



ΠΑΝΕΠΙΣΤΗΜΙΟ ΔΥΤΙΚΗΣ ΑΤΤΙΚΗΣ
ΣΧΟΛΗ ΜΗΧΑΝΙΚΩΝ
ΤΜΗΜΑ ΗΛΕΚΤΡΟΛΟΓΩΝ & ΗΛΕΚΤΡΟΝΙΚΩΝ ΜΗΧΑΝΙΚΩΝ

Διπλωματική Εργασία

**Ανάπτυξη Ολοκληρωμένου Συστήματος Ελέγχου Ροής με Διατάξεις
Μικροελεγκτών**

Φοιτητής: Βούλαρης Κωνσταντίνος
ΑΜ: 18387060

Επιβλέπων Καθηγητής

Γρηγόριος Καλτσάς
Καθηγητής

ΑΘΗΝΑ-ΑΙΓΑΛΕΩ, ΙΟΥΛΙΟΣ 2024



UNIVERSITY OF WEST ATTICA
FACULTY OF ENGINEERING
DEPARTMENT OF ELECTRICAL & ELECTRONICS ENGINEERING

Diploma Thesis

**Development of an Integrated Flow Control System
Utilizing Microcontrollers**

Student: Voulgaris Konstantinos
Registration Number: 18387060

Supervisor

Grigoris Kaltsas
Professor

ATHENS-EGALEO, JULY 2024

Η Διπλωματική Εργασία έγινε αποδεκτή και βαθμολογήθηκε από την εξής τριμελή επιτροπή:

Γρηγόριος Καλτσάς, Καθηγητής	Γεώργιος Πάτσης, Καθηγητής	Δημήτριος Γουστουρίδης, Αν. Καθηγητής
(Υπογραφή)	(Υπογραφή)	(Υπογραφή)

Copyright © Με επιφύλαξη παντός δικαιώματος. All rights reserved.

ΠΑΝΕΠΙΣΤΗΜΙΟ ΔΥΤΙΚΗΣ ΑΤΤΙΚΗΣ και Κωνσταντίνος Βούλγαρης
Ιούλιος, 2024

Απαγορεύεται η αντιγραφή, αποθήκευση και διανομή της παρούσας εργασίας, εξ ολοκλήρου ή τμήματος αυτής, για εμπορικό σκοπό. Επιτρέπεται η ανατύπωση, αποθήκευση και διανομή για σκοπό μη κερδοσκοπικό, εκπαιδευτικής ή ερευνητικής φύσης, υπό την προϋπόθεση να αναφέρεται η πηγή προέλευσης και να διατηρείται το παρόν μήνυμα. Ερωτήματα που αφορούν τη χρήση της εργασίας για κερδοσκοπικό σκοπό πρέπει να απευθύνονται προς τους συγγραφείς.

Οι απόψεις και τα συμπεράσματα που περιέχονται σε αυτό το έγγραφο εκφράζουν τον/την συγγραφέα του και δεν πρέπει να ερμηνευθεί ότι αντιπροσωπεύουν τις θέσεις του επιβλέποντος, της επιτροπής εξέτασης ή τις επίσημες θέσεις του Τμήματος και του Ιδρύματος.

ΔΗΛΩΣΗ ΣΥΓΓΡΑΦΕΑ ΔΙΠΛΩΜΑΤΙΚΗΣ ΕΡΓΑΣΙΑΣ

Ο κάτωθι υπογεγραμμένος Κωνσταντίνος Βούλγαρης του Μιχαήλ, με αριθμό μητρώου 18387060 φοιτητής του Πανεπιστημίου Δυτικής Αττικής της Σχολής ΜΗΧΑΝΙΚΩΝ του Τμήματος ΗΛΕΚΤΡΟΛΟΓΩΝ ΚΑΙ ΗΛΕΚΤΡΟΝΙΚΩΝ ΜΗΧΑΝΙΚΩΝ,

δηλώνω υπεύθυνα ότι:

«Είμαι συγγραφέας αυτής της διπλωματικής εργασίας και ότι κάθε βοήθεια την οποία είχα για την προετοιμασία της είναι πλήρως αναγνωρισμένη και αναφέρεται στην εργασία. Επίσης, οι όποιες πηγές από τις οποίες έκανα χρήση δεδομένων, ιδεών ή λέξεων, είτε ακριβώς είτε παραφρασμένες, αναφέρονται στο σύνολό τους, με πλήρη αναφορά στους συγγραφείς, τον εκδοτικό οίκο ή το περιοδικό, συμπεριλαμβανομένων και των πηγών που ενδεχομένως χρησιμοποιήθηκαν από το διαδίκτυο. Επίσης, βεβαιώνω ότι αυτή η εργασία έχει συγγραφεί από μένα αποκλειστικά και αποτελεί προϊόν πνευματικής ιδιοκτησίας τόσο δικής μου, όσο και του Ιδρύματος.

Παράβαση της ανωτέρω ακαδημαϊκής μου ευθύνης αποτελεί ουσιώδη λόγο για την ανάκληση του διπλώματός μου.

Επιθυμώ την απαγόρευση πρόσβασης στο πλήρες κείμενο της εργασίας μου πάρα μόνο και έπειτα από αίτησή μου στη Βιβλιοθήκη και έγκριση του επιβλέποντος καθηγητή.»

Ο Δηλών
Κωνσταντίνος Βούλγαρης



ΕΥΧΑΡΙΣΤΙΕΣ

Πρώτα απ' όλα θα ήθελα να ευχαριστήσω τον επιβλέποντα καθηγητή, κο Γρηγόριο Καλτσά για την βοήθεια και την καθοδήγηση που μου παρείχε, καθ' όλη τη διάρκεια της παρούσας διπλωματικής εργασίας. Επιπλέον, τον ευχαριστώ για τη δυνατότητα που μου έδωσε να πραγματοποιήσω το πείραμα της διπλωματική εργασία στο ερευνητικό εργαστήριο MicroSenses, τον εξοπλισμό που μου παρείχε και τις απαιτούμενες παροχές. Ευχαριστώ τα μέλη του εργαστήριου, τον μεταδιδακτορικό ερευνητή κο Δημήτρη Μπαρμπάκο, το υποψήφιο διδάκτορα κο Απόστολο Αποστολάκη και τον μεταπτυχιακό φοιτητή κο Αλέξανδρο Καλλέργη για την υποστήριξη και τη βοήθεια τους καθ' όλη τη διάρκεια της εκπόνησης της εργασίας. Τον υποψήφιο διδάκτορα κο Τζούλιαν Κούτση για τις χρήσιμες συμβουλές και τη πολύτιμη βοήθεια του στη πραγματοποίηση της πειραματικής διαδικασίας. Θα ήθελα ακόμη να ευχαριστήσω όλους τους φίλους μου που ήταν δίπλα μου τα χρόνια των προπτυχιακών σπουδών. Τέλος ευχαριστώ την οικογένεια μου για την οικονομική στήριξη, την υπομονή και την κατανόηση τους όλα αυτά τα χρόνια.

Περίληψη

Στόχος της παρούσας διπλωματικής εργασίας είναι η μελέτη, ο σχεδιασμός και η ανάπτυξη ενός ολοκληρωμένου συστήματος ελέγχου ροής μάζας με διατάξεις μικροελεγκτών. Πιο συγκεκριμένα, το σύστημα που υλοποιήθηκε δέχεται τιμή εισόδου από το χρήστη και εκτελώντας τις ανάλογες παραμετροποιήσεις επιφέρει στη διάταξη τα επιθυμητά επίπεδα ροής. Για την ανάπτυξη του συστήματος, μελετήθηκαν βασικές αρχές λειτουργίας του ελέγχου ροής και έγινε έρευνα αντιστοίχων συστημάτων ελέγχου ροής που υπάρχουν στο εμπόριο και σε ερευνητικό στάδιο.

Τα πειράματα έλαβαν χώρα στις εγκαταστάσεις του ερευνητικού εργαστήριου Μικροσυστημάτων, Αισθητήρων Ενσωματωμένων Διατάξεων και Αυτοματισμού, MicroSenses. Το υλικό που χρησιμοποιήθηκε αποτελείται από: το αναπτυξιακό σύστημα STMF401RE σε συνδυασμό με μια ηλεκτρική σφαιρική βαλβίδα και τον αισθητήρα ροής μάζας M-20SLPM-D της Alicat Scientific. Ένα ICMAX232 level shifter χρησιμοποιήθηκε για την μετατροπή των σημάτων επικοινωνίας μεταξύ του μικροελεγκτή και του αισθητήρα. Η τροφοδοσία ροής του συστήματος πραγματοποιείται από έναν συμπιεστή αέρα που παραχωρήθηκε από το εργαστήριο για τις ανάγκες του πειράματος. Στη συνέχεια, πραγματοποιήθηκαν οι αναγκαίες συνδέσεις για τη δημιουργία του κυκλώματος οδήγησης της σφαιρικής βαλβίδας, καθώς και οι συνδέσεις για την επικοινωνία του αισθητήρα με τον μικροελεγκτή. Τα στοιχεία της σφαιρικής βαλβίδας και του αισθητήρα, επικοινωνούν μεταξύ τους χρησιμοποιώντας ειδικούς σωλήνες με αντοχή σε υψηλές πιέσεις, καθώς και δυο ρυθμιστές πίεσης, με τον πρώτο να τοποθετείται στην είσοδο του αισθητήρα, και τον δεύτερο σε διακλάδωση κατά την είσοδο του συστήματος και πριν από την σφαιρική βαλβίδα.

Μετά τη σύνθεση του υλικού πραγματοποιήθηκαν μετρήσεις σε ένα ευρύ φάσμα τιμών για την βαθμονόμηση της σφαιρικής βαλβίδας. Συγκεκριμένα, ακολουθήθηκαν δυο διαφορετικά είδη μετρήσεων για την πολύπλευρη προσέγγιση του συστήματος. Έπειτα δημιουργήθηκε το κυρίως πρόγραμμα στο αναπτυξιακό STMF401RE για τον έλεγχο του συστήματος του ελεγκτή ροής μάζας. Τέλος, πραγματοποιήθηκαν μετρήσεις για να διαπιστωθεί η συμπεριφορά του συστήματος, ειδικότερα, ο χρόνος που απαιτεί για να σταθεροποιηθεί στην επιθυμητή τιμή, καθώς και η απόκλιση που εμφανίζει σε κάθε τιμή.

Λέξεις – κλειδιά

Ροή, Ελεγκτής Ροής Μάζας, Ενσωματωμένα Συστήματα, μικροελεγκτής

Abstract

The aim of this thesis is the study, design and development of an integrated mass flow control system utilizing microcontroller devices. Specifically, the implemented system accepts an input value from the user and by performing the appropriate parameterizations sets the system to the desired flow levels. For the implementation of the system, the basic principles of flow control operation were taken into consideration. Moreover a study on flow control systems, which are available commercially and used in the research phase, was conducted.

The experiment took place at the MicroSenses, Microsystems, Sensors, Embedded Devices and Automation Research Laboratory. The hardware consisted of: the STM401RE development board combined with an electric ball valve and the Alicat Scientific M-20SLPM-D mass flow sensor. An ICMAX232 level shifter was used to convert the communication signals between the microcontroller and the sensor. The flow supply for the system was provided by an air compressor provided by the laboratory for the needs of the experiment. Furthermore, the necessary connections were made for the fabrication of the ball valve driving circuit and the connections between the sensor and the microcontroller. The ball valve and sensor components communicate with each other using special high-pressure resistant tubes, as well as two pressure regulators, with the first placed at the inlet of the sensor, and the second at a junction at the inlet of the system and before the ball valve.

After the synthesis of the system, measurements were carried out over a wide range of values to calibrate the ball valve. Specifically, two different types of measurements were performed for the multi-faceted approach of the system. Afterwards the main program was created on the development board STM401RE to test the mass flow controller system. Finally, measurements were made to determine the behaviour of the system, in particular, the time it takes to deliver the requested value, as well as the deviation of each value.

Keywords

Flow, Mass Flow Controller, Embedded Systems, microcontroller

Περιεχόμενα

Κατάλογος Πινάκων	10
Κατάλογος Σχημάτων	10
Αλφαβητικό Ευρετήριο	13
ΕΙΣΑΓΩΓΗ	15
Δομή	16
1 ΚΕΦΑΛΑΙΟ 1^ο : ΜΕΤΡΗΣΗ ΚΑΙ ΕΛΕΓΧΟΣ ΡΟΗΣ (FLOW CONTROL)	17
1.1 Ιστορική αναδρομή	17
1.1.1 Θερμικός μετρητής ροής μάζας	17
1.1.2 Θερμικός ελεγκτής ροής μάζας (Mass Flow Controller -MFC)	21
1.2 Βασικές αρχές λειτουργίας	24
1.2.1 Βασικά χαρακτηριστικά και λειτουργία του θερμικού μετρητή ροής μάζας (MFM).....	24
1.2.2 Βασικά χαρακτηριστικά και λειτουργία του θερμικού ελεγκτή ροής μάζας (MFC).....	25
1.2.3 Βασικά χαρακτηριστικά και λειτουργία του θερμικού ελεγκτή μάζας υγρών (MFC)	31
1.3 Εμπορικές εφαρμογές	35
1.4 Ερευνητικές εφαρμογές	36
1.4.1 Χρήση των MFCs για περιβαλλοντικές συνθήκες.....	36
1.4.2 Χρήση των MFCs στο διάστημα	37
1.4.3 Άλλες εφαρμογές.....	37
2 ΚΕΦΑΛΑΙΟ 2^ο : ΔΟΜΗ ΚΑΙ ΑΡΧΗ ΛΕΙΤΟΥΡΓΙΑΣ ΤΟΥ ΣΥΣΤΗΜΑΤΟΣ ΠΡΟΣ ΥΛΟΠΟΙΗΣΗ	38
2.1 Επικοινωνία μέσω του διαύλου RS-232	38
2.1.1 Εισαγωγή στο πρωτόκολλο RS-232 (EIA-232)	38
2.1.2 Σήματα και Συνδέσεις (Προδιάγραφες πρωτοκόλλου)	39
2.1.3 Τα χαρακτηριστικά του πρωτοκόλλου RS-232	40
2.2 Ο Μικροελεγκτής STM32F401RE	42
2.2.1 Εισαγωγή στο STM32F401RE.....	42
2.2.2 Βασικά χαρακτηριστικά.....	43
2.3 Ο Αισθητήρας ροής μάζα M-20SLPM-D της εταιρίας Alicat Scientific	46
2.3.1 Προδιάγραφες του αισθητήρα M-20SLPM-D.....	46
2.3.2 Συνδέσεις αισθητήρα (Pin out).....	47
2.3.3 Βασικές ρυθμίσεις	48
2.3.4 Συμβολοσειρά δεδομένων (Data String)	50
2.3.5 Status messages	51
2.4 IC MAX232 (RS232 - TTL Level Shifting)	51
2.4.1 Η λειτουργία του IC MAX232	52
2.5 Ηλεκτρική σφαιρική βαλβίδα	53
2.5.1 Ενεργοποιητής.....	53
2.5.2 Σφαιρική βαλβίδα ροής	54
3 ΚΕΦΑΛΑΙΟ 3^ο : ΜΕΘΟΔΟΛΟΓΙΑ ΠΟΥ ΑΚΟΛΟΥΘΗΘΗΚΕ	56
3.1 Κυκλωματική διάταξη ελεγκτή ελέγχου ροής	56
3.1.1 Κυκλωματική διάταξη μικροελεγκτή-αισθητήρα.....	56
3.1.2 Κύκλωμα ελέγχου ηλεκτρικής σφαιρικής βαλβίδας με χρήση Relays.....	57
3.2 STM32CubeIDE	59
3.2.1 Εισαγωγή στο STM32CubeIDE	59
3.2.2 Δημιουργία περιβάλλοντος επικοινωνίας με τον αισθητήρα.....	60
3.3 Μετρήσεις	62
3.3.1 Παροχή	62

3.3.2	Παράγοντες που προέκυψαν στη διαδικασία των μετρήσεων.....	63
3.3.3	Καθορισμός δυναμικής απόκρισης της βαλβίδας.....	64
3.3.4	Καθορισμός ροής μέσο ελέγχου της βαλβίδας.....	67
3.4	Κύριο πρόγραμμα.....	71
3.4.1	Ρουτίνα εύρεση της επιθυμητής πλευράς.....	72
3.4.2	Πρώτο σκέλος του προγράμματος υπολογισμού.....	73
3.4.3	Δεύτερο σκέλος πρόγραμμα υπολογισμού (εντός του ατέρμονου βρόχου "while(1)").....	74
3.4.4	Αλληλεπίδραση αναμεσα στα δυο σκέλη.....	75
4	ΚΕΦΑΛΑΙΟ 4^ο : ΑΠΟΤΕΛΕΣΜΑΤΑ, ΕΠΑΛΗΘΕΥΣΗ ΛΕΙΤΟΥΡΓΙΑΣ.....	77
4.2	Μετρήσεις επαλήθευσης λειτουργίας.....	77
4.2.1	Ακρίβεια.....	78
4.2.2	Χρονική απόκριση.....	78
5	ΣΥΜΠΕΡΑΣΜΑΤΑ ΚΑΙ ΜΕΛΛΟΝΤΙΚΕΣ ΕΠΕΚΤΑΣΕΙΣ.....	80
5.1	Συμπερασματα.....	80
5.2	Μελλοντικές επεκτάσεις.....	80
	Βιβλιογραφία.....	82
	Παράρτημα Α.....	87
	Παράρτημα Β.....	90
	Παράρτημα Γ.....	93
	Παράρτημα Δ.....	97

Κατάλογος Πινάκων

Πίνακας 1: Συντελεστές μετατροπής IPA και άλλων υγρών – Πηγή [14].....	34
Πίνακας 2: RS-232 Baud Rates - Πηγή [39].....	41
Πίνακας 3: Βασικά χαρακτηριστικά μεγέθη STM32F401RE - Πηγή [37].....	43
Πίνακας 4: Κατάτμηση (partition) τομέων μνήμης flash μικροελεγκτή – Πηγή [48].....	44
Πίνακας 5: Βέλτιστη απόδοση του αισθητήρα M-20SLPM-D - Πηγή [53].....	46
Πίνακας 6: 8-Pin Mini-DIN Pinout - Πηγή [54].....	47
Πίνακας 7: Μαθηματική προσέγγιση μοντέλου ExpGrow2.....	71
Πίνακας 8: Πίνακας αποτελεσμάτων των δοκιμαστικών μετρήσεων, περιλαμβάνει επιθυμητή τιμή και χρόνο απόκρισης.....	77

Κατάλογος Σχημάτων

Σχήμα 1.1: Ο μετρητής ροής μάζας που κατασκεύασε ο Thomas- Πηγή [5].....	17
Σχήμα 1.2: Ο μετρητής ροής μάζας που κατασκεύασε ο Laub-Πηγή[6].....	18
Σχήμα 1.3: Σχηματική απεικόνιση του MFM του Benson-Πηγή [7].....	19
Σχήμα 1.4: Σχηματική απεικόνιση μέτρησης ροής αισθητήρα MFM - Πηγή [2].....	20
Σχήμα 1.5: Σχεδιάγραμμα διαμόρφωσης παράκαμψης αισθητήρα MFM – Πηγή [8].....	20
Σχήμα 1.6:Σχηματική αναπαράσταση της θερμική βαλβίδας – Πηγή [2].....	21
Σχήμα 1.7: Σχηματική αναπαράσταση της βαλβίδας μέτρησης – Πηγή [2].....	22
Σχήμα 1.8: Σχηματική αναπαράσταση ελέγχου ροής αερίου με ηλεκτρομαγνητική βαλβίδα δύο σταδίων – Πηγή [2].....	23
Σχήμα 1.9: MFC με ονομασία EL-FLOW της εταιρείας Bronkhost – Πηγή [13].....	26
Σχήμα 1.10: Διάφοροι τρόποι διασύνδεσης και επικοινωνίας MFCs– Πηγή [13].....	27
Σχήμα 1.11: Προφίλ θερμοκρασίας αισθητήρα στον τριχοειδή σωλήνα – Πηγή [13].....	28
Σχήμα 1.12: Ηλεκτρομαγνητική βαλβίδα ελέγχου – Πηγή [13].....	29
Σχήμα 1.13:Τυπική συνάρτηση μεταφοράς ηλεκτρομαγνητικής βαλβίδας ελέγχου για διάφορα αέρια – Πηγή [13].....	30
Σχήμα 1.14:Βέλτιστες ρυθμίσεις ελεγκτή για διάφορα αέρια ως συνάρτηση της (α) απόλυτης ροής βαλβίδας, (β) σχετικής ροής βαλβίδας – Πηγή [13].....	30

Σχήμα 1.15:Σχηματική απεικόνιση MFC και αισθητήρα MFC – Πηγή [14].....	31
Σχήμα 1.16: Σχηματική αναπαράσταση του ηλεκτρονικού κυκλώματος ενός MFC – Πηγή [15].....	33
Σχήμα 1.17:Μπλοκ διάγραμμα βρόγχου ελέγχου MFC – Πηγή [14].....	35
Σχήμα 2.1:D-25, D-9 pin out- Πηγή [37].....	39
Σχήμα 2.2:Συνδεσμολογία μεταξύ DTE και DCE - Πηγή [34].....	40
Σχήμα 2.3:Απεικόνιση της δομής ενός frame- Πηγή [40].....	41
Σχήμα 2.4:Απεικόνιση του ολοκληρωμένου STM32F401RE- Πηγή [50].....	42
Σχήμα 2.5:Χαρτογράφηση μνήμης flash (flash memory mapping) μικροελεγκτή STM32F401RE - Πηγή [48].....	45
Σχήμα 2.6:Απεικόνιση 8-Pin Mini-DIN στο πάνω μέρος του αισθητήρα - Πηγή [54].....	47
Σχήμα 2.7:Αντιστοίχιση των ακροδεκτών του DB9 σε Mini-Din - Πηγή [54].....	48
Σχήμα 2.8:Η κωδικοποιημένη συμβολοσειράς όπως εμφανίζεται στο σειριακό τερματικό του H/Y - Πηγή [54].....	50
Σχήμα 2.9:Απεικόνιση της συμβολοσειράς με δεκαδικούς αριθμούς.....	50
Σχήμα 2.10:Τα μνήματα καταστάσεις που χρησιμοποιεί ο αισθητήρας M-20SLPM-D - Πηγή [54].....	51
Σχήμα 2.11:Κυκλωματική απεικόνιση του MAX232 - Πηγή [57],[58].....	52
Σχήμα 2.12:Απεικόνιση εσωτερικής λειτουργίας του MAX232 - Πηγή [58].....	53
Σχήμα 2.13:Σχεδιάγραμμα συνδεσμολογίας ενεργοποιητή ηλεκτρικής σφαιρικής βαλβίδας - Πηγή [59].....	54
Σχήμα 2.14:Απεικόνιση σφαιρικής βαλβίδας ροής - Πηγή [60].....	54
Σχήμα 3.1:Μπλοκ διάγραμμα STM32F401RE-MFM.....	56
Σχήμα 3.2:Κυκλωματική διάταξη STM32F401RE-MFM.....	57
Σχήμα 3.3:Κυκλωματική διάταξη συστήματος ελεγκτή ροής μάζας.....	58
Σχήμα 3.4:Στιγμιότυπο από το περιβάλλον STM32CubeIDE.....	59
Σχήμα 3.5:Μπλόκ διάγραμμα για το πρόγραμμα περιβάλλοντος επικοινωνίας με τον αισθητήρα.....	60
Σχήμα 3.6:Μπλόκ διάγραμμα για το πρόγραμμα του καθορισμού δυναμικής απόκρισης της βαλβίδας	64

Σχήμα 3.7:Γραφική απεικόνιση ροής για μεταβολές βαλβίδας με βήμα 50ms σε συνάρτηση με το χρόνο (ανά χρονικά διαστήματα 10-15sec).....	65
Σχήμα 3.8:Γραφική απεικόνιση ροής για μεταβολές βαλβίδας με βήμα 35ms σε συνάρτηση με το χρόνο (ανά χρονικά διαστήματα 10-15sec).....	66
Σχήμα 3.9:Γραφική απεικόνιση ροής για μεταβολές βαλβίδας με βήμα 25ms σε συνάρτηση με το χρόνο (ανά χρονικά διαστήματα 10-15sec).....	66
Σχήμα 3.10:Μπλόκ διάγραμμα για το πρόγραμμα του καθορισμού ροής μέσω ελέγχου της βαλβίδας	68
Σχήμα 3.11:Γραφική απεικόνιση ροής συνάρτηση του χρόνο, σύμφωνα με τη μεθοδολογία καθορισμού ροής μέσω ελέγχου της βαλβίδας	69
Σχήμα 3.12:Γραφική απεικόνιση χρόνου συνάρτηση της ροής, σύμφωνα με την δεύτερη προσέγγιση προσδιορισμού της ροής	70
Σχήμα 3.13:Μπλόκ διάγραμμα του κυρίως προγράμματος.....	72

Αλφαβητικό Ευρετήριο

ADC	Analog to D igital C onverter
ARM	Acorn R ISC M achine
ASCII	A merican S tandard C ode for I nformation I nterchange
CF	Conversion F actor,
COD	C oefficient O f D etermination
CTA	C onstant T emperature A nemometry
CTS	C lear T o S end
CCVD	C atalytic C hemical V apour D eposition,
DC	D ata C arrier
DCE	D ata C ircuit- T erm E quipment
DIL	D ual I n L ine
DSR	D ata S et R eady
DTE	D ata T erminal E quipment
DTR	D ata T erminal R eady
EIA	E lectronic I ndustries A ssociation
GPIO	G eneral- P urpose I nput/ O utput
IC	I ntegrated C ircuit
I²C	I nter- I ntegrated C ircuit
I/O	I nput/ O utput
ISO	I nternational O rganization for S tandardization
IPA	I so P ropyl A lcohol
ITU	I nternational T elecommunication U nion
JTAG	J oint T est A ction G roup
L	L ive (wire)
LCD	L iquid C rystal D isplay
LFE	L aminar F low E lement,
MFC	M ass F low C ontroler
MFM	M ass F low M eter
MWNT	M ulti W alled N ano T ubes
N	N eutral (wire)
NVIC	N ested V ector I nterrupt C ontrol
OEM	O riginal E quipment M anufacturer
PIC	P rogrammable I ntegrated C ircuit
PID	P roportional I ntegral D erivative
PPB	P arts P er B illion
PWM	P ulse W idth M odulation
RI	R I n g
RISC	R educed I nstruction S et C omputer
RS	R ecommended S tandard
RTC	R eal T ime C lock
RTS	R equest T o S end

Rx	Receive
SIG	SIG nal
SLPM	Standard Liter Per Minute
SPI	Serial Peripheral Interface
SRAM	Static Random-Access Memory
SWD	Serial Wire Debug
TIA	Telecommunications Industry Association
TTL	Transistor-Transistor Logic
Tx	Transmit
USART	Universal Synchronous/Asynchronous Receiver/Transmitter
USB	Universal Serial Bus

ΕΙΣΑΓΩΓΗ

Η ακριβής μέτρηση της ροής είναι πολύ σημαντική για τις βιομηχανικές διεργασίες παρασκευής, οπότε αναπτύχθηκαν πολλαπλές συσκευές που επιτρέπουν τον λεπτομερή έλεγχο της ροής, ο οποίος εξασφαλίζει ότι τα εργαλεία μέτρησης θα παρουσιάσουν τα μικρότερα δυνατά σφάλματα [1-3]. Ιδιαίτερα στις βιομηχανίες ημιαγωγών και χημικών, η ακρίβεια και ο έλεγχος της ποσότητας των εγγεόμενων υγρών και αερίων αποτελούν κρίσιμους παράγοντες στην παραγωγική διαδικασία. Μέθοδοι μέτρησης όπως αυτή με βάση τον όγκο κρίνεται ακατάλληλη για το σύνολο των αερίων, καθώς η μεταβολή της θερμοκρασίας ή της πίεσης επηρεάζουν την πυκνότητά τους. Ο συνδυασμός των μετρήσεων του ρυθμού ροής του όγκου και της πυκνότητας σε ίδιο χρόνο, καθορίζουν τη μέτρηση της ροής μάζας ενός αερίου [2].

Η βασική λειτουργία των ελεγκτών ροής μάζας είναι ότι ελέγχουν την ροή αερίων και υγρών, εξασφαλίζοντας ακριβείς και αξιόπιστες μετρήσεις, βοηθώντας σημαντικούς κλάδους όπως: τη χημική βιομηχανία, τη βιομηχανία ημιαγωγών καθώς και σε διαστημικές εφαρμογές. Το χαρακτηριστικό αυτό συμβάλλει στον αποτελεσματικό έλεγχο της διαδικασίας παραγωγής, στη μείωση της οποίας απώλειας ή ατέλειας στην παραγωγή, εξασφαλίζοντας έτσι την βέλτιστη ποιότητα στο τελικό προϊόν. Χρησιμοποιούνται επίσης ευρέως σε πολλούς επιστημονικούς τομείς όπως, στην ηλεκτρονική, σε τομείς ενέργειας, τη φαρμακευτική και τη βιοτεχνολογία, επιτελώντας μερικές από τις ακόλουθες λειτουργίες [2]:

- **Ακρίβεια στη διαχείριση της ροής υγρών και αερίων:** Χρησιμοποιώντας MFCs, επιτυγχάνεται μεγαλύτερη ακρίβεια στη ροή αερίων κατά τις διαδικασίες, διασφαλίζοντας έτσι την υψηλή ποιότητα και αποδοτικότητα των παραγόμενων προϊόντων.
- **Μείωση απωλειών και λαθών:** Οι MFCs βοηθούν στη μείωση των απωλειών και των λαθών που μπορεί να προκύψουν κατά τη ροή υγρών και αερίων στις διάφορες παραγωγικές διαδικασίες.
- **Ενεργειακή εξοικονόμηση:** Η παρακολούθηση και ο έλεγχος της ροής υγρών και αερίων με τη χρήση MFCs συμβάλλει στην αποφυγή υψηλότερων ροών από το επιθυμητό, οδηγώντας έτσι σε σημαντική εξοικονόμηση ενέργειας.
- **Αυτοματοποίηση και αύξηση παραγωγικότητας:** Με την ενσωμάτωση των MFCs στις γραμμές παραγωγής, επιτυγχάνεται αυτοματοποίηση των διαδικασιών, μείωση των σφαλμάτων, βελτίωση της ποιότητας των προϊόντων και μείωση του χρόνου παραγωγής.

Δομή

Η διπλωματική εργασία αποτελείται συνολικά από πέντε (5) κεφάλαια:

- Κεφάλαιο 1, Μέτρηση και έλεγχος ροής (flow control): Αναλύονται βασικές αρχές και έννοιες του ελέγχου ροής, γίνεται ερευνά των συστημάτων που υπάρχουν στο εμπόριο και σε ερευνητικά στάδια.
- Κεφάλαιο 2, Δομή και αρχή λειτουργίας του συστήματος: Διαχωρίζονται τα στοιχεία που αποτελούν το σύστημα και αναλύονται τα χαρακτηριστικά και η λειτουργίας τους.
- Κεφάλαιο 3, Μεθοδολογία που ακολουθήθηκε: Περιγράφονται τα βήματα που ακολουθήθηκαν για την υλοποίηση του συστήματος του ελεγκτή ροής μάζας, συμπεριλαμβανομένων, της κυκλωματικής διάταξης του συστήματος, των μετρήσεις και τον προγραμματισμό του μικροελεγκτή.
- Κεφάλαιο 4, Αποτελέσματα, επαλήθευση λειτουργίας: Διεξήχθησαν αναλυτικές μετρήσεις για να διαπιστωθεί η σωστή λειτουργία και οι επιδόσεις του συστήματος.
- Κεφάλαιο 5, Συμπεράσματα και μελλοντικές επεκτάσεις: Παρουσιάζεται μια σύνοψη της δουλειάς και πιθανές ιδέες για μελλοντική εργασία.

1 ΚΕΦΑΛΑΙΟ 1^ο : ΜΕΤΡΗΣΗ ΚΑΙ ΕΛΕΓΧΟΣ ΡΟΗΣ (FLOW CONTROL)

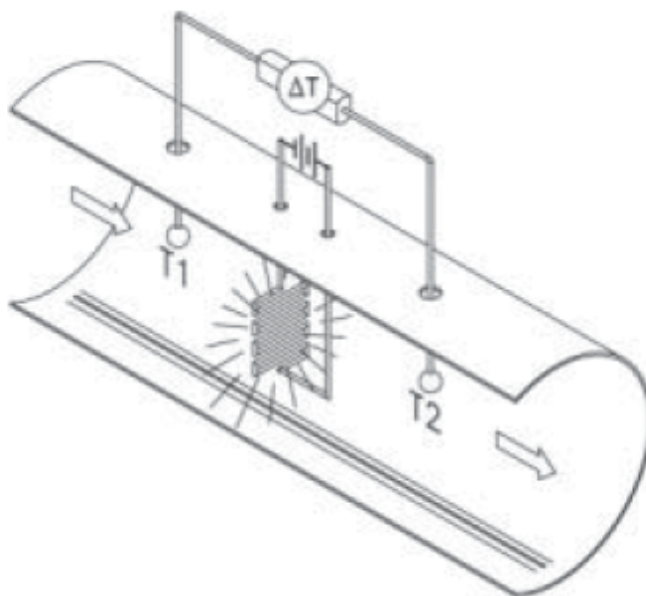
1.1 Ιστορική αναδρομή

Η ιστορική αναδρομή αποτελεί θεμέλιο πυλώνα για την κατανόηση της εξέλιξης της μηχανικής και της τεχνολογίας. Σε αυτή την ενότητα γίνεται ιστορική ανασκόπηση του θερμικού μετρητή ροής μάζας καθώς και η μετατροπή του αργότερα σε θερμικό ελεγκτή ροής μάζας με τη χρήση διαφόρων τύπων βαλβίδων.

1.1.1 Θερμικός μετρητής ροής μάζας

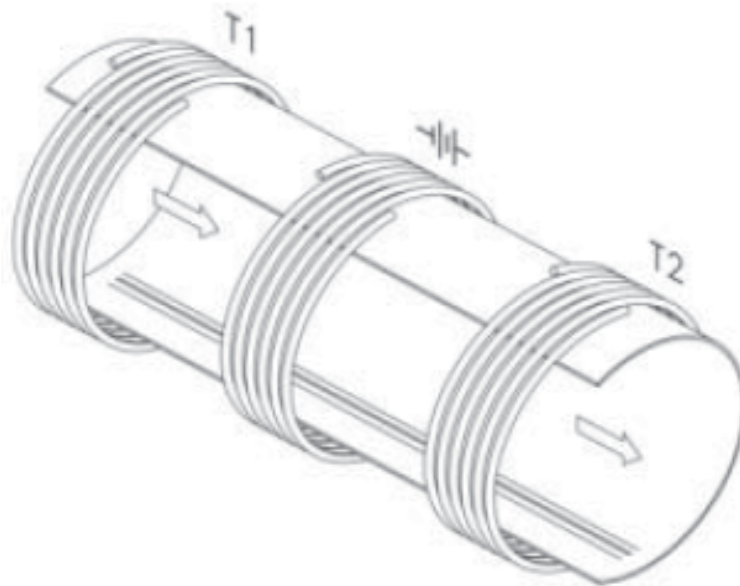
Οι θερμικοί ελεγκτές ροής μάζας δεν αποτελούν καινοτομία. Η αρχική ιδέα προήλθε από τον Thomas στις αρχές του 20ου αιώνα όπου στην εργασία του, περιέγραψε έναν ελεγκτή ροής (ροόμετρο) που περιείχε ένα θερμαντικό στοιχείο με δύο θερμομέτρα ανάντη και κατόντη της ροής του (Σχήμα 1.1) [4]. Μετρούσε τον ρυθμό ροής μάζας (M) του αερίου υπολογίζοντας το κλάσμα της θερμότητας στην είσοδο (H) προς τη χωρητικότητα θερμότητας (Cv) του αερίου επί την διαφορά της αύξηση θερμοκρασίας (Δt) , όπως αναγράφεται στην παρακάτω εξίσωση:

$$M = \frac{H}{Cv \cdot \Delta t}$$



Σχήμα 1.1. Ο μετρητής ροής μάζας που κατασκεύασε ο Thomas- Πηγή [5]

Ο μετρητής αυτός δεν μπήκε ποτέ στην παραγωγή [4]. Τριάντα (30) περίπου χρόνια αργότερα ο Laub πρότεινε την προσαρμογή του θερμαντικού στοιχείου και των θερμομέτρων έξω από την σωλήνα (Σχήμα 1.2).

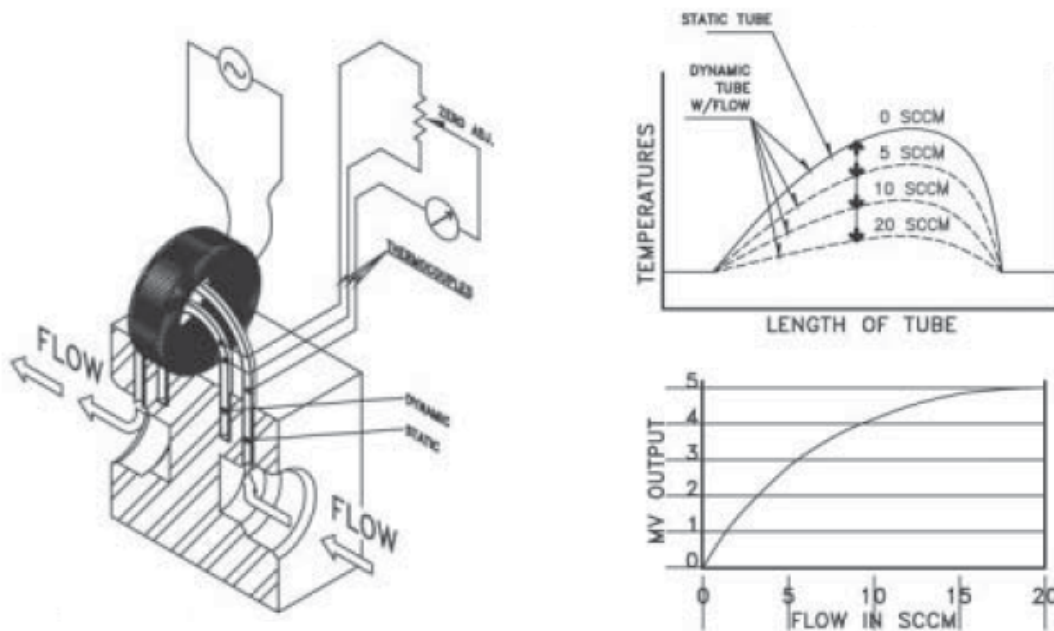


Σχήμα 1.2. Ο μετρητής ροής μάζας που κατασκεύασε ο Laub - Πηγή [6]

Παρόλο που φαινόταν ότι τα αποτελέσματα μέτρησης ήταν ικανοποιητικά κατά τη διεξαγωγή πειραμάτων (υπό ιδανικές συνθήκες), οι μετρήσεις του εμφάνιζαν αστάθεια όταν υπολογιζόντουσαν σε πραγματικές συνθήκες [6],[4]. Ο Thomas και ο Laub εκμεταλλεύτηκαν την ιδιότητα ενός ρεύματος αερίου να μεταφέρει θερμότητα μεταξύ δυο σημείων, προκειμένου να μετρήσουν τον ρυθμό ροής του. Για να το πετύχουν αυτό, εισήγαγαν θερμότητα με σταθερό ρυθμό σε ένα αέριο που ρέει κατά μήκος ενός σωλήνα και κατέγραψαν τις θερμοκρασίες πριν και μετά τη θέρμανση [5],[6]. Ωστόσο, οι μετρητές ροής τους είχαν το μειονέκτημα ότι καταλάωναν υψηλή ποσότητα ενέργειας για να καταφέρουν θερμάνουν όλη τη ποσότητα του αερίου καθώς επίσης και σωλήνες μεγάλου μήκους για να επιτευχθεί ομοιόμορφη θερμοκρασία [7].

Το 1965, ο Benson αντικατέστησε τον προϋπάρχοντα σωλήνα με ένα μικρό τριχοειδή σωλήνα. Όπως και στη συσκευή του Laub και του Thomas, θέρμανε το περίβλημα του σωλήνα, προκειμένου να πραγματοποιηθούν οι απαραίτητες μετρήσεις. Στη προκομένη περίπτωση, λόγω της πολύ μικρής εσωτερικής διάμετρού το ρεύμα ροής θερμαινόταν πλήρως, χρησιμοποιώντας πολύ λιγότερη ενέργεια. Είναι σημαντικό να αναφερθεί ότι, η διάμετρος του σωλήνα ήταν εκατό (100) φορές μικρότερη από το μήκος του, γεγονός που επέτρεπε στο αέριο να αναμιγνύεται πλήρως και να αποκτά ομοιόμορφη θερμοκρασία. Για να επιτύχει αυτό χρησιμοποιήθηκαν δύο (2) διαφορετικών ειδών σωλήνες. Ο πρώτος βασικός σωλήνας χρησιμοποιήθηκε για τη μέτρηση της ροής, ενώ ο δεύτερος σωλήνας για να παρέχετε στο σύστημα αναφορά μηδενικής ροής, όπως φαίνεται στο Σχήμα 1.3. Οι σωλήνες λειτουργούσαν ως μέρος του κυκλώματος ενός μετασχηματιστή, θερμαίνονταν δηλαδή επαγωγικά και οι μετρήσεις της θερμοκρασίας γίνονταν με θερμοστοιχεία.

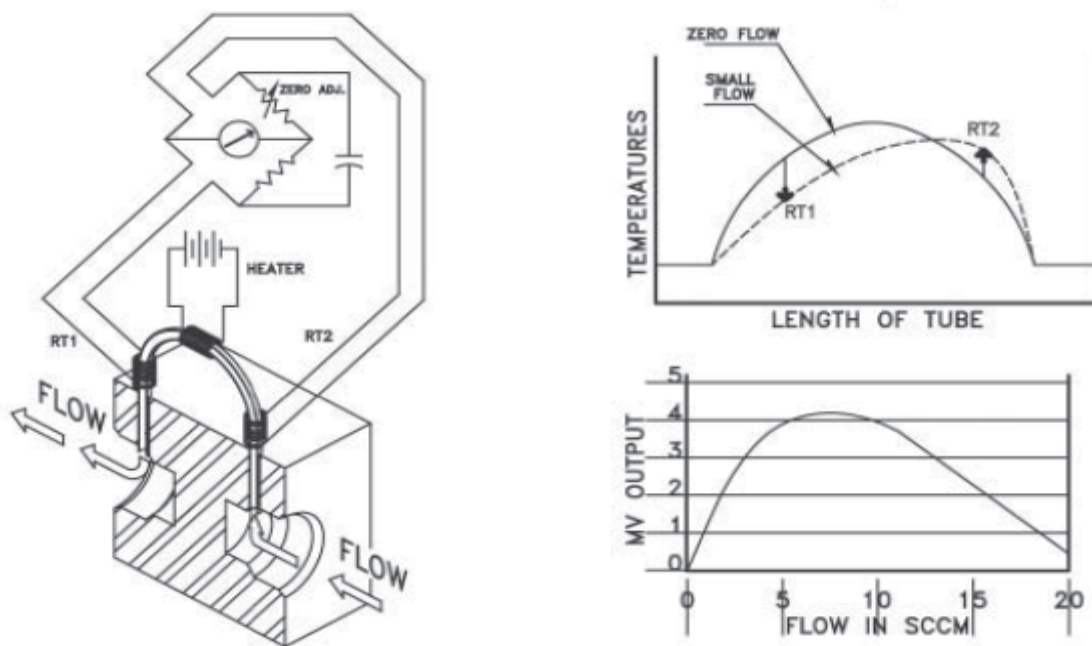
Ο ερευνητής απέδειξε ότι ο μετρητής μπορούσε να καταγράψει ροή από συνθήκες κενού μερικών Torr έως και χίλια (1000) psig και σε εύρος θερμοκρασιών από μείον είκοσι (-20) έως και εκατό (100) βαθμούς κελσίου (°C).



Σχήμα 1.3. Σχηματική απεικόνιση του MFM του Benson - Πηγή [7]

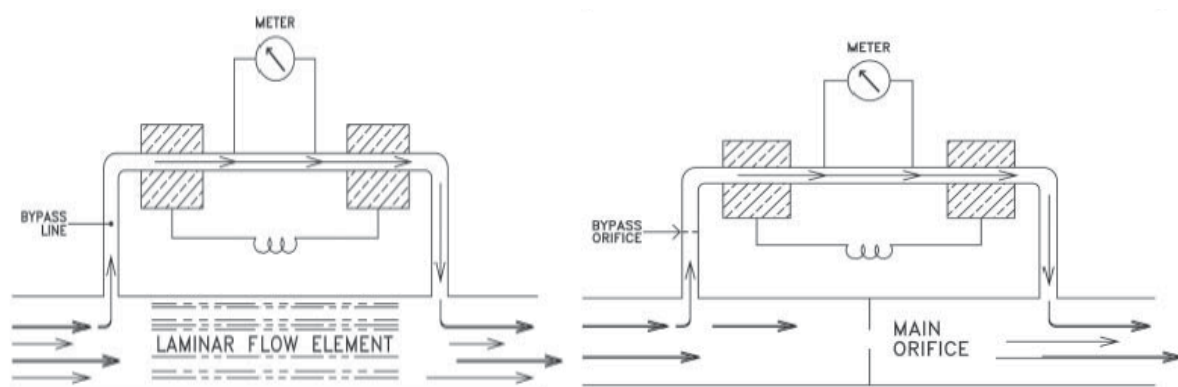
Αρκετοί μηχανικοί προσπαθήσαν να κατασκευάσουν παρομοίους μετρητές ροής μάζας εκείνη την εποχή. Με ποιο δημοφιλή προσέγγιση εκείνη του μονού τριχοειδή σωλήνα που περιείχε αντίσταση τυλιγμένη γύρω από το κέντρο του, ενώ περιλάμβανε θερμοστοιχεία τυλιγμένα στην αρχή και στο τέλος του περιβλήματος του. Τα μηχανήματα μέτρησης τοποθετήθηκαν στην ίδια απόσταση από την αντίσταση, επιτρέποντας στις θερμοκρασίες τους να είναι ίσες σε περίπτωση μηδενικής ροής. Συγκεκριμένα, κατά την παρουσίαση ροής, η θερμοκρασία του αισθητήρα στο άνω μέρος μειώνεται καθώς το αέριο αφομοιώνει τη θερμότητα του σωλήνα. Καθώς το αέριο διασχίζει το τμήμα της αντίστασης, φτάνει σε μια μέγιστη θερμοκρασία και αρχίζει να επιστρέφει τη θερμότητα αυτή πίσω στον σωλήνα, με αποτέλεσμα η θερμοκρασία του αισθητήρα στο κάτω μέρος να αυξηθεί. Τα δύο θερμοστοιχεία συνδέονται σε ένα κύκλωμα γέφυρας, όπου προκύπτει ένα σήμα ανάλογο με το ρυθμό ροής μάζας.

