ΔΙΕΘΝΕΣ
ΠΑΝΕΠΙΣΤΗΜΙΟ
ΤΗΣ ΕΛΛΑΔΟΣ

ΣΧΟΛΗ ΜΗΧΑΝΙΚΩΝ

ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ
ΚΑΙ ΗΛΕΚΤΡΟΝΙΚΩΝ ΣΥΣΤΗΜΑΤΩΝ

ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ

«ΗΛΕΚΤΡΟΝΙΚΟΣ ΠΡΟΠΟΝΗΤΗΣ ΠΟΛΕΜΙΚΩΝ ΤΕΧΝΩΝ ΜΕ ΜΕΤΡΗΣΗ ΚΑΙ ΕΝΔΕΙΞΗ ΙΚΑΝΟΤΗΤΩΝ»

eCoach

Του φοιτητή
Δημήτριος Ζωγράφος.
Αρ. Μητρώου: 512125

Επιβλέπων
Μιχάλης Σπάσος
Αναπληρωτής Καθηγητής

15/01/2021

Τίτλος Π.Ε. «ΗΛΕΚΤΡΟΝΙΚΟΣ ΠΡΟΠΟΝΗΤΗΣ ΠΟΛΕΜΙΚΩΝ ΤΕΧΝΩΝ ΜΕ ΜΕΤΡΗΣΗ ΚΑΙ ΕΝΔΕΙΞΗ ΙΚΑΝΟΤΗΤΩΝ»

Κωδικός Π.Ε. 19119

Ονοματεπώνυμο φοιτητή ΖΩΓΡΑΦΟΣ ΔΗΜΗΤΡΙΟΣ

Ονοματεπώνυμο εισηγητή ΣΠΑΣΟΣ ΜΙΧΑΛΗΣ

Ημερομηνία ανάληψης Π.Ε. 19/03/2019

Ημερομηνία περάτωσης Π.Ε. 15/01/2021

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

«Αφιέρωση»

Η πτυχιακή αυτή είναι αφιερωμένη στους γονείς μου , οι οποίοι με στήριξαν με υπομονή όλα αυτά τα χρόνια και οφείλω όλη την διαδρομή των σπουδών μου μέχρι τώρα.

Πρόλογος

Καθώς τα αθλήματα ήταν πάντα βασικό κομμάτι της καθημερινότητας πολλών ανθρώπων, οι πολεμικές τέχνες έχουν δείξει άνοδο τα τελευταία χρόνια. Η άνοδος αυτή καθιστά τις προπονήσεις πιο απαιτητικές και δύσκολες με αποτέλεσμα οι προπονητές ερασιτεχνικών και επαγγελματικών συλλόγων, να ψάχνουν καινούργιες μεθόδους προπόνησης. Ακόμα ένα πρόβλημα της ανόδου αυτής, είναι η οργάνωση που χρειάζεται προκειμένου να καλυφτεί ο όγκος των απαραιτήτων ασκήσεων για την σταδιακή και ομαλή εξέλιξη των αθλητών. Τα προβλήματα αυτά, μπορούν πολύ εύκολα να λυθούν με την εφαρμογή της τεχνολογίας μέσα στους συλλόγους πολεμικών τεχνών. Με αυτή την αφορμή, προκλήθηκε η απόφαση δημιουργίας ενός ηλεκτρονικού προπονητή στα πλαίσια της ΠΕ. Σκοπός του ηλεκτρονικού προπονητή είναι να αντικαταστήσει τον φυσικό προπονητή σε εξουθενωτικές διαδικασίες, που επαναλαμβάνονται συχνά και η παρουσία του δε χρήζει απαραίτητη. Καθώς και η καταγραφή των ικανοτήτων κάθε προπόνησης με οργανωτικότητα, θα ήταν απαραίτητη για να κάνει την διάφορα μέσα σε ένα τόσο ανταγωνιστικό περιβάλλον όσο ο αθλητισμός.

Περίληψη

Η παρούσα πτυχιακή εργασία έχει ως στόχο την υλοποίηση ενός κυκλώματος καταγραφής και μίας εφαρμογής ένδειξης ικανοτήτων των αθλητών που ασχολούνται με τις πολεμικές τέχνες.

Τα κύρια στοιχεία που την απαρτίζουν είναι τρία. Το πρώτο αφορά την δειγματοληψία και την επεξεργασία σήματος. Τα ερεθίσματα που δέχεται ο αισθητήρας, την μετατροπή τους σε ηλεκτρικό σήμα και στην συνέχεια την επεξεργασία τους προκειμένου να τα μεταφέρει σε ψηφιακό περιβάλλον. Το δεύτερο κύριο στοιχείο, περιστρέφεται γύρω από τις λειτουργίες του BLE εξυπηρετητή (server), ο οποίος φροντίζει για την επεξεργασία των δεδομένων που ελήφθησαν από τον αισθητήρα και καθιστά εφικτή την αποστολή τους σε όποιον BLE πελάτη (client) του έχει δώσει εντολή. Το τρίτο κύριο στοιχείο, αφορά την σύνδεση του BLE πελάτη σε έναν BLE εξυπηρετητή, την αίτηση δεδομένων που καταγράφθηκαν, την ένδειξή τους και στην συνέχεια την αποθήκευση τους σε μία μικρή βάση δεδομένων.

Συμπερασματικά, πρόκειται για μία εφαρμογή η οποία έχει σαν στόχο τους αθλητές και τους προπονητές που ασχολούνται με τις πολεμικές τέχνες, με σκοπό να τους βοηθήσει μέσα από την μέθοδο της αξιολόγησης και της ανατροφοδότησης, να αναπτύξουν ικανότητες όπως η δύναμη και η ταχύτητα.

«MARTIAL ARTS ELECTRONIC COACH WITH MEASURING AND INDICATING SKILLS»

ZOGRAFOS DIMITRIOS

Abstract

The current thesis has as main object the implementation of a data acquisition circuit and an application which indicates the skills of people working on martial arts.

It consist of three basic parts. The first one is about sampling the data acquired by the sensor, after that the transducer takes place so the data could be converted into electric analog signal which after some processing, digital signal takes its place. The second basic part is about the BLE server processing the data and sending them of to a BLE client, after completing handshake with it. The third part is about the completion of the digital BLE handshake from the client's point of view, the request of the data acquired from the sensor, then indicating them on a screen and saving them on a local database based on the Smartphone.

In conclusion, it is about an application with its target group consisting of martial arts fighters and coaches who would like to see their skills upgrading through evaluation and feedback.

Ευχαριστίες

Αρχικά, χρωστάω ένα μεγάλο ευχαριστώ στον καθηγητή μου, κ. Σπάσο Μιχάλη, του οποίου η καθοδήγηση και η κατανόηση συνέβαλαν αδιαμφισβήτητα στην περάτωση αυτής της πτυχιακής εργασίας.

Επίσης θα ήθελα να ευχαριστήσω όλους τους φίλους που με βοήθησαν ο καθένας με τον δικό του τρόπο όλο αυτό τον καιρό.

Τέλος, ευχαριστώ πολύ τους γονείς μου και την αδερφή μου, καθώς ήταν δίπλα μου και πρόθυμοι να με βοηθήσουν με αυτή την πτυχιακή εργασία.

Περιεχόμενα

Κατάλογος Σχημάτων

Κεφάλαιο 4

Κεφάλαιο 5

Κεφάλαιο 6

Κεφάλαιο 7

Κεφάλαιο 8

Κεφάλαιο 9

Κεφάλαιο 10

Κατάλογος Πινάκων

Κεφάλαιο 4

Κεφάλαιο 5

Κεφάλαιο 6

Κεφάλαιο 7

Συντομογραφίες

Δ.Ε.	Διπλωματική Εργασία
ΔΙΠΑΕ	Διεθνές Πανεπιστήμιο Ελλάδος
Π.Ε.	Πτυχιακή Εργασία
S.G.	Strain Gauge
A.D.C.	Analog to Digital Converter
M.I.T.	Massachusetts Institute of Technology
G.F.	Gauge Factor
P.C.B.	Printed Circuit Board
OS	Operating System
NDK	Native Development Kit
DC	Direct Current
IC	Integrated Circuit
Op-Amp	Operational Amplifier
In-Amp	Instumentation Amplifier
Ref	Reference
CMRR	Common Mode Reject Ratio.
BLE	Bluetooth Low Energy
USB	Universal Serial Bus

ΕΙΣΑΓΩΓΗ

Η φύση έχει προικίσει τον άνθρωπο με χιλιάδες ικανότητες. Συγκεκριμένα, τον έχει καταστήσει ως την πιο εξελιγμένη μορφή ζωής πάνω στον πλανήτη. Οι δυνατότητες του είναι με τέτοιο τρόπο δομημένες, ώστε αργά η γρήγορα να ξεπερνά κάθε εμπόδιο, νοητό και φυσικό. Και για την ικανοποίηση των δυνατοτήτων αυτών κρίνεται απαραίτητη η άσκηση, τόσο του μυαλού όσο και του σώματος.

Με τον αθλητισμό υπάρχει η δυνατότητα εκγύμνασης και του μυαλού και του σώματος. Για την ακρίβεια τα οφέλη του είναι τόσα πολλά, που η ανάλυση τους θα ξέφευγε από το θέμα αυτής της πτυχιακής εργασίας. Υπάρχει όμως μια πολύ σημαντική διαχωριστική γραμμή, που όσοι αγαπάνε τον αθλητισμό, τείνουν να ξεχνάμε την ύπαρξη της. Η γραμμή αυτή διαχωρίζει τον αθλητισμό από τον πρωταθλητισμό. Ένα τρανταχτό παράδειγμα είναι πως στον αθλητισμό δεν υπάρχουν αντίπαλοι, σε αντίθεση με τον πρωταθλητισμό.

Σε ρήξη με αυτό τον διαχωρισμό, έρχονται τα μαχητικά αθλήματα. Τα μαχητικά αθλήματα και συγκεκριμένα οι πολεμικές τέχνες είναι από την φύση τους ανταγωνιστικά στα πλαίσια αθλητισμού, και κατά πολύ περισσότερο στα πλαίσια του πρωταθλητισμού. Τα συναισθήματα που προκαλούνται ξέροντας πως όταν ξεκινήσει η προσομοίωση μάχης, θα πρέπει να προστατευθεί κανείς και να προκαλέσει πόνο στον αντίπαλο, προκειμένου να κερδηθούν οι πόντοι και κατ’επέκταση η μάχη, θυμίζουν αρχέγονες καταστάσεις και τείνουν προς τα ένστικτα επιβίωσης.

Σχήμα 0.1: Στιγμιότυπο επίθεσης με το πόδι σε ερασιτεχνικό αγώνα Taekwon-Do.

Οι λόγοι αυτοί έχουν αυξήσει ραγδαία το ενδιαφέρον και κατά συνέπεια τον ανταγωνισμό στον χώρο των πολεμικών τεχνών, αναγκάζοντας πολλούς προπονητές να εφαρμόζουν νέες τακτικές προπόνησης και να δοκιμάζουν νέες τεχνολογίες.

Έτσι, λογά της σχέσης μου με τον χώρο των πολεμικών τεχνών, θέλησα να συμβάλω στην εξέλιξη μιας σχετικά πρωτοποριακής εφαρμογής, στοχεύοντας στη δημιουργία ενός ηλεκτρονικού προπονητή. Πρόκειται για ένα εργαλείο στα χεριά των προπονητών και των αθλητών που μέσω της αξιολόγησης των ικανοτήτων τους, θα υπάρχει περιθώριο εύρεσης, για ακόμα πιο διακριτικά λάθη από ότι γίνεται μέχρι τώρα, καθώς και περιθώριο διόρθωσης.

Μερικές από τις λειτουργιές που θα έχουν οι ενδείξεις στην εφαρμογή ενε:

- Θα εμφανίζονται σε πραγματικό χρόνο,
καθώς είναι πολύτιμο σε μια δραστηριότητα που οι κινήσεις περιέχουν έκρηξη και διαρκούν μόλις μερικές δεκάδες χιλιοστά του δευτερόλεπτου.

- Θα παρουσιάζονται με την μορφή kg,
για την καλύτερη κατανόηση της κλίμακας από τους χρηστές.

- Θα αφορούν την δύναμη την ταχύτητα και την έκρηξη των χτυπημάτων.

- Θα παρουσιάζονται αναλύσεις
για παράδειγμα ο μέσος όρος χτυπημάτων ανά δευτερόλεπτο, ποσά χτυπήματα πραγματοποιήθηκαν στον δεδομένο χρόνο , διάφορα χρόνου από χτύπημα σε χτύπημα.

- θα αποθηκεύονται
προκειμένου να υπάρχει η καλύτερη δυνατή αξιολόγηση οποιαδήποτε στιγμή.

Για την υλοποίηση του ηλεκτρονικού προπονητή, ένας αισθητήρας SG κολλημένος πάνω σε επιφάνεια Plexiglas, θα τοποθετηθεί μέσα σε έναν στόχο όπως το Σχήμα 0.2.

Σχήμα 0.2: Στόχος προπόνησης για λακτίσματα και γροθιές.

Θα ακολουθήσει ένα κύκλωμα ενίσχυσης σήματος που θα καταλήγει στον ADC του μικροελεγκτή ESP32, και μέσω της ασύρματης επικοινωνίας BLE, ο μικροελεγκτής με τον ρολό του BLE εξυπηρετητή (server) θα μπορεί να στείλει τα δεδομένα σε όποιο Android κινητό έχει εγκατεστημένη την εφαρμογή του ηλεκτρονικού προπονητή.

Κεφάλαιο 1ο: Αισθητήρας Strain Gauge

1.1 Εισαγωγή

Το πρώτο εργαλείο που επιλέχθηκε για την υλοποίηση της κατασκευής του ηλεκτρονικού προπονητή είναι ο αισθητήρας Strain Gauge. Πρόκειται για έναν ενεργητικό αισθητήρα που λειτούργει με το φαινόμενο της πιεζοαντίστασης (piezoresistive) και ανήκει στην κατηγορία των αισθητηρίων πίεσης/δύναμης (Pressure/Force sensors). Το Strain Gauge, όπως φανερώνει και το όνομα του, είναι ένας αισθητήρας που χρησιμοποιείται για τη μέτρηση της καταπόνησης που δέχεται ένα στέρεο σώμα το οποίο έχει την ιδιότητα της ελαστικής παραμόρφωσης.

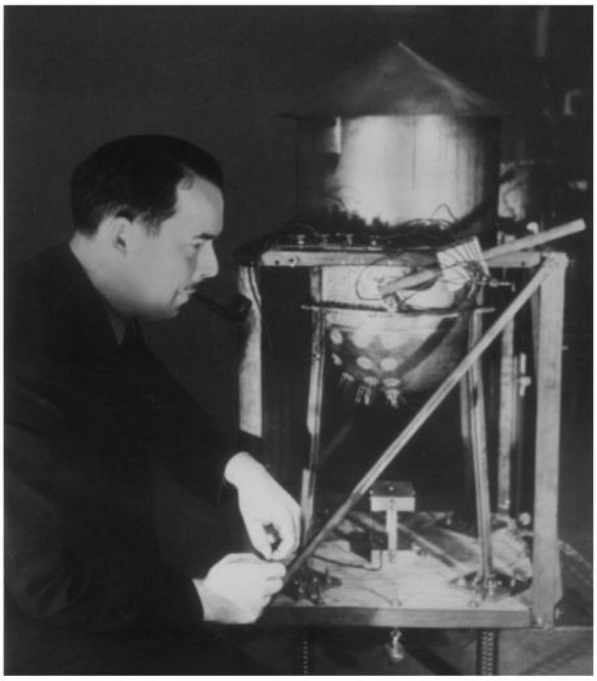
1.2 Ιστορική αναδρομή

Η αρχή λειτουργίας του αισθητήρα βασίστηκε πάνω στη σχέση των μεταλλικών αγωγών και της ελαστικής παραμόρφωσης τους που ανακαλύφθηκε το 1856 από τον βρετανό φυσικό William Thompson, γνωστό και ως Λόρδο Kelvin. Αναλυτικότερα, ο λόρδος Kelvin παρατήρησε τη διαφορά της ηλεκτρικής αντίστασης που παρουσιάζεται πάνω σε λεπτούς μεταλλικούς αγωγούς, όπως ο σίδηρος και ο χαλκός, όταν εφαρμόζεται σε αυτούς εφελκυσμός ή θλίψη.

Η πρώτη αξιοσημείωτη εφαρμογή του SG πραγματοποιήθηκε στις αρχές του 1930 που ο Charles Kearns προσπάθησε να καταγράψει την καταπόνηση των προπελών λόγο των δονήσεων που προκαλείται από τις μεγάλες ταχύτητες.

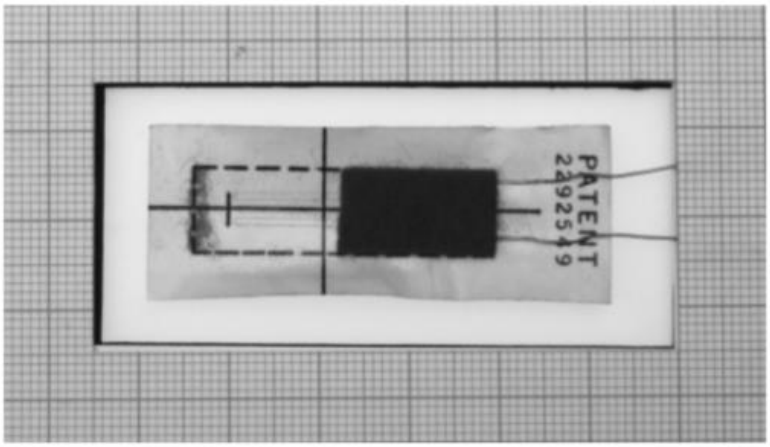
Ο αισθητήρας SG εφευρέθηκε σχεδόν την ίδια περίοδο από δύο διαφορετικά πρόσωπα στην Αμερική, που δούλευαν ανεξάρτητα ο ένας από τον άλλον. Ο τότε φοιτητής και ερευνητικός βοηθός στο Τεχνολογικό Ινστιτούτο της Καλιφόρνιας (Caltech), Edward E. Simmons ήταν ο πρώτος εφευρέτης το 1936. Έκανε πειράματα για την καταπόνηση και την παραμόρφωση που μπορούν να εμφανίσουν τα μέταλλα υπό μηχανική πίεση. Τα πειράματα του Simmons ήταν μέρος μιας ερευνητικής εργασίας των Dätwyler και Clark που ξεκίνησε το 1936 αλλά ολοκληρώθηκε και δημοσιεύτηκε δύο χρόνια αργότερα, το 1938.

Την ίδια χρονιά ο δεύτερος εφευρέτης Arthur C. Ruge, καθηγητής στο Τεχνολογικό Ινστιτούτο Μασαχουσέτης (MIT), εφηύρε τον αισθητήρα SG ώστε να βοηθήσει τον συνεργάτη του J. Hanns Maier να ερευνήσει την επίδραση των σεισμών στις μηχανικές κατασκευές. Η κατασκευή πάνω στην όποια γίνονταν τα πειράματα ήταν μια μινιατούρα δεξαμενής νερού βασισμένη πάνω σε ένα τραπέζι που δονείται με σκοπό την αναπαράσταση σεισμού. Ωστόσο οι μετρήσεις ήταν λανθασμένες με τα όργανα που διέθεταν καθώς η καταπόνηση της δεξαμενής ήταν πολύ μικρή και η κατασκευή της πολύ λεπτή. Έτσι ο Ruge αναγκάστηκε να χρησιμοποιήσει λεπτά καλώδια από ένα ποτενσιόμετρο και να καταλήξει στην ευρεσιτεχνία του αισθητήρα SG.

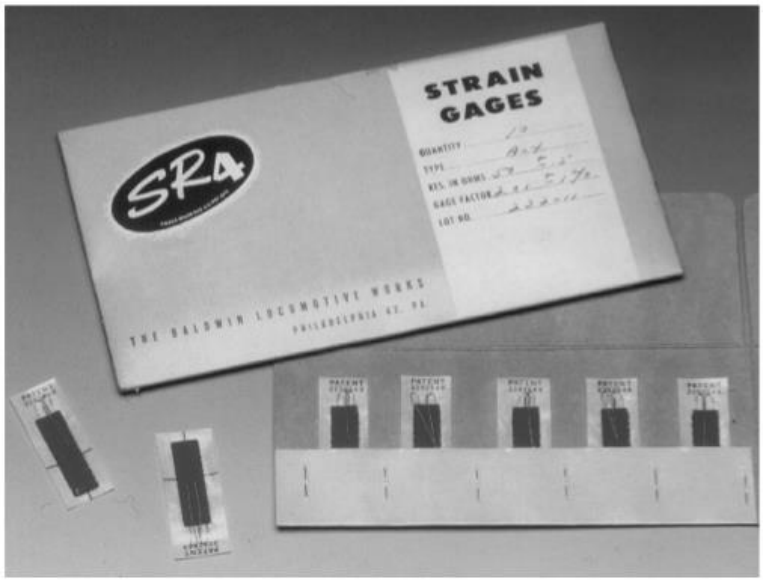


Σχήμα 1.1: Ο Arthur C. Ruge εκτελεί πειράματα πάνω στη μινιατούρα δεξαμενής νερού.

Αξιοσημείωτη είναι η δικαιοδοσία που έδωσε το MIT στον Ruge για την πατέντα που έφτιαξε καθώς του έδωσε το δικαίωμα να την αξιοποιήσει όπως ήθελε. Έτσι ο Ruge μαζί με τον συνάδελφο του Alfred V. deForest προχώρησαν σε συνεργασία με την εταιρία Baldwin-Southwark Corp. και οι πωλήσεις απεδείχθησαν κερδοφόρες. Στη συνέχεια η εταιρία Baldwin-Southwark έφερε σε συμφωνία τους Simmons-Ruge-deForest ώστε να αναγνωριστεί και ο Simmons ως εφευρέτης του αισθητήρα SG. Το προϊόν που θα ξεκινήσει να πωλείτε το 1941 από αυτή τη συμφωνία και θα ονομαστεί "SR-4", όπου "SR" είναι από τα αρχικά των δυο εφευρετών Simmons-Ruge και ο αριθμός "4" από τον αριθμό των ατόμων που συνέβαλαν σε αυτή τη συμφωνία.



Σχήμα 1.2: Η πατέντα του αισθητήρα SG από τον Arthur C. Ruge.



Σχήμα 1.3: Φωτογραφία του πρώτου SR4 που βγήκε στην αγορά.

Το 1952 η Αγγλική αεροναυτιλιακή εταιρεία Saunders-Roe αναζητώντας λύσεις για την εφαρμογή των SG σε πιο απαιτητικό περιβάλλον βρίσκει στήριγμα στην τότε ταχύρυθμα αναπτυσσομένη τεχνολογία των τυπωμένων κυκλωμάτων. Έτσι ξεκίνησε η κατασκευή SG αποτυπώνοντας το σχέδιο των πολύ λεπτών καλωδίων πάνω σε λεπτό φύλλο μέταλλου. Η κατασκευή πάνω σε φύλλα μέταλλου έχει πολλά πλεονεκτήματα όπως το πολύ χαμηλό κόστος παράγωγης και το μικρότερο μέγεθος.

1.3 Αρχές λειτουργίας

Ο αισθητήρας SG, όπως προαναφέραμε, λειτούργει με το φαινόμενο της πιεζοαντίστασης. Αναλυτικότερα είναι η αλλαγή της ηλεκτρικής αντίστασης ενός στέρεου σώματος μέταλλου ή ημιαγωγού, όταν εφαρμόζεται πάνω του κάποια μηχανική καταπόνηση με αποτέλεσμα την ελαστική του παραμόρφωση.

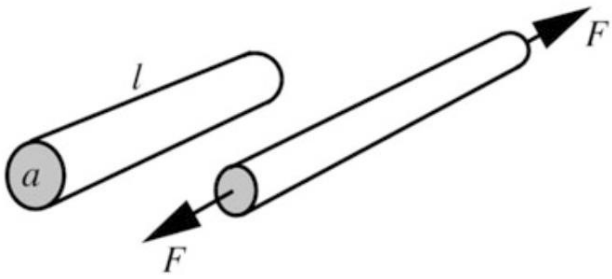
1.3.1 Το φαινόμενο της πιεζοαντίστασης

Η ηλεκτρική αντίσταση R για τους αγωγούς ορίζεται από την παρακάτω σχέση

$$R = \rho \frac{l}{\alpha} \tag{1.1}$$

όπου ρ είναι η ειδική αντίσταση του υλικού σε Ω (Α), l είναι το μήκος του αγωγού σε m (μετρά) και α είναι η περιοχή εγκάρσιας διατομής σε m^2 που παίρνει το ρεύμα. Η αναλόγια $\frac{l}{\alpha}$ είναι γνωστή και ως γεωμετρικός συντελεστής (geometry factor).

Με το Σχήμα 1.4 γίνεται κατανοητό πως όταν σε ένα καλώδιο εφαρμόσουμε μια δύναμη F, το μήκος του l θα μεγαλώσει και η περιοχή εγκάρσιας διατομής α θα μικρύνει με αποτέλεσμα να μεγαλώσει η ηλεκτρική αντίσταση του καλωδίου.



Σχήμα 1.4: Άσκηση δύναμης F και η παραμόρφωση που έχει σαν αποτέλεσμα σε ένα καλώδιο.

1.3.2 Τι είναι μηχανική καταπόνηση

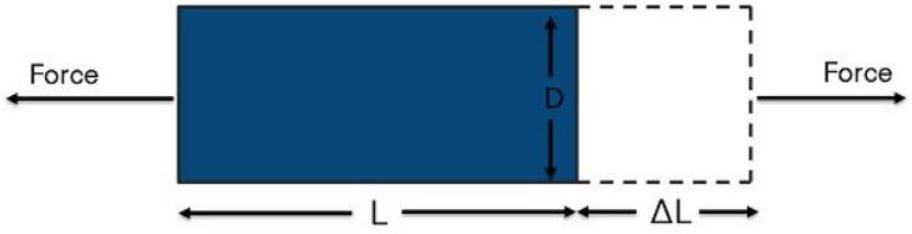
Ως μηχανική καταπόνηση θεωρείται η επίδραση των δυνάμεων και ροπών πάνω σε ένα στερεό σώμα με αποτέλεσμα την παραμόρφωση του. Χωρίζεται σε έξι βασικά ειδή:

- Εφελκυσμός
- Θλίψη
- Διάτμηση
- Κάμψη
- Στρέψη
- Λογισμός

Κάθε άλλο είδος είναι αποτέλεσμα του συνδυασμού αυτών των έξι ειδών.

1.3.3 Τι είναι παραμόρφωση (Strain)

Εφαρμόζοντας στις άκρες του σώματος της εικόνας 1.5 με μήκος *L*, δυο δυνάμεις με φορά από το κέντρο του σώματος, αντίθετες μεταξύ τους και κατά μήκος του αντικείμενου, πετυχαίνουμε τον εφελκυσμό του με παραμόρφωση κατά *ΔL*.



Σχήμα 1.5: Παράδειγμα επιμήκυνσης υλικού.

Η παραμόρφωση (Strain) συμβολίζεται με το ελληνικό πεζό γράμμα έψιλον *ε* και η μονάδα μέτρησης του είναι τα *με* (micro-strain), που είναι $\varepsilon \times 10^{-6}$. Ορίζεται με την σχεση **(1.2)** και ισούται με το λόγο του μήκους της παραμόρφωσης προς το πραγματικό μήκος του σώματος, όταν βρίσκεται σε ηρεμία.

