

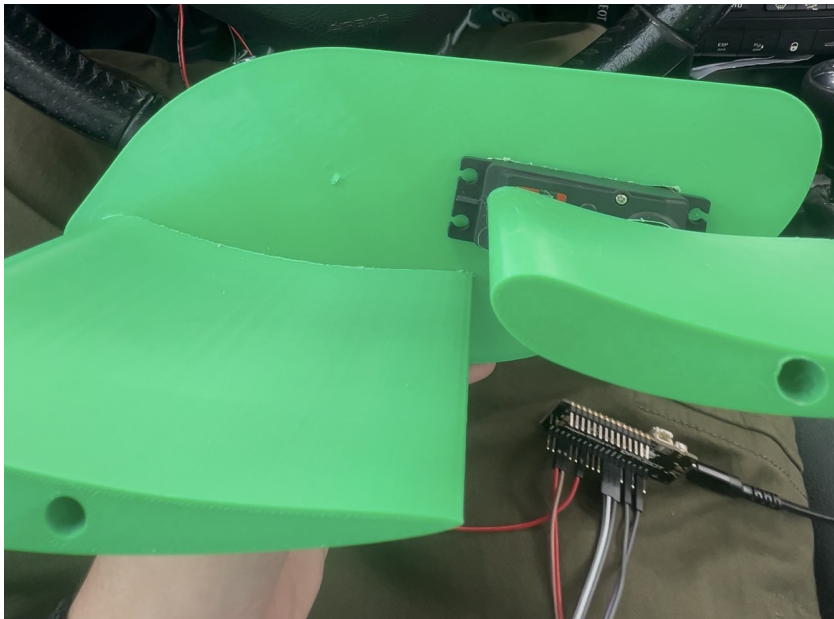


ΔΙΕΘΝΕΣ
ΠΑΝΕΠΙΣΤΗΜΙΟ
ΤΗΣ ΕΛΛΑΔΟΣ

ΠΟΛΥΤΕΧΝΙΚΗ ΣΧΟΛΗ
ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ
ΚΑΙ ΗΛΕΚΤΡΟΝΙΚΩΝ ΣΥΣΤΗΜΑΤΩΝ

ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ

«Ανάπτυξη Εφαρμοσμένου Συστήματος Ελέγχου
Ενεργών Αεροδυναμικών μερών μέσω CAN bus»



Του φοιτητή
Τραντούδη Αχιλλέα
Αρ. Μητρώου: 516135

Επιβλέπων
Ιορδάνης Κιοσκερίδης
Καθηγητής

Ιούνιος 2026

Κωδικός Δ.Ε. 26130
Ονοματεπώνυμο φοιτητή: Τραντούδης Αχιλλέας
Ονοματεπώνυμο εισηγητή: Ιορδάνης Κιοσκερίδης
Ημερομηνία ανάληψης Π.Ε.: 25/2/2026
Ημερομηνία περάτωσης Π.Ε.: 31/5/2026

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

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

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

Πρόλογος

Το αυτοκίνητο αποτέλεσε ανέκαθεν πηγή έμπνευσης και πάθους. Η διαπίστωση ότι τα σύγχρονα εμπορικά οχήματα αφήνουν ελάχιστα περιθώρια παρέμβασης στον χρήστη — με συστήματα όπως η ενεργή αεροδυναμική να παραμένουν προνόμιο λίγων και εξαιρετικά δαπανηρών οχημάτων — αποτέλεσε το κύριο κίνητρο για την εκπόνηση της παρούσας εργασίας.

Στόχος ήταν να αποδειχθεί ότι με προσιτό εξοπλισμό και ανοιχτού κώδικα εργαλεία είναι δυνατή η ανάπτυξη συστήματος ενεργής αεροδυναμικής που αξιοποιεί τα δεδομένα του ίδιου του οχήματος.

Θα ήθελα να ευχαριστήσω τον επιβλέποντα καθηγητή μου για την καθοδήγηση και υποστήριξή του, καθώς και όλους όσους συνέβαλαν στην ολοκλήρωση της εργασίας.

Θεσσαλονίκη, 2026

Περίληψη

Η παρούσα εργασία πραγματεύεται την ανάπτυξη συστήματος ελέγχου ενεργής αεροτομής μέσω ανάγνωσης δεδομένων από το δίκτυο CAN bus εμπορικού επιβατικού οχήματος. Στόχος της εργασίας είναι η αποκωδικοποίηση των ιδιόκτητων μηνυμάτων του κατασκευαστή που σχετίζονται με τη θέση του πεντάλ γκαζιού και την κατάσταση του φρένου, και η αξιοποίησή τους για τον έλεγχο μηχανισμού ενεργής αεροδυναμικής σε πραγματικό χρόνο.

Αρχικά πραγματοποιήθηκε θεωρητική ανάλυση του πρωτοκόλλου CAN bus, καλύπτοντας τη δομή των frames, τον μηχανισμό arbitration, την ανίχνευση σφαλμάτων και τη διεπαφή OBD-II. Στη συνέχεια παρουσιάστηκαν τα εξαρτήματα και τα εργαλεία λογισμικού που χρησιμοποιήθηκαν, συμπεριλαμβανομένων του CAN controller MCP2515, του μικροελεγκτή ESP32 FireBeetle και των βιβλιοθηκών ανάπτυξης.

Η διαδικασία υλοποίησης ξεκίνησε με αρχική εξερεύνηση μέσω ELM327, η οποία κατέδειξε τους περιορισμούς του τυποποιημένου OBD-II. Αναπτύχθηκε μεθοδολογία συστηματικής αποκωδικοποίησης που συνδύαζε καταγραφή frames με MCP2515, εξαγωγή δεδομένων σε Microsoft Excel και οπτικοποίηση μεταβολών μέσω γραφικού περιβάλλοντος Tkinter με σύστημα οπτικής σήμανσης. Μέσω πειραματικής επαλήθευσης υπό πραγματικές συνθήκες οδήγησης εντοπίστηκαν επιτυχώς τα CAN frames που αντιστοιχούν στο γκάζι και το φρένο.

Το σύστημα ελέγχου υλοποιήθηκε με ESP32 FireBeetle και MCP2515, επιτυγχάνοντας ανάγνωση CAN frames και έλεγχο servo κινητήρων αεροτομής σε πραγματικό χρόνο με γωνία κίνησης 0°-30°. Η αεροτομή κατασκευάστηκε σε μειωμένη κλίμακα με τεχνολογία τρισδιάστατης εκτύπωσης για την επαλήθευση της λειτουργίας του συστήματος.

Τα αποτελέσματα της εργασίας αποδεικνύουν ότι η ανάπτυξη συστήματος ενεργής αεροδυναμικής βασισμένου σε δεδομένα CAN bus είναι τεχνικά εφικτή με προσιτό εξοπλισμό και ανοιχτού κώδικα εργαλεία, θέτοντας τα θεμέλια για μελλοντική υλοποίηση σε πραγματική κλίμακα.

«CAN Bus-Based Embedded Control System for Active Aerodynamic Components »

Trantoudes Achilleas

Abstract

This thesis presents the development of an active aerodynamic wing control system through real-time data acquisition from the CAN bus network of a commercial passenger vehicle. The primary objective of this work is the decoding of manufacturer-proprietary messages related to the accelerator pedal position and brake status, and their utilization for the real-time control of an active aerodynamic mechanism.

Initially, a theoretical analysis of the CAN bus protocol is presented, covering frame structure, arbitration mechanism, error detection and management, and the OBD-II interface. Subsequently, the hardware components and software tools employed are described, including the MCP2515 CAN controller, the ESP32 FireBeetle microcontroller, and the associated development libraries.

The implementation process began with an initial exploration using an ELM327 diagnostic adapter, which revealed the limitations of the standardized OBD-II protocol regarding access to manufacturer-specific messages. A systematic decoding methodology was developed, combining CAN frame logging with the MCP2515, data export to Microsoft Excel for analysis, and real-time visualization of byte changes through a custom Tkinter graphical interface with a visual flag system. Through experimental verification under real driving conditions, the CAN frames corresponding to the accelerator pedal and brake were successfully identified.

The control system was implemented using the ESP32 FireBeetle and MCP2515, achieving real-time CAN frame reading and servo motor control with an angular range of 0° - 30° . Due to cost and time constraints, the aerodynamic wing was manufactured at reduced scale using FDM 3D printing technology with PETG material, serving as a proof of concept for the control system.

The results of this work demonstrate that the development of a CAN bus-based active aerodynamic system is technically feasible using affordable hardware and open-source tools, establishing the foundation for a future full-scale implementation.

Ευχαριστίες

Θα ήθελα να εκφράσω τις ειλικρινείς μου ευχαριστίες στον επιβλέποντα καθηγητή μου κ. Κιοσκερίδη, για την πολύτιμη βοήθεια και την εμπιστοσύνη που μου έδειξε καθ' όλη τη διάρκεια εκπόνησης της παρούσας πτυχιακής εργασίας. Η καθοδήγηση και η υποστήριξή του υπήρξαν καθοριστικές για την ολοκλήρωσή της.

Επίσης, θα ήθελα να ευχαριστήσω το σύνολο των καθηγητών του τμήματος για τη στήριξη και τις γνώσεις που μου μετέδωσαν κατά τη διάρκεια των σπουδών μου.

Περιεχόμενα

Πρόλογος.....	5
Περίληψη.....	6
Abstract.....	7
Ευχαριστίες.....	9
Κατάλογος Σχημάτων.....	14
Κατάλογος Πινάκων.....	14
Κεφάλαιο 1ο: Εισαγωγή.....	1
1.1 Αεροδυναμική οχημάτων.....	1
1.2 Ενεργά αεροδυναμικά συστήματα.....	1
1.3 Εφαρμογές ενεργών αεροδυναμικών συστημάτων.....	2
1.4 Προβλήματα που αντιμετωπίζουν τα συμβατικά οχήματα.....	8
1.5 Σκοπός της εργασίας.....	8
Κεφάλαιο 2ο: Πρωτόκολλο CAN bus.....	9
2.1 Εισαγωγή.....	9
2.2 Βασική αρχή λειτουργίας του CAN Bus.....	9
2.2.1 Dominant και Recessive Bits.....	9
2.2.2 Τοπολογία δικτύου και τερματισμός.....	10
2.2.3 Ταχύτητα επικοινωνίας CAN Bus.....	10
2.2.4 Δομή ενός CAN Frame.....	11
2.3 Μηχανισμός Arbitration.....	13
2.4 Ανίχνευση και Διαχείριση Σφαλμάτων.....	13
2.5 Διεπαφή OBD-II και σύνδεση με το CAN Bus.....	14
2.6 Συμπέρασμα.....	15
Κεφάλαιο 3ο: Εξολισμός Εργασίας.....	16
3.1 Παρουσίαση Εξοπλισμού και Κριτήρια Επιλογής.....	16
3.2 Διαγνωστικό ELM327.....	17
3.2.1 Γενική περιγραφή.....	17
3.2.2 Αρχιτεκτονική και Τεχνικά Χαρακτηριστικά.....	18
3.2.3 Υποστηριζόμενα Πρωτόκολλα OBD-II.....	19
3.3 MCP2515.....	21
3.3.1 Αρχιτεκτονική και λειτουργία.....	22
3.3.2 Transceiver TJ1050.....	22
3.3.3 Ακροδέκτες (Pins).....	22
3.3.4 Διεπαφή SPI.....	23

3.3.5 Φίλτρα και μάσκες υλικού.....	24
3.3.6 Περιορισμοί.....	25
3.4 Arduino Uno Minima R4.....	25
3.4.1 Τεχνικά χαρακτηριστικά.....	26
3.4.2 Το πρωτόκολλο SPI στο Arduino UNO R4 Minima.....	27
3.5 Firebeetle 2 ESP32-E.....	28
3.5.1 Τεχνικά Χαρακτηριστικά.....	28
3.5.2 Πρωτόκολλα ασύρματης επικοινωνίας.....	29
3.5.3 Πρωτόκολλο SPI.....	30
3.5.4 Two-Wire Automotive Interface.....	30
3.6 CAN Transceiver SN65HVD230.....	31
3.7 Servo Motor DS5180.....	32
3.8 Step-Down Converter LM2596.....	33
Κεφάλαιο 4ο: Λογισμικό Εργασίας.....	36
4.1 Εισαγωγή.....	36
4.2 Arduino IDE.....	36
4.2.1 Χαρακτηριστικά και δυνατότητες.....	36
4.2.2 Χρήση στο Παρόν Σύστημα.....	37
4.2.3 Περιορισμοί.....	37
4.3 Visual Studio Code.....	37
4.3.1 Αρχιτεκτονική.....	37
4.3.2 Βασικά χαρακτηριστικά.....	37
4.3.3 Επιλογή για το παρόν Σύστημα.....	38
4.4 PlatformIO.....	38
4.4.1 Αρχιτεκτονική και Λειτουργία.....	39
4.4.2 Βασικά Χαρακτηριστικά.....	39
4.4.3 Σύγκριση με Arduino IDE.....	40
4.4.4 Επιλογή για το Παρόν Σύστημα.....	40
4.5 Python.....	40
4.5.1 Βασικά Χαρακτηριστικά.....	40
4.5.2 PySerial.....	41
4.5.3 Tkinter.....	41
4.5.4 Επιλογή Python για το Παρόν Σύστημα.....	42
4.6 Βιβλιοθήκες Embedded.....	42
4.6.1 MCP_CAN (coryjfowler).....	42

4.6.2 Βιβλιοθήκη SPI.....	43
4.6.3 Βιβλιοθήκη ESP32Servo.....	43
4.6.4 ESP32 CAN / TWAI.....	44
4.7 Συμπεράσματα.....	45
Κεφάλαιο 5ο: Υλοποίηση και Διαδικασία Ανίχνευσης.....	47
5.1 Αρχική προσέγγιση.....	47
5.2 Αρχική Εξερεύνηση με ELM327.....	47
5.2.1 Σύνδεση και Λογισμικό.....	47
5.2.2 Διαθέσιμοι Αισθητήρες.....	48
5.2.3 Περιορισμοί και Απόφαση Μετάβασης στο MCP2515.....	49
5.3 Καταγραφή CAN Frames με MCP2515 + Arduino.....	49
5.3.1 Συνδεσμολογία.....	49
5.3.2 Βιβλιοθήκη MCP_CAN και Αρχικός Κώδικας.....	51
5.3.3 Εντοπισμός Καθυστέρησης και Βελτιστοποίηση.....	51
5.3.4 Μεταφορά Επεξεργασίας σε Python/PySerial.....	52
5.4 Συλλογή Δεδομένων.....	52
5.4.1 Μεθοδολογία Συλλογής.....	52
5.4.2 Μέτρηση με Σβηστή Μηχανή.....	52
5.4.3 Μέτρηση στα 50 km/h.....	52
5.4.4 Μέτρηση στα 70 km/h.....	53
5.4.5 Εξαγωγή Δεδομένων σε .txt.....	53
5.5 Ανάλυση και Αποκωδικοποίηση CAN Frames.....	53
5.5.1 Ανάλυση σε Excel.....	53
5.5.2 Οπτικοποίηση με Tkinter — Flag System.....	55
5.5.3 Απομόνωση και Επαλήθευση Κωδικών.....	56
5.5.4 Εντοπισμός της Χρήσιμης Πληροφορίας.....	56
5.5.5 Γωνία Τιμονιού — Γιατί Δεν Εντοπίστηκε.....	57
5.6 Υλοποίηση Συστήματος Ελέγχου.....	58
5.6.1 Μετάβαση από Arduino σε ESP32 FireBeetle.....	58
5.6.2 Συνδεσμολογία ESP32 + MCP2515.....	58
5.6.3 Εφαρμογή Φίλτρων και Λογική Ελέγχου.....	59
5.6.4 Τροφοδοσία Συστήματος — LM2596.....	59
5.7 Αεροτομή και Μηχανισμός Κίνησης.....	60
5.7.1 Υπολογισμός Δύναμης και Επιλογή Servo.....	60
5.7.2 Σχεδίαση Αεροτομής.....	60

5.7.3 Κατασκευή με 3D Εκτύπωση.....	61
5.7.4 Γωνία Λειτουργίας και Έλεγχος Κίνησης.....	62
Κεφάλαιο 6ο: Συμπεράσματα και Μελλοντικές Βελτιώσεις.....	64
6.1 Συμπεράσματα.....	64
6.1.1 Συμπεράσματα ως προς την Αποκωδικοποίηση CAN Bus.....	64
6.1.2 Συμπεράσματα ως προς το Σύστημα Ελέγχου.....	64
6.1.3 Γενικά Συμπεράσματα.....	64
6.2 Μελλοντικές Βελτιώσεις.....	65
6.2.1 Εφαρμογή Αεροτομής Πραγματικής Κλίμακας.....	65
6.2.2 Ασύρματη Τηλεμετρία μέσω Wi-Fi.....	65
6.2.3 Σχεδίαση Ολοκληρωμένου Κυκλώματος.....	66
6.2.4 Native CAN Αρχιτεκτονική.....	66
6.2.5 Εντοπισμός Γωνίας Τιμονιού.....	66
6.3 Επίλογος.....	67
ΒΙΒΛΙΟΓΡΑΦΙΑ.....	68

Κατάλογος Σχημάτων

Σχήμα 1.1.....	3
Σχήμα 1.2.....	3
Σχήμα 1.3.....	4
Σχήμα 1.4.....	5
Σχήμα 1.5.....	5
Σχήμα 1.6.....	6
Σχήμα 1.7.....	7
Σχήμα 1.8.....	8
Σχήμα 2.1.....	13
Σχήμα 2.2.....	16
Σχήμα 3.1.....	19
Σχήμα 3.2.....	21
Σχήμα 3.3.....	23
Σχήμα 3.4.....	28
Σχήμα 3.5.....	29
Σχήμα 3.6.....	30
Σχήμα 3.7.....	31
Σχήμα 3.8.....	33
Σχήμα 3.9.....	35
Σχήμα 3.10.....	36
Σχήμα 3.11.....	36
Σχήμα 5.1.....	49
Σχήμα 5.2.....	50
Σχήμα 5.3.....	51
Σχήμα 5.4.....	52
Σχήμα 5.5.....	55
Σχήμα 5.6.....	55
Σχήμα 5.7.....	56
Σχήμα 5.8.....	57
Σχήμα 5.9.....	58
Σχήμα 5.10.....	62
Σχήμα 5.11.....	63

Κατάλογος Πινάκων

Πίνακας 3.1.....	16
Πίνακας 3.2.....	16
Πίνακας 3.3.....	22
Πίνακας 3.4.....	30
Πίνακας 3.5.....	30
Πίνακας 3.6.....	32
Πίνακας 4.1.....	38
Πίνακας 4.2.....	39
Πίνακας 4.3.....	40
Πίνακας 4.4.....	40
Πίνακας 4.5.....	43
Πίνακας 4.6.....	44
Πίνακας 4.7.....	45
Πίνακας 4.8.....	45
Πίνακας 5.1.....	48
Πίνακας 5.2.....	50
Πίνακας 5.3.....	58
Πίνακας 5.4.....	59
Πίνακας 5.5.....	62

Κεφάλαιο 1ο: Εισαγωγή

1.1 Αεροδυναμική οχημάτων

Η αεροδυναμική αποτελεί έναν από τους σημαντικότερους παράγοντες στη σχεδίαση και την απόδοση των σύγχρονων οχημάτων. Από τα απλά καθημερινά αυτοκίνητα μέχρι και οχήματα που κατασκευάζονται για αγωνιστική χρήση, επηρεάζονται από τον αέρα. Ο τρόπος με τον οποίο ο αέρας κινείται γύρω από το αμάξωμα επηρεάζει άμεσα την ταχύτητα, τη σταθερότητα, την κατανάλωση ενέργειας και τη συνολική οδηγική συμπεριφορά.

Η αεροδυναμική ενός οχήματος σχετίζεται κυρίως με δύο βασικές δυνάμεις: την αεροδυναμική αντίσταση (drag) και την παραγόμενη κάθετη δύναμη (downforce). Όσο η κάθετη δύναμη αυξάνεται τόσο και μεγαλώνει και η αντίσταση του αέρα, η αντίσταση του αέρα ωστόσο δεν σημαίνει απαραίτητα ότι θα μεγαλώσει και την κάθετη δύναμη καθώς μπορεί να οφείλεται σε λάθος σχεδίαση του οχήματος ή των αεροδυναμικών στοιχείων. Όταν η οπισθέλκουσα είναι ιδιαίτερα υψηλή, επιβραδύνει το όχημα και αυξάνει την απαιτούμενη ενέργεια για την κίνησή του, ενώ η κάθετη δύναμη συμβάλλει στη σταθερότητα και την πρόσφυση των τροχών στο οδόστρωμα. Έτσι όταν σχεδιάζεται ένα όχημα ή τροποποιείται, είναι σημαντικό να προσδιοριστεί εξαρχής ο σκοπός και η χρήση του.

Η εξέλιξη της τεχνολογίας και των υπολογιστικών συστημάτων έχει επιτρέψει τη συνεχή βελτίωση της αεροδυναμικής των οχημάτων. Η εξέλιξη αυτή επέτρεψε στους μηχανικούς να αναλύουν με ακρίβεια τη συμπεριφορά του αέρα και να πραγματοποιούν στοχευμένες τροποποιήσεις. Έτσι, η αεροδυναμική δεν αποτελεί μόνο στοιχείο σχεδίασης, αλλά έναν κρίσιμο τομέα της μηχανολογίας που επηρεάζει άμεσα την ασφάλεια, την οικονομία και τις επιδόσεις κάθε αυτοκινήτου. Αυτό οδήγησε στην ανάπτυξη ενεργών αεροδυναμικών διατάξεων που έχουν τη δυνατότητα να μεταβάλλουν δυναμικά τη συμπεριφορά τους ανάλογα με τις συνθήκες λειτουργίας του οχήματος.

Η λειτουργία των ενεργών αεροδυναμικών συστημάτων βασίζεται σε ένα πρωτόκολλο επικοινωνίας που είναι υπεύθυνο για την μετάδοση της πληροφορίας. Το πρωτόκολλο αυτό ονομάζεται CAN bus (Controller Area Network) και συνδέει όλα τα υποσυστήματα του οχήματος μεταξύ τους. Επιτρέπει την επικοινωνία των υπομονάδων τόσο μεταξύ τους όσο και με την κεντρική ECU (Electronic Control Unit) του οχήματος.

1.2 Ενεργά αεροδυναμικά συστήματα

Σε αντίθεση με τα συμβατικά παθητικά αεροδυναμικά στοιχεία, τα ενεργά συστήματα έχουν τη δυνατότητα να προσαρμόζουν τη γεωμετρία ή τη θέση συγκεκριμένων εξαρτημάτων ανάλογα με τις συνθήκες λειτουργίας. Με τον τρόπο αυτό επιτυγχάνεται βελτιστοποίηση της αεροδυναμικής συμπεριφοράς του οχήματος, εξασφαλίζοντας καλύτερη ισορροπία μεταξύ αεροδυναμικής αντίστασης και παραγόμενης κάθετης δύναμης.

Η λειτουργία των Ενεργών Αεροδυναμικών Μερών βασίζεται στη συνεχή συλλογή και επεξεργασία δεδομένων από αισθητήρες του οχήματος. Κάποια παραδείγματα παραμέτρων είναι:

- η ταχύτητα του οχήματος
- η γωνία διεύθυνσης (γωνία τιμονιού)
- η θέση του πεντάλ επιτάχυνσης
- η θέση του πεντάλ πέδησης

Οι πληροφορίες καταγράφονται από αισθητήρες, μεταδίδονται από την τοπική μονάδα μέσω του CAN bus πρωτοκόλλου και αποστέλλονται στην κεντρική μονάδα. Η κεντρική ECU αναλύει, επεξεργάζεται τα δεδομένα και αποστέλλει την εντολή για τη διαχείριση των ενεργοποιητών (actuators), οι οποίοι μεταβάλλουν τη θέση ή τη γεωμετρία των αεροδυναμικών στοιχείων του οχήματος.

Η ενσωμάτωση ενεργών αεροδυναμικών συστημάτων επιτρέπει τη δυναμική προσαρμογή της αεροδυναμικής συμπεριφοράς του οχήματος μέσω διαφορετικών προφίλ λειτουργίας. Τα αεροδυναμικά στοιχεία παραμένουν σε κλειστή ή σε ουδέτερη θέση με σκοπό τη μείωση της αντίστασης του αέρα, εξασφαλίζοντας άνετη οδήγηση και χαμηλή κατανάλωση ενέργειας. Αντίθετα, με την επιλογή πιο δυναμικών λειτουργιών, το σύστημα μεταβάλλει άμεσα τη γεωμετρία των αεροδυναμικών επιφανειών, αυξάνοντας ή μειώνοντας την παραγόμενη κάθετη δύναμη, ανάλογα με το επιλεγμένο προφίλ, βελτιώνοντας τη σταθερότητα, την πρόσφυση, τη συνολική οδηγική απόδοση σε υψηλές ταχύτητες και την ασφάλεια.

Οι ενεργές αεροτομές αποτελούν μια από τις συνηθέστερες εφαρμογές των Ενεργών Αεροδυναμικών Συστημάτων όπως και τα κινητά πτερύγια τα οποία είναι τοποθετημένα στην εμπρόσθια μάσκα του οχήματος. Πρόκειται για κινητά αεροδυναμικά στοιχεία που μεταβάλλουν τη γωνία κλίσης ή το ύψος τους ανάλογα με τις απαιτήσεις του οχήματος και είναι υπεύθυνα για την διαχείριση της ροής του αέρα.

Σε υψηλές ταχύτητες, η αεροτομή και τα πτερύγια μπορεί να μειώνουν τη γωνία προσβολής ώστε να περιορίζεται η αντίσταση του αέρα, ενώ στις στροφές αυξάνουν τη γωνία της με σκοπό την παραγωγή μεγαλύτερης κάθετης δύναμης. Συμβάλουν σημαντικά και στην πέδηση μέσω της λειτουργίας τύπου “air brake”, αυξάνοντας την αντίσταση του αέρα, μειώνοντας την απόσταση πέδησης και διατηρώντας την ευστάθεια του οχήματος. Επιπλέον, τα μπροστινά πτερύγια χρησιμοποιούνται για την ψύξη των φρένων και του κινητήρα.

1.3 Εφαρμογές ενεργών αεροδυναμικών συστημάτων

Τα Ενεργά Αεροδυναμικά Συστήματα εφαρμόζονται πλέον σε μεγάλο εύρος σύγχρονων οχημάτων υψηλών επιδόσεων, αποτελώντας βασικό στοιχείο της σχεδίασης και της δυναμικής συμπεριφοράς τους. Χαρακτηριστικά παραδείγματα αποτελούν οχήματα όπως η Porsche 911 GT3 RS (σχήμα 1.1) και η Mercedes-AMG ONE (σχήμα 1.2), τα οποία ενσωματώνουν προηγμένες ενεργές αεροδυναμικές διατάξεις.



