτροκίνητου οχήματος θα γίνεται μέσω μικροελεγκτή (αυτόματου πιλότου - ardupilot) με πρόσθετη δυνατότητα ραδιοκυματικού τηλεχειρισμού ή ασύρματης τηλε-διαχείρισης μέσω υπολογιστή. Ο μικροελεγκτής θα συλλέγει δεδομένα (στίγμα GPS, κλίσεις 3 αξόνων, κατεύθυνση ψηφιακής πυξίδας) για την απαραίτητη κίνηση του οχήματος και ένας δεύτερος μικροελεγκτής θα παρακολουθεί για τυχόν λήψη σήματος από τον επίγειο ραδιοπομπό μικρής εμβέλειας, θα καταγράφει απαραίτητα δεδομένα θέσης και θα υπολογίζει την εκτιμώμενη θέση του επίγειου πομπού. Τα συλλεγόμενα δεδομένα (θέσης, στίγματος GPS, ώρας, εκτιμώμενη θέση επίγειου πομπού) θα αποθηκεύονται σε κάρτα μνήμης.

ABSTRACT

Abstract: In this degree thesis, I implemented an automated system on top of an unmanned ground vehicle (UGV) in order to monitor the signal from a transmitter of unknown position, within certain limits, thereby keeping track of the vehicle's movements. The design and programming of the system has been implemented by taking into consideration the simplicity and cost effectiveness of the whole project and the lowest possible power consumption of the transmitter. Next, I developed the positioning calculation algorithm in a way that the output is as close as possible to the real position of the transmitter and by storing the position in case of failure.

AC	Alternating Current
ADC	Analog-to-digital converter
AREF	Analog Reference
CPU	Central Processing Unit
DC	Direct Current
DTR	Data Transfer Request
EEPROM	Electrically Erasable Programmable Read – Only Memory
FET	Field-Effect Transistor
FTDI	Future Technology Devices International
GND	Ground
GNU	Gnu’s Not Unix
GPS	Global Positioning System
I2C	Inter-Integrated Circuit
ICSP	In-Circuit Serial Programming
IDE	Integrated Development Environment
I/O	Input / Output
LCD	Liquid Crystal Display
LED	Light-Emitting Diode
OS	Operation System
PWM	Pulse Width Modulation
SPI	Serial Peripheral Interface
SRAM	Static Random-Access Memory
TTL	Transistor – Transistor Logic
UART	Universal Asynchronous Receiver / Transmitter
USB	Universal Serial Bus

1. ΕΙΣΑΓΩΓΗ

Ένα σύστημα εντοπισμού θέσης ραδιοπομπού, χρειάζεται ένα σύστημα εκπομπής ραδιοσήματος και ένα σύστημα λήψης αυτού του ραδιοσήματος. Η βασική αυτή μορφή του συστήματος δεν προσφέρει σχεδόν καμία ακρίβεια, αλλά είναι η πιο απλή μορφή και με το χαμηλότερο κόστος κατασκευής. Για να επιτευχθεί ακρίβεια χρειάζονται τουλάχιστον 2 συστήματα λήψης, ενώ για να έχουμε ικανοποιητική και αξιοποιήσιμη ακρίβεια θέλουμε 3 ή και περισσότερα συστήματα λήψης στα οποία γνωρίζουμε ανά πάσα στιγμή την θέση τους.

Με την χρήση ενός συστήματος λήψης του ραδιοσήματος, θα πρέπει να πραγματοποιήσουμε προσομοίωση των επιπλέον συστημάτων λήψης με επιπλέον εικονικά συστήματα λήψης. Ο πιο απλός τρόπος είναι η μετακίνηση του ενός και μοναδικού συστήματος λήψης που διαθέτουμε, σε γνωστές θέσεις και χρόνους. Αυτό μας δίνει την δυνατότητα να απλοποιήσουμε την δημιουργία σε ένα συνολικό σύστημα που θα είναι και πιο οικονομικό στην κατασκευή του.

Το πιο διαδεδομένο σύστημα εντοπισμού θέσης, είναι το γνωστό σε όλους μας GPS (Global Positioning System).

ΙΣΤΟΡΙΚΗ ΑΝΑΔΡΟΜΗ

Τα σημεία του ορίζοντα, ή ακόμη και τα αστέρια, χρησιμοποιούνταν από την αρχαιότητα για τον προσανατολισμό των ανθρώπων. Ένα σταθερό άστρο στον ουρανό, με γνωστή γεωγραφική θέση ως προς το σημείο παρατήρησης, αποτελούσε σημείο αναφοράς και βοηθούσε τους ανθρώπους στο να βρουν τη σωστή πορεία τους. Στον προσανατολισμό συνέβαλαν αργότερα και άλλα μέσα, όπως η πυξίδα και ο εξάντας. Ωστόσο ο εξάντας είναι εύχρηστος μόνο για τον προσδιορισμό του γεωγραφικού πλάτους, ενώ η χρήση του για τον προσδιορισμό του γεωγραφικού μήκους είναι δύσκολη και εξαιρετικά σύνθετη, πράγμα που αποτελεί ένα σημαντικό μειονέκτημα για προσδιορισμό του στίγματος στην θάλασσα. Ως αποτέλεσμα, τον 17ο αιώνα, το Ηνωμένο Βασίλειο συνέστησε ένα συμβούλιο επιστημόνων, το οποίο θα επιβράβευε χρηματικά όποιον θα μπορούσε να εφεύρει ένα όργανο, το οποίο θα επέτρεπε τον ακριβή υπολογισμό και των δύο γεωγραφικών συντεταγμένων, δηλαδή μήκους και πλάτους.

Το 1761, ο Άγγλος ωρολογοποιός Τζον Χάρισσον (John Harrison), ύστερα από προσπάθειες δώδεκα ετών, κατασκεύασε ένα όργανο, το οποίο δεν ήταν άλλο από το γνωστό σημερινό χρονόμετρο. Σε συνδυασμό με τον εξάντα, το χρονόμετρο επέτρεπε τον υπολογισμό του στίγματος των πλοίων με εξαιρετική ακρίβεια (για τα δεδομένα της εποχής). Πέρασαν αρκετά χρόνια μέχρι να δημιουργηθούν τα πρώτα συστήματα εντοπισμού θέσης που βασίζονταν σε ηλεκτρομαγνητικά κύματα (ραντάρ, στα μέσα του 20ού αιώνα. Τα συστήματα αυτά χρησιμοποιήθηκαν ευρύτατα κατά τη διάρκεια του Δευτέρου Παγκοσμίου Πολέμου (και χρησιμοποιούνται ακόμη). Τα συστήματα εντοπισμού θέσης της εποχής αποτελούνταν από ένα δίκτυο σταθμών βάσης και κατάλληλους δέκτες.

Ανάλογα με την ισχύ του σήματος που λάμβανε κάθε δέκτης από σταθμούς γνωστής γεωγραφικής θέσης, σχηματίζονταν δύο ή περισσότερες συντεταγμένες, μέσω των οποίων προσδιοριζόταν η θέση των σημείων ενδιαφέροντος επάνω σε ένα χάρτη. Στην περίπτωση αυτή, όμως, συνέβαιναν υπήρχαν δύο διαφορετικά προβλήματα: Στην πρώτη περίπτωση η χρήση σταθμών βάσης, που θα εξέπεμπαν σήμα σε υψηλή συχνότητα, διέθεταν μεν υψηλή ακρίβεια εντοπισμού, αλλά είχαν μικρή εμβέλεια. Στη δεύτερη περίπτωση συνέβαινε το ακριβώς αντίθετο, δηλαδή ο σταθμός βάσης χρησιμοποιούσε μεν χαμηλή συχνότητα εκπομπής σήματος, προσφέροντας έτσι υψηλότερη εμβέλεια, αλλά και η ακρίβεια που παρείχε ήταν χαμηλή.

Έστω και με αυτά τα προβλήματα, η αρχή της χρήσης ραδιοκυμάτων για τον εντοπισμό της θέσης ενός σημείου είχε ήδη γίνει. Το Global Positioning System στη σημερινή του μορφή βασίζεται σε παρεμφερή τεχνολογία. Συνδυάζει όλες τις μεθόδους που είχαν χρησιμοποιηθεί στον ουρανό, δηλαδή την τεχνολογία των ηλεκτρομαγνητικών κυμάτων καθώς και την παρατήρηση ενός –τεχνητού αυτή τη φορά- ουράνιου σώματος. Οι σταθμοί βάσης που λαμβάνουν και δέχονται τα απαραίτητα ηλεκτρομαγνητικά κύματα δεν είναι πλέον επίγειοι, αλλά εδρεύουν σε δορυφόρους.

Ένα δίκτυο πολυάριθμων (24 - 32) δορυφόρων που βρίσκεται σε σταθερή θέση γύρω από τον πλανήτη μας, βοηθά τους δέκτες GPS να παράξουν το ακριβές στίγμα ενός σημείου οπουδήποτε στον κόσμο. Όταν, το 1957, πραγματοποιήθηκε η εκτόξευση του δορυφόρου Σπούτνικ, οι άνθρωποι είχαν ήδη αντιληφθεί ότι ένα τεχνητό ουράνιο σώμα κοντά στη Γη είναι δυνατό να χρησιμοποιηθεί για να εντοπιστεί η θέση ενός σημείου πάνω στον πλανήτη. Αμέσως μετά την εκτόξευσή του, οι ερευνητές του Ινστιτούτου Τεχνολογίας της Μασαχουσέτης (MIT) διαπίστωσαν ότι το σήμα που λαμβανόταν από τον δορυφόρο αυξανόταν καθώς αυτός πλησίαζε προς το επίγειο σημείο παρατήρησης και μειωνόταν όταν ο δορυφόρος απομακρυνόταν από αυτό. Αυτό ήταν και το πρώτο βήμα για την υλοποίηση της τεχνολογίας που σήμερα αποκαλείται Global Positioning System. Με τον ίδιο τρόπο που η θέση ενός δορυφόρου μπορούσε να εντοπιστεί ανάλογα με την ισχύ του σήματος που λαμβάνεται από αυτόν, υπήρχε και η δυνατότητα να συμβεί το ακριβώς αντίθετο: Ο δορυφόρος να εντοπίσει την ενός σημείου θέση με ιδιαίτερη ακρίβεια. Στην πραγματικότητα ένας δορυφόρος δεν είναι αρκετός για να υπάρξουν ακριβή αποτελέσματα, αλλά απαιτούνται τουλάχιστον τρεις.

Το GPS αρχικά δημιουργήθηκε αποκλειστικά για στρατιωτική χρήση και ανήκε στη δικαιοδοσία του αμερικανικού Υπουργείου Εθνικής Άμυνας. Στα μέσα της δεκαετίας του 1960 το σύστημα δορυφορικής πλοήγησης, γνωστό τότε με την ονομασία Transit System, χρησιμοποιήθηκε ευρέως από το αμερικανικό ναυτικό. Απαιτήθηκαν αρκετές δεκαετίες, μέχρι δηλαδή τα μέσα της δεκαετίας του 1990, ώστε το σύστημα GPS να εξελιχθεί, να γίνει ιδιαίτερα ακριβές και να αρχίσει να διατίθεται για ελεύθερη χρήση από το ευρύ κοινό.

2. ΜΗ ΕΠΑΝΔΡΩΜΕΝΑ ΑΥΤΟΝΟΜΑ ΟΧΗΜΑΤΑ

2.1. Περιγραφή μη επανδρωμένων οχημάτων

Το πλέον διαδεδομένο είδος αυτόνομων οχημάτων, είναι τα ιπτάμενα. Τα μη επανδρωμένα ιπτάμενα οχήματα είναι το είδος που αποτελεί την βάση για την εξέλιξη όλων των μη επανδρωμένων οχημάτων και μπορούμε να αναφερθούμε σε αυτά για να μελετήσουμε και τα επίγεια.

Αναλυτικότερα, μη επανδρωμένες ιπτάμενες μηχανές (Unmanned Aerial vehicles - UAV ή Remotely piloted systems - RPS), ονομάζονται τα κάθε είδους ιπτάμενα μηχανήματα που κινούνται: 1) στον αέρα (πάνω από την επιφάνεια της Γης), 2) στην επιφάνεια της θάλασσας και 3) υπό τη θάλασσα, αυτόνομα (χωρίς πιλότο ή κυβερνήτη), προγραμματισμένα ή τηλεκατευθυνόμενα, (αερομοντέλα) σε μορφή μικρού αεροπλάνου ή ελικοπτέρου με έναν ή περισσότερους κινητήρες και έλικες συντονισμένους για πλήρως ελεγχόμενη πτήση από ειδικό πρόγραμμα ή χειριστήριο εδάφους.

2.2. Είδη ιπτάμενων μηχανών

Τα είδη των ιπτάμενων μη επανδρωμένων μηχανών χωρίζονται σε δύο βασικές κατηγορίες: 1. Με σταθερές πτέρυγες, όπως τα αεροπλάνα. 2. Με έλικες όπως τα ελικόπτερα.

Για τον προσδιορισμό του αριθμού των κινητήρων ή ελίκων χρησιμοποιείται ο όρος Multicopter ή Multirotor και διαχωρίζονται σε Quadcopter (με 4 έλικες), Hexacopter, Octacopter κλπ.

Οι ιπτάμενες μηχανές με το όνομα drones χρησιμοποιούνται μέχρι σήμερα για πολεμικούς σκοπούς (αναγνωριστικούς, κατασκοπευτικούς ή και για μεταφορά βομβών) ή για ερασιτεχνικούς και ερευνητικούς σκοπούς.

2.3. Μελλοντικές χρήσεις

Μελετάται η μελλοντική χρήση μη επανδρωμένων ιπτάμενων μηχανών για ειρηνικούς σκοπούς, όπως μεταφορά φαρμάκων, βιολογικών υλικών για ιατρικές εξετάσεις και τροφίμων από και προς δύσβατες περιοχές, αξιοποιώντας τις μελέτες και την τεχνολογία η οποία τείνει να καθιερωθεί υπό τον όρο MatterNet αναφερόμενη στην εφαρμογή αυτόματου τρόπου μετακίνησης υλικών.

2.4. Χρήση του όρου “Drone”

Σχετικά με τον όρο «drone» και τη μη χρήση του όρου αυτού για ειρηνικούς σκοπούς εξηγεί ο Michael Toscano, πρόεδρος, της Association for Unmanned Vehicle Systems International. Στο βίντεο "A New Technology Takes Flight" του Michael Toscano και της Missy Cummings - Associate Professor of Aeronautics and Astronautics, MIT, αναφέρονται στη χρήση του όρου, άλλα και στο civilian cargo όπως αυτό θα διαμορφωθεί τα επόμενα 10-15 χρόνια, για το οποίο έχουν δείξει ενδιαφέρον μεγάλες διεθνείς εταιρείες μεταφορών όπως η Fedex και η UPS.

3. ΠΕΡΙΓΡΑΦΗ ΥΛΙΚΟΥ

3.1. ARDUINO ΠΛΑΤΦΟΡΜΑ



Όπως το περιγράφει ο δημιουργός του, το Arduino είναι μια «ανοικτού κώδικα» πλατφόρμα «προτυποποίησης» ηλεκτρονικών βασισμένη σε ευέλικτο και εύκολο στη χρήση hardware και software που προορίζεται για οποιονδήποτε έχει λίγη προγραμματιστική εμπειρία, στοιχειώδεις γνώσεις ηλεκτρονικών και ενδιαφέρεται να δημιουργήσει διαδραστικά αντικείμενα ή περιβάλλοντα.

Στην ουσία, πρόκειται για ένα ηλεκτρονικό κύκλωμα που βασίζεται στον μικροελεγκτή ATmega της Atmel και του οποίου όλα τα σχέδια, καθώς και το software που χρειάζεται για την λειτουργία του, διανέμονται ελεύθερα και δωρεάν ώστε να μπορεί να κατασκευαστεί από τον καθένα. Μπορεί να συμπεριφερθεί σαν ένας μικροσκοπικός υπολογιστής, αφού ο χρήστης μπορεί να συνδέσει επάνω του πολλαπλές μονάδες εισόδου/εξόδου και να προγραμματίσει τον μικροελεγκτή να δέχεται δεδομένα από τις μονάδες εισόδου, να τα επεξεργάζεται και να στέλνει κατάλληλες εντολές στις μονάδες εξόδου. Το κύριο πλεονέκτημά του είναι η τεράστια κοινότητα που το υποστηρίζει και η οποία έχει δημιουργήσει, συντηρεί και επεκτείνει ένα ανάλογο μεγέθους OnLine forum.

Από τις επίσημες εκδόσεις (Duemilanove, Diecimila, Nano, Mega, Bluetooth, LilyPad, Mini, Mini USB, Pro, Pro Mini, Serial και Serial SS) συνιστάται κυρίως η αγορά του Arduino Duemilanove ή τουλάχιστον των Diecimila ή Mega επειδή διαθέτουν υποδοχή USB και είναι συμβατές με τα shields.

3.1.1. Ο ΜΙΚΡΟΕΛΕΓΚΤΗΣ ΤΟΥ ARDUINO (Σχήμα 3.1)

Το Arduino βασίζεται στον ATmega328, έναν 8bit RISC μικροελεγκτή, ο οποίος χρονίζει στα 16MHz. Ο ATmega328 διαθέτει ενσωματωμένη μνήμη τριών τύπων:

- 2Kb μνήμης SRAM που είναι η ωφέλιμη μνήμη που μπορούν να χρησιμοποιήσουν τα προγράμματα για να αποθηκεύουν μεταβλητές, πίνακες κ.λπ. κατά την εκτέλεση τους. Όπως και σε έναν υπολογιστή, αυτή η μνήμη χάνει τα δεδομένα της όταν η παροχή ρεύματος στο Arduino σταματήσει ή αν γίνει reset.
- 1Kb μνήμης EEPROM η οποία μπορεί να χρησιμοποιηθεί για «ωμή» εγγραφή/ανάγνωση δεδομένων (χωρίς datatype) ανά byte από τα προγράμματά σας κατά το runtime. Σε αντίθεση με την SRAM, η EEPROM δεν χάνει τα περιεχόμενά της με



Σχήμα 3.1
 Μικροελεγκτής ATMEGA 328
 Χρησιμοποιείται ως CPU στον
 arduino

απώλεια τροφοδοσίας ή reset οπότε είναι το ανάλογο του σκληρού δίσκου. Για αυτό την χρησιμοποιούμε και εμείς κατά την εκτέλεση του προγράμματος μας για να αποθηκεύονται τα αποθέματα των προϊόντων, και να μην χάνονται ακόμα και αν το μηχάνημα δεν τροφοδοτείται.

- 32Kb μνήμης Flash, από τα οποία τα 2Kb χρησιμοποιούνται από το firmware του Arduino που έχει εγκαταστήσει ήδη ο κατασκευαστής του. Τα υπόλοιπα 30Kb της μνήμης Flash χρησιμοποιούνται για την αποθήκευση των προγραμμάτων, αφού πρώτα

3.1.2. ARDUINO MEGA (rev3) (Σχήμα 3.2)

□ Το **Arduino Mega (rev3)** είναι η πιο εξελιγμένη έκδοση, με τον μικροελεγκτή Atmega2560 και αρκετά μεγαλύτερο μέγεθος. Κάποια από τα χαρακτηριστικά του είναι:

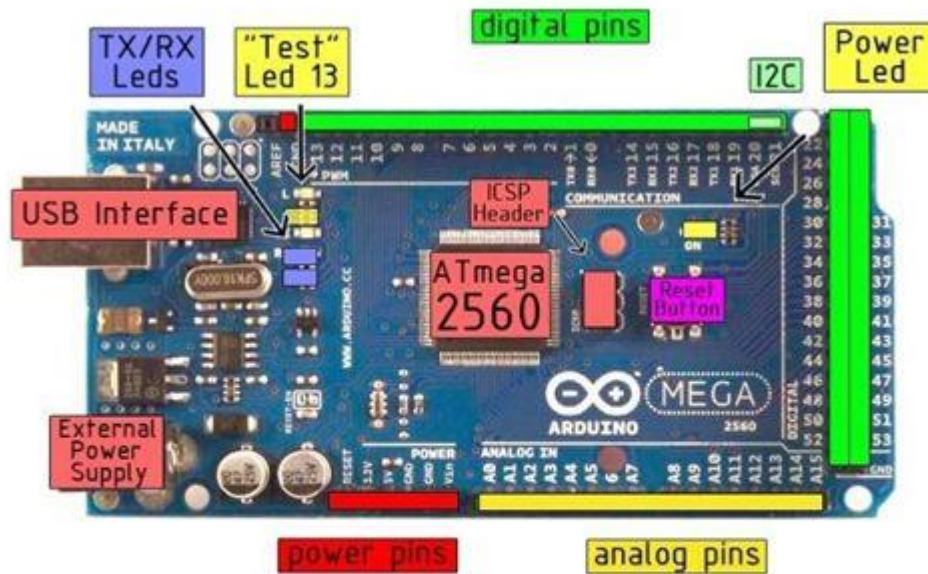
- Τάση τροφοδοσίας 5V
- Δέχεται εξωτερική τάση τροφοδοσίας από 6 – 20V (συνιστώμενη τάση 7 – 12V).
- Τετραπλάσια μνήμη σε σχέση με άλλες εκδόσεις (8Kb SRAM, 4Kb EEPROM, 256Kb Flash).
- DC ρεύμα για I/O pin = 40mA
- 70 ψηφιακά pin εισόδου/εξόδου εκ των οποίων τα 16 λειτουργούν και ως αναλογικές εισοδοί
- Υποστήριξη ψευδοαναλογικής εξόδου PWM σε 14 ψηφιακά pin
- Υποστήριξη εξωτερικού interrupt σε 6 ψηφιακά pin
- 4 σειριακά interface από τα οποία το ένα προωθείται στον ελεγκτή Serial-Over-USB, για σύνδεση με τον υπολογιστή.

Κάθε μία από τις 70 ψηφιακές θύρες στο Mega μπορεί να χρησιμοποιηθεί ως είσοδος ή έξοδος, χρησιμοποιώντας *pinMode ()*, *digitalWrite ()* και *digitalRead ()* λειτουργίες.

Λειτουργούν στα 5V. Επιπλέον, μερικές θύρες έχουν εξειδικευμένες λειτουργίες:

- Serial: Χρησιμοποιείται για να λαμβάνει (RX) και να μεταδίδει (TX) TTL σειριακά δεδομένα.
- Εξωτερικές Διακοπές: Αυτοί οι θύρες μπορούν να ρυθμιστούν για να προκαλέσουν μια διακοπή σε μια χαμηλή τιμή, μια άνοδο ή την πτώση ακμής, ή αλλαγή τιμής.
- PWM: Παρέχει 8-bit έξοδο PWM με την *analogWrite ()* λειτουργία.
- SPI: Αυτές οι θύρες υποστηρίζουν επικοινωνία SPI, η οποία, αν και παρέχεται από το υποκείμενο υλικό, επί του παρόντος δεν περιλαμβάνονται στη γλώσσα Arduino.
- LED: 13. Υπάρχει ένα ενσωματωμένο LED που συνδέεται με την ψηφιακή θύρα 13. Όταν η θύρα είναι HIGH value, η ενδεικτική λυχνία είναι αναμμένη, όταν είναι LOW, δεν είναι αναμμένη.

- I2C: 20 (SDA) και 21 (SCL). Υποστήριξη I2C (TWI) επικοινωνίας γίνεται χρησιμοποιώντας τη βιβλιοθήκη Wire.



Σχήμα 3.2

Arduino MEGA, βελτιωμένη έκδοση του arduino UNO, με περισσότερες I/O και μεγαλύτερη CPU

3.1.3. ΤΡΟΦΟΔΟΣΙΑ

Το Arduino μπορεί να τροφοδοτηθεί με ρεύμα είτε από τον υπολογιστή μέσω της σύνδεσης USB, είτε από εξωτερική τροφοδοσία που παρέχεται μέσω μιας υποδοχής φιν των 2.1mm (θετικός πόλος στο κέντρο) και βρίσκεται στην κάτω-αριστερή γωνία του Arduino. Για να μην υπάρχουν προβλήματα, η εξωτερική τροφοδοσία πρέπει να είναι από 7 ως 12V και μπορεί να προέρχεται από ένα κοινό μετασχηματιστή του εμπορίου, από μπαταρίες ή οποιαδήποτε άλλη πηγή DC.

Δίπλα από τα pin αναλογικής εισόδου, υπάρχει μια ακόμα συστοιχία από 6 pin με την σήμανση POWER. Η λειτουργία του καθενός έχει ως εξής:

Οι θύρες ισχύος είναι οι εξής :

- Η πρώτη, με την ένδειξη RESET, όταν γειωθεί έχει ως αποτέλεσμα την επανεκκίνηση του Arduino.
- Η δεύτερη, με την ένδειξη 3.3V, μπορεί να τροφοδοτήσει εξαρτήματα με τάση 3.3V. Η τάση αυτή δεν προέρχεται από την εξωτερική τροφοδοσία αλλά παράγεται από τον ελεγκτή Serial-Over-USB και έτσι η μέγιστη ένταση που μπορεί να παρέχει είναι μόλις 50mA.
- Το τρίτο, με την ένδειξη 5V, μπορεί να τροφοδοτήσει εξαρτήματά με τάση 5V. Ανάλογα με τον τρόπο τροφοδοσίας του ίδιου του Arduino, η τάση αυτή προέρχεται είτε άμεσα από την

θύρα USB (που ούτως ή άλλως λειτουργεί στα 5V), είτε από την εξωτερική τροφοδοσία αφού αυτή περάσει από ένα ρυθμιστή τάσης για να την «φέρει» στα 5V.

- Το τέταρτο και το πέμπτο pin, με την ένδειξη GND, είναι φυσικά γειώσεις.
- Το έκτο και τελευταίο pin, με την ένδειξη Vin έχει διπλό ρόλο. Σε συνδυασμό με το pin γείωσης δίπλα του, μπορεί να λειτουργήσει ως μέθοδος εξωτερικής τροφοδοσίας του Arduino, στην περίπτωση που δεν σας βολεύει να χρησιμοποιήσετε την υποδοχή του φισ των 2.1mm. Αν όμως έχετε ήδη συνδεδεμένη εξωτερική τροφοδοσία μέσω του φισ, μπορείτε να χρησιμοποιήσετε αυτό το pin για να τροφοδοτήσετε εξαρτήματα με την πλήρη τάση της εξωτερικής τροφοδοσίας (7~12V), πριν αυτή περάσει από τον ρυθμιστή τάσης όπως γίνεται με το pin των 5V.

