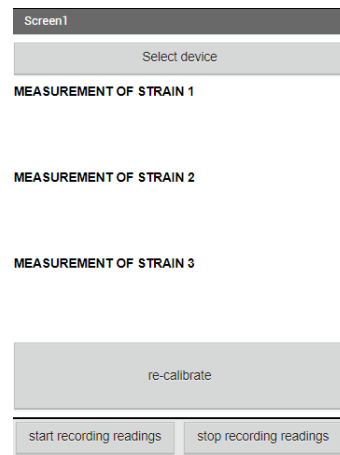
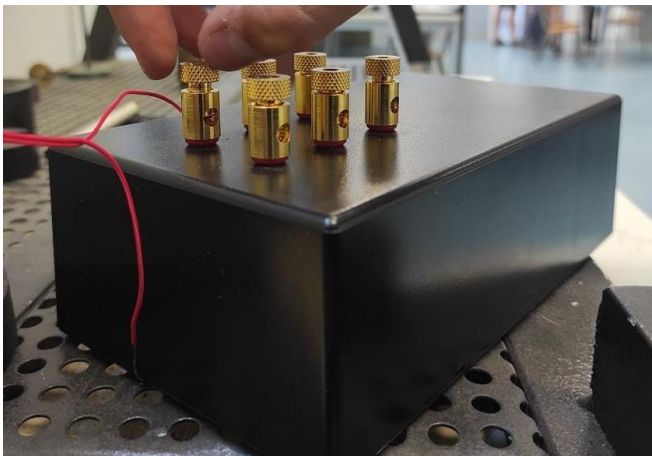


ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΛΟΠΟΝΝΗΣΟΥ
ΣΧΟΛΗ ΜΗΧΑΝΙΚΩΝ
ΤΜΗΜΑ ΜΗΧΑΝΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ

ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ

ΕΞΕΛΙΞΗ ΣΥΣΤΗΜΑΤΟΣ ΜΕΤΡΗΣΗΣ ΠΑΡΑΜΟΡΦΩΣΕΩΝ ΜΕ ΧΡΗΣΗ ΗΛΕΚΤΡΟΕΠΙΜΗΚΥΝΣΙΟΜΕΤΡΩΝ (STRAIN GAGES) ΚΑΙ ARDUINO



ΦΟΙΤΗΤΗΣ: ΜΠΟΥΖΟΥΛΑΣ ΔΗΜΗΤΡΙΟΣ (Α.Μ. 7725)

ΕΠΙΒΛΕΠΩΝ ΚΑΘΗΓΗΤΗΣ: ΙΩΑΝΝΗΣ ΔΙΑΜΑΝΤΑΚΟΣ

ΠΑΤΡΑ 2022

ΠΡΟΛΟΓΟΣ

Η παρούσα πτυχιακή εργασία εκπονήθηκε στο Τμήμα Μηχανολόγων Μηχανικών του Πανεπιστημίου Πελοποννήσου και αφορά στη σχεδίαση, κατασκευή και παρουσίαση συσκευής μετρήσεως παραμόρφωσης με χρήση μικροελεγκτή Arduino και ηλεκτρο-επιμηκυνσιομέτρων strain gages. Αρχικά μελετώνται εισαγωγικά αντικείμενα της εργασίας, όπως η έννοια της παραμόρφωσης, η γέφυρα Wheatstone, το ηλεκτρεπιμηκυνσιόμετρο και ο μικροελεγκτής Arduino. Στη συνέχεια, επεξηγείται η ανάλυση, υλοποίηση και λειτουργία του συστήματος. Τέλος λαμβάνονται μετρήσεις, οι οποίες συγκρίνονται με αντίστοιχες από προηγούμενο σύστημα και με θεωρητικές προβλέψεις με σκοπό την εκτίμηση της ακρίβειας του συστήματος που αναπτύχθηκε. Ευχαριστώ θερμά τον Επιβλέποντα κ. Διαμαντάκο Ιωάννη, για την πολύτιμη βοήθεια και καθοδήγηση που μου προσέφερε κατά την πραγματοποίηση της εργασίας.

Υπεύθυνη Δήλωση Φοιτητή: Ο κάτωθι υπογεγραμμένος Φοιτητής έχω επίγνωση των συνεπειών του Νόμου περί λογοκλοπής και δηλώνω υπεύθυνα ότι είμαι συγγραφέας αυτής της Πτυχιακής Εργασίας, έχω δε αναφέρει στην Βιβλιογραφία μου όλες τις πηγές τις οποίες χρησιμοποίησα και έλαβα ιδέες ή δεδομένα. Δηλώνω επίσης ότι, οποιοδήποτε στοιχείο ή κείμενο το οποίο έχω ενσωματώσει στην εργασία μου προερχόμενο από Βιβλία ή άλλες εργασίες ή το διαδίκτυο, γραμμένο ακριβώς ή παραφρασμένο, το έχω πλήρως αναγνωρίσει ως πνευματικό έργο άλλου συγγραφέα και έχω αναφέρει ανελλιπώς το όνομά του και την πηγή προέλευσης.

Ο Φοιτητής

ΜΠΟΥΖΟΥΛΑΣ ΔΗΜΗΤΡΙΟΣ

.....

ΠΕΡΙΛΗΨΗ

Στην παρούσα πτυχιακή εργασία υλοποιείται ένα σύστημα, το οποίο βασίζεται στην αναπτυξιακή πλατφόρμα Arduino και έχει τη δυνατότητα μέτρησης παραμορφώσεων σε υλικά με χρήση ηλεκτροεπιμηκυνσιομέτρων strain gage. Τα ηλεκτροεπιμηκυνσιόμετρα είναι αισθητήρες που χρησιμοποιούνται για τη μέτρηση παραμορφώσεων και στην παρούσα εργασία χρησιμοποιούνται σε συνδυασμό με τον μικροελεγκτή Arduino UNO, τον ενισχυτή - ADC μετατροπέα HX711, Bluetooth breakout board HC-05, microSD card breakout board της Adafruit και κατάλληλη γέφυρα Wheatstone. Η λήψη και ανάγνωση των μετρήσεων και ο έλεγχος της συσκευής γίνονται από το κινητό τηλέφωνο του χρήστη μέσω εφαρμογής Android η οποία αναπτύχθηκε ειδικά για την συσκευή. Η παρουσίαση της εργασίας γίνεται σε τέσσερα κεφάλαια. Στο πρώτο κεφάλαιο παρουσιάζονται εισαγωγικές και γενικές έννοιες για τις μετρήσεις, τη παραμόρφωση, τη μετρολογία, το ηλεκτροεπιμηκυνσιόμετρο, τα ενσωματωμένα συστήματα, τη γέφυρα Wheatstone, τους μικροελεγκτές και το bluetooth. Στο δεύτερο κεφάλαιο περιγράφονται αναλυτικά τα υλικά που χρησιμοποιήθηκαν για την υλοποίηση του παραπάνω συστήματος, η χρήση τους καθώς και ο ρόλος του καθενός στο εν λόγω σύστημα. Στο τρίτο κεφάλαιο αναπτύσσεται η διαδικασία σχεδίασης, κατασκευής και προγραμματισμού του συστήματος μέτρησης παραμορφώσεων. Αναλυτικότερα, περιγράφεται η συνδεσμολογία του συστήματος και δίνεται λεπτομερή περιγραφή της λειτουργίας του. Επίσης, αναλύεται ο κώδικας που εκτελεί ο μικροελεγκτής και η εφαρμογή κινητού τηλεφώνου που αναπτύχθηκε. Στο τέταρτο κεφάλαιο, παρουσιάζονται οι μετρήσεις παραμόρφωσης που πραγματοποιήθηκαν με το παραπάνω σύστημα και η σύγκριση τους με μετρήσεις που πραγματοποιήθηκαν αντίστοιχο όργανο μέτρησης παραμόρφωσης και με υπολογισμούς από το μαθηματικό μοντέλο που επιλύθηκε για τη πειραματική διάταξη. Στο τέλος του τέταρτου κεφαλαίου, παρουσιάζονται τα αποτελέσματα καθώς και αναφέρονται κάποιες προτάσεις για την μελλοντική επέκταση της συσκευής που υλοποιήθηκε.

ΠΕΡΙΕΧΟΜΕΝΑ

ΠΡΟΛΟΓΟΣ	2
ΠΕΡΙΛΗΨΗ	3
ΠΕΡΙΕΧΟΜΕΝΑ	4
ΚΕΦΑΛΑΙΟ 1 - ΕΙΣΑΓΩΓΗ	5
1.1 ΣΤΟΧΟΣ ΕΡΓΑΣΙΑΣ	5
1.2 ΕΙΣΑΓΩΓΗ ΣΤΙΣ ΜΕΤΡΗΣΕΙΣ ΚΑΙ ΤΗ ΜΕΤΡΟΛΟΓΙΑ	6
1.3 ΑΝΤΟΧΗ ΤΩΝ ΥΛΙΚΩΝ ΚΑΙ ΠΑΡΑΜΟΡΦΩΣΗ	9
1.4 ΑΙΣΘΗΤΗΡΕΣ ΜΕΤΡΗΣΕΩΝ	14
1.5 ΕΝΣΩΜΑΤΩΜΕΝΑ ΣΥΣΤΗΜΑΤΑ	16
1.6 ΜΙΚΡΟΕΛΕΓΚΤΕΣ	23
1.7 ΓΕΦΥΡΑ WHEATSTONE	25
1.8 BLUETOOTH	28
ΚΕΦΑΛΑΙΟ 2 - HARDWARE	32
2.1 ARDUINO MICROCONTROLLER	32
2.2 HX711	34
2.3 BLUETOOTH MODULE HC-05	36
2.4 SDCARD MODULE	38
ΚΕΦΑΛΑΙΟ 3 – ΣΥΝΔΕΣΜΟΛΟΓΙΑ ΚΑΙ ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΣ	43
3.1 ΣΥΝΔΕΣΜΟΛΟΓΙΑ	43
3.2 ΜΑΘΗΜΑΤΙΚΟ ΜΟΝΤΕΛΟ ΣΥΣΤΗΜΑΤΟΣ	46
3.3 ΑΝΑΛΥΣΗ ΤΟΥ ΚΩΔΙΚΑ ΠΟΥ ΧΡΗΣΙΜΟΠΟΙΗΘΗΚΕ	51
3.4 ΑΝΑΛΥΣΗ ΤΟΥ ΚΩΔΙΚΑ ΤΗΣ ΕΦΑΡΜΟΓΗΣ ΤΟΥ ΤΗΛΕΦΩΝΟΥ	62
ΚΕΦΑΛΑΙΟ 4 – ΠΕΙΡΑΜΑΤΙΚΗ ΕΦΑΡΜΟΓΗ	65
4.1 ΠΡΑΚΤΙΚΗ ΕΦΑΡΜΟΓΗ ΤΟΥ ΜΕΤΡΗΤΙΚΟΥ ΟΡΓΑΝΟΥ	65
4.2 ΣΥΓΚΡΙΣΗ ΜΕΤΡΗΣΕΩΝ ΜΕ ΑΝΤΙΣΤΟΙΧΟ ΟΡΓΑΝΟ ΚΑΙ ΘΕΩΡΗΤΙΚΟ ΜΟΝΤΕΛΟ	67
4.3 ΣΥΜΠΕΡΑΣΜΑΤΑ ΚΑΙ ΠΙΘΑΝΕΣ ΒΕΛΤΙΩΣΕΙΣ	75
ΠΑΡΑΡΤΗΜΑ	78
ΟΔΗΓΙΕΣ ΧΡΗΣΗΣ	78
ΣΗΜΑΝΤΙΚΕΣ ΠΑΡΑΤΗΡΗΣΕΙΣ	83
Ο ΚΩΔΙΚΑΣ ΠΟΥ ΧΡΗΣΙΜΟΠΟΙΗΘΗΚΕ	84
ΜΠΛΟΚ ΕΝΤΟΛΩΝ ΕΦΑΡΜΟΓΗΣ	94
ΒΙΒΛΙΟΓΡΑΦΙΑ	95

ΚΕΦΑΛΑΙΟ 1 - ΕΙΣΑΓΩΓΗ

1.1 ΣΤΟΧΟΣ ΕΡΓΑΣΙΑΣ

Στόχος της παρούσας πτυχιακής εργασίας είναι η ανάπτυξη μιας ηλεκτρονικής διάταξης μέτρησης παραμόρφωσης. Η μετρητική διάταξη βασίζεται στη χρήση ηλεκτροεπιμηκυνσιομέτρων (strain gages) και στην πλατφόρμα Arduino, η οποία μέσω του μετατροπέα HX711 και γέφυρας Wheatstone δίνει τη δυνατότητα μέτρησης της παραμόρφωσης. Τα ηλεκτροεπιμηκυνσιόμετρα (strain gages) κολλιούνται στην κατασκευή και αυτό έχει ως αποτέλεσμα η αντίσταση τους να μεταβάλλεται ανάλογα με την παραμόρφωση στα αντίστοιχα σημεία της κατασκευής. Το κάθε ηλεκτροεπιμηκυνσιομέτρο είναι μέρος μιας γέφυρας Wheatstone η οποία μετατρέπει τη μεταβολή της αντίστασης σε διαφορά τάσης. Στη συνέχεια αυτή η διαφορά τάσης ενισχύεται από τον μετατροπέα HX711 και εισάγεται στο μικροελεγκτή Arduino. Ακόλουθα, γίνεται υπολογισμός της παραμόρφωσης της επιφάνειας πάνω στην οποία τοποθετήθηκε το ηλεκτροεπιμηκυνσιομέτρο. Τα αποτελέσματα προβάλλονται στην οθόνη του smart phone του χρήστη, μέσω κατάλληλης εφαρμογής. Ταυτόχρονα ο χρήστης έχει τη δυνατότητα να ξεκινήσει και να σταματήσει τη καταγραφή δεδομένων σε κάρτα SD με σκοπό τη μελλοντική χρήση τους. Το όργανο μέτρησης σχεδιάστηκε με τέτοιο τρόπο ώστε κατά την ενεργοποίησή του να πραγματοποιεί αυτόματη αρχικοποίηση και να κάνει αυτόματη διαγραφή σφάλματος μόνιμης κατάστασης. Ο χρήστης έχει τη δυνατότητα να μηδενίσει τις μετρήσεις μέσω του κινητού του τηλεφώνου, οποιαδήποτε στιγμή.

Τα κυρία πλεονεκτήματα του οργάνου είναι η μέτρηση της παραμόρφωσης σε πραγματικό χρόνο, η δυνατότητα μέτρησης πολλαπλών ηλεκτροεπιμηκυνσιομέτρων ταυτόχρονα και η δυνατότητα αποθήκευσης των μετρήσεων. Κάποιοι παράγοντες που οδήγησαν στη διαμόρφωση αυτής της μετρητικής διάταξης ήταν το μικρό κόστος, μέγεθος και βάρος συγκριτικά με αντίστοιχα όργανα. Τα βασικά μέρη του και τεχνολογίες συγκεκριμένου επιμηκυνσιόμετρου, τα οποία θα αναλυθούν στα επόμενα κεφάλαια είναι ο μικροελεγκτής Arduino Uno, ο μετατροπέας HX711, η γέφυρα Wheatstone, τα ηλεκτροεπιμηκυνσιόμετρα, ο τρόπος αποθήκευσης δεδομένων και η τεχνολογία bluetooth.

1.2 ΕΙΣΑΓΩΓΗ ΣΤΙΣ ΜΕΤΡΗΣΕΙΣ ΚΑΙ ΤΗ ΜΕΤΡΟΛΟΓΙΑ

Ως μέτρηση ορίζουμε τη σύγκριση ενός μεγέθους με ένα όμοιο του το οποίο έχει οριστεί ως μονάδα. Ως αποτέλεσμα για κάθε μέγεθος υπάρχει αντίστοιχη μονάδα μέτρησης.

Μετρολογία είναι η επιστήμη που ασχολείται με τις μετρήσεις, τα όργανα μέτρησης, την αξιοπιστία των οργάνων και των μετρήσεων, τις μονάδες μέτρησης και γενικότερα με ότι αφορά στον ακριβή και αποδεκτό προσδιορισμό των τιμών των μεγεθών που προσδιορίζουν τις ιδιότητες των σωμάτων ή των συστημάτων.

Ιστορικά οι άνθρωποι ξεκίνησαν να χρησιμοποιούν μετρήσεις πολύ πριν τη σύγχρονη εποχή. Οι πρώτες μονάδες που εμφανίστηκαν στην αρχαιότητα είναι οι ανθρωπομετρικές μονάδες. Αυτές οι μονάδες προέκυψαν από το ανθρώπινο σώμα, παραδείγματα αυτών είναι το βήμα και η παλάμη. Παράλληλα αναπτύχθηκαν και μεγάλες μονάδες μέτρησης όπως η περίοδος του ηλίου και της σελήνης. Από αυτές τις αρχαίες μονάδες υπάρχουν πολλές που χρησιμοποιούνται ακόμα και σήμερα σε διάφορα μέρη του πλανήτη. Λόγω της παράλληλης εξέλιξης διάφορων πλουτισμών σε διάφορα μέρη της γης εξελίχθηκαν και πολλά μετρητικά συστήματα. Η παράλληλη αυτή ανάπτυξη οδήγησε σε προβλήματα στις συναλλαγές αλλά λόγω του τότε μικρού πληθυσμού του πλανήτη δεν ήταν αρκετό ώστε να οδηγήσει στην ανάπτυξη ενός παγκόσμιου αποδεκτού μετρητικού συστήματος. Με τη πάροδο του χρόνου οι μετρήσεις και η επιστήμη της μετρολογίας βελτιώθηκαν και έφτασαν στο στάδιο στο οποίο βρίσκονται σήμερα. Κατά τη πρόοδο αυτή υπάρχουν πολλά σημεία ορόσημα της εξέλιξης της μετρολογίας, μερικά από αυτά είναι:

1. Το πιο αρχαίο διατηρημένο πρότυπο μήκους, είναι το πόδι του αγάλματος του Gudea, ο οποίος ήταν κυβερνήτης της Lagash, της Μεσοποταμίας κατά το 3000π.Χ.
2. Στη πολη Mohenjio-Daro (“ο λόφος των νεκρών”) στο Πακιστάν, βρέθηκαν τα πρώτα σταθμά και μετρά ακρίβειας. Η πόλη αυτή δημιουργήθηκε κατά 5000-4000π.Χ. και εγκαταλείφθηκε περί το 1700 π.Χ. και παρόλη την αρχαιότητα της ανέδειξε μεγάλη εξέλιξη και πρόοδο για την εποχή της όπως ρυμοτομία, αποχετευτικό σύστημα, λουτρά (τα οποία ίσως ήταν και θερμαινόμενα), αυτή η εξέλιξη έχει συσχετιστεί με τη πρόοδο της πόλης τις μετρήσεις και τη μετρολογία.

3. Κατασκευάζεται ο Παρθενώνας στην Αθήνα, ένας γενεσιουργός πυρήνας πολλών καινοτομιών, μια κατασκευή μεγίστου μετρολογικού ενδιαφέροντος. Υπό την εποπτεία του γλύπτη Φειδία, οι αρχιτέκτονες Ικτίνος και Καλλικράτης και πολλοί τεχνίτες (όπως Αθηναίοι, Ίωνες και Κυκλαδίτες), κατασκευάζουν το μεγαλύτερο και τελειότερο δωρικό ναό του ελληνικού κόσμου, ο οποίος, και ολοκληρώθηκε και είναι ο μοναδικός κατασκευασμένος εξ ολοκλήρου από μάρμαρο (Πεντέλης). Η βάση του είναι 30.9m x 69.5m. Ο εσωτερικός ναός έχει διαστάσεις 29.8m x 19.2m, με εσωτερικές κολωνοστοιχίες δωρικού ρυθμού σε δύο σειρές, απαραίτητες για τη στήριξη της οροφής. Πρόκειται για το μόνο κτίσμα στο οποίο εφαρμόστηκαν με τον πληρέστερο τρόπο οι τεχνικές λάξευσης του μαρμάρου με σχολαστική ακρίβεια, πράγμα που επιβάρυνε ιδιαίτερα το κόστος κατασκευής. Το συμπαγές βάθρο και θεμέλιο του νέου ναού είχε 80m μήκος και 12m ύψος. Ένα από τα μυστικά της αξεπέραστης τελειότητας-ακριβείας της κατασκευής ήταν η απaráμιλλη επιδεξιότητα των αρχαίων λιθοξόνων καθώς βέβαια και η μεταλλουργία των εργαλείων τους. Τα εργαλεία εκείνης της εποχής ήταν πολύ ανώτερα από τα σημερινά, όπως μαρτυρούν ίχνη που βρέθηκαν στα μαρμαρά. Είναι ευνόητο ότι, μετά από πολύ αυστηρή πειραματική έρευνα, είχαν καταλήξει σε μμεταλλουργικές τεχνικές που ήταν (και είναι) αξεπέραστες. Υπάρχουν μάλιστα και βάσιμοι ισχυρισμοί ότι ακόμη και με την υπάρχουσα σημερινή τεχνολογία, θα ήταν αδύνατο να χτιστεί ο Παρθενώνας στο σημαντικά μικρό χρονικό διάστημα που τον έχτισαν οι αρχαίοι (η κατασκευή άρχισε το 447 π.Χ. και τελείωσε το 438 π.Χ., οι δε καλλιτεχνικές δε εργασίες έφτασαν μέχρι και το 433 π.Χ). Ο Παρθενώνας αποτέλεσε προφανώς και ένα μοναδικό "έργο-εργαστήριο" Μετρολογίας, το οποίο συνδύασε πληθώρα ειδικοτήτων (πολύπλοκοι συνμοναδικό "έργο-εργαστήριο" Μετρολογίας, το οποίο συνδύασε πληθώρα ειδικοτήτων (πολύπλοκοι συνδυασμοί γεωλογικών, γεωμετρικών, μηχανικών αλλά και καλλιτεχνικών παραμέτρων).
4. 1215 μ.Χ. Ο βασιλιάς της Αγγλίας (1199–1216) Ιωάννης ο Ακτήμονα της Magna Carta υπέγραψε το πιο διάσημο έγγραφο στην συνταγματική ιστορία της Βρετανίας και παράλληλα, εισαγωγή ενιαίου συστήματος μετρήσεων.
5. 1960 μ.Χ. Επίσημη υιοθέτηση του Διεθνούς Συστήματος Μονάδων (Systeme International d'Unites - SI), με το 11ο Γενικό Συνέδριο για τα βάρη και τα μέτρα (General Conference on Weights and Measures - CGPM).

Στη σύγχρονη εποχή η επιστήμη της μετρολογίας χωρίζεται σε δυο κύριους κλάδους, τη θεωρητική μετρολογία και την εφαρμοσμένη μετρολογία.

Ο πρώτος κλάδος αναφέρεται στην Θεωρητική ή Θεμελιώδη μετρολογία η οποία έχει ως στόχο την ανάπτυξη θεωρίας, προβλήματα μέτρησης μεγεθών, μονάδων τους, μεθόδους μέτρησης. Η θεωρητική μετρολογία εργάζεται σε κοινά προβλήματα που προκύπτουν όταν γίνονται μετρήσεις σε έναν συγκεκριμένο τομέα της τεχνολογίας, τις ανθρωπιστικές επιστήμες ή ακόμα και στη διασταύρωση πολλών, μερικές φορές των πιο διαφορετικών τομέων γνώσης. Οι μετρολόγοι-θεωρητικοί μπορούν να ασχοληθούν, για παράδειγμα, με τα ζητήματα της μέτρησης γραμμικών διαστάσεων, όγκου και βαρύτητας σε n -διαστατό χώρο, να αναπτύξουν μεθόδους οργανικής εκτίμησης της έντασης ακτινοβολίας των κοσμικών σωμάτων σε σχέση με τις συνθήκες των διαπλανητικών πτήσεων ή να δημιουργήσουν πλήρως νέες τεχνολογίες που καθιστούν δυνατή την αύξηση της έντασης της διαδικασίας, του επιπέδου ακρίβειας και των άλλων παραμέτρων της, βελτιώνουν τα τεχνικά μέσα που εμπλέκονται σε αυτήν κ.λπ. Με τον ένα ή τον άλλο τρόπο, σχεδόν κάθε επιχείρηση σε οποιαδήποτε δραστηριότητα ξεκινά με μια θεωρία και μόνο μετά από μια τέτοια επεξεργασία περνά στη σφαίρα της συγκεκριμένης εφαρμογής.

Ο δεύτερος κλάδος αναφέρεται στην εφαρμοσμένη ή Πρακτική Μετρολογία ασχολείται με το θέμα της μετρολογικής υποστήριξης, πρακτικής χρήσης της θεωρητικής μετρολογικής εξέλιξης, εφαρμογής νομοθετικών μετρολογικών διατάξεων. Καθήκον του είναι να προσαρμόσει τις γενικές διατάξεις και τους θεωρητικούς υπολογισμούς της προηγούμενης ενότητας σε ένα σαφώς καθορισμένο, εξαιρετικά εξειδικευμένο παραγωγικό ή επιστημονικό πρόβλημα. Έτσι, εάν απαιτείται αξιολόγηση της αντοχής του άξονα του κινητήρα, βαθμονόμηση μεγάλου αριθμού κυλίνδρων ρουλεμάν ή παροχή, για παράδειγμα, ολοκληρωμένου μετρολογικού ελέγχου στη διαδικασία εργαστηριακής έρευνας, οι επαγγελματίες θα επιλέξουν την κατάλληλη τεχνολογία από έναν μεγάλο αριθμό ήδη γνωστά, να το επεξεργαστούν και ενδεχομένως να το συμπληρώσουν σε εφαρμογή σε αυτές τις συνθήκες, θα καθορίσουν τον απαραίτητο εξοπλισμό και εργαλεία, τον αριθμό και τα προσόντα του προσωπικού, καθώς και θα αποσυναρμολογήσουν πολλές άλλες τεχνικές πτυχές μιας συγκεκριμένης διαδικασίας.

Η τιμή ενός μεγέθους εκφράζεται συνήθως ως το γινόμενο ενός αριθμού και μίας μονάδας. Η μονάδα μέτρησης είναι απλά μια συγκεκριμένη ποσότητα του

μεγέθους η οποία χρησιμοποιείται ως τιμή αναφοράς και ο αριθμός είναι το πηλίκο της μετρούμενης ποσότητας με την μονάδα μέτρησης.

Σχεδόν όλες οι χώρες στον κόσμο έχουν κανονισμούς σχετικά με την υιοθέτηση και την χρήση των μονάδων μέτρησης. Εξαιτίας της σημασίας και της ανάγκης ύπαρξης μίας ομάδας από καλά καθορισμένες και εύχρηστες μονάδες, παγκοσμίως αποδεκτές, για τις όλες τις πολυδιάστατες και πολύπλοκες εφαρμογές στη σημερινή μας κοινωνία, οι μονάδες μέτρησης επιλέγονται έτσι ώστε να είναι άμεσα διαθέσιμες σε όλους, να είναι σταθερές στο χώρο και στο χρόνο, και να είναι δυνατόν να υλοποιούνται με υψηλή ακρίβεια.

