



ΤΕΙ ΠΕΙΡΑΙΑ
ΣΧΟΛΗ ΤΕΧΝΟΛΟΓΙΚΩΝ ΕΦΑΡΜΟΓΩΝ
ΤΜΗΜΑ ΗΛΕΚΤΡΟΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΙΚΩΝ ΣΥΣΤΗΜΑΤΩΝ

ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ

Φωνητικές εντολές για κίνηση αμαξιδίου μέσω Android

Γεώργιος Ζησόπουλος Α.Μ. 34527

Εισηγητής: Ιωάννης Ν. Έλληνας , Καθηγητής

ΑΙΓΑΛΕΩ
ΜΑΙΟΣ 2017

ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ

Φωνητικές εντολές για κίνηση αμαξιδίου μέσω Android IE

Γεώργιος Ζησόπουλος

A.M 34527

Εισηγητής:
Ιωάννης Ν. Έλληνας , Καθηγητής

Εξεταστική Επιτροπή:

Ημερομηνία εξέτασης / /2017

ΕΥΧΑΡΙΣΤΙΕΣ

Η παρούσα πτυχιακή εργασία με θέμα “Φωνητικές Εντολές Για Κίνηση Αμαξιού Μέσω AndroidIE”, πραγματοποιήθηκε στο πλαίσιο της πτυχιακής εργασίας του τμήματος “Ηλεκτρονικών Υπολογιστικών Συστημάτων”. Η ολοκλήρωση της υλοποιήθηκε με την υποστήριξη ενός αριθμού ανθρώπων, στους οποίους θα ήθελα να εκφράσω τις θερμότερες ευχαριστίες μου.

Αρχικά θα ήθελα να ευχαριστήσω τον επιβλέποντα καθηγητή μου κύριο Ιωάννη Ν. Έλληνα για την συνεχή καθοδήγηση, την αμέριστη υποστήριξη, τις ουσιώδεις συμβουλές, καθώς και την αδιάκοπη συμπαράσταση και ενθάρρυνση που μου παρείχε σε όλο αυτό το διάστημα.

Επίσης θα ήθελα να ευχαριστήσω θερμά τους κοντινούς φίλους μου για τη πολύτιμη βοήθεια τους σχετικά με το υλικό της πτυχιακής μου εργασίας.

Τέλος θα ήθελα να ευχαριστήσω την οικογένεια μου που μου έμαθαν να προσπερνά τα εμπόδια και συνέβαλαν με τη συμπαράσταση τους την υπομονή και την θετική τους σκέψη στην εκπλήρωση του στόχου μου.

ΠΕΡΙΛΗΨΗ

Ο στόχος της παρούσας πτυχιακής είναι η ανάπτυξη μιας εφαρμογής Android για να ελέγξει ένα ρομπότ. Μια εκ των δυνατοτήτων ανάπτυξης της πλατφόρμας Android είναι η συγχώνευση της με τον μικροεπεξεργαστή Arduino.

Η παρούσα πτυχιακή εστιάζει στο αποτέλεσμα κατά τον συνδυασμό Android και Arduino για την δημιουργία ενός αμαξιδίου που θα ελέγχεται από μια Android συσκευή και θα λειτουργεί μέσω φωνητικών εντολών.

ΕΠΙΣΤΗΜΟΝΙΚΗ ΠΕΡΙΟΧΗ : Ανάπτυξη Εφαρμογών Android
ΛΕΞΕΙΣ ΚΛΕΙΔΙΑ : Arduino, Android, Φωνητικές Εντολές, Αμαξίδιο

ΠΕΡΙΟΧΟΜΕΝΑ

ΚΕΦΑΛΑΙΟ 1 : ΕΙΣΑΓΩΓΗ	17
1.1 Σκοπός της πτυχιακής εργασίας	17
1.2 Δομή πτυχιακής.....	17
ΚΕΦΑΛΑΙΟ 2 : ARDUINO.....	19
2.1 Ιστορία Arduino	19
2.2 Τι είναι ένας Arduino.....	20
2.3 Τι μπορεί να κάνει	22
2.4 Πως είναι το board του Arduino.....	23
2.5 Γιατί να χρησιμοποιήσω έναν Arduino	26
2.6 Το περιβάλλον ανάπτυξης του Arduino.....	27
2.7 Γράφοντας προγράμματα στον Arduino(Sketches).....	28
ΚΕΦΑΛΑΙΟ 3 : MOTORSHIELD	31
3.1 Εισαγωγή	31
3.2 Τι είναι Motor shield κι από τι αποτελείται	32
3.3 Τι απαιτείται για να το προγραμματίσετε	34
3.4 Χρησιμοποιώντας κινητήρες συνεχούς ρεύματος (dcmotors).....	35
ΚΕΦΑΛΑΙΟ 4 : BLUETOOTH	37
4.1 Εισαγωγή	37
4.2 Ιστορία του Bluetooth	37
4.3 Εφαρμογές	38
4.4 Γιατί να χρησιμοποιήσω Bluetooth	39
4.5 Πως το συνδέεται το Bluetooth στον Arduino και από τι αποτελείται.....	39

ΚΕΦΑΛΑΙΟ 5 : ANDROID	41
5.1 Εισαγωγή στο Android.....	41
5.2 Η αρχιτεκτονική του Android.....	43
5.3 Ο πυρήνας του Android	44
5.4 Η ανάπτυξη του Android και οι εκδόσεις του	45
5.4.1 Android 1.5 CUPCAKE	46
5.4.2 Android 1.6 DONUT	47
5.4.3 Android 2.0 ECLAIR.....	48
5.4.4 Android 2.2 FROYO	49
5.4.5 Android 2.3 GINGERBREAD	50
5.4.6 Android 3.0 HONEYCOMB	51
5.4.7 Android 4.0 ICE CREAM SANDWICH	52
5.4.8 Android 4.1.0 JELLY BEAN	53
5.4.9 Android 4.4 KITKAT	54
5.4.10 Android 5.0 - 5.1 LOLLIPOP	55
5.5 Διαχείριση μνήμης στο Android	57
5.6 Βιβλιοθήκες στο Android.....	59
5.7 Πλαίσιο εφαρμογής – Application Framework	60
5.8 Playstore	61
5.9 Λογαριασμός Google–Gmail	62
5.10 Υπηρεσίες και εφαρμογές Google	63
5.11 Google Maps	63
5.12 Εφαρμογές Android – Android Applications.....	64
ΚΕΦΑΛΑΙΟ 6 : ΔΗΜΙΟΥΡΓΙΑ ΕΦΑΡΜΟΓΗΣ ΣΕ ANDROID ΠΕΡΙΒΑΛΛΟΝ.....	65
6.1 Εγκατάσταση Περιβάλλοντος Υλοποίησης της Εφαρμογής - Android Studio	65
6.2 Η εργασία μας	74

6.2.1 Σημαντικά αρχεία Android Studio	74
6.2.2 Activities	75
6.2.3 Επεξήγηση Συναρτήσεων και Μεθόδων της Εφαρμογής.....	77
6.2.4 Επεξήγηση της Εφαρμογής	79
6.3 Αμαξίδιο Εργασίας	83
6.4 Κώδικας Arduino	84
ΚΕΦΑΛΑΙΟ 7 : ΣΥΜΠΕΡΑΣΜΑΤΑ.....	89
Παράρτημα	91
Βιβλιογραφία.....	99

ΠΙΝΑΚΑΣ ΕΙΚΟΝΩΝ

Εικόνα 2 - 1 : Λογότυπο Arduino.....	19
Εικόνα 2 - 2 : Πλακέτα Arduino Uno.....	22
Εικόνα 2 - 3 : Τι βρίσκεται στην πλακέτα Arduino	23
Εικόνα 3 - 1 : Motorshield	31
Εικόνα 3 - 2 : Μέρη motorshield.....	33
Εικόνα 3 - 3 : Κινητήρας Ρεύματος.....	35
Εικόνα 4 - 1 : Bluetooth.....	39
Εικόνα 4 - 2 : Σύνδεση Arduino με Bluetooth	39
Εικόνα 5 - 1 :Λογότυπο Android.....	41
Εικόνα 5 - 2 : Επίπεδα Στοιβάς Android.....	43
Εικόνα 5 - 3 : Εκδόσεις Android	45
Εικόνα 5 - 4 : Android 1.5 Cupcake.....	46
Εικόνα 5 - 5 : Android 1.6 DONUT	47
Εικόνα 5 - 6 : Android 2.0 ECLAIR.....	48
Εικόνα 5 - 7 : Android 2.2 FROYO.....	49
Εικόνα 5 - 8 : Android 2.3 GINGERBREAD	50
Εικόνα 5 - 9 : Android 3.0 HONEYCOMB	51
Εικόνα 5 - 10 : Android Ice Cream Sandwich.....	52
Εικόνα 5 - 11 : Android JellyBean 4.1.0	53
Εικόνα 5 - 12 : Android 4.4 KitKat	54
Εικόνα 5 - 13 : Android 5.0 - 5.1 Lollipop	55
Εικόνα 5 - 14 : Google Playstore	61
Εικόνα 5 - 15 : Google Maps.....	63
Εικόνα 5 - 16 : Εφαρμογές Android.....	64
Εικόνα 6 - 1 : Download Android Studio.....	65
Εικόνα 6 - 2 : Κατέβασμα JDK	66
Εικόνα 6 - 3 : Εγκατάσταση JDK.....	67
Εικόνα 6 - 4 : Εγκατάσταση Android Studio	68
Εικόνα 6 - 5 : JDK	68
Εικόνα 6 - 6 : Επίλογη components για εγκατάσταση	69
Εικόνα 6 - 7 : Configuration Settings.....	69
Εικόνα 6 - 8 : Quick Start	70
Εικόνα 6 - 9 : Configure new project	71
Εικόνα 6 - 10 : Select form factors	72
Εικόνα 6 - 11 : Εισαγωγή Activity.....	73
Εικόνα 6 - 12 : voiceRecoPro.....	74
Εικόνα 6 - 13 : Αρχεία Android Studio.....	74
Εικόνα 6 - 14 : Κύκλος ζωής Activity	77
Εικόνα 6 - 15 : Οθόνη Εφαρμογής 1	80
Εικόνα 6 - 16 : Οθόνη Εφαρμογής 2	80
Εικόνα 6 - 17 : Οθόνη Εφαρμογής 3	81
Εικόνα 6 - 18 : Οθόνη Εφαρμογής 4	82
Εικόνα 6 - 19 :Αμαξίδιο Εργασίας	83

ΠΙΝΑΚΑΣ ΠΙΝΑΚΩΝ

Πίνακας 2 - 1 Εκδόσεις Arduino	20
Πίνακας 2 - 2: Κουμπιά Γραμμής Εργαλείων	29
Πίνακας 6 - 1 :Μέθοδοι ενός Activity	76
Πίνακας 6 - 2 : Συνάρτηση list ()	78
Πίνακας 6 - 3 : Σύνδεση με Συσκευή.....	78
Πίνακας 6 - 4 : Συνάρτησης onActivityResult().....	79
Πίνακας 6 - 5 : Συναρτήση forward().....	79
Πίνακας 6 - 6 : Κώδικας Arduino.....	84
Πίνακας 6 - 7 : Εισαγωγή βιβλιοθήκης.....	85
Πίνακας 6 - 8 : Συνάρτηση setup().....	85
Πίνακας 6 - 9 : Συνάρτηση loop().....	86
Πίνακας 6 - 10 : Κίνηση αμαξιδίου αριστερά ή δεξιά.....	86

ΚΕΦΑΛΑΙΟ 1 : ΕΙΣΑΓΩΓΗ

Η παρούσα πτυχιακή απαρτίζεται από δύο κυρίως κεφάλαια το ένα αφορά το λειτουργικό Android, τα χαρακτηριστικά, την αρχιτεκτονική του κ.α. Ενώ σημαντικό κεφάλαιο για την συγκεκριμένη πτυχιακή θα αναφέρεται στον μικροεπεξεργαστή Arduino Uno για τον οποίο θα αναφέρουμε ιστορικά στοιχεία και θα εξηγήσουμε τι είναι το Arduino, γιατί και πως το χρησιμοποιούμε.

1.1 Σκοπός της πτυχιακής εργασίας

Σκοπός της εργασίας είναι η εκμάθηση και εξοικείωση με το Arduino και το Android, καθώς και η απόκτηση εμπειρίας στην κατασκευή εφαρμογών σε Android περιβάλλον. Στην πτυχιακή αυτή θα δούμε πως δημιουργούμε ένα αμαξίδιο, το οποίο θα εκτελεί βασικές κινήσεις (μπροστά, πίσω, δεξιά, αριστερά) δεχόμενο φωνητικές εντολές. Όλα αυτά θα τα δούμε και θα τα αναλύσουμε στα παρακάτω κεφάλαια.

1.2 Δομή πτυχιακής

Στην πτυχιακή αυτή εργασία παρουσιάζουμε την δημιουργία ενός αμαξιδίου και το πώς αυτό κινείται μέσω φωνητικών εντολών. Πιο αναλυτικά :

- **Κεφάλαιο 1 - Περίληψη** : Πρόκειται γι' αυτό το κεφάλαιο, το οποίο περιέχει γενικές πληροφορίες για την εργασία, τους σκοπούς και τους στόχους της εργασίας.
- **Κεφάλαιο 2 - Arduino** : παρουσιάζεται το Arduino, που αποτελεί βασικό στοιχείο της πτυχιακής μας.
- **Κεφάλαιο 3 - Εισαγωγή στα Motorshield** : Στο κεφάλαιο αυτό γίνεται περιγραφή στα Motorshield τι είναι και από τι αποτελούνται.

- **Κεφάλαιο 4 - Bluetooth** : Σε αυτό το κεφάλαιο παρουσιάζεται η ασύρματη τεχνολογία Bluetooth, λόγω του ότι είναι και το πρωτόκολλο που χρησιμοποιήθηκε για τη σύνδεση του Arduino με την εφαρμογή Android που δημιουργήθηκε.
- **Κεφάλαιο 5 - Android** : Σε αυτό το κεφάλαιο γίνεται μια γνωριμία με το τι είναι Android. Ξεκινώντας από μια ιστορική αναδρομή ενώ στη συνέχεια αναφέρονται οι εκδόσεις του λειτουργικού καθώς και η αρχιτεκτονική του.
- **Κεφάλαιο 6 - Δημιουργία εφαρμογής σε Android περιβάλλον:** Δείχνουμε πως μπορεί να δημιουργηθεί μια εφαρμογή σε Android περιβάλλον, γίνεται εγκατάσταση του Android Studio και οι ρυθμίσεις που έγιναν. Παρουσιάζεται η εφαρμογή που δημιουργήθηκε στο Android Studio και πως λειτουργεί καθώς και επεξήγηση κάποιων σημείων του κώδικά.
- **Κεφάλαιο 7 - Συμπεράσματα** : Γίνεται σύνοψη της εργασίας.
- **Παράρτημά** : Δίνεται μέρος του κώδικα της εργασίας μας.
- **Βιβλιογραφία** όπου δίνονται οι πηγές από όπου αντλήθηκαν πληροφορίες για την πτυχιακή.

ΚΕΦΑΛΑΙΟ 2 : ARDUINO



Εικόνα 2 - 1 : Λογότυπο Arduino

2.1 Ιστορία Arduino

Το 2005, ένα σχέδιο κίνησε προκειμένου να φτιαχτεί μία συσκευή για τον έλεγχο προγραμμάτων διαδραστικών σχεδίων από μαθητές, η οποία θα ήταν πιο φθηνή από άλλα πρωτότυπα συστήματα διαθέσιμα εκείνη την περίοδο. Οι ιδρυτές Massimo Banzι και David Cuertielles ονόμασαν το σχέδιο από τον Arduin της Inrea και ξεκίνησαν να παράγουν πλακέτες σε ένα μικρό εργοστάσιο στην Ιβρέα, κωμόπολη της επαρχίας Τορίνο στην περιοχή Πεδεμόντιο της βορειοδυτικής Ιταλίας- την ίδια περιοχή στην οποία στεγαζόταν η εταιρία υπολογιστών Olivetti.

Το σχέδιο Arduino είναι μία διακλάδωση της πλατφόρμας Wiring για λογισμικό ανοικτού κώδικα και προγραμματίζεται χρησιμοποιώντας μια γλώσσα βασισμένη στο Wiring (σύνταξη και βιβλιοθήκες), παρόμοια με την C++ με απλοποιήσεις και αλλαγές, καθώς και ένα ολοκληρωμένο περιβάλλον ανάπτυξης (IDE) [1].

Εκδόσεις Arduino

Χρονολογία έκδοσης	Έκδοση	Τεχνολογία που βασίστηκε
2006	Arduino Mini	
2008	Arduino Duemilanove	Atmel Atmega328
2009	Arduino Mega	Atmel Atmega1280
2012	Arduino Leonardo	Atmel ATmega32u4
2012	Arduino Due	Atmel SAM3X8E με πυρήνα ARM Cortex-M3
2012	Arduino Micro	Atmel ATmega32u4
2013	Arduino Robot	Atmel Atmega32u4 πρώτο Arduino με ρόδες
2013	Arduino Yun	Atmel Atmega32u4 & Atheros AR9331 και είναι το πρώτο προϊόν wifi που συνδιάζει το Arduino με το Linux

Πίνακας 2 - 1 Εκδόσεις Arduino

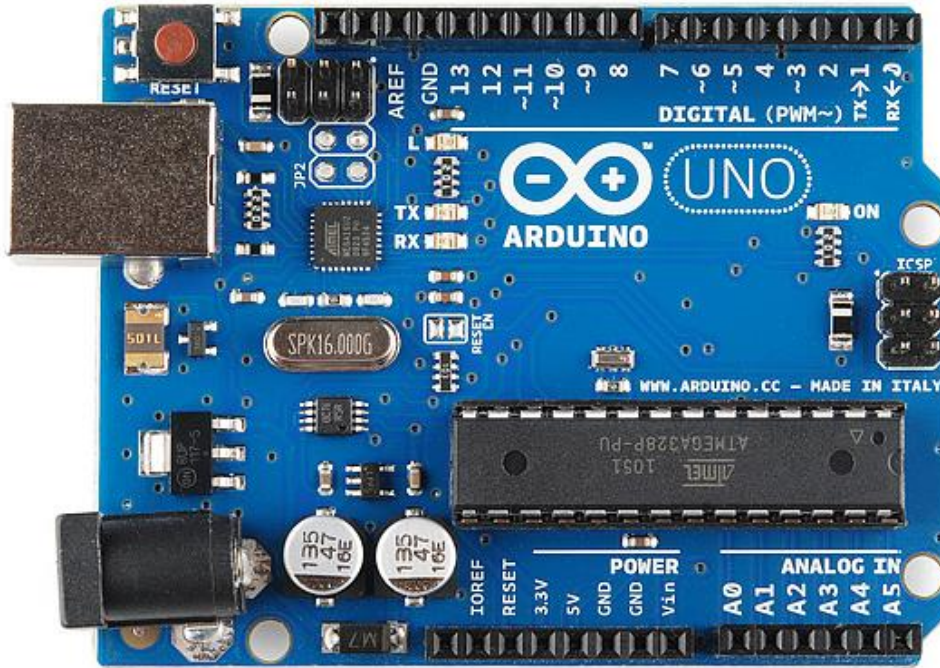
2.2 Τι είναι ένας Arduino

Το Arduino είναι μια ανοιχτού-κώδικα πλατφόρμα που χρησιμοποιείται για να κατασκευάσουμε ηλεκτρονικά προγράμματα. Το Arduino αποτελείται τόσο από μια φυσική πλακέτα για προγραμματιζόμενο κύκλωμα (που αναφέρεται ως προγραμματιστής ή μικροελεγκτής) και ένα κομμάτι του λογισμικού η IDE Ολοκληρωμένο Περιβάλλον

Ανάπτυξης (Integrated Development Environment) που τρέχει στον υπολογιστή σας και χρησιμοποιείται για να γράψουμε και να ανεβάσουμε κώδικα στη πλακέτα.

Η πλατφόρμα Arduino έχει γίνει αρκετά δημοφιλής με τους ανθρώπους που για πρώτη φορά ασχολούνται με τα ηλεκτρονικά, και για καλό λόγο. Σε αντίθεση με προηγούμενες προγραμματιστικές πλακέτες κυκλωμάτων, το Arduino δεν χρειάζεται ένα ξεχωριστό κομμάτι του υλικού (που ονομάζεται προγραμματιστής) προκειμένου να φορτώσει νέο κώδικα πάνω στη προγραμματιστική πλακέτα. Μπορείτε απλά να χρησιμοποιήσετε ένα καλώδιο USB. Επιπλέον, το Arduino IDE χρησιμοποιεί μια απλοποιημένη έκδοση της C ++, πράγμα που καθιστά ευκολότερο να τον μάθουμε και να τον προγραμματίσουμε. Τέλος, το Arduino προβλέπει μια τυποποιημένη μορφή φόρμα η οποία παρέχει τις λειτουργίες του μικροελεγκτή σε ένα πιο προσιτό πακέτο.

Με λίγα λόγια, το **Arduino** είναι ένας single-board μικροελεγκτής, δηλαδή μια απλή μητρική πλακέτα ανοικτού κώδικα, με ενσωματωμένο μικροελεγκτή και εισόδους/εξόδους, και η οποία μπορεί να προγραμματιστεί με τη γλώσσα Wiring (ουσιαστικά πρόκειται για τη γλώσσα προγραμματισμού C++ και ένα σύνολο από βιβλιοθήκες, υλοποιημένες επίσης στην C++). Το Arduino μπορεί να χρησιμοποιηθεί για την ανάπτυξη ανεξάρτητων διαδραστικών αντικειμένων αλλά και να συνδεθεί με υπολογιστή μέσω προγραμμάτων σε Processing, Max/MSP, PureData, Supercollider. Οι περισσότερες εκδόσεις του Arduino μπορούν να αγοραστούν προ-συναρμολογημένες· το διάγραμμα και πληροφορίες για το υλικό είναι ελεύθερα διαθέσιμα για αυτούς που θέλουν να συναρμολογήσουν το Arduino μόνοι τους [2].



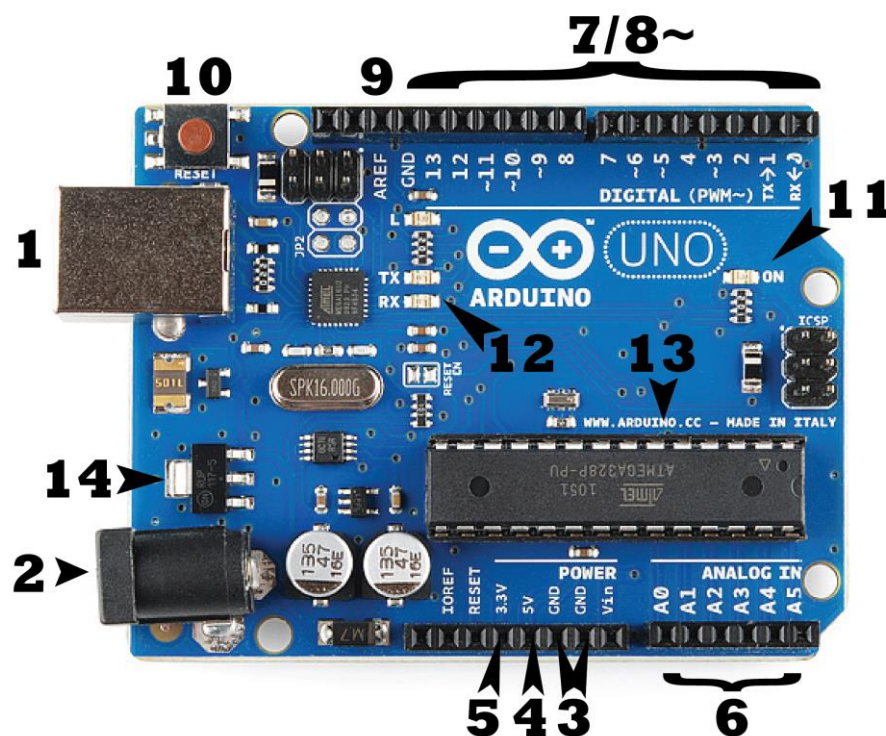
Εικόνα 2 - 2 : Πλακέτα Arduino Uno

2.3 Τι μπορεί να κάνει

Το υλικό και το λογισμικό του Arduino έχει σχεδιαστεί για τους καλλιτέχνες, σχεδιαστές, χομπίστες, χάκερ, αρχάριους και γενικά όσους ενδιαφέρονται για τη δημιουργία διαδραστικών αντικειμένων ή περιβάλλον. Το Arduino μπορεί να αλληλοεπιδρά με κουμπιά, λυχνίες LED, κινητήρες(motors), ηχεία, συσκευές GPS, φωτογραφικές μηχανές(cameras), το Διαδίκτυο, ακόμη και με smart-phone ή την τηλεόρασή σας! Αυτή η ευελιξία σε συνδυασμό με το γεγονός ότι το λογισμικό Arduino είναι δωρεάν, και η πλακέτα με το κύκλωμά είναι πολύ φθηνή, και τόσο το λογισμικό όσο και το υλικό είναι εύκολο να τα μάθουμε οδήγησε σε μια μεγάλη κοινότητα χρηστών που έχουν συνεισφέρει σε κώδικα και έχουν κυκλοφορήσει οδηγίες για μια τεράστια ποικιλία προγραμμάτων στον Arduino. Για τα πάντα, από ένα απλό ρομπότ μέχρι μία εξεζητημένη μηχανή, το Arduino μπορεί να χρησιμοποιηθεί ως ο εγκέφαλος πίσω από το κάθε ηλεκτρονικό έργο [3].

