



ΠΑΝΕΠΙΣΤΗΜΙΟ ΘΕΣΣΑΛΙΑΣ
ΣΧΟΛΗ ΘΕΤΙΚΩΝ ΕΠΙΣΤΗΜΩΝ
ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ ΜΕ ΕΦΑΡΜΟΓΕΣ ΣΤΗΝ ΒΙΟΪΑΤΡΙΚΗ

Σχεδιασμός και Κατασκευή Ενσωματωμένου
Συστήματος Μέτρησης σε Κηδεμόνα
Παρακολούθησης Θεραπείας Ιδιοπαθούς Σκολίωσης

ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ

της

ΔΗΜΗΤΡΑΣ ΠΡΑΠΑ

(ΑΜ: 02288)

Επιβλέπων : Παναγιώτης Βαρθολομαίος
Επίκουρος Καθηγητής, Ρομποτικά Συστήματα

Λαμία Οκτώβριος - 2025



ΠΑΝΕΠΙΣΤΗΜΙΟ ΘΕΣΣΑΛΙΑΣ
ΣΧΟΛΗ ΘΕΤΙΚΩΝ ΕΠΙΣΤΗΜΩΝ
ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ ΜΕ ΕΦΑΡΜΟΓΕΣ ΣΤΗΝ ΒΙΟΪΑΤΡΙΚΗ

Εγκρίθηκε από την εξής εξεταστική επιτροπή στις 2/10/2025

Αθανάσιος Κακαρούντας

Καθηγητής - Πρόεδρος Τμήματος

Ενσωματωμένα Υπολογιστικά Συστήματα

Τσαρούχας Παναγιώτης

Καθηγητής

Ανάλυση Αξιοπιστίας σε συστήματα παραγωγής

Λαμία Οκτώβριος - 2025

Copyright © 2025 – ΔΗΜΗΤΡΑ ΠΡΑΠΑ

Απαγορεύεται η αντιγραφή, αποθήκευση και διανομή της παρούσας εργασίας, εξ ολοκλήρου ή τμήματος αυτής, για εμπορικό σκοπό. Επιτρέπεται η ανατύπωση, αποθήκευση και διανομή για σκοπό μη κερδοσκοπικό, εκπαιδευτικής ή ερευνητικής φύσης, υπό την προϋπόθεση να αναφέρεται η πηγή προέλευσης και να διατηρείται το παρόν μήνυμα.

Οι απόψεις και τα συμπεράσματα που περιέχονται σε αυτό το έγγραφο εκφράζουν αποκλειστικά τον συγγραφέα και δεν αντιπροσωπεύουν τις επίσημες θέσεις του Πανεπιστημίου Θεσσαλίας.

Ως συγγραφέας της παρούσας εργασίας δηλώνω πως η παρούσα εργασία δεν αποτελεί προϊόν λογοκλοπής και δεν περιέχει υλικό από μη αναφερόμενες πηγές.

Περίληψη

Η διπλωματική εργασία αυτή εκπονήθηκε με σκοπό την δημιουργία μιας wearable συσκευής η οποία εφαρμόζεται πάνω σε νάρθηκα (κηδεμόνα) που χρησιμοποιείται για την θεραπεία ιδιοπαθούς σκολίωσης. Περιλαμβάνει τόσο τον σχεδιασμό της συσκευής και της εφαρμογής παρακολούθησης των δεδομένων σε πραγματικό χρόνο όσο και την ανάπτυξή τους και την πειραματική επαλήθευση των αποτελεσμάτων τους. Μέσω της συγκεκριμένης συσκευής, η οποία αποτελεί ενσωματωμένο σύστημα, γίνεται παρακολούθηση σε πραγματικό χρόνο (real time monitoring) της τοποθέτησης του κηδεμόνα, μέσω αισθητήρων πίεσης και συνδέεται, μέσω bluetooth, με εφαρμογή σε κινητό τηλέφωνο ή tablet. Με τη χρήση της προαναφερόμενης εφαρμογής ελέγχεται το σύνολο των ωρών ημερησίως που εφαρμόζεται ο νάρθηκας στο σώμα του ασθενούς καθώς και το αν αυτός εφαρμόζεται σωστά ή έχει χάσει την κατάλληλη για την θεραπεία θέση του, ενώ ταυτόχρονα διατηρείται αρχείο καταγραφών για το σύνολο αυτών των πληροφοριών.

Λέξεις Κλειδιά: real time monitoring, wearable embedded system, force sensors, Bluetooth connection, mobile application, brace for scoliosis

Abstract

This undergraduate thesis was conducted with the aim of developing a wearable device that can be applied to a brace used in the treatment of idiopathic scoliosis. The project includes the design and implementation of both the device and a real-time data monitoring application, as well as their development and experimental validation. Through this device, which functions as an embedded system, real-time monitoring of the brace's placement on the patient's body is achieved using force sensors. The system connects via Bluetooth to a mobile phone or tablet application. Using this application, it is possible to monitor the total daily duration for which the brace is worn, as well as to determine whether it is correctly positioned or has deviated from its proper therapeutic placement. At the same time, a data log is maintained, recording all relevant information.

Keywords: real-time monitoring, wearable embedded system, force sensors, Bluetooth connection, mobile application, scoliosis brace

Πίνακας Περιεχομένων

1	Εισαγωγή.....	15
1.1	Κίνητρο και οφέλη	15
1.2	Σύντομη περιγραφή προβλήματος.....	16
1.3	Συνεισφορά της εργασίας.....	16
1.4	Δομή της εργασίας.....	17
2	Εφηβική Ιδιοπαθής Σκολίωση (AIS).....	18
2.1	Ορισμός εφηβικής ιδιοπαθούς σκολίωσης (AIS).....	18
2.2	Τρόποι αντιμετώπισης της εφηβικής ιδιοπαθούς σκολίωσης (AIS)	19
2.3	Ανάλυση θεραπείας ιδιοπαθούς σκολίωσης μέσω κηδεμόνων.....	19
2.4	Τρόπος κατασκευής νάρθηκα.....	20
2.5	Προβλήματα κατά την εφαρμογή του νάρθηκα στην θεραπεία.....	23
2.6	Βιβλιογραφική Ανασκόπηση (τι έχουν κάνει άλλοι επιστήμονες/μηχανικοί σε αυτό το χώρο).....	25
3	Στόχοι Σχεδιασμού.....	28
3.1	Βελτίωση συμμόρφωσης	28
3.2	Βελτίωση ακρίβειας της εφαρμογής και της θεραπείας	29
3.3	Βελτίωση ελέγχου συμμόρφωσης από τους γονείς, θεραπευτές και ιατρούς	29
3.4	Όφελος για τον ίδιο τον ιατρό και τον θεραπευτή.....	30
4	Σχεδιασμός και Ανάπτυξη Φορητής Συσκευής	32
4.1	Προδιαγραφές Συστήματος	32
4.2	Αρχιτεκτονική και Σχεδιασμός Συστήματος	33
5	Ανάπτυξη Συστήματος.....	37
5.1	Μοντέλα εργαλείων που χρησιμοποιήθηκαν	37
5.2	Λόγος επιλογής των συγκεκριμένων μοντέλων	37
5.3	Συνδεσμολογία Υλικών και Πληροφορίες για τον Τρόπο Υλοποίησης και Λειτουργίας	42

5.3.1	A) Bluetooth module	42
5.3.2	B) Force Sensors (FSR)	48
5.3.3	C) SD Card Module	53
5.3.4	D) RTC Module	55
5.3.5	E) Lithium Battery, Boost Module and Charging Module for Lithium Battery 56	
5.4	Σταθεροποίηση Συστήματος Μέσω Συγκόλλησης	62
6	Σχεδιασμός και Υλοποίηση Λογισμικού	64
6.1	Τρόπος Λειτουργίας	64
6.2	Διαδικασία δημιουργίας εφαρμογής μέσω MIT App Inventor.....	71
6.3	Αναλυτική Περιγραφή Συλλογιστικής Πορείας και Λειτουργικών Ενοτήτων ...	72
6.3.1	Εγκατάσταση και Σύνδεση Bluetooth	72
6.3.2	Επεξεργασία Εισερχόμενων Δεδομένων	72
6.3.3	Οπτικοποίηση Αισθητήρων	72
6.3.4	Χειρισμός Χρόνου και Ρύθμιση Ρολογιού	72
6.3.5	Οπτική Διεπαφή και Διαχείριση Οθονών	73
6.4	Οδηγίες για Δημιουργία της Εφαρμογής.....	73
6.4.1	Βήμα 1: Σχεδίαση GUI στο MIT App Inventor	73
6.4.2	Βήμα 2: Δημιουργία Μεταβλητών και Λιστών	73
6.4.3	Βήμα 3: Bluetooth Ροή Δεδομένων.....	73
6.4.4	Βήμα 4: Χρονικός Συγχρονισμός	73
6.4.5	Βήμα 5: Αποστολή Εντολών	74
6.4.6	Βήμα 6: Build Εφαρμογής	74
6.5	Στιγμιότυπο του κώδικα σε blocks.....	75
7	Ανάλυση Λογισμικού ελέγχου φορητής συσκευής.....	77
7.1	Δηλώσεις	77
7.2	Setup().....	80
7.2.1	Ρύθμιση αναλογικού μετατροπέα & σειριακών διεπαφών	82

7.2.2	Δήλωση pins FSR ως είσοδοι & αρχικό μήνυμα.....	82
7.2.3	SD κάρτα: αρχικοποίηση, σημάνσεις κατάστασης, δημιουργία αρχείου .	83
7.2.4	RTC (DS3231): I2C, ανίχνευση και επαναφορά χρόνου	83
7.3	Συναρτήσεις.....	84
7.3.1	Η συνάρτηση readSmoothedFSR()	84
7.3.2	Η συνάρτηση checkForBluetoothCommands()	85
7.4	Loop().....	88
8	Πειραματικές δοκιμές συνολικού συστήματος	93
8.1	Συμπεράσματα Πειραματικών Δοκιμών.....	105
9	Συμπεράσματα.....	107
A.	Παράρτημα Κώδικα σε περιβάλλον Arduino IDE	108
	Πληροφορίες Προγράμματος Δημιουργίας Κώδικα	108
	Παράρτημα συνολικού κώδικα	111
B.	Παράρτημα κώδικα σε MATLAB	131
10	Αναφορές	134

Λίστα Εικόνων

Εικόνα 1: Σκολίωση.....	18
Εικόνα 2: Scanning with Formetric 4D	21
Εικόνα 3: Result on the screen from Formetric 4D	21
Εικόνα 4: Κατεργασία σφουγγαριού με εργαλειομηχανή.	21
Εικόνα 5: Σφουγγάρι.	21
Εικόνα 6: Application of plastic material.....	22
Εικόνα 7: The cutting of the plastic	22
Εικόνα 8: Διάγραμμα DFD.....	36
Εικόνα 10: Teensy Microcontroller b.....	38
Εικόνα 9: Teensy Microcontroller a.....	38
Εικόνα 11: FSR Sensor	38
Εικόνα 12: SD Card Module	39
Εικόνα 13: SD Card 16GB.....	39
Εικόνα 14: Πίσω όψη RTC DS3231	40
Εικόνα 15: Μπροστά όψη RTC DS3231	40
Εικόνα 16: Coin Cell Battery CR2032.....	40
Εικόνα 17: Li-Po Battery	40
Εικόνα 18: Boost Module FIT0471	41
Εικόνα 19: Charging Module TP4056	41
Εικόνα 20: Bluetooth Module HC-05	42
Εικόνα 21: Κατασκευή FSR	49
Εικόνα 22: Καμπύλη Αντίστασης έναντι Δύναμης για FSR.....	49
Εικόνα 23: Κύκλωμα διαχωριστή τάσης και καμπύλες Vout vs Force για διαφορετικές τιμές R, όπου g σημαίνει γραμμάρια.	52
Εικόνα 24: Master-Slaves	53
Εικόνα 25: Ένας boost converter στην κατάσταση ενεργοποίησης. Το ρεύμα τροφοδοσίας και το ρεύμα φορτίου υποδεικνύονται και τα δύο με πράσινο χρώμα.....	57
Εικόνα 26: Ροή ρεύματος μέσω ενός boost converter στην κατάσταση απενεργοποίησης.	58
Εικόνα 27: Η δίοδος επιτρέπει τη ροή ρεύματος από τον επαγωγέα στον πυκνωτή, αλλά όχι αντίστροφα.....	59
Εικόνα 28: Αναλυτική Συνδεσμολογία	60
Εικόνα 29: Συνδεσμολογία Υλικών σε bradboard 2	61

Εικόνα 29: Συνδεσμολογία Υλικών σε bradboard 1	61
Εικόνα 31: Διακόπτης.....	62
Εικόνα 32: Υλικά για Συγκόλληση.....	62
Εικόνα 33: Τελικό αποτέλεσμα συγκόλλησης.....	63
Εικόνα 34: Τελικό Αποτέλεσμα με διακόπτη, μπαταρία και φορτιστή	63
Εικόνα 40: Εμφάνιση μηνύματος για περισσότερη προσπάθεια	66
Εικόνα 39: Εμφάνιση μηνύματος επιτυχίας	66
Εικόνα 37: Ενημέρωση ημερομηνίας και ώρας. Με το πάτημα του κουμπιού "For Cell Battery" και έπειτα του " Time Update".....	66
Εικόνα 36: Μετά την Σύνδεση μέσω του κουμπιού “ Connect Bluetooth”και σε περίπτωση όπου η κάρτα SD δεν καταγράφει.....	66
Εικόνα 35: Αρχική Εικόνα Εφαρμογής Πριν την Σύνδεση	66
Εικόνα 38: Εικόνα Εφαρμογής αφού έχει πατηθεί το κουμπί "See the Sensors Results" και ενώ και η κάρτα SD καταγράφει και ασκούνται δυνάμεις και στους 3 αισθητήρες..	66
Εικόνα 41: Εμφάνιση μηνύματος για το ότι είναι κοντά στον στόχο	67
Εικόνα 42: Αλλαγή στόχου σε 1 λεπτό μέσω της εφαρμογής	67
Εικόνα 43: Δοκιμή μετά την αλλαγή του στόχου μέσω της εφαρμογής	67
Εικόνα 45: Daily Report από το αρχείο .txt. για μη επίτευξη στόχου.....	68
Εικόνα 44: Daily Report από το αρχείο .txt. για επίτευξη στόχου	68
Εικόνα 48: Μήνυμα για το ότι η κάρτα SD καταγράφει ξανά σε περίπτωση που για κάποιο διάστημα δεν γινόταν καταγραφή.	69
Εικόνα 47: Μήνυμα στο .txt αρχείο για ενημέρωση ημερομηνίας και ώρας από την εφαρμογή.....	69
Εικόνα 46: Μήνυμα στο .txt αρχείο για ανανέωση στόχου εφαρμογής	69
Εικόνα 49: Desing MIT App Inventor.....	71
Εικόνα 50: Περιβάλλον MIT App Inventor	74
Εικόνα 51: Στιγμιότυπο Κώδικα σε Blocks από MIT App Inventor.....	75
Εικόνα 52: Τοποθέτηση νάρθηκα στο σώμα μπροστά και πίσω όψη.....	94
Εικόνα 53: Εφαρμογή , πείραμα 1 (α)	95
Εικόνα 54: Διάγραμμα σε matlab, πείραμα 1 (α)	95
Εικόνα 58: Διάγραμμα σε matlab, πείραμα 1 (γ)	96
Εικόνα 57: Εφαρμογή , πείραμα 1 (γ).....	96
Εικόνα 56: Διάγραμμα σε matlab, πείραμα 1 (β)	96
Εικόνα 55: Εφαρμογή , πείραμα 1 (β)	96
Εικόνα 59: Μήνυμα για επίτευξη στόχου 1ου πειράματος.....	97

Εικόνα 60: Αντιστοίχιση μετρήσεων πειράματος 1 με το αρχείο .txt καταγραφής δεδομένων	98
Εικόνα 64: Διάγραμμα matlab, πείραμα 2 (β).....	99
Εικόνα 63: Εφαρμογή , πείραμα 2 (β).....	99
Εικόνα 62: Διάγραμμα σε matlab, πείραμα 2 (α)	99
Εικόνα 61: Εφαρμογή, πείραμα 2 (α).....	99
Εικόνα 65: Μήνυμα κοντά στην επίτευξη στόχου 2ου πειράματος	100
Εικόνα 66: Αντιστοίχιση μετρήσεων πειράματος 2 με το αρχείο .txt καταγραφής δεδομένων	100
Εικόνα 68: Μήνυμα αποτυχίας επίτευξης στόχου 3ου πειράματος.....	102
Εικόνα 67: Μετρήσεις 3ου πειράματος	102
Εικόνα 69: Αντιστοίχιση μετρήσεων πειράματος 3 με το αρχείο .txt καταγραφής δεδομένων	103
Εικόνα 72: Μήνυμα επίτευξης στόχου 4ου πειράματος	104
Εικόνα 71: Μετρήσεις 4ου πειράματος	104
Εικόνα 70: Αλλαγή στόχου μέσω εφαρμογής, 4ου πειράματος.....	104
Εικόνα 73: Αντιστοίχιση μετρήσεων πειράματος 4 με το αρχείο .txt καταγραφής δεδομένων	104
Εικόνα 74: Περιβάλλον Arduino IDE.....	108
Εικόνα 75: Επαλήθευση Ελέγχει τον κώδικά για σφάλματα κατά τη μεταγλώττιση. ...	109
Εικόνα 76: Μεταφόρτωση Μεταγλωττίζει τον κώδικά σας και τον μεταφορτώνει στην διαμορφωμένη πλακέτα.	109
Εικόνα 77: Εισαγωγή Βιβλιοθηκών στο Arduino IDE	110
Εικόνα 78: Επιλογή Serial Monitor στο Arduino IDE	110
Εικόνα 79: Επιλογή μικροελεγκτή	111

Λίστα Πινάκων

Table 1: Υλικά/Εργαλεία	37
Table 2: Πίνακας AT Εντολών	44
Table 3: Συνοπτικός Πίνακας Data/ AT mode	45
Table 4: Συνδεσμολογία Bluetooth Module HC-05 με Teensy Microcontroller	46
Table 5: Συνδεσμολογία FSR με Teensy Microcontroller	51
Table 6: Επιρροή σταθερής αντίστασης στο Vout	52
Table 7: Συνδεσμολογία SD Card Module με Teensy Microcontroller	54
Table 8: Συνδεσμολογία RTC Module με Teensy Microcontroller	56
Table 9: Συνδεσμολογία Lithium Battery με Charging Module for Lithium Battery	59
Table 10: Συνδεσμολογία Charging Module for Lithium Battery με Boost Module	59
Table 11: Περιγραφή συνάρτησης readSmoothedFSR()	84

1 Εισαγωγή

Στην παρούσα εργασία, αφού γίνει αναλυτικός προσδιορισμός της ιδιοπαθούς σκολίωσης, περιγράφεται ο σχεδιασμός ενός φορητού ενσωματωμένου συστήματος το οποίο θα τοποθετείται σε μόνιμη βάση πάνω στον νάρθηκα θεραπείας ιδιοπαθούς σκολίωσης. Εξηγείται το κίνητρο του σχεδιασμού, μέσω της παρουσίας των προβλημάτων που δημιουργούνται κατά την θεραπεία της συγκεκριμένης πάθησης μέσω του κηδεμόνα σε συνδυασμό με τα οφέλη της παρούσας συσκευής και της συνεχούς παρακολούθησης μέσω εφαρμογής.

Αναφέρεται αναλυτικά ο σχεδιασμός του συστήματος. Αυτός περιλαμβάνει την παρουσίαση της αρχιτεκτονικής, της επιλογής των υλικών και των εξαρτημάτων, την τοπολογία και τη συνδεσμολογία των υποσυστημάτων, καθώς και η διαδικασία προγραμματισμού τους και υλοποίησης σε περιβάλλον arduino IDE. [1]

Επιπρόσθετα παρατίθεται και ο τρόπος δημιουργίας της κατάλληλης εφαρμογής σε κινητό τηλέφωνο ή tablet σε μορφή προσομοίωσης μέσω mit app inventor. [2]

1.1 Κίνητρο και οφέλη

Το βασικό κίνητρο για την ανάπτυξη του εν λόγω συστήματος προέκυψε από την ανάγκη αντιμετώπισης της αντικειμενικής δυσκολίας που παρουσιάζεται στην συνεχόμενη παρακολούθηση της εφηβικής ιδιοπαθούς σκολίωσης (AIS). Η πρόκληση αυτή αφορά αφενός τη δυσκολία καταγραφής της εξέλιξης της παραμόρφωσης της σπονδυλικής στήλης και αφετέρου την παρακολούθηση της σωστής και σταθερής εφαρμογής του θεραπευτικού νάρθηκα (κηδεμόνα), που αποτελεί κρίσιμο παράγοντα επιτυχίας της συντηρητικής αντιμετώπισης της πάθησης.

Η ανάπτυξη ενός τέτοιου ολοκληρωμένου συστήματος παρέχει σημαντικά πλεονεκτήματα τόσο για τον ασθενή όσο και για το υποστηρικτικό του περιβάλλον — γονείς, ιατρούς και θεραπευτές. Συμβάλλει στην ακριβέστερη και πιο αξιόπιστη παρακολούθηση της θεραπείας, διευκολύνει την έγκαιρη λήψη αποφάσεων, μειώνει την εξάρτηση από επιβλαβείς διαγνωστικές μεθόδους και ενισχύει τη συμμόρφωση του ασθενούς μέσω της άμεσης ανατροφοδότησης και της διαρκούς επίγνωσης της θεραπευτικής πορείας. Ως αποτέλεσμα, επιδιώκεται μια πιο αποτελεσματική, στοχευμένη και σύντομη θεραπευτική διαδρομή.

1.2 Σύντομη περιγραφή προβλήματος

Η εφηβική ιδιοπαθής σκολίωση αποτελεί μια πάθηση που εξελίσσεται παράλληλα με τη σωματική ανάπτυξη του ατόμου, γεγονός που καθιστά απαραίτητη τη **συνεχή και συστηματική παρακολούθηση της πορείας της [3]**. Ωστόσο, η παρακολούθηση αυτή βασίζεται κυρίως στη συχνή λήψη ακτινογραφιών, μια διαδικασία που, αν και διαγνωστικά πολύτιμη, συνεπάγεται την έκθεση του ασθενούς σε επαναλαμβανόμενη ακτινοβολία (κάτι που από ιατρική σκοπιά δεν θεωρείται ενδεδεδειγμένο, ιδίως σε νεαρές ηλικίες) [4]. Παράλληλα, η θεραπευτική προσέγγιση της σκολίωσης συχνά περιλαμβάνει τη χρήση ειδικού νάρθηκα (κηδεμόνα), ο οποίος έχει στόχο να επαναφέρει τη σπονδυλική στήλη σε πιο φυσιολογική θέση, ασκώντας σταθερή πίεση και ενθαρρύνοντας μια σταδιακή, ενδεχομένως μόνιμη, διόρθωση της καμπυλότητας. Παρά τη σημασία του στη θεραπεία, η χρήση του νάρθηκα παρουσιάζει σημαντικές προκλήσεις, καθώς απαιτεί μακροχρόνια συμμόρφωση από τον έφηβο ασθενή, κάτι που συχνά συνοδεύεται από **δυσφορία, ψυχολογική πίεση και χαμηλά ποσοστά συνέπειας στη χρήση του [5]**.

1.3 Συνεισφορά της εργασίας

Η παρούσα πτυχιακή εργασία προσφέρει μια ολοκληρωμένη και πρακτικά εφαρμόσιμη προσέγγιση για τη δημιουργία ενός σύγχρονου συστήματος παρακολούθησης και υποστήριξης της θεραπείας της εφηβικής ιδιοπαθούς σκολίωσης. Περιλαμβάνει αναλυτική περιγραφή των απαραίτητων υλικών για την κατασκευή του συστήματος, επεξηγεί με σαφήνεια τον τρόπο σύνδεσης των επιμέρους εξαρτημάτων, καθώς και τον προγραμματισμό τους, βασισμένη σε υλοποίηση που πραγματοποιήθηκε σε εργαστηριακό περιβάλλον και δοκιμάστηκε με επιτυχία. Επιπλέον, παρουσιάζεται αναλυτικά η διαδικασία δημιουργίας εφαρμογής προσομοίωσης, μέσω της οποίας επιτυγχάνεται ο έλεγχος και η παρακολούθηση των δεδομένων σε πραγματικό χρόνο. Αυτή είναι μια λειτουργικότητα η οποία, έως σήμερα, δεν έχει παρουσιαστεί συγκεντρωμένα και ολοκληρωμένα στη σχετική βιβλιογραφία.

Παράλληλα, η εργασία εμπλουτίζεται με εκτενή ανάλυση της εφηβικής ιδιοπαθούς σκολίωσης, της φύσης και της εξέλιξής της, καθώς και της καθιερωμένης θεραπευτικής αντιμετώπισης μέσω της χρήσης κηδεμόνα. Περιγράφεται λεπτομερώς η διαδικασία κατασκευής του κηδεμόνα, ενώ τονίζονται τα πρακτικά και λειτουργικά προβλήματα που δύναται να επιλυθούν με την αξιοποίηση της προτεινόμενης συσκευής. Συνολικά, η εργασία συνδυάζει επιστημονική τεκμηρίωση, τεχνολογική καινοτομία και εφαρμοσμένη

μηχανική, προσφέροντας ένα πολύτιμο εργαλείο για επαγγελματίες υγείας, ερευνητές και ασθενείς.

1.4 Δομή της εργασίας

Η παρούσα εργασία είναι δομημένη με τρόπο που να καθοδηγεί σταδιακά τον αναγνώστη από τη θεωρητική κατανόηση του προβλήματος μέχρι την τεχνική του επίλυση και την αξιολόγηση της προτεινόμενης λύσης. Αρχικά, παρουσιάζονται το κίνητρο και η σημασία της εργασίας, καθώς και μια σύντομη περιγραφή της εφηβικής ιδιοπαθούς σκολίωσης και των θεραπευτικών προσεγγίσεων που εφαρμόζονται, με έμφαση στη χρήση κηδεμόνα. Ακολουθεί ανάλυση των δυσκολιών που προκύπτουν κατά την εφαρμογή της θεραπείας και εξετάζονται ερευνητικές προσπάθειες και τεχνολογικές λύσεις που έχουν προταθεί στη διεθνή βιβλιογραφία.

Στη συνέχεια, περιγράφεται η σχεδίαση και λειτουργία της φορητής συσκευής παρακολούθησης, καθώς και της εφαρμογής μέσω της οποίας γίνεται η διαχείριση των δεδομένων. Παρουσιάζονται τα υλικά και τα ηλεκτρονικά εξαρτήματα που χρησιμοποιήθηκαν, τεκμηριώνεται η επιλογή τους και αναλύεται ο τρόπος σύνδεσης και ολοκλήρωσης του συστήματος. Περιλαμβάνεται επίσης αναλυτική παρουσίαση του προγραμματιστικού κώδικα που αναπτύχθηκε για την καταγραφή, αποθήκευση και αξιοποίηση των μετρήσεων, καθώς και η μεθοδολογία δημιουργίας της συνοδευτικής εφαρμογής.

Η εργασία ολοκληρώνεται με την παρουσίαση πειραματικών δοκιμών, όπου αξιολογείται η πρακτική απόδοση του συστήματος, και με την επισήμανση των πλεονεκτημάτων που προσφέρει η λύση αυτή τόσο για τον ασθενή όσο και για το οικογενειακό και ιατρικό του περιβάλλον. Τέλος, παρατίθενται οι βιβλιογραφικές πηγές που χρησιμοποιήθηκαν για τη θεωρητική και τεχνολογική τεκμηρίωση της εργασίας

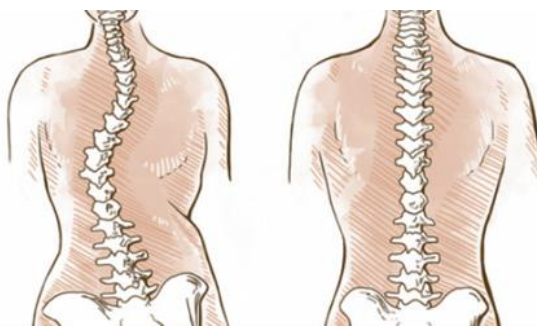
2 Εφηβική Ιδιοπαθής Σκολίωση (AIS)

2.1 Ορισμός εφηβικής ιδιοπαθούς σκολίωσης (AIS)

Η Εφηβική Ιδιοπαθής Σκολίωση, Adolescent Idiopathic Scoliosis (AIS), είναι η **παραμόρφωση του σκελετού**, κατά την οποία η σπονδυλική στήλη σταδιακά χάνει το σχήμα της ευθείας γραμμής που έχει στους μη πάσχοντες και δημιουργεί κύρτωμα (άνω των 10 μοιρών) είτε προς τη μία είτε προς την άλλη κατεύθυνση. Η καμπύλη μπορεί να βρίσκεται είτε στη θωρακική είτε στη οσφυϊκή μοίρα ενώ μπορούν να συνυπάρχουν και οι δύο καμπύλες ταυτόχρονα. Σπανιότερη αλλά υπάρχουσα κατάσταση είναι και η εμφάνιση σκολίωσης στην αυχενική μοίρα. Τα ποσοστά διάδοσής της εξαρτώνται από την ηλικία, τον φύλλο και τον τύπο της.

Συνήθως εμφανίζεται κατά την περίοδο της ανάπτυξης κατά την εφηβεία (εφηβική ιδιοπαθής σκολίωση), αλλά μπορεί να προκύψει και από άλλες αιτίες, όπως γενετικές ανωμαλίες και εκφυλιστικές παθήσεις. Είναι πολυπαραγοντική με γενετικά, ορμονικά, μυοσκελετικά και νευρολογικά αίτια.

Η ήπια σκολίωση κατά βάση δεν εμφανίζει συμπτώματα, μπορεί να συμβάλλει σε μυοσκελετικό πόνο στην πλάτη, αλλά δεν υπάρχουν ενδείξεις ότι προκαλεί αναπηρία ή λειτουργική βλάβη. Ασθενείς με σοβαρή σκολίωση (άνω των 40 μοιρών) μπορεί να έχουν σωματικό πόνο, αισθητική παραμόρφωση, ψυχοκοινωνική δυσφορία ή, σπάνια, πνευμονικές διαταραχές, λόγω της έντονης θωρακικής κάμψης που πιέζει τα πλευρά και τους πνεύμονες, παραμορφώνοντάς τα [6], [7] .



Εικόνα 1: Σκολίωση

[8]

Ποιοι είναι οι παράγοντες κινδύνου της σκολίωσης;

- Ηλικία. Τα συμπτώματα συνήθως ξεκινούν κατά την περίοδο της μέγιστης ανάπτυξης, η οποία ξεκινάει λίγο πριν την εφηβεία.
- Φύλο. Είναι πιο πιθανό οι γυναίκες να έχουν σκολίωση από τους άνδρες.
- Οικογενειακό ιστορικό. Αν και δεν έχουν όλα τα άτομα με οικογενειακό ιστορικό σκολίωση [9].

2.2 Τρόποι αντιμετώπισης της εφηβικής ιδιοπαθούς σκολίωσης (AIS)

Ο τρόπος με τον οποίο γίνεται προσπάθεια για έλεγχο της ιδιοπαθούς σκολίωσης και συνεπώς της ανάπτυξης και εξέλιξης του κυρτώματος, ώστε αυτό να παραμείνει σε ένα επιθυμητό, μη αυξανόμενο στάδιο, ή ακόμα και να μειωθεί είναι είτε η **χορήγηση φυσικοθεραπειών** [10], είτε η **χρήση νάρθηκα**, ενώ σε πιο σπάνιες και σοβαρές περιπτώσεις η **χειρουργική επέμβαση**. Η τελευταία αποτελεί μια λύση που στις περισσότερες περιπτώσεις γίνεται προσπάθεια να αποφευχθεί, εφ' όσον αποτελεί ένα δύσκολο και με μεγάλη επικινδυνότητα χειρουργείο. Ενώ οι άλλοι δύο τρόποι αντιμετώπισης (φυσικοθεραπείες και νάρθηκας) μπορούν να συνδυαστούν για μεγαλύτερη αποδοτικότητα.

2.3 Ανάλυση θεραπείας ιδιοπαθούς σκολίωσης μέσω κηδεμόνων

Στην παρούσα εργασία το ενδιαφέρον επικεντρώνεται στον τρόπο θεραπείας μέσω κηδεμόνων.

Ο **κηδεμόνας για σκολίωση**, πέρα από το ότι βοηθά στην πρόληψη της επιδείνωσης της παραμόρφωσης, μπορεί επίσης να καταστήσει λιγότερο πιθανή τη χειρουργική επέμβαση. Αποτελώντας την δραστικότερη θεραπεία που μπορεί να επιβραδύνει την εξέλιξη της σκολίωσης σε ένα παιδί ή έναν έφηβο του οποίου τα οστά εξακολουθούν να μεγαλώνουν.

Υπάρχουν διαφορετικοί τύποι κηδεμόνων. Ο πιο συνηθισμένος **κηδεμόνας για σκολίωση** σήμερα είναι σε μορφή **νάρθηκα** και καλύπτει το σώμα από τις μασχάλες μέχρι τους γλουτούς, στοχεύει στην υπερδιόρθωση, δηλαδή στην κάμψη του σώματος προς την αντίθετη κατεύθυνση από αυτή που η σκολίωση δημιουργεί. Ένας επίσης

γνωστός και διαδεδομένος κηδεμόνας είναι ο Boston, ο οποίος δεν στοχεύει στην υπερδιόρθωση, αλλά στον απλό ευθειασμό της σπονδυλικής στήλης .

Στη **σκολίωση**, ο **κηδεμόνας** διαμορφώνεται με τέτοιο τρόπο ώστε να κρατά τη σπονδυλική στήλη σε ίσια θέση, ασκώντας πίεση στα κατάλληλα σημεία, τα οποία διαφοροποιούνται αναλόγως την εκάστοτε παραμόρφωση. Διαμορφώνεται κατόπιν λήψης μέτρων και έπειτα από γνωμάτευση του ιατρού και τις απαραίτητες **απεικονιστικές εξετάσεις**.

Ο κηδεμόνας κορμού είναι πλήρως **εξατομικευμένος** για κάθε ασθενή και με την πολύωρη ημερησίως τοποθέτησή του έχει ως στόχο να μην επιτρέψει στην σπονδυλική στήλη να συνεχίσει να αναπτύσσεται με ανεπιθύμητο τρόπο. Αυτό τον καθιστά και έναν από τους αποτελεσματικότερους τρόπους αντιμετώπισης της σκολίωσης, διότι δεν εφαρμόζεται μία συγκεκριμένη ώρα μέσα στη μέρα, αλλά ένα μεγάλο εύρος ωρών το οποίο προσαρμόζεται ανάλογα την **σοβαρότητα** και **εξέλιξη του προβλήματος** [11], [12].

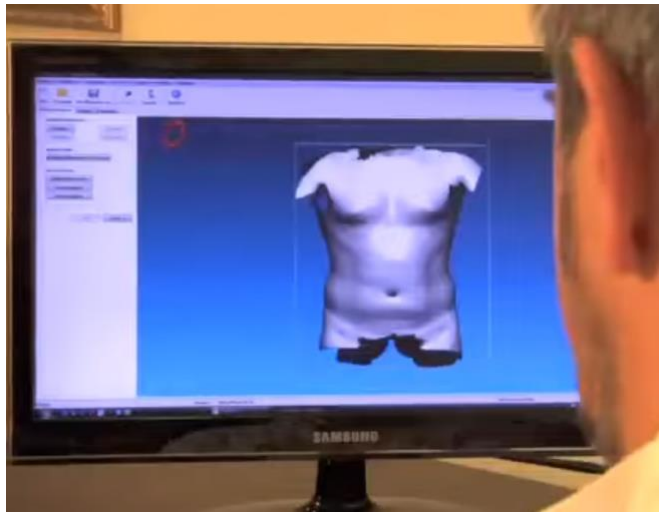
2.4 Τρόπος κατασκευής νάρθηκα

Στα πλαίσια της συγκεκριμένης πτυχιακής εργασίας, έγινε επίσκεψη στο **Scoliosis Spine Laser Center**, με έδρα την Αθήνα [13], μέσω του οποίου παρουσιάστηκαν αναλυτικά **τα βήματα κατασκευής ενός νάρθηκα** για θεραπεία παραμόρφωσης της σπονδυλικής στήλης.

Αρχικά **δημιουργείται ένα καλούπι** μέσω ειδικού μηχανήματος (Formetric 4D), το οποίο αποτυπώνει μια τρισδιάστατη απεικόνιση και ανάλυση, υψηλής ακρίβειας του σώματος του ασθενή **χωρίς ακτινοβολία**. Το καλούπι αυτό δείχνει ακριβώς το σχήμα του σώματος **στην οθόνη** ώστε να μπορεί να κατασκευαστεί εξατομικευμένα ο νάρθηκας πάνω σε αυτό. Η **σάρωση με το Formetric 4D** χρησιμοποιείται για την αξιολόγηση και την παρακολούθηση της καμπυλότητας, της στροφής και της παρεκτόπισης του άξονα της σπονδυλικής στήλης. Αυτό γίνεται με την απομόνωση πολυάριθμων ανατομικών δεικτών της πλάτης του εξεταζόμενου ατόμου. [14].



Εικόνα 2: Scanning with Formetric 4D



Εικόνα 3: Result on the screen from Formetric 4D

Στην συνέχεια, χρησιμοποιούνται μεγάλα κυλινδρικά σφουγγάρια, τα οποία, ανάλογα την σωματοδομή του εκάστοτε ασθενή, διατίθενται σε 3 διαφορετικά μεγέθη και είναι κατασκευασμένα από ειδικό υλικό εύκολο στο κόψιμο. Αυτά τα σφουγγάρια τοποθετούνται σε ένα μηχάνημα το οποίο τα περιστρέφει κόβοντάς τα στο σχήμα απ' το καλούπι του σώματος που έχει δημιουργηθεί μετά το ειδικό scanning. Αυτό επιτυγχάνεται, παίρνοντας τα δεδομένα απ' το Formetric 4D.

Αυτή η διαδικασία, έχει ως αποτέλεσμα, να δημιουργηθεί το **καλούπι** που προαναφέρθηκε, σε **φυσική μορφή και στις πραγματικές διαστάσεις του σώματος που αναπαριστά.**



Εικόνα 5: Σφουγγάρι.



Εικόνα 4: Κατεργασία σφουγγαριού με εργαλειομηχανή.

Ειδικό **πλαστικό υλικό, θερμαίνεται** σε κατάλληλους φούρνους αναπτύσσοντας μεγάλες θερμοκρασίες και δίνοντας του την ικανότητα ανάπλασης. Ενώ το πλαστικό έχει ακόμα υψηλή θερμοκρασία τοποθετείται με ειδικά γάντια πάνω στο καλούπι που έχει εισαχθεί σε ειδική βάση, η οποία του δίνει δυνατότητα περιστροφής, και έτσι **παίρνει ακριβώς το σχήμα του σώματος του ασθενή.**

Το συγκεκριμένο πλαστικό υλικό είναι λεπτό και ελαφρύ ώστε να μην δυσχεραίνει τον χρήστη κατά την πολύωρη εφαρμογή του πάνω στο σώμα, ενώ ταυτόχρονα είναι ιδιαίτερα ανθεκτικό σε χτυπήματα, μειώνοντας σε τεράστιο βαθμό την πιθανότητα να σπάσει ή να φθαρεί.



