



ΣΧΟΛΗ ΜΗΧΑΝΙΚΩΝ
ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ
ΗΛΕΚΤΡΟΝΙΚΩΝ ΣΥΣΤΗΜΑΤΩΝ
ΠΡΟΓΡΑΜΜΑ ΜΕΤΑΠΤΥΧΙΑΚΩΝ ΣΠΟΥΔΩΝ
«ΕΦΑΡΜΟΣΜΕΝΑ ΗΛΕΚΤΡΟΝΙΚΑ ΣΥΣΤΗΜΑΤΑ»

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ: Μέτρηση ποιότητας αέρα με χρήση
περιβαντολογικών αισθητήρων

Αλέξανδρος Τιντίνης

ΕΠΙΒΛΕΠΩΝ ΚΑΘΗΓΗΤΗΣ: Γιακουμής Άγγελος

ΘΕΣΣΑΛΟΝΙΚΗ, ΦΕΒΡΟΥΑΡΙΟΣ 2024

Περιεχόμενα

Ευχαριστίες	6
1 Πρόλογος	7
1.1 Σύντομη περιγραφή	8
1.2 Αναλυτική Περιγραφή	8
2 Ανίχνευση αερίων – γενικά στοιχεία	10
2.1 Αιωρούμενα σωματίδια	10
2.2 Πτητικές οργανικές ενώσεις (VOC)	10
2.3 Οξείδια του αζώτου (NOx)	10
2.4 Ανίχνευση αερίων	11
2.4.1 Ιστορική αναδρομή	11
2.4.2 Σύγχρονα συστήματα ανίχνευσης	11
2.5 Τύποι αισθητήρων ανίχνευσης αερίων	11
2.5.1 Καταλυτικοί αισθητήρες	11
2.5.2 Αισθητήρες θερμικής αγωγιμότητας	11
2.5.3 Ηλεκτροχημικοί αισθητήρες	12
2.5.4 Οπτικοί αισθητήρες	12
2.5.5 Αισθητήρες ημιαγωγού	12
2.5.6 Αισθητήρες ακουστικού κύματος	12
3 Συστήματα παρακολούθησης και ελέγχου	13
3.1 Ηλεκτρικές συνδέσεις	13
3.2 Ανιχνευτές	13
3.2.1 Αναλογικοί ανιχνευτές	14
3.3 Κεντρικές μονάδες	14
3.4 Περιφερειακές μονάδες εισόδου - εξόδου	15
3.5 Διευθυνσιοδοτούμενα συστήματα - πυρανίχνευση	16
3.6 Δικτύωση πινάκων	18
3.7 Δίκτυο περιβαλλοντολογικών αισθητήρων	18
4 Σύστημα GreenLink Pro	19
4.1 Βαθμίδες συστήματος	19
4.1.1 Δίκτυο αισθητήρων RS-485	20
4.1.2 Πύλη (gateway) RS-485	20
4.1.3 Διαχειριστής δικτύου αισθητήρων / Web Server	20
4.1.4 Web client	21

4.2	Προδιαγραφές συστήματος	21
	4.2.1 Απαιτήσεις συστήματος	21
	4.2.2 Βασικές προδιαγραφές συστήματος	22
	4.2.3 Καταστάσεις συστήματος	23
	4.2.4 Στάθμες συναγερμού	24
	4.2.5 Περιπτώσεις σφαλμάτων	25
4.3	Περιβαντολογικός αισθητήρας SEN55	25
	4.3.1 Μπλοκ διάγραμμα	26
	4.3.2 Τεχνικά χαρακτηριστικά	26
	4.3.3 Λειτουργικές καταστάσεις	29
4.4	Αισθητήρας θερμοκρασίας	30
5	Δίκτυο επικοινωνίας αισθητήρων	30
5.1	Τα βασικά του RS-485	30
5.2	Πρωτόκολλο επικοινωνίας	31
5.3	Πακέτα πρωτοκόλλου	32
	5.3.1 Πακέτο αποστολής master (request packet)	33
	5.3.2 Master payload	34
	5.3.3 Πακέτο αποστολής slave (response packet)	34
	5.3.4 Slave payload	35
	5.3.5 Διαχωρισμός σε επίπεδα (L1 – L2)	35
5.4	Δικλίδες διασφάλισης πακέτου	36
6	Hardware	36
6.1	Θεωρητικό σχέδιο πλακέτας αισθητήρα	36
	6.1.1 Αναγνώριση βαθμίδων	36
	6.1.2 Τροφοδοσία	36
	6.1.3 Οδηγός RS-485	38
	6.1.4 Περιβαντολογικός αισθητήρας	39
	6.1.5 Αισθητήρας θερμοκρασίας	39
	6.1.6 Ψηφιακοί είσοδοι – έξοδοι	40
6.2	Πλακέτα αισθητήρα (PCB)	41
	6.2.1 Σχέδιο πλακέτας	42
	6.2.2 Ενδεικτικό κόστος πλακέτας αισθητήρα	43
6.3	Προσαρμογέας δικτύου RS-485	44
	6.3.1 Περιγραφή λειτουργίας	44

6.3.2	Ηλεκτρονικά σχέδια	44
7	Firmware αισθητήρα	46
7.1	Οργάνωση προγράμματος	46
7.1.1	Λογισμικό υψηλού επιπέδου (επίπεδο εφαρμογής)	48
7.1.2	Επίπεδο υπομονάδων (modules)	50
7.1.3	Επίπεδο HAL χρήστη	50
7.1.4	Επίπεδο HAL STM32	51
7.1.5	Επίπεδο firmware	51
7.2	Ρυθμίσεις MCU αισθητήρα – αρχείο .ioc	51
8	Μετρήσεις αισθητήρα	53
8.1	Περιβαντολογικός αισθητήρας	54
8.1.1	Έκθεση σε φως	54
8.1.2	Μακροπρόθεσμη μέτρηση ηρεμίας σε εσωτερικό χώρο	55
8.1.3	Μεταβολές σε θερμοκρασία – υγρασία	56
8.1.4	Σύγκριση ένδειξης θερμοκρασίας με πραγματικό θερμόμετρο	58
8.1.5	Μεταβολές σε μικροσωματίδια και NOx index	58
8.1.6	Μεταβολές σε μικροσωματίδια και VOC index	60
8.1.7	Μεταβολές στον VOC index	62
8.2	Αναλογικός αισθητήρας θερμοκρασίας	63
8.2.1	Μακροπρόθεσμη μέτρηση ηρεμίας σε εσωτερικό χώρο	63
8.3	Σύγκριση τιμών μεταξύ των δύο τύπων αισθητηρίων	64
8.3.1	Μετρήσεις	65
8.4	Μετρήσεις ρεύματος	67
8.4.1	Αξιολόγηση μετρήσεων ρεύματος	69
8.4.2	Μελέτη κατανάλωσης ρεύματος	71
8.5	Μετρήσεις RS-485	71
9	Εγκατάσταση και εκκίνηση συστήματος	73
9.1	Συνδέσεις συστήματος	73
9.1.1	Επιλογή διεύθυνσης αισθητήρα	75
9.2	Εγκατάσταση λογισμικού ΚΕΣ (GreenLink Pro)	76
9.3	Εκκίνηση με γραφικά	76
9.3.1	Αρχείο δικτύου αισθητήρων	77
9.4	Εκκίνηση με γραμμή εντολών	78
10	Software Κεντρικού Ελεγκτή Συστήματος (GreenLink Pro)	79

10.1	Threading system	79
	10.1.1 Εργασίες thread διαχείρισης αισθητήρα	80
	10.1.2 Εργασίες thread τοπικού διακομιστή (server)	80
10.2	Thread διαχείρισης αισθητήρα	81
	10.2.1 Application interface	83
	10.2.2 Module interface	83
	10.2.3 Module δικτύου RS-485	85
	10.2.4 Module αισθητήρα	86
	10.2.5 Module αποθήκευσης (storage)	88
	10.2.6 Module διεπαφής χρήστη (User Interface ή UI module)	88
10.3	Thread τοπικού διακομιστή (Server)	89
	10.3.1 Presenter	89
	10.3.2 Model	91
	10.3.3 View	91
	10.3.4 Use case interactor	91
	10.3.5 Διάγραμμα ακολουθίας server	91
10.4	Logging module	92
	10.4.1 Ορίσματα	93
10.5	Σημείο εισόδου χρήστη (Entry point)	94
	10.5.1 Εκκίνηση με γραφικά (start.py)	94
	10.5.2 main.py	94
10.6	Σύστημα οργάνωσης αρχείων (repository)	96
11	Ιστοσελίδα (Web page)	96
11.1	Επεξήγηση λειτουργίας	97
	11.1.1 Πρόσβαση από το τοπικό δίκτυο	99
11.2	Components (.html)	100
11.3	Στυλ (.css)	100
11.4	Λειτουργικότητα (.js)	100
12	Κατασκευή παρουσίασης	101
13	Μελλοντικές επεκτάσεις	102
14	Επίλογος – Συμπεράσματα	103
15	Βιβλιογραφικές αναφορές	105
16	Πίνακες περιεχομένων εικόνων και πινάκων	108
16.1	Πίνακας περιεχομένων εικόνων	108

16.2	Πίνακας περιεχομένων πινάκων	110
17	Παραρτήματα	111
17.1	Συντομογραφίες	111

Ευχαριστίες

Ολοκληρώνοντας την παρούσα διπλωματική εργασία, ολοκληρώνω και τον κύκλο σπουδών μου στο Πρόγραμμα Μεταπτυχιακών Σπουδών «Εφαρμοσμένα Ηλεκτρονικά Συστήματα».

Η πιο θερμή μου ευχαριστία πηγαίνει στην οικογένειά μου. Στη σύζυγό μου Νατάσα και στα δύο μου παιδιά Μάνο και Εριέττα, για την ολόψυχη στήριξη που μου παρείχαν και την υπομονή που έδειξαν καθ' όλη τη διάρκεια των μεταπτυχιακών μου σπουδών.

Νοιώθω επίσης την ανάγκη να ευχαριστήσω τον υπεύθυνο καθηγητή της διπλωματικής μου εργασίας, κύριο Άγγελο Γιακουμή, για τις χρήσιμες συμβουλές και παρεμβάσεις του κατά τη διάρκεια της εργασίας.

Τέλος θα ήθελα να ευχαριστήσω τους συναδέρφους και φίλους Θανάση Βούλγαρη, Μιχάλη Κατσικούλη και Περικλή Συμεωνίδη οι οποίοι, με τις γνώσεις τους, με βοήθησαν σε κάποια εξειδικευμένα σημεία της εργασίας.

1 Πρόλογος

Σε αυτήν τη διπλωματική εργασία θα παρουσιαστεί ένα σύστημα ασφαλείας, αποτελούμενο από δίκτυο αισθητήρων. Ο κύριος αισθητήρας είναι ο περιβαντολλογικός, ο οποίος είναι πολυ-αισθητήρας που μετράει οκτώ διαφορετικά φυσικά μεγέθη: μικροσωματίδια τεσσάρων διαφορετικών διατομών (1, 2.5, 4 και 10 μικρόμετρα), πτητικές οργανικές ενώσεις (VOC), οξειδία του αζώτου (NO και NO₂), θερμοκρασία και υγρασία. Εναλλακτική περίπτωση είναι και ο απλός αισθητήρας θερμοκρασίας, ο οποίος έχει υλοποιηθεί, ώστε να προσδώσει ποικιλία αισθητήρων στο σύστημα. Με την υποστήριξη πολλαπλών αισθητήρων η εφαρμογή γίνεται πιο σύνθετη, με μεγαλύτερη υποδομή και πιο πολύπλοκη υλοποίηση, όμως προσδίδει ευελιξία στο σύστημα, φέρνοντάς το πολύ κοντά σε μια πραγματική εφαρμογή του εμπορίου.

Ο κεντρικός ελεγκτής του συστήματος είναι λογισμικό υλοποιημένο σε ηλεκτρονικό υπολογιστή (H/Y) και επικοινωνεί με τους αισθητήρες σε δίκτυο RS-485. Συλλέγει τις τιμές από τους αισθητήρες και ανάλογα ενημερώνει την κατάσταση του συστήματος με τα αντίστοιχα σφάλματα ή συναγερμούς. Οι αισθητήρες διαθέτουν οπτικοακουστικά μέσα (βομβητές σε ρόλο σειρήνας και LED ένδειξης των καταστάσεων του συστήματος), ώστε ο χρήστης να ειδοποιείται σε περίπτωση κρίσιμων συμβάντων και σφαλμάτων.

Το πρόγραμμα στον H/Y περιλαμβάνει και ένα δεύτερο κομμάτι που λειτουργεί παράλληλα και είναι ο διακομιστής (server), ο οποίος στέλνει όλα τα δεδομένα σε ιστοσελίδα όπου, μέσω άλλου λογισμικού (αυτό του client), εμφανίζονται στον χρήστη. Ο τελευταίος μπορεί στο τοπικό δίκτυο και μέσω της ιστοσελίδας, να δει όλα όσα χρειάζεται για το σύστημα και να στείλει και κάποιες απλές εντολές (πχ «καθαρισμός αρχείου συμβάντων»).

Η εργασία έχει ως κεντρικό πυρήνα την εφαρμογή αυτή κάθε αυτή, περιγράφοντας το hardware και το λογισμικό του αισθητήρα, το πρωτόκολλο επικοινωνίας, καθώς και το λογισμικό του κεντρικού ελεγκτή, του server και του client. Δίνεται έμφαση στην αρχιτεκτονική των συστημάτων και στην οργάνωση του προγράμματος, για τα οποία υπάρχει εκτενής περιγραφή σχετικά με τη μεθοδολογία σύμφωνα με την οποία δομήθηκαν (Clean Architecture).

Πέρα από την ανάλυση και την περιγραφή της κατασκευής, γίνεται αναφορά στα αέρια που ανιχνεύονται από την εφαρμογή και την επικινδυνότητά τους για την ανθρώπινη υγεία, για τα συστήματα ασφαλείας γενικότερα αλλά και πιο ειδικά τα συστήματα ανίχνευσης αερίων. Παρουσιάζεται επίσης μελέτη του ανταγωνισμού και συγκρίνεται η συγκεκριμένη κατασκευή με αυτά που προσφέρει ο ανταγωνισμός.

Σημαντικό στοιχείο είναι ότι η συγκεκριμένη κατασκευή δεν μπορεί να ανταγωνιστεί συστήματα ασφαλείας του εμπορίου. Για να συμβεί αυτό άλλωστε χρειάζεται ομάδα ανάπτυξης, η οποία να δουλέψει επάνω στην εφαρμογή για κάποια χρόνια, πραγματοποιώντας εξαντλητικές μελέτες επάνω στα υπό ανίχνευση αέρια. Απαιτούνται επίσης δοκιμές που απαιτούν ακριβό εξοπλισμό, που σε πολλές περιπτώσεις ούτε οι ίδιες οι εταιρίες δεν διαθέτουν, στέλνοντας τις συσκευές για δοκιμή σε εξωτερικά εργαστήρια. Όμως η επιτυχής λειτουργία του συστήματος όχι μόνο αποτελεί την απόδειξη της ιδέας (proof of concept), αλλά υλοποιεί και αυτό που στην αγορά ονομάζεται «Ελάχιστο Βιώσιμο Προϊόν» (Minimum Viable Product ή MVP). Ένα δοκίμιο κοινώς, που πριν κυκλοφορήσει επισήμως το

προϊόν στο εμπόριο μπορεί να δοθεί σε κάποιους συγκεκριμένους πελάτες για δοκιμή, ώστε να συλλεχτεί feedback και να βελτιωθεί το προϊόν προτού καν βγει στην αγορά. Κάτι τέτοιο είναι κοινή πρακτική και αρέσει τόσο στις κατασκευάστριες εταιρίες οι οποίες βελτιώνουν πρώιμα του προϊόν τους, αλλά και στους πελάτες, οι οποίοι έχουν λόγο και άποψη πριν καν βγει το προϊόν στην αγορά.

Το συγκεκριμένο σύστημα αποτελεί ιδιοκατασκευή. Τα θεωρητικά ηλεκτρονικά σχέδια (schematic) και τα σχέδια της πλακέτας (PCB) σχεδιάστηκαν από την αρχή για τις ανάγκες της εργασίας. Δεν θα βρει κάποιος τις πλακέτες έτοιμες στο εμπόριο, ούτε τα σχέδια έτοιμα στο internet. Φυσικά και χρησιμοποιήθηκαν αναφορές για τη σχεδίαση των ηλεκτρονικών, αλλά το τελικό αποτέλεσμα είναι προσαρμοσμένο στη συγκεκριμένη κατασκευή.

Στο τέλος της εργασίας παρατίθενται όλες οι αναφορές που χρησιμοποιήθηκαν. Στο κομμάτι της περιγραφής της κατασκευής προφανώς δεν υπάρχουν αναφορές αφού πρόκειται για ιδιοκατασκευή, όμως όλες οι πληροφορίες, καθώς και το θεωρητικό και το τεχνικό υπόβαθρο για να φτάσει το σύστημα στην ολοκλήρωσή του, παρατίθενται ευκρινώς εντός της εργασίας.

1.1 Σύντομη περιγραφή

Δίκτυο διευθυνσιοδοτούμενων αισθητήρων σε σύνδεση RS-485 για μέτρηση θερμοκρασίας, υγρασίας και επιπέδου συγκέντρωσης επικίνδυνων αερίων. Το πρόγραμμα ελέγχου βρίσκεται σε τοπικό διακομιστή, ο οποίος μέσω ιστοσελίδας, δίνει στον χρήστη τη δυνατότητα απομακρυσμένης πρόσβασης.

1.2 Αναλυτική Περιγραφή

Η εφαρμογή μπορεί να χωριστεί στις εξής ενότητες:

- Αισθητήρας
- Δίκτυο αισθητήρων
- Κεντρικός Ελεγκτής Συστήματος (ΚΕΣ)
- Διακομιστής (Server) τοπικού δικτύου
- Διεπαφή Χρήστη (User Interface ή UI)

Αισθητήρας

Ο αισθητήρας είναι ειδικής κατασκευής και διαθέτει προ-εγκατεστημένο λογισμικό με ειδικούς αλγόριθμους, ώστε οι αλλαγές της θερμοκρασίας να γίνονται αντιληπτές ταχύτατα, χωρίς να χάνεται η ακρίβεια των μετρήσεων. Η θερμοκρασία είναι σημαντική μέτρηση του αισθητήρα όχι μόνο για την ένδειξη αυτής, αλλά και γιατί χρησιμοποιείται και στους τύπους υπολογισμού της συγκέντρωσης των αερίων.

Δίκτυο αισθητήρων

Το σύστημα αισθητήρων θα είναι σε δίκτυο RS-485 και μέσω μετατροπέα USB/RS-485 θα συνδέεται στον Η/Υ μέσω της USB θύρας. Η επικοινωνία θα είναι single master, που σημαίνει ότι οι αισθητήρες θα απαντάνε αποκλειστικά στις αιτήσεις (request) της ΚΕΣ. Το δίκτυο θα δέχεται μέγιστο 250 συσκευές. Οι αισθητήρες θα είναι δύο τύπων: οι περιβαντολλογικοί και οι απλοί αισθητήρες θερμοκρασίας.

Κεντρικός Ελεγκτής Συστήματος (ΚΕΣ)

Είναι ένας τερματικός Η/Υ. Το λογισμικό θα επικοινωνεί με τους αισθητήρες και θα ενημερώνει κατάλληλα την κατάσταση του συστήματος (πχ κατάσταση ηρεμίας, κατάσταση συναγερμού κοκ). Ταυτόχρονα θα μπορεί να στέλνει και κάποιες εντολές προς τους αισθητήρες, εάν αυτό είναι απαραίτητο από την εφαρμογή.

Διακομιστής (Server) τοπικού δικτύου

Για λόγους απλότητας θα χρησιμοποιηθεί ο ίδιος Η/Υ στον οποίο θα βρίσκεται εγκατεστημένο το λογισμικό του ΚΕΣ. Ο διακομιστής θα είναι διαφορετικό πρόγραμμα από τον ΚΕΣ. Θα μιλάει με τον ΚΕΣ μέσω ειδικά σχεδιασμένων API, από την οποία θα παίρνει την κατάσταση του συστήματος και θα τα στέλνει μέσω του τοπικού δικτύου σε ιστοσελίδα (UI). Ταυτόχρονα θα μπορεί να παραλαμβάνει και εντολές από τον χρήστη, που θα έρχονται μέσω του UI.

Διεπαφή Χρήστη (User Interface -UI-)

Πρόκειται για ιστοσελίδα η οποία θα εμφανίζει στην οθόνη όλες τις πληροφορίες του συστήματος και θα δέχεται και εντολές από τον χρήστη. Επικοινωνεί μέσω HTML με τον τοπικό διακομιστή.

Καταστάσεις λειτουργίας συστήματος

Συναγερμός: τουλάχιστον ένας αισθητήρας βρίσκεται σε κατάσταση συναγερμού, που σημαίνει ότι κάποιο μετρούμενο φυσικό μέγεθος (πχ συγκέντρωση αερίου) ξεπέρασε το επιτρεπτό όριο.

Σφάλμα: κανένας αισθητήρας δεν βρίσκεται σε κατάσταση συναγερμού, αλλά τουλάχιστον ένας αισθητήρας βρίσκεται σε κατάσταση σφάλματος (πχ εσωτερικό σφάλμα αισθητήρα ή αισθητήρας αποσυνδεδεμένος από το δίκτυο).

Ηρεμία: κανένας αισθητήρας δεν βρίσκεται ούτε σε κατάσταση συναγερμού ούτε σε κατάσταση σφάλματος.

2 Ανίχνευση αερίων – γενικά στοιχεία

2.1 Αιωρούμενα σωματίδια

Ο όρος «αιωρούμενα σωματίδια» (particulate matter) ή «μικροσωματίδια» αναφέρεται σε διάφορα μεγέθη, συμπεριλαμβανομένων των PM₁₀, PM_{2.5}, και των εξαιρετικά λεπτών PM_{0.1}. Η σύσταση αυτών των σωματιδίων και οι επιπτώσεις τους στην ανθρώπινη υγεία αποτελούν αντικείμενο συνεχούς έρευνας και συζήτησης. [1]

Τα μικροσωματίδια αποτελούν ένα σύνθετο μείγμα με ποικιλία στην προέλευση (ανθρωπογενής ή γεωγονική) και τα φυσικοχημικά χαρακτηριστικά (πχ διαλυτότητα). Με πρακτικούς όρους το μέγεθος των σωματιδίων εκφράζεται συνήθως σε αεροδυναμική διάμετρο. Τα μικροσωματίδια τύπου PM_{2.5} έχουν διάμετρο 2.5 μικρόμετρα, τα PM₁₀ έχουν διάμετρο 10 μικρόμετρα κοκ. Οι μελέτες σχετικά με την επίπτωση των αιωρούμενων σωματιδίων στην ανθρώπινη υγεία έχουν να κάνουν συνήθως με τη θνητότητα, την επίδραση στις ζωτικές λειτουργίες, τις εισαγωγές σε νοσοκομεία κα. [1]

Σύμφωνα με την Υπηρεσία Περιβάλλοντος των Ηνωμένων Πολιτειών (United States Environmental Protection Agency ή EPA), μελέτες έχουν δείξει συσχέτιση των μικροσωματιδίων με καρδιαγγειακά προβλήματα όπως αρρυθμία και έμφραγμα, αλλά και αναπνευστικά προβλήματα όπως άσθμα και βρογχίτιδα. Το μέγεθος των μικροσωματιδίων σχετίζεται με τις εκάστοτε επιπτώσεις, με τα λεπτά σωματίδια (fine particles), μικρότερα ή ίσα του P_{2.5}, να αποτελούν τον μεγαλύτερο κίνδυνο, επηρεάζοντας την καρδιά και τους πνεύμονες. [2]

2.2 Πτητικές οργανικές ενώσεις (VOC)

Οι πτητικές οργανικές ενώσεις (Volatile Organic Compounds ή VOC) αποτελούν πιθανό κίνδυνο για την υγεία, καθώς βρίσκονται συνήθως σε εσωτερικούς χώρους. Η έκθεση σε αυτές τις ενώσεις, που εκπέμπονται από καθημερινά αντικείμενα όπως οικοδομικά υλικά και χρώματα, μπορεί να οδηγήσουν σε ανεπιθύμητες ενέργειες: από ήπιο ερεθισμό έως σοβαρές ασθένειες όπως καρκίνος. Τα στοιχεία υποδεικνύουν πράγματι σύνδεση με αναπνευστικά συμπτώματα και ερεθισμό των αισθητηρίων οργάνων. [3]

2.3 Οξείδια του αζώτου (NO_x)

Τα οξείδια του αζώτου (Nitrogen Oxides - NO_x) είναι υποπροϊόντα καύσεων σε υψηλές θερμοκρασίες και απελευθερώνουν δηλητήρια στην ατμόσφαιρα. Οι αρνητικές επιπτώσεις αφορούν την ανθρώπινη υγεία, το περιβάλλον και τα βιολογικά οικοσυστήματα. [4]

Ο όρος «NO_x» αναφέρεται στο μονοξείδιο του αζώτου (NO) και το διοξείδιο του αζώτου (NO₂), τα οποία εμφανίζονται κυρίως σε αστικές περιοχές, αφού είναι παράγωγα των μηχανών εσωτερικής καύσης. Η παρατεταμένη έκθεση σε αυξημένα επίπεδα NO_x μπορεί να οδηγήσει σε δηλητηρίαση και φλεγμονώδεις διεργασίες στον ανθρώπινο οργανισμό. [5]

2.4 Ανίχνευση αερίων

2.4.1 Ιστορική αναδρομή

Ιστορικά η ανάγκη για ανίχνευση αερίων εκδηλώθηκε για πρώτη φορά στα ορυχεία, όταν οι άνθρωποι αντιλήφθηκαν το πόσο επιβλαβή μπορεί να είναι κάποια αέρια και ειδικά το μεθάνιο, το οποίο συν τοις άλλοις είναι και εύφλεκτο. Η ανίχνευση εκείνη την εποχή γινόταν από τους ίδιους τους ανθρώπους με τη χρήση δάδας, ώστε να δοκιμαστεί εάν υπάρχουν αναφλέξεις. Στη συνέχεια χρησιμοποιήθηκαν καναρίνια, τα οποία τοποθετούνταν εντός του ορυχείου, ώστε από τη μετέπειτα υγεία τους να διαπιστωθεί εάν εκτέθηκαν ή όχι σε επιβλαβή αέρια. Τις βιομηχανίες εξόρυξης ακολούθησαν και άλλες βιομηχανίες όπως αυτές της επεξεργασίας λυμάτων και τα διυλιστήρια. Οι πρώτες πραγματικές συσκευές παρακολούθησης ήταν με βαλβίδα που χρησιμοποιούσε αισθητήρα καταλυτικής διάχυσης και η οποία έδειχνε πόσο μεθάνιο υπάρχει στην ατμόσφαιρα. Η συσκευή για να λειτουργήσει και να πάρει μέτρηση, χρειαζόταν κάποιος άνθρωπος να πατήσει το κουμπί. [6]

2.4.2 Σύγχρονα συστήματα ανίχνευσης

Τα μοντέρνα συστήματα παρακολουθούν τα επίπεδα έκθεσης στα επικίνδυνα αέρια τόσο σε βραχυπρόθεσμο επίπεδο, όσο και σε βάθος χρόνου, σημαίνοντας συναγερμό όταν τα επίπεδα συγκέντρωσης ξεπεράσουν το επιτρεπτό όριο. Οι αισθητήρες έχουν ελαχιστοποιήσει το μέγεθος τους, συλλέγουν δεδομένα για στατιστική ανάλυση και έχουν δημιουργηθεί συστήματα παρακολούθησης πραγματικού χρόνου, τα οποία παρακολουθούν ολόκληρα δίκτυα αισθητήρων, που επικοινωνούν ενσύρματα ή ασύρματα. [7]

2.5 Τύποι αισθητήρων ανίχνευσης αερίων

Υπάρχουν διάφοροι τύποι αισθητήρων που μπορούν να χρησιμοποιηθούν. Κάποιοι δημοφιλείς παρουσιάζονται παρακάτω.

2.5.1 Καταλυτικοί αισθητήρες

Από μία συγκέντρωση αερίου και πάνω, δημιουργούνται μικρές «εκρήξεις» στο εσωτερικό του αισθητήρα, που προκαλούν μεταβολή στην αντίσταση εξόδου του. Με τη χρήση γέφυρας Wheatstone μπορούμε να μετρήσουμε την αλλαγή στην αντίσταση αυτή και συνεπώς και την αλλαγή στη συγκέντρωση του αερίου. Οι τύποι των καταλυτικών αισθητήρων είναι τα pellistor και οι θερμοηλεκτρικοί αισθητήρες. [8]

2.5.2 Αισθητήρες θερμικής αγωγιμότητας

Χρησιμοποιούνται για ανίχνευση αερίων μεγαλύτερης θερμικής αγωγιμότητας από τον αέρα, όπως το μεθάνιο και το υδρογόνο. Η αρχή λειτουργίας τους σχετίζεται με τη μεταφορά θερμότητας από ένα θερμό σώμα, προς ένα ψυχρότερο. Η έξοδος τους είναι μεταβλητή αντίσταση, η οποία μπορεί να μετρηθεί με γέφυρα Wheatstone. Υπάρχουν δύο διαφορετικές εκδοχές του αισθητήρα αυτού. Η μία εκδοχή βασίζεται στο ότι κατά την παρουσία του αερίου

υπάρχει μεταβολή της θερμοκρασίας της αντίστασης εξόδου, εξαιτίας της θερμικής αγωγιμότητας του αερίου υπό ανίχνευση, σε σχέση με κάποιο αέριο αναφοράς (συνήθως ο ατμοσφαιρικός αέρας). Ο δεύτερος τύπος αισθητήρα δεν χρησιμοποιεί αέριο αναφοράς, αλλά χρησιμοποιεί δύο στοιχεία, ένα θερμό και ένα ψυχρό, ώστε η αντίσταση εξόδου να μεταβάλλεται με τη μεταφορά θερμότητας από το ένα στοιχείο στο άλλο, κατά την παρουσία του υπό ανίχνευση αερίου. [8]

2.5.3 Ηλεκτροχημικοί αισθητήρες

Παρέχουν έξοδο ηλεκτρικού σήματος, το οποίο εξαρτάται από τη συγκέντρωση του υπό ανίχνευση αερίου. Στο εσωτερικό τους πραγματοποιείται χημική αντίδραση κατά την παρουσία του αερίου, η οποία είναι ανάλογη με τη συγκέντρωσή του. [8]

Οι τύποι αυτού του αισθητήρα είναι ο αισθητήρας τύπου αμπερομέτρου, όπου εφαρμόζεται σταθερή τάση ώστε να παρθεί έξοδος ρεύματος και ο αισθητήρας τύπου ποτενσιόμετρου, όπου ανάλογα με τη συγκέντρωση του αερίου, εμφανίζεται διαφορά δυναμικού στα δύο άκρα ενός ηλεκτροδίου. [8]

2.5.4 Οπτικοί αισθητήρες

Περιέχουν στο εσωτερικό τους έναν πομπό και έναν δέκτη υπέρυθρων (οπτοδίοδοι). Εκμεταλλευόμενοι το φαινόμενο της διάθλασης κατά την παρουσία ορατού αερίου, αλλάζει το ποσό της ακτινοβολίας που φτάνει στον δέκτη από τον πομπό, ώστε ο αισθητήρας να αντιληφθεί τη συγκέντρωση του υπό ανίχνευση αερίου. Υπάρχουν δύο ειδών αισθητήρων τέτοιου τύπου. Ο αισθητήρας διπλής δέσμης χρόνου (time-double beam) ο οποίος χρησιμοποιεί έναν πομπό και δύο δέκτες, ενώ ο διπλής δέσμης χρόνου (space-double beam) χρησιμοποιεί δύο πομπούς και έναν δέκτη. [8]

2.5.5 Αισθητήρες ημιαγωγού

Λειτουργούν με βάση την αρχή της αναστρέψιμης διαδικασίας της απορρόφησης του αερίου, στην επιφάνεια του θερμαινόμενου οξειδίου. Η απορρόφηση του αερίου έχει ως αποτέλεσμα την αλλαγή της ηλεκτρικής αντίστασης του οξειδίου. Διαχωρίζονται σε αισθητήρες τύπου η για ανίχνευση αερίων όπως το υδρογόνο, το μεθάνιο και το μονοξείδιο του άνθρακα και τύπου ρ για ανίχνευση των οξειδωτικών αερίων. [8]

2.5.6 Αισθητήρες ακουστικού κύματος

Ο μηχανισμός ανίχνευσης είναι ένα ακουστικό κύμα. Κατά την παρουσία του υπό ανίχνευση αερίου, αλλάζουν τα χαρακτηριστικά της διαδρομής μετάδοσης του κύματος, με συνέπεια να αλλάζει η συχνότητα και η φάση του. Με τη χρήση μορφομετατροπέα (transducer) η απόκριση του σήματος μετατρέπεται σε ηλεκτρικό σήμα. Ένας δημοφιλής τύπος αυτού του αισθητήρα είναι ο ακουστικός αισθητήρας επιφάνειας, όπου ένα ηλεκτρικό σήμα μετατρέπεται σε ακουστικό και στη συνέχεια ξανά σε ηλεκτρικό. Ο χρόνος μετάδοσης από το ένα άκρο του μορφομετατροπέα στο άλλο, εξαρτάται από την παρουσία και τη συγκέντρωση

του υπό ανίχνευση αερίου. Θεωρούνται οι πιο σύγχρονοι αισθητήρες και η «επόμενη γενιά» στην ανίχνευση αερίων. [8]

3 Συστήματα παρακολούθησης και ελέγχου

3.1 Ηλεκτρικές συνδέσεις

Είναι προφανές ότι τα συστήματα που περιγράφονται ανιχνεύουν εύφλεκτα, σε πολλές περιπτώσεις, αέρια ή/και ορατό καπνό από πυρκαγιά. Συνεπώς τα καλώδια που χρησιμοποιούνται πρέπει να εξασφαλίζουν το γεγονός ότι κατά τη διάρκεια ενός τέτοιου συμβάντος, θα αντέξουν σε μεγάλες θερμοκρασίες και άρα πρέπει να είναι πυρίμαχα. Στην εικόνα 1 φαίνεται ένα τέτοιου τύπου καλώδιο από την εταιρία Accordia.



Εικόνα 1 - Το πυρίμαχο καλώδιο RH120 FE180 [9]

3.2 Ανιχνευτές

Οι ανιχνευτές είναι οι κατασκευές που φέρουν επάνω τους τα αισθητήρια. Υπάρχουν διάφοροι τύποι ανιχνευτών, αλλά συνήθως οι ανιχνευτές που χρησιμοποιούνται είναι αντιεκρηκτικού τύπου (ATEX), οι οποίοι είναι ειδικής κατασκευής και ογκώδεις, ώστε να αντέχουν σε εκρήξεις και πυρκαγιές. Στην εικόνα 2 φαίνεται ένας ανιχνευτής τέτοιου τύπου, ο Sensepoint XCD της Honeywell, του μεγαλύτερου ίσως κατασκευαστή και εμπόρου συστημάτων ασφαλείας παγκοσμίως.



Εικόνα 2 - Ο ανιχνευτής αερίων Sensepoint XCD [10]

Στην εικόνα 3 παρατίθενται τα τεχνικά χαρακτηριστικά του Sensepoint XCD. Ενδιαφέρον παρουσιάζει ο τρόπος που ο συγκεκριμένος αισθητήρας μπορεί να μετρήσει πολλαπλούς τύπους αερίων, με τη χρήση του πρωτοκόλλου 4-20mA. Ο ανιχνευτής δέχεται κάψουλες, ανάλογα με το τι αέριο θέλει να μετρήσει ο πελάτης. Εάν κανείς προσαρμόσει την κάψουλα μεθανίου, ο ανιχνευτής θα ανιχνεύει μεθάνιο, με την κάψουλα προπανίου θα ανιχνεύει προπάνιο κοκ. Η ακρίβεια του ανιχνευτή είναι είτε σε ppm (parts per million) για τα μη εκρηκτικά αέρια είτε σε LEL (Lower Explosion Limit) στην ανίχνευση εκρηκτικών αερίων.

Detectable Gases and XCD Sensor Performance									
Gas	User Selectable Full Scale Range	Default Range	Steps	User Selectable Cal Gas Range	Default Cal Point	Response Time (T90) Secs	Accuracy	Operating Temperature*	
								Min	Max
Electrochemical Sensors									
Oxygen	25.0%Vol. only	25.0%Vol.	n/a	20.9%Vol. (Fixed)	20.9%Vol.	<30	<+/-0.5%Vol.	-20°C / -4°F	55°C / 131°F
Hydrogen Sulphide	10.0 to 100.0ppm	50.0ppm	1ppm	30 to 70% of selected full scale range	25ppm	<50	<+/-1ppm	-20°C / -4°F	55°C / 131°F
Carbon Monoxide	100 to 1,000ppm	300ppm	100ppm		100ppm	<30	<+/-6ppm	-20°C / -4°F	55°C / 131°F
Hydrogen	1,000ppm only	1,000ppm	n/a		500ppm	<65	<+/-25ppm	-20°C / -4°F	55°C / 131°F
Nitrogen Dioxide	10.0 to 50.0ppm	10.0ppm	5.0ppm		5.0ppm	<40	<+/-3ppm or +/-20%	-20°C / -4°F	55°C / 131°F
Catalytic Bead Sensors				25 to 95% of selected full scale range					
Flammable 1 to 8*	20 to 100%LEL	100%LEL	10%LEL		50%LEL	<25	<+/-1.5%LEL	-20°C / -4°F	55°C / 131°F
Infrared Sensors				30 to 70% of selected full scale range					
Methane	20 to 100%LEL	100%LEL	10%LEL		50%LEL	<30	<+/-1.5%LEL	-20°C / -4°F	55°C / 131°F
Propane	20 to 100%LEL	100%LEL	10%LEL		50%LEL	<30	<+/-1.5%LEL	-20°C / -4°F	55°C / 131°F
Carbon Dioxide	2.00%Vol.	2.00%Vol.	n/a		1.00%Vol.	<30	<+/-0.04%Vol.	-20°C / -4°F	55°C / 131°F

Εικόνα 3 - Τα τεχνικά χαρακτηριστικά του Sensepoint XCD [10]

3.2.1 Αναλογικοί ανιχνευτές

Ιδιαίτερη μνεία θα πρέπει να γίνει στους αναλογικούς ανιχνευτές, οι οποίοι ακόμα και στη σημερινή εποχή έχουν τη μερίδα του λέοντος στην αγορά. Ένας σημαντικός λόγος είναι ότι υπάρχουν ήδη εγκατεστημένα συστήματα με εκατομμύρια αναλογικούς ανιχνευτές παγκοσμίως, που σημαίνει ότι η συγκεκριμένη μέθοδος είναι αποδεδειγμένα αποτελεσματική και λειτουργική. Υπάρχουν όμως και άλλοι καλοί λόγοι για τη χρήση τους και ένας από αυτούς είναι η χρήση του πρωτοκόλλου 4-20mA, το οποίο προσφέρει ανοχή στις παρεμβολές, αλλά και σύνδεση μόνο με δύο καλώδια, αφού το καλώδιο της τροφοδοσίας χρησιμοποιείται και ως καλώδιο σήματος. Η τεχνολογία προχωράει και τα συστήματα εξελίσσονται, αλλά είναι βέβαιο ότι τα 4-20mA θα είναι για πολύ καιρό ακόμα το βασικό πρωτόκολλο στους αναλογικούς ανιχνευτές αερίων.

3.3 Κεντρικές μονάδες

Για να συσταθεί ένα δίκτυο αισθητήρων χρειάζεται μια κεντρική μονάδα επεξεργασίας, η οποία θα συλλέγει τα δεδομένα από τους αισθητήρες, θα τα αξιολογεί και ανάλογα θα ενημερώνει την κατάσταση του συστήματος και θα οδηγεί κατάλληλα τα ενδεικτικά και τις σειρήνες. Επιπλέον οι μοντέρνοι πίνακες προσφέρουν απομακρυσμένη πρόσβαση μέσω GSM, WiFi ή Ethernet, προγράμματα σε H/Y για ρύθμιση του συστήματος ενώ και η υποστήριξη μέσω εφαρμογής σε κινητό, μετατράπηκε με τον καιρό από πολυτέλεια σε ανάγκη.

Ενδεικτικά παρατίθενται ως αναφορά οι πίνακες της σειράς BS-3xx του μεγαλύτερου κατασκευαστή και εξαγωγέα συστημάτων ασφαλείας στην Ελλάδα, της Olympia Electronics (εικόνα 4).



Εικόνα 4 - Ο πίνακας ανίχνευσης αερίων BS-3xx [11]

Τα χαρακτηριστικά της σειράς BS-3xx φαίνονται στην εικόνα 5. Η διαφοροποίηση στα χαρακτηριστικά των πινάκων της ίδιας σειράς, έγκειται στις διαθέσιμες εισόδους του πίνακα, όπου για μεγάλες εγκαταστάσεις παρέχεται πίνακας με 16 εισόδους. Επιπλέον προσφέρονται και έξοδοι με ρελέ, ώστε να μπορούν να συνδεθούν ηχητικές και ενδεικτικές διατάξεις για ειδοποιήσεις σε συγκεκριμένα συμβάντα, όπως για παράδειγμα σε συναγερμό.

	BS-304	BS-308	BS-312	BS-316
	Gas detection control panel up to 16 inputs			
Power Supply	100-240V AC / 50-60Hz			
Consumption	23W/60VA			
Battery type	12V / 7Ah Pb or 2X12V/7Ah			
Charge Circuit	Power supply 13.8V / 500mA			
Inputs	4+AUX	8+AUX	12+AUX	16+AUX
Programmable relays (5A/230V)	5	9	13	17
Outputs	Up to 16 programmable relays (5A/230VAC). 1 fault output 20V/300mA, 1 AUX			
Wrap protection class	IP30			
Operating temperature range	5 to 40 °C			
Humidity	Up to 95% humidity			
Dimensions	445 x 345 x 95 mm			
Weight	6160 gr			
In compliance with	EN50270 (TYPE I), EN61000-6-3(Use of Ferrite required), EN61000-3-2, EN61000-3-3, EN62368-1:2014			
Warranty	2 years			

Εικόνα 5 - Τα τεχνικά χαρακτηριστικά του πίνακα ανίχνευσης αερίων BS-3xx [11]

3.4 Περιφερειακές μονάδες εισόδου - εξόδου

Εκτός από τους ανιχνευτές και τον κεντρικό πίνακα, τα συστήματα ανίχνευσης αερίων προσφέρουν και κάποια ακόμα περιφερειακά. Τα πιο συνηθισμένα είναι οι κάρτες εισόδων

– εξόδων, τα GSM modules, οι σειρήνες, τα ενδεικτικά και τα μπουτόν πανικού. Συνήθως οι εταιρίες προσφέρουν κάρτες εξόδων με ρελέ, ώστε οι πελάτες να μπορούν να προγραμματίζουν από τον πίνακα το πότε τα ρελέ θα ανοιγοκλείνουν. Για παράδειγμα ένα ρελέ μπορεί να σπλίζει σε περίπτωση συναγερμού ώστε να συνδεθεί με σειρήνα, ενώ ένα άλλο ρελέ να σπλίζει σε περίπτωση σφάλματος, ώστε να συνδεθεί ένα ενδεικτικό. Οδήγηση GSM από εξόδους μπορεί να πραγματοποιηθεί σε όλες τις παραπάνω περιπτώσεις, στέλνοντας μάλιστα διαφορετικό SMS, ανάλογα με το ποια είσοδος θα ενεργοποιηθεί (συναγερμός ή σφάλμα).

Λιγότερο χρησιμοποιούμενες αλλά χρήσιμες είναι και οι μονάδες εισόδου, όπου ο χρήστης μπορεί να προσθέσει μπουτόν πανικού, ώστε να το πατάει σε περίπτωση που υπάρξει κάποια επείγουσα κατάσταση για να ηχήσει κάποια σειρήνα, πχ διότι ξέσπασε μόλις μια μικρή εστία φωτιάς, την οποία ο πίνακας χρειάζεται λίγο ακόμα χρόνο για να ανιχνεύσει.

Οι παραπάνω συσκευές μαζί με τους ανιχνευτές, τον κεντρικό πίνακα και την ψηφιακή υποστήριξη (H/Y, web pages, εφαρμογές σε κινητό), συνθέτουν ένα πλήρες σύστημα ανίχνευσης αερίων.

3.5 Διευθυνσιοδοτούμενα συστήματα - πυρανίχνευση

Στα αναλογικά συστήματα αναγνωρίζονται οι συναγερμοί και τα σφάλματα από τον πίνακα, αλλά όχι με ακρίβεια σημείου (μεμονωμένου ανιχνευτή), αλλά ομάδας. Όπου «ομάδα» ο όρος που χρησιμοποιείται στα συστήματα ασφαλείας είναι «ζώνη». Οι ζώνες είναι διαφορετικές γραμμές που φεύγουν από τον πίνακα προς διαφορετικά σημεία της εγκατάστασης, επάνω στις οποίες συνδέονται οι ανιχνευτές. Σε ένα αναλογικό σύστημα, εάν ανιχνευτεί κάποιο συμβάν σε κάποια ζώνη δεν είναι γνωστό στον πίνακα ποιος ανιχνευτής έδωσε το συμβάν αυτό. Συνεπώς ο πίνακας δεν θα μπορεί να γνωρίζει για παράδειγμα σε ένα ξενοδοχείο ότι δόθηκε συναγερμός από τον ανιχνευτή «δωμάτιο 32», αλλά από τη ζώνη «όροφος 3». Υπάρχουν και σημειακά συστήματα ανίχνευσης αερίων, αλλά χρησιμοποιούνται για μεγάλες εγκαταστάσεις όπου απαιτείται εκτεταμένη ανίχνευση σε πολλούς διαφορετικούς χώρους, ίσως και διαφορετικά κτήρια.

Τα συστήματα πυρανίχνευσης λόγω των αυστηρών προτύπων και τις απαιτήσεις της αγοράς οφείλουν να ξεπεράσουν τον περιορισμό των αναλογικών συστημάτων (αν και υπάρχουν συμβατικοί πίνακες πυρανίχνευσης για μικρές εγκαταστάσεις), συνθέτοντας αυτό που ονομάζεται «διευθυνσιοδοτούμενο σύστημα». Σε ένα τέτοιο σύστημα ο πίνακας θα ενημερώνεται ακριβώς από ποιο δωμάτιο ήρθε το συμβάν. Οι ζώνες συνεχίζουν να υπάρχουν, αλλά μόνο ως λογικός διαχωρισμός και χωρίς φυσικούς περιορισμούς. Μπορεί, για παράδειγμα, σε ένα ξενοδοχείο να εκτείνεται μια γραμμή (βρόχος ή loop) από τον πίνακα προς τον όροφο 1 και ένα άλλο loop προς τον όροφο 2. Μέσω κατάλληλου προγραμματισμού θα μπορούσαν να υπάρχουν δύο ζώνες («όροφος 1» και «όροφος 2») ή και μία ζώνη («όροφοι 1 και 2»). Το ίδιο ισχύει και για τους ασύρματους αισθητήρες, οι οποίοι αναγνωρίζονται επίσης ως «σημεία». Είναι καθαρά θέμα προγραμματισμού του πίνακα για το ποια σημεία θα εντάξει στην κάθε ζώνη, σε αντίθεση με τα αναλογικά συστήματα, όπου ο όρος «ζώνη» είναι ταυτισμένος με την κοινή ηλεκτρολογική σύνδεση των αισθητήρων.

Ένα τέτοιο σύστημα πυρανίχνευσης είναι το CF1000 της Eaton (εικόνα 6).



Εικόνα 6 - Ο πίνακας ανίχνευσης αερίων BS-3xx [12]

Ο CF1000 είναι ο πιο μικρός πίνακας της σειράς, αφού έχει ένα loop και δέχεται 200 σημεία. Το πρότυπο με το οποίο όλοι οι πίνακες πυρανίχνευσης πρέπει να συμμορφώνονται είναι το EN54 σε όλες τις επεκτάσεις του (EN54-2, EN54-4 κοκ). Σε περίπτωση που κάποιος πίνακας πυρανίχνευσης δεν είναι πιστοποιημένος με αυτό το πρότυπο όχι απλά δεν είναι αποδεκτός από την αγορά, αλλά απαγορεύεται να πουληθεί. Επίσης είναι υποχρεωτικό από τους κανονισμούς αυτούς, να υπάρχει μπαταρία και μάλιστα να αντισταθμίζεται η φόρτισή της με τη θερμοκρασία, ώστε να εγγυάται έναν ελάχιστο χρόνο ζωής.

Code	CF1100
Description	1 loop control panel
Standards	EN54 Pt2,1997, A1:2006, EN54 Pt4, 1997 A1:2002, A2:2006, EN54 Pt13: 2005
Specification	
Number of loops	1
Addresses per loop	200
Number of conventional sounder circuits	2 monitored for open and short circuit (max 1.5A combined)
Auxiliary fire routing equipment output (monitored)	24V dc 30mA (max)
Auxiliary fire protection equipment output (monitored)	24V dc 30mA (max)
Auxiliary fault routing equipment output (monitored)	12V dc 30mA (max)
System operating voltage	24V dc (nom)
Mains input supply	230V ac +10% / -15%
Class change facility	Terminals for connection of external contacts, can also be instigated via input interface
Auxiliary relay	1 set of changeover contacts operate in event of fire condition
Output ports	RS485, RS232 for connection of repeaters etc
Standby duration	Dependant on loop loading and battery configuration
Battery	2 x 12V 7Ah

Εικόνα 7 - Τα τεχνικά χαρακτηριστικά του πίνακα πυρανίχνευσης CF1100 [12]

3.6 Δικτύωση πινάκων

Κάποιοι από τους πίνακες ανίχνευσης αερίων μπορούν να έχουν προγραμματισμό ώστε κάποιες είσοδοι να δεχτούν και ανιχνευτές καπνού, αλλά το σύνηθες σενάριο είναι να προσαρμόζονται ανιχνευτές αερίων σε πίνακες πυρανίχνευσης. Στη δεύτερη περίπτωση οι ανιχνευτές είναι συνήθως «αυτόνομοι», που σημαίνει ότι ανιχνεύουν μόνοι τους το πότε η συγκέντρωση ξεπεράσει το επιτρεπτό όριο, οπότε και αλλάζουν την κατάσταση της εξόδου τους, ώστε ο πίνακας να ενημερωθεί και να κάνει τις απαραίτητες ενέργειες.

Η πιο σύνθετη αλλά και πιο ολοκληρωμένη εγκατάσταση είναι όταν υπάρχουν ξεχωριστοί πίνακες πυρανίχνευσης και ανίχνευσης αερίων. Ο πίνακας ανίχνευσης αερίων συνδέεται στον πίνακα πυρανίχνευσης ως περιφερειακό και αυτό διότι οι πίνακες πυρανίχνευσης είναι πιο μεγάλες και σύνθετες κατασκευές και υπόκεινται σε εξαιρετικά αυστηρούς κανόνες και πρότυπα. Τέτοιες περιπτώσεις σύνθετων διευθυνσιοδοτούμενων συστημάτων μπορεί να συναντήσει κανείς σε βενζινάδικα ή σε βιομηχανίες εύφλεκτων και επικίνδυνων υλικών, όπου είναι εξαιρετικά κρίσιμο να ανιχνεύονται τα εύφλεκτα αέρια, αλλά και ο ορατός καπνός που παράγεται από τα προϊόντα καύσης της πυρκαγιάς. Μάλιστα σε μεγάλες εγκαταστάσεις ένας πίνακας πυρανίχνευσης δεν είναι αρκετός και τοποθετούνται παραπάνω, έχοντας όμως την «υποχρέωση» να συνεργάζονται μεταξύ τους, συνθέτοντας ένα τελικό σύστημα που αποτελείται από πολλά υποσυστήματα. Αυτή η εγκατάσταση επιτυγχάνεται με τη δικτύωση πινάκων, οι οποίοι συνδέονται όλοι και επικοινωνούν στο ίδιο καλώδιο, ώστε τα συμβάντα οποιουδήποτε σημείου σε οποιοδήποτε πίνακα, να μπορούν να φαίνονται σε όλους τους πίνακες του δικτύου και να οδηγούνται κατάλληλα οι σειρήνες και τα ενδεικτικά σε όλη την εγκατάσταση και όχι μόνο στον πίνακα ο οποίος έδωσε το συμβάν.