Σχήμα 1.1: Όχημα Porsche 911 GT3 RS με ενεργοποιημένο σύστημα Low Downforce, για χαμηλή παραγωγή κάθετης δύναμης, η πίσω μεταβλητό κομμάτι της αεροτομής βρίσκεται στην ελάχιστη γωνία προσβολής. Πηγή: Porsche AG [1]



Σχήμα 1.2: Το Mercedes-AMG ONE με όλα τα αεροδυναμικά στοιχεία σε θέση μέγιστης κάθετης δύναμης. Οι περσίδες είναι ενεργοποιημένες κατευθύνοντας τη ροή του αέρα γύρω από το αμάξωμα και προς την πίσω αεροτομή. Η αεροτομή βρίσκεται στη θέση μέγιστης κάθετης δύναμης με την προέκταση ανασηκωμένη, αυξάνοντας τη γωνία προσβολής του αέρα. Πηγή: Mercedes-AMG [2]

Η Porsche έχει αναπτύξει το δικό της σύστημα αεροδυναμικής, ονομάζεται Porsche Active Aerodynamics (PAA) και το εφαρμόζει σε επιβατηγά οχήματα [1]. Το PAA έχει ενσωματωθεί στο αυτοκίνητο της παράγωγης Porsche 911 GT3 RS. Διαθέτει ενεργή πίσω αεροτομή και εμπρόσθια πτερύγια μεταβλητής γωνίας, διαθέτει δύο επιλογές low downforce και dynamic downforce. Το low downforce χρησιμοποιείται για να παράγεται χαμηλή κάθετη δύναμη, τα εμπρόσθια πτερύγια τοποθετούνται σε τέτοια κλίση ώστε να κατευθύνουν τον αέρα κάτω από το όχημα (σχήμα 1.3) με σκοπό να μην περνάει ο αέρας από τους αεραγωγούς. Η θέση της πίσω αεροτομής προσαρμόζει την γωνία προσβολής ώστε να μην υπάρχει αντίσταση του αέρα (σχήμα 1.4.) Στην επιλογή dynamic downforce, η αεροτομή βρίσκεται σε ιδιαίτερα μεγάλη γωνία προσβολής (σχήμα 1.5) ενώ τα εμπρόσθια πτερύγια οδηγούν την ροή του αέρα μέσα από τους αεραγωγούς προκαλώντας μεγαλύτερη κάθετη δύναμη (σχήμα 1.6). Το Drag Reduction System (DRS) αλλάζει προσωρινά από dynamic downforce σε low downforce, κατ' αυτόν τον τρόπο βοηθάει το όχημα να αναπτύξει ταχύτητα γρηγορότερα ή ακόμη και να αυξήσει την τελική ταχύτητα του οχήματος. Ενεργοποιείται τον οδηγό εφόσον το κρίνει απαραίτητο. Για να ενεργοποιηθεί το DRS το όχημα θα πρέπει να βρίσκεται σε λειτουργία dynamic downforce.



Σχήμα 1.3: Η ροή του αέρα οδηγείται κάτω από το αμάξωμα, αποτρέποντας τον αέρα να διέλθει από τους αεραγωγούς. Πηγή: Porsche AG [1]



Σχήμα 1.4: Η κλίση της πίσω αεροτομής βρίσκεται σε ιδιαίτερα μικρή γωνία προσβολής με αποτέλεσμα η αντίσταση του αέρα είναι σε εξαιρετικά χαμηλότερα επίπεδα. Πηγή: Porsche AG [1]



Σχήμα 1.5: Η γωνία κλίσης της αεροτομής βρίσκεται σε 34 μοίρες, η επιφάνεια που έρχεται σε επαφή με τον αέρα προκαλώντας μεγαλύτερη κάθετη δύναμη και αυξημένη αντίσταση του αέρα. Πηγή: Porsche AG [1]



Σχήμα 1.6: Η γωνία από εμπρόσθια πτερύγια, οδηγεί την ροή του αέρα μέσα από τους αεραγωγούς, μειώνοντας τη ροή του αέρα κάτω από το αυτοκίνητο. Πηγή: Porsche AG [1]

Αντίστοιχα, η Mercedes-AMG ONE αξιοποιεί ιδιαίτερα εξελιγμένα ενεργά αεροδυναμικά συστήματα, τα οποία έχουν επηρεαστεί άμεσα από την τεχνολογία της Formula 1 [2]. Το όχημα διαθέτει πολλαπλά κινητά αεροδυναμικά στοιχεία, όπως ενεργή πίσω αεροτομή, μεταβαλλόμενα εμπρόσθια πτερύγια και περσίδες για τον έλεγχο της ροής του αέρα στο εμπρόσθιο μέρος του οχήματος. Η λειτουργία τους προσαρμόζεται συνεχώς ανάλογα με την ταχύτητα, την επιτάχυνση, την πέδηση και τις πλευρικές δυνάμεις που αναπτύσσονται κατά την κίνηση. Η ενεργή πίσω αεροτομή παρουσιάζει δύο βασικές θέσεις λειτουργίας. Στην κλειστή θέση (σχήμα 1.7) παραμένει ενσωματωμένη στο αμάξωμα με στόχο τη μείωση της αεροδυναμικής αντίστασης. Στην ανοιχτή θέση (σχήμα 1.2) αναδύεται από το αμάξωμα, ενώ παράλληλα αναπτύσσεται μια μικρή προέκταση που μεταβάλλει τη γωνία προσβολής του αέρα, αυξάνοντας την παραγόμενη κάθετη δύναμη και την αεροδυναμική αντίσταση όταν αυτό απαιτείται. Επιπλέον, οι περσίδες στα εμπρόσθια φτερά λειτουργούν με τρόπο on/off για τη διαχείριση της ροής του αέρα γύρω από το αμάξωμα, ενώ τα εμπρόσθια πτερύγια ελέγχουν τη ροή κάτω από το όχημα (σχήμα 1.8) [3].



Σχήμα 1.7: Τα ενεργά αεροδυναμικά είναι πλήρως απενεργοποιημένα. Οι περσίδες στα εμπρόσθια φτερά του οχήματος και η αεροτομή στο πίσω μέρος βρίσκονται σε πλήρη απορρόφηση από την μορφή του οχήματος για την ελαχιστοποίηση της παραγωγής της κάθετης δύναμης. Πηγή: Mercedes-AMG [2]



Σχήμα 1.8: Εμπρόσθια ενεργά πτερύγια στο εμπρόσθιο κάτω μέρος του οχήματος και σε θέση ώστε να κατευθύνουν τον αέρα κάτω από το όχημα. Πηγή: Mercedes-AMG [3]

Αν και οι επιδόσεις τους διαφέρουν, παρατηρείται κοινή λογική στη διαχείριση της αεροδυναμικής, τόσο ως προς την τοποθέτηση των στοιχείων όσο και ως προς τον τρόπο ελέγχου τους. Τα παραδείγματα

αυτά καταδεικνύουν ότι η ενσωμάτωση ενεργών αεροδυναμικών συστημάτων αποτελεί πλέον καθοριστικό παράγοντα στη σχεδίαση οχημάτων υψηλών επιδόσεων, παραμένοντας ωστόσο απύσχα από τα συμβατικά οχήματα.

1.4 Προβλήματα που αντιμετωπίζουν τα συμβατικά οχήματα

Τα οχήματα υψηλών επιδόσεων κατασκευάζονται εξ αρχής με ιδιαίτερη έμφαση στην αεροδυναμική και στις επιδόσεις. Οι εταιρείες που τα σχεδιάζουν επενδύουν σημαντικά σε αεροδυναμικές δοκιμές, χρήση ελαφριών υλικών και προηγμένων τεχνολογιών, με αποτέλεσμα τα οχήματα αυτά να διαθέτουν ήδη βελτιστοποιημένη συμπεριφορά.

Ωστόσο, τα συμβατικά αυτοκίνητα αποτελούν την πιο δημοφιλή επιλογή για βελτίωση καθώς συνδυάζουν χαμηλότερο κόστος αγοράς, τροποποίησης και καθημερινής χρήσης. Εκτός από την επιλογή αναβάθμισης για ιδιωτική χρήση, επιλέγονται συμβατικά αυτοκίνητα και για αγωνιστική χρήση.

Το πρόβλημα εμφανίζεται όταν τέτοια οχήματα χρειάζονται αεροδυναμική υποστήριξη. Η μετατροπή του αμαξώματος για την αύξηση της σταθερότητας είναι απαραίτητη. Σε τέτοιες περιπτώσεις αυτά τα οχήματα περιορίζονται στη χρήση παθητικών αεροδυναμικών συστημάτων. Η ρύθμιση πρέπει να γίνεται χειροκίνητα και σε πολλές περιπτώσεις δεν υπάρχει η δυνατότητα ρύθμισης, με αποτέλεσμα να περιορίζονται οι δυνατότητες του οχήματος. Η ενσωμάτωση ενεργών αεροδυναμικών σε συμβατικά οχήματα καθημερινής χρήσης είναι δυσεύρετη και πολλές φορές μεγάλου κόστους. Η ανάπτυξη ενός προσιτού και προσαρμόσιμου ενεργού αεροδυναμικού συστήματος για συμβατικά οχήματα αποτελεί επομένως ένα υπαρκτό τεχνολογικό κενό.

1.5 Σκοπός της εργασίας

Σκοπός της εργασίας είναι η ανάπτυξη ενός συστήματος που διαχειρίζεται ενεργά αεροδυναμικά μέρη ενός συμβατικού οχήματος, για την δυνατότητα ελέγχου της κάθετης δύναμης, της οπισθέλκουσας και της ροής του αέρα. Είναι ένα σύστημα πραγματικού χρόνου για τον έλεγχο servo κινητήρων με σκοπό την εφαρμογή τους σε αεροτομή. Η εργασία υλοποιήθηκε με την χρήση μικροελεγκτή ESP32, CAN controller MCP2515, κινητήρες servo 80kg/cm και step-down converter LM2596 για τη διαχείριση της τάσης. Πραγματοποιήθηκε καταγραφή των δεδομένων, ανάλυση και εντοπισμός της χρήσιμης πληροφορίας, μετάφραση και επεξεργασία των bytes με σκοπό την οδήγηση των servo κινητήρων.

Κεφάλαιο 2ο: Πρωτόκολλο CAN bus

2.1 Εισαγωγή

Το πρωτόκολλο CAN bus (Controller Area Network) αποτελεί ένα πρωτόκολλο σειριακής επικοινωνίας που σχεδιάστηκε για να επιτρέπει την αξιόπιστη και γρήγορη ανταλλαγή δεδομένων μεταξύ ηλεκτρονικών μονάδων ελέγχου (Electronic Control Units - ECUs) σε ένα κοινό δίκτυο. Δημιουργήθηκε από την Bosch με στόχο τη μείωση της πολυπλοκότητας των καλωδιώσεων στα οχήματα και την ταυτόχρονη αύξηση της αξιοπιστίας της επικοινωνίας μεταξύ των ηλεκτρονικών συστημάτων [4].

Σήμερα, το CAN bus χρησιμοποιείται σχεδόν σε κάθε σύγχρονο όχημα, καθώς και σε βιομηχανικούς αυτοματισμούς, ρομποτική και αεροναυπηγικές εφαρμογές. Η μετάδοση δεδομένων πραγματοποιείται μέσω ενός κοινού διαύλου επικοινωνίας, επιτρέποντας την ανταλλαγή πληροφοριών μεταξύ διαφορετικών υποσυστημάτων του οχήματος με υψηλή αξιοπιστία και ανθεκτικότητα σε ηλεκτρομαγνητικές παρεμβολές.

Από την άποψη του μοντέλου OSI (Open Systems Interconnection), το CAN bus υλοποιεί κυρίως το Φυσικό Επίπεδο (Physical Layer) και το Επίπεδο Ζεύξης Δεδομένων (Data Link Layer). Πρωτόκολλα ανώτερου επιπέδου, όπως το CANopen, το SAE J1939 και το ISO 15765-4 (βάση του OBD-II), χτίζονται πάνω στο CAN για να παρέχουν τυποποιημένη διεπαφή επικοινωνίας σε διάφορες βιομηχανικές και αυτοκινητιστικές εφαρμογές.

2.2 Βασική αρχή λειτουργίας του CAN Bus

Σε ένα δίκτυο CAN bus δεν υπάρχει κεντρικός υπολογιστής που να ελέγχει όλες τις μονάδες. Αντίθετα, κάθε ECU μπορεί να λειτουργεί τόσο ως πομπός όσο και ως δέκτης δεδομένων. Όλες οι μονάδες συνδέονται πάνω σε ένα κοινό δίκτυο δύο καλωδίων, γνωστό ως CAN High (CAN-H) και CAN Low (CAN-L).

Η επικοινωνία πραγματοποιείται διαφορικά (differential signaling), γεγονός που επιτρέπει στο σύστημα να είναι ιδιαίτερα ανθεκτικό στον ηλεκτρικό θόρυβο. Όταν μεταδίδεται ένα bit, το CAN-H αυξάνει τάση ενώ το CAN-L μειώνει τάση. Η διαφορά τάσης μεταξύ των δύο γραμμών είναι αυτή που «διαβάζεται» ως πληροφορία.

2.2.1 Dominant και Recessive Bits

Θεμελιώδης έννοια του πρωτοκόλλου CAN είναι η διάκριση μεταξύ δύο λογικών καταστάσεων του διαύλου: του dominant bit (κυρίαρχο, λογικό 0) και του recessive bit (υποχωρητικό, λογικό 1).

- i. Dominant bit (0): Το CAN-H βρίσκεται σε υψηλή τάση (~3.5V) και το CAN-L σε χαμηλή (~1.5V), με διαφορά ~2V. Αυτή η κατάσταση «επιβάλλεται» στον δίαυλο.
- ii. Recessive bit (1): Και οι δύο γραμμές βρίσκονται στα ~2.5V, με μηδενική διαφορά τάσης. Αυτή η κατάσταση ισχύει μόνο όταν κανένας κόμβος δεν μεταδίδει dominant.

Ο κανόνας αυτός διέπει ολόκληρη τη λογική του πρωτοκόλλου: όταν δύο κόμβοι μεταδίδουν ταυτόχρονα και ο ένας εκπέμπει dominant (0) ενώ ο άλλος recessive (1), ο δίαυλος λαμβάνει πάντα την κατάσταση dominant. Αυτή η ιδιότητα δεν είναι τυχαία — αποτελεί τον πυρήνα του μηχανισμού arbitration, ο οποίος αναλύεται διεξοδικά στην ενότητα 2.4.

2.2.2 Τοπολογία δικτύου και τερματισμός

Το δίκτυο CAN υιοθετεί γραμμική τοπολογία (bus topology), στην οποία όλοι οι κόμβοι συνδέονται παράλληλα στον ίδιο δίαυλο δύο καλωδίων. Στα δύο άκρα του διαύλου τοποθετούνται αντιστάσεις τερματισμού (termination resistors) τιμής 120Ω η καθεμία. Οι δύο αντιστάσεις συνδέονται παράλληλα, δίνοντας συνολική αντίσταση 60Ω, η οποία αντιστοιχεί στη χαρακτηριστική αντίσταση του καλωδίου και εξαλείφει τα φαινόμενα ανάκλασης σήματος [5].

Η απουσία τερματικών αντιστάσεων ή η χρήση λανθασμένων τιμών μπορεί να οδηγήσει σε αναξιόπιστη επικοινωνία, σφάλματα πλαισίων και αδυναμία επικοινωνίας σε υψηλές ταχύτητες. Κατά τη σχεδίαση οποιουδήποτε συστήματος που συνδέεται στο CAN bus ενός οχήματος, η επαλήθευση της ύπαρξης τερματικών αντιστάσεων αποτελεί βασικό βήμα επαλήθευσης.

2.2.3 Ταχύτητα επικοινωνίας CAN Bus

Ένα από τα βασικά χαρακτηριστικά του δικτύου CAN είναι η ταχύτητα επικοινωνίας (bit rate), η οποία εκφράζει τον αριθμό των bits που μεταδίδονται ανά δευτερόλεπτο. Η επιλογή της κατάλληλης ταχύτητας εξαρτάται από τις απαιτήσεις του εκάστοτε συστήματος, όπως ο όγκος των δεδομένων, οι απαιτήσεις πραγματικού χρόνου και η φυσική έκταση του δικτύου.

Στα δίκτυα CAN των οχημάτων χρησιμοποιούνται συνήθως ταχύτητες [4] [6]:

- iii. 125 kbps – συστήματα άνεσης αμαξώματος
- iv. 250 kbps – ενδιάμεσα συστήματα
- v. 500 kbps – κρίσιμα συστήματα ελέγχου
- vi. 1 Mbps – μέγιστη ταχύτητα κλασικού CAN

Η πιο διαδεδομένη ταχύτητα στα δίκτυα υψηλής προτεραιότητας είναι τα 500 kbps, καθώς προσφέρει ικανοποιητική ισορροπία μεταξύ ταχύτητας μετάδοσης, αξιοπιστίας και ανοχής στον ηλεκτρομαγνητικό θόρυβο. Ωστόσο, η αύξηση της ταχύτητας επικοινωνίας συνεπάγεται περιορισμό του μέγιστου μήκους του διαύλου, εξαιτίας ζητημάτων συγχρονισμού και υποβάθμισης του σήματος.

Σε χαμηλότερες ταχύτητες εφαρμόζονται κυρίως σε υποσυστήματα άνεσης και αμαξώματος, όπως ο φωτισμός, ο κλιματισμός, τα ηλεκτρικά παράθυρα και το κεντρικό κλείδωμα, όπου οι απαιτήσεις χρονικής απόκρισης είναι μικρότερες. Αντίθετα, υψηλότερες ταχύτητες χρησιμοποιούνται σε κρίσιμα συστήματα ελέγχου, όπως η μονάδα κινητήρα (ECU), το σύστημα πέδησης ABS, το ηλεκτρονικό σύστημα ευστάθειας (ESP) και τα συστήματα υποβοήθησης οδηγού (ADAS), στα οποία απαιτείται ταχεία και συνεχής ανταλλαγή δεδομένων.

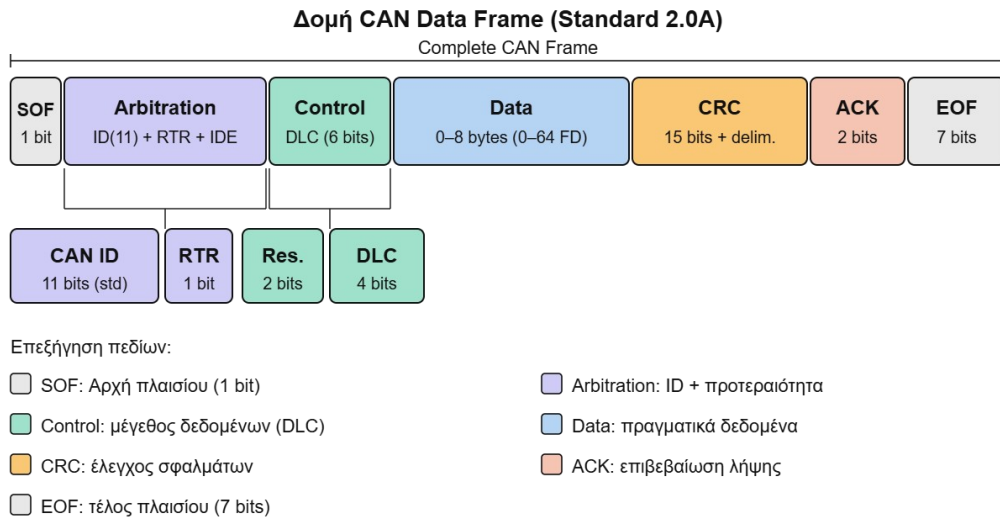
Στα νεότερα οχήματα χρησιμοποιείται το CAN FD (Flexible Data Rate), το οποίο αποτελεί εξέλιξη του κλασικού CAN. Το CAN FD επιτρέπει μεγαλύτερο μέγεθος πλαισίου δεδομένων (έως 64 bytes αντί για 8 bytes) καθώς και υψηλότερες ταχύτητες μετάδοσης κατά τη φάση μεταφοράς δεδομένων, οι οποίες μπορούν να φτάσουν αρκετά Mbps.

Η επιλογή της ταχύτητας σε ένα δίκτυο CAN αποτελεί κρίσιμο παράγοντα σχεδιασμού, καθώς επηρεάζει άμεσα την απόδοση, την αξιοπιστία και τη λειτουργικότητα των ηλεκτρονικών συστημάτων του οχήματος.

2.2.4 Δομή ενός CAN Frame

Η επικοινωνία στο δίκτυο CAN βασίζεται στη μετάδοση πλαισίων δεδομένων (CAN frames) μέσω κοινού διαύλου επικοινωνίας. Κάθε frame ακολουθεί καθορισμένη δομή, η οποία ορίζεται από το πρότυπο ISO 11898, και αποτελείται από τα εξής πεδία (σχήμα 2.1) [4] [5]:

- vii. Start of Frame (SOF): Σηματοδοτεί την έναρξη της μετάδοσης με ένα dominant bit.
- viii. Arbitration Field: Περιέχει τον αναγνωριστικό κωδικό (CAN ID) που καθορίζει την προτεραιότητα πρόσβασης στον διάυλο.
- ix. Control Field: Προσδιορίζει το μέγεθος των δεδομένων μέσω του πεδίου DLC (Data Length Code).
- x. Data Field: Μεταφέρει τα πραγματικά δεδομένα (0–8 bytes στο κλασικό CAN, έως 64 bytes στο CAN FD).
- xi. Cyclic Redundancy Check (CRC): Εξασφαλίζει την ακεραιότητα των δεδομένων μέσω ελέγχου σφαλμάτων.
- xii. Acknowledgement Field (ACK): Επιβεβαιώνει την επιτυχή λήψη του frame από τους κόμβους του δικτύου.
- xiii. End of Frame (EOF): Σηματοδοτεί την ολοκλήρωση της μετάδοσης.



Σχήμα 2.1: Δομή CAN Data Frame (Standard 2.0A). Ίδια κατασκευή βάσει: Robert Bosch GmbH, CAN Specification 2.0, 1991 [4].

Το CAN ID αποτελεί κεντρικό στοιχείο του πρωτοκόλλου, καθώς καθορίζει τόσο την προτεραιότητα του μηνύματος όσο και το είδος της πληροφορίας που μεταφέρει. Σε αντίθεση με άλλα πρωτόκολλα επικοινωνίας, το CAN ID δεν αποτελεί διεύθυνση παραλήπτη, αλλά αναγνωριστικό του τύπου και του περιεχομένου του μηνύματος. Διακρίνονται δύο τύποι: ο standard (11-bit, 2.048 μοναδικές τιμές) που απαντάται κυρίως σε επιβατικά οχήματα, και ο extended (29-bit) που χρησιμοποιείται σε βαρέα οχήματα και βιομηχανικές εφαρμογές (π.χ. SAE J1939).

Το Data Field μεταφέρει τα πραγματικά δεδομένα που ανταλλάσσονται μεταξύ των ECUs. Κάθε byte κωδικοποιεί συγκεκριμένη παράμετρο, όπως στροφές κινητήρα, θερμοκρασία λαδιού, πίεση υπερπλήρωσης, θέση πεταλούδας γκαζιού ή θέση πεντάλ φρένου/γκαζιού. Η ακριβής αντιστοίχιση των bytes σε φυσικές ποσότητες ορίζεται από τα αρχεία περιγραφής δικτύου (DBC files) ή από ιδιόκτητες προδιαγραφές του εκάστοτε κατασκευαστή.

Εκτός από τα Data Frames, το πρωτόκολλο CAN ορίζει και τα Remote Frames (RTR frames). Πρόκειται για πλαίσια αιτήματος, μέσω των οποίων ένας κόμβος ζητά από έναν άλλο κόμβο να αποστείλει ένα συγκεκριμένο μήνυμα δεδομένων. Τα Remote Frames έχουν την ίδια δομή με τα Data Frames, με τη διαφορά ότι το πεδίο δεδομένων είναι κενό και το bit RTR (Remote Transmission Request) του Arbitration Field είναι recessive.

Συνοψίζοντας, το πρωτόκολλο CAN προσφέρει αξιόπιστη και αποδοτική επικοινωνία μεταξύ των ECUs του οχήματος, μέσω μιας καλά ορισμένης δομής πλαισίων και μηχανισμών ελέγχου σφαλμάτων. Η κατανόηση αυτών των εννοιών κρίνεται απαραίτητη για την ανάλυση του υλικού και των εξαρτημάτων

που χρησιμοποιήθηκαν στο πλαίσιο της παρούσας εργασίας, τα οποία παρουσιάζονται στο επόμενο κεφάλαιο.

2.3 Μηχανισμός Arbitration

Ένα από τα πιο σημαντικά χαρακτηριστικά του CAN bus είναι ο μηχανισμός διαίτησίας (arbitration), ο οποίος επιλύει συγκρούσεις όταν δύο ή περισσότεροι κόμβοι επιχειρούν να μεταδώσουν ταυτόχρονα. Η διαδικασία αυτή είναι μη-καταστροφική (non-destructive), δηλαδή δεν χάνεται κανένα μήνυμα κατά τη διάρκειά της.

Η διαδικασία arbitration βασίζεται στη bitwise σύγκριση των CAN IDs κατά τη μετάδοση [4] [6]. Κάθε κόμβος μεταδίδει τα bits του ID του ένα-ένα, ενώ ταυτόχρονα «ακούει» τον δίαυλο. Εφαρμόζεται η λογική "dominant wins": εάν ένας κόμβος μεταδίδει recessive (1) αλλά διαβάζει dominant (0) στον δίαυλο, σημαίνει ότι κάποιος άλλος κόμβος με ίδιο ή ανώτερης προτεραιότητας ID μεταδίδει dominant. Ο κόμβος που ανιχνεύει αυτή τη διαφορά σταματά αμέσως τη μετάδοση και μεταπίπτει σε κατάσταση αναμονής.

Το αποτέλεσμα είναι ότι ο κόμβος με το χαμηλότερο αριθμητικό CAN ID «κερδίζει» το arbitration και συνεχίζει τη μετάδοση χωρίς καθυστέρηση ή απώλεια δεδομένων. Οι κόμβοι που «έχασαν» επαναλαμβάνουν αυτόματα τη μετάδοση όταν ο δίαυλος αποδεσμευτεί. Αυτό σημαίνει ότι μηνύματα κρίσιμης σημασίας (με χαμηλό ID) πάντα προηγούνται έναντι μηνυμάτων χαμηλότερης προτεραιότητας.