Εικόνα 6: Application of plastic material



Εικόνα 7: The cutting of the plastic

Αφού αφαιρεθεί για να κρυώσει και πάρει τη μόνιμη μορφή του, (χωρίς πλέον να έχει τη δυνατότητα επαναπροσαρμογής), κόβεται στα κατάλληλα σημεία και τοποθετούνται οι απαιτούμενες **πιέσεις** και οι απαραίτητοι **ιμάντες** που του δίνουν **την τελική λειτουργική του μορφή, έχοντας έτσι έναν πλήρως εξατομικευμένο νάρθηκα.**

2.5 Προβλήματα κατά την εφαρμογή του νάρθηκα στην θεραπεία

Εφ' όσον γίνεται λόγος για εφηβική ιδιοπαθή σκολίωση η ανάπτυξη διαδραματίζει ιδιαίτερα σημαντικό παράγοντα στην εξέλιξη της πάθησης.

Η ανάπτυξη της σπονδυλικής στήλης είναι ένα πολύπλοκο βιολογικό φαινόμενο που περιλαμβάνει αύξηση σε μήκος και αλλαγές στη μορφολογία της, από την ενδομήτρια ζωή μέχρι την ενηλικίωση. Επηρεάζεται από γενετικούς παράγοντες, ορμονικές επιδράσεις και μηχανικά φορτία. Ο ρυθμός αύξησης της σπονδυλικής στήλης κατά τα στάδια ανάπτυξης του οργανισμού δίνει ανομοιογενείς διακυμάνσεις εκτάσεως και ταχύτητας [3].

Είναι εύκολα κατανοητό λοιπόν πως θα πρέπει να γίνονται συνεχώς μετρήσεις για να παρακολουθείται τόσο κατά πόσο αλλάζει η θέση της σπονδυλικής στήλης με τον νάρθηκα κατά την εφαρμογή του όσο και το αν αυτός συνεχίζει να εφαρμόζεται σωστά μετά από κάποιο χρονικό διάστημα (λόγω ανάπτυξης). Σε κάποιες περιπτώσεις μπορεί να γίνονται αλλαγές μετά από το διάστημα ενός μήνα (ή και λιγότερο) και σε άλλες μετά το πέρας πολλών μηνών. Όσο λιγότερες είναι οι αλλαγές τόσο το καλύτερο γιατί σηματοδοτεί την σταθεροποίηση του προβλήματος και άρα την μείωση σταδιακά και των ωρών εφαρμογής του νάρθηκα.

Επιπρόσθετα, κάθε σκολίωση έχει **διαφορετική συμπεριφορά και εξέλιξη**. Άλλες είναι πιο επιθετικές, με αποτέλεσμα να αυξάνονται και να μορφοποιούνται ταχύτερα και άρα να είναι δυσκολότερες στη θεραπεία, ενώ άλλες έχουν πιο σταθερό ρυθμό εξέλιξης [15]. Αυτό ωστόσο αποτελεί έναν αστάθμητο παράγοντα, ο οποίος παρατηρείται μόνο μέσω της συχνής παρακολούθησης.

Ένα βασικό πρόβλημα λοιπόν, που προκύπτει με βάση αυτά που προαναφέρθηκαν είναι πως τόσο λόγω της **ανάπτυξης** όσο και λόγω της **γενικότερης εξέλιξης της πάθησης**, ένας **κηδεμόνας μπορεί να χάσει την απολύτως ακριβή εφαρμογή** του σε ακαθόριστο διάστημα. Αυτό σημαίνει, πως ειδικά για τις περιπτώσεις όπου οι αλλαγές είναι ταχύτερες και καθοριστικές και επομένως οι επιπλοκές και τα συμπτώματα σοβαρότερα, κρίνεται σημαντικό να γίνεται γνωστό, όσο το δυνατόν πιο άμεσα, η στιγμή που ο νάρθηκας δεν θα έχει την πλήρη λειτουργικότητα, ώστε να εφαρμοστούν οι κατάλληλες αλλαγές και να προληφθεί η επιδείνωση.

Γι' αυτό το σκοπό γίνεται παρακολούθηση με **λήψη ακτινογραφιών ανά χρονικό διάστημα** το οποίο ορίζεται από τον γιατρό (ορθοπεδικό) που έχει αναλάβει την εκάστοτε περίπτωση [5]. Ωστόσο, και με αυτόν τον τρόπο, ακόμα και στην περίπτωση που γίνεται

συνταγογράφηση ακτινογραφιών μηνιαία (πιο συχνά δεν εγκρίνεται λόγω της ακτινοβολίας που προσδίδουν, και ειδικά όταν αναφερόμαστε σε παιδιά και εφήβους) [4] μπορεί η διαπίστωση της μειωμένης πλέον για το πρόβλημα πίεσης που ασκεί ο νάρθηκας να γίνει μέρες ή και εβδομάδες μετά την στιγμή που όντως πραγματοποιήθηκε αυτή η αξιοσημείωτη αλλαγή (προφανώς γίνεται σταδιακά αλλά από ένα σημείο και μετά πρέπει να εφαρμοστούν τροποποιήσεις στον κηδεμόνα για να υπάρξει διόρθωση και θεραπεία).

Αυτό **το διάστημα μη σημαντικής διόρθωσης που μπορεί να υπάρξει**, διαδραματίζει **καθοριστικό ρόλο για την εξέλιξη της πάθησης** και ταυτόχρονα δημιουργεί ανώφελη πίεση στον ασθενή, ο οποίος φοράει έναν νάρθηκα που στερεί την πλήρη ελευθερία κίνησης και προκαλεί πολλές φορές δυσφορία χωρίς ωστόσο να είναι και πλήρως λειτουργικός, λόγω της μετέπειτα πορείας της πάθησης του.

Ένα ακόμη σημαντικό και συνηθέστερο πρόβλημα που μπορεί να εμφανιστεί κατά τη διάρκεια μιας τέτοιου είδους θεραπείας, είναι **η μη συμμόρφωση** σε αυτή. Η συμμόρφωση στη θεραπεία της σκολίωσης είναι κρίσιμη για την αποτελεσματικότητά της, είτε πρόκειται για συντηρητική αγωγή με κηδεμόνα και φυσικοθεραπεία είτε για χειρουργική επέμβαση. Η μη συμμόρφωση μπορεί να οδηγήσει σε επιδείνωση της σκολίωσης, μείωση των αποτελεσμάτων της θεραπείας και ανάγκη για περαιτέρω παρεμβάσεις. Ο κηδεμόνας όσο σύγχρονος και εύχρηστος και αν έχει γίνει με την πάροδο των χρόνων, δεν παύει να αποτελεί ένα μέσο που προκαλεί πόνο και ταυτόχρονα μειώνει την πλήρη κινητικότητα του σώματος αφού είναι κατασκευασμένος από άκαμπτο υλικό. Η χρήση του είναι τέτοια ώστε να προσπαθεί να αλλάξει τη θέση της σπονδυλικής στήλης γεγονός που φανερώνει ότι για να επιτύχει την λειτουργία του δεν μπορεί να το κάνει εντελώς ανώδυνα [16], [17].

Επίσης σε συνδυασμό με το ότι πρέπει να φορεθεί από παιδιά και εφήβους που δεν μπορούν να καταλάβουν πολλές φορές σε βάθος την σημαντικότητα του να υποστούν μια τέτοιου είδους δύσκολη θεραπεία, δημιουργεί **το πρόβλημα της αποφυγής**. Αυτό, καθιστά στους γονείς και στους θεραπευτές, δύσκολο το να επιβάλουν την εφαρμογή του κηδεμόνα τις απαραίτητες ώρες ημερησίως καθώς και το να γνωρίζουν με ακρίβεια το πόσες ώρες γίνεται στην πραγματικότητα η εφαρμογή.

2.6 Βιβλιογραφική Ανασκόπηση (τι έχουν κάνει άλλοι επιστήμονες/μηχανικοί σε αυτό το χώρο)

Η διεθνής έρευνα στον χώρο της τεχνολογικής υποστήριξης της AIS έχει παρουσιάσει σημαντική πρόοδο κατά την τελευταία δεκαετία, με εστίαση σε συστήματα παρακολούθησης συμμόρφωσης, συσκευές μέτρησης στάσης και real-time feedback στους ασθενείς.

Αρχικά, μελέτες αναπτύσσουν **υβριδικά συστήματα ενσωματωμένων αισθητήρων** (π.χ. πίεσης), σε συνδυασμό με μικρούς μικροελεγκτές, τα οποία παρέχουν **άμεση ανατροφοδότηση στον χρήστη** σχετικά με τη στάση του σώματος, επιτρέποντας ανίχνευση ακόμα και σε κλίμακα 1° στροφής [18]. Παράλληλα, έχουν πραγματοποιηθεί κλινικές δοκιμές με αισθητήρες θερμοκρασίας και πίεσης ενσωματωμένους σε νάρθηκες σκολίωσης (όπως ο αισθητήρας "Cricket") οποίος απέδειξε αξιοπιστία με απόκλιση μόλις 2%, και κατέγραψε compliance ~78% .

Μελέτες έχουν επίσης παρουσιάσει **συστήματα παρακολούθησης πραγματικού χρόνου**, όπως εφαρμογές που επικοινωνούν μέσω smartphone/cloud και παρέχουν συνεχόμενη καταγραφή και αξιολόγηση χρήσης νάρθηκα, βελτιώνοντας τη θεραπευτική συμμόρφωση και αυξάνοντας την ικανοποίηση ασθενών, με πάνω από 96% θετικό feedback και με 30 συμμετέχοντες. [18]

Παράλληλα, συστηματικές ανασκοπήσεις καταδεικνύουν ότι η χρήση φορητών συστημάτων **βελτιώνει την αποτελεσματικότητα και επιτρέπει απομακρυσμένη παρακολούθηση** αλλά τονίζουν την ανάγκη για περαιτέρω έρευνα στην αξιοπιστία και τη μακροπρόθεσμη χρήση . Μάλιστα, οι εφαρμογές αισθητήρων βασισμένοι στο συνδυασμό θερμοκρασίας-πίεσης φαίνεται να ξεπερνούν τις παραδοσιακές μεθόδους καταγραφής ημερολογίων ή ερωτηματολογίων, αποφεύγοντας την υποκειμενικότητα [18].

Επιπλέον, έχει εξεταστεί και η αποδοχή των ασθενών και των γονέων, οι οποίοι έχουν αξιολογήσει θετικά την αντικειμενική παρακολούθηση μέσω αισθητήρων, σχολιάζοντας ότι βελτιώνει τη συμμόρφωση χωρίς να επιβαρύνει τη σχέση ασθενούς-ιατρού .

Ένα σημαντικό παράδειγμα από την αγορά είναι η συσκευή **ScolioSense** της εταιρείας Aidplex. Πρόκειται για ένα ολοκληρωμένο οικοσύστημα που περιλαμβάνει:

- Ένα μικρό, φορετό module με αισθητήρες πίεσης που ενσωματώνεται στον κηδεμόνα και καταγράφει με ακρίβεια τη διάρκεια χρήσης και τη σωστή εφαρμογή.

- Μια εφαρμογή για κινητά (Android/iOS) με gamified περιβάλλον που καθοδηγεί και επιβραβεύει τον ασθενή, παρέχοντας συντονισμένα μηνύματα, ημερήσια στατιστικά, ερωτηματολόγιο και ενημερώσεις .
- Μια διαδικτυακή πλατφόρμα για ιατρούς και ορθοθεραπευτές, που εμφανίζει γραφικά δεδομένα χρήσης και ειδοποιήσεις αποκλίσεων, υποστηρίζοντας την έγκαιρη παρέμβαση και την απομακρυσμένη διαχείριση της θεραπείας .
- Επιπλέον, οι γονείς λαμβάνουν εβδομαδιαίες ενημερώσεις μέσω SMS, ενισχύοντας την εμπλοκή τους χωρίς να απαιτείται η συνεχής χρήση της εφαρμογής .

Το **ScolioSense** έχει σχεδιαστεί και αναπτυχθεί, με στόχο να προσφέρει ένα αξιόπιστο, διακριτικό και εύκολα ενσωματώσιμο προϊόν σε υπάρχοντες κηδεμόνες (π.χ. Boston, Cheneau), και έχει λάβει έγκριση χρήσης σε κλινικό περιβάλλον. Η πρακτική αξία του έχει αναγνωριστεί καθώς συνδυάζει αξιοπιστία στη μέτρηση, ευκολία στη χρήση και υποστήριξη συμμετέχοντα ιατρού-οικογένειας, ικανοποιώντας ένα σημαντικό κενό: την πραγματική υποστήριξη της θεραπείας σε καθημερινή βάση χωρίς υπερβολική εξάρτηση από ιατρικά ραντεβού ή διαδικασίες [19].

Τέλος, άλλες τεχνολογικές λύσεις εφαρμόζουν **λέιζερ/υπερήχους, video-based gait ανάλυση, ακόμη και ρομποτικές πλατφόρμες**, με στόχο αντικατάσταση ή συμπλήρωση της ακτινογραφίας, μέσω μη επεμβατικής και ασφαλούς διάγνωσης [20].

Συνολικά, η έρευνα αποδεικνύει την αποτελεσματικότητα των φορητών συστημάτων σε πραγματικό χρόνο με αισθητήρες θερμοκρασίας και πίεσης, αλλά και την αναγκαιότητα έρευνας για βελτίωση της ακρίβειας, της χρηστικότητας και της εις βάθος αξιολόγησης μακροπρόθεσμα. Με αυτά τα δεδομένα, η παρούσα εργασία επιχειρεί να συνδυάσει την τεχνολογία των αισθητήρων, την εφαρμογή παρακολούθησης σε πραγματικό χρόνο (real-time monitoring) και τη χρηστικότητα για ταυτόχρονη υποστήριξη ασθενούς και κλινικού προσωπικού.

Τελευταία, μια μελέτη από ομάδα του University Malaya, Swinburne University και University Bayreuth ανέπτυξε οικονομικούς piezoresistive αισθητήρες για νάρθηκες (TLSO), με βάση πολυμερές Velostat, τοποθετημένους ανάμεσα σε αφρώδη υλικά, επιτυγχάνοντας μέσο σφάλμα (~12% RMSE) σε εύρος 0,01–0,13 MPa, και δοκιμάζοντας τους σε πραγματικές συνθήκες καθημερινής χρήσης—κατά τις αναπνοές, το περπάτημα και διαφορετικές στάσεις σώματος [21]. Αυτή η έρευνα αποδεικνύει ότι μπορούν να

κατασκευαστούν χαμηλού κόστους, φορητά και έμπιστα συστήματα παρακολούθησης πίεσης εντός ιατρικών ναρθήκων. [22]

Συμπερασματικά, η υπάρχουσα βιβλιογραφία (συμπεριλαμβανομένης της πρακτικής εφαρμογής του ScolioSense και της μελέτης Velostat) επιβεβαιώνει ότι η συνδυαστική χρήση αισθητήρων πίεσης, εφαρμογών real-time και πλατφορμών υποστήριξης αποτελεί ένα ιδιαίτερα υποσχόμενο πεδίο. Ωστόσο, η αξιοπιστία, η μακροχρόνια ροή δεδομένων και η ευχρηστία σε καθημερινή κλινική χρήση εξακολουθούν να απαιτούν περαιτέρω μελέτη, περιοχή στην οποία η παρούσα εργασία επιδιώκει να συμβάλει με μια προσιτή, εργαστηριακή αλλά και άμεσα εφαρμοστέα λύση.

3 Στόχοι Σχεδιασμού

3.1 Βελτίωση συμμόρφωσης

Το παιδί ή ο έφηβος όπου φοράει τον νάρθηκα πάνω στον οποίο ενσωματώνεται η συσκευή που κατασκευάστηκε στα πλαίσια της παρούσας πτυχιακής εργασίας, αποκτά ένα σαφέστερο και ευκολοδιάκριτο **κίνητρο** να επιτύχει τον στόχο των ωρών που πρέπει να φορά τον νάρθηκα ημερησίως.

Ενισχύεται η συμμετοχή του παιδιού και βελτιώνεται η αποτελεσματικότητα της θεραπείας του, αφού επιτυγχάνεται μεγαλύτερη αναπαραστατικότητα. Το παιδί, που αποτελεί τον θεραπευόμενο, αντιλαμβάνεται την θεραπεία ως μια πιο ευχάριστη διαδικασία κατά την οποία μπορεί να βλέπει και μόνο του μέσα από το κινητό του τηλέφωνο πόσο κοντά ή όχι βρίσκεται στον στόχο που του έχει ανατεθεί παρέχοντάς του ταυτόχρονα οπτική ανατροφοδότηση και κίνητρο για πρόοδο. Η οπτικοποίηση (π.χ. γραφήματα, χρώματα, επιτεύγματα) παρέχει **άμεση επιβράβευση**, που είναι σημαντικό κίνητρο για τα παιδιά. Επίσης με την προσθήκη παιχνιδιού ή «πόντων προόδου» θα επιτευχθεί μεγαλύτερη παρακίνηση.

Με αυτόν τον τρόπο θα προσπαθεί να επιτύχει ένα ιστορικό με όσο το δυνατό περισσότερες επιτυχημένες προσπάθειες. Ενώ ταυτόχρονα σε περίπτωση μη σωστής διαχείρισης δεν το αποθαρρύνει αλλά προσπαθεί να το εμψυχώσει και να του δείξει ότι πολύ πιθανόν να μην απέχει πολύ από τα επιθυμητά όρια.

Όταν το παιδί βλέπει τα δεδομένα του (π.χ. μετρήσεις πίεσης, ημερήσια πρόοδο, στατιστικά), αρχίζει να αντιλαμβάνεται **ότι έχει ενεργό ρόλο** στη θεραπεία του και ενισχύεται η αυτονομία και η υπευθυνότητά του. Μαθαίνει στην ουσία, να **αναλαμβάνει την ευθύνη** για τη φροντίδα του εαυτού του.

Επίσης, αν το παιδί βλέπει μόνο του τα δεδομένα, μπορεί να είναι πιο πρόθυμο να **συζητήσει τα αποτελέσματα** με τους γονείς ή τον γιατρό του, ενισχύοντας την **επικοινωνία** γύρω από τη θεραπεία με τρόπο ώριμο και εποικοδομητικό.

Ακόμα, βλέποντας πώς εξελίσσεται η πάθησή του (αν βελτιώνεται ή επιδεινώνεται), το παιδί **κατανοεί τις συνέπειες** των επιλογών και της συμπεριφοράς του έτσι ώστε να μπορεί να συνδέσει πιο άμεσα, π.χ., την **καλή εφαρμογή της θεραπείας με θετικά αποτελέσματα**.

Όταν ένα παιδί **γνωρίζει τι συμβαίνει στο σώμα του**, μειώνεται το αίσθημα φόβου ή άγχους και η ενημέρωση και η κατανόηση προάγουν την **ψυχική ηρεμία**.

Τέλος, ως αποτέλεσμα της χρήσης μιας τέτοιου είδους συσκευής και εφαρμογής για τον ίδιο τον θεραπευόμενο είναι ότι μαθαίνει από νωρίς να **χειρίζεται ψηφιακά εργαλεία υγείας** και προετοιμάζεται καλύτερα για την περίοδο που θα χρειαστεί να **διαχειρίζεται πλήρως μόνο του την πάθησή του**.

3.2 Βελτίωση ακρίβειας της εφαρμογής και της θεραπείας

Όσο αφορά την ακρίβεια της θεραπείας, είναι σημαντικό να παρέχεται πληροφορία σχετικά με την σωστή τοποθέτηση του νάρθηκα ακόμα και μετά το πέρασμα των βδομάδων και των μηνών όπου μπορεί να υπάρξουν μεταβολές λόγω εξέλιξης της πάθησης ή λόγω ανάπτυξης. Αυτές οι αλλαγές δεν μπορούν να γίνουν αντιληπτές από τον ασθενή ή από τους γονείς/κηδεμόνες του, με αποτέλεσμα να μην επιτυγχάνεται άμεση και έγκαιρη επιδιόρθωση.

Γίνεται ακριβέστερη παρακολούθηση και καταγραφή .Τα δεδομένα (π.χ. πίεση, χρόνος εφαρμογής συσκευής) **καταγράφονται αυτόματα και σε πραγματικό χρόνο**, μειώνοντας τα λάθη της μνήμης ή της υποκειμενικής εκτίμησης.

Μέσω του συγκεκριμένου, συνεχή τρόπου παρακολούθησης, που προσφέρει η δεδομένη εφαρμογή μέσα από τα δεδομένα που λαμβάνει σε πραγματικό χρόνο από την συσκευή , μπορεί να γίνεται έλεγχος της σημαντικής αυξομείωσης των δυνάμεων που ασκεί ο νάρθηκας στο σώμα του ασθενή ανά πάσα στιγμή. Σε περίπτωση που παρατηρηθεί κάποια διαφορά σε αυτές θα είναι μια σημαντική ένδειξη για επανέλεγχο και επίσκεψη στον υπεύθυνο γιατρό.

Οι ειδοποιήσεις ή οι ενδείξεις προόδου στην εφαρμογή υπενθυμίζουν **πότε και για πόση ώρα πρέπει να εφαρμόσει** τη θεραπεία, μειώνοντας την πιθανότητα λάθους ή παράλειψης, γεγονός που διαδραματίζει σημαντικό ρόλο στην ακρίβεια της θεραπείας.

3.3 Βελτίωση ελέγχου συμμόρφωσης από τους γονείς, θεραπευτές και ιατρούς

Μέσω της συγκεκριμένης εφαρμογής παρέχεται, άμεση και αξιόπιστη ενημέρωση των γονέων. Οι γονείς μπορούν να βλέπουν **αν και πότε εφαρμόστηκε η θεραπεία**, χωρίς να

χρειάζονται υποκειμενικές αναφορές, μειώνοντας έτσι τη **σύγκρουση ή τη δυσπιστία**, ειδικά σε παιδιά που δυσκολεύονται να συνεργαστούν ή να δώσουν ακριβείς πληροφορίες.

Οι θεραπευτές και οι γιατροί έχουν πρόσβαση σε **ακριβή, χρονολογημένα δεδομένα συμμόρφωσης**. Αυτό τους επιτρέπει να **αξιολογήσουν την πρόοδο αντικειμενικά** και να προσαρμόσουν τη θεραπεία με βάση πραγματικά στοιχεία. Γνωρίζουν, με βάση τα δεδομένα που τους παρέχονται, τους λόγους προόδου ή μη της θεραπείας. Αν ας πούμε υπάρχει **χαμηλή συμμόρφωση** (π.χ. σπάνια χρήση, διακοπές στη χρήση), εντοπίζεται άμεσα και μπορεί να **αναζητηθεί η αιτία** (π.χ. ενόχληση, ψυχολογική δυσφορία, λανθασμένη χρήση), επιτρέποντας με αυτόν τον τρόπο την **έγκαιρη παρέμβαση** και συζήτηση με το παιδί. Αυτό σημαίνει πως αν ελέγξουν ότι οι πιέσεις παραμένουν αμετάβλητες αλλά η συμμόρφωση ως προς τον χρόνο εφαρμογής είναι μειωμένη, θα προσπαθήσουν να πειθαρχήσουν περισσότερο τον ίδιο τον ασθενή και δεν θα υλοποιήσουν τροποποιήσεις στον νάρθηκα.

Ακόμα, **βελτιώνεται η συνεργασία του θεραπευτή με τον ίδιο τον ασθενή αλλά και με τους γονείς του**. Οι επαγγελματίες δεν χρειάζεται να βασίζονται μόνο στα λεγόμενα των παιδιών και των γονιών τους, αλλά μπορούν να δείχνουν με **χειροπιαστά στοιχεία** την ανάγκη για συμμόρφωση. Έτσι αυξάνεται η **αμοιβαία εμπιστοσύνη** και η εστίαση στο κοινό στόχο.

3.4 Όφελος για τον ίδιο τον ιατρό και τον θεραπευτή

Ο ιατρός και ο θεραπευτής, που αναλαμβάνει την κατασκευή και τοποθέτηση του κηδεμόνα, αποκτούν **συγκεκριμένα και καταγεγραμμένα στοιχεία** για τη συμμόρφωση και την εξέλιξη της θεραπείας (π.χ. συχνότητα χρήσης, ένταση πίεσης, διάρκεια). Αυτό οδηγεί σε **τεκμηριωμένες κλινικές αποφάσεις**, χωρίς να βασίζονται αποκλειστικά σε αναφορές τρίτων ή υποκειμενικές κρίσεις.

Ακόμα, τους βοηθάει στην όσο τον δυνατό πιο εξατομικευμένη θεραπεία, αφού με βάση τα δεδομένα, μπορούν να **προσαρμόσουν τη θεραπεία πιο στοχευμένα** (π.χ. αλλαγή πρωτοκόλλου, προσαρμογή συχνότητας, διαφοροποίηση στόχων). Αυτό αυξάνει την **αποτελεσματικότητα της παρέμβασης** και μειώνει περιττές καθυστερήσεις ή γενικές προσεγγίσεις.

Επιπρόσθετα, εξοικονομείται χρόνος για τους ίδιους και την πορεία της διεξαγωγής της δουλειάς τους εφ' όσον δεν χρειάζεται να αφιερώνεται πολύτιμος χρόνος σε ερωτήσεις

που αφορούν το αν και το πόσο χρησιμοποιήθηκε ο κηδεμόνας. Η συζήτηση εστιάζει **κατευθείαν σε ουσιαστικά θέματα**: πρόοδο, δυσκολίες, ψυχολογικούς παράγοντες κ.λπ.

Σημαντική ακόμα είναι η επίτευξη του έγκαιρος εντοπισμού μη συμμόρφωσης ή επιδείνωσης. Αυτό είναι ιδιαίτερα σημαντικό για τον ιατρό ή θεραπευτή, οι οποίοι αποφεύγουν καθυστερήσεις που θα οδηγούσαν σε **επιπλοκές ή χαμηλότερα αποτελέσματα**. Έτσι γίνεται πιο αποτελεσματική η δουλειά τους αυξάνοντας την ζήτησή τους.

Ακόμα, για τους επαγγελματίες είναι σημαντικό να υπάρχει όσο το δυνατόν πιο ισχυρή και συνεργατική σχέση με τον ασθενή, γεγονός που επιτυγχάνεται μέσα από τον ενεργό ρόλο που αυτός αποκτά. Συνεπώς, εξασφαλίζεται διαφάνεια , η οποία δημιουργεί **εμπιστοσύνη** και καλύτερη επικοινωνία.

Τέλος είναι ιδιαίτερα αξιοσημείωτο, πως καθώς οι ιατροί και οι θεραπευτές αξιοποιούν ψηφιακά εργαλεία δείχνουν **καινοτομία και σύγχρονη προσέγγιση**, κάτι που ενισχύει και την **επαγγελματική τους εικόνα** και την **ποιότητα των υπηρεσιών** τους.

4 Σχεδιασμός και Ανάπτυξη Φορητής Συσκευής

4.1 Προδιαγραφές Συστήματος

Η συσκευή που κατασκευάστηκε διαθέτει 3 **αισθητήρες δύναμης** [23], μέσω των οποίων γίνεται μέτρηση της πίεσης που ασκεί ο νάρθηκας στα σημεία υψηλού ενδιαφέροντος (εκείνα όπου προσαρμόζονται ανάλογα την κάμψη της σπονδυλικής στήλης). Μέσω αυτών των αισθητήρων γίνεται καταμέτρηση των ωρών που τοποθετείται ο κηδεμόνας. Όσο είναι φορεμένος, οι αισθητήρες καταγράφουν δυνάμεις, και στην συνέχεια μέσω κατάλληλου προγραμματισμού γίνεται υπολογισμός των ωρών κατά τις οποίες λάβαμε μετρήσεις δυνάμεων και συνεπώς της χρονικής διάρκειας που ο νάρθηκας ήταν τοποθετημένος στο σώμα του ασθενή.

Ταυτόχρονα με τον ακριβή υπολογισμό των δυνάμεων, σε περίπτωση μεταβολών του σώματος και της ανάπτυξης της σπονδυλικής στήλης, γίνεται παρατήρηση διαμορφώσεων που έχουν ως συνέπεια πιθανή αλλαγή στην ακριβή εφαρμογή του νάρθηκα. Αυτό, πιο συγκεκριμένα, θα επιτυγχάνεται με την καταγραφή των αρχικών δυνάμεων κατά την τοποθέτηση του κηδεμόνα (όπου είναι σίγουρα ακριβής και σωστή) και την σύγκρισή τους με όλες τις επακόλουθες μετρήσεις που θα γίνονται καθημερινά και για όσο ο νάρθηκας είναι τοποθετημένος. Σε περίπτωση που έχουμε ιδιαίτερη αύξηση ή μείωση αυτών, θα αποτελεί ένδειξη για λήψη ακτινογραφίας σε προγενέστερο ίσως χρόνο από τον δειγματοληπτικά καθορισμένο.

Επιπλέον, η συγκεκριμένη συσκευή διαθέτει **κάρτα SD**, καταγραφής δεδομένων, ώστε όλες οι μετρήσεις που γίνονται μέσα στη μέρα να αποθηκεύονται σε ένα αρχείο .txt στο οποίο μπορεί να έχει πρόσβαση τόσο ο ίδιος ο ασθενής και οι κηδεμόνες του, όσο και ο υπεύθυνος ιατρός και θεραπευτής κατά τις επισκέψεις. Για να παρέχεται ωστόσο ακριβές αρχείο είναι σημαντικό πέρα απ' την καταγραφή των μετρήσεων σε αυτό, να γίνεται η κατάλληλη αντιστοίχιση της ημερομηνίας και της ώρας που έγινε η εκάστοτε μέτρηση. Για τον λόγο αυτό χρησιμοποιείται **RTC (Real Time Clock) module**, το οποίο περιλαμβάνει Coin Cell Battery, μέσω της οποίας γίνεται αυτόνομη καταμέτρηση χρόνου, χωρίς να χρειάζεται να λαμβάνει αυτό το δεδομένο από εξωτερική συσκευή.

Παράλληλα, μέσω του κατάλληλου προγραμματισμού, σε περίπτωση που η κάρτα SD για κάποιο λόγο σταματήσει να καταγράφει, με την επόμενη επιτυχημένη καταγραφή εμφανίζεται και αντίστοιχο μήνυμα στο αρχείο .txt, το οποίο ενημερώνει για το ότι η

καταγραφή ξεκίνησε και πάλι επιτυχώς, ώστε να παρατηρηθεί ευκολότερα και να γνωστοποιηθεί το διάστημα μη λειτουργίας της.

Ιδιαίτερα σημαντική είναι η χρήση ενός **bluetooth module**, το οποίο δίνει την δυνατότητα σύνδεσης με κινητό τηλέφωνο ή tablet για να εμφανίζονται τα αποτελέσματα μέσω εφαρμογής.

Τέλος, η συσκευή χρησιμοποιεί **μπαταρία λιθίου (LiPo battery)** που φορτίζεται μέσω USB και μπορεί να διαρκέσει περίπου 22,5 ημέρες με μία μόνο φόρτιση [18], κάνοντας με αυτό τον τρόπο το σύστημα φορητό, χωρίς να είναι απαραίτητη η συνεχόμενη σύνδεση με ρεύμα.

Όλα όσα προαναφέρθηκαν συνδέονται στον **μικροελεγκτή (microcontroller), teensy 4.0**. Ο μικροελεγκτής αποτελεί ολοκληρωμένο σύστημα σε ένα chip με CPU, μνήμη (RAM/Flash), περιφερειακά (UART, SPI, ADC κ.ά.) και είναι σχεδιασμένος για **embedded εφαρμογές**.

4.2 Αρχιτεκτονική και Σχεδιασμός Συστήματος

Η αρχιτεκτονική του συστήματος αποτελείται από τέσσερα βασικά επίπεδα/υποσυστήματα, τα οποία συνεργάζονται για την επίτευξη των στόχων της εφαρμογής.

1. Επίπεδο Αισθητήρων & Υλικού (Hardware Layer)

Αφορά το υλικό που συλλέγει τα δεδομένα:

- **FSR (Force Sensitive Resistors)**: καταγράφουν πίεση εφαρμογής του κηδεμόνα στο σώμα.
- **RTC (DS3231)**: για χρονική σήμανση των μετρήσεων (timestamping).
- **SD Card Module**: για αποθήκευση ιστορικών δεδομένων τοπικά.
- **Bluetooth Module (π.χ. HC-05)**: για ασύρματη μετάδοση σε εφαρμογή.
- **Μικροελεγκτής (π.χ. Teensy 4.0)**: επεξεργάζεται δεδομένα σε πραγματικό χρόνο και ελέγχει τις μονάδες I/O.

2. Επίπεδο Επεξεργασίας & Λογικής (Embedded Software Layer)

- Ανάγνωση αισθητήρων σε σύντομα διαστήματα.
- Εφαρμογή φίλτρων και κατωφλίων (thresholds) για καταγραφή ή αγνόηση μετρήσεων.
- Καταμέτρηση διάρκειας πίεσης (για τον έλεγχο συμμόρφωσης) και δημιουργία ημερήσιας αναφοράς.
- Αποθήκευση με format σε SD. Δηλαδή αποθήκευση κάθε φορά των μετρήσεων, της ημερήσιας αναφοράς και των απαραίτητων μηνυμάτων στην κάρτα.
- Ασύρματη μετάδοση επιλεγμένων δεδομένων μέσω Bluetooth.

3. Επίπεδο Εφαρμογής Χρήστη (Mobile App Layer)

- Ανάπτυξη εφαρμογής (π.χ. MIT App Inventor).
- Λήψη και απεικόνιση πραγματικού χρόνου δεδομένων.
- Παρουσίαση στατιστικών, ενημερώσεων και προειδοποιήσεων.
- Πιθανή αποστολή δεδομένων στο cloud για απομακρυσμένη πρόσβαση (future extension).

4. Επίπεδο Ιατρικής/Θεραπευτικής Υποστήριξης

- Το σύστημα παρέχει **ποσοτικά και αντικειμενικά δεδομένα** στον ιατρό για τη συμμόρφωση.
- Διευκολύνει τη λήψη αποφάσεων (π.χ. αλλαγή ρύθμισης κηδεμόνα).
- Δυνατότητα εβδομαδιαίας εξαγωγής αναφορών για γονείς/θεραπευτές.

Ο σχεδιασμός του συστήματος έχει ως βασικές προτεραιότητες τη **φορητότητα**, την **ευκολία χρήσης**, τη **χαμηλή κατανάλωση ενέργειας** και την **επεκτασιμότητα σε μελλοντικές λειτουργίες ή αναβαθμίσεις**. Επιπλέον, λήφθηκαν υπόψη αρχές του **Human-Centered Design (HCD)** [24], δεδομένου ότι οι τελικοί χρήστες της συσκευής είναι έφηβοι ασθενείς. Το HCD ορίζεται ως μια πολυσύνθετη προσέγγιση επίλυσης

προβλημάτων που τοποθετεί τις ανάγκες, τις ιδιότητες και τη συμπεριφορά των ανθρώπων στο κέντρο της διαδικασίας σχεδιασμού, διασφαλίζοντας ότι προϊόντα και υπηρεσίες είναι πρακτικά χρήσιμα, εργονομικά και φιλικά προς τον χρήστη . [25]

Οι βασικές αρχές του HCD, σύμφωνα με τη διεθνή προδιαγραφή ISO 9241-210 [26], περιλαμβάνουν την κατανόηση των χρηστών, των περιβαλλόντων και των εργασιών, τη διαρκή συμμετοχή των χρηστών σε όλη τη διάρκεια του σχεδιασμού, τους επαναλαμβανόμενους κύκλους σχεδίασης με αξιολόγηση χρηστικότητας και σκοπό τη βελτίωση, τη συνολική αντιμετώπιση της εμπειρίας χρήστη και τη σύνθεση ομάδων με ποικιλία επιστημονικών πεδίων .

Σε αυτό το πλαίσιο, η εργασία εφαρμόζει HCD με μεθόδους όπως:

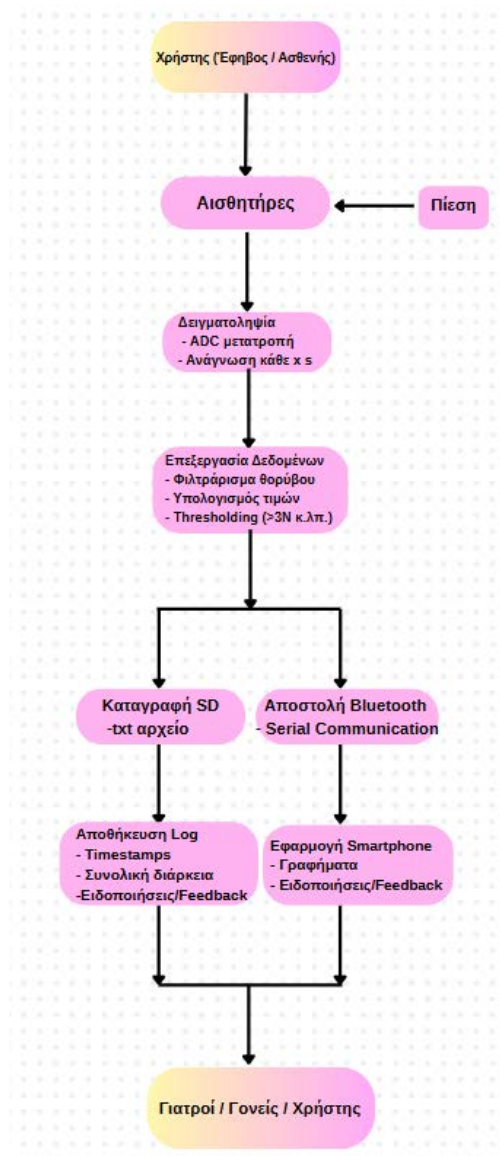
- **Κατανόηση του πλαισίου χρήστη**, των περιορισμών και των δυσκολιών που αντιμετωπίζουν οι έφηβοι στη χρήση κηδεμόνα.
- **Διαμόρφωση πρωτοτύπων και δοκιμές πεδίου** (user-testing prototypes with adolescents) ώστε να διασφαλιστεί η προσβασιμότητα, η άνεση και η καλλιέργεια θετικής εμπειρίας χρήσης.
- **Επαναληπτική βελτίωση**, όπου τα συμπεράσματα από δοκιμές ενσωματώνονται σε επόμενες εκδόσεις του συστήματος.

Παράλληλα, αναγνωρίζονται και τεκμηριώνονται οι **περιορισμοί του σχεδιασμού**, όπως το μέγεθος της μονάδας (module), η καθυστέρηση μετάδοσης μέσω Bluetooth (latency), η χωρητικότητα αποθήκευσης στην κάρτα SD και η αυτονομία της μπαταρίας. Η τεκμηρίωση αυτών των περιορισμών επιτρέπει την εκτίμηση των trade-offs και την αναζήτηση ισορροπίας μεταξύ λειτουργικότητας, διάρκειας και χρηστικότητας.

Σημαντικό ακόμα για την κατανόηση της αρχιτεκτονικής του συστήματος και της μετάδοσης των πληροφοριών και των δεδομένων είναι η κατασκευή ενός DFD διαγράμματος. Τα Διαγράμματα Ροής Δεδομένων (**Data Flow Diagrams – DFDs**) αποτελούν ένα βασικό εργαλείο μοντελοποίησης συστημάτων, καθώς παρέχουν μια απλοποιημένη, γραφική αναπαράσταση της ροής πληροφοριών μέσα σε ένα σύστημα. Χρησιμοποιούνται ευρέως για την κατανόηση, την ανάλυση και τον σχεδιασμό συστημάτων λογισμικού, καθώς επικεντρώνονται στον τρόπο με τον οποίο τα δεδομένα εισέρχονται, επεξεργάζονται, αποθηκεύονται και εξέρχονται από το σύστημα. Ένα DFD αποτυπώνει βασικές οντότητες όπως εξωτερικοί χρήστες ή συστήματα (external entities), διαδικασίες (processes), αποθηκευτικοί χώροι δεδομένων (data stores) και τις

ίδιες τις ροές δεδομένων (data flows). Η απλότητα και η σαφήνειά τους καθιστούν τα DFD ιδανικά για την επικοινωνία μεταξύ τεχνικών και μη τεχνικών ενδιαφερομένων, και χρησιμοποιούνται συχνά στα αρχικά στάδια ανάπτυξης ενός πληροφοριακού συστήματος [27], [28].

