



ΕΛΛΗΝΙΚΟ ΑΝΟΙΚΤΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΣΧΟΛΗ ΘΕΤΙΚΩΝ ΕΠΙΣΤΗΜΩΝ ΚΑΙ ΤΕΧΝΟΛΟΓΙΑΣ

**ΜΕΤΑΠΤΥΧΙΑΚΟ ΠΡΟΓΡΑΜΜΑ ΣΠΟΥΔΩΝ
ΤΕΧΝΟΛΟΓΙΑ ΥΛΙΚΟΥ ΚΑΙ ΛΟΓΙΣΜΙΚΟΥ: ΣΧΕΔΙΑΣΗ ΚΑΙ ΑΝΑΠΤΥΞΗ
ΔΙΑΧΥΤΩΝ ΣΥΣΤΗΜΑΤΩΝ ΥΠΟΛΟΓΙΣΜΟΥ**

ΜΕΤΑΠΤΥΧΙΑΚΗ ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

ΗΟΥ-CS-PGP-2026-05

**«ΣΧΕΔΙΑΣΜΟΣ ΚΑΙ ΥΛΟΠΟΙΗΣΗ ΕΝΣΩΜΑΤΩΜΕΝΟΥ ΣΥΣΤΗΜΑΤΟΣ, ΓΙΑ ΤΟΝ ΕΛΕΓΧΟ
ΗΛΕΚΤΡΙΚΩΝ ΚΙΝΗΤΗΡΩΝ»**

ΖΑΡΚΑΔΟΥΛΑΣ ΚΩΝΣΤΑΝΤΙΝΟΣ

ΕΠΙΒΛΕΠΩΝ ΚΑΘΗΓΗΤΗΣ: ΣΚΛΑΒΟΣ ΝΙΚΟΛΑΟΣ



HELLENIC OPEN UNIVERSITY SCHOOL OF SCIENCES AND TECHNOLOGY





© ΕΑΠ, 2026

Η παρούσα διατριβή, η οποία εκπονήθηκε στα πλαίσια του ΜΠΣ ΣΔΥ, και τα λοιπά αποτελέσματα της αντίστοιχης Μεταπτυχιακής Διπλωματικής Εργασίας (ΜΔΕ) αποτελούν συνιδιοκτησία του ΕΑΠ και του φοιτητή, ο καθένας από τους οποίους έχει το δικαίωμα ανεξάρτητης χρήσης και αναπαραγωγής τους (στο σύνολο ή τμηματικά) για διδακτικούς και ερευνητικούς σκοπούς, σε κάθε περίπτωση αναφέροντας τον τίτλο και το συγγραφέα και το ΕΑΠ όπου εκπονήθηκε η ΠΕ καθώς και τον επιβλέποντα και την επιτροπή κρίσης.



*Ζαρκαδούλας Κωνσταντίνος, Σχεδιασμός και υλοποίηση ενσωματωμένου
συστήματος, για τον έλεγχο ηλεκτρικών κινητήρων*

**Σχεδιασμός και υλοποίηση ενσωματωμένου συστήματος, για τον
έλεγχο ηλεκτρικών κινητήρων**

Κωνσταντίνος Ζαρκαδούλας

**Ονοματεπώνυμο
Επιβλέποντα**

Σκλάβος Νικόλαος

**Ονοματεπώνυμο
Μέλους 1**

Νικοπολιτίδης Πέτρος



Περίληψη

Η παρούσα Μεταπτυχιακή Διπλωματική Εργασία πραγματεύεται τον σχεδιασμό και την υλοποίηση ενός ενσωματωμένου συστήματος για τον έλεγχο και την απομακρυσμένη παρακολούθηση ηλεκτρικού κινητήρα, αξιοποιώντας τεχνολογίες ενσωματωμένων συστημάτων και Internet of Things (IoT). Στόχος της εργασίας είναι η ανάπτυξη μιας ολοκληρωμένης αρχιτεκτονικής χαμηλού κόστους, ικανής να συνδυάζει λειτουργίες πραγματικού χρόνου με δυνατότητες συλλογής, μετάδοσης και εποπτείας δεδομένων.

Αρχικά παρουσιάζονται οι βασικές αρχές των ενσωματωμένων συστημάτων, των μικροελεγκτών και των συστημάτων ελέγχου ηλεκτρικών κινητήρων, καθώς και σύγχρονες προσεγγίσεις απομακρυσμένης παρακολούθησης μέσω IoT πλατφορμών. Παράλληλα, πραγματοποιείται ανάλυση της αρχιτεκτονικής του συστήματος, με έμφαση στον διαχωρισμό μεταξύ του επιπέδου ελέγχου πραγματικού χρόνου και του επιπέδου επικοινωνίας και εποπτείας.

Η προτεινόμενη υλοποίηση βασίζεται σε μικροελεγκτή STM32, ο οποίος αναλαμβάνει τις χρονικά κρίσιμες λειτουργίες του συστήματος, όπως η παραγωγή σήματος PWM, η συλλογή δεδομένων από αισθητήρες και η επεξεργασία μετρήσεων σε πραγματικό χρόνο. Επιπλέον, χρησιμοποιείται μονάδα ESP32 για τη διασύνδεση του συστήματος με δίκτυο WiFi και την αποστολή δεδομένων προς την πλατφόρμα ThingsBoard μέσω πρωτοκόλλου MQTT.

Το σύστημα περιλαμβάνει αισθητήρες μέτρησης ρεύματος, θερμοκρασίας, κραδασμών και ταχύτητας περιστροφής, επιτρέποντας την παρακολούθηση πολλαπλών παραμέτρων λειτουργίας του κινητήρα. Παράλληλα, υλοποιείται μηχανισμός alarms και απομακρυσμένης εποπτείας, παρέχοντας δυνατότητα ανίχνευσης μη φυσιολογικών συνθηκών λειτουργίας.

Στο πλαίσιο της εργασίας αναπτύχθηκε πειραματική διάταξη μικρής κλίμακας και πραγματοποιήθηκε αξιολόγηση της λειτουργίας του συστήματος μέσω μετρήσεων και δοκιμών σε πραγματικές συνθήκες λειτουργίας. Τα αποτελέσματα επιβεβαιώνουν την ορθή λειτουργία της αρχιτεκτονικής, την αξιόπιστη επικοινωνία μεταξύ των υποσυστημάτων και τη δυνατότητα απομακρυσμένης παρακολούθησης σε πραγματικό χρόνο. Η προτεινόμενη αρχιτεκτονική μπορεί να αποτελέσει βάση για



μελλοντική επέκταση προς εφαρμογές προγνωστικής συντήρησης και βιομηχανικά IoT συστήματα.

Λέξεις-κλειδιά: Ενσωματωμένα Συστήματα, Έλεγχος Ηλεκτρικών Κινητήρων, Διαδίκτυο των Πραγμάτων (IoT), Αισθητήρες, Απομακρυσμένη Παρακολούθηση, Προγνωστική Συντήρηση

Περιεχόμενο: Κείμενο, κυκλωματικά διαγράμματα, μπλοκ διαγράμματα, πρόγραμμα σε γλώσσα C/C++, firmware ενσωματωμένων συστημάτων, πειραματική διάταξη και πρωτότυπο σύστημα απομακρυσμένης παρακολούθησης μέσω IoT.



*Ζαρκαδούλας Κωνσταντίνος, Σχεδιασμός και υλοποίηση ενσωματωμένου
συστήματος, για τον έλεγχο ηλεκτρικών κινητήρων*

Design and Implementation of an Embedded System for the Control of Electric Motors

Konstantinos Zarkadoulas

**Supervisor's
Full name**

Sklavos Nikolaos

**Member 1
Full name**

Nikopolitidis Petros



Abstract

This Master's Thesis examines the design and implementation of an embedded system for the control and remote monitoring of an electric motor, utilizing embedded systems technologies and Internet of Things (IoT) concepts. The objective of this work is the development of a low-cost integrated architecture capable of combining real-time operation with data acquisition, transmission and remote supervision capabilities.

Initially, the fundamental principles of embedded systems, microcontrollers and electric motor control systems are presented, along with modern approaches for remote monitoring through IoT platforms. In addition, the architecture of the proposed system is analyzed, with emphasis on the separation between the real-time control layer and the communication and supervision layer.

The proposed implementation is based on an STM32 microcontroller, which is responsible for the time-critical functions of the system, such as PWM signal generation, sensor data acquisition and real-time processing of measurements. Furthermore, an ESP32 module is utilized for WiFi connectivity and transmission of telemetry data to the ThingsBoard platform through the MQTT protocol.

The system incorporates sensors for current measurement, temperature monitoring, vibration detection and rotational speed estimation, enabling the monitoring of multiple operating parameters of the motor. In parallel, an alarm management and remote supervision mechanism is implemented, providing the capability to detect abnormal operating conditions.

As part of this work, a small-scale experimental setup was developed and the operation of the system was evaluated through measurements and testing under realistic operating conditions. The results confirm the proper operation of the proposed architecture, the reliable communication between the subsystems and the capability for real-time remote monitoring. The proposed architecture can serve as a basis for future expansion towards predictive maintenance applications and industrial IoT systems.



Keywords: Embedded Systems, Electric Motor Control, Internet of Things (IoT),
Sensors, Remote Monitoring, Predictive Maintenance



Design and Implementation of an Embedded System for Electric Motor Control and Monitoring

1. Topic Description

Electric motors are widely used in industrial automation, robotics, transportation systems and embedded control applications. Modern motor control systems increasingly require not only reliable local control but also remote monitoring, data collection and communication capabilities.

The objective of this thesis was the design and implementation of an embedded system capable of controlling and monitoring an electric motor while integrating Internet of Things (IoT) technologies. The proposed system combines real-time motor control, sensor acquisition, telemetry transmission and cloud-based monitoring within a unified architecture.

The implementation is based on an STM32 microcontroller responsible for the real-time control functions and an ESP32 module acting as an IoT gateway. The system supports remote supervision through the ThingsBoard platform using MQTT communication.

2. State of the Art

Recent developments in embedded systems have enabled the integration of motor control, sensor monitoring and cloud connectivity within low-cost platforms. Modern industrial systems increasingly employ IoT technologies to collect operational data, support predictive maintenance and improve system availability.

The combination of embedded controllers, wireless communication and cloud-based monitoring platforms allows continuous supervision of electrical machines and industrial equipment. At the same time, real-time requirements impose strict constraints on timing behavior and system reliability.

Current research focuses on architectures that separate deterministic control functions from communication and cloud services. This approach improves reliability while allowing advanced monitoring, alarm generation and remote management capabilities.



3. Methodology

The proposed system was implemented using a layered architecture consisting of:

- STM32 NUCLEO-F446RE microcontroller for real-time motor control,
- ESP32 communication module for WiFi and MQTT connectivity,
- ThingsBoard cloud platform for visualization and remote supervision.

The experimental setup incorporates several sensors, including:

- ACS712 current sensor,
- DS18B20 temperature sensor,
- MPU6050 accelerometer,
- inductive proximity sensor,
- rotary encoder for speed measurement,
- voltage monitoring circuit.

The firmware was primarily implemented using a bare-metal approach based on hardware timers, interrupts and peripheral drivers. A closed-loop speed control algorithm based on a PI controller was developed using encoder feedback. Experimental work was also performed using an alternative FreeRTOS-based firmware architecture in order to evaluate task-based real-time scheduling and software modularity.

Telemetry data are transmitted from the STM32 to the ESP32 through UART communication and subsequently forwarded to the ThingsBoard platform using the MQTT protocol. Remote commands and alarm functions are supported through the cloud interface.

4. Results

The experimental evaluation demonstrated stable and reliable operation of the complete system. The implemented PI controller successfully regulated motor speed and improved response under varying load conditions compared to open-loop operation.

All sensors operated successfully and provided useful monitoring information regarding motor speed, current consumption, temperature, acceleration and proximity



events. Telemetry data were transmitted reliably to the ThingsBoard platform, where they were visualized through dashboards and historical charts.

The implemented IoT infrastructure enabled remote monitoring, alarm generation and remote control through RPC commands. The separation between the real-time control layer and the communication layer proved effective, ensuring that network delays and cloud operations did not affect motor control performance.

The results confirm that the proposed architecture can successfully combine embedded real-time control with modern IoT technologies, providing a scalable platform for motor monitoring, remote supervision and future predictive maintenance applications.



Περιεχόμενα

Περίληψη	4
Abstract.....	7
Κατάλογος Εικόνων/Σχημάτων	15
Συντομογραφίες και Ακρωνύμια	16
1 Εισαγωγή	17
1.1 Αντικείμενο και σκοπός της Διπλωματικής Εργασίας	18
1.2 Κίνητρα και εφαρμογές	18
1.3 Στόχοι και αναμενόμενα αποτελέσματα	19
1.4 Συσχέτιση με τους Στόχους Βιώσιμης Ανάπτυξης του ΟΗΕ	20
1.5 Δομή της εργασίας.....	21
2 Θεωρητικό υπόβαθρο	22
2.1 Αρχές λειτουργίας ηλεκτρικών κινητήρων.....	22
2.1.1 Βασικές αρχές ηλεκτρομηχανικής μετατροπής ενέργειας.....	22
2.1.2 Σχέση ροπής, ταχύτητας και ηλεκτρικών μεγεθών.....	23
2.1.3 Δυναμική συμπεριφορά και ανάγκη ελέγχου	23
2.2 Τεχνικές ελέγχου ηλεκτρικών κινητήρων.....	23
2.2.1 Έλεγχος ταχύτητας με PWM	24
2.2.2 Έλεγχος ανοιχτού και κλειστού βρόγχου	24
2.2.3 Ρόλος των αισθητήρων στην ανάδραση	25
2.2.4 Εισαγωγή στους αλγόριθμους ελέγχου.....	25
2.3 Ενσωματωμένα συστήματα	25
2.3.1 Μικροελεγκτές και λειτουργία πραγματικού χρόνου	26
2.3.2 Περιορισμοί πόρων και σχεδιαστικές επιλογές	26
2.3.3 Αξιοπιστία και ο ρόλος των ενσωματωμένων συστημάτων στον έλεγχο	27
2.4 Διαδίκτυο των Πραγμάτων (IoT) στον έλεγχο ηλεκτρικών κινητήρων	27
2.4.1 Αρχιτεκτονική IoT συστημάτων ελέγχου	27
2.4.2 Πρωτόκολλα επικοινωνίας και μετάδοση δεδομένων	28
2.4.3 Απομακρυσμένη παρακολούθηση και καταγραφή δεδομένων	28
2.4.4 Υπολογισμός αξιολόγησης απόδοσης και προγνωστική συντήρηση 28	
2.5 Συναφείς εργασίες και εφαρμογές	29
3 Ανάλυση Απαιτήσεων και Προδιαγραφές Συστήματος	31
3.1 Ανάλυση αναγκών και σενάριο χρήσης.....	31
3.2 Λειτουργικές απαιτήσεις.....	32
3.3 Μη λειτουργικές απαιτήσεις.....	33



3.4	Προδιαγραφές υλικού	34
3.4.1	Ενσωματωμένη υπολογιστική μονάδα	34
3.4.2	Στάδιο οδήγησης και ισχύος κινητήρα	34
3.4.3	Αισθητήρες μέτρησης	35
3.4.4	Υποσύστημα επικοινωνίας	35
3.4.5	Τροφοδοσία και προστασία	35
3.5	Προδιαγραφές λογισμικού	36
3.5.1	Βασική δομή και ροή εκτέλεσης.....	36
3.5.2	Υλοποίηση βρόγχου ελέγχου.....	36
3.5.3	Διαχείριση αισθητήρων και δεδομένων.....	37
3.5.4	Επικοινωνία και καταγραφή δεδομένων.....	37
3.5.5	Χειρισμός σφαλμάτων και ασφαλής λειτουργία	37
3.6	Μετρικές απόδοσης	37
4	Σχεδιασμός Συστήματος	39
4.1	Συνολική αρχιτεκτονική συστήματος	39
4.2	Σχεδιασμός υλικού.....	42
4.2.1	Επιλογή ενσωματωμένης πλατφόρμας (MCU) και κριτήρια	42
4.2.2	Στάδιο οδήγησης κινητήρα και διεπαφή PWM	43
4.2.3	Αισθητήρες και σήματα μέτρησης.....	44
4.2.4	Τροφοδοσία και προστασίες	45
4.2.5	Επιλογή αισθητήρων και διασύνδεση σημάτων	46
4.2.6	Διασύνδεση STM32 – ESP32 και πρωτόκολλο επικοινωνίας.....	50
4.3	Σχεδιασμός λογισμικού.....	55
4.3.1	Δομή λογισμικού (modules)	55
4.3.2	Χρονισμός και υλοποίηση βρόγχου ελέγχου	57
4.3.3	Λογική Ελέγχου και Διαχείριση Κατάστασης.....	59
4.3.4	Διαχείριση Δεδομένων και Τηλεμετρίας	61
4.3.5	Διαχωρισμός Πραγματικού Χρόνου και Δικτυακής Επικοινωνίας ...	66
4.4	Υλοποίηση Λογισμικού και Δομή Firmware	67
4.4.1	Περιβάλλον Εκτέλεσης και Γενική Δομή Λογισμικού.....	67
4.4.2	Δομή Εργασιών και Ιεράρχηση Προτεραιοτήτων στο Περιβάλλον RTOS	69
4.4.3	Διαχείριση Περιφερειακών και Διακοπών (Interrupts)	71
4.4.4	Δομή Πλαισίων Επικοινωνίας και Έλεγχος Ακεραιότητας (CRC) ...	73
4.4.5	Μηχανισμοί Προστασίας και Διασφάλισης Λειτουργίας.....	75
4.4.6	Υλοποίηση Επιπέδου IoT (ESP32 και MQTT)	77
5	Πειραματική Διάταξη και Αξιολόγηση Συστήματος.....	80



5.1	Πειραματική Διάταξη	80
5.1.1	Υποσύστημα Αισθητήρων	86
5.2	Μεθοδολογία Δοκιμών	93
5.3	Μετρήσεις και Συλλογή Δεδομένων.....	96
5.4	Παρατηρήσεις και Περιορισμοί Συστήματος	97
5.5	Υλοποίηση και Αξιοποίηση Πλατφόρμας ThingsBoard	97
5.5.1	Δημιουργία Συσκευής και Ρύθμιση Επικοινωνίας	98
5.5.2	Μετάδοση και Απεικόνιση Τηλεμετρίας.....	98
5.5.3	Υλοποίηση Απομακρυσμένου Ελέγχου (RPC)	101
5.5.4	Σύστημα Συναγερμών (Alarm System)	103
5.5.5	Τοπικός Έλεγχος Ορίων και Βασική Προστασία Συστήματος μέσω Firmware	105
5.5.6	Συνολική Αξιολόγηση Πλατφόρμας.....	106
5.6	Ανάλυση Αποτελεσμάτων	107
5.6.1	Επιβεβαίωση Πειραματικών Μετρήσεων και Λειτουργίας Συστήματος.....	107
5.6.2	Συμπεριφορά Συστήματος Ελέγχου.....	108
5.6.3	Αξιολόγηση Μετρήσεων Αισθητήρων	109
5.6.4	Αξιολόγηση Επικοινωνίας και IoT Λειτουργιών.....	110
5.6.5	Συνολική Αξιολόγηση Συστήματος.....	111
5.7	Εναλλακτική Υλοποίηση με FreeRTOS	112
5.8	Συμπεράσματα Κεφαλαίου	114
6	Συμπεράσματα και Μελλοντική Έρευνα.....	116
6.1	Συμπεράσματα	116
6.2	Περιορισμοί Συστήματος.....	117
6.3	Μελλοντική Έρευνα και Επεκτάσεις.....	118
6.4	Τελική Αποτίμηση	120
	Αναφορές	122
	Παράρτημα	124



Κατάλογος Εικόνων/Σχημάτων

Εικόνα 4.1. Μπλοκ διάγραμμα της προτεινόμενης αρχιτεκτονικής.....	40
Εικόνα 4.2. Διάγραμμα τηλεμετρίας	53
Εικόνα 4.3. Διάγραμμα εντολών.....	54
Εικόνα 4.4. Διάγραμμα ροής δεδομένων (data flow)	65
Εικόνα 4.5. Μπλοκ απεικόνιση της αλληλεπίδρασης περιφερειακών μονάδων, ρουτινών διακοπών και εργασιών FreeRTOS στον μικροελεγκτή STM32.....	73
Εικόνα 5.1. Ηλεκτρικός κινητήρας πειραματικής διάταξης	81
Εικόνα 5.2. Συσκευή οδήγησης ηλεκτρικού κινητήρα	81
Εικόνα 5.3. Πλατφόρμα ανάπτυξης με μικροελεγκτή, STM32-Nucleo F446RE (μονάδα ελέγχου)	82
Εικόνα 5.4. Μονάδα επικοινωνίας ESP32	82
Σχήμα 5.5. Κυκλωματικό διάγραμμα της πειραματικής διάταξης	83
Εικόνα 5.6. Υλοποίηση πειραματικής διάταξης	86
Εικόνα 5.7. Αισθητήρας ρεύματος ACS712	87
Εικόνα 5.8. Αισθητήρας θερμοκρασίας DS18B20	88
Εικόνα 5.9. Αισθητήρας επιτάχυνσης MPU6050.....	89
Εικόνα 5.10. Αισθητήρας εγγύτητας	90
Εικόνα 5.11. Ενεργοποίηση εισόδων-εξόδων GPIO στον μικροελεγκτή από το περιβάλλον STM32CubeMX.....	92
Εικόνα 5.12. Καταχώρηση συσκευής στην πλατφόρμα Thingsboard.....	98
Εικόνα 5.13. Dashboard της πλατφόρμας Thingsboard με indicators τύπου arc gauge	100
Εικόνα 5.14. Dashboard της πλατφόρμας Thingsboard με γραφικές βασικών παραμέτρων συναρτήσει του χρόνου.....	101
Εικόνα 5.15. Κουμπί για απομακρυσμένη εντολή ON/OFF και setpoint RPM μέσω του dashboard της πλατφόρμας Thingsboard. Αριστερά τους φαίνεται η κάρτα με τα ενεργοποιημένα alarms	103
Εικόνα 5.16. Δημιουργία alarm rules για το σύστημα ειδοποιήσεων (μηχανισμός συναγερμών) της διάταξης.....	105



Συντομογραφίες και Ακρωνύμια

Συντομογραφία Πλήρης όρος

ADC	Analog-to-Digital Converter
CAN	Controller Area Network
CRC	Cyclic Redundancy Check
DC	Direct Current
EXTI	External Interrupt
GPIO	General-Purpose Input/Output
I2C	Inter-Integrated Circuit
IoT	Internet of Things
ISR	Interrupt Service Routine
JSON	JavaScript Object Notation
MCU	Microcontroller Unit
MQTT	Message Queuing Telemetry Transport
NTC	Negative Temperature Coefficient
PID	Proportional–Integral–Derivative
PWM	Pulse Width Modulation
RAM	Random Access Memory
RPC	Remote Procedure Call
RPM	Revolutions Per Minute
RTOS	Real-Time Operating System
UART	Universal Asynchronous Receiver/Transmitter
USART	Universal Synchronous/Asynchronous Receiver/Transmitter
WiFi	Wireless Fidelity



1 Εισαγωγή

Η παρούσα Μεταπτυχιακή Διπλωματική Εργασία πραγματεύεται τον σχεδιασμό και την υλοποίηση ενός ενσωματωμένου συστήματος για τον έλεγχο ηλεκτρικών κινητήρων, με έμφαση στη σύγχρονη προσέγγιση της ενσωμάτωσης τεχνολογιών Διαδικτύου των Πραγμάτων (IoT) και στη δυνατότητα παρακολούθησης και αξιολόγησης της λειτουργίας του συστήματος σε πραγματικό χρόνο. Οι ηλεκτρικοί κινητήρες αποτελούν βασικό στοιχείο σε πλήθος βιομηχανικών, ενεργειακών και αυτοματοποιημένων εφαρμογών, γεγονός που καθιστά τον αποδοτικό, αξιόπιστο και ασφαλή έλεγχό τους ιδιαίτερα κρίσιμο.

Τα τελευταία χρόνια, η ραγδαία εξέλιξη των ενσωματωμένων συστημάτων, των μικροελεγκτών και των αισθητήρων χαμηλής κατανάλωσης έχει επιτρέψει την ανάπτυξη ευέλικτων και οικονομικά αποδοτικών λύσεων ελέγχου και παρακολούθησης ηλεκτρικών κινητήρων. Παράλληλα, η αξιοποίηση τεχνολογιών IoT παρέχει τη δυνατότητα συλλογής, αποθήκευσης και ανάλυσης δεδομένων λειτουργίας, συμβάλλοντας τόσο στη βελτιστοποίηση της απόδοσης όσο και στην έγκαιρη ανίχνευση σφαλμάτων και ανωμαλιών.

Στο πλαίσιο αυτό, η παρούσα εργασία εστιάζει στον σχεδιασμό ενός ολοκληρωμένου συστήματος, το οποίο βασίζεται σε ενσωματωμένη πλατφόρμα ελέγχου και συνδυάζει τη μέτρηση κρίσιμων παραμέτρων λειτουργίας ενός ηλεκτρικού κινητήρα, όπως η ταχύτητα, το ρεύμα και η θερμοκρασία, με αλγορίθμους ελέγχου και μηχανισμούς επικοινωνίας. Στόχος είναι η επίτευξη αξιόπιστου ελέγχου, καθώς και η παροχή δυνατότητας απομακρυσμένης εποπτείας και αξιολόγησης της απόδοσης μέσω κατάλληλης διεπαφής.

Η εργασία φιλοδοξεί να αναδείξει τη σημασία της σύζευξης των ενσωματωμένων συστημάτων με σύγχρονες IoT υποδομές, προτείνοντας μία πρακτική και επεκτάσιμη λύση που μπορεί να αξιοποιηθεί σε πραγματικές εφαρμογές αυτοματισμού και βιομηχανικού ελέγχου. Παράλληλα, παρουσιάζονται τα πλεονεκτήματα και οι περιορισμοί της προτεινόμενης προσέγγισης, καθώς και δυνατότητες μελλοντικής επέκτασης, όπως η ενσωμάτωση τεχνικών προληπτικής συντήρησης και ανάλυσης ανωμαλιών.



1.1 Αντικείμενο και σκοπός της Διπλωματικής Εργασίας

Αντικείμενο της παρούσας διπλωματικής εργασίας είναι ο σχεδιασμός και η υλοποίηση ενός ενσωματωμένου συστήματος για τον έλεγχο ηλεκτρικών κινητήρων, το οποίο συνδυάζει λειτουργίες ελέγχου, παρακολούθησης και αξιολόγησης της απόδοσης μέσω σύγχρονων τεχνολογιών ενσωματωμένων συστημάτων και Διαδικτύου των Πραγμάτων (IoT). Το σύστημα στοχεύει στη συλλογή και επεξεργασία δεδομένων λειτουργίας του κινητήρα σε πραγματικό χρόνο, επιτρέποντας τόσο τον αποτελεσματικό έλεγχο όσο και την εποπτεία της κατάστασης του.

Σκοπός της εργασίας είναι η ανάπτυξη μίας λειτουργικής και επεκτάσιμης λύσης, βασισμένης σε μικροελεγκτή, η οποία θα μπορεί να διαχειρίζεται κρίσιμες παραμέτρους λειτουργίας ενός ηλεκτρικού κινητήρα, όπως η ταχύτητα περιστροφής, το ρεύμα και η θερμοκρασία. Παράλληλα, επιδιώκεται η υλοποίηση μηχανισμών επικοινωνίας που θα επιτρέπουν τη μετάδοση των δεδομένων σε εξωτερικό σύστημα εποπτείας ή πλατφόρμα IoT, διευκολύνοντας την απομακρυσμένη παρακολούθηση και την ανάλυση της απόδοσης.

Μέσω της συγκεκριμένης προσέγγισης, η εργασία αποσκοπεί στην ανάδειξη των δυνατοτήτων που προσφέρουν τα ενσωματωμένα συστήματα στον τομέα του ελέγχου ηλεκτρικών κινητήρων, καθώς και στη διερεύνηση του τρόπου με τον οποίο η αξιοποίηση δεδομένων λειτουργίας μπορεί να συμβάλει στη βελτιστοποίηση της απόδοσης και στη βελτίωση της αξιοπιστίας του συστήματος.

1.2 Κίνητρα και εφαρμογές

Η επιλογή του συγκεκριμένου θέματος βασίζεται στη διαρκώς αυξανόμενη χρήση ηλεκτρικών κινητήρων σε βιομηχανικά συστήματα, εγκαταστάσεις αυτοματισμού και σύγχρονες ενεργειακές εφαρμογές. Οι ηλεκτρικοί κινητήρες αποτελούν βασικά στοιχεία σε γραμμές παραγωγής, αντλίες, ανεμιστήρες, συστήματα μεταφοράς και πλήθος άλλων εφαρμογών, όπου η αδιάλειπτη και αποδοτική λειτουργία τους είναι καθοριστικής σημασίας.

Παράλληλα, η εξέλιξη των ενσωματωμένων συστημάτων και η διάδοση των τεχνολογιών IoT δημιουργούν νέες δυνατότητες στον τρόπο με τον οποίο τα



συστήματα ελέγχου σχεδιάζονται και υλοποιούνται. Η δυνατότητα συνεχούς παρακολούθησης παραμέτρων λειτουργίας και η απομακρυσμένη πρόσβαση στα δεδομένα προσφέρουν σημαντικά πλεονεκτήματα, όπως η έγκαιρη ανίχνευση δυσλειτουργιών, η μείωση του κόστους συντήρησης και η βελτίωση της συνολικής απόδοσης των εγκαταστάσεων.

Το προτεινόμενο σύστημα μπορεί να βρει εφαρμογή σε περιβάλλοντα βιομηχανικού αυτοματισμού, σε εργαστηριακές εκπαιδευτικές διατάξεις, καθώς και σε συστήματα όπου απαιτείται αξιόπιστος έλεγχος και εποπτεία ηλεκτρικών κινητήρων με περιορισμένους πόρους υλικού και ενέργειας. Επιπλέον, η αρχιτεκτονική που προτείνεται δύναται να αποτελέσει βάση για μελλοντική ενσωμάτωση λειτουργιών προληπτικής συντήρησης και ανάλυσης ανωμαλιών, αξιοποιώντας δεδομένα λειτουργίας και κατάλληλες τεχνικές επεξεργασίας.

1.3 Στόχοι και αναμενόμενα αποτελέσματα

Οι βασικοί στόχοι της παρούσας διπλωματικής εργασίας επικεντρώνονται στον σχεδιασμό, την υλοποίηση και την αξιολόγηση ενός ενσωματωμένου συστήματος για τον έλεγχο ηλεκτρικών κινητήρων, το οποίο θα συνδυάζει λειτουργίες ελέγχου, παρακολούθησης και απομακρυσμένης εποπτείας. Πιο συγκεκριμένα, οι στόχοι της εργασίας συνοψίζονται ως εξής:

- Η ανάπτυξη ενός ενσωματωμένου συστήματος βασισμένου σε μικροελεγκτή, ικανού να ελέγχει τη λειτουργία ενός ηλεκτρικού κινητήρα μέσω κατάλληλων τεχνικών, όπως η ρύθμιση της ταχύτητας με χρήση PWM και μηχανισμών ανάδρασης.
- Η ενσωμάτωση αισθητήρων για τη μέτρηση βασικών παραμέτρων λειτουργίας του κινητήρα, όπως το ρεύμα, η θερμοκρασία, και η ταχύτητα περιστροφής, με στόχο τη συνεχή παρακολούθηση της κατάστασης του.
- Η υλοποίηση αλγορίθμων ελέγχου και επεξεργασίας δεδομένων, που θα επιτρέπουν την αξιόπιστη λειτουργία του συστήματος και τη σταθερή απόκριση στις μεταβολές φορτίου ή συνθηκών λειτουργίας.
- Η σχεδίαση και υλοποίηση μηχανισμού επικοινωνίας για τη μετάδοση των δεδομένων λειτουργίας σε εξωτερικό σύστημα εποπτείας ή πλατφόρμα IoT, επιτρέποντας την απομακρυσμένη παρακολούθηση και καταγραφή των μεγεθών.



- Η αξιολόγηση της απόδοσης του συστήματος μέσω κατάλληλων μετρικών απόδοσης, με σκοπό τη διερεύνηση της αποτελεσματικότητας και των περιορισμών της προτεινόμενης λύσης.

Τα αναμενόμενα αποτελέσματα της εργασίας περιλαμβάνουν την παρουσίαση ενός λειτουργικού πρωτοτύπου ενσωματωμένου συστήματος ελέγχου ηλεκτρικού κινητήρα, καθώς και την πειραματική αξιολόγηση της λειτουργίας του σε ελεγχόμενο περιβάλλον. Επιπλέον, αναμένεται η εξαγωγή συμπερασμάτων σχετικά με την αξιοπιστία, την ακρίβεια και τη δυνατότητα επέκτασης του συστήματος, καθώς και η ανάδειξη προοπτικών για μελλοντική ενσωμάτωση προηγμένων τεχνικών, όπως η προληπτική συντήρηση και ανάλυση ανωμαλιών βάσει δεδομένων λειτουργίας.

1.4 Συσχέτιση με τους Στόχους Βιώσιμης Ανάπτυξης του ΟΗΕ

Η παρούσα Μεταπτυχιακή Διπλωματική Εργασία σχετίζεται με τους Στόχους Βιώσιμης Ανάπτυξης (Sustainable Development Goals – SDGs) του Οργανισμού Ηνωμένων Εθνών και ειδικότερα με τον Στόχο 9 «Βιομηχανία, Καινοτομία και Υποδομές» (Industry, Innovation and Infrastructure). Η ανάπτυξη ενός ενσωματωμένου συστήματος ελέγχου και απομακρυσμένης παρακολούθησης ηλεκτρικού κινητήρα αξιοποιεί σύγχρονες τεχνολογίες ενσωματωμένων συστημάτων και Internet of Things (IoT), συμβάλλοντας στην προώθηση της τεχνολογικής καινοτομίας και των έξυπνων βιομηχανικών υποδομών.

Παράλληλα, η εργασία σχετίζεται με τον Στόχο 12 «Υπεύθυνη Κατανάλωση και Παραγωγή» (Responsible Consumption and Production), καθώς η συνεχής παρακολούθηση κρίσιμων παραμέτρων λειτουργίας του κινητήρα επιτρέπει την έγκαιρη ανίχνευση δυσλειτουργιών και τη βελτιστοποίηση της λειτουργίας του εξοπλισμού. Η δυνατότητα απομακρυσμένης εποπτείας και προγνωστικής συντήρησης μπορεί να συμβάλει στη μείωση άσκοπης κατανάλωσης ενέργειας και στη βελτίωση της αξιοπιστίας των βιομηχανικών συστημάτων.

Επιπλέον, η εργασία παρουσιάζει έμμεση συσχέτιση με τον Στόχο 7 «Φθηνή και Καθαρή Ενέργεια» (Affordable and Clean Energy), μέσω της αποδοτικότερης διαχείρισης ηλεκτρομηχανολογικού εξοπλισμού και της υποστήριξης πρακτικών ενεργειακής παρακολούθησης και βελτιστοποίησης.



1.5 Δομή της εργασίας

Η παρούσα διπλωματική εργασία οργανώνεται σε έξι κεφάλαια, τα οποία καλύπτουν σταδιακά το θεωρητικό υπόβαθρο, τον σχεδιασμό, την υλοποίηση και την αξιολόγηση του προτεινόμενου συστήματος.

Στο **Κεφάλαιο 1** παρουσιάζεται η εισαγωγή της εργασίας, όπου περιγράφεται το αντικείμενο, ο σκοπός, τα κίνητρα και οι στόχοι της διπλωματικής καθώς και τα αναμενόμενα αποτελέσματα.

Στο **Κεφάλαιο 2** αναλύεται το θεωρητικό υπόβαθρο που αφορά τους ηλεκτρικούς κινητήρες, τις βασικές τεχνικές ελέγχου, τις αρχές λειτουργίας των ενσωματωμένων συστημάτων και τις τεχνολογίες Διαδικτύου των Πραγμάτων που σχετίζονται με τον έλεγχο και την παρακολούθηση ηλεκτρομηχανολογικών συστημάτων.

Στο **Κεφάλαιο 3** παρουσιάζεται η ανάλυση απαιτήσεων και οι προδιαγραφές του προτεινόμενου συστήματος, τόσο σε επίπεδο υλικού όσο και λογισμικού, ενώ καθορίζονται και οι βασικές μετρικές απόδοσης που χρησιμοποιούνται για την αξιολόγηση της λειτουργίας του.

Στο **Κεφάλαιο 4** περιγράφεται ο συνολικός σχεδιασμός του συστήματος, συμπεριλαμβανομένης της αρχιτεκτονικής, της σχεδίασης του υλικού και του λογισμικού, καθώς και της ροής δεδομένων και της επικοινωνίας μεταξύ των επιμέρους υποσυστημάτων.

Στο **Κεφάλαιο 5** παρουσιάζεται η υλοποίηση του συστήματος, με αναλυτική περιγραφή της επιλεγμένης πλατφόρμας ελέγχου, της κατασκευής του κυκλώματος, της ανάπτυξης του λογισμικού και της διασύνδεσης με το σύστημα εποπτείας.

Στο **Κεφάλαιο 6** παρουσιάζονται τα τελικά συμπεράσματα της εργασίας, οι βασικοί περιορισμοί της προτεινόμενης αρχιτεκτονικής, καθώς και προτάσεις για μελλοντική έρευνα και πιθανές επεκτάσεις του συστήματος, όπως η ενσωμάτωση προηγμένων τεχνικών ελέγχου και προγνωστικής συντήρησης.



2 Θεωρητικό υπόβαθρο

2.1 Αρχές λειτουργίας ηλεκτρικών κινητήρων

Οι ηλεκτρικοί κινητήρες αποτελούν συσκευές ηλεκτρομηχανικής μετατροπής ενέργειας, οι οποίες μετατρέπουν την ηλεκτρική ενέργεια σε μηχανική κίνηση μέσω της αλληλεπίδρασης ηλεκτρικών ρευμάτων και μαγνητικών πεδίων. Ανεξαρτήτως του τύπου του κινητήρα, η βασική αρχή λειτουργίας στηρίζεται στους νόμους του ηλεκτρομαγνητισμού, σύμφωνα με τους οποίους η διέλευση ρεύματος από αγωγό που βρίσκεται εντός μαγνητικού πεδίου οδηγεί στην ανάπτυξη δύναμης και, κατ' επέκταση, μηχανικής ροπής.

Η κατανόηση των θεμελιωδών αυτών αρχών είναι απαραίτητη για τον αποτελεσματικό σχεδιασμό συστημάτων ελέγχου ηλεκτρικών κινητήρων, καθώς επιτρέπει τη σύνδεση των ηλεκτρικών μεγεθών με τη μηχανική συμπεριφορά του συστήματος. Στο πλαίσιο της παρούσας εργασίας, η ανάλυση περιορίζεται στις βασικές έννοιες που είναι άμεσα σχετικές με τον έλεγχο, την παρακολούθηση και την αξιολόγηση της λειτουργίας του κινητήρα.

2.1.1 Βασικές αρχές ηλεκτρομηχανικής μετατροπής ενέργειας

Η ηλεκτρομηχανική μετατροπή ενέργειας βασίζεται στη δημιουργία μαγνητικών πεδίων από ηλεκτρικά ρεύματα και στην αλληλεπίδραση αυτών των πεδίων με αγωγίμα στοιχεία. Στους ηλεκτρικούς κινητήρες, τα τυλίγματα του στάτη και του δρομέα δημιουργούν μαγνητικές ροές, οι οποίες προκαλούν την ανάπτυξη δυνάμεων που οδηγούν σε περιστροφική κίνηση.

Η παραγόμενη ροπή εξαρτάται από την ένταση του ρεύματος, την ένταση του μαγνητικού πεδίου και τη γεωμετρία της μηχανής. Η δυνατότητα ρύθμισης των ηλεκτρικών μεγεθών τροφοδοσίας επιτρέπει τον έλεγχο της μηχανικής απόδοσης, γεγονός που καθιστά τους ηλεκτρικούς κινητήρες ιδιαίτερα κατάλληλους για εφαρμογές αυτοματισμού και ελέγχου.



2.1.2 Σχέση ροπής, ταχύτητας και ηλεκτρικών μεγεθών

Σε βασικό επίπεδο, η ροπή που αναπτύσσει ένας ηλεκτρικός κινητήρας σχετίζεται άμεσα με το ρεύμα που διαρρέει τα τυλίγματά του, ενώ η ταχύτητα περιστροφής επηρεάζεται από τις συνθήκες τροφοδοσίας και το εφαρμοζόμενο μηχανικό φορτίο. Η μεταβολή του φορτίου οδηγεί σε αντίστοιχες μεταβολές των ηλεκτρικών μεγεθών, γεγονός που επιτρέπει την έμμεση εκτίμηση της λειτουργικής κατάστασης του κινητήρα.

