ΠΟΛΥΤΕΧΝΕΙΟ ΚΡΗΤΗΣ
ΣΧΟΛΗ ΜΗΧΑΝΙΚΩΝ ΠΑΡΑΓΩΓΗΣ ΚΑΙ ΔΙΟΙΚΗΣΗΣ
ΣΥΣΤΗΜΑΤΑ ΠΑΡΑΓΩΓΗΣ

Αυτοματοποιημένη καταγραφή και γραφική αναπαράσταση σε πραγματικό χρόνο αέριων ρύπων με χρήση μη-επανδρωμένων ιπτάμενων συσκευών (drones).

Κωνσταντίνος Λουκάκης

Επιβλέπων καθηγητής : Παπαευθυμίου Σπυρίδων
Επιτροπή : Πουλιέζος Αναστάσιος
Καλαϊτζάκης Κωνσταντίνος

Χανιά 2017

Περίληψη

Σκοπός της παρούσας εργασίας είναι η δημιουργία ενός αυτόματου ιπτάμενου συστήματος το οποίο θα σαρώνει μια προκαθορισμένη περιοχή και θα δίνει αναλυτικές πληροφορίες σχετικά με τους αέριους ρύπους και τις συνθήκες πίεσης, θερμοκρασίας και υγρασίας που επικρατούν στην περιοχή αυτή. Ο χρήστης θα επιλέγει από κατάλληλο λογισμικό την περιοχή που θέλει να μελετήσει και στη συνέχεια το αεροσκάφος θα σαρώνει αυτόματα την περιοχή και θα αποστέλλει τις μετρήσεις. Οι πληροφορίες αυτές θα αποστέλλονται ασύρματα και σε επίγεια βάση δεδομένων. Ο χρήστης θα μπορεί να αντλεί τα δεδομένα από τη βάση και να τα μελετάει σύμφωνα με τις ανάγκες τις έρευνάς του. Επίσης θα υπάρχει η δυνατότητα για τον χρήστη να παρακολουθεί μέσω διαδικτύου την λεπτομερή καταγραφή των μετρήσεων πάνω στον χάρτη σε πραγματικό χρόνο ή να κάνει ανασκόπηση παλαιότερων καταγραφών.

Σαν ιπτάμενο μέσο χρησιμοποιήθηκε το “Phantom 4” drone της DJI (www.dji.com), ενώ οι αισθητήρες και τα ηλεκτρονικά εξαρτήματα είναι της εταιρίας Libelium. (www.libelium.com). Για την απεικόνιση των δεδομένων χρησιμοποιήθηκε το διαδικτυακό γραφικό περιβάλλον της GpsGate (www.gpsgate.com) , αφού πρώτα παραμετροποιήθηκε και προσαρμόστηκε σύμφωνα με τις ανάγκες της εργασίας. Για την εγκατάσταση των αισθητήρων στο αεροσκάφος, χρειάστηκε εφαρμογή αντίστροφης μηχανικής στο ηλεκτρονικό σύστημα αισθητήρων, όπως επίσης και στο ίδιο το αεροσκάφος. Τα σχέδια της βάσης στήριξης έγιναν με χρήση του λογισμικού CAD “Pro/ENGINEER” και τα τελικά εξαρτήματα τυπώθηκαν από τον 3D-Printer “Duplicator I3 Plus” της εταιρίας WANHAO.

Περιεχόμενα

1. Ηλεκτρονικό σύστημα

1.1. Μη-επανδρωμένο ιπτάμενο όχημα (Drone)

Στην παρούσα εργασία χρησιμοποιήθηκε σαν ιπτάμενο μέσο το Phantom 4 (εικόνα 1-1) της DJI, τα τεχνικά χαρακτηριστικά του οποίου παρουσιάζονται στον πίνακα 1-1. Οι βασικοί λόγοι που καθιστούν το συγκεκριμένο αεροσκάφος ιδανικό για την παρούσα εργασία είναι δύο. Ο πρώτος λόγος είναι η εξαιρετικά σταθερή πτητική ικανότητά του, η οποία επιτρέπει την προσθήκη του επιπλέον βάρους του ηλεκτρονικού συστήματος, διατηρώντας τον έλεγχό του εξίσου εύκολο για τον χειριστή. Ο δεύτερος λόγος είναι η πληθώρα εφαρμογών που έχουν σχεδιαστεί για το συγκεκριμένο αεροσκάφος από ανεξάρτητες εταιρίες. Ο χρήστης λοιπόν μπορεί να χρησιμοποιήσει εφαρμογές, είτε της DJI είτε κάποιας όλη εταιρίας, και να προγραμματίσει το αεροσκάφος ώστε να ακολουθήσει ένα προεπιλεγμένο σχέδιο πτήσης. Με αυτόν τον τρόπο, μπορεί να αυτοματοποιηθεί έως και 100% η καταγραφή των μετρήσεων. Πιο συγκεκριμένα το αεροσκάφος μπορεί να προγραμματιστεί έτσι ώστε να απογειωθεί μόνο του, να σαρώσει μία προκαθορισμένη περιοχή και να προσγειωθεί στο σημείο από το οποίο απογειώθηκε.

Εικόνα 1-1. Το Phantom 4 μαζί με το τηλεχειριστήριο του, το οποίο μεταφέρει την εικόνα στο tablet.

(Πηγή: http://hiconsumption.com/2017/04/dji-phantom-4-advanced-drone)

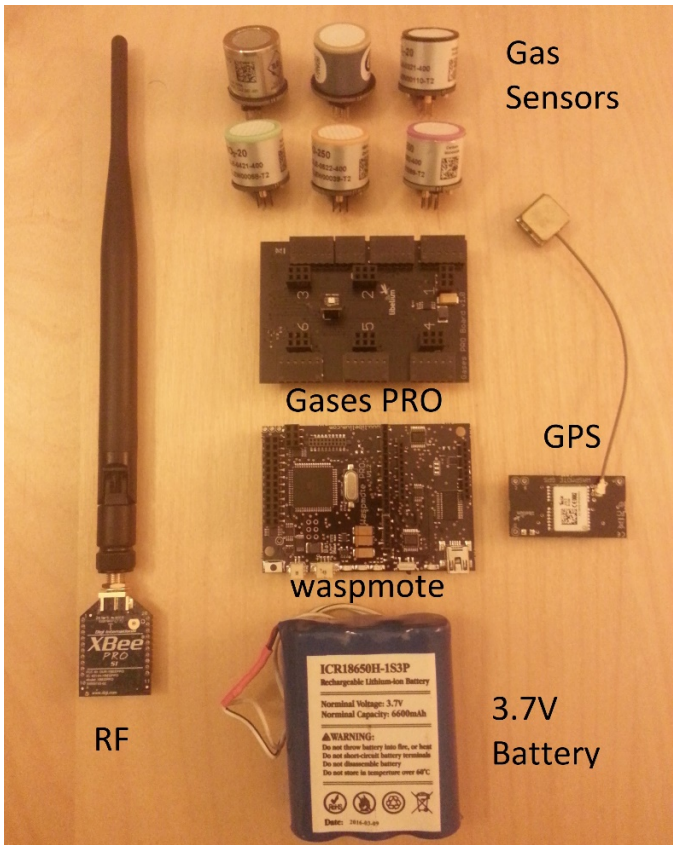
AIRCRAFT	
Weight (Including Battery)	1380 g
Max Ascent Speed	6 m/s (Sport mode)
Max Descent Speed	4 m/s (Sport mode)
Max Speed	20 m/s (Sport mode)
Max Service Ceiling Above Sea Level	19685 feet (6000 m)
Max Flight Time	Approx. 28 minutes
Operating Temperature Range	32° to 104° F (0° to 40° C)
Satellite Systems	GPS / GLONASS
OBSTACLE SENSING SYSTEM	
Obstacle Sensory Range	2 - 49 feet (0.7 - 15 m)
Operating Environment	Surface with clear pattern and adequate lighting (lux > 15)
VISION POSITIONING SYSTEM	
Velocity Range	≤10 m/s (2 m above ground)
Altitude Range	0 - 33 feet (0 - 10 m)
Operating Range	0 - 33 feet (0 - 10 m)
Operating Environment	Surfaces with a clear pattern and adequate lighting (lux > 15)
REMOTE CONTROLLER	
Operating Frequency	2.400 GHz to 2.483 GHz
Max Transmission Distance	FCC Compliant: 3.1 mi (5 km);
CE Compliant: 2.2 mi (3.5 km)(Unobstructed, free of interference)	
Operating Temperature	32° to 104° F (0° to 40° C)
Battery	6000 mAh LiPo 2S
Transmitter Power (EIRP)	FCC: 23 dBm; CE: 17 dBm
Operating Voltage	7.4V @ 1.2A
INTELLIGENT FLIGHT BATTERY	
(PH4 - 5350 MAH -15.2 V)	
Capacity	5350 mAh
Voltage	15.2 V
Battery Type	LiPo 4S
Energy	81.3 Wh
Net Weight	462 g
Operating Temperature	14° to 104° F (-10° to 40° C)
Max Charging Power	100 W

Πίνακας 1-1. Τεχνικά χαρακτηριστικά του Phantom 4.

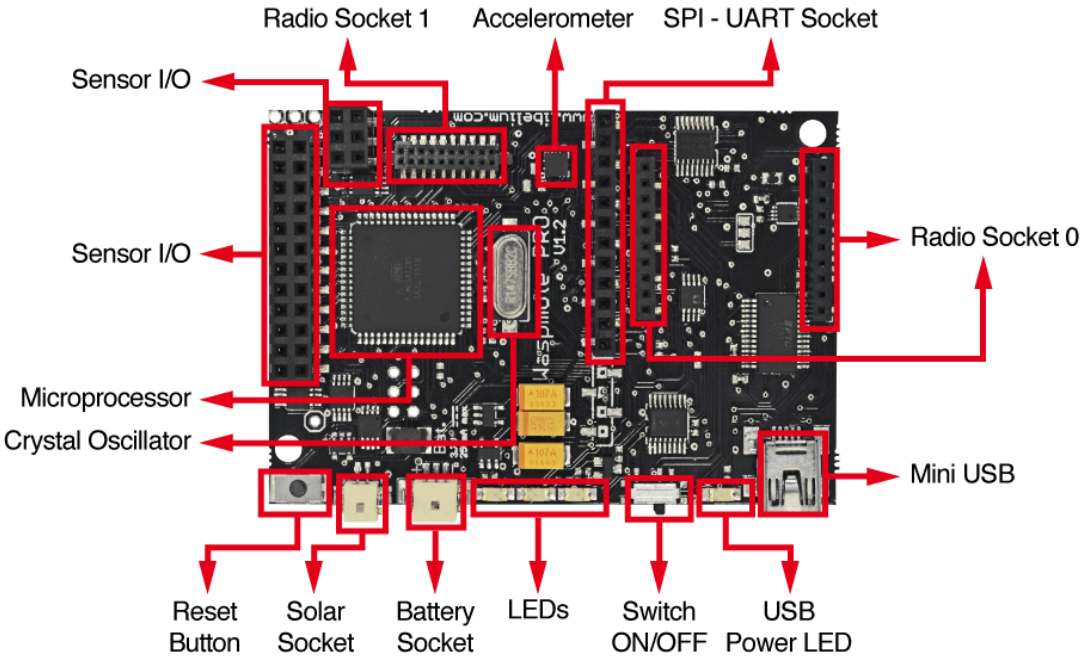
(Πηγή: http://www.drone-world.com/dji-phantom-4-specs)

1.2. Σύστημα αισθητήρων

Το σύστημα αισθητήρων (εικόνα 1-2) που χρησιμοποιήθηκε στην παρούσα εργασία προμηθεύτηκε από την εταιρία "Libelium". Σαν βάση χρησιμοποιήθηκε το "waspmote" (εικόνα 1-3), όπως το έχει ονομάσει η εταιρία, το οποίο μπορεί να θεωρηθεί σαν την "μητρική κάρτα" του συστήματος. Ο μικροελεγκτής που χρησιμοποιείται είναι ο ATmega1281 και προγραμματίζεται σε γλώσσα C. Οι αισθητήρες τοποθετούνται σε μια επιπρόσθετη πλακέτα, η οποία ονομάζεται "Gases PRO". Για την βέλτιστη ακρίβεια των μετρήσεων, όλοι οι αισθητήρες που χρησιμοποιήθηκαν προμηθεύτηκαν βαθμονομημένοι από το εργοστάσιο. Για τον προσδιορισμό των γεωγραφικών συντεταγμένων η Libelium χρησιμοποιεί τον GPS δέκτη JN3 της εταιρίας Telit, ο οποίος τοποθετείται επίσης στο waspmote. Η ασύρματη μεταφορά των δεδομένων πραγματοποιείται με τον πομποδέκτη "XBee PRO", ο οποίος λειτουργεί στα 2.4GHz και έχει εμβέλεια 1500m. Τα δεδομένα συλλέγονται από το "Meshlium" (εικόνα 1-4) και αποθηκεύονται εσωτερικά σε αυτό, σε βάση δεδομένων SQL, η οποία είναι προσβάσιμη και μέσο του διαδικτύου. Το waspmote τροφοδοτείται από μπαταρία λιθίου χωρητικότητας 6600mAh και τάσης 3.7V. Στη συνέχεια παρουσιάζονται κάποια βασικά χαρακτηριστικά των αισθητήρων.



Εικόνα 1-2. Το ηλεκτρονικό σύστημα των αισθητήρων.



Εικόνα 1-3. Η ανατομία του Waspmote

(Πηγή: http://www.libelium.com/products/waspmote/hardware/)

Εικόνα 1-4. Meshlium

(Πηγή: http://amicus.com.sg/shop/waspmote/wireless-waspmote/meshlium-scanner-3g-ap/)

Carbon Monoxide (CO)

Sensor: CO-A4
Nominal Range: 0 to 25 ppm
Maximum Overload: 2000 ppm
Long Term Sensitivity Drift: < 10% change/year in lab air, monthly test
Long Term zero Drift: < ±100 ppb equivalent change/year in lab air
Response Time (T90): ≤ 20 seconds
Sensitivity: 220 to 375 nA/ppm
Accuracy: as good as ±0.1 ppm (ideal conditions)
H2S filter capacity: 250000 ppm·hrs

Carbon Dioxide (CO_2)

Sensor: INE20-CO2P-NCVSP
Nominal Range: 0 to 5000 ppm
Long Term Output Drift: < ± 250 ppm/year
Warm up time: 60 seconds @ 25 ºC
At least 30 min for full specification @ 25 °C
Response Time (T90): ≤ 60 seconds
Resolution: 25 ppm
Accuracy: as good as ±50 ppm, from 0 to 2500 ppm range (ideal conditions)
as good as ±200 ppm, from 2500 to 5000 ppm range (ideal conditions)

Ozone (O_3)

Sensor: OX-A431
Nominal Range: 0 to 18 ppm
Maximum Overload: 50 ppm
Long Term sensitivity Drift: -20 to -40% change/year
Response Time (T90): ≤ 45 seconds
Sensitivity: -200 to -550 nA/ppm
Accuracy: as good as ±0.2 ppm (ideal conditions)

Nitric Oxide (NO)

Sensor: 4-NO-250
Nominal Range: 0 to 250 ppm
Maximum Overload: 1000 ppm
Long Term Output Drift: < 2% signal/month
Response Time (T90): ≤ 30 seconds
Sensitivity: 400 ± 80 nA/ppm
Accuracy: as good as ±0.5 ppm (ideal conditions)

Nitric Dioxide (NO_2)

Sensor: 4-NO2-20
Nominal Range: 0 to 20 ppm
Maximum Overload: 250 ppm
Long Term Output Drift: < 2% signal/month
Response Time (T90): ≤ 30 seconds
Sensitivity: 600 ± 150 nA/ppm
Accuracy: as good as ±0.2 ppm (ideal conditions)

Sulfur Dioxide (SO_2)

Sensor: 4-SO2-20
Nominal Range: 0 to 20 ppm
Maximum Overload: 150 ppm
Long Term Output Drift: < 2% signal/month
Response Time (T90): ≤ 45 seconds
Sensitivity: 500 ± 150 nA/ppm
Accuracy: as good as ±0.2 ppm (ideal conditions)

Temperature, Humidity and Pressure sensor (BME280)

Temperature sensor:
Operational range: -40 ~ +85 ºC
Full accuracy range: 0 ~ +65 ºC
Accuracy: ±1 ºC (range 0 ºC ~ +65 ºC)
Response time: 1.65 seconds (63% response from +30 to +125 °C).
Typical consumption: 1 µA measuring

Humidity sensor:
Measurement range: 0 ~ 100% of Relative Humidity (for temperatures < 0 °C and > 60 °C see figure below)
Accuracy: < ±3% RH (at 25 ºC, range 20 ~ 80%)
Hysteresis: ±1% RH
Operating temperature: -40 ~ +85 ºC
Response time (63% of step 90% to 0% or 0% to 90%): 1 second
Typical consumption: 1.8 µA measuring
Maximum consumption: 2.8 µA measuring

Pressure sensor
Measurement range: 30 ~ 110 kPa
Operational temperature range: -40 ~ +85 ºC
Full accuracy temperature range: 0 ~ +65 ºC
Absolute accuracy: ±0.1 kPa (0 ~ 65 ºC)
Typical consumption: 2.8 µA measuring
Maximum consumption: 4.2 µA measuring

(Πηγή: Libelium)

2. Εγκατάσταση αισθητήρων

2.1. Κριτήρια σχεδίασης της βάσης

Η βάση στήριξης των αισθητήρων σχεδιάστηκε με τα παρακάτω κριτήρια:

- Τοποθέτηση των αισθητήρων εκτός της ροής αέρα των ελίκων.

Οι έλικες του αεροσκάφους δημιουργούν μια πολύ έντονη κάθετη ροή αέρα, η οποία μπορεί να γίνει αντιληπτή σε απόσταση μεγαλύτερη των 10 μέτρων κάτω από το αεροσκάφος, επηρεάζοντας έτσι τις μετρήσεις των αισθητήρων. Για την μελέτη της επίδρασης της ροής στον περιβάλλον χώρο, τοποθετήθηκαν μικρές κορδέλες σε διάφορα σημεία γύρω από το αεροσκάφος. Παρατηρήθηκε λοιπόν ότι η ροή είναι ιδιαίτερα κάθετη και δεν επηρεάζει την περιοχή περιμετρικά του αεροσκάφους, ακόμα και σε μικρή απόσταση από αυτό.

- Συμμετρία

Η βάση θα πρέπει να είναι κατά το δυνατό συμμετρική, ώστε να επηρεάζεται το ελάχιστο από τον φαινόμενο άνεμο. Η πρωτότυπη βάση που παρουσιάζεται στις εικόνες 2-1 και 2-2 δεν είναι συμμετρική. Αυτό έχει σαν αποτέλεσμα να δημιουργείται τάση περιστροφής του αεροσκάφους σε περίπτωση που υπάρχει πλευρικός άνεμος.

Εικόνα 2-1. Πλάγια όψη πρωτότυπης βάσης.

Εικόνα 2-2. Πρόσοψη πρωτότυπης βάσης.

- Ελάχιστο βάρος.