DFD Diagram του Συστήματος:



Εικόνα 8: Διάγραμμα DFD

* Το διάγραμμα της παραπάνω εικόνας δημιουργήθηκε μέσω του προγράμματος **Canva** [29], [30]

5 Ανάπτυξη Συστήματος

5.1 Μοντέλα εργαλείων που χρησιμοποιήθηκαν

Για την κατασκευή της προσομοίωσης που δημιουργήθηκε χρησιμοποιήθηκαν τα εξής υλικά και εργαλεία:

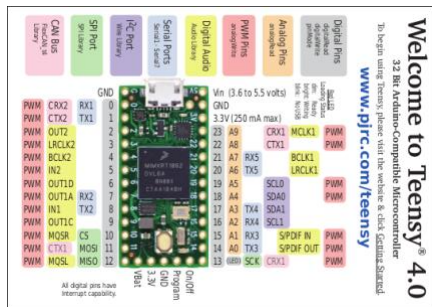
ΥΛΙΚΑ/ΕΡΓΑΛΕΙΑ	ΜΟΝΤΕΛΟ	ΠΟΣΟΤΗΤΑ
Microcontroller	Teensy 4.0	1
Force Sensor	FSR 402, 0.5"	3
SD Card Module	MicroSD Card Module V1.0, DF- DFR0229	1
SD Card	MicroSD, 16GB	1
RTC Module	DS3231	1
RTC Battery	CR2032	1
Lithium Polymer Battery	LP502248 3.7V, 450 mAh	1
Boost Module	DC-DC boost module V1.0, FIT0471	1
Charging Module for Lithium Battery	TP4056	1
Bluetooth Module	HC-05	1

Table 1: Υλικά/Εργαλεία

5.2 Λόγος επιλογής των συγκεκριμένων μοντέλων

1. Teensy 4.0

- Πολύ υψηλή ταχύτητα (600 MHz)
- Μικρές διαστάσεις
- Χαμηλή κατανάλωση
- Κατάλληλο για επεξεργασία πολλών σημάτων σε πραγματικό χρόνο (FSR + RTC + BT + SD). [31], [32], [33]



Εικόνα 10: Teensy Microcontroller a.



Εικόνα 9: Teensy Microcontroller b.

2. FSR 402, 0.5"

- Πολύ λεπτός
- Εύκαμπτος και ελαφρύς
- Κατάλληλος για επαφή με το σώμα σε wearable.
- Ανθεκτικός για πολύωρη χρήση
- Διαθέσιμος σε σταθερή ποιότητα και χαμηλό κόστος. [34], [21], [35]



Εικόνα 11: FSR Sensor

3. MicroSD Module DFR0229

- Σταθερή λειτουργία με Teensy
- Εύκολη ενσωμάτωση μέσω SPI
- Πολύ μικρές διαστάσεις
- Αξιόπιστη λειτουργία για συνεχή εγγραφή [36]



Εικόνα 12: SD Card Module

4. MicroSD 16GB

- Επαρκής χωρητικότητα για μακροχρόνια αποθήκευση δεδομένων αισθητήρων
- Προτιμάται έναντι EEPROM/FRAM λόγω κόστους/χωρητικότητας



Εικόνα 13: SD Card 16GB

5. RTC DS3231

- Πολύ ακριβές ρολόι ($\pm 2\text{ppm}$)
- Περιλαμβάνει θερμοκρασιακή αντιστάθμιση και υποστήριξη μπαταρίας για αυτόνομη λειτουργία
- Πολύ πιο ακριβές από π.χ. DS1307. [37], [38]



Εικόνα 15: Μπροστά όψη RTC DS3231



Εικόνα 14: Πίσω όψη RTC DS3231

6. CR2032

- Παρέχει αυτονομία μηνών στο RTC χωρίς εξωτερική τροφοδοσία
- Μικρό και ελαφρύ
- Πληροί προδιαγραφές για μικρές φορητές συσκευές [39].



Εικόνα 16: Coin Cell Battery CR2032

7. Li-Po Battery LP502248 (3.7V, 450 mAh)

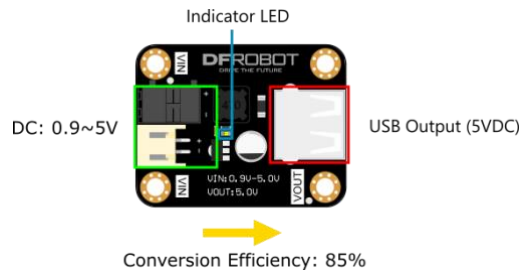
- Ικανοποιητική αυτονομία για ιατρική χρήση ημερήσιας διάρκειας
- Πολύ ελαφριά και λεπτή (5mm πάχος)
- Κατάλληλη για φορητές συσκευές [40]



Εικόνα 17: Li-Po Battery

8. Boost Module FIT0471

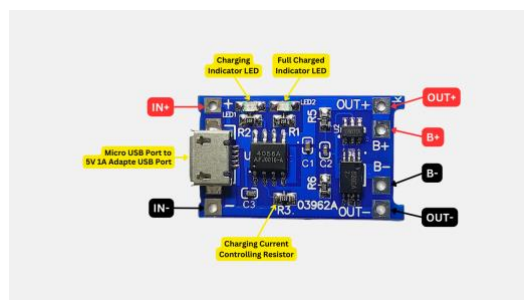
- Μετατροπέας συνεχούς τάσης σε συνεχή με τρανζίστορ ισχύος. Επιτυγχάνει αξιόπιστη μετατροπή $3.7V \rightarrow 5V$ για την τροφοδοσία του Teensy και περιφερειακών
- Μικρό και ελαφρύ
- Αποδοτικότερο σε φορητές συσκευές. [41], [42]



Εικόνα 18: Boost Module FIT0471

9. Charging Module TP4056

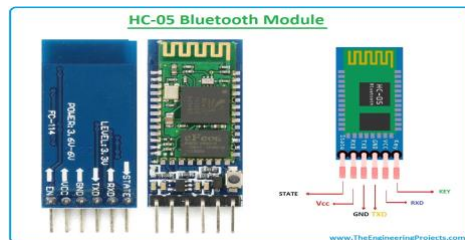
- Μικρός φορτιστής Li-Po μέσω USB
- Παρέχει προστασίες (overcharge/discharge), απαραίτητο για ασφάλεια σε wearable
- Καθιερωμένος στη DIY και wearable κοινότητα [43], [44]



Εικόνα 19: Charging Module TP4056

10. HC-05 Bluetooth Module

- Συμβατό με Arduino/Teensy
- Πολύ διαδεδομένο
- Σταθερή λειτουργία σε χαμηλή κατανάλωση
- Εύκολη σύνδεση με Android (MIT App Inventor) [45], [46]



Εικόνα 20: Bluetooth Module HC-05

5.3 Συνδεσμολογία Υλικών και Πληροφορίες για τον Τρόπο Υλοποίησης και Λειτουργίας

5.3.1 A) Bluetooth module

Είναι σημαντικό να διευκρινίσουμε πως το πρώτο από τα υλικά που πρέπει να συνδεθεί με τον μικροελεγκτή, είναι το Bluetooth module για την σύνδεση της συσκευής που θα δημιουργηθεί, με το κινητό τηλέφωνο. Μέσω των δοκιμών που υλοποιήθηκαν αντιμετωπίστηκε πρόβλημα όταν η συνδεσμολογία ξεκίνησε με του άλλους αισθητήρες και τα modules που χρησιμοποιήθηκαν. Επίσης είναι σημαντικό να αναφέρουμε τις ιδιαιτερότητες του συγκεκριμένου module HC-05 [47].

Αρχικά, μπορεί να βρίσκεται σε δύο mode, είτε σε **Data Mode**, είτε σε **AT mode**.

Πιο αναλυτικά:

1. Data Mode (Λειτουργία Δεδομένων)

- **Χρήση:** Κανονική λειτουργία για αποστολή/λήψη δεδομένων μέσω Bluetooth.
- **Περιγραφή:** Το HC-05 λειτουργεί για σειριακή επικοινωνία μεταξύ δύο συσκευών (π.χ. teensy microcontroller ↔ κινητό).

- **Κατάσταση LED:** Αναβοσβήνει γρήγορα (συνήθως 2 φορές/δευτερόλεπτο όταν δεν είναι συνδεδεμένο και σταθερό όταν συνδεθεί).
- **Παράδειγμα χρήσης:** Στέλνονται μετρήσεις από τον teensy στο MIT App Inventor και ό,τι στέλνεται από την μία συσκευή εμφανίζεται στην άλλη

2. AT Mode (Command Mode / Λειτουργία Εντολών)

- **Χρήση:** Ρύθμιση παραμέτρων του HC-05 (π.χ. όνομα, ταχύτητα baud, PIN, master/slave mode).
- **Περιγραφή:** Το HC-05 **δεν στέλνει δεδομένα**, αλλά **δέχεται εντολές AT**, π.χ. AT+NAME=Scoliosis, AT+BAUD=9600.

Πίνακας AT εντολών:

Εντολή AT	Περιγραφή	Απάντηση
AT	Έλεγχος σύνδεσης με το module	OK
AT+VERSION?	Έκδοση firmware	π.χ. +VERSION:2.0-20100601
AT+ROLE?	Ρόλος συσκευής (0=Slave, 1=Master, 2=Slave-Loop) ¹	+ROLE:0
AT+ROLE=x	Ορισμός ρόλου (x=0/1/2)	OK
AT+NAME?	Προβολή ονόματος συσκευής	+NAME:HC-05
AT+NAME=όνομα	Ορισμός ονόματος Bluetooth	OK ²
AT+UART?	Προβολή baud rate, stop bits, parity	+UART:9600,0,0
AT+UART=baud,stop,parity	Ορισμός σειριακών παραμέτρων	OK
AT+PSWD?	Προβολή PIN σύνδεσης	+PSWD:1234
AT+PSWD=xxxx	Αλλαγή PIN	OK ³
AT+ADDR?	Εμφανίζει τη Bluetooth MAC διεύθυνση	π.χ. +ADDR:98D3:31:123456

¹ Χρειαζόμαστε slave, σε περίπτωση που έρθει σε master το μετατρέπουμε

² Στη συγκεκριμένη πτυχιακή εργασία δόθηκε το όνομα **Scoliosis**

³ Παρέμεινε το ίδιο, για ευκολία σύνδεσης, αλλά χρειάζεται μόνο ,μια φορά να εισαχθεί κατά τον προγραμματισμό του

AT+RESET	Κάνει επανεκκίνηση του module	OK
AT+ORGL	Επαναφέρει τις εργοστασιακές ρυθμίσεις	OK
AT+CMODE?	Σύνδεση με οποιαδήποτε συσκευή (0=μόνο paired, 1=οποιαδήποτε)	+CMODE:0
AT+CMODE=1	Επιτρέπει σύνδεση με οποιαδήποτε διεύθυνση	OK
AT+BIND?	Εμφανίζει δεσμευμένη διεύθυνση για σύνδεση	+BIND:98D3:31:XXX XXX
AT+BIND=addr⁴	Ορισμός συγκεκριμένης συσκευής για σύνδεση ⁵	OK
AT+INIT	Ενεργοποιεί SPP profile (αν χρειάζεται)	OK ή ERROR:(17) αν ήδη ενεργό
AT+INQ	Κάνει αναζήτηση για άλλες Bluetooth συσκευές	λίστα συσκευών
AT+PAIR=addr,timeout⁶	Κάνει ζεύξη με άλλη συσκευή	OK
AT+LINK=addr	Σύνδεση με paired συσκευή	OK

Table 2: Πίνακας AT Εντολών

Πολλές από αυτές τις λειτουργίες χρησιμοποιήθηκαν για δοκιμές και ελέγχους

- **Κατάσταση LED:** Αναβοσβήνει **αργά** (περίπου 1 φορά ανά 2 δευτερόλεπτα).
- **Ενεργοποίηση:**
 - ο **Τρόπος 1** (με button module): Κρατιέται πατημένο το κουμπί του HC-05 πριν την τροφοδοσία.
 - ο **Τρόπος 2** (χωρίς button): Εισάγεται το pin "Key" ή "EN" στο HIGH πριν δοθεί ρεύμα.

⁴ Όπου addr η διεύθυνση (address) της συσκευής που θέλουμε

⁵ Δεν χρησιμοποιήθηκε

⁶ Όπου timeout ο χρόνος που εισάγουμε ως διακοπή

Συνοπτικά:

Λειτουργία	Χρήση	LED	Επικοινωνία
Data Mode	Κανονική ανταλλαγή δεδομένων, για σύζευξη (pairing)	Γρήγορο άναμμα-σβήσιμο ή σταθερό	Serial (π.χ. Teensy ↔ App)
AT Mode	Ρυθμίσεις module	Αργό άναμμα-σβήσιμο	Μόνο εντολές AT

Table 3: Συνοπτικός Πίνακας Data/ AT mode

Συνδεσμολογία Bluetooth module, HC-05 [48], [49], [50] με Teensy microcontroller [32],

Παρακάτω φαίνονται οι ακροδέκτες που διαθέτει το module HC-05, καθώς και σε ποια σημεία του microcontroller Teensy, πρέπει να συνδεθούν

Bluetooth module, HC-05	Σημασία	Teensy
EN (ENABLE)	Αυτή η ακίδα χρησιμοποιείται για τον έλεγχο της κατάστασης ισχύος της μονάδας, δηλαδή για την εναλλαγή μεταξύ της Λειτουργίας Δεδομένων (ρύθμιση χαμηλής) και της λειτουργίας εντολής AT (ρύθμιση υψηλής). Από προεπιλογή, βρίσκεται στη Λειτουργία Δεδομένων.	- ⁷
VCC	Τροφοδοσία μονάδας	+5V
GND	Ακροδέκτης γείωσης της μονάδας, σύνδεση στη γείωση του συστήματος.	GND
TXD	Μεταδίδει σειριακά δεδομένα. Όλα τα δεδομένα που λαμβάνονται μέσω Bluetooth θα δίνονται από αυτόν τον ακροδέκτη ως σειριακά δεδομένα. (Πομπός)	RX1 (pin 0) or RX2 (pin 7) or RX3 (pin 15) or RX4 (pin16) or RX5 (pin 21)

⁷ Χρησιμοποιήθηκε μόνο για τις δοκιμές αλλά δεν χρειάζεται να μείνει συνδεδεμένο με τον μικροελεγκτή

RXD	Λήψη σειριακών δεδομένων. Κάθε σειριακό δεδομένο που δίνεται σε αυτόν τον ακροδέκτη θα μεταδίδεται μέσω Bluetooth. (Δέκτης)	TX1 (pin 1) or TX2 (pin 8) or TX3 (pin14) or TX4 (pin 17) or TX5 (pin 20)
STATE	Η ακίδα κατάστασης είναι συνδεδεμένη με την ενσωματωμένη λυχνία LED και μπορεί να χρησιμοποιηθεί ως ανατροφοδότηση για να ελεγχθεί εάν το Bluetooth λειτουργεί σωστά.	- ⁸

Table 4: Συνδεσμολογία Bluetooth Module HC-05 με Teensy Microcontroller

Όσο αφορά το pin RXD:

Η πλακέτα break-out HC-05 διαθέτει έναν ρυθμιστή 3.3V που επιτρέπει τάση εισόδου από 3.6V έως 6V, αλλά οι ακροδέκτες TX και RX εξακολουθούν να είναι 3.3V. Αυτό σημαίνει ότι μπορεί να χρησιμοποιηθεί η έξοδος 5V από το Teensy για να τροφοδοτηθεί η πλακέτα, αλλά δεν μπορεί να συνδεθεί ο Teensy απευθείας στον ακροδέκτη RX HC-05. Αυτό σημαίνει πως ο ακροδέκτης RX της μονάδας HC-05, δεν είναι ανεκτικός στα 5V (λειτουργεί στα 3,3V). Επομένως, απαιτείται **διαιρέτης τάσης** για την μείωση του σήματος στα ασφαλή 3,3V. [51]

Για τον διαιρέτη τάσης χρησιμοποιήθηκαν δύο αντιστάσεις 1kΩ και 2kΩ

- $R1 = 1K$, $R2 = 2K$
- $1K + 2K = 3K$.
- Το 1K είναι το $\frac{1}{3}$ του 3K, επομένως μειώνει την τάση κατά το $\frac{1}{3}$.
- Το $\frac{1}{3}$ των 5V είναι 1,66 και $5 - 1,66 = 3,33$, που είναι το επιθυμητό. Αν τοποθετήσουμε τις αντιστάσεις από την άλλη πλευρά, η τάση θα μειωθεί κατά $\frac{2}{3}$.

Ποιο αναλυτικά, έχουμε κοινό ρεύμα το οποίο δίνεται από τον τύπο:

$$I = \frac{V_{in}}{R1 + R2}$$

⁸ Χρησιμοποιήθηκε μόνο για τις δοκιμές, δεν χρειάζεται να είναι συνδεδεμένο

Ενώ το V_{out} που έχουμε από τις δύο αυτές αντιστάσεις είναι ίσο με:

$$V_{out} = I * R2 = \frac{R2}{R1+R2} * V_{in} \quad ^9$$

Ο κανόνας του διαιρέτη τάσης είναι μια θεμελιώδης αρχή στην ανάλυση κυκλώματος που υπαγορεύει τον τρόπο με τον οποίο μια τάση εισόδου κατανέμεται αναλογικά στα συνδεδεμένα σε σειρά εξαρτήματα. Αυτός ο κανόνας είναι ιδιαίτερα σημαντικός στα κυκλώματα συνεχούς ρεύματος, όπου η σχέση μεταξύ τάσης, ρεύματος και αντίστασης ακολουθεί τον νόμο του Ohm με απλό τρόπο. Σύμφωνα με τον κανόνα του διαιρέτη τάσης, η πτώση τάσης σε κάθε αντίσταση σε ένα κύκλωμα σειράς είναι ευθέως ανάλογη με την αντίστασή του σε σχέση με τη συνολική αντίσταση του κυκλώματος [52]. Είναι σημαντικό να εξασφαλίζεται ότι το ίδιο ρεύμα διαρρέει τις αντιστάσεις. Αυτό επιτυγχάνεται εάν η αντίσταση στην οποία «βλέπει» η έξοδος V_{out} είναι πολύ μεγάλη (ιδανικά άπειρη). Στη συγκεκριμένη περίπτωση αυτό επιτυγχάνεται από τον ακροδέκτη RX του κυκλώματος HC-05, ο οποίος εσωτερικά συνδέεται με κύκλωμα πολύ μεγάλης αντίστασης. Διαφορετικά θα έπρεπε η έξοδος V_{out} του διαιρέτη να συνδεθεί με κύκλωμα τελεστικού ενισχυτή emitter-follower (το οποίο προσφέρει πολύ μεγάλη αντίσταση εισόδου) και στη συνέχεια να συνδεθεί η έξοδος του emitter-follower με το κύκλωμα το οποίο θέλουμε να τροφοδοτήσουμε με 3.3V.

Έτσι, εν συνεχεία της παραπάνω συνδεσμολογίας και του διαιρέτη τάσης ακολουθούμε τα εξής:

- Το ελεύθερο άκρο της $R1 = 1K$ αντίστασης, συνδέεται στον ακροδέκτη TX του Teensy
- Το ελεύθερο άκρο της $R2 = 2K$ αντίστασης, συνδέεται στο GND του Teensy
- Η ένωση των δύο αντιστάσεων (όπου συναντώνται οι αντιστάσεις $R1 = 1K$ και $R2 = 2K$) συνδέεται με τον ακροδέκτη RX του HC-05 [51]

⁹ $R1, R2$ είναι συνδεδεμένες σε σειρά, εφ' όσον ανήκουν στον ίδιο κλάδο και έχουν κοινό άκρο

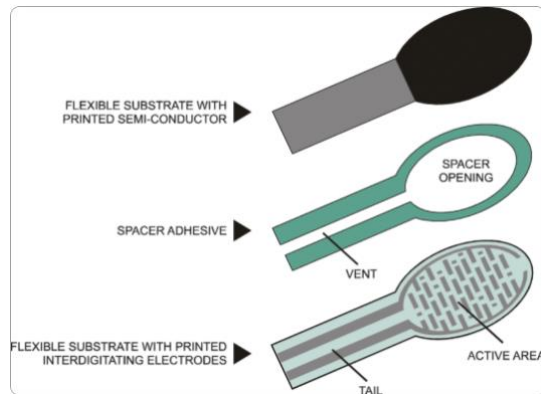
Επίσης είναι σημαντικό να διευκρινιστεί πως ο μικροελεγκτή Teensy 4.0 [32], σε αντίθεση με άλλους (πχ. Arduino) έχει πολλαπλές θύρες UART και γι' αυτόν τον λόγο μπορούμε να κάνουμε απευθείας σύνδεση με τα RX, TX χωρίς να χρειάζεται να ορίσουμε μέσα από τον κώδικα τα pins που θα χρησιμοποιήσουμε γι' αυτή την διαδικασία.

Αν, για παράδειγμα, αναφερόμασταν στο Arduino, θα έπρεπε να χρησιμοποιήσουμε στον κώδικα τη συνάρτηση *SoftwareSerial BTSerial(2, 3)* (η οποία ανήκει στη βιβλιοθήκη *< SoftwareSerial.h >*) [53], διότι στο Arduino τα pins 0 (RX) και 1 (TX) είναι συνδεδεμένα εσωτερικά με το USB-to-Serial chip που επικοινωνεί με το **Serial Monitor** του υπολογιστή.

Έτσι, αν συνδέαμε απευθείας το RX και TX του HC-05 στις ψηφιακές εισόδους 0 και 1 του Arduino, θα υπήρχε σύγκρουση: και ο υπολογιστής (μέσω USB) και το HC-05 θα προσπαθούσαν να χρησιμοποιήσουν την ίδια σειριακή γραμμή. Γι' αυτόν τον λόγο, χρησιμοποιούμε το *SoftwareSerial BTSerial(2, 3)*, ώστε το Arduino να επικοινωνεί με το HC-05 μέσω των pins 2 και 3, **χωρίς να επηρεάζεται** η σύνδεση του USB Serial Monitor. [54]

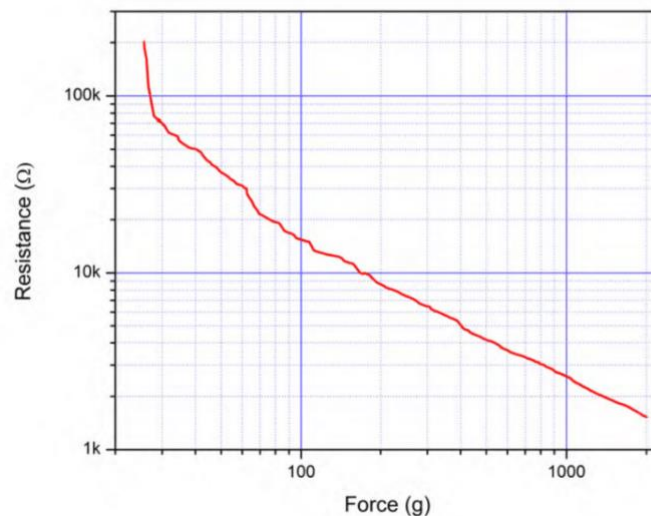
5.3.2 B) Force Sensors (FSR)

Ένας FSR αποτελείται από δύο μεμβράνες και μια κόλλα διαχωρισμού. Οι αγωγίμες μεμβράνες διαχωρίζονται από ένα λεπτό κενό αέρα όταν δεν εφαρμόζεται πίεση. Μία από τις μεμβράνες περιέχει δύο ίχνη που εκτείνονται από την άκρη προς την περιοχή ανίχνευσης (το στρογγυλό μέρος). Αυτά τα ίχνη είναι υφασμένα μεταξύ τους, αλλά δεν εφάπτονται. Η άλλη μεμβράνη είναι επικαλυμμένη με αγωγίμο μελάνι. Όταν πιέζεται ο αισθητήρας, το μελάνι βραχυκυκλώνει τα δύο ίχνη μεταξύ τους με μια αντίσταση που εξαρτάται από την πίεση.



Εικόνα 21: Κατασκευή FSR

Το παρακάτω γράφημα εμφανίζει την καμπύλη αντίστασης έναντι δύναμης για τον αισθητήρα FSR 402. Σημειώστε ότι τα δεδομένα απεικονίζονται σε λογαριθμικές κλίμακες. Η **απόκριση δεν είναι γραμμική**. Υπάρχει μια τεράστια πτώση στην αντίσταση όταν εφαρμόζεται μια μικρή ποσότητα πίεσης. Μετά από αυτό, η αντίσταση είναι αντιστρόφως ανάλογη με την εφαρμοζόμενη δύναμη. Στα περίπου 10 kg (δεν φαίνεται στο γράφημα) ο αισθητήρας είναι κορεσμένος και μια αύξηση της δύναμης αποφέρει μικρή έως καθόλου μείωση της αντίστασης. [55]



Εικόνα 22: Καμπύλη Αντίστασης έναντι Δύναμης για FSR

Όσο αφορά τους αισθητήρες δύναμης λοιπόν, δημιουργούμε κλειστά κυκλώματα **με διαιρέτη τάσης**. Αυτό γίνεται διότι, η χρήση κυκλώματος **διαιρέτη τάσης** για αισθητήρες δύναμης όπως ο **FSR (Force Sensitive Resistor)** είναι απαραίτητη, αφού οι FSR **δεν**

παράγουν τάση από μόνοι τους, αλλά μεταβάλλουν την αντίστασή τους ανάλογα με τη δύναμη που τους ασκείται. Επομένως, για να πάρουμε αναγνώσιμη τιμή (τάση) από έναν μικροελεγκτή (όπως ο Teensy που χρησιμοποιείται στην προκειμένη περίπτωση), πρέπει να μετατρέψουμε αυτή την αντίσταση σε τάση. [56]

Ο διαιρέτης τάσης αποτελείται από **δύο αντιστάσεις σε σειρά** μεταξύ Vcc και GND. Η μέτρηση της τάσης γίνεται στο σημείο ανάμεσα στις δύο αντιστάσεις. Όταν η μία από τις δύο αντιστάσεις είναι ο **FSR**, τότε καθώς μεταβάλλεται η δύναμη, μεταβάλλεται η αντίσταση, αλλάζει η τάση στο μεσαίο σημείο και άρα ο μικροελεγκτής διαβάζει διαφορετική αναλογική τιμή.

Η εξίσωση εξόδου του **FSR** είναι:

$$V_{out} = \frac{V}{1 + R_{FSR}/R_M}$$

και η τάση αυξάνεται με τη δύναμη [56]

Όταν χρησιμοποιείται διαιρέτης τάσης, όπως στην προκειμένη περίπτωση, το Vout είναι:

$$V_{out} = \frac{V_{cc} \cdot R_0}{(R_0 + R_s)}$$

Όπου, Vcc = 3,3V και συνιστώνται τιμές για γραμμικότητα, π.χ. $R_0 \approx \frac{1}{3} - \frac{1}{2}$ της RFSR

και έτσι,

$$V_{out} = \frac{3,3V \cdot R_{pulldown}}{(R_{FSR} + R_{pulldown})}$$

- $R_{pulldown} = 10K\Omega$ [34]
- $R_{FSR} \rightarrow$ μειώνεται όσο αυξάνεται η δύναμη [57], δηλαδή, η τάση είναι ανάλογη του **αντίστροφου** της αντίστασης FSR.

Ο FSR αλλάζει την αντίστασή του, ανάλογα την δύναμη που του ασκείται

Ο FSR επίσης, έχει **2 ακροδέκτες** (είναι απλός μεταβλητός αντιστάτης):

- Ένας ακροδέκτης συνδέεται στο **Vcc (ή GND)**
- Ο άλλος ακροδέκτης συμμετέχει στον **διαίρετη τάσης**, συνδεόμενος με σταθερή αντίσταση (π.χ. 10kΩ) προς GND (ή Vcc, αντίστροφα)
- Το **σημείο ανάμεσα στον FSR και την αντίσταση** πάει στην **αναλογική είσοδο** του μικροελεγκτή.

Αισθητήρας	Ακροδέκτης 1	Ακροδέκτης 2	Ελεύθερο άκρο Αντίστασης	Ακροδέκτης 2
FSR1	Vcc	10KΩ	GND	Αναλογική είσοδος A0 (pin 14)
FSR2	Vcc	10KΩ	GND	Αναλογική είσοδος A1 (pin 15)
FSR3	Vcc	10KΩ	GND	Αναλογική είσοδος A2 (pin 16)

Table 5: Συνδεσμολογία FSR με Teensy Microcontroller

Χρησιμοποιήθηκε 10KΩ αντίσταση διότι:

1. Συνήθως το FSR έχει αντίσταση από 1MΩ (χωρίς πίεση) έως 1kΩ (με μεγάλη πίεση)
2. Υπάρχουν οι εξής συνδυασμοί με τα παρακάτω αποτελέσματα:
 - Αν επιλέγαμε πολύ μικρή τιμή (πχ 1kΩ αντίσταση), τότε χωρίς πίεση το Vout θα ήταν πολύ μικρό (σχεδόν 0V) και με πίεση το Vout δεν θα πλησίαζε το 3.3V (στενό δυναμικό εύρος)
 - Αν βάζαμε πολύ μεγάλη τιμή (πχ 100kΩ αντίσταση), τότε ακόμα και με μεγάλη πίεση το Vout θα έμενε πολύ κοντά στα 3.3V (και πάλι περιορισμένο εύρος)
 - Ενώ στα 10kΩ:

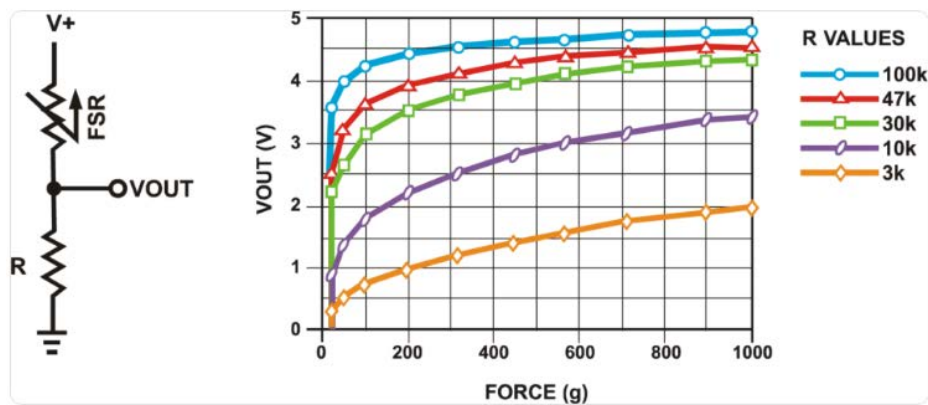
- ο $R_{FSR} \rightarrow \infty$, $V_{out} = 3.3\text{ V} * \frac{10\text{ k}\Omega}{\infty + 10\text{ k}\Omega} \approx 0\text{ V}$, $V_{out} \rightarrow 0$ (χωρίς πίεση)
- ο $R_{FSR} \approx 1\text{ k}\Omega \rightarrow V_{out} \approx 3.3\text{ V} * \frac{10000}{(1000 + 10000)} \approx 3.0\text{ V}$ (χωρίς πίεση)

Με αυτόν τον τρόπο καλύπτεται όλο το εύρος του ADC (0 – 3.3V) σχεδόν πλήρως. [58], [59], [60]

Συνοπτικά :

Σενάριο	FSR χωρίς πίεση ($\approx \infty\Omega$)	FSR μεγάλη πίεση ($\approx 1\text{ k}\Omega$)	Παρατηρήσεις
Rfixed = 1 kΩ	$V_{out} \approx 0\text{ V}$	$3.3\text{ V} \times 1 / (1 + 1) = 1.65\text{ V}$	Στενό εύρος
Rfixed = 10 kΩ	$V_{out} \approx 0\text{ V}$	$3.3\text{ V} \times 10 / (1 + 10) \approx 3.0\text{ V}$	Καλό εύρος
Rfixed = 100 kΩ	$V_{out} \approx 0\text{ V}$	$3.3\text{ V} \times 100 / (1 + 100) \approx 3.27\text{ V}$	Πολύ κοντά στο V_{cc}

Table 6: Επιρροή σταθερής αντίστασης στο V_{out}

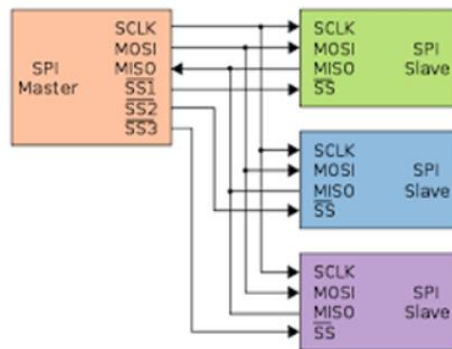


Εικόνα 23: Κύκλωμα διαχωριστή τάσης και καμπύλες V_{out} vs $Force$ για διαφορετικές τιμές R , όπου g σημαίνει γραμμάρια.

5.3.3 C) SD Card Module

Το SD card module αποτελεί ένα περιφερειακό αποθήκευσης δεδομένων, το οποίο επιτρέπει την εγγραφή και ανάγνωση ψηφιακών πληροφοριών σε κάρτες τύπου microSD. Η επικοινωνία με τον μικροελεγκτή (όπως π.χ. Arduino ή Teensy) επιτυγχάνεται μέσω του σειριακού πρωτοκόλλου SPI (Serial Peripheral Interface), αξιοποιώντας γραμμές όπως MOSI (Master Out Slave In), MISO (Master In Slave Out), SCK (Serial Clock) και CS (Chip Select). Το SPI πρωτόκολλο επιτρέπει στον μικροελεγκτή να επικοινωνεί σειριακά και με τρόπο σύγχρονο με τις περιφερειακές συσκευές, που σημαίνει ότι χρησιμοποιεί ξεχωριστές γραμμές για τα δεδομένα και για το σήμα ρολογιού, με αποτέλεσμα οι SPI συσκευές να έχουν τέλειο συγχρονισμό.

Σε ένα SPI δίκτυο, συνήθως υπάρχει ένας SPI master ενώ μπορούν να υπάρχουν πολλαπλοί SPI slaves συσκευές, που με μια γραμμή επιλογής, ο master επιλέγει εκείνη που θέλει να επικοινωνήσει.



Εικόνα 24: Master-Slaves

1] Serial clock (SCK). Το SCK σήμα ρολογιού είναι έξοδος από τον master και είσοδος για τον slave. Χρησιμοποιείται για το συγχρονισμό της μεταφοράς δεδομένων μεταξύ master και slave, που γίνονται πάνω στις γραμμές MOSI και MISO. Το σήμα ρολογιού παράγεται από τον master και αγνοείται από τους slaves που δεν έχουν ενεργοποιηθεί (επιλεγεί).

2] Master in, slave out (MISO). Αυτό το σήμα είναι έξοδος από τον slave και είσοδος για τον master. Χρησιμοποιείται για τη μεταφορά δεδομένων σειριακά από τον slave στον master. Ο ακροδέκτης MISO τίθεται σε κατάσταση υψηλής αντίστασης, όταν η SPI συσκευή slave δεν είναι ενεργοποιημένη (επιλεγμένη).

3] Master out, slave in (MOSI). Αυτό το σήμα είναι έξοδος από τον master και είσοδος για τους slaves συσκευές. Χρησιμοποιείται για τη μεταφορά δεδομένων σειριακά από τον master στο slave.

4] Slave select (SS). Καθώς μια SPI συσκευή διαμορφώνεται ως slave, αυτός ο ακροδέκτης είναι πάντα είσοδος, το οποίο επιτρέπει την επιλογή του επιθυμητού slave. Το σήμα SS δεν ελέγχεται από το πρωτόκολλο SPI και θα πρέπει να γράψουμε ξεχωριστό κώδικα για αυτό. Όταν η είσοδος SS της SPI slave συσκευής οδηγείται σε κατάσταση high ο slave απενεργοποιείται. Ο επιθυμητός SPI slave ενεργοποιείται και γίνεται μεταφορά δεδομένα μ' αυτόν, όταν ο ακροδέκτης εισόδου SS οδηγείται σε κατάσταση low. Ο ακροδέκτης Slave select (SS) στον master πρέπει να διαμορφώνεται ως έξοδος, αφού επιλέγει τον slave για SPI επικοινωνία και αν έχουμε πολλαπλούς slaves θα πρέπει να υπάρχουν πολλές γραμμές SS, οι οποίες ελέγχονται από τον κώδικα που θα αναλυθεί στην πορεία. [61]

Μέσω της κατάλληλης βιβλιοθήκης (π.χ. SD.h ή SdFat), ο μικροελεγκτής μπορεί να δημιουργεί, διαβάζει και επεξεργάζεται αρχεία στο αποθηκευτικό μέσο, όπως αρχεία τύπου .txt ή .csv. Η χρήση τέτοιων modules είναι ιδιαίτερα διαδεδομένη σε εφαρμογές καταγραφής δεδομένων αισθητήρων (data logging), όπου απαιτείται αποθήκευση με δυνατότητα ανάγνωσης εκτός συστήματος (π.χ. μέσω υπολογιστή).

Όσο αφορά λοιπόν το SD Module, έχουμε την εξής συνδεσμολογία με τον μικροελεγκτή Teensy 4.0

SD Module	Teensy Microcontroller
MISO	MISO (pin12)
SCK	SCK (pin13)
SS	CS (pin10)
MOSI	MOSI (pin 11)
GND	GND
+5V	3.3V

Table 7: Συνδεσμολογία SD Card Module με Teensy Microcontroller

[62], [63]

Στο SS χρησιμοποιούμε μία αντίσταση 10kΩ της οποίας το ελεύθερο άκρο το συνδέουμε στα 3.3V, ενώ το σημείο στο οποίο συνδέεται το άλλο άκρο αυτής της αντίστασης με το

SS του SD Module το συνδέουμε στο pin 10 (CS) του Teensy, όπως φαίνεται και στον παραπάνω πίνακα.

Ο λόγος που χρησιμοποιείται η συγκεκριμένη αντίσταση είναι για να περιορίζει το ρεύμα και να δημιουργεί πτώση της τάσης προς 3.3 V, προστατεύοντας με αυτόν τον τρόπο την SD κάρτα. [64], [65], [66]

Επίσης η **κάρτα SD** που χρησιμοποιήθηκε έχει χωρητικότητα, **16GB**. Ο λόγος που χρησιμοποιήθηκε κάρτα της συγκεκριμένης χωρητικότητας είναι αρχικά διότι πρέπει να την μετατρέψουμε σε μορφή **FAT16** ή **FAT32** και νεότερες κάρτες όπου έχουν χωρητικότητα μεγαλύτερη ή ίση των **32GB** δεν δίνουν δυνατότητα μετατροπής σε αυτή τη μορφή, αλλά μόνο σε **exFAT** η οποία δεν αναγνωρίζεται από την βιβλιοθήκη **SD.h** που χρησιμοποιήθηκε για τον προγραμματισμό της (όπως θα αναλυθεί και στη συνέχεια). Επίσης εφ' όσον γίνεται λόγος για αρχείο .txt δεν είναι αναγκαίο να έχουμε πολύ μεγάλη χωρητικότητα, αφού το κείμενο σε αυτή τη μορφή καταναλώνει λίγο χώρο και δεν περιέχει εικόνες ή βίντεο τα οποία θα γέμιζαν πολύ γρήγορα την κάρτα δημιουργώντας πρόβλημα. [67], [68], [69], [70]

5.3.4 D) RTC Module

Το RTC module (Real-Time Clock) είναι ένα εξάρτημα υπεύθυνο για την ακριβή διατήρηση της ώρας και της ημερομηνίας σε ηλεκτρονικά συστήματα, ανεξαρτήτως της τροφοδοσίας του μικροελεγκτή. Το μοντέλο **DS3231** αποτελεί μία από τις πιο αξιόπιστες επιλογές, καθώς ενσωματώνει θερμοσταθεροποιημένο ταλαντωτή (temperature-compensated crystal oscillator – TCXO), που εξασφαλίζει υψηλή χρονική ακρίβεια. Η επικοινωνία με τον μικροελεγκτή γίνεται μέσω του διαύλου **I²C**, με τις γραμμές SDA (data) και SCL (clock). Το RTC διατηρεί δεδομένα όπως δευτερόλεπτα, λεπτά, ώρες, ημέρες, μήνες και έτος, ενώ συχνά συνοδεύεται από εφεδρική μπαταρία τύπου **CR2032**, η οποία εξασφαλίζει συνεχή λειτουργία του ρολογιού ακόμη και όταν το υπόλοιπο σύστημα είναι απενεργοποιημένο. Η ακριβής χρονική πληροφορία είναι κρίσιμη σε εφαρμογές όπως χρονοσήμανση δεδομένων (timestamping), αυτόματες ενεργοποιήσεις και ελεγχόμενα χρονοπρογράμματα.