Η παρακολούθηση μεγεθών όπως το ρεύμα και η ταχύτητα αποτελεί βασικό στοιχείο στα σύγχρονα συστήματα ελέγχου, καθώς παρέχει πληροφορίες σχετικά με την καταπόνηση, την απόδοση και τη σταθερότητα λειτουργίας του κινητήρα. Η αξιοποίηση αυτών των μεγεθών είναι ιδιαίτερα σημαντική σε εφαρμογές όπου απαιτείται αξιόπιστος και αποδοτικός έλεγχος, καθώς και έγκαιρη ανίχνευση αποκλίσεων από την κανονική λειτουργία.

2.1.3 Δυναμική συμπεριφορά και ανάγκη ελέγχου

Η λειτουργία ενός ηλεκτρικού κινητήρα χαρακτηρίζεται από δυναμική συμπεριφορά, καθώς η ταχύτητα και η ροπή μεταβάλλονται ως απόκριση σε αλλαγές του φορτίου ή των συνθηκών τροφοδοσίας. Σε απουσία κατάλληλου μηχανισμού ελέγχου, οι μεταβολές αυτές ενδέχεται να οδηγήσουν σε αστάθεια, μειωμένη απόδοση ή ακόμη και σε λειτουργία εκτός ασφαλών ορίων.

Για τον λόγο αυτό, τα σύγχρονα συστήματα ελέγχου βασίζονται σε τεχνικές ανάδρασης, οι οποίες επιτρέπουν τη συνεχή παρακολούθηση της κατάστασης του κινητήρα και την προσαρμογή των παραμέτρων λειτουργίας του. Η ύπαρξη μηχανισμών ελέγχου καθιστά δυνατή τη σταθεροποίηση της ταχύτητας, τη βελτίωση της απόκρισης του συστήματος και τη διασφάλιση της αξιόπιστης λειτουργίας σε μεταβαλλόμενες συνθήκες.

Η ανάγκη για αποτελεσματικό έλεγχο αποτελεί το θεωρητικό υπόβαθρο πάνω στο οποίο βασίζεται η υλοποίηση των τεχνικών ελέγχου που παρουσιάζονται στα επόμενα κεφάλαια της παρούσας εργασίας.

2.2 Τεχνικές ελέγχου ηλεκτρικών κινητήρων

Ο έλεγχος ηλεκτρικών κινητήρων αποσκοπεί στη ρύθμιση βασικών μεγεθών λειτουργίας, όπως η ταχύτητα, η ροπή και η κατεύθυνση περιστροφής, ώστε ο



κινητήρας να λειτουργεί σύμφωνα με τις απαιτήσεις της εφαρμογής. Στα σύγχρονα συστήματα αυτοματισμού, ο έλεγχος υλοποιείται κυρίως μέσω ενσωματωμένων συστημάτων, τα οποία επιτρέπουν την ευέλικτη και ακριβή ρύθμιση των παραμέτρων λειτουργίας, καθώς και τη συνεχή παρακολούθηση της κατάστασης του κινητήρα.

Οι τεχνικές ελέγχου μπορούν να διακριθούν σε απλές μεθόδους ανοικτού βρόχου και σε πιο προηγμένες μεθόδους κλειστού βρόχου, οι οποίες βασίζονται σε μηχανισμούς ανάδρασης. Στο πλαίσιο της παρούσας εργασίας, η ανάλυση επικεντρώνεται στις βασικές τεχνικές που είναι άμεσα εφαρμόσιμες σε ενσωματωμένα συστήματα χαμηλού κόστους και περιορισμένων πόρων.

2.2.1 Έλεγχος ταχύτητας με PWM

Μία από τις πλέον διαδεδομένες τεχνικές ελέγχου ταχύτητας ηλεκτρικών κινητήρων είναι η χρήση διαμόρφωσης εύρους παλμού (Pulse Width Modulation – PWM). Η τεχνική αυτή βασίζεται στη ρύθμιση του λόγου ενεργού χρόνου ενός ψηφιακού σήματος, μέσω του οποίου ελέγχεται η μέση τιμή της τάσης που εφαρμόζεται στον κινητήρα.

Με τη μεταβολή του duty cycle του σήματος PWM, καθίσταται δυνατή η ομαλή ρύθμιση της ταχύτητας περιστροφής, χωρίς την ανάγκη μεταβολής της τάσης τροφοδοσίας. Η μέθοδος PWM παρουσιάζει υψηλή ενεργειακή απόδοση και είναι ιδιαίτερα κατάλληλη για υλοποίηση μέσω μικροελεγκτών, γεγονός που την καθιστά βασική επιλογή σε ενσωματωμένα συστήματα ελέγχου.

2.2.2 Έλεγχος ανοικτού και κλειστού βρόχου

Στον έλεγχο ανοικτού βρόχου, η έξοδος του συστήματος δεν επηρεάζει άμεσα το σήμα ελέγχου. Στην περίπτωση αυτή, η ταχύτητα ή η ροπή του κινητήρα ρυθμίζεται χωρίς να λαμβάνονται υπόψη πιθανές μεταβολές του φορτίου ή εξωτερικές διαταραχές. Παρότι η προσέγγιση αυτή είναι απλή στην υλοποίηση, παρουσιάζει περιορισμένη ακρίβεια και ευαισθησία σε μεταβολές των συνθηκών λειτουργίας.

Αντιθέτως, στον έλεγχο κλειστού βρόχου χρησιμοποιείται ανάδραση από αισθητήρες, ώστε το σήμα ελέγχου να προσαρμόζεται με βάση τη διαφορά μεταξύ της επιθυμητής και της πραγματικής τιμής της ελεγχόμενης μεταβλητής. Η χρήση κλειστού βρόχου επιτρέπει τη βελτίωση της σταθερότητας και της ακρίβειας του συστήματος, καθιστώντας τον έλεγχο πιο αξιόπιστο σε εφαρμογές με μεταβαλλόμενο φορτίο.



2.2.3 Ρόλος των αισθητήρων στην ανάδραση

Οι αισθητήρες αποτελούν κρίσιμο στοιχείο στα συστήματα ελέγχου κλειστού βρόχου, καθώς παρέχουν τις απαραίτητες πληροφορίες σχετικά με την κατάσταση λειτουργίας του κινητήρα. Μετρήσεις όπως η ταχύτητα περιστροφής, το ρεύμα και η θερμοκρασία χρησιμοποιούνται για την αξιολόγηση της απόδοσης και την ανίχνευση αποκλίσεων από την κανονική λειτουργία.

Η ενσωμάτωση αισθητήρων σε ένα ενσωματωμένο σύστημα ελέγχου επιτρέπει τη συνεχή παρακολούθηση και την προσαρμογή των παραμέτρων λειτουργίας, συμβάλλοντας στη βελτίωση της αξιοπιστίας και της ασφάλειας του κινητήρα. Παράλληλα, τα δεδομένα που συλλέγονται μπορούν να αξιοποιηθούν για σκοπούς καταγραφής και απομακρυσμένης εποπτείας, όπως παρουσιάζεται στα επόμενα κεφάλαια της εργασίας.

2.2.4 Εισαγωγή στους αλγόριθμους ελέγχου

Για την υλοποίηση ελέγχου κλειστού βρόχου σε ενσωματωμένα συστήματα, χρησιμοποιούνται συχνά απλοί αλγόριθμοι ελέγχου, οι οποίοι προσφέρουν ικανοποιητική απόδοση με περιορισμένο υπολογιστικό κόστος. Οι αλγόριθμοι αυτοί βασίζονται στη σύγκριση της επιθυμητής τιμής με τη μετρούμενη τιμή και στη ρύθμιση του σήματος ελέγχου ώστε να ελαχιστοποιείται το σφάλμα.

Η χρήση τέτοιων αλγορίθμων επιτρέπει την επίτευξη σταθερής λειτουργίας και ομαλής απόκρισης του κινητήρα, ενώ παράλληλα παραμένει συμβατή με τις δυνατότητες μικροελεγκτών χαμηλής κατανάλωσης. Η πρακτική υλοποίηση των αλγορίθμων αυτών παρουσιάζεται αναλυτικά στο κεφάλαιο της υλοποίησης.

Στην παρούσα εργασία χρησιμοποιείται βασικός αναλογικός-ολοκληρωτικός ελεγκτής (PI), ο οποίος ρυθμίζει το duty cycle του PWM με βάση το σφάλμα μεταξύ επιθυμητής και μετρούμενης ταχύτητας.

2.3 Ενσωματωμένα συστήματα

Τα ενσωματωμένα συστήματα αποτελούν υπολογιστικά συστήματα ειδικού σκοπού, τα οποία έχουν σχεδιαστεί για να εκτελούν συγκεκριμένες λειτουργίες ελέγχου, παρακολούθησης ή επεξεργασίας δεδομένων εντός ενός μεγαλύτερου φυσικού συστήματος. Σε αντίθεση με τα γενικού σκοπού υπολογιστικά συστήματα, τα ενσωματωμένα συστήματα χαρακτηρίζονται από αυστηρούς περιορισμούς σε πόρους,



όπως η υπολογιστική ισχύς, η μνήμη και η κατανάλωση ενέργειας, καθώς και από απαιτήσεις αξιοπιστίας και προβλέψιμης λειτουργίας.

Στο πλαίσιο εφαρμογών ελέγχου ηλεκτρικών κινητήρων, τα ενσωματωμένα συστήματα διαδραματίζουν κεντρικό ρόλο, καθώς αναλαμβάνουν τη συλλογή δεδομένων από αισθητήρες, την εκτέλεση αλγορίθμων ελέγχου και τη δημιουργία σημάτων οδήγησης προς τα στοιχεία ισχύος του κινητήρα. Η επιλογή της ενσωματωμένης πλατφόρμας και ο σχεδιασμός του λογισμικού επηρεάζουν άμεσα την απόκριση, τη σταθερότητα και την αξιοπιστία του συνολικού συστήματος.

2.3.1 Μικροελεγκτές και λειτουργία πραγματικού χρόνου

Ένα κρίσιμο χαρακτηριστικό των ενσωματωμένων συστημάτων ελέγχου είναι η ικανότητά τους να λειτουργούν σε πραγματικό χρόνο. Στις εφαρμογές ελέγχου κινητήρων, η έγκαιρη απόκριση του συστήματος σε μεταβολές των μετρούμενων μεγεθών είναι καθοριστικής σημασίας για τη σταθερότητα και την ασφάλεια της λειτουργίας.

Η έννοια του πραγματικού χρόνου δεν σχετίζεται απαραίτητα με την υψηλή υπολογιστική ισχύ, αλλά με την προβλέψιμη εκτέλεση των λειτουργιών εντός καθορισμένων χρονικών ορίων. Για τον λόγο αυτό, ο σχεδιασμός του λογισμικού βασίζεται συνήθως σε επαναλαμβανόμενους βρόχους ελέγχου, χρονικά καθορισμένες διεργασίες και σαφή διαχείριση των διακοπών (interrupts).

2.3.2 Περιορισμοί πόρων και σχεδιαστικές επιλογές

Τα ενσωματωμένα συστήματα λειτουργούν υπό σημαντικούς περιορισμούς πόρων, οι οποίοι επηρεάζουν άμεσα τις σχεδιαστικές επιλογές. Η περιορισμένη μνήμη και υπολογιστική ισχύς καθιστούν απαραίτητη την επιλογή απλών και αποδοτικών αλγορίθμων, καθώς και τη βελτιστοποίηση του κώδικα.

Στο πλαίσιο της παρούσας εργασίας, οι περιορισμοί αυτοί λαμβάνονται υπόψη ώστε το προτεινόμενο σύστημα να παραμένει λειτουργικό, αξιόπιστο και επεκτάσιμο, χωρίς περιττή πολυπλοκότητα. Η αποφυγή υπερβολικά απαιτητικών υπολογιστικών τεχνικών συμβάλλει στη σταθερή λειτουργία και διευκολύνει την υλοποίηση σε πλατφόρμες μικροελεγκτών.



2.3.3 Αξιοπιστία και ο ρόλος των ενσωματωμένων συστημάτων στον έλεγχο

Η αξιοπιστία αποτελεί βασική απαίτηση στα ενσωματωμένα συστήματα ελέγχου, ιδιαίτερα σε εφαρμογές που σχετίζονται με ηλεκτρομηχανολογικό εξοπλισμό. Η συνεχής παρακολούθηση των βασικών μεγεθών λειτουργίας και η έγκαιρη ανίχνευση αποκλίσεων συμβάλλουν στην ασφαλή λειτουργία και στη μείωση της πιθανότητας αστοχιών.

Μέσω της κατάλληλης σχεδίασης του υλικού και του λογισμικού, τα ενσωματωμένα συστήματα μπορούν να υποστηρίξουν μηχανισμούς προστασίας, καταγραφής δεδομένων και απομακρυσμένης εποπτείας. Οι δυνατότητες αυτές αποτελούν τη βάση για την περαιτέρω αξιοποίηση τεχνολογιών παρακολούθησης και αξιολόγησης της απόδοσης, όπως παρουσιάζεται στα επόμενα κεφάλαια της εργασίας.

2.4 Διαδίκτυο των Πραγμάτων (IoT) στον έλεγχο ηλεκτρικών κινητήρων

Το Διαδίκτυο των Πραγμάτων (Internet of Things – IoT) αναφέρεται στη διασύνδεση φυσικών συσκευών με υπολογιστικά συστήματα και δίκτυα επικοινωνίας, με σκοπό τη συλλογή, τη μετάδοση και την αξιοποίηση δεδομένων λειτουργίας. Στον τομέα του ελέγχου ηλεκτρικών κινητήρων, η αξιοποίηση τεχνολογιών IoT επιτρέπει τη μετάβαση από τοπικά και απομονωμένα συστήματα ελέγχου σε συστήματα εποπτείας με δυνατότητες απομακρυσμένης παρακολούθησης και καταγραφής δεδομένων.

Η ενσωμάτωση IoT λειτουργιών δεν μεταβάλλει τον βασικό ρόλο του ενσωματωμένου συστήματος ελέγχου, αλλά τον επεκτείνει, παρέχοντας πρόσθετες δυνατότητες εποπτείας, ανάλυσης και αξιολόγησης της απόδοσης του κινητήρα.

2.4.1 Αρχιτεκτονική IoT συστημάτων ελέγχου

Μια τυπική αρχιτεκτονική IoT στον έλεγχο ηλεκτρικών κινητήρων περιλαμβάνει τρία βασικά επίπεδα: το επίπεδο συσκευών (device layer), το επίπεδο επικοινωνίας και το επίπεδο εφαρμογών. Στο επίπεδο συσκευών εντάσσονται οι αισθητήρες και το ενσωματωμένο σύστημα ελέγχου, το οποίο συλλέγει τα δεδομένα λειτουργίας και εκτελεί τις βασικές λειτουργίες ελέγχου.

Το επίπεδο επικοινωνίας αναλαμβάνει τη μετάδοση των δεδομένων προς ανώτερα συστήματα, μέσω ενσύρματων ή ασύρματων τεχνολογιών, ενώ το επίπεδο εφαρμογών



παρέχει διεπαφές εποπτείας, αποθήκευσης και απεικόνισης των δεδομένων. Η αρχιτεκτονική αυτή επιτρέπει τη διατήρηση της αυτονομίας του τοπικού ελέγχου, ενώ παράλληλα προσφέρει δυνατότητες απομακρυσμένης πρόσβασης στα δεδομένα.

2.4.2 Πρωτόκολλα επικοινωνίας και μετάδοση δεδομένων

Η επικοινωνία μεταξύ του ενσωματωμένου συστήματος και των IoT εφαρμογών υλοποιείται μέσω κατάλληλων πρωτοκόλλων επικοινωνίας, τα οποία επιλέγονται με βάση τις απαιτήσεις αξιοπιστίας, καθυστέρησης και κατανάλωσης πόρων. Στις εφαρμογές ελέγχου ηλεκτρικών κινητήρων, η μετάδοση δεδομένων αφορά κυρίως μετρήσεις κατάστασης και όχι σήματα άμεσου ελέγχου.

Η επιλογή απλών και ελαφρών πρωτοκόλλων επιτρέπει τη διατήρηση χαμηλού υπολογιστικού και ενεργειακού κόστους, ενώ διασφαλίζει τη συμβατότητα με ενσωματωμένες πλατφόρμες περιορισμένων πόρων. Η επικοινωνία αυτή λειτουργεί συμπληρωματικά προς τον τοπικό έλεγχο και δεν τον αντικαθιστά.

Στην παρούσα εργασία η μετάδοση τηλεμετρικών δεδομένων προς την πλατφόρμα IoT βασίζεται στο πρωτόκολλο MQTT, το οποίο είναι κατάλληλο για αξιόπιστη και αποδοτική επικοινωνία σε περιβάλλοντα IoT και ενσωματωμένων συστημάτων.

2.4.3 Απομακρυσμένη παρακολούθηση και καταγραφή δεδομένων

Ένα από τα βασικά πλεονεκτήματα της ενσωμάτωσης IoT τεχνολογιών είναι η δυνατότητα απομακρυσμένης παρακολούθησης της λειτουργίας του κινητήρα. Μέσω της συνεχούς καταγραφής μεγεθών, όπως η ταχύτητα, το ρεύμα και η θερμοκρασία, καθίσταται δυνατή η αξιολόγηση της συμπεριφοράς του συστήματος σε βάθος χρόνου.

Η αποθήκευση των δεδομένων λειτουργίας επιτρέπει την ανάλυση τάσεων και την αναγνώριση αποκλίσεων από τη φυσιολογική λειτουργία. Οι δυνατότητες αυτές είναι ιδιαίτερα χρήσιμες σε περιβάλλοντα όπου απαιτείται εποπτεία πολλαπλών συστημάτων ή περιορισμένη φυσική πρόσβαση στον εξοπλισμό.

2.4.4 Υπολογισμός αξιολόγησης απόδοσης και προγνωστική συντήρηση

Η αξιοποίηση δεδομένων λειτουργίας μέσω IoT υποδομών δημιουργεί τις προϋποθέσεις για την εφαρμογή προληπτικών προσεγγίσεων στη συντήρηση και την αξιολόγηση της απόδοσης. Χωρίς να απαιτείται η υλοποίηση σύνθετων αλγορίθμων,



η απλή παρακολούθηση βασικών μεγεθών μπορεί να προσφέρει ενδείξεις για την κατάσταση του κινητήρα και την πιθανή εμφάνιση δυσλειτουργιών.

Στο πλαίσιο της παρούσας εργασίας, οι IoT δυνατότητες αντιμετωπίζονται ως εργαλείο υποστήριξης της εποπτείας και της αξιολόγησης του συστήματος, χωρίς να επηρεάζουν τον βασικό βρόχο ελέγχου. Η προσέγγιση αυτή διατηρεί την απλότητα και την αξιοπιστία του συστήματος, ενώ ταυτόχρονα επιτρέπει μελλοντικές επεκτάσεις.

2.5 Συναφείς εργασίες και εφαρμογές

Στη διεθνή βιβλιογραφία έχουν παρουσιαστεί πληθώρα εργασιών που αφορούν τον έλεγχο και την παρακολούθηση ηλεκτρικών κινητήρων μέσω ενσωματωμένων συστημάτων και τεχνολογιών IoT. Οι περισσότερες από αυτές εστιάζουν είτε στη βελτίωση της απόδοσης του ελέγχου είτε στη συλλογή και αξιοποίηση δεδομένων λειτουργίας για σκοπούς εποπτείας και συντήρησης.

Σε αρκετές μελέτες προτείνεται η χρήση μικροελεγκτών σε συνδυασμό με αισθητήρες ρεύματος, ταχύτητας και θερμοκρασίας, με στόχο τη συνεχή παρακολούθηση της κατάστασης του κινητήρα και τη μετάδοση των δεδομένων σε απομακρυσμένες πλατφόρμες. Οι προσεγγίσεις αυτές αναδεικνύουν τη σημασία των ενσωματωμένων συστημάτων χαμηλού κόστους ως βασικών δομικών στοιχείων σύγχρονων συστημάτων ελέγχου.

Παράλληλα, έχουν παρουσιαστεί εφαρμογές που αξιοποιούν IoT υποδομές για την υποστήριξη προληπτικής συντήρησης, βασιζόμενες κυρίως στην ανάλυση ιστορικών δεδομένων λειτουργίας. Οι εργασίες αυτές καταδεικνύουν ότι ακόμη και η απλή καταγραφή βασικών μεγεθών μπορεί να προσφέρει χρήσιμες ενδείξεις για την κατάσταση του κινητήρα και τη μελλοντική του συμπεριφορά, χωρίς απαραίτητα να απαιτούνται σύνθετοι αλγόριθμοι ή υψηλή υπολογιστική ισχύς.

Ωστόσο, σε αρκετές περιπτώσεις παρατηρείται αυξημένη πολυπλοκότητα στη σχεδίαση, είτε λόγω της χρήσης απαιτητικών υπολογιστικών τεχνικών είτε λόγω της εξάρτησης από εκτεταμένες υποδομές επεξεργασίας δεδομένων. Η προσέγγιση αυτή ενδέχεται να μην είναι κατάλληλη για εφαρμογές με περιορισμένους πόρους ή αυξημένες απαιτήσεις αξιοπιστίας^[Marwedel, 2011; Samie et al., 2016].

Η παρούσα εργασία διαφοροποιείται υιοθετώντας μία πιο ισορροπημένη προσέγγιση, όπου ο βασικός έλεγχος υλοποιείται τοπικά μέσω ενσωματωμένου συστήματος, ενώ



οι δυνατότητες IoT αξιοποιούνται συμπληρωματικά για εποπτεία και αξιολόγηση της απόδοσης. Με τον τρόπο αυτό, επιτυγχάνεται συνδυασμός απλότητας, αξιοπιστίας και επεκτασιμότητας, χωρίς υπερβολική πολυπλοκότητα στον σχεδιασμό.



3 Ανάλυση Απαιτήσεων και Προδιαγραφές Συστήματος

3.1 Ανάλυση αναγκών και σενάριο χρήσης

Η ανάγκη για αξιόπιστο έλεγχο και παρακολούθηση ηλεκτρικών κινητήρων προκύπτει από το γεγονός ότι οι κινητήρες αποτελούν κρίσιμο στοιχείο σε πλήθος εφαρμογών αυτοματισμού. Η λειτουργία τους επηρεάζεται από μεταβολές φορτίου, συνθήκες τροφοδοσίας και θερμική καταπόνηση, οι οποίες μπορούν να οδηγήσουν σε μείωση απόδοσης ή ανεπιθύμητες αποκλίσεις από το επιθυμητό σημείο λειτουργίας. Για τον λόγο αυτό, απαιτείται ένα σύστημα που να παρέχει: (α) τοπικό έλεγχο με σταθερή και προβλέψιμη συμπεριφορά και (β) παρακολούθηση βασικών μεγεθών λειτουργίας με δυνατότητα καταγραφής και εποπτείας.

Στο πλαίσιο της παρούσας διπλωματικής, το προτεινόμενο σύστημα αντιμετωπίζεται ως ένα ολοκληρωμένο ενσωματωμένο σύστημα ελέγχου, το οποίο αναλαμβάνει τη συλλογή δεδομένων από αισθητήρες (π.χ. ρεύμα, θερμοκρασία, ταχύτητα κ.α.), την επεξεργασία τους σε πραγματικό χρόνο και τη δημιουργία σημάτων οδήγησης προς το στάδιο ισχύος του κινητήρα. Παράλληλα, υποστηρίζεται η μετάδοση επιλεγμένων δεδομένων σε εξωτερικό σύστημα εποπτείας (IoT πλατφόρμα ή αντίστοιχη διεπαφή), ώστε να είναι δυνατή η απομακρυσμένη παρακολούθηση και η αξιολόγηση της απόδοσης.

Το σενάριο χρήσης βασίζεται σε έναν κινητήρα που λειτουργεί υπό μεταβαλλόμενες συνθήκες (π.χ. αλλαγές φορτίου ή στόχου ταχύτητας). Ο χρήστης/χειριστής ορίζει μία επιθυμητή κατάσταση λειτουργίας (ενδεικτικά, στόχο ταχύτητας), ενώ το σύστημα ελέγχου ρυθμίζει το σήμα οδήγησης ώστε να επιτυγχάνεται σταθερή λειτουργία. Ταυτόχρονα, οι μετρήσεις καταγράφονται και παρουσιάζονται, επιτρέποντας την αναγνώριση τάσεων (π.χ. σταδιακή αύξηση θερμοκρασίας ή αυξημένο ρεύμα σε ίδια ταχύτητα), οι οποίες μπορούν να λειτουργήσουν ως ενδείξεις επιβάρυνσης ή αποκλίσεων.

Με βάση τα παραπάνω, η παρούσα εργασία δίνει έμφαση σε μία ισορροπημένη προσέγγιση: ο βασικός έλεγχος υλοποιείται τοπικά, με στόχο την αξιοπιστία και την προβλέψιμη απόκριση, ενώ οι δυνατότητες IoT αξιοποιούνται συμπληρωματικά για εποπτεία και αξιολόγηση, χωρίς να επηρεάζουν τον βρόχο ελέγχου.



3.2 Λειτουργικές απαιτήσεις

Οι λειτουργικές απαιτήσεις περιγράφουν τις βασικές λειτουργίες που οφείλει να υποστηρίζει το προτεινόμενο σύστημα, προκειμένου να επιτελεί τον σκοπό του ελέγχου και της παρακολούθησης ενός ηλεκτρικού κινητήρα. Οι απαιτήσεις αυτές καθορίζουν τη συμπεριφορά του συστήματος από την οπτική του χρήστη και του χειριστή, καθώς και τις βασικές διεργασίες που πρέπει να εκτελούνται από το ενσωματωμένο σύστημα.

Στο πλαίσιο της παρούσας εργασίας, το σύστημα πρέπει να ικανοποιεί τις ακόλουθες λειτουργικές απαιτήσεις:

- Το σύστημα πρέπει να είναι σε θέση να ελέγχει τη λειτουργία ενός ηλεκτρικού κινητήρα, ρυθμίζοντας την ταχύτητα περιστροφής του μέσω κατάλληλου σήματος οδήγησης.
- Το σύστημα πρέπει να υποστηρίζει την ανάγνωση και επεξεργασία μετρήσεων από αισθητήρες που αφορούν βασικές παραμέτρους λειτουργίας του κινητήρα, όπως το ρεύμα, η θερμοκρασία, η ταχύτητα κ.α.
- Το σύστημα πρέπει να εκτελεί τον βρόχο ελέγχου σε τακτά χρονικά διαστήματα, ώστε να διασφαλίζεται η σταθερή και προβλέψιμη απόκριση του κινητήρα σε μεταβολές των συνθηκών λειτουργίας.
- Το σύστημα πρέπει να παρέχει τη δυνατότητα καθορισμού επιθυμητής τιμής λειτουργίας (π.χ. στόχος ταχύτητας) από τον χρήστη ή τον χειριστή.
- Το σύστημα πρέπει να καταγράφει τις βασικές μετρήσεις λειτουργίας, επιτρέποντας την αποθήκευση και μετέπειτα αξιολόγηση των δεδομένων.
- Το σύστημα πρέπει να υποστηρίζει τη μετάδοση επιλεγμένων δεδομένων λειτουργίας σε εξωτερικό σύστημα εποπτείας ή πλατφόρμα IoT, με σκοπό την απομακρυσμένη παρακολούθηση.
- Το σύστημα πρέπει να διατηρεί τη βασική λειτουργία ελέγχου ανεξάρτητα από τη διαθεσιμότητα της σύνδεσης με το εξωτερικό σύστημα εποπτείας.
- Το σύστημα πρέπει να είναι σε θέση να εντοπίζει καταστάσεις εκτός προκαθορισμένων ορίων λειτουργίας (π.χ. υπερβολικό ρεύμα ή θερμοκρασία) και να τις καταγράφει ως γεγονότα λειτουργίας.

Οι παραπάνω λειτουργικές απαιτήσεις αποτελούν τη βάση για τον σχεδιασμό της αρχιτεκτονικής του συστήματος και καθοδηγούν τόσο την επιλογή του υλικού όσο και



την ανάπτυξη του λογισμικού, όπως παρουσιάζεται στα επόμενα κεφάλαια της εργασίας.

3.3 Μη λειτουργικές απαιτήσεις

Οι μη λειτουργικές απαιτήσεις καθορίζουν τα ποιοτικά χαρακτηριστικά και τους περιορισμούς του προτεινόμενου συστήματος, τα οποία επηρεάζουν τον τρόπο λειτουργίας του χωρίς να αφορούν άμεσα συγκεκριμένες λειτουργίες. Οι απαιτήσεις αυτές είναι ιδιαίτερα σημαντικές σε ενσωματωμένα συστήματα ελέγχου, καθώς συνδέονται με ζητήματα αξιοπιστίας, χρονικής συμπεριφοράς και αποδοτικής χρήσης πόρων.

Στο πλαίσιο της παρούσας εργασίας, το σύστημα πρέπει να ικανοποιεί τις ακόλουθες μη λειτουργικές απαιτήσεις:

- Το σύστημα πρέπει να παρουσιάζει προβλέψιμη και σταθερή χρονική συμπεριφορά, ώστε ο βρόχος ελέγχου να εκτελείται εντός καθορισμένων χρονικών ορίων.
- Το σύστημα πρέπει να λειτουργεί αξιόπιστα σε συνθήκες συνεχούς λειτουργίας, χωρίς να απαιτείται συχνή επανεκκίνηση ή παρέμβαση από τον χρήστη.
- Το σύστημα πρέπει να είναι ανθεκτικό σε προσωρινές απώλειες επικοινωνίας με εξωτερικά συστήματα εποπτείας, χωρίς να επηρεάζεται η βασική λειτουργία ελέγχου.
- Το σύστημα πρέπει να χρησιμοποιεί αποδοτικά τους διαθέσιμους υπολογιστικούς πόρους, λαμβάνοντας υπόψη τους περιορισμούς μνήμης και επεξεργαστικής ισχύος της ενσωματωμένης πλατφόρμας.
- Το σύστημα πρέπει να χρησιμοποιεί αποδοτικά τους διαθέσιμους ενεργειακούς πόρους.
- Το σύστημα πρέπει να επιτρέπει την εύκολη παραμετροποίηση βασικών ρυθμίσεων, χωρίς να απαιτούνται εκτεταμένες τροποποιήσεις στον πηγαίο κώδικα.
- Το σύστημα πρέπει να είναι επεκτάσιμο, επιτρέποντας την προσθήκη επιπλέον αισθητήρων ή λειτουργιών χωρίς ουσιαστική αλλαγή της βασικής αρχιτεκτονικής.



- Το σύστημα πρέπει να υποστηρίζει ασφαλή λειτουργία, διασφαλίζοντας ότι καταστάσεις εκτός ορίων δεν οδηγούν σε ανεξέλεγκτη ή επικίνδυνη συμπεριφορά του κινητήρα.

Οι μη λειτουργικές απαιτήσεις συμπληρώνουν τις λειτουργικές απαιτήσεις και αποτελούν καθοριστικό παράγοντα για τον σχεδιασμό και την υλοποίηση του συστήματος, επηρεάζοντας τόσο την επιλογή της ενσωματωμένης πλατφόρμας όσο και τις σχεδιαστικές αποφάσεις σε επίπεδο λογισμικού.

3.4 Προδιαγραφές υλικού

Οι προδιαγραφές υλικού καθορίζουν τα βασικά χαρακτηριστικά των επιμέρους στοιχείων που συνθέτουν το προτεινόμενο σύστημα, με στόχο την υποστήριξη των λειτουργικών και μη λειτουργικών απαιτήσεων που αναλύθηκαν στις προηγούμενες ενότητες. Η επιλογή του υλικού γίνεται με γνώμονα την αξιοπιστία, τη συμβατότητα με εφαρμογές ελέγχου ηλεκτρικών κινητήρων και τη δυνατότητα υλοποίησης σε ενσωματωμένη πλατφόρμα περιορισμένων πόρων.

3.4.1 Ενσωματωμένη υπολογιστική μονάδα

Το σύστημα πρέπει να βασίζεται σε μικροελεγκτή κατάλληλο για εφαρμογές ελέγχου σε πραγματικό χρόνο. Η ενσωματωμένη υπολογιστική μονάδα πρέπει να διαθέτει επαρκή υπολογιστική ισχύ και μνήμη ώστε να υποστηρίζει τον βρόχο ελέγχου, την επεξεργασία σημάτων από αισθητήρες και τη βασική επικοινωνία με εξωτερικά συστήματα.

Απαιτείται η ύπαρξη ενσωματωμένων περιφερειακών, όπως χρονομετρητές, μετατροπείς αναλογικού σε ψηφιακό σήμα (ADC) και μονάδες παραγωγής σημάτων PWM, τα οποία είναι απαραίτητα για την υλοποίηση του ελέγχου του κινητήρα.

3.4.2 Στάδιο οδήγησης και ισχύος κινητήρα

Το σύστημα πρέπει να περιλαμβάνει κατάλληλο στάδιο οδήγησης, το οποίο να επιτρέπει τη διασύνδεση της ενσωματωμένης υπολογιστικής μονάδας με τον ηλεκτρικό κινητήρα. Το στάδιο αυτό οφείλει να υποστηρίζει τον έλεγχο της ταχύτητας μέσω σημάτων PWM και να είναι σχεδιασμένο ώστε να αντέχει τα ρεύματα και τις τάσεις λειτουργίας του κινητήρα.



Ιδιαίτερη σημασία δίνεται στην ηλεκτρική απομόνωση και την προστασία της ενσωματωμένης μονάδας από υπερτάσεις ή υπερένταση, προκειμένου να διασφαλίζεται η ασφαλής και αξιόπιστη λειτουργία του συστήματος.

3.4.3 Αισθητήρες μέτρησης

Το σύστημα πρέπει να υποστηρίζει τη σύνδεση αισθητήρων για τη μέτρηση βασικών παραμέτρων λειτουργίας του κινητήρα. Ενδεικτικά, απαιτούνται αισθητήρες για τη μέτρηση του ρεύματος κατανάλωσης, της θερμοκρασίας, της ταχύτητας περιστροφής και όποιας άλλης παραμέτρου κρίνεται απαραίτητη για την ορθή λειτουργία του κινητήρα.

Οι αισθητήρες πρέπει να παρέχουν σήματα συμβατά με τα διαθέσιμα περιφερειακά της ενσωματωμένης μονάδας (π.χ. αναλογικά σήματα προς ADC ή ψηφιακές εξόδους), ενώ η ακρίβεια και η αξιοπιστία τους πρέπει να είναι επαρκείς για σκοπούς ελέγχου και εποπτείας.

3.4.4 Υποσύστημα επικοινωνίας

Για την υποστήριξη λειτουργιών απομακρυσμένης παρακολούθησης, το σύστημα πρέπει να διαθέτει υποσύστημα επικοινωνίας, το οποίο να επιτρέπει τη μετάδοση επιλεγμένων δεδομένων λειτουργίας προς εξωτερικό σύστημα εποπτείας ή IoT πλατφόρμα.

Το υποσύστημα επικοινωνίας πρέπει να είναι σχεδιασμένο έτσι ώστε να λειτουργεί συμπληρωματικά προς τον τοπικό έλεγχο και να μην επηρεάζει τη χρονική συμπεριφορά του βρόχου ελέγχου. Η επιλογή της τεχνολογίας επικοινωνίας λαμβάνει υπόψη την απλότητα, την αξιοπιστία και τη συμβατότητα με ενσωματωμένα συστήματα.

3.4.5 Τροφοδοσία και προστασία

Το σύστημα πρέπει να διαθέτει κατάλληλο κύκλωμα τροφοδοσίας για την παροχή σταθερών τάσεων στα επιμέρους υποσυστήματα. Απαιτείται η πρόβλεψη βασικών μηχανισμών προστασίας, όπως φίλτρα, ασφάλειες ή περιοριστές ρεύματος, ώστε να διασφαλίζεται η ασφαλής λειτουργία σε περιπτώσεις ανωμαλιών.

Η σωστή σχεδίαση της τροφοδοσίας και των μηχανισμών προστασίας είναι κρίσιμη για τη μακροχρόνια αξιοπιστία του συστήματος και την αποφυγή αστοχιών στο υλικό.



3.5 Προδιαγραφές λογισμικού

Οι προδιαγραφές λογισμικού καθορίζουν τη δομή και τη λειτουργικότητα του λογισμικού που εκτελείται στην ενσωματωμένη υπολογιστική μονάδα, με στόχο την υποστήριξη του ελέγχου και της παρακολούθησης του ηλεκτρικού κινητήρα. Το λογισμικό πρέπει να είναι σχεδιασμένο ώστε να ανταποκρίνεται στις απαιτήσεις πραγματικού χρόνου, αξιοπιστίας και αποδοτικής χρήσης πόρων που αναλύθηκαν στις προηγούμενες ενότητες.

Η ανάπτυξη του λογισμικού βασίζεται σε αρθρωτή (modular) αρχιτεκτονική, ώστε να διευκολύνεται η συντήρηση, η επεκτασιμότητα και η μελλοντική αναβάθμιση του συστήματος.

3.5.1 Βασική δομή και ροή εκτέλεσης

Το λογισμικό πρέπει να υλοποιεί έναν κεντρικό βρόχο εκτέλεσης, στον οποίο εκτελούνται περιοδικά οι βασικές λειτουργίες του συστήματος. Η ροή εκτέλεσης πρέπει να είναι προβλέψιμη και να βασίζεται σε χρονικά καθορισμένα διαστήματα, ώστε να διασφαλίζεται η σταθερή λειτουργία του βρόχου ελέγχου.

Οι λειτουργίες υψηλής προτεραιότητας, όπως η ανάγνωση αισθητήρων και η ενημέρωση του σήματος οδήγησης, πρέπει να εκτελούνται με σταθερή περιοδικότητα, ενώ δευτερεύουσες λειτουργίες, όπως η επικοινωνία ή η καταγραφή δεδομένων, μπορούν να εκτελούνται σε μεγαλύτερα χρονικά διαστήματα.

3.5.2 Υλοποίηση βρόχου ελέγχου

Το λογισμικό πρέπει να περιλαμβάνει μηχανισμό υλοποίησης του βρόχου ελέγχου του κινητήρα, ο οποίος να βασίζεται στη σύγκριση της επιθυμητής τιμής λειτουργίας με τις μετρούμενες τιμές από τους αισθητήρες. Με βάση το σφάλμα που προκύπτει, το σύστημα υπολογίζει το κατάλληλο σήμα οδήγησης, το οποίο εφαρμόζεται στον κινητήρα μέσω του υποσυστήματος PWM.

Στην παρούσα εργασία χρησιμοποιείται βασικός αναλογικός–ολοκληρωτικός (PI) μηχανισμός ελέγχου για τη ρύθμιση της ταχύτητας περιστροφής.

Η υλοποίηση του βρόχου ελέγχου πρέπει να είναι απλή και αποδοτική, ώστε να ανταποκρίνεται στους περιορισμούς της ενσωματωμένης πλατφόρμας και να διασφαλίζει σταθερή απόκριση.



3.5.3 Διαχείριση αισθητήρων και δεδομένων

Το λογισμικό πρέπει να υποστηρίζει την αξιόπιστη ανάγνωση δεδομένων από τους συνδεδεμένους αισθητήρες, καθώς και τη βασική επεξεργασία των μετρήσεων. Απαιτείται η εφαρμογή κατάλληλων μηχανισμών φιλτραρίσματος ή μέσου όρου, όπου αυτό κρίνεται απαραίτητο, ώστε να μειώνεται η επίδραση του θορύβου στις μετρήσεις. Οι μετρήσεις πρέπει να αποθηκεύονται προσωρινά στη μνήμη του συστήματος και να είναι διαθέσιμες τόσο για τον βρόχο ελέγχου όσο και για σκοπούς καταγραφής και μετάδοσης.

3.5.4 Επικοινωνία και καταγραφή δεδομένων

Το λογισμικό πρέπει να υποστηρίζει μηχανισμό επικοινωνίας με εξωτερικό σύστημα εποπτείας ή IoT πλατφόρμα, μέσω του οποίου μεταδίδονται επιλεγμένα δεδομένα λειτουργίας. Η επικοινωνία αυτή πρέπει να είναι μη παρεμβατική ως προς τον βρόχο ελέγχου και να λειτουργεί ανεξάρτητα από την κύρια λειτουργία του συστήματος. Παράλληλα, το λογισμικό πρέπει να υποστηρίζει βασική καταγραφή δεδομένων και γεγονότων λειτουργίας, ώστε να είναι δυνατή η μεταγενέστερη αξιολόγηση της συμπεριφοράς του συστήματος.

3.5.5 Χειρισμός σφαλμάτων και ασφαλής λειτουργία

Το λογισμικό πρέπει να περιλαμβάνει μηχανισμούς ανίχνευσης και χειρισμού καταστάσεων σφάλματος, όπως μετρήσεις εκτός προκαθορισμένων ορίων ή αστοχίες αισθητήρων. Σε τέτοιες περιπτώσεις, το σύστημα πρέπει να μεταβαίνει σε ασφαλή κατάσταση λειτουργίας, αποτρέποντας ανεπιθύμητη ή επικίνδυνη συμπεριφορά του κινητήρα.