3.7 Δίκτυο περιβαντολλογικών αισθητήρων

Έχοντας παραθέσει όλο το απαραίτητο υπόβαθρο για τα συστήματα ανίχνευσης αερίων, εξετάζεται ο ανταγωνισμός και στα συστήματα δικτύων περιβαντολλογικών αισθητήρων. Το σύστημα που κατασκευάστηκε για τις ανάγκες της παρούσας εργασίας, θα πρέπει να κρατάει επαφή με τα υπάρχοντα συστήματα της αγοράς. Όμως η έρευνα έδειξε ότι δεν υπάρχουν παρόμοια συστήματα στην αγορά.

Μέσω διαδικτυακής αναζήτησης εμφανίζονται πολλά αποτελέσματα για τη φράση «environmental sensor networks», αλλά στην πραγματικότητα κανένα από αυτά δεν είναι κοντά με τη συγκεκριμένη κατασκευή. Κάποια από αυτά συλλέγουν μετεωρολογικά δεδομένα και τα αποθηκεύουν σε βάσεις δεδομένων για επεξεργασία και μελέτη των στοιχείων, κάποια άλλα διαθέτουν αισθητήρες χαμηλής κατανάλωσης (και συνεπώς φιλικούς προς το περιβάλλον), αλλά ο όρος «environmental sensor networks» που χρησιμοποιούν δεν αναφέρεται στον τύπο του αισθητήρα, αλλά στον τύπο του συστήματος.

Ότι πιο κοντινό βρέθηκε είναι ένα το σύστημα εξωτερικού χώρου «Black Carbon» (εικόνα 8).



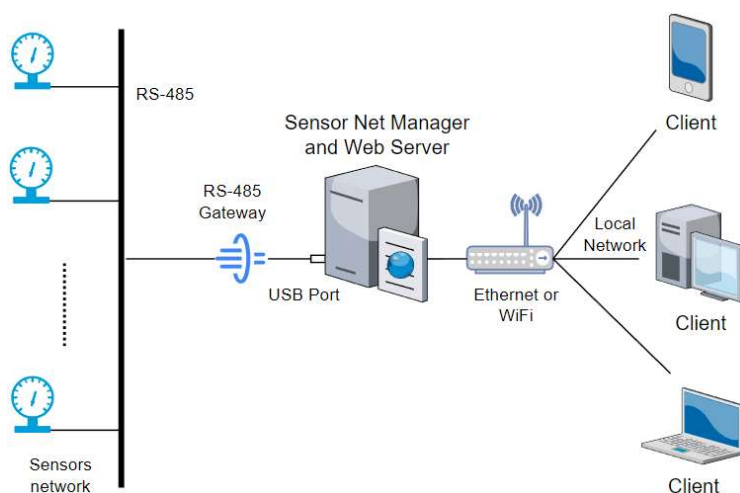
Εικόνα 8 - Το module ανίχνευσης εξωτερικού χώρου «Black Carbon» [13]

Το σύστημα «black carbon» διαφέρει παρόλα αυτά από το σύστημα που παρουσιάζεται στην παρούσα εργασία, αφού είναι ένα σύστημα εξωτερικού χώρου και ασύρματο, με ότι αυτό συνεπάγεται. Επίσης στέλνει απευθείας στο cloud, κάτι που σημαίνει ότι επιβάλλεται να έχει συγκεκριμένα επίπεδα διαδικτυακής ασφάλειας για να το δεχτεί η αγορά. Πρόκειται λοιπόν όχι τόσο για ένα «δίκτυο αισθητήρων», όσο για μεμονωμένους αισθητήρες που στέλνουν στο cloud. Σε κάθε περίπτωση το σύστημα δεν έχει βγει ακόμα στην αγορά και αναμένεται, σύμφωνα με την κατασκευάστρια εταιρία, να βγει το δεύτερο τρίμηνο του 2024. Δεν δίνονται περισσότερες πληροφορίες για τα λογισμικά απομακρυσμένης πρόσβασης και συνεπώς δεν μπορούν να γίνουν και πολλές συγκρίσεις με το σύστημα της παρούσας εργασίας.

4 Σύστημα GreenLink Pro

4.1 Βαθμίδες συστήματος

Φαίνονται στην εικόνα 9.



Εικόνα 9 - Βαθμίδες συστήματος

4.1.1 Δίκτυο αισθητήρων RS-485

Όλοι οι αισθητήρες επικοινωνούν επάνω στην ίδια γραμμή, που είναι το διαφορικό ζεύγος καλωδίων του RS-485. Δεν είναι απαραίτητο ο κάθε αισθητήρας να συνδέεται απευθείας με το gateway σε δίκτυο αστέρα (για την ακρίβεια θα πρέπει να αποφεύγεται). Το RS-485 δίνει την ευελιξία να μπορούν οι κόμβοι δικτύου να μπορούν να τοποθετηθούν διαδοχικά ο ένας μετά τον άλλον, αρκεί στους τερματικούς κόμβους να υπάρχει ο κατάλληλος τερματισμός γραμμής, που επιτυγχάνεται με μια αντίσταση 120 Ohm. Κάθε αισθητήρας που είναι συνδεδεμένος στο δίκτυο έχει μια αποκλειστική διεύθυνση δικτύου.

4.1.2 Πύλη (gateway) RS-485

Είναι ένας μετατροπέας ή μεταφραστής σειριακού πρωτοκόλλου σε φυσικό επίπεδο, δηλαδή σε επίπεδο ηλεκτρικών σημάτων. Τα πακέτα στέλνονται από τον Η/Υ μέσω της θύρας USB, ενώ από τους αισθητήρες στέλνονται μέσω του RS-485 frontend. Πρόκειται για δύο διαφορετικούς «κόσμους», όπου τα σήματα μεταδίδονται με διαφορετικό τρόπο και με διαφορετικές στάθμες. Στην περίπτωση του USB υπάρχει μετάδοση ψηφιακού σήματος 0-5V σε ένα καλώδιο, με το δεύτερο καλώδιο να είναι η κοινή γείωση. Μάλιστα εάν πρόκειται για σταθερό Η/Υ ή laptop που είναι συνδεδεμένο στον φορτιστή, τότε αυτό το «κοινό» είναι ενωμένο με τη γείωση της ΔΕΗ. Στο RS-485 από την άλλη δεν υπάρχει «κοινό», αφού η μετάδοση γίνεται με τη χρήση διαφορικού ζεύγους καλωδίων, σε στάθμες κοινού σήματος από -7V έως +12V.

Όπως καθίσταται σαφές δεν μπορούμε απλά να ενώσουμε τα καλώδια των αισθητήρων με αυτά του USB. Χρειάζεται κάποιος μετατροπέας και η πύλη RS-485 κάνει ακριβώς αυτήν τη δουλειά. Διαθέτει εσωτερικό μετατροπέα πρωτοκόλλου με χρήση RS-485 frontend. Επίσης φροντίζει όταν δέχεται δεδομένα από το PC, να κρατάει τη γραμμή «κατελιημμένη» ώστε το PC (master) να μπορεί να στέλνει χωρίς να είναι επιφορτισμένο με την οδήγηση των σημάτων ελέγχου ροής. Αντίθετα όταν το PC δεν στέλνει, ο μετατροπέας «απελευθερώνει» τη γραμμή, ώστε να μπορούν να μιλήσουν οι κόμβοι του δικτύου (slaves) και το PC να «ακούει».

4.1.3 Διαχειριστής δικτύου αισθητήρων / Web Server

Πρόκειται για το software που τρέχει εντός του Η/Υ. Είναι το ίδιο πρόγραμμα αλλά με δύο διαφορετικά threads που τρέχουν μεταξύ τους ανεξάρτητα, μοιραζόμενα κάποιους κοινούς πόρους μεταξύ τους (δεδομένα).

Κεντρικός ελεγκτής συστήματος (ΚΕΣ): Είναι το thread διαχείρισης αισθητήρων. Επικοινωνεί με τους αισθητήρες, διαχειρίζεται τα δεδομένα που παραλαμβάνει από αυτούς και εξάγει τις καταστάσεις του συστήματος βάσει των δεδομένων αυτών. Επίσης εκτελεί τις εντολές του χρήστη που έρχονται από την ιστοσελίδα.

Web Server: Σε αυτό το thread δεν παίρνεται καμιά απόφαση για την κατάσταση του συστήματος και δεν εκτελείται καμία εντολή χρήστη από την ιστοσελίδα. Οι εργασίες του server είναι οι εξής:

1. Να «σηκώσει» την ιστοσελίδα κατά την εκκίνηση του συστήματος.
2. Να τερματίσει την ιστοσελίδα κατά τον τερματισμό του συστήματος.
3. Να διαβάζει τα αποθηκευμένα δεδομένα από τον ΚΕΣ, ώστε να τα στείλει στην ιστοσελίδα όταν η τελευταία τα ζητήσει.
4. Να στείλει τα δυναμικά κείμενα (μετρήσεις και καταστάσεις του συστήματος) στην ιστοσελίδα, ώστε να απεικονιστούν στον χρήστη.
5. Να διαμορφώσει τα χρώματα των δυναμικών κειμένων της ιστοσελίδας (πχ μετρήσεις από τους αισθητήρες), ώστε να αποτυπώνονται ευκρινώς προς τον χρήστη οι καταστάσεις συναγερμών και σφαλμάτων. Τα χρώματα των γραμματοσειρών στέλνονται μαζί με τα δυναμικά κείμενα.
6. Να διαβάζει τις εντολές του χρήστη από την ιστοσελίδα και να τις αποθηκεύει σε κατάλληλες θέσεις, ώστε στη συνέχεια να μπορούν να αναγνωστούν και να εκτελεστούν από τον ΚΕΣ.

4.1.4 Web client

Είναι το κομμάτι που διαμορφώνει την ιστοσελίδα που βλέπει ο χρήστης στην οθόνη του. Με εξαίρεση τα δυναμικά κείμενα (πχ μετρήσεις αισθητήρων) και τα χρώματά τους που καθορίζονται από τον server, όλα τα υπόλοιπα ελέγχονται από το πρόγραμμα του client. Σε αυτά ανήκουν τα χρώματα των στατικών στοιχείων, οι γραμματοσειρές, η θέση του κάθε στοιχείου, αλλά και κάποιες ειδικές λειτουργίες, με σημαντικότερη την αναγνώριση των εντολών από τον χρήστη και την αποστολή τους προς τον server. Εν κατακλείδι το κομμάτι του client αφορά εξολοκλήρου τον σχεδιασμό και τη λειτουργικότητα της ιστοσελίδας.

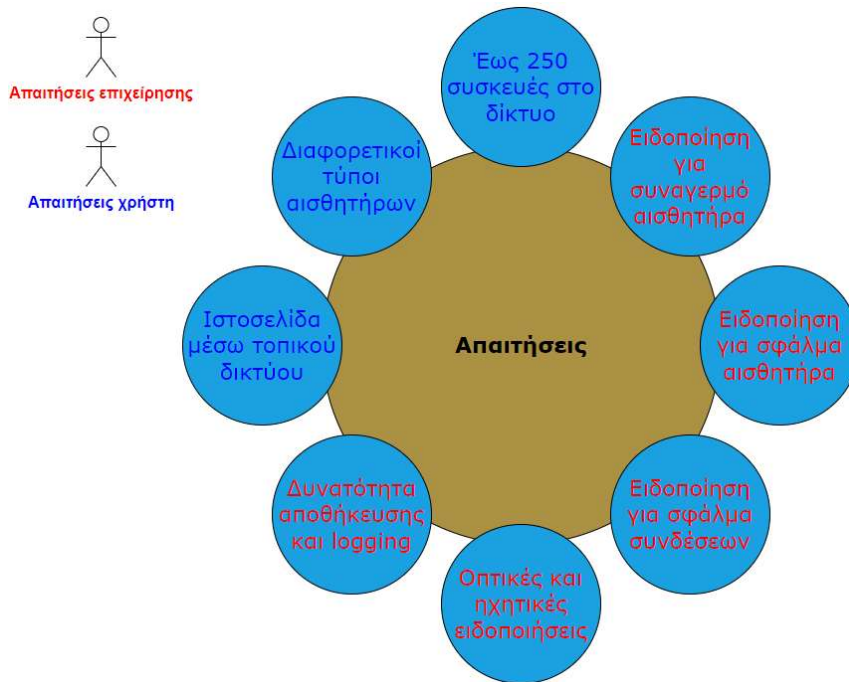
4.2 Προδιαγραφές συστήματος

4.2.1 Απαιτήσεις συστήματος

Οι απαιτήσεις και οι προδιαγραφές εξορισμού δεν είναι ένα και το αυτό. Κάποιες από τις απαιτήσεις είναι υποσύνολο των προδιαγραφών και κάποιες άλλες μεταφράζονται σε προδιαγραφές, μέσω της εν γένει μελέτης του συστήματος πριν την υλοποίηση. Για να γίνουν όλα αυτά πιο σαφή θα αναλυθεί η εικόνα 10, όπου φαίνονται οι απαιτήσεις του συστήματος.

Παίρνοντας ως παράδειγμα από την εικόνα 10 την απαίτηση «Έως 250 αισθητήρες» βρισκόμαστε στην περίπτωση, όπου μια απαίτηση γίνεται αυτόματα και προδιαγραφή, αφού η απαίτηση είναι περιγραφική από μόνη της και δεν επιδέχεται περεταίρω εξήγησης.

Εξετάζοντας από την άλλη την απαίτηση «Διαφορετικοί τύποι αισθητήρων» είμαστε στην περίπτωση της απαίτησης που, με κάποιον τρόπο, θα πρέπει να «μεταφραστεί» σε προδιαγραφή, διότι μια προδιαγραφή δεν μπορεί να μείνει σε τόσο γενικό επίπεδο. Θα πρέπει, μέσω των προδιαγραφών, να γίνει συγκεκριμένο προς τον χρήστη ποιοι ακριβώς τύποι αισθητήρων υποστηρίζονται και πώς αυτοί μπορούν να «κουμπώσουν» στο σύστημα. Η απάντηση σε αυτά τα ερωτήματα θα δοθεί μέσω μελετών που θα γίνει από μηχανικούς και μεταξύ άλλων θα περιλαμβάνει εξέταση των τύπων των αισθητήρων και έρευνα αγοράς για το τι υποστηρίζει ο ανταγωνισμός και τι ζητάει η αγορά.



Εικόνα 10 - : Απαιτήσεις συστήματος

Ως προς την «ελαστικότητα», υπάρχουν προδιαγραφές που δεν μπορούν να τροποποιηθούν ούτε στο ελάχιστο από τις αρχικές απαιτήσεις. Ως παράδειγμα μπορεί να παρατεθεί μια λειτουργία που προβλέπεται από κάποιο ευρωπαϊκό πρότυπο και είναι υποχρεωτική. Κάποιες άλλες απαιτήσεις είναι πιο ελαστικές, διότι τις επιβάλλει όχι κάποιος κανονισμός, αλλά ο ανταγωνισμός. Οι 250 συσκευές για παράδειγμα θα μπορούσαν να γίνουν 20, αλλά κάτι τέτοιο θα κατέτασσε αυτόματα το προϊόν χαμηλά ποιοτικά σε σχέση με τον ανταγωνισμό. Θα μπορούσαν όμως εφόσον υπήρχε δυσκολία υλοποίησης των 250 συσκευών να χαμήλωνε το μέγιστο στις 240. Είναι μια τιμή που είναι κοντά στον ανταγωνισμό και ίσως θεωρείται αποδεκτή από την αγορά. Σε κάθε περίπτωση, σε αντίθεση με τις ανελαστικές απαιτήσεις που πρέπει να τηρηθούν στο έπακρο για να βγει το προϊόν στην αγορά, οι υπόλοιπες απαιτήσεις μπορούν σε μικρότερο ή μεγαλύτερο βαθμό να θεωρηθούν «ελαστικές» και να τροποποιηθούν ελαφρώς στην πορεία. Και αφού οι ανελαστικές απαιτήσεις πληρούνται (θεωρητικά) από όλους, οι ελαστικές απαιτήσεις είναι αυτές που, μέσω των προδιαγραφών που διαμορφώνουν, δίνουν τις ποιοτικές διαφοροποιήσεις και τα ανταγωνιστικά πλεονεκτήματα μεταξύ των κατασκευαστών.

4.2.2 Βασικές προδιαγραφές συστήματος

Οι βασικές προδιαγραφές διαμορφώνονται από τις απαιτήσεις στην εικόνα 10. Όπως φαίνεται από την εικόνα, χωρίζονται σε δύο κατηγορίες:

Α. Απαιτήσεις επιχείρησης: Είναι στον κεντρικό πυρήνα της επιχείρησης. Αφορούν στην ουσία την ίδια την επιχείρηση και τα αγαθά που προσφέρονται στον πελάτη. Ο πελάτης στη συγκεκριμένη περίπτωση πληρώνει την ασφάλεια της ανθρώπινης ζωής με την έγκαιρη ανίχνευση παθογόνων καταστάσεων που προκαλούνται από διαρροές επικίνδυνων αερίων και άμεση ειδοποίηση των ανθρώπων στον χώρο αυτόν. Οι απαιτήσεις αυτές, μαζί με

ανελαστικές απαιτήσεις που προκύπτουν από υποχρεωτικούς κανονισμούς, ανήκουν στην κατηγορία αυτή. Σύμφωνα με την εικόνα 10 οι απαιτήσεις αυτές είναι οι εξής:

- Ειδοποίηση για συναγερμό μέτρησης: η κύρια λειτουργία του συστήματος. Εάν μια τιμή μέτρησης ξεπεράσει ένα καθορισμένο όριο, θα πρέπει να δοθεί από το σύστημα συναγερμός.
- Ειδοποίηση για σφάλμα μέτρησης: το σφάλμα μέτρησης είναι και αυτό κρίσιμο να το λαμβάνει ο χρήστης, ώστε να έχει επίγνωση ότι ο συγκεκριμένος αισθητήρας είναι σε κάποια δεδομένη στιγμή εκτός λειτουργίας.
- Ειδοποίηση για σφάλμα σύνδεσης αισθητήρα: το σφάλμα σύνδεσης αισθητήρα είναι και αυτό κρίσιμο να το λαμβάνει ο χρήστης, ώστε να έχει επίγνωση ότι ο συγκεκριμένος αισθητήρας είναι κατά την τρέχουσα στιγμή εκτός λειτουργίας.
- Οπτικές και ηχητικές ειδοποιήσεις: ο χρήστης πρέπει να έχει όχι μόνο οπτικές ειδοποιήσεις στην ιστοσελίδα, αλλά και επάνω στους αισθητήρες, ώστε να υπάρχει απευθείας ειδοποίηση και στο πεδίο λειτουργίας των αισθητήρων.
- Δυνατότητα αποθήκευσης δεδομένων και συμβάντων: χωρίς τη λειτουργία αυτή ο πελάτης δεν έχει επίγνωση του τι έχει συμβεί και δεν μπορεί να επέμβει αποτελεσματικά για να επιδιορθώσει μια ανεπιθύμητη κατάσταση (πχ μια διαρροή σε κάποια συσκευή).

B. Απαιτήσεις χρήστη: Έχουν να κάνουν με απαιτήσεις που διαμορφώνονται από τους πελάτες και από την αγορά. Δίνουν υπεραξία και μεγαλύτερη ποιότητα στο προϊόν, κατατάσσοντάς το υψηλότερα στον ανταγωνισμό. Σύμφωνα με την εικόνα 10 οι απαιτήσεις αυτές είναι:

- Ιστοσελίδα παρακολούθησης και ελέγχου: εφόσον δεν υπάρχει φυσικό μέσο (ηλεκτρολογικός πίνακας) για την παρακολούθηση και έλεγχο του συστήματος, θα πρέπει αυτό να γίνει μέσω του Η/Υ. Ένα λογισμικό που θα έτρεχε τοπικά και αποκλειστικά σε ένα τερματικό δεν αρκεί, συνεπώς χρειάζεται να υπάρχει παρακολούθηση και έλεγχος από οποιοδήποτε σημείο εντός του κτηρίου, κάτι το οποίο μπορεί να προσφέρει από μια ιστοσελίδα στο τοπικό δίκτυο.
- Διαφορετικοί τύποι αισθητήρων και συσκευών: ο κρίσιμος παράγοντας που καλύπτεται με την προδιαγραφή αυτή είναι η επεκτασιμότητα. Το σύστημα πρέπει να είναι θωρακισμένο ώστε να μπορεί να επιδέχεται μελλοντικών τροποποιήσεων και βελτιώσεων. Συνεπώς η υποδομή πρέπει να είναι επεκτάσιμη και για περισσότερους τύπους αισθητήρων, αλλά και συσκευών όπως σειρήνες, μπουτόν πανικού κτλ.
- Έως και 250 συσκευές δικτύου: Σε κάποιες περιπτώσεις το σύστημα θα χρειαστεί να καλύψει μεγάλη έκταση. Η δυνατότητα για πολλές συσκευές είναι κάτι που πρέπει να δοθεί στον χρήστη, ώστε η συσκευή να συμβαδίζει με τον ανταγωνισμό.

4.2.3 Καταστάσεις συστήματος

- **Κατάσταση συναγερμού**: το σύστημα βρίσκεται σε κατάσταση συναγερμού όταν τουλάχιστον ένας αισθητήρας βρίσκεται σε κατάσταση συναγερμού. Ένας αισθητήρας βρίσκεται σε κατάσταση συναγερμού όταν τουλάχιστον ένα από τα είδη

μετρήσεων του (εάν πρόκειται για πολύ-αισθητήρα) έχει ξεπεράσει το προκαθορισμένο όριο συναγερμού.

- **Κατάσταση σφάλματος:** το σύστημα βρίσκεται σε κατάσταση σφάλματος όταν κανένας αισθητήρας δεν βρίσκεται σε κατάσταση συναγερμού και έχει αναγνωριστεί ως ενεργή τουλάχιστον μία από τις περιπτώσεις σφάλματος, που περιγράφονται στην παράγραφο 4.2.5.
- **Κατάσταση ηρεμίας:** το σύστημα βρίσκεται σε κατάσταση ηρεμίας όταν κανένας αισθητήρας δεν βρίσκεται σε κατάσταση συναγερμού και δεν έχει αναγνωριστεί ως ενεργή καμία από τις περιπτώσεις σφάλματος.

4.2.4 Στάθμες συναγερμού

Για να δώσει το σύστημα συναγερμό, συγκρίνει τις εισερχόμενες τιμές των αισθητηρίων με συγκεκριμένες τιμές. Οι τιμές αυτές είναι ενδεικτικές και προέκυψαν από τον συνδυασμό των μετρήσεων που παρουσιάζονται σε παρακάτω κεφάλαιο, αλλά και κάποιων αναφορών από συγκεκριμένες πηγές. Συγκεκριμένα:

Αιωρούμενα σωματίδια: 100 $\mu\text{g}/\text{m}^3$. Σύμφωνα με την επίσημη ιστοσελίδα της πολιτείας της Ν. Υόρκης των ΗΠΑ, η επικινδυνότητα στις ευπαθείς ομάδες ξεκινάει από τη συγκέντρωση των 100 $\mu\text{g}/\text{m}^3$ [14], οπότε η τιμή που επιλέχτηκε είναι αυτή.

VOC και NOx: 250. Οι συγκεκριμένες τιμές αποτελούν δείκτες που δίνει το αισθητήριο προς τον χρήστη και εξάγεται από συγκεκριμένους αλγόριθμους εντός του μικροελεγκτή του SEN55. Η τιμή αυτή επιλέχτηκε από συγκεκριμένο παράδειγμα εφαρμογής που μετουσιώθηκε σε γράφημα από την κατασκευάστρια εταιρία, την SENSIRION [15].

Θερμοκρασία: 45°C. Η τιμή αυτή επιλέχτηκε εμπειρικά, αφού σε κλειστό χώρο τέτοια θερμοκρασία θα σήμαινε την αρχή κάποιου συμβάντος, πιθανότατα πυρκαγιάς.

Υγρασία: Δεν υπάρχει συναγερμός. Η τιμή της χρησιμοποιείται απλώς για απεικόνιση,

Σε μια πραγματική εφαρμογή του εμπορίου χρειάζονται μακροχρόνια πειράματα ετών για να προσδιοριστούν με σχετική ακρίβεια οι τιμές αυτές, και μάλιστα ανάλογα με την εγκατάσταση μπορεί και να ποικίλλουν. Το πιθανότερο είναι να δίνεται η δυνατότητα και στον χρήστη να μπορεί να μεταβάλλει τις στάθμες συναγερμού προς τα πάνω ή προς τα κάτω.

Θα μπορούσαν, επίσης, να προστεθούν και πιο πολύπλοκοι συναγερμοί, όπως για παράδειγμα συναγερμοί που προκύπτουν συνδυαστικά από πολλές μετρούμενες τιμές. Εάν για παράδειγμα παρατηρηθεί ταυτόχρονη αύξηση θερμοκρασίας και σωματιδίων κάτω μεν από τα όρια συναγερμών τους, αλλά ταυτόχρονα και απότομα, ίσως αυτό να είναι ένδειξη ότι κάποια φλόγα έχει ανάψει και είναι ακόμα σε αρχικό στάδιο για να ανιχνευτεί από το σύστημα πυρανίχνευσης του κτηρίου. Οι επιλογές είναι πολλές και όλες τους απαιτούν χιλιάδες μετρήσεις υπό διαφορετικές συνθήκες, καθώς και μελέτη πολλών πηγών, όπως πχ στο συγκεκριμένο παράδειγμα γύρω από τη φλόγα, το πώς ανεβαίνει η θερμοκρασία και σε τι ποσοστό υπάρχει έκκριση σωματιδίων στο αρχικό στάδιο μιας πυρκαγιάς.

4.2.5 Περιπτώσεις σφαλμάτων

- **USB θύρα αποσυνδεδεμένη:**
Δεν αναγνωρίζεται η σύνδεση USB. Πιθανώς αποσυνδεδεμένο USB καλώδιο ή βλάβη στο RS-485 gateway.
- **Κόμβος αποσυνδεδεμένος από το δίκτυο:**
Ο αισθητήρας δεν έχει απαντήσει στο δίκτυο για κάποιον προκαθορισμένο χρόνο. Πιθανή αποσύνδεση ενός εκ των δύο καλωδίων του RS-485 στον αισθητήρα.
- **Σφάλμα μέτρησης αισθητήρα:**
Η τιμή που στέλνει ο αισθητήρας δεν ανήκει στο αποδεκτό εύρος τιμών. Πιθανή μόνιμη βλάβη του αισθητήρα.
- **Εσφαλμένος τύπος αισθητήρα:**
Ο αισθητήρας στέλνει στο δίκτυο άλλον τύπο από αυτόν που είναι καταχωρημένος στο αρχείο δικτύωσης αισθητήρων. Πιθανό λάθος του αρχείου δικτύωσης αισθητήρων.
- **Σφάλμα αποθήκευσης μετρήσεων:**
Οι μετρήσεις των αισθητήρων και οι καταστάσεις του συστήματος δεν μπορούν να αποθηκευτούν επιτυχώς στο μέσο αποθήκευσης (σκληρός δίσκος). Πιθανό σφάλμα στο μέσο αποθήκευσης.
- **Σφάλμα ανάγνωσης μετρήσεων:**
Οι αποθηκευμένες μετρήσεις των αισθητήρων και οι καταστάσεις του συστήματος δεν μπορούν να αναγνωστούν επιτυχώς από το μέσο αποθήκευσης (σκληρός δίσκος). Πιθανό σφάλμα στο μέσο αποθήκευσης ή στο αρχείο μετρήσεων.
- **Σφάλμα ανοιχτού αισθητήρα:**
Ο αισθητήρας επικοινωνεί στο δίκτυο, αλλά το αισθητήριο όργανο είναι αποσυνδεδεμένο από την πλακέτα του αισθητήρα.
- **Σφάλμα βραχυκυκλωμένου αναλογικού αισθητήρα:**
Ο αισθητήρας επικοινωνεί στο δίκτυο, αλλά το αισθητήριο όργανο έχει τα άκρα μέτρησής του βραχυκυκλωμένα. Εφαρμόσιμο μόνο στα αισθητήρια όργανα αναλογικού τύπου.

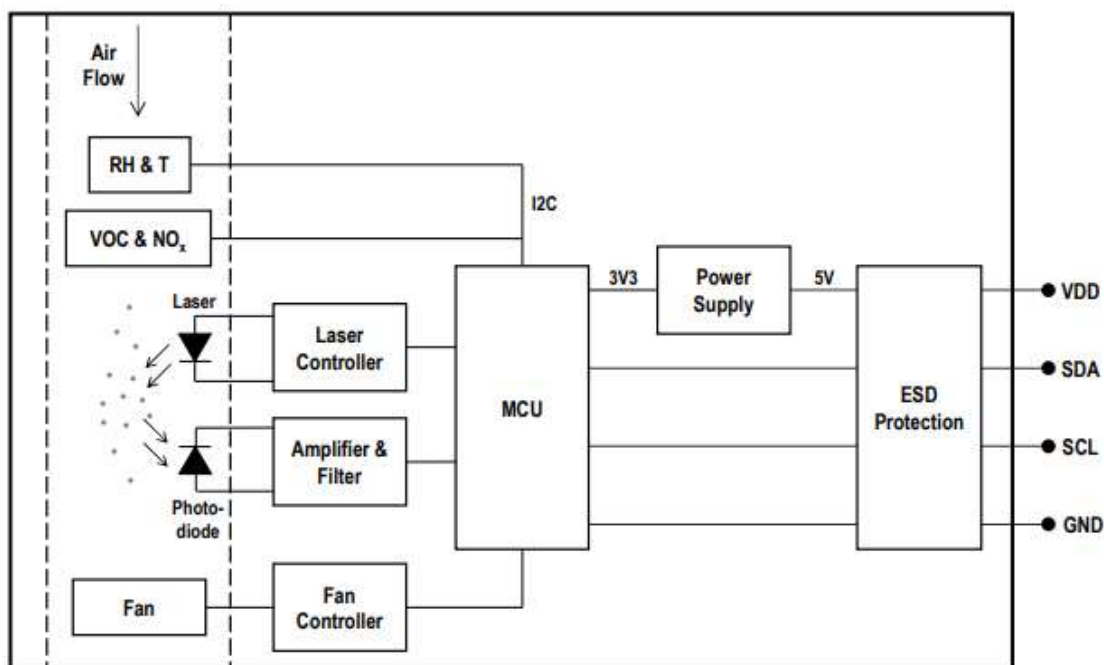
4.3 Περιβαντολλογικός αισθητήρας SEN55

Ο περιβαντολλογικός αισθητήρας που επιλέχτηκε για τη συγκεκριμένη κατασκευή είναι το module SEN55 της ελβετικής εταιρίας SENSIRION. Αναφέρεται ως «module» διότι δεν πρόκειται απλά για ένα αισθητήριο όργανο που συναρμολογείται στην πλακέτα με ένα κύκλωμα προσαρμογής, αλλά για πολύ-αισθητήρα με ενσωματωμένη CPU που εφαρμόζει κυλιόμενο MO με ιστορικό μίας ημέρας, καθώς και πατενταρισμένους αλγόριθμους του οποίους η εταιρία δεν τους παραθέτει στα έγγραφα που δημοσιεύει, προφανώς για λόγους διατήρησης της αποκλειστικότητας χρήσης τους στον ανταγωνισμό. Ο χρήστης όμως μπορεί να ρυθμίσει κάποιες σημαντικές παραμέτρους αυτών των αλγορίθμων, όπως για παράδειγμα την offset των σημάτων ηρεμίας. Επίσης διαθέτει ενσωματωμένες προστασίες από

ηλεκτρομαγνητικές παρεμβολές, καθώς και εσωτερικό ανεμιστήρα για να βελτιώσει τους χρόνους απόκρισης.

4.3.1 Μπλοκ διάγραμμα

Στην εικόνα 11 φαίνεται το μπλοκ διάγραμμα του SEN55. Στο αριστερό τμήμα της εικόνας, φαίνεται το κομμάτι των αισθητηρίων οργάνων, τα οποία ομαδοποιούνται στα εξής τρία μπλοκ: RH & T (θερμοκρασία – υγρασία), VOC & NO_x και αιωρούμενα σωματίδια. Αυτά συνδέονται στον μικροελεγκτή, ο οποίος οδηγεί και τον ανεμιστήρα. Η εξωτερική τροφοδοσία είναι 5V, η οποία εσωτερικά υποβαθμίζεται σε 3.3V για να τροφοδοτήσει τον μικροελεγκτή. Στο δεξί τμήμα της εικόνας φαίνονται οι εξωτερικές συνδέσεις του module, που είναι η τροφοδοσία και οι δύο συνδέσεις του I²C, το οποίο είναι και το interface προς τον χρήστη. Συνεπώς για να δοθούν εντολές προς το module ή να διαβαστούν οι τιμές των αισθητηρίων οργάνων, θα πρέπει ο εξωτερικός μικροελεγκτής (I²C master) να επικοινωνήσει με το module (I²C slave) μέσω του συγκεκριμένου interface.



Εικόνα 11 - Μπλοκ διάγραμμα αισθητήρα SEN55 [16]


4.3.2 Τεχνικά χαρακτηριστικά

Τα τεχνικά χαρακτηριστικά των τριών μπλοκ των αισθητηρίων οργάνων φαίνονται στην εικόνα 12 (θερμοκρασία / υγρασία), στην εικόνα 13 (αιωρούμενα σωματίδια), και στην εικόνα 14 (VOC και NO_x).

Parameter	Conditions	Value ⁷	Units
Compensated outputs ⁸	-	Temperature and Relative Humidity	°C %RH
Typical accuracy temperature	@ 15-30 °C, 50 %RH	±0.45	°C
Max. accuracy temperature	@ 15-30 °C, 50 %RH	±0.7	°C
Repeatability temperature	@ 25 °C, 50 %RH	0.1	°C
Response time temperature ⁹	@ 25 °C, 50 %RH, τ 63%	< 60	s
Typical accuracy relative humidity	@ 25 °C, 30-70 %RH	±4.5	%RH
Max. accuracy relative humidity	@ 25 °C, 30-70 %RH	±6	%RH
Repeatability relative humidity	@ 25 °C, 50 %RH	±1	%RH
Response time relative humidity ¹⁰	@ 25 °C, 50 %RH, τ 63%	< 20	s

Εικόνα 12 - Τεχνικά χαρακτηριστικά SEN55 για τη μέτρηση θερμοκρασίας και υγρασίας [16]

Παρατηρώντας τα τεχνικά χαρακτηριστικά στα αιωρούμενα σωματίδια (εικόνα 13), φαίνεται ότι ενώ στις μικρότερης διατομής σωματίδια έως 2.5 $\mu\text{g}/\text{m}^3$ η ακρίβεια είναι καλή (5 $\mu\text{g}/\text{m}^3$ για την «ασφαλή» για την ανθρώπινη υγεία περιοχή), στα μεγαλύτερα σωματίδια η ακρίβεια είναι 25 $\mu\text{g}/\text{m}^3$. Συνεπώς μια εφαρμογή που χρησιμοποιεί τον SEN55 θα πρέπει να δίνει μεγαλύτερο βάρος στη μέτρηση των μικρότερων σωματιδίων, τα οποία μάλιστα είναι και τα πιο βλαβερά για την υγεία.

Parameter	Conditions	Value	Units	
Mass concentration specified range	-	0 to 1'000	$\mu\text{g}/\text{m}^3$	
Mass concentration size range	PM1.0	0.3 to 1.0	μm	
	PM2.5	0.3 to 2.5	μm	
	PM4	0.3 to 4.0	μm	
	PM10	0.3 to 10.0	μm	
Mass concentration precision ^{2,3} for PM1 and PM2.5 ⁴	0 to 100 $\mu\text{g}/\text{m}^3$	±5 $\mu\text{g}/\text{m}^3$ AND 5 % m.v.		
	100 to 1000 $\mu\text{g}/\text{m}^3$	±10	% m.v.	
Mass concentration precision ^{2,3} for PM4, PM10 ⁵	0 to 100 $\mu\text{g}/\text{m}^3$	±25	$\mu\text{g}/\text{m}^3$	
	100 to 1000 $\mu\text{g}/\text{m}^3$	±25	% m.v.	
Maximum long-term mass concentration precision limit drift	0 to 100 $\mu\text{g}/\text{m}^3$	±1.25	$\mu\text{g}/\text{m}^3$ / year	
	100 to 1000 $\mu\text{g}/\text{m}^3$	±1.25	% m.v. / year	
Typical start-up time ⁶	number concentration	200 – 3000 #/cm ³	8	s
		100 – 200 #/cm ³	16	s
		50 – 100 #/cm ³	30	s
Sensor output characteristics	PM2.5 mass concentration	Calibrated to TSI DustTrak™ DRX 8533 Ambient Mode		
Additional T-dependent mass precision limit drift ³	temperature difference to 25°C	typ.	±0.5	% m.v. / °C
Laser wavelength (DIN EN 60825-1 Class 1)		typ.	660	nm

Εικόνα 13 - Τεχνικά χαρακτηριστικά SEN55 για τη μέτρηση αιωρούμενων σωματιδίων [16]

Parameter	Comments	Values			Units	
		Min.	Typ. ¹¹	Max.		
Output signals	VOC Index	1	–	500	VOC Index points	
	NO _x Index	1	–	500	NO _x Index points	
Device-to-device variation	VOC Index ¹²	–	<±15 <±15	–	VOC Index points or % VOC Index m.v. (the larger)	
	NO _x Index ¹²	–	<±50 <±50	–	NO _x Index points or % NO _x Index m.v. (the larger)	
Repeatability	VOC Index ¹²	–	<±5 <±5	–	VOC Index points or % VOC Index m.v. (the larger)	
	NO _x Index ¹²	–	<±10 <±10	–	NO _x Index points or % NO _x Index m.v. (the larger)	
Response time	Changing concentration from 5 to 10 ppm of ethanol, at sampling interval of 1 s	τ ₆₃	–	<10	–	s
		τ ₉₀	–	<30	–	s
Switch-on behavior	Time until reliably detecting events ¹³	–	<60	–	s	
	Time until specifications in this table are met	VOC Index	<1	–	h	
		NO _x Index	<6	–	h	

Εικόνα 14 - Τεχνικά χαρακτηριστικά SEN55 για τη μέτρηση των αερίων [16]

Σε ότι αφορά τη θερμοκρασία (εικόνα 12), η μέγιστη απόκλιση του 0.7°C είναι ικανοποιητική για τη συγκεκριμένη εφαρμογή και ακόμη πιο ικανοποιητικός είναι ο χρόνος απόκρισης των 60 δευτερολέπτων, αφού από τη μία είναι γρήγορος για ένα φαινόμενο που μεταβάλλεται αργά όπως η θερμοκρασία και από την άλλη παρέχει την απαραίτητη ολοκλήρωση της μέτρησης, ώστε να αποφεύγονται οι αναπηδήσεις της τιμής. Μάλιστα χρησιμοποιείται για τη θερμοκρασία ιδιόκτητος από την SENSIRION αλγόριθμος με το όνομα «STAR» [17].

Το 6% μέγιστο για την υγρασία είναι αρκετά μεγάλη ως απόκλιση, οπότε αυτή η μέτρηση μπορεί να χρησιμοποιηθεί μόνο ενδεικτικά και αυτός είναι και ο λόγος που δεν παρέχεται συναγερμός για την υγρασία, αν και από τη φύση της η υγρασία δεν είναι ένα μέγεθος που δημιουργεί άμεση επικινδυνότητα όπως τα άλλα μετρούμενα μεγέθη. Ο χρόνος απόκρισης των 20 δευτερολέπτων παρόλα αυτά, είναι ικανοποιητικός.

Στους δείκτες VOC και NO_x (εικόνα 14) δεν μπορούν να εξαχθούν συμπεράσματα για την ακρίβεια, αφού οι τιμές αυτές προκύπτουν από αλγόριθμους της κατασκευάστριας εταιρίας και δεν αναφέρονται σε συγκεκριμένες μονάδες. Στα αξιοσημείωτα εντάσσονται η καλή επαναληψιμότητα με απόκλιση μικρότερη του 5% από αισθητήρα σε αισθητήρα, η χαμηλή διαφορά στη διακύμανση για το VOC index (5 μονάδες) και η όχι και τόσο χαμηλή διαφορά στη διακύμανση για τον NO_x (50 μονάδες).

Τα ηλεκτρικά χαρακτηριστικά του SEN55 φαίνονται στην εικόνα 15.

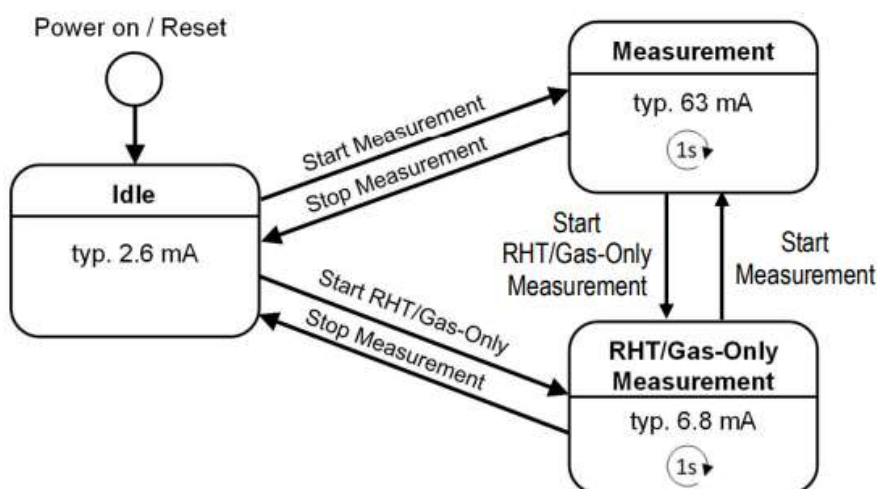
Parameter	Conditions	Min	Typ	Max	Unit	
Supply voltage	-	4.5	5.0	5.5	V	
Sensor startup time (Time after power-on until I2C communication can be started)	-	-	-	50	ms	
Average supply current	Idle Mode (first 10 seconds)	SEN55	-	3.8	4.2	mA
		SEN54	-	0.7	1	
		SEN50	-	0.7	1	
	Idle Mode (after first 10 seconds)	SEN55	-	2.6	3	
		SEN54	-	0.7	1	
		SEN50	-	0.7	1	
	RHT/Gas-only Measurement Mode	SEN55	-	6.8	8	
		SEN54	-	6.5	7.7	
	Measurement-Mode (first 60 seconds)	SEN55	-	70	100	
		SEN54	-	70	100	
		SEN50	-	70	100	
	Measurement-Mode (after first 60 seconds)	SEN55	-	63	80	
SEN54		-	63	80		
SEN50		-	63	80		
Peak supply current	Measurement mode (pulse width of 16μs)	-	100	110	mA	
Input high level voltage (V _{HI})	-	2.31	-	5.5	V	
Input low level voltage (V _{LI})	-	0	-	0.99		
Output low level voltage (V _{OL})	Sink current <6mA	0	0	0.4		

Εικόνα 15 - Ηλεκτρικά χαρακτηριστικά του SEN55 για τη μέτρηση των αερίων [16]

Η μηχανολογική εγκατάσταση του αισθητήρα θα πρέπει να γίνει σύμφωνα με τις οδηγίες του κατασκευαστή [18].

4.3.3 Λειτουργικές καταστάσεις

Στην εικόνα 16 διακρίνονται οι λειτουργικές καταστάσεις του συστήματος. Συγκεκριμένα το σύστημα βρίσκεται κατά κύριο λόγο στην κατάσταση idle, όπου η κατανάλωση είναι 2.6mA. Ανά ένα δευτερόλεπτο διαβάζει τα αισθητήρια όργανα. Για τη θερμοκρασία και την υγρασία η κατανάλωση κατά τη μέτρηση είναι 6.8mA, ενώ κατά τις υπόλοιπες μετρήσεις η κατανάλωση είναι 63mA.



Εικόνα 16 - Λειτουργικές καταστάσεις (operating modes) του SEN55 [16]

4.4 Αισθητήρας θερμοκρασίας

Χρησιμοποιήθηκε ο ΚΤΥ81-110 της NXP Semiconductors, ο οποίος είναι αναλογικός και μεταβάλλει την αντίστασή του σύμφωνα με τη θερμοκρασία. Ενδεικτικά η τιμή της αντίστασης είναι 1000 Ohm στους 25°C, με μέγιστη απόκλιση 1.27%, που είναι μια καλή τιμή [19]. Ως παθητικό στοιχείο δεν περιέχει τις «έξυπνες» λειτουργίες και το φιλτράρισμα τιμών του SEN55 και επίσης χρειάζεται και εξωτερικό κύκλωμα προσαρμογής (εικόνα 24).

5 Δίκτυο επικοινωνίας αισθητήρων

Τα γενικά χαρακτηριστικά του δικτύου φαίνονται στον πίνακα1:

Φυσικό μέσο επικοινωνίας	RS-485
Τύπος δικτύου	Single master
Ταχύτητα επικοινωνίας	115200bps
Μέγιστο πλήθος κόμβων	250

Πίνακας 1 - Χαρακτηριστικά δικτύου αισθητήρων

5.1 Τα βασικά του RS-485

Για την καλύτερη κατανόηση του δικτύου αισθητήρων θα δοθούν κάποια γενικά θεωρητικά στοιχεία για το RS-485. Το RS-485 είναι στην αγορά περίπου 40 χρόνια και έχει αποδείξει τη χρήση του. Τα πλεονεκτήματα και οι περιορισμοί του είναι ευρέως γνωστοί, τα προστατευτικά του κυκλώματα δοκιμασμένα και τεκμηριωμένα από πολλούς κατασκευαστές, η τεχνολογία του είναι ώριμη και συνεπώς οικονομική και παράλληλα πολλές και μεγάλες εταιρίες παράγουν RS-485 drivers, δίνοντας ποικιλία επιλογών στους χρήστες.

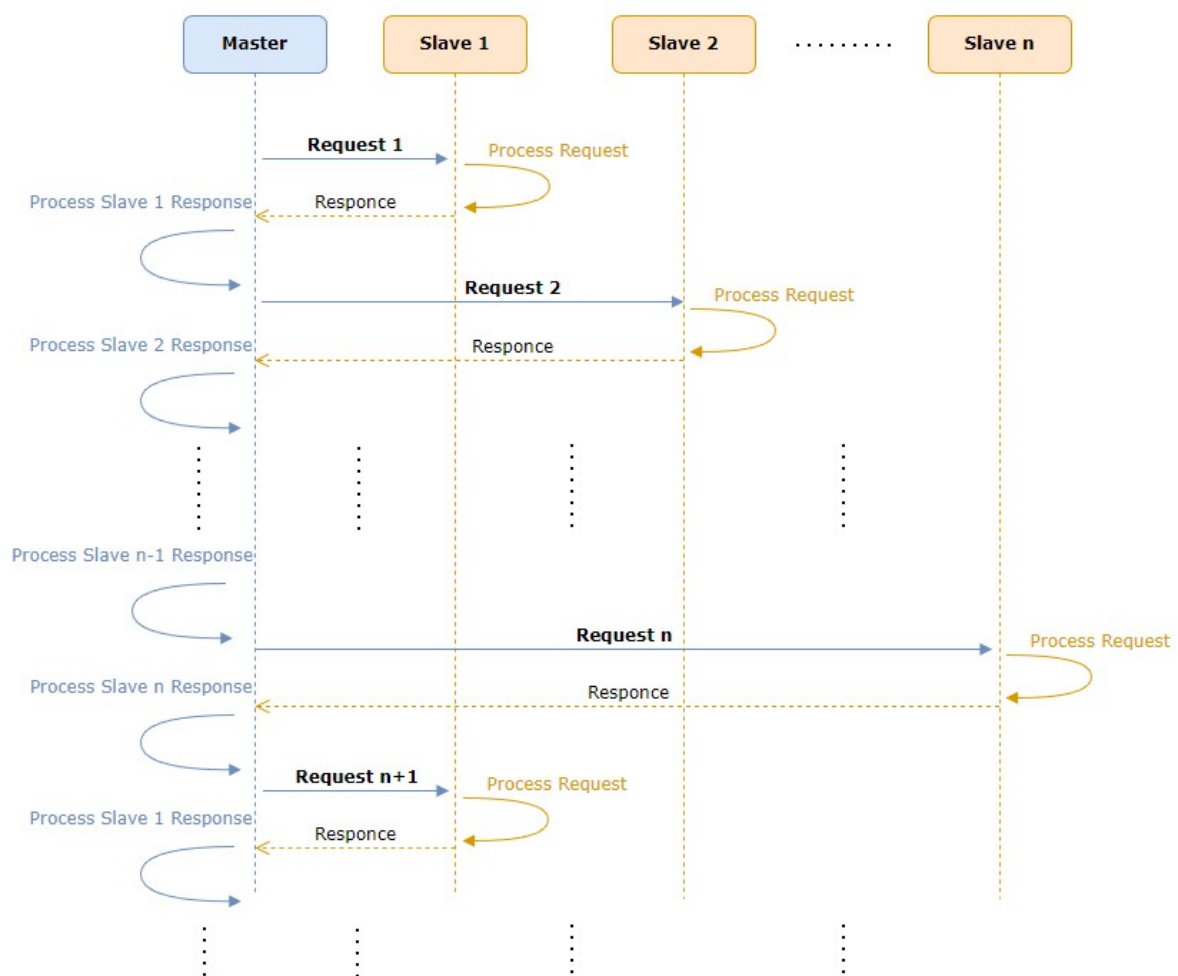
Το RS-485 είναι διεθνές πρότυπο μόνο στο φυσικό επίπεδο των ηλεκτρικών συνδέσεων όπως οι ηλεκτρικές στάθμες, ο αριθμός και ο τύπος των καλωδίων κοκ. Δεν περιορίζει στο να χρησιμοποιηθεί συγκεκριμένο πρωτόκολλο σε software επίπεδο, πώς θα χτιστεί το πακέτο ή τι bytes θα στέλνονται. Είναι ένα πρότυπο που φτιάχτηκε για να επιτρέψει την ταχύτατη και αξιόπιστη μετάδοση δεδομένων σε μεγάλες αποστάσεις. Χρησιμοποιεί τοπολογία τύπου «daisy-chain» και μπορεί να χρησιμοποιηθεί τόσο στην half-duplex εκδοχή (μόνο αποστολή ή μόνο λήψη σε κάποια δεδομένη στιγμή), όσο και στην full-duplex, όπου είναι δυνατή η ταυτόχρονη αποστολή και λήψη [20].