2.4 Πως είναι το board του Arduino

Υπάρχουν πολλές ποικιλίες από πλακέτες Arduino που μπορούν να χρησιμοποιηθούν για διαφορετικούς σκοπούς. Μερικές πλακέτες είναι λίγο διαφορετικές από τη παρακάτω, αλλά οι περισσότεροι Arduinos έχουν στην πλειοψηφία τους τα παρακάτω κοινά χαρακτηριστικά [3].



Εικόνα 2 - 3 : Τι βρίσκεται στην πλακέτα Arduino

Power (USB / BarrelJack) : Κάθε πλακέτα Arduino χρειάζεται έναν τρόπο να συνδεθεί με μια πηγή ρεύματος. Ο ArduinoUNO μπορεί να τροφοδοτείται από ένα καλώδιο USB που προέρχεται από τον υπολογιστή σας ή ένα τροφοδοτικό τοίχου (όπως αυτό) που τερματίζεται σε μια υποδοχή . Στη φωτογραφία παρακάτω η σύνδεση με USB

είναι σημειωμένη με **(1)** και η υποδοχή που συνδέεται με το μετασχηματιστή έχει επισημανθεί με **(2)**.

Η σύνδεση USB επίσης χρησιμοποιείται για να φορτώσετε το κώδικα στην πλακέτα του Arduino σας. Περισσότερα για το πως να προγραμματίσετε ένα Arduino θα σας δοθούν σε επόμενο κεφάλαιο.

ΣΗΜΕΙΩΣΗ: Μην χρησιμοποιείτε τροφοδοτικό με μεγαλύτερη από 20 Volts τάση, καθώς θα εξουδετερώσει (και ως εκ τούτου να καταστρέψει) τον Arduino σας. Η συνιστάμενη τάση για τα περισσότερα μοντέλα Arduino είναι μεταξύ 6 και 12 Volts.

Ακίδες-Pins (5V, 3.3V, GND, Analog, Digital, PWM, AREF) : Οι ακίδες στο Arduino σας είναι τα μέρη όπου μπορείτε συνδέστε τα καλώδια για να κατασκευάσει ένα κύκλωμα .Πιθανώς σε συνδυασμό με μια ηλεκτρονική πλακέτα(breadboard)και με κάποια καλώδια. Συνήθως έχουν μαύρο πλαστικό "κεφαλίδες» που σας επιτρέπουν να συνδέσετε μόνο ένα καλώδιο στη κάθε εσοχή της πλακέτας. Το Arduinoέχει αρκετές διαφορετικά είδη ακίδων, κάθε ένα από τα οποία έχει μία ετικέτα πάνω στην πλακέτα και χρησιμοποιείται για διάφορες λειτουργίες.

- **GND (3)**: Συντόμευση για γείωση(Ground). Υπάρχουν πολλές ακίδες GND στον Arduino, οποιαδήποτε από αυτές μπορεί να χρησιμοποιηθεί για τη γείωση του κυκλώματος σας.
- **5V (4) & 3.3V (5)**: Όπως μπορείτε να μαντέψετε, η ακίδα 5V αναφέρεται σε ηλεκτρική τάση των 5 volts ,καθώς και η ακίδα 3.3V αναφέρεται σε ηλεκτρική τάση των 3.3 volts. Τα περισσότερα από τα απλά εξαρτήματα που συνδέονται με το Arduino χρησιμοποιούν τάση 5 ή 3,3 volts .
- **Analog (6)**: Η περιοχή των ακίδων με την ετικέτα "Αναλογική" (δηλαδή από την ακίδα A0 έως A5 στο UNO) είναι αναλογικές ακίδες. Αυτές οι ακίδες μπορούν να διαβάσουν το σήμα από έναν αναλογικό αισθητήρα (όπως είναι ένας αισθητήρας θερμοκρασίας) και να το μετατρέψουν αυτό σε μια ψηφιακή τιμή που θα μπορείτε να διαβάσετε.
- **Digital (7)**: Απέναντι από τις αναλογικές ακίδες είναι οι ψηφιακές ακίδες (από 0 έως 13 στον UNO).Αυτές οι ακίδες μπορούν να χρησιμοποιηθούν εξίσου για μια ψηφιακή είσοδο (όπως η ένδειξη ότι ένα κουμπί είναι πατημένο) όσο και ως ψηφιακή έξοδο (για τη τροφοδότηση ενός LED)

- **PWM (8)**: Μπορεί να έχετε παρατηρήσει την περισπωμένη (~) δίπλα σε μερικές από τις ψηφιακές ακίδες (3, 5, 6, 9, 10, και 11 στον UNO). Αυτές οι ακίδες δρουν ως απλές ψηφιακές ακίδες, αλλά μπορούν να χρησιμοποιηθούν και ως Παλμός-Πλάτους Διαμόρφωσης (Pulse-Width Modulation)
- **AREF (9)**: Αντιπροσωπεύει μια Αναλογική αναφορά (Analog Reference). Μερικές φορές που χρησιμοποιείται για να ορίσετε μια εξωτερική τάση αναφοράς (μεταξύ 0 και 5 Volts) ή το ανώτατο όριο για τις ακίδες σε μια αναλογική είσοδο

ResetButton : Ακριβώς όπως και το πρωτότυπο της Nintendo, το Arduino έχει ένα κουμπί επαναφοράς (**10**). Πιέζοντας το κουμπί, αυτό θα συνδέσει προσωρινά την ακίδα επαναφοράς με τη γείωση (GRN) και θα επανεκκινήσει οποιοδήποτε κώδικα έχετε φορτώσει στον Arduino. Αυτό μπορεί να είναι πολύ χρήσιμο αν ο κώδικας σας δεν επαναλαμβάνεται, αλλά εσείς θέλετε να το δοκιμάσετε πολλές φορές. Σε αντίθεση με το πρωτότυπο Nintendo, η επανεκκίνηση στον Arduino συνήθως δεν διορθώνει πολλά προβλήματα.

PowerLEDIndicator : Ακριβώς από κάτω και στα δεξιά της λέξης "UNO" στο κύκλωμα της πλακέτας σας, υπάρχει ένα μικροσκοπικό LED δίπλα στη λέξη 'ON' (**11**). Αυτό το LED πρέπει να ανάβει κάθε φορά που θα συνδέσετε το Arduino σας σε μια πηγή ενέργειας. Αν αυτό το φως δεν ανάβει, υπάρχει μια καλή πιθανότητα κάτι να είναι λάθος. Ώρα να ελέγξετε εκ νέου το κύκλωμα σας!

TXRXLEDs : Το TX είναι η συντόμευση για τη μετάδοση (transmit) , και το RX είναι μια συντόμευση για τη λήψη (receive). Αυτές οι επισημάνσεις εμφανίζονται αρκετά στα ηλεκτρονικά για να υποδεικνύουν τις ακίδες που είναι υπεύθυνες για την σειριακή επικοινωνία. Στη δική μας περίπτωση, υπάρχουν δύο σημεία στον Arduino UNO που εμφανίζονται τα TX και RX - μία φορά στις ψηφιακές ακίδες 0 και 1, και μια δεύτερη φορά δίπλα στα LEDs δείκτης TX και RX (**12**). Αυτά τα LEDs θα σας δώσουν ενδείξεις για το πότε το Arduino στέλνει ή δέχεται δεδομένα (όπως όταν φορτώνετε ένα καινούριο πρόγραμμα στην πλακέτα).

MainIC : Το μαύρο πράγμα με όλα τα μεταλλικά πόδια είναι ένα IC (Integrated Circuit) ή ολοκληρωμένο κυκλώματος (**13**). Σκεφτείτε το σαν τον εγκέφαλο του Arduino σας. Το κεντρικό ολοκληρωμένο κύκλωμα είναι διαφορετικό από πλακέτα σε πλακέτα, αλλά ανήκει συνήθως σε μια από τις 2 αυτές εταιρίες την ATmega ή την ATMEL. Αυτό

μπορεί να είναι σημαντικό, εφόσον χρειαστείτε να γνωρίζετε τον τύπο IC (Μαζί με τον τύπο της πλακέτας σας) πριν από τη φόρτωση ενός νέου προγράμματος από το λογισμικό Arduino. Αυτή η πληροφορία μπορεί συνήθως να βρίσκεται στην επάνω πλευρά του IC. Αν θέλετε να μάθετε περισσότερα σχετικά με τη διαφορά μεταξύ των διαφόρων IC's, διαβάζοντας τα δελτία είναι συχνά μια καλή ιδέα

VoltageRegulator

Ο ρυθμιστής τάσης **(14)** δεν είναι πραγματικά κάτι που μπορείτε να αλληλοεπιδράσετε στο Arduino. Αλλά είναι δυνητικά χρήσιμο να ξέρετε ότι είναι εκεί και γιατί είναι εκεί. Ο ρυθμιστής τάσης δεν κάνει ακριβώς αυτό που λέει. Απλά ελέγχει το ποσό της τάσης που υπάρχει στην πλακέτα Arduino. Σκεφτείτε το ως ένα είδος φύλακα. Θα αποκόψει μια επιπλέον τάση η οποία θα μπορούσε να βλάψει το κύκλωμα. Φυσικά, αυτό έχει τα όριά του, έτσι ώστε να μην συνδέεται το Arduino σας σε οποιαδήποτε τάση μεγαλύτερη των 20 volts [3].

2.5 Γιατί να χρησιμοποιήσω έναν Arduino

Υπάρχουν πολλοί άλλοι μικροελεγκτές και πλατφόρμες μικροελεγκτή διαθέσιμα για τη Φυσικό Προγραμματισμό. Parallax Basic Stamp, NETMEDIA του BX-24, Phidgets, Handyboard του MIT, και πολλοί άλλοι προσφέρουν παρόμοια λειτουργικότητα. Όλα αυτά τα εργαλεία παίρνουν τις πιο δύσκολες λεπτομέρειες του προγραμματισμού του μικροελεγκτή και τα τυλίγουν σε ένα εύκολο πακέτο για το χρήστη. Επίσης το Arduino απλοποιεί τη διαδικασία της εργασίας με τους μικροελεγκτές, αλλά προσφέρει κάποιο πλεονέκτημα για τους εκπαιδευτικούς, μαθητές και άλλα ενδιαφερόμενους ερασιτέχνες σε σχέση με τα συστήματα:

- **Ανέξοδο** - Η πλακέτα Arduino είναι σχετικά φθηνή σε σύγκριση με άλλες πλατφόρμα μικροελεγκτή. Η πιο φθηνή έκδοση της πλακέτας Arduino μπορεί να συναρμολογηθεί με το χέρι, και ακόμη και οι πιο προ-συναρμολογημένες πλακέτες Arduino κοστίζουν λιγότερο από 50 \$.

- **Cross-platform** (εφαρμογή σε πολλές πλατφόρμες) - Το λογισμικό Arduino τρέχει σε Windows, Macintosh OSX, και στα λειτουργικά συστήματα Linux. Τα περισσότερα συστήματα μικροελεγκτή περιορίζονται μόνο στα Windows.
- **Απλό προγραμματιστικό περιβάλλον** - Το προγραμματιστικό περιβάλλον του Arduino είναι εύκολο στη χρήση για αρχάριους, αλλά και αρκετά ευέλικτη για προχωρημένους χρήστες ώστε να επωφεληθούν από αυτό. Για τους εκπαιδευτικούς, αυτό είναι βολικά βασισμένο στο προγραμματιστικό περιβάλλον επεξεργασίας, έτσι ώστε οι μαθητές που μαθαίνουν να προγραμματίζουν σε αυτό περιβάλλον να είναι εξοικειωμένοι με την εμφάνιση και την χρήση του Arduino.
- **Ανοιχτού κώδικα(open source) και επεκτάσιμο Λογισμικό** - Το λογισμικό Arduino δημοσιεύεται ως ανοιχτού κώδικα ,που είναι διαθέσιμο για επέκταση από έμπειρους προγραμματιστές. Η γλώσσα μπορεί να επεκταθεί και με τη χρήση C ++ βιβλιοθηκών, και τους ανθρώπους που θέλουν να κατανοήσουν τις τεχνικές λεπτομέρειες μπορούν να κάνουν το άλμα από το Arduino στη γλώσσα προγραμματισμού AVR C που είναι βασισμένο. Ομοίως, μπορείτε να προσθέσετε τον AVR-C κώδικα απευθείας στα προγράμματα σας στον Arduino εάν το θέλετε.
- **Ανοιχτού κώδικα και επεκτάσιμο υλικό** - Το Arduino βασίζεται στους ATMEGA8 και ATMEGA168 μικροελεγκτές της Atmel. Τα σχέδια για αυτές τις ενότητες δημοσιεύονται με άδεια της Creative Commons, έτσι έμπειροι σχεδιαστές κυκλωμάτων να φτιάχνουν τις δικές τους εκδόσεις για τις πλακέτες, συμβάλλοντας στην επέκταση και τη βελτίωσή τους. Ακόμα και σχετικά άπειροι χρήστες μπορούν να αναπτύξουν την πλακέτα προκειμένου να κατανοήσουν πώς λειτουργεί και να εξοικονομήσουν χρήματα [2].

2.6 Το περιβάλλον ανάπτυξης του Arduino

Το περιβάλλον ανάπτυξης του Arduino περιλαμβάνει έναν επεξεργαστή κειμένου(text editor) για τη σύνταξη κώδικα, μια περιοχή μηνυμάτων , μια κονσόλα κειμένου, μια γραμμή εργαλείων με κουμπιά για κοινές λειτουργίες, καθώς και μια σειρά από μενού. Αυτό συνδέεται με το υλικό Arduino για να φορτώσετε τα προγράμματα και να επικοινωνήσετε με αυτά.

2.7 Γράφοντας προγράμματα στον Arduino(Sketches)

Το λογισμικό που χρησιμοποιείτε για το Arduino ονομάζεται sketches. Τα sketches γράφονται στο πρόγραμμα επεξεργασίας κειμένου. Τα sketches αποθηκεύονται με την επέκταση αρχείου .ino. Έχει τα χαρακτηριστικά γνωρίσματα για την cutting/pasting και για searching/replacing κειμένου. Η περιοχή μηνυμάτων ή λαθών σας δείχνει τα σφάλματα κατά την αποθήκευση ή την εξαγωγή του προγράμματος. Η κονσόλα εμφανίζει κείμενο εξόδου από το περιβάλλον του Arduino συμπεριλαμβάνει πλήρη μηνύματα σφάλματος και άλλες πληροφορίες. Η κάτω δεξιά γωνία του παραθύρου εμφανίζει τη συγκεκριμένη πλακέτα που χρησιμοποιείται και τη σειριακή θύρα(Serial port). Τα κουμπιά της γραμμής εργαλείων σας επιτρέπουν να μπορείτε να ελέγχετε και να φορτώσετε τα προγράμματα σας, να δημιουργήσετε, να ανοίξετε και να αποθηκεύσετε sketches, και να ανοίξει το σειριακή οθόνη(Serial monitor).



Verify

Ελέγχει τον κώδικα σας αν έχει σφάλματα(errors).



Upload

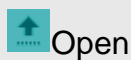
Συντάσσει(Compile) τον κωδικό σας και το ανεβάζει στη πλακέτα Arduino I / O σας.

Σημείωση: Εάν χρησιμοποιείτε μια εξωτερικό προγραμματιστή μπορείτε να κρατήσετε πατημένο το πλήκτρο "Shift" στον υπολογιστή σας όποτε χρησιμοποιείτε αυτό το εικονίδιο. Το κείμενο θα αλλάξει σε "Upload using Programmer"(Ανεβάστε με τη χρήση Προγραμματιστή)".



New

Δημιούργει ένα καινούριο sketch



Open

Παρουσιάζει ένα μενού με όλα τα sketches(προγράμματα)στο λογισμικό του Arduino σας. Κάνοντας κλικ σε ένα από

αυτά θα το ανοίξει στο τρέχων παράθυρο.



Save

Αποθηκεύει το πρόγραμμα σας



SerialMonitor

Ανοίγει το Serialmonitor(Σειριακή οθόνη)

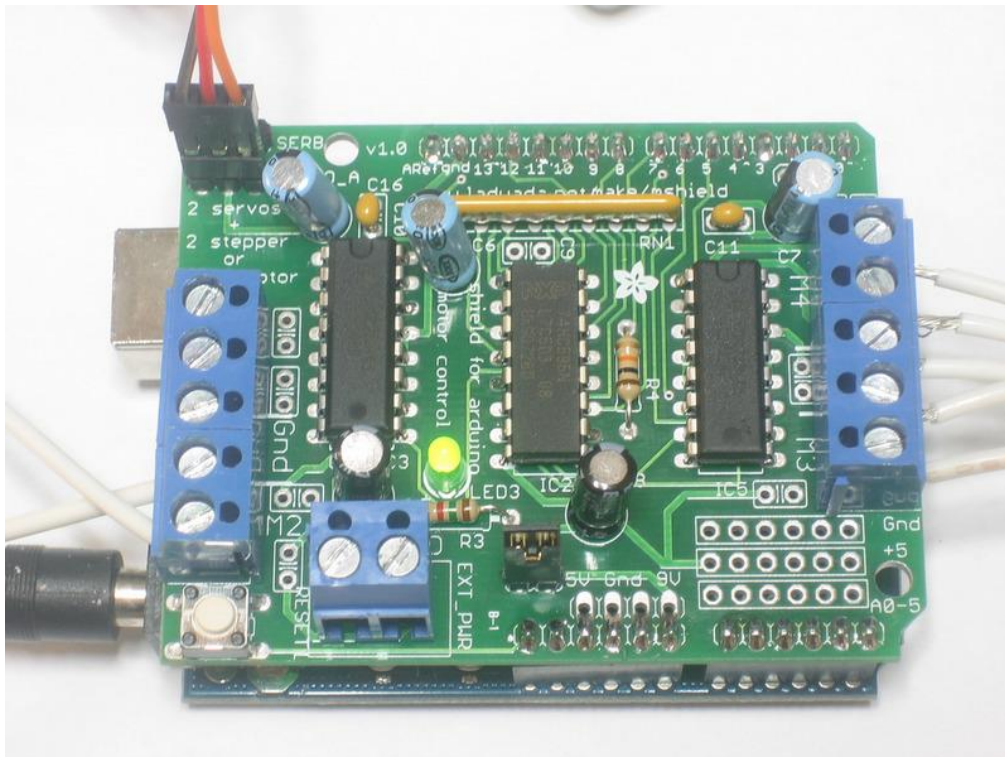
Πίνακας 2 - 2: Κουμπιά Γραμμής Εργαλείων

Το Arduino χρησιμοποιεί την έννοια του sketchbook(ένα μέρος για να αποθηκεύετε τα τυποποιημένα προγράμματα σας (sketches). Τα προγράμματα στο sketchbook σας μπορείτε να τα ανοίξετε από το File > Sketchbook ή από το κουμπί Άνοιγμα(open) στη γραμμή εργαλείων. Η πρώτη φορά που θα εκτελέσετε το λογισμικό Arduino, αυτό θα δημιουργήσει αυτόματα ένα φάκελο για sketchbook σας. Μπορείτε να δείτε ή να αλλάξετε τη θέση του sketchbook από το παράθυρο διαλόγου Προτιμήσεις(Preferences) [4].

ΚΕΦΑΛΑΙΟ 3 : MOTORSHIELD

3.1 Εισαγωγή

Το MotorShield είναι μια τέλεια πλατφόρμα για τη ρομποτική και μηχανολογικές εφαρμογές. Επιτρέπει το Arduino να οδηγήσουν δυο κινητήρες συνεχούς ρεύματος ή ένα βηματικό κινητήρα τεσσάρων καλωδίων και δυο φάσεων. Βασίζεται στο ολοκληρωμένο κύκλωμα του χιπδιπλής γέφυρας L298N , απαιτεί μια 6V ή 15V παροχή ρεύματος για την τροφοδοσία του κινητήρα και περιλαμβάνει επίσης ένα ενσωματωμένο ρυθμιστή τάσης 5V για την τροφοδοσία της κύριας πλακέτας του Arduino. Πρόσθετοι plug-and-play αισθητήρες μπορούν να συνδεθούν στις 13 υποδοχές του [5].



Εικόνα 3 - 1 : Motorshield

3.2 Τι είναι Motor shield κι από τι αποτελείται

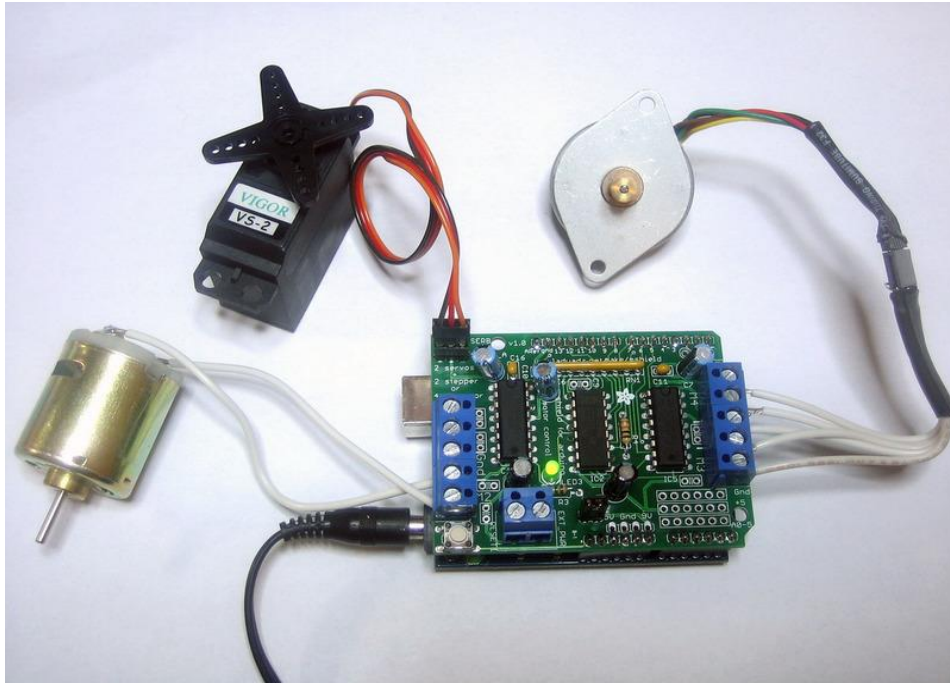
Το αρχικό σετ AdafruitMotorshield είναι μία από τις πιο αγαπημένες εξαρτήσεις μας, η οποία είναι ο λόγος που αποφάσισαν να κάνουν κάτι ακόμα καλύτερο. Έχουν αναβαθμίσει τη πλακέτα για να κάνουν το καλύτερο, πιο εύκολος τρόπος πλέον για να οδηγείτε συνεχούς ρεύματος και βηματικούς κινητήρες. Αυτή η πλακέτα θα κάνει πιο γρήγορη την υλοποίηση στο επόμενο έργο ρομποτικής σας! Κράτησαν την ικανότητα οδήγησης μέχρι και 4 κινητήρων συνεχούς ρεύματος ή 2 βηματικούς κινητήρες, αλλά προσθέσαμε πολλές βελτιώσεις:

Αντί ενός οδηγού L293D Ντάρλινγκτον, έχουμε τώρα τη TB6612 οδηγό MOSFET: με 1.2A ανά κανάλι και κορυφή 3A τρέχουσα ικανότητα. Επίσης, έχει πολύ μικρότερη πτώση δυναμικού του κινητήρα, ώστε να πάρει περισσότερη ροπή από τις μπαταρίες σας, και υπάρχουν ενσωματωμένες δίοδοι κίνησης επιστροφής.