Η ύπαρξη μηχανισμών ασφαλούς λειτουργίας ενισχύει την αξιοπιστία του συστήματος και αποτελεί βασική απαίτηση σε εφαρμογές ελέγχου ηλεκτρομηχανολογικού εξοπλισμού.

3.6 Μετρικές απόδοσης

Οι μετρικές απόδοσης χρησιμοποιούνται για την ποσοτική αξιολόγηση της λειτουργίας του προτεινόμενου συστήματος, με βάση τους στόχους και τις απαιτήσεις που έχουν τεθεί στα προηγούμενα τμήματα. Οι μετρικές αυτές επιτρέπουν την



αντικειμενική αποτίμηση της συμπεριφοράς του συστήματος και τη σύγκριση της απόδοσής του υπό διαφορετικές συνθήκες λειτουργίας.

Στο πλαίσιο της παρούσας εργασίας, οι μετρικές απόδοσης επιλέγονται ώστε να είναι άμεσα μετρήσιμες μέσω του ενσωματωμένου συστήματος και να σχετίζονται με τη λειτουργία και την αξιοπιστία του ελέγχου του ηλεκτρικού κινητήρα.

Οι βασικές μετρικές απόδοσης του συστήματος περιλαμβάνουν:

- **Ακρίβεια ελέγχου ταχύτητας**, η οποία εκφράζει την απόκλιση μεταξύ της επιθυμητής τιμής και της πραγματικής τιμής της ταχύτητας περιστροφής του κινητήρα.
- **Χρόνος απόκρισης**, δηλαδή ο απαιτούμενος χρόνος για την επίτευξη της επιθυμητής κατάστασης λειτουργίας μετά από μεταβολή του στόχου ή του φορτίου.
- **Σταθερότητα λειτουργίας**, που αφορά την ικανότητα του συστήματος να διατηρεί σταθερή συμπεριφορά χωρίς ανεπιθύμητες ταλαντώσεις.
- **Αξιοπιστία μετρήσεων**, η οποία σχετίζεται με τη συνέπεια και την εγκυρότητα των δεδομένων που συλλέγονται από τους αισθητήρες.
- **Κατανάλωση πόρων**, όπως η χρήση υπολογιστικής ισχύος και μνήμης, σε σχέση με τις δυνατότητες της ενσωματωμένης πλατφόρμας.
- **Διαθεσιμότητα συστήματος**, που εκφράζει την ικανότητα του συστήματος να παραμένει λειτουργικό σε συνθήκες συνεχούς λειτουργίας ή σε περιπτώσεις προσωρινής απώλειας επικοινωνίας με εξωτερικά συστήματα.
- **Ικανότητα ανίχνευσης αποκλίσεων**, δηλαδή η δυνατότητα εντοπισμού καταστάσεων εκτός προκαθορισμένων ορίων λειτουργίας, όπως αυξημένο ρεύμα ή θερμοκρασία.

Οι παραπάνω μετρικές χρησιμοποιούνται στο κεφάλαιο της αξιολόγησης για την παρουσίαση και ανάλυση των πειραματικών αποτελεσμάτων, επιτρέποντας την τεκμηριωμένη αποτίμηση της αποτελεσματικότητας του προτεινόμενου συστήματος.



4 Σχεδιασμός Συστήματος

4.1 Συνολική αρχιτεκτονική συστήματος

Η σχεδίαση του προτεινόμενου συστήματος βασίζεται στις απαιτήσεις και προδιαγραφές που καθορίστηκαν στο Κεφάλαιο 3, με κύριο στόχο την υλοποίηση ενός αξιόπιστου ενσωματωμένου συστήματος ελέγχου ηλεκτρικού κινητήρα, το οποίο υποστηρίζει παράλληλα παρακολούθηση βασικών μεγεθών λειτουργίας και απομακρυσμένη εποπτεία. Η αρχιτεκτονική έχει διαμορφωθεί ώστε ο τοπικός έλεγχος να παραμένει ο πυρήνας της λειτουργίας (real-time συμπεριφορά και προβλέψιμη απόκριση), ενώ οι λειτουργίες επικοινωνίας και αποστολής δεδομένων να λειτουργούν συμπληρωματικά και να μην επηρεάζουν τον βρόχο ελέγχου.

Σε υψηλό επίπεδο, το σύστημα αποτελείται από τα ακόλουθα υποσυστήματα:

1. **Υποσύστημα ελέγχου (μικροελεγκτής):** εκτελεί τον βασικό βρόχο ελέγχου, διαχειρίζεται τα περιφερειακά (PWM, ADC, timers) και συντονίζει τη ροή δεδομένων από τους αισθητήρες προς τις αντίστοιχες λειτουργίες.
2. **Υποσύστημα ισχύος / οδήγησης:** λαμβάνει το σήμα PWM από τον μικροελεγκτή και το μετατρέπει σε κατάλληλη οδήγηση για τον κινητήρα, εξασφαλίζοντας λειτουργία εντός επιτρεπτών ηλεκτρικών ορίων.
3. **Υποσύστημα αισθητήρων:** συλλέγει μετρήσεις που σχετίζονται με την κατάσταση λειτουργίας του κινητήρα (ενδεικτικά, ρεύμα κατανάλωσης, θερμοκρασία, ταχύτητα περιστροφής κ.α.).
4. **Υποσύστημα καταγραφής και γεγονότων:** οργανώνει τις μετρήσεις σε μορφή κατάλληλη για αξιολόγηση, διαχειρίζεται δεδομένα λειτουργίας και καταγράφει γεγονότα λειτουργίας (π.χ. υπέρβαση ορίων).
5. **Υποσύστημα επικοινωνίας / εποπτείας:** μεταδίδει επιλεγμένα δεδομένα σε εξωτερικό σύστημα (π.χ. υπολογιστή ή πλατφόρμα εποπτείας), χωρίς να παρεμβαίνει στη χρονική συμπεριφορά του ελέγχου.

Η λογική διασύνδεσης των παραπάνω υποσυστημάτων μπορεί να περιγραφεί ως εξής: οι αισθητήρες παρέχουν μετρήσεις στον μικροελεγκτή, ο οποίος εκτελεί περιοδικά τον βρόχο ελέγχου και παράγει σήμα PWM προς το στάδιο ισχύος, επιτυγχάνοντας την επιθυμητή συμπεριφορά του κινητήρα. Για τη ρύθμιση της ταχύτητας περιστροφής



χρησιμοποιείται βασικός αναλογικός–ολοκληρωτικός (PI ελεγκτής) μηχανισμός ελέγχου, ο οποίος προσαρμόζει το duty cycle του PWM με βάση το σφάλμα μεταξύ επιθυμητής και μετρούμενης ταχύτητας.

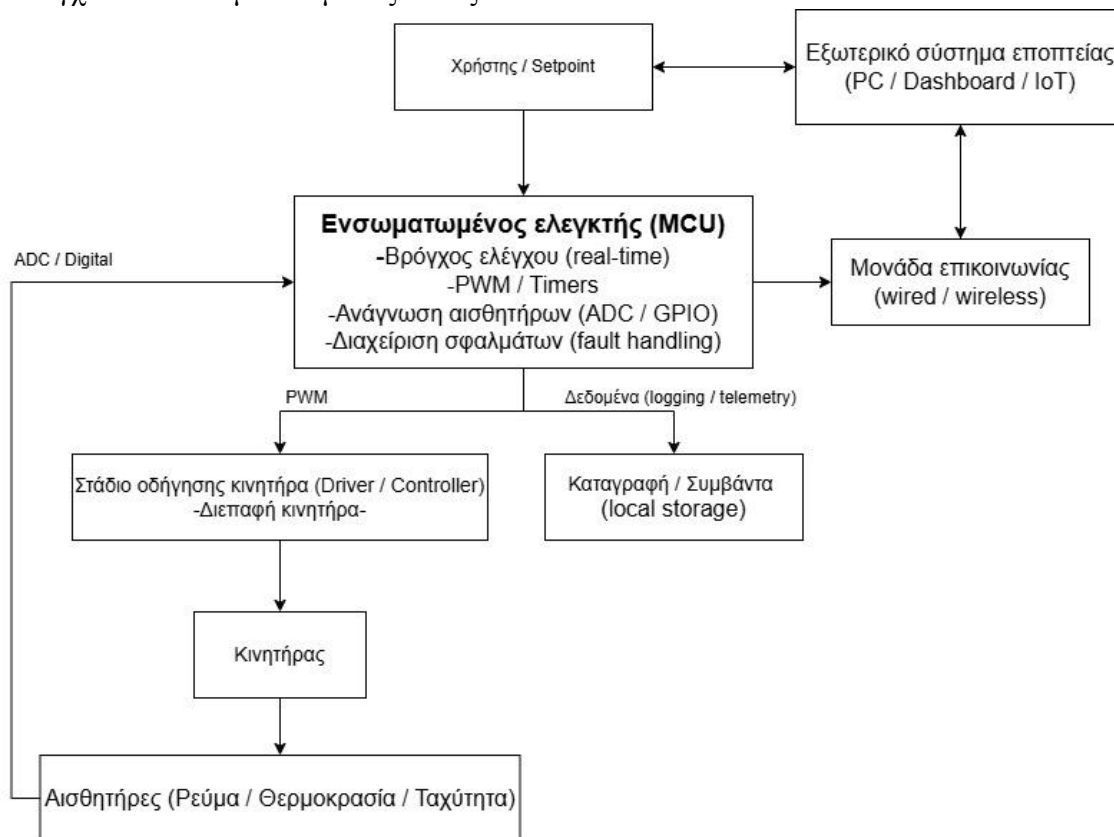
Παράλληλα, οι μετρήσεις επεξεργάζονται τοπικά και, όταν είναι διαθέσιμη επικοινωνία, μεταδίδονται προς το εξωτερικό σύστημα εποπτείας για παρακολούθηση και αξιολόγηση.

Κρίσιμο στοιχείο της αρχιτεκτονικής αποτελεί ο σαφής διαχωρισμός μεταξύ:

- λειτουργιών υψηλής προτεραιότητας (έλεγχος, ανάγνωση κρίσιμων αισθητήρων), και
- λειτουργιών δευτερεύουσας προτεραιότητας (επικοινωνία, αποστολή δεδομένων),

ώστε να διασφαλίζεται ότι ο έλεγχος παραμένει σταθερός και προβλέψιμος ακόμη και σε περιπτώσεις καθυστέρησης ή προσωρινής απώλειας επικοινωνίας.

Η συνολική αρχιτεκτονική του προτεινόμενου συστήματος παρουσιάζεται στην Εικόνα 4.1, όπου απεικονίζονται τα βασικά υποσυστήματα και η ροή σημάτων ελέγχου και δεδομένων μεταξύ τους.



Εικόνα 4.1. Μπλοκ διάγραμμα της προτεινόμενης αρχιτεκτονικής



Όπως φαίνεται στην Εικόνα 4.1, ο ενσωματωμένος ελεγκτής αποτελεί τον πυρήνα του συστήματος, διαχειριζόμενος τόσο τον βρόχο ελέγχου όσο και τη συλλογή και επεξεργασία δεδομένων. Το στάδιο οδήγησης λειτουργεί ως διεπαφή ισχύος μεταξύ του μικροελεγκτή και του κινητήρα, ενώ οι αισθητήρες παρέχουν τις αναγκαίες μετρήσεις για την υλοποίηση του κλειστού βρόχου ελέγχου. Οι λειτουργίες επικοινωνίας και καταγραφής λειτουργούν συμπληρωματικά, χωρίς να επηρεάζουν την πραγματικού χρόνου λειτουργία του συστήματος.

Στην παρούσα υλοποίηση επιλέγεται διαχωρισμός λειτουργιών σε δύο επίπεδα: α) επίπεδο ελέγχου, το οποίο υλοποιείται σε πλατφόρμα STM32 (NUCLEO-F446RE) και εκτελεί τον βρόχο ελέγχου σε πραγματικό χρόνο, β) επίπεδο επικοινωνίας/εποπτείας, το οποίο υλοποιείται σε ESP32 και λειτουργεί ως gateway προς την IoT πλατφόρμα. Ο διαχωρισμός αυτός διασφαλίζει ότι η ασφάλεια και ο έλεγχος παραμένουν τοπικά στο STM32, ενώ οι λειτουργίες τηλεμετρίας και απομακρυσμένων εντολών πραγματοποιούνται μέσω ThingsBoard.

Στην προτεινόμενη αρχιτεκτονική, ο STM32 λειτουργεί ως edge κόμβος, καθώς υλοποιεί τον βρόχο ελέγχου σε πραγματικό χρόνο, διαχειρίζεται τους μηχανισμούς προστασίας και λαμβάνει τις κρίσιμες αποφάσεις που σχετίζονται με τη λειτουργία του κινητήρα. Η λειτουργία αυτή παραμένει πλήρως ανεξάρτητη από το επίπεδο επικοινωνίας και την πλατφόρμα εποπτείας.

Ως edge ορίζεται το επίπεδο του συστήματος που βρίσκεται πλησιέστερα στο φυσικό περιβάλλον, όπου πραγματοποιείται η άμεση απόκτηση δεδομένων και η τοπική λήψη αποφάσεων με απαιτήσεις χρονικής προβλεψιμότητας. Το επίπεδο αυτό λειτουργεί αυτόνομα από την υποδομή cloud και διατηρεί την κρίσιμη λειτουργία του συστήματος ακόμη και σε περίπτωση απώλειας δικτύου.

Η ESP32 λειτουργεί ως IoT gateway, αναλαμβάνοντας αποκλειστικά τη διαχείριση της επικοινωνίας με το δίκτυο και τη μεταφορά τηλεμετρίας προς την πλατφόρμα ThingsBoard. Το cloud επίπεδο επιτρέπει την οπτικοποίηση, καταγραφή και απομακρυσμένη εποπτεία, χωρίς να επηρεάζει την τοπική λειτουργία ελέγχου.



4.2 Σχεδιασμός υλικού

4.2.1 Επιλογή ενσωματωμένης πλατφόρμας (MCU) και κριτήρια

Η επιλογή της ενσωματωμένης πλατφόρμας ελέγχου αποτελεί κρίσιμη σχεδιαστική απόφαση, καθώς επηρεάζει άμεσα τη δυνατότητα υλοποίησης του βρόχου ελέγχου σε πραγματικό χρόνο, την αξιοπιστία της συλλογής μετρήσεων και τη συνολική επεκτασιμότητα του συστήματος. Στην παρούσα εργασία, η πλατφόρμα επιλέγεται με βάση τεχνικά κριτήρια που απορρέουν από τις λειτουργικές και μη λειτουργικές απαιτήσεις του Κεφαλαίου 3, με έμφαση στην προβλέψιμη χρονική συμπεριφορά και στην επάρκεια των διαθέσιμων περιφερειακών.

Βασικό κριτήριο αποτελεί η ύπαρξη κατάλληλων timers και μονάδων παραγωγής PWM, ώστε να είναι δυνατή η σταθερή παραγωγή σήματος οδήγησης προς το στάδιο οδήγησης του κινητήρα, με επαρκή ανάλυση (resolution) και δυνατότητα ρύθμισης συχνότητας. Παράλληλα, απαιτείται ενσωματωμένος ADC με κατάλληλη ανάλυση και ρυθμό δειγματοληψίας, ώστε να υποστηρίζεται η αξιόπιστη ανάγνωση αναλογικών μεγεθών, όπως το ρεύμα κατανάλωσης ή οι μετρήσεις θερμοκρασίας, χωρίς να επιβαρύνεται η χρονική λειτουργία του ελέγχου.

Επιπλέον, η πλατφόρμα πρέπει να διαθέτει επαρκείς πόρους μνήμης (RAM/Flash) για την υλοποίηση της αρθρωτής δομής λογισμικού (modules), όπως περιγράφεται στις προδιαγραφές λογισμικού, καθώς και για βασικές λειτουργίες επεξεργασίας και διαχείρισης μετρήσεων και γεγονότων. Η ύπαρξη μηχανισμών διακοπών (interrupts) και δυνατότητα καθορισμού προτεραιοτήτων είναι επίσης σημαντική, καθώς επιτρέπει τον σαφή διαχωρισμό λειτουργιών υψηλής προτεραιότητας (έλεγχος/μετρήσεις) από λειτουργίες δευτερεύουσας προτεραιότητας (επικοινωνία/αποστολή δεδομένων), σύμφωνα με την αρχιτεκτονική στο Κεφάλαιο 4.1.

Η πλατφόρμα πρέπει επίσης να υποστηρίζει μηχανισμούς ακριβούς χρονισμού και διακοπών για την αξιόπιστη ανάγνωση παλμών του encoder και για την υλοποίηση κλειστού βρόχου ελέγχου.

Τέλος, λαμβάνεται υπόψη η πρακτική δυνατότητα ανάπτυξης και δοκιμών, όπως η διαθεσιμότητα εργαλείων προγραμματισμού/αποσφαλμάτωσης και η ευκολία πρόσβασης στα βασικά περιφερειακά, ώστε το σύστημα να μπορεί να υλοποιηθεί και να αξιολογηθεί με τρόπο επαναλήψιμο και τεκμηριωμένο.



Με βάση τα παραπάνω κριτήρια, επιλέγεται η πλατφόρμα NUCLEO-F446RE (STM32F446RE) ως κύρια μονάδα ελέγχου, ενώ η διασύνδεση με δίκτυο και cloud υλοποιείται από ανεξάρτητη μονάδα ESP32.

4.2.2 Στάδιο οδήγησης κινητήρα και διεπαφή PWM

Το στάδιο οδήγησης κινητήρα αποτελεί το υποσύστημα ισχύος που διασυνδέει τον ενσωματωμένο ελεγκτή (MCU) με τον ηλεκτρικό κινητήρα. Δεδομένου ότι ο μικροελεγκτής λειτουργεί σε χαμηλά επίπεδα τάσης και ρεύματος, δεν είναι δυνατόν να οδηγήσει απευθείας τον κινητήρα. Για τον λόγο αυτό, παρεμβάλλεται ενδιάμεση μονάδα οδήγησης (motor driver), η οποία μετατρέπει το σήμα ελέγχου χαμηλής ισχύος σε κατάλληλο σήμα ισχύος για τη λειτουργία του κινητήρα.

Η διεπαφή μεταξύ MCU και driver βασίζεται σε σήμα PWM, το οποίο παράγεται από κατάλληλη μονάδα timer του μικροελεγκτή. Η μεταβολή του duty cycle του PWM επιτρέπει τον έλεγχο της μέσης εφαρμοζόμενης τάσης στον κινητήρα και, κατ' επέκταση, της ταχύτητας περιστροφής του. Η συχνότητα του PWM επιλέγεται ώστε να επιτυγχάνεται ικανοποιητική ποιότητα ελέγχου, αποφεύγοντας παράλληλα ανεπιθύμητες ακουστικές ή ηλεκτρομαγνητικές επιδράσεις. Το PWM παράγεται από hardware timer του STM32, επιτρέποντας ακριβή και σταθερή χρονική παραγωγή του σήματος οδήγησης.

Το στάδιο οδήγησης υλοποιείται μέσω ολοκληρωμένης μονάδας driver ισχύος (VNH5019 motor driver), η οποία ενσωματώνει MOSFET στάδια ισχύος και βασικούς μηχανισμούς προστασίας και η οποία είναι ικανή να διαχειριστεί τα απαιτούμενα επίπεδα τάσης και ρεύματος του κινητήρα. Η μονάδα αυτή παρέχει:

- ηλεκτρική συμβατότητα με τα επίπεδα λογικής του MCU,
- ικανότητα διαχείρισης των ρευμάτων εκκίνησης και φορτίου,
- βασικούς μηχανισμούς προστασίας (π.χ. προστασία υπέρντασης ή θερμική προστασία),
- υποστηρίζει επίσης τον καθορισμό της φοράς περιστροφής μέσω κατάλληλων ψηφιακών σημάτων ελέγχου.

Η έξοδος του σταδίου οδήγησης προσαρμόζεται δυναμικά από τον βρόχο ελέγχου του μικροελεγκτή, με βάση τις μετρήσεις ταχύτητας και λειτουργικής κατάστασης του κινητήρα.



Ιδιαίτερη έμφαση δίνεται στον σωστό διαχωρισμό των κυκλωμάτων ισχύος από τα κυκλώματα ελέγχου, ώστε να αποφεύγεται η μεταφορά θορύβου ή παρεμβολών προς τον μικροελεγκτή. Ο σχεδιασμός αυτός ενισχύει τη σταθερότητα του συστήματος και διασφαλίζει την αξιόπιστη λειτουργία του βρόχου ελέγχου, ακόμη και υπό μεταβαλλόμενες συνθήκες φορτίου.

Συνολικά, το στάδιο οδήγησης λειτουργεί ως κρίσιμος ενδιάμεσος κρίκος μεταξύ λογικού ελέγχου και ηλεκτρομηχανικής ισχύος, επιτρέποντας την ασφαλή και αποδοτική υλοποίηση του ελέγχου του κινητήρα σύμφωνα με τις απαιτήσεις που έχουν τεθεί.

4.2.3 Αισθητήρες και σήματα μέτρησης

Η υλοποίηση κλειστού βρόχου ελέγχου προϋποθέτει τη συλλογή αξιόπιστων μετρήσεων από τον κινητήρα και το σύστημα ισχύος. Το υποσύστημα αισθητήρων παρέχει τις απαραίτητες πληροφορίες προς τον ενσωματωμένο ελεγκτή (MCU), επιτρέποντας τόσο την προσαρμογή του σήματος ελέγχου όσο και την καταγραφή λειτουργικών μεγεθών για σκοπούς αξιολόγησης.

Στο πλαίσιο της παρούσας εργασίας, λαμβάνονται υπόψη οι ακόλουθες βασικές κατηγορίες μετρήσεων:

- Μέτρηση ρεύματος κατανάλωσης, η οποία επιτρέπει την παρακολούθηση του φορτίου του κινητήρα και μπορεί να χρησιμοποιηθεί για ανίχνευση ανωμαλιών ή υπερφορτίσεων.
- Μέτρηση θερμοκρασίας, είτε στο περίβλημα του κινητήρα είτε στο στάδιο οδήγησης, με στόχο την προστασία του συστήματος από θερμική καταπόνηση.
- Μέτρηση ταχύτητας περιστροφής, μέσω κατάλληλου αισθητήρα (π.χ. ενσωματωμένο encoder ή εξωτερικό αισθητήρα Hall), για την υλοποίηση ή επαλήθευση του ελέγχου ταχύτητας.
- Διάφορες μετρήσεις που αφορούν κραδασμούς, τάση τροφοδοσίας κ.α.

Τα σήματα των αισθητήρων μπορούν να είναι είτε αναλογικά είτε ψηφιακά χρησιμοποιώντας και πρωτόκολλα επικοινωνίας. Τα αναλογικά σήματα οδηγούνται σε μετατροπέα ADC του μικροελεγκτή, ενώ τα ψηφιακά σήματα διαβάζονται μέσω κατάλληλων θυρών εισόδου (GPIO ή εξειδικευμένων περιφερειακών). Η επιλογή της μορφής σήματος γίνεται με γνώμονα την ακρίβεια μέτρησης, την απλότητα διασύνδεσης και τη συμβατότητα με τα διαθέσιμα περιφερειακά της πλατφόρμας.



Ιδιαίτερη σημασία έχει η σωστή δειγματοληψία των μετρήσεων, ώστε να διασφαλίζεται ότι οι τιμές που χρησιμοποιούνται στον βρόχο ελέγχου είναι αντιπροσωπευτικές της πραγματικής κατάστασης του συστήματος. Η συχνότητα δειγματοληψίας επιλέγεται λαμβάνοντας υπόψη τη δυναμική του κινητήρα και τις απαιτήσεις του ελέγχου, αποφεύγοντας ταυτόχρονα περιττή επιβάρυνση του επεξεργαστή.

Πέραν της χρήσης τους στον έλεγχο, οι μετρήσεις αξιοποιούνται και για σκοπούς καταγραφής και αξιολόγησης, όπως ορίστηκε στις μετρικές απόδοσης του Κεφαλαίου 3. Η ύπαρξη αξιόπιστων δεδομένων μέτρησης επιτρέπει την τεκμηριωμένη αποτίμηση της ακρίβειας ελέγχου, του χρόνου απόκρισης και της σταθερότητας λειτουργίας του συστήματος.

Με τον τρόπο αυτό, το υποσύστημα αισθητήρων δεν λειτουργεί μόνο ως πηγή ανάδρασης για τον έλεγχο, αλλά και ως βασικό εργαλείο τεκμηρίωσης της απόδοσης του προτεινόμενου ενσωματωμένου συστήματος.

4.2.4 Τροφοδοσία και προστασίες

Η ορθή σχεδίαση του υποσυστήματος τροφοδοσίας αποτελεί κρίσιμο παράγοντα για την αξιόπιστη λειτουργία του ενσωματωμένου συστήματος ελέγχου. Δεδομένου ότι το σύστημα περιλαμβάνει τόσο κυκλώματα ελέγχου χαμηλής ισχύος (MCU, αισθητήρες, μονάδα επικοινωνίας) όσο και κυκλώματα ισχύος (driver και κινητήρας), απαιτείται σαφής διαχωρισμός και κατάλληλη σταθεροποίηση των επιμέρους επιπέδων τάσης.

Το στάδιο οδήγησης κινητήρα τροφοδοτείται από πηγή ισχύος κατάλληλη για τις απαιτήσεις τάσης και ρεύματος του κινητήρα, ενώ ο μικροελεγκτής και τα περιφερειακά του λειτουργούν σε χαμηλότερα επίπεδα τάσης (π.χ. 3.3V ή 5V). Για τον σκοπό αυτό χρησιμοποιούνται κατάλληλοι μετατροπείς τάσης (π.χ. DC-DC converters ή γραμμικοί ρυθμιστές), οι οποίοι διασφαλίζουν σταθερή και καθαρή παροχή τάσης προς τα κυκλώματα ελέγχου.

Ιδιαίτερη προσοχή δίνεται στη γείωση και στη διάταξη των γραμμών τροφοδοσίας, ώστε να περιορίζεται η μεταφορά ηλεκτρικού θορύβου από το στάδιο ισχύος προς το στάδιο ελέγχου. Ο διαχωρισμός αναλογικής και ψηφιακής γείωσης, όπου απαιτείται, συμβάλλει στη βελτίωση της ακρίβειας των μετρήσεων μέσω ADC και στη σταθερότητα του συστήματος.

Παράλληλα, το σύστημα πρέπει να ενσωματώνει βασικούς μηχανισμούς προστασίας, όπως:



- προστασία υπερέντασης στο στάδιο οδήγησης,
- προστασία από υπέρταση ή πτώση τάσης (under/over-voltage conditions),
- θερμική προστασία σε κρίσιμα στοιχεία ισχύος,
- κατάλληλη ασφάλεια ή περιοριστή ρεύματος στην είσοδο τροφοδοσίας.

Η ύπαρξη αυτών των μηχανισμών ενισχύει τη λειτουργική ασφάλεια και περιορίζει την πιθανότητα καταστροφής εξαρτημάτων σε περίπτωση ανωμαλίας. Επιπλέον, συμβάλλει στη σταθερή και επαναλήψιμη συμπεριφορά του συστήματος κατά τη διάρκεια των πειραματικών δοκιμών και της αξιολόγησης.

Συνολικά, ο σχεδιασμός της τροφοδοσίας και των προστασιών συμπληρώνει τον σχεδιασμό του σταδίου οδήγησης και της ενσωματωμένης πλατφόρμας, διαμορφώνοντας ένα ολοκληρωμένο και λειτουργικά συνεκτικό υποσύστημα υλικού.

4.2.5 Επιλογή αισθητήρων και διασύνδεση σημάτων

Η επιλογή των αισθητήρων βασίστηκε στις λειτουργικές απαιτήσεις που διατυπώθηκαν στο Κεφάλαιο 3, καθώς και στην ανάγκη υλοποίησης ενός αξιόπιστου κλειστού βρόχου ελέγχου σε πραγματικό χρόνο. Στόχος είναι η μέτρηση βασικών ηλεκτρικών και μηχανικών μεγεθών του κινητήρα με τρόπο τεχνικά ρεαλιστικό, επαναλήψιμο και πλήρως συμβατό με την επιλεγμένη πλατφόρμα ελέγχου STM32 (NUCLEO-F446RE).

Στο προτεινόμενο σύστημα επιλέγονται οι ακόλουθες κατηγορίες μετρήσεων:

i. Μέτρηση ρεύματος – ACS712 (Hall effect sensor)

Για τη μέτρηση του ρεύματος κατανάλωσης του κινητήρα χρησιμοποιείται αισθητήρας τύπου ACS712, βασισμένος στο φαινόμενο Hall, ο οποίος παρέχει αναλογική έξοδο τάσης ανάλογη του ρεύματος που διαρρέει τον αγωγό ισχύος.

Η επιλογή Hall-based αισθητήρα επιτρέπει:

- ηλεκτρική απομόνωση μεταξύ κυκλώματος ισχύος και κυκλώματος ελέγχου,
- απλή διασύνδεση με κανάλι ADC του STM32,
- επαρκή ακρίβεια για σκοπούς παρακολούθησης και ανίχνευσης υπερέντασης.

Λαμβάνεται υπόψη ότι ο κινητήρας μπορεί να εμφανίσει αιχμές ρεύματος έως περίπου 7A κατά την εκκίνηση ή σε κατάσταση stall. Για τον λόγο αυτό επιλέγεται έκδοση αισθητήρα με κατάλληλο εύρος μέτρησης ώστε να καλύπτονται οι αναμενόμενες δυναμικές μεταβολές χωρίς κορεσμό του σήματος.



Η αναλογική έξοδος του αισθητήρα οδηγείται σε είσοδο ADC του STM32, όπου πραγματοποιείται δειγματοληψία και μετατροπή σε μονάδες ρεύματος (A) μέσω κατάλληλης κλιμάκωσης στο λογισμικό.

ii. Μέτρηση θερμοκρασίας – DS18B20

Για τη θερμική παρακολούθηση του κινητήρα χρησιμοποιείται ψηφιακός αισθητήρας θερμοκρασίας τύπου DS18B20. Ο αισθητήρας επιλέχθηκε λόγω της απλής διασύνδεσης, της αξιόπιστης λειτουργίας και της δυνατότητας άμεσης ψηφιακής μέτρησης θερμοκρασίας χωρίς ανάγκη επιπλέον αναλογικών κυκλωμάτων μετατροπής.

Η χρήση του DS18B20 προσφέρει:

- ψηφιακή μέτρηση θερμοκρασίας με ικανοποιητική ακρίβεια,
- μειωμένη ευαισθησία σε αναλογικό θόρυβο,
- απλή ενσωμάτωση σε ενσωματωμένα συστήματα και IoT εφαρμογές,
- δυνατότητα συνεχούς παρακολούθησης της θερμικής κατάστασης του συστήματος.

Η επικοινωνία με τον αισθητήρα πραγματοποιείται μέσω του πρωτοκόλλου 1-Wire, το οποίο απαιτεί τη χρήση αντίστασης pull-up στη γραμμή δεδομένων για τη σωστή λειτουργία και σταθεροποίηση του σήματος επικοινωνίας. Οι μετρήσεις θερμοκρασίας επεξεργάζονται από το λογισμικό του μικροελεγκτή και χρησιμοποιούνται τόσο για την εποπτεία της λειτουργίας του κινητήρα όσο και για την ενεργοποίηση μηχανισμών alarm σε περιπτώσεις υπερθέρμανσης.

*Εναλλακτικά για τη μέτρηση της θερμοκρασίας θα μπορούσε να χρησιμοποιηθεί και θερμίστορ NTC 10kΩ (B=3950)

iii. Μέτρηση κραδασμών / επιτάχυνσης – MPU6050

Για την παρακολούθηση κραδασμών και μηχανικών μεταβολών του κινητήρα χρησιμοποιείται ο αισθητήρας MPU6050, ο οποίος αποτελεί μονάδα αδρανειακής μέτρησης (IMU) με ενσωματωμένο τριαξονικό επιταχυνσιόμετρο και τριαξονικό γυροσκόπιο. Ο αισθητήρας διαθέτει μετατροπείς ADC 16-bit για κάθε άξονα μέτρησης και επικοινωνεί με τον STM32 μέσω διαύλου I2C.^[MPU-6050 Product Specification]

Στο πλαίσιο της παρούσας υλοποίησης, οι μετρήσεις του αισθητήρα χρησιμοποιούνται για την παρακολούθηση κραδασμών και μη φυσιολογικών μεταβολών της λειτουργίας του κινητήρα. Οι τιμές επιτάχυνσης αποστέλλονται μέσω τηλεμετρίας προς την πλατφόρμα ThingsBoard, συμβάλλοντας στην απομακρυσμένη εποπτεία της



κατάστασης του συστήματος και στην ενεργοποίηση alarm σε περιπτώσεις αυξημένων κραδασμών ή ασυνήθιστης μηχανικής συμπεριφοράς.

iv. Μέτρηση τάσης τροφοδοσίας – Διαιρέτης τάσης

Για την παρακολούθηση της τάσης τροφοδοσίας του κινητήρα χρησιμοποιείται απλός διαιρέτης τάσης, κατάλληλα διαστασιοποιημένος ώστε η έξοδος να βρίσκεται εντός των ορίων εισόδου του ADC του STM32.

Η μέτρηση αυτή επιτρέπει:

- έλεγχο συνθηκών υπότασης ή υπέρτασης,
- συσχέτιση της τάσης τροφοδοσίας με τη δυναμική συμπεριφορά του κινητήρα,
- ενεργοποίηση μηχανισμών προστασίας σε ακραίες συνθήκες λειτουργίας.

Η επιλογή των αντιστάσεων γίνεται ώστε να διασφαλίζεται ασφαλής λειτουργία του μικροελεγκτή και σταθερότητα μέτρησης.

v. Μέτρηση ταχύτητας περιστροφής – Encoder

Ο επιλεγμένος κινητήρας διαθέτει ενσωματωμένο encoder, ο οποίος αξιοποιείται για την εκτίμηση της ταχύτητας περιστροφής (RPM) και την υλοποίηση βασικού κλειστού βρόχου ελέγχου ταχύτητας. Η μέτρηση πραγματοποιείται μέσω καταμέτρησης παλμών από τον STM32 με χρήση interrupts, ενώ στο firmware χρησιμοποιείται κατάλληλη τιμή pulses per revolution (PPR) για τον υπολογισμό των RPM σύμφωνα με τα χαρακτηριστικά του encoder και του συστήματος μετάδοσης του κινητήρα.

Τα σήματα του encoder οδηγούνται σε ψηφιακές εισόδους του STM32 και μετρώνται μέσω timer/counter ή input capture μηχανισμού, επιτρέποντας:

- ακριβή υπολογισμό ταχύτητας,
- εκτίμηση επιτάχυνσης,
- βελτιωμένη ρύθμιση παραμέτρων PI,
- ανάλυση μεταβατικής απόκρισης.

Η υψηλή ανάλυση του encoder ενισχύει τη δυνατότητα πειραματικής αξιολόγησης της απόδοσης του ελέγχου και καθιστά το σύστημα κατάλληλο για μελέτη δυναμικής συμπεριφοράς.

vi. Αισθητήρας προσέγγισης – Inductive Proximity Sensor LJ8A3-2-Z/AX

Για την ανίχνευση της περιστροφικής κίνησης και της παρουσίας μεταλλικού αντικειμένου στον άξονα του κινητήρα χρησιμοποιείται επαγωγικός αισθητήρας προσέγγισης τύπου LJ8A3-2-Z/AX. Ο αισθητήρας επιλέχθηκε λόγω της αξιοπιστίας



λειτουργίας του σε εφαρμογές βιομηχανικού τύπου και της δυνατότητας ανίχνευσης χωρίς μηχανική επαφή.

Η χρήση του αισθητήρα προσφέρει:

- αξιόπιστη ανίχνευση μεταλλικών αντικειμένων,
- λειτουργία χωρίς μηχανική φθορά,
- σταθερή λειτουργία σε υψηλές ταχύτητες περιστροφής,
- δυνατότητα χρήσης σε εφαρμογές παρακολούθησης κίνησης και στροφών.

Για την ασφαλή διασύνδεση του αισθητήρα με το σύστημα ελέγχου χρησιμοποιείται οπτοζεύκτης (optocoupler), ο οποίος παρέχει ηλεκτρική απομόνωση μεταξύ του αισθητήρα και του μικροελεγκτή. Με τον τρόπο αυτό προστατεύεται το κύκλωμα ελέγχου από ηλεκτρικό θόρυβο, μεταβατικά φαινόμενα και πιθανές διαφορές τάσης. Το σήμα του αισθητήρα αξιοποιείται από το λογισμικό του συστήματος για την παρακολούθηση της περιστροφής του κινητήρα και την ενεργοποίηση λειτουργιών εποπτείας και alarm σε περιπτώσεις απώλειας ανίχνευσης.

vii. Συνολική διασύνδεση και αξιοποίηση δεδομένων

Οι αναλογικοί αισθητήρες, όπως ο αισθητήρας ρεύματος και το κύκλωμα μέτρησης τάσης μέσω διαιρέτη τάσης, συνδέονται σε κανάλια ADC του STM32. Ο αισθητήρας θερμοκρασίας DS18B20 επικοινωνεί ψηφιακά με τον μικροελεγκτή, ενώ ο αισθητήρας MPU6050 συνδέεται μέσω διαύλου I2C για τη μέτρηση επιτάχυνσης και κραδασμών. Ο encoder και ο επαγωγικός αισθητήρας προσέγγισης συνδέονται σε ψηφιακές εισόδους του STM32 με δυνατότητα καταμέτρησης και χρονικής μέτρησης παλμών. Η επεξεργασία των μετρήσεων πραγματοποιείται στο module απόκτησης δεδομένων του λογισμικού (βλ. 4.3.1). Τα δεδομένα χρησιμοποιούνται:

- στον βρόχο ελέγχου του κινητήρα,
- στον μηχανισμό προστασίας (υπερένταση, υπερθέρμανση, ανωμαλία τάσης),
- στη λειτουργία καταγραφής και τηλεμετρίας.

Η τηλεμετρία προωθείται προς το επίπεδο IoT μέσω της μονάδας ESP32, η οποία λειτουργεί ως gateway επικοινωνίας και υποστηρίζει αποστολή δεδομένων μέσω MQTT προς την πλατφόρμα ThingsBoard για απομακρυσμένη καταγραφή, εποπτεία και οπτικοποίηση.

Με τον τρόπο αυτό διαμορφώνεται ένα ολοκληρωμένο και τεχνικά συνεκτικό υποσύστημα μετρήσεων, πλήρως εναρμονισμένο με την επιλεγμένη αρχιτεκτονική



STM32–ESP32 και κατάλληλο για πειραματική αξιολόγηση και τεκμηρίωση της απόδοσης του προτεινόμενου συστήματος.

4.2.6 Διασύνδεση STM32 – ESP32 και πρωτόκολλο επικοινωνίας

Η αρχιτεκτονική του προτεινόμενου συστήματος βασίζεται σε διαχωρισμό λειτουργιών μεταξύ επιπέδου ελέγχου και επιπέδου επικοινωνίας. Ο μικροελεγκτής STM32 (NUCLEO-F446RE) αναλαμβάνει αποκλειστικά τις λειτουργίες πραγματικού χρόνου (real-time motor control και προστασία), ενώ η μονάδα ESP32 λειτουργεί ως gateway επικοινωνίας προς το δίκτυο και την IoT πλατφόρμα (ThingsBoard).

Η διασύνδεση μεταξύ STM32 και ESP32 υλοποιείται μέσω σειριακής επικοινωνίας UART, λόγω της απλότητας, της αξιοπιστίας και της επαρκούς ταχύτητας για μεταφορά τηλεμετρίας και εντολών.

i. Ροή δεδομένων από STM32 προς ESP32

Ο STM32 αποστέλλει περιοδικά πακέτα τηλεμετρίας που περιλαμβάνουν:

- Ταχύτητα περιστροφής (RPM)
- Ρεύμα κινητήρα (A)
- Θερμοκρασία (°C)
- Τάση τροφοδοσίας (V)
- Λοιπές μετρήσεις
- Καταστάσεις προστασίας ή ενδείξεις υπέρβασης ορίων λειτουργίας

Η μετάδοση πραγματοποιείται περιοδικά, με ρυθμό κατάλληλο για απομακρυσμένη εποπτεία, χωρίς να επηρεάζεται ο βρόχος ελέγχου. Η αποστολή γίνεται σε μορφή απλού πλαισίου δεδομένων (frame), με καθορισμένη δομή ώστε να επιτυγχάνεται αξιόπιστη ανάλυση από την ESP32.