Είναι σημαντικό να μειωθεί στο ελάχιστο το επιπρόσθετο βάρος του αεροσκάφους. Η αύξηση του βάρους συνεπάγεται αυξημένη κατανάλωση ενέργειας, δηλαδή μείωση της διάρκειας πτήσης. Επίσης, με την αύξηση του βάρους δυσχεραίνεται η δυνατότητα ελιγμών του αεροσκάφους (π.χ. επιτάχυνση - επιβράδυνση, αλλαγή πορείας).

- Κέντρο βάρους

Το κέντρο βάρους του αεροσκάφους θα πρέπει να διατηρηθεί στο κέντρο του αεροσκάφους και χαμηλότερα από το κέντρο περιστροφής του (roll center). Σε περίπτωση που το κέντρο βάρους μετατοπισθεί κατά το μήκος ή το πλάτος του αεροσκάφους, οι κινητήρες λειτουργούν δυσανάλογα μεταξύ τους, αυξάνοντας την κατανάλωση και μειώνοντας την πτητική ικανότητα του αεροσκάφους. Το κέντρο περιστροφής είναι το σημείο εκείνο, γύρω από το οποίο περιστρέφεται το αεροσκάφος όταν παίρνει κλίση. Εάν το κέντρο βάρους μετατοπισθεί ψηλότερα από το κέντρο περιστροφής, δημιουργείται η τάση να γυρίσει ανάποδα το αεροσκάφος. Στην περίπτωση αυτή, όπως αποδείχτηκε από δεύτερη πρωτότυπη βάση που κατασκευάστηκε, το αεροσκάφος γίνεται εξαιρετικά ασταθές και ανίκανο για πτήση.

- Αποφυγή χρήσης μαγνητικών υλικών

Η χρήση μαγνητικών υλικών επηρεάζει τις ψηφιακές πυξίδες του αεροσκάφους. Στην βάση των εικόνων 2-1 και 2-2 το μεταλλικό τμήμα που χρησιμοποιήθηκε σαν προέκταση, δημιούργησε τόσο μεγάλη παρεμβολή στις πυξίδες, που το λογισμικό του αεροσκάφους απαγόρευσε την εκκίνηση των κινητήρων. Για τον λόγο αυτό η μεταλλική προέκταση αντικαταστάθηκε με ξύλινη.

- Εφαρμογή στο αεροσκάφος

Η βάση πρέπει να είναι ανθεκτική και σταθερή, ενώ δεν θα πρέπει να θέτει σε κίνδυνο για θραύση τα τμήματα του αεροσκάφους. Επίσης, θα πρέπει να μπορεί να εγκαθίσταται και να απεγκαθίσταται εύκολα στο αεροσκάφος για λόγους φορητότητας.

- Αισθητήρες του αεροσκάφους

Το αεροσκάφος διαθέτει αισθητήρες απόστασης στο μπροστινό και κάτω μέρος του για την αποφυγή τυχόν εμποδίων. Η βάση πρέπει να είναι αόρατη σε αυτούς. Σε διαφορετική περίπτωση, όπως συνέβη με την διορθωμένη βάση των εικόνων 2-1 και 2-2, το αεροσκάφος θεωρεί το ηλεκτρονικό σύστημα σαν εμπόδιο και επηρεάζεται σε σημαντικό βαθμό η δυνατότητα ελέγχου. Το αεροσκάφος επίσης διαθέτει περιστρεφόμενη κάμερα υφηλής ανάλυσης, της οποίας το οπτικό πεδίο θα πρέπει να είναι καθαρό.

2.2. Σχεδίαση βάσης στήριξης

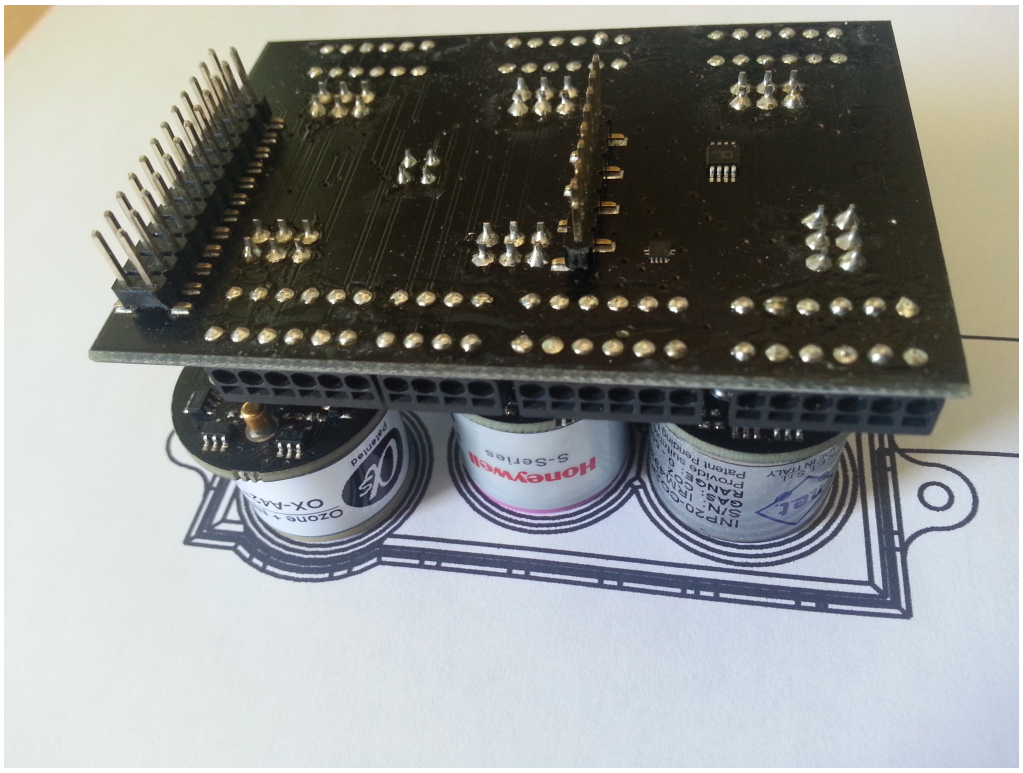
Για τους λόγους που προαναφέρθηκαν ζυγίσθηκαν όλα τα εξαρτήματα. Παρατηρήθηκε ότι το σύστημα των αισθητήρων ζυγίζει σχεδόν όσο η μπαταρία λιθίου (εικόνα 2-3). Για τον λόγο αυτό, η μπαταρία απομονώθηκε από το υπόλοιπο σύστημα και τοποθετήθηκε στην τελική βάση αντιδιαμετρικά. Προκειμένου να μείνουν ανεπηρέαστοι οι αισθητήρες του αεροσκάφους, αλλά και η κάμερα, η μπαταρία εγκαταστάθηκε στην αριστερή πλευρά και το υπόλοιπο σύστημα στα δεξιά του αεροσκάφους. Όλα τα σχέδια έγιναν στο λογισμικό Pro/ENGINEER.

Εικόνα 2-3. Ζύγισμα του συστήματος

2.2.1. Κουτί αισθητήρων και μπαταρίας

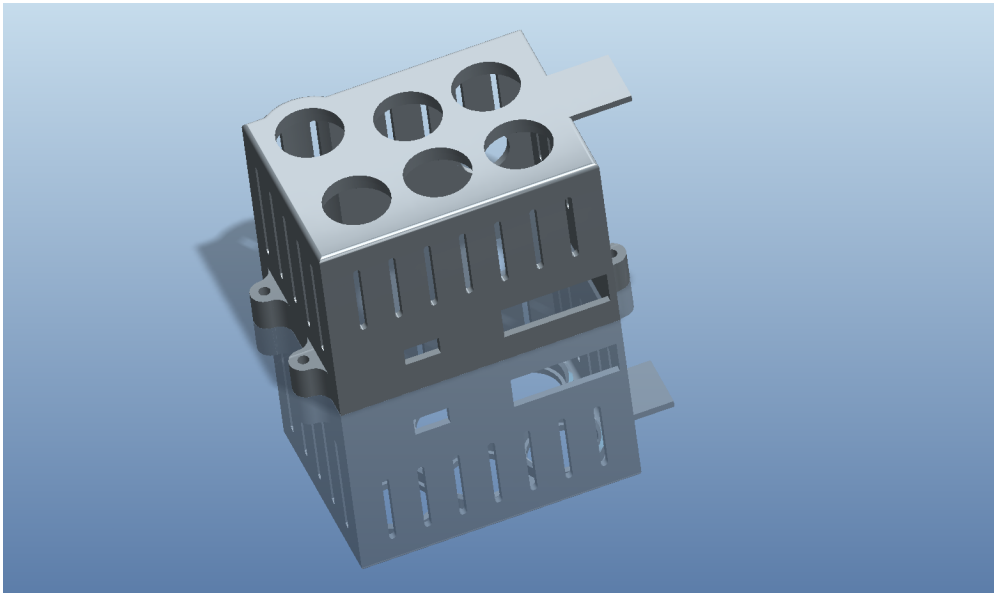
Για την ασφάλεια των αισθητήρων κυρίως, αλλά και των υπόλοιπων εξαρτημάτων, χρειάστηκε να τοποθετηθούν εντός κουτιού. Η Libelium τοποθετεί του αισθητήρες πάνω στην κάρτα "Gases PRO", χωρίς ωστόσο να διασφαλίζει την παραμονή τους σε αυτή και τους αφήνει ελεύθερους να αποσυνδεθούν. Το αεροσκάφος είναι ένα κινούμενο μέσο που μπορεί να υποστεί έντονες επιταχύνσεις προς όλες τις κατευθύνσεις, ενώ ταυτόχρονα οι κινητήρες του δημιουργούν δονήσεις. Στην βάση των εικόνων 2-1 και 2-2 οι αισθητήρες κινδυνεύουν να αποσυνδεθούν κατά τη διάρκεια της πτήσης.

Για τον λόγο αυτό, έγινε αντίστροφη μηχανική στο σύστημα των αισθητήρων, προκειμένου να καταγραφούν οι ακριβείς διαστάσεις του. Οι μετρήσεις έγιναν με παχύμετρο και οι διαστάσεις εξακριβώθηκαν με διαδοχικές εκτυπώσεις σε χαρτί Α4 (εικόνα 2-4).

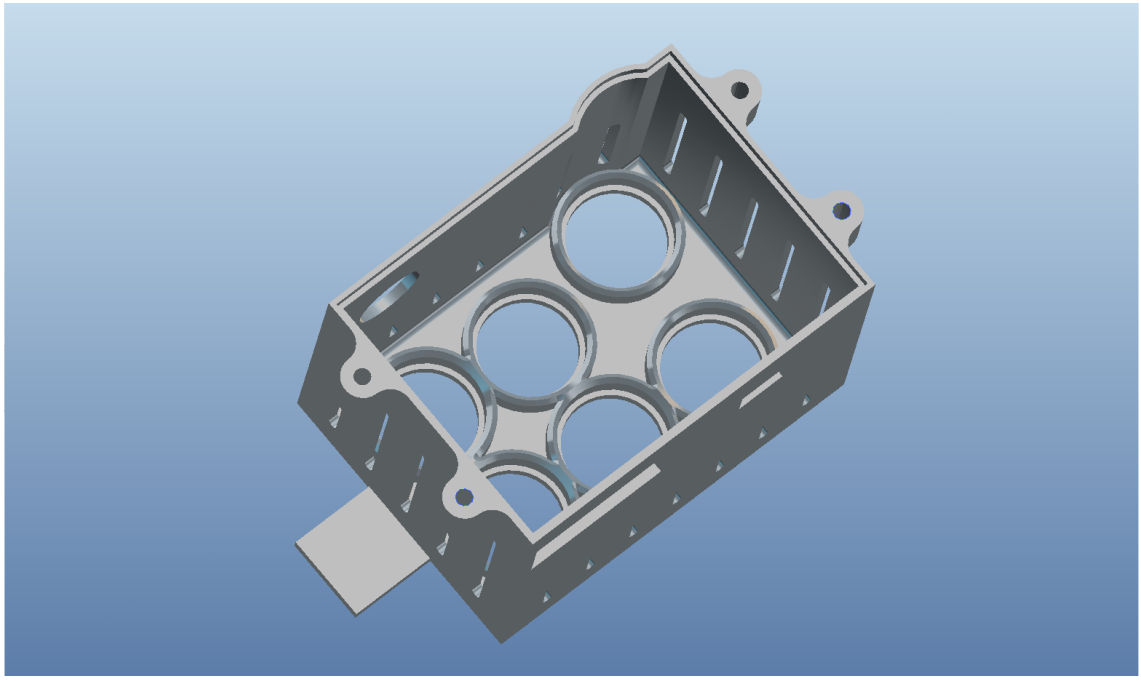


Εικόνα 2-4. Επαλήθευση των διαστάσεων των αισθητήρων

Το κουτί των αισθητήρων (εικόνα 2-5) σταθεροποιεί τον κάθε αισθητήρα ξεχωριστά με κυκλικό τοίχωμα (εικόνα 2-6) γύρω από αυτόν και απαγορεύει οποιαδήποτε κίνηση. Επίσης, τα τοιχώματα του κουτιού έχουν οπές οι οποίες εξασφαλίζουν ότι εντός και εκτός του κουτιού επικρατούν οι ίδιες συνθήκες (π.χ. πίεσης, υγρασίας, θερμοκρασίας).

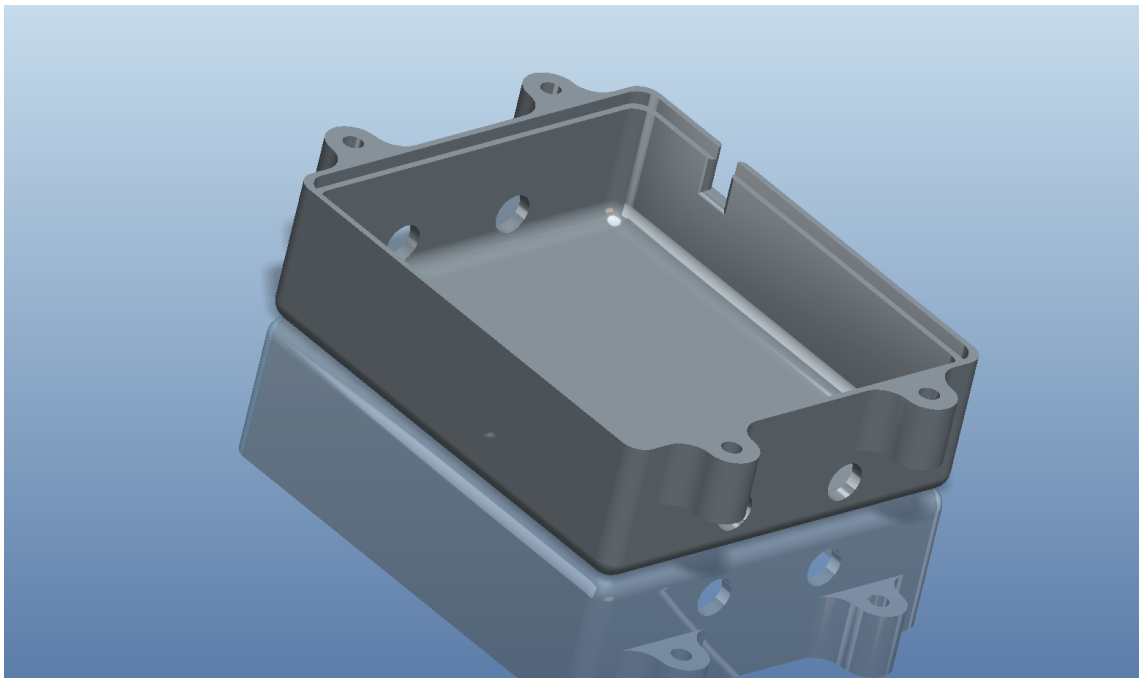


Εικόνα 2-5. 3D Μοντέλο του κουτιού των αισθητήρων.

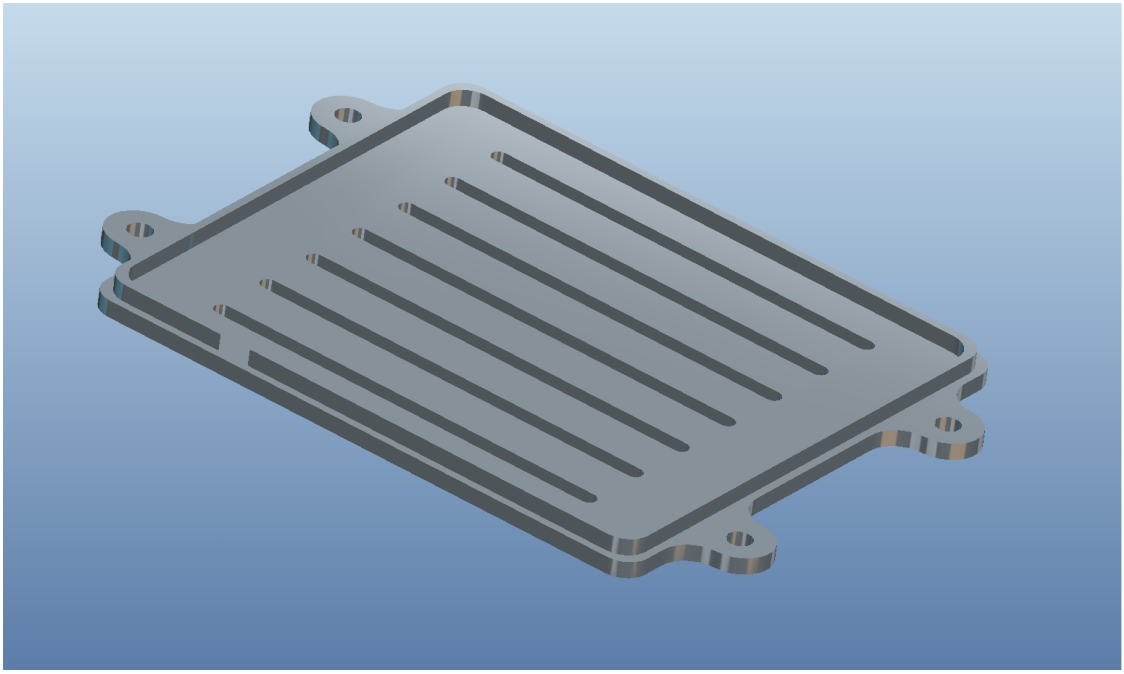


Εικόνα 2-6. Τα τοιχώματα συγκράτησης των αισθητήρων.

Για την συγκράτηση της μπαταρίας, κατασκευάστηκε με όμοιο τρόπο ένα δεύτερο κουτί (εικόνες 2-7 και 2-8).