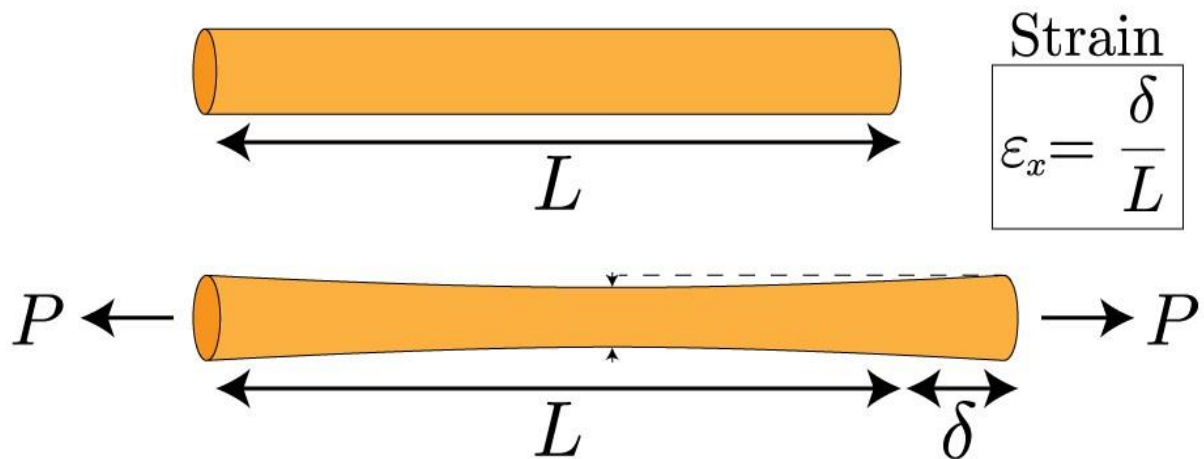
Οι μονάδες χωρίζονται σε δυο σημαντικές κατηγορίες τις βασικές και τις παραγόμενες. Οι βασικές μονάδες είναι μια ομάδα από επτά σαφώς καθορισμένες μονάδες που κατά συνθήκη, θεωρούνται ως διαστασιακά ανεξάρτητες: το μέτρο, το χιλιόγραμμο, το δευτερόλεπτο, το αμπέρ, το kelvin, το μολ, και η καντέλα. Παραγόμενες μονάδες είναι αυτές που σχηματίζονται από συνδυασμό των βασικών μονάδων, σύμφωνα με τις αλγεβρικές σχέσεις που συνδέουν τις αντίστοιχες ποσότητες. Τα ονόματα και τα σύμβολα από τις μονάδες που έχουν σχηματιστεί μπορούν να αντικατασταθούν από ειδικά ονόματα και σύμβολα τα οποία μπορούν με τη σειρά τους να χρησιμοποιηθούν για να σχηματίσουν εκφράσεις και σύμβολα άλλων παραγόμενων μονάδων.

1.3 ΑΝΤΟΧΗ ΤΩΝ ΥΛΙΚΩΝ ΚΑΙ ΠΑΡΑΜΟΡΦΩΣΗ

Η Αντοχή των Υλικών, είναι η επιστήμη, που μελετά τα στοιχεία των κατασκευών και τις αντίστοιχες παραμορφώσεις τους, όταν πάνω σε αυτά ενεργούν εξωτερικές δυνάμεις, με κύριο στόχο τον καθορισμό των συνθηκών που εξασφαλίζουν την αντοχή τους.

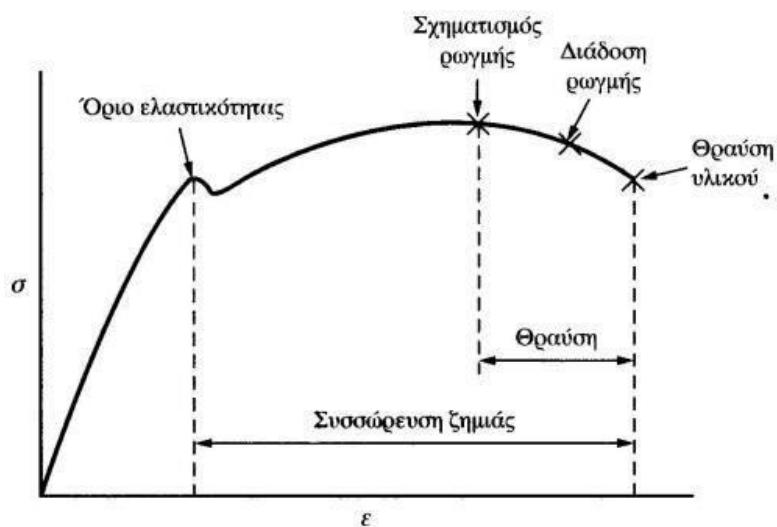
Παραμόρφωση, είναι η απόκριση (αντίδραση) των υλικών στην εφαρμογή της τάσης (δηλ. φυσική μεταβολή όπως επιμήκυνση λόγω εφελκυσμού).

Πειράματα αντοχής υλικών:



Σχήμα 1.1 πείραμα εφελκυσμού.

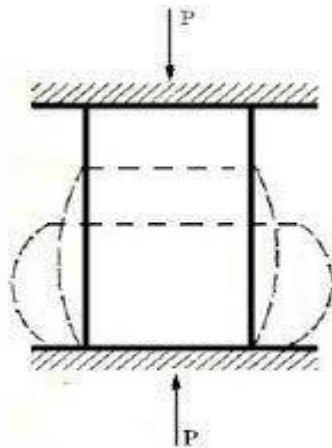
Πείραμα εφελκυσμού (Σχήμα 1.1): Το πείραμα του εφελκυσμού είναι ίσως και το σημαντικότερο πείραμα στην επιστήμη της αντοχής των υλικών. Κατά το πείραμα του εφελκυσμού, ασκούμε στο δοκίμιο δυο δυνάμεις, ίδιου μέτρου, ίδιας διεύθυνσης και αντίθετης φοράς κατά τον κεντροβαρικό διαμήκη άξονα του δοκιμίου, με σημεία εφαρμογής της τάσης τα άκρα του δοκιμίου.



Σχήμα 1.2: Διάγραμμα τάσης-παραμόρφωσης.

Από το πείραμα του εφελκυσμού προκύπτει το διάγραμμα τάσης-παραμόρφωσης (Εικόνα 1.2). Στο διάγραμμα παρατηρούμε δυο βασικές περιοχές, την ελαστική περιοχή η οποία βρίσκεται πριν από το όριο ελαστικότητας και τη πλαστική η οποία είναι μετά από αυτό. Στην ελαστική περιοχή το δοκίμιο επανέρχεται στην

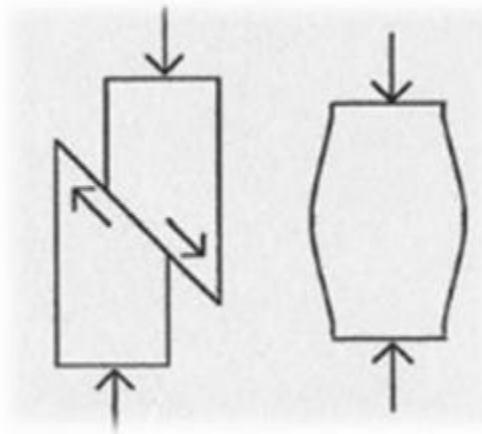
αρχική του μορφή εάν αφαιρέσουμε τις δυνάμεις από αυτό ενώ στη πλαστική το δοκίμιο θα έχει μόνιμες παραμορφώσεις ακόμα και μετά την αφαίρεση των δυνάμεων επίσης σημαντικό σημείο είναι το σημείο θραύσης στο οποίο το δοκίμιο σπάει και χωρίζεται σε δυο μέρη. Στη συνέχεια από το διάγραμμα μπορεί να κατηγοριοποιηθεί το υλικό, οι δυο κυρίες κατηγορίες είναι τα όλκιμα και τα ψαθυρά. Τα ψαθυρά υλικά μετά την ελαστική περιοχή οδηγούνται σε άμεση θραύση ενώ τα όλκιμα μπορούν να φέρουν πλαστικές (μόνιμες) παραμορφώσεις προτού σπάσουν.



Σχήμα 1.3 πείραμα θλίψεως.

Πείραμα θλίψης (Σχήμα 1.3): Κατά το πείραμα της θλίψεως το δοκίμιο φορτίζεται από δυο δυνάμεις ίσου μέτρου, ίδιας διεύθυνσης και αντίθετης φοράς η οποία οδηγεί στη συμπίεση του υλικού.

Η διαδικασία αυτή είναι η ακριβώς αντίθετη από αυτή του πειράματος του εφελκυσμού ως αποτέλεσμα το διάγραμμά τάσης - παραμόρφωσης για τη θλίψη είναι όμοιο με αυτό του εφελκυσμού με κυρία διαφορά το ότι σχεδιάζεται στο 3ο τεταρτημόριο διότι και η παραμόρφωση και οι δυνάμεις είναι αρνητικές.



Σχήμα 1.4 Σημείο θραύσης όλκιμων και ψαθυρών υλικών στο πείραμα θλίψεως.

Στο πείραμα θλίψεως όπως και στο πείραμα του εφελκυσμού τα υλικά χωρίζονται σε όλκιμα και ψαθυρά. Η μεγάλη διαφορά αναμεσα στα υλικά κατά το πείραμα θλίψεως παρατηρείται στο σημείο θραύσης, στο σημείο αυτό τα ψαθυρά υλικά έχουν σχισμό στο μέγιστο διατμητικό επίπεδο, ενώ τα όλκιμα εμφανίζουν μια πλευρική διόγκωση, το φαινόμενο αυτό ονομάζεται "βαρελοποίηση" και είναι το σημείο λήξης του πειράματος (Σχήμα 1.4).

Πείραμα κάμψης: Το πείραμα κάμψης στοχεύει την εφαρμογή καμπτηρών και μόνο σε φορτωμένη δοκό, με τελικό αποτέλεσμα τη κάμψη της και την ανάπτυξη ορθών μηχανικών τάσεων που δεν προέρχονται από αξονικά φορτία αλλά από καμπτικές ροπές, γι' αυτό ονομάζονται καμπτικές τάσεις.

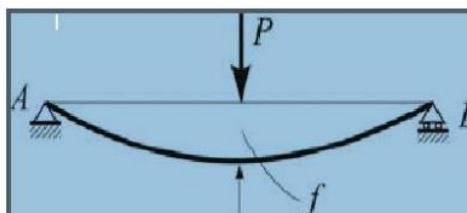
Οι ακόλουθες συνθήκες πρέπει να πληρούνται προκειμένου να έχουμε συνθήκες καθαρής κάμψης:

$$\Sigma F_x = 0$$

$$\Sigma F_y = 0$$

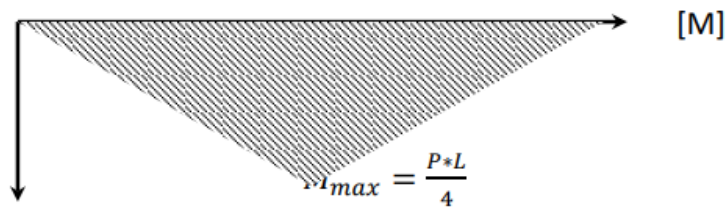
$$\Sigma M_k \neq 0$$

$$\Sigma M_t = 0$$



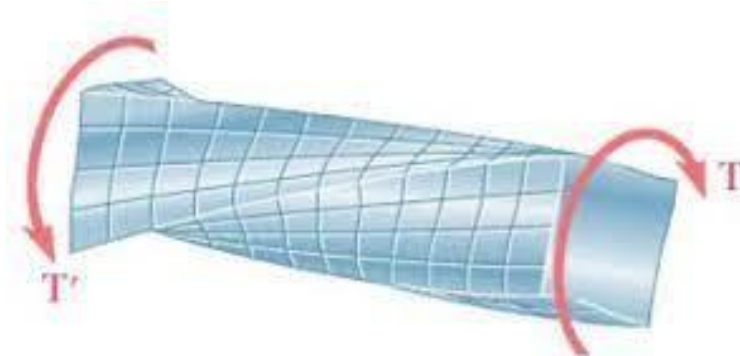
Σχήμα 1.5 Κάμψη τριών σημείων αμφιέριστης δοκού.

Στο σχήμα 1.5 φαίνεται η κάμψη τριών σημείων (3 point bending test), κατά τη κάμψη αυτή η δοκός στηρίζεται αμφιέριστα ή αμφιπροέχουσα και στο μέσον του μήκους της φέρει σημειακό φορτίο P. Κατά την επίλυση των στερεοστατικών εξισώσεων ισορροπίας μπορεί να σχεδιαστεί το διάγραμμα καμπτικών ροπών (Σχήμα 1.6):



Σχήμα 1.6 Διάγραμμα καμπτικών ροπών πειράματος κάμψης τριών σημείων.

Πείραμα στρέψης (Σχήμα 1.7): Το πείραμα στρέψης στοχεύει κύριος ράβδους και δοκούς. Ένα δοκίμιο καταπονείται σε στρέψη όταν σε αυτό ασκούνται δυο δυνάμεις ίσων μέτρων και αντίθετων φορών των οποίων τα επίπεδα είναι κάθετα στον κεντροβαρικό άξονα.

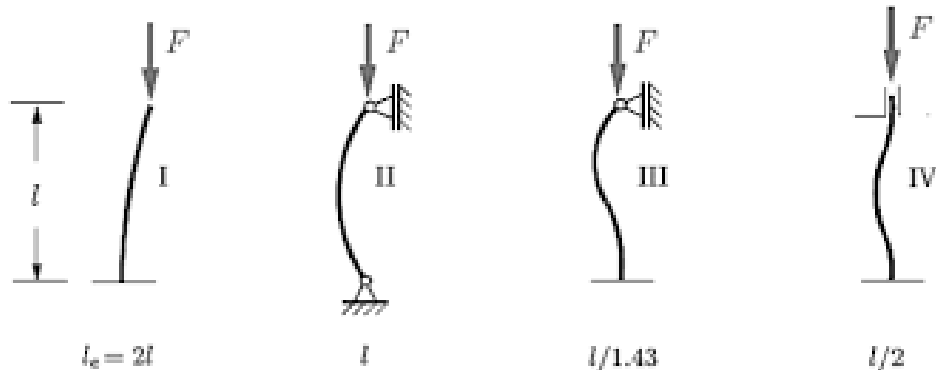


Σχήμα 1.7 Πείραμα στρέψης.

Αποτέλεσμα των δυο δυνάμεων που ασκούνται είναι η διαμόρφωση ροπής σε κάθε διατομή της ράβδου η οποία ονομάζεται ροπή στρέψης ή στερεοτική ροπή.

Πείραμα λογισμού (Σχήμα 1.8): Λογισμό έχουμε όταν ένα ελαστικό δοκίμιο με μορφή ράβδου υποβάλλεται σε κεντρική θλίψη με συνεχώς αυξανόμενο φορτίο. Μετά

από μια οριακή τιμή φορτίου το δοκίμιο μας οδηγείται σε κατάσταση ασταθούς ισορροπίας, η ισορροπία αυτή ονομάζεται ασταθής διότι με τη παραμικρή διαταραχή στο φορτίο το δοκίμιο θα καταρρεύσει.

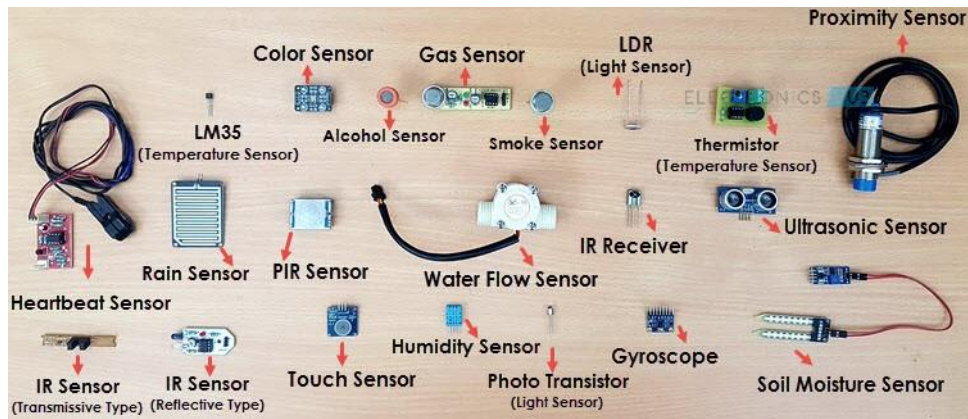


Σχήμα 1.8: Διάφορες μορφές στήριξης σε πείραμα λυγμού.

Η διαταραχή που οδηγεί στη κατάρρευση δεν προέρχεται από αύξηση του φορτίου αλλά από το ίδιο το δοκίμιο λόγο των μικρών ατελειών που έχει, επίσης στη κατάρρευση συντελεί και η μη συμμετρική φόρτιση.

1.4 ΑΙΣΘΗΤΗΡΕΣ ΜΕΤΡΗΣΕΩΝ

Οι αισθητήρες, γνωστοί και ως μετατροπείς, είναι ένα από τα θεμελιώδη δομικά στοιχεία του σύγχρονου κόσμου. Οι αισθητήρες είναι συσκευές οι οποίες αντιδρούν σε αλλαγές στο περιβάλλον τους, οι αλλαγές αυτές μπορεί να είναι φυσικές, χημικές ή ηλεκτρικές. Οι αισθητήρες μετατρέπουν φυσικά φαινόμενα σε σήμα εξόδου το οποίο μπορούμε να επεξεργαστούμε και να αναλύσουμε. Οι αισθητήρες μπορούν να χρησιμοποιηθούν για τη μέτρηση μεγάλου πλήθους φυσικών φαινομένων, με διάφορους μηχανισμούς μετατροπής και διαφορά σήματα εξόδου (Σχήμα 1.9).



Σχήμα 1.9 Διαφορά είδη αισθητήρων.

Οι αισθητήρες κατηγοριοποιούνται με βάση τα υλικά που χρησιμοποιούνται, τις χρήσεις τους, τρόπους παραγωγής, κόστος και ακρίβεια. Υπάρχουν δυο κυρίες κατηγορίες αισθητήρων, οι παθητικοί και οι ενεργοί αισθητήρες. Οι παθητικοί αισθητήρες δεν απαιτούν παροχή καμίας μορφής ενέργειας, στη περίπτωση τους το σήμα εξόδου είναι απευθείας αποτέλεσμα ερεθίσματος εξωτερικής πηγής. Αρά οι παθητικοί αισθητήρες κάνουν απευθείας μετατροπή από σήμα εισόδου σε σήμα εξόδου. Παραδείγματα αυτών είναι οι θερμικοί, χημικοί και σεισμικοί αισθητήρες. Οι ενεργοί αισθητήρες προκειμένου να λειτουργήσουν χρειάζονται μια εξωτερική πηγή ενέργειας, η οποία ονομάζεται σήμα διέγερσης. Προκειμένου να πράξουν σήματα εξόδου οι αισθητήρες τροποποιούν ανάλογα το σήμα διέγερσης. Οι ενεργοί αισθητήρες ονομάζονται και παραμετρικοί αισθητήρες λόγο των ιδιοτήτων τους οι οποίες μπορούν να αλλάξουν με βάση τα ερεθίσματα που δέχονται από το περιβάλλον, οι ιδιότητες αυτές στη συνέχεια μπορούν να μετατραπούν σε ηλεκτρικά σήματα.

Προκειμένου να μπορέσουμε να συνδέουμε τα ερεθίσματα του φυσικού κόσμου με τα ηλεκτρικά σήματα επινοήθηκε η συνάρτηση μεταφοράς. Η συνάρτηση μεταφοράς δείχνει τη σχέση μεταξύ του φυσικού ερεθίσματος (s) και του ηλεκτρικού σήματος εξόδου (S), με $S = f(s)$. Η συνάρτηση αυτή μπορεί να είναι γραμμική ή και μη ανάλογα με τη σχέση εισόδου και εξόδου. Η μη γραμμικότητα μπορεί να έχει πολλαπλές μορφές, κάποιες από αυτές είναι λογαριθμική, εκθετική ή συνάρτηση ισχύος. Η πιο συχνή μορφή με την οποία ορίζονται οι σχέσεις είναι μια πρωτοβάθμια συνάρτηση της μορφής: $S = a + b(s)$, όπου το a συμβολίζει τη τιμή που έχει το σήμα εξόδου για μηδενικό σήμα εισόδου b είναι η κλίση.

Εύρος τιμών εισόδου ή Full-scale input (FSI), είναι η διαφορά ελάχιστης και μέγιστης τιμής εισόδου που μπορούμε να αναπαραστήσουμε με ντεσιμπέλ. Τα ντεσιμπέλ υπολογίζονται έτσι ώστε να είναι ίσα με είκοσι φορές τον λογάριθμο της τιμής εισόδου: $1 \text{ dB} = 20 \log(s_2/s_1)$. Με s_1 και s_2 τις ελάχιστες και μέγιστες τιμές εισόδου.

Εύρος τιμών εξόδου ή Full-scale output (FSO), δείχνει την διαφορά αναμεσά στο μέγιστο και ελάχιστο σήμα εξόδου όταν αντίστοιχα στον αισθητήρα εφαρμοστούν μέγιστα και ελάχιστα ερεθίσματα. Το εύρος τιμών εξόδου επίσης περιλαμβάνει όλες τις αποκλίσεις από την ιδανική συνάρτηση μεταφοράς.

Η ακρίβεια είναι ένα σημαντικό χαρακτηριστικό των αισθητήρων που υπολογίζεται με βάση το σφάλμα στη μέτρηση και ορίζεται ως η διαφορά μεταξύ της μετρούμενης τιμής και της πραγματικής τιμής. Αντιπροσωπεύεται ως % της πλήρους κλίμακας ή % της μέτρησης:

$$\text{Απόλυτο σφάλμα} = | \text{Μετρούμενη τιμή} - \text{Πραγματική τιμή} |$$

Όπου η Πραγματική τιμή υπολογίζεται παίρνοντας τον μέσο όρο ενός άπειρου αριθμού μετρήσεων και το σχετικό σφάλμα μπορεί επίσης να υπολογιστεί ως:

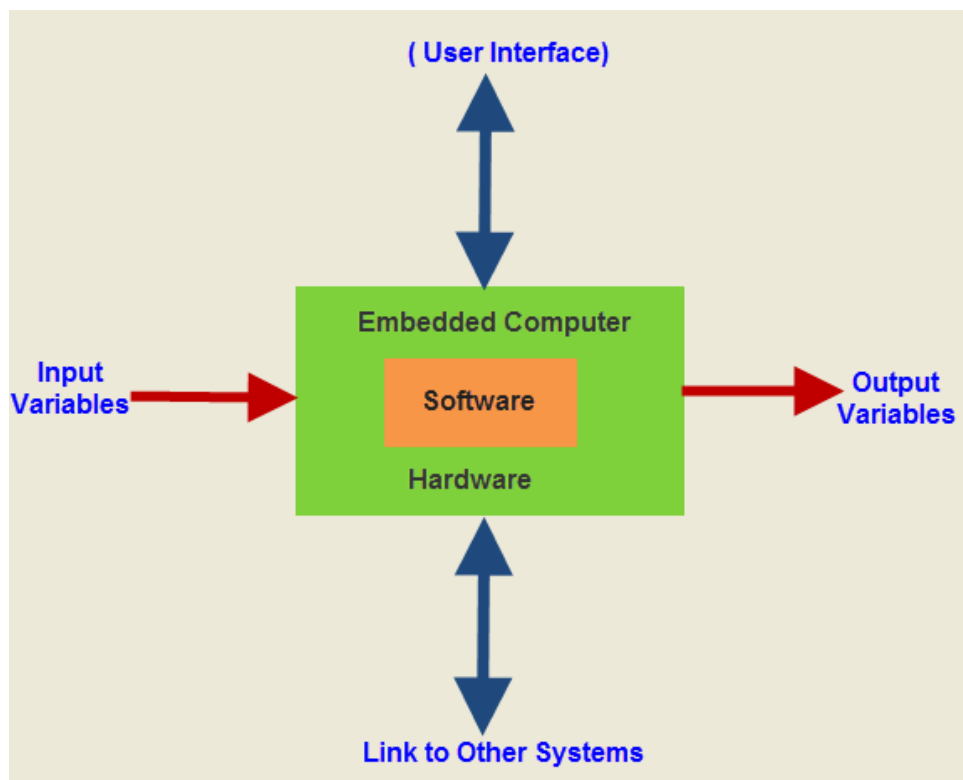
$$\text{Σχετικό σφάλμα} = \text{Απολυτό σφάλμα} / \text{Πραγματική τιμή}$$

Η ακρίβεια υποδεικνύει μια συλλογική επίδραση της γραμμικότητας, βαθμονόμησης, σφαλμάτων επαναληψιμότητας, νεκρής ζώνης κ.λπ., σε μετρήσεις που χρησιμοποιούνται σε αισθητήρες.

1.5 ΕΝΣΩΜΑΤΩΜΕΝΑ ΣΥΣΤΗΜΑΤΑ

Ένα ενσωματωμένο σύστημα είναι ένα σύστημα υπολογιστή που έχει σχεδιαστεί κυρίως για απόδοση σε διάφορες εργασίες όπως η πρόσβαση, η επεξεργασία, η αποθήκευση και επίσης ο έλεγχος των δεδομένων σε διάφορα συστήματα που βασίζονται σε ηλεκτρικές συσκευές. Τα ενσωματωμένα συστήματα είναι ένας συνδυασμός υλικού και λογισμικού όπου Το λογισμικό είναι συνήθως γνωστό ως "μόνιμο πρόγραμμα του λογισμικού της ROM" ή firmware που είναι ενσωματωμένο

στο υλικό. Ένα από τα πιο σημαντικά χαρακτηριστικά αυτών των ενσωματωμένων συστημάτων είναι ότι δίνουν την έξοδο εντός των χρονικών ορίων. Εμείς λοιπόν χρησιμοποιούμε συχνά ενσωματωμένα συστήματα σε απλές και σύνθετες συσκευές (Σχήμα 1.10). Οι εφαρμογές των ενσωματωμένων συστημάτων αφορούν κυρίως πολλές συσκευές όπως το φούρνο μικροκυμάτων, αριθμομηχανές, τηλεχειριστήρια, οικιακή ασφάλεια και συστήματα ελέγχου της κυκλοφορίας.



Σχήμα 1.10 Διάγραμμα δομής ενσωματωμένου συστήματος.

Βασικό χαρακτηριστικό του ενσωματωμένου συστήματος είναι ο συνδυασμός του υλικού συστήματος και του λογισμικού του συστήματος. Ένα ενσωματωμένο σύστημα χρησιμοποιεί μια υλική πλατφόρμα για να εκτελέσει τις λειτουργίες του. Τα ενσωματωμένα συστήματα συναρμολογούνται μαζί με τουλάχιστον έναν μικροεπεξεργαστή/μικροελεγκτή. Έχουν τα στοιχεία όπως διεπαφές εισόδου/εξόδου, μνήμη, διεπαφή με τον χρήστη και οθόνη. Γενικά, ένα ενσωματωμένο σύστημα αποτελείται από τα ακόλουθα:

- Παροχή ηλεκτρικού ρεύματος
- Μνήμη
- Επεξεργαστής

- Χρονοδιακόπτες
- Κυκλώματα εξόδου/εξόδου
- Σειριακές θύρες επικοινωνίας
- SASC (Συστήματα ειδικά κυκλώματα εφαρμογής)

Ενσωματωμένο λογισμικό συστήματος είναι το λογισμικό ενός ενσωματωμένου συστήματος και είναι γραμμένο για να εκτελεί μια συγκεκριμένη λειτουργία. Συνήθως γράφεται σε μια γλώσσα προγραμματισμού υψηλού επιπέδου και στη συνέχεια μεταγλωττίζεται για να προσφέρει κώδικα που μπορεί να αποθηκευτεί μέσα στον μικροεπεξεργαστή/μικροελεγκτή του συστήματος. Ένα ενσωματωμένο λογισμικό συστήματος στοχεύει να τηρήσει τις ακόλουθες προδιαγραφές:

- Ευκολία μνήμης συστήματος
- Ευκολία στην ταχύτητα του επεξεργαστή
- Όταν το ενσωματωμένο σύστημα λειτουργεί συνεχώς, υπάρχει ανάγκη περιορισμού της ισχύος διάχυση για ενέργειες όπως τρέξιμο, στάση και ξύπνημα.

Χαρακτηριστικά ενσωματωμένων συστημάτων:

- Γενικά, ένα ενσωματωμένο σύστημα εκτελεί μια συγκεκριμένη λειτουργία και κάνει το ίδιο συνεχώς.
- Όλα τα υπολογιστικά συστήματα έχουν περιορισμούς στο σχεδιασμό, αλλά αυτοί μπορεί να είναι ιδιαίτερα αυστηροί. Τα μέτρα σχεδίασης είναι ένα μέτρο των χαρακτηριστικών εκτέλεσης όπως το μέγεθος, η ισχύς, το κόστος και επίσης εκτέλεση.
- Πρέπει να αποδίδει αρκετά γρήγορα και να καταναλώνει λιγνή ενέργεια για να αυξήσει τη διάρκεια ζωής της μπαταρίας.
- Πολλά ενσωματωμένα συστήματα θα πρέπει να αντιδρούν συνεχώς σε αλλαγές στο σύστημα και επίσης να υπολογίζουν συγκεκριμένα αποτελέσματα σε πραγματικό χρόνο χωρίς καμία καθυστέρηση. Για παράδειγμα, η λειτουργία cruise control στα οχήματα πρέπει συνεχώς να αποκρίνεται σε αλλαγές ταχύτητας και σε φρεναρίσματα. Πρέπει υπολογίζεται

η επιτάχυνση/επιβράδυνση συχνά σε περιορισμένο χρόνο. Ένας αργός υπολογισμός μπορεί να οδηγήσει το να χαθεί ο έλεγχος του οχήματος.

