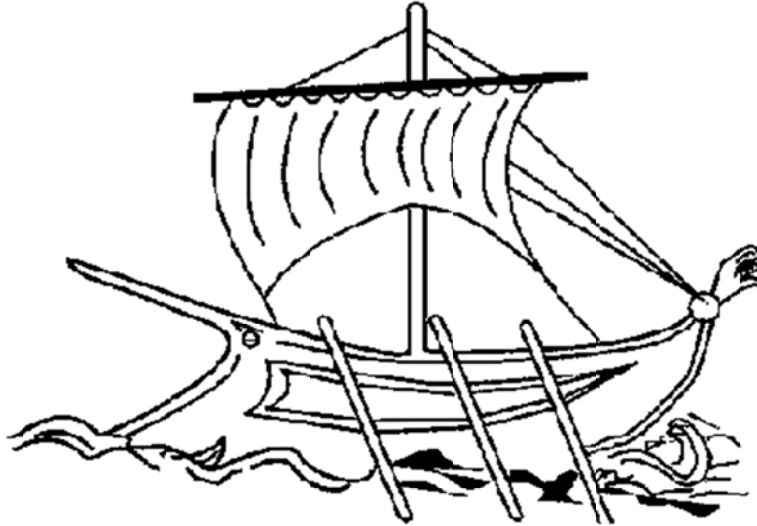


ΤΕΧΝΟΛΟΓΙΚΟ ΕΚΠΑΙΔΕΥΤΙΚΟ ΙΔΡΥΜΑ ΠΕΙΡΑΙΑ



Τ.Ε.Ι ΠΕΙΡΑΙΑ

Ανάπτυξη εφαρμογής για τον απομακρυσμένο έλεγχο λυχνιών Led

ΟΡΕΣΤΗΣ ΝΙΚΟΠΟΥΛΟΣ

ΔΗΜΗΤΡΗΣ ΠΥΡΟΜΑΛΗΣ

ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ

ΣΧΟΛΗ ΤΕΧΝΟΛΟΓΙΚΩΝ ΕΦΑΡΜΟΓΩΝ

ΤΜΗΜΑ ΑΥΤΟΜΑΤΙΣΜΟΥ

ΦΕΒΡΟΥΑΡΙΟΣ 2017 (20/2/2017)

ΤΜΗΜΑ ΑΥΤΟΜΑΤΙΣΜΟΥ
Π Ράλλη & Θηβών 250, 12244 Αιγάλεω, Αθήνα-Ελλάδα.
Τηλ.: 210-5381488

ΔΗΛΩΣΗ ΣΥΓΓΡΑΦΕΑ ΠΤΥΧΙΑΚΗΣ ΕΡΓΑΣΙΑΣ

Ο / Η κάτωθι υπογεγραμμένος / χ Ορέτζης Νικόπουλος,
του Λεωνίδα - Γεωργιάδη, με αριθμό μητρώου 32419 φοιτητής / τριά του
Τμήματος **Μηχανικών Αυτοματισμού Τ.Ε.** του Α.Ε.Ι. Πειραιά Τ.Τ. πριν αναλάβω την
εκπόνηση της Πτυχιακής Εργασίας μου, δηλώνω ότι ενημερώθηκα 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 απονεμίσει 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 βμήνου από την ημερομηνία ανάθεσης της. Κατά τα λοιπά εφαρμόζονται τα
προβλεπόμενα στο άρθρο 18, παρ. 5 του ισχύοντος Εσωτερικού Κανονισμού.»

Ο Δηλών



Ημερομηνία

5/1/2019

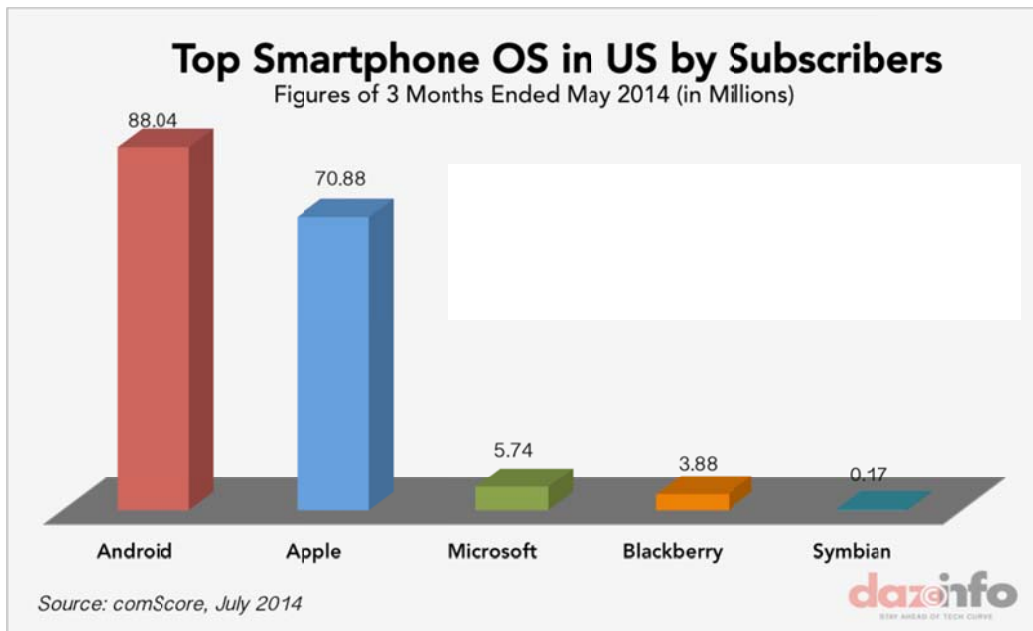
Πίνακας Περιεχομένων

Αίτηση Εκπόνησης Πτυχιακής Εργασίας.....	σελ.: 3
Αίτηση Υποβολής Πτυχιακής Εργασίας	σελ.: 4
Πρακτικό Εξέτασης & Βαθμολόγησης Πτυχιακής Εργασίας	σελ.: 5
Εισηγητική Έκθεση Πτυχιακής Εργασίας.....	σελ.: 6
Πίνακας Περιεχομένων	σελ.: 7
Περίληψη	σελ.: 8
Λειτουργικό Σύστημα.....	σελ.: 9
Το Android της Google	σελ.: 11
Το iOS της Apple	σελ.: 13
Διαφορές των δύο λειτουργικών	σελ.: 14
Εφαρμογές κινητών συσκευών	σελ.: 16
Bluetooth	σελ.: 18
Εφαρμογές του Bluetooth	σελ.: 19
Η λειτουργία και δομή του Bluetooth	σελ.: 20
Bluetooth 4 – Bluetooth Low Energy (BLE)	σελ.: 21
Bluetooth σε συσκευές τις Apple	σελ.: 23
NFC	σελ.: 24
Σύγκριση Bluetooth και NFC	σελ.: 25
Προγραμματισμός	σελ.: 27
Γλώσσα προγραμματισμού C++	σελ.: 29
Γλώσσα προγραμματισμού Java	σελ.: 30
Διαφορές μεταξύ Java και C++	σελ.: 33
Εργαλεία ανάπτυξης εφαρμογών	σελ.: 35
Πληροφορίες εφαρμογής	σελ.: 37
Μικροελεγκτής	σελ.: 41
Arduino	σελ.: 43
Συνδεσμολογία	σελ.: 44
Προγραμματισμός Πλακέτας Arduino	σελ.: 45
Σκοπός	σελ.: 46
Παράρτημα κώδικα προγραμματισμού	σελ.: 47
Κατάλογος εικόνων	σελ.: 61
Βιβλιογραφία	σελ.: 62
Ηλεκτρονική μορφή εργασίας	σελ.: 63

Περίληψη

Η παρούσα πτυχιακή εργασία αφορά την ανάπτυξη εφαρμογής, για κινητές συσκευές η οποίες χρησιμοποιούν για την λειτουργία τους το λογισμικό Android της Google. Οι χρήστες της εφαρμογής αυτής έχουν την δυνατότητα τον έλεγχο των λαμπτήρων led μέσω της διασύνδεση Bluetooth. Περιγράφεται λεπτομερώς η αρχιτεκτονική των εξαρτημάτων, η διαδικασία υλοποίησης της εφαρμογής, της συνδεσμολογίας των λαμπτήρων καθώς και ο προγραμματισμός αυτών.

Για να μπορέσουμε να αναλύσουμε το θέμα της εφαρμογής που θα χρησιμοποιήσουμε για τον έλεγχο του καταγραφικού, θα πρέπει να αναφερθούμε αρχικά στον ορισμό «λειτουργικό σύστημα». Στις κινητές συσκευές, όπως για παράδειγμα τα κινητά, τα tablets κλπ., έχουμε πέντε λειτουργικά συστήματα, το Android, το iOS, τα Windows, το RIM Blackberry και το Symbian OS. Εμείς, επιλέξαμε να αναλύσουμε τα δυο πιο δημοφιλή, αλλά και καλύτερα της αγοράς, το Android και το iOS. Αυτή η επιλογή έγινε σύμφωνα με άρθρα, αλλά και του παρακάτω στατιστικού πίνακα που παρουσιάζει τα καλύτερα κινητά σύμφωνα με το λειτουργικό τους.



Εικόνα 1

Λειτουργικό Σύστημα

Με τον όρο λειτουργικό σύστημα υπολογιστών ή λογισμικό (software), ορίζεται η συλλογή από προγράμματα και διαδικασίες που εκτελούν ορισμένες εργασίες σε ένα υπολογιστικό σύστημα.

Ο όρος περιλαμβάνει:

- Το λειτουργικό εφαρμογών, όπως οι επεξεργαστές κειμένου, που εκτελούν παραγωγικές εργασίες για τους χρήστες.
- Το λειτουργικό συστήματος, που παρέχει τις αναγκαίες υπηρεσίες του υλικού στο λειτουργικό εφαρμογών.
- Το ενδιάμεσο λειτουργικό, που ελέγχει και συντονίζει τα καταναμημένα συστήματα.
- Το υλικολειτουργικό που προγραμματίζει σε χαμηλό επίπεδο το υλικό ενός υπολογιστή ή τα περιφερειακά του.