3.1.4. ΓΛΩΣΣΑ ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΥ

Η γλώσσα του Arduino βασίζεται στη γλώσσα Wiring, μια παραλλαγή C/C++ για μικροελεγκτές αρχιτεκτονικής AVR όπως ο ATmega, και υποστηρίζει όλες τις βασικές δομές της C καθώς και μερικά χαρακτηριστικά της C++. Για compiler χρησιμοποιείται ο AVR gcc και ως βασική βιβλιοθήκη C χρησιμοποιείται η AVR libc. Λόγω της καταγωγής της από την C, στην γλώσσα του Arduino μπορείτε να χρησιμοποιήσετε ουσιαστικά τις ίδιες βασικές εντολές και συναρτήσεις, με την ίδια σύνταξη, τους ίδιους τύπων δεδομένων και τους ίδιους τελεστές όπως και στην C. Πέρα από αυτές όμως, υπάρχουν κάποιες ειδικές εντολές, συναρτήσεις και σταθερές που βοηθούν για την διαχείριση του ειδικού hardware του Arduino.

Επιπλέον, στην γλώσσα του Arduino κάθε πρόγραμμα αποτελείται από δύο βασικές ρουτίνες ώστε να έχει την γενική δομή:

- Η βασική ρουτίνα **setup()** εκτελείται μια φορά μόνο κατά την εκκίνηση του προγράμματος
- Η βασική ρουτίνα **loop()** περιέχει τον βασικό κορμό του προγράμματος και η εκτέλεσή της επαναλαμβάνεται συνέχεια σαν ένας βρόγχος while(true).

3.2. ΑΥΤΟΝΟΜΟ ΕΠΙΓΕΙΟ ΟΧΗΜΑ – ΠΛΑΤΦΟΡΜΑ ΚΙΝΗΣΕΩΣ

3.2.1. ΣΚΑΦΟΣ ΟΧΗΜΑΤΟΣ QUANUM VANDAL (Σχήμα 4.1)

Το όχημα που χρησιμοποιήθηκε είναι ένα τηλεκατευθυνόμενο τετρακίνητο όχημα ανώμαλου δρόμου με μεταλλική μετάδοση κίνησης. Χρησιμοποιεί αναρτήσεις ελατηρίων με αποσβεστήρες πλήρωσης σιλικόνης, ηλεκτρικό κινητήρα και ψηφιακό ελεγκτή ταχύτητας.

- Μήκος 415 mm
- Πλάτος 255 mm
- Ύψος 142 mm
- Μεταξόνιο 275/280 mm



Σχήμα 4.1

Σκάφος οχήματος Quantum Vandal
Τετρακίνητο όχημα με δυνατότητες κίνησης
σε ανώμαλο έδαφος

3.2.2. ΚΙΝΗΤΗΡΑΣ ΟΧΗΜΑΤΟΣ (Σχήμα 4.2)

Ο κινητήρας που χρησιμοποιήθηκε είναι ενδεδειγμένος για το συγκεκριμένο όχημα και είναι της εταιρείας Turnigy, το μοντέλο TrackStar Waterproof Brushless 1/10 Super Stock Power System 3000Kv/45A

- Τύπος brushless χωρίς καρβονάκια και χωρίς απαίτηση συντήρησης
- Αδιαβροχοποιημένος για χρήση σε υγρά περιβάλλοντα
- Κατηγορία τυλιγμάτων κινητήρα 12.5T
- KV κινητήρα 3000kv
- Μέγεθος κινητήρα 36x50mm
- Διάμετρος άξονα 3.175mm
- Βάρος κινητήρα 195g



Σχήμα 4.2
 Κινητήρας οχήματος,
 ηλεκτρικός τύπου brushless,
 αδιαβροχοποιημένος, κλίμακας
 1:10

3.2.3. ΗΛΕΚΤΡΟΝΙΚΟΣ ΕΛΕΓΚΤΗΣ ΤΑΧΥΤΗΤΑΣ ΚΙΝΗΤΗΡΑ (Σχήμα 4.3)

Τα προτερήματα ενός brushless κινητήρα, δηλαδή η μηδενική συντήρησή του και η απευθείας κίνηση του άξονα, απαιτούν την ύπαρξη ενός ηλεκτρονικού ελεγκτή. Στην συγκεκριμένη περίπτωση επιλέχθηκε και χρησιμοποιήθηκε ο ενδεδειγμένος ελεγκτής για τον προαναφερθέντα κινητήρα, της εταιρίας Turnigy, το μοντέλο Turnigy AE-45A Brushless ESC

- Συνεχές ρεύμα τροφοδοσίας 45A
- Μέγιστο ρεύμα στιγμιαίας ζήτησης 90A
- Για οχήματα κλίμακας 1/10
- Τροφοδοσία από μπαταρίες 2 ή 3 κελιών
- Επιπλέον έξοδος S-BEC 5.0V / 2A
- Μέγεθος 40.5 x 37.5 x 24mm
- Βάρος ελεγκτή: 45g
- Δυνατότητα αναστροφής τάσης
- Θερμική προστασία κινητήρα
- Μέγεθος καλωδίωσης τροφοδοσίας 14AWG
- Μέγεθος καλωδίωσης κινητήρα 14AWG

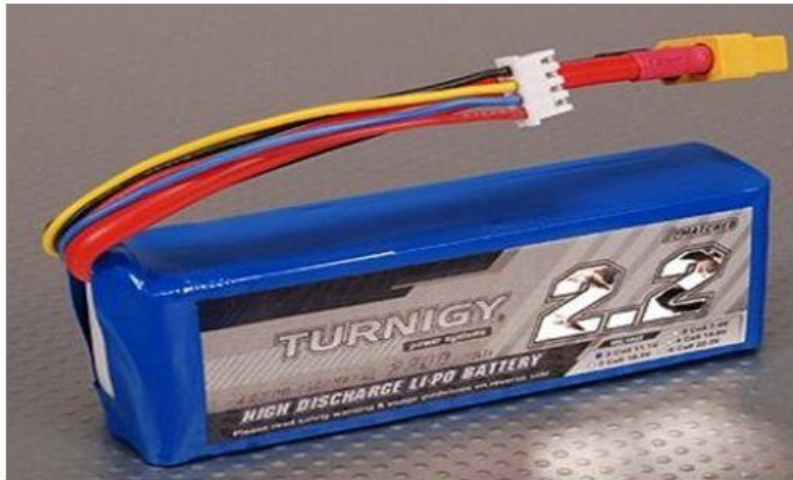


Σχήμα 4.3
 Ηλεκτρονικός ελεγκτής ταχύτητας κινητήρα
 Electronic Speed Controller (ESC)
 Συνεχές ρεύματος 45A (Peak 90A)
 Τροφοδοσία 2 ή 3 κελιών (7,4V ή 11,1V)

3.2.4. ΤΡΟΦΟΔΟΣΙΑ ΟΧΗΜΑΤΟΣ (Σχήμα 4.4)

Η τροφοδοσία του ηλεκτρικού οχήματος γίνεται από μπαταρίες τεχνολογίας πολυμερών στοιχείων λιθίου (Lithium polymer – LiPo). Οι συγκεκριμένες είναι τελευταίας τεχνολογίας, οι πιο αποδοτικές στον τομέα της αποθήκευσης και παροχής ηλεκτρικής ενέργειας, αλλά χρειάζονται ειδικούς φορτιστές και

εξειδικευμένους κύκλους φόρτισης. Χρησιμοποιήθηκαν μπαταρίες 3 κελιών (11.1 V), χωρητικότητας 2200mAh και μέγιστου ρυθμού εκφόρτισης 40 φορές την ονομαστική χωρητικότητα (40C).



Σχήμα 4.4

Μπαταρία Λιθίου πολυμερών στοιχείων
 Lithium Polymer Battery (Li-Po)
 Τριών κελιών 11,1V (3x3,7V)
 Χωρητικότητας 2200mAh
 Μέγιστου ρυθμού εκφόρτισης 25C

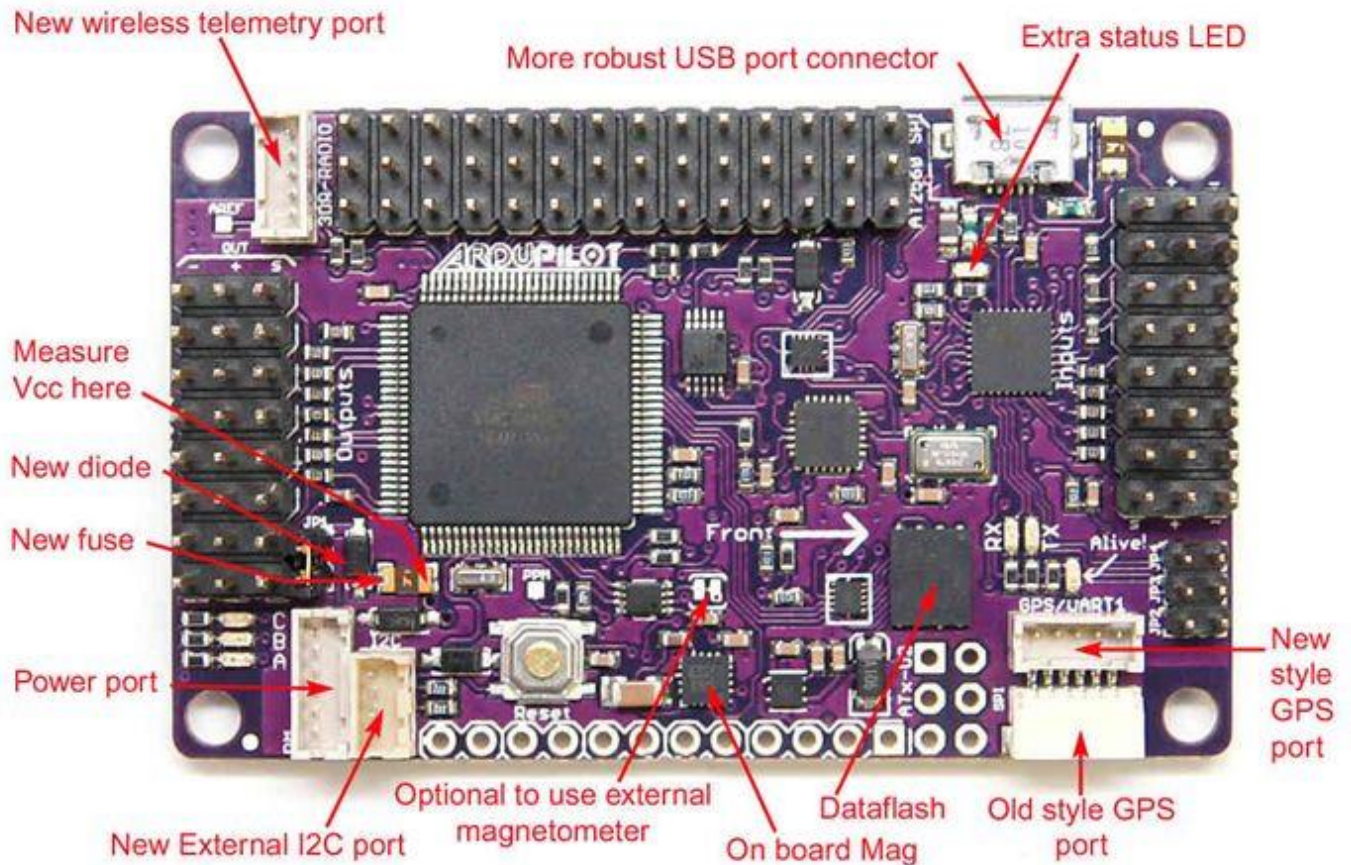
3.3. ΣΥΣΤΗΜΑ ΑΥΤΟΜΑΤΗΣ ΠΛΟΗΓΗΣΗΣ

3.3.1. Ελεγκτής αυτόματης πλοήγησης – Αυτόματος πιλότος (Σχήμα 4.5)

Ο ελεγκτής αυτόματης πλοήγησης είναι ο κορυφαίος για μετατροπή οχημάτων μικρής κλίμακας σε πλήρως αυτόνομα οχήματα. Ο ardupilot 2.6 της εταιρείας 3drobotics χρησιμοποιείται ευρέως στην ανερχόμενη αγορά των αυτόνομων ιπτάμενων οχημάτων (UAV ή Drone) με σκοπό την κινηματογράφηση, την χαρτογράφηση, τους αγώνες, την παρακολούθηση αλλά ακόμα και την απλή ευχαρίστηση. Συνοδεύεται από περιφερειακά όπως ηλεκτρονική πυξίδα, γυροσκόπιο, gps, βαρόμετρο, μετρητή απόστασης και μηχανισμό προβολής ενδείξεων σε εξωτερική οθόνη. Ο έλεγχός του γίνεται κατά κύριο λόγο μέσω ηλεκτρονικού υπολογιστή, ενώ απλή χρήση μπορεί να επιτευχθεί και μέσω τηλεχειρισμού.

- Επεξεργαστής Atmega2560 16MHz
- Εφεδρικός κύκλωμα έκτακτης ανάγκης με πολυπλέκτη και επεξεργαστή ATmega328
- Τάση τροφοδοσίας 5V
- SRAM 8KB
- EEPROM 4KB
- Μνήμη Flash 4MB
- DC ρεύμα για I/O pin = 40mA

- 40 ψηφιακά pin εισόδου/εξόδου
- 16 αναλογικές εισοδοι
- Υποστήριξη ψευδοαναλογικής εξόδου PWM σε 14 ψηφιακά pin
- 4 σειριακά interface τα οποία χρησιμοποιούνται αποκλειστικά για τηλεμετρία (η σύνδεση με ηλεκτρονικό υπολογιστή λαμβάνεται ως τηλεμετρία)
- Ενσωματωμένα LEDs για ενδείξεις τροφοδοσίας, ανάγκης, κατάστασης και κλειδώματος στίγματος GPS



Σχήμα 4.5

Ελεγκτής αυτόματης πλοήγησης - Αυτόματος πιλότος Ardupilot 2.6
Συγκεντρώνει πλήθος περιφερειακών και χρησιμοποιείται ευρέως
σε εφαρμογές αυτόνομων οχημάτων μικρού μεγέθους

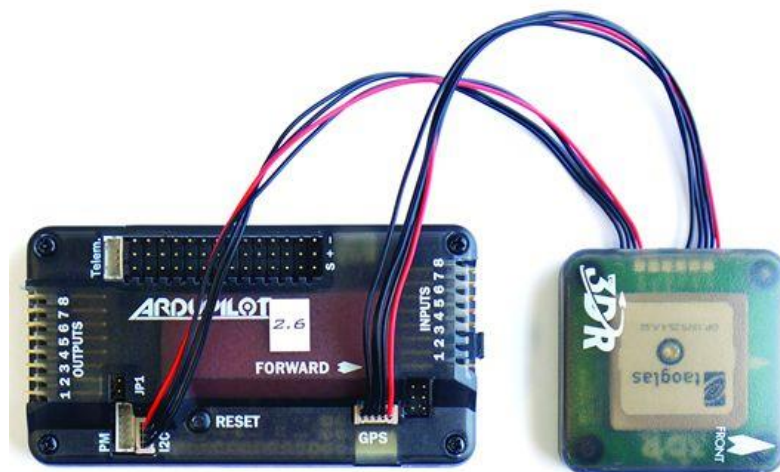
3.3.2. Απαραίτητα περιφερειακά αυτόματης πλοήγησης (Σχήμα 4.6 & 4.7)

Τα περιφερειακά του ελεγκτή αυτόματης πλοήγησης που είναι απαραίτητα για την σωστή λειτουργία του, χαρακτηρίζονται ως *περιφερειακά* διότι απαιτείται να μην βρίσκονται ενσωματωμένα σε έναν κοινό επεξεργαστή, αλλά ενσωματωμένα στην ίδια πλακέτα, ενώ άλλα δεν βρίσκονται σε αυτήν. Με αυτόν τον τρόπο αποφεύγονται παρεμβολές σε αυτά.

- Ψηφιακή πυξίδα με chip HMC5883L
- Αισθητήρας βαρομετρικής πίεσης MS5611-01BA03
- Γυροσκόπιο 6 αξόνων
- Επιταχυνσιόμετρο MPU-6000
- Συσκευή GPS uBlox LEA-6H



Σχήμα 4.6
 Εξωτερικό σύστημα λήψης δεδομένων GPS με ενσωματωμένη ψηφιακή πυξίδα



Σχήμα 4.7
 Ardupilot 2.6 συνδεδεμένος με το εξωτερικό GPS και την ψηφιακή πυξίδα για την αποφυγή παρεμβολών από τα υπόλοιπα περιφερειακά, μέσα στις θήκες προστασίας τους

4. ΠΕΡΙΓΡΑΦΗ ΛΟΓΙΣΜΙΚΟΥ

4.1. ΜΕΘΟΔΟΣ ΥΠΟΛΟΓΙΣΜΟΥ ΚΕΝΤΡΟΥ ΒΑΡΟΥΣ ΓΕΩΓΡΑΦΙΚΩΝ ΣΥΝΤΕΤΑΓΜΕΝΩΝ.

Κάθε σημείο πάνω στον πλανήτη Γη αποτελείται από 2 συντεταγμένες :

- α) Γεωγραφικό Πλάτος
- β) Γεωγραφικό Μήκος

Αυτές οι συντεταγμένες μπορούν να προβληθούν σε ένα καρτεσιανό σύστημα συντεταγμένων 3 αξόνων με αρχή των αξόνων το κέντρο της Γης.

Τα γεωγραφικά σημεία που αποτελούν τις κορυφές του πολυγώνου, με την σειρά τους μπορούν να μετατραπούν σε σημεία καρτεσιανού συστήματος 3 αξόνων (X,Y,Z) με κέντρο 0,0,0 το κέντρο της Γης.

Σε πολύ μικρές εκτάσεις, μπορούμε να θεωρήσουμε ότι η έκταση που μας ενδιαφέρει είναι επίπεδη και δεν παρουσιάζει καμπυλότητα λόγω του σφαιρικού σχήματος της Γης. Έτσι, το βάρος του κάθε σημείου είναι ίσο με 1 και έχουμε ένα πολύγωνο 2 διαστάσεων.

Στον συγκεκριμένο υπολογισμό, παρότι δεν μας ενδιαφέρει η καμπυλότητα λόγω των μικρών αποστάσεων, στο κέντρο βάρους που προκύπτει από το καρτεσιανό σύστημα των 3 αξόνων λαμβάνεται υπόψιν και η προβολή του σημείου που προκύπτει, στην επιφάνεια της Γης.

Συνοπτικά, τα βήματα που ακολουθούνται για τον υπολογισμό του κέντρου βάρους, είναι τα εξής:

1. Μετατροπή του γεωγραφικού μήκους και του γεωγραφικού πλάτους σε ακτίνια.
 (Γεωγραφικό πλάτος σε ακτίνια) = (Γεωγραφικό πλάτος σε μοίρες) * (π) / 180
 (Γεωγραφικό μήκος σε ακτίνια) = (Γεωγραφικό μήκος σε μοίρες) * (π) / 180
2. Μετατροπή του γεωγραφικού μήκους και του γεωγραφικού πλάτους σε καρτεσιανές συντεταγμένες.
 $X1 = \text{συνημίτονο}(\text{Γεωγραφικό πλάτος σε ακτίνια}) * \text{συνημίτονο}(\text{Γεωγραφικό μήκος σε ακτίνια})$
 $Y1 = \text{συνημίτονο}(\text{Γεωγραφικό πλάτος σε ακτίνια}) * \text{ημίτονο}(\text{Γεωγραφικό μήκος σε ακτίνια})$
 $Z1 = \text{ημίτονο}(\text{Γεωγραφικό πλάτος σε ακτίνια})$
3. Επαναλαμβάνουμε τα παραπάνω βήματα για το σύνολο των γεωγραφικών σημείων.
4. Υπολογισμός συνολικού βάρους των σημείων.
Επειδή τα βάρη μας είναι όλα ίσα με 1, το συνολικό βάρος ισούται με τον συνολικό αριθμό των σημείων.
5. Υπολογισμός των μέσων σημείων για κάθε καρτεσιανό άξονα.
 $X = (X1 + X2 + \dots + Xn) / n$
 $Y = (Y1 + Y2 + \dots + Yn) / n$
 $Z = (Z1 + Z2 + \dots + Zn) / n$

6. Μετατροπή των καρτεσιανών συντεταγμένων 3 αξόνων (X, Y, Z) σε γεωγραφικό πλάτος και γεωγραφικός μήκος, σε ακτίνια.

Γεωγραφικό μήκος σε ακτίνια = Αντίστροφη εφαπτομένη(Y, X)

Υποτείνουσα = $\sqrt{(X^2 * Y^2)}$

Γεωγραφικό πλάτος σε ακτίνια = Αντίστροφη εφαπτομένη(Z, Υποτείνουσα)

7. Μετατροπή του γεωγραφικού μήκους και του γεωγραφικού πλάτους από ακτίνια σε μοίρες.

(Γεωγραφικό πλάτος σε μοίρες) = (Γεωγραφικό πλάτος σε ακτίνια) * 180 / (π)

(Γεωγραφικό μήκος σε μοίρες) = (Γεωγραφικό μήκος σε ακτίνια) * 180 / (π)

4.2. ΠΡΟΓΡΑΜΜΑ ΕΙΚΟΝΙΚΟΥ ΤΡΙΓΩΝΙΣΜΟΥ

4.2.1 Πρόγραμμα λήψης και εικονικού τριγωνισμού – Προσθήκη βιβλιοθηκών

```

1 #include <VirtualWire.h>
2 #include <TinyGPS.h>
3 #include <SD.h>
4 #include <stdlib.h>

```

1 ^η γραμμή	Συμπεριλαμβάνω την βιβλιοθήκη VirtualWire η οποία είναι απαραίτητη για την δημιουργία του εικονικού καναλιού επικοινωνίας και την διαχείριση της λήψης των πληροφοριών μέσω του δέκτη, δηλαδή του μηνύματος διάσωσης
2 ^η γραμμή	Συμπεριλαμβάνω την βιβλιοθήκη TinyGPS η οποία είναι απαραίτητη για την αποκωδικοποίηση του στίγματος του GPS σε διαχειρίσιμη πληροφορία
3 ^η γραμμή	Συμπεριλαμβάνω την βιβλιοθήκη SD η οποία είναι απαραίτητη για την διαχείριση της κάρτας μνήμης και την αποθήκευση πληροφοριών σε αυτήν
4 ^η γραμμή	Συμπεριλαμβάνω την βιβλιοθήκη stdlib η οποία είναι συντομογραφία του Standard Libraries και είναι απαραίτητη για την λειτουργία τους προγράμματος, καθώς περιλαμβάνει βασικές διεργασίες, όπως η μετατροπή δεδομένων σε χαρακτήρες και, η διαχείριση διεργασιών

4.2.2 Πρόγραμμα λήψης και εικονικού τριγωνισμού – Δήλωση σταθερών και μεταβλητών

```

7 TinyGPS gps;
8 int CS =53; //For Arduino Mega ONLY!! else it is 10.
9 int LED = 6;
10 int state = 0;

```

7 ^η γραμμή	Η συγκεκριμένη εντολή είναι απαραίτητη για την ενεργοποίηση της βιβλιοθήκης TinyGPS
8 ^η γραμμή	Δημιουργώ και δηλώνω μία μεταβλητή integer CS ίση με 53, καθώς ο συγκεκριμένος ακροδέκτης του arduino Mega είναι ο SS (Slave Select) για την διαχείριση των περιφερειακών. Η ονομασία CS είναι τα αρχικά του Chip Select
9 ^η γραμμή	Δημιουργώ και δηλώνω μία μεταβλητή integer LED ίση με 6, γιατί στον συγκεκριμένο ακροδέκτη του Arduino Mega έχω συνδέσει μία φωτεινή ένδειξη LED για την προβολή πληροφοριών (αναλύονται παρακάτω)
10 ^η γραμμή	Δημιουργώ και δηλώνω μία μεταβλητή integer state ίση με 0 η οποία θα χρησιμοποιηθεί για τον έλεγχο της χειροκίνητης λειτουργίας, δηλαδή την δημιουργία ενός εικονικού μηνύματος διάσωσης που θα έχει ως αποτέλεσμα την αποθήκευση πληροφοριών κατά βούληση

4.2.3 Πρόγραμμα λήψης και εικονικού τριγωνισμού – Δήλωση σταθερών και μεταβλητών

```

11 // If newPass and newPass1 are both 1 (TRUE), then store and calculate new GPS waypoint
12 int newPass = 0; //If new signal - last signal > interval = TRUE (newPass = 1), FALSE (newPass = 0).
13 int newPass1 = 0; //If data from RF link is confirmed, then newPass = 1 (TRUE).
14 int NuMax = 0;
15 const int receive_pin = 5;
16 const char OK[4] = "SOS";
17 int ground_found = 0; // +1 each time we receive a complete correct message

```

11 ^η γραμμή	Με τις 2 πλάγιες γραμμές σημειώνω παρατηρήσεις για ευκολία μετέπειτα παρακολούθησης του προγράμματος είτε από εμένα, είτε από τρίτους
12 ^η γραμμή	Δημιουργώ και δηλώνω μία μεταβλητή integer newPass ίση με 0. Η συγκεκριμένη μεταβλητή είναι απαραίτητη για τον διαχωρισμό 2 μηνυμάτων διάσωσης τα οποία χωρίζονται από ένα χρονικό διάστημα ικανό για να δηλωθεί ως νέο πέρασμα από την εμβέλεια του συστήματος εκπομπής του μηνύματος διάσωσης
13 ^η γραμμή	Δημιουργώ και δηλώνω μία μεταβλητή integer newPass1 ίση με 0. Η συγκεκριμένη μεταβλητή θα χρησιμοποιηθεί για να δηλωθεί το νέο πέρασμα από το σύστημα εκπομπής του μηνύματος διάσωσης. Είναι όμοια με την μεταβλητή newPass αλλά χρησιμοποιείται για διαφορετικό σκοπό. Η χρήση της θα αναλυθεί παρακάτω
14 ^η γραμμή	Δημιουργώ και δηλώνω μία μεταβλητή integer NuMax ίση με 0. Η συγκεκριμένη μεταβλητή θα χρησιμοποιηθεί για να αποθηκεύεται ο μέγιστος αριθμός αποθηκευμένων μηνυμάτων διάσωσης για κάθε πέρασμα του σκάφους διάσωσης έτσι ώστε να γίνονται σωστά οι υπολογισμοί. Η χρήση της θα αναλυθεί παρακάτω
15 ^η γραμμή	Δημιουργώ και δηλώνω μία σταθερά integer receive_pin ίση με 5. Στον ακροδέκτη 5 του Arduino Mega θα συνδεθεί η έξοδος του δέκτη που θα λαμβάνει το μήνυμα