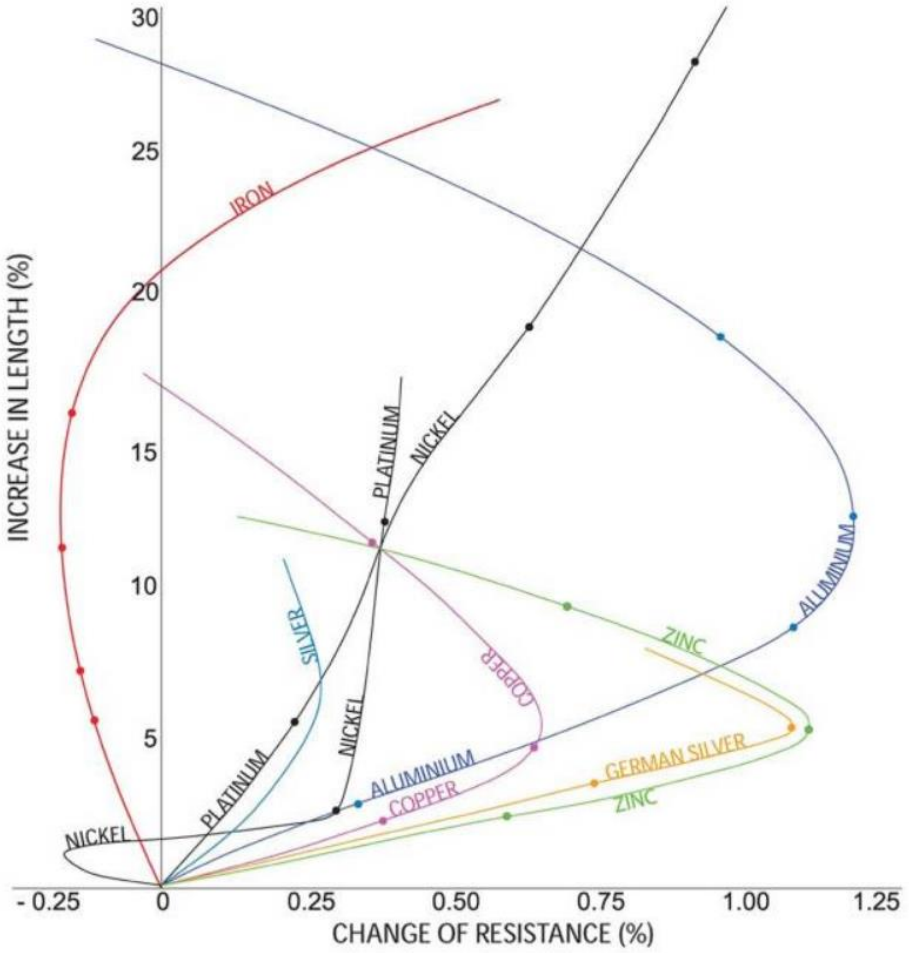
$$\varepsilon = \frac{\Delta L}{L} \tag{1.2}$$

1.3.4 Ευαισθησία στην παραμόρφωση

Ένας ακόμα πολύ σημαντικός συντελεστής ανάμεσα στις αρχές λειτουργιάς του αισθητήρα SG, είναι η ευαισθησία του στην παραμόρφωση. Ονομάζεται συντελεστής παραμόρφωσης, είναι γνωστός ως *Gauge Factor* ή *Strain Factor*, συμβολίζεται με *GF*, δεν έχει μονάδα μέτρησης και ορίζεται από την σχέση **(1.3)**

$$GF = \frac{\frac{\Delta R}{R}}{\frac{\Delta L}{L}} = \frac{\frac{\Delta R}{R}}{\varepsilon} \tag{1.3}$$

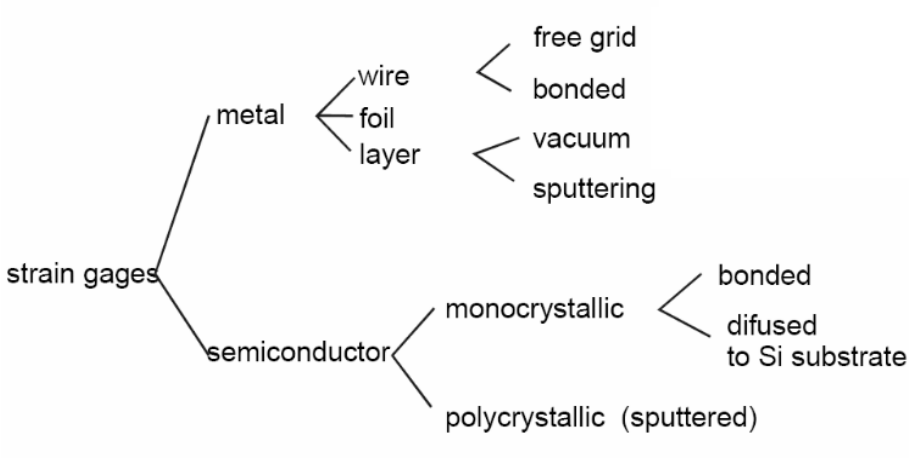
Αναλυτικότερα ως *Gauge Factor* σε έναν αγωγό έχουμε την κλασματική διάφορα της αντίστασης του προς την κλασματική διάφορα του μήκους του ή αλλιώς παραμόρφωση *Strain*. Για τα περισσότερα μέταλλα ο *GF* είναι περίπου 2, ενδεικτικά παρουσιάζονται από μερικά στοιχεία ακόμα στο Σχήμα 1.6.



Σχήμα 1.6: Διάγραμμα αύξησης αντίστασης υλικών για κάθε ποσοστό αύξησης του μήκους τους.

1.4 Δομή αισθητήρα

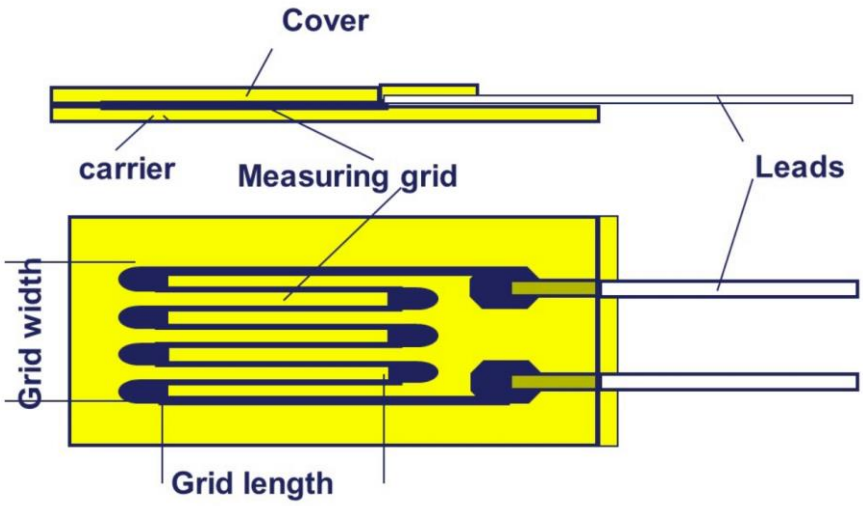
Όταν οι αισθητήρες SG ξεκίνησαν να χρησιμοποιούνται, αποτελούνταν από ένα απλό καλώδιο που προσάρμοζαν και στη συνέχεια ασκούσαν καταπόνηση/πιεση ανάλογα με τις ανάγκες της μέτρησης. Όσο η τεχνολογία προχωρούσε και οι ανάγκες για την εφαρμογή τους γίνονταν πιο απαιτητικές, αναπτύχτηκαν πολλοί διαφορετικοί αισθητήρες SG. Ένα δενδροδιάγραμμα κατηγοριοποίησης του αισθητήρα φαίνεται στο Σχήμα 1.7. Σε γενικές γραμμές χρησιμοποιούνται ακόμα και σήμερα απλά



Σχήμα 1.7: Διάγραμμα κατηγοριοποίησης του SG.

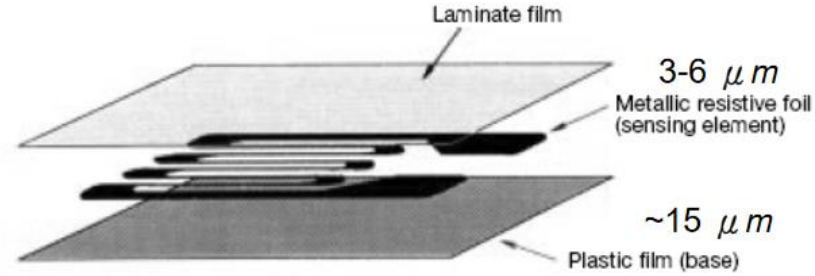
καλώδια για την ανίχνευση της παραμόρφωσης όμως με μερικές διαφοροποιήσεις. Η κατηγορία που έχει επικρατήσει είναι από μεταλλικό φύλλο (*metal foil strain gauges*), καθώς είναι γενικής χρήσης και εύκολο στην εγκατάσταση.

Η πιο συνηθισμένη μορφή SG από μεταλλικό φύλλο φαίνεται στην εικόνα 1.8. Η περιοχή που γίνονται οι μετρήσεις των διακριτικών παραμορφώσεων είναι το πλέγμα που φαίνεται με σκούρο μπλε χρώμα (Measuring grid). Το πλέγμα αποτελείται από πολύ λεπτό φύλλο μέταλλου ή άλλου αγώγιμου υλικού συνήθως 3-6 μ. Ο προσανατολισμός του πλέγματος προσδιορίζει τον άξονα πάνω στον όποιο πραγματοποιείται η παραμόρφωση. Οι μετρήσεις λαμβάνονται από τα ποδαράκια (Leads), αποτελούνται από αγώγιμο υλικό και η σύνδεση τους πραγματοποιείται με την τεχνική συγκόλλησης μέταλλων που χρησιμοποιούμε σε όλες τις πλακέτες PCB.



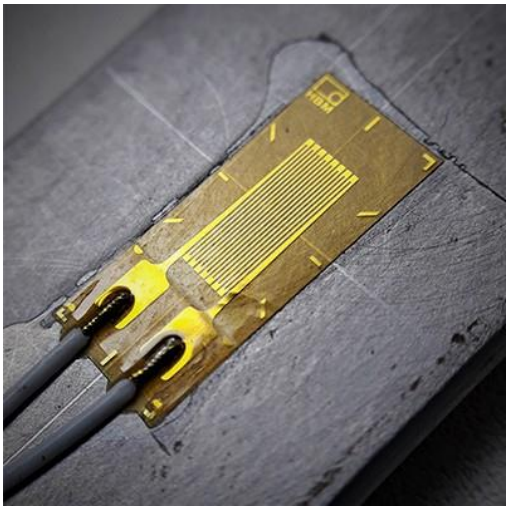
Σχήμα 1.8: Πλάγια όψη (πάνω) και κάτοψη (κάτω) του αισθητήρα SG.

Η κίτρινη περιοχή που φαίνεται είναι το προστατευτικό κάλυμμα (Carrier). Αναλυτικότερα στην εικόνα 1.9 το μεταλλικό πλέγμα βρίσκεται ανάμεσα σε ένα λεπτό φύλλο laminate και ένα λεπτό φύλλο πλαστικής βάσης για την καλύτερη προστασία του αισθητήρα. Ο αισθητήρας εφαρμόζεται με τη βάση να εφάπτεται πάνω στο αντικείμενο που επιθυμούμε να παρατηρήσουμε.



Σχήμα 1.9: Τα τρία στρώματα ενός αισθητήρα SG.

Η κόλληση της βάσης είναι αρκετά σημαντική αφού θα πρέπει να μένει σταθερή στις παραμορφώσεις του αντικείμενου και επιτυγχάνεται με κυανοακρυλικη κόλλα, γνωστή και ως “κόλλα στιγμής”. Επίσης εάν επιθυμούμε να παρατηρήσουμε την ανοχή ενός αντικείμενου, η τοποθεσία θα πρέπει να ελεγχτεί με προσοχή συμφώνα με το πιο επιρρεπή σε παραμόρφωση μέρος του.



Σχήμα 1.10: Αισθητήρας SG όπως εφαρμόζεται σε πραγματικό περιβάλλον.

1.5 Επίλογος

Για αρχή αναφέρθηκαν η ιστορική ανάδρομη και τα πρόσωπα που έπαιξαν μεγάλο ρολό στην υλοποίηση του αισθητήρα SG. Στη συνέχεια έγιναν αναφορές και επεξηγήσεις της φυσικής ικανότητας των υλικών όπως η καταπόνηση ενός υλικού και οι συνέπειες που έχει. Τέλος έγινε ανάλυση της δομής του αισθητήρα και των υλικών που το απαρτίζουν.

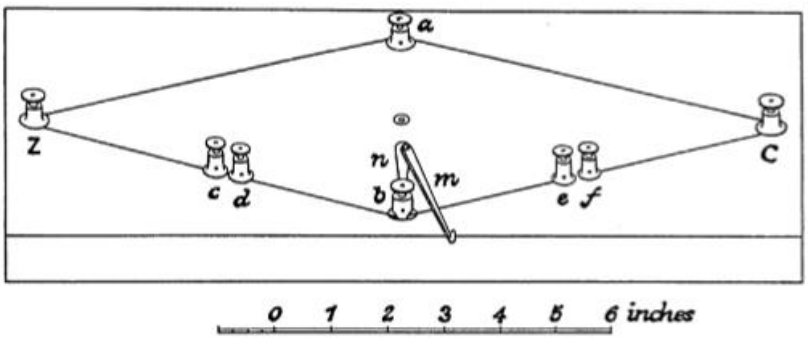
Κεφάλαιο 2ο: Γέφυρα Wheatstone

2.1 Εισαγωγή

Η γέφυρα Wheatstone είναι μια συνδεσμολογία αντιστάσεων που χρησιμοποιείται για την εύρεση άγνωστων αντιστάσεων. Η βασική δουλεία του κυκλώματος είναι να συγκρίνει τις άγνωστες αντιστάσεις με τις γνωστές. Η δυνατότητα της γέφυρας να βρίσκει πολύ μικρές διαφορές αντίστασης ανάμεσα στην συνδεσμολογία της, την έχουν καταστήσει βασικό εργαλείο μέτρησης των αισθητήριων SG. Αποτελεί το όργανο μέτρησης του κυκλώματος που υλοποιήθηκε για την παρούσα πτυχιακή εργασία.

2.2 Ιστορική αναδρομή

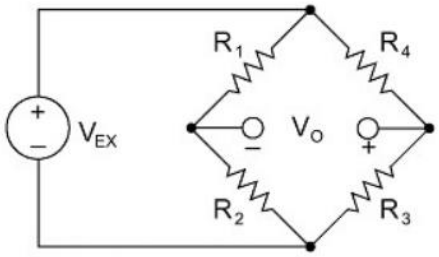
Όλες οι ιστορικές αναφορές που υπάρχουν στο προηγούμενο κεφάλαιο για τους αισθητήρες SG, πραγματοποιήθηκαν σε συνδεσμολογία γέφυρας Wheatstone. Η γέφυρα εφευρέθηκε από δυο βρετανούς επιστήμονες που δούλευαν ανεξάρτητα ο ένας από τον άλλο και με διάφορα μιας δεκαετίας. Και οι δυο είχαν σκοπό να βρουν τις τιμές αντιστάσεων, ανάμεσα τους και καλωδίων. Πρώτος θεωρείται ο Samuel Hunter Christie που το 1833 δημοσίευσε πληροφορίες για το κύκλωμα του χωρίς ιδιαίτερη ανταπόκριση. Μια δεκαετία αργότερα, το 1843 ο Charles Wheatstone εφευρέτης του ροοστάτη, δημοσιεύει την ερευνά του για το κύκλωμα της γέφυρας χαρακτηρίζοντας το "διαφορικός μετρητής αντιστάσεων". Παρόλο που έδωσε τα εύσημα και στον Samuel, η ευρεσιτεχνία πηρέ το όνομα της από τον Wheatstone. Η θέση του μέσα στο Royal Society of London του επέτρεψε να προωθήσει την γέφυρα και να της δώσει δημοσιότητα. Επίσης κατάφερε να αναπτύξει την συνδεσμολογία για την μέτρηση άγνωστης συχνότητας, χωρητικότητας και άλλων ιδιοτήτων.



Σχήμα 2.1:Σχεδιο πρότυπου γέφυρας μέτρησης αντιστάσεων όπως υλοποιήθηκε από τον Wheatstone.

2.3 Αρχές λειτουργίας

Όπως αναφέρθηκε στο προηγούμενο κεφάλαιο, όταν υπάρχει παραμόρφωση οι αισθητήρες SG παρουσιάζουν διάφορα αντίστασης της τάξης των $10^{-3}\Omega$ και για την ανιχνευση τους χρησιμοποιειται κυκλωμα αντιστάσεων Wheatstone. Η γενική μορφή του κυκλώματος παρουσιάζεται στην εικόνα 2.2. Αποτελείται από ένα δίκτυο τεσσάρων αντιστατών με τροφοδοσία τάσης V_{EX} και έξοδο V_O.



Σχήμα 2.2: Γενική μορφή γέφυρας Wheatstone.

Αναλύοντας το κύκλωμα παρατηρούμε πως αποτελείται από δυο διαιρέτες τάσης. Ο πρώτος είναι οι αντιστάσεις R_1, R_2 και ο δεύτερος οι αντιστάσεις R_3, R_4. Η έξοδος, που πραγματοποιείται και η μέτρηση του κυκλώματος παρουσιάζεται ανάμεσα στους δυο κόμβους που βρίσκονται στη μέση του κάθε διαιρετή τάσης.

$$V_O = V_{EX}\left(\frac{R_3}{R_4+R_3}\right) - V_{EX}\left(\frac{R_2}{R_1+R_2}\right) \quad \textbf{(2.1)}$$

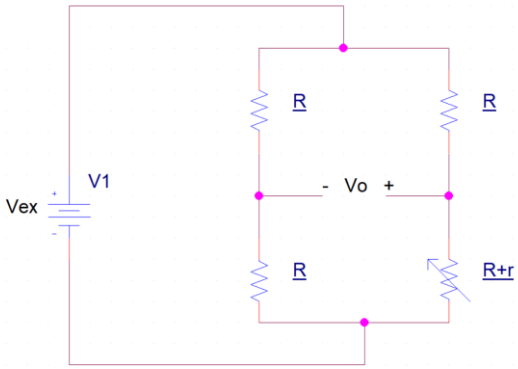
$$V_O = V_{EX}\left[\frac{R3}{R4+R3} - \frac{R2}{R1+R2}\right] \quad \textbf{(2.2)}$$

Από την παραπάνω εξίσωση (2.1) είναι κατανοητό πως αν $\frac{R_1}{R_2} = \frac{R_4}{R_3}$ **(2.3)** τότε η τάση εξόδου στα άκρα της V_{EX} ειναι μηδενικη. Συνεπως η γεφυρα βρισκεται σε ισορροπία. Σε καταστάση ισορροπίας καθιστά πιο εύκολη την εύρεση της άγνωστης αντίστασης αφού με βάση την εξίσωση (2.3) συνεπάγεται πως $I_{1,2} = I_{3,4}$ **(2.4)**. Έτσι αν για παράδειγμα η αντίσταση R_4 είναι άγνωστη, μπορεί να βρεθεί από την εξίσωση $R_4 = R_3\left(\frac{R_1}{R_2}\right)$. Αν παλι η τιμη απο οποιαδηποτε αντίσταση αλλάξει ελάχιστα τότε η έξοδος της γέφυρας θα είναι μη μηδενική.

2.3.1 Γέφυρα ¼ (Quarter bridge)

Αντικαθιστώντας μια από τις αντιστάσεις με ενεργό SG μεταβάλουμε την αντίσταση του αισθητήρα κατά r, όπου $r < \frac{1}{10}R$.Ετσι σε καταστάση ηρεμιας οι αντιστασεις $R_1 = R_2 = R_4 = R$ και $R_3 = R + r$. Η εξισωση (2.2) θα γινει

$$V_O = V_{EX}\frac{r}{4R+2r} \quad \textbf{(2.5)}$$



Σχήμα 2.3: Σχηματικό γέφυρας Wheatstone ¼ .

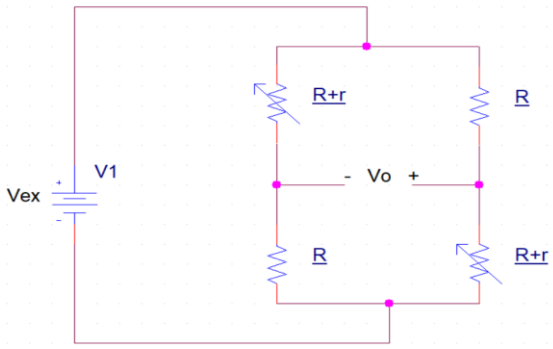
2.3.2 Γέφυρα ½ (Half bridge)

Η γέφυρα αντιστάσεων ½ είναι δυο φορές πιο ευαίσθητη από την γέφυρα ¼. Αποτελείται από δυο αντιστάσεις σταθερές και όσες, και από δύο που μεταβάλλονται είτε όμοια είτε αντίθετα.

Μεταβολή αντιστάσεων με ίδια πολικότητα

Μεταβαλλόντας ταυτόχρονα δυο αντιστάσεις με την ιδία πολικότητα έχουμε $R_2 = R_4 = R$ και $R_1 = R_3 = R + r$. Όποτε η εξίσωση (2.2) θα γίνει

$$V_O = V_{EX} \frac{r}{2R+r} \qquad \textbf{(2.6)}$$

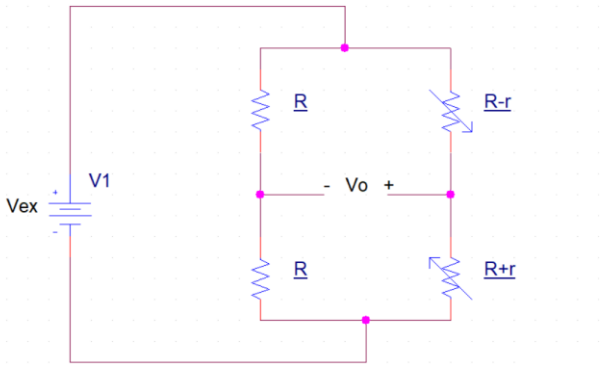


Σχήμα 2.4: Σχηματικό γέφυρας Wheatstone ½ με την ιδία πολικότητα.

Μεταβολή αντιστάσεων με διαφορική πολικότητα

Μεταβαλλόντας ταυτόχρονα δυο αντιστάσεις με διαφορική πολικότητα έχουμε $R_1 = R_2 = R$ και $R_3 = R + r$ και $R_4 = R - r$. Οποτε η εξισωση (2.2) θα γινει

$$V_O = V_{EX}\frac{r}{2R} \quad (2.7)$$

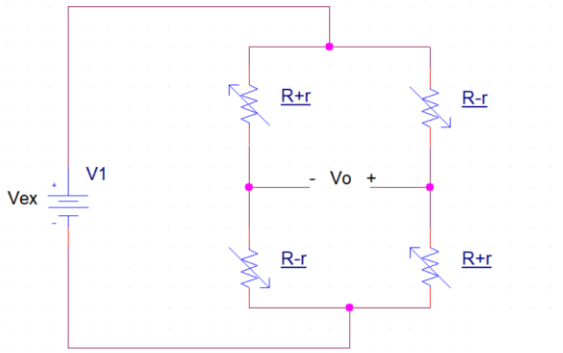


Σχήμα 2.5: Σχηματικό γέφυρας Wheatstone ½ με διαφορική πολικότητα.

2.3.3 Γέφυρα 4/4 (Full bridge)

Τέλος είναι η πλήρη γέφυρα ή αλλιώς 4/4, η όποια περιέχει τα λιγότερα σφάλματα σε σχέση με τις άλλες τρεις, περιέχει τέσσερις αισθητήρες SG και κατ επέκταση τέσσερις μεταβαλλόμενες αντιστάσεις κατά r. Με $R_1 = R_3 = R + r$ και $R_2 = R_4 = R - r$ η εξισωση (2.2) θα γινει

$$V_O = V_{EX}\frac{r}{2R} \quad (2.8)$$



Σχήμα 2.6: Σχηματικό πλήρους γέφυρας Wheatstone.

2.3.4 Επίλογος

Στο κεφάλαιο επισημάνθηκε η ιστορική ανάδρομη της γέφυρας και πάνω σε ποιες αρχές βασίστηκε και στη συνέχεια έγινε η ανάλυση της εστιάζοντας στον τρόπο που λειτούργει και τις τέσσερις κατηγορίες που χωρίζεται.

Κεφάλαιο 3ο: Ενισχυτής οργανολογίας

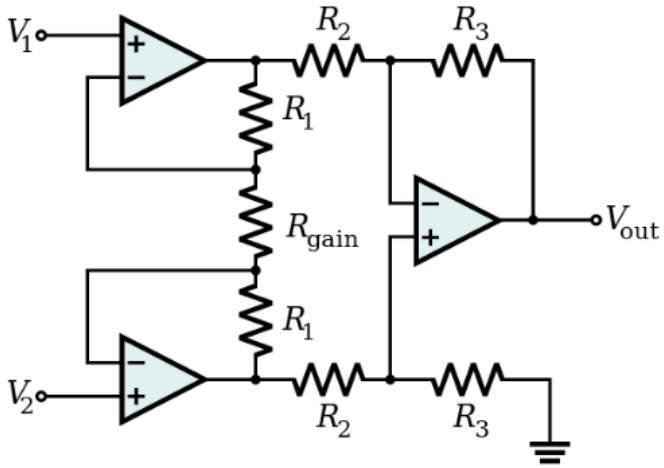
3.1 Εισαγωγή

Ανάμεσα στους ενισχυτές, που χρησιμοποιούνται στα ηλεκτρονικά κυκλώματα, ξεχωρίζουν οι ενισχυτές οργανολογίας. Η αγγλική τους ονομασία είναι “Instrumentation amplifier” και αρκετά συχνά αναφέρονται με την συντομογραφία “in-amp”. Βρίσκουν μεγάλη χρησιμότητα σε βιομηχανικές εφαρμογές, σε κυκλώματα λήψης μετρήσεων που υπάρχουν μεγάλα κοινά σήματα και σε πολύ θορυβώδη κυκλώματα που τα DC σήματα και η ενίσχυση τους πρέπει να διατηρηθούν με μεγάλη ακρίβεια.

3.2 Αρχές λειτουργίας

Παρά το γεγονός ότι ένας ενισχυτής οργανολογίας στη γενική του μορφή αποτελείται από τρεις τελεστικούς ενισχυτές, θα λέγαμε ότι παρουσιάζει κάποιες σημαντικές διαφορές σε σχέση με έναν τελεστικό ενισχυτή. Για παράδειγμα ένας τελεστικός ενισχυτής θεωρείται ένα γενικής χρήσης εξάρτημα ενίσχυσης που ο χρήστης μπορεί να συνδυαστεί με στοιχειά αντιστάσεων και πυκνωτών ώστε να αλλάξει τη λειτουργιά του. Ενώ ο ενισχυτής οργανολογίας έχει πολύ καθορισμένη λειτουργιά και περιορισμένη ενίσχυση. Η λειτουργιά του αφορά τη λήψη μετρήσεων από αισθητήρες όπως SG, την ενίσχυση του διαφορικού σήματος τους, και την αποκοπή μεγάλου φάσματος κοινού σήματος θορύβων που παράγουν συνήθως οι αισθητήρες και τα υλικά σε ένα κύκλωμα.

Σε γενικές γραμμές ένας ενισχυτής οργανολογίας είναι κλειστού βρόγχου και μεγάλης ακριβείας. Αποτελείται από τρεις τελεστικούς ενισχυτές, οι δυο σε συνδεσμολογία ακόλουθου τάσης (Buffer) και ο τρίτος σε διαφορικού ενισχυτή. Όπως φαίνεται και στο σχήμα 4.1, στις εισόδους του ενισχυτή οργανολογίας V_1, V_2 το σημα λαμβανεται διαφορικα από τους δυο ακολουθους και στη συνεχεια ενισχυεται με τον διαφορικο ενισχυτη αναλογα με την τιμη της αντιστασης R_{gain}. Τελος το ενισχυμενο πλεον διαφορικο σημα βρισκει εξοδο στο σημειο V_{out}.



Σχήμα 3.1: Σχηματικό ενισχυτή οργανολογίας με τρεις διαφορικούς ενισχυτές και το εσωτερικό ενός IC ενισχυτή οργανολογίας.

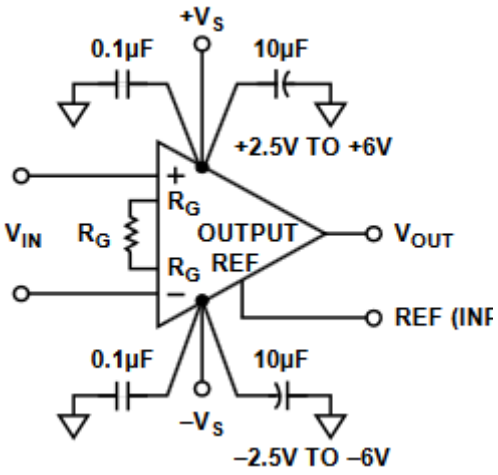
Η εξίσωση κέρδους A_V (ενίσχυσης) του παραπάνω ενισχυτή οργανολογίας καθώς και η εξίσωση τάσης εξόδου V_{OUT} φαίνεται παρακάτω.

$$A_V = \left(1 + \frac{2R_1}{R_{gain}}\right)\frac{R_3}{R_2} \quad \textbf{(3.1)}$$

$$V_{OUT} = (V_2 - V_1) \times A_V \quad \textbf{(3.2)}$$

3.3 Χαρακτηριστικά ενισχυτή οργανολογίας

Για να πραγματοποιηθεί η συνδεσμολογία με τη μεγαλύτερη δυνατή απόδοση του ενισχυτή οργανολογίας με τρεις τελεστικούς ενισχυτές χρειάζονται είτε τρία διαφορετικά ολοκληρωμένα κυκλώματα (IC) σε συνδυασμό με τουλάχιστον επτά αντιστάσεις, είτε ένα ολοκληρωμένο κύκλωμα με μια αντίσταση για την ρύθμιση της ενίσχυσης. Για την καλύτερη δυνατή ανάλυση θα αναφερόμαστε στην εικόνα 4.2.



Σχήμα 3.2: Σχηματικό AD623 διπλής τροφοδοσίας.