Το λογισμικό περιλαμβάνει τα προγράμματα, τους ιστότοπους, τα βιντεοπαιχνίδια και άλλα προγράμματα που έχουν αναπτυχθεί σε μια γλώσσα προγραμματισμού, όπως για παράδειγμα η C, η C++ ή η Java. Το λογισμικό είναι κωδικοποιημένο με συγκεκριμένο τρόπο, με τη βοήθεια ενός δυαδικού ψηφιακού συστήματος, ώστε να είναι «κατανοητό» από το υλικό. Ο όρος «λογισμικό», χρησιμοποιείται μερικές φορές και σε ένα ευρύτερο πλαίσιο για να περιγράψει κάτι το οποίο δεν είναι υλικό, αλλά χρησιμοποιείται με το υλικό, όπως οι ταινίες, οι δίσκοι μουσικής και τα CD.

Ως λογισμικό θεωρείται ότι δεν ανήκει στο υλικό (hardware) του υπολογιστή ή άλλων συσκευών. Στο υλικό περιλαμβάνονται τα αντικείμενα που έχουν υλική υπόσταση, ενώ στο λογισμικό περιλαμβάνονται τα προγράμματα και οι εφαρμογές που υπάρχουν στο εσωτερικό των συσκευών. Το λογισμικό καλύπτει ένα ευρύτατο φάσμα προϊόντων και τεχνολογιών που αναπτύσσονται με χρήση διαφορετικών τεχνικών, όπως οι γλώσσες προγραμματισμού, οι γλώσσες μορφοποίησης κλπ. Οι διαφορετικοί τύποι λογισμικού, περιλαμβάνουν ιστοσελίδες που αναπτύσσονται με τις τεχνολογίες HTML, PHP, Perl, JSP, ASP.NET, XML και εφαρμογές που εμφανίζονται στην επιφάνεια εργασίας, όπως το Microsoft Word και το OpenOffice, που αναπτύσσονται στις γλώσσες C, C++, Java, C#, κλπ. Το λογισμικό εκτελείται συνήθως πάνω από ένα λειτουργικό σύστημα (που είναι επίσης λογισμικό), όπως τα Microsoft Windows, το Linux (με περιβάλλον GNOME ή KDE) ή το Solaris της Sun. Ως λογισμικό θεωρούνται και τα βιντεοπαιχνίδια όπως το Super Mario και το Grand Theft Auto για προσωπικούς υπολογιστές ή για κονσόλες βιντεοπαιχνιδιών. Αυτά τα παιχνίδια, αναπτύσσονται από εφαρμογές CGI (computer generated imagery) αφού έχουν σχεδιαστεί πρώτα τα γραφικά τους από εφαρμογές όπως το Maya, το 3ds Max ή το Blender. Επίσης, ένα λογισμικό συνήθως λειτουργεί πάνω σε μια πλατφόρμα λογισμικού που παρέχεται είτε από λειτουργικό σύστημα είτε από ανεξάρτητες πλατφόρμες, όπως η πλατφόρμα της Java και η πλατφόρμα της NET.

Λογισμικό γραμμένο για μία αρχιτεκτονική λογισμικού ή μία πλατφόρμα, συνήθως δεν μπορεί να λειτουργεί και σε άλλες πλατφόρμες. Για παράδειγμα, οι εφαρμογές των Microsoft Windows δεν θα είναι σε θέση να λειτουργήσουν σε Mac OS, λόγω των διαφορών που σχετίζονται με τις πλατφόρμες και τα πρότυπά τους. Οι εφαρμογές αυτές, μπορούν να λειτουργήσουν μόνο εάν μεταφερθούν, χρησιμοποιώντας ένα διερμηνέα ή με μεταφορά (port) του πηγαίου κώδικα στην εν λόγω πλατφόρμα.

[[11] <http://el.dbpedia.org/page/Λογισμικο> & [12] <http://el.wikipedia.org/wiki/Λογισμικο>]

To Android της Google

Το Android, είναι λειτουργικό σύστημα για συσκευές κινητής τηλεφωνίας, tablet, tv box κλπ., το οποίο, τρέχει τον πυρήνα του λειτουργικού Linux. Αρχικά, αναπτύχθηκε από τη Google και αργότερα από την Open Handset Alliance. Επιτρέπει στους κατασκευαστές λογισμικού να συνθέτουν κώδικα με την χρήση της γλώσσας προγραμματισμού Java, ελέγχοντας τη συσκευή μέσω βιβλιοθηκών λογισμικού ανεπτυγμένων από τη Google.

Η πρώτη παρουσίαση της πλατφόρμας Android, έγινε στις 5 Νοεμβρίου 2007, παράλληλα με την ανακοίνωση της ίδρυσης του οργανισμού Open Handset Alliance, μίας κοινοπραξίας σαράντα οχτώ (48) τηλεπικοινωνιακών εταιριών, εταιριών λογισμικού καθώς και κατασκευής hardware, οι οποίες είναι αφιερωμένες στην ανάπτυξη και εξέλιξη ανοιχτών προτύπων στις συσκευές κινητής τηλεφωνίας. Η Google δημοσίευσε το μεγαλύτερο μέρος του κώδικα του Android υπό τους όρους της Apache License, μίας ελεύθερης άδειας λογισμικού. Το λογότυπο για το λειτουργικό σύστημα Android, είναι ένα ρομπότ σε χρώμα πράσινου μήλου και σχεδιάστηκε από τη γραφίστρια Irina Blok.

Το Android είναι η πρώτη πλήρης, ανοικτή και δωρεάν διαθέσιμη πλατφόρμα. Οι προγραμματιστές έχουν ένα πλήρες «κιτ» ανάπτυξης λογισμικού, με άφθονα εργαλεία για ανάπτυξη ισχυρών και πλούσιων σε χαρακτηριστικά εφαρμογών. Η πλατφόρμα είναι ανοικτού πηγαίου κώδικα και βασίζεται σε δοκιμασμένα και σωστά πρότυπα ανοικτού πηγαίου κώδικα, με τα οποία οι προγραμματιστές είναι εξοικειωμένοι. Σημαντικό είναι, ότι δεν υπάρχουν ακριβοί φραγμοί εισόδου για τους προγραμματιστές και δεν απαιτούνται χρεώσεις για άδειες χρήσης. Οι προγραμματιστές του Android έχουν πολλές επιλογές για διανομή και εμπορική εκμετάλλευση των εφαρμογών τους.

Σε κάθε συσκευή με λογισμικό Android είναι προεγκατεστημένη η εφαρμογή Google Play Store. Αυτή επιτρέπει στον χρήστη να κατεβάζει εφαρμογές, όπως παιχνίδια, μουσική, ταινίες, βιβλία κλπ., είτε δωρεάν είτε επί πληρωμή. Σήμερα το Google Play Store θεωρείται το μεγαλύτερο κατάστημα εφαρμογών σε σχέση με τα υπόλοιπα τις ίδιας κατηγορίας.

[[A1]<http://pcmagazino.blogspot.gr/2016/03/android-1/html>]



Εικόνα 2

Στον παρακάτω πίνακα, παρουσιάζονται οι εκδόσεις του λογισμικού Android, η ημερομηνία έκδοσης κάθε μίας, καθώς και η χαρακτηριστική ονομασία της κάθε έκδοσης.

Έκδοση	Ονομασία	Ημερομηνία
7	Nougat	15 Σεπτεμβρίου 2016
6	Marshmallow	20 Οκτωβρίου 2015
5	Lollipop	3 Νοεμβρίου 2014
4,4	<i>KitKat</i>	31 Οκτωβρίου 2013
4,3	Jelly Bean	24 Ιουλίου 2013
4.2.X	<i>Jelly Bean</i>	13 Νοεμβρίου 2012
4.1.X	<i>Jelly Bean</i>	9 Ιουλίου 2012
4.0.X	<i>Ice Cream Sandwich</i>	16 Δεκεμβρίου 2011
3,2	<i>Honeycomb</i>	15 Ιουλίου 2011
2.3-2.3.7	<i>Gingerbread</i>	9 Φεβρουάριου 2011
2,2	<i>Froyo</i>	20 Μαΐου 2010
2.0-2.1	<i>Eclair</i>	26 Οκτωβρίου 2009
1,6	<i>Donut</i>	15 Σεπτεμβρίου 2009



ANDROID

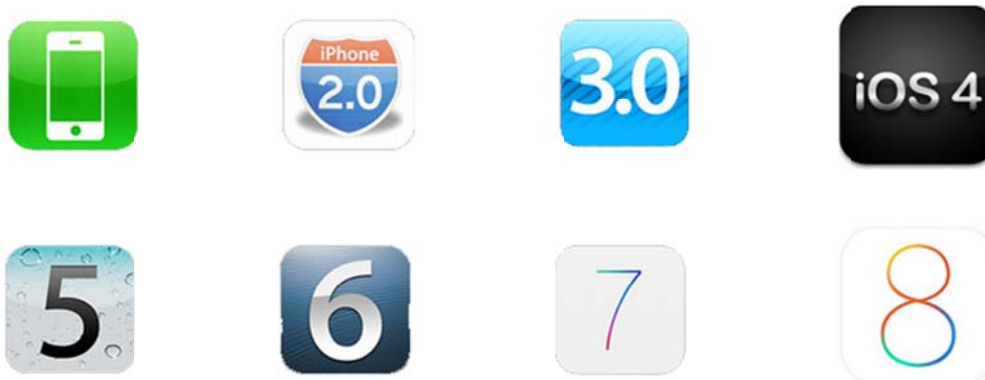
Εικόνα 3

Το iOS της Apple

Το λειτουργικό της Apple, το λεγόμενο iOS, είναι ένα λογισμικό για κινητά, το οποίο αναπτύχθηκε και διανέμεται από την Apple Inc. Αρχικά, παρουσιάστηκε το 2007 για το iPhone, ενώ υποστηρίζει και άλλες συσκευές της Apple όπως, το iPod touch (Σεπτέμβριος 2007), το iPad (Ιανουάριος 2010) και το Apple TV (δεύτερης γενιάς) (Σεπτέμβριος 2010). Η Apple δεν δίνει την άδεια για την εγκατάσταση του λογισμικού iOS σε συσκευές που δεν είναι κατασκευής Apple. Οι συσκευές της Apple, έχουν εγκατεστημένη την εφαρμογή App Store, με την οποία οι χρήστες μπορούν να εγκαταστήσουν στην συσκευή εφαρμογές αλλά και παιχνίδια.