Η πραγματική «επανάσταση» στην κατασκευή των MFMs όμως συντελέστηκε τη δεκαετία του 1970 όπως συνέβη και με άλλες βιομηχανικές διεργασίες από τις εταιρείες Sierra Instruments και Kurz Instruments οι οποίες ανέπτυξαν αισθητήρες πυριτίου σε πολύ μικρή κλίμακα οι οποίοι είχαν ταχύτερη απόκριση σε μικροδευτερόλεπτα [4]. Παράλληλα, αναπτύχθηκαν τεχνικές για γραμμικότερη απόκριση του σήματος εξόδου του θερμοστοιχείου, λόγω του ότι το σήμα γίνεται μη γραμμικό σε μεγάλες ροές. Πραγματοποιήθηκε επίσης επέκταση του εύρους μέτρησης του μετρητή. [2].



Σχήμα 1.4. Σχηματική απεικόνιση μέτρησης ροής αισθητήρα MFM - Πηγή [2]

Πολλοί κατασκευαστές δημιούργησαν συσκευές μέτρησης ροής που υπολόγιζαν τη θερμότητα (ή ισχύ) που ήταν απαραίτητη προκειμένου να διατηρηθεί μια σταθερή διαφορά θερμοκρασίας. Αυτές οι συσκευές παρήγαγαν ένα σήμα εξόδου που ήταν γραμμικά ανάλογο με τη ροή μάζας. Επιπλέον, προσέφεραν γρηγορότερους χρόνους απόκρισης και αύξησαν σημαντικά την εμβέλεια της συσκευής. Οι προηγούμενες εκδόσεις του αισθητήρα είχαν τυπικό εύρος ροής το οποίο κυμαινόταν μεταξύ δέκα (10) και πενήντα (50) sccm. Το γεγονός αυτό, ώθησε τους κατασκευαστές το 1970 να πραγματοποιήσουν εντατικές προσπάθειες για την αύξηση του εύρους του αισθητήρα δοκιμάζοντας διαφορετικές διαμορφώσεις (διαμορφώσεις παράκαμψης)[8]. Στο σχήμα 1.5 απεικονίζεται ένα σχεδιάγραμμα μιας διαμόρφωσης παράκαμψης. Ορισμένες διαμορφώσεις παράκαμψης (Bypass) περιλάμβαναν διάφορους μικρούς διαδρόμους, όπως συντηγμένα μεταλλικά φίλτρα ή στοίβες από φίλτρα πλέγματος που είχαν πλάτος μόνο λίγα μικρόμετρα (μm). Άλλες διατάξεις περιλάμβαναν ένα κωνικό βύσμα ροής σε έναν κωνικό διάδρομο έτσι ώστε να δημιουργηθεί ένα είδος ρυθμιζόμενης ροής.

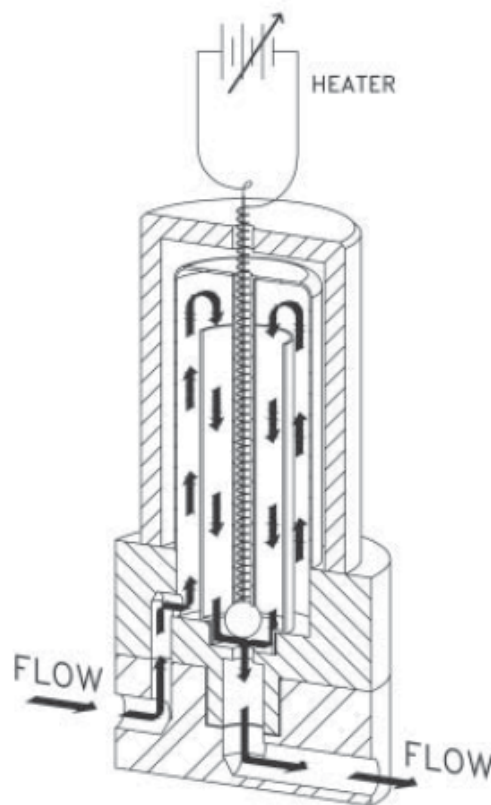


Σχήμα 1.5. Σχεδιάγραμμα διαμόρφωσης παράκαμψης αισθητήρα MFM – Πηγή [8]

1.1.2 Θερμικός ελεγκτής ροής μάζας (Mass Flow Controller -MFC)

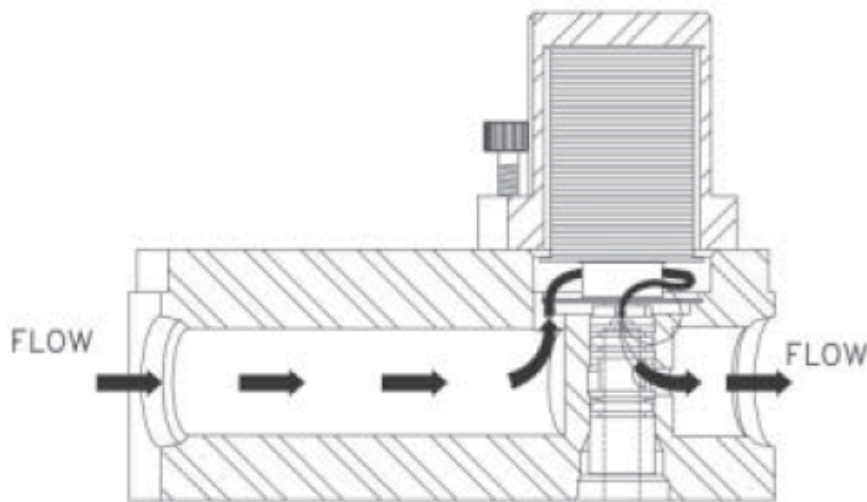
Η μέτρηση της ροής αποτέλεσε ένα αρχικό και σημαντικό βήμα. Παρ'όλα αυτά, οι ανάγκες της βιομηχανίας ήταν ο έλεγχος της ροής. Αρχικά, η ροή ελεγχόταν χειροκίνητα μέσω μηχανοκίνητων δοσομετρικών βαλβίδων. Οι πρώτες προσπάθειες για τον έλεγχο της ροής, έγιναν με την αυτοματοποίηση της ήδη υπάρχουσας δοσομετρικής βαλβίδας. Η πρακτική αυτή ήταν ένας αποτελεσματικός τρόπος ελέγχου της ροής γεγονός που έδωσε λύση στο γενικό πρόβλημα. Σημαντικός παράγοντας αποτέλεσε το γεγονός ότι το μέγεθος της μηχανοκίνητης δοσομετρικής βαλβίδας σε σχέση με τους μετρητές ροής ήταν εσθήτα δυσανάλογο. Κατά συνέπεια, το σύστημα δεν ανταποκρινόταν στις βιομηχανικές απαιτήσεις για μικρότερους και πιο συμπακνωμένους ελεγκτές ροής μάζας [2], [4].

Η εταιρία Tylan δημιούργησε μια καινοτόμο θερμική βαλβίδα η οποία αποτελούνταν από μια μικρή σφαίρα που αιωρούνταν από έναν σωλήνα που βρισκόταν πάνω από ένα μικρό στόμιο, όπως φαίνεται στο σχήμα 1.6. Ο σωλήνας στήριζε τη σφαίρα πάνω από το στόμιο, επιτρέποντας στη ροή να περνάει μέσα από αυτό. Κατά τη θέρμανση του σωλήνα μέσω ηλεκτρικής αντίστασης (θερμαντήρα), διαστελλόταν, πιέζοντας έτσι τη σφαίρα προς το στόμιο και μειώνοντας τη ροή που μπορούσε να μεταβεί μέσα από το σύστημα.



Σχήμα 1.6. Σχηματική αναπαράσταση της θερμικής βαλβίδας – Πηγή [2]

Το ρεύμα που θερμαίνει τον σωλήνα προκαλεί τη διαστολή του, η οποία με τη σειρά της ρυθμίζει τη ροή μέσω της βαλβίδας. Το μέγεθος του στομίου της βαλβίδας είναι υπεύθυνο για τη μέγιστη εμβέλεια της βαλβίδας. Η νέα βαλβίδα αποτελούσε το ιδανικό μέγεθος (ήταν σχεδιασμένη τόσο μικρή) έτσι ώστε να μπορέσει να ενσωματωθεί στο μετρητή ροής ο οποίος αργότερα έγινε Ελεγκτής Ροής Μάζας. Ήταν εξαιρετικά αποδοτική και αποτελεσματική, με μοναδικό μειονέκτημα τις μιγμένες δυνατότητες ως προς τη διαφορική πίεση. Παρόλα αυτά, οι μικρές και συμπαγείς διαστάσεις της ήταν ακριβώς αυτό που χρειαζόταν η βιομηχανία.

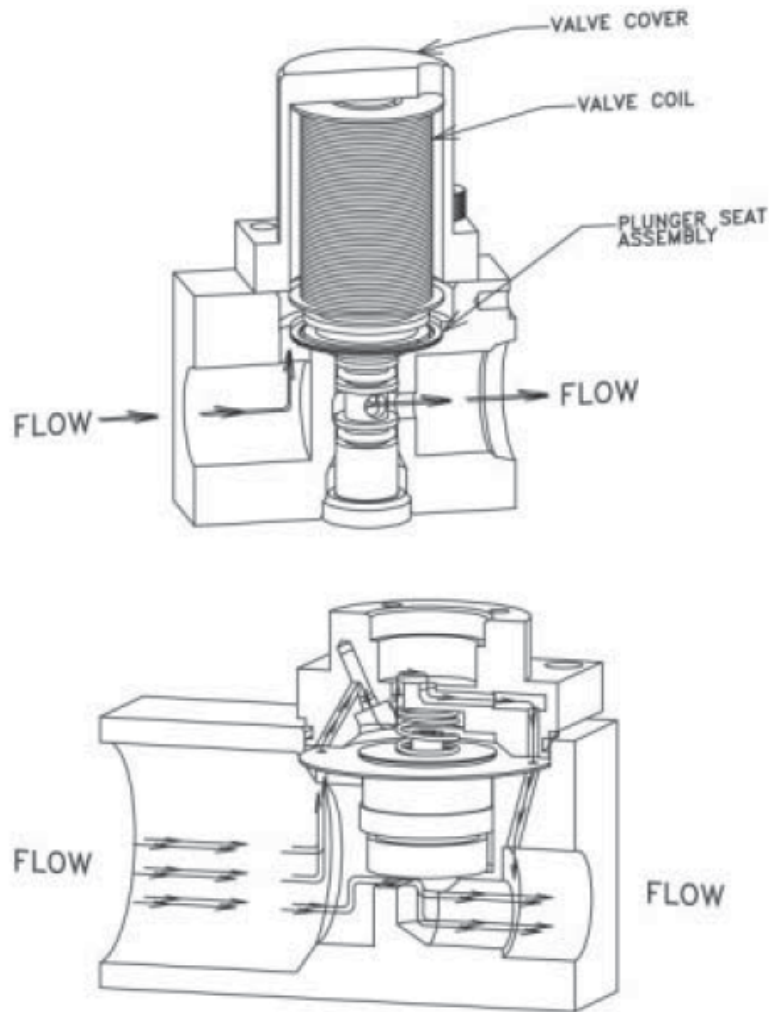


Σχήμα 1.7. Σχηματική αναπαράσταση της βαλβίδας μέτρησης – Πηγή [2]

Για την αντιμετώπιση των διαφορών πίεσης που παρατηρούνται στις βιομηχανικές εφαρμογές, τα βλήματα στρέφοντα στις βαλβίδα με τη λογική "ON-OFF". Η λογική πίσω από αυτό βασίζεται σε ένα πηνίο οπύ ανοιγοκλείνει τη βαλβίδα ηλεκτρομαγνητικά. Πιο αναλυτικά με την ενεργοποίηση του πηνίου, η ηλεκτρομαγνητικής βαλβίδα ασκεί μαγνητική πίεση σε ένα έμβολο, το οποίο κλείνει το στόμιο και σταματά τη ροή. Με την απενεργοποίηση του πηνίου, τα ελατήρια σπρώχνουν το έμβολο μακριά από το στόμιο, επιτρέποντας έτσι τη διέλευση ροής. (Σχήμα 1.7). Η βαλβίδα μπορούσε να ελέγχει τη ροή κρατώντας το έμβολο ακριβώς πάνω από το στόμιο. Πιο συγκεκριμένα, το εύρος λειτουργίας προσδιοριζόταν από τη διατομή του στομίου και την πίεση λειτουργίας, ενώ τα όρια της πίεσης καθορίζονταν από το μέγεθος ισχύς του πηνίου και την αντίσταση που ασκούσαν τα ελατήρια. Αξίζει να αναφερθεί ότι, αν κατά τη λειτουργία προκύψουν υδρατμοί, το σύστημα ατμοποίησης που εμπεριέχεται στη βαλβίδα εξατμίζει το υγρό και ένα θερμαινόμενο σύστημα τοποθετεί τον ατμό σε ιδικό θάλαμο συγκράτησης. Η μέτρηση του υγρού στον ατμοποιητή γίνεται μέσω περισταλτικής αντλίας.

Για μεγαλύτερους ρυθμούς ροής, θα μπορούσαν να χρησιμοποιηθούν δυο ηλεκτρομαγνητικές βαλβίδες διαφορετικών διαστάσεων, όπου η μικρότερη βαλβίδα θα ελέγχει (πνευματικά) τη μεγαλύτερη. Ένα σχετικό παράδειγμα απεικονίζεται στο σχήμα 1.8

Η κύρια ηλεκτρομαγνητική βαλβίδα μέτρησης είναι αποδοτική και μικρού μεγέθους. Αξίζει να σημειωθεί ότι, η ηλεκτρομαγνητική βαλβίδα δύο σταδίων ικανοποιεί τον αρχικό στόχο για τη δημιουργία μιας συμβατής βαλβίδας, διότι είναι σημαντικά μικρότερη σε σχέση με τις παλαιότερες βαλβίδες μηχανικής κίνησης. Αυτές τις μέρες, η πλειονότητα των ελεγκτών ροής μάζας, ενσωματώνουν μια μορφή της ηλεκτρομαγνητικής βαλβίδας για τον έλεγχο ροής.



Σχήμα 1.8. Σχηματική αναπαράσταση ελέγχου ροής αερίου με ηλεκτρομαγνητική βαλβίδα δύο σταδίων – Πηγή [2]

1.2 Βασικές αρχές λειτουργίας

1.2.1 Βασικά χαρακτηριστικά και λειτουργία του θερμικού μετρητή ροής μάζας (MFM)

Η μέτρηση της άμεσης ροής μάζας είναι προτιμότερη από τη χρήση άλλου χαρακτηριστικού μεγέθους διότι, αυτού του τύπου η μέτρηση δεν παρεμβάλλεται αρνητικά από την πίεση, την πυκνότητα, την αγωγιμότητα, τη θερμοκρασία και το ιξώδες [2]. Γεγονός το οποίο, δίνει τη δυνατότητα στα συστήματα μέτρησης ροής μάζας να επικρατούν στην αγορά. Πρόκειται για περίπλοκες μηχανοτρονικές συσκευές, ο σχεδιασμός και η κατασκευή των οποίων ενσωματώνει πολλαπλές τεχνολογίες και επιστημονικούς τομείς όπως θερμοδυναμική, μηχανολογία, θερμική μηχανική, ηλεκτρονική και επιστήμη των υπολογιστών [10]. Επιπλέον, χρησιμοποιούνται ευρέως για διάφορα συστήματα κατασκευαστών πρωτότυπου εξοπλισμού (OEM) και έχουν τη δυνατότητα μέτρησης σε διάφορες τιμές ανάλογα με το μέγεθος της συσκευής, γενικότερα όμως έχουν μεγάλη δυνατότητα ακρίβειας, αξιοπιστίας και σταθερότητας στο χρόνο [2].

Για τον υπολογισμό της ροής, οι θερμικοί μετρητές ροής μάζας καταγράφουν την ποσότητα της μάζας που μεταφέρετε σε ένα σταθερό χρονικό περιθώριο μέσω ενός εγκάρσιου τμήματος [9]:

- Ενός ελεύθερου αγωγού που θα μπορούσε να είναι ένα κανάλι, ένα ποτάμι ή και μία σωλήνωση ή
- Ενός εξαναγκασμένου αγωγού δηλαδή μίας σωλήνωσης με θετική ή αρνητική πίεση.

Η λειτουργία του θερμικού μετρητή ροής μάζας στηρίζεται στη θέρμανση του ρευστού που διαχέεται από τον σωλήνα και τη μέτρηση της θερμότητας του μέσω αισθητήρων που τοποθετούνται πριν και μετρά τον θερμαντήρα. Πιο αναλυτικά, όταν το αέριο περνάει μέσα από τον αισθητήρα, προκαλεί μια μεταβολή στη θερμοκρασία του αισθητήρα, όπου μέσω της μεταβολής αυτής υπολογίζεται η ροή του αερίου. Οι διατάξεις ποικίλουν για αυτού του τύπου τους θερμαντήρες, που συνθέτουν έναν αισθητήρα ροής μάζας. Στην απλή διάταξη, μια μόνο αντίσταση είναι τοποθετημένη στο κέντρο του σωλήνα του αισθητήρα, και δύο θερμοστοιχεία (αισθητήρες θερμοκρασίας) εντοπίζουν την αλλαγή στο θερμοκρασιακό προφίλ λόγω της ροής. Ένας δεύτερος τύπος αισθητήρα αποτελεί ο αισθητήρας διπλής αντίστασης, όπου περιλαμβάνει δύο θερμαντήρες τοποθετημένους σε ένα κύκλωμα γέφυρας Wheatstone, οι οποίοι επίσης αντιδρούν στην αλλαγή του θερμοκρασιακού προφίλ. Μια άλλη διαρρύθμιση υποστηρίζει τρεις θερμαντήρες, με τον κεντρικό θερμαντήρα να προσφέρει το σήμα του αισθητήρα. Πιο αναλυτικά οι τρεις θερμαντήρες διατηρούν την θερμοκρασία τους σταθερή και η τάση που εφαρμόζεται στον μεσαίο θερμαντήρα μεταφράζεται ως το βασικό σήμα εξόδου.

Είναι συχνή τακτική η βαθμονόμηση του αισθητήρα ροής να πραγματοποιείται με ένα συγκεκριμένο τύπο αερίου ενώ κατά τη χρήση του να λειτουργεί με διαφορετικό αέριο. Σε αυτή την περίπτωση, ένας συντελεστής διόρθωσης αερίου καθορίζει την ευαισθησία του δεύτερου αερίου. Αυτό μπορεί να οδηγήσει σε ανακριβή αποτελέσματα, ειδικά για αέρια με υψηλή θερμική διάχυση, όπως το H₂ και το He.

Με αποτέλεσμα, η πρώτη διάταξη όσο αναφορά τη μεταφοράς θερμότητας δεν είναι επαρκής και η διόρθωση που πρέπει να γίνει αναμεσα στα αέρια μπορεί να χρειάζεται τη χρήση μιας συνάρτησης αντί για έναν απλό παράγοντα [1].

1.2.2 Βασικά χαρακτηριστικά και λειτουργία του θερμικού ελεγκτή ροής μάζας (MFC)

Κάθε ελεγκτής ροής μάζας (MFC) είναι ειδικά σχεδιασμένος και βαθμονομημένος ώστε να ελέγχει έναν συγκεκριμένο τύπο υγρού ή αερίου σε ένα καθορισμένο εύρος ροής. Παρόλο που είναι δυνατόν να ρυθμιστεί ώστε να ελέγχει την ροή από μηδενική τιμή έως και στο 100%, συνήθως χρησιμοποιείται για έλεγχο από το 10 έως το 90% ώστε να επιτυγχάνεται καλύτερη ακρίβεια. Μπορεί να είναι αναλογικός ή ψηφιακός ελεγκτής. Οι αναλογικοί ελεγκτές συνήθως περιορίζονται μόνο στο ρευστό ή αέριο που έχουν βαθμονομηθεί αρχικά, ενώ οι ψηφιακοί ελεγκτές παρέχουν τη δυνατότητα ελέγχου για περισσότερα ρευστά ή αέρια [4], [11]. Οι θερμικοί ελεγκτές ροής μάζας είναι μετρητές ροής μάζας μαζί με ένα σύστημα ελέγχου που συμπεριλαμβάνει, βαλβίδες ελέγχου και τα ανάλογα ηλεκτρομηχανολογικά στοιχεία ελέγχου ροής [11]. Τα βασικά μέρη του συστήματος περιλαμβάνουν μια υποδοχή εισόδου και μια υποδοχή εξόδου, έναν μετρητή ροής μάζας και μια βαλβίδα. Βασικό στοιχείο του συστήματος αποτελεί το κλειστό κύκλωμα ελέγχου. Πιο συγκεκριμένα, λαμβάνει μια εντολή από τον χειριστή ή από ένα άλλο κύκλωμα ή computer, συγκρίνει το σήμα αυτό με την ένδειξη του αισθητήρα ροής και ενεργοποιεί τη βαλβίδα προκειμένου να προσαρμοστεί η ροή στο επιθυμητό επίπεδο. Ο ρυθμός ροής ορίζεται ως ποσοστό της βαθμονομημένης ροής πλήρους κλίμακας και εισάγεται στον ελεγκτή ροής μάζας μέσω σήματος τάσης. Οι ελεγκτές ροής μάζας χρειάζεται να έχουν σταθερή πίεση (ένα σταθερό εύρος πίεσης) όσο αναφορά στη παροχή του ρευστού. Πιο αναλυτικά, υψηλότερα επίπεδα πίεσης μπορούν να δημιουργήσουν μη σταθερό ρυθμό ροής ενώ χαμηλότερα επίπεδα πίεσης μπορούν να εμποδίσουν την επίτευξη του σημείου ρύθμισης.

Υπάρχουν αρκετές μεθοδολογίες για τη μέτρηση της ροής, που κατά συνέπεια βοηθούν και στον έλεγχο της. Αυτές περιλαμβάνουν θερμικές, δυνάμεις Coriolis, διαφορική πίεση (ΔP), υπερήχους, διαφορική θερμοκρασία (ΔT), στροβίλους και ηλεκτρομαγνητικές [12]. Η τυπική μορφή ενός θερμικού MFC φαίνεται στο παρακάτω σχήμα (Σχήμα 1.9) [13]. Το μέγεθος των τωρινών ελεγκτών ροής είναι αρκετά μικρό με μέγεθος: 25mm x 80mm x 105mm.

Τα βασικά χαρακτηριστικά για την επιλογή του ιδανικού ελεγκτή ροής για μία εξειδικευμένη διεργασία είναι:

- Ρυθμός ροής
- Τύπος μέσου που θα ελεγχθεί
- Πίεση στην είσοδο και στην έξοδο
- Θερμοκρασία λειτουργίας
- Σήμα στην είσοδο και στην έξοδο

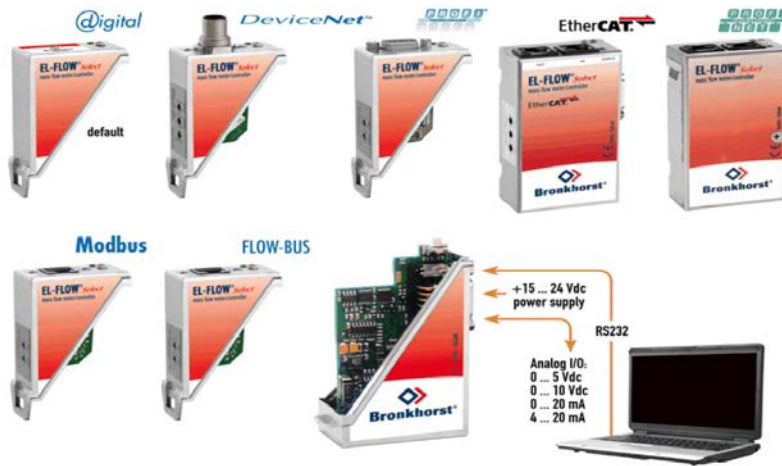
Στα σύγχρονα MFCs, κάποιες τυπικές τιμές για αυτές τις παραμέτρους είναι:

- Ο ρυθμός ροής: 0 - 50 l/min
- Ο τύπος μέσου που θα ελεγχθεί: υγρά και αέρια, με ενδεικτική συμβατότητα για αέρια όπως ο αέρας, το αργόν (Ar), το μεθάνιο (CH₄), το CO, το CO₂, το φρέον (C₂F₆), το ακετυλένιο (C₂H₂), το ήλιο (He), το κρυπτόν (Kr), το υδρογόνο (H₂), η αμμωνία (NH₃), το άζωτο (N₂), το νιτρικό οξείδιο (NO), το υποοξείδιο του αζώτου (N₂O), το οξυγόνο (O₂), το εξαφθοριούχο θείο (SF₆), το σιλάνιο (SiH₄), και το ξένο (Xe).
- Πίεση: έως 100 bar
- Θερμοκρασία λειτουργίας: έως 150 °C
- Σήματα στην είσοδο και στην εξόδο: 0-12V DC, 0-20mA.

Επιπρόσθετος, ένα σημαντικό κομμάτι των ελεγκτών ροής αποτελεί η δυνατότητα επικοινωνίας τους με το εξωτερικό περιβάλλον. Αναλυτικότερα, οι ελεγκτές ροής μπορούν να συνδεθούν με ένα εξωτερικό σύστημα χρησιμοποιώντας αναλογική σύνδεση, με σήματα εισόδου και εξόδου που κυμαίνονται από 0 έως 24V DC, ψηφιακή σύνδεση για επικοινωνία με computer μέσω σειριακής RS-232 ή USB, ή δικτυακή, όπως μέσω Ethernet και διάφορα πρωτόκολλα βιομηχανικού ελέγχου όπως ModBus ή FlowBus, όπως φαίνεται στο παρακάτω σχήμα (Σχήμα 1.10) [13].



Σχήμα 1.9. MFC με ονομασία EL-FLOW της εταιρείας Bronkhost – Πηγή [13]



Σχήμα 1.10. Διάφοροι τρόποι διασύνδεσης και επικοινωνίας MFCs– Πηγή [13]

Βασικά εξαρτήματα MFCs

Ο θερμικός ελεγκτής ροής μάζας αποτελείται από τέσσερα (4) βασικά στοιχεία: i) Το πεδίο στρωτής ροής, ii) Τον αισθητήρα ροής, iii) Την ηλεκτρομαγνητική βαλβίδα ελέγχου και το iv) Το κύκλωμα ελέγχου, το οποίο θα αναλυθεί για τον ελεγκτή ροής μάζας υγρού L20 στην επόμενη υπό-ενότητα.

Πεδίο στρωτής ροής (Laminar flow element, LFE)

Το πεδίο στρωτής ροής αποτελεί το τμήμα της συσκευής στο οποίο η τυρβώδης ροή αερίου μετατρέπεται σε στρωτή ροή. Το εξάρτημα αυτό, αποτελείται από λεπτές πλάκες με πολύ ρηγά αυλάκια και τοποθετείται στη υποδοχή του συστήματος. Σε αυτές τις συνθήκες η πτώση πίεσης είναι ανάλογη με τον ρυθμό ροής όγκου λόγω του χαμηλού αριθμού Reynolds. Η αναλογία ροής μεταξύ του αισθητήρα παράκαμψης και του βασικού σωλήνα ροής καθορίζεται από την αντίσταση που δημιουργεί το πεδίο στρωτής ροής.

Αισθητήρας ροής διπλής περιέλιξης (DTB)

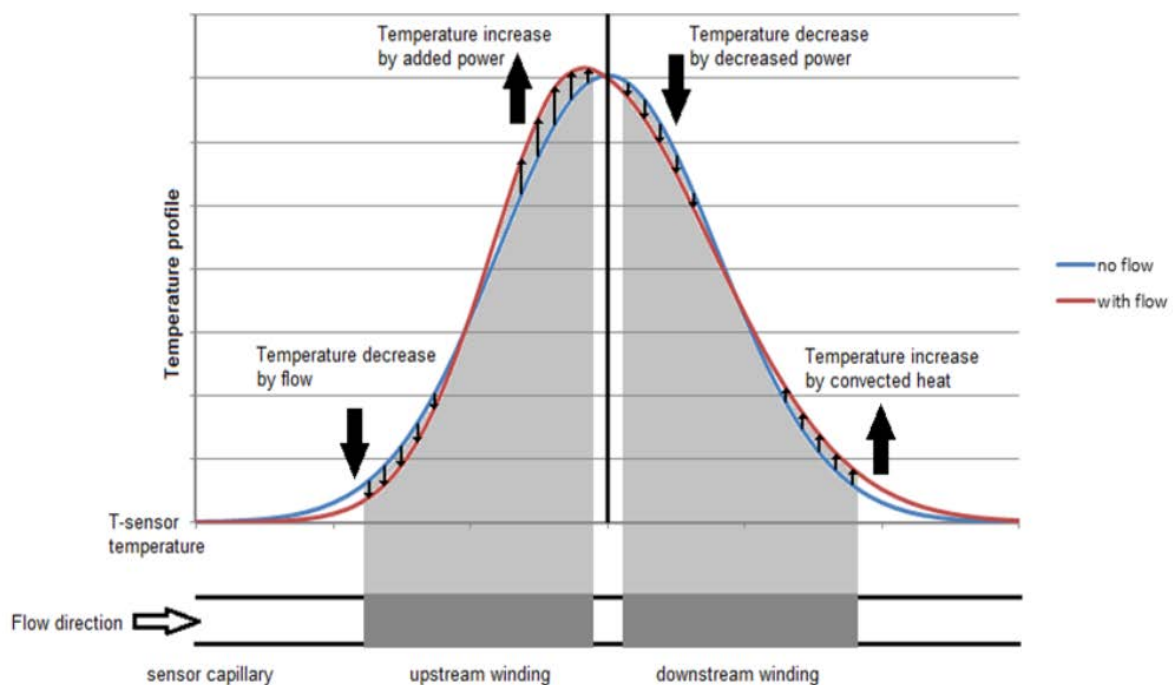
Η δομή ενός αισθητήρα ροής μάζας διπλής περιέλιξης περιλαμβάνει ένα λεπτό τριχοειδή σωλήνα με δυο περιελίξεις για τη θέρμανση και τη μέτρηση του αερίου, καθώς και ένα επιπλέον αισθητήρα θερμοκρασίας έτσι ώστε να λαμβάνονται ποιο σταθερές μετρήσεις. Το πεδίο στρωτής ροής διασφαλίζει ότι ένα μικρό τμήμα της ροής θα διακλαδωθεί και θα περάσει μέσα από τον αισθητήρα. Με τη τροφοδοσία των δυο περιελίξεων ξεκινάει η θέρμανση του σωλήνα. Όσο δεν υπάρχει ροή, παρατηρείται ότι η ισχύς που καταναλώνετε για την ομοιόμορφη θέρμανση των περιελίξεων είναι ισότιμη και για τις δυο. Σε αυτή τη περίπτωση η τιμή του αισθητήρα είναι μηδενική (δεν υπάρχει ροή). Στη περίπτωση όπου υπάρχει ροή, παρατηρείται ότι πέφτει η θερμοκρασία της πρώτης περιέλιξης λόγω του αερίου που εισέρχεται. Για να σταθεροποιηθεί η θερμοκρασία της περιέλιξης η ισχύς που καταναλώνει αυξάνεται. Παράλληλα η ισχύς της δεύτερης περιέλιξης μειώνεται αφού έχει μετρήσει ότι η θερμοκρασία του αερίου έχει αυξηθεί, με αυτό τον τρόπο διατηρείτε μια σταθερή μέση θερμοκρασία και στη δεύτερη περιέλιξη.

Η διαφορά της ισχύος που καταναλώνει κάθε θερμαντήρας για να διατηρήσει μια σταθερή θερμοκρασία αποτελεί ένα άμεσο μέτρο για τον υπολογισμό της ροής (η αρχή της διαφορικής εξισορρόπησης θερμοκρασίας DTB).

Η γραμμική σχέση ανάμεσα στο σήμα εξόδου και του ρυθμού ροής αερίου του αισθητήρα επιτυγχάνει τη βέλτιστη απόδοση στο σύστημα. Για να είναι εφικτό αυτό, θα πρέπει η περιελίξεις να μην ψήχονται και να παραμένουν στην ίδια θερμοκρασία[13]. Στο παρακάτω διάγραμμα (Σχήμα 1.11) απεικονίζεται το σήμα εξόδου του αισθητήρα ροής συναρτήσει της ροής αερίου από τον μετρητή ροής. Και οι δυο περιελίξεις στον αισθητήρα ρυθμίζονται με ισχύ σε $T_{amb} 35^{\circ}C$.

Με την προσθήκη ενός θερμικού αισθητήρα είναι εφικτό να πραγματοποιηθεί σταθεροποίηση στις μετρήσεις του αισθητήρα ροής. Μπορεί να χρησιμοποιηθεί για την επιλογή των αντίστοιχων ιδιοτήτων αερίου, όπως: πυκνότητα, θερμική χωρητικότητα ακόμη και την προσαρμογή των τιμών μέτρησης σε δύσκολες περιβαλλοντικές συνθήκες.

Είναι σημαντικό να αναφερθεί ότι μέσα από προσπάθειες για γίνει γραμμική η σχέση μεταξύ του σήματος εξόδου και του ρυθμού ροής αερίου καθώς και λιγότερο ευάλωτη στις ανοχές, επιτεύχθηκε η μείωση της απόκλισης που υπήρχε μεταξύ των συσκευών μέτρησης ροής. Αυτό είχε σαν αποτέλεσμα, να γίνει ευκολότερος ο εντοπισμός και η αντικατάσταση μιας συσκευής που λάμβανε λανθασμένες μετρήσεις. Από αυτό επωφελούνται χρήστες και κατασκευαστές καθώς δεν χρειάζονται ιδιαίτερες ρυθμίσεις σε όλο το σύστημα και επιπρόσθετος γίνεται μείωση του χρόνου διακοπής λειτουργίας της εγκατάστασης κατά τη συντήρηση.



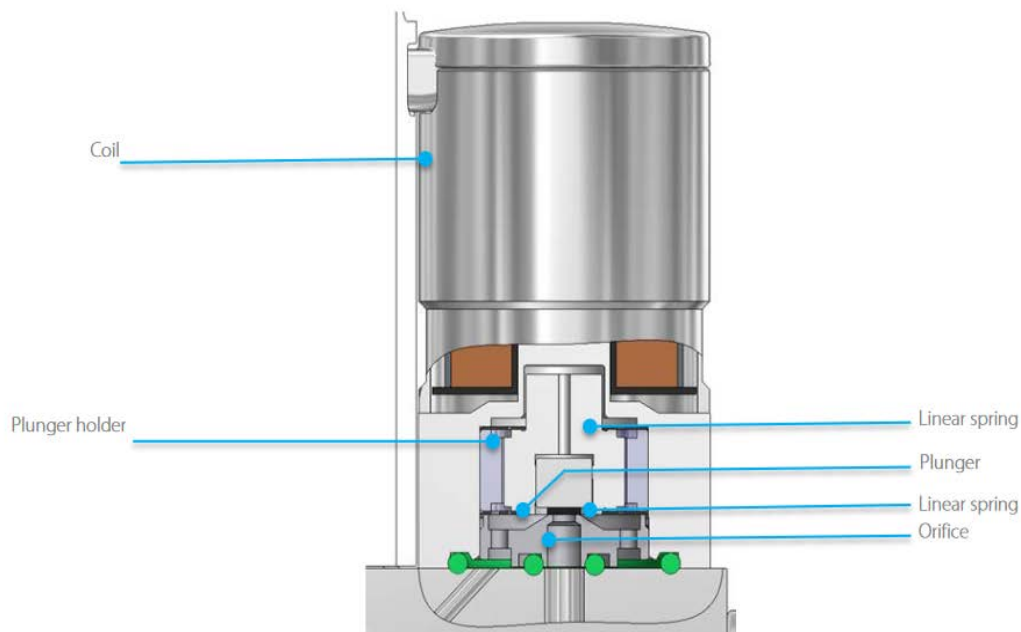
Σχήμα 1.11. Προφίλ θερμοκρασίας αισθητήρα στον τριχοειδή σωλήνα – Πηγή [13]

Ηλεκτρομαγνητική βαλβίδα ελέγχου

Υπάρχει μεγάλη ποικιλία βαλβίδων ελέγχου, με μερικές από αυτές να είναι: θερμικές, ηλεκτρομαγνητικές και πιεζοηλεκτρικές. Η ηλεκτρομαγνητική βαλβίδα ελέγχου χρησιμοποιείται συχνότερα σε συσκευές ελέγχου ροής, έτσι, επιλέχθηκε αυτού του τύπου η βαλβίδα για περαιτέρω ανάλυση.

Η ηλεκτρομαγνητική βαλβίδα ελέγχου (Σχήμα 1.12) αποτελείται από ένα μαγνητικό έμβολο στο οποίο ασκείται δύναμη από γραμμικά ελατήρια, προκειμένου να παραμένει η βαλβίδα σε κλειστή θέση. Ένα πηνίο ασκεί δύναμη στο έμβολο. Η δύναμη αυτή απελευθερώνει το έμβολο από το στόμιο με αποτέλεσμα να υπάρχει ροή στη συσκευή. Για τον έλεγχο της επιθυμητής ροής, αρκεί να υπολογιστεί η αντίστοιχη απόσταση του εμβόλου σε σχέση με το στόμιο. Αυτό επιτυγχάνεται με την ανάλογη ένταση ρεύματος που ασκείται στο πηνίο. Το σήμα από τον αισθητήρα ροής συγκρίνεται με το σημείο ρύθμισης στο κύκλωμα ελέγχου με αποτέλεσμα η ένταση του ρεύματος του πηνίου να υπολογίζεται έτσι ώστε η απόσταση μεταξύ του εμβόλου και του στομίου να είναι ακριβώς η ιδανική για την επιθυμητή ποσότητα ροής [13].

Τα γραμμικά ελατήρια έχουν το ρόλο της ανάρτησης στο σύστημα προσφέροντας ένα πιο ήπιο και χωρίς τριβές έλεγχο στη ροή. Η προσθήκη των ελατήριων δεν επηρεάζει την στεγανότητα της βαλβίδα αντίθετος διασφαλίζει την ομαλή λειτουργία της, εξασφαλίζοντας μεγαλύτερη διάρκεια ζωής.

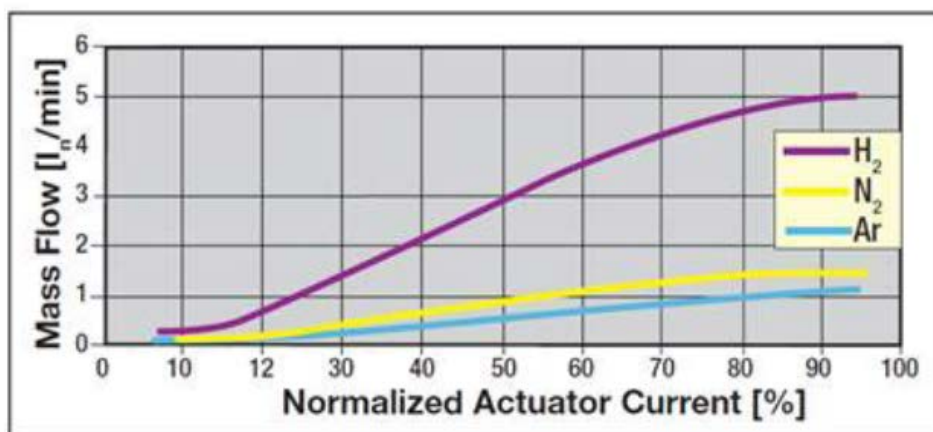


Σχήμα 1.12. Ηλεκτρομαγνητική βαλβίδα ελέγχου – Πηγή [13]

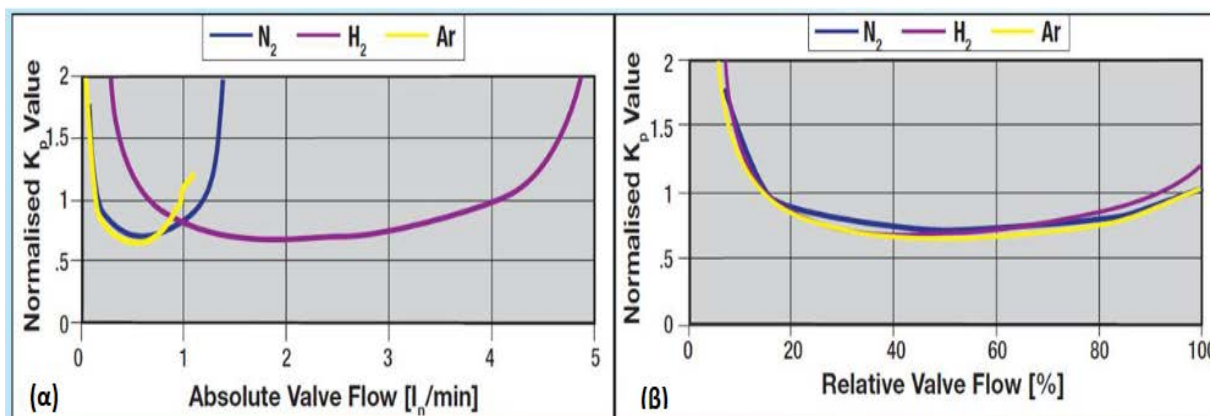
Η γραμμικότητα μεταξύ της σχέσης του ρεύματος του πηνίου και της ροής (απόσταση του εμβόλου σε σχέση με το στόμιο) εξασφαλίζει μεγαλύτερη απόδοση και πιο αξιόπιστες μετρήσεις στο σύστημα.

Η σχεδίαση της διαδρομής της μαγνητικής ροής έχει αναδιαμορφωθεί και συνδυάζεται με ένα μοντέλο συντονισμού που μπορεί να προβλέψει με ακρίβεια πώς θα συμπεριφερθεί η μη γραμμική συνάρτηση μεταφοράς (ρεύμα πηνίου σε σχέση με την απόσταση ανάμεσα στο έμβολο και το στόμιο). Αυτό το μοντέλο χρησιμοποιείται για να ρυθμίσει την επίδραση της μη γραμμικότητας της βαλβίδας στην απόκριση του ελεγκτή.

Όπως απεικονίζεται στα παρακάτω διαγράμματα (Σχήμα 1.13, 1.14), παρόλο που το άνοιγμα της βαλβίδας παραμένει σταθερό, η ποσότητα της ροής μάζας αλλάζει για διαφορετικούς τύπους αερίων. Κατά συνέπεια, η μέγιστη ποσότητα που είναι δυνατόν να περάσει συγκριτικά με τη σχετική ποσότητα αερίου που διέρχεται από το στόμιο είναι πάντοτε ίδια. Δεν πρέπει να γίνεται σύγκριση της πραγματικής ροής, αλλά της σχετικής ροής για διάφορους παράγοντες όπως θερμοκρασία, πίεση και τύπος αερίου. Παρατηρείται ότι οι καμπύλες συμπίπτουν στη μέγιστη ροή βαλβίδας, σε όλες τις άλλες περιπτώσεις απλά εμφανίζουν παρόμοια χαρακτηριστικά.



Σχήμα 1.13. Τυπική συνάρτηση μεταφοράς ηλεκτρομαγνητικής βαλβίδας ελέγχου για διάφορα αέρια – Πηγή [13]

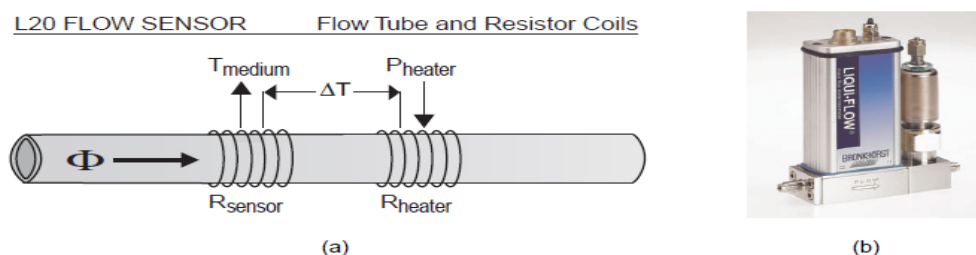


Σχήμα 1.14. Βέλτιστες ρυθμίσεις ελεγκτή για διάφορα αέρια συναρτήσει α) ροή βαλβίδας(απόλυτη τιμή) , β) σχετική ροή βαλβίδας – Πηγή [13]

Ένα εύκαμπτο μονωτικό υλικό είναι τοποθετημένο στο έμβολο προκειμένου να αποφευχθούν τυχόν διαρροές κατά τη λειτουργία της βαλβίδας. Τα μονωτικά υλικά αυτού του τύπου έχουν ένα μερίδιο ευθύνης για την αύξηση της θερμοκρασίας στο σύστημα. Επιπλέον, μετά από κάποιο όριο, υπόκεινται σε διόγκωση. Η μείωση της ποσότητας των μονωτικών υλικών ελαττώνει τα επίπεδα θερμοκρασίας της βαλβίδας. Οδηγώντας, σε καλύτερες επιδόσεις του ελεγκτή PID. Ένα επιπλέον μειονέκτημα αυτών των μονωτικών υλικών είναι ότι, μετά από μεγάλο χρονικό διάστημα που το έμβολο της βαλβίδας παραμένει κλειστό, παραμορφώνονται γύρο από το στόμιο. Με το άνοιγμα της βαλβίδας, το εύκαμπτο υλικό παραμένει για λίγο κολλημένο στο στόμιο, με αποτέλεσμα ο εκλεκτής να δυσκολεύεται να απαλείψει το νεκρό χρονικό διάστημα που μεσολαβεί. Τη λύση στο πρόβλημα έρχεται να δώσει η τοποθέτηση μιας αντικολλητικής επίστρωσης, η οποία στεγανοποιεί εξίσου τη βαλβίδα με λιγότερες παρεμβολές στην ομαλή λειτουργία του συστήματος. Αυτού του τύπου η μεμβράνη είναι χρήσιμη σε εφαρμογές που επαναλαμβάνουν κινήσεις μικρού εύρους, προκειμένου να ελέγχονται μικρές ποσότητες ροής, μετά από τις οποίες το έμβολο ξανά κλείνει. Εν κατακλείδι, το μεγαλύτερο μέρος των αερίων συμμορφώνονται στα πρότυπα στεγανοποίησης, παρ' όλα αυτά πολλά αέρια επιδρούν αρνητικά στο μονωτικό υλικό, το οποίο παραμορφώνεται η αλλοιώνεται με το καιρό. Αυτό έχει σαν αποτέλεσμα, να αναφέρετε στις προδιαγραφές των ελεγκτών ροής μάζας, τα αέρια τα οποία δεν επηρεάζουν τη στεγανοποίηση της βαλβίδας [13].