Το RTC Module (DS3231) αποτελείται από τους εξής ακροδέκτες και ακολουθεί την παρακάτω συνδεσμολογία:

RTC Module (DS3231)	Teensy Microcontroller
32K	- ¹⁰
SQW	- ¹¹
SCL	SCL0 (pin 19) or SCL1 (pin 16)
SDA	SDA0 (pin 18) or SDA1 (pin 17)
VCC	3,3 V
GND	GND

Table 8: Συνδεσμολογία RTC Module με Teensy Microcontroller

Επίσης στη θέση/θήκη μπαταρίας βάζουμε την CR2032 coin cell battery, η οποία είναι σημαντικό να επισημάνουμε ότι πρέπει να κρατάει χρόνο ανεξάρτητα, είτε το module τροφοδοτείται από ρεύμα, είτε όχι. [71], [72]

5.3.5 E) Lithium Battery, Boost Module and Charging Module for Lithium Battery

Οι επαναφορτιζόμενες μπαταρίες λιθίου, όπως οι τύποι **Li-ion** και **Li-Po**, χρησιμοποιούνται ευρέως για την παροχή φορητής ενέργειας σε ηλεκτρονικές διατάξεις. Οι συγκεκριμένες μπαταρίες χαρακτηρίζονται από υψηλή ενεργειακή πυκνότητα, χαμηλό βάρος και μεγάλη διάρκεια ζωής. Τυπικά, η ονομαστική τους τάση κυμαίνεται γύρω στα **3.7V**, ενώ η πλήρως φορτισμένη τάση φθάνει τα **4.2V**. Η χρήση τους είναι κρίσιμη σε ασύρματα και φορητά συστήματα, στα οποία η σταθερότητα και η αυτονομία αποτελούν βασικές απαιτήσεις.

Το module φόρτισης μπαταριών, όπως το **TP4056**, αποτελεί ένα ολοκληρωμένο κύκλωμα ελεγχόμενης φόρτισης για μπαταρίες λιθίου. Η τυπική είσοδος του κυκλώματος είναι τάση **5V**, η οποία μπορεί να παρέχεται μέσω υποδοχής **microUSB** ή απευθείας από εξωτερική τροφοδοσία μέσω ακροδεκτών VCC/GND. Ο φορτιστής ρυθμίζει το ρεύμα φόρτισης (συνήθως στα **500mA** ή **1A**) ώστε να προσαρμόζεται στις προδιαγραφές της μπαταρίας. Πολλά μοντέλα TP4056 ενσωματώνουν και κύκλωμα **προστασίας** (Protection Circuit Module – PCM), προσφέροντας προστασία από υπερφόρτιση, υπερβολική

¹⁰ Δεν χρειάζεται για βασική λειτουργία

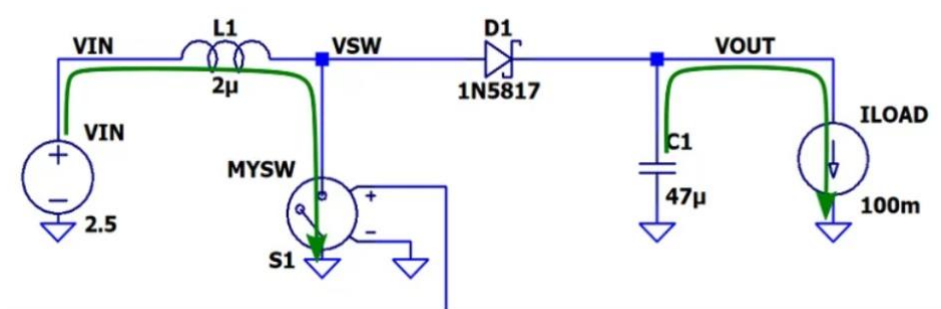
¹¹ Δεν χρειάζεται για βασική λειτουργία

εκφόρτιση και βραχυκύκλωμα. Το κύκλωμα φέρει συνήθως δύο φωτεινές ενδείξεις: **κόκκινη**, που δηλώνει ότι η φόρτιση είναι σε εξέλιξη, και **πράσινη**, που υποδηλώνει ότι η μπαταρία έχει φορτιστεί πλήρως. Η σωστή επιλογή και χρήση τέτοιων modules είναι ζωτικής σημασίας για την ασφαλή λειτουργία και τη μακροχρόνια απόδοση των συστημάτων με μπαταρία.

Όσο αφορά τον μετατροπέα boost (**boost converter module**), έχει δύο βασικές λειτουργικές καταστάσεις, μία όταν ο διακόπτης τροφοδοσίας είναι κλειστός (κατάσταση ενεργοποίησης) και μία όταν είναι ανοιχτός (κατάσταση απενεργοποίησης).

Κατάσταση Ενεργοποίησης

Όταν ο διακόπτης είναι αγωγίμος, το ρεύμα από την τροφοδοσία εισόδου διακλαδώνεται στη γείωση. Ρέει μέσω του επαγωγέα, μέσω του διακόπτη και στον κόμβο γείωσης. Σε αυτή την περίπτωση, δεν διέρχεται από τη διόδο ούτε φτάνει στον πυκνωτή. Κατά τη διάρκεια αυτού του χρόνου, το ρεύμα του επαγωγέα αυξάνεται και ο επαγωγέας «φορτίζεται», και άρα το ενεργειακό περιεχόμενο του ηλεκτρομαγνητικού του πεδίου αυξάνεται.



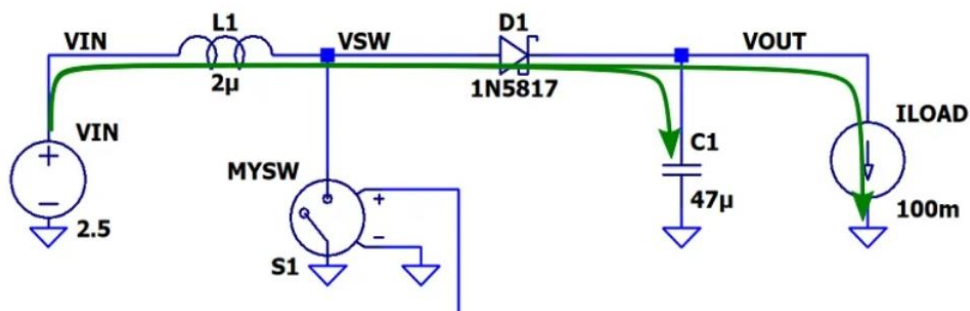
Εικόνα 25: Ένας boost converter στην κατάσταση ενεργοποίησης. Το ρεύμα τροφοδοσίας και το ρεύμα φορτίου υποδεικνύονται και τα δύο με πράσινο χρώμα.

Κατάσταση Απενεργοποίησης

Ρεύμα έρρε μέσα από τον επαγωγέα όταν ο διακόπτης ήταν κλειστός και είναι στη φύση της επαγωγής να αντιστέκεται στις αλλαγές στη ροή του ρεύματος. Επομένως, το ρεύμα του επαγωγέα θα συνεχίσει να ρέει όταν ανοίξει ο διακόπτης, ακολουθώντας τη μόνη λογική διαθέσιμη διαδρομή, τη διόδο.

Κατά τη διαδικασία διοχέτευσης του ρεύματος μέσω της διόδου, ο επαγωγέας πρέπει να αυξήσει την τάση στον κόμβο V_{SW} μέχρι η διόδος να πολωθεί ορθά. Για να συμβεί αυτό, ο επαγωγέας πρέπει να παράγει μια τάση V_{SW} που είναι τουλάχιστον ίση με $(V_{OUT} + V_F)$, όπου V_F υποδηλώνει την ορθή τάση της διόδου. Η τάση στον δεξιό ακροδέκτη του επαγωγέα θα είναι επομένως μεγαλύτερη από V_{OUT} καθώς και μεγαλύτερη από V_{IN} .

Μόλις η τάση στα άκρα της διόδου επαρκεί για αγωγιμότητα, το ρεύμα θα ρέει από την πηγή, μέσω του επαγωγέα και της διόδου, και στον πυκνωτή και το φορτίο. [73]



Εικόνα 26: Ροή ρεύματος μέσω ενός boost converter στην κατάσταση απενεργοποίησης.

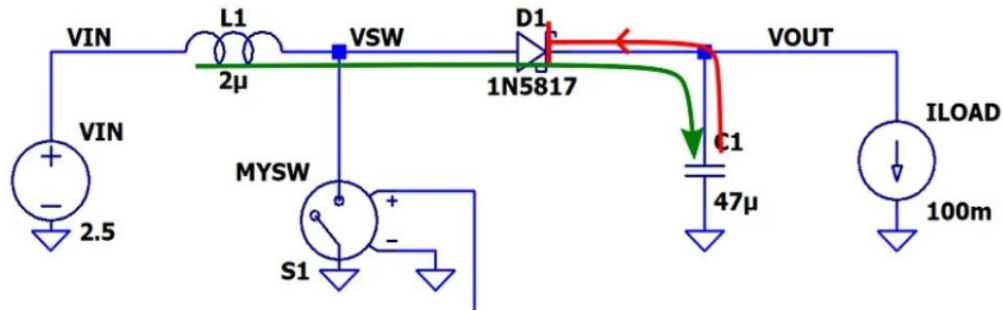
Επίτευξη Boost

Οι πυκνωτές μπορούν να φορτίσουν έως και τάσεις που υπερβαίνουν τις τάσεις τροφοδοσίας σε ένα σύστημα. Αυτό είναι εμφανές από την εξίσωση που συσχετίζει την χωρητικότητα (C), το αποθηκευμένο φορτίο (Q) και την τάση (V) ενός πυκνωτή:

$$V = \frac{Q}{C}$$

Η διόδος του μετατροπέα boost η οποία λειτουργεί ως μονόδρομη βαλβίδα για το ρεύμα, παρέχει και τις δύο αυτές ενέργειες:

- Το ρεύμα του επαγωγέα μπορεί να ρέει μέσω της διόδου και να φορτίζει τον πυκνωτή σε τάσεις πάνω από V_{IN} .
- Ο πυκνωτής δεν μπορεί να εκφορτιστεί πίσω στην πηγή μέσω του επαγωγέα επειδή η διόδος εμποδίζει τη ροή ρεύματος προς αυτή την κατεύθυνση. [73]



Εικόνα 27: Η διόδος επιτρέπει τη ροή ρεύματος από τον επαγωγέα στον πυκνωτή, αλλά όχι αντίστροφα

Για την μπαταρία τροφοδοσίας ακολουθούμε τα εξής:

1. Αρχικά την συνδέουμε με έναν φορτιστή μέσω του οποίου θα επιτυγχάνεται ανατροφοδότηση, όποτε αυτό είναι απαραίτητο:

Lithium Battery	Charging Module for Lithium Battery
Red Cable (+)	B+
Black Cable (-)	B-
-	OUT+
-	OUT-

Table 9: Συνδεσμολογία Lithium Battery με Charging Module for Lithium Battery

2. Έπειτα, και επειδή ο Teensy Microcontroller δέχεται ως Vin 5V, εγ' όσον η μπαταρία που χρησιμοποιήθηκε είναι στα 3.7V θα χρησιμοποιήσουμε boost converter, όπου θα μετατρέπει τα Volt που λαμβάνει ως είσοδο από την μπαταρία Vin(0.9V – 5.0V) σε Vout(5.0V)

Charging Module for Lithium Battery	Boost Module
B+	Vin(+) (ειδική εσοχή για καλώδιο) ¹²
B-	Vin(-) (ειδική εσοχή για καλώδιο)
OUT+	-
OUT-	-

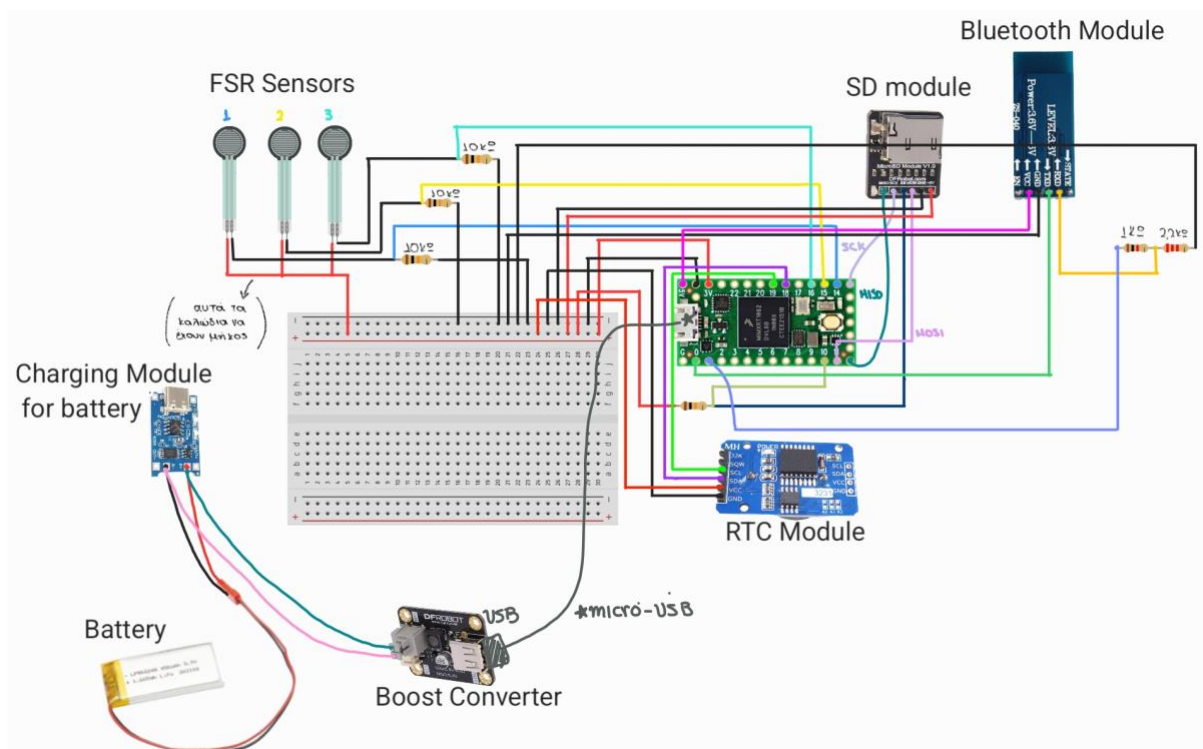
Table 10: Συνδεσμολογία Charging Module for Lithium Batter με Boost Module

¹² Με το πάτημα του γκρι κουμπιού τοποθετείται το καλώδιο στην ειδική εσοχή και μόλις αυτό μπει το κουμπί αφήνεται και έτσι επιτυγχάνεται η σύνδεση

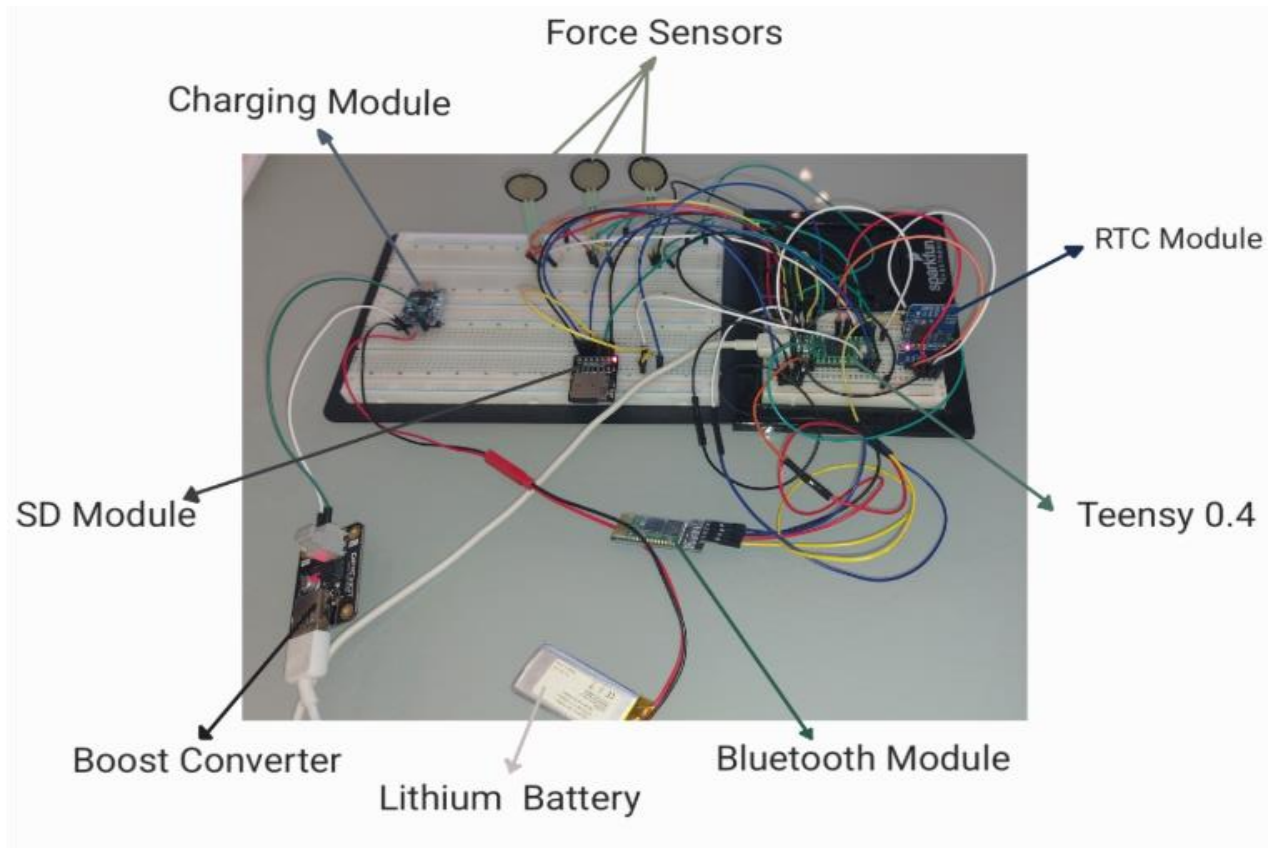
- Τέλος στην θήρα USB του Boost Module τοποθετείται το καλώδιο που έρχεται από τον Teensy (συνδέεται σε αυτόν με micro USB και στην άλλη του άκρη έχει ελεύθερη USB θήρα μέσω της οποίας έγινε και ο προγραμματισμός του με το Arduino IDE)

Με αυτόν τον τρόπο επιτυγχάνεται η συνολική τροφοδοσία του συστήματος και γίνεται wearable και ανεξάρτητο από άλλες συσκευές ή πρίζα [74], [75]

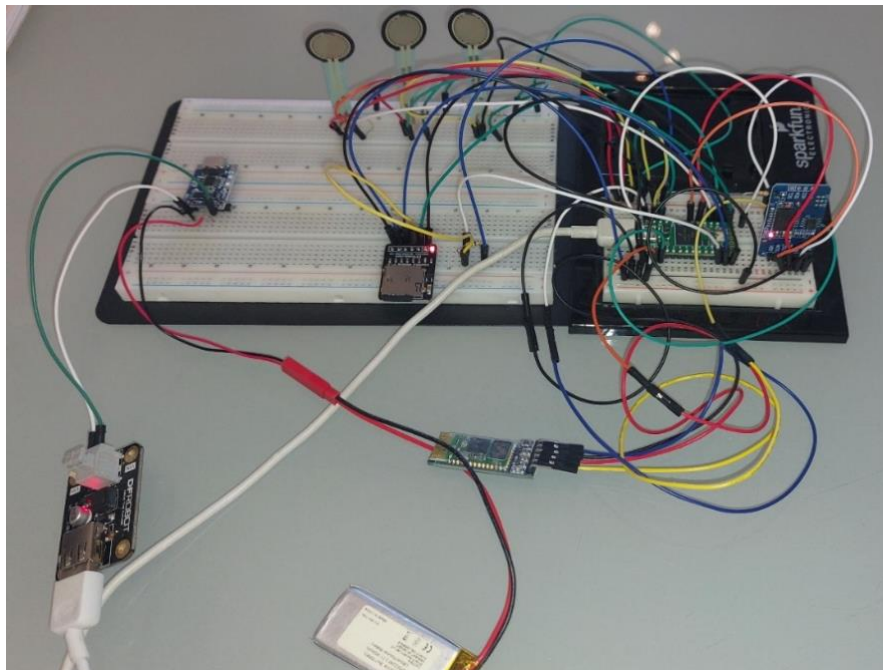
Παρακάτω φαίνεται ο τρόπος συνδεσμολογίας, τόσο αναλυτικά όσο και όπως πραγματοποιήθηκε στο εργαστήριο:



Εικόνα 28: Αναλυτική Συνδεσμολογία



Εικόνα 30: Συνδεσμολογία Υλικών σε breadboard 1



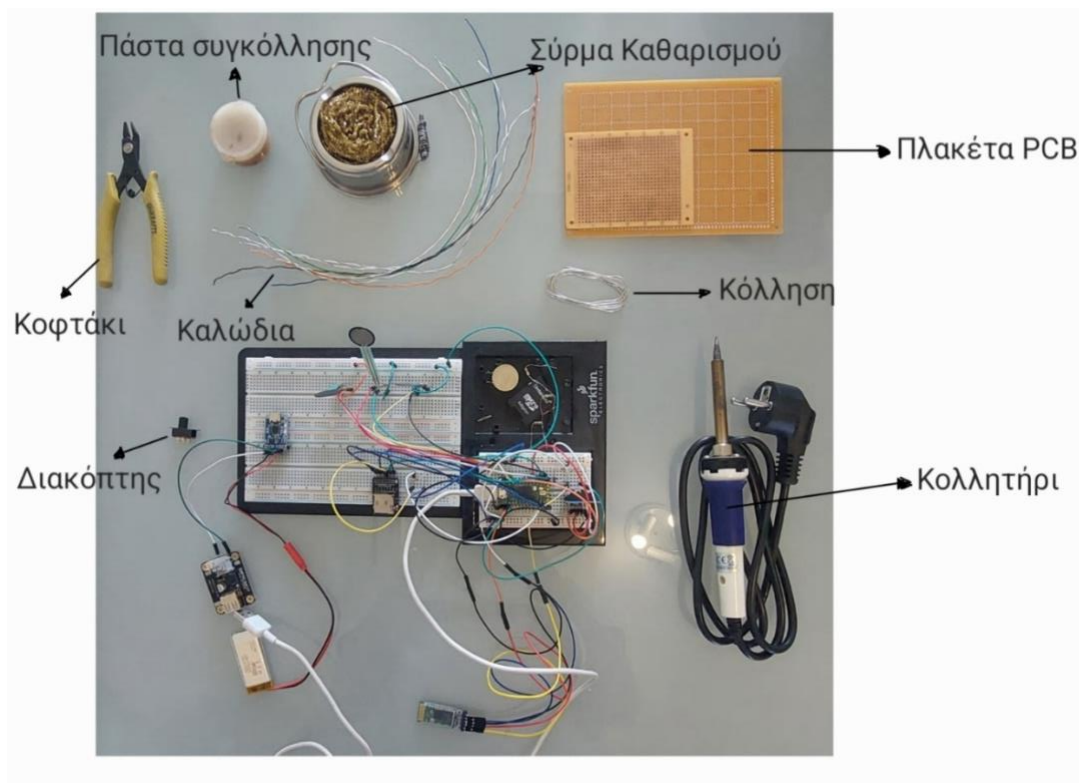
Εικόνα 29: Συνδεσμολογία Υλικών σε breadboard 2

5.4 Σταθεροποίηση Συστήματος Μέσω Συγκόλλησης

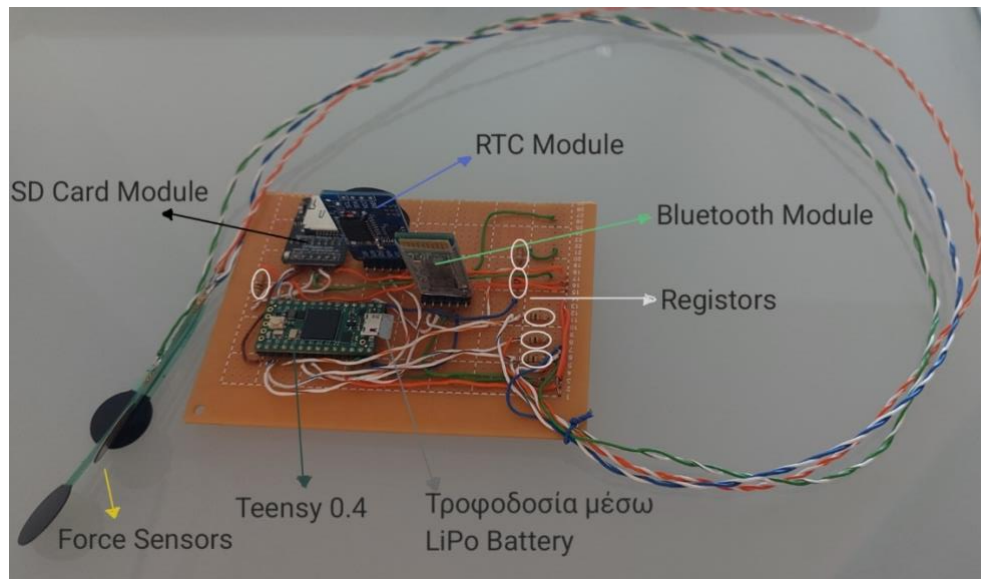
Κάτα την συγκόλληση, ενσωματώθηκε επιπλέον ένας διακόπτης λειτουργίας, ο οποίος επιτρέπει τον άμεσο και αποτελεσματικό έλεγχο της ενεργοποίησης και απενεργοποίησης της συσκευής. Η προσθήκη αυτού του μηχανισμού καθιστά τη διαδικασία χειρισμού πιο εργονομική και φιλική προς τον χρήστη, συμβάλλοντας παράλληλα στη βελτίωση της ενεργειακής απόδοσης και στην παράταση της διάρκειας ζωής του συστήματος.



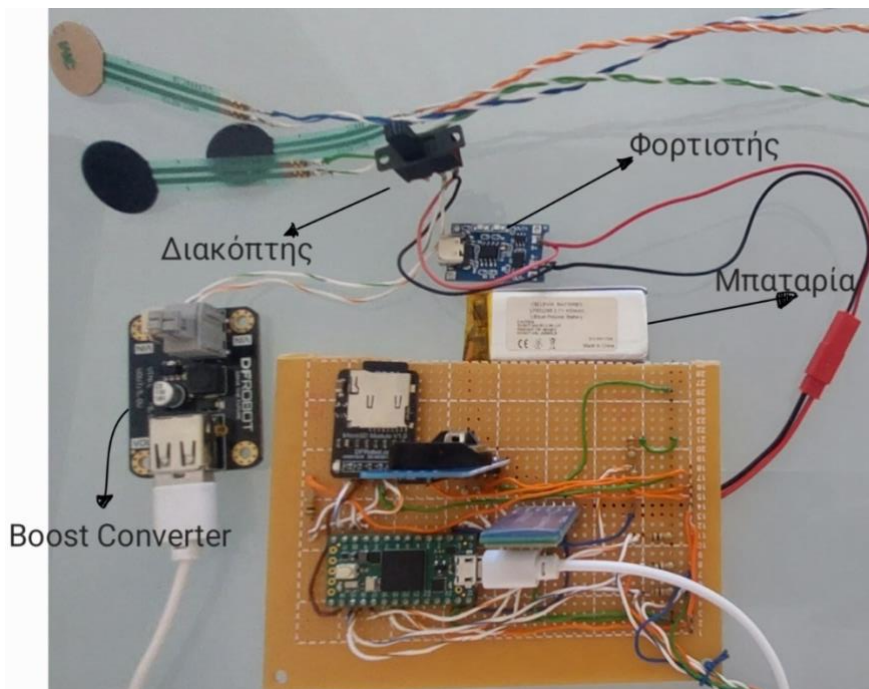
Εικόνα 31: Διακόπτης



Εικόνα 32: Υλικά για Συγκόλληση



Εικόνα 33: Τελικό αποτέλεσμα συγκόλλησης



Εικόνα 34: Τελικό Αποτέλεσμα με διακόπτη, μπαταρία και φορτιστή

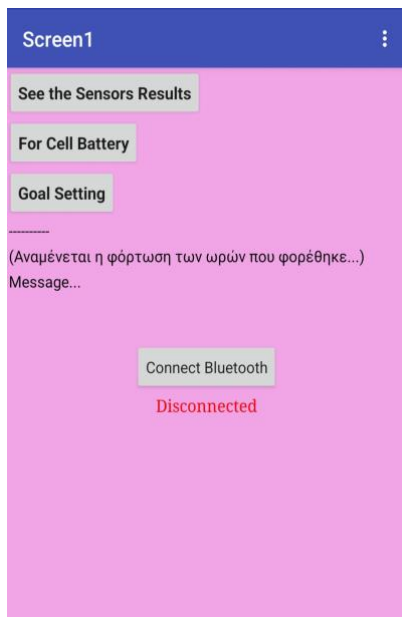
6 Σχεδιασμός και Υλοποίηση Λογισμικού

6.1 Τρόπος Λειτουργίας

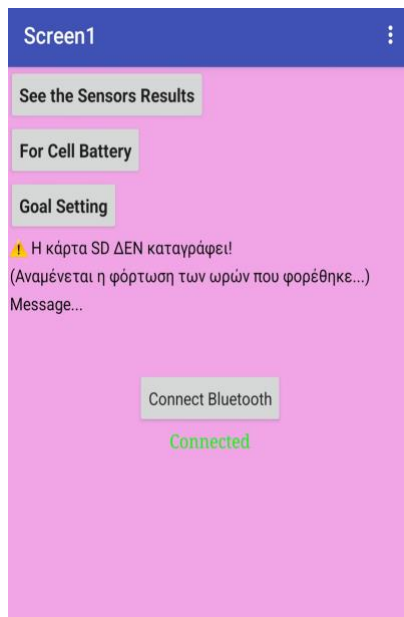
Μέσω της εφαρμογής που δημιουργήθηκε παρέχονται οι παρακάτω δυνατότητες:

- Διαθέτει κουμπί για επιλογή των διαθέσιμων συσκευών bluetooth και σύνδεση με την συσκευή που δημιουργήθηκε.
- Παρουσιάζει τις δυνάμεις, σε μονάδες Newton, που ασκούνται στον καθένα από τους 3 αισθητήρες δύναμης ξεχωριστά.
- Απεικονίζει διάγραμμα που αντιπροσωπεύει την αυξομείωση της εκάστοτε δύναμης, με διαφορετικό χρώμα για τον κάθε αισθητήρα, και επομένως διαγραμματική, παραστατική απεικόνιση των αποτελεσμάτων για πιο εύκολη και γρήγορη παρατήρηση (λαμβάνει τα δεδομένα των δυνάμεων που αναγράφονται).
- Η συγκεκριμένη εφαρμογή προσφέρει οπτικοποίηση δεδομένων σε πραγματικό χρόνο και διασφαλίζει τη σωστή χρήση του κηδεμόνα, βάσει των οδηγιών των κλινικών ιατρών.
- Διαθέτει κουμπί για ενημέρωση της ημερομηνίας και ώρας από την συσκευή που χρησιμοποιείται. Σε περίπτωση όπου η μπαταρία του RTC module, σταματήσει να λειτουργεί και χαθεί η σωστή καταμέτρηση του χρόνου, με την τοποθέτηση νέας μπαταρίας δίνεται η δυνατότητα επαναρύθμισης της ημερομηνίας και ώρας μέσω της εφαρμογής. Όταν η ενημέρωση πραγματοποιηθεί εμφανίζει κατάλληλο μήνυμα ότι η ημερομηνία και η ώρα ενημερώθηκαν (το αντίστοιχο μήνυμα καταγράφεται και στην κάρτα SD για να γίνεται γνωστό πότε χάθηκε η σωστή καταμέτρηση και πότε επαναπροσδιορίστηκε σωστά μέσω του αρχείου)
- Μια συγκεκριμένη ώρα μέσα στη μέρα (αυτή η ώρα είναι ορισμένη ως 00:00, με την αλλαγή της μέρας), εμφανίζεται ο συνολικός αριθμός των ωρών που ασκούνται οι δυνάμεις στους αισθητήρες και συνεπώς των ωρών που τοποθετείται ο νάρθηκας καθώς και μήνυμα για το αν ήταν κοντά στον στόχο που έπρεπε να επιτευχθεί, αν τον πέτυχε ή αν πρέπει να προσπαθήσει περισσότερο (το αντίστοιχο μήνυμα καταγράφεται και στην κάρτα SD για να γίνεται γνωστό πόσες ώρες φορέθηκε την προηγούμενη μέρα ακόμα και μετά το κλείσιμο της εφαρμογής).
- Δίνει δυνατότητα προσδιορισμού αυτών των ωρών, που αποτελούν τον στόχο, μέσω του κινητού, διότι αυτό διαφοροποιείται από ασθενή σε ασθενή και ανάλογα την εξέλιξη και την πορεία της πάθησης μπορεί να μεταβληθεί με βάση τα νέα

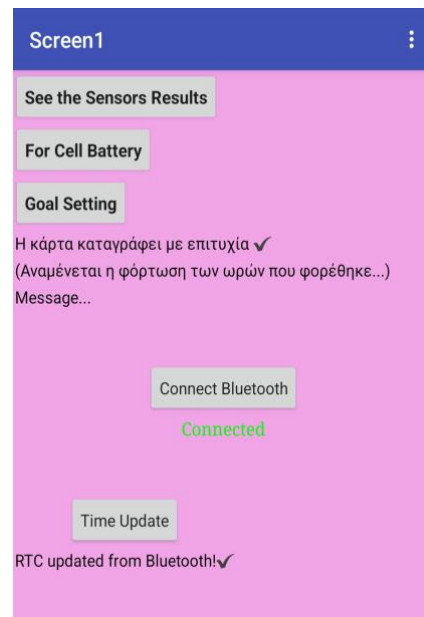
δεδομένα κι τις νέες απαιτήσεις. Έτσι δίνεται η δυνατότητα προσδιορισμού του στόχου για να εμφανίζονται και τα σωστά μηνύματα για το αν επιτεύχθηκε ή όχι. Το αντίστοιχο μήνυμα καταγράφεται και στην κάρτα SD για να γίνεται γνωστό πότε άλλαξε ο στόχος και με ποιο τρόπο μεταβλήθηκε.



Εικόνα 39: Αρχική Εικόνα Εφαρμογής Πριν την Σύνδεση



Εικόνα 38: Μετά την Σύνδεση μέσω του κουμπιού "Connect Bluetooth" και σε περίπτωση όπου η κάρτα SD δεν καταγράφει



Εικόνα 37: Ενημέρωση ημερομηνίας και ώρας. Με το πάτημα του κουμπιού "For Cell Battery" και έπειτα του "Time Update"



Εικόνα 40: Εικόνα Εφαρμογής αφού έχει πατηθεί το κουμπί "See the Sensors Results" και ενώ και η κάρτα SD καταγράφει και ασκούνται δυνάμεις και στους 3 αισθητήρες



Εικόνα 36: Εμφάνιση μηνύματος επιτυχίας

Φόρτωση ωρών που φορέθηκε ο κηδεμόνας (έχοντας ορίσει ως δοκιμαστικό στόχο 3 λεπτά) και εφ' όσον αυτός επιτεύχθηκε (στην συγκεκριμένη περίπτωση μάλιστα ξεπεράστηκε και για 1m και 54s).



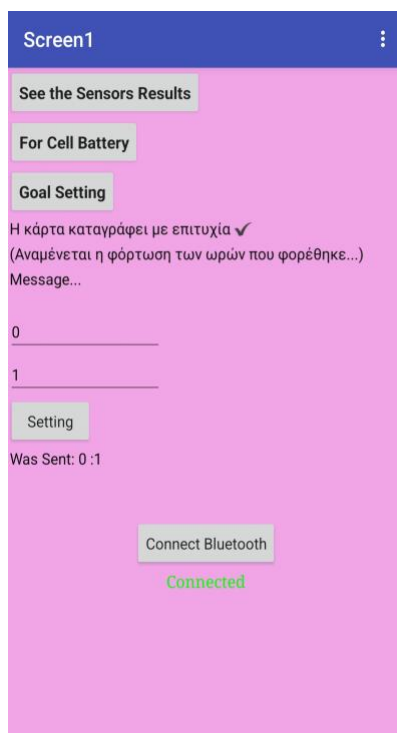
Εικόνα 35: Εμφάνιση μηνύματος για περισσότερη προσπάθεια

Φόρτωση ωρών που φορέθηκε ο κηδεμόνας (έχοντας ορίσει ως δοκιμαστικό στόχο 3 λεπτά) και εφ' όσον αυτός δεν επιτεύχθηκε (στην συγκεκριμένη περίπτωση μάλιστα φορέθηκε μόνο για 1m και 5s).



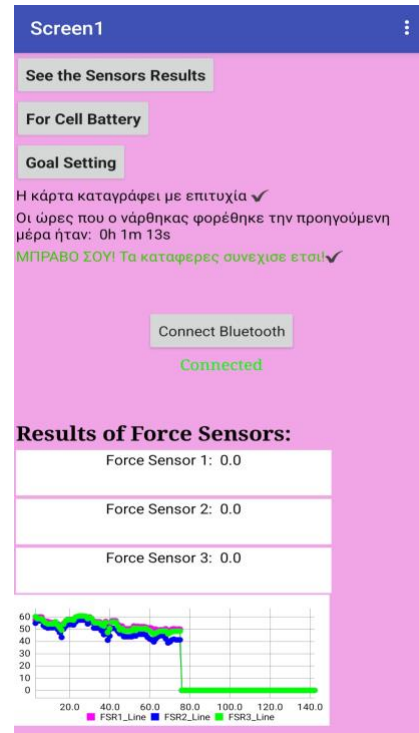
Εικόνα 41: Εμφάνιση μηνύματος για το ότι είναι κοντά στον στόχο

Φόρτωση ωρών που φορέθηκε ο κηδεμόνας (έχοντας ορίσει ως δοκιμαστικό στόχο 3 λεπτά) και εφ' όσον αυτός δεν επιτεύχθηκε για μικρή απόκλιση (στην συγκεκριμένη περίπτωση μάλιστα φορέθηκε για 2m και 30s). Η απόκλιση που χρειάζεται για να θεωρηθεί το αποτέλεσμα κοντά στον στόχο έχει οριστεί για τις δοκιμές 1 λεπτό. Εμφάνιση μηνύματος για το ότι είναι κοντά στον στόχο



Εικόνα 42: Αλλαγή στόχου σε 1 λεπτό μέσω της εφαρμογής

Με το πάτημα του κουμπιού "Goal Setting" και εμφάνιση μηνύματος ότι ο στόχος στάλθηκε και άρα ενημερώθηκε



Εικόνα 43: Δοκιμή μετά την αλλαγή του στόχου μέσω της εφαρμογής

Μήνυμα επιτυχίας με εφαρμογή μόνο 1m και 13s

Για τις συγκεκριμένες δοκιμές θέταμε ως αρχικό στόχο 3 λεπτά μέσω του προγραμματισμού και ορίζαμε την ώρα όπου θα φόρτωναν οι συνολικές ώρες εφαρμογής λίγα λεπτά μετά την εκάστοτε δοκιμή, ώστε να έχουμε γρήγορα αποτελέσματα.