Ενδεικτικά, το πλαίσιο μπορεί να έχει μορφή:

```
RPM_A=...;RPM_B=...;TARGET=...;ERR=...;DUTY=...;CUR=...;VOLT=...;TEMP=...;  
AX=...;AY=...;AZ=...;PROX=...
```

Η επιλογή απλού πρωτοκόλλου επιτρέπει ευκολία αποσφαλμάτωσης και χαμηλό υπολογιστικό κόστος στον STM32.

ii. Ροή δεδομένων από ESP32 προς STM32

Η ESP32 δύναται να αποστέλλει εντολές προς τον STM32, όπως:

- Νέα τιμή setpoint (RPM)
- Ενεργοποίηση/απενεργοποίηση κινητήρα



- Reset σφαλμάτων

Οι εντολές αυτές προέρχονται είτε από τοπικό σύστημα (π.χ. PC) είτε από την IoT πλατφόρμα μέσω MQTT RPC μηχανισμού. Ο STM32 ελέγχει την εγκυρότητα των εντολών και εφαρμόζει τις αλλαγές μόνο εφόσον δεν παραβιάζονται συνθήκες ασφαλείας.

Ιδιαίτερη σημασία δίνεται στο γεγονός ότι η λειτουργία ασφάλειας παραμένει αποκλειστικά στο STM32. Σε περίπτωση απώλειας επικοινωνίας με την ESP32 ή το cloud, ο έλεγχος συνεχίζει κανονικά με τις τελευταίες έγκυρες παραμέτρους.

iii. Ενσωμάτωση με ThingsBoard

Η ESP32 αναλαμβάνει:

- Σύνδεση σε δίκτυο WiFi
- Δημοσίευση τηλεμετρίας μέσω MQTT
- Λήψη RPC εντολών
- Μετατροπή δεδομένων σε μορφή συμβατή με ThingsBoard

Η πλατφόρμα ThingsBoard παρέχει:

- Ζωντανή απεικόνιση (live monitoring)
- Καταγραφή ιστορικών δεδομένων
- Ορισμό alarm conditions
- Απομακρυσμένο έλεγχο μέσω dashboard

Ο διαχωρισμός αυτός επιτρέπει την πλήρη αποσύνδεση της λειτουργίας ελέγχου από τη λειτουργία δικτύου, εξασφαλίζοντας προβλέψιμη χρονική συμπεριφορά στο επίπεδο ελέγχου.

iv. Αρχιτεκτονική Αξιοπιστίας

Η συνολική προσέγγιση σχεδιασμού βασίζεται στην αρχή:

Ο έλεγχος και η ασφάλεια παραμένουν τοπικά, ενώ η επικοινωνία λειτουργεί επικουρικά.^[Wolf]

Σε περίπτωση:

- Απώλειας WiFi
- Αποτυχίας σύνδεσης MQTT
- Καθυστέρησης cloud



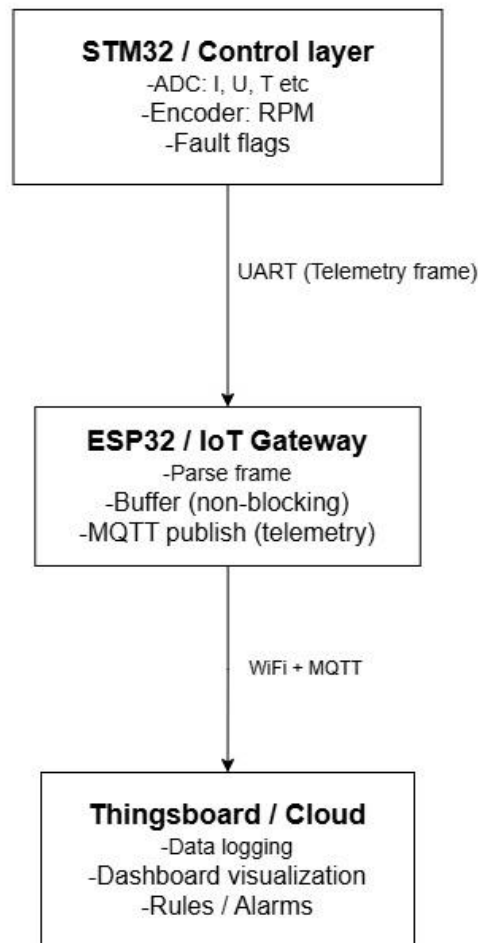
το σύστημα συνεχίζει να λειτουργεί κανονικά υπό τον έλεγχο του STM32. Οι μηχανισμοί προστασίας (υπερένταση, υπερθέρμανση, ανωμαλία τάσης) λειτουργούν ανεξάρτητα από το IoT επίπεδο.

Με τον τρόπο αυτό επιτυγχάνεται σύγχρονη, επεκτάσιμη και τεχνικά ορθή αρχιτεκτονική, κατάλληλη για εργαστηριακή υλοποίηση αλλά και για προσομοίωση μεγαλύτερων βιομηχανικών συστημάτων.

Η ροή τηλεμετρίας από το επίπεδο ελέγχου προς την πλατφόρμα IoT ακολουθεί διαδοχικά στάδια μετάδοσης, από τον μικροελεγκτή STM32 προς τη μονάδα ESP32 και στη συνέχεια προς την πλατφόρμα ThingsBoard μέσω MQTT.

Ο STM32 συλλέγει τις μετρήσεις (ρεύμα, τάση, θερμοκρασία, RPM) και τις οργανώνει σε πλαίσιο τηλεμετρίας (telemetry frame), το οποίο μεταδίδεται σειριακά μέσω UART. Η ESP32 αναλαμβάνει την ανάλυση (parsing) και προσωρινή αποθήκευση των δεδομένων σε buffer, πριν την αποστολή τους στο cloud μέσω WiFi και MQTT.

Η συνολική ροή παρουσιάζεται στην Εικόνα 4.2.



Εικόνα 4.2. Διάγραμμα τηλεμετρίας

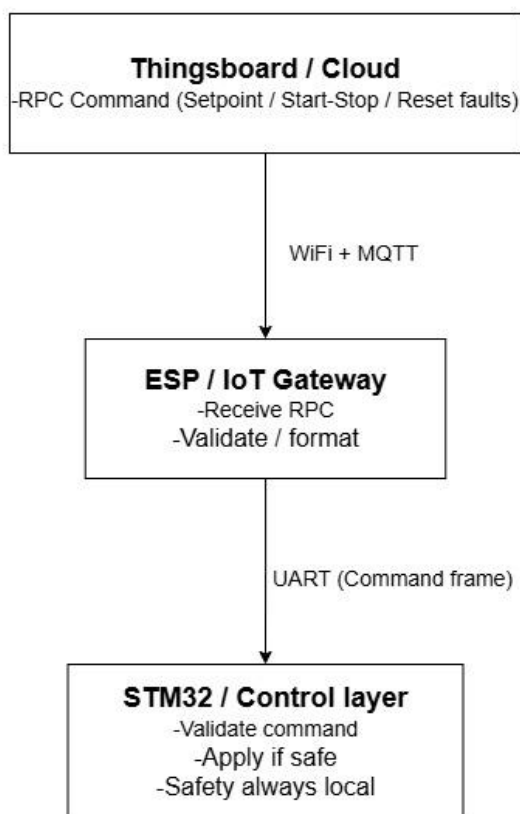
Όπως φαίνεται στην Εικόνα 4.2, η μετάδοση δεδομένων πραγματοποιείται με σαφή διαχωρισμό επιπέδων. Ο STM32 περιορίζεται στις λειτουργίες ελέγχου και απόκτησης μετρήσεων, ενώ η ESP32 λειτουργεί ως gateway επικοινωνίας. Με τον τρόπο αυτό διασφαλίζεται ότι η δικτυακή δραστηριότητα δεν επηρεάζει τη χρονική συμπεριφορά του βρόχου ελέγχου.

Εκτός από τη ροή τηλεμετρίας, το σύστημα υποστηρίζει και απομακρυσμένες εντολές (Remote Procedure Calls – RPC) από την πλατφόρμα ThingsBoard προς τον ελεγκτή.



Οι εντολές αυτές μπορεί να αφορούν αλλαγή setpoint, εκκίνηση ή παύση λειτουργίας, καθώς και επαναφορά κατάστασης σφάλματος.

Η ESP32 λαμβάνει τα μηνύματα μέσω MQTT, τα μετατρέπει σε κατάλληλες σειριακές εντολές και τα προωθεί στον STM32 μέσω UART. Η ροή αυτή παρουσιάζεται στην Εικόνα 4.3.



Εικόνα 4.3. Διάγραμμα εντολών

Ο STM32 ελέγχει την εγκυρότητα και τη συμβατότητα κάθε εντολής με τις συνθήκες ασφαλείας του συστήματος πριν την εφαρμογή της. Σε περίπτωση απώλειας επικοινωνίας ή αποτυχίας σύνδεσης με το cloud, ο βρόχος ελέγχου συνεχίζει να λειτουργεί κανονικά με τις τελευταίες έγκυρες παραμέτρους ή με ασφαλή προεπιλεγμένη κατάσταση. Οι μηχανισμοί προστασίας (υπερένταση, υπερθέρμανση, ανωμαλία τάσης) παραμένουν ενεργοί και ανεξάρτητοι από το επίπεδο επικοινωνίας.



4.3 Σχεδιασμός λογισμικού

Το παρόν υποκεφάλαιο περιγράφει τον σχεδιασμό του λογισμικού του προτεινόμενου ενσωματωμένου συστήματος, το οποίο υλοποιεί τον βρόχο ελέγχου του κινητήρα και υποστηρίζει λειτουργίες μέτρησης, καταγραφής και επικοινωνίας. Ο σχεδιασμός βασίζεται στις απαιτήσεις που διατυπώθηκαν στο Κεφάλαιο 3 και ευθυγραμμίζεται με τη συνολική αρχιτεκτονική που παρουσιάστηκε στο Κεφάλαιο 4.1.

Ιδιαίτερη έμφαση δίνεται στη διασφάλιση προβλέψιμης χρονικής συμπεριφοράς (real-time λειτουργία), στον σαφή διαχωρισμό κρίσιμων και μη κρίσιμων λειτουργιών, καθώς και στη δομή του λογισμικού με τρόπο αρθρωτό (modular), ώστε να διευκολύνεται η συντήρηση και η μελλοντική επέκταση του συστήματος.

Στις επόμενες υποενότητες παρουσιάζονται αναλυτικά η δομή των επιμέρους modules, ο χρονισμός και η υλοποίηση του βρόχου ελέγχου, καθώς και οι μηχανισμοί καταγραφής και επικοινωνίας.

Η κύρια πειραματική υλοποίηση βασίστηκε σε bare-metal προσέγγιση με χρήση περιφερειακών και διακοπών του STM32, ενώ παράλληλα υλοποιήθηκε και εναλλακτική δομή με FreeRTOS για τη διερεύνηση καλύτερης οργάνωσης των λειτουργιών σε διακριτές εργασίες.

4.3.1 Δομή λογισμικού (modules)

Ο σχεδιασμός του λογισμικού του προτεινόμενου ενσωματωμένου συστήματος βασίζεται σε αρθρωτή δομή (modular design), ώστε να εξασφαλίζεται σαφής διαχωρισμός λειτουργιών, ευκολία συντήρησης και δυνατότητα μελλοντικών επεκτάσεων. Η επιλογή αυτή ευθυγραμμίζεται με τις προδιαγραφές λογισμικού που τέθηκαν στο Κεφάλαιο 3 και υποστηρίζει τον διαχωρισμό λειτουργιών υψηλής προτεραιότητας (έλεγχος και κρίσιμες μετρήσεις) από λειτουργίες δευτερεύουσας προτεραιότητας (επικοινωνία και καταγραφή).

Σε υψηλό επίπεδο, το λογισμικό οργανώνεται στα ακόλουθα βασικά modules:

- **Module απόκτησης μετρήσεων (Sensor acquisition):** είναι υπεύθυνο για τη συλλογή δεδομένων από τους αισθητήρες, είτε μέσω ADC είτε μέσω ψηφιακών διεπαφών. Περιλαμβάνει τη βασική μετατροπή των ακατέργαστων τιμών σε μετρήσεις χρήσιμες για έλεγχο και εποπτεία (π.χ. κλιμάκωση σε φυσικές μονάδες).



- **Module ελέγχου (Control module):** υλοποιεί τη λογική του βρόχου ελέγχου, υπολογίζοντας την κατάλληλη τιμή ελέγχου με βάση την επιθυμητή τιμή λειτουργίας (setpoint) και τις μετρούμενες παραμέτρους. Το module αυτό αποτελεί τον πυρήνα πραγματικού χρόνου του συστήματος και εκτελείται με σταθερή περιοδικότητα.
- **Module παραγωγής σήματος οδήγησης (PWM output):** αναλαμβάνει την ενημέρωση του duty cycle και τη διαχείριση των ορίων λειτουργίας κατά την παραγωγή του σήματος PWM προς το στάδιο οδήγησης κινητήρα. Η ύπαρξή του ως διακριτό module επιτρέπει τον έλεγχο της διεπαφής με το hardware με σαφή και ασφαλή τρόπο.
- **Module καταγραφής (Logging / events):** οργανώνει και αποθηκεύει τις βασικές μετρήσεις και τα γεγονότα λειτουργίας (events). Το module αυτό λειτουργεί συμπληρωματικά και δεν πρέπει να παρεμβαίνει στη χρονική λειτουργία του βρόχου ελέγχου. Η καταγραφή μπορεί να πραγματοποιείται είτε μέσω προσωρινού buffer στη μνήμη του μικροελεγκτή είτε μέσω αποστολής δεδομένων προς εξωτερικό σύστημα (π.χ. PC ή IoT πλατφόρμα). Σε περίπτωση χρήσης κατάλληλης δικτυακής πλατφόρμας (ThingsBoard), τα δεδομένα δύνανται να αποθηκεύονται σε απομακρυσμένο περιβάλλον (cloud-based storage). Στην παρούσα υλοποίηση, η κύρια καταγραφή και ιστορική απεικόνιση των δεδομένων πραγματοποιείται στο επίπεδο ThingsBoard, ενώ στον STM32 τα δεδομένα διατηρούνται προσωρινά για επεξεργασία και αποστολή τηλεμετρίας.
- **Module επικοινωνίας (Communication module):** είναι υπεύθυνο για τη μετάδοση επιλεγμένων δεδομένων προς εξωτερικό σύστημα εποπτείας (PC / dashboard / IoT). Η λειτουργία του σχεδιάζεται ώστε να είναι μη παρεμβατική, με χρήση buffer και μηχανισμών που αποτρέπουν την καθυστέρηση κρίσιμων λειτουργιών.
- **Module χειρισμού σφαλμάτων (Fault handling):** συγκεντρώνει τους ελέγχους ορίων, όπως υπέρβαση ρεύματος ή ανωμαλία τάσης, και ενεργοποιεί βασικές ενέργειες προστασίας ή alarm, όταν απαιτείται.

Η παραπάνω δομή επιτρέπει αφενός την καθαρή αντιστοίχιση των λειτουργιών λογισμικού με τα υποσυστήματα της αρχιτεκτονικής που παρουσιάστηκε στο Κεφάλαιο 4.1 και αφετέρου τη συστηματική τεκμηρίωση της υλοποίησης στο



Κεφάλαιο 5. Παράλληλα, διευκολύνει τον καθορισμό πειραματικών σεναρίων και την αξιολόγηση των μετρικών απόδοσης στο Κεφάλαιο 6, καθώς οι λειτουργίες καταγραφής και επικοινωνίας μπορούν να ενεργοποιούνται ή να προσαρμόζονται χωρίς να επηρεάζεται ο βασικός έλεγχος.

4.3.2 Χρονισμός και υλοποίηση βρόγχου ελέγχου

Ο βρόγχος ελέγχου του κινητήρα αποτελεί τη χρονικά κρίσιμη λειτουργία του συστήματος και υλοποιείται στον μικροελεγκτή STM32, ο οποίος λειτουργεί ως το deterministic επίπεδο ελέγχου (control layer). Η σχεδίαση του χρονισμού βασίζεται στην ανάγκη προβλέψιμης συμπεριφοράς (real-time execution), ώστε οι υπολογισμοί ελέγχου να εκτελούνται με σταθερή περιοδικότητα και χωρίς παρεμβολές από δευτερεύουσες λειτουργίες.

Η βασική δομή του συστήματος ακολουθεί περιοδική εκτέλεση (periodic execution model), όπου ο βρόγχος ελέγχου ενεργοποιείται με σταθερό χρονικό βήμα T_s . Η επιλογή του χρόνου δειγματοληψίας σχετίζεται με τη δυναμική του κινητήρα και τις απαιτήσεις σταθερότητας του ελεγκτή. Για την πειραματική διάταξη μικρής κλίμακας, χρησιμοποιείται χρονικό διάστημα δειγματοληψίας της τάξης των λίγων milliseconds (1-5), επαρκές για ομαλό έλεγχο ταχύτητας κινητήρα συνεχούς ρεύματος.

Δομή Εκτέλεσης

Η υλοποίηση βασίζεται σε έναν από τους ακόλουθους μηχανισμούς:

- Χρήση περιοδικού interrupt από hardware timer ή
- Υλοποίηση περιοδικού task σε περιβάλλον FreeRTOS

Η κύρια πειραματική υλοποίηση βασίζεται σε bare-metal προσέγγιση με χρήση hardware timers και interrupts. Παράλληλα, υλοποιήθηκε εναλλακτική δομή με FreeRTOS, όπου ο ίδιος βρόγχος ελέγχου οργανώνεται ως περιοδικό task.

Στην προτεινόμενη αρχιτεκτονική, η λειτουργία ελέγχου εκτελείται είτε σε interrupt ρουτίνα (ISR) υψηλής προτεραιότητας είτε σε dedicated real-time task, ώστε να εξασφαλίζεται σταθερή χρονική απόκριση.

Η γενική ακολουθία εκτέλεσης του βρόγχου ελέγχου είναι:

1. Ανάγνωση αισθητήρων (ADC, encoder)
2. Υπολογισμός σφάλματος ως:

$$e(k) = r(k) - y(k)$$



όπου $r(k)$ η επιθυμητή τιμή (setpoint) και $y(k)$ η μετρούμενη ταχύτητα [Παρασκευόπουλος, 2008].

3. Υπολογισμός της εντολής ελέγχου μέσω βασικού PI μηχανισμού, με βάση το σφάλμα μεταξύ setpoint και μετρούμενης ταχύτητας.
4. Ενημέρωση duty cycle PWM
5. Έλεγχος ορίων ασφαλείας

Η εκτέλεση ολοκληρώνεται εντός του χρονικού παραθύρου T_s , πριν την επόμενη ενεργοποίηση.

Ο συνολικός χρόνος εκτέλεσης του βρόχου ελέγχου είναι σημαντικά μικρότερος από το χρονικό διάστημα δειγματοληψίας T_s , ώστε να διασφαλίζεται χρονικό περιθώριο (timing margin) και να αποφεύγεται υπέρβαση προθεσμίας (deadline miss).

Διαχωρισμός Κρίσιμων και Μη Κρίσιμων Λειτουργιών

Για τη διατήρηση της χρονικής προβλεψιμότητας, το λογισμικό διαχωρίζει αυστηρά:

Κρίσιμες λειτουργίες (real-time critical):

- Ανάγνωση encoder
- Υπολογισμός ελέγχου
- Ενημέρωση PWM
- Έλεγχος προστασίας (overcurrent, overtemperature etc.)

Μη κρίσιμες λειτουργίες:

- Καταγραφή δεδομένων (logging)
- Μετάδοση τηλεμετρίας
- Διαχείριση επικοινωνίας UART

Οι μη κρίσιμες λειτουργίες εκτελούνται εκτός του control loop, είτε σε χαμηλότερης προτεραιότητας task είτε σε main loop με μη παρεμβατικό τρόπο (non-blocking execution). Με τον τρόπο αυτό αποτρέπεται η εισαγωγή jitter ή καθυστερήσεων στον βρόχο ελέγχου.

Διαχείριση Χρονικής Συμπεριφοράς (Jitter Control)

Η ποιότητα του ελέγχου και η προβλεψιμότητα του συστήματος επηρεάζονται από τη μεταβλητότητα του χρόνου εκτέλεσης (jitter). Για τον περιορισμό του:

- Αποφεύγονται blocking κλήσεις εντός του control loop
- Η επικοινωνία UART πραγματοποιείται μέσω buffer



- Η επεξεργασία MQTT μεταφέρεται πλήρως στην ESP32

Η απομόνωση της επικοινωνίας από τον STM32 διασφαλίζει ότι η δικτυακή δραστηριότητα δεν επιδρά στη χρονική συμπεριφορά του ελέγχου.

Μηχανισμοί Ασφάλειας

Ο έλεγχος ασφαλείας υλοποιείται εντός του ίδιου χρονικού πλαισίου με τον βρόχο ελέγχου. Σε κάθε κύκλο εκτελείται:

- Έλεγχος υπέρβασης ρεύματος
- Έλεγχος θερμοκρασίας
- Έλεγχος ανωμαλίας τάσης

Σε περίπτωση παραβίασης ορίου, ο STM32 θέτει το PWM σε ασφαλή τιμή και σηματοδοτεί κατάσταση σφάλματος. Η λειτουργία αυτή είναι πλήρως ανεξάρτητη από την ESP32 και την IoT υποδομή.

Συνολική Λογική Χρονισμού

Η υλοποίηση ακολουθεί την αρχή:

Τοπικού ντετερμινιστικού ελέγχου και της ασύγχρονης εξωτερικής επικοινωνίας [Wolf], [Marwedel].

Ο STM32 διατηρεί πλήρη έλεγχο της χρονικής συμπεριφοράς του συστήματος, ενώ η ESP32 λειτουργεί ως ασύγχρονη μονάδα επικοινωνίας. Η αρχιτεκτονική αυτή επιτρέπει πειραματική αξιολόγηση του ελεγκτή, καταγραφή δεδομένων και απομακρυσμένη εποπτεία, χωρίς να διακυβεύεται η σταθερότητα του βρόχου ελέγχου. Ο χρονοπρογραμματισμός (scheduling) αναφέρεται στον μηχανισμό με τον οποίο ο μικροελεγκτής καθορίζει ποια εργασία εκτελείται και σε ποια χρονική στιγμή. Στο προτεινόμενο σύστημα, ο έλεγχος στην κύρια bare-metal υλοποίηση βασίζεται σε περιοδική εκτέλεση και μηχανισμούς διακοπών, ενώ στην εναλλακτική υλοποίηση FreeRTOS οργανώνεται ως περιοδική εργασία υψηλής προτεραιότητας. Ο όρος jitter περιγράφει τη χρονική απόκλιση από την ιδανική περιοδικότητα εκτέλεσης μιας εργασίας. Μικρές διακυμάνσεις στον χρόνο ενεργοποίησης του βρόχου ελέγχου μπορεί να επηρεάσουν τη σταθερότητα του συστήματος, γι' αυτό και επιδιώκεται η ελαχιστοποίησή τους.

4.3.3 Λογική Ελέγχου και Διαχείριση Κατάστασης

Η λογική ελέγχου του συστήματος υλοποιείται στον μικροελεγκτή STM32 και αποτελεί τον πυρήνα του επιπέδου ελέγχου. Ο σχεδιασμός περιλαμβάνει τον



υπολογισμό της εντολής PWM, την επεξεργασία της ανάδρασης από τον encoder, την εφαρμογή περιορισμών λειτουργίας και τη διαχείριση βασικών συνθηκών προστασίας.

Υπολογισμός Εντολής Ελέγχου

Σε κάθε περίοδο δειγματοληψίας T_s , ο STM32 εκτελεί τα ακόλουθα βήματα:

1. Ανάγνωση της μετρούμενης ταχύτητας από τον encoder.
2. Υπολογισμός της διαφοράς μεταξύ επιθυμητής και πραγματικής τιμής:

$$e(k) = r(k) - y(k)$$

όπου:

- $r(k)$ είναι το setpoint,
 - $y(k)$ είναι η μετρούμενη ταχύτητα.
3. Υπολογισμός της εντολής ελέγχου βάσει της τιμής του σφάλματος.
 4. Μετατροπή της εντολής σε duty cycle PWM.
 5. Εφαρμογή περιορισμών (saturation limits).

Οι περιορισμοί εξασφαλίζουν ότι το duty cycle παραμένει εντός αποδεκτών ορίων και δεν οδηγεί σε υπερβολική κατανάλωση ρεύματος ή θερμική καταπόνηση.

Ο υπολογισμός της εντολής παραμένει ελαφρύς υπολογιστικά, ώστε να ολοκληρώνεται εντός του χρονικού παραθύρου εκτέλεσης και να διατηρείται χρονικό περιθώριο (timing margin).

Διαχείριση Καταστάσεων (State Machine)

Η συνολική λειτουργία του συστήματος οργανώνεται σε διακριτές καταστάσεις:

- **Idle** – Ο κινητήρας είναι απενεργοποιημένος.
- **Starting** – Ελεγχόμενη εκκίνηση και σταδιακή εφαρμογή PWM.
- **Running** – Κανονική λειτουργία με ενεργό βρόχο ελέγχου.
- **Fault** – Ενεργοποίηση μηχανισμού προστασίας.
- **Safe Stop** – Ασφαλής παύση λειτουργίας.

Η χρήση λογικής καταστάσεων επιτρέπει:

- Ελεγχόμενες μεταβάσεις μεταξύ λειτουργιών.
- Αποφυγή αιφνίδιων αλλαγών PWM.
- Διαχωρισμό κανονικής και ανώμαλης λειτουργίας.
- Σαφή χειρισμό εξωτερικών εντολών.



Η μετάβαση σε κατάσταση Fault έχει προτεραιότητα έναντι οποιασδήποτε άλλης λειτουργίας.

Διαχείριση Εξωτερικών Εντολών

Οι εντολές που λαμβάνονται από την ESP32 (π.χ. αλλαγή setpoint ή start/stop μέσω RPC) δεν εφαρμόζονται άμεσα. Πριν την εφαρμογή τους:

- Ελέγχεται η εγκυρότητα των παραμέτρων.
- Επαληθεύεται η συμβατότητα με την τρέχουσα κατάσταση.
- Διασφαλίζεται ότι δεν παραβιάζονται συνθήκες ασφαλείας.

Με αυτόν τον τρόπο, ο STM32 διατηρεί πλήρη έλεγχο της λειτουργίας και δεν επιτρέπει στο επίπεδο cloud να υπερκεράσει τους τοπικούς μηχανισμούς προστασίας.

Ιεραρχία Προτεραιοτήτων

Η λογική λειτουργίας ακολουθεί την εξής ιεραρχία:

1. Μηχανισμοί προστασίας (ρεύμα, θερμοκρασία, τάση)
2. Κατάσταση συστήματος (state machine)
3. Τοπικές ρυθμίσεις
4. Εξωτερικές εντολές IoT

Η ιεράρχηση αυτή διασφαλίζει ότι η ασφάλεια του συστήματος προηγείται οποιασδήποτε απομακρυσμένης εντολής.

Τοπική αυτονομία και απομακρυσμένη εποπτεία

Η λογική ελέγχου παραμένει πλήρως εντοπισμένη στο edge επίπεδο (STM32), ενώ το cloud λειτουργεί αποκλειστικά ως εποπτικό και διαχειριστικό επίπεδο. Η προσέγγιση αυτή ενσωματώνει τη βασική αρχή των ubiquitous συστημάτων: κατανεμημένη λειτουργία με τοπική αυτονομία και απομακρυσμένη εποπτεία.

Η σχεδίαση διασφαλίζει ότι ακόμα και σε περίπτωση απώλειας επικοινωνίας, ο βρόγχος ελέγχου συνεχίζει να λειτουργεί κανονικά και με ασφάλεια.

4.3.4 Διαχείριση Δεδομένων και Τηλεμετρίας

Η διαχείριση δεδομένων στο προτεινόμενο σύστημα αφορά τη συλλογή, οργάνωση και αποστολή επιλεγμένων μετρήσεων προς το επίπεδο εποπτείας, χωρίς να επηρεάζεται η χρονικά κρίσιμη λειτουργία του βρόχου ελέγχου. Η αρχιτεκτονική βασίζεται σε σαφή διαχωρισμό μεταξύ του επιπέδου ελέγχου (STM32) και του επιπέδου επικοινωνίας (ESP32), με στόχο τη διατήρηση προβλέψιμης χρονικής συμπεριφοράς στο control layer.



Συλλογή και Οργάνωση Μετρήσεων

Ο STM32 συλλέγει σε κάθε κύκλο λειτουργίας:

- Ταχύτητα περιστροφής (RPM) από encoder
- Ρεύμα κινητήρα (μέσω αισθητήρα Hall)
- Θερμοκρασία (DS18B20)
- Τάση τροφοδοσίας
- Καταστάσεις προστασίας ή ενδείξεις υπέρβασης ορίων
- Τρέχουσα κατάσταση λειτουργίας του συστήματος (state machine)

Οι μετρήσεις οργανώνονται σε εσωτερική δομή δεδομένων (data structure), η οποία περιλαμβάνει:

- Τρέχουσες τιμές
- Οριακές καταστάσεις
- Κατάσταση λειτουργίας

Η οργάνωση αυτή επιτρέπει σαφή αντιστοίχιση των φυσικών μεγεθών με τα πεδία τηλεμετρίας που θα μεταδοθούν.

Ρυθμός Αποστολής Τηλεμετρίας

Η αποστολή τηλεμετρίας δεν πραγματοποιείται σε κάθε κύκλο ελέγχου. Αντίθετα, εφαρμόζεται διαχωρισμός ρυθμών:

- Ο βρόχος ελέγχου εκτελείται με σημαντικά υψηλότερο ρυθμό από την αποστολή τηλεμετρίας.
- Η τηλεμετρία αποστέλλεται περιοδικά με χαμηλότερο ρυθμό, κατάλληλο για απομακρυσμένη εποπτεία.

Ο χαμηλότερος ρυθμός τηλεμετρίας:

- Μειώνει το επικοινωνιακό φορτίο
- Αποτρέπει συμφόρηση UART
- Περιορίζει τον κίνδυνο jitter

Η προσέγγιση αυτή διασφαλίζει ότι ο έλεγχος κινητήρα παραμένει ανεπηρέαστος από τη διαδικασία αποστολής δεδομένων.

Μορφοποίηση Πλαισίου Επικοινωνίας (Telemetry Frame)

Η μετάδοση δεδομένων STM32–ESP32 πραγματοποιείται μέσω UART, με χρήση προκαθορισμένου πλαισίου (frame). Το πλαίσιο περιλαμβάνει:

- Αναγνωριστικό αρχής (start byte)



- Πεδία μετρήσεων (RPM, ρεύμα, θερμοκρασία, τάση)
- Κατάσταση συστήματος
- Βασικό έλεγχο ορθότητας του πλαισίου δεδομένων (checksum ή απλό validation)
- Αναγνωριστικό τέλους

Η απλή και σαφής μορφοποίηση μειώνει την υπολογιστική επιβάρυνση του STM32 και επιτρέπει αξιόπιστο parsing από την ESP32.

Ρόλος της ESP32 ως IoT Gateway

Η ESP32 λειτουργεί αποκλειστικά ως μονάδα επικοινωνίας και αναλαμβάνει:

- Ανάγνωση των UART frames
- Ανάλυση και επεξεργασία των UART frames
- Buffering δεδομένων
- Σύνδεση σε δίκτυο WiFi
- Δημοσίευση δεδομένων μέσω MQTT

Η χρήση MQTT επιτρέπει αποδοτική μεταφορά τηλεμετρίας προς την πλατφόρμα ThingsBoard, όπου τα δεδομένα αποθηκεύονται και οπτικοποιούνται.

Η δικτυακή λειτουργία παραμένει πλήρως απομονωμένη από το control layer.

Καταγραφή Δεδομένων (Logging)

Το σύστημα υποστηρίζει δύο επίπεδα καταγραφής:

1. Τοπική οργάνωση δεδομένων στον STM32 για άμεση ανάλυση.
2. Καταγραφή στο cloud μέσω ThingsBoard.

Η πλατφόρμα επιτρέπει:

- Ιστορικό αποθήκευσης μετρήσεων
- Οπτικοποίηση σε dashboards
- Ορισμό κανόνων και alarm conditions
- Απομακρυσμένη παρακολούθηση

Η αποθήκευση πραγματοποιείται στο επίπεδο IoT, αποφεύγοντας ανάγκη για εκτεταμένη μνήμη στο STM32.

Μη Παρεμβατική Επικοινωνία

Η επικοινωνία σχεδιάζεται ως non-blocking:

- Ο STM32 δεν αναμένει επιβεβαίωση από την ESP32.
- Η αποστολή γίνεται μέσω buffer.



- Οποιαδήποτε καθυστέρηση στο WiFi ή MQTT δεν επηρεάζει τον βρόχο ελέγχου.

Η αρχή που εφαρμόζεται είναι:

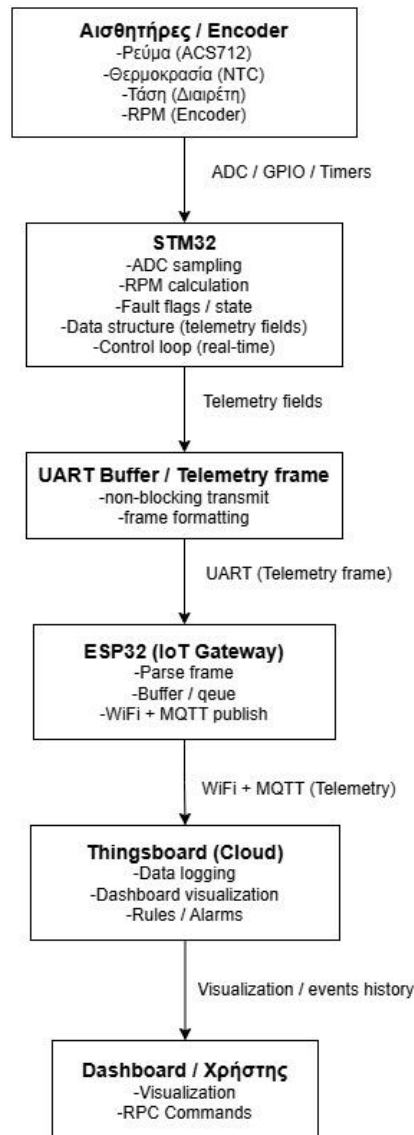
Deterministic control locally – asynchronous communication externally.

Αρχιτεκτονικός Διαχωρισμός Επιπέδων

Η υλοποίηση επιβεβαιώνει τον σαφή διαχωρισμό:

- STM32 → real-time, safety-critical λειτουργία
- ESP32 → δικτυακή και IoT λειτουργία
- ThingsBoard → εποπτεία και ιστορικό

Σε περίπτωση απώλειας επικοινωνίας, ο έλεγχος κινητήρα συνεχίζει κανονικά, διατηρώντας πλήρη αυτονομία στο edge επίπεδο.



Εικόνα 4.4. Διάγραμμα ροής δεδομένων (data flow)

Σημειώνεται ότι το παραπάνω διάγραμμα απεικονίζει τη ροή δεδομένων όσον αφορά την τηλεμετρία, προς το υπολογιστικό νέφος (upstream telemetry flow). Ωστόσο, το σύστημα υποστηρίζει επίσης αμφίδρομη επικοινωνία μέσω εντολών απομακρυσμένου ελέγχου (RPC – Remote Procedure Calls), οι οποίες εκκινούν από το περιβάλλον του χρήστη στο cloud και ακολουθούν την αντίστροφη διαδρομή (Dashboard → Thingsboard → ESP32 → STM32). Η αντίστροφη αυτή ροή εντολών είναι λογικά διαχωρισμένη από τον βρόχο ελέγχου πραγματικού χρόνου, ώστε να διασφαλίζεται η ντετερμινιστική λειτουργία του συστήματος και να αποφεύγεται η επίδραση δικτυακών καθυστερήσεων στη βασική λειτουργία ελέγχου του κινητήρα.



4.3.5 Διαχωρισμός Πραγματικού Χρόνου και Δικτυακής Επικοινωνίας

Η σχεδίαση του συστήματος βασίζεται σε σαφή αρχιτεκτονικό διαχωρισμό μεταξύ του επιπέδου ελέγχου πραγματικού χρόνου (real-time control layer) και του επιπέδου δικτυακής επικοινωνίας (communication layer). Ο διαχωρισμός αυτός αποτελεί θεμελιώδη επιλογή σχεδίασης με στόχο τη διασφάλιση της ντετερμινιστικής λειτουργίας του ελεγκτή κινητήρα.

Ο μικροελεγκτής STM32 αναλαμβάνει αποκλειστικά τις λειτουργίες που απαιτούν αυστηρούς χρονικούς περιορισμούς, όπως:

- Δειγματοληψία αναλογικών σημάτων (ADC sampling)
- Υπολογισμό στροφών (RPM calculation)
- Ανίχνευση σφαλμάτων (fault detection)
- Εκτέλεση του βρόχου ελέγχου
- Διαχείριση κρίσιμων διακοπών (interrupt handling)

Οι παραπάνω λειτουργίες εκτελούνται με χρήση χρονιστών (timers) και μηχανισμών διακοπών, εξασφαλίζοντας προβλέψιμη χρονική συμπεριφορά (deterministic execution).

Αντίθετα, η μονάδα ESP32 λειτουργεί ως πύλη IoT (IoT gateway), αναλαμβάνοντας:

- Ανάλυση τηλεμετρικών πλαισίων (telemetry frame parsing)
- Προσωρινή αποθήκευση και διαχείριση δεδομένων πριν την αποστολή MQTT (buffer/queue management)
- Σύνδεση WiFi
- Δημοσίευση δεδομένων μέσω MQTT
- Λήψη και προώθηση εντολών RPC

Η επικοινωνία μεταξύ STM32 και ESP32 υλοποιείται μέσω σειριακής διεπαφής UART με μη-αποκλειστική μετάδοση (non-blocking transmit), ώστε να μην παρεμποδίζεται η εκτέλεση του βρόχου ελέγχου.

Η ασύγχρονη φύση των δικτυακών πρωτοκόλλων (WiFi, MQTT) συνεπάγεται μεταβλητές καθυστερήσεις (variable latency), οι οποίες δεν είναι αποδεκτές σε συστήματα ελέγχου πραγματικού χρόνου. Για τον λόγο αυτό, οι λειτουργίες επικοινωνίας είναι πλήρως απομονωμένες από τον εσωτερικό βρόχο ελέγχου του STM32.

Οι εντολές RPC που προέρχονται από το cloud δεν επεμβαίνουν άμεσα στον βρόχο ελέγχου, αλλά εισάγονται ως μη-κρίσιμες παράμετροι ή αιτήματα, τα οποία



αξιολογούνται από το STM32 υπό ελεγχόμενες συνθήκες. Με τον τρόπο αυτό διασφαλίζεται ότι:

- Η βασική λειτουργία του κινητήρα παραμένει ανεξάρτητη από τη δικτυακή κατάσταση.
- Πιθανές αποσυνδέσεις ή καθυστερήσεις δικτύου δεν επηρεάζουν τη σταθερότητα του συστήματος.
- Η ασφάλεια λειτουργίας (operational safety) παραμένει τοπικά εγγυημένη.

Η αρχιτεκτονική αυτή προσεγγίζει το μοντέλο “edge control – cloud monitoring”, όπου ο έλεγχος πραγματοποιείται τοπικά (edge device), ενώ το υπολογιστικό νέφος χρησιμοποιείται για εποπτεία, καταγραφή και διαχείριση υψηλότερου επιπέδου.

Ο διαχωρισμός αυτός μειώνει τη σύζευξη (coupling) μεταξύ κρίσιμων και μη κρίσιμων λειτουργιών, αυξάνει την αξιοπιστία του συστήματος και επιτρέπει μελλοντική επεκτασιμότητα χωρίς μεταβολή του βασικού πυρήνα ελέγχου.

4.4 Υλοποίηση Λογισμικού και Δομή Firmware

4.4.1 Περιβάλλον Εκτέλεσης και Γενική Δομή Λογισμικού

Η κύρια πειραματική υλοποίηση του λογισμικού του μικροελεγκτή STM32 βασίζεται σε bare-metal προσέγγιση, με χρήση περιφερειακών υλικού (hardware peripherals), χρονιστών (timers) και μηχανισμών διακοπών (interrupts) για την υλοποίηση του βρόχου ελέγχου και των λειτουργιών τηλεμετρίας. Η επιλογή αυτή επιτρέπει απλή και προβλέψιμη εκτέλεση των χρονικά κρίσιμων λειτουργιών του συστήματος, χωρίς την επιβάρυνση πλήρους συστήματος πραγματικού χρόνου.

Παράλληλα, αναπτύχθηκε και εναλλακτική αρχιτεκτονική βασισμένη στο FreeRTOS, με στόχο τη διερεύνηση διαφορετικής οργάνωσης των λειτουργιών του firmware σε διακριτές εργασίες (tasks) και τη μελέτη μηχανισμών χρονοπρογραμματισμού πραγματικού χρόνου. Η υλοποίηση αυτή χρησιμοποιήθηκε συμπληρωματικά και δεν αποτέλεσε την κύρια πειραματική πλατφόρμα αξιολόγησης του συστήματος.