3.3.1 Τροφοδοσία

Η τροφοδοσίας ενός ολοκληρωμένου κυκλώματος ενισχυτή οργανολογίας χωρίζεται σε δυο κατηγορίες:

- Μονή τροφοδοσία (Single Supply)
  Δηλαδή τροφοδοτείται από μόνο μια πηγή είτε θετική είτε αρνητική, αναλόγως την περιοχή λειτουργιάς του.

- Διπλή τροφοδοσία (Dual Supply)

Δηλαδή η τροφοδοσία του πραγματοποιείται από δυο πήγες εκ των οποίων η μια θα είναι θετική και η άλλη αρνητική.

Ορίζεται από τον κατασκευαστή και οριοθετεί την περιοχή λειτουργιάς του ενισχυτή.

3.3.2 Είσοδος

Η είσοδος του είναι διαφορική και έχει δυο ποδαράκια (pins), την αναστρέψιμη (V_{IN-}) και τη μη αναστρεψιμη (V_{IN+}). Οι δυο εισοδοι παρουσιάζουν πολύ υψηλή συνθετη αντισταση (impedance) και σχεδον ίση η μια με την άλλη ώστε να μην πεσει το σήμα πηγής εισόδου που θα επηρεάσει αρνητικά και την τάση σήματος εισόδου. Συνήθως η σύνθετη αντίσταση κυμαίνεται από $10^9\Omega$ μεχρι $10^{12}\Omega$ και τα ρεύματα πόλωσης από $1nA$ έως $50nA$.

3.3.3 Έξοδος και τάση αναφοράς

Η αντίσταση εξόδου του ενισχυτή οργανολογίας συνήθως είναι μόλις μερικά ω. Το σήμα εξόδου V_{OUT} μπορει να πατησει πανω στην τάση αναφοράς που συνδεσμολογείται στο ποδαρακι *REF* με ανεξάρτητο κύκλωμα ώστε σε έναν ενισχυτή μόνης τροφοδοσίας να εμφανίζεται στην μέση της περιοχής λειτουργιάς του καθώς μπορεί να είναι της τάξης μερικών *mV*. Άλλη επιλογή είναι να το συνδέουμε στην γείωση όταν δεν το χρειαζόμαστε ή με οποιοδήποτε άλλο τρόπο μας ορίσει ο κατασκευαστής.

3.3.4 Ενίσχυση σήματος

Η ενίσχυση του σήματος μπορεί να ρυθμιστεί είτε επιλέγοντας μια εσωτερική αντίσταση με τη βοήθεια του κατασκευαστή, είτε επιλέγοντας μια αντίσταση εξωτερική στα ποδαράκια R_G. Σε αντίθεση με τα κυκλώματα τριών τελεστικών ενισχυτών, το ολοκληρωμένο κύκλωμα ενισχυτή οργανολογίας έχει περιορισμένο εύρος τιμών ενίσχυσης.

3.3.5 Λόγος απόρριψης κοινού σήματος

Προκειμένου να θεωρείται αποτελεσματικό ένα ολοκληρωμένο κύκλωμα ενισχυτή οργανολογίας θα πρέπει να είναι σε θέση να ενισχύει σήματα της τάξης των $10^{-6}V$ ενώ ταυτόχρονα απορρίπτει θορύβους και κοινά σήματα που παρουσιάζονται στις δυο εισόδους του. Αυτό προϋποθέτει ότι τα ολοκληρωμένα κυκλώματα αυτά θα έχουν πολύ υψηλό λόγο απόρριψης κοινών σημάτων (CMRR). Τυπικές τιμές είναι από *70dB* μέχρι και περισσότερο από *100dB*. Αξίζει να αναφέρουμε πως σε βιομηχανικές εφαρμογές δεν είναι τα DC σήματα που παράγουν θόρυβο αλλά τα AC σε συχνότητα *50/60Hz*. Έτσι είναι σημαντικό να ορίζεται και η συχνότητα του λόγου απόρριψης κοινών σημάτων εκτός από την DC τιμή του.

3.4 Επίλογος

Κλείνοντας το κεφάλαιο αυτό περιγράφονται οι αρχές λειτουργιάς του ενισχυτή οργανολογίας και αναλύθηκαν οι τύποι που μπορεί να υπολογιστεί το κέρδος του και η τιμή της εξόδου του. Και στη συνέχεια έγινε ανάλυση των χαρακτηριστικών που κάνουν έναν ενισχυτή οργανολογίας απαραίτητο στη χρήση.

Κεφάλαιο 4ο: ESP32

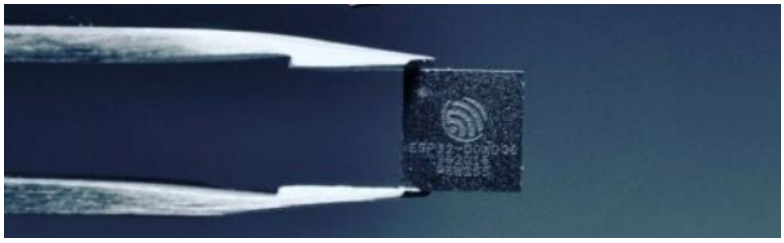
4.1 Εισαγωγή

ESP32 είναι το όνομα ενός μικροελεγκτή (microcontroller) που σχεδιάστηκε από την Espressif Systems, μια κινέζικη εταιρεία που εδρεύει έξω από την Σαγκάη. Ο μικροελεγκτής θεωρείται διάδοχος του ESP8266 και μια πολύ καλή λύση σύνδεσης των ήδη υπάρχων μικροελεγκτών με δίκτυο Wi-Fi. Η παραγωγή του ξεκίνησε στα τέλη του 2016 από την TSMC χρησιμοποιώντας *40nm* μέθοδο κατασκευής chip. Με την πρόοδο νέων τεχνολογιών καθιερώθηκε σαν μια σειρά μικροελεγκτών πάνω σε ολοκληρωμένο κύκλωμα (μικροϋπολογιστής), χαμηλού κόστους, μικρής κατανάλωσης, με ενσωματωμένο Wi-Fi και δυο λειτουργιές Bluetooth. Η σειρά μικροελεγκτών ESP32 έχει βασιστεί πάνω στα παρακάτω χαρακτηριστικά

Χαρακτηριστικά	Λεπτομέρειες
Τάση λειτουργίας	3.3V
Κατανάλωση ρεύματος	20 μA-260mA (εξαρτάται από την λειτουργία)
Μνήμη flash	Προσκολλημένη πάνω στη μονάδα (module)
Επεξεργαστής	Tensilica LX6 32-bit
Ταχύτητα επεξεργαστή	Διπύρηνος στα 160MHz (80-240MHz)
RAM	520K
GPIO	34
ADC (Analog to Digital Converter)	7
Υποστήριξη 802.11	11b/g/n/e/i
Bluetooth	BLE και κλασικό Bluetooth
Μέγιστος αριθμός ταυτόχρονων συνδέσεων TCP	16
SPI	3
I^2S	2
I^2C	2
UART	3

Πίνακα 4.1: Πινάκας βασικών προδιαγραφών και υποστηριζόμενων τεχνολογιών του ESP32.

Αξιοσημείωτο χαρακτηριστικό είναι ο χρόνος λειτουργιάς του μικροελεγκτή με μπαταριές, μιας και πολλές εφαρμογές του είναι φορητές. Πράγμα το όποιο εξαρτάται από την κατάσταση λειτουργιάς του, όποτε και η κατανάλωση ρεύματος είναι κάθε άλλο παρά σταθερή. Η κατάσταση λειτουργιάς του μικροελεγκτή ρυθμίζεται από τον προγραμματισμό του, κοινώς το πρόγραμμα που θα φορτώσουμε στην μνήμη flash η όποια είναι ξεχωριστό εξάρτημα προσκολλημένο στη μονάδα (module) που θα επιλέξει ο καταναλωτής.



Σχήμα 4.1: Ένδειξη μεγέθους μικροελεγκτή ESP32-D0WDQ6.

4.2 ESP-WROOM-32

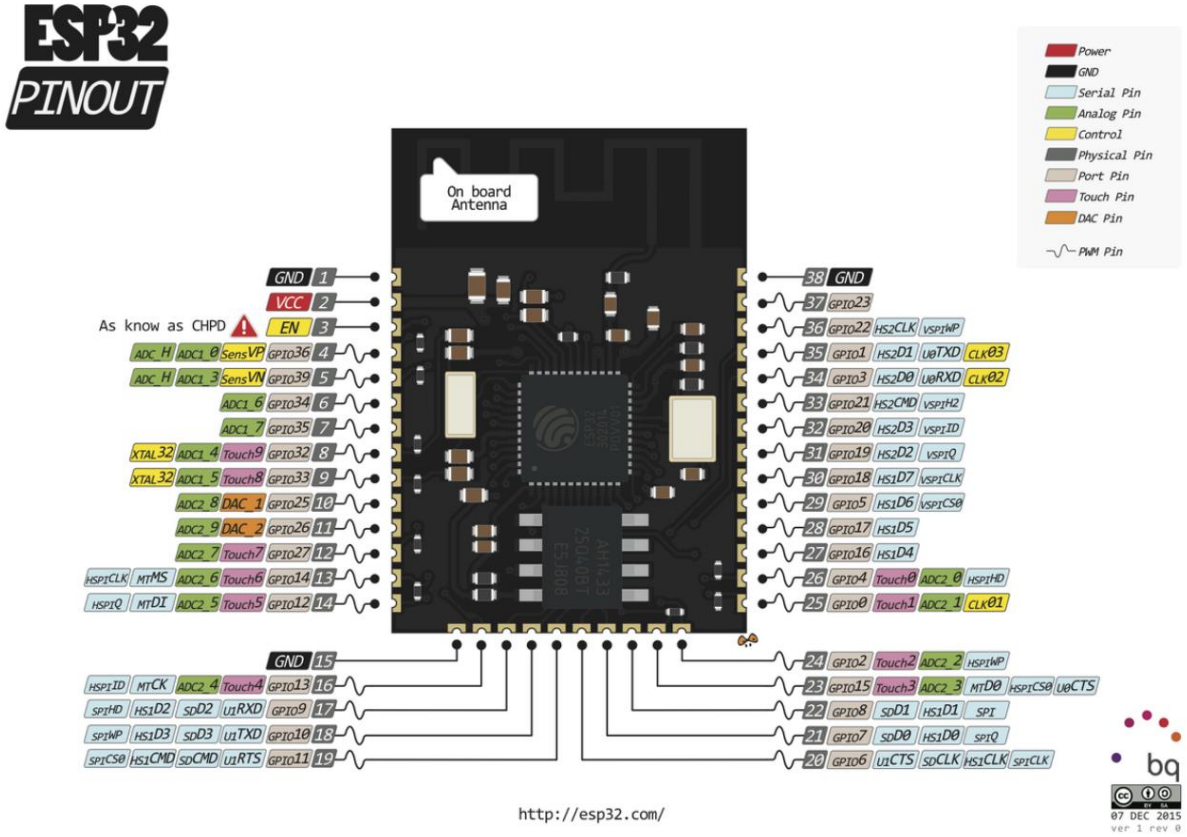
Καθώς οι διαστάσεις του μικροελεγκτή ESP32 στα *5x5mm* δεν το καθιστούν φιλικό προς χρήση από έναν άνθρωπο, διάφορες εταιρίες κατασκευάζουν μονάδες που φέρουν πάνω τους τον μικροελεγκτή.

Σχήμα 4.2: Κάτοψη ESP-WROOM-32.

Μια από αυτές τις μονάδες είναι η ESP-WROOM-32 κατασκευασμένη από την Espressif με έξτρα hardware όπως μνήμη Flash *4 MB* ή κεραία. Παρά το ότι μονάδες σαν κι αυτή δεν είναι κατάλληλες για τη δημιουργία πειραμάτων και κατασκευών πάνω σε ράστερ (breadboard), χρησιμοποιείται πιο εύκολα από εταιρίες και ερευνητές με καλές δεξιότητες στα ηλεκτρονικά.

Στο κέντρο της μονάδας (εικόνα 4.3) φαίνεται το chip ESP32-D0WDQ6. Με χαρακτηριστικά όπως διπύρηνο επεξεργαστή ρυθμιζόμενης συχνότητας *80-240MHz* και δευτερεύον επεξεργαστή χαμηλής κατανάλωσης που μπορεί να χρησιμοποιηθεί αντί του κύριου για εργασίες που δεν απαιτούν μεγάλη υπολογιστική δύναμη. Δυνατότητες όπως Bluetooth, Bluetooth LE και Wi-Fi το καθιστούν κατάλληλο για την υλοποίηση πολλών διαφορετικών εφαρμογών και επιπλέον μπορούν να συνδυαστούν περισσότερες από μια δυνατότητες. Για παράδειγμα λήψη δεδομένων από το Internet με χρήση Wi-Fi και παράλληλη μεταφορά τους στο κινητό μέσω Bluetooth.

Το λειτουργικό του σύστημα είναι freeRTOS με ενσωματωμένη λειτουργιά επιτάχυνσης του hardware και αναβάθμισης.



Σχήμα 4.3: Σχηματικό λειτουργιάς κάθε pin του ESP-WROOM-32.

4.2.1 Γενικές πληροφορίες

Στον πινάκα 4.2 φαίνονται οι βασικές πληροφορίες του ESP32-WROOM-32.

Κατηγορίες	Αντικείμενα	Προδιαγραφές
Πιστοποιητικά (Certifications)	RF	FCC/CE-RED/IC/TELEC/KCC/SRRC/NCC
	Wi-Fi	Wi-Fi Alliance
	Bluetooth	BQB
	Green	RoHS/REACH
Test	Reliability	HTOL/HTSL/uHAST/TCT/ESD
Wi-Fi	Πρωτόκολλα	802.11 b/g/n (80.211n μέχρι και 150Mbps)
		A-MPDU και A-MSDU
	Συχνότητα	2.4~2.5 GHz
Bluetooth	Πρωτόκολλα	Bluetooth v4.2 BR/EDR και BLE
	Radio	NZIF δέκτης με ευαισθησία -97dBm
		Class-1, Class-2, Class-3 πομπός
		AFH
	Audio	CVSD και SBC

Hardware	Περιφερειακά	• SD card • UART • SPI • SDIO • I^2C • LED PWM • Motor PWD • I^2S	• IR • μετρητής παλμού • GPIO • capacitive touch sensor • ADC • DAC • TWAI • RMII
	Αισθητήρας πάνω στο chip	Hall sensor	
	Κρύσταλλος	40 MHz	
	SPI flash	4 MB	
	Τάση λειτουργιάς/ Τροφοδοσία	3.0 V ~ 3.6 V	
	Ρεύμα λειτουργιάς	80 mA	
	Ελάχιστο ρεύμα τροφοδοσίας	500 mA	
	Θερμοκρασία λειτουργιάς	-40 ^oC ~ +85 ^oC	
	Διαστάσεις μονάδας	18.00±0.10 mm × 25.50±0.10 mm × 3.10±0.10 mm	
	Ευαισθησία υγρασίας	Level 3	

Πίνακας 4.2: Πινάκας πληροφοριών ESP32-WROOM-32

4.3 ESP32 DEVKIT V1 DOIT Board

Καθώς το ESP32-WROOM-32 ή ανάλογες μονάδες δεν είναι εύχρηστες για εφαρμογές από άτομα που ασχολούνται σε ποιο ερασιτεχνικό επίπεδο με τα ηλεκτρονικά και τους μικροελεγκτής ή άτομα που δε διαθέτουν τον ανάλογο εξοπλισμό για την σύνθεση τους σε κύκλωμα, διάφοροι κατασκευαστές έβγαλαν μικροελεγκτής της σειράς ESP32 πάνω σε πλακέτες (boards).

Σχήμα 4.4: Διαφορές πλακέτες που φέρουν πάνω τους μικροελεγκτή της σειράς ESP32.

Οι πλακέτες αυτές εφαρμόζονται πιο εύκολα πάνω σε ένα ράστερ για δόκιμες και διαθέτουν εργαλεία που κάνουν τους μικροελεγκτής πιο εύκολους στη χρήση. Παρά τις επιλογές που διαθέτουμε όπως φαίνονται στην εικόνα 4.4 μερικές, θα εστιάσουμε πάνω στην πλακέτα ESP32 DEVKIT V1 DOIT, που φέρει εργαλεία για τη σταθεροποίηση της τάσης εισόδου και τη σύνδεση του με υπολογιστή μέσω USB.

4.3.1 Βασικά εξαρτήματα

Με μια πρώτη μάτια στο ESP32 DEVKIT V1 DOIT της εικόνας 4.5 ξεχωρίζουν κάποια βασικά εξαρτήματα.

Σχήμα 4.5: Κάτοψη του ESP32 DEVKIT V1 DOIT.

WROOM-32

Όπως αναλύσαμε και σε προηγούμενο κεφάλαιο πρόκειται για μια μονάδα που φέρει πάνω της τον μικροελεγκτή της σειράς ESP32 και τον συνθέτει με εργαλεία όπως κεραία με ρυθμό μεταφοράς δεδομένων έως *150Mbps* και μνήμη flash *4 MB* που δίνει την δυνατότητα επαναπρογραμματισμού έως και 100.000 φορές.

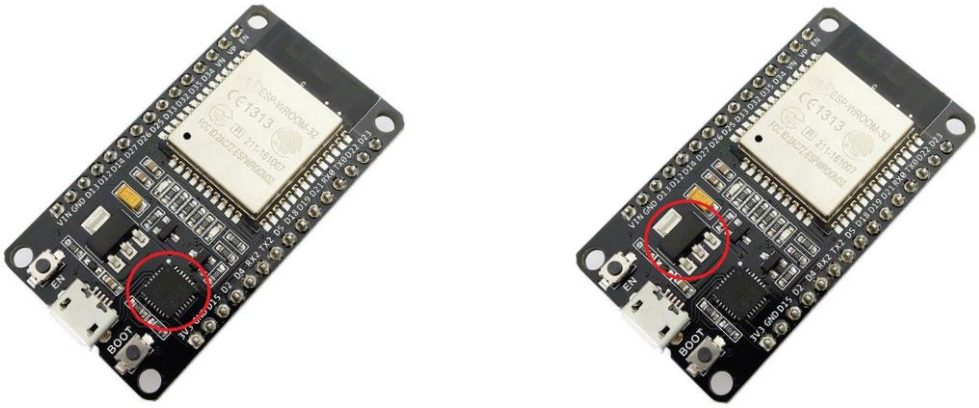
Σχήμα 4.6: Ένδειξη τοποθεσίας μικροελεγκτή ESP-WROOM-32

AMS1117

Από τα χαρακτηριστικά των μικροελεγκτών της σειράς ESP32 είδαμε πως για την λειτουργιά τους χρειάζεται τροφοδοσία DC τάσης στα *3.3V*. Την ενέργεια αυτή έχει αναλάβει ο σταθεροποιητής τάσης (voltage regulator) *AMS1117T33*. Πρόκειται για μια σειρά ρυθμιζόμενων σταθεροποιητών τάσης που παρέχουν μεχρι *1A* ρεύμα εξόδου και μεγίστη διάφορα τάσης εισόδου-εξόδου *1.3V*. Οι σταθερές τιμές που μπορεί να ρυθμίσει ο χρήστης μετά από συνδεσμολογία ενός εξωτερικού κυκλώματος αντιστάσεων είναι στα *1.5V, 1.8V, 2.5V, 2.85V, 3.3V* και *5.0V*.

CP2102

Η σύνδεση του ESP32-WROOM-32 με έναν υπολογιστή και κατ' επέκταση ο προγραμματισμός του πραγματοποιείται μέσω του ενσωματωμένου κυκλώματος UART που διαθέτει. Το κοινό που στοχεύει όμως η παραγωγή του ESP32 δεν είναι τόσο εξοικειωμένο με την τεχνολογία UART όσο με το USB και προσκόλλησε στην πλακέτα το CP2102 που φαίνεται και στην εικόνα 4.7. Είναι ένα chip *5x5 mm* που δρα σαν γέφυρα μεταξύ USB και UART εξασφαλίζοντας μια εύκολη λύση όσων μικροελεγκτών δουλεύουν με το πρότυπο σειριακής επικοινωνίας RS-232.



Σχήμα 4.7: Ένδειξη τοποθεσίας CP2102(δεξιά) και AMS1117(αριστερά) πάνω στον ESP32.

4.4 Εφαρμογές

Μερικά παραδείγματα που ο μικροελεγκτής έχει βρει εφαρμογή είναι:

- Α αισθητήρας χαμηλής κατανάλωσης
- Α Data Logger χαμηλής κατανάλωσης
- Αναγνώριση φωνής
- Αναγνώριση εικόνας
- Αυτοματισμοί σε έξυπνα σπίτια
- Έξυπνες πρίζες
- Έλεγχος φωτισμού
- Έξυπνος έλεγχος πύλης
- Διαχείριση ενεργείας εργοστάσιου
- Έξυπνος φωτισμός
- Βιομηχανικοί αυτοματισμοί όπως στη γεωργία
- Wearable electronics
- Αυτοματισμοί υγείας όπως έξυπνο πάτωμα

4.5 Επίλογος

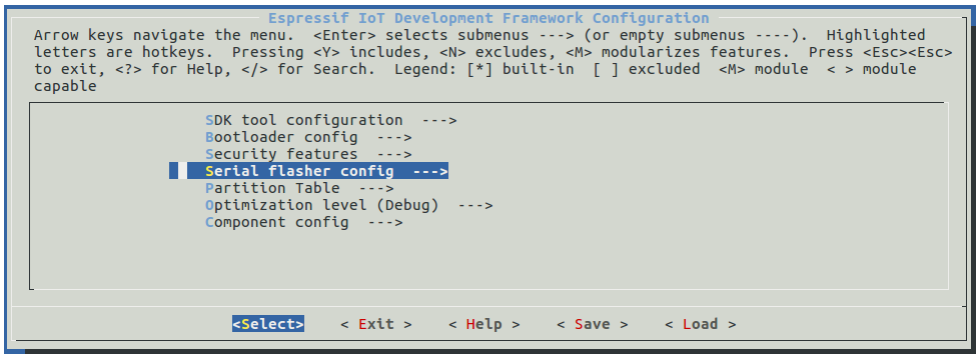
Για αρχή σε αυτό το κεφάλαιο αναλύθηκε ο μικροελεγκτής ESP-WROOM-32 και αναφέρθηκαν τα κυρία χαρακτηριστικά του και πως το μέγεθος του κρίνει απαραίτητη την εφαρμογή του πάνω σε μια πλακέτα όπως την ESP32 DEVKIT V1 DOIT. Όπου και παρουσιαστήκαν οι επιπρόσθετες λειτουργιές που έχει καθώς και τα εξαρτήματα που το καθιστούν έτοιμο για χρήση από έναν ερασιτέχνη στο σπίτι του.

Κεφάλαιο 5ο: Arduino IDE

5.1 Εισαγωγή

Ο ESP32 μας δίνει τη δυνατότητα να γράψουμε, να αποθηκεύσουμε και να τρέχουμε προγράμματα μέσα από την μνήμη του. Μέσα από μια διαδικασία που ονομάζεται “flashing” μεταφέρουμε τον κώδικα που γράψαμε σε γλωσσά προγραμματισμού, στη μνήμη του μικροελεγκτή. Ο κώδικας μεταφράζεται (compile) σε γλωσσά κατανοητή προς τον ESP32 έτσι ώστε το πρόγραμμα μας να μπορεί να αλληλεπιδράσει με το περιβάλλον του μικροελεγκτή. Στον κώδικα μπορούμε με μερικές έτοιμες συναρτήσεις να χρησιμοποιήσουμε τις λειτουργιές και τα περιφερειακά του ESP32. Για παράδειγμα αν χρειαστεί σε κάποια εφαρμογή η λειτουργιά εύρεσης και σύνδεσης σημείου Wi-A ο ESP32 έχει έτοιμες συναρτήσεις, καταγεγραμμένες στην επίσημη σελίδα του και χωρίς να χρειάζεται να τις θυμόμαστε όλες αρκεί μόνο να γνωρίζουμε την ύπαρξη τους και να ανατρέχουμε στα εγχειρίδια χρήσης.

Η εταιρία Espressif που σχεδίασε τον μικροελεγκτή έχει δημιουργήσει ένα περιβάλλον εργασίας ή αλλιώς framework που βοηθού σε ενέργειες όπως τη σύνδεση του με τον υπολογιστή και τον προγραμματισμό του. Το framework ονομάζεται Espressif IDF όπου Espressif είναι το όνομα της εταιρίας και IDF βγαίνει από το “Internet-of-Things Development Framework”. Το framework της



Σχήμα 5.1: Παράδειγμα ρύθμισης μικροεπεξεργαστή με το Espressif IDF.

Espressif όμως δεν είναι το μοναδικό περιβάλλον και οι γλώσσες προγραμματισμού ποικίλουν. Μερικά παραδείγματα γλωσσών είναι:

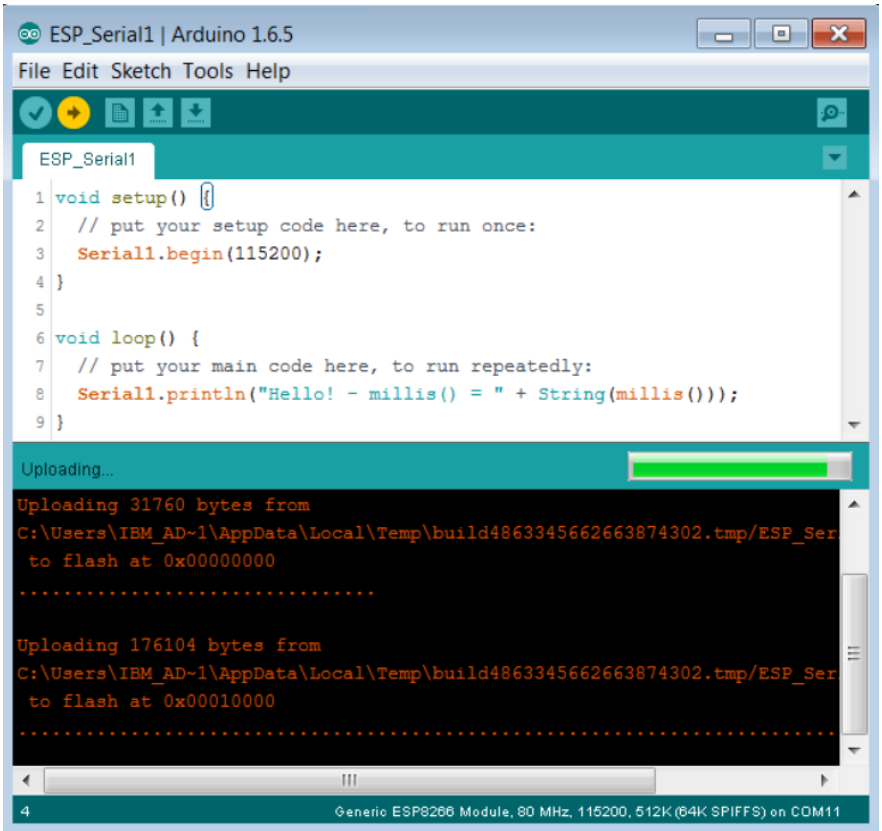
Γλώσσα προγραμματισμού	Συνήθεις λόγοι χρήσης.
MicroPython	Γραμμένη σε C και βασισμένη πάνω στην Python3, κατάλληλη για μικροελεγκτής
Javascript	Σημαντική για την δημιουργία WebApps, συνήθως client-side
Lua	Γενικής χρήσεως γλωσσά για παιχνίδια και WebApps, scripting language

Πίνακας 5.1: Γλώσσες προγραμματισμού που μπορούν να χρησιμοποιηθούν για τον προγραμματισμό του ESP32

Το framework που είναι πιο φιλικό προς τον χρηστή είναι το Arduino IDE. Πρόκειται για ένα cross-platform περιβάλλον προγραμματισμού μικροελεγκτών που χρησιμοποιεί συναρτήσεις της γλώσσας C και C++. Ήταν από την πρώτη μέρα δωρεάν σε όποιον ήθελε να ασχοληθεί με τους μικροελεγκτής ακόμα κι από το σπίτι του. Θεωρείται μεγάλο βήμα για την κοινότητα open source hardware. Πέρα από τις πλακέτες Arduino δίνεται η δυνατότητα προγραμματισμού και άλλων συμβατών πλακετών.

5.2 Αρχές λειτουργίας

Το μεγαλύτερο μειονέκτημα χρήσης Arduino IDE είναι ότι αφού παρέχει ένα υποσύνολο συναρτήσεων σε σχέση με ένα κανονικό IDE, υπολογίζεται ότι καλύπτει το 90% των δυνατοτήτων της γλώσσας C και C++. Ένα απλό παράδειγμα προγραμματισμού στο περιβάλλον αυτό φαίνεται στην εικόνα 5.2.



Σχήμα 5.2: Επιφάνεια εργασίας Arduino IDE.

