



ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ

ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ
ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ

ΤΟΜΕΑΣ ΕΠΙΚΟΙΝΩΝΙΩΝ, ΗΛΕΚΤΡΟΝΙΚΗΣ
ΚΑΙ ΣΥΣΤΗΜΑΤΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ

Ανάπτυξη συστήματος μετάδοσης και λήψης μετρήσεων αισθητήρων

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

Γεώργιος Δ. Μαχαίρας

Επιβλέπων Καθηγητής:

Ευάγγελος Β. Χριστοφόρου

Καθηγητής Ε.Μ.Π.

Αθήνα, Μάρτιος 2020



ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ

ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ
ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ

ΤΟΜΕΑΣ ΕΠΙΚΟΙΝΩΝΙΩΝ, ΗΛΕΚΤΡΟΝΙΚΗΣ
ΚΑΙ ΣΥΣΤΗΜΑΤΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ

Ανάπτυξη συστήματος μετάδοσης και λήψης μετρήσεων αισθητήρων

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

Γεώργιος Μαχαίρας

Εγκρίθηκε από την τριμελή εξεταστική επιτροπή την 5^η Μαρτίου 2020.

.....
Ευάγγελος Χριστοφόρου
Καθηγητής Ε.Μ.Π

.....
Ιωάννης Γκόνος
Αναπλ. Καθηγητής Ε.Μ.Π.

.....
Παναγιώτης Τσαραμπάρης
Επ. Καθηγητής Ε.Μ.Π.

ΑΘΗΝΑ, ΜΑΡΤΙΟΣ 2020

.....
Γεώργιος Δ. Μαχαίρας

Διπλωματούχος Ηλεκτρολόγος Μηχανικός και Μηχανικός Υπολογιστών Ε.Μ.Π.

Copyright © Γεώργιος Μαχαίρας, 2020.

Με επιφύλαξη παντός δικαιώματος. All rights reserved.

Απαγορεύεται η αντιγραφή, αποθήκευση και διανομή της παρούσας εργασίας, εξ' ολοκλήρου ή τμήματος αυτής, για εμπορικό σκοπό. Επιτρέπεται η ανατύπωση, αποθήκευση και διανομή για σκοπό μη κερδοσκοπικό, εκπαιδευτικής ή ερευνητικής φύσης, υπό την προϋπόθεση να αναφέρεται η πηγή προέλευσης και να διατηρείται το παρόν μήνυμα. Ερωτήματα που αφορούν τη χρήση της εργασίας για κερδοσκοπικό σκοπό πρέπει να απευθύνονται προς τον συγγραφέα.

Οι απόψεις και τα συμπεράσματα που περιέχονται σε αυτό το έγγραφο εκφράζουν τον συγγραφέα και δεν πρέπει να ερμηνευθεί ότι αντιπροσωπεύουν τις επίσημες θέσεις του Εθνικού Μετσόβιου Πολυτεχνείου.

Περίληψη

Η παρούσα εργασία αφορά στο σχεδιασμό και την υλοποίηση μιας εφαρμογής σε προγραμματιστικό περιβάλλον Android. Η εφαρμογή αυτή έχει σχεδιαστεί να εμφανίζει στην οθόνη του κινητού τηλεφώνου, κατηγορίας smartphone, μετρήσεις από έναν αισθητήρα μαγνητικού πεδίου, Hall. Η διάταξη της εργασίας περιλαμβάνει επίσης μία συσκευή Arduino Uno που συνδέεται με τον αισθητήρα και επεξεργάζεται τις μετρήσεις που λαμβάνει αυτός. Η μετάδοση αυτών των μετρήσεων στο κινητό γίνεται μέσω ενός Bluetooth (HC-05), το οποίο είναι επίσης συνδεδεμένο με το Arduino. Η συγκεκριμένη εφαρμογή μπορεί να επεκταθεί και για άλλους αισθητήρες και παρόμοιες εφαρμογές, χρησιμοποιώντας απλά τον κατάλληλο κώδικα προγραμματισμού της συσκευής Arduino για τον εκάστοτε αισθητήρα (π.χ. αισθητήρα πίεσης, θερμοκρασίας, υγρασίας κ.α.). Η πλατφόρμα η οποία χρησιμοποιήθηκε για τη συγγραφή, διόρθωση, μεταγλώττιση και εκτέλεση του κώδικα της εφαρμογής είναι το Android Studio. Η γλώσσα που επιλέχθηκε για τη συγγραφή είναι η Java. Οι λόγοι επιλογής αυτής της γλώσσας προγραμματισμού είναι ότι αυτή είναι η γλώσσα που χρησιμοποιείται πιο ευρέως στη συγγραφή Android εφαρμογών, η πληθώρα των βιβλιοθηκών που υπάρχουν και συντελούν στην πιο ευχάριστη εμπειρία για το χρήστη της και διότι είναι αντικειμενοστρεφής. Το Android παρέχει έτοιμες βιβλιοθήκες, εντολές καθώς και ολόκληρες συναρτήσεις για διάφορα πεδία προγραμματισμού, όπως η σύνδεση Bluetooth με την εφαρμογή καθώς και αποστολή μηνυμάτων. Χρήση αυτών έγινε για την περάτωση της παρούσης εργασίας.

Λέξεις κλειδιά: Arduino, Android, Bluetooth, Αισθητήρας Hall

ABSTRACT

This thesis is about designing and implementing an application in Android programming environment. This app is designed to display on the screen of a cellphone, smartphone's model, measurements by a Hall magnetic field sensor. The thesis' layout also includes an Arduino Uno device that connects to the sensor and processes the measurements it receives. These measurements are transmitted to the mobile via a Bluetooth (HC-05), which is also connected to the Arduino. This application can be extended to other sensors and similar applications by simply using the appropriate programming code of the Arduino device for each sensor (e.g. pressure, temperature, humidity sensor, etc.). The platform used for writing, editing, compiling, and executing the application code is the Android Studio. The language to be chosen for the code's writing is Java. The reasons for choosing this programming language is it's wide use in writing Android applications, the abundance of libraries that exist and contribute to its user-friendly experience and because it is object-oriented. Android provides ready libraries, commands as well as complete functions for various programming areas, such as connecting the Bluetooth to the application as well as sending messages. Some of them were used to complete this thesis.

Key-words: Arduino, Android, Bluetooth, Hall Sensor

Ευχαριστίες

Σε αυτό το σημείο θα ήθελα να ευχαριστήσω όλους όσους συνέβαλαν ουσιαστικά στην εκπόνηση της παρούσας εργασίας. Αρχικά, τον κ. Ε. Χριστοφόρου, ο οποίος είναι ο επιβλέπων καθηγητής μου, για την κατανόηση που επέδειξε και την καθοριστική συμβολή του στην αποπεράτωση της εργασίας. Ακόμα, θέλω να ευχαριστήσω την οικογένεια μου για την πολύτιμη στήριξη της καθ' όλη τη διάρκεια των σπουδών μου. Η αγάπη αυτών και της κοπέλας μου και η ανάγκη που έβλεπα στα μάτια τους να με δουν πτυχιούχο, όπως και τους κόπους μου και τους δικούς τους να ανταμείβονται, μου έδωσαν την ώθηση να τα καταφέρω. Τέλος, θέλω να ευχαριστήσω τους φίλους μου, άλλους για τη σημαντική ψυχολογική υποστήριξη τους και άλλους για την καίρια συμβολή τους κατά την περίοδο των σπουδών μου. ΕΥΧΑΡΙΣΤΩ από τα βάθη της καρδιάς μου!!!

Περιεχόμενα

ΚΕΦΑΛΑΙΟ 1. ΕΙΣΑΓΩΓΗ	1
1.1 Γενική Ανασκόπηση.....	1
1.2 Αισθητήρας	1
1.3 Σκοπός Εργασίας.....	2
ΚΕΦΑΛΑΙΟ 2. ΘΕΩΡΗΤΙΚΟ ΜΕΡΟΣ	3
2.1 Αισθητήρες.....	3
2.1.1 Γενικά.....	3
2.1.2 Φαινόμενο Hall	5
2.1.3 Αισθητήρας Hall	6
2.2 Bluetooth HC-05	9
2.2.1 Εισαγωγικά	9
2.2.2 Ομάδες πρωτόκολλων συσκευών Bluetooth	9
2.2.3 Πλεονεκτήματα Bluetooth	10
2.2.4 Ευπάθειες Bluetooth και Pairing	11
2.2.5 Πρωτόκολλο SSP.....	11
2.2.6 Bluetooth HC-05.....	12
2.3 Arduino.....	16
2.3.1 Εισαγωγικά	16
2.3.2 Arduino Uno	17
2.3.3 Σχεδιάγραμμα Arduino	18
2.3.4 Ακριβή Χαρακτηριστικά του Arduino Uno.....	21
2.3.5 Ανανεωμένες εκδόσεις του Arduino Uno.....	22
2.3.6 Επικοινωνία	22
2.3.7 Λογισμικό	23
2.3.7.1 Βιβλιοθήκες Arduino	24
2.3.7.2 Arduino sketch.....	25
2.3.7.3 Arduino Bootloader	25
2.3.7.4 Arduino Serial Monitor	26
2.3.7.5 Αυτόματη επαναφορά λογισμικού	27
2.3.8 Λειτουργίες Arduino.....	27
2.3.9 Επιλογή Arduino για την εργασία	28
2.3.10 Σύγκριση συσκευών Arduino	29
2.3.11 Προϊόντα Arduino.....	31
2.4 ANDROID	32
2.4.1 Γενικά.....	32
2.4.2 Λειτουργικό Σύστημα Android.....	32

2.4.3 Εκδόσεις Android	34
2.4.4 Γλώσσα Προγραμματισμού Android.....	36
2.4.5 Δομικά Χαρακτηριστικά Εφαρμογών Android	36
2.4.5.1 Δηλωτικό Αρχείο Android – Αρχείο XML.....	37
2.4.5.2 Συστατικά Στοιχεία - Components	38
2.4.5.3 Components και Λειτουργικότητα	38
2.4.5.4 Components και Εμπειρία Χρήστη.....	39
2.4.6 Android Studio.....	41
2.4.6.1 Δομή Έργου-Project.....	42
2.4.6.2 Επισκόπηση projects	44
2.4.6.3 Modules-Ενότητες	45
2.4.6.4 Αρχεία και Κατάλογοι των Android Projects.....	46
2.4.6.5 Ρυθμίσεις Δομής Project.....	48
2.4.6.6 Σύνταξη διάταξης (Layout Editor)	49
2.4.6.7 APK Analyzer.....	49
2.4.6.8 Android SDK Platform.....	50
2.4.7 Επισκόπηση Bluetooth.....	50
ΚΕΦΑΛΑΙΟ 3. ΜΕΘΟΔΟΛΟΓΙΑ.....	53
3.1 Εισαγωγικά	53
3.2 HC-05 και διασύνδεση με Arduino.....	53
3.3 Κώδικας Arduino	56
3.4 Κώδικας Android Εφαρμογής.....	60
3.4.1 Αρχεία .java	60
3.4.1.1 MainActivity.java	60
3.4.1.2 MonitoringScreen.java	69
3.4.1.3 ActivityHelper.java	78
3.4.1.4 PreferencesActivity.java	79
3.4.2 Αρχεία .xml.....	82
3.4.2.1 AndroidManifest.xml.....	82
3.4.2.2 activity_main.xml	83
3.4.2.3 activity_monitoring_screen.xml.....	84

ΚΕΦΑΛΑΙΟ 4. ΣΥΣΤΑΣΗ ΚΑΙ ΑΠΟΤΕΛΕΣΜΑΤΑ ΕΦΑΡΜΟΓΗΣ	87
ΚΕΦΑΛΑΙΟ 5. ΣΥΜΠΕΡΑΣΜΑΤΑ	93
4.1 Προβληματισμοί και Προοπτικές	93
4.2 Μελλοντικές Εργασίες	94
ΒΙΒΛΙΟΓΡΑΦΙΑ	95

Κατάλογος Σχημάτων

Σχήμα 1: Το φαινόμενο Hall για διαφορετικές κατευθύνσεις του ηλεκτρικού ρεύματος και του μαγνητικού πεδίου.	5
Σχήμα 2: Διάταξη μελέτης φαινομένου Hall	6
Σχήμα 3: (a) Όταν οι φορείς φορτίου είναι αρνητικά φορτισμένοι, (b) Όταν οι φορείς φορτίου είναι θετικά φορτισμένοι.....	6
Σχήμα 4: Ο αισθητήρας φαινομένου Hall σε λειτουργία διακόπτη μεταγωγής: (A) η μαγνητική ροή ενεργοποιεί τον αισθητήρα (ON), (B) η μαγνητική ροή εμποδίζεται από μεταλλική πλάκα (OFF) [2]	8
Σχήμα 5: Τα βασικά δομικά στοιχεία της πλατφόρμας Android [16]	33
Σχήμα 6: Σχεδιάγραμμα διασύνδεσης του Bluetooth HC-05 με το Arduino [20].....	54
Σχήμα 7: Σχεδιάγραμμα διασύνδεσης του Bluetooth HC-05 με το Arduino με χρήση κυκλώματος διαίρεσης τάσης [20].....	55

Κατάλογος Πινάκων

Πίνακας 1 : Περιγραφή Ακροδεκτών Bluetooth HC-05.....	13
Πίνακας 2: Τα χαρακτηριστικά του Arduino Uno	21
Πίνακας 3: Χαρακτηριστικά διαφόρων συσκευών Arduino [13].....	30
Πίνακας 4: Χαρακτηριστικά απαρχαιωμένων συσκευών Arduino [13].....	30

Κατάλογος Εικόνων

Εικόνα 1: Η μονάδα HC-05 με το Breakout board zs-040	13
Εικόνα 2: Τα αρχεία του project στην προβολή Android.	44
Εικόνα 3: Αρχεία και κατάλογοι στη δομή ενός Android Project.	48
Εικόνα 4: Σφάλμα Μεταφόρτωσης Κώδικα	57
Εικόνα 5: Serial Monitor του Arduino sketch με τις μετρήσεις του αισθητήρα Hall..	59
Εικόνα 6: Άδεια χρήσης των bluetooth της Android συσκευής	87
Εικόνα 7α: Διεπαφή activity_main.xml Εικόνα 7β: Έξοδος MainActivity.java σε συνάρτηση με activity_main.xml.....	88
Εικόνα 8: Μετρήσεις και Αποτελέσματα	89
Εικόνα 9: Φωτογραφίες ρεαλιστικής διάταξης και αποτελεσμάτων	91

Κεφάλαιο 1. Εισαγωγή

1.1 Γενική Ανασκόπηση

Στη σύγχρονη τεχνολογική πραγματικότητα οι εξελίξεις είναι αδιαμφισβήτητα ραγδαίες. Τα τεχνολογικά επιτεύγματα που έχουν αρχίσει να ενσωματώνονται και ίσως μερικές φορές να αποτελούν αναπόσπαστο κομμάτι της καθημερινής ζωής των ανθρώπων, έχουν φτάσει σε σημείο να γίνονται απαραίτητα στον άνθρωπο σε σημείο που η απώλεια τους μπορεί να προκαλέσει πλήρη σύγχυση. Τα επιτεύγματα αυτά μπορεί να σχετίζονται με τον τρόπο λειτουργίας μεγάλων εργοστασίων και επιχειρήσεων μέχρι και αυτόν των νοικοκυριών. Είναι πρόδηλο, μάλιστα, ότι έχουν βελτιώσει σε μεγάλο βαθμό το βιοτικό επίπεδο και έχουν διευκολύνει την καθημερινή ρουτίνα. Ένας κλάδος που έχει συμβάλλει καταλυτικά σε αυτή την κατεύθυνση είναι αυτός της ηλεκτρονικής και των αισθητήρων. Σχεδόν οτιδήποτε ηλεκτρονικό χρησιμοποιούμε σήμερα περιέχει αισθητήρες.

1.2 Αισθητήρας

Αισθητήρας ονομάζεται μία συσκευή που ανιχνεύει ένα φυσικό μέγεθος και παράγει από αυτό μία μετρήσιμη έξοδο. Οι αισθητήρες χρησιμοποιούνται σε καθημερινά αντικείμενα, όπως κουμπιά ανελκυστήρων με αισθητήρες αφής και λάμπες φωτισμού με αισθητήρες κίνησης. Υπάρχουν αναρίθμητες ακόμη χρήσεις που οι περισσότεροι άνθρωποι δεν αντιλαμβάνονται. Εφαρμογές τους συναντούμε στα αυτοκίνητα, σε μηχανές, στην αεροναυπηγική, την ιατρική, τη βιομηχανία και τη ρομποτική. Αισθητήρες οπτικούς, αφής, κίνησης, μεταβολών του περιβάλλοντος και πολλούς άλλους.

1.3 Σκοπός Εργασίας

Στην παρούσα εργασία θα μελετηθεί η ανάγκη που προέκυψε να απεικονίζονται τα δεδομένα από έναν αισθητήρα μέτρησης μαγνητικού πεδίου, κυρίως για επαγγελματική και ερευνητική χρήση. Ο αισθητήρας αυτός λαμβάνει εξωτερική τάση τροφοδοσίας, και ανιχνεύει ένταση μαγνητικού πεδίου η οποία έχει παράξει τάση Hall. Ο αισθητήρας καταγράφει μεταβολές στο μαγνητικό του περιβάλλον τοποθετούμενος κοντά σε μαγνητικά υλικά ή παρεμβάλλοντας κάποιο μέσο που εμποδίζει μερικώς ή καταλυτικά την επαφή του με το μαγνητικό πεδίο, προκαλώντας τον περιορισμό της τάσης Hall, μέχρι το μηδενισμό της.

Για να επιτευχθεί η λύση και η κάλυψη αυτής της ανάγκης δημιουργήθηκε μια εφαρμογή για κινητά τηλέφωνα σε περιβάλλον Android με τη συνδρομή της πλακέτας Arduino και τη χρήση Bluetooth για τη μετάδοση των δεδομένων.

Κεφάλαιο 2. Θεωρητικό Μέρος

2.1 Αισθητήρες

2.1.1 Γενικά

Αισθητήρας ονομάζεται μία συσκευή που ανιχνεύει ένα φυσικό μέγεθος και παράγει από αυτό μία μετρήσιμη έξοδο. Ένας αισθητήρας μαγνητικού πεδίου όπως ο αισθητήρας Hall, που χρησιμοποιήθηκε στην παρούσα εργασία, μετατρέπει το μαγνητικό σήμα σε χρήσιμο ηλεκτρικό σήμα, τάση ή ρεύμα ανάλογα το είδος του αισθητήρα. Η χρήση των αισθητήρων στην αυτοματοποίηση διάφορων συστημάτων συντέλεσε σε πληθώρα βελτιώσεων σε πολλούς τομείς της βιομηχανίας αλλά και της καθημερινότητας.

- Αύξηση της παραγωγικότητας
- Χαμηλό κόστος παραγωγής
- Συνεχή λειτουργία χωρίς σφάλματα
- Ασφάλεια του προσωπικού και των μηχανημάτων
- Απλοποίηση των τεχνικών
- Μοντελοποίηση της παραγωγής
- Προτυποποίηση συνθηκών

Οι αισθητήρες γενικά χρησιμοποιούνται για συλλογή πληροφοριών και για έλεγχο συστημάτων. Για την επιλογή του ιδανικού αισθητήρα σε κάθε περίπτωση και για τη γνώση σχετικά με τη λειτουργία του υπάρχουν κάποια διακριτά χαρακτηριστικά:

- Εύρος αξιόπιστων ορίων λειτουργίας.
- Ακρίβεια της τιμής εξόδου προς τη τιμή εισόδου.
- Βαθμονόμηση σε μονάδες.

-Γραμμικότητα: Ο βαθμός στον οποίο η γραφική παράσταση της εξόδου προσεγγίζει ευθεία ως προς την είσοδο του αισθητήρα.

-Σφάλμα μετρούμενης σε σχέση με την πραγματική τιμή.

-Στατικό σφάλμα: Σταθερό σφάλμα σε όλο το εύρος λειτουργίας, το οποίο μπορεί να αντισταθμιστεί.

-Ανοχή: Το μέγιστο σφάλμα που μπορεί να δημιουργήσει ο αισθητήρας.

-Διακριτική ικανότητα: Η μικρότερη αλλαγή τιμής εισόδου που μπορεί να ανιχνεύσει.

-Ευαισθησία: Η ικανότητα του αισθητήρα να ανιχνεύει μικρές διακυμάνσεις μετρούμενου μεγέθους

-Νεκρή ζώνη: Το μέγιστο ποσό αλλαγής της εισόδου που δεν επιφέρει αλλαγή στην έξοδο.

-Απόκριση: Ο χρόνος που απαιτείται για να λάβει τη τελική τιμή η έξοδος.

-Ευστάθεια: Η μεταβολή της εξόδου σε μεγάλη χρονική περίοδο, χωρίς μεταβολή της εισόδου και των συνθηκών.

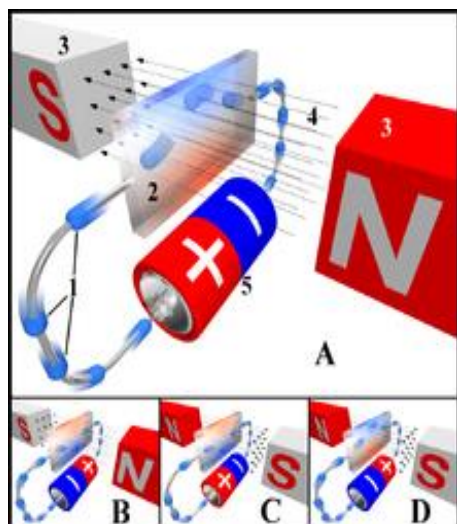
-Υστέρηση: Η διαφορά στην έξοδο όταν η κατεύθυνση της μεταβολής της εισόδου αντιστραφεί.

-Επαναληψιμότητα: Η παραγωγή του ίδιου αποτελέσματος, σε διαφορετικές χρονικές στιγμές, με την ίδια είσοδο.

-Ολίσθηση: Η μεταβολή των χαρακτηριστικών του αισθητήρα με το χρόνο και το περιβάλλον [1].

2.1.2 Φαινόμενο Hall

Το 1879 ο Edwin Hall ανακάλυψε ότι όταν ένα αγώγιμο έλασμα που διαρρέεται από ρεύμα εισαχθεί σε μαγνητικό πεδίο, τότε αναπτύσσεται διαφορά δυναμικού σε διεύθυνση που είναι κάθετη στο ρεύμα και στο μαγνητικό πεδίο. Αυτή η διαφορά δυναμικού καλείται τάση Hall (V_H). Το φαινόμενο αυτό, γνωστό ως φαινόμενο Hall,

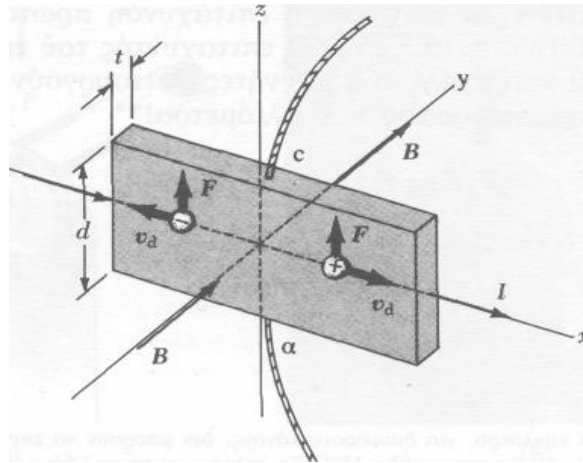


Σχήμα 1: Το φαινόμενο Hall για διαφορετικές κατευθύνσεις του ηλεκτρικού ρεύματος και του μαγνητικού πεδίου.

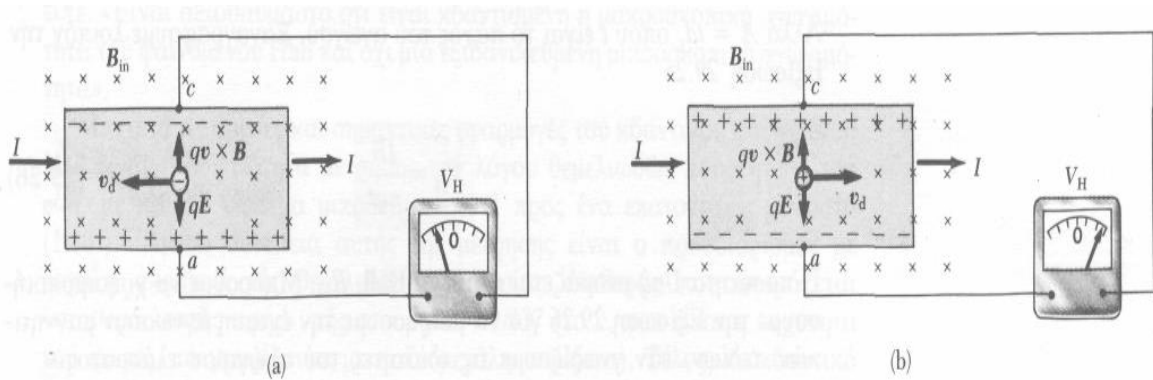
το προξενεί η μονομερής απόκλιση των φορέων φορτίου, λόγω της αλληλεπίδρασης με το μαγνητικό πεδίο.