2.4 Ανίχνευση και Διαχείριση Σφαλμάτων

Το πρωτόκολλο CAN ενσωματώνει εξελιγμένους μηχανισμούς ανίχνευσης και διαχείρισης σφαλμάτων, οι οποίοι το καθιστούν ιδιαίτερα αξιόπιστο σε απαιτητικά περιβάλλοντα. Ορίζονται πέντε τύποι σφαλμάτων [4]:

- xiv. Bit Error: Ο κόμβος ανιχνεύει ότι το bit που μετέδωσε διαφέρει από αυτό που διαβάζει στον δίαυλο (εκτός arbitration/ACK).
- xv. Stuff Error: Παραβίαση του κανόνα bit stuffing (μετά από 5 consecutive ίδια bits πρέπει να εισαχθεί ένα αντίθετο bit).
- xvi. CRC Error: Το CRC που υπολόγισε ο δέκτης δεν συμφωνεί με αυτό που μεταδόθηκε.
- xvii. Form Error: Εντοπισμός bits λανθασμένης λογικής τιμής σε πεδία σταθερής μορφής (EOF, ACK delimiter).
- xviii. ACK Error: Κανένας κόμβος δεν επιβεβαίωσε (ACK) την επιτυχή λήψη του frame.

Κάθε κόμβος διατηρεί δύο μετρητές σφαλμάτων: τον Transmit Error Counter (TEC) και τον Receive Error Counter (REC). Ανάλογα με τις τιμές τους, ο κόμβος μεταπίπτει σε διαφορετικές καταστάσεις

λειτουργίας:

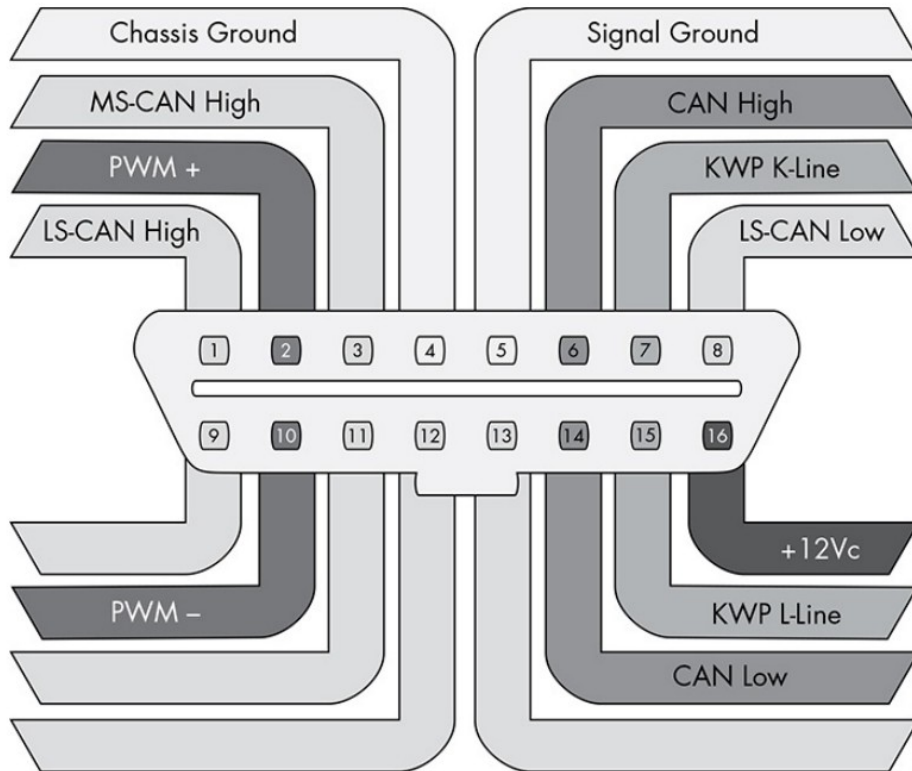
- xix. Error Active (TEC/REC < 128): Κανονική λειτουργία. Ο κόμβος μεταδίδει Active Error Frames όταν ανιχνεύει σφάλμα.
- xx. Error Passive (TEC ή REC \geq 128): Ο κόμβος συνεχίζει να συμμετέχει στο δίκτυο, αλλά μεταδίδει Passive Error Frames και αναμένει επιπλέον χρόνο μεταξύ μεταδόσεων.
- xxi. Bus Off (TEC \geq 256): Ο κόμβος αποσυνδέεται πλήρως από τον διάυλο και δεν συμμετέχει σε καμία επικοινωνία. Απαιτείται επανεκκίνηση για επιστροφή σε λειτουργία.

Ο μηχανισμός αυτός διασφαλίζει ότι ένας δυσλειτουργικός κόμβος δεν μπορεί να «μπλοκάρει» ολόκληρο το δίκτυο CAN, εγγυώμενας την αξιοπιστία του συστήματος ακόμα και σε παρουσία βλαβών.

2.5 Διεπαφή OBD-II και σύνδεση με το CAN Bus

Το πρότυπο OBD-II (On-Board Diagnostics II) υποχρεώνει τα οχήματα από το 1996 και έπειτα να διαθέτουν τυποποιημένη θύρα διάγνωσης για την παρακολούθηση εκπομπών και την απόδοση του κινητήρα [7]. Η θύρα αυτή, γνωστή ως J1962 connector, αποτελεί το φυσικό σημείο πρόσβασης στο δίκτυο CAN bus του οχήματος και χρησιμοποιείται τόσο για διαγνωστικούς σκοπούς όσο και για την ανάγνωση δεδομένων σε πραγματικό χρόνο.

Ο συνδετήρας J1962 είναι 16 ακίδων και πρέπει υποχρεωτικά να βρίσκεται εντός ενός μέτρου από το τιμόνι, στην πλευρά του οδηγού [6]. Για οχήματα μοντέλου 2008 και νεότερα, είναι υποχρεωτική η χρήση του πρωτοκόλλου ISO 15765-4 (CAN bus), με τις ακίδες 6 (CAN-H) και 14 (CAN-L) να μεταφέρουν το διαφορικό σήμα επικοινωνίας, ενώ οι ακίδες 4 και 16 παρέχουν γείωση και τροφοδοσία 12V αντίστοιχα (σχήμα 2.2).



Σχήμα 2.2: Διάταξη ακίδων θύρας OBD-II (J1962) Πηγή: [6]

Το πρωτόκολλο ISO 15765-4 ορίζει τον τρόπο με τον οποίο τα διαγνωστικά δεδομένα OBD-II μεταδίδονται πάνω από το CAN bus, χρησιμοποιώντας ως βάση μεταφοράς το ISO-TP (ISO 15765-2), το οποίο επιτρέπει την επικοινωνία payloads που υπερβαίνουν τα 8 bytes ενός κλασικού CAN frame. Στην πράξη, το πιο κοινό προφίλ για σύγχρονα επιβατικά οχήματα είναι το ISO 15765-4 με 11-bit CAN ID και ταχύτητα 500 kbps.

Στο πλαίσιο της παρούσας εργασίας, η θύρα OBD-II αξιοποιείται ως το σημείο σύνδεσης με το δίκτυο CAN bus του οχήματος, επιτρέποντας την ανάγνωση σε πραγματικό χρόνο παραμέτρων όπως η θέση του πεντάλ γκαζιού και η πίεση φρένου, οι οποίες στη συνέχεια χρησιμοποιούνται ως είσοδοι για τον έλεγχο του συστήματος ενεργητικής αεροδυναμικής.

2.6 Συμπέρασμα

Συνοψίζοντας, το πρωτόκολλο CAN προσφέρει αξιόπιστη και αποδοτική επικοινωνία μεταξύ των ECUs του οχήματος, μέσω μιας καλά ορισμένης δομής πλαισίων, μηχανισμών ελέγχου σφαλμάτων και έξυπνης διαχείρισης προτεραιοτήτων. Η κατανόηση αυτών των εννοιών — και ιδιαίτερα του τρόπου με τον οποίο παράμετροι όπως η θέση του πεντάλ γκαζιού και η πίεση φρένου κωδικοποιούνται στα πεδία δεδομένων των CAN frames — κρίνεται απαραίτητη για την ανάπτυξη του ηλεκτρονικού συστήματος ενεργητικής αεροδυναμικής που παρουσιάζεται στα επόμενα κεφάλαια.

Κεφάλαιο 3ο: Εξολισμός Εργασίας

3.1 Παρουσίαση Εξοπλισμού και Κριτήρια Επιλογής

Στο παρόν κεφάλαιο παρουσιάζονται αναλυτικά τα εξαρτήματα και τα υλικά που χρησιμοποιήθηκαν για την υλοποίηση του συστήματος. Η επιλογή κάθε εξαρτήματος πραγματοποιήθηκε με βάση συγκεκριμένα κριτήρια, όπως η συμβατότητα με το πρωτόκολλο CAN, η διαθεσιμότητα στην αγορά και η ευκολία ενσωμάτωσης στο σύστημα. Επιπλέον, λήφθηκε υπόψη το κόστος απόκτησης, καθώς η εργασία υλοποιήθηκε με περιορισμένο προϋπολογισμό.

Ο παρακάτω πίνακας συνοψίζει τον εξοπλισμό που χρησιμοποιήθηκε και τον ρόλο κάθε εξαρτήματος στο σύστημα:

Πίνακας 3.1: Ρόλος εξαρτημάτων στο σύστημα.

Εξάρτημα	Ρόλος στο Σύστημα
ELM327	Αρχική εξερεύνηση δικτύου CAN μέσω OBD-II
MCP2515 + Arduino UNO R4 Minima	Καταγραφή CAN frames (sniffing)
MCP2515 + ESP32 FireBeetle 2	Επεξεργασία δεδομένων και έλεγχος
SN65HVD230 + ESP32 FireBeetle 2	Επεξεργασία δεδομένων και έλεγχος
Servo 80 kg/cm 270°	Ενεργοποιητής αεροτομής
LM2596	Ρυθμιστικό τάσης τροφοδοσίας servo

Παρακάτω ακολουθεί ο πίνακας κόστους εξαρτημάτων μονάδας:

Πίνακας 3.2: Κόστος εξαρτημάτων.

Εξάρτημα	Κόστος
ELM327	5.70€
MCP2515	4.80€
Arduino UNO R4 Minima	22.80€
ESP32 FireBeetle 2	12.80€
SN65HVD230	1.29€
Servo 80 kg/cm 270°	24.88€
LM2596	0.69€

Παρακάτω παρουσιάζεται αναλυτικά κάθε εξάρτημα, με έμφαση στα τεχνικά χαρακτηριστικά του, στον ρόλο που επιτελεί στο σύστημα και στους λόγους που οδήγησαν στην επιλογή τους.

3.2 Διαγνωστικό ELM327

3.2.1 Γενική περιγραφή

Το ολοκληρωμένο κύκλωμα ELM327 (σχήμα 3.1) αναπτύχθηκε από την канаδική εταιρεία ELM Electronics ως μεταφραστής πρωτοκόλλων OBD-II προς σειριακή επικοινωνία για σύνδεση με υπολογιστές και άλλες εξωτερικές συσκευές. Πρόκειται για ένα ολοκληρωμένο κύκλωμα βασισμένο στον 8-bit μικροελεγκτή PIC18F2480 της Microchip Technology με το αρχικό firmware της ELM. Η ονομασία "327" αποτελεί την πιο καθολική έκδοση της οικογένειας "32x", καθώς υποστηρίζει τον μεγαλύτερο αριθμό πρωτοκόλλων.

Η επιλογή του ELM327 πραγματοποιήθηκε με στόχο την πρόσβαση στα δεδομένα των ηλεκτρονικών συστημάτων του οχήματος, καθώς και την αναγνώριση και παρακολούθηση των μετρήσεων που παρέχονται από τους αισθητήρες και τις ηλεκτρονικές μονάδες ελέγχου (ECUs). Η χρήση του κρίθηκε κατάλληλη έπειτα από τη διαπίστωση της συμβατότητάς του με το συγκεκριμένο όχημα. Το ELM327 αποτελεί ένα ευρέως διαδεδομένο διαγνωστικό εργαλείο, το οποίο παρέχει εύκολη και οικονομικά αποδοτική πρόσβαση σε δεδομένα του συστήματος OBD-II, επιτρέποντας την ανάκτηση και ανάλυση παραμέτρων λειτουργίας του οχήματος σε πραγματικό χρόνο.



Σχήμα 3.1: OBD-II με ELM327 προσαρμογέα με την επαφή εισόδου.

Στο πλαίσιο της παρούσας πτυχιακής εργασίας, ο προσαρμογέας ELM327 χρησιμοποιήθηκε σε πρώτη φάση για την κατανόηση της λειτουργίας του συστήματος OBD-II του οχήματος και σε δεύτερη φάση για τον εντοπισμό των αισθητήρων που αφορούν στα δεδομένα οδήγησης (θέση φρένου, θέση γκαζιού κ.ά.), τα οποία απαιτούνταν για το σύστημα ενεργού αεροδυναμικού ελέγχου.

3.2.2 Αρχιτεκτονική και Τεχνικά Χαρακτηριστικά

Ο ELM327 αποτελεί ουσιαστικά μία γέφυρα μεταξύ της θύρας OBD-II του οχήματος και εξωτερικών συσκευών, όπως φορητοί υπολογιστές, smartphones ή ενσωματωμένα συστήματα. Εκτελεί ιδιόκτητο firmware το οποίο ερμηνεύει τυποποιημένες εντολές OBD-II (PID requests), διαχειρίζεται πολλαπλά πρωτόκολλα επικοινωνίας οχημάτων, όπως CAN, ISO 9141-2, ISO 14230-4 (KWP2000), SAE J1850 PWM και SAE J1850 VPW, και μορφοποιεί τις αποκρίσεις σε αναγνώσιμο ASCII κείμενο μέσω σειριακής επικοινωνίας UART.

Στη συγκεκριμένη συσκευή OBD-II που χρησιμοποιήθηκε στην παρούσα εργασία, η επικοινωνία με το δίκτυο CAN υλοποιείται μέσω του ολοκληρωμένου MCP2551 CAN transceiver (Σχήμα 3.2), ο οποίος συνεργάζεται με τον ELM327 για την ανάγνωση και μετάδοση των δεδομένων του οχήματος. Η συσκευή υποστηρίζει επικοινωνία με πρωτόκολλα OBD-II που λειτουργούν σε ρυθμούς μετάδοσης από 10,4 kbps έως 500 kbps, ανάλογα με το χρησιμοποιούμενο πρωτόκολλο. Παράλληλα, η επικοινωνία του ELM327 με την εξωτερική συσκευή πραγματοποιείται μέσω UART με τυπικούς ρυθμούς μετάδοσης από 9.600 έως 115.200 baud rate, ανάλογα με την έκδοση και το firmware της συσκευής.

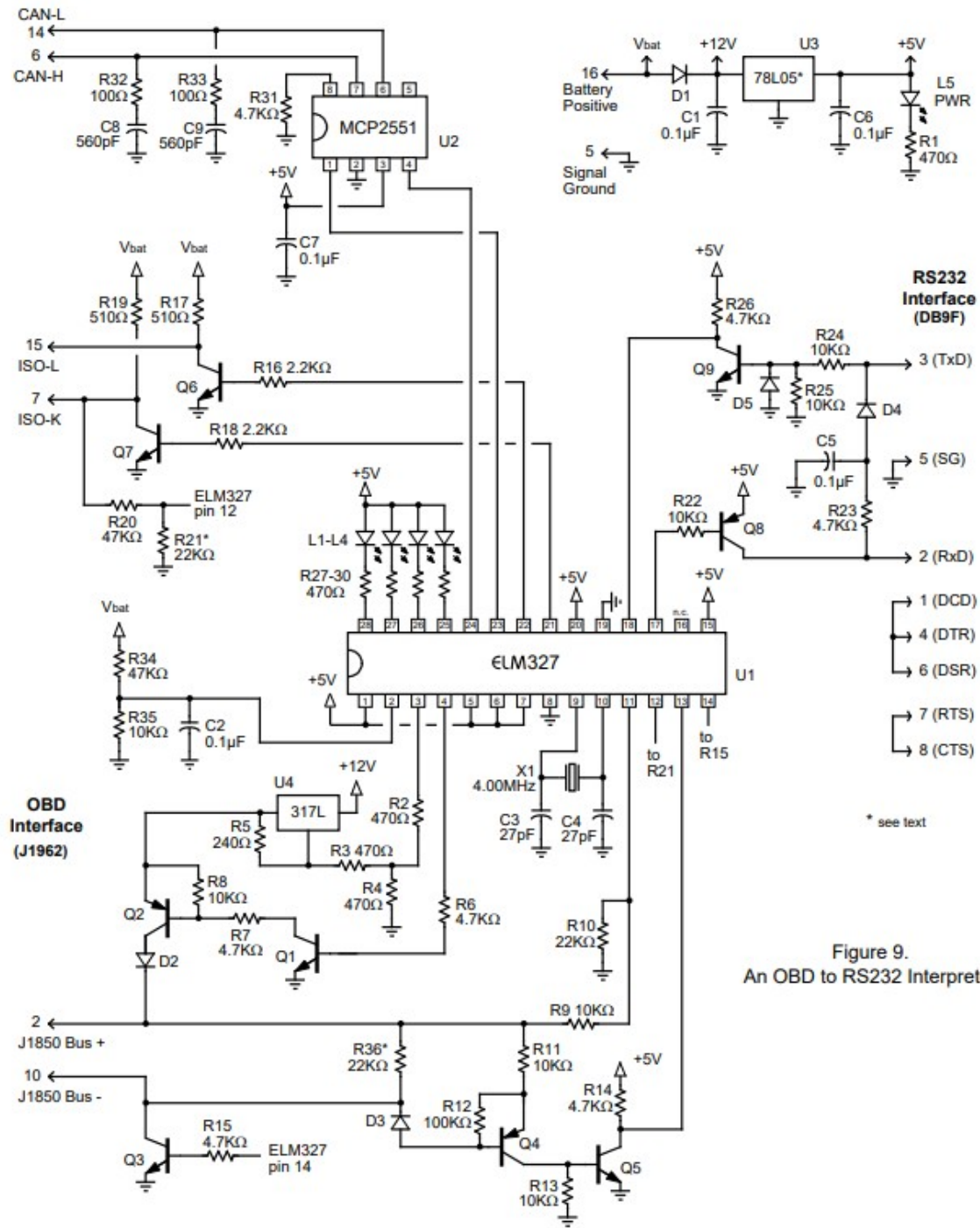


Figure 9.
An OBD to RS232 Interpreter

Σχήμα 3.2: εσωτερική διάταξη κυκλώματος του OBD reader. Πηγή:[8]

3.2.3 Υποστηριζόμενα Πρωτόκολλα OBD-II

Το ELM327 αποτελεί έναν ολοκληρωμένο μικροελεγκτή που σχεδιάστηκε για τη μετάφραση των πρωτοκόλλων επικοινωνίας του συστήματος OBD-II (On-Board Diagnostics II) σε μορφή κατανοητή από ηλεκτρονικές συσκευές. Η βασική λειτουργία του ELM327 είναι η διαμεσολάβηση μεταξύ του συστήματος διαγνωστικού ελέγχου του οχήματος και της συσκευής του χρήστη, μετατρέποντας τα

δεδομένα που ανταλλάσσονται σε κατανοητές εντολές και πληροφορίες. Συνδέεται στη θύρα OBD-II, η οποία βρίσκεται συνήθως κάτω από το ταμπλό του οχήματος. Μέσω αυτής της σύνδεσης αποκτά πρόσβαση στις ηλεκτρονικές μονάδες ελέγχου του οχήματος και συλλέγει δεδομένα σχετικά με τη λειτουργία του κινητήρα και άλλων υποσυστημάτων.

Η επικοινωνία πραγματοποιείται μέσω διαφόρων πρωτοκόλλων, όπως:

- ISO 9141-2
- ISO 14230-4 (KWP2000)
- ISO 15765-4 (CAN Bus)
- SAE J1850 PWM
- SAE J1850 VPW
- SAE J1939

Ο μικροελεγκτής αναγνωρίζει αυτόματα το πρωτόκολλο που χρησιμοποιεί το όχημα και μετατρέπει τις πληροφορίες σε σειριακά δεδομένα, τα οποία μπορούν να αναγνωστούν από λογισμικό διάγνωσης.

Το ELM327 προσφέρει πληθώρα λειτουργιών που το καθιστούν ιδιαίτερα χρήσιμο τόσο για επαγγελματίες τεχνικούς όσο και για ιδιώτες χρήστες. Μεταξύ των σημαντικότερων δυνατοτήτων του περιλαμβάνονται:

- Ανάγνωση διαγνωστικών κωδικών βλαβών (Diagnostic Trouble Codes – DTCs).
- Διαγραφή αποθηκευμένων κωδικών σφαλμάτων.
- Παρακολούθηση δεδομένων σε πραγματικό χρόνο (Real-Time Data).
- Έλεγχος παραμέτρων λειτουργίας κινητήρα, όπως:
 - στροφές κινητήρα (RPM),
 - θερμοκρασία ψυκτικού υγρού,
 - ταχύτητα οχήματος,
 - κατανάλωση καυσίμου,
 - πίεση πολλαπλής εισαγωγής.
- Καταγραφή και αποθήκευση δεδομένων για μεταγενέστερη ανάλυση.

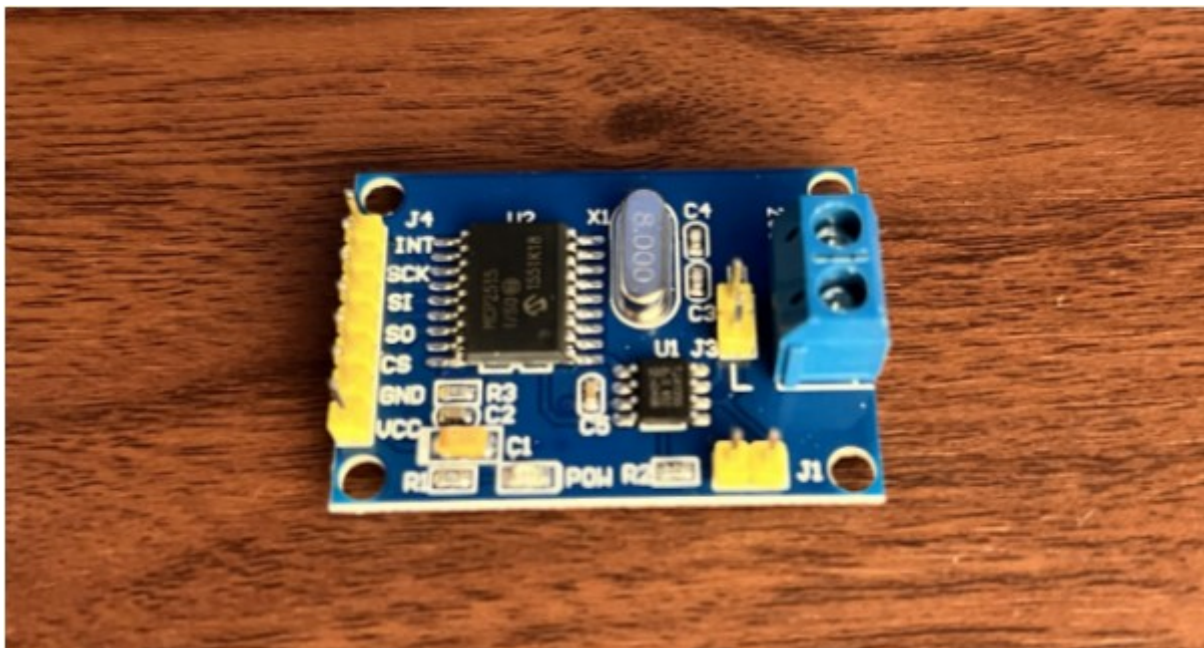
Οι περισσότερες συσκευές ELM327 υποστηρίζουν συνδέσεις USB, Bluetooth ή Wi-Fi, διευκολύνοντας τη χρήση τους σε διαφορετικές πλατφόρμες λογισμικού.

Η υποστήριξη πολλαπλών πρωτοκόλλων OBD-II, η δυνατότητα παρακολούθησης δεδομένων σε πραγματικό χρόνο και η ευρεία συμβατότητα με εφαρμογές λογισμικού έχουν συμβάλει στην καθιέρωσή του ως βασικό μέσο διάγνωσης από χρήστες [8]. Παρότι δεν μπορεί να αντικαταστήσει πλήρως τα εξειδικευμένα επαγγελματικά διαγνωστικά συστήματα, το ELM327 παραμένει μια

αποτελεσματική και προσιτή λύση για την παρακολούθηση και τον έλεγχο της λειτουργίας των σύγχρονων οχημάτων.

3.3 MCP2515

Το MCP2515 (σχήμα 3.3) αποτελεί έναν αυτόνομο ελεγκτή δικτύου CAN (Controller Area Network) που αναπτύχθηκε από τη Microchip Technology. Επιλέχθηκε λόγω της συμβατότητάς του με την πλατφόρμα Arduino μέσω της διεπαφής SPI, η οποία υποστηριζόταν ήδη από το υφιστάμενο υλικό του συστήματος. Επιπλέον, ο MCP2515 αποτελεί μία ευρέως χρησιμοποιούμενη επιλογή για εφαρμογές δικτύων CAN σε ενσωματωμένα συστήματα, γεγονός που έχει οδηγήσει στη διαθεσιμότητα πληθώρας ώριμων βιβλιοθηκών λογισμικού, παραδειγμάτων υλοποίησης και τεχνικής τεκμηρίωσης. Τα χαρακτηριστικά αυτά διευκόλυναν την ανάπτυξη, τη διαμόρφωση και την αποσφαλμάτωση του συστήματος.



Σχήμα 3.3: Μονάδα MCP2515 CAN Bus Controller που χρησιμοποιήθηκε για τη διασύνδεση του μικροελεγκτή με το δίκτυο CAN του οχήματος.

Η κύρια λειτουργία του είναι η διαχείριση της επικοινωνίας μέσω του πρωτοκόλλου CAN χωρίς να επιβαρύνει τον κεντρικό μικροελεγκτή με τις χαμηλού επιπέδου διαδικασίες μετάδοσης και λήψης δεδομένων. Ο MCP2515 είναι συμβατός με το πρότυπο CAN 2.0B, η επικοινωνία του με τον κεντρικό επεξεργαστή πραγματοποιείται μέσω της σειριακής διεπαφής SPI (Serial Peripheral Interface), γεγονός που επιτρέπει την εύκολη ενσωμάτωσή του σε πληθώρα ενσωματωμένων συστημάτων. Υποστηρίζει τυπικά πλαίσια μηνυμάτων (11-bit identifiers) όσο και εκτεταμένα πλαίσια (29-bit identifiers) [9].

3.3.1 Αρχιτεκτονική και λειτουργία

Η αρχιτεκτονική του MCP2515 αποτελείται από διάφορες λειτουργικές μονάδες που συνεργάζονται για την υλοποίηση της επικοινωνίας CAN.

Οι βασικές μονάδες περιλαμβάνουν:

- Τον πυρήνα πρωτοκόλλου CAN (CAN Protocol Engine).
- Τη μονάδα ελέγχου μετάδοσης και λήψης μηνυμάτων.
- Τρεις καταχωρητές μετάδοσης (Transmit Buffers).
- Δύο καταχωρητές λήψης (Receive Buffers).
- Τη μονάδα φιλτραρίσματος και μασκών.
- Τη διεπαφή SPI.
- Τη μονάδα χρονισμού (Bit Timing Logic).