Στην τερμινολογία των Arduino το πρόγραμμα που γραφούμε ονομάζεται “sketch”. Ο έλεγχος του κώδικα που σκοπεύει ο χρήστης να φορτώσει γίνεται από το κουμπί “Verify” που για εικονίδιο έχει ένα σύμβολο check ✓. Η φόρτωση του κώδικα σε έναν μικροελεγκτή γίνεται με το κουμπί “Upload” που για εικονίδιο έχει ένα σύμβολο βελάκι. Επίσης οι δημιουργητές του προγραμματιστικού περιβάλλοντος έχουν αυτοματοποιήσει τις ρυθμίσεις που θα έπρεπε να γίνουν από τον χρηστή προκειμένου να είναι συμβατή η πλακέτα που θα προγραμματίσουμε με τον υπολογιστή. Σημειώνεται πως η πολυπλοκότητα ρυθμίσεων ενός μικροελεγκτή απαιτεί τεχνικές γνώσεις που δεν συναντάς σε έναν προγραμματιστή εφαρμογών υπολογιστή.

5.3 Επίλογος

Κλείνοντας, σε αυτό το κεφάλαιο παρουσιαστήκαν οι πιο συνηθισμένοι τρόποι προγραμματισμού ESP32, εστιάζοντας στο Arduino που επιλέχτηκε για τον προγραμματισμό αυτής της ΠΕ. Αναλύοντας τον τρόπο που λειτούργει είδαμε ποσό εύκολο είναι να προγραμματίσουμε έναν μικροελεγκτή με το Arduino IDE.

Κεφάλαιο 6ο: Android

6.1 Εισαγωγή

Το λειτουργικό σύστημα (Operating System) Android χρησιμοποιήθηκε για την ένδειξη των αποτελεσμάτων της παρούσας εργασίας. Πρόκειται για το πιο πολυχρησιμοποιημένο ανάμεσα στα υπόλοιπα λειτουργικά συστήματα είτε ηλεκτρονικού υπολογιστή είτε έξυπνου τηλέφωνου (Smartphone). Είναι ανοιχτού κώδικα (Open Source) και αναπτύχθηκε, από τον οργανισμό τηλεπικοινωνιακών εταιρειών Open Handset Alliance με επικεφαλή τη Google, πάνω στον πυρήνα των Linux (Kernel) με σκοπό την χρήση του σε συσκευές έξυπνων τηλεφώνων (Smartphones). Στη συνέχεια εφαρμόστηκε σε πολλές τεχνολογίες, ανάμεσα τους είναι:

- ταμπλέτες (Tablets)
- ενδύσεις όπως έξυπνα ρολόγια (Smartwatch)
- τηλεοράσεις (Android TV)
- αυτοκίνητα
- διαδίκτυο των πραγμάτων (Internet of Things)

Η έκδοση που τρέχουν τα τελευταία μοντέλα με λειτουργικό σύστημα Android είναι η ενδεκάτη ($11^{η}$) και ονομάζεται Android Heck.

6.2 Χαρακτηριστικά

Παρακάτω αναλύονται τα χαρακτηριστικά που εύκολα θα λέγαμε πως έχουν βοηθήσει το Android να γίνει το κορυφαίο λειτουργικό σύστημα.

6.2.1 Δωρεάν χρήση

Βασισμένο πάνω στον ανοιχτό κώδικα (Open source) και καταχωρημένο με την άδεια χρήσης GNU GPLv2 (General Public Licence version 2) δίνεται η δυνατότητα χρήσης του λειτουργικού συστήματος σε οποιοδήποτε βαθμό. Όπως τη διανομή του από τρίτους είτε δωρεάν, είτε επί πληρωμή συνήθως από εταιρίες. Επιτρέπεται να έχει οποιοσδήποτε πρόσβαση στον κώδικα του με σκοπό να τον αντιγράψει, να τον αλλάξει προσθέτοντας και αφαιρώντας δικό του κώδικα, με οποιοδήποτε σκοπό κερδοσκοπικό και μη εφόσον παραμένει μέσα στα όρια που του ορίζει η άδεια ανοιχτού κώδικα. Εκεί που οι περισσότερες εταιρίες δίνουν βάση είναι οι αλλαγές πάνω στο περιβάλλον που δραστηριοποιείται ο χρήστης UI (User Interface).

6.2.2 Προγραμματισμός

Ο κώδικας τόσο του λειτουργικού συστήματος όσο και των εφαρμογών που υπάρχουν στο Android περιβάλλον γίνεται με δυο πολύ διαδεδομένες γλώσσες προγραμματισμού που υπάρχουν στον χώρο πάνω από 25 χρονιά η κάθε μια. Οι γλώσσες αυτές είναι η C++ και η Java. Ευρέως διαδεδομένες και απαραίτητες για την σταδιοδρομία ενός προγραμματισμού υψηλού επίπεδου. Τα εργαλεία τα όποια μπορεί κάνεις να χρησιμοποιήσει προκειμένου να αναπτύξει κώδικα με τις γλώσσες αυτές παρέχονται επίσης δωρεάν. Για την γλωσσά C++ είναι το NDK (Native Development Kit) και για την γλωσσά Java το SDK (Software Development Kit).

6.2.3 Διανομή εφαρμογών

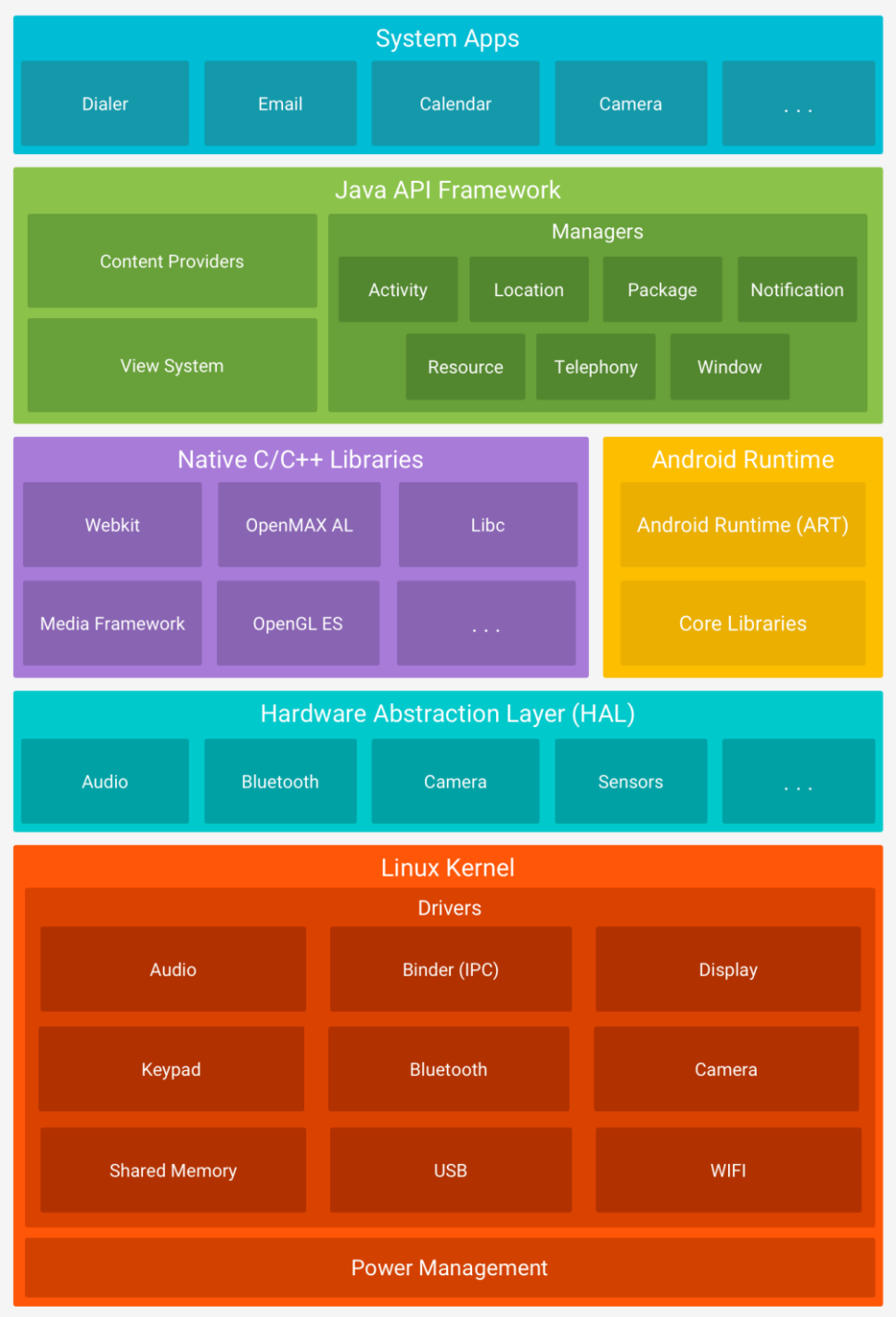
Το κομμάτι της διανομής εφαρμογών κατέχει η εταιρεία Google με την πλατφόρμα Google Play Store, που εξασφαλίζει τον έλεγχο των εφαρμογών από κακόβουλο λογισμικό και διατίθενται δωρεάν προς τους χρηστές ή με πολύ χαμηλό κόστος.

6.3 Αρχιτεκτονική

Τα βασικά μέρη που απαρτίζουν το λειτουργικό σύστημα Android, από την σύνθεση που κάνει το λογισμικό (software) με τα ηλεκτρονικά εξαρτήματα (hardware) μέχρι και τις εφαρμογές που βλέπουμε στις οθόνες των έξυπνων κινητών (Smartphones), είναι 6 και αναλύονται παρακάτω.

Βασικά μέρη	Περιγραφή
System Apps	Τους οι εφαρμογές που έχει επαφή ο χρήστης.
Java API Framework	Βιβλιοθήκες και προγράμματα σε Java.
Native C/C++ libraries	Κώδικας και υπηρεσίες που παρέχει το Android σε γλώσσα C++.
Android Runtime	Βασικές βιβλιοθήκες προγραμματισμού σε γλώσσα Java.
Hardware Abstraction Layer	Διεπαφές (Interfaces) για την αλληλεπίδραση των ηλεκτρονικών εξαρτημάτων με των λογισμικών.
Linux kernel	Οδηγοί (Drivers) ηλεκτρονικών εξαρτημάτων και διαχείρισή τους. Πυρήνας (βάση) του Android.

Πίνακας 6.1: Βασικά μέρη λειτουργικού συστήματος Android και σύντομη περιγραφή.



Σχήμα 6.1: Αρχιτεκτονική λειτουργικού συστήματος Android.

6.4 Επίλογος

Σε αυτό το κεφάλαιο είδαμε περιληπτικά την δομή του λειτουργικού συστήματος Android και τους βασικούς λογούς που έχει γίνει το πιο πολυχρησιμοποιημένο λειτουργικό σύστημα μεχρι σήμερα.

Κεφάλαιο 7ο: MIT App Inventor

7.1 Εισαγωγή

Στο κεφάλαιο αυτό θα αναλύσουμε την πλατφόρμα δημιουργίας εφαρμογών Android που προσφέρει δωρεάν το Τεχνολογικό Ινστιτούτο Μασαχουσέτης (Massachusetts Institute of Technology) υπό την άδεια Creative Commons και φέρει το όνομα MIT App Inventor.

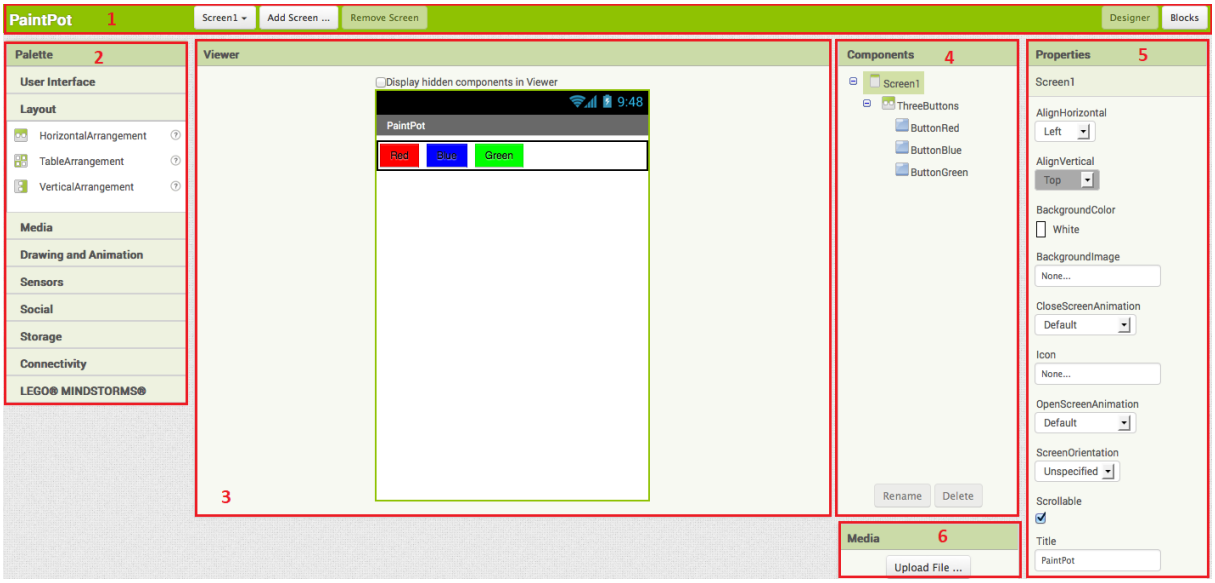
Για τον προγραμματισμό μιας εφαρμογής που δουλεύει σε περιβάλλον Android συνηθίζεται η χρήση των γλωσσών προγραμματισμού Java, Kotlin και C++ σε συνεργασία με εργαλεία ανάπτυξης εφαρμογών όπως το Android Studio. Μια πρωτοβουλία της Google που παρέδωσε στο MIT όμως έχει αλλάξει τα δεδομένα στον χώρο των ερασιτεχνών καθώς με αυτήν την πλατφόρμα υπάρχει η δυνατότητα ανάπτυξης εφαρμογών Android με βασικές γνώσεις προγραμματισμού, χωρίς να χρειάζεται να εμβαθύνει κάνεις σε γλώσσες όπως η Java.

7.2 Αρχές λειτουργίας

Η πλατφόρμα αναφέρεται σε άτομα που ασχολούνται ερασιτεχνικά με τον προγραμματισμό ή κάνουν τα πρώτα τους βήματα κάνοντας χρήση του φυλλομετρητή (browser) καθώς στεγάζεται σε Cloud. Χρησιμοποιεί την μέθοδο drag-and-drop τόσο στον σχεδιασμό όσο και στον προγραμματισμό. Για τη δημιουργία μιας Android εφαρμογής πρέπει να χρησιμοποιηθούν τρία βασικά εργαλεία. Το πρώτο είναι ο σχεδιαστής (Designer) που φροντίζει για τα γραφικά και τις Διεπαφές (interface) που θα έχει αλληλεπίδραση ο χρήστης της εφαρμογής και το δεύτερο είναι είναι ο editor που γίνεται η σύνταξη του προγράμματος μέσα από γεωμετρικά μπλοκ (blocks) και καθορίζεται οτιδήποτε συμβαίνει στην εφαρμογή. Το τελευταίο εργαλείο πρόκειται για τις επιλογές που προσφέρονται από το MIT App Inventor ώστε να δοκιμαστεί (testing) η εφαρμογή σε πραγματικό χρόνο.

7.2.1 Designer

Ο σχεδιαστής όπως φαίνεται στην εικόνα 7.1 αποτελείται από μια διάταξη έξι διαφορετικών παραθύρων.



Σχήμα 7.1: Επιφάνεια εργασίας σχεδίασης (Designer) εφαρμογής Android του MIT App Inventor.

- Το παράθυρο με αριθμό 1 που βρίσκεται στο πάνω μέρος του σχεδιαστή γραφεί πάντα την επικεφαλίδα του έργου που τελείται στο αριστερό του μέρος. Δίνονται οι επιλογές πρόσθεσης, αφαίρεσης και εναλλαγής οθονών της εφαρμογής και τέλος στο δεξί μέρος του παράθυρου υπάρχουν δυο κουμπιά που οδηγούν είτε στον σχεδιαστή είτε στον προγραμματισμό του έργου.

- Το 2ο παράθυρο στην αριστερή μεριά του σχεδιαστή ονομάζεται παλέτα (Palette) και είναι το μέρος που ξεκάνει ο σχεδιασμός αφού από εκεί επιλεγούμε ότι στοιχειό και γενικότερα χρήση θα έχει η εφαρμογή. Αναλυτικότερα παρακάτω είναι όλες οι επιλογές στοιχείων που προσφέρονται από αυτό το παράθυρο.

Κατηγορίες	• Στοιχειά
Διεπαφή χρήστη (User Interface)	• Κουμπιά • Κουτιά έλεγχου (Checkbox) • Επιλογή Ημερομηνίας • Εικόνες • Ετικέτες (Label) • Επιλογές από λίστα • Προβολές λίστας • Ειδοποιήσεις • Κείμενο για κωδικούς • Στοιχειά ολίσθησης • Διακόπτες • Πεδία ελευθέρου κειμένου • Προβολείς σελίδων
Διατάξεις/Διαρρυθμισεις (Layouts)	• Οριζόντιες • Κάθετες • Οριζόντιες διατάξεις με κύλιση (scroll) • Κάθετες διατάξεις με κύλιση (scroll) • Σύνθετη διάταξη
Μέσα (Media)	• Εγγραφή κάμερας • Κάμερα

	• Αναπαραγωγή εικόνας • Αναπαραγωγή ήχου • Αναπαραγωγή βίντεο • Ηχογράφηση ήχου • Αναγνώριση ήχου • Μετατροπή κειμένου σε λόγο • Μεταγλωττιστή (διερμηνέα)
Ζωγραφική και Animation	• Μπάλα (λειτουργιά πινέλου) • Καμβάς
Χάρτες (Maps)	• Κύκλος για την περίμετρο • Γραμμή για την δημιουργία διαδρομών • Χάρτη • Τοποθέτηση σημείου • Πλοήγηση • Γεωμετρικά σχήματα για την εύρεση περιμέτρων
Αισθητήρες (Sensors)	• Επιταχυνσιογράφο • Barcode σκάνερ • Βαρόμετρο • Ρολόι • Γυροσκόπιο • Υγρόμετρο • Αισθητήρας φωτός • Αισθητήρας τοποθεσίας • αισθητήρας μαγνητικού πεδίου • Αισθητήρας προσανατολισμού • Βηματομετρητή • Μετρητής απόστασης • Θερμόμετρο
Μέσα επικοινωνίας (Social)	• Επιλογή επαφής κινητού • Επιλογή e-mail • Τηλέφωνο • Επιλογή αριθμού τηλέφωνου • Κοινοποίηση • Αποστολή μηνύματος
Αποθήκευση (Storage)	• Βάση δεδομένων σε σύννεφο (CloudDB) • Αρχείο • Βάση δεδομένων στο κινητό (TinyDB) • Βάση δεδομένων σε ιστοσελίδα (TinyWebDB)
Συνδεσιμότητα (Connectivity)	• Εκκινητής δραστηριότητας • Bluetooth • Σειριακό • Ιστοσελίδα (HTTP requests)
Lego	Διεπαφές και αισθητήρες για ρομποτική με Lego
Σε πειραματικό στάδιο (Experimental)	• Βάση δεδομένων Firebase
Επιπρόσθετα (Extensions)	Εισαγωγή επιπρόσθετων που προγραμματίζονται με Java

Πίνακας 7.1: Κατηγορίες στοιχείων για την σχεδίαση Android εφαρμογής στο MIT App Inventor.

- Στο 3^{ο} παράθυρο βρίσκεται ο Viewer που είναι μια αναπαράσταση της οθόνης που πάμε να δημιουργήσουμε. Επιλέγοντας τα στοιχειά από το παράθυρο της παλέτας μπορούμε να τα σύρουμε και να τα κατανέμουμε με όποιον τρόπο μας ταιριάζει

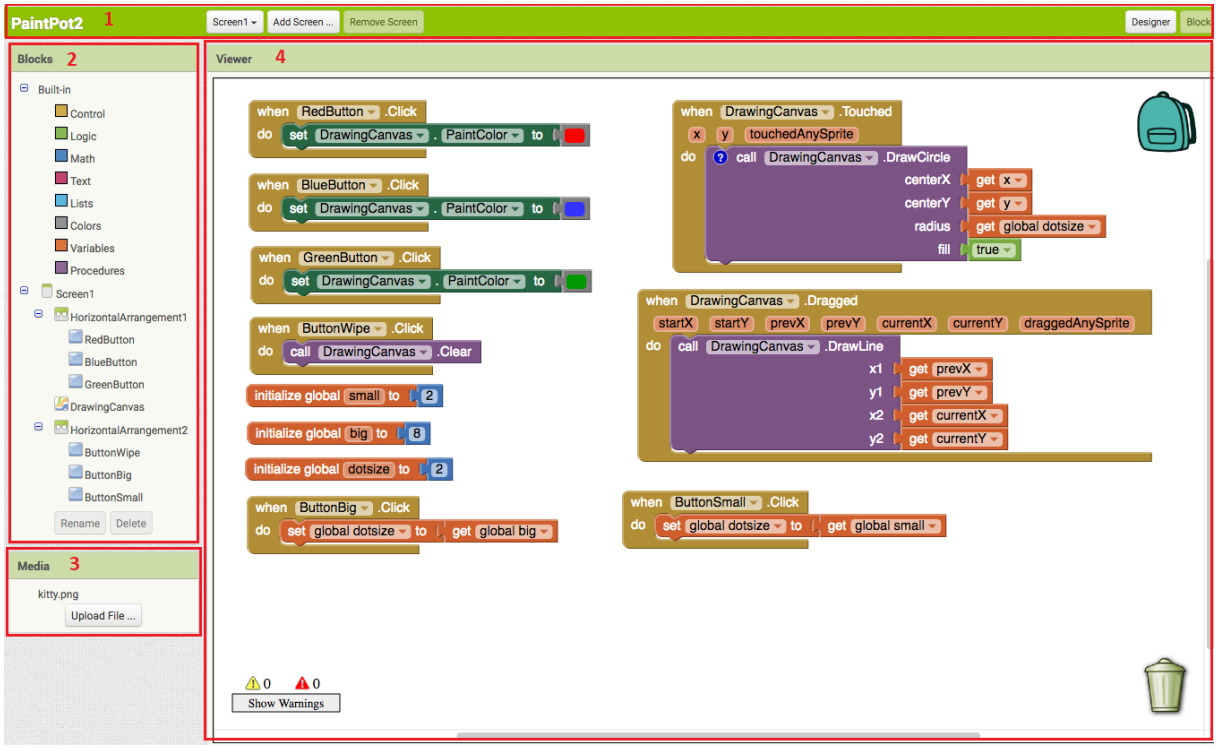
- Στο 4ο παράθυρο του σχεδιαστή βρίσκονται όλα τα ορατά και μη στοιχειά που έχουμε προσθέσει στην οθόνη. Ουσιαστικά είναι ένας χάρτης με την όψη δενδροδιαγράμματος που μας δίνει την δυνατότητα να επιλέξουμε, να διαγράψουμε και να μετονομάσουμε τα στοιχειά που έχουμε στην οθόνη.

- Το 5ο παράθυρο πρόκειται για τις ιδιότητες κάθε στοιχείου που έχουμε επιλέξει για την οθόνη. Οι ιδιότητες αφορούν τα χρώματα, τα κείμενα, τις γραμματοσειρές, που θα φαίνονται στην οθόνη, τα σχήματα και τα μεγέθη που θα έχουν τα στοιχειά καθώς και πολλά άλλα.

- Τέλος στο παράθυρο 6 είναι τα μέσα (Media), μια επιλογή που μας επιτρέπει να ανεβάσουμε πολυμέσα όπως ήχους και εικόνες από τον υπολογιστή.

7.2.2 Blocks editor

Ο προγραμματισμός της εφαρμογής, που αντικαθιστά τις γλώσσες προγραμματισμού με γεωμετρικά σχήματα-κομμάτια πάζλ και καθορίζει την λειτουργιά της, πραγματοποιείται στο δεύτερο κύριο εργαλείο που δίνεται από το MIT App Inventor, με το όνομα Blocks Editor. Έχοντας τις επιλογές από το παράθυρο 2 της εικόνας 7.2 μας δίνεται η δυνατότητα να σύρουμε κομμάτια κώδικα με την μέθοδο drag-and-drop μέσα στον Viewer (4ο παράθυρο της εικόνας 7.2) και να τα συνδέσουμε με κομμάτια κώδικα από τα στοιχειά που προσθέσαμε από τον σχεδιαστή (Designer) ώστε ορίσουμε την λειτουργιά της εφαρμογής. Παρακάτω αναλύονται με βάση την εικόνα 7.2, τα 4 παράθυρα του Block Editor.



Σχήμα 7.2: Block Editor στο MIT App Inventor.

- Το παράθυρο 1 και οι λειτουργιές του παραμένει ίδιο με το παράθυρο 1 του σχεδιαστή.

- Στο παράθυρο 2 παρουσιάζονται τα μπλοκ που σε διάφορα σχήματα περιέχουν καθορισμένο και αμετάβλητο κώδικα. Υπάρχουν εννέα κύριες προεγκατεστημένες (Built-in) κατηγορίες για τα κομμάτια κώδικα που καλύπτουν βασικές αρχές του προγραμματισμού όπως λούπες και συναρτήσεις, και μετά υπάρχουν τα κομμάτια κώδικα που καλύπτουν την επεξεργασία των στοιχείων που προσθέσαμε στον σχεδιαστή (Designer). Η επεξεργασία αυτή αφορά τις ιδιότητες των στοιχείων και βοηθού περισσότερο οπτικά στην εφαρμογή παρά στον καθορισμό της λειτουργιάς της.

Κατηγορίες προεγκατεστημένων μπλοκ	Παραδείγματα
Έλεγχου (Control)	• Λούπες while, for • if else • Άνοιγμα, κλείσιμο οθόνης • Κλείσιμο εφαρμογής
Λογικής (Logic)	• True, False • not, and, or
Μαθηματικών (Math)	• Εισαγωγή αριθμών • πρόσθεση, αφαίρεση, διαίρεση • έσο με, μεγαλύτερο έσο, διάφορο • σύγκριση για μέγιστο και μικρότερο • απολυτή τιμή, τετραγωνική ριζά
Κειμένου (Text)	• Σύνθεση, μετατροπή, διαχωρισμός, αντικατάσταση
Λίστες (Lists)	• Δημιουργία λίστας, • πρόσθεση κελιών και αντικειμένων, • αντικατάσταση κελιών, διαγραφή κελιών
Dictionary	• Δημιουργία dictionary με κλειδί και τιμή • ορισμός διαγραφή αντικατάσταση τιμής ή κλειδιού
Χρώματα (Colors)	• Μαύρο, κόκκινο, κίτρινο, μπλε • σύνθεση χρώματος
Μεταβλητών (Variables)	• Δημιουργία μεταβλητής Global ή local • ορισμός μεταβλητής • εμφάνιση μεταβλητής
Διεργασιών (Procedures)	• ορισμός διεργασιών-συναρτησεων με ή χωρίς επιστροφή τιμής • κάλεσμα διεργασιών-συναρτησεων

Πίνακας 7.2: Κατηγορίες προεγκατεστημένες μπλοκ κώδικα και η χρήση τους στο MIT App Inventor.

- Η λειτουργιά του 3ου παράθυρου που ονομάζεται μέσα (Media) παραμένει ίδια με του παράθυρου 6 από τον σχεδιαστή.

- Στο 4ο παράθυρο γίνεται η σύνθεση των μπλοκ κώδικα σαν πάζλ με την μέθοδο drag-and-drop. Είναι το παράθυρο που υπάρχει η περισσότερη δραστηριότητα αφού από εδώ ορίζουμε την λειτουργιά της εφαρμογής. Επίσης υπάρχει και μια λειτουργιά που μοιάζει με compiller και μας προειδοποιεί για τυχόν συντακτικά λάθη.

7.2.3 Άλλα εργαλεία

Με την πλατφόρμα μας δίνεται επίσης η δυνατότητα να βλέπουμε σε πραγματικό χρόνο (real-time) τις αλλαγές που κάνουμε σε μια εφαρμογή. Αυτή η δυνατότητα πραγματοποιείται είτε με σύνδεση USB του κινητού με τον υπολογιστή που είναι ανοιχτός ο brower, είτε με το AI Companion που είναι μια εφαρμογή για Smartphone και προσφέρει σύνδεση μέσω τοπικού δικτύου, είτε μέσω προσομοιωτή smartphone που λειτούργει στον φυλλομετράτε.