Το φαινόμενο Hall χρησιμοποιείται στη μέτρηση μαγνητικών πεδίων και εμφανίζεται πιο έντονα στους ημιαγωγούς. Η παρατήρηση του φαινομένου πραγματοποιείται με την εφαρμογή ενός μαγνητικού πεδίου σε έναν αγώγο ο οποίος διαρρέεται από ρεύμα. Όταν το ρεύμα έχει την κατεύθυνση του άξονα των x και το μαγνητικό πεδίο αυτή του άξονα των y , όπως στο Σχήμα 2, τότε οι φορείς θετικού φορτίου καθώς και οι

φορείς αρνητικού φορτίου εκτρέπονται από το μαγνητικό πεδίο προς τα επάνω. Ανάμεσα στα σημεία a και c μετρούμε την τάση Hall. Όταν οι φορείς φορτίου είναι αρνητικά φορτισμένοι, η επάνω πλευρά αποκτά περίσσειμα αρνητικού φορτίου και το σημείο c έχει χαμηλότερο δυναμικό από το a . Όταν οι φορείς φορτίου είναι θετικά φορτισμένοι, η επάνω πλευρά αποκτά περίσσειμα θετικού φορτίου και το σημείο c έχει υψηλότερο δυναμικό από το a . Και στις δύο περιπτώσεις παύουν να εκτρέπονται οι φορείς φορτίου όταν οι πλευρές φορτιστούν πλήρως, δηλαδή όταν επέλθει ισορροπία ανάμεσα στην ηλεκτροστατική δύναμη και στη μαγνητική δύναμη [2].



Σχήμα 2: Διάταξη μελέτης φαινομένου Hall



Σχήμα 3: (a) Όταν οι φορείς φορτίου είναι αρνητικά φορτισμένοι, (b) Όταν οι φορείς φορτίου είναι θετικά φορτισμένοι

Σήμερα εφαρμογές που αξιοποιούν το φαινόμενο Hall χρησιμοποιούνται σε ένα μεγάλο εύρος συσκευών, όπως υπολογιστές, ραπτομηχανές, αυτοκίνητα, αεροσκάφη, ακόμα και ιατρικά μηχανήματα [3].

2.1.3 Αισθητήρας Hall

Ένας καίριος κλάδος χρήσης των αισθητήρων όπως προαναφέρθηκε είναι ο βιομηχανικός. Η διεξαγωγή μη καταστροφικών δοκιμών που συνιστά βασικό πεδίο εφαρμογών κάνει χρήση αισθητήρων ανίχνευσης μικρής έντασης πεδίου, με περισσότερο χρησιμοποιούμενους τους αισθητήρες που βασίζονται στο φαινόμενο Hall. Οι αισθητήρες Hall έχουν ευαισθησία της τάξης του 0.1mT.

Ο αισθητήρας Hall μετράει τη μαγνητική ένταση \vec{H} .

Θεωρητικά ορίζεται από τον τύπο: $\vec{B} = \mu_0 \vec{H} + \vec{M}$ (1)

όπου \vec{B} είναι η μαγνητική πυκνότητα ροής, \vec{M} η παραμένουσα μαγνήτιση και μ_0 η μαγνητική διαπερατότητα του κενού.

Ο αισθητήρας Hall έχει τρεις ακροδέκτες:

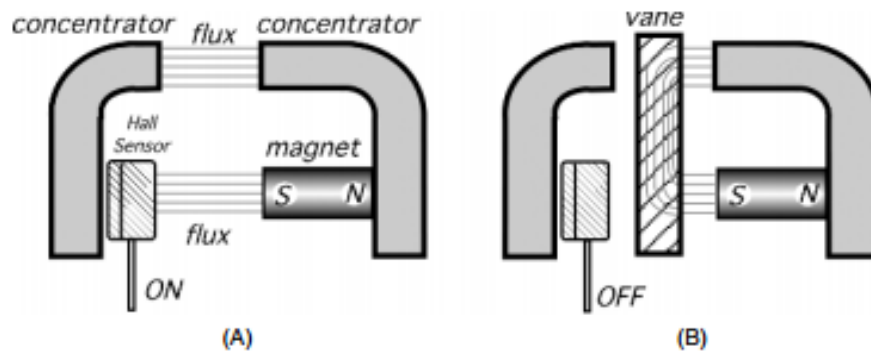
α. για τη γείωση,

β. για την τάση αναφοράς και

γ. για το σήμα εξόδου

Ο αισθητήρας τροφοδοτούμενος με μια τάση αναφοράς της τάξης των 5 έως 12V ανάλογα με την εφαρμογή, παράγει αντίστοιχο σήμα εξόδου το οποίο είναι σχεδόν τετραγωνικό και δεν απαιτεί μετατροπή σε ψηφιακό για να διαβαστεί από τον υπολογιστή. Αυτό συμβαίνει λόγω του σταθερού σήματος τάσης που παράγουν οι αισθητήρες Hall και μεταβάλλεται απότομα από τη μέγιστη στη μηδενική τάση και αντίστροφα.

Ένας αισθητήρας Hall χρησιμοποιώντας ένα κινούμενο μέσο λειτουργεί ως διακόπτης μεταγωγής. Ειδικότερα, όταν στο χώρο μεταξύ του αισθητήρα Hall και του μαγνήτη δεν παρεμβάλλεται μέσο-έλασμα, ο προκύπτων διακόπτης θεωρείται ανοικτός-ON (Σχήμα 4 (A)), με αποτέλεσμα να παράγει μια σταθερή τάση εξόδου. Κατά συνέπεια της προηγούμενης υπόθεσης, όταν μια φερρομαγνητική πλάκα (μεταλλικό έλασμα) εισάγεται στο χώρο μεταξύ αισθητήρα και μαγνήτη προκαλεί τη μείωση της τάσης εξόδου, που τείνει στο μηδέν καθώς το έλασμα εμποδίζει πλήρως τον αισθητήρα να εντοπίσει το μαγνητικό πεδίο που προκαλεί ο μαγνήτης (Σχήμα 4 (B)). Τότε, θεωρούμε ότι ο αισθητήρας είναι «κλειστός» (OFF). Αξιοποιώντας την λειτουργία διακόπτη, ο αισθητήρας Hall χρησιμοποιείται ευρέως για την ανίχνευση μαγνητικού πεδίου, καθώς και για την ανίχνευση της θέσης ενός αντικειμένου [4].



Σχήμα 4: Ο αισθητήρας φαινομένου Hall σε λειτουργία διακόπτη μεταγωγής: (A) η μαγνητική ροή ενεργοποιεί τον αισθητήρα (ON), (B) η μαγνητική ροή εμποδίζεται από μεταλλική πλάκα (OFF) [2]

2.2 Bluetooth HC-05

2.2.1 Εισαγωγικά

Υπάρχουν διάφοροι τρόποι ασύρματης επικοινωνίας, όπως RF και nRF, ZigBee, Wi-Fi και Bluetooth. Η ασύρματη και συνάμα αμφίδρομη σύνδεση και επικοινωνία συσκευών δικτύου σε σχετικά μικρές αποστάσεις υλοποιείται με τη χρήση μονάδων Bluetooth [5]. Η τεχνολογία των συσκευών Bluetooth επινοήθηκε το 1994 από τον L.M. Ericsson. Το πρωτόκολλο Bluetooth αποτελεί μια προσιτή μέθοδος επικοινωνίας σε δίκτυο WPAN*, με πλήρως πιστοποιημένη διαμόρφωση Bluetooth V2.0 + EDR (Enhanced Data Rate), με μέγιστο ρυθμό δεδομένων στα 1-3Mbps (ενώ μπορεί να αγγίξει και τα 25Mbps), που λειτουργεί σε ονομαστική εμβέλεια μικρότερη των 100 μέτρων, το φυσικό στρώμα του οποίου λειτουργεί στη Βιομηχανική, Επιστημονική και Ιατρική (Industrial, Scientific and Medical-ISM) ζώνη συχνοτήτων των 2,4GHz είναι ένας κοινός τρόπος ασύρματης επικοινωνίας αποδεκτός παγκοσμίως, λόγω της ISM ζώνης συχνοτήτων[6]. Η φιλικότητα προς τον χρήστη και το χαμηλό κόστος της επικοινωνίας της τεχνολογία των Bluetooth είναι αυτά που τα ανέδειξαν ευρέως. Η μονάδα HC-05, που χρησιμοποιήθηκε στην εργασία, είναι μια μονάδα Bluetooth που χρησιμοποιεί σειριακή επικοινωνία, και βρίσκει εφαρμογή κυρίως σε ηλεκτρονικές κατασκευές [7].

*Ασύρματα Δίκτυα Προσωπικού Χώρου(WPAN-Personal Area Network): Τα δίκτυα προσωπικού χώρου είναι δίκτυα υπολογιστών τα οποία χρησιμοποιούνται για μετάδοση δεδομένων, σύνηθες παράδειγμα είναι η σύνδεση ενός κινητού τηλεφώνου σε έναν υπολογιστή [8].

2.2.2 Ομάδες πρωτόκολλων συσκευών Bluetooth

Τρεις ομάδες πρωτοκόλλων διακρίνουν την αρχιτεκτονική των Bluetooth:

- α. πρωτοκόλλων μεταφοράς,
- β. πρωτόκολλων middleware(μεσαίας κατηγορίας) και
- γ. εφαρμογών

Η ομάδα πρωτόκολλου μεταφοράς αποτελείται από τα ακόλουθα στρώματα: ραδιοσυχνότητες, βασική ζώνη, διαχειριστή συνδέσεων και πρωτόκολλο ελέγχου και προσαρμογής λογικής σύνδεσης (L2CAP-Logical Link Control and Adaptation Protocol). Μέσω αυτών των στρωμάτων καθίσταται εφικτός ο εντοπισμός των συσκευών Bluetooth μεταξύ τους κατά τη διαχείριση των φυσικών και λογικών συνδέσεων.

Η ομάδα πρωτόκολλων middleware (το λογισμικό που συνδέει αιτήματα δικτύου στα back-end δεδομένα που αιτείται ένας πελάτης-client) καθιστά εφικτή τη λειτουργία σε υπάρχουσες και νέες εφαρμογές μέσω συνδέσεων Bluetooth. Ορισμένα γνωστά πρωτόκολλα middleware είναι το πρωτόκολλο Internet (IP), το πρωτόκολλο ελέγχου μετάδοσης (TCP) και ο εξομοιωτής σειριακής θύρας (RFCOMM).

Οι εφαρμογές που χρησιμοποιούν τις συνδέσεις Bluetooth απαρτίζουν την ομάδα εφαρμογών.

2.2.3 Πλεονεκτήματα Bluetooth

Στη συνέχεια, παρατίθενται συνοπτικά τα πλεονεκτήματα της τεχνολογίας των Bluetooth:

- α. ευρέως διαθέσιμη,
- β. δωρεάν,
- γ. αντικαθιστά τη χρήση καλωδίων,
- δ. υποστηρίζει τόσο φωνή όσο και δεδομένα και
- ε. λειτουργεί στην ISM σταθερή ζώνη συχνοτήτων διαθέσιμη παγκοσμίως

2.2.4 Ευπάθειες Bluetooth και Pairing

Στην τεχνολογία των Bluetooth, όπως και σε άλλες τεχνολογίες που διαχειρίζονται μετάδοση δεδομένων, εντοπίζονται ευπάθειες που αφορούν στην υποκλοπή δεδομένων κατά τη διάρκεια της σύζευξης των συσκευών. Η υποκλοπή αυτή έχει λάβει χώρα είτε λόγω έλλειψης κρυπτογράφησης στο επίπεδο εφαρμογής ή δυνατότητας του υποκλοπέα να μιμηθεί μια συσκευή. Η υποκλοπή επιτυγχάνεται μέσω παρεμβατικής αναμετάδοσης. Ο υποκλοπέας C επικοινωνεί με τη μονάδα A, φερόμενος ως μονάδα B και στη μονάδα B φερόμενος ως μονάδα A.

Για τον παραπάνω λόγο, η ενίσχυση της ασφάλειας χρήσης των Bluetooth κρίθηκε αναγκαία. Η διασφάλιση της χρήσης των μονάδων Bluetooth έλαβε χώρα μέσω μιας διαδικασίας που καλείται αντιστοίχιση (pairing). Η διαδικασία αντιστοίχισης αποσκοπεί στη δημιουργία ενός προσαρμοσμένου κλειδιού σύνδεσης το οποίο ουσιαστικά συνάπτει μια αξιόπιστη σχέση μεταξύ δύο συσκευών κάνοντας χρήση μυστικών κωδικών, γνωστών με τον όρο ακίδες (pins). Η ταυτοποίηση των συσκευών σύνδεσης επιτυγχάνεται, εν τέλει, μέσω της αλληλεπίδρασης των χρηστών που επιβεβαιώνουν τη γνησιότητα της συσκευής τους και επιτρέπουν τη μετάδοση δεδομένων.

2.2.5 Πρωτόκολλο SSP

Το πρωτόκολλο Διασφάλισης Απλούς Αντιστοίχισης (Secure Simple Pairing-SSP) είναι το πιο ευρέως χρησιμοποιούμενο πρωτόκολλο αντιστοίχισης. Το πρωτόκολλο αυτό καθορίζει τις απαραίτητες προϋποθέσεις δημιουργίας ασφαλούς συνδέσμου επικοινωνίας μεταξύ δύο συσκευών Bluetooth. Οι παρακάτω έξι φάσεις συνθέτουν το πρωτόκολλο SSP:

Φάση 1: Συσκευές που κάνουν pairing για πρώτη φορά ή επανασυνδέονται και έχουν πρόσβαση σε οθόνες και πληκτρολόγια, χρησιμοποιούν το μοντέλο αριθμητικής σύγκρισης (Numeric Comparison-NC).

Φάση 2: Οι συσκευές παράγουν τα ζεύγη δημόσιου-ιδιωτικού κλειδιού και το κλειδί ανταλλαγής Diffie-Hellman (χρήση μίξης χρωμάτων) πριν ανταλλάξουν τα pins.

Φάση 3: Στην περίπτωση του μοντέλου NC, εμφανίζεται ένας 6ψήφιος αριθμός και στις δύο συσκευές που επιχειρούν να πραγματοποιήσουν μια σύνδεση. Οι χρήστες είναι υπεύθυνοι για τη σύγκριση και την επιβεβαίωση των αριθμών πριν ολοκληρώσουν τη διαδικασία αντιστοίχισης.

Φάση 4: Οι συσκευές ανταλλάσσουν τις τιμές, συγκρίνουν και επαληθεύουν την ακεραιότητά τους.

Φάση 5: Οι συσκευές υπολογίζουν το κλειδί σύνδεσης χρησιμοποιώντας τις δικές τους διευθύνσεις Bluetooth, τις προηγούμενες τιμές δημιουργίας και το κλειδί Diffie-Hellman που κατασκευάστηκε κατά τη διάρκεια της φάσης 2.

Φάση 6: Πιστοποίηση πρωτόκολλου διαχειριστή διασύνδεσης-LMP (Link Manager Protocol) και κρυπτογράφηση: Δημιουργία των κλειδιών κρυπτογράφησης.

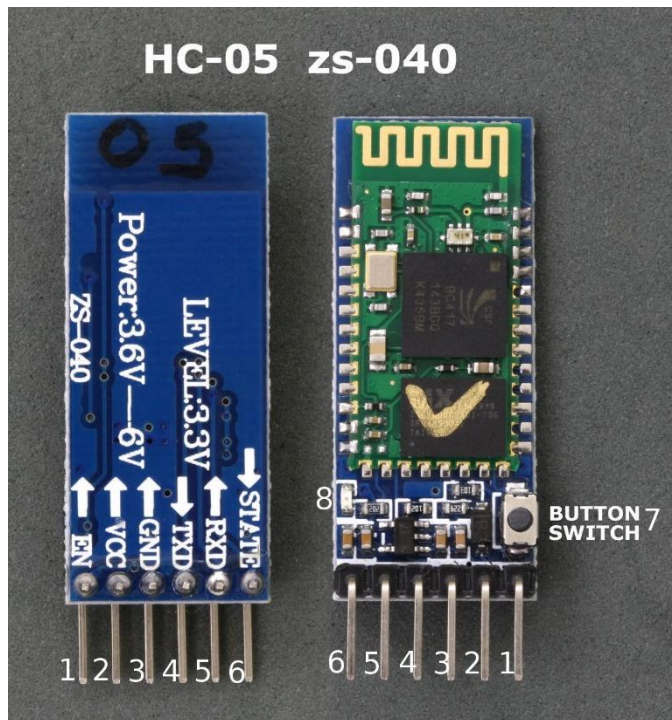
Μόλις οι συσκευές ολοκληρώσουν με επιτυχία τη διαδικασία αντιστοίχισης, δημιουργείται σύνδεσμος μεταξύ των συσκευών και η επικοινωνία μπορεί να συνεχιστεί [9].

2.2.6 Bluetooth HC-05

Για την εκπόνηση της παρούσας εργασίας χρησιμοποιήθηκε η μονάδα Bluetooth HC-05. συγκεκριμένη μονάδα Bluetooth ελέγχεται από το πρωτόκολλο σειριακής θύρας (Serial Port Protocol- SPP) και αποτελεί μία εύχρηστη μονάδα ασύρματης σειριακής σύνδεσης, αφού η επικοινωνία πραγματοποιείται σειριακά με τη βοήθεια εσωτερικής κεραίας και ως εκ τούτου, η διασύνδεση με υπολογιστή ή άλλο ελεγκτικό μέσω είναι εύκολη.

Η τάση εισόδου για την συγκεκριμένη μονάδα ορίζεται στα 3.6-5 Volts σταθερής τάσης (DC 5V).

Ορισμός ακροδεκτών:



Εικόνα 1: Η μονάδα HC-05 με το Breakout board zs-040

Πίνακας 1 : Περιγραφή Ακροδεκτών Bluetooth HC-05

ΑΚΡΟΔΕΚΤΗΣ	ΠΕΡΙΓΡΑΦΗ	ΛΕΙΤΟΥΡΓΙΑ
KEY-ENABLED (1)	Είσοδος διακόπτη λειτουργίας	Εάν εισάγεται χαμηλή στάθμη ή συνδεθεί με τον αέρα, η μονάδα βρίσκεται σε κατάσταση ζεύξης ή επικοινωνίας ή σε λειτουργία δεδομένων. Εάν εισέρχεται σε υψηλό επίπεδο, η μονάδα θα τεθεί σε λειτουργία AT (που επιτρέπει την αλλαγή ρυθμίσεων του Bluetooth, ή την ανίχνευση της

		λειτουργίας του ως master ή slave)**.
VCC (2)	+5V	Σύνδεση στα +5V
GND (3)	Γείωση(Ground)	Σύνδεση στη Γείωση
TXD (4)	UART_TXD, Ακροδέκτης αποστολής σειριακού σήματος Bluetooth	Σύνδεση με τον ακροδέκτη RXD του μικροελεγκτή (MCU-Microcontroller) κ.α.--- του Arduino UNO στην περίπτωση μας
RXD (5)	UART_RXD, Ακροδέκτης λήψης σειριακού σήματος Bluetooth	Σύνδεση με τον ακροδέκτη TXD του μικροελεγκτή (MCU-Microcontroller) κ.α.--- του Arduino UNO στην περίπτωση μας
STATE (6)	Συνδεδεμένο με το LED της μονάδας Bluetooth	Όταν το LED ανάβει δίνει έξοδο HIGH αλλιώς δίνει έξοδο LOW.

Επίσης διαθέτει μια ένδειξη LED (8) η οποία ανάλογα με τον ρυθμό που αναβοσβήνει δίνει πληροφορία για την κατάσταση της μονάδας, καθώς και ένα κουμπί (7) το οποίο επιτρέπει την αλλαγή κατάστασης της μονάδας.

Για την ένδειξη στο LED:

α. Αναβοσβήνει μία φορά σε 2 δευτερόλεπτα: Η μονάδα έχει μπει σε λειτουργία εντολών

β. Επαναλαμβανόμενο αναβοσβήσιμο: Αναμονή σύνδεσης στη λειτουργία δεδομένων,

γ. Αναβοσβήνει δύο φορές σε 1 δευτερόλεπτο: Η σύνδεση είναι επιτυχής στη λειτουργία δεδομένων.

** Η μονάδα Bluetooth HC-05 διέπεται από την αρχή της εναλλαγής κύριας-υποτελούς λειτουργίας, τη γνωστή, στη διεθνή επιστημονική κοινότητα, ως «master-slave switching mode», με αποτέλεσμα να μην δύναται να κάνει χρήση των δεδομένων που λαμβάνει ή μεταδίδει. Επομένως, λειτουργεί καθαρά σαν δίαυλος και απαιτεί άλλες συσκευές που μπορούν να απεικονίσουν τα δεδομένα που μεταδίδει, όπως μια συσκευή Arduino ή μια Android εφαρμογή, όπως επιτελείται και στην εργασία αυτή [10].

2.3 Arduino

2.3.1 Εισαγωγικά

Το υλικό (hardware) ανοιχτού κώδικα αποτελεί σχετικά καινούργια προσθήκη στον χώρο της ηλεκτρονικής και του προγραμματισμού, σε αντίθεση με το λογισμικό (software) ανοιχτού κώδικα που τυγχάνει ευρείας αξιοποίησης στις τάξεις της τεχνολογίας υπολογιστικών συστημάτων, εδώ και δεκαετίες. Μία απ' της δημοφιλέστερες και πιο αξιοποιήσιμες μονάδες hardware ανοιχτού κώδικα είναι το Arduino. Μία μονάδα που σχεδιάστηκε και υλοποιήθηκε το 2005 στην Ιταλία στο Ινστιτούτο Σχεδίασης Αλληλεπίδρασης Inrea ως ένα σύστημα που επέτρεψε στους μαθητές να αναπτύξουν διαδραστικά σχέδια.

Το Arduino προσομοιάζει συσκευή λήψης δεδομένων που δέχεται αναλογικές / ψηφιακές εισόδους από εξωτερικά μέσα, όπως κουμπιά, αισθητήρες κ.α., και παρέχει σήμα PWM (Pulse-Width Modulation) για την οδήγηση αναλογικής συσκευής μέσω διαμορφωμένου ψηφιακού σήματος εξόδου. Με τη διαμόρφωση αυτή μειώνεται η μέση ισχύς που παρέχεται από το ηλεκτρικό σήμα, με την μετατροπή του σε σήμα διακριτού χρόνου. Η μέση τιμή της τάσης που τροφοδοτείται στο φορτίο ελέγχεται με την ταχύρρυθμη ενεργοποίηση και απενεργοποίηση του διακόπτη μεταξύ τροφοδοσίας και φορτίου.

Ως υλικό ανοιχτού κώδικα το Arduino κυκλοφορεί με ελεύθερη άδεια υπό GPL (Gnu General Public License) και κάθε λεπτομέρεια της αρχιτεκτονικής του είναι διαθέσιμη. Αρκετές εμπορικές μονάδες Arduino είναι διαθέσιμες σε σχετικά χαμηλό κόστος και χρησιμοποιούν κεντρικό επεξεργαστή της σειράς Atmel megaAVR. Σημειωτέον, η ιδιότητα του Arduino να αποτελεί υλικό ανοιχτού κώδικα συντελεί στην ύπαρξη της δυνατότητας στον οποιοδήποτε το επιθυμεί να μπορεί να δημιουργήσει τη δικιά του έκδοση Arduino. Έτσι, σήμερα υπάρχει μια μεγάλη ποικιλία μονάδων Arduino με διαφορετικούς επεξεργαστές, μεγέθη και χαρακτηριστικά συνδεσιμότητας [11].

