

**ΑΕΙ ΠΕΙΡΑΙΑ Τ.Τ.
ΣΧΟΛΗ ΤΕΧΝΟΛΟΓΙΚΩΝ ΕΦΑΡΜΟΓΩΝ
ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΗΛΕΚΤΡΟΝΙΚΩΝ
ΥΠΟΛΟΓΙΣΤΙΚΩΝ ΣΥΣΤΗΜΑΤΩΝΤ.Ε.**

ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ

Κίνηση Δίτροχου Οχήματος με την βοήθεια Arduino και Android

**Ελευθέριος Κ. Κουτράκης
Ιωάννης Ι. Μαραγκάκης**

Εισηγητής: Δρ Ιωάννης Έλληνας, Καθηγητής

**ΑΘΗΝΑ
ΑΠΡΙΛΙΟΣ 2017**

ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ

Κίνηση Δίτροχου Οχήματος με την βοήθεια Arduino και Android

Ελευθέριος Κ. Κουτράκης

A.M. 43078

Ιωάννης Ι. Μαραγκάκης

A.M. 41773

Εισηγητής:

Δρ Ιωάννης Έλληνας, Καθηγητής

Εξεταστική Επιτροπή:

Ημερομηνία εξέτασης

ΔΗΛΩΣΗ ΣΥΓΓΡΑΦΕΑ ΠΤΥΧΙΑΚΗΣ ΕΡΓΑΣΙΑΣ

Οι κάτωθι υπογεγραμμένοι Κουτράκης Ελευθέριος του Κωνσταντίνου, με αριθμό μητρώου 43078 και Μαραγκάκης Ιωάννης του Ιωάννη, με αριθμό μητρώου 41773 φοιτητές του Τμήματος Μηχανικών Η/Υ Συστημάτων Τ.Ε. του Α.Ε.Ι. Πειραιά Τ.Τ. πριν αναλάβουμε την εκπόνηση της Πτυχιακής Εργασίας μας, δηλώνουμε ότι ενημερωθήκαμε για τα παρακάτω:

«Η Πτυχιακή Εργασία (Π.Ε.) αποτελεί προϊόν πνευματικής ιδιοκτησίας τόσο του συγγραφέα, όσο και του Ιδρύματος και θα πρέπει να έχει μοναδικό χαρακτήρα και πρωτότυπο περιεχόμενο.

Απαγορεύεται αυστηρά οποιοδήποτε κομμάτι κειμένου της να εμφανίζεται αυτούσιο ή μεταφρασμένο από κάποια άλλη δημοσιευμένη πηγή. Κάθε τέτοια πράξη αποτελεί προϊόν λογοκλοπής και εγείρει θέμα Ηθικής Τάξης για τα πνευματικά δικαιώματα του άλλου συγγραφέα. Αποκλειστικός υπεύθυνος είναι ο συγγραφέας της Π.Ε., ο οποίος φέρει και την ευθύνη των συνεπειών, ποινικών και άλλων, αυτής της πράξης.

Πέραν των όποιων ποινικών ευθυνών του συγγραφέα σε περίπτωση που το Ίδρυμα του έχει απονείμει Πτυχίο, αυτό ανακαλείται με απόφαση της Συνέλευσης του Τμήματος. Η Συνέλευση του Τμήματος με νέα απόφαση της, μετά από αίτηση του ενδιαφερόμενου, του αναθέτει εκ νέου την εκπόνηση της Π.Ε. με άλλο θέμα και διαφορετικό επιβλέποντα καθηγητή. Η εκπόνηση της εν λόγω Π.Ε. πρέπει να ολοκληρωθεί εντός τουλάχιστον ενός ημερολογιακού 6μήνου από την ημερομηνία ανάθεσης της. Κατά τα λοιπά εφαρμόζονται τα προβλεπόμενα στο άρθρο 18, παρ. 5 του ισχύοντος Εσωτερικού Κανονισμού.»

ΕΥΧΑΡΙΣΤΙΕΣ

Η παρούσα πτυχιακή εργασία ολοκληρώθηκε μετά από επίμονες προσπάθειες, σε ένα ενδιαφέρον γνωστικό αντικείμενο της δημιουργίας εφαρμογής Android και την κατασκευή ρομποτικού οχήματος με Arduino. Την προσπάθειά μας αυτή υποστήριξε ο επιβλέπων καθηγητής μας, Ιωάννης Έλληνας τον οποίο θα θέλαμε να ευχαριστήσουμε.

ΠΕΡΙΛΗΨΗ

Κάνοντας μια εισαγωγή στις εφαρμογές Android και στον μικροελεγκτή Arduino η παρούσα εργασία εξετάζει την δημιουργία δίτροχου οχήματος και κίνηση του με την βοήθεια Arduino και Android. Αρχικά, αναφέρονται τα βασικά γνωρίσματα του λογισμικού Android, η αρχιτεκτονική του, καθώς και η σχεδίαση και η υλοποίηση της εφαρμογής μας. Στην συνέχεια, γίνεται παρουσίαση των βασικών χαρακτηριστικών του μικροελεγκτή Arduino, διάφορες εφαρμογές του, καθώς και η παρουσίαση της κατασκευής μας. Στο τρίτο μέρος, παρουσιάζεται η δημιουργία δίτροχου ρομποτικού οχήματος, η σύνδεση του Arduino με το Android, ο σχεδιασμός για τον εντοπισμό εμποδίων με αισθητήρες, καθώς και η υλοποίηση της ισορροπίας του ρομποτικού οχήματος σε δύο ρόδες. Κλείνοντας, παρατείνονται τα συμπεράσματα μας και οι προοπτικές εξέλιξης της κατασκευής μας.

ABSTRACT

Making an introduction to Android applications and Arduino microcontroller, this paper examines the creation of a two-wheel vehicle which the movement will be controlled by Android and Arduino. Originally referred to the architecture and the basic features of the Android and then to the design and implementation of our application. Then we present some main features of Arduino microcontroller and some various applications of it as well as the presentation of our application. The third part presents the creation of two-wheel robotic vehicle, the Arduino connection with the Android, the planning to identify obstacles with sensors, and the implementation of the balance of the robotic vehicle on two wheels. Finally, we give our conclusions and prospects of our manufacturing development.

ΕΠΙΣΤΗΜΟΝΙΚΗ ΠΕΡΙΟΧΗ: Μικροελεγκτές και Εφαρμογή Android
ΛΕΞΕΙΣ ΚΛΕΙΔΙΑ: μικροελεγκτής, application, wiring, java, αισθητήρες

ΠΕΡΙΕΧΟΜΕΝΑ

1.	ΕΙΣΑΓΩΓΗ.....	13
1.1	Περιγραφή του αντικειμένου της πτυχιακής εργασίας	13
1.2	Ιστορική αναδρομή	14
1.2.1	Ιστορική αναδρομή Android.....	14
1.2.2	Ιστορική αναδρομή Arduino.....	16
1.3	Ανασκόπηση της πτυχιακής εργασίας	17
2.	ΕΦΑΡΜΟΓΗ ANDROID.....	19
2.1	Εισαγωγή.....	19
2.1.1	Εφαρμογές Λογισμικού.....	19
2.1.2	Εφαρμογές Android.....	20
2.1.3	Ανάπτυξη μιας εφαρμογής.....	21
2.2	Βασικά Χαρακτηριστικά.....	22
2.3	Οι εκδόσεις του Android.....	23
2.3.1	Android 1.0 - 1.1.....	24
2.3.2	Android 1.5.....	25
2.3.3	Android 1.6.....	26
2.3.4	Android 2.0 - 2.1.....	27
2.3.5	Android 2.2 - 2.2.3.....	28
2.3.6	Android 2.3 - 2.3.7.....	29
2.3.7	Android 3.0 - 3.2.6.....	30
2.3.8	Android 4.0 - 4.0.4.....	31
2.3.9	Android 4.1 - 4.3.1.....	32
2.3.10	Android 4.4 - 4.4.4.....	32
2.3.11	Android 5.0 – 5.1.1.....	33
2.3.12	Android 6.0 – 6.0.1.....	34
2.3.13	Android 7.0 - 7.1.1.....	35
2.4	Αρχιτεκτονική του Android.....	36
2.5	Παρουσίαση της εφαρμογής μας.....	41
3.	ΜΙΚΡΟΕΛΕΓΚΤΗΣ ARDUINO.....	65

3.1 Εισαγωγή.....	65
3.1.1 Εισαγωγή στους Μικροελεγκτές.....	65
3.1.2 Εισαγωγή στον Arduino.....	66
3.2 Εκδόσεις του Arduino.....	67
3.3 Arduino Shields.....	72
3.4 Χαρακτηριστικά του Arduino.....	74
3.5 Πλεονεκτήματα του Arduino.....	77
3.6 Παρουσίαση της κατασκευή μας.....	78
4. ΥΛΟΠΟΙΗΣΗ ΔΙΤΡΟΧΟΥ ΟΧΗΜΑΤΟΣ.....	87
4.1 Εξαρτήματα Κατασκευής.....	87
4.1.1 HC-06 Bluetooth Module.....	88
4.1.2 HC-SR04 Sensor.....	88
4.1.3 MiniQ Base.....	88
4.1.4 MPU6050.....	89
4.1.5 L293D Shield.....	90
4.2 Συνδεσμολογία Εξαρτημάτων.....	90
5. ΣΥΜΠΕΡΑΣΜΑΤΑ ΚΑΙ ΠΡΟΟΠΤΙΚΕΣ.....	93
5.1 Σύνοψη της πτυχιακής εργασίας	93
5.2 Προοπτικές	93
6. ΒΙΒΛΙΟΓΡΑΦΙΑ.....	95

ΚΑΤΑΛΟΓΟΣ ΠΙΝΑΚΩΝ

Πίνακας 2.1: Εκδόσεις του Android	24
Πίνακας 2.2: Αρχιτεκτονική του Android.....	36
Πίνακας 3.1: Χαρακτηριστικά του Arduino.....	74

ΚΑΤΑΛΟΓΟΣ ΕΙΚΟΝΩΝ

Εικόνα 1.1: Δίτροχο Όχημα.....	14
Εικόνα 1.2:T-Mobile G1	15
Εικόνα 2.1: Mobile Development.....	19
Εικόνα 2.2: Το λογότυποντου Android.....	20
Εικόνα 2.3: Περιβάλλον Android 1.0.....	24
Εικόνα 2.4: Περιβάλλον Android 1.5.....	25
Εικόνα 2.5: Περιβάλλον Android 1.6.....	26
Εικόνα 2.6: Περιβάλλον Android 2.0.....	27
Εικόνα 2.7: Περιβάλλον Android 2.2.....	28
Εικόνα 2.8: Περιβάλλον Android 2.3	29
Εικόνα 2.9: Περιβάλλον Android 3.0.....	30
Εικόνα 2.10: Περιβάλλον Android 4.0.....	31
Εικόνα 2.11: Περιβάλλον Android 4.1-4.3.....	32
Εικόνα 2.12: Περιβάλλον Android 4.4.....	33
Εικόνα 2.13: Logo Android 5.0-5.1.1	34
Εικόνα 2.14: Περιβάλλον Android 6.0.....	35
Εικόνα 2.15: Logo Android 7.0-7.1.1	35
Εικόνα 2.16: Κύρια οθόνη εφαρμογής	41
Εικόνα 2.17: Εφαρμογή OpenCv Manager.....	49
Εικόνα 3.1:Το λογότυπο του Arduino.....	67
Εικόνα 3.2: Arduino Uno.....	68
Εικόνα 3.3: Arduino LilyPad	68
Εικόνα 3.4: Arduino Leonardo.....	69
Εικόνα 3.5: Arduino Ethernet.....	69

Εικόνα 3.6: Arduino Mega 2560.....	70
Εικόνα 3.7: Arduino Yun.....	70
Εικόνα 3.8: Arduino Mini.....	71
Εικόνα 3.9: Arduino Due	71
Εικόνα 3.10: Arduino Motor Shield.....	73
Εικόνα 3.11: Arduino Bluetooth Shield.....	73
Εικόνα 4.1: HC-06 Bluetooth Module.....	87
Εικόνα 4.2: HC-SR04 Sensor.....	88
Εικόνα 4.3: MiniQ.....	88
Εικόνα 4.4: Συνδεδεμένο MiniQ.....	89
Εικόνα 4.5: MPU6050.....	89
Εικόνα 4.6: L293D Shield.....	90
Εικόνα 4.7: Συνδεσμολογία Bluetooth με Arduino.....	91
Εικόνα 4.8: Συνδεσμολογία HC-SR04 με Arduino.....	92
Εικόνα 4.9: Συνδεσμολογία MPU6050 με Arduino.....	92

ΣΥΝΤΟΜΟΓΡΑΦΙΕΣ

PID PROPORTIONAL INTEGRAL DERIVATE

ROM READ ONLY MEMORY

CPU CENTRAL PROCCESsing UNIT

EEPROM ELECTRICALLY ERASABLE PROGRAMMABLE READ ONLY
MEMORY

USB UNIVERSAL SERIAL BUS

RAM RANDOM ACCESS MEMORY

WiFi WIRELESS FIDELITY

SD SECURE DIGITAL

CDMA CODE DIVISION MULTIPLE ACCESS

WVGA WIDE VIDEO GRAPHICS ARRAY

MP MEGAPIXEL

PWM PULSE WIDTH MODULATION

ICSP IN CIRCUIT SERIAL PROGRAMMING

MHz MEGAHEARTZ

mA MILLIAMPERE

V VOLT

UUID UNIVERSAL UNIQUE IDENTIFIER

ΚΕΦΑΛΑΙΟ 1

ΕΙΣΑΓΩΓΗ

1.1 Περιγραφή του αντικειμένου της πτυχιακής εργασίας

Σε αυτό το κεφάλαιο θα εξηγήσουμε με απλά λόγια το θέμα της πτυχιακής εργασίας μας. Η εργασία έχει τίτλο «*Κίνηση 2-τροχου με τη χρήση Android και Arduino*». Από το τίτλο καταλαβαίνουμε ότι έχουμε να κάνουμε με δυο διαφορετικά πράγματα, το Arduino και το Android. Ας ξεκινήσουμε με το Android, σε αυτήν την εφαρμογή έχουμε δημιουργήσει την επικοινωνία με τον Arduino. Τώρα πως γίνεται αυτό, αρχικά δημιουργούμε κώδικα ο οποίος ενεργοποιεί την κάμερα του κινητού μας και μέσω επεξεργασίας και με τη βοήθεια της βιβλιοθήκης «OpenCV» ξεχωρίζουμε από την εικόνα που παίρνουμε από τη κάμερα το μεγαλύτερο αντικείμενο χρώματος μαύρου, δηλαδή από ότι βλέπει η κάμερα βάζει σε ένα περίγραμμα το μεγαλύτερο σε έκταση μαύρο αντικείμενο. Αφού τώρα έχουμε εντοπίσει το αντικείμενο στην οθόνη του κινητού μας με τον κατάλληλο κώδικα ελέγχουμε ένα το αντικείμενο μας βρίσκεται δεξιά, αριστερά ή στο κέντρο της οθόνης. Την στιγμή που γίνεται ο έλεγχος με το που βρίσκετε το μαύρο αντικείμενο ταυτόχρονα μέσω τις επικοινωνίας με τον Arduino στέλνετε το κατάλληλο byte. Επίσης έχουμε φτιάξει ένα επιπλέον layer στην εφαρμογή το οποίο μας βοηθάει στη ρύθμιση του PID (θα εξηγήσουμε στη συνέχεια) και αποτελείται από «buttons» και «textbox» και με το πάτημα του καθενός στέλνετε ένα διαφορετικό byte μέσω Bluetooth στον Arduino. Αυτό ήταν με λίγα λόγια η περιγραφή της εφαρμογής Android. Ας πάμε και στον Arduino, για να επιτύχουμε τη κατασκευή του 2-τροχου χρειαζόμαστε κάποια εξαρτήματα. Τα εξαρτήματα που χρειαζόμαστε είναι αρχικά ο Arduino όπου θα συνδεθούν όλα πάνω σε αυτόν. Για την κίνηση έχουμε 2 ρόδες με τα motors, μια πλαστική βάση όπου πάνω σε αυτήν δημιουργήσαμε το σκελετό της κατασκευής μας από ξύλο και πάνω στο ξύλο τοποθετήσαμε το Arduino, εισήγαμε πάνω στο Arduino ένα shield για τη κίνηση των 2 motor, το Bluetooth module για την επικοινωνία με το Android και τέλος 2 αισθητήρες έναν όπου με τη χρήση ηχητικών κυμάτων εντοπίζει τα εμπόδια μπροστά του και έναν αισθητήρα ο οποίος είναι «accelerometer» και «gyroscope»

και μας βοηθάει στην ισορροπία του 2-τροχου. Όλα αυτά είναι συνδεδεμένα στα «pins» του Arduino και είναι έτοιμα για χρήση. Με τον κατάλληλο κώδικα (όπου εξηγείται στην υπόλοιπη πτυχιακή εργασία) ενώνουμε όλα τα κομμάτια μεταξύ τους και ο Arduino είναι έτοιμος να εκτελέσει των κώδικα για την καθοδήγηση του 2-τροχου και για την ισορροπία του.



Εικόνα 1.1 Δίτροχο Όχημα

1.2 Ιστορική αναδρομή

1.2.1 Ιστορική αναδρομή Android

Η Google, έχοντας εντοπίσει αυξημένη χρήση του internet και αναζητήσεων στον παγκόσμιο ιστό μέσω κινητών συσκευών (mobile devices) εξαγοράζει το 2005 την Android Inc με σκοπό την ανάπτυξη ενός Linux-based λειτουργικό σύστημα για κινητά τηλέφωνα και άλλες κινητές συσκευές. Ο στόχος τους ήταν ένα ευέλικτο λειτουργικό σύστημα καθώς και αναβαθμίσιμο.

Το 2007 δημιουργείται ένας οργανισμός που αποτελείται από μεγάλο αριθμό εταιρειών τηλεπικοινωνιακού εξοπλισμού καθώς και εταιρείες πληροφορικής όπως η Google, η T-Mobile, η Motorola, η Samsung, η Sony Ericsson, η Intel, η Vodafone, η Toshiba κ.α. με όνομα Open Handset Alliance (<http://www.openhandsetalliance.com>) και σκοπό την έρευνα και την ανάπτυξη

τεχνολογιών για την παραγωγή συσκευών που θα διευκολύνουν τόσο τους παρόχους κινητής τηλεφωνίας όσο και τους κατασκευαστές κινητών τηλεφώνων αλλά και τους προγραμματιστές εφαρμογών. Ξεκίνησαν με στόχο να αναπτύξουν open standards για mobile συσκευές και έτσι το πρώτο τους προϊόν ήταν η πλατφόρμα Android που ήταν χτισμένη πάνω στον πυρήνα του Linux. Τα μέλη της συμμαχίας δεσμεύτηκαν να παρέχουν τις τεχνολογίες αυτές βάσει του μοντέλου ανοιχτού πηγαίου κώδικα Apache. Η πρώτη 'early look' έκδοση του Android SDK δημοσιεύτηκε το Νοέμβριο του 2007, ενώ το πρώτο smartphone που έκανε χρήση λειτουργικού Android ήταν το G1 της T-Mobile.



Εικόνα 1.2 T-Mobile G1

Λίγες μέρες αργότερα η Google ανακοίνωσε την διαθεσιμότητα του Android SDK Release Candidate 1.0. Το Android ήταν διαθέσιμο σαν opensource λογισμικό από τον Οκτώβριο του 2008. Μέσω του Apache, ιδιωτικές εταιρίες μπορούσαν να προσθέσουν τις δικές τους εφαρμογές και επεκτάσεις και να τις πουλήσουν χωρίς να υποχρεούνται να τις υποβάλλουν στην open-source κοινότητα. Στα τέλη του 2008 η Google ανακοίνωσε μια συσκευή με το όνομα Android Dev Phone 1 που είχε την δυνατότητα να τρέχει Android εφαρμογές χωρίς να είναι άμεσα συνδεδεμένη με κάποιο δίκτυο κινητής τηλεφωνίας. Ο στόχος της συσκευής αυτής ήταν να επιτρέψει στους προγραμματιστές να πειραματιστούν με μια πραγματική συσκευή με Android χωρίς κάποιο συμβόλαιο με εταιρία κινητής τηλεφωνίας. Οι συσκευές Android άρχισαν να διαδίδονται με γρήγορο ρυθμό κυρίως λόγω της δυνατότητας της πλατφόρμας να εκμεταλλεύεται το μοντέλο cloud computing αλλά και της έμφυτης υποστήριξης για συνεργασία με μία σχεσιακή βάση δεδομένων (SQLite). Ακολούθησαν αρκετές αναβαθμισμένες

εκδόσεις του Android, κάθε μία προσθέτοντας νέα χαρακτηριστικά και λειτουργίες όπως θα δούμε αναλυτικότερα στο Κεφάλαιο 2.

1.2.2 Ιστορική αναδρομή 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 Cueartielles ονόμασαν το σχέδιο από τον Arduino της Ivrea και ξεκίνησαν να παράγουν πλακέτες.