- Πρέπει να βασίζεται σε μικροελεγκτή ή μικροεπεξεργαστή.
- Πρέπει να απαιτεί μνήμη, όπως το λογισμικό του εισάγει γενικά στη ROM.
- Ένα ενσωματωμένο σύστημα είναι ενσωματωμένο με το υλικό και το λογισμικό, όπου το υλικό μέρος χρησιμοποιείται για ασφάλεια και απόδοση και το λογισμικό χρησιμοποιείται για μεγαλύτερη ευελιξία και δυνατότητες.

Οι βασικές εφαρμογές ενός ενσωματωμένου συστήματος περιλαμβάνουν έξυπνες κάρτες, δικτύωση υπολογιστών, δορυφόρους, τηλεπικοινωνίες, ψηφιακά ηλεκτρονικά είδη, πύραυλοι κ.λπ.

Τα ενσωματωμένα συστήματα ταξινομούνται σε τέσσερις κατηγορίες με βάση την απόδοσή τους και τις λειτουργικές απαιτήσεις τους:

- Αυτόνομα ενσωματωμένα συστήματα
- Ενσωματωμένα συστήματα πραγματικού χρόνου
- Δικτυωμένα ενσωματωμένα συστήματα
- Κινητά ενσωματωμένα συστήματα

Τα ενσωματωμένα συστήματα ταξινομούνται σε τρεις τύπους με βάση την απόδοση του μικροελεγκτή όπως

- Ενσωματωμένα συστήματα μικρής κλίμακας
- Ενσωματωμένα συστήματα μεσαίας κλίμακας
- Εξελιγμένα ενσωματωμένα συστήματα

Αυτόνομα ενσωματωμένα συστήματα, είναι αυτά τα οποία δεν απαιτούν ένα κεντρικό σύστημα όπως έναν υπολογιστή, αλλά λειτουργούν αυτόνομα. Παίρνουν την είσοδο από τις θύρες εισόδου είτε αναλογικές είτε ψηφιακές και την επεξεργάζονται, υπολογίζουν και μετατρέπουν τα δεδομένα και δίνουν τα δεδομένα που προκύπτουν μέσω της συνδεδεμένης συσκευής που είτε ελέγχει, οδηγεί και εμφανίζει τις συνδεδεμένες συσκευές. Παραδείγματα αυτόνομων ενσωματωμένων συστημάτων είναι συσκευές αναπαραγωγής mp3, ψηφιακές φωτογραφικές μηχανές, κονσόλες βιντεοπαιχνιδιών, φούρνοι μικροκυμάτων και συστήματα μέτρησης θερμοκρασίας.

Ένα ενσωματωμένο σύστημα σε πραγματικό χρόνο ορίζεται ως, ένα σύστημα που δίνει μια απαιτούμενη έξοδο σε συγκεκριμένο χρόνο. Αυτοί οι τύποι ενσωματωμένων συστημάτων ακολουθούν τις χρονικές προθεσμίες για την ολοκλήρωση μιας εργασίας. Τα ενσωματωμένα συστήματα πραγματικού χρόνου ταξινομούνται σε δύο τύπους, όπως μαλακά και σκληρά συστήματα σε πραγματικού χρόνου.

Μαλακά ενσωματωμένα συστήματα πραγματικού χρόνου, σε αυτούς τους τύπους ενσωματωμένων συστημάτων ο χρόνος/προθεσμία δεν τηρείται τόσο αυστηρά. Αν η προθεσμία της εργασίας παρήλθε (σημαίνει ότι το σύστημα δεν έδωσε αποτέλεσμα στον καθορισμένο χρόνο) ακόμα αποτέλεσμα ή έξοδος γίνεται αποδεκτό.

Σκληρά ενσωματωμένα συστήματα πραγματικού χρόνου, σε αυτούς τους τύπους ενσωματωμένων συστημάτων τηρείται αυστηρά ο χρόνος/προθεσμία εργασίας. Το έργο του συστήματος πρέπει να ολοκληρωθεί εντός του χρονικού πλαισίου (καθορισμένο χρονικό διάστημα) διαφορετικά το αποτέλεσμα/έξοδος μπορεί να μην γίνει αποδεκτό. Παραδείγματα αυτών είναι τα σύστημα ελέγχου κυκλοφορίας, στρατιωτικά συστήματα άμυνας, εφαρμογές στον τομέα της υγείας.

Δικτυωμένα Ενσωματωμένα Συστήματα είναι οι τύποι ενσωματωμένων συστημάτων που συνδέονται με ένα δίκτυο για πρόσβαση σε πόρους. Το συνδεδεμένο δίκτυο μπορεί να είναι LAN, WAN ή το Διαδίκτυο. Η σύνδεση μπορεί να είναι οποιαδήποτε ενσύρματη ή ασύρματη. Αυτός ο τύπος ενσωματωμένων συστημάτων είναι ο ταχύτερα αναπτυσσόμενος κλάδος σε εφαρμογές ενσωματωμένων συστημάτων. Ο ενσωματωμένος web server είναι ένας τύπος συστήματος όπου όλες οι ενσωματωμένες συσκευές είναι συνδεδεμένοι σε web server και έχουν πρόσβαση και ελέγχονται από ένα πρόγραμμα περιήγησης δικτιού. Παράδειγμα για το ενσωματωμένο σύστημα δικτύων LAN είναι ένα σύστημα οικιακής ασφάλειας στο οποίο βρίσκονται όλοι οι αισθητήρες που συνδέονται και λειτουργούν με πρωτόκολλο TCP/IP.

Κινητά Ενσωματωμένα Συστήματα είναι ενσωματωμένα συστήματα που χρησιμοποιούνται σε φορητές ενσωματωμένες συσκευές όπως κινητά τηλέφωνα, φωτογραφικές μηχανές και mp3 players κ.λπ.Ο βασικός περιορισμός του Αυτές οι συσκευές είναι οι πόροι και ο περιορισμός της μνήμης.

Ενσωματωμένα συστήματα μικρής κλίμακας είναι οι τύποι ενσωματωμένων συστημάτων έχουν σχεδιαστεί με έναν μόνο μικροελεγκτή 8 ή 16 bit, που μπορεί να ενεργοποιηθεί ακόμη και από μπαταρία. Για την ανάπτυξη ενσωματωμένου λογισμικού για μικρή κλίμακα ενσωματωμένα συστήματα, τα κύρια εργαλεία προγραμματισμού είναι editor, assembler, cross assembler και ολοκληρωμένο περιβάλλον ανάπτυξης (IDE).

Ενσωματωμένα συστήματα μεσαίας κλίμακας Αυτοί οι τύποι ενσωματωμένων συστημάτων σχεδιάζονται με έναν μικροελεγκτή μονής ή 16 ή 32 bit, RISC ή DSP. Αυτοί οι τύποι ενσωματωμένων συστημάτων διαθέτουν πολυπλοκότητες υλικές και λογιστικές. Για την ανάπτυξη ενσωματωμένου λογισμικού για ενσωματωμένα συστήματα μεσαίας κλίμακας, το κύρια εργαλεία προγραμματισμού είναι C, C++, JAVA, Visual C++, RTOS, εντοπισμός σφαλμάτων, πηγαίος κώδικας εργαλείο μηχανικής, προσομοιωτής και IDE.

Εξελιγμένα Ενσωματωμένα Συστήματα είναι οι τύποι ενσωματωμένων συστημάτων έχουν τεράστιες πολυπλοκότητες υλικού και λογισμικού και μπορεί να χρειάζονται ASIP, IP, PLA, επεκτάσιμους ή διαμορφώσιμους επεξεργαστές.

Οι πιο βασικές απαιτήσεις στα ενσωματωμένα συστήματα είναι: Η Αξιοπιστία, Η σχέση κόστους-αποτελεσματικότητας, Η Χαμηλή κατανάλωση ενέργειας, Αποτελεσματική χρήση της επεξεργαστικής ισχύος, Αποτελεσματική χρήση της μνήμης

Προκειμένου ένα ενσωματωμένο σύστημα να θεωρηθεί αξιόπιστο πρέπει να λειτουργεί χωρίς την ανάγκη επαναφοράς ή επανεκκίνησης. Αυτό απαιτεί πολύ αξιόπιστο υλικό και λογισμικό. Για παράδειγμα: αν ένα ενσωματωμένο σύστημα σταματήσει λόγω σφάλματος υλικού, το σύστημα πρέπει να επαναρυθμιστεί χωρίς να χρειάζεται ανθρώπινη παρέμβαση. Ωστόσο, οι προγραμματιστές ενσωματωμένου λογισμικού θα πρέπει να κάνουν την αξιόπιστο το υλικό καθώς και του λογισμικό.

Η σχέση κόστους-αποτελεσματικότητας Εάν ένα ενσωματωμένο σύστημα έχει σχεδιαστεί για έναν πολύ ειδικό σκοπό, όπως για διαστημικά συστήματα ή για σταθμό πυρηνικής ενέργειας μπορεί να μην αποτελεί πρόβλημα. Ωστόσο, εάν το ενσωματωμένο σύστημα είναι σχεδιασμένο για μαζική αγορά όπως συσκευές αναπαραγωγής CD, παιχνίδια και φορητές συσκευές το κόστος είναι πολύ σημαντικό.

Το Application Specific Integrated Circuit (ASIC) χρησιμοποιείται από τους σχεδιαστές για να μειώσει τα εξαρτήματα υλικού και συνεπώς το κόστος

Χαμηλή κατανάλωση ενέργειας Τα περισσότερα από τα ενσωματωμένα συστήματα τροφοδοτούνται από μπαταρία και όχι από κεντρική παροχή. Σε αυτή την περίπτωση η κατανάλωση ρεύματος θα πρέπει να ελαχιστοποιηθεί εφόσον οι ενεργειακοί πόροι είναι περιορισμένοι. Για παράδειγμα: με τη μείωση του αριθμού των εξαρτημάτων μπορεί η κατανάλωση ενέργειας να μειωθεί. Καθώς και με τη σχεδίαση του επεξεργαστή για επαναφορά σε λειτουργία χαμηλής κατανάλωσης ενέργειας ή αναστολής λειτουργίας όταν δεν πρέπει να εκτελέσει κάποια λειτουργία.

Προκειμένου να υπάρχει αποτελεσματική χρήση της επεξεργαστικής ισχύος, διατίθεται μεγάλη ποικιλία επεξεργαστών με διαφορετικές επεξεργαστικές ικανότητες. Οι προγραμματιστές πρέπει να έχουν υπόψη τους την επεξεργαστική ισχύ, τη μνήμη και το κόστος ενώ επιλέγουν τον σωστό επεξεργαστή. Η απαίτηση επεξεργαστικής ισχύος καθορίζεται με το μέτρο απόδοσης MIPS (Million Instruction Per Second). Με τη διαθεσιμότητα τόσο πολλών επεξεργαστών, Η επιλογή ενός επεξεργαστή έχει γίνει δύσκολη.

Αποτελεσματική χρήση της μνήμης, τα περισσότερα από τα ενσωματωμένα συστήματα δεν διαθέτουν δευτερεύουσα αποθήκευση, όπως ο σκληρός δίσκος. Το τσιπ μνήμης που διατίθεται στα ενσωματωμένα συστήματα είναι μνήμη μόνο για ανάγνωση (Read Only Memory ή ROM) και τυχαία Πρόσβαση στη μνήμη (Random Access Memory ή RAM.). Καθώς τα περισσότερα από τα ενσωματωμένα συστήματα δεν διαθέτουν δευτερεύουσα αποθήκευση, "φλας μνήμη" χρησιμοποιείται για την αποθήκευση του προγράμματος. Σήμερα μικροελεγκτής και οι επεξεργαστές ψηφιακού σήματος διαθέτουν επίσης ενσωματωμένη μνήμη. Τέτοιοι επεξεργαστές χρησιμοποιούνται για μικρά ενσωματωμένα σύστημα καθώς το κόστος είναι γενικά χαμηλό και η εκτέλεση είναι γενικά γρήγορη.

Κατάλληλος χρόνος εκτέλεσης, σε ενσωματωμένα συστήματα πραγματικού χρόνου, ορισμένες εργασίες πρέπει να εκτελούνται μέσα σε ένα καθορισμένο χρονικό διάστημα. Κανονικά ο υπολογιστής δεν μπορεί να επιτύχει απόδοση σε πραγματικό χρόνο. Επομένως, ειδικό λειτουργικά συστήματα γνωστά ως λειτουργικά συστήματα πραγματικού χρόνου τρέχουν σε αυτά τα ενσωματωμένα συστήματα. Σε σκληρά ενσωματωμένα συστήματα, πραγματικού χρόνου, οι προθεσμίες του ενσωματωμένου

συστήματος πρέπει να τηρούνται αυστηρά, αλλά εντός μαλακού ενσωματωμένου συστήματος σε πραγματικό χρόνο η εργασία ενδέχεται να μην εκτελεστεί έγκαιρα. Ο προγραμματιστής λογισμικού πρέπει να εξακριβώσει εάν το ενσωματωμένο σύστημα είναι σκληρό ή μαλακό και πρέπει να εκτελέσει την ανάλυση απόδοσης ανάλογα.

1.6 ΜΙΚΡΟΕΛΕΓΚΤΕΣ

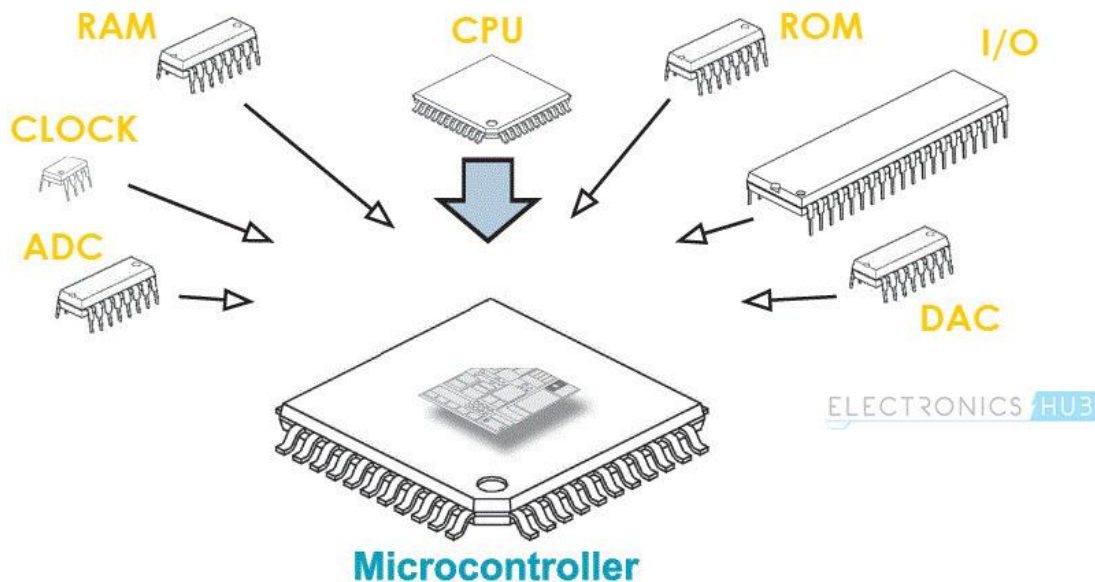
Οι μικροελεγκτές έχουν υπάρξει μόνο για μερικές δεκαετίες, αλλά ο αντίκτυπός τους (άμεσα ή έμμεσα) στις ζωές μας είναι μεγάλος. Συνήθως αυτοί υποτίθεται ότι είναι απλώς επεξεργαστές δεδομένων που εκτελούν εξαντλητικές αριθμητικές πράξεις. Αλλά η παρουσία τους είναι απαραίτητη στο μεγαλύτερο μέρος του κόσμου όπως:

- Σε σούπερ μάρκετ σε Ταμειακές Μηχανές, Ζυγαριές κ.λπ.
- Στο σπίτι σε Φούρνους, Πλυντήρια, Ξυπνητήρια κ.λπ.
- Στο γραφείο σε Φωτοτυπικά, Ανελκυστήρες κ.λπ.
- Στη βιομηχανία σε Βιομηχανικούς Αυτοματισμούς, συστήματα ασφαλείας κ.λπ.
- Σε δρόμους σε Αυτοκίνητα, Σήματα της Τροχαίας κ.λπ.

Οι μικροελεγκτές είναι αυτό που κάνει αυτά τα μηχανήματα έξυπνα. Η δημιουργία εφαρμογών για τους μικροελεγκτές είναι διαφορετική από οποιαδήποτε άλλη εξέλιξη στα ηλεκτρονικά και στους υπολογιστές. Πριν επιλέγει μια συγκεκριμένη συσκευή για μια εφαρμογή, Είναι σημαντική η αντίληψη των διαφορετικών επιλογών και χαρακτηριστικών και τι αυτά μπορεί να σημαίνουν όσον αφορά την ανάπτυξη της εφαρμογής.

Ένας ελεγκτής χρησιμοποιείται για τον έλεγχο κάποιας διαδικασίας. Κάποτε οι ελεγκτές κατασκευάζονταν αποκλειστικά από λογικές πύλες, και ήταν συνήθως μεγάλα, βαριά κουτιά. Αργότερα, μικροεπεξεργαστές χρησιμοποιήθηκαν και ολόκληρος ο ελεγκτής μπορούσε να χωρέσει σε μια μικρή πλακέτα. Καθώς η διαδικασία σμίκρυνσης συνεχιζόταν, όλα τα εξαρτήματα που απαιτούνταν για έναν ελεγκτή συρναμολογούνταν ακριβώς πάνω σε ένα τσιπ. Ως αποτέλεσμα ήρθαν στον κόσμο υπολογιστές και μικροελεγκτες με ένα μόνο τσιπ.

Μια κεντρική μονάδα επεξεργασίας (CPU) ενσωματωμένη σε ένα μόνο τσιπ ονομάζεται μικροεπεξεργαστής, περιέχει αριθμητική και λογική μονάδα (ALU), Μονάδα αποκωδικοποίησης και ελέγχου εντολών, Μητρώο εντολών, Μετρητή προγράμματος (PC), κύκλωμα ρολοί (εσωτερικό ή εξωτερικό), κύκλωμα επαναφοράς (εσωτερικό ή εξωτερικό).



Σχήμα 1.11 Βασικά μέρη μικροελεγκτή.

Ένας μικροελεγκτής είναι ένα ενσωματωμένο τσιπ, το οποίο περιλαμβάνει σε ένα μόνο τσιπ, όλα ή τα περισσότερα από τα εξαρτήματα που χρειάζονται για έναν ελεγκτή (Σχήμα 1.11). Ο μικροελεγκτής συνήθως περιλαμβάνει: CPU (Κεντρική μονάδα επεξεργασίας), RAM (μνήμη τυχαίας πρόσβασης), EPROM / PROM / ROM (Μνήμη μόνο για ανάγνωση με δυνατότητα διαγραφής), I/O (είσοδος / έξοδος) - σειριακή και παράλληλη, χρονόμετρα, ελεγκτή διακοπής.

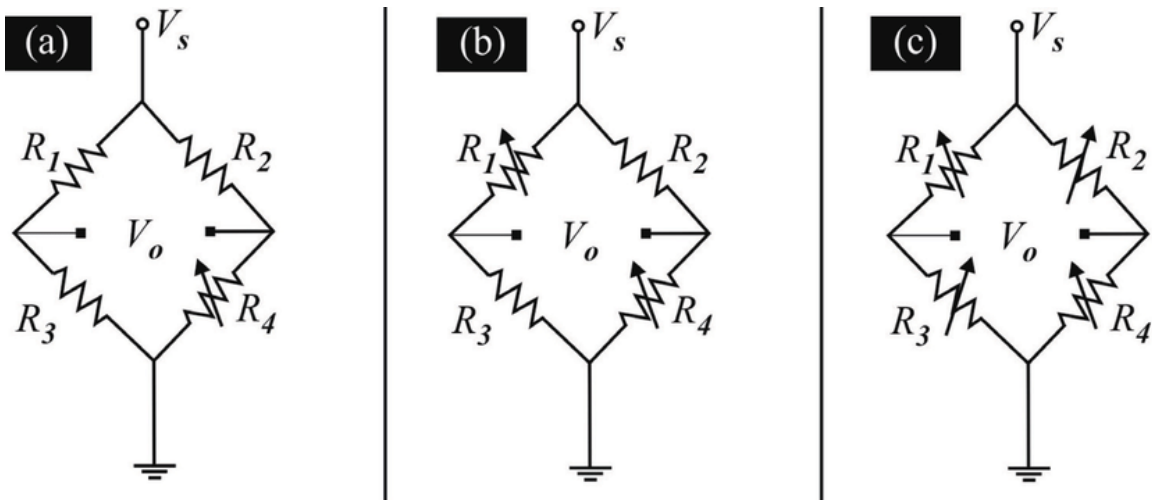
Εκτός από εφαρμογές ελέγχου όπως το σύστημα παρακολούθησης στο σπίτι, οι μικροελεγκτές βρίσκονται συχνά σε ενσωματωμένες εφαρμογές. Ανάμεσα στις πολλές εφαρμογές μπορούν ευκολά να βρεθούν ένας ή και περισσότεροι μικροελεγκτές: εφαρμογές αυτοκινήτων, οικιακές συσκευές (φούρνος μικροκυμάτων, ψυγεία, τηλεοράσεις και στερεοφωνικά), αυτοκίνητα (έλεγχος κινητήρα, διαγνωστικά, έλεγχος κλίματος), περιβαλλοντικός έλεγχος (θερμοκήπια, εργοστάσια, σπίνια), αεροδιαστημική, και χιλιάδες άλλες χρήσεις. Οι μικροελεγκτές χρησιμοποιούνται ευρέως στη ρομποτική. Σε αυτή την εφαρμογή, πολλές εργασίες μπορεί να

κατανεμηθούν σε ένα μεγάλο αριθμό μικροελεγκτών μέσα στο σύστημα. Οι επικοινωνία μεταξύ κάθε μικροελεγκτή και ενός κεντρικού, πιο ισχυρού μικροελεγκτή (ή μικροϋπολογιστή, ή ακόμα και κανονικού υπολογιστή) θα επέτρεπε οι πληροφορίες να επεξεργάζονται από τον κεντρικό υπολογιστή ή να διαβάζεται από άλλους μικροελεγκτές μέσα στο σύστημα. Μια ειδική εφαρμογή για την οποία ταιριάζουν οι μικροελεγκτές είναι η καταγραφή δεδομένων. Μια τέτοια εφαρμογή είναι η παρακολούθηση και να καταγραφή περιβαλλοντικών παραμέτρων (θερμοκρασία, υγρασία, βροχή, κ.λπ.). Το μικρό μέγεθος, η χαμηλή κατανάλωση ενέργειας και η ευελιξία καθιστούν αυτές τις συσκευές ιδανικές για καταγραφή δεδομένων χωρίς επίβλεψη παρακολούθηση και καταγραφή.

1.7 ΓΕΦΥΡΑ WHEATSTONE

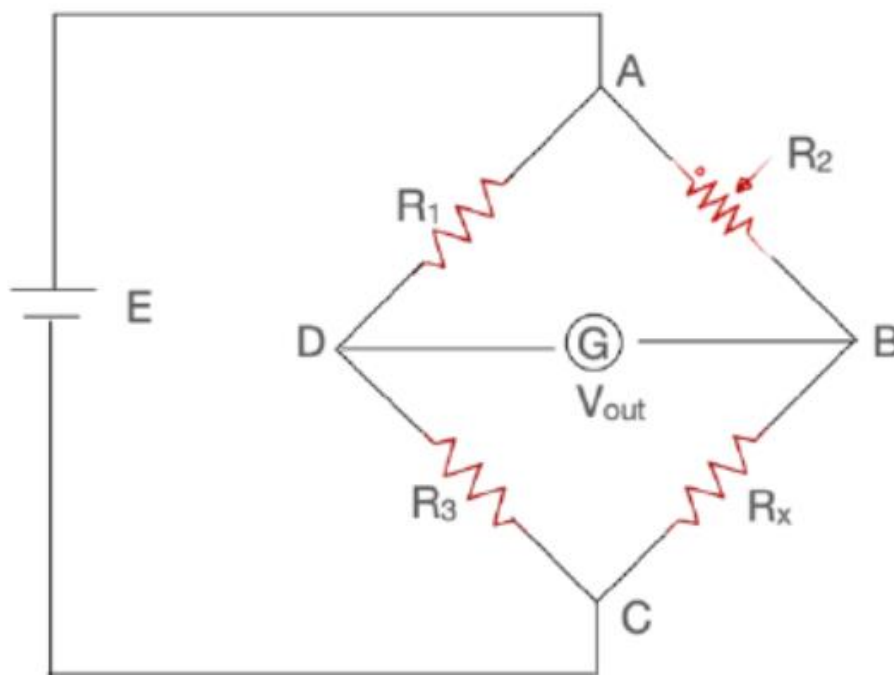
Το 1833 ο S. Hunter Christie δημοσίευσε τη βασική εφαρμογή της μεταγενέστερης γέφυρας Wheatstone στο Philosophical Transactions of the Royal Society of London. Μέσα από μια σειρά πειραμάτων, ο S.H. H Christie κατάφερε να ισορροπήσει μια συσκευή μελετώντας την ισχύ της αγωγιμότητας των αγωγών, η οποία εξαρτάται άμεσα από τη διατομή του σύρματος. Δέκα χρόνια αργότερα, ο Charles Wheatstone στο έργο του "A account of many instruments and processes for determining the constants of voltaic circuits" περιγράφει για πρώτη φορά τη γέφυρα Wheatstone με τη μορφή που τη γνωρίζουμε σήμερα.

Η γέφυρα Wheatstone έχει μια σειρά από εφαρμογές που διακρίνονται σε τρεις βασικούς τομείς: α) τη μέτρηση της ακρίβειας, τις πολύ μικρές αλλαγές στις αντιστάσεις, β) τη μέτρηση των φυσικών παραμέτρων (μηχανικές δυνάμεις, φως, θερμοκρασία κ.λπ.) και γ) τη μέτρηση της επαγωγής, της χωρητικότητας και της σύνθετης αντίστασης. Ενδεικτικές εφαρμογές της γέφυρας Wheatstone είναι αισθητήρες καπνού, αισθητήρες θερμοκρασίας.



Σχήμα 1.12 βασικές παραλλαγές γέφυρας Wheatstone.

Οι βασικές παραλλαγές της γέφυρας Wheatstone φαίνονται στο σχήμα 1.12. Στην περίπτωση α) ένας από τους αντιστάτες είναι μεταβλητός και οι τρεις είναι σταθεροί, στην περίπτωση β) τα δύο αντίθετα είναι σταθερά ενώ τα άλλα δύο είναι μεταβλητά και στην περίπτωση γ) και οι τέσσερις αντιστάσεις είναι μεταβλητές.



Σχήμα 1.13 1/4 γέφυρα Wheatstone.