Αντί να χρησιμοποιεί ένα μάνταλο και τις ακίδες του PWM του Arduino, έχουν ένα πλήρως αφιερωμένο τσιπ οδήγησης PWM επί της πλακέτας. Αυτό το τσιπ χειρίζεται όλα τα μοτέρ και την ταχύτητα ελέγχου πάνω I2C. Μόνο δύο ακίδες (SDA και SCL) απαιτούνται για την κίνηση των πολλαπλών κινητήρων, και δεδομένου ότι είναι I2C μπορείτε επίσης να συνδέσετε οποιαδήποτε άλλη συσκευή I2C ή ασπίδες για τις ίδιες ακίδες. Αυτό το καθιστά επίσης συμβατό με οποιοδήποτε Arduino, όπως το Uno, Leonardo, Due και Mega R3.

Πολλές άλλες μικρές βελτιώσεις όπως ένα FET προστασίας πολικότητας στις ακίδες τροφοδοσίας και μια μεγάλη έκταση πρωτοτύπων. Και η πλακέτα συναρμολογείται και δοκιμάζεται στο Adafruit έτσι το μόνο που έχετε να κάνετε είναι να κολλήσετε σε ευθεία ή να στοιβάξετε κεφαλίδες και τα μπλοκ ακροδεκτών.

Εδώ είναι ένα σχέδιο για μια πλήρως εξοπλισμένη πλακέτα κινητήρα που θα είναι σε θέση να τροφοδοτήσει πολλές απλές σε έργα μεσαίας πολυπλοκότητας [6]



Εικόνα 3 - 2 : Μέρη motorshield

Με λίγα λόγια το motorshield αποτελείται από τα παρακάτω:

- 2 συνδέσεις για 5V σερβοκινητήρες που συνδέονται σε έναν υψηλής ανάλυσης timer του Arduino.
- μέχρι 4 αμφίδρομους κινητήρες συνεχούς ρεύματος με ατομική επιλογή ταχύτητας 8-bit
- μέχρι 2 βηματικούς κινητήρες (μονοπολικούς ή διπολικούς) με μοναδικό πηνίο, διπλό πηνίο, διαστρωμένα ή μικρό-βηματικούς.
- 4 γέφυρες: L293D τσιπ που παρέχει 0.6A ανά γέφυρα (1.2A κορυφή) με κλείσιμο θερμικής προστασία , 4.5V σε 25V
- pull down αντιστάσεις που κρατάνε τους κινητήρες απενεργοποιημένους κατά την εκκίνηση.
- μεγάλες υποδοχές ακροδεκτών εύκολα να συνδέσετε τα καλώδια (10-22AWG) και τη τροφοδοσία.
- το κουμπί επαναφοράς του Arduino
- 2 ακίδες ακροδεκτών για να συνδέσετε εξωτερική πηγή ενέργειας για την τροφοδοσία των κινητήρων

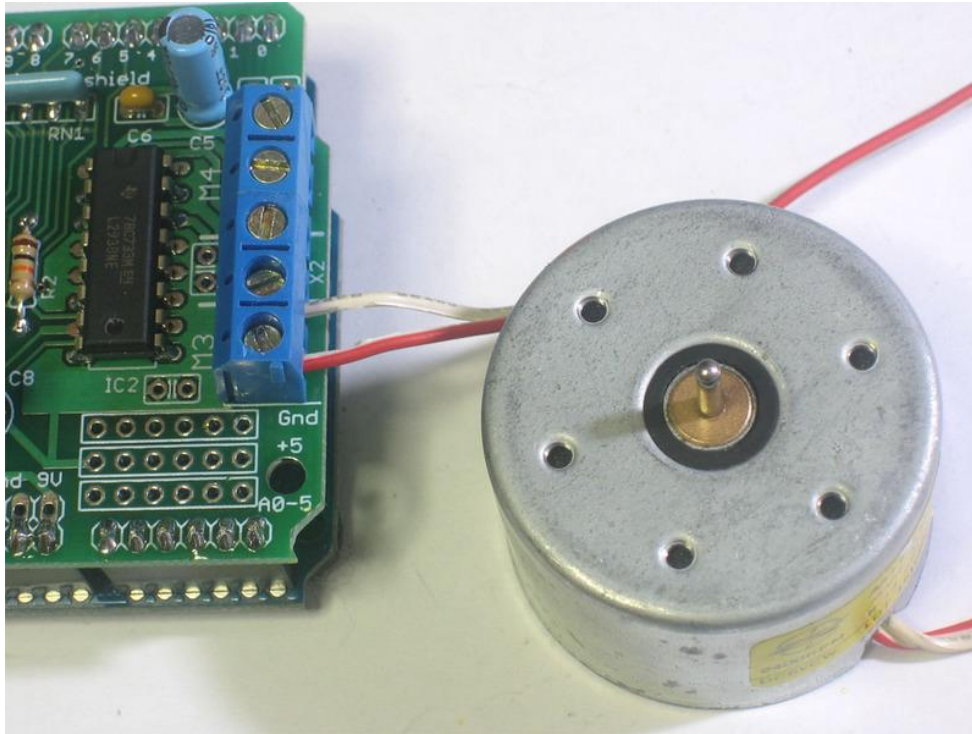
- δοκιμασμένο και συμβατό με το Mega, Diecimila, & Duemilanove

3.3 Τι απαιτείται για να το προγραμματίσετε

Για να μπορέσετε να χρησιμοποιήσετε το Motorshield, θα πρέπει να εγκαταστήσετε τη βιβλιοθήκη AF_Motor του Arduino - αυτό θα δώσει εντολή στον Arduino πώς να μιλήσω με τη πλακέτα Adafruit Motorshield, και αυτό δεν είναι προαιρετικό!

1. Πρώτον, κατεβάστε τη βιβλιοθήκη από το GitHub.
2. Αποσυμπιέστε το αρχείο ZIP στην επιφάνεια εργασίας σας.
3. Μετονομάστε τον αποσυμπιεσμένο φάκελο σε AFMotor.
4. Ελέγξτε ότι μέσα στο φάκελο AFMotor είναι τα αρχεία AFMotor.cpp και AFMotor.h. Εάν όχι, ελέγξτε τα παραπάνω βήματα.
5. Τοποθετήστε το φάκελο AFMotor στο φάκελο σας arduinosketchfolder/ libraries. Για τα Windows, αυτό πιθανότατα θα είναι κάτι σαν MY Documents/Arduino/libraries για τα Mac θα είναι κάτι σαν Documents/arduino/libraries. Αν αυτή είναι η πρώτη φορά που εγκαθιστάτε μια βιβλιοθήκη, θα πρέπει να δημιουργήσετε το φάκελο libraries . Φροντίστε να το ονομάσουμε libraries ακριβώς και όχι κάποιο άλλο όνομα.
6. Ελέγξτε ότι μέσα στο φάκελο libraries υπάρχει ο φάκελος AFMotor, και μέσα στο φάκελο AFMotor υπάρχουν τ αρχεία AFMotor.cpp AFMotor.h και κάποια άλλα αρχεία
7. Κλείστε και κάντε επανεκκίνηση του IDE. Πρέπει τώρα να έχετε ένα υπομενού που ονομάζεται File->Examples->AFMotor->MotorParty

3.4 Χρησιμοποιώντας κινητήρες συνεχούς ρεύματος (dcmotors)



Εικόνα 3 - 3 : Κινητήρας Ρεύματος

Οι κινητήρες συνεχούς ρεύματος χρησιμοποιούνται για κάθε είδους ρομποτικών έργων.

Το Motor shield μπορεί να οδηγήσει μέχρι και 4 κινητήρες συνεχούς ρεύματος αμφίδρομα. Αυτό σημαίνει ότι μπορεί να οδηγείται προς τα εμπρός και προς τα πίσω. Η ταχύτητα μπορεί επίσης να μεταβάλλεται το πολύ 0,5% σε αυξήσεις χρησιμοποιώντας το υψηλής ποιότητας ενσωματωμένο PWM. Αυτό σημαίνει ότι η ταχύτητα είναι πολύ ομαλή και δεν θα μεταβάλλεται! Το τσιπ της γέφυρα δεν προορίζεται για φορτία οδήγησης πάνω από 0.6A ή με κορυφές πάνω από 1.2A, έτσι αυτό είναι για τους μικρούς κινητήρες. Ελέγξτε το φύλλο δεδομένων (datasheet) για πληροφορίες σχετικά με τον κινητήρα για να σιγουρευτείτε ότι είναι OK.

Για να συνδέσετε έναν κινητήρα, απλά κολλήστε τα δύο καλώδια στα τερματικά και στη συνέχεια συνδέστε τα είτε στο **M1**, **M2**, **M3**, ή **M4**. Στη συνέχεια, ακολουθήστε τα παρακάτω βήματα στο sketch σας.

1. Βεβαιωθείτε ότι έχετε κάνει `#include` το `<AFMotor.h>`.
2. Δημιουργήστε το αντικείμενο `AF_DCMotor` με **`AF_DCMotor (motor#,frequency)`**, για την εγκατάσταση του κινητήρα της γέφυρα και των ασφαλειών του. Ο κατασκευαστής λαμβάνει δύο ορίσματα. Το πρώτο είναι ποια θύρα ο κινητήρας είναι συνδεδεμένος με, 1, 2, 3 ή 4. και το δεύτερο η είναι συχνότητα που ορίζει τη ταχύτητα που θα κινηθεί ο κινητήρας. Για τους κινητήρες 1 και 2 μπορείτε να επιλέξετε **`MOTOR12_64KHZ`**, **`MOTOR12_8KHZ`**, **`MOTOR12_2KHZ`**, **or****`MOTOR12_1KHZ`**. Ένα υψηλής ταχύτητας, όπως 64KHz συνηθίζεται, αλλά η χαμηλή ταχύτητα όπως 1KHz θα καταναλώνει λιγότερη ενέργεια. Οι κινητήρες 3 & 4 είναι δυνατό να τρέξουν μόνο στο 1KHz και θα αγνοήσουν οποιαδήποτε ρύθμιση δίνεται.
3. Στη συνέχεια, μπορείτε να ρυθμίσετε την ταχύτητα του κινητήρα χρησιμοποιώντας **`setSpeed (speed)`**, όπου η ταχύτητα κυμαίνεται από 0 (stopped) έως 255 (full speed). Μπορείτε να ρυθμίσετε την ταχύτητα όποτε θέλετε.
4. Για να κινήσετε το κινητήρα, καλέστε **`run(direction)`** όπου κατεύθυνση είναι FORWARD, BACKWARD or RELEASE. Φυσικά, το Arduino πραγματικότητα δεν ξέρει αν ο κινητήρας είναι «προς τα εμπρός» ή «προς τα πίσω», οπότε αν θέλετε να αλλάξει τον τρόπο που σκέφτεται το προς τα εμπρός, απλά να ανταλλάξετε τα δύο καλώδια από τον κινητήρα για την πλακέτα.

ΚΕΦΑΛΑΙΟ 4 : BLUETOOTH

4.1 Εισαγωγή

Το **Bluetooth** είναι ένα βιομηχανικό πρότυπο για ασύρματα προσωπικά δίκτυα υπολογιστών (Wireless Personal Area Networks, WPAN). Πρόκειται για μια ασύρματη τηλεπικοινωνιακή τεχνολογία μικρών αποστάσεων, η οποία μπορεί να μεταδώσει σήματα μέσω μικροκυμάτων σε ψηφιακές συσκευές. Επομένως το Bluetooth είναι ένα πρωτόκολλο το οποίο παρέχει προτυποποιημένη, ασύρματη επικοινωνία ανάμεσα σε PDA, κινητά τηλέφωνα, φορητοί υπολογιστές, προσωπικοί υπολογιστές, εκτυπωτές, καθώς και ψηφιακές φωτογραφικές μηχανές ή ψηφιακές κάμερες, μέσω μιας ασφαλούς, φθηνής και παγκοσμίως διαθέσιμης χωρίς ειδική άδεια ραδιοσυχνότητας μικρής εμβέλειας. Από τεχνικής άποψης το Bluetooth είναι ένα πρωτόκολλο ασύρματης δικτύωσης σε φυσικό επίπεδο, υποεπίπεδο MAC και, προαιρετικά, υποεπίπεδο LLC.

4.2 Ιστορία του Bluetooth

Ως τα τέλη της δεκαετίας του 1990 δεν υπήρχε κάποιο ευρέως αποδεκτό πρότυπο WPAN, ούτε φυσικά ανάλογες εμπορικές εφαρμογές / πομποδέκτες. Όμως τότε η Ericsson έθεσε τις βάσεις για την ανάπτυξη μίας τεχνολογίας η οποία θα επέτρεπε τον σχηματισμό τοπικών δικτύων πολύ μικρής εμβέλειας με σκοπό την ασύρματη και adhoc δικτύωση ετερογενών φορητών συσκευών. Το πρότυπο που προέκυψε υιοθετήθηκε στη συνέχεια από την IEEE ως το πρότυπο 802.15 για WPAN. Οι σχεδιαστές του κλήθηκαν να επιλέξουν το όνομα με το οποίο αυτή η τεχνολογία θα γινόταν αργότερα γνωστή σε όλο τον κόσμο. Οι Σουηδοί εμπνευστές του 802.15 ήταν βέβαιοι ότι το νέο πρότυπο θα επικρατούσε και θα έφερνε ακόμη πιο κοντά τους ανθρώπους και τις συσκευές τους. Οι Σκανδιναβοί μάλιστα διαφημίζουν ότι κάτι ανάλογο έκανε και ο Δανός Βασιλιάς Χάραλντ ο Κυανόδους, ο οποίος έζησε στα τέλη του 10ου αιώνα μ.Χ.: κατέλαβε με τα στρατεύματά του πολλές χώρες, ενώ λέγεται ότι κατάφερε να ενώσει τη Δανία με τη Νορβηγία.

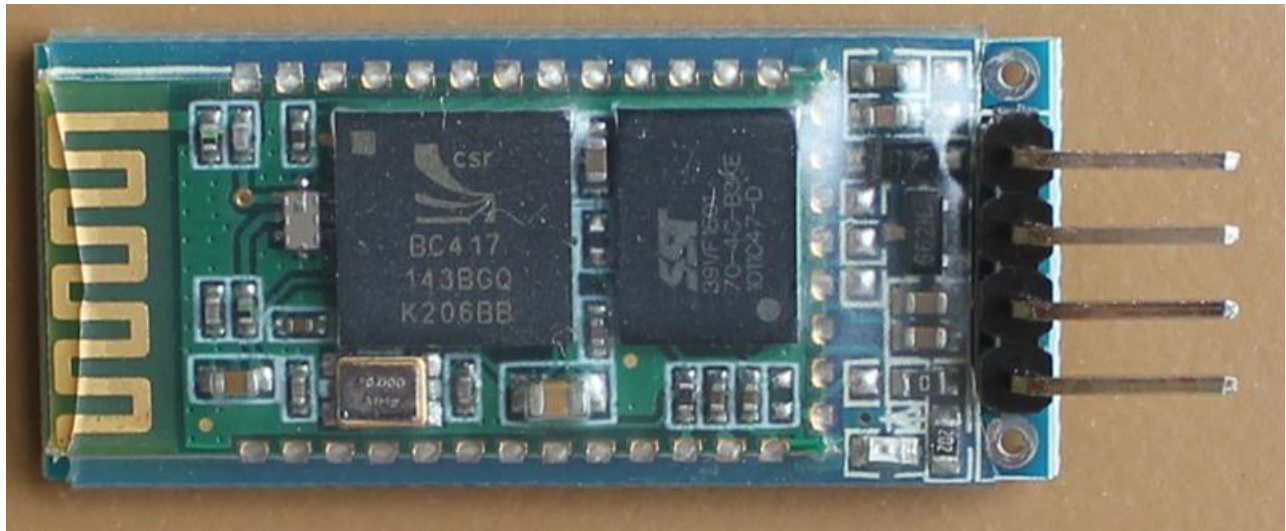
4.3 Εφαρμογές

Το Bluetooth επιτρέπει την κατάργηση όλων των καλωδίων τα οποία παλαιότερα ήταν απαραίτητα για τη «διασύνδεση» μεταξύ υπολογιστών, φορητών υπολογιστών χειρός, κινητών τηλεφώνων και άλλων ψηφιακών συσκευών, όπως ψηφιακές κάμερες, σαρωτές, εκτυπωτές, μικρόφωνα, ακουστικά, ραδιόφωνα κ.α. Το Bluetooth επιτρέπει την σύνδεση του κινητού με τον υπολογιστή, τη μεταφορά δεδομένων, όπως εικόνες, επαφές και σημειώσεις από κινητό προς κινητό, τη σύνδεση στο Internet κ.α. Όλα αυτά χωρίς καλώδια και πολύπλοκες ρυθμίσεις.

Οι εφαρμογές του λοιπόν είναι πολλαπλές:

- Ασύρματη δικτύωση μεταξύ επιτραπέζιου και φορητού υπολογιστή, σε έναν περιορισμένο χώρο με ελάχιστο διαθέσιμο εύρος ζώνης.
- Ασύρματα περιφερειακά, όπως εκτυπωτές, ποντίκια και πληκτρολόγια, τα οποία επικοινωνούν με κάποιον επιτραπέζιο ή φορητό υπολογιστή.
- Ασύρματη μεταφορά ψηφιακών αρχείων (εικόνες, mp3 κλπ.) ανάμεσα σε κινητά τηλέφωνα και PDA.
- Ασύρματα ακουστικά για κινητά τηλέφωνα και Smartphone.
- Ιατρικές εφαρμογές – δοκιμάζονται συσκευές από εταιρίες που παρέχουν ηλεκτρονικές συσκευές προχωρημένης ιατρικής.
- Ορισμένοι δέκτες GPS μεταφέρουν πληροφορίες NMEA μέσω Bluetooth.
- Ασύρματη τηλεφωνία στο αυτοκίνητο: Το Bluetooth δίνει τη δυνατότητα σε χρήστες καταλλήλως εξοπλισμένων κινητών τηλεφώνων να χρησιμοποιούν κάποιες βασικές λειτουργίες τους με ασύρματα ακουστικά. Ανάλογο σύστημα υπάρχει ενσωματωμένο και σε κράνη οδηγών μοτοσικλέτας, επιτρέποντας τη συνομιλία κατά την οδήγηση.
- Απομακρυσμένος έλεγχος συσκευών, όπου έως την εμφάνιση του Bluetooth χρησιμοποιούνταν τεχνολογία υπέρυθρων ακτινών.

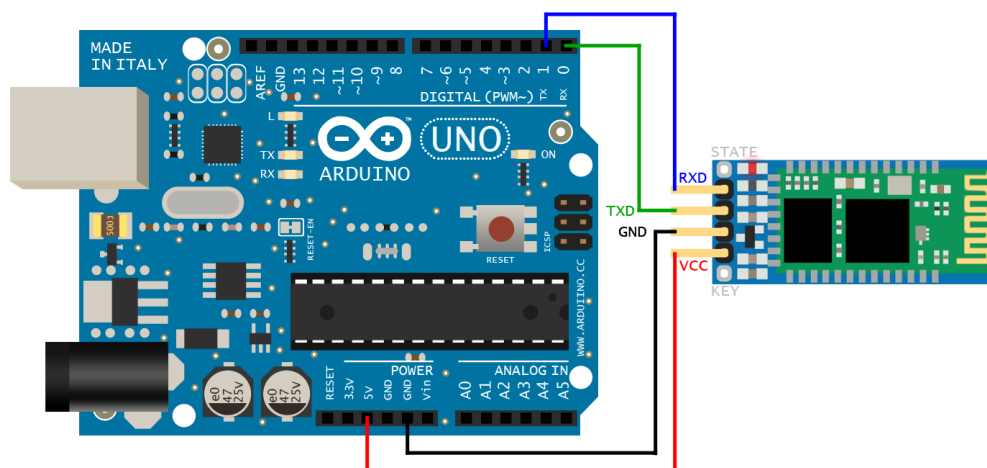
4.4 Γιατί να χρησιμοποιήσω Bluetooth



Εικόνα 4 - 1 : Bluetooth

Αυτό το μικρό τσιπ είναι ένα Bluetooth με σειριακή θύρα που είναι ενσωματωμένη σε φορητές συσκευές όπως αν χρησιμοποιείται ένα καλώδιο τύπου RS232. Είναι τέλειο για μικρής απόστασης ασύρματης επικοινωνίας έργα που απαιτούν σειριακή πρόσβαση από κονσόλα, επειδή μπορείτε να τα αγοράσετε από το eBay για πολύ λίγα χρήματα. Το μόνο δύσκολο κομμάτι είναι να βρείτε πλήρες εγχειρίδιο για το πώς λειτουργούν [7].

4.5 Πως το συνδέεται το Bluetooth στον Arduino και από τι αποτελείται



Εικόνα 4 - 2 : Σύνδεση Arduino με Bluetooth

Το Bluetooth επικοινωνεί με το Arduino μέσω μιας σειριακής σύνδεσης. Έχει τέσσερις ακίδες που θα χρησιμοποιήσετε:

- VCC χρησιμοποιείται για να τροφοδοτήσει τη μονάδα. Θα πρέπει να συνδεθεί με την ακίδα 5v του Arduino.
- GND είναι ο ακροδέκτης γείωσης. Πρέπει να συνδεθεί με την ακίδα γείωσης(GND) του Arduino
- TXD χρησιμοποιείται για την αποστολή δεδομένων από τη μονάδα στο Arduino. Πρέπει να συνδέεται με τη σειριακή ακίδα (RX) του Arduino, η οποία είναι η ακίδα 0 στην περίπτωση του Arduino Uno. Εάν χρησιμοποιείτε μια διαφορετική πλακέτα Arduino, ελέγξτε τις σχηματικές παραστάσεις της για να βεβαιωθείτε ότι έχετε τη σωστή ακίδα.
- RXD χρησιμοποιείται για τη λήψη δεδομένων από το Arduino. Θα πρέπει να συνδεθεί με το σειριακό ακροδέκτη μετάδοσης (TX) του Arduino, το οποίο είναι η ακίδα 1 στην περίπτωση του Arduino Uno.

Οι προεπιλεγμένες ρυθμίσεις του Bluetooth είναι:

- Baudrate: 9600
- Όνομα: linvor
- Κωδικός Σύζευξης: 1234

ΚΕΦΑΛΑΙΟ 5 : ANDROID



Εικόνα 5 - 1 :Λογότυπο Android

5.1 Εισαγωγή στο Android

Το Android είναι ένα λειτουργικό σύστημα για συσκευές κινητής τηλεφωνίας, το οποίο τρέχει τον πυρήνα του λειτουργικού Linux. Είναι μια στοίβα λογισμικού για κινητές συσκευές η οποία περιλαμβάνει λειτουργικό σύστημα, ενδιάμεσο λογισμικό (middleware) και βασικές εφαρμογές. Αρχικά αναπτύχθηκε από την Google και αργότερα από την OpenHandsetAlliance. Επιτρέπει στους κατασκευαστές λογισμικού να γράφουν κώδικα με τη χρήση της γλώσσας προγραμματισμού Java, ελέγχοντας την συσκευή μέσω βιβλιοθηκών λογισμικού ανεπτυγμένων από την Google. Τον Ιούλιο του 2005, η Google αγόρασε την AndroidInc, μια μικρή εταιρεία με έδρα το PaloAlto στην California των ΗΠΑ [8].

Εκείνη την εποχή ελάχιστα ήταν γνωστά για τις λειτουργίες της AndroidInc, εκτός του ότι ανέπτυσαν λογισμικό για κινητά τηλέφωνα. Στην Google, η ομάδα με επικεφαλής τον Rubin ανέπτυξε μια κινητή πλατφόρμα που στηρίζεται στον πυρήνα του Linux, την οποία προώθησαν με την παροχή ενός ευέλικτου, αναβαθμίσιμου συστήματος. Τον Σεπτέμβριο του 2007, η InformationWeek κάλυψε μια μελέτη αξιολόγησης αναφέροντας ότι η Google έχει καταθέσει αρκετές πατέντες στον τομέα της κινητής τηλεφωνίας. Τελικά η Google παρουσίασε το smartphone της NexusOne που χρησιμοποιεί το λειτουργικό σύστημα Android. Η συσκευή κατασκευάστηκε από την HTC Corporation της Ταϊβάν, και έγινε διαθέσιμη στις 5 Ιανουαρίου 2010. Η πρώτη παρουσίαση της πλατφόρμας Android

έγινε στις 5 Νοεμβρίου 2007, παράλληλα με την ανακοίνωση της ίδρυσης του οργανισμού OpenHandsetAlliance, μιας κοινοπραξίας 48 τηλεπικοινωνιακών εταιριών, εταιριών λογισμικού καθώς και κατασκευής hardware, οι οποίες είναι αφιερωμένες στην ανάπτυξη και εξέλιξη ανοιχτών προτύπων στις συσκευές κινητής τηλεφωνίας. Η Google δημοσίευσε το μεγαλύτερο μέρος του κώδικα του Android υπό τους όρους της ApacheLicense, μιας ελεύθερης άδειας λογισμικού.