Κατά τη λειτουργία του συστήματος, ο μικροελεγκτής αποστέλλει δεδομένα στον MCP2515 μέσω SPI. Τα δεδομένα αποθηκεύονται στους transmit buffers και στη συνέχεια ο ελεγκτής αναλαμβάνει την αποστολή τους στο δίκτυο CAN. Αντίστοιχα, κατά τη λήψη ενός πλαισίου CAN, ο MCP2515 ελέγχει αν το μήνυμα πληροί τα κριτήρια των φίλτρων αποδοχής. Εφόσον γίνει αποδεκτό, αποθηκεύεται στους receive buffers και παράγεται διακοπή (interrupt) προς τον μικροελεγκτή. Η υποστήριξη μηχανισμών ανίχνευσης σφαλμάτων, διαιτησίας (arbitration) και επαναμετάδοσης πραγματοποιείται πλήρως από τον ίδιο τον MCP2515 σύμφωνα με τις προδιαγραφές του πρωτοκόλλου CAN [9].

3.3.2 Transceiver TJ1050

Ο TJA1050 αποτελεί τον μετατροπέα μεταξύ του φυσικού δικτύου CAN και του ελεγκτή MCP2515. Κατά τη μετάδοση μετατρέπει τα λογικά σήματα TXCAN του MCP2515 σε διαφορικά σήματα CAN-H και CAN-L κατάλληλα για το δίκτυο CAN. Αντίστοιχα, κατά τη λήψη μετατρέπει τα διαφορικά σήματα CAN-H και CAN-L σε λογικό σήμα RXCAN, το οποίο μπορεί να επεξεργαστεί ο MCP2515. Με τον τρόπο αυτό υλοποιείται η διεπαφή μεταξύ του επιπέδου φυσικής μετάδοσης (Physical Layer) και του επιπέδου ελέγχου πρόσβασης στο μέσο (Data Link Layer) του πρωτοκόλλου CAN [9].

3.3.3 Ακροδέκτες (Pins)

Ο MCP2515 διατίθεται συνήθως σε συσκευασία 18 ακροδεκτών. Οι σημαντικότεροι ακροδέκτες παρουσιάζονται στον ακόλουθο πίνακα.

Πίνακας 3.3: Ακροδέκτες ολοκληρωμένου MCP2515.

Ακροδέκτης	Περιγραφή
VDD	Τροφοδοσία από 3.3-5V
VSS	Γείωση

OSC1	Είσοδος ταλαντωτή
OSC2	Έξοδος ταλαντωτή
RESET	Επαναφορά συσκευής
CS	Chip Select για τη διεπαφή SPI
SCK	Ρολόι SPI
SI	Είσοδος δεδομένων SPI (MOSI)
SO	Έξοδος δεδομένων SPI (MISO)
INT	Έξοδος διακοπών
TXCAN	Έξοδος προς τον CAN transceiver
RXCAN	Είσοδος από τον CAN transceiver
CLKOUT	Έξοδος ρολογιού
RX1BF	Ένδειξη κατάστασης Receive Buffer 1
TX0RTS	Αίτηση μετάδοσης για Buffer 0
TX1RTS	Αίτηση μετάδοσης για Buffer 1
TX2RTS	Αίτηση μετάδοσης για Buffer 2

Οι ακροδέκτες TXCAN και RXCAN δεν συνδέονται απευθείας στο δίκτυο CAN αλλά σε εξωτερικό transceiver, ο οποίος αναλαμβάνει τη μετατροπή των λογικών επιπέδων σε διαφορεικά σήματα CAN-H και CAN-L [9].

3.3.4 Διεπαφή SPI

Η επικοινωνία μεταξύ του MCP2515 και του μικροελεγκτή βασίζεται στη σειριακή διεπαφή SPI, η οποία λειτουργεί σε αρχιτεκτονική master-slave. Ο μικροελεγκτής λειτουργεί ως master και ο MCP2515 ως slave. Η επικοινωνία πραγματοποιείται μέσω τεσσάρων βασικών γραμμών:

- CS (Chip Select)
- SCK (Serial Clock)
- SI (Serial Input)
- SO (Serial Output)

Μέσω ειδικών εντολών SPI ο μικροελεγκτής μπορεί να:

- Διαβάζει καταχωρητές του MCP2515.

- Τροποποιεί παραμέτρους λειτουργίας.
- Φορτώνει δεδομένα προς μετάδοση.
- Ανακτά ληφθέντα μηνύματα.
- Διαχειρίζεται διακοπές και καταστάσεις σφάλματος.

Η χρήση του SPI επιτρέπει υψηλές ταχύτητες επικοινωνίας και εύκολη διασύνδεση με πλατφόρμες όπως οι Arduino, STMicroelectronics STM32, Raspberry Pi Foundation Raspberry Pi και ESP32 [9].

3.3.5 Φίλτρα και μάσκες υλικού

Ένα από τα σημαντικότερα χαρακτηριστικά του MCP2515 είναι η δυνατότητα φιλτραρίσματος των εισερχόμενων μηνυμάτων σε επίπεδο υλικού (hardware filtering).

Ο ελεγκτής διαθέτει:

- 2 μάσκες αποδοχής (Acceptance Masks).
- 6 φίλτρα αποδοχής (Acceptance Filters).

Οι μάσκες καθορίζουν ποια bits του αναγνωριστικού (Identifier) θα ληφθούν υπόψη κατά τη σύγκριση, ενώ τα φίλτρα ορίζουν τις επιθυμητές τιμές των συγκεκριμένων bits.

Η διαδικασία αυτή επιτρέπει:

- Μείωση του φόρτου επεξεργασίας του μικροελεγκτή.
- Απόρριψη μη επιθυμητών μηνυμάτων πριν αποθηκευτούν.
- Βελτίωση της συνολικής απόδοσης του συστήματος.

Για παράδειγμα, σε ένα δίκτυο με εκατοντάδες διαφορετικά μηνύματα, ο μικροελεγκτής μπορεί να λαμβάνει μόνο όσα αφορούν συγκεκριμένους αισθητήρες ή υποσυστήματα.

Ένα από τα σημαντικότερα χαρακτηριστικά του MCP2515 είναι η δυνατότητα φιλτραρίσματος των εισερχόμενων μηνυμάτων σε επίπεδο υλικού (hardware filtering).

Ο ελεγκτής διαθέτει:

- 2 μάσκες αποδοχής (Acceptance Masks).
- 6 φίλτρα αποδοχής (Acceptance Filters).

Οι μάσκες καθορίζουν ποια bits του αναγνωριστικού (Identifier) θα ληφθούν υπόψη κατά τη σύγκριση, ενώ τα φίλτρα ορίζουν τις επιθυμητές τιμές των συγκεκριμένων bits.

Η διαδικασία αυτή επιτρέπει:

- Μείωση του φόρτου επεξεργασίας του μικροελεγκτή.
- Απόρριψη μη επιθυμητών μηνυμάτων πριν αποθηκευτούν.
- Βελτίωση της συνολικής απόδοσης του συστήματος.

Για παράδειγμα, σε ένα δίκτυο με εκατοντάδες διαφορετικά μηνύματα, ο μικροελεγκτής μπορεί να λαμβάνει μόνο όσα αφορούν συγκεκριμένους αισθητήρες ή υποσυστήματα [9].

3.3.6 Περιορισμοί

Παρά τα πλεονεκτήματά του, ο MCP2515 παρουσιάζει ορισμένους περιορισμούς που πρέπει να λαμβάνονται υπόψη κατά τον σχεδιασμό ενός συστήματος. Η ανταλλαγή δεδομένων μέσω SPI εισάγει επιπλέον καθυστέρηση σε σχέση με μικροελεγκτές που διαθέτουν ενσωματωμένο CAN controller. Η καθυστέρηση αυτή μπορεί να είναι σημαντική σε εφαρμογές πραγματικού χρόνου με αυστηρές απαιτήσεις χρονισμού, ιδιαίτερα όταν υπάρχει μεγάλος όγκος δεδομένων. Η περιορισμένη μνήμη προκαλεί υψηλό φόρτο κυκλοφορίας ενδέχεται να προκληθούν απώλειες μηνυμάτων εάν οι buffers γεμίσουν πριν αναγνωστούν από τον μικροελεγκτή.

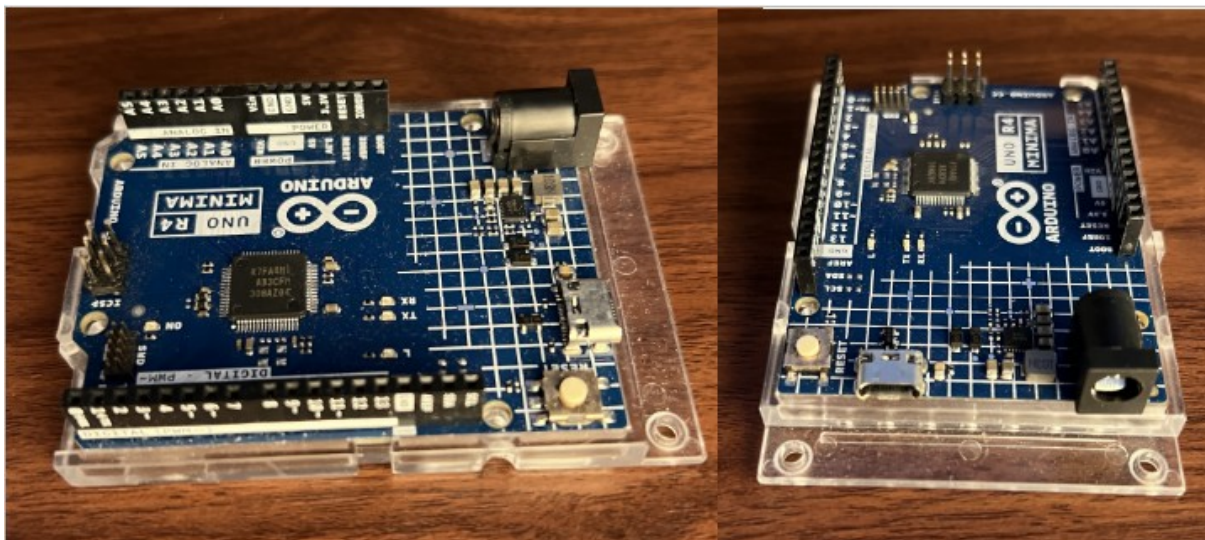
Η συσκευή διαθέτει μόνο:

- 3 transmit buffers.
- 2 receive buffers.

Ο MCP2515 υποστηρίζει αποκλειστικά το πρότυπο CAN 2.0A/B και δεν είναι συμβατός με το νεότερο πρωτόκολλο CAN FD (Flexible Data Rate), το οποίο επιτρέπει μεγαλύτερα πλαίσια δεδομένων και υψηλότερους ρυθμούς μετάδοσης [9].

3.4 Arduino Uno Minima R4

Κατά την αρχική φάση του έργου χρησιμοποιήθηκε το Arduino Uno R4 Minima λόγω της άμεσης διαθεσιμότητάς του (σχήμα 3.4). Τα τεχνικά χαρακτηριστικά της πλατφόρμας κρίθηκαν επαρκή για την ανάπτυξη και αξιολόγηση της πρώτης προσέγγισης του συστήματος, ενώ η εκτεταμένη υποστήριξη από διαθέσιμες βιβλιοθήκες και εργαλεία ανάπτυξης διευκόλυνε σημαντικά τη διαδικασία υλοποίησης. Η χρήση του περιορίστηκε κυρίως στην ανάγνωση και την αρχική επεξεργασία των δεδομένων. Έπειτα, κρίθηκε απαραίτητη η αντικατάστασή του από μικροελεγκτή με μεγαλύτερες δυνατότητες και περισσότερους διαθέσιμους πόρους, καθώς προσέφερε μεγαλύτερη ευελιξία και καλύτερες προοπτικές για μελλοντικές επεκτάσεις του συστήματος.



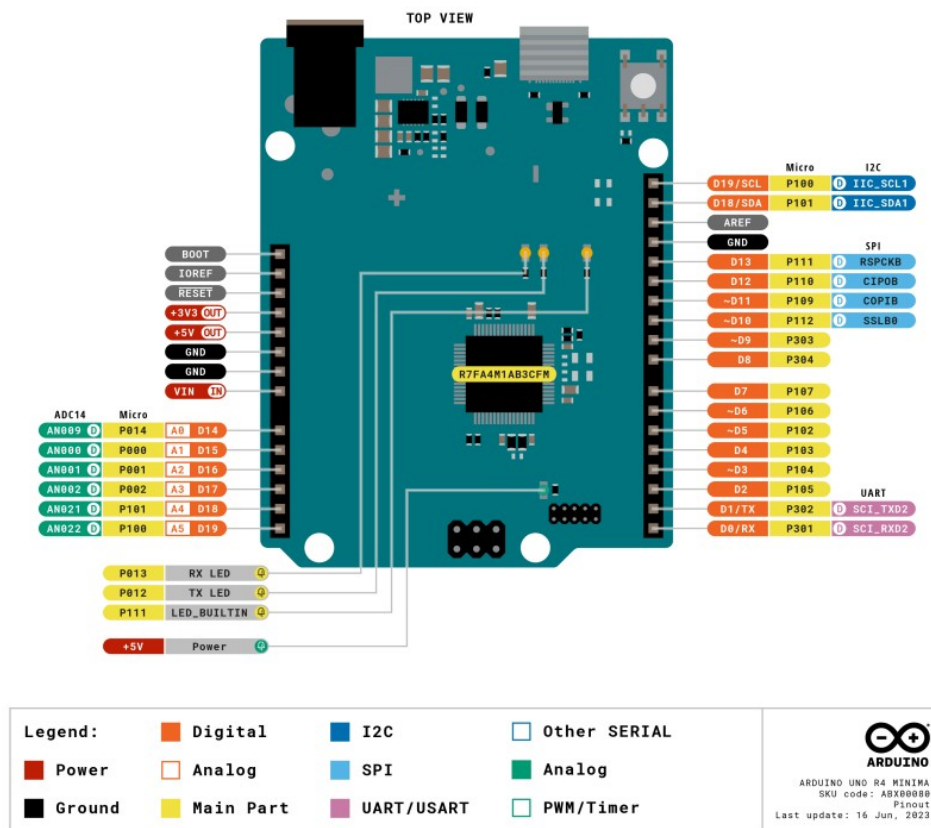
Σχήμα 3.4: Πλακέτα ανάπτυξης Arduino Uno R4 Minima βασισμένη στον μικροελεγκτή Renesas RA4M1.

3.4.1 Τεχνικά χαρακτηριστικά

Η πλαφόρμας Arduino Uno R4 Minima αποτελεί τη νεότερη εξέλιξη της δημοφιλούς οικογένειας Arduino Uno και σηματοδοτεί τη μετάβαση από την αρχιτεκτονική 8-bit AVR σε μια σύγχρονη πλατφόρμα 32-bit βασισμένη στον μικροελεγκτή Renesas RA4M1. Η πλακέτα διατηρεί τη συμβατότητα με τη μορφή (form factor) και τα περισσότερα shields του Arduino Uno, ενώ προσφέρει σημαντικά αυξημένες επιδόσεις, μεγαλύτερη μνήμη και προηγμένα περιφερειακά.

Ο πυρήνας του Arduino Uno R4 Minima είναι ο μικροελεγκτής Renesas RA4M1 με αρχιτεκτονική Arm Cortex-M4 και συχνότητα λειτουργίας 48 MHz. Η πλατφόρμα διαθέτει 256 kB μνήμης Flash, 32 kB SRAM και 8 kB μνήμης EEPROM, προσφέροντας σημαντικά αυξημένους πόρους σε σχέση με το Arduino Uno R3. Επιπλέον, λειτουργεί στα 5V, διατηρώντας τη συμβατότητα με υπάρχοντα ηλεκτρονικά κυκλώματα και περιφερειακές συσκευές.

Η πλακέτα περιλαμβάνει 14 ψηφιακές εισόδους/εξόδους, 6 αναλογικές εισόδους με δυνατότητα ανάλυσης έως 14-bit από αναλογικό σε ψηφιακό (ADC), ενσωματωμένο μετατροπέα ψηφιακού σε αναλογικό σήμα (DAC) με δυνατότητα ανάλυσης 12-bit. Υποστηρίζει επίσης πρωτόκολλα επικοινωνίας όπως SPI, I2C, UART και άλλα. Παράλληλα, μπορεί να τροφοδοτηθεί μέσω USB-C ή από εξωτερική πηγή τάσης έως 24 V μέσω του ακροδέκτη VIN (σχήμα 3.5) [10].



Σχήμα 3.5: Διάγραμμα ακροδεκτών (pinout) της πλακέτας ανάπτυξης Arduino Uno R4 Minima. Πηγή: [10]

3.4.2 Το πρωτόκολλο SPI στο Arduino UNO R4 Minima

Το SPI αποτελεί ένα σύγχρονο σειριακό πρωτόκολλο επικοινωνίας τύπου master-slave, το οποίο επιτρέπει την ταυτόχρονη αποστολή και λήψη δεδομένων (full-duplex communication). Στο Arduino UNO R4 Minima, η διεπαφή SPI υλοποιείται μέσω τεσσάρων βασικών ακροδεκτών:

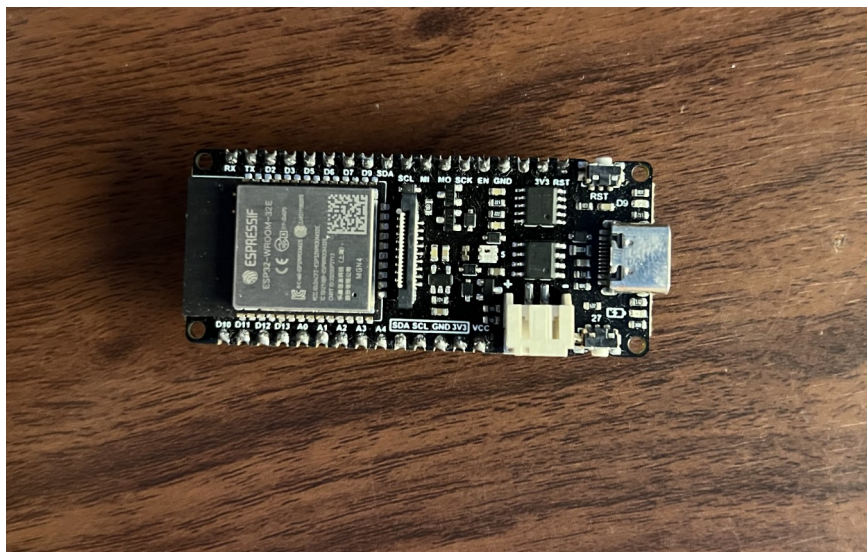
- D13: SCK (Serial Clock)
- D12: COPI (Controller Out Peripheral In)
- D11: CIPO (Controller In Peripheral Out)
- D10: CS (Chip Select)

Η λειτουργία του SPI βασίζεται στη δημιουργία ενός σήματος χρονισμού από τη συσκευή master, το οποίο συγχρονίζει τη μεταφορά δεδομένων προς και από τις συνδεδεμένες περιφερειακές συσκευές. Η γραμμή CS χρησιμοποιείται για την επιλογή της επιθυμητής περιφερειακής μονάδας, επιτρέποντας τη σύνδεση πολλαπλών συσκευών στο ίδιο δίαυλο επικοινωνίας.

Η πλακέτα διαθέτει 14 ψηφιακές εισόδους/εξόδους, 6 αναλογικές εισόδους, υποστήριξη PWM, ενσωματωμένο DAC (Digital-to-Analog Converter), RTC (Real-Time Clock) και δυνατότητα HID (Human Interface Device), επιτρέποντας την προσομοίωση συσκευών όπως πληκτρολόγια και ποντίκια USB [10].

3.5 Firebeetle 2 ESP32-E

Η επιλογή μετάβασης από το Arduino Uno R4 Minima στο ESP32-E FireBeetle 2 (σχήμα 3.6) πραγματοποιήθηκε με γνώμονα τις αυξημένες απαιτήσεις του συστήματος σε υπολογιστικούς πόρους και δυνατότητες επικοινωνίας. Συγκεκριμένα, η ανάγκη για αποδοτικότερη διαχείριση ροών δεδομένων και η προοπτική ασύρματης μετάδοσης (wireless telemetry) ανέδειξαν τη σημασία της ενσωματωμένης συνδεσιμότητας WiFi και Bluetooth του ESP32, η οποία απουσιάζει από την έκδοση Minima. Παράλληλα, η υψηλότερη συχνότητα λειτουργίας και η μεγαλύτερη επεξεργαστική ισχύς του ESP32 προσφέρουν βελτιωμένη απόκριση σε εφαρμογές με αυξημένο υπολογιστικό φόρτο και περιορισμένο χρόνο επεξεργασίας, συμβάλλοντας έτσι στη σταθερότητα και στην αποδοτικότητα του συνολικού συστήματος [11].



Σχήμα 3.6: Πλακέτα ανάπτυξης FireBeetle 2 ESP32-E, που χρησιμοποιήθηκε για την υλοποίηση του συστήματος.

3.5.1 Τεχνικά Χαρακτηριστικά

Ο μικροελεγκτής βασίζεται σε 32-bit αρχιτεκτονική Xtensa dual-core LX6, με μέγιστη συχνότητα λειτουργίας έως 240 MHz. Διαθέτει 4 MB Flash μνήμη, 520 KB SRAM και 448 KB ROM. Η ονομαστική τάση λειτουργίας είναι 3.3V, ενώ η τροφοδοσία μπορεί να πραγματοποιηθεί μέσω θύρας USB-C ή μέσω του pin VCC με εύρος 3.3–5V. Επιπλέον, υποστηρίζεται τροφοδοσία από μπαταρία λιθίου 3.7–4.2V μέσω συνδετήρα PH2.0. Η χαμηλή ενεργειακή κατανάλωση καθιστά τη συσκευή κατάλληλη για εφαρμογές Internet of Things (IoT).

για συνεχή ροή δεδομένων και περιφερειακές συσκευές υψηλότερου bandwidth, ενώ το BLE προσανατολίζεται σε εφαρμογές χαμηλής κατανάλωσης ενέργειας, όπως αισθητήρες και συσκευές IoT. Η αρχιτεκτονική BLE επιτρέπει σημαντικά μειωμένη κατανάλωση ισχύος μέσω περιοδικής μετάδοσης δεδομένων και λειτουργίας σε καταστάσεις ύπνου (sleep modes), καθιστώντας το κατάλληλο για ενεργειακά αποδοτικά ενσωματωμένα συστήματα [11].

3.5.3 Πρωτόκολλο SPI

Το ESP32 ενσωματώνει SPI (Serial Peripheral Interface), οι οποίοι υποστηρίζουν σύγχρονη, υψηλής ταχύτητας σειριακή επικοινωνία σε αρχιτεκτονική master-slave με πλήρη αμφίδρομη (full-duplex) λειτουργία. Η διεπαφή αξιοποιείται ευρέως για τη διασύνδεση με εξωτερικές μνήμες, αισθητήρες και περιφερειακές μονάδες, μέσω των γραμμών MOSI, MISO, SCK και SCL [11].

Πίνακας 3.4: Ακροδέκτες SPI του ESP32

Σήμα	GPIO	Περιγραφή
MOSI	GPIO 23	Master Out Slave In
MISO	GPIO 19	Master In Slave Out
SCK	GPIO 18	Serial Clock
SCL	GPIO 22	Chip Select

3.5.4 Two-Wire Automotive Interface

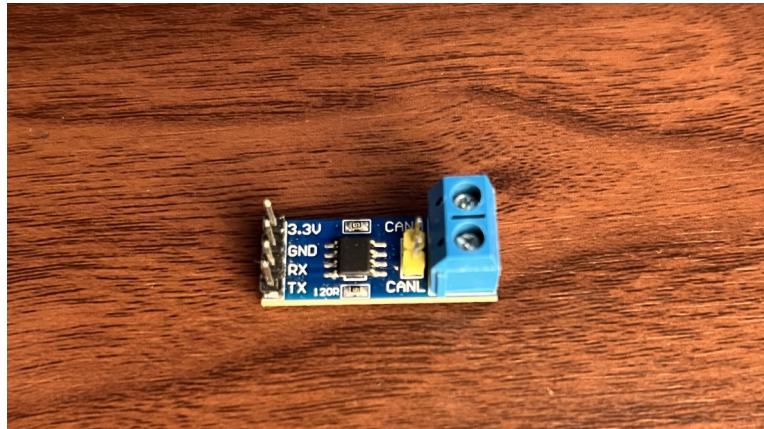
Το ESP32 υποστηρίζει άμεσα το πρωτόκολλο CAN (native CAN) και είναι γνωστό ως Two-Wire Automotive Interface (TWAI). Η σύνδεση πραγματοποιείται με την χρήση των ακροδεκτών RX & TX, η αρχιτεκτονική αυτή επιτρέπει την υλοποίηση επικοινωνίας CAN χωρίς την ανάγκη χρήσης εξωτερικού ελεγκτή πρωτοκόλλου (CAN transceiver). Ωστόσο όμως δεν βασίζεται στο πρωτόκολλο UART. Το ESP32 είναι συμβατός με το πρότυπο CAN 2.0B και υποστηρίζει ταχύτητες έως 1 Mbps και δεν είναι συμβατό με το πρότυπο CAN FD [11].

Πίνακας 3.5: Ακροδέκτες RX & TX του ESP32

Σήμα TWAI	GPIO	Περιγραφή
TX	GPIO 5	Σήμα μετάδοσης CAN
RX	GPIO 4	Σήμα λήψης CAN

3.6 CAN Transceiver SN65HVD230

Η διερεύνηση μίας δεύτερης, εναλλακτικής αρχιτεκτονικής οδήγησε στην επιλογή του CAN Transceiver SN65HVD230 (σχήμα 3.8). Η συγκεκριμένη λύση επιλέχθηκε λόγω της δυνατότητας άμεσης διασύνδεσης με τον μικροελεγκτή, χωρίς την παρεμβολή επιπρόσθετων κυκλωμάτων μετατροπής ή τη χρήση ενδιάμεσων πρωτοκόλλων επικοινωνίας, όπως το SPI. Η προσέγγιση αυτή εκτιμήθηκε ότι θα συνέβαλλε στη μείωση της καθυστέρησης επικοινωνίας, καθώς και στην απλούστευση της συνολικής αρχιτεκτονικής του συστήματος.



Σχήμα 3.8: Πλακέτα του SN65HVD230 CAN Bus Transceiver για διασύνδεση μικροελεγκτών με δίκτυα CAN

Ο SN65HVD230 αποτελεί έναν πομποδέκτη (CAN Transceiver) που χρησιμοποιείται για τη φυσική διασύνδεση μικροελεγκτών με δίκτυα CAN υψηλής ταχύτητας. Είναι συμβατός με το πρότυπο ISO 11898-2 και υποστηρίζει ρυθμούς μετάδοσης δεδομένων έως 1 Mbps. Ένα από τα βασικά χαρακτηριστικά του είναι η λειτουργία του με τάση τροφοδοσίας 3,3 V, γεγονός που επιτρέπει την απευθείας σύνδεσή του με σύγχρονους μικροελεγκτές χαμηλής τάσης, χωρίς την ανάγκη πρόσθετων κυκλωμάτων προσαρμογής λογικών επιπέδων τάσης.