	διάσωσης. Η χρήση σταθεράς και όχι μεταβλητής είναι απαραίτητη διότι δεν αλλάζει στο κύκλωμα κάτι και θα πρέπει να εξασφαλίσω ότι οι πληροφορίες που αναζητά το σύστημα θα λαμβάνονται πάντα από τον επεξεργαστή μου
16 ^η γραμμή	Δημιουργώ και δηλώνω ένα σύνολο χαρακτήρων ως σταθερά OK[4] ίση με “SOS”. Με αυτόν τον τρόπο έχω δημιουργήσει έναν πίνακα που περιέχει του 3 χαρακτήρες του μηνύματος διάσωσης και τον χαρακτήρα τερματισμού, ο οποίος δηλώνεται αυτόματα από το πρόγραμμα. Επιλέγω σταθερά καθώς οι συγκεκριμένοι χαρακτήρες αποτελούν το μήνυμα διάσωσης και είναι αυτό που έχει δηλωθεί για να αποστέλλεται και από το σύστημα εκπομπής
17 ^η γραμμή	Δημιουργώ και δηλώνω μία μεταβλητή integer ground_found ίση με 0. Όπως φαίνεται και από τα σχόλια, η συγκεκριμένη μεταβλητή αυξάνεται κατά 1 κάθε φορά που ένα ολοκληρωμένο μήνυμα διάσωσης λαμβάνεται από το σύστημα λήψης. Είναι χρήσιμη γιατί σε αντίθεση με την μεταβλητή NuMax η τιμή της μηδενίζεται μόνο στο τέλος του κυρίως μέρους του προγράμματος λήψης. Η χρήση της θα δειχθεί παρακάτω

4.2.4 Πρόγραμμα λήψης και εικονικού τριγωνισμού – Δήλωση σταθερών και μεταβλητών

```

18 int msgLED = 7;
19 int msgLEDstate = 0;
20 const float pi = 3.14159;

```

18 ^η γραμμή	Δημιουργώ και δηλώνω μία μεταβλητή integer msgLED ίση με 7, καθώς στον συγκεκριμένο ακροδέκτη του Arduino Mega έχει συνδεθεί φωτεινή ένδειξη για την οπτική επαφή μου όσο τρέχει το πρόγραμμα, για την ένδειξη της λήψης του μηνύματος διάσωσης
19 ^η γραμμή	Δημιουργώ και δηλώνω μία μεταβλητή integer msgLEDstate ίση με 0. Η συγκεκριμένη μεταβλητή είναι απαραίτητη για την διευκόλυνση της εναλλαγής κατάστασης της παραπάνω φωτεινής ένδειξης κατά την διάρκεια λήψης και αδράνειας του δέκτη. Η χρήση της θα δειχθεί παρακάτω αναλυτικότερα όπου και θα γίνει κατανοητό γιατί χρησιμοποιώ ολόκληρη μεταβλητή για μία τόσο απλή διεργασία
20 ^η γραμμή	Δημιουργώ και δηλώνω μία σταθερά float pi ίση με 3.14159. Η δήλωση του αριθμού Π σε μία σταθερά είναι απαραίτητη γιατί δεν έχω συμπεριλάβει κάποια βιβλιοθήκη μαθηματικών και γιατί είναι παγκόσμια σταθερά. Η συγκεκριμένη σταθερά χρησιμοποιείται από το κυρίως πρόγραμμα για τους υπολογισμούς του εικονικού τριγωνισμού. Η χρήση της θα δειχθεί παρακάτω αναλυτικότερα

4.2.5 Πρόγραμμα λήψης και εικονικού τριγωνισμού – Δήλωση σταθερών και μεταβλητών

```

23 //Define String
24 String SD_date_time = 0;
25 float SD_lat = 0;
26 float SD_lon = 0;

```

23 ^η γραμμή	Στο συγκεκριμένο σημείο έκρινα απαραίτητη την προσθήκη σχολίου, γιατί υπάρχουν δεδομένα που δεν αποθηκεύονται ως αριθμοί αποκλειστικά, αλλά ως ένα σύνολο χαρακτήρων και αριθμών με αποτέλεσμα να είναι σημαντική η επισήμανσή των παρακάτω δηλώσεων
24 ^η γραμμή	Δημιουργώ και δηλώνω μία μεταβλητή συνόλου χαρακτήρων SD_date_time ίση με 0, καθώς δεν υπάρχουν συγκεκριμένα δεδομένα για την αρχικοποίησή της. Η συγκεκριμένη μεταβλητή θα χρησιμοποιηθεί από την διεργασία της βιβλιοθήκης TinyGPS για την αποθήκευση της ημερομηνίας και ώρας που θα εξάγεται από κάθε ζητούμενο στίγμα του GPS. Επειδή περιέχει και χαρακτήρες για να είναι σε εύκολα αναγνώσιμη μορφή από τον άνθρωπο, δεν είναι δυνατή η αποθήκευση των συγκεκριμένων δεδομένων σε κάποια άλλη μορφή μεταβλητής, όπως οι παρακάτω μεταβλητές float. Η χρήση της θα δειχθεί και θα αναλυθεί παρακάτω
25 ^η γραμμή	Δημιουργώ και δηλώνω μία μεταβλητή float SD_lat ίση με 0, καθώς δεν υπάρχουν συγκεκριμένα δεδομένα για την αρχικοποίησή της. Η συγκεκριμένη μεταβλητή θα χρησιμοποιηθεί από την διεργασία της βιβλιοθήκης TinyGPS για την αποθήκευση του γεωγραφικού πλάτους που θα εξάγεται από κάθε ζητούμενο στίγμα του GPS. Η χρήση μεταβλητής float είναι απαραίτητη, καθώς το γεωγραφικό πλάτος αναπαρίσταται από έναν αριθμό πολλών ψηφίων και με χρήση υποδιαστολής. Η χρήση της θα δειχθεί και θα αναλυθεί παρακάτω
26 ^η γραμμή	Δημιουργώ και δηλώνω μία μεταβλητή float SD_lon ίση με 0, καθώς δεν υπάρχουν συγκεκριμένα δεδομένα για την αρχικοποίησή της. Η συγκεκριμένη μεταβλητή θα χρησιμοποιηθεί από την διεργασία της βιβλιοθήκης TinyGPS για την αποθήκευση του γεωγραφικού μήκους που θα εξάγεται από κάθε ζητούμενο στίγμα του GPS. Η χρήση μεταβλητής float είναι απαραίτητη, καθώς το γεωγραφικό μήκος αναπαρίσταται από έναν αριθμό πολλών ψηφίων και με χρήση υποδιαστολής. Η χρήση της θα δειχθεί και θα αναλυθεί παρακάτω

4.2.6 Πρόγραμμα λήψης και εικονικού τριγωνισμού – Δήλωση σταθερών και μεταβλητών

```

28 //Define GPS coordinates' polygon vertices as floats
29 //Assume a polygon with maximum 7 vertices
30 float Lat[8]={0};
31 float Long[8]={0};
32

```

28 ^η &29 ^η γραμμή	Στο συγκεκριμένο πρόγραμμα έχω λάβει υπόψη μου ότι ο μέγιστος αριθμός μηνυμάτων διάσωσης που ενδέχεται να λάβει το σύστημα λήψης, είναι 7. Αυτός ο αριθμός προέκυψε από δοκιμές και τον επέλεξα ως ικανοποιητικό για τον υπολογισμό του εικονικού πολυγωνισμού
30 ^η γραμμή	Δημιουργώ και δηλώνω ένα σύνολο μεταβλητών float Lat[8] και όλες οι θέσεις είναι ίσες με 0. Το όνομα Lat αποτελεί συντομογραφία του Latitude. Στον συγκεκριμένο πίνακα θα αποθηκεύονται οι τιμές της μεταβλητής SD_lat για τον υπολογισμό του πιθανού πραγματικού γεωγραφικού πλάτους του συστήματος εκπομπής του μηνύματος διάσωσης. Το μέγεθος είναι ίσο με 8, καθώς για τον σωστό υπολογισμό ως 8 ^ο γεωγραφικό πλάτος χρησιμοποιείται το γεωγραφικό πλάτος από το 1 ^ο στίγμα. Η χρήση του θα δειχθεί και θα αναλυθεί παρακάτω
31 ^η γραμμή	Δημιουργώ και δηλώνω ένα σύνολο μεταβλητών float Long[8] και όλες οι θέσεις είναι ίσες με 0. Το όνομα Long αποτελεί συντομογραφία του Longitude. Στον συγκεκριμένο πίνακα θα αποθηκεύονται οι τιμές της μεταβλητής SD_lon για τον υπολογισμό του πιθανού πραγματικού γεωγραφικού μήκους του συστήματος εκπομπής του μηνύματος διάσωσης. Το μέγεθος είναι ίσο με 8, καθώς για τον σωστό υπολογισμό ως 8 ^ο γεωγραφικό μήκος χρησιμοποιείται το γεωγραφικό μήκος από το 1 ^ο στίγμα. Η χρήση του θα δειχθεί και θα αναλυθεί παρακάτω

4.2.7 Πρόγραμμα λήψης και εικονικού τριγωνισμού – Δήλωση σταθερών και μεταβλητών

```

33 int Nu = 0 ; //Integer used instead of i,
34 //for pointing Lat[i] and Long[i] arrays
35 // (here is LatNu and LongNu)
36

```

33 ^η γραμμή	Δημιουργώ και δηλώνω μία μεταβλητή integer Nu με τιμές αρχικοποίησης ίση με 0. Η συγκεκριμένη μεταβλητή θα χρησιμοποιηθεί αντί των γνωστών i και j ως ενδείκτης
------------------------	--

(pointer) από τις λειτουργίες που αφορούν τις θέσεις μέσα στους πίνακες **Lat[8]** και **Long[8]**. Η ονομασία προέκυψε από την συντομογραφία της αγγλικής λέξης Number και την επέλεξα διότι πληροί τα εξής χαρακτηριστικά :

- Είναι διαφορετική από τις συνηθισμένες i και j που χρησιμοποιούνται ευρέως και ελλοχεύει κίνδυνος χρήσης αυτών και περαιτέρω σύγχυσης εντός του προγράμματος
- Παραμένει απλή και κατανοητή για την χρήση που προορίζεται
- Προέρχεται από συγκεκριμένη λέξη, οπότε καθίσταται εύκολη η κατανόησή της
- Μπορεί εύκολα να συνδυαστεί για να δημιουργήσει νέες ονοματολογίες για χρήση σε μεταβλητές που περιέχουν δεδομένα που σχετίζονται με αυτήν, όπως η **NuMax**

Η χρήση μίας κοινής μεταβλητής δίνει την αποφυγή λάθους στην καταγραφή και ανάγνωση των άκρως σημαντικών δεδομένων του στίγματος του GPS.

Η χρήση της θα δειχθεί και θα αναλυθεί παρακάτω

4.2.8 Πρόγραμμα λήψης και εικονικού τριγωνισμού – Δήλωση μεταβλητών χρονικών προσδιορισμών

```

37 unsigned long lastSignal = 0; //Time since last valid signal
38 unsigned long interval = 30000; //Time difference greater than which
39                                     //last signal and new signal are treated as new pass
40 unsigned long newSignal = 0;
41

```

Όπως ανέφερα παραπάνω στην δήλωση των μεταβλητών **newPass** και **newPass1**, πρέπει να υπάρχει διαχωρισμός των μηνυμάτων διάσωσης που λαμβάνονται από το σύστημα λήψης με βάση χρονικά κριτήρια, έτσι ώστε κατά τα περάσματα του σκάφους που μεταφέρει το σύστημα λήψης να μην εμπλέκονται τα δεδομένα από τα στίγματα του GPS με παλαιότερα που όχι μόνο δεν είναι χρήσιμα, αλλά μπορεί να γίνουν και επικίνδυνα για την σωστό υπολογισμό.

Ο διαχωρισμός επιτυγχάνεται με την αποθήκευση του χρόνου λήψης κάθε μηνύματος και τον διαχωρισμό σε παρελθοντικό και παρόν σήμα και σύγκριση της χρονικής διαφοράς μεταξύ των δύο χρόνων. Αν η διαφορά προκύψει μεγαλύτερη από ένα περιθώριο, το οποίο προκύπτει μετά από δοκιμές αναλόγως του σκάφους μεταφοράς του συστήματος λήψης, των δυνατοτήτων του συστήματος εκπομπής αλλά και εξωγενών παραγόντων, γίνονται οι κατάλληλες ενέργειες από το πρόγραμμα για τον ορθό υπολογισμό και την επίτευξη του σκοπού δημιουργίας του όλου συστήματος.

Η χρήση μη προσημασμένων μεταβλητών είναι απαραίτητη, γιατί ο χρόνος δεν λαμβάνει αρνητικές τιμές και η μη χρήση προσήμου μας δίνει την δυνατότητα η μεταβλητή να πάρει πολύ μεγαλύτερες τιμές με αποτέλεσμα

την ασφάλεια των μετρήσεων, καθώς τα όρια της μη προσημασμένης μεταβλητής είναι πολύ μεγαλύτερα από τα όρια του συγκεκριμένου πειράματος.

37 ^η γραμμή	Δημιουργώ και δηλώνω μία μη προσημασμένη μεταβλητή long lastSignal με αρχικοποιημένη τιμή ίση με 0. Η συγκεκριμένη μεταβλητή θα χρησιμοποιηθεί για την αποθήκευση του χρόνου λήψης του προτελευταίου μηνύματος διάσωσης. Ο μετρούμενος χρόνος αναφέρεται σε χιλιοστά του δευτερολέπτου (milliseconds). Η χρήση της θα δειχθεί και θα αναλυθεί παρακάτω
38 ^η γραμμή	Δημιουργώ και δηλώνω μία μη προσημασμένη μεταβλητή long interval με αρχικοποιημένη τιμή ίση με 30000. Η συγκεκριμένη μεταβλητή αναφέρεται σε χρόνο μετρούμενο σε χιλιοστά του δευτερολέπτου (milliseconds) και η τιμή της προκύπτει από πειράματα και δοκιμές σύμφωνα με το σύστημα στο οποίο εκτελούνται αυτά. Η τιμή 30000 είναι αποδεκτή για τα χαρακτηριστικά των ηλεκτρονικών των συστημάτων της συγκεκριμένης εργασίας και για σκάφος που κινείται με ταχύτητα περίπου 1.4 μέτρων το δευτερόλεπτο (5.0 km/h). Η χρήση της θα δειχθεί και θα αναλυθεί παρακάτω
40 ^η γραμμή	Δημιουργώ και δηλώνω μία μη προσημασμένη μεταβλητή long newSignal με αρχικοποιημένη τιμή ίση με 0. Η συγκεκριμένη μεταβλητή θα χρησιμοποιηθεί για την αποθήκευση του χρόνου λήψης του πιο πρόσφατου σήματος, δηλαδή του τελευταίου μηνύματος διάσωσης. Ο μετρούμενος χρόνος αναφέρεται σε χιλιοστά του δευτερολέπτου (milliseconds). Η χρήση της θα δειχθεί και θα αναλυθεί παρακάτω

4.2.9 Πρόγραμμα λήψης και εικονικού τριγωνισμού – Δήλωση μεταβλητών διεργασίας υπολογισμού κέντρου βάρους γεωγραφικών συντεταγμένων

```

42 //floats used for calculating centroid. For more info,
43 //refer to (http://www.geomidpoint.com/calculation.html)
44
45 float LatRad[8] = {0}, LongRad[8] = {0};
46 float CartX[8] = {0}, CartY[8] = {0}, CartZ[8] = {0};
47 float Xsum = 0, Ysum = 0, Zsum = 0, CentX = 0, CentY = 0, CentZ = 0;
48 float CentLongRad = 0, Hyp = 0, CentLatRad = 0;
49 float CentLat = 0, CentLong = 0;

```

45 ^η γραμμή	Δημιουργώ και δηλώνω ένα σύνολο μεταβλητών float LatRad[8] και όλες οι θέσεις είναι ίσες με 0. Ο συγκεκριμένος πίνακας θα χρησιμοποιηθεί για την μετατροπή του γεωγραφικού πλάτους σε ακτίνια, έτσι ώστε να υπολογίσω το κέντρο βάρους των
------------------------	---

	<p>γεωγραφικών πλατών που αποθήκευσε το σύστημα λήψης στα σημεία που έλαβε το μήνυμα διάσωσης. Η χρήση του θα δειχθεί και θα αναλυθεί παρακάτω</p> <p>Δημιουργώ και δηλώνω ένα σύνολο μεταβλητών float LongRad[8] και όλες οι θέσεις είναι ίσες με 0. Ο συγκεκριμένος πίνακας θα χρησιμοποιηθεί για την μετατροπή του γεωγραφικού μήκους σε ακτίνια, έτσι ώστε να υπολογίσω το κέντρο βάρους των γεωγραφικών μηκών που αποθήκευσε το σύστημα λήψης στα σημεία που έλαβε το μήνυμα διάσωσης. Η χρήση του θα δειχθεί και θα αναλυθεί παρακάτω</p>
46 ^η γραμμή	<p>Δημιουργώ και δηλώνω ένα σύνολο μεταβλητών float CartX[8] και όλες οι θέσεις είναι ίσες με 0. Ο συγκεκριμένος πίνακας θα χρησιμοποιηθεί για τον υπολογισμό του κέντρου βάρους των γεωγραφικών συντεταγμένων, αφού έχει προηγηθεί μετατροπή τους σε καρτεσιανό σύστημα.</p> <p>Δημιουργώ και δηλώνω ένα σύνολο μεταβλητών float CartY[8] και όλες οι θέσεις είναι ίσες με 0. Ο συγκεκριμένος πίνακας θα χρησιμοποιηθεί για τον υπολογισμό του κέντρου βάρους των γεωγραφικών συντεταγμένων, αφού έχει προηγηθεί μετατροπή τους σε καρτεσιανό σύστημα.</p> <p>Δημιουργώ και δηλώνω ένα σύνολο μεταβλητών float CartZ[8] και όλες οι θέσεις είναι ίσες με 0. Ο συγκεκριμένος πίνακας θα χρησιμοποιηθεί για τον υπολογισμό του κέντρου βάρους των γεωγραφικών συντεταγμένων, αφού έχει προηγηθεί μετατροπή τους σε καρτεσιανό σύστημα.</p>
47 ^η γραμμή	<p>Δημιουργώ και δηλώνω μία μεταβλητή float Xsum ίση με 0, καθώς δεν υπάρχουν συγκεκριμένα δεδομένα για την αρχικοποίησή της. Η συγκεκριμένη μεταβλητή θα χρησιμοποιηθεί από την διεργασία υπολογισμού του κέντρου βάρους των γεωγραφικών συντεταγμένων, αφού έχει προηγηθεί μετατροπή τους σε καρτεσιανό σύστημα.</p> <p>Δημιουργώ και δηλώνω μία μεταβλητή float Ysum ίση με 0, καθώς δεν υπάρχουν συγκεκριμένα δεδομένα για την αρχικοποίησή της. Η συγκεκριμένη μεταβλητή θα χρησιμοποιηθεί από την διεργασία υπολογισμού του κέντρου βάρους των γεωγραφικών συντεταγμένων, αφού έχει προηγηθεί μετατροπή τους σε καρτεσιανό σύστημα.</p> <p>Δημιουργώ και δηλώνω μία μεταβλητή float Zsum ίση με 0, καθώς δεν υπάρχουν συγκεκριμένα δεδομένα για την αρχικοποίησή της. Η συγκεκριμένη μεταβλητή θα χρησιμοποιηθεί από την διεργασία υπολογισμού του κέντρου βάρους των γεωγραφικών συντεταγμένων, αφού έχει προηγηθεί μετατροπή τους σε καρτεσιανό σύστημα.</p> <p>Δημιουργώ και δηλώνω μία μεταβλητή float CentX ίση με 0, καθώς δεν υπάρχουν συγκεκριμένα δεδομένα για την αρχικοποίησή της. Η συγκεκριμένη μεταβλητή θα χρησιμοποιηθεί από την διεργασία υπολογισμού του κέντρου βάρους των γεωγραφικών</p>

	<p>συντεταγμένων για την αποθήκευση του κέντρου βάρους στον καρτεσιανό άξονα των X.</p> <p>Δημιουργώ και δηλώνω μία μεταβλητή float CentY ίση με 0, καθώς δεν υπάρχουν συγκεκριμένα δεδομένα για την αρχικοποίησή της. Η συγκεκριμένη μεταβλητή θα χρησιμοποιηθεί από την διεργασία υπολογισμού του κέντρου βάρους των γεωγραφικών συντεταγμένων για την αποθήκευση του κέντρου βάρους στον καρτεσιανό άξονα των Y.</p> <p>Δημιουργώ και δηλώνω μία μεταβλητή float CentZ ίση με 0, καθώς δεν υπάρχουν συγκεκριμένα δεδομένα για την αρχικοποίησή της. Η συγκεκριμένη μεταβλητή θα χρησιμοποιηθεί από την διεργασία υπολογισμού του κέντρου βάρους των γεωγραφικών συντεταγμένων για την αποθήκευση του κέντρου βάρους στον καρτεσιανό άξονα των Z.</p>
48 ^η γραμμή	<p>Δημιουργώ και δηλώνω μία μεταβλητή float CentLongRad ίση με 0, καθώς δεν υπάρχουν συγκεκριμένα δεδομένα για την αρχικοποίησή της. Η συγκεκριμένη μεταβλητή θα χρησιμοποιηθεί από την διεργασία υπολογισμού του κέντρου βάρους των γεωγραφικών συντεταγμένων για τον υπολογισμό του κέντρου βάρους του γεωγραφικού μήκους.</p> <p>Δημιουργώ και δηλώνω μία μεταβλητή float Hyp ίση με 0, καθώς δεν υπάρχουν συγκεκριμένα δεδομένα για την αρχικοποίησή της. Η συγκεκριμένη μεταβλητή θα χρησιμοποιηθεί από την διεργασία υπολογισμού του κέντρου βάρους των γεωγραφικών συντεταγμένων για τον υπολογισμό του κέντρου βάρους.</p> <p>Δημιουργώ και δηλώνω μία μεταβλητή float CentLatRad ίση με 0, καθώς δεν υπάρχουν συγκεκριμένα δεδομένα για την αρχικοποίησή της. Η συγκεκριμένη μεταβλητή θα χρησιμοποιηθεί από την διεργασία υπολογισμού του κέντρου βάρους των γεωγραφικών συντεταγμένων για τον υπολογισμό του κέντρου βάρους του γεωγραφικού πλάτους.</p>
49 ^η γραμμή	<p>Δημιουργώ και δηλώνω μία μεταβλητή float CentLat ίση με 0, καθώς δεν υπάρχουν συγκεκριμένα δεδομένα για την αρχικοποίησή της. Η συγκεκριμένη μεταβλητή θα χρησιμοποιηθεί για την αποθήκευση του γεωγραφικού πλάτους των γεωγραφικών συντεταγμένων του κέντρου βάρους εικονικού πολυγωνισμού που γίνεται από το σύστημα λήψης, αφού έχουν μετατραπεί από καρτεσιανές συντεταγμένες σε μοίρες αναγνώσιμες από το παγκόσμιο σύστημα γεωγραφικών συντεταγμένων. Η χρήση της θα δειχθεί και θα αναλυθεί παρακάτω</p> <p>Δημιουργώ και δηλώνω μία μεταβλητή float CentLong ίση με 0, καθώς δεν υπάρχουν συγκεκριμένα δεδομένα για την αρχικοποίησή της. Η συγκεκριμένη μεταβλητή θα χρησιμοποιηθεί για την αποθήκευση του γεωγραφικού μήκους των γεωγραφικών συντεταγμένων του κέντρου βάρους εικονικού πολυγωνισμού που γίνεται από το</p>

	<p>σύστημα λήψης, αφού έχουν μετατραπεί από καρτεσιανές συντεταγμένες σε μοίρες αναγνώσιμες από το παγκόσμιο σύστημα γεωγραφικών συντεταγμένων. Η χρήση της θα δειχθεί και θα αναλυθεί παρακάτω</p>
--	---

4.2.10 Πρόγραμμα λήψης και εικονικού τριγωνισμού – Δήλωση μεταβλητής δοκιμών

```

51 //Low state indicates than no values shall be stored to the Lat,Long tables.
52 //Changing state to HIGH (when button is pressed or signal received) stores the next value
53 volatile int buttonstate = 0;
54

```

53 ^η γραμμή	<p>Δημιουργώ και δηλώνω μία μεταβλητή integer buttonstate ίση με 0. Η συγκεκριμένη μεταβλητή θα χρησιμοποιηθεί ως κατά βούληση εικονικό μήνυμα διάσωσης πατώντας ένα button που θα βρίσκεται τοποθετημένο στον Arduino Mega που θα λειτουργεί ως σύστημα λήψης πάνω στο σκάφος. Η μεταβλητή buttonstate δηλώνεται στο πρόγραμμα ως <i>volatile</i> γιατί είναι μία μεταβλητή που η κατάστασή της δεν ελέγχεται από το ίδιο το πρόγραμμα, αλλά από εξωτερικούς παράγοντες, δηλαδή στην συγκεκριμένη περίπτωση από button.</p>
------------------------	--

4.2.11 Πρόγραμμα λήψης και εικονικού τριγωνισμού – Δήλωση λειτουργιών κύριου προγράμματος

```

55 static void smartdelay(unsigned long ms);
56 static void print_float(float val, float invalid, int len, int prec, int SD_val, int Nu);
57 static void print_date(TinyGPS &gps);
58 static void print_str(const char *str, int len);
59 void sdLoggingTEMP();
60 void midPointLog();
61

```

Στο συγκεκριμένο σημείο θα πρέπει να εξηγήσω ότι μία λειτουργία (function) είναι μία ομάδα δηλώσεων που μαζί εκτελούν μία εργασία. Κάθε πρόγραμμα στην γλώσσα C++ έχει τουλάχιστον μία λειτουργία (ή συνάρτηση), η οποία είναι η κύρια [**void main()**].

Συνήθως, οι επιπλέον λειτουργίες/συναρτήσεις προστίθενται έτσι ώστε το κυρίως πρόγραμμα, δηλαδή η **void main()** να παραμένει σχετικά ευανάγνωστη, αλλά και επειδή πολλές από τις λειτουργίες μπορεί να επαναλαμβάνονται μέσα στην ροή του προγράμματος.

Οι λειτουργίες/συναρτήσεις δηλώνονται στην αρχή του προγράμματος, συνήθως μετά από την δήλωση των μεταβλητών και των σταθερών που θα χρησιμοποιηθούν στο πρόγραμμα, υλοποιούνται στο τέλος, μετά το κυρίως πρόγραμμα για λόγου δομημένης γραφής, και καλούνται εντός του προγράμματος απλώς με την γραφή

του ονόματός τους για να εκτελέσουν όλες τις εντολές που περιλαμβάνουν, σύμφωνα με τα δεδομένα που χρειάζεται να τους δοθούν.

Μία λειτουργία/συνάρτηση **void** δηλώνει ότι δεν θα επιστρέψει κανένα δεδομένο κατά την κλήση της, πέρα από τις αλλαγές που θα πραγματοποιήσει σύμφωνα με τις εντολές που περιλαμβάνονται σε αυτήν.

Η διαφορά ανάμεσα σε μία λειτουργία/συνάρτηση **static void** και μία **void** είναι κατά την στατική δήλωση μίας λειτουργίας δεσμεύεται από τον μεταφραστή η ονομασία της μόνο για εκείνο το πρόγραμμα στο οποίο έχει δηλωθεί, ενώ η μη στατική δεσμεύεται για όλα υποπρογράμματα. Αυτό έχει ως αποτέλεσμα αν βρεθούν δύο λειτουργίες που έχουν το ίδιο όνομα σε διαφορετικά προγράμματα και που δεν είναι δηλωμένες ως στατικές, να εμφανιστεί μήνυμα λάθους και ίσως να σταματήσει να εκτελείται το πρόγραμμα.