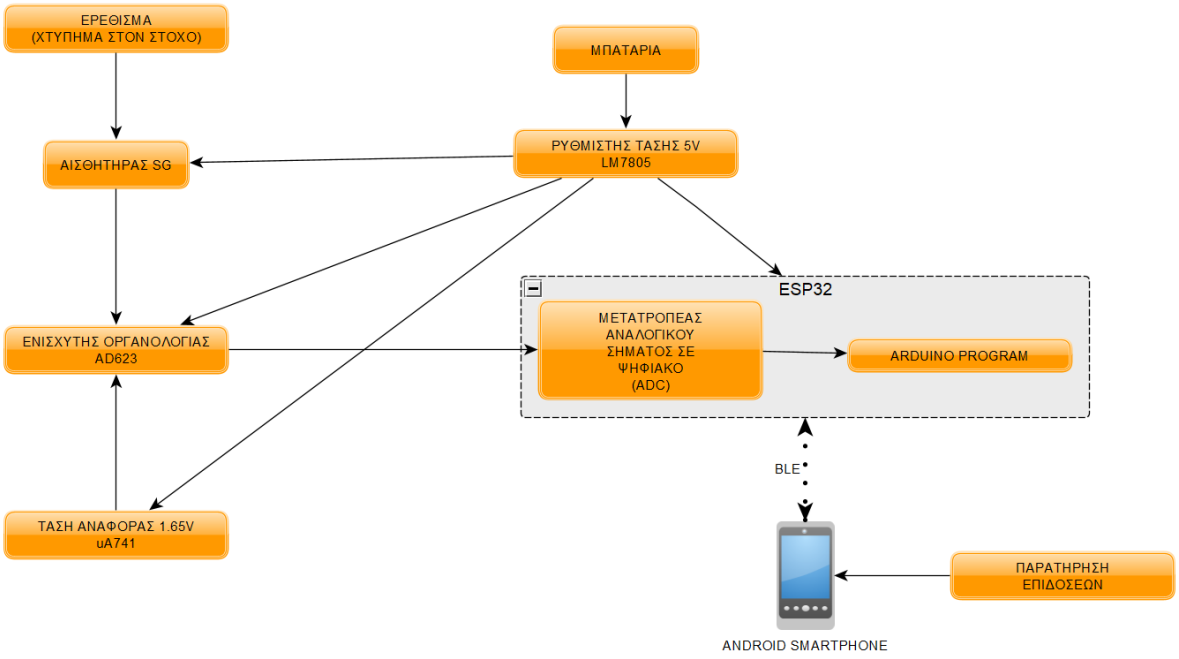
7.3 Επίλογος

Σε αυτό το κεφάλαιο αναλύσαμε τον τρόπο λειτουργιάς της πλατφόρμας MIT App Inventor, τα εργαλεία που χρησιμοποιεί προκειμένου να ολοκληρωθεί μια εφαρμογή για Android καθώς και την ορολογία της.

Κεφάλαιο 8ο: Σύνθεση κυκλώματος

8.1 Εισαγωγή

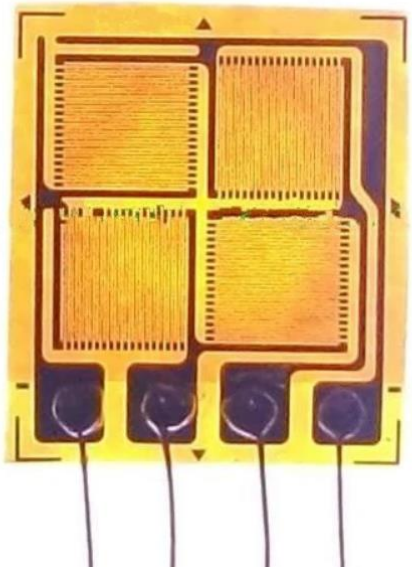
Το επιθυμητό αποτέλεσμα και ο στόχος της ΠΕ αυτής, είναι η υλοποίηση ενός ηλεκτρονικού προπόνηση που θα δέχεται, θα επεξεργάζεται και θα εμφανίζει τα χτυπήματα ενός αθλητή πολεμικών τεχνών. Για την υλοποίηση του ηλεκτρονικού προπονητή εκτελεστήκαν αρκετά βήματα και συνδυαστήκαν πολλές διαφορετικές τεχνολογίες. Αφού προηγήθηκε στα προηγούμενα κεφάλαια ανάλυση του θεωρητικού υπόβαθρου καθώς και του τρόπου λειτουργιάς των εξαρτημάτων και των τεχνολογιών αυτών, θα ακολουθήσει ανάλυση του τρόπου υλοποίησης του κυκλώματος και της σύνθεσης του. Παρακάτω φαίνεται το διάγραμμα ροής ολοκλήρου του κυκλώματος ξεκινώντας από το ερέθισμα του αισθητήρα μεχρι και τις ενδείξεις που θα φτάνουν στο κινητό Smartphone.



Σχήμα 8.1: Διάγραμμα ροής λειτουργίας κυκλώματος

8.2 Αισθητήρας Strain Gauge

Ο αισθητήρας που επιλέχτηκε βρίσκεται πάνω σε βάση με διαστάσεις $9.6 \times 8.2mm$ από το οποιο το πλεγμα του είναι $6.5 \times 6.5mm$ και εχει κολλημενα και έτοιμα για χρήση τέσσερα ποδαράκια (pins), όπως φαίνεται στην εικόνα 8.1. Όταν βρίσκεται σε κατάσταση ηρεμίας, η ένδειξη που θα ληφθεί είναι τα *1000Ω* ενώ σε κατάσταση καταπόνησης οι δυο ακραίες τιμές που θα πάρει ένια *997Ω* και *1003Ω,* ανάλογα με τις κατευθύνσεις των καταπονήσεων. Έχει ευαισθησία εξόδου στα *2.1mV/V* που σημαίνει ότι στη μεγίστη δυνατή παραμόρφωση του, η ένδειξη που θα παρουσιάσει στην έξοδο του θα είναι *2.1mV* για κάθε *Volt* που τροφοδοτείται. Ο συντελεστής ευαισθησίας GF είναι 2, όπως συνηθίζεται για τους αισθητήρες SG μεταλλικού φύλλου (metal foil). Τέλος από την εξίσωση (3) φαίνεται πως η παραμόρφωση ε (Strain) του αισθητήρα είναι ώση με $1500\mu\varepsilon$.



Σχήμα 8.2: Αισθητήρας τετραπλού SG σε συνδεσμολογία πλήρους γέφυρας Wheatstone.

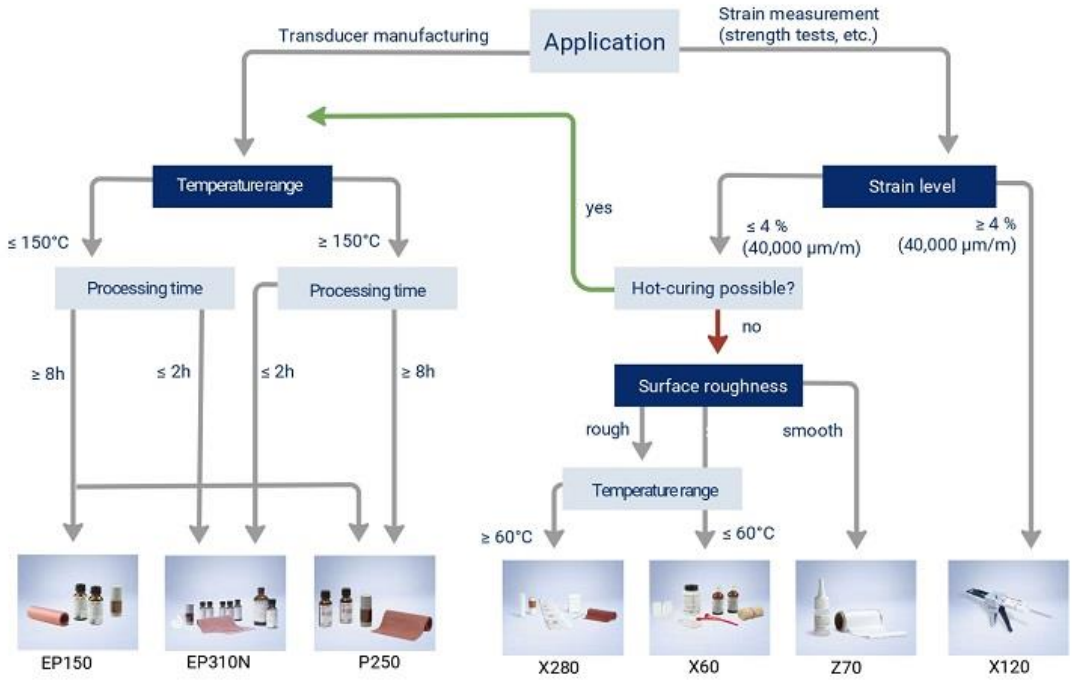
Στην εικόνα διακρίνεται πως δεν πρόκειται για έναν αισθητήρα SG αλλά για τέσσερις συνδεσμολογημένους σε πλήρη γέφυρα Wheatstone ή αλλιώς γέφυρα 4/4. Η γέφυρα τεσσάρων SG βοηθούν ιδιαίτερα στην κόλληση της με τα υπόλοιπα εξαρτήματα όπως και στη εφαρμογή της πάνω στο αντικείμενο προς μέτρηση, καθώς δεν πρόκειται για τέσσερα SG σε διαφορετικά μεταλλικά φύλλα αλλά σε ένα.

8.3 Εφαρμογή SG πάνω σε Plexiglass

Η εφαρμογή κάθε αισθητήρα SG προϋποθέτει την κόλληση του πάνω σε κάποιο στερεό υλικό. Έτσι με την καταπόνηση του στέρεου, υπάρχει ερέθισμα που αντιλαμβάνεται το SG και αλλάζει ελάχιστα την αντίσταση του. Το στερεό υλικό που επιλέχτηκε για αυτή την ΠΕ είναι το Plexiglass. Πρόκειται για ένα πλαστικό υλικό με ανθεκτική ελαστική ικανότητα που με την άσκηση ανάλογης δύναμης θα υποστεί μηχανική καταπόνηση ανάλογη της κάμψης. Η δύναμη που θα ασκείται είναι τα χτυπήματα που θα εκτελεί ο αθλητής πάνω στο plexiglass. Λόγω του ανθρωπινού παράγοντα, τα χτυπήματα είναι αδύνατο να είναι πάντα με ακρίβεια στο κέντρο του Plexiglass που θα εφαρμοστεί ο αισθητήρας και θα λαμβάνεται η πραγματική καταπόνηση και κατ επέκταση η καλύτερη δυνατή μέτρηση. Οι διαστάσεις της επιφάνειας του Plexiglass είναι κρίσιμες και επιλέχθηκαν το ύψος 20cm, το πλάτος 15 cm και το πάχος 5mm. Αρκετά μεγάλες σε σχέση με την επιφάνεια του χεριού ή του ποδιού του αθλητή που θα έρχεται σε κρούση και ταυτόχρονα αρκετά μικρές για να μπορεί να χωρέσει μέσα σε έναν κοινό στόχο προπόνησης πολεμικών τεχνών. Επίσης με την επιλογή διαστάσεων ορίζεται και η καταπόνηση του plexiglass στο σημείο που θα προκαλέσει μόνιμη παραμόρφωση. Κάτι που είναι αντίθετο του επιθυμητού αποτελέσματος.

Ένα μείζον θέμα που τίθεται με την εφαρμογή αισθητήρα σε στόχο προπόνησης είναι η σταθερότητα του. Όταν για παράδειγμα ο αθλητής, για την προπόνηση του θα κτυπάει με γροθιές και λακτίσματα

τον στόχο που αποτελείται από Plexiglass, θα πρέπει ο αισθητήρας να παραμένει σταθερός και αμετάβλητος. Για τον λόγο αυτό χρησιμοποιήθηκε ειδική κόλλα για να κολληθεί πάνω στο Plexiglass το SG. Η επιλογή της κόλλας έγινε με βάση το παρακάτω διάγραμμα, όπως προτείνεται από τις εταιρίες παράγωγης μετατροπέων (transducer) και αισθητήρων.



Σχήμα 8.3: Διάγραμμα επιλογής συγκόλλησης SG.

Ξεκινώντας από την αρχή του διαγράμματος:

- Η εφαρμογή πρόκειται για μέτρηση δύναμης μέσω της καταπόνησης του Plexiglass.
- Η παραμόρφωση είναι μικρότερη από *40,000με*.
- Δε θα χρησιμοποιηθεί μέθοδος heat-curing (βιομηχανική επεξεργασία σκλήρυνσης υλικών).
- Η επιφάνεια εφαρμογής του αισθητήρα και της κόλλας είναι το Plexiglass και θεωρείται λεία.

Με βάση τα παραπάνω, πιο κατάλληλο είδος κόλλας κρίνεται αυτό που περιέχει κυανοακρυλικό υλικό όπως η Z70 (δεύτερη από το τέλος), αντέχει σε θερμοκρασίες από *-55C^o* μεχρι *100C^o* και η συγκόλληση διαρκεί συνήθως 1-2 λεπτά με πίεση του αντίχειρα.

Ειδική προετοιμασία και μεταχείριση χρειάζεται επίσης η πλακέτα Plexiglass, προκειμένου η συγκόλληση να γίνει σωστά. Τα βήματα που κατέστησαν εφικτή την συγκόλληση του αισθητήρα και της πλακάς από Plexiglass με κυανοακρυλικη κόλλα είναι:

1. Πιάσιμο της πλακάς Plexiglass σε μέγγενη πάγκου ώστε να παραμείνει σταθερή κατά την όλη διαδικασία.

2. Σχεδίαση των ορίων που θα θέτουν την θέση του αισθητήρα, με μαρκαδόρους πινάκα, στην αντίθετη μεριά από αυτή που πρόκειται να κολλήσει ο αισθητήρας και χωρίς να χρησιμοποιηθεί ο αισθητήρας αλλά οποιοδήποτε καθαρό αντικείμενο που φέρει τις διαστάσεις του, όπως ένα κομμάτι χαρτί.

3. Κάλος καθαρισμός με καθαρό πανί ώστε να φύγει η σκόνη από το σημείο εφαρμογής.

4. Εφαρμογή Acetone πάνω στην επιφάνεια που θα κολληθεί το SG.

5. Καλό τρίψιμο του Acetone σε κυκλικές κινήσεις, με καινούργιο πανί, για τον πλήρη καθαρισμό της επιφάνειας, μεχρι να στεγνώσει.

6. Εφαρμογή του αισθητήρα SG πάνω στην επιφάνεια και μέσα στα όρια που σχεδιαστήκαν.

7. Εφαρμογή Sellotape (κολλητική ταινία) πάνω από τον αισθητήρα και κατά μήκος της επιφάνειας, προκειμένου να πραγματοποιηθούν τα επόμενα βήματα χωρίς να έρθει σε επαφή ανθρώπινο χέρι ή οποιοδήποτε αντικείμενο με το πλέγμα του αισθητήρα. Λογά ευαισθησίας του πλέγματος του SG, οι κατασκευαστικές εταιρίες απαγορεύουν την επαφή του με οτιδήποτε.

8. Τραβώντας το Sellotape ώστε να ξεκολλήσει από τα όρια που σχεδιαστήκαν στο Plexiglass μαζί με τον αισθητήρα αλλά όχι να αποκολληθεί εντελώς, γίνεται ρίψη μια ή δυο σταγόνων από την κυανοακρυλικη κόλλα, μέσα στα όρια.

9. Αμέσως γίνεται κόλληση του Sellotape μαζί με τον αισθητήρα στην ίδια θέση που ήταν μεχρι και το $7^{ο}$ βήμα.

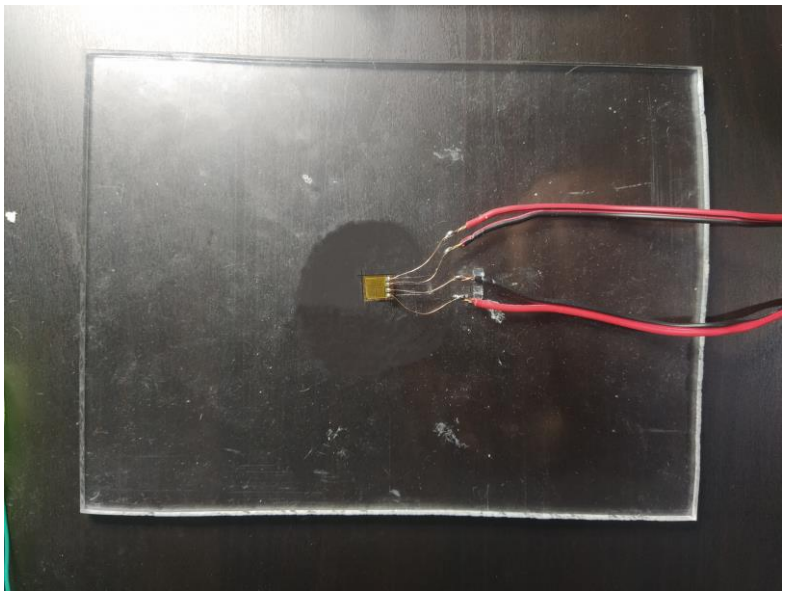
10. Ακολουθεί πίεση με τον αντίχειρα εφαρμοσμένο πάνω στο Sellotape στο σημείο που βρίσκεται το SG και έχει εφαρμοστεί η κόλλα προκειμένου να ολοκληρωθεί η συγκόλληση. Η πίεση συνήθως διαρκεί 1-2 λεπτά.

11. Αφαίρεση του Sellotape.

12. Εφαρμογή χαρτοταινίας χωρίς πίεση για την σταθεροποίηση και την καλύτερη προστασία του αισθητήρα και των βραχυκυκλωμάτων που έχει κάθε ποδαράκι (pin).

Στη συνέχεια χρησιμοποιώντας την τεχνική συγκόλλησης καλωδίων (soldering) με εργαλεία όπως καλάι και κολλητήρι, συγκολλήθηκαν στα τέσσερα ποδαράκια του αισθητήρα από ένα καλώδιο ώστε να υπάρχει απόσταση ανάμεσα στον αισθητήρα και στο κύκλωμα που ακόλουθη για την επεξεργασία του σήματος. Τα καλώδια κολλιούνται με την κυανοακρυλικη κόλλα πάνω στο Plexiglass δίνοντας

λίγο χώρο στα ποδαράκια του αισθητήρα, ώστε να μην ξηλωθούν. Η απόσταση είναι απαραίτητη προκειμένου το κύκλωμα να μην παθαίνει ζημία κατά την διάρκεια της προπόνησης.



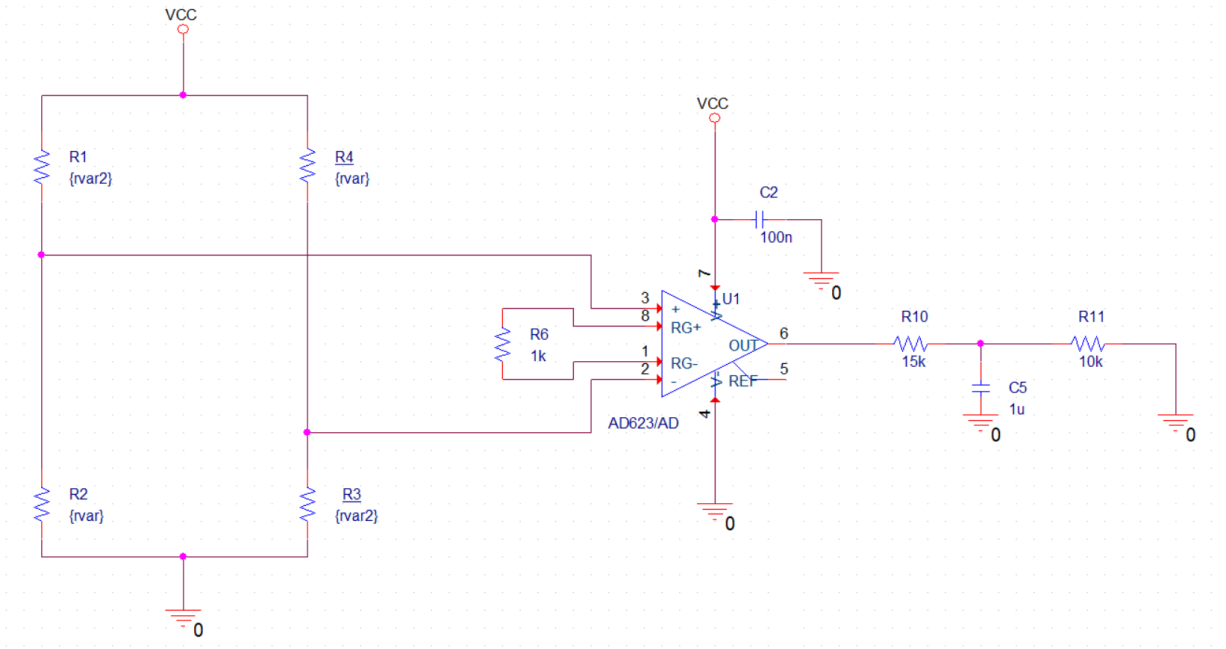
Σχήμα 8.4: Αισθητήρας SG πάνω στο Plexiglass μετά την σύνδεση του με καλώδια.

8.4 Ενισχυτής οργανολογίας

Το σήμα που έχει έξοδο από τον αισθητήρα με τέσσερα SG είναι της τάξεως των *α*. Καθώς μερικά *mV* καθιστούν ανέφικτη την επεξεργασία τους σε ένα κύκλωμα και στη συνέχεια την ψηφιοποίηση τους από έναν ADC, χρησιμοποιείται ενισχυτής οργανολογίας.

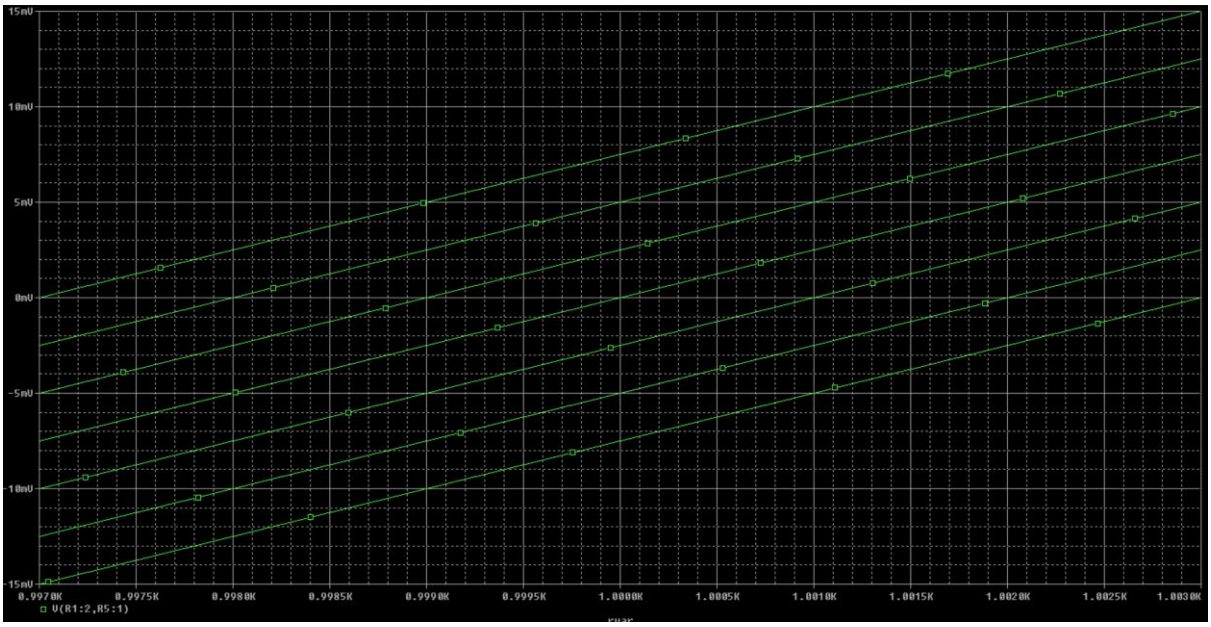
Ο ενισχυτής που επιλέχτηκε για το κύκλωμα αυτής της ΠΕ, είναι ο *AD623*. Πρόκειται για έναν ενισχυτή οργανολογίας που μπορεί να συνδεσμολογηθεί είτε με μονή είτε με διπλή συμμετρική τροφεία. Χρησιμοποιώντας για τροφοδοσία από *2.7V* μεχρι *12V* έχει τη δυνατότητα εξερχόμενου σήματος που αγγίζει τα όρια τροφοδοσίας του (rail-to-rail). Το εσωτερικό κύκλωμα του ενισχυτή επιτρέπει την επιλογή ενίσχυσης κέρδους μεχρι και 1000 φορές, με μια εξωτερική αντίσταση καθώς με την απουσία της εξασφαλίζεται μοναδιαίο κέρδος. Αξιοσημείωτη είναι επίσης η μεγάλη ακρίβεια που κατέχει σε DC σήματα με μοναδιαίο κέρδος, που παρουσιάζουν σφάλμα *0.10%* και σε κέρδος μεγαλύτερο της μονάδας, μόλις *0.35%*. Η μεγάλη του ακρίβεια επίσης έχει σαν αποτέλεσμα την αύξηση του λόγου απόρριψης κοινού σήματος (CMRR) στα AC σήματα, όσο μεγαλώνει και το κέρδος.

Η συνδεσμολογία του ενισχυτή οργανολογίας και του τετραπλού αισθητήρα SG έχει πραγματοποιηθεί όπως φαίνεται παρακάτω.



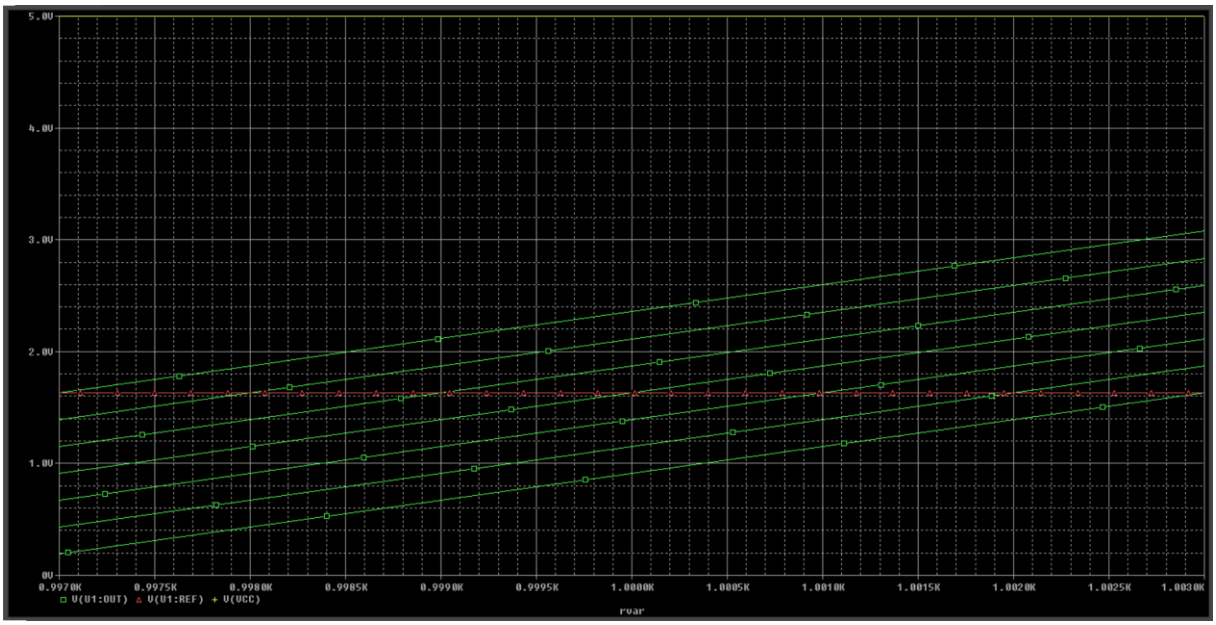
Σχήμα 8.5: Συνδεσμολογία ενισχυτή οργανολογίας και γέφυρας Wheatstone 4/4.

Η τροφοδοσία σε ολόκληρο το κύκλωμα θα παρέχεται από την πλακέτα του ESP32 και ορίζεται ίδια με την περιοχή λειτουργιάς του μικροελεγκτή, που είναι τα *5V*. Οι αντιστάσεις R1, R2, R3 και R4 αντιπροσωπεύουν την γέφυρα Wheatstone με τους τέσσερις αισθητήρες SG. Για τον λόγο αυτό είναι μεταβαλλόμενες στην προσημείωση και παίρνουν μεταβαλλόμενες τιμές, από *997Ω* μεχρι *1007Ω*, όπως γίνεται και στη πραγματικότητα με έναν αισθητήρα όταν υπόκειται για παράδειγμα κάμψη. Η διάφορα των εισόδων του ενισχυτή κυμαίνεται από *-15mV* έως *+15mV* και είναι ίση με την διάφορα που θα έχουν οι αντιστάσεις κατά την διάρκεια που βρίσκεται σε ένταση ο αισθητήρας όπως ορίζεται και από τα δεδομένα του SG που επιλέχτηκε ότι για κάθε τροφοδοτούμενο Volt θα παρουσιάζει έξοδο *3mV*. Έτσι διακρίνεται και στα αποτελέσματα της προσομοίωσης που πραγματοποιήθηκε με το πρόγραμμα Pspice της OrCAD στην Σχήμα 8.6.