Χρησιμοποιεί δύο καλώδια για το κάθε σήμα και με αυτόν τον τρόπο «ξεκλειδώνει» και τα πλεονεκτήματα του. Το «στρεπτό ζεύγος» καλωδίων (UTP) έχει μεγάλη ανοχή στην ηλεκτρομαγνητική παρεμβολή, δίνοντας τη δυνατότητα στο σήμα να μεταδοθεί σε μεγάλες αποστάσεις και με μεγάλες ταχύτητες. Οι δέκτες RS-485 αναγνωρίζουν ως ωφέλιμο σήμα διαφορικές τάσεις και μάλιστα σε πολλές περιπτώσεις χαμηλές έως και 200mV, ενισχύοντας ακόμα περισσότερο τη δυνατότητα μετάδοσης σε μεγάλες αποστάσεις [20]. Ενδεικτικά για ταχύτητα 115200bps, η απόσταση μπορεί να φτάσει θεωρητικά έως και το 1.5 χιλιόμετρο [20]. Επίσης με αυτόν τον τρόπο απαλλάσσει την εγκατάσταση από την ανάγκη για επιπλέον καλώδιο τοπικής γείωσης. Σε κάποιες περιπτώσεις αυτό είναι απαραίτητο λόγω της φύσης

της εγκατάστασης, αλλά σε κάθε περίπτωση το RS-485 δεν το χρειάζεται για να λειτουργήσει. Αυτό που είναι απαραίτητο στο RS-485 είναι ο κατάλληλος τερματισμός της γραμμής. Οι δύο τερματικοί κόμβοι στο daisy chain πρέπει να τερματίζουν τη γραμμή με αντίσταση 120 Ohm, ώστε να αποφεύγονται οι αντανακλάσεις του σήματος, που οδηγούν στην εξασθένησή του [20].

5.2 Πρωτόκολλο επικοινωνίας

Για την καλύτερη κατανόηση του πρωτοκόλλου παρατίθεται το διάγραμμα ακολουθίας (sequence diagram) στην εικόνα 17.



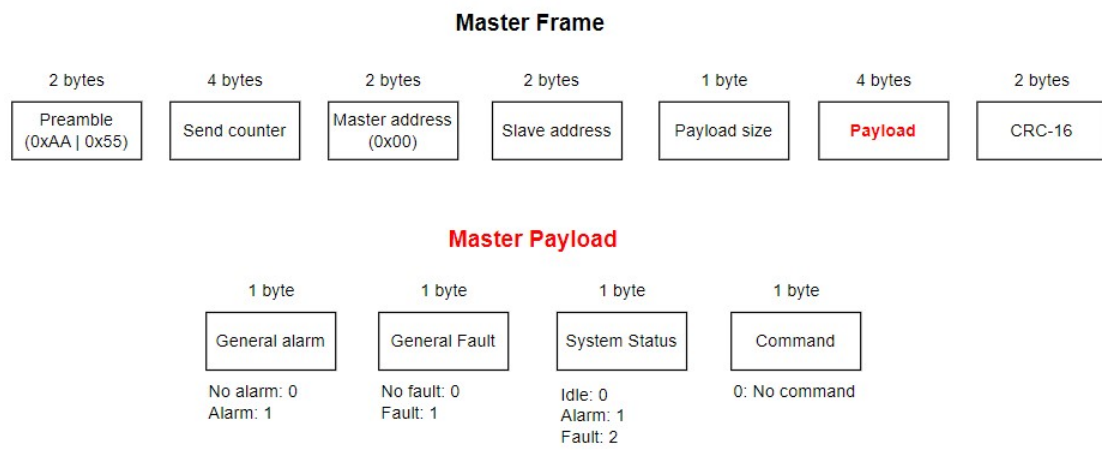
Εικόνα 17 - Διάγραμμα ακολουθίας πρωτοκόλλου επικοινωνίας αισθητήρων

Στο διάγραμμα της εικόνας 17 γίνεται υπόθεση για n κόμβους. Ο master του δικτύου (PC) στέλνει request στον slave 1 (request 1), ο οποίος παραλαμβάνει το πακέτο, το επεξεργάζεται και επιστρέφει την απάντησή του (response) προς τον master. Στη συνέχεια ο master παραλαμβάνει την απάντηση, την επεξεργάζεται και προχωράει με το επόμενο request στον

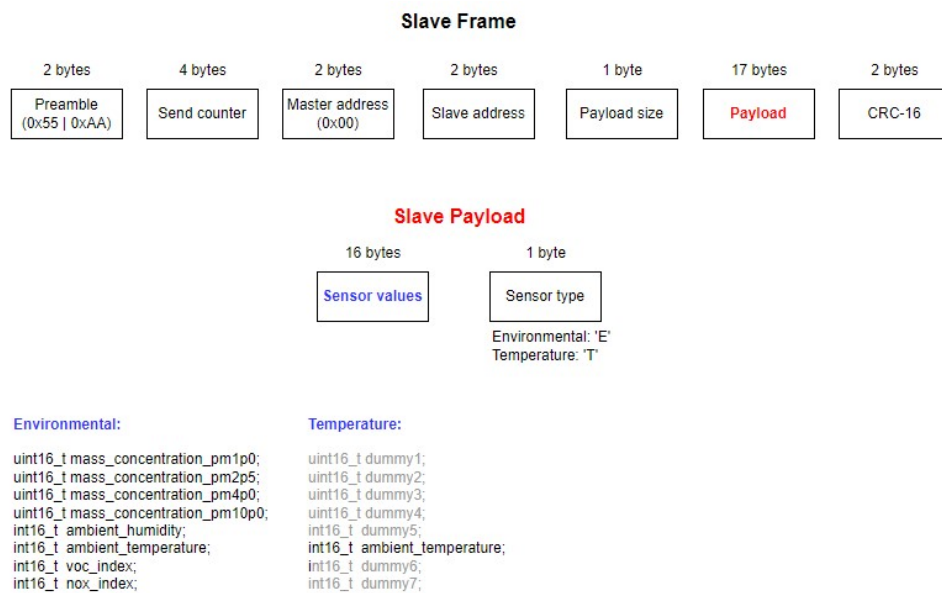
slave 2, ο οποίος με τη σειρά του θα επεξεργαστεί το request και θα απαντήσει, ώστε ο master στη συνέχεια να στείλει request στον επόμενο κόμβο κ.ο.κ. Όταν σταλεί request και παρθεί απάντηση και από τον τελευταίο κόμβο η, τότε ο master αρχίζει να «σαρώνει» τους κόμβους και πάλι από την αρχή, στέλνοντας το request υπ' αριθμόν n+1 στον κόμβο 1, σηματοδοτώντας τον επόμενο κύκλο επικοινωνίας.

5.3 Πακέτα πρωτοκόλλου

Τα διακρίνουμε στο πακέτο του master (εικόνα 18) και στο πακέτο του slave (εικόνα 19).



Εικόνα 18 - Πακέτο αποστολής από τον master προς τον slave



Εικόνα 19 - Πακέτο αποστολής από τον slave προς τον master

5.3.1 Πακέτο αποστολής master (request packet)

Αποτελείται από συνολικά 17 bytes, όπως φαίνεται και στην εικόνα 18, τα οποία χωρίζονται σε 7 διαφορετικές θεματικές ενότητες, οι οποίες περιγράφονται παρακάτω:

Preamble (2 byte): Τα byte αυτά είναι πάντα τα ίδια (0xAA, 0x55). Σκοπός τους είναι ο συγχρονισμός των slave στην αρχή του πακέτου. Εάν κάποιος slave θεωρήσει ότι έχει λάβει τα 2 πρώτα byte του εισερχόμενου πακέτου αλλά αυτά δεν συμφωνούν με το προκαθορισμένο master preamble, τότε συνεχίζει να περιμένει τα 2 byte αυτά ώστε, όταν τα λάβει, να θεωρήσει ότι το εισερχόμενο πακέτο έχει ξεκινήσει να έρχεται.

Μετρητής αποστολής (4 byte): Με τη χρήση του προσδίδεται μια «μοναδικότητα» σε κάποιο πακέτο. Εάν δεν υπήρχε αυτός ο counter, τότε πολλά πακέτα θα μπορούσαν να είναι πανομοιότυπα μεταξύ τους. Για παράδειγμα όσο η κατάσταση του συστήματος δεν αλλάζει, τότε όλα τα byte του πακέτου θα είναι ακριβώς τα ίδια. Αν και εντός του firmware στους slave δεν υπάρχει έλεγχος μοναδικότητας πακέτου, είναι κρίσιμο να προσφερθεί αυτή η επεκτασιμότητα, ώστε μελλοντικά να μπορεί να αποφευχθεί η επίθεση επανάληψης (replay attack), όπου κάποιος κακόβουλος αντιγράφει απλά ένα πακέτο χωρίς να γνωρίζει καν τα περιεχόμενά του και το στέλνει προς τον slave, «ξεγελώντας» τον ουσιαστικά (μέσω του πακέτου αυτού) και επιβάλλοντάς του μια συγκεκριμένη κατάσταση στη σειρά και τα ενδεικτικά του.

Διεύθυνση master (2 byte): Στο δίκτυο η διεύθυνση του master είναι πάντα 0. Εάν οι slave παραλάβουν οποιαδήποτε άλλη διεύθυνση εκτός από αυτήν, δεν απαντούν στο request.

Διεύθυνση slave (2 byte): Η διεύθυνση του slave που αναμένεται να απαντήσει στο request του master. Κάποιος slave θα απαντήσει εφόσον στο πεδίο αυτό αναγράφεται η διεύθυνσή του. Σε διαφορετική περίπτωση θα αγνοήσει το πακέτο. Συνεπώς αυτό το πεδίο είναι πολύ σημαντικό, διότι εξασφαλίζει την ομαλότητα της επικοινωνίας στο επίπεδο των συνομιλητών, ότι δηλαδή θα απαντήσει πάντα ο κατάλληλος κόμβος και όχι κάποιος άλλος.

Μέγεθος payload (1 byte): Αν και το μέγεθος του payload είναι πάντα το ίδιο, αυτό το byte παρέχει μια σημαντική επεκτασιμότητα ώστε να μπορούν μελλοντικά να προστεθούν και άλλα πεδία στο payload, χωρίς να χαθεί η συμβατότητα με τις παλαιότερες εκδόσεις προγράμματος (backwards compatibility). Ο slave θα θεωρήσει το πακέτο έγκυρο, μόνο σε περίπτωση που το μήκος του πακέτου συμφωνεί με το μέγεθος που «δηλώνεται» στο byte αυτό.

Payload (4 byte): Επεξηγείται αναλυτικά στην παράγραφο 5.3.2.

CRC-16 (2 byte): Δικλίδα ασφαλείας, ώστε οι πιθανότητες κάποιο αλλοιωμένο πακέτο να θεωρηθεί έγκυρο να είναι πρακτικά μηδαμινές (1:65536). Ο slave υπολογίζει το CRC του πακέτου που παρέλαβε και θα πρέπει το υπολογισμένο CRC, να συμφωνεί με το CRC που παραλήφθηκε από τον master στο πεδίο αυτό. Εάν δεν συμφωνεί, τότε το πακέτο απορρίπτεται ως ελαττωματικό.

5.3.2 Master payload

Στην εικόνα 18 φαίνεται το payload που στέλνεται από τον master προς τους slave. Πρόκειται για τα εξής τέσσερα byte:

Γενικός συναγερμός (1 byte): Από την τιμή του (0 ή 1) θα εξαρτηθεί η οδήγηση του αντίστοιχου ενδεικτικού συναγερμού στις πλακέτες των slave.

Γενικό σφάλμα (1 byte): Από την τιμή του (0 ή 1) θα εξαρτηθεί η οδήγηση του αντίστοιχου ενδεικτικού σφάλματος στις πλακέτες των slave.

Κατάσταση συστήματος (1 byte): Από την τιμή του (0, 1 ή 2) θα εξαρτηθεί η οδήγηση της σειρήνας στις πλακέτες των slave.

Εντολή (1 byte): Δεσμευμένο για μελλοντική χρήση, σε περίπτωση που ο master χρειάζεται να στείλει κάποια εντολή προς τους slave.

5.3.3 Πακέτο αποστολής slave (response packet)

Αποτελείται από συνολικά 30 bytes, όπως φαίνεται και στην εικόνα 19, τα οποία χωρίζονται σε 7 διαφορετικές θεματικές ενότητες, οι οποίες περιγράφονται παρακάτω:

Preamble (2 byte): Τα byte αυτά είναι πάντα τα ίδια (0x55, 0xAA). Σκοπός τους είναι ο συγχρονισμός του master στην αρχή του πακέτου. Εάν ο master θεωρήσει ότι έχει λάβει τα 2 πρώτα byte του εισερχόμενου πακέτου αλλά αυτά δεν συμφωνούν με το προκαθορισμένο slave preamble, τότε συνεχίζει να περιμένει τα 2 byte αυτά ώστε, όταν τα λάβει, να θεωρήσει ότι το εισερχόμενο πακέτο έχει ξεκινήσει να έρχεται.

Μετρητής αποστολής (4 byte): Με τη χρήση του προσδίδεται μια «μοναδικότητα» σε κάποιο πακέτο. Εάν δεν υπήρχε αυτός ο counter, τότε πολλά πακέτα θα μπορούσαν να είναι πανομοιότυπα μεταξύ τους. Για παράδειγμα όσο η μέτρηση του αισθητήρα έχει την ίδια τιμή, τότε όλα τα byte του πακέτου θα είναι ακριβώς τα ίδια. Αν και εντός του προγράμματος στον master δεν υπάρχει έλεγχος μοναδικότητας πακέτου, είναι κρίσιμο να προσφερθεί αυτή η επεκτασιμότητα, ώστε μελλοντικά να μπορεί να αποφευχθεί η επίθεση επανάληψης (replay attack), όπου κάποιος κακόβουλος αντιγράφει απλά ένα πακέτο χωρίς να γνωρίζει καν τα περιεχόμενά του και το στέλνει προς τον master, «ξεγελώντας» τον ουσιαστικά (μέσω του πακέτου αυτού) και επιβάλλοντάς του μια συγκεκριμένη κατάσταση συστήματος.

Διεύθυνση master (2 byte): Εάν ο master στο εισερχόμενο πακέτο διαβάσει διεύθυνση άλλη από τη δική του (0), τότε απορρίπτει το πακέτο, μη θεωρώντας το έγκυρο.

Διεύθυνση slave (2 byte): Η διεύθυνση του slave που απάντησε στο request του master. Εάν η διεύθυνση αυτή δεν είναι η ίδια με τη διεύθυνση που «έγραψε» ο master στο αντίστοιχο πεδίο του request, τότε σημαίνει ότι κάποιος άλλος slave απάντησε από αυτόν που αναμένονταν και συνεπώς ο master θα απορρίψει το πακέτο, μη θεωρώντας το έγκυρο.

Μέγεθος payload (1 byte): Αν και το μέγεθος του payload είναι πάντα το ίδιο, αυτό το byte παρέχει μια σημαντική επεκτασιμότητα, ώστε να μπορούν μελλοντικά να προστεθούν και άλλα πεδία στο payload, χωρίς να χαθεί η συμβατότητα με τις παλαιότερες εκδόσεις προγράμματος (backwards compatibility). Ο master θα θεωρήσει το πακέτο έγκυρο, μόνο σε

περίπτωση που το μήκος του πακέτου συμφωνεί με το μέγεθος που «δηλώνεται» στο byte αυτό.

Payload (4 byte): Επεξηγείται αναλυτικά στην παράγραφο 5.3.4.

CRC-16 (2 byte): Δικλίδα ασφαλείας, ώστε οι πιθανότητες κάποιο αλλοιωμένο πακέτο να θεωρηθεί έγκυρο να είναι πρακτικά μηδαμινές (1:65536). Ο master υπολογίζει το CRC του πακέτου που παρέλαβε και θα πρέπει το υπολογισμένο CRC, να συμφωνεί με το CRC που παραλήφθηκε από τον slave στο πεδίο αυτό. Εάν δεν συμφωνεί, τότε το πακέτο απορρίπτεται ως ελαττωματικό.

5.3.4 Slave payload

Στην εικόνα 19 φαίνεται ποιο είναι το payload που στέλνεται από τους slave προς τον master. Πρόκειται για 16 byte που για κάθε είδος αισθητήρα έχει διαφορετική ερμηνεία. Το payload ουσιαστικά δίνει την πληροφορία για τις μετρήσεις του κάθε αισθητήρα. Για κάποιους αισθητήρες κάποια πεδία είναι ανενεργά. Στέλνονται μεν, χωρίς όμως να έχουν κάποια συγκεκριμένη έννοια και για αυτόν το λόγο αγνοούνται και από τον master. Στην εικόνα 19 φαίνονται ως «dummy».

5.3.5 Διαχωρισμός σε επίπεδα (L1 – L2)

Το πρωτόκολλο είναι χωρισμένο σε δύο βασικά επίπεδα, το επίπεδο 2 (L2) και το επίπεδο 1 (L1). Το L2 (υψηλό επίπεδο) έχει να κάνει με τα ληφθέντα αντικείμενα (πχ κατάσταση συστήματος) και την αποθήκευσή τους σε κατάλληλες μεταβλητές, ενώ το χαμηλότερο επίπεδο L1 διαχειρίζεται τα πεδία ελέγχου του πρωτοκόλλου (CRC, preamble κτλ). Με αναφορά τις εικόνες 18 και 19, τα πεδία στα οποία αναγράφεται με κόκκινο «payload» ανήκουν στο L2 και αφορούν την εφαρμογή αυτή κάθε αυτή. Όλα τα υπόλοιπα ανήκουν στο L1 και αφορούν αποκλειστικά τη λειτουργία του δικτύου.

Το L2 δεν έχει πρόσβαση στις πληροφορίες του L1 και στις εσωτερικές διεργασίες του πρωτοκόλλου. Δεν το αφορούν. Αυτό που χρειάζεται από το L1 είναι απλά να του δώσει τα αντικείμενα (payload) ως ορίσματα συναρτήσεων, ώστε το L1 να τα στείλει στο δίκτυο. Όπως και όταν υπάρχει εισερχόμενο frame, να παραλάβει τα ληφθέντα αντικείμενα από το L1.

Το L1 έχει την ευθύνη πριν στείλει τα αντικείμενα του L2 στο δίκτυο, να τα ενθυλακώσει (encapsulate) στο ευρύτερο frame προς αποστολή (εικόνες 18 και 19), ενώ όταν παραλάβει frame από κάποιον slave θα πρέπει να αποφλοιώσει (de-capsulate) τα αντικείμενα και να τα στείλει στο L2. Πριν όμως, θα πρέπει να κάνει όλους τους απαραίτητους ελέγχους ότι το παραληφθέν πακέτο είναι έγκυρο. Σε αντίθετη περίπτωση, επιστρέφει ένα κενό μήνυμα στο L2, ώστε το τελευταίο να διαχειριστεί αυτήν την περίπτωση κατάλληλα, σηματοδοτώντας για παράδειγμα, αν χρειαστεί, σφάλμα επικοινωνίας.

5.4 Δικλίδες διασφάλισης πακέτου

Οι δικλίδες αφορούν τόσο τον master (ΚΕΣ), όσο και τον slave (πλακέτα αισθητήρα). Προκύπτουν από την περιγραφή του πακέτου που παρατέθηκε στις προηγούμενες ενότητες του κεφαλαίου και συνοψίζονται παρακάτω:

- CRC16, ώστε να εξασφαλίζεται η ορθότητα των δεδομένων.
- Συγκεκριμένο start και stop byte, ώστε να αναγνωρίζεται η αρχή και το τέλος του πακέτου.
- Υπάρχει πληροφορία εντός του ίδιου του πακέτου για το μέγεθός του, ώστε ο δέκτης να είναι σίγουρος ότι έλαβε όσα byte του έστειλε ο αποστολέας.

6 Hardware

Όλα τα ηλεκτρονικά σχέδια (schematics και PCB) περιέχονται στο παραδοτέο της εργασίας τόσο στη μορφή αρχείου σχεδίου, όσο και σε pdf. Όλα τα σχέδια πραγματοποιήθηκαν στο σχεδιαστικό Pulsonix v9.1. Παρακάτω παρατίθενται τα σημαντικότερα κομμάτια που χρήζουν ανάλυσης και τεκμηρίωσης.

6.1 Θεωρητικό σχέδιο πλακέτας αισθητήρα

6.1.1 Αναγνώριση βαθμίδων

Οι κύριες βαθμίδες του σχεδίου του αισθητήρα είναι:

- Τροφοδοσία
- Οδηγός RS-485
- Περιβαντολλογικός αισθητήρας
- Αισθητήρας θερμοκρασίας
- Οπτικοακουστικά μέσα (ψηφιακές είσοδοι / έξοδοι)

Το υπόλοιπο σχέδιο δεν χαρακτηρίζεται από άλλες βαθμίδες, παρά μεμονωμένες συνδέσεις (πχ κρύσταλλος μικροελεγκτή, μπουτόν reset κτλ.), τα οποία δεν χρήζουν ιδιαίτερης ανάλυσης.

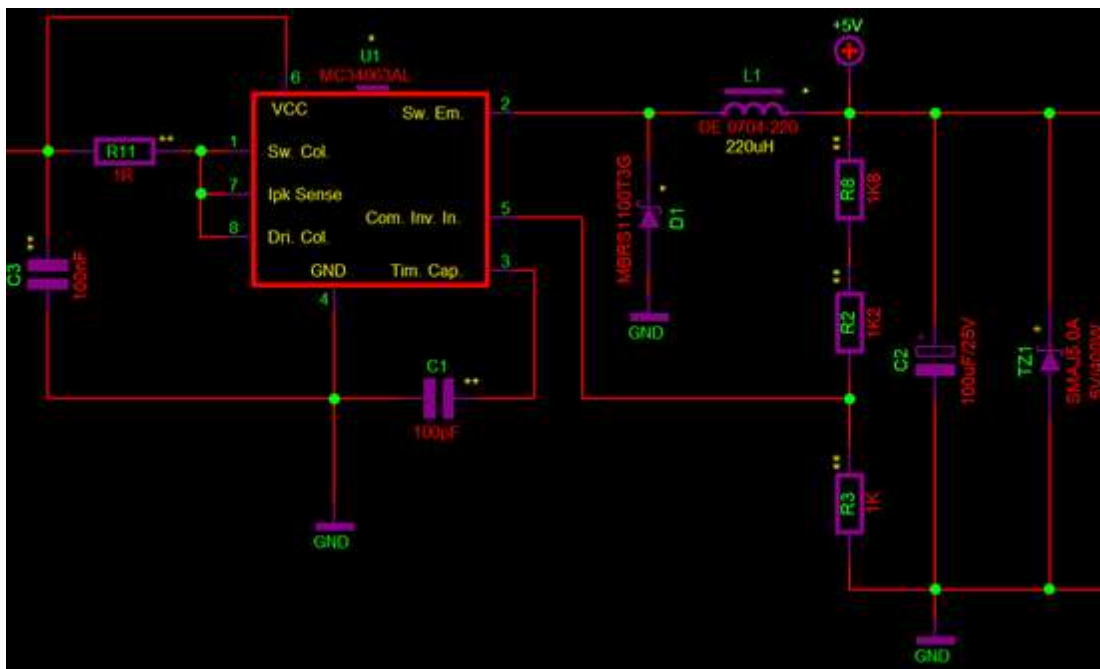
6.1.2 Τροφοδοσία

Η εξωτερική τροφοδοσία της πλακέτας είναι 12VDC, τα οποία υποβαθμίζονται εντός της πλακέτας σε 5V, με τη χρήση ενός DC/DC converter τύπου switching regulator. Δεν επιλέχτηκε τυχαία αυτός ο τύπος τροφοδοτικού. Τα 5V που παράγει θα οδηγηθούν στην τροφοδοσία του αισθητήρα SEN55, ο οποίος έχει μέση κατανάλωση περίπου 50mA. Αν επιλέγονταν γραμμικό τροφοδοτικό, τότε η ισχύς που θα καταναλώνονταν επάνω του θα ήταν

$$W = V * I = (12V - 5V) * 50mA \Rightarrow W = 350mW$$

Τα 350mW θα δημιουργούσαν έντονη θέρμανση στο τροφοδοτικό και συνεπώς για να αποφευχθεί αυτό, χρησιμοποιείται τροφοδοτικό με διακοπτική έξοδο και συγκεκριμένα το MC34063 της On Semiconductors. Τα διακοπτικά τροφοδοτικά (switching regulators), αν και πιο πολύπλοκα από τα γραμμικά, δεν υποφέρουν εκ φύσεως από υπερθέρμανση εξαιτίας της πτώσης τάσης στα άκρα τους. Ο διακόπτης εξόδου μένει για λίγο κλειστός και μετά ανοίγει και δεν παρέχει έξοδο, αφήνοντας τα ενεργητικά και παθητικά στοιχεία της εξόδου (δίοδο, πηνίο και πυκνωτή) να διατηρήσουν την τάση στα επιθυμητά επίπεδα, μέσω της δυναμικής τους συμπεριφοράς.

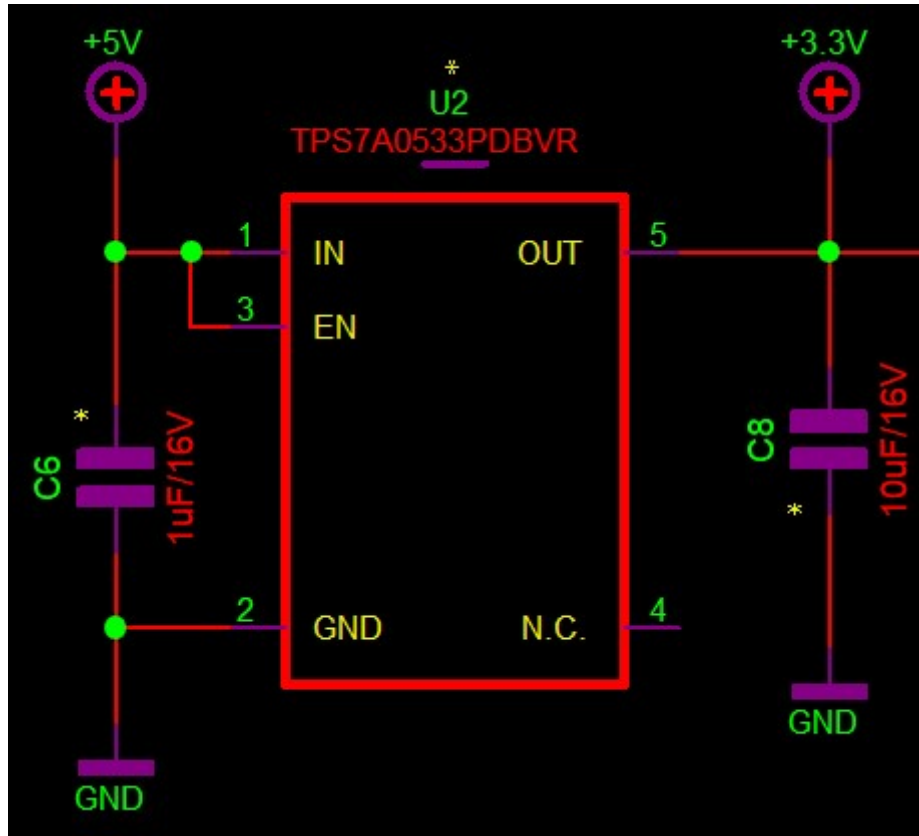
Το U1 είναι το MC34063 (σχήμα 20). Η αντίσταση R111 αριστερά του U1 είναι για τον περιορισμό του ρεύματος εισόδου στα 350mA περίπου. Ο C1 κάτω ακριβώς από το U1 χρησιμοποιείται για τον χρονισμό του διακόπτη εξόδου. Στα δεξιά του U1 είναι το κύκλωμα εξόδου. Συγκεκριμένα η D1 σε συνδυασμό με το L1 δημιουργούν υπερτάσεις (voltage spikes) στην έξοδο, τα οποία ολοκληρώνονται στον πυκνωτή C2, ώστε τα να δημιουργηθεί η DC τάση εξόδου. Η μεσαία λήψη του διαιρέτη τάσης που ορίζεται από τις R8 – R2 και R3 οδηγείται σε εσωτερικό συγκριτή του U1, ώστε η στάθμη της τάσης να είναι στα 5V DC [21]. Τέλος η TZ1 είναι δίοδος TVS, η οποία θα γίνει αγώγιμη μόνο στην περίπτωση που η έξοδος ανέβει πάνω από τα 5V, ώστε να μην την αφήσει να ανέβει παραπάνω, με κίνδυνο να καταστρέψει κάποιο από τα υλικά της πλακέτας και κυρίως τον αισθητήρα SEN55, ο οποίος είναι και το πιο ακριβό υλικό της πλακέτας.



Εικόνα 20 - LDO regulator 5V DC σε 3.3V DC

Τα 5V τροφοδοτούν αποκλειστικά τον αισθητήρα. Τα υπόλοιπα κυκλώματα τροφοδοτούνται από το γραμμικό τροφοδοτικό U2 με κωδικό TPS7A0533PDBVR (εικόνα 21), τύπου LDO (Low Dropout), το οποίο στην ουσία είναι ένας ενισχυτής σφάλματος με εσωτερική τάση αναφοράς [22]. Οι τιμές των πυκνωτών εισόδου και εξόδου έχουν επιλεγεί σύμφωνα με τις

οδηγίες του κατασκευαστή [22]. Ο κίνδυνος υπερθέρμανσης του τροφοδοτικού δεν υπάρχει σε αυτήν την περίπτωση, αφού το κύκλωμα που τροφοδοτεί δεν καταναλώνει παραπάνω από 10mA, τα οποία με 1.7V πτώση τάσης στα άκρα του U2, δίνουν μόλις 17mW.

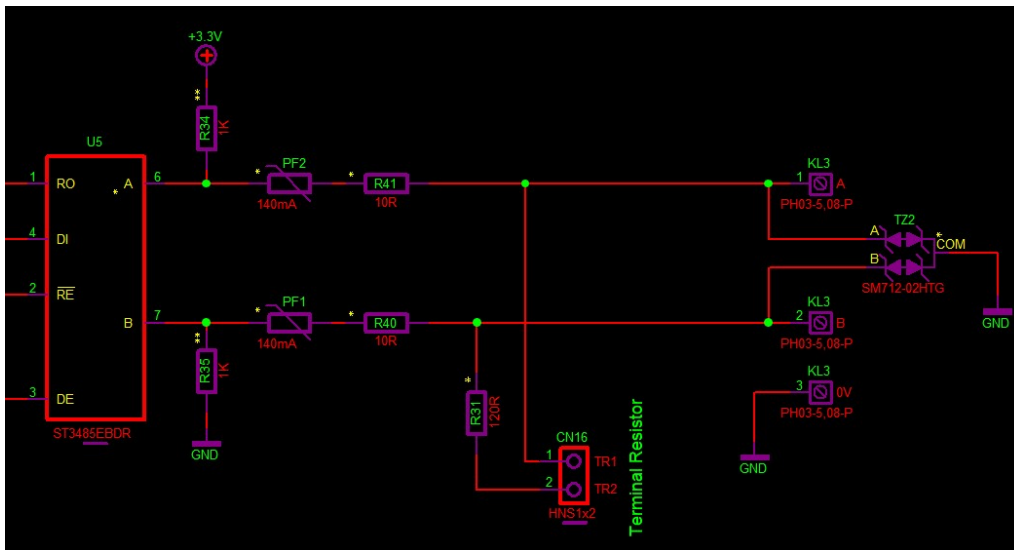


Εικόνα 21 - LDO regulator 5V DC σε 3.3V DC

6.1.3 Οδηγός RS-485

Στην εικόνα 22 φαίνεται το κύκλωμα του RS-485, το οποίο χρησιμοποιεί τον οδηγό (driver) ST3485EBDR της ST Microelectronics (U5). Τα pins στην αριστερή πλευρά του U5 είναι στη στάθμη των 3.3V. Συνδέονται απευθείας στον μικροελεγκτή και είναι τα pins λήψης και αποστολής της σειριακής θύρας, καθώς και σήματα ελέγχου για το πότε ο driver θα λαμβάνει και πότε θα στέλνει [23].

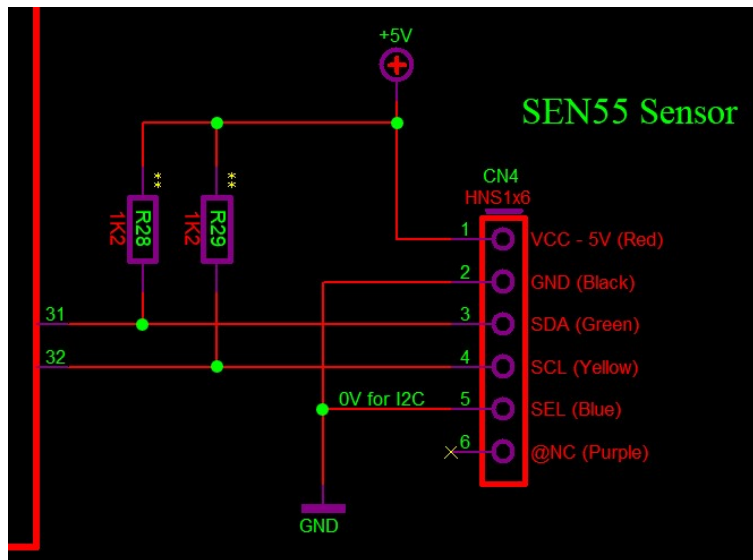
Το μεγαλύτερο ενδιαφέρον παρουσιάζεται στην πλευρά που βρίσκονται οι στάθμες του RS-485 (δεξιά του U5), όπου το πιο κρίσιμο λειτουργικά κομμάτι είναι η αντίσταση R31 (120 Ohm). Όταν η πλακέτα αποτελεί τερματικό κόμβο του δικτύου RS-485, η γραμμή πρέπει να τερματιστεί με 120 Ohm και συνεπώς θα πρέπει να γεφυρωθεί ο βραχυκυκλωτήρας (jumper) CN16. Εάν η πλακέτα δεν είναι τερματικός κόμβος, τότε το jumper θα πρέπει να παραμείνει ασύνδετο. Τα PVC PF1, PF2 με τις αντιστάσεις R41, R42 είναι προστασίες σειράς στη γραμμή (για τις αιχμές ρεύματος), ενώ το T22 προστατεύει τη γραμμή από υπερτάσεις. Οι αντιστάσεις R34 και R35 θα δώσουν δυναμικό στη γραμμή, σε περίπτωση που η έξοδος παραμείνει ασύνδετη, ώστε τα pins του driver να μη μείνουν χωρίς δυναμικό. Κατά τα άλλα δεν παίζουν κάποιον άλλο λειτουργικό ρόλο.



Εικόνα 22 - Οδηγός RS-485

6.1.4 Περιβαντολογικός αισθητήρας

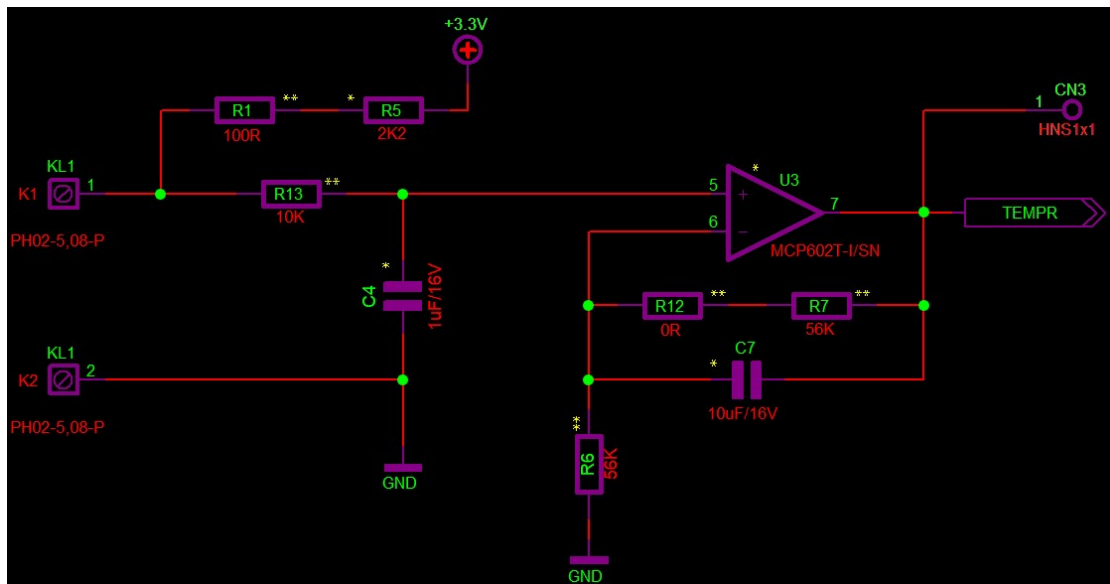
Στην εικόνα 23 φαίνεται η σύνδεση του μικροελεγκτή με το SEN55. Είναι οι γραμμές τροφοδοσίας (5V DC) και του πρωτοκόλλου I²C, όπως εξηγήθηκαν στην παράγραφο 4.3.1.



Εικόνα 23 - Συνδέσεις αισθητήρα SEN55

6.1.5 Αισθητήρας θερμοκρασίας

Χρησιμοποιήθηκε ο KTY81-110, ο οποίος συνδέεται στα pins 1 και 2 της κλέμματος KL1 (εικόνα 24).



Εικόνα 24 - Συνδέσεις αισθητήρα KTY81-110

Εφόσον ο KTY81-110 συνδέεται στην KL1 δημιουργεί διαιρέτη τάσης με τις R1 και R5. Η τάση αυτή οδηγείται στη μη αναστρέφουσα είσοδο του μη αναστρέφοντα ενισχυτή U3, ο οποίος έχει απολαβή 2, σύμφωνα με τις τιμές των R12 και R7. Αυτό διότι για την έξοδο του μη αναστρέφοντα ενισχυτή ισχύει:

$$V_o = V_{in} * (1 + (R_6 / R_7)) \Rightarrow (V_o/V_{in}) = 2$$

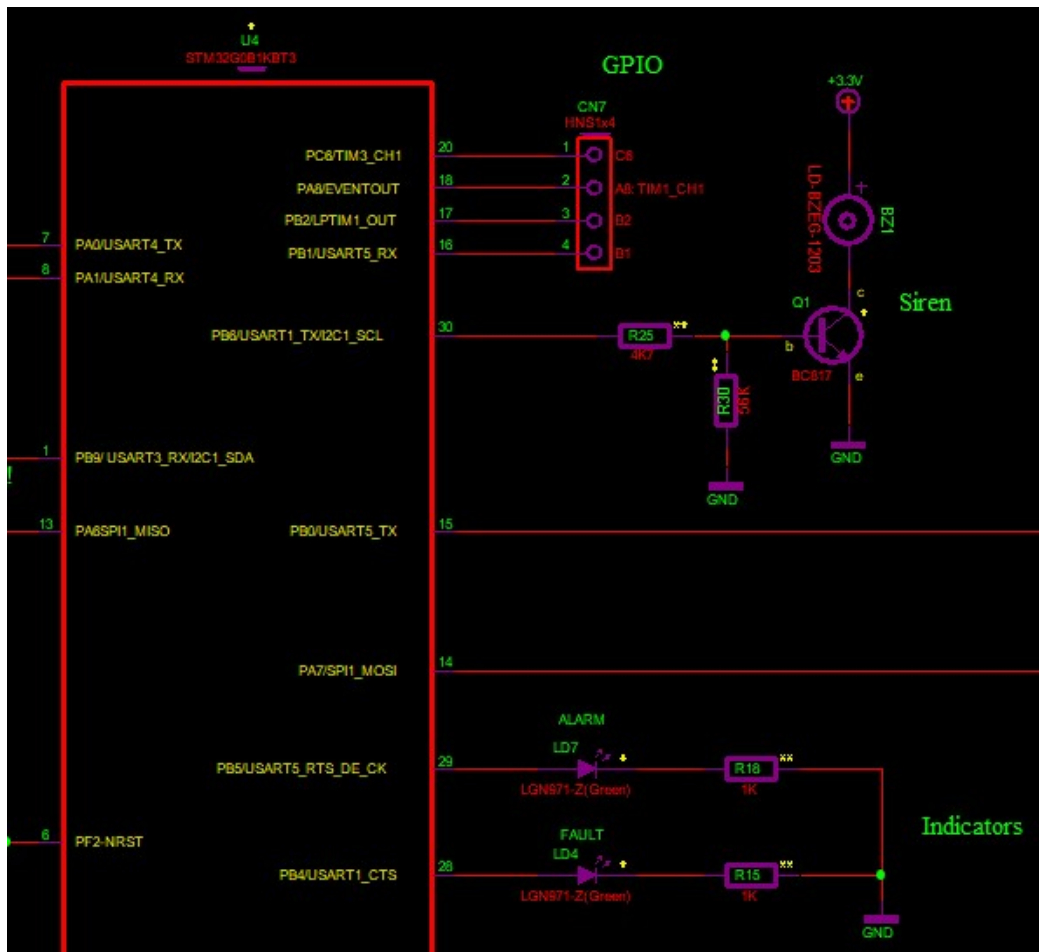
Ο C7 στην ανάδραση του ενισχυτή, σε συνδυασμό με τις αντιστάσεις στην ανάδραση, δημιουργεί ένα χαμηλοπερατό φίλτρο, ώστε να μειώνει της αναπηδήσεις τις εξόδου. Η θερμοκρασία άλλωστε είναι ένα αργό σε μεταβολή φαινόμενο, οπότε η χρήση ενός μεγάλου πυκνωτή τάξεως μF ενδείκνυται για την περίπτωση αυτή.

6.1.6 Ψηφιακοί είσοδοι – έξοδοι

Αποτελείται από τον βομβητή (BZ1), δύο ενδεικτικά LED για σφάλμα και συναγερμό (LD7 και LD4 αντίστοιχα) και 4 εισόδους / εξόδους για μελλοντική χρήση (CN7).

Στην εικόνα 25 φαίνεται ότι η σειρήνα οδηγείται από ένα τρανζίστορ ηρη σε συνδεσμολογία κοινού εκπομπού. Αυτό σημαίνει ότι όταν η έξοδος του μικροελεγκτή είναι 0, η βάση του τρανζίστορ δεν διαρρέεται από ρεύμα, το τρανζίστορ δεν είναι αγώγιμο και συνεπώς ο βομβητής δεν ηχεί. Στην περίπτωση που η έξοδος είναι 1, το τρανζίστορ γίνεται αγώγιμο, επιτρέποντας το ρεύμα να περάσει από τον συλλέκτη προς τον εκπομπό και κατά συνέπεια το ρεύμα αυτό θα περάσει και μέσα από τον βομβητή, κάνοντάς τον να ηχεί.

Για τα ενδεικτικά LED (εικόνα 25) δεν υπάρχει κάτι ιδιαίτερο να ειπωθεί, αφού η οδήγησή τους είναι μέσω μιας αντίστασης. Η αντίσταση επιλέχθηκε με βάση τη φωτεινότητα των LED όταν είναι οη. Οι άνοδοι των LED συνδέονται με την έξοδο του μικροελεγκτή, που σημαίνει ότι τα LED είναι οη μόνο όταν οι έξοδοι που τα οδηγούν είναι 1.



Εικόνα 25 - Συνδέσεις ψηφιακών εισόδων - εξόδων

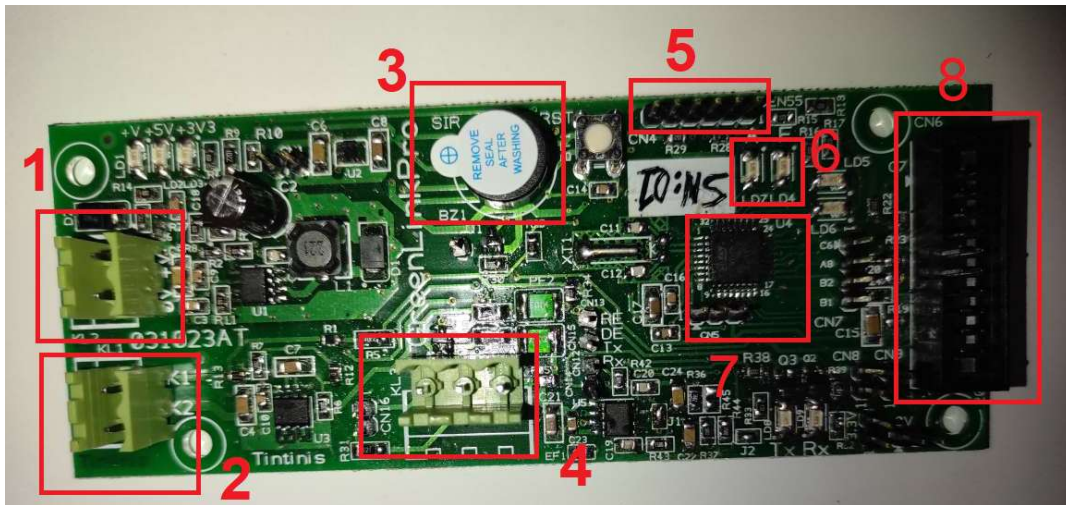
6.2 Πλακέτα αισθητήρα (PCB)

Η κεντρική τροφοδοσία έρχεται από εξωτερικό τροφοδοτικό πρίζας 230V AC σε 12VDC / 0.6A, το οποίο φαίνεται στην εικόνα 26.



Εικόνα 26 - Εξωτερικό τροφοδοτικό πρίζας 12V DC / 16A της πλακέτας του αισθητήρα

Στην εικόνα 27 φαίνεται σε φωτογραφία η πλακέτα του αισθητήρα, με τα σημαντικότερά της μέρη εντός των κόκκινων αριθμημένων πλαισίων.



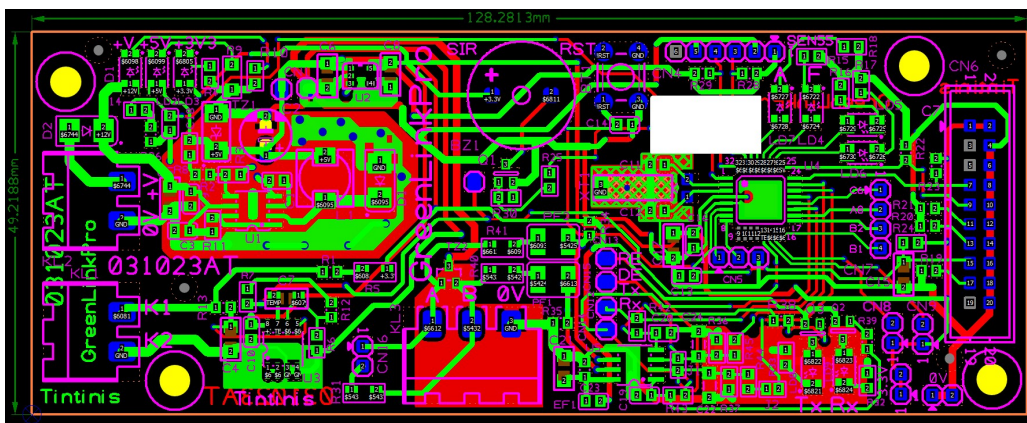
Εικόνα 27 - Εξωτερικό τροφοδοτικό πρίζας 12V DC / 16A της πλακέτας του αισθητήρα

Τα κόκκινα πλαίσια αντιστοιχούν στα εξής:

1. Κλέμμα εισόδου τροφοδοσίας 12V.
2. Κλέμμα σύνδεσης αισθητήρα θερμοκρασίας.
3. Βομβητής (buzzer).
4. Κλέμμα σημάτων RS-485.
5. Connector σύνδεσης περιβαλλοντολογικού αισθητήρα.
6. Ενδεικτικά LED για συναγερμό και σφάλμα.
7. Μικροελεγκτής.
8. Κλέμμα προγραμματισμού του μικροελεγκτή.

6.2.1 Σχέδιο πλακέτας

Η κάτοψη της πλακέτας, παρμένη από το σχεδιαστικό πρόγραμμα, φαίνεται στην εικόνα 28.



Εικόνα 28 - Κάτοψη της πλακέτας όπως φαίνεται εντός του σχεδίου PCB

6.2.2 Ενδεικτικό κόστος πλακέτας αισθητήρα

Για να υπολογιστεί το κόστος της πλακέτας, χρειάζεται να κάνουμε αρχικά μια λίστα με τα πιο κοστοβόρα υλικά και για τα δύο ήδη αισθητήρα. Οι τιμές παρακάτω είναι ενδεικτικές, αφού ποικίλλουν από προμηθευτή σε προμηθευτή. Ο προμηθευτής αναφέρεται πάντα δίπλα από το κόστος. Επίσης κάποιες τιμές που ήταν σε δολάρια μετατράπηκαν σε ευρώ, με την ισχύουσα ισοτιμία κατά τη συγγραφή της εργασίας. Τα κόστη συνεπώς είναι όλα σε Ευρώ. Τέλος είναι σημαντικό να αναφερθεί ότι το κόστος που θα υπολογιστεί είναι αυτό που ονομάζεται «κόστος πρωτότυπου». Σε περίπτωση μαζικής παραγωγής το κόστος αυτό θα μειωθεί δραματικά, αφού οι μεγάλες ποσότητες υλικών στις παραγγελίες, οδηγούν και σε χαμηλότερες τιμές αγοράς των υλικών (τιμές χονδρικής).

- **Module αισθητήρα SEN55: 30.75E** (Sensirion) – Μόνο για τον περιβατολογικό αισθητήρα
- **Αισθητήρας KTY81-110: 1.57E** (Mouser) – Μόνο για τον αισθητήρα θερμοκρασίας
- **Operational Amplifier MCP602-I/SN: 0.77E** (TME) – Μόνο για τον αισθητήρα θερμοκρασίας
- **Πλακέτα αισθητήρα: 23.99E** (Micron 20)
- **STM32G0B1KBT3 MCU: 5.47E** (Mouser)
- **3 ζευγάρια κλεμμες με 3 pin (αρσενικά – θηλυκά): 2E** (TME)
- **ST3485EBDR RS-485 driver: 1.52E** (Mouser)
- **TVS Diode SM712-02HTG: 0.72E** (Mouser)
- **Βομβητής LD-BZEG-1203: 0.55E** (TME)
- **LDO TPS7A0533PDBVR: 0.34E** (TME)
- **PTC MINISMDC014F-2 140mA: 0.18E** (Farnell)
- **Πυκνωτής 100uF/35V SMD: 0.14E** (TME)
- **Υπόλοιπα υλικά (αντιστάσεις, πυκνωτές, δίοδοι, τρανζίστορ κτλ): 2.10E**

Σύνολο υλικών: 70,10 E

Σύνολο πλακέτας αισθητήρα θερμοκρασίας: 39,35 E

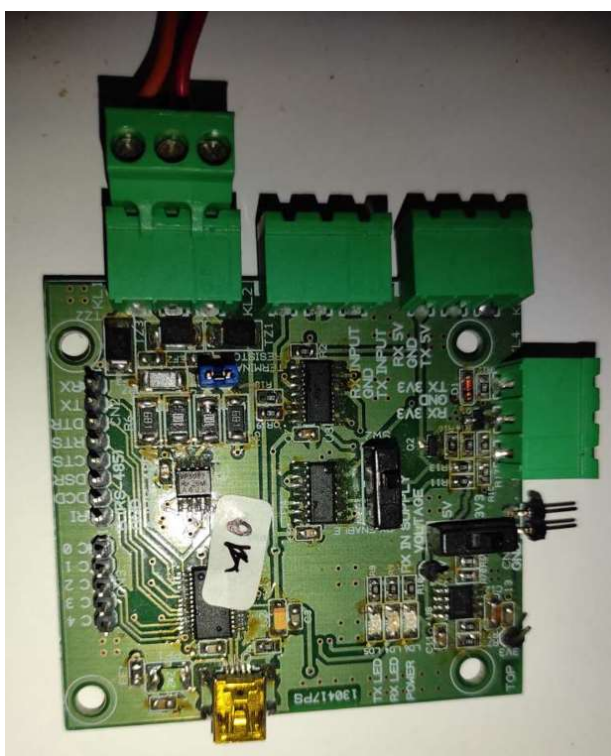
Σύνολο πλακέτας περιβατολογικού αισθητήρα: 67,76 E

Τα παραπάνω κόστη καταδεικνύουν τον λόγο που επιλέχτηκε να αναπτυχθεί παραπάνω από ένα είδος αισθητήρα στην εφαρμογή. Ο περιβατολογικός αισθητήρας έχει μεγάλο κόστος υλικών και συνεπώς θα ήταν ασύμφορο να τοποθετηθούν για παράδειγμα εκατοντάδες τέτοιοι αισθητήρες σε σημεία που χρειάζονται πιο απλές μετρήσεις. Πχ σε κάποιο σημείο μπορεί να ενδιαφέρει να τοποθετηθεί όντως πολύ-αισθητήρας, αλλά σε άλλα σημεία να ενδιαφέρει μόνο η μέτρηση της θερμοκρασίας ή η μέτρηση μόνο κάποιου αερίου (πχ μεθάνιο) κοκ. Συνεπώς με την υποστήριξη πολλαπλών τύπων αισθητήρων, δίνεται η δυνατότητα στην εγκατάσταση να προσαρμοστεί κοστολογικά ακριβώς στις ανάγκες του εκάστοτε χώρου.