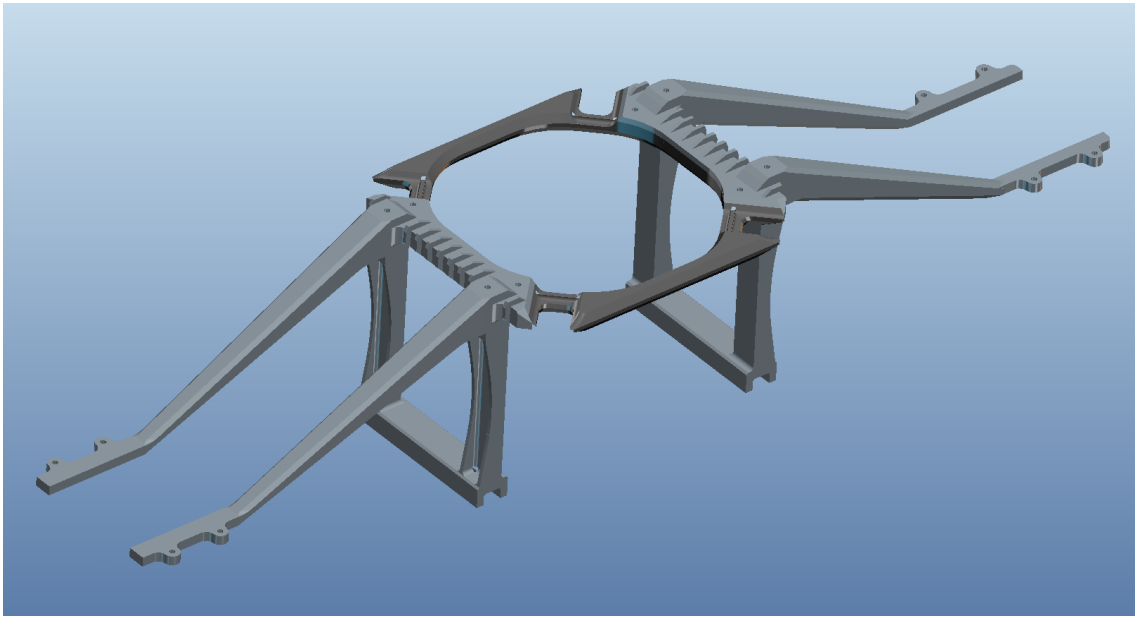
Εικόνα 2-7. Κουτί μπαταρίας (άνω τμήμα).



Εικόνα 2-8. Κουτί μπαταρίας(κάτω τμήμα)

2.2.2. Βάση στήριξης

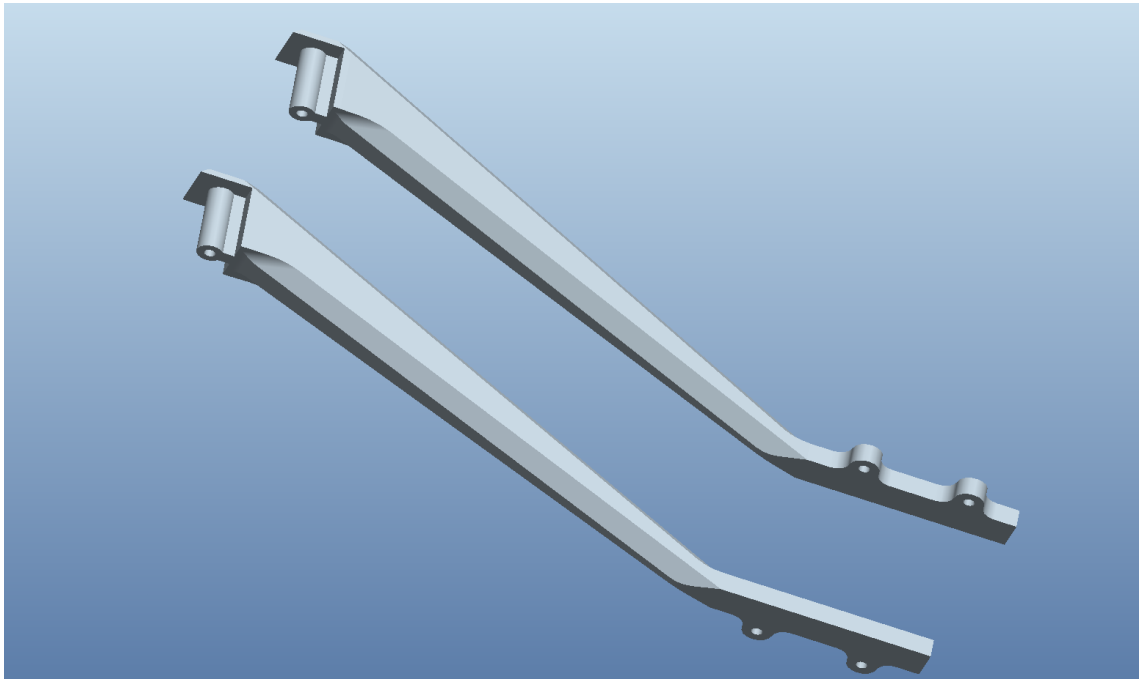
Στην παρακάτω εικόνα (εικόνα 2-9) παρουσιάζεται συναρμολογημένη η βάση, στην οποία εδράζουν το κουτί των αισθητήρων και το κουτί της μπαταρίας. Όλα τα τμήματα της κατασκευής συνδέονται εφαρμοστά μεταξύ τους, αλλά συγκρατούνται και με κοχλίες για λόγους ασφαλείας. Στη συνέχεια παρουσιάζονται τα τμήματα που την αποτελούν.



Εικόνα 2-9. Η βάση συναρμολογημένη.

Προεκτάσεις κουτιού αισθητήρων και μπαταρίας

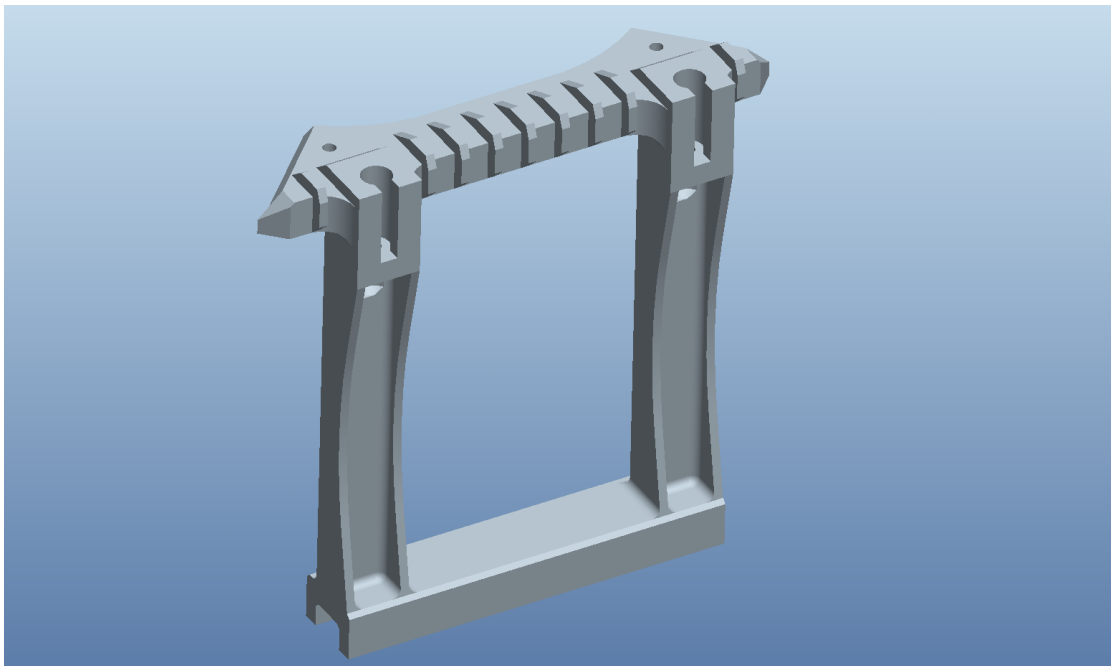
Οι προεκτάσεις του κουτιού των αισθητήρων (εικόνα 2-10) και του κουτιού της μπαταρίας έχουν σαν σκοπό να απομακρύνουν τους αισθητήρες από το κάθετο ρεύμα αέρα που δημιουργούν οι έλικες. Το κουτί της μπαταρία πρέπει να απομακρυνθεί ίδια απόσταση και αντιδιαμετρικά, για τους λόγους κατανομής του βάρους που αναλύθηκαν προηγουμένως.



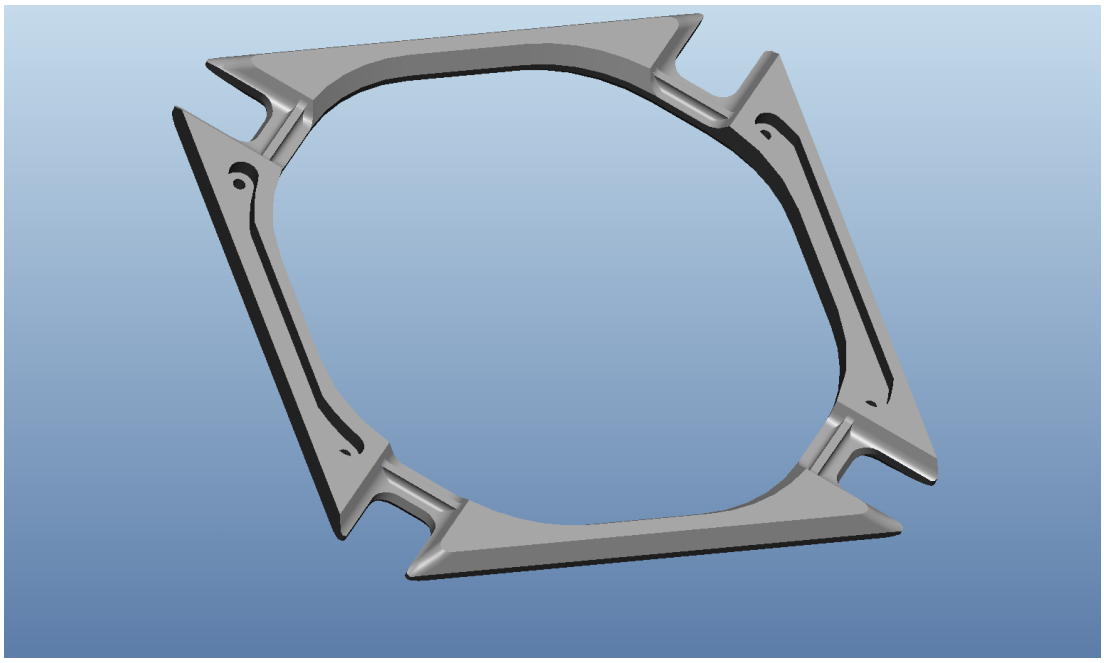
Εικόνα 2-10. Προεκτάσεις του κουτιού των αισθητήρων.

Κεντρικό τμήμα της βάσης

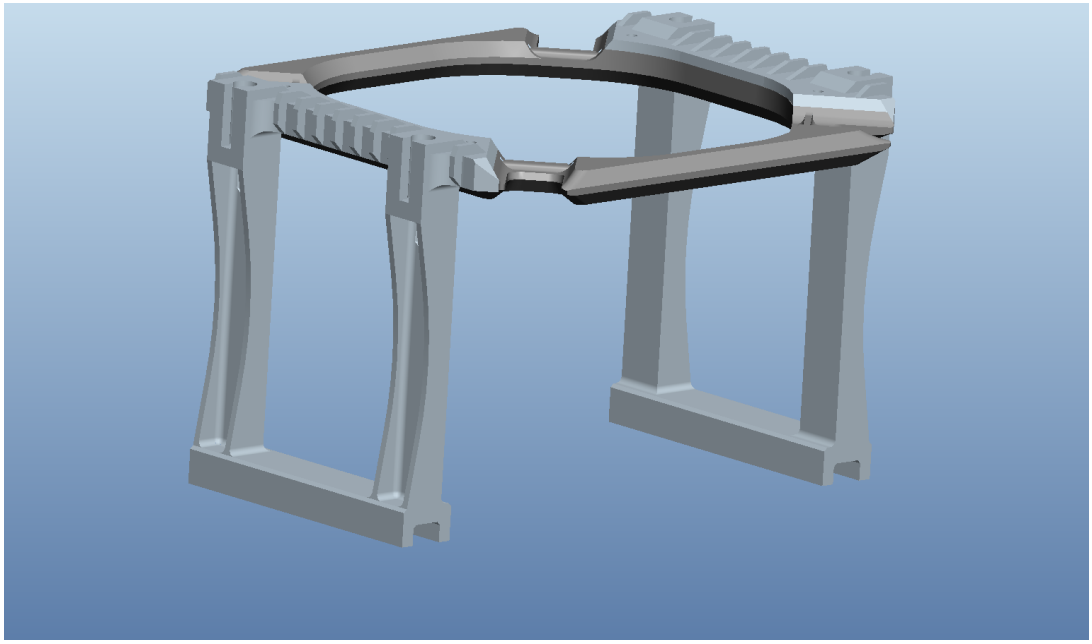
Το κεντρικό τμήμα της βάσης αποτελείται από δύο ζευγάρια δοκών (εικόνα 2-11), οι οποίες είναι ενωμένες μεταξύ τους με μια στεφάνη (εικόνα 2-12). Η στεφάνη αυτή απαγορεύει στην βάση να περιστραφεί ή να μετακινηθεί προς τα πάνω. Το ζευγάρι δοκών απαγορεύει επίσης στην βάση να περιστραφεί, αλλά και να μετακινηθεί προς τα κάτω. Οι προεκτάσεις εφαρμόζουν στο ζευγάρι δοκών, όπως φαίνεται στην εικόνα 2-13.



Εικόνα 2-11. Ζευγάρι δοκών στήριξης.

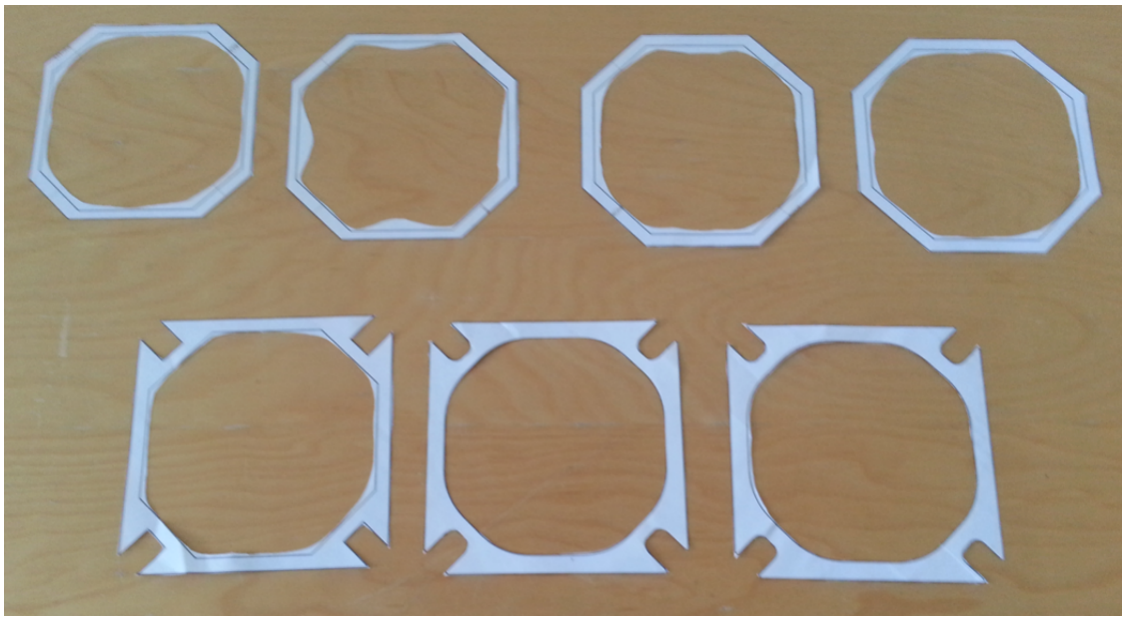


Εικόνα 2-12. Η στεφάνη που ενώνει τα δυο ζευγάρια δοκών .



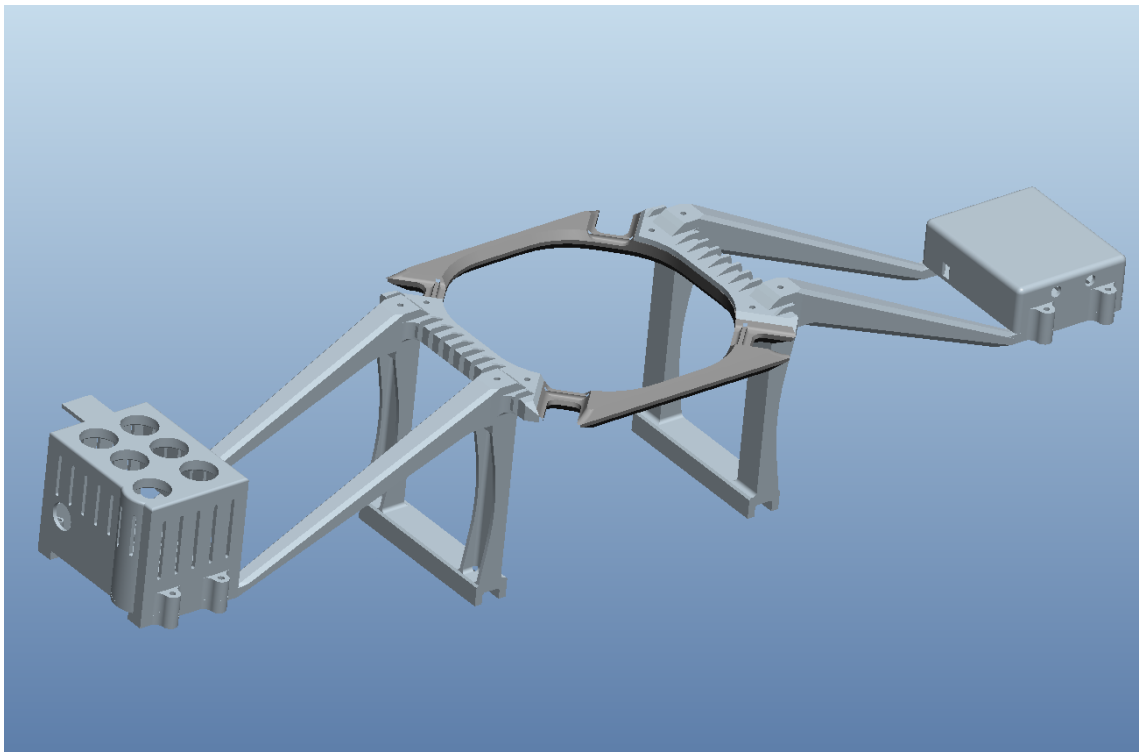
Εικόνα 2-13. Η στεφάνη μαζί με τα ζευγάρια δοκών.

Για την σχεδίαση της στεφάνης, ακολουθήθηκε η ίδια μεθοδολογία όπως και στο κουτί των αισθητήρων. Η μεγαλύτερη δυσκολία κατά την σχεδίαση, οφείλεται στο γεγονός ότι το αεροσκάφος είναι καμπύλο σε όλα του τα σημεία, καθιστώντας το συνεπώς δύσκολο παράδειγμα για εφαρμογή αντίστροφης μηχανικής. Επειδή η στεφάνη είναι πάρα πολύ σημαντικό τμήμα για την εφαρμογή της βάση στο αεροσκάφος, εκτός από τις εκτυπώσεις σε χαρτί A4 (εικόνα 2-14), έγινε και τρισδιάστατη εκτύπωση (εικόνα 2-15) για την επαλήθευση των διαστάσεων.