Μερικές από τις εταιρίες αυτές αναγράφονται παρακάτω:

- Intel
- Vodafone
- Motorola
- Samsung
- Sony Ericsson
- eBay
- Google
- Verizon
- Texas Instruments
- HTC

5.2 Η αρχιτεκτονική του Android



Εικόνα 5 - 2 : Επίπεδα Στοιβάς Android

Το Android είναι μια στοίβα λογισμικού. Η λογική πίσω από αυτή την έκφραση και σε όλη την φιλοσοφία του Android, κρύβεται στο ακόλουθο διάγραμμα με τα βασικά συστατικά του. Η αρχιτεκτονική του Android ενθαρρύνει την έννοια επαναχρησιμοποίησης κώδικα. Έτσι γίνεται δυνατή η επέκταση και βελτιστοποίηση υπαρχόντων εφαρμογών ή δημιουργίας καινούργιων, χρησιμοποιώντας τον ίδιο κωδικό και κάνοντας κάποια επέκτασή του αναλόγως την εφαρμογή που θέλουμε να δημιουργήσουμε. Οι ακόλουθες υπηρεσίες εφαρμογών είναι οι πιο σημαντικές στην αρχιτεκτονική όλων των εφαρμογών Android :

- Διαχειριστής Δραστηριοτήτων (ActivityManager) - Ελέγχει τον κύκλο ζωής των δραστηριοτήτων.

- Όψεις (Views) - Χρησιμοποιούνται για να κατασκευάσουν διεπαφές χρήστη (userinterface) για τις δραστηριότητες (activities).
- Διαχειριστή Ειδοποιήσεων (NotificationManager) - Παρέχει ένα συνεπή και αποτελεσματικό μηχανισμό για να προειδοποιεί τον χρήστη.
- Παροχέας Περιεχομένου (ContentProviders) - Επιτρέπουν στις εφαρμογές να μοιραστούν δεδομένα μεταξύ τους.
- Διαχειριστής Πόρων (ResourceManager) - Υποστηρίζει πόρους εκτός κώδικα όπως γραφικά [9].

5.3 Ο πυρήνας του Android

Το Android είναι βασισμένο στα θεμέλια του Linux. Ο πυρήνας Linux είναι δοκιμασμένος, σταθερός και πετυχημένος και μπορεί να βρεθεί παντού. Το Linux παρέχει στο Android το αφαιρετικό επίπεδο υλικού, επιτρέποντας του να μπορεί να χρησιμοποιηθεί σε μεγάλη ποικιλία πλατφορμών στο μέλλον. Ειδικότερα, το Android χρησιμοποιεί τον πυρήνα Linux για την διαχείριση μνείας, την διαχείριση διεργασιών, την δικτύωση και άλλες υπηρεσίες του λειτουργικού συστήματος. Ο πυρήνας του Android περιέχει: οδηγό Wi-Fi και Bluetooth, οδηγό προβολής οθόνης, οδηγό κάμερας. Ο πυρήνας του Android βασίζεται στο πυρήνα Linux αλλά συνάμα έχει διαφορές και από αυτόν. Είναι αρκετά δύσκολο να τρέξουν εφαρμογές (applications) από τη μια πλατφόρμα στην άλλη.

5.4 Η ανάπτυξη του Android και οι εκδόσεις του



Εικόνα 5 - 3 : Εκδόσεις Android

Η δυνατότητα ή όχι της αναβάθμισης σε νεότερη έκδοση android είναι καθαρά θέμα κατασκευαστή, αφού η «καθαρή» έκδοση, πρέπει να προσαρμοστεί από τον εκάστοτε κατασκευαστή για τις συσκευές του. Μέχρι τώρα υπάρχουν 8 επιβεβαιωμένες εκδόσεις Android. Η πρώτη έκδοση του AndroidSDK που εμφανίστηκε τον Νοέμβριο του 2007, χαρακτηρίστηκε από τους κατασκευαστές του σαν μια πρώτη ματιά στο SDK του Android, κάτι το οποίο πολλοί παράλειψαν και βιάστηκαν να κατακρίνουν το Android σαν ένα προβληματικό σύστημα. Στην ουσία όμως το Android παρουσίαζε προβλήματα τα οποία παρουσιάζει οποιοδήποτε σύστημα σε τέτοιο πρώιμο στάδιο. Έτσι το Σεπτέμβριο του 2008, η T-Mobile ανακοινώνει την διαθεσιμότητα του T-MobileG1, του πρώτου έξυπνου τηλεφώνου (smartphone), βασισμένο στην πλατφόρμα του Android. Λίγες μέρες αργότερα (Οκτώβριο 2008), η Google ανακοινώνει την απελευθέρωση του SDKReleaseCandidate 1.0. Ακολούθησε τον Φεβρουάριο του 2009 η έκδοση 1.1 σαν μια ανανεωμένη έκδοση του 1.0. Μέχρι τότε το Android δεν υποστήριζε ακόμη την χρήση κουμπιών αφής.

5.4.1 Android 1.5 CUPCAKE



Εικόνα 5 - 4 : Android 1.5 Cupcake

Τον Μάιο του 2009 εμφανίστηκε η έκδοση Android 1.5, επωνομαζόμενη 'Cupcake', η οποία εισάγει κάποια καινούργια χαρακτηριστικά στην διεπαφή του χρήστη.

User Interface:

- Ικανότητα για καταγραφή και παρακολούθηση βίντεο μέσα από την λειτουργία της βιντεοκάμερας, μεταφόρτωση βίντεο στο YouTube και φωτογραφιών στο Picasa απευθείας από το τηλέφωνο, καινούργιο μαλακό πληκτρολόγιο (αφής) με πρόβλεψη κειμένου.
- Υποστήριξη προτύπου BluetoothA2DP και AVRCP.
- Ικανότητα αυτόματης σύνδεσης σε μικροσυσκευή Bluetooth από μια συγκεκριμένη απόσταση.
- Καινούργια widgets και φάκελοι που μπορούν να δημοσιευτούν στην αρχική οθόνη.
- Κινούμενες μεταβάσεις οθόνης.

5.4.2 Android 1.6 DONUT



Donut
Android 1.6

Εικόνα 5 - 5 : Android 1.6 DONUT

Το 'Donut', Android 1.6, ήρθε τον Σεπτέμβριο του 2009. Η έκδοση αυτή εισάγει κάποια καινούργια χαρακτηριστικά όπως:

- Βελτιωμένο Android Market.
- Ενσωματωμένη φωτογραφική μηχανή, βιντεοκάμερα και διεπαφή (interface) γκαλερί.
- Η γκαλερί επιτρέπει πλέον στους χρήστες την επιλογή πολλαπλών φωτογραφιών προς διαγραφή.
- Ανανεωμένη φωνητική αναζήτηση, με ταχύτερη απόκριση και βαθύτερη ολοκλήρωση με εγγενείς (native) εφαρμογές, συμπεριλαμβανομένης της δυνατότητας κλήσης επαφών.
- Ανανεωμένη αναζήτηση με την δυνατότητα αναζήτησης σελιδοδεικτών, ιστορικού, επαφών και στο διαδίκτυο από την αρχική οθόνη.
- Ανανεωμένη υποστήριξη τεχνολογιών για CDMA/EVDO, 802.1x, VPNs και με μηχανή μετατροπής κειμένου σε ομιλία (text-to-speech).
- Υποστήριξη για ανάλυση οθονών WVGA.
- Βελτιώσεις στην ταχύτητα αναζήτησης και των εφαρμογών της φωτογραφικής μηχανής.

5.4.3 Android 2.0 ECLAIR



Εικόνα 5 - 6 : Android 2.0 ECLAIR

Ακολουθεί το 'Eclair' Android 2.0 τον Νοέμβριο 2009, με τις επανεκδόσεις του σε Android 2.0.1 τον Δεκέμβριο 2009 (Eclair 0.1) και τον Ιανουάριο 2010 με το Android 2.1 (EclairMR1).

Ανάμεσα στις άλλες αλλαγές είναι και:

- Βέλτιστη ταχύτητα υλικού.
- Υποστήριξη για περισσότερες οθόνες και αναλύσεις.
- Βελτιωμένη διεπαφή του χρήστη.
- Καινούργια διεπαφή χρήσης για την μηχανή αναζήτησης και υποστήριξη του προτύπου HTML5.
- Καινούριες λίστες επαφών.
- Καλύτερος λόγος άσπρου – μαύρου για φόντα.
- Βελτιωμένοι χάρτες Google (googlemaps) 3.1.2.
- Υποστήριξη Microsoft Exchange.
- Ενσωματωμένη υποστήριξη flash για την Camera.
- Ψηφιακή μεγέθυνση(zoom).
- Κλάση MotionEvent βελτιωμένη ώστε οι κατασκευαστές να μπορούν να παρακολουθούν αποτελεσματικότερα τα γεγονότα πολλαπλής αφής.
- Ανανεωμένο εικονικό πληκτρολόγιο.
- Bluetooth 2.1

5.4.4 Android 2.2 FROYO



Froyo Android 2.2

Εικόνα 5 - 7 : Android 2.2 FROYO

Ακολουθεί το **Android 2.2** με το όνομα 'Froyo' τον Μάιο του 2010. Η έκδοση FROYO περιλαμβάνει:

- Βελτιστοποιήσεις στην ταχύτητα γενικά του λειτουργικού συστήματος, στην μνήμη και στην απόδοση.
- Ενσωμάτωση στην μηχανή αναζήτησης, της μηχανής Javascript του ChromeV8.
- Αυξημένη υποστήριξη MicrosoftExchange (σε πολιτικές ασφαλείας, συγχρονισμού ημερολογίου, auto – discovery, GALlook-up, remotewipe).
- Βελτιωμένος προωθητής εφαρμογής (applicationlauncher), με συντομεύσεις προς τις εφαρμογές τηλεφώνου και εφαρμογές της Μηχανής Αναζήτησης.
- Σύνδεση USB και λειτουργία δυναμικής ζώνης (hotspot) WiFi.
- Ανανεωμένη εφαρμογή Αγοράς (Market) με αυτόματη ανανέωση.
- Επιλογή για απαγόρευση πρόσβασης δεδομένων μέσω ενός δικτύου κινητής τηλεφωνίας.
- Γρήγορη εναλλαγή ανάμεσα σε πολλαπλές γλώσσες του πληκτρολογίου και των λεξικών τους.
- Φωνητική κλήση και διαμοιρασμός επαφών με Bluetooth.
- Υποστήριξη για αριθμητικούς και αλφαριθμητικούς κωδικούς.
- Η μηχανή αναζήτησης μπορεί να αποτυπώσει κινούμενα GIFs.
- Υποστήριξη για πεδία μεταφόρτωσης αρχείων στην μηχανή αναζήτησης.
- Υποστήριξη για εγκατάσταση εφαρμογών στην επεκτάσιμη μνήμη 14.
- Υποστήριξη Adobe Flash 10.1.

5.4.5 Android 2.3 GINGERBREAD



Εικόνα 5 - 8 : Android 2.3 GINGERBREAD

Η έκδοση **Android 2.3** με το όνομα “Gingerbread” με την επανέκδοση του σε Android 2.3.3 τον Φεβρουάριο του 2011.

Οι αλλαγές που έχουν γίνει είναι οι ακόλουθες:

- Βελτιωμένο UI (UserInterface) για απλότητα και ταχύτητα.
- Πιο γρήγορη, πιο διαισθητική εισαγωγή κειμένου.
- Επιλογή λέξεων και αντιγραφή/επικόλληση με ένα άγγιγμα.
- Βελτιωμένη ενεργειακή διαχείριση, υποστήριξη NFC (NearFieldCommunication).
- Υποστήριξη video κλήσης.
- Υποστήριξη του πρωτόκολλου WebM για αναπαραγωγή video.

5.4.6 Android 3.0 HONEYCOMB



Εικόνα 5 - 9 : Android 3.0 HONEYCOMB

Η έκδοση **Android 3.0** με το όνομα “Honeycomb”, η οποία είναι στην διάθεση των χρηστών και προγραμματιστών από τον Φεβρουάριο του 2011, λίγες μέρες μετά την επανέκδοση του Android 2.3.3, και προορίζεται αποκλειστικά για ταμπλέτες, στις οποίες γίνεται αναλυτικότερη αναφορά στην συνέχεια.

Μερικά από τα χαρακτηριστικά του είναι:

- Υποστηρίζει διπύρηνους και τετραπύρηνους επεξεργαστές.
- Βελτιωμένη υποστήριξη των ταμπλετών.
- Ανάπτυξη λογισμικού (scripting) για 3D, σε γλώσσα η οποία καλείται "Renderscript".
- Video chat μέσω Google Talk.
- Google eBooks.
- "Ιδιωτική περιήγηση".

5.4.7 Android 4.0 ICE CREAM SANDWICH



Εικόνα 5 - 10 : Android Ice Cream Sandwich

Η επόμενη έκδοση όπως ανακοινώθηκε στο GoogleI/O 2011 στο SanFrancisco, είναι το “IceCreamSandwich”, το οποίο αποτελεί την προσπάθεια της εταιρίας για ενιαίο λειτουργικό σύστημα για όλες τις συσκευές. Είναι διαθέσιμο από το τελευταίο τρίμηνο του 2011 και φιλοδοξία της Google είναι να χρησιμοποιηθεί σε smartphones, tablets, laptops, netbooks. Το AndroidIceCreamSandwich έφερε μαζί του 3DUI, πλουσιότερα widgets, ενισχυμένο multitasking, νέα APIs και φυσικά είναι πλήρως opensource. Μια καλή πρώτη γεύση του AndroidIceCreamSandwich δόθηκε όταν η Google παρουσίασε τη λειτουργία facetracking, η οποία δεν περιορίζεται μόνο στις κινήσεις του κεφαλιού, αλλά ακολουθεί τα μάτια, τη μύτη και το στόμα. Στην περίπτωση που πραγματοποιείται video κλήση, “ζουμάρει” αυτόματα προς αυτόν που μιλά εκείνη τη στιγμή. Το Android έχει καταπληκτικά χαρακτηριστικά και πολλαπλές δυνατότητες, ενώ συνεχίζει να εκσυγχρονίζεται. Επιπρόσθετα, παρέχει καταπληκτικά εργαλεία για την ανάπτυξη εφαρμογών.

5.4.8 Android 4.1.0 JELLY BEAN



Android 4.1 Jelly Bean

Εικόνα 5 - 11 : Android JellyBean 4.1.0

Η έκδοση **Android 4.1** στις 9 Ιουλίου του 2012 ονομάστηκε JellyBean. Μέσα στο διάστημα από τον Ιούλιο του 2012 μέχρι τον Νοέμβρη του 2012 κυκλοφόρησε σαν δυο επανεκδόσεις. Η έκδοση αυτή μαζί με τις επανεκδόσεις έχουν διορθώσεις σφαλμάτων, βελτιώσεις στην απόδοση και χρησιμοποιούν APIlevel 16. Τα χαρακτηριστικά της είναι:

- Επέκταση κοινοποιήσεων.
- Νέα εφαρμογή αναζήτησης που ονομάζεται GoogleNow.
- Καλύτερη φωνητική αναζήτηση.
- Νέα εφαρμογή κάμερας υψηλής ανάλυσης φωτογραφιών Google+.

Έπειτα στις 13 Νοέμβρη του 2012 κυκλοφόρησε η έκδοση 4.2 JellyBean και ακολούθησαν οι επανεκδόσεις 4.2.1 στις 27 Νοέμβρη 2012 και 4.2.2 στις 11 Φεβρουαρίου του 2013 η οποία έχει διόρθωση σφάλματος σχετικά με τη ροή ήχου Bluetooth κ.α. Η έκδοση 4.2 έχει βελτίωση για το κλείδωμα οθόνης, δυνατότητα ομαδικής επικοινωνίας, νέα εφαρμογή ρολογιού με παγκόσμιο ρολόι, χρονόμετρο και χρονοδιακόπτη ενσωματωμένο. Η έκδοση αυτή καθώς και οι επανεκδόσεις της χρησιμοποιούν APIlevel 17.

5.4.9 Android 4.4 KITKAT



Android 4.4 KitKat

Εικόνα 5 - 12 : Android 4.4 KitKat

Το νέο **Android 4.4 KitKat** είναι πλέον επίσημο και διαθέσιμο. Η νέα έκδοση του Android 4.4 έχει τόσες σημαντικές και πολλές βελτιώσεις που αξίζει να σταθούμε στα βασικά. Το UI και UX έχουν αλλάξει και έχουν βελτιωθεί στα σημεία. Αυτό σημαίνει ότι τα γραφικά είναι λίγο διαφορετικά και η εμπειρία χρήσης του AndroidKitKat καλύτερη από ποτέ. Σημαντική αλλαγή που φαίνεται από την πρώτη ματιά είναι η απώλεια των notification bar και control bar που έχουν εξαφανιστεί και έχουν γίνει διάφανες. Νέο είναι και το GoogleNow που έχει βελτιωθεί σημαντικά και μπορεί να προσφέρει περισσότερες και πιο σχετικές πληροφορίες. Το Hangoutsapp αντικαθιστά τα SMS και MMS. Η σημαντικότερη όμως αλλαγή στο Android 4.4 KitKat είναι το ίδιο το Android. Το μεγάλο αφεντικό του Chrome και Android, SundarPichai, δήλωσε ότι παρά τις νέες δυνατότητες του Android 4.4 η νέα έκδοση μπορεί να τρέξει και σε smartphones παλαιότερης τεχνολογίας. Το Google φρόντισε το AndroidKitKat να καταναλώνει λιγότερη μνήμη από ποτέ. Χαρακτηριστικό παράδειγμα ο Chrome που στο Android 4.4 καταναλώνει 16% λιγότερη RAM σε σύγκριση με το Android 4.3. Δεν είναι ξεκάθαρο αν το Android 4.4 έχει ενεργά όλα τα χαρακτηριστικά ακόμα και σε παλιότερα smartphones. Όπως είπε ο SundarPichai, η νέα έκδοση του Android 4.4 KitKat είναι σχεδιασμένη για το επόμενο δισεκατομμύριο νέων χρηστών του internet και έχει κατασκευαστεί ώστε να τρέχει σε high-endsmartphones. Το γεγονός ότι το Android 4.4 είναι “πιο ελαφρύ” από το Android 4.3 είναι θετικό. Στο dock έχει εξαφανιστεί η γραμμή που χωρίζει τις αγαπημένες μας εφαρμογές από την υπόλοιπη homescreen και έχει αντικατασταθεί από τις τελείες που υποδεικνύουν σε ποια οθόνη βρισκόμαστε. Και αυτό δίνει την αίσθηση ότι η οθόνη έχει μεγαλώσει καθώς

υποσυνείδητα το μάτι του χρήστη πηγαίνει κυρίως στα widget και τις υπόλοιπες εφαρμογές, επειδή ίσως θεωρεί δεδομένες τις εφαρμογές του dock. Η ομοιομορφία συνεχίζεται και μέσα στο συρτάρι των εφαρμογών, όπου και εκεί το μαύρο φόντο έχει αντικατασταθεί από μία διαφάνεια η οποία φυσικά παίρνει το φόντο της εκάστοτε homescreen, ενώ το grid έχει μειωθεί σε τέσσερις στήλες (από πέντε), κάνοντας τα εικονίδια να εμφανίζονται μεγαλύτερα. Όσον αφορά τις ρυθμίσεις βλέπουμε ότι υπάρχει η νέα επιλογή "Tap&pay" η οποία παραπέμπει σε πληρωμές μέσω NFC όπως είδαμε παλιότερα στο GoogleWallet. Εφόσον όμως το Wallet δεν ήταν διαθέσιμο εκτός Η.Π.Α., δεν ξέρουμε αν η επιλογή θα είναι διαθέσιμη σε όλες τις χώρες. Στην εύρεση της τοποθεσίας μας μπορεί να βοηθήσει το GPS, το WiFi και το δίκτυο κινητής τηλεφωνίας αλλά οι επιλογές που υπάρχουν σήμερα δεν είναι ξεκάθαρες. Πλέονυπάρχουν τρεις επιλογές:

- η "Υψηλή ακρίβεια", η οποία χρησιμοποιεί και τα τρία
- η "Εξοικονόμηση μπαταρίας" η οποία δε χρησιμοποιεί το GPS και
- οι "Αισθητήρες συσκευής" η οποία χρησιμοποιεί μόνο το GPS και όχι τα δίκτυα.

5.4.10 Android 5.0 - 5.1 LOLLIPOP



Εικόνα 5 - 13 : Android 5.0 - 5.1 Lollipop

Η Google στις 12 Νοεμβρίου του 2014 έκανε διαθέσιμη την έκδοση του λειτουργικού **Android 5.0 Lollipop**, με την συσκευή που θα δέχτηκε πρώτη την αναβάθμιση να είναι το Nexus 5. Η μεγάλη αλλαγή είναι το MaterialDesign, που αλλάζει αισθητά την εμφάνιση του λειτουργικού και των εφαρμογών της Google, με πιο bold, ρεαλιστικά χρώματα και

γραφικά. Το Android Lollipop υπόσχεται να βελτιώσει και την αυτονομία της μπαταρίας η οποία θα παρατείνεται για επιπλέον 90 λεπτά με το `batterysaverfeature`, ενώ ο χρήστης θα μπορεί να δει για πόση ώρα θα διαρκέσει η μπαταρία και πόση ώρα απομένει μέχρι να φορτίσει πλήρως. Το Android 5.0 Lollipop είναι φτιαγμένο ώστε να δίνει την ίδια αίσθηση σε όλες τις συσκευές με διαφορετικό μέγεθος οθόνης και να προσφέρει την ίδια εμπειρία όταν ο χρήστης μετακινείται από μια συσκευή σε μια άλλη. Μια από τις σημαντικότερες αλλαγές αφορά τις ειδοποιήσεις (notifications) οι οποίες γίνονται αρκετά καλύτερες και πλέον εμφανίζονται και στην οθόνη κλειδώματος, ταξινομημένες με βάση το πόσο σημαντική είναι η κάθε μία, ενώ ο χρήστης θα μπορεί να κρύψει όποια δε θεωρεί σημαντική και ποιες ειδοποιήσεις από συγκεκριμένες επαφές θέλει να εμφανίζονται. Επιπλέον, έχουν προστεθεί τα `multipleprofiles`, τα οποία είναι χρήσιμα αν τη συσκευή τη χρησιμοποιούν παιδιά και ο χρήστης θέλει να έχει τον απόλυτο έλεγχο. Όσον αφορά στην ασφάλεια το νέο λειτουργικό φέρνει `killswitch`, το οποίο με τη χρήση ενός κωδικού θα αχρηστεύει τη συσκευή για όποιον δεν γνωρίζει τον κωδικό. Αυτό γίνεται με την εύρεση της τοποθεσίας που βρίσκεται εκείνη τη στιγμή η συσκευή και η οποία θα πρέπει να έχει χαρακτηριστεί από πριν ως ασφαλής. Επίσης, όπως σας είχαμε ενημερώσει, η κρυπτογράφηση δεδομένων στο Android L θα είναι προεπιλεγμένη [10].

Ακολούθησαν οι εκδόσεις 5.0.1 και 5.0.2 με μικροδιορθώσεις φτάνοντας στις 9 Μαρτίου του 2015 που η Google ανακοίνωσε επίσημα τη διάθεση του update **Android 5.1 Lollipop** το οποίο υπόσχεται μεγαλύτερη σταθερότητα και καλύτερη απόδοση, ενώ φέρνει νέες λειτουργίες όπως υποστήριξη πολλαπλής κάρτας SIM, HDvoice για συμβατές συσκευές και προστασία της συσκευής σε περίπτωση κλοπής.

- Υποστήριξη πολλαπλών SIMcards: Αν χρησιμοποιείτε smartphone με δύο κάρτες SIM, τότε θα μπορείτε στο εξής να τις εκμεταλλεύεστε ταυτόχρονα.
- Device Protection: Σε περίπτωση κλοπής ή απώλειας της συσκευής σας, θα παραμένει κλειδωμένη μέχρι να πραγματοποιήσετε sign-in με το προκαθορισμένο Google Account, ακόμη και αν ο κλέφτης καταφέρει να το γυρίσει σε εργοστασιακές ρυθμίσεις. Η λειτουργία θα είναι διαθέσιμη για την πλειοψηφία των Android Smartphones /Tablets με Android 5.1 Lollipop.
- HD Voice: Πεντακάθαρος ήχος στις κλήσεις μεταξύ συσκευών που χρησιμοποιούν Android 5.1 Lollipop.