Στην συγκεκριμένη εργασία το πρόγραμμα που εκτελείται είναι μόνο ένα, οπότε η δήλωση των λειτουργιών είτε ως στατικές είτε ως μη στατικές δεν επιφέρει καμία αλλαγή.

55 ^η γραμμή	Δημιουργώ και δηλώνω μία συνάρτηση τύπου static void και όνομα smartdelay με παράμετρο τύπου unsigned long και όνομα ms . Η συγκεκριμένη συνάρτηση θα παίρνει μόνο την τιμή της παραμέτρου ms και δεν θα επιστρέφει κάποιο αποτέλεσμα. Θα χρησιμοποιηθεί για να προσθέσει καθυστέρηση στο πρόγραμμα με συγκεκριμένο τρόπο. Η χρήση της θα δειχθεί και θα αναλυθεί παρακάτω
56 ^η γραμμή	Δημιουργώ και δηλώνω μία συνάρτηση τύπου static void και όνομα print_float με τις εξής παραμέτρους τύπου float και όνομα val , τύπου float και όνομα invalid , τύπου int και όνομα len , τύπου int και όνομα prec , τύπου int και όνομα SD_val , τύπου int και όνομα Nu . Η χρήση της θα δειχθεί και θα αναλυθεί παρακάτω
57 ^η γραμμή	Δημιουργώ και δηλώνω μία συνάρτηση τύπου static void και όνομα print_date με παράμετρο τύπου TinyGPS (είναι “τύπος” από συγκεκριμένη βιβλιοθήκη που χρησιμοποιείται στο πρόγραμμα) και όνομα &gps (δηλαδή θα έχει όνομα εκεί που θα δείχνει η συγκεκριμένη μεταβλητή και όχι το όνομα gps αυτό καθαυτό). Η χρήση της θα δειχθεί και θα αναλυθεί παρακάτω
58 ^η γραμμή	Δημιουργώ και δηλώνω μία συνάρτηση τύπου static void και όνομα print_str με παραμέτρους τύπου const char και όνομα str (ο χαρακτήρας αστεράκι * δηλώνει ονομασία δυναμικού μεγέθους, δηλαδή η παράμετρος str δεν θα είναι πίνακας χαρακτήρων συγκεκριμένου μεγέθους), και τύπου int και όνομα len . Η χρήση της θα δειχθεί και θα αναλυθεί παρακάτω

59 ^η γραμμή	<p>Δημιουργώ και δηλώνω μία συνάρτηση τύπου void και όνομα sdLoggingTEMP. Δεν θα έχει παραμέτρους, καθώς θα εκτελεί εντολές που δεν χρειάζονται να αλλάξουν τιμές σε μεταβλητές του προγράμματος.</p> <p>Η συγκεκριμένη λειτουργία θα χρησιμοποιηθεί για την αποθήκευση δεδομένων από τα γεωγραφικά σημεία που βρέθηκε το μήνυμα διάσωσης. Ο λόγος ύπαρξής της είναι η αποσφαλμάτωση του προγράμματος κατά τον υπολογισμό του γεωγραφικού κέντρου βάρους.</p> <p>Η χρήση της θα δειχθεί και θα αναλυθεί παρακάτω</p>
60 ^η γραμμή	<p>Δημιουργώ και δηλώνω μία συνάρτηση τύπου void και όνομα midPointLog. Δεν θα έχει παραμέτρους, καθώς θα εκτελεί εντολές που δεν χρειάζονται να αλλάξουν τιμές σε μεταβλητές του προγράμματος. Η συγκεκριμένη λειτουργία θα χρησιμοποιηθεί για την αποθήκευση του γεωγραφικού κέντρου βάρους που θα προκύψει από τον εικονικό πολυγωνισμό μετά από την συλλογή δεδομένων από τα γεωγραφικά σημεία που βρέθηκε το μήνυμα διάσωσης.</p> <p>Η χρήση της θα δειχθεί και θα αναλυθεί παρακάτω</p>

4.2.12 Πρόγραμμα λήψης και εικονικού τριγωνισμού – Βασική λειτουργία [void setup()]

```

62 void setup()
63 {
64   pinMode (10, OUTPUT);
65   pinMode (CS, OUTPUT); //ChipSelect pin for the SD card
66   pinMode (LED, OUTPUT); //LED indicator (changes state when vertices meet desired number)
67   pinMode (msgLED, OUTPUT);
68   digitalWrite(LED, LOW);
69   vw_set_rx_pin(receive_pin);
70   vw_setup(2000); // Bits per sec
71
72   Serial.begin(115200);
73   Serial1.begin(38400);
74   SD.begin(CS);
75   Serial.println("Initializing board in 3...2...1...");
76   Serial.println("Initialization complete!");
77 }

```

64 ^η γραμμή	Ο ακροδέκτης 10 του Arduino Mega δηλώνεται στον επεξεργαστή ότι θα είναι έξοδος
65 ^η γραμμή	Ο ακροδέκτης που αντιστοιχεί στην τιμή της μεταβλητής CS που δηλώθηκε στην αρχή του προγράμματος, εδώ δηλώνεται στον επεξεργαστή ότι θα είναι έξοδος. Επειδή ο συγκεκριμένος ακροδέκτης θα χρησιμοποιηθεί από την βιβλιοθήκη SD.h για την επικοινωνία με το περιφερειακό που είναι υπεύθυνο για την διαχείριση της κάρτας μνήμης, απαιτείται να δηλωθεί ως έξοδος, αλλιώς θα αποτύχει η αποθήκευση των δεδομένων στην εξωτερική κάρτα μνήμης.

66 ^η γραμμή	Ο ακροδέκτης που αντιστοιχεί στην τιμή της μεταβλητής LED που δηλώθηκε στην αρχή του προγράμματος, εδώ δηλώνεται στον επεξεργαστή ότι θα είναι έξοδος
67 ^η γραμμή	Ο ακροδέκτης που αντιστοιχεί στην τιμή της μεταβλητής msgLED που δηλώθηκε στην αρχή του προγράμματος, εδώ δηλώνεται στον επεξεργαστή ότι θα είναι έξοδος
68 ^η γραμμή	Με την εντολή digitalWrite δηλώνω στο πρόγραμμα ότι ο ακροδέκτης που θα ακολουθήσει θα λάβει μία από τις 2 δυνατές ψηφιακές τιμές. Οι τρόποι έκφρασης του ζεύγους των ψηφιακών τιμών είναι πολλοί, όπως 0–1, FALSE – TRUE, LOW – HIGH. Στην συγκεκριμένη περίπτωση, ο ακροδέκτης που αντιστοιχεί στην τιμή της μεταβλητής LED θα πάρει την ψηφιακή τιμή LOW, το οποίο θα έχει ως αποτέλεσμα να σβήσει.
69 ^η γραμμή	Δηλώνεται στην βιβλιοθήκη VirtualWire ότι ο ακροδέκτης που θα χρησιμοποιηθεί για την λήψη των δεδομένων μέσω του εναέριου καναλιού, θα είναι η τιμή της σταθεράς receive_pin που δηλώθηκε στην αρχή του προγράμματος. Σε αυτόν τον ακροδέκτη θα συνδεθεί η είσοδος του δέκτη.
70 ^η γραμμή	Δηλώνεται στην βιβλιοθήκη VirtualWire ότι ο ρυθμός μετάδοσης θα είναι 2000bps. Αυτό δηλώνεται και στο πρόγραμμα του συστήματος εκπομπής. Αν γίνει λάθος, δεν θα μπορεί να αναγνωσθεί το μήνυμα διάσωσης.
72 ^η γραμμή	Με την συγκεκριμένη εντολή δηλώνεται στον Arduino Mega ότι το ζεύγος των ακροδεκτών που αντιστοιχούν στην σειριακή μετάδοση νούμερο 0, θα ξεκινήσουν την επικοινωνία με ρυθμό μετάδοσης 115200 bits per second. Η συγκεκριμένη σειριακή θύρα χρησιμοποιείται για την επικοινωνία με τον ηλεκτρονικό υπολογιστή για την αποστολή κάποιων εντολών που χρησιμοποιούνται για την αποσφαλμάτωση και την παρακολούθηση της ομαλής λειτουργίας του προγράμματος.
73 ^η γραμμή	Με την συγκεκριμένη εντολή δηλώνεται στον Arduino Mega ότι το ζεύγος των ακροδεκτών που αντιστοιχούν στην σειριακή μετάδοση νούμερο 1, θα ξεκινήσουν την επικοινωνία με ρυθμό μετάδοσης 38400 bits per second. Η συγκεκριμένη σειριακή θύρα χρησιμοποιείται για την λήψη των μη επεξεργασμένων δεδομένων από το GPS. Από αυτή την θύρα λαμβάνονται όλα τα απαραίτητα δεδομένα που χρησιμοποιούνται από την βιβλιοθήκη TinyGPS, που σε συνδυασμό με τα δεδομένα που λαμβάνονται από τον δέκτη του συστήματος, μας δίνουν το επιθυμητό αποτέλεσμα. Η ταχύτητα που επελέγη είναι η ταχύτητα με την οποία επικοινωνεί το GPS.
74 ^η γραμμή	Με την συγκεκριμένη εντολή γίνεται αρχικοποίηση της βιβλιοθήκης SD.h και της εξωτερικής κάρτας μνήμης. Επειδή το περιφερειακό που είναι υπεύθυνο για την διαχείριση της κάρτας μνήμης συνδέεται με συγκεκριμένο τρόπο και σε συγκεκριμένους ακροδέκτες του Arduino Mega, σε αντίθεση με τις εντολές Serial.begin και Serial1.begin δεν χρειάζεται δήλωση της ταχύτητας με την οποία θα επικοινωνήσουν,

	αλλά μόνο η δήλωση του ακροδέκτη που θα ενεργοποιεί το περιφερειακό της κάρτας μνήμης, όταν χρειάζεται.
75 ^η γραμμή	Με την συγκεκριμένη εντολή δηλώνω στο πρόγραμμα να στείλει το σύνολο των χαρακτήρων <i>Initializing board in 3...2...1...</i> μέσω της σειριακής θύρας 0. Οι συγκεκριμένοι χαρακτήρες στέλνονται στον ηλεκτρονικό υπολογιστή με σκοπό την παρακολούθηση της ομαλής εκκίνησης του προγράμματος.
76 ^η γραμμή	Με την συγκεκριμένη εντολή δηλώνω στο πρόγραμμα να στείλει το σύνολο των χαρακτήρων <i>Initialization complete!</i> μέσω της σειριακής θύρας 0. Οι συγκεκριμένοι χαρακτήρες στέλνονται στον ηλεκτρονικό υπολογιστή με σκοπό την παρακολούθηση της ομαλής εκκίνησης του προγράμματος.

4.2.13 Πρόγραμμα λήψης και εικονικού τριγωνισμού – Λειτουργία επανάληψης [void loop()]

```

79 void loop()
80 {
81   float flat, flon, val;
82   unsigned long age, date, time, chars = 0;
83   unsigned short sentences = 0, failed = 0;
84

```

Η λειτουργία/συνάρτηση loop στο πρόγραμμα χρησιμοποιείται γιατί τρέχει συνεχόμενα, εφόσον δεν διακόπτεται από κάποια εντολή. Δηλαδή, σε αντίθεση με την void setup(), η void loop() μετά την τελευταία της εντολή επιστρέφει στην πρώτη της εντολή.

Στην συγκεκριμένη περίπτωση, έχω σπάσει την void loop() σε κομμάτια για να γίνει πιο εύκολη η επεξήγηση και η κατανόησή της, ανάλογα με τα επιμέρους σύνολα των εντολών που την απαρτίζουν.

Πολλές μεταβλητές δεν χρειάζεται να δημιουργούνται και να δηλώνονται στην αρχή του προγράμματος, έξω και πριν από οποιαδήποτε διεργασία. Μπορούν να δημιουργούνται και να δηλώνονται εντός κάποιας διεργασίας ή ακόμα και στην ονομασία της ίδιας της διεργασίας. Με αυτόν τον τρόπο, στην συγκεκριμένη περίπτωση, κερδίζουμε χώρο στην μνήμη τυχαίας προσπέλασης (RAM).

79 ^η γραμμή	Με την εισαγωγή της εντολής void loop(), δίνω στο πρόγραμμα να καταλάβει ότι θα ξεκινήσει το σύνολο των εντολών που αποτελούν την διεργασία επανάληψης. Με το σύμβολο “{” σημειώνεται η αρχή της διεργασίας, ενώ με το σύμβολο “}” σημειώνεται το τέλος της. Στο ενδιάμεσο εμφανίζονται και άλλα όμοια σύμβολα, αλλά αφορούν τις κατά τόπους ενάρξεις και λήξεις των εκάστοτε διεργασιών.
------------------------	---

81 ^η γραμμή	<p>Δημιουργώ και δηλώνω μία μεταβλητή float flat χωρίς αρχική τιμή. Η συγκεκριμένη μεταβλητή θα χρησιμοποιηθεί από την βιβλιοθήκη TinyGPS για την προσωρινή αποθήκευση του γεωγραφικού πλάτους κατά την εξαγωγή του από τα δεδομένα του GPS.</p> <p>Δημιουργώ και δηλώνω μία μεταβλητή float flon χωρίς αρχική τιμή. Η συγκεκριμένη μεταβλητή θα χρησιμοποιηθεί από την βιβλιοθήκη TinyGPS για την προσωρινή αποθήκευση του γεωγραφικού μήκους κατά την εξαγωγή του από τα δεδομένα του GPS.</p> <p>Δημιουργώ και δηλώνω μία μεταβλητή float val χωρίς αρχική τιμή. Η συγκεκριμένη μεταβλητή θα χρησιμοποιηθεί από την βιβλιοθήκη TinyGPS για την προσωρινή αποθήκευση δεδομένων από το GPS κατά την διαδικασία εξαγωγής των απαραίτητων δεδομένων για εμάς, δηλαδή του γεωγραφικού μήκους, του γεωγραφικού πλάτους και της ημερομηνίας και ώρες.</p>
82 ^η γραμμή	<p>Δημιουργώ και δηλώνω μία μη προσημασμένη μεταβλητή long age με μη αρχικοποιημένη τιμή. Η συγκεκριμένη μεταβλητή θα χρησιμοποιηθεί από την βιβλιοθήκη TinyGPS για την προσωρινή αποθήκευση της ημερομηνίας και της ώρας που θα εξάγονται από τα δεδομένα του GPS.</p> <p>Δημιουργώ και δηλώνω μία μη προσημασμένη μεταβλητή long date με μη αρχικοποιημένη τιμή. Η συγκεκριμένη μεταβλητή θα χρησιμοποιηθεί από την βιβλιοθήκη TinyGPS για την προσωρινή αποθήκευση της ημερομηνίας που θα εξάγεται από τα δεδομένα του GPS.</p> <p>Δημιουργώ και δηλώνω μία μη προσημασμένη μεταβλητή long time με μη αρχικοποιημένη τιμή. Η συγκεκριμένη μεταβλητή θα χρησιμοποιηθεί από την βιβλιοθήκη TinyGPS για την προσωρινή αποθήκευση της ώρας που θα εξάγεται από τα δεδομένα του GPS. Αποθηκεύονται ώρες, λεπτά και δευτερόλεπτα.</p> <p>Δημιουργώ και δηλώνω μία μη προσημασμένη μεταβλητή long chars με αρχικοποιημένη τιμή ίση με το 0. Η συγκεκριμένη μεταβλητή θα χρησιμοποιηθεί από την βιβλιοθήκη TinyGPS για την προσωρινή αποθήκευση χαρακτήρων που περιλαμβάνουν τα εξαγόμενα δεδομένα του GPS.</p>
83 ^η γραμμή	<p>Δημιουργώ και δηλώνω μία μη προσημασμένη μεταβλητή short sentences με αρχικοποιημένη τιμή ίση με το 0. Η συγκεκριμένη μεταβλητή θα χρησιμοποιηθεί από την βιβλιοθήκη TinyGPS για την προσωρινή αποθήκευση δεδομένων που εξάγονται από το GPS. Η βασική αρχή του GPS στηρίζεται σε σύνολα προτάσεων που περιέχουν τα απαραίτητα δεδομένα.</p> <p>Δημιουργώ και δηλώνω μία μη προσημασμένη μεταβλητή short failed με αρχικοποιημένη τιμή ίση με το 0. Η συγκεκριμένη μεταβλητή θα χρησιμοποιηθεί από</p>

<p>την βιβλιοθήκη TinyGPS για την προσωρινή αποθήκευση των εσφαλμένων δεδομένων που εξάγονται από το GPS. Δεδομένα που δεν μπορούν να αναγνωσθούν και να κατηγοριοποιηθούν από την βιβλιοθήκη TinyGPS, καταγράφονται ως αποτυχημένα, χωρίς να γίνεται οποιαδήποτε διαδικασία εξαγωγής περαιτέρω δεδομένων.</p>
--

4.2.14 Πρόγραμμα λήψης και εικονικού τριγωνισμού – Εξαγωγή δεδομένων GPS

```

85  gps.f_get_position(&flat, &flon, &age);
86  print_float(flat, TinyGPS::GPS_INVALID_F_ANGLE, 10, 6, 1); //LATITUDE
87  print_float(flou, TinyGPS::GPS_INVALID_F_ANGLE, 11, 6, 2); //LONGITUDE
88
89  print_date(gps); //DATE AND TIME
90
91  gps.stats(&chars, &sentences, &failed);
92

```

Οι παρακάτω εντολές αποτελούν μέρος ενός μεγάλου προγράμματος που χρησιμοποιείται από την βιβλιοθήκη TinyGPS και είναι απαραίτητα για την εξαγωγή όλων των δεδομένων που λαμβάνονται από έναν δέκτη GPS και μας παρέχονται ελεύθερα από το σύνολο των δορυφόρων που βρίσκονται σε τροχιά γύρω από την Γη.

Στο συγκεκριμένο πρόγραμμα, μετέβαλα καταλλήλως και πρόσθεσα τις απαραίτητες εντολές που χρειάζομαι για να εξάγω μόνο τα δεδομένα που είναι χρήσιμα στην εργασία μου.

85 ^η γραμμή	Η συγκεκριμένη εντολή χρησιμοποιείται από την βιβλιοθήκη TinyGPS για να εξάγει δεδομένα από το GPS. Εξάγονται τα δεδομένα του γεωγραφικού πλάτους, του γεωγραφικού μήκους και της ημερομηνίας και ώρας. Η μορφή που εξάγονται και αποθηκεύονται προσωρινά, δεν είναι ανθρωπίνως ευανάγνωστη.
86 ^η γραμμή	Η συγκεκριμένη εντολή χρησιμοποιείται από την βιβλιοθήκη TinyGPS για να μετατρέψει το γεωγραφικό πλάτος από μη ανθρωπίνως ευανάγνωστη μορφή, σε μορφή που χρησιμοποιείται παγκοσμίως. Καλείται η συνάρτηση print_float για να εκτελέσει την επαναλαμβανόμενη διεργασία της μετατροπής.
87 ^η γραμμή	Η συγκεκριμένη εντολή χρησιμοποιείται από την βιβλιοθήκη TinyGPS για να μετατρέψει το γεωγραφικό μήκος από μη ανθρωπίνως ευανάγνωστη μορφή, σε μορφή που χρησιμοποιείται παγκοσμίως. Καλείται η συνάρτηση print_float για να εκτελέσει την επαναλαμβανόμενη διεργασία της μετατροπής.
89 ^η γραμμή	Η συγκεκριμένη εντολή χρησιμοποιείται από την βιβλιοθήκη TinyGPS για να μετατρέψει την ημερομηνία και την ώρα από μη ανθρωπίνως ευανάγνωστη μορφή, σε μορφή που χρησιμοποιείται παγκοσμίως. Καλείται η συνάρτηση print_date για να εκτελέσει την επαναλαμβανόμενη διεργασία της μετατροπής.

	Τα δεδομένα που εξάγονται κατά την εκτέλεση της διεργασίας print_date περιέχουν και χαρακτήρες, πέρα από αριθμητικά δεδομένα σε αντίθεση με το γεωγραφικό πλάτος και το γεωγραφικό μήκος.
91 ^η γραμμή	<p>Η συγκεκριμένη εντολή χρησιμοποιείται από την βιβλιοθήκη TinyGPS για στατιστικούς λόγους. Δεν είναι κρίσιμη στην λειτουργία του προγράμματος, αλλά περιέχει δεδομένα σχετικά με την σωστή λειτουργία του δέκτη GPS. Μας δείχνει πόσοι χαρακτήρες ελήφθησαν από τον δέκτη GPS, πόσες προτάσεις (sentences) οι οποίες είναι σωστές και πόσα δεδομένα απέτυχαν να ληφθούν σωστά.</p> <p>Οι ενδείξεις που περιμένουμε να δούμε κατά την εκτέλεση αυτής της εντολής είναι :</p> <ul style="list-style-type: none"> • chars = αριθμός συνεχώς σταθερά αυξανόμενος • sentences = σταθερός αριθμός, ανάλογα το σύνολο των προτάσεων που λαμβάνονται από τον δέκτη GPS • failed = μηδενικός αριθμός

4.2.15 Πρόγραμμα λήψης και εικονικού τριγωνισμού – Εκτέλεση εντολών δέκτη μηνύματος διάσωσης

```

93  vw_rx_start();
94  uint8_t buf[VW_MAX_MESSAGE_LEN];
95  uint8_t buflen = VW_MAX_MESSAGE_LEN;
96  vw_wait_rx_max(500);
97

```

93 ^η γραμμή	Με την εντολή vw_rx_start(); δηλώνεται στην βιβλιοθήκη VirtualWire να ενεργοποιηθεί ο ακροδέκτης που είναι συνδεδεμένος ο δέκτης που περιμένουμε να λάβει το μήνυμα διάσωσης. Χωρίς αυτή την εντολή, οποιαδήποτε πληροφορία ληφθεί από τον δέκτη, δεν θα αποθηκευτεί. Η συγκεκριμένη εντολή, για όσο χρονικό διάστημα αναμένει το σύστημα να λάβει μήνυμα, θα μας δώσει τιμή TRUE εφόσον ληφθεί κάποιο μήνυμα, ανεξάρτητα από το αν είναι σωστό ή λάθος, πλήρες ή λειψό.
94 ^η γραμμή	Με την συγκεκριμένη εντολή, αποθηκεύεται το μήνυμα που λήφθηκε, στην μεταβλητή uint8_t buf. Η συγκεκριμένη μεταβλητή χρησιμοποιείται από την βιβλιοθήκη VirtualWire ως buffer για τα δεδομένα που λαμβάνονται κατά την εκτέλεσή της.
95 ^η γραμμή	Με την συγκεκριμένη εντολή, αποθηκεύεται το μήκος του μηνύματος που λήφθηκε, στην μεταβλητή uint8_t buflen. Η συγκεκριμένη μεταβλητή χρησιμοποιείται από την βιβλιοθήκη VirtualWire για τον έλεγχο της σωστής λήψης δεδομένων σε σχέση με θόρυβο λόγω της εναέριας ασύρματης μετάδοσης

96 ^η γραμμή	<p>Με την εντολή <code>vw_wait_rx_max(500);</code> δηλώνεται στην βιβλιοθήκη VirtualWire ο χρόνος για τον οποίο θα σταματήσει την ομαλή ροή της εκτέλεσης του κυρίου προγράμματος και θα περιμένει να λάβει κάποια δεδομένα από τον δέκτη της ασύρματης ραδιομετάδοσης. Στην παρένθεση εισάγουμε τον χρόνο για τον οποίο θέλουμε να “ακούμε”.</p> <p>Στην συγκεκριμένη εργασία, ικανοποιητικός μεγάλος χρόνος χωρίς επιπτώσεις στην ομαλή ροή της εκτέλεσης του κύριου προγράμματος, είναι τα 500 milliseconds. Ο χρόνος αυτός προέκυψε μετά από δοκιμές.</p>
------------------------	---

4.2.16 Πρόγραμμα λήψης και εικονικού τριγωνισμού – Εκτέλεση εντολών ελέγχου ορθού μηνύματος

```

98  if (vw_get_message(buf, &buflen)) // Non-blocking
99  {
100     digitalWrite(msgLED, msgLEDstate); //LED showing correct message reception
101     msgLEDstate = !msgLEDstate;
102     for ([int i = 0; i < buflen; i++])
103     {
104         if(buf[i]==OK[i])
105         {
106             buttonstate = 1;
107             newPass1 = 1;
108         }

```

98 ^η γραμμή	<p>Στην συγκεκριμένη γραμμή, με την εντολή if ελέγχεται η συνθήκη που βρίσκεται μέσα στην παρένθεση. Η συγκεκριμένη συνθήκη είναι της βιβλιοθήκης VirtualWire για να ελεγχθεί ότι το μήνυμα που έλαβα μέσω του εικονικού καναλιού μετάδοσης είναι ένα ολοκληρωμένο και σωστό μήνυμα. Αν έχει ληφθεί ένα ολοκληρωμένο και σωστό μήνυμα, τότε εκτελούνται οι εντολές που βρίσκονται μέσα στην <i>if</i>.</p>
100 ^η γραμμή	<p>Με την συγκεκριμένη εντολή γράφεται μία ψηφιακή τιμή στον ακροδέκτη που βρίσκεται ως μεταβλητή μέσα στην παρένθεση. Σε αυτό το σημείο στέλνω σήμα στον ακροδέκτη που βρίσκεται συνδεδεμένο το msgLED για να ανάψει ή να σβήσει, ανάλογα την τιμή που έχει η μεταβλητή msgLEDstate. Την πρώτη φορά που τρέχει το πρόγραμμα, η τιμή είναι 0. Η αλλαγή στην κατάσταση του LED είναι μία απλή, οπτική επιβεβαίωση ότι ελήφθη ένα σωστό και ολοκληρωμένο μήνυμα.</p>
101 ^η γραμμή	<p>Με την συγκεκριμένη εντολή, αποθηκεύω την αντίθετη τιμή της μεταβλητής msgLEDstate στην μεταβλητή msgLEDstate. Με αυτόν τον τρόπο, καθώς η τιμή της μεταβλητής msgLEDstate παίρνει ψηφιακές τιμές (0 – 1, false – true) αντιστρέφω την τιμή της συγκεκριμένης μεταβλητής χωρίς να περιπλέκω το πρόγραμμα. Αυτή η</p>

	αλλαγή στην αποθηκευμένη τιμή είναι η μόνη που χρειάζεται για να πετύχω την απλή, οπτική επιβεβαίωση, όπως προανέφερα.
102 ^η γραμμή	Στην συγκεκριμένη γραμμή, με την εντολή for πραγματοποιείται το σύνολο των εντολών που την ακολουθούν, για όσες φορές ισχύει η συνθήκη που βρίσκεται μέσα στην παρένθεση. Εδώ, η επανάληψη εκτελείται για τόσες φορές όσες και το πλήθος των χαρακτήρων του μηνύματος που έλαβε η εντολή της γραμμής 95.
104 ^η γραμμή	Στην συγκεκριμένη γραμμή, με την εντολή if ελέγχεται η συνθήκη που βρίσκεται μέσα στην παρένθεση. Με την συγκεκριμένη συνθήκη ελέγχω αν η τιμή της εκάστοτε θέσης του πίνακα <code>buf[]</code> είναι ίση με την τιμή της αντίστοιχης θέσης του πίνακα <code>OK[]</code> . Με αυτόν τον τρόπο, εφόσον έχω ήδη αναφέρει ότι ο πίνακας <code>OK[]</code> περιέχει το μήνυμα διάσωσης που πρέπει να ληφθεί, ελέγχω ότι το μήνυμα που έλαβε η βιβλιοθήκη <code>VirtualWire</code> είναι το σωστό και όχι κάποια τυχαία μετάδοση.
106 ^η γραμμή	Η μεταβλητή buttonstate χρησιμοποιείται για δοκιμές με τοπική και κατά βούληση προσομοίωση του μηνύματος διάσωσης, αλλά και ως σημαία για την μη παράκαμψη εντολών του προγράμματος.
107 ^η γραμμή	Με την αλλαγή της μεταβλητής newPass1 από 0 σε τιμή 1, δηλώνεται ότι το σύστημα λήψης έχει βρει ένα νέο στίγμα από το σύστημα εκπομπής του μηνύματος διάσωσης. Η συγκεκριμένη μεταβλητή θα χρησιμοποιηθεί παρακάτω για να το έλεγχο και την εκτέλεση συγκεκριμένων εντολών.