Σχήμα 8.6: Αποτελέσματα προσομοίωσης διαφορικής εξόδου γέφυρας τετραπλού SG στο πρόγραμμα OrCAD.

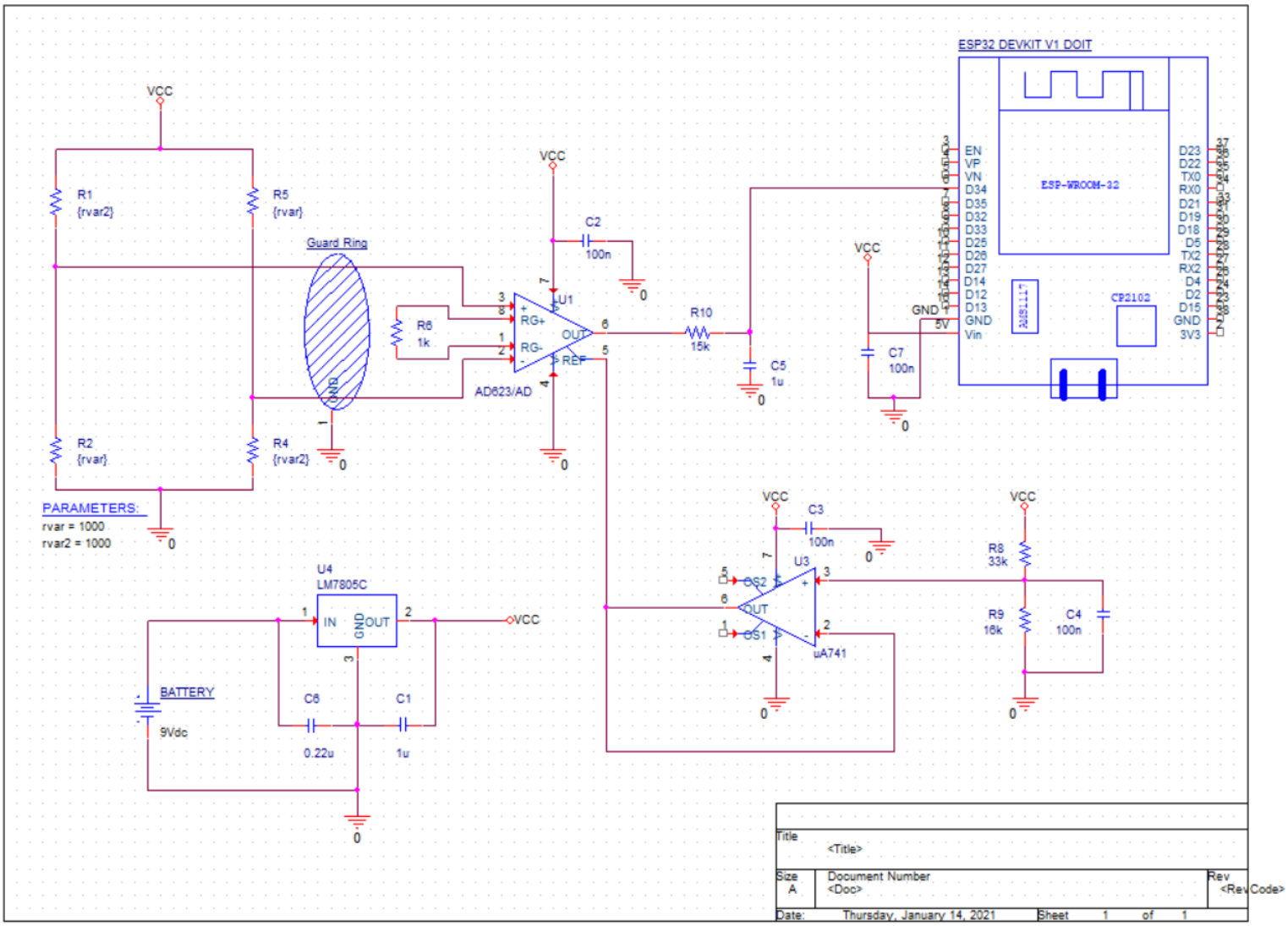
Στη συνέχεια έχοντας δεδομένο πως ο ADC του ESP32 δέχεται τιμές που κυμαίνονται από *0,1V* μεχρι *3.1V*, θα δοθεί ενίσχυση του διαφορικού σήματος κατά 100 φορές από τον ενισχυτή AD623 τοποθετώντας μια αντίσταση *1,02kΩ* όπως ορίζουν τα φύλλα δεδομένων του. Το αποτέλεσμα έπειτα από την τάση αναφοράς που θα δώσει ο γενικής χρήσεως τελεστικός ενισχυτής us741 στα *1.65V,* θα είναι μεταξύ *0.1V* και *3.1V,* όπως φαίνεται και στην προσομοίωση που παρουσιάζει το Σχήμα 8.7. Όπου με κίτρινη γραμμή (πάνω) φαίνεται η σταθερή τάση τροφοδοσίας του κυκλώματος από τον σταθεροποιητή τάσης LM7805C, με κόκκινη γραμμή (στη μέση) φαίνεται η σταθερή τάση αναφοράς που δίνεται από τον us741 και οι εφτά πράσινες γραμμές βασισμένες πάνω στην τάση αναφοράς, είναι οι ενισχυμένες κατά 100 φορές τάσεις εξόδου που θα παραδίδονται στον ADC του ESP32 ανάλογα με την κάμψη του που αντιλαμβάνεται το SG.



Σχήμα 8.7: Αποτέλεσμα προσομοίωσης κυκλώματος ενίσχυσης διαφορικού σήματος.

8.5 Κύκλωμα ηλεκτρονικού προπονητή

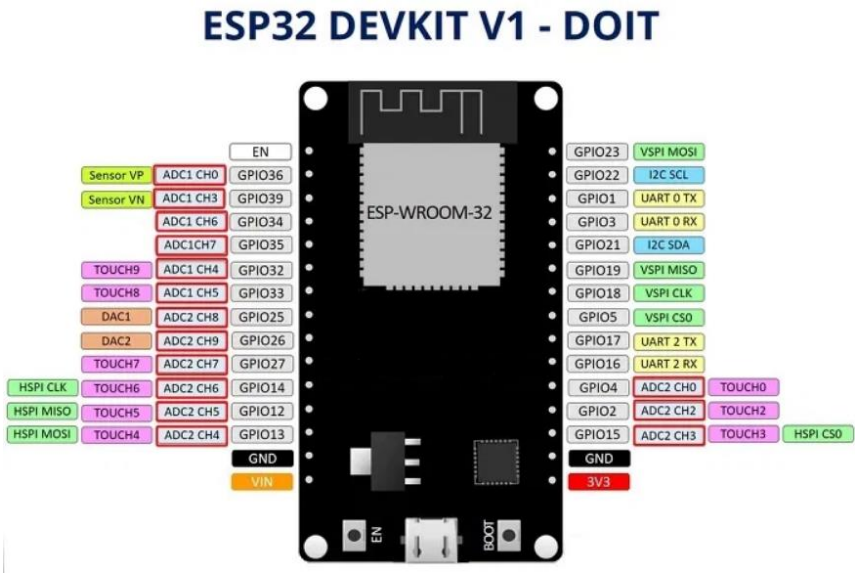
Στο παρακάτω σχήμα φαίνεται αναλυτικά η αναπαράσταση του κυκλώματος που δημιουργήθηκε για την παρούσα ΠΕ. Παρατηρώντας το διακρίνονται κάποια στοιχειά που δεν αναλύθηκαν σε προηγούμενο κεφάλαιο. Το πρώτο είναι ένα δαχτυλίδι προστασίας από πολύ μικρά ρεύματα που δημιουργούνται στην επιφάνεια ενός PCB από άλλες διαδρομές χαλκού και ονομάζεται Guard ring. Προκειμένου να λειτουργήσει τοποθετείται γύρω από τις εισόδους του ενισχυτή οργανολογίας χωρίς να τις ακουμπήσει και ενώνεται με την γείωση. Στη συνέχεια διακρίνονται τέσσερις πυκνωτές των *100no* στις τροφοδοσίες των AD623, us741, ESP32 στο ποδαράκι V_{in} και στον διαιρετή τάσης του us741 και πρόκειται για πυκνωτές αποκοπής (decouple capacitors). Ακόμα τοποθετήθηκε ένα χαμηλοπερατό φίλτρο anti-aliasing στην έξοδο του ενισχυτή οργανολογίας ώστε να κόβει όλα τα σήματα ανεπιθύμητων συχνοτήτων, με συχνότητα αποκοπής υπολογισμένη στα *15.6Hz*. Τέλος, οι δυο πυκνωτές που φαίνονται στο σχηματικό του σταθεροποιητή τάσης LM7805C προτείνονται από τα φύλλα δεδομένων του και συμβάλουν στην προστασία και την σωστή λειτουργιά του.



Σχήμα 8.8: Αναπαράσταση συνολικού κυκλώματος στο πρόγραμμα OrCAD.

8.6 Μετατροπή αναλογικού σήματος σε ψηφιακό

Όπως αναφέρθηκε και σε προηγούμενο κεφάλαιο, ο ADC βρίσκεται μέσα στον μικροελεγκτή ESP32 και οι τιμές που δέχεται κυμαίνονται από 0.1V μεχρι 3.1. Είναι ένας ADC 12-bit που σημαίνει ότι διακρίνει τις αναλογικές τιμές που θα πάρει σε 4096 ψηφιακές τιμές bit. Καθώς παρουσιάζονται πολλά προβλήματα στην διακριτική του ικανότητα με αποτέλεσμα να λαμβάνονται λανθασμένες τιμές και δίνεται η δυνατότητα να τον αξιοποιήσουμε σε λιγότερα bit θα προχωρήσουμε στον προγραμματισμό του στα 10-bit με 1024 διακριτές ψηφιακές τιμές. Συμφώνα με το Σχήμα 8.9 παρατηρούμε πως τα ποδαράκια που μπορούν να χρησιμοποιηθούν για τον ADC είναι αρκετά, ανάμεσα τους επιλέχτηκε το ποδαράκι GPIO34 που αποκαλείται και ADC1 CH6. Επίσης γίνεται κατανοητό πως στο ποδαράκι Vin θα συνδεθεί η τροφοδοσία του ESP32 και στα ποδαράκια GND θα μπουν οι γειώσεις του κυκλώματος.



Σχήμα 8.9: Σχέδιο λειτουργίας pins του ESP32 DEVKIT V1 DOIT.

8.7 Επίλογος

Στο κεφάλαιο που προηγήθηκε παρουσιαστήκαν οι λόγοι επιλογής κάθε εξαρτήματος, στη συνέχεια έγινε περιγραφή της συνδεσμολογίας κάθε τομέα του κυκλώματος ξεχωριστά και με την ανάλυση τους επιτεύχθηκε η κατανόηση της λειτουργιάς του συνολικού κυκλώματος. Τέλος παρουσιάστηκε ο τρόπος σύνθεσης του ηλεκτρονικού προπονητή καθώς και οι διαδικασίες με τα απαραίτητα μετρά για την υλοποίηση του.

Κεφάλαιο 9ο: Προγραμματισμός ESP32

9.1 Εισαγωγή

Μετά την συνδεσμολογία του κυκλώματος έπεται ο προγραμματισμός του ESP32 προκειμένου να δέχεται να επεξεργάζεται και να αποστέλλει τα δεδομένα σε ένα Android κινητό.

9.2 Σύνδεση με υπολογιστή

Για τον προγραμματισμό του θα χρειαστεί η σύνδεση του με έναν υπολογιστή και στην συνέχεια η αναγνώριση του από το Adriano IDE, που επιλέχτηκε για την δημιουργία του προγράμματος που θα το θέτει σε λειτουργιά. Με την εξέλιξη της open source κοινότητας, και κατ' επέκταση των εργαλείων που υποστηρίζει, όπως το Arduino IDE, η διαδικασία την αναγνώρισης ενός μικροελεγκτή έχει απλοποιηθεί σε τεράστιο βαθμό. Έτσι, το μόνο που αρκεί από τον χρηστή είναι να συνδέσει με το κατάλληλο καλώδιο τον μικροελεγκτή που επιθυμεί να χρησιμοποιήσει και στην συνέχεια να ανοίξει το πρόγραμμα Adriano IDE.

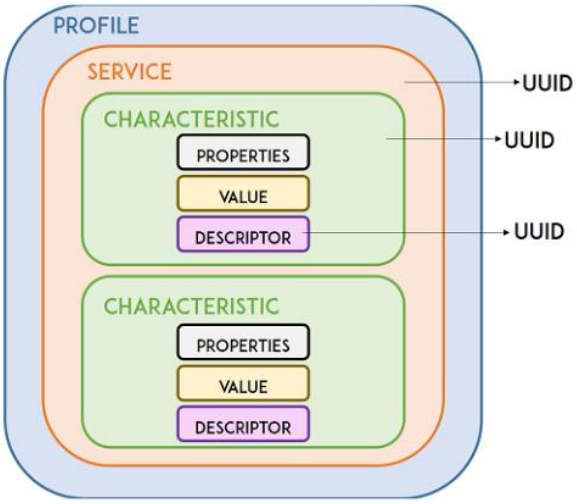
9.3 Μεταφορά δεδομένων με BLE

Το BLE (Bluetooth Low Energy) πρόκειται για μια εξέλιξη της τεχνολογίας Bluetooth που στοχεύει σε εφαρμογές για την λειτουργιά τους στο Διάδικο των Πραγμάτων (Internet Of Things). Για να επιτευχτεί αυτό θα πρέπει οι εφαρμογές να είναι ικανές σε χαμηλή κατανάλωση ενεργείας με διαχείριση δεδομένων μικρού όγκου, να έχουν την ικανότητα ασύρματης μετάδοσης δεδομένων και ταυτόχρονα να μπορούν να συνδέονται με πολλές συσκευές και σε μικρή απόσταση μεταξύ τους. Ο τρόπος που καταναλώνουν λιγότερη ενεργεία επιτυγχάνεται με την διαρκή τους κατάσταση σε sleep mode, εκτός από την στιγμή που μεταφέρουν δεδομένα.

	Bluetooth Low Energy (LE)	Bluetooth Basic Rate/ Enhanced Data Rate (BR/EDR)
Optimized For...	Short burst data transmission	Continuous data streaming
Frequency Band	2.4 GHz (2.402 GHz to 2.480 GHz)	2.4 GHz (2.402 GHz to 2.480 GHz)
Channels	40 channels with 2 MHz spacing (3 advertising channels/37 data channels)	79 channels with 1 MHz spacing
Channel Usage	Adaptive Frequency Hopping (AFH) 1600 hops/sec	Adaptive Frequency Hopping (AFH) 1600 hops/sec
Modulation	GFSK	GFSK, π/4 DQPSK, 8DPSK
Power Consumption	~0.01x to 0.5x of reference (depending on use case)	1 (reference value)
Data Rate	LE 2M PHY: 2 Mb/s LE 1M PHY: 1 Mb/s LE Coded PHY (S=2): 500 Kb/s LE Coded PHY (S=8): 125 Kb/s	EDR PHY (8DPSK): 3 Mb/s EDR PHY (π/4 DQPSK): 2 Mb/s BR PHY (GFSK): 1 Mb/s
Max Tx Power*	Class 1: 100 mW (+20 dBm) Class 1.5: 10 mW (+10 dbm) Class 2: 2.5 mW (+4 dBm) Class 3: 1 mW (0 dBm)	Class 1: 100 mW (+20 dBm) Class 2: 2.5 mW (+4 dBm) Class 3: 1 mW (0 dBm)
Network Topologies	Point-to-Point (including piconet) Broadcast Mesh	Point-to-Point (including piconet)

Σχήμα 9.1: Διαφορές BLE και κλασικού BT.

Ένα επίσης σημαντικό χαρακτηριστικό του BLE είναι η ιεραρχία που δομούνται τα δεδομένα μιας συσκευής κατά την μεταφορά τους και ονομάζεται GATT (Generic Attributes). Με βάση το Σχήμα 9.2 όλα τα δεδομένα περιλαμβάνονται μέσα σε ένα προφίλ (Profile) το όποιο μπορεί να αποτελείται από πολλές υπηρεσίες (Service) . Η κάθε υπηρεσία περιλαμβάνει τουλάχιστον ένα χαρακτηριστικό (Characteristic) που περιέχει υποχρεωτικά την τιμή του (Value) και πληροφορίες για αυτό (Properties) και προαιρετικά επιπλέον δεδομένα (Descriptor) για την τιμή και το χαρακτηριστικό. Επίσης κάθε υπηρεσία, χαρακτηριστικό και τα επιπλέον δεδομένα έχουν ένα μοναδικό αριθμό ορισμένο από τα πρότυπα του Bluetooth Low Energy. Ο αριθμός αυτό είναι 128 bit και ονομάζεται UUID (Universally Unique Identifier).



Σχήμα 9.2: Δομή δεδομένων BLE.

9.4 Ανάλυση κώδικα Arduino

Για αρχή προστίθενται οι κατάλληλες βιβλιοθήκες για την χρήση των απαραιτήτων συναρτήσεων ώστε ο μικροελεγκτής να μπορεί να λειτουργήσει σαν υπηρέτης (server) και να μπορεί να διαφημιστεί στις κατάλληλη συχνότητα (που είναι τα 2.4GHz όπως και του Wi-Fi) για τις υπόλοιπες συσκευές BLE, που στην συνέχεια θα συνδεθεί μέσω των πρωτόκολλων που ακολουθεί η τεχνολογία αυτή.

Με το ορισμό των SERVICE_UUID και CHARACTERISTIC_UUID δηλώνουμε την υπηρεσία και το χαρακτηριστικό της ώστε να αναγνωριστούν από τις άλλες συσκευές BLE.

Στη συνέχεια δηλώνουμε τις μεταβλητές που θα χρησιμοποιηθούν στο πρόγραμμα καθώς και την συνάρτηση “kilo” που θα δηλώνει ποια bit που λαμβάνουμε από τον αισθητήρα SG μέσω του ADC ανάλογα σε ποια κιλά δύναμης που εφάρμοσε ο αθλητής στον στόχο. Η συνάρτηση λειτούργει με “if” και “else if”.

Έπειτα στην συνάρτηση “setup” προσδιορίζονται οι λειτουργιές του ESP32 προκειμένου να διαφημίζεται σαν υπηρέτης BLE, να στέλνει και να παίρνει δεδομένα καθώς και να ειδοποιεί για την κάθε του κίνηση, στην συνδεδεμένη σε αυτόν συσκευή. Επίσης πραγματοποιείται η “χειραψία” μεταξύ των συσκευών που θα επικοινωνεί.

Η συνάρτηση “loop” είναι που πραγματοποιείται το μεγαλύτερο μέρος της εφαρμογής αφού σε κάθε εκκίνηση είναι η μονή συνάρτηση που τρέχει ασταμάτητα. Στην περίπτωση του προγράμματος που αναλύουμε ξεκάνει με μια καθυστέρηση *1ms* για την ανασυγκρότηση της μνήμης του μικροελεγκτή και την αποφυγή λαθών από προηγούμενες λούπες. Έπειτα ελέγχει για νέες εντολές συσκευές που είναι ενωμένο. Αν η εντολή είναι “spdstrt”, “pwrstrt”, “frstrt” ή “gmstrt” τοτε το αναγκάζει να ελέγχει αν οι τιμές που λαμβάνει από τον ADC είναι κατάλληλες προς μέτρηση, δηλαδή αν ο αισθητήρας SG έχει κάποιο ερέθισμα, μεχρι να ληφθεί κάποια από τις εντολές “spdstp”, “pwrstp”, “frstp”, “gmstp” που το αναγκάζουν να σταματήσει την αποστολή δεδομένων. Άμα το ερέθισμα είναι αρκετά μεγάλο τοτε για *100ms* μέτριοι 1 χτύπημα ανά ms συγκρίνει ποιο είναι το μεγαλύτερο και το στέλνει στην συσκευή που το απαιτεί με εντολή, άμα το ερέθισμα δεν είναι μεγάλο τοτε συνεχίζει τις λούπες του.

9.5 Επίλογος

Σε αυτό το κεφάλαιο έγινε ανάλυση της μεταφοράς δεδομένων μέσω BLE και των κύριων χαρακτηριστικών του καθώς σημειωθήκαν και οι βασικές διαφορές σε σχέση με το κλασσικό Bluetooth. Στη συνέχεια πραγματοποιήθηκε ανάλυση του κώδικα που προγραμματίστηκε να λειτούργει ο μικροελεγκτής προκειμένου να λαμβάνει εντολές και στέλνει δεδομένα στις συνδεδεμένες σε αυτό συσκευές.

Κεφάλαιο 10ο: Προγραμματισμός εφαρμογής Android

Εισαγωγή

Στο κεφάλαιο αυτό ακόλουθη ανάλυση του κώδικα που φτιάχτηκε με την τεχνική Drag-and-Drop που παρέχει η πλατφόρμα δημιουργίας Android εφαρμογών, MIT App Inventor.

Η εφαρμογή είναι ικανή να στέλνει εντολές στον μικροελεγκτή μέσω BLE και να δέχεται τα αποτελέσματα σε πραγματικό χρόνο. Τα επεξεργάζεται και μετά την εμφάνιση τους στην οθόνη του χρηστή, προχωρεί στην αποθήκευση τους μέσα σε μια βάση δεδομένων που βρίσκεται τοπικά (local) στο Android Smartphone που έχει εγκατασταθεί η εφαρμογή, για την ανάκτηση τους οποιαδήποτε στιγμή. Παρέχει τέσσερα διαφορετικά ειδή προπόνησης εκ των οποίων το ένα είναι παιχνίδι εναντία στον χρόνο. Τα υπόλοιπα τρία αφορούν την καταγραφή των ικανοτήτων του αθλητή σε θέματα δύναμης και ταχύτητας.

Στο σύνολο η εφαρμογή αποτελείται από σχεδόν 800 μπλοκ κώδικα.

Ανάλυση εφαρμογής MIT App Inventor

Με την εκκίνηση της εφαρμογής ανοίγει αυτόματα το Bluetooth του Smartphone και στη συνέχεια παρουσιάζεται η επιλογή σύνδεσης στον μικροελεγκτή. Μόλις το Android συνδεθεί επιτυχώς με τον μικροελεγκτή ο χρήστης έχει να επιλέξει ανάμεσα σε πέντε επιλογές.

Στην πρώτη επιλογή πραγματοποιείται καταγραφή της ικανότητας της δύναμης, παρουσιάζοντας την δύναμη σε kg όλων των χτυπημάτων που πραγματοποιήθηκαν από την εκκίνηση της καταγραφής.

Στη δεύτερη, γίνεται καταγραφή της ικανότητας της ταχύτητας. Δηλαδή μετρούνται τα χτυπήματα στον στόχο μέσα σε συγκεκριμένο χρόνο και καταγράφοντας τον μέσο ορό των χτυπημάτων ανά δευτερόλεπτο και ανά λεπτό.

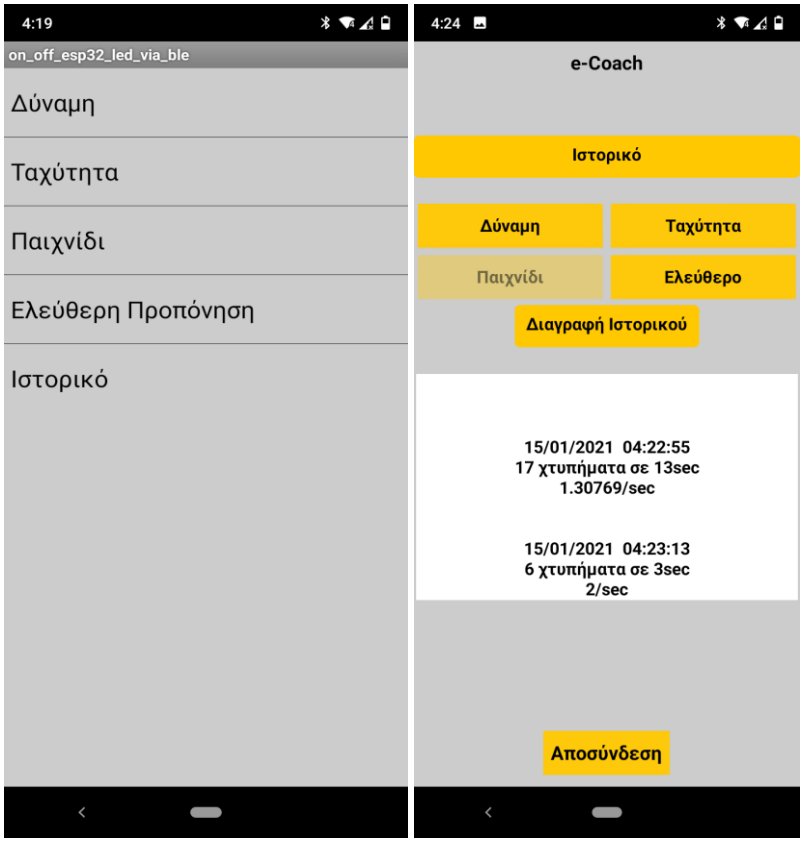
Στην τρίτη επιλογή βρίσκεται ένα παιχνίδι. Με την εκκίνηση του παιχνιδιού, η μπάρα που βρίσκεται στο πάνω μέρος της οθόνης μειώνεται ανά δευτερόλεπτο, ο αθλητής έχει την δυνατότητα να αυξήσει τον πήχη της μπάρας βαρώντας τον στόχο που βρίσκεται ο αισθητήρας. Το παιχνίδι τελειώνει μόλις η μπάρα φτάσει σε ένα από τα δυο άκρα της και τα δεδομένα που εμφανίζονται αφορούν την δύναμη και την ταχύτητα που είχε ο αθλητής κατά την διάρκεια του παιχνιδιού καθώς και τον χρόνο που διήρκησε το παιχνίδι.

Η τέταρτη επιλογή πρόκειται για μια προσομοίωση ελεύθερης προπόνησης καταγράφοντας την δύναμη.

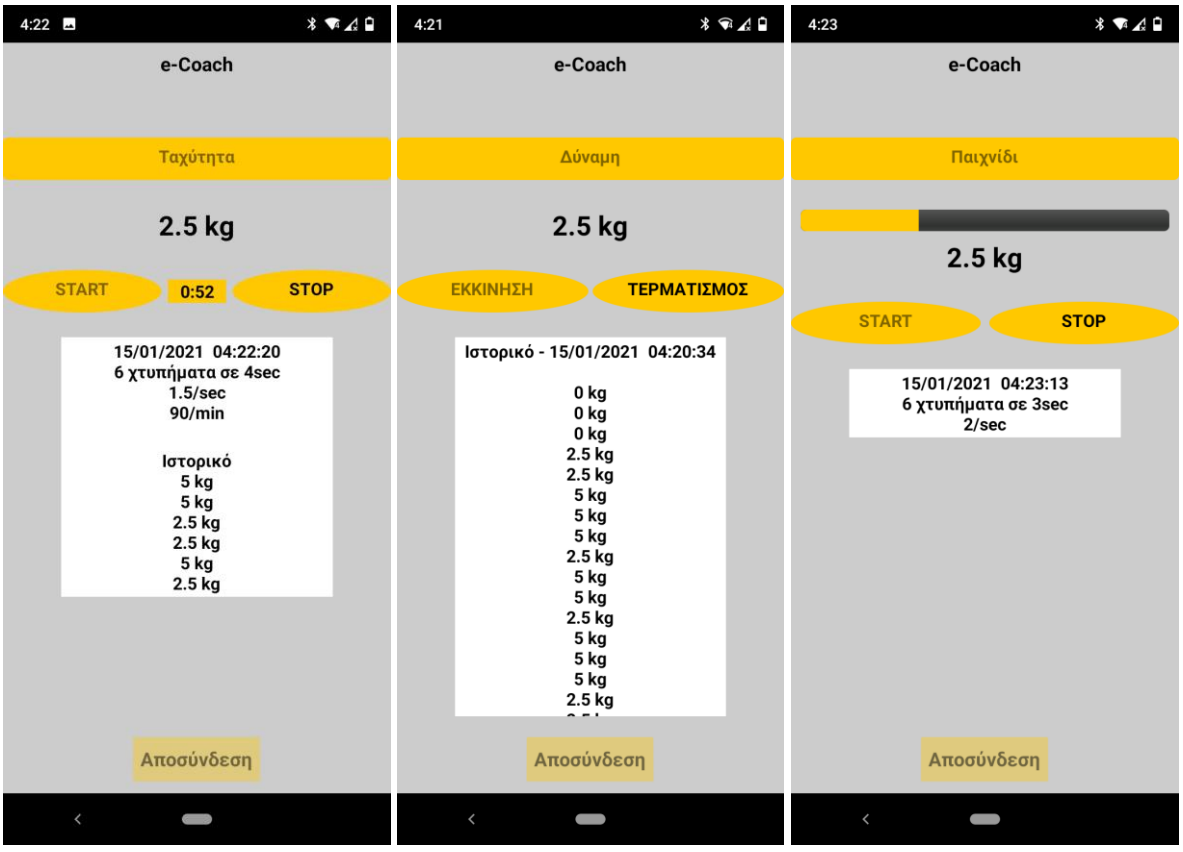
Με την λήξη κάθε προπόνησης είτε εξαιτίας του χρόνου, είτε διακοπής από τον χρηστή, όλα τα δεδομένα αποθηκεύονται στην βάση δεδομένων με χρονολογική σειρά.