- Δυνατότητα διαχείρισης του WiFi και ελέγχου των συζευγμένων συσκευών Bluetooth απευθείας από το μενού γρήγορων ρυθμίσεων

5.5 Διαχείριση μνήμης στο Android

Το android είναι θεωρητικά φτιαγμένο για multitasking. Έτσι, όταν κλείνεις μια εφαρμογή, το σύστημα δε συμπεριφέρεται όπως τα άλλα λειτουργικά συστήματα (π.χ. Windows), δηλαδή δεν κλείνει τελείως όλες τις διεργασίες (processes) που είχε ανοίξει η εφαρμογή, αλλά τις βάζει στο background, και τις σκοτώνει όταν νομίζει ότι χρειάζεται. Συγκεκριμένα, τις σκοτώνει όταν η μνήμη του συστήματος πέσει κάτω από ένα όριο. Η ιδέα είναι ότι αυτές οι ανενεργές διεργασίες δε βλάπτουν τη συνολική απόδοση του συστήματος, και αν χρειαστείς να ξανανοίξεις την εφαρμογή όσο τρέχουν στο υπόβαθρο, θα ανοίξει πιο γρήγορα, με αποτέλεσμα ένα γρήγορο και αποδοτικό λειτουργικό. Οι διεργασίες σκοτώνονται με σειρά παλαιότητας, όταν η μνήμη πέσει κάτω από το όριο. Ο εσωτερικός διαχειριστής μνήμης στο android διαχωρίζει τις εφαρμογές/διεργασίες που τρέχουν στο τηλέφωνο σε 6 καταστάσεις, από EmptyApp (νεκρή διεργασία, δεν κάνει απολύτως τίποτα και περιμένει να διαγραφεί ή να ξανανοίξεις την εφαρμογή που την καλεί), μέχρι ForegroundApp (που τρέχει εκείνη τη στιγμή). Όταν σταματήσεις να βλέπεις μια εφαρμογή, είτε την βάλεις στο background με το Home Button είτε την κλείσεις με το Back Button, σταματάει αυτόματα να είναι σε κατάσταση ForegroundApp (προφανώς), και ανάλογα με τη λειτουργία της και τον τρόπο που έχει κωδικοποιηθεί, μπαίνει στο υπόβαθρο σε μια άλλη κατάσταση. Αυτό δεν είναι κακό και είναι βασική λειτουργία του android. Το πρόβλημα δυστυχώς είναι ότι η ιδέα δε δουλεύει τέλεια, γιατί προϋποθέτει καλά κωδικοποιημένες εφαρμογές (πράγμα που δυστυχώς δεν ισχύει από τη στιγμή που ο οποιοσδήποτε μπορεί να ανεβάσει την εφαρμογή του στο market) και εξαρτάται επίσης από τις δυνατότητες της συσκευής. Το default όριο μνήμης για να κλείσει μια EMPTY_APP διεργασία είναι χαμηλό, π.χ. στο GalaxyS είναι 48 MB (σε άλλες συσκευές είναι ακόμα λιγότερο, μέχρι και 24 MB). Δηλαδή το τηλέφωνο αρχίζει να σκοτώνει apps που βρίσκονται σε κατάσταση empty, όταν η μνήμη πέσει κάτω από 48 MB. Αυτό το όριο είναι πολύ χαμηλό για τα τελευταίας κατηγορίας κινητά, που τρέχουν με 150-200Mbeλεύθερα. Μέχρι να πέσει η μνήμη στα 48 MB, το υπόβαθρο γεμίζει με νεκρές διεργασίες και καθυστερεί επειδή δεν έχει αρκετή ελεύθερη μνήμη, χωρίς λόγο. Επίσης η

μπαταρία, η οποία ΔΕΝ επηρεάζεται από την ελεύθερη μνήμη (κάτι που πιστεύουν λανθασμένα πολλοί), επηρεάζεται από το αν λειτουργεί ο επεξεργαστής ή όχι. Αν οι (άσχημα κωδικοποιημένες) διεργασίες στο υπόβαθρο ζητάνε ακόμα υπολογιστική ισχύ, από κει χάνεται και μπαταρία. Σ' αυτά ακριβώς πατάνε οι task managers.

Ποια είναι η λύση; Υπάρχουν δυο κοινά διαδεδομένες λύσεις, η πρώτη άκομψη και ενάντια στη φιλοσοφία του android, και η δεύτερη σαφώς καλύτερη.

1. Task manager/Taskkiller. Ένας καλός task manager μπορεί να δει τις διεργασίες που βρίσκονται και στις 6 καταστάσεις, και ειδικά τις Empty. Και όπως στα Windows (ctrl-alt-del), σου δίνει τη δυνατότητα να σκοτώσεις όλα τα apps που έχουν διεργασίες που τρέχουν στο background. Από τις εκδόσεις Froyo και μετά, οι εφαρμογές δε σκοτώνονται, αλλά απλά ξαναρχίζουν από την αρχή (re-initialize). Αυτό το σκότωμα/επαναρχικοποίηση δεν ενδείκνυται, τις περισσότερες φορές μάλιστα είναι κακή λύση, γιατί με το να κλείνεις διεργασίες που το android δε σκόπευε να κλείσει, μπορεί να δημιουργήσεις πρόβλημα, και μάλιστα να κάνεις το τηλέφωνό σου πιο αργό, γιατί πρέπει να τα ξαναρχίζει όλα απ' την αρχή. Μερικοί task managers έχουν τη δυνατότητα να σκοτώνουν αυτόματα τις διεργασίες στο background μετά από ορισμένη ώρα. Υπάρχουν task managers που σε προστατεύουν από λάθη έχοντας safe modes και μπορείς να αποκλείσεις κάποιες εφαρμογές από το να σκοτώνονται αν ξέρεις ότι δημιουργείται πρόβλημα (ένα παράδειγμα ενός ολοκληρωμένου task manager είναι ο AdvancedTaskKillerPro). Παρόλα αυτά παραμένει μια βασικά λάθος λύση, και τελειώς λάθος λογική στον κόσμο του Android.
2. Τα default όρια για τις έξη καταστάσεις, κάτω από τα οποία σκοτώνονται αυτόματα οι εφαρμογές, βρίσκονται αποθηκευμένα στο αρχείο `/sys/module/lowmemorykiller/parameters/minfree` . Αυτές οι τιμές λοιπόν μπορούν να αλλάξουν, και έτσι αντί το σύστημα να περιμένει να φτάσει η μνήμη στα 48MB για να κλείσει τα EMPTY_APPS, να τα σκοτώνει όταν πέφτει κάτω από 100 MB. Με αυτόν τον τρόπο ακολουθείται η λογική του android και υπάρχει πάντα ελεύθερη μνήμη.

Μια εφαρμογή που το κάνει αυτό είναι το Autokiller. Η εφαρμογή αυτή δίνει κάποιες προεπιλεγμένες τιμές για τα παραπάνω όρια, και συνήθως προτείνεται το Optimum, το

Strict ή το Aggressive. Μόλις διαλέξουμε κάποια τιμή, είτε από τις preset είτε κάποια άλλη, αλλάζει το minifree αρχείο. Δεν κάνει κάτι μόνη της, δεν τρέχει συνεχώς στο υπόβαθρο, και αφήνει το kernel του android να κάνει τη δουλειά του όπως αυτό νομίζει. Φορτώνει απλά στην εκκίνηση, γιατί τα settings αλλιώς χάνονται. Το tweaks στα καινούρια CF-rootkernels του Chainfire (για τις Samsungsσκευές) περιέχει κι αυτό έναν παρόμοιο manager, καθώς επίσης υπάρχουν πολλές παρόμοιες εφαρμογές καθώς και script που κάνουν το ίδιο δηλαδή αλλάζουν τις τιμές του minifree. Η δεύτερη λύση είναι η πιο σωστή. Φυσικά το minifree αρχείο είναι αρχείο συστήματος, έτσι χρειάζεται root για να εγκαταστήσεις τον autokiller (ή κάτι παρόμοιο). Σε συνδυασμό με ένα πρόγραμμα σαν το autostarts, με το οποίο μπορείς να ελέγξεις αν κάποιες εφαρμογές ανοίγουν όταν δεν πρέπει και να τις σταματήσεις, το τηλέφωνο θα λειτουργεί όπως πρέπει. Δηλαδή δε χρειαζομαι καθόλου task manager. Θεωρητικά όχι, πρακτικά, λόγω των άσχημα κωδικοποιημένων εφαρμογών, είναι καλό να υπάρχει κι ένας task manager που να μην τρέχει συνέχεια και φυσικά χωρίς αυτόματο kill. Το autokiller περιέχει έναν task manager. Και δυστυχώς, η λύση του task manager παραμένει η μόνη λύση για παλιότερα τηλέφωνα (Eclair και κάτω), και γι' αυτά που δεν έχουν δικαιώματα root.

5.6 Βιβλιοθήκες στο Android

Οι βιβλιοθήκες(Libraries) περιέχουν όλο τον κώδικα που περιέχει το AndroidOS. Για παράδειγμα, η SQLite βιβλιοθήκη παρέχει υποστήριξη έτσι ώστε μια εφαρμογή να χρησιμοποιήσει την αποθήκευση δεδομένων, η Webkit βιβλιοθήκη παρέχει λειτουργίες για το διαδικτυακό σερφάρισμα. Οι βιβλιοθήκες βρίσκονται πάνω από τον πυρήνα Linux και είναι γραμμένες σε γλώσσες προγραμματισμού C και C++. Επίσης δεν μπορούν να τρέξουν μόνες τους σαν προγράμματα και μπορούν μόνο να κληθούν. Από την έκδοση 1.6 DonutAndroid και μετά οι προγραμματιστές έχουν τη δυνατότητα να γράφουν τις δικές τους βιβλιοθήκες χρησιμοποιώντας την εργαλειοθήκη NDK(NativeDevelopmentKit) του Android.

Οι βασικές βιβλιοθήκες είναι:

- SurfaceManager: Είναι υπεύθυνη για την πρόσβαση στο σύστημα προβολής και συνθέτει δυσδιάστατα και τρισδιάστατα επίπεδα γραφικών.
- OpenGL ES: Ενσωματωμένο υποσύνολο της OpenGL. Συνοψίζει το μεγάλο σε όγκο OpenGL σε λίγα βασικά στοιχεία ώστε να μπορεί να εφαρμόζεται και σε συσκευές με χαμηλά τεχνικά χαρακτηριστικά. Έχει λιγότερη κατανάλωση μπαταρίας και ελάχιστες ενεργειακές απαιτήσεις. Είναι διαθέσιμο σαν πρότυπο σε Android καθώς και iOS smartphones.
- SGL: Χαμηλού επιπέδου βιβλιοθήκη δισδιάστατων γραφικών.
- MediaFramework: Παρέχει αποκωδικοποιητές για την αναπαραγωγή πολυμέσων.
- FreeType: Βιβλιοθήκη ανάπτυξης λογισμικού που διατίθεται σε πηγή και σε δυαδική μορφή όπου παρέχει ευκρίνεια στο bitmaps και στις γραμματοσειρές.
- SSL: Χρησιμοποιεί μεθόδους κρυπτογράφησης δεδομένων για ασφάλεια κατά τη μετάδοση δεδομένων στο διαδίκτυο.
- SQLite: Είναι ένα σχεσιακό σύστημα διαχείρισης βάσης δεδομένων των εφαρμογών.
- WebKit: Παρέχει λειτουργίες για την περιήγηση στο διαδίκτυο.
- Libc: Συλλογή από επικεφαλίδες, αρχεία και συναρτήσεις βιβλιοθήκης στη C, υλοποιούν κάποιες κοινές λειτουργίες και είναι τροποποιημένες για Android συσκευές.

5.7 Πλαίσιο εφαρμογής – Application Framework

Είναι το επόμενο επίπεδο μετά τις βιβλιοθήκες. Περιέχει έτοιμες λειτουργίες που δίνουν τη δυνατότητα σε προγραμματιστές να αναπτύξουν εφαρμογές Android. Το πλαίσιο αυτό είναι προ εγκατεστημένο στο Android και είναι επεκτάσιμο. Τα δομικά στοιχεία του πλαισίου εφαρμογής είναι:

- Διαχειριστής δραστηριοτήτων(Activity Manager): Είναι υπεύθυνο για τον έλεγχο του κύκλου ζωής μιας εφαρμογής και τη διατήρηση μιας στοίβας που δίνει τη δυνατότητα στο χρήστη να πλοηγηθεί σε προηγούμενες οθόνες.
- Διαχειριστής παραθύρων(Window Manager): Είναι υπεύθυνο για την οργάνωση της οθόνης. Τοποθετεί επιφάνειες και αποφασίζει σε ποιο επίπεδο θα μπουν αυτές.

- Πάροχος περιεχομένου(Content Providers): Δίνει τη δυνατότητα να μοιράζονται και να ανταλλάσσονται δεδομένα μέσω των εφαρμογών.
- Σύστημα προβολών(View System): Περιέχει αντικείμενα GUI τα οποία μπορούν να χρησιμοποιηθούν για κατά τη διάρκεια του σχεδιασμού μιας συσκευής.
- Διαχειριστής κοινοποιήσεων(Notification Manager): Ενημερώνει το χρήστη για τα γεγονότα που συμβαίνουν στο παρασκήνιο.
- Διαχειριστής πακέτων(Package Manager): Είναι συλλογή από εργαλεία λογισμικού που χρησιμοποιούνται για την αυτοματοποίηση της διαδικασίας εγκατάστασης, την αναβάθμιση, τη ρύθμιση και την αφαίρεση πακέτων λογισμικού για το λειτουργικό σύστημα Android.
- Διαχειριστής πόρων(Resource Manager): Παρέχει πρόσβαση σε υλικό που δεν είναι σε μορφή κώδικα όπως εικόνες και αρχεία ήχου.
- Διαχειριστής τοποθεσίας(Location Manager): Παρέχει υπηρεσίες που επιτρέπουν στις εφαρμογές να λαμβάνουν συχνά ενημερώσεις της γεωγραφικής θέσης της συσκευής.
- XMPPService: Σύνολο ανοικτών πρωτοκόλλων άμεσης επικοινωνίας βασισμένα στο xml το οποίο παρέχει τη δυνατότητα επικοινωνίας με οποιονδήποτε χρήστη ανεξαρτήτως των εμπλεκόμενων διακομιστών.

5.8 Playstore



Εικόνα 5 - 14 : Google Playstore

Το Google Play είναι μια υπηρεσία η οποία παρέχεται από την Google Ireland Limited και υπόκειται στους Όρους Παροχής Υπηρεσιών Google. Είναι μια "Υπηρεσία", όπως αυτή καθορίζεται από την Google, και αυτοί οι Όροι παροχής Υπηρεσιών του Google Play αποτελούν επιπλέον όρους οι οποίοι ισχύουν για τη χρήση του Google Play.

5.9 Λογαριασμός Google–Gmail

Για να αρχίσετε να χρησιμοποιείτε πραγματικά το Android σε ένα smartphone πρέπει να δημιουργήσετε έναν λογαριασμό Google. Είναι δωρεάν και είναι πολύ εύκολο. Πρέπει επίσης να έχετε σύνδεση στο Internet, είτε μέσω ενός δικτύου κινητής τηλεφωνίας είτε μέσω Wi-Fi. Με έναν λογαριασμό Google μπορείτε να χρησιμοποιήσετε τις περισσότερες υπηρεσίες Google. Μπορείτε να δημιουργήσετε έναν λογαριασμό ή να συνδεθείτε σε έναν ήδη υπάρχοντα από το smartphone σας. Αν το προτιμάτε, μπορείτε να δημιουργήσετε τον λογαριασμό από έναν υπολογιστή, επισκεπτόμενοι τη διεύθυνση www.google.com. Ένας λογαριασμός παρέχει πολλά πλεονεκτήματα. Μερικά από αυτά είναι τα εξής:

Κατέβασμα εφαρμογών από το κατάστημα εφαρμογών που ονομάζεται Google Play.

Μέσω το Gmail στέλνονται, λαμβάνονται και οργανώνονται συζητήσεις μέσω ηλεκτρονικού ταχυδρομείου.

Όταν αποθηκεύονται οι επαφές σε λογαριασμό Google στην εφαρμογή Επαφές, τηρείται αυτόματα ένα αντίγραφο ασφαλείας

Μέσω έγκρισης, το email, τα συμβάντα του ημερολογίου και οι φωτογραφίες συγχρονίζονται αυτόματα στον λογαριασμό Google.

Καθώς ο λογαριασμός είναι διαθέσιμος από οποιαδήποτε συσκευή, μπορείτε να συγχρονιστούν πολλές συσκευές και να παρέχεται πρόσβαση στο περιεχόμενο από οπουδήποτε.

5.10 Υπηρεσίες και εφαρμογές Google

Η Google προσφέρει πολλές διαφορετικές υπηρεσίες και εφαρμογές. Ορισμένες από τις υπηρεσίες της Google μπορεί να χρησιμοποιηθούν χωρίς να σύνδεση στον λογαριασμό Google , όπως για παράδειγμα η αναζήτηση Google, το Google Maps ή το YouTube. Άλλες είναι διαθέσιμες μόνο μέσω σύνδεσης, όπως για παράδειγμα το Gmail, η υπηρεσία online αποθήκευσης Google Drive, το Google+ ή το Google Calendar.

5.11 Google Maps



Εικόνα 5 - 15 : Google Maps

Το Google Maps επιτρέπει στο χρήστη να παρακολουθεί την τρέχουσα τοποθεσία του, να προβάλει τις συνθήκες οδικής κυκλοφορίας σε πραγματικό χρόνο και να λαμβάνει λεπτομερείς οδηγίες σχετικά με το πώς θα φθάσει στον προορισμό του. Επίσης, παρέχει ένα εργαλείο αναζήτησης όπου ο χρήστης μπορεί να εντοπίσει περιοχές ενδιαφέροντος ή διεύθυνση σε χάρτη ή να βλέπει τοποθεσίες σε επίπεδο δρόμου.

- Χρειάζεται ενεργή σύνδεση δεδομένων ή Wi-Fi

- Για να βρεθεί η τοποθεσία με το Google Maps, πρέπει να ενεργοποιηθούν οι πηγές τοποθεσίας.
- Το Google Maps δεν καλύπτει κάθε χώρα ή πόλη.

5.12 Εφαρμογές Android – Android Applications



Εικόνα 5 - 16 : Εφαρμογές Android

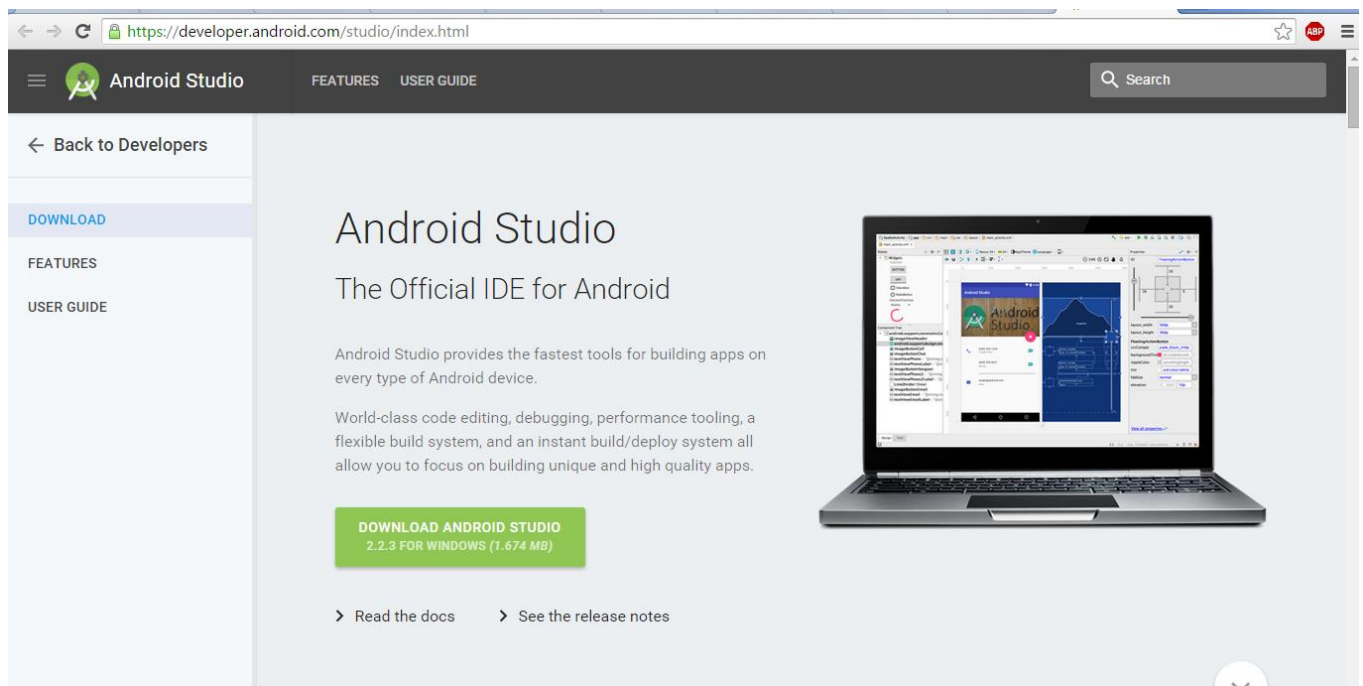
“Apps” είναι η συντομογραφία της αγγλικής λέξης applications (εφαρμογές), οι οποίες στον κόσμο των κινητών συσκευών είναι μικρά προγράμματα λογισμικού κατασκευασμένα ειδικά για smartphone. Οι εφαρμογές βρίσκονται στο ανώτερο επίπεδο της αρχιτεκτονικής του Android και καθιστούν το smartphone ένα πολυλειτουργικό και πολύ προσωπικό εργαλείο, αφού ο χρήστης αποφασίζει ποιες εφαρμογές θα ληφθούν και θα χρησιμοποιηθούν. Οι εφαρμογές έχουν δημιουργηθεί από προγραμματιστές σε γλώσσα προγραμματισμού JAVA χρησιμοποιώντας το Android Software Development Kit(SDK) στο περιβάλλον ανάπτυξης IDE Eclipse με τη βοήθεια της επέκτασης Android Development Tools(ADT) . Υπάρχουν πάρα πολλές κατηγορίες εφαρμογών, όπως για παράδειγμα παιχνίδια, εφαρμογές για ειδήσεις και περιοδικά, υγεία και fitness, για τον καιρό, τα ταξίδια, την εκπαίδευση και τις επιχειρήσεις.

ΚΕΦΑΛΑΙΟ 6 : ΔΗΜΙΟΥΡΓΙΑ ΕΦΑΡΜΟΓΗΣ ΣΕ ANDROID ΠΕΡΙΒΑΛΛΟΝ

Για την δημιουργία της εφαρμογής μας το πρώτο πράγμα που πρέπει να γίνει είναι η εγκατάσταση ενός κατάλληλου περιβάλλοντος. Για την δημιουργία της συγκεκριμένης πτυχιακής χρησιμοποιήθηκε το Android Studio σε λειτουργικό Windows.

6.1 Εγκατάσταση Περιβάλλοντος Υλοποίησης της Εφαρμογής - Android Studio

Το πρώτο βήμα που πρέπει να γίνει είναι το κατέβασμα του Android Studio στον υπολογιστή μας. Το Android Studio είναι ένα από τα προγράμματα ανάπτυξης Android εφαρμογών που παρέχονται δωρεάν. Για να το κατεβάσουμε μπαίνουμε στο link <https://developer.android.com/studio/index.html> και επιλέγουμε download Android Studio.



Εικόνα 6 - 1 : Download Android Studio

Κατέβασμα JDK

Το Android Studio χρειάζεται μια Java πλατφόρμα για να λειτουργήσει. Όποτε, πριν ξεκινήσουμε την εγκατάσταση του Android Studio θα πρέπει να κατεβάσουμε το JDK (Java Development Kit) που μας παρέχει αυτή τη δυνατότητα. Για να κατεβάσουμε το JDK πηγαίνουμε στο παρακάτω link :

<http://www.oracle.com/technetwork/java/javase/downloads/jdk8-downloads-2133151.html>

αποδεχόμαστε τους όρους και τις πολιτικές και το κατεβάζουμε αυτό που είναι συμβατό με το λειτουργικό και τον επεξεργαστή μας.