1.2.3 Βασικά χαρακτηριστικά και λειτουργία του θερμικού ελεγκτή μάζας υγρών (MFC)

Σύμφωνα με τον Lötters, προτάθηκε το 2001 ένας θερμικός μαζικός ελεγκτής ροής υγρών που χαρακτηρίζεται από την ταχύτητα και την εύκολη διαχείριση εύρους τιμών ροής, καλύπτοντας τις βιομηχανικές απαιτήσεις σε ότι αφορά τους ημιαγωγούς, όπου είναι απαραίτητη η υψηλή ακρίβεια. Ο συγκεκριμένος αισθητήρας ροής, του ελεγκτή L20 περιλαμβάνει έναν σωλήνα από ανοξείδωτο χάλυβα, στον οποίο είναι τυλιγμένος ένας θερμαντήρας και ένα πηνίο αντίστασης αισθητήρα, στο Σχήμα 1.15 απεικονίζεται ο αισθητήρας. Ο θερμαντήρας αναπτύσσει μια σταθερή θερμοκρασία θερμαίνοντας το σωλήνα ενώ, η αντίσταση του αισθητήρα μετρά και αντισταθμίζει τις μεταβολές της μέσης θερμοκρασίας. Η απαιτούμενη ισχύς του θερμαντήρα προκειμένου να διατηρηθεί σταθερή θερμοκρασίας ΔT μεταφράζεται σε μετρούμενη ποσότητα ροής. Αυτός ο τρόπος μέτρησης έχει εξαιρετικά μειωμένο χρόνο αποίκισης και είναι γνωστός ως ανεμόμετρά σταθερής θερμοκρασίας (CTA).



Σχήμα 1.15. Σχηματική απεικόνιση MFC και του αισθητήρα του – Πηγή [14]

Πλεονεκτήματα

Στους ελεγκτές ροής μάζας, ο αισθητήρας ροής αποτελείται από έναν σωλήνα κεντρικής ροής κατασκευασμένο από ανοξείδωτο χάλυβα, ένας μικρότερος αισθητήρας χρειάζεται προκειμένου να επιτευχθεί μικρή πτώση πίεσης, καλύτερη χρήση του όγκου και ελάχιστο υπόλειμμα για το περιβάλλον. Το μέγεθος του σωλήνα ροής είναι υπεύθυνο για το εύρος μέτρησης του αισθητήρα. Αλλάζοντας το μέγεθος του σωλήνα ροής, μπορεί να επιτευχθεί ένα ευρύ φάσμα μετρήσεων χρησιμοποιώντας την ίδια συσκευή. Η επιλογή του ανοξείδωτου χάλυβα για την κατασκευή του σωλήνα ροής έχει σαν σκοπό την προστασία των ευάλωτων εξαρτημάτων, τα οποία βρίσκονται εκτός του σωλήνα και δεν υπάρχει κίνδυνος βραχύν. Η διαφορά θερμοκρασίας των δυο στοιχείων του αισθητήρα, διασφαλίζουν ταχιά απόκριση στις μεταβολές της ροής. Ο βρόχος ελέγχου CTA σχεδιάζεται με τέτοιο τρόπο ώστε να παρέχει φυσική αντιστάθμιση θερμοκρασίας. Η βαθμονόμηση του αισθητήρα γίνεται όλο και μικρότερη απαίτηση, αφού η κινήσεις του αισθητήρα μπορούν να προβλεφθούν μέσω υπολογισμών.

Θερμικής μεταφοράς

Η μεταφορά θερμότητας λόγω εξαναγκασμένης μεταφοράς μέσω μιας στρωτής ροής υπολογίζεται με τη σχέση του Graetz (το περίβλημα του σωλήνα είναι εξωθερμικό) [15]:

$$Nu = CGz^{0.333}, Gz > 10, Re < 2300$$

Όταν ο αριθμός Graetz (Gz) είναι μεγαλύτερος από δέκα (10) και ο αριθμός Reynolds (Re) είναι μικρότερος από 2300. Ο αριθμός Nusselt (Nu) αντιπροσωπεύει τον αδιάστατο συντελεστή θερμικής μεταφοράς, η σταθερά C είναι μια συγκεκριμένη τιμή και ο αριθμός Graetz (Gz) ο οποίος διατυπώνεται:

$$Gz = Re \cdot Pr \cdot D/L$$

Ο αριθμός Reynolds (Re) εκφράζει την ροή μάζας, ο αριθμός Prandtl (Pr) είναι αδιάστατος, και D και L αντιπροσωπεύουν τη διάμετρο και το μήκος του σωλήνα αντίστοιχα [15].

Στον L20, η αντίσταση του θερμαντήρα (Rheater) θερμαίνεται σε μια σταθερή διαφορά θερμοκρασίας (ΔT). Η απαιτούμενη ισχύς του θερμαντήρα (Pheater) για τη διατήρηση αυτής της διαφοράς θερμοκρασίας, εκφρασμένη σε βατ (W), εξαρτάται από τη ροή μάζας, διατυπώνεται παρακάτω:

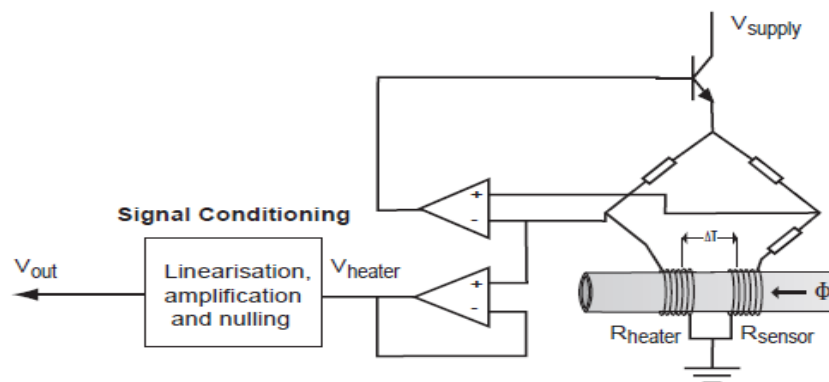
$$P_{heater} = K \cdot Nu \cdot \Delta T$$

Ο K (σε W/°C), όπου συμπεριλαμβάνει το μικρός του θερμαντήρα, του σωλήνα ροής και τις ιδιότητες του υγρού. Με τη χρήση αυτής της εξίσωσης, η συμπεριφορά του αισθητήρα ροής μπορεί να υπολογιστεί για ένα μεγάλο αριθμό υγρών.

Κύκλωμα Αισθητήρα

Το κύκλωμα αποτελείται από δυο βασικούς τομείς: το κύκλωμα ρύθμισης σήματος, που εξασφαλίζει ότι η τάση εξόδου είναι γραμμικά συνδεδεμένη με τη ροή και ο βρόχος ελέγχου ο οποίος διατηρεί σταθερή τη διαφορά θερμοκρασίας, ρυθμίζοντας την ισχύ του θερμαντήρα σε σχέση με τη ροή. Είναι αδιαμφισβήτητο το πιο κρίσιμο κομμάτι του κυκλώματος διότι, εξασφαλίζει ότι ο θερμαντήρας λαμβάνει επιπλέον ρεύμα για να εξισορροπήσει την έλλειψη θερμότητας λόγω της υποχρεωτικής μεταφοράς, όπως αναφέρθηκε και προηγουμένως.

Αυτός ο βρόχος αποτελείται από μια διάταξη γέφυρας Wheatstone με δύο αντιστάσεις (αντιστάσεις ανίχνευσης) και έναν ελεγκτή για την επίτευξη μέγιστης απόκρισης. Η μη γραμμική τάση του θερμαντήρα μετατρέπεται σε σήμα εξόδου (V_{out}) από το κύκλωμα ρύθμισης σήματος και μεταβάλλεται γραμμικά με τη ροή ανάμεσα σε 0 – 5 V. Για να συμβεί αυτό, το κύκλωμα ρύθμισης σήματος περιλαμβάνει: το στάδιο μηδενισμού, της γραμμικοποίησης και της ενίσχυσης. Παρακάτω απεικονίζεται ένα σχεδιάγραμμα του κυκλώματος ελέγχου (Σχήμα 1.16):



Σχήμα 1.16. Σχηματική αναπαράσταση του ηλεκτρονικού κυκλώματος ενός MFC – Πηγή [15]

Σήμα Εξόδου

Όπως αναφέρθηκε προηγούμενος το κύκλωμα ρύθμισης σήματος πρέπει να επιτύχει την έξοδο ενός χρήσιμου γραμμικού σήματος μετατρέποντας της ισχύς του θερμαντήρα (V_{heater}). Αυτό επιτυγχάνεται, σε πρώτο στάδιο μετατρέποντας την σε τάση θερμαντήρα με τη σχέση:

$$V_{heater} = \sqrt{P_{heater} \cdot R_{heater}}$$

Έπειτα, η τάση του θερμαντήρα γραμμικοποιείται και λόγω P_{offset} η τάση μετατόπισης καταργείται. Η γραμμική τάση εξόδου V_{out} που προκύπτει μπορεί να βρεθεί με τη σχέση:

$$V_{(out)} = K \cdot \Phi_m$$

οπού K είναι μια σταθερά (σε V s/kg) η οποία δίνει έμφαση στις διαστάσεις του θερμαντήρα, τον συντελεστή απολαβής του κυκλώματος και τις ιδιότητες του υγρού όπως: λ (σε W/(m K)) η θερμοαγωγιμότητα, c_p (σε J/(kg K)), η ειδική θερμότητα και μ (σε Pa s) το δυναμικό ιξώδες και Φ_m (σε kg/s) η ροή μάζας υγρού.

Συντελεστής Μετατροπής

Ο ελεγκτής ροής υγρού L20 είναι δυνατόν να χρησιμοποιηθεί με πολλούς τύπους υγρών. Λαμβάνοντας υπόψη την τεράστια ποικιλία υγρών που υπάρχουν στο κόσμο, είναι γεγονός ότι αν η συσκευή βαθμονομούταν για κάθε υγρό ξεχωριστά θα ήταν χάσιμο χρήματος και χρόνου. Για αυτό το λόγο η βαθμονόμηση γίνεται μόνο με ένα υγρό (υγρό αναφοράς), ένα τέτοιο υγρό αποτελεί η ισο-προπυλική αλκοόλη (IPA). Έπειτα ο συντελεστής μετατροπής (CF) δίνει τη σχέση ανάμεσα στο υγρό αναφοράς και το υγρό που είναι προς μέτρηση. Με βάση τα όσα αναφέρθηκαν, ο συντελεστής μετατροπής υπολογίζεται με τη σχέση:

$$CF = \frac{V_{out,IPA} | \Phi_m = C}{V_{out,LIQUID} | \Phi_m = C} = \frac{K_{IPA}}{K_{LIQUID}}$$

Στον παρακάτω πίνακα (Πίνακας 1), παρουσιάζονται οι συντελεστές μετατροπής ενός L20 ιδανικό για 1000 g/h IPA και για κάποια άλλα υγρά. Από τον Πίνακα 1, παρατηρούμε ότι ο συντελεστής μετατροπής της μεθανόλης σε σχέση με το IPA είναι 0,62 σε θερμοκρασία T = 20 °C και πίεση p = 1 bar. Συνεπώς, το ισοδύναμο των 1000 g/h IPA είναι ίσο με 1000 × 0,62 = 620 g/h μεθανόλης. Αντίστοιχα, το ισοδύναμο των 1000 g/h μεθανόλης είναι 1000 / 0,62 = 1613 g/h IPA.

ΥΠΟΛΟΓΙΣΜΕΝΟΙ ΣΥΝΤΕΛΕΣΤΕΣ ΜΕΤΑΤΡΟΠΗΣ*

L20 MFC κατάλληλο για

1000 g/h IPA, T = 20°C, p = 1bar

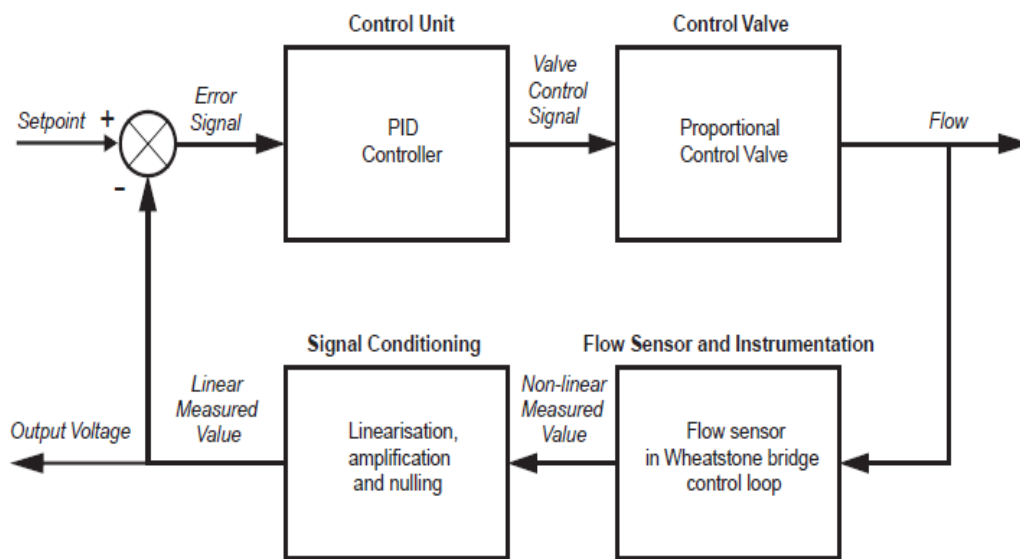
IPA	1.00	SiCl ₄	5.56
Νερό	0,14	TiCl ₄	5,80
Μεθανόλη	0,62	TEOS	1.30
Αιθανόλη	1.00	CupraSelect	4.12
Προπανόλη	1.31	HMDS	1.26

Πίνακας 1. Συντελεστές μετατροπής IPA και άλλων υγρών – Πηγή [14]

Βρόχος Ελέγχου

Ο βρόχος ελέγχου δομείται: από τον αισθητήρα ροής μάζας, το κύκλωμα αισθητήρα, τη βαλβίδα και τη μονάδα ελέγχου. Εφαρμόζοντας τιμή ρύθμισης στο βρόχο ελέγχου, ρυθμίζεται η επιθυμητή ροή. Μια επιθυμητή ροή βασίζεται σε μια συγκεκριμένη τιμή ρύθμισης. Θα μπορούσε δηλαδή, μια τιμή ρύθμισης 40% να αντικατοπτρίζει σε ροή 400 g/h IPA, σε περίπτωση L20 ρυθμισμένο στα 1000 g/h IPA. Όταν οριστεί η τιμή ρύθμισης, ο βρόχος ελέγχου δεσμεύεται ότι θα επιτευχθεί η ζητούμενη ροή, όσο το γίνεται πιο σύντομα με

ήπια απόκριση. Ένα μπλοκ διάγραμμα βρόγχου ελέγχου του ελεγκτή ροής μάζας υγρού L20 απεικονίζεται στο παρακάτω σχήμα (Σχήμα 1.17).



Σχήμα 1.17. Μπλοκ διάγραμμα βρόγχου ελέγχου MFC – Πηγή [14]

1.3 Εμπορικές εφαρμογές

Κατά την συγγραφή της παρούσης εργασίας υπήρχαν στην αγορά διάφοροι θερμικοί ελεγκτές ροής μάζας υγρών και αερίων διαφόρων ειδών, κατάλληλοι για αρκετές εφαρμογές, όπως η κατανομή υγρών και αερίων σε στενευμένα σημεία της παραγωγικής διαδικασίας, η πρόσμιξη διαφορετικών αερίων και υγρών και ο έλεγχος της πίεσης και της ροής αερίων σε διάφορα επίπεδα της παραγωγής, κα. [16].

Οι ελεγκτή ροής μάζας, μπορούν να χρησιμοποιηθούν στον θερμικό ψεκασμό όπου βελτιώνουν την ακρίβεια παροχής υλικού επίστρωσης με σταθερό ρυθμό [17]. Έχουν αξιοποιηθεί στην εξέλιξη νανοσωλήνων άνθρακα πολλών τοιχωμάτων (MWNTs) μέσω καταλυτικής χημικής καθίζησης ατμών (cCVD) [18]. Η χρήση τους σε συστήματα εναπόθεσης ατμών δεν περιορίζεται στην χημική εναπόθεση ατμών καθώς έχουν ενσωματωθεί και σε συστήματα φυσικής εναπόθεσης ατμών [19]. Έχει αναφερθεί η χρήση τους στην ανάπτυξη βιομηχανικών αντιδραστήρων μικρής κλίμακας για παραγωγή ταχύτατων εξώθερων αντιδράσεων [20]. Θα πρέπει να σημειωθεί πάντως ότι χρησιμοποιούνται αρκετά συχνά για την μέτρηση μάζας αέρα για βιομηχανικές εφαρμογές σε συνθήκες υψηλής πίεσης [21].

Ιδιαίτερη αναφορά πρέπει να γίνει για τη χρήση τους στη βιομηχανία ημιαγωγών. Οι εφαρμογές στην κατασκευή ημιαγωγών και ολοκληρωμένων κυκλωμάτων απαιτούν συχνά τη χρήση ελεγκτών ροής μάζας για να στεφθούν εις πέρας. Αυτές οι συσκευές είναι ζωτικής σημασίας για τη διοχέτευση ακριβών ποσοτήτων αερίου κατά τη διάρκεια διαφόρων σταδίων της παραγωγής ημιαγωγών, όπως η μίξη ατμοσφαιρικών συνθηκών, η απομάκρυνση των αποβλήτων και η απόθεση υλικών. Η χρήση τους είναι κρίσιμη για την ελαχιστοποίηση των αποκλίσεων στη διαδικασία κατασκευής, καθώς και για τη διασφάλιση της ομοιομορφίας και της ακρίβειας στη δομή των ημιαγωγών [15].

Σημαντικό ζήτημα αποτελεί ο χρόνος απόκρισης στον έλεγχο των μεταβολών της ροής. Πολλαπλοί ερευνητές προσπαθούν να βελτιώσουν τον ψηφιακό έλεγχο των ελεγκτών ροής μάζας έτσι ώστε να επιτευχθεί ταχύτερος χρόνος απόκρισης [10], [22], [23].

Οι ελεγκτές ροής μάζας, αποτελούν αναγκαία συστήματα για την υποστήριξη βασικών μηχανημάτων στο κλάδο της υγείας, όπως σε συσκευές αναισθησίας καθώς και συσκευές τεχνικής υποστήριξης στην αναπνοή. Οι ελεγκτές δίνουν τη δυνατότητα ακριβούς ρύθμισης των απαιτούμενων αερίων ελαχιστοποιώντας την πιθανότητα αρνητικής επίδρασης στη ζωή των ασθενών. Τα βασικά αέρια που ελέγχονται είναι το οξυγόνο, το διοξείδιο του άνθρακα, το όζον κα [24].

1.4 Ερευνητικές εφαρμογές

Ο Zarkar σε workshop του Αμερικανικού Εμπορίου το 2001 παρατηρούσε ότι η συνεχής πρόοδος και οι βελτιώσεις λόγω της σμίκρυνσης των τσιπ των ημιαγωγών, δημιούργησαν νεότερα, μικρότερα και ταχύτερα εξαρτήματα ανταλλακτικά χειρισμού υγρών και αερίων. Ταυτόχρονα η σύγχρονη ανάγκη για δημιουργία κατασκευών με λεπτομερέστερα χαρακτηριστικά ανάγκασε τη βιομηχανία παραγωγής υλικών για χειρισμό αερίων να περάσει από την κατασκευή παρτίδων (batch) σε μοναδικές κατασκευές (single wafer), επηρεάζοντας έτσι όλη την σειρά ανταλλακτικών.

Παρατηρούσε ότι ημιαγωγοί και νέες λειτουργίες τους όπως CVD, πλάσμα και LED, θα επηρεάσουν κυρίως τους ελεγκτές μάζας ροής, οι οποίοι είναι οι κυρίως υπεύθυνοι για την ακεραιότητα του προϊόντος, κάτι που τελικά αποδείχθηκε στην πράξη. Η βιομηχανία απαίτησε νέες και συνεχώς βελτιωμένες γενιές MFC ανώτερες σε απόδοση, πιο ευέλικτες στο χειρισμό πολλαπλών αερίων, καθώς και πιο αξιόπιστες που δεν θα δεσμεύονται από περιορισμούς πατέντας. Για να επιτευχθούν τα καλύτερα αποτελέσματα, η επιλογή εξαρτημάτων παροχής αερίου διαδραματίζει τον ζωτικότερο ρόλο στην εξασφάλιση των αυστηρότερων και πιο απαιτητικών απαιτήσεων της διαδικασίας [25].

1.4.1 Χρήση των MFCs για περιβαλλοντικές συνθήκες

Η χρήση ελεγκτών ροής μάζας για την δημιουργία μειγμάτων αερίων, χρησιμοποιούνται στην επίβλεψη περιβαλλοντικών παραμέτρων, αναφέρουν οι Loianno & Mensitieri το 2022. Στη χρήση έτοιμων μειγμάτων αερίων, τα οποία μετακινούνται σε σωλήνες μεγάλης πίεσης έρχεται να δώσει λύση η παραπάνω πρόταση. Αυτό επιτυγχάνεται διότι μπορεί να γίνει άμεση χρήση του ελεγκτή ροής μάζας και η σύσταση του μείγματος αερίου μπορεί να προσαρμόζεται κάθε φορά που αυτό απαιτείται.

1.4.2 Χρήση των MFCs στο διάστημα

Οι ελεγκτές ροής μάζας χρησιμοποιούνται ευρέως στη βιομηχανία των διαστημικών εφαρμογών. Τα διαστημικά συστήματα και τα συστημάτων υποστήριξης ζωής, έχουν ως βάση τους ελεγκτές ροής μάζας οι οποίοι τους προσφέρουν σταθερότητα και ακρίβεια [26-29]. Επιπλέον, συμβάλλουν στην αποφυγή υψηλότερων ροών από το επιθυμητό, οδηγώντας έτσι σε σημαντική εξοικονόμηση ενέργειας κατά τη διάρκεια λειτουργίας του συστήματος [26]. Τα συστήματα προώθησης για την κίνηση των διαστημοπλοίων, καθώς και οι συσκευές

θέρμανσης και ψύξης, είναι υπό την εποπτεία αντίστοιχων ελεγκτών ροής μάζας, εξασφαλίζοντας έτσι την ασφάλεια και αποτελεσματικότητα της λειτουργίας τους, μέσω του ακριβούς ελέγχου της ροής. Η σχεδίαση των ελεγκτών ροής μάζας είναι ιδιαίτερα σημαντική για τις εφαρμογές στο διάστημα, καθώς πρέπει να λειτουργούν αποτελεσματικά σε συνθήκες μηδενικής βαρύτητας. Επιπλέον, τα ηλεκτρονικά των διαστημικών συστημάτων πρέπει να αντέχουν σε αυστηρές δοκιμές δόνησης κατασκευής με ένταση 6,8 G RMS [16]. Χαρακτηριστικά παραδείγματα εφαρμογής των MCFs σε διατάξεις που θα χρησιμοποιηθούν στο διάστημα είναι θερμικά συστήματα ελέγχου ροής μάζας για προωθητές συμπυκνώσιμου προωθητικού με Hall-effect [28], στον έλεγχο της ροής πρόωσης των διαστημοπλοίων [27], για μετρήσεις αερίων των συστημάτων ψύξης σε διαστημικούς σταθμούς [30], ή ακόμα και για τον έλεγχο υγρών σε συνθήκες μηδενικής βαρύτητας [26].

1.4.3 Άλλες εφαρμογές

Αναπόφευκτα γίνεται προσπάθεια ένταξης των MFCs στις τεχνολογίες αιχμής για την παραγωγή ενέργειας αλλά και στις ανάγκες της εποχής. Για παράδειγμα οι Bekraoui & Hadjadj προσπάθησαν να εντάξουν τη χρήση MCFs για την μέτρηση ροής φυσικού αερίου, εφαρμογή στην οποία δε χρησιμοποιείται συνήθως. Στη διάταξη χρησιμοποιήθηκε η τεχνολογία ανεμόμετρων θερμού σύρματος [31]. Η πανδημία από την άλλη ανέδειξε την ανάγκη για μετρήσεις ζωτικών λειτουργιών του σώματος σε συνθήκες νοσηλείας, εκτός εργαστηρίου δίνοντας ώθηση στις συσκευές lab-on-chips. Η Gopinath ανέπτυξε πρόσφατα έναν MCF σε συσκευή lab-on-chip για έλεγχο μικρορευστών [32].

Η γενικότερη τάση όμως είναι η υλοποίηση ερευνών για τη βελτίωση των μικροχαρακτηριστικών, της ακρίβειας των μετρήσεων και του μη επηρεασμού τους από τις συνθήκες ή τα υλικά [33].

2 ΚΕΦΑΛΑΙΟ 2^ο : ΔΟΜΗ ΚΑΙ ΑΡΧΗ ΛΕΙΤΟΥΡΓΙΑΣ ΤΟΥ ΣΥΣΤΗΜΑΤΟΣ ΠΡΟΣ ΥΛΟΠΟΙΗΣΗ

Σε αυτό το κεφάλαιο περιγράφεται η λειτουργία των στοιχείων που συνθέτουν το σύστημα. Πιο συγκεκριμένα, αναλύεται το πρωτόκολλο που χρησιμοποιήθηκε στην επικοινωνία μικροελεγκτή και αισθητήρα. Οι διατάξεις που χρησιμοποιήθηκαν είναι: ο μικροελεγκτής STM32F401RE, ο αισθητήρας M-20SLPM-D της εταιρίας Alicat Scientific, το IC MAX232 και η σφαιρική βαλβίδα ελέγχου ροής.

2.1 Επικοινωνία μέσω του διαύλου RS-232

Η επικοινωνία του μικροελεγκτή με τον αισθητήρα, αποτελεί μείζον ζήτημα για την πορεία υλοποίησης του ελεγκτή ροής μάζας. Η διαδικασία της σύνδεσης των δυο συσκευών είναι κρίσιμη για τη σωστή λειτουργία και αποδοτικότητα του συστήματος. Μέσω της αμφίδρομης επικοινωνίας για την ανταλλαγή πληροφοριών, επιτυγχάνεται η ανατροφοδότηση καθ' όλη τη διάρκεια της λειτουργίας του συστήματος. Τα δεδομένα που λαμβάνει το STM32F401RE από το MFM στέλλονται μέσω της χρήσης του πρότυπου RS-232. Το RS-232 αποτελεί ένα πρωτόκολλο επικοινωνίας για σειριακή μετάδοση σημάτων (διαδίκων δεδομένων) μεταξύ δυο συσκευών. Με τη χρήση του υλικού RS232 to TTL converter (IC MAX232) επιτυγχάνεται η μετατροπή του σήματος από RS-232 σε επίπεδο TTL, προκειμένου να είναι σε θέση ο μικροελεγκτής να επεξεργαστεί το εισερχόμενο σήμα. Παρακάτω αναλύονται οι προδιαγραφές του πρότυπου RS-232 και η λειτουργία του.

2.1.1 Εισαγωγή στο πρωτόκολλο RS-232 (EIA-232)

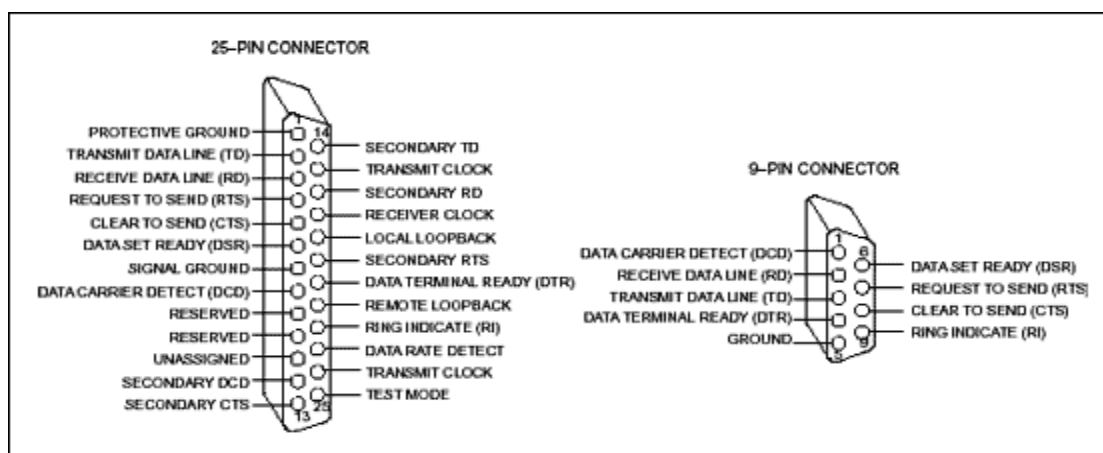
Το πρωτόκολλο RS-232 ήρθε σε κυκλοφορία για πρώτη φορά το 1969 στις ΗΠΑ από το μηχανικό τμήμα της EIA. Μετά από ορισμένες διευθετήσεις, το πρωτόκολλο ορίστηκε σε RS-232C. Ωστόσο, το 1988 έλαβε την ονομασία EIA/TIA-232, αντικαθιστώντας το πρόθεμα “RS”, και αργότερα τον Ιούλιο του 1991 ορίστηκε η τελική ονομασία του, EIA/TIA-232E, οπότε και εντάχθηκε στα διεθνή πρότυπα ITU V.24, ITU V.28 και ISO 2110 [34] Παρόλα αυτά, ενώ το RS-232 υπήρξε σταθερός τρόπος σύνδεσης υπολογιστών με περιφερειακές συσκευές, η χρήση του έχει μειωθεί στις σύγχρονες εφαρμογές λόγω της εμφάνισης πιο εξελιγμένων προτύπων επικοινωνίας όπως το USB, Ethernet κλπ.

Το RS-232 αναπτύχθηκε με μοναδικό σκοπό τη σειριακή δυαδική ανταλλαγή δεδομένων τερματικών σε μόντεμ. Συγκεκριμένα, στοχεύει στη διασύνδεση μεταξύ συσκευών DTE (Data Terminal Equipment) και DCE (Data Circuit-Term Equipment). Το RS-232 χρησιμοποιεί μια μέθοδο σειριακής επικοινωνίας, μεταδίδοντας ένα bit τη φορά και υποστηρίζεται για σύγχρονες ή ασύγχρονες μεταδόσεις σε (Full or Half-Duplex). Η μεγάλη ζήτηση τού οφείλεται στην ευελιξία που το εν προκειμένω πρωτόκολλο παρέχει και στο χαμηλό κόστος που διασφαλίζει.

Η τυπική διαμόρφωση των μόντεμ βασίζεται στη θηλυκή υποδοχή τύπου D-25. Όμως, το πρωτόκολλο RS-232 μπορεί να λειτουργεί σε διάφορα επίπεδα πολυπλοκότητας. Το πιο απλό επίπεδο συνιστά μια απλουστευμένη εκδοχή του D-25, διότι αποτελείται μόνο από μια γραμμή μετάδοσης, λήψης και γείωσης. Παράλληλα, η διεπαφή επιτρέπει την ταυτόχρονη αμφίδρομη μετάδοση δεδομένων, ακόμη και σε αυτή τη μορφή [34],[35].

2.1.2 Σήματα και συνδέσεις (Προδιάγραφες πρωτοκόλλου)

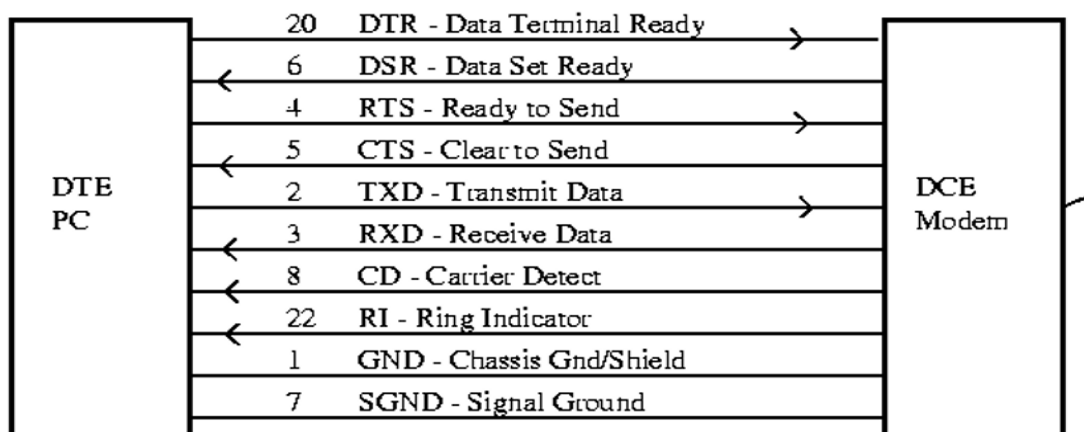
Το πρωτόκολλο RS-232 έχει μέγιστο εύρος τάσης λειτουργίας τα 25 V. Ο πομπός παράγει διπολικές τάσεις λειτουργίας σε ένα επίπεδο από ± 15 έως ± 25 V, επίπεδα σήματος ± 5 V, ± 10 V, ± 12 V και ± 15 V εμφανίζονται συνήθως ανάλογα με τις διαθέσιμες τάσεις στο κύκλωμα οδηγού. Για τον σχηματισμό του λογικού "1", τα επίπεδα τάσης του εκπομπού κυμαίνονται από -5 V έως -25 V, ενώ αντίστοιχα για το λογικό "0" είναι από $+5$ V έως $+25$ V. Όσον αφορά στον δέκτη, τα επίπεδα τάσης ορίζονται από -3 V έως -25 V για το λογικό "1" και από $+3$ V έως $+25$ V για το λογικό "0", λαμβάνοντας ως σημείο αναφοράς την Κοινή Γείωση. Ακολούθως, παρατηρείται ότι ο πομπός RS-232 προσεγγίζει υψηλότερα επίπεδα τάσης σε σύγκριση με εκείνα που δέχεται ο δέκτης. Αυτό συμβαίνει διότι απαιτείται ελαφρώς υψηλότερη τάση, ούτως ώστε να καλυφθεί η πτώση τάσης κατά μήκος της γραμμής. Το πρωτόκολλο RS-232 υποστηρίζει δυο τύπους ακροδεκτών, όπως προδίδει και το όνομα τους: το D-9 με 9 ακροδέκτες και το D-25 με τους 25 ακροδέκτες [34] (Σχήμα 2.1).



Σχήμα 2.1. D-25, D-9 pin out- Πηγή [37]

Για τον έλεγχο της ροής των δεδομένων, το πρωτόκολλο RS-232 εκπέμπει τα κατάλληλα σήματα ελέγχου, προκειμένου κάθε συσκευή να αντιλαμβάνεται πότε η έτερη συσκευή είναι έτοιμη να προωθήσει ή να δεχθεί δεδομένα την εκάστοτε χρονική στιγμή που αυτό θα ζητηθεί. Ο συντονισμός της επικοινωνίας μεταξύ του DTE και του DCE γίνεται με τα σήματα RTS (Request to Send) και CTS (Clear To Send). Κάθε φορά που το DTE έχει δεδομένα για αποστολή, ενεργοποιείται το σήμα RTS, και όταν το DCE είναι έτοιμο να αποδεχθεί τα δεδομένα αυτά, ενεργοποιείται αντιστοίχως το σήμα CTS. Στη συνέχεια, για να δεχθεί το DTE δεδομένα θα πρέπει πρώτα να ενεργοποιηθεί το σήμα DTR (Data Terminal Ready). Μάλιστα, το DTR θα πρέπει να βρίσκεται σε ενεργή κατάσταση, πριν την ενεργοποίηση του DSR (Data Set Ready), προκειμένου να δημιουργηθεί μια σχέση επικοινωνίας από τη πλευρά του DCE [34],[36].

Η επαλήθευση της ποιοτικής επικοινωνίας αυτών των δύο γίνεται με τη χρήση του σήματος DC (Data Carrier), το οποίο χρησιμοποιείται από το DCE για να υποδείξει στο DTE ότι λαμβάνει το επιθυμητό σήμα. Η εντολή RI (Ring) υποδηλώνει την ένδειξη μιας εισερχόμενης τηλεφωνικής κλήσης. Η συνδεσμολογία απεικονίζεται στο παρακάτω σχήμα [34] (Σχήμα 2.2).



Σχήμα 2.2. Συνδεσμολογία μεταξύ DTE και DCE - Πηγή [34]

Τα σήματα αυτά (σήματα ελέγχου), όπως αναφέρθηκε και προηγουμένως, χρησιμοποιούνται κατά κύριο λόγο για την ανταλλαγή δεδομένων μεταξύ μόντεμ. Συνεπώς, ορισμένα από αυτά τα σήματα να μην είναι συμβατά με άλλου τύπου συσκευές. Η παραπάνω συνδεσμολογία επιτυγχάνεται με τη χρήση του συνδέσμου D-25. Η πλειονότητα των εφαρμογών χρησιμοποιεί την απλουστευμένη εκδοχή με σήματα: Tx, Rx και GRD (γραμμή μετάδοσης, λήψης και γείωσης) [36]. Ως απαραίτητη προϋπόθεση κατά την συνδεσμολογία των δυο αυτών συσκευών τίθεται η περιστροφή των ακροδεκτών των σημάτων Rx και Tx, έτσι ώστε ο ακροδέκτης μετάδοσης της μια συσκευής να συνδέεται με τον ακροδέκτη λήψης της έτερης συσκευής και αντίστροφα.

2.1.3 Τα χαρακτηριστικά του πρωτοκόλλου RS-232

Baud Rate

Το πρωτόκολλο RS-232 αποτελείται από ένα σύνολο κανόνων, η τήρηση των οποίων είναι αναγκαία για την ανταλλαγή δεδομένων μεταξύ συσκευών. Είναι σημαντικό να αναφερθεί ότι, το εν προκειμένω πρωτόκολλο λειτουργεί ασύγχρονα, γι' αυτό δεν μεταδίδεται αποκλειστικό σήμα ρολογιού μαζί με τα δεδομένα. Κατ' επέκταση, για να συγχρονιστούν οι συσκευές που επικοινωνούν σειριακά (handshake), τόσο ο αποστολέας όσο και ο παραλήπτης, συμφωνούν σε ένα κοινό ρυθμό επικοινωνίας baud rate. Παρ' όλα αυτά, δεν είναι εφικτός ο τέλειος συντονισμός των συστημάτων. Όμως, τα συστήματα μπορούν να συγχρονιστούν σε έναν αρκετά καλό βαθμό.

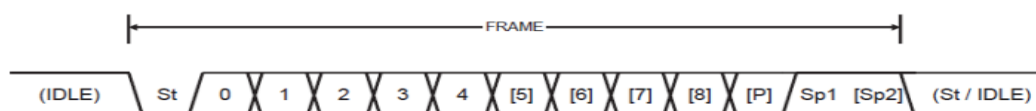
Η ταχύτητα διάδοσης των δεδομένων είναι ανάλογη της απόστασης μεταξύ των εκάστοτε συσκευών. Ειδικότερα, η ταχύτητα ανέρχεται στα 19,2Kbps για μέγιστη απόσταση 15 μέτρα. Σε μικρότερες, αποστάσεις η ταχύτητα μπορεί να ανέρθει μέχρι τα 115,2 Kbps (115000 bit/sec). Όμως, με τη χρήση ειδικού εξοπλισμού οι ταχύτητες μπορούν να αυξηθούν ακόμη περισσότερο. Παρακάτω απεικονίζεται ένας πίνακας με τους πιο διαδομένους ρυθμούς μετάδοσης Baud Rate [38],[39]. (Πίνακας 2).

Baud Rate	Time (ms)
1200	8.33
2400	4.16
4800	2.08
9600	1.04
19200	0.520
38400	0.260
57600	0.176
115200	0.0868

Πίνακας 2. RS-232 Baud Rates - Πηγή [39]

Frame format

Εκτός από την ταχύτητα αποστολής των δεδομένων, θα πρέπει να γνωστοποιηθούν στις συσκευές που επικοινωνούν τα ακόλουθα: η χρονική στιγμή που τα δεδομένα ξεκινούν να στέλνονται, ο αριθμός των bits προς αποστολή, η επαλήθευση του μηνύματος και η στιγμή του τερματισμού. Πιο αναλυτικά, κατά τη διαδικασία της επικοινωνίας το IDLE είναι high. Όμως, όταν το start bit δώσει σήμα, το high γίνεται low και τότε εκκινεί η διαδικασία της μετάδοσης. Η επαλήθευση της αλληλουχίας των δεδομένων επιτυγχάνεται μέσω του bit parity. Συγκεκριμένα, τοποθετείται στο τέλος κάθε μηνύματος και μπορεί να λάβει άρτια ή περιττή τιμή. Σε ένα ζυγό σύστημα ισοτιμίας το bit λαμβάνει την τιμή μηδέν, ενώ σε ένα περιττό σύστημα λαμβάνει την τιμή ένα. Αν το ληφθέν bit parity δεν αντιστοιχεί με την τιμή που αυτό αναμένεται να προσλάβει, τότε η αλληλουχία των δεδομένων είναι λάθος. Το παρακάτω σχήμα (Σχήμα 2.3) απεικονίζει ένα παράδειγμα μηνύματος σειριακών δεδομένων συμβολισμένο σε bit. Η αποτύπωση του σήματος υποδηλώνει ότι το σήμα δεν είναι ιδανικό, γεγονός που αποδεικνύεται λόγω της καθυστέρησης που παρατηρείται σε κάθε αλλαγή στον παλμό από 0 σε 1 και αντιστρόφως.



- St** Start bit, always low.
- (n)** Data bits (0 to 8).
- P** Parity bit. Can be odd or even.
- Sp** Stop bit, always high.
- IDLE** No transfers on the communication line (RxDn or TxDn). An IDLE line must be high.

Σχήμα 2.3. Απεικόνιση της δομής ενός frame- Πηγή [40]

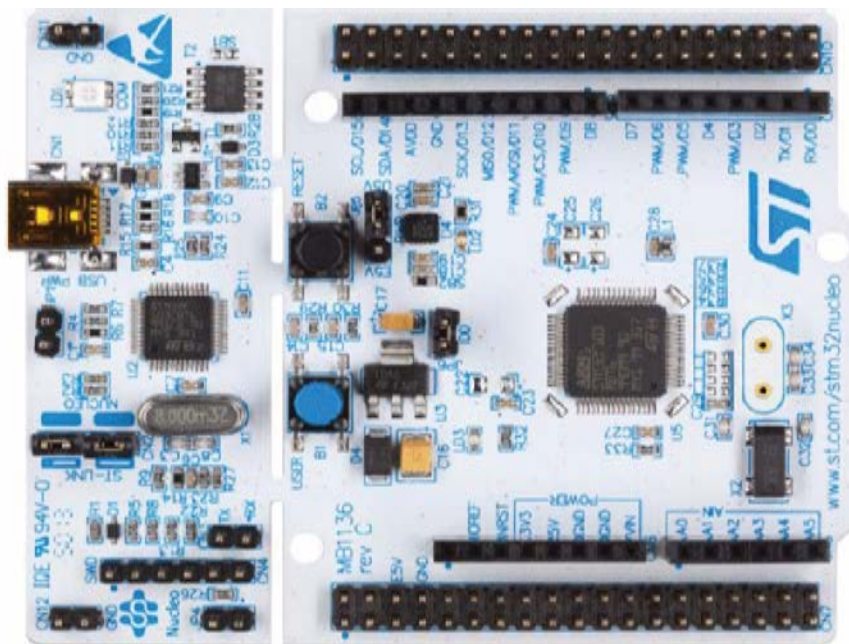
Ο τερματισμός του μηνύματος επιτυγχάνεται με το stop bit. Μάλιστα, σε ορισμένες περιπτώσεις υπάρχουν δύο stop bit, ούτως ώστε να μην υπάρξει σύγχυση μεταξύ του high του stop bit και του τελευταίου bit του μηνύματος. Αξίζει να σημειωθεί ότι μπορούν να προκύψουν έως τριάντα διαφορετικοί συνδυασμοί των ανωτέρω μεταβλητών. Όμως, όσο περισσότερα είναι τα πρόσθετα bit (Over head), τόσο αυξάνεται η ποσότητα των συνολικών δεδομένων του μηνύματος. Κατ' επέκταση, να μειώνεται ο συνολικός όγκος πληροφοριών που μπορεί να μεταφερθεί (Bandwidth). Τα byte των δεδομένων αντιστοιχούν σε χαρακτήρες του κώδικα ASCII. Δίνεται η δυνατότητα αποστολής έως 9 bits για Extended ASCII [40].

2.2 Ο μικροελεγκτής STM32F401RE

2.2.1 Εισαγωγή στο STM32F401RE

Ο μικροελεγκτής STM32F401RE περιέχει τον επεξεργαστή Cortex-M4 της «οικογένειας» Arm. Οι επεξεργαστές της εταιρείας Arm Limited βασίζονται στην αρχιτεκτονική υπολογιστών με μειωμένο σετ εντολών (Reduced instruction set computer – RISC) [41]. Θεωρούνται χαμηλού κόστους, απαιτούν μικρά ποσά ενέργειας για λειτουργία, παράγουν χαμηλά επίπεδα θερμότητας, έτσι είναι πολύ δημοφιλείς για χρήση σε ελαφριές φορητές συσκευές που τροφοδοτούνται από μπαταρίες όπως έξυπνα κινητά, φορητούς υπολογιστές, tablets αλλά και ενσωματωμένα συστήματα (embedded systems) [42]. Οι μικροελεγκτές της σειράς STM32F είναι ευρέως διαθέσιμοι στην αγορά ηλεκτρονικών [41] και είναι συμβατοί με άλλους τύπους μικροελεγκτών όπως PIC και Arduino [41],[43],[44].