Το iOS της Apple δεν διέθετε ένα επίσημο όνομα μέχρι την κυκλοφορία του iPhone στις 6 Μαρτίου 2008. Πριν από αυτό, η Apple απλά δήλωσε ότι το iPhone τρέχει μία έκδοση του OS X. Όταν καθιερώθηκε, ονομάστηκε iPhone OS, όπου μετονομάστηκε σε iOS στις 7 Ιουνίου 2010. Επειδή, το iPhone δεν ήταν πλέον η μόνη συσκευή που έτρεχε το iOS, η Apple καταχώρησε σαν ονομασία του λειτουργικού της, το «iOS».

Έκδοση	Ημερομηνία που κυκλοφόρησε
iPhone OS 1.x	6 Μαρτίου 2008
iPhone OS 2.x	11 Ιουλίου 2008
iPhone OS 3.x	17 Ιουνίου 2009
iOS 4.x	21 Ιουνίου 2010
iOS 5.x	6 Ιουνίου 2011
iOS 6.x	11 Ιουνίου 2012
iOS 7.x	10 Ιούνιος 2013
iOS 8.x	2 Ιουνίου 2014
iOS 9.x	15 Ιουνίου 2015
iOS 10.x	15 Σεπτέμβριος 2016



Εικόνα 4

Διαφορές των δύο λειτουργικών

Κατά την τελευταία δεκαετία, τα κινητά τηλέφωνα έχουν εξελιχθεί από μία απλή συσκευή που χρησιμοποιείται για επικοινωνία μέσω φωνητικών κλήσεων, σε ένα gadget που χειρίζεται σχεδόν οτιδήποτε μπορεί να κάνει κάποιος σε έναν υπολογιστή. Σερφάρισμα στο διαδίκτυο και παιχνίδια, είναι κάποιες από τις δραστηριότητες που μπορεί να πραγματοποιήσει κάποιος στις περισσότερες φορητή συσκευή σήμερα. Μετά από την ανάγκη των χρηστών των φορητών συσκευών να πραγματοποιούν όσο γίνεται περισσότερες διεργασίες, τα κινητά μετονομάστηκαν σε «έξυπνα κινητά» (Smartphone). Η λέξη «έξυπνο», χρησιμοποιήθηκε επειδή οι συσκευές αυτές ξεκίνησαν να κατασκευάζονται με σκοπό να καλύπτουν όλες τις ανάγκες του χρήστη, πέρα από τις απλές λειτουργίες, όπως είναι η ανταλλαγή μηνυμάτων και φωνητικών κλήσεων. Οι εταιρίες ξεκίνησαν με σκοπό της δημιουργία αυτών των «έξυπνων» συσκευών, όμως ταυτόχρονα ξεκίνησαν και την ανάπτυξη των λειτουργικών συστημάτων. Έτσι, η σύγκριση έρχεται σε δύο διαφορετικά λειτουργικά συστήματα κινητών. Το IOS της Apple που αφορά αποκλειστικά το iPhone, αλλά και άλλες φορητές συσκευές της Apple και το Android της Google, που είναι διαθέσιμο σε τηλέφωνα, tablet και tv box μέσα από μια ποικιλία κατασκευαστών. Αυτά τα δύο λειτουργικά συστήματα, καλύπτουν ένα μεγάλο ποσοστό της αγοράς των smartphone. Ανεξάρτητα από το γεγονός ότι τα δύο αυτά OS, χρησιμοποιούν τον ίδιο τύπο συσκευής, υπάρχουν σημαντικές διαφορές μεταξύ τους. Συγκεκριμένα, έχουν δημιουργηθεί από διαφορετικές εταιρίες και η σύγκριση τους δεν γίνεται μόνο ως προς την εμφάνιση αλλά και ως προς τις λειτουργίες τους.

✓ Σύγκριση βάση την εμφάνιση:

Οι συσκευές της Apple είναι πιο απλές στη λειτουργία τους και αυτό τις κάνει πιο προσιτές σε χρήστες, οι οποίοι δεν είναι τόσο εξοικειωμένοι με την τεχνολογία. Από την άλλη, οι συσκευές με το λειτουργικό Android, μπορούν να φανούν πιο δύσκολες ως προς την λειτουργία τους, αλλά δεν γίνεται να μη λάβουμε υπόψη, τις ρυθμίσεις που παρέχουν, έτσι ώστε ο κάτοχος της συσκευής να εξατομικεύσει το περιβάλλον στη μορφή που του αρέσει, αλλά και τον διευκολύνει. Αυτό, δεν μπορεί να πραγματοποιηθεί στις άλλες συσκευές. Στις παρακάτω εικόνες, παρουσιάζεται η αρχική οθόνη σε κινητά με λειτουργικό Android αλλά και με iOS. Η Apple, όσον αφορά τη λήψη εφαρμογών, γνωστά ως «apps», έχει αυστηρότερες κατευθυντήριες γραμμές τόσο για τα προγράμματα, όσο και για τους προγραμματιστές.

Το Android επιτρέπει μεγαλύτερη ελευθερία στη λειτουργία του, σε αντίθεση με το iOS, διότι είναι multitasking. Αυτή η ελευθερία, προχωρά στο μέτρο όπου επιτρέπει στον χρήστη ενός Android να εγκαταστήσει μια νέα εφαρμογή, ενώ κάποια άλλη να «τρέχει» στο παρασκήνιο. Αυτό, δεν είναι δυνατόν με το iOS καθώς ελέγχονται αυστηρά οι δραστηριότητες του.



Εικόνα 5

- ✓ Ένα σημείο το οποίο χρήζει ιδιαίτερης σημασίας για τη σύγκριση των δύο λειτουργικών και ενδιαφέρει περισσότερο τους προγραμματιστές και κατ' επέκταση και εμάς που θα δημιουργήσουμε μία εφαρμογή, είναι ότι το Android αναπτύχθηκε ως μέρος του Open Handset Alliance, μία συνεργασία από ογδόντα τέσσερις (84) εταιρείες ηλεκτρονικών, με σκοπό τη δημιουργία ενός κινητού με λειτουργικό σύστημα σταθερών κανόνων. Το Android πιστώνεται στη Google, δεδομένου ότι χρηματοδότησε την ανάπτυξη του λειτουργικού συστήματος των Smartphones, που αγόρασαν τα δικαιώματα το 2005. Το Android, είναι ένα λειτουργικό σύστημα ανοικτού κώδικα (Open source), δηλαδή ο κώδικας διατίθεται σε όποιον θέλει να το δει και να το χρησιμοποιήσει. Πολλοί κατασκευαστές μπορούν να εργαστούν με την άποψη του σχεδιασμού και της λειτουργικότητας. Αυτός είναι και ο λόγος που βλέπουμε μία πληθώρα επιλογών από εταιρείες όπως η LG, η HTC και η Samsung. Κάθε κατασκευαστής, μπορεί να ερμηνεύσει το Android με το δικό του τρόπο και στη συνέχεια να φέρει αυτό το σχέδιο και τη λειτουργικότητα στα τηλέφωνα του, έχοντας την ελευθερία να αλλάξει το λειτουργικό τους σύστημα. Για παράδειγμα, το Amazon έχει το Kindle. Επίσης, επιτρέπει ερασιτέχνες ή επαγγελματίες συγγραφείς κώδικα, να παράγουν εφαρμογές που μπορούν να πωλούν ή να προσφέρουν δωρεάν στους χρήστες του Android. Ένα μεγάλο μέρος από τις εφαρμογές που γίνονται για το Android αναπτύσσονται από ιδιώτες προγραμματιστές που δεν έχουν ως στόχο να δημιουργήσουν κέρδη για τους ίδιους. Η Apple (iPhone και iPad προϊόντα χρήσης iOS) λειτουργεί σε ένα κλειστό περιβάλλον, όπου κανείς δεν πωλεί συσκευές που χρησιμοποιούν το συγκεκριμένο λειτουργικό εκτός από την ίδια την εταιρία. Αυτό έχει ως αποτέλεσμα, τα προϊόντα της να σχεδιάζονται έτσι ώστε ο χρήστης να μην έχει έλεγχο των λειτουργικών επιπέδων του συστήματος.
- ✓ Ένα ακόμη σημαντικό σημείο σε αυτήν τη σύγκριση, είναι η διασύνδεση Bluetooth. Εδώ η Apple, περιορίζει σε μεγάλο βαθμό τους χρήστες της, διότι η μεταφορά αρχείων (π.χ. επαφών, φωτογραφιών και άλλων δεδομένων), δεν μπορεί να πραγματοποιηθεί χωρίς την χρήση κάποιας εφαρμογής με συσκευή άλλης εταιρίας παρά μόνο με συσκευές της Apple. Ενώ με την χρήση λειτουργικού Android ο κάτοχος δεν περιορίζεται σε αυτό τον τομέα. Συνήθως οι χρήστες δεν δυσαρεστούνται, διότι το πρόβλημα αυτό μπορεί να λυθεί με κάποια εφαρμογή από το App Store της Apple. Όμως, για τους προγραμματιστές αλλά και για τους μηχανικούς όπου κατασκευάζουν κάποια συσκευή, η οποία θα ελέγχεται μέσω κινητής συσκευής και συνδέεται με αυτή μέσω Bluetooth, όπως στη συγκεκριμένη περίπτωση, δημιουργεί προβλήματα, αλλά και αυξάνει το κόστος της κατασκευής, αφού θα πρέπει ο μηχανικός να προμηθευτεί το κατάλληλο Bluetooth Module, το οποίο βοηθάει στη σύνδεση με Bluetooth των συσκευών της Apple με την κατασκευή. [\[A3\]](#)