6.3 Προσαρμογέας δικτύου RS-485

6.3.1 Περιγραφή λειτουργίας

Ο προσαρμογέας δικτύου RS-485 (εικόνα 29) είναι απλός ως κατανόηση. Συνδέεται από τη μία άκρη στο δίκτυο RS-485 και από την άλλη στο USB. Μετατρέποντας τις στάθμες USB σε στάθμες RS-485 και τούμπαλιν, συνδέει ουσιαστικά τους 2 αυτούς «διαφορετικούς κόσμους» από άποψη ηλεκτρικής τάσης σημάτων, με συνέπεια να επιτρέπει την επικοινωνία μεταξύ Η/Υ (USB) και αισθητήρων (RS-485). Η «καρδιά» του συστήματος είναι το FT232RL, το οποίο είναι ένα serial to UART interface. Εκτός από τη σύνδεση στην πλακέτα, απαιτεί και κατάλληλη ρύθμιση στα pins του, μέσω του προγράμματος FTPROG, το οποίο προσφέρεται δωρεάν από την κατασκευάστρια εταιρία (FTDI).



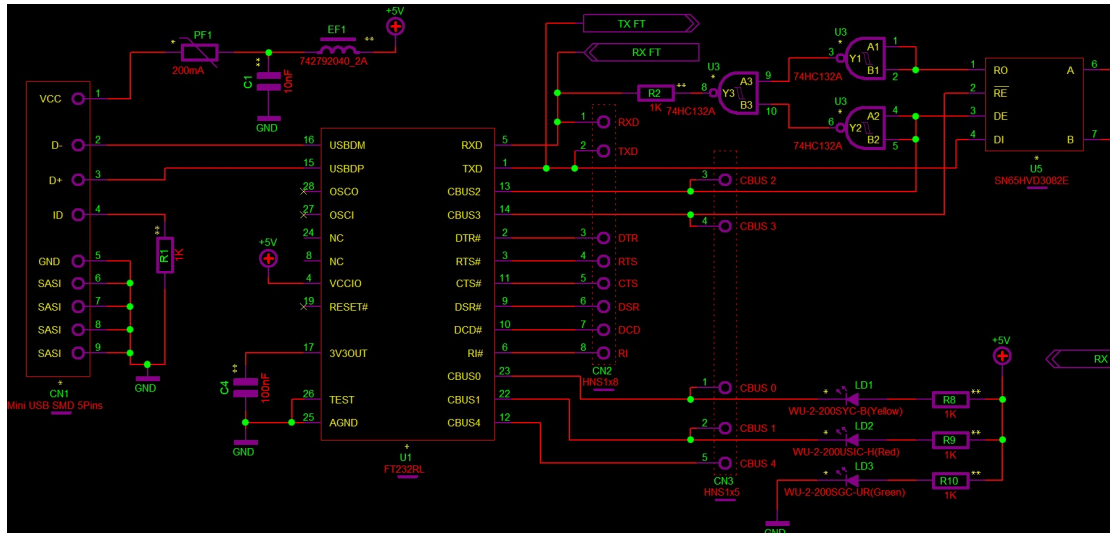
Εικόνα 29 - Φωτογραφία προσαρμογέας δικτύου

Να σημειωθεί ότι ο προσαρμογέας δεν θα αναλυθεί σε τόσο βάθος όπως ο αισθητήρας. Ο λόγος είναι ότι θα μπορούσε κανείς να αγοράσει και έτοιμους αντίστοιχους προσαρμογείς από το εμπόριο, αλλά και επίσης σε ένα δίκτυο ακόμα και εκατοντάδων αισθητήρων, ο προσαρμογέας θα είναι και πάλι ένας, έχοντας συνεπώς ασήμαντη κοστολογική βαρύτητα στο συνολικό δίκτυο.

6.3.2 Ηλεκτρονικά σχέδια

Τα σχέδια του προσαρμογέα επισυνάπτονται στο παραδοτέο της εργασίας. Η «καρδιά» του προσαρμογέα φαίνεται στην εικόνα 30. Το U1 στο κέντρο της εικόνας, μετατρέπει τη στάθμη USB σε στάθμες 3.3V. Το U5 στα δεξιά κάνει τη μετατροπή από τα 3.3V σε στάθμες RS-485.

Οι λογικές πύλες (U3) ανάμεσα στο U1 και U5 εφαρμόζουν μια απλή λογική, ώστε τα εισερχόμενα δεδομένα να φτάνουν στο USB μόνο όταν όντως στέλνεται κάτι από τους άλλους κόμβους του δικτύου RS-485 και ταυτόχρονα η λήψη να είναι ενεργοποιημένη από το ίδιο το U1. Το CN1 (αριστερά) είναι ο USB connector, ενώ στην κλέμμη KL1 (δεν εμφανίζεται στην εικόνα) συνδέονται τα σήματα A, B του RS-485.



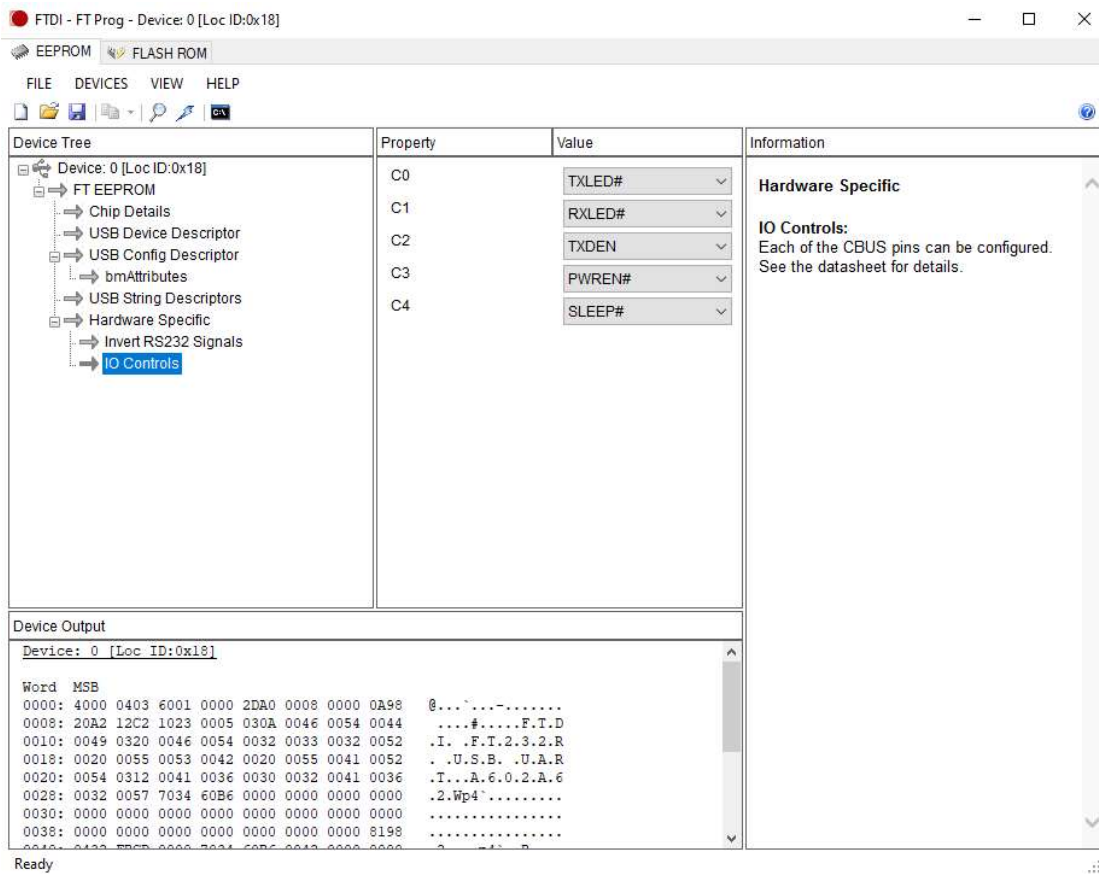
Εικόνα 30 - Προσαρμογέας δικτύου: σύνδεση FT232R με USB και RS-485

Για να επιτευχθεί ο «αυτοματισμός» της ενεργοποίησης – απενεργοποίησης των εισερχόμενων δεδομένων, θα πρέπει να γίνουν και κάποιες ρυθμίσεις στο ίδιο το U1. Η δυνατότητα για την πραγματοποίηση των ρυθμίσεων αυτών προσφέρονται από το πρόγραμμα «FT_PROG» της κατασκευάστριας εταιρίας του FT232R, που είναι η FTDI. Το πρόγραμμα προσφέρεται δωρεάν.

Οι ρυθμίσεις που πρέπει να γίνουν είναι για τα GPIO του U1 (εικόνα 31). Συγκεκριμένα:

- C0: Ενδεικτικό LED αποστολής πακέτων από τη θύρα USB. Είναι οn μόνο όταν πραγματοποιείται αποστολή από το USB προς το RS-485.
- C1: Ενδεικτικό LED λήψης πακέτων από τη θύρα USB. Είναι οn μόνο όταν πραγματοποιείται αποστολή από το RS-485 προς το USB.
- C2: Ενεργοποίηση αποστολής από τη θύρα USB. Η αποστολή από το USB προς το RS-485, πραγματοποιείται μόνο όταν το συγκεκριμένο pin είναι 0.
- C3: Ενεργοποίηση λήψης από τη θύρα USB. Η αποστολή από το RS-485 προς το USB, πραγματοποιείται μόνο όταν το συγκεκριμένο pin είναι 1.

Οι παραπάνω λειτουργίες στα συγκεκριμένα pin δίνονται από τον κατασκευαστή, την FTDI [24]. Αυτές οι λειτουργίες θα πρέπει να προγραμματιστούν στον driver μέσω του FT_PROG. Έχοντας την πλακέτα συνδεδεμένη στο USB, μπορεί να πραγματοποιηθεί σύνδεση με τον driver, ώστε να υλοποιηθούν οι ρυθμίσεις για τη λειτουργία αυτών των pin. Στην εικόνα 31 φαίνεται ότι για τα pin C0, C1, C2, C3 έγιναν οι ρυθμίσεις σύμφωνα με την παραπάνω περιγραφή, ενώ το C4 που δεν χρησιμοποιείται, παρέμεινε στις εργοστασιακές του ρυθμίσεις.



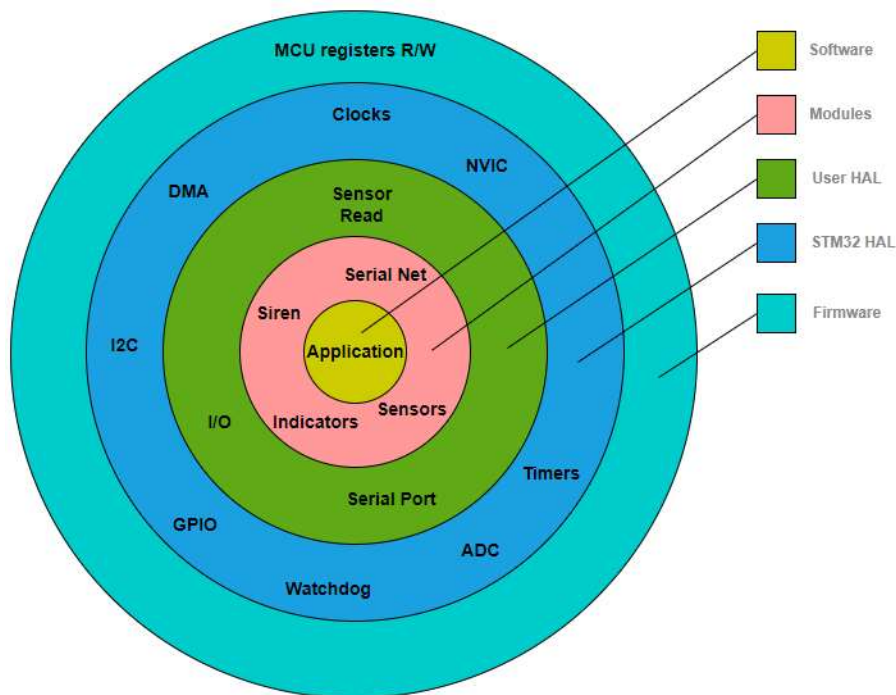
Εικόνα 31 - Προσαρμογέας δικτύου: σύνδεση FT232R με USB και RS-485

Συμπερασματικά, ο προσαρμογέας είναι μια απλή στην κατανόηση συσκευή, με το μόνο κρίσιμο σημείο να είναι η σωστή ρύθμιση των GPIO του, οι οποίες θα χρειαστεί να υποστηριχτούν από απλά εξωτερικά κυκλώματα (ενδεικτικά LED και λογικές πύλες).

7 Firmware αισθητήρα

7.1 Οργάνωση προγράμματος

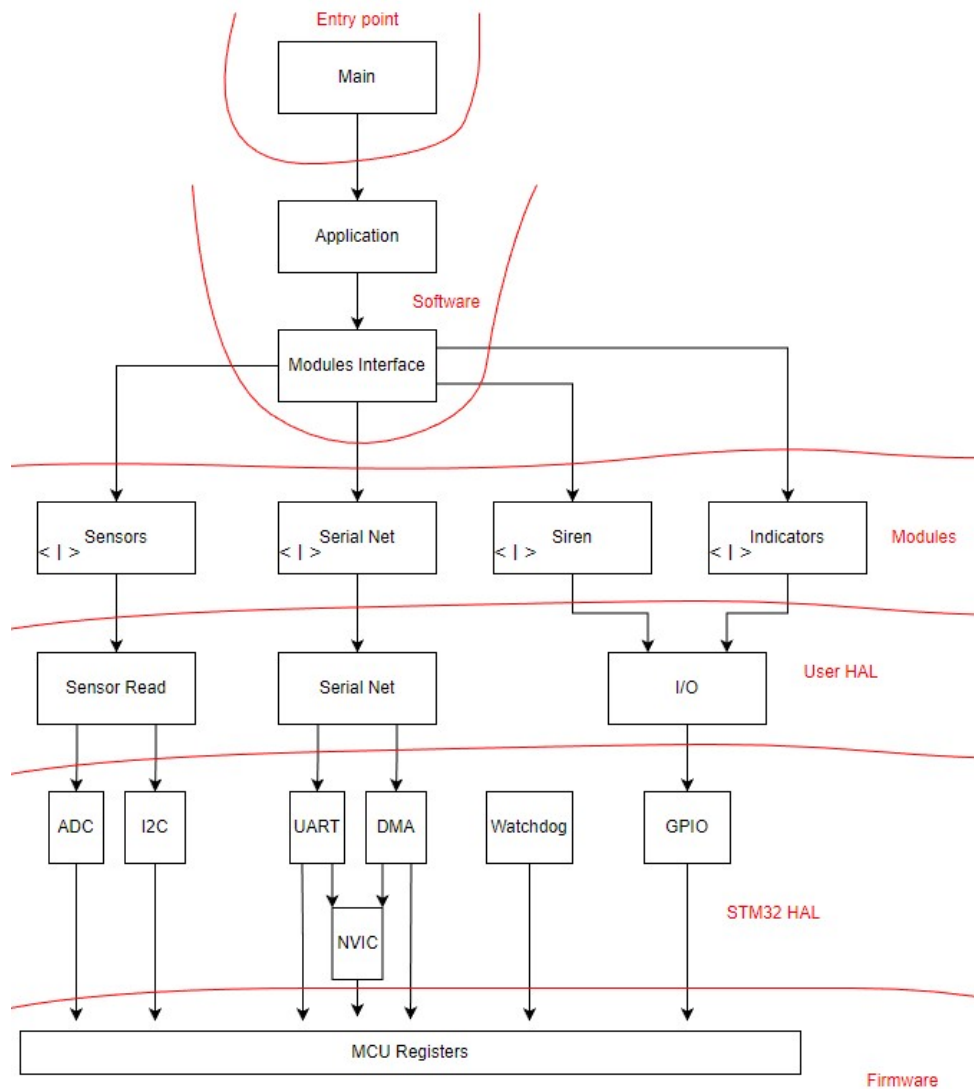
Τα διαστρωματωμένα επίπεδα του προγράμματος του αισθητήρα φαίνονται στην εικόνα 32. Όπως εύκολα διακρίνεται, το firmware του αισθητήρα έχει χωριστεί σε 5 διαφορετικά επίπεδα, τα οποία περιγράφονται στις αμέσως παρακάτω παραγράφους. Να τονιστεί ότι η τελική κατάτμηση των επιπέδων και των υπο-μονάδων (modules) που αυτά περιλαμβάνουν, έγινε με βάση τη μέθοδο της «Καθαρής Αρχιτεκτονικής» (Clean Architecture). Σύμφωνα με αυτήν τη μεθοδολογία, τα ανώτερα στρώματα του προγράμματος δεν μπορούν να εξαρτιούνται από τα κατώτερα.



Εικόνα 32 - Διαστρωμάτωση προγράμματος αισθητήρα με βάση το Clean Architecture [25]

Στην εικόνα 33 φαίνεται με περισσότερη λεπτομέρεια ποια είναι τα συστατικά (components) του κάθε επιπέδου προγράμματος. Να σημειωθεί ότι σε αυτό το σχήμα φαίνεται ακόμα ένα επίπεδο, αυτό του σημείου εισαγωγής (entry point). Δεν αποτελεί στην πραγματικότητα ξεχωριστό επίπεδο, αλλά λογίζεται ως ξεχωριστό component από όλα τα υπόλοιπα, αφού η main.c (που είναι το entry point) παράγεται αυτόματα από .ios αρχείο και συνεπώς δεν έχει καμία απολύτως σχέση με την υπόλοιπη αρχιτεκτονική του προγράμματος. Το πρόγραμμα εισέρχεται στην main(), όπου πραγματοποιούνται (και πάλι αυτόματα παραγόμενες) οι αρχικοποιήσεις του hardware και όταν το πρόγραμμα ολοκληρώσει με τις αρχικοποιήσεις αυτές, διακλαδίζεται εντός του application layer, από όπου δεν επιστρέφει ποτέ.

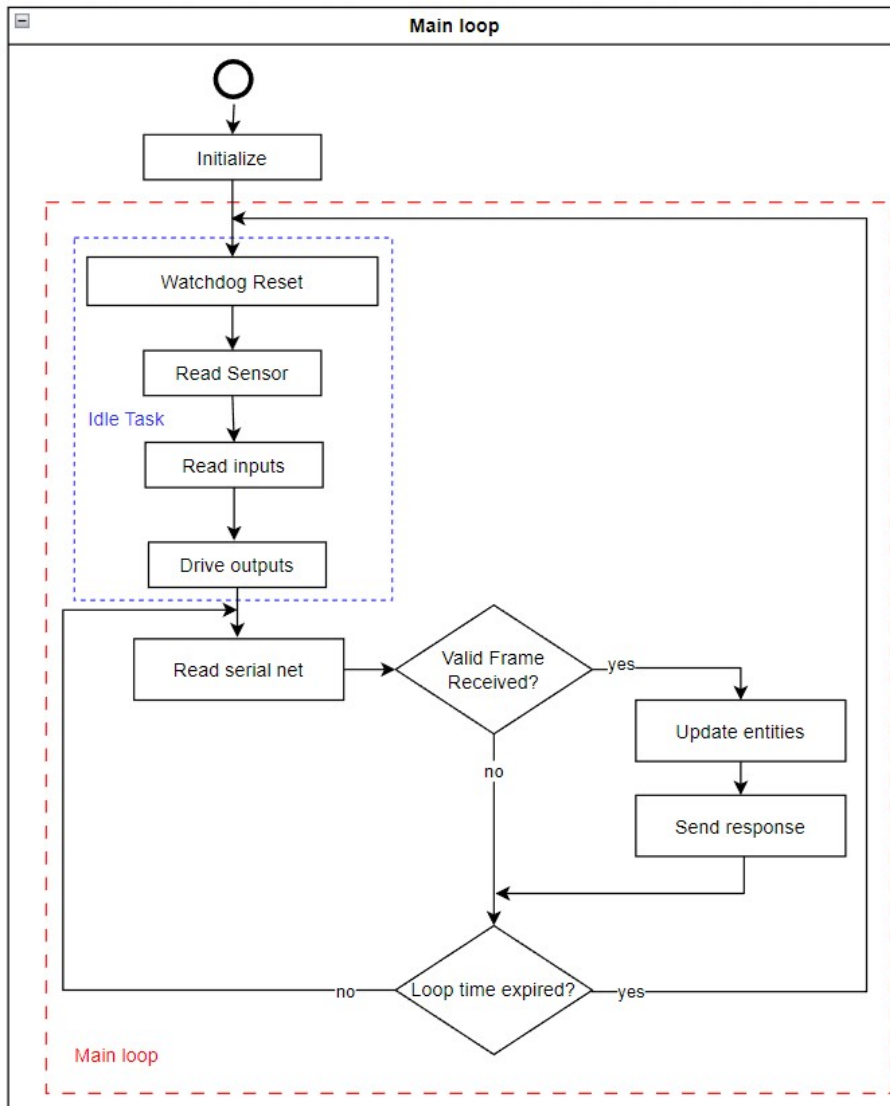
Πρέπει να τονιστεί ότι στο επίπεδο των modules, φαίνεται εντός των πλαισίων να υπάρχει και το σύμβολο < / > που σημαίνει interface. Πρακτικά όμως στη γλώσσα C, στην οποία γράφτηκε το firmware του αισθητήρα, δεν υπάρχουν interfaces, όπως αυτά υλοποιούνται στις γλώσσες υψηλού επιπέδου (πχ C++, Java). Αυτός είναι και ο λόγος που δεν εμφανίζονται δύο ξεχωριστά block (ένα του module και ένα του interface). Όμως ο τρόπος που δομήθηκε το πρόγραμμα παραπέμπει σε στυλ τύπου interface, αφού το υψηλότερο επίπεδο του software γνωρίζει μόνο τα ορίσματα των modules και τις τιμές που επιστρέφουν. Δεν έχει άλλη πληροφορία για τις λεπτομέρειες υλοποίησης. Αυτός είναι και ο λόγος που σε κάθε module block υπάρχει το σύμβολο του interface, αφού το module εκθέτει μόνο όση λειτουργικότητα χρειάζεται να ξέρουν τα παραπάνω επίπεδα και τίποτα παραπάνω.



Εικόνα 33 - Components αισθητήρα ανά επίπεδο προγράμματος βάσει Clean Architecture [25]

7.1.1 Λογισμικό υψηλού επιπέδου (επίπεδο εφαρμογής)

Πρόκειται ουσιαστικά για το επίπεδο της ίδιας της εφαρμογής. Αφορά τη βασική διαχείριση των εργασιών στο πλαίσιο του κύριου βρόγχου (main loop). Το διάγραμμα δραστηριοτήτων (activity diagram) στο επίπεδο αυτό, φαίνεται στην εικόνα 34. Το idle task εντός του μπλε πλαισίου τρέχει 1 φορά το δευτερόλεπτο, ενώ τον υπόλοιπο χρόνο τρέχει το πρωτόκολλο, διαβάζοντας συνεχώς τη γραμμή RS-485. Αυτός ο τρόπος δόμησης του προγράμματος, δίνει τη δυνατότητα στον ανιχνευτή να απαντήσει ταχύτατα στο PC, αφού μόνο για μια φορά εντός του δευτερολέπτου θα σταματήσει να τρέχει για λίγο, ώστε να τρέξουν οι εργασίες του idle task. Όπως φαίνεται από το διάγραμμα, σε περίπτωση που ληφθεί πακέτο εξυπηρετείται άμεσα. Οι εισερχόμενες πληροφορίες μένουν διαθέσιμες και θα χρησιμοποιηθούν από το idle task, με μέγιστο χρόνο εξυπηρέτησης 1 δευτερόλεπτο, όσο διαρκεί δηλαδή και ο κύκλος του main loop, το οποίο μόλις ξεκινήσει τον επόμενο κύκλο θα τρέξει πρώτα το idle task.



Εικόνα 34 - Διάγραμμα δραστηριοτήτων στο επίπεδο της εφαρμογής

Στην εικόνα 35 φαίνεται ο ατέρμονας βρόχος του συστήματος (main loop), ο οποίος υλοποιεί τη λογική που περιγράφηκε παραπάνω και βρίσκεται στο αρχείο *appif.c*.

```

18 //**
19 * @brief Initialize the high level components and runs the main loop.
20 */
21 void app_run(void) {
22     app_init();
23
24     while (1) {
25         app_loop_tick_start();
26         wdt_rst();
27         app_idle_handler(); // ~26ms
28
29         while (!app_loop_tick_expired()) {
30             // runs ~206K times per sec (~4.7us average per cycle)
31             net_handler(); // ~1ms worst case for all net logic to run
32         }
33     }
34 }

```

Εικόνα 35 - Firmware αισθητήρα: main loop (*appif.c*)

7.1.2 Επίπεδο υπομονάδων (modules)

Όπως φαίνεται στην εικόνα 33 τα modules του συστήματος είναι τέσσερα:

- Module αισθητήρα: Μετράει τον αισθητήρα και σώζει τις μετρήσεις σε κατάλληλες μεταβλητές, ώστε να μπορούν να χρησιμοποιηθούν από τα υπόλοιπα modules.
- Module δικτύου RS-485: Διαβάζει από τις κατάλληλες μεταβλητές τις ήδη αποθηκευμένες μετρήσεις από το module του αισθητήρα και τις στέλνει στον ΚΕΣ όταν του ζητηθούν. Από τον ΚΕΣ παραλαμβάνει την κατάσταση του συστήματος και τη σώζει στις κατάλληλες μεταβλητές, ώστε να μπορούν να χρησιμοποιηθούν από τα υπόλοιπα modules.
- Module ψηφιακών εισόδων / εξόδων: Διαβάζει από τις κατάλληλες μεταβλητές τις ήδη αποθηκευμένες τιμές στην κατάσταση του συστήματος και ανάλογα με αυτές οδηγεί την σειρήνα και τα ενδεικτικά (on ή off). Οι εισοδοί είναι τα jumpers, που χρησιμοποιούνται για την επιλογή διεύθυνσης του αισθητήρα.
- Module ενδεικτικών: Διαβάζει από τις κατάλληλες μεταβλητές τις ήδη αποθηκευμένες τιμές στην κατάσταση του συστήματος και ανάλογα με αυτές οδηγεί τα ενδεικτικά LED (on ή off).

Στην εικόνα 36 φαίνεται η «interface» function του serial net module. Όπως έχει ήδη αναφερθεί στην C δεν υπάρχουν interfaces, όμως η κλήση από τα ανώτερα στρώματα γίνεται χωρίς ορίσματα και επιστρεφόμενες τιμές. Αυτά προστίθενται εντός του αρχείου *modif.c*, διατηρώντας έτσι το υψηλό επίπεδο πλήρως διαχωρισμένο από τις λεπτομέρειες εφαρμογής των modules.

```
14 /**
15  * @brief Handles network communication.
16  */
17 void net_handler(void) {
18     if (l2_received() == true) {
19         // update application variables from rx payload
20         entities_update(&modif_objects.rx_payload.entities);
21         modif_objects.rx_command = modif_objects.rx_payload.command;
22         update_tx_payload_from_app_sensor();
23         l2_send();
24     }
25 }
26
```

Εικόνα 36 - "Interface" function του serial net module (*modif.c*)

7.1.3 Επίπεδο HAL χρήση

Προσφέρει ένα επίπεδο απομόνωσης μεταξύ της εφαρμογής και των hardware block του προγράμματος. Τα modules καλούν συναρτήσεις από αυτό το επίπεδο για να οδηγήσουν για παράδειγμα μια συγκεκριμένη λογική έξοδο (πχ σειρήνα) και όχι κάποιο συγκεκριμένο pin του μικροελεγκτή. Επίσης στο επίπεδο αυτό είναι άγνωστο τι είδους πρωτόκολλα χρησιμοποιούνται για την ανάγνωση των αισθητηρίων ή την επικοινωνία με τον ΚΕΣ. Οι

κλήσεις από τα ανώτερα στρώματα είναι στα επίπεδα εφαρμογής και μόνο και δεν σχετίζονται με τα hardware χαρακτηριστικά των περιφερειακών.

7.1.4 Επίπεδο HAL STM32

Αυτό το επίπεδο προγράμματος παρέχεται ως βιβλιοθήκη από την κατασκευάστρια εταιρία του μικροελεγκτή (ST Microelectronics) και παρέχει απομόνωση μεταξύ των επιλεγμένων περιφερειακών και του πιο χαμηλού επιπέδου των καταχωρητών του μικροελεγκτή. Σε αυτό το επίπεδο αποσαφηνίζονται όλες οι λεπτομέρειες λειτουργίας του κάθε module. Ότι για παράδειγμα η επικοινωνία στο RS-485 γίνεται με κάποια συγκεκριμένη UART, η οποία χρησιμοποιεί το DMA, ότι ο περιβαντολλογικός αισθητήρας χρησιμοποιεί ως επικοινωνία τη θύρα I²C κ.ο.κ. Η απομόνωση είναι σημαντική, αφού ο χρήστης δεν χρειάζεται να μπει στις λεπτομέρειες να οδηγήσει τους καταχωρητές του μικροελεγκτή, αλλά με πιο «υψηλού επιπέδου» κλήσεις να επιτύχει τον ίδιο στόχο. Συνεπώς σε αυτό το επίπεδο υπάρχουν εντολές τύπου «στείλε στην UART1 αυτό το frame», απαλλαγμένοι από τις λεπτομέρειες του χαμηλότερου επιπέδου του firmware. Σε αυτό σπουδαίο ρόλο παίζει και η χρήση του αρχείου .ioc στο project του αισθητήρα (παράγραφος 7.2), το οποίο δεν αποτελεί προφανώς το ίδιο το επίπεδο αυτό κάθε αυτό, αλλά δίνει ένα πολύτιμο γραφικό βοήθημα για τις ρυθμίσεις, παράγοντας αυτόματα τον κώδικα στα χαμηλά επίπεδα του hardware.

7.1.5 Επίπεδο firmware

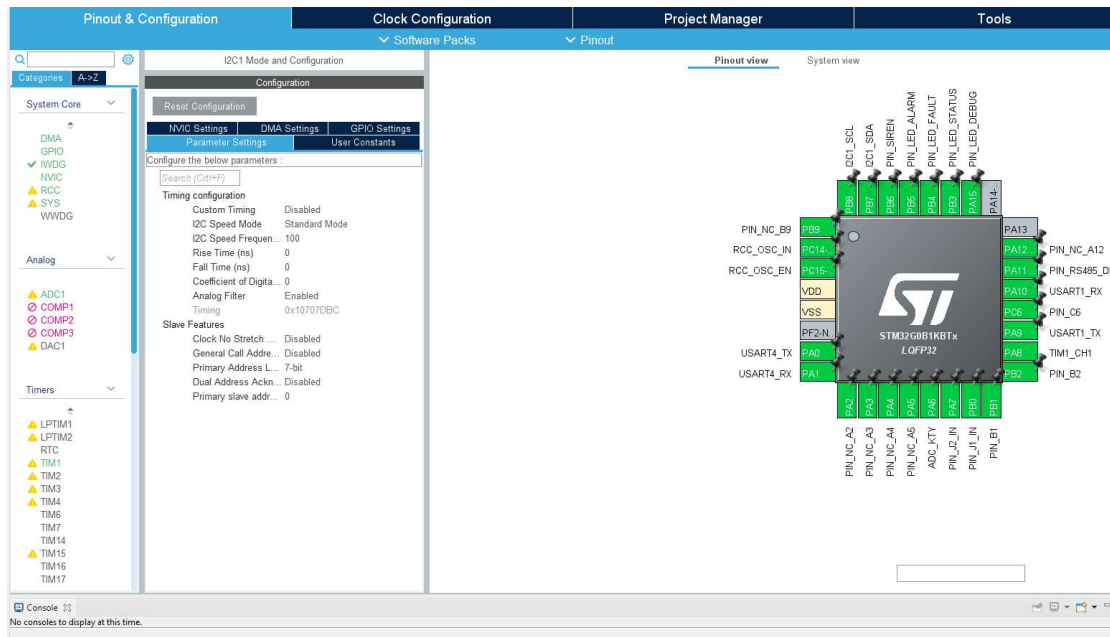
Και αυτό (όπως και το αμέσως ανώτερο) επίπεδο προγράμματος παρέχεται ως βιβλιοθήκη από την κατασκευάστρια εταιρία του μικροελεγκτή (ST Microelectronics). Πρόκειται για την οδήγηση των καταχωρητών του μικροελεγκτή, ανάλογα με τις εντολές που έρχονται από τα ανώτερα επίπεδα. Στο επίπεδο αυτό έχει παραληφθεί για παράδειγμα κάποιο frame από τα ανώτερα επίπεδα, το οποίο θα σταλεί στη γραμμή της UART1 μέσω κατάλληλης οδήγησης των καταχωρητών της MCU. Αυτήν την «κατάλληλη οδήγηση» αναλαμβάνει να εκτελέσει αυτό το επίπεδο. Δεν είναι άμεσα ορατό προς τον χρήστη, αφού βρίσκεται εσωτερικά του επιπέδου «HAL STM32», αλλά αυτός είναι και ο σκοπός του. Να απαλλάξει δηλαδή τον χρήστη από οποιαδήποτε επίγνωση γύρω από τους καταχωρητές της MCU, ώστε να μπορεί να αφοσιωθεί στο κτίσιμο της εφαρμογής του, έχοντας επαφή αποκλειστικά και μόνο με το επίπεδο «STM32 HAL».

7.2 Ρυθμίσεις MCU αισθητήρα – αρχείο .ioc

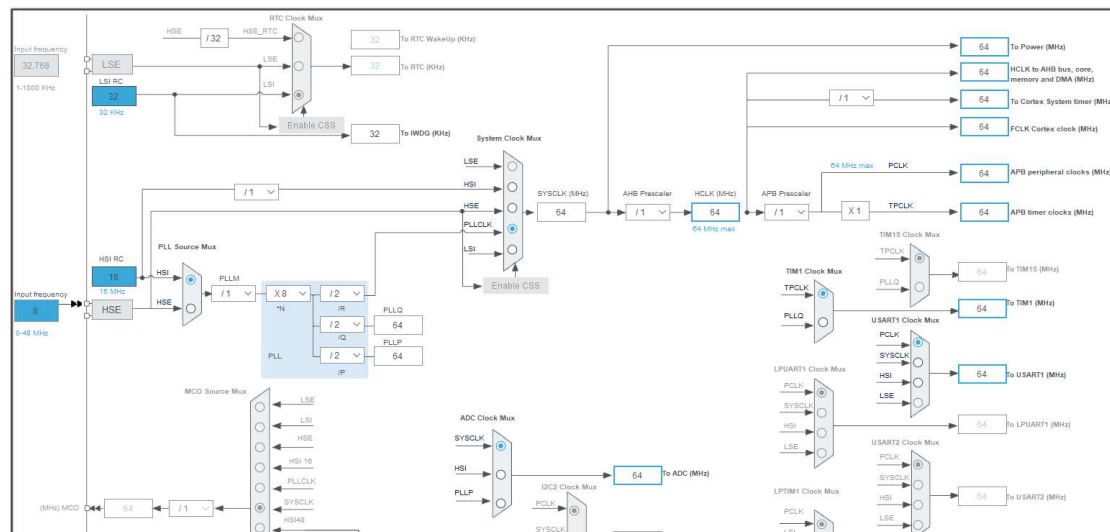
Το MXCubeIDE είναι το περιβάλλον ανάπτυξης των μικροελεγκτών της οικογένειας STM32, έναν από τους οποίους (STM32G01B1KBT3) χρησιμοποιεί ο αισθητήρας του συστήματος. Όπως αναφέρθηκε στην παράγραφο 7.1.4, οι ρυθμίσεις των περιφερειακών και ο κώδικας για τη λειτουργία του hardware παράγεται αυτόματα. Για να συμβεί αυτό θα πρέπει να πραγματοποιηθούν οι κατάλληλες ρυθμίσεις στο αρχείο GreenLinkPro.ioc, το οποίο είναι αρχείο κειμένου. Παρόλα αυτά προσφέρεται λογισμικό το οποίο αποδίδει σε γραφικό περιβάλλον το περιεχόμενο του αρχείου, ενώ αν κάποια ρύθμιση αλλάξει από τον χρήστη

μέσω του γραφικού περιβάλλοντος, το αρχείο ενημερώνεται κατάλληλα, ώστε να παραχθεί ο νέος κώδικας βάσει των νέων ρυθμίσεων.

Οι ρυθμίσεις γύρω από τα περιφερειακά και τα pins του μικροελεγκτή γίνονται στην καρτέλα «Pinout & Configuration» (εικόνα 37), ενώ για τους εσωτερικούς χρονισμούς του μικροελεγκτή οι ρυθμίσεις γίνονται στην καρτέλα «Clock Configuration» (εικόνα 38). Το αρχείο .ioc συμπεριλαμβάνεται στα παραδοτέα της εργασίας και βρίσκεται εντός του project του αισθητήρα.



Εικόνα 37 - Ρυθμίσεις pins και των περιφερειακών μικροελεγκτή



Εικόνα 38 - Ρυθμίσεις εσωτερικών clock του μικροελεγκτή

8 Μετρήσεις αισθητήρα

Όλες οι μετρήσεις που παρατίθενται σε αυτό το κεφάλαιο συλλέχτηκαν και παρουσιάστηκαν με τη χρήση δύο python scripts: *sensor_logger.py* και *sensor_analyzer.py*, τα οποία είναι μέρος της διπλωματικής εργασίας και επισυνάπτονται και αυτά στα παραδοτέα της διπλωματικής, μαζί με όλα τα log αρχεία.

Το πρώτο script συλλέγει τα δεδομένα που στέλνονται από τον αισθητήρα στη σειριακή θύρα του Η/Υ. Στη συνέχεια έχοντας πλέον το log file, γίνεται χρήση του δεύτερου script, ώστε να μετατραπούν οι «ωμές» (raw) τιμές του αισθητήρα σε ουσιαστική πληροφορία, πχ να μετατραπούν οι τιμές του ADC που αναγνώστηκαν από τον αναλογικό αισθητήρα, σε τιμές που αντιστοιχούν σε βαθμούς Κελσίου. Επίσης το script αυτό εξάγει και τις γραφικές παραστάσεις (απόκριση των αισθητήρων στον χρόνο).

Σημειώνεται ότι για τις ανάγκες της απεικόνισης, σε κάποια γραφήματα μπορεί να μη χρησιμοποιήθηκαν όλες οι τιμές του εκάστοτε log αρχείου, όμως όλη η πληροφορία βρίσκεται άθικτη στα log αρχεία των παραδοτέων. Ένα υπόδειγμα log αρχείου για τον αισθητήρα SEN55 φαίνεται στην εικόνα 39, όπου διακρίνεται στην πρώτη γραμμή η σηματοδότηση έναρξης των μετρήσεων και ποια είναι η διεύθυνση του αισθητήρα που μετρείται (3 στη συγκεκριμένη περίπτωση). Παρακάτω και από αριστερά προς τα δεξιά διακρίνονται ο αύξοντας αριθμός της καταχώρησης, ημερομηνία και ώρα, ο τύπος του αισθητήρα («E» για τον περιβαντολλογικό) και στη συνέχεια όλες οι raw μετρήσεις όπως παίρνονται από τον αισθητήρα.

```
1 START, 04/01/2024 20:57:31.001, address=3
2 1223/1223, 04/01/2024 20:58:56.479, E, pm1p0=173, pm2p5=186, pm4p0=190, pm10p0=192, temp=4437, hum=6097, voc=940, nox=10
3 1224/1224, 04/01/2024 20:58:56.541, E, pm1p0=173, pm2p5=186, pm4p0=190, pm10p0=192, temp=4437, hum=6097, voc=940, nox=10
4 1225/1225, 04/01/2024 20:58:56.619, E, pm1p0=173, pm2p5=186, pm4p0=190, pm10p0=192, temp=4437, hum=6097, voc=940, nox=10
5 1226/1226, 04/01/2024 20:58:56.682, E, pm1p0=173, pm2p5=186, pm4p0=190, pm10p0=192, temp=4437, hum=6097, voc=940, nox=10
6 1227/1227, 04/01/2024 20:58:56.758, E, pm1p0=173, pm2p5=186, pm4p0=190, pm10p0=192, temp=4437, hum=6097, voc=940, nox=10
7 1228/1228, 04/01/2024 20:58:56.823, E, pm1p0=173, pm2p5=186, pm4p0=190, pm10p0=192, temp=4437, hum=6097, voc=940, nox=10
8 1229/1229, 04/01/2024 20:58:56.900, E, pm1p0=173, pm2p5=186, pm4p0=190, pm10p0=192, temp=4437, hum=6097, voc=940, nox=10
9 1230/1230, 04/01/2024 20:58:56.964, E, pm1p0=173, pm2p5=186, pm4p0=190, pm10p0=192, temp=4437, hum=6097, voc=940, nox=10
10 1231/1231, 04/01/2024 20:58:57.027, E, pm1p0=173, pm2p5=186, pm4p0=190, pm10p0=192, temp=4437, hum=6097, voc=940, nox=10
11 1232/1232, 04/01/2024 20:58:57.102, E, pm1p0=173, pm2p5=187, pm4p0=191, pm10p0=193, temp=4437, hum=6087, voc=940, nox=10
12 1233/1233, 04/01/2024 20:58:57.180, E, pm1p0=173, pm2p5=187, pm4p0=191, pm10p0=193, temp=4437, hum=6087, voc=940, nox=10
13 1234/1234, 04/01/2024 20:58:57.242, E, pm1p0=173, pm2p5=187, pm4p0=191, pm10p0=193, temp=4437, hum=6087, voc=940, nox=10
```

Εικόνα 39 - Υπόδειγμα αρχείου log

Για κάθε log αρχείο με τιμές, υπάρχει και ένα log αρχείο με τα συμβάντα, ώστε να μπορούν αυτά να αποδίδονται στη γραφική παράσταση ευκρινώς με τη χρήση διακεκομμένων κόκκινων γραμμών. Τα αρχεία αυτά συμπληρώνονται χειροκίνητα. Τελειώνουν πάντα σε “_notes.log”. Ένα υπόδειγμα ενός τέτοιου αρχείου συμβάντων φαίνεται στην εικόνα 40. Στο συγκεκριμένο αρχείο κρατήθηκε η ημερομηνία και ώρα δύο συγκεκριμένων συμβάντων και συγκεκριμένα η έναρξη και ο τερματισμός παροχής υγραποιημένου αερίου στον περιβαντολλογικό αισθητήρα.

```
1 07/01/2024 17:26:00.113 gas started
2 07/01/2024 17:26:15.157 gas ended
```

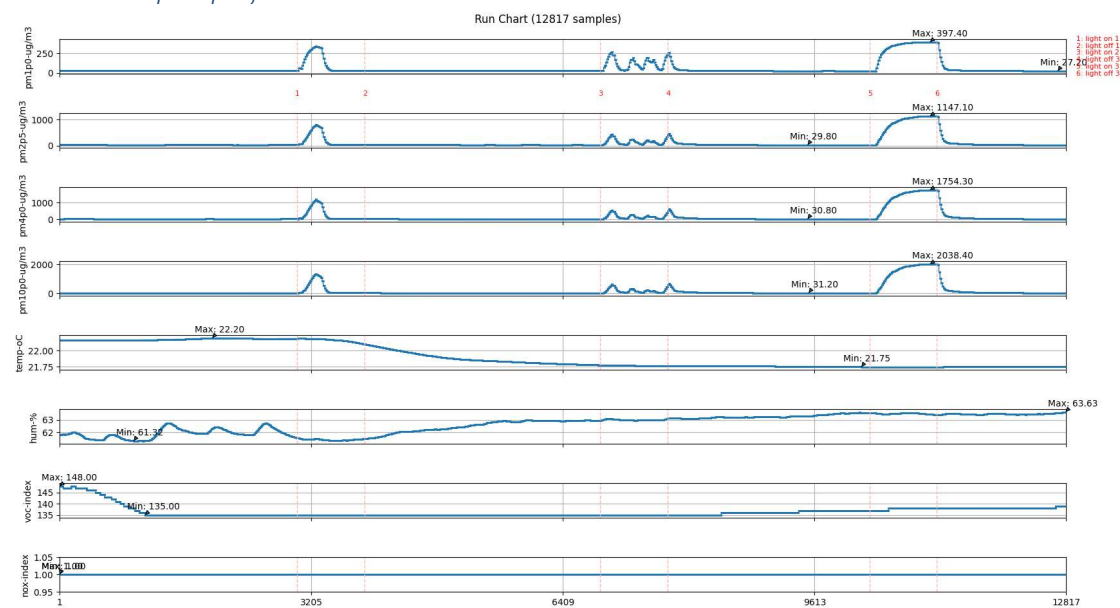
Εικόνα 40 - Υπόδειγμα αρχείου “_notes.log”

Σημαντική σημείωση: Δεν υπάρχει «εμπεριστατωμένος» τρόπος για την πραγματοποίηση των συγκεκριμένων μετρήσεων, παρά μόνο με εξειδικευμένα και ακριβά εργαλεία. Κάποιες επιχειρήσεις διαθέτουν κάποια από αυτά (πχ περιβατολογικούς θαλάμους), αλλά για τα αέρια οι μετρήσεις αυτές γίνονται μόνο σε ειδικά και εγκεκριμένα από την ΕΕ εργαστήρια, με μηχανήματα αξίας εκατοντάδων χιλιάδων ή και εκατομμυρίων ευρώ. Στα πλαίσια μιας διπλωματικής εργασίας όπως η παρούσα, μπορούν να παρθούν μετρήσεις και τα αποτελέσματά τους να βοηθήσουν στην κατανόηση της λειτουργίας του αισθητήρα. Όμως χωρίς τη χρήση των προαναφερόμενων διακριβωμένων εργαλείων αναφοράς, μπορεί να παρατηρηθεί απλά η συμπεριφορά του αισθητήρα και μόνο, χωρίς να μπορεί να αποδειχτεί η ακρίβεια των απόλυτων τιμών που παρουσιάζονται. Σίγουρα πάντως είναι κοντά στην πραγματικότητα και από πλευράς engineering «βγάζουν νόημα».

8.1 Περιβατολογικός αισθητήρας

Ο περιβατολογικός αισθητήρας SEN55 της SENSIRION προσφέρει 8 διαφορετικές μετρήσεις, οι οποίες επηρεάζονται από τη συγκέντρωση συγκεκριμένων αερίων, αλλά και από τη θερμοκρασία και την υγρασία. Έγιναν συγκεκριμένες μετρήσεις με συγκεκριμένους τύπου συμβάντων, τα οποία παρουσιάζονται παρακάτω.

8.1.1 Έκθεση σε φως



Εικόνα 41 - Απόκριση περιβατολογικού αισθητήρα στην έκθεσή του σε φως από φακό

Όπως έχει ήδη εξηγηθεί η βασική λειτουργία του αισθητήρα SEN55, δεν παρέχεται αισθητήριο φωτός. Όμως η έκθεση σε φως είναι πάντα μια δοκιμή που πρέπει να γίνεται, διότι κάποιοι αισθητήρες επηρεάζονται από την επίδρασή του.

Κρατώντας φακό σε πολύ κοντινή απόσταση από τον αισθητήρα (περίπου 3cm) πάρθηκαν μετρήσεις οι οποίες φαίνονται στην εικόνα 41. Τρεις φορές έπεσε φως στον αισθητήρα για

διάστημα ενός λεπτού. Η συνολική διάρκεια του γραφήματος είναι 14.5 λεπτά. Τα στοιχεία για όλες τις καταγεγραμμένες τιμές, καθώς και για τις χρονικές στιγμές που το φως έπεφτε επάνω στον αισθητήρα, υπάρχουν στα αρχεία *20240104_sn03_env_light.log* και *20240104_sn03_env_light_notes.log*. Τα τρία διαστήματα κατά τα οποία έπεφτε φως στον αισθητήρα, φαίνονται ανάμεσα στις διακεκομμένες γραμμές 1 με 2, 3 με 4 και 5 με 6 αντίστοιχα.

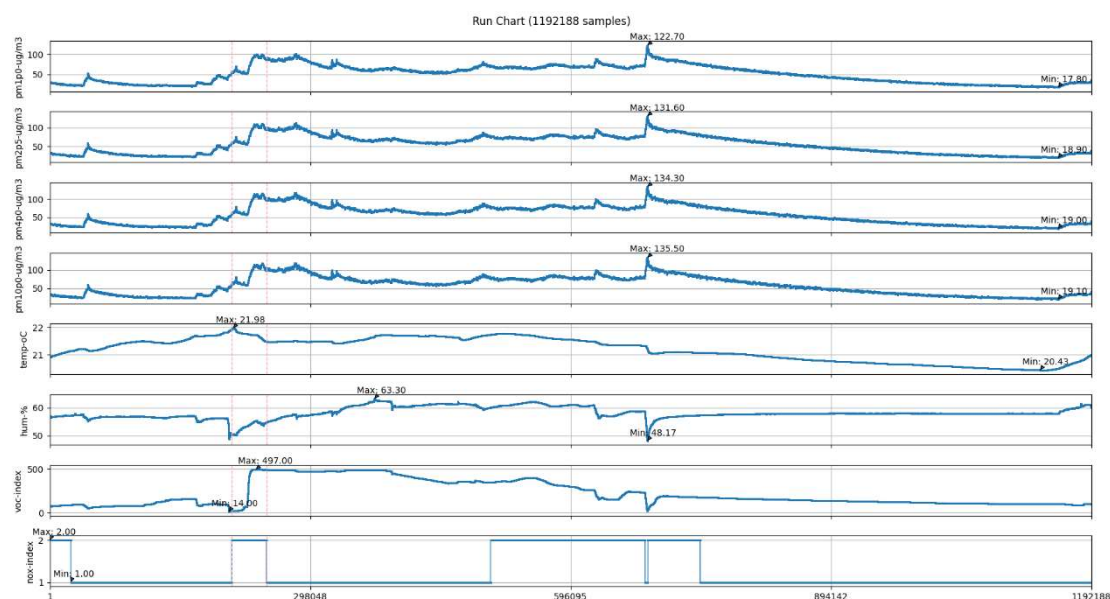
Τα συμπεράσματα που εξάγονται από αυτήν την εικόνα είναι τα εξής:

1. Και τις τρεις φορές που ο αισθητήρας εκτίθεται σε φως, προκαλείται ένα τοπικό μέγιστο στη συγκέντρωση των μικροσωματιδίων. Το μέγιστο αυτό είναι για τα μεγαλύτερα σωματίδια περίπου δύο τάξεις μεγέθους μεγαλύτερο από την τιμή ηρεμίας (από τιμές περίπου 30 ug/m³ σε τιμές πάνω από 2000 ug/m³).
2. Κανένα άλλο μέγεθος δεν φάνηκε να έχει αλλαγή στην τιμή του, πέρα από τα μικροσωματίδια.