Ακόμα και μέσω της αλλαγής του στόχου από την εφαρμογή ορίστηκε ως νέος στόχος μόνο το 1 λεπτό ώστε να ελεγχθεί άμεσα το αν έγινε σωστή ενημέρωση αυτού. Για τον λόγο αυτό μάλιστα ως μικρή απόκλιση για την εμφάνιση μηνύματος κοντά στον στόχο, ορίστηκε το 1 λεπτό.

Ωστόσο, γίνεται εύκολα αντιληπτό πως για την τελική δημιουργία η ώρα ενημέρωσης ωρών θα γίνεται μια συγκεκριμένη ώρα μέσα στη μέρα η οποία δεν θα αλλάζει και θα είναι η 00:00, με την αλλαγή της μέρας, καθώς και ότι ως μικρή απόκλιση θα οριστεί 1 με 1 μιση ώρα αφού ο επιθυμητός στόχος θα είναι σίγουρα κάποιες ώρες και όχι μόνο λίγα λεπτά.

```
data4 - Σημειωματάριο
Αρχείο Επεξεργασία Μορφή Προβολή Βοήθεια

2025-07-03 05:50:58
Force1(N): 51.7 Force2(N): 48.2 Force3(N): 56.4
Pressure1(kPa): 407.0 Pressure2(kPa): 379.9 Pressure3(kPa): 444.2

2025-07-03 05:50:59
Force1(N): 51.4 Force2(N): 48.5 Force3(N): 56.5
Pressure1(kPa): 404.6 Pressure2(kPa): 381.7 Pressure3(kPa): 445.1

2025-07-03 05:51:01
Force1(N): 51.0 Force2(N): 48.0 Force3(N): 55.8
Pressure1(kPa): 401.6 Pressure2(kPa): 377.6 Pressure3(kPa): 439.5

===== DAILY REPORT =====
2025-07-03 05:51:01
Total time under pressure: 0h 4m 54s

❑ Επτυχία! Πέτυχες ή ξεπέρασες τον στόχο!

2025-07-03 05:51:02
Force1(N): 51.8 Force2(N): 49.6 Force3(N): 56.8
Pressure1(kPa): 407.7 Pressure2(kPa): 390.3 Pressure3(kPa): 447.5
```

Εικόνα 45: Daily Report από το αρχείο .txt. για επίτευξη στόχου

Φορτώνει και τις ώρες εφαρμογής και το αντίστοιχο μήνυμα, για έλεγχο σε περίπτωση που δεν έχει ελεγχθεί μέσω του app

```
data4 - Σημειωματάριο
Αρχείο Επεξεργασία Μορφή Προβολή Βοήθεια

Force1(N): 0.0 Force2(N): 0.0 Force3(N): 6.9
Pressure1(kPa): 0.0 Pressure2(kPa): 0.0 Pressure3(kPa): 54.7

2025-07-03 07:00:00
Force1(N): 0.0 Force2(N): 0.0 Force3(N): 6.7
Pressure1(kPa): 0.0 Pressure2(kPa): 0.0 Pressure3(kPa): 52.5

===== DAILY REPORT =====
2025-07-03 07:00:00
Total time under pressure: 0h 2m 30s

⚠ Είσαι πολύ κοντά στον στόχο! Μπορείς λίγο ακόμα!

2025-07-03 07:00:01
Force1(N): 0.0 Force2(N): 0.0 Force3(N): 6.6
Pressure1(kPa): 0.0 Pressure2(kPa): 0.0 Pressure3(kPa): 52.1

2025-07-03 07:00:03
Force1(N): 0.0 Force2(N): 0.0 Force3(N): 6.9
Pressure1(kPa): 0.0 Pressure2(kPa): 0.0 Pressure3(kPa): 54.0
```

Εικόνα 44: Daily Report από το αρχείο .txt. για μη επίτευξη στόχου

```
data4 - Σημειωματάριο
Αρχείο Επεξεργασία Μορφή Προβολή Βοήθεια

2025-07-03 08:21:41

Force1(N): 0.0 Force2(N): 0.0 Force3(N): 0.0
Pressure1(kPa): 0.0 Pressure2(kPa): 0.0 Pressure3(kPa): 0.0

2025-07-03 08:21:42

Force1(N): 0.0 Force2(N): 0.0 Force3(N): 0.0
Pressure1(kPa): 0.0 Pressure2(kPa): 0.0 Pressure3(kPa): 0.0

2025-07-03 08:21:42 - Goal updated to: 0h 1m

2025-07-03 08:21:44

Force1(N): 0.0 Force2(N): 0.0 Force3(N): 0.0
Pressure1(kPa): 0.0 Pressure2(kPa): 0.0 Pressure3(kPa): 0.0
```

Εικόνα 48: Μήνυμα στο .txt αρχείο για ανανέωση στόχου εφαρμογής

```
data4 - Σημειωματάριο
Αρχείο Επεξεργασία Μορφή Προβολή Βοήθεια

2025-07-07 11:06:59

Force1(N): 0.0 Force2(N): 0.0 Force3(N): 0.0
Pressure1(kPa): 0.0 Pressure2(kPa): 0.0 Pressure3(kPa): 0.0

2025-07-07 11:07:01

Force1(N): 0.0 Force2(N): 0.0 Force3(N): 0.0
Pressure1(kPa): 0.0 Pressure2(kPa): 0.0 Pressure3(kPa): 0.0

2025-07-07 11:07:02 - RTC updated from Bluetooth!

2025-07-07 11:07:03

Force1(N): 0.0 Force2(N): 0.0 Force3(N): 0.0
Pressure1(kPa): 0.0 Pressure2(kPa): 0.0 Pressure3(kPa): 0.0
```

Εικόνα 47: Μήνυμα στο .txt αρχείο για ενημέρωση ημερομηνίας και ώρας από την εφαρμογή

```
data4 - Σημειωματάριο
Αρχείο Επεξεργασία Μορφή Προβολή Βοήθεια

2025-07-06 02:19:27

Force1(N): 0.0 Force2(N): 0.0 Force3(N): 0.0
Pressure1(kPa): 0.0 Pressure2(kPa): 0.0 Pressure3(kPa): 0.0

2025-07-06 02:19:49 ===== RECORDING STARTED AGAIN =====

2025-07-06 02:19:49

Force1(N): 0.0 Force2(N): 0.0 Force3(N): 0.0
Pressure1(kPa): 0.0 Pressure2(kPa): 0.0 Pressure3(kPa): 0.0

2025-07-06 02:19:51

Force1(N): 0.0 Force2(N): 0.0 Force3(N): 0.0
Pressure1(kPa): 0.0 Pressure2(kPa): 0.0 Pressure3(kPa): 0.0
```

Εικόνα 46: Μήνυμα για το ότι η κάρτα SD καταγράφει ξανά σε περίπτωση που για κάποιο διάστημα δεν γινόταν καταγραφή.

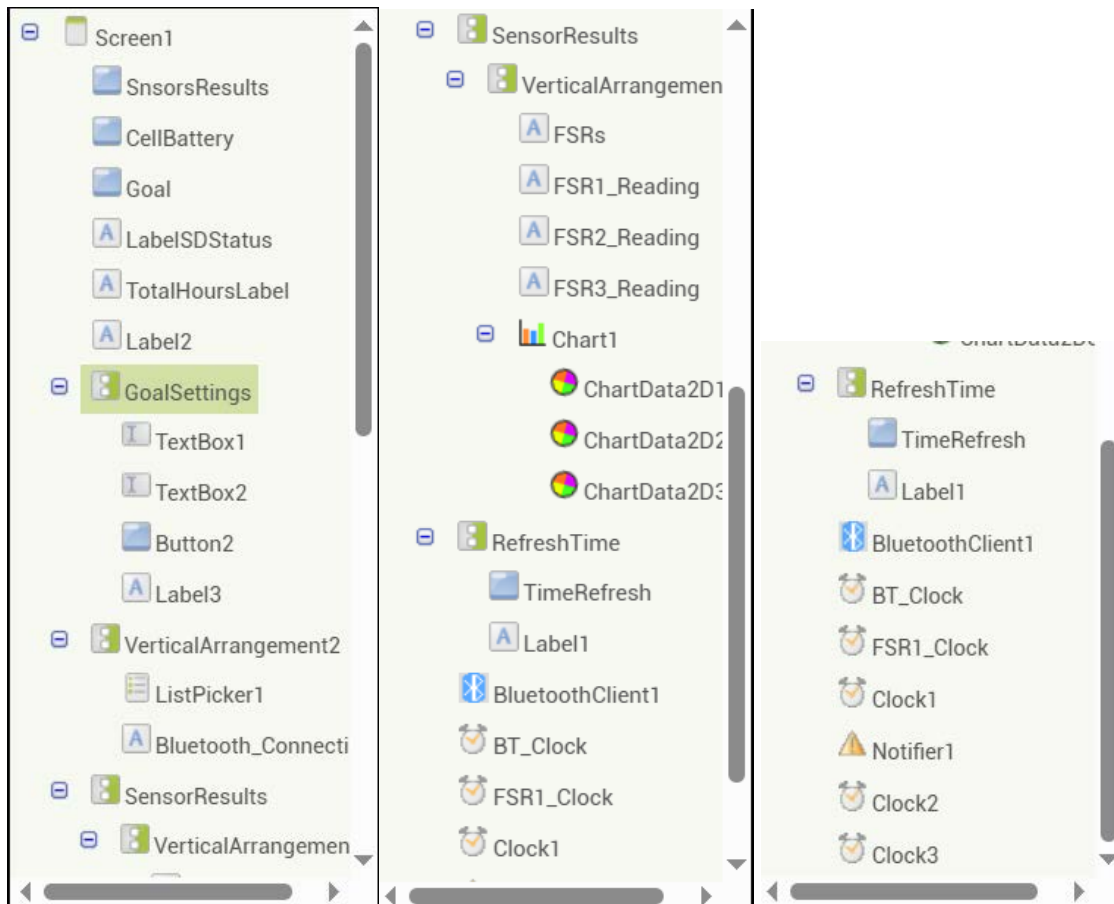
Εδώ η καταγραφή σταμάτησε στις 02:19:27 για 22s και ξεκίνησε ξανά στις 02:19:49

Η κάρτα SD χρησιμοποιείται ως μέσο μόνιμης αποθήκευσης των δεδομένων και των μηνυμάτων που καταγράφονται από το σύστημα, λειτουργώντας ως αρχείο καταγραφής και αναφοράς. Η εφαρμογή που αναπτύχθηκε έχει σχεδιαστεί για την προσομοίωση καταγραφής δεδομένων σε πραγματικό χρόνο (real-time). Ωστόσο, λόγω των περιορισμών των φορητών συσκευών και της φύσης της εφαρμογής, δεν είναι εφικτή η αποθήκευση δεδομένων όταν η εφαρμογή παραμένει πλήρως ανενεργή ή κλειστή. Για την κάλυψη αυτής της ανάγκης, ενσωματώθηκε η κάρτα SD στο σύστημα, επιτρέποντας την αυτόνομη αποθήκευση όλων των μετρήσεων ανεξάρτητα από τη χρήση ή μη της εφαρμογής. Με αυτόν τον τρόπο διασφαλίζεται η συνεχής παρακολούθηση των δεδομένων χωρίς την απαίτηση αδιάλειπτης λειτουργίας της εφαρμογής από τον χρήστη.

Για τις ανάγκες των δοκιμών και της αξιολόγησης του συστήματος, τα δεδομένα καταγράφονταν ανά δευτερόλεπτο, ώστε να επιτυγχάνεται υψηλή χρονική ανάλυση και να ελέγχεται η ακρίβεια της λειτουργίας. Ωστόσο, σε ένα πραγματικό σενάριο χρήσης, η καταγραφή σε τόσο μικρά χρονικά διαστήματα δεν είναι απαραίτητη. Η θεραπευτική πρόοδος και η συμμόρφωση του ασθενούς με τη χρήση του κηδεμόνα αξιολογούνται σε ημερήσια βάση, και επομένως, η ανανέωση των καταγραφών ανά μερικές ώρες κρίνεται επαρκής. Στόχος δεν είναι ο εντοπισμός της ακριβούς χρονικής στιγμής (δευτερόλεπτο) κατά την οποία υπήρξε ενδεχόμενη απόκλιση από τη σωστή εφαρμογή του νάρθηκα — καθώς κάτι τέτοιο, αν συμβεί, θα εξελίσσεται προοδευτικά — αλλά η συνολική αξιολόγηση ανά ημέρα, ώστε να είναι δυνατή η λήψη διορθωτικών μέτρων από τον θεράποντα ιατρό.

6.2 Διαδικασία δημιουργίας εφαρμογής μέσω MIT App Inventor

Αρχικά στην κατηγορία Desing προσθέτουμε τα εξής :



Εικόνα 49: Desing MIT App Inventor

Αυτά προστίθενται από την στήλη Palette στα αριστερά. Στην προκειμένη περίπτωση έχουν χρησιμοποιηθεί:

User Interface: Button, Label, TextBox, ListPicker

Charts: ChartData2D

Connectivity: BluetoothClient

Sensors: Clock

Layout: VerticalArrangement

6.3 Αναλυτική Περιγραφή Συλλογιστικής Πορείας και Λειτουργικών Ενοτήτων

6.3.1 Εγκατάσταση και Σύνδεση Bluetooth

- **ListPicker1.BeforePicking:** Φορτώνει τα διαθέσιμα ονόματα και διευθύνσεις Bluetooth.
- **ListPicker1.AfterPicking:** Επιλέγεται η συσκευή και γίνεται προσπάθεια σύνδεσης.
- **BT_Clock.Timer:** Ελέγχει περιοδικά τη σύνδεση και εφόσον υπάρχει, διαβάζει δεδομένα από τη σειριακή θύρα.

6.3.2 Επεξεργασία Εισερχόμενων Δεδομένων

- Τα ληφθέντα δεδομένα αναλύονται με βάση την παρουσία ειδικών λέξεων (π.χ. “SD ERROR”, “SUCCESS”, “RTC updated”).
- Τα μηνύματα φιλτράρονται με split και αποθηκεύονται σε λίστες για περαιτέρω επεξεργασία.
- Ανάλογα με το περιεχόμενο, προβάλλονται κατάλληλα μηνύματα στον χρήστη ή ενημερώνονται ετικέτες (Labels).
- Αν ανιχνευθούν δεδομένα αισθητήρων (FSRA, FSRB, FSRC), εμφανίζονται στις ετικέτες και αποθηκεύονται για απεικόνιση.

6.3.3 Οπτικοποίηση Αισθητήρων

- Οι τιμές των αισθητήρων καταγράφονται στον άξονα X με βάση την μεταβλητή TimeCounter και εισάγονται σε γραφήματα μέσω ChartData2D1.AddEntry.

6.3.4 Χειρισμός Χρόνου και Ρύθμιση Ρολογιού

- Ο χρήστης μπορεί να αποστείλει τον τρέχοντα χρόνο στη συσκευή με το κουμπί TimeRefresh.
- Ο χρήστης μπορεί επίσης να ορίσει στόχο ώρας και λεπτών (π.χ., για ενεργοποίηση ειδοποίησης) και να αποστείλει την εντολή μέσω του Button2.

6.3.5 Οπτική Διεπαφή και Διαχείριση Οθονών

- Πλήκτρα (SensorsResults, CellBattery, Goal) επιτρέπουν την εναλλαγή οθονών ενεργοποιώντας ή αποκρύπτοντας τα κατάλληλα Layouts (για την εναλλαγή των screens) .
- Το Notifier1.ChoosingCanceled διαχειρίζεται την ακύρωση της επιλογής συσκευής Bluetooth.

6.4 Οδηγίες για Δημιουργία της Εφαρμογής

6.4.1 Βήμα 1: Σχεδίαση GUI στο MIT App Inventor

- Δημιουργείται ListPicker, Button, Clock, Label, TextBox και ChartData2D αντικείμενα.
- Οργανώνονται σε τρία ξεχωριστά Layouts (SensorResults, RefreshTime, GoalSettings).

6.4.2 Βήμα 2: Δημιουργία Μεταβλητών και Λιστών

- Ορίζονται μεταβλητές όπως Reading, FSRA, FSRB, FSRC, goalHour, goalMinute, CurrentDateTime, TimeCounter, κ.ά.
- Αρχικοποιούνται σε κενές συμβολοσειρές ή μηδενικά.

6.4.3 Βήμα 3: Bluetooth Ροή Δεδομένων

- Χρησιμοποιείται BluetoothClient1.ReceiveText με βάση τα bytes που είναι διαθέσιμα.
- Διαχωρίζονται τα πακέτα με split at \n ή άλλους χαρακτήρες ώστε να μπορούν να διαχειριστούν κατάλληλα.
- Χρησιμοποιούνται ετικέτες για να εμφανίζονται μηνύματα κατάστασης ή αποτελέσματα αισθητήρων.

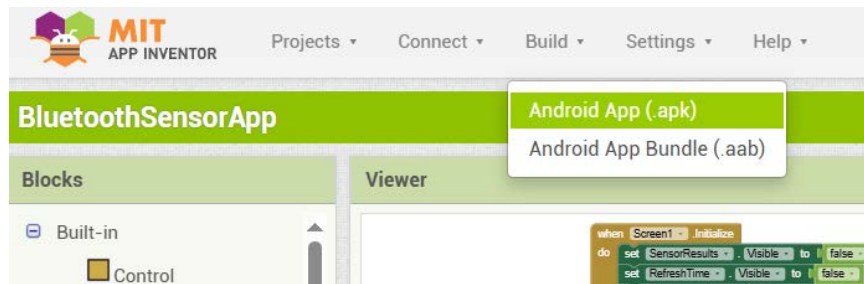
6.4.4 Βήμα 4: Χρονικός Συγχρονισμός

- Προγραμματίζονται τα Clock.Timer blocks να εκτελούνται ανά τακτά χρονικά διαστήματα (π.χ. κάθε 500ms) για να ελέγχουν την επικοινωνία και να καταγράφουν τιμές.

6.4.5 Βήμα 5: Αποστολή Εντολών

- Ο χρήστης συμπληρώνει ώρα και λεπτά και με το πάτημα του κουμπιού Button2 δημιουργείται η εντολή SETGOAL:X:Y, η οποία αποστέλλεται μέσω Bluetooth.
- Παρόμοια, η αποστολή του συστήματος ώρας γίνεται με την εντολή SETTIME:yyyy/MM/dd hh:mm:ss

6.4.6 Βήμα 6: Build Εφαρμογής

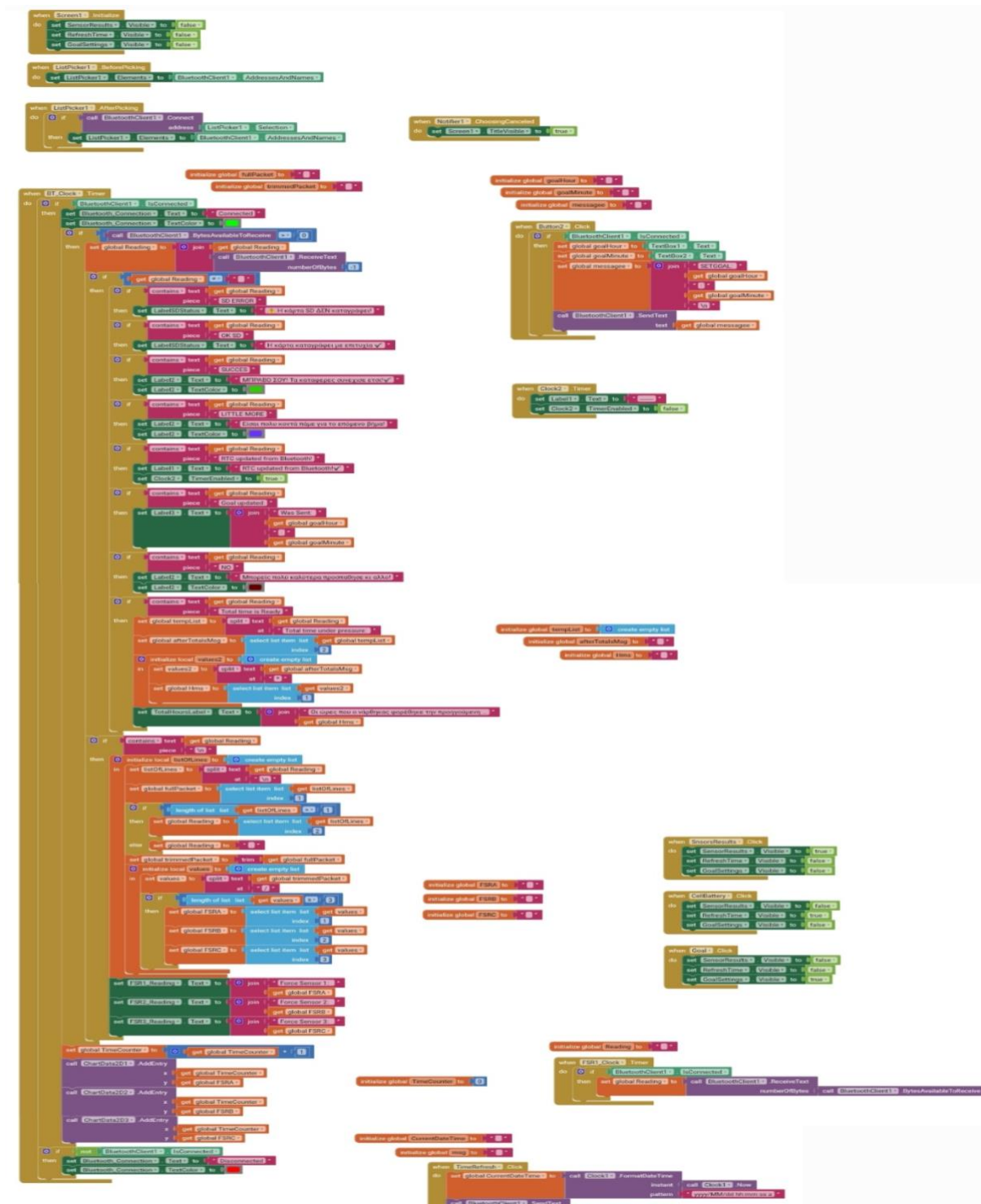


Εικόνα 50: Περιβάλλον MIT App Inventor

Για την εγκατάσταση της εφαρμογής στο κατάλληλο τηλέφωνο, επιλέγεται από το μενού η εντολή **Build** και στη συνέχεια η επιλογή **Android App (.apk)**. Κατόπιν, εμφανίζεται ένας κωδικός QR, ο οποίος πρέπει να σαρωθεί ώστε να ξεκινήσει η διαδικασία εγκατάστασης της εφαρμογής στη συσκευή.

Για την πραγματοποίηση της σάρωσης, απαιτείται η προηγούμενη εγκατάσταση της εφαρμογής **MIT App Inventor** από το **Google Play Store**. Η συγκεκριμένη εφαρμογή παρέχει το κατάλληλο περιβάλλον και την απαραίτητη κάμερα για την ανάγνωση του QR κωδικού και τη συνακόλουθη εγκατάσταση του παραγόμενου αρχείου εφαρμογής .apk.

6.5 Στιγμιότυπο του κώδικα σε blocks



Εικόνα 51: Στιγμιότυπο Κώδικα σε Blocks από MIT App Inventor

Σύνδεσμος MIT App Inventor: <https://gallery.appinventor.mit.edu/?galleryid=ad60702f-497a-479d-a7fc-c2a1a62a8de4>

7 Ανάλυση Λογισμικού ελέγχου φορητής συσκευής

7.1 Δηλώσεις

Εισαγωγές βιβλιοθηκών

- `#include <SD.h>` – Εισάγει τη βιβλιοθήκη για λειτουργίες με κάρτα SD (αποθήκευση δεδομένων).
- `#include <SPI.h>` – Απαιτείται για επικοινωνία SPI με την SD.
- `#include <Wire.h>` – Εισάγει το πρωτόκολλο I²C (χρήσιμο για το RTC).
- `#include <RTClib.h>` – Βιβλιοθήκη για διαχείριση του RTC (DS3231).

Σταθερές (define και const)

Αρχικά είναι σημαντικό να διευκρινιστεί ότι η οδηγία `#define` αποτελεί εντολή προεπεξεργαστή, η οποία πραγματοποιεί απλή αντικατάσταση κειμένου πριν τη μεταγλώττιση, χωρίς να διαθέτει τύπο ή μηχανισμό ελέγχου, γεγονός που μπορεί να οδηγήσει σε σφάλματα και δυσκολία εντοπισμού τους κατά το debugging. Αντίθετα, η δήλωση `const` ορίζει μια σταθερή μεταβλητή με σαφώς καθορισμένο τύπο, η οποία ελέγχεται από τον compiler ως προς τη συμβατότητα και το εύρος ορατότητάς της, προσφέροντας μεγαλύτερη ασφάλεια και καθαρότερο κώδικα. [78]

- `#define SENSOR_AREA 0.000127` – Ορίζει το εμβαδόν του αισθητήρα FSR σε m^2 ($0,5'' \sim 1,27 \text{ cm}^2$). Χρησιμοποιείται για τον υπολογισμό πίεσης (kPa).

Τι αντιπροσωπεύει το **SENSOR_AREA**:

Ο FSR (Force Sensitive Resistor) με διάμετρο **0,5''** έχει περίπου επιφάνεια:

$$D = 0,5'' = 0,5 \cdot 2,54 \text{ cm} = 1,27 \text{ cm}$$

Η ακτίνα:

$$r = \frac{D}{2} = \frac{1,27}{2} = 0,635 \text{ cm}$$

Εμβαδόν κύκλου:

$$A = \pi r^2 = \pi (0,635)^2 \text{ cm}^2$$

$$A \approx 3,1416 \cdot 0,403 \approx 1,267 \text{ cm}^2$$

Άρα η επιφάνεια του FSR είναι περίπου $1,27 \text{ cm}^2$

Για να μετατρέψουμε σε m^2 :

$$1 \text{ cm}^2 = 10^{-4} \text{ m}^2$$

Άρα:

$$A = 1,27 \text{ cm}^2 = 1,27 \times 10^{-4} \text{ m}^2 = 0,000127 \text{ m}^2$$

Αυτό το εμβαδόν αντιστοιχεί στην επιφάνεια που ασκείται η δύναμη.

Η **πίεση (Pressure, P)** σε Pascal, ορίζεται φυσικά ως η **δύναμη (F)** σε Newton ανά μονάδα **επιφάνειας (A)** σε m^2 :

$$P = \frac{F}{A}$$

[79]

Ωστόσο, επειδή ο πίνακας του FSR παρουσιάζει τη δύναμη σε grams (g). Θα πρέπει να γίνει μετατροπή από g σε N. Αυτό γίνεται μέσω του τύπου:

$$F(N) = F(gf) \times 0.00980665$$

[80]

Αν ο αισθητήρας υπολογίζει τη δύναμη F, τότε για να πάρουμε την πίεση:

$$P = \frac{F}{\text{SENSOR_AREA}}$$

Επειδή $1 \text{ kPa} = 1000 \text{ Pa}$, συχνά υπολογίζουμε:

$$P = \frac{F}{\text{SENSOR_AREA}} \cdot \frac{1}{1000}$$

Έτσι το SENSOR_AREA είναι απαραίτητο για να γίνει η μετατροπή από **δύναμη (N)** σε **πίεση (kPa)**.

- `#define READ_INTERVAL 1000` – Ορίζει το διάστημα μεταξύ αναγνώσεων αισθητήρων (ms).

- *const int SD_CS_PIN = 10* – Ο αριθμός **pin (Chip Select)** για την κάρτα SD.
- *const unsigned long sdCheckInterval = 2000* – Ανά πόσα **ms** θα γίνεται προσπάθεια επανασύνδεσης της SD αν αποτύχει.
- *const int FSR1_PIN = A0* – Pin του πρώτου FSR αισθητήρα.
- *const int FSR2_PIN = A1* – Pin του δεύτερου FSR αισθητήρα.
- *const int FSR3_PIN = A2* – Pin του τρίτου FSR αισθητήρα.

Μεταβλητές για τον χρόνο ανάγνωσης

- *unsigned long lastRead = 0* – Χρησιμοποιείται για να μετράει το χρονικό σημείο της τελευταίας ανάγνωσης αισθητήρων και να υλοποιεί το διάστημα READ_INTERVAL.

Μεταβλητές για SD κάρτα

- *File dataFile* – Αντικείμενο για το αρχείο δεδομένων που γράφεται στην SD.
- *bool sdOk = false* – Κατάσταση SD (true = διαθέσιμη και λειτουργεί, false = πρόβλημα).
- *unsigned long lastSdCheck = 0* – Αποθηκεύει πότε έγινε τελευταία προσπάθεια ελέγχου της SD.

Μεταβλητές για RTC

- *RTC_DS3231 rtc* – Μεταβλητή για το RTC module DS3231 (πραγματικός χρόνος και ημερομηνία).

Μεταβλητές για χρόνο πίεσης

- *unsigned long pressureStart = 0* – Σημειώνει το χρόνο (millis) όταν ξεκινάει η πίεση > 3N.
- *unsigned long totalPressureTime = 0* – Αθροίζει συνολικά τον χρόνο που υπήρξε πίεση μέσα στην ημέρα.

- *bool pressureActive = false* – Σημαία που δείχνει αν υπάρχει τρέχουσα πίεση (*true* = η πίεση είναι ενεργή).
- *int cutoffHour = 00* – **Ώρα** στην οποία γίνεται **καταγραφή** και **μηδενισμός** της **ημερήσιας διάρκειας**.
- *int cutoffMinute = 00* – **Λεπτό της ώρας** για το **ημερήσιο reset**.
- *bool reportedToday = false* – Αποτρέπει πολλαπλές αναφορές μέσα στην ίδια ημέρα.

Μεταβλητές για τον ημερήσιο στόχο

- *int goalHours = 15* – Ώρες του επιθυμητού στόχου (π.χ. 15h).
- *int goalMinutes = 3* – Λεπτά του στόχου (παράδειγμα 3').
- *double deviationHours = 0.016667* – Επιτρεπτή απόκλιση από τον στόχο (σε ώρες, π.χ. ~1 λεπτό), για τις δοκιμές είχαμε εισάγει το 0.016667 διότι τόσες ώρες είναι το 1 λεπτό που θέλαμε να έχουμε ως απόκλιση για τον έλεγχο, ωστόσο στη συνέχεια, αυτή η μεταβλητή ορίζεται ως **1** διότι θεωρούμε κοντινό στόχο μέχρι την μία ώρα απόκλιση.

7.2 Setup()

Κατά την εκτέλεση της συνάρτησης *setup()*, πραγματοποιούνται μια σειρά από κρίσιμες αρχικοποιήσεις και ελέγχους με σκοπό τη διασφάλιση της ακρίβειας και της αξιοπιστίας του συστήματος. Πρώτα, ορίζεται η ανάλυση του αναλογικού-ψηφιακού μετατροπέα (ADC) στα 12 bit, προκειμένου να επιτευχθούν ακριβείς μετρήσεις από τους αισθητήρες δύναμης (FSR). Στη συνέχεια, ενεργοποιούνται τόσο η σειριακή επικοινωνία μέσω USB για σκοπούς debugging, όσο και η UART για τη σύνδεση Bluetooth. Η κάρτα SD αρχικοποιείται και, εφόσον είναι διαθέσιμη, τίθεται σχετική σήμανση διαθεσιμότητας και δημιουργείται αρχείο με κατάλληλο header για την καταγραφή δεδομένων. Επιπλέον, εκκινείται η επικοινωνία I2C και γίνεται έλεγχος της παρουσίας του ρολογιού πραγματικού χρόνου (RTC) τύπου DS3231. Σε περίπτωση που ανιχνευθεί απώλεια ρεύματος στο RTC, η ώρα επαναρυθμίζεται με βάση τον χρόνο μεταγλώττισης του προγράμματος (compile time), ενώ η ενέργεια αυτή καταγράφεται. Το σύστημα υιοθετεί ρητές σημάνσεις, όπως η "sdOk", και εφαρμόζει προσέγγιση "fail fast" σε περιπτώσεις

όπου το RTC απουσιάζει, με στόχο τη διασφάλιση της χρονικής ακεραιότητας των καταγεγραμμένων δεδομένων.

Ακολουθεί ανάλυση μέσω ψευδοκώδικα και συλλογιστική πορεία.

setup():

ρυθμίζω ανάλυση ADC στα 12 bits

ανοίγω USB Serial στα 9600

ανοίγω UART Serial1 στα 9600

ορίζω τα pin των FSR ως εισόδους

εκτυπώνω μήνυμα ετοιμότητας

αν αποτυχία SD.begin():

δήλωσε sdOk = false

lastSdCheck = millis()

αλλιώς:

sdOk = true

άνοιξε/δημιούργησε "data4.txt" για εγγραφή

αν άνοιξε:

γράψε "RESULTS:"

κλείσε το αρχείο

αλλιώς:

δήλωσε σφάλμα σε Serial & Serial1

sdOk = false

Wire.begin() // για I2C

αν αποτυχία rtc.begin():

τύπωσε "RTC not found" και κόλλησε σε ατέρμονα βρόχο

αν `rtc.lostPower()`:

τύπωσε και κατέγραψε στο SD ότι χάθηκε η ισχύς

ρύθμισε το RTC με την ώρα μεταγλώττισης (`__DATE__`, `__TIME__`)

// (προαιρετικό) μία φορά για τον πρώτο συγχρονισμό της ώρας: `rtc.adjust(...)`

7.2.1 Ρύθμιση αναλογικού μετατροπέα & σειριακών διεπαφών

- **`analogReadResolution(12)`**: Στο Teensy 4.0 ο ADC δίνει 12-bit τιμές (0–4095). Αυτό είναι κρίσιμο για σωστό scaling των FSR μετρήσεων (μετατροπή σε Newton). [81], [82]
- **`Serial.begin(9600)`**: Ενεργοποιεί την USB–CDC θύρα προς το Serial Monitor στα 9600 bps (το baud στο USB Serial στον Teensy είναι σε μεγάλο βαθμό τυπικό, αλλά διατηρείται για συμβατότητα). [83]
- **`Serial1.begin(9600)`**: Θύρα UART του MCU προς HC-05.

7.2.2 Δήλωση pins FSR ως είσοδοι & αρχικό μήνυμα

- **`pinMode()`**: Ρυθμίζει τον καθορισμένο ακροδέκτη ώστε να συμπεριφέρεται είτε ως είσοδος (με ή χωρίς εσωτερική ασθενή αντίσταση pull-up ή pull-down) είτε ως έξοδος. Στην προκειμένη περίπτωση, ορίζονται μέσω των `pinMode(FSR1,2,3_PIN, INPUT)`, οι αισθητήρες δύναμης ως είσοδοι, αυτό δηλώνεται με το INPUT (ενώ αν θέλαμε να δηλώσουμε κάτι σαν έξοδο θα το δηλώναμε με το OUTPUT) [84]

7.2.3 SD κάρτα: αρχικοποίηση, σημάνσεις κατάστασης, δημιουργία αρχείου

- **SD.begin(SD_CS_PIN):** Ελέγχει αν το υποσύστημα SD λειτουργεί (SPI επικοινωνία, παρουσία κάρτας, σωστό CS pin). [85]
- **sdOk:** σήμανση διαθεσιμότητας SD. Χρήσιμη για αποφυγή μελλοντικών μη εγγραφών όταν η SD δεν λειτουργεί.
- **lastSdCheck = millis():** καταγραφή χρόνου αποτυχίας, η συνάρτηση millis() επιστρέφει τον αριθμό των χιλιοστών του δευτερολέπτου από τότε που η συσκευή ξεκίνησε να εκτελεί το τρέχον πρόγραμμα. [86]
- **Άνοιγμα data4.txt** (ονομασία αρχείου με οποιοδήποτε όνομα και .txt) **με FILE_WRITE:** δημιουργεί (αν δεν υπάρχει) ή ανοίγει για append. Η γραμμή "RESULTS:" λειτουργεί ως header.
- Σε αποτυχία ανοίγματος, το σφάλμα προωθείται και στο **Serial1** (π.χ. προς app μέσω Bluetooth).

7.2.4 RTC (DS3231): I2C, ανίχνευση και επαναφορά χρόνου

- **Wire.begin():** Εκκινεί το I2C bus και απαιτείται πριν την επικοινωνία με DS3231. [87]
- **rtc.begin():** Επιστρέφει false αν δεν βρίσκεται η συσκευή στη διεύθυνσή της. Σε αυτήν την περίπτωση, το πρόγραμμα σταματάει (while (1);), γεγονός που επιλέγεται συνειδητά για να μη συνεχίσει χωρίς εγκυρότητα χρόνου.
- **rtc.lostPower():** Το DS3231 ενημερώνει όταν έχει χαθεί τροφοδοσία (ή η μπαταρία αποφορτίστηκε). Αν ναι:
 - ο Καταγράφεται το γεγονός στο SD (αν είναι διαθέσιμη).
 - ο **rtc.adjust(DateTime(F(__DATE__), F(__TIME__)))**: Θέτει την ώρα με την **ημερομηνία/ώρα μεταγλώττισης** (compile time). Το F() τοποθετεί τα string literals στη flash (PROGMEM). [88]
- Η σχολιασμένη γραμμή **“One Time for adjusting”** είναι ένας ασφαλής τρόπος να ευθυγραμμιστεί χειροκίνητα το RTC με το χρόνο του υπολογιστή **μια φορά**. Αυτό ωστόσο δεν πρέπει να μείνει ενεργή μόνιμα, διότι κάθε reset θα προγραμματιζόταν η σωστή ημερομηνία και ώρα στη compile-time στιγμή. [88]

7.3 Συναρτήσεις

7.3.1 Η συνάρτηση readSmoothedFSR()

Ορισμός:

- float readSmoothedFSR(int pin, int samples = 10, int delayMs = 5)

Όνομα	Τύπος	Περιγραφή
pin	int	Ο αναλογικός ακροδέκτης (π.χ. A0) όπου είναι συνδεδεμένος ο αισθητήρας FSR.
samples	int (προεπιλογή = 10)	Πόσες μετρήσεις να ληφθούν για εξομάλυνση.
delayMs	int (προεπιλογή = 5 ms)	Χρονική καθυστέρηση μεταξύ των μετρήσεων.

Table 11: Περιγραφή συνάρτησης readSmoothedFSR()

Στη συνάρτηση readSmoothedFSR(), αρχικά πραγματοποιείται η **αρχικοποίηση της μεταβλητής sum**, η οποία λειτουργεί ως αθροιστής των επιμέρους δειγμάτων που πρόκειται να ληφθούν από τον αισθητήρα. Η μεταβλητή αυτή είναι τύπου long, ώστε να επιτρέπει την ασφαλή συγκέντρωση μεγάλου αριθμού αναλογικών μετρήσεων χωρίς κίνδυνο υπερχείλισης, ιδιαίτερα σε περιπτώσεις που χρησιμοποιούνται υψηλές αναλύσεις ADC (π.χ. 12-bit, εύρος 0–4095).