Το σχήμα 1,13 δείχνει την περίπτωση της γέφυρας Wheatstone 1/4 όπου μόνο μια αντίσταση είναι μεταβλητή και μια είναι άγνωστη. Ένα γαλβανόμετρο τοποθετείται

μεταξύ του BD, το οποίο χρησιμοποιείται για τη μέτρηση του ρεύματος. Είναι ένα όργανο που χρησιμοποιείται για τη μέτρηση χαμηλών ηλεκτρικών τάσεων, ανάλογα με την ύπαρξη ρεύματος και μηχανικής απόκλισης (περιστροφή) ενός πηνίου. Αντιστάσεις R1 και R3 είναι σταθερά ενώ η R2 είναι μεταβλητή. Μεταβάλλοντας την τιμή της αντίστασης R2 καταφέρνουμε να κάνουμε το γαλβανόμετρο να δείχνει μηδέν που δείχνει ότι δεν διαρρέει ρεύμα μεταξύ του BD. Σε αυτή τη συνθήκη το σύστημα είναι ισορροπημένο και ισχύει:

$$\frac{R1}{R3} = \frac{R2}{Rx}$$

Κατά βάση αυτό που συμβαίνει είναι ότι η διαφορά δυναμικού αναμεσά σε B και D είναι μηδέν ($V_{BD} = 0$) και περιγράφεται από:

$$VG = V_{out} = VD - VB = VR3 - VRx = 0$$

Αρά ισχύει:

$$RD = \frac{R3}{R3 + R1}$$

$$RB = \frac{Rx}{Rx + R2}$$

Ενώ σε ισορροπία ισχύουν τα ακόλουθα:

$$RD = RB$$

$$\frac{R3}{R1 + R3} = \frac{Rx}{Rx + R2}$$

$$R3(R2 + Rx) = Rx(R1 + R3)$$

$$R3R2 = RxR1$$

$$Rx = \frac{R3R2}{R1}$$

1.8 BLUETOOTH

Το 1993, ο Jaap Haartsen, μηχανικός ασύρματων επικοινωνιών που εργάστηκε στην Ericsson πηρέ την αποστολή να αναπτύξει μια σύνδεση μικρής εμβέλειας για κινητά τηλέφωνα επειδή τότε, τα καλώδια/σύρματα παρεμπόδιζαν τα συστήματα επικοινωνίας.



Σχήμα 1.14 Jaap Haartsen.

Το 1995 ο Haartsen (που φαίνεται στο σχήμα 1,14) μπόρεσε να αναπτύξει μια τεχνολογία που ονομάζεται πολυεπικοινωνιακός δεσμός με τη βοήθεια του συνεργάτη μηχανικού του, Sven Mattison. Αν και αυτοί οι δύο που πάντα αναφέρονται ως οι δημιουργοί του Bluetooth, ήταν ο CTO της Ericsson που είχε την ιδέα και ξεκίνησε την ανάπτυξη της τέσσερα χρόνια πριν από το Haartsen διοριστεί για το έργο.

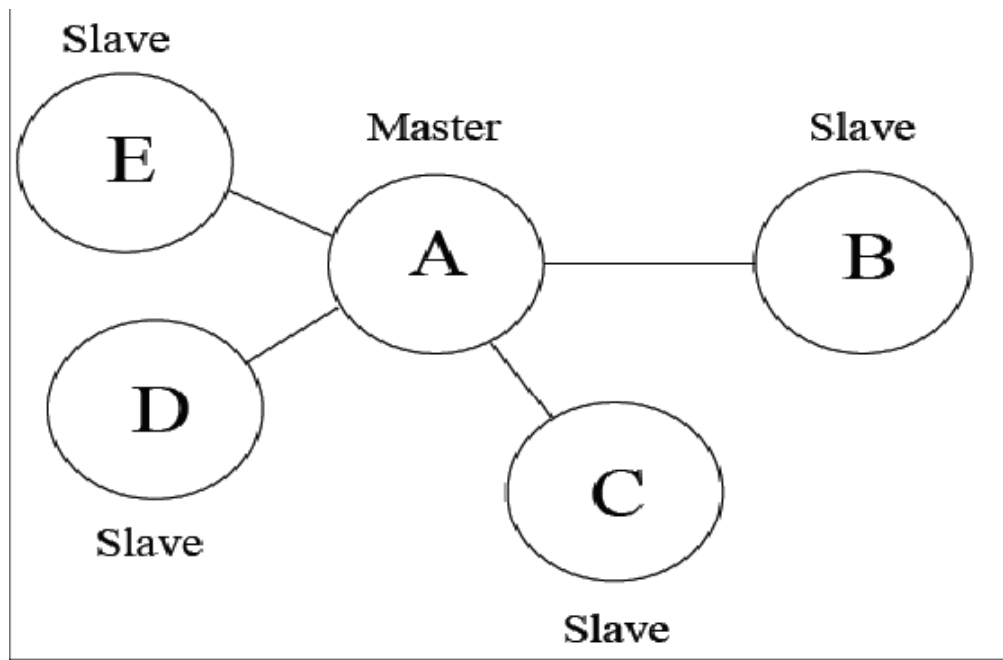


Σχήμα 1.15 Bluetooth adapter.

Όπως κάθε άλλο ασύρματο μέσο επικοινωνίας, το Bluetooth λειτουργεί με την αποστολή και λήψη δεδομένων με τη μορφή ραδιοκυμάτων με τη βοήθεια μιας κάρτας γνωστή ως Bluetooth adapter (Σχήμα 1.15) στη συσκευή με δυνατότητα Bluetooth. Ο Bluetooth adapter έχει τη λειτουργία αποστολής και λήψης δεδομένων και λειτουργεί σε ένα εύρος σύνδεσης. Ο adapter πρέπει να βρίσκεται εντός εμβέλειας για να επικοινωνήσει ο άλλος adapter μαζί του.

Όπως αναφέρθηκε ήδη παραπάνω, το Bluetooth χρησιμοποιεί ραδιοκύματα για την αποστολή δεδομένων. Όταν δύο συσκευές αναζητούν η μία την άλλη για σύζευξη, αναζητούν μια κοινή συχνότητα μέσω του οποίου μπορούν να στέλνουν και να λαμβάνουν δεδομένα. Υπάρχουν 79 κανάλια συχνότητας με συχνότητα 2,4 GHz στα οποία μπορούν να αντιστοιχιστούν οι συσκευές η μια με την άλλη. Όταν η κοινή συχνότητα ανακαλυφθεί, οι συσκευές βρίσκονται και μπορούν στη συνέχεια να στείλουν και να λάβουν δεδομένα. Ένα καλό πράγμα σχετικά με το Bluetooth είναι ότι οι δύο συσκευές που συνδέονται δεν παρεμποδίζουν άλλες συσκευές που συνδέονται στην ίδια περιοχή επειδή υπάρχουν διαφορετικά κανάλια (79) στα οποία μπορούν να επικοινωνούν οι συσκευές.

Ένα άλλο καλό με τη σύνδεση Bluetooth είναι ότι δεν περιορίζεται μόνο σε δύο συσκευές τη στιγμή που σημαίνει ότι πολλές συσκευές μπορούν να στέλνουν και να λαμβάνουν δεδομένα ταυτόχρονα στο ίδιο κανάλι. Όταν συμβεί αυτό, δημιουργείται ένα μικρό δίκτυο και ένα τέτοιο μικροδίκτυο που σχηματίζεται μεταξύ ηλεκτρονικών συσκευών ονομάζεται piconet (Σχήμα 1.16).



Σχήμα 1.16 Διάγραμμα piconet.

Στο piconet, πρέπει να υπάρχουν περισσότερες από δύο συσκευές και όχι περισσότερες από επτά συσκευές. Το Piconet χρησιμοποιεί ένα είδος σύνδεσης master/slave στο οποίο μία από τις συσκευές πρόκειται να είναι master που δημιουργεί ή δίνει εντολή για τη δημιουργία της σύνδεσης.

Όταν μια συσκευή προσπαθεί να δημιουργήσει μια σύνδεση με μια άλλη συσκευή, περνάει μια σειρά διαδικασιών πριν γίνει η σύνδεση, οι οποίες περιλαμβάνουν:

- > Παθητική λειτουργία
- > Ερώτηση
- > Συγχρονισμός
- > Ανακάλυψη σημείου πρόσβασης
- > Δημιουργία καναλιού
- > Σύζευξη
- > Χρήση του δικτύου

Μια συσκευή που λειτουργεί κανονικά βρίσκεται σε παθητική λειτουργία και κατά τη διάρκεια αυτής της περιόδου, ακούει το δίκτυο. Όταν μια κύρια συσκευή (Συσκευή που στέλνει αίτημα) προσπαθεί να δημιουργήσει σύνδεση με συσκευές στην εμβέλειά του, στέλνει ερωτήματα που ονομάζονται «σημεία πρόσβασης» σε αυτές τις

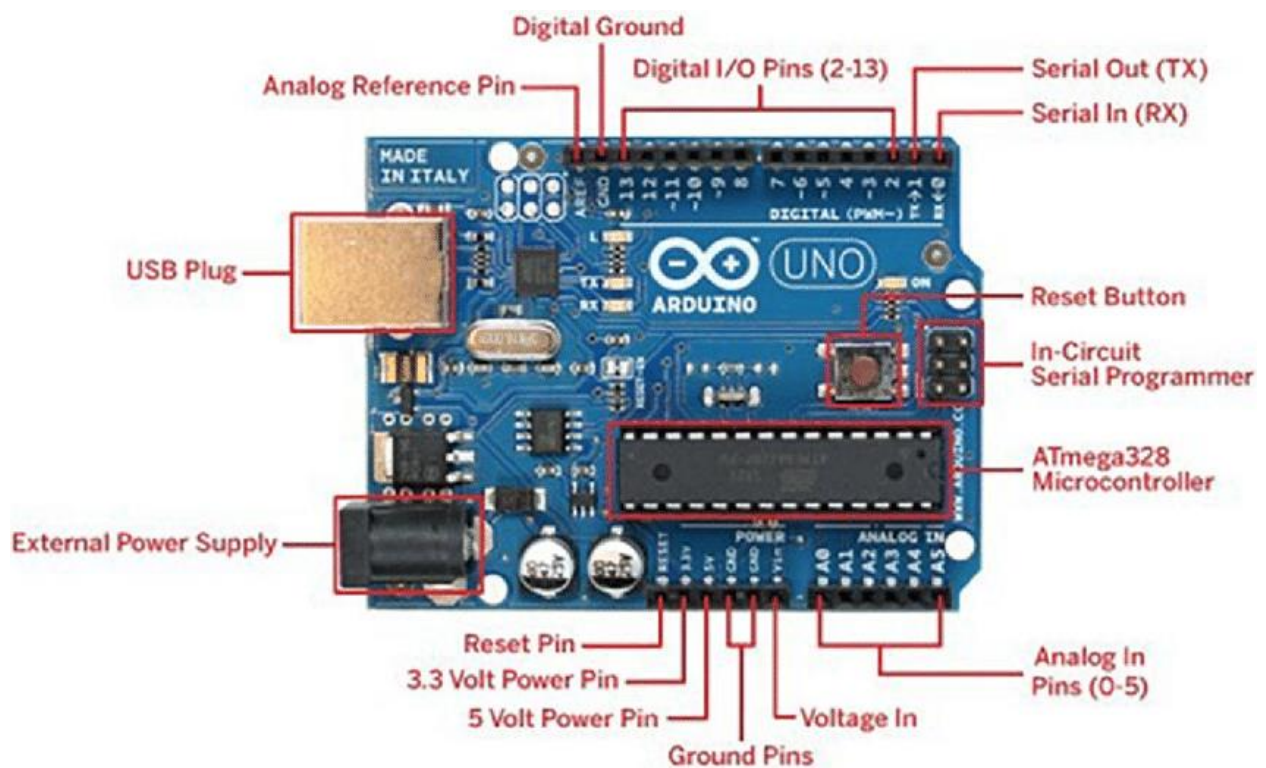
συσκευές και τότε οι συσκευές απαντούν με τη διεύθυνσή τους. Σε μια διαδικασία γνωστή ως "paging", η κύρια συσκευή επιλέγει στη συνέχεια τη διεύθυνση (Αυτό είναι συνήθως το όνομα των συσκευών που βλέπουμε όταν προσπαθούμε να συνδεθούμε σε άλλη συσκευή) και μετά συγχρονίζεται με το σημείο πρόσβασης συγχρονίζοντας το εσωτερικό ρολόι και τη συχνότητα.

Σε αυτό το σημείο, δημιουργήθηκε τώρα ένας σύνδεσμος με το σημείο πρόσβασης που επιτρέπει την κύρια συσκευή για να εισέλθει στη φάση ανακάλυψης σημείου πρόσβασης χρησιμοποιώντας μια μέθοδο που ονομάζεται "service discovery method". Στο τέλος αυτής της φάσης, η κύρια συσκευή είναι τώρα έτοιμη να δημιουργήσει ένα κανάλι επικοινωνίας χρησιμοποιώντας πρωτόκολλο L2CAP Για να περιοριστεί η πρόσβαση άλλων συσκευών στην κρυπτογραφημένη σύνδεση, το σημείο πρόσβασης στέλνει ένα αίτημα σύζευξης στην κύρια συσκευή που ζητά από τον χρήστη να εισαγάγει ένα κωδικό "pin" πριν προχωρήσει στη σύνδεση. Αυτό το αίτημα pin δεν συμβαίνει σε ορισμένες συσκευές. Όταν οι δύο συσκευές έχουν αντιστοιχιστεί, η κύρια συσκευή μπορεί πλέον να χρησιμοποιήσει το καθιερωμένο κανάλι επικοινωνίας.

ΚΕΦΑΛΑΙΟ 2 – HARDWARE

2.1 ARDUINO MICROCONTROLLER

Το Arduino (Σχήμα 2.1) είναι μια προγραμματιζόμενη πλακέτα ανοιχτού κώδικα που μπορεί να ενσωματωθεί σε μια μεγάλη ποικιλία εφαρμογών τόσο απλών όσο και πολύπλοκων. Αυτή η πλακέτα περιέχει ένα μικροελεγκτή που μπορεί να προγραμματιστεί να ανιχνεύει και να ελέγχει αντικείμενα στο φυσικό κόσμο. Με το να ανταποκρίνεται σε αισθητήρες και σήματα εισόδου, το Arduino είναι σε θέση να αλληλοεπιδρά με ένα μεγάλο αριθμό αισθητήρων και εξαρτημάτων όπως LED, κινητήρες και οθόνες. Εξαιτίας της ευελιξίας και του χαμηλού κόστους, το Arduino έχει γίνει μια πολύ δημοφιλής επιλογή για κατασκευαστές. Το Arduino παρουσιάστηκε το 2005 στην Ιταλία από τον Massimo Banzi ως ένας τρόπος για τους μη μηχανικούς να έχουν πρόσβαση σε ένα χαμηλού κόστους εργαλείο για τη δημιουργία ηλεκτρονικών κατασκευών. Δεδομένου ότι η πλακέτα είναι ανοιχτού κώδικα, κυκλοφορεί με Creative Commons άδεια που επιτρέπει σε οποιονδήποτε να παράγει τη δικιά του πλακέτα



Σχήμα 2.1 Arduino UNO.

Μία από τις πιο δημοφιλείς πλακέτες Arduino είναι το Arduino UNO. Αν και δεν ήταν στην πραγματικότητα η πρώτη πλακέτα που κυκλοφόρησε, παραμένει η πιο ευρέως αναγνωρισμένη στην αγορά. Λόγω της εξαιρετικής δημοτικότητάς του, το Arduino Uno διαθέτει πολλά εργαλεία εκπαιδευτικών προγραμμάτων και φόρουμ έργων στον ιστό που μπορούν να είναι χρήσιμα κατά τη διαμόρφωση ενός συστήματος.

Ανάλυση της πλακέτας:

Εδώ είναι τα εξαρτήματα που συνθέτουν μια πλακέτα Arduino και τι καθένα από αυτά λειτουργίες είναι.

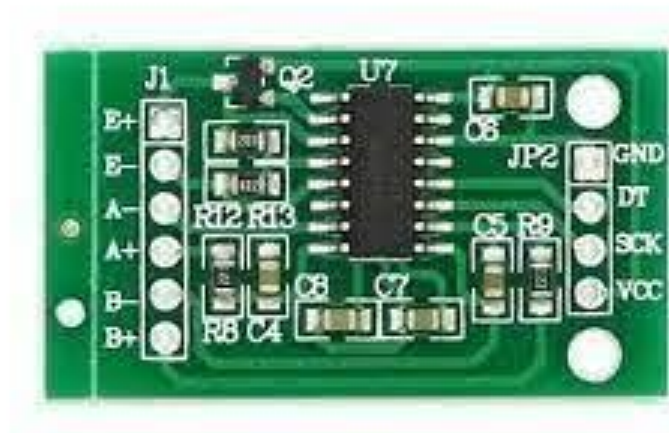
1. Reset Button – Αυτό θα επανεκκινήσει οποιονδήποτε κωδικό έχει φορτωθεί στην πλακέτα του Arduino.
2. Analog Reference pin – Χρησιμοποιείται για να ορίσετε μια εξωτερική τάση αναφοράς.
3. Ground Pins – Υπάρχουν μερικές ακίδες γείωσης στο Arduino και όλες λειτουργούν ίδιο.
4. Digital I/O – Οι ακίδες 0-13 μπορούν να χρησιμοποιηθούν για μια ψηφιακή είσοδο ή έξοδο.
5. Σύνδεση USB – Χρησιμοποιείται για την τροφοδοσία του Arduino και τη φόρτωση κώδικα.
6. TX/RX – Μετάδοση και λήψη δεδομένων ένδειξης.
7. Μικροελεγκτής ATmega – Αυτός είναι ο εγκέφαλος και εκεί βρίσκεται αποθηκευμένο πρόγραμμα.
8. DC Power Barrel Jack – Χρησιμοποιείται για την τροφοδοσία του Arduino.
9. 3,3 V Pin – Αυτή η ακίδα παρέχει ισχύ 3,3 βολτ στη κατασκευή σας.
10. 5 V Pin – Αυτή η ακίδα παρέχει 5 βολτ ρεύματος στη κατασκευή σας.
11. Analog Pins – Αυτές οι ακίδες μπορούν να διαβάσουν το σήμα από έναν αναλογικό αισθητήρα και να το μετατρέψουν σε ψηφιακό.
12. Rreset pin - Εάν αυτή η ακίδα δεχτεί τάση κάνει επαναφορά στη πλακέτα.

Το Arduino Uno χρειάζεται μια πηγή ρεύματος για να λειτουργήσει και να μπορεί να τροφοδοτηθεί με ποικίλους τρόπους. Ο πιο συχνός είναι η άμεση σύνδεση της πλακέτας με τον υπολογιστή μέσω καλωδίου USB, άλλος τρόπος προκειμένου να γίνει

φορητή η συσκευή είναι η χρήση μιας μπαταρίας 9V, επίσης μπορεί χρησιμοποιηθεί τροφοδοτικό προκειμένου να παρακαμφθεί ο υπολογιστής.

2.2 HX711

Βασισμένο στη πατενταρισμένη τεχνολογία Avia Semiconductor, το HX711 (Σχήμα 2.2) είναι ένας μετατροπέας ακριβείας 24-bit (ADC) που μετατρέπει αναλογικό σήμα σε ψηφιακό, έχει σχεδιαστεί για ζυγαριές και εφαρμογές βιομηχανικού ελέγχου σε συνδυασμό με έναν αισθητήρα γέφυρας (γέφυρα Wheatstone). Το multiplexer εισόδου συνδέει οποιαδήποτε είσοδο A ή B στον προγραμματιζόμενο ενισχυτή (PGA). Κανάλι A μπορεί να προγραμματιστεί με κέρδος 128 ή 64, που αντιστοιχεί σε μια διαφορική είσοδο πλήρους κλίμακας τάσης $\pm 20\text{mV}$ ή $\pm 40\text{mV}$ αντίστοιχα, όταν μια τροφοδοσία 5V είναι συνδεδεμένη στην αναλογική τροφοδοσία AVDD. Το κανάλι B έχει σταθερό κέρδος 32. Ο ρυθμιστής τροφοδοσίας Onchip (Σχήμα 2.3) εξαλείφει την ανάγκη για έναν εξωτερικό ρυθμιστή τροφοδοσίας για παροχή αναλογικού ισχύος για το ADC και τον αισθητήρα. Το clock input είναι εύκαμπτο. Μπορεί να είναι από εξωτερική πηγή ή τον ταλαντωτή στο τσιπ που δεν απαιτεί κάποιο εξωτερικό εξάρτημα. Το κύκλωμα power-on-reset στο chip απλοποιεί την ψηφιακή διεπαφή αρχικοποίησης. Δεν χρειάζεται προγραμματισμός του μετατροπέα. Όλοι οι έλεγχοι του HX711 είναι μέσα από τα pins.



Σχήμα 2.2 HX711.

Ανάλυση της πλακέτας:

E+ = θετικό άκρο τροφοδοσίας

E- = αρνητικό άκρο τροφοδοσίας

A+ = Αναλογική είσοδος κανάλι A θετικό

A- = Αναλογική είσοδος κανάλι A αρνητικό

B- = Αναλογική είσοδος κανάλι A θετικό

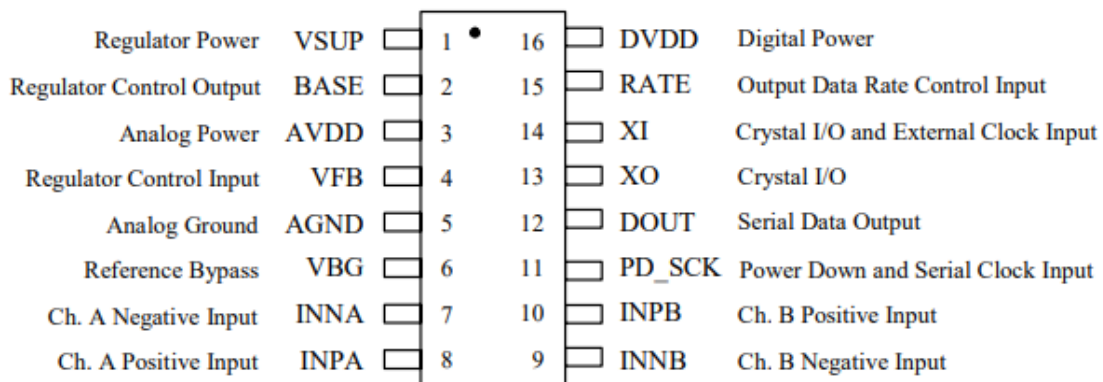
B+ = Αναλογική είσοδος κανάλι A αρνητικό

GND = γείωση

DT = σειριακή έξοδος δεδομένων

SCK = σειριακό ρολόι

VCC = παροχή τάσης 5V



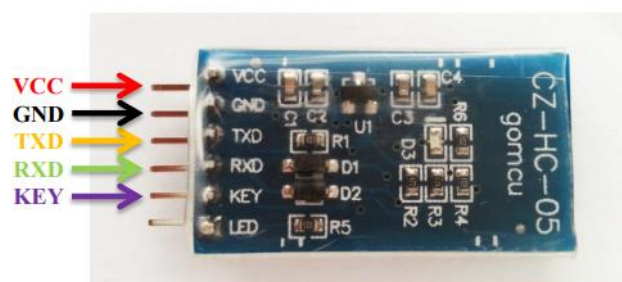
Σχήμα 2.3 πηνία μικροσίπ που φέρει ο μετατροπέας HX711.

Προκειμένου να υπάρχει σωστή τροφοδοσία του HX711 πρέπει η παροχή να είναι η ίδια με αυτή του μικροελεγτή κοινώς δε μπορεί να τροφοδοτείται από διαφορετική πηγή ενέργειας. Όταν χρησιμοποιείται εσωτερικός ρυθμιστής παροχής τάσης, Η διαφορά αναμεσα σε τάση εισόδου και εξόδου εξαρτάται από το εξωτερικό τρανζίστορ που χρησιμοποιείται. Η τάση εξόδου είναι ίση με $V_{AVDD} = V_{BG} \cdot (R1 + R2) / R1$. Αυτή η τάση θα πρέπει να σχεδιάζεται το ελάχιστο 100 mV κάτω από την τάση VSUP. Εάν ο ρυθμιστής αναλογικής παροχής στο chip δεν είναι χρησιμοποιείται, ο ακροδέκτης VSUP θα πρέπει να συνδεθεί σε ένα από τα δύο AVDD ή DVDD, ανάλογα με το ποια τάση είναι πιο ψηλά.

Ο λόγος για τον οποίο χρησιμοποιείται το HX711 στην εφαρμογή μας είναι για να βελτιώσει την ακρίβεια και το εύρος τιμών τις οποίες μπορεί να μετρήσει ο μικροελεγκτής arduino UNO. Ο μικροελεγκτής στη προκειμένη εφαρμογή είναι ικανός να μετρήσει τιμές από 0V έως και 5V , και έχει διακριτική ικανότητα από 0-1024 το οποίο σημαίνει ότι η μικρότερη τιμή την οποία μπορούμε να μετρήσουμε είναι $5V/1024 = 4,88mV$. Η ακρίβεια αυτή όμως δεν είναι ικανοποιητική για τη μέτρηση της παραμόρφωσης διότι οι διάφορες τάσης που θα μετρηθούν είναι υπερβολικά μικρές και έτσι χρησιμοποιήθηκε ο μετατροπέας HX711 ο οποίος αυξάνει το εύρος τιμών από 0-1024 σε 0 – 16777216 πράγμα το οποίο αυξάνει τη διακριτική ικανότητα από 4,88mV σε $5V/16777216 = 0,298 \mu V$, αίτιο του εύρους αυτού είναι το ότι ο μετατροπέας HX711 είναι 24bit πράγμα το οποίο του δίνει τη διακριτική ικανότητα $2^{24} = 16777216$.

2.3 BLUETOOTH MODULE HC-05

Το HC-05 (Σχήμα 2.4&2.5) είναι μια δημοφιλής μονάδα bluetooth. Μπορεί να χρησιμοποιηθεί προκειμένου να προσθέσει ασύρματη επικοινωνία σε οποιοδήποτε κατασκευή με αποτέλεσμα τη δυνατότητα επικοινωνίας μεταξύ μικροελεγκτών η μικροελεγκτή - άλλης συσκευής, στη περίπτωση της συσκευής που διαμορφώθηκε έχουμε επικοινωνία Arduino UNO - κινητό τηλέφωνο. Η μονάδα επικοινωνεί με τη βοήθεια του USART με ρυθμό baud 9600, επομένως είναι εύκολη η διασύνδεση με οποιονδήποτε μικροελεγκτή που υποστηρίζει USART. Μπορούμε επίσης να διαμορφώσουμε τις προεπιλεγμένες τιμές της μονάδας χρησιμοποιώντας τη λειτουργία εντολής. Ωστόσο, η μονάδα αυτή δεν είναι ικανή να μεταφέρει μεγάλο όγκο δεδομένων όπως φωτογραφίες, βίντεο και τραγούδια.



Σχήμα 2.4 HC-05 pins.

Pin	περιγραφή	λειτουργία
-----	-----------	------------

VCC	5 V	Συνδέεται σε παροχή ρεύματος
GND	γείωση	Συνδέεται σε γείωση
TXD	Σειριακή αποστολή δεδομένων μέσω bluetooth	Συνδέεται στο RXD Pin του μικροελεγκτή
RXD	Σειριακή λήψη δεδομένων μέσω bluetooth	Συνδέεται στο TXD Pin του μικροελεγκτή
KEY	Αλλαγή λειτουργίας εισόδου	Εάν είναι συνδεδεμένο σε τίποτα η σε χαμηλή είσοδο είναι σε λειτουργία pairing, αλλιώς αν είναι συνδεδεμένο σε υψηλό τότε είναι σε σύνδεση

Πίνακας 1 : Ανάλυση των pin του Bluetooth module.



Σχήμα 2.5 HC-05.

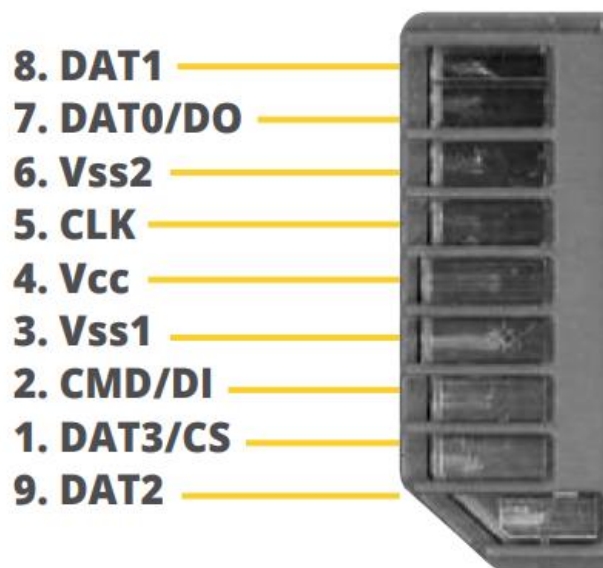
Στη περίπτωση της παρούσας εφαρμογής το bluetooth χρησιμοποιείται για επικοινωνία arduino UNO – smartphone, όπου για να γίνεται σωστά η ανταλλαγή δεδομένων διαμορφώθηκε ειδική εφαρμογή για το smartphone του χρήστη η οποία είναι υπεύθυνη για την λήψη, αναπαράσταση δεδομένων και αποστολή εντολών το χρήστη στο arduino UNO, η ανάλυση την οποίας θα γίνει στη συνέχεια της εργασίας.