➔ **Συμπέρασμα:** Η έκθεση του αισθητήρα SEN55 στο φως προκαλεί αυξημένη τιμή στη συγκέντρωση των μικροσωματιδίων. Θα πρέπει να δίνεται σαφής οδηγία στους εγκαταστάτες να μην τοποθετούν τον αισθητήρα πολύ κοντά σε φωτεινή πηγή και σίγουρα όχι σε μέρος στο οποίο κατά τη διάρκεια της ημέρας φωτίζεται (έστω και λίγο) από τον ήλιο.

8.1.2 Μακροπρόθεσμη μέτρηση ηρεμίας σε εσωτερικό χώρο

Η απόκριση του αισθητήρα σε αυτή τη μέτρηση φαίνεται στην εικόνα 42.



Εικόνα 42 - Απόκριση ηρεμίας περιβαλλοντολογικού αισθητήρα σε διάρκεια είκοσι τριών ωρών

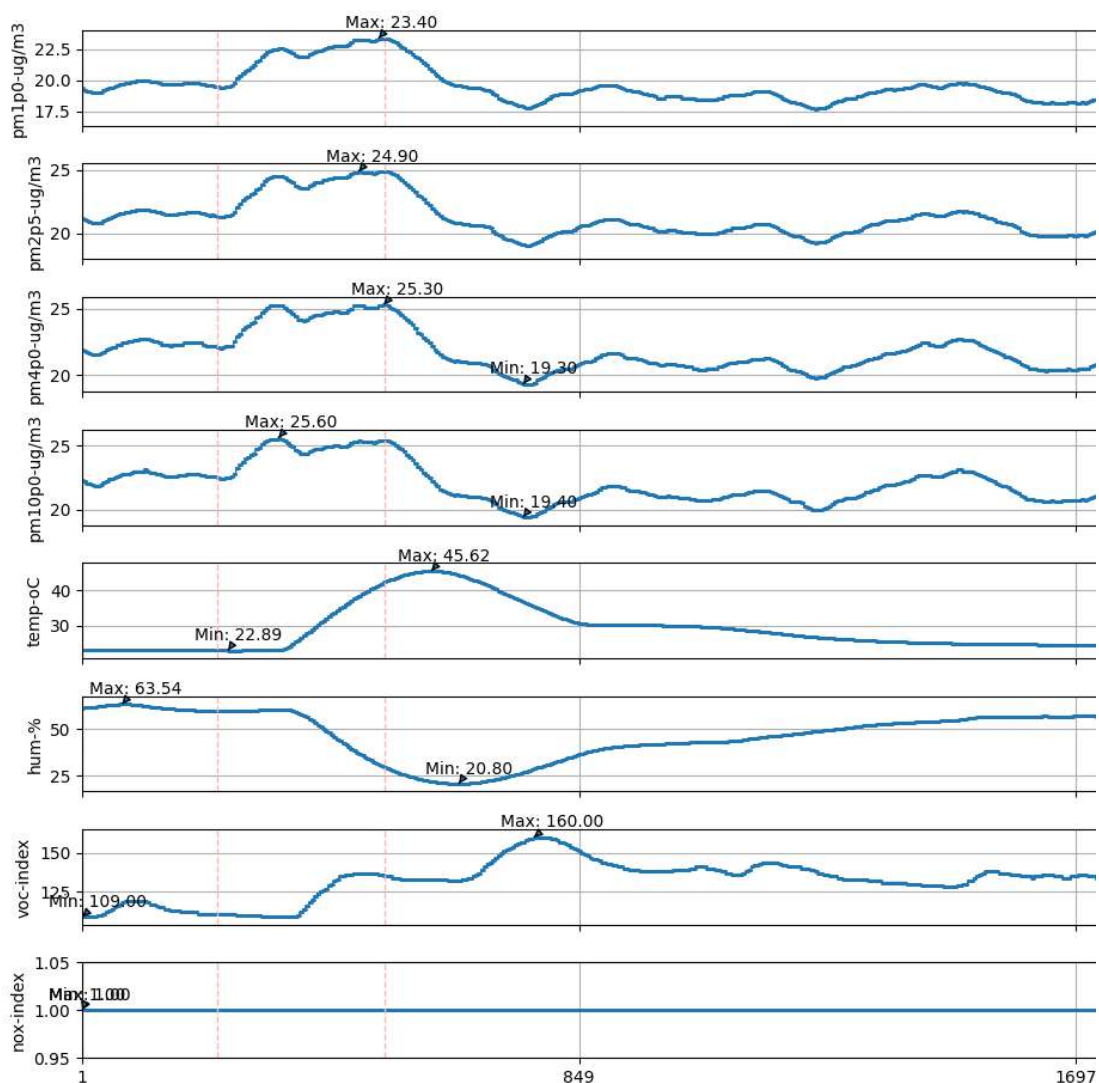
Κατά τη διάρκεια της μέτρησης δεν υπάρχει κάποιο συμβάν που να προκαλεί αλλαγή στην συμπεριφορά του αισθητήρα, εκτός από την έκθεση του στον ήλιο (διάστημα ανάμεσα στις

διακεκομμένες κόκκινες γραμμές). Ο χρόνος που αντιστοιχεί στο γράφημα είναι 23 ώρες. Οι πληροφορίες αυτές υπάρχουν εντός του αρχείου *20240107_sn03_env23h_final.log* και οι χρονικές στιγμές των κόκκινων γραμμών βρίσκονται στο αρχείο *20240107_sn03_env23h_final_notes.log*.

➔ **Συμπέρασμα:** Δεν φαίνεται να υπάρχει κάποια περίεργη συμπεριφορά στην μέτρηση του αισθητήρα. Κάποιες μεταβολές της θερμοκρασίας και της υγρασίας εντός της ημέρας είναι αναμενόμενες. Η μικρή αύξηση των σωματιδίων και του VOC index έχει επεξηγηθεί στην παράγραφο 8.1.1 με τις μετρήσεις του αισθητήρα κατά την έκθεσή του στο φως, κάτι που όπως αποδεικνύεται θα πρέπει να αποφεύγεται.

8.1.3 Μεταβολές σε θερμοκρασία – υγρασία

Η απόκριση του αισθητήρα σε αυτή τη μέτρηση φαίνεται στην εικόνα 43.



Εικόνα 43 - Απόκριση περιβαλλοντολογικού αισθητήρα στη χρήση στεγνωτήρα μαλλιών

Ένας τρόπος να παρατηρήσουμε ταυτόχρονα τη μεταβολή της θερμοκρασίας και της υγρασίας, είναι με τη χρήση στεγνωτήρα για τα μαλλιά (σεσουάρ). Η θερμοκρασία

αναμένεται προφανώς να αυξηθεί κατά πολλούς βαθμούς, αλλά ο ζεστός αέρας θα ξηράνει τον αισθητήρα και συνεπώς αναμένεται και σημαντική μείωση της υγρασίας.

Κρατώντας τον στεγνωτήρα σε απόσταση 30cm, ζεστός αέρας έρρεε προς τον αισθητήρα για διάστημα ενός λεπτού. Ο χρόνος κατά τον οποίο ο στεγνωτήρας ήταν ανοιχτός, φαίνεται ανάμεσα στις δύο διακεκομμένες κόκκινες γραμμές. Ο συνολικός χρόνος στο γράφημα είναι περίπου 6 λεπτά. Οι μετρήσεις βρίσκονται εντός του αρχείου *20240104_sn03_env_hair_dryer.log* και οι χρονικές στιγμές των κόκκινων γραμμών στο αρχείο *20240104_sn03_env_hair_dryer_notes.log*.

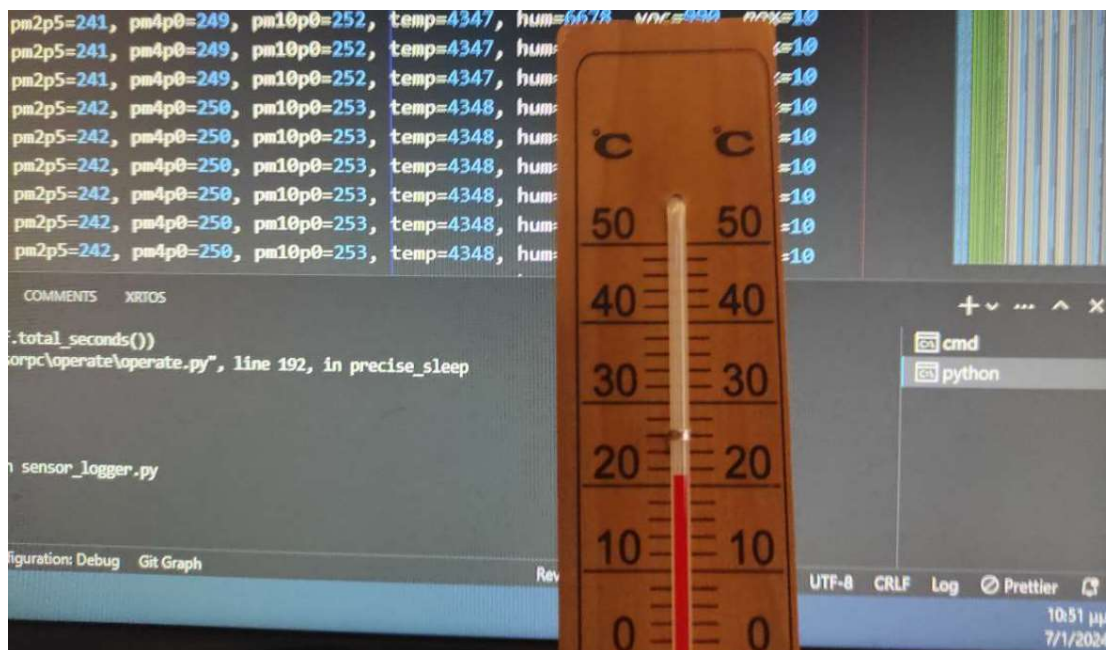
Τα συμπεράσματα που εξάγονται από αυτήν την εικόνα είναι τα εξής:

1. Όπως είναι απολύτως αναμενόμενο η τιμή της θερμοκρασίας αυξάνεται δραματικά με τη ζεστή ροή αέρα, από τους 22.9°C στους 45.6°C.
2. Η τιμή της υγρασίας μειώνεται δραματικά, από το 63.5% στο 20.8%. Αυτό διότι η υψηλή θερμοκρασία του στεγνωτήρα, ξηραίνει τον αέρα που ρέει προς τον αισθητήρα.
3. Οι μετρήσεις θερμοκρασίας και υγρασίας παρουσιάζουν μια καθυστέρηση (περίπου 30 δευτερόλεπτα) στο να ανταποκριθούν τόσο στην έναρξη ροής του ζεστού αέρα, όσο και στον τερματισμό της ροής αυτής. Αυτό είναι λογικό βάσει της δομής του αισθητήρα ο οποίος, για τις μετρήσεις αυτές, έχει εσωτερικό αλγόριθμο με κυλιόμενο ΜΟ.
4. Η θερμοκρασία και η υγρασία χρειάζονται περίπου 4 λεπτά να επανέλθουν από τη διακοπή ροής του ζεστού αέρα στις μετρήσεις ηρεμίας (πριν την έναρξη ροής του ζεστού αέρα).
5. Υπάρχει μια μικρή αύξηση στις τιμές όλων των ειδών μικροσωματιδίων, η οποία είναι ανεπαίσθητη για να είναι αξιολογήσιμη, αφού η τάξη μεγέθους παραμένει η ίδια και η συγκέντρωση είναι πολύ κατώτερη από τα επίπεδα συναγερμού του συστήματος. Η αύξηση αυτή θα μπορούσε να οφείλεται σε συσσώρευση μικροσκοπικών κόκκων σκόνης εντός του στεγνωτήρα.
6. Υπάρχει μια μικρή αύξηση στις τιμές του VOC index, η οποία είναι ανεπαίσθητη για να είναι αξιολογήσιμη, αφού η τάξη μεγέθους παραμένει η ίδια και η συγκέντρωση είναι πολύ κατώτερη από τα επίπεδα συναγερμού του συστήματος. Θα μπορούσε να οφείλεται σε μόρια πλαστικού του στεγνωτήρα που υπερθερμαίνονται και αεριοποιούνται σε πολύ μικρές ποσότητες.
7. Δεν υπάρχει επίδραση της χρήσης του στεγνωτήρα στις τιμές NOx.

➔ **Τελικό συμπέρασμα:** Υπάρχει ραγδαία αύξηση της θερμοκρασίας και μείωση της υγρασίας κατά τη χρήση του στεγνωτήρα. Οι τιμές επανέρχονται περίπου 4 λεπτά από τη διακοπή χρήσης του. Αυξήσεις στις τιμές των μικροσωματιδίων και του VOC index είναι μικρές και μη αξιολογήσιμες.

8.1.4 Σύγκριση ένδειξης θερμοκρασίας με πραγματικό θερμόμετρο

Θα συγκριθεί η ένδειξη του SEN55 σε σχέση με ένα απλό αναλογικό θερμόμετρο. Στις 07/01/2024 - 22:51 το θερμόμετρο δείχνει τιμή λίγο κάτω από τους 22°C (εικόνα 44). Κατά τη στιγμή εκείνη τρέχει και το `sensor_logger.py` (στο κάτω αριστερά μέρος της εικόνας) και καταχωρεί τιμές. Η τιμή όπως στέλνεται από τον αισθητήρα εκείνη την ώρα είναι (σύμφωνα με την οθόνη του Η/Υ) 4348. Αυτή η τιμή εάν διαιρεθεί με το 20 θα δώσει την τιμή σε °C (παρ. 10.2.4), και συνεπώς η τιμή αυτή είναι περίπου 21.74°C. Άρα οι δύο τιμές είναι πολύ κοντά μεταξύ τους, ώστε να θεωρήσουμε ότι ο SEN55 παράγει τιμή που αντιστοιχεί στην πραγματικότητα.



Εικόνα 44 - Ένδειξη πραγματικού θερμομέτρου στις 07/01/2024 – 22:51

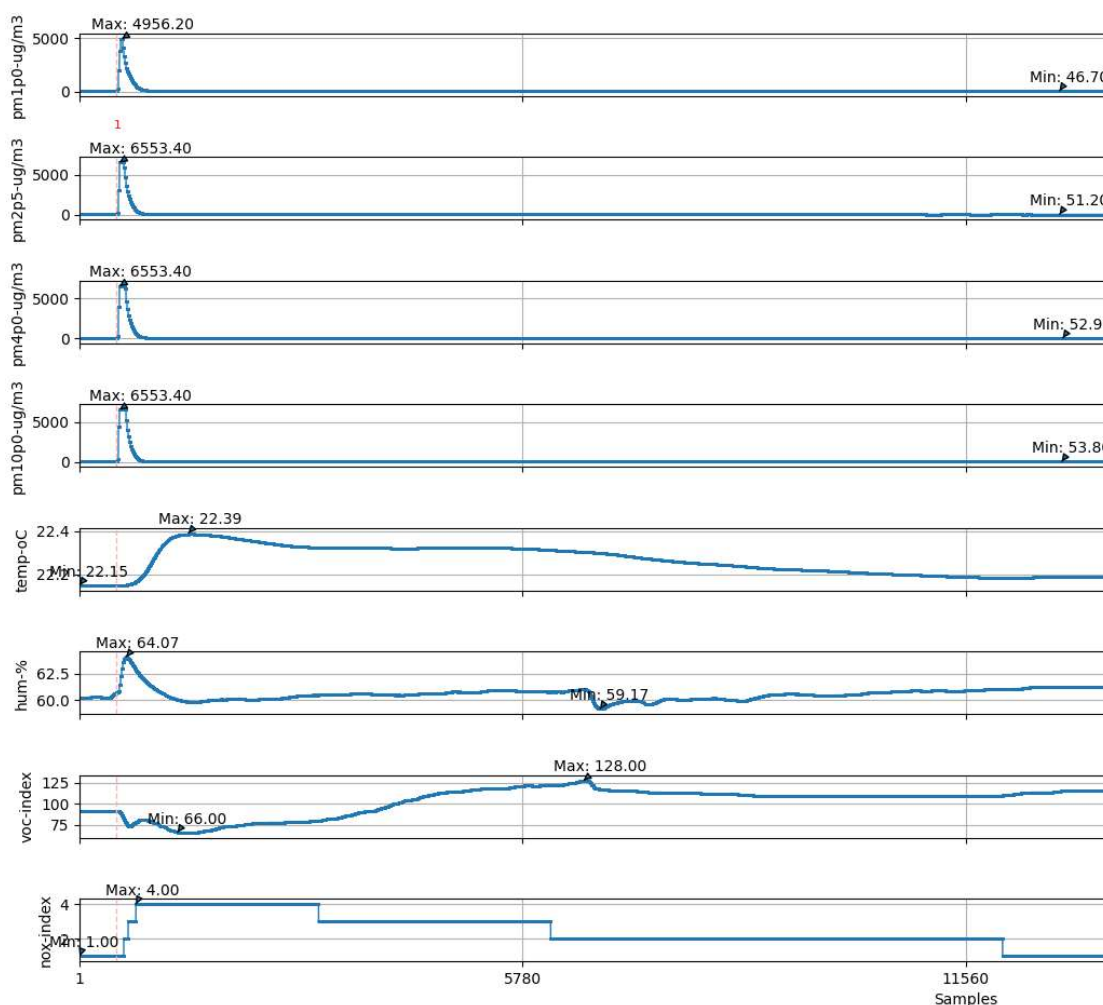
Το αρχείο καταγραφής και το αρχείο συμβάντων για τη μέτρηση αυτή είναι τα `20240107_sn03_env_thermometer.log` και `20240107_sn03_env_thermometer_notes.log`.

8.1.5 Μεταβολές σε μικροσωματίδια και NOx index

Ένας τρόπος έκθεσης του αισθητήρα σε μικροσωματίδια και πτητικές οργανικές ενώσεις, είναι να εκτεθεί σε καπνό που παράγεται κατά το άναμμα σπύρτου. Στο σπύρτο αντιδρούν αρκετά στοιχεία μεταξύ τους όπως τριθειούχο αντιμόνιο (Sb_2S_3), χλωρικό κάλιο ($KClO_3$) και κόκκινος φώσφορος. Κατά του άναμμα του σπύρτου και λόγω των χημικών αντιδράσεων παράγονται άκαυστα μικροσωματίδια από ατελείς καύσεις και συνεπώς αναμένεται σημαντική αύξηση στα μικροσωματίδια. Από την άλλη οι καύσεις κατά το άναμμα του σπύρτου αντιδρούν και με τα στοιχεία της ατμόσφαιρας, όπως το άζωτο. Συνεπώς αναμένουμε αύξηση και στο NOx index, αλλά θα πρέπει να διαπιστωθεί εάν η τόσο σύντομη έκθεση του αισθητήρα στον καπνό του σπύρτου, είναι ικανή να οδηγήσει τον αισθητήρα σε άνοδο της τιμής του δείκτη αυτού. Τέλος κάποια πτητικά αέρια όπως για παράδειγμα λευκός φώσφορος παράγονται κατά την καύση της πυρίτιδας στην κεφαλή του σπύρτου, αλλά και πάλι θα πρέπει να διαπιστωθεί αν αυτά είναι αρκετά σε ποσότητα, ώστε να οδηγήσουν σε

αύξηση το VOC index. Τέλος στα προϊόντα της καύσης περιέχονται και υδρατμοί από την καύση του ξύλου, συνεπώς δεν θα προκαλούσε εντύπωση και παροδική αύξηση της υγρασίας. [26]

Στο συγκεκριμένο πείραμα το σπίρτο άναψε ακριβώς δίπλα στον αισθητήρα, περίπου σε 1cm απόσταση. Στην εικόνα 45 και συγκεκριμένα στην κάθετη διακεκομμένη κόκκινη γραμμή, είναι η στιγμή που ο αισθητήρας εκτέθηκε στον καπνό του σπύρτου (περίπου 1 δευτερόλεπτο). Το συνολικό γράφημα αντιστοιχεί σε διάρκεια 15 λεπτών. Τα αρχεία όπου βρίσκονται οι καταγεγραμμένες μετρήσεις και η χρονική στιγμή χρήσης του σπύρτου, βρίσκονται στα αρχεία `20240104_sn03_env_match.log` και `20240104_sn03_env_match_notes.log`.



Εικόνα 45 - Απόκριση περιβαλλοντικού αισθητήρα στην έκθεση σε καπνό ανάμματος σπύρτου

Τα συμπεράσματα από την απόκριση του αισθητήρα είναι τα εξής:

1. Παρατηρείται ραγδαία αύξηση των μικροσωματιδίων κατά εκατοντάδες φορές στα μικρότερα σωματίδια, ενώ στα μεγάλα σωματίδια ο αισθητήρας έρχεται σε κορεσμό και μετράει τη μέγιστη τιμή (6553.4 $\mu\text{g}/\text{m}^3$).

2. Οι συγκεντρώσεις των μικροσωματιδίων επιστρέφουν σχετικά γρήγορα (περίπου 45 δευτερόλεπτα) στα επίπεδα ηρεμίας.
3. Το VOC index φαίνεται παροδικά να πέφτει ελάχιστα και στη συνέχεια να αυξάνεται ελαφρώς, αλλά η αύξηση είναι μικρή (από 90 σε 125) και συνεπώς δεν κρίνεται αξιολογήσιμη.
4. Το NOx index τετραπλασιάζει την τιμή του από το 1 στο 4 και συνεπώς φαίνεται να υπάρχει κάποια αύξηση, χωρίς όμως να αλλάζει η τάξη μεγέθους της μέτρησης.
5. Το NOx index χρειάζεται περίπου 13.5 λεπτά από το άναμμα του σπύρτου για να επιστρέψει στην τιμή ηρεμίας του (1).
6. Υπάρχει παροδική αύξηση της θερμοκρασίας κατά 0.2°C κατά το άναμμα του σπύρτου, αλλά η έκθεση στη φλόγα για τόσο σύντομο χρονικό διάστημα (ένα με δύο δευτερόλεπτα), δεν είναι αρκετή να προκαλέσει σημαντική αύξηση στη θερμοκρασία, ειδικά με τον κυλιόμενο MO που χρησιμοποιεί εσωτερικά ο αισθητήρας.
7. Η αύξηση στην υγρασία κατά περίπου 4%, οφείλεται στους υδρατμούς που απελευθερώνονται από την καύση του ξύλου του σπύρτου. Το φαινόμενο αποσβένεται μετά από περίπου 2 λεπτά και 15 δευτερόλεπτα, όταν και η τιμή της υγρασίας επιστρέφει στα επίπεδα ηρεμίας.

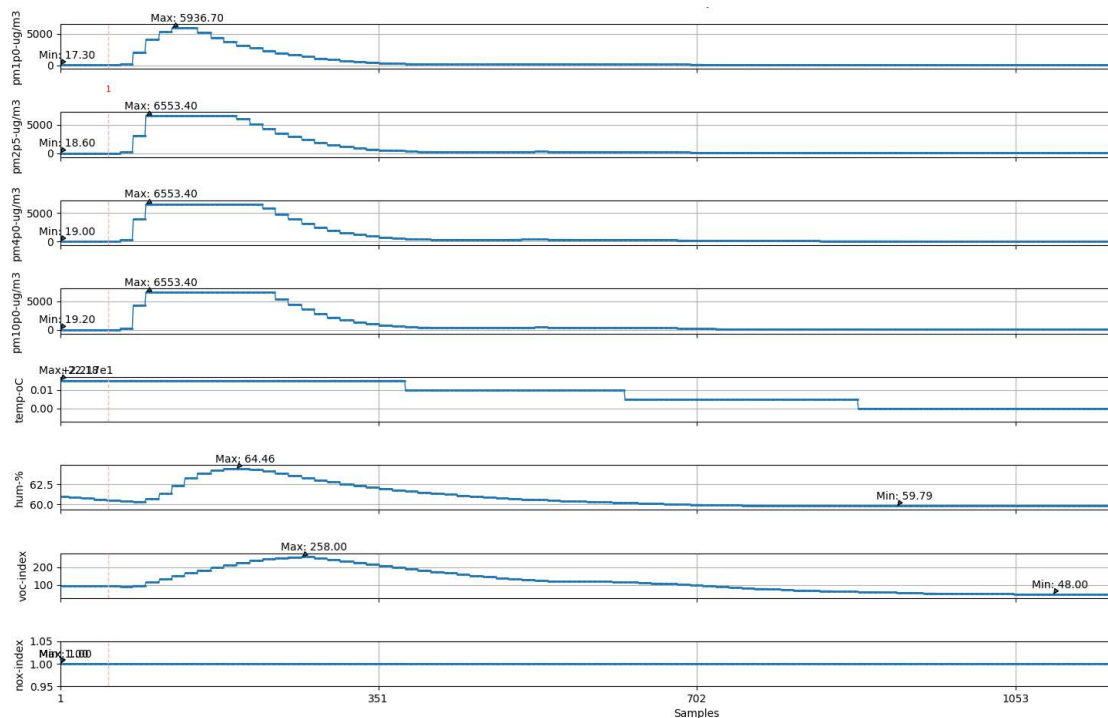
➔ **Τελικό συμπέρασμα:** Τα μεγέθη που μεταβάλλονται κατά το άναμμα του σπύρτου είναι κατά κύριο λόγο τα μικροσωματίδια. Μικρές αυξήσεις καταγράφονται και στα επίπεδα του NOx index (προφανώς από τις αντιδράσεις με το άζωτο της ατμόσφαιρας), αλλά και της υγρασίας, λόγω της απελευθέρωσης υδρατμών από το ξύλο που καίγεται.

8.1.6 Μεταβολές σε μικροσωματίδια και VOC index

Μπορούμε να παρατηρήσουμε τις μεταβολές των μικροσωματιδίων και του VOC index, εάν εκθέσουμε τον αισθητήρα σε αιθάλη. Ένας άμεσος τρόπος είναι με τον καπνό. Ο καπνός του τσιγάρου συγκεκριμένα, είναι πλούσιος σε μικροσωματίδια όπως πίσσα, νικοτίνη και μονοξειδία του άνθρακα (CO) [27], αλλά και πτητικές οργανικές ενώσεις όπως ακρυλονιτρίλιο (C₃H₃N), φουράνιο (C₄H₄O) και χλωριούχο βινύλιο, που έχει χημικό τύπο (C₂H₃Cl)_n [28], καθώς και οξειδία του αζώτου και κυρίως διοξείδιο του αζώτου (NO₂) [29].

Συνεπώς εκθέτοντας τον αισθητήρα σε καπνό τσιγάρου, αναμένουμε σημαντική αύξηση τόσο στα μικροσωματίδια, όσο και στο VOC index, ενώ για το NOX index θα πρέπει να εξακριβωθεί κατά πόσο επηρεάζεται από την παρουσία μόνο του διοξειδίου του αζώτου.

Καπνός τσιγάρου εκπνεύστηκε για μία φορά επάνω στον αισθητήρα από πολύ κοντινή απόσταση (περίπου 3 cm) και για περίπου 2 δευτερόλεπτα. Στην εικόνα 46 η στιγμή που αισθητήρας εκτέθηκε στον καπνό, είναι εκείνη στην οποία είναι σχεδιασμένη η κόκκινη διακεκομμένη γραμμή. Ο συνολικός χρόνος στο παρακάτω γράφημα είναι περίπου 3 λεπτά. Η κόκκινη γραμμή βρίσκεται στην αρχή του γραφήματος. Οι πληροφορίες αυτές μπορούν να φανούν εντός του αρχείου *20240104_sn03_env_hair_dryer.log* και οι χρονικές στιγμές των κόκκινων γραμμών στο αρχείο *20240104_sn03_env_hair_dryer_notes.log*.



Εικόνα 46 - Απόκριση περιβαλλοντικού αισθητήρα στην έκθεση σε καπνό από τσιγάρο

Τα συμπεράσματα που εξάγονται από αυτήν την εικόνα είναι τα εξής:

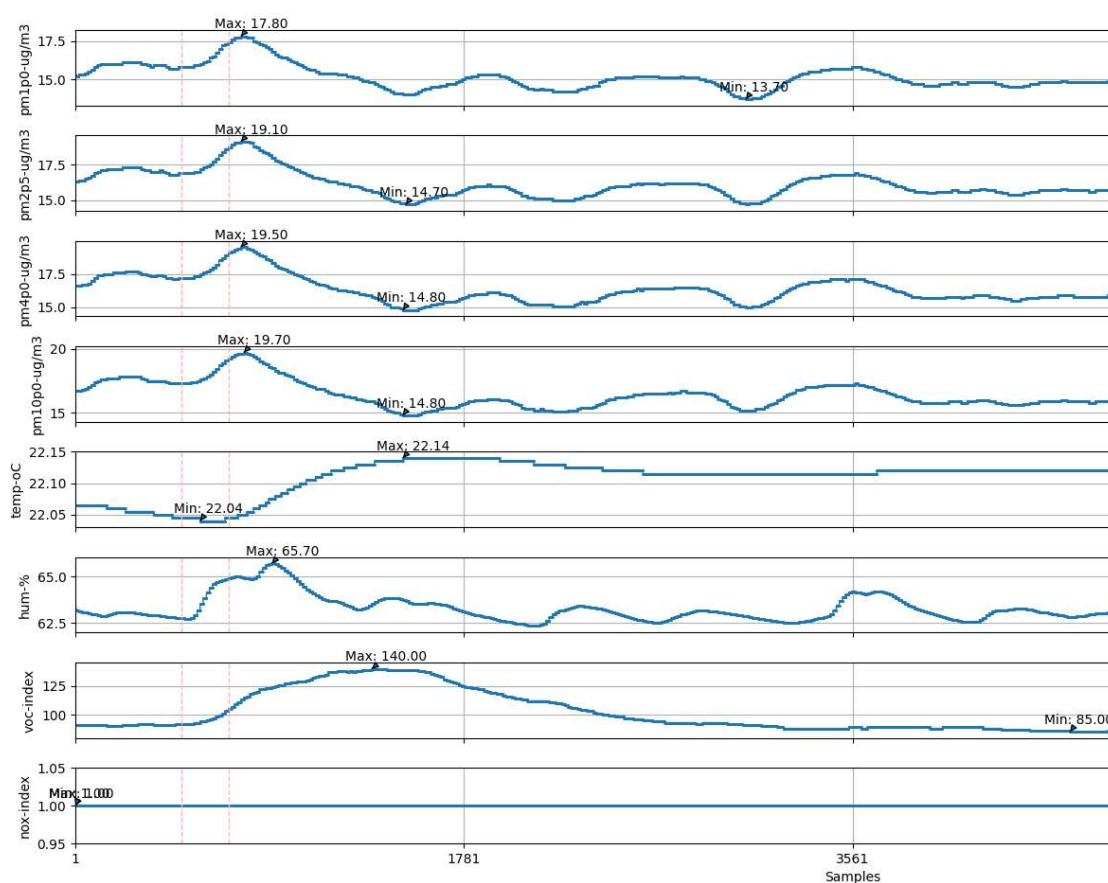
1. Υπάρχει «εκτόξευση» στα μικροσωματίδια τόσο μεγάλη, ώστε μόνο στα μικρότερα σωματίδια να μην ξεπερνιέται το όριο μέτρησης του αισθητήρα (6553.4 ug/m^3).
2. Υπάρχει σημειωτέα αύξηση στο VOC index, όμως όχι τόσο μεγάλη για να προκαλούσε συναγερμό στο σύστημα. Θα χρειαζόταν μεγαλύτερη ποσότητα καπνού για να συμβεί αυτό.
3. Δεν υπάρχει αύξηση στο NOx index. Το διοξείδιο του αζώτου που περιέχει ο καπνός του τσιγάρου δεν είναι αρκετός από ένα μόνο φύσημα των 2 δευτερολέπτων να ανεβάσει τον δείκτη αυτόν.
4. Οι τιμές όλων των παραπάνω συγκεντρώσεων επέστρεψαν γρήγορα στα αναμενόμενα επίπεδα. Τα μικροσωματίδια μετά από περίπου 1 λεπτό, ενώ το VOC index μετά από περίπου 2 λεπτά.
5. Η αύξηση της υγρασίας κατά περίπου 5%, έχει να κάνει προφανώς από την υγρασία που προστίθεται λόγω της αναπνοής κατά την εκπνοή του καπνού από το τσιγάρο και μάλιστα από απόσταση λίγων εκατοστών. Το ποσοστό υγρασίας που μετράει ο αισθητήρας επέστρεψε στα φυσιολογικά επίπεδα μετά από περίπου 2 λεπτά, από τη στιγμή που ανιχνεύθηκε ο καπνός.

➔ **Τελικό συμπέρασμα:** Τα μικροσωματίδια παρουσιάζουν ραγδαία αύξηση, αφού η τιμή γίνεται κατά χιλιάδες φορές μεγαλύτερη. Ο VOC index αυξάνεται αλλά σε πολύ μικρότερο βαθμό, περίπου 2.5 φορές. Δεν παρατηρήθηκε αύξηση στο NOx index σε αυτές τις ποσότητες καπνού στις οποίες εκτέθηκε ο αισθητήρας, αν και ο καπνός του τσιγάρου περιέχει κάποιες ποσότητες οξειδίων του αζώτου και κυρίως διοξειδίου.

8.1.7 Μεταβολές στον VOC index

Ο πιο εύκολος τρόπος να δούμε τον VOC index να ανεβαίνει είναι να εκθέσουμε τον αισθητήρα σε βουτάνιο, το οποίο είναι εξαιρετικά πτητικό. Ο αναπτήρας είναι ένας πολύ απλός τρόπος για να παρέχουμε το βουτάνιο.

Ο αναπτήρας κρατήθηκε σε πολύ κοντινή απόσταση, περίπου 2 cm και για περίπου 15 δευτερόλεπτα. Στην εικόνα 47 το διάστημα κατά το οποίο ο αισθητήρας εκτέθηκε στο βουτάνιο, είναι ανάμεσα στις δύο κόκκινες διακεκομμένες γραμμές. Ο συνολικός χρόνος στο γράφημα είναι περίπου 6 λεπτά. Οι πληροφορίες αυτές μπορούν να φανούν εντός του αρχείου 20240107_sn03_env_gas.log και οι χρονικές στιγμές των κόκκινων γραμμών στο αρχείο 20240107_sn03_env_gas_notes.log.



Εικόνα 47 - Απόκριση περιβαλλοντολογικού αισθητήρα στην έκθεση σε βουτάνιο

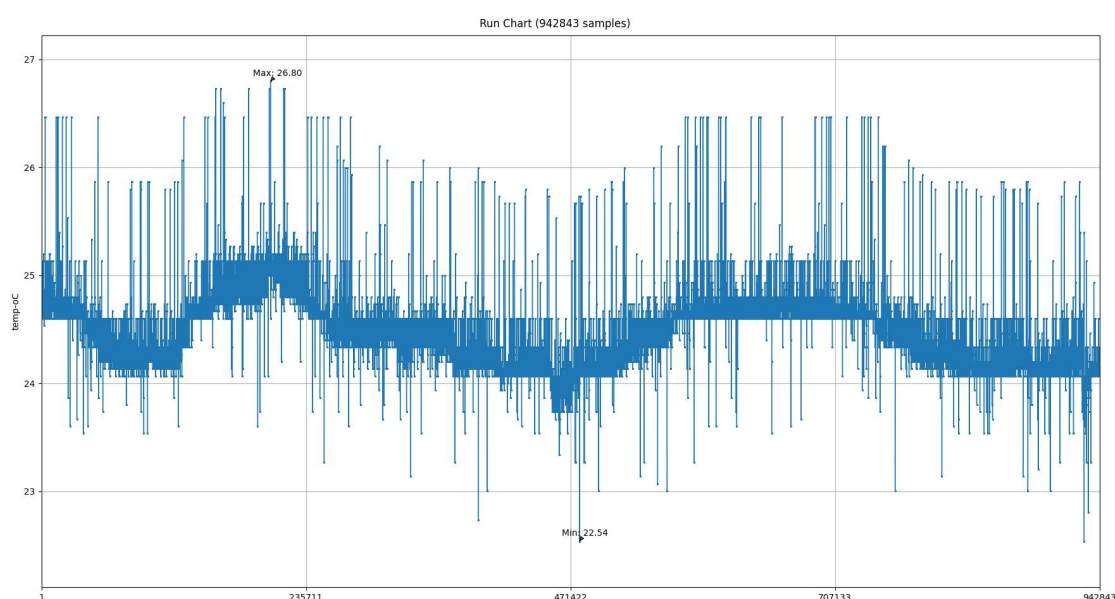
- ➔ **Συμπέρασμα:** Τα αποτελέσματα δείχνουν αναμενόμενα, αν και το πείραμα κράτησε μόνο 15 δευτερόλεπτα, ώστε να μην υπάρχει περίπτωση καταστροφής του αισθητήρα από την έκθεσή του σε πολύ μεγάλες συγκεντρώσεις βουτανίου. Παρόλα αυτά το VOC index ξεκίνησε να ανεβαίνει σταθερά περίπου από το 85 στο 140. Εάν αφήναμε και άλλο τον αναπτήρα να αναδύει βουτάνιο, είναι σίγουρο από τη δυναμική του γραφήματος ότι αυτή η τιμή θα ανέβαινε και άλλο. Αύξηση παρατηρήθηκε και στην υγρασία, αφού το βουτάνιο βγαίνοντας από το στόμιο του αισθητήρα είναι αρχικά σε υγρή μορφή, πριν αεριοποιηθεί ταχύτατα.

8.2 Αναλογικός αισθητήρας θερμοκρασίας

Μετρήθηκε ο αισθητήρας ΚΤΥ81-110. Έγιναν συγκεκριμένες μετρήσεις με συγκεκριμένους τύπους συμβάντων, τα οποία παρουσιάζονται παρακάτω.

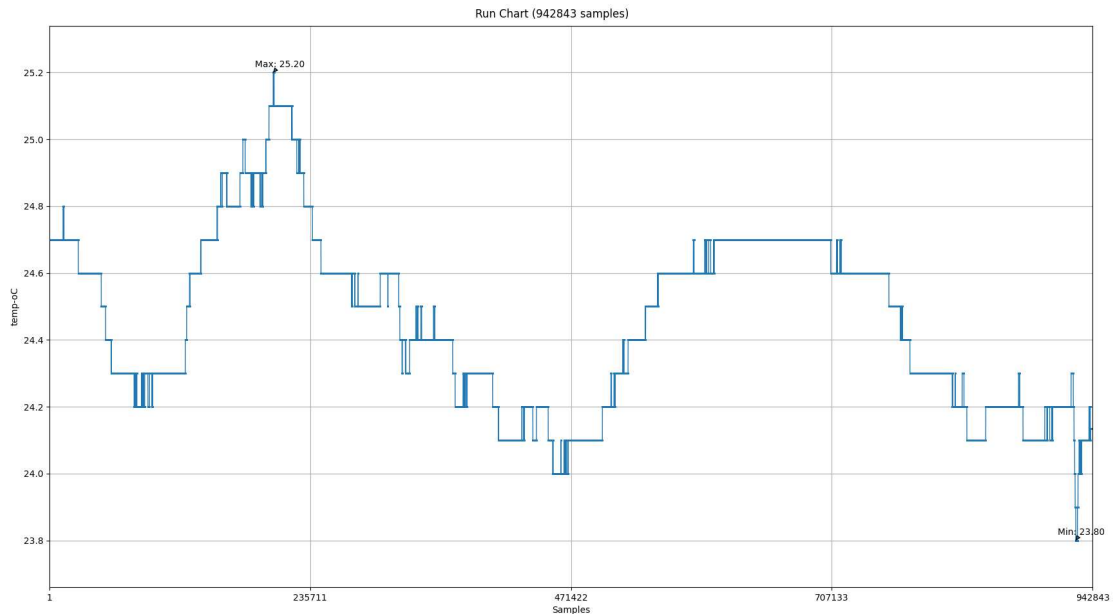
8.2.1 Μακροπρόθεσμη μέτρηση ηρεμίας σε εσωτερικό χώρο

Στην εικόνα 48 φαίνεται η απόκριση του αισθητήρα σε ηρεμία, χωρίς κάποιο συμβάν που να προκαλεί αλλαγή στην συμπεριφορά του. Ο χρόνος που αντιστοιχεί στο γράφημα είναι 17 ώρες. Οι πληροφορίες αυτές μπορούν να φανούν εντός του αρχείου *20231231_sn01_tmp17h.log*.



Εικόνα 48 - Απόκριση ηρεμίας αναλογικού αισθητήρα

Όπως φαίνεται από την εικόνα 48 υπάρχουν κυματώσεις, ακόμα και για ανεπαίσθητες μεταβολές στη μέτρηση του αισθητήρα. Αυτό έχει να κάνει με το ότι ο αισθητήρας είναι αναλογικός. Εάν η ψηφιακή λέξη που διαβάζεται από τον ADC μεταβληθεί ακόμα και κατά ένα ψηφίο, δημιουργούνται αυτές οι κυματώσεις. Στο κυρίως πρόγραμμα δεν υπάρχει κάποιο πρόβλημα με αυτό και επιπλέον η ΚΕΣ διαθέτει EMA filter (εκθετικό ΜΟ). Συνεπώς εφαρμόζοντας ένα απλό φίλτρο κυλιόμενου ΜΟ (εικόνα 49) μπορούμε να πάρουμε πιο εύκολα αναλύσιμη απεικόνιση. Τα συνολικά δείγματα είναι 942843 και συνεπώς ένα φίλτρο 2000 δειγμάτων είναι επαρκές, αφού και ο ΜΟ θα χρησιμοποιεί πολλά δείγματα, αλλά και τα δείγματα αυτά δεν θα «λείψουν» από τον συνολικό αριθμό δειγμάτων.



Εικόνα 49 – Η απόκριση της εικόνας 48, με εφαρμογή κυλιόμενου ΜΟ 2000 δειγμάτων

➔ **Συμπέρασμα:** Δεν φαίνεται να υπάρχει κάποια περίεργη συμπεριφορά στην μέτρηση του αισθητήρα. Κάποιες μεταβολές της θερμοκρασίας εντός της ημέρας (περίπου 1.5°C) είναι αναμενόμενες.

8.3 Σύγκριση τιμών μεταξύ των δύο τύπων αισθητηρίων

Τα δύο αισθητήρια διαθέτουν και τα δύο μέτρηση θερμοκρασίας, οπότε υπό κάποιες συνθήκες μπορούμε να συγκρίνουμε τα αποτελέσματά τους. Για παράδειγμα αν τοποθετηθούν οι αισθητήρες ο ένας δίπλα στον άλλον, μπορούμε να δούμε την απόκριση στη θερμοκρασία ηρεμίας αλλά και σε κάποιο συμβάν, όπως η τροφοδοσία με ζεστό αέρα, προερχόμενο από στεγνωτήρα μαλλιών (σεσουάρα). Θα πρέπει απλώς να ληφθεί υπόψη ότι ακόμα και αν όντως «τοποθετηθούν οι αισθητήρες ο ένας δίπλα στον άλλον», οι μετρήσεις είναι και πάλι προσεγγιστικές, αφού για πραγματική ακρίβεια θα έπρεπε να γίνει η μέτρηση σε περιβαλλοντικό θάλαμο.

Συνεπώς το αρχείο *sensor_logger.py* ρυθμίστηκε, στην πρώτη γραμμή της *main* (εικόνα 50), να επικοινωνεί και με τους τρεις αισθητήρες, στις διευθύνσεις 1, 2 και 3, όπου στη διεύθυνση τρία επικοινωνεί ο περιβαλλοντικός αισθητήρας SEN55 και στις διευθύνσεις 1 και 2 οι αισθητήρες θερμοκρασίας KTY81-110.

```

74  if __name__ == '__main__':
75      # Init L2
76      log_node = [1, 2, 3] # node address to log

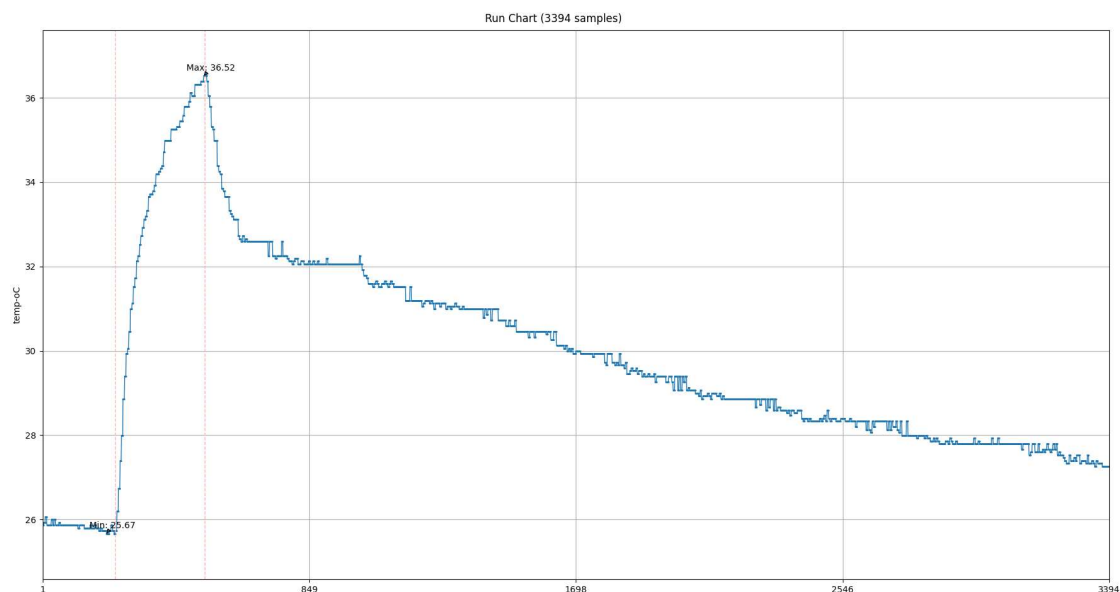
```

Εικόνα 50 - Το αρχείο *sensor_logger.py* ρυθμίζεται να επικοινωνεί με τρεις διευθύνσεις (1, 2, 3)

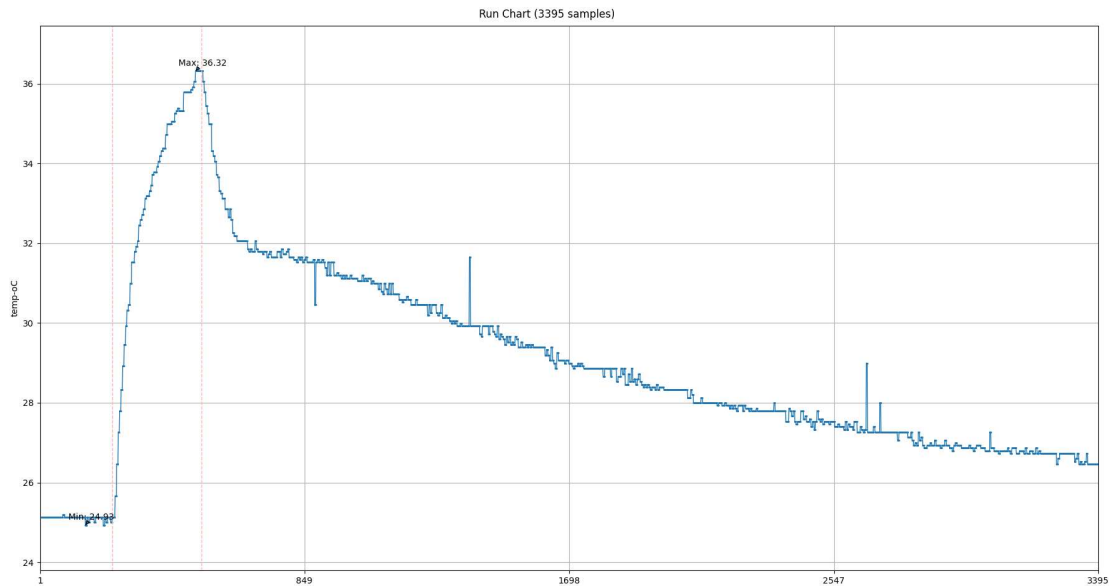
Το αρχείο στο οποίο καταχωρήθηκαν οι τιμές είναι το "20240104_sn01_temp_sn02_temp_sn03_env_hair_dryer.log". Όμως για να εξαχθούν οι γραφικές παραστάσεις θα πρέπει να απομονώσουμε τις μετρήσεις του κάθε αισθητήρα σε τρία διαφορετικά αρχεία, ώστε να διαβαστούν ξεχωριστά από το *sensor_analyzer.py*. Για να διαχωριστεί το συνολικό αρχείο σε τρία, χρησιμοποιήθηκε ένα τρίτο *python script*, το οποίο επισυνάπτεται στα παραδοτέα και είναι το *sensor_splitter.py*. Τα τρία διαφορετικά αρχεία που προέκυψαν αποθηκεύτηκαν με τις ονομασίες *20240104_sn01_temp_hair_dryer.log*, *20240104_sn02_temp_hair_dryer.log* και *20240104_sn03_env_hair_dryer.log*, ενώ τα αρχεία συμβάντων είναι αντίστοιχα τα *20240104_sn01_temp_hair_dryer_notes.log*, *20240104_sn02_temp_hair_dryer_notes.log* και *20240104_sn03_env_hair_dryer_notes.log*. Από αυτά τα αρχεία θα προκύψουν και τα γραφήματα που θα παρουσιαστούν στην παρακάτω παράγραφο.

8.3.1 Μετρήσεις

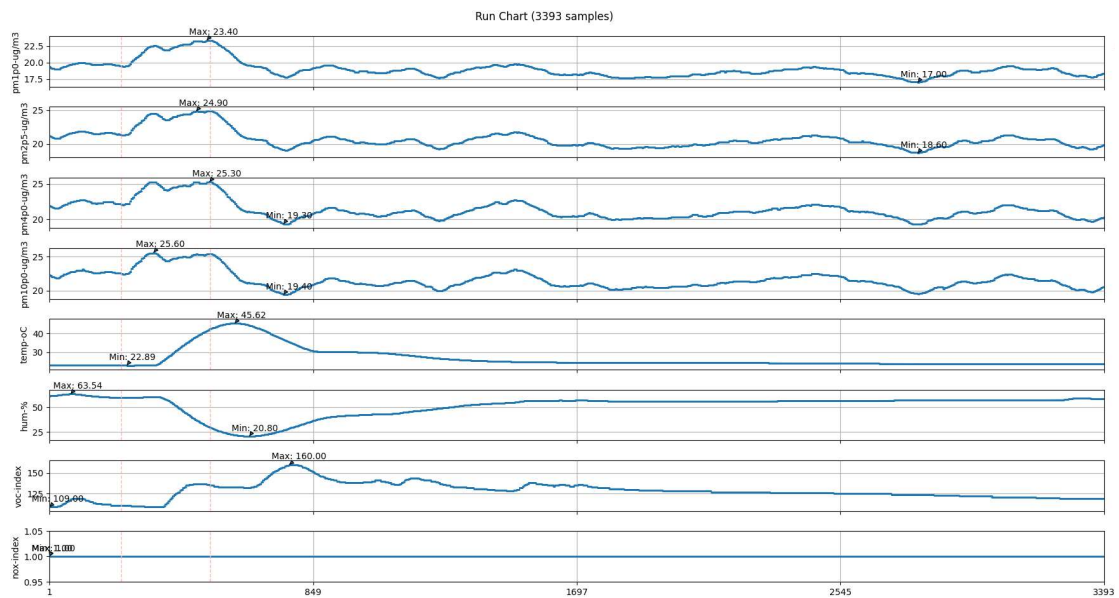
Αρχικά οι αισθητήρες ξεκίνησαν μετρώντας τη θερμοκρασία περιβάλλοντος, στη συνέχεια για 1 λεπτό δέχτηκαν θερμό αέρα από τον στεγνωτήρα μαλλιών και τέλος αφήθηκαν για κάποια ώρα να «ηρεμήσουν» και να επανέλθουν στην πραγματική θερμοκρασία περιβάλλοντος (εικόνα 51, εικόνα 52, εικόνα 53).