Ακολουθεί μια **επαναληπτική διαδικασία** η οποία εκτελείται όσες φορές καθορίζονται από την παράμετρο samples. Σε κάθε επανάληψη πραγματοποιείται **ανάγνωση αναλογικής τιμής** από το καθορισμένο ακροδέκτη (pin) μέσω της εντολής analogRead() [89], η οποία αποτυπώνει την παρούσα ηλεκτρική τάση που προκύπτει από τον αισθητήρα πίεσης. Η τιμή αυτή προστίθεται στο sum, συγκεντρώνοντας σταδιακά το συνολικό άθροισμα των μετρήσεων. Μεταξύ των διαδοχικών αναγνώσεων εισάγεται **χρονική καθυστέρηση** διάρκειας delayMs χιλιοστών του δευτερολέπτου, με σκοπό να διασφαλιστεί χρονική απόσταση μεταξύ των δειγμάτων, αποφεύγοντας την ταυτόχρονη ανάγνωση παροδικών ή στιγμιαίων μεταβολών του σήματος.

Ο βασικός στόχος της διαδικασίας είναι η εφαρμογή ενός **απλού μέσου όρου** (simple moving average) [90], ώστε να επιτευχθεί **εξομάλυνση (smoothing)** του αναλογικού

σήματος [91], [92]. Η τεχνική αυτή είναι ιδιαίτερα χρήσιμη στην επεξεργασία σημάτων από FSR αισθητήρες, οι οποίοι συχνά εμφανίζουν διακυμάνσεις λόγω περιβαλλοντικού ή ηλεκτρικού θορύβου.

Τέλος, η συνάρτηση επιστρέφει τη μέση τιμή των μετρήσεων ως δεκαδικός αριθμός (float). Πριν την επιστροφή, το άθροισμα sum μετατρέπεται ρητά σε float, ώστε να αποφευχθεί ο ακέραιος διαχωρισμός (integer division), που θα οδηγούσε σε απώλεια ακρίβειας κατά την πράξη της διαίρεσης.

Παρακάτω φαίνεται ο αντίστοιχος ψευδοκώδικας της συγκεκριμένης συνάρτησης για καλύτερη κατανόηση:

```
function readSmoothedFSR(pin, samples = 10, delayMs = 5):
```

```
    sum = 0
```

```
    repeat samples times:
```

```
        value = analogRead(pin)
```

```
        sum = sum + value
```

```
        wait for delayMs milliseconds
```

```
    return sum / samples as float
```

7.3.2 Η συνάρτηση checkForBluetoothCommands()

Η συνάρτηση checkForBluetoothCommands() είναι υπεύθυνη για την **ανάγνωση, ερμηνεία και εκτέλεση εντολών** που αποστέλλονται μέσω Bluetooth στο σύστημα (π.χ. από εφαρμογή σε smartphone). Οι εντολές αποστέλλονται μέσω της θύρας Serial1, η οποία συνδέεται με το Bluetooth module (π.χ. HC-05).

Παρακάτω φαίνεται ο αντίστοιχος ψευδοκώδικας της συγκεκριμένης συνάρτησης για καλύτερη κατανόηση:

αν υπάρχουν διαθέσιμα bytes από Bluetooth:

```
    διάβασε τη συμβολοσειρά μέχρι το \n
```

```
    αφαίρεσε τα περιττά κενά
```

αν ξεκινά με "SETTIME:"

εξήγαγε και ανέλυσε ημερομηνία-ώρα

ενημέρωσε το RTC

εκτύπωσε επιβεβαίωση και αποθήκευσε στο αρχείο SD

αλλιώς αν ξεκινά με "SETGOAL:"

ανέλυσε την ώρα στόχου

έλεγε εγκυρότητα

ενημέρωσε τις global μεταβλητές

εκτύπωσε επιβεβαίωση και αποθήκευσε στο SD

Είναι σημαντικό να διευκρινιστεί ότι τα δεδομένα που αποστέλλονται από τον μικροελεγκτή προς το Bluetooth module HC-05 μεταφέρονται υπό τη μορφή ενιαίας συμβολοσειράς (string). Προκειμένου να καταστεί δυνατός ο διαχωρισμός αυτής της συμβολοσειράς σε επιμέρους τμήματα πληροφορίας, καθένα από τα οποία αντιστοιχεί σε διαφορετική παράμετρο ή μέτρηση από αισθητήρα, απαιτείται η ενσωμάτωση κατάλληλων διαχωριστικών στοιχείων. Αυτά μπορεί να περιλαμβάνουν χαρακτήρες αλλαγής γραμμής (όπως \n), κόμματα, ή ειδικές λέξεις-κλειδιά που καθορίζονται από τον προγραμματιστή με βάση τις ανάγκες της εφαρμογής. Η χρήση τέτοιων διαχωριστικών επιτρέπει στη συσκευή λήψης (π.χ. μία εφαρμογή στο MIT App Inventor) [2] να επεξεργάζεται αξιόπιστα τη ροή δεδομένων και να εξάγει τις επιμέρους πληροφορίες με ακρίβεια. [93], [94]

Αντιστοίχως, κατά την αντίστροφη ροή πληροφορίας, δηλαδή όταν η εφαρμογή στέλνει δεδομένα προς τον μικροελεγκτή μέσω Bluetooth, η επεξεργασία των εισερχόμενων στοιχείων προϋποθέτει ανάλογη αποσαφήνιση των δεδομένων και κατάλληλη ανάλυση (parsing) από το λογισμικό του μικροελεγκτή. Ο μικροελεγκτής οφείλει να διαβάσει τη ροή χαρακτήρων που καταφθάνει μέσω της σειριακής θύρας, να ανιχνεύει τα διαχωριστικά σημεία, και να αντιστοιχίζει τα επί μέρους τμήματα σε συγκεκριμένες εντολές ή παραμέτρους, γι' αυτόν τον σκοπό χρησιμοποιούνται οι λέξεις κλειδιά "SETTIME:" και "SETGOAL:" [94]

Κάποιες βασικές συναρτήσεις που χρησιμοποιήθηκαν εντός της συγκεκριμένης συνάρτησης και η εξήγησή τους:

Η **Serial1.readStringUntil('\n')** διαβάζει τη συμβολοσειρά έως τον χαρακτήρα αλλαγής γραμμής **\n**. [95] . Αυτό το αποθηκεύουμε σε μια μεταβλητή `input` τύπου `string` (`String input = Serial1.readStringUntil('\n');`) στην οποία στη συνέχεια θέλουμε να αφαιρέσουμε τα κενά με την συνάρτηση `trim()`, (`input.trim();`).

Η **trim()** αφαιρεί κενά ή αόρατους χαρακτήρες από την αρχή και το τέλος, βελτιώνοντας την αξιοπιστία ανάλυσης. [96]

Η **substring()** αφαιρεί τόσους χαρακτήρες όσους ορίζει ο αριθμός που βρίσκεται μέσα στην παρένθεση (π.χ το `substring(8)` αφαιρεί 8 χαρακτήρες και χρησιμοποιείται για την αφαίρεση του προθέματος "SETTIME:"). [97]

Η ίδια συνάρτηση, με δύο ορίσματα, **substring(start, end)**, χρησιμοποιείται και ως εξής:

```
int year = datetime.substring(0, 4).toInt();  
int month = datetime.substring(5, 7).toInt();  
int day = datetime.substring(8, 10).toInt();  
int hour = datetime.substring(11, 13).toInt();  
int minute = datetime.substring(14, 16).toInt();  
int second = datetime.substring(17, 19).toInt();
```

Το παραπάνω τμήμα κώδικα χρησιμοποιεί τη μέθοδο `substring()` για την εξαγωγή επιμέρους συστατικών στοιχείων (έτος, μήνας, ημέρα, ώρα, λεπτά, δευτερόλεπτα) από μια συμβολοσειρά τύπου ημερομηνίας-ώρας, σύμφωνα με τη μορφή ISO 8601, π.χ. "2025-08-03 14:35:22" [98]. Η μέθοδος **substring(start, end)** [97] ανακτά το τμήμα της συμβολοσειράς μεταξύ των θέσεων `start` (συμπεριλαμβάνεται) και `end` (δεν συμπεριλαμβάνεται). Έτσι:

- Οι χαρακτήρες στις θέσεις 0–4 αντιστοιχούν στο έτος,
- 5–7 στον μήνα,
- 8–10 στην ημέρα,
- 11–13 στην ώρα,

- 14–16 στα λεπτά,
- 17–19 στα δευτερόλεπτα.

Η επιλογή αυτών των θέσεων βασίζεται στην προκαθορισμένη μορφή της συμβολοσειράς (με σταθερό μήκος και διάταξη), επιτρέποντας την αξιόπιστη και αποδοτική εξαγωγή και μετατροπή κάθε τμήματος σε ακέραια αριθμητική τιμή (int) για περαιτέρω επεξεργασία.

Η **rtc.adjust(DateTime(year, month, day, hour, minute, second))**, ρυθμίζει το RTC (DS3231) με την αποσταλμένη ώρα. [99]

Με την **input.startsWith(" ")** δηλώνουμε πως αν το μήνυμα που λαμβάνεται από το bluetooth ξεκινάει με τη λέξη που βρίσκεται εντός των εισαγωγικών (και την οποία έχουμε ορίσει εμείς) τότε προχωρά στην ανάλυση (μπαίνει μέσα σε else if()) [100], [101]

Με την **indexOf(':', , firstColon + 1)** βρείσκεται η πρώτη εμφάνιση του ":", ξεκινώντας από τη θέση , firstColon + 1

```
int firstColon = input.indexOf(':');
```

```
int secondColon = input.indexOf(':', firstColon + 1);
```

με αυτόν τον τρόπο αποθηκεύεται η θέση στην οποία βρίσκεται η πρώτη “ : ” στην μεταβλητή firstColon και η δεύτερη στην μεταβλητή secondColon [102]

Μέσω της συνάρτησης **toInt()**, μετατρέπονται σε ακέραιους τα strings που εξάγονται για τις ώρες και τα λεπτά, αυτό γίνεται μέσω των εξής γραμμών κώδικα:

```
int h = hStr.toInt();
```

```
int m = mStr.toInt();
```

ενώ στη συνέχεια ελέγχεται αν αυτές οι τιμές είναι έγκυρες μέσω της εξής συνθήκης:

```
if (h >= 0 && m >= 0 && m < 60)
```

7.4 Loop()

Η συνάρτηση loop() του συγκεκριμένου κώδικα αποτελεί το βασικό κομμάτι του προγράμματος και εκτελείται συνεχώς από τον μικροελεγκτή Teensy 4.0. Ο κύριος σκοπός της είναι:

- Η λήψη και αποστολή εντολών μέσω Bluetooth (Serial1).
- Η περιοδική μέτρηση πίεσης από 3 αισθητήρες FSR.
- Ο υπολογισμός δύναμης και πίεσης σε πραγματικό χρόνο.
- Η αποθήκευση δεδομένων σε κάρτα SD.
- Η παρακολούθηση συνολικού χρόνου εφαρμογής πίεσης.
- Η αποστολή ημερήσιας αναφοράς και σύγκριση με στόχο.

Αρχικά περνάμε τη συσκευή σε "AT Mode" (για ρύθμιση HC-05), προωθούνται δεδομένα ανάμεσα σε Serial (USB προς υπολογιστή) και Serial1 (Bluetooth).

Αν έρθει δεδομένο από τον υπολογιστή:

Στείλτο προς Bluetooth (Serial1)

Αν έρθει δεδομένο από το Bluetooth:

Στείλτο προς υπολογιστή (Serial)

Έπειτα με το **if (millis() - lastRead >= READ_INTERVAL)** σκοπός είναι να γίνονται μετρήσεις σε τακτά χρονικά διαστήματα (π.χ. κάθε 500ms). [103]

Με τις εξής γραμμές κώδικα για τους αισθητήρες πίεσης:

```
float raw = readSmoothedFSR(FSR_PIN);
```

```
float voltage = raw * (3.3 / 4095.0);
```

```
float force = (voltage >= 0.1) ? pow(voltage/3.3, 2.5) * 100.0 : 0.0;
```

```
float pressure = (force / SENSOR_AREA) / 1000.0;
```

πραγματοποιούνται τα παρακάτω:

- readSmoothedFSR() παίρνει τον μέσο όρο 10 μετρήσεων από τον αναλογικό ακροδέκτη.
- Η αναλογική τιμή μετατρέπεται σε τάση.
 - ο Μετατρέπει την **ψηφιακή τιμή από τον ADC (Analog to Digital Converter)** σε τάση σε Volt.
 - ο **raw** είναι η αναλογική είσοδος από το analogRead() (0–4095 σε Teensy 4.0, που έχει **12-bit ADC**).

- ο **3.3** είναι η αναφορά τάσης (V_{ref}), δηλαδή η μέγιστη δυνατή τάση στην είσοδο. [104]
- ο **4095.0** είναι η μέγιστη τιμή 12-bit ADC ($2^{12} - 1$), αφαιρείται 1 επειδή οι τιμές ξεκινούν από το 0, άρα υπάρχουν 2^n διακριτά επίπεδα, αριθμημένα από 0 έως $2^n - 1$. [105]
- Η τάση μετατρέπεται εμπειρικά σε δύναμη (μονάδα: Newton) με εκθετική σχέση.
 - ο **pow(voltage / 3.3, 2.5)** είναι μια **μη γραμμική σχέση** μεταξύ τάσης και δύναμης, συνηθισμένη στα FSRs, που έχουν **log-log** ή **power law** συμπεριφορά. [106]
 - ο Πολλαπλασιάζεται επί 100.0 για να μετατραπεί σε εκτίμηση δύναμης (σε N).
 - ο **voltage >= 0.1** ορίζει **κατώφλι θορύβου**, κάτω από 0.1V θεωρείται μηδενική δύναμη λόγω αστάθειας/θορύβου του σήματος.
 - ο Ο εκθέτης **2.5** είναι **εμπειρικός** και προτείνεται από τον FSR 402. [106]
- Η πίεση προκύπτει από τη σχέση: **$P = F / A$** , με μετατροπή σε kPa.
 - ο Δύναμη / Εμβαδόν = Πίεση (σε $N/m^2 = Pascal$)
 - ο Διαιρείται με 1000.0 για να μετατρέψει από **Pascal σε kPa**

Για κάθε FSR:

Μέτρησε την αναλογική τιμή

Μετατροπή σε τάση

Αν τάση > 0.1V:

Υπολόγισε δύναμη με εμπειρικό τύπο

Αλλιώς:

Δύναμη = 0

Υπολόγισε πίεση = δύναμη / επιφάνεια αισθητήρα

Έπειτα, παρακολουθείται αν και πόσο διάστημα και οι 3 αισθητήρες δέχονται πίεση πάνω από ένα όριο (>3N), (η τιμή 3 είναι προσεγγιστική).

Αν και οι 3 αισθητήρες > 3N:

Αν δεν ήταν ήδη ενεργή πίεση:

Ξεκίνα χρονομέτρηση

Αλλιώς:

Αν ήταν ενεργή πίεση:

Σταμάτα χρονομέτρηση και πρόσθεσε χρόνο στο totalPressureTime

Στη συνέχεια για κάθε αισθητήρα εκτυπώνεται: Force (N), Pressure (kPa), Raw, Voltage και μεταδίδονται οι τιμές μέσω Bluetooth σε μορφή N1/N2/N3 για να είναι έτοιμες να διαχωριστούν στη συνέχεια ώστε να εμφανιστούν στη συνδεδεμένη συσκευή μέσω MIT App Inventor.

Ακόμα, γίνεται διαχείριση της SD κάρτας

Αν SD δεν δουλεύει και πέρασε αρκετός χρόνος:

Δοκίμασε επανεκκίνηση SD

Αν πετύχει:

Κατάγραψε "RECORDING STARTED AGAIN"

Αν SD δουλεύει:

Άνοιξε αρχείο

Κατάγραψε δεδομένα πίεσης + δύναμης με timestamp

Μέσα στο loop προγραμματίζεται και ο τρόπος εμφάνισης της ημερήσιας αναφοράς και των στόχων, πιο συγκεκριμένα

Αν ώρα == cutoff και δεν έχουμε στείλει αναφορά σήμερα:

Αν πίεση ενεργή:

Σταμάτα χρονομέτρηση

Υπολόγισε συνολική διάρκεια (h, m, s)

Αν χρόνος >= στόχος:

Εκτύπωσε "Επιτυχία"

Αλλιώς αν κοντά στο στόχο:

Εκτύπωσε "Είσαι πολύ κοντά"

Αλλιώς:

Εκτύπωσε "Προσπάθησε αύριο"

Κατάγραψε όλα τα παραπάνω σε SD

Reset σε totalPressureTime

Σημείωσε ότι στάλθηκε η αναφορά σήμερα

Τέλος γίνεται και κατάργηση του flag reportedToday με το εξής κομμάτι κώδικα:

```
if (now.hour()!=cutoffHour || now.minute()!=cutoffMinute)
{
    reportedToday = false;
}
```

Όστε να εξασφαλίζει ότι την επόμενη μέρα θα μπορεί ξανά να σταλεί αναφορά.

8 Πειραματικές δοκιμές συνολικού συστήματος

Για την αξιολόγηση της λειτουργικότητας και της αξιοπιστίας του υλοποιημένου συστήματος, πραγματοποιήθηκαν στοχευμένες πειραματικές δοκιμές σε ελεγχόμενο περιβάλλον. Σκοπός των δοκιμών ήταν:

- να διερευνηθεί η απόκριση των αισθητήρων πίεσης (FSRs),
- η ακρίβεια χρονοσήμανσης του RTC module,
- η ορθότητα καταγραφής των δεδομένων στην κάρτα SD
- η συμπεριφορά του συστήματος υπό τροφοδοσία από μπαταρία λιθίου και
- να επιβεβαιωθεί η επαναληψιμότητα των μετρήσεων.

Κατά τη διάρκεια των δοκιμών, εφαρμόστηκαν μηχανικά φορτία διαβαθμισμένης έντασης στους αισθητήρες FSR ώστε να καταγραφεί η ευαισθησία και η γραμμικότητα της μέτρησης. Παράλληλα, ελέγχθηκε η καταγραφή της ένδειξης πίεσης σε συνδυασμό με το αποθηκευμένο χρονικό αποτύπωμα από το RTC. Οι μετρήσεις επαναλήφθηκαν για διαφορετικές χρονικές περιόδους προκειμένου να επιβεβαιωθεί η σταθερότητα του ρολογιού και η ακεραιότητα των δεδομένων που αποθηκεύονται στην κάρτα SD.

Πιο συγκεκριμένα, στο πλαίσιο των πειραματικών δοκιμών, πραγματοποιήθηκαν τροποποιήσεις στην προκαθορισμένη ώρα δημιουργίας της ημερήσιας αναφοράς (daily report), προκειμένου να αξιολογηθεί η ορθή λειτουργία του συστήματος. Σε κάθε περίπτωση, οριζόταν ως ώρα εκτέλεσης του ημερήσιου ελέγχου μια χρονική στιγμή λίγα λεπτά μεταγενέστερη από την τρέχουσα, ενώ ο στόχος εφαρμογής της πίεσης καθοριζόταν επίσης σε περιορισμένο χρονικό εύρος (ελάχιστα λεπτά). Με τον τρόπο αυτό, ελέγχθηκε αφενός η σωστή εμφάνιση των μηνυμάτων που υποδεικνύουν την επίτευξη ή μη του στόχου και αφετέρου, η ακριβής καταγραφή των αντίστοιχων πληροφοριών στην κάρτα SD τη χρονική στιγμή που ορίζεται από το ρολόι πραγματικού χρόνου.

Το σύστημα τέθηκε σε λειτουργία υπό φορητές συνθήκες, με αποκλειστική τροφοδοσία από μπαταρία λιθίου μέσω κυκλώματος φόρτισης TP4056, προκειμένου να εξεταστεί η αυτονομία και η ενεργειακή απόδοση του συνόλου. Κατά τις δοκιμές, έγινε παρακολούθηση της διάρκειας λειτουργίας έως την πλήρη αποφόρτιση της μπαταρίας, καθώς και αξιολόγηση της απόδοσης του κυκλώματος φόρτισης κατά την επαναφόρτιση.

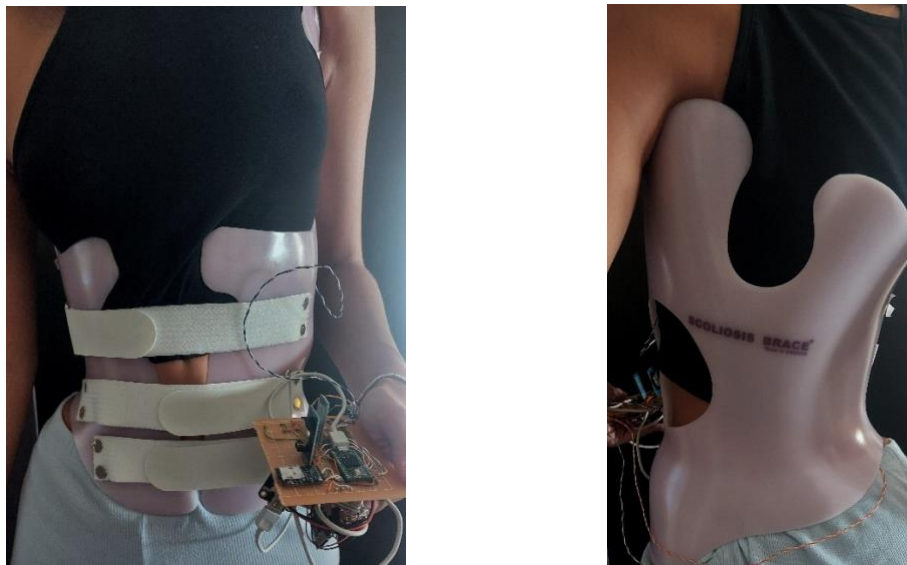
Τα αποτελέσματα έδειξαν ικανοποιητική λειτουργία του συστήματος σε όλα τα επίπεδα, με συνέπεια στην αποθήκευση δεδομένων, επαναληψιμότητα μετρήσεων και καλή διαχείριση ισχύος.

Αναλυτικότερα, πραγματοποιήθηκαν τέσσερις διακριτές δοκιμές με σκοπό τον έλεγχο της αποτελεσματικότητας του συστήματος. Η διαδικασία προέβλεπε τη χρήση του νάρθηκα μία φορά σε κάθε περίπτωση, ώστε κάθε δοκιμή να αντιπροσωπεύει μία διαφορετική ημέρα εφαρμογής. Η αξιολόγηση εστίασε:

- στην εμφάνιση των κατάλληλων μηνυμάτων στο *daily report*, με βάση τον προκαθορισμένο χρονικό στόχο εφαρμογής και την προκαθορισμένη επιτρεπτή απόκλιση από αυτόν, και
- στην ορθότητα των μετρούμενων δυνάμεων που ασκήθηκαν στους αισθητήρες πίεσης.

Η εγκυρότητα των μετρήσεων επιβεβαιώθηκε μέσω γραφημάτων στο MATLAB, τα οποία παρουσίασαν συμφωνία με τα διαγράμματα της εφαρμογής.

Για τις πειραματικές δοκιμές ορίστηκε διάρκεια εφαρμογής τριών λεπτών, με επιτρεπτό όριο απόκλισης ενός λεπτού. Η συγκεκριμένη ανοχή καθορίζει το πλαίσιο εντός του οποίου η διάρκεια θεωρείται ικανοποιητικά κοντά στον επιθυμητό στόχο, επιτρέποντας έτσι την εμφάνιση του αντίστοιχου μηνύματος στο *daily report*.

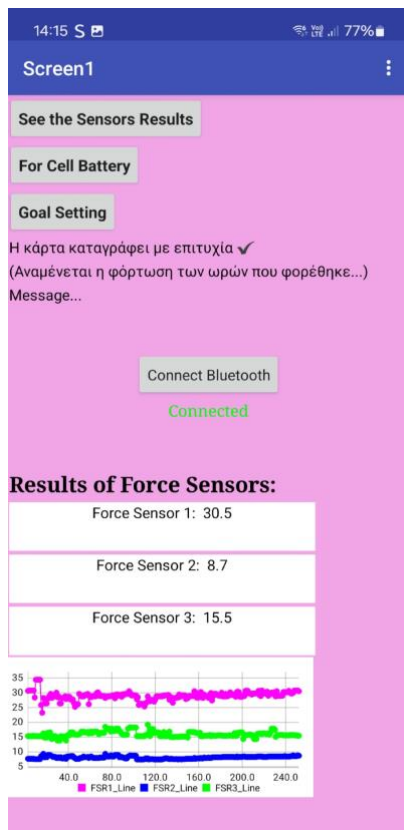


Εικόνα 52: Τοποθέτηση νάρθηκα στο σώμα μπροστά και πίσω όψη

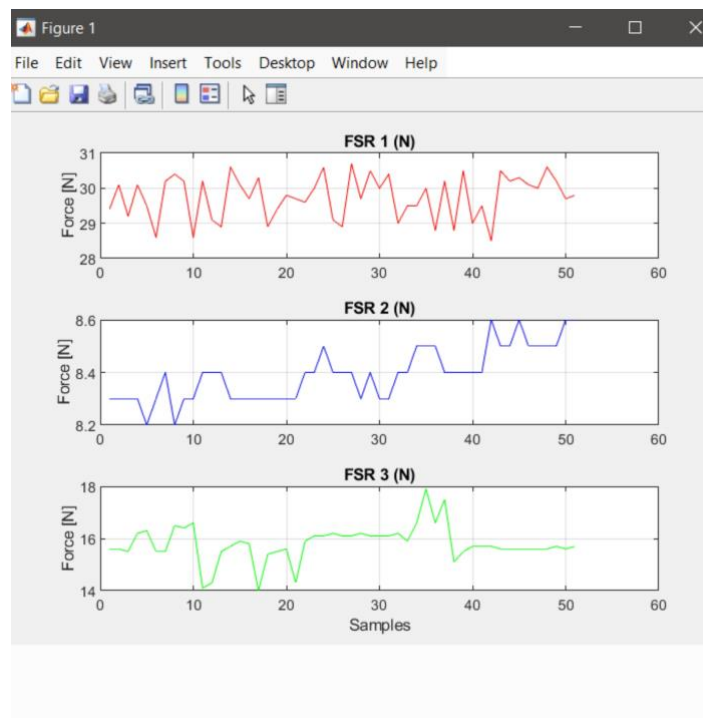
Παράλληλα, σε κάθε δοκιμή καθοριζόταν διαφορετική χρονική στιγμή για την καταμέτρηση της συνολικής διάρκειας εφαρμογής (εμφάνιση *daily report*), λίγα λεπτά μετά την τοποθέτηση του νάρθηκα, σύμφωνα με τις απαιτήσεις του εκάστοτε πειράματος.

1^η δοκιμή

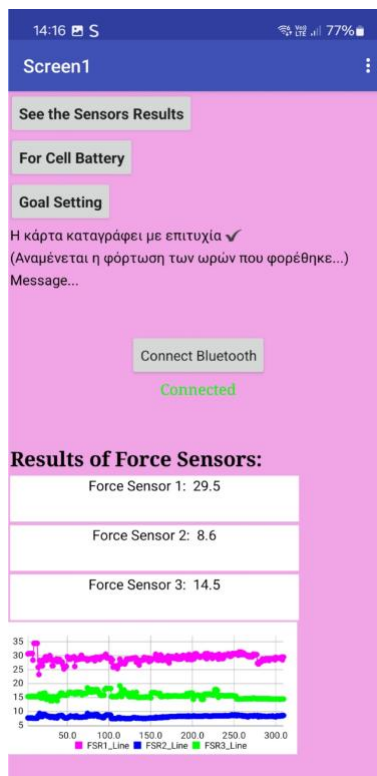
Περιγραφή: Σκοπός είναι ο στόχος να επιτευχθεί καθώς ο νάρθηκας τοποθετείται για χρόνο $\geq 3^{\text{ων}}$ λεπτών



Εικόνα 53: Εφαρμογή, πείραμα 1 (α)



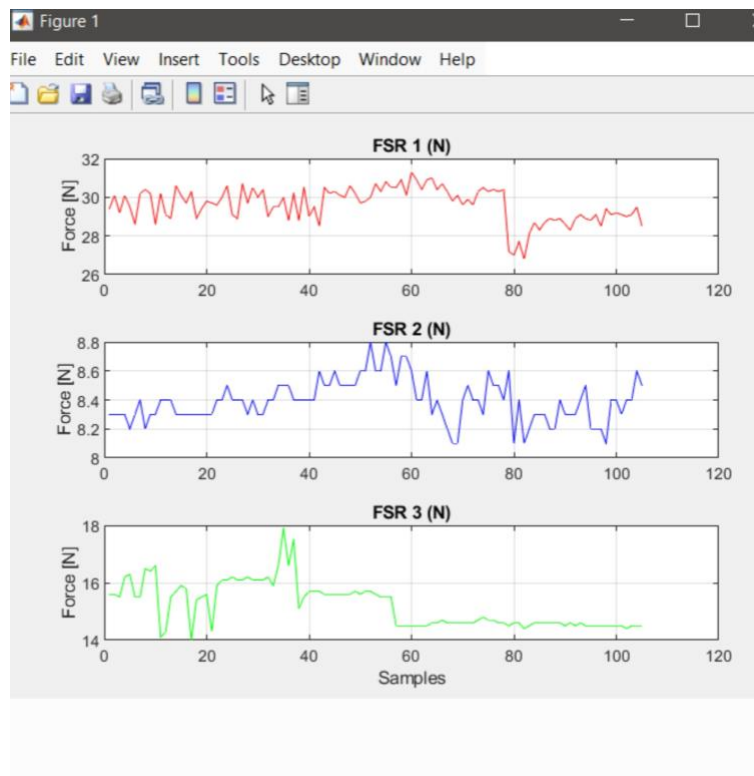
Εικόνα 54: Διάγραμμα σε matlab, πείραμα 1 (α)



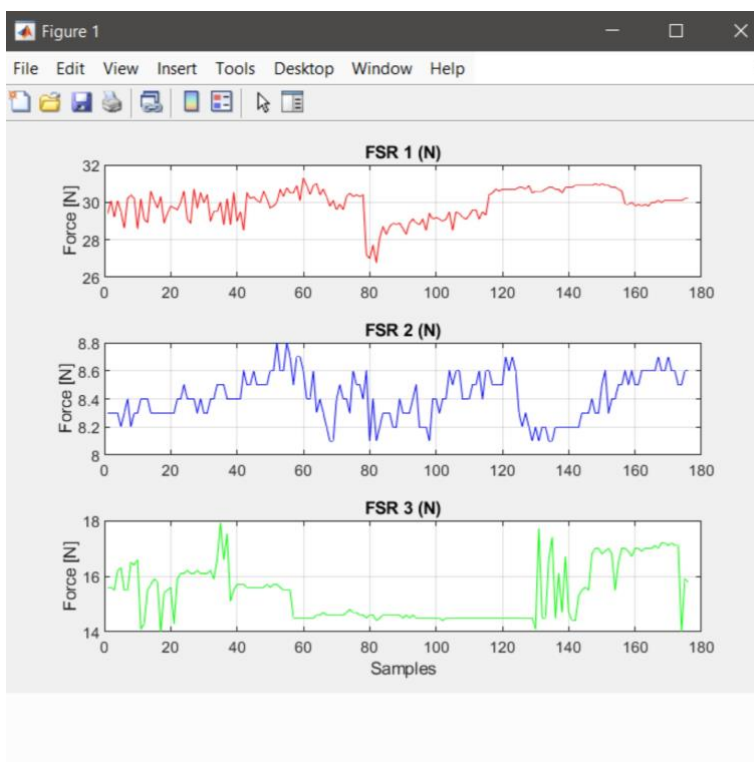
Εικόνα 58: Εφαρμογή, πείραμα 1 (β)



Εικόνα 56: Εφαρμογή, πείραμα 1 (γ)



Εικόνα 57: Διάγραμμα σε matlab, πείραμα 1 (β)



Εικόνα 55: Διάγραμμα σε matlab, πείραμα 1 (γ)

Με βάση τα παραπάνω αποτελέσματα, διαπιστώνεται ότι τα διαγράμματα στο MATLAB παρουσιάζουν συμφωνία με εκείνα της εφαρμογής, αποτυπώνοντας με ακρίβεια τις τιμές που αντιστοιχούν στις δυνάμεις οι οποίες ασκούνται στους αισθητήρες πίεσης κατά την τοποθέτηση του νάρθηκα στο σώμα (τα στιγμιότυπα λήφθηκαν κατά την διάρκεια εφαρμογής του νάρθηκα).

Ενώ παρακάτω αποτυπώνεται και το τελικό αποτέλεσμα που εμφανίζεται με το daily report και επιβεβαιώνει πως πράγματι ο στόχος επιτεύχθηκε για τα συγκεκριμένα δεδομένα.



Εικόνα 59: Μήνυμα για επίτευξη στόχου 1ου πειράματος

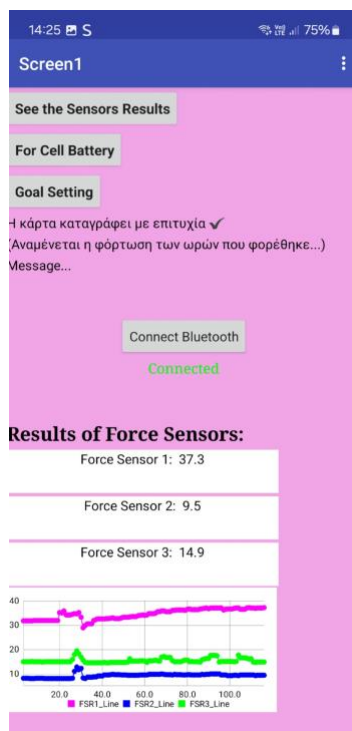
data4 - Σημειωματάριο	data4 - Σημειωματάριο
Αρχείο Επεξεργασία Μορφή Προβολή Βοήθεια	Αρχείο Επεξεργασία Μορφή Προβολή Βοήθεια
2025-08-25 02:15:56	2025-08-25 02:19:56
Force1(N): 28.7 Force2(N): 8.3 Force3(N): 14.6 Pressure1(kPa): 225.9 Pressure2(kPa): 65.6 Pressure3(kPa): 114.6	Force1(N): 31.0 Force2(N): 9.1 Force3(N): 17.7 Pressure1(kPa): 244.0 Pressure2(kPa): 71.4 Pressure3(kPa): 139.5
2025-08-25 02:15:58	2025-08-25 02:19:57
Force1(N): 28.3 Force2(N): 8.3 Force3(N): 14.6 Pressure1(kPa): 222.9 Pressure2(kPa): 65.4 Pressure3(kPa): 114.7	Force1(N): 31.1 Force2(N): 9.0 Force3(N): 17.6 Pressure1(kPa): 244.6 Pressure2(kPa): 71.2 Pressure3(kPa): 138.3
2025-08-25 02:15:59	2025-08-25 02:19:58
Force1(N): 28.7 Force2(N): 8.3 Force3(N): 14.6 Pressure1(kPa): 225.8 Pressure2(kPa): 65.0 Pressure3(kPa): 114.6	Force1(N): 31.1 Force2(N): 9.1 Force3(N): 18.1 Pressure1(kPa): 245.2 Pressure2(kPa): 71.5 Pressure3(kPa): 142.5
2025-08-25 02:16:00	2025-08-25 02:20:00
Force1(N): 28.9 Force2(N): 8.2 Force3(N): 14.6 Pressure1(kPa): 227.6 Pressure2(kPa): 64.3 Pressure3(kPa): 114.8	Force1(N): 31.2 Force2(N): 9.1 Force3(N): 18.0 Pressure1(kPa): 245.5 Pressure2(kPa): 71.8 Pressure3(kPa): 141.8
2025-08-25 02:16:01	===== DAILY REPORT ===== 2025-08-25 02:20:00 Total time under pressure: 0h 9m 18s ☒ Επιτυχία! Πέτυχες ή ξεπέρασες τον στόχο!

Εικόνα 60: Αντιστοίχιση μετρήσεων πειράματος 1 με το αρχείο .txt καταγραφής δεδομένων

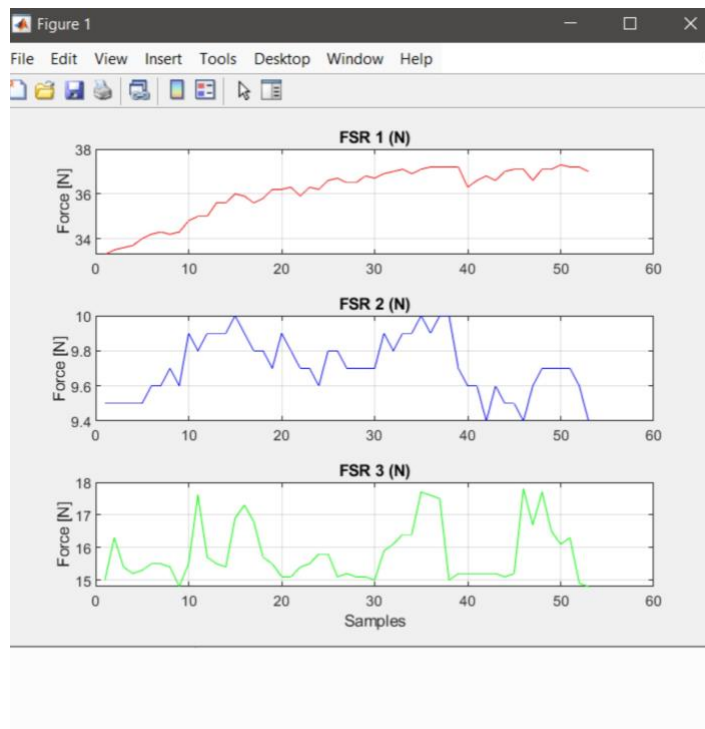
Για τα ίδια ακριβώς χρονικά διαστήματα παρατηρείται ότι οι μετρήσεις καταγράφονται με συνέπεια και στο αρχείο .txt της κάρτας SD, όπου εμφανίζεται επίσης το *daily report* στην ίδια χρονική στιγμή και με τα ορθά αποτελέσματα. Επιπλέον, διαπιστώνεται ότι η ώρα και η διάρκεια που αποτυπώνονται στο αρχείο αυτό συμφωνούν με εκείνες του κινητού τηλεφώνου, όπως τεκμηριώνεται από τα στιγμιότυπα που παρουσιάζονται στις ανωτέρω εικόνες, όπου η ώρα εμφανίζεται στην επάνω αριστερή γωνία.

2^η δοκιμή

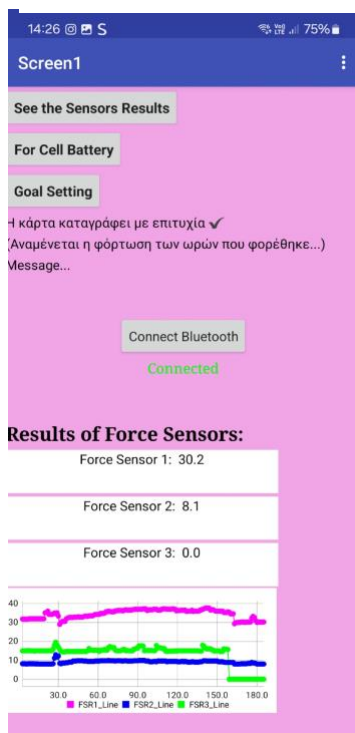
Περιγραφή: Σκοπός είναι ο νάρθηκας να τοποθετηθεί για χρόνο ≥ 2 λεπτών και $\leq 3^{ωv}$ λεπτών ώστε να βρίσκεται εντός της αποδεκτής απόκλισης, του ενός λεπτού, από τον στόχο.