Εφαρμογές κινητών συσκευών

Με την κυκλοφορία των κινητών συσκευών παρουσιάστηκε μεγάλη ανάπτυξη στον τομέα προγραμματισμού εφαρμογών για τις συγκεκριμένες συσκευές. Σε όλες τις συσκευές υπάρχει εγκατεστημένη εφαρμογή η οποία μπορεί να θεωρηθεί ένα ηλεκτρονικό κατάστημα για εφαρμογές. Η ονομασία αυτής της εφαρμογής ποικίλει από λειτουργικό σε λειτουργικό, για παράδειγμα στο Android ονομάζεται Play Store, ενώ στο iOS App Store. Σε αυτήν την εφαρμογή ο κάτοχος της συσκευής μπορεί να προμηθευτεί εφαρμογές της αρεσκείας του, οι οποίες μπορεί να είναι παιχνίδια, εφαρμογές για την ενημέρωσή του, ψυχαγωγικές όπως βιβλία, αλλά και πολλές χρήσιμες εφαρμογές γραφείου ή και για την καθημερινή του ευκολία στο σπίτι ή στο αυτοκίνητο. Μεγάλη ανάπτυξη έχει παρουσιαστεί στην δημιουργία εφαρμογών που παρέχουν τον έλεγχο συσκευών από απόσταση με την χρήση είτε μέσω δικτύου (wi-fi) είτε με κάποια άλλη μορφή διασύνδεσης όπως είναι το Bluetooth και το NFC. Όπως έχουμε προαναφέρει η Apple περιορίζει τους χρήστες της και με τις εφαρμογές, επειδή ο μόνος τρόπος για να προμηθευτεί κάποιος μια εφαρμογή σε συσκευές οι οποίες έχουν λειτουργικό iOS γίνεται αποκλειστικά μέσω του App Store, δηλαδή το ηλεκτρονικό κατάστημα της Apple. Σε αντίθεση με τις Android συσκευές που μπορεί κανείς να κατεβάσει εφαρμογές από διάφορα διαδικτυακά καταστήματα εφαρμογών. Χωρίς αυτό να αποκλείει την χρήση του Play Store της Google. Επειδή οι διαθέσιμες εφαρμογές είναι πάρα πολλές, συχνά κατατάσσονται σε κατηγορίες μέσα στο ηλεκτρονικό κατάστημα, σε τομείς όπως η Ψυχαγωγία, τα Παιχνίδια, η Μουσική, οι Αγορές, η Κοινωνική Δικτύωση, ο Αθλητισμός και τα Ταξίδια. Το κόστος ποικίλλει ανάλογα με την εφαρμογή, κάποιες εφαρμογές μπορεί να είναι δωρεάν, ενώ για άλλες θα πρέπει ο χρήστης να πληρώσει. Οι μέθοδοι πληρωμής διαφέρουν μεταξύ καταστημάτων εφαρμογών και ίσως περιλαμβάνουν τις πιστωτικές/χρεωστικές κάρτες, τους λογαριασμούς στο PayPal και τους λογαριασμούς στο iTunes. Σε ορισμένες εφαρμογές υπάρχει περίπτωση να υπάρχουν χρεώσεις εντός της εφαρμογής, για διάφορες υπηρεσίες. Πολλές εφαρμογές για την λειτουργία τους, χρειάζονται σύνδεση στο ίντερνέτ. Σε αυτές τις περιπτώσεις όσο η εφαρμογή τρέχει, γίνεται χρέωση από τον πάροχο σύμφωνα με τα πακέτα δεδομένων που ο κάθε χρήστης έχει συμφωνήσει ή γίνεται χρήση του ίντερνέτ μέσω ασύρματου δικτύου wi-fi.



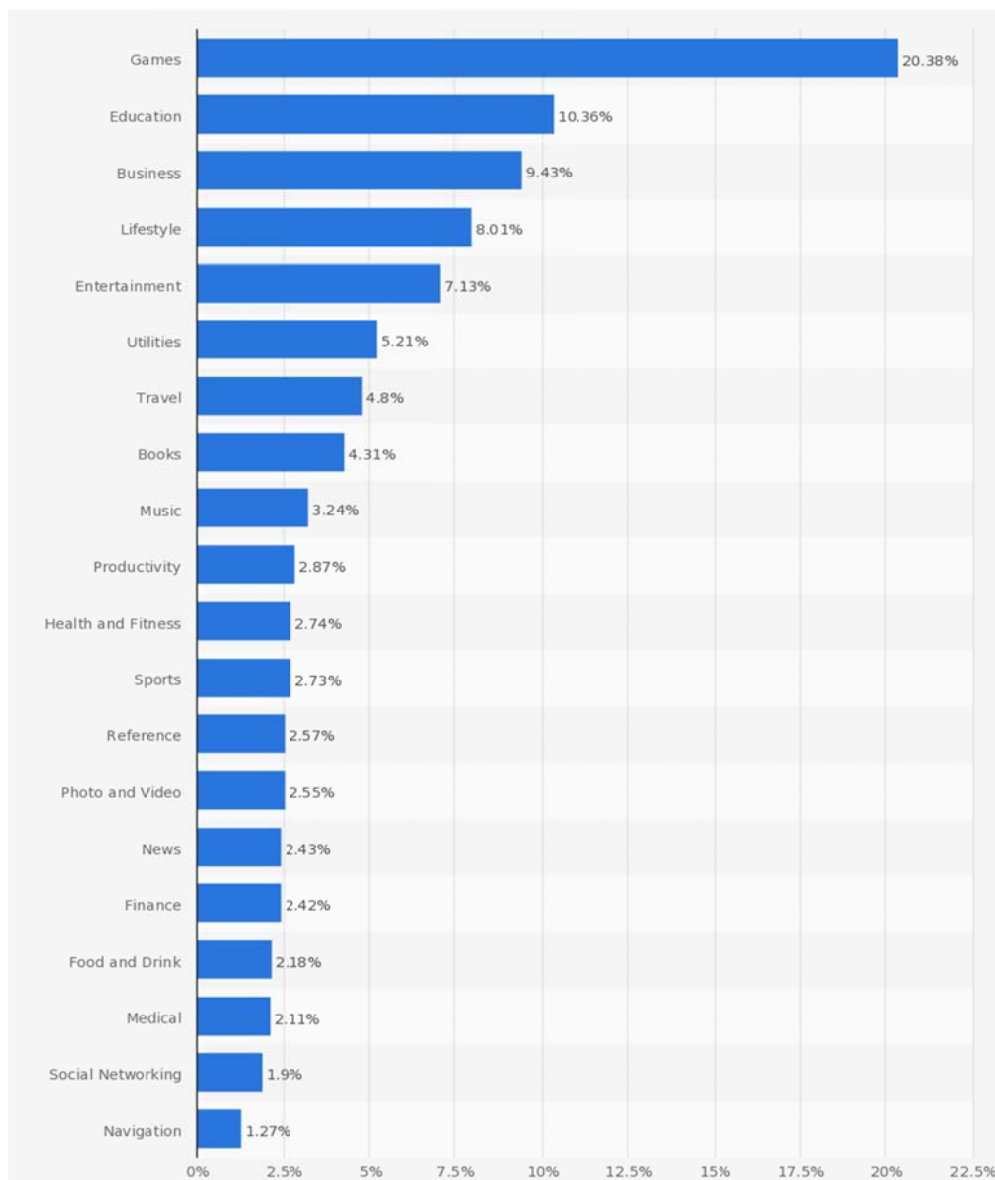
App Store



Google play

Εικόνα 6

Στο παρακάτω γράφημα παρουσιάζεται η ζήτηση των εφαρμογών ανά κατηγορία το 2015 Στα δυο ηλεκτρονικά καταστήματα της Apple και της Google (App Store, Play Store). Διακρίνουμε μία μεγάλη ζήτηση στα παιχνίδια. Οι εφαρμογές που έχουν σχέση με τον απομακρυσμένο έλεγχο συσκευών μέσα από κινητές συσκευές, βρίσκονται στην κατηγορία Business και Utilities, που σύμφωνα με το διάγραμμα είναι στην 3^η και 6^η θέση αντίστοιχα.



Εικόνα 7

Bluetooth

Το Bluetooth είναι μια ασύρματη τεχνολογία για τη μεταφορά δεδομένων σε μικρές αποστάσεις. Με τη βοήθεια του Bluetooth, ένας χρήστης μπορεί να συνδέσει διάφορες κινητές ή σταθερές συσκευές του. Για παράδειγμα, οι συσκευές αυτές μπορεί να είναι συσκευές εισόδου (Human Interface Device) για υπολογιστή, όπως ποντίκια και πληκτρολόγια, mp3 players για ασύρματη σύνδεση με ηχεία και αναπαραγωγή μουσικής, κινητά τηλέφωνα για μεταφορά δεδομένων (π.χ. επαφών) μεταξύ τους ή σύνδεση με κάποιο περιφερειακό ακουστικό (headset, hands-free). Η τελευταία περίπτωση χρήσης μπορεί να θεωρηθεί ως ένας από τους βασικότερους λόγους που το Bluetooth είναι δημοφιλές. Βρίσκεται σε ευρεία χρήση ακόμη και σήμερα, παρότι άλλες περιπτώσεις χρήσης, όπως η μεταφορά δεδομένων μεταξύ κινητών, δε χρησιμοποιούνται, ή ακόμα και δεν υποστηρίζονται πλέον, όσο παλαιότερα. Όπως θα δούμε παρακάτω, με την έλευση του Bluetooth Low Energy, η γκάμα των συσκευών που μπορούν να περιλαμβάνονται στο PAN του χρήστη, εμπλουτίζεται με μια ολοένα και αυξανόμενη ποικιλία διαφορετικών προϊόντων.

Η προέλευση του ονόματος του Bluetooth, που άρχισε να χρησιμοποιείται επισήμως το 1998, αντικατοπτρίζει ακριβώς τη δυνατότητα, που παρέχει η τεχνολογία αυτή στον χρήστη, να συνδέει με ευκολία τις διάφορες συσκευές του. Καθώς και την ενοποίηση των πολλών διαφορετικών προτύπων επικοινωνίας σε ένα. Το όνομα προέρχεται από απόδοση στα αγγλικά του επιθέτου ενός Δανού βασιλιά του 10ου αιώνα, ο οποίος κατάφερε να ενώσει τις διάφορες δανικές φυλές, που ήταν ως τότε διαχωρισμένες, σε ένα ενιαίο βασίλειο. Το όνομα του βασιλιά αυτού ήταν Harald Bluetooth. Επίσης το επίσημο logo του Bluetooth, προέρχεται από τη συνένωση των γραμμάτων του ρουνικού αλφάβητου, που αποτελούν τα αρχικά του Harald Bluetooth.