Η αναπτυξιακή πλατφόρμα του STM32F401RE θεωρείται πιο ισχυρή συσκευή σε σύγκριση με την αντίστοιχη συσκευή Arduino Uno με τον μικροελεγκτή ATmega328P της «οικογένειας» AVR των 8 bit [45]. Θεωρείται μάλιστα συσκευή η οποία αξιοποιεί την αιχμή της τεχνολογίας που επιτρέπει την συλλογή δεδομένων, την προ-επεξεργασία, την εξαγωγή συμπερασμάτων, αναγνώρισης συσκευών με τη χρήση αλγορίθμων βαθιάς μάθησης (deep learning) [46]. Από την άλλη οι μικροελεγκτές της σειράς M4 όπως είναι ο STM32F401RE έχουν περίπου την μισή τιμή αυτών της σειράς M7 επιτυγχάνοντας συγκρίσιμη λειτουργικότητα αλλά και ακρίβεια ρυθμίσεων και αποτελεσμάτων στην ίδια περίπου ταχύτητα ADC (2,4 MSPS), παρόλο που οι μικροελεγκτές της σειράς M7 έχουν μεγαλύτερη μνήμη flash [47].



Σχήμα 2.4. Απεικόνιση του ολοκληρωμένου STM32F401RE- Πηγή [50]

2.2.2 Βασικά χαρακτηριστικά

Ο μικροελεγκτής STM32F401RE έχει συχνότητα ρολογιού (clock frequency) μέχρι και 84 MHz, έξι (6) εισόδους αισθητήρων (sensor inputs) και 8 εξόδους ελέγχου (control outputs) [41]. Τα χαρακτηριστικά του μικροελεγκτή φαίνονται στο παρακάτω πίνακα (Πίνακας 3):

Core	ARM 32 Cortex M4
Χρήση Ελεγκτή	Γενικού σκοπού (General Purpose Direct Memory Access Controller - GPDMA)
Συχνότητα Κεντρικής Μονάδας Επεξεργασίας	84 MHz
Flash Memory	512 KB
Στατική μνήμη τυχαίας προσπέλασης (Static random-access memory)	96 KB
Ασφάλεια	Μονάδα Προστασίας Μνήμης (Memory Protection Unit)
Τύπος θύρας USB	USB OTG FS
Μέγιστη τάση τροφοδοσίας	3,6 Volt
Ρεύμα τροφοδοσίας ανά MHz	137 μ A
Μετατροπέας (Converter)	1 x 12 bit, 2.4 Megasamples per second (MSPS) A/D Converter

Πίνακας 3. Βασικά χαρακτηριστικά μεγέθη STM32F401RE - Πηγή [47]

Περιλαμβάνονται: ενσωματωμένες μνήμες υψηλής ταχύτητας, χωρητικότητας, 512 KB η flash memory και 96 KB η SRAM, μέχρι και 11 χρονοδιακόπτες (timers), δύο χρονιστές επαγρύπνησης (watchdog timers) και χρονισμό SysTick. Ακόμη περιλαμβάνεται έναν 12-bit ADC, ένα on-chip real time clock (RTC) χαμηλής τάσης [49], έξι χρονιστές γενικής χρήσης των 16-bit, ένα χρονιστή PWM για τον έλεγχο κινητήρα και δύο χρονιστές των 32-bit για γενική χρήση. Σημαντικό στοιχείο είναι ότι δίνει τη δυνατότητα συμβατικής και προχωρημένης επικοινωνίας μέσω διεπαφών, καθώς εσωκλείει ογδονταμία (81) I/O θύρες με δυνατότητα διακοπής, τρεις (3) I2 C, τρεις (3) USARTs, τέσσερα 4 SPIs και θύρα USB πρωτοκόλλου 2.0 [48],[50]. Η τάση εξόδου μπορεί να ανέλθει στα 3,3 ή 5 Volts [41]. Η αποσφαλμάτωση (debugging) μπορεί να γίνει μέσω σειριακής επικοινωνίας με καλώδιο SWD ή JTAG [48].

Έχει τέσσερις (4) λειτουργίες ισχύος οι οποίες ποικίλουν σε κατανάλωση ισχύος από 2,4 $\mu\text{A}/\text{MHz}$ όταν είναι σε κατάσταση αναμονής (standby mode) έως 146 $\mu\text{A}/\text{MHz}$ όταν είναι σε λειτουργία χωρίς διασυνδεδεμένες περιφερειακές μονάδες (run mode with peripheral off) κατανάλωση ενέργειας η οποία θεωρείται ιδιαίτερα μικρή [51],[48], αν και εμφανίζει σημαντικά μεγαλύτερη κατανάλωση συγκρινόμενος με άλλους μικροελεγκτές της ίδιας σειράς όπως είναι ο STM32L476RG [48].

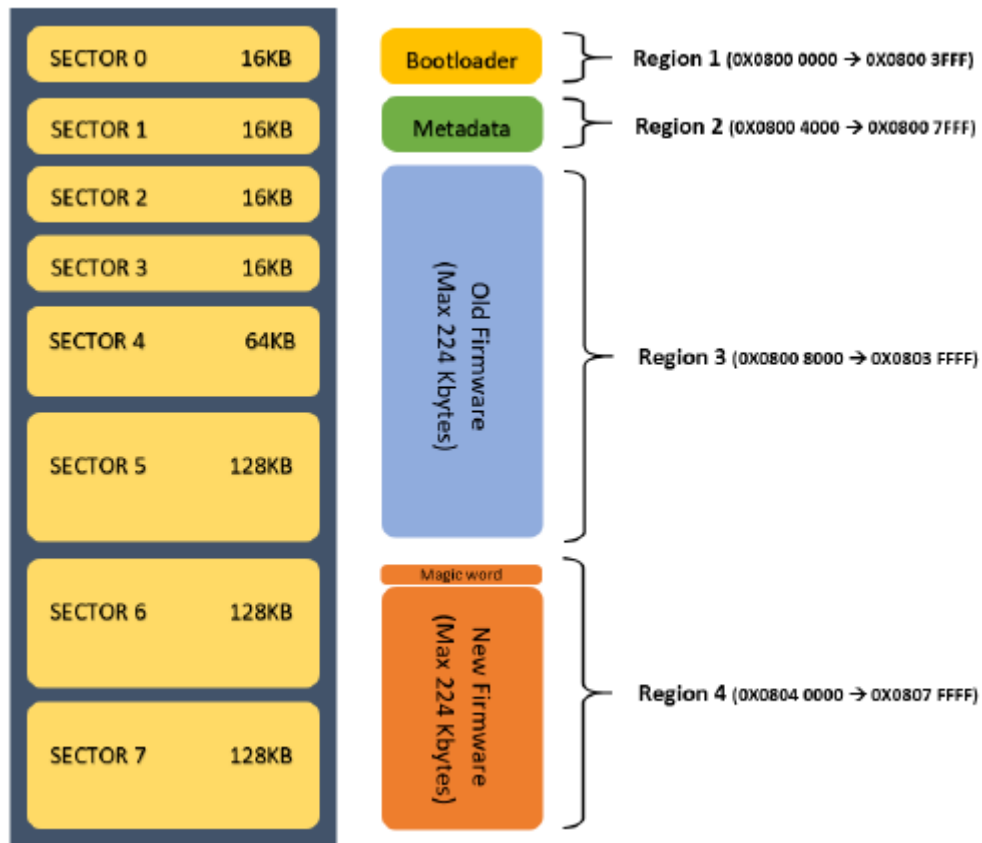
Η flash μνήμη του μικροελεγκτή, όπως προαναφέρθηκε, είναι συνολικής χωρητικότητας 512Kb, τα οποία κατανέμονται σε επτά (7) τομείς όπως δείχνει ο παρακάτω πίνακας (Πίνακας 4):

Block μνήμης	Όνομα τομέα	Διεύθυνση μνήμης	Χωρητικότητα
Κύρια μνήμη	0	0x0800 0000 - 0x0800 3FFF	16 KB
	1	0x0800 4000 - 0x0800 7FFF	16 KB
	2	0x0800 8000 - 0x0800 BFFF	16 KB
	3	0x0800 C000 - 0x0800 FFFF	16 KB
	4	0x0801 0000 - 0x0801 FFFF	64 KB
	5	0x0802 0000 - 0x0803 FFFF	128 KB
	6	0x0804 0000 - 0x0805 FFFF	16 KB
	7	0x0806 0000 - 0x0807 FFFF	16 KB

Πίνακας 4. Κατάτμηση (partition) τομέων μνήμης flash μικροελεγκτή – Πηγή [48]

Η κατάτμηση (partition) της μνήμης flash του μικροελεγκτή δε θεωρείται ιδιαίτερα λεπτόκοκκη (grained) οπότε έχει ενσωματωθεί η ελάχιστη δυνατή δυνατότητα αναβάθμισης του υλικολογισμικού (firmware) καθώς η τεχνολογία flash απαιτεί σβήσιμο και επανεγγραφή ολόκληρων τομέων και όχι μέρους τους [48].

Η αναβάθμιση του μικροελεγκτή STM32F401RE είναι δυνατή όταν παραλαμβάνει αίτημα ανανέωσης από τον διακομιστή (server) ελέγχοντας αν η τιμή του πόρου (STATE Resource είναι ίση με τη λήψη (Downloaded). Αυτό υποδηλώνει ότι η λήψη του νέου firmware ολοκληρώθηκε. Ακολούθως η τιμή αλλάζει σε Update και τα δεδομένα αποθηκεύονται στην περιοχή 2 μνήμης 2, και προστίθεται η συμβολοσειρά MAGIC WORD στην νέα εικόνα firmware όπως φαίνεται στο παρακάτω σχήμα [48],[50] (Σχήμα 2.5):



Σχήμα 2.5. Χαρτογράφηση μνήμης flash (flash memory mapping) μικροελεγκτή STM32F401RE - Πηγή [48]

Με την επανεκκίνηση εκτελείται ο bootloader και ελέγχει αν είναι παρούσα η συμβολοσειρά MAGIC WORD στην περιοχή 4, ακολούθως σβήνει την περιοχή 3 και αντιγράφει το περιεχόμενο της 4 στην νέα περιοχή 3. Αν η παραπάνω διαδικασία είναι επιτυχής, οι τιμές των πόρων (resources) STATE και UPDATE αλλάζουν σε idle και Successful αντίστοιχα. Η παραπάνω μέθοδος αναβάθμισης (Push) firmware καλείται Push και ο μικροελεγκτής STM32F401RE υποστηρίζει και την μέθοδο Pull όχι όμως τη λειτουργία dual boot όπως άλλοι μικροελεγκτές της ίδιας σειράς [48].

2.3 Ο αισθητήρας ροής μάζας M-20SLPM-D της εταιρίας Alicat Scientific

Ο αισθητήρας M-20SLPM-D (Mass Flow Meter) συνιστά ένα όργανο μέτρησης ροής μάζας, σχεδιασμένο και κατασκευασμένο από την εταιρία Alicat Scientific. Η Alicat Scientific ιδρύθηκε το 1991, με σκοπό την κατασκευή και προσαρμογή οργάνων ροής μάζας και πίεσης. Ορισμένες από τις εφαρμογές των συσκευών Alicat Scientific είναι η δοκιμή συστημάτων παροχής οξυγόνου σε νοσοκομεία και ερευνητικά εργαστήρια, η παραγωγή καυσίμου από βιομάζα και η καθοδήγηση πυραύλων [52].

2.3.1 Οι προδιαγραφές του αισθητήρα M-20SLPM-D

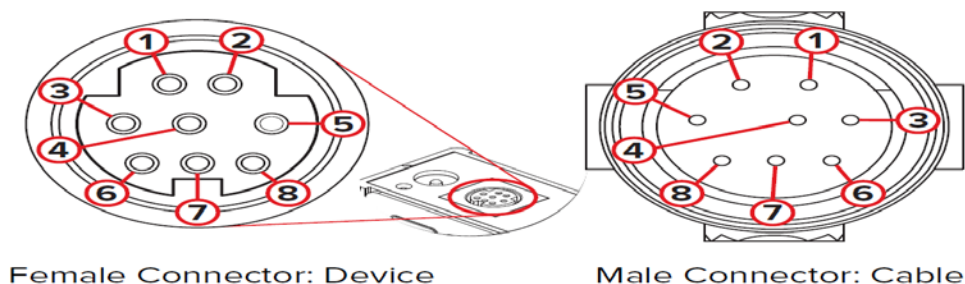
Ο αισθητήρας M-20SLPM-D έχει τη δυνατότητα να επεξεργάζεται ροή αερίου έως και 25 τυπικά λίτρα ανά λεπτό (SLPM), επιτρέποντας έτσι αξιόπιστες μετρήσεις σε ένα ευρύ φάσμα ρυθμών ροής. Η ακρίβεια μαζικής ροής (Mass flow Accuracy) περιλαμβάνει την τυπική ακρίβεια ανάγνωσης (Standard Accuracy), που εκτείνεται σε $\pm 0,6\%$ ή $\pm 0,1\%$ της πλήρους κλίμακας. Ενώ η επιλογή υψηλής ακρίβειας (High-Accuracy Option) κυμαίνεται από $\pm 0,5\%$ της ανάγνωσης ή $\pm 0,1\%$ της πλήρους κλίμακας, παρέχοντας σταθερά και ακριβή αποτελέσματα. Το εύρος μετρήσεων ροής (Flow Measurement Range) είναι από 0,01% έως και 100% της πλήρους κλίμακας με turndown ratio 10.000 προς 1. Η ακρίβεια πίεσης (Pressure accuracy) διατηρείται στο $\pm 0,5\%$ της ένδειξης για τιμές πάνω από 1 atm και σε $\pm 0,07$ PSIA για τιμές κάτω από 1 atm. Η θερμοκρασία λειτουργίας παίρνει τιμές από -10 έως 60°C σε ακρίβεια θερμοκρασίας $\pm 0,75^{\circ}\text{C}$. Ο χρόνος προθέρμανσης βρίσκεται κάτω του 1sec, ενώ αντίστοιχα ο χρόνος απόκρισης του αισθητήρα κυμαίνεται υπό του 1 ms και ο χρόνος απόκρισης ένδειξης κάτω από 10 ms. Έτσι, επαληθεύεται ότι η συσκευή μπορεί να προσαρμοστεί σε ευέλικτο περιβάλλον και εξειδικευμένες καταστάσεις. Τα χαρακτηριστικά της συσκευής απεικονίζονται συνοπτικά στον παρακάτω πίνακα [53] (Πίνακας 5).

SENSOR PERFORMANCE	
Mass flow accuracy ¹	Standard accuracy: $\pm 0.6\%$ of reading or $\pm 0.1\%$ of full scale, whichever is greater High-accuracy option: $\pm 0.5\%$ of reading or $\pm 0.1\%$ of full scale, whichever is greater
Repeatability (2 σ)	$\pm (0.1\% \text{ of reading} + 0.02\% \text{ of full scale})$
Pressure accuracy ¹	Above 1 atm: $\pm 0.5\%$ of reading Below 1 atm: ± 0.07 PSIA
Flow measurement range	0.01–100% of full scale (10,000:1 turndown ratio)
Operating pressure	11.5–160 PSIA
Pressure sensitivity	Mass flow zero shift: $\pm 0.01\%$ of full scale per atm from tare pressure Mass flow span shift: $\pm 0.1\%$ of reading per atmosphere from calibration conditions
Temperature sensitivity	Mass flow zero shift: $\pm 0.01\%$ of full scale per $^{\circ}\text{C}$ from tare temperature Mass flow span shift: $\pm 0.01\%$ of reading per $^{\circ}\text{C}$ from 25°C
Temperature accuracy	$\pm 0.75^{\circ}\text{C}$
Operating temperature range	-10 – 60°C (ambient and gas)
Totalizer volume uncertainty	$\pm 0.1\%$ of reading additional uncertainty
Sensor response time	<1 ms
Typical indication response time	<10 ms, flow rate dependent
Typical warm-up time	<1 s

Πίνακας 5. Βέλτιστη απόδοση του αισθητήρα M-20SLPM-D - Πηγή [53]

2.3.2 Συνδέσεις αισθητήρα (Pin out)

Το παρόν μοντέλο του αισθητήρα (M-20SLPM-D) περιλαμβάνει μια υποδοχή ομοαξονικού συνδέσμου (coaxial power connector) στο πάνω μέρος της συσκευής για την τροφοδοσία τής. Μια υποδοχή 8-Pin Mini-DIN εμπεριέχεται επίσης στο πάνω μέρος της συσκευής, διπλά από την υποδοχή ομοαξονικού συνδέσμου, με σκοπό την επικοινωνία και έξοδο του σήματος (Σχήμα 2.6). Η συσκευή μπορεί επίσης να τροφοδοτηθεί από την υποδοχή 8-Pin Mini-DIN, εφόσον το σύστημα που θα συνδεθεί παρέχει την απαιτούμενη τροφοδοσία. Παρακάτω γίνεται ανάλυση των ακροδεκτών της υποδοχής 8-Pin Mini-DIN [54].



Σχήμα 2.6. Απεικόνιση 8-Pin Mini-DIN στο πάνω μέρος του αισθητήρα - Πηγή [54]

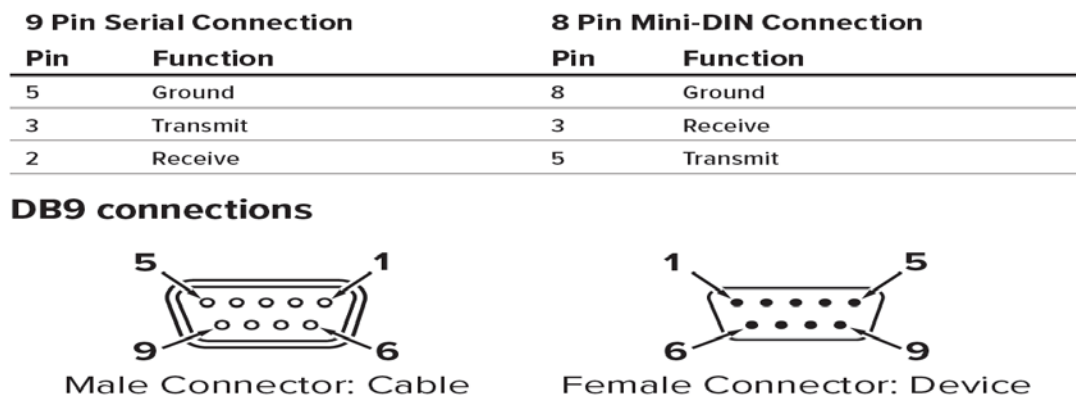
Standard 8-Pin Mini-DIN Pinout

Pin	Function
1	Not Connected (or optional 4–20 mA Primary Output Signal)
2	Static 5.12 Vdc (or optional Secondary Analog Output [4–20 mA, 0–5 Vdc, 1–5V dc, 0–10 Vdc] or Basic Alarm)
3	Serial RS-232RX / RS-485(–) Input Signal (receive)
4	Remote Tare (Ground to Tare)
5	Serial RS-232TX / RS-485(+) Output Signal (send)
6	0–5 Vdc (or optional 1–5 Vdc or 0–10 Vdc) Output Signal
7	Power In (as described above)
8	Ground (common for power, digital communications, analog signals and alarms)

Πίνακας 6. 8-Pin Mini-DIN Pinout - Πηγή [54]

Όπως αναγράφεται στον παραπάνω πίνακα (Πίνακας 6), ο πρώτος ακροδέκτης δεσμεύεται ή μπορεί να λειτουργήσει ως υποδοχή σήματος κυρίας εξόδου 4-20 mA, προσφέροντας με αυτόν τον τρόπο ευελιξία για μελλοντικές διαμορφώσεις. Ο δεύτερος ακροδέκτης παρέχει σταθερή DC τάση 5.12 V και συγχρόνως υποστηρίζει αναλογικές εξόδους: 4–20 mA, DC 0–5 V, DC 1–5 V ή DC 0–10 V, εξασφαλίζοντας προσαρμοστικότητα για διάφορα συστήματα. Μάλιστα, απαιτούνται τουλάχιστον DC 12 V και 100 mA για εξόδους αναλογικού σήματος 4–20 mA και DC 12 V για εξόδους DC 0–10 V. Οι ακροδέκτες 3 και 5 αποτελούν το Receive και το Transmit για την επικοινωνία της συσκευής μέσω του πρωτοκόλλου RS-232. Ο ακροδέκτης 4 χρησιμεύει στην απομακρυσμένη χρήση γείωσης αναφοράς, επιτρέποντας την προσαρμοστικότητα των μετρήσεων σε σχέση με ένα σημείο αναφοράς. Τέλος, ο ακροδέκτης 6 αποτελεί έξοδος για αναλογικά σήματα με DC τάσης εξόδου: 0–5 V, 1–5 V και 0–10 V, επιτρέποντας την χρήση διαφορετικών συσκευών. Ο ακροδέκτης 7 αποτελεί την υποδοχή τροφοδοσίας της συσκευής με DC τάση λειτουργίας: 9 έως 24 V με μέγιστη παραγόμενη ένταση μέχρι 50 mA. Η κοινή γείωση της συσκευής

τοποθετείται στον ακροδέκτη 8 του συνδέσμου 8-Pin Mini-DIN. Ο αισθητήρας χρησιμοποιεί καλώδιο, το οποίο μετατρέπει το 8-Pin Mini-DIN σε απόληξη DB9 Female Connector. Παρακάτω απεικονίζεται η αντιστοίχιση των εν προκειμένω ακροδεκτών (Σχήμα 2.7).



Σχήμα 2.7. Αντιστοίχιση των ακροδεκτών του DB9 σε Mini-Din - Πηγή [54]

2.3.3 Βασικές ρυθμίσεις

Ρυθμίσεις σύνδεσης

Στην οθόνη lcd του αισθητήρα παρέχεται άμεση απόκριση στην απεικόνιση δεδομένων που αφορούν το μετρούμενου αέριο, καθώς και μια πληθώρα ρυθμίσεων για την καλύτερη λειτουργία του αισθητήρα. Παράλληλα, καθίστανται εφικτές οι ανωτέρω λειτουργίες κι εξ' αποστάσεως μέσω της υποδοχής 8-Pin Mini-DIN to DB9. Συγκεκριμένα, με τη σύνδεση του αισθητήρα σε έναν υπολογιστή/μικροελεγκτή δίνεται η δυνατότητα καταγραφής των δεδομένων και η εκτέλεση όλων των επιμέρους ρυθμίσεων. Ο αισθητήρας συνδέεται κι επικοινωνεί ψηφιακά μέσω της υποδοχής επικοινωνίας της εκάστοτε συσκευής. Πριν την σύζευξη του αισθητήρα, απαιτείται η ρύθμιση των παραμέτρων της σειριακής επικοινωνίας, ούτως ώστε να καταστεί δυνατή η ανταλλαγή δεδομένων μεταξύ του αισθητήρα και του STM32F401RE, όπως εξετάζεται εν προκειμένω. Ο μετρητής ροής έχει προεπιλεγμένο ρυθμό, Baud Rate 19200 (bits ανά δευτερόλεπτο) [54]. Για το συγκεκριμένο πείραμα ορίζεται στα 9600, τα δεδομένα των Data bits, Parity, Stop bits, Flow control παραμένουν στις εργοστασιακές ρυθμίσεις όπως αναφέρονται παρακάτω:

Baud Rate: 9600,

Data bits: 8,

Parity: none, Stop bits: 1,

Flow control: none

Οι παραπάνω ενέργειες ρυθμίζονται από το μενού του αισθητήρα, επιλέγοντας COMM και μετά SETUP.

Unit ID

Το Unit ID είναι το αναγνωριστικό που χρησιμοποιεί ο υπολογιστής/μικροελεγκτής, για να διακρίνει τον αισθητήρα από έτερες συσκευές Alicat, σε περίπτωση που είναι συνδεδεμένοι περισσότεροι αισθητήρες σε ένα δίκτυο. Η λειτουργία αναγνώρισης ονομάζεται Polling mode, με άμεσο ακόλουθο αυτής την αποστολή δεδομένων, κάθε φορά που αυτή ζητείται, με την χρήση του Unit ID.

Το Unit ID ορίζεται με χαρακτήρες από A-Z. Με την επιλογή του στοιχείου "@" ως Unit ID, ο αισθητήρας εισέρχεται σε λειτουργία streaming mode (συνεχή μετάδοση). Στη λειτουργία αυτή, ο αισθητήρας αποστέλλει συνεχώς μια γραμμή δεδομένων (Data string) σε τακτά χρονικά διαστήματα, κάθε 50 ms, χωρίς να απαιτείται πρόσθετη ενέργεια του χρήστη. Ωστόσο, σε μια δεδομένη θύρα, ένας μόνο αισθητήρας μπορεί να βρίσκεται σε λειτουργία συνεχούς ροής κάθε φορά [54]. Η θέση της λειτουργίας αυτής είναι ιδιαίτερα χρήσιμη για το σύστημα, διότι επιτρέπει τη συνεχή και σε πραγματικό χρόνο παρακολούθηση των δεδομένων, με γρήγορη και διαρκή ανάδραση, ευνοώντας γρήγορες προσαρμογές σε μεταβαλλόμενες συνθήκες. Για τις ανάγκες του εν προκειμένω πειράματος, θα χρησιμοποιηθεί η λειτουργία streaming mode.

Επιλογή τύπου αεριού

Ο αισθητήρας βαθμονομήθηκε στο εργοστάσιο της Alicat. Η Gas Select Team εξασφάλισε τη δυνατότητα εκ νέου ρύθμισης του αισθητήρα, ώστε να ρέει κάθε φορά διαφορετικού τύπου αερίου, χωρίς να απαιτείται η επιστροφή στο εργοστάσιο για επαναβαθμονόμηση. Επομένως, το flow meter μπορεί να λάβει μετρήσεις για 98 διαθέσιμα αέρια [54].

Παράλληλα, προσφέρεται η λειτουργία σύνθεσης περισσότερων αερίων μέσω του Composer. Ειδικότερα, το Composer επιτρέπει την δημιουργία ενός νέου μίγματος με την πρόσμιξη έως και πέντε διαφορετικών αερίων. Η ανάλυση σύνθεσης μπορεί να οριστεί εντός 0,01% για καθένα από τα πέντε αέρια. Ο αισθητήρας ρυθμίζεται με βάση τις μοριακές (ογκομετρικές) αναλογίες των αερίων που εμπεριέχονται στο μείγμα. Μάλιστα, το νέο μείγμα αποθηκεύεται στη λίστα των αερίων (Gas Select) που εμπεριέχονται στον αισθητήρα, ενώ μπορούν να συντεθούν από το χρήστη μέχρι 20 μείγματα αερίων [54]. Στο παρόν πείραμα, ο τύπος του αερίου παραμένει στη προκαθορισμένη επιλογή "AIR", χωρίς να γίνει χρήση κάποιου νέου μίγματος ή κάποιου αεριού από το Gas Select.

Display as zero/ Taring flow

Η λειτουργία Display as Zero ρυθμίζεται χωριστά από τον χρήστη κι αναλόγως το εκάστοτε triggered point η τιμή της μέτρησης θα εμφανίζεται ως μηδενική στην οθόνη lcd. Ειδικότερα, όταν η τιμή της μέτρησης είναι κατώτερη του triggered point, ο αισθητήρας θα αναγράφει μηδενική ροή. Επιπλέον, η λειτουργία αυτή εφαρμόζεται σε μετρήσεις πίεσης κατά τη χρήση του προαιρετικού βαρόμετρου [54].

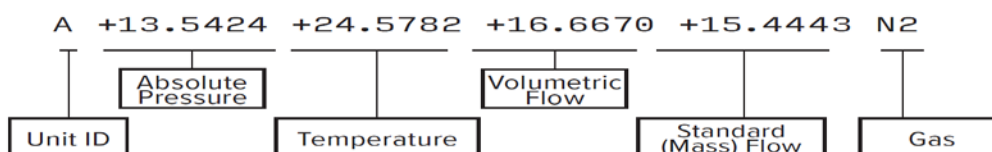
Εν προκειμένω, ο αισθητήρας, που χρησιμοποιείται για την υλοποίηση του πειράματος, έχει ρυθμιστεί με τιμή μηδενικής ζώνης 0,4%. Συνεπώς, εμφανίζονται ως 0,00 slpm όλες τις μετρήσεις κάτω από 0,1 slpm. Οι ρυθμίσεις μηδενικής ζώνης δεν επηρεάζουν τις τιμές που καταγράφονται μέσω του σειριακού τερματικού. Στη συνέχεια, πριν από κάθε νέα μέτρηση, θα πρέπει να ρυθμιστεί η ροή αναφοράς (taring flow).

Αυτή η λειτουργία δίνει στον αισθητήρα μια καλή μηδενική αναφορά, ούτως ώστε να μην υπάρχει απόκλιση μεταξύ των μετρήσεων. Η ρύθμιση αυτή επιτυγχάνεται με το πάτημα του κουμπιού TARE που βρίσκεται πάνω δεξιά στην οθόνη της συσκευής [54].

2.3.4 Συμβολοσειρά δεδομένων (Data string)

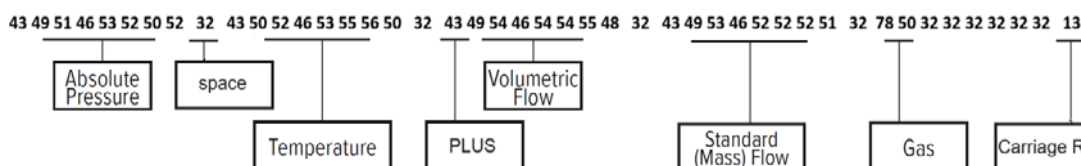
Το μήνυμα που αποστέλλεται από τον αισθητήρα στο σειριακό τερματικό παρέχει σε πραγματικό χρόνο ένα ολοκληρωμένο στιγμιότυπο των μετρήσεων και της κατάστασης της λειτουργίας του. Η συμβολοσειρά αυτή τυπικά περιλαμβάνει έξι στοιχεία, εκ των οποίων τα τέσσερα αποτελούν τιμές μετρήσεων. Πιο αναλυτικά, ξεκινώντας από τα αριστερά προς τα δεξιά η συμβολοσειρά περιλαμβάνει: το Unit ID, την απόλυτη πίεση, τη θερμοκρασία, την ογκομετρική ροή, την τυπική ροή μάζας και τέλος τον προεπιλεγμένο τύπο αερίου. Ωστόσο, στη λειτουργία Streaming mode δεν εμφανίζεται το Unit ID [54] (Σχήμα 2.8). Η διαφορά μεταξύ των δύο ροών έγκειται στο γεγονός ότι η ροή μάζας συνιστά τη μέτρηση του αριθμού των μορίων σε ρέον αέριο, ενώ η ροή όγκου αποτελεί τη μέτρηση του χώρου που καταλαμβάνουν τα μόρια αυτά.

Κάθε τιμή μέτρησης προσλαμβάνει θετικό ή αρνητικό πρόσημο, το οποίο ενδέχεται να μεταβληθεί ανάλογα με τις εκάστοτε υφιστάμενες συνθήκες. Αξίζει να υπογραμμιστεί ότι στο πείραμα που θα πραγματοποιηθεί δεν θα υπάρξουν αρνητικές τιμές. Κάθε στοιχείο της συμβολοσειράς χωρίζεται με κενό διάστημα από το επόμενο.



Σχήμα 2.8. Η κωδικοποιημένη συμβολοσειράς όπως εμφανίζεται στο σειριακό τερματικό του H/Y - Πηγή [54]

Η μετάφρασή του παραπάνω μηνύματος σε κώδικα ASCII καθίσταται απαραίτητη, προκειμένου να είναι πιο εύκολη η επεξεργασία του, αργότερα στο προγραμματισμό του μικροελεγκτή. Το μήνυμα στη δεκαδική του αναπαράσταση καθιστά εμφανή τα κενά διαστήματα που υπάρχουν μεταξύ των στοιχείων (στοιχείο 32 - "space"). Είναι σημαντικό να αναφερθεί ότι, τα συσσωρευμένα κενά διαστήματα που υπάρχουν στο πεδίο του αερίου (έξι στοιχεία του αριθμού 32) προσαρμόζονται (αυξάνονται/μειώνονται) ανάλογα με τον χώρο που καταλαμβάνει η ονομασία του αερίου που θα επιλεγθεί. Το μήνυμα τερματίζεται με την εντολή Carriage Return. Παρακάτω απεικονίζεται η δεκαδική μορφή του μηνύματος (Κώδικας ASCII) (Σχήμα 2.9).



Σχήμα 2.9. Απεικόνιση της συμβολοσειράς με δεκαδικούς αριθμούς

2.3.5 Status messages

Η ένδειξη κατάστασης κατ' αρχήν παρέχει σημαντικές πληροφορίες για τις μετρούμενες τιμές του εκάστοτε τύπου αερίου. Για την ακρίβεια, όταν οι τιμές της πίεσης, της ροής μάζας, της θερμοκρασίας και της ογκομετρικής ροής υπερβούν την εμβέλεια μέτρησης, στην οποία έχει σχεδιαστεί και προγραμματιστεί ο αισθητήρας, τότε ενδείξεις όπως POV, MOV, TOV και VOV (Over Range of Device) αντιστοίχως, ενσωματώνονται στα δεδομένα του μηνύματος που προωθεί ο αισθητήρας στην εκάστοτε συσκευή που είναι συνδεδεμένος. Ταυτόχρονα, ενδείξεις για την ομαλή λειτουργία του Totalizer προστίθενται στη λίστα των μηνυμάτων κατάστασης, όπως OVR (Totalizer Rolled Over to Zero) και TMF (Totalizer Missed out of Range Flow). Επιπλέον, μηνύματα αναφορικά με την κατάσταση της οθόνης κλειδώματος, για παράδειγμα LCK (Front Display is Locked), ειδοποιούν το χρήστη ότι η οθόνη του αισθητήρα έχει τεθεί σε αδράνεια [54].

Η κατάσταση κανονικής λειτουργίας της συσκευής επιβεβαιώνει στους χρήστες ότι όλα λειτουργούν όπως αναμένεται. Αντιθέτως, σε περίπτωση οποιονδήποτε προβλημάτων ή αποκλίσεων από το πρότυπο, τα μηνύματα κατάστασης ενδέχεται να ενσωματώνουν κωδικούς σφάλματος, όπως ο κωδικός ADC (Analog to Digital Converter Error) που δηλώνει την ύπαρξη προβλήματος στο μετατροπέα αναλογικού σε ψηφιακού σήματος [54]. Τα μηνύματα κατάστασης αποσκοπούν στη ευκολότερη αντιμετώπιση των προβλημάτων. Έτσι, ο χρήστης βασιζόμενος στα μηνύματα κατάστασης για άμεση ανατροφοδότηση, μπορεί να παρακολουθεί και να ελέγχει τη διαδικασία ροής του αερίου με απόλυτη ακρίβεια (Σχήμα 2.10).

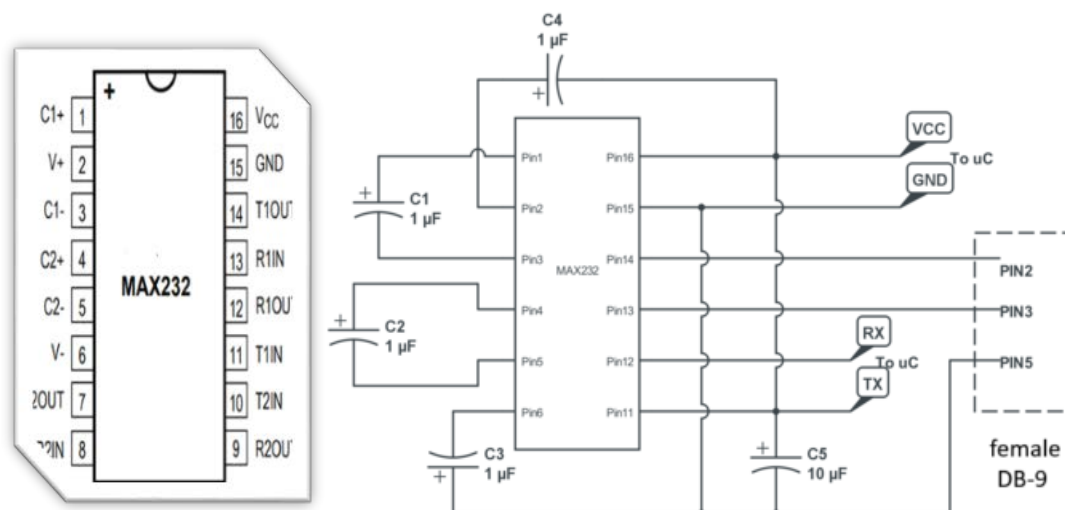
Status Messages			
ADC	Analog-digital converter error	MOV	Mass flow over range of device
LCK	Front display is locked	TMF	Totalizer missed out of range flow
OVR	Totalizer rolled over to zero	TOV	Temperature over range of device
POV	Pressure over range of device	VOV	Volumetric flow over range of device

Σχήμα 2.10. Τα μηνύματα καταστάσεις που χρησιμοποιεί ο αισθητήρας M-20SLPM-D - Πηγή [54]

2.4 IC MAX232 (RS232 - TTL Level Shifting)

Όπως αναφέρθηκε και προηγουμένως, τα επίπεδα τάσης λειτουργίας του πρωτοκόλλου RS-232 ανέρχονται έως και 25V. Οι περισσότεροι μικροελεγκτές, συμπεριλαμβανομένου και του STM32f401RE, έχουν υποδοχή σειριακής επικοινωνίας για σύνδεση συγχρόνων και ασύγχρονων πρωτοκόλλων. Η σειριακή πόρτα αποτελείται από το ολοκληρωμένο κύκλωμα USART (Universal Synchronous/Asynchronous Receiver/Transmitter), το οποίο παρέχει όλα τα απαραίτητα λογικά σήματα και διεπαφές για την ομαλή λειτουργία των πρωτοκόλλων [38]. Η USART σειριακή θύρα χρησιμοποιεί την τυπική λογική TTL/CMOS (Transistor to Transistor Logic), στην οποία οι τάσεις λειτουργίας κυμαίνονται από 0V έως 5V. Συγκεκριμένα, το Logic "LOW" αντιστοιχεί στην τιμή bit 0 και το λογικό "High" αντιστοιχεί στην τιμή bit 1 [55].

Για την εξομάλυνση των επιπέδων τάσης της λογικής TTL και του RS-232 θα χρησιμοποιηθεί το υλικό IC MAX232. Το MAX232 είναι ένα τσιπ DIL (dual in-line package) με τη δυνατότητα αμφίδρομης μετάδοσης σημάτων. Αποτελεί ένα δημοφιλές IC (Integrated circuit) για την μετατροπή των επιπέδων τάσης TTL σε RS232 και αντιστρόφως [36]. Παρακάτω απεικονίζεται το pinout του MAX232 και οι απαραίτητες συνδέσεις που πρέπει να πραγματοποιηθούν, προκειμένου να είναι λειτουργικό (Σχήμα 2.11).



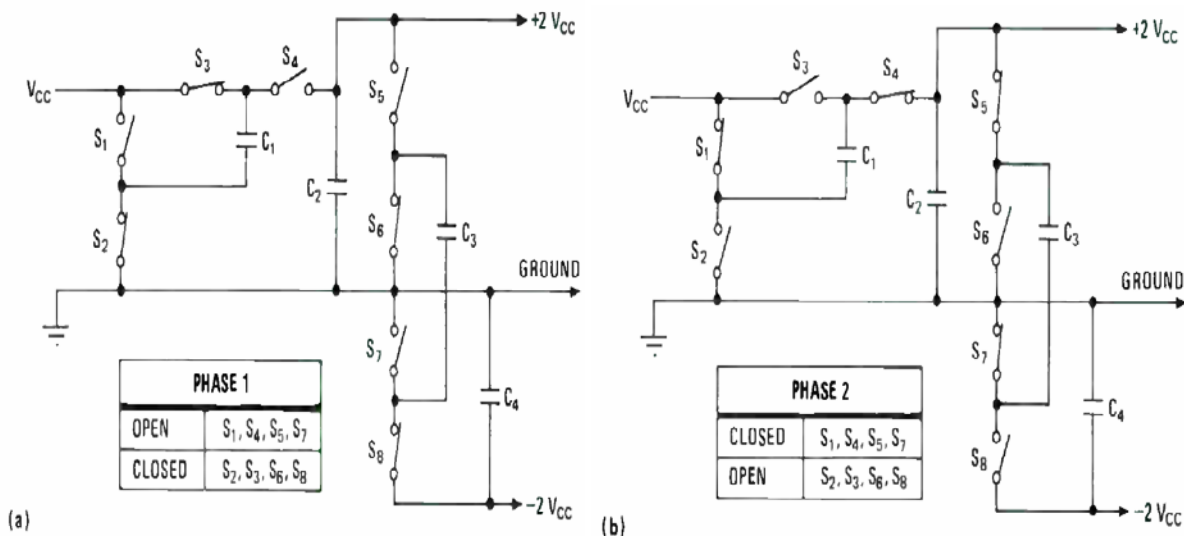
Σχήμα 2.11. Κυκλωματική απεικόνιση του MAX232 - Πηγή [57],[58]

2.4.1 Η λειτουργία του IC MAX232

Το IC MAX232 περιλαμβάνει δύο πομπούς, δύο δέκτες και δυο μετατροπείς αντλίας φόρτισης. Το IC μπορεί να λειτουργήσει με ένα μόνο τροφοδοτικό τάσεως 5V. Το γεγονός ότι το MAX 232 έχει κατασκευαστεί για να δέχεται και να στέλνει σήματα RS232, το καθιστά ικανό να αντέχει στη είσοδο του, τάσεις που ανέρχονται στα $\pm 30V$, ακόμη και όταν η συσκευή δεν είναι συνδεδεμένη στη τάση τροφοδοσίας. Οι δέκτες εισόδου έχουν υστέρηση 500 mV για την απαλοιφή του θορύβου. Αυτό που ξεχωρίζει το MAX232 είναι ότι συνδυάζει δύο λειτουργίες σε μία μόνο συσκευή. Ο ένας από τους μετατροπείς διπλασιάζει την τάση τροφοδοσίας και ο άλλος αλλάζει την πολικότητα της παραγόμενης τάσης, παράγοντας ταυτόχρονα θετικά και αρνητικά σήματα εξόδου. Τα κυκλώματα αντλίας φόρτισης φορτίζουν με μια τάση εισόδου έναν πυκνωτή. Στη συνέχεια, προσθέτοντας, αφαιρώντας ή αναστρέφοντας την τάση αυτή με θετική ή αρνητική λογική, επιτυγχάνεται η επιθυμητή τιμή. Το φορτίο αυτό μεταφέρεται σε έναν πυκνωτή, όπου και συγκρατείται.

Οι εξωτερικοί πυκνωτές στήριξης C2 και C4 δημιουργούν πολικότητα, ενώ οι πυκνωτές C1 και C3 διπλασιάζουν την τάση. Η διαδικασία μετατροπής του σήματος χωρίζεται σε δύο φάσεις. Πιο αναλυτικά, κατά την πρώτη φάση, ο πυκνωτής C1 φορτίζεται στην τάση τροφοδοσίας μέσω των διακοπών S2 και S3. Κατά τη δεύτερη φάση, η τάση στο C1 προστίθεται στην τάση τροφοδοσίας μέσω των διακοπών S1 και S4, παράγοντας ένα σήμα κατά μήκος του πυκνωτή C2 ίσο με δύο φορές την τάση τροφοδοσίας. Ταυτόχρονα, ο πυκνωτής C3 φορτίζεται σε παρόμοιο επίπεδο μέσω των διακοπών S5 και S7. Στη συνέχεια, αναστρέφεται ως προς το GRD μέσω των διακοπών S6 και S8, προκειμένου να παραχθεί αρνητική τάση διπλάσια της τάσης τροφοδοσίας κατά τη πρώτη φάση [58].

Οι διακόπτες συγχρονίζονται βάσει του εσωτερικού ρολογιού με ρυθμό 16 kHz. Παρακάτω απεικονίζονται οι συνδέσεις για κάθε φάση ξεχωριστά (Σχήμα 2.12).



Σχήμα 2.12. Απεικόνιση εσωτερικής λειτουργίας του MAX232 - Πηγή [58]

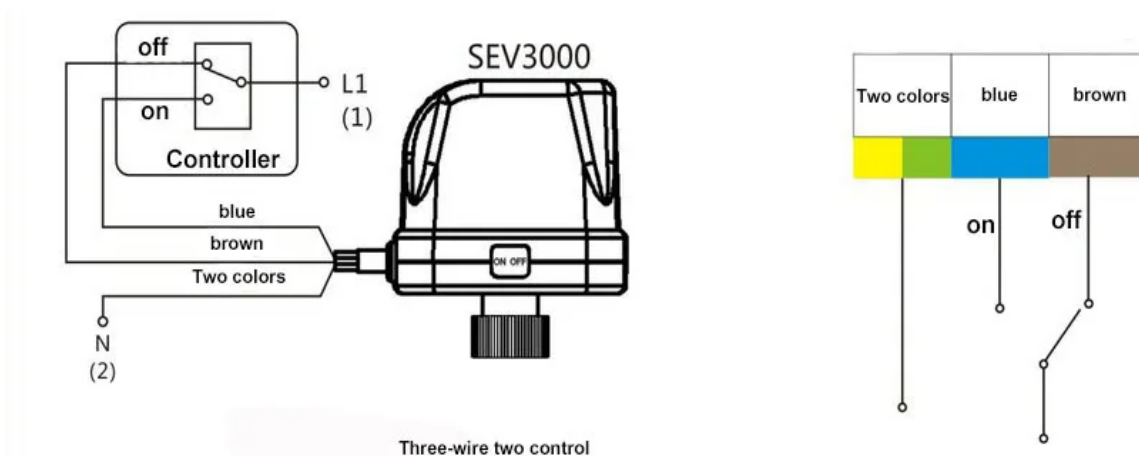
2.5 Ηλεκτρική σφαιρική βαλβίδα (Electric valve)

Η ηλεκτρική σφαιρική βαλβίδα αποτελείται από έναν ενεργοποιητή, ισχύος 6W (ηλεκτροκινητήρας) (Σχήμα 2.13) και μια σφαιρική βαλβίδα ροής (Σχήμα 2.14). Ένα σύστημα μετάδοσης μεταφέρει την κίνηση του ηλεκτροκινητήρα στην σφαιρική βαλβίδα ροής, επιτρέποντας έτσι την αυξομείωση της ροής.