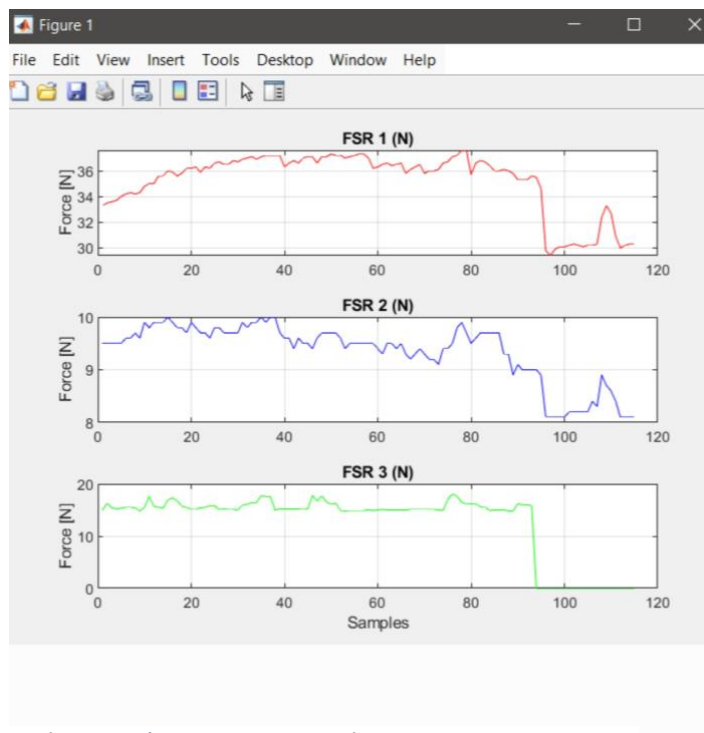
Εικόνα 64: Εφαρμογή, πείραμα 2 (α)



Εικόνα 63: Διάγραμμα σε matlab, πείραμα 2 (α)



Εικόνα 62: Εφαρμογής, πείραμα 2 (β)



Εικόνα 61: Διάγραμμα matlab, πείραμα 2 (β)



Εικόνα 65: Μήνυμα κοντά στην επίτευξη στόχου 2ου πειράματος

data4 - Σημειωματάριο	data4 - Σημειωματάριο
Αρχείο Επεξεργασία Μορφή Προβολή Βοήθεια	Αρχείο Επεξεργασία Μορφή Προβολή Βοήθεια
2025-08-25 02:25:57	2025-08-25 02:26:56
Force1(N): 36.0 Force2(N): 9.3 Force3(N): 14.8 Pressure1(kPa): 283.8 Pressure2(kPa): 73.2 Pressure3(kPa): 116.4	Force1(N): 30.8 Force2(N): 8.2 Force3(N): 0.0 Pressure1(kPa): 242.3 Pressure2(kPa): 64.2 Pressure3(kPa): 0.0
2025-08-25 02:25:58	2025-08-25 02:26:58
Force1(N): 35.8 Force2(N): 8.9 Force3(N): 14.8 Pressure1(kPa): 281.7 Pressure2(kPa): 69.7 Pressure3(kPa): 116.6	Force1(N): 30.8 Force2(N): 8.2 Force3(N): 0.0 Pressure1(kPa): 242.3 Pressure2(kPa): 64.6 Pressure3(kPa): 0.0
2025-08-25 02:25:59	2025-08-25 02:26:59
Force1(N): 35.3 Force2(N): 9.1 Force3(N): 16.2 Pressure1(kPa): 277.9 Pressure2(kPa): 71.8 Pressure3(kPa): 127.3	Force1(N): 30.8 Force2(N): 8.2 Force3(N): 0.0 Pressure1(kPa): 242.7 Pressure2(kPa): 64.8 Pressure3(kPa): 0.0
2025-08-25 02:26:00	2025-08-25 02:27:00
Force1(N): 35.3 Force2(N): 9.0 Force3(N): 16.0 Pressure1(kPa): 278.3 Pressure2(kPa): 70.6 Pressure3(kPa): 125.6	Force1(N): 30.9 Force2(N): 8.2 Force3(N): 0.0 Pressure1(kPa): 243.5 Pressure2(kPa): 64.7 Pressure3(kPa): 0.0
2025-08-25 02:26:01	===== DAILY REPORT ===== 2025-08-25 02:27:00 Total time under pressure: 0h 2m 52s △ Είσαι πολύ κοντά στον στόχο! Μπορείς λίγο ακόμα!

Εικόνα 66: Αντιστοίχιση μετρήσεων πειράματος 2 με το αρχείο .txt καταγραφής δεδομένων

Και στην παρούσα δοκιμή ισχύουν τα ευρήματα του πρώτου πειράματος, καθώς τα διαγράμματα της εφαρμογής παρουσιάζουν συμφωνία με εκείνα του MATLAB, ενώ οι αντίστοιχες τιμές καταγράφονται ομοίως και στο αρχείο .txt για το συγκεκριμένο χρονικό διάστημα. Το daily report εμφανίζεται ορθά, υποδεικνύοντας ότι η διάρκεια εφαρμογής είναι κοντά στον προκαθορισμένο στόχο των τριών λεπτών, εντός του επιτρεπτού ορίου απόκλισης του ενός λεπτού, δεδομένου ότι ο νάρθηκας φορέθηκε για 2 λεπτά και 52 δευτερόλεπτα.

Παράλληλα, μέσω της δοκιμής αυτής ελέγχθηκε και η περίπτωση εσφαλμένης τοποθέτησης αισθητήρα, όταν δηλαδή ένας από αυτούς καταγράφει δύναμη μικρότερη από 3 N. Σε αυτήν την περίπτωση ο χρόνος εφαρμογής δεν θα πρέπει να καταμετράται, καθώς το σύστημα θεωρεί ότι ο νάρθηκας δεν έχει τοποθετηθεί ορθά. Πράγματι, όπως προκύπτει από τα σχετικά διαγράμματα, ο αισθητήρας 3 (που αντιστοιχεί στις πράσινες γραμμές των διαγραμμάτων) αφαιρείται, με αποτέλεσμα να μην καταμετράται ως τοποθετημένος μετά το χρονικό σημείο των 2 λεπτών και 52 δευτερολέπτων, παρότι ο νάρθηκας συνέχισε να παραμένει στο σώμα για λίγο περισσότερο από 3 λεπτά.

Η παρατήρηση αυτή επιβεβαιώνεται τόσο από τα διαγράμματα όσο και από τα τελικά αποτελέσματα, αποδεικνύοντας ότι η συσκευή λειτουργεί αξιόπιστα και αποτρέπει την πιθανότητα λανθασμένης τοποθέτησης του νάρθηκα χωρίς να γίνει αντιληπτό

3^η δοκιμή

Περιγραφή: Σκοπός είναι ο νάρθηκας να τοποθετηθεί για χρόνο < 2 λεπτών ώστε να απέχει από τον στόχο των 3^{ων} λεπτών και να μην είναι εντός της αποδεκτής απόστασης.



Εικόνα 68: Μετρήσεις 3ου πειράματος



Εικόνα 67: Μήνυμα αποτυχίας επίτευξης στόχου 3ου πειράματος

Στο συγκεκριμένο πείραμα ο νάρθηκας τοποθετήθηκε εκ νέου, αυτή τη φορά όμως για μικρότερη χρονική διάρκεια (1 λεπτό και 12 δευτερόλεπτα), η οποία δεν εμπίπτει στο αποδεκτό όριο απόκλισης του ενός λεπτού από τον προκαθορισμένο στόχο των τριών λεπτών. Σκοπός της δοκιμής ήταν να διερευνηθεί κατά πόσο το daily report θα εμφάνιζε το κατάλληλο μήνυμα και σε αυτήν την περίπτωση, κάτι που πράγματι επιβεβαιώθηκε βάσει των σχετικών στιγμιότυπων.

Παράλληλα, πραγματοποιήθηκε εκ νέου έλεγχος της λειτουργικότητας που αφορά τη μη καταμέτρηση της διάρκειας εφαρμογής όταν δεν ασκείται πίεση μεγαλύτερη των 3 N και στους τρεις αισθητήρες ταυτόχρονα. Συγκεκριμένα, παρότι ο νάρθηκας συνέχισε να παραμένει τοποθετημένος, αφαιρέθηκε ο τρίτος αισθητήρας (απεικονιζόμενος με πράσινη γραμμή στα διαγράμματα). Παρατηρείται έτσι ότι, μολονότι μεσολάβησαν περίπου τρία λεπτά από την έναρξη των μετρήσεων (βλ. εικόνα 69 με καταγραφή ώρας 14:29 και εικόνα 68 με καταγραφή ώρας 14:32, που αντιστοιχεί στο χρονικό σημείο έκδοσης του daily report), ο συνολικός χρόνος εφαρμογής αποτυπώθηκε ως 1 λεπτό και

12 δευτερόλεπτα, καθώς από εκείνο το χρονικό σημείο και έπειτα ο αισθητήρας είχε αφαιρεθεί.

Κατ' αυτόν τον τρόπο, η δοκιμή προσομοίωσε επιτυχώς μια περίπτωση πρόωρης αφαίρεσης του νάρθηκα μέσω απενεργοποίησης ενός μόνο αισθητήρα, επιβεβαιώνοντας την ορθή λειτουργία του συστήματος.

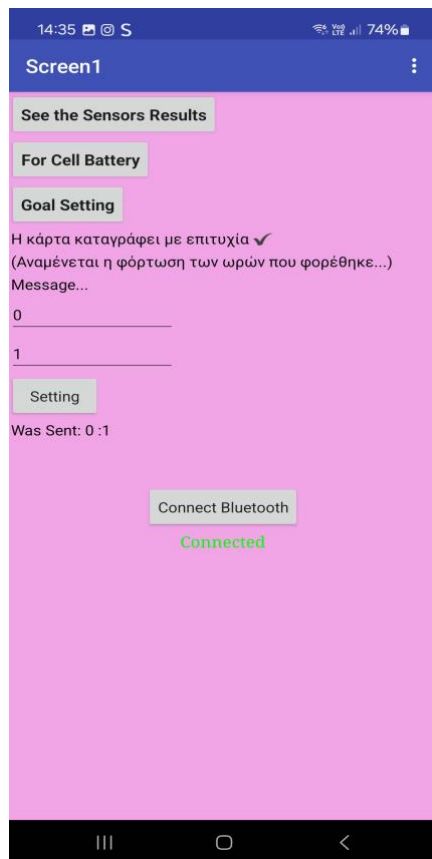
data4 - Σημειωματάριο	data4 - Σημειωματάριο
Αρχείο Επεξεργασία Μορφή Προβολή Βοήθεια Pressure1(kPa): 245.7 Pressure2(kPa): 77.8 Pressure3(kPa): 202.6	Αρχείο Επεξεργασία Μορφή Προβολή Βοήθεια Pressure1(kPa): 241.0 Pressure2(kPa): 79.5 Pressure3(kPa): 0.0
2025-08-25 02:29:49 Force1(N): 31.4 Force2(N): 9.8 Force3(N): 25.8 Pressure1(kPa): 247.1 Pressure2(kPa): 77.5 Pressure3(kPa): 203.0	2025-08-25 02:31:57 Force1(N): 30.6 Force2(N): 10.1 Force3(N): 0.0 Pressure1(kPa): 241.2 Pressure2(kPa): 79.7 Pressure3(kPa): 0.0
2025-08-25 02:29:51 Force1(N): 31.7 Force2(N): 9.8 Force3(N): 25.8 Pressure1(kPa): 249.9 Pressure2(kPa): 76.9 Pressure3(kPa): 202.8	2025-08-25 02:31:58 Force1(N): 30.7 Force2(N): 10.1 Force3(N): 0.0 Pressure1(kPa): 242.0 Pressure2(kPa): 79.8 Pressure3(kPa): 0.0
2025-08-25 02:29:52 Force1(N): 31.9 Force2(N): 9.7 Force3(N): 25.8 Pressure1(kPa): 251.0 Pressure2(kPa): 76.3 Pressure3(kPa): 202.8	2025-08-25 02:31:59 Force1(N): 30.8 Force2(N): 10.1 Force3(N): 0.0 Pressure1(kPa): 242.8 Pressure2(kPa): 79.7 Pressure3(kPa): 0.0
2025-08-25 02:29:53 Force1(N): 31.9 Force2(N): 9.5 Force3(N): 25.8 Pressure1(kPa): 251.4 Pressure2(kPa): 75.2 Pressure3(kPa): 203.0	2025-08-25 02:32:00 Force1(N): 30.8 Force2(N): 10.1 Force3(N): 0.0 Pressure1(kPa): 242.3 Pressure2(kPa): 79.7 Pressure3(kPa): 0.0
2025-08-25 02:29:54 Force1(N): 31.8 Force2(N): 8.8 Force3(N): 25.8 Pressure1(kPa): 250.8 Pressure2(kPa): 69.6 Pressure3(kPa): 203.1	===== DAILY REPORT ===== 2025-08-25 02:32:00 Total time under pressure: 0h 1m 12s X Προσπάθησε περισσότερο αύριο!

Εικόνα 69: Αντιστοίχιση μετρήσεων πειράματος 3 με το αρχείο .txt καταγραφής δεδομένων

Παρατηρείται ακόμα πως οι τιμές τις εφαρμογής συμβαδίζουν με αυτές που καταγράφονται στο αρχείο .txt καθώς και πως το daily report γίνεται την κατάλληλη ώρα εμφανίζοντας το σωστό μήνυμα.

4^η δοκιμή

Περιγραφή: Σκοπός της συγκεκριμένης δοκιμής ήταν η τοποθέτηση του νάρθηκα για χρονικό διάστημα μικρότερο των δύο λεπτών, ώστε η διάρκεια να αποκλίνει από τον προκαθορισμένο στόχο των τριών λεπτών και να μην εμπίπτει στο αποδεκτό όριο απόκλισης. Παράλληλα, μέσω της εφαρμογής τροποποιήθηκε ο στόχος σε μόλις ένα λεπτό, προκειμένου να εξεταστεί εάν το σύστημα θα αναγνωρίσει τη διάρκεια αυτή ως επιτυχή εφαρμογή και θα εμφανίσει το αντίστοιχο μήνυμα επιτυχίας.



Εικόνα 72: Αλλαγή στόχου μέσω εφαρμογής, 4ου πειράματος



Εικόνα 71: Μετρήσεις 4ου πειράματος



Εικόνα 70: Μήνυμα επίτευξης στόχου 4ου πειράματος

data4 - Σημειωματάριο	data4 - Σημειωματάριο
Αρχείο Επεξεργασία Μορφή Προβολή Βοήθεια	Αρχείο Επεξεργασία Μορφή Προβολή Βοήθεια
Force1(N): 31.9 Force2(N): 10.2 Force3(N): 23.8 Pressure1(kPa): 251.5 Pressure2(kPa): 80.0 Pressure3(kPa): 187.7	
2025-08-25 02:35:30	
Force1(N): 31.6 Force2(N): 10.2 Force3(N): 23.8 Pressure1(kPa): 248.9 Pressure2(kPa): 80.3 Pressure3(kPa): 187.5	
2025-08-25 02:35:32	
Force1(N): 32.0 Force2(N): 10.3 Force3(N): 23.9 Pressure1(kPa): 251.7 Pressure2(kPa): 80.9 Pressure3(kPa): 188.3	
2025-08-25 02:35:32 - Goal updated to: 0h 1m	
2025-08-25 02:35:33	
Force1(N): 32.0 Force2(N): 10.1 Force3(N): 24.0 Pressure1(kPa): 251.7 Pressure2(kPa): 79.8 Pressure3(kPa): 188.8	
2025-08-25 02:35:34	
Force1(N): 32.0 Force2(N): 10.1 Force3(N): 24.0 Pressure1(kPa): 252.0 Pressure2(kPa): 79.9 Pressure3(kPa): 189.0	
	2025-08-25 02:36:57
	Force1(N): 0.0 Force2(N): 0.0 Force3(N): 0.0 Pressure1(kPa): 0.0 Pressure2(kPa): 0.0 Pressure3(kPa): 0.0
	2025-08-25 02:36:58
	Force1(N): 0.0 Force2(N): 0.0 Force3(N): 0.0 Pressure1(kPa): 0.0 Pressure2(kPa): 0.0 Pressure3(kPa): 0.0
	2025-08-25 02:36:59
	Force1(N): 0.0 Force2(N): 0.0 Force3(N): 0.0 Pressure1(kPa): 0.0 Pressure2(kPa): 0.0 Pressure3(kPa): 0.0
	2025-08-25 02:37:00
	Force1(N): 0.0 Force2(N): 0.0 Force3(N): 0.0 Pressure1(kPa): 0.0 Pressure2(kPa): 0.0 Pressure3(kPa): 0.0
	===== DAILY REPORT ===== 2025-08-25 02:37:00 Total time under pressure: 0h 1m 26s ☑️ Επιτυχία! Πέτυχε ή ξεπέρασε τον στόχο!

Εικόνα 73: Αντιστοίχιση μετρήσεων πειράματος 4 με το αρχείο .txt καταγραφής δεδομένων

Στο τέταρτο πείραμα παρατηρείται ότι οι μετρήσεις πραγματοποιούνται κανονικά, με τους αισθητήρες να καταγράφουν με ακρίβεια τις εφαρμοζόμενες δυνάμεις και τα δεδομένα να παρουσιάζουν συνέπεια με εκείνα που αποθηκεύονται στο αρχείο .txt. Στη συγκεκριμένη περίπτωση, ο νάρθηκας φορέθηκε για 1 λεπτό και 26 δευτερόλεπτα και στη συνέχεια αφαιρέθηκε πλήρως από το σώμα. Το daily report αποτύπωσε ορθά τη διάρκεια αυτής της εφαρμογής, καταγράφοντας τον ακριβή χρόνο τοποθέτησης.

Παράλληλα, κατά τη διάρκεια που ο νάρθηκας ήταν τοποθετημένος, πραγματοποιήθηκε μέσω της εφαρμογής αλλαγή του προκαθορισμένου στόχου, ο οποίος τροποποιήθηκε στο 1 λεπτό. Η αλλαγή αυτή εφαρμόστηκε επιτυχώς, καθώς καταγράφηκε τόσο με το αντίστοιχο μήνυμα στην οθόνη του κινητού όσο και στο αρχείο .txt, όπου αναφέρεται με σαφήνεια το χρονικό σημείο της τροποποίησης καθώς και η νέα τιμή του στόχου. Το γεγονός αυτό επιβεβαιώνεται επιπλέον από το daily report, το οποίο καταγράφει επιτυχή εφαρμογή με διάρκεια 1 λεπτό και 26 δευτερόλεπτα, κάτι που δεν θα ίσχυε εάν ο στόχος δεν είχε μεταβληθεί (δεδομένου ότι προηγουμένως ήταν ορισμένος στα 3 λεπτά).

Επιπλέον, καθίσταται σαφές ότι μετά την πλήρη αφαίρεση του νάρθηκα οι τιμές και των τριών αισθητήρων μηδενίζονται αυτόματα, γεγονός που αποτυπώνεται τόσο στο διάγραμμα της εικόνας 71 όσο και στο αρχείο .txt λίγο πριν την καταγραφή του daily report.

8.1 Συμπεράσματα Πειραματικών Δοκιμών

Από τη συνολική αξιολόγηση του συστήματος μέσω των τεσσάρων πειραματικών δοκιμών προκύπτουν ορισμένα καίρια συμπεράσματα που τεκμηριώνουν τη λειτουργικότητα και την αξιοπιστία του:

- **Αξιοπιστία μετρήσεων αισθητήρων:** Οι αισθητήρες πίεσης (FSRs) κατέγραψαν με συνέπεια τις εφαρμοζόμενες δυνάμεις, με τις τιμές να αποτυπώνονται ορθά τόσο στην εφαρμογή όσο και στο αρχείο καταγραφής της κάρτας SD. Η σύγκλιση των διαγραμμάτων της εφαρμογής με εκείνα που παρήχθησαν στο MATLAB ενισχύει την εγκυρότητα των μετρήσεων.
- **Ορθότητα χρονικού συγχρονισμού:** Το RTC module λειτούργησε με ακρίβεια, καθώς τα χρονικά αποτυπώματα στις μετρήσεις συμβάδιζαν με τις πραγματικές χρονικές στιγμές, όπως επιβεβαιώθηκε και από τα στιγμιότυπα της φορητής συσκευής.

- **Επιτυχής λειτουργία του *daily report*:** Σε όλες τις δοκιμές το σύστημα παρήγαγε τα κατάλληλα μηνύματα, είτε επρόκειτο για επιτυχή είτε για μη επιτυχή επίτευξη του στόχου. Το γεγονός αυτό καταδεικνύει την ορθή παραμετροποίηση των χρονικών ορίων και την ακριβή αξιολόγηση των δεδομένων.
- **Ασφάλεια στη χρήση – αποτροπή λανθασμένης τοποθέτησης:** Διαπιστώθηκε ότι σε περιπτώσεις ελλιπούς ή εσφαλμένης εφαρμογής, όπου ένας ή περισσότεροι αισθητήρες κατέγραφαν δύναμη μικρότερη από 3 N, ο χρόνος εφαρμογής δεν καταμετρούταν. Αυτό διασφαλίζει ότι το σύστημα δεν αναγνωρίζει λανθασμένα τον νάρθηκα ως τοποθετημένο, μειώνοντας τον κίνδυνο ψευδών μετρήσεων.
- **Δυναμική αλλαγή παραμέτρων μέσω εφαρμογής:** Ιδιαίτερα στο τέταρτο πείραμα, επαληθεύτηκε η δυνατότητα τροποποίησης του στόχου σε πραγματικό χρόνο, με την αλλαγή να καταγράφεται και να αποτυπώνεται σωστά τόσο στην εφαρμογή όσο και στο αρχείο καταγραφής. Η λειτουργία αυτή ενισχύει την ευελιξία και την προσαρμοστικότητα του συστήματος.
- **Διαχείριση ενέργειας:** Κατά τη λειτουργία με μπαταρία λιθίου, το σύστημα παρουσίασε ικανοποιητική αυτονομία και καλή συνεργασία με το κύκλωμα φόρτισης, επιβεβαιώνοντας την καταλληλότητά του για φορητές εφαρμογές.

Συνολικά, οι δοκιμές κατέδειξαν ότι το υλοποιημένο σύστημα λειτουργεί με συνέπεια, ακρίβεια και ασφάλεια, παρέχοντας αξιόπιστα δεδομένα και ενδείξεις που το καθιστούν κατάλληλο για πρακτική χρήση και περαιτέρω εφαρμογές σε πραγματικές συνθήκες.

9 Συμπεράσματα

Η παρούσα εργασία είχε ως αντικείμενο τον σχεδιασμό, την ανάπτυξη και την αξιολόγηση ενός ολοκληρωμένου, φορητού συστήματος παρακολούθησης της συμμόρφωσης ασθενών με εφηβική ιδιοπαθή σκολίωση (AIS) στη θεραπεία μέσω νάρθηκα. Μέσα από την ανάλυση του προβλήματος, τη μελέτη της υπάρχουσας βιβλιογραφίας και τις πειραματικές δοκιμές, προέκυψαν ορισμένα κρίσιμα συμπεράσματα.

Αρχικά, αναδείχθηκε η ανάγκη για τεχνολογικές λύσεις που θα ενισχύουν τη συμμόρφωση των ασθενών, αντιμετωπίζοντας τις δυσκολίες που συχνά προκύπτουν από τη μακροχρόνια χρήση του νάρθηκα. Η συσκευή που αναπτύχθηκε βασίστηκε σε προσεκτικά επιλεγμένα ηλεκτρονικά εξαρτήματα, όπως αισθητήρες πίεσης, μονάδα RTC, κάρτα SD και Bluetooth module, τα οποία συνδυάστηκαν σε ένα σταθερό και αξιόπιστο σύστημα. Παράλληλα, η ανάπτυξη μιας εφαρμογής μέσω MIT App Inventor προσέφερε ένα φιλικό περιβάλλον παρακολούθησης και διαχείρισης των δεδομένων, επιτρέποντας την εύκολη οπτικοποίηση και την άμεση ενημέρωση του ασθενούς.

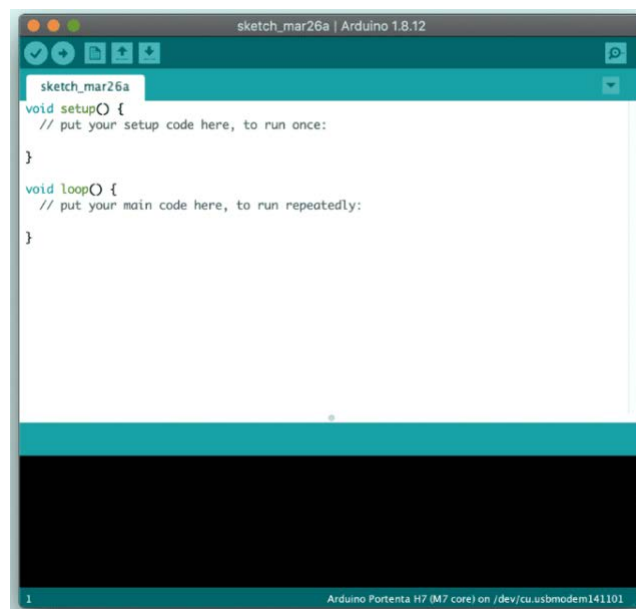
Οι πειραματικές δοκιμές κατέδειξαν ότι το σύστημα ανταποκρίνεται με ακρίβεια στις απαιτήσεις: οι αισθητήρες κατέγραφαν αξιόπιστα τις δυνάμεις, τα δεδομένα συγχρονίζονταν χρονικά και αποθηκεύονταν χωρίς απώλειες, ενώ το *daily report* παρείχε σαφή και τεκμηριωμένη ανατροφοδότηση για τη διάρκεια και την ποιότητα της εφαρμογής. Επιπλέον, το σύστημα απέδειξε την ευελιξία του, επιτρέποντας δυναμική αλλαγή παραμέτρων (όπως ο στόχος διάρκειας εφαρμογής) και εξασφαλίζοντας την αποφυγή ψευδών καταγραφών σε περίπτωση ελλιπούς ή λανθασμένης τοποθέτησης του νάρθηκα.

Συνολικά, η εργασία αυτή αποδεικνύει ότι είναι εφικτή η ανάπτυξη μιας χαμηλού κόστους, φορητής και αξιόπιστης συσκευής, η οποία μπορεί να συμβάλει ουσιαστικά στην ενίσχυση της συμμόρφωσης των ασθενών με AIS, προσφέροντας ταυτόχρονα στους ιατρούς και θεραπευτές ένα ισχυρό εργαλείο παρακολούθησης και αξιολόγησης. Η προτεινόμενη λύση δημιουργεί προοπτικές για περαιτέρω έρευνα και μελλοντικές επεκτάσεις, όπως η βελτίωση της εργονομίας, η ενσωμάτωση επιπλέον βιοσημάτων και η ανάπτυξη πιο εξελιγμένων αλγορίθμων ανάλυσης δεδομένων.

Α. Παράρτημα Κώδικα σε περιβάλλον Arduino IDE

Πληροφορίες Προγράμματος Δημιουργίας Κώδικα

Το Arduino IDE (Integrated Development Environment) είναι λογισμικό που χρησιμοποιείται για τη σύνταξη και τη μεταφόρτωση κώδικα σε πλακέτες Arduino αλλά και άλλων μικροελεγκτών. Παρέχει ένα πρόγραμμα επεξεργασίας κειμένου για τη σύνταξη κώδικα, έναν μεταγλωττιστή για τη μετατροπή του σε μορφή που μπορεί να κατανοήσει ο πύρακλος και εργαλεία για τη μεταφόρτωση του κώδικα στο υλικό. Το IDE είναι διαθέσιμο τόσο ως εφαρμογή υπολογιστή όσο και ως ηλεκτρονικός επεξεργαστής.



Εικόνα 74: Περιβάλλον Arduino IDE

Κάθε κώδικας στο περιβάλλον προγραμματισμού Arduino IDE, περιέχει τις συναρτήσεις **setup()** και **loop()**, οι οποίες αποτελούν βασικά δομικά στοιχεία κάθε προγράμματος. Η συνάρτηση **setup()** εκτελείται μία μόνο φορά κατά την έναρξη του σκίτσου (δηλαδή του προγράμματος με τον κώδικα), με σκοπό την αρχικοποίηση των μεταβλητών, τη ρύθμιση των καταστάσεων των ακροδεκτών και τη διαμόρφωση των απαραίτητων επικοινωνιών. Έπειτα, η συνάρτηση **loop()** εκτελείται συνεχώς σε βρόχο, ενσωματώνοντας τον κεντρικό αλγόριθμο και τις λειτουργικές διεργασίες του προγράμματος. [76]

Στο περιβάλλον ανάπτυξης του λογισμικού **Arduino IDE**, τα προγράμματα που δημιουργούνται ονομάζονται *sketches*. Τα εν λόγω αρχεία συντάσσονται μέσω του ενσωματωμένου επεξεργαστή κειμένου και αποθηκεύονται με την επέκταση *.ino*. Ο επεξεργαστής προσφέρει βασικές λειτουργίες επεξεργασίας, όπως αποκοπή, επικόλληση, καθώς και εργαλεία αναζήτησης και αντικατάστασης κειμένου.

Η περιοχή μηνυμάτων διαδραματίζει υποστηρικτικό ρόλο, παρέχοντας ενημερώσεις και ειδοποιήσεις σχετικά με την αποθήκευση ή τη μεταγλώττιση του κώδικα, ενώ επιπλέον εμφανίζει τυχόν σφάλματα που προκύπτουν κατά την εκτέλεση. Συμπληρωματικά, η κονσόλα του IDE εμφανίζει αναλυτικά μηνύματα και πληροφορίες, συμπεριλαμβανομένων των διαγνωστικών δεδομένων και των σφαλμάτων μεταγλώττισης.

Στο κάτω δεξί τμήμα του παραθύρου εμφανίζονται πληροφορίες σχετικά με την επιλεγμένη πλακέτα ανάπτυξης (board) και τη σειριακή θύρα επικοινωνίας. Τέλος, η γραμμή εργαλείων παρέχει άμεση πρόσβαση σε βασικές λειτουργίες, όπως η επαλήθευση και η μεταφόρτωση των *sketch*, η δημιουργία νέων αρχείων, το άνοιγμα και η αποθήκευση υπαρχόντων, καθώς και η ενεργοποίηση της σειριακής οθόνης για επικοινωνία με τη συσκευή. [77]



Εικόνα 75: Επαλήθευση Ελέγχει τον κώδικά για σφάλματα κατά τη μεταγλώττιση.



Εικόνα 76: Μεταφόρτωση Μεταγλωττίζει τον κώδικά σας και τον μεταφορτώνει στην διαμορφωμένη πλακέτα.

Για την εγκατάσταση των απαιτούμενων βιβλιοθηκών στο περιβάλλον του Arduino IDE, ο χρήστης μεταβαίνει στο εικονίδιο των βιβλιοθηκών που βρίσκεται στην αριστερή πλευρά της διεπαφής. Στη συνέχεια, μέσω του πεδίου αναζήτησης με την ένδειξη *“Filter your search”*, εισάγει την ονομασία της επιθυμητής βιβλιοθήκης. Μετά τον εντοπισμό της σχετικής καταχώρησης, η διαδικασία ολοκληρώνεται με την επιλογή του πλήκτρου *Install*, το οποίο ενεργοποιεί την εγκατάσταση της βιβλιοθήκης στο σύστημα ανάπτυξης.



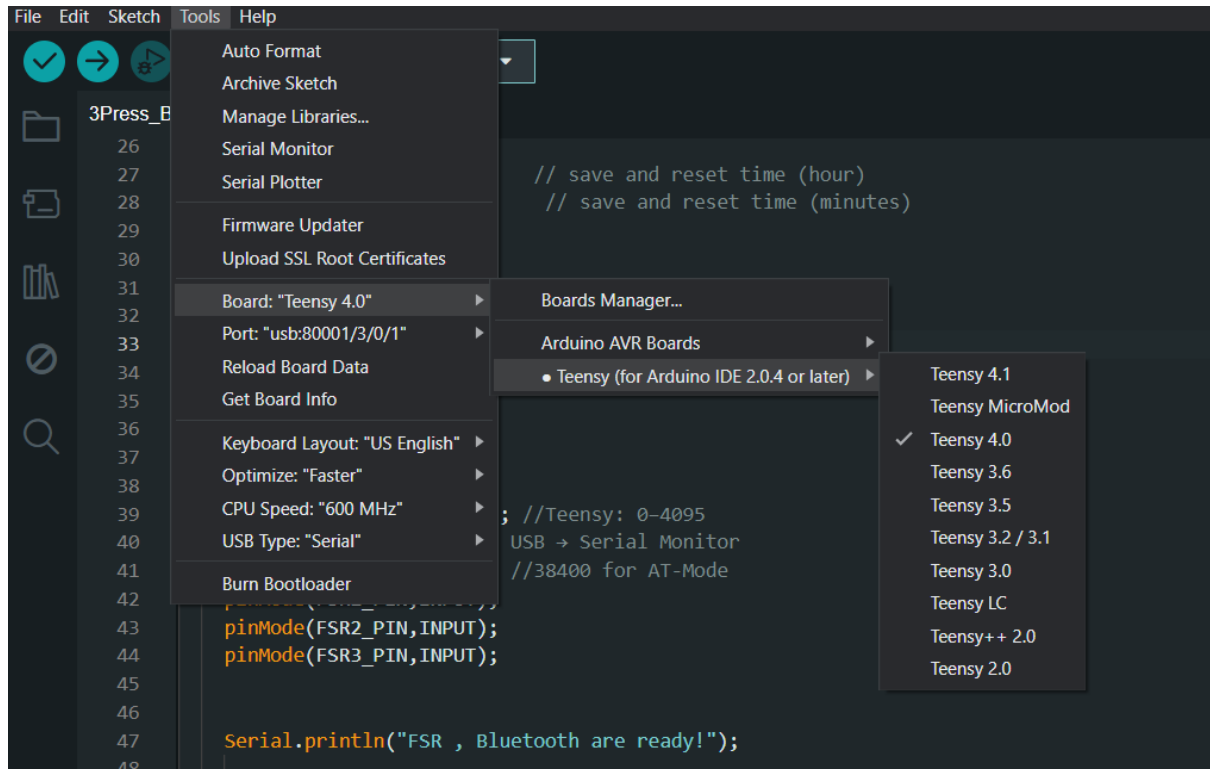
Εικόνα 77: Εισαγωγή Βιβλιοθηκών στο Arduino IDE

Κάτω δεξιά επιλέγουμε στο Serial Monitor την επιλογή Both NL & CR



Εικόνα 78: Επιλογή Serial Monitor στο Arduino IDE

Στη γραμμή μενού, στο επάνω μέρος του περιβάλλοντος ανάπτυξης, επιλέγουμε **Tools → Board: ... → Boards Manager**. Στη συνέχεια, από τη λίστα των διαθέσιμων πλακετών, γίνεται η επιλογή του μικροελεγκτή που πρόκειται να χρησιμοποιηθεί, ο οποίος στην παρούσα περίπτωση είναι ο **Teensy 4.0**.