4.2.17 Πρόγραμμα λήψης και εικονικού τριγωνισμού – Εκτέλεση εντολών ελέγχου ορθού μηνύματος

```

109         else
110         {
111             delay(5);
112         }
113     }
114     ground_found = ground_found + 1; //appears ONLY when a complete correct message was received
115 }
116 vw_rx_stop();
117

```

109 ^η γραμμή	Στην συγκεκριμένη γραμμή, η εντολή else έρχεται σε συνέχεια της if για να δηλώσει ένα σύνολο εντολών που θα εκτελεστούν σε περίπτωση που δεν ισχύσει η συνθήκη που βρίσκεται εντός της παρένθεσης στην παραπάνω if .
111 ^η γραμμή	Με την συγκεκριμένη εντολή delay δίνω εντολή στο πρόγραμμα να αναμείνει για 5 χιλιοστά του δευτερολέπτου, ως αποτέλεσμα της μη λήψης του σωστού και ολοκληρωμένου μηνύματος διάσωσης.

114 ^η γραμμή	<p>Με την συγκεκριμένη εντολή, αυξάνω την τιμή της μεταβλητής ground_found κατά 1. Με την συγκεκριμένη μεταβλητή μετρώ τις φορές που θεωρώ ότι βρήκε το σύστημα λήψης ένα πλήρες μήνυμα διάσωσης από το σύστημα εκπομπής. Η μεταβλητή ground_found θα χρησιμοποιηθεί από το πρόγραμμα ως επιπλέον έλεγχος κατά την διαδικασία του εικονικού πολυγωνισμού για τον υπολογισμό της προσέγγισης του πραγματικού σημείου που βρίσκεται το σύστημα εκπομπής του μηνύματος διάσωσης.</p>
116 ^η γραμμή	<p>Στην συγκεκριμένη γραμμή και αφού έχουν ολοκληρωθεί οι διαδικασίες που πρέπει να γίνουν κατά την αναμονή και την λήψη του μηνύματος διάσωσης, θα μπει η εντολή <code>vw_rx_stop()</code> της βιβλιοθήκης <code>VirtualWire</code> για να σταματήσει η εκτέλεση του συνόλου των εντολών που σχετίζονται με την μετάδοση δεδομένων μέσω του εικονικού καναλιού, έτσι ώστε να απελευθερωθούν οι πόροι που χρησιμοποιούσε η βιβλιοθήκη <code>VirtualWire</code>, όπως ο ακροδέκτης που συνδέεται ο δέκτης και να συνεχιστεί η ροή του προγράμματος.</p>

4.2.18 Πρόγραμμα λήψης και εικονικού τριγωνισμού – Εκτέλεση εντολών αποθήκευσης στίγματος GPS

```

118   if (buttonstate == 1)
119   {
120       buttonstate = 0; //Preparing for next triggering
121       Lat[Nu] = SD_lat; //Store current latitude into a vertice's X point
122       Long[Nu] = SD_lon; //Store current latitude into a vertice's Y point
123
124       midPointLog();
125
126       Nu++;
127
128       Serial.println(Nu);
129       lastSignal = newSignal;
130   }
131   lastPass();
132

```

118 ^η γραμμή	<p>Στην συγκεκριμένη γραμμή, με την εντολή if ελέγχεται η συνθήκη που βρίσκεται μέσα στην παρένθεση. Με την συγκεκριμένη συνθήκη ελέγχεται αν η τιμή της μεταβλητής buttonstate είναι ίση με 1. Η τιμή θα είναι ίση με 1, εφόσον έχει ληφθεί ένα πλήρες και σωστό μήνυμα διάσωσης, όπως ανέφερα και έδειξα παραπάνω. Αν είναι ίση με 1 και άρα έχει ληφθεί το μήνυμα διάσωσης, τότε εκτελούνται οι εντολές που βρίσκονται μέσα στην <i>if</i>.</p>
-------------------------	--

120 ^η γραμμή	Στην συγκεκριμένη γραμμή, δίνεται στην μεταβλητή buttonstate η τιμή 0, καθώς προετοιμάζεται για την επόμενη λήψη του μηνύματος διάσωσης. Μέχρι τότε, οπότε και θα πάρει την τιμή 1, θα πρέπει να παρακάμπτονται οι εντολές που εκτελούνται και επεξηγώ αυτή την στιγμή.
121 ^η γραμμή	Με την συγκεκριμένη εντολή, αποθηκεύεται η τιμή που έχει η μεταβλητή SD_lat στην αντίστοιχη θέση του πίνακα Lat[] που δείχνει η τιμή της μεταβλητής Nu . Η μεταβλητή SD_lat έχει αποθηκευμένο το γεωγραφικό πλάτος του σημείου που λήφθηκε ένα σωστό και ολοκληρωμένο μήνυμα διάσωσης. Με αυτόν τον τρόπο, έχει αρχίσει να σχηματίζεται το πολύγωνο που θα χρησιμοποιηθεί για τον εικονικό πολυγωνισμό για τον υπολογισμό της προσέγγισης του πραγματικού σημείου που βρίσκεται το σύστημα εκπομπής του μηνύματος διάσωσης.
122 ^η γραμμή	Με την συγκεκριμένη εντολή, αποθηκεύεται η τιμή που έχει η μεταβλητή SD_lon στην αντίστοιχη θέση του πίνακα Long[] που δείχνει η τιμή της μεταβλητής Nu . Η μεταβλητή SD_lat έχει αποθηκευμένο το γεωγραφικό μήκος του σημείου που λήφθηκε ένα σωστό και ολοκληρωμένο μήνυμα διάσωσης. Με αυτόν τον τρόπο, έχει αρχίσει να σχηματίζεται το πολύγωνο που θα χρησιμοποιηθεί για τον εικονικό πολυγωνισμό για τον υπολογισμό της προσέγγισης του πραγματικού σημείου που βρίσκεται το σύστημα εκπομπής του μηνύματος διάσωσης.
124 ^η γραμμή	Στην συγκεκριμένη γραμμή, η εντολή midPointLog() καλεί την διεργασία midPointLog η οποία είναι υπεύθυνη για την αποθήκευση ενός συνόλου πληροφοριών σχετικά με τα ενδιάμεσα σημεία λήψης του μηνύματος διάσωσης. Οι εντολές που εκτελούνται και τα αποτελέσματα της, θα δειχθούν παρακάτω αναλυτικότερα.
126 ^η γραμμή	Η μεταβλητή Nu χρησιμοποιείται ως αύξουσα μεταβλητή ένδειξης του συνόλου των μηνυμάτων διάσωσης που ελήφθησαν σωστά και ολοκληρωμένα από το σύστημα εκπομπής και των αποθηκευμένων γεωγραφικών πλατών και μηκών. Στην συγκεκριμένη γραμμή, η συγκεκριμένη εντολή Nu αυξάνει την τιμή της κατά 1, αφού υπήρξε μία λήψη του σωστού και ολοκληρωμένου μηνύματος διάσωσης και μία πλήρης αποθήκευση του γεωγραφικού πλάτους και του γεωγραφικού μήκους της θέσης κατά την οποία ελήφθη το συγκεκριμένο μήνυμα διάσωσης.
128 ^η γραμμή	Με την εντολή Serial.println() δίνεται η εντολή στο πρόγραμμα να τυπώσει (ή καλύτερα να στείλει) στην σειριακή θύρα του επεξεργαστή τα δεδομένα εντός της παρένθεσης σε μορφή κατανοητή από τον άνθρωπο. Στην συγκεκριμένη περίπτωση, η σειριακή θύρα που χρησιμοποιείται είναι η 0, αλλά δεν γράφεται ως Serial0.println σε

	<p>αντίθεση με τις υπόλοιπες σειριακές θύρες (Serial1, Serial2, Serial3) που έχει ο συγκεκριμένος Arduino Mega.</p> <p>Η προσθήκη ln στο τέλος της εντολής print, ξεκινάει μία νέα γραμμή μετά την εκτύπωση των δεδομένων.</p> <p>Η συγκεκριμένη εντολή, εκτυπώνει το περιεχόμενο της μεταβλητής Nu στην οθόνη του υπολογιστή που είναι συνδεδεμένος ο Arduino Mega για λόγους παρακολούθησης της σωστής λειτουργίας του προγράμματος.</p>
129 ^η γραμμή	<p>Με της εντολή lastSignal = newSignal; αποθηκεύεται η χρονική τιμή που λήφθηκε το τελευταίο μήνυμα διάσωσης από το σύστημα εκπομπής και η οποία είναι ήδη αποθηκευμένη στην μεταβλητή newSignal, στην μεταβλητή lastSignal. Με αυτόν τον τρόπο θα γίνει διαχωρισμός ανάμεσα στα περάσματα του συστήματος λήψης, όπως έχω προαναφέρει, για να μην υπάρχει σύγχυση μεταξύ των δεδομένων που θα έχει ως συνέπεια μη ακριβή αποτελέσματα.</p>
131 ^η γραμμή	<p>Στην συγκεκριμένη γραμμή, εκτελείται η εντολή lastPass; με την οποία καλείται η αντίστοιχη διεργασία. Η συγκεκριμένη διεργασία υπολογίζει τους χρόνους μεταξύ των ληφθέντων σημάτων και κάνει τον διαχωρισμό ανάμεσα στα περάσματα των συστημάτων λήψης και εκπομπής του μηνύματος διάσωσης.</p> <p>Η χρήση της συγκεκριμένης διεργασίας θα δειχθεί και θα αναλυθεί μετά το τέλος του κύριου προγράμματος, όπου βρίσκονται όλες οι διεργασίες για λόγους ομοιόμορφης γραφής.</p>

4.2.19 Πρόγραμμα λήψης και εικονικού τριγωνισμού – Εκτέλεση εντολών υπολογισμού κέντρου βάρους

```

134   if (Nu == 7 || (newPass == 1&& newPass1 == 1)) //Polygon's 7th vertice reached or interval exceeded
135   {
136       newPass = 0;
137       NuMax = Nu;
138       Serial.println("NuMax");
139       Serial.println(NuMax);
140

```

134 ^η γραμμή	<p>Στην συγκεκριμένη γραμμή, με την εντολή if ελέγχεται η συνθήκη που βρίσκεται μέσα στην παρένθεση. Με την συγκεκριμένη γραφή της εντολής ελέγχονται 2 συνθήκες. Στην πρώτη συνθήκη ελέγχεται αν η μεταβλητή Nu έχει τιμή ίση με 7 ή () αν ισχύει η συνθήκη μέσα στην δεύτερη παρένθεση.</p> <p>Με την εισαγωγή δεύτερης παρένθεσης, δηλαδή παρένθεσης μέσα στην παρένθεση, ξεχωρίζω ένα σύνολο συνθηκών από ένα άλλο.</p>
-------------------------	---

	<p>Έτσι, η δεύτερη συνθήκη που ελέγχεται είναι εάν η τιμή της μεταβλητής newPass είναι ίση με 1 και (&&) εάν η τιμή της μεταβλητής newPass1 είναι και αυτή ίση με 1.</p> <p>Αν ισχύουν οι παραπάνω συνθήκες, εκτελούνται οι εντολές που περικλείονται από τα σύμβολα ορίων της εντολής if.</p>
136 ^η γραμμή	<p>Στην συγκεκριμένη γραμμή, με την εντολή newPass = 0 αποθηκεύεται η τιμή 0 στην συγκεκριμένη μεταβλητή η οποία χρησιμοποιείται ως σημαία. Με αυτόν τον τρόπο μηδενίζω την συγκεκριμένη σημαία, αφού έχει χρησιμοποιηθεί για τον έλεγχο της προηγούμενης εντολής (134^η γραμμή), για να χρησιμοποιηθεί στην επόμενη επανεκκίνηση του κύριου προγράμματος.</p>
137 ^η γραμμή	<p>Στην συγκεκριμένη γραμμή, η εντολή NuMax = Nu; αποθηκεύει την τιμή της μεταβλητής Nu στην μεταβλητή NuMax. Η τελευταία θα χρησιμοποιηθεί από την διαδικασία επανάληψης κατά τον υπολογισμό του κέντρου βάρους του συνόλου των γεωγραφικών συντεταγμένων.</p> <p>Η χρήση της θα δειχθεί παρακάτω αναλυτικότερα.</p>
138 ^η γραμμή	<p>Με την εντολή Serial.println() δίνεται η εντολή στο πρόγραμμα να τυπώσει (ή καλύτερα να στείλει) στην σειριακή θύρα του επεξεργαστή τα δεδομένα εντός της παρένθεσης σε μορφή κατανοητή από τον άνθρωπο. Στην συγκεκριμένη περίπτωση, η σειριακή θύρα που χρησιμοποιείται είναι η 0, αλλά δεν γράφεται ως Serial0.println σε αντίθεση με τις υπόλοιπες σειριακές θύρες (Serial1, Serial2, Serial3) που έχει ο συγκεκριμένος Arduino Mega.</p> <p>Η προσθήκη ln στο τέλος της εντολής print, ξεκινάει μία νέα γραμμή μετά την εκτύπωση των δεδομένων.</p> <p>Η συγκεκριμένη εντολή, εκτυπώνει την γραμματοσειρά (ή σύνολο χαρακτήρων) NuMax στην οθόνη του υπολογιστή που είναι συνδεδεμένος ο Arduino Mega για λόγους παρακολούθησης της σωστής λειτουργίας του προγράμματος.</p>
139 ^η γραμμή	<p>Με την εντολή Serial.println() δίνεται η εντολή στο πρόγραμμα να τυπώσει (ή καλύτερα να στείλει) στην σειριακή θύρα του επεξεργαστή τα δεδομένα εντός της παρένθεσης σε μορφή κατανοητή από τον άνθρωπο. Στην συγκεκριμένη περίπτωση, η σειριακή θύρα που χρησιμοποιείται είναι η 0, αλλά δεν γράφεται ως Serial0.println σε αντίθεση με τις υπόλοιπες σειριακές θύρες (Serial1, Serial2, Serial3) που έχει ο συγκεκριμένος Arduino Mega.</p> <p>Η προσθήκη ln στο τέλος της εντολής print, ξεκινάει μία νέα γραμμή μετά την εκτύπωση των δεδομένων.</p>

	<p>Η συγκεκριμένη εντολή, εκτυπώνει το περιεχόμενο της μεταβλητής NuMax στην οθόνη του υπολογιστή που είναι συνδεδεμένος ο Arduino Mega για λόγους παρακολούθησης της σωστής λειτουργίας του προγράμματος.</p>
--	---

4.2.20 Πρόγραμμα λήψης και εικονικού τριγωνισμού – Εκτέλεση εντολών υπολογισμού κέντρου βάρους

```

141 //Calculating Centroid
142     for(Nu = 0;Nu <NuMax; Nu++ ) //Getting 1 point at a time
143     {
144         void midPointLog();
145         Serial.println(Lat[Nu], 6);
146         Serial.println(Long[Nu], 6);
147     }

```

141 ^η γραμμή	<p>Στην συγκεκριμένη γραμμή υπάρχουν μόνο σχόλια για το σύνολο των εντολών που ακολουθούν. Τα σχόλια χρησιμοποιούνται για την διευκόλυνση κατανόησης μεμονωμένων εντολών, τμημάτων ή άλλων σημείων του προγράμματος. Στην συγκεκριμένη περίπτωση, τα σχόλια μας πληροφορούν ότι οι εντολές που ακολουθούν είναι για τον υπολογισμό του κέντρου βάρους του συνόλου των γεωγραφικών συντεταγμένων.</p>
142 ^η γραμμή	<p>Στην συγκεκριμένη γραμμή, με την εντολή for πραγματοποιείται το σύνολο των εντολών που την ακολουθούν, για όσες φορές ισχύει η συνθήκη που βρίσκεται μέσα στην παρένθεση. Εδώ, η επανάληψη εκτελείται για τόσες φορές όσες είναι η τιμή της μεταβλητής NuMax. Αυτός ο αριθμός αντιστοιχεί στις φορές που ελήφθη το μήνυμα διάσωσης από το σύστημα εκπομπής και άρα ισούται με το σύνολο των αποθηκευμένων γεωγραφικών πλατών και μηκών.</p> <p>Το σχόλιο της συγκεκριμένης γραμμής αναφέρει ότι η επανάληψη της for υπολογίζει 1 σημείο σε κάθε εκτέλεσή της.</p>
144 ^η γραμμή	<p>Στην συγκεκριμένη γραμμή, εκτελείται η εντολή void midPointLog(); με την οποία καλείται η αντίστοιχη διεργασία. Η εκτέλεση void σημαίνει ότι δεν θα μας επιστρέψει κάποιο αποτέλεσμα η διεργασία για να χρειάζεται αποθήκευση αυτού, παρά μόνο θα εκτελέσει το σύνολο των εντολών της και οποιαδήποτε αλλαγή σε μεταβλητές, θα πραγματοποιηθεί εντός της.</p> <p>Η συγκεκριμένη διεργασία αποθηκεύει σε μία κάρτα μνήμης, το σύνολο των μέσων γεωγραφικών πλατών, καθώς και ημερομηνία και ώρα.</p>
145 ^η γραμμή	<p>Με την εντολή Serial.println() δίνεται η εντολή στο πρόγραμμα να τυπώσει (ή καλύτερα να στείλει) στην σειριακή θύρα του επεξεργαστή τα δεδομένα εντός της</p>

	<p>παρένθεσης σε μορφή κατανοητή από τον άνθρωπο. Στην συγκεκριμένη περίπτωση, η σειριακή θύρα που χρησιμοποιείται είναι η 0, αλλά δεν γράφεται ως Serial0.println σε αντίθεση με τις υπόλοιπες σειριακές θύρες (Serial1, Serial2, Serial3) που έχει ο συγκεκριμένος Arduino Mega.</p> <p>Η προσθήκη ln στο τέλος της εντολής print, ξεκινάει μία νέα γραμμή μετά την εκτύπωση των δεδομένων.</p> <p>Η συγκεκριμένη εντολή, εκτυπώνει το περιεχόμενο της εκάστοτε θέσης, η οποία ορίζεται από την μεταβλητή Nu, του πίνακα Lat[] με ακρίβεια 6 δεκαδικών ψηφίων, στην οθόνη του υπολογιστή που είναι συνδεδεμένος ο Arduino Mega για λόγους παρακολούθησης της σωστής λειτουργίας του προγράμματος.</p>
146 ⁿ γραμμή	<p>Με την εντολή Serial.println() δίνεται η εντολή στο πρόγραμμα να τυπώσει (ή καλύτερα να στείλει) στην σειριακή θύρα του επεξεργαστή τα δεδομένα εντός της παρένθεσης σε μορφή κατανοητή από τον άνθρωπο. Στην συγκεκριμένη περίπτωση, η σειριακή θύρα που χρησιμοποιείται είναι η 0, αλλά δεν γράφεται ως Serial0.println σε αντίθεση με τις υπόλοιπες σειριακές θύρες (Serial1, Serial2, Serial3) που έχει ο συγκεκριμένος Arduino Mega.</p> <p>Η προσθήκη ln στο τέλος της εντολής print, ξεκινάει μία νέα γραμμή μετά την εκτύπωση των δεδομένων.</p> <p>Η συγκεκριμένη εντολή, εκτυπώνει το περιεχόμενο της εκάστοτε θέσης, η οποία ορίζεται από την μεταβλητή Nu, του πίνακα Long[] με ακρίβεια 6 δεκαδικών ψηφίων, στην οθόνη του υπολογιστή που είναι συνδεδεμένος ο Arduino Mega για λόγους παρακολούθησης της σωστής λειτουργίας του προγράμματος.</p>

4.2.21 Πρόγραμμα λήψης και εικονικού τριγωνισμού – Εκτέλεση εντολών υπολογισμού κέντρου βάρους

```

148 /* When calculating last point, the last coordinate's
149    next cell (Nu+1) has no value stored (or the value
150    from last pass).
151    That's why we have to store current's pass first |
152    coordinate there (en.wikipedia.org/wiki/Centroid#Centroid_of_polygon)
153 */
154     if (Nu == NuMax-1)
155     {
156         Lat[NuMax] = Lat[0];
157         Long[NuMax] = Long[0];
158     }
159     Deg2Rad();
160     Rad2Cart();
161     Xsum = Xsum + CartX[Nu];
162     Ysum = Ysum + CartY[Nu];
163     Zsum = Zsum + CartZ[Nu];
164 }

```

Για το υπολογισμό του κέντρου βάρους, θα πρέπει το σχήμα για το οποίο θα γίνει υπολογισμός, να είναι κλειστό. Δηλαδή η τελευταία κορυφή του πολυγώνου θα πρέπει να είναι στο ίδιο ακριβώς σημείο με την πρώτη. Επειδή αυτό είναι ανέφικτο και ανούσιο για το συγκεκριμένο πείραμα, πολύ απλά η πρώτη κορυφή θα πρέπει να υπολογιστεί ως διπλή.

Αυτό σημαίνει ότι κατά την αποθήκευση των γεωγραφικών συντεταγμένων του πειράματος, οι τελευταίες γεωγραφικές συντεταγμένες που λήφθηκε ένα πλήρες και σωστό μήνυμα διάσωσης από το σύστημα εκπομπής, θα λάβουν θέση αποθήκευσης την προτελευταία θέση του πίνακα. Σε εκείνο το σημείο, οι συντεταγμένες που βρίσκονται στην 1^η θέση αυτόματα θα αντιγράφονται και στην τελευταία θέση του πίνακα.