Επιπλέον, ο SN65HVD230 διαθέτει ενσωματωμένους μηχανισμούς προστασίας έναντι ηλεκτροστατικών εκφορτίσεων (ESD), θερμικής υπερφόρτωσης και σφαλμάτων στη γραμμή επικοινωνίας, ενισχύοντας την αξιοπιστία και τη σταθερότητα της λειτουργίας του. Η σύνδεσή του με το δίκτυο CAN πραγματοποιείται μέσω των ακροδεκτών CANH και CANL, οι οποίοι συνδέονται απευθείας στον δίαυλο επικοινωνίας του οχήματος. Για την τροφοδοσία της διάταξης χρησιμοποιούνται οι ακροδέκτες VCC και GND, ενώ η επικοινωνία με τον μικροελεγκτή επιτυγχάνεται μέσω των ακροδεκτών TXD και RXD, οι οποίοι αντιστοιχούν στις γραμμές μετάδοσης και λήψης δεδομένων αντίστοιχα.

Η επιλογή του SN65HVD230 συνέβαλε στη δημιουργία μίας απλούστερης και πιο αποδοτικής αρχιτεκτονικής, μειώνοντας τον αριθμό των απαιτούμενων εξαρτημάτων και επιτρέποντας την άμεση πρόσβαση του μικροελεγκτή στο δίκτυο CAN του οχήματος.

Πίνακας 3.6: Σύνδεση ακροδεκτών SN65HVD230 στο σύστημα

Ακροδέκτης	Περιγραφή	Σύνδεση
VCC	Τροφοδοσία του κυκλώματος (3,3 V)	Τροφοδοτικό συστήματος
GND	Γείωση	Κοινή γείωση συστήματος
TXD	Είσοδος δεδομένων προς μετάδοση από τον μικροελεγκτή	Ακροδέκτης TX του μικροελεγκτή
RXD	Έξοδος δεδομένων που λαμβάνονται από το δίκτυο CAN	Ακροδέκτης RX του μικροελεγκτή
CANH	Γραμμή υψηλού δυναμικού του διαφορικού διαύλου CAN	Δίαυλος CAN του οχήματος
CANL	Γραμμή χαμηλού δυναμικού του διαφορικού διαύλου CAN	Δίαυλος CAN του οχήματος

Από τους παραπάνω ακροδέκτες, οι CANH και CANL αποτελούν το φυσικό μέσο σύνδεσης με τον δίαυλο CAN, ενώ οι TXD και RXD χρησιμοποιούνται για την ανταλλαγή δεδομένων με τον μικροελεγκτή. Οι ακροδέκτες VCC και GND εξασφαλίζουν την τροφοδοσία της διάταξης [12].

3.7 Servo Motor DS5180

Για την υλοποίηση του συστήματος ενεργοποίησης επιλέχθηκαν δύο σερβοκινητήρες DS5180 SSG (σχήμα 3.9). Συγκεκριμένα, χρησιμοποιούνται δύο σερβοκινητήρες σε κάθε πλευρά του μηχανισμού, με ονομαστική ροπή λειτουργίας έως 80 kg/cm ανά σερβοκινητήρα. Η διάταξη αυτή επιλέχθηκε ώστε να αυξηθεί η διαθέσιμη ροπή και η αξιοπιστία του συστήματος.



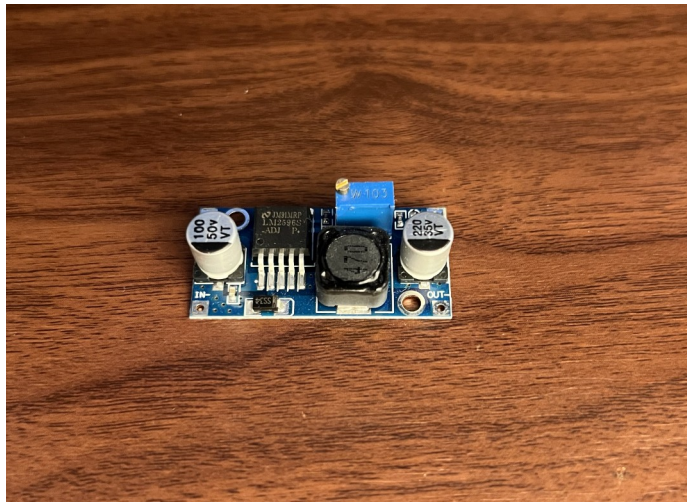
Σχήμα 3.9: Κινητήρας Servo DS5180 που χρησιμοποιήθηκε για την υλοποίηση της εργασίας.

Η επιλογή των σερβοκινητήρων πραγματοποιήθηκε λαμβάνοντας υπόψη τις απαιτήσεις του ηλεκτρονικού συστήματος ελέγχου, την ακρίβεια τοποθέτησης, τον χρόνο απόκρισης και την ευκολία διασύνδεσης με τον μικροελεγκτή. Σε σύγκριση με εναλλακτικές λύσεις, όπως πνευματικά ή υδραυλικά συστήματα ενεργοποίησης, οι σερβοκινητήρες προσφέρουν απλούστερη υλοποίηση, ακριβέστερο έλεγχο θέσης και μειωμένες απαιτήσεις σε βοηθητικό εξοπλισμό.

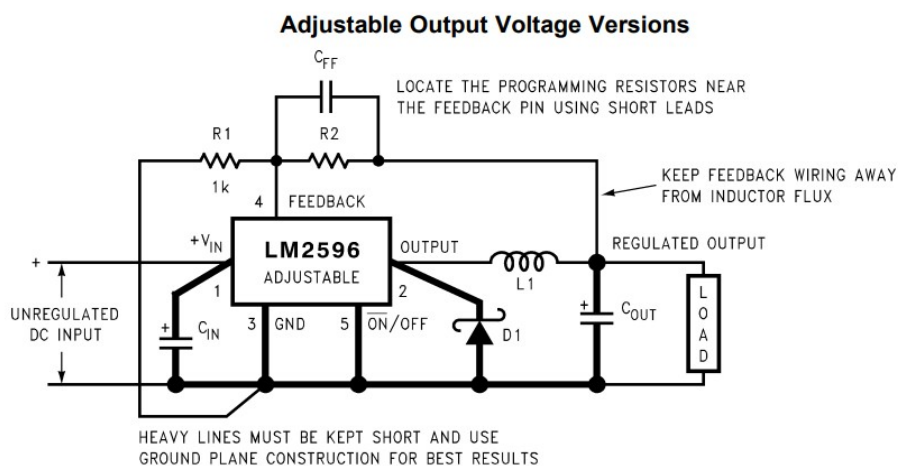
Οι σερβοκινητήρες DS5180 υποστηρίζουν μέγιστη γωνία περιστροφής έως 270° , ενώ το εύρος τάσης λειτουργίας τους είναι από 6V έως 8,4V. Η μέγιστη παραγόμενη ροπή εξαρτάται από την τάση τροφοδοσίας, όπως και η κατανάλωση ρεύματος. Στον Πίνακα παρουσιάζονται τα τεχνικά χαρακτηριστικά του κινητήρα και η αντιστοίχιση τάσης τροφοδοσίας, διαθέσιμης ροπής και μέγιστου ρεύματος λειτουργίας [13].

3.8 Step-Down Converter LM2596

Η τροφοδοσία του συστήματος πραγματοποιείται από την ηλεκτρική εγκατάσταση του οχήματος, η οποία παρέχει ονομαστική τάση 12V. Ωστόσο, ο μικροελεγκτής και οι σερβοκινητήρες απαιτούν χαμηλότερα επίπεδα τάσης για την ασφαλή και ορθή λειτουργία τους. Για τον λόγο αυτό επιλέχθηκε η χρήση του μετατροπέα υποβιβασμού τάσης (buck converter) LM2596 (σχήμα 3.10), ο οποίος επιτρέπει τη μετατροπή της τάσης εισόδου στην απαιτούμενη τάση εξόδου με υψηλό βαθμό απόδοσης (σχήμα 3.11).



Σχήμα 3.10: Step-Down converter που χρησιμοποιήθηκε για την υλοποίηση του συστήματος.



Σχήμα 3.11: Ενδεικτικό κύκλωμα προσαρμόσιμης εξόδου LM2596 Πηγή: [14]

Ο LM2596 υποστηρίζει εύρος τάσης εισόδου από 4,5V έως 40V, ενώ η μέγιστη συνεχής ένταση ρεύματος εξόδου ανέρχεται στα 3A. Το ολοκληρωμένο κύκλωμα βασίζεται σε τεχνολογία διακοπτικής λειτουργίας (switching regulation), γεγονός που του επιτρέπει να επιτυγχάνει υψηλή ενεργειακή απόδοση, η οποία σύμφωνα με τα τεχνικά χαρακτηριστικά του κατασκευαστή μπορεί να προσεγγίσει το 80–90%, ανάλογα με τις συνθήκες λειτουργίας, το φορτίο και τη διαφορά μεταξύ τάσης εισόδου και εξόδου.

Η χρήση διακοπτικού μετατροπέα αντί γραμμικού σταθεροποιητή μειώνει σημαντικά τις απώλειες ισχύος υπό μορφή θερμότητας, βελτιώνοντας τη συνολική ενεργειακή απόδοση του συστήματος και περιορίζοντας τις θερμικές καταπονήσεις των ηλεκτρονικών εξαρτημάτων.

Παρά τα πλεονεκτήματα αυτά, η μέγιστη δυνατότητα παροχής ρεύματος των 3 A αποτελεί περιοριστικό παράγοντα σε περιπτώσεις αυξημένου φορτίου, ιδιαίτερα όταν λειτουργούν ταυτόχρονα πολλαπλοί σερβοκινητήρες ή άλλες ενεργοβόρες συσκευές. Σε τέτοιες συνθήκες ενδέχεται να απαιτηθεί η χρήση μετατροπέα μεγαλύτερης ισχύος ή η υλοποίηση ξεχωριστών γραμμών τροφοδοσίας για τον διαμοιρασμό του φορτίου.

Στο πλαίσιο της παρούσας εργασίας και των απαιτήσεων του υπό ανάπτυξη συστήματος, ο μετατροπέας LM2596 κάλυψε επιτυχώς τις ενεργειακές ανάγκες της διάταξης κατά τη φάση των προσομοιώσεων και των εργαστηριακών δοκιμών, παρέχοντας σταθερή τάση εξόδου και αξιόπιστη λειτουργία των επιμέρους υποσυστημάτων [14].

Κεφάλαιο 4ο: Λογισμικό Εργασίας

4.1 Εισαγωγή

Το παρόν κεφάλαιο παρουσιάζει το περιβάλλον ανάπτυξης λογισμικού που χρησιμοποιήθηκε για την υλοποίηση του συστήματος. Η επιλογή των εργαλείων πραγματοποιήθηκε με βάση συγκεκριμένα κριτήρια, όπως η συμβατότητα με τις πλατφόρμες υλικού, η διαθεσιμότητα βιβλιοθηκών για επικοινωνία CAN bus και η ευκολία ανάπτυξης και αποσφαλμάτωσης κώδικα.

Η ανάπτυξη του συστήματος ακολούθησε εξελικτική πορεία ως προς τα εργαλεία που χρησιμοποιήθηκαν. Αρχικά χρησιμοποιήθηκε το Arduino IDE για την ανάπτυξη και δοκιμή του κώδικα καταγραφής CAN frames. Στη συνέχεια, λόγω των περιορισμών του Arduino IDE σε θέματα διαχείρισης μεγάλου όγκου δεδομένων, η ανάπτυξη μεταφέρθηκε σε περιβάλλον Visual Studio Code με PlatformIO, το οποίο προσφέρει σημαντικά πιο προηγμένες δυνατότητες. Παράλληλα αναπτύχθηκαν Python scripts για την ανάγνωση, επεξεργασία και οπτικοποίηση των δεδομένων του CAN bus, με χρήση γραφικού περιβάλλοντος Tkinter για την αποκωδικοποίηση των frames.

Στις επόμενες ενότητες παρουσιάζεται αναλυτικά κάθε εργαλείο, με έμφαση στα χαρακτηριστικά του, στον ρόλο που επιτέλεσε στην ανάπτυξη του συστήματος και στους λόγους που οδήγησαν στην επιλογή του.

4.2 Arduino IDE

Το Arduino IDE (Integrated Development Environment) αποτελεί το επίσημο περιβάλλον ανάπτυξης της πλατφόρμας Arduino, το οποίο αναπτύχθηκε από την Arduino LLC και κυκλοφόρησε για πρώτη φορά το 2005 παράλληλα με την πρώτη πλακέτα Arduino. Από τότε έχει εξελιχθεί σε ένα από τα πιο διαδεδομένα εργαλεία ανάπτυξης στον χώρο των embedded systems και του physical computing, με εκατομμύρια χρήστες παγκοσμίως. Διατίθεται ελεύθερα ως λογισμικό ανοιχτού κώδικα και είναι πλήρως συμβατό με Windows, macOS και Linux.

4.2.1 Χαρακτηριστικά και δυνατότητες

Το Arduino IDE βασίζεται σε γλώσσα προγραμματισμού C/C++ με απλοποιημένο συντακτικό, γεγονός που το καθιστά ιδιαίτερα προσίτο ακόμα και σε χρήστες χωρίς εκτεταμένη εμπειρία σε embedded προγραμματισμό. Τα βασικά χαρακτηριστικά του περιλαμβάνουν:

- Ενσωματωμένο Serial Monitor για παρακολούθηση δεδομένων σε πραγματικό χρόνο μέσω σειριακής θύρας
- Library Manager για εύκολη εγκατάσταση βιβλιοθηκών τρίτων κατασκευαστών
- Υποστήριξη μεγάλου αριθμού πλατφορμών (Arduino, ESP32, Teensy κτλ.)
- Ενσωματωμένος compiler και uploader για άμεση φόρτωση κώδικα
- Διαθέσιμο για Windows, macOS και Linux

4.2.2 Χρήση στο Παρόν Σύστημα

Στο αρχικό στάδιο ανάπτυξης το Arduino IDE χρησιμοποιήθηκε για τη συγγραφή και φόρτωση του κώδικα καταγραφής CAN frames στο Arduino UNO R4 Minima. Μέσω του Serial Monitor πραγματοποιήθηκε η πρώτη παρακολούθηση της κυκλοφορίας του δικτύου CAN bus του οχήματος, επιβεβαιώνοντας την ορθή λειτουργία του MCP2515 και την επιτυχή λήψη frames.

4.2.3 Περιορισμοί

Παρά την ευκολία χρήσης του, το Arduino IDE παρουσίασε σημαντικούς περιορισμούς κατά την καταγραφή μεγάλου όγκου CAN frames:

- Το Serial Monitor δεν επιτρέπει αξιόπιστη αποθήκευση δεδομένων σε αρχείο
- Η εκτύπωση μεγάλου αριθμού Serial.print() επιβραδύνει τον κώδικα με αποτέλεσμα απώλεια frames
- Απουσία εργαλείων για αυτοματοποιημένη επεξεργασία και ανάλυση δεδομένων
- Δυσκολία διαχείρισης πολλαπλών αρχείων κώδικα σε μεγαλύτερα projects

Οι περιορισμοί αυτοί κατέστησαν αναγκαία τη μετάβαση σε πιο προηγμένο περιβάλλον ανάπτυξης, όπως παρουσιάζεται στην επόμενη ενότητα.

4.3 Visual Studio Code

Το Visual Studio Code (VSCode) αποτελεί ένα ελεύθερο, ανοιχτού κώδικα πρόγραμμα επεξεργασίας κώδικα (source code editor) που αναπτύχθηκε από τη Microsoft και κυκλοφόρησε για πρώτη φορά το 2015. Παρά τη σχετικά σύντομη ιστορία του, κατέστη γρήγορα το πιο δημοφιλές εργαλείο ανάπτυξης λογισμικού παγκοσμίως, σύμφωνα με την ετήσια έρευνα του Stack Overflow Developer Survey. Διατίθεται ελεύθερα για Windows, macOS και Linux και υποστηρίζει πρακτικά οποιαδήποτε γλώσσα προγραμματισμού μέσω του συστήματος επεκτάσεων (extensions) που διαθέτει.

4.3.1 Αρχιτεκτονική

Το VSCode βασίζεται στο framework Electron, το οποίο επιτρέπει την ανάπτυξη desktop εφαρμογών με τεχνολογίες web (HTML, CSS, JavaScript). Ο editor χρησιμοποιεί τον Monaco Editor ως πυρήνα επεξεργασίας κειμένου — τον ίδιο engine που χρησιμοποιεί και το Arduino IDE 2.x. Η αρχιτεκτονική επεκτάσεων του VSCode επιτρέπει στους χρήστες να προσαρμόζουν πλήρως το περιβάλλον ανάπτυξης ανάλογα με τις ανάγκες τους, προσθέτοντας υποστήριξη για νέες γλώσσες, εργαλεία debugging, themes και integrations με εξωτερικές υπηρεσίες.

4.3.2 Βασικά χαρακτηριστικά

Πίνακας 4.1: Χαρακτηριστικά Visual Studio Code

Χαρακτηριστικό	Περιγραφή
Κατασκευαστής	Microsoft
Πρώτη κυκλοφορία	2015
Τρέχουσα έκδοση	1.x (ενημερώνεται μηνιαία)
Λειτουργικά συστήματα	Windows, macOS, Linux
Άδεια χρήσης	MIT (ανοιχτός κώδικας)
Γλώσσες προγραμματισμού	Όλες (μέσω extensions)

Τα βασικά εργαλεία που παρέχει περιλαμβάνουν:

- IntelliSense — αυτόματη συμπλήρωση κώδικα, ανίχνευση σφαλμάτων και προτάσεις κατά τη συγγραφή
- Integrated Terminal — ενσωματωμένο τερματικό για εκτέλεση εντολών χωρίς έξοδο από το περιβάλλον
- Git Integration — ενσωματωμένη υποστήριξη για έλεγχο εκδόσεων
- Debugger — προηγμένα εργαλεία αποσφαλμάτωσης για πολλαπλές γλώσσες
- Extensions Marketplace — πρόσβαση σε χιλιάδες επεκτάσεις για προσαρμογή του περιβάλλοντος
- Multi-language support — υποστήριξη C/C++, Python, JavaScript και δεκάδων άλλων γλωσσών στο ίδιο περιβάλλον

4.3.3 Επιλογή για το παρόν Σύστημα

Το VSCode επιλέχθηκε ως κύριο περιβάλλον ανάπτυξης για δύο λόγους. Πρώτον, μέσω της επέκτασης PlatformIO επιτρέπει την ανάπτυξη embedded κώδικα για Arduino και ESP32 στο ίδιο περιβάλλον με σημαντικά πιο προηγμένες δυνατότητες σε σχέση με το Arduino IDE. Δεύτερον, υποστηρίζει πλήρως την Python, επιτρέποντας την ανάπτυξη των scripts επεξεργασίας δεδομένων CAN bus στο ίδιο εργαλείο χωρίς εναλλαγή περιβάλλοντος.

4.4 PlatformIO

Το PlatformIO αποτελεί ένα ανοιχτού κώδικα οικοσύστημα ανάπτυξης για embedded systems, το οποίο κυκλοφόρησε για πρώτη φορά το 2014. Είναι διαθέσιμο κυρίως ως επέκταση (extension) του Visual Studio Code, μετατρέποντάς το σε πλήρες IDE για ανάπτυξη κώδικα μικροελεγκτών. Σε αντίθεση με το Arduino IDE, το οποίο σχεδιάστηκε με γνώμονα την απλότητα, το PlatformIO απευθύνεται σε πιο

απαιτητικές εφαρμογές και επαγγελματικά projects, προσφέροντας προηγμένα εργαλεία διαχείρισης κώδικα, βιβλιοθηκών και αποσφαλμάτωσης.

4.4.1 Αρχιτεκτονική και Λειτουργία

Το PlatformIO βασίζεται σε ένα σύστημα διαχείρισης πλατφορμών και βιβλιοθηκών που λειτουργεί ανεξάρτητα από το λειτουργικό σύστημα. Κάθε project ορίζεται από ένα αρχείο ρυθμίσεων `platformio.ini`, στο οποίο καθορίζονται:

- Η πλατφόρμα-στόχος (π.χ. Arduino, ESP32)
- Το framework ανάπτυξης (π.χ. Arduino framework)
- Οι βιβλιοθήκες που απαιτούνται
- Οι παράμετροι μεταγλώττισης και φόρτωσης

Με αυτόν τον τρόπο το project είναι πλήρως φορητό και αναπαραγωγίσιμο σε οποιοδήποτε σύστημα χωρίς χειροκίνητη εγκατάσταση εξαρτήσεων.

4.4.2 Βασικά Χαρακτηριστικά

Πίνακας 4.2: Χαρακτηριστικά PlatformIO

Χαρακτηριστικό	Περιγραφή
Πρώτη κυκλοφορία	2014
Λειτουργικά συστήματα	Windows, macOS, Linux
Άδεια χρήσης	Apache 2.0 (ανοιχτός κώδικας)
Υποστηριζόμενες πλατφόρμες	40+ (Arduino, ESP32, STM32, Teensy κτλ.)
Υποστηριζόμενα frameworks	Arduino, ESP-IDF, CMSIS, FreeRTOS κτλ.
Βιβλιοθήκες	10.000+ μέσω του PlatformIO Registry

Τα βασικά εργαλεία που παρέχει περιλαμβάνουν:

- Unified Debugger — ενσωματωμένο debugger για όλες τις υποστηριζόμενες πλατφόρμες
- Library Manager — αυτόματη εγκατάσταση και διαχείριση βιβλιοθηκών με έλεγχο εκδόσεων
- Serial Monitor — προηγμένο Serial Monitor με δυνατότητα αποθήκευσης δεδομένων
- Unit Testing — εργαλεία δοκιμών κώδικα απευθείας στον μικροελεγκτή
- Static Code Analysis — στατική ανάλυση κώδικα για εντοπισμό σφαλμάτων πριν τη μεταγλώττιση
- Multi-platform builds — ταυτόχρονη μεταγλώττιση για πολλαπλές πλατφόρμες

4.4.3 Σύγκριση με Arduino IDE

Ο παρακάτω πίνακας συνοψίζει τις βασικές διαφορές μεταξύ Arduino IDE και PlatformIO:

Πίνακας 4.3: Χαρακτηριστικά Arduino IDE

Χαρακτηριστικό	Arduino IDE	PlatformIO
Διαχείριση βιβλιοθηκών	Χειροκίνητη	Αυτόματη μέσω platformio.ini
IntelliSense	Περιορισμένο	Πλήρες
Debugging	Βασικό (IDE 2.x)	Προηγμένο
Πολλαπλές πλατφόρμες	Ναι	Ναι (40+)
Serial Monitor	Βασικό	Προηγμένο με logging
Κατάλληλο για	Αρχάριους	Προχωρημένα projects

4.4.4 Επιλογή για το Παρόν Σύστημα

Το PlatformIO επιλέχθηκε λόγω της δυνατότητας διαχείρισης και των δύο πλατφορμών του συστήματος — Arduino UNO R4 Minima και ESP32 FireBeetle — μέσα από το ίδιο περιβάλλον, καθώς και λόγω της αυτόματης διαχείρισης βιβλιοθηκών που απλοποίησε σημαντικά την ανάπτυξη.

4.5 Python

Η Python αποτελεί μια γλώσσα προγραμματισμού υψηλού επιπέδου, γενικής χρήσης, η οποία αναπτύχθηκε από τον Guido van Rossum και κυκλοφόρησε για πρώτη φορά το 1991. Σήμερα αποτελεί μία από τις πιο διαδεδομένες γλώσσες προγραμματισμού παγκοσμίως, με εφαρμογές σε τομείς όπως η επιστήμη δεδομένων, η τεχνητή νοημοσύνη, η ανάπτυξη web εφαρμογών και η αυτοματοποίηση διαδικασιών. Διατίθεται ελεύθερα ως ανοιχτού κώδικα λογισμικό και υποστηρίζεται από μια από τις μεγαλύτερες κοινότητες προγραμματιστών παγκοσμίως.

4.5.1 Βασικά Χαρακτηριστικά

Πίνακας 4.4: Χαρακτηριστικά Python

Χαρακτηριστικό	Περιγραφή
Δημιουργός	Guido van Rossum
Πρώτη κυκλοφορία	1991
Τρέχουσα έκδοση	Python 3.x

Άδεια χρήσης	PSF License (ανοιχτός κώδικας)
Λειτουργικά συστήματα	Windows, macOS, Linux
Paradigm	Object-oriented, functional, procedural

Τα βασικά χαρακτηριστικά της Python που την καθιστούν ιδιαίτερα κατάλληλη για εφαρμογές επεξεργασίας δεδομένων περιλαμβάνουν:

- Απλό και ευανάγνωστο συντακτικό — ο κώδικας είναι κατανοητός ακόμα και από μη ειδικούς
- Δυναμική τυποποίηση — δεν απαιτείται δήλωση τύπου μεταβλητών
- Πλούσιο οικοσύστημα βιβλιοθηκών — χιλιάδες βιβλιοθήκες διαθέσιμες μέσω του PyPI (Python Package Index)
- Διερμηνευόμενη γλώσσα — ο κώδικας εκτελείται άμεσα χωρίς μεταγλώττιση
- Cross-platform — ο ίδιος κώδικας τρέχει σε Windows, macOS και Linux χωρίς τροποποιήσεις

4.5.2 PySerial

Η PySerial αποτελεί βιβλιοθήκη της Python που παρέχει πρόσβαση σε σειριακές θύρες (serial ports) σε περιβάλλον Python. Αναπτύχθηκε από τον Chris Liechti και διατίθεται ελεύθερα μέσω του PyPI. Υποστηρίζει όλα τα κύρια λειτουργικά συστήματα και επιτρέπει την ανάγνωση και εγγραφή δεδομένων μέσω σειριακής θύρας με απλές εντολές Python.

Τα βασικά χαρακτηριστικά της PySerial περιλαμβάνουν:

- Υποστήριξη RS-232, USB-to-Serial και άλλων σειριακών διεπαφών
- Ρυθμιζόμενη ταχύτητα επικοινωνίας (baud rate)
- Δυνατότητα ασύγχρονης και σύγχρονης ανάγνωσης δεδομένων
- Υποστήριξη timeout για αξιόπιστη επικοινωνία
- Πλήρης έλεγχος παραμέτρων σειριακής επικοινωνίας (bits, parity, stop bits)

Η PySerial επιλέχθηκε λόγω της απλότητας χρήσης της και της πλήρους συμβατότητας με τον μικροελεγκτή Arduino UNO R4 Minima, επιτρέποντας την αξιόπιστη ανάγνωση των CAN frames που αποστέλλονται μέσω της σειριακής θύρας.