2.5.1 Ενεργοποιητής

Η λειτουργία του ηλεκτροκινητήρα προϋποθέτει ρεύμα συνεχούς τάσης. Αυτό επιτυγχάνεται με τη χρήση κυκλώματος τροφοδοσίας (μετασηματιστής), το οποίο μετατρέπει το εναλλασσόμενο ρεύμα του δικτύου (230V AC, 50-60Hz) σε συνεχές, εν προκειμένω σε 12V. Ο ενεργοποιητής αποτελεί το κύριο σύστημα, επί του οποίου προσαρμόζεται η σφαιρική βαλβίδα ροής μέσω βιδωτού συνδέσμου. Ακολούθως, οι διαστάσεις της βαλβίδας προσαρμόζονται με βάση τη ποσότητα ρευστού που είναι αναγκαία για την εκάστοτε εφαρμογή.

Για την τροφοδοσία και τον έλεγχο του ενεργοποιητή συνδέεται ο ουδέτερος του δικτύου (N) στο καλώδιο δύο χρωμάτων (κίτρινο-πράσινο) του ενεργοποιητή. Ταυτόχρονα, η φάση (L) έχει δυο σημεία συνδέσεις και μεταφράζεται σε δύο καταστάσεις για τον ενεργοποιητή. Πιο αναλυτικά, συνδέοντας την φάση του δικτύου στο μπλε καλώδιο του ενεργοποιητή, το σύστημα αυξάνει βαθμιαία την ροή (Κατάσταση 1). Σε αντίθετη περίπτωση, συνδέοντας τη φάση του δικτύου στο καφέ καλώδιο του ενεργοποιητή το σύστημα σταδιακά θα διακόπτει τη ροή (Κατάσταση 2).

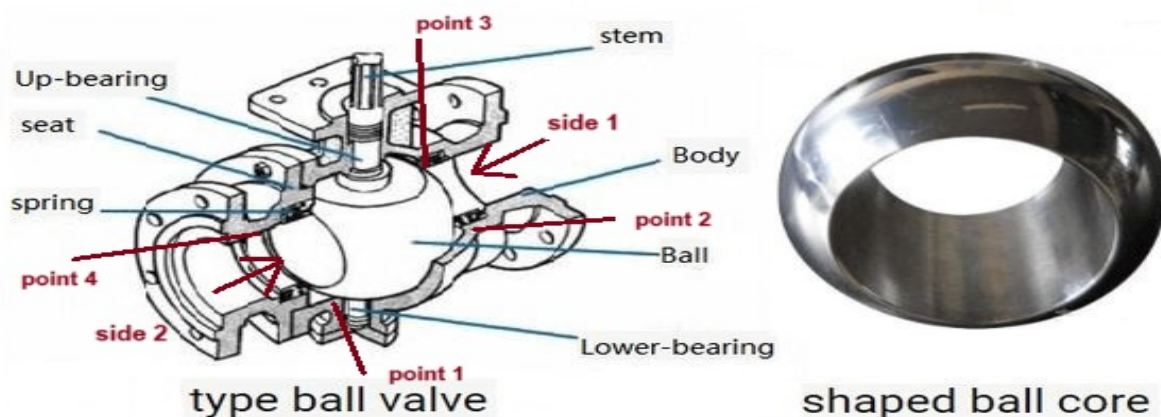


Σχήμα 2.13. Σχεδιάγραμμα συνδεσμολογίας ενεργοποιητή ηλεκτρικής σφαιρικής βαλβίδας - Πηγή [59]

Ο ενεργοποιητής τίθεται σε αδράνεια με την ταυτόχρονη τροφοδοσία και των δυο καταστάσεών του, καθώς και με τη διακοπή της τροφοδοσίας της εκάστοτε κατάστασης. Το σύστημα μπορεί να δεχθεί μόνο μια κατάσταση τη φορά και δεν μπορεί να ενεργοποιηθεί μια νέα κατάσταση προτού τερματιστεί η προηγούμενη. Κατά την έναρξη και τον τερματισμό του ενεργοποιητή υπάρχουν διακόπτες ορίου (ή οριοδιακόπτες), που ενεργοποιούνται και απενεργοποιούνται, προκειμένου να γνωστοποιήσουν στο σύστημα πότε μια κατάσταση έχει ολοκληρωθεί και είναι έτοιμο να εκτελέσει την επομένη.

2.5.2 Σφαιρική βαλβίδα ροής

Η σφαιρική βαλβίδα ροής αποτελείται από δύο πλευρές, έχει δύο καταστάσεις και τέσσερα σημεία, από τα οποία διέρχεται μέχρι να ολοκληρώσει ένα πλήρη κύκλο 360° και να επανέλθει στο αρχικό σημείο όπου ξεκίνησε. Κάθε κατάσταση έχει περιστροφή 90° και η βαλβίδα περιστρέφεται αριστερόστροφα.



Σχήμα 2.14. Απεικόνιση σφαιρικής βαλβίδας ροής - Πηγή [60]

Η αλληλουχία των ενεργειών για την εκτέλεση ενός κύκλου 360° της σφαιρικής βαλβίδας είναι η ακόλουθη. Ειδικότερα, εκκινώντας σε κλειστή θέση, η σφαιρική βαλβίδα ροής βρίσκεται στην κατάσταση δύο (2) (κλειστός θάλαμος), ενεργοποιώντας την κατάσταση ένα (1) το σύστημα περιστρέφεται προς τα αριστερά μέχρι να προσεγγίσει το σημείο ένα (1), οπότε και ολοκληρώνεται η κατάσταση αυτή (ανοικτός θάλαμος). Στο σημείο αυτό, η πλευρά ένα της βαλβίδας είναι πλήρως ανοιχτή, με αποτέλεσμα η ροή να μεγιστοποιείται. Στη συνέχεια, απενεργοποιώντας την κατάσταση ένα (1) και ενεργοποιώντας την κατάσταση δύο (2), τότε η βαλβίδα συνεχίζει την κίνηση της προς τα αριστερά, μέχρι να προσεγγίσει το σημείο δύο (2), οπότε ολοκληρώνεται η κατάσταση δύο (2). Πλέον, η βαλβίδα έχει επανέλθει στην αρχική της κατάσταση, έχοντας εκτελέσει μια στροφή 180°.

Όταν απενεργοποιηθεί η κατάσταση δύο (2) και ενεργοποιηθεί η κατάσταση ένα (1), το σύστημα ξεκινάει την περιστροφή του έως ότου προσεγγίσει το σημείο τρία (3), οπότε ολοκληρώνεται η κατάσταση ένα (1). Έτσι, η πλευρά δύο (2) της βαλβίδας είναι πλήρως ανοιχτή και η ροή μεγιστοποιείται.

Όταν απενεργοποιηθεί η κατάσταση ένα (1) και ενεργοποιηθεί η κατάσταση δύο (2), τότε η βαλβίδα συνεχίζει την κίνηση της αριστερόστροφα μέχρι να προσεγγίσει στο σημείο τέσσερα (4), οπότε ολοκληρώνεται η κατάσταση δύο (2). Σε αυτό το σημείο η βαλβίδα έχει επανέλθει στην αρχική της θέση (σημείο τέσσερα (4)) εκτελώντας μια πλήρη περιστροφή 360°. Η διαδικασία αυτή επαναλαμβάνεται ανάλογα με την κατάσταση και την πλευρά που είναι επιθυμητή τη δεδομένη χρονική στιγμή.

3 ΚΕΦΑΛΑΙΟ 3^ο : ΜΕΘΟΔΟΛΟΓΙΑ ΠΟΥ ΑΚΟΛΟΥΘΗΘΗΚΕ

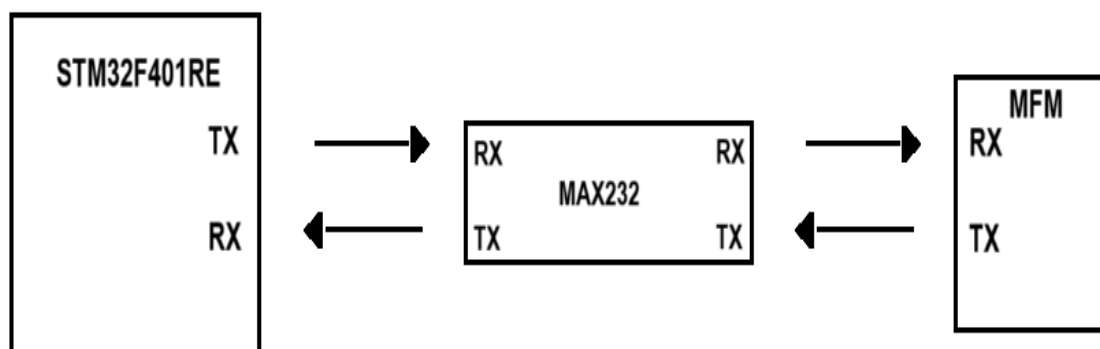
Σε αυτό το κεφάλαιο περιγράφεται η μεθοδολογία που ακολουθήθηκε για τον σχεδιασμό του του ελεγκτή ροή μάζας. Το πείραμα αρχίζει με την ανάπτυξη της κυκλωματικής διάταξης του συστήματος. Προκειμένου να επιτευχθεί η επικοινωνία του αισθητήρα με τον μικροελεγκτή δημιουργήθηκε ένα βοηθητικό πρόγραμμα με τη χρήση του STM32CubeIDE. Αφού πραγματοποιήθηκε μια πρώτη προεργασία στο σύστημα και επιβεβαιώθηκε ότι τα μέρη επικοινωνούν αρμονικά μεταξύ τους, η διαδικασία προχώρησε στο στάδιο των μετρήσεων. Η σφαιρική βαλβίδα βαθμονομήθηκε χρησιμοποιώντας δυο μεθοδολογίες μετρήσεων. Τέλος, τα αποτελέσματα χρησιμοποιήθηκαν για την υλοποίηση του κυρίως προγράμματος οδήγησης του συστήματος.

3.1 Κυκλωματική διάταξη ελεγκτή ελέγχου ροής

Η υλοποίηση του τελικού κυκλώματός του ελεγκτή ροή μάζας διακρίνεται σε δυο κύρια μέρη: την σύνδεση του αισθητήρα μέτρησης ροής μάζας και την σύνδεση του κυκλώματος οδήγησης της ηλεκτρικής σφαιρικής βαλβίδας με τον μικροελεγκτή. Παρακάτω αναλύεται η δομή και η λειτουργία των βασικών κυκλωμάτων, καθώς και τα εξαρτήματα που χρησιμοποιήθηκαν για να υλοποιηθούν οι συνδέσεις.

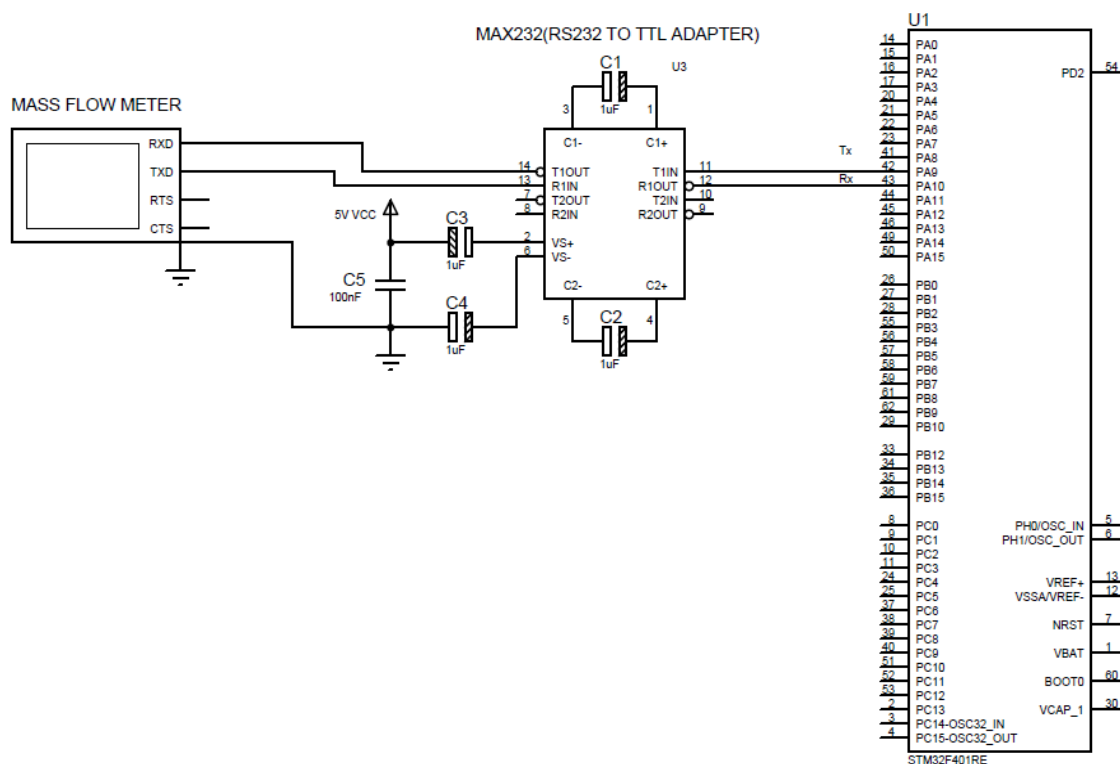
3.1.1 Κυκλωματική διάταξη μικροελεγκτή-αισθητήρα

Σε πρώτο στάδιο θα πρέπει να συνδεθεί ο μικροελεγκτής με τον αισθητήρα της Alicat, ούτως ώστε το σύστημα να επεξεργάζεται την τιμή της εκάστοτε κατάστασης ροής, καθ' όλη τη διάρκεια της λειτουργίας του. Για την πραγματοποίηση της επικοινωνίας αυτής θα γίνει χρήση του IC MAX232. Στη προκειμένη περίπτωση δεν απαιτείται έλεγχος ροής δεδομένων, επειδή η ροή των δεδομένων είναι συνεχής με σταθερή συχνότητα αποστολής μεταξύ των δηγμάτων (50ms default συχνότητα αποστολής του αισθητήρα). Συνεπώς, απαιτούνται μόνο γραμμές σήματος TX και RX και μια γραμμή γείωσης (GND) για μετάδοση και λήψη δεδομένων. Η ακίδα TX του μικροελεγκτή συνδέεται στην υποδοχή IC του μετατροπέα τάσης MAX232 και η έξοδος αυτού συνδέεται στην είσοδο του αισθητήρα. Ομοίως, η έξοδος της εξωτερικής συσκευής του αισθητήρα συνδέεται με την είσοδο RX του μικροελεγκτή μέσω του IC MAX232. Το μπλοκ διάγραμμα του κυκλώματος φαίνεται στο παρακάτω σχήμα (Σχήμα 3.1).



Σχήμα 3.1. Μπλόκ διάγραμμα STM32F401RE-MFM

Πιο αναλυτικά, ο ακροδέκτης PA9 (TX) της σειριακή πόρτας USART1 του μικροελεγκτή συνδέεται στην υποδοχή 11 (TX1 IN) του IC του μετατροπέα MAX232. Ο ακροδέκτης 14 (T1 Out or female DB-9, pin 2) του IC συνδέεται στην RX υποδοχή του αισθητήρα (female 8-pin mini-din, pin 3). Εξίσου, η έξοδος TX (pin 5) της συσκευής του αισθητήρα συνδέεται με την υποδοχή εισόδου 13 (RX1 IN or female DB-9, pin 3) του IC MAX232 και ο ακροδέκτης υποδοχής εξόδου 12 (RX1 Out) συνδέεται με τον ακροδέκτη εισόδου PA10 (RX) της σειριακή πόρτας USART1 του μικροελεγκτή. Η τροφοδοσία του IC MAX232 πραγματοποιείται μέσω της εξωτερικής τροφοδοσίας των 5V (CN6, pin 5) του αναπτυξιακού (STM32F401RE). Το σύστημα του αισθητήρα (female 8-pin mini-din, pin 8) και MAX232 (pin 15) γειώνονται αμφότερα στο αναπτυξιακό. Το πλήρες διάγραμμα κυκλώματος φαίνεται στο παρακάτω σχήμα (Σχήμα 3.2).

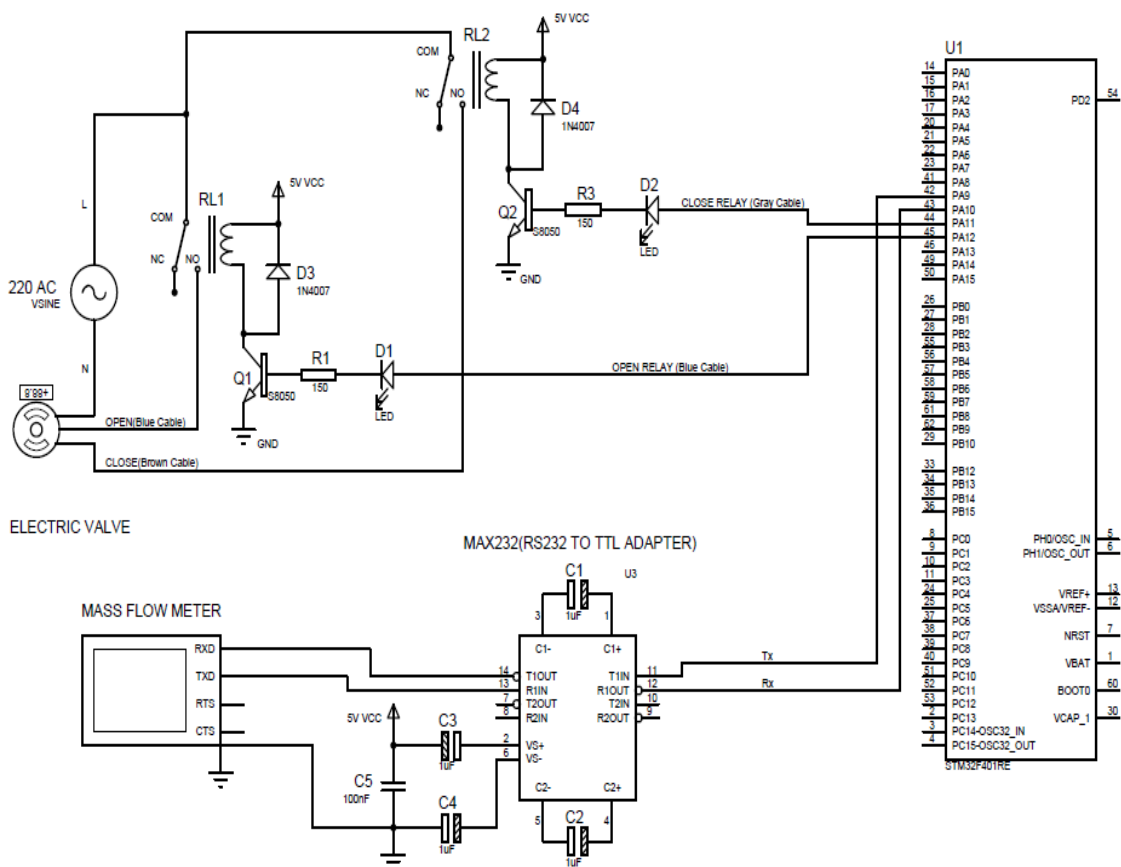


Σχήμα 3.2. Κυκλωματική διάταξη STM32F401RE-MFM

3.1.2 Κύκλωμα ελέγχου ηλεκτρικής σφαιρικής βαλβίδας με χρήση relays

Η ηλεκτρική σφαιρική βαλβίδα έχει τάση τροφοδοσίας 230V AC, η τάση αυτή καθορίζει την κίνησή της, ανάλογα την κατάσταση στη οποία θα εφαρμοστεί. Η χρήση των ηλεκτρονόμων αποτελεί ένα απλό τρόπο για τον διαχωρισμό της τάσης του δικτύου από το υπόλοιπο σύστημα. Στο κύκλωμα οδήγησης χρησιμοποιήθηκαν δυο ηλεκτρονόμοι τύπου Module KY-019, ένας για κάθε εντολή κατάστασης (ανοίγματος-κλεισίματος). Ειδικότερα, το Relay Module KY-019 λειτουργεί σε επίπεδο τάσης τιμής 5V, γεγονός που επιτρέπει σε μικροελεγκτές, όπως το STM32F401RE, τον έλεγχο αυτού, χωρίς να απαιτείται επιπρόσθετη τροφοδοσία. Οι ηλεκτρονόμοι αυτού του τύπου αντέχουν μέγιστη τιμή τάσης 250V AC έως 10A κι ενσωματώνονται κατ' αρχήν σε μια pcb πλακέτα, η οποία περιλαμβάνει ήδη μια αντίσταση, μια φωτοδίοδο, μια δίοδο κι ένα τρανζίστορ. Συγκεκριμένα, η αντίσταση διασφαλίζει τη μείωση της τάσης κατά την είσοδο του σήματος στη μονάδα.

Η φωτοδίοδος αποτελεί ένα οπτικό ερέθισμα για το χρήστη, έτσι ώστε να γνωρίζει πότε το ρελέ είναι ενεργοποιημένο ή όχι. Ακολουθώντας, το τρανζίστορ χρησιμοποιείται ως διακόπτης, για να ενεργοποιεί το ρελέ, όταν δεχθεί το σήμα ελέγχου. Η δίοδος (flyback) αποτελεί μια δικλείδα ασφαλείας, η οποία συνδέεται κατά μήκος του ηλεκτρομαγνητικού πηνίου του ρελέ κι εμποδίζει την τάση να επιστρέψει στην ακίδα εξόδου. Η μονάδα αποτελείται από τρεις ακροδέκτες: το VCC (τροφοδοσία), το SIG (σήμα ελέγχου) και το GND (γείωση). Ο ακροδέκτης VCC επιτρέπει την παροχή τάσης στη μονάδα (5V), ενώ ο ακροδέκτης SIG χρησιμοποιείται για τον έλεγχο του ηλεκτρονόμου κι απαιτεί ψηφιακό σήμα (HIGH ή LOW) από το STM32F401RE για την ενεργοποίηση ή απενεργοποίηση του ρελέ. Ο ακροδέκτης GND συνδέεται με τη γείωση του μικροελεγκτή. Το συνολικό κύκλωμα του ελεγκτή απεικονίζεται στο παρακάτω σχήμα (Σχήμα 3.3).



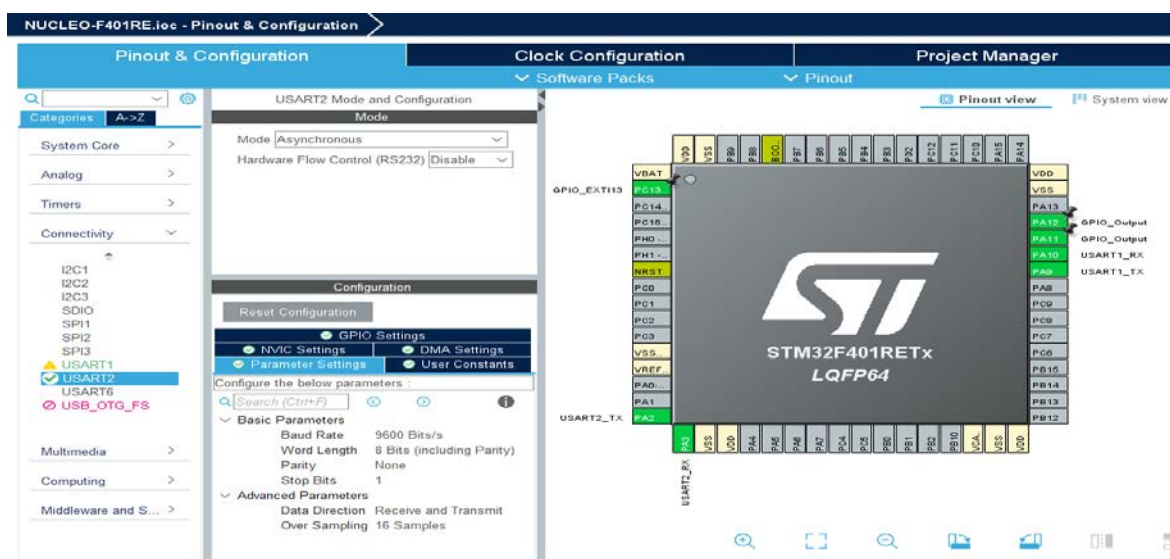
Σχήμα 3.3. Κυκλωματική διάταξη συστήματος ελεγκτή ροής μάζας

Για τον έλεγχο των δύο καταστάσεων δεσμεύονται δύο ακροδέκτες γενικής χρήσης (GPIO). Το πρώτο ρελέ χρησιμοποιείται στην κατάσταση ανοίγματος και συνδέεται στον ακροδέκτη PA 12, ενώ το δεύτερο ρελέ χρησιμοποιείται στην κατάσταση κλεισίματος και συνδέεται στον ακροδέκτη PA11. Η φάση του δικτύου ενώνεται στα δύο com των ρελέ ένα (RL1) και δύο (RL2), ανάλογα με το σήμα ελέγχου (SIG) που θα δοθεί από τους ακροδέκτες PA11, 12 θα ενεργοποιηθεί το αντίστοιχο ρελέ και θα εκτελεστεί η ανάλογη κατάσταση.

3.2 STM32CubeIDE

Το STM32CubeIDE αποτελεί ένα ολοκληρωμένο περιβάλλον ανάπτυξης, σχεδιασμένο για το προγραμματισμό της οικογενείας των μικροελεγκτών STM32. Χάρη σ' αυτό γίνεται εύκολη η αρχικοποίηση και διαμόρφωση του μικροελεγκτή, λαμβάνοντας υπόψιν ότι υποστηρίζει αρκετές γλώσσες προγραμματισμού, συμπεριλαμβανομένων των "C" και "C++". Επιπλέον, περιλαμβάνει προηγμένες δυνατότητες, όπως εύκολο και γρήγορο μενού για την επιλογή των ακροδεκτών του (σχηματική απεικόνιση), ανάλυση κώδικα σε πραγματικό χρόνο και ενσωματωμένα εργαλεία εντοπισμού σφαλμάτων.

3.2.1 Εισαγωγή στο STM32CubeIDE

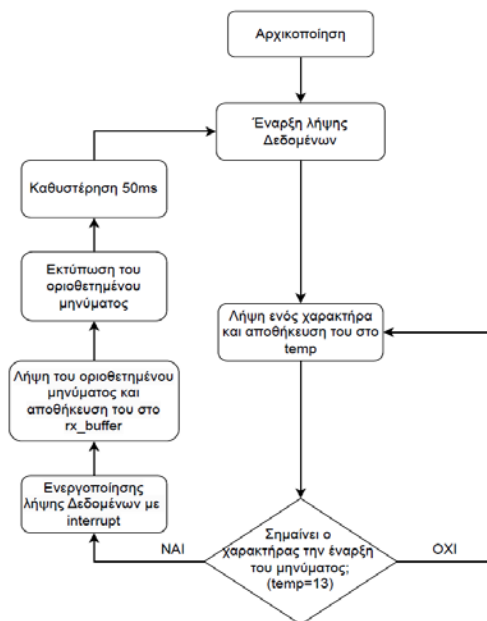


Σχήμα 3.4. Στιγμιότυπο από το περιβάλλον STM32CubeIDE

Για τις ανάγκες του πειράματος, ενεργοποιήθηκαν οι δυο σειριακές πόρτες USART 1, 2, προκειμένου να επιτευχθεί η επικοινωνία μικροελεγκτή-αισθητήρα (USART 1) και η λήψη/προώθηση των δεδομένων στο σειριακό τερματικό του υπολογιστή (USART 2). Η αρχικοποίηση των παραμέτρων της σειριακής επικοινωνίας έλαβε χώρα και για τις δυο θύρες, όπως ακριβώς είχε ρυθμιστεί ο αισθητήρας (2.3.3 Ρυθμίσεις Σύνδεσης). Οι ακροδέκτες που δεσμεύονται είναι οι PA9 και PA10 για τη USART1 και οι PA2, PA3 για τη USART2. Επιπλέον, ορίστηκαν ως έξοδοι οι πόρτες γενικής χρήσης των ακροδεκτών PA11 και PA12, οι οποίοι ελέγχουν το κύκλωμα οδήγησης της σφαιρικής βαλβίδας, όπως φάνηκε και στη κυκλωματική διάταξη του ελεγκτή (3.1.2 Κύκλωμα ελέγχου ηλεκτρικής σφαιρικής βαλβίδας με χρήση relays). Στην συνέχεια, ενεργοποιήθηκε ο ακροδέκτης PC13, διότι είναι συνδεδεμένος με το ενσωματωμένο μπουττόν του μικροελεγκτή. Ο ακροδέκτης αυτός τέθηκε σε λειτουργία εξωτερικής διακοπής με falling edge trigger detection. Εν προκειμένω, η ενεργοποίηση του ακροδέκτη PC13 αποτελεί απαραίτητη προϋπόθεση για τη διεξαγωγή των μετρήσεων, όπως θα περιγράψει στα επόμενα κεφάλαια. Τέλος, στο πεδίο "NVIC" των διακοπών, ενεργοποιήθηκαν οι διακοπές των δυο USART και το EXTI line[15:10] για το μπουττόν. Το κύριο πρόγραμμα του μικροελεγκτή συμπεριλαμβάνει τις ανωτέρω ρυθμίσεις.

3.2.2 Δημιουργία περιβάλλοντος επικοινωνίας με τον αισθητήρα

Το πρόγραμμα που δημιουργήθηκε για την επικοινωνία του αισθητήρα με τον μικροελεγκτή επισυνάπτεται στο παράρτημα Α. Στη συνέχεια περιγράφεται αναλυτικά η δομή και η λειτουργία του προγράμματος χωρισμένη σε τέσσερα (4) επιμέρους σκέλη: λήψη και αποστολή του μηνύματος του αισθητήρα, αντιμετώπιση Over Flow του μηνύματος αισθητήρα, εκτύπωση δεδομένων, μετατροπή της αναγνώσιμης τιμής σε επεξεργάσιμη. Παρακάτω φαίνεται το σχετικό μπλοκ διάγραμμα (Σχήμα 3.5).



Σχήμα 3.5. Μπλόκ διάγραμμα για το πρόγραμμα περιβάλλοντος επικοινωνίας με τον αισθητήρα

Λήψη και αποστολή του μηνύματος του αισθητήρα

Ο αισθητήρας αποστέλλει μήνυμα της τάξεως των σαράντα (40) bytes, εκ των οποίων τα τριάντα εννέα (39) bytes συνιστούν την ποσότητα του μηνύματος και το ένα (1) byte το "Carriage return", όπως έχει προαναφερθεί (2.3.4 Συμβολοσειρά δεδομένων (Data string)). Με την χρήση της εντολής "HAL_UART_Receive_IT" το ανωτέρω μήνυμα λαμβάνεται από τον μικροελεγκτή μέσω του UART1 (RS-232, D-9 Cable to TTL) και αποθηκεύεται σε ένα πίνακα "rx_buffer" σαράντα (40) θέσεων. Ακολούθως, η εντολή "HAL_UART_Transmit" εκτυπώνει το μήνυμα μέσω του UART2 (Micro-USB to USB-A Cable) στο σειριακό τερματικό του υπολογιστή.

Αντιμετώπιση Over Flow του μηνύματος αισθητήρα

Όταν ο αισθητήρας ανέλθει στη μέγιστη δυνατή τιμή, που μπορεί να προσλάβει για οποιαδήποτε από τις μετρούμενες μεταβλητές (πχ. 25.6 SLPM, για τη ροή), ή όταν ανακλύσει πρόβλημα ή error με τον Totallizer ή την οθόνη κλειδώματος, τότε ο αισθητήρας εμφανίζει και την σχετική ένδειξη μέσω του μηνύματος των δεδομένων, για να δηλώσει την κατάσταση στην οποία βρίσκεται (2.3.5 Status messages). Με την ενέργεια αυτή, το μήνυμα του αισθητήρα θα αυξηθεί. Συνεπώς, το μέγεθος του πίνακα που είχε οριστεί αρχικά, στον οποίο αποθηκεύονται τα δεδομένα (rx_buffer), να μην είναι επαρκές σε χωρητικότητα για το νέο μήνυμα που προκύπτει, δηλαδή το κανονικό μήνυμα των σαράντα (40) θέσεων και επιπλέον το μήνυμα κατάστασης. Τα νέα bytes θα αποθηκευτούν στις αρχικές θέσεις του επομένου πίνακα, με αποτέλεσμα να γίνεται συνεχώς over flow και μονίμως να χάνονται δεδομένα.

Η χρήση της συνάρτησης "HAL_UART_Receive_IT" γίνεται μια μόνο φορά στη αρχή της "main" και πριν τη "while(1)". Η "Callback" (HAL_UART_RxCpltCallback) καλείται κάθε φορά που υπάρχει διαθέσιμο μήνυμα, προκειμένου να μην διακοπεί η επικοινωνία με τον αισθητήρα. Παράλληλα, καλείται κι ένας βρόχος ελέγχου (while-break), ο οποίος διακρίνει το επιθυμητό προς εκτύπωση μήνυμα. Συγκριμένα, ο βρόχος χρησιμοποιεί μια Blocking mode Receive (HAL_UART_Receive), η οποία, μετά τη φόρτωση κάθε byte στον buffer, μετράει συνεχώς το complete flag μέχρι να ολοκληρωθεί η ανάγνωση. Στην πραγματικότητα, ενώ ακόμη μεταδίδονται δεδομένα, το πρόγραμμα οδήγησης αποκλείει οποιονδήποτε άλλο κώδικα από τη χρήση του μικροελεγκτή.

Τα bytes του μη επιθυμητού μηνύματος αποθηκεύονται μεμονωμένα με τη χρήση της "HAL_UART_Receive" σε μια προσωρινή μεταβλητή "temp". Όταν έρθει η σειρά του τελευταίου byte (Carriage return (13 για DEC)), τότε ενεργοποιείται το break (τερματισμός του βρόχου) και ξεκινάει το επόμενο Receive (HAL_UART_Receive_IT), για να αναγνωστεί το καινούργιο επιθυμητό μήνυμα. Ουσιαστικά, με αυτό το τρόπο, ελέγχεται το μήνυμα κι οροθετείται. Δηλαδή, αν αυτό δεν έχει την σωστή δομή, σύμφωνα με την οποία ο τελευταίος χαρακτήρας πρέπει να είναι CR, τότε θα αποθηκευτεί στην προσωρινή μεταβλητή "temp". Επομένως, το μήνυμα αυτό θα χαθεί. Έτσι, το επόμενο μήνυμα, δηλαδή το σωστά δομημένο, θα αποθηκευτεί στον "rx_buffer".

Για την αποφυγή διαγράψης του οριοθετημένου μηνύματος από τον βρόχο ελέγχου, ο "rx_buffer" ορίζεται τελικά, σε τριάντα ενεά (39) θέσεις. Συνεπώς, όταν αναγνωστεί το μήνυμα των σαράντα (40) bytes με την χρήση της "HAL_UART_Receive", το τελευταίο byte του μηνύματος (Carriage return) παραμένει στον αισθητήρα. Έτσι, γίνεται ανάγνωση μόνο των πρώτων τριάντα ενεά (39) bytes. Στο επόμενο μήνυμα, ο αισθητήρας αποστέλλει μόνο το byte του Carriage return. Στη συνέχεια, όταν αυτό εισέρθει στον βρόχο ελέγχου while-break, θα είναι το μοναδικό στοιχείο που θα αποθηκευτεί στην προσωρινή μεταβλητή "temp". Συνεπώς, είναι το μόνο που θα χαθεί.

Εκτύπωση δεδομένων

Για την μετάδοση των ληφθέντων δεδομένων του αισθητήρα στο σειριακό τερματικό του υπολογιστή, έγινε αντιστοίχιση της UART2 στη συνάρτηση "printf()". Πιο αναλυτικά, ορίστηκε η μακροεντολή "PUTCHAR_PROTOTYPE", η οποία προγραμματίστηκε να λαμβάνει μια ακέραια παράμετρο ή έναν ακέραιο κι έναν δείκτη αρχείου ως παραμέτρους και να επιστρέφει έναν ακέραιο. Σε κάθε περίπτωση, η λειτουργία αυτή χρησιμοποίησε την συνάρτηση "HAL_UART_Transmit" για τη μετάδοση των bytes στο σειριακό τερματικό του υπολογιστή μέσω του UART2. Για να καταστεί δυνατή η ανάγνωση του μηνύματος, τα τριάντα ενεά (39) στοιχεία του "rx_buffer" διακρίνονται σε τμήματα ανάλογα με το μέγεθος της τιμής προς εκτύπωση: 0-7 θέση για την απόλυτη πίεση, 8-15 για την θερμοκρασία, 16-23 για την ογκομετρική ροή, 24-35 για την τυπική ροή (μάζας) και 36-39 για τον τύπο του αερίου. Αξίζει να τονιστεί ότι χρησιμοποιήθηκαν δεκατρία (13) "printf()" για την εκτύπωση των δεδομένων τόσο σε μορφή ASCII όσο και σε δεκαδική μορφή, ούτως ώστε να είναι ευανάγνωστα στον χρήστη και να μπορέσει να τα επεξεργαστεί. Οι ενδιάμεσες γραμμές χρησιμεύουν ως διαχωριστικό, δημιουργώντας ένα σαφές όριο μεταξύ των διαφορετικών συνόλων των δεδομένων.

Μετατροπή της αναγνώσιμης τιμής σε επεξεργάσιμη

Στη συνέχεια του παρόντος πειράματος, δημιουργήθηκε μια συνάρτηση, η οποία μετατρέπει την τιμή που εκτυπώνεται στο σειριακό τερματικό, δηλαδή ένα σύνολο μη επεξεργασμένων bytes, μετατρέπεται σε μια πιο χρησιμοποιήσιμη μορφή κινητής υποδιαστολής για περαιτέρω επεξεργασία. Ειδικότερα, για τον προσδιορισμό του αρχικού δείκτη του σχετικού πεδίου δεδομένων εντός του "rx_buffer χρησιμοποιήθηκε μια δήλωση διακόπτη 1-4. Στην πράξη, δημιουργήθηκε μια παράμετρος επιλογής, η οποία αντιστοιχεί στα διαφορετικά τμήματα των δεδομένων του αισθητήρα, όπως η απόλυτη πίεση, η θερμοκρασία, η ογκομετρική ροή, η τυπική ροή (μάζας) και ο τύπος του αερίου. Η ανωτέρω συνάρτηση διαβάζει τα αντίστοιχα κομμάτια του rx_buffer, λαμβάνει το ακέραιο και το δεκαδικό μέρος κάθε μεταβλητής (ακόμα σε πίνακες συμβολοσειρών 16 χαρακτήρων), μετατρέπει αμφότερα τα μέρη σε ακέραιους με την χρήση της συνάρτησης "atritude to integer" (atoi) και σχηματίζει έναν αριθμό κινητής υποδιαστολής (float), συνδυάζοντάς τα.

Το ακέραιο μέρος αντιπροσωπεύει το αντίστοιχο τμήμα της μέτρησης, ενώ το κλασματικό μέρος αντιπροσωπεύει το δεκαδικό τμήμα. Ο αριθμός κινητής υποδιαστολής που προκύπτει κατασκευάστηκε, προσθέτοντας το ακέραιο μέρος στο κλασματικό μέρος, διαιρούμενο με το 100.0, προκειμένου να τοποθετηθεί σωστά η υποδιαστολή. Στη περίπτωση που δηλωθεί τιμή που δεν ανήκει στο πεδίο τιμών που δέχεται η συνάρτηση ($1 \leq \text{choice} \leq 4$), τότε η συνάρτηση επιστρέφει -1 και εκτυπώνεται μήνυμα error "ERROR @ convert_to_float, choice must be $1 \leq \text{choice} \leq 4$ ".

3.3 Μετρήσεις

Το κύριο ζήτημα εστιάζεται στον τρόπο, με τον οποίο ο μικροελεγκτής θα πρέπει να διαχειρίζεται τη λειτουργία της ηλεκτρικής σφαιρικής βαλβίδας, ανάλογα με τις απαιτήσεις του συστήματος. Η ηλεκτρική σφαιρική βαλβίδα δέχεται μόνο ως σήμα ελέγχου την τάση τροφοδοσίας της. Λαμβάνοντας αυτό ως μοναδικό κριτήριο χειρισμού, δημιουργείται η απαίτηση να βρεθεί ο χρόνος για τον οποίο θα εφαρμοστεί αυτό το σήμα και θα επιφέρει την ανάλογη μεταβολή σε ροή μάζας. Για το λόγο αυτό, έγινε βαθμονόμηση της ηλεκτρικής σφαιρικής βαλβίδας, εκτελώντας μετρήσεις σε ένα εύρος χρονικών καθυστερήσεων. Συγκεκριμένα, ακολουθηθήκαν δυο διαφορετικά είδη μετρήσεων για την καλύτερη απόδοση του συστήματος. Αξίζει να σημειωθεί ότι κατά τη διάρκεια δοκιμαστικών μετρήσεων διαπιστώθηκε ότι υπάρχουν κάποιες ιδιαιτερότητες που χαρακτηρίζουν την σφαιρική βαλβίδα, γεγονός το οποίο καθορίζει την τελική διαμόρφωση του συστήματος. Πριν ξεκινήσει οποιαδήποτε μέτρηση, ρυθμίζεται κατάλληλα η παροχή, προκειμένου να προωθούνται στο σύστημα 25 slpm ροής.

3.3.1 Παροχή

Η παροχή αποτελείται από ένα συμπιεστή αέρος, το οποίο είναι σχεδιασμένο να λειτουργεί σε υψηλές αποδόσεις, προσφέροντας αξιοπιστία σε μια πληθώρα εφαρμογών. Τροφοδοτείται από ένα μονοφασικό ηλεκτροκινητήρα με ισχύ 1.55 hp και ενσωματώνει αεροφυλάκιο χωρητικότητας 50 λίτρων. Με αυτό τον τρόπο, προσφέρεται αποθηκευτικός χώρος για αυτονομία σε καταστάσεις αδράνειας ή τερματισμού λειτουργίας του ηλεκτροκινητήρα. Ο συμπιεστής αέρος αποτελεί ένα ασφαλές μέσο, καθώς είναι εξοπλισμένο με αντισταθμιστή υπερφόρτωσης κι έτερα σύστημα ασφαλείας, αποτρέποντας έτσι το ενδεχόμενο ατυχήματος.

Ρύθμιση της παροχής

Ο συμπιεστής αέρος παράγει περισσότερη ροή από εκείνη που μπορεί να διαχειριστεί το σύστημα. Γι' αυτό το λόγο, υπάρχουν κατάλληλοι ρυθμιστές, οι οποίοι ρυθμίζουν χειροκίνητα την παροχή για τη σωστή απόκριση του συστήματος. Εν προκειμένω, υπάρχουν δυο ρυθμιστές στο σύστημα. Ο πρώτος πλαισιώνεται στην είσοδο του αισθητήρα, ενώ ο δεύτερος τοποθετείται σε διακλάδωση κατά την είσοδο του συστήματος ελέγχου ροής και πριν από την ηλεκτρική σφαιρική βαλβίδα. Ακολουθώντας, ο πρώτος ρυθμίζει σε πρώτο στάδιο το επίπεδο ροής, ενώ ο δεύτερος λειτουργεί ως εκτόνωση του "περιττού" φορτιού για το σύστημα. Ο συνδυασμός των δυο επιφέρει την ιδανική ροή για την ομαλή λειτουργία του συστήματος. Η ρύθμιση γίνεται με πλήρως ανοιχτή τη σφαιρική βαλβίδα, προκειμένου να διασφαλιστεί ότι η μέγιστη ροή που θα περάσει από το σύστημα θα είναι τα 25 slpm. Για την σωστή διεξαγωγή των μετρήσεων θα πρέπει να υπάρχει σταθερή ροή στο σύστημα. Συνεπώς, η ρύθμιση της παροχής γίνεται, αφού παρέλθει ορισμένο χρονικό διάστημα μέχρις ότου το αεροφυλάκιο του συμπιεστή να είναι σχεδόν άδειο και η ροή να έχει σταθεροποιηθεί. Έτσι, αποφεύγονται οι αλλαγές πίεσης από το γέμισμα/άδειασμα του αεροφυλακίου.

3.3.2 Παράγοντες που προέκυψαν στη διαδικασία των μετρήσεων

Κατά τις αρχικές μετρήσεις, διαπιστώθηκαν ορισμένα σημαντικά στοιχεία λειτουργίας που χαρακτηρίζουν την ηλεκτρική σφαιρική βαλβίδα. Πιο αναλυτικά, διαπιστώθηκε ότι η βαλβίδα έχει μια καθυστέρηση από τον αρχικό οριοδιακόπτη μέχρις ότου ξεκινήσει το πρώτο δείγμα ροής. Παράλληλα, η βαλβίδα, προτού έρθει σε επαφή με τον τελικό οριοδιακόπτη, έχει ήδη ανέλθει στην μέγιστη δυνατή ροή της. Είναι σημαντικό να αναφερθεί ότι οι πλευρές της βαλβίδας έδειξαν διαφορετική συμπεριφορά σε ίδιες μετρήσεις που έγιναν, γεγονός που αποδεικνύει ότι οι πλευρές είναι ανομοιόμορφες.

Εύρεση των ορίων της βαλβίδας (start/end lag)

Έπειτα από μετρήσεις που έγιναν, παρατηρήθηκε ότι πάρα το γεγονός ότι είχε δοθεί σήμα να ανοίξει η σφαιρική βαλβίδα, σημειώθηκε καθυστέρηση μέχρι να δοθεί ένδειξη ροής (start lag). Ο χρόνος αυτός θεωρείται νεκρός χρόνος για το σύστημα και οφείλεται στο κενό που υπάρχει από τον οριοδιακόπτη μέχρι το σημείο που ανοίγει η σφαιρική βαλβίδα για πρώτη φορά. Η απαλοιφή του νεκρού χρόνου θα επιτρέψει στο σύστημα να βρίσκεται στη σωστή θέση (λίγο πριν ανοίξει), όταν αυτό ζητηθεί, με αποτέλεσμα να έχει καλύτερη απόκριση κατά την έναρξη. Υπολογίζεται ότι θα πρέπει να δοθεί σήμα για χρονική διάρκεια 2200 ms στη σφαιρική βαλβίδα, προκειμένου αυτή να περιέλθει σε θέση ετοιμότητας, δηλαδή κατά το χρονικό σημείο λίγο πριν ξεκινήσει η ροή, ώστε να είναι έτοιμη να δώσει την απαιτούμενη ροή.