Το Bluetooth αναπτύχθηκε αρχικά, από τους μηχανικούς της εταιρίας προϊόντων τηλεπικοινωνίας Ericsson, το 1994 σε μια προσπάθειά τους να απαλλαγούν από τη χρήση των σειριακών καλωδίων RS-232, που χρησιμοποιούνταν ευρέως εκείνη την εποχή. Το 1998, η Ericsson μαζί με 4 άλλες εταιρίες (Intel, IBM, Nokia, Toshiba) δημιουργούν μια εμπορική ένωση με το όνομα Bluetooth SIG (Special Interest Group), με σκοπό την εξέλιξη και προώθηση της νέας τεχνολογίας. Μέχρι το τέλος του έτους το Bluetooth SIG απαρτιζόταν ήδη πάνω από 400 μέλη, ενώ σήμερα αποτελείται από περισσότερα από 20,000 μέλη.

Το SIG επιβλέπει την εξέλιξη της τεχνολογίας και των προτύπων του Bluetooth. Επίσης ελέγχει τη χρήση της τεχνολογίας και των εμπορικών σημάτων από τους διάφορους κατασκευαστές συσκευών, μέσω ειδικών τεστ τα οποία πρέπει να περάσουν οι συσκευές πριν βγουν στην κυκλοφορία, προκειμένου να θεωρηθούν συμβατές με το πρότυπο και να μπορούν να χρησιμοποιήσουν το λογότυπο του Bluetooth. Κατά καιρούς διοργανώνει επίσης, τα επονομαζόμενα Unplug Fest, στα οποία μηχανικοί από εταιρίες μέλη του SIG μπορούν να συναντηθούν και να ανταλλάξουν ιδέες σχετικά με την τεχνολογία, ενώ παράλληλα πραγματοποιούν τεστ, στα οποία πολλές παλιές και νέες συσκευές Bluetooth ενεργοποιούνται ταυτόχρονα στον ίδιο χώρο, προκειμένου να ελεγχθεί η δυνατότητα interoperability υπό δύσκολες συνθήκες, κάτι που αποτελεί ένα από τα χαρακτηριστικά προώθησης της συγκεκριμένης τεχνολογίας.

Το 1999 βγαίνει η πρώτη έκδοση του Bluetooth specification (1.0), ενώ το ενδιαφέρον για τη νέα αυτή τεχνολογία αυξάνεται. Την επόμενη χρονιά εμφανίζονται στην αγορά τα πρώτα προϊόντα που χρησιμοποιούν το Bluetooth για ασύρματη επικοινωνία, όπως το πρώτο κινητό τηλέφωνο με υποστήριξη Bluetooth καθώς και το πρώτο Bluetooth ακουστικό. Από εκεί και πέρα, η τεχνολογία γίνεται ολοένα και πιο δημοφιλής και τα προϊόντα που τη χρησιμοποιούν αυξάνονται με πολύ γρήγορους ρυθμούς, για να φτάσουμε στο 2005 να έχουμε πάνω από 5 εκατομμύρια Bluetooth chipsets να εισέρχονται στην αγορά κάθε εβδομάδα, και το 2008 να υπάρχουν σχεδόν 2 δισεκατομμύρια προϊόντα με υποστήριξη του πρωτοκόλλου. Εν τω μεταξύ, το 2004, βγαίνει η έκδοση 2.0 του Bluetooth, η οποία περιλαμβάνει ως βασικό νέο χαρακτηριστικό το Enhanced Data Rate (EDR), το οποίο όπως λέει και το όνομά του, παρέχει υψηλότερη ταχύτητα από την προηγούμενη έκδοση, αναφερόμενη από εκεί και πέρα ως Basic Rate (BR).

Στα μέσα του 2007 βγαίνει η έκδοση 2.1. βασικό χαρακτηριστικό της οποίας ήταν το Secure Simple Pairing (SSP), το οποίο απλοποιεί σε μεγάλο βαθμό τη διαδικασία ασφαλούς σύνδεσης (pairing ή bonding) μεταξύ δύο συσκευών, αυξάνοντας παράλληλα το παρεχόμενο επίπεδο ασφάλειας.

Το 2009 έχουμε την έκδοση 3.0, με βασικό χαρακτηριστικό το Alternate MAC PHY (Media Access Control / Physical Layer), το οποίο, χρησιμοποιώντας ένα διαφορετικό μέσο μεταφοράς στο χαμηλότερο επίπεδο (το specification ορίζει το S2.1 MAC PHY), μπορεί να επιτύχει ακόμα μεγαλύτερες ταχύτητες μεταφοράς δεδομένων.

Έτσι, φτάνουμε πλέον στο 2010, οπότε με την έκδοση 4.0 του Bluetooth, εμφανίζεται μια νέα τεχνολογία με το όνομα Bluetooth Low Energy (BLE). Από εδώ και πέρα, η παλαιότερη έκδοση του Bluetooth αναφέρεται ως Bluetooth Classic ή Bluetooth BREDR..

Στην διπλωματική μου εργασία θα χρησιμοποιήσουμε και το BLE αλλά και το Bluetooth Classic, επειδή πρέπει η εφαρμογή μας να μπορεί να χρησιμοποιηθεί και από όλες τις συσκευές ακόμα και από αυτές που δεν υποστηρίζουν την διασύνδεση BLE.

Παρότι έχουν το ίδιο όνομα και προέρχονται από τον ίδιο οργανισμό, τα Bluetooth Classic και Bluetooth Low Energy, ενώ έχουν κάποιες ομοιότητες σε ορισμένα σημεία τους, εμφανίζουν μεγάλες διαφορές σε άλλα. Αυτό είναι λογικό αν σκεφτεί κανείς ότι δημιουργήθηκαν για να εξυπηρετήσουν διαφορετικούς σκοπούς. Ακολουθούν, επιγραμματικά τα βασικά χαρακτηριστικά του Bluetooth Classic. Κάποια από αυτά υπάρχουν και στο BLE, όπου χρησιμοποιούνται με διαφορετικό, συνήθως απλούστερο τρόπο. Αυτή είναι και μια βασική διαφορά μεταξύ των δύο τεχνολογιών. Το Bluetooth Classic είναι ένα αρκετά πολύπλοκο πρωτόκολλο, το BLE αντίθετα κατασκευάστηκε εξαρχής και με γνώμονα την απλότητα των διαδικασιών και των πρωτοκόλλων. [\[\[A2\] www.myphone.gr/library/article-39.html\]](http://www.myphone.gr/library/article-39.html)

Εφαρμογές Bluetooth

Οι εφαρμογές του Bluetooth είναι πολλαπλές και καλύπτουν μια μεγάλη γκάμα ενεργειών.

- Ασύρματη δικτύωση μεταξύ επιτραπέζιου και φορητού υπολογιστή, σε έναν περιορισμένο χώρο με ελάχιστο διαθέσιμο εύρος ζώνης.
- Ασύρματα περιφερειακά, όπως εκτυπωτές, ποντίκια και πληκτρολόγια, τα οποία επικοινωνούν με κάποιον υπολογιστή ή φορητή συσκευή.
- Ασύρματη μεταφορά ψηφιακών αρχείων, για παράδειγμα εικόνων, ανάμεσα σε κινητά τηλέφωνα και PDA.
- Ορισμένοι δέκτες GPS μεταφέρουν πληροφορίες μέσω Bluetooth.
- Το Bluetooth δίνει τη δυνατότητα σε χρήστες καταλλήλως εξοπλισμένων κινητών τηλεφώνων να χρησιμοποιούν κάποιες βασικές λειτουργίες τους με ασύρματα ακουστικά. Ανάλογο σύστημα υπάρχει ενσωματωμένο και σε κράνη οδηγών μοτοσυκλέτας, επιτρέποντας τη συνομιλία κατά την οδήγηση.
- Απομακρυσμένος έλεγχος συσκευών, όπου έως την εμφάνιση του Bluetooth χρησιμοποιούνταν τεχνολογία υπέρυθρων ακτίνων. [\[\[A2\] www.myphone.gr/library/article-39.html\]](http://www.myphone.gr/library/article-39.html)



Εικόνα 8

Λειτουργία και Δομή Bluetooth

Οι προδιαγραφές του Bluetooth καθορίζουν την «ασύρματη» τεχνολογία χαμηλού κόστους και χαμηλής ισχύος, που εξαλείφει τα καλώδια μεταξύ των κινητών συσκευών και επιτρέπει τη διασύνδεσή τους. Το Bluetooth λειτουργεί στο «αδέσμευτο» φάσμα συχνοτήτων των 2,4 GHz, ώστε οι συσκευές που το ενσωματώνουν να μπορούν να λειτουργήσουν απροβλημάτιστα σε οποιοδήποτε σημείο του πλανήτη. Για να περιοριστούν στο ελάχιστο οι παρεμβολές από παρεμφερείς συσκευές, το Bluetooth εκμεταλλεύεται την αμφίδρομη επικοινωνία και τη μέθοδο μετάδοσης με διασπορά φάσματος Frequency Hopping (έως και 1600 εναλλαγές συχνότητας ανά δευτερόλεπτο). Από φυσική άποψη επίσης το Bluetooth λειτουργεί περίπου στα 2.4 GHz, προδιαγράφει τρία επίπεδα ισχύος της εκπομπής από τα οποία εξαρτάται και η εμβέλεια επικοινωνίας (πάντα μικρότερη των 10 μέτρων σε PAN), ενώ η τακτική αλλαγή της συχνότητας εκπομπής λόγω της αξιοποίησης του FHSS καθορίζεται τυχαία από έναν κεντρικό κόμβο, τον Master.