Εικόνα 51 - Απόκριση αισθητήρα θερμοκρασίας με διεύθυνση 1



Εικόνα 52 - Απόκριση αισθητήρα θερμοκρασίας με διεύθυνση 2



Εικόνα 53 - Απόκριση περιβατολογικού αισθητήρα με διεύθυνση 3

Θα πρέπει να παρατηρήσουμε και να συγκρίνουμε τα εξής:

- Σύγκριση θερμοκρασίας περιβάλλοντος μεταξύ των δύο αισθητήρων θερμοκρασίας.
- Σύγκριση θερμοκρασίας μεταξύ των αισθητήρων θερμοκρασίας και του περιβατολογικού αισθητήρα.
- Σύγκριση απόκρισης στο συμβάν μεταξύ των δύο αισθητήρων θερμοκρασίας.
- Σύγκριση απόκρισης στο συμβάν μεταξύ των αισθητήρων θερμοκρασίας και του περιβατολογικού αισθητήρα.

Από τις μετρήσεις στις εικόνες 51, 52 και 53 προκύπτουν τα εξής συμπεράσματα:

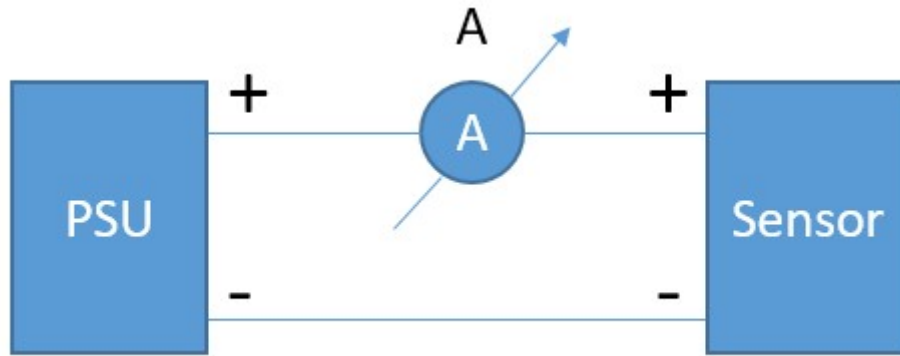
1. Οι δύο αναλογικοί αισθητήρες έχουν αξιοπρόσεκτη διαφορά μεταξύ τους στην ηρεμία, συγκεκριμένα 0.74°C. Ο ένας μετράει αρχικά 25.67°C και ο άλλος 24.93°C.
2. Ο SEN55 δείχνει 22.89 °C, διαφορά 2.4°C από τη μέση τιμή των δύο αναλογικών αισθητήρων.
3. Στην peak τιμή από τον ζεστό αέρα του στεγνωτήρα ο ένας αισθητήρας θερμοκρασίας έφτασε στους 36.52 °C και ο άλλος στους 36.32 °C, διαφορά 0.2 °C.
4. Ο SEN55 έδειξε peak τιμή 45.62 °C, διαφορά 9.2°C από τη μέση τιμή των αναλογικών.
5. Οι δύο αναλογικοί αισθητήρες ανταποκρίνονται ακριβώς με τον ίδιο τρόπο στο συμβάν θέρμανσης με τον στεγνωτήρα.
6. Όπως αναμένονταν ο περιβαντολογικός αισθητήρας παρουσιάζει καθυστέρηση στην απόκριση, λόγω του κυλιόμενου MO που εφαρμόζει εσωτερικά ο SEN55.

→ **Τελικό συμπέρασμα:** Οι αναλογικοί αισθητήρες χρειάζονται βαθμονόμηση (calibration) σύμφωνα με κάποιο όργανο αναφοράς πριν τη χρήση τους. Στη συγκεκριμένη περίπτωση και επειδή ο SEN55 έχει συγκριθεί με πραγματικό θερμόμετρο (παρ. 8.1.4) μπορεί να λογιστεί ως όργανο αναφοράς. Συνεπώς 2.4°C διαφορά από την πραγματική θερμοκρασία είναι σημαντική και πρέπει να αντισταθμιστεί. Σε σχέση με την απόκριση, οι KTY81-110 αποκρίνονται άμεσα, ενώ ο SEN55 έχει καθυστέρηση λόγω του εσωτερικού MO που εφαρμόζει, αλλά φαίνεται ότι με αυτόν τον τρόπο όταν ξεκινάει και μεταβάλλεται σύμφωνα με το συμβάν, έχει πιο γρήγορη προσαρμογή (μεγαλύτερος ρυθμός ανόδου της καμπύλης). Επίσης κατά τον τερματισμό της μέτρησης ο SEN55 έχει επιστρέψει στην αρχική θερμοκρασία ηρεμίας, ενώ οι KTY81-110 ακόμα βρίσκονται και οι δύο πάνω από 1°C μακριά από την τιμή αυτήν.

→ **Τελική σύγκριση:** Ο εσωτερικός MO που εφαρμόζει ο SEN55 κάνει την απόκριση ταχύτερη και ακριβέστερη, σε σχέση με τους αναλογικούς. Εξαιρεση αποτελούν οι πρώτες στιγμές του συμβάντος θέρμανσης, που και πάλι λόγω του εσωτερικού MO, ο SEN55 έχει πιο αργή αντίδραση στο φαινόμενο, όμως αυτό είναι κάτι παροδικό. Επίσης ως απόλυτη τιμή και πάλι ο SEN55, σύμφωνα με το τελικό συμπέρασμα, βρίσκεται πιο κοντά στην πραγματικότητα, με τους KTY81-110 να χρειάζονται βαθμονόμηση από το software πριν τη χρήση τους.

8.4 Μετρήσεις ρεύματος

Ενδιαφέρον στις μετρήσεις ρεύματος παρουσιάζουν οι τιμές όταν το σύστημα έχει και τους δύο αισθητήρες συνδεδεμένους, αλλά και όταν κάποιος από τους δύο ή και οι δύο απουσιάζουν. Πάρθηκαν μετρήσεις με αμπερόμετρο και στις 3 πλακέτες slave, χρησιμοποιώντας τη διάταξη της εικόνας 54, όπου ανάμεσα στα '+' του τροφοδοτικού και της πλακέτας παρεμβάλλεται σε σειρά το αμπερόμετρο, ενώ τα '-' των συσκευών συνδέονται απευθείας μεταξύ τους.



Εικόνα 54 - Μετρήσεις ρεύματος με τη χρήση αμπερομέτρου

Το αμπερόμετρο χρησιμοποιήθηκε στη λειτουργία «max/min hold», όπου καταγράφεται η μέση, ελάχιστη και μέγιστη τιμή, για όσο χρόνο είναι ενεργοποιημένη η λειτουργία αυτή. Στον πίνακα 2 παρουσιάζονται οι τιμές και για τις τρεις πλακέτες με σειριακό αριθμό 1, 2 και 3.

	I_a mA	I_{amx} mA	I_{mn} mA	I_{ae} mA	I_{mxe} mA	I_{mne} mA	I_{at} mA	I_{mxt} mA	I_{mnt} mA	I_{aet} mA	I_{mxt} mA	I_{mnet} mA
SN01	47.8	50.6	46.5	13.5	13.9	13.1	47.1	51.1	46.3	12.6	13.3	12.5
SN02	47.6	51	46.4	13.6	14.1	13	46.8	51.8	46	12.8	13.1	12.6
SN03	47.6	50.8	46.7	13.6	13.9	13	46.6	51.4	45.8	12.6	13.4	12.5
Avg	47.7	50.8	46.5	13.6	14	13	46.8	51.4	46	12.7	13.3	12.5

Πίνακας 2 – Μετρήσεις ρεύματος αισθητήρων

Όλες οι μετρήσεις πάρθηκαν σε διάρκεια χρόνου 10 λεπτών. Όλα τα ρεύματα είναι σε mA και οι ονομασίες του κάθε ρεύματος είναι οι εξής:

I_a : Μέσο ρεύμα πλακέτας

I_{amx} : Μέγιστο ρεύμα πλακέτας (maximum peak)

I_{amn} : Ελάχιστο ρεύμα πλακέτας (minimum peak)

I_{ae} : Μέσο ρεύμα πλακέτας με τον περιβαλλοντικό αισθητήρα αποσυνδεδεμένο

I_{amxe} : Μέγιστο ρεύμα πλακέτας με τον περιβαλλοντικό αισθητήρα αποσυνδεδεμένο

I_{amne} : Ελάχιστο ρεύμα πλακέτας με τον περιβαλλοντικό αισθητήρα αποσυνδεδεμένο

I_{at} : Μέσο ρεύμα πλακέτας με τον αισθητήρα θερμοκρασίας αποσυνδεδεμένο

I_{amxt} : Μέγιστο ρεύμα πλακέτας με τον αισθητήρα θερμοκρασίας αποσυνδεδεμένο

I_{amnt} : Ελάχιστο ρεύμα πλακέτας με τον αισθητήρα θερμοκρασίας αποσυνδεδεμένο

I_{aet} : Μέσο ρεύμα πλακέτας και με τους δύο αισθητήρες αποσυνδεδεμένους

I_{amxet} : Μέγιστο ρεύμα πλακέτας και με τους δύο αισθητήρες αποσυνδεδεμένους

I_{amnet} : Ελάχιστο ρεύμα πλακέτας και με τους δύο αισθητήρες αποσυνδεδεμένους

8.4.1 Αξιολόγηση μετρήσεων ρεύματος

Η αξιολόγηση γίνεται με αναφορά τον πίνακα 2. Οι μέγιστες και ελάχιστες τιμές δεν μπορούν εύκολα να αξιολογηθούν. Το μόνο ενδιαφέρον στοιχείο είναι αν είναι κοντά μεταξύ τους και πράγματι είναι. Η στήλη I_{amx} για παράδειγμα, έχει τρεις τιμές κοντινές μεταξύ τους, με ΜΟ 50.8mA. Το ίδιο συμβαίνει και με τις μέσες τιμές, οι οποίες είναι και αυτές κοντά. Συνεπώς η περεταίρω αξιολόγηση έχει να κάνει με τις καταναλώσεις αυτές κάθε αυτές.

Μελετάμε τις μέσες τιμές: $I_a = 47.7$ mA, $I_{ae} = 13.5$ mA, $I_{at} = 47.1$ mA, $I_{aet} = 12.6$ mA. Από αυτές τις τιμές μπορούμε να εξάγουμε συμπεράσματα για τις επιμέρους καταναλώσεις της πλακέτας και των αισθητήρων. Συγκεκριμένα:

Κατανάλωση πλακέτας δίχως τους αισθητήρες: Προκύπτει από την τιμή I_{aet} , όπου και τα δύο αισθητήρια είναι αποσυνδεδεμένα (12.6 mA).

Κατανάλωση αισθητήρων: Από υπολογισμούς με τις τιμές του πίνακα 2 προκύπτουν και οι καταναλώσεις των αισθητήρων. Όπου I_e , I_t είναι το μέσο ρεύμα του περιβαλλοντικού αισθητήρα και του αισθητήρα θερμοκρασίας αντίστοιχα. Επίσης έχει ήδη αναφερθεί ακριβώς παραπάνω ότι $I_{aet} = 12.6$ mA.

$$I_{ae} + I_e = I_a \Rightarrow 13.6 + I_e = 47.7 \Rightarrow I_e = 34.1\text{mA}$$

$$I_{aet} + I_e + I_t = I_a \Rightarrow 12.6 + 34.1 + I_t = 47.7 \Rightarrow I_t = 1\text{mA}$$

Άρα

Κατανάλωση πλακέτας χωρίς τους αισθητήρες: 12.6mA

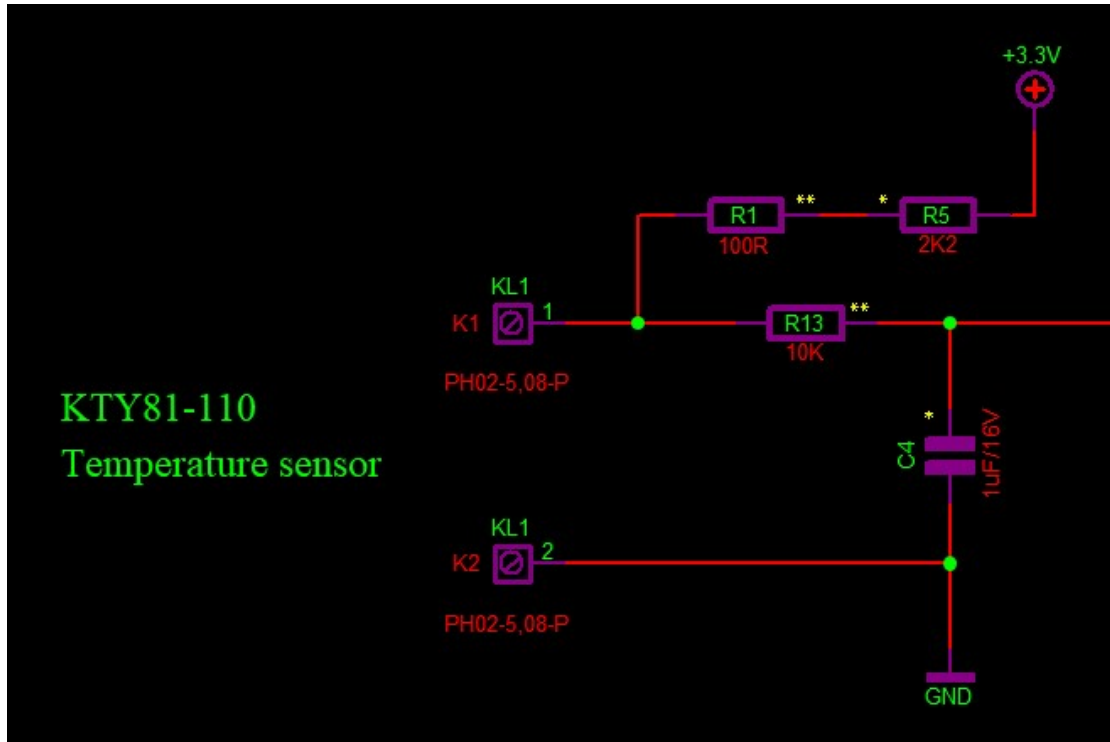
Κατανάλωση περιβαλλοντικού αισθητήρα: 34.1mA

Κατανάλωση αισθητήρα θερμοκρασίας: 1mA

Για τον περιβαλλοντικό αισθητήρα αυτή η κατανάλωση φαντάζει λογική (SENSIRION Datasheet, 2022, p. 11), αφού οι καταναλώσεις που δίνονται από τον κατασκευαστή είναι όντως τάξεως δεκάδων mA.

Δεν μπορεί να αξιολογηθεί με ακρίβεια το αποτέλεσμα, παρά ως τάξη μεγέθους. Ο λόγος είναι ότι η μέτρηση δεν είναι συνεχόμενη όπως υποθέτει το datasheet, αλλά συμβαίνει μια φορά το δευτερόλεπτο για κάποιες δεκάδες millisecond, τα οποία δεν μπορούν να μετρηθούν με ακρίβεια, αφού λαμβάνουν χώρα εσωτερικά στις βιβλιοθήκες του SEN55. Ως τάξη μεγέθους συνεπώς, το datasheet με τις μετρήσεις βρίσκονται στο ίδιο επίπεδο.

Σε σχέση με τον αισθητήρα θερμοκρασίας και εκεί δεν μπορεί να αποδοθεί με ακρίβεια η αντιστοιχία μετρήσεων με datasheet, αφού η αντίσταση του αισθητήρα αλλάζει με τη θερμοκρασία. Το κύκλωμα στο οποίο συνδέεται ο ΚΤΥ81-110 φαίνεται στην εικόνα 55.



Εικόνα 55 - Σύνδεση αισθητήρα ΚΤΥ81-110 στην πλακέτα

Επίσης από το datasheet του ΚΤΥ81-110 (εικόνα 56), προκύπτει ότι η αντίσταση στους 25°C είναι περίπου 1K. Οι μετρήσεις πάρθηκαν σε θερμοκρασία δωματίου, οπότε με καλή προσέγγιση υποθέτουμε ότι η τιμή είναι περίπου 1K.

Ambient temperature		Temperature coefficient (%/K)	ΚΤΥ81/110 Resistance (Ω)		
(°C)	(°F)		Min	Typ	Max
-55	-67	0.99	475	490	505
-50	-58	0.98	500	515	530
-40	-40	0.96	552	567	582
-30	-22	0.93	609	624	638
-20	-4	0.91	669	684	698
-10	14	0.88	733	747	761
0	32	0.85	802	815	828
10	50	0.83	874	886	898
20	68	0.80	950	961	972
25	77	0.79	990	1000	1010

Εικόνα 56 - Η τυπική αντίσταση του ΚΤΥ81-110 στους 25°C είναι 1K [19]

Άρα στο κύκλωμα του αισθητήρα εάν μεταξύ των ακροδεκτών K1 και K2 υποθέσουμε αντίσταση 1K, τότε από τον διαιρέτη τάσης, η τάση στα άκρα του αισθητήρα (V_{K1}) θα είναι:

$$V_{K1} = 3.3V * (1K / 3.3K) \Rightarrow V_{K1} = 1V.$$

Που σημαίνει ότι για αντίσταση 1K και εφαρμόζοντας τον νόμο του Ohm προκύπτει ότι το ρεύμα του αισθητήρα (I_s) είναι:

$I_s = 1mA$, που είναι και η τιμή που προέκυψε από τις μετρήσεις.

Συμπέρασμα: Τα ρεύματα που προέρχονται από τις μετρήσεις μπορούν να θεωρηθούν αξιόπιστα με βάσει τις αναμενόμενες, μέσω των datasheet, τιμών.

8.4.2 Μελέτη κατανάλωσης ρεύματος

Σύμφωνα με τα παραπάνω στοιχεία, το ερώτημα που τίθεται είναι εάν θα μπορούσε αυτή η εφαρμογή να εξελιχθεί σε δίκτυο αισθητήρων χαμηλής κατανάλωσης. Η απάντηση είναι σαφώς «όχι», αφού μια μπαταρία 6V/1550mAh θα χρειάζονταν μόλις 32 ώρες για να αδειάσει.

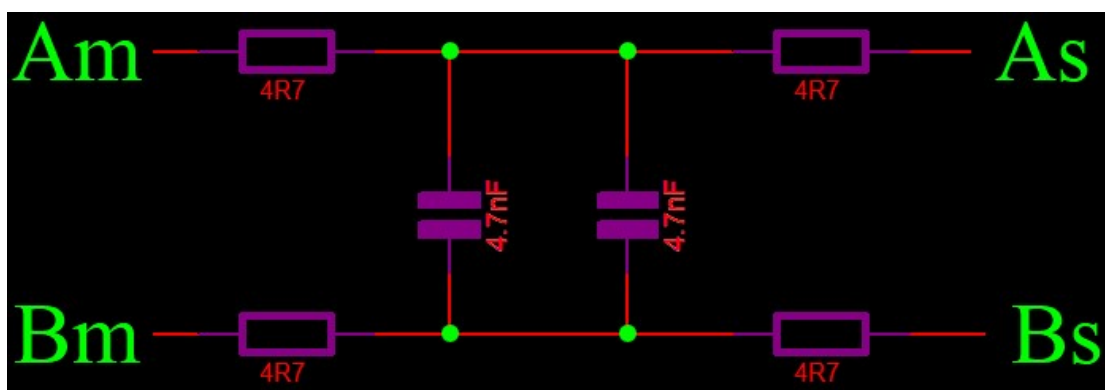
Άρα είναι σαφές ότι η συσκευή θα πρέπει να συνοδεύεται από τροφοδοτικό, το οποίο θα παίρνει τα 230V της ΔΕΗ και θα βγάζει έξοδο DC, πχ για τη συγκεκριμένη εφαρμογή τα 9V ή τα 12V είναι εφαρμόσιμη περίπτωση. Τα τροφοδοτικά αυτά δεν βγαίνουν με μεγάλο κόστος, αφού υπάρχουν έτοιμα “modules”, πολλά μάλιστα από αυτά τοποθετούνται απευθείας και επάνω στην πλακέτα. Όπως σε όλες αυτού του τύπου συσκευές που δεν τροφοδοτούνται από μπαταρία, οι εγκαταστάτες τα τοποθετούν δίπλα στα κουτιά διακλάδωσης (μπουάτ), ώστε να μη χρειάζεται να τραβήξουν πολλά μέτρα καλώδια (πόσο μάλλον καλωδίων στα 230V).

Να σημειωθεί ότι με τον συγκεκριμένο αισθητήρα δεν μπορεί να εφαρμοστεί η λύση να κόβεται η τροφοδοσία του αισθητήρα και να περνάει η πλακέτα σε λειτουργία χαμηλής κατανάλωσης. Ο συγκεκριμένος αισθητήρας χρειάζεται περίπου 1 λεπτό για να παίρνει τις ακριβέστερες δυνατές μετρήσεις και επίσης διαθέτει εσωτερικό κυλιόμενο ΜΟ, ο οποίος έχει χρόνο ολοκλήρωσης μία μέρα. Άρα η λύση του επιπλέον τροφοδοτικού είναι μονόδρομος στη συγκεκριμένη περίπτωση.

8.5 Μετρήσεις RS-485

Ένα επιπλέον ποιοτικό στοιχείο είναι η σταθερότητα των επικοινωνιών στο δίκτυο RS-485. Η σταθερότητα αυτή αποκτά ακόμα μεγαλύτερο νόημα για μεγάλα μήκη καλωδίων, αφού το RS-485 στην ταχύτητα επικοινωνίας 115200bps θα πρέπει να λειτουργεί αποτελεσματικά έως και περίπου 1300 μέτρα καλώδιο [30]. Για τη συγκεκριμένη εφαρμογή ακόμα και τα 400 μέτρα θεωρούνται επαρκή, αλλά η συγκεκριμένη δοκιμή είναι δύσκολη να γίνει, αφού θα χρειαστούν εκατοντάδες μέτρα καλώδιο. Μπορεί όμως να γίνει προσομοίωση με μια κατάλληλη αντίσταση σειράς. Επίσης πολύ σημαντική είναι και η χωρητικότητα του

καλωδίου, ίσως και πιο σημαντικό από την αντίσταση. Για να αποκτηθεί αντίληψη για τις τιμές αντίστασης και χωρητικότητας του καλωδίου, θα πρέπει να γίνει αναφορά σε κάποιο καλώδιο RS-485, ενός γνωστού, παγκοσμίως, κατασκευαστή. Η αμερικάνικη Belden είναι ένας τέτοιος κατασκευαστής. Στο καλώδιο RS-485 που κατασκευάζει, τα χαρακτηριστικά που παραθέτει είναι για την αντίσταση 6.9 Ohm/1000ft και για την χωρητικότητα 12.3 pF/ft [31]. Το κύκλωμα που προσομοιώνει τα 400m καλωδίου φαίνεται στην εικόνα 57.



Εικόνα 57 - Κύκλωμα προσομοίωσης για 400 μέτρα στρεπτού ζεύγους καλωδίου

Όπου Am, Bm τα σήματα A, B από την πλευρά του master και As, Bs τα αντίστοιχα σήματα των slave. Οι αντιστάσεις χωρίστηκαν σε δύο ίσα μέρη για κάθε γραμμή, ώστε να προσομοιώνουν τα 400 μέτρα σε δύο μέρη των 200 μέτρων. Αυτό χρειάζεται ώστε στη μέση του καλωδίου να προστεθεί χωρητικότητα, για να προσομοιωθεί η χειρότερη περίπτωση χωρητικότητας, η οποία είναι στη μέση, δηλαδή στα 200 μέτρα. Παρακάτω οι υπολογισμοί που οδήγησαν στην επιλογή των τιμών αυτών.

Τα δεδομένα είναι 400 μέτρα καλώδιο, με αντίσταση 6.9 Ohm/1000ft και χωρητικότητα 12.3 pF/ft.

Υπολογισμός αντίστασης στα 400 μέτρα:

Ένα πόδι (ft) αντιστοιχεί σε 0,3048 μέτρα. Αυτό δίνει αντίσταση ανά μέτρο ίση με:

$$2.83 \text{ Ohm}/1000\text{ft} = 2.83\text{Ohm}/304,8\text{m} \Rightarrow R_{1\text{m}} \approx 0.023\text{Ohm}/\text{m}$$

$$\text{Για } 400\text{m}: R_{400\text{m}} = 0.023 * 400 \Rightarrow R_{400\text{m}} = 9.2 \text{ Ohm}$$

Υπολογισμός χωρητικότητας στα 200 μέτρα (η μισή διαδρομή του καλωδίου):

$$12.3 \text{ pF}/\text{ft} = 12.3\text{pF}/0.3048\text{m} \Rightarrow C_{1\text{m}} \approx 40.35\text{pF}/\text{m}$$

$$\text{Για } 200\text{m}: C_{200\text{m}} = 40.35 * 200 \Rightarrow C_{200\text{m}} \approx 8071\text{pf}$$

Στην πράξη δεν διατίθενται αυτές οι τιμές των αντιστάσεων, οπότε επιλέχθηκαν τιμές που μπορούν να βρεθούν στο εμπόριο: $R=4.7 \text{ Ohm}$ και $C = 9.4\text{nF}$ ($2 \times 4.7\text{nF}$ παράλληλα).

Οι τελευταίοι υπολογισμοί αφορούν την επαλήθευση για το πόσα, περίπου, μέτρα προσομοιώνουν αυτές οι τιμές.

9.4 Ohm: για 0.023Ohm/m προσομοιώνονται $9.4/0.023 \approx 409\text{m}$ από άκρο σε άκρο

9.4 nF: για 40.35pF/m προσομοιώνονται $9400/40.35 \approx 233\text{m}$ από άκρο στη μέση της διαδρομής του καλωδίου.

Όπως είναι φανερό και από τους παραπάνω υπολογισμούς δεν μπορεί να αποδοθεί με ακρίβεια το πόσα μέτρα ακριβώς προσομοιώνονται, αφού η προσομοίωση των μέτρων για την αντίσταση διαφέρουν ελαφρώς από αυτήν της χωρητικότητας. Όμως στη συγκεκριμένη περίπτωση δεν ενδιαφέρει και τόσο η ακρίβεια, όσο το να αποκτηθεί μια εικόνα για το τι συμβαίνει στα περίπου 400m καλωδίου, ώστε να γίνει κατανοητό εάν το δίκτυο μπορεί να λειτουργήσει όντως επιτυχώς σε αυτά τα μήκη καλωδίου. Ενδιαφέρει όχι μόνο το να υπάρχει επικοινωνία (προφανώς), αλλά και τα χαμένα πακέτα (αν υπάρχουν) να βρίσκονται σε αποδεκτά επίπεδα (μικρότερα του 1% των συνολικών πακέτων).

Χρησιμοποιήθηκαν και οι τρεις πλακέτες. Το δικτύωμα RC προστέθηκε από τον μετατροπέα USB σε RS-485 (ακροδέκτες Am και Bm στην εικόνα 57), προς τους slave (ακροδέκτες As και Bs). Οι μετρήσεις φαίνονται παρακάτω:

Χρόνος πειράματος: 15 λεπτά

Επικοινωνίες που πραγματοποιήθηκαν: 12857

Πακέτα που χάθηκαν: 59

Ποσοστό σφάλματος πακέτων: $59/12857 = 0.46\%$

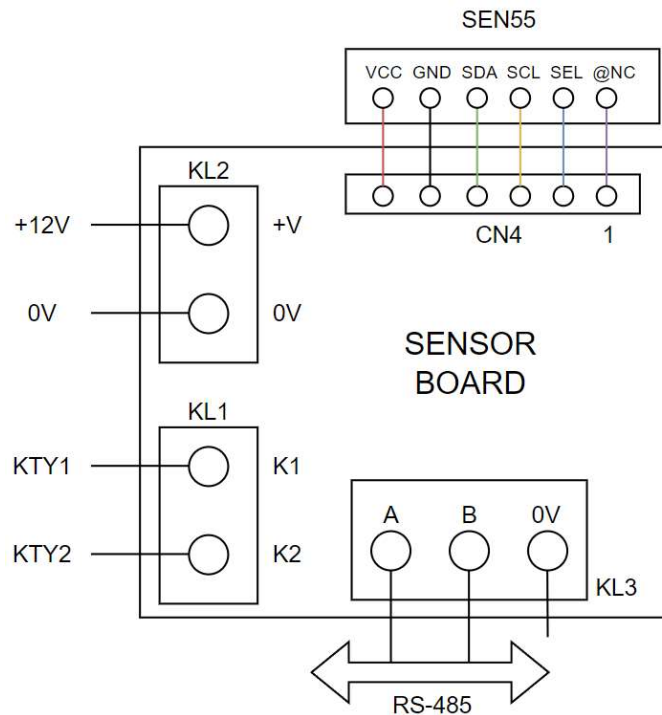
Το ποσοστό αυτό είναι ικανοποιητικό, όμως ο κατάλληλος τερματισμός γραμμής θα πρέπει να τηρείται πάντα, ώστε το ποσοστό αυτό να μη χειροτερέψει με την αύξηση των κόμβων.

9 Εγκατάσταση και εκκίνηση συστήματος

9.1 Συνδέσεις συστήματος

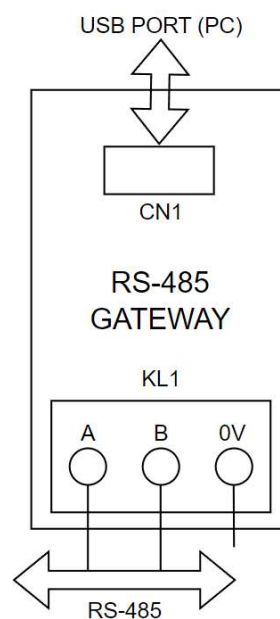
Οι συνδέσεις του συστήματος πριν την εγκατάσταση είναι απλή. Οι ανιχνευτές θα πρέπει να τροφοδοτούνται με τάση 12V DC και να είναι συνδεδεμένοι στο δίκτυο RS-485, όπου θα πρέπει να είναι συνδεδεμένος και ο προσαρμογέας δικτύου. Ο τελευταίος θα πρέπει να έχει συνδεθεί μέσω USB θύρας και στον Η/Υ (εικόνα 58). Στις πλακέτες των αισθητήρων θα πρέπει να είναι

συνδεδεμένος και ο αντίστοιχος αισθητήρας, ο SEN55 στην περίπτωση του περιβαλλοντικού αισθητήρα και ο KTY81-110 στην περίπτωση του αισθητήρα θερμοκρασίας. Να σημειωθεί ότι στην περίπτωση του SEN55 τα χρώματα των συνδέσεων στον connector CN4 (εικόνα 58), αντιστοιχεί με τα πραγματικά χρώματα των καλωδίων επάνω στον SEN55, ενώ στον αισθητήρα θερμοκρασίας ο αισθητήρας δεν έχει πολικότητα (είναι απλά μια μεταβλητή αντίσταση) και συνεπώς η φορά σύνδεσης στην κλέμμα KL1 δεν έχει σημασία.



Εικόνα 58 - Συνδέσεις πλακέτας αισθητήρα

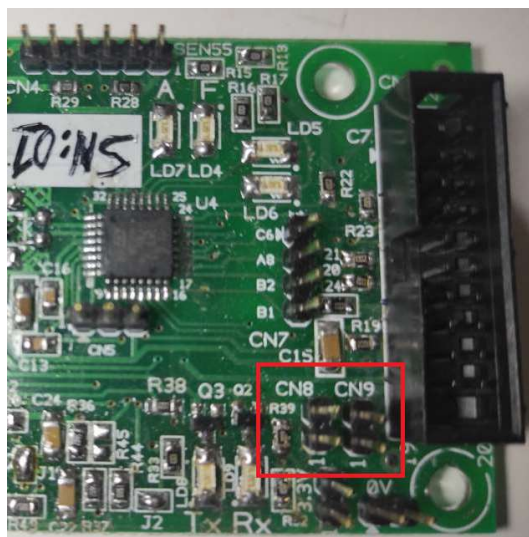
Οι αντίστοιχες συνδέσεις του προσαρμογέα δικτύου, φαίνονται στην εικόνα 59.



Εικόνα 59 - Συνδέσεις προσαρμογέα δικτύου RS-485

9.1.1 Επιλογή διεύθυνσης αισθητήρα

Γίνεται από τα jumper CN8 και CN9 (εικόνα 60).



Εικόνα 60 - Επιλογή διεύθυνσης αισθητήρα μέσω των CN8, CN9

Στις συσκευές του εμπορίου διατίθεται DIP switch (κυλιόμενοι διακόπτες), ώστε να μπορούν να γίνουν 2ⁿ διαφορετικές επιλογές διευθύνσεων, όπου n ο αριθμός των διακοπών. Συνεπώς στην περίπτωση οκταπλού dip switch, θα μπορούσαν να επιλεγούν 256 διαφορετικές διευθύνσεις. Τα μοντέρνα συστήματα τείνουν να καταργήσουν ακόμα και τα dip switch, ώστε η επιλογή διεύθυνσης να γίνεται από ειδικά προγράμματα ή συσκευές που επικοινωνούν με τον κάθε αισθητήρα ξεχωριστά και τους αναθέτουν τη διεύθυνση που καταχωρεί ο χρήστης. Με αυτόν τον τρόπο μειώνεται το κόστος κατασκευής της πλακέτας του αισθητήρα, αφού απαλλάσσεται από το κόστος του dip switch.

Για τις ανάγκες της παρούσας εργασίας τα jumper δίνουν τέσσερις διαφορετικούς συνδυασμούς, από τους οποίους προκύπτουν τέσσερις διαφορετικές διευθύνσεις (πίνακας 3). Η διεύθυνση που θα ρυθμιστεί για κάθε πλακέτα, είναι αυτή που θα στέλνεται στον master του δικτύου αισθητήρων (PC).

CN8	CN9	Διεύθυνση
Ανοιχτός	Ανοιχτός	1
Ανοιχτός	Κλειστός	2
Κλειστός	Ανοιχτός	3
Κλειστός	Κλειστός	4

Πίνακας 3 - Πιθανές διευθύνσεις αισθητήρα, ανάλογα με την επιλογή των CN8, CN9

→ **Σημείωση:** Η επιλογή της διεύθυνσης θα πρέπει να γίνει πριν ο αισθητήρας τροφοδοτηθεί με τάση, αφού η ανάγνωση των CN8 και CN9 και η καταχώρηση της διεύθυνσης στη μνήμη του, γίνεται μία και μοναδική φορά κατά την αρχικοποίηση.

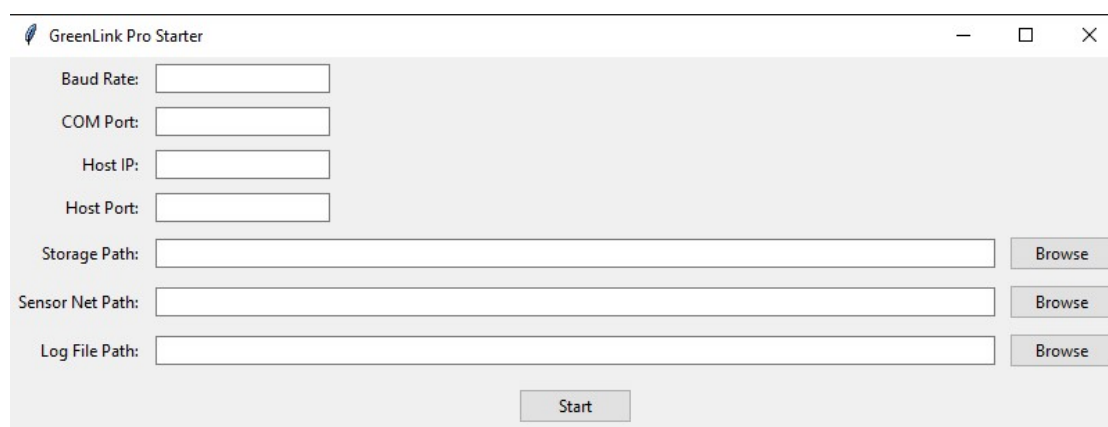
Όταν ξεκινήσουν να παίρνονται μετρήσεις και μετά, δεν μπορεί να αλλάξει η διεύθυνσή αυτή, ακόμα και αν αλλάξει η κατάσταση στα CN8 και CN9.

9.2 Εγκατάσταση λογισμικού ΚΕΣ (GreenLink Pro)

Έχοντας πραγματοποιήσει τις σωστές συνδέσεις, το επόμενο και τελευταίο στάδιο πριν την πλήρη λειτουργία του συστήματος είναι η εγκατάσταση της ΚΕΣ. Είναι εξαιρετικά απλή, αρκεί να διαθέτει κανείς το αρχείο εγκατάστασης «sensorpc.zip» που επισυνάπτεται στο παραδοτέο της εργασίας. Μπορεί να εγκατασταθεί σε οποιοδήποτε φάκελο του συστήματος και λειτουργεί σε όλα τα λειτουργικά συστήματα (Windows, Linux, MAC). Αφού αποσυμπίεστεί το αρχείο, στη συνέχεια στον κεντρικό φάκελο υπάρχει το αρχείο «README.md», το οποίο στην ενότητα «Installation» περιέχει τα 3 απλά βήματα εγκατάστασης, με τη χρήση εκτελέσιμων αρχείων τύπου *.bat*.

9.3 Εκκίνηση με γραφικά

Η διαδικασία περιγράφεται εντός του αρχείου «README.md» στην ενότητα «Running the Application». Όταν γίνεται η εκκίνηση του προγράμματος εμφανίζεται η οθόνη της εικόνας 61.



Εικόνα 61 - Αρχική οθόνη εκκίνησης του λογισμικού του ΚΕΣ «GreekLinkPro»

Ο χρήστης θα πρέπει να ενημερώσει τα πεδία με τις σωστές τιμές και στη συνέχεια να πατήσει «Start». Εάν κάποια (ή και όλα) τα πεδία παραμείνουν κενά, τότε θα χρησιμοποιηθούν οι προεπιλεγμένες (default) τιμές, οι οποίες παρατίθενται στο αρχείο «README.md» στην ενότητα «Default Values». Η επεξήγηση για κάθε πεδίο χωριστά δίνεται αμέσως παρακάτω:

- **Baud Rate:** Η ταχύτητα της σειριακής επικοινωνίας. Οι αισθητήρες λειτουργούν στα 115200, οπότε για την ομαλή λειτουργία του συστήματος η τιμή αυτή δεν πρέπει να αλλάξει.

- **COM Port:** Η θύρα επικοινωνίας USB του Η/Υ, που συνδέεται με τον μετατροπέα δικτύου. Φαίνεται από τις ρυθμίσεις του λειτουργικού (πχ στα Windows φαίνεται στον Device Manager).
- **Host IP:** Η IP του server που θα επικοινωνεί με την ιστοσελίδα.
- **Host Port:** Η θύρα του server που θα επικοινωνεί με την ιστοσελίδα.
- **Storage Path:** Η διαδρομή για το αρχείο που θα χρησιμοποιηθεί για να σώζονται οι πληροφορίες του συστήματος. Σε περίπτωση που το αρχείο δεν υπάρχει, θα δημιουργηθεί αυτόματα.
- **Sensor Net Path:** Η διαδρομή για το αρχείο που θα χρησιμοποιηθεί, από το οποίο προκύπτει πόσοι αισθητήρες υπάρχουν, η διεύθυνση (id) του καθενός, καθώς και τι τύπου αισθητήρας είναι (περιβαντολογικός ή αισθητήρας θερμοκρασίας). Σε περίπτωση που το αρχείο δεν υπάρχει, θα δημιουργηθεί αυτόματα ένα κενό αρχείο. Όμως σε αυτήν την περίπτωση (ή στην περίπτωση που υπάρχει ήδη αρχείο αλλά είναι κενό ή με μη έγκυρη μορφή), το δίκτυο θα δημιουργηθεί χωρίς ενεργούς αισθητήρες, που σημαίνει ότι η εφαρμογή θα τρέχει μεν, αλλά χωρίς να κάνει κάτι ουσιαστικό, μη έχοντας αισθητήρες καταχωρημένους να επικοινωνήσει. Πάντως ακόμα και την ώρα που τρέχει η εφαρμογή, το αρχείο μπορεί να αλλάξει και η εφαρμογή θα το αντιληφθεί αυτό, αλλάζοντας δυναμικά το set-up των αισθητήρων σύμφωνα με τις αλλαγές του αρχείου.
- **Log File Path:** Η διαδρομή για το αρχείο που θα χρησιμοποιηθεί για να σώζονται τα συμβάντα του συστήματος (αρχείο log). Σε περίπτωση που το αρχείο δεν υπάρχει, θα δημιουργηθεί αυτόματα.

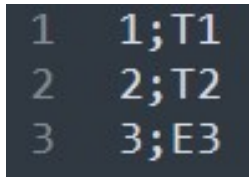
9.3.1 Αρχείο δικτύου αισθητήρων

Σε αυτό το αρχείο ο χρήστης καταχωρεί όλους τους αισθητήρες του δικτύου. Είναι πολύ απλό στην κατασκευή και την τροποποίηση. Το αρχείο είναι τύπου .csv και μπορεί να δουλευτεί τόσο ως αρχείο κειμένου, όσο και από το excel. Ένα παράδειγμα φαίνεται στην εικόνα 62.

	A	B
1	1	T1
2	2	T2
3	3	E3
4		

Εικόνα 62 - Παράδειγμα αρχείου δικτύωσης αισθητήρων στο Excel

Το ίδιο αρχείο (αρκεί πάντα να έχει επέκταση .csv) μπορεί να δουλευτεί και από κειμενογράφο (πχ windows notepad), αρκεί να τηρείται ο διαχωρισμός με ελληνικό ερωτηματικό και χωρίς κενά ενδιάμεσα, όπως φαίνεται στην εικόνα 63.



```
1 1;T1
2 2;T2
3 3;E3
```

Εικόνα 63 - Παράδειγμα αρχείου δικτύωσης αισθητήρων σε κειμενογράφο

Η κάθε γραμμή αποτελείται από τρεις διαφορετικές πληροφορίες

1. Αύξων αριθμός. Βρίσκεται πριν το ερωτηματικό. Είναι προαιρετικό πεδίο και όχι κρίσιμο για το δίκτυο αισθητήρων που θα δημιουργηθεί. Θα μπορούσε να χρησιμοποιηθεί οποιοσδήποτε αριθμός ή κείμενο, ή ακόμα θα μπορούσε να παραμείνει και κενό. Παρόλα αυτά είναι χρήσιμο εάν χρησιμοποιηθεί ως αύξοντας αριθμός, ώστε να διακρίνεται εύκολα πόσοι αισθητήρες υπάρχουν στο δίκτυο.
2. Τύπος αισθητήρα. Είναι το πρώτο γράμμα ακριβώς μετά το ερωτηματικό (ή το πρώτο γράμμα της δεύτερη στήλης στο excel). Επιτρεπτές τιμές είναι το αγγλικό κεφαλαίο 'E' (για τον περιβαντολλογικό αισθητήρα) και το αγγλικό 'T' (για τον αισθητήρα θερμοκρασίας).
3. Διεύθυνση (ID) αισθητήρα. Βρίσκεται ακριβώς μετά από το γράμμα που δείχνει τον τύπο αισθητήρα. Επιτρεπτές τιμές είναι όλοι οι ακέραιοι αριθμοί από το 1 έως και το 65535.

Εάν σε κάποια γραμμή ανιχνευτούν στα 2 και 3 οποιοσδήποτε τιμές πέρα από τις επιτρεπτές, αυτό θα οδηγήσει σε αγνόηση της συγκεκριμένης γραμμής. Από τις υπόλοιπες γραμμές (υποθέτοντας ότι είναι ορθά καταχωρημένες), οι αισθητήρες θα δημιουργηθούν κανονικά.

Συνεπώς στο παράδειγμα των εικόνων 62 και 63, θα δημιουργηθεί δίκτυο με τους εξής τρεις αισθητήρες:

ID 1: Αισθητήρας θερμοκρασίας

ID 2: Αισθητήρας θερμοκρασίας

ID 3: Περιβαντολλογικός αισθητήρας

Είναι αυτονόητο ότι οι διευθύνσεις αυτές θα πρέπει να συμφωνούν με την αντίστοιχη επιλογή των DIP switch της κάθε πλακέτας (παράγραφος 9.1.1).

9.4 Εκκίνηση με γραμμή εντολών

Το GreenLink Pro προσφέρει υποστήριξη και για εκκίνηση χωρίς να υπάρχει γραφικό περιβάλλον, εκκινώντας από τη γραμμή εντολών του εκάστοτε λειτουργικού συστήματος. Σε αυτήν την περίπτωση οι αρχικές ρυθμίσεις από τον χρήστη δίνονται ως ορίσματα στη γραμμή εντολών. Οδηγίες για το πώς γίνεται η εκκίνηση από τη γραμμή εντολών και πώς μπαίνουν

τα ορίσματα, δίνονται εντός του αρχείου «README.md», στην ενότητα «Running from the Command Line».

10 Software Κεντρικού Ελεγκτή Συστήματος (GreenLink Pro)

Είναι γραμμένο εξ' ολοκλήρου σε γλώσσα python. Το σύστημα αποτελείται από δύο βασικά threads:

- Σύστημα διαχείρισης αισθητήρα
- Web server

Παρακάτω θα περιγραφούν τα δύο αυτά threads και στη συνέχεια θα παρατεθεί ο τρόπος που συνδέονται μεταξύ τους. Στη συνέχεια θα αναλυθούν όλα τα modules του διαστρωματωμένου τμήματος (διαχείριση αισθητήρα), ενώ ειδική ανάλυση θα γίνει για το logging module, το οποίο ανήκει από μόνο του σε ένα ξεχωριστό επίπεδο, αυτό που ονομάζεται «δια-συστημικό επίπεδο» (system wide layer).

Ειδική περίπτωση αποτελεί το module αποθήκευσης (storage) το οποίο χρησιμοποιείται, όπως και το logging module, και από τα δύο threads, επειδή όμως ο server το χρησιμοποιεί μέσω interface που ανήκει στο σύστημα διαχείρισης αισθητήρα (UI interface), μπορεί να υπολογιστεί ως τμήμα του τελευταίου. Θα μπορούσε παρόλα αυτά να υπολογιστεί και αυτό ως μέρος του system wide layer, αφού βρίσκεται στα αρχιτεκτονικά όρια μεταξύ των δύο.

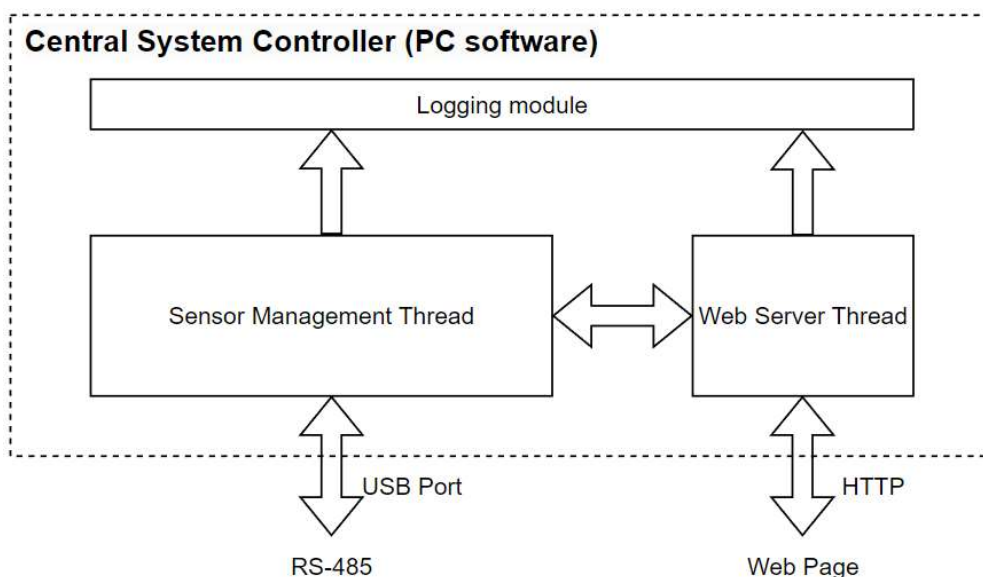
Ιδιαίτερη έμφαση σε αυτό το κεφάλαιο δίνεται η επεξήγηση όχι μόνο του λογισμικού και της αρχιτεκτονικής του, αλλά και για ποιον λόγο δομήθηκε κατ' αυτόν τον τρόπο και βάση ποιας μεθοδολογίας (Clean Architecture).

10.1 Threading system

Το σύστημα αποτελείται από δύο threads. Με τον όρο «threads» εννοούμε δύο ξεχωριστά τμήματα προγραμμάτων, τα οποία τρέχουν ανεξάρτητα μεταξύ τους. Τη διαχείριση των threads την αναλαμβάνει ο scheduler, ο οποίος τρέχει από ειδική βιβλιοθήκη της python (threading library).

Για να μπορέσουν τα threads να λειτουργήσουν αρμονικά μεταξύ τους και να συνεργαστούν, θα πρέπει να ακολουθούνται κάποιοι κανόνες. Τα διαφορετικά threads ενός προγράμματος έχουν συνήθως πρόσβαση σε κάποιους κοινούς πόρους. Για παράδειγμα θα μπορούσαν δύο threads να γράφουν στον ίδιο πίνακα δεδομένων, αλλά θα πρέπει να ληφθεί μέριμνα ώστε αυτό να μη γίνει ταυτόχρονα, αλλιώς υπάρχει κίνδυνος ο πίνακας να καταλήξει με «μπερδεμένα» δεδομένα και από τα δύο threads, καθιστώντας την πληροφορία λανθασμένη. Σε αυτό το σημείο αναλαμβάνει δράση ο scheduler, αρκεί ο χρήστης από το πρόγραμμα να έχει δηλώσει για τον πίνακα αυτόν ότι είναι κοινός πόρος. Δεν χρειάζεται να γίνει κάτι άλλο από το κυρίως πρόγραμμα, αφού ο scheduler τα κάνει όλα αυτόματα στο background, επιτρέποντας την πρόσβαση σε μόνο ένα thread τη φορά.

Για τη συγκεκριμένη εφαρμογή που περιγράφεται, το γενικό μπλοκ των threads φαίνεται στην εικόνα 64.



Εικόνα 64 - Μπλοκ διάγραμμα των thread του προγράμματος

Το ένα από τα δύο threads είναι αυτό της διαχείρισης του αισθητήρα, το οποίο επωμίζεται όλη τη διαχείριση της εφαρμογής, πλην της διασύνδεσης με την ιστοσελίδα, την οποία διαχειρίζεται ο server.

10.1.1 Εργασίες thread διαχείρισης αισθητήρα

Το thread διαχείρισης αισθητήρα:

- Στέλνει τα δεδομένα στο δίκτυο RS-485 και συλλέγει τις απαντήσεις από τους αισθητήρες.
- Ενημερώνει τις μετρήσεις και την κατάσταση των αισθητήρων (σφάλματα – συναγερμοί αισθητήρα).
- Ενημερώνει την κατάσταση του συστήματος (ηρεμία – σφάλμα – συναγερμός).
- Αποθηκεύει ανά τακτά χρονικά διαστήματα τα δεδομένα στο μέσο αποθήκευσης.
- Ελέγχει αν ο χρήστης έχει στείλει εντολές από την ιστοσελίδα. Οι εντολές έχουν σωθεί σε συγκεκριμένες μεταβλητές από το thread του server.