Θετικό του bluetooth στο κομμάτι του προγραμματισμού είναι το ότι δεν απαιτείται η χρήση κάποιας βιβλιοθήκης, ειδικών εντολών ή συνάρτησης. Τα μηνύματα που κοινοποιεί το Arduino κοινώς ένα μήμα που θα έβλεπε ο χρήστης στην οθόνη του υπολογιστή αυτόματα θα είχε σταλθεί και μέσω του bluetooth.

2.4 SDCARD MODULE

Η e Secure Digital Card (κάρτα SD, για συντομία) παρουσιάστηκε αρχικά για να καλύψει την ανάγκη για της αξιόπιστης αποθήκευσης δεδομένων σε μικρή μορφή για φορητές ηλεκτρικές συσκευές, όπως κινητά τηλέφωνα, φωτογραφικές μηχανές κ.λπ. Πρόσφατα, υπάρχει αυξανόμενο ενδιαφέρον για την υιοθέτηση της κάρτας SD για χρήση και στη βιομηχανία.

Οι κάρτες SD (Σχήμα 2.6) όπως και οι κάρτες υπολογιστή ή οι κάρτες CF περιέχουν έναν εσωτερικό ελεγκτή που χειρίζεται όλες τις εσωτερικές λειτουργίες της flash μνήμης. Η μεταφορά δεδομένων μεταξύ του κεντρικού υπολογιστή και της κάρτας γίνεται μέσω λειτουργίας σειριακού ρολογιού, σε μπλοκ των 512 byte. Τα πιο συνηθισμένα συστήματα αρχείων είναι FAT12/16 και FAT 32.



Σχήμα 2.6 πλάκες επαφής κάρτας SD.

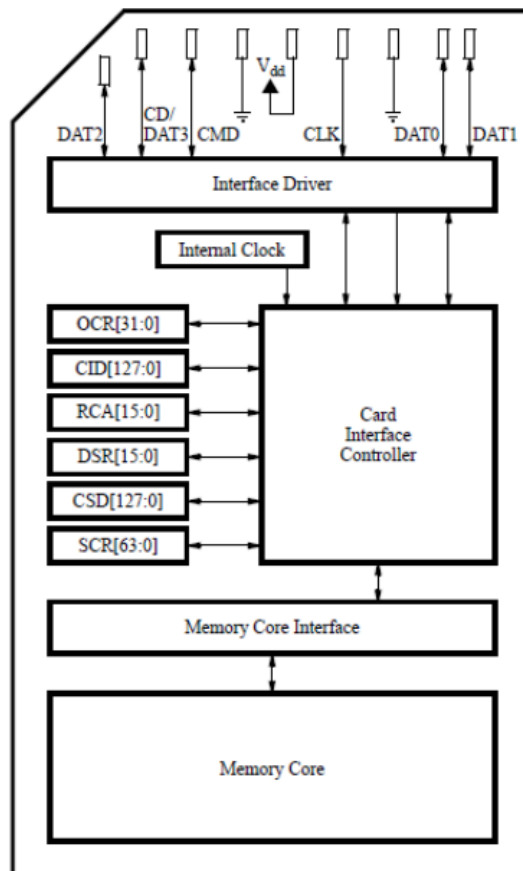
Σε αντίθεση με άλλες κάρτες μνήμης οι κάρτες SD δε χρησιμοποιούν pins για να κάνουν επαφή αλλά πλάκες επαφής όπως φαίνεται στο σχήμα 2,6. Ο λόγος για τον οποίο το 9ο pin διαφέρει στη τοποθέτηση του από τα υπόλοιπα είναι διατηρείται η συμβατότητα με κάρτες MMC.

Οι τυπικές κάρτες SD λειτουργούν σε ένα εύρος τροφοδοσίας 2,7-3,3 V. Ειδικές κάρτες SD χαμηλής τάσης μπορούν να λειτουργούν σε επίπεδο τροφοδοσίας έως και 1,6 V. Οι κάρτες SD λειτουργούν σε δύο ταχύτητες. Η προεπιλεγμένη ταχύτητα λειτουργίας είναι 0-25 MHz ενώ η λειτουργία υψηλής ταχύτητας είναι 0-50MHz.

Υπάρχουν δύο τρόποι διασύνδεσης με κάρτες SD - λειτουργία SPI και λειτουργία SDIO. Η λειτουργία SDIO είναι πιο γρήγορη, αλλά είναι πιο περίπλοκη και, απαιτεί υπογραφή Non Disclosure Agreement. Για αυτόν τον λόγο, δεν συναντιέται συχνά το SDIO. Αντίθετα, κάθε κάρτα SD έχει λειτουργία SPI «χαμηλότερης ταχύτητας», εύκολο στη χρήση για κάθε μικροελεγκτή. Η λειτουργία SPI απαιτεί τέσσερα pins.

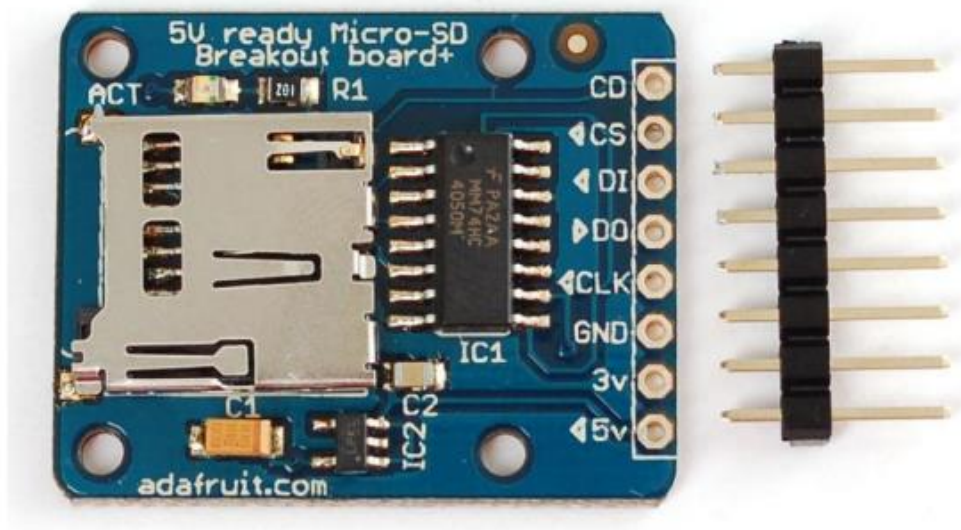
Οι κάρτες SD διατίθενται σε δύο δημοφιλείς εκδοχές - microSD και SD. Η διεπαφή, ο κώδικας, δομή, κλπ είναι όλα ίδια. Η μόνη διαφορά είναι το μέγεθος. Οι microSD είναι πολλά μικρότερο σε φυσικό μέγεθος.

Οι κάρτες SD είναι «ακατέργαστος» χώρος αποθήκευσης. Είναι απλώς τομείς σε ένα τσιπ flash αρά δεν υπάρχει δομή που πρέπει να χρησιμοποιηθεί (Σχήμα 2.7). Αυτό σημαίνει ότι μια κάρτα SD μπορεί να διαμορφωθεί ώστε να είναι της μορφής ενός συστήματος αρχείων Linux, ένα σύστημα αρχείων FAT (DOS) ή ένα σύστημα αρχείων Mac. Μπορεί επίσης να μην υπάρχει κανένα σύστημα αρχείων. Ωστόσο, το 99% των υπολογιστών, των καμερών, των καταγραφών GPS κλπ απαιτούν FAT16 ή FAT32 για το σύστημα αρχείων. Το καλό στη προκείμενη περίπτωση για τους μικροελεγκτες είναι ότι δεν απαιτεί πολύ από τη μνήμη flash και RAM εφόσον δεν υπάρχει συνθέτη μορφή χειρισμού αρχείων.



Σχήμα 2.7 Block diagram κάρτας SD.

Προκειμένου να χρησιμοποιηθεί κάρτα SD χρειάστηκε να προστεθεί στο Arduino μια αποσπώμενη πλακέτα (Σχήμα 2.8). Στην παρούσα εφαρμογή επιλέχθηκε η αποσπώμενη πλακέτα της Adafruit.



Σχήμα 2.8 Adafruit SDcard breakout board.

Επειδή οι κάρτες SD απαιτούν μεγάλη μεταφορά δεδομένων, έχουν τη καλύτερη απόδοση όταν συνδέονται σε SPI pins στον μικροελεγκτή. Τα SPI pins του μικροελεγκτή είναι πολύ πιο γρηγορά από τα αλλά pin του Arduino. Για τα «κλασικά» Arduinos, όπως τα Duemilanove/Diecimila/Uno, αυτές οι ακίδες είναι ψηφιακές 13 (SCK), 12 (MISO) και 11 (MOSI). Απαιτείται και ένα τέταρτο pin για γραμμή 'chip/secondary select' (SS). Συνηθώς αυτό είναι το pin 10, αλλά μπορούν να χρησιμοποιηθούν και άλλα pin.

Συνδεσμολογία:

Breakout board Pin	Arduino UNO Pin
5V	5V
GND	GND
CLK	13
DO	12
DI	11
CS	10

Πίνακας 2 : Συνδεσμολογία SDcard breakout board με Arduino.

Επίσης υπάρχει και το pin CD το οποίο χρησιμοποιείται για ελέγξουμε αν υπάρχει κάρτα SD. Προκειμένου να χρησιμοποιηθεί αυτό το pin πρέπει να του

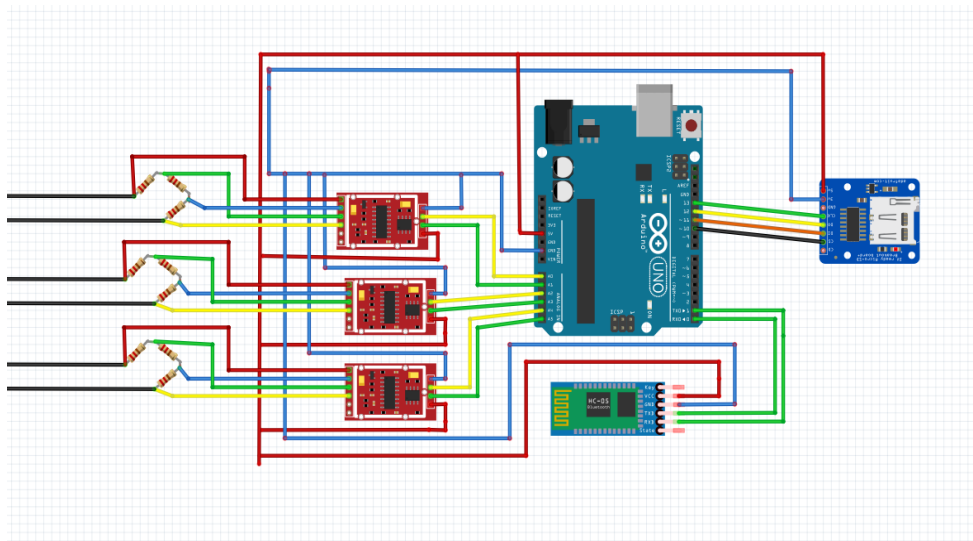
προσθέσουμε μια αντίσταση 10KΩ. Στη παρούσα εφαρμογή δεν χρησιμοποιήθηκε το pin αυτό.

Η σύνδεση με μια κάρτα SD είναι πολύ δύσκολη αλλά υπάρχουν βιβλιοθήκες οι οποίες λύνουν το πρόβλημα αυτό. Στη προκείμενη περίπτωση η βιβλιοθήκη που χρησιμοποιήθηκε είναι η SD.h η οποία διαμορφώθηκε από τον William G. Οι βιβλιοθήκες στο προγραμματισμό αποτελούν έτοιμα κομμάτια κώδικα τα οποία περιέχουν συναρτήσεις (functions) με αποτέλεσμα ο χρήστης τους να μπορεί τις καλεί χωρίς να χρειάζεται να τις γράψει ο ίδιος. Η βιβλιοθήκη SD.h δίνει τη δυνατότητα στο χρήστη να δημιουργεί αρχεία/έγγραφα (txt) στην κάρτα και στη συνέχεια να αποθηκεύει δεδομένα μέσα σε αυτά. Στην εφαρμογή αυτή η κάρτα χρησιμοποιείται για την αποθήκευση των μετρήσεων παραμόρφωσης.

ΚΕΦΑΛΑΙΟ 3 – ΣΥΝΔΕΣΜΟΛΟΓΙΑ ΚΑΙ ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΣ

3.1 ΣΥΝΔΕΣΜΟΛΟΓΙΑ

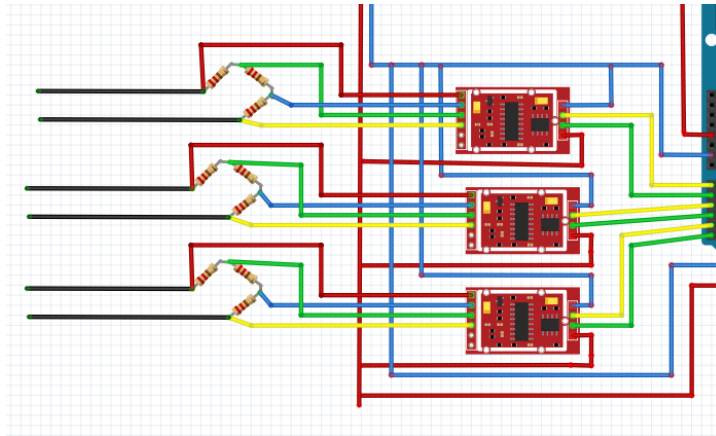
Τα εξαρτήματα τα οποία αναλυθήκαν στο προηγούμενο κεφάλαιο συναρμολογήθηκαν καταλληλά προκειμένου να λειτουργήσουν ταυτόχρονα και σε συνδυασμό μεταξύ τους.



Σχήμα 3.1 συνδεσμολογία της τελικής συσκευής.

Στο σχήμα 3,1 φαίνεται η καλωδίωση της διάταξης που υλοποιήθηκε. Όπως φαίνεται και στο σχήμα τα εξαρτήματα έχουν όλα κοινή παροχή ενέργειας ο λόγος για τον οποίο έγινε αυτό είναι η απλοποίηση της καλωδίωσης εφόσον η τάση είναι αρκετή για να τροφοδοτήσει όλα τα εξαρτήματα. Αντίστοιχη διαδικασία ακολουθήθηκε και για τη γείωση.

Συνδεσμολογία hx711:

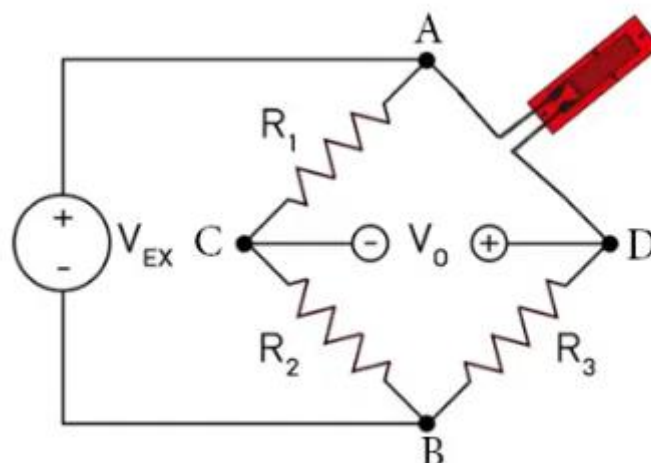


Σχήμα 3.2 συνδεσμολογία των HX711.

Όπως φαίνεται στο σχήμα 3,2 οι μετασχηματιστές HX711 έχουν όλοι την ίδια συνδεσμολογία.

HX711 (1)	
DT	A0
SCK	A1
HX711 (2)	
DT	A2
SCK	A3
HX711 (3)	
DT	A4
SCK	A5

Πίνακας 3 : Συνδεσμολογία HX711 με Arduino.



Σχήμα 3.3 συνδεσμολογία γέφυρας wheatstone.

Κατά την καλωδίωση της γέφυρας όπως φαίνεται και στο σχήμα 3,3 παρέχουμε μια σταθερή τάση στα δυο της άκρα. Ο μετασχηματιστής HX711 το κάνει αυτό μέσω των πηνίων E+ και E-. Επίσης πρέπει να γίνει μέτρηση της διαφοράς τάσης αναμεσά στα άκρα C και D, αυτό πραγματοποιείται μέσω των πηνίων A- και A+ του μετασχηματιστή.

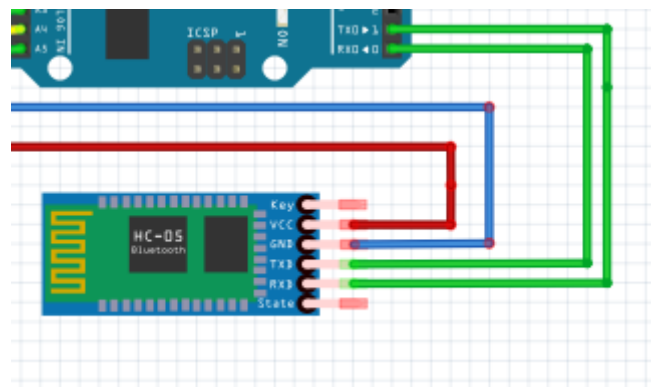
E+ -> A

E- -> B

A- -> C

A+ -> D

Συνδεσμολογία Bluetooth:

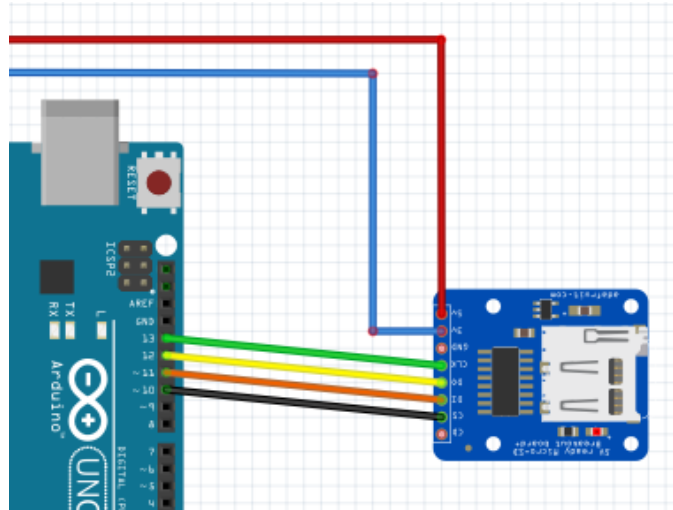


Σχήμα 3.4 συνδεσμολογία Bluetooth module.

Bluetooth pin	Arduino pin
TX	TX
RX	RX

Πίνακας 4 : Συνδεσμολογία Arduino – bluetooth.

Συνδεσμολογία SDcard module:



Σχήμα 3.5 συνδεσμολογία SDcard module.

SDcard module	Arduino
5V	5V
GND	GND
CLK	13
DO	12
DI	11
CS	10

Πίνακας 5 : Συνδεσμολογία Arduino – SDcard module.

3.2 ΜΑΘΗΜΑΤΙΚΟ ΜΟΝΤΕΛΟ ΣΥΣΤΗΜΑΤΟΣ

Μέσω του ακόλουθου μαθηματικού μοντέλου συνδέεται η μέτρηση που παίρνει ο μετατροπέας με τη τιμή της παραμόρφωσης.

Αρχικά ο συντελεστής παραμόρφωσης υπολογίζεται με τον τύπο 3.6:

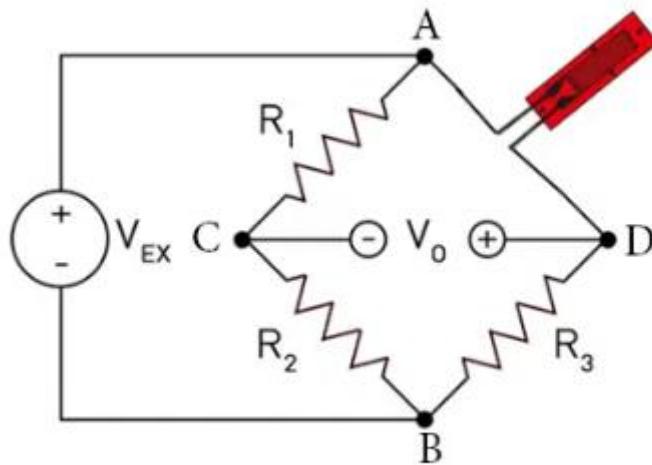
$$GF = \frac{\frac{\Delta R}{R}}{\frac{\Delta L}{L}} = \frac{\Delta R/R}{\varepsilon}$$

(3,6)

ΔR	αλλαγή στην αντίσταση του αισθητήρα λόγω παραμόρφωσης του δοκιμίου
R	αρχική αντίσταση αισθητήρα
ΔL	απολυτή αλλαγή μήκους δοκιμίου
L	Αρχικό μήκος δοκιμίου
ε	παραμόρφωση
GF	Συντελεστής παραμόρφωσης

Πίνακας 6 : Ανάλυση τύπου συντελεστή παραμόρφωσης.

Στον παραπάνω τύπο ο συντελεστής παραμόρφωσης και η παραμόρφωση (ε & GF) δεν έχουν μονάδες μέτρησης εφόσον όπως φαίνεται και από τον τύπο γίνεται απαλοιφή των μονάδων της αντίστασης και του μήκους.



Σχήμα 3.6 γέφυρα Wheatstone.

Για την γέφυρα Wheatstone (Σχήμα 3.6) της παρούσας εφαρμογής ισχύουν τα εξής:

- Όλες οι αντιστάσεις είναι ίσες με 120Ω .
- $V_0 = 5V$ (οπού V_0 διαφορά τάσης αναμεσα από σημεία C & D).
- Ο αισθητήρας έχει αντίσταση 120Ω σε κατάσταση ηρεμίας.

- GF = 2 δεδομένο κατασκευαστή του αισθητήρα.

*Για την επίλυση του μοντέλου ο αισθητήρας θα αναφέρεται ως R4

Για το κύκλωμα της γέφυρας ισχύει:

$$V_O = \frac{V_S \cdot (R_4 \cdot R_1 - R_3 \cdot R_2)}{((R_2 + R_4) \cdot (R_1 + R_3))} \quad (1)$$

Μετά τη παραμόρφωση ισχύει: $R_4' = R_4 + \Delta R_4$ (2)

$$(1), (2) \Rightarrow V_O = \frac{V_S \cdot ((R_4 + \Delta R_4) \cdot R_1 - R_3 \cdot R_2)}{((R_2 + R_4 + \Delta R_4) \cdot (R_1 + R_3))} \quad (3)$$

$$\Leftrightarrow V_O = \frac{V_S \cdot \Delta R_4 \cdot R_1}{((R_2 + R_4 + \Delta R_4) \cdot (R_1 + R_3))}$$

Εφόσον η διάφορες στην αντίσταση του αισθητήρα είναι πάρα πολύ μικρές ισχύει ότι: $\Delta R_4 \ll R_4$ και ως αποτέλεσμα ισχύει : $R_4 + \Delta R_4 = R_4$ (2)

$$\Rightarrow V_O = \frac{V_S \cdot \Delta R_4 \cdot R_1}{((R_2 + R_4) \cdot (R_1 + R_3))}$$

Από τον τύπο του συντελεστή παραμόρφωσης μπορούμε να καταλήξουμε στην ακόλουθη σχέση:

$$GF = \frac{\frac{\Delta R}{R}}{\frac{\Delta L}{L}} = \frac{\Delta R}{R} \cdot \frac{L}{\Delta L} = \frac{\Delta R}{R} \cdot \frac{1}{\varepsilon}$$

$$\Leftrightarrow \Delta R = GF \cdot \varepsilon \cdot R_4$$

$$\Leftrightarrow \varepsilon = \frac{(V_O \cdot (R_2 + R_4)^2)}{(V_S \cdot GF \cdot R_2 \cdot R_4)} \quad (4)$$

Έτσι καταλήγουμε στη γενική μορφή της εξίσωσης παραμόρφωσης:

$$\Rightarrow \varepsilon = \frac{4V_o}{(GF * V_s)}$$

*με δεδομένο ότι: $GF = 2$

$$\Rightarrow \varepsilon = \frac{4V_o}{2 * V_s} (5)$$

Απλοποιημένη μορφή της εξίσωσης παραμόρφωσης:

$$\Rightarrow \varepsilon = \frac{2V_o}{V_s}$$

Η εξίσωση αυτή προκύπτει θεωρώντας ότι όλες οι αντιστάσεις έχουν αντίσταση ίση με 120 Ohm άλλα στο πραγματικό κόσμο είναι σχεδόν αδύνατον να επιτευχθεί αυτό, για αυτό και στόχος είναι η χρήση αντιστάσεων μεγάλης ακριβείας εφόσον οι αποκλήσεις των αντιστάσεων απο τις θεωρητικές τιμές οδηγούν σε μετρήσεις με μεγάλο σφάλμα. Αυτό φαίνεται στο παρακάτω παράδειγμα:

$R_1 = 121\text{Ohm}$, $R_2 = 119\text{Ohm}$, $R_3 = 120.5\text{Ohm}$, $R_4 = 120\text{Ohm}$, $DR_4 = 0.002\text{ Ohm}$,
 $V_s = 5\text{V}$

Από τη σχέση 3 προκύπτει:

$$V_o = \frac{(5 * ((120 + 0.002) * 121 - 120.5 * 119))}{((119 + 120 + 0.002) * (121 + 120.5))}$$

$$\Rightarrow V_o = 0.0156\text{V}$$

Από τη σχέση 3 προκύπτει:

$$\varepsilon = \frac{(0.0156 * (119 + 120)^2)}{5 * 2 * 119 * 120}$$

$$\varepsilon = 0.00624 \text{ strain}$$

Αντίστοιχα για ιδανικές αντιστάσεις:

$R_1=120\Omega, R_2=120\Omega, R_3=120\Omega, R_4=120\Omega, DR_4=0.002 \Omega, V_s=5V$

$$\Rightarrow V_o = 0.0000208V$$

$$\varepsilon = 0.00000832 \text{ strain}$$

Έτσι προκύπτει ότι ενώ η απόκλιση των αντιστάσεων είναι μικρότερη από 1% η τελική μέτρηση που παίρνουμε έχει αποχή 74900% από τη θεωρητική τιμή. Ως αποτέλεσμα είναι σημαντικό οι αντιστάσεις που χρησιμοποιούνται να είναι όσο το δυνατόν πιο κοντά στη θεωρητική τιμή. Γενικά για τέτοιες εφαρμογές αποδέκτες είναι οι αντιστάσεις με ανοχή μικρότερη ή ίση με 0.1%. Κατά τη διαμόρφωση του μηχανήματος χρησιμοποιήθηκαν αντιστάσεις με ανοχή 0.1% προκειμένου να επιτευχθεί όσο τον δυνατόν μεγαλύτερη ακρίβεια.

Στη μορφή αυτή της εξίσωσης έχουμε μόνο έναν άγνωστο συντελεστή το V_o ο οποίος είναι και η μέτρηση την οποία παίρνουμε. Αφού αποκτήσουμε τη μέτρηση της παραμόρφωσης μπορούμε επίσης να υπολογίσουμε και την επιμήκυνση της επιφάνειας του δοκιμίου που μετρήσαμε.

$$\Delta L = \varepsilon \cdot L$$

Οι τιμές τις οποίες παρουσιάζει η γέφυρα Wheatstone κυμαίνονται από +3.6μV έως και - 3.6μV δίνοντας μας ένα εύρος τιμών ίσο με 7.214μV. Ο μετατροπέας HX711 έχει διακριτική ικανότητα 24 μπιτ πράγμα το οποίο σημαίνει ότι μπορεί να μετρήσει $2^{24} = 16777215$ διαφορετικές τιμές.

$$7.214/16777215 = 0.43 \text{ nano Volt}$$

Ο μετατροπέας HX711 παίρνει μόνο θετικές τιμές κοινώς το μηδέν βρίσκεται στο μέσον του εύρους τιμών το οποίο μπορεί να μετρήσει αρά οι αρνητικές τιμές έχουν ένα εύρος τιμών από 0 έως 8388607 και για θετικές τιμές από 8388607 έως 16777215. Αρά όταν δεν υπάρχει καθόλου παραμόρφωση ο μετασχηματιστής θα στέλνει την τιμή 8388607 + το σφάλμα μόνιμης κατάστασης. Το σφάλμα μόνιμης κατάστασης υπολογίζεται ως ο μέσος όρος 100 μετρήσεων, οι μετρήσεις αυτές γίνονται όταν δεν ασκούνται δυνάμεις άρα δεν υπάρχει παραμόρφωση. Ως αποτέλεσμα αφού γίνει η διαδικασία υπολογισμού μόνιμου σφάλματος, το μόνιμο σφάλμα αφαιρείται από τις μετρήσεις προκειμένου να υπολογιστεί η πραγματική τιμή.