2.3.2 Arduino Uno

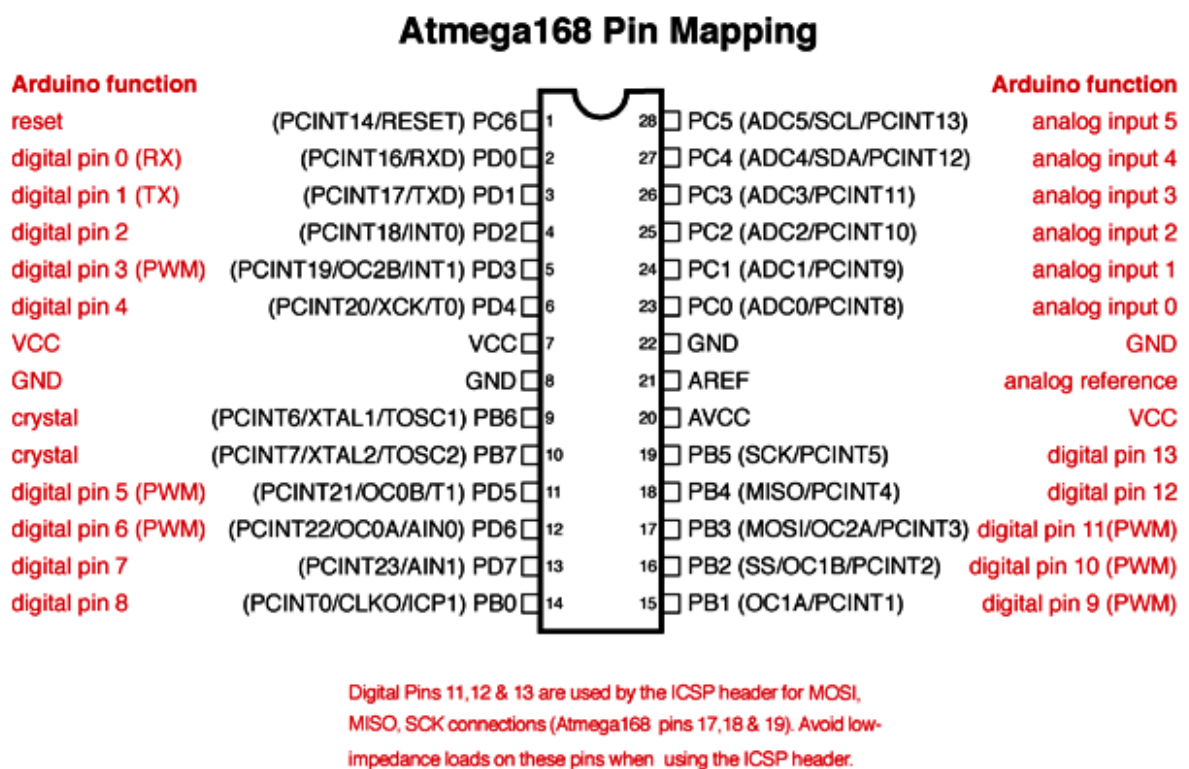
Στην παρούσα εργασία χρησιμοποιήθηκε ο μικροελεγκτής ανοιχτού κώδικα Arduino Uno. Βασικό στοιχείο που συνθέτει την πλακέτα του Arduino Uno είναι ένας single-chip (μονής πλακέτας) μικροελεγκτής 8-bit επεξεργαστή RISC 16MHz κρυστάλλου χαλαζία (κρυσταλλικό ταλαντωτή), ο ATmega328. Έχει 14 ψηφιακούς ακροδέκτες εισόδου / εξόδου, εκ των οποίων 6 μπορούν να χρησιμοποιηθούν ως έξοδοι PWM και 6 ως αναλογικές εισοδοί. Αυτός ο μικροελεγκτής προσφέρει υπολογιστική ισχύ περίπου 300 χιλιάδων γραμμών κώδικα ανά δευτερόλεπτο. Η μονάδα του Arduino συνδέεται με τον υπολογιστή μέσω καλωδίου USB (Universal Serial Bus) ώστε να ξεκινήσει να λειτουργεί, αυτή είναι η μία πηγή τροφοδοσίας του. Επιπλέον, διαθέτει υποδοχή για εξωτερική τροφοδοσία, όπως μπαταρία ή προσαρμογέα AC/DC. Η πηγή ενέργειας επιλέγεται αυτόματα. Ο προσαρμογέας μπορεί να συνδεθεί με το Arduino συνδέοντας ένα κεντρικό βύσμα 2,1 mm στην υποδοχή τροφοδοσίας της συσκευής. Οι οδηγήσεις από μια μπαταρία μπορούν να εισαχθούν από τα pins GND και Vin της υποδοχής POWER. Στην πλακέτα υπάρχουν ακόμα μια κεφαλίδα ICSP-In Circuit Serial Programming (μέθοδος προγραμματισμού μικροελεγκτών) και κουμπί επαναφοράς-reset.

Το «Uno» σημαίνει ένα (1) στην ιταλική γλώσσα και επιλέχθηκε για να σηματοδοτήσει την κυκλοφορία του λογισμικού Arduino (IDE-Integrated Development Environment-Ολοκληρωμένο Περιβάλλον Ανάπτυξης) 1.0. Η μονάδα του Arduino Uno και η έκδοση 1.0 του λογισμικού Arduino (IDE) ήταν οι εκδόσεις αναφοράς του Arduino, οι οποίες έχουν εξελιχθεί και έχουν προσαρμοστεί στις ανάγκες των χρηστών, αφού αυτοί είναι άλλωστε που έχουν το δικαίωμα να τα ανανεώσουν και να τα προσαρμόσουν στις περιπτώσεις που θέλουν να εξετάσουν και τους εξυπηρετούν, βάσει της αρχής του ανοιχτού κώδικα, όπως προαναφέρθηκε. Η μονάδα Uno είναι το εφαλτήριο μιας σειράς USB Arduino που δημιουργήθηκαν στη συνέχεια και κατ' επέκταση, αποτελεί το μοντέλο αναφοράς για την πλατφόρμα του Arduino. Ένα πλεονέκτημα του Arduino Uno για αρχάριους, αλλά και προχωρημένους χρήστες, καθώς το να υποπέσεις σε λάθος δεν είναι προνόμιο μόνο των αρχάριων, είναι ότι η συγκεκριμένη μονάδα είναι αρκετά οικονομική, εύκολη στη χρήση, ενώ

διαθέτει μια πολυλειτουργική επαναφορά που προστατεύει τις θύρες USB του υπολογιστή, με τον οποίο είναι συνδεδεμένη η μονάδα, από βραχυκύκλωμα και υπερένταση. Αν και οι περισσότεροι υπολογιστές παρέχουν την δική τους εσωτερική προστασία, η ασφάλεια παρέχει επιπλέον προστασία. Εάν εφαρμοστεί ένταση ρεύματος μεγαλύτερη των 500 mA στη θύρα USB, η ασφάλεια θα διακόψει αυτόματα τη σύνδεση μέχρι να αποκατασταθεί το βραχυκύκλωμα ή η υπερφόρτωση.

Το Uno διαφέρει από όλες τις προηγούμενες πλακέτες στο ότι δεν χρησιμοποιεί το chip driver FTDI USB-to-serial. Αντίθετα, διαθέτει το Atmega16U2 (Atmega8U2 μέχρι την έκδοση R2) προγραμματισμένο ως μετατροπέας USB σε σειριακό.

2.3.3 Σχεδιάγραμμα Arduino



Σχήμα 5: Χαρτογράφηση ATmega168 / 328P-Arduino Pin

Η χαρτογράφηση για τα Atmega8, 168 και 328 είναι ίδια. Η πλακέτα μπορεί να λειτουργεί με εξωτερική τροφοδοσία από 6 έως 20 Volt. Το pin των 5V πρέπει να

τροφοδοτείται με πάνω από 7V, ειδάλλως, ενδέχεται να παρέχει λιγότερα από 5 και η πλακέτα μπορεί να γίνει ασταθής. Αντιθέτως, αν χρησιμοποιηθούν πάνω από 12V, ο ρυθμιστής τάσης μπορεί να υπερθερμανθεί και να προκαλέσει βλάβη στην πλακέτα. Οπότε, τελικά το συνιστάμενο εύρος τιμών τάσης τροφοδοσίας είναι 7 έως 12 Volts.

Οι ακίδες-pins τροφοδοσίας της πλακέτας του Arduino είναι οι εξής:

Vin. Η τάση εισόδου στην πλακέτα Arduino / Genuino όταν χρησιμοποιεί εξωτερική πηγή τροφοδοσίας (σε αντίθεση με τα 5 Volts από τη σύνδεση USB ή άλλη ρυθμισμένη πηγή τροφοδοσίας). Είτε τροφοδοτείται το Arduino με τάση μέσω αυτού του συγκεκριμένου pin ή μέσω της υποδοχής τροφοδοσίας.

5V. Αυτός ο ακροδέκτης εκπέμπει 5V από τον ρυθμιστή στην πλακέτα. Η πλακέτα μπορεί να τροφοδοτηθεί με τροφοδοσία είτε από την υποδοχή ρεύματος συνεχούς ρεύματος από το USB (5V), είτε από τον ακροδέκτη Vin του πίνακα (7-12V). Η τροφοδοσία τάσης μέσω των ακροδεκτών 5V ή 3.3V παρακάμπτει τον ρυθμιστή και μπορεί να βλάψει την πλακέτα, οπότε δεν συνιστάται.

3V3. Μια τροφοδοσία 3.3 Volt που παράγεται από το ρυθμιστή που είναι ενσωματωμένος στην πλακέτα. Η μέγιστη ροή ρεύματος είναι 50 mA.

GND. Ακροδέκτες γείωσης.

IOREF. Αυτή η ακίδα στον πίνακα Arduino / Genuino παρέχει την τάση αναφοράς με την οποία λειτουργεί ο μικροελεγκτής. Μια κατάλληλα διαμορφωμένη ασπίδα-shield μπορεί να διαβάσει την τάση στο IOREF pin και να επιλέξει την κατάλληλη πηγή ισχύος ή να ενεργοποιήσει τους μετατροπείς τάσης στις εξόδους για να λειτουργήσει με τα 5V ή 3.3V.

Κάθε μία από τις 14 ψηφιακές ακίδες στο Uno μπορεί να χρησιμοποιηθεί ως είσοδος ή έξοδος χρησιμοποιώντας τις λειτουργίες `pinMode()`, `digitalWrite()` και `digitalRead()`. Η τάση λειτουργίας τους είναι τα 5 Volt, όπως προαναφέρθηκε. Κάθε ακροδέκτης μπορεί να παρέχει ή να λαμβάνει 20 mA ως συνιστάμενη κατάσταση λειτουργίας και έχει εσωτερική αντίσταση (αποσυνδεδεμένη από προεπιλογή) 20-50k Ohm. Η ένταση

ρεύματος δεν θα πρέπει να υπερβαίνει τα 40mA σε οποιονδήποτε ακροδέκτη I/O για να αποφευχθεί η μόνιμη βλάβη του μικροελεγκτή.

Παρατίθενται εξειδικευμένες λειτουργίες κάποιον από τα pins:

Σειριακό: 0 (RX) και 1 (TX). Χρησιμοποιείται για τη λήψη-RX και τη μετάδοση-TX TTL (Time to live) σειριακών δεδομένων. Δηλαδή, δεδομένων με περιορισμένο χρόνο ζωής ή προκαθορισμένο αριθμό συμβάντων, ο οποίος μόλις ξεπεραστεί τα δεδομένα απορρίπτονται ή επαναπροωθούνται. Αυτό επιτυγχάνεται με ένα μετρητή ή χρονική σήμανση TTL, που ενσωματώνεται στα δεδομένα. Οι αναφερθείσες ακίδες συνδέονται με τους αντίστοιχους ακροδέκτες του σειριακού chip ATmega8U2 USB-to-TTL.

Εξωτερικοί διακόπτες: 2 και 3. Αυτές οι ακίδες μπορούν να διαμορφωθούν για να προκαλέσουν διακοπή σε χαμηλή τιμή, ακραία άνοδο ή πτώση ή αλλαγή τιμής. Μέσω της λειτουργίας `attachInterrupt()`.

PWM: 3, 5, 6, 9, 10 και 11. Παρέχετε έξοδο PWM 8-bit με τη λειτουργία `analogWrite()`.

SPI: 10 (SS), 11 (MOSI), 12 (MISO), 13 (SCK). Αυτές οι ακίδες υποστηρίζουν την επικοινωνία SPI χρησιμοποιώντας τη βιβλιοθήκη SPI.

LED: 13. Υπάρχει μια ενσωματωμένη LED που οδηγείται από το ψηφιακό pin 13. Όταν ο ακροδέκτης έχει τιμή HIGH, η λυχνία LED είναι αναμμένη, όταν ο ακροδέκτης έχει τιμή LOW, η λυχνία είναι κλειστή.

TWI: Ακίδα A4 ή SDA και ακίδα A5 ή SCL. Υποστήριξη επικοινωνίας TWI χρησιμοποιώντας τη βιβλιοθήκη Wire.

Το Uno διαθέτει 6 αναλογικές εισόδους, με την ένδειξη A0 έως A5, καθεμία από τις οποίες παρέχει 10 ψηφία ανάλυσης (δηλαδή 1024 διαφορετικές τιμές). Από προεπιλογή, ορίζονται στα 5 Volts, αν και είναι δυνατή η αλλαγή του ανώτερου άκρου του εύρους τους χρησιμοποιώντας το AREF pin και την `analogReference()` λειτουργία.

Τα δύο τελευταία pins στην πλακέτα είναι:

AREF. Τάση αναφοράς για τις αναλογικές εισόδους. Χρησιμοποιείται με analogReference().

Reset. Στην τιμή LOW εκτελεί επαναφορά του μικροελεγκτή. Συνήθως χρησιμοποιείται ως κουμπί επαναφοράς στις ασπίδες-shields που μπλοκάρουν το κουμπί επαναφοράς της πλακέτας του Arduino [12].

2.3.4 Ακριβή Χαρακτηριστικά του Arduino Uno

Στον παρακάτω πίνακα υπάρχουν συγκεντρωμένα τα χαρακτηριστικά του μικροελεγκτή Arduino Uno:

Πίνακας 2: Τα χαρακτηριστικά του Arduino Uno

Μικροελεγκτής	ATmega328P
Τάση Λειτουργίας	5V
Τάση Εισόδου (προτεινόμενη)	7-12V
Τάση Εισόδου (όρια)	6-20V
Ψηφιακές Ακίδες (Pins) I/O	14 (6 παρέχουν έξοδο PWM)
PWM Ψηφιακά Pins I/O	6
Pins αναλογικής εισόδου	6
DC Ρεύμα ανά pin I/O	20 mA
DC Ρεύμα για pin 3.3V	50 mA
Flash Memory (μνήμη που διατηρεί δεδομένα απουσία τροφοδοσίας)	32 KB (ATmega328P) από τα οποία 0.5 KB χρησιμοποιούνται από bootloader (τμήμα κώδικα που εκτελείται πρώτο όταν θέσουμε σε λειτουργία τη συσκευή. Ενεργοποιεί τον πυρήνα της συσκευής και προστατεύει το κύριο λειτουργικό σύστημα καθώς επίσης και τα τμήματα του flash chip, έτσι ώστε να μην μπορούν να διαγραφούν ή να παραβιαστούν από κάποιο κακόβουλο πρόγραμμα.)
SRAM (Static-RAM)	2 KB (ATmega328P)
EEPROM (Ηλεκτρικά απολειψίμη προγραμματίσιμη μνήμη μόνο για ανάγνωση)	1 KB (ATmega328P)
Ταχύτητα Ρολογιού	16 MHz
LED_BUILTIN (λαμπάκι)	13

Μήκος	68.6 mm
Πλάτος	53.4 mm
Βάρος	25 g

2.3.5 Ανανεωμένες εκδόσεις του Arduino Uno.

Η 3^η ανασκευή της πλακέτας , η οποία χρησιμοποιήθηκε στην παρούσα εργασία, έχει τις ακόλουθες προσθήκες:

->1.0 pinout: Επιπρόσθετα pins SDA και SCL που βρίσκονται κοντά στον ακροδέκτη AREF και δύο άλλες νέες ακίδες τοποθετημένες κοντά στον ακροδέκτη RESET, το IOREF που επιτρέπει στα shields να προσαρμόζονται στην τάση που παρέχεται από την πλακέτα. Στο μέλλον, τα shields θα είναι συμβατά τόσο με την πλακέτα που χρησιμοποιεί το AVR, το οποίο λειτουργεί με 5V όσο και με το Arduino Due που λειτουργεί με 3.3V. Το δεύτερο είναι ένα μη συνδεδεμένο pin, που προορίζεται για μελλοντικούς σκοπούς.

->Κατέχει ισχυρότερο κύκλωμα RESET.

->Το Atmega 16U2 αντικαθιστά το 8U2.

2.3.6 Επικοινωνία

Το Arduino/Genuino Uno δύναται να επικοινωνήσει με έναν υπολογιστή, ένα άλλο Arduino/Genuino ή άλλους μικροελεγκτές. Το ATmega328 παρέχει σειριακή ασύγχρονη επικοινωνία UART TTL (5V), η οποία διατίθεται μέσω των ψηφιακών ακίδων 0 (RX) και 1 (TX). Ένα ATmega16U2 στον πίνακα διοχετεύει αυτή τη σειριακή επικοινωνία μέσω USB και εμφανίζεται ως μια εικονική θύρα com για λογισμικό στον υπολογιστή. Το υλικολογισμικό (firmware) 16U2 χρησιμοποιεί τα πρότυπα προγράμματα οδήγησης USB COM και δεν απαιτείται εξωτερικό πρόγραμμα οδήγησης. Ωστόσο, στα Windows, απαιτείται ένα αρχείο .inf. Το λογισμικό Arduino (IDE) περιλαμβάνει μια σειριακή οθόνη που επιτρέπει την αποστολή απλών

στοιχείων κειμένου προς και από την πλακέτα. Οι ενδεικτικές λυχνίες RX και TX στην πλακέτα θα αναβοσβήνουν όταν τα δεδομένα μεταδίδονται μέσω του chip USB-to-serial και της σύνδεσης USB στον υπολογιστή (αλλά όχι για σειριακή επικοινωνία στις ακίδες 0 και 1). Μια βιβλιοθήκη λογισμικού SoftwareSerial επιτρέπει την σειριακή επικοινωνία σε οποιαδήποτε ψηφιακή ακίδα της Uno.

2.3.7 Λογισμικό

Όσον αφορά στο λογισμικό για την κωδικοποίηση στην πλατφόρμα του Arduino, το Arduino Uno μπορεί να προγραμματιστεί σε ένα ολοκληρωμένο περιβάλλον ανάπτυξης (IDE-Integrated Development Environment) του Arduino με βάση το IDE επεξεργασίας (<http://processing.org>), χρησιμοποιώντας μια διασύνδεση USB, ενώ το πρόγραμμα αποθηκεύεται στην εσωτερική EEPROM(Electrically Erasable Programmable Read-Only Memory) του μικροελεγκτή. Ένας επεξεργαστής κώδικα, ο μεταγλωττιστής και η σειριακή επικοινωνία με τους πίνακες Arduino, είναι όλα ενσωματωμένα σε μια ενιαία εφαρμογή Java. Το λογισμικό του Arduino προγραμματίζεται σε Wiring, που κατά βάση είναι C / C ++, που επιτρέπει στο χρήστη να δημιουργήσει ένα πολύπλοκο αντικειμενοστρεφές πρόγραμμα σε πολλά αρχεία, από ένα απλό πρόγραμμα που βασίζεται σε διαδικασίες σε ένα μόνο αρχείο. Επιτρέπει τη διαχείριση σκίτσων-sketches με περισσότερα από ένα αρχεία (καθένα από τα οποία εμφανίζεται σε διαφορετική καρτέλα). Αυτά μπορεί να είναι αρχεία κώδικα Arduino (με επέκταση .ino), αρχεία σε C (με επέκταση .c), αρχεία σε C ++ (.cpp) ή αρχεία κεφαλίδας (.h). Το λογισμικό που χρησιμοποιήθηκε στην παρούσα εργασία είναι το Arduino 1.8.9 .

Διακρίνονται τρία βασικά μέρη της γλώσσας προγραμματισμού του Arduino, ως ακολούθως:

- 1.Συναρτήσεις, ψηφιακών και αναλογικών εισόδων-εξόδων, μαθηματικών, χρόνου, ψηφίων-bits', διακοπών, επικοινωνίας,

2. Τιμές (μεταβλητές και σταθερές) και

3. Δομή

Το Arduino δύναται να τρέξει στα παρακάτω λειτουργικά συστήματα:

1. Macintosh OSX

2. Windows

3. Linux

2.3.7.1 Βιβλιοθήκες Arduino

Το Arduino χρησιμοποιεί κομμάτια κώδικα ως δεδομένα, μαύρα κουτιά, όπως γίνεται και στις περισσότερες πλατφόρμες προγραμματισμού. Αυτά τα μαύρα κουτιά ονομάζονται βιβλιοθήκες, που βοηθούν τον προγραμματιστή στη σύνταξη του κώδικα χρήσης του Arduino. Με τη χρήση βιβλιοθηκών επιτυγχάνεται η επέκταση των δυνατοτήτων και η αύξηση της λειτουργικότητας του περιβάλλοντος του Arduino. Ένας αριθμός βιβλιοθηκών εγκαθίσταται με το IDE. Παρ' όλα αυτά, υπάρχει η δυνατότητα λήψης εκατοντάδων άλλων βιβλιοθηκών από το διαδίκτυο ή δημιουργίας νέων από τους χρήστες, οι οποίες θα πρέπει να εγκατασταθούν για να καταστεί εφικτή η χρήση τους. Η σύνδεση με αισθητήρες, οθόνες, μονάδες και διάφορα άλλα υλικά ή περιφερειακές μονάδες, πραγματοποιείται εύκολα κάνοντας χρήση των βιβλιοθηκών της πλατφόρμας του Arduino. Η χρήση βιβλιοθηκών στο sketch γίνεται εισάγοντας μία ή περισσότερες «#include» δηλώσεις, συνδέοντας με αυτό τον τρόπο τη βιβλιοθήκη με το sketch. Επιπροσθέτως, οι βιβλιοθήκες επιτρέπουν ακόμα την εργασία με hardware και τη διαχείριση δεδομένων.

2.3.7.2 Arduino sketch

Sketches ονομάζονται τα προγράμματα που γράφονται χρησιμοποιώντας το λογισμικό IDE του Arduino. Τα sketches συντάσσονται στον επεξεργαστή κειμένου και αποθηκεύονται με την επέκταση «.ino». Επιπλέον, μπορεί να γίνει διάγνωση σφαλμάτων και δίνεται ανάδραση στο χρήστη μέσω μηνυμάτων κατά την αποθήκευση και την εξαγωγή του sketch. Η κονσόλα της πλατφόρμας του λογισμικού IDE εμφανίζει την έξοδο κειμένου του sketch, συμπεριλαμβανομένων πλήρων μηνυμάτων σφάλματος και άλλων πληροφοριών. Στην κάτω δεξιά γωνία του παραθύρου εμφανίζεται η διαμορφωμένη πλακέτα και η σειριακή θύρα. Μέσω της γραμμής εργαλείων γίνεται η επαλήθευση και το ανέβασμα των προγραμμάτων.

→ Sketch Uploading

Για το ανέβασμα του sketch στη μονάδα του Arduino θα πρέπει να επιλέξουμε την επιθυμητή συσκευή και την κατάλληλη σειριακή θύρα σύνδεσης. Σε λειτουργικό σύστημα Windows, η σειριακή θύρα μπορεί να είναι η COM1 ή COM2 (για μια σειριακή πλακέτα) ή COM3, COM4, COM5, COM7 ή υψηλότερη (για μια κάρτα USB). Για την ταυτοποίηση της συσκευής μας ως σειριακή συσκευή USB μπορούμε να ανατρέξουμε στην ενότητα «θύρες» της Διαχείρισης Συσκευών των Windows. Στις περισσότερες πλακέτες Arduino, τα λαμπάκια-leds των RX και TX αναβοσβήνουν καθώς φορτώνεται το sketch. Το λογισμικό Arduino (IDE) θα εμφανίσει ένα μήνυμα ολοκλήρωσης της μεταφόρτωσης ή θα εμφανίσει μήνυμα σφάλματος.

2.3.7.3 Arduino Bootloader

Ένα μικρό πρόγραμμα που έχει φορτωθεί στον μικροελεγκτή της πλακέτας του Arduino και ονομάζεται εκκινητής φόρτωσης (bootloader) χρησιμοποιείται όταν ένα sketch φορτώνεται. Επιτρέπει τη φόρτωση κώδικα χωρίς πρόσθετο hardware και επικοινωνεί με βάση το πρωτόκολλο STK500. Ο bootloader είναι ενεργός για μερικά δευτερόλεπτα όταν η πλακέτα επαναρυθμίζεται. Εν συνεχεία, ανοίγει το τελευταίο sketch που ήταν ανεβασμένο στον μικροελεγκτή. Ο εκκινητής φόρτωσης θα αναβοσβήνει την ενδεικτική λυχνία (pin 13) κατά την εκκίνηση (δηλ. Όταν η πλακέτα

επαναφέρεται). Ωστόσο, υπάρχει η δυνατότητα παράκαμψης του bootloader μέσω της κεφαλίδας ICSP (σειριακός προγραμματισμός εντός κυκλώματος) χρησιμοποιώντας τον ISP(In-System Programming) Arduino ή παρόμοιο, για τον προγραμματισμό του μικροελεγκτή.

2.3.7.4 Arduino Serial Monitor

Το λογισμικό του Arduino περιλαμβάνει τη λεγόμενη σειριακή οθόνη (Serial Monitor). Το serial monitor εμφανίζει σειριακά αποστελλόμενα δεδομένα από την πλακέτα Arduino ή Genuino μέσω USB ή σειριακής σύνδεσης. Για αποστολή δεδομένων στην πλακέτα αρκεί η εισαγωγή κειμένου και η αποστολή του. Η ταχύτητα μετάδοσης (baud rate) επιλέγεται από αναδυόμενο μενού να ταιριάζει με το ρυθμό που ορίστηκε μέσω της εντολής `Serial.begin` στο sketch. Αξιοσημείωτο είναι ότι στα Windows, Mac ή Linux η πλακέτα θα επαναφέρει το sketch όταν συνδεθεί με τη σειριακή οθόνη. Επειδή το serial monitor δεν μπορεί να επεξεργασθεί χαρακτήρες ελέγχου χρησιμοποιείται εξωτερικό τερματικό πρόγραμμα που συνδέεται στη θύρα COM που έχει αντιστοιχιστεί στον πίνακα Arduino, για πλήρη διαχείριση της σειριακής επικοινωνίας με χαρακτήρες ελέγχου.

Οι πλακέτες Arduino διαφέρουν μεταξύ τους, οπότε ο χρήστης μπορεί να επιλέξει την πλακέτα που αρμόζει, λαμβάνοντας υπόψη τον καθορισμό των επιθυμητών παραμέτρων (π.χ. ταχύτητα CPU και ρυθμό baud) που θα χρησιμοποιούνται κατά τη σύνταξη-compiling και φόρτωση του sketch, καθώς επίσης τον καθορισμό των ρυθμίσεων αρχείων και ασφαλειών που χρησιμοποιούνται από την εντολή εκκίνησης του bootloader.