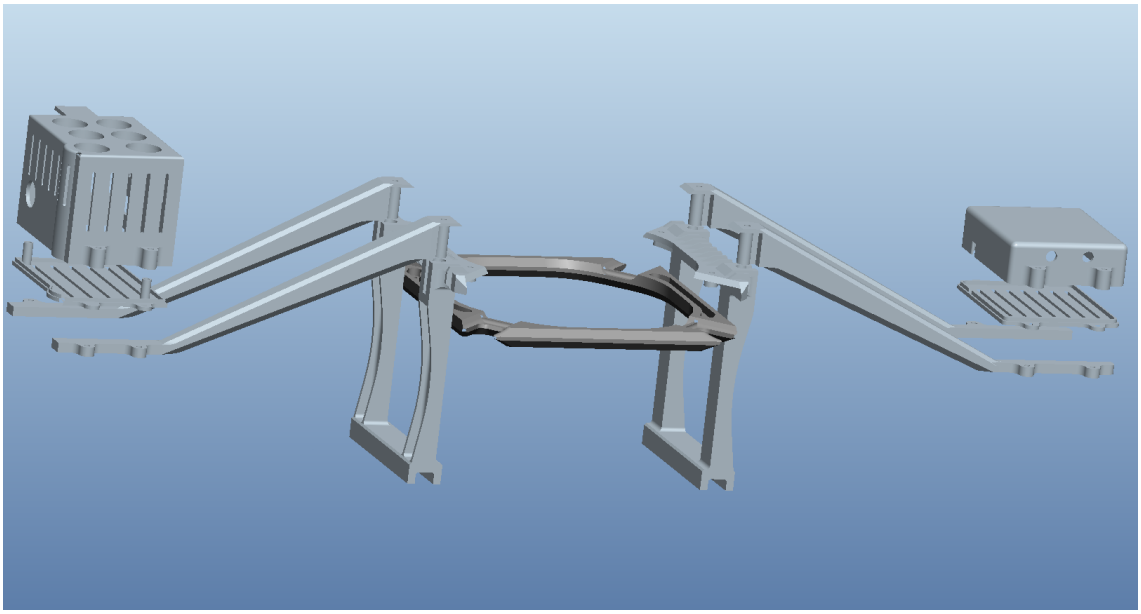


Εικόνα. 2-14. Δοκιμαστικές εκτυπώσεις σε χαρτί.

Εικόνα 2-15. Δοκιμαστική τρισδιάστατη εκτύπωση



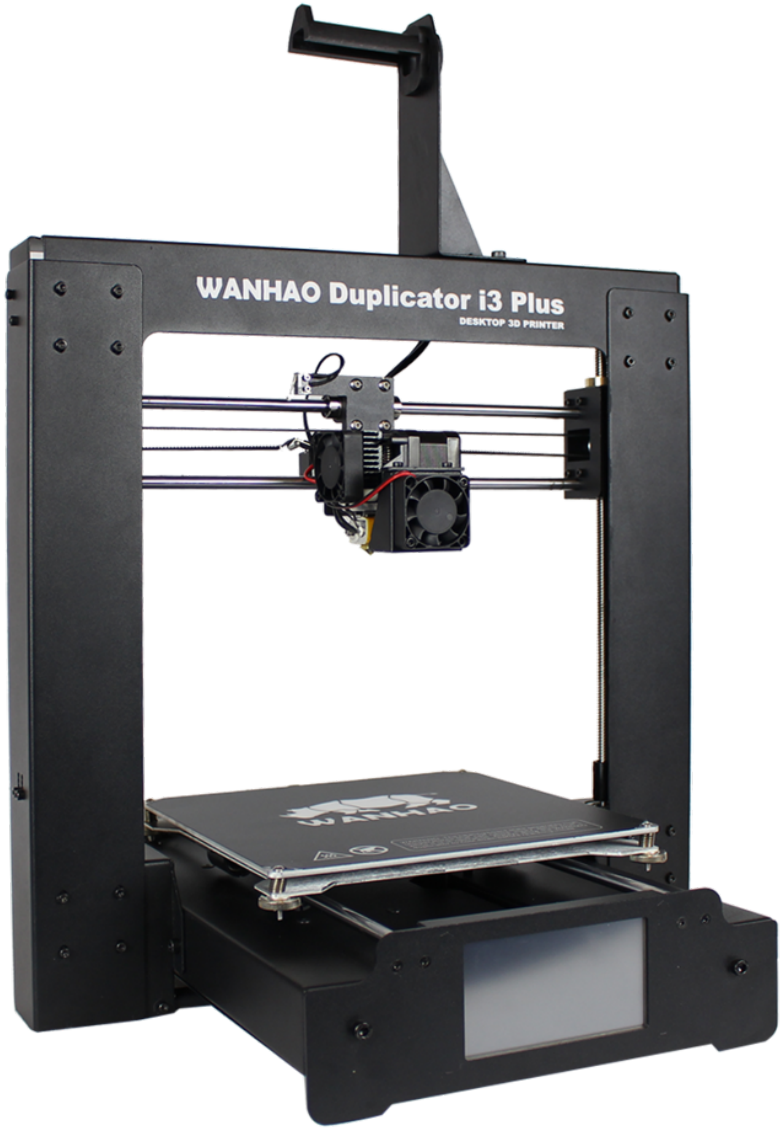
Εικόνα 2-16. Όλα τα τμήματα συναρμολογημένα.



Εικόνα 2-17. Όλα τα τμήματα σε ανεπτυγμένη μορφή.

2.3. 3D-Εκτύπωση

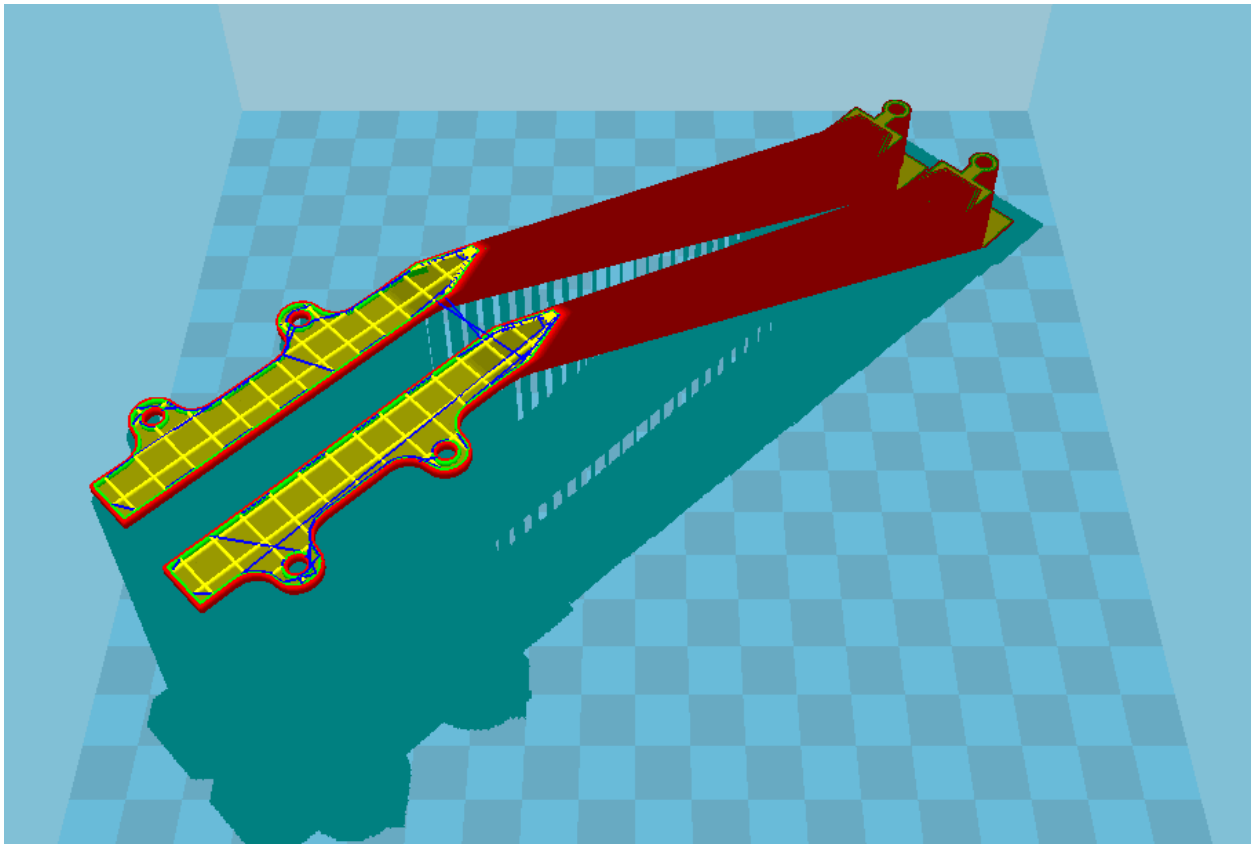
Όπως αναφέρθηκε προηγουμένως, όλα τα τμήματα της βάσης σχεδιάστηκαν στο λογισμικό Pro/ENGINEER. Από το λογισμικό αυτό δημιουργήθηκαν αρχεία τύπου "stl", ώστε να εισαχθούν στο λογισμικό "WANHAO-Cura". Στο WANHAO-Cura ρυθμίστηκαν όλες οι παράμετροι ώστε να δημιουργηθεί ο τελικός G-code που θα εισαχθεί στον τρισδιάστατο εκτυπωτή. Ο G-code είναι ο κώδικας τον οποίο θα εκτελέσει ο εκτυπωτής προκειμένου να δημιουργήσει το τελικό αντικείμενο. Ο εκτυπωτής που χρησιμοποιήθηκε στην παρούσα εργασία ήταν ο WANHAO Duplicator i3 Plus (εικόνα 2-18).



Εικόνα 2-18. Το 713ο επίπεδο εκτύπωσης των προεκτάσεων για το κουτί των αισθητήρων.

(Πηγή: https://www.3dprima.com)

Τα τελικά αντικείμενα κατασκευάστηκαν από υλικό PLA και δεν είναι συμπαγή. Το εσωτερικό τους αποτελείται από νεύρα που σχηματίζουν τετράγωνες κυψέλες. Με τον τρόπο αυτό μειώνεται σημαντικά το βάρος τους και ο χρόνος εκτύπωσης, χωρίς ωστόσο να τίθεται σε κίνδυνο η αντοχή τους. Για τα διαφορετικά τμήματα της βάσης χρησιμοποιήθηκαν διαφορετικές ρυθμίσεις, ανάλογα με τις απαιτήσεις ακαμψίας και αντοχής. Οι βασικές ρυθμίσεις που τροποποιήθηκαν ήταν το πάχος των τοιχωμάτων και η πυκνότητα των νεύρων στο εσωτερικό τους. Στην εικόνα 2-19 παρουσιάζεται το 713ο επίπεδο εκτύπωσης από τα συνολικά 749 που χρειάστηκαν για να κατασκευαστούν οι προεκτάσεις για το κουτί των αισθητήρων.

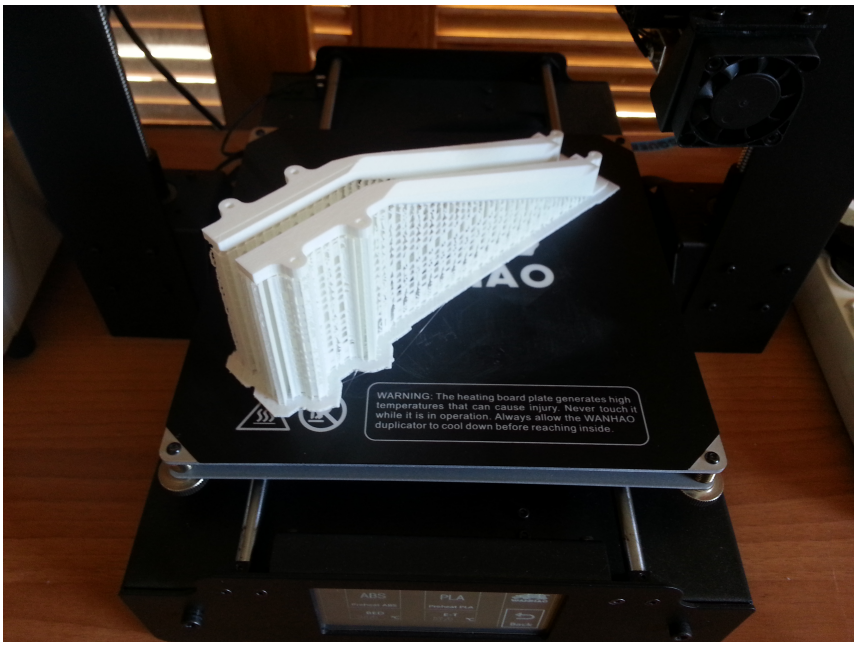


Εικόνα 2-19. Το 713ο επίπεδο εκτύπωσης των προεκτάσεων για το κουτί των αισθητήρων.

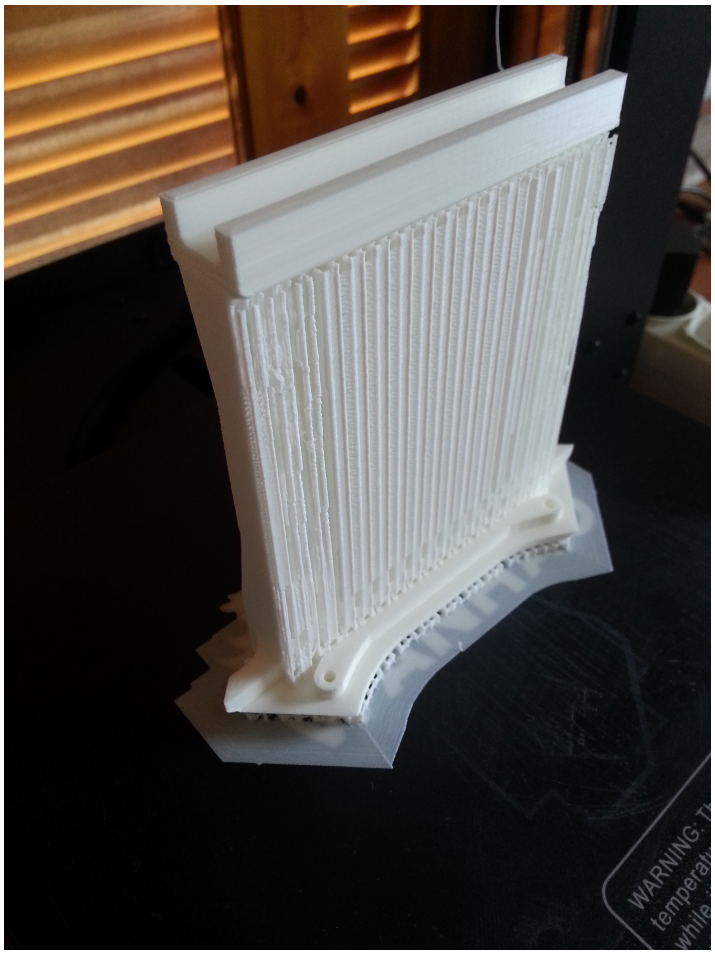
Για ορισμένα τμήματα, όπως φαίνεται και στην εικόνα 2-19, χρειάστηκε να εκτυπωθεί και κατάλληλη υποστήριξη, ώστε να μην υπάρχουν επίπεδα που τυπώνονται πάνω στο κενό και κινδυνεύουν να παραμορφωθούν. Ωστόσο, οι επιφάνειες που έρχονται σε επαφή με την υποστήριξη δεν παρουσιάζουν ομαλή και λεία τελική επιφάνεια όπως οι υπόλοιπες, εξαιτίας του υλικού που αποκολλάται από την υποστήριξη και προσκολλάται στην επιφάνεια (εικόνα 2-20).

Εικόνα 2-20. Υλικό από την υποστήριξη που έχει προσκολληθεί την επιφάνεια του αντικειμένου.

Για τον λόγο αυτό, τα τμήματα τοποθετήθηκαν στον εκτυπωτή με τέτοιο τρόπο, ώστε οι επιφάνειες που είναι σημαντικές για την σωστή εφαρμογή με τα υπόλοιπα τμήματα της βάσης να είναι καθαρές και να μην έχουν ελαττώματα. Τα τμήματα που χρειάστηκαν υποστήριξη ήταν οι τέσσερεις προεκτάσεις (δύο για τους αισθητήρες και δύο για την μπαταρία) και τα δύο ζευγάρια δοκών. Στις εικόνες 2-21 και 2-22 παρουσιάζονται τα προαναφερθέντα τμήματα, ακατέργαστα, αμέσως μετά το τέλος της εκτύπωσης. Το συνολικό βάρος όλων των τμημάτων ανέρχεται σε 278 γραμμάρια (εικόνα 2-23).

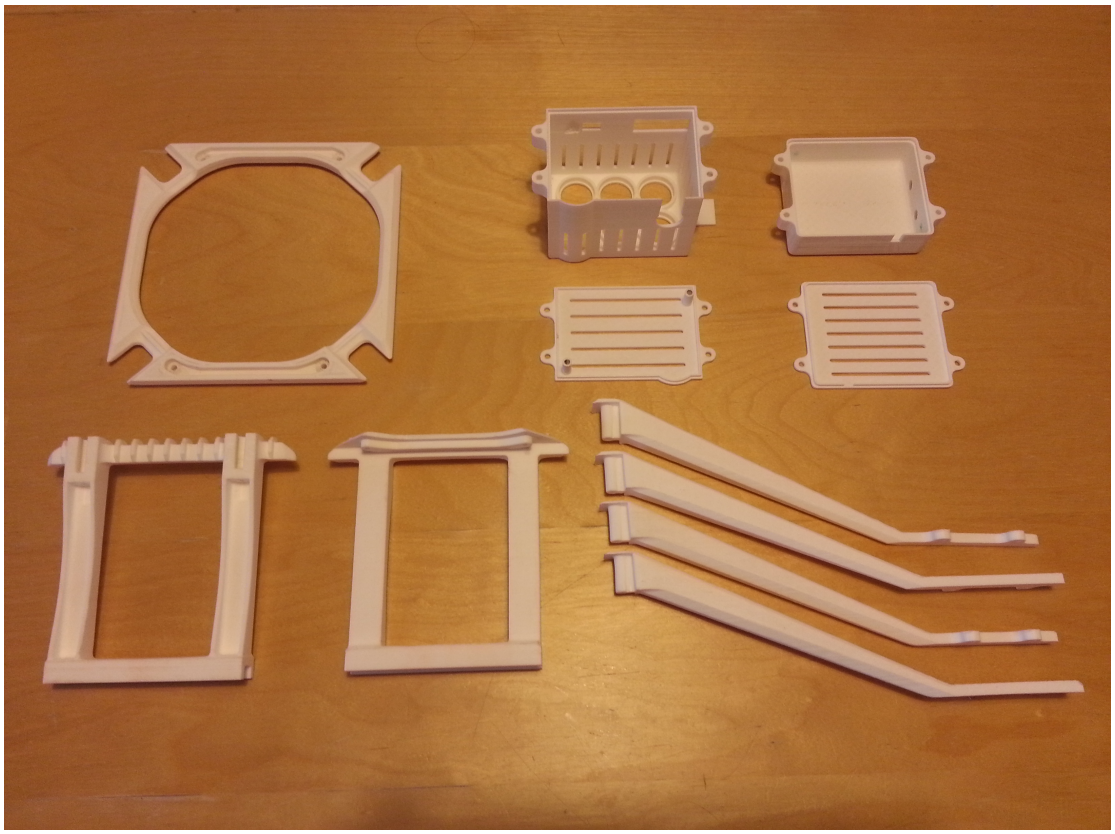


Εικόνα 2-21.Οι προεκτάσεις του κουτιού των αισθητήρων αμέσως μετά την εκτύπωση.