Στην περίπτωση που ασκείται μια δύναμη στο δοκίμιο η οποία μας δίνει τιμή ίση με $V_0 = 2mV$ και το σφάλμα μόνιμης κατάστασης είναι μηδέν από τον ισχύει ότι:

$$\frac{2mV}{0.43nV} = 4651160$$

$$4651160 + 8388607 = 13039767$$

Με αποτέλεσμα το Arduino να δέχεται τη τιμή 13039767 και η τάση V_0 προκύπτει ως εξής:

$$V_0 = ((13039767 - (8388607 + \text{σφάλμα μόνιμης κατάστασης}) \cdot 0.43nV) = 2mV$$

Με αποτέλεσμα να μπορούμε να υπολογίσουμε τη παραμόρφωση:

$$\varepsilon = \frac{2 \cdot (2mV)}{3.77V} = 0.0010608$$

$$0.0010608 \text{ strain} = 1060.8\mu\text{Strain}$$

3.3 ΑΝΑΛΥΣΗ ΤΟΥ ΚΩΔΙΚΑ ΠΟΥ ΑΝΑΠΤΥΧΘΗΚΕ

Ο μικροελεγκτής Arduino UNO προγραμματίστηκε με τη χρήση του προγραμματιστικού περιβάλλοντος Arduino IDE. Το προγραμματιστικό περιβάλλον αυτό επιτρέπει τον προγραμματισμό του Arduino με τη χρήση της C++ η οποία έχει δεχτεί κάποιες μικρές τροποποιήσεις προκειμένου να ικανοποιεί καλύτερα τις απαιτήσεις του Arduino, επίσης μας επιτρέπει να ελέγξουμε τον κώδικα και να τον φορτώσουμε στο Arduino. Ακολουθεί η αναλυτική ανάλυση του κώδικα που φορτώθηκε στο Arduino.

Αρχικά έγινε η φόρτωση των απαιτούμενων βιβλιοθηκών:

```
#include <SD.h> //SD CARD BREAKOUT BOARD LIBRARY
```

Η διαδικασία αυτή γίνεται στην αρχή εφόσον για να χρησιμοποιηθούν οι συναρτήσεις της βιβλιοθήκης πρέπει να έχει προηγηθεί η φόρτωση της.

Δήλωση μεταβλητών:

```
//VARIABLE DECLARATIONS
```

```
long sample = 0 ; //TEMPORARY STORAGE VALUE FOR CALIBRARION
```

```

long sample_temp; //TEMPORARY VALUE THAT STORES
int x; //VALUE REPRESENTING THE HX USED (1,2,3)
int e1,e2,e3; //variables that represent the 3 strain outputs
long count = 0; // Instant Reading value
long sample1 = 0; // Sample for calibrating
long sample2 = 0; // Sample for calibrating
long sample3 = 0; // Sample for calibrating

unsigned long time_from_reading_start; //time point that the first recorded reading
//took place
unsigned long general_time;
char Incoming_value = 0; //VALUE INPUTED TO THE ARDUINO FROM
//PHONE, RESPONSIBLE FOR WITING SO SD CARD AND RECALIBRATING THE
DEVICE
int memory_controller; //VARIABLE USED TO STORE THE VALUE THAT //IS
INPUTED FROM THE PHONE APPLICATION
int readings_counter; //USED TO STORE THE AMOUNT OF TIMES WE
//WROTE DATA TO THR SD CARD
File myFile; //NAME OF THE SD CARD FILE
byte DT1=14; //DATA
byte SCK1=15; //SPI Clock
byte DT2=16; //
byte SCK2=17; //
byte DT3=18; //
byte SCK3=19; //

```

Οι παραπάνω μεταβλητές είναι αυτές που θα χρησιμοποιηθούν μέσα στο πρόγραμμα και θα συγκρατούν όλα τα δεδομένα που χρειάζονται προκειμένου να λειτουργήσει η εφαρμογή.

Ανάγνωση δεδομένων από το HX711:

```

##### READING ONCE FROM HX711#####
unsigned long readrealcount(int x){
    ### SELECTION OF INPUT PINS BASED ON FUNCTION INPUT

```

```

if(x==1){  /// FIRST HX711
  SCK1=15;
  DT1=14;
}
if(x==2){  /// SECOND HX711
  SCK1=SCK2;
  DT1=DT2;
}
if(x==3){  /// THIRD HX711
  SCK1=SCK3;
  DT1=DT3;
}

/*pulse reading function to avoid using the standart hx711 library
* as we want to get the voltage diffrence in order to calculate strain
* and not the weight */

unsigned long Count;
unsigned char i;
pinMode(DT1, OUTPUT);  //SETTING DT PIN TO OUTPUT MODE
digitalWrite(DT1,HIGH); //SETTING DT HIGH
digitalWrite(SCK1,LOW); //SETTING SCK TO LOW
Count=0;
pinMode(DT1, INPUT);
while(digitalRead(DT1)); //WAIT UNTILL DT(DATA LINE) GOES LOW
for (i=0;i<24;i++)  //PULSE THE CLOCK PIN(SCK) 24 TIMES TO READ THE
//DATA(24BIT SIGNED INTEGER) FROM HX711
{
  digitalWrite(SCK1,HIGH);
  Count=Count<<1;
  digitalWrite(SCK1,LOW);
  if(digitalRead(DT1))

```

```

Count++;
}
digitalWrite(SCK1,HIGH);
Count=Count^0x800000;
digitalWrite(SCK1,LOW);
return(Count);      //RETURN THE FINAL READING
}

```

Η συνάρτηση αυτή παίρνει ως είσοδο τον αριθμό του HX711 από το οποίο θέλουμε να πάρουμε μέτρηση. Στη συνέχεια με βάση αυτή την είσοδο προσδιορίζει τα pins από τα οποία θα πάρει την μέτρηση. Έπειτα ακολουθεί μια συνάρτηση παλμού η οποία μας επιστρέφει την τελική μέτρηση.

Συνάρτηση Setup:

```

void setup()
{
readings_counter = 0;      //NUMBER OF TIMES WE WROTE OUR READINGS
//TO THE SD CARD IS SET TO 0

Serial.begin(9600);        //SERIAL PORT INITIALIZATION

pinMode(10, OUTPUT);      //SETTING PIN 10 AS OUTPUT IN ORDER TO SET
//UP THE SD CARD BREAKOUT BOARD

pinMode(15, OUTPUT);      // CLOCK PIN HX711(1ST)
pinMode(17, OUTPUT);      // CLOCK PIN HX711(2ND)
pinMode(19, OUTPUT);      // CLOCK PIN HX711(3RD)

```

```

##### CHECK IF SD CARD HAS BEEN INITIALIZED #####
if (!SD.begin(10)) {
  Serial.print("initialization failed!");
  Serial.print("|");
  Serial.print("initialization failed!");
  Serial.print("|");

```

```

Serial.print("initialization failed!");
Serial.print("|");
return;
}
Serial.print("initialization of sd done");
Serial.print("|");
Serial.print("initialization of sd done");
Serial.print("|");
Serial.print("initialization of sd done");
Serial.print("|");
//
myFile = SD.open("test.txt", FILE_WRITE); // OPEN THE SD CARD FILE
//

##### CHECK THAT THE OUR FILE OPENED IN THE SD CARD #####
if (myFile) {
  Serial.print("Ready to record");
  Serial.print("|");
  Serial.print("Ready to record");
  Serial.print("|");
  Serial.print("Ready to record");
  Serial.print("|");
} else {
  Serial.println("unable to record");
  Serial.print("|");
  Serial.println("unable to record");
  Serial.print("|");
  Serial.println("unable to record");
  Serial.print("|");
}

```

```

##### CALIBRATION SEQUENCE FOR ALL THE HX711 #####
//the " " and the "|" are needed in order to print them in order and have them
//centered in the app
Serial.print("calibrating 1");
Serial.println(" ");
Serial.println(" ");
Serial.print("|");
Serial.print("calibrating 2");
Serial.println(" ");
Serial.println(" ");
Serial.print("|");
Serial.print("calibrating 3");
Serial.println(" ");
Serial.println(" ");
Serial.print("|");
delay(500);

Incoming_value = 0; //THIS VALUE IS SET TO ZERO SO THE DEVICE
//STARTS OFF WITHOUT WRITING TO THE SD CARD
sample1 = calibration(1); // calibration of the 1ST hx711
sample2 = calibration(2); // calibration of the 2ND hx711
sample3 = calibration(3); // calibration of the 3RD hx711
}

```

Η συνάρτηση setup τρέχει μόνο μια φορά, ο στόχος της είναι να κάνει όλες τις διαδικασίες που θέλουμε να κάνουμε μόνο μια φορά πριν τρέξουμε το κύριο πρόγραμμα. Αρχικά γίνεται ενεργοποίηση των σειριακών θηρών οι οποίες είναι υπεύθυνες για την αποστολή δεδομένων. Στη συνέχεια θέτουμε τα pin 10,15,17,19 ως εξόδους. Μετά γίνεται έλεγχος του SDcard module, πιο συγκεκριμένα ελέγχεται εάν ανοίγει, αφού εγκριθεί ότι άνοιξε επιτυχώς η κάρτα SD δημιουργείται ένα αρχείο .txt το οποίο στη συνέχεια περνάει έλεγχο για να βεβαιωθούμε πως άνοιξε, το αρχείο αυτό

είναι εκείνο στο οποίο θα αποθηκεύονται οι μετρήσεις. Τέλος γίνεται έλεγχος μόνιμου σφάλματος για κάθε μετασχηματιστή HX711.

Κυρίος κώδικας (Main loop):

```
void loop()
{
myFile = SD.open("test.txt", FILE_WRITE); //OPENING THE TEXT FILE THAT WILL
//STORE OUR READINGS

Serial.setTimeout(100);           //THE TIMEOUT FUNCTIONS STOPS WAITING
//FOR AN INPUT AFTER A SET AMOUNT OF TIME

                                   //IN OUR CASE WE NEED IT BECAUSE WE DONT
//HAVE A CONSTANT INPUT STRAM FROM THE USER

//### TAKING A READING FROM EVERY HX711 ###
e1 = reading(1);
delay(50);
e2 = reading(2);
delay(50);
e3 = reading(3);
delay(50);
//
Incoming_value = Serial.parseInt(); //READING VALUE FROM PHONE(WAITS
//FOR 100 MS)
```

```
// the incoming value resets to zero after every loop because if the phone doesnt
//input anything it takes it as a zero

// so we save the value of the input in a variable:

// THE INCOMING VALUE RESETS TO ZERO IF THE IS NO INPUT FROM THE
//PHONE

// AS A RESULT WE NEED TO SAVE THE INCOMING VALUE INTO A SEPARATE
//VALUE
```

```

if(Incoming_value == 1) {          //RECALIBRATION SEZUENCE
    memory_controller=1;          //SAVING THE INCOMING VALUE SO IT
//DOESNT RESET TO 0 IN THE NEXT LOOP
}
else if(Incoming_value == 2){      //START WRITING TO SD CARD
    time_from_reading_start = millis(); //saves time of start to a value
    readings_counter+=1;          //INCREASE THE NUMBER OF READING
//CYCLES BY ONE
    memory_controller=2;          //SAVING THE INCOMING VALUE SO IT
//DOESNT RESET TO 0 IN THE NEXT LOOP
    myFile.print("set of readings: "); //WRITING TO THE SD CARD THE NUM OF
//THE CURRENT READING SYCLE
    myFile.println(readings_counter);
}
else if(Incoming_value == 3){      //STOP WRITING TO THE SD CARD
    memory_controller=3;          //SAVING THE INCOMING VALUE SO IT
//DOESNT RESET TO 0 IN THE NEXT LOOP
}

##### READ/WRITE TO SD AND RECALIBRATE #####

if(memory_controller == 2){        //PRINT TO SD CARD

    general_time = millis()-time_from_reading_start; //SAVING THE TIME OF THE
//READING
    myFile.print(general_time/1000);          //RESETING THE TIME FROM
//SECONDS TO MS
    myFile.print(" ");                      //WRITING THE READINGS
    myFile.print(e1);
    myFile.print(" ");
    myFile.print(e2);
    myFile.print(" ");

```

```

myFile.print(e3);
myFile.println(" ");
}else if(memory_controller == 3){           //STOP PRINTING READINGS(DO
//NOTHING IF INPUTEED VAL == 3)
}else if(memory_controller == 1){         //START RECALIBRATION
//SEQUENCE
    delay(100);
    Serial.print("calibrating 1");        //PRINT TO THE PHONE THAT THE
//DEVICE IS CALIBRATING
    Serial.println(" ");
    Serial.println(" ");
    Serial.print("|");
    Serial.print("calibrating 2");
    Serial.println(" ");
    Serial.println(" ");
    Serial.print("|");
    Serial.print("calibrating 3");
    Serial.println(" ");
    Serial.println(" ");
    Serial.print("|");
    sample1 = calibration(1);             //CALIBRATION OF 1ST HX711
    sample2 = calibration(2);             //CALIBRATION OF 2ND HX711
    sample3 = calibration(3);             //CALIBRATION OF 1RD HX711
    delay(100);
    memory_controller = 3;                //stopping the recordings on the sd card
//((DO NOTHING)
}

#####PRINTING TO PHONE###

##### PRINTING VALUES OF THE FIRST GAGE ###
Serial.print("1: ");

```

```

Serial.print(int(e1));
Serial.println(" ");
Serial.println(" ");
Serial.print("|");

##### PRINTING VALUES OF THE SECOND GAGE ###
Serial.print("2: ");
Serial.print(int(e2));
Serial.println(" ");
Serial.println(" ");
Serial.print("|");

##### PRINTING VALUES OF THE THIRD GAGE ###
Serial.print("3: ");
Serial.print(int(e3));
Serial.println(" ");
Serial.println(" ");
Serial.print("|");
delay(500);

myFile.close(); //CLOSING THE SD CARD FILE
}

```

Ο παραπάνω κώδικας αποτελεί το κύριο μέρος του προγράμματος. Αρχικά ανοίγει το αρχείο .txt που δημιουργήθηκε πριν. Μετά με τη εντολή: `Serial.setTimeout(100);` θέτουμε ένα όριο στον χρόνο για τον οποίο ο κώδικας θα περιμένει να πάρει είσοδο. Αμέσως μετά γίνονται οι 3 μετρήσεις και αποθηκεύονται σε μεταβλητές. Στη συνέχεια το πρόγραμμα δέχεται εντολή από το τηλέφωνο με τη μορφή ενός αριθμού, στη περίπτωση που ο χρήστης δε πατήσει τίποτα η μεταβλητή αυτόματος γίνεται μηδέν. Ανάλογα με την είσοδο που δέχτηκε το πρόγραμμα υπάρχουν τρεις επιλογές για το τι θα γίνει. Η πρώτη κάνει πάλι τη διαδικασία ελέγχου μονίμου σφάλματος. Η δεύτερη ξεκινάει την αποθήκευση δεδομένων στη κάρτα SD.

Ενώ η τρίτη σταματάει την αποθήκευση δεδομένων στη κάρτα. Τέλος γίνεται η αποστολή των μετρήσεων που έγιναν μέσω bluetooth προκειμένου να ληφθούν από τη συσκευή του χρήστη και κλείνει το αρχείο .txt που είχε ανοιχθεί στην αρχή.

Επεξεργασία μέτρησης:

```
unsigned long reading(int x){ //function for taking a reading from the HX711
  if(x==1){sample_temp = sample1;
}else if(x==2){sample_temp = sample2;
}else if(x==3){sample_temp = sample3;}

  count = readrealcount(x); // SAVING THE READING VALUE INTO A VARIABLE

  long val=(count-sample_temp); // DELETING THE ERROR THAT WE DETECTED
//IN THE CALIBRATION SEQUENCE

  float vout=(val*0.43); // CONVRETING THE VAKUE INTO VOLTS (in nVolts )
//1=0.43nV

  int e=4*(vout/3.77)*0.002; // CONVERSION TO MICROSTRAIN

  return e; // RETURNING THE MICROSTRAIN
}
```

Η συνάρτηση αυτή παίρνει την ακατέργαστη μέτρηση και μέσω του μαθηματικού μοντέλου που αναλύθηκε προηγούμενος επιστρέφει την τελική παραμόρφωση.

Αφαίρεση μονίμου σφάλματος:

```
unsigned long calibration(int x){ //CALIBRAION FUNCTION

  sample = 0; //RSET TEMPORARY DATA STORAGE VALUE TO
//ZERO

  for(int i=0;i<100;i++) //SAVING 100 READINGS INTO THE TEMPORARY
//VALUE
  {
    count = readrealcount(x);

    sample+=count;

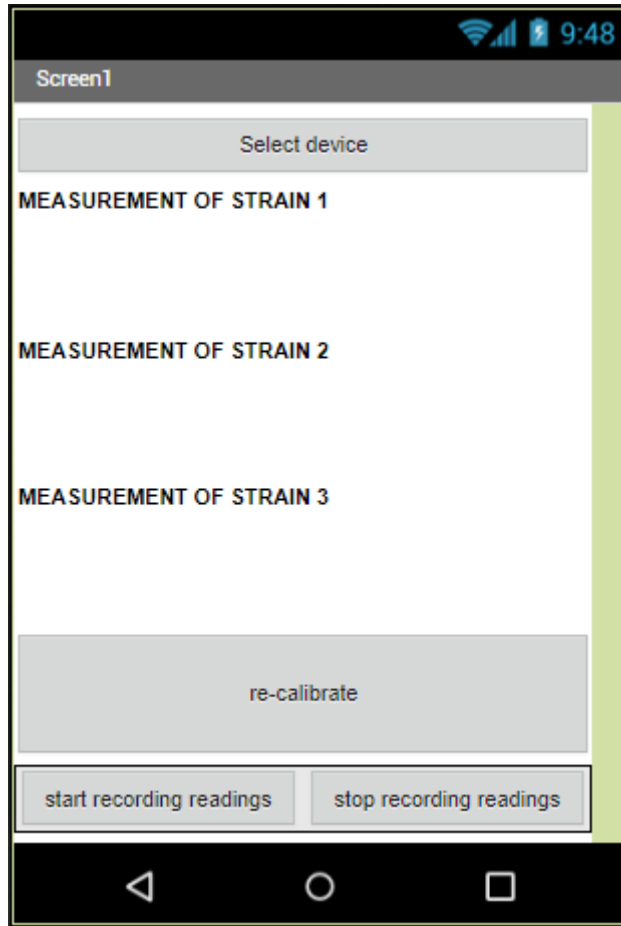
  }
}
```

```
sample/=100;           //AVERAGING THEM OUT
count = 0;             //RESETTING THE VALUES TO ZERO
return sample;
}
```

Αυτό που επιτυγχάνεται με τη παραπάνω συνάρτηση είναι η αφαίρεση του σφάλματος. Πιο συγκεκριμένα η συνάρτηση παίρνει 100 μετρήσεις από τον μετασχηματιστή και βγάζει τον μέσον ορό αυτών καταλήγοντας έτσι σε ένα σφάλμα μόνιμης κατάστασης.

3.4 ΑΝΑΛΥΣΗ ΤΟΥ ΚΩΔΙΚΑ ΤΗΣ ΕΦΑΡΜΟΓΗΣ ΤΟΥ ΤΗΛΕΦΩΝΟΥ

Όπως αναφέρθηκε και πριν προκειμένου να γίνει ανάγνωση δεδομένων και αποστολή εντολών από το χρήστη στη συσκευή χρησιμοποιείται μια εφαρμογή που διαμορφώθηκε ειδικά για την εργασία αυτή. Η εφαρμογή διαμορφώθηκε με χρήση της online πλατφόρμας MITappinventor και μπορεί να χρησιμοποιηθεί μόνο από συσκευές android. Ο προγραμματισμός της εφαρμογής έγινε με μπλοκ εντολών.



Σχήμα 3,7 Εφαρμογή smartphone που προγραμματιστικέ.

Αρχικά η εφαρμογή έχει προγραμματιστεί να περιμένει τον χρήστη να συνδεθεί με την συσκευή μέσω bluetooth (Σχήμα 3,7). Στη συνέχεια έχει ανακοινωθεί μια λίστα από εισόδους οι οποίες θα εμφανίζονται στην οθόνη, οι εισοδοι αυτοί είναι οι μετρήσεις παραμόρφωσης.

```

when ListPicker1 .BeforePicking
do set ListPicker1 . Elements to BluetoothClient1 . AddressesAndNames

when ListPicker1 .AfterPicking
do set ListPicker1 . Selection to call BluetoothClient1 .Connect
address ListPicker1 . Selection

if BluetoothClient1 . IsConnected
then
set Clock1 . TimerAlwaysFires to true
set ListPicker1 . Visible to false
set MEASUREMENT_1 . Visible to true
set MEASUREMENT2 . Visible to true
set MEASUREMENT_3 . Visible to true
set M1 . Visible to true
set M2 . Visible to true
set M3 . Visible to true

initialize global LIST to create empty list
initialize global INPUT to ""

```

Εφόσον έχει γίνει η σύνδεση με το μηχάνημα η εφαρμογή αρχίζει και δέχεται μηνύματα. Τα μηνύματα αυτά προκειμένου να διαχωριστούν πρέπει να τεθεί ένας ειδικός χαρακτήρας διαχωρισμού, επιλέχθηκε ο χαρακτήρας “|” με αποτέλεσμα να χρησιμοποιούνται αυτά τα διαχωριστικά μέσα στον κώδικα ενώ στέλνουμε τα μηνύματα προκειμένου να διαχωριστούν οι μετρήσεις στην εφαρμογή.

```
when Clock1.Timer
do
  if BluetoothClient1.IsConnected
  then
    if call BluetoothClient1.BytesAvailableToReceive > 0
    then
      set global INPUT to call BluetoothClient1.ReceiveText
      set global LIST to split text get global INPUT
      at |
      set M1.Text to select list item list get global LIST
      index 1
      set M2.Text to select list item list get global LIST
      index 2
      set M3.Text to select list item list get global LIST
      index 3
      set global INPUT to ""
      set global LIST to create empty list
```

Τέλος δίνονται τιμές στα κουμπιά της εφαρμογής. Αυτές οι τιμές είναι υπεύθυνες για τις λειτουργίες του μηχανήματος.

```
when Button1.Click
do
  call BluetoothClient1.SendText
  text " 1 "

when Button2.Click
do
  call BluetoothClient1.SendText
  text " 2 "

when Button3.Click
do
  call BluetoothClient1.SendText
  text " 3 "
```


ΚΕΦΑΛΑΙΟ 4 – ΠΕΙΡΑΜΑΤΙΚΗ ΕΦΑΡΜΟΓΗ

4.1 ΠΡΑΚΤΙΚΗ ΕΦΑΡΜΟΓΗ ΤΟΥ ΜΕΤΡΗΤΙΚΟΥ ΟΡΓΑΝΟΥ

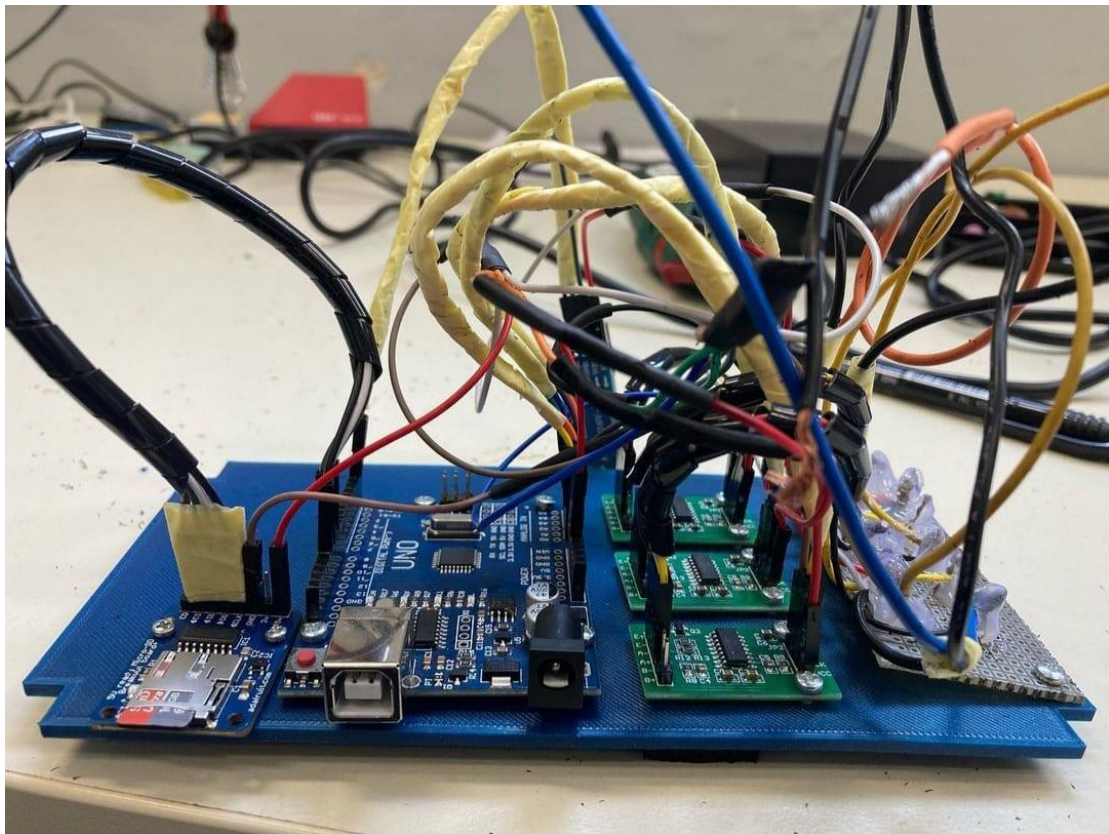
Το πειραματικό όργανο που αναπτύχθηκε (Σχήμα 4.1&4.2) είναι απλό και εύκολο στη χρήση του. Οι μοναδικές απαιτήσεις που υπάρχουν προκειμένου κάποιος να χρησιμοποιήσει τη πειραματική διάταξη είναι να υπάρχει διαθέσιμη παροχή ρεύματος και να έχει εγκαταστήσει την εφαρμογή που διαμορφώθηκε στο κινητό του τηλέφωνο. Προκειμένου κάποιος να χρησιμοποιήσει το μηχάνημα πρέπει να το συνδέσει στη παροχή ρεύματος και στη συνέχεια να συνδεθεί σε αυτό μέσω του smartphone του. Το μηχάνημα με το που συνδέεται σε παροχή ρεύματος ξεκινάει τη διαδικασία υπολογισμού του σφάλματος μόνιμης κατάστασης. Ενώ γίνεται αυτή η διαδικασία ο χρήστης βλέπει το μήνμα “calibrating” στη θέση των τριών μετρήσεων.



Σχήμα 4.1 Τελική μορφή της συσκευής.