Η γενική δομή του firmware ακολουθεί τη λογική:

Αρχικοποίηση υλικού (hardware initialization):

- Ρύθμιση ρολογιού συστήματος (system clock)
- Διαμόρφωση περιφερειακών (ADC, timers, PWM, UART, GPIO)
- Αρχικοποίηση μεταβλητών τηλεμετρίας και επικοινωνίας



- Ενεργοποίηση watchdog και βασικών μηχανισμών προστασίας

Στην κύρια bare-metal υλοποίηση, ο βρόχος ελέγχου εκτελείται περιοδικά μέσω hardware timer και μηχανισμών διακοπών, ενώ οι λειτουργίες τηλεμετρίας και επικοινωνίας UART υλοποιούνται με μη παρεμβατικό τρόπο, ώστε να μην επηρεάζεται η χρονική συμπεριφορά του ελέγχου.

Στην εναλλακτική υλοποίηση FreeRTOS, η δομή του firmware οργανώνεται σε διακριτές εργασίες διαφορετικής προτεραιότητας:

- Δημιουργία tasks
- Δημιουργία queues και buffers επικοινωνίας
- Καθορισμός προτεραιοτήτων εργασιών
- Εκκίνηση scheduler και μετάβαση σε πολυδιεργασιακή λειτουργία

Με την εκκίνηση του scheduler, οι χρονικά κρίσιμες εργασίες, όπως ο βρόχος ελέγχου και οι βασικοί μηχανισμοί προστασίας, εκτελούνται με υψηλότερη προτεραιότητα σε σχέση με τις λειτουργίες τηλεμετρίας και επικοινωνίας.

Η χρήση RTOS επιτρέπει:

- Προβλέψιμη εκτέλεση περιοδικών εργασιών
- Διαχωρισμό κρίσιμων και μη κρίσιμων λειτουργιών
- Ευκολότερη επεκτασιμότητα του συστήματος
- Βελτιωμένη δομή και αναγνωσιμότητα του κώδικα

Στην υλοποίηση FreeRTOS, οι περιοδικές λειτουργίες μπορούν να οργανωθούν με χρήση μηχανισμών χρονισμού του RTOS (όπως `vTaskDelayUntil`) σε συνδυασμό με hardware interrupts, ώστε να διατηρείται σταθερή χρονική συμπεριφορά και περιορισμός jitter.

Η αρχιτεκτονική αυτή συνάδει με τη φιλοσοφία διαχωρισμού επιπέδων που παρουσιάζεται στο παρόν Κεφάλαιο, όπου ο STM32 λειτουργεί ως τοπικός κόμβος ελέγχου πραγματικού χρόνου, ανεξάρτητος από την ασύγχρονη δικτυακή δραστηριότητα του επιπέδου IoT.

Στην αρχιτεκτονική αυτή, οι χρονικά κρίσιμες λειτουργίες, όπως ο βρόχος ελέγχου και οι βασικοί μηχανισμοί προστασίας, εκτελούνται μέσω εργασιών υψηλής προτεραιότητας και μηχανισμών διακοπών (ISR), ενώ οι λειτουργίες τηλεμετρίας και επικοινωνίας UART οργανώνονται σε χαμηλότερης προτεραιότητας tasks και buffers.

Ο διαχωρισμός αυτός μειώνει την αλληλεπίδραση μεταξύ του βρόχου ελέγχου και της



δικτυακής δραστηριότητας, διατηρώντας προβλέψιμη χρονική συμπεριφορά στο επίπεδο ελέγχου.

4.4.2 Δομή Εργασιών και Ιεράρχηση Προτεραιοτήτων στο Περιβάλλον RTOS

Παρότι η κύρια πειραματική υλοποίηση του συστήματος βασίστηκε σε bare-metal αρχιτεκτονική, αναπτύχθηκε και εναλλακτική δομή firmware με χρήση του FreeRTOS, με στόχο τη μελέτη οργάνωσης των λειτουργιών του συστήματος σε διακριτές εργασίες (tasks) διαφορετικής προτεραιότητας. Η προσέγγιση αυτή επιτρέπει καλύτερο διαχωρισμό μεταξύ λειτουργιών ελέγχου, τηλεμετρίας και επικοινωνίας, διατηρώντας παράλληλα προβλέψιμη χρονική συμπεριφορά.

Η χαρτογράφηση των εργασιών σχεδιάστηκε με βάση τις λειτουργικές απαιτήσεις στο Κεφάλαιο 3 – Ανάλυση απαιτήσεων και Προδιαγραφές Συστήματος και τη λογική αρχιτεκτονικής στο Κεφάλαιο 4 – Σχεδιασμός Συστήματος.

Δομή Εργασιών

1. ControlTask (Υψηλότερη Προτεραιότητα)

Η ControlTask αποτελεί την κρίσιμη εργασία πραγματικού χρόνου του συστήματος. Εκτελείται περιοδικά με σταθερό χρονικό βήμα T_s , χρησιμοποιώντας τον μηχανισμό `vTaskDelayUntil()` ώστε να εξασφαλίζεται σταθερή περίοδος εκτέλεσης.

Κύριες λειτουργίες:

- Ανάγνωση μετρήσεων (encoder, ADC)
- Υπολογισμός σφάλματος
- Εκτέλεση νόμου ελέγχου
- Ενημέρωση PWM
- Έλεγχος βασικών ορίων ασφαλείας

Η εργασία αυτή δεν περιλαμβάνει blocking κλήσεις και δεν εκτελεί επικοινωνιακές λειτουργίες.

2. SafetyTask (Ιση ή Ελαφρώς Χαμηλότερη Προτεραιότητα)

Η SafetyTask συγκεντρώνει και αξιολογεί τις ενδείξεις σφαλμάτων (fault flags) που παράγονται είτε από την ControlTask είτε από interrupts.

Λειτουργίες:

- Έλεγχος υπερέντασης
- Έλεγχος υπερθέρμανσης



- Έλεγχος ανωμαλίας τάσης
- Μετάβαση σε ασφαλή κατάσταση (PWM disable)

Η ύπαρξη ξεχωριστής εργασίας ασφάλειας επιτρέπει τον σαφή διαχωρισμό ελέγχου και προστασίας.

3. TelemetryTask (Μεσαία Προτεραιότητα)

Η TelemetryTask εκτελείται με χαμηλότερη συχνότητα (π.χ. 10–20 Hz) και είναι υπεύθυνη για:

- Συγκέντρωση στιγμιότυπου μετρήσεων
- Δημιουργία τηλεμετρικού πλαισίου
- Υπολογισμό CRC
- Τοποθέτηση δεδομένων σε ουρά μετάδοσης UART

Η αποστολή δεδομένων γίνεται μέσω queue, ώστε να αποφεύγεται blocking.

4. HousekeepingTask (Χαμηλότερη Προτεραιότητα)

Η εργασία αυτή αναλαμβάνει μη κρίσιμες λειτουργίες όπως:

- Ενημέρωση κατάστασης συστήματος
- Διαχείριση watchdog refresh
- Διαγνωστικά μηνύματα

Η ανανέωση του watchdog πραγματοποιείται μόνο εφόσον το σύστημα βρίσκεται σε κανονική λειτουργία.

Διαχείριση Συγχρονισμού

Για την ανταλλαγή δεδομένων χρησιμοποιούνται:

- Queues για επικοινωνία μεταξύ tasks
- Event flags για ένδειξη καταστάσεων
- Interrupt service routines (ISR) για άμεση καταγραφή εξωτερικών γεγονότων

Δεν χρησιμοποιούνται blocking καθυστερήσεις ή busy-wait loops.

Καθορισμός Προτεραιοτήτων

Η ιεράρχηση των εργασιών ακολουθεί την αρχή:

ControlTask > SafetyTask ≥ TelemetryTask > HousekeepingTask

Η δομή αυτή εξασφαλίζει ότι:

- Ο βρόχος ελέγχου δεν καθυστερεί από επικοινωνιακές λειτουργίες.
- Οι μη κρίσιμες εργασίες εκτελούνται μόνο όταν υπάρχουν διαθέσιμοι πόροι.
- Η χρονική συμπεριφορά του συστήματος παραμένει προβλέψιμη.



Η παραπάνω κατηγοριοποίηση επιτρέπει σαφή απομόνωση λειτουργιών, μειώνει τη σύζευξη μεταξύ επιπέδων και καθιστά το σύστημα επεκτάσιμο χωρίς μεταβολή του βασικού πυρήνα ελέγχου.

4.4.3 Διαχείριση Περιφερειακών και Διακοπών (Interrupts)

Η διαχείριση των περιφερειακών μονάδων (peripherals) του μικροελεγκτή STM32 υλοποιείται με τρόπο που εξασφαλίζει σαφή διαχωρισμό μεταξύ του επιπέδου υλικού και της λογικής ελέγχου, διατηρώντας παράλληλα προβλέψιμη χρονική συμπεριφορά. Η αρχιτεκτονική βασίζεται κυρίως σε hardware timers και μηχανισμούς διακοπών (interrupt-driven design), τόσο στην κύρια bare-metal υλοποίηση όσο και στην εναλλακτική αρχιτεκτονική FreeRTOS.

Ρύθμιση Αναλογικού Ψηφιακού Μετατροπέα (ADC)

Ο ADC χρησιμοποιείται για τη δειγματοληψία φυσικών μεγεθών (π.χ. ρεύμα, τάση). Η λειτουργία του ενεργοποιείται περιοδικά μέσω hardware timer trigger, ώστε:

- Να εξασφαλίζεται σταθερή συχνότητα δειγματοληψίας.
- Να αποφεύγεται εξάρτηση από λογισμικές καθυστερήσεις.
- Να μειώνεται το jitter στη διαδικασία μέτρησης.

Η ολοκλήρωση της μετατροπής σηματοδοτείται μέσω interrupt, το οποίο ενημερώνει κατάλληλη μεταβλητή ή queue χωρίς να εκτελεί βαριά επεξεργασία εντός της ISR.

Χρονιστές (Timers) και Παραγωγή PWM

Οι hardware timers αξιοποιούνται για:

- Δημιουργία σήματος PWM για τον έλεγχο του κινητήρα.
- Παραγωγή χρονικού σήματος βάσης (time base) για περιοδικές λειτουργίες.

Η ενημέρωση του duty cycle πραγματοποιείται από την ControlTask, ενώ η πραγματική παραγωγή του σήματος γίνεται πλήρως σε επίπεδο υλικού. Με τον τρόπο αυτό:

- Η ακρίβεια του PWM δεν εξαρτάται από την εκτέλεση του RTOS.
- Ο έλεγχος παραμένει σταθερός ακόμη και υπό φόρτο CPU.

Σειριακή Επικοινωνία (UART) με Χρήση Διακοπών

Η επικοινωνία μεταξύ STM32 και ESP32 υλοποιείται μέσω UART σε interrupt-driven λειτουργία.

Λήψη δεδομένων (Rx)



- Κάθε εισερχόμενο byte προκαλεί interrupt.
- Τα δεδομένα αποθηκεύονται σε ring buffer ή queue.
- Η αποκωδικοποίηση πλαισίων γίνεται σε επίπεδο task και όχι μέσα στην ISR.

Αποστολή δεδομένων (Tx)

- Τα προς μετάδοση δεδομένα τοποθετούνται σε queue.
- Η ISR αναλαμβάνει την αποστολή byte προς byte.
- Δεν χρησιμοποιείται blocking transmit.

Η επιλογή interrupt-based UART επιτρέπει:

- Αποφυγή busy-wait loops.
- Μείωση καθυστερήσεων.
- Διατήρηση της προτεραιότητας της ControlTask.

Σχεδιαστική Φιλοσοφία ISR

Οι Interrupt Service Routines (ISR) έχουν ελάχιστη πολυπλοκότητα και εκτελούν μόνο:

- Καταγραφή γεγονότος.
- Ενημέρωση flag ή buffer.
- Άμεση επιστροφή.

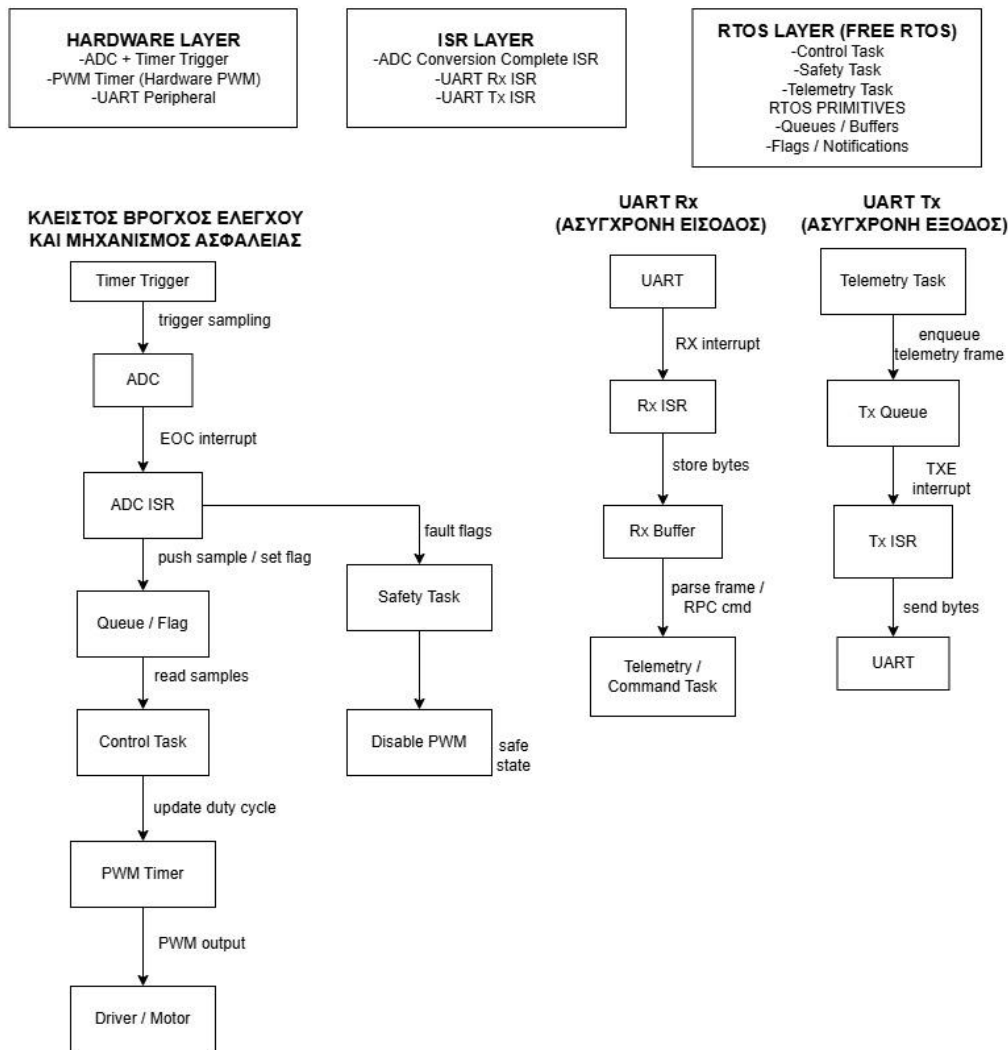
Δεν εκτελούνται υπολογισμοί ελέγχου ή βαριά επεξεργασία εντός ISR, ώστε να περιορίζεται η διάρκεια απενεργοποίησης των interrupts και να διατηρείται χαμηλό latency.

Διαχωρισμός Επιπέδου Υλικού και Λογικής

Η πρόσβαση στα περιφερειακά υλοποιείται μέσω Hardware Abstraction Layer (HAL), ώστε:

- Η λογική ελέγχου να μην εξαρτάται άμεσα από registers.
- Να είναι δυνατή μελλοντική μεταφορά σε διαφορετικό STM32.
- Να βελτιώνεται η αναγνωσιμότητα και συντήρηση του κώδικα.

Η παραπάνω προσέγγιση επιτρέπει στον STM32 να λειτουργεί ως τοπικός ελεγκτής πραγματικού χρόνου, όπου οι χρονικά κρίσιμες λειτουργίες βασίζονται κυρίως σε μηχανισμούς υλικού και όχι σε λογισμικές καθυστερήσεις, διασφαλίζοντας σταθερή συμπεριφορά υπό διαφορετικά φορτία εκτέλεσης.



Εικόνα 4.5. Μπλοκ απεικόνιση της αλληλεπίδρασης περιφερειακών μονάδων, ρουτινών διακοπών και εργασιών FreeRTOS στον μικροελεγκτή STM32.

Όπως φαίνεται στην Εικόνα 4.5, στην αρχιτεκτονική αυτή, οι ρουτίνες διακοπών (ISR) εκτελούν ελάχιστη επεξεργασία και περιορίζονται κυρίως στην ενημέρωση buffers, flags ή μηχανισμών συγχρονισμού. Η κύρια επεξεργασία πραγματοποιείται σε επίπεδο εργασιών (tasks), διατηρώντας προβλέψιμη χρονική συμπεριφορά. Οι βασικοί μηχανισμοί προστασίας λειτουργούν ανεξάρτητα από τη δικτυακή δραστηριότητα και μπορούν να οδηγήσουν το σύστημα σε ασφαλή κατάσταση λειτουργίας όταν απαιτείται.

4.4.4 Δομή Πλαισίων Επικοινωνίας και Έλεγχος Ακεραιότητας (CRC)

Η επικοινωνία μεταξύ της μονάδας ελέγχου (STM32) και της μονάδας IoT (ESP32) υλοποιείται μέσω σειριακής διεπαφής UART, με χρήση δομημένων πλαισίων (frames)



δεδομένων. Η επιλογή συγκεκριμένης δομής πλαισίου επιτρέπει την αξιόπιστη μετάδοση δεδομένων τηλεμετρίας και εντολών ελέγχου, καθώς και βασικό έλεγχο ορθότητας των δεδομένων.

Δομή Τηλεμετρικού Πλαισίου

Κάθε τηλεμετρικό πλαίσιο περιλαμβάνει:

- Start byte (π.χ. 0xAA) για συγχρονισμό
- Identifier πεδίου ή τύπου πλαισίου
- Payload δεδομένων, όπως:
 - Ταχύτητα (RPM)
 - Ρεύμα
 - Τάση
 - Θερμοκρασία
 - Κατάσταση σφάλματος
- CRC (Cyclic Redundancy Check)
- End byte (π.χ. 0x55)

Η ύπαρξη start και end delimiters επιτρέπει την επανασυγχρονισμό σε περίπτωση απώλειας byte.

Βασικός Έλεγχος Ακεραιότητας

Για τον περιορισμό σφαλμάτων μετάδοσης μπορεί να χρησιμοποιηθεί βασικός μηχανισμός ελέγχου ορθότητας δεδομένων, όπως checksum ή CRC (Cyclic Redundancy Check). Ο έλεγχος αυτός επιτρέπει:

- Ανίχνευση αλλοίωσης δεδομένων
- Περιορισμό επιρροής θορύβου στη σειριακή επικοινωνία
- Αποφυγή εσφαλμένης επεξεργασίας εντολών ή τηλεμετρίας

Στην περίπτωση λανθασμένης δομής πλαισίου ή ασυμφωνίας ελέγχου ορθότητας:

- Το πλαίσιο απορρίπτεται
- Δεν ενημερώνονται μεταβλητές ελέγχου
- Η λειτουργία του βρόχου ελέγχου συνεχίζεται ανεξάρτητα

Η επικοινωνία θεωρείται μη κρίσιμη για τη βασική λειτουργία του κινητήρα, επομένως πιθανές αστοχίες μετάδοσης δεν επηρεάζουν τη σταθερότητα του βρόχου ελέγχου.



Δομή Πλαισίου Εντολών (RPC Frame)

Οι εντολές απομακρυσμένου ελέγχου (RPC commands) ακολουθούν αντίστοιχη δομή:

- Start byte
- Τύπος εντολής
- Παράμετρος (π.χ. νέο setpoint)
- CRC
- End byte

Η αποκωδικοποίηση γίνεται σε επίπεδο task στον STM32, μετά από έλεγχο εγκυρότητας CRC. Οι εντολές δεν εφαρμόζονται άμεσα εντός ISR, αλλά περνούν από λογική επαλήθευσης ώστε να διασφαλίζεται ότι:

- Δεν παραβιάζονται όρια ασφαλείας.
- Δεν τροποποιείται ο βρόχος ελέγχου σε ακατάλληλη χρονική στιγμή.

Διαχείριση Σφαλμάτων Επικοινωνίας

Σε περίπτωση ανίχνευσης λανθασμένου CRC ή ασύμβατης δομής πλαισίου:

- Το πλαίσιο απορρίπτεται.
- Δεν ενημερώνεται καμία μεταβλητή ελέγχου.
- Καταγράφεται σχετικό event (προαιρετικά για διαγνωστικούς σκοπούς).

Η επικοινωνία θεωρείται μη κρίσιμη για τη βασική λειτουργία ελέγχου, επομένως τυχόν αποτυχία μετάδοσης δεν επηρεάζει τη σταθερότητα του κινητήρα.

Η υλοποίηση δομημένων πλαισίων με έλεγχο ακεραιότητας επιτρέπει τον σαφή διαχωρισμό μεταξύ επιπέδου ελέγχου και επιπέδου επικοινωνίας, ενισχύοντας την αξιοπιστία και την ανθεκτικότητα του συστήματος.

4.4.5 Μηχανισμοί Προστασίας και Διασφάλισης Λειτουργίας

Η διασφάλιση αξιοπιστίας και ασφαλούς λειτουργίας αποτελεί βασική απαίτηση του συστήματος, καθώς ο ελεγχόμενος κινητήρας αποτελεί ηλεκτρομηχανικό φορτίο που μπορεί να οδηγήσει σε ανεπιθύμητη λειτουργία σε περίπτωση αστοχίας ελέγχου ή επικοινωνίας. Για τον λόγο αυτό, το σύστημα περιλαμβάνει βασικούς μηχανισμούς προστασίας και εποπτείας τόσο σε επίπεδο firmware όσο και σε επίπεδο συνολικής αρχιτεκτονικής.

Χρήση Watchdog



Στον STM32 ενεργοποιείται ο ανεξάρτητος watchdog (Independent Watchdog – IWDG), ο οποίος επιβλέπει τη συνεχή εκτέλεση του firmware.

Η ανανέωση (refresh) του watchdog πραγματοποιείται μόνο εφόσον:

- Ο βρόχος ελέγχου έχει εκτελεστεί επιτυχώς.
- Δεν έχει ανιχνευθεί κατάσταση σφάλματος.
- Το σύστημα βρίσκεται σε κανονική λειτουργία.

Με τον τρόπο αυτό αποτρέπεται η επανεκκίνηση του watchdog σε περίπτωση που το σύστημα έχει εισέλθει σε ανεπιθύμητη ή ασταθή κατάσταση.

Σε περίπτωση μη ανανέωσης εντός προκαθορισμένου χρονικού διαστήματος, ο watchdog προκαλεί αυτόματη επανεκκίνηση (system reset), επαναφέροντας το σύστημα σε ασφαλή αρχική κατάσταση.

Λογική Ανίχνευσης Σφαλμάτων (Fault Detection)

Ο STM32 παρακολουθεί κρίσιμες παραμέτρους λειτουργίας, όπως:

- Υπέρβαση ρεύματος
- Υπερθέρμανση
- Ανωμαλία τάσης
- Μη έγκυρη εντολή RPC

Σε περίπτωση ανίχνευσης παραβίασης ορίων:

1. Το PWM μπορεί να μηδενιστεί ή να απενεργοποιηθεί.
2. Ενεργοποιείται ένδειξη προστασίας ή alarm.
3. Το σύστημα μεταβαίνει σε κατάσταση ασφαλούς λειτουργίας (safe state).

Η μετάβαση αυτή πραγματοποιείται τοπικά στον STM32 και δεν εξαρτάται από το επίπεδο IoT.

Στρατηγική Safe State

Η ασφαλή κατάσταση περιλαμβάνει:

- Duty cycle = 0%
- Απενεργοποίηση εξόδου PWM
- Διατήρηση βασικής επικοινωνίας UART για τηλεμετρία ή διαγνωστικούς σκοπούς
- Αναμονή χειροκίνητης ή απομακρυσμένης επανεκκίνησης (εφόσον επιτρέπεται)



Η λογική αυτή διασφαλίζει ότι η απώλεια επικοινωνίας ή η δυσλειτουργία δικτύου δεν οδηγεί σε απώλεια ελέγχου του κινητήρα.

Διαχείριση Σφαλμάτων Επικοινωνίας

Η επικοινωνία UART και MQTT θεωρείται μη κρίσιμη για τη βασική λειτουργία ελέγχου. Σε περίπτωση:

- Καθυστέρησης MQTT
- Απώλειας σύνδεσης WiFi
- Αποτυχίας βασικού ελέγχου ορθότητας δεδομένων (αποτυχία CRC)
- Μη απόκρισης του cloud

ο έλεγχος κινητήρα συνεχίζεται κανονικά με τις τελευταίες έγκυρες παραμέτρους.

Συνολική Φιλοσοφία Διασφάλισης Αξιόπιστης Λειτουργίας

Ο συνδυασμός watchdog, ελέγχου ορίων και λογικής ασφαλούς κατάστασης δημιουργεί ένα σύστημα με ανθεκτικότητα σε:

- Σφάλματα λογισμικού
- Προσωρινές αστοχίες επικοινωνίας
- Παρεμβολές ή θόρυβο στη σειριακή μετάδοση
- Ακραίες συνθήκες λειτουργίας

Η προσέγγιση αυτή καθιστά το σύστημα κατάλληλο για πειραματική αλλά και επεκτάσιμη βιομηχανική εφαρμογή, όπου η ασφάλεια και η προβλέψιμη συμπεριφορά αποτελούν βασικές προδιαγραφές.

4.4.6 Υλοποίηση Επιπέδου IoT (ESP32 και MQTT)

Το επίπεδο IoT υλοποιείται μέσω της πλακέτας ESP32, η οποία λειτουργεί ως πύλη επικοινωνίας (communication gateway) μεταξύ της μονάδας ελέγχου (STM32) και της πλατφόρμας ThingsBoard. Η μονάδα αυτή αναλαμβάνει τις λειτουργίες δικτυακής επικοινωνίας και τηλεμετρίας, διατηρώντας πλήρη απομόνωση από τον βρόχο ελέγχου πραγματικού χρόνου του STM32.

Αρχιτεκτονική Λειτουργίας ESP32

Η ESP32 εκτελεί τις ακόλουθες βασικές λειτουργίες:

- Σύνδεση σε ασύρματο δίκτυο (WiFi client mode)
- Δημιουργία και διατήρηση σύνδεσης MQTT με την πλατφόρμα ThingsBoard
- Λήψη τηλεμετρικών πλαισίων από τον STM32 μέσω UART



- Αποκωδικοποίηση και δημοσίευση δεδομένων (publish)
- Λήψη εντολών RPC και προώθησή τους στον STM32

Η λειτουργία της ESP32 βασίζεται σε ασύγχρονο μοντέλο επικοινωνίας, όπου η διαχείριση WiFi και MQTT πραγματοποιείται ανεξάρτητα από τον ρυθμό άφιξης των δεδομένων από τον STM32.

Ροή Τηλεμετρίας

Η διαδικασία μετάδοσης τηλεμετρίας περιλαμβάνει:

1. Λήψη δεδομένων μέσω UART.
2. Βασικό έλεγχο ορθότητας του πλαισίου (έλεγχο CRC).
3. Μετατροπή των τιμών σε μορφή JSON.
4. Δημοσίευση μέσω MQTT προς την πλατφόρμα ThingsBoard.

Η αποστολή τηλεμετρίας πραγματοποιείται περιοδικά ή όταν υπάρχουν νέα διαθέσιμα δεδομένα από τον STM32.

Διαχείριση RPC Εντολών

Η ESP32 μπορεί να λαμβάνει απομακρυσμένες εντολές μέσω MQTT RPC μηχανισμών της πλατφόρμας ThingsBoard. Όταν ληφθεί εντολή:

1. Ελέγχεται η δομή και η εγκυρότητα των δεδομένων.
2. Δημιουργείται αντίστοιχο πλαίσιο UART.
3. Αποστέλλεται στον STM32 μέσω interrupt-driven μετάδοσης.

Η εφαρμογή της εντολής πραγματοποιείται αποκλειστικά στον STM32, ο οποίος αξιολογεί εάν η εντολή είναι αποδεκτή βάσει των ορίων ασφαλείας.

Διαχείριση Απώλειας Σύνδεσης

Σε περίπτωση απώλειας σύνδεσης WiFi ή MQTT:

- Η ESP32 επιχειρεί επανασύνδεση με προκαθορισμένη στρατηγική (retry mechanism).
- Δεν επηρεάζεται η λειτουργία του STM32.
- Δεν μεταβάλλεται ο έλεγχος του κινητήρα.

Η αρχιτεκτονική αυτή βασίζεται στην αρχή ότι το επίπεδο IoT λειτουργεί επικουρικά και όχι ως κρίσιμος παράγοντας ελέγχου.

Διαχωρισμός Επιπέδων

Η συνολική δομή μπορεί να συνοψιστεί ως εξής:

- STM32 → Πραγματικός έλεγχος και προστασία



- ESP32 → Δικτυακή επικοινωνία
- ThingsBoard → Εποπτεία και απομακρυσμένη διαχείριση

Ο διαχωρισμός αυτός επιτρέπει:

- Τοπική αυτονομία ελέγχου
- Απομόνωση δικτυακών καθυστερήσεων
- Επεκτασιμότητα της IoT λειτουργικότητας

Το προτεινόμενο σύστημα έχει υλοποιηθεί πλήρως σε επίπεδο αρχιτεκτονικής και λογισμικού, με σαφή διαχωρισμό μεταξύ επιπέδου ελέγχου και επιπέδου IoT. Ο διαχωρισμός αυτός ενισχύει την αξιοπιστία και τη χρονική προβλεψιμότητα της λειτουργίας. Στο επόμενο κεφάλαιο παρουσιάζεται η πειραματική υλοποίηση και αξιολόγηση του συστήματος. Η συνολική αρχιτεκτονική σχεδιάστηκε έτσι ώστε οι λειτουργίες ελέγχου και ασφάλειας να παραμένουν τοπικές στον μικροελεγκτή, ενώ οι λειτουργίες επικοινωνίας και απομακρυσμένης εποπτείας να λειτουργούν επικουρικά μέσω της IoT πλατφόρμας.



5 Πειραματική Διάταξη και Αξιολόγηση Συστήματος

Στο παρόν κεφάλαιο παρουσιάζεται η πειραματική διάταξη που χρησιμοποιήθηκε για την αξιολόγηση του προτεινόμενου ενσωματωμένου συστήματος ελέγχου ηλεκτρικού κινητήρα, καθώς και η μεθοδολογία δοκιμών που εφαρμόστηκε. Στόχος είναι να επιβεβαιωθεί στην πράξη η ορθή λειτουργία της αρχιτεκτονικής που σχεδιάστηκε στο Κεφάλαιο 4 – Σχεδιασμός Συστήματος, τόσο ως προς τη συμπεριφορά του βρόχου ελέγχου όσο και ως προς τις λειτουργίες τηλεμετρίας και απομακρυσμένης εποπτείας μέσω IoT.

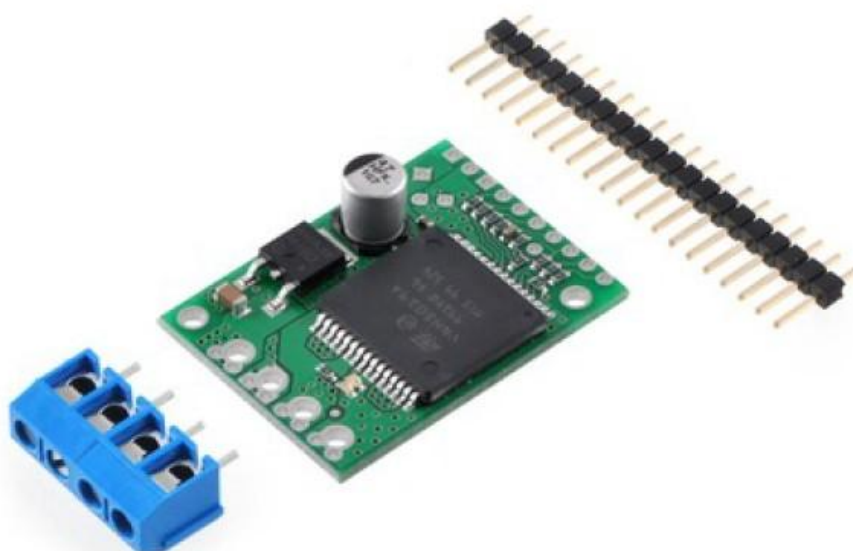
Η αξιολόγηση βασίζεται σε μετρήσεις βασικών μεγεθών λειτουργίας (όπως ταχύτητα, ρεύμα, τάση και θερμοκρασία), καθώς και σε παρατηρήσεις σχετικά με τη σταθερότητα της λειτουργίας, τη συμπεριφορά σε μεταβολές setpoint και τη συμπεριφορά του συστήματος σε περιπτώσεις απώλειας επικοινωνίας ή ενεργοποίησης βασικών μηχανισμών προστασίας. Τα δεδομένα συλλέγονται τόσο τοπικά από το επίπεδο ελέγχου όσο και μέσω της πλατφόρμας ThingsBoard, ώστε να είναι δυνατή η σύγκριση και η εξαγωγή συμπερασμάτων.

5.1 Πειραματική Διάταξη

Η πειραματική διάταξη μικρής κλίμακας αποτελείται από τον ηλεκτρικό κινητήρα, το στάδιο οδήγησης (motor driver), τη μονάδα ελέγχου (STM32), τη μονάδα επικοινωνίας (ESP32), τους αισθητήρες μέτρησης και την αντίστοιχη τροφοδοσία. Η διάταξη σχεδιάστηκε ώστε να αναπαριστά ρεαλιστικά ένα σενάριο ελέγχου κινητήρα με ενσωματωμένη εποπτεία, διατηρώντας παράλληλα απλότητα και δυνατότητα επαναληψιμότητας στις δοκιμές.



Εικόνα 5.1. Ηλεκτρικός κινητήρας πειραματικής διάταξης

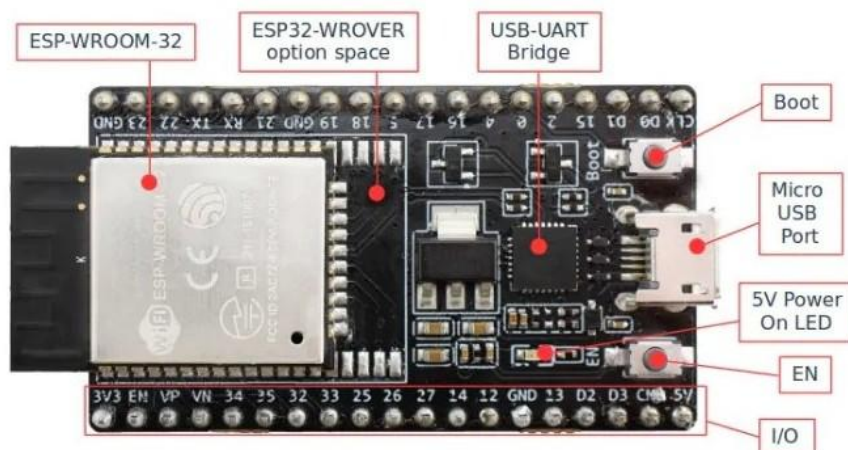


VNH5019 motor driver carrier with included hardware.

Εικόνα 5.2. Συσκευή οδήγησης ηλεκτρικού κινητήρα



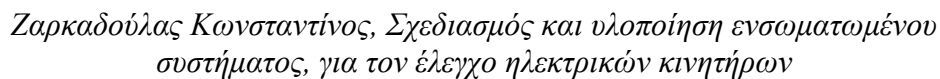
Εικόνα 5.3. Πλατφόρμα ανάπτυξης με μικροελεγκτή, STM32-Nucleo F446RE (μονάδα ελέγχου)



Εικόνα 5.4. Μονάδα επικοινωνίας ESP32

Η επιλογή των επιμέρους στοιχείων πραγματοποιήθηκε με κριτήριο τη δυνατότητα αναπαράστασης βασικών λειτουργιών ενός πραγματικού βιομηχανικού συστήματος ελέγχου, διατηρώντας παράλληλα χαμηλό κόστος και ευκολία υλοποίησης.

Η ανάπτυξη του λογισμικού πραγματοποιήθηκε στο περιβάλλον STM32CubeIDE και STM32CubeMX για τον μικροελεγκτή STM32, ενώ για τη μονάδα ESP32 χρησιμοποιήθηκε το περιβάλλον Arduino IDE. Για την υλοποίηση της επικοινωνίας αξιοποιήθηκαν βιβλιοθήκες για UART και MQTT, ενώ η διαδικασία ελέγχου και αποσφαλμάτωσης (debugging) πραγματοποιήθηκε μέσω σειριακής επικοινωνίας και εργαλείων σειριακής παρακολούθησης και αποσφαλμάτωσης.

[illegible]

Σχήμα 5.5. Κυκλωματικό διάγραμμα της πειραματικής διάταξης



Η τροφοδοσία του συστήματος πραγματοποιείται μέσω πηγής συνεχούς τάσης 12 V, η οποία χρησιμοποιείται για την οδήγηση του κινητήρα μέσω του driver, ενώ μέσω μετατροπέα υποβιβασμού τάσης (buck converter) παράγεται σταθεροποιημένη τάση 5 V για την τροφοδοσία των μικροελεγκτών και των αισθητήρων. Όλα τα υποσυστήματα μοιράζονται κοινή γείωση, ώστε να διασφαλίζεται η ορθή λειτουργία των σημάτων και η αξιόπιστη επικοινωνία.

Η μονάδα ελέγχου βασίζεται στην πλακέτα STM32 NUCLEO-F446RE, η οποία αναλαμβάνει τις χρονικά κρίσιμες λειτουργίες του συστήματος, όπως η παραγωγή σήματος PWM για την οδήγηση του κινητήρα, η συλλογή δεδομένων από τους αισθητήρες και η επεξεργασία τους σε πραγματικό χρόνο. Παράλληλα, ο μικροελεγκτής υποστηρίζει την εκτέλεση εντολών απομακρυσμένου ελέγχου που λαμβάνονται από την IoT πλατφόρμα μέσω της μονάδας ESP32, επιτρέποντας την ενεργοποίηση ή απενεργοποίηση του κινητήρα (on/off) μέσω κατάλληλων σημάτων ελέγχου καθώς και τον ορισμό setpoint για RPM. Η μονάδα επικοινωνίας υλοποιείται με χρήση της πλακέτας ESP32 DevKitC, η οποία λειτουργεί ως Gateway προς το δίκτυο, αναλαμβάνοντας τη μετάδοση των δεδομένων προς την IoT πλατφόρμα και τη λήψη απομακρυσμένων εντολών.

Η επικοινωνία μεταξύ των δύο μικροελεγκτών υλοποιείται μέσω σειριακής διεπαφής UART με ρυθμό μετάδοσης 115200 baud, με διασταύρωση των γραμμών TX–RX και κοινή γείωση.

Συγκεκριμένα, η γραμμή μετάδοσης του STM32 (PA9 – USART1_TX, pin D8 στο development board του STM32) συνδέεται με τη γραμμή λήψης του ESP32, ενώ η γραμμή λήψης του STM32 (PA10 – USART1_RX, pin D2 στο development board του STM32) συνδέεται με τη γραμμή μετάδοσης του ESP32 (pins 16 RX-17 TX), διασφαλίζοντας τη σωστή αμφίδρομη επικοινωνία μεταξύ των δύο μονάδων.

Για την πειραματική διάταξη επιλέχθηκε ένας ηλεκτρικός κινητήρας συνεχούς ρεύματος (DC) 12 V με ενσωματωμένο μειωτήρα στροφών και τετραπλό κωδικοποιητή (quadrature encoder). Ο κινητήρας διαθέτει υψηλό λόγο μετάδοσης (περίπου 100:1), προσφέροντας χαμηλή ταχύτητα περιστροφής (~100 rpm) και αυξημένη ροπή, χαρακτηριστικά που τον καθιστούν κατάλληλο για εφαρμογές ελεγχόμενης και σταθερής λειτουργίας. Επιπλέον, η ύπαρξη του κωδικοποιητή επιτρέπει την αξιόπιστη εκτίμηση της ταχύτητας, συμβάλλοντας στην παρακολούθηση και αξιολόγηση της απόδοσης του συστήματος. Η επιλογή του



συγκεκριμένου κινητήρα βασίστηκε στην ισορροπία μεταξύ απλότητας, αξιοπιστίας και συμβατότητας με ενσωματωμένα συστήματα ελέγχου.

Το στάδιο οδήγησης του κινητήρα βασίζεται στον οδηγό VNH5019, ο οποίος επιτρέπει την αμφίδρομη λειτουργία του κινητήρα μέσω των σημάτων INA και INB, ενώ η ταχύτητα ελέγχεται μέσω σήματος PWM.