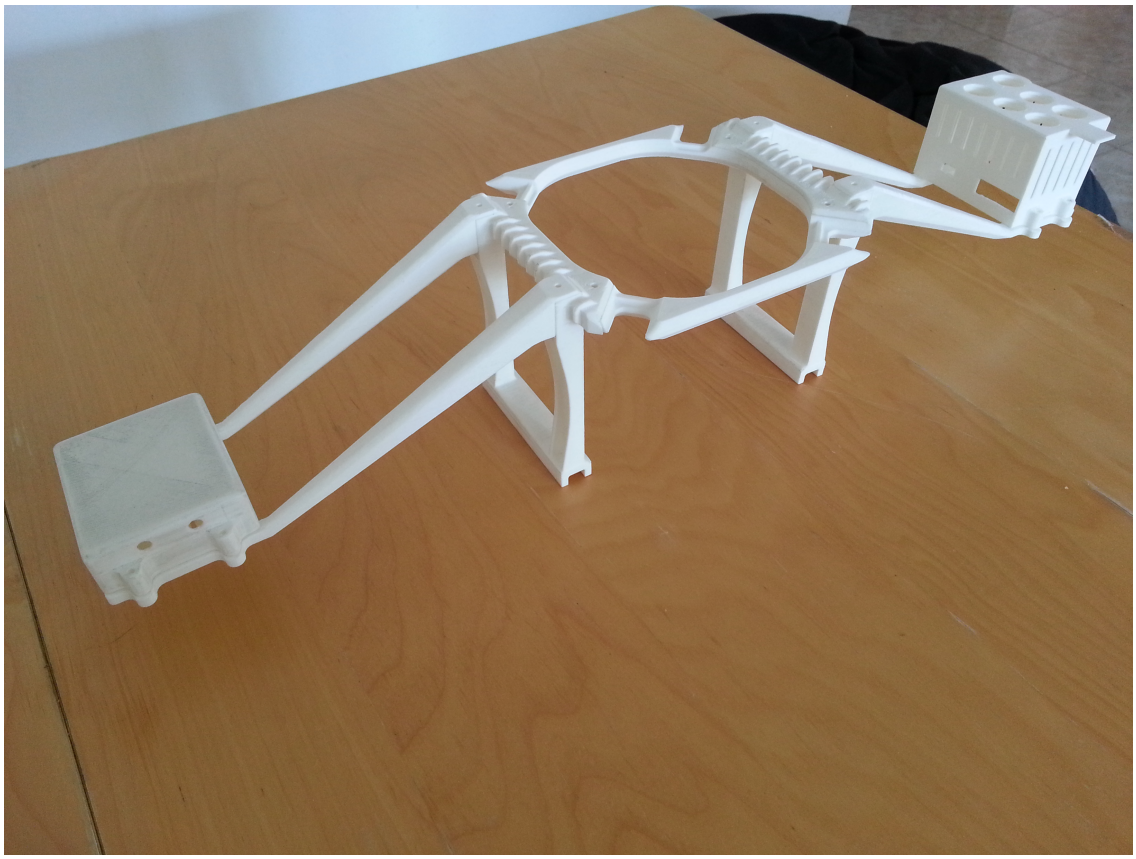


Εικόνα 2-22. Το ζευγάρι δοκών αμέσως μετά την εκτύπωση.

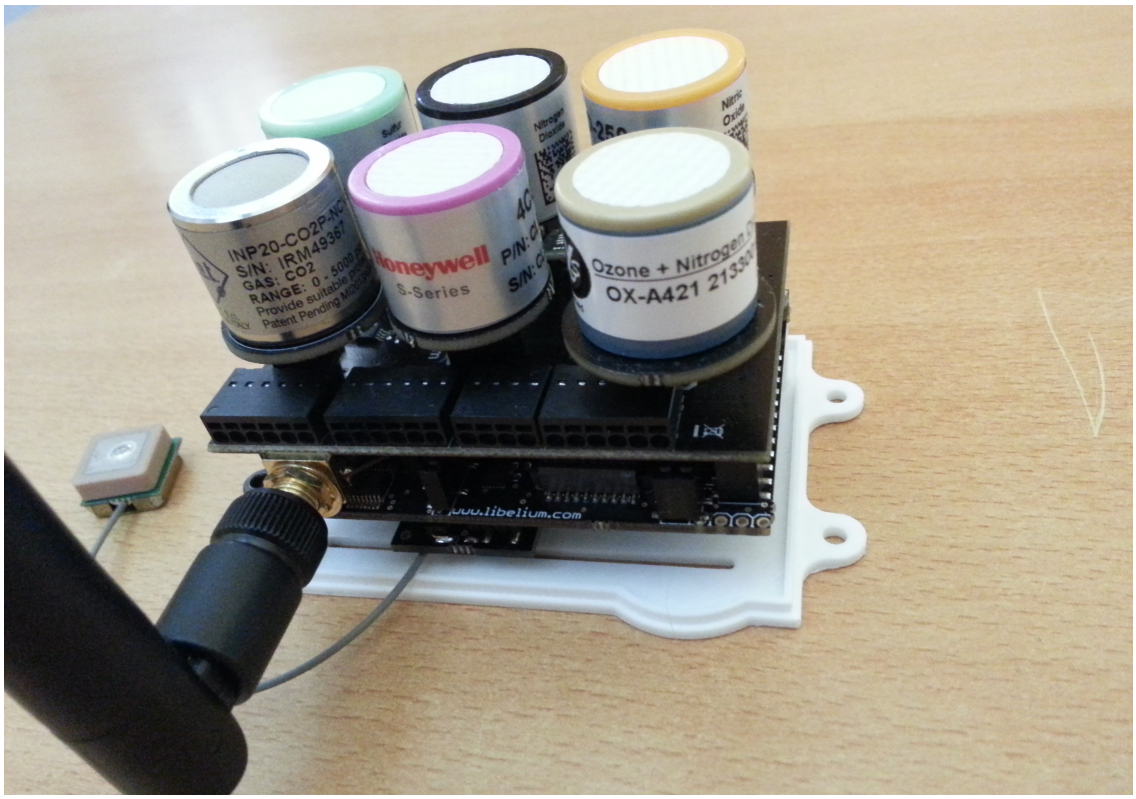
Εικόνα 2-23. Το συνολικό βάρος όλων των τμημάτων.



Εικόνα 2-24. Τα τελικά τμήματα συγκεντρωμένα.

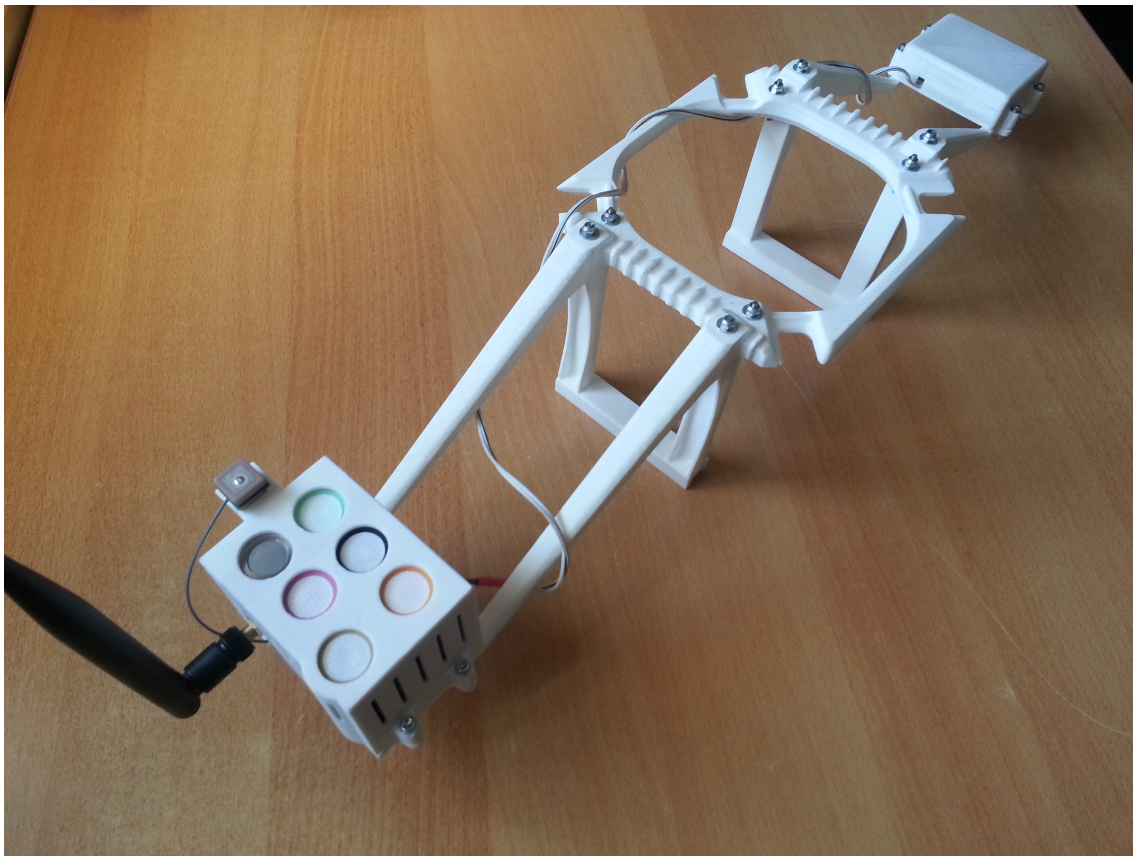


Εικόνα 2-25. Η τελική εικόνα της βάσης.



Εικόνα 2-26. Το ηλεκτρονικό σύστημα βιδωμένο στο κάτω μέρος του κουτιού.

Εικόνα 2-27. Το ηλεκτρονικό σύστημα εγκατεστημένο στο κουτί του.



Εικόνα 2-28. Το ηλεκτρονικό σύστημα και η βάση, έτοιμα για να εγκατασταθούν στο αεροσκάφος.

Εικόνα 2-29. Το ηλεκτρονικό σύστημα εγκατεστημένο στο αεροσκάφος.

3. Συλλογή και απεικόνιση δεδομένων

3.1. Προγραμματισμός των αισθητήρων

Ο προγραμματισμός των αισθητήρων έγινε σε γλώσσα "C" και ακολουθήθηκε η εξής λογική. Αρχικά ενεργοποιούνται οι αισθητήρες, ο ασύρματος πομπός και ο GPS δέκτης. Μετά την ενεργοποίηση των αισθητήρων, το σύστημα μπαίνει σε κατάσταση ύπνου για ένα διάστημα 3 λεπτών προκειμένου να έρθουν οι αισθητήρες σε θερμοκρασία λειτουργίας. Στη συνέχεια, αφού ο GPS δέκτης κλειδώσει στους δορυφόρους, διαβάζονται οι τιμές από όλους τους αισθητήρες και αποστέλλονται στην βάση ως ένα "frame". Ένα frame είναι η στιγμιαία εικόνα του συστήματος και αποτελείται από μια επικεφαλίδα και ένα σώμα (εικόνα 3-1).

HEADER										PAYLOAD						
<=>	Frame Type	Num Fields	#	Serial ID	#	Waspmote ID	#	Sequence	#	Sensor_1	#	Sensor_2	#	...	Sensor_n	#

Εικόνα 3-1. Απεικόνιση ενός frame.

Η επικεφαλίδα αποτελείται από:

- Την ακολουθία συμβόλων "<=>", που δηλώνει την αρχή του frame
- Τον τύπο του frame (binary ή ASCII)
- Τον αριθμό των πεδίων που θα αποσταλούν
- Τον σειριακό αριθμό του Waspmote
- Το όνομα του Waspmote που έχει δώσει ο χρήστης
- Τον αύξον αριθμό του frame

Το σώμα αποτελείται από τις παρακάτω τιμές που μετρούνται:

- Ποσοστό φόρτισης μπαταρίας
- Συγκέντρωση CO_2, CO, SO_2, NO_2, NO, O_3
- Θερμοκρασία, υγρασία, πίεση
- Υφόμετρο, ταχύτητα, προσανατολισμός
- Γεωγραφικό μήκος και πλάτος
- Ώρα και ημερομηνία

Το μέγεθος του κάθε frame είναι περίπου 300 bytes. Το γεγονός αυτό αποτέλεσε πρόβλημα καθώς ο buffer του πομπού έχει μικρότερο μέγεθος, με αποτέλεσμα το frame να μην μπορεί να δημιουργηθεί και να σταλεί με μια εκπομπή.

Για τον λόγο αυτό, τροποποιήθηκε κατάλληλα η βιβλιοθήκη που σχετίζεται με το μέγιστο μέγεθος του frame, ώστε να μπορεί να δημιουργηθεί ένα με το κατάλληλο μέγεθος. Στη συνέχεια το κάθε ένα απεστάλη με διαδοχικές εκπομπές μεγέθους μέχρι 100 bytes.

3.2. Ο κώδικας του Waspmote σε γλώσσα C

```
/*
The code run at first the setup() function and the then loops infinitely the loop()
function. The setup() function turns on the Xbee module(the radio transceiver), the
sensors and then tries to lock GPS satellites. The loop() function checks first for
the GPS signal. If GPS signal exists, then the readSensors() function reads the
sensor values, the createNewFrame() function creates the new frame with those values
and finally the sendFrame() function transmits the frame (in parts of 100 bytes
maximum, because of the modules buffer limitation). Optionally, the printOutput()
functions prints the data to the serial port.

Need to change MAX_FRAME in "WaspFrame.h" to higher value to have all the data to
one frame (i.e. "#define MAX_FRAME 512").
*/

/* ============================ Initializations ============================*/
#include <WaspSensorGas_Pro.h>
#include <WaspFrame.h>
#include <WaspGPS.h>
#include <WaspXBee802.h>

// Time for GPS timeout
#define TIMEOUT 60
// Define BROADCAST MAC address
char RX_ADDRESS[] = "000000000000FFFF";
// Define the Waspmote ID
char WASPMOTE_ID[] = "node_01";

uint8_t error;
uint8_t myBuffer[100];

Gas CO2(SOCKET_1);
Gas CO(SOCKET_2);
Gas O3(SOCKET_3);
Gas SO2(SOCKET_4);
Gas NO2(SOCKET_5);
Gas NO(SOCKET_6);

float temperature;  // Stores the temperature in ºC
float humidity;     // Stores the realitve humidity in %RH
float pressure;     // Stores the pressure in Pa
float concCO2;      // Stores the concentration of CO2 in ppm
float concCO;       // Stores the concentration of CO in ppm
float concO3;       // Stores the concentration of O3 in ppm
float concSO2;      // Stores the concentration of SO2 in ppm
```

```
float concNO2;      // Stores the concentration of NO2 in ppm
float concNO;       // Stores the concentration of NO in ppm

char node_ID[] = "node1";
// Define status variable for GPS connection
bool status = false;

/* ======================== End of Initializations ========================*/

/* ============================== Functions ==============================*/

void readSensors() {
      ////////////////////////////////////////
      // Reading sensor values
      ////////////////////////////////////////

      // Read the CO sensor and compensate with the temperature internally
      concCO = CO.getConc();
      // Read the CO2 sensor and compensate with the temperature internally
      concCO2 = CO2.getConc();
      // Read the SO2 sensor and compensate with the temperature internally
      concSO2 = SO2.getConc();
      // Read the NO2 sensor and compensate with the temperature internally
      concNO2 = NO2.getConc();
      // Read the NO sensor and compensate with the temperature internally
      concNO = NO.getConc();
      // Read the O3 sensor and compensate with the temperature internally
      concO3 = O3.getConc();
      // Read temperature, humidity and pressure
      temperature = SO2.getTemp();
      humidity = SO2.getHumidity();
      pressure = SO2.getPressure();
}

void printOutput() {

      // Printing values via USB
      USB.println(F("***************************************"));
      USB.print(F("CO2 concentration: "));
      USB.print(concCO2);
      USB.println(F(" ppm"));
      USB.print(F("CO concentration: "));
      USB.print(concCO);
      USB.println(F(" ppm"));
      USB.print(F("SO2 concentration: "));
      USB.print(concSO2);
      USB.println(F(" ppm"));
      USB.print(F("NO2 concentration: "));
      USB.print(concNO2);
      USB.println(F(" ppm"));
      USB.print(F("NO concentration: "));
      USB.print(concNO);
      USB.println(F(" ppm"));
      USB.print(F("O3 concentration: "));
      USB.print(concO3);
      USB.println(F(" ppm"));
      USB.print(F("Temperature: "));
      USB.print(temperature);
      USB.println(F(" Celsius degrees"));
      USB.print(F("RH: "));
      USB.print(humidity);
      USB.println(F(" %"));
      USB.print(F("Pressure: "));
```

```
        USB.print(pressure);
        USB.println(F(" Pa"));
}

void createNewFrame() {
        ///////////////////////////////////////
        // Create ASCII frame
        ///////////////////////////////////////

        // Create new frame (ASCII)
        frame.createFrame(ASCII);

        // Add frame fields
        frame.addSensor(SENSOR_STR, "new_sensor_frame");
        // Add gpslatitude
        frame.addSensor(SENSOR_GP_NH3, GPS.latitude);
        // Add gpslongitude
        frame.addSensor(SENSOR_GP_H2, GPS.longitude);
        // Add battery Level
        frame.addSensor(SENSOR_BAT, PWR.getBatteryLevel());
        // Add CO2 concentration
        frame.addSensor(SENSOR_GP_CO2, concCO2);
        // Add CO concentration
        frame.addSensor(SENSOR_GP_CO, concCO);
        // Add SO2 concentration
        frame.addSensor(SENSOR_GP_SO2, concSO2);
        // Add NO2 concentration
        frame.addSensor(SENSOR_GP_NO2, concNO2);
        // Add NO concentration
        frame.addSensor(SENSOR_GP_NO, concNO);
        // Add O3 concentration
        frame.addSensor(SENSOR_GP_O3, concO3);
        // Add temperature
        frame.addSensor(SENSOR_GP_TC, temperature);
        // Add humidity
        frame.addSensor(SENSOR_GP_HUM, humidity);
        // Add pressure
        frame.addSensor(SENSOR_GP_PRES, pressure);
        // Add global position [degrees]
        frame.addSensor(SENSOR_GPS,
                GPS.convert2Degrees(GPS.latitude, GPS.NS_indicator),
                GPS.convert2Degrees(GPS.longitude, GPS.EW_indicator));
        // Add altitude [m]
        frame.addSensor(SENSOR_ALTITUDE, GPS.altitude);
        // Add speed [km/h]
        frame.addSensor(SENSOR_SPEED, GPS.speed);
        // Add course [degrees]
        frame.addSensor(SENSOR_COURSE, GPS.course);
        // Add time
        frame.addSensor(SENSOR_TIME, GPS.timeGPS);
        // Add date
        frame.addSensor(SENSOR_DATE, GPS.dateGPS);
        // Add gpslat
}

void GPSlock() {
        ///////////////////////////////////////
        // GPS connection
        ///////////////////////////////////////
        USB.println(F("Wait for GPS connection..."));
        // Power on GPS
        GPS.ON();
        status = false;
```

```
	while (status == false) {
		status = GPS.waitForSignal(TIMEOUT);
		if (status == true) {
			// getPosition function gets all basic data
			status = GPS.getPosition();
			USB.println(F("GPS connected."));
		}
		else {
			USB.println("GPS not locked trying again");
		}
	}
}

void sendFrame() {
	// number of total bytes to send
	int totalLength = frame.length;
	// number of remaining bytes to send
	int remaining = totalLength;
	// the current byte to send
	int currentByte = 0;
	int maxBytes;
	int i;

	// Send the whole frame in parts
	while (currentByte < totalLength) {
		if (totalLength - currentByte >= 100) {
			maxBytes = 100;
		}
		else {
			maxBytes = totalLength - currentByte;
		}
		for (i = 0; i<maxBytes; i++) {
			myBuffer[i] = frame.buffer[currentByte];
			currentByte++;
		}
		error = xbee802.send(RX_ADDRESS, myBuffer, maxBytes);
	}

	// check TX flag
	if (error == 0) {
		frame.showFrame();
		// blink green LED
		Utils.blinkGreenLED();
	}
	else {
		USB.println(F("send error"));
		// blink red LED
		Utils.blinkRedLED();
	}
}
/* ============================ End of Functions ==============================*/


/* ======================== setup function and main loop ====================*/
void setup() {
	USB.println(F("###########\r\nCode verion:gases3.1\r\n###########\r\n"));

	// Power on XBee transceiver
	xbee802.ON();
	// Set the Waspmote ID
	frame.setID(node_ID);
	USB.println(F("Warming up sensors..."));
```

```
        // Turn on the sensors
        CO.ON();
        SO2.ON();
        NO2.ON();
        NO.ON();
        CO2.ON();
        O3.ON();

        // Sleep for 3 minutes to warm up the sensors
        PWR.deepSleep("00:00:03:00", RTC_OFFSET, RTC_ALM1_MODE1, ALL_ON);
        USB.println(F("OK"));
        // Lock GPS cordinates
        GPSlock();
}

void loop() {
        status = GPS.waitForSignal(TIMEOUT);
        if (status == true) {
                readSensors();
                GPS.getPosition();
                createNewFrame();
                sendFrame();
                printOutput();
        }
        else {
                USB.println(F("GPS didn't lock."));
        }
        // Wait for 5 seconds till next loop
        delay(5000);
}
/* ================================== End ====================================*/
```