4.5.3 Tkinter

Η Tkinter αποτελεί την επίσημη βιβλιοθήκη γραφικών διεπαφών χρήστη (GUI) της Python και περιλαμβάνεται στην τυπική εγκατάσταση της γλώσσας, χωρίς να απαιτείται επιπλέον εγκατάσταση. Βασίζεται στη βιβλιοθήκη Tk που αναπτύχθηκε από τον John Ousterhout και παρέχει ένα σύνολο από γραφικά στοιχεία (widgets) για την κατασκευή παραθύρων, κουμπιών, πινάκων και άλλων στοιχείων διεπαφής.

Τα βασικά χαρακτηριστικά της Tkinter περιλαμβάνουν:

- Cross-platform — τα γραφικά στοιχεία εμφανίζονται σωστά σε Windows, macOS και Linux
- Ενσωματωμένη στην Python — δεν απαιτείται εξωτερική εγκατάσταση
- Πλούσιο σύνολο widgets — παράθυρα, κουμπιά, πίνακες, labels, entry fields κτλ.
- Event-driven προγραμματισμός — αντίδραση σε γεγονότα χρήστη όπως κλικ και πληκτρολόγηση
- Canvas widget — δυνατότητα σχεδίασης γραφικών και οπτικοποίησης δεδομένων

Η Tkinter επιλέχθηκε για την ανάπτυξη του γραφικού περιβάλλοντος αποκωδικοποίησης CAN frames λόγω της άμεσης διαθεσιμότητάς της στην Python και της ευκολίας κατασκευής πινάκων με δυναμική ενημέρωση δεδομένων σε πραγματικό χρόνο.

4.5.4 Επιλογή Python για το Παρόν Σύστημα

Η Python επιλέχθηκε ως γλώσσα ανάπτυξης των scripts επεξεργασίας δεδομένων λόγω της ευκολίας χρήσης της, της διαθεσιμότητας των βιβλιοθηκών PySerial και Tkinter και της δυνατότητας γρήγορης ανάπτυξης και δοκιμής κώδικα. Η δυνατότητα εξαγωγής δεδομένων σε αρχείο TXT για περαιτέρω ανάλυση σε Microsoft Excel, σε συνδυασμό με την οπτικοποίηση των CAN frames μέσω Tkinter, κατέστησαν την Python το ιδανικό εργαλείο για την ανάλυση και αποκωδικοποίηση των δεδομένων του δικτύου CAN bus.

4.6 Βιβλιοθήκες Embedded

Στο παρόν υποκεφάλαιο παρουσιάζονται οι βιβλιοθήκες που χρησιμοποιήθηκαν για την ανάπτυξη του embedded κώδικα του συστήματος. Κάθε βιβλιοθήκη επιλέχθηκε με βάση τη συμβατότητά της με τις πλατφόρμες υλικού, την ποιότητα της τεκμηρίωσής της και την ευρεία χρήση της από την κοινότητα.

4.6.1 MCP_CAN (coryjfowler)

Η βιβλιοθήκη MCP_CAN αναπτύχθηκε από τον Cory J. Fowler και αποτελεί μία από τις πιο διαδεδομένες βιβλιοθήκες για επικοινωνία με τον CAN controller MCP2515 σε περιβάλλον Arduino και ESP32. Διατίθεται ελεύθερα μέσω του GitHub και του PlatformIO Registry και υποστηρίζει πλήρως το πρότυπο CAN 2.0B, καλύπτοντας τόσο standard (11-bit) όσο και extended (29-bit) αναγνωριστικά.

Τα βασικά χαρακτηριστικά της βιβλιοθήκης περιλαμβάνουν:

- Αποστολή και λήψη standard και extended CAN frames
- Υποστήριξη hardware filters και masks του MCP2515
- Ρύθμιση bit rate επικοινωνίας (125 kbps έως 1 Mbps)
- Υποστήριξη interrupt-based λήψης frames
- Συμβατότητα με Arduino, ESP32 και άλλες πλατφόρμες
- Πλήρης υποστήριξη του SPI interface του MCP2515

Πίνακας 4.5: Χαρακτηριστικά βιβλιοθήκης MCP_CAN.

Χαρακτηριστικό	Περιγραφή
Δημιουργός	Cory J. Fowler
Άδεια χρήσης	MIT
Πλατφόρμες	Arduino, ESP32
Πρωτόκολλο	CAN 2.0B
Μέγιστη ταχύτητα	1 Mbps

4.6.2 Βιβλιοθήκη SPI

Η βιβλιοθήκη SPI.h αποτελεί την επίσημη βιβλιοθήκη του Arduino για επικοινωνία μέσω του πρωτοκόλλου SPI (Serial Peripheral Interface) και περιλαμβάνεται στην τυπική εγκατάσταση του Arduino IDE και PlatformIO. Το SPI είναι ένα σύγχρονο πρωτόκολλο σειριακής επικοινωνίας που χρησιμοποιείται ευρέως για την επικοινωνία μικροελεγκτών με περιφερειακά κυκλώματα όπως ο MCP2515.

Τα βασικά χαρακτηριστικά της βιβλιοθήκης περιλαμβάνουν:

- Υποστήριξη Master/Slave αρχιτεκτονικής
- Ρυθμιζόμενη ταχύτητα επικοινωνίας (clock speed)
- Υποστήριξη SPI modes (0, 1, 2, 3)
- Διαχείριση Chip Select (CS) για πολλαπλές συσκευές
- Πλήρης συμβατότητα με Arduino και ESP32

Η SPI.h χρησιμοποιείται ως βιβλιοθήκη βάσης από την MCP_CAN για την επικοινωνία με τον MCP2515 και δεν απαιτεί χειροκίνητη διαχείριση από τον προγραμματιστή στις περισσότερες περιπτώσεις.

4.6.3 Βιβλιοθήκη ESP32Servo

Η βιβλιοθήκη ESP32Servo αναπτύχθηκε από τον Kevin Harrington και αποτελεί την εξειδικευμένη έκδοση της κλασικής Arduino Servo.h για πλατφόρμες ESP32. Η ανάγκη για ξεχωριστή βιβλιοθήκη προκύπτει από το γεγονός ότι ο ESP32 διαθέτει διαφορετικό σύστημα παραγωγής PWM σήματος σε σχέση με τους κλασικούς μικροελεγκτές Arduino, καθιστώντας μη συμβατή την απευθείας χρήση της κλασικής Servo.h.

Τα βασικά χαρακτηριστικά της βιβλιοθήκης περιλαμβάνουν:

- Χρήση του LEDC peripheral του ESP32 για παραγωγή PWM σήματος
- Έλεγχος γωνίας servo σε μοίρες ή σε τιμή pulse (microseconds)
- Υποστήριξη πολλαπλών servo ταυτόχρονα
- Συμβατό API με την κλασική Arduino Servo.h για εύκολη μεταφορά κώδικα
- Διαθέσιμη μέσω PlatformIO Registry και Arduino Library Manager

Πίνακας 4.6: Χαρακτηριστικά βιβλιοθήκης ESP32Servo

Χαρακτηριστικό	Περιγραφή
Δημιουργοί	Kevin Harrington
Άδεια χρήσης	MIT
Πλατφόρμα	ESP32
Μέθοδος PWM	LEDC peripheral
Γωνία ελέγχου	0° έως 270°

4.6.4 ESP32 CAN / TWAI

Το TWAI (Two-Wire Automotive Interface) αποτελεί τον ενσωματωμένο CAN controller του ESP32, ο οποίος υποστηρίζεται από την επίσημη βιβλιοθήκη του ESP-IDF framework. Το TWAI είναι συμβατό με το πρότυπο CAN 2.0B και επιτρέπει την άμεση επικοινωνία με το δίκτυο CAN bus χωρίς την ανάγκη εξωτερικού CAN controller όπως ο MCP2515, απαιτώντας μόνο έναν εξωτερικό CAN transceiver.

Τα βασικά χαρακτηριστικά περιλαμβάνουν:

- Native CAN controller ενσωματωμένος στον ESP32
- Υποστήριξη standard (11-bit) και extended (29-bit) αναγνωριστικών
- Ταχύτητες επικοινωνίας έως 1 Mbps
- Ενσωματωμένος error handling μηχανισμός
- Υποστήριξη hardware filters
- Interrupt-based επικοινωνία για χαμηλό latency

Πίνακας 4.7: Χαρακτηριστικά βιβλιοθήκης Native CAN.

Χαρακτηριστικό	Περιγραφή
Τύπος	Native CAN Controller
Πρωτόκολλο	CAN 2.0B
Μέγιστη ταχύτητα	1 Mbps
Framework	ESP-IDF / Arduino

4.7 Συμπεράσματα

Στο παρόν κεφάλαιο παρουσιάστηκε το σύνολο των εργαλείων λογισμικού που χρησιμοποιήθηκαν για την ανάπτυξη του συστήματος. Η επιλογή των εργαλείων ακολούθησε εξελικτική λογική, με κάθε εργαλείο να επιλέγεται για τον ρόλο που εξυπηρετεί καλύτερα στο αντίστοιχο στάδιο ανάπτυξης.

Ο παρακάτω πίνακας συνοψίζει τα εργαλεία και τον ρόλο τους στο σύστημα:

Πίνακας 4.8: Συνδυασμός software και hardware.

Εργαλείο	Κατηγορία	Ρόλος στο Σύστημα
Arduino IDE	IDE	Αρχική ανάπτυξη και δοκιμή κώδικα
Visual Studio Code	Code Editor	Κύριο περιβάλλον ανάπτυξης
PlatformIO	IDE Extension	Embedded ανάπτυξη Arduino + ESP32
Python	Γλώσσα προγραμματισμού	Scripts επεξεργασίας δεδομένων
PySerial	Βιβλιοθήκη Python	Ανάγνωση σειριακής θύρας
Tkinter	Βιβλιοθήκη Python	Γραφικό περιβάλλον αποκωδικοποίησης
MCP_CAN	Βιβλιοθήκη Embedded	Επικοινωνία με MCP2515
SPI.h	Βιβλιοθήκη Embedded	Διεπαφή SPI για MCP2515
ESP32Servo	Βιβλιοθήκη Embedded	Έλεγχος servo κινητήρα σε ESP32
ESP32 TWAI	Βιβλιοθήκη Embedded	Native CAN επικοινωνία ESP32

Η συνδυαστική χρήση των παραπάνω εργαλείων επέτρεψε την ανάπτυξη ενός ολοκληρωμένου συστήματος που καλύπτει όλα τα επίπεδα — από την καταγραφή των CAN frames σε επίπεδο υλικού έως την οπτικοποίηση και αποκωδικοποίησή τους σε επίπεδο λογισμικού. Η αναλυτική παρουσίαση της διαδικασίας υλοποίησης με χρήση των εργαλείων αυτών πραγματοποιείται στο Κεφάλαιο 5.

Κεφάλαιο 5ο: Υλοποίηση και Διαδικασία Ανίχνευσης

5.1 Αρχική προσέγγιση

Το παρόν κεφάλαιο περιγράφει αναλυτικά τη διαδικασία υλοποίησης του συστήματος ελέγχου ενεργής αεροτομής μέσω δικτύου CAN bus. Η ανάπτυξη του συστήματος ακολούθησε εξελικτική προσέγγιση, στην οποία κάθε στάδιο βασίστηκε στα συμπεράσματα και τις διαπιστώσεις του προηγούμενου, με στόχο τη σταδιακή ανάπτυξη ενός λειτουργικού και αξιόπιστου συστήματος.

Η διαδικασία ξεκίνησε με την αρχική εξερεύνηση του δικτύου CAN bus του οχήματος μέσω του διαγνωστικού εργαλείου ELM327, με σκοπό τον εντοπισμό των παραμέτρων λειτουργίας που απαιτούνται για τον έλεγχο της αεροτομής — συγκεκριμένα η θέση του πεντάλ γκαζιού, η θέση του πεντάλ φρένου και η γωνία τιμονιού. Η διαπίστωση των περιορισμών του ELM327 ως προς την πρόσβαση σε ιδιότητα μηνύματα του κατασκευαστή οδήγησε στη χρήση του MCP2515 σε συνδυασμό με Arduino UNO R4 Minima για την καταγραφή ολόκληρης της κυκλοφορίας του CAN bus.

Στη συνέχεια αναπτύχθηκε μεθοδολογία συστηματικής ανάλυσης και αποκωδικοποίησης των καταγεγραμμένων frames, η οποία συνδυάζει εξαγωγή δεδομένων σε txt και την ανάλυσή τους σε Excel περιβάλλον, ανάπτυξη γραφικού περιβάλλοντος οπτικής σήμανσης μεταβολών με Tkinter και πειραματική επαλήθευση υπό πραγματικές συνθήκες οδήγησης. Η διαδικασία αυτή οδήγησε στον επιτυχή εντοπισμό των CAN frames που αντιστοιχούν στη θέση του πεντάλ γκαζιού και στην κατάσταση του φρένου.

Στο τελικό στάδιο υλοποιήθηκε το σύστημα ελέγχου με αναβάθμιση του μικροελεγκτή σε ESP32 FireBeetle, ο οποίος αναλαμβάνει την ανάγνωση των CAN frames σε πραγματικό χρόνο και την έκδοση εντολών ελέγχου προς τους servo κινητήρες της αεροτομής. Το κεφάλαιο ολοκληρώνεται με την παρουσίαση του μηχανισμού κίνησης της αεροτομής.

5.2 Αρχική Εξερεύνηση με ELM327

5.2.1 Σύνδεση και Λογισμικό

Ως πρώτο βήμα στη διαδικασία εντοπισμού των παραμέτρων λειτουργίας του οχήματος, χρησιμοποιήθηκε ο προσαρμογέας ELM327 σε συνδυασμό με την εφαρμογή κινητού YMOBD. Ο προσαρμογέας συνδέθηκε στη θύρα OBD-II του οχήματος, η οποία βρίσκεται κάτω από το ταμπλό στην πλευρά του οδηγού (σχήμα 5.1), και η επικοινωνία με την εφαρμογή πραγματοποιήθηκε μέσω Bluetooth. Δοκιμάστηκαν και άλλες εφαρμογές διάγνωσης, ωστόσο μόνο η YMOBD κατάφερε να πραγματοποιήσει επιτυχή σύνδεση με το συγκεκριμένο όχημα.



Σχήμα 5.1: Θέση θύρας obd του οχήματος και σύνδεση του ELM327.

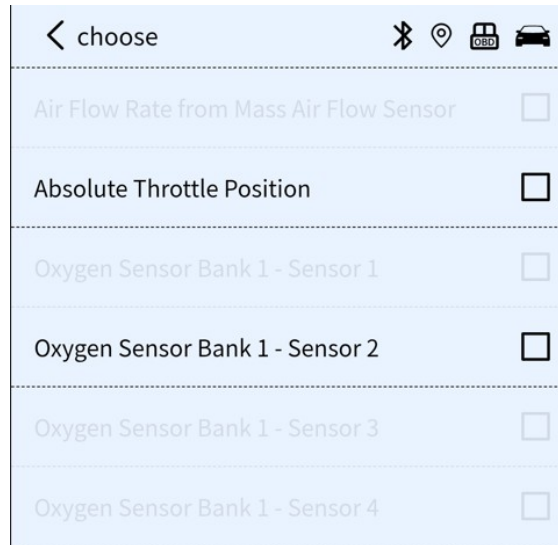
5.2.2 Διαθέσιμοι Αισθητήρες

Μέσω της εφαρμογής YMOBD έγινε πρόσβαση στη λίστα των διαθέσιμων αισθητήρων που εκτίθενται μέσω του τυποποιημένου πρωτοκόλλου OBD-II. Από τις παραμέτρους που αναζητήθηκαν, διαπιστώθηκαν τα εξής:

Πίνακας 5.1: Σχετικοί αισθητήρες που εντοπίστηκαν μέσω του ELM327

Παράμετρος	Διαθεσιμότητα μέσω OBD-II
Θέση πεταλούδας	Διαθέσιμη
Θέση πεντάλ γκαζιού	Μη διαθέσιμη
Κατάσταση φρένου	Μη διαθέσιμη
Γωνία τιμονιού	Μη διαθέσιμη

Η θέση της πεταλούδας γκαζιού εμφανιζόταν στη λίστα των διαθέσιμων αισθητήρων, ωστόσο η κατάσταση φρένου, η γωνία τιμονιού και το πεντάλ γκαζιού δεν ήταν προσβάσιμες μέσω του τυποποιημένου πρωτοκόλλου OBD-II του συγκεκριμένου οχήματος (σχήμα 5.2).



Σχήμα 5.2: Επιλογή ενδείξεων μετρήσεων από αισθητήρες του οχήματος.

5.2.3 Περιορισμοί και Απόφαση Μετάβασης στο MCP2515

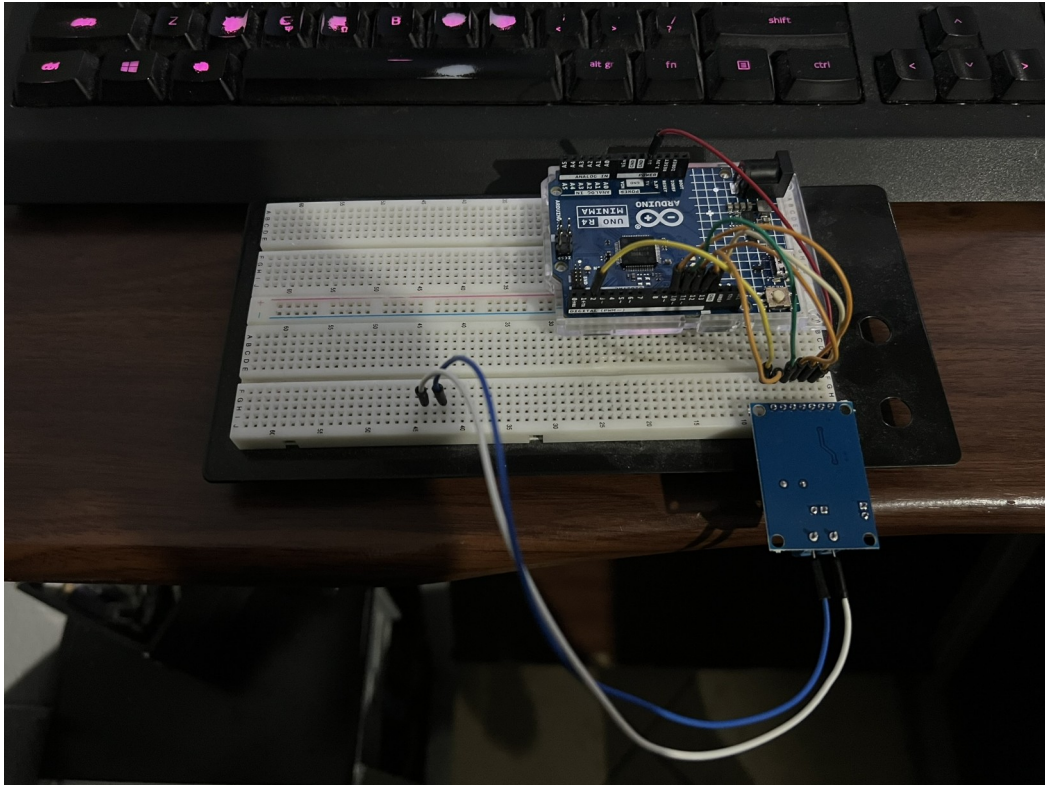
Η εξερεύνηση μέσω ELM327 κατέδειξε ότι το τυποποιημένο πρωτόκολλο OBD-II δεν παρέχει πρόσβαση σε όλες τις απαραίτητες παραμέτρους για τον έλεγχο της αεροτομής. Συγκεκριμένα, η κατάσταση φρένου, η θέση του πεντάλ γκαζιού και η γωνία τιμονιού μεταδίδονται μέσω ιδιόκτητων CAN frames του κατασκευαστή, τα οποία δεν εκτίθενται μέσω του τυποποιημένου OBD-II.

Η διαπίστωση αυτή κατέστησε αναγκαία τη χρήση εξοπλισμού που επιτρέπει την παρακολούθηση ολόκληρης της κυκλοφορίας του CAN bus, ανεξάρτητα από το πρωτόκολλο. Για τον λόγο αυτό επιλέχθηκε ο CAN controller MCP2515 σε συνδυασμό με το Arduino UNO R4 Minima, όπως παρουσιάζεται στην επόμενη ενότητα.

5.3 Καταγραφή CAN Frames με MCP2515 + Arduino

5.3.1 Συνδεσμολογία

Για την καταγραφή της κυκλοφορίας του δικτύου CAN bus του οχήματος, το module MCP2515 συνδέθηκε με το Arduino UNO R4 Minima μέσω της διεπαφής SPI (σχήμα 5.3).



Σχήμα 5.3: Σύνδεση μεταξύ CAN module mcp2515 και Arduino Uno R4 Minima.

Ο παρακάτω πίνακας παρουσιάζει την αντιστοίχιση των ακροδεκτών μεταξύ των δύο συσκευών:

Πίνακας 5.2: Αντιστοίχισης ακροδεκτών CAN module MCP2515 – Arduino Uno R4 Minima

MCP2515	Arduino UNO R4 Minima
VCC	5V
GND	GND
CS	Pin 10
MOSI	Pin 11
MISO	Pin 12
SCK	Pin 13
INT	Pin 2

Οι ακροδέκτες CAN-H και CAN-L του module MCP2515 συνδέθηκαν στις αντίστοιχες γραμμές του δικτύου CAN bus του οχήματος μέσω της θύρας OBD-II, συγκεκριμένα στις ακίδες 6 (CAN-H) και 14 (CAN-L) του συνδετήρα J1962 (σχήμα 5.4), όπως αναλύθηκε στο Κεφάλαιο 2.



Σχήμα 5.4: Σύνδεση ακροδεκτών του CAN module MCP2515 στις θύρες CAN-H (καλώδιο μπλε) και CAN-L (λευκό καλώδιο).

5.3.2 Βιβλιοθήκη MCP_CAN και Αρχικός Κώδικας

Για την ανάπτυξη του κώδικα καταγραφής χρησιμοποιήθηκε η βιβλιοθήκη MCP_CAN του Cory J. Fowler μέσω του Arduino IDE. Αξιοποιήθηκε το έτοιμο παράδειγμα καταγραφής (CAN_receive example) που παρέχεται από τη βιβλιοθήκη, το οποίο προσαρμόστηκε ώστε να εκτυπώνει μέσω της σειριακής θύρας το CAN ID και τα bytes δεδομένων κάθε frame σε δεκαεξαδική μορφή. Το αποτέλεσμα εμφανιζόταν στο Serial Monitor του Arduino IDE σε πραγματικό χρόνο.

5.3.3 Εντοπισμός Καθυστέρησης και Βελτιστοποίηση

Κατά τη διάρκεια της αρχικής καταγραφής διαπιστώθηκε σημαντική καθυστέρηση, ιδιαίτερα όταν η ανάγνωση εκτελούνταν για παρατεταμένο χρονικό διάστημα. Η αιτία εντοπίστηκε στον μεγάλο αριθμό κλήσεων Serial.print() που απαιτούνταν για την εκτύπωση κάθε frame, οι οποίες δημιουργούσαν φαινόμενο buffering στη σειριακή θύρα και επιβάρυναν υπολογιστικά τον μικροελεγκτή.

Για την αντιμετώπιση του προβλήματος εφαρμόστηκαν βελτιστοποιήσεις όπως η εφαρμογή hardware filters του MCP2515 για περιορισμό των frames που λαμβάνονται, μειώνοντας τον όγκο των δεδομένων προς επεξεργασία.

5.3.4 Μεταφορά Επεξεργασίας σε Python/PySerial

Παρά τις βελτιστοποιήσεις, η μετατροπή των δεδομένων από δυαδική σε δεκαεξαδική μορφή εντός του μικροελεγκτή εξακολουθούσε να αποτελεί πηγή επιβάρυνσης. Για τον λόγο αυτό αποφασίστηκε η μεταφορά ολόκληρης της διαδικασίας επεξεργασίας σε περιβάλλον Python.

Η νέα αρχιτεκτονική λειτουργούσε ως εξής:

- Ο μικροελεγκτής αναλάμβανε αποκλειστικά την ανάγνωση των CAN frames και την αποστολή του ακατέργαστου string (char msgString[128]) μέσω της σειριακής θύρας.
- Το Python script ανέλαβε την ανάγνωση του string μέσω PySerial και την πλήρη επεξεργασία και μετατροπή των δεδομένων

Με αυτόν τον τρόπο απαλλάχθηκε πλήρως ο μικροελεγκτής από το υπολογιστικό φορτίο της επεξεργασίας, εξαλείφοντας την καθυστέρηση που παρατηρούνταν στο αρχικό στάδιο.

5.4 Συλλογή Δεδομένων

5.4.1 Μεθοδολογία Συλλογής

Για την αποκωδικοποίηση των CAN frames και τον εντοπισμό των παραμέτρων που σχετίζονται με τη θέση του πεντάλ γκαζιού και την κατάσταση του φρένου, σχεδιάστηκε και εφαρμόστηκε μεθοδολογία συστηματικής συλλογής δεδομένων υπό διαφορετικές συνθήκες λειτουργίας του οχήματος. Στόχος ήταν η καταγραφή της κυκλοφορίας του CAN bus σε γνωστές και ελεγχόμενες συνθήκες, ώστε να είναι δυνατή η συσχέτιση των μεταβολών των bytes με συγκεκριμένες ενέργειες του οδηγού.

Για κάθε μέτρηση το Python script αποθήκευε αυτόματα τα καταγεγραμμένα frames σε αρχείο txt, με κάθε γραμμή να περιέχει το CAN ID και τα αντίστοιχα bytes δεδομένων σε δεκαεξαδική μορφή. Πραγματοποιήθηκαν τέσσερις διακριτές μετρήσεις υπό διαφορετικές συνθήκες, οι οποίες παρουσιάζονται αναλυτικά στις επόμενες υποενότητες.

5.4.2 Μέτρηση με Σβηστή Μηχανή

Η πρώτη μέτρηση πραγματοποιήθηκε με σβηστή μηχανή και ενεργοποιημένα τα ηλεκτρονικά του οχήματος (θέση ACC/ON). Κατά τη διάρκεια της μέτρησης ο οδηγός πατούσε εναλλάξ το πεντάλ γκαζιού και το πεντάλ φρένου, ενώ ταυτόχρονα καταγραφόταν η κυκλοφορία του CAN bus. Η μέτρηση αυτή είχε ως στόχο τον εντοπισμό των CAN frames που μεταβάλλονται αποκλειστικά κατά το πάτημα των πεντάλ, απομονωμένα από τον θόρυβο που εισάγει η λειτουργία του κινητήρα.

5.4.3 Μέτρηση στα 50 km/h

Η δεύτερη μέτρηση πραγματοποιήθηκε υπό πραγματικές συνθήκες οδήγησης με σταθερή ταχύτητα 50 km/h. Στόχος της μέτρησης ήταν η καταγραφή της κυκλοφορίας του CAN bus υπό σταθερές συνθήκες

λειτουργίας, επιτρέποντας τον εντοπισμό frames με σταθερές τιμές που αντιστοιχούν σε σταθερή ταχύτητα και φορτίο κινητήρα.