2.3.7.5 Αυτόματη επαναφορά λογισμικού

Αντί να απαιτείται πίεση του κουμπιού επαναφοράς πριν από κάθε ανέβασμα κώδικα, η πλακέτα Arduino / Genuino Uno σχεδιάστηκε κατά τρόπο που να επιτρέπει την επαναφορά της από λογισμικό που εκτελείται σε συνδεδεμένο υπολογιστή. Η δυνατότητα αυτή που παρέχει το λογισμικό Arduino (IDE) επιτρέπει το ανέβασμα κώδικα, απλά πατώντας το κουμπί αποστολής στη γραμμή εργαλείων της διεπαφής.

2.3.8 Λειτουργίες Arduino

Το Arduino μπορεί να λειτουργεί αυτόνομα χωρίς να είναι συνδεδεμένο με υπολογιστή. Διάφορες εξωτερικές μονάδες υλικού που ονομάζονται ασπίδες (shields) μπορούν να συνδεθούν με τη μονάδα του Arduino μέσω μιας ορισμένης ακίδας (pin) της πλακέτας και να επεκτείνουν τις λειτουργίες του. Αναφορικά, κάποια παραδείγματα λειτουργιών είναι η προσθήκη δυνατοτήτων δικτύωσης όπως το Ethernet, Bluetooth, ZigBee, δυνατότητα TFT αφής, δυνατότητα καταγραφής δεδομένων χωρίς την ανάγκη υπολογιστή, joystick και συνδυασμό κουμπιών για ανάγνωση εισόδου χρήστη και συγκεκριμένα shields για οδήγηση LED. Στην πραγματικότητα, η Ethernet shield δίνει στο Arduino μια διεύθυνση IP που του επιτρέπει να ελέγχεται μέσω του Διαδικτύου, η Bluetooth shield επιτρέπει την ασύρματη επικοινωνία με κινητά τηλέφωνα που εκτελούνται σε Android και iOS / iPhone και η αλληλεπίδραση ZigBee shield με ασύρματα δίκτυα αισθητήρων. Τα shields μπορούν να παρέχουν έλεγχο σε κινητήρες, GPS, Ethernet, LCD εικόνας ή breadboarding (προτυποποίηση με ηλεκτρονική πλακέτα) [13].

2.3.9 Επιλογή Arduino για την εργασία

Το Arduino ως υλικό ανοικτού κώδικα, επιτρέπει την υλοποίηση εμπορικών και μη εμπορικών εφαρμογών της ίδιας ιδέας. Ως εκ τούτου, είναι επόμενο μια εφαρμογή να αναγκαστεί να υποστεί την απόσυρση, λόγω δημιουργίας φθηνότερης ή μικρότερης πλατφόρμας ή εναλλακτικά να αναβαθμίσει ορισμένα χαρακτηριστικά.

Οι λόγοι επιλογής της συγκεκριμένης πλατφόρμας μικροελεγκτή του Arduino που επελέγη για την παρούσα εργασία και υπερισχύει μεταξύ άλλων είναι οι ακόλουθοι:

1. Είναι συγκριτικά φθηνότερη.
2. Προσαρμόζεται σε διάφορα λειτουργικά συστήματα
3. Έχει απλό στη σύλληψη και σαφές περιβάλλον προγραμματισμού. Είναι εύπεπτο για μαθησιακούς σκοπούς, αλλά και προχωρημένες ιδέες μπορούν να βρουν εφαρμογή λόγω της ευελιξίας της πλατφόρμας.
4. Διαθέτει λογισμικό (software) επεκτάσιμο, ανοιχτού κώδικα. Το λογισμικό Arduino δημοσιεύεται ως εργαλεία ανοιχτού κώδικα, που διατίθενται για επέκταση από έμπειρους προγραμματιστές με χρήση των βιβλιοθηκών της γλώσσας προγραμματισμού C ++ . Η γλώσσα προγραμματισμού AVR-C είναι η βάση συγγραφής του κώδικα του Arduino, και μπορεί να χρησιμοποιηθεί και απευθείας στα προγράμματα Arduino.
5. Διαθέτει επεκτάσιμο και ανοιχτού κώδικα υλικό (hardware). Δίνεται η δυνατότητα σε έμπειρους κυρίως σχεδιαστές κυκλωμάτων μέσω άδειας Creative Commons, βάση της οποίας δημοσιεύονται οι πλακέτες του Arduino, να επεκτείνουν ή να δημιουργήσουν νέες, καινοτόμες εκδόσεις πλακετών Arduino [14].

2.3.10 Σύγκριση συσκευών Arduino

Παρακάτω τίθεται ένας πίνακας σύγκρισης μεταξύ των διαφόρων μονάδων Arduino και Genuino:

NAME	Processor	Operating/ Input Voltage	CPU Speed	Analog In/Out	Digital IO/ PWM	EEPROM [kB]	SRAM [kB]	Flash [kB]	USB	UART
101	Intel® Curie	3.3 V/ 7-12V	32 MHz	6/0	14/4	-	24	196	Regular	-
Gemma	ATtiny85	3.3 V / 4-16 V	8 MHz	1/0	3/2	0.5	0.5	8	Micro	0
LilyPad	ATmega168V ATmega328P	2.7-5.5 V / 2.7-5.5 V	8 MHz	6/0	14/6	0.512	1	16	-	-
LilyPad SimpleSnap	ATmega328P	2.7-5.5 V / 2.7-5.5 V	8 MHz	4/0	9/4	1	2	32	-	-
LilyPad USB	ATmega32U4	3.3 V / 3.8-5 V	8 MHz	4/0	9/4	1	2.5	32	Micro	-
Mega 2560	ATmega2560	5 V / 7-12 V	16 MHz	16/0	54/15	4	8	256	Regular	4
Micro	ATmega32U4	5 V / 7-12 V	16 MHz	12/0	20/7	1	2.5	32	Micro	1
MKR1000	SAMD21 Cortex-M0+	3.3 V/ 5V	48 MHz	7/1	8/4	-	32	256	Micro	1
Pro	ATmega168 ATmega328P	3.3 V / 3.35- 12 V 5 V / 5-12 V	8 MHz 16 MHz	6/0	14/6	0.512 1	1 2	16 32	-	1
Pro Mini	ATmega328P	3.3 V / 3.35- 12 V 5 V / 5-12 V	8 MHz 16 MHz	6/0	14/6	1	2	32	-	1
Uno	ATmega328P	5 V / 7-12 V	16 MHz	6/0	14/6	1	2	32	Regular	1
Zero	ATSAMD21G1 8	3.3 V / 7-12 V	48 MHz	6/1	14/10	-	32	256	2 Micro	2
Due	ATSAM3X8E	3.3 V / 7-12 V	84 MHz	12/2	54/12	-	96	512	2 Micro	4
Esplora	ATmega32U4	5 V / 7-12 V	16 MHz	-	-	1	2.5	32	Micro	-
Ethernet	ATmega328P	5 V / 7-12 V	16 MHz	6/0	14/4	1	2	32	Regular	-

Leonardo	ATmega32U4	5 V / 7-12 V	16 MHz	12/0	20/7	1	2.5	32	Micro	1
Mega ADK	ATmega2560	5 V / 7-12 V	16 MHz	16/0	54/15	4	8	256	Regular	4
Mini	ATmega328P	5 V / 7-9 V	16 MHz	8/0	14/6	1	2	32	-	-
Nano	ATmega168 ATmega328P	5 V / 7-9 V	16 MHz	8/0	14/6	0.512 1	1 2	16 32	Mini	1
Yùn	ATmega32U4 AR9331 Linux	5 V	16 MHz 400 MHz	12/0	20/7	1	2.5 16MB	32 64MB	Micro	1
Arduino Robot	ATmega32u4	5 V	16 MHz	6/0	20/6	1 KB (ATmega 32u4)/ 512 Kbit (I2C)	2.5 KB (ATmeg a32u4)	32 KB (ATmega3 2u4) of which 4 KB used by bootloader	1	1
MKRZero	SAMD21 Cortex-M0+ 32bit low power ARM MCU	3.3 V	48 MHz	7 (ADC 8/10/1 2 bit)/1 (DAC 10 bit)	22/12	No	32 KB	256 KB	1	1

Πίνακας 3: Χαρακτηριστικά διαφόρων συσκευών Arduino [13]

Χαρακτηριστικά πλακετών Arduino που έχουν αποσυρθεί:

NAME	Processor	Operating/ Input Voltage	CPU Speed	Analog In/Out	Digital IO/ PWM	EEPROM [kB]	SRAM [kB]	Flash [kB]	USB	UART
BT	ATmega328P	5 V/ 2.5-12V	16 MHz	6/0	14/6	1	2	32	-	1
Fio	ATmega328P	3.3 V / 3.7-7 V	8 MHz	8/0	14/6	1	2	32	Mini	1

Πίνακας 4: Χαρακτηριστικά απαρχαιωμένων συσκευών Arduino [13]

2.3.11 Προϊόντα Arduino

Υπάρχουν προϊόντα Arduino για αρχάριους που μπορούν να χρησιμοποιηθούν σε απλές εφαρμογές. Οι συγκεκριμένες μονάδες Arduino είναι εύχρηστες και απλές στην κατανόηση, που βοηθούν έτσι το χρήστη να εξοικειωθεί με το περιβάλλον της ηλεκτρονικής και της κωδικοποίησης. Με αυτό τον τρόπο δίνεται, πιο προσιτά, η ευκαιρία υλοποίησης έργων και έκρηξης της δημιουργικότητας με απτά αποτελέσματα σε ένα εύκολα διαχειρίσιμο περιβάλλον.

Βέβαια, υπάρχουν και πιο προηγμένα μοντέλα Arduino με μεγαλύτερες και ταχύτερες επιδόσεις, πιο εξελιγμένες λειτουργίες και περισσότερες δυνατότητες για πιο προχωρημένους και εξειδικευμένους χρήστες με τα στοιχεία της ηλεκτρονικής και την κωδικοποίηση.

Ενώ υπάρχουν, ακόμα, σύγχρονες μονάδες που μπορούν να συνδεθούν με προϊόντα IoT-Internet of Things, μιας αναδυόμενης τεχνολογίας που επιτρέπει την διαχείριση ηλεκτρονικών και ηλεκτρικών συσκευών και διεργασιών απομακρυσμένα, με αισθητήρες και άλλες μεθόδους όπως ασύρματης επικοινωνίας (π.χ. Bluetooth, Wi-Fi κ.α.) [15].

2.4 ANDROID

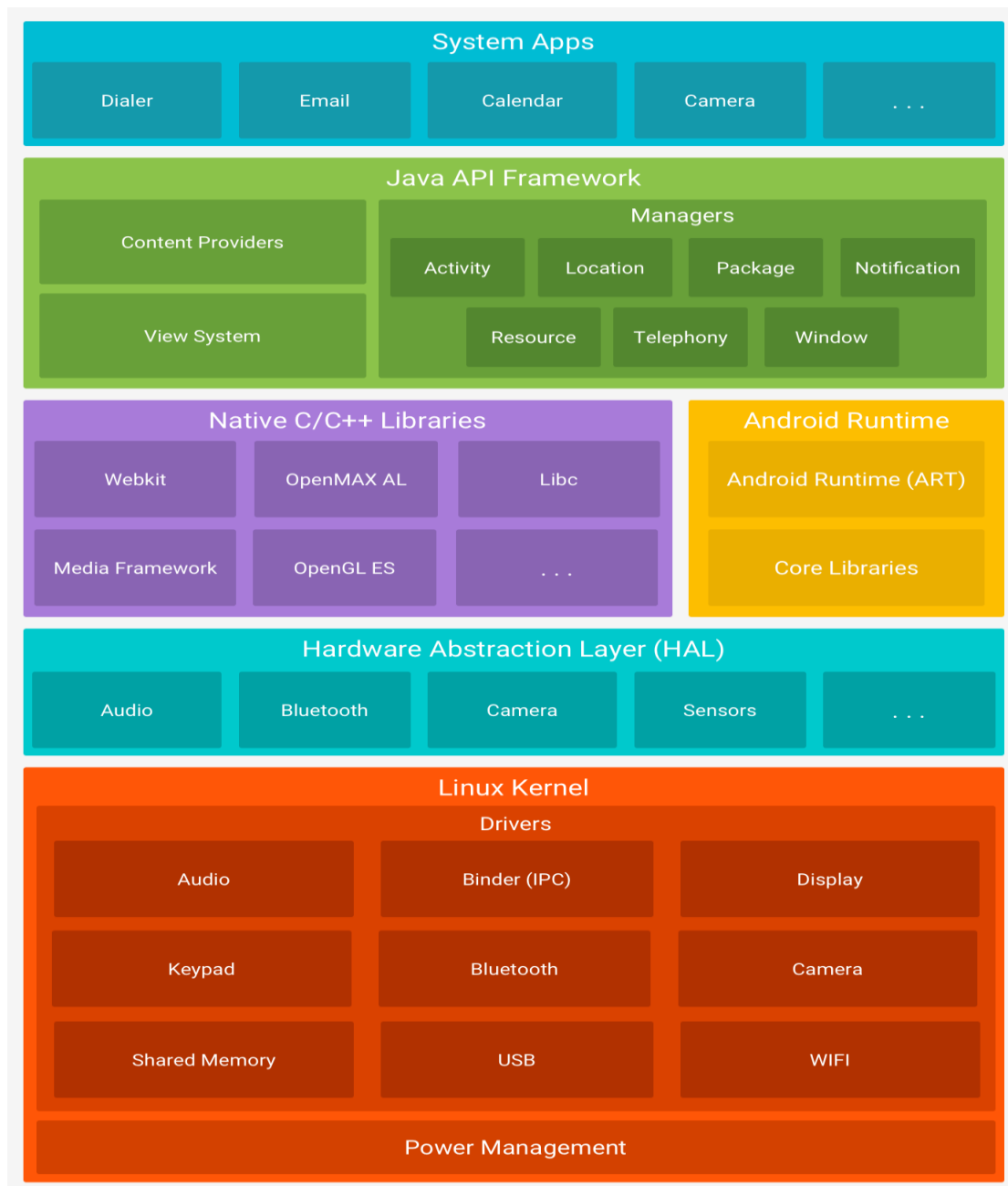
2.4.1 Γενικά

Στις μέρες μας, έχουν παραχθεί ποικίλες σύγχρονες υπολογιστικές συσκευές που έχουν πολλές δυνατότητες επικοινωνίας μεταξύ τους. Η αλληλεπίδραση αυτών των συσκευών με τους ανθρώπους είναι εφικτή σε πολλαπλά επίπεδα και εξέχουσας σημασίας για τους κατασκευαστές αλλά και για τους χρήστες. Μία από τις πιο δημοφιλείς συσκευές παγκοσμίως, που η αλληλεπίδραση τους με τους χρήστες αλλά και η συμβατότητά τους με άλλες συσκευές εξελίσσεται σε υποδειγματική, είναι τα έξυπνα κινητά (smartphones). Ο αριθμός των χρηστών κινητών συσκευών, συμπεριλαμβανομένων των χρηστών smartphones υφίστανται ραγδαία αύξηση σε παγκόσμιο επίπεδο. Αυτή η κοινωνική ροή έχει ωθήσει τη βιομηχανία που σχετίζεται με τις ηλεκτρονικές συσκευές στην ανάπτυξη ποικίλων και πολυάριθμων εφαρμογών για χρήση σε smartphones που διευκολύνουν την καθημερινότητα. Ο έλεγχος διάφορων συσκευών και διεργασιών που συνθέτουν την καθημερινή ζωή, πέρα από συνήθεις εσωτερικές διεργασίες, όπως πραγματοποίηση ή λήψη τηλεφωνικών κλήσεων, αποστολή μηνυμάτων κειμένου και εκτέλεση απομακρυσμένων τραπεζικών συναλλαγών είναι πλέον εφικτή για κάθε χρήστη smartphone μέσω ελεύθερων ή επί πληρωμή εφαρμογών. Εν ολίγοις, αυτό σημαίνει ότι μέσω εφαρμογών, μπορούμε να ελέγξουμε απομακρυσμένα μια ποικιλία εξωτερικών συσκευών όπως τηλεοράσεις, προβολείς για παρουσιάσεις, υπολογιστές, ακόμη και αυτοκίνητα ή οικιακές συσκευές. Αυτή η δυνατότητα έχει οδηγήσει στην εξέλιξη του κλάδου του Διαδικτύου των Πραγμάτων (IoT).

2.4.2 Λειτουργικό Σύστημα Android

Η Google είναι η εταιρεία που έχει αναπτύξει το λειτουργικό σύστημα ανοικτού κώδικα Android για κινητά τηλέφωνα και υπολογιστές ταμπλέτες ή πιο επιστημονικά προσωπικούς ψηφιακούς βοηθούς (tablets). Η δομή συστήματος του Android αποτελείται από τέσσερα επίπεδα: Επίπεδο εφαρμογής (application layer), επίπεδο

πλαίσιο εφαρμογής (application framework layer), επίπεδο χρόνου εκτέλεσης συστήματος (systems runtime layer) και επίπεδο πυρήνα Linux (Linux kernel layer).



Σχήμα 5: Τα βασικά δομικά στοιχεία της πλατφόρμας Android [16]

Το λειτουργικό σύστημα Android αποτελείται όχι μόνο από ένα λειτουργικό σύστημα, αλλά και από το ενδιάμεσο λογισμικό (Middleware), μια διεπαφή χρήστη (UI), ένα πρόγραμμα περιήγησης και διάφορες εφαρμογές. Περιλαμβάνει επίσης βιβλιοθήκες C / C ++ που χρησιμοποιούνται για τη σύσταση διαφόρων συστημάτων Android.

Το Android είναι η πιο δημοφιλής πλατφόρμα για εφαρμογές που μπορούν να χρησιμοποιηθούν σε κινητά τηλέφωνα, smartphones και tablets. Ο αριθμός των διαθέσιμων εφαρμογών για λήψη πλησιάζει τα 2.8 εκατομμύρια στο κατάστημα Google Play, ελεύθερες, αλλά και επί μισθώσεως. Είναι γεγονός πως η συγκεκριμένη πλατφόρμα εξελίσσεται με πολύ γρήγορους ρυθμούς. Σε πολύ σύντομο χρονικό διάστημα, περιόδου μιας δεκαετίας, η πλατφόρμα έχει εξελιχθεί σε επίπεδο 29 API (Android 10) [17].

2.4.3 Εκδόσεις Android

Παρακάτω αναγράφονται με σειρά απ' την πιο σύγχρονη στην παλαιότερη οι εκδόσεις του λογισμικού Android:

Android 10 (API level 29): Διαμορφώνει το πλαίσιο της ρηξικέλευθης εφαρμογής προηγμένης εκμάθησης μηχανών και υποστήριξης για κινητά τηλέφωνα 5^{ης} γενιάς. Αυξάνει την ασφάλεια χρήσης και ενισχύει το ιδιωτικό απόρρητο. Ακόμα, διευκολύνει την επίτευξη ισορροπίας των οικιακών χρηστών με την τεχνολογία και τις νέες ψηφιακές προοπτικές.

Android 9 (API level 28)

Android 8.1 (API level 27)

Android 8.0 (API level 26)

Android 7.1 (API level 25)

Android 7.0 (API level 24)

Android 6.0 (API level 23)

Android 5.1 (API level 22)

Android 5.0 (API level 21)

Android 4.4W (API level 20): Αυτή η έκδοση καθιστά το KitKat διαθέσιμο για Android Wear

Android 4.4 (API level 19)

Android 4.3 (API level 18)

Android 4.2 (API level 17)

Android 4.1 (API level 16)
Android 4.0.3 (API level 15)
Android 4.0 (API level 14)
Android 3.2 (API level 13)
Android 3.1 (API level 12)
Android 3.0 (API level 11)
Android 2.3.3 (API level 10)
Android 2.3 (API level 9)

API (Application Programming Interface)-Διεπαφή Προγραμματισμού Εφαρμογών: είναι πρωτόκολλο επικοινωνίας μεταξύ διαφορετικών μερών ενός προγράμματος, που επιτρέπει την εκτέλεση και τη συντήρηση του λογισμικού.

Το επίπεδο API (API level) αφορά τη λειτουργικότητα των εφαρμογών και το πλήθος των συσκευών που μπορούν να την χρησιμοποιήσουν αποτελεσματικά. Αυτό σημαίνει για παράδειγμα ότι αν ο προγραμματιστής επιλέξει ένα χαμηλό σχετικά επίπεδο API για την εφαρμογή του αυτή σίγουρα θα μπορεί να εγκατασταθεί και να λειτουργήσει σε περισσότερες συσκευές, αλλά πιθανότατα θα υστερεί σε λειτουργικότητα. Όπως επίσης, θα χρειαστεί πολύ περισσότερη δουλειά και γραμμές κώδικα, από την μεριά του προγραμματιστή για να επιτύχει να έχει η εφαρμογή του χαρακτηριστικά που θα είχε αν επέλεγε υψηλότερου επιπέδου API.

Οι περισσότερες από τις εφαρμογές επεκτείνουν τις τρέχουσες λειτουργίες, συμπεριλαμβανομένων ευαίσθητων λειτουργιών, προσθέτοντας στοιχεία και δικαιώματα. Ωστόσο, έχει παρατηρηθεί ότι η προσθήκη ή η αφαίρεση λειτουργιών με την πάροδο του χρόνου δεν επηρεάζει αρνητικά την επιτυχία τους. Παρ' όλα αυτά, οι περισσότερες εφαρμογές βελτιώνουν το σχεδιασμό και τις λειτουργίες της διεπαφής χρήστη μεταβάλλοντας χαρακτηριστικά, όπως θέμα, εικονίδιο και πλοήγηση. Οι προγραμματιστές εφαρμογών κάνουν συνεχώς αλλαγές στο περιβάλλον χρήστη για να βελτιώσουν τη συνολική ποιότητα των εφαρμογών [18].

2.4.4 Γλώσσα Προγραμματισμού Android

Το Android εκτός από λειτουργικό σύστημα, που όπως αναφέρθηκε παραπάνω βρίσκει απτή εφαρμογή και κατέχει κυρίαρχη θέση χρήσης στην πλειοψηφία των κινητών και ενσωματωμένων συσκευών, όπως κινητά τηλέφωνα, tablets και έξυπνες τηλεοράσεις, αποτελεί, επιπλέον, γλώσσα προγραμματισμού στην οποία προγραμματίζονται τα ανώτερα στρώματα των αναφερθέντων συσκευών και βασίζεται στην Java . Ως γλώσσα προγραμματισμού, το Android είναι ουσιαστικά Java με μια εκτεταμένη βιβλιοθήκη για κινητές και διαδραστικές εφαρμογές, με βάση αντικειμενοστρεφή αρχιτεκτονική, με δυναμική ποικιλία γραφικών από δηλωτικά αρχεία διαμόρφωσης-XML. Οποιοσδήποτε μεταγλωττιστής Java μπορεί να μεταγλωττίσει εφαρμογές Android, αλλά χρειάζεται επιπλέον μεταγλώττιση για να μπορεί η εφαρμογή να εγκατασταθεί στη συσκευή και να λειτουργήσει απρόσκοπτα στο περιβάλλον των συσκευών για αυτό γίνεται συνήθως χρήση του περιβάλλοντος προγραμματισμού Android Studio που είναι σχεδιασμένο να εκτελεί ολοκληρωμένα τις διεργασίες δημιουργίας μια εφαρμογής από το αρχικό στάδιο της δημιουργίας του αρχείου, που περιλαμβάνει το δηλωτικό αρχείο, τη βασική κλάση και το αρχείο παρουσίασης, μέχρι και το τελικό στάδιο της μεταγλώττισης και της εκτέλεσης του κώδικα [19].

2.4.5 Δομικά Χαρακτηριστικά Εφαρμογών Android

Οι εφαρμογές Android, που όπως αναφέρθηκε, γράφονται σε γλώσσα προγραμματισμού Java μπορούν να εκτελεστούν με δική τους διεργασία στο δικό τους εικονικό μηχάνημα. Οι εφαρμογές Android συμβαίνει να αλληλεξαρτώνται χρησιμοποιώντας και τμήματα άλλων εφαρμογών Android, ζητώντας άδεια χρήσης κάθε φορά ή άπαξ, καθώς και υπηρεσίες των ως άνω αναφερόμενων εφαρμογών, καλώντας τους χειριστές συμβάντων τους (event handlers), απευθείας ή μέσω του λειτουργικού συστήματος. Συγκεκριμένα, οι εφαρμογές Android περιέχουν δραστηριότητες (κώδικα που αλληλοεπιδρά με τον χρήστη μέσω οπτικής διεπαφής),

υπηρεσίες (λειτουργίες στο παρασκήνιο χωρίς αλληλεπίδραση με τον χρήστη), πάροχους περιεχομένου (όπως βάσεις δεδομένων) και δέκτες αναμετάδοσης (αντικείμενα που αντιδρούν σε μηνύματα εκπομπής). Οι χειριστές συμβάντων δεν διακρίνονται από ορισμένη διάταξη προγραμματισμού, με κάποιες αξιοσημείωτες εξαιρέσεις, όπως ο κύκλος ζωής των δραστηριοτήτων (activities). Στο πακέτο μιας εφαρμογής Android καταχωρείται-δημιουργείται με εντολές του χρήστη σε πλήρη αντιστοιχία με τις συναρτήσεις διαχείρισης που δημιουργούνται από τον προγραμματιστή ένα δηλωτικό, όπως καλείται αρχείο (manifest file) XML, το οποίο καταχωρεί τα στοιχεία μιας εφαρμογής. Επιτελεί ρόλο διαχειριστικό και χωρίς αυτό είναι αδύνατον να λειτουργήσει η εφαρμογή. Άλλα αρχεία XML περιγράφουν την οπτική διάταξη των δραστηριοτήτων.