Το πρώτο Arduino που φτιάχτηκε ονομάστηκε “Serial Arduino” και περιελάμβανε μια ATmega8 με σύνδεση RS-232 με τον μικροελεγκτή. Μετά ακολούθησαν εκδόσεις που περιελάμβαναν USB μετατροπέα. Ακολούθως κυκλοφόρησε το Arduino Extreme στο οποίο αυξήθηκαν τα επιφανειακά εξαρτήματα. Στην συνέχεια με το Arduino Nuova Generazione μετατρέπεται η ATmega8 σε ATmega168 και περιλαμβάνει έναν απλούστερο USB μετατροπέα. Οι βελτιώσεις συνεχίστηκαν σε επόμενες εκδόσεις όπως είναι το Arduino UNO που αναβαθμίζεται ο μικροελεγκτής σε ATmega328. Η τελική βασική έκδοση του Arduino είναι το Arduino Leonardo αναβαθμίζοντας το σε ATmega32U4 που περιέχει USB και ο σχεδιασμός σε μεγάλο βαθμό γίνεται πιο απλός. Τέλος, φυσικά υπάρχουν και πιο εξειδικευμένες εκδόσεις για το μέγεθος και τις λειτουργίες μιας κατασκευής όπως είναι το Arduino Nano και το Arduino Mini.

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.

1.3 Ανασκόπηση της πτυχιακής εργασίας

Για τη δημιουργία της πτυχιακής εργασίας αρχικά κάναμε έρευνα για να δούμε τι υλικά θα χρειαστούμε και στη συνέχεια πραγματοποιήσαμε έρευνα αγοράς για να δούμε πια είναι τα πιο οικονομικά υλικά και να τα προμηθευτούμε. Αφού παραλάβουμε τα υλικά αρχίσαμε να σχεδιάζουμε σιγά σιγά το πως θα τοποθετηθούν όλα μεταξύ τους και το πως όλα αυτά θα δημιουργούν το αυτόνομο πλέον 2-τροχο ρομπότ. Καταλήξαμε στη δημιουργία ενός ξύλινου σκελετού όπου βιδώθηκαν όλα τα εξαρτήματα που αγοράσαμε. Στη συνέχεια άρχισε ο προγραμματισμός πρώτα στην εφαρμογή Android και ταυτόχρονα δουλεύαμε και λίγο με το κώδικα του Arduino. Στην εφαρμογή Android πρώτα δημιουργήσαμε την διεπαφή με το Bluetooth, να δούμε ότι επικοινωνούν μεταξύ τους (Android-Arduino), στη συνέχεια φτιάξαμε ένα layout όπου άνοιξε η κάμερα του κινητού και πάνω σε αυτό κάναμε την κατάλληλη επεξεργασία ώστε να έχουμε επιλεγμένο ένα μαύρο αντικείμενο και το τελικό μέρος της εφαρμογής στο Android ήταν να στέλνει μέσω της επικοινωνίας με τον Arduino ανάλογα με το που βρίσκετε το μαύρο αντικείμενο τα ανάλογα δεδομένα. Έπειτα αρχίσαμε να ασχολούμαστε με την εφαρμογή στο Arduino. Γράψαμε κομμάτι-κομμάτι για κάθε εξάρτημα ξεχωριστά και στη συνέχεια αρχίσαμε να τα ενώνουμε μεταξύ τους. Πρώτα γράψαμε για το Bluetooth (να διαβάζει τα δεδομένα που του στέλνει η εφαρμογή από το κινητό), μετά γράψαμε το κώδικα, όπου ήταν και το πιο δύσκολο κομμάτι, για τον αισθητήρα «Accelerometer». Στο κομμάτι του κώδικα για τον αισθητήρα «Accelerometer» πρώτα προσπαθήσαμε να διαβάσουμε τα δεδομένα που μας έδινε ο αισθητήρας και αφού το καταφέραμε εφαρμόσαμε μέσα στο κώδικα ένα τύπο ο οποίος έπαιρνε όλα τα δεδομένα από τον αισθητήρα και μας έβγαζε μια τιμή που άλλαζε ανάλογα με τη κίνηση του αισθητήρα. Στη συνέχεια ακολούθησε κώδικας ο οποίος έπαιρνε τη τιμή που έβγαζε ο τύπος και με τον «PID» αλγόριθμο υπολογίζονταν η ταχύτητα που έπρεπε να καταλήξει στους 2 τροχούς. Όμως για τη καλύτερη και γρηγορότερη ρύθμιση του «PID» δημιουργήσαμε επίσης επιπρόσθετο κώδικα στην εφαρμογή του Android με τον οποίο μας βοήθησε στην άμεση ρύθμιση του «PID». Τέλος προσθέσαμε και το τελευταίο κομμάτι τον αισθητήρα για τα εμπόδια ο οποίος μέσω ηχητικών κυμάτων εντόπιζε οτιδήποτε βρίσκονταν μπροστά του και μέτραγε την απόσταση

Κίνηση Δίτροχου Οχήματος με την βοήθεια Arduino και Android

από το 2-τροχο και έτσι ότι βρισκόταν σε απόσταση λιγότερη από τα 4cm σταματούσε το 2-τροχο.

ΚΕΦΑΛΑΙΟ 2

ΕΦΑΡΜΟΓΗ ANDROID

2.1 Εισαγωγή

2.1.1 Εφαρμογές λογισμικού

Μια εφαρμογή λογισμικού είναι σχεδιασμένη να λειτουργεί σε κινητά τηλέφωνα τύπου smartphones, υπολογιστές τύπου tablet και άλλες φορητές συσκευές. Ένας χρήστης μιας φορητής συσκευής μπορεί να δημιουργήσει μια εφαρμογή, όπως θα σας παρουσιάσουμε την εφαρμογή μας σε επόμενη ενότητα. Μπορεί επίσης να βρει προεγκατεστημένες εφαρμογές με την αγορά της φορητής του συσκευής, όπως για e-mail, ημερολόγιο, κατάλογο επαφών, χρηματιστηριακές αγορές, πληροφορίες για τον καιρό, παιχνίδια, αυτοματισμούς εργοστασίων, GPS και location-based υπηρεσίες, banking, εξέλιξη παραγγελιών, καθώς και στις αγορές εισιτηρίων. Τέλος, ένας χρήστης μπορεί να εντοπίζει έτοιμες εφαρμογές για κάθε εργασία που είναι συνήθως διαθέσιμες μέσω πλατφορμών διανομής που ονομάζεται καταστήματα app. Τα καταστήματα αυτά είναι ηλεκτρονικά και συνήθως λειτουργούν από τον κάτοχο του κινητού λειτουργικού συστήματος, όπως το Apple App Store, Google Play, Windows Phone Store και BlackBerry App World. Ορισμένες εφαρμογές είναι δωρεάν, ενώ άλλες πρέπει να αγοραστούν.



Εικόνα 2.1 Mobile Development

2.1.2 Εφαρμογές Android

Το Android είναι λειτουργικό σύστημα για συσκευές κινητής τηλεφωνίας το οποίο τρέχει τον πυρήνα του λειτουργικού Linux. Αρχικά αναπτύχθηκε από την Google και αργότερα από την Open Handset Alliance . Η Google δημοσίευσε το μεγαλύτερο μέρος του κώδικα του Android υπό τους όρους της Apache License, μιας ελεύθερης άδειας λογισμικού. Επιτρέπει στους κατασκευαστές λογισμικού να συνθέτουν κώδικα με την χρήση της γλώσσας προγραμματισμού Java, ελέγχοντας την συσκευή μέσω βιβλιοθηκών λογισμικού ανεπτυγμένων από την Google. Το Android Software Development Kit αναλαμβάνει την μεταγλώττιση του κώδικα και των resource files και δημιουργεί το αρχείο εγκατάστασης με κατάληξη .apk . Με αυτό το αρχείο δύναται να εγκαθιστούμε εφαρμογές Android σε συσκευές με λειτουργικό σύστημα Android. Κάθε εφαρμογή που εγκαθίσταται έχει το δικό της στιγμιότυπο, πρόσβαση μόνο σε πόρους του συστήματος που χρειάζεται ώστε να διαθέτει ένα ασφαλές περιβάλλον με σωστή συμπεριφορά.

Το Android είναι κατά κύριο λόγο σχεδιασμένο για συσκευές με οθόνη αφής, όπως τα έξυπνα τηλέφωνα (smartphones) και τα tablet, με διαφορετικό περιβάλλον χρήσης για τηλεοράσεις , αυτοκίνητα και ρολόγια χειρός . Παρόλο που έχει αναπτυχθεί για συσκευές με οθόνη αφής, έχει χρησιμοποιηθεί σε κονσόλες παιχνιδιών, ψηφιακές φωτογραφικές μηχανές, Η/Υ και σε άλλες ηλεκτρονικές συσκευές.



Εικόνα 2.2 Το λογότυπο του Android

2.1.3 Ανάπτυξη μιας εφαρμογής

Η ανάπτυξη εφαρμογών για φορητές συσκευές απαιτεί την εξέταση των δυνατοτήτων και τυχόν περιορισμών αυτών των συσκευών. Οι φορητές συσκευές λειτουργούν με μπαταρία και έχουν λιγότερο ισχυρούς επεξεργαστές από τους ηλεκτρονικούς υπολογιστές αλλά διαθέτουν δυνατότητες, όπως τον εντοπισμό θέσης και την χρήση κάμερες. Οι προγραμματιστές πρέπει επίσης να εξετάσουν ένα ευρύ φάσμα μεγεθών οθόνης, προδιαγραφών κάθε συσκευής και διαμορφώσης της λόγω του έντονου ανταγωνισμού στο λογισμικό φορητών συσκευών και αλλαγών σε κάθε πλατφόρμα, αν και αυτά τα ζητήματα μπορούν να ξεπεραστούν με την ανίχνευση φορητών συσκευών και προσαρμογής κάθε εφαρμογής στις δυνατότητες της κάθε συσκευής. Η ανάπτυξη εφαρμογών για φορητές συσκευές απαιτεί τη χρήση εξειδικευμένων ολοκληρωμένων αναπτυξιακών περιβαλλόντων. Οι εφαρμογές για κινητά δοκιμάζονται πρώτα μέσα στο περιβάλλον ανάπτυξης χρησιμοποιώντας εξομοιωτές και αργότερα υποβάλλονται σε δοκιμές. Η κάθε συσκευή διαφέρει σε δυνατότητες και είναι πιθανόν να μην διαθέτει ο κάθε προγραμματιστής μεγάλη ποικιλία συσκευών ή να μην επιθυμεί να δημιουργήσει εφαρμογή για τις συσκευές που διαθέτει. Έτσι οι εξομοιωτές παρέχουν έναν φθηνό τρόπο για να δοκιμάσουν εφαρμογές σε φορητές συσκευές στα οποία οι προγραμματιστές ενδέχεται να μην έχουν φυσική πρόσβαση. Ο σχεδιασμός διεπαφής χρήστη για κινητά (UI) είναι επίσης απαραίτητος. Το UI Mobile εξετάζει τους περιορισμούς, το περιβάλλον, την οθόνη, την είσοδο για το σχεδιασμό. Η είσοδος χρήστη επιτρέπει στους χρήστες να χειρίζονται ένα σύστημα και η έξοδος της συσκευής επιτρέπει στο σύστημα να υποδείξει τα αποτελέσματα των χειρισμών των χρηστών. Οι περιορισμοί σχεδιασμού του φορητού περιβάλλοντος χρήστη περιλαμβάνουν περιορισμένους παράγοντες προσοχής και μορφής, όπως το μέγεθος της οθόνης μιας κινητής συσκευής για τον χρήστη. Τα περιβάλλοντα UI του κινητού τηλεφώνου σηματοδοτούν τα μηνύματα από τη δραστηριότητα των χρηστών, όπως η τοποθεσία και ο προγραμματισμός που μπορούν να εμφανιστούν από τις αλληλεπιδράσεις των χρηστών μέσα σε μια εφαρμογή για κινητά. Συνολικά, ο στόχος του σχεδιασμού του φορητού UI είναι κυρίως για ένα κατανοητό, φιλικό προς το χρήστη περιβάλλον.

2.2 Βασικά Χαρακτηριστικά

Το Android είναι μια ολοκληρωμένη πλατφόρμα για φορητές συσκευές που περιλαμβάνει το λειτουργικό σύστημα, τις βιβλιοθήκες και τις βασικές εφαρμογές. Οι συσκευές που χρησιμοποιούν Android έχουν την δυνατότητα να αναπαράγουν πολλαπλά μέσα (multimedia) και να εκτελούν ταυτόχρονα πολλαπλές εφαρμογές (multitasking) χωρίς καμία απώλεια πληροφορίας ή εργασίας του χρήστη.

Το Android είναι μια πλατφόρμα ανοιχτού κώδικα και εκμεταλλεύεται πλήρως τις δυνατότητες μιας συσκευής. Αυτό σημαίνει ότι μπορεί εύκολα να επεχταθεί και να τροποποιηθεί ώστε να συμβαδίζει με τις εξελίξεις και τις τελευταίες τεχνολογίες. Το γεγονός ότι πρόκειται για πλατφόρμα ανοιχτού κώδικα εξασφαλίζει ότι θα έχει συνεχή πρόοδο στην δημιουργία ελεύθερων για χρήση προηγμένων εργαλείων λογισμικού.

Η πλατφόρμα Android παρέχει σε κάθε χρήστη που ασχολείται με την ανάπτυξη εφαρμογών την δυνατότητα χρησιμοποίησης μεγάλης ποικιλίας βιβλιοθηκών και εργαλείων για την δημιουργία πιο εξελιγμένου λογισμικού. Με την ανάπτυξη έτοιμων εργαλείων αυξάνεται σημαντικά η εξέλιξη των Android εφαρμογών με μεγαλύτερη ταχύτητα και λιγότερα λάθη για τους προγραμματιστές. Για παράδειγμα, ένας προγραμματιστής Android μπορεί να δημιουργήσει πολύπλοκες εφαρμογές με μεγάλη λειτουργικότητα, όπως στην μετάδοση δεδομένων από το κινητό μέσω διαδικτύου και την άμεση λήψη και εμφάνιση στην οθόνη της συσκευής όσων πληροφοριών αναζητούσε ο χρήστης.

Όλες οι εφαρμογές Android έχουν πρόσβαση στις κύριες λειτουργίες της συσκευής κάτι που βοηθάει στους χρήστες να προσαρμόζουν την συσκευή τους ανάλογα τις ανάγκες τους και να απολαμβάνουν ένα ευρύ φάσμα εφαρμογών Android για αμέτρητους σκοπούς. Οι εφαρμογές και οι βασικές λειτουργίες μπορούν να τροποποιηθούν ακόμα και να αντικατασταθούν πλήρως από άλλες.

Βασικά χαρακτηριστικά του Android είναι η προσαρμογή σε πολλές αναλύσεις οθόνης, δισδιάστατες ψηφιακές γραφικές βιβλιοθήκες και τρισδιάστατα γραφικά. Η αποθήκευση δεδομένων γίνεται με χρήση βάσης δεδομένων SQLite. Επίσης, υποστηρίζει τεχνολογίες συνδεσιμότητας όπως είναι GSM, 4G, NFV και Wi-Fi. Για την περιήγηση στον ιστό διαθέτει φυλλομετρητή βασισμένο στην ανοιχτή τεχνολογία WebKit αλλά μέσω του ηλεκτρονικού καταστήματος Google Play υπάρχουν και άλλοι διαθέσιμοι. Το λογισμικό είναι γραμμένο με την γλώσσα

προγραμματισμού Java και δύναται να μεταγλωττιστεί και να εκτελεστεί στην εικονική μηχανή Dalvik. Επιπρόσθετα, το λειτουργικό Android υποστηρίζει διάφορες μορφές ήχου, στατικής και κινούμενης εικόνας, οθόνες αφής, GPS, αισθητήρες κ.α. Επίσης διατίθεται ένα περιβάλλον ανάπτυξης λογισμικού που περιλαμβάνει ένα προσομοιωτή συσκευής, εργαλεία για διόρθωση σφαλμάτων και ανάλυσης της απόδοσης του εκτελέσιμου λογισμικού. Τέλος διατίθεται για όλους τους χρήστες ένας κατάλογος εφαρμογών που μπορούν να εγκατασταθούν άνεσα στην συσκευή μέσω του ηλεκτρονικού καταστήματος του Google Play.

2.3 Οι εκδόσεις του Android

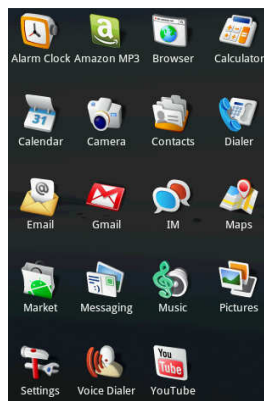
Το Android όπως αναφέραμε έχει δημιουργηθεί πάνω στο λειτουργικό σύστημα Linux προσαρμοσμένο σε κινητές συσκευές. Η εξέλιξή του λόγω ότι είναι σύστημα ανοιχτού κώδικα είναι ραγδαία και μέχρι σήμερα έχουν βγει αρκετές εκδόσεις. Στον παρακάτω πίνακα φαίνονται όλες οι εκδόσεις και πότε κυκλοφόρησαν.

Νούμερο Έκδοσης	Κωδικό Όνομα	Ημερομηνία κυκλοφορίας
1.0	-	23 Σεπτεμβρίου 2008
1.1	-	9 Φεβρουαρίου 2009
1.5	Cupcake	27 Απριλίου 2009
1.6	Donut	15 Σεπτεμβρίου 2009
2.0-2.1	Éclair	26 Οκτωβρίου 2009
2.2-2.2.3	Froyo	20 Μαΐου 2010
2.3-2.3.7	Gingerbread	6 Δεκεμβρίου 2010
3.0-3.2.6	Honeycomb	22 Φεβρουαρίου 2011
4.0-4.0.4	Ice Cream Sandwich	18 Οκτωβρίου 2011
4.1-4.3.1	Jelly Bean	9 Ιουλίου 2012
4.4-4.4.4	KitKat	31 Οκτωβρίου 2013
5.0-5.1.1	Lollipop	12 Νοεμβρίου 2014
6.0-6.0.1	Marshmallow	5 Οκτωβρίου 2015
7.0-7.1.1	Nougat	22 Αυγούστου 2016

Πίνακας 2.1: Εκδόσεις του Android

2.3.1 Android 1.0 – 1.1

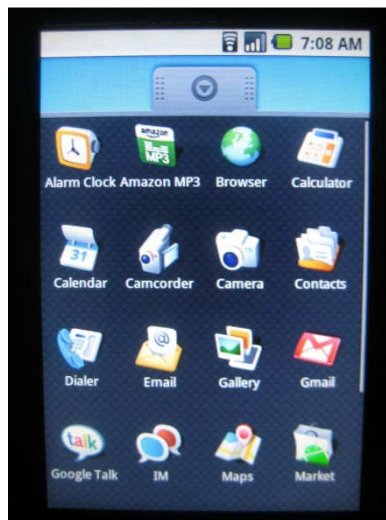
Οι πρώτες εκδόσεις κυκλοφόρησαν με την δημιουργία του πρώτου Android smartphone. Υπήρχαν διάφορες ενσωματωμένες εφαρμογές όπως ο περιηγητής, η κάμερα, οι χάρτες, η μουσική, το email, το ξυπνητήρι και άλλες.



Εικόνα 2.3 Περιβάλλον Android 1.0

2.3.2 Android 1.5

Το Android 1.5 Cupcake ήταν η τρίτη έκδοση του Android που αναπτύχθηκε από την Google. Είναι μια μεγάλη αναβαθμισμένη πλατφόρμα για τις συσκευές με λειτουργικό Android που κυκλοφόρησε το Απρίλη του 2009. Η έκδοση περιλαμβάνει νέα χαρακτηριστικά για τους χρήστες και τους προγραμματιστές. Υποστηρίζει νέες λειτουργίες για την κάμερα τις συσκευής, όπως διάθεση κάμερας 5MP με αυτόματη εστίαση, η καταγραφή και παρακολούθηση βίντεο από την λειτουργία της κάμερας και η άμεση μεταφόρτωση του βίντεο αλλά και των φωτογραφιών στο Youtube και το Picasa αντίστοιχα απευθείας από το τηλέφωνο. Έχει νέο εικονικό πληκτρολόγιο με πρόβλεψη λέξεων. Υποστηρίζει πρότυπο Bluetooth επίσης έχει και την ικανότητα να συνδέεται αυτόματα σε μικροσυσκευές Bluetooth από μια συγκεκριμένη απόσταση. Ακόμα στην έκδοση αυτή έχει νέο γραφικό περιβάλλον με κινούμενες μεταβάσεις οθόνης και την υποστήριξη διάφορων widgets.



Εικόνα 2.4 Περιβάλλον Android 1.5

2.3.3 Android 1.6

Το Android 1.6 Donut είναι μια έκδοση του λειτουργικού συστήματος ανοικτού κώδικα για κινητά Android που αναπτύχθηκε από την Google. Είναι μια αναβαθμισμένη πλατφόρμα για τις συσκευές με λειτουργικό Android που κυκλοφόρησε τον Σεπτέμβριο του 2009. Οι νέες λειτουργίες που υποστηρίζει είναι η ταχύτερη απόκριση σε σχέση με την προηγούμενη έκδοση. Υποστηρίζεται πλέον η επιλογή πολλαπλών αρχείων ταυτόχρονα, έχει ανανεωμένο γκάλερι και φωτογραφική μηχανή, καθώς και βελτιωμένο Android Market (μετατράπηκε αργότερα σε Google Play). Έχει ανανεωμένη φωνητική αναζήτηση όπως επίσης την δυνατότητα αναζήτησης σελιδοδεικτών, ιστορικού, επαφών αλλά και στο διαδίκτυο από την αρχική οθόνη. Υποστήριξη για ανάλυση οθονών WVGA. Ανανεωμένη υποστήριξη τεχνολογιών για CDMA/EVDO, 802.1x, VPNs και με μηχανή μετατροπής κειμένου σε ομιλία.