3.3. Κώδικας σε Python για συλλογή και επεξεργασία δεδομένων

Όπως αναφέρθηκε προηγουμένως στα τεχνικά χαρακτηριστικά των αισθητήρων (Κεφάλαιο 1), οι αισθητήρες έχουν κάποιο χρόνο απόκρισης στις μεταβολές του περιβάλλοντος . Ο χρόνος αυτός κυμαίνεται από 1 έως 60 δευτερόλεπτα. Στην παρούσα εργασία οι αισθητήρες κινούνται διαρκώς μαζί με το αεροσκάφος μέσα σε ένα ανεξαρτήτως κινούμενο ρευστό, τον αέρα. Επειδή η διάρκεια της πτήσης μειώθηκε στα 15 λεπτά περίπου, λόγο του επιπρόσθετου βάρους, θεωρήθηκε σκόπιμο να γίνουν κάποιες δοκιμές σχετικά με την απόκριση των αισθητήρων, ώστε οι μετρήσεις να γίνονται ανά 10 με 15 δευτερόλεπτα.

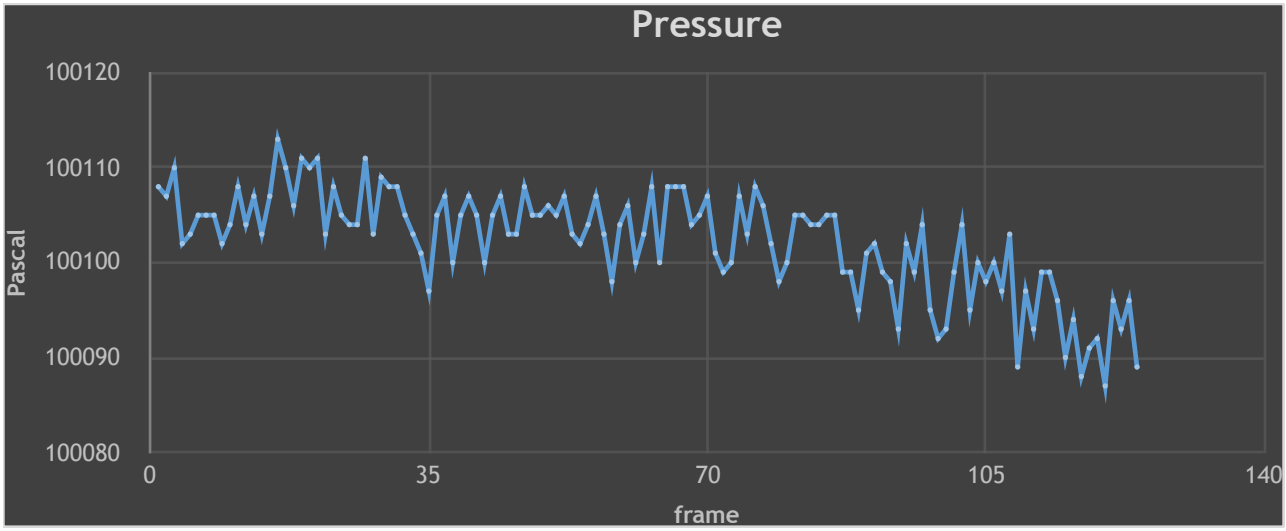
Για τον λόγο αυτό δημιουργήθηκε κατάλληλος κώδικας σε γλώσσα προγραμματισμού “Python”, ο οποίος διαβάζει το αρχείο δεδομένων που δημιουργεί ο προαναφερθείς κώδικας σε C του Waspmote (Παράγραφος 3.2) και δημιουργεί ένα δεύτερο αρχείο σε μορφή “xlsx”. Το αρχείο δεδομένων δημιουργείται είτε από τις καταγραφές των frames μέσω ασύρματου δέκτη, είτε από τις εκτυπώσεις που κάνει ο κώδικας μέσω της σειριακής θύρας (συνάρτηση USB.println()). Στο αρχείο xlsx που δημιουργεί ο κώδικας σε Python οι στήλες αντιστοιχούν στα υπό μέτρηση μεγέθη, για κάθε ένα από τα οποία δημιουργείται και το αντίστοιχο γράφημα. Στη συνέχεια παρουσιάζεται ο εν λόγο κώδικας.

```
#This code takes 'data.log' or 'sensor.log' as input file and outputs an xlsx
formated file (Excel file). The 'data.log' file contains the radio transmitted
frames and the 'sensor.log' file contains all the resial output that Waspmote
prints. The user has to choose (by comment-uncomment) which log file wants to read
from. GetFrames() function reads 'data.log' file and getSerialFrames() reads
'sensor.log' file. Both functions get the frames from the log file and
outputFrames() function creates the Excel worksheet.


import xlsxwriter

workbook = xlsxwriter.Workbook('output.xlsx')   #Create excel output file
worksheet = workbook.add_worksheet()

#worksheet.write( selected_cell , data_to_put_on_the_cell)
worksheet.write('A1', 'frame\sensors')
worksheet.write('B1', 'batt')
worksheet.write('C1', 'temp')
worksheet.write('D1', 'hum')
worksheet.write('E1', 'press')
worksheet.write('F1', 'CO2')
worksheet.write('G1', 'CO')
worksheet.write('H1', 'SO2')
worksheet.write('I1', 'NO2')
worksheet.write('J1', 'NO')
worksheet.write('K1', 'O3')

#Reading frames from the "data.log" file. It contains ONLY frames.
def getFrames():
    #frame[] is an array of frames
    #data[] is an array with the sensor names and values in a frame

    #Open and read the log file
    with open('data.log', 'r') as f:
        read_data = f.read()
    frame = read_data.split('<=>')
    print('Found ' + str(len(frame) - 1) + ' frames.')

    for i in range(1,len(frame)):
        data = frame[i].split('#')
        if len(data) == 24:
            outputFrames(i,data)                #Print output
```

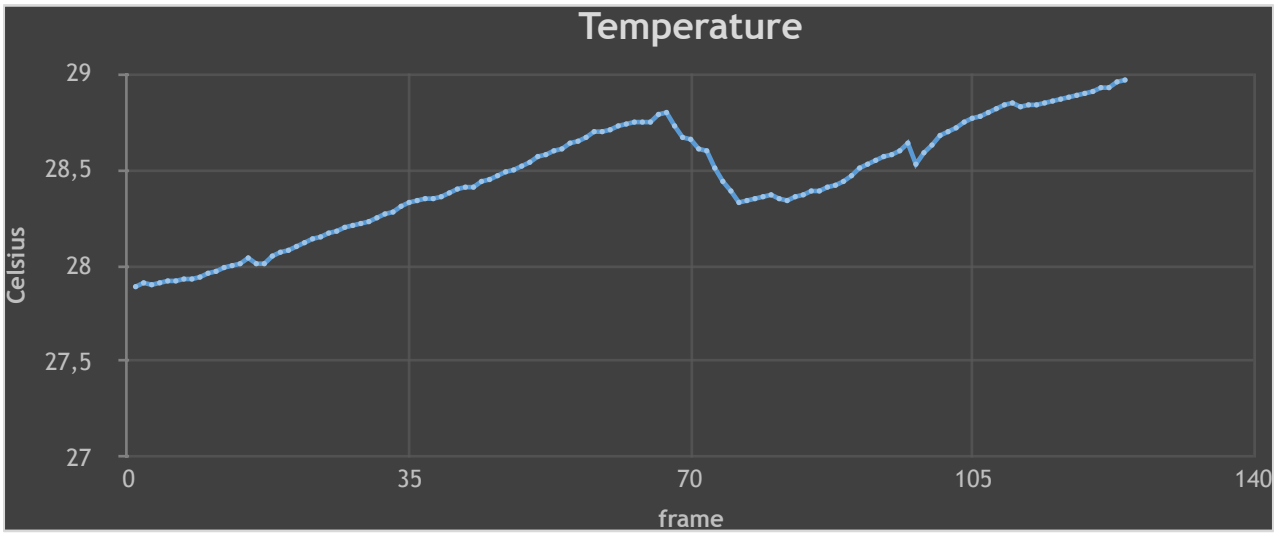
```
#Reading frames from the "sensor.log" file. It contains ALL serial output (with the
frames included).
def getSerialFrames():
    #frame[] is an array of frames
    #data[] is an array with the sensor names and values in a frame

    #Open and read the log file
    with open('sensor.log', 'r') as g:
        read_data = g.read()
    frame = read_data.split('<=>')
    print('Found ' + str(len(frame)/2) + ' frames.')
    for i in range(1,len(frame),2):
        data = frame[i].split('\r\n')[0]
        data = frame[i].split('#')
        if len(data) == 24:
            outputFrames(i,data)                #Print output


#Printing selected sensor data of a frame on the excel worksheet row(i+1) from cell
'A' to 'K'.
def outputFrames(i, data):
      #frame
      worksheet.write('A' + str(i+1), int(data[3].rpartition(':')[2]))
      #battery
      worksheet.write('B' + str(i+1), int(data[7].rpartition(':')[2]))
      #temperature
      worksheet.write('C' + str(i+1), float(data[14].rpartition(':')[2]))
      #humidity
      worksheet.write('D' + str(i+1), float(data[15].rpartition(':')[2]))
      #pressure
      worksheet.write('E' + str(i+1), float(data[16].rpartition(':')[2]))
      #CO2
      worksheet.write('F' + str(i+1), float(data[8].rpartition(':')[2]))
      #CO
      worksheet.write('G' + str(i+1), float(data[9].rpartition(':')[2]))
      #SO2
      worksheet.write('H' + str(i+1), float(data[10].rpartition(':')[2]))
      #NO2
      worksheet.write('I' + str(i+1), float(data[11].rpartition(':')[2]))
      #NO
      worksheet.write('J' + str(i+1), float(data[12].rpartition(':')[2]))
      #O3
      worksheet.write('K' + str(i+1), float(data[13].rpartition(':')[2]))


#Uncomment one of the two following functions below:
#getFrames()
getSerialFrames()

#Close excel worksheet file ('output.xlsx')
workbook.close()
```

3.4. Έλεγχος απόκρισης αισθητήρων

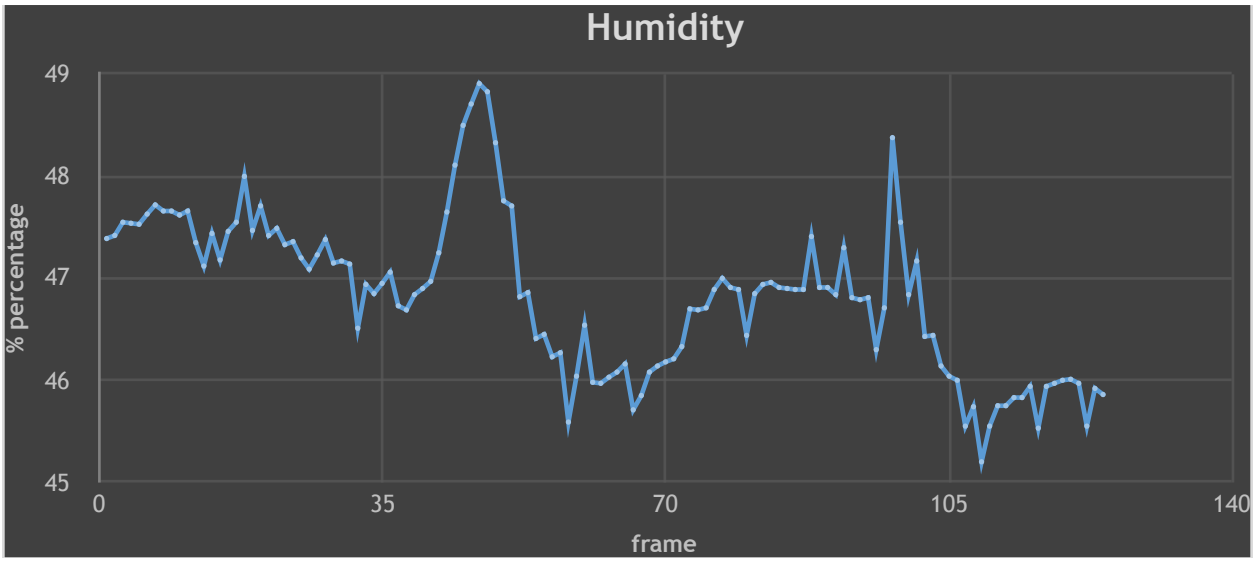
Για τον έλεγχο της απόκρισης των αισθητήρων διεξήχθη το εξής πείραμα. Οι αισθητήρες τοποθετήθηκαν σε ένα δοχείο, το οποίο ήταν γυρισμένος ανάποδα. Ενώ το σύστημα ισορροπούσε, τοποθετήθηκε μέσα στο δοχείο ένα μικρό κερί και μετά από ένα χρονικό διάστημα αφαιρέθηκε, όπως επίσης και το δοχείο. Αφού το σύστημα επανήλθε σε ισορροπία, άρχισε η προσθήκη όζοντος με χρήση ειδικής συσκευής. Η καταγραφή των δεδομένων έγινε ενσύρματα με χρήση της σειριακής θύρας και τα αποτελέσματα εξήχθησαν σε αρχείο excel μέσω του κώδικα σε Python που προαναφέρθηκε. Το κερί βρισκόταν στο δοχείο από το 40ο μέχρι το 50ο περίπου frame, ενώ η προσθήκη όζοντος έγινε μεταξύ του 75ου και 95ου frame. Κάθε frame είχε χρονική διαφορά 10 περίπου δευτερολέπτων με το επόμενο. Στα παρακάτω γραφήματα παρουσιάζονται τα αποτελέσματα των μετρήσεων.