Η πέμπτη και τελευταία επιλογή αφορά το ιστορικό των προπόνησαν που δίνει την δυνατότητα στον χρηστή να κάνει ανάδρομη σε όλα τα δεδομένα που έχουν καταγράφει ώστε να προχωρήσει με την αξιολόγηση της κάθε του προσπάθειας.

Παρακάτω φαίνονται οι εικόνες από κάθε επιλογή.



Σχήμα 10.1: Μενού εφαρμογής Android (αριστερά), εμφάνιση ιστορικού της επιλογής παιχνιδιού (δεξιά).



Σχέδιο 10.2: Επιλογή ταχύτητας (αριστερά), επιλογή δύναμης (μέση), επιλογή παιχνίδι (δεξιά).

Επίλογος

Το κεφάλαιο αυτό αναφέρθηκε στην λειτουργιά της εφαρμογής Android που λαμβάνει, επεξεργάζεται, εμφανίζει και αποθηκεύει τις ικανότητες ενός αθλητή. Αναλύθηκε η δομή της και οι πέντε βασικές επιλογές που έχει.

ΒΙΒΛΙΟΓΡΑΦΙΑ

[1] Stefan Keil, *Technology and Practical Use of Strain Gages: With Particular Consideration of Stress Analysis Using Strain Gages*, 1st ed.: Ernst & Sohn, 2017.

[2] Richard L. Pendleton and Mark E. Tuttle, *Manual on Experimental Methods for Mechanical Testing of Composites*, 1st ed.: Springer; Softcover reprint of the original , 1989.

[3] Charles Kitchin and Lew Counts, *A Designer's Guide to Instrumentation Amplifiers*, 3rd ed. USA: Analog Devices, 2006.

[4] Albert Malvino and David Bates, *Electronic Principles*, 8th ed. New York, USA: McGraw-Hill Education, 2016.

[5] Jacob Fraden, *Handbook of Modern Sensors Physics, Designs and Applications*, 5th ed. Switzerland: Springer International Publishing, 2016.

[6] Μιχάλης Σπάσος and Κώστας Αμοιρίδης, *Σύγχρονες Εφαρμογές Αναλογικών Ηλεκτρονικών*. Θεσσαλονίκη, Ελλάδα: Εκδόσεις ΑΪΒΑΖΗ, 2015.

[7] Μιχάλης Ν. Σπάσος, *ΑΝΑΛΟΓΙΚΗ ΕΠΕΞΕΡΓΑΣΙΑ ΣΗΜΑΤΩΝ ΑΙΣΘΗΤΗΡΙΩΝ*.: Εκδόσεις ΑΪΒΑΖΗ, 2018.

[8] KEITHLEY A Tektronix Company, *Low Level Measurements Handbook*, 7th ed. USA: Tektronix.

[9] Rui Santos and Sara Santos, *LEARN ESP32 with Arduino IDE*.

[10] Neil Kolban, *Kolban's Book on ESP32*., 2017.

[11] ANALOG DEVICES, *MT-O61 TUTORIAL Instrumentation Amplifiers (In-Amp) Basics*.: ANALOG DEVICES.

[12] NATIONAL INSTRUMENTS, *Application Note 078 Strain Gauge Measurement - A Tutorial*.: NATIONAL INSTRUMENTS.

[13] James Karki, "SLOA034 Signal Conditioning Wheatstone Resistive Bridge Sensors," Application Report 1999.

[14] Karl Hoffmann, *Applying the Wheatstone Bridge Circuit*.: HBM.

[15] SILICON LABORATORIES, *"SINGLE-CHIP USB TO UART BRIDGE" CP2102*.

[16] Espressif, *ESP32-S2 TECHNICAL REFERENCE MANUAL*.

[17] Espressif, *ESP32-WROOM-32 DATASHEET*.

[18] TEXAS INSTRUMENTS, *LM340, LM340A and LM7805 Family Wide VIN 1.5-A Fixed Voltage Regulators*.

[19] TEXAS INSTRUMENTS, *μA741 General-Purpose Operational Amplifiers*.

[20] ANALOG DEVICES, *"Single and Dual-Supply, Rail-to-Rail, Low Cost Instrumentation Amplifier" AD623*.

[21] TEXAS INSTRUMENTS, *"Circuit for Driving a Switched-Capacitor SAR ADC With an Instrumentation Amplifier" Analog Engineer's Circuit: ADCs*.

[22] Sensing Systems. (2016, January) Sensing Systems. [Online]. https://www.sensing-systems.com/blog/strain-gauge-technology-in-field-testing

[23] Cadence PCB solutions. CADENCE. [Online]. https://resources.pcb.cadence.com/blog/2019-what-is-a-guard-ring-and-how-to-design-it-properly

[24] A. Alvin Barlian, Woo-Tae Park, Jr Joseph R. Mallon, Ali J. Rastegar, and Beth L. Pruitt. (2009) US National Library of Medicine. [Online]. https://www.ncbi.nlm.nih.gov/pmc/articles/PMC2829857/

[25] MIT. MIT Musseum. [Online]. http://museum.mit.edu/150/82

[26] Editors of Encyclopaedia Britannica. Britannica. [Online]. https://www.britannica.com/technology/strain-gauge

[27] Sensorland. Sensorland. [Online]. https://www.sensorland.com/HistPage003.html

[28] National Instruments. National Instruments. [Online]. https://www.ni.com/en-us/innovations/white-papers/07/measuring-strain-with-strain-gages.html

[29] HBM. HBM. [Online]. https://www.hbm.com/en/4314/selecting-adhesives-for-strain-gauge-installation/

[30] Juan Antonio. AppInventorMIT. [Online]. https://community.appinventor.mit.edu/t/ble-esp32-bluetooth-send-receive-arduino-ide/1980

[31] Espressif. Espressif. [Online]. https://docs.espressif.com/projects/esp-idf/en/release-v4.1/hw-reference/modules-and-boards.html#esp32-wroom-32

[32] Arduino. Arduino. [Online]. https://www.arduino.cc/en/Reference/ArduinoBLE

[33] Advanced Monolithic Systems, *AMS1117*.

Α: ΚΩΔΙΚΑΣ ARDUINO

```
#include <BLEDevice.h>

#include <BLEUtils.h>

#include <BLEServer.h>



#define SERVICE_UUID        "4fafc201-1fb5-459e-8fcc-c5c9c331914b"

#define CHARACTERISTIC_UUID "beb5483e-36e1-4688-b7f5-ea07361b26a8"



  String analogSensor;

  int sentFlag = 0; //flag for the "sent" loop

  int i = 0;

  int s[100];

  int endOfStrikeFlag = 1;

  int beginStrike = 0;

  int maxStrike = 0;

  int strikeTime = 0;

  std::string androidtxt;

  BLECharacteristic *pCharacteristic;



float kilo (int x){

  float y = 0;
```

```
if ((x >= 600) and (x < 620)){
  y = 1;
}
else if ((x >= 620) and (x < 650)){
  y = 2;
}
else if ((x >= 650) and (x < 670)){
  y = 3;
}
else if ((x >= 670) and (x < 690)){
  y = 4;
}
else if ((x >= 690) and (x < 710)){
  y = 5;
}
else if ((x >= 710) and (x < 730)){
  y = 6;
}
else if ((x >= 730) and (x < 750)){
  y = 7;
}
```

```
else if ((x >= 750) and (x < 770)){
  y = 8;
}
else if ((x >= 770) and (x < 790)){
  y = 9;
}
else if ((x >= 790) and (x < 810)){
  y = 10;
}
else if ((x >= 810) and (x < 830)){
  y = 11;
}
else if ((x >= 830) and (x < 850)){
  y = 12;
}
else if ((x >= 850) and (x < 870)){
  y = 13;
}
else if ((x >= 870) and (x < 890)){
  y = 14;
}
```

```
else if ((x >= 890) and (x < 910)){
  y = 15;
}
else if ((x >= 910) and (x < 930)){
  y = 16;
}
else if ((x >= 930) and (x < 950)){
  y = 17;
}
else if ((x >= 950) and (x < 970)){
  y = 18;
}
else if ((x >= 970) and (x < 990)){
  y = 19;
}
else if (x >= 995){
  y = 20;
}
else {
  y = 0;
}
```

```
  return y;
}
void setup() {
  Serial.begin(115200);
  pinMode(LED_BUILTIN, OUTPUT);
  analogReadResolution(10);

  BLEDevice::init("ESP32");
  BLEServer *pServer = BLEDevice::createServer();
  BLEService *pService = pServer->createService(SERVICE_UUID);
  pCharacteristic = pService->createCharacteristic(
                                    CHARACTERISTIC_UUID,
                                    BLECharacteristic::PROPERTY_READ |
                                    BLECharacteristic::PROPERTY_WRITE  |
                                    BLECharacteristic::PROPERTY_NOTIFY  |
                                    BLECharacteristic::PROPERTY_INDICATE
                                  );

  //pCharacteristic->setCallbacks(new MyCallbacks());
  pCharacteristic->setValue("Hello World");
  pService->start();
```

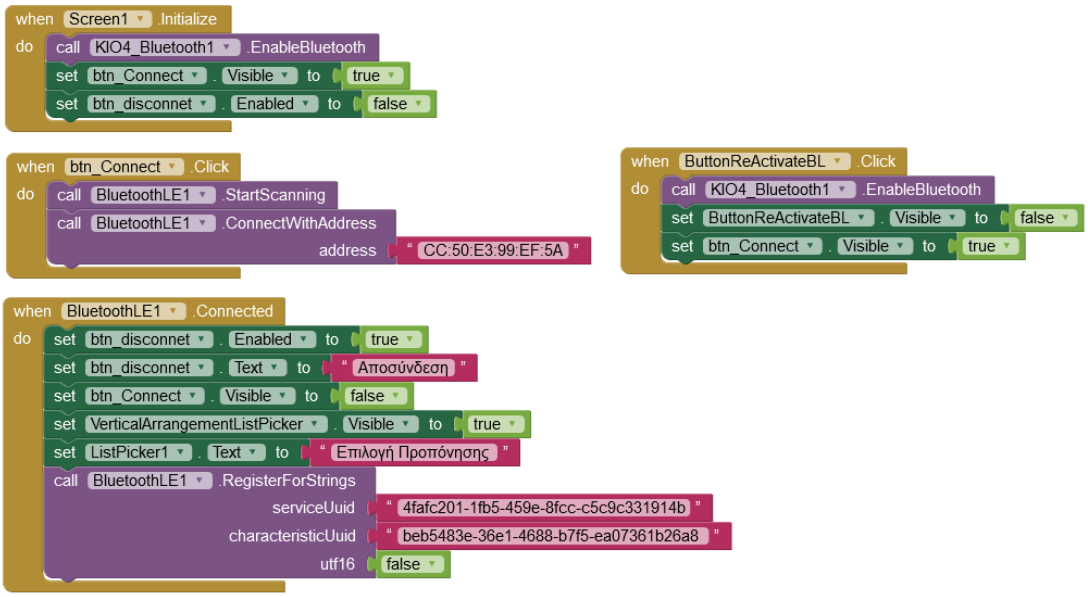
```
  BLEAdvertising *pAdvertising = pServer->getAdvertising();
  pAdvertising->start();

}

void loop() {
  // main code running every 10ms
  delay(1);
  std::string androidtxt = pCharacteristic->getValue();

//Check if the signal frome the andoid is for power or speed or free practice
  if (androidtxt.find("pwrstrt") != -1) {
    strikeTime = 99;    //Power comsumes more time and less strikes
    sentFlag = 1;
  }
  else if (androidtxt.find("spdstrt") != -1) {
    sentFlag = 2;
    strikeTime = 99;    //Speed consumes less time and more strikes
  }
  else if (androidtxt.find("gmstrt") != -1) {
```

```
  sentFlag = 3;

  strikeTime = 99;    //Game training

}

else if (androidtxt.find("frstrt") != -1) {

  sentFlag = 4;

  strikeTime = 99;    //Free training must have both speed and power, so strikes must be in the midle

}

//interupt measurement for FREE TRAINING

else if (((androidtxt.find("pwrstp") != -1) or (androidtxt.find("spdstp") != -1) or
    (androidtxt.find("gmstp") != -1) or (androidtxt.find("frstp") != -1)) and (sentFlag != 0)) {

  Serial.println("ANDROID DOES NOT WANT A VALUE");

  sentFlag = 0;

  digitalWrite(LED_BUILTIN, HIGH);   // turn the LED on (HIGH is the voltage level)

  delay(500);                       // wait for 500ms

  digitalWrite(LED_BUILTIN, LOW);    // turn the LED off by making the voltage LOW

  delay(500);                       //wait for 500ms

}



//Measure POWER

if ((sentFlag == 1) or (sentFlag == 2) or (sentFlag == 3) or (sentFlag == 4)) {

  delay(1);

  //Measure ONLY 1 strike
```

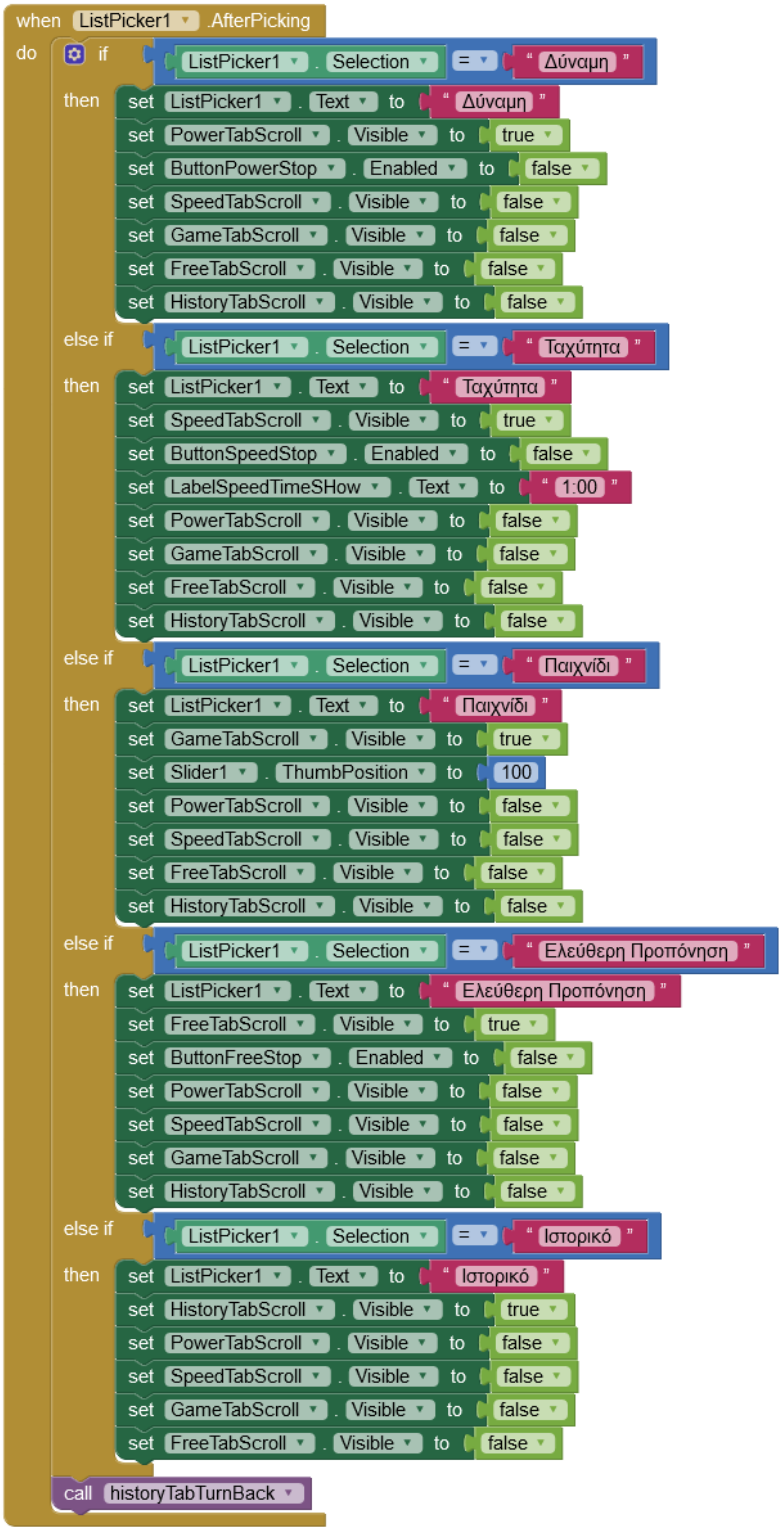
```
beginStrike = analogRead(34);
if (beginStrike >= 630){
  endOfStrikeFlag = 0;
}
while ((beginStrike >= 630) or (endOfStrikeFlag == 0)){
  // read the ongoing strike
  if (i >= strikeTime){
    endOfStrikeFlag = 1;
    i = 0;
    break;
  }
  s[i] = analogRead(34);
  if (maxStrike != s[i]){
    maxStrike = max(maxStrike, s[i]);   //get the max value on each iteration
  }
  s[i] = 0;                          //zeroing each cell of the array to use it in the next strike
  delay(1);
  i = i + 1;
  beginStrike = analogRead(34);
}
if (maxStrike >= 630){
```

```
      analogSensor = String(kilo(maxStrike));
      pCharacteristic->setValue(analogSensor.c_str());     //sent analogSensor value on android
      pCharacteristic->notify();
      delay(5);
      Serial.println(maxStrike);
      Serial.println(maxStrike);
      Serial.println(maxStrike);
    }
    maxStrike = 0;
  }
}
```

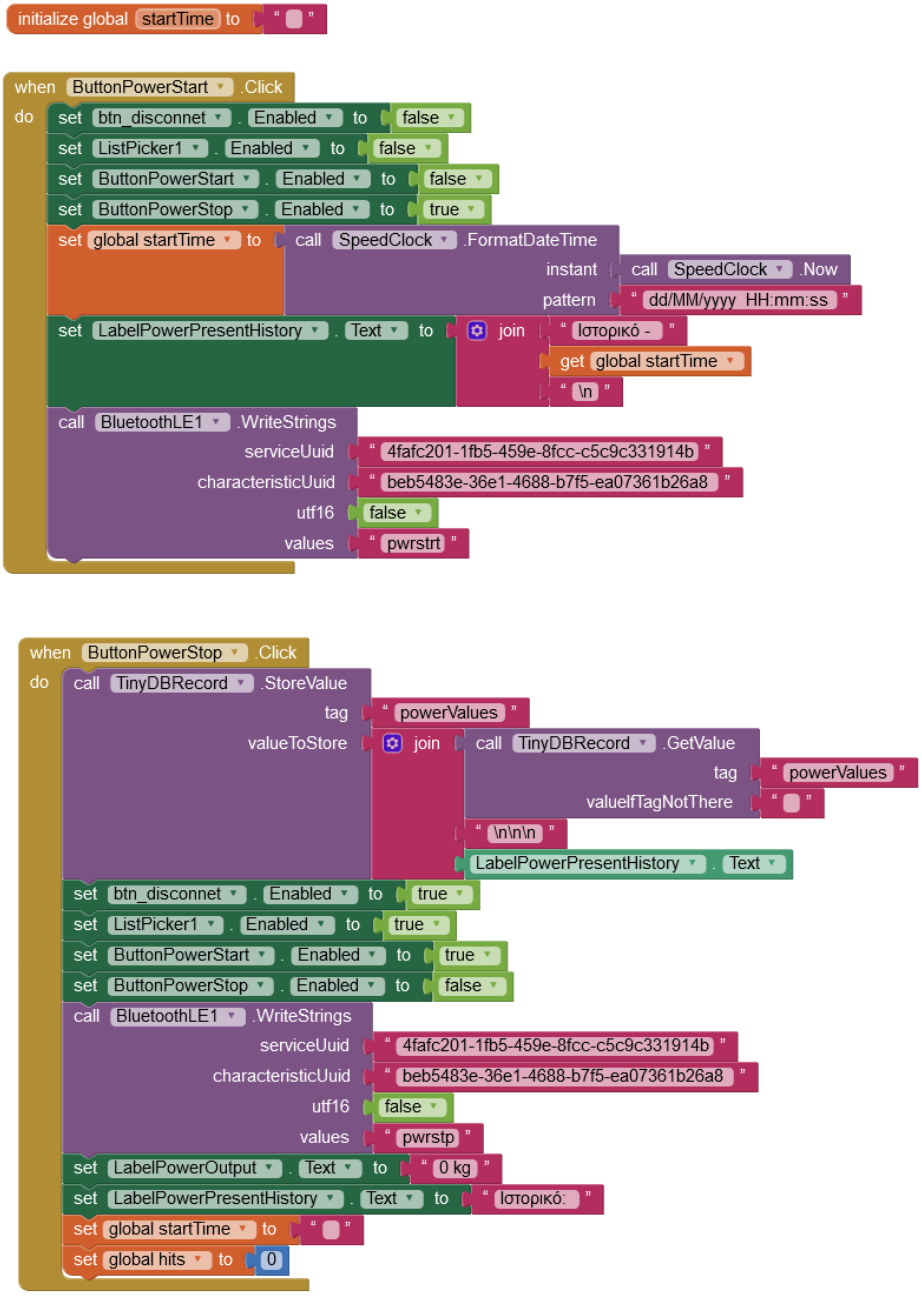
Β: ΜΠΛΟΚ ΚΩΔΙΚΑΣ ΤΟΥ MIT APP INVENTOR



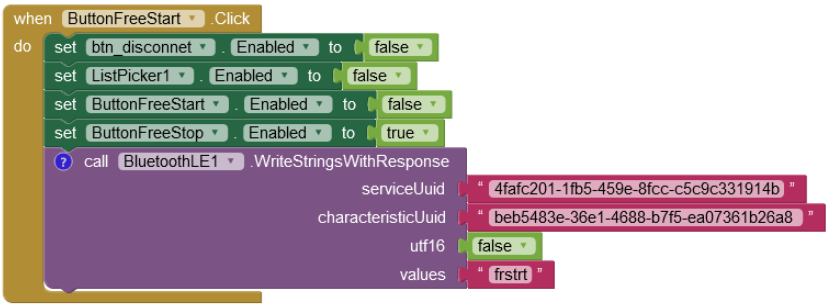
(ΕΚΚΙΝΗΣΗ ΕΦΑΡΜΟΓΗΣ)

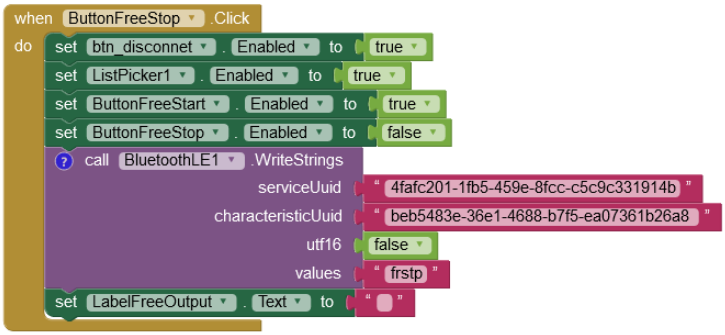

when ListPicker1 .AfterPicking
do if ListPicker1 . Selection = " Δύναμη "
then set ListPicker1 . Text to " Δύναμη "
set PowerTabScroll . Visible to true
set ButtonPowerStop . Enabled to false
set SpeedTabScroll . Visible to false
set GameTabScroll . Visible to false
set FreeTabScroll . Visible to false
set HistoryTabScroll . Visible to false
else if ListPicker1 . Selection = " Ταχύτητα "
then set ListPicker1 . Text to " Ταχύτητα "
set SpeedTabScroll . Visible to true
set ButtonSpeedStop . Enabled to false
set LabelSpeedTimeSHow . Text to " 1:00 "
set PowerTabScroll . Visible to false
set GameTabScroll . Visible to false
set FreeTabScroll . Visible to false
set HistoryTabScroll . Visible to false
else if ListPicker1 . Selection = " Παιχνίδι "
then set ListPicker1 . Text to " Παιχνίδι "
set GameTabScroll . Visible to true
set Slider1 . ThumbPosition to 100
set PowerTabScroll . Visible to false
set SpeedTabScroll . Visible to false
set FreeTabScroll . Visible to false
set HistoryTabScroll . Visible to false
else if ListPicker1 . Selection = " Ελεύθερη Προπόνηση "
then set ListPicker1 . Text to " Ελεύθερη Προπόνηση "
set FreeTabScroll . Visible to true
set ButtonFreeStop . Enabled to false
set PowerTabScroll . Visible to false
set SpeedTabScroll . Visible to false
set GameTabScroll . Visible to false
set HistoryTabScroll . Visible to false
else if ListPicker1 . Selection = " Ιστορικό "
then set ListPicker1 . Text to " Ιστορικό "
set HistoryTabScroll . Visible to true
set PowerTabScroll . Visible to false
set SpeedTabScroll . Visible to false
set GameTabScroll . Visible to false
set FreeTabScroll . Visible to false
call historyTabTurnBack

(ΜΕΝΟΥ)

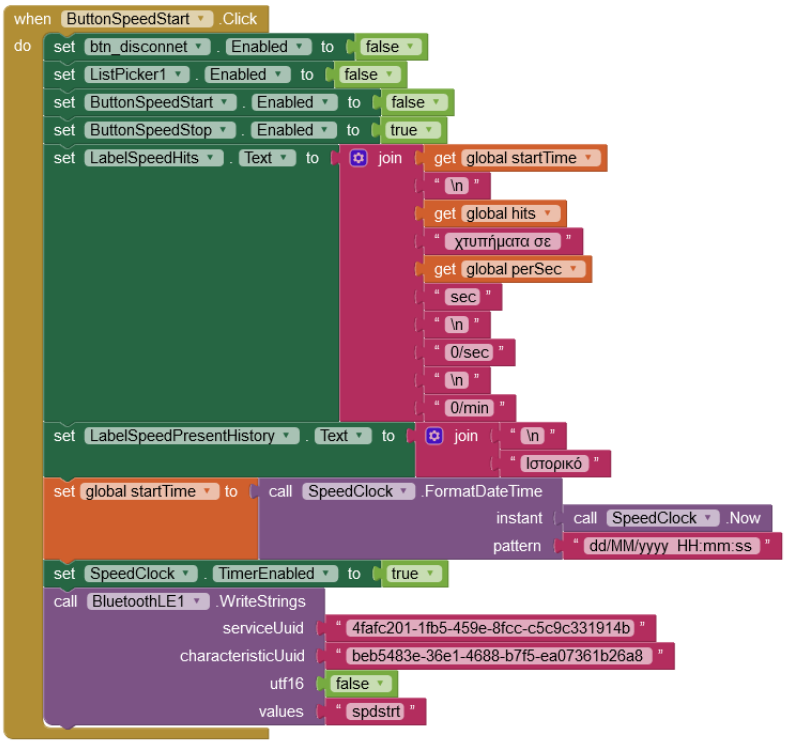


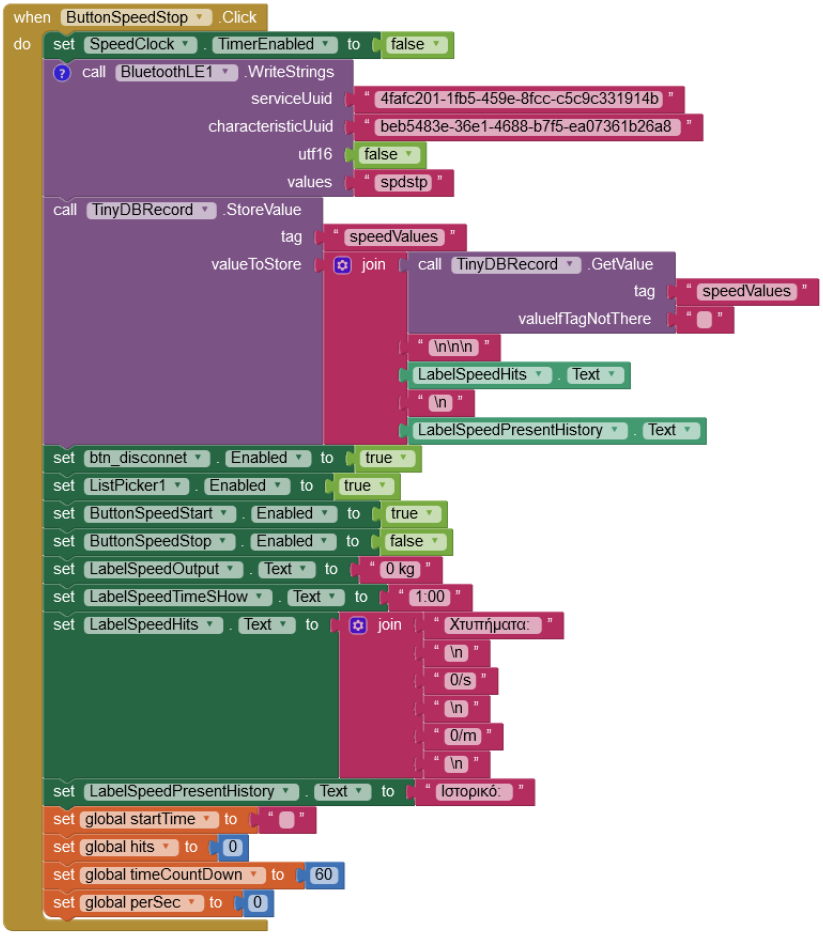
(ΚΟΥΜΠΙΑ ΔΥΝΑΜΗΣ)