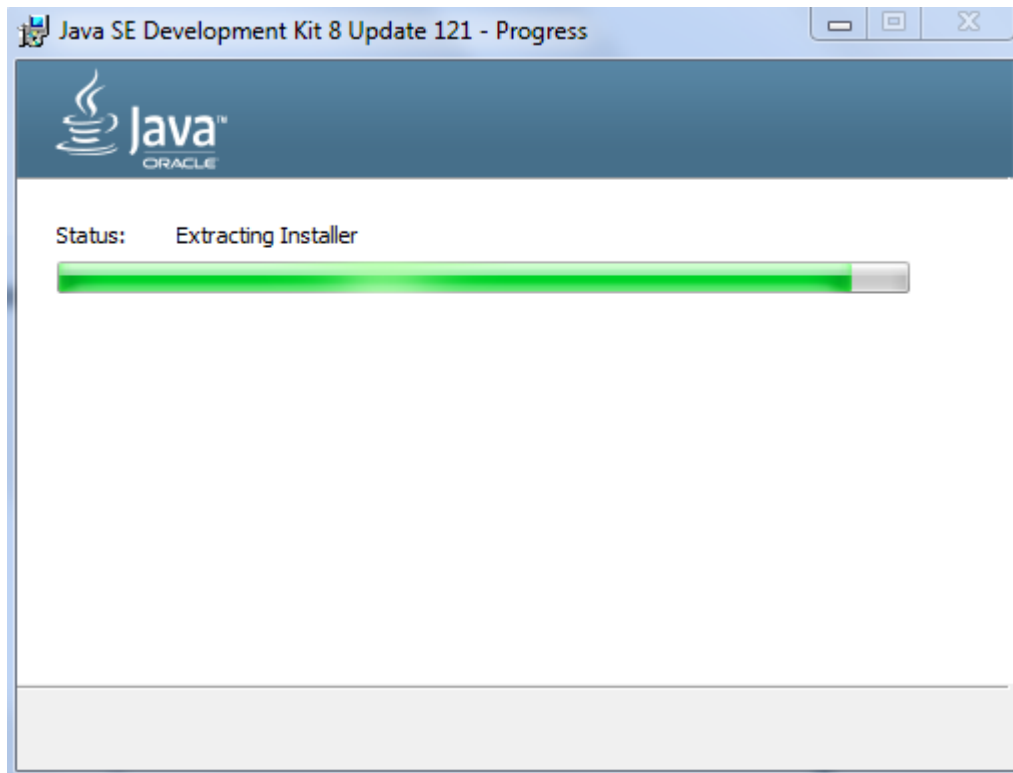
The screenshot shows the Oracle Java SE Development Kit 8 Downloads page. The page is titled "Java SE Development Kit 8 Downloads" and includes a navigation menu with links for Overview, Downloads, Documentation, Community, Technologies, and Training. The main content area features a table of download links for various operating systems and architectures. The table has columns for Product / File Description, File Size, and Download. Below the table, there is a section for "Java SE Development Kit 8u121 Demos and Samples Downloads" with a similar table.

Product / File Description	File Size	Download
Linux ARM 32 Hard Float ABI	77.58 MB	jdk-8u121-linux-arm32-vfp-hflt.tar.gz
Linux ARM 64 Hard Float ABI	74.83 MB	jdk-8u121-linux-arm64-vfp-hflt.tar.gz
Linux x86	162.41 MB	jdk-8u121-linux-i586.tar.gz
Linux x86	177.13 MB	jdk-8u121-linux-i686.tar.gz
Linux x84	156.96 MB	jdk-8u121-linux-x84.rpm
Linux x84	174.78 MB	jdk-8u121-linux-x84.tar.gz
Mac OS X	223.21 MB	jdk-8u121-macosx-x84.dmg
Solaris SPARC 64-bit	136.84 MB	jdk-8u121-solaris-sparcv9.tar.z
Solaris SPARC 64-bit	96.07 MB	jdk-8u121-solaris-sparcv9.tar.gz
Solaris x84	140.42 MB	jdk-8u121-solaris-x84.tar.z
Solaris x84	96.9 MB	jdk-8u121-solaris-x84.tar.gz
Windows x86	186.36 MB	jdk-8u121-windows-i586.exe
Windows x84	195.51 MB	jdk-8u121-windows-x84.exe

Product / File Description	File Size	Download
Linux ARM 32 Hard Float ABI	9.62 MB	jdk-8u121-linux-arm32-vfp-hflt-demos.tar.gz
Linux ARM 64 Hard Float ABI	8.84 MB	jdk-8u121-linux-arm64-vfp-hflt-demos.tar.gz
Linux x86	62.66 MB	jdk-8u121-linux-i586-demos.rpm
Linux x86	62.51 MB	jdk-8u121-linux-i686-demos.tar.gz
Linux x84	63.25 MB	jdk-8u121-linux-x84-demos.rpm

Εικόνα 6 - 2 : Κατέβασμα JDK

Στη συνέχεια το εγκαθιστούμε στον υπολογιστή μας, η εγκατάσταση είναι εύκολη, απλά ακολουθούμε τα βήματα.

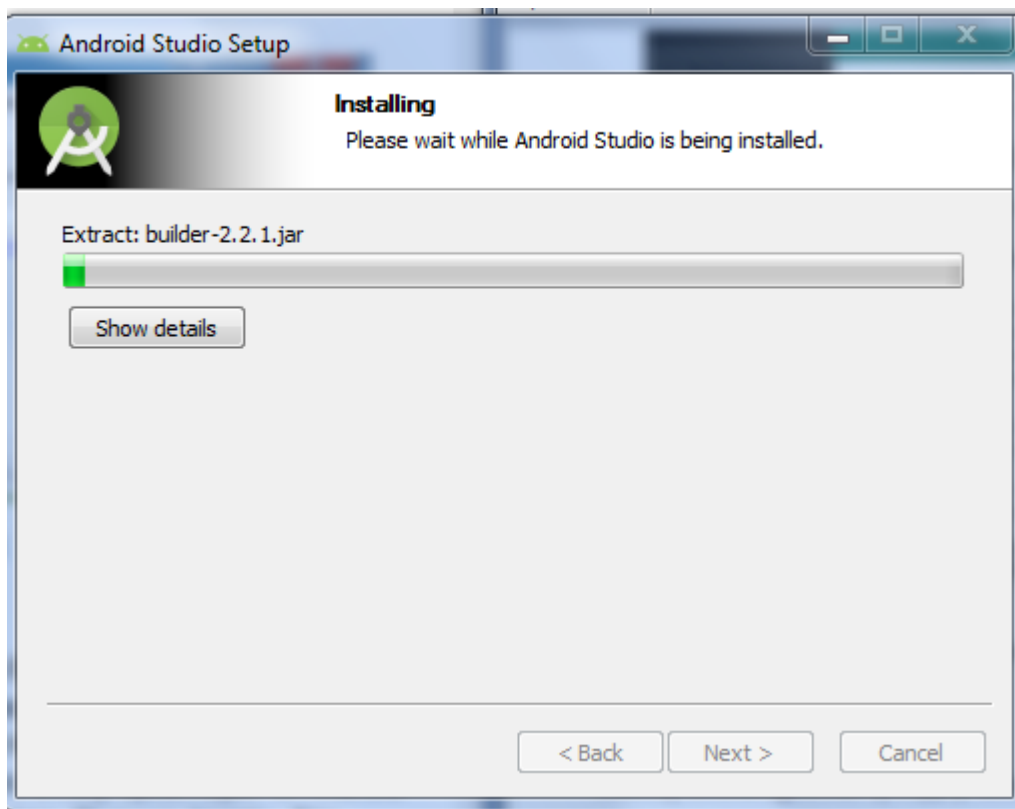


Εικόνα 6 - 3 : Εγκατάσταση JDK

Αυτό το βήμα είναι προαιρετικό, σε περίπτωση που έχουμε εργαστεί σε περιβάλλον java ξανά, διότι θα υπάρχει ήδη μια πλατφόρμα.

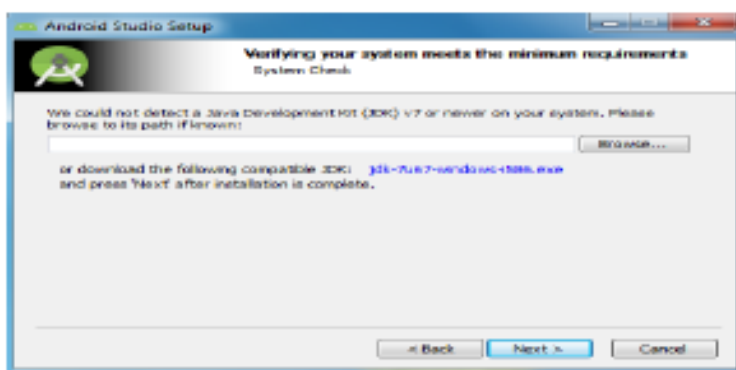
Εγκατάσταση Android Studio

Για να κατεβάσουμε το αρχείο του Android Studio πρέπει να δεχτούμε τους όρους και τις προϋποθέσεις. Μόλις Ολοκληρωθεί η λήψη του Android Studio, ανοίγουμε το αρχείο που κατεβάσαμε, πατάμε **Εκτέλεση** και έπειτα **Next**



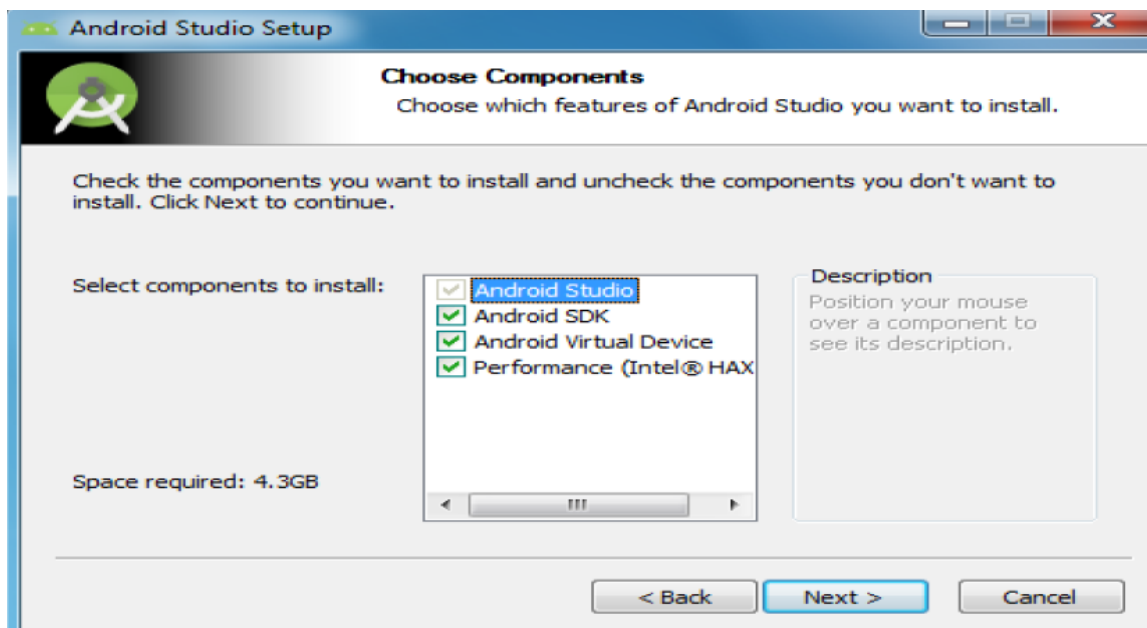
Εικόνα 6 - 4 : Εγκατάσταση Android Studio

Σε επόμενο βήμα μας ζητά να περάσουμε το JDK.



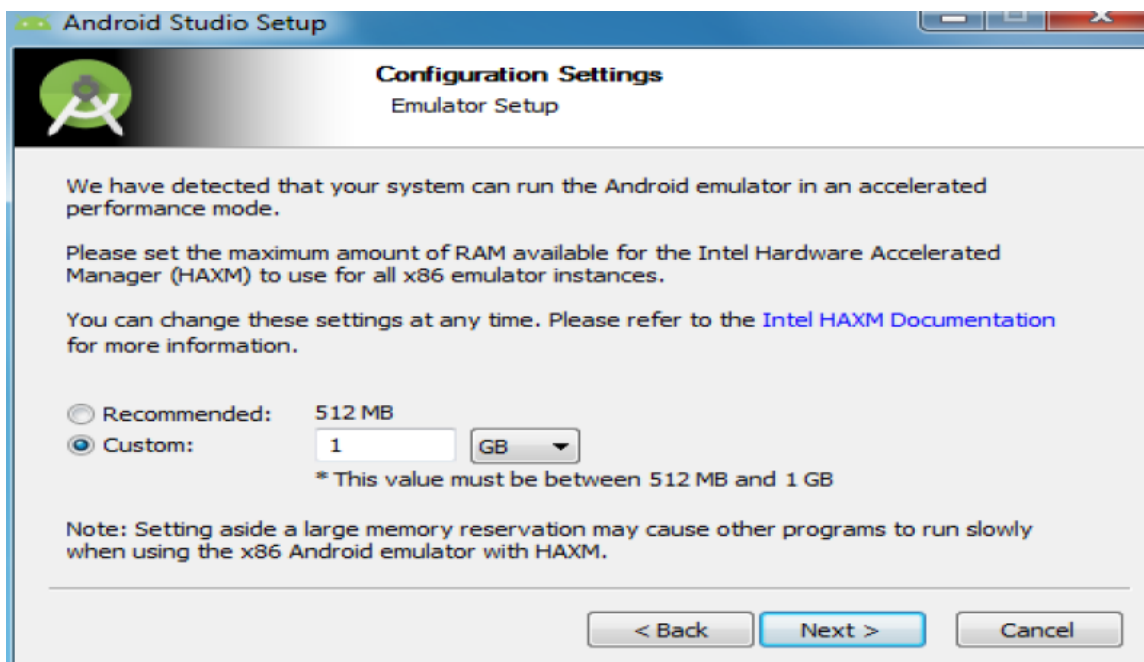
Εικόνα 6 - 5 : JDK

Έπειτα μεταφερόμαστε στο εξής παράθυρο όπως φαίνεται στην *Εικόνα 6 – 5* στο οποίο μας ζητάει να επιλέξουμε πια components θα εγκατασταθούν.



Εικόνα 6 - 6 : Επίλογη components για εγκατάσταση

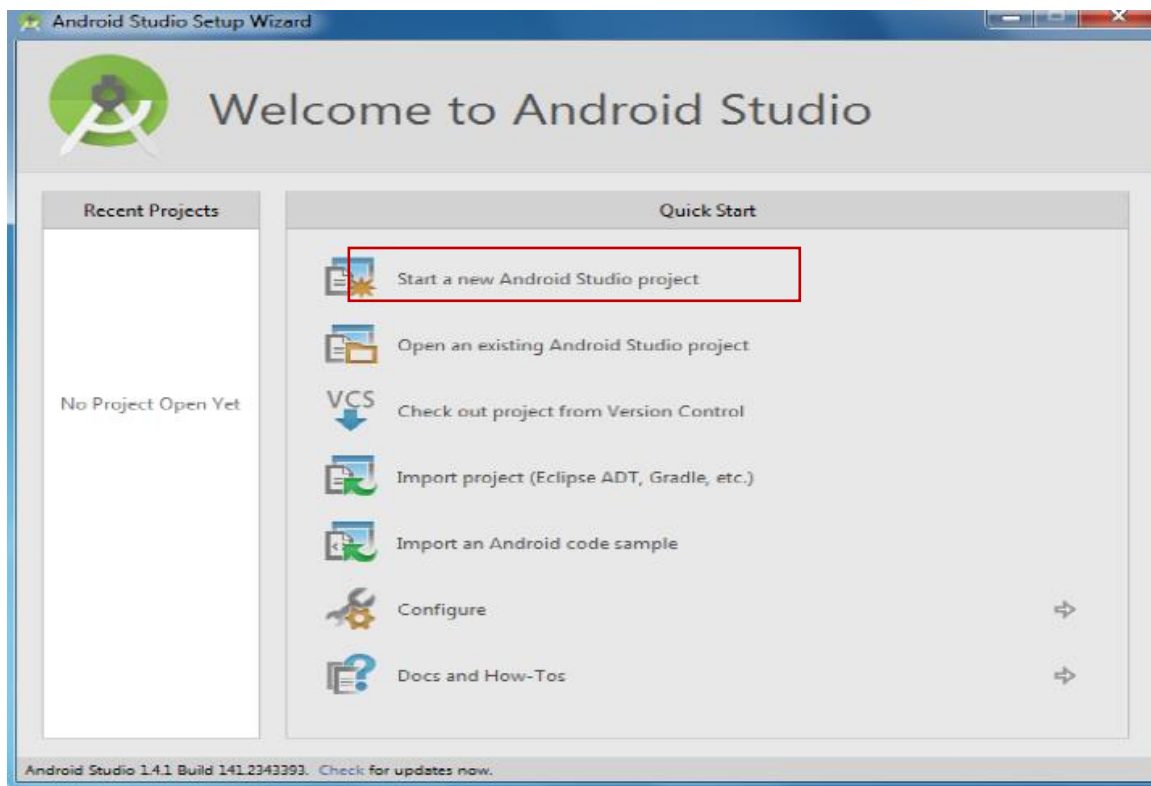
Πατάμε **next** και επιλέγουμε που θα αποθηκευτεί το Android Studio. Σε επόμενο βήμα πρέπει να επιλέξουμε τα MB που θα δεσμεύονται όταν θα ανοίγουμε το Android Studio.



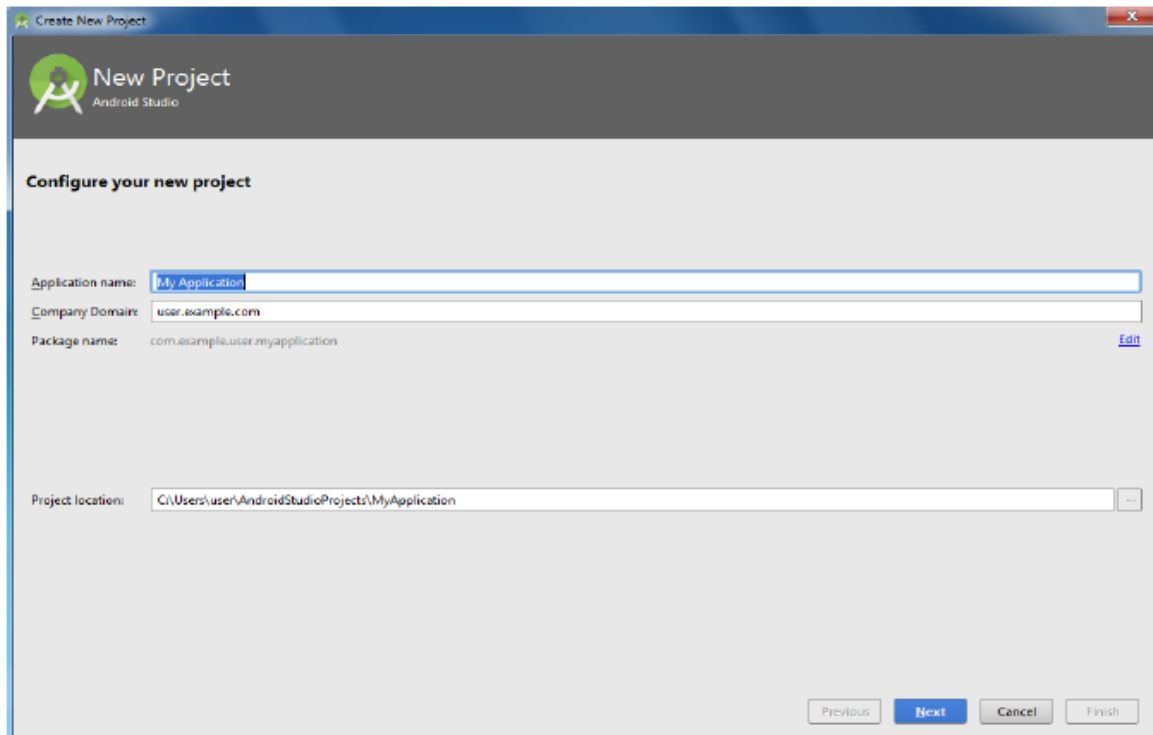
Εικόνα 6 - 7 : Configuration Settings

Έπειτα ακολουθεί η εγκατάσταση του προγράμματος.

Μετά την εγκατάσταση του προγράμματος ανοίγουμε το Android Studio όπου εμφανίζεται ένα παράθυρο στο οποίο πρέπει να επιλέξουμε τις ρυθμίσεις για το project. Για την συγκεκριμένη εργασία επιλέχθηκε **Start a new Android Studio Project** , όπως φαίνεται και στην *Εικόνα 6 - 8* . ενώ στην *Εικόνα 6 – 9*, πρέπει να δοθεί όνομα για την main class.



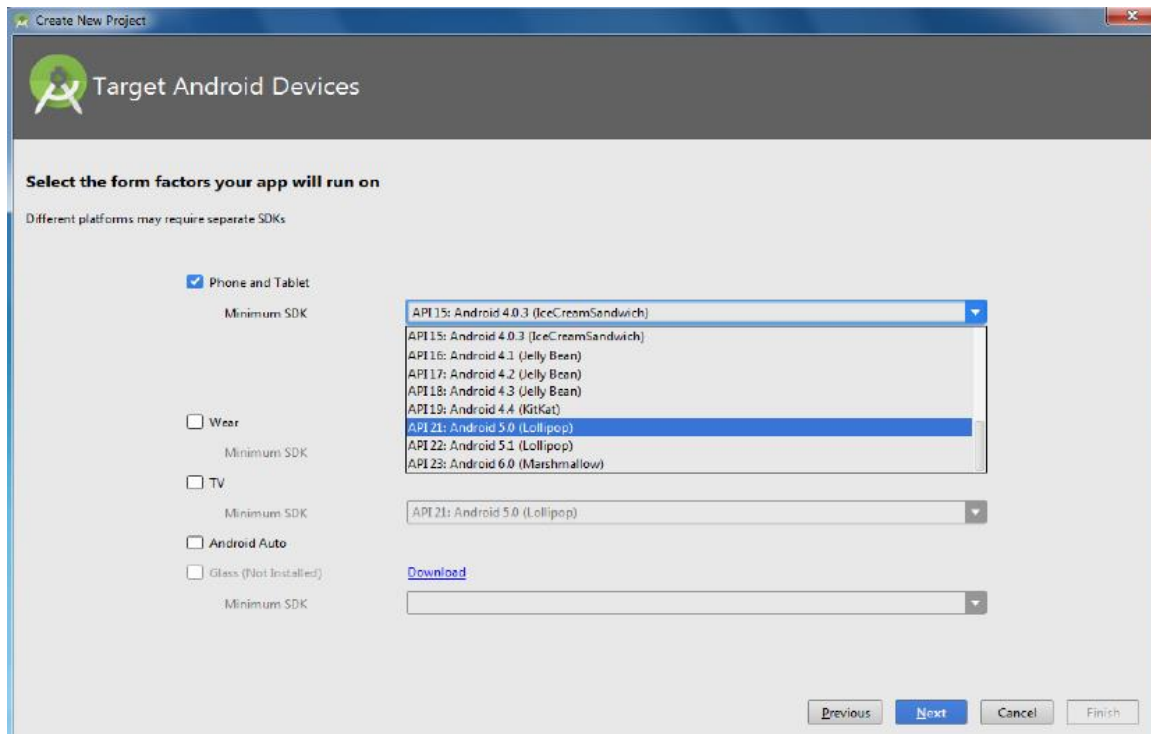
Εικόνα 6 - 8 : Quick Start



Εικόνα 6 - 9 : Configure new project

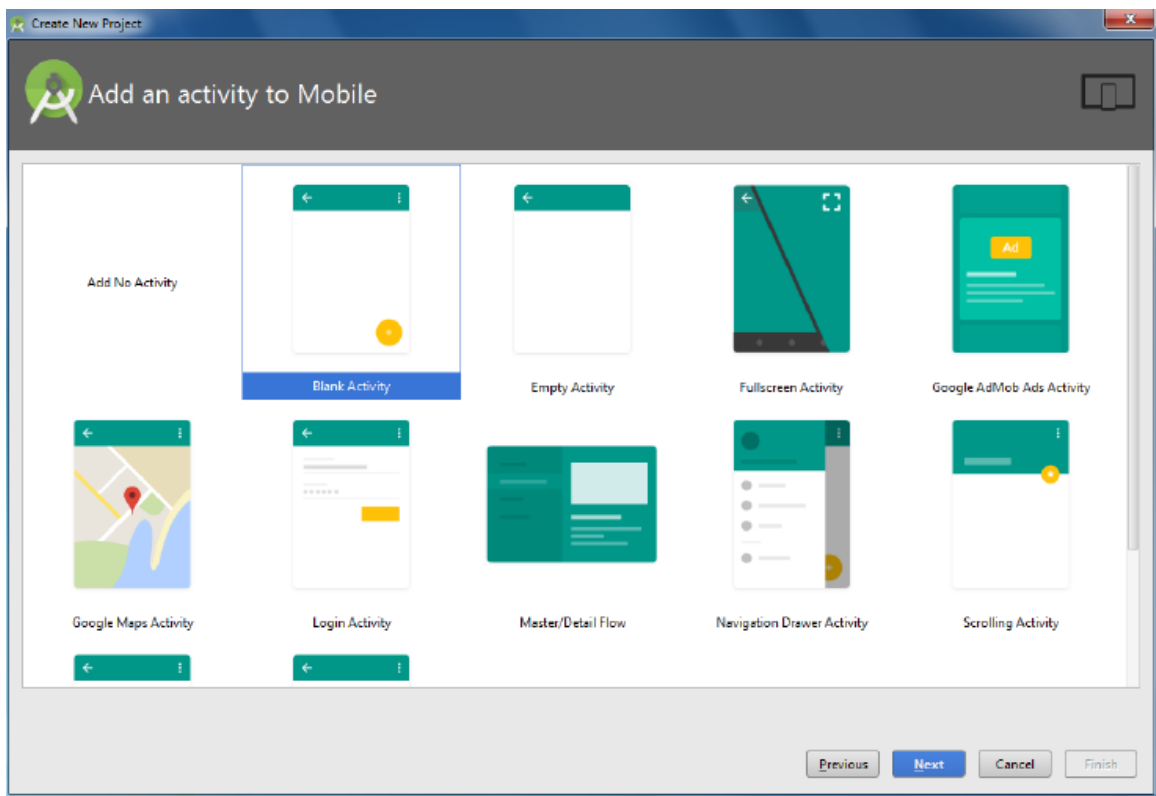
Στο **Application name** δίνουμε το όνομα της κλάσης. Ενώ στο **Company Domain** πρέπει να δώσουμε ένα χαρακτηριστικό όνομα για το πακέτο της εφαρμογής αφού αυτό θα είναι το όνομα του πακέτου που θα φαίνεται όταν δημοσιευτεί στο google play

Παρακάτω, μπορούμε να επιλέξουμε τι μορφή θέλουμε να υποστηρίζεται από την εφαρμογή. Το επίπεδο API (Application Programming Interface), είναι η διεπαφή των προγραμματιστικών διαδικασιών που παρέχει ένα λειτουργικό σύστημα, βιβλιοθήκη ή εφαρμογή.