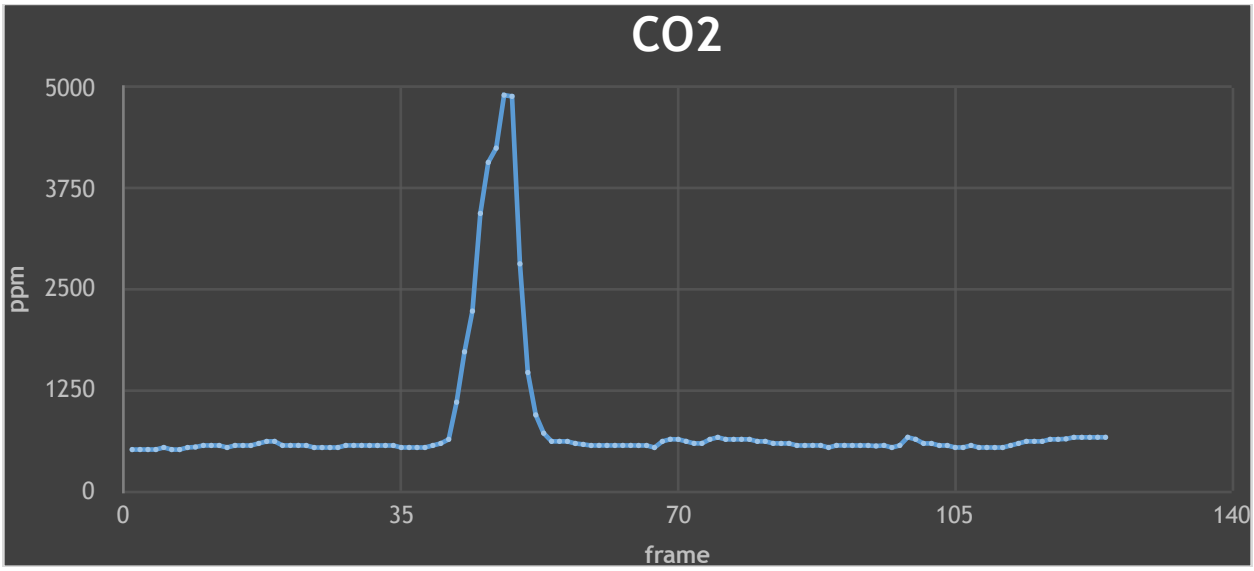
Διάγραμμα πίεσης - αριθμού επαναλήψεων



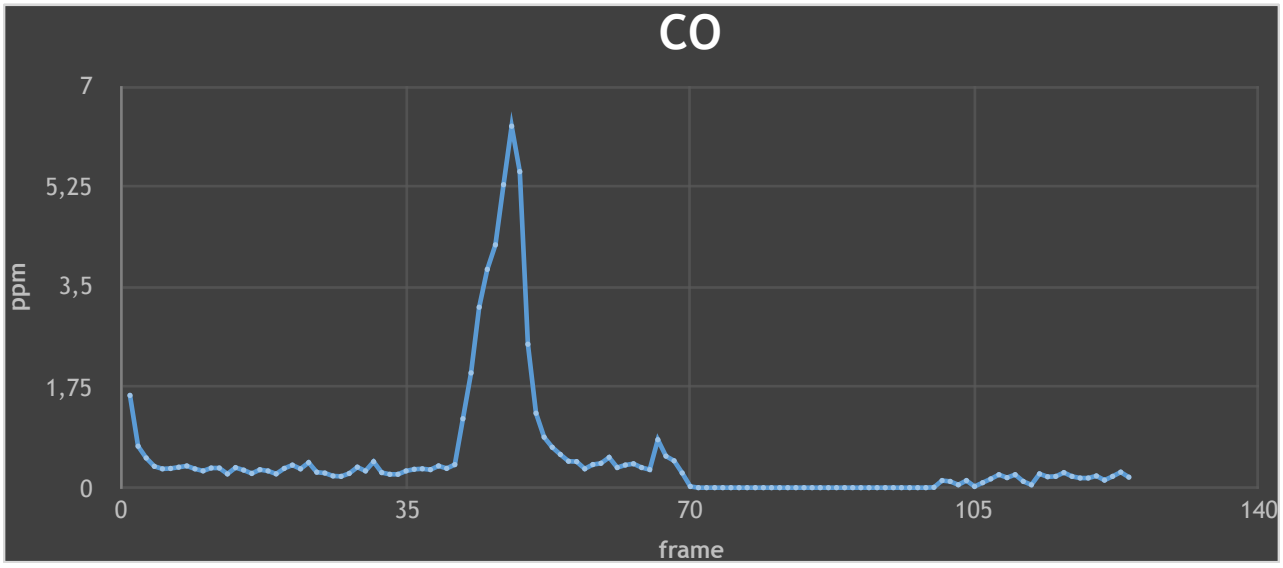
Διάγραμμα θερμοκρασίας - αριθμού επαναλήψεων



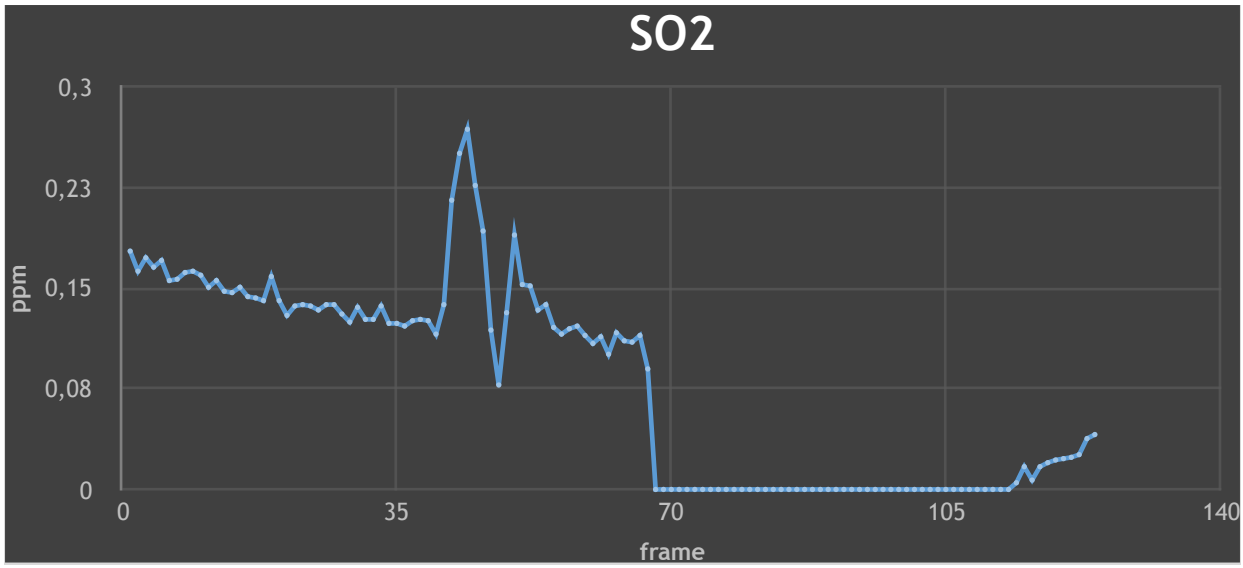
Διάγραμμα υγρασίας - αριθμού επαναλήψεων



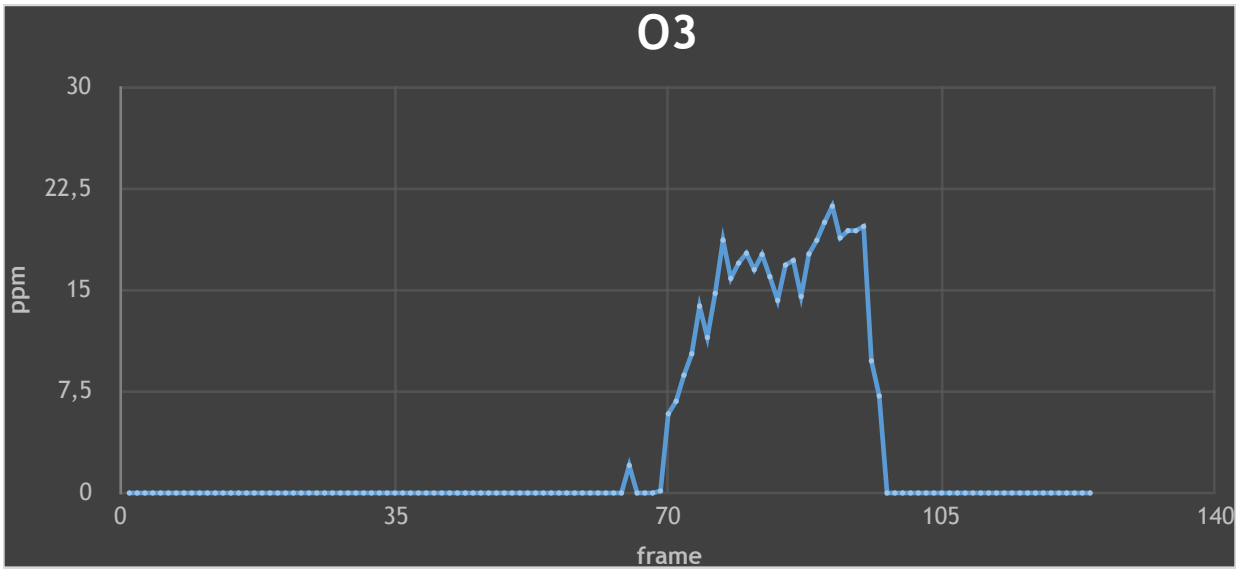
Διάγραμμα CO2 - αριθμού επαναλήψεων



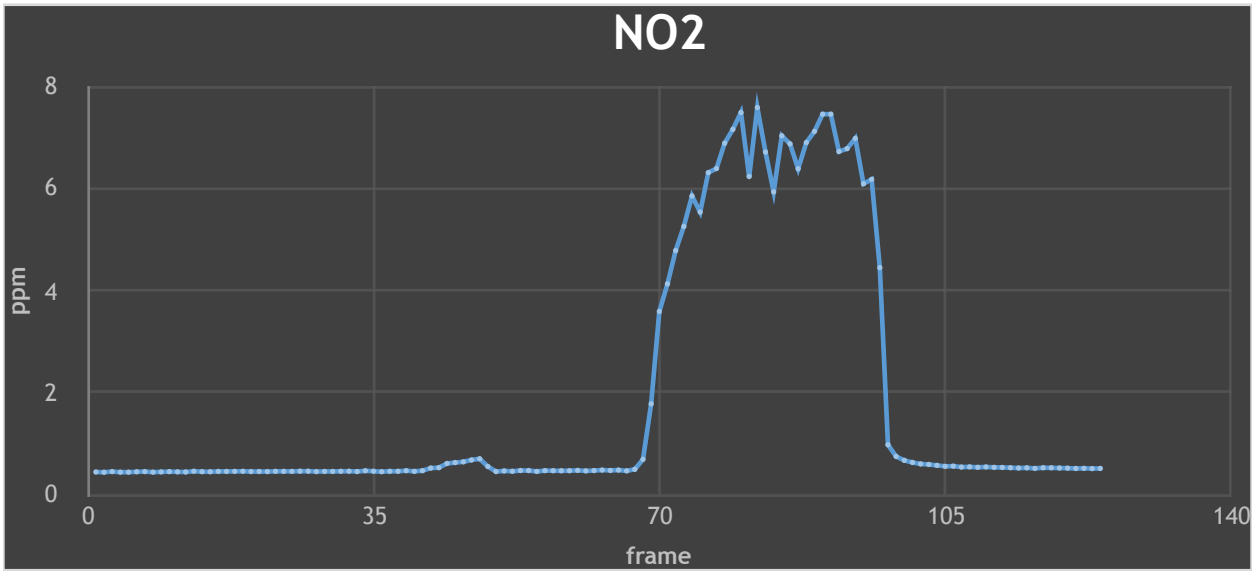
Διάγραμμα CO - αριθμού επαναλήψεων



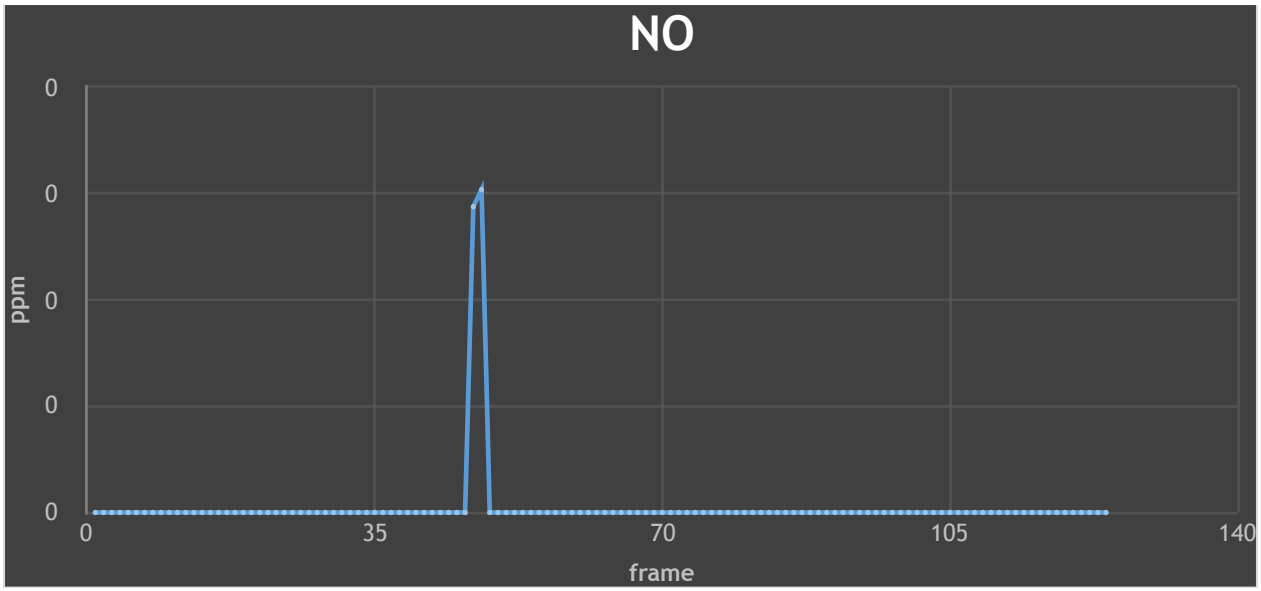
Διάγραμμα SO2 - αριθμού επαναλήψεων



Διάγραμμα O3 - αριθμού επαναλήψεων



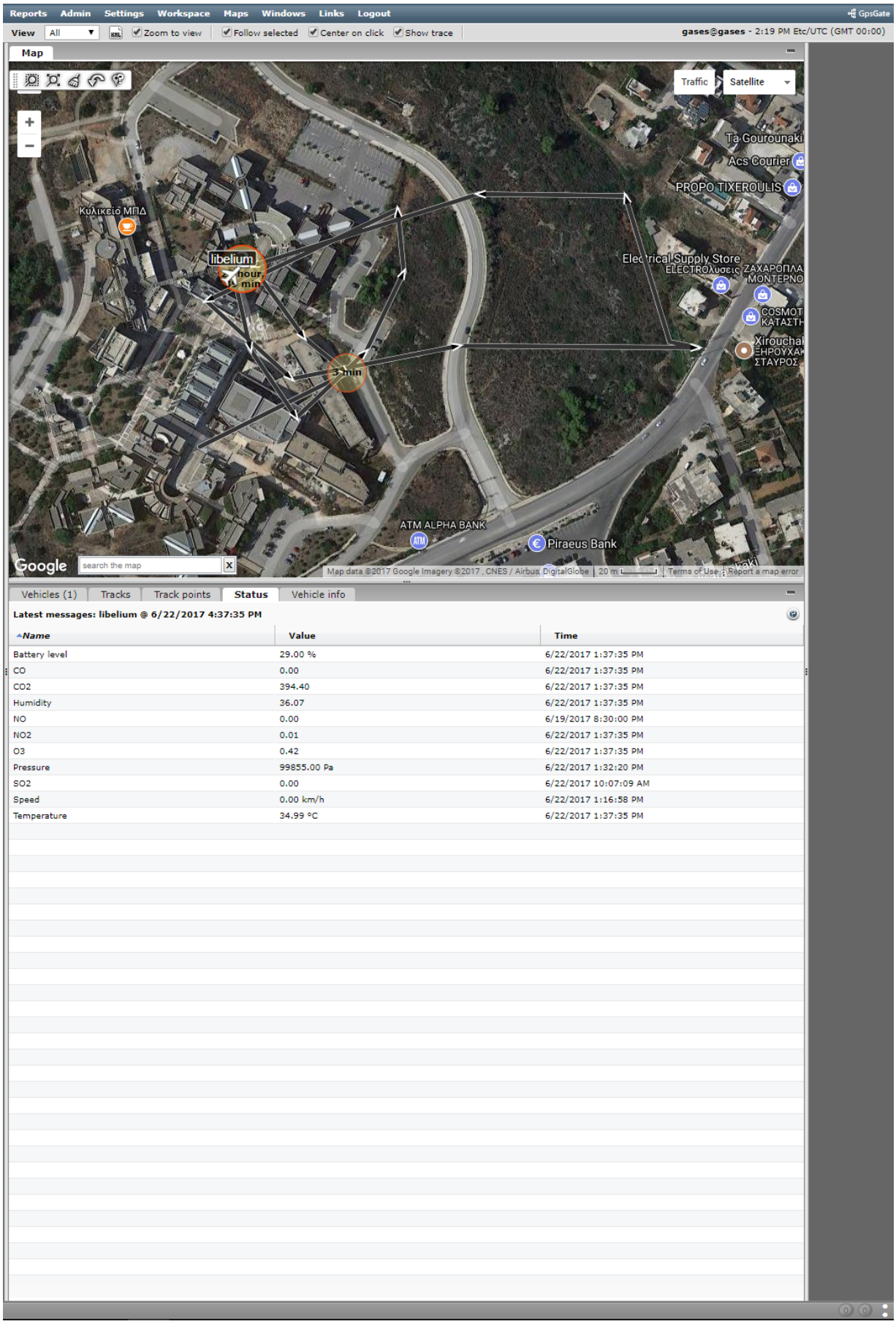
Διάγραμμα NO2 - αριθμού επαναλήψεων



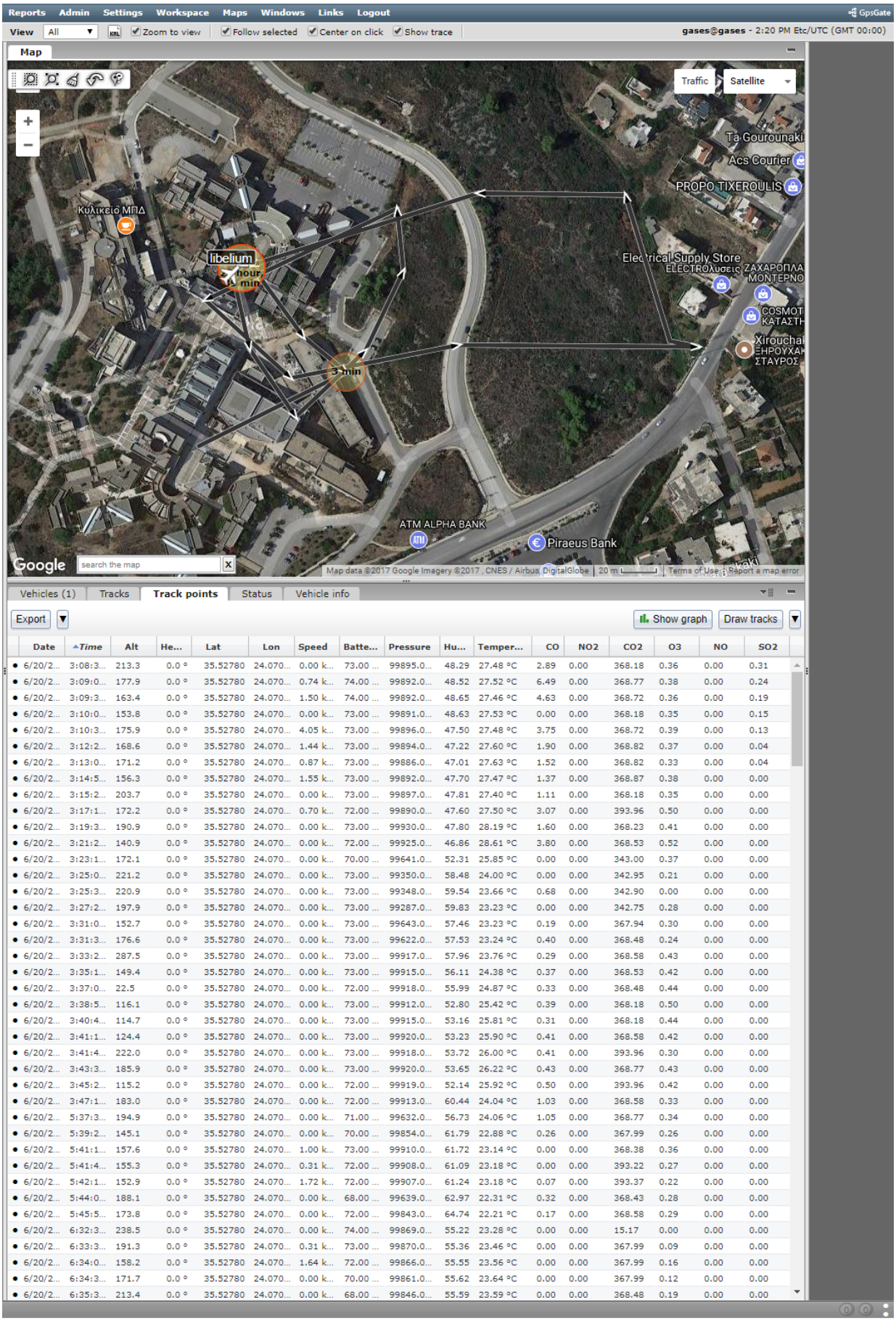
Διάγραμμα NO - αριθμού επαναλήψεων

3.5. Ζωντανή απεικόνιση δεδομένων μέσω διαδικτύου

Για την ζωντανή απεικόνιση των μετρήσεων χρησιμοποιήθηκε η πλατφόρμα της GpsGate. Πρόκειται για ένα "online" γραφικό περιβάλλον που χρησιμοποιείται για επαγγελματική παρακολούθηση οχημάτων και στόλων. Η πλατφόρμα εγκαταστάθηκε σε server που παραχωρήθηκε από το Πολυτεχνείο Κρήτης και παραμετροποιήθηκε κατάλληλα ώστε να μπορεί να δέχεται σαν μεταβλητές τα δεδομένα των αισθητήρων. Κάθε frame που φτάνει ασύρματα στο Meshlium, αφού αποθηκευτεί εσωτερικά σε αυτό, αποστέλλεται αυτομάτως στην πλατφόρμα με κατάλληλο HTTP /GET request, η οποία με τη σειρά της το αποθηκεύει στην δικιά της βάση δεδομένων και το απεικονίζει στον χρήστη. Το περιβάλλον της πλατφόρμας είναι απολύτως παραμετροποιήσιμο, και επιτρέπει στον χρήστη να λαμβάνει εύκολα την πληροφορία που χρειάζεται, είτε σε πραγματικό χρόνο (εικόνα 3-2), είτε μεταγενέστερα ανατρέχοντας σε δεδομένα που έχουν αποθηκευτεί στην βάση δεδομένων (εικόνα 3-3). Από την πλατφόρμα μπορούν να εξαχθούν τα δεδομένα σε αρχεία διαφόρων μορφών (π.χ. kml, csv), ή μπορούν να εξαχθούν γραφήματα με τα επιλεγμένα δεδομένα (εικόνα 3-4).