2.4.5.1 Δηλωτικό Αρχείο Android – Αρχείο XML

Το Android manifest file είναι ένα αρχείο ρυθμίσεων που χρησιμοποιείται για να δηλώσει διάφορα βασικά χαρακτηριστικά των εφαρμογών, επαφίεται στην ευχέρεια σύνταξης των προγραμματιστών και είναι σε μορφή .xml . Οι προγραμματιστές δηλώνουν διάφορα στοιχεία-elements και χαρακτηριστικά-attributes στο Android manifest file που περιγράφει βασικές πληροφορίες σχετικά με την εφαρμογή. Οι πληροφορίες στο Android manifest file χρησιμοποιούνται όχι μόνο από το περιβάλλον εκτέλεσης των εφαρμογών, αλλά και από εξωτερικές πλατφόρμες, όπως το Google Play.

Οι πληροφορίες που περιέχονται σε ένα manifest file είναι βασικές για τη λειτουργία της εφαρμογής. Το manifest file περιέχει βασικά λειτουργικά χαρακτηριστικά και χαρακτηριστικά ασφαλείας, πέραν των ποιοτικών χαρακτηριστικών, όπως το θέμα, το εικονίδιο και το όνομα εφαρμογής. Αυτά τα attributes ποιότητας, είναι σημαντικοί παράγοντες που επηρεάζουν την επιλογή εφαρμογών από τους χρήστες Android.

2.4.5.2 Συστατικά Στοιχεία - Components

Τα στοιχεία-components αποτελούν τις δομικές μονάδες εφαρμογών Android. Χαρακτηριστικά-features ή οι λειτουργίες-functionalities σε εφαρμογές Android υλοποιούνται μέσω των components. Οι εφαρμογές Android μπορούν να κατασκευαστούν χρησιμοποιώντας τέσσερις τύπους components που μπορούν να επεκτείνουν τις λειτουργίες μιας εφαρμογής:

α.δραστηριότητες-activities,

β.υπηρεσίες-services,

γ.δέκτες αναμετάδοσης-broadcast receivers και

δ.παρόχους περιεχομένου-content providers,

οι οποίοι δηλώνονται στο Android manifest file χρησιμοποιώντας τα στοιχεία-elements <activity>, <service>, <receiver> και <provider> , αντίστοιχα. Στο Android manifest file πρέπει να δηλώνονται τα συστατικά στοιχεία-components μιας εφαρμογής, τα οποία μπορεί να εκκινήσει το σύστημα. Κάθε component έχει ξεχωριστή λειτουργικότητα στις εφαρμογές Android.

2.4.5.3 Components και Λειτουργικότητα

Μια δραστηριότητα-activity εξομοιώνει την οθόνη ενός χρήστη, μια υπηρεσία-service χρησιμοποιείται για να χειριστεί διεργασίες που χρειάζεται να τρέχουν για ώρα, ένας δέκτης αναμετάδοσης-broadcast receiver χειρίζεται τα γεγονότα αναμετάδοσης και ένας πάροχος περιεχομένου content provider διαχειρίζεται την πρόσβαση στα δεδομένα.

2.4.5.4 Components και Εμπειρία Χρήστη

Ο προγραμματιστής της εφαρμογής επωμίζεται, εκτός των άλλων και το σχεδιασμό μιας λειτουργικής, εύχρηστης και ευχάριστης προς το χρήστη διεπαφής (UI-Used Interface Design). Μια πετυχημένη διεπαφή μπορεί να εκτοξεύσει την ζήτηση της εφαρμογής. Για αυτό πρέπει να σχεδιάζεται με προσοχή και μεθοδικότητα. Κάθε εφαρμογή πρέπει να ορίσει ένα όνομα πακέτου-package name στο αρχείο δήλωσης Android, το οποίο χρησιμοποιείται γενικά ως το μοναδικό αναγνωριστικό της εφαρμογής στο σύστημα και στο Google Play. Ορισμένα attributes μπορούν να καθοριστούν τόσο για τα στοιχεία όσο και για την εφαρμογή, όπως εικονίδιο-icon, θέμα-theme και πλοήγηση-navigation.

Η πλοήγηση στην εφαρμογή είναι ένα σημαντικό στοιχείο που επηρεάζει άμεσα την εμπειρία του χρήστη. Οι δραστηριότητες-activities είναι διατεταγμένες σε μια στοίβα που λειτουργεί στο παρασκήνιο. Το σύστημα διατηρεί τις πληροφορίες πλοήγησης σε αυτή τη στοίβα, κατά τη διάρκεια της περιήγησης στην εφαρμογή. Αυτό επιτρέπει στο χρήστη να επιστρέψει σε προηγούμενη οθόνη στην οποία έχει πλοηγηθεί νωρίτερα, πατώντας το πλήκτρο της επιστροφής. Το σύστημα διαχειρίζεται τις διεργασίες και τη στοίβα παρασκηνίου τοποθετώντας όλες τις διαδοχικές δραστηριότητες στην ίδια διεργασία σε διάταξη lifo (last in first out).

➔ Πλοήγηση

Η ιδιότητα πλοήγησης καθορίζει τον τρόπο με τον οποίο οι χρήστες μπορούν να πλοηγηθούν μεταξύ οθονών ή δραστηριοτήτων, όπως παραδείγματος χάριν με κουμπιά (button) ή με ολίσθηση (sliding). Από την αρχική οθόνη και μετά είναι πολύ χρήσιμο και φιλικό στο χρήστη να υπάρχει κουμπί που θα μπορεί να επιστρέψει το χρήστη σε προηγούμενη-προγονική οθόνη της τρέχουσας.

➔ Προσανατολισμός Συσκευής

Οι συσκευές Android υποστηρίζουν διάφορους προσανατολισμούς οθόνης, όπως πορτρέτο και τοπίο. Οι προγραμματιστές πρέπει να παρέχουν υποστήριξη σε όλους τους προσανατολισμούς στις εφαρμογές τους για μέγιστη προσαρμογή στις ανάγκες των διάφορων συσκευών και των τύπων εφαρμογών.

➔ Όνομα Πακέτου

Κάθε εφαρμογή πρέπει να έχει δηλώσει ένα όνομα πακέτου στο αρχείο δήλωσης Android. Το όνομα του πακέτου χρησιμοποιείται γενικά ως αναγνωριστικό εφαρμογής, το οποίο πρέπει να είναι μοναδικό στο Play Store. Ωστόσο, το όνομα του πακέτου και το αναγνωριστικό εφαρμογής μπορεί να διαφέρουν σε μια εφαρμογή. Μόλις δημοσιευτεί η εφαρμογή στο Play Store, η νέα έκδοση της εφαρμογής πρέπει να έχει το ίδιο αναγνωριστικό εφαρμογής. Διαφορετικά, η νέα έκδοση της εφαρμογής θεωρείται διαφορετική εφαρμογή.

➔ Εικονίδιο Εφαρμογής

Ένα εικονίδιο εφαρμογής δηλώνεται χρησιμοποιώντας ένα attribute εικονιδίου στο στοιχείο <application> του αρχείου προέλευσης του Android. Αντιπροσωπεύει ένα εικονίδιο για την εφαρμογή και το προεπιλεγμένο εικονίδιο για κάθε στοιχείο της εφαρμογής. Το εικονίδιο μιας εφαρμογής αποτελεί πολύ κρίσιμο παράγοντα για την επιλογή ή μη της εφαρμογής από τους χρήστες.

➔ Θέμα (Theme) Εφαρμογών

Ένα θέμα είναι ένας τύπος στυλ που εφαρμόζεται σε μια ολόκληρη εφαρμογή ή μια δραστηριότητα. Εφαρμόζεται μέσω της δήλωσης ενός theme attribute στα στοιχεία <application> ή <activity> του αρχείου manifest του Android. Αν τα διαφορετικά θέματα δηλώνονται στα στοιχεία <application> και <activity>, το θέμα της δραστηριότητας αντικαθιστά το θέμα της εφαρμογής. Η πλατφόρμα Android παρέχει διάφορα ενσωματωμένα θέματα και βιβλιοθήκες υποστήριξης για θέματα. Ωστόσο,

είναι στη διάθεση των προγραμματιστών να δημιουργήσουν και να χρησιμοποιήσουν προσαρμοσμένα στις επιθυμίες τους θέματα.

→ Δημιουργία Αντιγράφων Ασφαλείας και Επαναφορά

Το Android επιτρέπει στις εφαρμογές να δημιουργούν αντίγραφα ασφαλείας και να επαναφέρουν τα δεδομένα του χρήστη. Ξεκινώντας από το επίπεδο API 23, το Android εκτελεί αυτόματη δημιουργία αντιγράφων ασφαλείας.

→ Ασφάλεια Εφαρμογών

Η ασφάλεια είναι μια κρίσιμη πτυχή των εφαρμογών Android. Το Android χρησιμοποιεί ασφάλεια βάσει αδειών σε επίπεδο εφαρμογής. Οι άδειες ορίζονται, δηλώνονται και επιβάλλονται μέσω διάφορων στοιχείων-elements και χαρακτηριστικών-attributes του manifest file, για την προστασία του ευαίσθητου συστήματος και των πόρων της εφαρμογής. Οι ευαίσθητοι πόροι του συστήματος, όπως η τοποθεσία, προστατεύονται με προκαθορισμένες άδειες συστήματος, ενώ οι ευαίσθητοι πόροι εφαρμογών προστατεύονται από τον ορισμό των προσαρμοσμένων αδειών στο manifest file μέσω <permission> elements. Για να καταστεί δυνατή η απόκτηση πρόσβασης στο προστατευμένο σύστημα ή τους πόρους της εφαρμογής από μια άλλη εφαρμογή πρέπει να έχουν δηλωθεί αντίστοιχες άδειες στο manifest file μέσω των δηλωτικών στοιχείων <uses-permission> ή <uses-permission-sdk>.

2.4.6 Android Studio

Η πλατφόρμα η οποία χρησιμοποιήθηκε για τη συγγραφή, διόρθωση, εκτέλεση και διαμόρφωση της εφαρμογής είναι το Android Studio. Το Android Studio είναι το επίσημο ολοκληρωμένο περιβάλλον ανάπτυξης (IDE) για την ανάπτυξη εφαρμογών

Android, βασισμένο στο IntelliJ IDEA. Πολλοί λόγοι είναι αυτοί που ωθούν έναν προγραμματιστή εφαρμογών να χρησιμοποιήσει αυτή την πλατφόρμα. Τόσο τα ποικίλα και εύχρηστα εργαλεία επεξεργασίας κώδικα και ανάπτυξης IntelliJ, όσο και οι ακόλουθες λειτουργίες που βελτιώνουν την παραγωγικότητα:

- α. Ευελιξία συστήματος κατασκευής με βάση το Gradle (εργαλείο αυτοματισμού ανοιχτού κώδικα που επικεντρώνεται στην ευελιξία και την απόδοση).
- β. Ταχύτητα και ποικιλία σε χαρακτηριστικά εξομοίωσης
- γ. Ενιαίο περιβάλλον ανάπτυξης εφαρμογών για όλες τις συσκευές Android
- δ. Δυνατότητα εφαρμογής αλλαγών κώδικα και πόρων στην τρέχουσα εφαρμογή χωρίς να εκτελεστεί η εφαρμογή.
- ε. Πρότυπα κώδικα και ολοκλήρωση GitHub για τη δημιουργία κοινών λειτουργιών εφαρμογής και εισαγωγής δείγματος κώδικα.
- στ. Εκτεταμένα εργαλεία και πλαίσια δοκιμών.
- ζ. Εργαλεία Lint για να αποτυπώνουν την απόδοση, τη χρηστικότητα, τη συμβατότητα έκδοσης και άλλα προβλήματα
- η. Υποστήριξη C++ και NDK
- θ. Ενσωματωμένη υποστήριξη για την πλατφόρμα Google Cloud, διευκολύνοντας την ενσωμάτωση του Google Cloud Messaging και του App Engine.

2.4.6.1 Δομή Έργου-Project

Κάθε έργο στο Android Studio περιέχει μία ή περισσότερες ενότητες με αρχεία πηγαίου κώδικα και αρχεία πόρων. Οι τύποι ενότητων περιλαμβάνουν:

- α. Μονάδες εφαρμογών Android
- β. Μονάδες βιβλιοθήκης
- γ. Υπομονάδες μηχανής Google App

Το Android Studio εμφανίζει τα αρχεία του έργου στην προβολή έργου του Android, όπως φαίνεται στην Εικόνα 2, βάση προεπιλογής της πλατφόρμας. Αυτή η προβολή

οργανώνεται από ενότητες για γρήγορη πρόσβαση στα βασικά αρχεία προέλευσης του έργου.

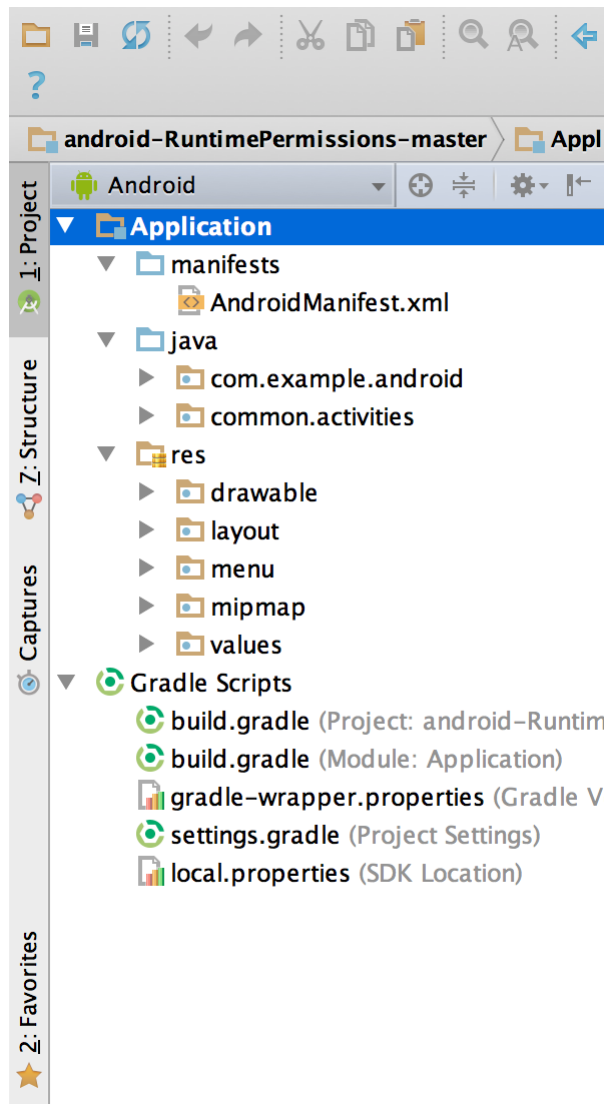
Όλα τα αρχεία δημιουργίας (build files) παρατίθενται στο επίπεδο κάτω από τα Gradle Scripts και κάθε ενότητα εφαρμογών περιέχει τους ακόλουθους φακέλους:

α.manifests: Περιέχει το αρχείο AndroidManifest.xml.

β.java: Περιέχει τα αρχεία πηγαίου κώδικα Java, συμπεριλαμβανομένου του δοκιμαστικού κώδικα JUnit.

γ.res: Περιέχει όλους τους μη κωδικούς πόρους, όπως διατάξεις XML, συμβολοσειρές UI και εικόνες bitmap.

Η δομή έργου Android στο δίσκο διαφέρει από αυτήν τη διευρυμένη αναπαράσταση. Για την προβολή της πραγματικής δομής του αρχείου του έργου, επιλέγεται Project από το αναπτυσσόμενο μενού Project (στην Εικόνα 2, εμφανίζεται ως Android).



Εικόνα 2: Τα αρχεία του project στην προβολή Android.

2.4.6.2 Επισκόπηση projects

Ένα project στο Android Studio περιέχει όλα όσα ορίζουν το χώρο εργασίας για τη δημιουργία μιας εφαρμογής, από τον πηγαίο κώδικα και τα assets, μέχρι τον δοκιμαστικό κώδικα και τη δημιουργία παραμέτρων. Με την έναρξη ενός νέου project στο Android Studio δημιουργείται η απαραίτητη δομή για όλα τα απαραίτητα αρχεία.

2.4.6.3 Modules-Ενότητες

Ένα module είναι μια συλλογή από αρχεία προέλευσης και build ρυθμίσεων που επιτρέπουν την κατάτμηση του project σε διακριτές μονάδες λειτουργικότητας. Το project μπορεί να έχει μία ή περισσότερες ενότητες και κάθε module μπορεί να είναι εξαρτώμενο ενός άλλου module. Η δημιουργία, η διόρθωση και η εκτέλεση κάθε module δύναται να τρέχει ανεξάρτητα.

Πρόσθετες ενότητες είναι χρήσιμες κατά τη δημιουργία βιβλιοθηκών κώδικα στο project και διαφορετικών συνόλων κώδικα και πόρων για διαφορετικούς τύπους συσκευών, όπως τηλέφωνα και άλλες φορητές συσκευές, όταν είναι επιθυμητή η διατήρηση όλων των αρχείων που έχουν εφαρμογή στο ίδιο project και η χρήση από κοινού κάποιων κομματιών κώδικα.

Οι διαφορετικοί τύποι των modules του Android Studio είναι οι παρατιθέμενοι:

α. Module Android εφαρμογής

Περιέχει τον πηγαίο κώδικα της εφαρμογής, τα αρχεία πόρων και τις ρυθμίσεις του επιπέδου εφαρμογής, όπως το αρχείο δημιουργίας σε επίπεδο ενότητας και το αρχείο Manifest του Android. Όταν δημιουργείτε ένα νέο έργο, το προεπιλεγμένο όνομα της λειτουργικής μονάδας είναι «app».

Στο παράθυρο Δημιουργία νέας μονάδας, το Android Studio προσφέρει τους ακόλουθους τύπους ενοτήτων εφαρμογών:

- Module τηλεφώνου & tablet
- Wear OS Module
- Android TV Module
- Glass Module

β. Module δυναμικών στοιχείων

Αντιπροσωπεύει ένα διαμορφωμένο χαρακτηριστικό της εφαρμογής που μπορεί να επωφεληθεί από τη δυναμική παράδοση του Google Play.

γ. Module βιβλιοθήκης

Περιέχει κώδικα που μπορεί να αποτελέσει εξάρτηση σε άλλα modules εφαρμογών ή κομμάτι άλλων projects. Δομικά, ένα module βιβλιοθήκης είναι ίδιο με ένα module εφαρμογής, αλλά όταν δημιουργείται, δημιουργεί ένα αρχείο αρχειοθέτησης κώδικα αντί για ένα APK, επομένως δεν μπορεί να εγκατασταθεί σε μια συσκευή.

Κατηγορίες modules βιβλιοθήκης:

-Βιβλιοθήκη Android: Αυτός ο τύπος βιβλιοθήκης μπορεί να περιέχει όλους τους τύπους αρχείων που υποστηρίζονται σε ένα Android project, συμπεριλαμβανομένου του πηγαίου κώδικα, των πόρων και των αρχείων δήλωσης. Το αποτέλεσμα της κατασκευής είναι ένα αρχείο Android Archive(AAR) το οποίο μπορεί να αποτελέσει εξάρτηση για τα modules εφαρμογών Android.

-Βιβλιοθήκη Java: Αυτός ο τύπος βιβλιοθήκης μπορεί να περιέχει μόνο αρχεία προέλευσης Java. Το αποτέλεσμα της κατασκευής είναι ένα αρχείο Java Archive(JAR) το οποίο μπορεί να αποτελέσει εξάρτηση για τα modules εφαρμογών Android ή άλλα Java projects.

-Google Cloud Module

Περιέχει τον κώδικα υποστήριξης του Google Cloud. Αυτή η ενότητα έχει τον απαιτούμενο κώδικα και εξαρτήσεις για ένα backend της εφαρμογής Java App Engine που χρησιμοποιεί απλά HTTP, Endpoint Cloud και Cloud Messaging για να συνδεθεί με την εφαρμογή.

2.4.6.4 Αρχεία και Κατάλογοι των Android Projects

Η δομή των Android projects συνίσταται από φακέλους και διάφορα αρχεία που συνθέτουν την εφαρμογή και τις λειτουργίες της. Η ιεραρχική δόμησή τους φαίνεται στην Εικόνα 3 και παρακάτω αναφέρονται περιεκτικά οι λειτουργίες που επιτελούν και για τις οποίες συνίστανται.

module-name /

`build/`: Περιέχει εξόδους κατασκευής.

`libs/` :Περιέχει ιδιωτικές βιβλιοθήκες.

`src /` :Περιέχει όλα τα αρχεία κώδικα και πόρους για την ενότητα στους ακόλουθους υποκαταλόγους:

`androidTest /` :Περιέχει κώδικα για δοκιμές οργάνων που εκτελούνται σε μια συσκευή Android.

`main/` :Περιέχει τα "κύρια" αρχεία πηγαίου κώδικα: τον κώδικα Android και τους πόρους που μοιράζονται όλες οι παραλλαγές κατασκευής (τα αρχεία για άλλες παραλλαγές κατασκευής βρίσκονται σε καταλόγους της ίδιας κατηγορίας, όπως το `src/` και `debug/` για τον τύπο δημιουργίας εντοπισμού σφαλμάτων).

`AndroidManifest.xml` :Περιγράφει τη φύση της εφαρμογής και των συστατικών στοιχείων της.

`Java/` : Περιέχει πηγές κώδικα Java.

`jni /` :Περιέχει εγγενή κώδικα χρησιμοποιώντας Java Native Interface (JNI).

`gen /` :Περιέχει τα αρχεία Java που δημιουργούνται από το Android Studio, όπως το αρχείο `R.java` και οι διεπαφές που έχουν δημιουργηθεί από αρχεία AIDL.

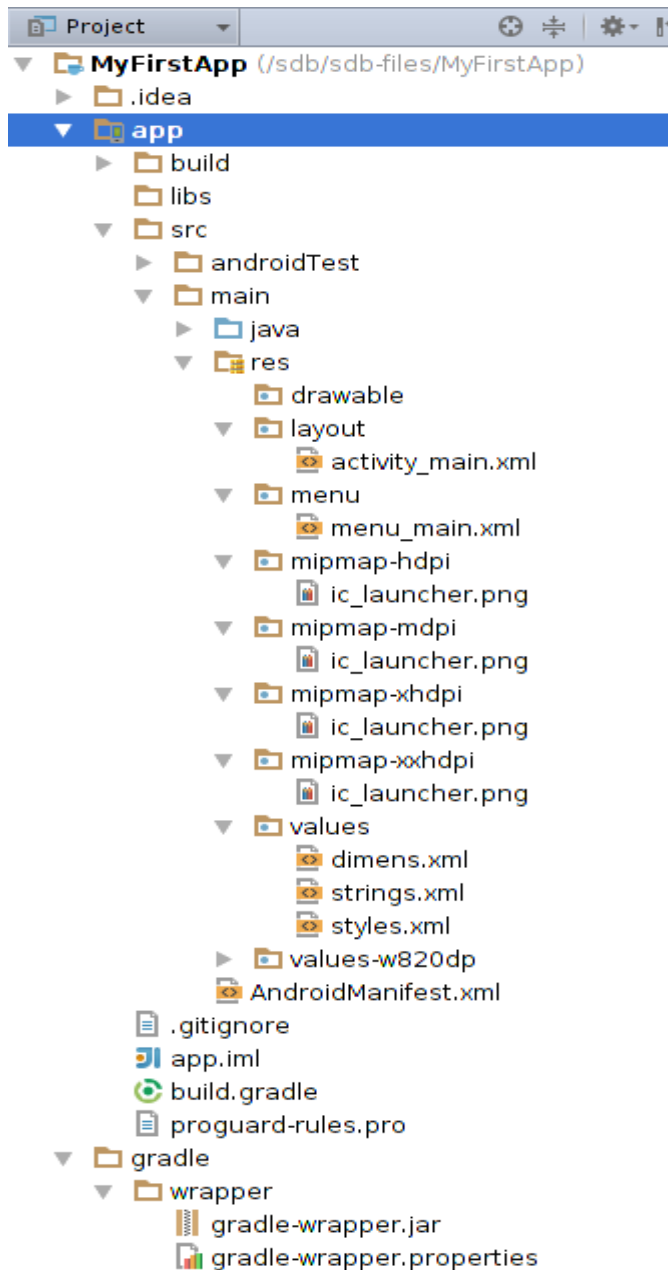
`res /` :Περιέχει πόρους εφαρμογής, όπως αρχεία με δυνατότητα ανανέωσης, αρχεία διάταξης και συμβολοσειρά UI.

`assets/`:Περιέχει αρχείο που θα πρέπει να μετατραπεί σε αρχείο `.apk`.

`test/` :Περιέχει κώδικα για τοπικές δοκιμές που εκτελούνται στον κεντρικό υπολογιστή σας JVM.

`build.gradle (module)` :Αυτό καθορίζει τις συγκεκριμένες διαμορφώσεις κατασκευής μονάδας.

`build.gradle (project)` :Αυτό καθορίζει τη διαμόρφωση κατασκευής που ισχύει για όλα τα modules.



Εικόνα 3: Αρχεία και κατάλογοι στη δομή ενός Android Project.

2.4.6.5 Ρυθμίσεις Δομής Project

Τοποθεσία SDK (Software Development Kit): Ορίζει την τοποθεσία των εκτελεστικών εργαλείων ανάπτυξης που χρησιμοποιεί το project. JDK (Java Development Kit) , Android SDK, που αποτελεί βιβλιοθήκη βασισμένη σε γλώσσα προγραμματισμού C και NDK (Native Development Kit) Android, βασισμένη σε γλώσσα προγραμματισμού C/C++.