Εικόνα 6 - 10 : Select form factors

Επόμενο βήμα είναι να προσθέσουμε ένα Activity (Δραστηριότητα) στην εφαρμογή μας, η παρακάτω εικόνα εμφανίζει μια μεγάλη γκάμα από Activities, αναλόγως της επιλογής που κάναμε στο παραπάνω βήμα.



Εικόνα 6 - 11 : Εισαγωγή Activity

6.2 Η εργασία μας

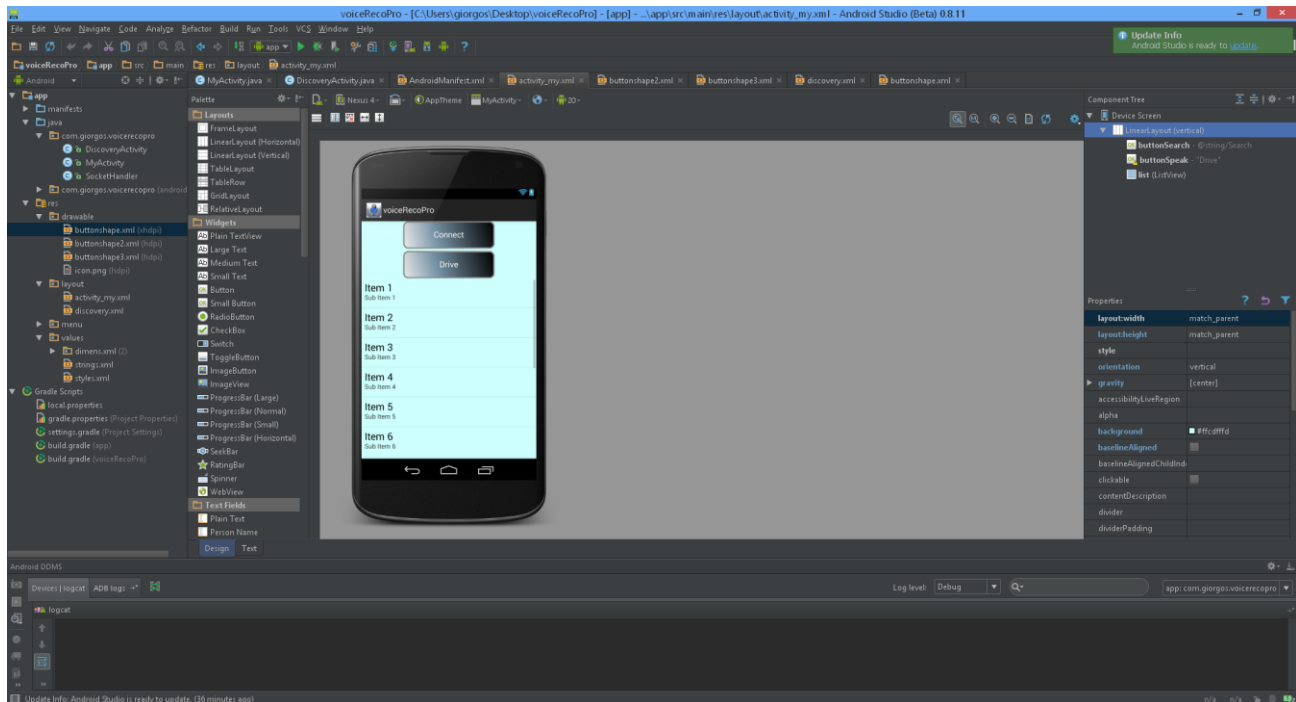
Σε αυτό το σημείο θα εξηγήσουμε και θα παρουσιάσουμε τα σημαντικότερα σημεία της εφαρμογής και κάποια σημεία του κώδικα, για να δούμε πως δουλεύει η εφαρμογή.

Η εφαρμογή που δημιουργήσαμε ονομάζεται “voiceRecoPro”



Εικόνα 6 - 12 : voiceRecoPro

6.2.1 Σημαντικά αρχεία Android Studio



Εικόνα 6 - 13 : Αρχεία Android Studio

- **Αρχείο AndroidManifest.xml**

Ένα από τα πιο σημαντικά αρχεία για την δημιουργία μιας εφαρμογής είναι το “AndroidManifest.xml”, καθώς χρησιμοποιείται για την ρύθμιση και τον καθορισμό των παραμέτρων μιας εφαρμογής. Δηλαδή, το όνομα, την έκδοση και από ποια στοιχεία αποτελείται. Το αρχείο AndroidManifest.xml βρίσκεται πάντα στο κορυφαίο επίπεδο κάθε εφαρμογής Android.

- **Φάκελος res**

Αυτός ο φάκελος περιέχει 3 υποφάκελους (όπως φαίνεται και στην Εικόνα 7-2) :

1. **drawable** → φωτογραφίες που χρησιμοποιούνται για την εφαρμογή
2. **layout** → δημιουργία των xml αρχείων από τα οποία παίρνουν πληροφορίες τα activity που καλούνται ώστε να παρουσιαστεί στο χρήστη η οθόνη της εφαρμογής.
3. **values** → στον φάκελο values αποθηκεύονται αρχεία xml, τα οποία περιέχουν τα χρώματα, το θέμα και πληροφορίες της εφαρμογής.

- **Αρχείο Gradle**

Σε αυτό το αρχείο συμπεριλαμβάνονται ο καθορισμός του έργου και πως πρέπει να δημιουργηθεί. Αυτό το αρχείο δημιουργείται αυτόματα.

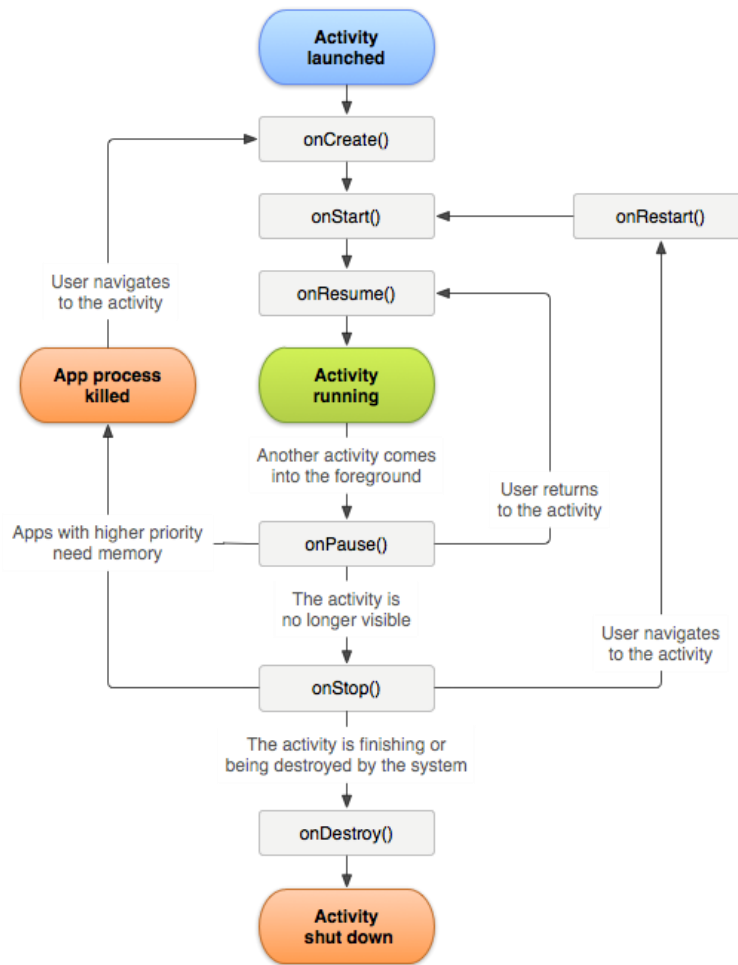
6.2.2 Activities

Μια εφαρμογή αποτελείται από activities τα οποία είναι συνδεδεμένα μεταξύ τους. Ένα activity αποτελεί βασικό στοιχείο μιας εφαρμογής. Ουσιαστικά, ένα activity είναι το γραφικό περιβάλλον μιας εφαρμογής το πώς δηλαδή αναπαριστάται η οθόνη της εφαρμογής, ώστε οι χρήστες να μπορούν για παράδειγμα να κάνουν κλήση ή να τραβήξουν μια φωτογραφία.

Κάθε Activity κλάση ορίζεται από τα event που καθορίζουν τον κύκλο ζωής ενός activity.

onCreate ()	Καλείται όταν δημιουργείται για πρώτη φορά ένα Activity. Ακολουθείται πάντα από την μέθοδο onStart ().
onRestart ()	Καλείται μετά που θα σταματήσει το Activity και πριν αυτό αρχίσει και πάλι. Ακολουθείται πάντα από την μέθοδο onStart ().
onStart ()	Καλείται όταν η δραστηριότητα γίνεται ορατή στο χρήστη. Ακολουθείται από την μέθοδο onResume () εάν η δραστηριότητα έρχεται στο προσκήνιο ή την μέθοδο onStop () αν γίνεται κρυφά.
onResume ()	Καλείται όταν το Activity θα αρχίσει να αλληλεπιδρά με το χρήστη. Σε αυτό το σημείο το Activity είναι στην κορυφή της στοίβας. Ακολουθείται πάντα από την μέθοδο onPause ().
onPause ()	Καλείται όταν το σύστημα είναι έτοιμο να ξεκινήσει την επιστροφή στην προηγούμενη δραστηριότητα. Αυτή η μέθοδος χρησιμοποιείται Ακολουθείται είτε από την μέθοδο onResume () εάν η δραστηριότητα επιστρέφει πίσω είτε την onStop () αν δεν είναι ορατό στο χρήστη.
onStop ()	Καλείται όταν η δραστηριότητα δεν είναι πλέον ορατή στο χρήστη, επειδή μια άλλη δραστηριότητα έχει ξεκινήσει και καλύπτει την πρώτη. Ακολουθούμενη είτε από την μέθοδο onRestart () αν αυτή η δραστηριότητα έρχεται πίσω για να αλληλεπιδράσει με το χρήστη ή την μέθοδο onDestroy () αν αυτή η δραστηριότητα αποσύρεται.
onDestroy ()	Η τελική κλήση που λαμβάνετε πριν καταστραφεί το Activity.

Πίνακας 6 - 1 :Μέθοδοι ενός Activity



Εικόνα 6 - 14 : Κύκλος ζωής Activity

6.2.3 Επεξήγηση Συναρτήσεων και Μεθόδων της Εφαρμογής

Συνάρτηση list () : Ο σκοπός της μεθόδου list(), είναι να εμφανίσει στην κατάλληλη λίστα τις συσκευές που έχουν σχηματίσει ζεύξη με την εφαρμογή.

```
public void list(View view) {
//   get paired devices
    pairedDevices=myBluetoothAdapter.getBondedDevices();

//   put it's one to the adapter
    for(BluetoothDevice device: pairedDevices){
        BTArrayAdapter.clear();
        BTArrayAdapter.add(device.getName()+"\n"+device.getAddress());
    }
    if(!BTArrayAdapter.isEmpty()) {
//       Toast.makeText(getApplicationContext(), "Show Paired Devices",
//       Toast.LENGTH_LONG).show();
    }
}
```

Πίνακας 6 - 2 : Συνάρτηση list ()

Σύνδεση σε συσκευή : Σε αυτό το σημείο είναι που συνδέονται η δυο ασύρματες συσκευές, η συγκεκριμένη συνάρτηση χρησιμοποιείται όταν επιλέξουμε μέσω της λίστα την συσκευή στη οποία θα συνδεθούμε. Η σύνδεσή επιτυγχάνεται χρησιμοποιώντας την διεύθυνση της συσκευής και δημιουργώντας ένα **createRfcommSocketToServiceRecord (my_UUID)**, το οποίο είναι αυτό που εξασφαλίζει την εξερχόμενη σύνδεση με την συσκευή. Έπειτα, καλείται η connect() που θα μας εξασφαλίσει την σύνδεση με τη ασύρματη συσκευή.

```
//   connect the device when item is click
BluetoothDevice connect_device=myBluetoothAdapter.getRemoteDevice(address);

try{
    socket=connect_device.createRfcommSocketToServiceRecord(my_UUID);
    socket.connect();
    SocketHandler.setSocket(socket);
} catch (IOException e){
    e.printStackTrace();
}
```

Πίνακας 6 - 3 : Σύνδεση με Συσκευή

Συνάρτησης onActivityResult : Η συσκευή καταλαβαίνει /ακούει 4 λέξεις :Go, back, left, right. Μέσω της συνάρτησης onActivityResult (int requestCode, int resultCode, Intent data) γίνεται έλεγχος για κάθε μία λέξη και καλείται η αντίστοιχη συνάρτηση για την

κάθε ενέργεια. Για παράδειγμα αν ακούσει την λέξη Go(Προχώρα) τότε καλεί την συνάρτηση forward. Παρακάτω φαίνεται το κομμάτι κώδικα ελέγχου .

```
if(matches.get(i).equals("go")){
    try {
        forward();
    }
    catch (IOException ex) {
        showMessage("SEND FAILED");
    }
}else if(matches.get(i).equals("back")){
    try {
        backward();
    } ..
}else if(matches.get(i).equals("left")){
    try {
        leftward();
    } ..
}else if(matches.get(i).equals("right")){
    try {
        rightward();
    } ..
}else{
    Toast.makeText(getApplicationContext(),"Dwse front,back,right or
left",Toast.LENGTH_LONG).show();
}
```

Πίνακας 6 - 4 : Συνάρτησης onActivityResult()

Συναρτήσεις forward(), backward(), leftward(), rightward() : Οι κάθε συνάρτηση στέλνει ένα γράμμα , ένα χαρακτήρα μέσω Bluetooth στο Arduino Και ανάλογα εκτελεί την αντίστοιχη κίνηση. Για παράδειγμα η συνάρτηση forward έχει σχεδιαστεί έτσι ώστε το αμαξίδιο να προχωράει μπροστά . Έτσι στέλνει τον χαρακτήρα 'F'.

```
private void forward() throws IOException {
    mmSocket=SocketHandler.getSocket();
    mmOutputStream = mmSocket.getOutputStream();
    mmInputStream = mmSocket.getInputStream();
    mmOutputStream.write('F');
```

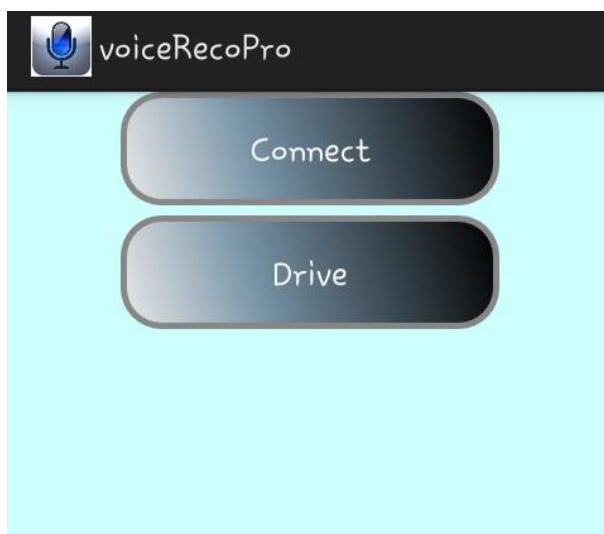
Πίνακας 6 - 5 : Συναρτήση forward()

Η ίδια λογική ισχύει και για τις συναρτήσεις backward(), leftward(), rightward().

6.2.4 Επεξήγηση της Εφαρμογής

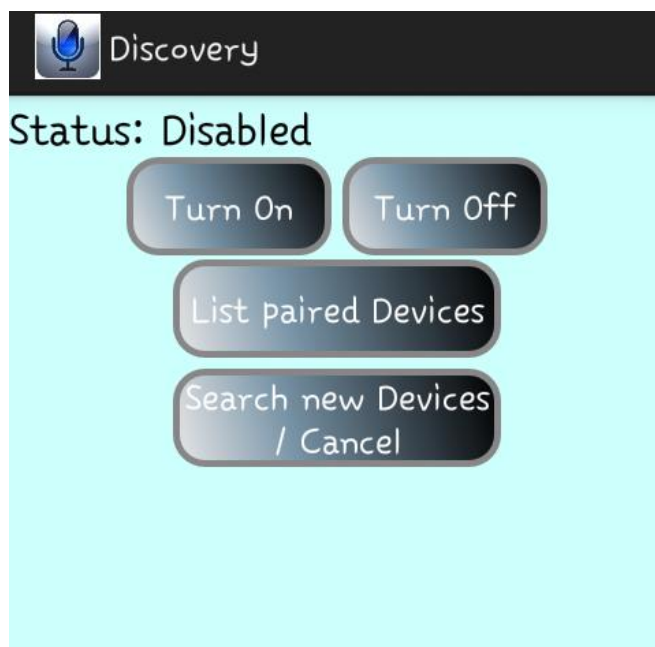
Σε αυτό το σημείο θα δείξουμε μέσω print screen το interface της εφαρμογής και θα εξηγήσουμε πως δουλεύει.

Στην παρακάτω εικόνα βλέπουμε την αρχική οθόνη της εφαρμογής όπου έχουμε δύο επιλογές η μια είναι το κουμπί “Connect” και το κουμπί “Drive”



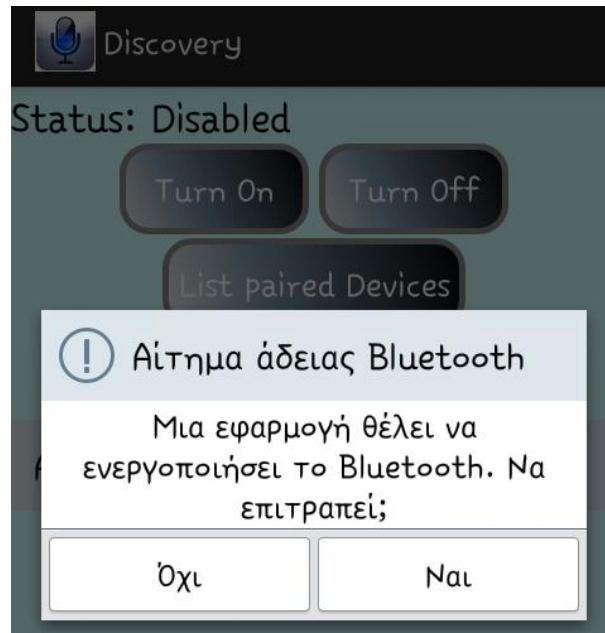
Εικόνα 6 - 15 : Οθόνη Εφαρμογής 1

Πατώντας το κουμπί “Connect” εμφανίζεται μια δεύτερη οθόνη (Εικόνα 7-4) στην οποία ανοίγουμε τα Bluetooth πατώντας το κουμπί “Turn on”



Εικόνα 6 - 16 : Οθόνη Εφαρμογής 2

Στην εφαρμογή έχουμε τοποθετήσει κάποια κουμπιά όπως την ενεργοποίηση και απενεργοποίηση του Bluetooth. Όπως βλέπουμε στην παρακάτω εικόνα πατώντας το κουμπί “Turn on” μας εμφανίζει ένα αίτημα άδειας αν θέλουμε επιτραπεί η ενεργοποίηθουν τα Bluetooth.

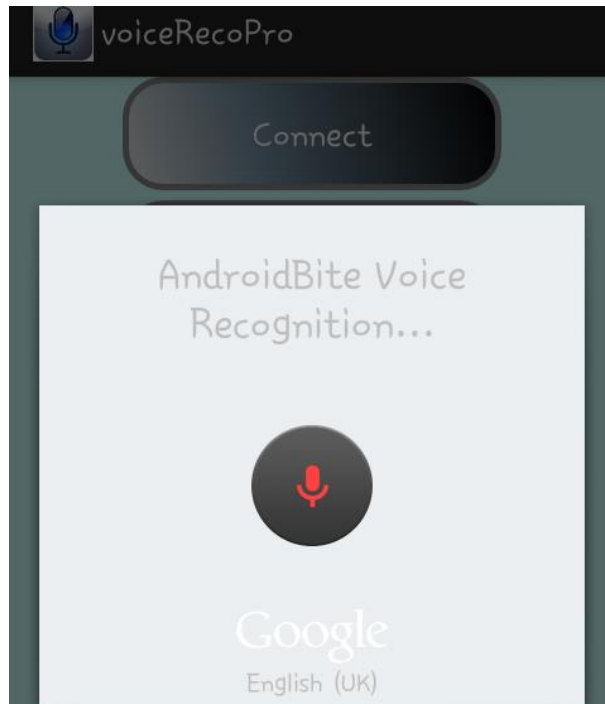


Εικόνα 6 - 17 : Οθόνη Εφαρμογής 3

Άλλες επιλογές που έχουμε είναι τα παρακάτω κουμπιά (Εικόνα 7 - 4) :

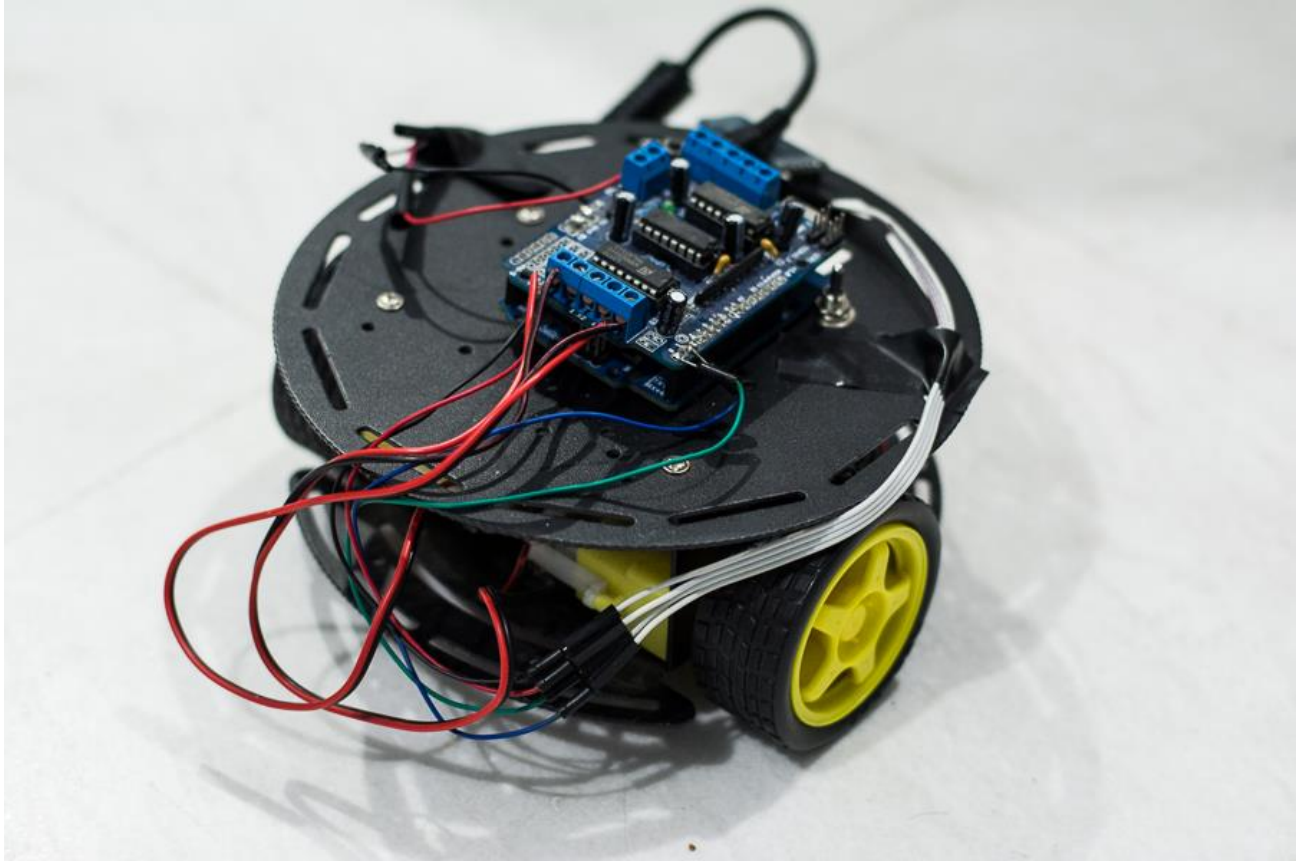
- **“Turn Off”** : Απενεργοποιήσει του Bluetooth.
- **“List paired Devices”** : Χρησιμοποιώντας το BluetoothAdapter, μπορούμε να βρούμε απομακρυσμένες συσκευές Bluetooth είτε μέσω της ανακάλυψης της συσκευής ή από την αναζήτηση στη λίστα των συζευγμένες συσκευές.
- **“Search new Devices/Cancel”** : Αναζήτηση νέων συσκευών για σύνδεση.