5.4.4 Μέτρηση στα 70 km/h

Η τρίτη μέτρηση πραγματοποιήθηκε με σταθερή ταχύτητα 70 km/h, με τον ίδιο στόχο με τη μέτρηση στα 50 km/h. Η αναζήτηση θα περιοριζόταν σε σταθερές τιμές των frames των οποίων οι τιμές θα ήταν ανάλογες με την ταχύτητα του οχήματος, διευκολύνοντας τον εντοπισμό του CAN ID που αντιστοιχεί στην ταχύτητα.

5.4.5 Εξαγωγή Δεδομένων σε .txt

Το σύνολο των μετρήσεων αποθηκεύτηκε αυτόματα από το Python script σε ξεχωριστά αρχεία .txt, ένα για κάθε συνθήκη μέτρησης. Η δομή κάθε αρχείου ήταν ενιαία, με κάθε γραμμή να περιέχει:

- Το CAN ID σε δεκαεξαδική μορφή
- Τα bytes δεδομένων του frame σε δεκαεξαδική μορφή

Η ύπαρξη ξεχωριστών αρχείων για κάθε συνθήκη μέτρησης επέτρεψε τη συγκριτική ανάλυσή τους στο επόμενο στάδιο, διευκολύνοντας τον εντοπισμό πιθανών CAN frames που παρουσίαζαν συστηματική συσχέτιση με τις ενέργειες του οδηγού.

5.5 Ανάλυση και Αποκωδικοποίηση CAN Frames

5.5.1 Ανάλυση σε Excel

Τα αρχεία .txt (σχήμα 5.5) που προέκυψαν από τις μετρήσεις μεταφέρθηκαν σε περιβάλλον Microsoft Excel για συστηματική ανάλυση. Κάθε CAN ID οργανώθηκε σε ξεχωριστή στήλη και τα bytes δεδομένων του τοποθετήθηκαν σε χρονολογική σειρά, επιτρέποντας την οπτική επισκόπηση της εξέλιξης των τιμών κάθε frame στον χρόνο (σχήμα 5.6).

Η ανάλυση επικεντρώθηκε στον εντοπισμό frames που παρουσίαζαν:

- Σταθερές τιμές κατά τη σταθερή οδήγηση — υποψήφια frames για παραμέτρους που δεν μεταβάλλονται με την ταχύτητα
- Συστηματικές μεταβολές κατά το πάτημα των πεντάλ — υποψήφια frames για γκάζι και φρένο
- Διαφορετικές σταθερές τιμές μεταξύ μέτρησης 50 km/h και 70 km/h — υποψήφια frames για ταχύτητα οχήματος

```

[2025-12-07 20:52:07.246] Standard ID: 0x30D DLC: 8 Data: 0x13 0x11 0x12 0xF3 0x13 0x02 0x13 0x02
[2025-12-07 20:52:07.246] Standard ID: 0x305 DLC: 6 Data: 0x00 0x0F 0x00 0x07 0x5D 0x00
[2025-12-07 20:52:07.254] Standard ID: 0x40D DLC: 8 Data: 0x75 0x4A 0xE9 0xA5 0xD0 0x4B 0xA6 0x0A
[2025-12-07 20:52:07.254] Standard ID: 0x2F5 DLC: 3 Data: 0x60 0xFE 0x80
[2025-12-07 20:52:07.254] Standard ID: 0x34D DLC: 8 Data: 0x00 0x03 0xFA 0xFA 0x00 0x09 0x00 0xFF
[2025-12-07 20:52:07.262] Standard ID: 0x588 DLC: 7 Data: 0xFE 0xFF 0x00 0x00 0x00 0x02 0x80
[2025-12-07 20:52:07.270] Standard ID: 0x608 DLC: 8 Data: 0x02 0x71 0x8E 0x8A 0xFF 0xFF 0x02 0x22
[2025-12-07 20:52:07.270] Standard ID: 0x488 DLC: 8 Data: 0x70 0x67 0x58 0x00 0x1E 0x5F 0xFF 0xFF
[2025-12-07 20:52:07.278] Standard ID: 0x3CD DLC: 8 Data: 0xFF 0xFD 0x00 0x02 0x00 0x22 0x60 0x00
[2025-12-07 20:52:07.278] Standard ID: 0x305 DLC: 6 Data: 0x00 0x0F 0x00 0x07 0x6E 0x00
[2025-12-07 20:52:07.291] Standard ID: 0x612 DLC: 5 Data: 0x73 0x2B 0x01 0x08 0x00
[2025-12-07 20:52:07.291] Standard ID: 0x2F5 DLC: 3 Data: 0x41 0xFF 0x80
[2025-12-07 20:52:07.299] Standard ID: 0x348 DLC: 8 Data: 0x30 0x37 0x1F 0x40 0x01 0x03 0x00 0x40
[2025-12-07 20:52:07.312] Standard ID: 0x208 DLC: 8 Data: 0x45 0x30 0x38 0x1F 0x4C 0xFF 0xFF 0xFF
[2025-12-07 20:52:07.314] Standard ID: 0x072 DLC: 5 Data: 0x02 0x00 0x00 0x00 0x00
[2025-12-07 20:52:07.314] Standard ID: 0x30D DLC: 8 Data: 0x13 0x0E 0x12 0xF0 0x12 0xF0 0x13 0x13
[2025-12-07 20:52:07.323] Standard ID: 0x305 DLC: 6 Data: 0x00 0x0F 0x00 0x07 0x7F 0x00
[2025-12-07 20:52:07.331] Standard ID: 0x208 DLC: 8 Data: 0x45 0x30 0x38 0x1F 0x4C 0xFF 0xFF 0xFF
[2025-12-07 20:52:07.331] Standard ID: 0x2F5 DLC: 3 Data: 0x32 0xFF 0x80
[2025-12-07 20:52:07.348] Standard ID: 0x34D DLC: 8 Data: 0x00 0x03 0xFA 0xFA 0x00 0x0A 0x00 0xFF
[2025-12-07 20:52:07.348] Standard ID: 0x3CD DLC: 8 Data: 0x00 0x07 0x00 0x02 0x00 0x22 0x60 0x00
[2025-12-07 20:52:07.358] Standard ID: 0x305 DLC: 6 Data: 0x00 0x0F 0x00 0x07 0x80 0x00
[2025-12-07 20:52:07.366] Standard ID: 0x2F5 DLC: 3 Data: 0x23 0xFF 0x80
[2025-12-07 20:52:07.366] Standard ID: 0x348 DLC: 8 Data: 0x30 0x37 0x1F 0x40 0x01 0x03 0x00 0x40
[2025-12-07 20:52:07.375] Standard ID: 0x468 DLC: 3 Data: 0x00 0x00 0x00
[2025-12-07 20:52:07.383] Standard ID: 0x208 DLC: 8 Data: 0x45 0x30 0x38 0x1F 0x4C 0xFF 0xFF 0xFF
[2025-12-07 20:52:07.383] Standard ID: 0x517 DLC: 3 Data: 0xA6 0xAA 0xAA
[2025-12-07 20:52:07.392] Standard ID: 0x30D DLC: 8 Data: 0x13 0x05 0x12 0xF5 0x13 0x07 0x13 0x04
[2025-12-07 20:52:07.400] Standard ID: 0x34D DLC: 8 Data: 0x00 0x03 0xFA 0xFA 0x00 0x0B 0x00 0xFF
[2025-12-07 20:52:07.400] Standard ID: 0x40D DLC: 8 Data: 0x75 0x64 0xE9 0xD9 0xD0 0xBB 0xA6 0xDC
[2025-12-07 20:52:07.409] Standard ID: 0x2F5 DLC: 3 Data: 0x14 0xFF 0x80

```

Σχήμα 5.5: ενδεικτικό στιγμιότυπο από το .txt αρχείο της μέτρησης των 50km/h.

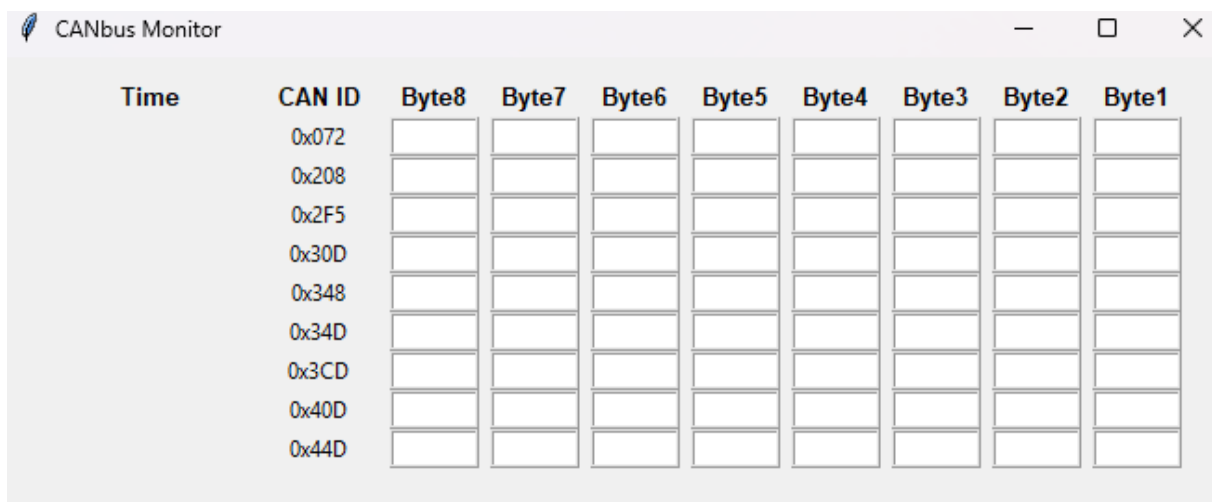
ID:	208	Data:	45	3C	38	1F	4C	FF	FF	FF		072
ID:	208	Data:	45	3C	38	1F	4C	FF	FF	FF		208
ID:	208	Data:	45	40	38	1F	4C	FF	FF	FF		2F5
ID:	208	Data:	45	34	38	1F	4C	FF	FF	FF		305
ID:	208	Data:	45	34	38	1F	4C	FF	FF	FF		300
ID:	208	Data:	45	28	38	1F	4C	FF	FF	FF		348
ID:	208	Data:	45	20	38	1F	4C	FF	FF	FF		340
ID:	208	Data:	45	20	36	1F	4C	FF	FF	FF		380
ID:	208	Data:	45	08	38	1F	4C	FF	FF	FF		3CD
ID:	208	Data:	44	FC	38	1F	4C	FF	FF	FF		400
ID:	208	Data:	44	F8	38	1F	4C	FF	FF	FF		412
ID:	208	Data:	45	04	38	1F	4C	FF	FF	FF		432
ID:	208	Data:	45	24	38	1F	4C	FF	FF	FF		440
ID:	208	Data:	45	34	38	1F	4C	FF	FF	FF		468
ID:	208	Data:	45	38	38	1F	4C	FF	FF	FF		488
ID:	208	Data:	45	2C	36	1F	4C	FF	FF	FF		495
ID:	208	Data:	45	08	38	1F	4C	FF	FF	FF		500
ID:	208	Data:	44	F4	38	1F	4C	FF	FF	FF		50E
ID:	208	Data:	44	E4	36	1F	4C	FF	FF	FF		50F
ID:	208	Data:	44	F0	38	1F	4C	FF	FF	FF		517
ID:	208	Data:	44	F4	38	1F	4C	FF	FF	FF		552
ID:	208	Data:	44	F4	38	1F	4C	FF	FF	FF		588
ID:	208	Data:	45	18	38	1F	4C	FF	FF	FF		592
ID:	208	Data:	45	28	36	1F	4C	FF	FF	FF		608
ID:	208	Data:	45	28	36	1F	4C	FF	FF	FF		60F
ID:	208	Data:	45	28	36	1F	4C	FF	FF	FF		612
ID:	208	Data:	45	28	36	1F	4C	FF	FF	FF		788
ID:	208	Data:	45	28	36	1F	4C	FF	FF	FF		780
ID:	208	Data:	45	28	36	1F	4C	FF	FF	FF		78F
ID:	208	Data:	45	28	36	1F	4C	FF	FF	FF		792
ID:	208	Data:	45	28	36	1F	4C	FF	FF	FF		795
ID:	208	Data:	45	28	36	1F	4C	FF	FF	FF		797
ID:	208	Data:	45	28	36	1F	4C	FF	FF	FF		7E2
ID:	208	Data:	45	28	36	1F	4C	FF	FF	FF		7F2

Σχήμα 5.6: Στιγμιότυπο από το αντίστοιχο αρχείο Excel των 50km/h, έχει σημειωθεί με κόκκινο ο κωδικός από δίπλα διότι υπάρχουν αρκετά σταθερά bytes. Ακριβώς από δίπλα δίνονται οι κωδικοί οι οποίοι εμφανίστηκαν.

Η διαδικασία αυτή, παρόλο που ήταν χρονοβόρα λόγω του μεγάλου όγκου δεδομένων, οδήγησε στον εντοπισμό ενός συνόλου υποψήφιων CAN IDs που χρήζαν περαιτέρω διερεύνησης. Η ανάλυση αυτή διήρκεσε αρκετές εβδομάδες λόγω του πλήθους των frames που κυκλοφορούν στο δίκτυο CAN bus του οχήματος.

5.5.2 Οπτικοποίηση με Tkinter — Flag System

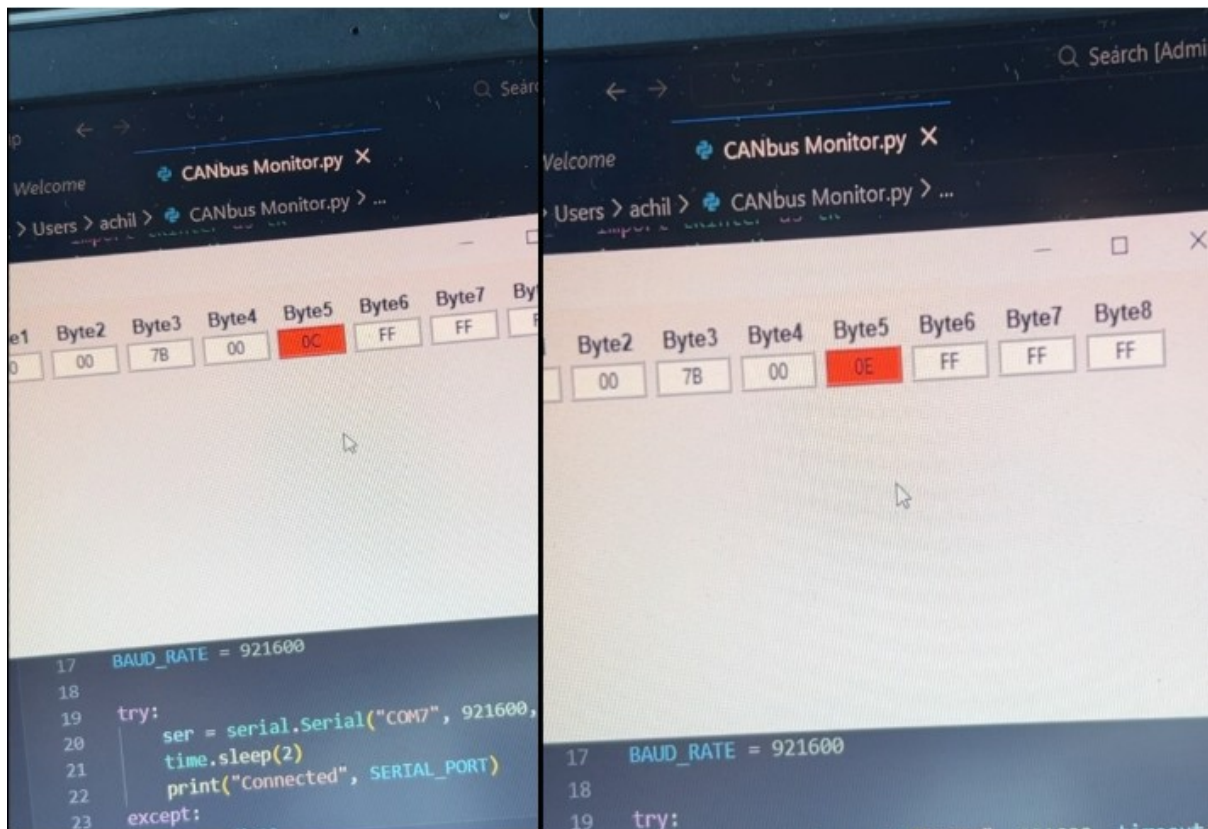
Για την αποτελεσματικότερη διερεύνηση των υποψήφιων CAN IDs αναπτύχθηκε γραφικό περιβάλλον με χρήση της βιβλιοθήκης Tkinter. Το περιβάλλον αυτό λειτουργούσε συμπληρωματικά με τον κώδικα καταγραφής του μικροελεγκτή και εμφάνιζε σε πίνακα σε πραγματικό χρόνο τα bytes δεδομένων των υποψήφιων CAN IDs που ορίζονταν εκ των προτέρων στον κώδικα (σχήμα 5.7).



Time	CAN ID	Byte8	Byte7	Byte6	Byte5	Byte4	Byte3	Byte2	Byte1
	0x072								
	0x208								
	0x2F5								
	0x30D								
	0x348								
	0x34D								
	0x3CD								
	0x40D								
	0x44D								

Σχήμα 5.7: Γραφικό περιβάλλον για περαιτέρω διερεύνηση των υποψηφίων ID.

Το βασικό χαρακτηριστικό του περιβάλλοντος ήταν το flag system: κάθε φορά που ένα byte άλλαζε τιμή, το αντίστοιχο κελί του πίνακα χρωματιζόταν αυτόματα με κόκκινο φόντο (σχήμα 5.8), παρέχοντας άμεση οπτική σήμανση των μεταβολών. Η λειτουργία αυτή επέτρεψε την παρακολούθηση έως 7-9 CAN IDs ταυτόχρονα σε πραγματικό χρόνο.



Σχήμα 5.8: Ανάγνωση CAN ID: 0x208, μεταβολή 5ου byte κατά την ώρα της πέδησης.

5.5.3 Απομόνωση και Επαλήθευση Κωδικών

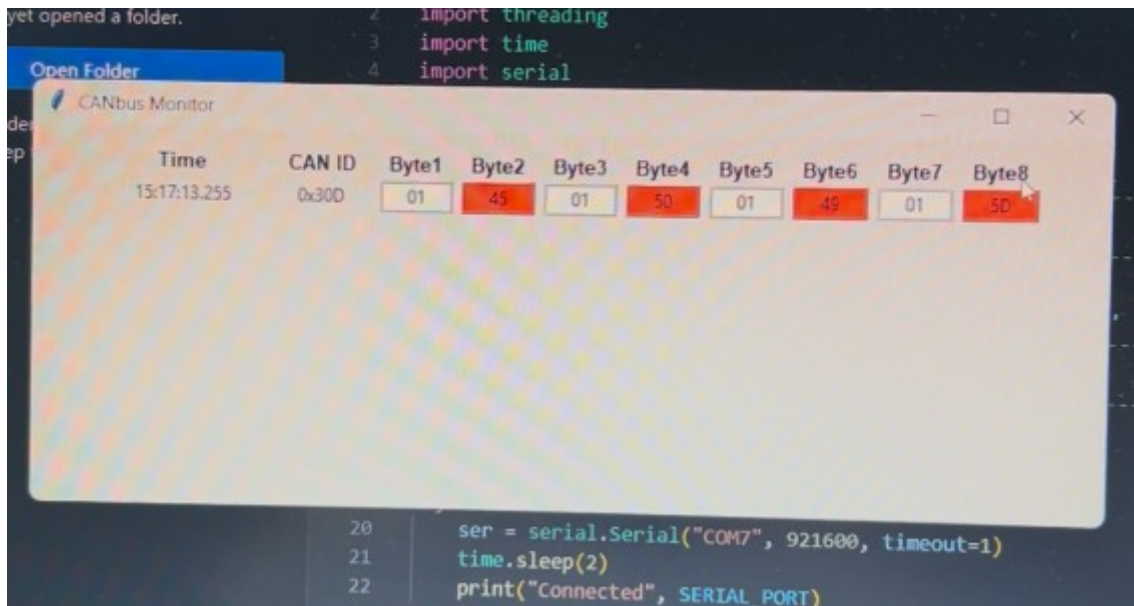
Η ταυτόχρονη παρακολούθηση πολλαπλών CAN IDs με το flag system αποδείχθηκε δυσχερής λόγω του μεγάλου αριθμού ταυτόχρονων μεταβολών. Για τον λόγο αυτό υιοθετήθηκε μεθοδολογία απομόνωσης: κάθε φορά τροποποιούνταν ο κώδικας ώστε να παρακολουθείται ένα μόνο CAN ID, ενώ ο οδηγός εκτελούσε συγκεκριμένες ενέργειες — πάτημα γκαζιού ή φρένου — παρατηρώντας την απόκριση του flag system (σχήμα 5.8).

Η διαδικασία επαναλήφθηκε για κάθε υποψήφιο CAN ID ξεχωριστά μέχρι την επιβεβαίωση ή απόρριψή του. Η μεθοδολογία αυτή, παρόλο που ήταν επίπονη, παρείχε αδιαμφισβήτητη επαλήθευση της συσχέτισης κάθε CAN ID με τη συγκεκριμένη παράμετρο λειτουργίας.

5.5.4 Εντοπισμός της Χρήσιμης Πληροφορίας

Μέσω της παραπάνω μεθοδολογίας εντοπίστηκαν επιτυχώς τα CAN frames που αντιστοιχούν στη θέση του πεντάλ γκαζιού και στην κατάσταση του φρένου. Το CAN ID που ήταν κοντά στα αναμενόμενα αποτελέσματα ήτα ο κωδικός 0x208, συγκεκριμένα το byte που φάνηκε να αντιστοιχεί στο φρένο ήταν τα 4LSB στο 5ο byte με τιμές 0x – C, όταν το φρένο δεν ήτα πατημένο, ενώ η τιμή που έπαιρνε όταν το φρένο ήταν πατημένο ήταν 0x – E. Το 4ο byte φάνηκε να μεταφέρει τις τιμές του πεντάλ γκαζιού, με

την μικρότερη να είναι 0x00 και την μέγιστη 0xC7. Κατά την διάρκεια ανάλυσης επίσης φάνηκε ο κωδικός 0x30D μεταφέρει την πληροφορία από τους αισθητήρες των τροχών για την καταγραφή των χιλιομέτρων στα bytes 2-4-6-8 για κάθε τροχό αντίστοιχα. Οι ακραίες τιμές τους ήταν αδύνατον να εντοπιστούν ωστόσο σε κατάσταση ηρεμίας η τιμή ήταν 0x00 για όλα τα bytes (σχήμα 5.9).



Σχήμα 5.9: Εξέταση κωδικού 30D, εκτιμάται ότι φέρει την πληροφορία των τροχών, με σκοπό την μέτρηση των χιλιομέτρων.

Σημειώνεται ότι η αντιστοίχιση των frames σε φυσικές ποσότητες πραγματοποιήθηκε αποκλειστικά πειραματικά, μέσω παρατήρησης των μεταβολών των δεδομένων υπό γνωστές συνθήκες λειτουργίας, και δεν βασίζεται σε επίσημη τεκμηρίωση του κατασκευαστή.

5.5.5 Γωνία Τιμονιού — Γιατί Δεν Εντοπίστηκε

Κατά τη διάρκεια της ανάλυσης διαπιστώθηκε ότι η γωνία τιμονιού δεν κυκλοφορεί στο κεντρικό δίκτυο CAN bus του οχήματος. Μετά από διαβούλευση με ειδικό του χώρου επιβεβαιώθηκε ότι στο συγκεκριμένο όχημα η πληροφορία γωνίας τιμονιού κυκλοφορεί αποκλειστικά εντός της υπομονάδας ABS μέσω δικτύου LIN bus και όχι μέσω του κεντρικού CAN bus. Η πρόσβαση σε αυτή την πληροφορία θα απαιτούσε ξεχωριστή διεπαφή με το δίκτυο LIN της μονάδας ABS, κάτι που ξεπερνά το πεδίο της παρούσας εργασίας. Για τον λόγο αυτό η γωνία τιμονιού δεν συμπεριλήφθηκε στις παραμέτρους ελέγχου του συστήματος.

5.6 Υλοποίηση Συστήματος Ελέγχου

5.6.1 Μετάβαση από Arduino σε ESP32 FireBeetle

Μετά από αυτή την εκτίμηση των CAN frames που αντιστοιχούν στο γκάζι και το φρένο, το επόμενο βήμα ήταν η υλοποίηση του συστήματος ελέγχου της αεροτομής. Αποφασίστηκε η αναβάθμιση του μικροελεγκτή από το Arduino UNO R4 Minima στο ESP32 FireBeetle 2, για τους εξής λόγους:

- Μεγαλύτερη υπολογιστική ισχύς — ο διπύρηνος επεξεργαστής Xtensa LX6 στα 240 MHz επιτρέπει την ταυτόχρονη επεξεργασία CAN frames και τον έλεγχο των servo κινητήρων χωρίς καθυστέρηση
- Μικρότερο μέγεθος — το compact form factor του FireBeetle διευκολύνει την ενσωμάτωσή του στο όχημα
- Μεγαλύτερος αριθμός ακροδεκτών — παρέχει τη δυνατότητα μελλοντικής επέκτασης σε εμπρόσθια αεροδυναμικά κινούμενα μέρη
- Ενσωματωμένο Wi-Fi και Bluetooth — παρέχει τη δυνατότητα μελλοντικής υλοποίησης ασύρματης τηλεμετρίας

Ο κώδικας που είχε αναπτυχθεί για το Arduino μεταφέρθηκε στο ESP32 μέσω του περιβάλλοντος PlatformIO και επαληθεύτηκε η ορθή λειτουργία του συστήματος καταγραφής CAN frames με τον νέο μικροελεγκτή.

5.6.2 Συνδεσμολογία ESP32 + MCP2515

Το module MCP2515 συνδέθηκε με το ESP32 FireBeetle μέσω της διεπαφής SPI. Ο παρακάτω πίνακας παρουσιάζει την αντιστοίχιση των ακροδεκτών:

Πίνακας 5.3: Αντιστοίχιση ακροδεκτών MCP2515 – ESP32 FireBeetle

MCP2515	ESP32 FireBeetle
VCC	3.3V
GND	GND
CS	GPIO 25
MOSI	GPIO 23
MISO	GPIO 19
SCK	GPIO 18
INT	GPIO 26

Οι ακροδέκτες CAN-H και CAN-L του MCP2515 συνδέθηκαν και πάλι στις ακίδες 6 και 14 της θύρας OBD-II του οχήματος.