Η βασική δομική μονάδα ενός δικτύου Bluetooth είναι το piconet, στο οποίο όλοι οι κόμβοι που μετέχουν (μέχρι 7 συσκευές Slaves) μοιράζονται τον ίδιο κώδικα διασποράς και υπόκεινται στον έλεγχο ενός κοινού Master. Ο τελευταίος διαμοιράζει στους σταθμούς Slaves την πρόσβαση στο κοινό μέσο (τον ελεύθερο χώρο) με τη μέθοδο TDMA/TDD, όπου ο χρόνος διαμερίζεται σε αυστηρές χρονοθυρίδες, ο Master εκπέμπει στις περιττές και οι Slaves στις άρτιες (εναλλάξ), κάθε κόμβος που θέλει να εκπέμπει λαμβάνει περιοδικά από τον Master το δικαίωμα μετάδοσης σε 1, 3 ή 5 συνεχόμενες χρονοθυρίδες και κατά τη διάρκεια εκπομπής ενός πλαισίου δεν γίνεται εναλλαγή συχνότητας. Τα τερματικά μεταδίδουν μόνο στον Master, ο οποίος αποστέλλει στη συνέχεια τα πλαίσιά τους προς τον τελικό παραλήπτη, και διακρίνονται από μία παγκόσμια μοναδική 48-bit διεύθυνση. Δύο ή περισσότερα piconet μπορούν να βρίσκονται στον ίδιο χώρο, με τους κόμβους να μπορούν να συμμετέχουν σε παραπάνω από ένα ταυτόχρονα, και να επικοινωνούν μεταξύ τους δημιουργώντας ένα μεγαλύτερης κλίμακας scatternet. Υπάρχουν δύο τύποι συνδέσεων:

- 1) Σύγχρονες ή SCO. Επιτρέπουν τη διέλευση χρονικά κρίσιμων πληροφοριών (συνήθως φωνής), κάθε κόμβος μπορεί να δεσμεύσει μόνο μέχρι μία χρονοθυρίδα, έχουν ρυθμό μετάδοσης δεδομένων 64 kbps, υλοποιούν συνδεσμοστρεφή επικοινωνία αυστηρά από σημείο σε σημείο, χρησιμοποιούν αλγορίθμους ανίχνευσης και διόρθωσης σφαλμάτων (FEC), ενώ δεν υπάρχουν επανεκπομπές ή επιβεβαιώσεις.
- 2) Ασύγχρονες ή ACL. Τυπικά χρησιμοποιούνται για τη μετάδοση δεδομένων, κάθε κόμβος μπορεί να δεσμεύσει 1, 3 ή 5 χρονοθυρίδες για την εκπομπή ενός πλαισίου, είναι ασυνδεσμικές με έλεγχο ροής, έλεγχο σφαλμάτων (με αριθμούς ακολουθίας 1-bit και θετικές/αρνητικές επιβεβαιώσεις) και δυνατότητα πολυδιανομής, ενώ ο ρυθμός μετάδοσης μπορεί να ανέβει ως τα 724 kbps. [\[\[A2\] www.myphone.gr/library/article-39.html\]](http://www.myphone.gr/library/article-39.html)



Bluetooth 4 – Bluetooth Low Energy (BLE)

Το Bluetooth Low Energy (BLE) είναι ένα ασύρματο δίκτυο που έχει σχεδιαστεί και διατίθεται στο εμπόριο από την Bluetooth Special Interest Group με στόχο τη καινοτόμο εφαρμογή του στην υγειονομική περίθαλψη, στα γυμναστήρια, στην ασφάλεια, στις βιομηχανίες αλλά και στην οικιακή ψυχαγωγία. Σε σύγκριση με το κλασικό Bluetooth, το BLE προορίζεται για να παρέχει σημαντικά μειωμένη κατανάλωση ενέργειας και κόστος, διατηρώντας παράλληλα ένα παρόμοιο εύρος επικοινωνίας.

Το BLE εισήχθη αρχικά με την επωνυμία Wibree από τη Nokia το 2006. Αργότερα συγχωνεύθηκε στο κύριο πρότυπο Bluetooth το 2010 με την έκδοση του Bluetooth Core Προδιαγραφές έκδοσης 4.0.

Τα λειτουργικά συστήματα, συμπεριλαμβανομένων iOS, Android, Windows Phone και BlackBerry, καθώς και OS X, Linux και Windows 8, υποστηρίζουν την τεχνολογία Bluetooth Low Energy. Το Bluetooth SIG προβλέπει ότι το 2018 πάνω από το 90% των κινητών συσκευών, που πληρούν τις προδιαγραφές για να υποστηρίζουν Bluetooth, θα παρέχουν στον χρήστη την τεχνολογία BLE.

Παρόλο που το Bluetooth 4 χρησιμοποιεί και αυτό όπως και το Bluetooth Classic την ραδιοσυχνότητα 2,4 GHz δεν είναι συμβατά μεταξύ τους.

Ο στόχος της αναβάθμισης της συγκεκριμένης τεχνολογίας βοηθάει πολύ σε πολλούς τομείς, όπως για παράδειγμα στα έξυπνα σπίτια, στην υγεία, στον αθλητισμό κλπ. Αυτό επιτυγχάνεται γιατί με την τεχνολογία αυτή παρατηρούνται πολλά πλεονεκτήματα. Κάποια από τα οποία είναι η χαμηλή απαίτηση ισχύος, που επιτρέπει στις συσκευές να λειτουργούν για πολύ καιρό. Επίσης δεν απαιτείται πολύς χώρος. Λόγω του μικρού μεγέθους κατορθώνεται μύωση του κόστους. Τέλος υπάρχει συμβατότητα με πολλές συσκευές κινητών, tablets και υπολογιστές.

Η πρώτη συσκευή που κυκλοφόρησε με την τεχνολογία BLE ήταν το iPhone 4s της Apple, που κυκλοφόρησε τον Οκτώβριο του 2011. [\[\[A2\] www.myphone.gr/library/article-39.html\]](http://www.myphone.gr/library/article-39.html)

Στον παρακάτω πίνακα παρουσιάζονται συνοπτικά οι βασικές διαφορές των δύο τεχνολογιών, Bluetooth Classic και Bluetooth Low Energy.



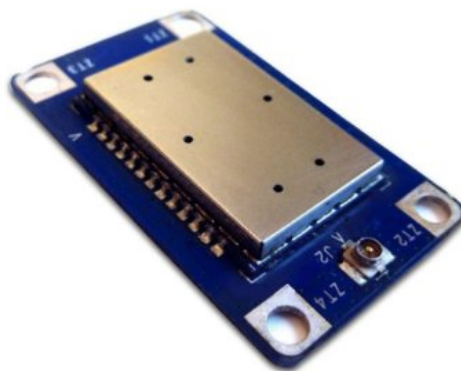
Εικόνα 9

Τεχνικές Προδιαγραφές	Bluetooth Classic	Bluetooth Low Energy
Απόσταση / Περιοχή	100 m	> 100 m
Ρυθμός δεδομένων	1-3 Mbit / s	1 Mbit / s
Απόδοση	0,7 έως 2,1 Mbit / s	0,27 Mbit / s
Ασφάλεια	56/128-bit	128-bit AES με Counter λειτουργία CBC-MAC
Ελάχιστος συνολικός χρόνος για την αποστολή δεδομένων	100 ms	3 ms
Κατανάλωση ρεύματος	1 W ως αναφορά	0,01 έως 0,5 W (ανάλογα με τη χρήση)
Peak κατανάλωση ρεύματος	<30 mA	<15 mA
Περιπτώσεις χρήσης	Κινητά τηλέφωνα, τυχερά παιχνίδια, ακουστικά, έξυπνα σπίτια, αυτοκίνητα, υπολογιστές, συστήματα ασφαλείας, στην υγειονομική περίθαλψη, τον αθλητισμό για την φυσική κατάσταση, κ.λπ.	Κινητά τηλέφωνα, παιχνίδια, υπολογιστές, ρολόγια, αθλητισμός και φυσική κατάσταση, την υγεία, συστήματα ασφαλείας, αυτοκίνητα, οικιακές συσκευές, συσκευές αυτοματισμού, Βιομηχανία, κ.λπ.

Bluetooth σε Apple συσκευές και Bluetooth Module

Όπως έχει προαναφερθεί οι συσκευές της Apple διαφέρουν από τις συσκευές που έχουν σαν λειτουργικό το Android, ως προς την διασύνδεση με Bluetooth. Αρχικά η Apple δεν επιτρέπει την μεταφορά αρχείων μεταξύ κινητών. Αλλά το πιο βασικό κομμάτι του βρίσκεται στις συσκευές με τις οποίες συνεργάζεται. Δηλαδή επειδή το Bluetooth στις συσκευές με iOS είναι κωδικοποιημένο θα πρέπει να χρησιμοποιηθεί για την κατασκευή μια ειδική συσκευής η οποία να συνδέεται με τα προϊόντα της Apple ένα εξάρτημα το οποίο να περιέχει τα πρωτόκολλα για την σύνδεση των συσκευών με λειτουργικό iOS. Αυτό το εξάρτημα ονομάζεται Bluetooth Module. Το συγκεκριμένο εξάρτημα μπορούμε να το προμηθευτούμε από το ηλεκτρονικό κατάστημα της Apple και κοστίζει περίπου 10\$.

Από την άλλη όλες οι συσκευές της Apple από το 2011 λειτουργούν με Bluetooth 4 (BLE) κάτι το οποίο στις συσκευές με λειτουργικό Android ξεκίνησε στα μοντέλα που δημιουργήθηκαν από το 2014 και έπειτα.



Εικόνα 10

NFC

Η επικοινωνία κοντινού πεδίου (near field communication, NFC) αποτελεί μια πρότυπη τεχνολογία συνδεσιμότητας, η οποία διαδίδεται και εξελίσσεται ραγδαία με κύριο σκοπό τη λύση αρκετών προβλημάτων, σύγχρονων αλλά και μελλοντικών. Είναι μια μικρής εμβέλειας ασύρματη τεχνολογία, η οποία λειτουργεί στη συχνότητα των 13,56 MHz και μεταφέρει δεδομένα με ρυθμό έως και 424 kbps και έχει γίνει γνωστή κυρίως μέσω της χρήσης της από τα κινητά τηλέφωνα τελευταίας γενιάς. Η λειτουργία της βασίζεται στην επαφή ή στην προσέγγιση, σε απόσταση περίπου τεσσάρων με πέντε εκατοστών, της συσκευής που περιέχει το τσιπ NFC, σε κάποια άλλη συσκευή που περιλαμβάνει τον κατάλληλο αισθητήρα.