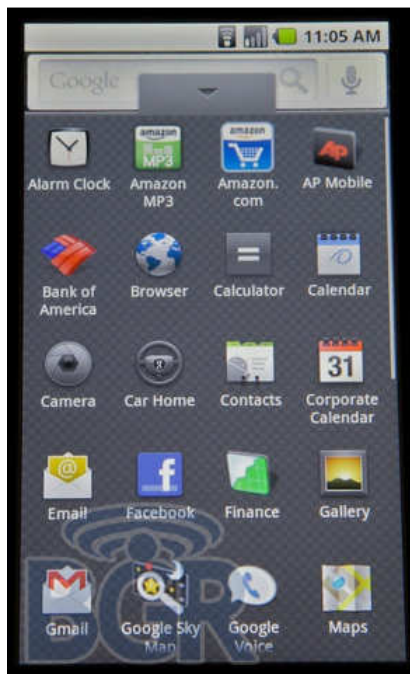
Τέλος, δείκτες χρήσης της μπαταρίας και αυτόματη περιστροφή οθόνης.



Εικόνα 2.5 Περιβάλλον Android 1.6

2.3.4 Android 2.0-2.1

Το Android 2.0-2.1 Eclair είναι μια έκδοση λειτουργικού συστήματος του Android που αναπτύχθηκε από την Google. Παρουσιάστηκε τον Οκτώβρη του 2009, το Android 2.1 στηρίζεται στις σημαντικές αλλαγές που έχουν γίνει στο Android 1.6 Donut. Η συγκεκριμένη έκδοση έχει ταχύτερη απόκριση του υλικού σε σχέση με τις δυο προηγούμενες και πλέον υποστηρίζονται περισσότερες οθόνες και αναλύσεις. Υπάρχει νέος browser ο οποίος υποστηρίζει το πρότυπο HTML5 με ανανεωμένο UI πρόγραμμα περιήγησης με μικρογραφίες σελιδοδείκτη, νέο User Interface, και βελτιωμένοι χάρτες Google Maps 3.1.2. Έχει ενσωματωθεί η υποστήριξη φλας για την κάμερα η οποία έχει πλέον και ψηφιακό zoom. Επίσης έχει βελτιωθεί η κλάση MotionEvent ώστε να υπάρχει η δυνατότητα για γεγονότα πολλαπλής αφής (multitouch events) και live wallpapers. Υποστηρίζεται Bluetooth 2.1 και έχει βελτιωθεί και το πληκτρολόγιο. Προσθέτει επίσης υποστήριξη για επικοινωνία κοντινού πεδίου (NFC), την ικανότητα να αναζητήσετε όλα τα αποθηκευμένα μηνύματα SMS και MMS και η υποστήριξη Exchange για την εφαρμογή Email.

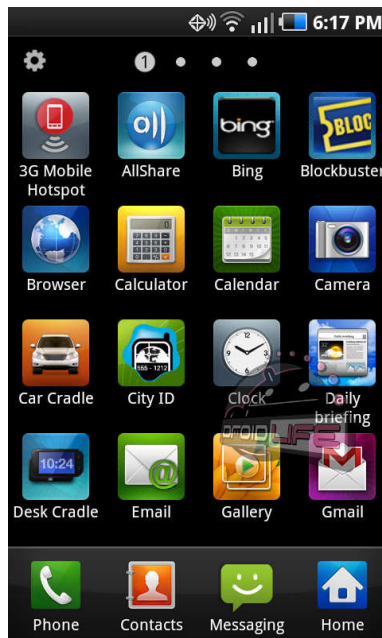


Εικόνα 2.6 Περιβάλλον Android 2.0

2.3.5 Android 2.2-2.2.3

Το Android 2.2-2.2.3 Froyo είναι μια έκδοση λειτουργικού συστήματος του Android που αναπτύχθηκε από την Google, που εκτείνονται σε εκδόσεις μεταξύ 2.2 και 2.2.3 και τα αποκαλυπτήρια του έγιναν τον Μάη του 2010.

Μία από τις πιο εξέχουσες αλλαγές στην κυκλοφορία αυτής της έκδοσης είναι οι βελτιστοποιήσεις στην ταχύτητα γενικά του λειτουργικού συστήματος, στην μνήμη και στην απόδοση. Έχει ενσωματωθεί ο μηχανισμός JavaScript του Chrome V8 στον browser, υπάρχει πλέον Adobe Flash 10.1, ενώ υποστηρίζεται καλύτερα πλέον το Microsoft Exchange. Έχει γίνει ανανέωση του Android Market. Ο χρήστης μπορεί πλέον να ελέγχει αν θα γίνεται ή όχι κίνηση πακέτων δεδομένων από το δίκτυο κινητής τηλεφωνίας. Υπάρχει η δυνατότητα εγκατάστασης εφαρμογών στην κάρτα μνήμης και η μεταφορά τους εκεί από τη μνήμη του τηλεφώνου. Επίσης στο κινητό βρίσκουμε και τη δυνατότητα χρήσης της συσκευής για διαμοιρασμό ίντερνετ μέσω Wi-Fi σε άλλες συσκευές (tethering). Τέλος μερικές ακόμα από τις βελτιώσεις ήταν σύνδεση μέσω USB και η γρήγορη εναλλαγή γλώσσας.



Εικόνα 2.7 Περιβάλλον Android 2.2

2.3.6 Android 2.3-2.3.7

Το Android 2.3-2.3.7 Gingerbread είναι μια έκδοση λειτουργικού συστήματος του Android που αναπτύχθηκε από την Google, και κυκλοφόρησε τον Δεκέμβριο του 2010. Υποστηρίζει πλέον πολύ μεγάλα μεγέθη οθονών και αναλύσεων, υπάρχουν αλλαγές στο User Interface το οποίο έχει γίνει πιο απλό και ταχύ, διαθέτει επανασχεδιασμένο multi-touch πληκτρολόγιο, προεγκατεστημένη υποστήριξη για τηλεφωνικές κλήσεις μέσω ίντερνετ (VoIP), download manager για κατέβασμα μεγάλων αρχείων, λειτουργίες copy-paste σε όλο το λειτουργικό, καθώς και προεγκατεστημένη υποστήριξη για πολλαπλές κάμερες. Υποστηρίζεται το NFC και έχει βελτιωθεί ο ήχος καθώς και οι λειτουργίες απεικόνισης για την ανάπτυξη παιχνιδιών. Τέλος το Gingerbread χρησιμοποιεί την έκδοση 2.6.35 του πυρήνα του Linux.



Εικόνα 2.8 Περιβάλλον Android 2.3

2.3.7 Android 3.0-3.2.6

Το Android 3.0-3.2.6 Honeycomb είναι μια έκδοση της πλατφόρμας Android που έχει σχεδιαστεί για συσκευές με μεγαλύτερα μεγέθη οθόνης, ειδικότερα τα tablets. Το Honeycomb έκανε το ντεμπούτο του τον Φεβρουάριο του 2011. Υπάρχει ένα νέο, εντελώς διαφορετικό, User Interface και υποστηρίζονται διπύρηνι και τετραπύρηνι επεξεργαστές. Οι βελτιώσεις της περιλαμβάνουν γρήγορη πρόσβαση σε χαρακτηριστικά της κάμερας, καλύτερο πληκτρολόγιο κατάλληλο για μεγάλες οθόνες, εκτέλεση πολλαπλών λειτουργιών και έχει απλοποιηθεί το multitasking έτσι ώστε ο χρήστης να μπορεί με τη χρήση ενός πλήκτρου να περνάει από μια εφαρμογή σε άλλη. Η έκδοση Honeycomb προσέθεσε την επιλογή να μεταφέρεται περιεχόμενο απευθείας από συσκευές USB, ενώ προσέθεσε διάφορες δυνατότητες και ευκολίες για χρήστες και developers όπως τη μεταφορά αρχείων από κάρτες SD και δυνατότητα Zoom to Fill. Τέλος, υπάρχει η δυνατότητα για Video Chat μέσω της εφαρμογής Google Talk, η ανάγνωση βιβλίων μέσω του Google eBooks Και μπορούν να κρυπτογραφηθούν όλα τα δεδομένα χρήστη.



Εικόνα 2.9 Περιβάλλον Android 3.0

2.3.8 Android 4.0-4.0.4

Το Android 4.0-4.0.4 Ice Cream Sandwich είναι μια έκδοση του λειτουργικό συστήματος για κινητά Android που αναπτύχθηκε από την Google. Παρουσιάστηκε τον Οκτώβρη του 2011, το Android 4.0 στηρίζεται στις σημαντικές αλλαγές που έγιναν από τα tablet με την κυκλοφορία του Android Honeycomb, σε μια προσπάθεια να δημιουργήσει μια εννοποιημένη πλατφόρμα για τα έξυπνα τηλέφωνα και τα tablets. Επίσης έχει βελτιωθεί η ταχύτητα και η απόδοση του συστήματος. Πλέον στο User Interface, το οποίο είναι και παλι διαφορετικό, υπάρχουν εικονικά πλήκτρα τα οποία παίρνουν τη θέση των φυσικών ή αφής που υπήρχαν στις συσκευές. Βελτίωση της ασφάλεια του συστήματος με την προσθήκη αναγνώρισης προσώπου για να ξεκλειδώσει η συσκευή. Ο browser μπορεί να ανοίξει ταυτόχρονα μέχρι και 16 καρτέλες. Υπάρχει η δυνατότητα ο χρήστης να τερματίσει εφαρμογές οι οποίες τρέχουν στο background, ενώ μπορεί να θέσει και όρια στην κίνηση πακέτων δεδομένων. Η εφαρμογή Android Beam αξιοποιεί πλέον το NFC αφού επιτρέπει την αποστολή δεδομένων από τη συσκευή σε όσες βρίσκονται εντός μιας μικρής ακτίνας εμβέλειας. Ακόμα με την ύπαρξη του Wi-Fi Direct συσκευές μπορούν να συνδεθούν μεταξύ τους ασύρματα χωρίς την μεσολάβηση κάποιου access point. Τέλος, μερικές άλλες δυνατότητες ήταν καλύτερη χρήση των φωνητικών εντολών, ανανεωμένο γραφικό περιβάλλον με αρκετά 3D στοιχεία, ειδική έκδοση του Gmail για tablets, δυνατότητα βιντεοκλήσεων μέσω εφαρμογής Google Talk, ανανεωμένη έκδοση Google Maps και βελτιστοποιημένη εφαρμογή για ανάγνωση Google e-books και υποστηρίζεται η εγγραφή βίντεο σε 1080p.



Εικόνα 2.10 Περιβάλλον Android 4.0

2.3.9 Android 4.1-4.3.1

Το Android 4.1-4.3.1 Jelly Bean είναι το όνομα που δίνεται σε τρεις μεγάλες κυρίες κυκλοφορίες του λειτουργικό συστήματος για κινητά Android που αναπτύχθηκε από την Google. Παρουσιάστηκε τον Ιούλη του 2012. Το περιβάλλον χρήσης και η απόκρισή του είναι πιο γρήγορη ενώ περιλαμβάνει πάρα πολλές μικρές βελτιώσεις σε όλο το σύστημα, όπως για παράδειγμα στην κάμερα, στην χρήση φωνής για υπαγόρευση κειμένου και στην απόδοση και το σχεδίασαν έτσι ώστε να δώσει το λειτουργικό σύστημα μια ομαλότερη και πιο δεκτική αίσθηση. Στην έκδοση 4.2 περιελάμβανε περαιτέρω βελτιστοποιήσεις, υποστήριξη πολλαπλών χρηστών για tablet, widgets στην οθόνη κλειδώματος, γρήγορες ρυθμίσεις και προφυλάξεις οθόνης.

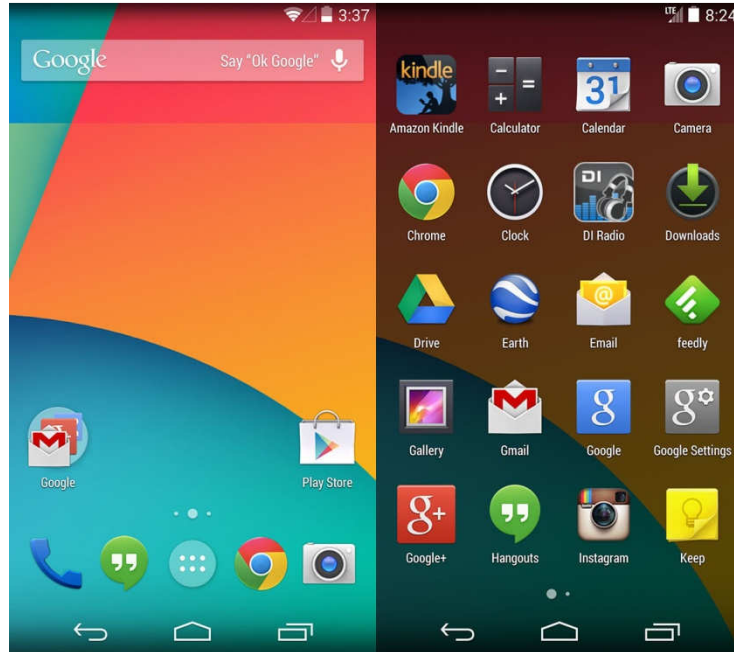


Εικόνα 2.11 Logo Android 4.1-4.3

2.3.10 Android 4.4-4.4.4

Το Android 4.4-4.4.4 KitKat είναι μια έκδοση του λειτουργικό συστήματος για κινητά Android που αναπτύχθηκε από την Google, Παρουσιάστηκε τον Οκτώβρη του 2013. Το KitKat επικεντρώθηκε κυρίως στη βελτιστοποίηση του λειτουργικού συστήματος και τη βελτίωση των επιδόσεων σε συσκευές entry-level που διαθέτουν περιορισμένους πόρους. Το KitKat πλέον μπορεί να λειτουργήσει σε συσκευές με RAM από 512MB και πάνω, φέρνοντας τις τελευταίες αναβαθμίσεις του Android ακόμη και σε οικονομικά προσιτά μοντέλα. Όπως

αναφέρει εξάλλου η Google, χρησιμοποιεί 16% λιγότερη μνήμη από το Jelly Bean, με αποτέλεσμα ένα συνολικά πιο ελαφρύ λειτουργικό σε ό,τι μοντέλο κι αν ενσωματώνεται.



Εικόνα 2.12 Περιβάλλον Android 4.4

2.3.11 Android 5.0-5.1.1

Το Android 5.0-5.1.1 Lollipop είναι μια έκδοση λειτουργικού συστήματος για κινητά Android που αναπτύχθηκε από την Google. Κυκλοφόρησε τον Νοέβρη του 2014. Μία από τις πιο βασικές αλλαγές στην κυκλοφορία του Lollipop είναι ένα επανασχεδιασμένο περιβάλλον εργασίας χρήστη χτισμένο γύρω από μια σχεδιαστική γλώσσα, η οποία έγινε για να διατηρήσει μια αίσθηση χαρτιού στο περιβάλλον. Άλλες αλλαγές περιλαμβάνουν βελτιώσεις στις κοινοποιήσεις, η οποίες είναι προσβάσιμες από την οθόνη κλειδώματος και εμφανίζονται μέσα σε εφαρμογές. Η Google έκανε επίσης εσωτερικές αλλαγές στην πλατφόρμα, με το Android Runtime (ART) να είναι επίσημο αντικαθιστώντας το Dalvik για βελτιωμένη απόδοση των εφαρμογών, καθώς και με τις αλλαγές που αποσκοπούν στη βελτίωση και τη βελτιστοποίηση της χρήσης της μπαταρίας.

Στον τομέα της ασφάλειας θα υπάρχει αυτόματη κρυπτογράφηση των δεδομένων σας και θα προστεθεί το έξυπνο κλείδωμα που επιτρέπει το αυτόματο ξεκλείδωμα μιας συσκευής. Επίσης, διαθέτει νέες γρήγορες ρυθμίσεις για εύκολη χρήση του φακού, hotspot, Wi-Fi, Bluetooth, ρύθμιση της φωτεινότητας, κ.ά Τέλος, θα μπορείτε να μοιράζεστε την συσκευή σας με άλλους χρήστες καθώς θα υποστηρίζει πολλούς χρήστες και χρήστη Guest.



Εικόνα 2.13 Logo Android 5.0-5.1.1

2.3.12 Android 6.0-6.0.1

Το Android 6.0-6.0.1 Marshmallow είναι η έκτη κύρια έκδοση του λειτουργικού συστήματος Android από την Google. Κυκλοφόρησε επίσημα τον Οκτώβριο του 2015. Το Marshmallow επικεντρώνεται κυρίως στη βελτίωση της συνολικής εμπειρίας του χρήστη σε σύγκριση με τον προκάτοχο του, Lollipop, εισήγαγε μια νέα αρχιτεκτονική στα δικαιώματα των εφαρμογών, ένα νέο σύστημα διαχείρισης ενέργειας που μειώνει την δραστηριότητα του παρασκηνίου όταν μια συσκευή δεν χρησιμοποιείται και USB τύπου-C, τη δυνατότητα να μεταφοράς των δεδομένων και τις εφαρμογές σε μια κάρτα microSD, και άλλες εσωτερικές αλλαγές. Επίσης αλλαγή έχουμε και στο Google Now, με το οποίο η Google επιχειρεί να αξιοποιήσει τις ολοένα αυξανόμενες πληροφορίες για τις δραστηριότητες του χρήστη, προς όφελός του με το Now on Tap. Τέλος, την δυνατότητα πληρωμών με το smartphone και την υποστήριξη αναγνωστών δακτυλικού αποτυπώματος.



Εικόνα 2.14 Περιβάλλον Android 6.0

2.3.13 Android 7.0-7.1.1

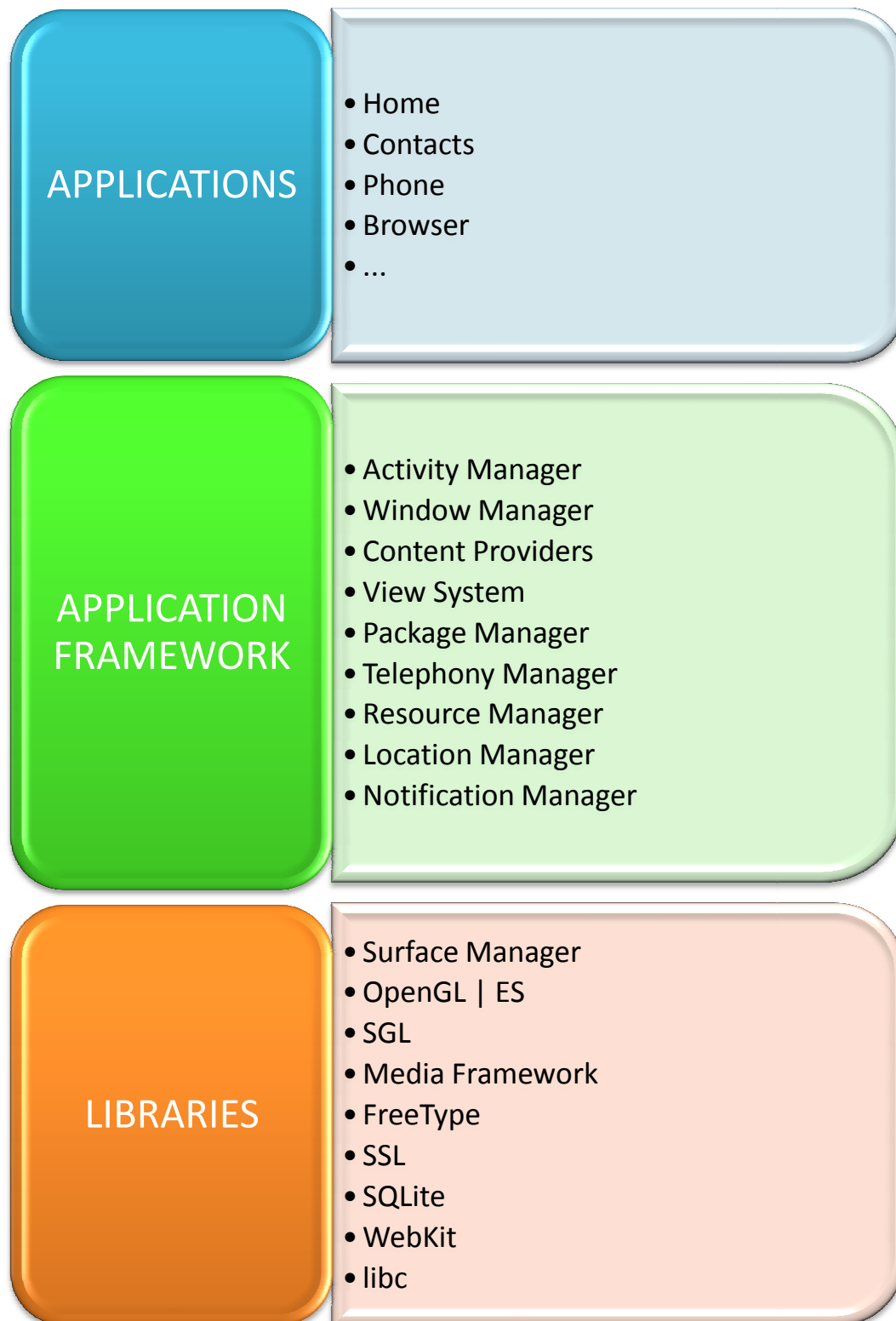
Το Android 7.0-7.1.1 Nougat είναι η έβδομη σημαντική έκδοση του λειτουργικού συστήματος Android από την Google. Κυκλοφόρησε για πρώτη φορά ως beta build τον Μαρτη του 2016 και επίσημα τον Αυγούστου του 2016. Το Nougat εισάγει σημαντικές αλλαγές στο λειτουργικό σύστημα και την πλατφόρμα ανάπτυξης του, συμπεριλαμβανομένης της δυνατότητας να εμφανίσει πολλαπλές εφαρμογές στην οθόνη με προβολή διαίρεσης της οθόνης, υποστήριξη για τις απαντήσεις ενσωματωμένη με τις κοινοποιήσεις, καθώς και περιβάλλον Java βασισμένο στο OpenJDK