5.6.3 Εφαρμογή Φίλτρων και Λογική Ελέγχου

Στο ESP32 εφαρμόστηκαν hardware filters του MCP2515 ώστε να γίνεται λήψη αποκλειστικά των CAN IDs και των συγκεκριμένων bytes που αντιστοιχούν στο γκάζι και το φρένο, απορρίπτοντας όλα τα υπόλοιπα frames. Αυτό μείωσε σημαντικά το υπολογιστικό φορτίο και εξάλειψε την καθυστέρηση επεξεργασίας.

Η λογική ελέγχου της αεροτομής βασίστηκε σε σύγκριση των τιμών των bytes με προκαθορισμένα όρια κατωφλίου:

- Όταν η τιμή του byte που αντιστοιχεί στο γκάζι υπερβαίνει το όριο κατωφλίου 0x50 → εντολή ανοίγματος αεροτομής στους servo κινητήρες
- Όταν ανιχνεύεται ενεργοποίηση φρένου 0x-E → εντολή κλεισίματος αεροτομής
- Όταν η τιμή από το γκάζι υπερβαίνει το όριο κατωφλίου και το φρένο την ίδια χρονική στιγμή είναι ενεργοποιημένο, δίνεται προτεραιότητα στο φρένο – εντολή κλεισίματος αεροτομής
- Όταν καμία συνθήκη δεν πληρούται → διατήρηση ουδέτερης θέσης αεροτομής

5.6.4 Τροφοδοσία Συστήματος — LM2596

Ένα κρίσιμο ζήτημα κατά την υλοποίηση ήταν η τροφοδοσία των εξαρτημάτων του συστήματος, καθώς το όχημα παρέχει τάση 12V μέσω της μπαταρίας, ενώ τα εξαρτήματα απαιτούν διαφορετικές τάσεις λειτουργίας. Για τον λόγο αυτό υιοθετήθηκε η εξής αρχιτεκτονική τροφοδοσίας:

Πίνακας 5.4: Τάση τροφοδοσίας εξαρτημάτων συστήματος.

Εξάρτημα	Τάση Λειτουργίας	Πηγή Τροφοδοσίας
ESP32 FireBeetle	5V	Φορτιστής αναπτήρα οχήματος (USB)
Servo κινητήρας	~7.4V	LM2596 buck converter από 12V
MCP2515	3.3V	ESP32 (εσωτερικός ρυθμιστής)

Το ρυθμιστικό τάσης LM2596 ρυθμίστηκε ώστε να μειώνει την τάση από 12V στα ~7.4V που απαιτούνται για την τροφοδοσία των servo κινητήρων. Η χρήση εξωτερικού ρυθμιστικού κρίθηκε απαραίτητη καθώς η άμεση σύνδεση των servo στην τάση 12V του οχήματος θα υπερέβαινε τα όρια λειτουργίας τους, ενώ η τροφοδοσία μέσω του ESP32 θα ήταν αδύνατη λόγω των υψηλών απαιτήσεων ρεύματος.

5.7 Αεροτομή και Μηχανισμός Κίνησης

5.7.1 Υπολογισμός Δύναμης και Επιλογή Servo

Για την επιλογή των κατάλληλων servo κινητήρων πραγματοποιήθηκε υπολογισμός της αεροδυναμικής δύναμης που ασκείται στο κινούμενο τμήμα της αεροτομής σε ακραία κατάσταση. Η δύναμη υπολογίστηκε από τη σχέση:

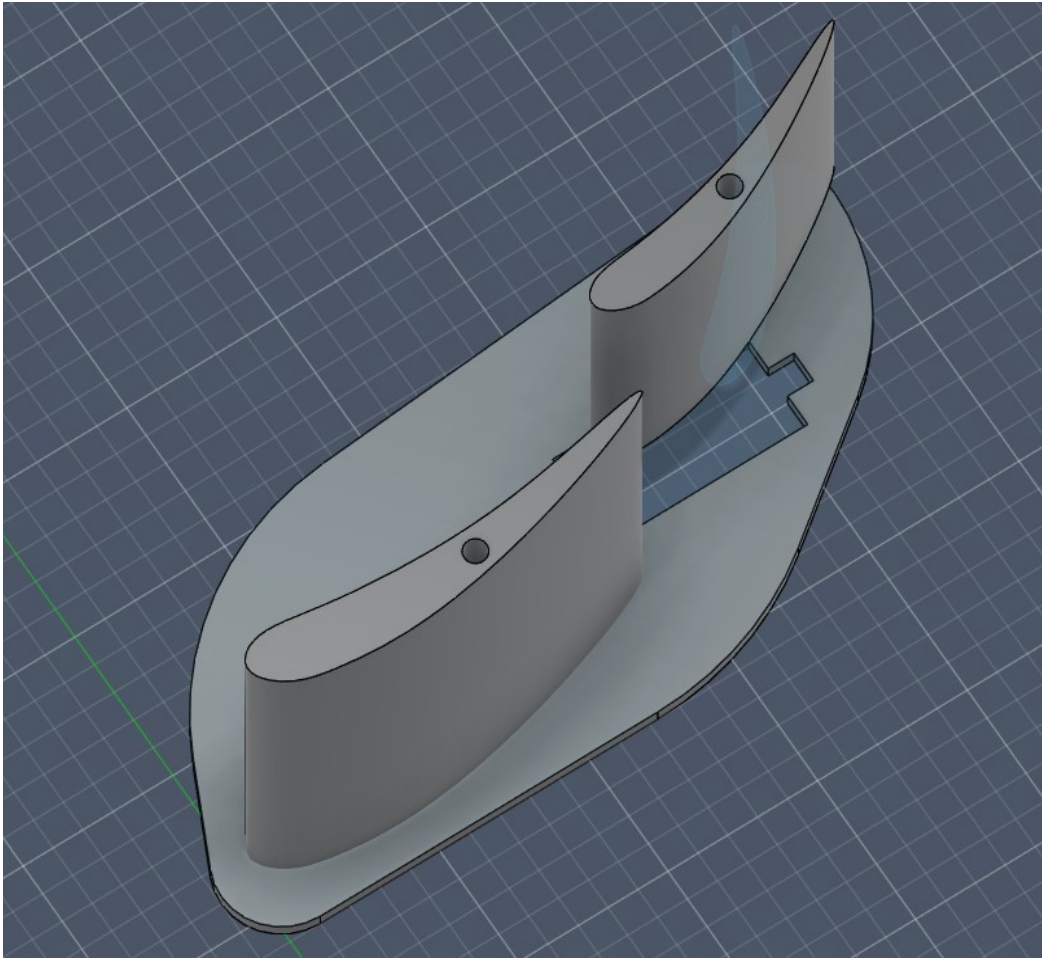
$$F = \frac{1}{2} \rho u^2 A C_L$$

όπου ρ είναι η πυκνότητα του αέρα, u η ταχύτητα του οχήματος, A η επιφάνεια του κινούμενου τμήματος της αεροτομής και C_L ο συντελεστής άντωσης. Για τους υπολογισμούς θεωρήθηκε επιφάνεια $A = 0.13125 \text{ m}^2$ ($10.5 \text{ cm} \times 12.5 \text{ cm}$), ταχύτητα αναφοράς οχήματος 250 km/h και $C_L = 2.2$, που αντιστοιχεί σε συνθήκες υψηλής αεροδυναμικής φόρτισης. Από τα αποτελέσματα προέκυψε ότι δύο servo κινητήρες ονομαστικής ροπής 80 kg/cm είναι επαρκείς για την αξιόπιστη κίνηση της αεροτομής. Η χρήση δύο κινητήρων επιλέχθηκε για λόγους συμμετρικής κατανομής των φορτίων και αυξημένης αξιοπιστίας του μηχανισμού.

Η γωνία λειτουργίας των 270° που διαθέτουν οι επιλεγμένοι servo κινητήρες προσφέρει μεγαλύτερη ευελιξία στη ρύθμιση της αεροδυναμικής συμπεριφοράς σε σχέση με τους συμβατικούς servo κινητήρες των 180° . Ωστόσο στο παρόν σύστημα αξιοποιείται εύρος κίνησης 0° έως 30° , το οποίο κρίθηκε επαρκές για την επίτευξη της επιθυμητής αεροδυναμικής απόκρισης.

5.7.2 Σχεδίαση Αεροτομής

Η γεωμετρία της αεροτομής σχεδιάστηκε με τη χρήση του λογισμικού Fusion 360 (σχήμα 5.10), βασιζόμενη σε σχεδιαστικές αρχές και γεωμετρικά χαρακτηριστικά που συναντώνται σε αγωνιστικές αεροδυναμικές διατάξεις. Στόχος της σχεδίασης ήταν η ανάπτυξη μιας διάταξης ικανής να παράγει αυξημένη κατακόρυφη δύναμη, διατηρώντας παράλληλα τη δυνατότητα ενσωμάτωσης του μηχανισμού κίνησης και τοποθέτησής της στο όχημα δοκιμών.



Σχήμα 5.10: Ανατομία τρισδιάστατης μορφής της αεροτομής με το πρόγραμμα Fusion 360.

Η τελική μορφή της αεροτομής ενδέχεται να διαφοροποιηθεί κατά τη φάση κατασκευής του πρωτοτύπου, καθώς δεν έχουν ακόμη οριστικοποιηθεί οι βάσεις στήριξης, τα στοιχεία σύνδεσης, οι ανοχές συναρμολόγησης και οι περιορισμοί που προκύπτουν από τη γεωμετρία του αμαξώματος.

Η παρούσα σχεδίαση αποτελεί προκαταρκτικό μοντέλο ανάπτυξης και χρησιμοποιήθηκε για τη μελέτη της κινηματικής λειτουργίας και της μηχανολογικής ενσωμάτωσης του συστήματος.

Δεν πραγματοποιήθηκε υπολογιστική αεροδυναμική ανάλυση (CFD) ή άλλη μορφή αεροδυναμικής προσομοίωσης. Ως εκ τούτου, οι αεροδυναμικές επιδόσεις της αεροτομής δεν έχουν επαληθευτεί και οι υπολογισμοί φορτίων βασίζονται σε συντηρητικές παραδοχές σχεδιασμού.

5.7.3 Κατασκευή με 3D Εκτύπωση

Λόγω περιορισμών κόστους και χρόνου, η αεροτομή κατασκευάστηκε σε μειωμένη κλίμακα με τη χρήση τεχνολογίας τρισδιάστατης εκτύπωσης FDM (Fused Deposition Modeling) και υλικού PETG

(σχήμα 5.11). Το πρωτότυπο αποτελεί ενδεικτική υλοποίηση της προτεινόμενης διάταξης και κατασκευάστηκε με σκοπό την επαλήθευση της ορθής λειτουργίας του μηχανισμού κίνησης και του συστήματος ελέγχου υπό πραγματικές συνθήκες λειτουργίας.



Σχήμα 5.11: Εκτυπωμένη ανατομία ενδεικτικής αεροτομής με την χρήση του 3D Printer Bambu Lab P1S

Η κατασκευή δεν αποσκοπεί στην αεροδυναμική αξιολόγηση ή στην εξαγωγή ποσοτικών αεροδυναμικών μετρήσεων, καθώς η γεωμετρία, οι διαστάσεις και τα υλικά του πρωτοτύπου διαφέρουν από αυτά μιας πλήρους κλίμακας κατασκευής. Ως εκ τούτου, το πρωτότυπο χρησιμοποιείται αποκλειστικά για τη λειτουργική επαλήθευση του συστήματος.

5.7.4 Γωνία Λειτουργίας και Έλεγχος Κίνησης

Ο έλεγχος της γωνίας της αεροτομής πραγματοποιείται μέσω της βιβλιοθήκης ESP32Servo, η οποία παράγει το κατάλληλο σήμα PWM προς τους servo κινητήρες. Το σύστημα λειτουργεί ως εξής:

Πίνακας 5.5: Αντιστοίχιση συνθηκών και θέσεων που μεταφράζονται σε μοίρες στην αεροτομή.

Συνθήκη	Εντολή Servo	Γωνία Αεροτομής
Γκάζι > κατώφλι	Άνοιγμα	~ 0°
Ενεργοποίηση φρένου	Κλείσιμο	~ 30°
Καμία συνθήκη	Ουδέτερη θέση	~ 15°

Η μετάβαση μεταξύ των θέσεων πραγματοποιείται ομαλά, αποφεύγοντας απότομες κινήσεις που θα μπορούσαν να προκαλέσουν μηχανική καταπόνηση της αεροτομής.

Με την ολοκλήρωση της υλοποίησης του μηχανισμού κίνησης της αεροτομής, το σύστημα ελέγχου ενεργής αεροδυναμικής έχει αναπτυχθεί στο σύνολό του, από την ανάγνωση των δεδομένων του CAN bus έως τον φυσικό έλεγχο της αεροτομής μέσω των servo κινητήρων. Στο επόμενο κεφάλαιο παρουσιάζονται τα συμπεράσματα που προέκυψαν από την ανάπτυξη και τη λειτουργία του συστήματος, καθώς και οι προτεινόμενες μελλοντικές βελτιώσεις που θα επεκτείνουν τις δυνατότητές του.

Κεφάλαιο 6ο: Συμπεράσματα και Μελλοντικές Βελτιώσεις

6.1 Συμπεράσματα

6.1.1 Συμπεράσματα ως προς την Αποκωδικοποίηση CAN Bus

Η διαδικασία αποκωδικοποίησης των CAN frames του οχήματος κατέδειξε ότι η πρόσβαση σε ιδιόκτητα μηνύματα κατασκευαστή απαιτεί εξοπλισμό και μεθοδολογία που υπερβαίνουν τις δυνατότητες του τυποποιημένου πρωτοκόλλου OBD-II. Η χρήση του MCP2515 σε συνδυασμό με τη μεθοδολογία συστηματικής ανάλυσης που αναπτύχθηκε απέδειξε ότι είναι εφικτός ο εντοπισμός συγκεκριμένων παραμέτρων λειτουργίας χωρίς πρόσβαση στην επίσημη τεκμηρίωση του κατασκευαστή.

Συγκεκριμένα, διαπιστώθηκε ότι:

- Η θέση του πεντάλ γκαζιού και η κατάσταση του φρένου εντοπίστηκαν επιτυχώς μέσω της πειραματικής μεθοδολογίας που αναπτύχθηκε
- Η γωνία τιμονιού δεν κυκλοφορεί στο κεντρικό CAN bus του συγκεκριμένου οχήματος, αλλά εντός της υπομονάδας ABS μέσω δικτύου LIN bus
- Ο συνδυασμός Excel και Tkinter flag system αποτέλεσε αποτελεσματικό εργαλείο για την οπτικοποίηση και αποκωδικοποίηση μεγάλου όγκου δεδομένων
- Η πειραματική επαλήθευση υπό πραγματικές συνθήκες οδήγησης αποτελεί απαραίτητο βήμα για την επιβεβαίωση της ορθότητας των εντοπισμένων frames

6.1.2 Συμπεράσματα ως προς το Σύστημα Ελέγχου

Η υλοποίηση του συστήματος ελέγχου κατέδειξε ότι είναι εφικτή η ανάπτυξη λειτουργικού συστήματος ενεργής αεροδυναμικής βασισμένου σε δεδομένα CAN bus με προσιτό εξοπλισμό και ανοιχτού κώδικα εργαλεία. Συγκεκριμένα:

- Ο συνδυασμός ESP32 FireBeetle και MCP2515 αποδείχθηκε αξιόπιστος για την ανάγνωση CAN frames σε πραγματικό χρόνο και τον ταυτόχρονο έλεγχο των servo κινητήρων
- Η εφαρμογή hardware filters του MCP2515 ήταν κρίσιμη για την αποφυγή καθυστερήσεων και την αξιόπιστη λειτουργία του συστήματος
- Η αρχιτεκτονική τροφοδοσίας με LM2596 buck converter επέτρεψε την αξιόπιστη τροφοδοσία των servo κινητήρων από το ηλεκτρικό σύστημα του οχήματος
- Το σύστημα επιτυγχάνει άμεση απόκριση στις εντολές γκαζιού και φρένου με γωνία κίνησης αεροτομής 0°-30°

6.1.3 Γενικά Συμπεράσματα

Η παρούσα εργασία απέδειξε ότι η ανάπτυξη συστήματος ενεργής αεροδυναμικής βασισμένου σε δεδομένα CAN bus είναι τεχνικά εφικτή με προσιτό εξοπλισμό και ανοιχτού κώδικα εργαλεία. Η

εξελικτική προσέγγιση που ακολουθήθηκε — από την αρχική εξερεύνηση με ELM327 έως την τελική υλοποίηση με ESP32 — κατέδειξε τη σημασία της συστηματικής μεθοδολογίας στην ανάπτυξη embedded συστημάτων αυτοκίνησης.

Ένα επιπλέον σημαντικό πλεονέκτημα του αναπτυχθέντος συστήματος είναι η πλήρης δυνατότητα παραμετροποίησής του. Σε αντίθεση με τα εμπορικά συστήματα ενεργής αεροδυναμικής, τα οποία λειτουργούν με προκαθορισμένη και μη τροποποιήσιμη λογική ελέγχου, το παρόν σύστημα επιτρέπει την ελεύθερη ρύθμιση των ορίων κατωφλίου, της γωνίας αεροτομής και της λογικής ενεργοποίησης, προσαρμόζοντας τη συμπεριφορά του στις εκάστοτε απαιτήσεις του χρήστη.

Το σύστημα που υλοποιήθηκε αποτελεί λειτουργική απόδειξη της αρχής λειτουργίας (proof of concept) και θέτει τα θεμέλια για την ανάπτυξη ενός πλήρους συστήματος ενεργής αεροδυναμικής κατάλληλου για εγκατάσταση σε πραγματικό όχημα.

6.2 Μελλοντικές Βελτιώσεις

Κατά τη διάρκεια της ανάπτυξης του συστήματος εντοπίστηκαν αρκετοί τομείς στους οποίους το σύστημα μπορεί να βελτιωθεί και να επεκταθεί. Οι προτεινόμενες βελτιώσεις οργανώνονται σε πέντε κατευθύνσεις, οι οποίες παρουσιάζονται κατά σειρά προτεραιότητας.

6.2.1 Εφαρμογή Αεροτομής Πραγματικής Κλίμακας

Η σημαντικότερη μελλοντική βελτίωση αφορά την κατασκευή και τοποθέτηση αεροτομής πραγματικής κλίμακας στο εξωτερικό του οχήματος. Η παρούσα υλοποίηση χρησιμοποιεί μειωμένης κλίμακας μοντέλο αποκλειστικά για την επαλήθευση της λογικής ελέγχου. Η εφαρμογή σε πραγματική κλίμακα θα απαιτήσει:

- Επανυπολογισμό των αεροδυναμικών φορτίων υπό πραγματικές συνθήκες οδήγησης
- Επανεπιλογή των servo κινητήρων βάσει των νέων φορτίων
- Σχεδίαση μηχανισμού σύνδεσης με το αμάξωμα του οχήματος
- Αξιολόγηση της αεροδυναμικής απόδοσης μέσω μετρήσεων σε πραγματικές συνθήκες

6.2.2 Ασύρματη Τηλεμετρία μέσω Wi-Fi

Το ESP32 FireBeetle διαθέτει ενσωματωμένο Wi-Fi το οποίο στην παρούσα εργασία δεν αξιοποιήθηκε. Η ανάπτυξη συστήματος ασύρματης τηλεμετρίας θα επέτρεπε την αποστολή δεδομένων σε πραγματικό χρόνο, όπως:

- Τρέχουσα γωνία αεροτομής
- Τιμές γκαζιού και φρένου
- Ταχύτητα οχήματος

- Κατάσταση συστήματος και τυχόν σφάλματα

Τα δεδομένα αυτά θα μπορούσαν να εμφανίζονται σε εφαρμογή κινητού ή web interface, παρέχοντας δυνατότητες παρακολούθησης και ανάλυσης της απόδοσης του συστήματος σε πραγματικό χρόνο.

6.2.3 Σχεδίαση Ολοκληρωμένου Κυκλώματος

Στην παρούσα υλοποίηση τα εξαρτήματα συνδέονται μέσω καλωδίων, κάτι που δεν είναι κατάλληλο για μόνιμη εγκατάσταση σε όχημα. Η σχεδίαση ολοκληρωμένης πλακέτας (PCB) που θα ενσωματώνει όλα τα εξαρτήματα σε μια compact μονάδα αποτελεί σημαντική κατεύθυνση βελτίωσης. Η πλακέτα αυτή θα περιλαμβάνει:

- Τον ESP32 FireBeetle ως κεντρική μονάδα επεξεργασίας
- Το module MCP2515 για την επικοινωνία CAN bus, είτε CAN transceiver VP230
- Τον ρυθμιστή τάσης LM2596 για την τροφοδοσία των servo
- Κατάλληλες προστασίες κυκλώματος από υπέρταση και αντίστροφη πολικότητα
- Συνδετήρες για την εύκολη σύνδεση με την θύρα OBD-II και τους servo κινητήρες

6.2.4 Native CAN Αρχιτεκτονική

Όπως αναφέρθηκε στο Κεφάλαιο 3, εξετάστηκε η υλοποίηση native CAN αρχιτεκτονικής με χρήση του ενσωματωμένου TWAI controller του ESP32 σε συνδυασμό με CAN transceiver VP230. Η προσέγγιση αυτή παρουσιάζει σημαντικά πλεονεκτήματα έναντι της τρέχουσας λύσης με MCP2515:

- Χαμηλότερο latency επικοινωνίας λόγω απουσίας SPI interface
- Μικρότερο υπολογιστικό φορτίο του επεξεργαστή
- Απλούστερο κύκλωμα χωρίς εξωτερικό CAN controller
- Μεγαλύτερη αξιοπιστία σε υψηλό CAN traffic

Η ολοκλήρωση αυτής της αρχιτεκτονικής αποτελεί σημαντική κατεύθυνση βελτιστοποίησης για μελλοντική έκδοση του συστήματος.

6.2.5 Εντοπισμός Γωνίας Τιμονιού

Ο εντοπισμός της γωνίας τιμονιού, η οποία στο συγκεκριμένο όχημα κυκλοφορεί εντός της υπομονάδας ABS μέσω δικτύου LIN bus, αποτελεί σημαντική μελλοντική βελτίωση. Η ενσωμάτωση της παραμέτρου αυτής στη λογική ελέγχου θα απαιτήσει:

- Ανάπτυξη διεπαφής LIN bus για πρόσβαση στη μονάδα ABS
- Αποκωδικοποίηση των LIN frames που περιέχουν την πληροφορία γωνίας τιμονιού

- Ενσωμάτωση της παραμέτρου στη λογική ελέγχου της αεροτομής

Η προσθήκη της γωνίας τιμονιού θα επέτρεπε στο σύστημα να αντιδρά και σε ελιγμούς του οχήματος, βελτιώνοντας σημαντικά την αεροδυναμική απόκριση κατά τη διάρκεια στροφών και καθιστώντας το σύστημα πιο ολοκληρωμένο.

6.3 Επίλογος

Η παρούσα εργασία αποτέλεσε μια ιδιαίτερα εμπλουτιστική εμπειρία, τόσο σε τεχνικό όσο και σε προσωπικό επίπεδο. Η διαδικασία ανάπτυξης του συστήματος — από την αρχική εξερεύνηση του δικτύου CAN bus έως την τελική υλοποίηση του μηχανισμού ελέγχου της αεροτομής — απαίτησε συνδυασμό γνώσεων από διαφορετικούς τομείς, όπως τα ηλεκτρονικά συστήματα οχημάτων, ο προγραμματισμός embedded συστημάτων και η αεροδυναμική.

Η πιο σημαντική διδαχή που προέκυψε από την εκπόνηση της εργασίας είναι ότι η συστηματική μεθοδολογία και η επιμονή στην αντιμετώπιση των προκλήσεων που ανακύπτουν είναι εξίσου σημαντικές με τις τεχνικές γνώσεις. Η αποκωδικοποίηση των ιδιόκτητων μηνυμάτων του κατασκευαστή, η αντιμετώπιση των καθυστερήσεων καταγραφής και ο εντοπισμός των σωστών CAN frames μέσα σε χιλιάδες μηνύματα αποτέλεσαν προκλήσεις που απαίτησαν δημιουργική σκέψη και επίμονη προσπάθεια.

Το αποτέλεσμα αποδεικνύει ότι η τεχνολογία που χρησιμοποιείται στα σύγχρονα οχήματα δεν είναι απροσπέλαστη. Με τα κατάλληλα εργαλεία και τη σωστή μεθοδολογία, είναι δυνατή η ανάπτυξη εξελιγμένων συστημάτων που συνδυάζουν την αγάπη για το αυτοκίνητο με την τεχνολογική καινοτομία.

ΒΙΒΛΙΟΓΡΑΦΙΑ

- [1] Porsche AG, “Porsche 911 GT3 RS world premiere”, Porsche Newsroom, 17/08/2022. Available: <https://newsroom.porsche.com/en/2022/products/porsche-911-gt3-rs-world-premiere-29177.html>.
- [2] Mercedes-AMG, “AMG ONE”, mercedes-amg.com. Available: <https://www.mercedes-amg.com/en/amg-one>.
- [3] Mercedes-AMG, “Mercedes-AMG ONE DEEP DIVE | Aerodynamics”, YouTube, 3/5/2024. Available: <https://www.youtube.com/watch?v=0qftfgCdzfs>.
- [4] Robert Bosch GmbH, “CAN Specification Version 2.0”, Stuttgart, Germany, Sep. 1991.
- [5] International Organization for Standardization, “ISO 11898-1:2015 — Road vehicles: Controller area network (CAN) — Part 1: Data link layer and physical signalling”, Geneva, Switzerland, 2015.
- [6] C. Smith, *The Car Hacker's Handbook: A Guide for the Penetration Tester*, San Francisco, CA: No Starch Press, 2016.
- [7] International Organization for Standardization, “ISO 15765-4 — Road vehicles: Diagnostics communication over Controller Area Network (DoCAN)”, Geneva, Switzerland, 2021
- [8] Elm Electronics, “ELM327: OBD to RS232 Interpreter”, ELM327 datasheet, rev. 2005.
- [9] Microchip, “MCP2515 - Stand-Alone CAN Controller With SPI™ Interface”, MCP2515 datasheet, rev. 2005
- [10] Arduino, “Arduino UNO R4 Minima: Product Reference Manual SKU: ABX00080”, Arduino Uno R4 Minima manual, rev.: 26/06/2023.
- [11] DFRobot, “FireBeetle 2 ESP32-E IoT Microcontroller SKU:DFR0654”, FireBeetle 2 ESP32-E manual, rev.: Jan., 2026
- [12] Texas Instruments, “SN65HVD23x 3.3-V CAN Bus Transceivers”, SN65HVD230 datasheet, rev.: April, 2018.
- [13] ANNIMOS DSSERVO Technology, “7.4V 80kg RC Digital Servo”, DS5180 datasheet
- [14] Texas Instruments, “LM2596 SIMPLE SWITCHER Power Converter 150 kHz 3A Step-Down Voltage Regulator”, LM2596 datasheet, rev.: April, 2013