Αφού έχουμε συνδεθεί με την συσκευή , μπορούμε να ξεκινήσουμε την οδήγηση του αμαξιδίου , πηγαίνοντας πίσω στην πρώτη οθόνη (Εικόνα 7-3) και πατώντας αυτή τη φορά το κουμπί “Drive”. Με το πάτημα του κουμπιού εμφανίζεται η παρακάτω εικόνα. Σε αυτό το σημείο πατάμε το κουμπί με το μικρόφωνο και λέμε μια από της λέξεις που έχουμε θέσει ως αναγνωρίσιμες στον κώδικα.



Εικόνα 6 - 18 : Οθόνη Εφαρμογής 4

6.3 Αμαξίδιο Εργασίας



Εικόνα 6 - 19 :Αμαξίδιο Εργασίας

Στην παραπάνω εικόνα βλέπουμε το αμαξίδιο που δημιουργήσαμε για την εκπόνηση της πτυχιακής. Το συγκεκριμένο αυτοκίνητο αποτελείται από δύο ρόδες οι οποίες ελέγχονται η κάθε μια ξεχωριστά από δύο κινητήρες που βρίσκονται στο κάτω μέρος του αμαξιδίου.

Βασικό εξάρτημα είναι το Arduino Uno, το οποίο βρίσκεται πάνω στο αμαξίδιο μας. Πάνω στο Arduino Uno έχει τοποθετηθεί και ένα επιπλέον εξάρτημα (Motorshield), το οποίο έχει υποδοχές σύνδεσης για τα καλώδια που συνδέονται οι κινητήρες και για Bluetooth.

6.4 Κώδικας Arduino

```
#include <AFMotor.h>
AF_DCMotor motor3(3,MOTOR12_1KHZ);
AF_DCMotor motor4(4,MOTOR12_1KHZ);
char state='K';

void setup(){
  Serial.begin(9600);
  motor3.setSpeed(200);
  motor4.setSpeed(200);
}

void loop(){
  if(Serial.available()){
    state=Serial.read();
    if(state=='F'){
      motor3.run(RELEASE);
      motor4.run(RELEASE);
      motor3.run(FORWARD);
      motor4.run(FORWARD);
    }

    if(state=='B'){
      motor3.run(RELEASE);
      motor4.run(RELEASE);
      motor3.run(BACKWARD);
      motor4.run(BACKWARD);
    }
    if(state=='L'){
      motor3.run(RELEASE);
      motor4.run(FORWARD);
      delay(500);
      motor3.run(RELEASE);
      motor4.run(RELEASE);
    }
    if(state=='R'){

      motor3.run(FORWARD);
      motor4.run(RELEASE);
      delay(500);
      motor3.run(RELEASE);
      motor4.run(RELEASE);
    }
    if(state=='S'){
      motor3.run(RELEASE);
      motor4.run(RELEASE);
    }
  }
}
```

Πίνακας 6 - 6 : Κώδικας Arduino

Η κλάση AF_DCMotor παρέχει την ταχύτητα και την κατεύθυνση ελέγχου για έως και τέσσερις κινητήρες συνεχούς ρεύματος (εδώ χρησιμοποιούμε δύο κινητήρες) όταν χρησιμοποιείται με το Adafruit Motor Shield. Για να το χρησιμοποιήσουμε αυτό θα πρέπει πρώτα να προσθέσουμε την ακόλουθη γραμμή στην αρχή του κώδικα μας (πίνακας 6-7):

```
#include <AFMotor.h>
AF_DCMotor motor3(3,MOTOR12_1KHZ); // define motor on channel 3 with 1KHz
AF_DCMotor motor4(4,MOTOR12_1KHZ); // define motor on channel 4 with 1KHz
```

Πίνακας 6 - 7 : Εισαγωγή βιβλιοθήκης

Στη συνέχεια επιλέγουμε σε πιο κανάλι θα συνδεθούν οι κινητήρες και έπειτα επιλέγουμε τη συχνότητα. Αν δεν έχει επιλεγθεί η συχνότητα, τότε από προεπιλογή χρησιμοποιείται 1KHz.

Στο Arduino έχουν υλοποιηθεί 2 βασικές συναρτήσεις που αφορούν την συμπεριφορά του αμαξιδίου η συνάρτηση setup() και η συνάρτηση loop().

Συνάρτηση **setup ()** : Η συνάρτηση αυτή υλοποιεί τις αρχικές ρυθμίσεις συνδέσεις του αμαξιδίου. Πιο συγκεκριμένα, για την μετάδοση των δεδομένων χρησιμοποιεί σειριακή σύνδεση με ταχύτητα των δεδομένων (data rate) στα 9600 bps. Επιπλέον, ορίζει την ταχύτητα του αμαξιδίου με αρχική ταχύτητα της αριστερή και της δεξιάς ρόδας στα 200 bps. Παράδειγμα της συνάρτησης setup() φαίνεται παρακάτω:

```
void setup(){
  Serial.begin(9600); //Set the baud rate to your Bluetooth module
  motor3.setSpeed(200); //set the left wheel speed to 200
  motor4.setSpeed(200); //set the right wheel speed to 200
}
```

Πίνακας 6 - 8 : Συνάρτηση setup()

Συνάρτηση **loop()** : Εφόσον έχει γίνει το setup η συνάρτηση αυτή λαμβάνει τα διαθέσιμα bytes (χαρακτήρες) τα οποία παίρνανε από την σειριακή (μόνο αν υπάρχουν) μέσω της συνάρτησης available() και τα διαβάζει μέσω της συνάρτησης read() . Στη συνέχεια γίνονται έλεγχοι αν έχουν έρθει οι χαρακτήρες (F=Forward, B=Back, L=Left , R=Rigth) που στέλνονται από το αρχείο "Myactivity.java"για κάθε κίνηση του αμαξιδίου . Παραδείγματος χάριν αν έχει έρθει ο χαρακτήρας (bytes) 'F' τότε ελευθέρωσε την

αριστερή ρόδα και την δεξιά για να προχωρήσει το αμαξίδιο μπροστά. Με την ίδια λογική δουλεύει και η κίνηση προς τα πίσω.

```
void loop() {
  if (Serial.available()) {
    state = Serial.read();
    if (state == 'F') {
      motor3.run(RELEASE);
      motor4.run(RELEASE);
      motor3.run(FORWARD);
      motor4.run(FORWARD);
    }

    if (state == 'B') {
      motor3.run(RELEASE);
      motor4.run(RELEASE);
      motor3.run(BACKWARD);
      motor4.run(BACKWARD);
    }
  }
}
```

Πίνακας 6 - 9 : Συνάρτηση loop()

Για να μπορέσει το αμαξίδιο να ακούσει την κίνηση αριστερά ελέγχει αν έχει έρθει ο χαρακτήρας (bytes) 'L' αν ισχύει εκτελεί τα παρακάτω βήματα:

ΒΗΜΑ 1^ο : ελευθερώνει την αριστερή ρόδα

ΒΗΜΑ 2^ο : πηγαίνει μπροστά με την δεξιά ρόδα

ΒΗΜΑ 3^ο : περιμένει για 500 ms

ΒΗΜΑ 4^ο : ελευθερώνει και τις δύο ρόδες για να προχωρήσει (με την ίδια λογική δουλεύει και η κίνηση προς τα δεξιά).

```
if (state == 'L') {
  motor3.run(RELEASE); //turn motor3 off
  motor4.run(FORWARD); //rotate the motor clockwise
  delay(500); // waits for 500 ms
  motor3.run(RELEASE);
  motor4.run(RELEASE);
}
if (state == 'R') {

  motor3.run(FORWARD);
  motor4.run(RELEASE);
  delay(500);
  motor3.run(RELEASE);
  motor4.run(RELEASE);
}
```

Πίνακας 6 - 10 : Κίνηση αμαξιδίου αριστερά ή δεξιά

Τέλος για να σταματήσει την κίνηση του αμαξιδίου ελέγχει αν έχει έρθει ο χαρακτήρας 'S' και δίνει εντολή να σταματήσουν οι ρόδες.

```
if(state=='S'){  
    motor3.run(RELEASE); //turn motor3 off  
    motor4.run(RELEASE); //turn motor4 off  
}
```


ΚΕΦΑΛΑΙΟ 7 : ΣΥΜΠΕΡΑΣΜΑΤΑ

Σκοπός της συγκεκριμένης πτυχιακής ήταν η υλοποίηση μιας εφαρμογής για συσκευή Android. Στην πτυχιακή παρουσιάστηκε αρκετά η πλατφόρμα του Android και του Arduino.

Το αποτέλεσμα της εργασίας είναι ένα αμαξίδιο που ελέγχεται από μια συσκευή. Η παρούσα πτυχιακή εργασία έχει ως στόχο να παρέχει απλές οδηγίες για άτομα που ενδιαφέρονται για την κατασκευή ρομπότ.

Μελλοντική εξέλιξη της εφαρμογής, θα μπορούσε να είναι η σύνδεση της με ένα αυτοκίνητο το οποίο θα μπορούμε να το κινούμε δίνοντας του εντολές ενώ βρισκόμαστε σε απόσταση. Επιπλέον, η χαρακτηριστική λειτουργία αυτής της εφαρμογής θα μπορούσε να χρησιμοποιηθεί από άτομα με φυσικούς περιορισμούς ώστε να μπορέσουν να αντισταθμίσουν τις ικανότητες τους.

Παράρτημα

Στο παράρτημα αυτό παρατίθεται ο κώδικας ανάπτυξης της παρούσας εφαρμογής.

MyActivity.class

```
package com.giorgos.voicerecopro;

import android.app.Activity;
import android.bluetooth.BluetoothSocket;
import android.content.Intent;
import android.content.pm.PackageManager;
import android.content.pm.ResolveInfo;
import android.os.Bundle;
import android.speech.RecognizerIntent;
import android.view.Menu;
import android.view.MenuItem;
import android.view.View;
import android.widget.AdapterView;
import android.widget.Button;
import android.widget.ListView;
import android.widget.Toast;
import java.io.IOException;
import java.io.InputStream;
import java.io.OutputStream;
import java.util.ArrayList;
import java.util.List;

public class MyActivity extends Activity {

    private Button buttonSearch, buttonSpeak;
    private ListView resultList;
    private static final int REQUEST_CODE=1234;
        OutputStream mmOutputStream;
        InputStream mmInputStream;
    private BluetoothSocket mmSocket;
        @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_my);

        buttonSearch= (Button) findViewById(R.id.buttonSearch);
        buttonSearch.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                Intent next = new Intent(MyActivity.this,DiscoveryActivity.class);
                startActivity(next);
            }
        });

        buttonSpeak=(Button) findViewById(R.id.buttonSpeak);
        resultList=(ListView) findViewById(R.id.list);
        PackageManager pm = getPackageManager();
```

```

        List<ResolveInfo> activities=pm.queryIntentActivities(new
Intent(RecognizerIntent.ACTION_RECOGNIZE_SPEECH),0);
if (activities.size()==0){
buttonSpeak.setEnabled(false);
        Toast.makeText(getApplicationContext(),"Recognizer Not Found",Toast.LENGTH_LONG
).show();
    }
buttonSpeak.setOnClickListener(new View.OnClickListener() {
@Override
public void onClick(View view) {
        startVoiceRecognitionActivity();
    }
});
}
nb

private void startVoiceRecognitionActivity(){
    Intent intent=new Intent(RecognizerIntent.ACTION_RECOGNIZE_SPEECH);
    intent.putExtra(RecognizerIntent.EXTRA_LANGUAGE_MODEL,
        RecognizerIntent.LANGUAGE_MODEL_FREE_FORM);
    intent.putExtra(RecognizerIntent.EXTRA_PROMPT,
        "AndroidBite Voice Recognition...");
    startActivityForResult(intent,REQUEST_CODE);
}

@Override
protected void onActivityResult(int requestCode, int resultCode, Intent data) {
if(requestCode==REQUEST_CODE && resultCode==RESULT_OK){
    ArrayList<String> matches =
data.getStringArrayListExtra(RecognizerIntent.EXTRA_RESULTS);
    for(int i=0;i<matches.size();i++){
if(matches.get(i).equals("go")){
        try {
            forward();
        }
        catch (IOException ex) {
            showMessage("SEND FAILED");
        }
    }else if(matches.get(i).equals("back")){
        try {
            backward();
        }
        catch (IOException ex) {
            showMessage("SEND FAILED");
        }
    }else if(matches.get(i).equals("left")){
        try {
            leftward();
        }
        catch (IOException ex) {
            showMessage("SEND FAILED");
        }
    }else if(matches.get(i).equals("right")){
        try {
            rightward();
        }
        catch (IOException ex) {
            showMessage("SEND FAILED");
        }
    }
}
}

```

```

        }
        }else{
            Toast.makeText(getApplicationContext(),"Dwse front,back,right or
left",Toast.LENGTH_LONG).show();
        }
    }
    resultList.setAdapter(new ArrayAdapter<String>(this,
        android.R.layout.simple_list_item_1, matches));
}
super.onActivityResult(requestCode, resultCode, data);
}
private void forward() throws IOException {
    mmSocket=SocketHandler.getSocket();
    mmOutputStream = mmSocket.getOutputStream();
    mmInputStream = mmSocket.getInputStream();
    mmOutputStream.write('F');
}
private void backward() throws IOException{
    mmSocket=SocketHandler.getSocket();
    mmOutputStream = mmSocket.getOutputStream();
    mmInputStream = mmSocket.getInputStream();
    mmOutputStream.write('B');
}
private void leftward() throws IOException{
    mmSocket=SocketHandler.getSocket();
    mmOutputStream = mmSocket.getOutputStream();
    mmInputStream = mmSocket.getInputStream();
    mmOutputStream.write('L');
}
private void rightward() throws IOException{
    mmSocket=SocketHandler.getSocket();
    mmOutputStream = mmSocket.getOutputStream();
    mmInputStream = mmSocket.getInputStream();
    mmOutputStream.write('R');
}

private void showMessage(String theMsg) {
    Toast msg = Toast.makeText(getApplicationContext(),
        theMsg, (Toast.LENGTH_LONG));
    msg.show();
}
}
}

```

DiscoveyActivity.class

```

package com.giorgos.voicerecopro;

import android.app.Activity;
import android.bluetooth.BluetoothAdapter;
import android.bluetooth.BluetoothDevice;
import android.bluetooth.BluetoothSocket;
import android.content.BroadcastReceiver;
import android.content.Context;
import android.content.Intent;
import android.content.IntentFilter;
import android.os.Bundle;
import android.view.View;
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 java.io.IOException;
import java.io.InputStream;
import java.io.OutputStream;
import java.net.Socket;
import java.util.Set;
import java.util.UUID;

/**
 * Created by giorgos on 9/19/2014.
 */
public class DiscoveryActivity extends Activity {

    private static final int REQUEST_ENABLE_BT=1;

    private Button onBtn, offBtn, listBtn, findBtn;
    private TextView text;
    private BluetoothAdapter myBluetoothAdapter;
    private Set<BluetoothDevice> pairedDevices;
    private ListView myListView;
    private ArrayAdapter<String> BTArrayAdapter;
    private final UUID my_UUID = UUID.fromString("00001101-0000-1000-8000-00805f9b34fb");
    private BluetoothSocket socket;
    private String status="Status: ";
    /** Called when the activity is first created. */

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.discovery);
        //take an instance of BluetoothAdapter - Bluetooth radio
        myBluetoothAdapter =BluetoothAdapter.getDefaultAdapter();
        // elenxoume an ipostirizei bluetooth i siskevi
        if(myBluetoothAdapter==null){
            onBtn.setEnabled(false);
            offBtn.setEnabled(false);
            listBtn.setEnabled(false);
            findBtn.setEnabled(false);
            text.setText(status+"not supported");
        }else{
            text=(TextView) findViewById(R.id.bluetoothstate);

```

```

        if(myBluetoothAdapter.isEnabled()){
            text.setText(status+"Enabled");
        }else{
            text.setText(status+"Disabled");
        }
    }
    onBtn=(Button) findViewById(R.id.turnOn);
    onBtn.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View v) {
//            TODO Auto-generated method stub
            on(v);
        }
    });

    offBtn=(Button) findViewById(R.id.turnOff);
    offBtn.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View v) {
//            TODO Auto-generated method stub
            off(v);
        }
    });

    listBtn=(Button) findViewById(R.id.paired);
    listBtn.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View v) {
//            TODO Auto-generated method stub
            list(v);
        }
    });
    findBtn=(Button) findViewById(R.id.search);
    findBtn.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View v) {
//            TODO Auto-generated method stub
            find(v);
        }
    });
//    myListView=(ListView) findViewById(R.id.listView1);
//    create the arrayAdapter that contains the BTDevices, and set it to the ListView
    BTArrayAdapter = new ArrayAdapter<String>(this,android.R.layout.simple_list_item_1);
    myListView.setAdapter(BTArrayAdapter);
}

myListView.setOnItemClickListener(new AdapterView.OnItemClickListener() {
    @Override
    public void onItemClick(AdapterView<?> adapterView, View view, int position, long id) {

        myBluetoothAdapter.cancelDiscovery();
        final String info=((TextView)view).getText().toString();

        String address=info.substring(info.length()-17 );

        Toast.makeText(getApplicationContext(),address,Toast.LENGTH_LONG).show();
//        connect the device when item is click
        BluetoothDevice connect_device=myBluetoothAdapter.getRemoteDevice(address);

        try{

```

```

        socket=connect_device.createRfcommSocketToServiceRecord(my_UUID);
        socket.connect();
        SocketHandler.setSocket(socket);

        }catch (IOException e){
            e.printStackTrace();
        }

    }
});
}

public void on(View view){
    if(!myBluetoothAdapter.isEnabled()){
        Intent turnOnIntent=new Intent(BluetoothAdapter.ACTION_REQUEST_ENABLE);
        startActivityForResult(turnOnIntent,REQUEST_ENABLE_BT);

//        Toast.makeText(getApplicationContext(),"Bluetooth turned on",
//        Toast.LENGTH_LONG).show();
    }else{
//        Toast.makeText(getApplicationContext(),"Bluetooth is already on",
//        Toast.LENGTH_LONG).show();
    }
}

@Override
protected void onActivityResult(int requestCode, int resultCode, Intent data) {
    super.onActivityResult(requestCode, resultCode, data);
//    TODO Auto-generated method stub
    if(requestCode==REQUEST_ENABLE_BT){
        if(myBluetoothAdapter.isEnabled()){
            text.setText(status+"Enabled");
        }else{
            text.setText(status+"Disabled");
        }
    }
}

public void list(View view){
//    get paired devices
    pairedDevices=myBluetoothAdapter.getBondedDevices();

//    put it's one to the adapter
    for(BluetoothDevice device: pairedDevices){
        BTArrayAdapter.clear();
        BTArrayAdapter.add(device.getName()+"\n"+device.getAddress());}
    if(!BTArrayAdapter.isEmpty()) {
//        Toast.makeText(getApplicationContext(), "Show Paired Devices",
//        Toast.LENGTH_LONG).show();
    }
}

final BroadcastReceiver bReceiver = new BroadcastReceiver() {
    @Override
    public void onReceive(Context context, Intent intent) {

```



```

        String action=intent.getAction();
//      When discovery finds a device
        if(BluetoothDevice.ACTION_FOUND.equals(action)){
//      Get the BluetoothDevice object from the Intent
            BluetoothDevice device= intent.getParcelableExtra(BluetoothDevice.EXTRA_DEVICE);
//      add the name and the MAC of the object to the arrayAdapter
            BTArrayAdapter.add(device.getName()+"\n"+device.getAddress());
            BTArrayAdapter.notifyDataSetChanged();
        }
    }
};

public void find(View view){
    if(myBluetoothAdapter.isDiscovering()){
        myBluetoothAdapter.cancelDiscovery();
    }else {
        BTArrayAdapter.clear();
        myBluetoothAdapter.startDiscovery();
        registerReceiver(bReceiver,new IntentFilter(BluetoothDevice.ACTION_FOUND));
    }
}

public void off(View view){
    myBluetoothAdapter.disable();
    text.setText(status+"Disabled");

//      Toast.makeText(getApplicationContext(),"Bluetooth turned off",
//      Toast.LENGTH_LONG).show();
}

@Override
protected void onDestroy() {

    super.onDestroy();
    try {
        unregisterReceiver(bReceiver);
    }catch(IllegalArgumentException e){

    }
}
}
}

```

SocketHandler.java

```
package com.giorgos.voicerecopro;

import android.bluetooth.BluetoothSocket;

import java.net.Socket;

public class SocketHandler {
    private static BluetoothSocket socket;

    public static BluetoothSocket getSocket(){
        return socket;
    }

    public static void setSocket(BluetoothSocket socket){
        SocketHandler.socket = socket;
    }
}
```

Βιβλιογραφία

- [1] «wikipedia,» [Ηλεκτρονικό]. Available: <https://el.wikipedia.org/wiki/Arduino>.
- [2] «Arduino,» [Ηλεκτρονικό]. Available: <https://www.arduino.cc/en/Guide/Introduction>.
- [3] «Sparkfun,» [Ηλεκτρονικό]. Available: <https://learn.sparkfun.com/tutorials/what-is-an-arduino>.
- [4] «Arduino,» [Ηλεκτρονικό]. Available: <https://www.arduino.cc/en/Guide/Environment>.
- [5] «Seeedstudio,» [Ηλεκτρονικό]. Available: http://wiki.seeedstudio.com/wiki/Motor_Shield_V1.0.
- [6] «Adafruit,» [Ηλεκτρονικό]. Available: <https://learn.adafruit.com/adafruit-motor-shield-v2-for-arduino/overview>.
- [7] «Wikipedia-Bluetooth,» [Ηλεκτρονικό]. Available: <https://el.wikipedia.org/wiki/Bluetooth>.
- [8] «Android,» [Ηλεκτρονικό]. Available: <https://el.wikipedia.org/wiki/Android>.
- [9] «Android_(operating_system),» [Ηλεκτρονικό]. Available: [https://en.wikipedia.org/wiki/Android_\(operating_system\)](https://en.wikipedia.org/wiki/Android_(operating_system)).
- [10] «Developer Android,» [Ηλεκτρονικό]. Available: <https://developer.android.com/index.html>.
- [11] «Wikipedia-Bluetooth,» [Ηλεκτρονικό]. Available: <https://el.wikipedia.org/wiki/Bluetooth>.