Εικόνα 79: Επιλογή μικροελεγκτή

Παράρτημα συνολικού κώδικα

```
#include <SD.h>

#include <SPI.h>

#include <Wire.h>

#include <RTCLib.h>

#define SENSOR_AREA 0.000127 // m2 (for FSR 0.5" with approximately 1.27 cm2 surface area)

#define READ_INTERVAL 1000 // π.χ. 100 ms

unsigned long lastRead = 0;
```

```
const int SD_CS_PIN= 10; // Chip Select for SD card

File dataFile; // File for save in SD card

RTC_DS3231 rtc; // battery model


//For SD, Connection-Disconnection

bool sdOk = false; //boolean variable for SD card, true-> records| false-> fails to register

unsigned long lastSdCheck = 0;

const unsigned long sdCheckInterval = 2000;


const int FSR1_PIN = A0; // force sensor to analog A0

const int FSR2_PIN = A1; // force sensor to analog A1

const int FSR3_PIN = A2; // force sensor to analog A2


// Pressure timing variables

unsigned long pressureStart= 0;

unsigned long totalPressureTime= 0;

bool pressureActive= false;

int cutoffHour= 0;      // save and reset time (hour)

int cutoffMinute= 0;    // save and reset time (minutes)

bool reportedToday= false;


//for goals of the day

int goalHours = 15;     //goal-> 15h

int goalMinutes = 0;    // for examble

double deviationHours = 0.016667; // deviation from target


void setup()

{
```



```
analogReadResolution(12); //Teensy: 0–4095

Serial.begin(9600); // USB → Serial Monitor

Serial1.begin(9600); //38400 for AT-Mode

pinMode(FSR1_PIN,INPUT);

pinMode(FSR2_PIN,INPUT);

pinMode(FSR3_PIN,INPUT);


Serial.println("FSR , Bluetooth are ready!");


if (!SD.begin(SD_CS_PIN))

{

    Serial.println("SD card initialization failed!");

    sdOk = false;

    lastSdCheck = millis();

}

else {

    Serial.println("SD card is ready.");

    sdOk = true;

}


dataFile= SD.open("data4.txt", FILE_WRITE);

if (dataFile)

{

    // Title for SD card

    dataFile.println("RESULTS:");

    dataFile.close();

} else
```

```
{  
  
  Serial.println("Failed to open file for writing!");  
  
  Serial1.println("SD ERROR: Failed to open file!" );  
  
  sdOk = false;  
  
}  
  
  
//For real-time date and hour  
  
Wire.begin();  
  
  
if (!rtc.begin())  
{  
  Serial.println("RTC not found");  
  
  while (1);  
}  
  
  
if (rtc.lostPower())  
{  
  
  Serial.println("RTC lost power, setting time!");  
  
  dataFile=SD.open("data4.txt",FILE_WRITE);  
  
  
  //save this and SD card  
  
  if (dataFile)  
  {  
  
    dataFile.println("RTC lost power, setting time!");  
  
    dataFile.close();  
  
  }  
}
```

```
    else {  
        Serial.println("Failed to open file to log RTC lost power!");  
    }  
  
    rtc.adjust(DateTime(F(__DATE__), F(__TIME__))); // Computer Time  
}  
  
//One Time for adjusting  
//rtc.adjust(DateTime(F(__DATE__), F(__TIME__)));  
  
}  
  
float readSmoothedFSR(int pin, int samples = 10, int delayMs = 5)  
{  
    long sum=0;  
    for (int i=0; i < samples; i++)  
    {  
        sum += analogRead(pin);  
        delay(delayMs);  
    }  
    return float(sum)/samples;  
}  
  
void checkForBluetoothCommands()  
{  
    if (Serial1.available())
```

```
{  
  
String input = Serial1.readStringUntil('\n');  
  
input.trim();  
  
if (input.startsWith("SETTIME:"))  
{  
    String datetime = input.substring(8); // break "SETTIME:"  
  
    int year = datetime.substring(0, 4).toInt();  
    int month = datetime.substring(5, 7).toInt();  
    int day = datetime.substring(8, 10).toInt();  
    int hour = datetime.substring(11, 13).toInt();  
    int minute = datetime.substring(14, 16).toInt();  
    int second = datetime.substring(17, 19).toInt();  
  
    rtc.adjust(DateTime(year, month, day, hour, minute, second));  
    Serial.println("RTC updated from Bluetooth!");  
    Serial1.println("RTC updated from Bluetooth!");  
  
    dataFile = SD.open("data4.txt", FILE_WRITE);  
  
    if (dataFile)  
    {  
        DateTime now=rtc.now();  
        dataFile.print( now.timestamp( DateTime::TIMESTAMP_DATE));  
        dataFile.print(" ");  
        dataFile.print( now.timestamp(DateTime::TIMESTAMP_TIME));  
    }  
}
```

```
        dataFile.print(" - ");

        dataFile.println("RTC updated from Bluetooth!");

        dataFile.close();

    } else

    {

        Serial.println( "Failed to open file to log RTC update from Bluetooth!" );

    }

}

else if(input.startsWith("SETGOAL:"))

{

    Serial.println("Detected SETGOAL command");

    int firstColon= input.indexOf(':');

    int secondColon= input.indexOf(':', firstColon + 1);

    if(firstColon != -1 && secondColon != -1)

    {

        String hStr= input.substring(firstColon + 1, secondColon);

        String mStr= input.substring(secondColon + 1);

        int h= hStr.toInt();

        int m= mStr.toInt();

        if(h >= 0 && m >= 0 && m < 60)

        {

            goalHours = h;
```

```
goalMinutes = m;

Serial.print("Goal time updated: ");

Serial.print(goalHours);

Serial.print("h ");

Serial.print(goalMinutes);

Serial.println("m");

Serial1.print("Goal updated: ");

Serial1.print(goalHours);

Serial1.print("h ");

Serial1.print(goalMinutes);

Serial1.println("m");

// for SD

dataFile = SD.open("data4.txt", FILE_WRITE);

if (dataFile) {

    DateTime now = rtc.now();

    dataFile.print(now.timestamp(DateTime::TIMESTAMP_DATE));

    dataFile.print(" ");

    dataFile.print(now.timestamp(DateTime::TIMESTAMP_TIME));

    dataFile.print(" - Goal updated to: ");

    dataFile.print(goalHours);

    dataFile.print("h ");

    dataFile.print(goalMinutes);

    dataFile.println("m");

    dataFile.close();

} else {
```

```
        Serial.println("Could not open data4.txt to log goal update.");
    }
} else {
    Serial.println("Invalid time: " + hStr + ":" + mStr);
    Serial1.println("Invalid time:.");
}
} else {
    Serial.println("Incorrect command format SETGOAL.");
    Serial1.println("Incorrect command format.");
}
}
}
}

void loop()
{
    //forward any AT-commands (if you ever enter AT-Mode)
    if (Serial.available())
    {
        Serial1.write(Serial.read());
    }
    if (Serial1.available())
    {
        Serial.write(Serial1.read());
    }

    if (millis() - lastRead >= READ_INTERVAL)
    {
```

```
lastRead = millis();

float raw1= readSmoothedFSR(FSR1_PIN);

float voltage1= raw1 * (3.3 / 4095.0);

float raw2= readSmoothedFSR(FSR2_PIN);

float voltage2 = raw2 * (3.3 / 4095.0);

float raw3= readSmoothedFSR(FSR3_PIN);

float voltage3 = raw3 * (3.3 / 4095.0);

float forceN1= (voltage1 >= 0.1) ? pow(voltage1/ 3.3, 2.5) * 100.0 : 0.0;
float forceN2= (voltage2 >= 0.1) ? pow(voltage2/ 3.3, 2.5) * 100.0 : 0.0;
float forceN3= (voltage3 >= 0.1) ? pow(voltage3/ 3.3, 2.5) * 100.0 : 0.0;

float press1kPa= (forceN1 / SENSOR_AREA)/ 1000.0;
float press2kPa= (forceN2 / SENSOR_AREA)/ 1000.0;
float press3kPa= (forceN3 / SENSOR_AREA)/ 1000.0;

bool allAboveThreshold = (forceN1 > 3.0 && forceN2 > 3.0 && forceN3 > 3.0);

if (allAboveThreshold)
{
    if (!pressureActive)
    {
        pressureStart = millis();
        pressureActive = true;
    }
}
```



```
    } else {  
        if (pressureActive)  
        {  
            totalPressureTime += millis() - pressureStart;  
            pressureActive = false;  
        }  
    }  
}
```

```
Serial.print("Force1(N): ");
```

```
Serial.print(forceN1,1);
```

```
Serial.print(" Pressure1(kPa): ");
```

```
Serial.print(press1kPa,1);
```

```
Serial.print(" Raw1: ");
```

```
Serial.print(raw1);
```

```
Serial.print(" V1: ");
```

```
Serial.println(voltage1, 3);
```

```
Serial.print("Force2(N): ");
```

```
Serial.print(forceN2,1);
```

```
Serial.print(" Pressure2(kPa): ");
```

```
Serial.print(press2kPa,1);
```

```
Serial.print(" Raw2: ");
```

```
Serial.print(raw2);
```

```
Serial.print(" V2: ");
```

```
Serial.println(voltage2, 3);
```

```
Serial.print("Force3(N): ");
```

```
Serial.print(forceN3,1);
```

```
Serial.print(" Pressure3(kPa): ");
```

```
Serial.print(press3kPa,1);
```

```
Serial.print(" Raw3: ");
```

```
Serial.print(raw3);
```

```
Serial.print(" V3: ");
```

```
Serial.println(voltage3, 3);
```

```
//For Bluetooth
```

```
Serial1.println();
```

```
Serial1.print(forceN1, 1);
```

```
Serial1.print("/");
```

```
Serial1.print(forceN2, 1);
```

```
Serial1.print("/");
```

```
Serial1.print(forceN3, 1);
```

```
Serial1.println();
```

```
Serial.println("Opening file...");
```

```
unsigned long now = millis();

if (!sdOk && (now- lastSdCheck>sdCheckInterval))

{

    Serial.println("Trying to re-init SD...");

    if (SD.begin(SD_CS_PIN))

    {

        Serial.println("SD re-init OK" );

        Serial1.println("SD OK");

        sdOk= true;

        DateTime now= rtc.now();

        File resumeFile= SD.open("data4.txt", FILE_WRITE);

        if(resumeFile)

        {

            resumeFile.print(now.timestamp(DateTime::TIMESTAMP_DATE));

            resumeFile.print(" ");

            resumeFile.print(now.timestamp(DateTime::TIMESTAMP_TIME));

            resumeFile.println(" ===== RECORDING STARTED AGAIN =====");

            resumeFile.close();

        }

    }else{

        Serial.println("Still no SD");

        Serial1.println("SD ERROR");

    }

    lastSdCheck= millis();

}
```

```
}

if (sdOk)
{
    if (!SD.begin(SD_CS_PIN))
    {
        Serial.println("SD.begin() failed just before writing!");
        Serial1.println("SD ERROR");
        sdOk= false;
        lastSdCheck= millis();

    } else {

        dataFile= SD.open("data4.txt",FILE_WRITE);

        if (dataFile)
        {
            DateTime now = rtc.now();
            dataFile.println( );
            dataFile.print(now.timestamp(DateTime::TIMESTAMP_DATE));
            dataFile.print(" ");
            dataFile.println(now.timestamp(DateTime::TIMESTAMP_TIME));
            dataFile.println( );

            dataFile.print("Force1(N): ");
            dataFile.print(forceN1,1);

            dataFile.print(" Force2(N): ");
            dataFile.print(forceN2,1);
```

```
dataFile.print(" Force3(N): ");  
  
dataFile.println(forceN3,1);  
  
  
dataFile.print("Pressure1(kPa): ");  
dataFile.print(press1kPa,1);  
dataFile.print(" Pressure2(kPa): ");  
dataFile.print(press2kPa,1);  
dataFile.print(" Pressure3(kPa): ");  
dataFile.println(press3kPa,1);  
dataFile.println();  
  
dataFile.close();  
Serial.println("Wrote & closed.");  
Serial1.print("OK SD");  
  
} else  
{  
    Serial.println("Failed to open file for writing! ");  
    Serial1.println("SD ERROR");  
    sdOk = false;  
    lastSdCheck = millis();  
}  
}  
}  
  
}  
  
  
//For real-time date and hour
```

```
DateTime now= rtc.now();

Serial.print(now.timestamp(DateTime::TIMESTAMP_DATE));

Serial.print(" ");

Serial.println(now.timestamp(DateTime::TIMESTAMP_TIME));

dataFile.flush();


checkForBluetoothCommands();


if (now.hour()== cutoffHour && now.minute()==cutoffMinute && !reportedToday)
{
    if (pressureActive)
    {
        totalPressureTime += millis()-pressureStart;
        pressureActive= false;
    }


    unsigned long totalSeconds= totalPressureTime/1000;

    unsigned int hours= totalSeconds/3600;

    unsigned int minutes= (totalSeconds%3600)/60;

    unsigned int seconds=totalSeconds%60;


    Serial.println("===== DAILY REPORT =====");

    Serial.print("Total time under pressure: ");

    Serial.print(hours);

    Serial.print("h ");

    Serial.print(minutes);

    Serial.print("m ");
```

```
Serial.print(seconds);

Serial.println("s");

//total time under pressure for Bluetooth

Serial1.print("Total time is Ready");

Serial1.print("Total time under pressure: ");

Serial1.print(hours);

Serial1.print("h ");

Serial1.print(minutes);

Serial1.print("m ");

Serial1.print(seconds);

Serial1.println("s");

Serial1.println("*");


// goal calculation

unsigned long goalSeconds= (goalHours*3600UL) + (goalMinutes*60UL);

long deltaSeconds=long(totalSeconds)-long(goalSeconds);    //can be negative


if (totalSeconds >= goalSeconds)
{
    Serial.println("Επιτυχία! Πέτυχες ή ξεπέρασες τον στόχο!");
    Serial1.println("SUCCES");
}

else if (abs(deltaSeconds) <= deviationHours * 3600)
{
    Serial.println("Είσαι πολύ κοντά στον στόχο! Μπορείς λίγο ακόμα!");
    Serial1.println("LITTLE MORE");
} else {
```

```
Serial.println("Προσπάθησε περισσότερο αύριο!");

Serial1.println("NO");

}

// Save to SD

if(sdOk && SD.begin(SD_CS_PIN))

{

    dataFile = SD.open("data4.txt",FILE_WRITE);

    if(dataFile)

    {

        dataFile.println("===== DAILY REPORT =====");

        dataFile.print(now.timestamp(DateTime::TIMESTAMP_DATE));

        dataFile.print(" ");

        dataFile.println(now.timestamp(DateTime::TIMESTAMP_TIME));

        dataFile.print("Total time under pressure: ");

        dataFile.print(hours);

        dataFile.print("h ");

        dataFile.print(minutes);

        dataFile.print("m ");

        dataFile.print(seconds);

        dataFile.println("s");

        dataFile.println();

        if (totalSeconds >= goalSeconds)

        {

            dataFile.println("✅ Επτυχία! Πέτυχες ή ξεπέρασες τον στόχο!");

        } else if (abs(deltaSeconds)<=deviationHours*3600)

        {

            dataFile.println("⚠ Είσαι πολύ κοντά στον στόχο! Μπορείς λίγο ακόμα!");

        }

    }

}
```



```
        } else
        {
            dataFile.println(" ❌ Προσπάθησε περισσότερο αύριο!");
        }

        dataFile.println();
        dataFile.close();

    }
}

totalPressureTime= 0;
reportedToday= true;
}

// Reset flag once we've passed the cutoff minute
if (now.hour()!=cutoffHour ||now.minute()!=cutoffMinute)
{
    reportedToday= false;
}

delay(1000);
}
```

Link για GitHub: <https://github.com/DemyPrapa/ArduinoIDE-Code/tree/main>

B. Παράρτημα κώδικα σε MATLAB

```
% Ορισμός σειριακής σύνδεσης

port = "COM6";

baud = 9600;

s = serialport(port, baud);


N = 200;

forces = zeros(N,3);


% Προετοιμασία figure

figure;

tiledlayout(3,1);


ax1 = nexttile;

h1 = plot(ax1, nan, nan, 'r'); grid on; title('FSR 1 (N)');

ylabel('Force [N]');


ax2 = nexttile;

h2 = plot(ax2, nan, nan, 'b'); grid on; title('FSR 2 (N)');

ylabel('Force [N]');


ax3 = nexttile;

h3 = plot(ax3, nan, nan, 'g'); grid on; title('FSR 3 (N)');

ylabel('Force [N]'); xlabel('Samples');
```

```
disp("Ξεκινά η λήψη δεδομένων...");

i = 0;

while i < N

    line = readline(s);    % Διάβασε μια γραμμή

    str = strtrim(line);    % Καθάρισε κενά

    % Έλεγξε αν η γραμμή περιέχει Force

    if contains(str,"Force1(N):")

        num = sscanf(str,"Force1(N): %f");

        i = i + 1;

        forces(i,1) = num;

    elseif contains(str,"Force2(N):")

        num = sscanf(str,"Force2(N): %f");

        forces(i,2) = num;

    elseif contains(str,"Force3(N):")

        num = sscanf(str,"Force3(N): %f");

        forces(i,3) = num;

    set(h1,'XData',1:i,'YData',forces(1:i,1));

    set(h2,'XData',1:i,'YData',forces(1:i,2));

    set(h3,'XData',1:i,'YData',forces(1:i,3));

    drawnow;
```

end

end

10 Αναφορές

1. Arduino IDE. [Ηλεκτρονικό] <https://www.arduino.cc/en/software/>.
2. MIT App Inventor. (n.d.). [Ηλεκτρονικό] <https://appinventor.mit.edu/>.
3. Βουτσινά, Σταύρο-Αθανάσιο Σ. *Διατριβή: Η φυσιολογική ανάπτυξη της θωρακικής και οσφυϊκής μοίρας της σπονδυλικής στήλης - Κωδικός: 1311. (n.d.).* . [Ηλεκτρονικό] 1984. <https://thesis.ekt.gr/thesisBookReader/id/1311?lang=el#page/14/mode/2up>.
4. Bhotla, H. K., Balasubramanian, B., Rengasamy, K. R. R., Arumugam, V. A., Alagamuthu, K. K., Chithravel, V., Chaudhary, A., Alanazi, A. M., Pappuswamy, M., & Meyyazhagan, A. Genotoxic repercussion of high-intensity radiation (x-rays) on hospital radiographers. *Environmental and Molecular Mutagenesis*. [Ηλεκτρονικό] 2022. <https://doi.org/10.1002/em.22523>.
5. Ζαχαράτος, Σπυρίδων. *Συγκριτική πολυπαραγοντική μελέτη των ψυχολογικών συνεπειών της ιδιοπαθούς εφηβικής σκολίωσης και κύφωσης Scheuermann σε εφήβους που αντιμετωπίστηκαν με εφαρμογή κορμικού κηδεμόνα.* [Ηλεκτρονικό] 2009. <https://thesis.ekt.gr/thesisBookReader/id/26492?lang=el#page/1/mode/2up>.
6. Admin. Scoliosi. Fisiorehab360 | Fisioterapia E Fisioterapia in Acqua a Carrara. [Ηλεκτρονικό] 20 October 2022. <https://fisiorehab360.it/scoliosi/>.
7. Konieczny, M. R., Senyurt, H., & Krauspe, R. Epidemiology of adolescent idiopathic scoliosis. *Journal of Children S Orthopaedics*. [Ηλεκτρονικό] 2012. <https://doi.org/10.1007/s11832-012-0457-4>.
8. Kuznia, A. L., Hernandez, A. K., & Lee, L. U. Adolescent idiopathic scoliosis: common questions and answers. *AAFP*. . [Ηλεκτρονικό] 1 January 2020. <https://www.aafp.org/pubs/afp/issues/2020/0101/p19.html>.
9. (n.d.), Chania Physiotherapy Clinic. *Σκολίωση: Απαντώντας στις πιο συχνές ερωτήσεις με τον πιο απλό τρόπο* . [Ηλεκτρονικό] <https://chpc.gr/%CF%83%CE%BA%CE%BF%CE%BB%CE%AF%CF%89%CF%83%CE%B7-%CE%B1%CF%80%CE%B1%CE%BD%CF%84%CF%8E%CE%BD%CF%84%CE%B1%CF%82-%CF%83%CF%84%CE%B9%CF%82-%CF%80%CE%B9%CE%BF-%CF%83%CF%85%CF%87%CE%BD%CE%AD%CF%82-%CE%B5/>.

10. Seleviciene, V., Cesnaviciute, A., Strukcinskiene, B., Marcinowicz, L., Strazdiene, N., & Genowska, A. Physiotherapeutic Scoliosis-Specific Exercise Methodologies used for conservative treatment of adolescent idiopathic scoliosis, and their effect. [Ηλεκτρονικό] 2022. <https://www.mdpi.com/1660-4601/19/15/9240>.
11. Vesalius. Κηδεμόνας σκολίωσης – Πόσο αποτελεσματικός είναι τελικά. [Ηλεκτρονικό] 16 October 2023. <https://www.vesalius.gr/blog/kidemonas-skoliosis>.
12. Meng, Z., Li, T., Xie, X., Luo, C., Lian, X., & Wang, Z. *Quality of life in adolescent patients with idiopathic scoliosis after brace treatment. Medicine.* [Ηλεκτρονικό] 2017. <https://doi.org/10.1097/md.0000000000006828>.
13. Scoliosis SLC, Spinal Lazer Center. [Ηλεκτρονικό] https://scoliosisslc.com/profil/?_gl=1*1dbxtvy*_up*MQ..*_gs*MQ..&gclid=Cj0KCQjwjo7DBhCrARIsACWauSkeM-fzAw3-DgtKX0daRQjg1Jf9OLS6p-mrcCZqEpJ03sbXBph__KgaAjIREALw_wcB&gbraid=0AAAAABmW5jv_QtvIOAewQjWMn_qMGRu0p.
14. Έλεγχος FORMETRIC 4D - Scoliosis SLC. [Ηλεκτρονικό] https://scoliosisslc.com/elegchos-formetric-4d/?gad_source=1&gad_campaignid=12654308150&gbraid=0AAAAABmW5jsyNOD3NqBS3YUq1eyfv44-L&gclid=CjwKCAjwsZPDBhBWEiwADuO6y1IFRGK_u1h0gROZ3IX1-u_fq0OKrlcN10pxJaw291QrbGWcHAI1RhOCcvMQAvD_BwE.
15. Χριστοδούλου, Ορθοπεδικός Χειρουργός Θεσσαλονίκη | Δρ. Παύλος. *Σκολίωση παιδιών - εφήβων | Αίτια & αντιμετώπιση. (n.d.).* [Ηλεκτρονικό] <https://spine-surgery.gr/pathiseis/skoliwsh-paidiwn-efivwn/#:~:text=%CE%95%CF%80%CE%B9%CF%80%CE%BB%CE%AD%CE%BF%CE%BD%2C%20%CE%B7%20%CE%B5%CE%BE%CE%AD%CE%BB%CE%B9%CE%BE%CE%B7%20%CF%84%CE%B7%CF%82%20%CF%83%CE%BA%CE%BF%CE%BB%CE%AF%CF%89%CF%83%CE%B7%CF%82%20%>.
16. Weinstein, S. L., Dolan, L. A., Wright, J. G., & Dobbs, M. B. *Effects of Bracing in Adolescents with Idiopathic Scoliosis. New England Journal of Medicine.* [Ηλεκτρονικό] 2013. <https://doi.org/10.1056/nejmoa1307337>.
17. Φ., Πέλλιος Σταύρος. *Αποτελεσματικότητα και επίδραση στην ποιότητα ζωής του τροποποιημένου κηδεμόνα τύπου Boston στην ιδιοπαθή εφηβική σκολίωση.* [Ηλεκτρονικό] 2016. <https://ikee.lib.auth.gr/record/287716>.

18. Zhu, C., Wu, Q., Xiao, B., Wang, J., Luo, C., Yu, Q., Liu, L., & Song, Y. *Zhu, C., Wu, Q., Xiao, B., WaA compliance real-time monitoring system for the management of the brace usage in adolescent idiopathic scoliosis patients: a pilot study. BMC Musculoskeletal Disorders.* [Ηλεκτρονικό] 2021. <https://doi.org/10.1186/s12891-021-03976-5>.
19. (n.d.), Aidplex. *ScolioSense • Product. Aidplex.* . [Ηλεκτρονικό] <https://www.scoliosense.com/Home/productEN>.
20. Kim, M., Ryu, S., Jang, S. J., Ko, J. H., Choi, H. W., & Lim, S. *Smart soft scoliosis brace: Integrated textile sensors for real-time pressure monitoring. Textile Research Journal.* . [Ηλεκτρονικό] 2025. <https://doi.org/10.1177/00405175251325889>.
21. Fuss, F. K., Ahmad, A., Tan, A. M., Razman, R., & Weizman, Y. *Pressure sensor system for customized scoliosis braces. Sensors, 21(4), 1153.* . [Ηλεκτρονικό] 2021. <https://doi.org/10.3390/s21041153>.
22. Franz Konstantin Fuss, Asliza Ahmad, Adin Ming Tan, Rizal Razman and Yehuda Weizman . *Pressure Sensor System for Customized Scoliosis Braces.* [Ηλεκτρονικό] <https://www.mdpi.com/1424-8220/21/4/1153>.
23. Fuss, F. K., Ahmad, A., Tan, A. M., Razman, R., & Weizman, Y. *Pressure sensor system for customized scoliosis braces. Sensors.* [Ηλεκτρονικό] 2021. <https://doi.org/10.3390/s21041153>.
24. Handbook of Energy and Environmental Security. *Human Centred Design.* [Ηλεκτρονικό] 2022. [https://www.sciencedirect.com/topics/engineering/human-centred-design#:~:text=Human%2Dcentered%20design%20is%20a,experiences%20\(Norman%2C%201988\)..](https://www.sciencedirect.com/topics/engineering/human-centred-design#:~:text=Human%2Dcentered%20design%20is%20a,experiences%20(Norman%2C%201988)..)
25. Interaction Design Foundation . *Human-Centered Design (HCD).* [Ηλεκτρονικό] [https://www.interaction-design.org/literature/topics/human-centered-design#:~:text=Human%2Dcentered%20design%20\(HCD\)%20is%20a%20methodology%20that%20places,their%20unique%20challenges%20and%20desires..](https://www.interaction-design.org/literature/topics/human-centered-design#:~:text=Human%2Dcentered%20design%20(HCD)%20is%20a%20methodology%20that%20places,their%20unique%20challenges%20and%20desires..)
26. ISO. *Ergonomics of human-system interaction — Part 210: Human-centred design for interactive systems.* [Ηλεκτρονικό] <https://www.iso.org/obp/ui/en/#iso:std:77520:en>.
27. Kosinski, Gregg Lindemulder & Matthew. *What is a data flow diagram (DFD)?* [Ηλεκτρονικό] <https://www.ibm.com/think/topics/data-flow-diagram>.

28. Lucidchart. *What is a Data Flow Diagram.* [Ηλεκτρονικό] <https://www.lucidchart.com/pages/data-flow-diagram#:~:text=the%20route%20that%20data%20takes%20between%20the,a%20short%20data%20name%2C%20like%20%E2%80%9CBilling%20details.%E2%80%9D>.
29. Canva . [Ηλεκτρονικό] <https://www.canva.com/graphs/data-flow-diagrams/>.
30. Canva Desing. [Ηλεκτρονικό] <https://www.canva.com/design/DAGtgs4WXMA/Qgh5lc5TPsyZDOQmNNCtPg/edit?referrer=data-flow-diagrams-landing-page>.
31. Zhang, R., & Lin, W. *A software development platform for wearable medical applications. Journal of Medical Systems*, 39(10). [Ηλεκτρονικό] 2015. <https://doi.org/10.1007/s10916-015-0309-0>.
32. TeenSy® 4.0. (n.d.). . [Ηλεκτρονικό] <https://www.pjrc.com/store/teensy40.html>.
33. MAHA S. DIAB, (Graduate Student Member, IEEE), and ESTHER RODRIGUEZ-VILLEGAS. *Embedded Machine Learning Using Microcontrollers in Wearable and Ambulatory Systems for Health and Care.* [Ηλεκτρονικό] 2022. https://www.researchgate.net/publication/363605825_Embedded_Machine_Learning_Using_Microcontrollers_in_Wearable_and_Ambulatory_Systems_for_Health_and_Care_Applications_A_Review.
34. (FSR), Force Sensitive Resistor. *Adafruit Learning System.* [Ηλεκτρονικό] 29 July 2012. <https://learn.adafruit.com/force-sensitive-resistor-fsr/using-an-fsr>.
35. Medeiros, D. et al. *"Force sensitive resistor in wearable pressure monitoring: Performance evaluation". Sensors.* [Ηλεκτρονικό]
36. Digital, Hellas. *MicroSD card module for Arduino (DFR0229).* (n.d.). . [Ηλεκτρονικό] <https://www.hellasdigital.gr/go-create/arduino-shields-and-accessories/microsd-card-module-for-arduino-dfr0229/>.
37. Digital., Hellas. *Home :: Electronics :: Sensors :: RTC :: DS3231 Real Time Clock Module for Arduino.* (n.d.). . [Ηλεκτρονικό] <https://www.hellasdigital.gr/electronics/sensors/rtc/ds3231-at24c32-iic-precision-real-time-clock-module-memory-module-arduino/>.
38. Mouser. *DS1307 Serial Real-Time Clock.* (n.d.). *Analog Devices / Maxim Integrated .* [Ηλεκτρονικό]

39. [Ηλεκτρονικό] <https://www.batteries.gr/gr/mpataria-varta-cr2032-lithiou-3v.html>.
40. [Ηλεκτρονικό] <https://www.emichos.gr/kitronik-2209-37v-450mah-mpataria-lipo-drone>.
41. [Ηλεκτρονικό] https://grobotronics.com/dc-dc-boost-module-0-9-5v-to-5v.html?gad_source=1&gad_campaignid=20999016384&gbraid=0AAAAADpl6QoQP25GZvMj8wzbMzw-zmFmj&gclid=CjwKCAjw7MLDBhAuEiwAleXGIRAGD7V5WXhk6tBGMBjKz51j23ehczVOujFZa7_1SRrtggIkYBJoORoCA-0QAvD_BwE.
42. Electromaker. *DC-DC Boost module*. . [Ηλεκτρονικό] <https://www.electromaker.io/shop/product/dc-dc-boost-module?srsltid=AfmBOopF-fBqpE0zfG4iXq907cbZL6D5yeEPsb0g3urDDM-Usg816FJ9>.
43. [Ηλεκτρονικό] <https://www.devobox.com/el/charging-modules/1065-lithium-battery-charging-module-tp4056-dual-type-c.html>.
44. Barman, S. (n.d.). . *TP4056 Lithium Cell Charger Module Circuit Working Explanation*. *Electrothinks*. . [Ηλεκτρονικό] <https://www.electrothinks.com/2024/02/tp4056-lithium-cell-charger-module.html>.
45. [Ηλεκτρονικό] <https://www.devobox.com/el/bluetooth/738-hc-05-serial-bluetooth-module-for-arduino-gr.html>.
46. Zahidali. *HC-05 Bluetooth Module Pinout, Datasheet, features & Applications*. *The Engineering Projects*. [Ηλεκτρονικό] 31 December 2023. <https://www.theengineeringprojects.com/2019/10/hc-05-bluetooth-module-pinout-datasheet-features-applications.html>.
47. (n.d.), iteadstudio.com. *HC-05 Bluetooth module*. . [Ηλεκτρονικό] https://components101.com/sites/default/files/component_datasheet/HC-05%20Datasheet.pdf.
48. SHENZHEN SHSD Technology Co., Ltd. (n.d.). *HC-05 Data Sheet*. . [Ηλεκτρονικό] <https://cdn.awsli.com.br/945/945993/arquivos/HC-05-BOARD.pdf>.
49. (n.d.), electronica60norte. *BLUETOOTH TO SERIAL PORT MODULE HC05*. . [Ηλεκτρονικό] <https://www.faranux.com/wp-content/uploads/2016/12/Datasheet.pdf>.
50. *HC-05 - Bluetooth Module*. (n.d.). *Components101*. [Ηλεκτρονικό] <https://components101.com/wireless/hc-05-bluetooth-module>.

51. Martyn. *Arduino with HC-05 Bluetooth module in slave mode*. [Ηλεκτρονικό] 14 June 2024. <https://www.martyncurrey.com/arduino-with-hc-05-bluetooth-module-in-slave-mode/>.
52. Electronic, Highleap. *Διαιρέτης τάσης: Οδηγός ρύθμισης τάσης σε ηλεκτρονικά κυκλώματα*. [Ηλεκτρονικό] <https://hilelectronic.com/el/voltage-divider/>.
53. [Ηλεκτρονικό] <https://docs.arduino.cc/learn/built-in-libraries/software-serial/>.
54. Arduino Serial: A look at the different serial libraries. [Ηλεκτρονικό] <https://www.martyncurrey.com/arduino-serial-a-look-at-the-different-serial-libraries/>.
55. Makerguides. *Force Sensing Resistor (FSR) with Arduino Tutorial*. [Ηλεκτρονικό] <https://www.makerguides.com/fsr-arduino-tutorial/>.
56. Electronics, Interlink. *FSR Integration Guide & Evaluation Parts Catalog*. [Ηλεκτρονικό] 2000. <https://www.sparkfun.com/datasheets/Sensors/Pressure/fsrguide.pdf>.
57. Film sensor Technology Co., Ltd. (n.d.). *Force Sensitive Resistor Product Description*. [Ηλεκτρονικό] https://www.laskakit.cz/user/related_files/data-sheet-of-rp-c10-lt-rev-a_1_-1.pdf.
58. Instructables. *FSR Tutorial!*. [Ηλεκτρονικό] 7 November 2017. <https://www.instructables.com/FSR-Tutorial/>.
59. (n.d.), Unknown. *Force sensing Resistor Theory and Applications. In FSR 101 (Revision 1.01)*. [Ηλεκτρονικό] https://www.sensitronics.com/pdf/Sensitronics_FSR_101.pdf.
60. Basics, Voltage Divider. *How It Works and How to Use It with FSR Sensors*. (n.d.). [Ηλεκτρονικό] <https://www.ceradex-sensor.com/articles/voltage-divider-basics-how-it-works-and-how-to-use-it-with-fsr-sensors>.
61. Learn Electronics. *Το πρωτόκολλο επικοινωνίας SPI*. [Ηλεκτρονικό] <https://learnelectronics.gr/%CF%84%CE%BF-%CF%80%CF%81%CF%89%CF%84%CF%8C%CE%BA%CE%BF%CE%BB%CE%BB%CE%BF-%CE%B5%CF%80%CE%B9%CE%BA%CE%BF%CE%B9%CE%BD%CF%89%CE%BD%CE%AF%CE%B1%CF%82-spi/#:~:text=%CE%A4%CE%BF%20%CF%80%CF%81%CF%89%CF%84%CF%8C%CE%BA%CE%BF%CE%BB%CE%>.

62. *How to link teensy4.0 with micro sd card [Online forum post]. Teensy Forum.* . [Ηλεκτρονικό] 10 September 2023. <https://forum.pjrc.com/index.php?threads/how-to-link-teensy4-0-with-micro-sd-card.73483/>.
63. (n.d.), M.Raw. *Teensy tutorial: SD Card.* M.B Raw. [Ηλεκτρονικό] <https://mb-raw.blogspot.com/2017/07/teensy-tutorial-sd-card.html>.
64. Semiconductors, NXP. *SD(HC)-memory card and MMC interface conditioning.* [Ηλεκτρονικό] 2010. <https://datasheet.octopart.com/IP4052CX20/LF,135-NXP-Semiconductors-datasheet-12555750.pdf>.
65. stackexchange. *SD card socket just need to wire with pull-up resistors?* [Ηλεκτρονικό] https://electronics.stackexchange.com/questions/598650/sd-card-socket-just-need-to-wire-with-pull-up-resistors?utm_source=chatgpt.com.
66. *Teensy 4.0 to nano communication* . [Ηλεκτρονικό] <https://forum.arduino.cc/t/teensy-4-0-to-nano-communication/1200579>.
67. stackoverflow. *SD.h not compatible with other libraries in Arduino/C++ environment.* [Ηλεκτρονικό] <https://stackoverflow.com/questions/13554570/sd-h-not-compatible-with-other-libraries-in-arduino-c-environment>.
68. Anne. [Ηλεκτρονικό] 18 February 2024. <https://www.diskgenius.com/resource/how-to-format-sd-card-to-fat32.html#:~:text=When%20your%20SD%20card%20is,to%20FAT32%20file%20system%20easily..>
69. *SdFat.* [Ηλεκτρονικό] <https://docs.arduino.cc/libraries/sdfat/>.
70. (n.d.), Greiman. *GitHub - greiman/SdFat: Arduino FAT16/FAT32 exFAT Library.* . [Ηλεκτρονικό] <https://github.com/greiman/SdFat>.
71. ProtoSupplies.com. *DS3231M RTC with EEPROM Module - ProtoSupplies.* [Ηλεκτρονικό] 18 June 2025. <https://protosupplies.com/product/ds3231-rtc-with-EEPROM-module/>.
72. Staff., LME Editorial. *Interface DS3231 Precision RTC Module with Arduino. Last Minute Engineers.* [Ηλεκτρονικό] 22 April 2025. <https://lastminuteengineers.com/ds3231-rtc-arduino-tutorial/>.

73. All About Circuits. *Understanding the Operation of a Boost Converter*. [Ηλεκτρονικό] <https://www.allaboutcircuits.com/technical-articles/understanding-the-operation-of-a-boost-converter/>.

74. stackexchange. *tp4056 charger circuit with protection:can i short bat- and out?* [Ηλεκτρονικό] <https://electronics.stackexchange.com/questions/683161/tp4056-charger-circuit-with-protectioncan-i-short-bat-and-out>.

75. Arduino. [Ηλεκτρονικό] <https://forum.arduino.cc/t/li-ion-3-7v-battery-tp4056-charger-sx1308-dc-step-up-regulator-low-voltag/588488>.

76. *Setup & loop in Arduino IDE*. [Ηλεκτρονικό] <https://docs.arduino.cc/language-reference/en/structure/sketch/loop/>.

77. Arduino, docs. [Ηλεκτρονικό] <https://docs.arduino.cc/software/ide-v1/tutorials/Environment/>.

78. GeeksforGeeks. *Difference between #define and const in C*. [Ηλεκτρονικό] 23 July 2025. <https://www.geeksforgeeks.org/c/difference-define-const-c/>.

79. Sensorworlds. *Πίεση: Ορισμός, Μονάδες Μέτρησης και Μετατροπή*. [Ηλεκτρονικό] 12 January 2024. <https://gr.sensorworlds.com/info/pressure-definition-measurement-units-and-c-94827937.html>.

80. scantips. *Dynamic range, 12 bits vs 16 bits. (n.d.)*. [Ηλεκτρονικό] <https://www.scantips.com/basics14.html>.

81. Arduino. *analogReadResolution()*. [Ηλεκτρονικό] <https://docs.arduino.cc/language-reference/en/functions/analog-io/analogReadResolution/>.

82. Cheich, M. . *What is Serial.begin(9600)? Programming Electronics Academy*. [Ηλεκτρονικό] 13 November 2023. <https://www.programmingelectronics.com/serial-begin-9600/>.

83. Reference. (n.d.). Particle. . *PinMode() - Input/Output* . [Ηλεκτρονικό] [https://docs.particle.io/reference/device-os/api/input-output/pinmode/#:~:text=pinMode\(\)%20configures%20the%20specified,resistor\)%2C%20or%20an%20output..](https://docs.particle.io/reference/device-os/api/input-output/pinmode/#:~:text=pinMode()%20configures%20the%20specified,resistor)%2C%20or%20an%20output..)

84. Pololu. *A-Star 32U4 Prime pinout and components. (n.d.)*. [Ηλεκτρονικό] [https://www.pololu.com/docs/0J61/5.2#:~:text=A%20fourth%20pin%2C%20CS%20\(c](https://www.pololu.com/docs/0J61/5.2#:~:text=A%20fourth%20pin%2C%20CS%20(c)

hip%20select)%2C%20must, pin%20low%20to%20select%20the%20microSD%20card
..

85. Particle. *millis()* - *Time* . [Ηλεκτρονικό] <https://docs.particle.io/reference/device-os/api/time/millis/>.

86. ArduinoDocs. *begin()*. [Ηλεκτρονικό] <https://docs.arduino.cc/language-reference/en/functions/communication/wire/begin/>.

87. Patel, H. *DS3231 RTC (Real Time Clock) Interfacing with Arduino to build DIY Digital Clock*. *CircuitDigest*. . [Ηλεκτρονικό] 11 May 2022. <https://circuitdigest.com/microcontroller-projects/interfacing-ds3231-rtc-with-arduino-and-diy-digital-clock>.

88. ArduinoDocs. *analogRead()*. [Ηλεκτρονικό] <https://docs.arduino.cc/language-reference/en/functions/analog-io/analogRead/>.

89. Academy, B. (n.d.). Binance Academy. *Simple moving Average (SMA)*. [Ηλεκτρονικό] <https://academy.binance.com/en/glossary/simple-moving-average-sma>.

90. Εθνικών και Καποδιστριακών Πανεπιστημίων Αθηνών. *Signal-Smoothing algorithms*. (n.d.). [Ηλεκτρονικό] http://webapps.chem.uoa.gr/efs/applets/AppletSmooth/AppL_Smooth1.html.

91. ArduinoDocs. *Smoothing Readings From an Analog Input*. [Ηλεκτρονικό] <https://docs.arduino.cc/built-in-examples/analog/Smoothing/>.

92. SparkFun Learn. (n.d.). *Bluetooth Basics* . [Ηλεκτρονικό] <https://learn.sparkfun.com/tutorials/bluetooth-basics/all>.

93. LME Editorial Staff. . *Interfacing HC05 Bluetooth Module with Arduino*. *Last Minute Engineers*. [Ηλεκτρονικό] 10 September 2023. <https://lastminuteengineers.com/hc05-bluetooth-arduino-tutorial/>.

94. ArduinoDocs. *Serial.readStringUntil()*. [Ηλεκτρονικό] <https://docs.arduino.cc/language-reference/en/functions/communication/serial/readStringUntil/>.

95. —. *trim()*. [Ηλεκτρονικό] <https://docs.arduino.cc/language-reference/en/variables/data-types/stringObject/Functions/trim/>.

96. —. *substring()*. [Ηλεκτρονικό] <https://docs.arduino.cc/language-reference/en/variables/data-types/stringObject/Functions/substring/>.

97. Labex. *How to create datetime objects from ISO-8601 date strings*. [Ηλεκτρονικό] <https://labex.io/tutorials/python-how-to-create-datetime-objects-from-iso-8601-date-strings-417942>.
98. Ardumotive Arduino Greek Playground. *Ωρα και ημερομηνία με το DS1307 RTC και Arduino*. (n.d.). [Ηλεκτρονικό] https://www.ardumotive.com/ds1307_rtc-gr.html.
99. W3Schools.com. (n.d.). *JavaScript String startsWith()*. [Ηλεκτρονικό] https://www.w3schools.com/jsref/jsref_startswith.asp.
100. ArduinoDocs. *startsWith()*. [Ηλεκτρονικό] <https://docs.arduino.cc/language-reference/en/variables/data-types/stringObject/Functions/startsWith/>.
101. W3Schools.com. (n.d.). *JavaScript String indexOf()*. [Ηλεκτρονικό] https://www.w3schools.com/jsref/jsref_indexof.asp.
102. ArduinoDocs. *millis()*. [Ηλεκτρονικό] <https://docs.arduino.cc/language-reference/en/functions/time/millis/>.
103. Texas Instruments, TI. *Voltage reference selection basics*. [Ηλεκτρονικό] 2018. https://www.ti.com/lit/wp/slpy003a/slpy003a.pdf?ts=1754202965853&ref_url=https%253A%252F%252Fwww.google.com%252F.
104. Grace. *Understanding ADC basic parameters and strategies to improve ADC performance - Andwin Circuits*. Andwin Circuits. [Ηλεκτρονικό] 28 May 2025. <https://www.andwinpcb.com/understanding-adc-basic-parameters-and-strategies-to-improve-adc-performance/>.
105. Texas Instruments. *Migration from 3.3-V to 2.5-V power supplies for logic devices*. [Ηλεκτρονικό] 1997. <https://www.ti.com/lit/an/scea005/scea005.pdf?ts=1754161729541>.