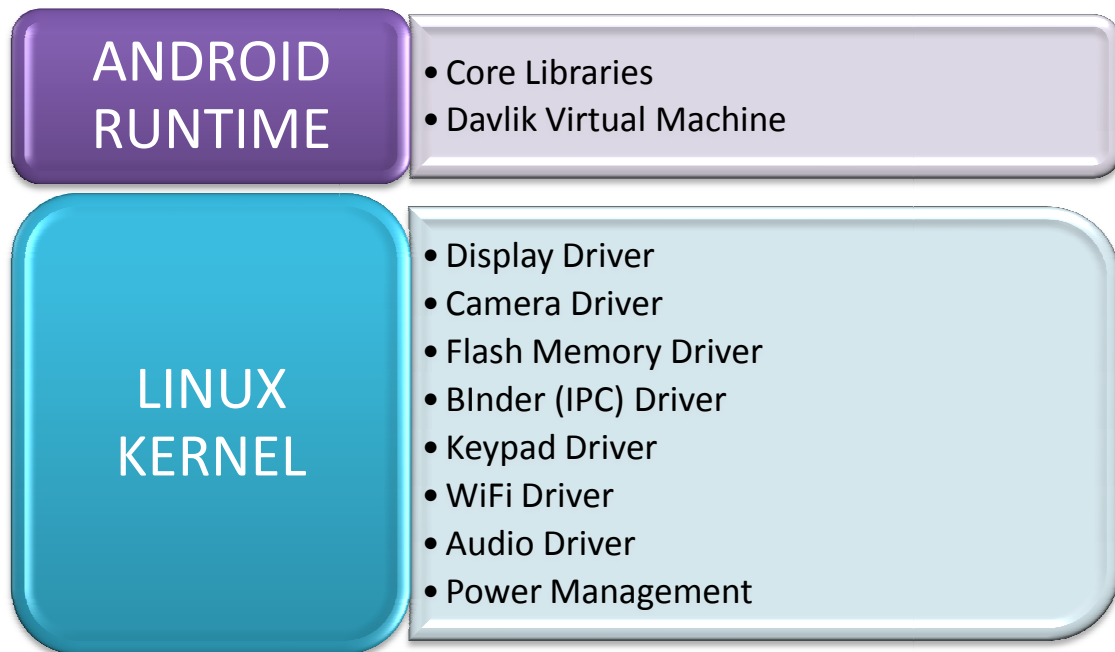


Εικόνα 2.15 Logo Android 7.0-7.1.1

2.4 Αρχιτεκτονική του Android

Παρακάτω απεικονίζεται το διάγραμμα αρχιτεκτονικής του Android:





Πίνακας 2.2: Αρχιτεκτονική του Android

Όπως βλέπουμε το Android χωρίζεται σε 5 βασικά επίπεδα:

- ✓ Τον πυρήνα Linux (Linux Kernel)
- ✓ Τις βιβλιοθήκες (Libraries)
- ✓ Τον χρόνο εκτέλεσης (Android Runtime)
- ✓ Το πλαίσιο εφαρμογής (Application Framework)
- ✓ Τις εφαρμογές (Applications)

2.4.1 Πυρήνας Linux (Linux Kernel)

Ο Linux Kernel είναι ο πυρήνας στον οποίο βασίζεται το Android και βρίσκεται στο χαμηλότερο επίπεδο. Το Android βασίζεται στον πυρήνα του Linux 2.6 για τις βασικές υπηρεσίες του συστήματος. Οι υπηρεσίες αυτές αφορούν διαχείριση μνήμης, διαχείριση διεργασιών, λειτουργίες δικτύου, ασφάλεια του λειτουργικού, και ένα σύνολο οδηγιών υλικού. Επίσης, ο πυρήνας του Android λειτουργεί ως ένα ενδιάμεσο επίπεδο αφαίρεσης μεταξύ της στοίβας λογισμικού και του υλικού και μπορεί να βασίζεται στον πυρήνα του Linux, αλλά διαφέρει

αρκετά από αυτόν. Ο λόγος είναι οι αλλαγές στην αρχιτεκτονική που έχει κάνει η Google για να είναι ελαφρύτερος και βελτιστοποιημένος για χρήση σε κινητές συσκευές. Τέλος, ο προγραμματιστής εφαρμογών δεν θα χρειαστεί να προγραμματίσει σε αυτό το επίπεδο.

2.4.2 Χρόνος Εκτέλεσης (Android Runtime)

Στο αμέσως επόμενο επίπεδο βρίσκεται το Android Runtime, το οποίο αποτελείται από τις Core libraries και την Dalvik Virtual Machine. Οι Core libraries μας παρέχουν όλες τις βασικές βιβλιοθήκες της Java που απαιτούνται για την ολοκληρωμένη λειτουργία. Επίσης η Dalvik Virtual Machine είναι μια εξειδικευμένη virtual machine μέσω της οποίας τρέχουν οι εφαρμογές του Android και εκτελεί αρχεία της μορφής .dex, ειδικά διαμορφωμένα για κινητές συσκευές που έχουν περιορισμένη μνήμη και ισχύ. Το Android είναι από τη φύση του multitasking λειτουργικό σύστημα και για αυτό επιτρέπει στις εφαρμογές του να τρέχουν σε πολλά νήματα ταυτόχρονα και το σύστημα είναι σε θέση να τρέχει πολλές εικονικές μηχανές ταυτόχρονα. Χάρη σε αυτό η κάθε εφαρμογή τρέχει μέσω τις δικής της εικονικής μηχανής στη δικιά της διεργασία και για αυτό το λόγο καμία εφαρμογή δεν έχει επαφή με την άλλη, ενώ εκτελούνται ταυτόχρονα.

2.4.3 Βιβλιοθήκες (Libraries)

Στο επόμενο επίπεδο βρίσκονται οι βιβλιοθήκες. Το Android περιλαμβάνει ένα σετ από βιβλιοθήκες, που χρησιμοποιούνται από διάφορα components του συστήματος και είναι γραμμένες σε C/C++. Οι βιβλιοθήκες από μόνες τους δεν αποτελούν εφαρμογές αλλά χρησιμοποιούνται από τις εφαρμογές για τις διάφορες λειτουργίες που παρέχει η καθεμία από αυτές. Ουσιαστικά αποτελούν ένα από τα δομικά υλικά των εφαρμογών. Οι δυνατότητες των βιβλιοθηκών του Android γίνονται εμφανείς στους προγραμματιστές στην στοίβα του πλαισίου εφαρμογής. Μερικές από τις βασικές βιβλιοθήκες είναι:

- ✓ System C Library: είναι μια ενσωμάτωση της standard βιβλιοθήκης συστήματος της C (libc), βελτιστοποιημένη για συσκευές που βασίζονται στο Linux
- ✓ Media Framework: Υποστηρίζει αναπαραγωγή και εγγραφή πολλών μέσων ήχου και εικόνας, όπως: MPEG4, H.264, MP3, PNG.
- ✓ SGL: είναι γνωστή μηχανή δισδιάστατων (2D) γραφικών.
- ✓ Surface Manager: διαχειρίζεται την πρόσβαση στο υποσύστημα προβολής και είναι υπεύθυνη για την δημιουργία γραφικών όπως layers 2D, 3D.
- ✓ SQLite: είναι μια πανίσχυρη και ταυτόχρονα πολύ ελαφριά σχεσιακή βάση δεδομένων που παρέχει υποστήριξη έτσι ώστε μια εφαρμογή να χρησιμοποιήσει την αποθήκευση δεδομένων και την ανάκτηση διάφορων στοιχείων όταν τα χρειαστούμε. Παρόμοια χρησιμοποιείται και στα iOS.
- ✓ FreeType: προσφέρει ευκρίνια στις γραμματοσειρές και τα γραφικά των εφαρμογών του συστήματος
- ✓ WebKit: μέσω αυτής της βιβλιοθήκης υποστηρίζεται ένας browser για την γρήγορη προβολή HTML περιεχομένου ο LibWebCore. Το ίδιο χρησιμοποιείται και από το Safari.
- ✓ OpenGL ES: βασίζει τις 3D βιβλιοθήκες. Οι βιβλιοθήκες χρησιμοποιούν είτε 3D υλικό είτε μια υψηλά βελτιωμένη τρισδιάστατη επιτάχυνση λογισμικού σε περίπτωση αν η συσκευή το διαθέτει.

2.4.4 Πλαίσιο Εφαρμογής (Application Framework)

Αυτό το επίπεδο παρέχει πλήρη προσβασιμότητα στους developers στα ίδια framework που χρησιμοποιούνται από τις εφαρμογές πυρήνα. Οι developers έχουν στην διάθεση τους τα εργαλεία που τους δίνονται όπως η δυνατότητα ελέγχου του υλικού της συσκευής και μέσω αυτής μπορούν να αποκτήσουν πρόσβαση σε υπηρεσίες εντοπισμού, εκτέλεση διεργασιών παρασκηνίου, να θέσουν χρονοδιακόπτες για την εμφάνιση ειδοποιήσεων κ.α. για να αναπτύξουν εφαρμογές. Παρακάτω αναφέρονται ένα σύνολο συστημάτων και υπηρεσιών για τις εφαρμογές:

- ✓ View System: αποτελεί ένα σύνολο από αντικείμενα GUI τα οποία μπορούν να χρησιμοποιηθούν κατά το σχεδιασμό μιας εφαρμογής όπως λίστες,

πίνακες, χώρος κειμένου, κουμπιά, πλέγματα, mapviews, ενσωματωμένος web browser κ.α

- ✓ Content Providers: επιτρέπουν στις εφαρμογές να έχουν πρόσβαση σε δεδομένα άλλων εφαρμογών ή να διαμοιράζονται με άλλες εφαρμογές τα δικά τους δεδομένα όπως είναι οι βάσεις δεδομένων.
- ✓ Resource Manager: παρέχει πρόσβαση σε υλικό το οποίο δεν είναι σε μορφή κώδικα, όπως είναι εικόνες, strings, αρχεία κ.α.
- ✓ Notification Manager: επιτρέπει σε όλες τις εφαρμογές πρόσβαση στις υπηρεσίες ειδοποιήσεων χρήστη όπως είναι να εμφανίζουν ειδοποιήσεις στο status bar, ήχος και δόνηση του κινητού κ.α.
- ✓ Activity Manager: διαχειρίζεται τον κύκλο ζωής των δραστηριοτήτων, την κατάσταση τους στην στοίβα και παρέχει δυνατότητα πλοήγησης από δραστηριότητα σε δραστηριότητα κρατώντας αποθηκευμένη στη μνήμη τη σειρά εκτέλεσης αυτών.

2.4.5 Εφαρμογές (Applications)

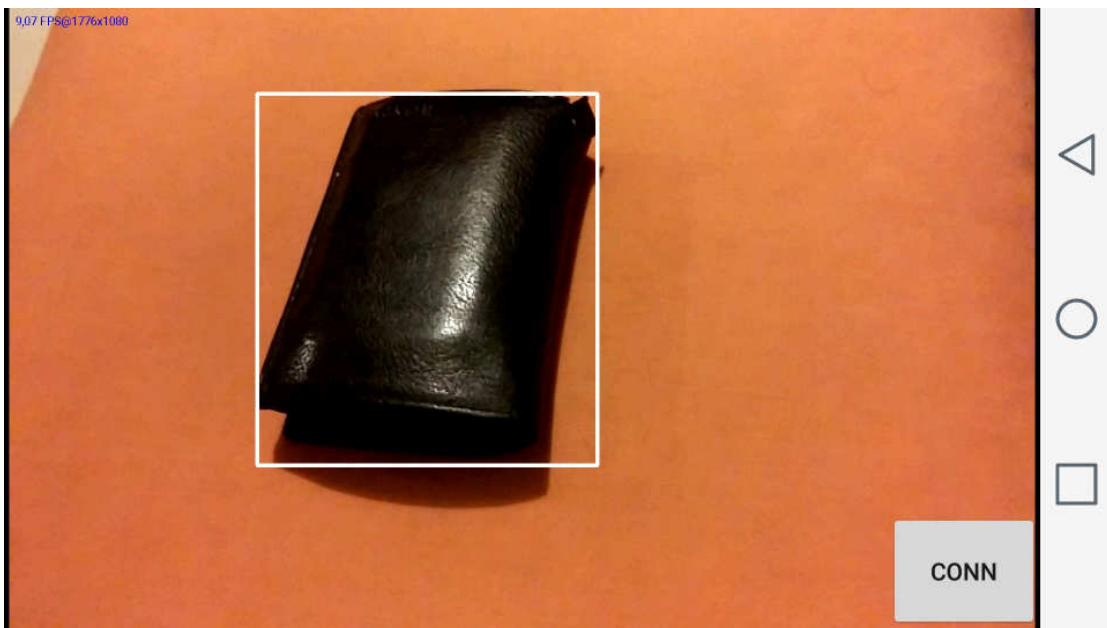
Το τελευταίο στρώμα στην αρχιτεκτονική του Android είναι τα applications και είναι αυτά που αντιλαμβάνεται ο χρήστης. Σε κάθε Android συσκευή του χρήστη υπάρχουν προεγκατεστημένες εφαρμογές όπως τηλέφωνο, επαφές, μουσική, ημερολόγια, mail, camera κ.α., όπως επίσης εφαρμογές που κάνει εγκατάσταση σύμφωνα με τις ανάγκες του. Οποιαδήποτε εφαρμογή που εγκαθιστά ο χρήστης εκτελείται σε αυτό το επίπεδο. Τέλος, όλες οι εφαρμογές είναι προγραμματισμένες σε γλώσσα προγραμματισμού Java. Μια Java εφαρμογή για Android δεν είναι συμβατή με προγράμματα Java γραμμένα για Java SE και Java ME πλατφόρμες.

2.5 Παρουσίαση της εφαρμογής μας

Για να κατασκευάσουμε την εφαρμογή Android «My Robot» με την χρήση του προγράμματος Android Studio αρχικά:

- ✓ Επιλέγουμε File → New → New Project
- ✓ Application Name = My_Robot
- ✓ Minimum Required SDK = API18
- ✓ Target SDK = API23
- ✓ Compiled With = API21 & API23

Η εφαρμογή μας αποτελείται από μία οθόνη διεπαφής χρήστη. Στην οθόνη ο χρήστης βλέπει την κάμερα και τον εντοπισμό του μεγαλύτερου μαύρου αντικειμένου με ένα λευκό ορθογώνιο περίγραμμα. Στο κάτω μέρος της οθόνης υπάρχει ένα πλήκτρο «Conn» για την σύνδεση της εφαρμογής «My Robot» με το Arduino.



Εικόνα 2.16 Κύρια οθόνη εφαρμογής

Τα περιγραφικά αρχεία που δημιουργούν την οθόνη διεπαφής είναι:

- ✓ Το αρχείο activity_main.xml που περιέχει το content_main.xml

```
<?xml version="1.0" encoding="utf-8"?>
<android.support.design.widget.CoordinatorLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:fitsSystemWindows="true"
    tools:context="com.example.lefte.opencv4test.MainActivity">

    <android.support.design.widget.AppBarLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:theme="@style/AppTheme.AppBarOverlay">

    </android.support.design.widget.AppBarLayout>

    <include layout="@layout/content_main" />

</android.support.design.widget.CoordinatorLayout>
```

- ✓ Το αρχείο content_main.xml δημιουργεί το πλήκτρο «Conn» και την επέκταση «JavaCameraView»

```
<?xml version="1.0" encoding="utf-8"?>
<FrameLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    xmlns:opencv="http://schemas.android.com/apk/res-auto"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content">

    <org.opencv.android.JavaCameraView
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:visibility="gone"
        android:id="@+id/MainActivityCameraView"
        opencv:show_fps="true"
        opencv:camera_id="any"
        android:layout_row="1"
        android:layout_column="0" />

    <Button
        android:layout_width="wrap_content"
        android:layout_height="70dp"
        android:text="Conn"
        android:id="@+id/conn"
        android:clickable="true"
        android:enabled="true"
        android:layout_gravity="right|bottom" />

</FrameLayout>
```

- ✓ Το αρχείο manifest.xml ορίζει τις βασικές πληροφορίες της εφαρμογής, όπως το όνομα, την διάταξη και τις άδειες χρήσεις για την σύνδεση του υλικού του κινητού τηλεφώνου Bluetooth και Camera με την εφαρμογή

```
<?xml version="1.0" encoding="utf-8" ?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.example.lefte.opencv4test"
    android:versionCode="1"
    android:versionName="1.0">
    <application
        android:allowBackup="true"
        android:icon="@mipmap/my_robot"
        android:label="My_robot"
        android:supportsRtl="true"
        android:theme="@style/AppTheme">
        <activity
            android:name=".MainActivity"
            android:label="My_robot"
            android:theme="@style/AppTheme.NoActionBar"
            android:screenOrientation="landscape">
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />
                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>
    </application>
    <uses-permission android:name="android.permission.BLUETOOTH" />
    <uses-permission android:name="android.permission.BLUETOOTH_ADMIN" />
    <uses-feature android:name="android.hardware.BLUETOOTH" android:required="false"/>
    <uses-permission android:name="android.permission.CAMERA" />
    <uses-feature android:name="android.hardware.camera" android:required="false"/>
    <uses-feature android:name="android.hardware.camera.autofocus" android:required="false"/>
    <uses-feature android:name="android.hardware.camera.front" android:required="false"/>
    <uses-feature android:name="android.hardware.camera.front.autofocus" android:required="false"/>
    <supports-screens android:resizeable="true"
        android:smallScreens="true"
        android:normalScreens="true"
        android:largeScreens="true"
        android:anyDensity="true" />
</manifest>
```


Στην συνέχεια ακολουθεί το αρχείο MainActivity.java που περιέχει τις λειτουργίες της εφαρμογής:

- ✓ Είσαγουμε τις ακόλουθες βιβλιοθήκες:

```
package com.example.lefte.opencv4test;

import android.bluetooth.BluetoothDevice;
import android.bluetooth.BluetoothManager;
import android.content.Context;
import android.content.Intent;
import android.os.Bundle;
import android.support.v7.app.AppCompatActivity;
import android.util.Log;
import android.view.SurfaceView;
import android.view.View;
import android.view.WindowManager;
import org.opencv.android.BaseLoaderCallback;
import org.opencv.android.CameraBridgeViewBase;
import org.opencv.android.JavaCameraView;
import org.opencv.android.LoaderCallbackInterface;
import org.opencv.android.OpenCVLoader;
import org.opencv.android.CameraBridgeViewBase.CvCameraViewListener2;
import org.opencv.core.Core;
import org.opencv.core.Mat;
import org.opencv.core.MatOfPoint;
import org.opencv.core.Rect;
import org.opencv.core.Scalar;
import org.opencv.core.Size;
import org.opencv.imgproc.Imgproc;
import android.bluetooth.BluetoothAdapter;
import android.bluetooth.BluetoothSocket;
import android.widget.Button;

import java.io.IOException;
import java.util.ArrayList;
import java.util.List;
import java.util.UUID;
```

✓ Ακολουθεί ο κώδικας με το OpenCv:

```

private BaseLoaderCallback mLoaderCallback = new BaseLoaderCallback(this) {
    @Override
    public void onManagerConnected(int status) {
        switch (status) {
            case LoaderCallbackInterface.SUCCESS: {
                Log.i(TAG, "Opencv loaded succ");
                mOpenCvCameraView.enableView();
                break;
            }
            default: {
                super.onManagerConnected(status);
            }
        }
    }
};

private JavaCameraView mOpenCvCameraView;
private static final String TAG = "MainActivity";
@Override
protected void onCreate(Bundle savedInstanceState) {

    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_main);
    getWindow().addFlags(WindowManager.LayoutParams.FLAG_KEEP_SCREEN_ON);
    getWindow().addFlags(WindowManager.LayoutParams.FLAG_FULLSCREEN);

    mOpenCvCameraView = (JavaCameraView) findViewById(R.id.MainActivityCameraView);
    mOpenCvCameraView.setVisibility(SurfaceView.VISIBLE);
    mOpenCvCameraView.setCvCameraViewListener(this);

    conn = (Button) findViewById(R.id.conn);
    conn.setOnClickListener((v) -> { bluetooth(); });
}
@Override
public void onPause() {
    super.onPause();
    if (mOpenCvCameraView!=null)
        mOpenCvCameraView.disableView();
}

```

Δημιουργούμε μία συνάρτηση callback για να ελένξουμε αν η βιβλιοθήκη OpenCv εάν λειτουργεί σωστά. Στην κλάση void onCreate δίνουμε τις παραμέτρους έτσι ώστε η εφαρμογή μας να είναι σε πλήρη οθόνη και να είναι πάντα η οθόνη ενεργοποιημένη όσο είναι εκτελέσιμη η εφαρμογή. Επίσης ενεργοποιούμε τις παραμέτρους που χρειάζεται το OpenCv για να εμφανίσει την επέκταση JavaCameraView. Στην κλάση public void onPause ελέγχουμε εάν το JavaCameraView λειτουργεί και το απενεργοποιούμε.

```

@Override
public Mat onCameraFrame(CameraBridgeViewBase.CvCameraViewFrame inputFrame) {
    mRgba = inputFrame.rgba();
    Imgproc.cvtColor(mRgba, imgHsv, Imgproc.COLOR_BGR2HSV);
    //range (0 to 180)+saturation (0 to 255)+ value 0 to 40
    Core.inRange(imgHsv, new Scalar(0, 0, 0, 0), new Scalar(180, 255, 40, 0), imgHsv2);
    Mat erode = Imgproc.getStructuringElement(Imgproc.MORPH_ERODE, new Size(1, 5));
    Imgproc.erode(imgHsv2, imgLine, erode);
    Mat dilate = Imgproc.getStructuringElement(Imgproc.MORPH_DILATE, new Size(1, 5));
    Imgproc.dilate(imgLine, imgHsv2, dilate);

    double maxVal = 0;
    int maxValIdx = -1;
    Imgproc.findContours(imgHsv2, mContours, new Mat(), 0, 2);
    int contoursCounter = mContours.size();
    for (int i = 0; i < contoursCounter; i++)
    {
        double contourArea = Imgproc.contourArea(mContours.get(i));
        if (maxVal < contourArea)
        {
            maxVal = contourArea;
            maxValIdx = i;
        }
    }
}

```

Στην παραπάνω κλάση σε κάθε εικόνα που παίρνουμε, την αποθηκεύουμε στην μεταβλητή mRgba. Μετά την μετατρέπουμε σε εικόνα hsv. Το hsv το χρησιμοποιούμε για να πάρουμε καλύτερα τις αποχρώσεις των χρωμάτων. Επειδή εμείς θέλουμε να εντοπίσουμε το μαύρο χρώμα στην εικόνα, ορίζουμε τα όρια του μαύρου να είναι:

- ❖ Εύρος(Range): 0-180
- ❖ Κορεσμός(Saturation): 0-255
- ❖ Τίμη(Value): 0-40

Ακολούθως προσθέτουμε ένα φίλτρο που ονομάζεται διάβρωση (erosion) μεγέθους {1,5} με σκοπό την απομάκρυνση των ανεπιθύμητων pixel. Επίσης με την χρήση του φίλτρου διαστολής (dilation) επαναφέρουμε το αντικείμενο μας στο αρχικό του μέγεθος {1,5}.