145 ^η γραμμή έως 153 ^η γραμμή	<p>Οι συγκεκριμένες γραμμές χρησιμοποιήθηκαν για να γραφούν σχόλια σχετικά με τις εντολές που ακολουθούν. Ο συγκεκριμένος τρόπος γραφής των σχολίων, τα καθιστά ευανάγνωστα για την άνετη κατανόηση του προγράμματος.</p> <p>Αντί του κλασσικού τρόπου ένδειξης σχολιασμού (//) χρησιμοποιήθηκε εναλλακτικός τρόπος, όπου σημειώνονται η αρχή των σχολίων (/*) και το τέλος αυτών (*/).</p> <p>Τα σχόλια που ακολουθούν, περιγράφουν τον λόγο που η πρώτη αποθηκευμένη τιμή που περιέχει γεωγραφικό μήκος και γεωγραφικό πλάτος, πρέπει να αποθηκευτεί και στην τελευταία θέση του πίνακα των γεωγραφικών συντεταγμένων.</p>
154 ^η γραμμή	<p>Στην συγκεκριμένη γραμμή, με την εντολή if ελέγχεται η συνθήκη που βρίσκεται μέσα στην παρένθεση. Με την συγκεκριμένη συνθήκη ελέγχεται αν η τιμή της μεταβλητής Nu είναι ίση με την τιμή της μεταβλητής NuMax-1. Η τιμή θα γίνει ίση κατά το τελευταίο ζεύγος αποθηκευμένων γεωγραφικών συντεταγμένων, εφόσον η τιμή της μεταβλητής NuMax προκύπτει από την μεταβλητή Nu. Όταν θα γίνει ίση και άρα έχει φτάσει το πρόγραμμα στο προτελευταίο ζεύγος αποθηκευμένων γεωγραφικών συντεταγμένων, τότε εκτελούνται οι εντολές που βρίσκονται μέσα στην <i>if</i>.</p> <p>Οι εντολές που εκτελούνται εντός της <i>if</i> θα δειχθούν παρακάτω αναλυτικότερα.</p>
156 ^η γραμμή	<p>Στην συγκεκριμένη γραμμή, η εντολή εκτελείται εφόσον έχει γίνει αληθής η συνθήκη της παραπάνω <i>if</i>. Με την συγκεκριμένη εντολή, αποθηκεύεται η τιμή του γεωγραφικού πλάτους του πρώτου σημείου που λήφθηκε ένα πλήρες και σωστό μήνυμα διάσωσης, στην τελευταία θέση του πίνακα Lat[] που υποδεικνύεται από την μεταβλητή NuMax διότι η τιμή της μεταβλητής δείχνει 1 θέση αριστερότερα στον πίνακα.</p>
157 ^η γραμμή	<p>Στην συγκεκριμένη γραμμή, η εντολή εκτελείται εφόσον έχει γίνει αληθής η συνθήκη της παραπάνω <i>if</i>. Με την συγκεκριμένη εντολή, αποθηκεύεται η τιμή του γεωγραφικού μήκους του πρώτου σημείου που λήφθηκε ένα πλήρες και σωστό μήνυμα διάσωσης,</p>

	στην τελευταία θέση του πίνακα Long [] που υποδεικνύεται από την μεταβλητή NuMax διότι η τιμή της μεταβλητής δείχνει 1 θέση αριστερότερα στον πίνακα.
159 ^η γραμμή	<p>Στην συγκεκριμένη γραμμή, η εντολή Deg2Rad() καλεί την διεργασία Deg2Rad η οποία είναι υπεύθυνη για την μετατροπή των γεωγραφικών συντεταγμένων από μοίρες σε ακτίνια, μονάδες οι οποίες είναι απαραίτητες για το ορθότερο υπολογισμό του κέντρου βάρους.</p> <p>Οι εντολές που εκτελούνται και τα αποτελέσματα της, θα δειχθούν παρακάτω αναλυτικότερα.</p>
160 ^η γραμμή	<p>Στην συγκεκριμένη γραμμή, η εντολή Rad2Cart() καλεί την διεργασία Rad2Cart η οποία είναι υπεύθυνη για την μετατροπή των γεωγραφικών συντεταγμένων από ακτίνια σε καρτεσιανές συντεταγμένες 3 αξόνων (X,Y,Z), μονάδες οι οποίες είναι απαραίτητες για το ορθότερο υπολογισμό του κέντρου βάρους.</p> <p>Οι εντολές που εκτελούνται και τα αποτελέσματα της, θα δειχθούν παρακάτω αναλυτικότερα.</p>
161 ^η γραμμή	<p>Στην συγκεκριμένη γραμμή, η εντολή Xsum = Xsum + CartX[Nu] αποθηκεύει στην μεταβλητή Xsum το άθροισμα της προηγούμενης τιμής της μαζί με την τιμή της εκάστοτε θέσης του πίνακα CartX.</p> <p>Η χρήση της συγκεκριμένης μεταβλητής, καθώς και ο σκοπός της, έχει εξηγηθεί στον τρόπο με τον οποίο υπολογίζεται το κέντρο βάρους των γεωγραφικών συντεταγμένων.</p>
162 ^η γραμμή	<p>Στην συγκεκριμένη γραμμή, η εντολή Ysum = Ysum + CartY[Nu] αποθηκεύει στην μεταβλητή Ysum το άθροισμα της προηγούμενης τιμής της μαζί με την τιμή της εκάστοτε θέσης του πίνακα CartY.</p> <p>Η χρήση της συγκεκριμένης μεταβλητής, καθώς και ο σκοπός της, έχει εξηγηθεί στον τρόπο με τον οποίο υπολογίζεται το κέντρο βάρους των γεωγραφικών συντεταγμένων.</p>
163 ^η γραμμή	<p>Στην συγκεκριμένη γραμμή, η εντολή Zsum = Zsum + CartZ[Nu] αποθηκεύει στην μεταβλητή Zsum το άθροισμα της προηγούμενης τιμής της μαζί με την τιμή της εκάστοτε θέσης του πίνακα CartZ.</p> <p>Η χρήση της συγκεκριμένης μεταβλητής, καθώς και ο σκοπός της, έχει εξηγηθεί στον τρόπο με τον οποίο υπολογίζεται το κέντρο βάρους των γεωγραφικών συντεταγμένων.</p>

4.2.22 Πρόγραμμα λήψης και εικονικού τριγωνισμού – Εκτέλεση εντολών υπολογισμού κέντρου βάρους

```

165
166 CentX = Xsum / NuMax;
167 CentY = Ysum / NuMax;
168 CentZ = Zsum / NuMax;
169
170 Cart2Rad();
171 Rad2Deg();
172
  
```

166 ^η γραμμή	<p>Στην συγκεκριμένη γραμμή, η εντολή CentX = Xsum / NuMax αποθηκεύει στην μεταβλητή CentX το αποτέλεσμα της διαίρεσης της μεταβλητής Xsum με την μεταβλητή NuMax.</p> <p>Η χρήση της συγκεκριμένης μεταβλητής, καθώς και ο σκοπός της, έχει εξηγηθεί στον τρόπο με τον οποίο υπολογίζεται το κέντρο βάρους των γεωγραφικών συντεταγμένων.</p>
167 ^η γραμμή	<p>Στην συγκεκριμένη γραμμή, η εντολή CentY = Ysum / NuMax αποθηκεύει στην μεταβλητή CentY το αποτέλεσμα της διαίρεσης της μεταβλητής Ysum με την μεταβλητή NuMax.</p> <p>Η χρήση της συγκεκριμένης μεταβλητής, καθώς και ο σκοπός της, έχει εξηγηθεί στον τρόπο με τον οποίο υπολογίζεται το κέντρο βάρους των γεωγραφικών συντεταγμένων.</p>
168 ^η γραμμή	<p>Στην συγκεκριμένη γραμμή, η εντολή CentZ = Zsum / NuMax αποθηκεύει στην μεταβλητή CentZ το αποτέλεσμα της διαίρεσης της μεταβλητής Zsum με την μεταβλητή NuMax.</p> <p>Η χρήση της συγκεκριμένης μεταβλητής, καθώς και ο σκοπός της, έχει εξηγηθεί στον τρόπο με τον οποίο υπολογίζεται το κέντρο βάρους των γεωγραφικών συντεταγμένων.</p>
170 ^η γραμμή	<p>Στην συγκεκριμένη γραμμή, η εντολή Cart2Rad() καλεί την διεργασία Cart2Rad η οποία είναι υπεύθυνη για την μετατροπή των γεωγραφικών συντεταγμένων από καρτεσιανές συντεταγμένες 3 αξόνων (X,Y,Z) σε ακτίνια, μονάδες οι οποίες είναι απαραίτητες για την απεικόνιση του κέντρου βάρους σε γεωγραφικό μήκος και γεωγραφικό πλάτος.</p> <p>Οι εντολές που εκτελούνται και τα αποτελέσματα της, θα δειχθούν παρακάτω αναλυτικότερα.</p>
171 ^η γραμμή	<p>Στην συγκεκριμένη γραμμή, η εντολή Rad2Deg() καλεί την διεργασία Rad2Deg η οποία είναι υπεύθυνη για την μετατροπή των γεωγραφικών συντεταγμένων από ακτίνια σε μοίρες, μονάδες οι οποίες είναι απαραίτητες για την απεικόνιση του κέντρου βάρους σε γεωγραφικό μήκος και γεωγραφικό πλάτος.</p>

	Οι εντολές που εκτελούνται και τα αποτελέσματα της, θα δειχθούν παρακάτω αναλυτικότερα.
--	---

4.2.23 Πρόγραμμα λήψης και εικονικού τριγωνισμού – Εκτέλεση εντολών υπολογισμού κέντρου βάρους

```

173     state = !state;
174     digitalWrite(LED, state);
175     Serial.print("Center Latitude : ");
176     Serial.println(CentLat, 6);
177     Serial.print("Center Longitude : ");
178     Serial.println(CentLong, 6);
  
```

173 ^η γραμμή	Με την συγκεκριμένη εντολή, αποθηκεύω την αντίθετη τιμή της μεταβλητής state στην μεταβλητή state . Με αυτόν τον τρόπο, καθώς η τιμή της μεταβλητής state παίρνει ψηφιακές τιμές (0 – 1, false – true) αντιστρέφω την τιμή της συγκεκριμένης μεταβλητής χωρίς να περιπλέκω το πρόγραμμα. Αυτή η αλλαγή στην αποθηκευμένη τιμή είναι η μόνη που χρειάζεται για να πετύχω την απλή, οπτική επιβεβαίωση ότι το πρόγραμμα έχει υπολογίσει ένα νέο κέντρο βάρους των ληφθέντων και αποθηκευμένων γεωγραφικών συντεταγμένων..
174 ^η γραμμή	Με την συγκεκριμένη εντολή γράφεται μία ψηφιακή τιμή στον ακροδέκτη που βρίσκεται ως μεταβλητή μέσα στην παρένθεση. Σε αυτό το σημείο στέλνω σήμα στον ακροδέκτη που βρίσκεται συνδεδεμένο το LED για να ανάψει ή να σβήσει, ανάλογα την τιμή που έχει η μεταβλητή state . Την πρώτη φορά που τρέχει το πρόγραμμα, η τιμή είναι 0. Η αλλαγή στην κατάσταση του LED είναι μία απλή, οπτική επιβεβαίωση ότι υπολογίστηκε ένα νέο κέντρο βάρους των ληφθέντων και αποθηκευμένων γεωγραφικών συντεταγμένων.
175 ^η γραμμή	Με την εντολή Serial.print() δίνεται η εντολή στο πρόγραμμα να τυπώσει (ή καλύτερα να στείλει) στην σειριακή θύρα του επεξεργαστή τα δεδομένα εντός της παρένθεσης σε μορφή κατανοητή από τον άνθρωπο. Στην συγκεκριμένη περίπτωση, η σειριακή θύρα που χρησιμοποιείται είναι η 0, αλλά δεν γράφεται ως Serial0.print σε αντίθεση με τις υπόλοιπες σειριακές θύρες (Serial1, Serial2, Serial3) που έχει ο συγκεκριμένος Arduino Mega Η συγκεκριμένη εντολή, εκτυπώνει το σύνολο των χαρακτήρων “Center Latitude : ”.
176 ^η γραμμή	Με την εντολή Serial.println() δίνεται η εντολή στο πρόγραμμα να τυπώσει (ή καλύτερα να στείλει) στην σειριακή θύρα του επεξεργαστή τα δεδομένα εντός της παρένθεσης σε μορφή κατανοητή από τον άνθρωπο. Στην συγκεκριμένη περίπτωση, η

	<p>σειριακή θύρα που χρησιμοποιείται είναι η 0, αλλά δεν γράφεται ως Serial0.println σε αντίθεση με τις υπόλοιπες σειριακές θύρες (Serial1, Serial2, Serial3) που έχει ο συγκεκριμένος Arduino Mega.</p> <p>Η προσθήκη ln στο τέλος της εντολής print, ξεκινάει μία νέα γραμμή μετά την εκτύπωση των δεδομένων.</p> <p>Η συγκεκριμένη εντολή, εκτυπώνει το περιεχόμενο της μεταβλητής CentLat με ακρίβεια 6 δεκαδικών ψηφίων.</p> <p>Η προβολή με ακρίβεια 6 δεκαδικών ψηφίων είναι αναγκαία, καθώς το γεωγραφικό πλάτος χρειάζεται τόσα για να επιτευχθεί ακρίβεια με απόκλιση εκατοστών, δηλαδή ανεκτή για το συγκεκριμένο πείραμα, επάνω στο έδαφος.</p>
177 ^η γραμμή	<p>Με την εντολή Serial.print() δίνεται η εντολή στο πρόγραμμα να τυπώσει (ή καλύτερα να στείλει) στην σειριακή θύρα του επεξεργαστή τα δεδομένα εντός της παρένθεσης σε μορφή κατανοητή από τον άνθρωπο. Στην συγκεκριμένη περίπτωση, η σειριακή θύρα που χρησιμοποιείται είναι η 0, αλλά δεν γράφεται ως Serial0.print σε αντίθεση με τις υπόλοιπες σειριακές θύρες (Serial1, Serial2, Serial3) που έχει ο συγκεκριμένος Arduino Mega</p> <p>Η συγκεκριμένη εντολή, εκτυπώνει το σύνολο των χαρακτήρων “<i>Center Longitude :</i>”.</p>
178 ^η γραμμή	<p>Με την εντολή Serial.println() δίνεται η εντολή στο πρόγραμμα να τυπώσει (ή καλύτερα να στείλει) στην σειριακή θύρα του επεξεργαστή τα δεδομένα εντός της παρένθεσης σε μορφή κατανοητή από τον άνθρωπο. Στην συγκεκριμένη περίπτωση, η σειριακή θύρα που χρησιμοποιείται είναι η 0, αλλά δεν γράφεται ως Serial0.println σε αντίθεση με τις υπόλοιπες σειριακές θύρες (Serial1, Serial2, Serial3) που έχει ο συγκεκριμένος Arduino Mega.</p> <p>Η προσθήκη ln στο τέλος της εντολής print, ξεκινάει μία νέα γραμμή μετά την εκτύπωση των δεδομένων.</p> <p>Η συγκεκριμένη εντολή, εκτυπώνει το περιεχόμενο της μεταβλητής CentLong με ακρίβεια 6 δεκαδικών ψηφίων.</p> <p>Η προβολή με ακρίβεια 6 δεκαδικών ψηφίων είναι αναγκαία, καθώς το γεωγραφικό μήκος χρειάζεται τόσα για να επιτευχθεί ακρίβεια με απόκλιση εκατοστών, δηλαδή ανεκτή για το συγκεκριμένο πείραμα, επάνω στο έδαφος.</p>

4.2.24 Πρόγραμμα λήψης και εικονικού τριγωνισμού – Εκτέλεση εντολών αποθήκευσης δεδομένων και μηδενισμού μεταβλητών

```

179
180     sdLoggingTEMP();
181
182     zeroValues(); //Zeroing incrementing values for
183                 //new pass, to avoid the error of adding the old value
184     Nu = 0;
185     newPass1 = 0;
186
187 }
  
```

180 ^η γραμμή	<p>Στην συγκεκριμένη γραμμή, η εντολή sdLoggingTEMP() καλεί την διεργασία sdLoggingTEMP η οποία είναι υπεύθυνη για την αποθήκευση των γεωγραφικών συντεταγμένων του κέντρου βάρους, καθώς και της ημερομηνίας και ώρας.</p> <p>Οι εντολές που εκτελούνται και τα αποτελέσματα της, θα δειχθούν παρακάτω αναλυτικότερα.</p>
182 ^η γραμμή	<p>Στην συγκεκριμένη γραμμή, η εντολή zeroValues() καλεί την διεργασία zeroValues η οποία είναι υπεύθυνη για τον μηδενισμό όλων των μεταβλητών που επηρεάζονται από την ροή του προγράμματος. Με αυτόν τον τρόπο γίνεται κατά μία έννοια <i>software reset</i> έτσι ώστε να μην περάσουν δεδομένα από προηγούμενες εκτελέσεις, το οποίο θα έχει ως συνέπεια την πλήρη αμφισβήτηση των δεδομένων και των αποτελεσμάτων.</p> <p>Οι εντολές που εκτελούνται και τα αποτελέσματα της, θα δειχθούν παρακάτω αναλυτικότερα.</p>
184 ^η γραμμή	<p>Στην συγκεκριμένη γραμμή, η εντολή Nu = 0 αποθηκεύει την τιμή 0 στην μεταβλητή Nu. Με αυτόν τον τρόπο επιτυγχάνεται η αρχικοποίηση της συγκεκριμένης μεταβλητής, έτσι ώστε να δεχθεί νέα δεδομένα κατά την επανεκτέλεση του προγράμματος.</p> <p>Η συγκεκριμένη αρχικοποίηση είναι αναγκαία, καθώς η μεταβλητή Nu είναι η μεταβλητή μέτρησης των μηνυμάτων διάσωσης που ελήφθησαν και τυχόν υπέρβαση του ορίου που έχει τεθεί στην αρχή του προγράμματος, θα προκαλέσει περαιτέρω αδυναμία εκτέλεσης του κύριου μέρους του προγράμματος.</p>
185 ^η γραμμή	<p>Στην συγκεκριμένη γραμμή, η εντολή newPass1 = 0 αποθηκεύει την τιμή 0 στην μεταβλητή newPass1. Με αυτόν τον τρόπο επιτυγχάνεται η αρχικοποίηση της συγκεκριμένης μεταβλητής, έτσι ώστε να δεχθεί νέα δεδομένα κατά την επανεκτέλεση του προγράμματος.</p>

<p>Η συγκεκριμένη αρχικοποίηση είναι αναγκαία, καθώς η μεταβλητή newPass1 είναι η μεταβλητή που διαφοροποιεί τα εκάστοτε όρια μεταξύ μηνυμάτων διάσωσης.</p>

4.2.25 Πρόγραμμα λήψης και εικονικού τριγωνισμού – Εκτέλεση εντολών απόρριψης δεδομένων εκτός ορίων

```

188     else if(Nu > 7) Nu = 0; //If, for some reason,
189                               // there is an 8th or greater trigger,
190                               // we ignore it
191
192     smartdelay(500);
193 }

```

<p>188^η γραμμή έως 190^η γραμμή</p>	<p>Στην συγκεκριμένη γραμμή, η εντολή else if είναι η 2η και τελευταία συνθήκη της εντολής if της 134^{ης} γραμμής, και ελέγχεται, εφόσον η εντολή if επιστρέψει FALSE και άρα δεν εκτελέσει τις εντολές που βρίσκονται εντός των ορίων της και πριν από την συγκεκριμένη εντολή else if.</p> <p>Η συνθήκη που ελέγχεται εντός της else if, είναι αν η τιμή της μεταβλητής Nu είναι μεγαλύτερη από 7. Εφόσον η μεταβλητή Nu είναι αυτή στην οποία αποθηκεύεται ο αριθμός των σωστών και πλήρων μηνυμάτων διάσωσης και έχει δηλωθεί από την αρχή του προγράμματος ότι ικανοποιητικός αριθμός δεδομένων που δίνει ορθά αποτελέσματα είναι έως 7, οποιαδήποτε μεταβολή της μεταβλητής Nu εκτός αυτών των ορίων, θεωρείται άσκοπη και πιθανότατα εσφαλμένη, οπότε απορρίπτεται.</p> <p>Αυτό επιτυγχάνεται με την εντολή Nu = 0, την οποία έγραψα στην ίδια γραμμή με την προηγούμενη εντολή, για να φανεί η διαφορά στην ευκολότερη κατανόηση του προγράμματος αν ακολουθείται η γραφή των εντολών σε ξεχωριστές γραμμές και με τα κατάλληλα κενά.</p> <p>Έτσι, η μεταβλητή Nu αρχικοποιείται πριν την επανεκκίνηση του προγράμματος και οποιοδήποτε σφάλμα την έθεσε εκτός ορίων, δεν επηρεάζει την επανεκτέλεση του προγράμματος.</p> <p>Δίπλα από τις εντολές, τα σχόλια που έχω γράψει, διευκολύνουν την κατανόηση των συγκεκριμένων εντολών και αναφέρουν συνοπτικά την παραπάνω επεξήγηση.</p>
<p>192^η γραμμή</p>	<p>Στην συγκεκριμένη γραμμή, μέσω της εντολής smartdelay(500) καλείται η διεργασία smartdelay, η οποία καθυστερεί το πρόγραμμα για 0,5 δευτερόλεπτα. Τα δεδομένα που στέλνονται σε αυτήν κατά την κλήση της, είναι ο αριθμός εντός της παρένθεσης που αναφέρεται στα milliseconds που επιθυμώ να καθυστερήσει το πρόγραμμα στο συγκεκριμένο σημείο.</p>

<p>Η λειτουργία της συγκεκριμένης διεργασίας θα δειχθεί παρακάτω αναλυτικότερα.</p>

4.2.26 Πρόγραμμα λήψης και εικονικού τριγωνισμού – Διεργασία καθυστέρησης smartdelay

```

195 static void smartdelay(unsigned long ms)
196 {
197     unsigned long start = millis();
198     do
199     {
200         while (Serial1.available())
201             gps.encode(Serial1.read());
202     } while (millis() - start < ms);
203 }

```

195 ^η γραμμή	<p>Στην συγκεκριμένη γραμμή, δημιουργείται η διεργασία smartdelay η οποία δηλώνεται πως είναι <i>void</i>, δηλαδή δεν επιστρέφει δεδομένα μετά την εκτέλεσή της. Επίσης, δηλώνεται ως <i>static</i>, δηλαδή δημιουργείται και υπάρχει μόνο εντός του συγκεκριμένου προγράμματος και δεν μπορεί κληθεί με την συγκεκριμένη ονομασία πέρα από αυτό. Στην συγκεκριμένη περίπτωση δεν δημιουργεί κάποιο θέμα αυτή η δήλωση, καθώς το πρόγραμμα που εκτελείται κάθε στιγμή, είναι ένα.</p> <p>Εντός της παρένθεσης βρίσκονται τα δεδομένα που θα πρέπει να παρέχονται στην διεργασία σε κάθε κλήση της μέσα στο πρόγραμμα. Με την δήλωση unsigned long ms δημιουργείται η μεταβλητή ms τύπου unsigned long.</p>
197 ^η γραμμή	<p>Στην συγκεκριμένη γραμμή, με την εντολή unsigned long start = millis() δημιουργείται η μεταβλητή start τύπου unsigned long και αποθηκεύονται σε αυτήν τα δεδομένα που παρέχονται από την κλήση της διεργασίας millis. Η συγκεκριμένη διεργασία χρησιμοποιείται από την βιβλιοθήκη TinyGPS και χρησιμοποιείται για να δείχνει τον χρόνο που έχει περάσει από την αρχή του προγράμματος σε milliseconds.</p>
198 ^η γραμμή	<p>Στην συγκεκριμένη γραμμή, η εντολή do εκτελεί τις εντολές εντός των ορίων της έως ότου εκπληρωθεί η συνθήκη που βρίσκεται στο τέλος της. Με αυτό τον τρόπο, δίνεται η δυνατότητα να εκτελεστούν οι εντολές μετά την εντολή do, τουλάχιστον 1 φορά.</p>
200 ^η γραμμή	<p>Στην συγκεκριμένη γραμμή, η εντολή while (Serial1.available()) εκτελείται επ' αόριστον όσο ικανοποιείται η συνθήκη εντός της παρένθεσης. Στην συγκεκριμένη περίπτωση, η συνθήκη ικανοποιείται όσο η σειριακή θύρα Serial 1 είναι διαθέσιμη.</p>
201 ^η γραμμή	<p>Η συγκεκριμένη γραμμή αποτελεί το σύνολο των εντολών που εκτελούνται εφόσον πληρείται η προηγούμενη συνθήκη. Η συγκεκριμένη εντολή χρησιμοποιείται από την βιβλιοθήκη TinyGPS και αποτελεί κλήση του κομματιού encode της διεργασίας gps, δηλαδή ξεχωρίζει τα απαραίτητα δεδομένα των γεωγραφικών συντεταγμένων.</p>

202 ^η γραμμή	<p>Η συγκεκριμένη γραμμή είναι η συνθήκη που πρέπει να ικανοποιείται για την εκτέλεση της προαναφερθείσας εντολής do. Όσο το αποτέλεσμα της αφαίρεσης του χρόνου που αυξάνεται και μετριέται σε milliseconds από τον στατικό χρόνο που ξεκίνησε η εκτέλεση της συγκεκριμένης διεργασίας, παραμένει μικρότερο από την τιμή της μεταβλητής ms και η οποία είναι ο χρόνος που εγώ δήλωσα κατά την κλήση της συγκεκριμένης διεργασίας, τότε οι εντολές εντός των ορίων της επανάληψης do – while επανεκτελούνται.</p>
-------------------------	--

4.2.27 Πρόγραμμα λήψης και εικονικού τριγωνισμού – Διεργασία εκτύπωσης γεωγραφικών δεδομένων

print_float

```

205 static void print_float(float val, float invalid, int len, int prec, int SD_val)
206 {
207     if (val == invalid)
208     {
209         0;
210     }
211     else
212     {
213         if (SD_val == 1) SD_lat = val;
214         else if (SD_val == 2) SD_lon = val;
215         int vi = abs((int)val);
216         int flen = prec + (val < 0.0 ? 2 : 1); // . and -
217         flen += vi >= 1000 ? 4 : vi >= 100 ? 3 : vi >= 10 ? 2 : 1;
218     }
219     smartdelay(0);
220 }

```

205 ^η γραμμή	<p>Δημιουργώ και δηλώνω μία συνάρτηση τύπου static void και όνομα print_float με τις εξής παραμέτρους τύπου float και όνομα val, τύπου float και όνομα invalid, τύπου int και όνομα len, τύπου int και όνομα prec, τύπου int και όνομα SD_val, τύπου int και όνομα Nu.</p> <p>Η διεργασία δηλώνεται πως είναι <i>void</i>, δηλαδή δεν επιστρέφει δεδομένα μετά την εκτέλεσή της. Επίσης, δηλώνεται ως <i>static</i>, δηλαδή δημιουργείται και υπάρχει μόνο εντός του συγκεκριμένου προγράμματος και δεν μπορεί κληθεί με την συγκεκριμένη ονομασία πέρα από αυτό. Στην συγκεκριμένη περίπτωση δεν δημιουργεί κάποιο θέμα αυτή η δήλωση, καθώς το πρόγραμμα που εκτελείται κάθε στιγμή, είναι ένα.</p> <p>Η λειτουργία της συγκεκριμένης διεργασίας είναι η αποκωδικοποίηση των δεδομένων που εισέρχονται από το GPS και η εξαγωγή απαραίτητων πληροφοριών σε μορφή <i>float</i>.</p>
-------------------------	--

207 ^η γραμμή έως 210 ^η γραμμή	<p>Στην συγκεκριμένη γραμμή, με την εντολή if ελέγχεται η συνθήκη που βρίσκεται μέσα στην παρένθεση. Με την συγκεκριμένη συνθήκη ελέγχεται αν η τιμή της μεταβλητής val είναι ίση με τιμή <i>invalid</i>.</p> <p>Αν είναι ίση με <i>invalid</i>, τότε εκτελείται η εντολή 0 και παρακάμπτονται οι εντολές που ακολουθούν.</p>
211 ^η γραμμή	<p>Στην συγκεκριμένη γραμμή, η εντολή else είναι η 2η και τελευταία συνθήκη της εντολής if της 207^{ης} γραμμής, και ελέγχεται, εφόσον η εντολή if επιστρέψει FALSE και άρα δεν εκτελέσει τις εντολές που βρίσκονται εντός των ορίων της και πριν από την συγκεκριμένη εντολή else.</p> <p>Η συνθήκη που ελέγχεται από την else, είναι οποιαδήποτε πληροί τις προϋποθέσεις για να είναι η τιμή της μεταβλητής val διαφορετική από <i>invalid</i>.</p>
213 ^η γραμμή	<p>Στην συγκεκριμένη γραμμή, με την εντολή if ελέγχεται η συνθήκη που βρίσκεται μέσα στην παρένθεση. Με την συγκεκριμένη συνθήκη ελέγχεται αν η τιμή της μεταβλητής SD_val είναι ίση με 1.</p> <p>Αν είναι ίση με 1, τότε εκτελείται η εντολή SD_lat = val η οποία αποθηκεύει την τιμή που περιέχει η μεταβλητή <i>val</i> στην μεταβλητή <i>SD_lat</i> που είναι η μεταβλητή αποθήκευσης του γεωγραφικού πλάτους.</p>
214 ^η γραμμή	<p>Στην συγκεκριμένη γραμμή, η εντολή else if είναι η 2η και τελευταία συνθήκη της εντολής if της 213^{ης} γραμμής, και ελέγχεται, εφόσον η εντολή if επιστρέψει FALSE και άρα δεν εκτελέσει τις εντολές που βρίσκονται εντός των ορίων της και πριν από την συγκεκριμένη εντολή else if.</p> <p>Η συνθήκη που ελέγχεται εντός της else if, είναι αν η τιμή της μεταβλητής SD_val είναι ίση με 2.</p> <p>Αν είναι ίση με 2, τότε εκτελείται η εντολή SD_lon = val η οποία αποθηκεύει την τιμή που περιέχει η μεταβλητή <i>val</i> στην μεταβλητή <i>SD_lon</i> που είναι η μεταβλητή αποθήκευσης του γεωγραφικού μήκους.</p>
215 ^η γραμμή έως 217 ^η γραμμή	<p>Οι συγκεκριμένες γραμμές αποτελούν ένα σύνολο εντολών που εκτελούνται από την βιβλιοθήκη TinyGPS και είναι υπεύθυνες για την μορφοποίηση των γεωγραφικών συντεταγμένων από απλή αριθμοσειρά που εξέρχεται από το GPS σε αριθμούς αποδεκτούς από το παγκόσμιο σύστημα συντεταγμένων.</p>
219 ^η γραμμή	<p>Στην συγκεκριμένη γραμμή, μέσω της εντολής smartdelay(0) καλείται η διεργασία <i>smartdelay</i>, η οποία καθυστερεί το πρόγραμμα για 0 δευτερόλεπτα. Τα δεδομένα που στέλνονται σε αυτήν κατά την κλήση της, είναι ο αριθμός εντός της παρένθεσης που αναφέρεται στα <i>milliseconds</i> που επιθυμώ να καθυστερήσει το πρόγραμμα στο συγκεκριμένο σημείο.</p>