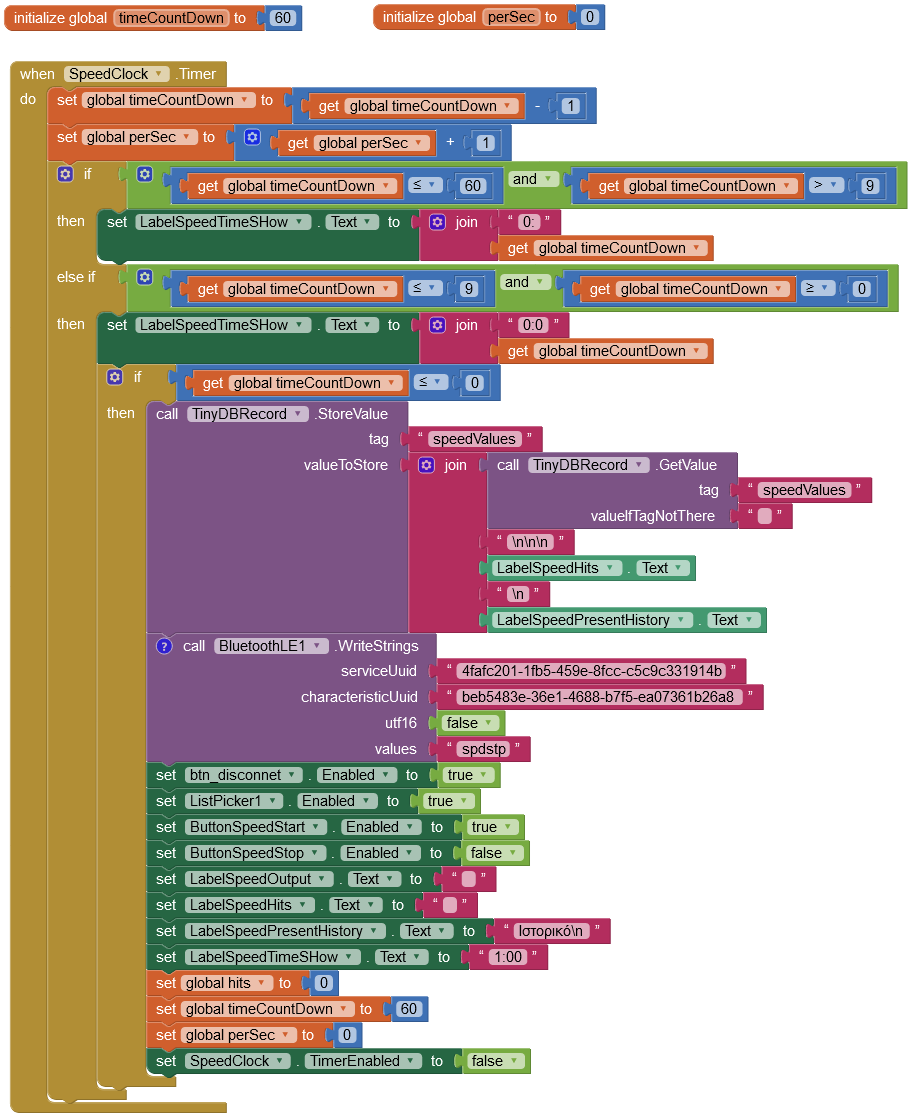


(ΚΟΥΜΠΙΑ ΕΛΕΥΘΕΡΗΣ ΠΡΟΠΟΝΗΣΗΣ)

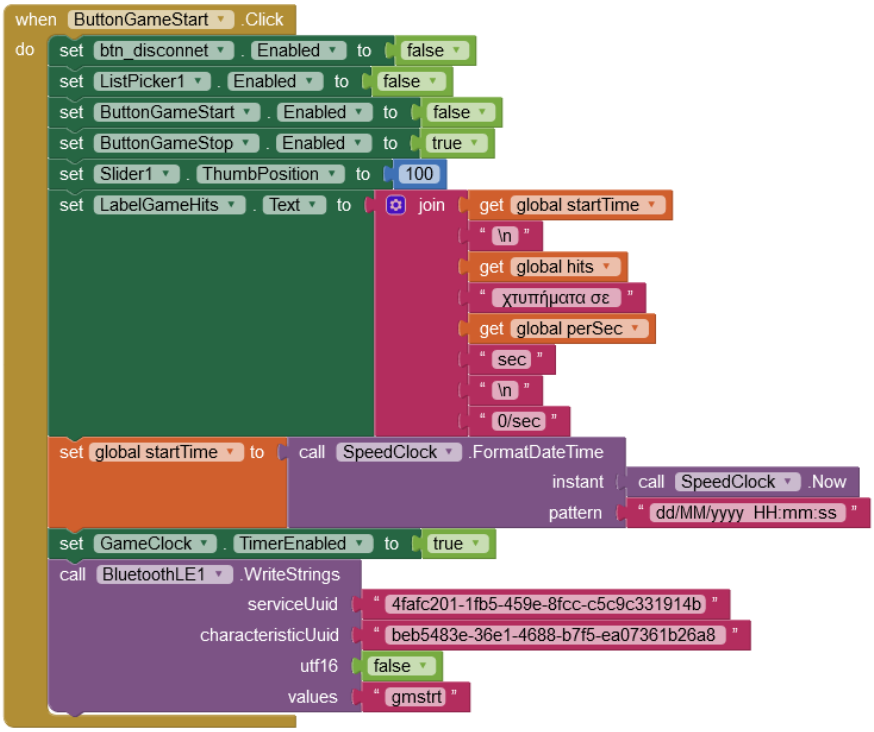


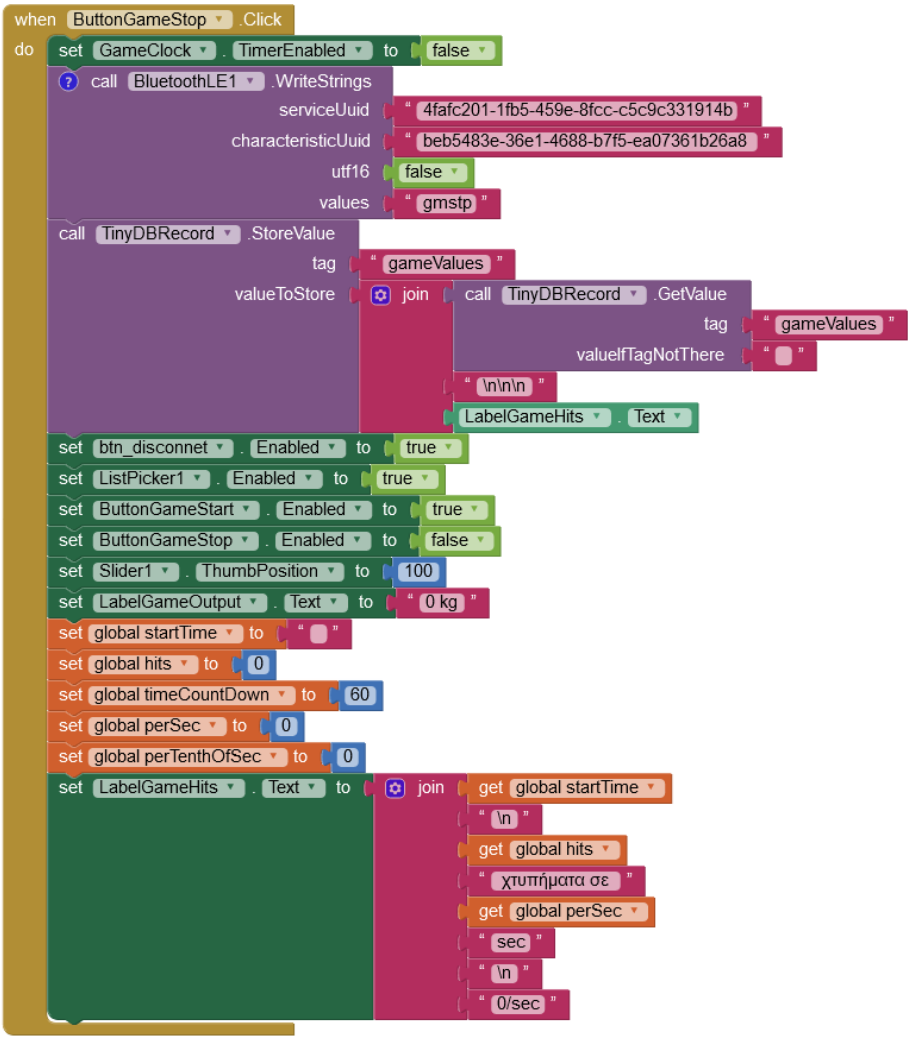


(ΚΟΥΜΠΙΑ ΤΑΧΥΤΗΤΑΣ)

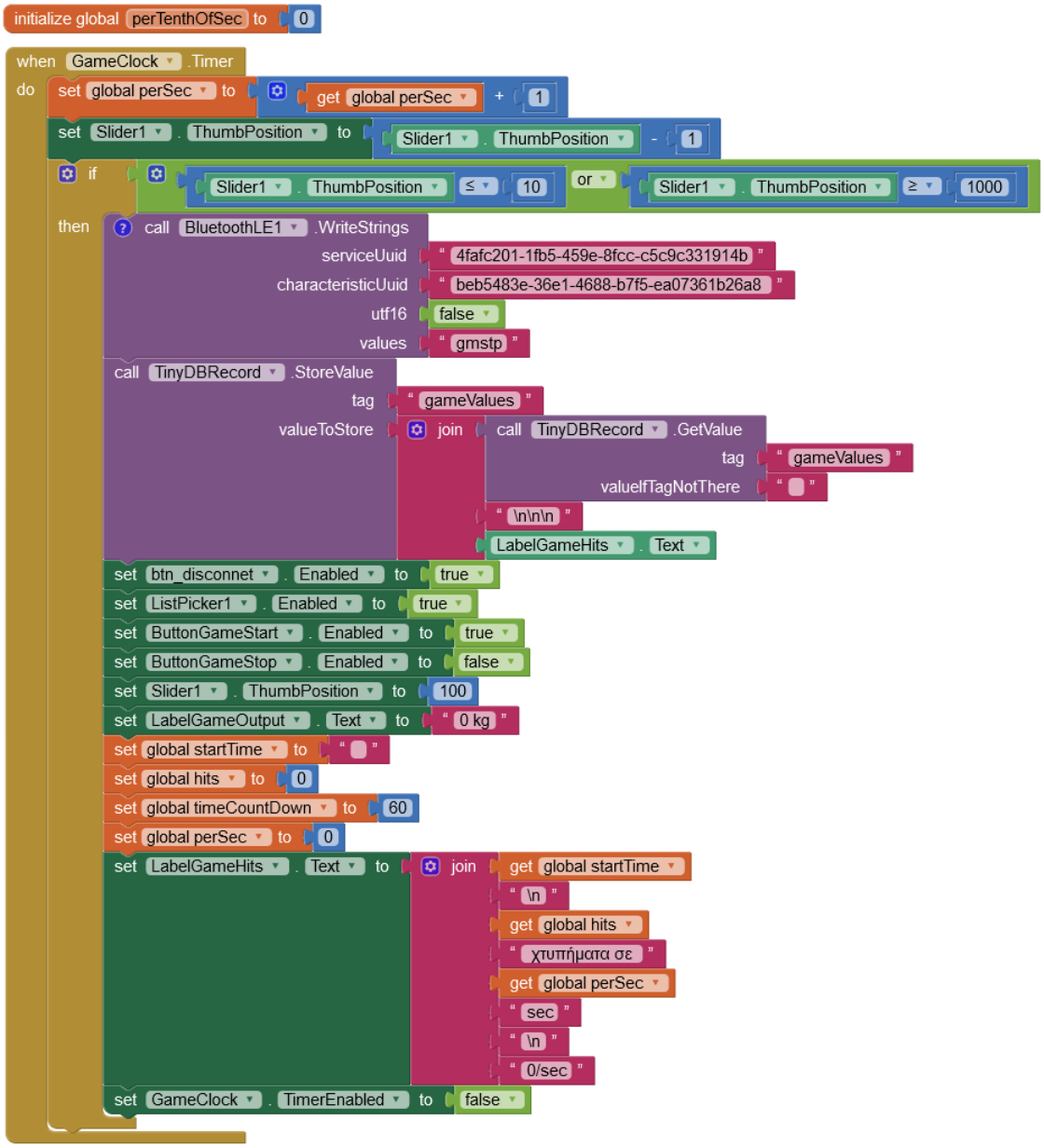


(ΧΡΟΝΟΣ ΤΑΧΥΤΗΤΑΣ)

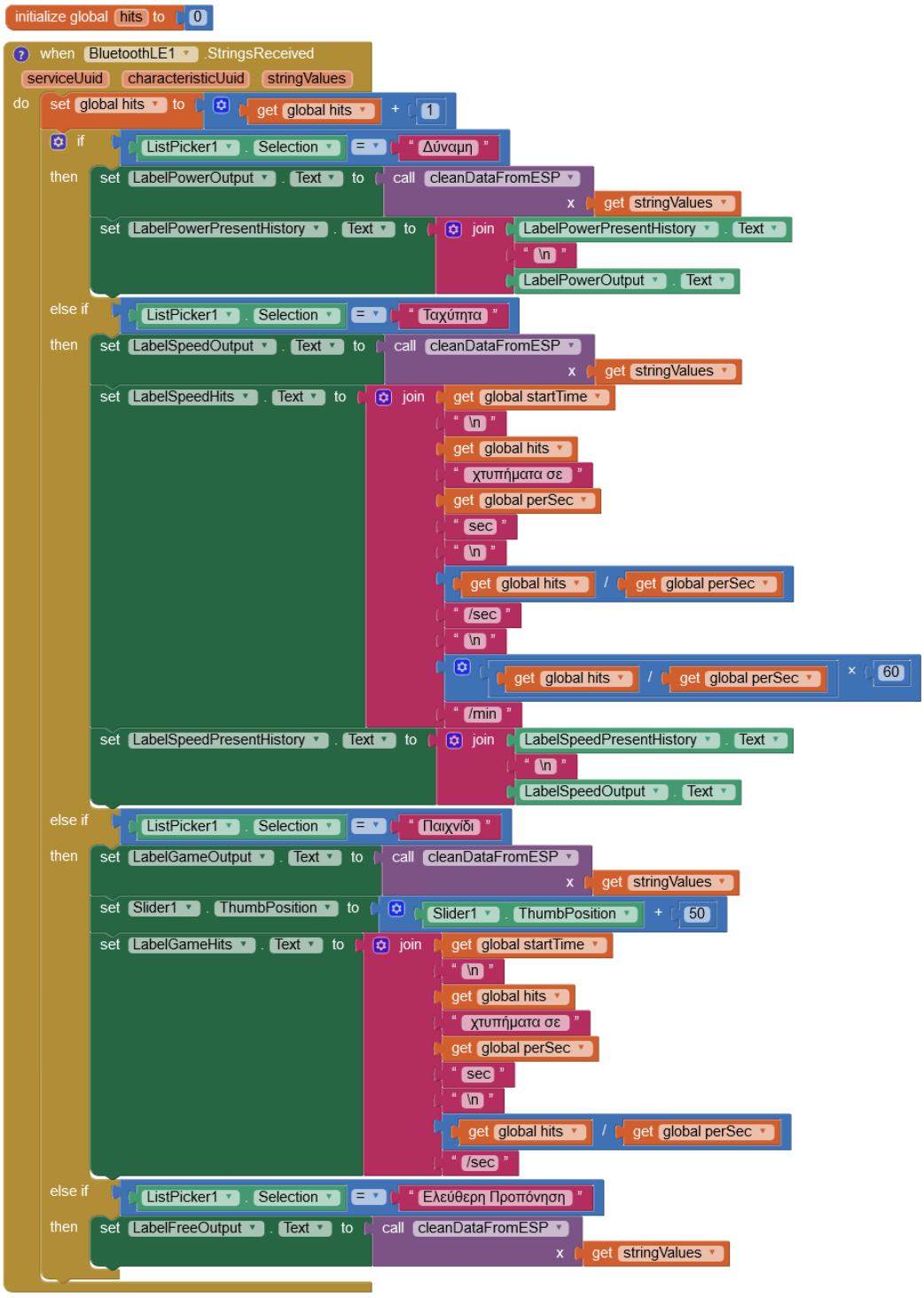

when ButtonGameStart .Click
do
set btn_disconnet . Enabled to false
set ListPicker1 . Enabled to false
set ButtonGameStart . Enabled to false
set ButtonGameStop . Enabled to true
set Slider1 . ThumbPosition to 100
set LabelGameHits . Text to join get global startTime
" \n "
get global hits
" χτυπήματα σε "
get global perSec
" sec "
" \n "
" 0/sec "
set global startTime to call SpeedClock .FormatDateTime
instant call SpeedClock .Now
pattern " dd/MM/yyyy HH:mm:ss "
set GameClock . TimerEnabled to true
call BluetoothLE1 .WriteStrings
serviceUuid " 4fafc201-1fb5-459e-8fcc-c5c9c331914b "
characteristicUuid " beb5483e-36e1-4688-b7f5-ea07361b26a8 "
utf16 false
values " gmstrt "


when ButtonGameStop .Click
do
set GameClock . TimerEnabled to false
call BluetoothLE1 .WriteStrings
serviceUuid " 4fafc201-1fb5-459e-8fcc-c5c9c331914b "
characteristicUuid " beb5483e-36e1-4688-b7f5-ea07361b26a8 "
utf16 false
values " gmstp "
call TinyDBRecord .StoreValue
tag " gameValues "
valueToStore join call TinyDBRecord .GetValue
tag " gameValues "
valueIfTagNotThere " "
" \n\n\n "
LabelGameHits . Text
set btn_disconnet . Enabled to true
set ListPicker1 . Enabled to true
set ButtonGameStart . Enabled to true
set ButtonGameStop . Enabled to false
set Slider1 . ThumbPosition to 100
set LabelGameOutput . Text to " 0 kg "
set global startTime to " "
set global hits to 0
set global timeCountDown to 60
set global perSec to 0
set global perTenthOfSec to 0
set LabelGameHits . Text to join get global startTime
" \n "
get global hits
" χτυπήματα σε "
get global perSec
" sec "
" \n "
" 0/sec "

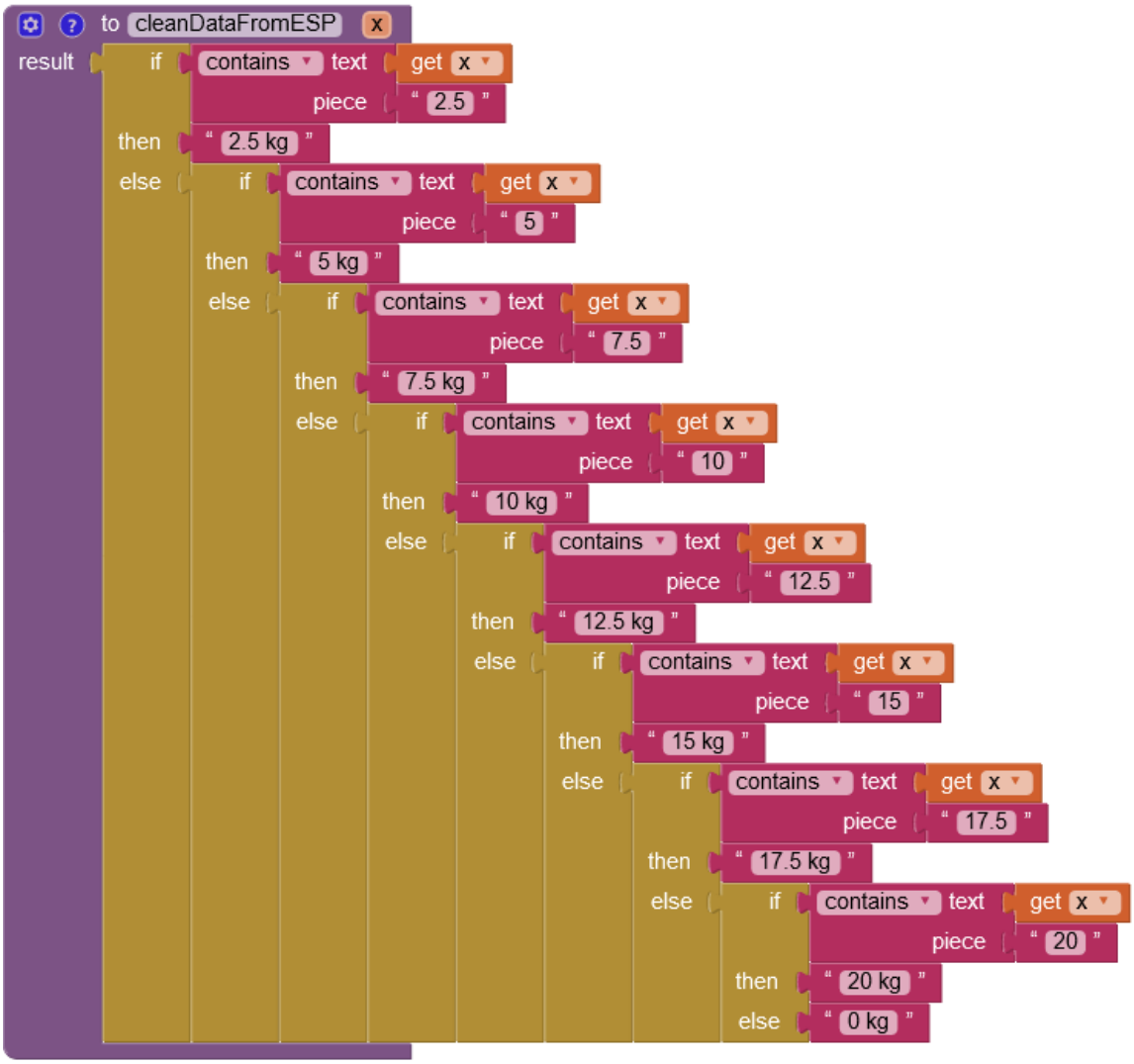
(ΚΟΥΜΠΙΑ ΠΑΙΧΝΙΔΙΟΥ ΜΕ ΜΠΑΡΑ)



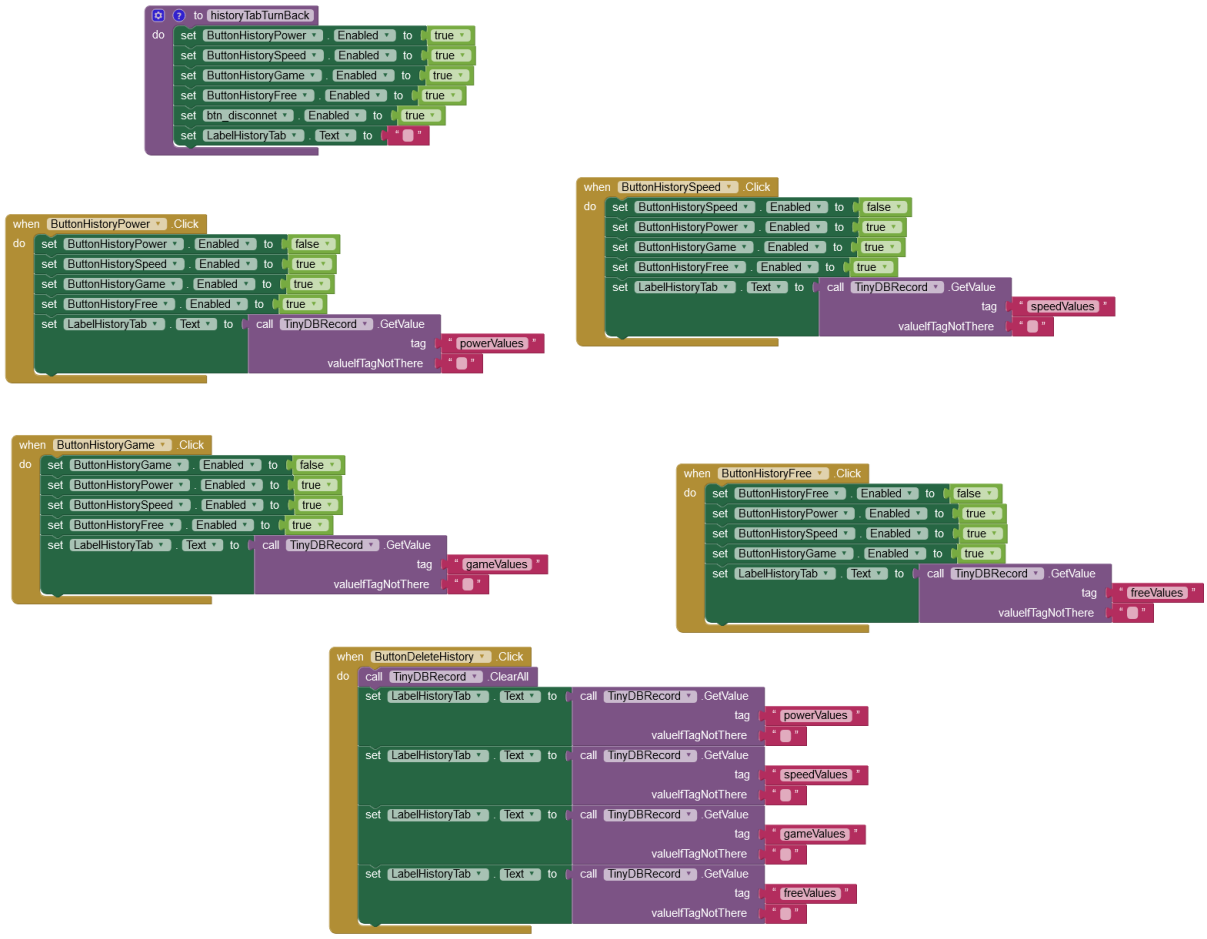
(ΧΡΟΝΟΣ ΠΕΧΝΙΔΙΟΥ ΜΠΑΡΑΣ)



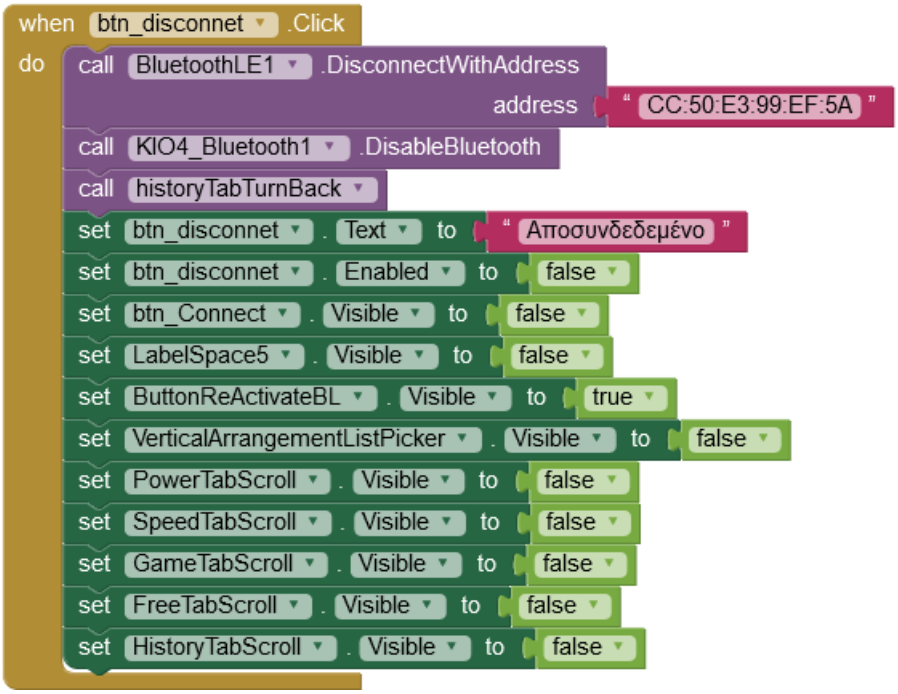
(ΕΜΦΑΝΗΣΗ ΔΕΔΟΜΕΝΩΝ)



(ΔΙΟΡΘΩΣΗ ΤΙΜΩΝ)



(ΕΜΦΑΝΗΣΗ ΙΣΤΟΡΙΚΟΥ)



(ΚΛΕΙΣΙΜΟ ΕΦΑΡΜΟΓΗΣ)