Στην συνέχεια εντοπίζουμε τα περιγράμματα και επιλέγουμε το μεγαλύτερο.

```
Rect rect = null;
if (maxValIdx > -1)
    rect = Imgproc.boundingRect(mContours.get(maxValIdx));
if (rect != null) {
    Core.rectangle(mRgba, rect.tl(), rect.br(), new Scalar(255, 255, 255), 5);
    int kuvos= rect.x+rect.width/2;
    int imgcenter=(mRgba.width()/2);
    if (kuvos < imgcenter-101){
        flag_ar=true;
        flag_dek=false;
        flag_stop=false;
        flag_go=false;
        //Log.d(TAG, "aristera");
    }
    if (kuvos > imgcenter+101) {
        flag_ar=false;
        flag_dek=true;
        flag_stop=false;
        flag_go=false;
        //Log.d(TAG, "deksia");
    }
    if (((kuvos) >= (imgcenter-100)) && ((kuvos) <= (imgcenter+100)))
        flag_go=true;
        flag_ar=false;
        flag_dek=false;
        flag_stop=false;
        //Log.d(TAG, "kentro");
    }
    else{
        flag_ar=false;
        flag_dek=false;
        flag_stop=true;
        flag_go=false;
        //Log.d(TAG, "adeio");
    }
}
```

Αφού έπιλεξουμε το μεγαλύτερο περίγραμμα του προσθέτουμε γύρω του ένα ορθογώνιο πλαίσιο. Για τον έλεγχο και την αποστολή δεδομένων στον Arduino μέσω Bluetooth (για την οδήγησή του) ελέγχουμε εάν το ορθογώνιο πλαίσιο βρίσκεται στο μέσο της οθόνης του κινητού με περιθώριο 100 pixels ή αν βρίσκεται δεξιά ή αριστερά της οθόνης. Ο έλεγχος αυτός γίνεται με την χρήση flags.

```
        if (conflag==true){
            check_dir(flag_ar,flag_dek,flag_stop,flag_go);
        }
        if (mContours != null)
            mContours.clear();
        return mRgba;
    }

    public void check_dir(boolean dir,boolean dir1,boolean dir2,boolean dir3){
        byte [] buffer=new byte[1];
        if (dir==true){
            buffer[0]=left;
        }
        else if (dir1==true){
            buffer[0]=right;
        }
        else if (dir2==true){
            buffer[0]=stop;
        }
        else if (dir3==true){
            buffer[0]=go;
        }
        try{
            bsocket.getOutputStream().write(buffer);
        }
        catch (Exception e){
            Log.d(TAG,"de mporo na grapso");
        }
    }
}
```

Καλούμε την συνάρτηση «check_dir» όπου βλέπουμε ποιο flag είναι αληθής και στέλουμε το ανάλογο στον Arduino.


```

@Override
public void onResume() {
    super.onResume();
    OpenCVLoader.initAsync(OpenCVLoader.OPENCV_VERSION_2_4_7, this, mLoaderCallback);
}

@Override
public void onDestroy() {
    super.onDestroy();
    mRgba.release();
    try {
        bsocket.close();
    } catch (IOException e) {
        e.printStackTrace();
    }
    if (mOpenCvCameraView != null) {
        mOpenCvCameraView.disableView();
    }
}

@Override
public void onCameraViewStarted(int width, int height) {

    mRgba = new Mat();
    imgLine = new Mat();
    imgHSV = new Mat();
    imgHSV2 = new Mat();
}
    
```

Στην κλάση onCameraViewStarted ορίζουμε τις εικόνες. Στην κλάση onDestroy απελευθερώνει την μεταβλητή mRgba. Στην κλάση onResume δηλώνουμε ποια έκδοση OpenCv Manager θα χρησιμοποιήσουμε στο κινητό τηλέφωνο. Στην συγκεκριμένη περίπτωση χρησιμοποιούμε την έκδοση 2.4.7 ή και μεταγενέστερη. Η εγκατάσταση της εφαρμογής αυτής γίνεται στο Google Play Store όπως βλέπουμε στην εικόνα 2.17.



Εικόνα 2.17 Εφαρμογή OpenCv Manager

✓ Δημιουργία σύνδεσης με το Bluetooth:

```
private BluetoothSocket bsocket = null;
private static UUID my_uuid = UUID.fromString("00001101-0000-1000-8000-00805F9B34FB");
private String address = "20:16:01:20:65:38";
BluetoothAdapter bluetoothAdapter;
```

Αρχικά ορίζουμε την φυσική (MAC) διεύθυνση του bluetooth του Arduino και το Universally Unique Identifier (UUID) για να γίνει η σύνδεση.

```
conn = (Button) findViewById(R.id.conn);
conn.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        bluetooth();
    }
});
```

Στην κλάση onCreate δημιουργούμε ένα πλήκτρο «Conn» το οποίο όταν πατηθεί από τον χρήστη καλεί την συνάρτηση Bluetooth.

```
public void bluetooth(){
    final BluetoothManager bluetoothManager =(BluetoothManager) getSystemService(Context.BLUETOOTH_SERVICE);
    BluetoothAdapter = bluetoothManager.getAdapter();
    if (bluetoothAdapter == null || !bluetoothAdapter.isEnabled()) {
        Intent enableBtIntent = new Intent(BluetoothAdapter.ACTION_REQUEST_ENABLE);
        startActivityForResult(enableBtIntent, 1);
    }
    BluetoothDevice arduino=bluetoothAdapter.getRemoteDevice(address);
    try{
        bsocket=arduino.createRfcommSocketToServiceRecord(my_uuid);
    }
    catch (IOException e){
        Log.d(TAG,"socket not created");
        e.printStackTrace();
    }
    try {
        bsocket.connect();
        conflag=true;
    } catch (IOException e) {
        Log.d(TAG, "Cannot connect");
        conflag=false;
    }
}
```

Με αυτήν την συνάρτηση ελέγχουμε εάν το Bluetooth του κινητού είναι ενεργοποιημένο και εάν δεν είναι στένουμε αίτημα ενεργοποίησης και στην συνέχεια ολοκληρώνουμε την σύνδεση με τον Arduino. Όταν γίνει η σύνδεση ορίζουμε το «conflag» αληθής για να δώσουμε την κατεύθυνση που θα ακολουθήσει το όχημά μας.

Στην συνέχεια θα παρουσιάσουμε την εφαρμογή Android “PID_Controller”. Σε αυτό το κομμάτι θα δούμε ένα μέρος της εφαρμογής του android όπου με τη χρήση κουμπιών στέλνουμε δεδομένα στο Arduino έτσι ώστε να έχουμε εύκολη πρόσβαση στις τιμές του PID αλγόριθμου και η άμεση τροποποίησή τους. Για να το επιτύχουμε αυτό πράττουμε ως εξής :

- Δημιουργούμε μια καινούργια JavaClass στο Project που έχουμε δημιουργήσει. Δεξί κλικ → New → Java Class
- Ονομάζουμε η JavaClass με όνομα “PID_Controller” και προσθέτουμε τις βιβλιοθήκες που θα χρησιμοποιήσουμε.

```
package com.example.lefte.opencv4test;

import android.bluetooth.BluetoothAdapter;
import android.bluetooth.BluetoothDevice;
import android.bluetooth.BluetoothManager;
import android.bluetooth.BluetoothSocket;
import android.content.Context;
import android.content.Intent;
import android.os.Bundle;
import android.support.v7.app.AppCompatActivity;
import android.util.Log;
import android.view.View;
import android.view.WindowManager;
import android.widget.Button;
import android.widget.TextView;
import java.io.IOException;
import java.text.DecimalFormat;
import java.util.UUID;
```


- Δημιουργούμε σύνδεση με το Arduino μέσω Bluetooth

```
public void bluetooth(){
    flag_conn=false;
    final BluetoothManager bluetoothManager =(BluetoothManager) getSystemService(Context.BLUETOOTH_SERVICE);
    bluetoothAdapter = bluetoothManager.getAdapter();
    if (bluetoothAdapter == null || !bluetoothAdapter.isEnabled()) {
        Intent enableBtIntent = new Intent(BluetoothAdapter.ACTION_REQUEST_ENABLE);
        startActivityForResult(enableBtIntent, 1);
    }
    BluetoothDevice arduino=bluetoothAdapter.getRemoteDevice(address);
    try{
        bsocket=arduino.createRfcommSocketToServiceRecord(my_uuid);
    }
    catch (IOException e){
        Log.d(TAG, "socket not created");
        e.printStackTrace();
    }
    try {
        bsocket.connect();
        flag_conn=true;
        try{
            bsocket.getOutputStream().write(25);
        }
        catch (Exception e){
            Log.d(TAG,"de mporo na grapso to '25'");
        }
    } catch (IOException e) {
        Log.d(TAG,"Cannot connect");
        flag_conn=false;
        e.printStackTrace();
    }
}
```

- Μεταφερόμαστε στο φάκελο layout του Project μας και δημιουργούμε 2 καινούργια αρχεία με όνομα activity_layout.xml και activity_main2.xml όπου το activity_layout.xml περιέχεται μέσα στο activity_main2.xml όπως θα δούμε παρακάτω.

```

<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:paddingLeft="16dp"
    android:paddingRight="16dp"
    android:paddingTop="16dp"
    android:paddingBottom="16dp"
    app:layout_behavior="android.support.design.widget.AppBarLayout$ScrollingView..."
    tools:showIn="@layout/activity_main2"
    tools:context="com.example.lefte.pid_values.MainActivity">

    <Button
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Conn"
        android:id="@+id/conn"
        android:layout_below="@+id/set"
        android:layout_alignParentEnd="true" />

    <Button
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="P+1"
        android:id="@+id/p"
        android:layout_alignParentTop="true"
        android:layout_alignParentStart="true" />

    <Button
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="I+10"
        android:id="@+id/i"
        android:layout_below="@+id/p"
        android:layout_alignParentStart="true" />

    <Button
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="D+0.05"
        android:id="@+id/d"
        android:layout_below="@+id/i"
        android:layout_alignParentStart="true" />

```

```

<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:paddingLeft="16dp"
    android:paddingRight="16dp"
    android:paddingTop="16dp"
    android:paddingBottom="16dp"
    app:layout_behavior="android.support.design.widget.AppBarLayout$ScrollingView..."
    tools:showIn="@layout/activity_main2"
    tools:context="com.example.lefte.pid_values.MainActivity">

    <Button
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Conn"
        android:id="@+id/conn"
        android:layout_below="@+id/set"
        android:layout_alignParentEnd="true" />

    <Button
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="P+1"
        android:id="@+id/p"
        android:layout_alignParentTop="true"
        android:layout_alignParentStart="true" />

    <Button
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="I+10"
        android:id="@+id/i"
        android:layout_below="@+id/p"
        android:layout_alignParentStart="true" />

    <Button
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="D+0.05"
        android:id="@+id/d"
        android:layout_below="@+id/i"
        android:layout_alignParentStart="true" />

```

```
<Button
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Setpoint+0.01"
    android:id="@+id/set"
    android:layout_below="@+id/d"
    android:layout_alignParentStart="true" />

<Button
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="P-1"
    android:id="@+id/p_"
    android:layout_alignParentStart="true"
    android:layout_below="@+id/conn" />

<Button
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="I-10"
    android:id="@+id/i_"
    android:layout_alignParentStart="true"
    android:layout_below="@+id/p_"
    android:layout_alignParentLeft="true" />

<Button
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="D-0.05"
    android:id="@+id/d_"
    android:layout_alignParentStart="true"
    android:layout_below="@+id/i_"
    android:layout_alignParentLeft="true" />

<Button
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Setpoint-0.01"
    android:id="@+id/set_"
    android:layout_alignParentStart="true"
    android:layout_below="@+id/d_"
    android:layout_alignParentLeft="true" />
```

```
<TextView
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="79"
    android:id="@+id/textView1"
    android:layout_toStartOf="@+id/conn"
    android:layout_alignParentTop="true"
    android:layout_above="@+id/i" />

<TextView
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="460"
    android:id="@+id/textView2"
    android:layout_below="@+id/textView1"
    android:layout_alignStart="@+id/textView1"
    android:layout_alignBottom="@+id/i" />

<TextView
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="1.7"
    android:id="@+id/textView3"
    android:layout_above="@+id/set"
    android:layout_alignStart="@+id/textView2"
    android:layout_alignTop="@+id/d" />

<TextView
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="-8.3"
    android:id="@+id/textView4"
    android:layout_below="@+id/textView3"
    android:layout_toStartOf="@+id/conn"
    android:layout_above="@+id/conn" />

<Button
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Reset"
    android:id="@+id/reset"
    android:layout_alignBottom="@+id/set_"
    android:layout_alignParentEnd="true" />
```

```
<Button
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="left"
    android:id="@+id/Left"
    android:layout_below="@+id/set_"
    android:layout_alignParentStart="true"
    android:layout_alignEnd="@+id/d_" />

<Button
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="go"
    android:id="@+id/go"
    android:layout_alignTop="@+id/set_"
    android:layout_centerHorizontal="true" />

<Button
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="right"
    android:id="@+id/Right"
    android:layout_below="@+id/reset"
    android:layout_toEndOf="@+id/textView4" />

<Button
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Stop"
    android:id="@+id/Stop"
    android:layout_alignBottom="@+id/Left"
    android:layout_centerHorizontal="true" />

<Button
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Camera"
    android:id="@+id/camera"
    android:layout_below="@+id/p_"
    android:layout_alignParentEnd="true" />

</RelativeLayout>
```

Οι παραπάνω εικόνες είναι ο κώδικας για την εμφάνιση του γραφικού αποτελέσματος που βλέπουμε παρακάτω



Όπως βλέπουμε έχουμε δημιουργήσει το γραφικό μας περιβάλλον με τα εξής κουμπιά :

- ❖ “P+1”, “I+10”, “D+0.05”, “SETPOINT+0.01”, “P-1”, “I-10”, “D-0.05”, “SETPOINT-0.01”. Αυτά τα κουμπιά είναι για την αυξομείωση των μεταβλητών P,I,D που υπάρχουν στην εφαρμογή του Arduino.
- ❖ “GO”, “STOP”, “LEFT”, “RIGHT”. Είναι για την χειροκίνητη οδήγηση του Arduino.
- ❖ “CONN”, “CAMERA”, “RESET”. Είναι για τη σύνδεση της εφαρμογής με το Arduino, για την εναλλαγή με την Java class “Camera” και τέλος για να επαναφέρει όλες τις μεταβλητές στις αρχικές τους τιμές.

Για να βάλουμε το activity_layout.xml μέσα στο activity_main2.xml χρειάζεται ο παρακάτω κώδικας.

```
<?xml version="1.0" encoding="utf-8"?>
<android.support.design.widget.CoordinatorLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:fitsSystemWindows="true"
    tools:context=".Camera">
    <android.support.design.widget.AppBarLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:theme="@style/AppTheme.AppBarOverlay">
    </android.support.design.widget.AppBarLayout>
    <include layout="@layout/activity_pid" />
</android.support.design.widget.CoordinatorLayout>
```

Τέλος για να τελειώσουμε το Project μας πηγαίνουμε στη class "Pid_Controller" και δηλώνουμε τα κουμπιά που έχουμε δημιουργήσει στο layout και με κάθε πάτημα ενός κουμπιού θα έχει και διαφορετική ενέργεια. Εμείς όμως θέλουμε με τα κουμπιά των μεταβλητών να αυξάνεται ή να μειώνεται η τιμή της μεταβλητής ανάλογα με το κουμπί, με το κουμπί "CONN" να γίνεται η σύνδεση μέσω του Bluetooth, με το κουμπί "RESET" απλά μηδενίζονται όλες οι τιμές και εμφανίζονται όπως τις είχαμε ορίσει αρχικά και με τα κουμπιά τις κίνησης του δίτροχου να στέλνεται στο Arduino κατάλληλο byte ώστε να επιτυγχάνεται η σωστή κίνησή του.


```

conn = (Button) findViewById(R.id.conn);
conn.setOnClickListener((v) -> { bluetooth(); });

p=(Button) findViewById(R.id.p);
p.setOnClickListener((v) -> {
    pv=pv+1;
    TextView plv=(TextView) findViewById(R.id.textView1);
    plv.setText(pv + "");
    try {
        if(flag_conn==true)
            bsocket.getOutputStream().write(1);
        Log.d(TAG, "egrapsa 1");
    } catch (Exception e) {
        e.printStackTrace();
    }
});
i=(Button) findViewById(R.id.i);
i.setOnClickListener((v) -> {
    iv=iv+10;
    TextView ilv=(TextView) findViewById(R.id.textView2);
    ilv.setText(iv+"");
    try {
        if(flag_conn==true)
            bsocket.getOutputStream().write(2);
        Log.d(TAG, "egrapsa 2");
    } catch (Exception e) {
        e.printStackTrace();
    }
});
d=(Button) findViewById(R.id.d);
d.setOnClickListener((v) -> {
    dv=dv+0.05;
    TextView dlv=(TextView) findViewById(R.id.textView3);
    dlv.setText(new DecimalFormat("##.##").format(dv));
    try {
        Log.d(TAG, "egrapsa 3");
        if(flag_conn==true)
            bsocket.getOutputStream().write(3);
    } catch (Exception e) {
        e.printStackTrace();
    }
});
});

```

```

set=(Button)findViewById(R.id.set));
set.setOnClickListener((v) -> {
    sv=sv+0.01;
    TextView slv=(TextView) findViewById(R.id.textView4);
    slv.setText(new DecimalFormat("##.##").format(sv));
    try {
        if(flag_conn==true)
            bsocket.getOutputStream().write(4);
        Log.d(TAG, "egrapsa 4");
    } catch (Exception e) {
        e.printStackTrace();
    }
});

p1=(Button)findViewById(R.id.p_));
p1.setOnClickListener((v) -> {
    pv=pv-1;
    TextView plv=(TextView) findViewById(R.id.textView1);
    plv.setText(pv+"");
    try {
        Log.d(TAG, "egrapsa 5");
        if(flag_conn==true)
            bsocket.getOutputStream().write(5);
    } catch (Exception e) {
        e.printStackTrace();
    }
});

i2=(Button)findViewById(R.id.i_));
i2.setOnClickListener((v) -> {
    iv=iv-10;
    TextView ilv=(TextView) findViewById(R.id.textView2);
    ilv.setText(iv+"");
    try {
        if(flag_conn==true)
            bsocket.getOutputStream().write(6);
        Log.d(TAG, "egrapsa 6");
    } catch (Exception e) {
        e.printStackTrace();
    }
});

```

```

d2=(Button)findViewById(R.id.d_);
d2.setOnClickListener((v) → {
    dv=dv-0.05;
    TextView dlv=(TextView) findViewById(R.id.textView3);
    dlv.setText(new DecimalFormat("##.##").format(dv));
    try {
        if(flag_conn==true)
            bsocket.getOutputStream().write(7);
        Log.d(TAG, "egrapsa 7");
    } catch (Exception e) {
        e.printStackTrace();
    }
});

set2=(Button)findViewById(R.id.set_);
set2.setOnClickListener((v) → {
    sv=sv-0.01;
    TextView slv=(TextView) findViewById(R.id.textView4);
    slv.setText(new DecimalFormat("##.##").format(sv));
    try {
        if(flag_conn==true)
            bsocket.getOutputStream().write(8);
        Log.d(TAG, "egrapsa 8");
    } catch (Exception e) {
        e.printStackTrace();
    }
});

go=(Button)findViewById(R.id.go);
go.setOnClickListener((v) → {
    TextView v4=(TextView) findViewById(R.id.textView4);
    sv=sv+0.2;
    v4.setText(new DecimalFormat("##.##").format(sv));
    try {
        if(flag_conn==true)
            bsocket.getOutputStream().write(9);
        Log.d(TAG, "egrapsa 9");
    } catch (Exception e) {
        e.printStackTrace();
    }
});