Η παραγωγή του σήματος PWM υλοποιείται μέσω του χρονομετρητή TIM2 του μικροελεγκτή STM32. Το duty cycle ενημερώνεται δυναμικά από τον αλγόριθμο ελέγχου, ενώ η ίδια η παραγωγή του σήματος πραγματοποιείται εξ ολοκλήρου από το υλικό (hardware timer), εξασφαλίζοντας σταθερή συχνότητα και χαμηλό χρονικό jitter. Ενδεικτικό τμήμα του firmware που υλοποιεί την ενημέρωση του duty cycle παρουσιάζεται παρακάτω ως τμήμα του κώδικα (1).

Τμήμα Firmware (1) Υλοποίηση παραγωγής σήματος PWM μέσω του χρονομετρητή TIM2

```
void Motor_SetDuty(float duty)
{
    if (duty < 0.0f)
        duty = 0.0f;

    if (duty > 100.0f)
        duty = 100.0f;

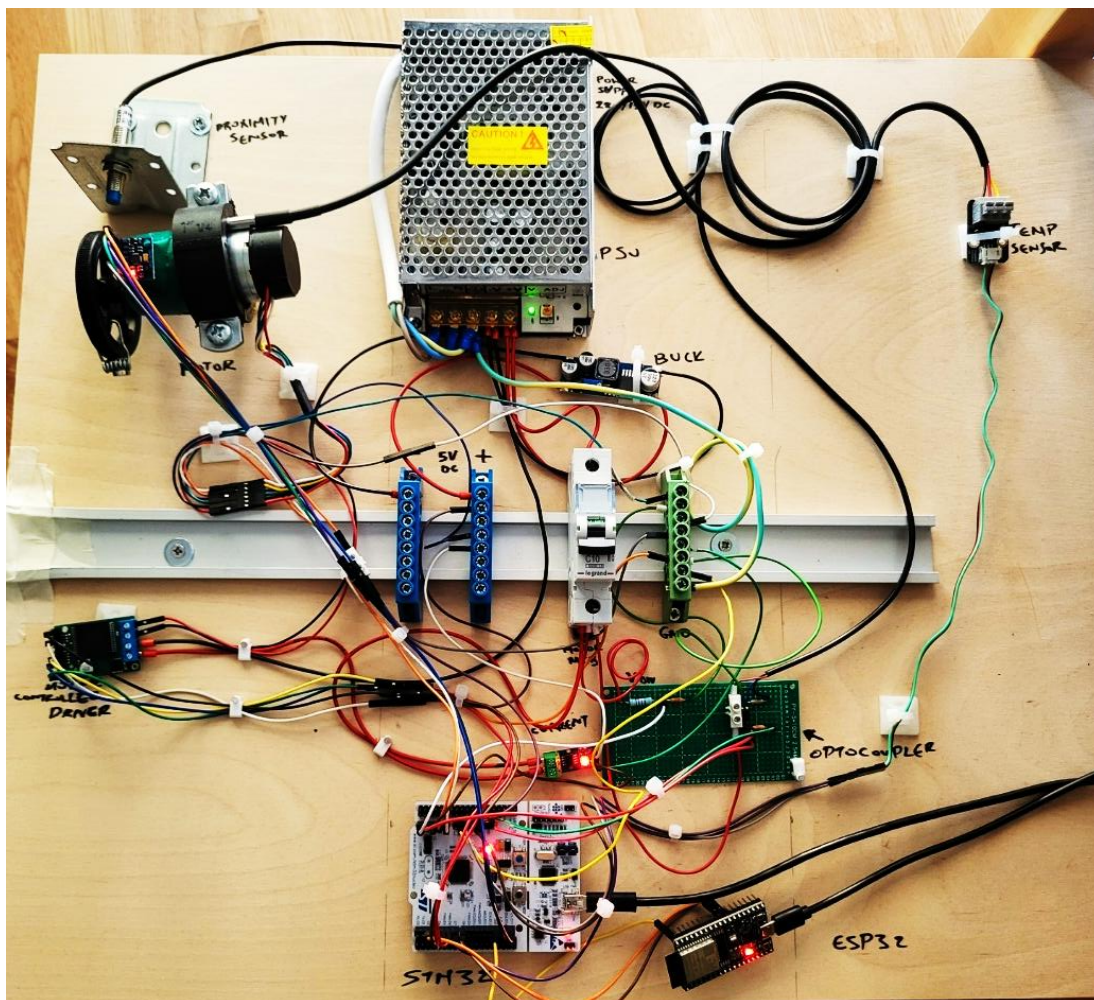
    uint32_t compare =
        (uint32_t)((duty / 100.0f) * (__HAL_TIM_GET_AUTORELOAD(&htim2)));

    __HAL_TIM_SET_COMPARE(&htim2, TIM_CHANNEL_3, compare);
}
```

Οι αισθητήρες που ενσωματώθηκαν περιλαμβάνουν αισθητήρα ρεύματος (ACS712), αισθητήρα θερμοκρασίας (DS18B20), αισθητήρα επιτάχυνσης (MPU6050), αισθητήρα εγγύτητας και encoder για τη μέτρηση της ταχύτητας περιστροφής.

Η καταμέτρηση των παλμών του encoder πραγματοποιείται μέσω μηχανισμού εξωτερικών διακοπών (EXTI), επιτρέποντας την ακριβή και αξιόπιστη εκτίμηση της ταχύτητας περιστροφής χωρίς απώλεια παλμών.

Στην Εικόνα 5.6 παρουσιάζεται η συνολική υλοποίηση της πειραματικής διάταξης στο εργαστηριακό περιβάλλον.



Εικόνα 5.6. Υλοποίηση πειραματικής διάταξης

Η διάταξη τοποθετήθηκε σε επίπεδη επιφάνεια εργαστηρίου, με σαφή διαχωρισμό των επιμέρους ενοτήτων (τροφοδοσία, έλεγχος, αισθητήρες), ώστε να διευκολύνεται η παρατήρηση και η επαναληψιμότητα των πειραμάτων.

Η πειραματική διάταξη υλοποιήθηκε με στόχο την επαλήθευση της λειτουργικότητας του συστήματος σε πραγματικές συνθήκες λειτουργίας και την αξιολόγηση της συμπεριφοράς του σε επίπεδο ελέγχου και επικοινωνίας.

Η χρήση οπτικού υλικού συμβάλλει στην τεκμηρίωση της υλοποίησης και στην καλύτερη κατανόηση της δομής του συστήματος.

5.1.1 Υποσύστημα Αισθητήρων

Το υποσύστημα αισθητήρων αποτελεί βασικό μέρος της πειραματικής διάταξης, καθώς παρέχει τα απαραίτητα δεδομένα για την παρακολούθηση της λειτουργίας του κινητήρα και την αξιολόγηση της απόδοσης του συστήματος. Οι αισθητήρες που

χρησιμοποιήθηκαν καλύπτουν διαφορετικά φυσικά μεγέθη και υλοποιούνται με διαφορετικές διεπαφές, όπως αναλογικές εισοδοι, ψηφιακές γραμμές και σειριακά πρωτόκολλα επικοινωνίας.

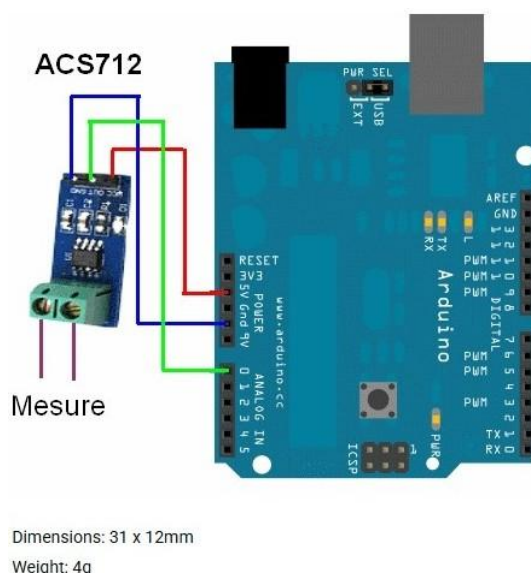
Αισθητήρας Ρεύματος (ACS712)

Ο αισθητήρας ρεύματος τύπου ACS712 βασίζεται σε φαινόμενο Hall και χρησιμοποιείται για τη μέτρηση του ρεύματος κατανάλωσης του κινητήρα.

- Τύπος σήματος: Αναλογικό
- Διεπαφή: ADC (Analog-to-Digital Converter)
- Σύνδεση: Η έξοδος του αισθητήρα συνδέεται σε αναλογική είσοδο του STM32 (PA4, pin A2 στο development board του STM32)
- Λειτουργία: Παρέχει τάση εξόδου ανάλογη του ρεύματος που διέρχεται από το κύκλωμα ισχύος
- Μετατροπή των μετρήσεων: Η έξοδος του αισθητήρα είναι κεντραρισμένη γύρω από τάση αναφοράς (περίπου $V_{cc}/2$). Το ρεύμα προκύπτει μέσω γραμμικής σχέσης:

$$I = \frac{V_{out} - V_{offset}}{Sensitivity}$$

όπου V_{offset} είναι η τάση ηρεμίας και Sensitivity η ευαισθησία του αισθητήρα (π.χ. mV/A).



Εικόνα 5.7. Αισθητήρας ρεύματος ACS712

Αισθητήρας Θερμοκρασίας (DS18B20)

Ο αισθητήρας DS18B20 χρησιμοποιείται για την παρακολούθηση της θερμοκρασίας λειτουργίας του κινητήρα.

- Πρωτόκολλο: 1-Wire
- Σύνδεση: Μία ψηφιακή γραμμή δεδομένων (PA8, pin D7 στο development board του STM32) με αντίσταση pull-up
- Λειτουργία: Ψηφιακή μέτρηση θερμοκρασίας με ενσωματωμένη μετατροπή
- Μετατροπή μετρήσεων: Η θερμοκρασία μεταδίδεται σε ψηφιακή μορφή και μετατρέπεται απευθείας σε βαθμούς Κελσίου μέσω της αντίστοιχης βιβλιοθήκης



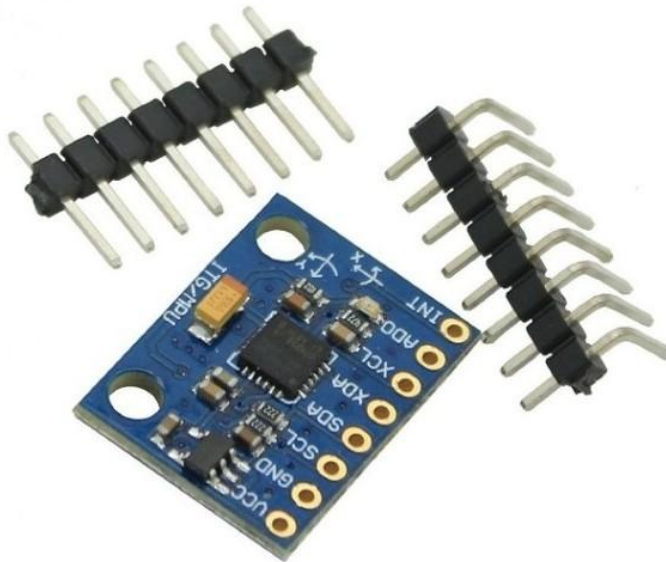
Εικόνα 5.8. Αισθητήρας θερμοκρασίας DS18B20

Αισθητήρας Επιτάχυνσης (MPU6050)

Ο αισθητήρας MPU6050 χρησιμοποιείται για την καταγραφή δυναμικών μεταβολών του συστήματος, όπως δονήσεις και μεταβολές θέσης.

- Πρωτόκολλο: I2C
- Σύνδεση:
 - SDA → PB9 (αντίστοιχο pin στο development board του STM32)
 - SCL → PB8 (αντίστοιχο pin στο development board του STM32)
- Λειτουργία: Μέτρηση επιταχύνσεων στους τρεις άξονες (X, Y, Z)

- Μετατροπή μετρήσεων: Τα δεδομένα λαμβάνονται ως ψηφιακές τιμές και μετατρέπονται σε μονάδες επιτάχυνσης (g) μέσω συντελεστών κλίμακας του αισθητήρα



Εικόνα 5.9. Αισθητήρας επιτάχυνσης MPU6050

Αισθητήρας Εγγύτητας (Inductive Proximity Sensor)

Χρησιμοποιήθηκε επαγωγικός αισθητήρας εγγύτητας: Model: LJ8A3-2-Z/AX (Proximity Inductive 8AX)

Ο αισθητήρας χρησιμοποιείται για την ανίχνευση μεταλλικού αντικειμένου που είναι τοποθετημένο στον άξονα του κινητήρα και περιστρέφεται μαζί με αυτόν.

- Τύπος σήματος: Ψηφιακό (ON/OFF)
- Τροφοδοσία: 12 V
- Η λειτουργία βασίζεται σε παλμούς ψηφιακής στάθμης (βιομηχανικός διακόπτης)
- Ενδιάμεση διάταξη: Optocoupler (PC817)

Ρόλος Optocoupler:

- Απομόνωση 12 V από 3.3 V
- Προστασία μικροελεγκτή
- Μετατροπή σήματος σε συμβατό επίπεδο λογικής



- Σύνδεση STM32: Ψηφιακή είσοδος GPIO (PA3, pin D3 στο development board)
- Λειτουργία: Παράγει παλμούς κατά τη διέλευση μεταλλικού αντικειμένου
- Χρήση: Συμπληρωματική εκτίμηση RPM και ανίχνευση γεγονότων



Εικόνα 5.10. Αισθητήρας εγγύτητας

Encoder (Ενσωματωμένος στον κινητήρα)

Ο encoder χρησιμοποιείται για την ακριβή μέτρηση της ταχύτητας περιστροφής του κινητήρα.

- Τύπος σήματος: Ψηφιακό (παλμοί)
- Δεν χρησιμοποιεί ψηφιακό πρωτόκολλο (pulse-based)
- Σύνδεση:
Channel A → PA0 (EXTI0, pin A0 στο development board του STM32)
Channel B → PA1 (EXTI1, pin A1 στο development board του STM32)
- Μηχανισμός: Εξωτερικές διακοπές (EXTI)
- Λειτουργία: Καταμέτρηση παλμών για υπολογισμό RPM
- Υπολογισμός RPM:

$$RPM = \frac{pulses}{time} \times scaling\ factor$$

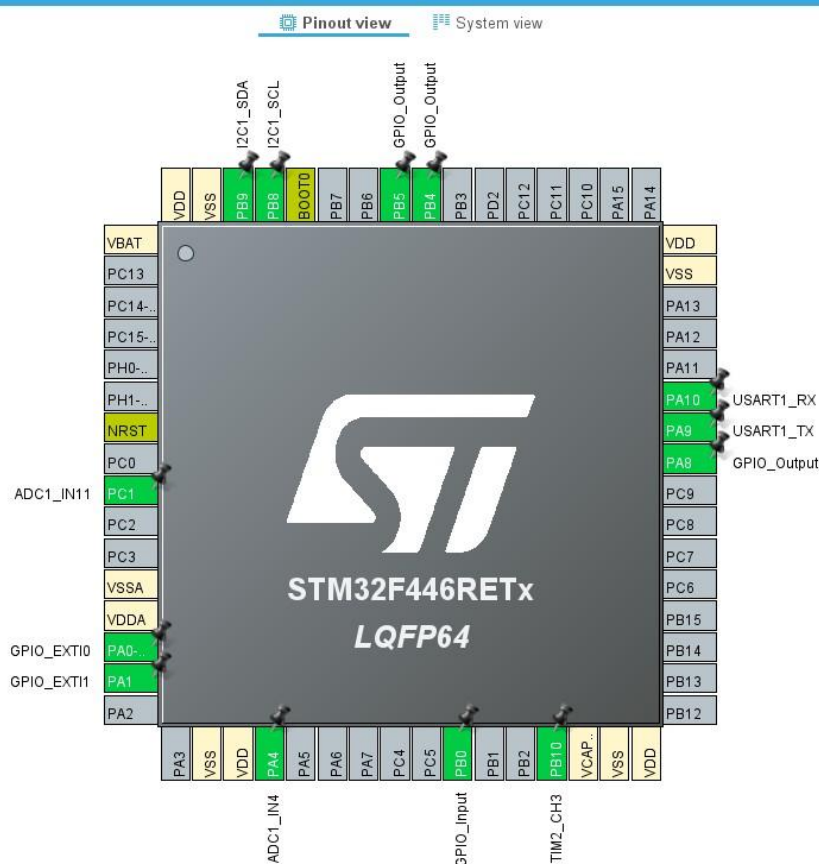


Κύκλωμα Μέτρησης Τάσης (Voltage Divider)

Η πειραματική διάταξη περιλαμβάνει επίσης κύκλωμα διαιρέτη τάσης (voltage divider), το οποίο χρησιμοποιείται για την προσαρμογή της τάσης τροφοδοσίας σε επίπεδα συμβατά με τις αναλογικές εισόδους του STM32. Παρότι το κύκλωμα αυτό δεν αποτελεί αισθητήρα με τη στενή έννοια του όρου, χρησιμοποιείται για την παρακολούθηση της τάσης τροφοδοσίας του συστήματος και την ανίχνευση πιθανών καταστάσεων υπότασης ή υπέρτασης. Επίσης είναι ένδειξη για το εάν η ασφάλεια του κυκλώματος είναι ανοιχτή/κλειστή.

- Τύπος κυκλώματος: Παθητικός διαιρέτης τάσης με αντιστάσεις
- Διεπαφή: ADC (Analog-to-Digital Converter)
- Σύνδεση: Η έξοδος του διαιρέτη συνδέεται σε αναλογική είσοδο του STM32 (PC1, pin A4 στο development board του STM32)
- Λειτουργία: Μείωση της τάσης εισόδου σε ασφαλές επίπεδο για τον μικροελεγκτή
- Χρήση: Παρακολούθηση τάσης τροφοδοσίας

Το κύκλωμα υλοποιήθηκε με χρήση αντιστάσεων 47 kΩ και 10 kΩ, επιτυγχάνοντας λόγο διαίρεσης κατάλληλο για ασφαλή σύνδεση της τάσης τροφοδοσίας στις αναλογικές εισόδους του STM32. Για τάση εισόδου 12 V, η τάση εξόδου του διαιρέτη είναι περίπου 2.1 V, τιμή συμβατή με τα επίπεδα λειτουργίας του ADC του μικροελεγκτή.



Εικόνα 5.11. Ενεργοποίηση εισόδων-εξόδων GPIO στον μικροελεγκτή από το περιβάλλον STM32CubeMX

Στην Εικόνα 5.11 παρουσιάζεται η αντιστοίχιση των βασικών εισόδων και εξόδων του μικροελεγκτή STM32F446RE που χρησιμοποιήθηκαν στην πειραματική διάταξη. Συγκεκριμένα, οι ακροδέκτες PA0 και PA1 χρησιμοποιούνται για τα κανάλια A και B του encoder και έχουν ρυθμιστεί ως εξωτερικές διακοπές (EXTI) για την καταμέτρηση παλμών και τον υπολογισμό των στροφών του κινητήρα. Ο PA4 χρησιμοποιείται ως αναλογική είσοδος ADC για τη μέτρηση ρεύματος μέσω του αισθητήρα ACS712, ενώ ο PC1 χρησιμοποιείται ως αναλογική είσοδος ADC για τη μέτρηση της τάσης τροφοδοσίας μέσω διαιρέτη τάσης. Οι PB8 και PB9 έχουν διαμορφωθεί ως δίαυλος I2C για την επικοινωνία με τον αισθητήρα MPU6050, ενώ ο PA8 χρησιμοποιείται ως ψηφιακή γραμμή 1-Wire για τον αισθητήρα θερμοκρασίας DS18B20. Ο PB0 λειτουργεί ως ψηφιακή είσοδος για τον επαγωγικό αισθητήρα εγγύτητας, οι PB4 και PB5 ως ψηφιακές εξοδοί ελέγχου κατεύθυνσης του οδηγού κινητήρα (INA/INB), ενώ ο PB10 έχει διαμορφωθεί ως έξοδος PWM μέσω του χρονομετρητή TIM2 για τον έλεγχο της ταχύτητας του κινητήρα. Τέλος, οι PA9 και PA10 χρησιμοποιούνται ως



διεπαφή UART για την επικοινωνία με τη μονάδα ESP32 και τη μετάδοση δεδομένων τηλεμετρίας και εντολών ελέγχου.

5.2 Μεθοδολογία Δοκιμών

Η αξιολόγηση του συστήματος πραγματοποιήθηκε μέσω σειράς ελεγχόμενων πειραματικών σεναρίων, τα οποία σχεδιάστηκαν ώστε να εξετάσουν τόσο τη λειτουργία του βρόχου ελέγχου όσο και τη συνολική συμπεριφορά του συστήματος σε πραγματικές συνθήκες.

Αρχικά, εξετάστηκε η βασική λειτουργία του κινητήρα τόσο σε ανοικτό όσο και σε κλειστό βρόχο ελέγχου. Στην περίπτωση ανοικτού βρόχου, η ταχύτητα καθορίζεται αποκλειστικά από το duty cycle του σήματος PWM. Πραγματοποιήθηκαν μεταβολές του duty cycle και παρατηρήθηκε η αντίστοιχη μεταβολή της ταχύτητας περιστροφής, όπως αυτή υπολογίζεται από τις μετρήσεις του encoder.

Στη συνέχεια, υλοποιήθηκε κλειστός βρόχος ελέγχου ταχύτητας με χρήση PI ελεγκτή (Proportional–Integral Controller). Ο ελεγκτής υπολογίζει το σφάλμα μεταξύ της επιθυμητής τιμής στροφών (setpoint) και της μετρούμενης ταχύτητας, ενώ η έξοδος του ελεγκτή χρησιμοποιείται για την ενημέρωση του duty cycle του PWM.

Η βασική λογική του κλειστού βρόχου ελέγχου υλοποιείται στο firmware του STM32 μέσω υπολογισμού του σφάλματος μεταξύ της επιθυμητής τιμής RPM και της μετρούμενης ταχύτητας από τον encoder. Ο αναλογικός όρος επιτρέπει την άμεση αντίδραση του συστήματος στο σφάλμα, ενώ ο ολοκληρωτικός όρος συμβάλλει στη σταδιακή μείωση της μόνιμης απόκλισης από το setpoint. Παράλληλα, εφαρμόζονται όρια τόσο στον ολοκληρωτικό όρο όσο και στο τελικό duty cycle, ώστε να αποφεύγονται υπερβολικές τιμές ελέγχου. Ενδεικτικό τμήμα του firmware που υλοποιεί τον PI ελεγκτή παρουσιάζεται παρακάτω ως τμήμα του κώδικα (2).

Τμήμα Firmware (2) Υλοποίηση PI κλειστού βρόχου ελέγχου ταχύτητας

```
control_error = target_rpm - rpmA;
integral_error += control_error * 1.0f;    // Ts = 1 second
if (integral_error > 500.0f)
{
    integral_error = 500.0f;
}
if (integral_error < -500.0f)
```



```
{  
    integral_error = -500.0f;  
}  
  
control_duty = base_duty + (Kp * control_error) + (Ki * integral_error);  
if (control_duty > 100.0f)  
{  
    control_duty = 100.0f;  
}  
if (control_duty < 0.0f)  
{  
    control_duty = 0.0f;  
}  
Motor_Forward((uint16_t)control_duty);
```

Η υλοποίηση περιλαμβάνει μεταβλητές για:

- επιθυμητή ταχύτητα περιστροφής (target_rpm),
- σφάλμα ελέγχου (control_error),
- duty cycle ελέγχου (control_duty),
- αναλογικό κέρδος (Kp),
- ολοκληρωτικό κέρδος (Ki),
- καθώς και ολοκληρωτικό όρο σφάλματος (integral_error).

Οι παράμετροι του ελεγκτή ρυθμίστηκαν πειραματικά, με ενδεικτικές τιμές:

- $K_p = 0.05$
- $K_i = 0.02$

Η προσέγγιση αυτή επιτρέπει σταδιακή διόρθωση της ταχύτητας περιστροφής και σταθεροποίηση της λειτουργίας του κινητήρα γύρω από την επιθυμητή τιμή λειτουργίας.

Στη συνέχεια, αξιολογήθηκε η απόκριση του συστήματος σε διακριτές μεταβολές λειτουργίας (εκκίνηση, παύση, αλλαγή setpoint), με στόχο την παρατήρηση της σταθερότητας και της ομαλότητας της λειτουργίας.

Παράλληλα, πραγματοποιήθηκε συνεχής καταγραφή των μεγεθών:

- ταχύτητα περιστροφής (RPM),
- ρεύμα λειτουργίας,
- θερμοκρασία,



- δεδομένα επιτάχυνσης,
- γεγονότα από τον αισθητήρα εγγύτητας.

Τα δεδομένα αυτά συλλέγονται από τον μικροελεγκτή και μεταδίδονται μέσω UART στη μονάδα ESP32, η οποία τα προωθεί στην πλατφόρμα ThingsBoard μέσω πρωτοκόλλου MQTT.

Η μορφή των δεδομένων κατά τη μετάδοση είναι δομημένη, ώστε να επιτρέπεται η εύκολη επεξεργασία και μετατροπή τους σε μορφή JSON. Για τον σκοπό αυτό, εφαρμόζεται κλιμάκωση των τιμών ($\times 100$), ώστε να αποφεύγεται η χρήση κινητής υποδιαστολής στο ενσωματωμένο σύστημα.

Επιπλέον, εξετάστηκε η λειτουργία απομακρυσμένου ελέγχου μέσω της πλατφόρμας, όπου ο χρήστης μπορεί να ενεργοποιεί ή να απενεργοποιεί τον κινητήρα και να ορίζει το RPM setpoint μέσω RPC εντολών. Η λειτουργία επιβεβαιώθηκε πειραματικά με παρακολούθηση της ροής εντολών από το cloud προς το σύστημα και της αντίστοιχης απόκρισης του κινητήρα.

Τέλος, πραγματοποιήθηκαν δοκιμές ανθεκτικότητας, όπως:

- προσωρινή απώλεια επικοινωνίας,
- εισαγωγή μη έγκυρων ή ελλιπών δεδομένων στη σειριακή επικοινωνία,
- αστάθεια σε αισθητήρες (π.χ. θερμοκρασία, δονήσεις),

με στόχο την αξιολόγηση της συμπεριφοράς του συστήματος σε μη ιδανικές συνθήκες.

Οι δοκιμές πραγματοποιήθηκαν σε ελεγχόμενο εργαστηριακό περιβάλλον, με επαναλαμβανόμενες μετρήσεις και παρατήρηση της συμπεριφοράς του συστήματος υπό διαφορετικά σενάρια λειτουργίας.

Η μεθοδολογία που ακολουθήθηκε επιτρέπει την ποιοτική και ποσοτική αξιολόγηση της απόδοσης του συστήματος, διασφαλίζοντας τη δυνατότητα σύγκρισης μεταξύ διαφορετικών σεναρίων λειτουργίας.



5.3 Μετρήσεις και Συλλογή Δεδομένων

Οι μετρήσεις συλλέγονται από τον STM32 και μεταδίδονται προς την πλατφόρμα IoT για απομακρυσμένη εποπτεία και καταγραφή.

Η μετάδοση τηλεμετρίας πραγματοποιείται με συχνότητα περίπου 10 Hz, ενώ ο έλεγχος του κινητήρα πραγματοποιείται σε πολύ υψηλότερη συχνότητα μέσω του hardware PWM, διασφαλίζοντας ομαλή λειτουργία.

Τα δεδομένα που συλλέγονται περιλαμβάνουν:

- ταχύτητα κινητήρα (RPM),
- ρεύμα κατανάλωσης,
- θερμοκρασία,
- επιταχύνσεις στους τρεις άξονες,
- καταγραφή γεγονότων εγγύτητας.
- Η αξιοπιστία των δεδομένων διασφαλίζεται μέσω:
- δομημένης μορφής μετάδοσης,
- φιλτραρίσματος μη έγκυρων μηνυμάτων στο επίπεδο ESP32,
- ελέγχου εγκυρότητας των τιμών.

Ιδιαίτερη προσοχή δόθηκε στη διαχείριση των αριθμητικών δεδομένων, με χρήση ακέραιων τιμών αντί κινητής υποδιαστολής, ώστε να μειωθεί το υπολογιστικό φορτίο και να αυξηθεί η σταθερότητα της επικοινωνίας.

Η επιλογή της συχνότητας μετάδοσης τηλεμετρίας αποτελεί συμβιβασμό μεταξύ ακρίβειας παρακολούθησης και φόρτου επικοινωνίας.

Τα δεδομένα χρησιμοποιούνται για την εποπτεία της λειτουργίας του συστήματος και για την αξιολόγηση της συμπεριφοράς του υπό διαφορετικές συνθήκες λειτουργίας.



5.4 Παρατηρήσεις και Περιορισμοί Συστήματος

Κατά τη διάρκεια των πειραμάτων παρατηρήθηκαν ορισμένοι πρακτικοί περιορισμοί και αποκλίσεις, οι οποίοι σχετίζονται κυρίως με τη φύση της πειραματικής διάταξης και των χρησιμοποιούμενων αισθητήρων.

Ειδικότερα παρατηρήθηκαν:

- περιστασιακή αστάθεια στις μετρήσεις του αισθητήρα θερμοκρασίας DS18B20, πιθανώς λόγω των χρονικών απαιτήσεων του πρωτοκόλλου 1-Wire,
- περιορισμένος αριθμός περιστασιακών αλλοιωμένων δεδομένων στη σειριακή επικοινωνία UART,
- μικρές διακυμάνσεις στις μετρήσεις RPM κατά τις απότομες μεταβολές φορτίου ή duty cycle,
- περιορισμένη ακρίβεια του PI ελεγκτή λόγω της πειραματικής και απλοποιημένης ρύθμισης των παραμέτρων K_p και K_i .

Παρά τους παραπάνω περιορισμούς, η συνολική λειτουργία του συστήματος κρίθηκε ικανοποιητική για την επιβεβαίωση της προτεινόμενης αρχιτεκτονικής, της λειτουργίας του βρόχου ελέγχου και της απομακρυσμένης εποπτείας μέσω IoT.

Οι περιορισμοί αυτοί δεν επηρεάζουν τη βασική λειτουργικότητα του συστήματος, αλλά αναδεικνύουν σημεία πιθανής μελλοντικής βελτίωσης, όπως:

- βελτιστοποίηση του closed-loop ελέγχου,
- βελτίωση της φιλτραρισμένης επεξεργασίας μετρήσεων,
- ενίσχυση των μηχανισμών ελέγχου ακεραιότητας επικοινωνίας.

Οι παραπάνω παρατηρήσεις θεωρούνται αναμενόμενες για πειραματική διάταξη μικρής κλίμακας και δεν αναιρούν τη συνολική αξιοπιστία της αρχιτεκτονικής του συστήματος.

5.5 Υλοποίηση και Αξιοποίηση Πλατφόρμας ThingsBoard

Η πλατφόρμα ThingsBoard χρησιμοποιήθηκε για την απομακρυσμένη εποπτεία και τον έλεγχο του συστήματος μέσω διαδικτύου. Αποτελεί μια ολοκληρωμένη λύση IoT, η οποία παρέχει δυνατότητες συλλογής, αποθήκευσης και οπτικοποίησης δεδομένων, καθώς και αποστολής εντολών προς τις συνδεδεμένες συσκευές.



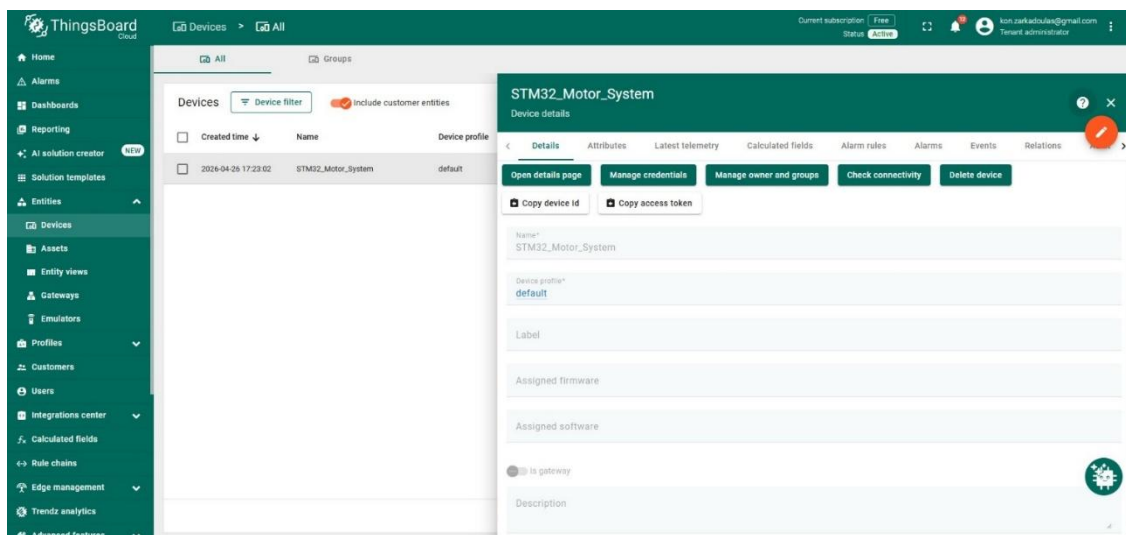
Η ενσωμάτωση της πλατφόρμας στο σύστημα επιτρέπει τη μετατροπή της πειραματικής διάταξης σε ένα λειτουργικό σύστημα απομακρυσμένης εποπτείας και τηλεμετρίας, επεκτείνοντας τις δυνατότητες του ενσωματωμένου ελέγχου.

5.5.1 Δημιουργία Συσκευής και Ρύθμιση Επικοινωνίας

Αρχικά, δημιουργήθηκε μια νέα συσκευή (device) στην πλατφόρμα ThingsBoard, η οποία αντιστοιχεί στο σύστημα ελέγχου του κινητήρα. Στη συνέχεια, χρησιμοποιήθηκε το μοναδικό access token της συσκευής για την ταυτοποίηση της μονάδας ESP32 κατά τη σύνδεση μέσω πρωτοκόλλου MQTT.

Η μονάδα ESP32 συνδέεται στο ασύρματο δίκτυο και δημιουργεί σύνδεση με τον MQTT broker της πλατφόρμας, μέσω της οποίας αποστέλλονται τα δεδομένα τηλεμετρίας.

Στην Εικόνα 5.12 παρουσιάζεται η καταχώρηση της συσκευής στην πλατφόρμα.



Εικόνα 5.12. Καταχώρηση συσκευής στην πλατφόρμα Thingsboard

5.5.2 Μετάδοση και Απεικόνιση Τηλεμετρίας

Τα δεδομένα που συλλέγονται από τον μικροελεγκτή STM32 μεταδίδονται μέσω της μονάδας ESP32 προς την πλατφόρμα σε μορφή δομημένων μηνυμάτων. Στην πλευρά του ESP32, τα δεδομένα μετατρέπονται σε μορφή JSON, ώστε να είναι συμβατά με τη δομή της πλατφόρμας.

Πριν από τη μετάδοση προς τη μονάδα ESP32, ο μικροελεγκτής STM32 συγκεντρώνει τις μετρήσεις των αισθητήρων και τις μορφοποιεί σε ένα δομημένο τηλεμετρικό πλαίσιο (telemetry frame), το οποίο μεταδίδεται μέσω της σειριακής διεπαφής UART.



Στο πλαίσιο περιλαμβάνονται οι βασικές παράμετροι λειτουργίας του συστήματος, όπως οι στροφές του κινητήρα, η επιθυμητή τιμή RPM, το duty cycle του PWM, το ρεύμα, η τάση τροφοδοσίας, η θερμοκρασία, οι μετρήσεις του αισθητήρα επιτάχυνσης, καθώς και οι σημαίες σφαλμάτων. Ενδεικτικό τμήμα του firmware παρουσιάζεται παρακάτω ως Τμήμα Firmware (5).

Τμήμα Firmware (5). Δημιουργία και μορφοποίηση τηλεμετρικού πλαισίου πριν από τη μετάδοση μέσω UART.

```
snprintf(uart_msg, sizeof(uart_msg),
"RPM=%d;"
"TARGET=%d;"
"DUTY=%d;"
"CUR=%d;"
"VOLT=%d;"
"TEMP=%d;"
"AX=%d;"
"AY=%d;"
"AZ=%d;"
"PROX=%lu;"
"FAULT=%d\r\n",
...
);
```

Μετά τη λήψη του τηλεμετρικού πλαισίου μέσω της διεπαφής UART, η μονάδα ESP32 αναλύει τα επιμέρους πεδία, επαναφέρει τις αριθμητικές τιμές στην πραγματική τους κλίμακα και δημιουργεί το αντίστοιχο μήνυμα JSON, το οποίο δημοσιεύεται στην πλατφόρμα ThingsBoard μέσω του πρωτοκόλλου MQTT. Ενδεικτικό τμήμα του firmware της ESP32 παρουσιάζεται παρακάτω ως Τμήμα Firmware (6).

Τμήμα Firmware (6). Μετατροπή τηλεμετρίας σε JSON και δημοσίευση μέσω MQTT από την ESP32

```
void publishTelemetry(String line)
{
int rpmA = extractValue(line, "RPM_A");
int rpmB = extractValue(line, "RPM_B");
int cur = extractValue(line, "CUR");
int temp = extractValue(line, "TEMP");
int volt = extractValue(line, "VOLT");
String payload = "{";

payload += "\"rpmA\":" + String(rpmA / 100.0, 2) + ",";
payload += "\"rpmB\":" + String(rpmB / 100.0, 2) + ",";
```



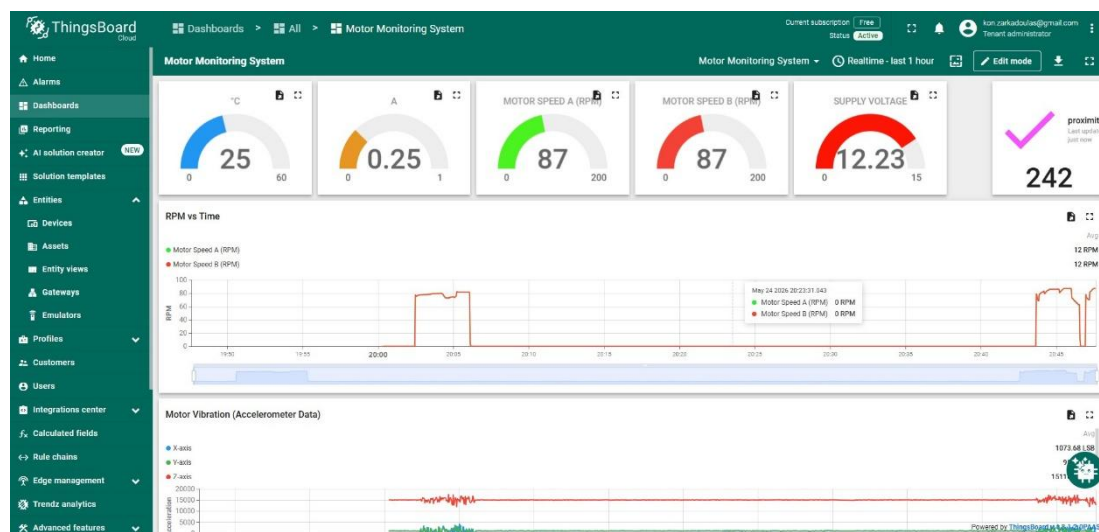
```
payload += "\"current\": \" + String(cur / 100.0, 2) + \",\";
payload += "\"temperature\": \" + String(temp / 100.0, 2) + \",\"; payload +=
\"voltage\": \" + String(volt / 100.0, 2);
payload += \"}\";
client.publish(\"v1/devices/me/telemetry\", payload.c_str());
}
```

Οι βασικές παράμετροι που μεταδίδονται περιλαμβάνουν:

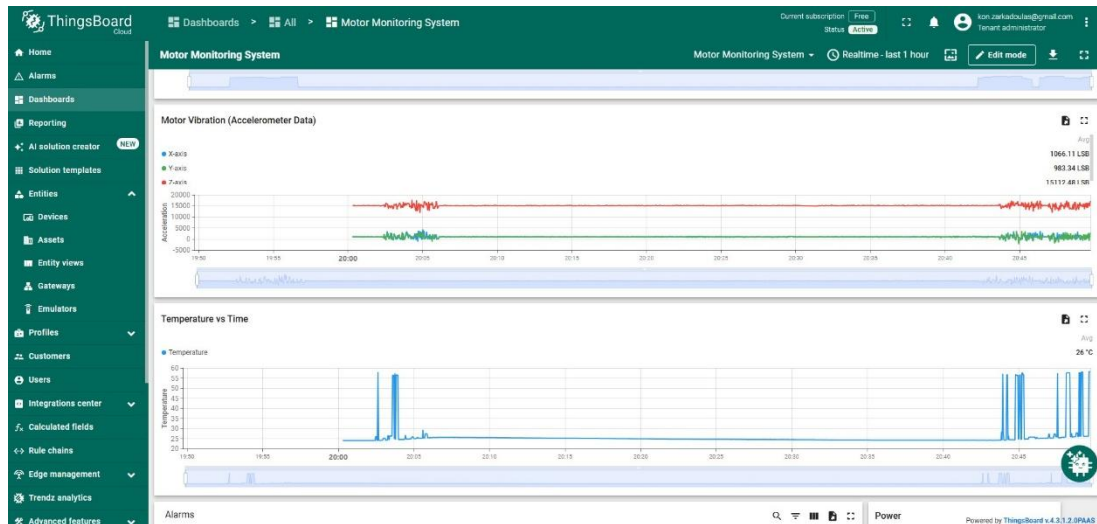
- ταχύτητα κινητήρα (RPM),
- ρεύμα κατανάλωσης,
- θερμοκρασία,
- δεδομένα επιτάχυνσης,
- γεγονότα από τον αισθητήρα εγγύτητας.