Project: Ορίζει την έκδοση και την πρόσθετη εφαρμογή Android για το Gradle και το όνομα της τοποθεσίας αποθήκευσης.

Υπηρεσίες προγραμματιστών: Περιέχει ρυθμίσεις για τα πρόσθετα του Android Studio από την Google ή άλλα τρίτα μέρη.

Ενότητες-Modules: Επεξεργάζεται διαμορφώσεις δημιουργίας συγκεκριμένων παραμέτρων modules, συμπεριλαμβανομένου του στόχου και του ελάχιστου SDK, της υπογραφής της εφαρμογής και των εξαρτήσεων της βιβλιοθήκης.

2.4.6.6 Σύνταξη διάταξης (Layout Editor)

Κατά την επεξεργασία αρχείων διαμόρφωσης XML, το Android Studio παρέχει ένα πρόγραμμα περιήγησης με έναν οπτικό επεξεργαστή drag-and-drop (σύρω-αφήνω) που απλοποιεί τη δημιουργία μιας νέας διάταξης. Ο επεξεργαστής διάταξης σε συνδυασμό με το API ConstraintLayout, επιτρέπει τη γρήγορη επεξεργασία μιας διάταξης που προσαρμόζεται σε διαφορετικά μεγέθη οθόνης, μεταφέροντας τις προβολές στη θέση τους και στη συνέχεια προσθέτοντας τους περιορισμούς της σχεδίασης με μερικά μόνο κλικ.

2.4.6.7 APK Analyzer

Ο αναλυτής APK καθιστά εύκολη την επιθεώρηση των περιεχομένων του APK φακέλου. Αποκαλύπτει το μέγεθος κάθε συστατικού, για τον εντοπισμό τρόπων μείωσης του συνολικού μεγέθους APK. Επιτρέπει επίσης την προεπισκόπηση των πακέτων στοιχείων, την επιθεώρηση των αρχείων DEX για την αντιμετώπιση πολλαπλών προβλημάτων και τη σύγκριση των διαφορών μεταξύ δύο αρχείων APK.

2.4.6.8 Android SDK Platform

Κάθε έκδοση της πλατφόρμας SDK περιλαμβάνει τα ακόλουθα πακέτα:

Το πακέτο πλατφόρμας SDK για Android: Για τη μεταγλώττιση της εφαρμογής για την αντίστοιχη έκδοση.

Αρκετά πακέτα εικόνας συστήματος: Τουλάχιστον ένα από αυτά απαιτείται για την εκτέλεση της έκδοσης στον εξομοιωτή Android.

Κάθε έκδοση πλατφόρμας περιλαμβάνει μια εικόνα συστήματος για κάθε υποστηριζόμενο παράγοντα σχεδιασμού (φορητές συσκευές, Android TV και Android Wear). Κάθε παράγοντας σχεδιασμού μπορεί να προσφέρει παραλλαγές που ταιριάζουν με την αρχιτεκτονική του επεξεργαστή του υπολογιστή (όπως Intel x86 και ARM EABI). Οι εικόνες του συστήματος που φέρουν την ετικέτα API Google περιλαμβάνουν πρόσβαση στις υπηρεσίες Google Play και εκείνες που φέρουν την ετικέτα Google Play περιλαμβάνουν επίσης το Google Play Store.

Το πακέτο «Πηγές για το Android»: Αυτό περιλαμβάνει τα αρχεία προέλευσης για την πλατφόρμα. Το Android Studio ενδέχεται να εμφανίζει γραμμές κώδικα από αυτά τα αρχεία, κατά τη διάρκεια του εντοπισμού σφαλμάτων στην εφαρμογή. Οι εικόνες του συστήματος ενδέχεται να λαμβάνουν ξεχωριστές ενημερώσεις, συνήθως για την επίλυση σφαλμάτων με τον εξομοιωτή [16].

2.4.7 Επισκόπηση Bluetooth

Για την εκτέλεση της παρούσας εργασίας μας ενδιαφέρει σε μεγάλο βαθμό και το παρακάτω χαρακτηριστικό της πλατφόρμας Android.

Η πλατφόρμα Android υποστηρίζει στοίβα δικτύου Bluetooth, η οποία επιτρέπει σε μια συσκευή ασύρματη ανταλλαγή δεδομένων με άλλες συσκευές Bluetooth. Το πλαίσιο εφαρμογής παρέχει πρόσβαση στις λειτουργίες Bluetooth μέσω των APIs Android Bluetooth. Αυτά τα APIs επιτρέπουν σε εφαρμογές να συνδέονται ασύρματα

σε άλλες συσκευές Bluetooth, επιτρέποντας ασύρματες πολυσημιακές και από σημείο σε σημείο λειτουργίες.

Μέσω APIs Bluetooth, μια εφαρμογή Android μπορεί να εκτελέσει τα εξής:

- Σάρωση για άλλες συσκευές Bluetooth

- Ενημέρωση του τοπικού προσαρμογέα Bluetooth για συνδυασμένες συσκευές Bluetooth

- Απόκτηση καναλιών εξυπηρετητών (servers) μέσω RFCOMM πρωτόκολλου

- Σύνδεση με άλλες συσκευές μέσω της ανεύρεσης υπηρεσίας

- Μεταφορά δεδομένων προς και από άλλες συσκευές

- Διαχείριση πολλαπλών συνδέσεων [18]

Κεφάλαιο 3. Μεθοδολογία

3.1 Εισαγωγικά

Ο σκοπός της εργασίας είναι η δημιουργία μιας διάταξης για τη λήψη μετρήσεων από έναν αισθητήρα μέτρησης μαγνητικού πεδίου, Hall. Για την επίτευξη αυτού του σκοπού αναπτύχθηκε μία ηλεκτρονική διάταξη με έναν αισθητήρα Hall συνδεδεμένο με μια πλακέτα Arduino Uno. Κάθε φορά που ένα μαγνητικό υλικό ανιχνεύεται από τον αισθητήρα η τιμή που παράγει μεταβάλλεται. Το ίδιο φαινόμενο συμβαίνει και με την παρεμβολή ενός ελάσματος ή διακένου μεταξύ του αισθητήρα και ενός μαγνήτη. Για να καταστούν, ωστόσο, αντιληπτές οι ως άνω μεταβολές θα πρέπει να υπάρχει ένας τρόπος προβολής. Ο πιο απτός τρόπος προβολής είναι μια οθόνη (monitor), η οποία σχεδιάστηκε και υλοποιήθηκε ως εφαρμογή για κινητά τηλέφωνα σε λογισμικό Android. Για την μεταφορά των μετρήσεων από τον αισθητήρα στο κινητό χρησιμοποιήθηκε μια συσκευή Bluetooth, HC-05. Η συσκευή αυτή συνδέθηκε σε ηλεκτρονική διάταξη με την πλακέτα του Arduino, ώστε να λαμβάνει ενσύρματα τα δεδομένα και ύστερα να τα μεταδίδει στην εφαρμογή.

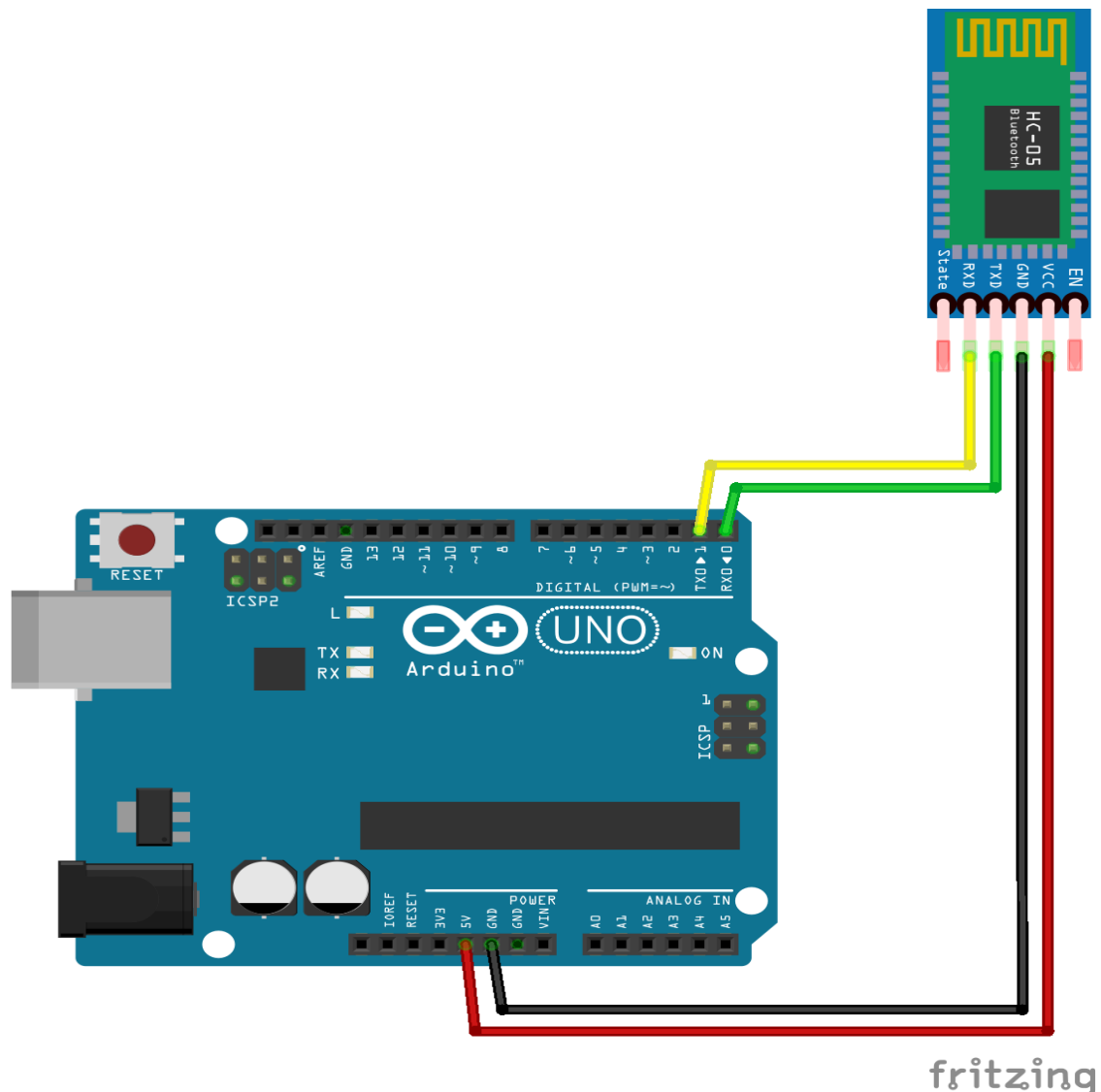
3.2 HC-05 και διασύνδεση με Arduino

Για να συνδεθεί το HC-05 με το Arduino χρειάζεται να στέλνει και να λαμβάνει δεδομένα αξιοποιώντας τα pins 0 και 1 του Arduino τα οποία έχουν την σήμανση RX και TX αντίστοιχα στην πλακέτα. Το pin 0 (RX) του Arduino δέχεται (Receive) δεδομένα από την μονάδα Bluetooth, ενώ το pin 1 (TX) του Arduino στέλνει (Transmit) δεδομένα στην μονάδα Bluetooth. Η σύνδεση έχει ως εξής:

- HC-05 VCC → Arduino 5V
- HC-05 GND → Arduino GND
- HC-05 TX → Arduino RX (pin 0)

- HC-05 RX → Arduino TX (pin 1)

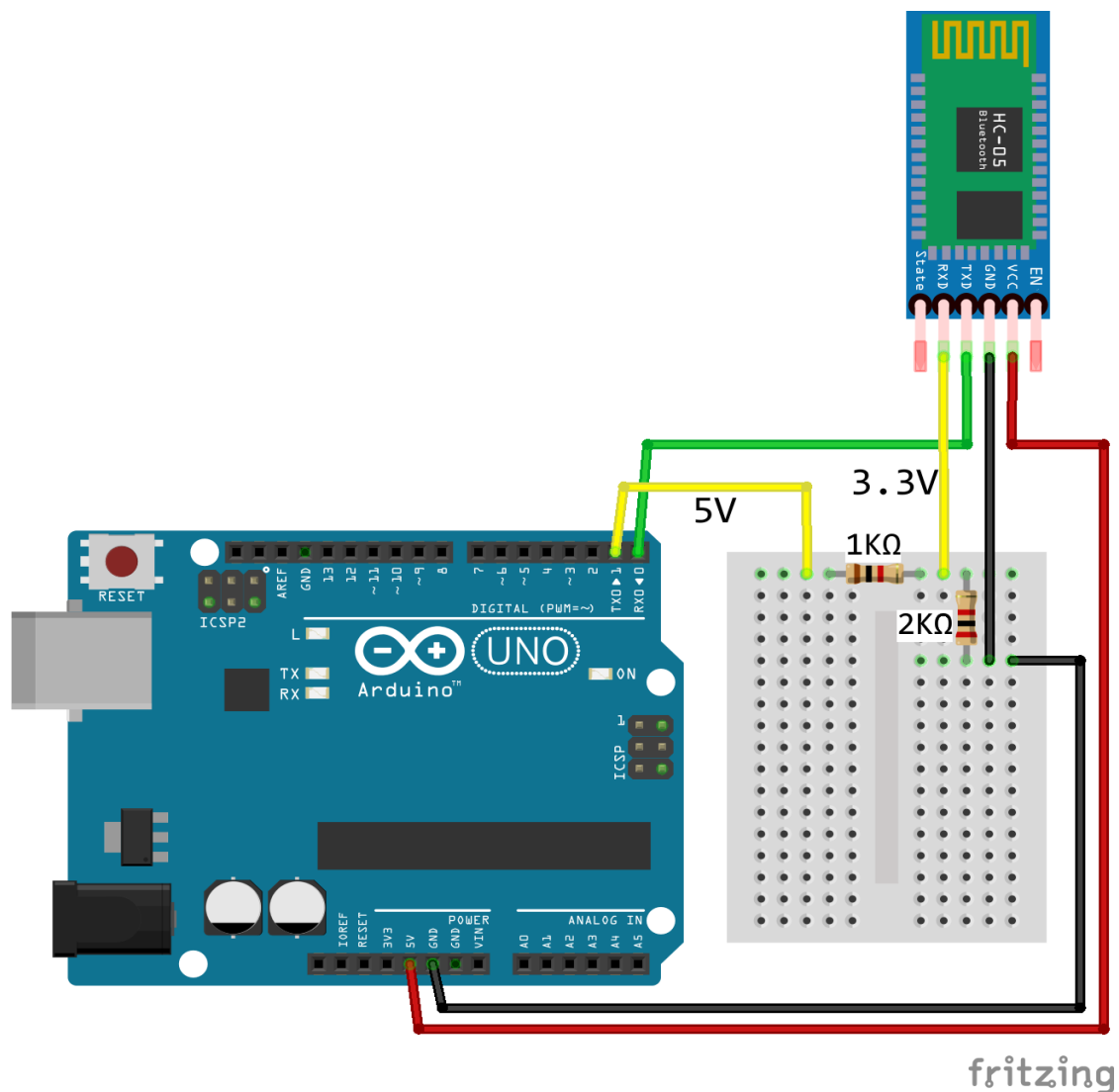
Το διάγραμμα που παρατίθεται στη συνέχεια, απεικονίζει τη σύνδεση υλικού μεταξύ της μονάδας Bluetooth HC-05 και του Arduino UNO:



Σχήμα 6: Σχεδιάγραμμα διασύνδεσης του Bluetooth HC-05 με το Arduino [20]

Για τη διαδικασία αποστολής δεδομένων στο Arduino μέσω Bluetooth ισχύουν κάποιες συνθήκες που αν δεν ληφθούν υπόψη μπορούν να προκληθούν σφάλματα στο υλικό. Ένα θέμα που προκύπτει είναι ότι η μονάδα HC-05 λειτουργεί στα 3.3V σε αντίθεση με το Arduino που λειτουργεί στα 5V. Η μονάδα HC-05, ωστόσο, έχει εσωτερικό ρυθμιστή στο VCC στα 3.3V και γι' αυτό μπορεί συνδεθεί με τάση 5V. Ωστόσο συνίσταται η σύνδεση στον ακροδέκτη τάσης 3.3V, καθώς η παροχή 5V στη

μονάδα ενδέχεται να προκαλέσει βλάβη. Η σύνδεση του TX pin του HC-05 με το RX pin του Arduino είναι ασφαλής από τη στιγμή που το ρεύμα οδεύει από το HC-05 προς το Arduino. Το RX pin όμως του HC-05 απαιτεί 3.3V ως είσοδο, ενώ δέχεται 5V από το Arduino. Προκειμένου να αποφευχθεί η ζημιά της μονάδας και να λειτουργήσει σωστά, συνίσταται η χρήση κυκλώματος διαίρεσης τάσης (5v έως 3.3v) μεταξύ του ακροδέκτη του arduino TX και του ακροδέκτη RX της μονάδας, όπως απεικονίζεται παρακάτω:



Σχήμα 7: Σχεδιάγραμμα διασύνδεσης του Bluetooth HC-05 με το Arduino με χρήση κυκλώματος διαίρεσης τάσης [20]

Όταν συνδέονται ο κύριος και ο υποτελής (master and slave), τα μπλε και κόκκινα

φωτάκια-LED στο πίνακα αναβοσβήνουν κάθε 2 δευτερόλεπτα. Εάν δεν είναι συνδεδεμένες, μόνο το μπλε αναβοσβήνει κάθε 2 δευτερόλεπτα.

3.3 Κώδικας Arduino

Για τη σύνδεση του HC-05 με το Arduino και τη μεταφορά δεδομένων, μεταφορτώθηκε ο παρακάτω κώδικας στο Arduino:

```
#include <SoftwareSerial.h> // βιβλιοθήκη για τη σειριακή επικοινωνία
του λογισμικού
SoftwareSerial BTSerial(10, 11); // σύνδεση του TX pin του Bluetooth
στο pin 10 του Arduino και το RX pin του Bluetooth στο pin 11 του
Arduino
void setup() //μία συνάρτηση που τρέχει μία φορά στην αρχή του
προγράμματος η οποία αρχικοποιεί τις ρυθμίσεις.
{
  pinMode(9, OUTPUT); // αυτό το pin θα θέσει το pin του HC-05 34 (key
pin) στην τιμή HIGH για να φέρει τη μονάδα σε λειτουργία AT, ώστε να
δέχεται εντολές.
  digitalWrite(9, HIGH);
  Serial.begin(9600); // διαμορφώνει το ρυθμό σειριακής μετάδοσης του
λογισμικού στα 9600
  Serial.println("Enter AT commands:");
  BTSerial.begin(38400); // καθορισμός ταχύτητας του HC-05 για να
εισέλθει η μονάδα στη λειτουργία εντολών AT
}
void loop() //μία συνάρτηση που καλείται συνέχεια μέχρι η πλακέτα να
απενεργοποιηθεί.
{
  // διάβασμα από το HC-05 και αποστολή στη σειριακή οθόνη (Serial
Monitor) του Arduino
  if (BTSerial.available())
    Serial.write(BTSerial.read());
  // διάβασμα από το Arduino Serial Monitor και αποστολή στο HC-05
  if (Serial.available())
    BTSerial.write(Serial.read());
}
```

Η μονάδα είναι πλέον ανιχνεύσιμη από άλλες συσκευές, όπως το κινητό τηλέφωνο που μας ενδιαφέρει στην παρούσα εργασία με όνομα επικοινωνίας HC-05, ο κωδικός πρόσβασης-σύζευξης είναι 1234 ή 0000 και ο ρυθμός μετάδοσης (baud rate) είναι 9600 από προεπιλογή (μπορεί να φτάσει τα 1382400) [21].

Πρόβλημα που προέκυψε κατά τη μεταφόρτωση του κώδικα στο Arduino ήταν το παρακάτω:

```
Προέκυψε ένα σφάλμα κατά το ανέβασμα του σχεδίου
Το σχέδιο χρησιμοποιεί 3652 bytes (11%) του χώρου αποθήκευσης του προγράμματος.
Οι καθολικές μεταβλητές χρησιμοποιούν 457 bytes (22%) δυναμικής μνήμης, αφήνοντας
avrdude: stk500_getsync() attempt 1 of 10: not in sync: resp=0x00
avrdude: stk500_getsync() attempt 2 of 10: not in sync: resp=0x00
avrdude: stk500_getsync() attempt 3 of 10: not in sync: resp=0x00
avrdude: stk500_getsync() attempt 4 of 10: not in sync: resp=0x00
avrdude: stk500_getsync() attempt 5 of 10: not in sync: resp=0x00
avrdude: stk500_getsync() attempt 6 of 10: not in sync: resp=0x00
avrdude: stk500_getsync() attempt 7 of 10: not in sync: resp=0x00
avrdude: stk500_getsync() attempt 8 of 10: not in sync: resp=0x00
avrdude: stk500_getsync() attempt 9 of 10: not in sync: resp=0x00
avrdude: stk500_getsync() attempt 10 of 10: not in sync: resp=0x00
Προέκυψε ένα σφάλμα κατά το ανέβασμα του σχεδίου
```

Εικόνα 4: Σφάλμα Μεταφόρτωσης Κώδικα

Για την αντιμετώπισή του πρέπει να αποσυνδεθεί το HC-05 (το καλώδιο στο pin VCC, ή τα καλώδια στο RX και TX pins) από το Arduino κατά την φάση ανεβάσματος του προγράμματος, για να μην είναι δεσμευμένη η σειριακή θύρα του Arduino από το Bluetooth, ώστε να μπορέσει η θύρα USB να χρησιμοποιήσει τα RX και TX pins [15].

Αυτή η διαδικασία όμως είναι ανέφικτη όταν για το κύκλωμα αυτό έχει κατασκευαστεί επένδυση και η πρόσβαση στα καλώδια είναι αδύνατη. Τη λύση στο προκύπτον ζήτημα έρχεται να δώσει ο παρακάτω κώδικας που αφού τον μεταφορτώσουμε στην πλακέτα επιβάλλει τη σύνδεση του TX pin του Bluetooth στο pin 10 του Arduino και το RX pin του Bluetooth στο pin 11 του Arduino. Με αυτό τον τρόπο, από εδώ και μετά δεν είναι αναγκαία η αποσύνδεση των καλωδίων από τα pin TR και RX του Arduino για να αν ανέβει ένα Arduino Sketch στο λογισμικό της πλακέτας.

➔ Αποστολή εντολών AT στη μονάδα Bluetooth HC-05 και αλλαγή των ρυθμίσεών της

Πατώντας και κρατώντας πατημένο το κουμπί που υπάρχει στην κάτω δεξιά πλευρά του HC-05, η μονάδα μεταβαίνει σε λειτουργία AT, από τη λειτουργία επικοινωνίας που βρίσκεται βάσει εργοστασιακών ρυθμίσεων. Η προεπιλεγμένη τιμή του ρυθμού μετάδοσης για να εισέλθει η μονάδα στη λειτουργία εντολών είναι 38400. Μετά τη μεταφόρτωση του κώδικα, οι εντολές μπορούν να υποστούν επεξεργασία χρησιμοποιώντας τη Σειριακή Παρακολούθηση (Serial Monitor).Στέλνοντας μια εντολή-COMMAND στη μονάδα, λαμβάνεται μια απόκριση-RESPONSE [22].