```

```
stop=(Button)findViewById(R.id.Stop);
stop.setOnClickListener((v) -> {
    TextView v4=(TextView)findViewById(R.id.textView4);
    v4.setText("-8.3");
    sv=-8.3;
    try {
        if(flag_conn==true)
            bsocket.getOutputStream().write(10);
        Log.d(TAG, "egrapsa 10");
    } catch (Exception e) {
        e.printStackTrace();
    }
});

left=(Button)findViewById(R.id.Left);
left.setOnClickListener((v) -> {
    try {
        if(flag_conn==true)
            bsocket.getOutputStream().write(11);
        Log.d(TAG, "egrapsa 11");
    } catch (Exception e) {
        e.printStackTrace();
    }
});

right=(Button)findViewById(R.id.Right);
right.setOnClickListener((v) -> {
    try {
        if(flag_conn==true)
            bsocket.getOutputStream().write(12);
        Log.d(TAG, "egrapsa 12");
    } catch (Exception e) {
        e.printStackTrace();
    }
});
```

```
reset=(Button) findViewById(R.id.reset);
reset.setOnClickListener((v) -> {
    TextView v1=(TextView) findViewById(R.id.textView1);
    TextView v2=(TextView) findViewById(R.id.textView2);
    TextView v3=(TextView) findViewById(R.id.textView3);
    TextView v4=(TextView) findViewById(R.id.textView4);
    v1.setText("79");
    v2.setText("460");
    v3.setText("1.7");
    v4.setText("-8.3");
    pv=79;
    iv=460;
    sv=-8.3;
    dv=1.7;
});
}
```

ΚΕΦΑΛΑΙΟ 3

ΜΙΚΡΟΕΛΕΓΚΤΗΣ ARDUINO

3.1 Εισαγωγή

3.1.1 Εισαγωγή στους μικροελεγκτές

Ο μικροελεγκτής είναι ένα ολοκληρωμένο, που περιέχει ενσωματωμένα ένα τύπο επεξεργαστή, ουσιαστικά μια παραλλαγή μικροεπεξεργαστή, ο οποίος μπορεί να λειτουργήσει με ελάχιστα εξωτερικά εξαρτήματα, λόγω των πολλών ενσωματωμένων υποσυστημάτων που διαθέτει και διάφορα περιφερειακά όπως μνήμες RAM, ROM, για αποθήκευση δεδομένων και λογισμικού αντίστοιχα, μνήμη flash για μόνιμη αποθήκευση, CPU, πόρτες I/O, χρονιστές/απαριθμητές, μετατροπέα αναλογικού σε ψηφιακό σήμα και το αντίστροφο, timers και διάφορα άλλα. Χρησιμοποιείται ευρύτατα σε όλα τα ενσωματωμένα συστήματα (embedded systems) ελέγχου χαμηλού και μεσαίου κόστους, όπως αυτά που χρησιμοποιούνται σε αυτοματισμούς όπως ηλεκτρικές και ηλεκτρονικές συσκευές, ηλεκτρονικά καταναλωτικά προϊόντα, ηλεκτρικές συσκευές, κάθε είδους αυτοκινούμενα τροχοφόρα οχήματα, αεροπλάνα, συστήματα δικτύωσης συγχρονης κατοικίας, μηχανές CNC, προγραμματιζόμενους λογικούς ελεγκτές PLC, συστήματα ελέγχου προσπέλασης και πολλές άλλες εφαρμογές ελέγχου και αυτοματισμού. Στις μέρες μας η χρήση του μικροελεγκτή είναι καθολική για το λόγο ότι κάθε προϊόν το οποίο αλληλεπιδρά με έναν χρήστη περιλαμβάνει έναν μικροελεγκτή, ο οποίος παίζει τον ρόλο του «εγκεφάλου» των ηλεκτρονικών κυκλωμάτων.

Η χρήση των μικροελεγκτών μας προσφέρει πολλά πλεονεκτήματα όπως την αυτονομία, μέσω της ενσωμάτωσης σύνθετων περιφερειακών υποσυστημάτων όπως μνήμες και θύρες επικοινωνίας, το χαμηλό κόστος, μεγάλη αξιοπιστία. Επίσης η ενσωμάτωση περιφερειακών σημαίνει ευκολότερη υλοποίηση εφαρμογών λόγω των απλούστερων διασυνδέσεων και οδηγεί σε χαμηλότερη κατανάλωση ισχύος. Επιπρόσθετα, μειωμένες εκπομπές

ηλεκτρομαγνητικών παρεμβολών και μειωμένη ευαισθησία σε αντίστοιχες παρεμβολές από άλλες ηλεκτρικές και ηλεκτρονικές συσκευές. Τέλος, το μικρό μέγεθος συνολικού υπολογιστικού συστήματος.

Η πιο διαδεδομένη γλώσσα προγραμματισμού των μικροελεγκτών είναι η C, η C++ και οι παραλλαγές τους. Σε τμήματα του λογισμικού όπου απαιτείται ταχύτητα η μικρό μέγεθος χρησιμοποιούμενης μνήμης, μπορεί να χρησιμοποιείται η Assembly. Όμως οι μεγαλύτερες απαιτήσεις σε λειτουργικότητα και η ευκολία προγραμματισμού της C έναντι της assembly, σε συνδυασμό με την επάρκεια μνήμης των σύγχρονων μικροελεγκτών, έχουν γενικά εκτοπίσει την Assembly από τις περισσότερες εφαρμογές.

3.1.2 Εισαγωγή στον Arduino

Το Arduino είναι ένας μικροελεγκτής μονής πλακέτας, δηλαδή μια απλή μητρική πλακέτα ανοικτού κώδικα με ενσωματωμένο μικροελεγκτή και εισόδους/εξόδους, η οποία μπορεί να προγραμματιστεί με τη γλώσσα Wiring. Η γλώσσα Wiring πρόκειται για τη γλώσσα προγραμματισμού C++ και ένα σύνολο από βιβλιοθήκες, υλοποιημένες επίσης στην C++.

Το Arduino πρόκειται για ένα ηλεκτρονικό κύκλωμα που βασίζεται στον μικροελεγκτή ATmega της Atmel και του οποίου όλα τα σχέδια, καθώς και το software που χρειάζεται για την λειτουργία του, διανέμονται ελεύθερα και δωρεάν ώστε να μπορεί να κατασκευαστεί από τον καθένα. Αφού κατασκευαστεί, μπορεί να συμπεριφερθεί σαν ένας μικροσκοπικός υπολογιστής, αφού ο χρήστης μπορεί να συνδέσει επάνω του πολλαπλές μονάδες εισόδου/εξόδου και να προγραμματίσει τον μικροελεγκτή να δέχεται δεδομένα από τις μονάδες εισόδου, να τα επεξεργάζεται και να στέλνει κατάλληλες εντολές στις μονάδες εξόδου.



Εικόνα 3.1: Το λογότυπο του Arduino

Το Arduino αξίζει να αναφερθεί ότι επιτρέπει την συνδεσιμότητα της κύριας πλακέτας με ένα πλήθος ανταλλάξιμων καρτών επέκτασης, τα shields, όπως GPS, Wireless, Bluetooth, Ethernet κ.α. Επίσης, έχουν γίνει πολύ δημοφιλείς που χρησιμοποιούνται έκτος από εμπορικές και σε ερασιτεχνικές εφαρμογές με χαμηλό κόστος και μεγάλη ευκολία τροποποίησης των προγραμμάτων.

3.2 Εκδόσεις του Arduino

Παρακάτω παρουσιάζονται κάποια κύρια μοντέλα του Arduino που κυκλοφορούν στην αγορά. Στην δική μας εργασία που θα σας παρουσιάσουμε σε επόμενη ενότητα χρησιμοποιήσαμε τον Arduino UNO.

✓ Arduino UNO

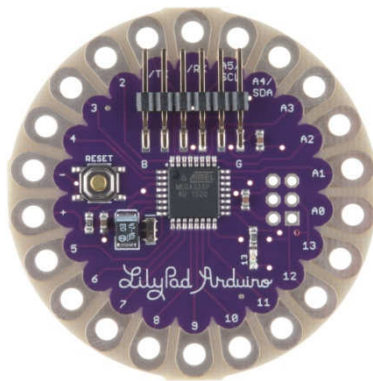
Το Arduino Uno χρησιμοποιεί τεχνολογία ATmega8U2 προγραμματισμένο ως σειριακός μετατροπέας. Έχει 14 ψηφιακές ακίδες εισόδου / εξόδου (από τις οποίες 6 μπορούν να χρησιμοποιηθούν ως έξοδο PWM), 6 αναλογικές εισοδοί, έναν ενισχυτή ήχου 16 MHz, μια σύνδεση USB, μια υποδοχή τροφοδοσίας, μια ICSP και ένα κουμπί reset.



Εικόνα 3.2: Arduino Uno

✓ Arduino LilyPad

Το Arduino LilyPad είναι ένα μινιμαλιστικό σχέδιο για εφαρμογές ένδυσης και E-textiles χρησιμοποιώντας τεχνολογία surface-mounted ATmega328. Έχει 14 ψηφιακές ακίδες εισόδου / εξόδου (από τις οποίες 6 μπορούν να χρησιμοποιηθούν ως έξοδο PWM), 6 αναλογικές εισοδοί. Η πλακέτα λειτουργεί από 2V έως 5V και έχει μεγάλες τρύπες για εύκολη σύνδεση.



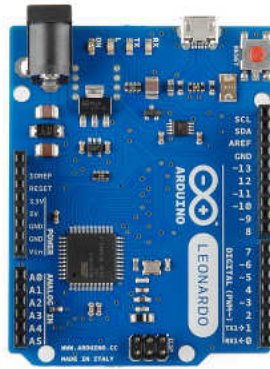
Εικόνα 3.3: Arduino LilyPad

✓ Arduino Leonardo

Το Arduino Leonardo χρησιμοποιεί ένα ATmega32U4 chip που εξαλείφει την ανάγκη για συνδεσιμότητα μέσω USB και μπορεί να χρησιμοποιηθεί ως ψηφιακό πληκτρολόγιο ή ποντίκι. Έχει 20 ψηφιακές ακίδες εισόδου / εξόδου (από τις οποίες 7 μπορούν να χρησιμοποιηθούν ως έξοδο PWM και 12 ως αναλογικές

Κίνηση Δίτροχου Οχήματος με την βοήθεια Arduino και Android

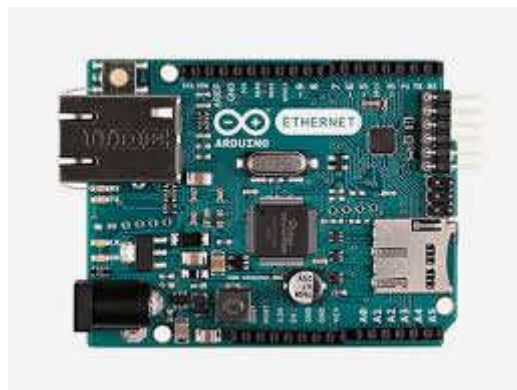
είσοδοι), έναν ταλαντωτή κρυστάλλου 16 MHz, μια σύνδεση micro USB, μια υποδοχή τροφοδοσίας, μια ICSP και ένα κουμπί reset.



Εικόνα 3.4: Arduino Leonardo

✓ Arduino Ethernet

Το Arduino Leonardo χρησιμοποιεί ένα ATmega328 chip. Έχει 14 ψηφιακές ακίδες εισόδου / εξόδου, 6 αναλογικές εισόδου, έναν ταλαντωτή κρυστάλλου 16 MHz, μια υποδοχή τροφοδοσίας, μια ICSP και ένα κουμπί reset.



Εικόνα 3.5: Arduino Ethernet

✓ Arduino Mega 2560

Το Arduino Mega 2560 χρησιμοποιεί τεχνολογία ATmega2560 φέρνοντας την ολική μνήμη στα 256kB. Επίσης ενσωματώνει την νέα τεχνολογία ATmega8U2 USB chipset. Διαθέτει 54 ψηφιακές υποδοχές εισόδου/εξόδου (από τις οποίες 14 μπορούν να χρησιμοποιηθούν ως PWM έξοδοι), 16 αναλογικές εισόδους, 4 UARTs (hardware serial ports), 1 κρυσταλλικό ταλαντωτή 16 MHz, υποδοχή USB, υποδοχή τροφοδοσίας ρεύματος, 1 ICSP και ένα κουμπί reset.



Εικόνα 3.6: Arduino Mega 2560

✓ Arduino Yun

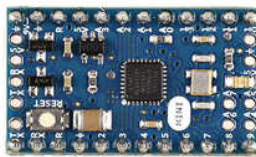
Το Arduino Yun το οποίο είναι βασισμένο στο ATmega32u4 και στο Atheros AR9331. Είναι συνδιασμός ενός κλασσικού Arduino Leonardo βασισμένο στον ATmega32U4 chip με ένα σύστημα WiFi σε ένα chip που τρέχει Linino.



Εικόνα 3.7: Arduino Yun

✓ Arduino Mini

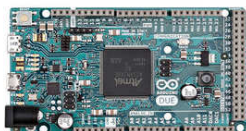
Το Arduino Mini είναι ένας μικρός μικροελεγκτής που αρχικά βασίστηκε στο ATmega168 αλλά τώρα βασίζεται στο ATmega328. Έχει 14 ψηφιακές ακίδες εισόδου / εξόδου(από τους οποίους 6 μπορούν να χρησιμοποιηθούν ως έξοδοι PWM), 8 αναλογικές εισοδοι, έναν ταλαντωτή κρυστάλλου 16 MHz.



Εικόνα 3.8: Arduino Mini

✓ Arduino Due

Το Arduino Due είναι ένα μικροχειριστήριο βασισμένο στην τεχνολογία Atmel SAM3X8E ARM Cortex-M3 CPU. Είναι το πρώτο board της Arduino βασισμένο σε επεξεργαστή 32-bit ARM microcontroller. Έχει 54 ψηφιακές ακίδες εισόδου / εξόδου(από τους οποίους 12 μπορούν να χρησιμοποιηθούν ως έξοδοι PWM), 12 αναλογικές εισοδοι, 4 UART (σειριακές θύρες), ρολόι 84MHz, ένα reset και ένα erase κουμπί.



Εικόνα 3.9: Arduino Due

3.3 Arduino Shields

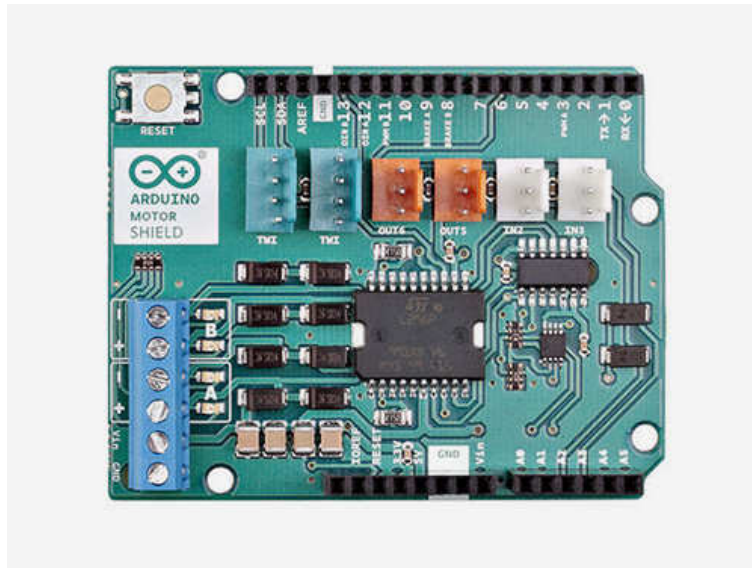
Τα Arduino χρησιμοποιούν την τεχνολογία των shields. Τα shield είναι τα εξαρτήματα που Arduino που είναι σχεδιασμένα ώστε να συνδέονται απευθείας με όλα τα pin του Arduino προεκτείνοντας την λειτουργικότητά του. Τα shield είναι σχεδιασμένα να προωθούν τις υποδοχές του, ώστε να μπορούμε να συνδέσουμε επιπλέον τα δικά μας εξαρτήματα ή να κουμπώσουμε και επόμενο shield. Φυσικά, το κάθε shield χρησιμοποιεί ορισμένους από τους πόρους συνδεσιμότητας του Arduino και έτσι δεν μπορούμε να συνδέσουμε απεριόριστα shield. Μάλιστα κάποια shield μπορεί να μην είναι συμβατά μεταξύ τους γιατί χρησιμοποιούν τα ίδια pin του Arduino για επικοινωνία με αυτό.

Ορισμένα δημοφιλή shields θα αναλύσουμε παρακάτω:

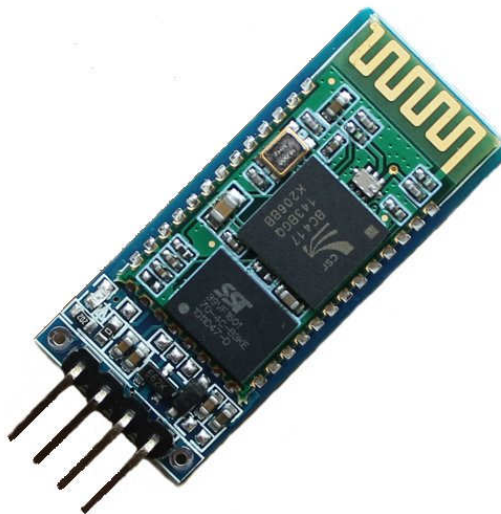
- ✓ Ethernet shield: Δίνει στο Arduino την δυνατότητα να δικτυωθεί σε ένα LAN ή στο internet μέσω ενός RJ45 καλώδιο.
- ✓ WiFi shield: Το Arduino WiFi Shield συνδέει το Arduino στο διαδίκτυο ασύρματα.
- ✓ Wireless SD Shield: Το Wireless SD Shield επιτρέπει σε μια πλακέτα Arduino να επικοινωνεί ασύρματα με μια ασύρματη μονάδα. Η μονάδα μπορεί να επικοινωνήσει έως και 100 πόδια σε εσωτερικούς χώρους ή σε εξωτερικούς χώρους ως 300 πόδια. Η μονάδα περιλαμβάνει μια θύρα υποδοχής SD.
- ✓ Screen Shield: Προσθέτουν οθόνες στο Arduino. Κυκλοφορούν από απλές οθόνες τύπου calculator μέχρι OLED touchscreen υψηλής ανάλυσης τύπου iPhone.
- ✓ Proto Shield: Μία προσχεδιασμένη πλακέτα συμβατή στις διαστάσεις του Arduino και χωρίς εξαρτήματα για να φτιάξετε το δικό σας shield.
- ✓ GPS shield: Προσθέτει δυνατότητες στο Arduino για εντοπισμό στίγματος.
- ✓ Wave shield: Δίνει στο Arduino την δυνατότητα να παίζει ήχους/μουσική από κάρτες SD.
- ✓ Motor Shield: Επιτρέπει την οδήγηση δύο DC κινητήρων από την ίδια συσκευή, ελέγχοντας την ταχύτητα και την κατεύθυνση του καθενός ξεχωριστά, όπως και άλλων μοτέρ servo, stepper κ.α.
- ✓ Bluetooth Shield: Δίνει στο Arduino την δυνατότητα να συνδεθεί με μια άλλη συσκευή μέσω Bluetooth

Κίνηση Δίτροχου Οχήματος με την βοήθεια Arduino και Android

Στην εργασία μας χρησιμοποιήσαμε διάφορα shields για την υλοποίηση του δίτροχου οχήματος όπως θα αναλυθεί στο κεφάλαιο 4.



Εικόνα 3.10: Arduino Motor Shield



Εικόνα 3.11: Arduino Bluetooth Shield

3.4 Χαρακτηριστικά του Arduino

Το Arduino βασίζεται στο μικροεπεξεργαστή ATmega328P της Atmel, έναν 8-bit AVR 8MHz μικροεπεξεργαστή ο οποίος διαθέτει 32KB ISP Flash μνήμη με δυνατότητες ανάγνωσης-εγγραφής, 1KB EEPROM, 2KB SRAM, έναν 16-bit και δύο 8-bit timers/counters, εσωτερικά και εξωτερικά interrupts, σειριακή προγραμματιζόμενη USART. Έχει επίσης εξωτερικό 16 MHz κρύσταλλο, με δυνατότητες επέκτασης στα 20 MHz. Ο κρύσταλλος αυτός συγχρονίζει τη λειτουργία του Arduino, αφού ενεργεί ως πηγή ρολογιού στέλνοντας σήματα ON-OFF τα οποία αλλάζουν τη κατάσταση του συστήματος.

Στον παρακάτω πίνακα αναφέρονται συνοπτικά τα χαρακτηριστικά του Arduino UNO που χρησιμοποιήσαμε:

Μικροελεγκτής	ATMEGA328
Τάση λειτουργίας	5V
Τάση εισόδου	7-12V
Όρια τάσης εισόδου	6-20V
Ψηφιακοί ακροδέκτες I/O	14, (6 PWM έξοδοι)
Αναλογικοί ακροδέκτες εισόδου	6
Ισχύς συνεχόμενου ρεύματος ανά ακροδέκτη	40mA
Ισχύς συνεχόμενου ρεύματος για ακροδέκτη τάσης 3.3V	50mA
Μνήμη flash	32KB(ATMEGA328)
Μνήμη SRAM	2KB (ATMEGA328)
Μνήμη EEPROM	1KB (ATMEGA328)
Ταχύτητα ρολογιού	16MHz

Πίνακας 3.1: Τα χαρακτηριστικά του Arduino

Το Arduino μπορεί να τροφοδοτηθεί μέσω θύρας USB (5V regulated). Η πλακέτα μπορεί να λειτουργήσει με εξωτερική τροφοδοσία 6-20V η οποία μειώνεται στα 5V από τον on-board regulator τάσης. Αν όμως τροφοδοτηθεί με λιγότερα από 7V τα pin εξόδου 5V δεν θα καταφέρουν να εξάγουν 5V. Αν ωστόσο

τροφοδοτήσουμε με 12V θα υπερθερμανθεί ο σταθεροποιητής τάσης στην πλακέτα και ενδεχομένως να καταστραφεί. Συνεπώς μια ιδανική τάση είναι τα 9V.