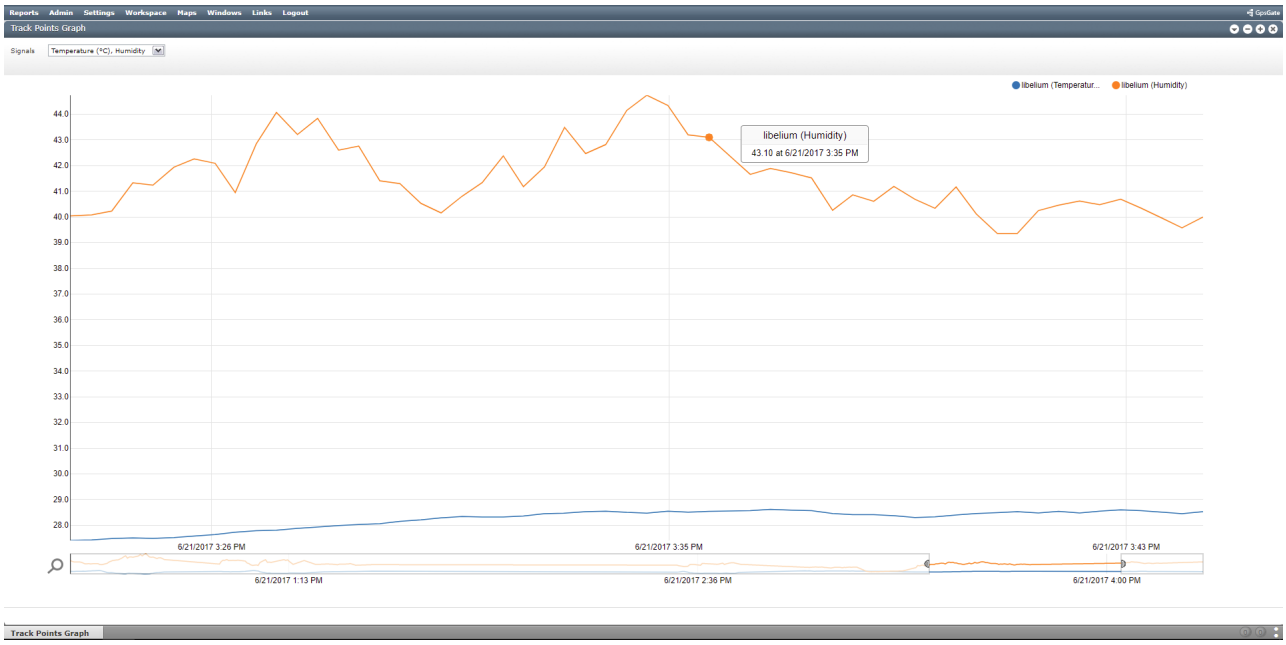


Εικόνα 3-2. Εικόνα με τις τελευταίες μετρήσεις που έχουν αποσταλεί για κάθε αισθητήρα.



Date	Time	Alt	He...	Lat	Lon	Speed	Batte...	Pressure	Hu...	Temper...	CO	NO2	CO2	O3	NO	SO2
6/20/2...	3:08:3...	213.3	0.0 °	35.52780	24.070...	0.00 k...	73.00 ...	99895.0...	48.29	27.48 °C	2.89	0.00	368.18	0.36	0.00	0.31
6/20/2...	3:09:0...	177.9	0.0 °	35.52780	24.070...	0.74 k...	74.00 ...	99892.0...	48.52	27.52 °C	6.49	0.00	368.77	0.38	0.00	0.24
6/20/2...	3:09:3...	163.4	0.0 °	35.52780	24.070...	1.50 k...	74.00 ...	99892.0...	48.65	27.46 °C	4.63	0.00	368.72	0.36	0.00	0.19
6/20/2...	3:10:0...	153.8	0.0 °	35.52780	24.070...	0.00 k...	73.00 ...	99891.0...	48.63	27.53 °C	0.00	0.00	368.18	0.35	0.00	0.15
6/20/2...	3:10:3...	175.9	0.0 °	35.52780	24.070...	4.05 k...	73.00 ...	99896.0...	47.50	27.48 °C	3.75	0.00	368.72	0.39	0.00	0.13
6/20/2...	3:12:2...	168.6	0.0 °	35.52780	24.070...	1.44 k...	73.00 ...	99894.0...	47.22	27.60 °C	1.90	0.00	368.82	0.37	0.00	0.04
6/20/2...	3:13:0...	171.2	0.0 °	35.52780	24.070...	0.87 k...	73.00 ...	99886.0...	47.01	27.63 °C	1.52	0.00	368.82	0.33	0.00	0.04
6/20/2...	3:14:5...	156.3	0.0 °	35.52780	24.070...	1.55 k...	73.00 ...	99892.0...	47.70	27.47 °C	1.37	0.00	368.87	0.38	0.00	0.00
6/20/2...	3:15:2...	203.7	0.0 °	35.52780	24.070...	0.00 k...	73.00 ...	99897.0...	47.81	27.40 °C	1.11	0.00	368.18	0.35	0.00	0.00
6/20/2...	3:17:1...	172.2	0.0 °	35.52780	24.070...	0.70 k...	72.00 ...	99890.0...	47.60	27.50 °C	3.07	0.00	393.96	0.50	0.00	0.00
6/20/2...	3:19:3...	190.9	0.0 °	35.52780	24.070...	0.00 k...	73.00 ...	99930.0...	47.80	28.19 °C	1.60	0.00	368.23	0.41	0.00	0.00
6/20/2...	3:21:2...	140.9	0.0 °	35.52780	24.070...	0.00 k...	72.00 ...	99925.0...	46.86	28.61 °C	3.80	0.00	368.53	0.52	0.00	0.00
6/20/2...	3:23:1...	172.1	0.0 °	35.52780	24.070...	0.00 k...	70.00 ...	99641.0...	52.31	25.85 °C	0.00	0.00	343.00	0.37	0.00	0.00
6/20/2...	3:25:0...	221.2	0.0 °	35.52780	24.070...	0.00 k...	73.00 ...	99350.0...	58.48	24.00 °C	0.00	0.00	342.95	0.21	0.00	0.00
6/20/2...	3:25:3...	220.9	0.0 °	35.52780	24.070...	0.00 k...	73.00 ...	99348.0...	59.54	23.66 °C	0.68	0.00	342.90	0.00	0.00	0.00
6/20/2...	3:27:2...	197.9	0.0 °	35.52780	24.070...	0.00 k...	73.00 ...	99287.0...	59.83	23.23 °C	0.00	0.00	342.75	0.28	0.00	0.00
6/20/2...	3:31:0...	152.7	0.0 °	35.52780	24.070...	0.00 k...	73.00 ...	99643.0...	57.46	23.23 °C	0.19	0.00	367.94	0.30	0.00	0.00
6/20/2...	3:31:3...	176.6	0.0 °	35.52780	24.070...	0.00 k...	73.00 ...	99622.0...	57.53	23.24 °C	0.40	0.00	368.48	0.24	0.00	0.00
6/20/2...	3:33:2...	287.5	0.0 °	35.52780	24.070...	0.00 k...	73.00 ...	99917.0...	57.96	23.76 °C	0.29	0.00	368.58	0.43	0.00	0.00
6/20/2...	3:35:1...	149.4	0.0 °	35.52780	24.070...	0.00 k...	73.00 ...	99915.0...	56.11	24.38 °C	0.37	0.00	368.53	0.42	0.00	0.00
6/20/2...	3:37:0...	22.5	0.0 °	35.52780	24.070...	0.00 k...	72.00 ...	99918.0...	55.99	24.87 °C	0.33	0.00	368.48	0.44	0.00	0.00
6/20/2...	3:38:5...	116.1	0.0 °	35.52780	24.070...	0.00 k...	73.00 ...	99912.0...	52.80	25.42 °C	0.39	0.00	368.18	0.50	0.00	0.00
6/20/2...	3:40:4...	114.7	0.0 °	35.52780	24.070...	0.00 k...	73.00 ...	99915.0...	53.16	25.81 °C	0.31	0.00	368.18	0.44	0.00	0.00
6/20/2...	3:41:1...	124.4	0.0 °	35.52780	24.070...	0.00 k...	73.00 ...	99920.0...	53.23	25.90 °C	0.41	0.00	368.58	0.42	0.00	0.00
6/20/2...	3:41:4...	222.0	0.0 °	35.52780	24.070...	0.00 k...	73.00 ...	99918.0...	53.72	26.00 °C	0.41	0.00	393.96	0.30	0.00	0.00
6/20/2...	3:43:3...	185.9	0.0 °	35.52780	24.070...	0.00 k...	73.00 ...	99920.0...	53.65	26.22 °C	0.43	0.00	368.77	0.43	0.00	0.00
6/20/2...	3:45:2...	115.2	0.0 °	35.52780	24.070...	0.00 k...	72.00 ...	99919.0...	52.14	25.92 °C	0.50	0.00	393.96	0.42	0.00	0.00
6/20/2...	3:47:1...	183.0	0.0 °	35.52780	24.070...	0.00 k...	72.00 ...	99913.0...	60.44	24.04 °C	1.03	0.00	368.58	0.33	0.00	0.00
6/20/2...	5:37:3...	194.9	0.0 °	35.52780	24.070...	0.00 k...	71.00 ...	99632.0...	56.73	24.06 °C	1.05	0.00	368.77	0.34	0.00	0.00
6/20/2...	5:39:2...	145.1	0.0 °	35.52780	24.070...	0.00 k...	70.00 ...	99854.0...	61.79	22.88 °C	0.26	0.00	367.99	0.26	0.00	0.00
6/20/2...	5:41:1...	157.6	0.0 °	35.52780	24.070...	1.00 k...	73.00 ...	99910.0...	61.72	23.14 °C	0.00	0.00	368.38	0.36	0.00	0.00
6/20/2...	5:41:4...	155.3	0.0 °	35.52780	24.070...	0.31 k...	72.00 ...	99908.0...	61.09	23.18 °C	0.00	0.00	393.22	0.27	0.00	0.00
6/20/2...	5:42:1...	152.9	0.0 °	35.52780	24.070...	1.72 k...	72.00 ...	99907.0...	61.24	23.18 °C	0.07	0.00	393.37	0.22	0.00	0.00
6/20/2...	5:44:0...	188.1	0.0 °	35.52780	24.070...	0.00 k...	68.00 ...	99639.0...	62.97	22.31 °C	0.32	0.00	368.43	0.28	0.00	0.00
6/20/2...	5:45:5...	173.8	0.0 °	35.52780	24.070...	0.00 k...	72.00 ...	99843.0...	64.74	22.21 °C	0.17	0.00	368.58	0.29	0.00	0.00
6/20/2...	6:32:3...	238.5	0.0 °	35.52780	24.070...	0.00 k...	74.00 ...	99869.0...	55.22	23.28 °C	0.00	0.00	15.17	0.00	0.00	0.00
6/20/2...	6:33:3...	191.3	0.0 °	35.52780	24.070...	0.31 k...	73.00 ...	99870.0...	55.36	23.46 °C	0.00	0.00	367.99	0.09	0.00	0.00
6/20/2...	6:34:0...	158.2	0.0 °	35.52780	24.070...	1.64 k...	72.00 ...	99866.0...	55.55	23.56 °C	0.00	0.00	367.99	0.16	0.00	0.00
6/20/2...	6:34:3...	171.7	0.0 °	35.52780	24.070...	0.00 k...	70.00 ...	99861.0...	55.62	23.64 °C	0.00	0.00	367.99	0.12	0.00	0.00
6/20/2...	6:35:3...	213.4	0.0 °	35.52780	24.070...	0.00 k...	68.00 ...	99846.0...	55.59	23.59 °C	0.00	0.00	368.48	0.19	0.00	0.00

Εικόνα 3-3. Εικόνα με σημειακές μετρήσεις κατά τη διάρκεια μιας πτήσης.



Εικόνα 3-4. Γράφημα θερμοκρασίας(μπλε) – υγρασίας(πορτοκαλί).

4. Συμπεράσματα

4.1. Αποτελέσματα μετρήσεων

Τα αποτελέσματα των μετρήσεων είναι αρκετά ικανοποιητικά τόσο στις δοκιμές που έγιναν σε κλειστό χώρο, όσο και στις δοκιμές που έγιναν στον αέρα. Αξίζει να σημειωθεί ότι σε μετρήσεις που έγιναν υπό τις ίδιες συνθήκες με την παράγραφο 3.4 αλλά με χρονική διαφορά μεγαλύτερη του μισού λεπτού μεταξύ των μετρήσεων, οι καμπύλες είχαν την ίδια μορφή. Το γεγονός αυτό δείχνει ότι οι αισθητήρες ανταποκρίνονται αρκετά γρήγορα σε απότομες μεταβολές του περιβάλλοντος.

Κατά την διαδικασία δοκιμής των αισθητήρων παρουσιάστηκαν αρχικά πολλά προβλήματα στην συμπεριφορά των αισθητήρων, αλλά και του GPS δέκτη. Τα προβλήματα αυτά δημιουργήθηκαν εξαιτίας τις χρήσης των πιο πρόσφατων βιβλιοθηκών της Libelium, οι οποίες αποδείχθηκε ότι δεν είναι συμβατές με την προγενέστερη έκδοση του Waspmote που χρησιμοποιήθηκε στην παρούσα εργασία.

4.2. Πτητική Ικανότητα αεροσκάφους

Η τελική πτητική ικανότητα του αεροσκάφους παρέμεινε σχεδόν ίδια με την αρχική, ενώ οι αισθητήρες και η κάμερα του αεροσκάφους παρέμειναν ανεμπόδιστοι από το ηλεκτρονικό σύστημα. Το αεροσκάφος όπως είναι φυσικό λόγο του επιπρόσθετου βάρους απέκτησε μεγαλύτερη αδράνεια, αλλά η διαφορά στον χειρισμό είναι αμελητέα. Το χαρακτηριστικό που δεν έμεινε ανεπηρέαστο από την προσθήκη του βάρους είναι η διάρκεια πτήσης, η οποία μειώθηκε περίπου στο μισό (αναλόγως των συνθηκών). Το συνολικό επιπρόσθετο βάρος ανέρχεται σε 551 γραμμάρια.

4.3. Μελλοντικές προτάσεις

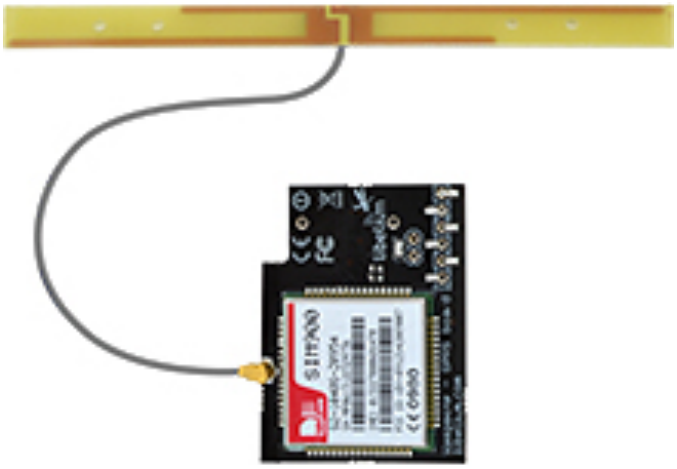
Μείωση βάρους

Το βάρος του συστήματος θα μπορούσε να μειωθεί με χρήση πιο ελαφριάς μπαταρίας. Η κίνηση αυτή θα είχε σαν αποτέλεσμα την επιπλέον μείωση του βάρους των πλαστικών τμημάτων της βάσης χρησιμοποιώντας πιο λεπτά τοιχώματα.

Η αντικατάσταση της μπαταρίας συνεπάγεται εξολοκλήρου νέα σχεδίαση της βάσης, καθώς σε αντίθετη περίπτωση το κέντρο βάρους του συστήματος θα μετατοπιζόταν σημαντικά προς την μεριά των αισθητήρων. Μια ιδέα για την σχεδίαση ενός πιο αποδοτικού συστήματος είναι η εγκατάσταση της μπαταρίας και του Waspmote κοντά στο κέντρο του αεροσκάφους και η απομάκρυνση των αισθητήρων μόνο εκτός της ροής αέρα, χρησιμοποιώντας προεκτάσεις καλωδίων. Οι αισθητήρες επικοινωνούν με το Waspmote ψηφιακά, οπότε η χρήση των προεκτάσεων δεν θα προκαλέσει σφάλματα στις μετρήσεις.

Χρήση GSM/GPRS module

Η χρήση του Meshlium είναι ιδανική για περιπτώσεις όπου η περιοχή που πρέπει να σαρωθεί είναι συγκεκριμένη καθώς η εμβέλειά του είναι αρκετά μεγάλη. Σε περιπτώσεις όπου η περιοχή σάρωσης είναι απομακρυσμένη, η μεταφορά του και η σύνδεσή του στο διαδίκτυο μπορεί να είναι δύσκολη. Για το λόγο αυτό θα μπορούσε να γίνει χρήση ενός GSM/GPRS module (εικόνα 4-1), το οποίο κάνοντας χρήση του δικτύου της κινητής τηλεφωνίας θα μπορεί να στέλνει τα δεδομένα στο Meshlium απομακρυσμένα.



Εικόνα 4-1 GSM/GPRS module.

(Πηγή: Libelium.com)