Κατά τη διαδικασία αυτή το μηχάνημα παίρνει εκατό μετρήσεις από κάθε strain Gage και βγάζει τον μέσο ορό τους υπολογίζοντας έτσι το σφάλμα. Το σφάλμα αυτό στη πορεία αφαιρείται από τη μέτρηση που παίρνει το μηχάνημα προκειμένου να υπολογιστεί η πραγματική τιμή της μέτρησης. Η διαδικασία υπολογισμού του

σφάλματος κρατάει 5 δευτερόλεπτα. Μετά τη διαδικασία αυτή ξεκινάει η αποστολή μετρήσεων, αρχικά χωρίς φορτίο εάν έχει πραγματοποιηθεί σωστά η διαδικασία αφαίρεσης μόνιμου σφάλματος πρέπει όλες οι μετρήσεις να είναι μηδέν σε άλλη περίπτωση ο χρήστης πρέπει να πατήσει το κουμπί “recalibrate” στην εφαρμογή του προκειμένου να επαναληφθεί η διαδικασία αφαίρεσης μόνιμου σφάλματος, μετά από αυτό οι μετρήσεις εκτυπώνονται σωστά. Στη περίπτωση που ο χρήστης θέλει να πάρει λιγότερες από τρεις μετρήσεις θα βλέπει την ελεύθερη υποδοχή μόνιμος ως μηδέν. Στη περίπτωση που ο χρήστης θέλει να κάνει καταγραφή των μετρήσεων του, θα πρέπει να πατήσει το κουμπί “start recording reading” για να αρχίσει η διαδικασία καταγραφής, για να σταματήσει αντίστοιχα θα πρέπει να πατήσει “stop recording reading”



Σχήμα 4.2 τελική μορφή συσκευής μετά τη καλωδίωση.

Οι μετρήσει στη κάρτα αποθηκεύονται σε αρχείο .txt και έχουν την ακόλουθη μορφή:

```
set of readings: 1
```

```
1 0 1 0
```

4 0 0 0
5 -16 21 25
9 -31 35 33
12 -63 69 67
16 -63 70 68

Στη πρώτη στήλη εμφανίζεται το δευτερόλεπτο στο οποίο τυπώθηκε η μέτρηση, στη δεύτερη η πρώτη μέτρηση και στη τρίτη και τέταρτη οι υπόλοιπες μετρήσεις αντίστοιχα. Πριν αρχίσουν να καταγράφονται οι μετρήσεις αναγράφεται ο αριθμός του σετ των μετρήσεων.

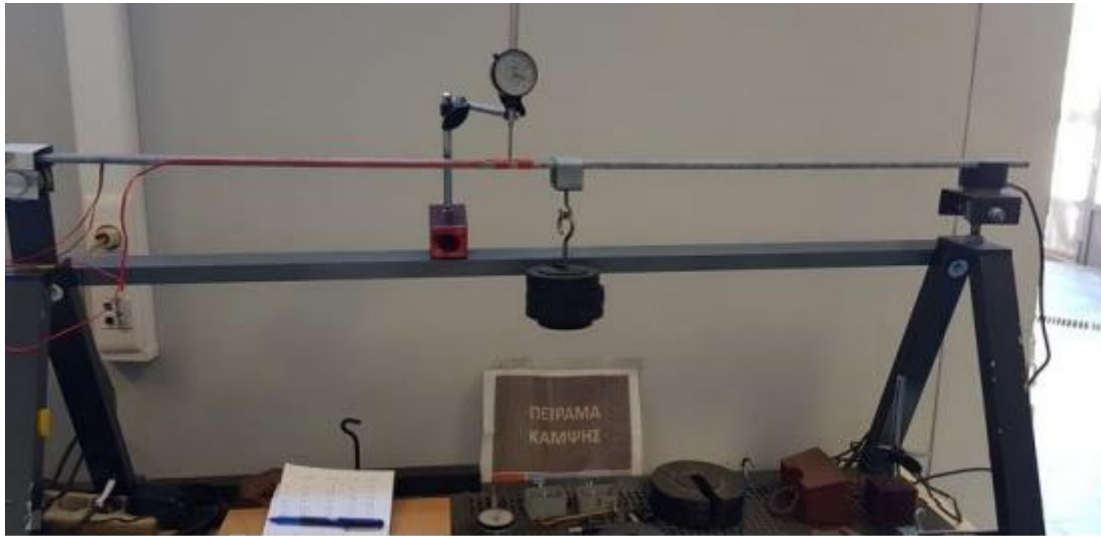
4.2 ΣΥΓΚΡΙΣΗ ΜΕΤΡΗΣΕΩΝ ΜΕ ΑΝΤΙΣΤΟΙΧΟ ΟΡΓΑΝΟ ΚΑΙ ΘΕΩΡΗΤΙΚΟ ΜΟΝΤΕΛΟ

Προκειμένου να επιβεβαιωθεί η σωστή λειτουργία του οργάνου πρέπει να περάσει από κάποιους ελέγχους. Για το όργανο αυτό έγιναν δυο έλεγχοι, ο πρώτος ήταν η σύγκριση των μετρήσεων με ένα όμοιο όργανο το οποίο διαμορφώθηκε από φοιτητή του τμήματος, ο δεύτερος είναι η σύγκριση των πειραματικών τιμών που προέκυψαν με τη θεωρητική τιμή που θα έπρεπε να έχουν οι τιμές.

Η συσκευή με την οποία έγινε η σύγκριση διαμορφώθηκε από φοιτητή του τμήματος. Η συσκευή αυτή έχει τη δυνατότητα μέτρησης ενός αισθητήρα και ανάπαιστοί τα δεδομένα σε οθόνη LCD. Οι δυο συσκευές χρησιμοποιούν το ίδιο μαθηματικό μοντέλο, οι μετρήσεις της ήδη υπάρχουσας συσκευής έχουν συγκριθεί με μηχάνημα που υπάρχει στην αγορά και όπως αποδείχθηκε έχει μεγάλη ακρίβεια, ως αποτέλεσμα εάν το μηχάνημα που διαμορφώθηκε έχει όμοιες μετρήσεις με το ήδη υπάρχον σημαίνει πως έχει μεγάλη ακρίβεια.

Προκειμένου να γίνει σύγκριση μεταξύ των δυο οργάνων πραγματοποιήθηκαν μετρήσεις και με τα δυο στην ίδια πειραματική διάταξη. Η πειραματική διάταξη αποτελείται από μια δοκό η οποία στηρίζεται και στα δυο άκρα της, ενώ στο μέσον της

έχουν προκληθεί ηλεκτροεπιμηκυνσιομέτρα, προκειμένου να παραμορφωθεί προθέτονται βαρίδια στο μέσον της.



Σχήμα 4.3 Πειραματική διάταξη.

Οι μετρήσεις πάρθηκαν στην πειραματική διάταξη της που φαίνεται στο σχήμα 4.3. Και για τα δυο μετρητικά ορίγανα έγινε η ίδια διαδικασία. Αρχικά συνδέθηκαν οι αισθητήρες στα όργανα, στη συνέχεια αφού το σύστημα ήταν σε κατάσταση ηρεμίας ενεργοποιήθηκαν τα μετρητικά προκειμένου να ξεκινήσει η διαδικασία αφαίρεσης μονίμου σφάλματος. Εφόσον είχε ολοκληρωθεί η διαδικασία υπολογισμού μονίμου σφάλματος και τα όργανα έδειχναν μηδενικές τιμές παραμόρφωσης ξεκίνησε η προσθήκη βαριδιών στη πειραματική διάταξη. Η προσθήκη έγινε ανά ένα κιλό τη φορά και οι μετρήσεις καταγράφονταν αφού τα όργανα σταθεροποιούνταν σε μια τιμή. Η προσθήκη βαριδιών έγινε με μορφή πυραμίδας ανά ένα κιλό ξεκινώντας από μηδέν κιλά και φτάνοντας στα τρία κιλά μέγιστο φορτίο με αποτέλεσμα να υπάρχουν επτά μετρήσεις για κάθε αισθητήρα.

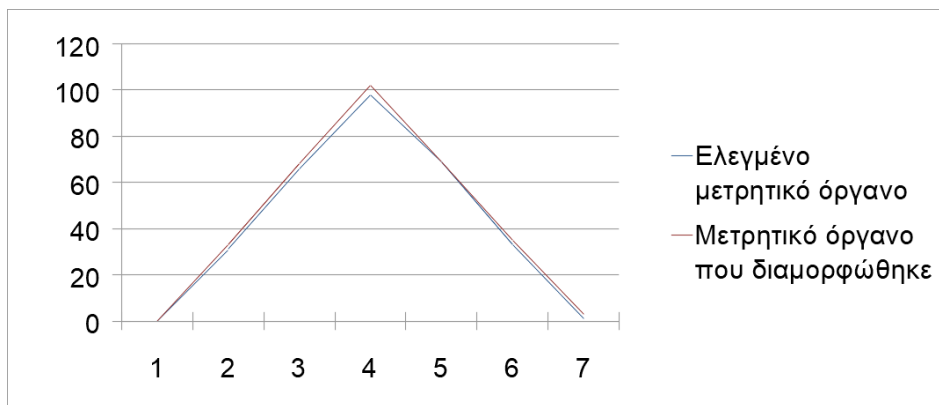
Ελεγμένο μετρητικό όργανο			
N (Newton)	GAGE A1	GAGE A2	GAGE A3
0	0	0	0
10	31	32	-31
20	66	69	-65
30	98	97	-100
20	69	67	-60
10	33	35	-26
0	1	2	-1

Πίνακας 7 : Μετρήσεις ελεγμένου μετρητικού οργάνου.

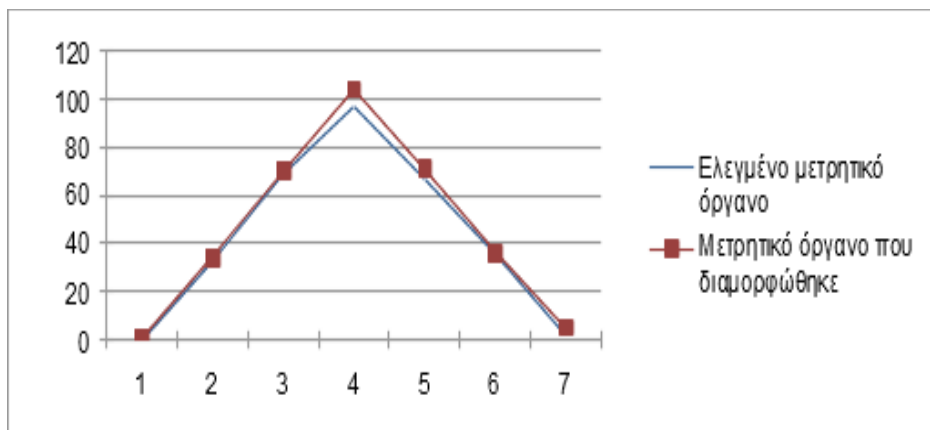
Μετρητικό όργανο που διαμορφώθηκε			
N (Newton)	GAGE A1	GAGE A2	GAGE A3
0	0	1	0
10	33	34	-31
20	68	70	-63
30	102	104	-95
20	69	71	-63
10	35	36	-31
0	3	5	3

Πίνακας 8 : Μετρήσεις διαμορφωμένου μετρητικού οργάνου.

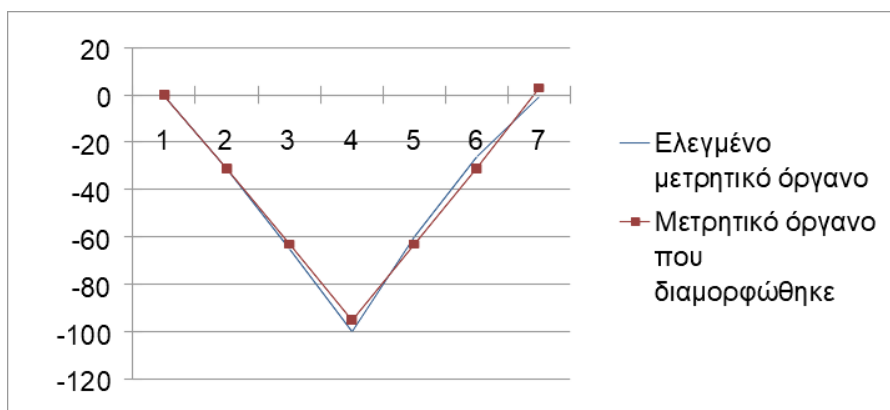
Σύγκριση μετρήσεων για τον αισθητήρα A1



Σύγκριση μετρήσεων για τον αισθητήρα A2



Σύγκριση μετρήσεων για τον αισθητήρα A3



Από τα παραπάνω διαγράμματα φαίνεται ότι τα αποτελέσματα των μετρήσεων των οργάνων είναι πολύ κοντά για όλες τις περιπτώσεις πράγμα το οποίο είναι ικανοποιητικό, εφόσον τώρα γνωρίζουμε πως το όργανο που διαμορφώθηκε είναι ικανό να δώσει μετρήσεις με μεγάλη ακρίβεια.

Διαφορά μετρήσεων (micro strain)

N (Newton)	GAGE 1	GAGE 2	GAGE 3
0	0	1	0
10	2	2	0
20	2	1	2
30	4	7	5
20	0	4	-3
10	2	1	-5
0	2	3	4

Πίνακας 9 : Διαφορά μετρήσεων ελεγμένου και διαμορφωμένου οργάνου.

Ποσοστιαία διαφορά

N (Newton)	GAGE 1	GAGE 2	GAGE 3
0	-	-	-
10	6.4	6.2	0
20	3	1.4	3
30	4	7.2	5
20	0	5.9	5
10	6	2.8	19.2
0	200	150	400

Πίνακας 10 : Ποσοστιαία διαφορά μετρήσεων ελεγμένου και διαμορφωμένου οργάνου.

Από τους παραπάνω πίνακες φαίνεται πως η απόκλιση αναμεσα στα δυο όργανα είναι σταθερή για τις περισσότερες μετρήσεις. Η μέση απόκλιση που παρατηρείται είναι ίση με $2.38 \text{ micro strain} = 0.00000238$ η οποία θεωρείται ικανοποιητική. Ποσοστιαία εάν εάν δεν ληφθούν υπόψιν οι τιμές για μηδενική φόρτιση φαίνεται ότι η διαφορά στις μετρήσεις είναι σταθερή και μικρή με μέση απόκλιση 3.9%

με αποτέλεσμα το όργανο να έχει ικανοποιητική ακρίβεια και σταθερότητα στις μετρήσεις του.

Η σύγκριση με το θεωρητικό μοντέλο πραγματοποιήθηκε συγκρίνοντας τις μετρήσεις που πάρθηκαν με τις θεωρητικές τιμές παραμόρφωσης τις οποίες θα έπρεπε να έχει πάρει το δοκίμιο.

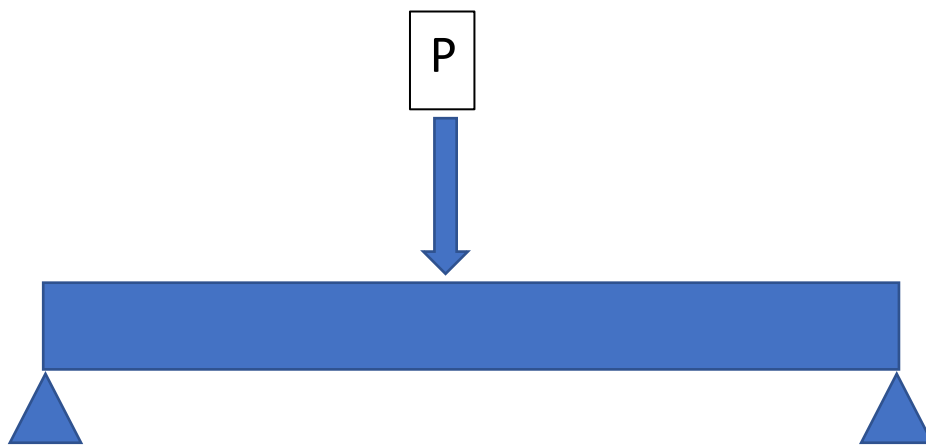
Επίλυση μαθηματικού μοντέλου:

Μήκος δοκιμιού: $L = 1372\text{mm}$

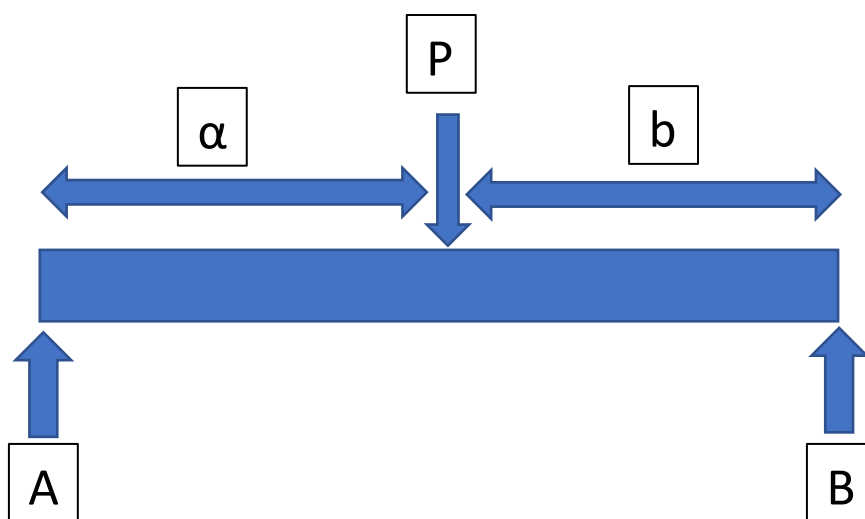
Πλάτος δοκιμιού: $b = 41\text{mm}$

Πάχος δοκιμιού: $h = 8.5\text{mm}$

Πειραματική διάταξη:



Διάγραμμα ελευθέρου σώματος:



$$Pa = B \cdot (a + b)$$

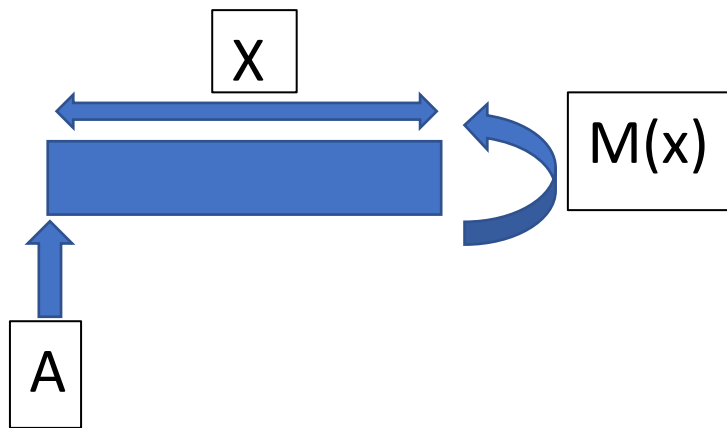
=>

$$B = \frac{pa}{a+b}$$

$$A = \frac{pb}{a+b}$$

Για τη ροπή αδράνειας ισχύει:

$$I = \frac{b \cdot h^3}{12} = \frac{41 \cdot 8.5^3}{12} = 2098 \text{mm}^4$$



$$M(x) = \frac{Pb}{a+b} \cdot x$$

$$a = 707 \text{mm}$$

$$b = 665 \text{mm}$$

$$x = 680 \text{mm}$$

$$\begin{aligned} M(680) &= P \cdot \frac{665}{1372} \cdot 680 \\ &= 329.6 \frac{P[N]}{\text{mm}} \end{aligned}$$

$$\sigma_{max} = -\frac{M(x)}{I} \cdot \frac{h}{2}$$

$$= -\frac{329.6 \cdot P}{2098} \cdot 4.25mm$$

$$\Rightarrow 0.668 \cdot P [MPa]$$

Έχοντας τα παραπάνω δεδομένα μπορούμε να υπολογίσουμε τη παραμόρφωση του δοκιμίου:

$$\varepsilon = \frac{0.668 \cdot P}{210 \cdot 10^3}$$

$$\Rightarrow \varepsilon = 3.18 \cdot 10^{-6} \cdot P$$

Αρά για $P = 10N$

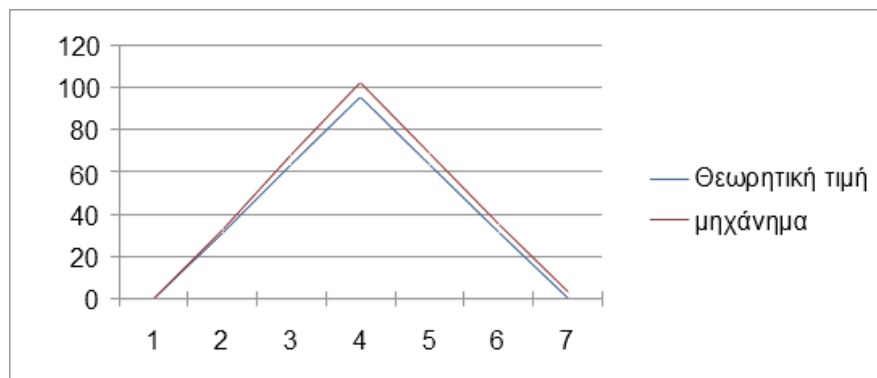
$$\Rightarrow \varepsilon = 3.18 \cdot 10^{-6} \cdot 10 = 31.8 \text{ micro strain}$$

Σύμφωνα με το θεωρητικό μοντέλο οι μετρήσεις θα έπρεπε να έχουν την εξής μορφή:

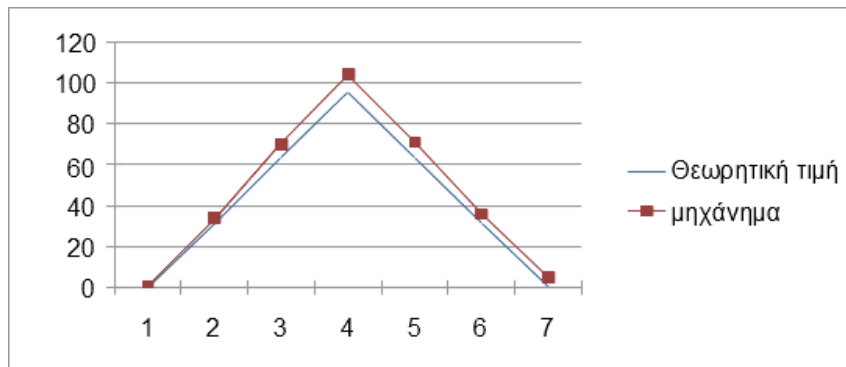
N	Micro strain
0	0
10	31.8
20	63.6
30	95.4
20	63.6
10	31.8
0	0

Πίνακας 11 : θεωρητική παραμόρφωση δοκιμίου.

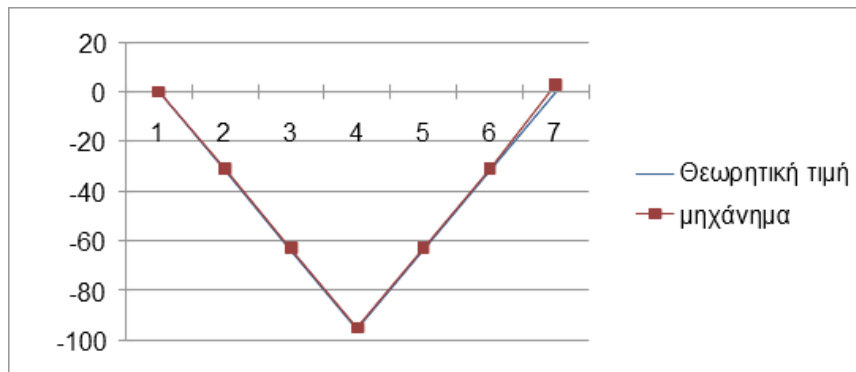
Gage 1



Gage 2



Gage 3



N	Gage 1	μηχάνημα	ds	Ποσοστό απόκλισης
0	0	0	0	0%
10	33	31.8	1.2	3.7%
20	68	63.6	4.4	6.9%
30	102	95.4	6.6	6.9%
20	69	63.6	5.4	8.4%
10	35	31.8	3.1	10%
0	3	0	3	-

Πίνακας 12 : αποτελέσματα σύγκρισης θεωρητικών τιμών με πειραματικές.

N	Gage 2	μηχάνημα	ds	Ποσοστό απόκλισης
0	1	0	1	-
10	34	31.8	2.2	6.9%
20	70	63.6	6.4	10%
30	104	95.4	8.6	9%
20	71	63,6	7.4	11.6%
10	36	31.8	4.2	13.2%
0	5	0	5	-

Πίνακας 13 : αποτελέσματα σύγκρισης θεωρητικών τιμών με πειραματικές.

N	Gage 3	μηχάνημα	ds	Ποσοστό απόκλισης
0	0	0	0	0%
10	-31	-31.8	0.8	2.5%
20	-63	-63.6	0.6	0.9%
30	-95	-95.4	0.4	0.4%
20	-63	-63.6	0.6	0.9%
10	-31	-31.8	0.8	2.5%
0	3	0	3	-

Πίνακας 14 : αποτελέσματα σύγκρισης θεωρητικών τιμών με πειραματικές.

Όπως φαίνεται και στα παραπάνω διαγράμματα οι τιμές του μετρητικού οργάνου που διαμορφώθηκε έρχονται πολύ κοντά με τις θεωρητικές τιμές. Η μέση αποχή που έχουν οι μετρήσεις του οργάνου με τις θεωρητικές τιμές είναι:

$$3.08 \text{ micro strain} = 0.00000308 \text{ strain}$$

Ενώ ποσοστιαία η μέση απόκλιση που παρατηρείται είναι 5.3%. Η ποσοστιαία απόκλιση αυξάνεται πολύ πιο έντονα από την απολυτή στις μικρές τιμές παραμόρφωσης εφόσον η οποιαδήποτε μεταβολή για μικρές μετρήσεις έχει μεγάλη ποσοστιαία μεταβολή. Παρατηρώντας τα αποτελέσματα των συγκρίσεων των πειραματικών δεδομένων και του θεωρητικού μοντέλου φαίνεται πως οι μετρήσεις είναι σταθερές και ακριβής δίνοντας έτσι ένα αποδεκτό αποτέλεσμα.

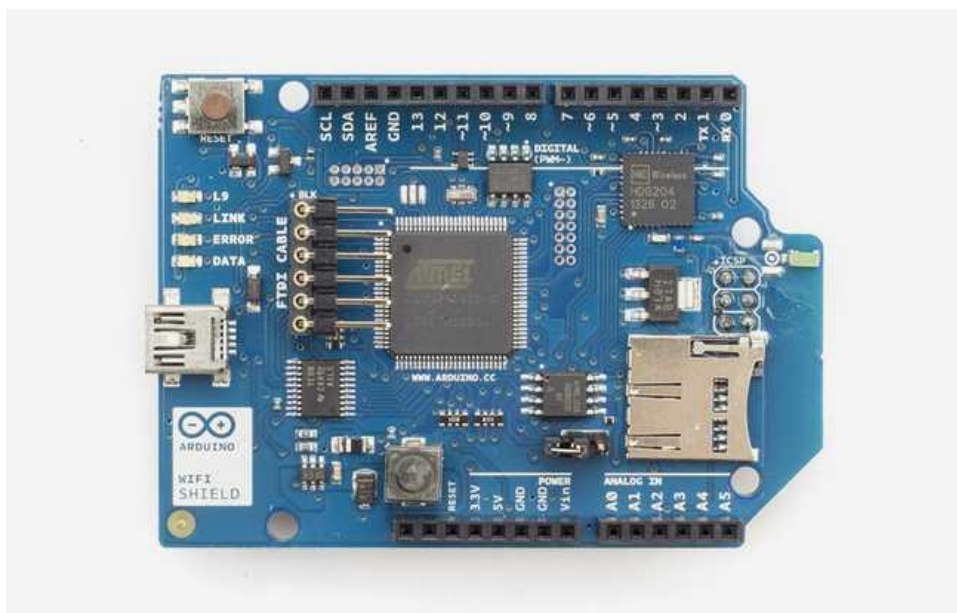
4.3 ΣΥΜΠΕΡΑΣΜΑΤΑ ΚΑΙ ΠΙΘΑΝΕΣ ΒΕΛΤΙΩΣΕΙΣ

Βλέποντας τα αποτελέσματα των μετρήσεων και συγκρίνοντας τα με την ελεγμένη συσκευή και τις θεωρητικές τιμές, παρατηρείται ότι το μαθηματικό μοντέλο που εφαρμόστηκε είναι αξιόπιστο, επίσης φαίνεται ότι η συσκευή διαθέτει επαναληψιμότητα και ακρίβεια στις μέτρησης, πράγμα το οποίο τη καθιστά ικανή για χρήση σε πειράματα και συλλογή δεδομένων. Έπειτα από τις δοκιμές που έγιναν φαίνεται ότι η εφαρμογή που αναπτύχθηκε λειτουργεί σύμφωνα με τις απαιτήσεις της, κοινώς είναι ικανή να λαμβάνει πολλαπλές μετρήσεις την ίδια χρονική στιγμή, να τις μοιράζεται μέσω bluetooth, στη συνέχεια είναι ικανή να δέχεται εντολές από το κινητό

του χρήστη και να τις εκτελεί, τέλος είναι ικανή να ξεκινάει και να σταματάει την αποθήκευση δεδομένων στη κάρτα SD.