Οι ακροδέκτες τροφοδοσίας είναι οι ακόλουθοι:

- ✓ Vin: Η τάση εισόδου της πλακέτας όταν χρησιμοποιεί εξωτερική πηγή ενέργειας. Η μη σταθεροποιημένη τάση γίνεται μέσω αυτού του ακροδέκτη.
- ✓ 5V: Η τάση που χρησιμοποιείται από τα διάφορα μέρη της πλακέτας(ηλεκτρικά στοιχεία) και το μικροελεγκτή. Η τάση αυτή, την οποία δίνει αυτός ο ακροδέκτης, είναι είτε η τάση 5V που δίνει η σύνδεση με USB, είτε η ρυθμισμένη τάση που δίνεται μέσω του Vin.
- ✓ 3.3V: Η τάση αυτή παράγεται από το ολοκληρωμένο FTDI. Το όριο άντλησης ρεύματος είναι 50mA.
- ✓ GND: Ακροδέκτης γείωσης του συστήματος.

Ο μικροεπεξεργαστής ATmega328 έχει τρεις ομάδες μνήμης. Διαθέτει flash memory, στην οποία αποθηκεύονται τα Arduino sketch ο κώδικας που ανεβάζουμε, SRAM (static random access memory), στην οποία δημιουργείται το sketch και αποθηκεύει τις μεταβλητές όταν τρέχει, και EEPROM, η οποία χρησιμοποιείται από τους προγραμματιστές για την αποθήκευση μακροχρόνιων πληροφοριών. Αναλυτικότερα:

- ✓ 32KB μνήμης Flash: 2 KB χρησιμοποιούνται από το firmware του Arduino που έχει εγκαταστήσει ήδη ο κατασκευαστής του. Το firmware ή αλλιώς bootloader είναι αναγκαίο για την εγκατάσταση προγραμμάτων στο μικροελεγκτή μέσω της θύρας USB. Τα υπόλοιπα 30KB της μνήμης Flash χρησιμοποιούνται για την αποθήκευση αυτών ακριβώς των προγραμμάτων, αφού πρώτα μεταγλωττιστούν στον υπολογιστή. Η μνήμη Flash, δε χάνει τα περιεχόμενά της με απώλεια τροφοδοσίας ή επανεκκίνησης.
- ✓ 2KB μνήμης SRAM: Η ωφέλιμη μνήμη που μπορούν να χρησιμοποιήσουν τα προγράμματα για να αποθηκεύουν μεταβλητές, πίνακες κ.λπ. Η μνήμη χάνει τα δεδομένα της όταν η παροχή ρεύματος στο Arduino σταματήσει ή πατηθεί το κουμπί επανεκκίνησης όπως ένας ηλεκτρονικός υπολογιστής.
- ✓ 1KB μνήμης EEPROM: Μπορεί να χρησιμοποιηθεί για εγγραφή ή ανάγνωση δεδομένων από τα προγράμματα. Σε αντίθεση με την SRAM, δε

χάνει τα περιεχόμενά της με απώλεια τροφοδοσίας ή επανεκκίνησης όπως ο σκληρός δίσκος.

Το Arduino Uno διαθέτει 14 ψηφιακά θυληκά pin αριθμημένα από το 0 έως το 13 τα οποία μπορούν να οριστούν είτε σαν είσοδοι ,είτε σαν έξοδοι. Λειτουργούν στα 5V και καθένα από τα pin μπορεί να παρέχει ή να δεχτεί μέχρι 40mA. Από τα 14 αυτά ψηφιακά pins κάποια εκτός από την λειτουργία I/O έχουν και άλλες επιπρόσθετες λειτουργίες. Συγκεκριμένα:

- ✓ Ακροδέκτες 0 και 1: λειτουργούν ως RX και TX της σειριακής θύρας όταν το πρόγραμμα ενεργοποιεί τη σειριακή θύρα (UART). Έτσι, όταν το πρόγραμμα στέλνει δεδομένα στη σειριακή θύρα, αυτά προωθούνται και στη θύρα USB μέσω του 25 ελεγκτή Serial-Over-USB, αλλά και στον ακροδέκτη 0 για να τα διαβάσει ενδεχομένως μια άλλη συσκευή. Αυτό φυσικά σημαίνει ότι αν στο πρόγραμμα ενεργοποιήσει το σειριακό interface, χάνει 2 ψηφιακές εισόδους/εξόδους η πλατφόρμα.
- ✓ Ακροδέκτες 2 και 3: λειτουργούν και ως εξωτερικά interrupt (interrupt 0 και 1 αντίστοιχα). Ρυθμίζονται μέσα από το πρόγραμμα ώστε να λειτουργούν αποκλειστικά ως ψηφιακές είσοδοι στις οποίες όταν συμβαίνουν συγκεκριμένες αλλαγές, η κανονική ροή του προγράμματος σταματάει άμεσα και εκτελείται μια συγκεκριμένη συνάρτηση. Τα εξωτερικά interrupt είναι ιδιαίτερα χρήσιμα σε εφαρμογές που απαιτούν συγχρονισμό μεγάλης ακρίβειας.
- ✓ Ακροδέκτες 3, 5, 6, 9, 10 και 11: μπορούν να λειτουργήσουν και ως αναλογικές έξοδοι με το σύστημα PWM (Pulse Width Modulation) για να έχουμε ψηφιακούς παλμούς μεταβλητού πλάτους.
- ✓ Ακροδέκτης 8: input capture που χρησιμοποιείται για τον υπολογισμό συχνότητας ενός σήματος.
- ✓ Ακροδέκτες 12, 13: υποστηρίζουν επικοινωνία μεταξύ του Arduino και εξωτερικών περιφερειακών μονάδων.

3.5 Πλεονεκτήματα του Arduino

Το Arduino διαφέρει από τους άλλους μικροελεγκτές γιατί απλοποιεί την διαδικασία να δουλεύει κάποιος με μικροελεγκτές, αλλά κάποια πλεονεκτήματα που προσφέρει σε σχέση με άλλους μικροελεγκτές για χρήση από δασκάλους, μαθητές, προγραμματιστές. Διαθέτει αναλυτικές οδηγίες για κατασκευή μιας πλακέτας, όπως επίσης κυκλοφορούν προπαρασκευασμένες πλακέτες Arduino στο διαδίκτυο ή στα καταστήματα ηλεκτρονικών καλύπτοντας μεγάλο φάσμα λειτουργιών για την εξυπηρέτηση όλων. Είναι πολύ φθηνός σε σχέση με άλλους μικροελεγκτές, ακόμα και μια προπαρασκευασμένη πλακέτα είναι προσιτή οικονομικά.

Οι μηχανικοί λογισμικού, ανέπτυξαν το περιβάλλον προγραμματισμού του Arduino για διάφορα Λειτουργικά Συστήματα όπως είναι τα Windows, Machinstoh OSX και τα Linux, ενώ τα περισσότερα προγράμματα ανάπτυξης μικροελεγκτών περιορίζονται στα Windows.

Το περιβάλλον προγραμματισμού ενός Arduino ενδείκνυται για αρχάριους, αλλά είναι ταυτόχρονα και ευέλικτο και για πιο προχωρημένους χρήστες. Είναι βασισμένο σε Java και περιέχει ένα πρακτικό περιβάλλον για τη συγγραφή των προγραμμάτων, με συντακτική χρωματική σήμανση, μερικές έτοιμες βιβλιοθήκες για προέκταση του, τον compiler για τη μεταγλώττιση των sketch, μία σειριακή οθόνη (serial monitor) που παρακολουθεί τις επικοινωνίες της σειριακής (USB), αναλαμβάνει να στείλει αλφαριθμητικά στο Arduino μέσω αυτής και είναι ιδιαίτερα χρήσιμο για την αποσφαλμάτωση των sketch και την επιλογή για ανέβασμα των μεταγλωττισμένων sketch στο Arduino. Η γλώσσα του Arduino βασίζεται στη γλώσσα Wiring [15] μια παραλλαγή C/C++ και μπορούν να χρησιμοποιηθούν ουσιαστικά οι ίδιες βασικές εντολές και συναρτήσεις, με την ίδια σύνταξη, τους ίδιους τύπων δεδομένων και τους ίδιους τελεστές όπως και στη C. Πέρα από αυτές όμως, οι άνθρωποι που θέλουν να ασχοληθούν περισσότερο με τους μικροελεγκτές μπορούν να μεταβούν από τον Arduino στην AVR-C που είναι για προγραμματισμό των Atmel μικροελεγκτών και η γλώσσα στην οποία βασίστηκε το λογισμικό του Arduino.

3.6 Παρουσίαση της κατασκευής μας

Σε αυτήν την ενότητα θα παρουσιάσουμε την εφαρμογή Arduino για την κίνηση και την ισορροπία δίτροχου οχήματος και την επικοινωνία με την εφαρμογή Android «My Robot».

Αρχικά:

- ✓ Ανοίγουμε το πρόγραμμα arduino.exe
- ✓ Επιλέγουμε File → New
- ✓ File → Save As → myrobot.ino

Για την συγγραφή του κώδικα οι βιβλιοθήκες που θα χρειαστούμε είναι οι εξής:

```
#include <PID_v1.h>
#include <MPU6050_6Axis_MotionApps20.h>
#include <I2Cdev.h>
```

Περιγραφή βιβλιοθηκών:

- ✓ Η βιβλιοθήκη <PID_v1.h> μας βοηθάει να υπολογίσουμε την ταχύτητα στους τροχούς με σκοπό την επίτευξη της ισορροπίας.
- ✓ Η βιβλιοθήκη <MPU6050_6Axis_MotionApps20.h> και η βιβλιοθήκη <I2Cdev.h> σκοπό έχουν να διαβάσουμε τα δεδομένα από το αξελερόμετρο (MPU6050) και να υπολογίσουμε την περιστροφή στους τρεις άξονες (pitch, roll and yaw axis)

```

void setup() {
  // join I2C bus (I2Cdev library doesn't do this automatically)
  #if I2CDEV_IMPLEMENTATION == I2CDEV_ARDUINO_WIRE
    Wire.begin();
    Wire.setClock(400000); // 400kHz I2C clock. Comment this line if having compilation difficulties
  #elif I2CDEV_IMPLEMENTATION == I2CDEV_BUILTIN_FASTWIRE
    Fastwire::setup(400, true);
  #endif
  Serial.begin(115200);
  mpu.initialize();
  pinMode(INTERRUPT_PIN, INPUT);
  devStatus = mpu.dmpInitialize();
  mpu.setXGyroOffset(106);
  mpu.setYGyroOffset(88);
  mpu.setZGyroOffset(35);
  // make sure it worked (returns 0 if so)
  if (devStatus == 0) {
    // turn on the DMP, now that it's ready
    mpu.setDMPEnabled(true);
    attachInterrupt(digitalPinToInterrupt(INTERRUPT_PIN), dmpDataReady, RISING);
    mpuIntStatus = mpu.getIntStatus();
    dmpReady = true;
    // get expected DMP packet size for later comparison
    packetSize = mpu.dmpGetFIFOPacketSize();
  }

  pinMode(E1, OUTPUT); //install motors
  pinMode(E2, OUTPUT);
  pinMode(E3, OUTPUT);
  pinMode(E4, OUTPUT);

  pid.SetMode(AUTOMATIC); //install pid
  pid.SetSampleTime(10);
  pid.SetOutputLimits(-255, 255);

  pinMode(trigPin, OUTPUT); // Sets the trigPin as an Output
  pinMode(echoPin, INPUT); // Sets the echoPin as an Input
  attachInterrupt(1, interrupt, CHANGE); // Attach interrupt to the sensor echo input
}

```

Ο κώδικας που περιέχει η κλάση void setup θα εκτελεστεί μια φορά. Η λειτουργία του είναι να ορίζουμε την επικοινωνία με το αξελερόμετρο, τις ρυθμίσεις του PID και την οδήγηση των τροχών. Αρχικά γίνεται η επικοινωνία με το αξελερόμετρο ρυθμίζοντας το ρολόι 400 kHz και την σειριακή επικοινωνία 115200. Μετά βρίσκουμε την μετατόπιση απο το γυροσκόπιο x,y,z και την στέλνουμε στο αξελερόμετρο ώστε να έχουμε ελάχιστο σφάλμα στο διάβασμα των δεδομένων από το αξελερόμετρο. Στη συνέχεια, ορίζουμε τους τροχούς σαν έξοδο (output) και δίνουμε τις παραμέτρους στο PID. Τέλος, παίρνουμε τους δύο ακροδέκτες (pins) από τον αισθητήρα εντοπισμού αντικειμένων (HC SR04) και τους ορίζουμε σαν είσοδο και έξοδο αντίστοιχα.

```

void loop() {
  obstacle();
  if (!dmpReady) return;
  // wait for MPU interrupt or extra packet(s) available
  while (!mpuInterrupt && fifoCount < packetSize) {
    set_vars2();
    if (inputy < 18 && inputy > -17)
    {
      pid.SetTunings(kp, ki, kd);
      pid.Compute();
      Drive_Motor(output);
    }
  }
  else {
    motor_stop();
  }
}
// reset interrupt flag and get INT_STATUS byte
mpuInterrupt = false;
mpuIntStatus = mpu.getIntStatus();
// get current FIFO count
fifoCount = mpu.getFIFOCount();
// check for overflow (this should never happen unless our code is too inefficient)
if ((mpuIntStatus & 0x10) || fifoCount == 1024) {
  // reset so we can continue cleanly
  mpu.resetFIFO();
  Serial.println(F("FIFO overflow!"));
}
// otherwise, check for DMP data ready interrupt (this should happen frequently)
else if (mpuIntStatus & 0x02) {
  // wait for correct available data length, should be a VERY short wait
  while (fifoCount < packetSize) fifoCount = mpu.getFIFOCount();
  // read a packet from FIFO
  mpu.getFIFOBytes(fifoBuffer, packetSize);
  // track FIFO count here in case there is > 1 packet available
  // (this lets us immediately read more without waiting for an interrupt)
  fifoCount -= packetSize;
  // display Euler angles in degrees
  mpu.dmpGetQuaternion(&q, fifoBuffer);
  mpu.dmpGetGravity(&gravity, &q);
  mpu.dmpGetYawPitchRoll(ypr, &q, &gravity);
  inputy = ypr[2] * 180/M_PI;
  //Serial.print("input r=");Serial.print(inputy);Serial.println("\t");
}
}
}

```

Η κλάση void loop εκτελείται συνέχεια μέχρι να απενεργοποιηθεί ο μικροελεγκτής Arduino. Αρχικά, δημιουργούμε την επικοινωνία με το αξελερόμετρο, ελέγχουμε αν είναι συνδεδεμένο σωστά και αρχίζουμε να διαβάζουμε δεδομένα από αυτό. Ακολουθώντας, με την βιβλιοθήκη *<MPU6050_6Axis_MotionApps20.h>* υπολογίζουμε τους τρεις άξονες (pitch, roll and yaw axis) και χρησιμοποιούμε τον άξονα roll. Ελέγχουμε αν η τιμή του άξονα βρίσκεται μεταξύ -17 και 18 (αυτές οι τιμές είναι τα όρια ώστε το δίτροχο όχημα να μην ακουμπήσει στο έδαφος). Το PID ενεργοποιείται αν ο άξονας roll βρίσκεται μες στα όρια αυτών των τιμών και το όχημα ισορροπεί αλλιώς απενεργοποιούνται οι τροχοί. Στην συνέχεια, η συνάρτηση set_vars2() μας βοηθάει να ορίσουμε τις κατάλληλες τιμές στο PID. Τέλος, καλείται η συνάρτηση obstacle() που ελέγχει αν υπάρχει κάποιο εμπόδιο μπροστά από το όχημα.

```
int Drive_Motor(int speed)
{
  if(speed < 0){
    speed=min(speed, -1*minAbsSpeed); //orizo to deadzone sta motors sto 53
    speed=max(speed, -255);
    analogWrite(E2, 0);
    analogWrite(E1, -speed);
    analogWrite(E3, 0);
    analogWrite(E4, -speed);
  }
  else if(speed > 0){
    speed = max(speed, minAbsSpeed);
    speed = min(speed, 255);
    analogWrite(E2, speed);
    analogWrite(E1, 0);
    analogWrite(E3, speed);
    analogWrite(E4, 0);
  }
  else{
    motor_stop();
  }
}

void motor_stop(){
  analogWrite(E2, 0);
  analogWrite(E1, 0);
  analogWrite(E3, 0);
  analogWrite(E4, 0);
}
```

Στην παραπάνω συνάρτηση Drive_Motor() ελέγχουμε την τιμή της εξόδου του PID και αναλόγως αν είναι θετική, αρνητική ή μηδενική δίνεται στους αναλογους ακροδέκτες (pins) η τιμή αυτή και το δίτροχο όχημα κινείται μπροστά, με όπισθεν ή σταματάει αντίστοιχα. Επίσης υπάρχει μια «νεκρή ζώνη» με όρια 0-53 στους τροχούς με συνέπεια να μην κινούνται.

```
void set_vars2(){
  int x;
  if(Serial.available()>0){
    x=Serial.read();
    switch (x){
      case 1:
        kp=kp+1;
        break;
      case 2:
        ki=ki+10;
        break;
      case 3:
        kd=kd+0.05;
        break;
      case 4:
        Setpoint=Setpoint+0.01;
        break;
      case 5:
        kp=kp-1;
        break;
      case 6:
        ki=ki-10;
        break;
      case 7:
        kd=kd-0.05;
        break;
      case 8:
        Setpoint=Setpoint-0.01;
        break;
    }
  }
}
```

Έχοντας κάνει μια περιληπτική αναφορά της συνάρτησης set_vars() παραπάνω έδω βλέπουμε την δομή της και το πως δέχεται τα δεδομένα από την εφαρμογή «PID_val» που θα παρουσιάσουμε αναλυτικότερα παρακάτω.

```
void drive_from_mobile(){
  Serial.flush();
  if (Serial.available() > 0) {
    int inc=Serial.read();
    if (inc==0x1){
      flag_go=true;
      flag_left=false;
      flag_right=false;
      flag_stop=false;
    }
    if (inc==0x2){
      flag_left=true;
      flag_right=false;
      flag_stop=false;
      flag_go=false;
    }
    if (inc==0x3){
      flag_right=true;
      flag_left=false;
      flag_stop=false;
      flag_go=false;
    }
    if(inc==0x9){
      flag_stop=true;
      flag_right=false;
      flag_left=false;
      flag_go=false;
    }
  }
}

while(flag_left!=false){
  int inc=Serial.read();
  if (inc==0x2){
    flag_left=true;
    flag_right=false;
    flag_stop=false;
    flag_go=false;
  }
  if (inc==0x3){
    flag_right=true;
    flag_left=false;
    flag_stop=false;
    flag_go=false;
  }
  if(inc==0x9){
    flag_stop=true;
    flag_right=false;
    flag_left=false;
    flag_go=false;
  }
  if (inc==0x1){
    flag_go=true;
    flag_left=false;
    flag_right=false;
    flag_stop=false;
  }
}
```



```
while(flag_right!=false){  
  
    int inc=Serial.read();  
    if (inc==0x2){  
        flag_left=true;  
        flag_right=false;  
        flag_stop=false;  
        flag_go=false;  
    }  
    if (inc==0x3){  
        flag_right=true;  
        flag_left=false;  
        flag_stop=false;  
        flag_go=false;  
    }  
    if(inc==0x9){  
        flag_stop=true;  
        flag_right=false;  
        flag_left=false;  
        flag_go=false;  
    }  
    if (inc==0x1){  
        flag_go=true;  
        flag_left=false;  
        flag_right=false;  
        flag_stop=false;  
    }  
}
```

```
while(flag_stop!=false){  
    Setpoint=originalSetpoint;  
    int inc=Serial.read();  
    if (inc==0x2){  
        flag_left=true;  
        flag_right=false;  
        flag_stop=false;  
        flag_go=false;  
    }  
    if (inc==0x3){  
        flag_right=true;  
        flag_left=false;  
        flag_stop=false;  
        flag_go=false;  
    }  
    if(inc==0x9){  
        flag_stop=true;  
        flag_right=false;  
        flag_left=false;  
        flag_go=false;  
    }  
    if (inc==0x1){  
        flag_go=true;  
        flag_left=false;  
        flag_right=false;  
        flag_stop=false;  
    }  
}
```

```

while(flag_go!=false){
  if(flag_distance==false){
    Setpoint=originalSetpoint+movingAngleOffset;
  }
  else if(flag_distance==true){
    Setpoint=originalSetpoint;
  }
  int inc=Serial.read();
  if (inc==0x2){
    flag_left=true;
    flag_right=false;
    flag_stop=false;
    flag_go=false;
  }
  if (inc==0x3){
    flag_right=true;
    flag_left=false;
    flag_stop=false;
    flag_go=false;
  }
  if(inc==0x9){
    flag_stop=true;
    flag_right=false;
    flag_left=false;
    flag_go=false;
  }
  if (inc==0x1){
    flag_go=true;
    flag_left=false;
    flag_right=false;
    flag_stop=false;
  }
}
}

```

Η κλάση void drive_from_mobile() επικοινωνεί με το κινητό και αναλόγως τις τιμές που λαμβάνει δίνει την κατεύθυνση στο δίτροχο όχημα.

```

void interrupt(){
  switch (digitalRead(echoPin))// Test to see if the signal is high or low
  {
    case HIGH:// High so must be the start of the echo pulse
      echo_end = 0;// Clear the end time
      echo_start = micros();// Save the start time
      break;
    case LOW:// Low so must be the end of hte echo pulse
      echo_end = micros();// Save the end time
      echo_duration = echo_end - echo_start;// Calculate the pulse duration
      break;
  }
}

void obstacle(){
  digitalWrite(trigPin, HIGH);
  delayMicroseconds(10);
  digitalWrite(trigPin, LOW);
  distance=echo_duration*0.034/2;
  if (distance <= 5){
    flag_distance=true;
  }
}
}

```

Στις παραπάνω δύο συναρτήσεις void interrupt() και void obstacle() γίνεται ο εντοπισμός του αντικειμένου με το εξής τρόπο. Αρχικά, ο ένας ακροδέκτης του αισθητήρα γίνεται 1 (HIGH) και για 10μsec στέλνει παλμό και ο άλλος ακροδέκτης περιμένει να λάβει τον παλμό και όταν το λάβει υπολογίζει την απόσταση. Επειδή αυτό μας προκαλούσε καθυστέρηση ο ακροδέκτης που περιμένει τον παλμό τον βάζουμε σε διακόπτη (interrupt).