Η τεχνολογία αυτή δημιουργήθηκε το 2004 μετά από συνεργασία της Nokia, της Philips και της Sony, με την εξάπλωση της να είναι μεγαλύτερη των προσδοκιών και να μελετάται συνεχώς για περισσότερους τομείς της καθημερινότητας, όπως για παράδειγμα η γρήγορη ανάγνωση και εγγραφή δεδομένων, η χρήση εικονικών πιστωτικών καρτών, η πιστοποίηση οντοτήτων αλλά και η ενεργοποίηση υπηρεσιών. Η τεχνολογία NFC συνδυάζει παλιότερες τεχνολογίες ασύρματης επικοινωνίας όπως το Bluetooth και η RFID, οι οποίες εναρμονίζονται ώστε να παρέχονται υπηρεσίες στους χρήστες στις παρακάτω ενδεικτικές περιπτώσεις:

- Έλεγχος πρόσβασης
- Ηλεκτρονικές συναλλαγές
- Ανταλλαγή και συλλογή πληροφοριών
- Πληρωμές
- Μεταφορές/Διαβιβάσεις

Η τεχνολογία της επικοινωνίας κοντινού πεδίου πληροί τις προδιαγραφές των standard ISO/IEC 14443 A&B και Felica (ISO 18092) και προωθήθηκε κυρίως μέσω του NFC Forum (2004) στο οποίο συμμετέχουν 140 γνωστές εταιρίες και από άλλους οργανισμούς.

Το NFC ορίζει τρεις κατηγορίες λειτουργίας:

- Λειτουργία γρήγορης ανάγνωσης/εγγραφής
- Λειτουργία Peer-to-Peer μέσω σύνδεσης δύο συσκευών ομότιμης σχέσης
- Λειτουργία NFC καρτών εξομοίωσης που επιτρέπει στη συσκευή να συμπεριφέρεται στα πρότυπα μιας smartcard

Η τεχνολογία NFC διαδίδεται με πολύ γρήγορους ρυθμούς λόγω της αξιοποίησης τους από τα έξυπνα κινητά. Για αυτό το λόγο τον Νοέμβριο του 2011 οι 45 μεγαλύτερες εταιρίες κινητής τηλεφωνίας του κόσμου με κοινή τους ανακοίνωση δεσμεύτηκαν να υποστηρίξουν τις νέες κάρτες SIM που θα ενσωματώνουν την τεχνολογία NFC.

Η διασύνδεση αυτή έχει βέβαια και πλεονεκτήματα και μειονεκτήματα:

Πλεονεκτήματα

- Οι NFC αλληλεπιδράσεις είναι εύκολες και απλές καθώς δεν χρειάζεται παρά μόνο ένα απλό άγγιγμα.
- Η χρήση NFC είναι ιδανική για το ευρύτερο φάσμα των επιχειρήσεων καθώς είναι εύκολη στη χρήση, βελτιώνει την επικοινωνία μεταξύ των μελών της επιχείρησης.
- Η NFC τεχνολογία διευκολύνει την απλή και γρήγορη εγκατάσταση των ασύρματων τεχνολογιών όπως το Bluetooth και το WiFi.
- Είναι εγγενώς ασφαλής η χρήση καθώς οι μεταδόσεις είναι μικρής εμβέλειας (από ένα άγγιγμα σε μόλις λίγα εκατοστά). Επίσης σημαντικό χαρακτηριστικό είναι ότι δεν μπορεί να γίνει υποκλοπή δεδομένων ασύρματα.
- Βρίσκει εφαρμογή σε πολλές χρήσεις όπως στις πληρωμές, στα εισιτήρια, στη διαφήμιση, στις έξυπνες κάρτες, στην ανταλλαγή δεδομένων, στην κρυπτογράφηση παρουσίας και στον έλεγχο πρόσβασης.
- Αξιοποιεί τα κινητά τηλέφωνα ως μέσο αλληλεπίδρασης. Είναι ευρέως διαδεδομένα και τα κουβαλάμε πάντα μαζί μας, έχουν επεξεργαστή, έχουν συνήθως πρόσβαση στο διαδίκτυο, είναι διαδραστικά (πληκτρολόγιο, οθόνη αφής).

Μειονεκτήματα

- Τα συστήματα NFC είναι εύκολο να υποκλαπούν. Οποιοσδήποτε είναι σε θέση να κλέψει τις προσωπικές πληροφορίες του καθενός πολύ εύκολα και αυτό γιατί δεν υπάρχει κάποιο αυστηρό μέτρο ασφαλείας. Μια προσθήκη θα μπορούσε να είναι ένα σύστημα αναγνώρισης προσώπου ή αναγνώρισης δακτυλικών αποτυπωμάτων.
- Η χρήση του NFC εκπέμπει ακτινοβολία.
- Ένα άλλο θέμα είναι ότι επειδή η λειτουργία του NFC γίνεται εξ αποστάσεως υπάρχει ο κίνδυνος απώλειας των δεδομένων. [[13] <http://el.wikipedia.org/wiki/NFC>]



Εικόνα 11

Σύγκριση Bluetooth και NFC

Το NFC και το Bluetooth είναι τεχνολογίες υψηλής συχνότητας ασύρματης επικοινωνίας μικρής εμβέλειας που περιλαμβάνονται σε ηλεκτρονικές συσκευές για εύκολη και ασφαλή αλληλεπίδραση ανάμεσα σε δύο ηλεκτρονικές συσκευές. Το NFC είναι μία τεχνολογία ασύρματης σύνδεσης η οποία μπορεί να χρησιμοποιηθεί για μια αμφίδρομη αλληλεπίδραση ανάμεσα σε ηλεκτρονικές συσκευές εντός κάποιων εκατοστών. Το Bluetooth είναι επίσης μία ασύρματη τεχνολογία η οποία σχεδιάστηκε για την αλληλεπίδραση μεταξύ των συσκευών επικοινωνίας εντός 10 μέτρων εμβέλειας χωρίς φυσική σύνδεση.

Το NFC εδραιώνει μια σύνδεση πιο γρήγορα από το κανονικό Bluetooth, αλλά όχι πιο γρήγορα από το Bluetooth χαμηλής ενέργειας. Αυτό συμβαίνει διότι το NFC εδραιώνει αυτόματα μια σύνδεση ανάμεσα σε δυο συσκευές σε λιγότερο από ένα δέκατο του δευτερολέπτου. Ο μέγιστος ρυθμός μεταφοράς δεδομένων για το NFC είναι 424 Kbit/s, και είναι πολύ πιο αργό από το Bluetooth που στις πρώτες του εκδόσεις ήταν 2,1 Mbit/s.

Το NFC απαιτεί ωστόσο λιγότερη ενέργεια υπό κανονικές συνθήκες, ενώ όταν δουλεύει με μία μη τροφοδοτούμενη με ρεύμα συσκευή όπως για παράδειγμα μια έξυπνη πιστωτική κάρτα, τότε η κατανάλωση ενέργειας ανεβαίνει. Το NFC βασίζεται στην επαγωγική ζεύξη, όπου τα αόριστα συνδεδεμένα επαγωγικά κυκλώματα μπορούν να χρησιμοποιηθούν για να μοιράζονται ενέργεια και δεδομένα ανάμεσα σε δύο συσκευές σε πολύ μικρή απόσταση. Το Bluetooth είναι ένα ιδιόκτητο πρωτόκολλο για μικρής εμβέλειας επικοινωνία με υψηλό επίπεδο ασφαλείας. Αναπτύχθηκε από την Telecom Vendor Ericsson. Λειτουργεί στη συχνότητα ISM (2,4 GHz). [[13] <http://el.wikipedia.org/wiki/NFC>]

	NFC	Bluetooth	Bluetooth Low Energy
Συμβατό με RFID	ISO 18000-3	ενεργό	ενεργό
Οργανισμός προτυποποίησης	ISO/IEC	Bluetooth SIG	Bluetooth SIG
Πρότυπο δικτύου	ISO 13157	IEEE 802.15.1	IEEE 802.15.1
Τύπος δικτύου	Σημείο-σε-σημείο (Point-to-point)	WPAN	WPAN
Κρυπτογράφηση	όχι με το RFID	διαθέσιμη	διαθέσιμη
Εμβέλεια	< 0.2 m	~10 m (κλάση 2)	~100 m
Συχνότητα	13.56 MHz	2.4–2.5 GHz	2.4–2.5 GHz
Ρυθμός δεδομένων (bit ανά δευτερόλεπτο)	424 kbit/s	2.1 Mbit/s	~1.0 Mbit/s
Χρόνος προετοιμασίας	< 0.1 s	< 6 s	< 0.006 s
Κατανάλωση ενέργειας	< 15mA (ανάγνωση)	εξαρτάται από την κλάση	< 15 mA (αποστολή ή λήψη)



Προγραμματισμός

Γλώσσα προγραμματισμού λέγεται μια τεχνητή γλώσσα που μπορεί να χρησιμοποιηθεί για τον έλεγχο μιας μηχανής, συνήθως ενός υπολογιστή. Οι γλώσσες προγραμματισμού, όπως και οι ανθρώπινες γλώσσες, ορίζονται από ένα σύνολο συντακτικών και εννοιολογικών κανόνων, που ορίζουν τη δομή και το νόημα, αντίστοιχα, των προτάσεων της γλώσσας. Οι γλώσσες προγραμματισμού χρησιμοποιούνται για να διευκολύνουν την οργάνωση και διαχείριση πληροφοριών, αλλά και για την ακριβή διατύπωση αλγορίθμων. Ορισμένοι ειδικοί χρησιμοποιούν τον όρο γλώσσα προγραμματισμού μόνο για τυπικές γλώσσες που μπορούν να εκφράσουν όλους τους πιθανούς αλγορίθμους. Μη υπολογιστικές γλώσσες όπως η HTML ή τυπικές γραμματικές όπως η BNF δεν λέγονται συνήθως γλώσσες προγραμματισμού. Υπάρχουν χιλιάδες διαφορετικές γλώσσες προγραμματισμού, και κάθε χρόνο δημιουργούνται περισσότερες. Εμείς θα αναλύσουμε περισσότερο την Java, την οποία θα χρησιμοποιήσουμε για να προγραμματίσουμε την εφαρμογή που θα τρέχει στις κινητές συσκευές, για να μπορεί να επικοινωνεί με το καταγραφικό που θα κατασκευάσουμε. Κάθε γλώσσα προγραμματισμού έχει το δικό της σύνολο τυπικών προδιαγραφών-κανόνων που αφορούν το συντακτικό, το λεξιλόγιο και το νόημα της. Για πολλές γλώσσες που χρησιμοποιούνται ευρέως και έχουν χρησιμοποιηθεί για αρκετό χρονικό διάστημα (π.χ. C, C++, Java, Scheme), υπάρχουν ειδικοί φορείς τυποποίησης, οι οποίοι μέσα από τακτές συναντήσεις δημιουργούν, τροποποιούν ή επεκτείνουν τις τυπικές προδιαγραφές που διέπουν τη χρήση μιας γλώσσας προγραμματισμού. Άλλες γλώσσες δεν περιγράφονται σε κάποιο επίσημο πρότυπο αλλά ορίζονται μόνο με βάση κάποια υλοποίησή τους, όπως η Python που περιγράφεται από την υλοποίηση CPython.