Ο αισθητήρας Hall που χρησιμοποιήθηκε αποστέλλει στο Arduino την τιμή της τάσης Hall (Voltage) η οποία βαθμονομήθηκε σύμφωνα με το offset του αισθητήρα, την τιμή του μαγνητικού πεδίου (Magnetic Field) που ανιχνεύει και μία μεταβλητή τιμή (Value) στο εύρος 0-1024 (0-1 byte) που αλλάζει ανά αισθητήρα και σύμφωνα με την αλλαγή στο περιβάλλον-μέτρηση. Για τη λήψη των δεδομένων που λαμβάνει, λοιπόν, ο αισθητήρας απ' το περιβάλλον του, συνετάχθει κώδικας-sketch στο πρόγραμμα συγγραφής του Arduino στον υπολογιστή και μεταφορτώθηκε στην πλακέτα του Arduino:

```
int pin = A0;           // pin εισόδου αισθητήρα Hall
float vcc = 5000;       // τάση τροφοδοσίας αισθητήρα Hall
float cal = 0.014;      // τιμή βαθμονόμησης (mV/uT)
float offset = 520;     // offset αισθητήρα Hall
int value = 0;          //μεταβλητή τιμής στο εύρος 0-1023
float voltage = 0.0;    //μεταβλητή τάσης στο εύρος 0-Vcc
float field = 0.0;      //μεταβλητή τιμής πεδίου

void setup() {
  pinMode(pin, INPUT);
  Serial.begin(9600); //έναρξη σειριακής επικοινωνίας
}

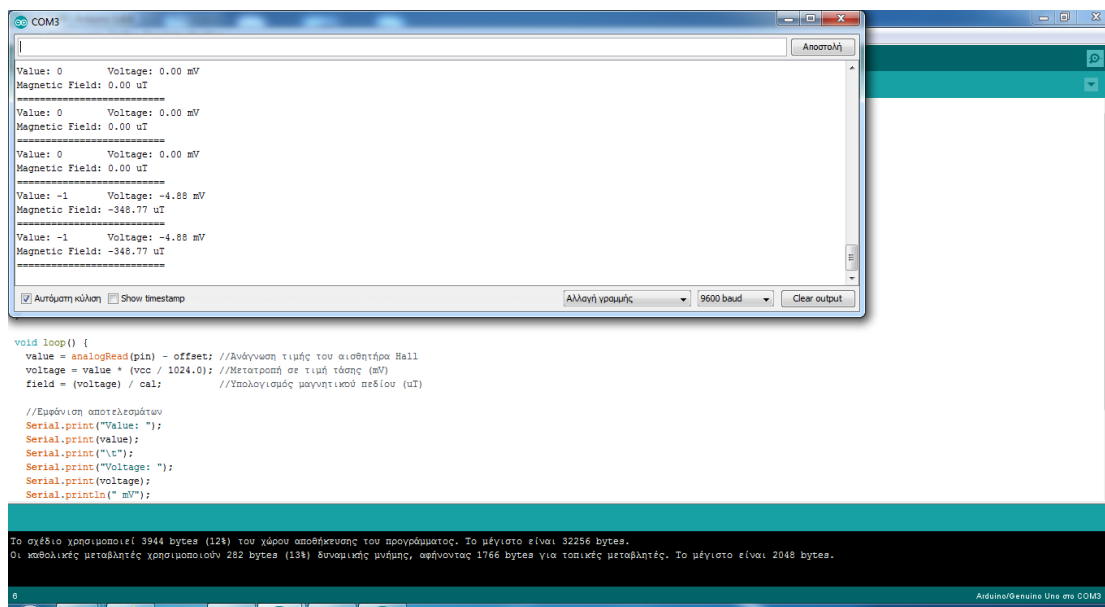
void loop() {
  value = analogRead(pin) - offset; //ανάγνωση τιμής του αισθητήρα
Hall
  voltage = value * (vcc / 1024.0); //μετατροπή σε τιμή τάσης (mV)
  field = (voltage) / cal;          //υπολογισμός μαγνητικού πεδίου (uT)
  //εμφάνιση αποτελεσμάτων
  Serial.print("Value: ");
  Serial.print(value);
  Serial.print("\t");
  Serial.print("Voltage: ");
  Serial.print(voltage);
  Serial.println(" mV");
```

```

Serial.print("Magnetic Field: ");
Serial.print(field);
Serial.println(" uT");
Serial.println("=====");
//λήψη νέας τιμής κάθε 1sec
delay(1000);
}

```

Στην οθόνη σειριακής παρακολούθησης εμφανίζεται η έξοδος του προγράμματος. Ανά δευτερόλεπτο έχει ρυθμιστεί να λαμβάνεται μέτρηση και να εμφανίζονται οι αντίστοιχες τιμές. Τις ίδιες τιμές και αντίστοιχη έξοδο θα λαμβάνεται και θα εμφανίζεται στην οθόνη της εφαρμογής του κινητού τηλεφώνου.



Εικόνα 5: Serial Monitor του Arduino sketch με τις μετρήσεις του αισθητήρα Hall

3.4 Κώδικας Android Εφαρμογής

3.4.1 Αρχεία .java

3.4.1.1 MainActivity.java

```
package com.example.sensormonitor;

import android.app.Activity;
//import android.support.v7.app.AppCompatActivity;
import androidx.appcompat.app.AppCompatActivity;
import android.os.Bundle;
import android.bluetooth.BluetoothAdapter;
import android.bluetooth.BluetoothDevice;
import android.content.Context;
import android.content.Intent;
import android.content.SharedPreferences;
import android.content.pm.ActivityInfo;
import android.graphics.Color;
import android.os.AsyncTask;
import android.preference.PreferenceManager;
import android.util.Log;
import android.view.LayoutInflater;
import android.view.Menu;
import android.view.MenuItem;
import android.view.View;
import android.view.ViewGroup;
import android.widget.AdapterView;
import android.widget.AdapterView.OnItemClickListener;
import android.widget.ArrayAdapter;
import android.widget.Button;
import android.widget.ListView;
import android.widget.TextView;
import android.widget.Toast;

import com.example.sensormonitor.R;

import java.util.ArrayList;
import java.util.List;
import java.util.Set;
```

```

import java.util.UUID;

public class MainActivity extends Activity {
    private Button search;
    private Button connect;
    private ListView;
    private BluetoothAdapter mBTAdapter;
    private static final int BT_ENABLE_REQUEST = 10; // κώδικας
ενεργοποίησης του bluetooth
    private static final int SETTINGS = 20;
    private UUID mDeviceUUID = UUID.fromString("00001101-0000-1000-
8000-00805F9B34FB");
    private int mBufferSize = 50000; //default
    public static final String DEVICE_EXTRA =
"com.example.sensormonitor.SOCKET";
    public static final String DEVICE_UUID =
"com.example.sensormonitor.uuid";
    private static final String DEVICE_LIST =
"com.example.sensormonitor.devicelist";
    private static final String DEVICE_LIST_SELECTED =
"com.example.sensormonitor.devicelistselected";
    public static final String BUFFER_SIZE =
"com.example.sensormonitor.buffersize";
    private static final String TAG = "BlueTest5-MainActivity";

    @Override
    protected void onCreate(Bundle savedInstanceState) { //έναρξη
δραστηριότητας
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

//σύνδεση κουμπιών με την activity_main.xml
        search = (Button) findViewById(R.id.search);
        connect = (Button) findViewById(R.id.connect);
//ορισμός τύπου προβολής διεπαφής
        listView = (ListView) findViewById(R.id.listview);
//δημιουργία λίστας συσκευών bluetooth
        if (savedInstanceState != null) {
            ArrayList<BluetoothDevice> list =

```

```

savedInstanceState.getParcelableArrayList(DEVICE_LIST);
        if (list != null) {
            initList(list);
            MyAdapter adapter = (MyAdapter)
listView.getAdapter();
            int selectedIndex =
savedInstanceState.getInt(DEVICE_LIST_SELECTED);
            if (selectedIndex != -1) {
                adapter.setSelectedIndex(selectedIndex);
                connect.setEnabled(true);
            }
        } else {
            initList(new ArrayList<BluetoothDevice>());
        }

    } else {
        initList(new ArrayList<BluetoothDevice>());
    }

    search.setOnClickListener(new View.OnClickListener() {
//μετάβαση από το πάτημα του κουμπιού search
        @Override
        public void onClick(View arg0) {
            mBTAdapter = BluetoothAdapter.getDefaultAdapter();

            if (mBTAdapter == null) {
                Toast.makeText(getApplicationContext(),
"Bluetooth not found", Toast.LENGTH_SHORT).show();
            } else if (!mBTAdapter.isEnabled()) {
                Intent enableBT = new
Intent(BluetoothAdapter.ACTION_REQUEST_ENABLE);
                startActivityForResult(enableBT,
BT_ENABLE_REQUEST);
            } else {
                new SearchDevices().execute();
            }
        }
    });

    connect.setOnClickListener(new View.OnClickListener() {
//μετάβαση από το πάτημα του κουμπιού connect
        @Override

```

```

        public void onClick(View arg0) {
            BluetoothDevice device = ((MyAdapter)
(listView.getAdapter())).getSelectedItem();
            Intent intent = new Intent(getApplicationContext(),
MonitoringScreen.class);
            intent.putExtra(DEVICE_EXTRA, device);
            intent.putExtra(DEVICE_UUID, mDeviceUUID.toString());
            intent.putExtra(BUFFER_SIZE, mBufferSize);
            startActivity(intent);
        }
    });
}

protected void onPause() {
// TODO Auto-generated method stub
    super.onPause();
}

@Override
protected void onStop() {
// TODO Auto-generated method stub
    super.onStop();
}
//κώδικας ενεργοποίησης bluetooth
@Override
protected void onActivityResult(int requestCode, int resultCode,
Intent data) {
    switch (requestCode) {
        case BT_ENABLE_REQUEST:
            if (resultCode == RESULT_OK) { //μήνυμα ορθής
ενεργοποίησης
                msg("Bluetooth Enabled successfully");
                new SearchDevices().execute();
            } else { //μήνυμα μη ενεργοποίησης
                msg("Bluetooth couldn't be enabled");
            }

            break;
        case SETTINGS: //σε περίπτωση που οι ρυθμίσεις έχουν
ανανεωθεί
            SharedPreferences prefs =

```

```

PreferenceManager.getDefaultSharedPreferences(this); //σύνδεση με
συνάρτηση PreferencesActivity.java
        String uuid = prefs.getString("prefUuid", "Null");
        mDeviceUUID = UUID.fromString(uuid);
        Log.d(TAG, "UUID: " + uuid);
        String bufSize = prefs.getString("prefTextBuffer",
"Null");

        mBufferSize = Integer.parseInt(bufSize);
//ορισμός προσανατολισμού οθόνης
        String orientation =
prefs.getString("prefOrientation", "Null");
        Log.d(TAG, "Orientation: " + orientation);
        if (orientation.equals("Landscape")) {

setRequestedOrientation(ActivityInfo.SCREEN_ORIENTATION_LANDSCAPE);
        } else if (orientation.equals("Portrait")) {

setRequestedOrientation(ActivityInfo.SCREEN_ORIENTATION_PORTRAIT);
        } else if (orientation.equals("Auto")) {

setRequestedOrientation(ActivityInfo.SCREEN_ORIENTATION_FULL_SENSOR);
        }
        break;
        default:
        break;
    }
    super.onActivityResult(requestCode, resultCode, data);
}

/**
 * Quick way to call the Toast
 */
private void msg(String str) {
    Toast.makeText(getApplicationContext(), str,
Toast.LENGTH_SHORT).show();
}

/**
 * Initialize the List adapter
 * @param objects
 */

```

```

    private void initList(List<BluetoothDevice> objects) {
        final MyAdapter adapter = new
MyAdapter(getApplicationContext(), R.layout.list_item,
R.id.lstContent, objects);
        listView.setAdapter(adapter);
        listView.setOnItemClickListener(new
AdapterView.OnItemClickListener() {

            @Override
            public void onItemClick(AdapterView<?> parent, View view,
int position, long id) {
                adapter.setSelectedIndex(position);
                connect.setEnabled(true);
            }
        });
    }

    /**
     * Δεν γίνεται σάρωση για συσκευές που έχουν αντιστοιχιστεί.
     * Συσκευές που έχουν αντιστοιχιστεί μέσω Ρυθμίσεων-> Bluetooth
     * θα εμφανιστούν με την παρακάτω κλάση.
     */
    private class SearchDevices extends AsyncTask<Void, Void,
List<BluetoothDevice>> {

        @Override
        protected List<BluetoothDevice> doInBackground(Void...
params) {
            Set<BluetoothDevice> pairedDevices =
mBTAdapter.getBondedDevices();
            List<BluetoothDevice> listDevices = new
ArrayList<BluetoothDevice>();
            for (BluetoothDevice device : pairedDevices) {
                listDevices.add(device);
            }
            return listDevices;
        }

        @Override

```

```

        protected void onPostExecute(List<BluetoothDevice>
listDevices) {
            super.onPostExecute(listDevices);
            if (listDevices.size() > 0) {
                MyAdapter adapter = (MyAdapter)
listView.getAdapter();
                adapter.replaceItems(listDevices);
            } else {
                msg("No paired devices found, please pair your serial
BT device and try again");
            }
        }

    }

    /**
     * Κώδικας προσαρμογέα για να εμφανίζονται οι ανιχνευόμενες
     * συσκευές στη λίστα.
     */
    private class MyAdapter extends ArrayAdapter<BluetoothDevice> {
        private int selectedIndex;
        private Context context;
        private int selectedColor = Color.parseColor("#abcdef");
        private List<BluetoothDevice> myList;

        public MyAdapter(Context ctx, int resource, int
textViewResourceId, List<BluetoothDevice> objects) {
            super(ctx, resource, textViewResourceId, objects);
            context = ctx;
            myList = objects;
            selectedIndex = -1;
        }

        public void setSelectedIndex(int position) {
            selectedIndex = position;
            notifyDataSetChanged();
        }

        public BluetoothDevice getSelectedItem() {
            return myList.get(selectedIndex);
        }
    }

```

```

    }

    @Override
    public int getCount() {
        return myList.size();
    }

    @Override
    public BluetoothDevice getItem(int position) {
        return myList.get(position);
    }

    @Override
    public long getItemId(int position) {
        return position;
    }

    private class ViewHolder {
        TextView tv;
    }

    public void replaceItems(List<BluetoothDevice> list) {
        myList = list;
        notifyDataSetChanged();
    }

    public List<BluetoothDevice> getEntireList() {
        return myList;
    }

    @Override
    public View getView(int position, View convertView, ViewGroup
parent) {
        View vi = convertView;
        ViewHolder holder;
        if (convertView == null) {
            vi =
LayoutInflater.from(context).inflate(R.layout.list_item, null);
            holder = new ViewHolder();

            holder.tv = (TextView)

```

```

vi.findViewById(R.id.lstContent);

        vi.setTag(holder);
    } else {
        holder = (ViewHolder) vi.getTag();
    }

    if (selectedIndex != -1 && position == selectedIndex) {
        holder.tv.setBackgroundColor(selectedColor);
    } else {
        holder.tv.setBackgroundColor(Color.WHITE);
    }
    BluetoothDevice device = myList.get(position);
    holder.tv.setText(device.getName() + "\n " +
device.getAddress());

        return vi;
    }

}

@Override
public boolean onCreateOptionsMenu(Menu menu) {
// Ανάπτυξη του μενού με την εμφάνιση των νέων στοιχείων
    //getMenuInflater().inflate(R.menu.homescreen, menu);
    return true;
}

@Override
public boolean onOptionsItemSelected(MenuItem item) {
    switch (item.getItemId()) {
        case R.id.action_settings:
            Intent intent = new Intent(MainActivity.this,
PreferencesActivity.class);
            startActivityForResult(intent, SETTINGS);
            break;
        }
    return super.onOptionsItemSelected(item);
}
}

```

3.4.1.2 MonitoringScreen.java

```
package com.example.sensormonitor;

import java.io.IOException;
import java.io.InputStream;
import java.util.UUID;
import android.app.Activity;
import android.app.ProgressDialog;
import android.bluetooth.BluetoothAdapter;
import android.bluetooth.BluetoothDevice;
import android.bluetooth.BluetoothSocket;
import android.content.Intent;
import android.os.AsyncTask;
import android.os.Bundle;
import android.text.method.ScrollingMovementMethod;
import android.util.Log;
import android.view.View;
import android.view.View.OnClickListener;
import android.widget.Button;
import android.widget.CheckBox;
import android.widget.EditText;
import android.widget.ScrollView;
import android.widget.TextView;
import android.widget.Toast;

import com.example.sensormonitor.R;

public class MonitoringScreen extends Activity {

    private static final String TAG = "BlueTest5-MainActivity";
    private int mMaxChars = 50000; //Default

    private UUID mDeviceUUID;

    private BluetoothSocket mBTSocket;

    private ReadInput mReadThread = null;
```

```

private boolean mIsUserInitiatedDisconnect = false;

// οι σύνδεσμοι-εντολές ελέγχου των στοιχείων της διεπαφής
activity_monitoring_screen.xml

private TextView mTxtReceive;

private Button mBtnClearInput;

private ScrollView scrollView;

private CheckBox chkScroll;

private CheckBox chkReceiveText;

private boolean mIsBluetoothConnected = false;
private BluetoothDevice mDevice;

private ProgressDialog progressDialog;

@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_monitoring_screen);
    ActivityHelper.initialize(this);
    Intent intent = getIntent();
    Bundle b = intent.getExtras();
    mDevice = b.getParcelable(MainActivity.DEVICE_EXTRA);
    mDeviceUUID =
UUID.fromString(b.getString(MainActivity.DEVICE_UUID));
    mMaxChars = b.getInt(MainActivity.BUFFER_SIZE);
    Log.d(TAG, "Ready");
    mTxtReceive = (TextView) findViewById(R.id.txtReceive);

// μέθοδοι βιβλιοθήκης που παρέχουν πρόσβαση σε οπτικές προβολές
    chkScroll = (CheckBox) findViewById(R.id.chkScroll);
    chkReceiveText = (CheckBox)
findViewById(R.id.chkReceiveText);
    scrollView = (ScrollView) findViewById(R.id.viewScroll);
    mBtnClearInput = (Button) findViewById(R.id.btnClearInput);
    mTxtReceive.setMovementMethod(new ScrollingMovementMethod());
    mBtnClearInput.setOnClickListener(new OnClickListener() {

```

```
        @Override
        public void onClick(View arg0) {
            mTxtReceive.setText("");
        }
    });
}

private class ReadInput implements Runnable {

    private boolean bStop = false;

    private Thread t;

    public ReadInput() {

        t = new Thread(this, "Input Thread");

        t.start();

    }

    public boolean isRunning() {

        return t.isAlive();

    }

    @Override

    public void run() {

        InputStream inputStream;

        try {

            inputStream = mBTSocket.getInputStream();

            while (!bStop) {

                byte[] buffer = new byte[256];

                if (inputStream.available() > 0) {
```

```

        inputStream.read(buffer);

        int i = 0;

        /*
         * Αυτό απαιτείται επειδή το νέο String
         (buffer) λαμβάνει ολόκληρο το buffer, δηλαδή 256 χαρακτήρες στο
         Android (http://stackoverflow.com/a/8843462/1287554)
         */

        for (i = 0; i < buffer.length && buffer[i] !=
0; i++) {

        }

        final String strInput = new String(buffer, 0,
i);

        /*
         * Αν η επιλογή είναι true-checked θα γίνει
         λήψη κειμένου, αν είναι false-unchecked θα σταματήσει.
         */

        if (chkReceiveText.isChecked()) {

            mTxtReceive.post(new Runnable() {

                @Override

                public void run() {

                    mTxtReceive.append(strInput);

                    int txtLength =
mTxtReceive.getEditableText().length();

                    if (txtLength > mMaxChars) {

mTxtReceive.getEditableText().delete(0, txtLength - mMaxChars);

                    }

```

```

        if (chkScroll.isChecked()) { //
Scroll οθόνης μόνο αν είναι αληθής η επιλογή

        scrollView.post(new
Runnable() { // http://stackoverflow.com/a/4612082/1287554

            @Override

            public void run() {

scrollView.fullScroll(View.FOCUS_DOWN);

            }

        });

    }

});

}

}

Thread.sleep(500); //χρόνος αδράνειας του νήματος
}

} catch (IOException e) {
// TODO Auto-generated catch block
    e.printStackTrace();
} catch (InterruptedException e) {
// TODO Auto-generated catch block
    e.printStackTrace();
}

}

public void stop() {
    bStop = true;
}

}

private class DisConnectBT extends AsyncTask<Void, Void, Void> {

    @Override

    protected void onPreExecute() {
    }

    @Override

```

```

protected Void doInBackground(Void... params) {

    if (mReadThread != null) {

        mReadThread.stop();

        while (mReadThread.isRunning())

            ; // αναμονή μέχρι τη διακοπή

        mReadThread = null;

    }

    try {

        mBTSocket.close();

    } catch (IOException e) {

// TODO Auto-generated catch block

        e.printStackTrace();

    }

    return null;

}

@Override

protected void onPostExecute(Void result) {

    super.onPostExecute(result);

    mIsBluetoothConnected = false;

    if (mIsUserInitiatedDisconnect) {

        finish();

    }

}

}

```

```

private void msg(String s) {

    Toast.makeText(getApplicationContext(), s,
Toast.LENGTH_SHORT).show();

}

@Override

protected void onPause() {//όταν γίνεται μετάβαση της
δραστηριότητας στο παρασκήνιο

    if (mBTSocket != null && mIsBluetoothConnected) {

        new DisconnectBT().execute();//αποσύνδεση με το bluetooth

    }

    Log.d(TAG, "Paused");

    super.onPause();

}

@Override

protected void onResume() {//όταν η δραστηριότητα αλληλοεπιδρά με
το χρήστη

    if (mBTSocket == null || !mIsBluetoothConnected) {

        new ConnectBT().execute();//σύνδεση με το bluetooth

    }

    Log.d(TAG, "Resumed");

    super.onResume();

}

@Override

protected void onStop() {//όταν σταματά η δραστηριότητα να είναι
εμφανής στο χρήστη

    Log.d(TAG, "Stopped");

```

```

        super.onStop();
    }

    @Override

    protected void onSaveInstanceState(Bundle outState) {
// TODO Auto-generated method stub=

        super.onSaveInstanceState(outState);
    }

    private class ConnectBT extends AsyncTask<Void, Void, Void> {

        private boolean mConnectSuccessful = true;

        @Override

        protected void onPreExecute() {

            progressDialog =
ProgressDialog.show(MonitoringScreen.this, "Hold on",
"Connecting");// http://stackoverflow.com/a/11130220/1287554
        }

        @Override

        protected Void doInBackground(Void... devices) {

            try {

                if (mBTSocket == null || !mIsBluetoothConnected) {

                    mBTSocket =
mDevice.createInsecureRfcommSocketToServiceRecord(mDeviceUUID);

BluetoothAdapter.getDefaultAdapter().cancelDiscovery();

                    mBTSocket.connect();

                }
            }
        }
    }

```

```

        } catch (IOException e) {

// αδύνατη η σύνδεση με τη συσκευή

            e.printStackTrace();

            mConnectSuccessful = false;

        }

        return null;

    }

    @Override

    protected void onPostExecute(Void result) {

        super.onPostExecute(result);

        if (!mConnectSuccessful) {

            Toast.makeText(getApplicationContext(), "Could not
connect to device. Is it a Serial device? Also check if the UUID is
correct in the settings", Toast.LENGTH_LONG).show();

            finish();

        } else {

            msg("Connected to device");

            mIsBluetoothConnected = true;

            mReadThread = new ReadInput(); // εκκινεί η ανάγνωση
εισοόδου

        }

        progressDialog.dismiss();

    }

}

```

3.4.1.3 ActivityHelper.java

```
package com.example.sensormonitor;

import android.app.Activity;

import android.content.SharedPreferences;

import android.content.pm.ActivityInfo;

import android.preference.PreferenceManager;

public class ActivityHelper {

    public static void initialize(Activity activity) {

        SharedPreferences prefs =
PreferenceManager.getDefaultSharedPreferences(activity);

        String orientation = prefs.getString("prefOrientation",
"Null");

        if ("Landscape".equals(orientation)) {

activity.setRequestedOrientation(ActivityInfo.SCREEN_ORIENTATION_LAND
SCAPE); //προσανατολισμός οθόνης τοπίο-οριζόντιος

        } else if ("Portrait".equals(orientation)) {

activity.setRequestedOrientation(ActivityInfo.SCREEN_ORIENTATION_PORT
RAIT); //προσανατολισμός οθόνης πορτρέτο-κάθετος

        } else {

activity.setRequestedOrientation(ActivityInfo.SCREEN_ORIENTATION_FULL
_SENSOR); //πλήρες πλαίσιο οθόνης

        }

    }

}
```

3.4.1.4 PreferencesActivity.java

```
package com.example.sensormonitor;

import java.util.Map;

import android.content.SharedPreferences;

import android.content.SharedPreferences.OnSharedPreferenceChangeListener;

import android.os.Bundle;

import android.preference.EditTextPreference;

import android.preference.ListPreference;

import android.preference.Preference;

import android.preference.PreferenceActivity;

import android.preference.PreferenceManager;

public class PreferencesActivity extends PreferenceActivity
implements OnSharedPreferenceChangeListener {

    @Override

    public void onCreate(Bundle savedInstanceState) {

        super.onCreate(savedInstanceState);

        ActivityHelper.initialize(this);

        // χρήση για συμβατότητα με τις συσκευές Android

    }

    public void onSharedPreferenceChanged(SharedPreferences
sharedPreferences, String key) {
```

```

        Preference pref = findPreference(key); //αποθήκευση κλειδιού
για ανάκτηση

        if (pref instanceof ListPreference) { //preference για λίστα

            ListPreference listPref = (ListPreference) pref;

            pref.setSummary(listPref.getEntry());

            ActivityHelper.initialize(this);

        }

        if (pref instanceof EditTextPreference) { //preference για
κείμενο

            EditTextPreference editPref = (EditTextPreference) pref;

            pref.setSummary(editPref.getText());

        }

    }

    @Override

    protected void onPause() {

PreferenceManager.getDefaultSharedPreferences(this).unregisterOnShare
dPreferenceChangeListener(this);

        super.onPause();

    }

    @Override

    protected void onResume() {

PreferenceManager.getDefaultSharedPreferences(this).registerOnShareD
referenceChangeListener(this);

        Map<String, ?> keys =
PreferenceManager.getDefaultSharedPreferences(this).getAll();

```

```
        for (Map.Entry<String, ?> entry : keys.entrySet()) {
// Log.d("map values", entry.getKey() + ": " +
entry.getValue().toString());
            Preference pref = findPreference(entry.getKey());
            if (pref != null) {
                pref.setSummary(entry.getValue().toString());
            } //ιεραρχία προτιμήσεων
        }
        super.onResume();
    }
}
```

3.4.2 Αρχεία .xml

3.4.2.1 AndroidManifest.xml

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.example.sensormonitor">
    <uses-sdk android:targetSdkVersion="8"/>
    <uses-permission android:name="android.permission.BLUETOOTH" />
    // άδεια ανίχνευσης της συσκευής bluetooth
    <uses-permission
android:name="android.permission.BLUETOOTH_ADMIN" /> // άδεια
διαχείρισης των δεδομένων και ρύθμισης της συσκευής bluetooth
    <uses-permission
android:name="android.permission.WRITE_EXTERNAL_STORAGE"/> //
άδεια στην εφαρμογή να γράφει σε εξωτερικό αποθηκευτικό χώρο
    <application
        android:allowBackup="true"
        android:icon="@mipmap/ic_launcher"
        android:label="@string/app_name"
        android:roundIcon="@mipmap/ic_launcher_round"
        android:supportsRtl="true"
        android:theme="@style/AppTheme">