4.2.28 Πρόγραμμα λήψης και εικονικού τριγωνισμού – Διεργασία εκτύπωσης ημερομηνίας και ώρας `print_date`

```

222 static void print_date(TinyGPS &gps)
223 {
224     int year;
225     byte month, day, hour, minute, second, hundredths;
226     unsigned long age;
227     gps.crack_datetime(&year, &month, &day, &hour, &minute, &second, &hundredths, &age);
228     if (age == TinyGPS::GPS_INVALID_AGE)
229     {
230         SD_date_time = "INVALID";
231     }
  
```

222 ^η γραμμή	<p>Δημιουργώ και δηλώνω μία συνάρτηση τύπου static void και όνομα print_date με παράμετρο τύπου TinyGPS (είναι “τύπος” από συγκεκριμένη βιβλιοθήκη που χρησιμοποιείται στο πρόγραμμα) και όνομα &gps (δηλαδή θα έχει όνομα εκεί που θα δείχνει η συγκεκριμένη μεταβλητή και όχι το όνομα gps αυτό καθαυτό).</p> <p>Η διεργασία δηλώνεται πως είναι <i>void</i>, δηλαδή δεν επιστρέφει δεδομένα μετά την εκτέλεσή της. Επίσης, δηλώνεται ως <i>static</i>, δηλαδή δημιουργείται και υπάρχει μόνο εντός του συγκεκριμένου προγράμματος και δεν μπορεί κληθεί με την συγκεκριμένη ονομασία πέρα από αυτό. Στην συγκεκριμένη περίπτωση δεν δημιουργεί κάποιο θέμα αυτή η δήλωση, καθώς το πρόγραμμα που εκτελείται κάθε στιγμή, είναι ένα.</p>
224 ^η γραμμή έως 226 ^η γραμμή	<p>Στις συγκεκριμένες γραμμές γίνεται δημιουργία και δήλωση των εξής μεταβλητών των εξής τύπων:</p> <ul style="list-style-type: none"> • int year • byte month, day, hour, minute, second, hundredths • unsigned long age <p>Οι μεταβλητές αυτές θα χρησιμοποιηθούν από την βιβλιοθήκη TinyGPS και για αποθήκευση των ημερολογιακών και ωρολογιακών δεδομένων που αποκωδικοποιούνται από την απλή αριθμοσειρά που εξάγεται από τα δεδομένα του GPS.</p>
227 ^η γραμμή	<p>Στην συγκεκριμένη γραμμή εκτελείται η εντολή gps.crack_datetime(&year, &month, &day, &hour, &minute, &second, &hundredths, &age) η οποία χρησιμοποιείται από την βιβλιοθήκη TinyGPS για να καλέσει την διεργασία <i>crack_datetime</i> και να αποκωδικοποιήσει τα ημερολογιακά και ωρολογιακά δεδομένα που αντιγράφονται από την απλή αριθμοσειρά που εξάγεται από τα δεδομένα του GPS.</p>

228 ^η γραμμή έως 230 ^η γραμμή	<p>Στην συγκεκριμένη γραμμή, με την εντολή if ελέγχεται η συνθήκη που βρίσκεται μέσα στην παρένθεση. Με την συγκεκριμένη συνθήκη ελέγχεται αν η τιμή της μεταβλητής age είναι ίση με <i>invalid</i> (<i>TinyGPS::GPS_INVALID_AGE</i>).</p> <p>Στην επόμενη γραμμή, και αν ισχύει η παραπάνω συνθήκη, εκτελείται η εντολή που η μεταβλητή SD_date_time παίρνει τιμή ίση με την συμβολοσειρά <i>INVALID</i>.</p>
---	---

4.2.29 Πρόγραμμα λήψης και εικονικού τριγωνισμού – Διεργασία εκτύπωσης ημερομηνίας και ώρας `print_date`

```

232     else
233     {
234         char sz[32];
235         sprintf(sz, "%02d/%02d/%02d %02d:%02d:%02d ",
236             month, day, year, hour, minute, second);
237         SD_date_time = sz;
238     }
239     smartdelay(0);
240 }
241

```

232 ^η γραμμή	<p>Στην συγκεκριμένη γραμμή, η εντολή else είναι η 2η και τελευταία συνθήκη της εντολής if της 228^{ης} γραμμής, και ελέγχεται, εφόσον η εντολή if επιστρέψει FALSE και άρα δεν εκτελέσει τις εντολές που βρίσκονται εντός των ορίων της και πριν από την συγκεκριμένη εντολή else.</p> <p>Η συνθήκη που ελέγχεται από την else, είναι οποιαδήποτε πληροί τις προϋποθέσεις για να είναι η τιμή της μεταβλητής age διαφορετική από <i>invalid</i>.</p>
234 ^η γραμμή έως 237 ^η γραμμή	<p>Αρχικά δημιουργείται και δηλώνεται η μεταβλητή sz τύπου char μεγέθους 32, δηλαδή ένας πίνακας χαρακτήρων 32 θέσεων.</p> <p>Στην συνέχεια, εκτελείται η εντολή sprint που αποτελεί εντολή της βιβλιοθήκης TinyGPS και είναι υπεύθυνη για την εκτύπωση των συμβολοσειρών των ημερολογιακών και ωρολογιακών δεδομένων, στον παραπάνω πίνακα sz.</p> <p>Το σύνολο των αποκωδικοποιηθέντων ημερολογιακών και ωρολογιακών δεδομένων που πλέον εμφανίζονται σε μορφή ευανάγνωστη και κατανοητή από τον άνθρωπο, αποθηκεύεται στην μεταβλητή SD_date_time.</p>
239 ^η γραμμή	<p>Στην συγκεκριμένη γραμμή, μέσω της εντολής smartdelay(0) καλείται η διεργασία <code>smartdelay</code>, η οποία καθυστερεί το πρόγραμμα για 0 δευτερόλεπτα. Τα δεδομένα που στέλνονται σε αυτήν κατά την κλήση της, είναι ο αριθμός εντός της παρένθεσης που αναφέρεται στα <code>milliseconds</code> που επιθυμώ να καθυστερήσει το πρόγραμμα στο συγκεκριμένο σημείο.</p>

4.2.30 Πρόγραμμα λήψης και εικονικού τριγωνισμού – Διεργασία μετατροπής μοιρών σε ακτίνια

```

242 void Deg2Rad()
243 {
244     LatRad[Nu] = (Lat[Nu]*pi)/180;
245     LongRad[Nu] = (Long[Nu]*pi)/180;
246 }
247
  
```

242 ^η γραμμή	<p>Δημιουργώ και δηλώνω μία συνάρτηση τύπου void και όνομα Deg2Rad χωρίς παραμέτρους. Η μεταβλητή Nu δεν αλλάζει κατά την κλήση και την εκτέλεση της συγκεκριμένης διεργασίας, οπότε δεν χρειάζεται να σταλεί ως παράμετρος και παρέχεται ως συγκεκριμένη τιμή από το κυρίως πρόγραμμα.</p> <p>Η διεργασία δηλώνεται πως είναι <i>void</i>, δηλαδή δεν επιστρέφει δεδομένα μετά την εκτέλεσή της, παρά μόνο εκτελεί τις εντολές που βρίσκονται εντός αυτής.</p> <p>Δημιούργησα την συγκεκριμένη διεργασία με σκοπό να συγκεντρώσω τις εντολές που εκτελούνται για την μετατροπή των γεωγραφικών συντεταγμένων από μοίρες σε ακτίνια, έτσι ώστε στο κυρίως πρόγραμμα, η κλήση της διεργασίας και η γραφή του προγράμματος να το καθιστούν ευανάγνωστο και λειτουργικό.</p>
244 ^η γραμμή	<p>Στην συγκεκριμένη γραμμή, το αποτέλεσμα των πράξεων $(\text{Lat}[\text{Nu}] * \pi) / 180$ αποθηκεύεται στην θέση του πίνακα LatRad[] που δείχνει η μεταβλητή Nu.</p> <p>Οι παραπάνω πράξεις μετατρέπουν το γεωγραφικό πλάτος από μοίρες σε ακτίνια. Η συγκεκριμένη πράξη είναι απαραίτητη για διαδικασία υπολογισμού του κέντρου βάρους των γεωγραφικών συντεταγμένων.</p> <p>Η διαδικασία αυτή δείχνεται σε βήματα και αναλύεται στο αντίστοιχο κομμάτι της εργασίας αυτής.</p>
245 ^η γραμμή	<p>Στην συγκεκριμένη γραμμή, το αποτέλεσμα των πράξεων $(\text{Long}[\text{Nu}] * \pi) / 180$ αποθηκεύεται στην θέση του πίνακα LongRad[] που δείχνει η μεταβλητή Nu.</p> <p>Οι παραπάνω πράξεις μετατρέπουν το γεωγραφικό μήκος από μοίρες σε ακτίνια. Η συγκεκριμένη πράξη είναι απαραίτητη για διαδικασία υπολογισμού του κέντρου βάρους των γεωγραφικών συντεταγμένων.</p> <p>Η διαδικασία αυτή δείχνεται σε βήματα και αναλύεται στο αντίστοιχο κομμάτι της εργασίας αυτής.</p>

4.2.31 Πρόγραμμα λήψης και εικονικού τριγωνισμού – Διεργασία μετατροπής ακτινίων σε καρτεσιανές συντεταγμένες

```

248 void Rad2Cart ()
249 {
250   CartX[Nu] = cos(LatRad[Nu]) * cos(LongRad[Nu]);
251   CartY[Nu] = cos(LatRad[Nu]) * sin(LongRad[Nu]);
252   CartZ[Nu] = sin(LatRad[Nu]);
253 }
254

```

248 ^η γραμμή	<p>Δημιουργώ και δηλώνω μία συνάρτηση τύπου void και όνομα Rad2Cart χωρίς παραμέτρους. Η μεταβλητή Nu δεν αλλάζει κατά την κλήση και την εκτέλεση της συγκεκριμένης διεργασίας, οπότε δεν χρειάζεται να σταλεί ως παράμετρος και παρέχεται ως συγκεκριμένη τιμή από το κυρίως πρόγραμμα.</p> <p>Η διεργασία δηλώνεται πως είναι <i>void</i>, δηλαδή δεν επιστρέφει δεδομένα μετά την εκτέλεσή της, παρά μόνο εκτελεί τις εντολές που βρίσκονται εντός αυτής.</p> <p>Δημιούργησα την συγκεκριμένη διεργασία με σκοπό να συγκεντρώσω τις εντολές που εκτελούνται για την μετατροπή των γεωγραφικών συντεταγμένων από ακτίνια σε καρτεσιανές συντεταγμένες, έτσι ώστε στο κυρίως πρόγραμμα, η κλήση της διεργασίας και η γραφή του προγράμματος να το καθιστούν ευανάγνωστο και λειτουργικό.</p>
250 ^η γραμμή	<p>Στην συγκεκριμένη γραμμή, το αποτέλεσμα των πράξεων cos(LatRad[Nu])*cos(LongRad[Nu]) αποθηκεύεται στην θέση του πίνακα CartX[] που δείχνει η μεταβλητή Nu.</p> <p>Οι παραπάνω πράξεις μετατρέπουν το γεωγραφικό πλάτος και το γεωγραφικό μήκος από ακτίνια σε καρτεσιανές συντεταγμένες του άξονα των X. Η συγκεκριμένη πράξη είναι απαραίτητη για διαδικασία υπολογισμού του κέντρου βάρους των γεωγραφικών συντεταγμένων.</p> <p>Η διαδικασία αυτή δείχνεται σε βήματα και αναλύεται στο αντίστοιχο κομμάτι της εργασίας αυτής.</p>
251 ^η γραμμή	<p>Στην συγκεκριμένη γραμμή, το αποτέλεσμα των πράξεων cos(LatRad[Nu])*sin(LongRad[Nu]) αποθηκεύεται στην θέση του πίνακα CartY[] που δείχνει η μεταβλητή Nu.</p> <p>Οι παραπάνω πράξεις μετατρέπουν το γεωγραφικό πλάτος και το γεωγραφικό μήκος από ακτίνια σε καρτεσιανές συντεταγμένες του άξονα των Y. Η συγκεκριμένη πράξη</p>

	<p>είναι απαραίτητη για διαδικασία υπολογισμού του κέντρου βάρους των γεωγραφικών συντεταγμένων.</p> <p>Η διαδικασία αυτή δείχνεται σε βήματα και αναλύεται στο αντίστοιχο κομμάτι της εργασίας αυτής.</p>
252 ^η γραμμή	<p>Στην συγκεκριμένη γραμμή, το αποτέλεσμα της πράξης sin(LatRad[Nu]) αποθηκεύεται στην θέση του πίνακα CartZ[] που δείχνει η μεταβλητή Nu.</p> <p>Οι παραπάνω πράξεις μετατρέπουν το γεωγραφικό πλάτος και το γεωγραφικό μήκος από ακτίνια σε καρτεσιανές συντεταγμένες του άξονα των Z. Η συγκεκριμένη πράξη είναι απαραίτητη για διαδικασία υπολογισμού του κέντρου βάρους των γεωγραφικών συντεταγμένων.</p> <p>Η διαδικασία αυτή δείχνεται σε βήματα και αναλύεται στο αντίστοιχο κομμάτι της εργασίας αυτής.</p>

4.2.32 Πρόγραμμα λήψης και εικονικού τριγωνισμού – Διεργασία μετατροπής καρτεσιανών συντεταγμένων σε ακτίνια

```

255 void Cart2Rad()
256 {
257     CentLongRad = atan2(CentY,CentX);
258     Hyp = hypot(CentX,CentY);
259     CentLatRad = atan2(CentZ,Hyp);
260 }
261

```

255 ^η γραμμή	<p>Δημιουργώ και δηλώνω μία συνάρτηση τύπου void και όνομα Cart2Rad χωρίς παραμέτρους.</p> <p>Η διεργασία δηλώνεται πως είναι <i>void</i>, δηλαδή δεν επιστρέφει δεδομένα μετά την εκτέλεση της, παρά μόνο εκτελεί τις εντολές που βρίσκονται εντός αυτής.</p> <p>Δημιούργησα την συγκεκριμένη διεργασία με σκοπό να συγκεντρώσω τις εντολές που εκτελούνται για την μετατροπή των καρτεσιανών συντεταγμένων του κέντρου βάρους σε ακτίνια, έτσι ώστε στο κυρίως πρόγραμμα, η κλήση της διεργασίας και η γραφή του προγράμματος να το καθιστούν ευανάγνωστο και λειτουργικό.</p>
257 ^η γραμμή	<p>Στην συγκεκριμένη γραμμή, το αποτέλεσμα της πράξης atan2(CentY, CentX) αποθηκεύεται στην μεταβλητή CentLongRad.</p> <p>Οι παραπάνω πράξεις χρησιμοποιούν τις καρτεσιανές συντεταγμένες του κέντρου βάρους των αξόνων X και Y για να υπολογίσουν το γεωγραφικό μήκος σε ακτίνια.</p>

	<p>Η διαδικασία αυτή δείχνεται σε βήματα και αναλύεται στο αντίστοιχο κομμάτι της εργασίας αυτής.</p>
258 ^η γραμμή	<p>Στην συγκεκριμένη γραμμή, το αποτέλεσμα της πράξης hypot(CentX, CentY) αποθηκεύεται στην μεταβλητή Hyp.</p> <p>Οι παραπάνω πράξεις χρησιμοποιούν τις καρτεσιανές συντεταγμένες του κέντρου βάρους των αξόνων X και Y για να υπολογίσουν την Υποτείνουσα.</p>
259 ^η γραμμή	<p>Στην συγκεκριμένη γραμμή, το αποτέλεσμα της πράξης atan2(CentZ, Hyp) αποθηκεύεται στην μεταβλητή CentLatRad.</p> <p>Οι παραπάνω πράξεις χρησιμοποιούν τις καρτεσιανές συντεταγμένες του κέντρου βάρους του άξονα Z και την Υποτείνουσα για να υπολογίσουν το γεωγραφικό πλάτος σε ακτίνια.</p> <p>Η διαδικασία αυτή δείχνεται σε βήματα και αναλύεται στο αντίστοιχο κομμάτι της εργασίας αυτής.</p>

4.2.33 Πρόγραμμα λήψης και εικονικού τριγωνισμού – Διεργασία μετατροπής ακτινίων σε μοίρες

```

262 void Rad2Deg ()
263 {
264     CentLat = CentLatRad*(180/pi);
265     CentLong = CentLongRad*(180/pi);
266 }
267

```

262 ^η γραμμή	<p>Δημιουργώ και δηλώνω μία συνάρτηση τύπου void και όνομα Rad2Deg χωρίς παραμέτρους.</p> <p>Η διεργασία δηλώνεται πως είναι <i>void</i>, δηλαδή δεν επιστρέφει δεδομένα μετά την εκτέλεση της, παρά μόνο εκτελεί τις εντολές που βρίσκονται εντός αυτής.</p> <p>Δημιούργησα την συγκεκριμένη διεργασία με σκοπό να συγκεντρώσω τις εντολές που εκτελούνται για την μετατροπή του κέντρου βάρους από ακτίνια σε μοίρες, έτσι ώστε στο κυρίως πρόγραμμα, η κλήση της διεργασίας και η γραφή του προγράμματος να το καθιστούν ευανάγνωστο και λειτουργικό.</p>
264 ^η γραμμή	<p>Στην συγκεκριμένη γραμμή, το αποτέλεσμα των πράξεων (CentLatRad *(180 / π)) αποθηκεύεται στην μεταβλητή CentLat.</p> <p>Οι παραπάνω πράξεις μετατρέπουν το γεωγραφικό πλάτος από ακτίνια σε μοίρες. Η συγκεκριμένη πράξη είναι απαραίτητη για διαδικασία παρουσίασης του κέντρου βάρους και αποτύπωσής του στον παγκόσμιο χάρτη.</p>

	<p>Η διαδικασία αυτή δείχνεται σε βήματα και αναλύεται στο αντίστοιχο κομμάτι της εργασίας αυτής.</p>
265 ^η γραμμή	<p>Στην συγκεκριμένη γραμμή, το αποτέλεσμα των πράξεων (CentLongRad *(180 / π)) αποθηκεύεται στην μεταβλητή CentLong.</p> <p>Οι παραπάνω πράξεις μετατρέπουν το γεωγραφικό μήκος από ακτίνια σε μοίρες. Η συγκεκριμένη πράξη είναι απαραίτητη για διαδικασία παρουσίασης του κέντρου βάρους και αποτύπωσής του στον παγκόσμιο χάρτη.</p> <p>Η διαδικασία αυτή δείχνεται σε βήματα και αναλύεται στο αντίστοιχο κομμάτι της εργασίας αυτής.</p>

4.2.34 Πρόγραμμα λήψης και εικονικού τριγωνισμού – Διεργασία μηδενισμού προσωρινών μεταβλητών

```

268 void zeroValues()
269 {
270     Serial.println("Zero values");
271     for (Nu = 0; Nu < NuMax; Nu++)
272     {
273         LatRad[Nu] = 0, LongRad[Nu] = 0, CartX[Nu] = 0, CartY[Nu] = 0, CartZ[Nu] = 0;
274     }
275     Xsum = 0, Ysum = 0, Zsum = 0, CentX = 0, CentY = 0, CentZ = 0;
276     CentLongRad = 0, Hyp = 0, CentLatRad = 0, CentLat = 0, CentLong = 0;
277 }
278

```

268 ^η γραμμή	<p>Δημιουργώ και δηλώνω μία συνάρτηση τύπου void και όνομα zeroValues χωρίς παραμέτρους.</p> <p>Η διεργασία δηλώνεται πως είναι <i>void</i>, δηλαδή δεν επιστρέφει δεδομένα μετά την εκτέλεση της, παρά μόνο εκτελεί τις εντολές που βρίσκονται εντός αυτής.</p> <p>Δημιούργησα την συγκεκριμένη διεργασία με σκοπό να συγκεντρώσω τις εντολές που εκτελούνται για τον μηδενισμό των μεταβλητών και των πινάκων που χρησιμοποιούνται και δεν χρειάζεται να αποθηκεύουν δεδομένα κατά την επανεκτέλεση του προγράμματος, έτσι ώστε στο κυρίως πρόγραμμα, η κλήση της διεργασίας και η γραφή του προγράμματος να το καθιστούν ευανάγνωστο και λειτουργικό.</p>
270 ^η γραμμή	<p>Με την εντολή Serial.println() δίνεται η εντολή στο πρόγραμμα να τυπώσει στην σειριακή θύρα του επεξεργαστή τα δεδομένα εντός της παρένθεσης σε μορφή κατανοητή από τον άνθρωπο. Στην συγκεκριμένη περίπτωση, η σειριακή θύρα που χρησιμοποιείται είναι η 0, αλλά δεν γράφεται ως Serial0.println σε αντίθεση με τις</p>

	<p>υπόλοιπες σειριακές θύρες (Serial1, Serial2, Serial3) που έχει ο συγκεκριμένος Arduino Mega.</p> <p>Η προσθήκη In στο τέλος της εντολής print, ξεκινάει μία νέα γραμμή μετά την εκτύπωση των δεδομένων.</p> <p>Η συγκεκριμένη εντολή, εκτυπώνει τους χαρακτήρες “Zero values” στην οθόνη του υπολογιστή που είναι συνδεδεμένος ο Arduino Mega για λόγους παρακολούθησης της σωστής λειτουργίας του προγράμματος.</p>
271 ^η γραμμή	<p>Στην συγκεκριμένη γραμμή, με την εντολή for πραγματοποιείται το σύνολο των εντολών που την ακολουθούν, για όσες φορές ισχύει η συνθήκη που βρίσκεται μέσα στην παρένθεση. Εδώ, η επανάληψη εκτελείται για τόσες φορές όσες είναι η τιμή της μεταβλητής NuMax. Αυτός ο αριθμός αντιστοιχεί στις φορές που ελήφθη το μήνυμα διάσωσης από το σύστημα εκπομπής και άρα ισούται με το σύνολο των αποθηκευμένων γεωγραφικών πλατών και μηκών.</p>
273 ^η γραμμή	<p>Στην συγκεκριμένη γραμμή, οι πίνακες LatRad[], LongRad[], CartX[], CartY[] και CartZ[] παίρνουν την τιμή 0, για κάθε θέση που δείχνει η μεταβλητή Nu, η οποία αλλάζει τιμή σύμφωνα με την παραπάνω επανάληψη for.</p> <p>Στους συγκεκριμένους πίνακες εισάγω την τιμή 0, στο σημείο όπου έχω εξάγει τα δεδομένα που χρειάζομαι και οι τιμές τους δεν μου είναι απαραίτητες.</p> <p>Με αυτόν τον τρόπο γίνεται κατά μία έννοια <i>software reset</i> έτσι ώστε να μην περάσουν δεδομένα από προηγούμενες εκτελέσεις, το οποίο θα έχει ως συνέπεια την πλήρη αμφισβήτηση των δεδομένων και των αποτελεσμάτων.</p>
275 ^η γραμμή	<p>Στην συγκεκριμένη γραμμή, οι μεταβλητές Xsum, Ysum, Zsum, CentX, CentY και CentZ παίρνουν την τιμή 0.</p> <p>Όπως προανέφερα, με αυτόν τον τρόπο γίνεται κατά μία έννοια <i>software reset</i> έτσι ώστε να μην περάσουν δεδομένα από προηγούμενες εκτελέσεις, το οποίο θα έχει ως συνέπεια την πλήρη αμφισβήτηση των δεδομένων και των αποτελεσμάτων.</p>
276 ^η γραμμή	<p>Στην συγκεκριμένη γραμμή, οι μεταβλητές CentLongRad, Hyp, CentLatRad, CentLat και CentLong παίρνουν την τιμή 0.</p> <p>Όπως προανέφερα, με αυτόν τον τρόπο γίνεται κατά μία έννοια <i>software reset</i> έτσι ώστε να μην περάσουν δεδομένα από προηγούμενες εκτελέσεις, το οποίο θα έχει ως συνέπεια την πλήρη αμφισβήτηση των δεδομένων και των αποτελεσμάτων.</p>

4.2.35 Πρόγραμμα λήψης και εικονικού τριγωνισμού – Διεργασία χρονικού διαχωρισμού περασμάτων

```

279 void lastPass()
280 {
281     newSignal = millis();
282     if(newSignal - lastSignal > interval)
283     {
284         newPass = 1;
285         Serial.print("New Pass");
286     }
287 }
288
  
```

279 ^η γραμμή	<p>Δημιουργώ και δηλώνω μία συνάρτηση τύπου void και όνομα lastPass χωρίς παραμέτρους.</p> <p>Η διεργασία δηλώνεται πως είναι <i>void</i>, δηλαδή δεν επιστρέφει δεδομένα μετά την εκτέλεση της, παρά μόνο εκτελεί τις εντολές που βρίσκονται εντός αυτής.</p> <p>Δημιούργησα την συγκεκριμένη διεργασία με σκοπό να διαχωρίσω τα μηνύματα διάσωσης που λαμβάνονται με χρονικά όρια, έτσι ώστε αν υπάρξει μεγάλο χρονικό κενό μεταξύ 2 λήψεων, να αποφευχθούν σφάλματα στον υπολογισμό του κέντρου βάρους και έτσι ώστε στο κυρίως πρόγραμμα, η κλήση της διεργασίας και η γραφή του προγράμματος να το καθιστούν ευανάγνωστο και λειτουργικό.</p>
281 ^η γραμμή	<p>Στην συγκεκριμένη γραμμή, με την εντολή newSignal = millis() αποθηκεύονται στην μεταβλητή newSignal τα δεδομένα που παρέχονται από την κλήση της διεργασίας millis. Η συγκεκριμένη διεργασία χρησιμοποιείται από την βιβλιοθήκη TinyGPS και χρησιμοποιείται για να δείχνει τον χρόνο που έχει περάσει από την αρχή του προγράμματος σε milliseconds.</p>
282 ^η γραμμή	<p>Στην συγκεκριμένη γραμμή, με την εντολή if ελέγχεται η συνθήκη που βρίσκεται μέσα στην παρένθεση. Με την συγκεκριμένη συνθήκη ελέγχω αν η διαφορά ανάμεσα στην μεταβλητή newSignal και στην μεταβλητή lastSignal είναι μεγαλύτερη από την τιμή της μεταβλητής interval.</p> <p>Αν ικανοποιείται αυτή η συνθήκη, τότε εκτελούνται οι εντολές εντός των ορίων της if.</p>
284 ^η γραμμή	<p>Στην συγκεκριμένη γραμμή, η συγκεκριμένη εντολή αλλάζει την τιμή της μεταβλητής newPass έτσι ώστε να ισούται με 1.</p> <p>Όπως έχω αναφέρει και κατά την επεξήγηση του κυρίως προγράμματος, η συγκεκριμένη μεταβλητή χρησιμοποιείται ως σημαία για να δηλώσει ένα νέο πέρασμα από την εμβέλεια του συστήματος εκπομπής του μηνύματος διάσωση.</p>