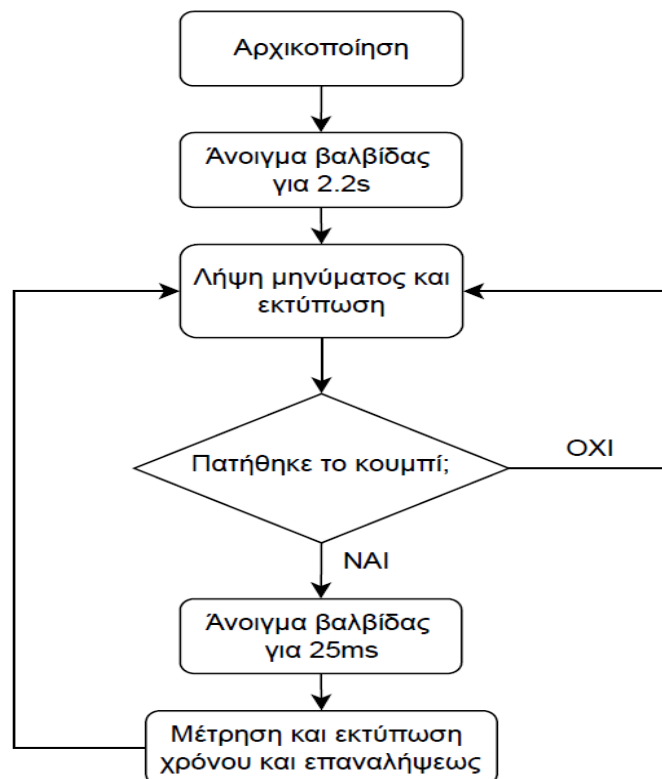
Παρατηρείται επίσης ότι, προτού η σφαιρική βαλβίδα φτάσει τον τελικό οριοδιακόπτη, έχει ήδη ανέλθει στο μέγιστο της ροής που μπορεί να αποδώσει (25 slpm από την παροχή) (end lag). Αυτό οφείλεται, λόγω της διατομής των σωληνώσεων της παροχής, που είναι αρκετά μικρότερη από αυτή της σφαιρικής βαλβίδας, γεγονός που επαληθεύει τη συμπεριφορά του συστήματος. Οι χρονικές καθυστερήσεις επαληθεύονται από τα γραφήματα των βασικών μετρήσεων που έγιναν παρακάτω(3.3.3 Καθορισμός δυναμικής απόκρισης της βαλβίδας, 3.3.4 Καθορισμός ροής μέσω ελέγχου της βαλβίδας).

Ανομοιομορφες πλευρές

Τα αποτελέσματα των μετρήσεων έδειξαν ότι κάθε πλευρά της ηλεκτρικής βαλβίδας έχει διαφορετικούς χρόνους όσον αναφορά στις καθυστερήσεις (start/end lag) και στα αποτελέσματά των τιμών ροής. Αυτό κατά πάσα πιθανότητα οφείλεται σε κατασκευαστικούς παράγοντες. Η ανομοιομορφία των δυο πλευρών περιορίζει το σύστημα όσον αναφορά στη λειτουργία του και θέτει επιπλέον παραμέτρους για την υλοποίησή του. Για την παρούσα πειραματική διαδικασία, χρησιμοποιήθηκε η πρώτη (1^η) πλευρά της ηλεκτρικής σφαιρικής βαλβίδας, ενώ η δεύτερη απομονώθηκε προγραμματιστικά από τον μικροελεγκτή. Η αρχική καθυστέρηση αποτελεί βασικό κριτήριο κατά τον διαχωρισμό των δυο πλευρών (θα χρησιμοποιηθεί αργότερα στον προγραμματισμό του μικροελεγκτή).

3.3.3 Καθορισμός δυναμικής απόκρισης της βαλβίδας

Σε πρώτο στάδιο, οι μετρήσεις πραγματοποιήθηκαν, εκκινώντας από ένα σημείο αναφοράς (start: 2200 ms) και προσθέτοντας κάθε φορά μια χρονική καθυστέρηση. Η διαδικασία αυτή επαναλήφθηκε έως ότου η ροή που διασχίζει τη σφαιρική βαλβίδα ανήλθε τα 25 slpm. Με αυτό το τρόπο, παρατηρήθηκε σε κάθε μέτρηση η μεταβολή της ροής. Ο απώτερος σκοπός των μετρήσεων ήταν η εύρεση του ελάχιστου δυνατού χρόνου μιας μεταβολής. Συγκεκριμένα, έγιναν δοκιμές για τις χρονικές τιμές: 50, 35, 25 και 10 ms, ξεκινώντας από τα 50 ms και καταλήγοντας στα 10 ms, οπότε και διαπιστώθηκε ότι η ηλεκτρική σφαιρική βαλβίδα δεν αποκρινόταν. Το πρόγραμμα που δημιουργήθηκε για το πρώτο σετ μετρήσεων επισυνάπτεται στο παράρτημα Β. Παρακάτω φαίνεται το σχετικό μπλοκ διάγραμμα (Σχήμα 3.6).



Σχήμα 3.6. Μπλόκ διάγραμμα για το πρόγραμμα του καθορισμού δυναμικής απόκρισης της βαλβίδας

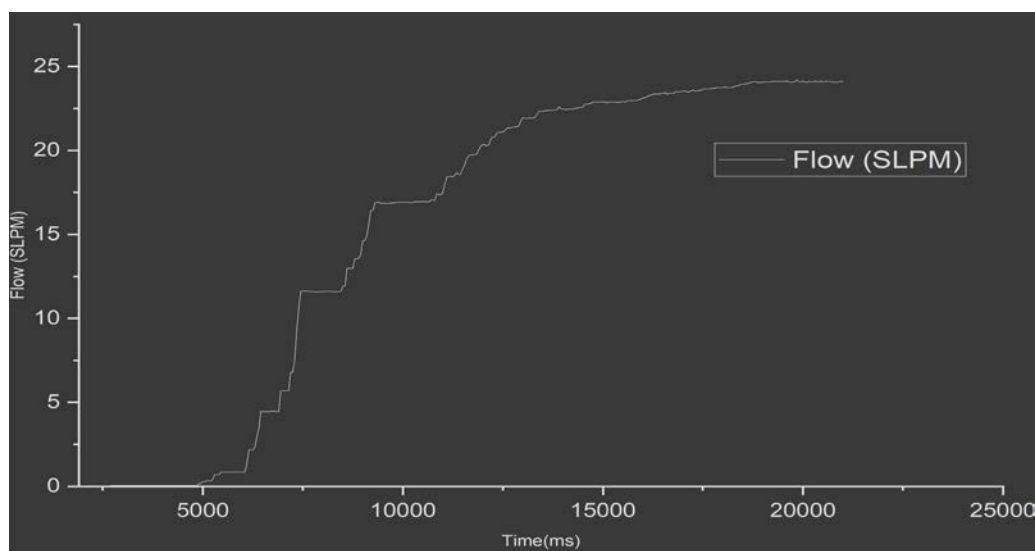
Διαδικασία υλοποίησης/Περιγραφή λειτουργίας του προγράμματος

Πριν εκτελεστούν οι μετρήσεις, η σφαιρική βαλβίδα είναι σε κλειστή θέση κι εφάπτεται του οριοδιακόπτη. Για το εν προκειμένω σετ μετρήσεων, το πρόγραμμα ξεκινάει, έχοντας τοποθετήσει την σφαιρική βαλβίδα σε θέση ετοιμότητας. Στην πράξη, απαλείφεται το κομμάτι, στο οποίο δεν υπάρχει ροή, εφαρμόζοντας σήμα ενεργοποίησης στη βαλβίδα για διάρκεια 2200ms. Στη συνέχεια, εκκινεί ο ατέρμονος βρόχος και, ταυτόχρονα, ενεργοποιείται ο βρόχος do-while. Λαμβάνεται η πρώτη μέτρηση για χρόνο 2200ms (αρχική θέση).

Οι πρώτες μετρήσεις έγιναν για σήμα χρονικής διάρκειας 50ms. Με το πάτημα του μπουττόν, η μεταβλητή "state" προσλαμβάνει την τιμή "1" και το πρόγραμμα εξέρχεται από το βρόχο do-while. Παράλληλα, εφαρμόζεται το σήμα ενεργοποίησης της βαλβίδας για διάρκεια 50ms, ενώ ο counter και ο timer αυξάνουν τις τιμές τους, οι οποίες εκτυπώνονται στο σειριακό τερματικό του υπολογιστή, ώστε ο αριθμός της μέτρησης (Πόσες φορές πατήθηκε το μπουττόν) να γίνει γνωστός όπως και ο συνολικός χρόνος που έχει παρέλθει (Το άθροισμα της χρονικής διάρκειας κάθε σήματος που έχει εφαρμοστεί). Το πρόγραμμα επιστρέφει στο βρόχο do-while και εκτυπώνεται η νέα μέτρηση για τη μετατόπιση της βαλβίδας που μόλις πραγματοποιήθηκε. Το πρόγραμμα παραμένει σε αυτή τη κατάσταση μέχρι να δοθεί το επόμενο ερέθισμα από το μπουττόν και να ληφθεί η επόμενη μέτρηση. Είναι σημαντικό να αναφερθεί ότι, η μετακίνηση της βαλβίδας είναι για διάρκεια 50ms (εφαρμόζετε σήμα ενεργοποίησης στη βαλβίδα για διάρκεια 50ms με κάθε πάτημα του μπουττόν) και σε κάθε στάδιο που έχει αλλάξει η θέση της βαλβίδας, μένει σταθερή η ροή για ένα ορισμένο χρονικό διάστημα περίπου 10-15 sec μέχρι να ληφθεί η νέα μέτρηση. Η διαδικασία επαναλαμβάνεται μέχρι η τιμή της ροής να ανέλθει τα 25slpm. Το ίδιο μοτίβο μετρήσεων ακολουθείται για τις τιμές χρόνου: 50, 35, 25 και 10 ms.

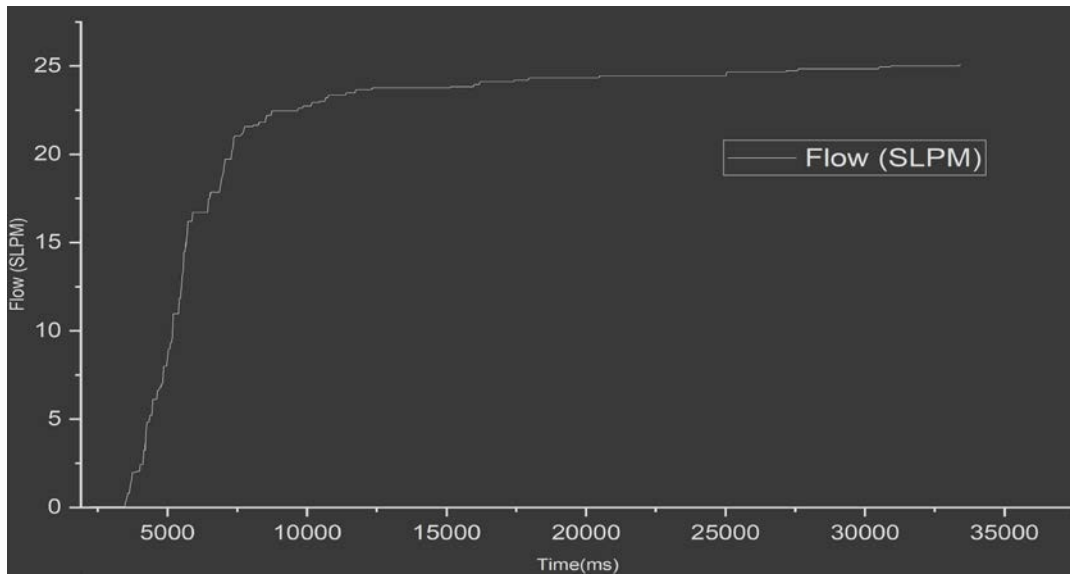
Παρακάτω απεικονίζονται τα γραφήματα για τέσσερις (4) διαφορετικές τιμές χρόνου. Εφαρμόζεται σήμα ενεργοποίησης ανοίγματος της βαλβίδας για χρόνο διάρκειας: 50, 35, 25, 10 ms και καταγράφεται η μεταβολή της ροής για κάθε μια φορά που εφαρμόστηκε το αντίστοιχο σήμα.

1^η Μέτρηση σήματος ενεργοποίησης διάρκειας 50ms



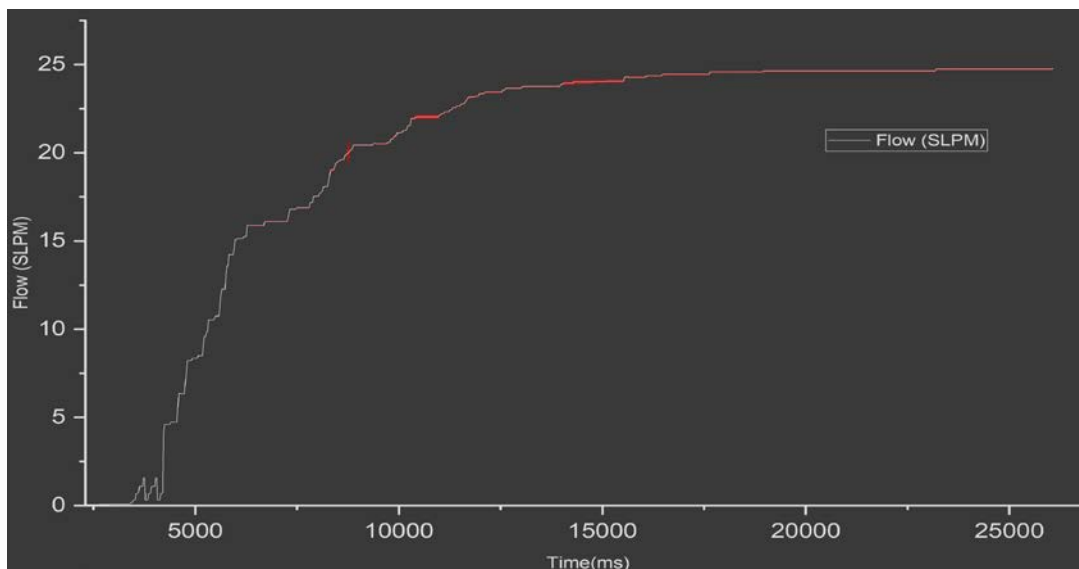
Σχήμα 3.7. Γραφική απεικόνιση ροής για μεταβολές βαλβίδας με βήμα 50ms σε συνάρτηση με το χρόνο (ανά χρονικά διαστήματα 10-15sec)

2^η Μέτρηση σήματος ενεργοποίησης διάρκειας 35ms



Σχήμα 3.8. Γραφική απεικόνιση ροής για μεταβολές βαλβίδας με βήμα 35ms σε συνάρτηση με το χρόνο (ανά χρονικά διαστήματα 10-15sec)

3^η Μέτρηση σήματος ενεργοποίησης διάρκειας 25ms



Σχήμα 3.9. Γραφική απεικόνιση ροής για μεταβολές βαλβίδας με βήμα 25ms σε συνάρτηση με το χρόνο (ανά χρονικά διαστήματα 10-15sec)

4^η Μέτρηση σήματος ενεργοποίησης διάρκειας 10ms

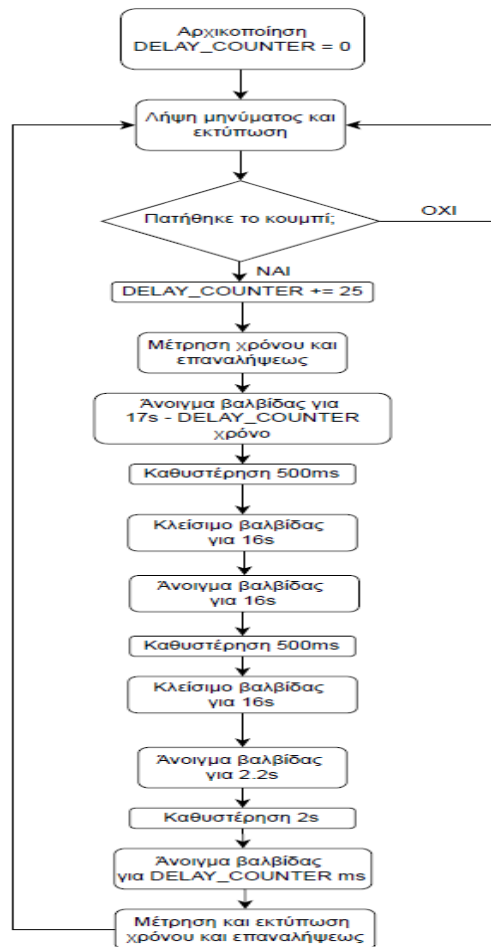
Μετά από επανειλημμένες προσπάθειες μεταβολής της ηλεκτρικής σφαιρικής βαλβίδας με χρονική καθυστέρηση 10ms, παρατηρήθηκε ότι ο χρόνος αυτός δεν επαρκεί για να μεταβάλει τη σφαιρική βαλβίδα (η σφαιρική βαλβίδα μένει ακίνητη).

Παρατηρήσεις

Όπως παρατηρείται από τα ανωτέρω διαγράμματα, η χρονική καθυστέρηση των 25ms αποτελεί τη χρονική διάρκεια που μπορεί να εφαρμοστεί σήμα στη βαλβίδα για την ελάχιστη δυνατή μεταβολή της ροής. Αυτή η διαδικασία μετρήσεων θα χρησιμοποιηθεί, προκειμένου το σύστημα να προσεγγίσει όσο το δυνατόν πιο κοντά την τιμή ροής που θα ζητηθεί από τον χρήστη. Οι διαφορές ανάμεσα στις μεταβολές των γραφημάτων είναι ελάχιστες. Παρατηρείται, ότι υπάρχουν σημεία στα διαγράμματα όπου φαίνεται να παραμένει σταθερή η ροή της βαλβίδας για ορισμένο χρονικό διάστημα ενώ αυξάνεται ο συνολικός αριθμός των σημάτων ενεργοποίησης που έχουν εφαρμοστεί. Πιο αναλυτικά, λόγω των πολύ μικρών χρονικών καθυστερήσεων που εφαρμοστήκαν, η σφαιρική βαλβίδα πολλές φορές δεν εμφάνιζε κάποια μεταβολή και χρειάστηκε να εφαρμοστεί η ίδια τιμή σήματος πολλές φορές, ώστε να υπάρξει κίνηση στη βαλβίδα. Αυτό οφείλεται σε κατασκευαστικούς παράγοντες που αφορούν στον ενεργοποιητή, καθώς και μηχανικά σφάλματα στο σύστημα μετάδοσής του. Στο γράφημα της τιμής των 25ms εμπεριέχονται με κόκκινα περιθώρια οι τιμές της τυπικής απόκλισης (Standard deviation). Η χαμηλή τυπική απόκλιση επαληθεύει την στενή συγκέντρωση των δεδομένων γύρω από τη μέση τιμή, υποδηλώνοντας ότι οι μετρήσεις ήταν αξιόπιστες με μικρές αποκλίσεις.

3.3.4 Καθορισμός ροής μέσω ελέγχου της βαλβίδας

Αφού βρέθηκε ο ελάχιστος χρόνος για μια μεταβολή (25ms), χρησιμοποιήθηκε πάνω σε ένα δεύτερο εναλλακτικό τρόπο μετρήσεων. Πιο αναλυτικά, έγινε χρήση μιας διαφορετικής μεθόδου, κατά την οποία ξεκινώντας από το ίδιο σημείο αναφοράς (Το σημείο αναφοράς ή αρχικό σημείο όπως αναφέρεται και προηγούμενος, υπολογίζεται όταν η βαλβίδα είναι επαπτόμενη του αρχικού οριοδιακόπτη και εφαρμόζεται σε αυτή σήμα ενεργοποίησης για διάρκεια 2200ms) η πρώτη μέτρηση λαμβάνεται εφαρμόζοντας σήμα ενεργοποίησης για διάρκεια 25ms. Όμως, αυτή τη φορά, η βαλβίδα δεν μετακινείται κατά 25ms από την προηγούμενη θέση, αλλά μετά από κάθε μέτρηση, εκτελεί μια πλήρη περιστροφή και επανέρχεται στο αρχικό σημείο αναφοράς (Η βαλβίδα κάνει ένα πλήρη κύκλο και επανέρχεται στο σημείο από όπου ξεκίνησε.). Στην αμέσως επόμενη μέτρηση εφαρμόζεται το ίδιο σήμα ενεργοποίησης, προσθέτοντας κάθε φορά το πλήθος των προηγούμενων σημάτων (Το σήμα ενεργοποίησης που εφαρμόζεται στη βαλβίδα για κάθε μέτρηση παίρνει τις ακόλουθες τιμές: 25, 50, 75, 100, 125... n+25, όπου n το άθροισμα όλων των προηγούμενων σημάτων). Αυτό συνεχίζεται μέχρι να φτάσει η ροή τα 25 slpm. Το πρόγραμμα που δημιουργήθηκε για το δεύτερο τρόπο προσδιορισμού της ροής και του ελέγχου βαλβίδας επισυνάπτεται στο παράρτημα Γ. Παρακάτω φαίνεται το σχετικό μπλοκ διάγραμμα (Σχήμα 3.10).



Σχήμα 3.10. Μπλόκ διάγραμμα για το πρόγραμμα του καθορισμού ροής μέσω ελέγχου της βαλβίδας

Διαδικασία υλοποίησης/Περιγραφή λειτουργίας του προγράμματος

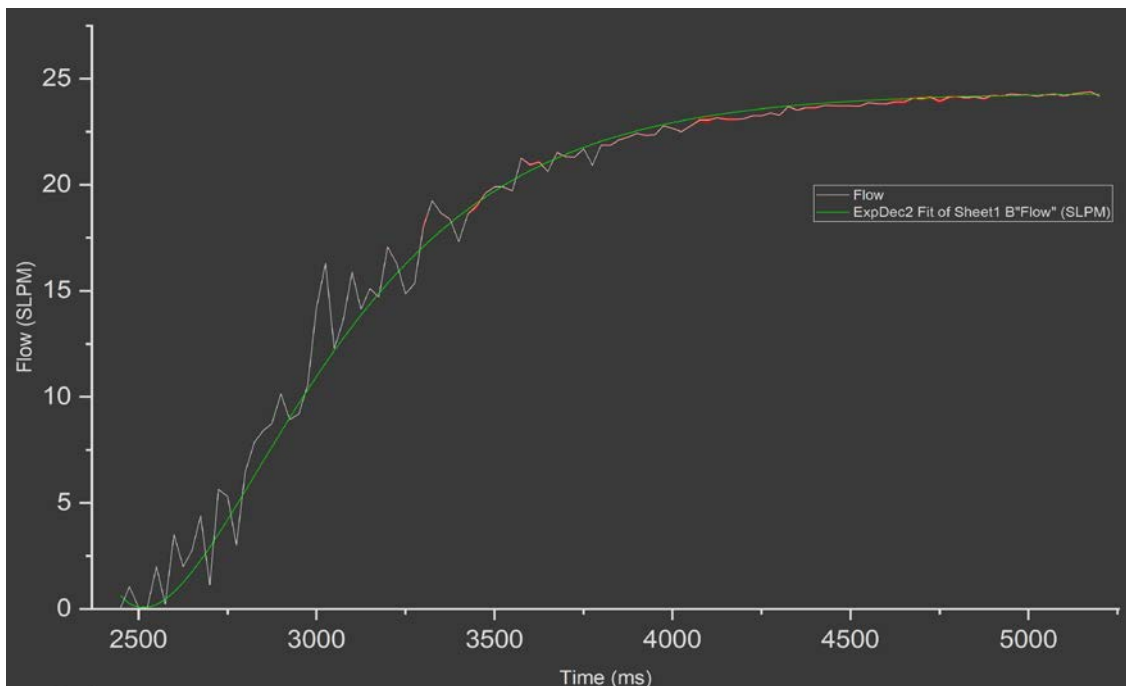
Το πρόγραμμα εκκινεί με τον ατέρμονο βρόχο και στην συνέχεια εισέρχεται στο βρόχο do-while, όπου γίνεται η λήψη της πρώτης μέτρησης. Η πρώτη μέτρηση δεν λήφθηκε υπόψη στο σετ των μετρήσεων. Με το πάτημα του μπουττόν, η μεταβλητή "state" προσλαμβάνει την τιμή "1" και το πρόγραμμα εξέρχεται από το βρόχο do-while.

Αφού εξέλθει του βρόχου, προστίθεται η τιμή του σήματος ενεργοποίησης των 25ms στον Delay_counter, ενώ την επόμενη στιγμή ο counter και ο timer αυξάνουν τις τιμές τους ανάλογα. Έπειτα, εκκινεί η ρουτίνα για την επανεκκίνηση της ηλεκτρικής σφαιρικής βαλβίδας. Ειδικότερα, η ρουτίνα επανεκκίνησης αποτελείται από τέσσερις φάσεις, σύμφωνα με τις οποίες η βαλβίδα ανοίγει και κλείνει δύο φορές. Οι δυο τελευταίες φάσεις λαμβάνουν χώρα, ώστε η βαλβίδα να παραλείψει την δεύτερη πλευρά και να επιστρέψει στην αρχική πλευρά μέχρι την επόμενη μέτρηση (Η βαλβίδα κάνει ένα πλήρη κύκλο και επανέρχεται στο σημείο αναφοράς δηλαδή στο σημείο από όπου ξεκίνησε).

Μετά την ολοκλήρωση της ρουτίνας η βαλβίδα βρίσκεται κλειστή απέχοντας 2200ms χρονική διαρκεί από τον αρχικό οριοδιακόπτη, στη πλευρά 1. Στη συνέχεια και αφού η βαλβίδα βρίσκεται στην αρχική της θέση εφαρμόζεται σε αυτή η τιμή του Delay-counter, ο οποίος εμπεριέχει την τιμή των n+25ms (25ms και το άθροισμα όλων των προηγούμενων

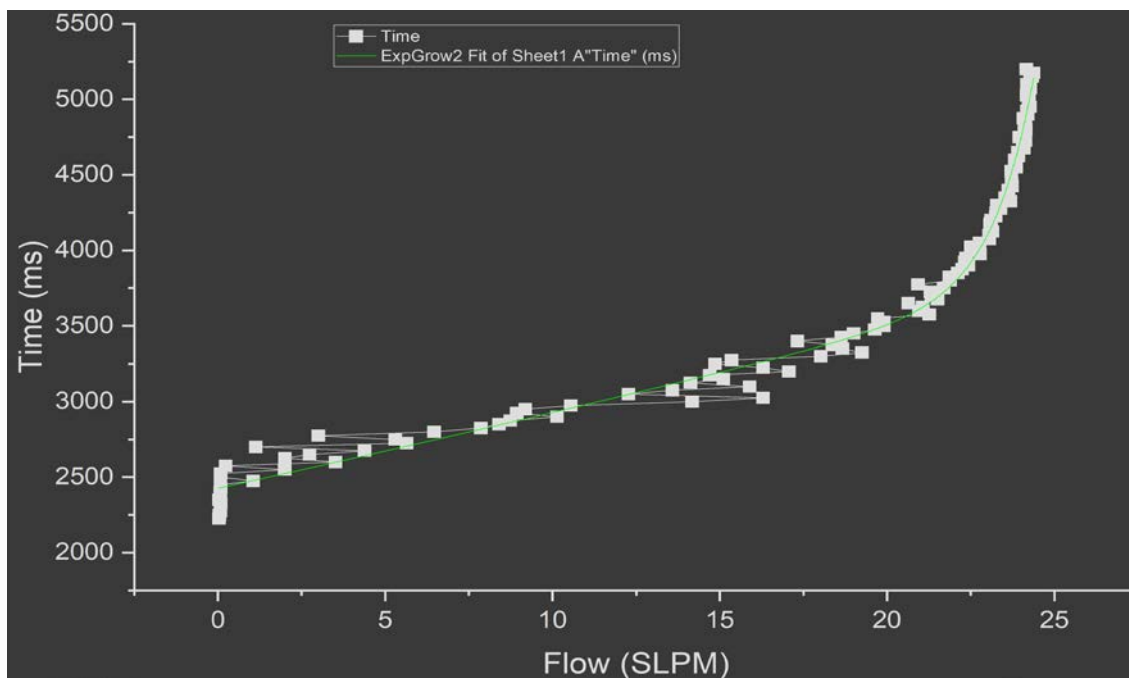
σημάτων). Το πρόγραμμα επιστρέφει στον βρόχο do-while και εκτυπώνεται η νέα μέτρηση για τη μεταβολή που μόλις πραγματοποιήθηκε.

Το πρόγραμμα παραμένει σε αυτή τη κατάσταση μέχρι να δοθεί το επόμενο ερέθισμα από το μπουττον και να ληφθεί η επόμενη μέτρηση (Χρειάστηκαν περίπου 10-15 sec μέχρι να καταγραφεί μια μέτρηση). Η διαδικασία επαναλαμβάνεται μέχρι η τελική τιμή της ροής να φτάσει τα 25slpm. Παρακάτω απεικονίζεται το γράφημα των μετρήσεων που έγιναν, ουσιαστικά φαίνεται πως μεταβάλλεται η ροή όσο αυξάνεται η διατομή της σφαιρική βαλβίδα (Σχήμα 3.8).



Σχήμα 3.11. Γραφική απεικόνιση ροής συνάρτηση του χρόνου, σύμφωνα με τη μεθοδολογία καθορισμού ροής μέσω ελέγχου της βαλβίδας

Σε πολλά σημεία στο διάγραμμα, παρατηρούνται ορισμένες μη φυσιολογικές διακυμάνσεις της ροής. Αναλυτικότερα παρατηρείται ότι το άνοιγμα της βαλβίδας μπορεί να είναι μικρότερο από μια προηγούμενη τιμή, με αποτέλεσμα ενώ έχει μεγαλώσει η χρονική διάρκεια που εφαρμόζεται το σήμα στη βαλβίδα ($n+25$), αυτή να ανοίγει λιγότερο από πριν και η τιμή ροής να είναι χαμηλότερη. Ουσιαστικά λόγω του ότι μετά από κάθε μέτρηση η βαλβίδα επιστρέφει στην αρχική της θέση, το γεγονός ότι θα εφαρμοστεί σε αυτή ένα σήμα ενεργοποίηση κατά 25ms μεγαλύτερης διάρκειας από το προηγούμενο, δεν εξασφαλίζει ότι η βαλβίδα θα επιστρέψει στην ίδια θέση που ήταν στη προηγούμενη μέτρηση. Όπως προαναφέρθηκε, αυτό οφείλεται σε μηχανικά σφάλματα στο σύστημα μετάδοσης λόγω της μικρής διαφοράς που έχουν τα σήματα ενεργοποίησης μεταξύ τους. Το φαινόμενο αυτό εντοπίζεται εντονότερα στις πρώτες μετρήσεις, ενώ φαίνεται να βελτιώνεται όσο αυξάνονται τα επίπεδα ροών που διαπερνούν το σύστημα. Στο παρακάτω σχήμα απεικονίζεται η γραφική απεικόνιση του χρόνου συνάρτηση της ροής για το ίδιο σετ μετρήσεων (Σχήμα 3.9):



Σχήμα 3.12. Γραφική απεικόνιση χρόνου συνάρτηση της ροής, σύμφωνα με την δεύτερη προσέγγιση προσδιορισμού της ροής

Αυτή η διαδικασία μετρήσεων αποτελεί μια μέθοδο υπολογισμού, βάσει της οποίας το σύστημα μπορεί να μεταβεί σε μια θέση ανάλογα με την τιμή ροής που θα του ζητηθεί. Μάλιστα, η μεθοδολογία αυτή θα εφαρμοστεί αργότερα στο κυρίως πρόγραμμα, ώστε να υπάρχει μια πρώτη εικόνα για το που περίπου θα πρέπει να μεταβεί η βαλβίδα προκειμένου να αποδώσει τη τιμή ροής που θα ζητηθεί από τον χρήστη.

Είναι σημαντικό να αναφερθεί ότι ο χρόνος που αναγράφεται στα διαγράμματα των σχημάτων 3.11 και 3.12 είναι το χρονικό διάστημα που χρειάζεται να εφαρμοστεί σήμα ενεργοποίησης στη βαλβίδα προκειμένου να μεταβεί από την αρχική της θέση στην αντίστοιχη τιμή ροής και δεν αποτελεί χρόνο προκειμένου να μεταβεί από μια τιμής ροής σε μια άλλη μεγαλύτερη τιμή.

Για την ανάλυση των δεδομένων χρησιμοποιήθηκε το πρόγραμμα "Origin", στο διάγραμμα του σχήματος 3.12 φαίνεται η μοντελοποιημένη γραφική αναπαράσταση που αποτυπώνεται με πράσινο χρώμα. Στον παρακάτω Πίνακα (Πίνακας 7) αναγράφεται η συνάρτηση μεταφοράς του συστήματος, καθώς και οι παράμετροι της. Για την μοντελοποίηση (Fitting) του συστήματος χρησιμοποιήθηκε το μοντέλο "ExpGrow2".

Το μοντέλο αυτό περιγράφει ένα μοτίβο ανάπτυξης με δύο εκθετικούς όρους, παρέχοντας λεπτομερείς πληροφορίες σχετικά με τα χαρακτηριστικά της ανάπτυξης, την αβεβαιότητα και την καλή προσαρμογή του μοντέλου στο σύστημα. Παράλληλα, μέσω του Reduced Chi-Square φαίνεται η εφαρμογή που έχει το μοντέλο πάνω στο σύστημα, ενώ το R-Square (COD) αποτελεί ένας συντελεστής προσδιορισμού, που υποδηλώνει την αναλογία της μεταβλητότητας στα δεδομένα που λαμβάνονται υπόψη από το μοντέλο. Η εφαρμογή που έγινε στο σύστημα ήταν η καλύτερη δυνατή, λαμβάνοντας υπόψιν τις ιδιαιτερότητες των τιμών.

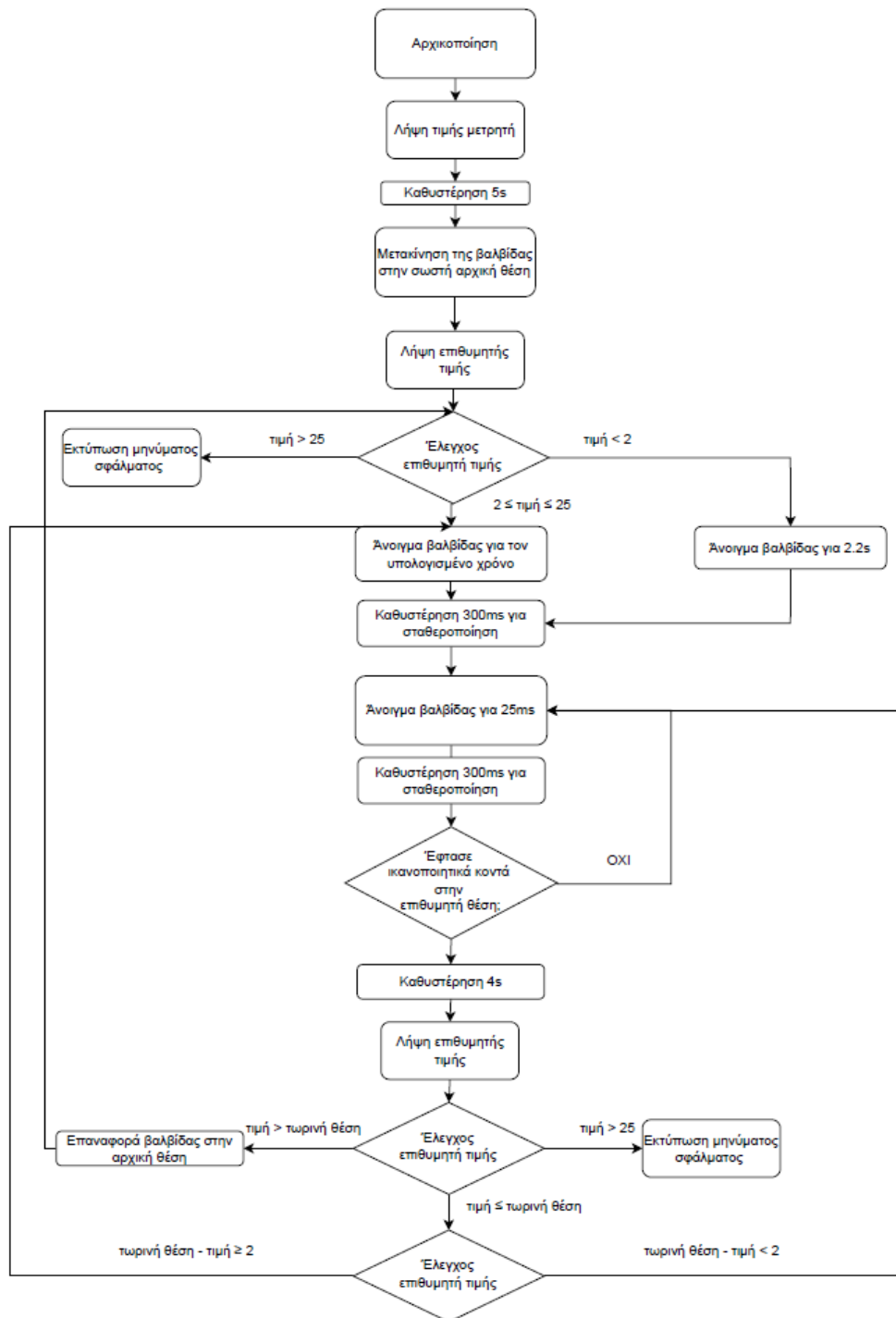
Model	ExpGrow2
Equation	$y = y_0 + A1 \cdot \exp((x-x_0)/t_1) + A2 \cdot \exp((x-x_0)/t_2)$
Plot	Time
y0	-4322.98261 ± 12963.224
x0	4.67662 ± 1368699.25425
A1	1.60641E-4 ± 178.55714
t1	1.23137 ± 0.10399
A2	6979.17895 ± 6.8527E7
t2	139.39601 ± 250.4393
Reduced Chi-Sqr	7407.42442
R-Square (COD)	0.99062
Adj. R-Square	0.99021

Πίνακας 7. Μαθηματική προσέγγιση μοντέλου ExpGrow2

3.4 Προγραμματισμός μικροελεγκτή

Μετά την ολοκλήρωση της κυκλωματικής διάταξης του ελεγκτή ροή μάζας, έγιναν οι απαραίτητες ενεργείες για την ομαλή λειτουργία των στοιχείων με τον μικροελεγκτή. Στη συνέχεια, πραγματοποιήθηκαν οι απαιτούμενες μετρήσεις για την μελέτη της απόκρισης της σφαιρική βαλβίδα σε ορισμένα χρονικά ερεθίσματα καθώς επίσης και η εύρεση του χρόνου για την ελάχιστη μεταβολή της ροής. Το τελικό βήμα για την ολοκλήρωση της υλοποίησης του συστήματος, αποτελεί η δημιουργία του κύριος προγράμματος για την οδήγηση του ελεγκτή ροής μάζας. Το πρόγραμμα προσαρμόστηκε στις ανάγκες του συστήματος, ενσωματώνοντας όλες τις παραμέτρους που έχουν μελετηθεί στα προηγούμενα κεφάλαια.

Η λογική πίσω από το κυρίως πρόγραμμα του ελεγκτή ροή μάζας είναι η εξής: Ο χρήστης πληκτρολογεί μια τιμή ροής στο σειριακό τερματικό του υπολογιστή που επικοινωνεί με τον μικροελεγκτή. Ο μικροελεγκτής κάνει τους απαραίτητους υπολογισμούς και καταλήγει στον χρόνο που απαιτείται να εφαρμοστεί σήμα στη σφαιρική βαλβίδα, προκειμένου να επιτευχθεί η επιθυμητή τιμή ροής. Παράλληλα καθόλη τη διάρκεια λειτουργίας του προγράμματος γίνεται έλεγχος των μετρήσεων του αισθητήρα προκειμένου να διατηρείται σταθερή η τιμή που έχει ορίσει ο χρήστης. αν και εφόσον προκύψουν αναταραχές στη ροή τότε το σύστημα τις απορρόφα κάνοντας τις απαιτούμενες ενέργειες. Το πρόγραμμα χωρίζεται σε τρία μέρη: 1) Την ρουτίνα που πραγματοποιεί η σφαιρική βαλβίδα προκειμένου να υλοποιήσει τους υπολογισμούς στη πλευρά που έχει επιλεχθεί (side1), 2) Πρώτο σκέλος του προγράμματος υπολογισμού, 3) Δεύτερο σκέλος πρόγραμμα υπολογισμού (εντός του ατέρμονου βρόχο "while(1)"). Το πρόγραμμα του ελεγκτή ροής μάζας επισυνάπτεται στο παράρτημα Δ. Στη συνέχεια αναλύεται η σύνθεση και η λειτουργία του προγράμματος. Στο σχήμα 3.13 φαίνεται το σχετικό μπλοκ διάγραμμα.



Σχήμα 3.13. Μπλόκ διάγραμμα του κυρίως προγράμματος

3.4.1 Ρουτίνα εύρεση της επιθυμητής πλευράς

Λαμβάνοντας υπόψιν τις αναφορές που έγιναν σε προηγούμενα κεφάλαια, κρίνεται απαραίτητο να απαλειφθεί η μια πλευρά της σφαιρικής βαλβίδας (2^η πλευρά). Όπως είναι λογικό, αυτό δεν μπορεί να συμβεί επεμβαίνοντας στο σύστημα της βαλβίδα. Επομένως, ο τρόπος για να πραγματοποιηθεί αυτή η ενεργεία είναι μέσω του κύριου προγράμματος. Για το λόγο αυτό, κατασκευάζεται μια συνάρτηση που εκτελεί μια ρουτίνα εύρεσης της επιθυμητής πλευράς. Αργότερα στο πρόγραμμα θα χρησιμοποιηθούν αντίστοιχες εντολές για την παράκαμψη κάθε φορά της μη επιθυμητής πλευράς.

Η εν προκειμένω συνάρτηση ονομάζεται "go_to_side1()". Είναι γνωστό από τις μετρήσεις που έγιναν ότι για χρονική διάρκεια παλμού 2370ms η πλευρά ένα (1) δίνει πάντα ροή μικρότερη από 00.12 slrm, γεγονός το οποίο μπορεί να χρησιμοποιηθεί σε μια συνθήκη "if()" για να βρεθεί η πλευρά στην οποία βρίσκεται εκείνη τη δεδομένη στιγμή η σφαιρική βαλβίδα. Αν διαπιστωθεί ότι η βαλβίδα βρίσκεται στη δεύτερη πλευρά τότε αρκεί να πραγματοποιήσει μια περιστροφή 180° και να βρεθεί στην πρώτη. Στη περίπτωση που βρίσκεται στη πρώτη πλευρά πρέπει να πραγματοποιήσει περιστροφή 360° προκειμένου να επανέρθει στην ίδια πλευρά (Η κίνηση αυτή πραγματοποιείται διότι έχει μετακινηθεί η βαλβίδα μετά το σήμα των 2370ms, δηλαδή δεν εφάπτεται του αρχικού οριοδιακόπτη).

Η περιστροφή 180° περιλαμβάνει την ενεργοποίηση δυο καταστάσεων: 1) η βαλβίδα ανοίγει, 2) η βαλβίδα κλείνει. Η περιστροφή 360° ουσιαστικά επαναλαμβάνει τις καταστάσεις αυτές από δυο φορές. Οι καταστάσεις μένουν ενεργοποιημένες για διάρκεια 16 sec η κάθε μια (χρόνος που χρειάζεται για να μεταβεί η βαλβίδα από τον αρχικό οριοδιακόπτη στο τελικό κι αντιστρόφως). Προτού εκτελεστεί η συνθήκη if() για την εύρεση της εκάστοτε πλευράς, εκτελείται μια περιστροφή δύο καταστάσεων προκειμένου να εξασφαλιστεί ότι το σήμα χρονικής διάρκειας 2370ms, θα εκτελεστεί σε μια πλευρά, κατά την οποία η βαλβίδα εφάπτεται του αρχικού οριοδιακόπτη. Αυτό γίνεται για να αποφευχθεί η περίπτωση στην οποία η βαλβίδα είχε μείνει στα μισά της διαδρομής της, από παλαιότερη χρήση (δηλαδή δεν εφάπτεται του αρχικού οριοδιακόπτη).

Αξίζει να σημειωθεί ότι, η συνάρτηση που είχε δημιουργηθεί σε προηγούμενο χρόνο, για τη μετατροπή της τιμής του αισθητήρα σε float τιμή "convert_to_float()" (3.2.2 Δημιουργία περιβάλλοντος επικοινωνίας με τον αισθητήρα-Μετατροπή της αναγνώσιμης τιμής σε επεξεργάσιμη), τοποθετείται εντός της συνάρτησης "callback". Με αποτέλεσμα κάθε τιμή του αισθητήρα να μετατρέπεται σε float τιμή, κάθε φορά που καλείται η συνάρτηση για μια νέα μέτρηση. Η συνάρτηση ρυθμίζεται λαμβάνοντας την επιλογή "4", για τη μετατροπή της τιμής της ροής Μάζας. Το αποτέλεσμα της συνάρτησης αποθηκεύεται στη μεταβλητή "meter_value". Η συνάρτηση "setpoint_convert_to_float()" ακολουθεί την ίδια λογική με τη συνάρτηση "convert_to_float()". Στη προκείμενη περίπτωση μετατρέπει τη τιμή που εισάγει ο χρήστης στο σειριακό τερματικό "set_point[6]" σε float τιμή, δηλαδή "set_point_f". Στη συνέχεια, με τη συνάρτηση υπολογισμού του χρόνου (Πίνακας 6), υπολογίζεται ο αντίστοιχος χρόνος για τη ροή, στην οποία βρίσκεται η σφαιρική βαλβίδα την εκάστοτε στιγμή (position_time). Αφαιρώντας αυτό το χρόνο από το χρόνο των 16sec, υπολογίζεται ο χρόνος που χρειάζεται μέχρι να ανοίξει πλήρως η βαλβίδα σε περίπτωση που απέχει του αρχικού οριοδιακόπτη. Η συνάρτηση "go_to_side1()" τοποθετείται υπό της συνάρτησης του receive των δεδομένων του αισθητήρα.

3.4.2 Πρώτο σκέλος του προγράμματος υπολογισμού

Αφού ολοκληρωθεί η ρουτίνα της συνάρτησης "go_to_side1()", η βαλβίδα βρίσκεται σε κλειστή θέση, εφάπτομένη στον οριοδιακόπτη. Το πρόγραμμα προωθεί στο σειριακό τερματικό (huart 2) στο οποίο έχει αλληλεπίδραση ο χρήστης, το μήνυμα που περιέχει ο tx_buffer[30] ("\r\nGive a setpoint e.g[000.00]:"). Ζητείται δηλαδή από τον χρήστη να εισάγει μια τιμή ροής πέντε ψηφίων της μορφής "000.00".