Η πλατφόρμα αποθηκεύει τα δεδομένα και επιτρέπει την απεικόνισή τους σχεδόν σε πραγματικό χρόνο μέσω διαδραστικών γραφημάτων.

Στην Εικόνα 5.13 και στην Εικόνα 5.14 παρουσιάζεται το dashboard που έχουμε δημιουργήσει στην πλατφόρμα με τις βασικές μετρήσεις του συστήματος.



Εικόνα 5.13. Dashboard της πλατφόρμας Thingsboard με indicators τύπου arc gauge



Εικόνα 5.14. Dashboard της πλατφόρμας Thingsboard με γραφικές βασικών παραμέτρων συναρτήσει του χρόνου

5.5.3 Υλοποίηση Απομακρυσμένου Ελέγχου (RPC)

Επιπλέον, υλοποιήθηκε δυνατότητα απομακρυσμένου ελέγχου του κινητήρα μέσω της πλατφόρμας, με χρήση μηχανισμού Remote Procedure Calls (RPC). Μέσω του dashboard, ο χρήστης μπορεί να αποστέλλει εντολές ενεργοποίησης ή απενεργοποίησης του κινητήρα ή αλλαγή της ταχύτητας περιστροφής από slider widget.

Οι εντολές αυτές αποστέλλονται από την πλατφόρμα προς την ESP32 μέσω MQTT και στη συνέχεια μετατρέπονται σε σειριακά μηνύματα, τα οποία αποστέλλονται στον STM32. Ο μικροελεγκτής λαμβάνει τις εντολές και, εφόσον αυτές είναι έγκυρες, εκτελεί την αντίστοιχη λειτουργία, ενεργοποιώντας, απενεργοποιώντας ή τροποποιώντας τη λειτουργία του PWM.

Η λήψη των εντολών στον STM32 πραγματοποιείται μέσω διακοπών UART (interrupt-driven reception). Κάθε εισερχόμενο byte αποθηκεύεται προσωρινά σε λογισμικό buffer μέχρι να ολοκληρωθεί η λήψη της εντολής. Στη συνέχεια, η εντολή προωθείται στη ρουτίνα επεξεργασίας, όπου αναγνωρίζεται ο τύπος της και εκτελείται η αντίστοιχη λειτουργία. Η προσέγγιση αυτή επιτρέπει την ασύγχρονη λήψη εντολών χωρίς να παρεμβάλλεται στη λειτουργία του βρόχου ελέγχου. Ενδεικτικό τμήμα του firmware παρουσιάζεται παρακάτω ως Τμήμα Firmware (3).

Τμήμα Firmware (3). Υλοποίηση λήψης εντολών μέσω UART με χρήση διακοπών.

```
volatile char uart_rx_buffer[64];  
volatile uint8_t uart_rx_index = 0;
```



```
void HAL_UART_RxCpltCallback(UART_HandleTypeDef *huart)
{
    if (uart_rx_byte == '\n' || uart_rx_byte == '\r')
    {
        uart_rx_buffer[uart_rx_index] = '\0';
        UART_ProcessCommand((char*)uart_rx_buffer);
        uart_rx_index = 0;
    }
    else
    {
        if (uart_rx_index < sizeof(uart_rx_buffer) - 1)
            uart_rx_buffer[uart_rx_index++] = uart_rx_byte;
    }

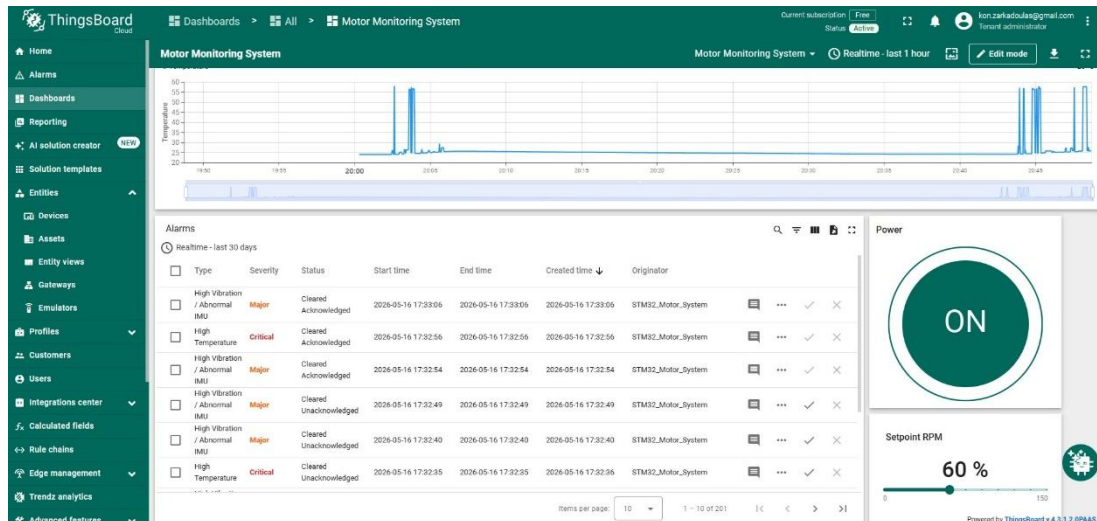
    HAL_UART_Receive_IT(&huart1, (uint8_t*)&uart_rx_byte, 1);
}
```

Η λήψη των δεδομένων ολοκληρώνεται όταν ανιχνευθεί χαρακτήρας τερματισμού γραμμής (newline ή carriage return). Στη συνέχεια, το περιεχόμενο του buffer προωθείται στη συνάρτηση επεξεργασίας εντολών, η οποία αναγνωρίζει τις εντολές απομακρυσμένου ελέγχου.

Τμήμα Firmware (4). Επεξεργασία εντολών απομακρυσμένου ελέγχου (RPC) στον STM32.

```
static void UART_ProcessCommand(char *cmd)
{
    if(strcmp(cmd,"MOTOR_ON")==0)
    {
        motor_remote_enable = 1;
    }
    else if(strcmp(cmd,"MOTOR_OFF")==0)
    {
        motor_remote_enable = 0;
    }
    else if(strncmp(cmd,"SET_RPM=",8)==0)
    {
        target_rpm = atof(cmd + 8);
    }
}
```

Στην Εικόνα 5.15 παρουσιάζεται παράδειγμα αποστολής εντολής μέσω της πλατφόρμας.



Εικόνα 5.15. Κουμπί για απομακρυσμένη εντολή ON/OFF και setpoint RPM μέσω του dashboard της πλατφόρμας Thingsboard. Αριστερά τους φαίνεται η κάρτα με τα ενεργοποιημένα alarms

Στην πλευρά της ESP32, οι εντολές RPC λαμβάνονται μέσω MQTT, μετατρέπονται στην αντίστοιχη σειριακή μορφή (MOTOR_ON, MOTOR_OFF ή SET_RPM) και προωθούνται στον STM32 μέσω της διεπαφής UART.

5.5.4 Σύστημα Συναγερμών (Alarm System)

Στο πλαίσιο της πλατφόρμας ThingsBoard υλοποιήθηκε μηχανισμός συναγερμών (alarms), με στόχο την ανίχνευση μη φυσιολογικών ή κρίσιμων συνθηκών λειτουργίας του κινητήρα και την άμεση ενημέρωση του χρήστη.

Οι συναγερμοί βασίζονται στα δεδομένα τηλεμετρίας που αποστέλλονται από το ενσωματωμένο σύστημα μέσω της αλυσίδας επικοινωνίας STM32 → ESP32 → ThingsBoard. Τα βασικά μεγέθη που αξιοποιούνται περιλαμβάνουν τη θερμοκρασία, το ρεύμα κατανάλωσης, τα δεδομένα του αισθητήρα εγγύτητας και τις επιταχύνσεις που καταγράφονται από τον αισθητήρα MPU6050.

Η υλοποίηση πραγματοποιείται μέσω του Rule Engine της πλατφόρμας, όπου τα εισερχόμενα δεδομένα αξιολογούνται σε πραγματικό χρόνο και ενεργοποιούνται συναγερμοί όταν ικανοποιούνται προκαθορισμένες συνθήκες.

Στο σύστημα υλοποιήθηκαν οι ακόλουθοι βασικοί τύποι συναγερμών:

Συναγερμός Υψηλής Θερμοκρασίας: Ο συναγερμός ενεργοποιείται όταν η θερμοκρασία υπερβεί ένα ανώτερο όριο (π.χ. 40 °C) και απενεργοποιείται όταν επανέλθει κάτω από ένα χαμηλότερο όριο (π.χ. 28 °C), ώστε να αποφεύγονται



συνεχείς εναλλαγές κατάστασης. Ο συναγερμός αυτός έχει επίπεδο σοβαρότητας *critical* και αποσκοπεί στην ανίχνευση συνθηκών υπερθέρμανσης του κινητήρα.

Συναγερμός Υπερέντασης: Ενεργοποιείται όταν το ρεύμα κατανάλωσης υπερβεί προκαθορισμένο όριο και απενεργοποιείται όταν επανέλθει σε ασφαλή επίπεδα. Τα όρια καθορίζονται πειραματικά, ανάλογα με τη συμπεριφορά του συστήματος. Ο συναγερμός έχει επίπεδο σοβαρότητας *major* και χρησιμοποιείται για την ανίχνευση αυξημένου φορτίου ή αυξημένου φορτίου ή μη φυσιολογικής λειτουργίας.

Συναγερμός Απώλειας Στόχου: Βασίζεται στον αισθητήρα εγγύτητας και ενεργοποιείται όταν δεν ανιχνεύεται το αναμενόμενο μεταλλικό αντικείμενο για συγκεκριμένο χρονικό διάστημα. Ο συναγερμός χρησιμοποιείται κυρίως για πειραματική επιβεβαίωση της λειτουργίας του αισθητήρα και της αλυσίδας τηλεμετρίας και έχει επίπεδο σοβαρότητας *warning*.

Συναγερμός Κραδασμών: Βασίζεται στις μετρήσεις του αισθητήρα MPU6050 και ενεργοποιείται όταν η επιτάχυνση στον άξονα Z υπερβεί ένα προκαθορισμένο όριο, υποδεικνύοντας αυξημένες δονήσεις ή δυναμικές μεταβολές. Ο συναγερμός απενεργοποιείται όταν οι τιμές επανέλθουν σε φυσιολογικά επίπεδα και έχει επίπεδο σοβαρότητας *major*.

Κύκλος Ζωής και Διαχείριση

Κάθε συναγερμός ακολουθεί έναν κύκλο ζωής που περιλαμβάνει τις καταστάσεις ενεργός (*active*), αναγνωρισμένος (*acknowledged*) και ανενεργός (*cleared*), οι οποίες διαχειρίζονται μέσω του μηχανισμού *alarms* της πλατφόρμας ThingsBoard.

Η απεικόνιση των συναγερμών πραγματοποιείται μέσω του *dashboard*, επιτρέποντας στον χρήστη την παρακολούθηση της κατάστασης του συστήματος σε πραγματικό χρόνο, καθώς και την αναγνώριση και διαχείριση των συμβάντων.

Σχεδιαστική Προσέγγιση

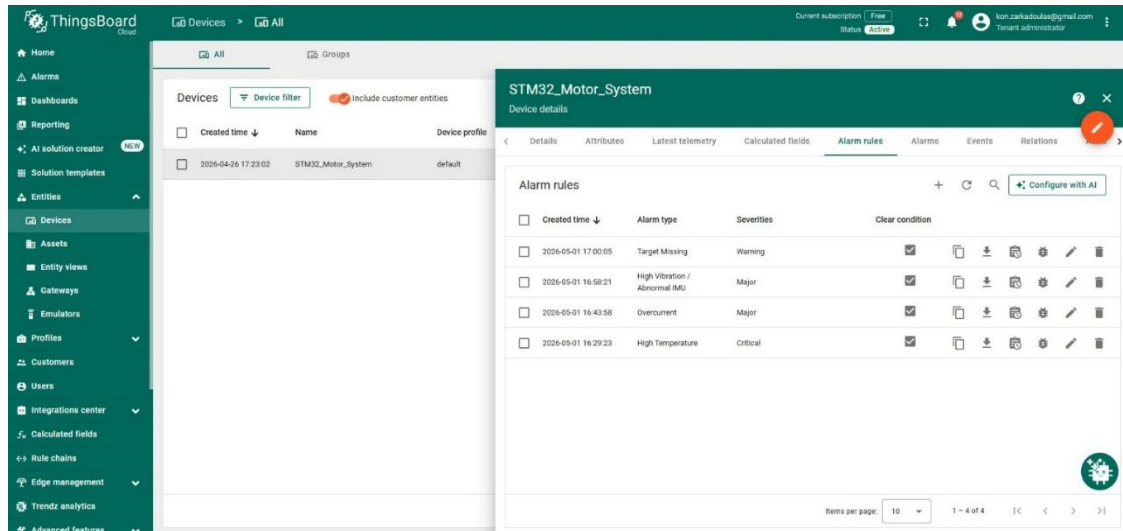
Η λογική των συναγερμών εποπτείας υλοποιείται στο επίπεδο της πλατφόρμας, ενώ οι βασικοί τοπικοί έλεγχοι ορίων παραμένουν στο *firmware* του STM32., γεγονός που παρέχει αυξημένη ευελιξία, καθώς επιτρέπει την τροποποίηση των ορίων και των συνθηκών χωρίς την ανάγκη επαναπρογραμματισμού του ενσωματωμένου συστήματος.

Η επιλογή αυτή διαχωρίζει σαφώς το επίπεδο ελέγχου από το επίπεδο εποπτείας και ενισχύει την επεκτασιμότητα της αρχιτεκτονικής.

Συμβολή στο Σύστημα



Η ενσωμάτωση του μηχανισμού συναγερμών μετατρέπει το σύστημα από απλή πλατφόρμα παρακολούθησης σε σύστημα ενεργής εποπτείας, επιτρέποντας την έγκαιρη ανίχνευση ανωμαλιών και δημιουργώντας τη βάση για μελλοντικές εφαρμογές προγνωστικής συντήρησης.



Εικόνα 5.16. Δημιουργία alarm rules για το σύστημα ειδοποιήσεων (μηχανισμός συναγερμών) της διάταξης

5.5.5 Τοπικός Έλεγχος Ορίων και Βασική Προστασία Συστήματος μέσω Firmware

Πέρα από τους συναγερμούς που υλοποιούνται στο επίπεδο της πλατφόρμας ThingsBoard, βασικοί έλεγχοι ορίων πραγματοποιούνται και τοπικά στο firmware του STM32. Συγκεκριμένα, ελέγχονται συνθήκες υπερέντασης, υπότασης και υπέρτασης της τροφοδοσίας. Σε περίπτωση παραβίασης κάποιου ορίου, ενεργοποιείται αντίστοιχη σημαία σφάλματος, απενεργοποιείται η εντολή λειτουργίας του κινητήρα, μηδενίζεται το duty cycle και εκτελείται παύση του κινητήρα. Με τον τρόπο αυτό, η βασική αντίδραση προστασίας παραμένει τοπική και ανεξάρτητη από τη σύνδεση με την πλατφόρμα IoT.

Τμήμα Firmware (7). Τοπικός έλεγχος ορίων και παύση κινητήρα σε κατάσταση σφάλματος

```
static void Safety_Check(void)
{
    fault_overcurrent = 0;
    fault_undervoltage = 0;
    fault_overvoltage = 0;
    system_fault = 0;

    if (current_amps > CURRENT_LIMIT_A)
```



```
{  
  fault_overcurrent = 1;  
  system_fault = 1;  
}  
  
if (supply_voltage < VOLTAGE_MIN_V)  
{  
  fault_undervoltage = 1;  
  system_fault = 1;  
}  
  
if (supply_voltage > VOLTAGE_MAX_V)  
{  
  fault_overvoltage = 1;  
  system_fault = 1;  
}  
  
if (system_fault)  
{  
  motor_remote_enable = 0;  
  control_duty = 0.0f;  
  integral_error = 0.0f;  
  Motor_Stop();  
}  
}
```

5.5.6 Συνολική Αξιολόγηση Πλατφόρμας

Η χρήση της πλατφόρμας ThingsBoard αποδείχθηκε ιδιαίτερα αποτελεσματική για την υλοποίηση λειτουργιών απομακρυσμένης εποπτείας και βασικού απομακρυσμένου ελέγχου του συστήματος. Η μετάδοση των δεδομένων πραγματοποιείται με μικρή καθυστέρηση και η απεικόνιση των μετρήσεων μέσω των dashboards είναι σχεδόν άμεση.

Η πλατφόρμα επιτρέπει την εύκολη επέκταση του συστήματος με επιπλέον αισθητήρες ή λειτουργίες, ενώ η υποστήριξη του πρωτοκόλλου MQTT διευκολύνει την ενσωμάτωση σε σύγχρονες IoT αρχιτεκτονικές.

Η αρχιτεκτονική που υλοποιήθηκε επιβεβαιώνει τον σαφή διαχωρισμό μεταξύ επιπέδου ελέγχου (STM32) και επιπέδου επικοινωνίας και εποπτείας (ESP32 και cloud), διασφαλίζοντας ότι η λειτουργία του βρόχου ελέγχου δεν επηρεάζεται από πιθανές καθυστερήσεις του δικτύου ή της πλατφόρμας IoT.

Η πειραματική υλοποίηση επιβεβαιώνει ότι η ενσωμάτωση cloud πλατφόρμας μπορεί να πραγματοποιηθεί χωρίς να επηρεάζεται η χρονικά κρίσιμη λειτουργία του συστήματος, επιτρέποντας παράλληλα απομακρυσμένη εποπτεία, τηλεμετρία και βασικές λειτουργίες διαχείρισης.



5.6 Ανάλυση Αποτελεσμάτων

Η ανάλυση των αποτελεσμάτων βασίζεται στα δεδομένα που συλλέχθηκαν κατά τη διάρκεια των πειραματικών δοκιμών και αποσκοπεί στην αξιολόγηση της συμπεριφοράς του συστήματος τόσο σε επίπεδο ελέγχου όσο και σε επίπεδο επικοινωνίας και εποπτείας.

Η ανάλυση πραγματοποιείται τόσο ποιοτικά όσο και ποσοτικά, με βάση τα καταγεγραμμένα δεδομένα.

5.6.1 Επιβεβαίωση Πειραματικών Μετρήσεων και Λειτουργίας Συστήματος

Για την αξιολόγηση της λειτουργίας του συστήματος πραγματοποιήθηκαν διαδοχικές δοκιμές τόσο σε ανοικτό όσο και σε κλειστό βρόχο ελέγχου. Στο αρχικό στάδιο αξιολόγησης εφαρμόστηκαν διαφορετικές τιμές duty cycle PWM, προκειμένου να επιβεβαιωθεί η βασική λειτουργία του κινητήρα και η ορθή απόκριση του συστήματος. Στη συνέχεια, πραγματοποιήθηκαν δοκιμές κλειστού βρόχου με χρήση PI ελεγκτή και διαφορετικές τιμές επιθυμητής ταχύτητας (RPM setpoint). Οι μετρήσεις ταχύτητας που εμφανίζονταν στο dashboard της πλατφόρμας ThingsBoard συγκρίθηκαν με τις αντίστοιχες τιμές αναφοράς και παρατηρήθηκε επιτυχής προσέγγιση και διατήρηση της επιθυμητής ταχύτητας, ακόμη και σε περιπτώσεις μεταβολής του μηχανικού φορτίου.

Η θερμοκρασία που καταγραφόταν από τον αισθητήρα DS18B20 ελέγχθηκε συγκριτικά με εξωτερικό θερμόμετρο χώρου, επιβεβαιώνοντας ότι οι μετρήσεις του συστήματος παρουσιάζουν ρεαλιστικές τιμές θερμοκρασίας κατά τη λειτουργία της πειραματικής διάταξης.

Παράλληλα, πραγματοποιήθηκαν τεχνητές μηχανικές μεταβολές και κραδασμοί στη διάταξη, ώστε να αξιολογηθεί η λειτουργία του αισθητήρα MPU6050. Παρατηρήθηκε μεταβολή των τιμών επιτάχυνσης στον άξονα Z (az), όπως ήταν αναμενόμενο, επιβεβαιώνοντας την ορθή λειτουργία του αισθητήρα και της μετάδοσης των δεδομένων προς την πλατφόρμα εποπτείας.

Η κατανάλωση ρεύματος του συστήματος μετρήθηκε μέσω του αισθητήρα ACS712 και συγκρίθηκε με τις ενδείξεις που εμφανίζονταν στο dashboard της πλατφόρμας. Οι ενδείξεις κρίθηκαν συνεπείς και συμβατές με την πραγματική λειτουργία του κινητήρα στις διαφορετικές συνθήκες PWM.



Επιπλέον, ο επαγωγικός αισθητήρας εγγύτητας τύπου LJ8A3-2-Z/AX λειτούργησε αξιόπιστα κατά την ανίχνευση του μεταλλικού αντικειμένου που ήταν προσαρμοσμένο στον άξονα περιστροφής. Κατά τη διάρκεια των δοκιμών δεν παρατηρήθηκε απώλεια γεγονότων ανίχνευσης και όλες οι μεταβολές καταγράφονταν σωστά στον μετρητή της πλατφόρμας ThingsBoard.

Τέλος, όλα τα alarms που είχαν υλοποιηθεί στην πλατφόρμα ενεργοποιούνταν όταν οι αντίστοιχες μετρήσεις υπερέβαιναν τα προκαθορισμένα όρια λειτουργίας. Αντίστοιχα, τα alarms απενεργοποιούνταν (“cleared”) αυτόματα όταν οι μετρήσεις επανέρχονταν εντός των προγραμματισμένων ορίων, επιβεβαιώνοντας την αναμενόμενη λειτουργία του μηχανισμού εποπτείας και ειδοποιήσεων του συστήματος.

5.6.2 Συμπεριφορά Συστήματος Ελέγχου

Η αξιολόγηση της συμπεριφοράς του συστήματος πραγματοποιήθηκε τόσο σε ανοικτό όσο και σε κλειστό βρόχο ελέγχου, με στόχο την παρατήρηση της απόκρισης του κινητήρα υπό διαφορετικές συνθήκες λειτουργίας.

Αρχικά, εξετάστηκε η λειτουργία σε ανοικτό βρόχο ελέγχου, όπου η ταχύτητα περιστροφής εξαρτάται άμεσα από το duty cycle του σήματος PWM. Παρατηρήθηκε ότι η αύξηση του duty cycle οδηγεί σε αντίστοιχη αύξηση της ταχύτητας του κινητήρα, επιβεβαιώνοντας τη βασική λειτουργία του συστήματος οδήγησης και της διάταξης PWM.

Στη συνέχεια, υλοποιήθηκε κλειστός βρόχος ελέγχου ταχύτητας με χρήση αρχικά αναλογικού ελεγκτή (P controller). Κατά τη διάρκεια των δοκιμών παρατηρήθηκε ότι, παρότι το σύστημα μπορούσε να προσεγγίσει την επιθυμητή τιμή RPM, η απόκριση παρουσίαζε σχετικά μεγάλο χρόνο σταθεροποίησης και μόνιμη απόκλιση από το setpoint σε ορισμένες συνθήκες λειτουργίας.

Για τη βελτίωση της συμπεριφοράς του συστήματος, ενσωματώθηκε ολοκληρωτικός όρος και υλοποιήθηκε PI ελεγκτής (Proportional–Integral Controller). Με τη χρήση του PI ελέγχου παρατηρήθηκε ταχύτερη προσέγγιση της επιθυμητής τιμής RPM και βελτιωμένη σταθερότητα της λειτουργίας, ιδιαίτερα κατά τις μεταβολές φορτίου.

Η συμπεριφορά του κλειστού βρόχου αξιολογήθηκε επίσης υπό αυξημένο μηχανικό φορτίο, το οποίο εφαρμόστηκε πειραματικά μέσω μηχανικής αντίστασης στον άξονα του κινητήρα κατά τη διάρκεια λειτουργίας. Στις περιπτώσεις αυτές παρατηρήθηκε στιγμιαία μείωση των στροφών, ενώ ο PI ελεγκτής προσαρμοζόταν δυναμικά



μεταβάλλοντας το duty cycle PWM, επιτυγχάνοντας επαναφορά της ταχύτητας κοντά στην επιθυμητή τιμή λειτουργίας.

Η εκκίνηση και η παύση του κινητήρα πραγματοποιούνταν ομαλά, χωρίς έντονες ταλαντώσεις ή ασταθή συμπεριφορά, ενώ η συνολική λειτουργία του συστήματος παρέμεινε σταθερή σε επαναλαμβανόμενους κύκλους λειτουργίας και μεταβολές συνθηκών.

Τα πειραματικά αποτελέσματα επιβεβαιώνουν ότι η χρήση ανάδρασης μέσω encoder και PI ελέγχου βελτιώνει σημαντικά τη συμπεριφορά του συστήματος σε σχέση με τη λειτουργία ανοικτού βρόχου, ιδιαίτερα σε περιπτώσεις μεταβολών φορτίου και διαταραχών λειτουργίας.

5.6.3 Αξιολόγηση Μετρήσεων Αισθητήρων

Οι μετρήσεις του αισθητήρα ρεύματος παρουσίασαν συνεπή συμπεριφορά και συσχετίζονται άμεσα με τη λειτουργία του κινητήρα. Παρατηρήθηκε αύξηση του ρεύματος κατά την εκκίνηση και σε υψηλότερες τιμές duty cycle, γεγονός που συμφωνεί με την αναμενόμενη λειτουργία του κινητήρα και του συστήματος οδήγησης.

Οι μετρήσεις του encoder επέτρεψαν αξιόπιστη εκτίμηση της ταχύτητας περιστροφής (RPM), μέσω καταμέτρησης παλμών με χρήση εξωτερικών διακοπών (EXTI interrupts). Η interrupt-based προσέγγιση συνέβαλε στη σταθερή λειτουργία του μηχανισμού μέτρησης ακόμη και σε αυξημένες ταχύτητες περιστροφής.

Αντίθετα, ο αισθητήρας θερμοκρασίας DS18B20 παρουσίασε περιστασιακές αποκλίσεις, όπως μη ρεαλιστικές τιμές για μικρό χρονικό διάστημα ή προσωρινή απώλεια επικοινωνίας. Το φαινόμενο αυτό πιθανώς σχετίζεται με τις χρονικές απαιτήσεις του πρωτοκόλλου 1-Wire και τη γενικότερη χρονική συμπεριφορά της πειραματικής διάταξης κατά την ταυτόχρονη εκτέλεση πολλαπλών λειτουργιών. Για τη βελτίωση της αξιοπιστίας των μετρήσεων θερμοκρασίας, εφαρμόστηκε στο firmware ένας απλός μηχανισμός φιλτραρίσματος και επικύρωσης τιμών. Συγκεκριμένα, κάθε νέα μέτρηση από τον αισθητήρα DS18B20 ελεγχόταν ως προς το αν βρίσκεται εντός ρεαλιστικών ορίων λειτουργίας πριν χρησιμοποιηθεί από το σύστημα. Σε περίπτωση μη έγκυρης ή στιγμιαία αλλοιωμένης μέτρησης, το σύστημα διατηρούσε την τελευταία έγκυρη τιμή θερμοκρασίας αντί να εμφανίσει λανθασμένα δεδομένα. Με την προσέγγιση αυτή επιτεύχθηκε σημαντική σταθεροποίηση της



τηλεμετρίας θερμοκρασίας κατά τη λειτουργία του κινητήρα και την ταυτόχρονη εκτέλεση πολλαπλών διεργασιών.

Οι μετρήσεις από τον αισθητήρα επιτάχυνσης MPU6050 παρουσίασαν σταθερή συμπεριφορά και επέτρεψαν την καταγραφή δυναμικών μεταβολών και κραδασμών του συστήματος, χωρίς να παρατηρηθούν εμφανή προβλήματα επικοινωνίας μέσω του διαύλου I2C.

Ο αισθητήρας εγγύτητας λειτούργησε αξιόπιστα κατά την ανίχνευση του μεταλλικού αντικειμένου που ήταν προσαρμοσμένο στον άξονα του κινητήρα. Η λειτουργία του παρέμεινε σταθερή, με την προϋπόθεση ότι διατηρούνταν κατάλληλη και σταθερή μηχανική ευθυγράμμιση μεταξύ αισθητήρα και μεταλλικού στόχου.

Επιπλέον, κατά τη διάρκεια των δοκιμών παρατηρήθηκε ότι η ένδειξη της τροφοδοσίας ακολουθούσε κανονικά τις μεταβολές τάσης του κυκλώματος κατά την ενεργοποίηση και απενεργοποίηση του μικροαυτόματου διακόπτη (MCB) του κινητήρα, επιβεβαιώνοντας τη σταθερή λειτουργία του κυκλώματος τροφοδοσίας της πειραματικής διάταξης, επιβλέποντας τη μέσω του διαιρέτη τάσης.

5.6.4 Αξιολόγηση Επικοινωνίας και IoT Λειτουργιών

Η σειριακή επικοινωνία μεταξύ STM32 και ESP32 παρουσίασε σταθερή συμπεριφορά στη συντριπτική πλειονότητα των περιπτώσεων. Η χρήση προκαθορισμένης και δομημένης μορφής δεδομένων συνέβαλε στη σωστή ερμηνεία των μηνυμάτων από τη μονάδα ESP32.

Σε περιορισμένο αριθμό περιπτώσεων παρατηρήθηκε αλλοίωση δεδομένων στη σειριακή επικοινωνία, η οποία αντιμετωπίστηκε μέσω βασικού φιλτραρίσματος και ελέγχου εγκυρότητας στο επίπεδο του ESP32.

Η μετάδοση δεδομένων προς την πλατφόρμα ThingsBoard μέσω του πρωτοκόλλου MQTT λειτούργησε ομαλά, επιτρέποντας την απεικόνιση των βασικών μεγεθών σχεδόν σε πραγματικό χρόνο μέσω του dashboard. Η καθυστέρηση μετάδοσης παρέμεινε μικρή και δεν επηρέασε τη λειτουργία του συστήματος, καθώς οι λειτουργίες ελέγχου υλοποιούνται τοπικά στον STM32.

Η λειτουργία απομακρυσμένου ελέγχου μέσω RPC αξιολογήθηκε επιτυχώς κατά τη διάρκεια των δοκιμών. Οι εντολές που αποστέλλονταν από την πλατφόρμα μεταφέρονταν μέσω MQTT προς την ESP32, μετατρέπονταν σε σειριακές εντολές UART και προωθούνταν στον STM32 για επεξεργασία και εκτέλεση.



Παρατηρήθηκε συνεπής λειτουργία της αλυσίδας επικοινωνίας, ενώ ο κινητήρας ανταποκρινόταν ομαλά στις εντολές ενεργοποίησης, απενεργοποίησης και μεταβολής της επιθυμητής ταχύτητας περιστροφής.

Η αρχιτεκτονική αυτή επιβεβαιώνει τον σαφή διαχωρισμό μεταξύ επιπέδου ελέγχου και επιπέδου επικοινωνίας και εποπτείας, επιτρέποντας την υλοποίηση IoT λειτουργιών χωρίς να επηρεάζεται η χρονικά κρίσιμη λειτουργία του βρόχου ελέγχου.

5.6.5 Συνολική Αξιολόγηση Συστήματος

Συνολικά, το σύστημα παρουσιάζει σταθερή και αξιόπιστη λειτουργία σε όλα τα βασικά επίπεδα:

- έλεγχος κινητήρα,
- συλλογή και μετάδοση δεδομένων,
- επικοινωνία μεταξύ STM32 και ESP32,
- απομακρυσμένη εποπτεία και βασικός απομακρυσμένος έλεγχος μέσω της πλατφόρμας ThingsBoard.

Η αρχιτεκτονική με διαχωρισμό επιπέδων (control layer – communication layer) αποδείχθηκε αποτελεσματική, καθώς επιτρέπει την απομόνωση των χρονικά κρίσιμων λειτουργιών ελέγχου από πιθανές καθυστερήσεις ή αστάθειες του δικτύου και της IoT πλατφόρμας.

Παράλληλα, η ενσωμάτωση της πλατφόρμας ThingsBoard προσφέρει σημαντικές δυνατότητες επέκτασης του συστήματος, όπως:

- απομακρυσμένη εποπτεία και διάγνωση,
- καταγραφή ιστορικών δεδομένων,
- υλοποίηση μηχανισμών alarms και ειδοποιήσεων,
- καθώς και δυνατότητα μελλοντικής ενσωμάτωσης πιο προηγμένων αλγορίθμων ελέγχου και ανάλυσης δεδομένων.

Η πειραματική υλοποίηση επιβεβαιώνει ότι είναι δυνατή η ανάπτυξη ενός ενσωματωμένου συστήματος ελέγχου κινητήρα με δυνατότητες IoT, τηλεμετρίας και απομακρυσμένης εποπτείας, διατηρώντας παράλληλα προβλέψιμη και σταθερή λειτουργία στο επίπεδο ελέγχου πραγματικού χρόνου.



5.7 Εναλλακτική Υλοποίηση με FreeRTOS

Στο πλαίσιο της εργασίας αναπτύχθηκε και εναλλακτική αρχιτεκτονική firmware με χρήση του λειτουργικού συστήματος πραγματικού χρόνου FreeRTOS, με στόχο τη μελέτη της οργάνωσης των λειτουργιών του συστήματος σε διακριτές εργασίες (tasks) και τη διερεύνηση μηχανισμών χρονοπρογραμματισμού πραγματικού χρόνου σε ενσωματωμένα συστήματα ελέγχου.

Η υλοποίηση βασίστηκε στην ίδια πειραματική διάταξη υλικού και αξιοποίησε το ίδιο υποσύστημα αισθητήρων, τον ίδιο βρόχο ελέγχου PI, καθώς και την ίδια αρχιτεκτονική επικοινωνίας STM32–ESP32–ThingsBoard. Στόχος ήταν η διατήρηση της ίδιας λειτουργικής συμπεριφοράς του συστήματος, αλλά με διαφορετική εσωτερική οργάνωση του firmware.

Στην αρχιτεκτονική FreeRTOS, οι βασικές λειτουργίες του συστήματος διαχωρίστηκαν σε ανεξάρτητες εργασίες:

- εργασία ελέγχου κινητήρα (ControlTask),
- εργασία ανάγνωσης αισθητήρων (SensorTask),
- εργασία τηλεμετρίας και επικοινωνίας UART (TelemetryTask),
- και εργασία επιτήρησης/ασφάλειας (SafetyTask).

Η παρακάτω ενδεικτική δομή firmware παρουσιάζει τον τρόπο με τον οποίο οι βασικές λειτουργίες μπορούν να οργανωθούν σε διακριτές εργασίες στο περιβάλλον FreeRTOS. Η ControlTask διατηρεί την υψηλότερη προτεραιότητα, ενώ οι λειτουργίες αισθητήρων, τηλεμετρίας και επιτήρησης εκτελούνται ανεξάρτητα με χαμηλότερες προτεραιότητες.

Τμήμα Firmware (8). Ενδεικτική οργάνωση εργασιών σε περιβάλλον FreeRTOS

```
xTaskCreate(ControlTask, "Control", 256, NULL, 4, NULL);  
xTaskCreate(SensorTask, "Sensors", 256, NULL, 3, NULL);  
xTaskCreate(TelemetryTask, "Telemetry", 256, NULL, 2, NULL);  
xTaskCreate(SafetyTask, "Safety", 256, NULL, 3, NULL);  
vTaskStartScheduler();
```

Ο βρόχος ελέγχου RPM υλοποιήθηκε ως εργασία υψηλής προτεραιότητας, η οποία εκτελεί περιοδικά:

- υπολογισμό RPM από τα encoder interrupts,
- υπολογισμό του PI controller,



- και ενημέρωση του PWM duty cycle του κινητήρα.

Παράλληλα, οι λειτουργίες ανάγνωσης αισθητήρων (ρεύμα, τάση, θερμοκρασία, MPU6050 και proximity sensor) μεταφέρθηκαν σε ξεχωριστή εργασία αισθητήρων, ενώ η δημιουργία και αποστολή του UART telemetry frame προς το ESP32 πραγματοποιήθηκε μέσω ανεξάρτητης εργασίας τηλεμετρίας.

Τα hardware interrupts παρέμειναν ενεργά και στην RTOS αρχιτεκτονική, ιδιαίτερα για:

- τα encoder EXTI interrupts,
- και τη λήψη δεδομένων UART,

ώστε τα χρονικά κρίσιμα γεγονότα να συνεχίσουν να εξυπηρετούνται άμεσα από το hardware επίπεδο.

Στην υλοποίηση χρησιμοποιήθηκαν επίσης software buffers και μηχανισμοί περιοδικού scheduling του FreeRTOS, ενώ διατηρήθηκε η ίδια μορφή telemetry frame ώστε το ESP32 και το ThingsBoard dashboard να λειτουργούν χωρίς τροποποιήσεις. Επιπλέον, ενσωματώθηκαν βασικοί μηχανισμοί αξιοπιστίας και προστασίας, που υπάρχουν και στη bare metal αρχιτεκτονική, όπως:

- Independent Watchdog (IWDG),
- έλεγχος υπερέντασης,
- έλεγχος υπέρτασης/υπότασης,
- και μηχανισμός ελέγχου εγκυρότητας μετρήσεων θερμοκρασίας.

Διερευνήθηκε επίσης η δυνατότητα χρήσης απλού checksum/CRC μηχανισμού για έλεγχο ακεραιότητας των telemetry frames μεταξύ STM32 και ESP32.

Η χρήση του FreeRTOS επέτρεψε σαφέστερο διαχωρισμό λειτουργιών και καλύτερη επεκτασιμότητα του firmware, ιδιαίτερα σε περιπτώσεις αύξησης των αισθητήρων, των λειτουργιών επικοινωνίας ή των απαιτήσεων πραγματικού χρόνου.

Ωστόσο, η κύρια πειραματική αξιολόγηση του συστήματος πραγματοποιήθηκε με bare-metal υλοποίηση, καθώς η προσέγγιση αυτή κρίθηκε επαρκής για τις απαιτήσεις της συγκεκριμένης πειραματικής διάταξης και επέτρεψε απλούστερη ανάλυση και αποσφαλμάτωση της χρονικής συμπεριφοράς του βρόχου ελέγχου.



Η εναλλακτική αρχιτεκτονική FreeRTOS επιβεβαίωσε ότι το σύστημα μπορεί να επεκταθεί σε πιο σύνθετες εφαρμογές πραγματικού χρόνου, διατηρώντας παράλληλα τον διαχωρισμό μεταξύ επιπέδου ελέγχου και επιπέδου επικοινωνίας.

5.8 Συμπεράσματα Κεφαλαίου

Στο παρόν κεφάλαιο παρουσιάστηκε η πειραματική διάταξη και η διαδικασία αξιολόγησης του προτεινόμενου συστήματος. Μέσω των δοκιμών αξιολογήθηκε η λειτουργία των επιμέρους υποσυστημάτων και η επιτυχής ενσωμάτωσή τους σε μία ενιαία αρχιτεκτονική ενσωματωμένου ελέγχου και IoT εποπτείας.

Η πειραματική υλοποίηση επιβεβαιώνει τη δυνατότητα ανάπτυξης ενός ενσωματωμένου συστήματος ελέγχου κινητήρα με δυνατότητες IoT, το οποίο συνδυάζει:

- πραγματικού χρόνου έλεγχο,
- πολυαισθητηριακή παρακολούθηση,
- αξιόπιστη επικοινωνία μεταξύ υποσυστημάτων,
- απομακρυσμένη εποπτεία και βασικό απομακρυσμένο έλεγχο.

Τα αποτελέσματα της αξιολόγησης ανέδειξαν τόσο τα πλεονεκτήματα της προτεινόμενης αρχιτεκτονικής όσο και ορισμένους πρακτικούς περιορισμούς της πειραματικής διάταξης, οι οποίοι μπορούν να αποτελέσουν αντικείμενο μελλοντικής βελτίωσης και περαιτέρω έρευνας.

Ιδιαίτερα σημαντικός αποδείχθηκε ο σαφής διαχωρισμός μεταξύ επιπέδου ελέγχου πραγματικού χρόνου (STM32) και επιπέδου επικοινωνίας και IoT λειτουργιών (ESP32 και ThingsBoard), καθώς επέτρεψε τη διατήρηση προβλέψιμης χρονικής συμπεριφοράς στον βρόχο ελέγχου, ανεξάρτητα από τη δικτυακή δραστηριότητα και τις λειτουργίες cloud.