ΚΕΦΑΛΑΙΟ 4

ΥΛΟΠΟΙΗΣΗ ΔΙΤΡΟΧΟΥ ΟΧΗΜΑΤΟΣ

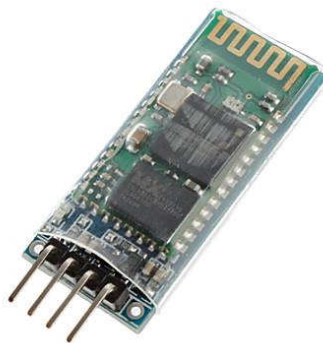
4.1 Εξαρτήματα Κατασκευής

Σε αυτήν την ενότητα θα παρουσιάσουμε τα εξαρτήματα που χρειαστήκαμε για να υλοποιήσουμε την κατασκευή δίτροχου οχήματος. Για το ηλεκτρονικό κομμάτι της κατασκευής χρειαζόμαστε τα εξής:

- ✓ 1x HC-06 Bluetooth Module
- ✓ 1x HC-SR04 Sensor
- ✓ 1x MiniQ Base
- ✓ 1x MPU6050
- ✓ 1x Arduino Uno
- ✓ 1x L293D Shield
- ✓ 1x Battery 9V
- ✓ 2x N20 Motor

4.1.1 HC-06 Bluetooth Module

Το HC-06 Bluetooth Module είναι μια ασύρματη σειριακή θύρα για την δημιουργία σύνδεσης μεταξύ Arduino και μιας συσκευής Bluetooth (π.χ. κινητό τηλέφωνο, ηλεκτρονικός υπολογιστής κ.α). Τα εργοστασιακά χαρακτηριστικά του είναι ο κωδικός για να γίνει ζεύξη με μια συσκευή είναι 1234 και ο ρυθμός μετάδοσης (Baud Rate) είναι 9600.



Εικόνα 4.1: HC-06 Bluetooth Module

4.1.2 HC-SR04 Sensor

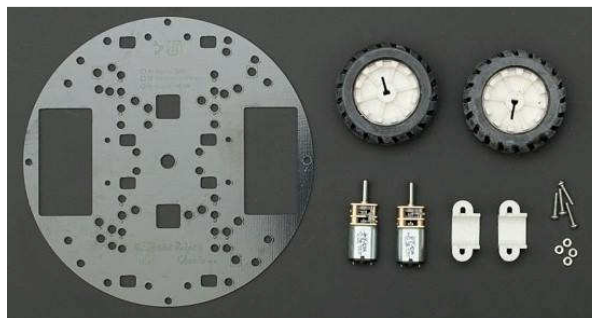
Το HC-SR04 Sensor είναι ένας αισθητήρας που εντοπίζει την απόσταση από ένα αντικείμενο. Αποτελείται από τέσσερις ακροδέκτες (pins), οι οποίοι είναι Power, Trigger, Echo και Ground. Για να εντοπίσουμε την απόσταση από ένα αντικείμενο θέτουμε τον ακροδέκτη Trigger «HIGH» για 10msec με αποτέλεσμα την αποστολή με ταχύτητα του ήχου 8 κύκλων παλμών και το δέχεται ο ακροδέκτης Echo.



Εικόνα 4.2: HC-SR04 Sensor

4.1.3 MiniQ Base

Το MiniQ είναι η πλαστική βάση της κατασκευής μας και αποτελείται από δύο κινητηράκια και δύο ρόδες.



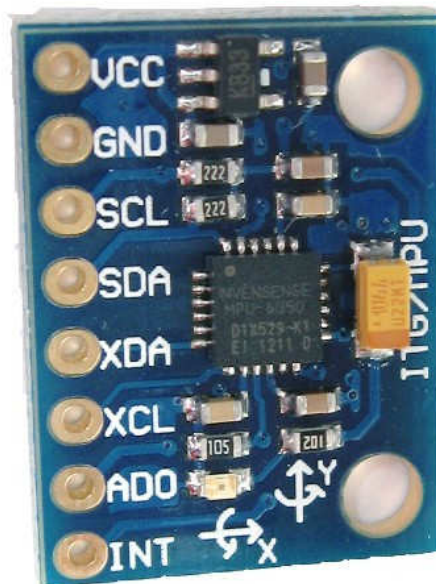
Εικόνα 4.3: MiniQ



Εικόνα 4.4: Συνδεδεμένο MiniQ

4.1.4 MPU6050

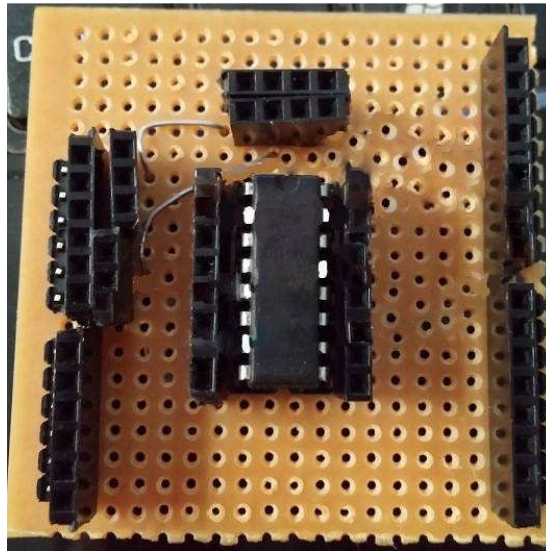
Το MPU6050 είναι μία πλακέτα που περιέχει αξελερόμετρο (accelerometer) και γυροσκόπιο (gyroscope). Οι βασικές λειτουργίες του είναι ότι μπορεί και διαβάζει ταυτόχρονα 3 κανάλια x,y,z στο αξελερόμετρο και στο γυροσκόπιο. Επίσης περιέχει και αισθητήρα θερμοκρασίας με ένα κανάλι που διαβάζει τα δεδομένα του.



Εικόνα 4.5: MPU6050

4.1.5 L293D Shield

Το L293D Shield είναι ένα ολοκληρωμένο που μας βοηθάει στην οδήγηση των 2 N20 κινητήρων. Επίσης μας παρέχει μία υποδοχή για την τοποθέτηση του Bluetooth.



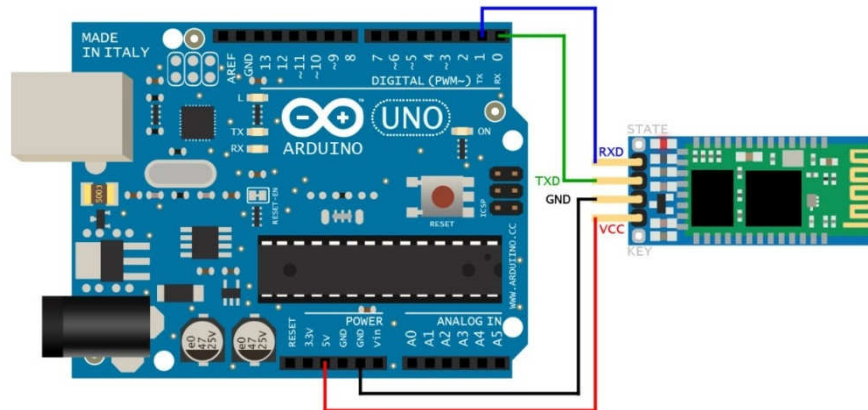
Εικόνα 4.6: L293D Shield

4.2 Συνδεσμολογία Εξαρτημάτων

Ο εγκέφαλος της κατασκευής μας είναι ο Arduino Uno και όλα τα εξαρτήματα συνδέονται σε αυτόν. Αρχικά, το ολοκληρωμένο L293D είναι προέκταση του Arduino και τον τοποθετούμε πάνω του σε όλους τους ακροδέκτες. Στην συνέχεια, συνδέουμε τα 2 κινητηράκια N20 στο ολοκληρωμένο L293D. Για το πρώτο κινητηράκι τον έναν ακροδέκτη τον συνδέουμε στον ακροδέκτη «2» του ολοκληρωμένου L293D ενώ τον άλλο στον ακροδέκτη «7» του ολοκληρωμένου. Για το δεύτερο κινητηράκι τον έναν ακροδέκτη τον συνδέουμε στον ακροδέκτη «10» του ολοκληρωμένου L293D ενώ τον άλλο στον ακροδέκτη «15» του ολοκληρωμένου. Επιπρόσθετα, για την κίνηση του ενός κινητήρα συνδέουμε τον ακροδέκτη «3» του ολοκληρωμένου L293D με τον ακροδέκτη «10» του Arduino και τον ακροδέκτη «6» του ολοκληρωμένου L293D με τον ακροδέκτη «9» του Arduino. Για την κίνηση του δεύτερου κινητήρα συνδέουμε τον ακροδέκτη «11» του ολοκληρωμένου L293D με τον ακροδέκτη «5» του Arduino και τον ακροδέκτη

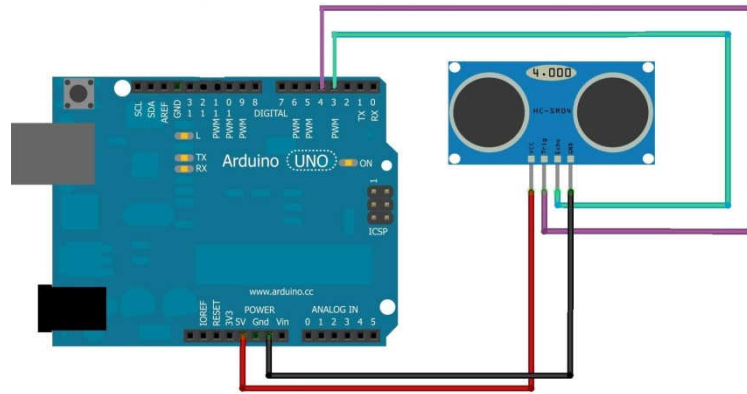
«14» του ολοκληρωμένου L293D με τον ακροδέκτη «6» του Arduino. Τέλος, για την τροφοδοσία του ολοκληρωμένου L293D συνδέουμε τον ακροδέκτη «8» με τον ακροδέκτη «5V» του Arduino, ενώ για την γείωση του ολοκληρωμένου L293D συνδέουμε τον ακροδέκτη «4» με τον ακροδέκτη «GND» του Arduino.

Για την συνδεσμολογία του Bluetooth με τον Arduino αρχικά τοποθετούμε το Bluetooth στην υποδοχή του Shield που διαθέτει τροφοδοσία και γείωση. Στην συνέχεια, συνδέουμε τον ακροδέκτη «Rx» του Bluetooth με τον ακροδέκτη «Tx» του Arduino και το αντίστροφο συνδέουμε τον ακροδέκτη «Tx» του Bluetooth με τον ακροδέκτη «Rx» του Arduino.



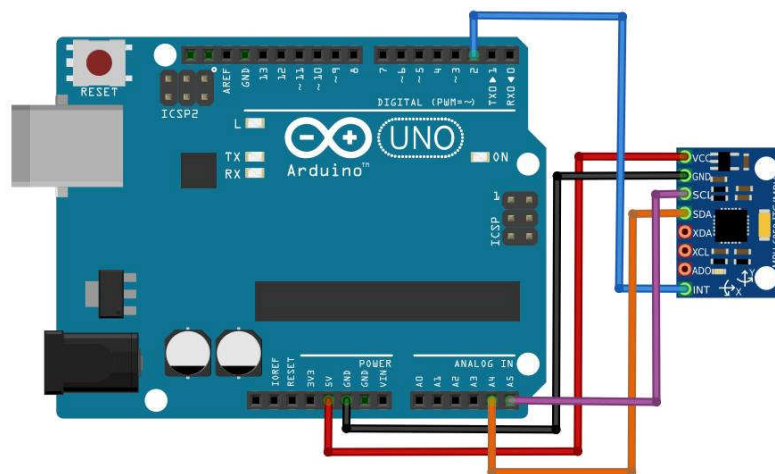
Εικόνα 4.7: Συνδεσμολογία Bluetooth με Arduino

Για την σύνδεση του αισθητήρα HC-SR04 με τον Arduino αρχικά τοποθετούμε τους δύο ακροδέκτες, τροφοδοσίας και γείωσης, με τους αντίστοιχους στον Arduino. Έπειτα τοποθετούμε τον ακροδέκτη «Trigger» στον ακροδέκτη «4» του Arduino και τον ακροδέκτη «Echo» στον ακροδέκτη «3» του Arduino (είναι και interrupt υποδοχή στον Arduino). Η τοποθέτηση του αισθητήρα HC-SR04 γίνεται μπροστά στο δίκυκλο όχημα και όσο το δυνατόν πιο κοντά στο έδαφος.



Εικόνα 4.8: Συνδεσμολογία HC-SR04 με Arduino

Η σύνδεση του MPU6050 με τον Arduino επιτυγχάνεται με την ένωση των ακροδεκτών τροφοδοσίας και γείωσης με τους αντίστοιχους του Arduino. Επίσης, ενώνουμε τον ακροδέκτη «SCL» στον αναλογικό ακροδέκτη «A5» του Arduino, ενώ τον ακροδέκτη «SDA» στον αναλογικό ακροδέκτη «A4» του Arduino. Τέλος, ο ακροδέκτης «INT» του MPU6050 στον ακροδέκτη «2» του Arduino (interrupt υποδοχή στον Arduino όπως και ο ακροδέκτης «3»). Η τοποθέτηση του MPU6050 γίνεται με προσοχή έτσι ώστε ο αισθητήρας να είναι όσο το δυνατόν ακίνητος.



Εικόνα 4.9: Συνδεσμολογία MPU6050 με Arduino

ΚΕΦΑΛΑΙΟ 5

ΣΥΜΠΕΡΑΣΜΑΤΑ ΚΑΙ ΠΡΟΟΠΤΙΚΕΣ

5.1 Σύνοψη της πτυχιακής εργασίας

Από τη πτυχιακή εργασία που ολοκληρώσαμε μπορούμε εύκολα να συμπεράνουμε ότι μια κατασκευή μπορεί να υλοποιηθεί εύκολα με πολύ απλά βήματα. Αρχικά η κατασκευή του 2-τροχου έγινε με κομμάτια ξύλου, με τον Arduino, με ένα shield(I293D) με 2 τροχούς με τα αντίστοιχα motors και τη συσκευή Bluetooth. Όλα τα παραπάνω συνδέθηκαν μεταξύ τους και δημιούργησαν το 2-τροχο. Στη συνέχεια δημιουργήσαμε την εφαρμογή Androidόπου συνδέεται μέσω του Bluetooth με το 2-τροχο και λειτουργεί σα τα «μάτια» του Arduino, δηλαδή ότι βλέπει η εφαρμογή στο Androidδίνει και τις αντίστοιχες εντολές στον Arduino. Από ότι βλέπουμε η επικοινωνία Arduino και Androidγίνετε πολύ εύκολα μέσω του Bluetoothκαι η επικοινωνία τους είναι άμεση και γρήγορη. Τέλος η ισορροπία του Arduinoήταν το πιο δύσκολο κομμάτι άλλα με πολύ υπομονή και πολλές προσπάθειες βρήκαμε τον κατάλληλο PIDαλγόριθμο με τις κατάλληλες τιμές για την καλύτερη ισορροπία που μπορούσαμε να πετύχουμε. Καταλήγουμε λοιπόν στο ότι η ισορροπία είναι ένα δύσκολο κομμάτι και ότι για κάθε κατασκευή οι τιμές του PIDείναι διαφορετικές.

5.2 Προοπτικές

Η πτυχιακή εργασία έχει ως σκοπό της την αυτονομία. Το δίτροχο μπορεί να ισορροπεί μόνο του χωρίς κάποιας εντολής του χρήστη ,είναι αυτό-ισορροπούμενο, να διαβάζει τα δεδομένα από την εφαρμογή στο κινητό να εστιάζει στο κέντρο το μαύρο αντικείμενο και να προχωράει προς τα μπροστά . Η εφαρμογή στο κινητό έχει ως σκοπό της τον εντοπισμό του μεγαλύτερου αντικειμένου χρώματος μαύρου, την ρύθμιση του PIDαλγόριθμου για την ισορροπία του δίτροχου και για την επικοινωνία με το δίτροχο (αποστολή δεδομένων). Αυτή είναι η βασική ιδέα της πτυχιακής εργασίας που μπορεί όμως να αναπτυχθεί και να δημιουργηθούν πολλές

καινούργιες προοπτικές όπως τον εντοπισμό πινακίδων και την ανάλογη αντίδραση από το δίτροχο, π.χ. καθώς ισορροπεί το δίτροχο να σταματάει αν η πινακίδα έχει το σύμβολο του STOP ή να στρίβει δεξιά ή αριστερά ανάλογα τη πινακίδα που θα εντοπίζει. Μια άλλη προοπτική είναι να μπορεί να επαναφέρεται σε θέση ισορροπίας αν το δίτροχο βρίσκεται εκτός των ορίων ισορροπίας ,δηλαδή εάν κάποια άκρη του (μπροστά ή πίσω) ακουμπάει στο έδαφος. Επίσης μια καινούργια λειτουργία θα μπορούσε να είναι η περιστροφή του δίτροχου για εύρεση κάποιου μαύρου αντικειμένου σε περίπτωση που δεν έχει εντοπίσει. Ακόμα μπορούμε να το προσθέσουμε την αποφυγή εμποδίων. Το δίτροχο στην πτυχιακή εργασία έχει έναν αισθητήρα αντικειμένων όπου εάν βρει κάποιο αντικείμενο (εμπόδιο για το δίτροχο) θα σταματήσει να κινείται. Για την αποφυγή εμποδίων μπορούμε να προσθέσουμε ακόμα 2-3 αισθητήρες έτσι ώστε να μπορεί να βρίσκει εναλλακτική διαδρομή για να αποφεύγει τα εμπόδια. Και επειδή όπως ξέρουμε στο προγραμματισμό δεν υπάρχουν όρια οι προοπτικές που έχει η κατασκευή μας είναι πάρα πολλές!

ΒΙΒΛΙΟΓΡΑΦΙΑ

- [1] Open Handset Alliance (5 Νοεμβρίου 2007). Industry Leaders Announce Open Platform for Mobile Devices. Δελτίο τύπου.
- [2] Shankland, Stephen (12 Νοεμβρίου 2007). «Google's Android parts ways with Java industry group». CNET News.
- [3] Brownlee, John. *"Conversational Interfaces, Explained"*. Fast Co. Design. Fast Company Inc. Retrieved July 4, 2016.
- [4] W.M. Lee, 2011, "Beginning Android Application Development", Wrox
- [5] M. Murphy, 2011, "Android Programming Tutorials, 3rd Edition", CommonsWare
- [6] <https://developer.android.com/about/index.html>
- [7] Schmidt, M. ["Arduino: A Quick Start Guide"], *Pragmatic Bookshelf*, January 22 2011, Pg. 201
- [8] "Shields fuer Arduino wenig Aufwand Selbst Bauen". web.de
- [9] Igoe, Tom (April 4, 2006). "Arduino Shields for Prototyping". tigoe.net
- [10] Arduino, <http://arduino.cc/en/Guide/Environment>, Σεπτέμβριος 2012
- [11] Wiring, <http://www.wiring.org.co/>, Σεπτέμβριος 2012
- [12] Arduino playground Ethernet, <http://arduino.cc/en/Reference/Ethernet>, Σεπτέμβριος 2012
- [13] Arduino Hardware. <http://arduino.cc/en/Main/Hardware>
- [14] Atmel ATmega328. <http://www.atmel.com/devices/atmega328.aspx>
- [15] <http://www.microplanet.gr/tutorials/microcontrollers/arduino>
- [16] <http://www.grobot.gr/index.php/2008-04-19-13-16-38/197-arduino-30432>
- [17] http://www.openhandsetalliance.com/android_overview.html
- [18] <http://www.allaboutandroid.gr>
- [19] <https://el.wikipedia.org/wiki/Arduino>
- [20] The history of Android, Ars Technica, by Ron Amadeo
- [21] <https://www.howtogeek.com/99041/the-making-of-arduino-geek-history/>
- [22] <http://www.android-app-market.com/androidarchitecture.html>.
- [23] Lahart, Justin (2009-11-27). «[Taking an Open-Source Approach to Hardware](#)»