Μετά την καταχώρηση της τιμής του χρήστη στο πρόγραμμα μέσω της "HAL_UART_Receive", εκτελούνται τρεις βρόχοι ελέγχου "if()", οπου εξετάζουν τρία πιθανά σενάρια: 1) Ελέγχεται αν η τιμή είναι μεγαλύτερη από 25, 2) μεγαλύτερη από 2 και μικρότερη από 25, 3) μικρότερη από 2. Πιο αναλυτικά, με τη πρώτη συνθήκη ελέγχου προλαμβάνεται το ενδεχόμενο λάθος του χρήστη, να πληκτρολογήσει τιμή μεγαλύτερη των 25 slpm. Στη περίπτωση αυτή ένα μήνυμα εμφανίζεται στο συριακό τερματικό ενημερώνοντας το χρήστη για το πεδίο τιμών που μπορεί να λάβει ο ελεγκτής "*****error value, choice must be 0<=choice<=25.". Στη συνέχεια, αν η τιμή προκύψει μεγαλύτερη ή ίση από 2 slpm τότε μέσω της συνάρτησης υπολογισμού χρόνου (Πίνακας 6), υπολογίζεται ο χρόνος που χρειάζεται να δοθεί σήμα στη σφαιρική βαλβίδα "time" για να αποδώσει τη ροή που ζήτησε ο χρήστης.

Σε αυτό το χρόνο συμπεριλαμβάνεται η αρχική καθυστέρηση από τον οριοδιακόπη μέχρι να ξεκινήσει η ροή (2200ms). Αξίζει να σημειωθεί ότι ο χρόνος που θα υπολογιστεί θα είναι για 2 slpm λιγότερα από αυτά που ζήτησε ο χρήστης. Προκείμενου τα slpm που υπολείπονται να αποδοθούν σταδιακά, παραθέτοντας στη συνέχεια το χρόνο για την ελαχίστη μεταβολή της ροής (25ms). Η ενέργεια αυτή βρίσκεται σε ένα βρόχο επανάληψης do-while και επαναλαμβάνεται κάθε 300ms. Ο βρόχος συνεχίζει να επαναλαμβάνεται μέχρις ότου η τιμή που αποδίδει το σύστημα να είναι μικρότερη η ίση κατά 0.30 slpm "SYSTEM_DEVIATION" από τη τιμή που είχε ορίσει ο χρήστης (Η απόκλιση που μπορεί να λάβει η τιμή ροής που αποδίδει το σύστημα υπολογιστική υστέρηση από δοκιμές που έγιναν). Στη περίπτωση που η τιμή του χρήστη είναι μικρότερη από τα 2 slpm τότε εκτελείται μόνο ο βρόχος do-while, χωρίς να χρησιμοποιηθεί ο χρόνος που υπολογιστική προηγούμενος. Αυτές οι εντολές ελέγχου και επανάληψης εκτελούνται στο πρώτο μέρος του προγράμματος όπου λόγω της εκτέλεσης της αρχικής ρουτίνας είναι γνωστό στο σύστημα ότι η σφαιρική βαλβίδα εκκινείται από κλειστή θέση και είναι εφαπτομένη του οριοδιακόπη.

3.4.3 Δεύτερο σκέλος πρόγραμμα υπολογισμού (εντός του ατέρμονου βρόχο "while(1)")

Εισερχόμενο το πρόγραμμα στον ατέρμονο βρόχο "while(1)" θα συνεχίσει να εκτελείται για το υπόλοιπο της λειτουργίας του. Στη συνέχεια, εκτελείται η ίδια διαδικασία εκχώρησης τιμής που πραγματοποιήθηκε και στην αρχή από το πρόγραμμα, αιτούμενο να καταχωρήσει ο χρήστης ένα νέο set point. Στη προκείμενη περίπτωση, εκτελείται, ένας βρόχος ελέγχου "if()" και ένας βρόχος ελέγχου "if() - else" οπου εξετάζουν τρία πιθανά σενάρια: 1) η τιμή είναι μεγαλύτερη από 25, 2) η τιμή που δίνει ο χρήστης είναι μικρότερη της τιμής στην οποία βρίσκεται η σφαιρική βαλβίδα, 3) η τιμή που δίνει ο χρήστης είναι μεγαλύτερη της τιμής στην οποία βρίσκεται η σφαιρική βαλβίδα. Πιο αναλυτικά, στη πρώτη συνθήκη επαναλαμβάνεται ο ίδιο έλεγχος που πραγματοποιήθηκε στο πρώτο μέρος του προγράμματος, προκείμενου να μην υπερβεί ο χρήστης την τιμή των 25 slpm. Στη συνέχεια, αφού έχει καταχωρηθεί τιμή από το χρήστη και λαμβάνοντας υπόψιν ότι η τιμή που καταχωρήθηκε είναι μικρότερη της τιμής στην οποία βρίσκεται η σφαιρική βαλβίδα. Τότε από τη συνάρτηση υπολογισμού χρόνου (Πίνακας 7), υπολογίζεται ο χρόνος που χρειάζεται να δοθεί σήμα στη σφαιρική βαλβίδα "time" για να αποδώσει τη ροή που ζήτησε ο χρήστης.

Υπολογίζεται επίσης ο χρόνος που χρειάστηκε για να φτάσει στο σημείο στο οποίο βρίσκεται η σφαιρική βαλβίδα "position_time". Αφαιρώντας τους δυο χρόνους, το σύστημα γνωρίζει το χρόνο που χρειάζεται να εφαρμοστεί σήμα στη σφαιρική βαλβίδα, προκειμένου να μεταβεί από τη τιμή που βρίσκεται κοντά στη τιμή που έθεσε ο χρήστης. Μια συνθήκη "if()-else", που τοποθετείται μέσα στη πρώτη, ελέγχει αν η αφαίρεση των δύο χρόνων είναι μεγαλύτερη των 2 slpm. Στη περίπτωση που ισχύει αυτό, τότε εκτελείται το ίδιο σετ εντολών για να φτάσει το σύστημα τη τιμή του χρήστη, κατά 2 slpm λιγότερα και στη συνέχεια το σετ των εντολών do-while, εφαρμόζοντας παλμούς των 25ms προκειμένου το σύστημα να πλησιάσει την τιμή του χρήστη σταδιακά όσο είναι μικρότερη η ίση κατά 0.30 slpm. Στη περίπτωση που η αφαίρεση των δύο χρόνων είναι μικρότερη από τα 2 slpm (else) τότε εκτελείται μόνο ο βρόχος do-while, χωρίς να χρησιμοποιηθεί ο χρόνος που υπολογιστικέ προηγούμενος από την αφαίρεση τους.

Τέλος, στη περίπτωση που η τιμή που δίνει ο χρήστης είναι μεγαλύτερη της τιμής στην οποία βρίσκεται η σφαιρική βαλβίδα, τότε ικανοποιείται η συνθήκη "else". Ειδικότερα, για να μπορέσει η σφαιρική βαλβίδα να επιτύχει μικρότερη τιμή από αυτή που εισήγαγε ο χρήστης, θα πρέπει να υλοποιήσει μια περιστροφή 360° έτσι ώστε να επιστρέψει στην αρχική της πλευρά και να ξεκινήσει η κατάσταση ανοίγματος από την αρχή. Όταν η σφαιρική βαλβίδα έρθει σε αυτή τη κατάσταση, εκτελούνται οι ίδιες εντολές που χρησιμοποιήθηκαν στο πρώτο μέρος του προγράμματος για την εύρεση της επιθυμητής ροής. Η διαδικασία ολοκληρώνεται όταν το πρόγραμμα εξέλθει του αρχικού βρόχου do-while. Δηλαδή όταν η τιμή του συστήματος βρίσκεται στο πεδίο τιμών : ± 0.30 slpm. Με την έξοδο του προγράμματος από το βρόχο do-while, το πρόγραμμα επιστρέφει στο αρχικό πεδίο όπου ζητάει από το χρήστη να εισάγει καινούργια τιμή. Η συνολική διαδικασία επαναλαμβάνετε για τη νέα τιμή που εισάγει ο χρήστης.

3.4.4 Αλληλεπίδραση αναμεσά στα δυο σκέλη

Είναι σημαντικό να αναφέρουμε ότι τα δυο τμητά κώδικα που δημιουργηθήκαν (3.4.2 Πρώτο σκέλος του προγράμματος υπολογισμού, 3.4.3 Δεύτερο σκέλος πρόγραμμα υπολογισμού) ουσιαστικά εκτελούν περίπου τις ίδιες διαδικασίες, με τη διαφορά ότι στο δεύτερο σκέλος προτίθεται και ο παράγοντας υπολογισμού της θέσης στην οποία βρίσκεται η σφαιρική βαλβίδα (position_time). Καθώς επίσης και η ρουτίνα επιστροφής της βαλβίδας στην θέση αφετηρίας, σε περιπτώσεις που δοθεί τιμή από το χρήστη μεγαλύτερη από αυτή την οποία αποδίδει εκείνη τη στιγμή το σύστημα. Φαίνεται λοιπόν ότι το πρώτο σκέλος είναι περιττό. Αφού το δεύτερο σκέλος μπορεί να υπερκαλύψει τις ανάγκες του συστήματος. Παρόλα αυτά το πρώτο σκέλος του προγράμματος υπολογισμού είναι απαραίτητο για την ομαλή λειτουργία του συστήματος.

Η συνάρτηση υπολογισμού του χρόνου "time", υπολογίζει το χρόνο που χρειάζεται η σφαιρική βαλβίδα, από την αρχική της θέση εφραπτομένη του αρχικού οριοδιακόπη μέχρι και 2 slpm λιγότερα από τη τιμή ροής που ζήτησε ο χρήστης. Στη συνέχεια αυτά τα 2 slpm που υπολείπονται, έρχονται εις πέρας με τη χρήση των δομών επανάληψης do-while, μετακινώντας κάθε φορά κατά 25ms τη βαλβίδα μέχρι να επιτύχει τη τιμή του χρήστη (με μια απόκλιση ± 0.30 slpm). Η συνάρτηση υπολογισμού της θέσης "position_time" υπολογίζει το χρόνο που χρειάζεται η σφαιρική βαλβίδα, από την αρχική της θέση εφραπτομένη του αρχικού οριοδιακόπη μέχρι να φτάσει στη θέση που βρίσκεται.

Μετά την ολοκλήρωση της ρουτίνας "go_to_side1()", η βαλβίδα βρίσκεται σε κλειστή θέση, εφαπτομένη του αρχικού οριοδιακόπτη. Σε αυτή τη θέση όπως έχει αναφερθεί και σε προηγούμενα κεφάλαια μεσολαβεί ένα διάστημα περίπου 2200ms μέχρι και λίγο πριν ξεκινήσουν τα πρώτα δείγματα ροής. Αυτή η χρονική υστέρηση, συμπεριλαμβάνεται στη συνάρτηση υπολογισμού χρόνου και θέσης. Δηλαδή, η συνάρτηση υπολογισμού θέσης θα υπολογίσει ότι χρειάζονται 2200ms, παρόλο που η σφαιρική βαλβίδα μπορεί να βρίσκεται σε κλειστή θέση, εφαπτομένη του αρχικού οριοδιακόπτη. Κάνοντας την αφαίρεση των δυο χρόνων "time-position_time" το αποτέλεσμα θα είναι ένας χρόνος μικρότερος από αυτόν που απαιτείται για να φτάσει η σφαιρική βαλβίδα στην επιθυμητή τιμή (2 slpm λιγότερα από τη τιμή ροής που ζήτησε ο χρήστης). Αυτό έχει σαν αποτέλεσμα η δομή επανάληψης do-while να χρειαστεί πολύ περισσότερο χρόνο μέχρι η βαλβίδα να επιτύχει τη τιμή του χρήστη.

Η δομή προγράμματος, αποτελούμενη από δυο επιμέρους σκέλη, αποσκοπεί, στο να μην εισέρχεται το πρόγραμμα στην if() συνάρτηση (if συνάρτηση στον ατέρμονο βρόχο που πραγματοποιείται η αφαίρεση time-position_time) όταν βρίσκεται στο κενό διάστημα μεταξύ οριοδιακόπτη και λίγο πριν ξεκινήσουν τα πρώτα δείγματα ροής. Δίνοντας ο χρήστης οποιαδήποτε τιμή εισόδου (μεγαλύτερη του μηδενός και μικρότερη του 25), η σφαιρική βαλβίδα εξέρχεται αυτού του κενού διαστήματος με τη χρήση του πρώτου σκέλους. Με την είσοδο του προγράμματος στον ατέρμονο βρόχο το σύστημα πάντα θα εκτελεί τον ανάλογο υπολογισμό για την τιμή στην οποία βρίσκεται, δηλαδή η αφαίρεση των δύο χρόνων θα πραγματοποιείται πάντα μετά από το κενό διάστημα.

4 ΚΕΦΑΛΑΙΟ 4^ο : ΑΠΟΤΕΛΕΣΜΑΤΑ, ΕΠΑΛΗΘΕΥΣΗ ΛΕΙΤΟΥΡΓΙΑΣ

Το πείραμα ολοκληρώνεται με τη πραγματοποίηση ενός σετ μετρήσεων, προκειμένου να φανεί η χρονική απόκριση και η ακρίβεια του συστήματος σε διάφορες καταστάσεις.

4.1 Μετρήσεις επαλήθευσης λειτουργίας

Καταχωρήθηκαν στο σύστημα του ελεγκτή ροής μάζας τριάντα (30) τυχαίες τιμές ροής (set points). Παράλληλα μετρήθηκε ο χρόνος που χρειάστηκε το σύστημα για να μεταβεί από τη μια τιμή στην άλλη. Ο υπολογισμός του χρόνου έγινε με τη χρήση σειριακού τερματικού που περιλαμβάνει χρονική απεικόνιση κάθε εντολής που εκτυπώνεται, σε πραγματικό χρόνο. Παρακάτω απεικονίζεται ο πίνακας των μετρήσεων (Πίνακας 8).

Set points (slpm)	MFC values (slpm)	Time (sec)	Accuracy (slpm)
12	12.44	14.187	0.44
5.5	5.4	73.641	0.1
18	18.21	1.390	0.21
9.8	10.08	71.511	0.28
23	22.99	29.389	0.01
4	4.92	77.795	0.92
14.2	14.33	2.984	0.13
7	7.27	92.511	0.27
20.6	20.45	7.499	0.15
2	2.34	79.421	0.34
16.4	16.45	2.883	0.05
8	9.58	78.937	1.58
3.7	3.8	80.402	0.1
22	22.66	1.698	0.66
11	11.34	77.570	0.34
19.1	19.23	11.524	0.13
1	1.04	82.070	0.04
10	9.84	4.715	0.16
25	24.96	101.475	0.04
6.03	6.61	73.084	0.58
13	13.53	4.241	0.53
17.08	17.1	29.521	0.02
21	21.16	17.768	0.16
10.07	10.19	74.049	0.12
24.5	24.4	47.239	0.1
15.2	15.8	72.287	0.6
9	10.19	77.609	1.19
5.1	5.44	79.703	0.34

20	19.9	3.260	0.1
3.1	3.53	76.657	0.43

Πίνακας 8. Πίνακας αποτελεσμάτων των δοκιμαστικών μετρήσεων, περιλαμβάνει επιθυμητή τιμή και χρόνο απόκρισης

4.1.1 Ακρίβεια

Ο πίνακας παρέχει δεδομένα σχετικά με τον ελεγκτή ροής μάζας, παρουσιάζοντας τιμές για τα επιθυμητά επίπεδα ροής (set points) σε τυπικά λίτρα ανά λεπτό, τις πραγματικές τιμές που καταγράφονται στον ελεγκτή ροής μάζας, εξίσου σε τυπικά λίτρα ανά λεπτό, καθώς και το χρόνο που χρειάζεται ο ελεγκτής για να μετατοπιστεί από τη μια τιμή στην άλλη σε δευτερόλεπτα. Εξετάζοντας τις συνολικές επιδόσεις του συστήματος, παρατηρείται ότι στις περισσότερες περιπτώσεις το σύστημα επιτυγχάνει ικανοποιητική ακρίβεια, με μέση τιμή ακρίβειας να είναι (Mean), $\bar{x} = 0.34 \text{ slpm}$ (διαφορά πραγματικής από μετρούμενη τιμή) και τυπική απόκλιση (Standard Deviation) $s = 0.37 \text{ slpm}$ (για όλες τις τιμές ακριβείας που προκύπτουν από τον πίνακα 8). Ωστόσο σε μεμονωμένες περιπτώσεις το σύστημα αποκλίνει της επιθυμητής τιμής. Η μεγαλύτερη απόκλιση που έχει καταγραφεί είναι στο παράδειγμα των 8 slpm όπου το σύστημα είχε απόκλιση κατά 1.58 slpm . Αυτό πιθανότατα οφείλεται, όπως είχε παρατηρηθεί και στη διαδικασία των αρχικών μετρήσεων της σφαιρικής βαλβίδας (3.3.3 Καθορισμός δυναμικής απόκρισης της βαλβίδας) σε μηχανικά σφάλματα στο σύστημα μετάδοσης (στασιμότητα και απότομη εκκίνηση γαντζώματος που μεταφέρει τη κίνηση του ενεργοποιείται στη σφαιρική βαλβίδα). Αυτό επαληθεύει το γεγονός ότι συμβαίνει σε μεμονωμένα περιστατικά, ενώ κοντινές τιμές όπως η τιμή των 7 slpm έχει αρκετά μικρότερη απόκλιση.

4.1.2 Χρονική απόκριση

Ο χρόνος απόκρισης του συστήματος αναφέρεται στο χρονικό διάστημα που απαιτείται, ούτως ώστε το σύστημα να φτάσει από μια αρχική κατάσταση σε μια νέα κατάσταση, μετά από μια αλλαγή στην είσοδό του. Ουσιαστικά, περιγράφει το πόσο γρήγορα ή αργά το σύστημα ανταποκρίνεται σε μια μεταβολή. Υπολογίζεται η μέση τιμή της χρονικής απόκρισης (Mean), $\bar{x} = 48.234 \text{ sec}$ (διαφορά πραγματικής από μετρούμενη τιμή) με τυπική απόκλιση (Standard Deviation) $s = 35.45 \text{ sec}$ (για όλες τις τιμές χρόνου που προκύπτουν από τον πίνακα 8).

Παρατηρείται ότι μεταβαίνοντας το σύστημα σε μια χαμηλότερη τιμή από αυτή στην οποία βρίσκεται, απαιτεί αρκετά μεγαλύτερο χρονικό διάστημα για να προσεγγίσει το επιθυμητό επίπεδο ροής. Αντίθετα, μεταβαίνοντας σε μια υψηλότερη τιμή απαιτεί λιγότερο χρόνο. Κύρια αίτια αυτής της χρονικής καθυστέρησης αποτελεί ο προστιθέμενος χρόνος που απαιτείται προκειμένου η σφαιρική βαλβίδα να εκτελέσει μια πλήρη περιστροφή 360° , και να επιστρέψει στην αρχική της πλευρά για να ξεκινήσει η κατάσταση ανοίγματος από την αρχή (Όπως αναφέρθηκε και στην ενότητα 4.5 Ηλεκτρική Σφαιρική Βαλβίδα (Electric Valve), η σφαιρική βαλβίδα δεν μπορεί να εκτελέσει κίνηση προς τα πίσω).

Λαμβάνοντας υπόψιν ότι ο χρόνος περιστροφής της βαλβίδας για 90° είναι 16sec, υπολογίζεται ότι ο μέγιστος προστιθέμενος χρόνος στη διαδικασία είναι 64 sec (Ο χρόνος αυτός μπορεί να είναι μικρότερος ανάλογα με το σημείο στο οποίο βρίσκεται εξαρχής η βαλβίδα, δηλαδή αφαιρείται από αυτόν το "position_time"). Αυτή η χρονική υστέρηση μειώνει αισθητά την ευελιξία του ελεγκτή ροής μάζας, παρόλα αυτά είναι απαραίτητη για την ομαλή λειτουργία του συστήματος. Απομονώνοντας το χρόνο αυτό, παρατηρείται, ότι όπως είναι λογικό το σύστημα απαιτεί μικρότερο χρόνο για να μεταβεί από μια τιμή σε μια άλλη που βρίσκεται κοντά σε αυτή. Ενώ για τιμές που έχουν μεγάλη απόσταση μεταξύ τους, απαιτείται μεγαλύτερο χρονικό διάστημα. Είναι σημαντικό να αναφερθεί ο τι για τιμές που ξεπερνούν τα 24 slpm (τιμές που πλησιάζουν το όριο των 25slpm, δηλαδή την μέγιστη τιμή που μπορεί να αποδώσει ο ελεγκτής ροής μάζας) απαιτεί αισθητά παραπάνω χρόνος από το σύστημα προκειμένου να διαμορφώνει τη ροή με επιτυχία σε αυτά τα επίπεδα. Γεγονός που επαληθεύεται, από τις μετρήσεις που είχαν πραγματοποιηθεί για τη σφαιρική βαλβίδα (4.8 Μετρήσεις). Το φαινόμενο αυτό οφείλεται σε κατασκευαστικούς παράγοντες και ο χρόνος που απαιτείται για να επιτευχθούν αυτά τα επίπεδα ροών εξαρτάται από τη καμπυλότητα που χαρακτηρίζει τη σφαιρικής βαλβίδας.

5 ΣΥΜΠΕΡΑΣΜΑΤΑ ΚΑΙ ΜΕΛΛΟΝΤΙΚΕΣ ΕΠΕΚΤΑΣΕΙΣ

5.1 Συμπεράσματα

Σκοπός της διπλωματικής εργασίας ήταν η υλοποίηση ενός συστήματος ελεγκτή ροής μάζας. Μετά τη μελέτη της λειτουργίας αντιστοιχών συστημάτων, τη συλλογή του υλικού και την ανάλυση της λειτουργίας κάθε εξαρτήματος, πραγματοποιήθηκε το κύριο κύκλωμα του συστήματος. Αξίζει να σημειωθεί ότι, ένα βασικό πρόβλημα που προέκυψε κατά τη πειραματική διαδικασία αποτέλεσε η επικοινωνία του αισθητήρα με το μικροελεγκτή. Όπως αναφέρθηκε και στο αντίστοιχο κεφάλαιο, τα μηνύματα κατάστασης του αισθητήρα κατέστρεφαν τη δομή του μηνύματος που δεχόταν ο μικροελεγκτής. Για τη λύση του προβλήματος χρειάστηκε να δημιουργηθεί ένας βρόχος ελέγχου για τον καθαρισμό και την οριοθέτηση του μηνύματος. Στη συνέχεια, έγιναν μετρήσεις στη σφαιρική βαλβίδα. Στη συνέχεια, το πεδίο των μετρήσεων αποτέλεσε σημαντικό κομμάτι για την υλοποίηση του συστήματος, αφού το μεγαλύτερο μέρος της εργασίας αφιερώθηκε στο να γίνουν οι μετρήσεις. Είναι σημαντικό να ανέρθει ότι υπήρξαν αρκετά προβλήματα κατά τη διαδικασία των μετρήσεων, με σημαντικότερο από αυτά να αποτελεί η υπερθέρμανση του συμπιεστή αέρος από την πολύωρη χρήση. Αυτό είχε σαν αποτέλεσμα να υπάρχει πτώση στη τιμή της ροής που είχε αρχικά είχε οριστεί. Επίσης, το κομμάτι των μετρήσεων αποτέλεσε μια αρκετά χρονοβόρα διαδικασία οπου για να γίνουν οι απαραίτητες μετρήσεις χιάστηκαν αρκετές ώρες. Τέλος, το πρόγραμμα του ελεγκτή χρειάστηκε αρκετές παραμετροποιήσεις για να φτάσει στη τελική του μορφή. Αυτό επιτεύχθηκε με συνεχείς δοκιμές και ελέγχους που έγιναν καθόλη τη διάρκεια της επεξεργασίας του, εξασφαλίζοντας έτσι στο σύστημα αμεσότητα και ακρίβεια. Ο ελεγκτής δοκιμαστικέ σε ένα σετ μετρήσεων αποτελούμενο από τριάντα (30) τυχαίες τιμές, οπου διαπιστώθηκε η ομαλή λειτουργία του και ένα σύνολο ικανοποιητικών επιδόσεων.

5.2 Μελλοντικές επεκτάσεις

Το σύστημα διαθέτει όλες τις απαραίτητες λειτουργίες για να φέρει εις πέρας τις απαιτήσεις του χρήστη σε ένα εύρος κλίμακας τιμών ροής από μηδέν (0) έως και εικοσιπέντε (25) slpm. Σε μελλοντικό στάδιο, θα μπορούσε να γίνει επαναπρογραμματισμός του μικροελεγκτή προκειμένου να γίνουν κάποιες βελτιστοποιήσεις στη δομή του προγράμματος. Με αυτό το τρόπο το σύστημα θα είναι σε θέση να ελέγχει περισσότερους παράγοντες κατά τη διάρκεια της λειτουργίας του. Ποιο συγκεκριμένα, θα μπορούσε να αποδεσμευτεί το "Receive με interrupt" (συνάρτηση call back) από εκείνο το κομμάτι του κώδικα, που διαβάζει και συγχρονίζει το μήμα του αισθητήρα. Η τιμή του αισθητήρα μπορεί να λαμβάνεται σε "blocking mode receive" μέσα στη "main". Στη συνέχεια, η συνάρτηση "call back" μπορεί να χρησιμοποιηθεί για τη τιμή "set_point" που εισάγει ο χρήστης. Πιο αναλυτικά, όταν εισάγει νέα τιμή ο χρήστης, τότε, θα τρέξει το "call back", θα διαβάσει τη τιμή, θα την επεξεργαστεί (θα τη μετατρέψει σε setpoint_f) , θα την αποθηκεύσει στη global μεταβλητή και θα ξανά καλέσει το "receive με interrupt" για να λαβή την επόμενη τιμή που ενδεχόμενος να εισάγει ο χρήστης σε επόμενη χρονική στιγμή.

Με αυτό το τρόπο κάθε φορά το callback ενημερώνει συνεχώς το "setpoint_f" για κάθε αλλαγή που μπορεί να πραγματοποιήσει ο χρήστης ενώ τρέχει το σύστημα. Έτσι αποφεύγεται να καλεστεί με "blocking" τρόπο η "Receive" από το σειριακό (θα πραγματοποιείται στα παρασκήνια μέσω του call back). Αυτή η δομή θα μπορούσε να επιτρέψει στο σύστημα να δέχεται πιο άμεσα πλήθος τιμών εισόδου, βελτιώνοντας την ευελιξία του και τον τρόπο που αντιμετωπίζει τις μεταβολές της παροχής.

Θα μπορούσε επίσης να γίνει αναβάθμιση στα υλικά που χρησιμοποιήθηκαν, πιο συγκεκριμένα η βαλβίδα θα μπορούσε να αντικατασταθεί με μια πιο βελτιωμένη έκδοση. Η τωρινή βαλβίδα αποτελεί μια πρώτη προσέγγιση για να αποδειχθεί η υλοποίηση αυτής της μεθοδολογίας. Σε επόμενη φάση θα μπορούσε να χρησιμοποιηθεί μια ταχύτερη βαλβίδα με μεγαλύτερη ακρίβεια. Μια τέτοια βαλβίδα θα μπορούσε να βελτιώσει σημαντικά την απόδοση του συστήματος. Επίσης, μια βαλβίδα με δυνατότητα ανάποδης περιστροφής, θα μείωνε αρκετά το χρόνο απόκρισης, απαλλάσσοντας το χρήστη από το χρόνο αναμονής που χρειάζεται η τωρινή βαλβίδα προκειμένου να κάνει μια πλήρη περιστροφή για να αποδώσει μια μικρότερη τιμή ροής.

Βιβλιογραφία

- [1] Hinkle, L.D. and Mariano, C.F. (1991, May). "Toward understanding the fundamental mechanisms and properties of the thermal mass flow controller", *Journal of Vacuum Science & Technology A: Vacuum, Surfaces, and Films*. [Online]. Vol. 9, issue 3 . Available: <https://doi.org/10.1116/1.577452>
- [2] Tison, S.A. (1996, July). "A critical evaluation of thermal mass flow meters", *Journal of Vacuum Science & Technology A: Vacuum, Surfaces, and Films*. [Online]. Vol. 14 issue 4. Available: <https://doi.org/10.1116/1.579985>
- [3] Kumar GN, S. and AK, M. (2011). "Understanding the Compatibility of Thermal Mass Flow Meter with Various Process Gases", *Journal of Chemical Engineering & Process Technology*, 01(S1). [Online]. Available: <https://doi.org/10.4172/2157-7048.s1-002>
- [4] Yoder, J. (2013). "The History & Evolution of Mass Flow Controllers", *Flow Control Magazine*, pp. 22–26. [Online]. Available: <https://www.sierrainstruments.com/userfiles/file/fc-article-history-evolution-thermal-flowmeters.pdf>
- [5] Thomas, C.C. (1911). "The measurement of gases", *Journal of the Franklin Institute*. [Online]. Vol. 172 issue 5. Available: [https://doi.org/10.1016/S0016-0032\(11\)90212-9](https://doi.org/10.1016/S0016-0032(11)90212-9)
- [6] Laub, J.H. (2013). "An electric flow meter", *Electrical Engineering*. [Online]. Vol. 66 issue 12. Available: <https://doi.org/10.1109/ee.1947.6443880>
- [7] Benson, J.M. (1965). "Thermal flowmeter". United States.[Online] Available: <https://patents.google.com/patent/US3229522A/en>
- [8] Alvesteffer, W. *et al.* (2010) 'A Brief History of the Thermal Mass Flow Meter and Controller', *SVC Summer Bulletin* [Preprint].
- [9] Castro, M.R. de (2016) "*Thermal Mass Flow Sensor Functioning Evaluation and Verification*". Universidade Federal de Santa Catarina. [Online]. Available: <https://doi.org/https://repositorio.ufsc.br/xmlui/handle/123456789/166385>
- [10] Couturier, P. (2009, June). "Advanced control strategy for a digital mass flow controller", *Mechatronics*. [Online]. Vol. 19 issue 4. Available: <https://doi.org/10.1016/j.mechatronics.2008.11.003>
- [11] Choi, Y.M. *et al.* (2014). "Characteristic test methods of the thermal mass flow controller", *Journal of Mechanical Science and Technology*. [Online]. Vol. 28 issue 3. Available: <https://doi.org/10.1007/s12206-013-1158-8>
- [12] Duffy, G.J. "*Thermal Mass Flow Measurement*" MsC. Declaration, Dept. Elect. Eng., Dublin City University, 2000. [Online] Available: <https://core.ac.uk/download/pdf/16510357.pdf>
- [13] Bronkhorst (2021) "*A Fully Optimised Thermal Mass Flow Instrument With the Help of Big Data White Paper*". Available: <https://www.bronkhorst.com/getmedia/f37b18e9-bf85-48d3-aa318d075c4427cf/White-paper-EL-FLOW-Prestige.pdf>

- [14] Lötters, J. (2001, May) "Thermal liquid mass flow controller with fast response and easy rangeability", in *Proceedings of Sensor*. Nurnberg, pp. 257–262. Available: <http://www.tuncell.com/userfiles/sensor01.pdf>
- [15] Incropera, F.P. *et al.* (2011) *Fundamentals of Heat and Mass Transfer. Sixth Edition*, Wiley and Sons, Inc., Hoboken, New Jersey.
- [16] Brooks Instrument (2023) *Mass Flow Controllers & Meters: Mass flow control technology. Unsurpassed process precision.* [Online]. Available: <https://www.brooksinstrument.com/en/products/mass-flow-controllers> (Accessed Jan 27, 2024).
- [17] Crawmer, D.E., Bartoe, R.L. and Kramer, J. (1987, Dec) "Technical note: Improved universal powder mass flow control for thermal spray applications", *Surface and Coatings Technology*. [Online]. Vol. 33(C). Available: [https://doi.org/10.1016/0257-8972\(87\)90201-5](https://doi.org/10.1016/0257-8972(87)90201-5)
- [18] Griffiths, O.G. *et al.* (2013, Mar) "Identifying the largest environmental life cycle impacts during carbon nanotube synthesis via chemical vapour deposition", *Journal of Cleaner Production*. [Online]. Vol. 42. Available: <https://doi.org/10.1016/j.jclepro.2012.10.040>
- [19] Mundra, S.S. *et al.* (2021,) "Development of an integrated physical vapour deposition and chemical vapour deposition system", in *Materials Today: Proceedings*. . [Online]. Vol. 46. Available: <https://doi.org/10.1016/j.matpr.2021.02.069>
- [20] Roberge, D.M. *et al.* (2008) "Development of an industrial multi-injection microreactor for fast and exothermic reactions - Part II", *Chemical Engineering and Technology*. [Online]. Vol. 31 issue 8. Available: <https://doi.org/10.1002/ceat.200800131>
- [21] Viswanathan, M., Rajesh, R. and Kandaswamy, A. (2002, June) "Design and development of thermal mass flowmeters for high pressure applications", *Flow Measurement and Instrumentation*. [Online]. Vol. 13 issue 3. Available: [https://doi.org/10.1016/S0955-5986\(02\)00021-3](https://doi.org/10.1016/S0955-5986(02)00021-3)
- [22] Boyd, K. *et al.* (2011) "A new device for highly accurate gas flow control with extremely fast response times", in *ASMC (Advanced Semiconductor Manufacturing Conference) Proceedings*. [Online] Available: <https://doi.org/10.1109/ASMC.2011.5898201>
- [23] Liso, A., Kehoe, A. and Saleem, M. (2019, Mar) "A new differential pressure sensor based mass flow controller for advanced semiconductor processing", in *China Semiconductor Technology International Conference 2019, CSTIC 2019*. [Online] Available: <https://doi.org/10.1109/CSTIC.2019.8755734>
- [24] Festo AB (2023) *Quiet, fast, inexpensive: mass flow controller for life sciences*, Life Science Sweden. Available: <https://www.lifesciencesweden.se/announcement/view/642146/quiet-fast-inexpensive-mass-flow-controller-for-life-sciences> (Accessed Jan 27, 2024).
- [25] Zarkar, K. (2001) "Requirements for the next generation gas mass flow controllers", in *Semiconductor Measurement Technology: Workshop on Mass Flow Measurement and Control for the Semiconductor Industry*. Washington, DC: US NATIONAL INSTITUTE OF STANDARDS AND TECHNOLOGY. [Online] Available:

<https://doi.org/https://doi.org/10.6028/NIST.SP.400-101>

- [26] Brooks Instrument (2011) *Quantim Coriolis Mass Flow Controllers in Space*. Available: <https://www.brooksinstrument.com/en/blog/quantim-coriolis-mass-flow-controllers-in-space#:~:text=The prototype consists of an,a bladder inside the accumulator> (Accessed Jan 27, 2024).
- [27] Snyder, J.S. *et al.* (2013) "Flow Control and Measurement in Electric Propulsion: Towards an AIAA Reference Standard", *International electric propulsion 33rd* [Preprint].
- [28] Hopkins, M.A. and King, L.B. (2014) "Magnesium hall thruster with active thermal mass flow control", *Journal of Propulsion and Power*. [Online]. Vol. 30 issue 3. Available: <https://doi.org/10.2514/1.B34888>
- [29] Zhou, G. *et al.* (2021) "Blockchain-Based Data Ownership Confirmation Scheme in Industrial Internet of Things", in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*. Available: https://doi.org/10.1007/978-3-030-85928-2_10
- [30] Yu, Y.M. *et al.* (2020, Jan) "Design and application of thermal mass flow meter in space", *Nuclear Instruments and Methods in Physics Research, Section A: Accelerators, Spectrometers, Detectors and Associated Equipment*. [Online]. Vol. 950. Available: <https://doi.org/10.1016/j.nima.2019.163003>
- [31] Bekraoui, A. and Hadjadj, A. (2020, Sep) "Thermal flow sensor used for thermal mass flowmeter", *Microelectronics Journal*. [Online]. Vol. 103. Available: <https://doi.org/10.1016/j.mejo.2020.104871>
- [32] Gopinath, A. (2023) *Design, Implementation and Characterization of a Thermal-Mass Flow Meter for Microfluidics*. Politecnico Milano. Available: <https://doi.org/https://hdl.handle.net/10589/210736>
- [33] Liu, Y. *et al.* (2023) "Experimental and numerical study on a novel thermal mass flowmeter that is insensitive to radial flow velocity distribution", *Flow Measurement and Instrumentation*. [Online]. Vol. 90. Available: <https://doi.org/10.1016/j.flowmeasinst.2023.102315>
- [34] S. Mackay, E. Wright, D. Reynders and J. Park, "EIA-232 overview" in *Practical Industrial Data Networks Design, Installation and Troubleshooting*, V. Mehra, Eds. GB: Newnes, 2004, pp. 32-52.
- [35] McGraw-Hili, "ANSI EIA/TIA-232-E Interface Standard", Datapro Information Services Group, Delran NJ 08075, USA, 1995.
- [36] D. Ibrahim, "RS232 Serial Communication Projects-RS232 Conectors" in *Microcontroller Projects in C for the 8051*, Eds. GB: Newnes, 2000, pp. 148-151.
- [37] RS232 Connector Pinouts(2024, Jan, 26)[Online]. Available: <http://www.piclist.com/techref/io/serial/rs232db.html> (Accessed Jan 27, 2024)

- [38] J.L. Francis and T.R.P. Martin,(1990,Dec). "Principles of interfacing computers to medical equipment" [Online]. vol 4, issue 4 , pp 787-795.
Available: www.sciencedirect.com/science/article/pii/S095035520580344X
- [39] Sea Bird Scientific. What is the maximum cable length for real-time RS-232 data?[Online]. Available: <https://blog.seabird.com/ufags/what-is-the-maximum-cable-length-for-real-time-rs-232-data/> (Accessed Jan 27, 2024).
- [40] G. Kaltsas, " Embedded Systems - Final_2022", MC&ES Presentation, Dept. Electrical & Electronics Eng.,Univ. of West Attica, Aigaleo, Ath, 2022.
- [41] Keskin, B., Celik, M. S., and Eminoglu, I., "Four-Axis Stepper Motor Training Set: STM32 Microcontroller, Algorithm and Teaching Considerations", in *Proceedings - 2022 International Conference on Artificial Intelligence of Things, (ICAIoT)*, Istanbul, TR, 2022.
- [42] Yokoyama, D., Schulze, B., Borges, F., and Mc Evoy, G.(2019, June). "The survey on ARM processors for HPC" *Journal of Supercomputing* [Online]. vol. 75, issue 10.
Available: <https://doi.org/10.1007/s11227-019-02911-9>
- [43] Irmak, E., Colak, I., Kabalci, E., and Kose, A., "Implementation of an interactive remote laboratory platform for stepper motor experiments" in *15th International Power Electronics and Motion Control Conference and Exposition (EPE/PEMC)*, Novi Sad, Serbia, 2012.
- [44] Chu, R. H., Lu, D. D. C., and Sathiakumar, S. (2008, Feb)."Project-based lab teaching for power electronics". *IEEE Transactions on Education* [Online], vol. 51, issue 1, pages 108 – 113. Available: <https://doi.org/10.1109/TE.2007.906607>
- [45] Aboshosha, B., Dessouky, M., Ramadan, R., El-Sayed, A., and Galalb, F.(2018, Sep). "Evaluation of Lightweight Block Ciphers Based on General Feistel Structure (GFS)" in *WAS Science Nature (WASSN)* [Online], vol. 1, issue 1.
Available: <https://worldscience.com/journals/index.php/wassn/article/view/32>
- [46] Zhang, J., Peng, G., Yang, H., Tan, C., Tan, Y., and Bai, H. (2023, Jan). " Real-Time Finger-Writing Character Recognition via ToF Sensors on Edge Deep Learning". *Electronics (Switzerland)*[Online]. vol.12, issue 3.Avaliable: <https://doi.org/10.3390/electronics12030685>
- [47] Staszewski, W., Jablonski, A., & Barszcz, T.,"Novel micro, wireless, mems-based condition monitoring system for modern machine tools with limited access" Dept. Mechanical Eng., AGH University of Science and Technology, Kraków, Polska, 2017.
Available:<https://eadn-wc01-4174605.nxedge.io/wp-content/uploads/2019/10/Paper- - Staszewski Jablonski Barszcz.pdf>
- [48] Mirisola, D., "Firmware update for 6LoWPAN networks of OMA-LwM2M IoT devices", MSc. Thesis, Dept. Computer Eng., Politecnico di Torino, Torino, Italy, 2018.