Παράλληλα, η εναλλακτική αρχιτεκτονική firmware με χρήση FreeRTOS ανέδειξε τα πλεονεκτήματα της οργανωμένης διαχείρισης λειτουργιών μέσω διακριτών εργασιών (tasks), καθώς και τη δυνατότητα μελλοντικής επέκτασης του συστήματος σε πιο σύνθετες εφαρμογές πραγματικού χρόνου. Η χρήση RTOS διευκολύνει τον διαχωρισμό κρίσιμων και μη κρίσιμων λειτουργιών, τη διαχείριση επικοινωνίας και τηλεμετρίας και τη βελτίωση της επεκτασιμότητας του firmware.



Συνολικά, τα αποτελέσματα επιβεβαιώνουν τη λειτουργική επάρκεια της προτεινόμενης αρχιτεκτονικής και την καταλληλότητά της για εφαρμογές ενσωματωμένου ελέγχου, παρακολούθησης και απομακρυσμένης εποπτείας σε πραγματικό χρόνο.



6 Συμπεράσματα και Μελλοντική Έρευνα

6.1 Συμπεράσματα

Στο πλαίσιο της παρούσας διπλωματικής εργασίας σχεδιάστηκε και υλοποιήθηκε ένα ενσωματωμένο σύστημα για τον έλεγχο και την παρακολούθηση ηλεκτρικού κινητήρα, το οποίο συνδυάζει λειτουργίες πραγματικού χρόνου με δυνατότητες απομακρυσμένης εποπτείας μέσω τεχνολογιών IoT.

Η αρχιτεκτονική του συστήματος βασίζεται στον σαφή διαχωρισμό μεταξύ επιπέδου ελέγχου και επιπέδου επικοινωνίας και εποπτείας. Συγκεκριμένα, ο μικροελεγκτής STM32 αναλαμβάνει τις χρονικά κρίσιμες λειτουργίες του βρόχου ελέγχου, ενώ η μονάδα ESP32 λειτουργεί ως πύλη επικοινωνίας προς την πλατφόρμα IoT. Ο διαχωρισμός αυτός επιβεβαιώθηκε πειραματικά ως ιδιαίτερα αποτελεσματικός, καθώς επιτρέπει τη διατήρηση προβλέψιμης χρονικής συμπεριφοράς ανεξάρτητα από πιθανές καθυστερήσεις ή αστάθειες του δικτύου.

Η πειραματική αξιολόγηση έδειξε ότι το σύστημα παρουσιάζει σταθερή και ικανοποιητική λειτουργία σε όλα τα βασικά επίπεδα της διάταξης. Ο έλεγχος του κινητήρα μέσω PWM πραγματοποιείται ομαλά, ενώ η υλοποίηση κλειστού βρόχου ελέγχου με χρήση PI ελεγκτή βελτίωσε σημαντικά τη συμπεριφορά του συστήματος κατά τις μεταβολές φορτίου και τη διατήρηση της επιθυμητής ταχύτητας περιστροφής. Παράλληλα, οι αισθητήρες του συστήματος παρείχαν χρήσιμα δεδομένα για την παρακολούθηση της λειτουργίας του κινητήρα και της συνολικής κατάστασης της διάταξης. Η χρήση εξωτερικών διακοπών (interrupts) για την καταμέτρηση παλμών του encoder επέτρεψε αξιόπιστη εκτίμηση της ταχύτητας περιστροφής, ενώ οι υπόλοιποι αισθητήρες συνέβαλαν στην παρακολούθηση θερμοκρασίας, κατανάλωσης ρεύματος, κραδασμών και γεγονότων εγγύτητας.

Η ενσωμάτωση της πλατφόρμας ThingsBoard επέτρεψε την επιτυχή υλοποίηση λειτουργιών απομακρυσμένης εποπτείας, τηλεμετρίας και βασικού απομακρυσμένου ελέγχου. Τα δεδομένα μεταδίδονται μέσω MQTT και απεικονίζονται μέσω διαδραστικών dashboards, ενώ υλοποιήθηκαν επίσης μηχανισμοί alarms και RPC εντολών για βασικές λειτουργίες διαχείρισης του συστήματος.

Επιπλέον, στο πλαίσιο της εργασίας αναπτύχθηκε και εναλλακτική αρχιτεκτονική firmware με χρήση FreeRTOS, με στόχο τη μελέτη της οργάνωσης των λειτουργιών



του συστήματος σε διακριτές εργασίες πραγματικού χρόνου. Η προσέγγιση αυτή ανέδειξε τα πλεονεκτήματα της χρήσης RTOS σε εφαρμογές ενσωματωμένων συστημάτων, ιδιαίτερα ως προς τον διαχωρισμό λειτουργιών, τη διαχείριση προτεραιοτήτων και τη μελλοντική επεκτασιμότητα του firmware.

Συνολικά, η εργασία επιβεβαιώνει τη δυνατότητα ανάπτυξης ενός ολοκληρωμένου ενσωματωμένου συστήματος ελέγχου και παρακολούθησης ηλεκτρικού κινητήρα, το οποίο συνδυάζει επεξεργασία πραγματικού χρόνου, τηλεμετρία, IoT λειτουργίες και απομακρυσμένη εποπτεία. Η προτεινόμενη αρχιτεκτονική κρίνεται κατάλληλη για εφαρμογές ενσωματωμένων και βιομηχανικών συστημάτων που απαιτούν αξιόπιστο έλεγχο, συλλογή δεδομένων και δυνατότητες δικτυακής διασύνδεσης.

6.2 Περιορισμοί Συστήματος

Παρά την επιτυχή υλοποίηση και τη συνολικά σταθερή λειτουργία του προτεινόμενου συστήματος, κατά την πειραματική αξιολόγηση εντοπίστηκαν ορισμένοι περιορισμοί, οι οποίοι σχετίζονται τόσο με τη φύση της πειραματικής διάταξης μικρής κλίμακας όσο και με τις τεχνολογικές επιλογές που υιοθετήθηκαν.

Ένας βασικός περιορισμός αφορά την πειραματική φύση του συστήματος και τη χρήση αισθητήρων χαμηλού κόστους, οι οποίοι, παρότι επαρκείς για την αξιολόγηση της λειτουργικότητας της αρχιτεκτονικής, δεν προσφέρουν την ακρίβεια και την ανθεκτικότητα εξειδικευμένου βιομηχανικού εξοπλισμού. Το γεγονός αυτό είναι περισσότερο εμφανές σε συνθήκες ηλεκτρομαγνητικού θορύβου ή κατά τις δυναμικές μεταβολές λειτουργίας του κινητήρα.

Επιπλέον, ο αισθητήρας θερμοκρασίας DS18B20 παρουσίασε περιστασιακές αποκλίσεις, όπως μη ρεαλιστικές τιμές ή προσωρινή απώλεια επικοινωνίας. Το φαινόμενο αυτό πιθανώς σχετίζεται με τις χρονικές απαιτήσεις του πρωτοκόλλου 1-Wire και τη συνύπαρξη πολλαπλών χρονικά κρίσιμων λειτουργιών στο σύστημα.

Σε επίπεδο επικοινωνίας, παρατηρήθηκαν περιορισμένες περιπτώσεις αλλοίωσης δεδομένων στη σειριακή διεπαφή UART μεταξύ STM32 και ESP32. Αν και το πρόβλημα αντιμετωπίστηκε σε σημαντικό βαθμό μέσω μηχανισμών ελέγχου εγκυρότητας, buffering και φιλτραρίσματος στο επίπεδο της ESP32, αναδεικνύεται η ανάγκη χρήσης πιο ισχυρών μηχανισμών ελέγχου ακεραιότητας και διαχείρισης επικοινωνίας σε μελλοντικές υλοποιήσεις.



Παράλληλα, η πειραματική διάταξη δεν περιλαμβάνει πλήρη βιομηχανική υποδομή, όπως προηγμένα κυκλώματα προστασίας, φίλτρα ηλεκτρομαγνητικού θορύβου, βιομηχανικές διεπαφές επικοινωνίας ή εξειδικευμένα στάδια ισχύος. Ως αποτέλεσμα, η παρούσα υλοποίηση δεν προορίζεται για άμεση χρήση σε απαιτητικά βιομηχανικά περιβάλλοντα χωρίς περαιτέρω βελτιώσεις και πιστοποίηση του συστήματος.

Αν και στην τελική πειραματική αξιολόγηση χρησιμοποιήθηκε κυρίως bare-metal αρχιτεκτονική, η ανάπτυξη εναλλακτικής υλοποίησης με FreeRTOS ανέδειξε ορισμένα σημαντικά πλεονεκτήματα αλλά και πρόσθετες απαιτήσεις. Η χρήση RTOS προσφέρει καλύτερο διαχωρισμό λειτουργιών, οργανωμένη διαχείριση εργασιών πραγματικού χρόνου και αυξημένη επεκτασιμότητα του firmware, ιδιαίτερα σε εφαρμογές με μεγάλο αριθμό αισθητήρων και λειτουργιών επικοινωνίας. Παράλληλα όμως, εισάγει αυξημένη πολυπλοκότητα στον σχεδιασμό, ανάγκη προσεκτικής διαχείρισης συγχρονισμού και μεγαλύτερες απαιτήσεις ως προς την ανάλυση της χρονικής συμπεριφοράς του συστήματος.

Κατά την πειραματική διερεύνηση της RTOS αρχιτεκτονικής διαπιστώθηκε ότι η σωστή ιεράρχηση προτεραιοτήτων και η αποφυγή blocking λειτουργιών αποτελούν κρίσιμους παράγοντες για τη διατήρηση σταθερής χρονικής συμπεριφοράς στις μετρήσεις και στον βρόχο ελέγχου.

Οι παραπάνω περιορισμοί θεωρούνται αναμενόμενοι για ένα σύστημα πειραματικής κλίμακας και δεν αναιρούν τη λειτουργική επάρκεια της προτεινόμενης αρχιτεκτονικής. Αντίθετα, αναδεικνύουν συγκεκριμένες τεχνικές κατευθύνσεις για τη μελλοντική βελτίωση και εξέλιξη του συστήματος.

6.3 Μελλοντική Έρευνα και Επεκτάσεις

Η παρούσα εργασία αποτελεί μία ολοκληρωμένη βάση για την ανάπτυξη πιο σύνθετων ενσωματωμένων συστημάτων ελέγχου και παρακολούθησης ηλεκτρικών κινητήρων. Παρότι το προτεινόμενο σύστημα παρουσίασε σταθερή και λειτουργικά επαρκή συμπεριφορά, υπάρχουν σημαντικές δυνατότητες μελλοντικής επέκτασης τόσο σε επίπεδο ελέγχου όσο και σε επίπεδο αισθητήρων, επικοινωνίας και επεξεργασίας δεδομένων.

Σε επίπεδο ελέγχου, η περαιτέρω βελτιστοποίηση του ήδη υλοποιημένου κλειστού βρόχου PI ελέγχου αποτελεί σημαντική κατεύθυνση εξέλιξης. Η εφαρμογή πλήρους PID ελέγχου, καθώς και πιο προηγμένων τεχνικών όπως προσαρμοστικός έλεγχος



(adaptive control), fuzzy control ή αλγόριθμοι βελτιστοποίησης, θα μπορούσε να βελτιώσει περαιτέρω τη δυναμική συμπεριφορά του κινητήρα υπό μεταβαλλόμενες συνθήκες φορτίου και λειτουργίας.

Παράλληλα, ιδιαίτερο ενδιαφέρον παρουσιάζει η αξιοποίηση τεχνικών τεχνητής νοημοσύνης και μηχανικής μάθησης (Artificial Intelligence / Machine Learning) για εφαρμογές προγνωστικής συντήρησης (predictive maintenance). Μέσω της συλλογής και ανάλυσης ιστορικών δεδομένων λειτουργίας, όπως θερμοκρασία, κατανάλωση ρεύματος, κραδασμοί και μεταβολές ταχύτητας, θα μπορούσε να επιτευχθεί έγκαιρη ανίχνευση ανωμαλιών και πρόβλεψη πιθανών βλαβών του κινητήρα ή της μηχανικής διάταξης.

Στο πλαίσιο αυτό, μία σημαντική μελλοντική επέκταση θα ήταν η ενσωμάτωση τοπικής υπολογιστικής μονάδας υψηλότερων δυνατοτήτων, όπως βιομηχανικού υπολογιστή ή embedded Linux platform (π.χ. Raspberry Pi ή Industrial PC), η οποία θα μπορούσε να χρησιμοποιηθεί για:

- τοπική αποθήκευση ιστορικών δεδομένων,
- προηγμένη ανάλυση μετρήσεων,
- υλοποίηση αλγορίθμων AI/ML,
- και λειτουργίες edge computing χωρίς εξάρτηση από το cloud.

Σε επίπεδο αισθητήρων, η χρήση βιομηχανικών αισθητήρων υψηλότερης ακρίβειας και ανθεκτικότητας θα βελτίωνε σημαντικά την ποιότητα και αξιοπιστία των μετρήσεων. Επιπλέον, η ενσωμάτωση πρόσθετων αισθητήρων, όπως αισθητήρες κραδασμών υψηλής ευαισθησίας, αισθητήρες θέσης ή αισθητήρες ενεργειακής κατανάλωσης, θα επέτρεπε πιο ολοκληρωμένη παρακολούθηση της κατάστασης του συστήματος.

Σημαντική κατεύθυνση εξέλιξης αποτελεί επίσης η περαιτέρω αξιοποίηση της αρχιτεκτονικής FreeRTOS που αναπτύχθηκε στο πλαίσιο της εργασίας. Η χρήση RTOS επιτρέπει καλύτερη οργάνωση των λειτουργιών σε διακριτές εργασίες πραγματικού χρόνου, βελτιωμένη διαχείριση προτεραιοτήτων, ευκολότερη επεκτασιμότητα και πιο οργανωμένη διαχείριση επικοινωνίας και τηλεμετρίας. Σε πιο σύνθετες εφαρμογές με μεγάλο αριθμό αισθητήρων ή πολλαπλά υποσυστήματα επικοινωνίας, η χρήση RTOS μπορεί να συμβάλει σημαντικά στη διατήρηση



προβλέψιμης χρονικής συμπεριφοράς και στη βελτίωση της συντηρησιμότητας του firmware.

Σε επίπεδο επικοινωνίας, η υποστήριξη βιομηχανικών πρωτοκόλλων όπως CAN, Modbus ή RS485 θα επέτρεπε την ενσωμάτωση του συστήματος σε πιο απαιτητικά βιομηχανικά περιβάλλοντα και σε δίκτυα βιομηχανικού αυτοματισμού. Παράλληλα, η εφαρμογή ισχυρότερων μηχανισμών ελέγχου ακεραιότητας δεδομένων και fault-tolerant επικοινωνίας θα μπορούσε να βελτιώσει περαιτέρω την αξιοπιστία του συστήματος.

Τέλος, ιδιαίτερο ενδιαφέρον παρουσιάζει η μετάβαση από την πειραματική διάταξη μικρής κλίμακας σε πλήρως βιομηχανική υλοποίηση. Η ενσωμάτωση πιο προηγμένων μηχανισμών προστασίας, φίλτρων ηλεκτρομαγνητικού θορύβου, βιομηχανικών σταδίων ισχύος και πιστοποιημένων διατάξεων ασφαλείας θα επιτρέψει τη χρήση της προτεινόμενης αρχιτεκτονικής σε πραγματικές εφαρμογές βιομηχανικού ελέγχου και παρακολούθησης.

Οι παραπάνω κατευθύνσεις αναδεικνύουν ότι το προτεινόμενο σύστημα δεν περιορίζεται σε μία απλή εκπαιδευτική ή πειραματική υλοποίηση, αλλά μπορεί να αποτελέσει τη βάση για την ανάπτυξη σύγχρονων, έξυπνων και επεκτάσιμων ενσωματωμένων συστημάτων ελέγχου και εποπτείας σε βιομηχανικό περιβάλλον.

6.4 Τελική Αποτίμηση

Η παρούσα διπλωματική εργασία οδήγησε στον σχεδιασμό και την υλοποίηση ενός ενσωματωμένου συστήματος ελέγχου και παρακολούθησης ηλεκτρικού κινητήρα με δυνατότητες IoT, συνδυάζοντας τεχνολογίες πραγματικού χρόνου, αισθητήρες, τηλεμετρία και απομακρυσμένη εποπτεία σε μία ενιαία αρχιτεκτονική.

Μέσω της πειραματικής αξιολόγησης επιβεβαιώθηκε η λειτουργική επάρκεια του συστήματος, καθώς και η δυνατότητα αξιοπιστής συνεργασίας μεταξύ του επιπέδου ελέγχου πραγματικού χρόνου και του επιπέδου επικοινωνίας και cloud εποπτείας. Η χρήση του μικροελεγκτή STM32 για τις χρονικά κρίσιμες λειτουργίες και της μονάδας ESP32 για τη διαχείριση της επικοινωνίας και των IoT υπηρεσιών επέτρεψε την ομαλή λειτουργία του συστήματος, διατηρώντας παράλληλα σαφή αρχιτεκτονικό διαχωρισμό μεταξύ των επιμέρους επιπέδων.

Η εργασία ανέδειξε επίσης τη σημασία της ενσωμάτωσης τεχνολογιών IoT σε σύγχρονα ενσωματωμένα συστήματα ελέγχου, καθώς μέσω της τηλεμετρίας, των



dashboards, των RPC λειτουργιών και των μηχανισμών alarms καθίσταται δυνατή η απομακρυσμένη εποπτεία, η συλλογή ιστορικών δεδομένων και η δημιουργία βάσης για πιο προηγμένες εφαρμογές διαγνωστικής και προγνωστικής συντήρησης.

Παράλληλα, η ανάπτυξη εναλλακτικής αρχιτεκτονικής firmware με χρήση FreeRTOS ανέδειξε τη δυνατότητα μελλοντικής επέκτασης του συστήματος σε πιο σύνθετες εφαρμογές πραγματικού χρόνου, με καλύτερη οργάνωση λειτουργιών, διαχείριση προτεραιοτήτων και αυξημένη επεκτασιμότητα του λογισμικού.

Συνολικά, το προτεινόμενο σύστημα αποτελεί μία ολοκληρωμένη και λειτουργική προσέγγιση στον σχεδιασμό σύγχρονων ενσωματωμένων συστημάτων ελέγχου και παρακολούθησης ηλεκτρικών κινητήρων. Η εργασία επιβεβαιώνει ότι ο συνδυασμός ενσωματωμένου ελέγχου πραγματικού χρόνου, αισθητήρων, τηλεμετρίας και τεχνολογιών IoT μπορεί να αξιοποιηθεί αποτελεσματικά για την ανάπτυξη ευέλικτων και επεκτάσιμων συστημάτων εποπτείας και ελέγχου σε σύγχρονα τεχνολογικά και βιομηχανικά περιβάλλοντα.



Αναφορές

- Banks, A., Gupta, R.: ‘MQTT Version 3.1.1’, OASIS Standard, 2014. ISO/IEC 20922:2016.
- Chapman, S.: ‘Electric Machinery Fundamentals’ (McGraw-Hill, New York, 2011).
- Hanes, D., Salgueiro, G., Grossetete, P., Barton, R., Henry, J.: ‘IoT Fundamentals: Networking Technologies, Protocols and Use Cases for Internet of Things’ (Cisco Press, Indianapolis, 2017).
- Krumm, J. (ed.): ‘Ubiquitous Computing Fundamentals’ (CRC Press, Boca Raton, 2010).
- Li, Y., Potkonjak, M., Wolf, W.: ‘Real-Time Operating Systems for Embedded Computing’, Proceedings of the International Conference on Computer Design (ICCD), 1997, σελ. 388–392.
- Madiseti, V., Bahga, A.: ‘Internet of Things: A Hands-On Approach’ (Universities Press, Hyderabad, 2014).
- Marwedel, P.: ‘Embedded System Design’ (Springer, Dordrecht, 2011).
- Mohammed, N. A., Abdulateef, O. F., Hamad, A. H.: ‘An IoT and Machine Learning-Based Predictive Maintenance System for Electrical Motors’, Journal of Electrical Systems and Automation (JESA), 56(4), (2023), σελ. 651–656.
- Mortazavizadeh, S. A., Mousavi, S. M. G.: ‘A Review on Condition Monitoring and Diagnostic Techniques of Rotating Electrical Machines’, Physical Science International Journal, 4(3), (2014), σελ. 310–338.
- Parpala, R. C., Iacob, R.: ‘Application of IoT Concept on Predictive Maintenance of Industrial Equipment’, MATEC Web of Conferences, 121, (2017), σελ. 02008.
- Samie, F., Bauer, L., Henkel, J.: ‘IoT Technologies for Embedded Computing: A Survey’, Proceedings of the International Conference on Hardware/Software Codesign and System Synthesis (CODES+ISSS), 2016.
- Vahid, F., Givargis, T.: ‘Embedded System Design: A Unified Hardware/Software Approach’ (University of California, Riverside, 1999).
- Weiser, M.: ‘The Computer for the 21st Century’, Scientific American, 265(3), (1991), σελ. 94–104.
- Wibowo, K. H. et al.: ‘IoT-based Control and Monitoring for DC Motor Fed by Photovoltaic System’, IOP Conference Series: Materials Science and Engineering, (2019).



Wolf, W.: ‘Computers as Components: Principles of Embedded Computing System Design’ (Morgan Kaufmann, San Francisco, 2012).

‘The FreeRTOS Reference Manual’, Amazon Web Services, FreeRTOS Version 10.0.0, 2017.

Παρασκευόπουλος, Π. Ν.: ‘Σύγχρονος Έλεγχος Συστημάτων’ (Εκδόσεις Συμμετρία, Αθήνα, 2008).

InvenSense Inc.: ‘MPU-6000 and MPU-6050 Product Specification Revision 3.4’, 2013.

Maxim Integrated: ‘DS18B20 Programmable Resolution 1-Wire Digital Thermometer Datasheet’, 2015.

Allegro MicroSystems: ‘ACS712 Fully Integrated, Hall-Effect-Based Linear Current Sensor IC Datasheet’, 2006.

STMicroelectronics: ‘STM32F446RE ARM Cortex-M4 32-bit MCU Datasheet’, 2023.

ThingsBoard Inc.: ‘ThingsBoard Open-Source IoT Platform’, <https://thingsboard.io>

Espressif Systems: ‘ESP32 Series Datasheet’, 2024.

STMicroelectronics: ‘STM32CubeIDE User Guide’, 2024.

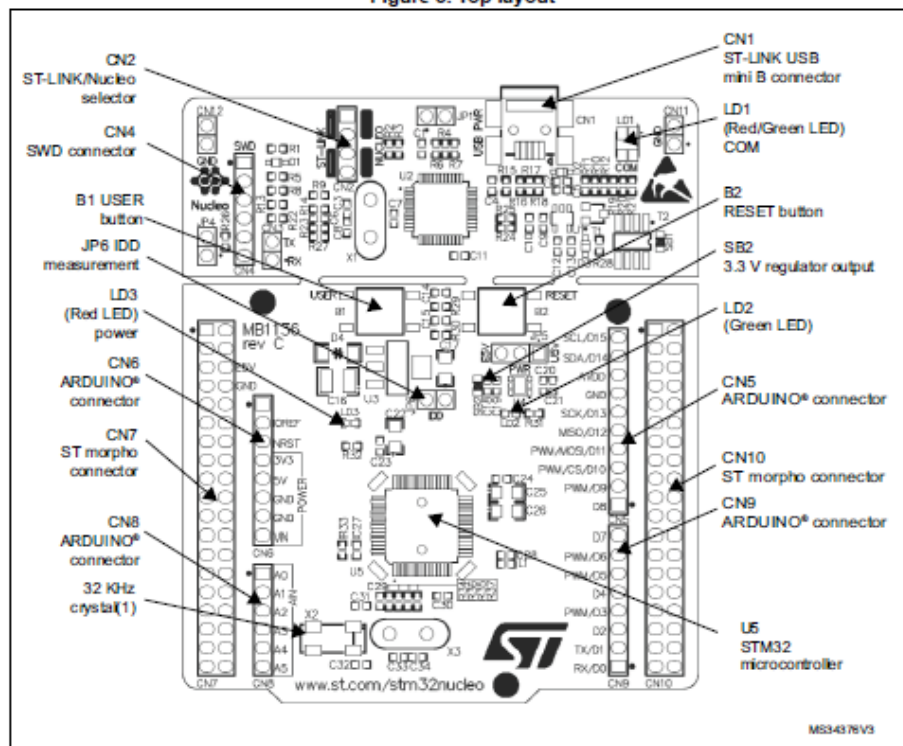
Παράρτημα

Στο Παράρτημα παρατίθενται ενδεικτικά τεχνικά φυλλάδια (datasheets) των βασικών εξαρτημάτων που χρησιμοποιήθηκαν στην πειραματική διάταξη, με στόχο την τεκμηρίωση των τεχνικών χαρακτηριστικών και των ηλεκτρικών προδιαγραφών του συστήματος.

Hardware layout and configuration

UM1724

Figure 3. Top layout



1. Crystal might be present or not depending on the board version. Refer to [Section 7.9.2](#).



STMicroelectronics NUCLEO-F446RE STM32 Nucleo-64 Development Board

STMicroelectronics NUCLEO-F446RE STM32 Nucleo-64 Development Board features the NUCLEO-F446RE MCU and supports Arduino™ and ST Morpho connectivity. The STMicroelectronics STM32 Nucleo-64 boards are affordable and flexible for trying out concepts and building prototypes. Choose from the various combinations of performance and power consumption features provided by the STM32 microcontroller. The SMPS significantly reduces power consumption in Run mode for the compatible boards. Arduino™ Uno V3 connectivity support and the ST Morpho headers allow the STM32 Nucleo open development platform to expand easily. A wide choice of specialized shields is available. The STM32 Nucleo-64 board does not require any separate probe as it integrates the ST-LINK/V2-1 debugger and programmer. The STM32 Nucleo-64 board comes with the STM32 comprehensive software HAL library and various packaged software examples. Also included is direct access to the Arm® Mbed™ online resources.

FEATURES

- STM32 microcontroller in LQFP64 package
- External SMPS to generate Vcore logic supply (only available on '-P' suffixed boards)
- 1 user LED shared with Arduino™
- 1 user and 1 reset push-buttons
- 32.768kHz LSE crystal oscillator
- Board expansion connectors
 - Arduino™ Uno V3
 - ST Morpho extension pin headers for full access to all STM32 I/Os
 - External SMPS experimentation dedicated connector (only available on '-P' suffixed boards)
- Flexible power-supply options: ST-LINK USB VBUS or external sources
- On-board ST-LINK/V2-1 debugger/programmer with USB re-enumeration capability. Three different interfaces are supported on USB: mass storage, Virtual COM port, and debug port
- Comprehensive free software libraries and examples available with the STM32Cube MCU Package
- Support of a wide choice of Integrated Development Environments (IDEs), including IAR™, Keil®, GCC-based IDEs, Arm Mbed
- Arm Mbed Enabled compliant (only for some Nucleo part numbers)



Description Specs (22) Pictures (16) Resou

Dimensions

Size:	37D × 72.5L mm ¹
Weight:	210 g
Shaft diameter:	6 mm ²

General specifications

Gear ratio:	102.08:1
No-load speed @ 12V:	100 rpm
No-load current @ 12V:	0.2 A
Stall current @ 12V:	5.5 A ³
Stall torque @ 12V:	34 kg·cm ³
Max output power @ 12V:	8 W ⁴
No-load speed @ 6V:	49 rpm ⁵
No-load current @ 6V:	0.15 A ⁵
Stall current @ 6V:	3.0 A ⁵
Stall torque @ 6V:	21 kg·cm ⁵
Motor type:	12V

Performance at maximum efficiency

Max efficiency @ 12V:	44 %
Speed at max efficiency:	87 rpm
Torque at max efficiency:	4.2 kg·cm
Current at max efficiency:	0.72 A
Output power at max efficiency:	3.8 W



Espressif Systems ESP32-DevKitC Development Board

Espressif Systems ESP32-DevKitC Development Board is a small space-saving board that supports various ESP32 modules. Most I/O pins are broken out to the pin headers on both sides for easy interfacing. Connect these pins to peripherals as needed. Standard headers on the Espressif Systems ESP32-DevKitC make development easy and convenient when using a breadboard.

FEATURES

- 32-bit Xtensa® dual-core at 240MHz
- Wi-Fi® IEEE® 802.11b/g/n 2.4GHz
- BLUETOOTH® 4.2 BR/EDR and BLE
- 520KB SRAM (16KB for cache)
- 448KB ROM
- 34x GPIOs, 4x SPI, 3x UART, 2x I²C, 2x I²S, RMT, LED PWM, and 1x host SD/eMMC/SDIO
- 1x slave SDIO/SPI, TWAI®, 12-bit ADC, and Ethernet

APPLICATIONS

- IoT projects
- Home automation
- Wearable devices
- Sensor data collection
- Prototyping
- Robotics
- Smart agriculture
- Mesh networking
- Audio applications
- Educational projects

REQUIRED EQUIPMENT

- USB A/micro USB B cable
- Computer running Windows, Linux, or macOS

FUNCTIONAL DESCRIPTION

Functional Description

• ESP32-WROOM-32

ESP32-WROOM-32 module soldered to the ESP32-DevKitC V4 board. Optionally an ESP32-WROOM-32D or ESP32-WROOM-32U module may be soldered instead of the ESP32-WROOM-32.

• USB-UART Bridge

A single-chip USB-UART bridge provides up to 3Mbps transfer rates.



Functional Description

- **ESP32-WROOM-32**

ESP32-WROOM-32 module soldered to the ESP32-DevKitC V4 board. Optionally an **ESP32-WROOM-32D** or **ESP32-WROOM-32U** module may be soldered instead of the ESP32-WROOM-32.

- **USB-UART Bridge**

A single-chip USB-UART bridge provides up to 3Mbps transfer rates.

- **Boot**

Download button: Holding the Boot button and pressing the EN button initiates the firmware download mode. The user can then download firmware through the serial port.

- **Micro USB Port**

The USB interface functions as the power supply for the board and the communication interface between the PC and the ESP module.

- **5V Power On LED**

The LED lights up when the USB or an external 5V power supply is applied to the board. For details, see the schematic in Related Documents.

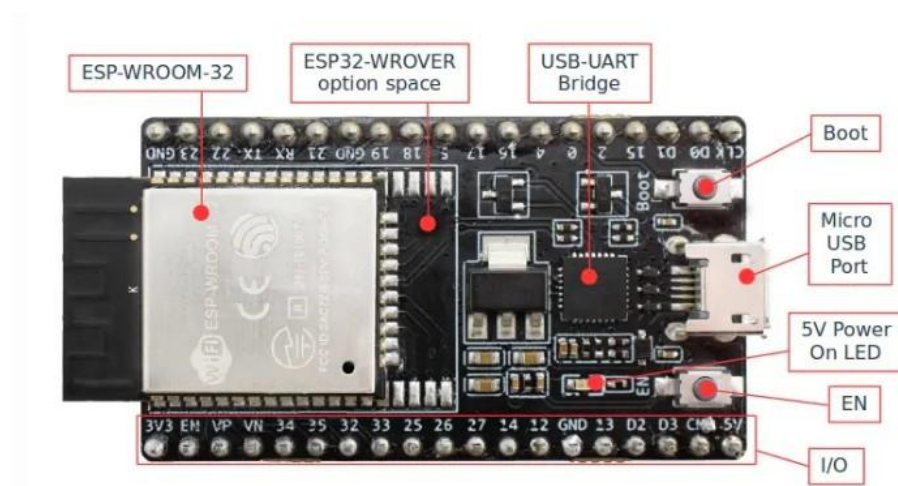
- **EN**

Reset button: pressing this button resets the system.

- **I/O**

Most of the pins on the ESP module are broken out into the pin headers on the board. Users can program ESP32 to enable multiple functions such as PWM, ADC, DAC, I²C, I²S, SPI, etc.

OVERVIEW





Pinout



PIN	Default State	Description
VIN		The connection point for the positive side of the 5.5 – 24 V motor power supply. Since the overvoltage protection can be as low as 24 V, we do not recommend using 24V batteries for VIN.
VDD		The connection point for the positive side of the logic power supply (typically 2.5 – 5 V). The only function of this pin is to power the internal pull-ups on the two enable lines, ENA and ENB.
VOUT		This pin gives you access to the motor power supply after the reverse-voltage protection MOSFET (see the board schematic below). It can be used to supply reverse-protected power to other components in the system, but it should not be used for high currents. This pin should only be used as an output.
GND		Ground connection points for logic and motor power supplies.
OUTA		Output of half-bridge A (connects to one terminal of a DC motor).
OUTB		Output of half-bridge B (connects to the other terminal of a DC motor).
PWM	LOW	Pulse width modulation input: a PWM signal on this pin corresponds to a PWM output on the motor outputs.
INA	FLOAT	Motor direction input A ("clockwise" input).
INB	FLOAT	Motor direction input B ("counterclockwise" input).
CS		Current sense output. The pin voltage is roughly 140 mV per amp of output current when the CS_DIS pin is low or disconnected. The current sense reading is more accurate at higher currents. The CS pin is designed for PWM frequencies of 5 kHz or higher. If you use a PWM frequency lower than 5 kHz and want to measure the current, we recommend adding an extra capacitor between the CS pin and GND to smooth out the signal. For example, if you use a PWM frequency of 490 Hz and want to measure the current, you should add a 1 μ F capacitor (or larger) between CS and GND. (Note that while the CS voltage can potentially exceed 3.3 V at high currents, the current sense circuit should be safe for use with many 3.3V analog inputs. Most MCUs have integrated protection diodes that will clamp the input voltage to a safe value, and since the CS circuit has a 10 k Ω resistor in series with the output, only a few hundred microamps at most will flow through that diode.)
ENA/DIAG A	HIGH	Combination enable input/diagnostic output for half-bridge A. When the driver is functioning normally, this pin acts as an enable input, with a logical high enabling half-bridge A and a logical low disabling half-bridge A. When a driver fault occurs, the IC drives this pin low and half-bridge A is disabled. This pin is connected to VDD through a pull-up resistor on the board.



ACS712

**Fully Integrated, Hall Effect-Based Linear Current Sensor
with 2.1 kV_{RMS} Voltage Isolation and a Low-Resistance Current Conductor**

Features and Benefits

- Low-noise analog signal path
- Device bandwidth is set via the new FILTER pin
- 5 μ s output rise time in response to step input current
- 80 kHz bandwidth
- Total output error 1.5% at $T_A = 25^\circ\text{C}$
- Small footprint, low-profile SOIC8 package
- 1.2 m Ω internal conductor resistance
- 2.1 kV_{RMS} minimum isolation voltage from pins 1-4 to pins 5-8
- 5.0 V, single supply operation
- 66 to 185 mV/A output sensitivity
- Output voltage proportional to AC or DC currents
- Factory-trimmed for accuracy
- Extremely stable output offset voltage
- Nearly zero magnetic hysteresis
- Ratiometric output from supply voltage



TUV America
Certificate Number:
U8V 06 05 54214 010



Package: 8 Lead SOIC (suffix LC)



Approximate Scale 1:1



Description

The Allegro® ACS712 provides economical and precise solutions for AC or DC current sensing in industrial, commercial, and communications systems. The device package allows for easy implementation by the customer. Typical applications include motor control, load detection and management, switched-mode power supplies, and overcurrent fault protection.

The device consists of a precise, low-offset, linear Hall sensor circuit with a copper conduction path located near the surface of the die. Applied current flowing through this copper conduction path generates a magnetic field which is sensed by the integrated Hall IC and converted into a proportional voltage. Device accuracy is optimized through the close proximity of the magnetic signal to the Hall transducer. A precise, proportional voltage is provided by the low-offset, chopper-stabilized BiCMOS Hall IC, which is programmed for accuracy after packaging.

The output of the device has a positive slope ($>V_{IOUT(Q)}$) when an increasing current flows through the primary copper conduction path (from pins 1 and 2, to pins 3 and 4), which is the path used for current sensing. The internal resistance of this conductive path is 1.2 m Ω typical, providing low power

Continued on the next page...



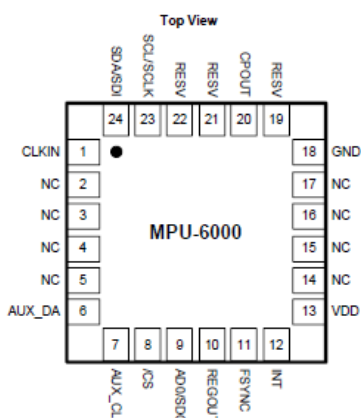
MPU-6000/MPU-6050 Product Specification

Document Number: PS-MPU-6000A-00
Revision: 3.4
Release Date: 08/19/2013

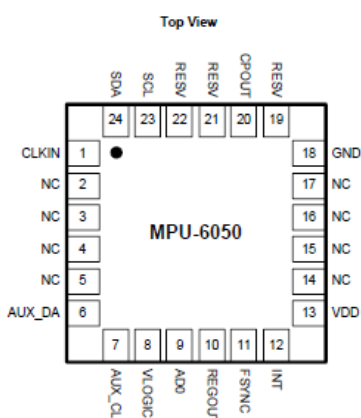
7 Applications Information

7.1 Pin Out and Signal Description

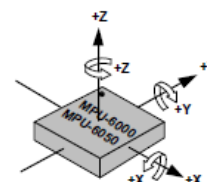
Pin Number	MPU-6000	MPU-6050	Pin Name	Pin Description
1	Y	Y	CLKIN	Optional external reference clock input. Connect to GND if unused.
6	Y	Y	AUX_DA	I ² C master serial data, for connecting to external sensors
7	Y	Y	AUX_CL	I ² C Master serial clock, for connecting to external sensors
8	Y		/CS	SPI chip select (0=SPI mode)
8		Y	VLOGIC	Digital I/O supply voltage
9	Y		AD0 / SDO	I ² C Slave Address LSB (AD0); SPI serial data output (SDO)
9		Y	AD0	I ² C Slave Address LSB (AD0)
10	Y	Y	REGOUT	Regulator filter capacitor connection
11	Y	Y	FSYNC	Frame synchronization digital input. Connect to GND if unused.
12	Y	Y	INT	Interrupt digital output (totem pole or open-drain)
13	Y	Y	VDD	Power supply voltage and Digital I/O supply voltage
18	Y	Y	GND	Power supply ground
19, 21	Y	Y	RESV	Reserved. Do not connect.
20	Y	Y	CPOUT	Charge pump capacitor connection
22	Y	Y	RESV	Reserved. Do not connect.
23	Y		SCL / SCLK	I ² C serial clock (SCL); SPI serial clock (SCLK)
23		Y	SCL	I ² C serial clock (SCL)
24	Y		SDA / SDI	I ² C serial data (SDA); SPI serial data input (SDI)
24		Y	SDA	I ² C serial data (SDA)
2, 3, 4, 5, 14, 15, 16, 17	Y	Y	NC	Not internally connected. May be used for PCB trace routing.



QFN Package



QFN Package



Orientation of Axes of Sensitivity and



DS18B20

Programmable Resolution 1-Wire Digital Thermometer

General Description

The DS18B20 digital thermometer provides 9-bit to 12-bit Celsius temperature measurements and has an alarm function with nonvolatile user-programmable upper and lower trigger points. The DS18B20 communicates over a 1-Wire bus that by definition requires only one data line (and ground) for communication with a central microprocessor. In addition, the DS18B20 can derive power directly from the data line ("parasite power"), eliminating the need for an external power supply.

Each DS18B20 has a unique 64-bit serial code, which allows multiple DS18B20s to function on the same 1-Wire bus. Thus, it is simple to use one microprocessor to control many DS18B20s distributed over a large area. Applications that can benefit from this feature include HVAC environmental controls, temperature monitoring systems inside buildings, equipment, or machinery, and process monitoring and control systems.

Applications

- Thermostatic Controls
- Industrial Systems
- Consumer Products
- Thermometers
- Thermally Sensitive Systems

Benefits and Features

- Unique 1-Wire® Interface Requires Only One Port Pin for Communication
- Reduce Component Count with Integrated Temperature Sensor and EEPROM
 - Measures Temperatures from -55°C to +125°C (-67°F to +257°F)
 - $\pm 0.5^\circ\text{C}$ Accuracy from -10°C to +85°C
 - Programmable Resolution from 9 Bits to 12 Bits
 - No External Components Required
- Parasitic Power Mode Requires Only 2 Pins for Operation (DQ and GND)
- Simplifies Distributed Temperature-Sensing Applications with Multidrop Capability
 - Each Device Has a Unique 64-Bit Serial Code Stored in On-Board ROM
- Flexible User-Definable Nonvolatile (NV) Alarm Settings with Alarm Search Command Identifies Devices with Temperatures Outside Programmed Limits
- Available in 8-Pin SO (150 mils), 8-Pin μSOP , and 3-Pin TO-92 Packages

Pin Configurations