285 ^η γραμμή	<p>Με την εντολή Serial.print() δίνεται η εντολή στο πρόγραμμα να τυπώσει στην σειριακή θύρα του επεξεργαστή τα δεδομένα εντός της παρένθεσης σε μορφή κατανοητή από τον άνθρωπο. Στην συγκεκριμένη περίπτωση, η σειριακή θύρα που χρησιμοποιείται είναι η 0, αλλά δεν γράφεται ως Serial0.println σε αντίθεση με τις υπόλοιπες σειριακές θύρες (Serial1, Serial2, Serial3) που έχει ο συγκεκριμένος Arduino Mega. Η συγκεκριμένη εντολή, εκτυπώνει τους χαρακτήρες “New Pass” στην οθόνη του υπολογιστή που είναι συνδεδεμένος ο Arduino Mega για λόγους παρακολούθησης της σωστής λειτουργίας του προγράμματος.</p>
-------------------------	--

4.2.36 Πρόγραμμα λήψης και εικονικού τριγωνισμού – Διεργασία αποθήκευσης μέσω σημείων

```

289 void sdLoggingTEMP()
290 {
291     //Write the newest information to the SD Card
292     //Open the Data CSV File
293     File dataFile = SD.open("LOG.csv", FILE_WRITE);
294     if(dataFile)
295     {
296         dataFile.println("-----");
297         dataFile.print(SD_date_time);
298         dataFile.print(",");
299         dataFile.print(CentLat, 6);
300         dataFile.print(",");
301         dataFile.println(CentLong, 6);
302         dataFile.close();
303     }
304 }
305

```

289 ^η γραμμή	<p>Δημιουργώ και δηλώνω μία συνάρτηση τύπου void και όνομα sdLoggingTEMP χωρίς παραμέτρους.</p> <p>Η διεργασία δηλώνεται πως είναι <i>void</i>, δηλαδή δεν επιστρέφει δεδομένα μετά την εκτέλεση της, παρά μόνο εκτελεί τις εντολές που βρίσκονται εντός αυτής.</p> <p>Δημιούργησα την συγκεκριμένη διεργασία με σκοπό να εκτελεί ένα σύνολο εντολών για την αποθήκευση των μέσω γεωγραφικών συντεταγμένων που προκύπτουν από τους υπολογισμούς σε ένα αρχείο csv (comma separated values) στην εξωτερική κάρτα μνήμης microSD και έτσι ώστε στο κυρίως πρόγραμμα, η κλήση της διεργασίας και η γραφή του προγράμματος να το καθιστούν ευανάγνωστο και λειτουργικό.</p>
293 ^η γραμμή	<p>Στην συγκεκριμένη γραμμή, η εντολή File dataFile = SD.open("LOG.csv", FILE_WRITE) ανοίγει από την εξωτερική κάρτα μνήμης το αρχείο LOG τύπου csv με σκοπό την εγγραφή δεδομένων στο τελευταίο σημείο του αρχείου. Το αρχείο που ανοίγεται και η σκοπός του, δηλαδή η εγγραφή σε αυτό, αποθηκεύονται στην</p>

	μεταβλητή <code>dataFile</code> για την βέλτιστη γραφή των υπόλοιπων εντολών. Αυτό είναι απαιτούμενο από την βιβλιοθήκη <code>SD</code> .
294 ^η γραμμή	Στην συγκεκριμένη γραμμή, με την εντολή if ελέγχεται η συνθήκη που βρίσκεται μέσα στην παρένθεση. Με την συγκεκριμένη συνθήκη ελέγχω αν το αρχείο άνοιξε κανονικά, δηλαδή αν υπάρχει μέσα στην κάρτα και αν είναι εφικτή η εγγραφή σε αυτό. Αν ικανοποιείται αυτή η συνθήκη, τότε εκτελούνται οι εντολές εντός των ορίων της if .
296 ^η γραμμή έως 301 ^η γραμμή	Με την εντολή dataFile.println() δίνεται η εντολή στο πρόγραμμα να τυπώσει, στο αρχείο που έχει ανοιχτεί από την εξωτερική κάρτα μνήμης, τα δεδομένα εντός της παρένθεσης σε μορφή κατανοητή από τον άνθρωπο. Σε αντίθεση με την εντολή dataFile.print() όπου μετά το τέλος την εκτύπωσης των δεδομένων ο κέρσορας παραμένει στο ίδιο σημείο, η προσθήκη ln στο τέλος της εντολής <code>print</code> , ξεκινάει μία νέα γραμμή μετά την εκτύπωση των δεδομένων. Το σύνολο των εντολών dataFile.println() και dataFile.print() που ακολουθούν είναι γραμμένο με τέτοιο τρόπο ώστε να γίνεται κατανοητό το πρόγραμμα από τρίτο πρόσωπο. Τα δεδομένα που εκτυπώνονται είναι τα εξής: <ul style="list-style-type: none"> • Ένα σύνολο οριζοντίων γραμμών για να γίνεται εύκολος οπτικός διαχωρισμός από τον χρήστη • Ημερομηνία και ώρα που αποθηκεύτηκαν αρχικά τα δεδομένα • Κόμμα για τον διαχωρισμό των δεδομένων • Το γεωγραφικό πλάτος του μέσου σημείου με ακρίβεια 6 δεκαδικών • Κόμμα για τον διαχωρισμό των δεδομένων • Το γεωγραφικό μήκος του μέσου σημείου με ακρίβεια 6 δεκαδικών
302 ^η γραμμή	Στην συγκεκριμένη γραμμή, η εντολή dataFile.close() εκτελεί την αποθήκευση του αρχείου που έχω ανοίξει παραπάνω και την αποδέσμευσή του. Με αυτόν τον τρόπο, τα δεδομένα βρίσκονται ασφαλή στην εξωτερική κάρτα μνήμης <code>microSD</code> και μπορούν να χρησιμοποιηθούν σε υπολογιστή για επεξεργασία τους και αποτύπωσή τους σε χάρτη.

4.2.37 Πρόγραμμα λήψης και εικονικού τριγωνισμού – Διεργασία αποθήκευσης νέων σημείων

```

306 void midPointLog()
307 {
308     File dataFile = SD.open("MID.csv", FILE_WRITE);
309     if(dataFile)
310     {
311         dataFile.print(Nu);
312         dataFile.print("New point ");
313         dataFile.println(SD_date_time);
314         dataFile.print(Lat[Nu], 6);
315         dataFile.print(",");
316         dataFile.println(Long[Nu], 6);
317         dataFile.close();
318     }
319 }
  
```

306 ^η γραμμή	<p>Δημιουργώ και δηλώνω μία συνάρτηση τύπου void και όνομα midPointLog χωρίς παραμέτρους.</p> <p>Η διεργασία δηλώνεται πως είναι <i>void</i>, δηλαδή δεν επιστρέφει δεδομένα μετά την εκτέλεση της, παρά μόνο εκτελεί τις εντολές που βρίσκονται εντός αυτής.</p> <p>Δημιούργησα την συγκεκριμένη διεργασία με σκοπό να εκτελεί ένα σύνολο εντολών για την αποθήκευση των νέων γεωγραφικών συντεταγμένων που προκύπτουν από τους υπολογισμούς σε ένα αρχείο csv (comma separated values) στην εξωτερική κάρτα μνήμης microSD και έτσι ώστε στο κυρίως πρόγραμμα, η κλήση της διεργασίας και η γραφή του προγράμματος να το καθιστούν ευανάγνωστο και λειτουργικό.</p>
308 ^η γραμμή	<p>Στην συγκεκριμένη γραμμή, η εντολή File dataFile = SD.open("MID.csv", FILE_WRITE) ανοίγει από την εξωτερική κάρτα μνήμης το αρχείο MID τύπου csv με σκοπό την εγγραφή δεδομένων στο τελευταίο σημείο του αρχείου. Το αρχείο που ανοίγεται και η σκοπός του, δηλαδή η εγγραφή σε αυτό, αποθηκεύονται στην μεταβλητή <i>dataFile</i> για την βέλτιστη γραφή των υπόλοιπων εντολών. Αυτό είναι απαιτούμενο από την βιβλιοθήκη SD.</p>
309 ^η γραμμή	<p>Στην συγκεκριμένη γραμμή, με την εντολή if ελέγχεται η συνθήκη που βρίσκεται μέσα στην παρένθεση. Με την συγκεκριμένη συνθήκη ελέγχω αν το αρχείο άνοιξε κανονικά, δηλαδή αν υπάρχει μέσα στην κάρτα και αν είναι εφικτή η εγγραφή σε αυτό.</p> <p>Αν ικανοποιείται αυτή η συνθήκη, τότε εκτελούνται οι εντολές εντός των ορίων της if.</p>
311 ^η γραμμή έως 316 ^η γραμμή	<p>Με την εντολή dataFile.println() δίνεται η εντολή στο πρόγραμμα να τυπώσει, στο αρχείο που έχει ανοιχτεί από την εξωτερική κάρτα μνήμης, τα δεδομένα εντός της παρένθεσης σε μορφή κατανοητή από τον άνθρωπο.</p>

	<p>Σε αντίθεση με την εντολή dataFile.print() όπου μετά το τέλος την εκτύπωσης των δεδομένων ο κέρσορας παραμένει στο ίδιο σημείο, η προσθήκη \n στο τέλος της εντολής print, ξεκινάει μία νέα γραμμή μετά την εκτύπωση των δεδομένων.</p> <p>Το σύνολο των εντολών dataFile.println() και dataFile.print() που ακολουθούν είναι γραμμένο με τέτοιο τρόπο ώστε να γίνεται κατανοητό το πρόγραμμα από τρίτο πρόσωπο. Τα δεδομένα που εκτυπώνονται είναι τα εξής:</p> <ul style="list-style-type: none"> • Την τιμή της μεταβλητής Nu • Τους χαρακτήρες «New point » για τον εύκολο οπτικό διαχωρισμό των δεδομένων • Ημερομηνία και ώρα ελήφθησαν τα συγκεκριμένα δεδομένα • Την τιμή της θέσης του πίνακα LatRad[] που δείχνει η μεταβλητή Nu, με ακρίβεια 6 δεκαδικών ψηφίων • Κόμμα για τον διαχωρισμό των δεδομένων • Την τιμή της θέσης του πίνακα LongRad[] που δείχνει η μεταβλητή Nu, με ακρίβεια 6 δεκαδικών ψηφίων
317 ^η γραμμή	<p>Στην συγκεκριμένη γραμμή, η εντολή dataFile.close() εκτελεί την αποθήκευση του αρχείου που έχω ανοίξει παραπάνω και την αποδέσμευσή του. Με αυτόν τον τρόπο, τα δεδομένα βρίσκονται ασφαλή στην εξωτερική κάρτα μνήμης microSD και μπορούν να χρησιμοποιηθούν σε υπολογιστή για επεξεργασία τους και αποτύπωσή τους σε χάρτη.</p>

4.3. ΠΡΟΓΡΑΜΜΑ ΣΥΣΤΗΜΑΤΟΣ ΕΚΠΟΜΠΗΣ ΜΗΝΥΜΑΤΟΣ ΔΙΑΣΩΣΗΣ

Το σύστημα εκπομπής ραδιοσήματος είναι απλής μορφής με κωδικοποίηση.

Για λόγους χαμηλότερου κοστολογίου, έχει χρησιμοποιηθεί μόνο ο επεξεργαστής του Arduino, ο Atmel ATMEGA328P-PU, συνδεδεσολογημένος σε ράστερ για την επιτυχή λειτουργία του.

Τροφοδοτείται από μπαταρία 11.1V, ενώ έχει προστεθεί ένας ρυθμιστής τάσης LM7805 για να παρέχεται η σωστή τάση των 5 Volt στον επεξεργαστή και τα περιφερειακά.

Για την αποστολή του ραδιοσήματος που περιέχει το κωδικοποιημένο μήνυμα διάσωσης, χρησιμοποιήθηκε ο πομπός του ζεύγους με συχνότητα 433MHz.

4.3.1 Πρόγραμμα εκπομπής μηνύματος διάσωσης – Προσθήκη βιβλιοθηκών & Δήλωση σταθερών και μεταβλητών

```

1 #include <VirtualWire.h>
2
3 const int led_pin = 8;
4 const int transmit_pin = 9;
5 const char msg[4] = {'S', 'O', 'S'};

```

1 ^η γραμμή	Συμπεριλαμβάνω την βιβλιοθήκη VirtualWire η οποία είναι απαραίτητη για την δημιουργία του εικονικού καναλιού επικοινωνίας και την διαχείριση της αποστολής των πληροφοριών μέσω του πομπού
3 ^η γραμμή	Δημιουργώ μία σταθερά led_pin = 8, γιατί στον συγκεκριμένο ακροδέκτη του επεξεργαστή έχω συνδέσει ένα LED
4 ^η γραμμή	Δημιουργώ μία σταθερά transmit_pin = 9, γιατί στον συγκεκριμένο ακροδέκτη του επεξεργαστή έχω συνδέσει την είσοδο του πομπού και από εκεί θα γίνεται η αποστολή προς τον πομπό και εν συνεχεία η μετάδοση
5 ^η γραμμή	<p>Δημιουργώ έναν πίνακα 4 χαρακτήρων. Ο χαρακτήρας τέλους πάντα πρέπει να υπολογίζεται επιπλέον του συνολικού μεγέθους του πίνακα.</p> <p>Στην συγκεκριμένη περίπτωση το μήνυμα που θέλω να μεταδώσω, πρέπει να έχει τα εξής χαρακτηριστικά :</p> <ul style="list-style-type: none"> • Να είναι σύντομο • Να είναι ευδιάκριτο στο ανθρώπινο μάτι μέσα στο πρόγραμμα • Να είναι συγκεκριμένο για να μπορεί να αναγνωρίζεται και να ελέγχεται εύκολα

4.3.2 Πρόγραμμα εκπομπής μηνύματος διάσωσης – Βασική λειτουργία [void setup()]

```

9 void setup()
10 {
11   pinMode(led_pin, OUTPUT);
12   // Initialise the IO and ISR
13   vw_set_tx_pin(transmit_pin);
14   vw_setup(2000);           // Bits per sec
15 }

```

11 ^η γραμμή	Ο ακροδέκτης που δηλώθηκε στην αρχή του προγράμματος για το LED, εδώ δηλώνεται στον επεξεργαστή ότι θα είναι έξοδος
13 ^η γραμμή	Δηλώνεται στην βιβλιοθήκη VirtualWire ότι ο ακροδέκτης που θα χρησιμοποιηθεί για την μετάδοση, θα είναι η τιμή της σταθεράς transmit_pin που δηλώθηκε στην αρχή του προγράμματος. Σε αυτόν τον ακροδέκτη θα συνδεθεί η είσοδος του πομπού.
14 ^η γραμμή	Δηλώνεται στην βιβλιοθήκη VirtualWire ότι ο ρυθμός μετάδοσης θα είναι 2000bps. Αυτό δηλώνεται και στο πρόγραμμα του συστήματος λήψης. Αν γίνει λάθος, δεν θα μπορεί να αναγνωσθεί το μήνυμα διάσωσης.

4.3.3 Πρόγραμμα εκπομπής μηνύματος διάσωσης – Κυκλική λειτουργία [void loop()]

```

19 void loop()
20 {
21   digitalWrite(led_pin, HIGH); // Flash a light to show transmitting
22   vw_send((uint8_t *)msg, 4);
23   vw_wait_tx(); // Wait until the whole message is gone
24   digitalWrite(led_pin, LOW);
25   delay(500);
26 }

```

21 ^η γραμμή	Στέλνω σήμα στον ακροδέκτη που βρίσκεται συνδεδεμένο το LED για να περάσουν 5V και να ανάψει. Με αυτόν τον τρόπο, έχω μία απλή, οπτική επιβεβαίωση ότι έχει ξεκινήσει η διαδικασία εκπομπής του μηνύματος διάσωσης κανονικά.
22 ^η γραμμή	Με την συγκεκριμένη εντολή, δηλώνεται στην βιβλιοθήκη VirtualWire ότι ο τύπος του μηνύματος που θα μεταδοθεί είναι unsigned character 8bit, το μήνυμα που θα μεταδοθεί και το μέγεθός του, συνολικά 4 Bytes.
23 ^η γραμμή	Λόγω αγνώστου χρόνου μετάδοσης, το πρόγραμμα περιμένει μέχρι να μεταδοθεί ολόκληρο το μήνυμα.

24 ^η γραμμή	Στέλνω σήμα στον ακροδέκτη που βρίσκεται συνδεδεμένο το LED για να σταματήσει η τροφοδοσία (να επιστρέψει στα 0V) και να σβήσει. Με αυτόν τον τρόπο, έχω μία απλή, οπτική επιβεβαίωση ότι έχει πλέον τελειώσει η διαδικασία εκπομπής του μηνύματος διάσωσης κανονικά.
25 ^η γραμμή	Σε αυτό το σημείο, το πρόγραμμα “σπαταλάει” χρόνο. Η επιλογή του μισού δευτερολέπτου έγινε μετά από μετρήσεις ως ένας ικανοποιητικός χρόνος αναμονής και διαχωρισμού του ραδιοσήματος μετάδοσης, αλλά παράλληλα όχι πολύ μεγάλος για να μην απομακρυνθεί ο δέκτης.

ΣΥΜΠΕΡΑΣΜΑΤΑ – ΠΑΡΑΤΗΡΗΣΕΙΣ

Στην πτυχιακή αυτή εργασία ανέπτυξα έναν συνδυασμό συστημάτων και συνένωσα αντικείμενα φαινομενικά ασύνδετα μεταξύ τους με σκοπό την δημιουργία ενός αποτελέσματος σχετικά οικονομικού, λειτουργικού και με δυνατότητες αναβάθμισης αλλά και μεταφοράς σε άλλες πλατφόρμες.

Τα υλικά που χρησιμοποίησα για τον πομπό ήταν τα πιο οικονομικά, τον οποίο σχεδίασα και κατασκεύασα με γνώμονα την χρήση σε πραγματικές συνθήκες, έτσι ώστε σε ενδεχόμενη δημιουργία ενός προϊόντος που θα καταλήξει στην αγορά, να είναι μικρό, οικονομικά προσιτό, αλλά και μη ενεργοβόρο για την αύξηση την διάρκειας συνεχούς λειτουργίας τους.

Στο κομμάτι του προγραμματισμού του συστήματος εκπομπής του μηνύματος διάσωσης, το πρόγραμμα είναι λιτό και συγκεκριμένο. Σκοπός μου και εδώ, ήταν η αύξηση της διάρκειας συνεχούς λειτουργίας και η εκπομπή ενός ραδιοσήματος που θα είναι πλήρες, αλλά και σύντομο. Σε συνθήκες διάσωσης, όπως για παράδειγμα ενός χαμένου ορειβάτη, η αυτονομία θα πρέπει να υπερβαίνει τις ώρες, ίσως να φτάνει και κάποιες ημέρες.

Το σύνολο των ηλεκτρονικών που χρησιμοποίησα για τον δέκτη, είχε και αυτό γνώμονα την απλοϊκότητα και την χαμηλή τιμή, ενώ η αυτονομία της μπαταρίας είναι χαμηλότερης σημασίας, καθώς το σκάφος το οποίο φέρει το σύστημα, έχει την δυνατότητα να μεταφέρει ικανοποιητικού όγκου και βάρους φορτία, τα οποία είναι περισσότερο από ικανοποιητικά για την μεγάλη αυτονομία του συστήματος λήψης του μηνύματος διάσωσης.

Στο κομμάτι του προγραμματισμού του συστήματος λήψης, έγραψα το πρόγραμμα με γνώμονα την αντιμετώπιση των ενδεχόμενων προβλημάτων που μπορεί να παρουσιαστούν κατά την αναζήτηση του συστήματος εκπομπής του μηνύματος διάσωσης. Είναι προγραμματισμένο να αποφεύγει την σύγκυση από την λήψη διαφορετικών σημάτων, που θα καταστήσουν τα αποτελέσματα εσφαλμένα. Επίσης, τα δεδομένα αποθηκεύονται σε σταθερό σημείο, έτσι ώστε να μην χάνονται σε περίπτωση οποιασδήποτε δυσλειτουργίας.

Ο εγκέφαλος – αυτόματος πιλότος που επιλέχθηκε ήταν ο πλέον βέλτιστος, καθώς υποστηρίζεται με αναβαθμίσεις, αλλά και εγχειρίδια για τον προγραμματισμό του. Οι δυνατότητές του χρησιμοποιήθηκαν στο έπακρο, αλλά όχι στο μέγιστο τους. Μπορεί να προσφέρει πολύ περισσότερα σε πιθανή εξέλιξη του συστήματος, καθιστώντας την επιλογή του την πλέον βέλτιστη για μελλοντική αναβάθμιση.

Το όχημα το επέλεξα μετά από μεγάλη αναζήτηση, καθώς διαπίστωσα ότι τα διάφορα οχήματα που θα μπορούσαν φιλοξενήσουν το όλο σύστημα, είτε ήταν πολύ ακριβά, είτε πολύ μικρά, είτε με αφιλόξενο σχεδιασμό για την χρήση που το ήθελα. Το συγκεκριμένο όχημα, μαζί με τα περιφερειακά του το καθιστούν και αυτό μία επιλογή με δυνατότητα εξέλιξης πάνω σε αυτό, καθώς είναι όχημα μικρής κατανάλωσης, μεγάλης δυνατότητας φόρτωσης και δυνατότητα υπερπήδησης μεγάλων εμποδίων, αναλογούντων των διαστάσεών του.

Σε πολλά σημεία, οι εμπειρίες από τα μαθήματα της σχολής, αλλά και οι διάφορες προσωπικές συζητήσεις που είχαμε με συναδέλφους και καθηγητές, στάθηκαν μεγίστης σημασίας για την επίλυση

προβλημάτων που παρουσιάστηκαν, και ειδικά σε αυτά που δεν ήταν εμφανή σε άπειρο και άσχετο άνθρωπο με το αντικείμενο.

Εν κατακλείδι, οι εμπειρίες που αποκόμισα και τα συμπεράσματα που εξήγαγα από την πραγματοποίηση της πτυχιακής εργασίας, με έκαναν να θέλω την περαιτέρω βελτίωση της συγκεκριμένης εργασίας και την συνεργασία με τους κατάλληλους ανθρώπους που θα με βοηθήσουν να την τελειοποιήσουμε και να παράξουμε ένα προϊόν το οποίο θα χρησιμοποιηθεί παγκόσμια για την έρευνα και διάσωση.

ΕΞΕΛΙΞΗ

Οι προοπτικές εξέλιξης του συστήματος είναι πολλές, καθώς το σύστημα σχεδιάστηκε και υλοποιήθηκε με σκοπό την αυτονομία του κάθε κομματιού. Το πιο σημαντικό βήμα είναι η αγορά και η ελάχιστη μετατροπή και στην συνέχεια η συντήρηση ενός μη επανδρωμένου ιπτάμενου οχήματος (UAV), το οποίο θα εκτινάξει τις δυνατότητες του συστήματος, ειδικά με την ταχύτητα και την έκταση κάλυψης, καθώς και την υπέρβαση των ανωμάτων και δυσπρόσβατων εδαφών.

Ενδεχόμενη χρηματοδότηση του όλου εγχειρήματος, θα παρέχει την δυνατότητα αγοράς ακριβότερων υλικών ή και σχεδιασμού νέων, που θα προσφέρουν ακρίβεια, απόσταση, κάλυψη, αυτονομία και τυποποίηση. Για παράδειγμα, κεραίες εκπομπής και λήψης με κάλυψη εκατοντάδων μέτρων ή και χιλιομέτρων, οχήματα για ταυτόχρονη αναζήτηση και πιθανότατα εγκεφάλους που θα ενσωματώνουν όλα τα ηλεκτρονικά.

ΚΟΣΤΟΛΟΓΙΟ

Επίγειο όχημα	60€
Ηλεκτρονικά οχήματος	80€
Εγκέφαλος Ardupilot	140€
Περιφερειακά Ardupilot	90€
Arduino	50€
Περιφερειακά Arduino	60€
Έξοδα από αναπλήρωση των παραπάνω υλικών	75€
Λοιπά έξοδα	50€
ΣΥΝΟΛΟ	605€

Λοιπά έξοδα όπως για τις μετακινήσεις για πραγματοποίηση των δοκιμών, την αναβάθμιση του εργαστηριακού εξοπλισμού, των ηλεκτρονικών υπολογιστών και άλλων, δεν μπορούν να υπολογιστούν με ακρίβεια, αλλά θα πρέπει να προϋπολογιστούν παρόμοια κόστη σε ενδεχόμενη εξέλιξη.

ΕΥΧΑΡΙΣΤΙΕΣ

Ευχαριστώ τον Δρ. Αλέξανδρο Αστάρα για την αρχική ανάληψη της εποπτείας την πτυχιακής μου εργασίας και την παροχή υλικών για την πραγματοποίησή της.

Ευχαριστώ τον κ. Μιχάλη Κιζήρογλου για την τελική εποπτεία και την καθοδήγησή του.

Ευχαριστώ όλους τους καθηγητές του τμήματος που μου παρείχαν έμμεσα ή άμεσα την γνώση για την επίλυση των προβλημάτων και των δυσκολιών που ανέκυψαν κατά της ανάπτυξη της πτυχιακής εργασίας.

ΒΙΒΛΙΟΓΡΑΦΙΑ – ΠΗΓΕΣ

- www.arduino.cc – Arduino Website
- www.adafruit.com – Adafruit Industries Website
- en.wikipedia.org – Wikipedia
- el.wikipedia.org – Βικιπαίδεια
- www.geomidpoint.com – Υπολογιστής γεωγραφικών αποστάσεων
- www.ti.com – Texas Instruments
- www.nap.edu – National Academies Press
- Autonomous Vehicles in Support of Naval Operations (2005)
- www.hobbyking.com – Online hobby shop
- Εξώφυλλο – www.nap.edu/read/11379/chapter/8
- Σχήμα 3.1 – www.hobbytronics.co.uk
- Σχήμα 3.2 – www.martarduino.com.br
- Σχήμα 4.2 – www.hobbyking.com
- Σχήμα 4.3 – www.hobbyking.com
- Σχήμα 4.4 – www.hobbyking.com
- Σχήμα 4.5 – www.ardupilot.org