- [49] Zhou, G., Yan, B., Wang, G., & Yu, J., "Blockchain-Based Data Ownership Confirmation Scheme in Industrial Internet of Things", in *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, Springer Nature, Switzerland, 2021
- [50] STMicroelectronics, "STM32F401xD STM32F401xE" [Online], Jan, 2015. Available: <https://www.st.com/content/ccc/resource/technical/document/datasheet/30/91/86/2d/db/94/4a/d6/DM00102166.pdf/files/DM00102166.pdf/jcr:content/translations/en.DM00102166.pdf>
- [51] Gök, M., Görgünoğlu, S., & Sefa, I. "Design of a real-time USB interfaced multi-channel power system harmonics detection system" in *ELECO - 9th International Conference on Electrical and Electronics Engineering*, Bursa, Turkey, 2015.
- [52] Alicat Scientific, Alicat Scientific, about as [Online]. Available: <https://www.alicat.com/about-us/>(Accessed Jan 27, 2024)
- [53] Alicat Scientific, "Technical Data for M-Series Mass Flow Meters"[Online], Nov. 2023. Available: <https://www.alicat.com/wp-content/documents/specifications/DOC-SPECS-M-MID.pdf>
- [54] Alicat Scientific, " DOC-MANUAL-M"[Online], Rev. 1, Jun. 2020. Available: <https://www.alicat.com/wp-content/documents/manuals/DOC-MANUAL-M.pdf>
- [55] N. Suresh, " EMBEDDED SYSTEMS DESIGN", lecture notes, ECE Dept, Malla Reddy College of Engineering & Technology, Hyderabad, Telangana, India, 2017.
- [56] Circuits-Diy, MAX232 Dual RS232 Receiver/Transmitter–Datasheet [Online] Available: <https://www.circuits-diy.com/max232-dual-rs232-receiver-transmitter-datasheet/> (accessed Jan 27, 2024)
- [57] Stack exchange, MAX3232 overheating/burnt after connecting to PC [Online]. Available: <https://electronics.stackexchange.com/questions/122769/max3232-overheating-burnt-after-connecting-to-pc> (Accessed Jan 27, 2024)
- [58] Electronics.(1986,7 24). *A new class of dsp chip: The vector signal processor* [Online]. Available: <https://www.worldradiohistory.com/Archive-Electronics/80s/86/Electronics-1986-07-24.pdf>
- [59] AliExpress, Electric Valve DN15 DN20 DN25 DN32 AC220V Electric Ball Valve Brass Motorized Ball Valve Switch type electric two-way valves [Online]. Available: <https://www.aliexpress.com/i/32948005141.html>
- [60] XHVAL, Ball Valve Direction: Which Way To Handle?[Online]. Available: <https://www.xhval.com/ball-valve-direction/>

Παράρτημα Α

Δημιουργία interface του αισθητήρα:

```

/* Includes -----*/
#include <stdio.h>
#include <usart.h>
#include <gpio.h>
#include <stdlib.h>
/* USER CODE BEGIN PV */
uint8_t rx_buffer[39];
/* USER CODE END PV */
/* Private function prototypes -----*/
void SystemClock_Config(void);
/* USER CODE BEGIN PFP */
#ifdef __GNUC__
#define PUTCHAR_PROTOTYPE int __io_putchar(int ch)
#else
#define PUTCHAR_PROTOTYPE int fputc(int ch, FILE *f)
#endif
PUTCHAR_PROTOTYPE
{
    HAL_UART_Transmit(&huart2, (uint8_t *)&ch, 1, 1000);
    return ch;} /* USER CODE END PFP */
/* Private user code -----*/
/* USER CODE BEGIN 0 */
float convert_to_float(uint8_t * rx_buffer, int choice)
{
    int start = -1;
    switch(choice)
    {
        case 1:
            start = 1;
            break;
        case 2:
            start = 9;
            break;
        case 3:
            start = 17;
            break;
        case 4:
            start = 25;
            break;
        default:
            fprintf(stderr, "ERROR @ convert_to_float, choice must be 1<=choice<=4.\n");
            return -1;
    }
    float f_num = 0.0;
    char integer_part[] = {rx_buffer[start],rx_buffer[start+1],rx_buffer[start+2]};
    char fractional_part[] = {rx_buffer[start+4],rx_buffer[start+5]};
    int integer_part_int = atoi(integer_part);
    int fractional_part_int = atoi(fractional_part);
    f_num = integer_part_int;
    f_num += fractional_part_int/100.0;
    return f_num;} /* USER CODE END 0 */

int main(void){
/* Reset of all peripherals, Initializes the Flash interface and the Systick. */
HAL_Init();
/* Configure the system clock */
SystemClock_Config();
/* Initialize all configured peripherals */
MX_GPIO_Init();
MX_USART1_UART_Init();

```

```

MX_USART2_UART_Init();
/* USER CODE BEGIN 2 */
HAL_UART_Receive_IT(&huart1, rx_buffer, sizeof(rx_buffer));
/* USER CODE END 2 */

while (1){/* USER CODE BEGIN WHILE */

//printing MFM data
//Printing Data (ASCII)
printf("-----ASCII-----\n\r");
printf(" Absolute Pressure: %c%c%c%c%c%c%c%c\n\r"
,rx_buffer[0],rx_buffer[1],rx_buffer[2],rx_buffer[3]
,rx_buffer[4],rx_buffer[5],rx_buffer[6],rx_buffer[7]);
printf("Temperature: %c%c%c%c%c%c%c%c\n\r"
,rx_buffer[8],rx_buffer[9],rx_buffer[10],rx_buffer[11]
,rx_buffer[12],rx_buffer[13],rx_buffer[14],rx_buffer[15]);
printf("Volumetric Flow: %c%c%c%c%c%c%c%c\n\r"
,rx_buffer[16],rx_buffer[17],rx_buffer[18],rx_buffer[19]
,rx_buffer[20],rx_buffer[21],rx_buffer[22],rx_buffer[23]);
printf("Standard (Mass)Flow: %c%c%c%c%c%c%c%c%c%c%c%c\n\r"
,rx_buffer[24],rx_buffer[25],rx_buffer[26],rx_buffer[27]
,rx_buffer[28],rx_buffer[29],rx_buffer[30],rx_buffer[31]
,rx_buffer[32],rx_buffer[33],rx_buffer[34],rx_buffer[35]);
printf("Gas Type: %c%c%c%c\n\r"
,rx_buffer[36],rx_buffer[37],rx_buffer[38],rx_buffer[39]);
//Printing Data (DECIMAL)
printf("-----DEC-----\n\r");
printf(" Absolute Pressure:%d %d %d %d %d %d %d %d\n\r"
,rx_buffer[0],rx_buffer[1],rx_buffer[2],rx_buffer[3]
,rx_buffer[4],rx_buffer[5],rx_buffer[6],rx_buffer[7]);
printf("Temperature:%d %d %d %d %d %d %d %d\n\r"
,rx_buffer[8],rx_buffer[9],rx_buffer[10],rx_buffer[11]
,rx_buffer[12],rx_buffer[13],rx_buffer[14],rx_buffer[15]);
printf("Volumetric Flow:%d %d %d %d %d %d %d %d\n\r"
,rx_buffer[16],rx_buffer[17],rx_buffer[18],rx_buffer[19]
,rx_buffer[20],rx_buffer[21],rx_buffer[22],rx_buffer[23]);
printf("Standard (Mass)Flow:%d %d %d %d %d %d %d %d %d %d %d %d\n\r"
//      +      0      0      0
//      43      48      48      48
,rx_buffer[24],rx_buffer[25],rx_buffer[26],rx_buffer[27]
//      .      0      0      [space]
//      46      48      48      32
,rx_buffer[28],rx_buffer[29],rx_buffer[30],rx_buffer[31]
//      [space]      [space]      [space]      [space]
//      32      32      32      32
,rx_buffer[32],rx_buffer[33],rx_buffer[34],rx_buffer[35]);
printf("Gas Type:%d %d %d %d\n\r"
,rx_buffer[36],rx_buffer[37],rx_buffer[38],rx_buffer[39]);
printf("-----\n\r");
//converts the printed value to a float variable
printf("AP: %.2f\n\r",convert_to_float(rx_buffer,1));
printf("T: %.2f\n\r",convert_to_float(rx_buffer,2));
printf("VF: %.2f\n\r",convert_to_float(rx_buffer,3));
printf("SMF: %.2f\n\r",convert_to_float(rx_buffer,4));
printf(" \n\r-----\n\r");
HAL_Delay(50);
}
}

void SystemClock_Config(void){
RCC_OscInitTypeDef RCC_OscInitStruct = {0};
RCC_ClkInitTypeDef RCC_ClkInitStruct = {0};
// Configure the main internal regulator output voltage

```



```

__HAL_RCC_PWR_CLK_ENABLE();
__HAL_PWR_VOLTAGESCALING_CONFIG(PWR_REGULATOR_VOLTAGE_SCALE2);
//Initializes the RCC Oscillators according to the specified parameters
//in the RCC_OscInitTypeDef structure.
RCC_OscInitStruct.OscillatorType = RCC_OSCILLATORTYPE_HSI;
RCC_OscInitStruct.HSIState = RCC_HSI_ON;
RCC_OscInitStruct.HSICalibrationValue = RCC_HSICALIBRATION_DEFAULT;
RCC_OscInitStruct.PLL.PLLState = RCC_PLL_ON;
RCC_OscInitStruct.PLL.PLLSource = RCC_PLLSOURCE_HSI;
RCC_OscInitStruct.PLL.PLLM = 8;
RCC_OscInitStruct.PLL.PLLN = 84;
RCC_OscInitStruct.PLL.PLLP = RCC_PLLP_DIV2;
RCC_OscInitStruct.PLL.PLLQ = 4;
if (HAL_RCC_OscConfig(&RCC_OscInitStruct) != HAL_OK)
{Error_Handler();}
//Initializes the CPU, AHB and APB buses clocks
RCC_ClkInitStruct.ClockType = RCC_CLOCKTYPE_HCLK|RCC_CLOCKTYPE_SYSCLK
|RCC_CLOCKTYPE_PCLK1|RCC_CLOCKTYPE_PCLK2;
RCC_ClkInitStruct.SYSCLKSource = RCC_SYSCLKSOURCE_PLLCLK;
RCC_ClkInitStruct.AHBCLKDivider = RCC_SYSCLK_DIV1;
RCC_ClkInitStruct.APB1CLKDivider = RCC_HCLK_DIV2;
RCC_ClkInitStruct.APB2CLKDivider = RCC_HCLK_DIV1;
if (HAL_RCC_ClockConfig(&RCC_ClkInitStruct, FLASH_LATENCY_2) != HAL_OK)
{Error_Handler();}
}
/* USER CODE BEGIN 4 */
// Function to flush unwanted data
void flushUartBuffer(UART_HandleTypeDef *huart) {}
// Keep reading bytes until the buffer is empty or a specific condition is met
void HAL_UART_RxCpltCallback(UART_HandleTypeDef *huart)
{
    uint8_t temp;
    //Synchronizes the MFM message with the rx_buffer
    while (1){
        HAL_UART_Receive(&huart1, &temp, 1, HAL_MAX_DELAY);
        //printf("%c",temp);
        //fflush(stdout);
//Check for a marker or delimiter to indicate the start of a new message
        if (temp == 13){
            //printf("\n\r");
            //fflush(stdout);
//Break if the marker is found to synchronize with the start of a new messg
            break;
        }
    }
    HAL_UART_Receive_IT(&huart1, rx_buffer, sizeof(rx_buffer));
}
/* USER CODE END 4 */
//@brief This function is executed in case of error occurrence.
//@retval None
void Error_Handler(void)
{
    __disable_irq();
    while (1){}
}
#ifdef USE_FULL_ASSERT
void assert_failed(uint8_t *file, uint32_t line) {}
#endif
/* USE_FULL_ASSERT */

```

Παράρτημα Β

Πρόγραμμα για την υλοποίηση του καθορισμού δυναμικής απόκρισης της βαλβίδας:

```

/* Includes -----*/
#include <main.h>
#include <stdio.h>
#include <usart.h>
#include <gpio.h>
#include <stdlib.h>
/* USER CODE BEGIN PV */
uint8_t rx_buffer[39];
uint8_t state;
int TIMER=2200;
int COUNTER=0;
/* USER CODE END PV */
/* Private function prototypes -----*/
void SystemClock_Config(void);
/* USER CODE BEGIN PFP */
#ifdef __GNUC__
#define PUTCHAR_PROTOTYPE int __io_putchar(int ch)
#else
#define PUTCHAR_PROTOTYPE int fputc(int ch, FILE *f)
#endif
PUTCHAR_PROTOTYPE
{HAL_UART_Transmit(&huart2, (uint8_t *)&ch, 1, 1000);
return ch;}/* USER CODE END PFP */
/* Private user code -----*/
/* USER CODE BEGIN 0 */
float convert_to_float(uint8_t * rx_buffer, int choice)
{int start = -1;
switch(choice){
case 1:
start = 1;
break;
case 2:
start = 9;
break;
case 3:
start = 17;
break;
case 4:
start = 25;
break;
default:
fprintf(stderr,"ERROR @ convert_to_float, choice must be 1<=choice<=4.\n");
return -1;
}
float f_num = 0.0;
char integer_part[] = {rx_buffer[start],rx_buffer[start+1],rx_buffer[start+2]};
char fractional_part[] = {rx_buffer[start+4],rx_buffer[start+5]};
int integer_part_int = atoi(integer_part);
int fractional_part_int = atoi(fractional_part);
f_num = integer_part_int;
f_num += fractional_part_int/100.0;
return f_num;}/* USER CODE END 0 */

int main(void){
//Reset of all peripherals,Initializes the Flash interface and the SysTick.
HAL_Init();
/* Configure the system clock */
SystemClock_Config();

```

```

/* Initialize all configured peripherals */
MX_GPIO_Init();
MX_USART1_UART_Init();
MX_USART2_UART_Init();
/* USER CODE BEGIN 2 */
HAL_UART_Receive_IT(&huart1, rx_buffer, sizeof(rx_buffer));

//.....START LAG ROUTINE.....
// SET ON RELAY THAT OPENS VALVE (Blue cable)
HAL_GPIO_WritePin(GPIOA, GPIO_PIN_11, GPIO_PIN_SET);
HAL_Delay(2200);
// SET OFF RELAY THAT OPENS VALVE (Blue cable)
HAL_GPIO_WritePin(GPIOA, GPIO_PIN_11, GPIO_PIN_RESET);
//.....
/* USER CODE END 2 */
while (1){/* USER CODE BEGIN WHILE */
    do{
        //.....Print DATA (ASCII).....
        printf("Standard (Mass)Flow:%c%c%c%c%c%c%c%c%c%c%c\n\r"
//          +          0          0          0
//          43         48         48         48
        ,rx_buffer[24],rx_buffer[25],rx_buffer[26],rx_buffer[27]
//          .          0          0          [space]
//          46         48         48         32
        ,rx_buffer[28],rx_buffer[29],rx_buffer[30],rx_buffer[31]
//          [space] [space] [space] [space]
//          32         32         32         32
        ,rx_buffer[32],rx_buffer[33],rx_buffer[34],rx_buffer[35]);
        HAL_Delay(200);
        //.....
        //checking if button state is not pressed
        if(HAL_GPIO_ReadPin(GPIOC, GPIO_PIN_13) == GPIO_PIN_RESET)
        {state=0;}
        else{state=1;}
        // stops reading and starts making changes when button is pressed
        }while (state == 1);
        state=0;
        //.....STEP.....
        // SET ON RELAY THAT OPENS VALVE (Blue cable)
        HAL_GPIO_WritePin(GPIOA, GPIO_PIN_11, GPIO_PIN_SET);
        HAL_Delay(25); // STEP:50,35 or 10ms
        // SET OFF RELAY THAT OPENS VALVE (Blue cable)
        HAL_GPIO_WritePin(GPIOA, GPIO_PIN_11, GPIO_PIN_RESET);
        //.....
        COUNTER = COUNTER + 1;
        TIMER= TIMER + 25; //50,35 or 10ms;
        printf("MEASURE:%d\n\r TIME:%dms\n\r",COUNTER,TIMER);
    }/* USER CODE END WHILE */
}

void SystemClock_Config(void){
RCC_OscInitTypeDef RCC_OscInitStruct = {0};
RCC_ClkInitTypeDef RCC_ClkInitStruct = {0};
// Configure the main internal regulator output voltage
__HAL_RCC_PWR_CLK_ENABLE();
__HAL_PWR_VOLTAGESCALING_CONFIG(PWR_REGULATOR_VOLTAGE_SCALE2);
//Initializes the RCC Oscillators according to the specified parameters
//in the RCC_OscInitTypeDef structure.
RCC_OscInitStruct.OscillatorType = RCC_OSCILLATORTYPE_HSI;
RCC_OscInitStruct.HSIState = RCC_HSI_ON;
RCC_OscInitStruct.HSICalibrationValue = RCC_HSICALIBRATION_DEFAULT;
RCC_OscInitStruct.PLL.PLLState = RCC_PLL_ON;
RCC_OscInitStruct.PLL.PLLSource = RCC_PLLSOURCE_HSI;
RCC_OscInitStruct.PLL.PLLM = 8;

```

```

RCC_OscInitStruct.PLL.PLLN = 84;
RCC_OscInitStruct.PLL.PLLP = RCC_PLLP_DIV2;
RCC_OscInitStruct.PLL.PLLQ = 4;
if (HAL_RCC_OscConfig(&RCC_OscInitStruct) != HAL_OK)
{Error_Handler();}
//Initializes the CPU, AHB and APB buses clocks
RCC_ClkInitStruct.ClockType = RCC_CLOCKTYPE_HCLK|RCC_CLOCKTYPE_SYSCLK
|RCC_CLOCKTYPE_PCLK1|RCC_CLOCKTYPE_PCLK2;
RCC_ClkInitStruct.SYSCLKSource = RCC_SYSCLKSOURCE_PLLCLK;
RCC_ClkInitStruct.AHBCLKDivider = RCC_SYSCLK_DIV1;
RCC_ClkInitStruct.APB1CLKDivider = RCC_HCLK_DIV2;
RCC_ClkInitStruct.APB2CLKDivider = RCC_HCLK_DIV1;
if (HAL_RCC_ClockConfig(&RCC_ClkInitStruct, FLASH_LATENCY_2) != HAL_OK)
{Error_Handler();}
}
/* USER CODE BEGIN 4 */
// Function to flush unwanted data
void flushUartBuffer(UART_HandleTypeDef *huart) {}
//Keep reading bytes until the buffer is empty or a specific condition met
void HAL_UART_RxCpltCallback(UART_HandleTypeDef *huart){
uint8_t temp;
//Synchronizes the MFM message with the rx_buffer
while (1){
HAL_UART_Receive(&huart1, &temp, 1, HAL_MAX_DELAY);
//printf("%c",temp);
//fflush(stdout);
// Check for a marker or delimiter to indicate the start of a new message
if (temp == 13){
//printf("\n\r");
//fflush(stdout);
//Break if the marker is found to synchronize with the start of a new message
break;}
}
HAL_UART_Receive_IT(&huart1, rx_buffer, sizeof(rx_buffer));
}
/* USER CODE END 4 */
//@brief This function is executed in case of error occurrence.
//@retval None
void Error_Handler(void)
{__disable_irq();
while (1){}}
#ifdef USE_FULL_ASSERT
void assert_failed(uint8_t *file, uint32_t line) {}
#endif /* USE_FULL_ASSERT */

```

Παράρτημα Γ

Πρόγραμμα για την υλοποίηση του καθορισμού ροής μέσω ελέγχου της βαλβίδας:

```

/* Includes -----*/
#include <stdio.h>
#include <main.h>
#include <usart.h>
#include <gpio.h>
#include <string.h>
#include <stdlib.h>
/* USER CODE BEGIN PV */
uint8_t rx_buffer[39];
uint8_t state;
int TIMER=2200;
int COUNTER=0;
int DELAY_COUNTER=0;
/* USER CODE END PV */
/* Private function prototypes -----*/
void SystemClock_Config(void);
/* USER CODE BEGIN PFP */
#ifdef __GNUC__
#define PUTCHAR_PROTOTYPE int __io_putchar(int ch)
#else
#define PUTCHAR_PROTOTYPE int fputc(int ch, FILE *f)
#endif
PUTCHAR_PROTOTYPE
{ HAL_UART_Transmit(&huart2, (uint8_t *)&ch, 1, 1000);
  return ch;
} /* USER CODE END PFP */
/* Private user code -----*/
/* USER CODE BEGIN 0 */
float convert_to_float(uint8_t * rx_buffer, int choice)
{ int start = -1;
  switch(choice){
    case 1:
      start = 1;
      break;
    case 2:
      start = 9;
      break;
    case 3:
      start = 17;
      break;
    case 4:
      start = 25;
      break;
    default:
      fprintf(stderr, "ERROR @ convert_to_float, choice must be 1<=choice<=4.\n");
      return -1; }
  float f_num = 0.0;
  char integer_part[] = {rx_buffer[start],rx_buffer[start+1],rx_buffer[start+2]};
  char fractional_part[] = {rx_buffer[start+4],rx_buffer[start+5]};
  int integer_part_int = atoi(integer_part);
  int fractional_part_int = atoi(fractional_part);
  f_num = integer_part_int;
  f_num += fractional_part_int/100.0;
  return f_num;} /* USER CODE END 0 */

int main(void){
//Reset of all peripherals,Initializes the Flash interface and the SysTick.
  HAL_Init();

```

```

/* Configure the system clock */
SystemClock_Config();
/* Initialize all configured peripherals */
MX_GPIO_Init();
MX_USART1_UART_Init();
MX_USART2_UART_Init();
/* USER CODE BEGIN 2 */
HAL_UART_Receive_IT(&huart1, rx_buffer, sizeof(rx_buffer));
/* USER CODE END 2 */
while (1){/* USER CODE BEGIN WHILE */
    do {
//.....Print DATA (ASCII).....
        printf("%%c%%c%%c%%c%%c%%c%%c%%c%%c%%c%%c%%c\n\r"
            //      +      0      0      0
            //      43      48      48      48
            ,rx_buffer[24],rx_buffer[25],rx_buffer[26],rx_buffer[27]
            //      .      0      0      [space]
            //      46      48      48      32
            ,rx_buffer[28],rx_buffer[29],rx_buffer[30],rx_buffer[31]
            //      [space]      [space]      [space]      [space]
            //      32      32      32      32
            ,rx_buffer[32],rx_buffer[33],rx_buffer[34],rx_buffer[35]);
        HAL_Delay(200);
//.....
        //checking if button state is not pressed
        if (HAL_GPIO_ReadPin(GPIOC, GPIO_PIN_13) == GPIO_PIN_RESET)
            {state = 0;}
        else{state = 1;}
        //stops reading and starts making changes when button is pressed
    }while (state == 1);
        state = 0;

        COUNTER = COUNTER + 1;
        DELAY_COUNTER = DELAY_COUNTER + 25;
        TIMER= TIMER + 25;

//.....RESET ROUTINE.....
//.....THE VALVE OPENS.....

        //SET ON RELAY THAT OPENS VALVE (Blue cable)
        HAL_GPIO_WritePin(GPIOA, GPIO_PIN_11, GPIO_PIN_SET);
        HAL_Delay(17000-DELAY_COUNTER); //Delay
        //SET OFF RELAY THAT OPENS VALVE (Blue cable)
        HAL_GPIO_WritePin(GPIOA, GPIO_PIN_11, GPIO_PIN_RESET);
        HAL_Delay(500); //Delay between change

        //.....THE VALVE CLOSES.....

        //SET ON RELAY THAT CLOSE VALVE(Grey cable)
        HAL_GPIO_WritePin(GPIOA, GPIO_PIN_12, GPIO_PIN_SET);
        HAL_Delay(16000); //Delay 16 sec
        //SET OFF RELAY THAT CLOSE VALVE(Grey cable)
        HAL_GPIO_WritePin(GPIOA, GPIO_PIN_12, GPIO_PIN_RESET);

        //.....THE VALVE OPENS.....

        //SET ON RELAY THAT OPENS VALVE (Blue cable)
        HAL_GPIO_WritePin(GPIOA, GPIO_PIN_11, GPIO_PIN_SET);
        HAL_Delay(16000); //Delay
        //SET OFF RELAY THAT OPENS VALVE (Blue cable)
        HAL_GPIO_WritePin(GPIOA, GPIO_PIN_11, GPIO_PIN_RESET);
        HAL_Delay(500); //Delay between change
    }
}

```

```

//.....THE VALVE CLOSES.....

//SET ON RELAY THAT CLOSE VALVE(Grey cable)
HAL_GPIO_WritePin(GPIOA, GPIO_PIN_12, GPIO_PIN_SET);
    HAL_Delay(16000);//Delay 16 sec
//SET OFF RELAY THAT CLOSE VALVE(Grey cable)
HAL_GPIO_WritePin(GPIOA, GPIO_PIN_12, GPIO_PIN_RESET);
//.....

//.....THE VALVE OPENS (START LAG).....

//SET ON RELAY THAT OPENS VALVE (Blue cable)
HAL_GPIO_WritePin(GPIOA, GPIO_PIN_11, GPIO_PIN_SET);
    HAL_Delay(2200);
//SET OFF RELAY THAT OPENS VALVE (Blue cable)
HAL_GPIO_WritePin(GPIOA, GPIO_PIN_11, GPIO_PIN_RESET);
    HAL_Delay(2000);//Delay before start

//.....THE VALVE OPENS FOR MEASUREMENT.....

//SET ON RELAY THAT OPENS VALVE (Blue cable)
HAL_GPIO_WritePin(GPIOA, GPIO_PIN_11, GPIO_PIN_SET);
    HAL_Delay(DELAY_COUNTER);// Step DELay
//SET OFF RELAY THAT OPENS VALVE (Blue cable)
HAL_GPIO_WritePin(GPIOA, GPIO_PIN_11, GPIO_PIN_RESET);

//Printing Counters
printf("-----\n\r-MEASURE:%d\n\r-SIDE:%d\n\r-TIME:%dms\n\r-
DELAY:%d\n\r",COUNTER,SIDE,TIMER,DELAY_COUNTER);
printf("\n\r");
} } /* USER CODE END WHILE */
void SystemClock_Config(void){
    RCC_OscInitTypeDef RCC_OscInitStruct = {0};
    RCC_ClkInitTypeDef RCC_ClkInitStruct = {0};
    // Configure the main internal regulator output voltage
    __HAL_RCC_PWR_CLK_ENABLE();
    __HAL_PWR_VOLTAGESCALING_CONFIG(PWR_REGULATOR_VOLTAGE_SCALE2);
    //Initializes the RCC Oscillators according to the specified parameters
    //in the RCC_OscInitTypeDef structure.
    RCC_OscInitStruct.OscillatorType = RCC_OSCILLATORTYPE_HSI;
    RCC_OscInitStruct.HSIState = RCC_HSI_ON;
    RCC_OscInitStruct.HSICalibrationValue = RCC_HSICALIBRATION_DEFAULT;
    RCC_OscInitStruct.PLL.PLLState = RCC_PLL_ON;
    RCC_OscInitStruct.PLL.PLLSource = RCC_PLLSOURCE_HSI;
    RCC_OscInitStruct.PLL.PLLM = 8;
    RCC_OscInitStruct.PLL.PLLN = 84;
    RCC_OscInitStruct.PLL.PLLP = RCC_PLLP_DIV2;
    RCC_OscInitStruct.PLL.PLLQ = 4;
    if (HAL_RCC_OscConfig(&RCC_OscInitStruct) != HAL_OK)
    {Error_Handler();}
    //Initializes the CPU, AHB and APB buses clocks
    RCC_ClkInitStruct.ClockType = RCC_CLOCKTYPE_HCLK|RCC_CLOCKTYPE_SYSCLK
        |RCC_CLOCKTYPE_PCLK1|RCC_CLOCKTYPE_PCLK2;
    RCC_ClkInitStruct.SYSCLKSource = RCC_SYSCLKSOURCE_PLLCLK;
    RCC_ClkInitStruct.AHBCLKDivider = RCC_SYSCLK_DIV1;

    RCC_ClkInitStruct.APB1CLKDivider = RCC_HCLK_DIV2;
    RCC_ClkInitStruct.APB2CLKDivider = RCC_HCLK_DIV1;
    if (HAL_RCC_ClockConfig(&RCC_ClkInitStruct, FLASH_LATENCY_2) != HAL_OK)
    {Error_Handler();}
}
/* USER CODE BEGIN 4 */
// Function to flush unwanted data

```

```
void flushUartBuffer(UART_HandleTypeDef *huart) {}
//Keep reading bytes until the buffer is empty or a specific condition met
void HAL_UART_RxCpltCallback(UART_HandleTypeDef *huart){
    uint8_t temp;
    //Synchronizes the MFM message with the rx_buffer
    while (1){
        HAL_UART_Receive(&huart1, &temp, 1, HAL_MAX_DELAY);
        //printf("%c",temp);
        //fflush(stdout);
    // Check for a marker or delimiter to indicate the start of a new message
        if (temp == 13)
        {
            //printf("\n\r");
            //fflush(stdout);
        //Break if the marker is found to synchronize with the start of a newmessage
            break;}}
        HAL_UART_Receive_IT(&huart1, rx_buffer, sizeof(rx_buffer)); /* USER CODE END 4 */
    // @brief This function is executed in case of error occurrence.
    // @retval None
void Error_Handler(void)
{__disable_irq();
 while (1){}}
#ifdef USE_FULL_ASSERT
void assert_failed(uint8_t *file, uint32_t line){ }
#endif /* USE_FULL_ASSERT */
```


Παράρτημα Δ

Πρόγραμμα ελεγκτή ροής μάζας:

```

/* Includes -----*/
#include <stdio.h>
#include <main.h>
#include <usart.h>
#include <gpio.h>
#include <string.h>
#include <stdlib.h>
#include <math.h>
/* USER CODE BEGIN PV */
uint8_t tx_buffer[30] = "\n\rGive a setpoint e.g[000.00]:";
uint8_t rx_buffer[39];
uint8_t set_point[6];
float time;
float position_time;
float meter_value;
float set_point_f;
float SYSTEM_DEVIATION = 00.30;
//Model      ExpGrow2 parameters
double x0 = 4.67662;
double A1 = 1.60641E-4;
double t1 = 1.23137;
double A2 = 6979.17895;
double t2 = 139.39601;
double y_0 = -4322.98261;
/* USER CODE END PV */
/* Private function prototypes -----*/
void SystemClock_Config(void);
/* USER CODE BEGIN PFP */
#ifdef __GNUC__
#define PUTCHAR_PROTOTYPE int __io_putchar(int ch)
#else
#define PUTCHAR_PROTOTYPE int fputc(int ch, FILE *f)
#endif
PUTCHAR_PROTOTYPE {
    HAL_UART_Transmit(&huart2, (uint8_t*) &ch, 1, HAL_MAX_DELAY);
    return ch;
}
/* USER CODE BEGIN 0 */
float setpoint_convert_to_float(uint8_t *set_point) {
    float setpoint_f_num = 0.0;
    char setpoint_integer_part[] = { set_point[0], set_point[1], set_point[2] };
    char setpoint_fractional_part[] = { set_point[4], set_point[5] };
    int setpoint_integer_part_int = atoi(setpoint_integer_part);
    int setpoint_fractional_part_int = atoi(setpoint_fractional_part);
    setpoint_f_num = setpoint_integer_part_int;
    setpoint_f_num += setpoint_fractional_part_int / 100.00;
    return setpoint_f_num;
}
float convert_to_float(uint8_t *rx_buffer, int choice) {
    int start = -1;
    switch (choice) {
        case 1:
            start = 1;
            break;
        case 2:
            start = 9;
            break;
        case 3:

```

```

        start = 17;
        break;
    case 4:
        start = 25;
        break;
    default:
        fprintf(stderr,
"ERROR @ convert_to_float, choice must be 1<=choice<=4.\n");
        return -1;
    }
    float f_num = 0.0;
    char integer_part[] = { rx_buffer[start], rx_buffer[start + 1],
        rx_buffer[start + 2] };
    char fractional_part[] = { rx_buffer[start + 4], rx_buffer[start + 5] };
    int integer_part_int = atoi(integer_part);
    int fractional_part_int = atoi(fractional_part);
    f_num = integer_part_int;
    f_num += fractional_part_int / 100.00;
    return f_num;
} /* USER CODE END 0 */

void go_to_side1() {
    //.....START ROUTINE.....
    printf("\n\rReset valve..... \n\r");
    printf("\n\rmeter_value is: %.2f \n\r", meter_value);
    //Find the time it takes to close
    position_time = y_0 + A1 * exp((meter_value - x0) / t1)
        + A2 * exp((meter_value - x0) / t2);

    //.....THE VALVE OPENS.....
    //SET ON RELAY THAT OPENS VALVE (Blue cable)
    HAL_GPIO_WritePin(GPIOA, GPIO_PIN_11, GPIO_PIN_SET);
    HAL_Delay(16000 - position_time); //Delay
    //SET OFF RELAY THAT OPENS VALVE (Blue cable)
    HAL_GPIO_WritePin(GPIOA, GPIO_PIN_11, GPIO_PIN_RESET);
    HAL_Delay(500); //Delay between change

    //.....THE VALVE CLOSES.....
    //SET ON RELAY THAT CLOSE VALVE(Grey cable)
    HAL_GPIO_WritePin(GPIOA, GPIO_PIN_12, GPIO_PIN_SET);
    HAL_Delay(16000); //Delay 16 sec
    //SET OFF RELAY THAT CLOSE VALVE(Grey cable)
    HAL_GPIO_WritePin(GPIOA, GPIO_PIN_12, GPIO_PIN_RESET);
    HAL_Delay(500); //Delay between change

    //.....THE VALVE OPENS.....
    //SET ON RELAY THAT OPENS VALVE (Blue cable)
    HAL_GPIO_WritePin(GPIOA, GPIO_PIN_11, GPIO_PIN_SET);
    HAL_Delay(2370); //Delay
    //SET OFF RELAY THAT OPENS VALVE (Blue cable)
    HAL_GPIO_WritePin(GPIOA, GPIO_PIN_11, GPIO_PIN_RESET);
    HAL_Delay(3000); //Delay between change
    meter_value = convert_to_float(rx_buffer, 4);
    printf("\n\rmeter_value is: %.2f \n\r", meter_value);

    //Find side 1
    if (meter_value >= 00.12)//then its side 2
        { //so do a cycle and go to side 1

    //.....THE VALVE OPENS.....
        //SET ON RELAY THAT OPENS VALVE (Blue cable)
        HAL_GPIO_WritePin(GPIOA, GPIO_PIN_11, GPIO_PIN_SET);

```

```

HAL_Delay(16000); //Delay
//SET OFF RELAY THAT OPENS VALVE (Blue cable)
HAL_GPIO_WritePin(GPIOA, GPIO_PIN_11, GPIO_PIN_RESET);
HAL_Delay(500); //Delay between change

//.....THE VALVE CLOSES.....
//SET ON RELAY THAT CLOSE VALVE(Grey cable)
HAL_GPIO_WritePin(GPIOA, GPIO_PIN_12, GPIO_PIN_SET);
HAL_Delay(16000); //Delay 16 sec
//SET OFF RELAY THAT CLOSE VALVE(Grey cable)
HAL_GPIO_WritePin(GPIOA, GPIO_PIN_12, GPIO_PIN_RESET);
HAL_Delay(500); //Delay between change
} else {
//do a reset to go back to side 1 and start from the beginning(2200 ms)
//.....THE VALVE OPENS.....
//SET ON RELAY THAT OPENS VALVE (Blue cable)
HAL_GPIO_WritePin(GPIOA, GPIO_PIN_11, GPIO_PIN_SET);
HAL_Delay(16000); //Delay
//SET OFF RELAY THAT OPENS VALVE (Blue cable)
HAL_GPIO_WritePin(GPIOA, GPIO_PIN_11, GPIO_PIN_RESET);
HAL_Delay(500); //Delay between change

//.....THE VALVE CLOSES.....
//SET ON RELAY THAT CLOSE VALVE(Grey cable)
HAL_GPIO_WritePin(GPIOA, GPIO_PIN_12, GPIO_PIN_SET);
HAL_Delay(16000); //Delay 16 sec
//SET OFF RELAY THAT CLOSE VALVE(Grey cable)
HAL_GPIO_WritePin(GPIOA, GPIO_PIN_12, GPIO_PIN_RESET);
HAL_Delay(500); //Delay between change

//.....THE VALVE OPENS.....
//SET ON RELAY THAT OPENS VALVE (Blue cable)
HAL_GPIO_WritePin(GPIOA, GPIO_PIN_11, GPIO_PIN_SET);
HAL_Delay(16000); //Delay
//SET OFF RELAY THAT OPENS VALVE (Blue cable)
HAL_GPIO_WritePin(GPIOA, GPIO_PIN_11, GPIO_PIN_RESET);
HAL_Delay(500); //Delay between change

//.....THE VALVE CLOSES.....
//SET ON RELAY THAT CLOSE VALVE(Grey cable)
HAL_GPIO_WritePin(GPIOA, GPIO_PIN_12, GPIO_PIN_SET);
HAL_Delay(16000); //Delay 16 sec
//SET OFF RELAY THAT CLOSE VALVE(Grey cable)
HAL_GPIO_WritePin(GPIOA, GPIO_PIN_12, GPIO_PIN_RESET);
HAL_Delay(500); //Delay between change
}
}
int main(void) {
//Reset of all peripherals,Initializes the Flash interface and the Systick
HAL_Init();
/* Configure the system clock */
SystemClock_Config();
/* Initialize all configured peripherals */
MX_GPIO_Init();
MX_USART1_UART_Init();
MX_USART2_UART_Init();
/* USER CODE BEGIN 2 */
HAL_UART_Receive_IT(&huart1, rx_buffer, sizeof(rx_buffer));
meter_value = convert_to_float(rx_buffer, 4);
HAL_Delay(5000);/* USER CODE END 2 */

go_to_side1();
//Electric valve is ready to start

```

```

printf("\n\rValve is ready!\n\r");

HAL_UART_Transmit(&huart2, tx_buffer, sizeof(tx_buffer), HAL_MAX_DELAY);
HAL_UART_Receive(&huart2, set_point, sizeof(set_point), HAL_MAX_DELAY);
set_point_f = setpoint_convert_to_float(set_point);

printf("\n\rsetpoint: %.2f\n\r", set_point_f);
printf("\n\rmeter value is: %.2f\n\r", meter_value);

if (set_point_f > 25) {
    printf("\n\r *****error value, choice must be 0<=choice<=25.");
}

if (25 >= set_point_f && set_point_f >= 02.00) {

    time = y_0 + A1 * exp(((set_point_f - 02.00) - x0) / t1)
            + A2 * exp(((set_point_f - 02.00) - x0) / t2);

    printf("\n\r time is: %f\n\r", time);

    //.....THE VALVE OPENS.....
    //SET ON RELAY THAT OPENS VALVE (Blue cable)
    HAL_GPIO_WritePin(GPIOA, GPIO_PIN_11, GPIO_PIN_SET);
    HAL_Delay(time); //Delay
    //SET OFF RELAY THAT OPENS VALVE (Blue cable)
    HAL_GPIO_WritePin(GPIOA, GPIO_PIN_11, GPIO_PIN_RESET);
    HAL_Delay(300); //Delay between change
    printf("\n\rmeter value: %.2f\n\r", meter_value);

    do {
        //.....THE VALVE OPENS.....

        //SET ON RELAY THAT OPENS VALVE (Blue cable)
        HAL_GPIO_WritePin(GPIOA, GPIO_PIN_11, GPIO_PIN_SET);
        HAL_Delay(25); //Delay
        //SET OFF RELAY THAT OPENS VALVE (Blue cable)
        HAL_GPIO_WritePin(GPIOA, GPIO_PIN_11, GPIO_PIN_RESET);
        HAL_Delay(300); //Delay between change
        printf("\n\rmeter value: %.2f\n\r", meter_value);

    } while ((set_point_f - SYSTEM_DEVIATION) >= meter_value);

}
if (set_point_f < 02.00) {

    //SET ON RELAY THAT OPENS VALVE (Blue cable)
    HAL_GPIO_WritePin(GPIOA, GPIO_PIN_11, GPIO_PIN_SET);
    HAL_Delay(2200); //Delay
    //SET OFF RELAY THAT OPENS VALVE (Blue cable)
    HAL_GPIO_WritePin(GPIOA, GPIO_PIN_11, GPIO_PIN_RESET);
    HAL_Delay(500); //Delay between change

    do {
        //.....THE VALVE OPENS.....
        //SET ON RELAY THAT OPENS VALVE (Blue cable)
        HAL_GPIO_WritePin(GPIOA, GPIO_PIN_11, GPIO_PIN_SET);
        HAL_Delay(25); //Delay
        //SET OFF RELAY THAT OPENS VALVE (Blue cable)
        HAL_GPIO_WritePin(GPIOA, GPIO_PIN_11, GPIO_PIN_RESET);
        HAL_Delay(300); //Delay between change
        printf("\n\rmeter value: %.2f\n\r", meter_value);
    } while ((set_point_f - SYSTEM_DEVIATION) >= meter_value);
}

```

```

}

HAL_Delay(4000); //Delay between change
printf("\\n\\rmeter value: %.2f\\n\\r", meter_value);

while (1) { /* USER CODE BEGIN WHILE */

HAL_UART_Transmit(&huart2, tx_buffer, sizeof(tx_buffer), HAL_MAX_DELAY);
HAL_UART_Receive(&huart2, set_point, sizeof(set_point), HAL_MAX_DELAY);
set_point_f = setpoint_convert_to_float(set_point);

if (set_point_f <= 25) {

    printf("\\n\\rsetpoint: %.2f \\n\\r", set_point_f);
    printf("\\n\\rmeter value: %.2f\\n\\r", meter_value);

    time = y_0 + A1 * exp((set_point_f - 02.00) - x0) / t1)
        + A2 * exp((set_point_f - 02.00) - x0) / t2);
    printf("\\n\\r time is: %f \\n\\r", time);
    position_time = y_0 + A1 * exp((meter_value - x0) / t1)
        + A2 * exp((meter_value - x0) / t2);
    printf("\\n\\r position time is: %f \\n\\r", position_time);
    printf("\\n\\r Sub is: %f \\n\\r", time - position_time);
    printf("\\n\\r*****wait the valve is moving\\n\\r*****");

do {
//-----
if (set_point_f > meter_value) {
//-----

if (set_point_f - meter_value >= 02.00) {

    //.....THE VALVE OPENS.....
    //SET ON RELAY THAT OPENS VALVE (Blue cable)
    HAL_GPIO_WritePin(GPIOA, GPIO_PIN_11, GPIO_PIN_SET);
    HAL_Delay(time - position_time); //Delay -position_time
    //SET OFF RELAY THAT OPENS VALVE (Blue cable)
    HAL_GPIO_WritePin(GPIOA, GPIO_PIN_11, GPIO_PIN_RESET);
    HAL_Delay(300); //Delay between change
    printf("\\n\\rmeter value: %.2f\\n\\r", meter_value);

do {
    //.....THE VALVE OPENS.....
    //SET ON RELAY THAT OPENS VALVE (Blue cable)
    HAL_GPIO_WritePin(GPIOA, GPIO_PIN_11, GPIO_PIN_SET);
    HAL_Delay(25); //Delay
    //SET OFF RELAY THAT OPENS VALVE (Blue cable)
    HAL_GPIO_WritePin(GPIOA, GPIO_PIN_11,
        GPIO_PIN_RESET);
    HAL_Delay(300); //Delay between change
    printf("\\n\\rmeter value: %.2f\\n\\r", meter_value);
    } while ((set_point_f - SYSTEM_DEVIATION) >= meter_value);
    }

else {

do {
    //.....THE VALVE OPENS.....
    //SET ON RELAY THAT OPENS VALVE (Blue cable)
    HAL_GPIO_WritePin(GPIOA, GPIO_PIN_11, GPIO_PIN_SET);
    HAL_Delay(25); //Delay
    //SET OFF RELAY THAT OPENS VALVE (Blue cable)
    HAL_GPIO_WritePin(GPIOA, GPIO_PIN_11, GPIO_PIN_RESET);

```

```

        HAL_Delay(300); //Delay between change
        printf("\n\rmeter valuet: %.2f\n\r", meter_value);
    } while ((set_point_f - SYSTEM_DEVIATION) >= meter_value);

}

}

//-----
else
//-----
{
//-----Reset valve to side 1-----
position_time = y_0 + A1 * exp((meter_value - x0) / t1)
                + A2 * exp((meter_value - x0) / t2);

//.....THE VALVE OPENS.....
//SET ON RELAY THAT OPENS VALVE (Blue cable)
HAL_GPIO_WritePin(GPIOA, GPIO_PIN_11, GPIO_PIN_SET);
HAL_Delay(16000 - position_time); //Delay
//SET OFF RELAY THAT OPENS VALVE (Blue cable)
HAL_GPIO_WritePin(GPIOA, GPIO_PIN_11, GPIO_PIN_RESET);
HAL_Delay(500); //Delay between change

//.....THE VALVE CLOSES.....
//SET ON RELAY THAT CLOSE VALVE(Grey cable)
HAL_GPIO_WritePin(GPIOA, GPIO_PIN_12, GPIO_PIN_SET);
HAL_Delay(16000); //Delay 16 sec
//SET OFF RELAY THAT CLOSE VALVE(Grey cable)
HAL_GPIO_WritePin(GPIOA, GPIO_PIN_12, GPIO_PIN_RESET);
HAL_Delay(500); //Delay between change

//.....THE VALVE OPENS.....
//SET ON RELAY THAT OPENS VALVE (Blue cable)
HAL_GPIO_WritePin(GPIOA, GPIO_PIN_11, GPIO_PIN_SET);
HAL_Delay(16000); //Delay
//SET OFF RELAY THAT OPENS VALVE (Blue cable)
HAL_GPIO_WritePin(GPIOA, GPIO_PIN_11, GPIO_PIN_RESET);
HAL_Delay(500); //Delay between change

//.....THE VALVE CLOSES.....
//SET ON RELAY THAT CLOSE VALVE(Grey cable)
HAL_GPIO_WritePin(GPIOA, GPIO_PIN_12, GPIO_PIN_SET);
HAL_Delay(16000); //Delay 16 sec
//SET OFF RELAY THAT CLOSE VALVE(Grey cable)
HAL_GPIO_WritePin(GPIOA, GPIO_PIN_12, GPIO_PIN_RESET);
HAL_Delay(500); //Delay between change

//-----

if (set_point_f >= 02.00) {
//.....THE VALVE OPENS.....
//SET ON RELAY THAT OPENS VALVE (Blue cable)
HAL_GPIO_WritePin(GPIOA, GPIO_PIN_11, GPIO_PIN_SET);
HAL_Delay(time); //Delay
//SET OFF RELAY THAT OPENS VALVE (Blue cable)
HAL_GPIO_WritePin(GPIOA, GPIO_PIN_11, GPIO_PIN_RESET);
HAL_Delay(300); //Delay between change
printf("\n\rmeter valuet: %.2f\n\r", meter_value);

do {

//.....THE VALVE OPENS.....

```

```

//SET ON RELAY THAT OPENS VALVE (Blue cable)
HAL_GPIO_WritePin(GPIOA, GPIO_PIN_11, GPIO_PIN_SET);
HAL_Delay(25); //Delay
//SET OFF RELAY THAT OPENS VALVE (Blue cable)
HAL_GPIO_WritePin(GPIOA, GPIO_PIN_11,
                    GPIO_PIN_RESET);
HAL_Delay(300); //Delay between change
printf("n\rmeter valuet: %.2f\n\r", meter_value);
} while ((set_point_f - SYSTEM_DEVIATION) >= meter_value);

} else {
//SET ON RELAY THAT OPENS VALVE (Blue cable)
HAL_GPIO_WritePin(GPIOA, GPIO_PIN_11, GPIO_PIN_SET);
HAL_Delay(2200); //Delay
//SET OFF RELAY THAT OPENS VALVE (Blue cable)
HAL_GPIO_WritePin(GPIOA, GPIO_PIN_11, GPIO_PIN_RESET);
HAL_Delay(300); //Delay between change

do {

//.....THE VALVE OPENS.....
//SET ON RELAY THAT OPENS VALVE (Blue cable)
HAL_GPIO_WritePin(GPIOA, GPIO_PIN_11, GPIO_PIN_SET);
HAL_Delay(25); //Delay
//SET OFF RELAY THAT OPENS VALVE (Blue cable)
HAL_GPIO_WritePin(GPIOA, GPIO_PIN_11,
                    GPIO_PIN_RESET);
HAL_Delay(300); //Delay between change
printf("n\rmeter valuet: %.2f\n\r", meter_value);
} while ((set_point_f - SYSTEM_DEVIATION) >= meter_value);

}

}
HAL_Delay(4000); //Delay between change
printf("n\rmeter valuet: %.2f\n\r", meter_value);

} while ((set_point_f + SYSTEM_DEVIATION) <= meter_value
        && (set_point_f - SYSTEM_DEVIATION) >= meter_value);
}

else {
printf("n\r*****error value, choice must be 0<=choice<=25.");
}

}

/* USER CODE END WHILE */

}

void SystemClock_Config(void) {
RCC_OscInitTypeDef RCC_OscInitStruct = { 0 };
RCC_ClkInitTypeDef RCC_ClkInitStruct = { 0 };
//Configure the main internal regulator output voltage
__HAL_RCC_PWR_CLK_ENABLE();
__HAL_PWR_VOLTAGESCALING_CONFIG(PWR_REGULATOR_VOLTAGE_SCALE2);
//Initializes the RCC Oscillators according to the specified parameters
//in the RCC_OscInitTypeDef structure.
RCC_OscInitStruct.OscillatorType = RCC_OSCILLATORTYPE_HSI;
RCC_OscInitStruct.HSIState = RCC_HSI_ON;
RCC_OscInitStruct.HSICalibrationValue = RCC_HSICALIBRATION_DEFAULT;
RCC_OscInitStruct.PLL.PLLState = RCC_PLL_ON;
RCC_OscInitStruct.PLL.PLLSource = RCC_PLLSOURCE_HSI;
RCC_OscInitStruct.PLL.PLLM = 8;

```

```

RCC_OscInitStruct.PLL.PLLN = 84;
RCC_OscInitStruct.PLL.PLLP = RCC_PLLP_DIV2;
RCC_OscInitStruct.PLL.PLLQ = 4;
if (HAL_RCC_OscConfig(&RCC_OscInitStruct) != HAL_OK) {
    Error_Handler();
}
//Initializes the CPU, AHB and APB buses clocks
RCC_ClkInitStruct.ClockType = RCC_CLOCKTYPE_HCLK | RCC_CLOCKTYPE_SYSCLK
    | RCC_CLOCKTYPE_PCLK1 | RCC_CLOCKTYPE_PCLK2;
RCC_ClkInitStruct.SYSCLKSource = RCC_SYSCLKSOURCE_PLLCLK;
RCC_ClkInitStruct.AHBCLKDivider = RCC_SYSCLK_DIV1;
RCC_ClkInitStruct.APB1CLKDivider = RCC_HCLK_DIV2;
RCC_ClkInitStruct.APB2CLKDivider = RCC_HCLK_DIV1;
if (HAL_RCC_ClockConfig(&RCC_ClkInitStruct, FLASH_LATENCY_2) != HAL_OK) {
    Error_Handler();
}
}
/* USER CODE BEGIN 4 */
// Function to flush unwanted data
void flushUartBuffer(UART_HandleTypeDef *huart) {
}
//Keep reading bytes until the buffer is empty or a specific conditionismet
void HAL_UART_RxCpltCallback(UART_HandleTypeDef *huart) {
    meter_value = convert_to_float(rx_buffer, 4);
    uint8_t temp;

    //Synchronizes the MFM message with the rx_buffer
    while (1) {
        HAL_UART_Receive(&huart1, &temp, 1, HAL_MAX_DELAY);
        //printf("%c", temp);
        //fflush(stdout);
// Check for a marker or delimiter to indicate the start of a new message
        if (temp == 13) {
            //printf("\n\r");
            //fflush(stdout);
// Break if the marker is found to synchronize with the start of a new message
            break;
        }
    }
    HAL_UART_Receive_IT(&huart1, rx_buffer, sizeof(rx_buffer));
}
/* USER CODE END 4 */
//@brief This function is executed in case of error occurrence.
//@retval None
void Error_Handler(void) {
    __disable_irq();
    while (1) {
    }
}
#ifdef USE_FULL_ASSERT
void assert_failed(uint8_t *file, uint32_t line){}
#endif /* USE_FULL_ASSERT */

```