        <activity android:name=".MainActivity">
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />

                <category
android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>
        <activity android:name=".MonitoringScreen"></activity>
    </application>

</manifest>
```

3.4.2.2 activity_main.xml

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
//Το LinearLayout είναι μια ομάδα προβολής που ευθυγραμμίζει όλα τα
παιδιά σε μία κατεύθυνση, κάθετα ή οριζόντια. Ο προσανατολισμός
καθορίζεται με το χαρακτηριστικό (attribute) προσανατολισμού android:
orientation.
xmlns:android="http://schemas.android.com/apk/res/android"
xmlns:app="http://schemas.android.com/apk/res-auto"
xmlns:tools="http://schemas.android.com/tools"
android:layout_width="match_parent" //στο μέγεθος του προγόνου
android:layout_height="match_parent"
android:orientation="vertical" //κάθετος προσανατολισμός
tools:context=".MainActivity">

<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="wrap_content" //μέγεθος απαραίτητο να
απεικονίζονται οι τιμές
    android:layout_gravity="center"
    android:layout_marginTop="20dp">

<Button //κουμπί
    android:id="@+id/search"
    android:layout_width="wrap_content"
    android:layout_height="match_parent"
    android:layout_marginLeft="90dp"
    android:text="Search"/>

<Button
    android:id="@+id/connect"
    android:layout_width="wrap_content"
    android:layout_height="match_parent"
    android:layout_marginLeft="50dp"
    android:text="Connect" />

</LinearLayout>
```

```

<ListView //λίστα
    android:id="@+id/listview"
    android:layout_width="match_parent"
    android:layout_height="match_parent">
</ListView>

</LinearLayout>

```

3.4.2.3 activity_monitoring_screen.xml

```

<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    tools:context=".MonitoringScreen">

    <LinearLayout
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:orientation="horizontal" > //οριζόντιος
        προσανατολισμός

        <Button
            android:id="@+id/btnClearInput"
            android:layout_width="165dp"
            android:layout_height="match_parent"
            android:layout_weight="5"
            android:text="Clear Input" />

        <Button
            android:id="@+id/button"
            android:layout_width="165dp"
            android:layout_height="match_parent"
            android:layout_weight="5"
            android:text="Graphs" />

```

```
</LinearLayout>
```

```
<LinearLayout
```

```
    android:layout_width="match_parent"  
    android:layout_height="match_parent"  
    android:orientation="vertical">
```

```
<CheckBox // κουτί
```

```
    android:id="@+id/chkScroll"  
    android:layout_width="fill_parent"  
    android:layout_height="wrap_content"  
    android:layout_margin="0dp"  
    android:checked="true"  
    android:text="Scroll" />
```

```
<CheckBox
```

```
    android:id="@+id/chkReceiveText"  
    android:layout_width="fill_parent"  
    android:layout_height="wrap_content"  
    android:checked="true"  
    android:text="Read" />
```

```
<ScrollView //οθόνη ποής
```

```
    android:id="@+id/viewScroll"  
    android:layout_width="fill_parent"  
    android:layout_height="0dp"  
    android:layout_weight="1" >
```

```
<TextView //κείμενο
```

```
    android:id="@+id/txtReceive"  
    android:layout_width="fill_parent"  
    android:layout_height="wrap_content"  
    android:layout_gravity="top"  
    android:background="#eeeeee"  
    android:typeface="monospace"  
    android:hint="Message shows up here ..."  
    android:scrollbars="vertical"  
    android:layout_marginTop="20dp"/>
```

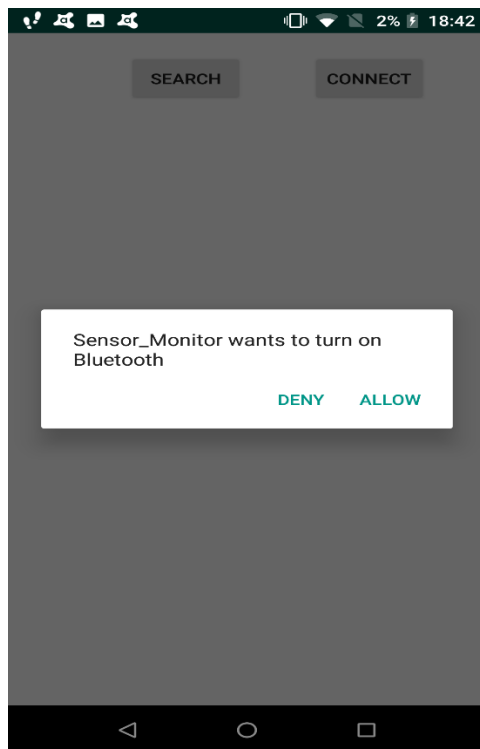
```
</ScrollView>
```

```
</LinearLayout>
```

```
</LinearLayout>
```

Σημείωση: Τα σχόλια γράφτηκαν ελεύθερα στο ελληνικό αλφάβητο για την παρούσα εργασία και δεν αναγνωρίζονται από τους συγγραφείς και μεταγλωττιστές κειμένου

Κεφάλαιο 4. Σύσταση και Αποτελέσματα Εφαρμογής

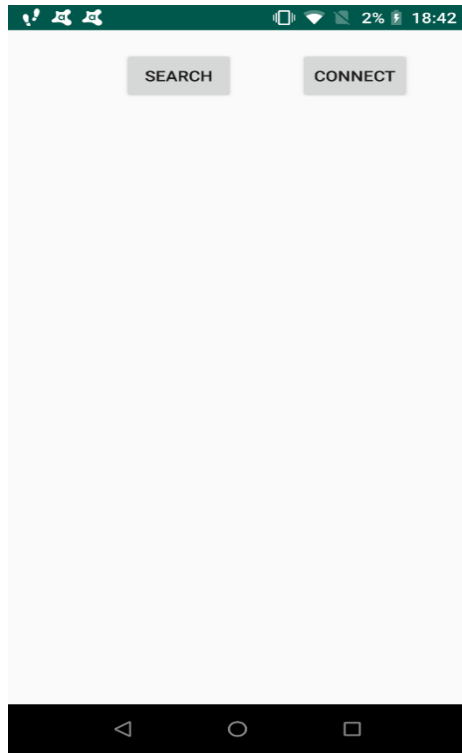


Εικόνα 6: Άδεια χρήσης των bluetooth της Android συσκευής

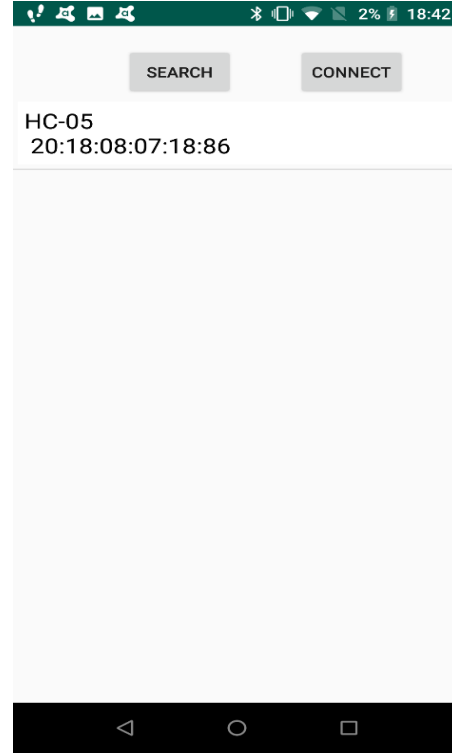
Η εφαρμογή για Android που δημιουργήθηκε αποτελείται από μια σειρά αρχείων με επέκταση .java που επιτελούν λειτουργικό ρόλο στην σύσταση της εφαρμογής, από μια σειρά αρχείων .xml, που είναι αυτά που καθορίζουν το διαδραστικό με τη μεριά του χρήστη, περιβάλλον και ένα αρχείο AndroidManifest.xml. Το πρώτο από αρχεία .java αποσκοπεί στην αναγνώριση των Bluetooth συσκευών που βρίσκονται εντός της εμβέλειας όπου υπάρχει δυνατότητα αποστολής και λήψης δεδομένων. Οπότε, η πρώτη ενέργεια που εκτελεί το αρχείο MainActivity.java είναι να αναζητά ενεργές συσκευές Bluetooth και να τις εμφανίζει σε μία λίστα, όπως είναι εμφανές στις Εικόνες 7α και 7β, όπου το αρχείο MainActivity.java έχει διασυνδεθεί με το αρχείο activity_main.xml. Στη διεπαφή (interface) που δημιουργήθηκε στο αρχείο activity_main.xml, διακρίνονται δύο κουμπιά. Όταν πατηθεί το κουμπί «SEARCH» αυτόματα μεταβαίνει στη συνάρτηση του αρχείου MainActivity.java. Η πρώτη ενέργεια που έχει δηλωθεί στο AndroidManifest.xml είναι να ζητηθεί άδεια χρήσης των bluetooth της Android συσκευής, όπως φαίνεται στην Εικόνα 6.

Ακολούθως, πραγματοποιείται αναζήτηση διαθέσιμων-ενεργών συσκευών Bluetooth, οι οποίες καταγράφονται σε λίστα στην οθόνη, όπως στην Εικόνα 7β, όπου διακρίνεται, βέβαια, μόνο το Bluetooth HC-05 που χρησιμοποιήθηκε για την εργασία.

Πατώντας, τώρα, πάνω στη συσκευή που επιθυμεί ο χρήστης και στη συνέχεια στο κώμβιο «CONNECT» της διεπαφής, μεταβαίνει ο έλεγχος στη MainActivity.java η οποία τελεί pairing με την επικείμενη προς χρήση συσκευή bluetooth και εν συνεχεία, συνδέεται με την MonitoringScreen.java, μέσω του αρχείου AndroidManifest.xml, το οποίο έχει καθοριστικό ρόλο στη διασύνδεση των αρχείων.



Εικόνα 7α: Διεπαφή activity_main.xml



Εικόνα 7β: Έξοδος MainActivity.java σε συνάρτηση με activity_main.xml

Το αρχείο MonitoringScreen.java λαμβάνει τα δεδομένα που αποστέλλει το bluetooth σε πραγματικό χρόνο και τα εμφανίζει με το ρυθμό αποστολής στην οθόνη του κινητού σε κατακόρυφη λίστα, όπως φαίνεται στην Εικόνα 8. Το interface δημιουργήθηκε στο αρχείο activity_monitoring_screen.xml και διαθέτει δύο check boxes. Το check box «Scroll» επιλεγμένο επιτρέπει τη ροή της οθόνης προς τα κάτω αυτόματα, για να συνάδει με την εμφάνιση κάθε νέας μέτρησης που δεν χωράει πλέον στην οθόνη. Αν το συγκεκριμένο κουμπί δεν είναι επιλεγμένο η οθόνη είναι στατική. Ωστόσο, οι μετρήσεις καταγράφονται, αλλά για να είναι εμφανείς στο χρήστη θα πρέπει να σύρει χειροκίνητα την οθόνη προς τα πάνω.



Εικόνα 8: Μετρήσεις και Αποτελέσματα

Με το check box «READ» επιλεγμένο οι μετρήσεις καταγράφονται, ενώ σε αντίθετη περίπτωση η διαδικασία καταγραφής και εμφάνισης διακόπτεται, έως ότου πραγματοποιηθεί επανεπιλογή από το χρήστη. Με το κουμπί «CLEAR INPUT» τα δεδομένα που υπάρχουν στην οθόνη διαγράφονται και η καταγραφή ξεκινά απ' την αρχή. Το κουμπί «GRAPHS», είναι απόρροια της πρότασης για μελλοντική εργασία που θα επεκτείνει την εφαρμογή προσθέτοντας γραφικές παραστάσεις και διάφορα άλλα γραφήματα των μετρήσεων και των αποτελεσμάτων.

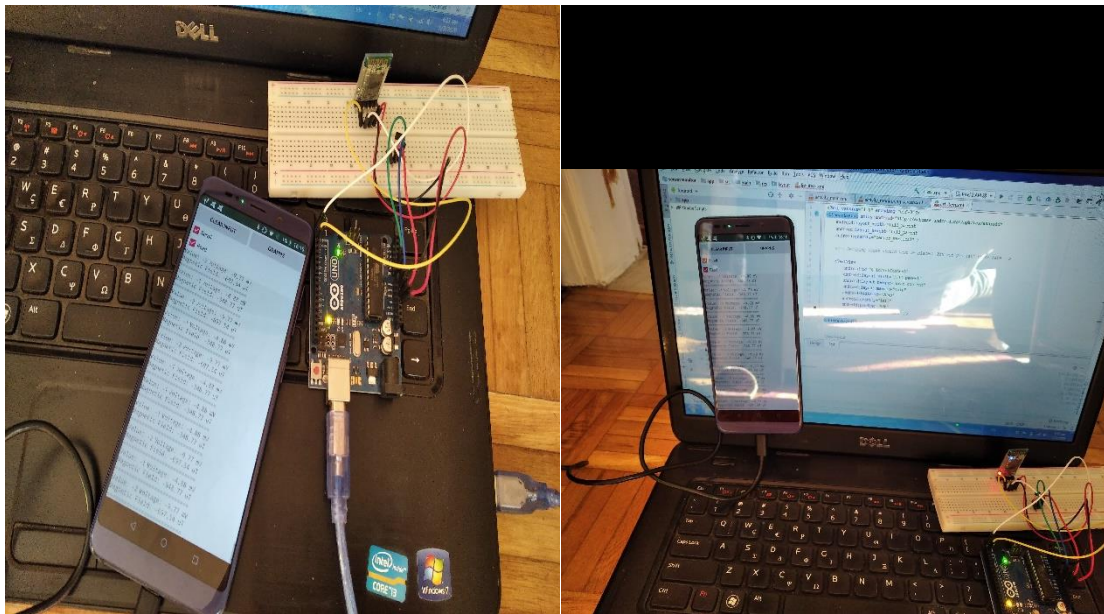
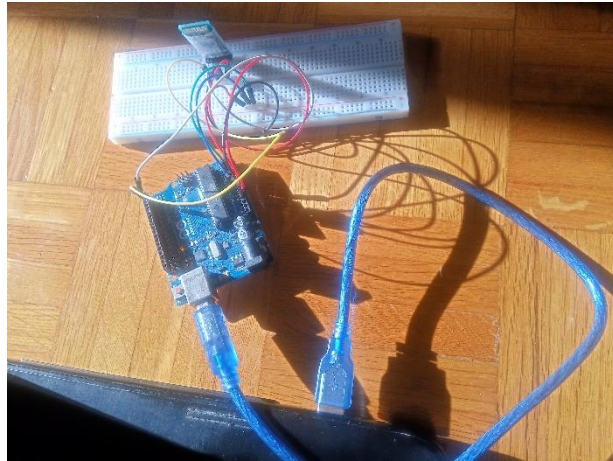
Το αρχείο ActivityHelper.java επωμίζεται το έργο της προσαρμογής του προσανατολισμού στις ανάγκες της οθόνης του κινητού.

Ενώ το αρχείο PreferencesActivity.java φέρει αναγνωρίσεις για τα preferences, δηλαδή, βιβλιοθήκες που χρειάζεται να χρησιμοποιηθούν από την εφαρμογή.

Ο κώδικας που εγγράφει στην πλατφόρμα συγγραφής Android εφαρμογών, του Android Studio έπρεπε να ελεγχθεί για σφάλματα λογικά, σημασιολογικά και συντακτικά. Για να υπάρχει όμως μια ανάδραση που θα επέτρεπε τον καλύτερο χειρισμό του ζητήματος, το κινητό τηλέφωνο τύπου Smartphone MLS MX PRO, χρησιμοποιήθηκε ως εργαλείο ελέγχου προγραμματισμού και διεπαφής, μέσω ενσύρματης διασύνδεσης με τον φορητό υπολογιστή όπου εκτελείτο η εφαρμογή. Στη συσκευή ενεργοποιήθηκαν οι επιλογές προγραμματιστή ως εξής. Στην καρτέλα ρυθμίσεις επιλέγεται η καρτέλα σύστημα ή σχετικά με το τηλέφωνο (about phone), και στη συνέχεια πρέπει να πατηθεί επτά φορές ο αριθμός έκδοσης-κατασκευής του κινητού (build number), έτσι ενεργοποιούνται οι επιλογές προγραμματιστή (developer options), που κάνουν ορατή τη συσκευή στο Android Studio και επιτρέπουν την εκτέλεση της εφαρμογής στη συγκεκριμένη συσκευή.

Στο τέλος της εργασίας και αφού έχει επιβεβαιωθεί η ορθότητα της εφαρμογής είναι εφικτή η εγκατάσταση αυτής στη συσκευή και η σχετικά αυτόνομη χρήσης της (εννοείται ότι απαιτείται και το υπόλοιπο ηλεκτρονικό κομμάτι Arduino-bluetooth-αισθητήρας Hall εν ενεργεία για να λειτουργήσει). Η εγκατάσταση πραγματοποιείται από το Android Studio με επιλογή από το κύριο μενού της καρτέλας «Build» στη συνέχεια «Build Bundle(s)/APK(s)» και «Build APK(s)». Ως εκ τούτου, δημιουργείται το αρχείο app-debug.apk, το οποίο είναι το εκτελέσιμο αρχείο της εφαρμογής.

Εδώ υπάρχουν και κάποιες φωτογραφίες από τη διάταξη όπως υλοποιήθηκε ρεαλιστικά:



Εικόνα 9: Φωτογραφίες ρεαλιστικής διάταξης και αποτελεσμάτων



Κεφάλαιο 5. Συμπεράσματα

4.1 Προβληματισμοί και Προοπτικές

Η γνώση της πληροφορίας κάθε στιγμή και η απτή παρουσίαση των δεδομένων είναι πάντα καίριο ζητούμενο σε πολλές εφαρμογές της επιστήμης και της καθημερινότητας. Με την εργασία αυτή έγινε εφικτή η απεικόνιση μετρήσεων μαγνητικού πεδίου και οι μεταβολές αυτού. Οι εφαρμογές αυτής της εργασίας έγκεινται κυρίως στη βιομηχανία, αλλά και σε ερευνητικό επίπεδο. Η απεικόνιση μετρήσεων του μαγνητικού πεδίου και των αλλαγών που μπορούν να επιφέρουν παρεμβαλλόμενα μέσα, και πόσο μάλλον σε πραγματικό χρόνο μπορεί να συντελέσει στη βελτίωση διαφόρων συστημάτων που χρησιμοποιούν αισθητήρα Hall και όχι μόνο.

→ Εφαρμογή Αισθητήρα Hall

Πρακτικά, οι αισθητήρες ή διακόπτες Hall χρησιμοποιούνται άρδην στη σύγχρονη βιομηχανία κινητήρων και είναι επιφορτισμένοι με τον εντοπισμό της θέσης και των στροφών του εκκεντροφόρου και του στροφαλοφόρου άξονα. Ο αισθητήρας Hall είναι ένας πολύ ακριβής τρόπος για να γνωρίζει ο εγκέφαλος του οχήματος ανά πάσα στιγμή την θέση και τη ταχύτητα των δυο αυτών αξόνων. Οι περισσότερες εφαρμογές χρησιμοποιούν διάκενα και ελάσματα. Το διάκενο αφήνει να περάσει το μαγνητικό πεδίο το οποίο δημιουργείται από ένα μόνιμο μαγνήτη. Όταν το έλασμα περάσει από το μαγνητικό πεδίο, τότε το διακόπτει και δημιουργείται μια αλλαγή στην τάση την οποία αντιλαμβάνεται ο εγκέφαλος. Όταν περνάει από το έλασμα τότε η τάση πέφτει πρακτικά στο μηδέν. Ενώ όταν παρεμβάλλεται το έλασμα, η τάση ανεβαίνει σε μια συγκεκριμένη τιμή.

Σε αντίθεση με τα μαγνητόμετρα fluxgate και τους αισθητήρες μαγνητοαντίστασης, οι αισθητήρες Hall παρέχουν πολύ χαμηλότερη ακρίβεια στις μετρήσεις, γεγονός το οποίο αποτελεί και το βασικό τους μειονέκτημα.

4.2 Μελλοντικές Εργασίες

Η εργασία αυτή είναι αφητηρία για καινούργιες ιδέες και εναλλακτικές διατάξεις. Η πρώτη ενέργεια που θα μπορούσε να γίνει είναι να γραφεί κώδικας που θα μπορούσε να αναπαριστά σε γραφικές παραστάσεις τις μετρήσεις που παράγει ο αισθητήρας Hall. Οι γραφικές παραστάσεις αυτές θα μπορούσαν να τρέχουν σε πραγματικό χρόνο και να απεικονίζουν σύγχρονα τις μετρήσεις. Θα μπορούσε επίσης να δημιουργηθεί βάση δεδομένων για περαιτέρω μελέτη και έρευνα των στοιχείων που προκύπτουν από τις μετρήσεις.

Μια άλλη εργασία θα μπορούσε να αφορά τη μετάδοση των δεδομένων μέσω ασύρματου δικτύου WiFi. Αυτή η μετατροπή θα συντελούσε σε μεταδόσεις μεγαλύτερης εμβέλειας και στη δημιουργία δυνατότητας πολλαπλότητας της χρήσης της εφαρμογής και της παρακολούθησης των μετρήσεων σε ένα τοπικό δίκτυο ηλεκτρονικών διευθύνσεων. Ακόμα, θα μπορούσε να υπάρχει αναμετάδοση σε απομακρυσμένες συσκευές μέσω WiFi και εφαρμογών κλειστού κύκλου, κάτι που εφαρμόζεται ήδη σε διάφορες εταιρείες, όσον αφορά σε παρακολούθηση δεδομένων και στατιστικών στοιχείων. Αυτό είναι πολύ εξυπηρετικό όταν απαιτείται εργασία και επίβλεψη από μακριά.

Ακόμα, πολλοί άλλοι αισθητήρες (πίεσης, θερμοκρασίας, υγρασίας κ.α.) θα μπορούσαν να πάρουν τη θέση του αισθητήρα Hall στη διάταξη για να ικανοποιήσουν τις εκάστοτε ανάγκες. Αυτό μπορεί να γίνει εύκολα με την κατάλληλη αλλαγή του κώδικα του Arduino για την καταγραφή και εμφάνιση των μετρήσεων των αντίστοιχων αισθητήρων.

Βιβλιογραφία

- [1]<http://old-2017.metal.ntua.gr/uploads/4701/1180/chap7.pdf>
- [2]http://circuits.ece.ntua.gr/documents/DiplomaTheses/Anastasia%20Tsalikidou_Design_and_Implementation_of_a_Closed_Loop_AMR_Magnetic_Sensor.pdf
- [3]<http://www.teicrete.gr/physics/lab/fdm/eppa/mk/Hall%20effect.pdf>
- [4]<https://electronics hobbyists.com/hc-05-bluetooth-module-interfacing-with-arduino-arduino-bluetooth-module-tutorial/>
- [5] Sabeen Tahira, Sheikh Tahir Bakhsha and Abdulrahman H. Altalhia, “An Efficient Route Maintenance Protocol for Dynamic Bluetooth Networks”, Accepted 28 July 2016
- [6] <http://ioarvanit.gr/archives/2889>
- [7]<https://create.arduino.cc/projecthub/electropeak/getting-started-with-hc-05-bluetooth-module-arduino-e0ca81>
- [8]https://www.fecegypt.com/uploads/dataSheet/1522241762_bluetooth%20module%20hc05.pdf
- [9] Pieterse H and Olivier MS, “Bluetooth Command and Control Channel, Computers & Security”, 25 May 2014
- [10] <https://components101.com/wireless/hc-05-bluetooth-module>
- [11] Helmut Faugel and Volodymyr Bobkov, “Open source hard- and software: Using Arduino boards to keep old hardware running”, 8 February 2013
- [12] Petteri Teikari, Raymond P. Najjar, Hemi Malkkic, Kenneth Knoblauch, Dominique Dumortier, Claude Gronfier and Howard M. Cooper, “An inexpensive Arduino-based LED stimulator system for vision research”, Accepted 13 September 2012
- [13] Masato Ishikawa *and* Ichiro Maruta, “Rapid Prototyping for Control Education using Arduino and Open-Source Technologies”, 2010

[14] F. A. Candelas, G. J. García, S. Puente, J. Pomares, C.A. Jara, J. Pérez and D. Mira, F. Torres, “Experiences on using Arduino for laboratory experiments of Automatic Control and Robotics”, 2015

[15] <https://www.arduino.cc/>

[16] <https://developer.android.com/guide/platform>

[17] Itienne Payet and Fausto Spoto “Static analysis of Android programs”, 25 May 2012

[18] Hae-Duck J. Jeong, Woojin Lee, Jiyoung Lim and WooSeok Hyun, “Utilizing a Bluetooth remote lock system for a smartphone”, 2015

[19] Ajay Kumar Jha, Sunghee Lee and Woojin Lee , “An empirical study of configuration changes and adoption in Android apps” , Accepted 24 June 2019

[20] <https://www.gme.cz/data/attachments/dsh.772-148.1.pdf>

[21]<https://www.instructables.com/id/Modify-The-HC-05-Bluetooth-Module-Defaults-Using-A/>

[22] Chenhui Xie, Jian Zhou, ShanShan Li and LuYing Jia, “The design and implementation of mobile monitoring system of transmitting station based on Android Platform”, 2012