10.1.2 Εργασίες thread τοπικού διακομιστή (server)

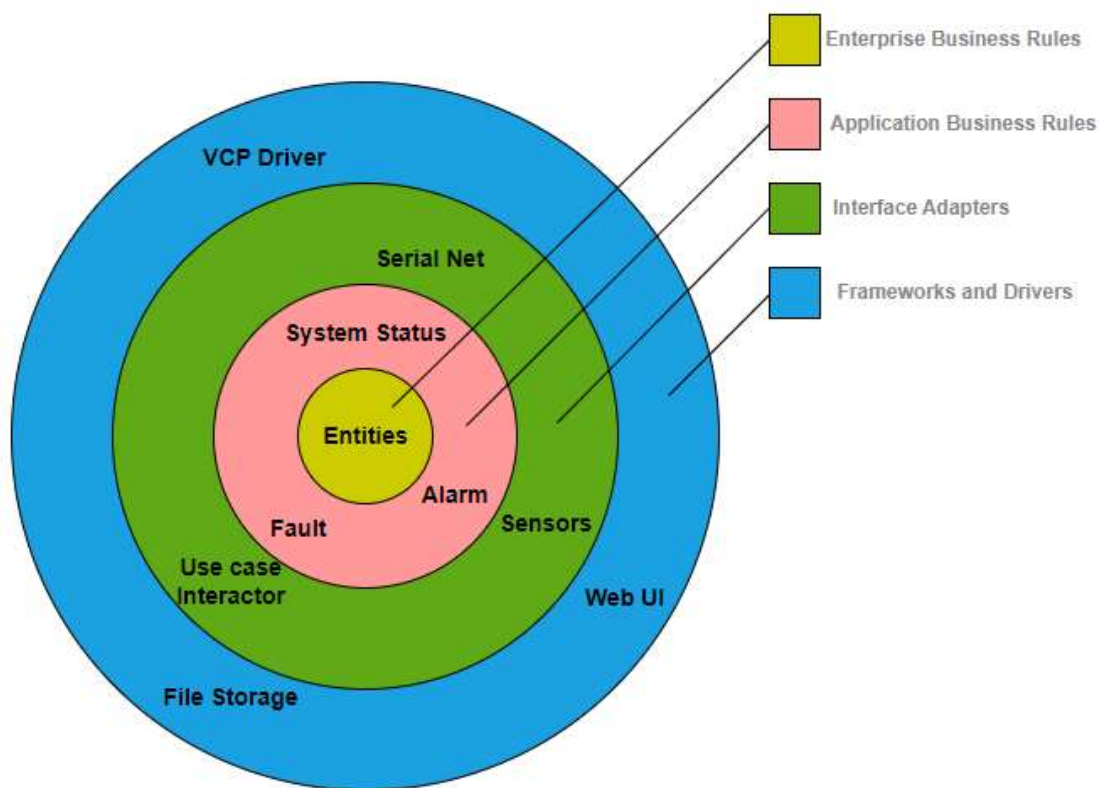
Το thread του server:

- Λαμβάνει το request από την ιστοσελίδα και διαβάζει τις εντολές χρήστη, τις οποίες σώζει σε συγκεκριμένες μεταβλητές, ώστε να είναι διαθέσιμες προς ανάγνωση από το thread διαχείρισης αισθητήρα.

- Συλλέγει τα δεδομένα προς αποστολή στην ιστοσελίδα από το μέσο αποθήκευσης (τα οποία δεδομένα έχουν σωθεί προηγουμένως από το thread διαχείρισης αισθητήρα).
- Διαμορφώνει κατάλληλα τα δεδομένα αυτά, ώστε να σταλούν στην ιστοσελίδα.
- Προσθέτει πληροφορίες για τα χρώματα (πχ εάν το σύστημα είναι σε συναγερμό, η κατάστασή του θα εμφανίζεται με κόκκινο χρώμα).
- Στέλνει τις παραπάνω πληροφορίες στην ιστοσελίδα.

10.2 Thread διαχείρισης αισθητήρα

Η οργάνωση του συγκεκριμένου thread, η γενικότερη αρχιτεκτονική και η κατάτμηση των software modules έγινε με βάση το Clean Architecture (εικόνα 65).



Εικόνα 65 - Διαστρωμάτωση προγράμματος ΚΕΣ με βάση το Clean Architecture [25]

Σύμφωνα με το Clean Architecture δεν επιτρέπεται η επικοινωνία κάποιου επιπέδου με κάποιο άλλο, εάν δεν είναι το αμέσως ανώτερο ή το αμέσως κατώτερο. Επίσης τα επίπεδα μεταξύ τους ανταλλάσσουν απλές μορφές δεδομένων, στο επίπεδο της γλώσσας προγραμματισμού και όχι στο επίπεδο της υλοποίησης. Δεν επιτρέπεται για παράδειγμα να περάσουν προς τα ανώτερα στρώματα απευθείας database queries, όσο «δεδειγμένο» και γρήγορο και αν ακούγεται αυτό. Θα πρέπει τα queries αυτά να μετασχηματιστούν σε απλούς τύπους δεδομένων (πχ αντικείμενα, string, απλές μεταβλητές κτλ.) και στη συνέχεια να περάσουν προς τα υψηλότερα στρώματα. Με αυτόν τον τρόπο εξασφαλίζουμε την προσέγγιση του «αγνωστικισμού» (agnostic approach), όπου τα υψηλότερα στρώματα δεν

επηρεάζονται ποτέ από τις αλλαγές που συμβαίνουν στα χαμηλότερα. Πράγματι, αν για παράδειγμα αλλάξει το framework της βάσης δεδομένων για τον οποιοδήποτε λόγο (πχ διότι η εταιρία αποφάσισε να πάει σε κάποιο άλλο που προσφέρεται δωρεάν), τότε αν τα queries περνάνε απευθείας προς τα «πάνω», θα έπρεπε να συμβούν αλλαγές και στα υψηλότερα επίπεδα για να υποστηριχτεί κάτι τέτοιο. Οπότε θα υπήρχε το παράδοξο οι κανόνες τις επιχείρησης ή της εφαρμογής (πχ αλγόριθμοι εξαγωγής συναγερμών και σφαλμάτων), να εξαρτιόνται από τις λεπτομέρειες της υλοποίησης, όπως το τι είδους framework χρησιμοποιείται για τη βάση δεδομένων.

Στο «κέντρο» της αρχιτεκτονικής και στο υψηλότερο επίπεδο τοποθετούνται τα entities (εικόνα 65), τα κομμάτια δηλαδή που ανήκουν σε αυτό που ονομάζεται «κανόνες επιχείρησης». Έχει να κάνει κυρίως με τα δεδομένα των αισθητήρων, όπως τις μετρήσεις και τις ρυθμίσεις χρήστη γύρω από αυτούς. Με τον όρο «κανόνες επιχείρησης» εννοείται πολύ απλά αυτό που «πουλάει» η επιχείρηση στους πελάτες της και στη συγκεκριμένη περίπτωση είναι ένα σύστημα ασφαλείας, το οποίο χωρίς τα δεδομένα των αισθητήρων δεν υφίσταται καν ως προϊόν. Άρα τα entities είναι κυρίως αποθηκευμένα δεδομένα (data).

Στο δεύτερο επίπεδο της εφαρμογής συναντιόνται οι απαιτήσεις του συστήματος, οι οποίες όμως έχουν να κάνουν περισσότερο με τους πελάτες και όχι με την επιχείρηση. Σε κάποιο σύστημα για παράδειγμα, ο χρήστης δεν μπορεί να απενεργοποιήσει κάποιον αισθητήρα ώστε να μη δίνει συναγερμό. Σε κάποιο άλλο σύστημα αυτό είναι εφικτό. Όμως και οι δύο αυτές περιπτώσεις σχετίζονται με τους πελάτες και διαμορφώνονται από την αγορά. Αποτελούν περισσότερο «συμπεριφορές» του συστήματος που βρίσκονται μεν στο υψηλό επίπεδο της εφαρμογής, αλλά όχι στο επίκεντρο της επιχείρησης, όπως τα entities. Στο επίπεδο της εφαρμογής ανήκει η κατάσταση του συστήματος, πότε αυτό δίνει συναγερμό, πότε σφάλμα και πότε είναι σε κατάσταση ηρεμίας. Σε αυτό το επίπεδο ανήκει και η λογική της διαχείρισης της εφαρμογής, το «application interface», που καθορίζει πώς καλούνται οι συναρτήσεις των modules, με τι ορίσματα και τι τιμές επιστρέφουν.

Στο τρίτο επίπεδο υπάρχουν αυτά που ονομάζονται «adapters». Τα λογικώς διαχωρισμένα κομμάτια της εφαρμογής δηλαδή ή αλλιώς οι θεματικές ενότητες που, στο σύνολό τους, συνθέτουν τη λειτουργία του συστήματος. Στην παρούσα εργασία οι adapters αναφέρονται συνήθως ως «modules» και είναι αυτά του αισθητήρα, του πρωτοκόλλου για το δίκτυο RS-485, της αποθήκευσης δεδομένων και του user interface, το οποίο διασυνδέει με τον τοπικό server και στη συνέχεια με την ιστοσελίδα. Υπάρχει και το logging module, το οποίο αναλύεται ως ξεχωριστή περίπτωση στην παράγραφο 10.4.

Το τέταρτο και τελευταίο επίπεδο είναι αυτό που ονομάζεται «Frameworks and Drivers». Έχει να κάνει με τις λεπτομέρειες υλοποίησης και συχνά περιλαμβάνει ειδικές βιβλιοθήκες με ειδικές συναρτήσεις, πχ για βάσεις δεδομένων ή http requests. Στην προκειμένη περίπτωση του GreenLink Pro η «ειδική» αυτή βιβλιοθήκη είναι το Flask, το οποίο είναι το framework που διεκπεραιώνει την επικοινωνία http. Τα κομμάτια αυτά βρίσκονται στο χαμηλότερο επίπεδο της εφαρμογής γιατί πολύ απλά δεν ενδιαφέρει κανέναν η υλοποίησή τους πέρα από τους μηχανικούς ανάπτυξης. Το αν αντί για το Flask χρησιμοποιούνταν για παράδειγμα το «Django», ουδόλως ενδιαφέρει τον πελάτη ή την επιχείρηση, αρκεί να πληρούνται όλα τα requirements των υψηλών επιπέδων. Τα framework και οι drivers θα πρέπει να ανακατασκευάζονται (ή ακόμα και να αντικαθίστανται εξολοκλήρου από άλλα framework),

χωρίς να χρειάζεται να αλλάξει ούτε μια γραμμή κώδικα στα υψηλότερα επίπεδα. Οι άλλοι δύο drivers που ανήκουν σε αυτό το επίπεδο είναι ο driver της σειριακής επικοινωνίας (USB) και οι συναρτήσεις προσπέλασης στο αποθηκευτικό μέσο.

10.2.1 Application interface

Όπως αναφέρθηκε στην προηγούμενη παράγραφο, τα modules είναι αυτά που επιτελούν τις βασικές λειτουργίες του συστήματος (επικοινωνία με αισθητήρες, ενημέρωση κατάστασης συστήματος κτλ) και τρέχουν εντός του thread διαχείρισης αισθητήρα.

Το thread διαχείρισης αισθητήρα χρησιμοποιεί ατέρμονα βρόχο (infinite loop). Καλεί τα modules ένα προς ένα (μέσω interfaces), τους περνάει τα ορίσματα που πρέπει και τις επιστρεφόμενες τιμές τις περνάει ως ορίσματα σε άλλα modules. Χωρίζεται σε δύο επίπεδα. Το υψηλότερο από αυτά απλώς κάνει τις κλήσεις χωρίς ορίσματα και επιστρεφόμενες τιμές, σε μια «καθαρή» μορφή ώστε να είναι εύκολα αναγνώσιμες από τον χρήστη (εικόνα 66). Ο βρόχος κατά σειρά εξυπηρετεί το δίκτυο RS-485, ανανεώνει τις μετρήσεις και την κατάσταση των αισθητήρων, ανανεώνει την κατάσταση του συστήματος, ανανεώνει τα entities και τέλος αποθηκεύει τα νέα δεδομένα.

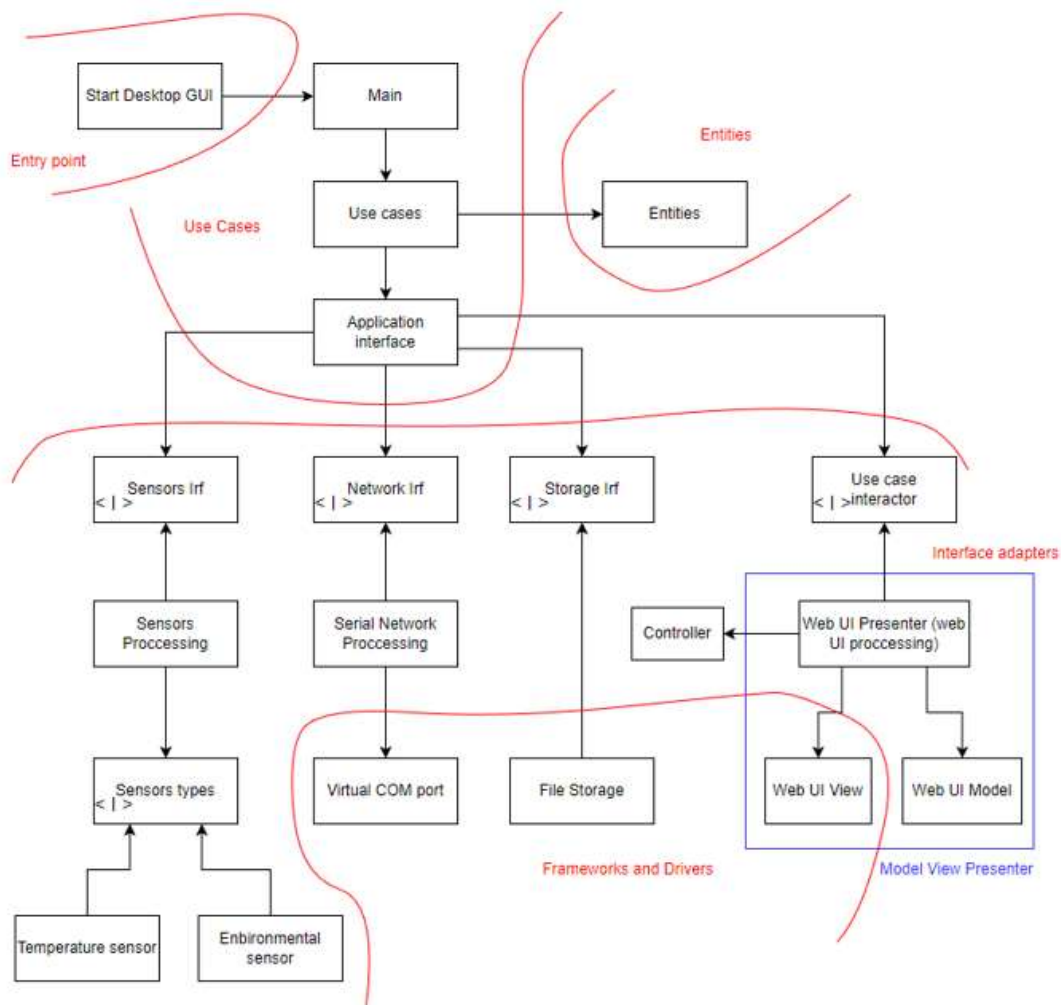
```
def run(self):
    """The tasks run in sequence.
    """
    self.app.net_request()
    self.app.sensor_update()
    self.update_app_status()
    self.update_entities()
    self.app.storage_access()
```

Εικόνα 66 - Ατέρμονας βρόχος της διαχείρισης αισθητήρα (app.py)

Οι περισσότερες κλήσεις από το αρχείο *app.py* (εικόνα 66) γίνονται προς το αρχείο *appif.py*, όπου προστίθενται τα ορίσματα για τις κλήσεις στα χαμηλότερα επίπεδα και παραλαμβάνονται οι επιστρεφόμενες τιμές από τις κλήσεις αυτές. Οι κλήσεις που δεν είναι προς το *appif.py*, αφού πρόκειται για κλήσεις υψηλού επιπέδου) είναι αυτές για την ανανέωση της κατάστασης του συστήματος και οι κλήσεις προς τα entities.

10.2.2 Module interface

Για να διατηρηθεί η επιθυμητή «καθαρότητα» του προγράμματος και σύμφωνα με την αρχιτεκτονική της εικόνας 67, θα πρέπει κατ' ελάχιστον τα layer των adapters (modules) να συνδέονται μέσω interfaces με το υψηλότερο επίπεδο και όπου χρειάζεται και με το χαμηλότερο. Με αυτόν τον τρόπο το κάθε module «κουμπώνει» στο application ως plug-in, για να διατηρήσει την ανεξαρτησία του application από αυτό.



Εικόνα 67 –GreenLink Pro components ανά layer προγράμματος, βάσει Clean Architecture [25]

Όπως καθίσταται σαφές από τις αρχιτεκτονικές γραμμές (με κόκκινο) της εικόνας 67, το application interface δεν καλεί απευθείας τα modules, αλλά μέσω του module interface (αρχείο *modif.py*). Το interface είναι στην ουσία ένα «συμβόλαιο» που υπογράφουν δύο layers μεταξύ τους. Το πιο κάτω layer (τα modules στην προκειμένη περίπτωση), δίνουν μια «υπόσχεση» στο application ότι κάποιες συγκεκριμένες συναρτήσεις γίνονται ορατές προς αυτό και όλη η υπόλοιπη υλοποίηση θα παραμείνει άγνωστη προς το application. Αξιοσημείωτο είναι ότι το sensor module είναι το μοναδικό που διαθέτει interface τόσο προς τα ανώτερα επίπεδα (module interface) αλλά και προς τα κατώτερα (component interface). Με αυτόν τον τρόπο ο κάθε αισθητήρας «κουμπώνει» ως plug-in, αποδεσμεύοντας το sensor module από την υποχρέωση να αλλάζει ο κώδικάς του, κάθε φορά που προστίθεται ή αφαιρείται ή μεταβάλλεται κάποιος αισθητήρας.

Για να γίνει πιο κατανοητός ο ρόλος του module interface, παρατίθεται το παράδειγμα στην εικόνα 68, όπου φαίνεται το interface του storage module. Η ρουτίνα αρχικοποίησης `__init__()` είναι ο constructor του interface. Οι δύο συναρτήσεις του interface είναι η `write()` και η `read()`. Η πρώτη παίρνει ως όρισμα τα entities και η δεύτερη επιστρέφει τα entities. Αυτές είναι και οι μοναδικές πληροφορίες που γνωρίζει το application interface για το storage

module. Δεν γνωρίζει τις λεπτομέρειες υλοποίησης, όπως που αποθηκεύει τα δεδομένα και σε ποια μορφή. Θα μπορούσε να είναι στο σκληρό δίσκο, σε απομακρυσμένο server, σε αρχείο *.json*, σε βάση δεδομένων κτλ. Με αυτόν τον τρόπο εξασφαλίζεται το γεγονός πως οποιεσδήποτε αλλαγές συμβούν εντός του module δεν θα επηρεάσουν ούτε μια γραμμή κώδικα στα υψηλότερα επίπεδα, αρκεί το module να τηρεί πάντα την «υπόσχεσή» του, ότι δηλαδή θα έχει δύο συναρτήσεις, την *write()* που θα παίρνει ως όρισμα τα entities και την *read()* που θα επιστρέφει τα entities.

```
52 class StorageIf:
53     def __init__(self, store_factory, storage_path):
54         """Initialize the StorageIf class with a storage factory and other arguments if needed.
55         """
56         self.store = store_factory(storage_path)
57
58     def write(self, entities):
59         """Writes entities to storage.
60         """
61         return self.store.write(entities)
62
63     def read(self):
64         """Reads entities from storage.
65         """
66         return self.store.read()
```

Εικόνα 68 - Storage module interface (modif.py)

10.2.3 Module δικτύου RS-485

Όπως έχει αναφερθεί στο κεφάλαιο 5, το module χωρίζεται σε δύο επίπεδα. Το υψηλότερο επίπεδο είναι το L2, το οποίο διαχειρίζεται το payload (μετρήσεις αισθητήρων). Το χαμηλό επίπεδο είναι το L1, το οποίο αναλαμβάνει τις λειτουργίες του πρωτοκόλλου, όπως την επιβεβαίωση της ορθότητας των εισερχομένων πακέτων, την προετοιμασία των πακέτων προς αποστολή (encapsulation) κτλ. Τα αρχεία για τα δύο αυτά επίπεδα είναι αντίστοιχα τα *l2.py* και *l1.py*. Το interface του συγκεκριμένου module με τα ανώτερα επίπεδα είναι η συνάρτηση «*net_request()*» (εικόνα 69). Οι λειτουργίες γραμμή προς γραμμή είναι οι εξής:

- Ανανεώνει τη λίστα με τους slave όπως τους παραλαμβάνει από το ανώτερο επίπεδο και ελέγχει την ορθότητά της.
- Ετοιμάζει το payload προς αποστολή για τους slave, που περιέχει τις πληροφορίες της εφαρμογής (πχ κατάσταση συστήματος).
- Στέλνει το payload προς το L1, το οποίο θα το διαμορφώσει κατάλληλα, θα το στείλει στο δίκτυο, θα λάβει απάντηση, θα εξετάσει το εισερχόμενο πακέτο και θα επιστρέψει στο L2 το payload και τη διεύθυνση του slave με τον οποίο επικοινωνήσε.
- Θα ανανεώσει την κατάσταση σφάλματος των κόμβων, αφού σε περίπτωση που κάποιος κόμβος δεν απαντήσει για 60 δευτερόλεπτα στις κλήσεις του master, η κατάσταση του ενημερώνεται ως «σφάλμα».
- Δημιουργεί τα entities που αφορούν το module του δικτύου, ώστε να είναι διαθέσιμες προς ανάγνωση από τα ανώτερα στρώματα προγράμματος.
- Επιστρέφει προς τα ανώτερα στρώματα του προγράμματος το εισερχόμενο payload και τη διεύθυνση του slave που το έστειλε. Σε περίπτωση που ο slave δεν απάντησε,

τότε επιστρέφεται κενό πακέτο προς τα ανώτερα στρώματα, τα οποία θα διαχειριστούν αυτήν την πληροφορία με όποιον τρόπο οφείλουν.

```
def net_request(self, status: Status, nodes_list: list, command: int) -> Dict[str, int]:
    """Implementation of interface.
    Send a request to the communication network. Returns A dictionary containing the parsed L2 payload values.
    `nodes_list` is the list with IDs of the network addresses of nodes.
    """
    self.update_network_from_new_list(nodes_list)

    if len(nodes_list) == 0:
        # empty nodes list will return empty payload and 0 address
        return b'', 0

    tx_payload = self.serialize_l2_to_l1(status.alarm, status.fault, status.app_status.value, command=command)
    rx_payload, slave_address = self._l1.l1_request(tx_payload)
    self._l2_net.nodes_update(rx_payload, slave_address, self._l1.serial_exception)
    self.create_net_entity()
    return self.parse_l1_to_l2(rx_payload), slave_address
```

Εικόνα 69 - Βασικές λειτουργίες του module του δικτύου RS-485 (l2.py)

10.2.4 Module αισθητήρα

Οι βασικές του λειτουργίες στο υψηλότερο επίπεδο βρίσκονται εντός του αρχείου *sensor.py* και φαίνονται στην εικόνα 70.

```
def update(self, sensor_data: dict, sensor_address: int) -> bool:
    """Implementation of interface.
    Update sensors with the latest received values.
    Returns `False` if sensor update fails and `True` otherwise.
    """
    # Get sensor full list from an input (eg a file) to work with it during sensor update.
    sensor_full_list = [SensorDescriptor(sensor_cfg[0], sensor_cfg[1]) for sensor_cfg in SensorIO.parse_sensors(
        SensorIO.csv_get_list_of_sensors(self.sensor_net_path))]

    if self.update_sensor_list(sensor_full_list) is True:
        # new sensor list so `sensor_address` could be obsolete, so skip sensor processing until next cycle.
        return True

    if not self.sensor_full_list:
        return True

    self.sensor_tick_count()

    if self.update_single_sensor(sensor_data, sensor_address) is False:
        logger.log(severity=sev.DEBUG, source=src.SENSOR, event=evt.SENSOR_UPDATE_FAILED, id=sensor_address)
        return False

    self.update_sensors_general_status()
    return True
```

Εικόνα 70 - Βασικές λειτουργίες του module αισθητήρα (sensor.py)

Αποτελεί το μεγαλύτερο module του συστήματος και το μοναδικό με τέτοια πολυπλοκότητα, ώστε να χρειάζεται interface όχι μόνο προς το ανώτερο, αλλά και προς το κατώτατο επίπεδο.

Οι λειτουργίες γραμμή προς γραμμή (εικόνα 70) είναι οι εξής:

- Διάβασμα και έλεγχος ορθότητας της λίστας αισθητήρων του χρήστη (σε περίπτωση που έχει αλλάξει), καθώς και ανανέωση της αντίστοιχης εσωτερικής του λίστας.

- Ανανέωση του χρονομέτρου. Το module αισθητήρα χρησιμοποιεί δική του χρονομέτρηση για κάποιες βασικές του λειτουργίες. Για παράδειγμα εάν οι μετρήσεις κάποιου αισθητήρα είναι στην περιοχή του σφάλματος ή του συναγερμού, το module δεν ενημερώνει απευθείας την κατάσταση του αισθητήρα, παρά το επιβεβαιώνει για 60 και 30 δευτερόλεπτα αντίστοιχα, ώστε να αποφευχθούν ψευδή συμβάντα (debounce mechanism).
- Εφόσον όλοι οι έλεγχοι ορθότητας των τιμών του αισθητήρα είναι επιτυχείς, ανανεώνει τις μετρήσεις του αισθητήρα.

Όπως αναφέρθηκε παραπάνω, ο τρόπος σύνδεσης του sensor module με τα ανώτερα layer (μέσω interface), εφαρμόζεται και με τα κατώτερα layer τα οποία ονομάζονται components και συνεπώς το αντίστοιχο interface ονομάζεται «component interface». Τα components είναι οι διαφορετικές «υλοποιήσεις» του αισθητήρα, στην προκειμένη περίπτωση του περιβαντολλογικού και του αισθητήρα θερμοκρασίας. Η χρήση του component interface δίνει την ευελιξία στο να μπορεί να μεταβληθεί ο κώδικας κάποιου αισθητήρα ή να διαγραφεί εντελώς ο αισθητήρας ή να προστεθεί ένας νέος, χωρίς να επηρεάσει το sensor module. Τα components περιλαμβάνουν ειδικές πληροφορίες για την εκάστοτε υλοποίηση. Για παράδειγμα στον περιβαντολλογικό αισθητήρα που έχει οκτώ διαφορετικές μετρήσεις, περιλαμβάνει για κάθε μία από αυτές πληροφορίες για το ποια είναι η αρχική τιμή της κάθε μέτρησης (default), ποια είναι τα όρια συναγερμού, το εύρος επιτρεπτών τιμών, τα scaling functions κτλ. Οι πληροφορίες αυτές βρίσκονται εντός των αρχείων *sensor_env.py* για το περιβαντολλογικό και *sensor_tmp.py* για τον αισθητήρα θερμοκρασίας.

Από τις πληροφορίες των components ιδιαίτερη αναφορά θα γίνει στον τρόπο διαχείρισης των μετρήσεων. Ο αισθητήρας στέλνει πάντα την «ωμή» (raw) τιμή στο δίκτυο. Η πλακέτα του περιβαντολλογικού αισθητήρα για παράδειγμα θα στείλει την τιμή της θερμοκρασίας όπως τη διάβασε από το I²C interface του SEN55. Στη συνέχεια η τιμή αυτή μέσω των εσωτερικών διεργασιών του sensor module, θα περάσει από χαμηλοπερατό φίλτρο και τέλος στη φιλτραρισμένη αυτή τιμή θα εφαρμοστεί αυτό που ονομάζεται “scale function”, η οποία θα μετατρέψει την τιμή σε κατανοητό προς τον άνθρωπο μέγεθος. Η συνάρτηση αυτή προκύπτει από το datasheet του εκάστοτε αισθητήρα, ενώ στους αναλογικούς αισθητήρες θα πρέπει να ληφθούν υπόψη και τα εξωτερικά κυκλώματα (πχ απολαβή ενίσχυσης του σήματος εισόδου). Στο παράδειγμα του περιβαντολλογικού αισθητήρα, αρκεί να διαιρέσουμε την εισερχόμενη τιμή με το 200 για να μετατρέψουμε σε °C. Συνεπώς θα ισχύει:

$$Value(^{\circ}C) = Value(I^2C) / 200$$

Η παραπάνω είναι η scale function της θερμοκρασίας για τον αισθητήρα SEN55, που σημαίνει ότι αν για παράδειγμα ο αισθητήρας στείλει στο PC την τιμή 4000, αυτό θα μεταφραστεί μέσω της scale function σε 20°C.

10.2.5 Module αποθήκευσης (storage)

Οι δύο βασικές του λειτουργίες (εικόνα 71) είναι η ανάγνωση και η εγγραφή από το αποθηκευτικό μέσο και βρίσκονται εντός του αρχείου *store.py*. Η συνάρτηση *read()* επιστρέφει τα entities όπως ακριβώς είναι αποθηκευμένα. Η ρουτίνα *write()* κάνει με τη σειρά τα εξής:

- Ελέγχει εάν ήρθε η ώρα να αποθηκεύσει (ανά 15 δευτερόλεπτα).
- Γράφει τα entities που πήρε ως όρισμα από τα ψηλότερα επίπεδα σε αρχείο *.json*. Σε περίπτωση σφάλματος εγγραφής, ενημερώνει αναλόγως το σύστημα.

```
def write(self, entities: dict) -> bool:
    """Implementation of interface.
    Write entities to a JSON file every `STORE_INTERVAL_SECONDS`.
    Returns `True` if the write operation took place, or `False` otherwise.
    """
    if not self.is_time_to_store():
        return False

    try:
        # Write entities to a file in JSON format, overwriting any previous contents
        with open(self.storage_path, 'w') as file:
            file.write(json.dumps(entities, indent=4))
    except Exception as e:
        logger.log(severity=sev.ERROR, source=src.STORAGE, event=evt.ENTITIES_WRITE_ERROR)
        e = e
        return False
    return True

def read(self) -> dict:
    """Implementation of interface.
    Read entities from a JSON file.
    Returns a dictionary containing the read entities.
    """
    return self.read_entities(self.storage_path)
```

Εικόνα 71 - Βασικές λειτουργίες του module αποθήκευσης (store.py)

Σημαντικό στοιχείο του module αποθήκευσης, είναι ότι, αν και αρχιτεκτονικά, έχει καταταχθεί στο thread διαχείρισης αισθητήρα, η συνάρτηση *read()* καλείται και από το thread του server, αφού τα entities πρέπει να σταλούν και στην ιστοσελίδα. Αυτό όμως συμβαίνει ασύγχρονα όταν υπάρχει request από την ιστοσελίδα, ενώ το module διαχείρισης αισθητήρα το διαχειρίζεται σύγχρονα και περιοδικά, όπως όλα τα άλλα modules. Συνεπώς δικαιωματικά μπορεί να καταταχθεί στο module διαχείρισης αισθητήρα. Το κομμάτι επικοινωνίας του UI module με το thread του server, περιγράφεται αναλυτικά στην παράγραφο 10.3.

10.2.6 Module διεπαφής χρήστη (User Interface ή UI module)

Πρόκειται για το module που διασυνδέει το thread διαχείρισης αισθητήρα με αυτό του server και βρίσκεται εντός του αρχείου *ui.py*. Στην εικόνα 72 φαίνεται η *rst_get()*, που είναι η interface function του module αυτού. Η λειτουργία της συνάρτησης αυτής είναι να διαβάσει

τις εντολές χρήστη από την ιστοσελίδα και στη συνέχεια να εξετάσει τα δύο παρακάτω ενδεχόμενα:

1. Σε περίπτωση που ο χρήστης έδωσε εντολή για καθάρισμα του log αρχείου και επειδή αυτό ανήκει στο system wide layer, την υλοποιεί εσωτερικά το module.
2. Σε περίπτωση που ο χρήστης έδωσε εντολή για reset συναγερμό και σφαλμάτων, αυτή η λειτουργία είναι πολύ πάνω από την ευθύνη του User Interface module. Το reset επιστρέφεται ως Boolean μεταβλητή στα ανώτερα επίπεδα προγράμματος, ώστε να τη διαχειριστούν κατάλληλα.

```
def rst_get(self) -> bool:
    """Implementation of interface. Get the input values from UI.
    """
    reset = False

    ui_inputs = self.presenter.controller.read_input_arguments()
    if ui_inputs:
        for key, value in ui_inputs.items():
            # Check if input's key and value exists in the common dictionary with the presenter
            if key in Controller.VIEW_INPUTS and value == Controller.VIEW_INPUTS[key]:
                if key == "reset_system":
                    reset = True
                elif key == "clear_log":
                    logger.clear_log_file()
    return reset
```

Εικόνα 72 - Βασικές λειτουργίες του User Interface module (ui.py)

Το κομμάτι επικοινωνίας του UI module με το thread του server, περιγράφεται αναλυτικά στην παράγραφο 10.3.

10.3 Thread τοπικού διακομιστή (Server)

Δομήθηκε σύμφωνα με το μοντέλο Model-View-Presenter ή MVP [25]. Τα διακριτά μέρη του thread του server περιγράφονται παρακάτω.

10.3.1 Presenter

Βρίσκεται εντός του αρχείου *ui_presenter.py*. Ανήκει στο υψηλότερο επίπεδο του server, αφού «κρατάει» τη λογική του thread. Όλοι οι υπολογισμοί, οι ενέργειες ανάλογα με τα δεδομένα που θα παραληφθούν από την ιστοσελίδα ή το thread διαχείρισης αισθητήρων και ότι άλλου είδους αποφάσεις θα χρειαστεί να πάρει ο server, ανήκουν στη λογική του presenter, ακόμα και με τι χρώμα θα εκτυπωθεί στην ιστοσελίδα το κάθε κείμενο.

Η κύρια λειτουργία του είναι να μετατρέπει τα δεδομένα από μία μορφή σε κάποια άλλη. Ο λόγος είναι ότι από την ιστοσελίδα παραλαμβάνεται πληροφορία σε συγκεκριμένη μορφή, η οποία πρέπει να μεταφερθεί προς τη διαχείριση αισθητήρων. Το ίδιο ισχύει και προς την αντίθετη φορά. Με τους κατάλληλους αυτοματισμούς του view που μπορεί να διαχειρίζεται άψογα τις συμβολοσειρές *json*, οι μετατροπές τύπου δεδομένων του presenter είναι σχεδόν

μηδενικές, αφού χρησιμοποιεί python dictionaries που είναι σχεδόν το ίδιο με το *json*. Παρόλα αυτά χρειάζεται να επεξεργαστεί σε μεγάλο βαθμό τα ήδη υπάρχοντα dictionaries πριν φύγουν στην ιστοσελίδα, αφού προσθέτει πληροφορίες για τα χρώματα σε όλα τα κείμενα της ιστοσελίδας, λειτουργικότητα που βρίσκεται εντός του αρχείου *ui_colors.py*.

Ο ρόλος του presenter είναι κομβικός, αφού συνδέει το thread διαχείρισης αισθητήρα με την ιστοσελίδα. Συγκεκριμένα επωμίζεται το request και το response από / προς την ιστοσελίδα (εικόνα 73).

```
def request(self, view_arguments):
    """ Called from model.
    Process incoming requests and provide a response.

    Args:
        view_arguments: Input arguments from the view module.

    Returns:
        Response and flag.
    """
    resp = {}, True

    with self.controller.input_arguments_lock:
        self.controller.input_arguments = self.controller.process_arguments(view_arguments)

        if not view_arguments:
            # if refresh page requested, then the template must be rendered
            self.controller.input_arguments = {}
            entities = self.model.read_entities()
            entities_colored = ColorsUI.add_colors_to_entities(entities)
            resp = entities_colored, True
        else:
            # Iterate over all keys in the request arguments
            for key, value in self.controller.input_arguments.items():
                # Call a function or perform actions based on the key
                if key in Controller.VIEW_INPUTS and value == Controller.VIEW_INPUTS[key]:
                    resp = self.create_response(key)

    return resp
```

Εικόνα 73 - Διαχείριση του http request από τον presenter (*ui_presenter.py*)

Αναλύοντας τον κώδικα, αρχικά εφαρμόζει το lock για να αποθηκεύσει τις εισόδους στον «controller», ο οποίος αποτελεί υπομονάδα του presenter. Στη συνέχεια διακρίνονται δύο περιπτώσεις:

1. Το request προέρχεται από απλή ανανέωση της ιστοσελίδας. Σε αυτήν την περίπτωση ο presenter θα διαβάσει τα entities, στα οποία μετά προσθέτει πληροφορίες για τα χρώματα των κειμένων της ιστοσελίδας. Αφού έχει ετοιμάσει την απάντηση, την επιστρέφει στο view ώστε το τελευταίο να το στείλει μετά στην ιστοσελίδα.
2. Το request προέρχεται από εντολή χρήστη (με πάτημα κουμπιού της ιστοσελίδας). Σε αυτήν την περίπτωση ανάλογα με το τι εντολή δόθηκε, ο presenter μετατρέπει την εντολή σε κατάλληλη μορφή για να διαβαστεί αργότερα από το thread διαχείρισης αισθητήρα και μετά ετοιμάζει το πακέτο απάντησης.

10.3.2 Model

Είναι ουσιαστικά η αποθήκη δεδομένων του server. Προσπέλαση στο model κάνει μόνο ο presenter και κανένας άλλος. Ο μοναδικός κώδικας του model είναι αυτός που διαβάζει τα δεδομένα από το μέσο αποθήκευσης (μέσω του storage module).

10.3.3 View

Ανήκει στο χαμηλότερο επίπεδο του server. Θεωρείται λεπτομέρεια υλοποίησης. Έχει δύο και μοναδικές υποχρεώσεις. Η πρώτη είναι κατά την εκκίνηση του συστήματος, όταν και αρχικοποιεί το thread του server. Το δεύτερο είναι κατά τη λειτουργία του συστήματος, όταν και μεταφέρει τα δεδομένα από την ιστοσελίδα προς τον presenter και αφού ο presenter επεξεργαστεί τα δεδομένα και ετοιμάσει την απάντηση την περνάει πίσω στο view, που με τη σειρά του τη στέλνει στην ιστοσελίδα.

Για την υλοποίηση του view χρησιμοποιήθηκε η βιβλιοθήκη της rython «Flask», η οποία έχει έτοιμους αυτοματισμούς μετατροπής του http πακέτου σε json format και τούμπαλιν. Συνεπώς ο κώδικας της εφαρμογής δεν εμπλέκεται με κατασκευή ή απευθείας ανάγνωση πακέτων http. Αυτά τα διαχειρίζεται το Flask, το οποίο παραλαμβάνει τα http request της ιστοσελίδας και περνάει προς το view μόνο τα ορίσματα (πχ ότι πατήθηκε από τον χρήστη το πλήκτρο «Clear Log»), ενώ από την άλλη δέχεται από το View πακέτα json και τα μετατρέπει σε http responses.

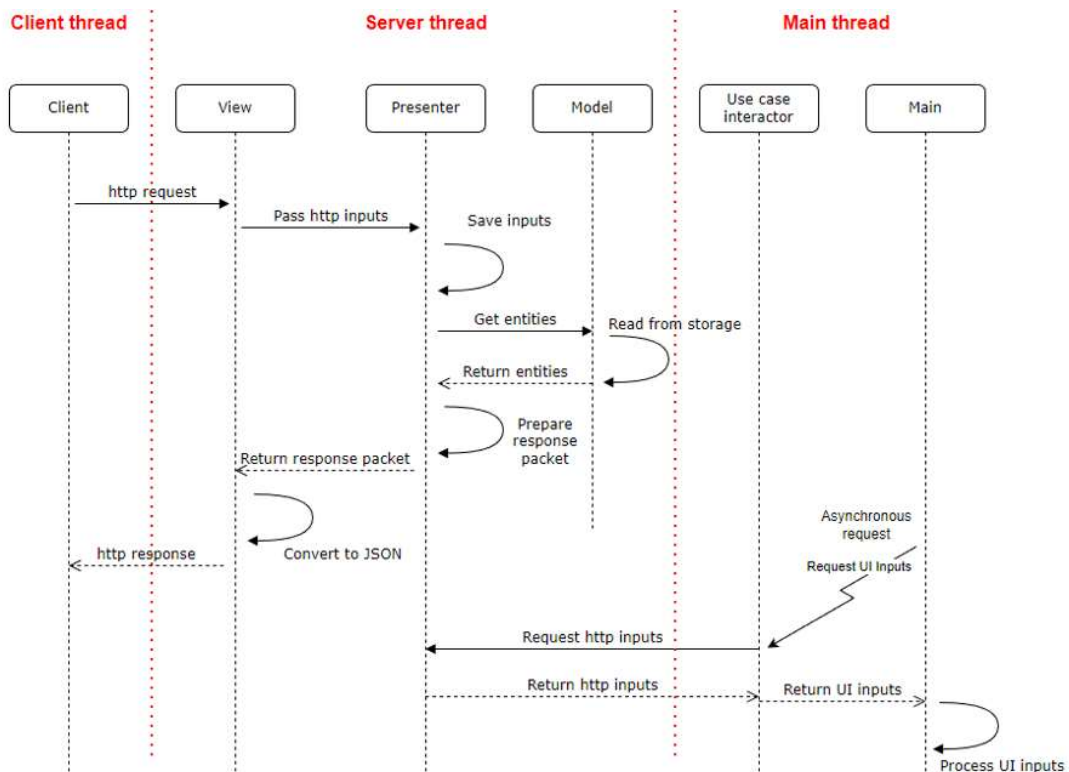
10.3.4 Use case interactor

Αποτελεί τον συνδετικό κρίκο μεταξύ των δύο βασικών thread του συστήματος. Είναι πολύ απλά το UI module, το οποίο έχει αναλυθεί πλήρως στην ενότητα 10.2.6.

10.3.5 Διάγραμμα ακολουθίας server

Φαίνεται στην εικόνα 74. Ξεκινώντας από πάνω αριστερά με το http request και ακολουθώντας τη φορά από τα βέλη, καταλαβαίνουμε ποια είναι η ροή της πληροφορίας, πώς συνδέονται μεταξύ τους τόσο τα δύο threads, όσο και το thread του server με την ιστοσελίδα (client thread), για το οποίο περισσότερες πληροφορίες δίνονται στο κεφάλαιο 11.

Όταν λαμβάνει χώρα το http request από την ιστοσελίδα, το model περνάει απευθείας τις εισόδους (ορίσματα) στον presenter ο οποίος τις σώζει, λαμβάνει τα entities από το model (μέσω του storage module) και στη συνέχεια αφού ετοιμάσει το πακέτο απάντησης το δίνει στο view, που με τη σειρά του το στέλνει στην ιστοσελίδα. Στα δεξιά και κάτω στο διάγραμμα φαίνεται τι συμβαίνει όταν κάποια στιγμή (ασύγχρονη από τη σκοπιά του server) ζητηθούν οι εισοδοί (εντολές χρήστη) από την ιστοσελίδα. Τότε μέσω του use case interactor (UI module), οι εισοδοί ζητούνται από τον presenter ο οποίος, μέσω και πάλι του use case interactor, τις επιστρέφει στη main για να τις επεξεργαστεί.



Εικόνα 74 - Sequence diagram του HTTP request

10.4 Logging module

Το log μπορεί να καταχωρείται σε δύο διαφορετικά σημεία. Σε εξωτερικό αρχείο με επέκταση `.log` ή στην κονσόλα. Οι δύο αυτές περιπτώσεις θα μπορούσαν να τρέχουν είτε μόνο η μία ξεχωριστά είτε μόνο η άλλη είτε και οι δύο μαζί. Στη συγκεκριμένη εφαρμογή τρέχουν και οι δύο μαζί. Επίσης μπορεί να ρυθμιστεί και για τους δύο τύπους καταχωρήσεων η κρισιμότητα (severity) των συμβάντων, πάνω από την οποία θα καταχωρήσουν. Στην περίπτωση της συγκεκριμένης εφαρμογής καταχωρούνται συμβάντα από severity «INFO» και πάνω, που σημαίνει ότι τα πιο χαμηλής κρισιμότητας συμβάντα τύπου «DEBUG» αγνοούνται από το module και δεν καταχωρούνται, άσχετα αν τα ψηλότερα επίπεδα προσπαθούν να καταχωρήσουν «DEBUG» συμβάντα.

```

def log(self, severity: LogSeverity = LogSeverity.INFO, source: LogSource = LogSource.APP,
        event: LogEvent = LogEvent.NO_EVENT, id=None, typ: LogType = None) -> None:
    """Log a message with the specified severity, source, event, id, and type.
    """
    # Don't continue if log severity is smaller than the configured. Otherwise log's SN will jump to larger values
    if severity.value < self.logger.level:
        return

    with self.lock:
        # Convert to proper logging type
        if typ is not None:
            tp = next((key for key, value in logtypes.items() if value == typ), None)
            if tp is None:
                tp = LogType.INVALID_TYPE
        else:
            tp = None
  
```

Εικόνα 75 - Η βασική συνάρτηση «log()» του logging module (log.py)

Το logging module (*log.py*) ακολουθεί το Singleton pattern, σύμφωνα με το οποίο υπάρχει ένα και μόνο αντικείμενο (Singleton object) το οποίο είναι προσπελάσιμο από όλα τα αρχεία του προγράμματος. Αυτό ακούγεται απλό (και είναι) στην περίπτωση που υπάρχει μόνο ένα thread. Στην περίπτωση περισσότερων thread πρέπει να ληφθεί μέριμνα, ώστε να μην προσπελαστεί ταυτόχρονα από περισσότερα του ενός thread. Αυτό επιτυγχάνεται με τη χρήση αυτού που συχνά αποκαλείται «mutex» (mutual exclusive) και στην ρυθμόν επιτυγχάνεται με τη βιβλιοθήκη «lock». Χρησιμοποιώντας αυτήν τη βιβλιοθήκη μπορεί το εκάστοτε thread που χρησιμοποιεί το Singleton object να το «κλειδώσει» ώστε όσο είναι σε χρήση, να μην μπορεί άλλο thread να το χρησιμοποιήσει. Η συνάρτηση που καλείται για να καταχωρηθεί κάποιο συμβάν είναι η *log()* (εικόνα 75).

10.4.1 Ορίσματα

Τα ορίσματα του logging module κατά την καταχώρηση των συμβάντων είναι τα εξής:

- **Severity** (κρίσιμότητα): Ένα σφάλμα που αποκαταστάθηκε δεν μπορεί να είναι το ίδιο κρίσιμο με έναν συναγερμό που μόλις ανιχνεύτηκε. Στην πρώτη περίπτωση υπάρχει ένα συμβάν πληροφορίας (INFO), ενώ στη δεύτερη ένα συμβάν κρίσιμο (CRITICAL).
- **Source** (πηγή): Ένα συμβάν θα μπορούσε να έρθει από διάφορες πηγές (πχ αισθητήρας, server κτλ).
- **Event** (συμβάν): Το συμβάν αυτό κάθε αυτό. Η αποτυχία εγγραφής στη μνήμη ή ο συναγερμός που έδωσε ένας αισθητήρας αποτελούν από μόνα του «συμβάντα».
- **Id**: Χρησιμοποιείται μόνο σε περίπτωση που η πηγή διαθέτει id. Για παράδειγμα το application δεν διαθέτει id (δεν υπάρχει application 1 και application 2). Ο αισθητήρας αντίθετα διαθέτει (αισθητήρας 1, αισθητήρας 2 κοκ).
- **Type** (τύπος): Παρόμοια περίπτωση με το id, από τη σκοπιά ότι τύπο διαθέτουν μόνο οι αισθητήρες (περιβαντολογικός ή θερμοκρασίας).

Στην εικόνα 76 φαίνεται ένα παράδειγμα καταχώρησης συμβάντων από το sensor module. Πρόκειται για το μοναδικό “CRITICAL” συμβάν και είναι αυτό του συναγερμού. Όπως φαίνεται τα ορίσματα που καταχωρούνται είναι:

Κρίσιμότητα συμβάντος: Κρίσιμη

Πηγή: Αισθητήρας

Συμβάν: Συναγερμός αισθητήρα

Id: Το id του αισθητήρα (προέρχεται από δυναμική επεξεργασία δεδομένων)

Τύπος: Ο τύπος του αισθητήρα (προέρχεται από δυναμική επεξεργασία δεδομένων)

```
logger.log(severity=sev.CRITICAL, source=src.SENSOR, event=evt.SENSOR_ALARM,
           id=int(self.id), typ=self.type.value)
```

Εικόνα 76 - Καταχώρηση συμβάντος «συναγερμός αισθητήρα»

10.5 Σημείο εισόδου χρήστη (Entry point)

Αποτελεί ειδική περίπτωση του προγράμματος, αφού η εφαρμογή επιτρέπει δύο τρόπους εκκίνησης: από γραφικά (αρχείο *start.py*) και από γραμμή εντολών (*main.py*). Τονίζεται ότι η *main.py* θα τρέξει έτσι κι αλλιώς, αφού είτε θα αποτελέσει το σημείο εισόδου είτε θα κληθεί από το *start.py* μετά την εκκίνηση. Για να επιτευχθεί αυτή η απαίτηση, θα πρέπει να υπάρξει ένα επιπλέον thread, αυτό της γραφικής εκκίνησης. Εξυπακούεται ότι σε περίπτωση εκκίνησης από τη γραμμή εντολών (απευθείας με την *main.py*), το thread εκκίνησης με τα γραφικά δεν θα τρέξει ποτέ.

10.5.1 Εκκίνηση με γραφικά (*start.py*)

Στην εικόνα 61 φαίνεται η αρχική οθόνη εκκίνησης με γραφικά. Όταν ο χρήστης ολοκληρώσει με τις ρυθμίσεις του και πατήσει το πλήκτρο «Start» τρέχει η συνάρτηση «*start_application()*», η οποία αφού κάνει της απαραίτητες αρχικοποιήσεις από τις μεταβλητές χρήστη, τρέχει στο τέλος τις γραμμές κώδικα που φαίνονται στην εικόνα 77, οι οποίες δημιουργούν και εκκινούν ένα νέο thread, το οποίο τρέχει στη συνάρτηση *main loop* (εντός του *main.py*). Αυτό το νέο thread είναι αυτό που έως τώρα έχει περιγραφεί ως «thread διαχείρισης αισθητήρων». Τα ορίσματα στην περίπτωση αυτή περνιούνται από το *start.py* στο *main.py* (πχ σειριακή θύρα, αρχείο αποθήκευσης συμβάντων κοκ).

```
# Start the main loop in a new thread
thread = Thread(target=self.main_instance.main_loop)
thread.start()
```

Εικόνα 77 – Εκκίνηση *main thread* από το thread της γραφικής εκκίνησης (*start.py*)

Το thread εκκίνησης γραφικών παραδίδει τον έλεγχο στο *main loop* και δεν θα τρέξει ξανά, παρά μόνο όταν ο χρήστης κλείσει το παράθυρο της εφαρμογής. Όταν συμβεί αυτό, θα τρέξει η συνάρτηση «*stop_loop_and_exit()*» εντός του *start.py*, η οποία θα τερματίσει το *main thread* (που με τη σειρά του θα τερματίσει το *server thread*) και στη συνέχεια θα τερματίσει τον ίδιο του τον εαυτό, ώστε να κλείσει το παράθυρο της εφαρμογής.