Υπάρχουν πολλαπλοί παράγοντες οι οποίοι επηρέασαν το τελικό αποτέλεσμα. Αρχικά το μηχάνημα διαμορφώθηκε ως ένα σύνολο πλακετών, αισθητήρων, μικροελέγκτων, και αντιστάσεων αυτό έχει ως αποτέλεσμα πολλές συγκολλήσεις καλωδίων pin και επαφών πράγμα το οποίο επηρεάζει την ακρίβεια του μηχανήματος. Άλλος παράγοντας είναι οι αντιστάσεις που χρησιμοποιήθηκαν, οι αντιστάσεις που χρησιμοποιήθηκαν είναι αντιστάσεις ακρίβειας και έχουν απόκλιση 0.1% πράγμα το οποίο τις καθιστά ικανοποιητικές για την εφαρμογή αλλά αφήνει περιθώρια βελτίωσης εφόσον θεωρητικά οι αντιστάσεις και ο αισθητήρας πρέπει να έχουν ακριβείς τιμές.

Παρόλες τις λειτουργίες που έχει η συσκευή υπάρχουν βελτιώσεις και προσθήκες που μπορούν να γίνουν. Προκειμένου να βελτιωθεί η ακρίβεια του οργάνου μπορούν να αλλάξουν οι αντιστάσεις σε αντίστοιχες με μεγαλύτερη ακρίβεια. Στη συνέχεια εφόσον το μηχάνημα διαθέτει Bluetooth, θα μπορούσε να συνδέεται με υπολογιστή και με βάση τη παραμόρφωση που μετριέται, να δημιουργείται σε πραγματικό χρόνο διάγραμμα παραμόρφωσης - χρόνου, αυτό μπορεί να γίνει είτε μέσω Bluetooth είτε μέσω Wifi. Στη περίπτωση χρησιμοποιηθεί Wifi για την επικοινωνία μπορεί στη ήδη υπάρχουσα διάταξη να προτεθεί ένα shield το οποίο θα δώσει την δυνατότητα στη συσκευή να εκπέμπει και να λαμβάνει πληροφορίες μέσω Wifi (Σχήμα 4,4).



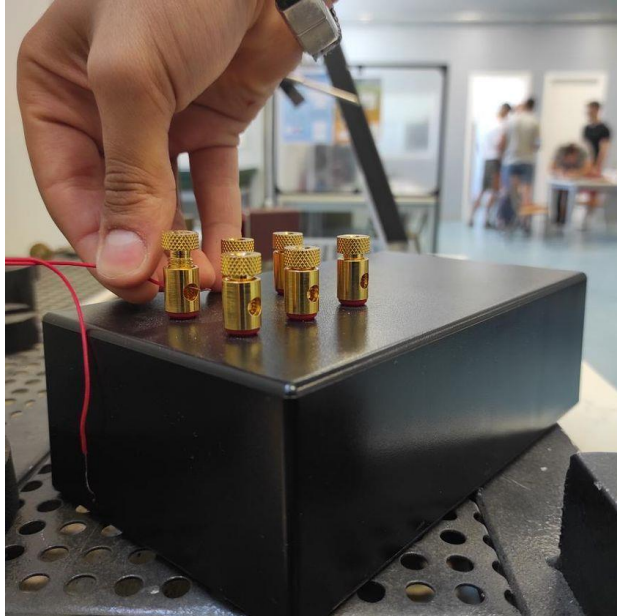
Σχήμα 4,4 Arduino wifi shield.

Στη συνέχεια μια βελτίωση που θα μπορούσε να γίνει είναι η αναβάθμιση από τις πλακέτες οι οποίες μέσω καλωδίων συνδέονται μεταξύ τους σε μια ενιαία πλακέτα. Προκειμένου να επιτευχθεί αυτό θα πρέπει και να παραχθεί μια πλακέτα η οποία θα έχει σωστή συνδεσμολογία και τα απαραίτητα εξαρτήματα ώστε να εκτελεί όλες τις λειτουργίες του μηχανήματος.

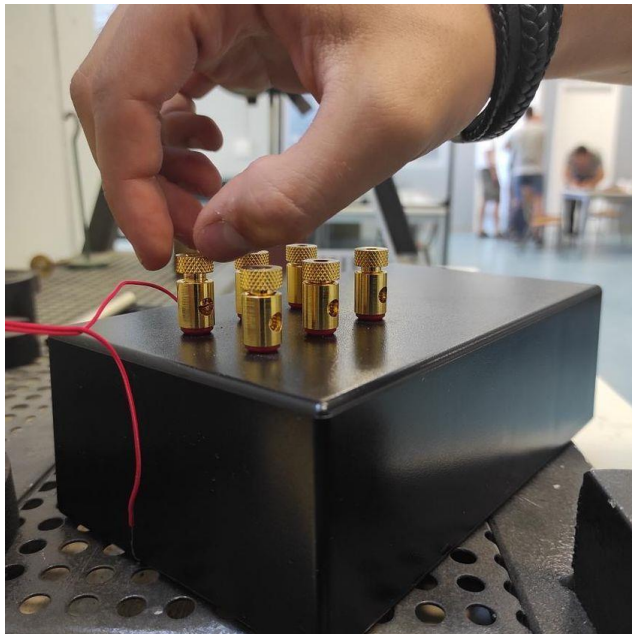
ΠΑΡΑΡΤΗΜΑΤΑ

Α. ΟΔΗΓΙΕΣ ΧΡΗΣΗΣ

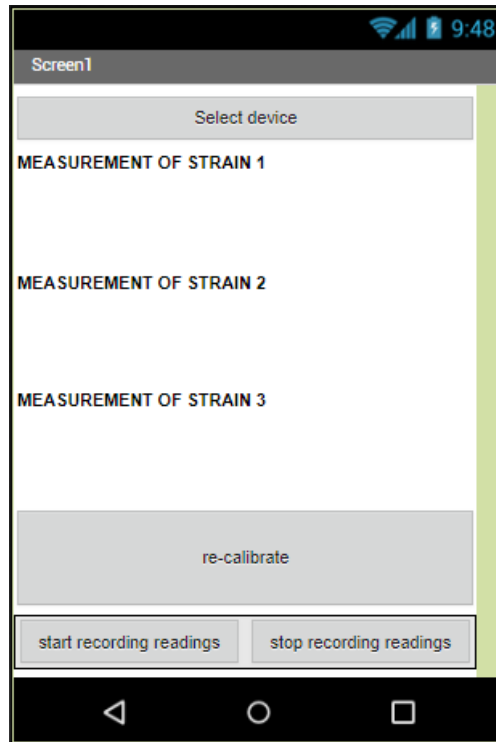
- 1) Αρχικά συνδέστε τα επημυκινσιόμετρα τα οποία επιθυμείτε να μετρήσετε στα κανάλια της συσκευής (μέγιστος αριθμός 3).



- 2) Βεβαιωθείτε πως υπάρχει επαφή στα πηνία και ότι έχουν βιδωθεί σωστά.



- 3) Ανοίξτε στην ειδικά διαμορφωμένη εφαρμογή στο κινητό σας τηλέφωνο και βεβαιωθείτε ότι έχετε ανοιχτό το Bluetooth.



- 4) Προσθέστε την κάρτα μνήμης στη συσκευή και σιγουρευτείτε πως έχει εισέρθει σωστά στην ειδική υποδοχή.

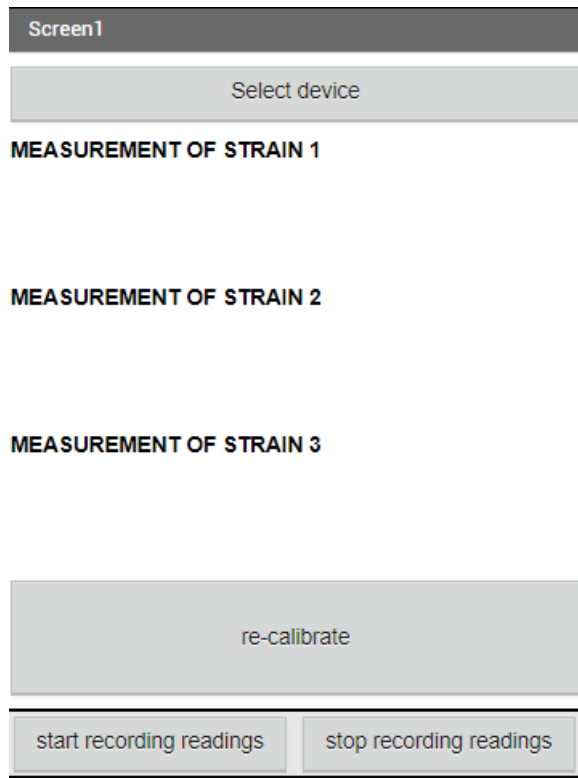


- 5) Πρωτού συνδέσετε τη συσκευή στη παροχή ρεύματος σιγουρευτείτε ότι το σύστημα βρίσκεται σε ηρεμία.
- 6) Στη συνέχεια συνδέστε τη συσκευή στη παροχή ρεύματος (5Vold, 1Amp).

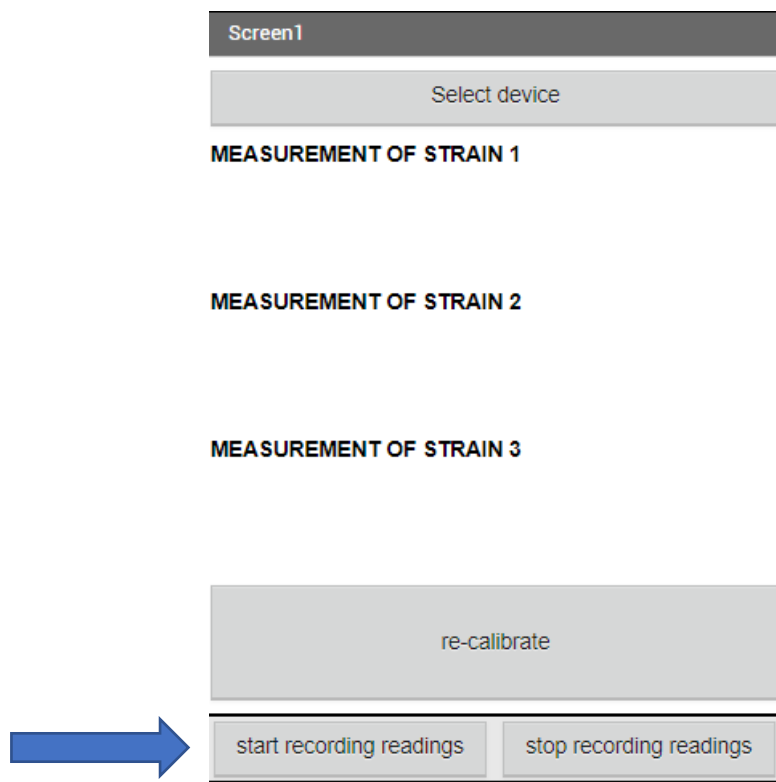
5V 1A



- 7) Με το που συνδεθείτε στη συσκευή θα πρέπει να δείτε το μήνμα “calibrating” στη θέση των τριών μετρήσεων
- 8) Αφού τελειώσει από τη διαδικασία υπολογισμού μονίμου σφάλματος θα αρχίσουν να εμφανίζονται οι μετρήσεις των επημκινσιόμετρων στην εφαρμογή.
- 9) Εάν επιθυμείτε να επαναλάβετε τη διαδικασία υπολογισμού σφάλματος σε οποιαδήποτε στιγμή κατά τη διάρκεια του πειράματος πιέστε το “re-calibrate”.



10) Σε περίπτωση που θέλετε να ξεκινήσετε τη καταγραφή των μετρήσεων πιέστε το κουμπί “start recording readings”.



11) Προκειμένου να σταματήσετε τη καταγραφή των μετρήσεων πιέστε το πλήκτρο “stop recording readings”.

Screen1

Select device

MEASUREMENT OF STRAIN 1

MEASUREMENT OF STRAIN 2

MEASUREMENT OF STRAIN 3

re-calibrate

start recording readings stop recording readings



ΣΗΜΑΝΤΙΚΕΣ ΠΑΡΑΤΗΡΗΣΕΙΣ

- 1) Το μηχάνημα μπορεί να πάρει από μια έως και τρεις μετρήσεις ταυτόχρονα.
- 2) Στη περίπτωση που έχει περάσει κάποιος χρόνος από την ενεργοποίηση της συσκευής και της σύνδεσής σας με αυτή μέσω του κινητού πιθανόν να παραληφθούν τα μηνύματα και να δείτε αμέσως τις μετρήσεις
- 3) Μόνο επιμηκυνσιόμετα με αντίσταση 120Ω και $GF = 2$ μπορούν να χρησιμοποιηθούν με τη συσκευή.

B. ΚΩΔΙΚΑΣ ARDUINO

```
#include <SD.h> //SD CARD BREAKOUT BOARD LIBRARY

long sample = 0 ; //TEMPORARY STORAGE VALUE FOR CALIBRARION
long sample_temp; //TEMPORARY VALUE THAT STORES
int x; //VALUE REPRESENTING THE HX USED (1,2,3)
int e1,e2,e3; //variables that represent the 3 strain outputs

//

long count = 0; // Instant Reading value
//
long sample1 = 0; // Sample for calibrating
long sample2 = 0; // Sample for calibrating
long sample3 = 0; // Sample for calibrating
//

//

unsigned long time_from_reading_start; //time point that the first recorded reading
//took place

unsigned long general_time;

char Incoming_value = 0; //VALUE INPUTED TO THE ARDUINO FROM
//PHONE, RESPONSIBLE FOR WITING SO SD CARD AND RECALIBRATING THE
//DEVICE

int memory_controller; //VARIABLE USED TO STORE THE VALUE THAT
//IS INPUTED FROM THE PHONE APPLICATION

int readings_counter; //USED TO STORE THE AMOUNT OF TIMES WE
//WROTE DATA TO THR SD CARD

File myFile;

//
```

```

//
byte DT1=14; //
byte SCK1=15; //SPI Clock
byte DT2=16; //
byte SCK2=17; //
byte DT3=18; //
byte SCK3=19; //
//

##### READING ONCE FROM HX711#####
unsigned long readrealcount(int x){
    ### SELECTION OF INPUT PINS BASED ON FUNCTION INPUT
    if(x==1){    ### FIRST HX711
        SCK1=15;
        DT1=14;
    }
    if(x==2){    ### SECOND HX711
        SCK1=SCK2;
        DT1=DT2;
    }
    if(x==3){    ### THIRD HX711
        SCK1=SCK3;
        DT1=DT3;
    }

    /*pulse reading function to avoid using the standart hx711 library
    * as we want to get the voltage diffrence in order to calculate strain
    * and not the weight */

    unsigned long Count;

```

```

unsigned char i;
pinMode(DT1, OUTPUT); //SETTING DT PIN TO OUTPUT MODE
digitalWrite(DT1,HIGH); //SETTING DT HIGH
digitalWrite(SCK1,LOW); //SETTING SCK TO LOW
Count=0;
pinMode(DT1, INPUT);
while(digitalRead(DT1)); //WAIT UNTILL DT(DATA LINE) GOES LOW
for (i=0;i<24;i++) //PULSE THE CLOCK PIN(SCK) 24 TIMES TO READ THE
//DATA(24BIT SIGNED INTEGER) FROM HX711
{
digitalWrite(SCK1,HIGH);
Count=Count<<1;
digitalWrite(SCK1,LOW);
if(digitalRead(DT1))
Count++;
}
digitalWrite(SCK1,HIGH);
Count=Count^0x800000;
digitalWrite(SCK1,LOW);
return(Count); //RETURN THE FINAL READING
}
//
void setup()
{
readings_counter = 0; //NUMBER OF TIMES WE WROTE OUR READINGS
//TO THE SD CARD IS SET TO 0
Serial.begin(9600); //SERIAL PORT INITIALIZATION
pinMode(10, OUTPUT); //SETTING PIN 10 AS OUTPUT IN ORDER TO SET
//UP THE SD CARD BREAKOUT BOARD

pinMode(15, OUTPUT); // CLOCK PIN HX711(1ST)

```

```

pinMode(17, OUTPUT);    // CLOCK PIN HX711(2ND)
pinMode(19, OUTPUT);    // CLOCK PIN HX711(3RD)

##### CHECK IF SD CARD HAS BEEN INITIALIZED #####
if (!SD.begin(10)) {
  Serial.print("initialization failed!");
  Serial.print("|");
  Serial.print("initialization failed!");
  Serial.print("|");
  Serial.print("initialization failed!");
  Serial.print("|");
  return;
}
Serial.print("initialization of sd done");
Serial.print("|");
Serial.print("initialization of sd done");
Serial.print("|");
Serial.print("initialization of sd done");
Serial.print("|");
//
myFile = SD.open("test.txt", FILE_WRITE); // OPEN THE SD CARD FILE
//

##### CHECK THAT THE OUR FILE OPENED IN THE SD CARD ###
if (myFile) {
  Serial.print("Ready to record");
  Serial.print("|");
  Serial.print("Ready to record");
  Serial.print("|");
  Serial.print("Ready to record");
}

```

```

Serial.print("|");
} else {
Serial.println("unable to record");
Serial.print("|");
Serial.println("unable to record");
Serial.print("|");
Serial.println("unable to record");
Serial.print("|");
}

##### CALIBRATION SEQUENCE FOR ALL THE HX711 #####
//the " " and the "|" are needed in order to print them in order and have them
//centered in the app
Serial.print("calibrating 1");
Serial.println(" ");
Serial.println(" ");
Serial.print("|");
Serial.print("calibrating 2");
Serial.println(" ");
Serial.println(" ");
Serial.print("|");
Serial.print("calibrating 3");
Serial.println(" ");
Serial.println(" ");
Serial.print("|");
delay(500);

Incoming_value = 0; //THIS VALUE IS SET TO ZERO SO THE DEVICE
//STARTS OFF WITHOUT WRITING TO THE SD CARD

```



```

sample1 = calibration(1); // calibration of the 1ST hx711
sample2 = calibration(2); // calibration of the 2ND hx711
sample3 = calibration(3); // calibration of the 3RD hx711

}

//*****

void loop()
{
myFile = SD.open("test.txt", FILE_WRITE); //OPENING THE TEXT FILE THAT WILL
//STORE OUR READINGS

Serial.setTimeout(100);          //THE TIMEOUT FUNCTIONS STOPS WAITING
//FOR AN INPUT AFTER A SET AMOUNT OF TIME

          //IN OUR CASE WE NEED IT BECAUSE WE DONT
//HAVE A CONSTANT INPUT STRAM FROM THE USER

//### TAKING A READING FROM EVERY HX711 ###
e1 = reading(1);
delay(50);
e2 = reading(2);
delay(50);
e3 = reading(3);
delay(50);
//

Incoming_value = Serial.parseInt(); //READING VALUE FROM PHONE(WAITS
//FOR 100 MS)

// the incoming value resets to zero after every loop because if the phone doesnt
//input anything it takes it as a zero

```

```

// so we save the value of the input in a variable:

// THE INCOMING VALUE RESETS TO ZERO IF THE IS NO INPUT FROM THE
//PHONE

// AS A RESULT WE NEED TO SAVE THE INCOMING VALUE INTO A SEPARATE
//VALUE

if(Incoming_value == 1) {          //RECALIBRATION SEZUENCE
    memory_controller=1;          //SAVING THE INCOMING VALUE SO IT
//DOESNT RESET TO 0 IN THE NEXT LOOP
}

else if(Incoming_value == 2){      //START WRITING TO SD CARD
    time_from_reading_start = millis(); //saves time of start to a value
    readings_counter+=1;          //INCREASE THE NUMBER OF READING
//CYCLES BY ONE

    memory_controller=2;          //SAVING THE INCOMING VALUE SO IT
//DOESNT RESET TO 0 IN THE NEXT LOOP

    myFile.print("set of readings: "); //WRITING TO THE SD CARD THE NUM OF
//THE CURRENT READING SYCLE
    myFile.println(readings_counter);
}

else if(Incoming_value == 3){      //STOP WRITING TO THE SD CARD
    memory_controller=3;          //SAVING THE INCOMING VALUE SO IT
//DOESNT RESET TO 0 IN THE NEXT LOOP
}

##### READ/WRITE TO SD AND RECALIBRATE #####

if(memory_controller == 2){        //PRINT TO SD CARD

    general_time = millis()-time_from_reading_start; //SAVING THE TIME OF THE
//READING

```

```

myFile.print(general_time/1000);          //RESETTING THE TIME FROM
//SECONDS TO MS

myFile.print(" ");                        //WRITING THE READINGS
myFile.print(e1);
myFile.print(" ");
myFile.print(e2);
myFile.print(" ");
myFile.print(e3);
myFile.println(" ");

}else if(memory_controller == 3){          //STOP PRINTING READINGS(DO
//NOTHING IF INPUTEED VAL == 3)

}else if(memory_controller == 1){          //START RECALIBRATION
//SEQUENCE

delay(100);
Serial.print("calibrating 1");            //PRINT TO THE PHONE THAT THE
//DEVICE IS CALIBRATING
Serial.println(" ");
Serial.println(" ");
Serial.print("|");
Serial.print("calibrating 2");
Serial.println(" ");
Serial.println(" ");
Serial.print("|");
Serial.print("calibrating 3");
Serial.println(" ");
Serial.println(" ");
Serial.print("|");

sample1 = calibration(1);                  //CALIBRATION OF 1ST HX711

```

```

sample2 = calibration(2);           //CALIBRATION OF 2ND HX711
sample3 = calibration(3);           //CALIBRATION OF 1RD HX711
delay(100);

memory_controller = 3;              //stopping the recordings on the sd card
//(DO NOTHING)
}

####PRINTING TO PHONE###

#### PRINTING VALUES OF THE FIRST GAGE ###
Serial.print("1: ");
Serial.print(int(e1));
Serial.println(" ");
Serial.println(" ");
Serial.print("|");

#### PRINTING VALUES OF THE SECOND GAGE ###
Serial.print("2: ");
Serial.print(int(e2));
Serial.println(" ");
Serial.println(" ");
Serial.print("|");

#### PRINTING VALUES OF THE THIRD GAGE ###
Serial.print("3: ");
Serial.print(int(e3));
Serial.println(" ");
Serial.println(" ");
Serial.print("|");
delay(500);

myFile.close(); //CLOSING THE SD CARD FILE

```

```

}

unsigned long reading(int x){ //function for taking a reading from the HX711

if(x==1){sample_temp = sample1;
}else if(x==2){sample_temp = sample2;
}else if(x==3){sample_temp = sample3;}

count = readrealcount(x); // SAVING THE READING VALUE INTO A VARIABLE

long val=(count-sample_temp); // DELETING THE ERROR THAT WE DETECTED
//IN THE CALIBRATION SEQUENCE

float vout=(val*0.43); // CONVRETING THE VAKUE INTO VOLTS (in nVolts )
//1=0.43nV

int e=4*(vout/3.77)*0.002; // CONVERSION TO MICROSTRAIN

return e; // RETURNING THE MICROSTRAIN
}

unsigned long calibration(int x){ //CALIBRAION FUNCTION

sample = 0; //RSET TEMPORARY DATA STORAGE VALUE TO
//ZERO

for(int i=0;i<100;i++) //SAVING 100 READINGS INTO THE TEMPORARY
//VALUE

{
count = readrealcount(x);

sample+=count;

}

sample/=100; //AVERAGING THEM OUT

count = 0; //RESETING THE VALUES TO ZERO

return sample;

}

```

ΜΠΛΟΚ ΕΝΤΟΛΩΝ ΕΦΑΡΜΟΓΗΣ

```
when ListPicker1 .BeforePicking
do set ListPicker1 .Elements to BluetoothClient1 .AddressesAndNames

when ListPicker1 .AfterPicking
do set ListPicker1 .Selection to call BluetoothClient1 .Connect
address ListPicker1 .Selection
if BluetoothClient1 .IsConnected
then set Clock1 .TimerAlwaysFires to true
set ListPicker1 .Visible to false
set MEASUREMENT_1 .Visible to true
set MEASUREMENT2 .Visible to true
set MEASUREMENT_3 .Visible to true
set M1 .Visible to true
set M2 .Visible to true
set M3 .Visible to true

initialize global LIST to create empty list
initialize global INPUT to ""

when Clock1 .Timer
do if BluetoothClient1 .IsConnected
then if call BluetoothClient1 .BytesAvailableToReceive > 0
then set global INPUT to call BluetoothClient1 .ReceiveText
numberOfBytes call BluetoothClient1 .BytesAvailableToReceive
set global LIST to split text get global INPUT
at ""
set M1 .Text to select list item list get global LIST
index 1
set M2 .Text to select list item list get global LIST
index 2
set M3 .Text to select list item list get global LIST
index 3
set global INPUT to ""
set global LIST to create empty list

when Button1 .Click
do call BluetoothClient1 .SendText
text "1"

when Button2 .Click
do call BluetoothClient1 .SendText
text "2"

when Button3 .Click
do call BluetoothClient1 .SendText
text "3"
```

ΒΙΒΛΙΟΓΡΑΦΙΑ

1. Τρικαλινός, Χ. (2014). Εισαγωγή στη θεωρία Σφαλμάτων (σημειώσεις). Αθήνα: Εθνικό Καποδιστριακό Πανεπιστήμιο Αθηνών.
2. Gridling G. & Weiss B. (2017). Introduction to Microcontrollers. Vienna, Austria. Institute of Computer Engineering Embedded Computing Systems Group.
3. International Vocabulary of Metrology – Basic and General Concepts and Associated Terms VIM, 3rd edition, JCGM 200:2008
4. Tutak P. (2015). Application of strain gages in measurements of strain distribution in complex objects. Lodz, Poland. University of Social Sciences
5. Δούκας Γ. Ιστορική εξέλιξη της επιστήμης της Μετρολογίας (σημειώσεις). Θεσσαλονίκη: Αριστοτέλειο πανεπιστήμιο Θεσσαλονίκης.
6. Πανεπιστήμιο Ιωαννίνων, Τμήμα Μηχανικών Επιστήμης Υλικών (σημειώσεις). (2016).
7. Ismailov A. & Jo'rayev Z. (2022). Study of Arduino microcontroller board. Andijan, Uzbekistan. Andijan Machine Building Institute.
8. Prof. Douglas P. Holmes. Lesson 2: strain mechanics of deformable bodies(σημειώσεις). Boston university.
9. Γιαννακόπουλος Ι. (2011). Σημειώσεις Πειραματικής Αντοχής Υλικών. Αθηνά. ΤΕΙ Πειραιώς.
10. Khamadamov U.(2020). Introduction to Microcontrollers. Tashkent, Uzbekistan. Tashkent University of Information Technology.
11. Bhagwati P., Sinha G. & Naveen G.(2020) . Advances to modern sensors. Bristol, UK. IOP Publishing.
12. Embedded system – SEC1320. Chennai, India. Sathyabama university.
13. Vordos N., Gkika D. & Bandekas D.(2020). Wheatstone bridge and bioengineering. Καβαλά. Διεθνές Πανεπιστήμιο Ελλάδος.
14. Ldi V.(2019). Bluetooth Technology and Application. Helsinki, Finland. Metropolia University of Applied Sciences.

15. Nawrocki W. (2005). Measurement Systems and Sensors. Norwood, USA. Artech House.
16. Ababei C.(2013). Lecture 12: SPI and SD cards. Buffalo, USA. University of Buffalo.
17. ΜΠΟΥΥΤΙΚΟΣ Χ. (2020). Ανάπτυξη και κατασκευή συστήματος μετρήσεων παραμορφώσεων με χρήση ηλεκτροεπιμηκυνσιομέτρων (strain gages) και Arduino. Πάτρα. Πανεπιστήμιο Πελοποννήσου.
18. Jayakumar M. (2016). Arduino and Android using MIT app inventor 2.0.
19. Kirch-Prinz U. & Prinz P (2002). A Complete Guide to Programming C++. Jones & Barlett learning.
20. Jayaram M.A.(2012). Strength of materials : A Fundamental Approach. Siddaganga, India. Siddaganga Institute of Technology.