10.5.2 *main.py*

Το *main.py* μπορεί να είναι το entry point (εκκίνηση με γραμμή εντολών) ή να είναι το thread που δημιούργησε το thread της εκκίνησης γραφικών. Σε κάθε περίπτωση είναι επωμισμένο με τις ίδιες εργασίες:

- Αρχικοποιεί το σύστημα και δημιουργεί το thread του server.
- Τρέχει τον ατέρμονα βρόχο του συστήματος.
- «Ακούει» το πληκτρολόγιο, ώστε να ανιχνεύσει τερματισμό του προγράμματος από το πληκτρολόγιο (με Ctrl + C).
- Τερματίζει το thread του server σε περίπτωση που δόθηκε εντολή τερματισμού από τον χρήστη (είτε από το πληκτρολόγιο είτε από το παράθυρο των γραφικών).

Για την αρχικοποίηση του συστήματος θα χρειαστούν κάποια ορίσματα που περνάει ο χρήστης, είτε με τη χρήση γραφικών είτε από το command line. Είναι ακριβώς τα ίδια ορίσματα, απλά αλλάζει ο τρόπος εισόδου τους προς το σύστημα. Με τα γραφικά ο τρόπος είναι απλός και ευθύς. Στο command line πρέπει να δοθεί το χαρακτηριστικό γράμμα που χρησιμοποιείται για το κάθε όρισμα, συνοδευόμενο από μία τιμή. Όλα τα ορίσματα περιγράφονται πλήρως στο αρχείο README.md, στις ενότητες «Running from the Command Line» και «Default values». Αυτά τα ορίσματα τα χρησιμοποιεί ο constructor της main και τα περνάει μέσω της κλάσης «ApplicationInputs» στο module αρχικοποίησης (*init.py*). Εκεί γίνονται όλες οι αρχικοποιήσεις του συστήματος που, έχουν κατά συντριπτική πλειοψηφία, να κάνουν με τη δημιουργία και αρχικοποίηση όλων των αντικειμένων του συστήματος.

Ο ατέρμονας βρόχος του συστήματος καλεί δύο συναρτήσεις (εικόνα 78). Η μία είναι εντός του *app.py* και τρέχει όλα τα modules (εικόνα 66) και η άλλη είναι μια χρονική καθυστέρηση, ώστε ο βρόχος να τρέχει συγχρονισμένα ανά 70 millisecond.

```
def main_loop(self):
    """Main loop of the program.
    """
    while not self.exit_flag:
        self.app.run()
        self.sleep_task()
```

Εικόνα 78 - Ο ατέρμονας βρόχος του συστήματος

Η τελευταία βασική εργασία με την οποία το main task είναι επωμισμένο, είναι αυτό που ονομάζεται «graceful termination», να τερματίσει η εφαρμογή δηλαδή ομαλά χωρίς να μείνουν ανοιχτά thread. Αυτό το επιτυγχάνει με τη χρήση της συνάρτησης «terminate_application» (εικόνα 79). Στην περίπτωση που η εκκίνηση έγινε από γραμμή εντολών, η ρουτίνα αυτή θα τρέξει και ακριβώς στο τέλος της θα τερματιστεί και η εφαρμογή. Στην περίπτωση της εκκίνησης από γραφικά, με το τέλος αυτής της συνάρτησης θα τερματιστεί και πάλι το thread της διαχείρισης αισθητήρα και το thread του server, αλλά η ροή του προγράμματος θα επιστρέψει για μία τελευταία φορά στο *start.py*, το οποίο θα τερματίσει τον ίδιο του τον εαυτό, ώστε να τερματίσει και συνολικά η εφαρμογή με το κλείσιμο του γραφικού παραθύρου.

```

def terminate_application(self, signum=None, frame=None):
    """Handles the SIGINT signal (Ctrl+C). Stops the main loop (and effectively terminates the whole application).

    Parameters:
        signum (int): The signal number.
        frame (frame): The current stack frame at the time of the signal.
    """
    self.exit_flag = True
    os.kill(os.getpid(), signal.SIGINT)

```

Εικόνα 79 - Η συνάρτηση τερματισμού του main thread

10.6 Σύστημα οργάνωσης αρχείων (repository)

Μετά την εγκατάσταση του προγράμματος, τα αρχεία είναι οργανωμένα όπως φαίνεται στην εικόνα 80.

Όνομα	Ημερομηνία τροποποι...	Τύπος	Μέγεθος
._pycache_	28/12/2023 5:42 μμ	Φάκελος αρχείων	
._setup	10/12/2023 10:07 πμ	Φάκελος αρχείων	
.app	19/12/2023 1:36 πμ	Φάκελος αρχείων	
.artifacts	15/11/2023 8:40 μμ	Φάκελος αρχείων	
.comm	28/12/2023 12:34 μμ	Φάκελος αρχείων	
.crc16	29/11/2023 1:24 πμ	Φάκελος αρχείων	
.entities	20/11/2023 2:23 πμ	Φάκελος αρχείων	
.init	18/12/2023 12:14 πμ	Φάκελος αρχείων	
.log	18/12/2023 12:15 πμ	Φάκελος αρχείων	
.modif	10/12/2023 9:54 πμ	Φάκελος αρχείων	
.operate	18/12/2023 12:28 πμ	Φάκελος αρχείων	
.sensors	19/12/2023 1:36 πμ	Φάκελος αρχείων	
.storage	19/12/2023 12:32 πμ	Φάκελος αρχείων	
.ui	19/12/2023 1:36 πμ	Φάκελος αρχείων	
.venv	29/12/2023 9:41 πμ	Φάκελος αρχείων	
._init_	21/5/2023 1:15 πμ	Python File	0 KB
CHANGELOG	19/12/2023 12:39 πμ	Αρχείο	2 KB
GreekLink_Pro	29/11/2023 1:24 πμ	Αρχείο δέσμης Wi...	1 KB
LICENSE	29/11/2023 1:24 πμ	Αρχείο	2 KB
main	28/12/2023 5:42 μμ	Python File	3 KB
README.md	10/12/2023 9:59 πμ	Αρχείο MD	3 KB
start	13/2/2024 12:06 πμ	Python File	7 KB

Εικόνα 80 - Οργάνωση αρχείων του project (repository)

Το σύστημα οργάνωσης των αρχείων ακολουθεί σε γενικές γραμμές τον διαχωρισμό σε modules (πχ το sensor περιέχει τα αρχεία του sensor module). Από εκεί και πέρα ο φάκελος «artifacts» περιέχει τα αρχεία των συμβάντων, του δικτύου αισθητήρων και των entities. Είναι ο προεπιλεγμένος φάκελος που χρησιμοποιείται και ο χρήστης μπορεί αν επιθυμεί να επιλέξει από τα ορίσματα εκκίνησης κάποιον άλλον φάκελο. Το “CHANGELOG” περιέχει όλες τις αλλαγές που λαμβάνουν χώρα από release σε release, ενώ το LICENSE επικυρώνει το γεγονός ότι αυτό είναι ένα «ανοιχτό» πρόγραμμα, ώστε να υπάρχει ξεκάθαρο καθεστώς πριν την παράδοση του συγκεκριμένου λογισμικού.

11 Ιστοσελίδα (Web page)

Η ιστοσελίδα είναι, πέρα από τα τυπικά (σχεδιασμός και ανανέωση των δυναμικών δεδομένων), επιφορτισμένη με σχεδόν μηδενική λογική, αφού η κύρια υλοποίηση λαμβάνει

χώρα στον server. Διακρίνεται σε τρία βασικά κομμάτια τα οποία είναι το σχεδιαστικό κομμάτι (σχήματα και χρώματα), το στυλ των κειμένων (γραμματοσειρά) και η λειτουργικότητα της σελίδας (πχ ανανέωση των δυναμικών δεδομένων όταν έρχονται από τον server).

11.1 Επεξήγηση λειτουργίας

Στιγμιότυπο από τη λειτουργία της ιστοσελίδας, σε κατάσταση ηρεμίας του συστήματος, φαίνεται στην εικόνα 81.

The screenshot shows the GreenLink Pro dashboard with the following components:

- System Status:** Alarm (false), Fault (false), Status (STATUS IDLE).
- Sensors Status:** Sensors Number (3), Alarm General (false), Fault General (false), Alarms Number (0), Faults Number (0).
- Network Status:** Fault Status (false), Faults Number (0), Serial Port (COM3).
- Sensors in Alarm:** List (No sensors in alarm).
- Sensors in Fault:** List (No sensors in fault).
- Nodes in Fault:** List (No nodes in fault).
- Sensors Table:**

ID	Type	Sensor Alarm	Sensor Fault	pm1p0 (ug/m ³)	pm2p5 (ug/m ³)	pm4p0 (ug/m ³)	pm10p0 (ug/m ³)	Temperature (oC)	Humidity (%)	VOC	NOX
1	SENSE_TYPE_TEMPERATURE	false	false	-	-	-	-	25.25	-	-	-
2	SENSE_TYPE_TEMPERATURE	false	false	-	-	-	-	24.6	-	-	-
3	SENSE_TYPE_ENVIRONMENT	false	false	150.1	170.5	176.1	178.8	22.93	47.22	91	1

Εικόνα 81 - Στιγμιότυπο από τη λειτουργία της ιστοσελίδας σε κατάσταση ηρεμίας

Παρατηρούνται κάποια διακριτά μέρη στην ιστοσελίδα:

- Πίνακας αισθητήρων (sensor table): Βρίσκεται στο κάτω μέρος και απεικονίζει όλες τις μετρήσεις όλων των αισθητήρων. Ο αριθμός γραμμών του πίνακα διαμορφώνεται δυναμικά. Υπάρχουν πάντα ακριβώς όσες γραμμές χρειάζονται για τον αριθμό των αισθητήρων που διαθέτει το δίκτυο. Στο συγκεκριμένο παράδειγμα υπάρχουν δύο αισθητήρες θερμοκρασίας (και για αυτό οι υπόλοιπες στήλες μετρήσεων πλην της θερμοκρασίας εμφανίζονται με παύλες) και ένας περιβαλλοντολογικός στην τελευταία γραμμή, με όλες τις στήλες να έχουν μετρήσεις.
- Εντολές χρήστη: Είναι τα γαλάζια κουμπιά στο επάνω μέρος της ιστοσελίδας. Με το «View Log» εμφανίζεται το log αρχείο στην ιστοσελίδα, με το «Clear Log» καθαρίζεται το αρχείο με τα συμβάντα, με το «View All» εμφανίζονται όλα τα entities και με το «Reset All» το σύστημα επανέρχεται «βίαια» σε κατάσταση ηρεμίας, με την υποσημείωση ότι εφόσον το συμβάν είναι ακόμα εκεί (πχ συγκέντρωση μικροσωματιδίων σε κρίσιμο επίπεδο) τότε ο συναγερμός μετά από λίγο θα επανέλθει.
- Containers: Είναι 6 στον αριθμό και περιέχουν πληροφορίες για την κατάσταση του συστήματος (System Status), την κατάσταση των αισθητήρων (Sensor Status), την κατάσταση του δικτύου RS-485 (Network Status), τη λίστα με τους αισθητήρες σε

συναγερμό (Sensor In Alarm), τη λίστα με τους αισθητήρες σε σφάλμα (Sensor In Fault) και τη λίστα με τους κόμβους δικτύου σε σφάλμα (Nodes In Fault).

Στην εικόνα 82 φαίνεται στιγμιότυπο ενώ το σύστημα βρίσκεται σε σφάλμα, αφού όλοι οι κόμβοι του δικτύου είναι εκτός σύνδεσης. Το χρώμα για την κατάσταση σφάλματος είναι το πορτοκαλί. Στο container «Nodes in Fault» φαίνεται ποιο κόμβοι βρίσκονται σε σφάλμα. Τονίζεται ότι στο «Sensors in Fault» εμφανίζεται άδεια λίστα και ο λόγος είναι ότι το σφάλμα δεν ήρθε από μέτρηση του αισθητήρα, αλλά από το δίκτυο RS-485. Επάνω στο δίκτυο επικοινωνούν φυσικά οι αισθητήρες, αλλά στα σφάλματα υπάρχει διαχωρισμός μεταξύ «σφάλμα κόμβου» που αφορά το δίκτυο αισθητήρων και «σφάλμα αισθητήρα» που αφορά τις μετρήσεις του αισθητήρα.

ID	Type	Sensor Alarm	Sensor Fault	pm1p0 (ug/m ³)	pm2p5 (ug/m ³)	pm4p0 (ug/m ³)	pm10p0 (ug/m ³)	Temperature (oC)	Humidity (%)	VOC	NOX
1	SENSE_TYPE_TEMPERATURE	false	false	-	-	-	-	25.13	-	-	-
2	SENSE_TYPE_TEMPERATURE	false	false	-	-	-	-	24.6	-	-	-
3	SENSE_TYPE_ENVIRONMENT	false	false	137.1	148.1	151.7	153.3	22.68	49.54	100	1

Εικόνα 82 - Στιγμιότυπο από τη λειτουργία της ιστοσελίδας σε κατάσταση σφάλματος

Στην εικόνα 83 φαίνεται στιγμιότυπο ενώ το σύστημα βρίσκεται σε συναγερμό, αφού η μέτρηση του αισθητήρα 2 έχει ξεπεράσει τα επιτρεπτά όρια και έχει «γράψει» 50.17 °C. Το χρώμα για την κατάσταση συναγερμού είναι το κόκκινο. Στο container «Sensors in Alarm» φαίνεται ποιο κόμβοι βρίσκονται σε συναγερμό.

ID	Type	Sensor Alarm	Sensor Fault	pm1p0 (ug/m ³)	pm2p5 (ug/m ³)	pm4p0 (ug/m ³)	pm10p0 (ug/m ³)	Temperature (oC)	Humidity (%)	VOC	NOX
1	SENSE_TYPE_TEMPERATURE	false	false	-	-	-	-	25.4	-	-	-
2	SENSE_TYPE_TEMPERATURE	true	false	-	-	-	-	50.17	-	-	-
3	SENSE_TYPE_ENVIRONMENT	false	false	121.7	129	130.1	130.6	23.93	45.09	45	1

Εικόνα 83 - Στιγμιότυπο από τη λειτουργία της ιστοσελίδας σε κατάσταση σφάλματος

11.1.1 Πρόσβαση από το τοπικό δίκτυο

Στις εικόνες 81, 82 και 83 η πρόσβαση έγινε απευθείας από τον server και συνεπώς χρησιμοποιείται στη γραμμή του browser απευθείας η διεύθυνση που «σηκώνει» ο server. Άρα στον browser στην προκειμένη περίπτωση θα πληκτρολογηθεί το «127.0.0.1:65535», όπου 65535 είναι η «πύλη» που χρησιμοποιείται στη διεύθυνση αυτή, η οποία είναι και αυτή παραμετροποιήσιμη από τον χρήστη κατά την εκκίνηση του συστήματος.

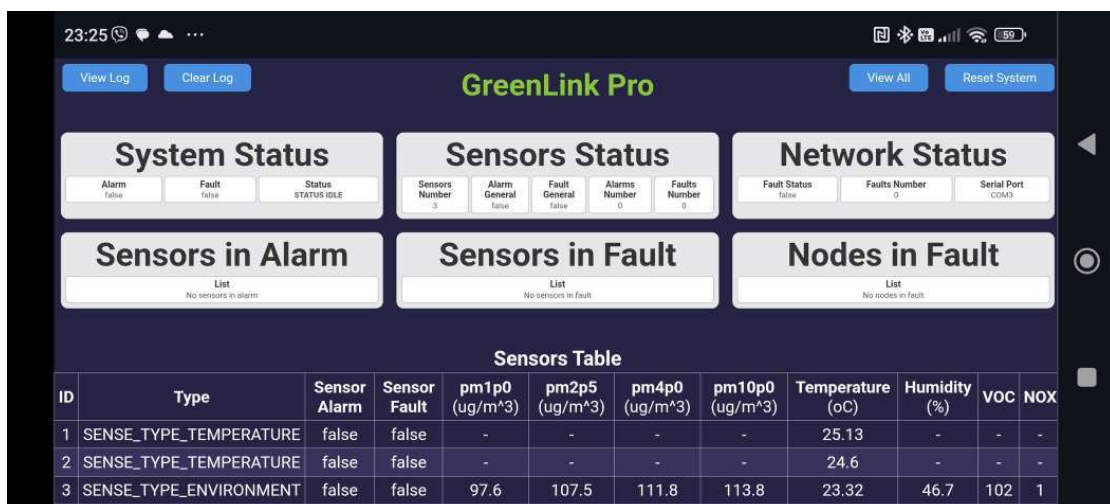
Σε περίπτωση που χρειάζεται να γίνει προσπέλαση από άλλη συσκευή στο τοπικό δίκτυο, τότε θα πρέπει να χρησιμοποιηθεί η IPv4 διεύθυνση του server στη γραμμή του browser που, στη συγκεκριμένη περίπτωση μέσω της εντολής «ipconfig», φαίνεται ότι είναι η 192.168.2.2 (εικόνα 84).

```
Ethernet adapter Ethernet 4:
Connection-specific DNS Suffix . :
IPv4 Address. . . . . : 192.168.2.2
```

Εικόνα 84 – Απεικόνιση στην κονσόλα της IPv4 διεύθυνσης μέσω της εντολής «ipconfig»

Συνεπώς για να γίνει προσπέλαση από το τοπικό δίκτυο (εικόνα 85), θα πρέπει να χρησιμοποιηθεί η IPv4 διεύθυνση του server, ακολουθούμενη και πάλι από την πύλη, δηλαδή στην προκειμένη περίπτωση στη γραμμή του browser θα πληκτρολογηθεί «192.168.2.2:65535».

Σημειώνεται ότι για να μπορεί να υπάρξει πρόσβαση από άλλη συσκευή του τοπικού δικτύου, χρειάστηκε να απενεργοποιηθεί το firewall στον server. Σε μια πραγματική εφαρμογή αυτό είναι ανεπίτρεπτο. Θα πρέπει το τμήμα IT να δώσει στο firewall τις συγκεκριμένες εξαιρέσεις, που είναι συνήθως static IP από συγκεκριμένες συσκευές, ώστε οι τελευταίες να θεωρούνται ασφαλείς από το firewall και να τις επιτρέπει την πρόσβαση.



Εικόνα 85 - Στιγμιότυπο από τη λειτουργία της ιστοσελίδας μέσω του κινητού τηλεφώνου

11.2 Components (.html)

Όλα τα στατικά χρώματα και σχήματα καθορίζονται από το αρχείο *index.html* (εικόνα 86). Ο όρος «στατικά» χαρακτηρίζει την εικόνα της ιστοσελίδας κατά την εκκίνηση. Στην πορεία κάποια από αυτά παραμένουν ως έχουν και δεν αλλάζουν ποτέ (πχ στη συγκεκριμένη ιστοσελίδα το σχήμα και το μέγεθος των containers), κάποια όμως που αλλάζουν καθορίζονται από τη λειτουργικότητα της ιστοσελίδας και τη λογική που εφαρμόζεται μέσω των αρχείων JavaScript.

```
<!-- Main Header -->
<h1 class="main-page-header">GreenLink Pro</h1>

<!-- Add View Log and Clear Log buttons on top-left corner -->
<div class="log-button-wrapper">
  <button id="view-log-button" class="log-button">View Log</button>
  <button id="clear-log-button" class="log-button">Clear Log</button>
</div>
```

Εικόνα 86 - Απόσπασμα από το *index.html*, όπου προστίθενται η επικεφαλίδα και δύο από τα κουμπιά

11.3 Στυλ (.css)

Όλα τα στατικά κείμενα καθορίζονται από το αρχείο *styles.css*. Ο όρος «στατικά» χαρακτηρίζει την εικόνα της ιστοσελίδας κατά την εκκίνηση. Στην πορεία κάποια από αυτά παραμένουν ως έχουν και δεν αλλάζουν ποτέ (πχ το κείμενο επάνω στα μπουτόν), κάποια όμως που αλλάζουν (πχ το χρώμα της επικεφαλίδας κατά τον συναγερμό) καθορίζονται από τη λειτουργικότητα της ιστοσελίδας και τη λογική που εφαρμόζεται μέσω των αρχείων JavaScript.

```
20  /* Main Header Styles */
21  .main-page-header {
22      font-size: 3vw;
23      text-align: center;
24      color: #8cc63f;
25      margin-top: 2vh; /* Adjusted margin-top */
26  }
```

Εικόνα 87 - Απόσπασμα από το αρχείο *styles.css*, που καθορίζει τη γραμματοσειρά στο κείμενο της επικεφαλίδας

11.4 Λειτουργικότητα (.js)

Η λειτουργικότητα της σελίδας απαιτεί τη χρήση γλώσσας που να είναι σε θέση να στέλνει και να λαμβάνει δεδομένα, να ανανεώνει δυναμικά τη σελίδα ανάλογα με αποφάσεις που παίρνει από τα δεδομένα αυτά και γενικά να έχει ικανότητα αποφάσεων και ενεργειών και

όχι απλά να καθορίζει θέσεις και χρώματα σχημάτων και κειμένων. Η γλώσσα για την εργασία αυτή είναι η JavaScript και τα αρχεία που γράφτηκαν για την εφαρμογή αυτή, είναι τα αρχεία *main.js* (εικόνα 88) και *module.js*.

```
// Support for Single Page Application (SPA)
// Set up an interval to regularly refresh the page
setInterval(() => {
  location.reload(true); // Pass true to force a reload from the server, bypassing the cache
}, Defines.UPDATE_INTERVAL_MS);
```

Εικόνα 88 - Απόσπασμα από το αρχείο *main.js*, όπου επιβάλλεται ανανέωση της ιστοσελίδας ανά 15 δευτερόλεπτα

12 Κατασκευή παρουσίασης

Στην εικόνα 89 φαίνεται η τελική κατασκευή, η οποία είναι προσαρμοσμένη στις ανάγκες παρουσίασης της διπλωματικής εργασίας. Βλέποντας τη φωτογραφία της εικόνας, αριστερά έχει τοποθετηθεί πολύμπριζο το οποίο έχει στερεωθεί με δετικά (tie wraps). Στη μέση βρίσκεται ο προσαρμογέας RS-485 με το USB συνδεδεμένο στην κάτω πλευρά του, ενώ κάτω από τον προσαρμογέα βρίσκεται το module SEN55, που περιέχει τον περιβαλλοντολογικό αισθητήρα. Οι τρεις πλακέτες με τους αισθητήρες βρίσκονται στα δεξιά. Περιμετρικά υπάρχουν κανάλια, μέσα στα οποία βρίσκονται τα καλώδια του RS-485, καθώς και τα καλώδια των τροφοδοτικών 12V, που δίνουν τάση στις πλακέτες. Τα καλώδια του SEN55 περνάνε και αυτά μέσα από τα κανάλια, για να καταλήξουν στην πλακέτα με διεύθυνση 3, η οποία βρίσκεται ακριβώς δεξιά του SEN55. Οι άλλες δύο πλακέτες μετράνε μόνο θερμοκρασία, με τον αισθητήρα KTY81-110 να βρίσκεται βιδωμένος απευθείας στις κλεμμες των πλακετών.



Εικόνα 89 - Κάτοψη της τελικής κατασκευής

Στην εικόνα 90 φαίνεται το σύστημα σε λειτουργία. Η οθόνη του Η/Υ απεικονίζει ζωντανά τα δεδομένα των αισθητήρων, μαζί με τις υπόλοιπες πληροφορίες του συστήματος.



Εικόνα 90 - Ζωντανή απεικόνιση του συστήματος στο GreenLink Pro

13 Μελλοντικές επεκτάσεις

Όπως έχει ήδη τονιστεί σε αρκετά σημεία της εργασίας, η παρούσα εφαρμογή δεν μπορεί να θεωρηθεί έτοιμη για το εμπόριο, αποτελεί όμως το ελάχιστο βιώσιμο προϊόν, το οποίο μπορεί να σταλεί σε κάποιον πελάτη για να δώσει πρώιμο feedback, το οποίο θα βελτιώσει το προϊόν. Επίσης οι δοκιμές που πρέπει να γίνουν απαιτούν ακριβό εξοπλισμό που πολλές φορές ούτε οι ίδιες οι επιχειρήσεις δεν διαθέτουν.

Στο πλαίσιο αυτό υπάρχουν πάρα πολλές επεκτάσεις που θα μπορούσε να σκεφτεί κάποιος είτε γιατί διατίθενται από τον ανταγωνισμό είτε για τον ακριβώς αντίθετο λόγο, για το ότι δηλαδή θα μπορούσε να δοθεί υπεραξία από ένα νέο feature που θα το κάνει να ξεχωρίσει από τον ανταγωνισμό. Σε κάθε περίπτωση μερικές μελλοντικές επεκτάσεις θα μπορούσαν να είναι οι παρακάτω:

- Περιφερειακά όπως σειρήνες, μπουτόν πανικού, επεκτάσεις εισόδων εξόδων κτλ, τα οποία είτε με Ethernet να μπορούν να επικοινωνούν απευθείας με το PC, είτε με RS-485 ώστε να επικοινωνούν επάνω στο δίκτυο των αισθητήρων.
- Νέοι αισθητήρες άλλου τύπου, όπως για παράδειγμα ανιχνευτής καπνού, μαγνητικοί αισθητήρες που να ανιχνεύονται αν κάποια πόρτα που οδηγεί σε χώρο με επικίνδυνα αέρια είναι ανοιχτή η κλειστή, υπέρυθρους ανιχνευτές κίνησης για να ανιχνεύουν αν βρίσκεται άνθρωπος σε χώρο που υπάρχει διαρροή κοκ.

- Αισθητήρες ασύρματου τύπου για σημεία όπου η πρόσβαση με καλώδιο είναι δύσκολη.
- Διαχωρισμός των ανιχνευτών του συστήματος με βάση λογικές ζώνες.
- Χρήση Docker, ώστε να μπορούν εύκολα να πραγματοποιηθούν τεστ και μάλιστα αυτοματοποιημένα.
- Ανέβασμα δεδομένων στο cloud για πρόσβαση από οπουδήποτε στον κόσμο.
- Βάση δεδομένων για αποθήκευση των μετρήσεων και πιο συγκεκριμένα διαχείριση big data, ώστε να υπάρχει ανάλυση σε βάθος των μετρήσεων. Ο απώτερος σκοπός είναι αλγόριθμοι τεχνητής νοημοσύνης, που σύμφωνα με τα data θα προβλέπουν εάν κάποιος αισθητήρας «οδεύει» προς τον συναγερμό, πριν καν οι μετρήσεις του ξεπεράσουν τα όρια συναγερμού.
- Mobile application σε περίπτωση που η εφαρμογή «βγει» στο cloud.
- Ο περιβαντολλογικός αισθητήρας θα μπορούσε να οδηγήσει στη δημιουργία και ενός άλλου συστήματος εξωτερικού χώρου, το οποίο θα έχει τη δυνατότητα να μετράει την ποιότητα του αέρα της ατμόσφαιρας και να εξάγει τον δείκτη δυσφορίας, ώστε να στέλνονται έγκαιρα οι πληροφορίες στους αρμόδιους φορείς για να δράσουν ανάλογα.

14 Επίλογος – Συμπεράσματα

Σε αυτήν την εργασία παρουσιάστηκε η υλοποίηση ενός δικτύου αισθητήρων στο πλαίσιο των συστημάτων ασφαλείας. Οι αισθητήρες που χρησιμοποιήθηκαν ήταν ο περιβαντολλογικός και ο αισθητήρας θερμοκρασίας.

Αναλύθηκε ο σκοπός ενός τέτοιου συστήματος, καθώς έκθεση του ανθρώπου σε συγκεκριμένα αέρια μπορεί να προκαλέσει μόνιμες βλάβες, μέχρι και θάνατο. Για τον λόγο αυτόν έγινε μια μικρή ιστορική αναδρομή σε σχέση με το ποιες ανάγκες οδήγησαν σε αυτά τα συστήματα, ποια ήταν η εξέλιξή τους και ποια τα σημερινά δεδομένα στην αγορά, μέσω κάποιων συγκεκριμένων συστημάτων του ανταγωνισμού που παρατέθηκαν.

Πραγματοποιήθηκε, επίσης, ανάλυση σε βάθος του συστήματος που υλοποιήθηκε σε επίπεδο αισθητήρα, δικτύου RS-485, κεντρικής μονάδας επεξεργασίας, Server και ιστοσελίδας. Η ανάλυση αφορούσε το hardware, τις μετρήσεις και το firmware του αισθητήρα, καθώς και το πρόγραμμα της κεντρικής μονάδας επεξεργασίας. Τα προγράμματα δομήθηκαν όλα με βάση το "Clean Architecture".

Το συμπέρασμα που βγαίνει είναι ότι ο περιβαντολλογικός αισθητήρας είναι το ποιοτικότερο χαρακτηριστικό του συστήματος, με τον αισθητήρα θερμοκρασίας να προσφέρει μια πιο οικονομική, αλλά και με λιγότερες δυνατότητες λύση. Οι μετρήσεις του είναι αρκετά πιο ασταθείς και σε σύγκριση με πραγματικό θερμόμετρο φάνηκε να έχει απόκλιση, σε αντίθεση με τον περιβαντολλογικό αισθητήρα, που μετράει τη θερμοκρασία πολύ κοντά στο θερμόμετρο, εφαρμόζοντας μάλιστα εσωτερικά software φίλτρα που βελτιώνουν ακόμη περισσότερο τη συμπεριφορά του. Στην περίπτωση του αισθητήρα θερμοκρασίας μάλιστα, παρατηρήθηκε απόκλιση ακόμα και ανάμεσα σε δύο ίδιους αισθητήρες μεταξύ τους. Επίσης ο περιβαντολλογικός αισθητήρας φάνηκε να ανταποκρίθηκε όπως αναμενόταν σε έκθεση σε

αέρια, θερμοκρασία και υγρασία, πάντα με τη σημείωση ότι πραγματικά αξιόπιστες μετρήσεις μπορούν να υπάρξουν μόνο σε ειδικούς χώρους και με ειδικά και ακριβά εργαλεία. Οι αισθητήρες θερμοκρασίας από την άλλη, λόγω των αποκλίσεων που παρουσιάζουν, θα πρέπει να βαθμονομηθούν κατά την παραγωγή πριν κυκλοφορήσουν στο εμπόριο.

Ως τελική εικόνα μένει ένα σύστημα που με λίγα χρόνια ανάπτυξης αλγορίθμων, βελτίωσης του hardware και του software, εξαντλητικού testing, καθώς και κάποιων επιπλέον επεκτάσεων, θα μπορούσε να σταθεί στο εμπόριο ως ένα ανταγωνιστικό (και ίσως καινοτόμο λόγω του περιβαντολλογικού αισθητήρα) σύστημα ασφαλείας.

Το συγκεκριμένο σύστημα όπως υλοποιήθηκε δεν είναι ίδιο με κανένα άλλο της αγοράς, υπό την έννοια ότι κατά την αναζήτηση αντίστοιχων συστημάτων ασφαλείας, δεν βρέθηκε άλλο που να χρησιμοποιεί περιβαντολλογικό αισθητήρα. Οι πιθανές επεκτάσεις του συστήματος είναι πάρα πολλές και περιλαμβάνουν μέχρι και κατασκευή αντίστοιχου συστήματος εξωτερικού χώρου, το οποίο θα μπορεί μέσω των μετρήσεων του SEN55, να εξάγει των δείκτη δυσφορίας και να ενημερώνει τις αρμόδιες κρατικές υπηρεσίες να αναλάβουν δράση (πχ με ρύθμιση της κυκλοφορίας των αυτοκινήτων στο σημείο που οι ρύποι είναι ανεβασμένοι).

Εν μέσω κλιματικής κρίσης και παγκόσμιας προσπάθειας για μείωση των ρύπων, τέτοιου είδους εφαρμογές θα είναι όλο και πιο αναγκαίες και η ζήτησή τους θα μεγαλώνει. Το σύστημα ασφαλείας που παρουσιάστηκε θα μπορούσε, μεταξύ άλλων, να θεωρηθεί και ένα proof of concept του περιβαντολλογικού αισθητήρα και στο κατά πόσο μπορεί ο τελευταίος να συμβάλλει ακόμα περισσότερο προς τους περιβαντολλογικούς, αυτούς, σκοπούς. Τα ευρήματα, όπως παρουσιάστηκαν, δεν αφήνουν αμφιβολία για την ποιότητα του αισθητήρα και τη δυνατότητα για πολύπλευρη χρήση, που περιλαμβάνει και τις περιβαντολλογικές εφαρμογές.

15 Βιβλιογραφικές αναφορές

- [1] Englert, N. (2004). Fine particles and human health—a review of epidemiological studies. *Toxicology letters*, 149(1-3), 235-242.
- [2] United States Environmental Protection Agency. (n.d.). How Does PM Affect Human Health? Ανακτήθηκε από <https://www3.epa.gov/region1/airquality/pm-human-health.html>
- [3] Rumchev, K., Brown, H., & Spickett, J. (2007). Volatile organic compounds: do they present a risk to our health?. *Reviews on environmental health*, 22(1), 39-56.
- [4] Boningari, T., & Smirniotis, P. G. (2016). Impact of nitrogen oxides on the environment and human health: Mn-based materials for the NO_x abatement. *Current Opinion in Chemical Engineering*, 13, 133-141.
- [5] Krzeszowiak, J., Stefanow, D., & Pawlas, K. (2016). The impact of particulate matter (PM) and nitric oxides (NO_x) on human health and an analysis of selected sources accounting for their emission in Poland. *Medycyna Środowiskowa-Environmental Medicine*, 19(3), 7-15.
- [6] Davalle, J. (2011, May 1). Gas Detection Through the Ages. EHS Today. <https://www.ehstoday.com/industrial-hygiene/article/21913604/gas-detection-through-the-ages>
- [7] Honeywell. (n.d.). The evolution and challenges of gas detection technologies. Ανακτήθηκε από <https://sps.honeywell.com/content/dam/honeywell-edam/sps/common/en-us/support/software-and-downloads/documents/sps-hgas-the-evolution-and-challenges-of-gas-detection-technologies.pdf>
- [8] Awang, Z. (2014). Gas sensors: A review. *Sens. Transducers*, 168(4), 61-75.
- [9] Accordia. (2023). PYROLINK F HS HyperShield. Ανακτήθηκε από https://www.accordia.gr/sites/default/files/2022-10/Pyrolink_F_HS_HyperShield_PH120_FE180_Tds_En.pdf
- [10] Honeywell. (2013). Sensepoint XCD. Ανακτήθηκε από <https://prod-edam.honeywell.com/content/dam/honeywell-edam/sps/his/en-ca/products/gas-and-flame-detection/documents/industrial-fixed/sensepoint-xcd/sps-his-12542-sensepoint-xcds01077-v5-02-13-emea-lr.pdf?download=false>
- [11] Olympia Electronics. (n.d.). Gas detection control panel up to 16 inputs. Ανακτήθηκε από https://www.olympia-electronics.com/sites/default/files/921316000_09_019_1.pdf
- [12] Eaton. (2019). CF1000 range - Intelligent addressable control panel. Ανακτήθηκε από <https://www.eaton.com/content/dam/eaton/products/safety-security-emergency-communications/fire/addressable/cf1000-panel/en/eaton-fire-addressable-panel-cf1100-datasheet-v1.1-0619.pdf>
- [13] Clarity. (2023). Identify air pollution sources with the Black Carbon Module. Ανακτήθηκε από https://click.clarity.io/hubfs/Marketing%20Assets%20-%20PDFs/Product%20and%20Specification%20Sheets/2%20Pager%20Flyer%20%2B%20Specifications%20%E2%80%94%20Black%20Carbon%20Module.pdf?_ga=2.95320732.141020714

[5.1707251800-](#)

[566316173.1707067604&_gac=1.84066923.1707251946.CjwKCAiAiP2tBhBXEiwACslnoaMQOhpAoRz33dHbuMetAnteKvFtrbZ4Y2om2oEGSuxLLptwBFAORoC56oQAvD_BwE](#)

[14] New York State Department of Health (n.d.). *Particle Pollution and Health*. Ανακτήθηκε από https://www.health.ny.gov/environmental/indoors/air/pmq_a.htm

[15] SENSIRION. (n.d.). What is Sensirion's NOx Index? Ανακτήθηκε από https://sensirion.com/media/documents/9F289B95/6294DFFC/Info_Note_NOx_Index.pdf

[16] SENSIRION. (2022). Datasheet SEN55. Ανακτήθηκε από https://SENSIRION.com/media/documents/6791EFA0/62A1F68F/SENSIRION_Datasheet_Environmental_Node_SEN5x.pdf

[17] SENSIRION. (2022). Temperature Acceleration and Compensation Instructions for SEN5x. Ανακτήθηκε από https://sensirion.com/media/documents/9B9DE2A7/61E957EB/Sensirion_Temperature_Acceleration_and_Compensation_Instructions_SEN.pdf

[18] SENSIRION. (2022). Mechanical Design and Assembly Guidelines for SEN5x. Ανακτήθηκε από https://sensirion.com/media/documents/546FBC5B/61E9586E/Sensirion_Mechanical_Design_and_Assembly_Guidelines_SEN5x.pdf

[19] NXP. (2008). KTY81 series. Ανακτήθηκε από https://www.nxp.com/docs/en/datasheet/KTY81_SER.pdf

[20] Texas Instruments. (2022). The RS-485 Design Guide. Ανακτήθηκε από https://www.ti.com/lit/an/slla272d/slla272d.pdf?ts=1703675361122&ref_url=https%253A%252F%252Fwww.google.com%252F

[21] On Semiconductors. (2019). Inverting Regulator - Buck, Boost, Switching. Ανακτήθηκε από <https://www.onsemi.com/pdf/datasheet/mc34063a-d.pdf>

[22] Texas Instruments. (2019). TPS7A05 1- μ A Ultralow IQ, 200-mA, Low-Dropout Regulator in a Small-Size Package. Ανακτήθηκε από https://www.ti.com/lit/ds/symlink/tps7a05.pdf?ts=1708882329605&ref_url=https%253A%252F%252Fwww.google.com%252F

[23] ST Microelectronics. (2019). ST3485EB, ST3485EC, ST3485EI, ST3485EIIY datasheet. Ανακτήθηκε από <https://www.st.com/resource/en/datasheet/st3485eb.pdf>

[24] FTDI. (2020). FT232R USB UART IC Datasheet (version 2.16). Ανακτήθηκε από https://ftdichip.com/wp-content/uploads/2020/08/DS_FT232R.pdf

[25] Martin, R. C. (2017). *Clean Architecture: A Craftsman's Guide to Software Structure and Design* (1st edition). Pearson Education, Inc. USA.

[26] Britannica. (2024). Understand the chemistry behind what makes a match light. Ανακτήθηκε από <https://www.britannica.com/video/188815/chemistry-match>

[27] Braun, M., Koger, F., Klingelhöfer, D., Müller, R., & Groneberg, D. A. (2019). Particulate matter emissions of four different cigarette types of one popular brand: Influence of tobacco strength and additives. *International journal of environmental research and public health*, 16(2), 263.

[28] Pazo, D. Y., Moliere, F., Sampson, M. M., Reese, C. M., Agnew-Heard, K. A., Walters, M. J., ... & Chambers, D. M. (2016). Mainstream smoke levels of volatile organic compounds in 50 US domestic cigarette brands smoked with the ISO and Canadian intense protocols. *Nicotine & tobacco research*, 18(9), 1886-1894

[29] Braun, M., Klingelhöfer, D., Müller, R., & Groneberg, D. A. (2021). The impact of second-hand smoke on nitrogen oxides concentrations in a small interior. *Scientific Reports*, 11(1), 11703.

[30] Maxim. (2014). HOW FAR AND HOW FAST CAN YOU GO WITH RS-485?. Ανακτήθηκε από <https://pdfserv.maximintegrated.com/en/an/AN3884.pdf>

[31] Belden. (2020). DataTray® 120Ω Twinax, 1 Pr #18 Str TC, PP Ins, OS+TC Brd, PVC Jkt, TC, PLTC, ITC, CMG. Ανακτήθηκε από <https://assets.belden.com/m/7321f04b5003f7cc/original/Data-Cable-3074F-2020-10-EN.pdf>

16 Πίνακες περιεχομένων εικόνων και πινάκων

16.1 Πίνακας περιεχομένων εικόνων

Εικόνα 1 - Το πυρίμαχο καλώδιο PH120 FE180 [9]	13
Εικόνα 2 - Ο ανιχνευτής αερίων Sensepoint XCD [10]	13
Εικόνα 3 - Τα τεχνικά χαρακτηριστικά του Sensepoint XCD [10]	14
Εικόνα 4 - Ο πίνακας ανίχνευσης αερίων BS-3xx [11]	15
Εικόνα 5 - Τα τεχνικά χαρακτηριστικά του πίνακα ανίχνευσης αερίων BS-3xx [11]	15
Εικόνα 6 - Ο πίνακας ανίχνευσης αερίων BS-3xx [12]	17
Εικόνα 7 - Τα τεχνικά χαρακτηριστικά του πίνακα πυρανίχνευσης CF1100 [12]	17
Εικόνα 8 - Το module ανίχνευσης εξωτερικού χώρου «Black Carbon» [13]	19
Εικόνα 9 - Βαθμίδες συστήματος	19
Εικόνα 10 - : Απαιτήσεις συστήματος	22
Εικόνα 11 - Μπλοκ διάγραμμα αισθητήρα SEN55 [16]	26
Εικόνα 12 - Τεχνικά χαρακτηριστικά SEN55 για τη μέτρηση θερμοκρασίας και υγρασίας [16]	27
Εικόνα 13 - Τεχνικά χαρακτηριστικά SEN55 για τη μέτρηση αιωρούμενων σωματιδίων [16]	27
Εικόνα 14 - Τεχνικά χαρακτηριστικά SEN55 για τη μέτρηση των αερίων [16]	28
Εικόνα 15 - Ηλεκτρικά χαρακτηριστικά του SEN55 για τη μέτρηση των αερίων [16]	29
Εικόνα 16 - Λειτουργικές καταστάσεις (operating modes) του SEN55 [16]	29
Εικόνα 17 - Διάγραμμα ακολουθίας πρωτοκόλλου επικοινωνίας αισθητήρων	31
Εικόνα 18 - Πακέτο αποστολής από τον master προς τον slave	32
Εικόνα 19 - Πακέτο αποστολής από τον slave προς τον master	32
Εικόνα 20 - LDO regulator 5V DC σε 3.3V DC	37
Εικόνα 21 - LDO regulator 5V DC σε 3.3V DC	38
Εικόνα 22 - Οδηγός RS-485	39
Εικόνα 23 - Συνδέσεις αισθητήρα SEN55	39
Εικόνα 24 - Συνδέσεις αισθητήρα KTY81-110	40
Εικόνα 25 - Συνδέσεις ψηφιακών εισόδων - εξόδων	41
Εικόνα 26 - Εξωτερικό τροφοδοτικό πρίζας 12V DC / 16A της πλακέτας του αισθητήρα	41
Εικόνα 27 - Εξωτερικό τροφοδοτικό πρίζας 12V DC / 16A της πλακέτας του αισθητήρα	42
Εικόνα 28 - Κάτοψη της πλακέτας όπως φαίνεται εντός του σχεδίου PCB	42
Εικόνα 29 - Φωτογραφία προσαρμογέας δικτύου	44
Εικόνα 30 - Προσαρμογέας δικτύου: σύνδεση FT232R με USB και RS-485	45
Εικόνα 31 - Προσαρμογέας δικτύου: σύνδεση FT232R με USB και RS-485	46
Εικόνα 32 - Διαστρωμάτωση προγράμματος αισθητήρα με βάση το Clean Architecture [25]	47
Εικόνα 33 - Components αισθητήρα ανά επίπεδο προγράμματος βάσει Clean Architecture [25]	48
Εικόνα 34 - Διάγραμμα δραστηριοτήτων στο επίπεδο της εφαρμογής	49
Εικόνα 35 - Firmware αισθητήρα: main loop (<i>appif.c</i>)	49
Εικόνα 36 - "Interface" function του serial net module (<i>modif.c</i>)	50
Εικόνα 37 - Ρυθμίσεις pins και των περιφερειακών μικροελεγκτή	52
Εικόνα 38 - Ρυθμίσεις εσωτερικών clock του μικροελεγκτή	52
Εικόνα 39 - Υπόδειγμα αρχείου log	53
Εικόνα 40 - Υπόδειγμα αρχείου “_notes.log”	53
Εικόνα 41 - Απόκριση περιβαλλοντικού αισθητήρα στην έκθεσή του σε φως από φακό	54
Εικόνα 42 - Απόκριση ηρεμίας περιβαλλοντικού αισθητήρα σε διάρκεια είκοσι τριών ωρών	55
Εικόνα 43 - Απόκριση περιβαλλοντικού αισθητήρα στη χρήση στεγνωτήρα μαλλιών	56
Εικόνα 44 - Ένδειξη πραγματικού θερμομέτρου στις 07/01/2024 – 22:51	58
Εικόνα 45 - Απόκριση περιβαλλοντικού αισθητήρα στην έκθεση σε καπνό ανάμματος σπύριου	59
Εικόνα 46 - Απόκριση περιβαλλοντικού αισθητήρα στην έκθεση σε καπνό από τσιγάρο	61

Εικόνα 47 - Απόκριση περιβαλλοντολογικού αισθητήρα στην έκθεση σε βουτάνιο	62
Εικόνα 48 - Απόκριση ηρεμίας αναλογικού αισθητήρα	63
Εικόνα 49 – Η απόκριση της εικόνας 48, με εφαρμογή κυλιόμενου MO 2000 δειγμάτων	64
Εικόνα 50 - Το αρχείο <code>sensor_logger.py</code> ρυθμίζεται να επικοινωνεί με τρεις διευθύνσεις (1, 2, 3)	64
Εικόνα 51 - Απόκριση αισθητήρα θερμοκρασίας με διεύθυνση 1	65
Εικόνα 52 - Απόκριση αισθητήρα θερμοκρασίας με διεύθυνση 2	66
Εικόνα 53 - Απόκριση περιβαλλοντολογικού αισθητήρα με διεύθυνση 3	66
Εικόνα 54 - Μετρήσεις ρεύματος με τη χρήση αμπερομέτρου	68
Εικόνα 55 - Σύνδεση αισθητήρα KTY81-110 στην πλακέτα	70
Εικόνα 56 - Η τυπική αντίσταση του KTY81-110 στους 25oC είναι 1K [19]	70
Εικόνα 57 - Κύκλωμα προσομοίωσης για 400 μέτρα στρεπτού ζεύγους καλωδίου	72
Εικόνα 58 - Συνδέσεις πλακέτας αισθητήρα	74
Εικόνα 59 - Συνδέσεις προσαρμογέα δικτύου RS-485	74
Εικόνα 60 - Επιλογή διεύθυνσης αισθητήρα μέσω των CN8, CN9	75
Εικόνα 61 - Αρχική οθόνη εκκίνησης του λογισμικού του ΚΕΣ «GreekLinkPro»	76
Εικόνα 62 - Παράδειγμα αρχείου δικτύωσης αισθητήρων στο Excel	77
Εικόνα 63 - Παράδειγμα αρχείου δικτύωσης αισθητήρων σε κειμενογράφο	78
Εικόνα 64 - Μπλοκ διάγραμμα των thread του προγράμματος	80
Εικόνα 65 - Διαστρωμάτωση προγράμματος ΚΕΣ με βάση το Clean Architecture [25]	81
Εικόνα 66 - Ατέρμονας βρόχος της διαχείρισης αισθητήρα (<code>app.py</code>)	83
Εικόνα 67 –GreenLink Pro components ανά layer προγράμματος, βάσει Clean Architecture [25]	84
Εικόνα 68 - Storage module interface (<code>modif.py</code>)	85
Εικόνα 69 - Βασικές λειτουργίες του module του δικτύου RS-485 (<code>l2.py</code>)	86
Εικόνα 70 - Βασικές λειτουργίες του module αισθητήρα (<code>sensor.py</code>)	86
Εικόνα 71 - Βασικές λειτουργίες του module αποθήκευσης (<code>store.py</code>)	88
Εικόνα 72 - Βασικές λειτουργίες του User Interface module (<code>ui.py</code>)	89
Εικόνα 73 - Διαχείριση του http request από τον presenter (<code>ui_presenter.py</code>)	90
Εικόνα 74 - Sequence diagram του HTTP request	92
Εικόνα 75 - Η βασική συνάρτηση « <code>log()</code> » του logging module (<code>log.py</code>)	92
Εικόνα 76 - Καταχώρηση συμβάντος «συναγερμός αισθητήρα»	94
Εικόνα 77 – Εκκίνηση main thread από το thread της γραφικής εκκίνησης (<code>start.py</code>)	94
Εικόνα 78 - Ο ατέρμονας βρόχος του συστήματος	95
Εικόνα 79 - Η συνάρτηση τερματισμού του main thread	96
Εικόνα 80 - Οργάνωση αρχείων του project (repository)	96
Εικόνα 81 - Στιγμιότυπο από τη λειτουργία της ιστοσελίδας σε κατάσταση ηρεμίας	97
Εικόνα 82 - Στιγμιότυπο από τη λειτουργία της ιστοσελίδας σε κατάσταση σφάλματος	98
Εικόνα 83 - Στιγμιότυπο από τη λειτουργία της ιστοσελίδας σε κατάσταση σφάλματος	98
Εικόνα 84 – Απεικόνιση στην κονσόλα της IPv4 διεύθυνσης μέσω της εντολής « <code>ipconfig</code> »	99
Εικόνα 85 - Στιγμιότυπο από τη λειτουργία της ιστοσελίδας μέσω του κινητού τηλεφώνου	99
Εικόνα 86 - Απόσπασμα από το <code>index.html</code> , όπου προστίθενται η επικεφαλίδα και δύο από τα κουμπιά	100
Εικόνα 87 - Απόσπασμα από το αρχείο <code>styles.css</code> , που καθορίζει τη γραμματοσειρά στο κείμενο της επικεφαλίδας	100
Εικόνα 88 - Απόσπασμα από το αρχείο <code>main.js</code> , όπου επιβάλλεται ανανέωση της ιστοσελίδας ανά 15 δευτερόλεπτα	101
Εικόνα 89 - Κάτοψη της τελικής κατασκευής	101
Εικόνα 90 - Ζωντανή απεικόνιση του συστήματος στο GreenLink Pro	102

16.2 Πίνακας περιεχομένων πινάκων

Πίνακας 1 - Χαρακτηριστικά δικτύου αισθητήρων	30
Πίνακας 2 – Μετρήσεις ρεύματος αισθητήρων	68
Πίνακας 3 - Πιθανές διευθύνσεις αισθητήρα, ανάλογα με την επιλογή των CN8, CN9	75

17 Παραρτήματα

17.1 Συντομογραφίες

ΕΕ: Ευρωπαϊκή Ένωση

Η/Υ: Ηλεκτρονικός Υπολογιστής

ΚΕΣ: Κεντρικός Ελεγκτής Συστήματος

ΜΟ: Μέσος Όρος

ADC: Μετατροπέας αναλογικού σήματος σε ψηφιακό (A/D Converter)

MCU: Microcontroller Unit (Μονάδα Μικροεπεξεργαστή)

MVP: Model-View-Presenter

NOx: Οξείδια του Αζώτου (Nitric oxide)

VOC: Πτητικές Οργανικές Ενώσεις (Volatile Organic Compounds)

UI: User Interface – Διεπαφή Χρήστη