Δεν υπάρχει απλός τρόπος να κατηγοριοποιηθούν οι γλώσσες προγραμματισμού. Αυτό συμβαίνει γιατί συνήθως κάθε γλώσσα προγραμματισμού περιέχει επιρροές από πολλές προηγούμενες γλώσσες, συνδυάζοντας θετικά στοιχεία και προσθέτοντας νέα. Χαρακτηριστικά που εμφανίζονται σε μια γλώσσα και έχουν θετική αποδοχή, συνήθως υιοθετούνται από μεταγενέστερες γλώσσες ακόμα και αν πρόκειται για γλώσσες που ανήκουν σε διαφορετική κατηγορία.

Η κατηγοριοποίηση είναι ακόμα πιο περίπλοκη για το λόγο ότι πολλές γλώσσες συνήθως ανήκουν σε παραπάνω από μία κατηγορίες. Για παράδειγμα, η Java είναι τόσο αντικειμενοστρεφής όσο και παράλληλη γλώσσα, δεδομένου ότι υποστηρίζει την οργάνωση των δεδομένων και υπολογισμών σε αντικείμενα, αλλά επιτρέπει επίσης και την δημιουργία προγραμμάτων με ταυτόχρονα νήματα (threads) που εκτελούνται παράλληλα.

Δεδομένης της δυσκολίας στην κατηγοριοποίηση, μπορούμε να κατηγοριοποιήσουμε τις γλώσσες προγραμματισμού με διάφορους τρόπους. Οι συνηθέστεροι τρόποι είναι:

- με βάση τον τρόπο οργάνωσης του προγράμματος
- με βάση τον στόχο που έχει η γλώσσα
- με βάση τον τρόπο που περιγράφουν το ζητούμενο αποτέλεσμα

Στην περίπτωση κατηγοριοποίησης με βάση τον τρόπο οργάνωσης του προγράμματος έχουμε τις εξής υποκατηγορίες :

- Διαδικαστικές γλώσσες (procedural) όπου το πρόγραμμα είναι οργανωμένο σε διαδικασίες, που αποτελούνται από σειρές εντολών που περιγράφουν αλγορίθμους. Παραδείγματα γλωσσών που ανήκουν σε αυτή την κατηγορία είναι η Pascal ή η C.
- Αντικειμενοστραφείς γλώσσες (object-oriented) όπου το πρόγραμμα είναι οργανωμένο σε αντικείμενα. Ένα αντικείμενο είναι μια μονάδα που αποτελείται από την περιγραφή κάποιων δεδομένων και την περιγραφή των αλγορίθμων που τα επεξεργάζονται. Ένα αντικειμενοστραφές πρόγραμμα αποτελείται από διάφορα αντικείμενα που αλληλεπιδρούν μεταξύ τους. Παραδείγματα αντικειμενοστραφών γλωσσών είναι η Java ή η C++.
- Συναρτησιακές γλώσσες (functional) όπου οι υπολογισμοί εκφράζονται ως εφαρμογές μαθηματικών συναρτήσεων, σε αντίθεση με τα άλλα είδη προγραμματισμού όπου οι υπολογισμοί εκφράζονται ως σειρές εντολών, όπου η κάθε μία αλλάζει με κάποιο τρόπο την κατάσταση του συστήματος. Θεωρητικό τους υπόβαθρο είναι ο λ-λογισμός. Χαρακτηριστικές συναρτησιακές γλώσσες είναι η Lisp, η Haskell και η OCaml.

Στην περίπτωση που η κατηγοριοποίηση των γλωσσών προγραμματισμού γίνει με βάση το στόχο που έχει η γλώσσα, υπάρχουν οι παρακάτω υποκατηγορίες:

- Γλώσσες γενικής χρήσης. Σε αυτήν την κατηγορία ταξινομούνται γλώσσες που δημιουργήθηκαν για τον προγραμματισμό γενικών εφαρμογών, καθώς και πολλές εκπαιδευτικές γλώσσες που αποδείχτηκαν χρήσιμες για την ανάπτυξη γενικών εφαρμογών, όπως η Pascal.
- Γλώσσες προγραμματισμού συστημάτων, που χρησιμοποιούνται συνήθως για τον προγραμματισμό λειτουργικών συστημάτων ή οδηγών (drivers) υλικού, όπου χρειάζεται πολλές φορές ο προγραμματιστής να έχει έλεγχο και γνώση του πως λειτουργεί το υλικό. Η πιο συχνά χρησιμοποιούμενη γλώσσα προγραμματισμού συστημάτων είναι η C.
- Γλώσσες σεναρίων (scripting). Αυτές οι γλώσσες χρησιμοποιούνται συνήθως για τη γρήγορη ανάπτυξη μικρών προγραμμάτων, σε περιπτώσεις που ο χρόνος του προγραμματιστή είναι πιο πολύτιμος από την ταχύτητα εκτέλεσης του προγράμματος, όπως για παράδειγμα συμβαίνει όταν το πρόγραμμα απλά αυτοματοποιεί απλές λειτουργίες. Παραδείγματα γλωσσών σεναρίων (scripting) είναι η Perl, η Python, η Ruby ή τα κελύφη του λειτουργικού συστήματος Unix (shells).
- Γλώσσες ειδικών εφαρμογών. Σε αυτή την κατηγορία ανήκουν γλώσσες που αναπτύχθηκαν ειδικά για μια συγκεκριμένη εφαρμογή. Για παράδειγμα, η γλώσσα PostScript είναι σχεδιασμένη ειδικά για να περιγράφονται με λεπτομέρεια κείμενα προς εκτύπωση, ενώ η γλώσσα Matlab είναι σχεδιασμένη για την επεξεργασία πινάκων από αριθμητικά δεδομένα.
- Παράλληλες ή καταναμημένες γλώσσες. Στη συγκεκριμένη κατηγορία ταξινομούνται γλώσσες που επιτρέπουν τη ανάπτυξη παράλληλων προγραμμάτων, όπου πολλές εντολές εκτελούνται ταυτόχρονα σε πολλούς υπολογιστές, έτσι ώστε το τελικό αποτέλεσμα να προκύψει γρηγορότερα. Οι παράλληλες γλώσσες προσφέρουν συνήθως εύκολους τρόπους επικοινωνίας μεταξύ των νημάτων που εκτελούνται παράλληλα, καθώς και τρόπους ώστε να δημιουργούνται καινούριες παράλληλες εκτελέσεις. Παραδείγματα γλωσσών που ανήκουν (και) σε αυτή την κατηγορία είναι η Go, η Java, η Erlang, η MultiLisp ή η Cilk.

Τέλος, στην περίπτωση που η κατηγοριοποίηση γίνεται με βάση τον τρόπο που περιγράφεται το ζητούμενο, υπάρχουν οι παρακάτω κατηγορίες:

- Προστακτικές γλώσσες προγραμματισμού (imperative) είναι οι γλώσσες που περιγράφουν το ζητούμενο αποτέλεσμα κατασκευαστικά, δίνοντας μια σειρά εντολών που όταν εκτελεστούν παράγουν το ζητούμενο αποτέλεσμα. Τέτοιες γλώσσες είναι η C, η Java αλλά και η OCaml.
- Δηλωτικές γλώσσες προγραμματισμού (declarative) είναι οι γλώσσες που περιγράφουν το ζητούμενο αποτέλεσμα χρησιμοποιώντας τις ιδιότητες που έχει, και όχι τον τρόπο με τον οποίο υπολογίζεται. Παραδείγματα δηλωτικών γλωσσών είναι η Haskell, η SQL και η Prolog. [B1]



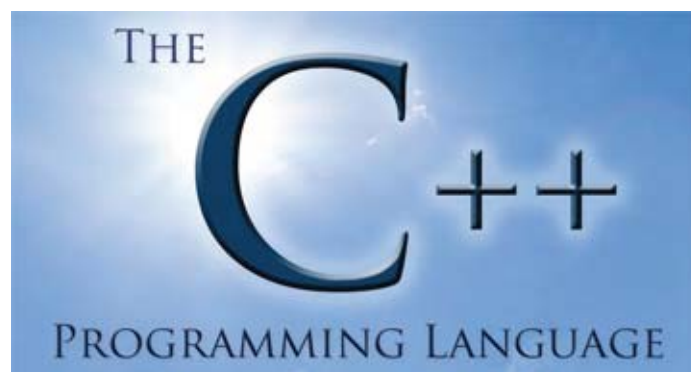
Γλώσσα προγραμματισμού C++

Η C++ είναι μια γενικού σκοπού γλώσσα προγραμματισμού Η/Υ. Θεωρείται μέσου επιπέδου γλώσσα, καθώς περιλαμβάνει έναν συνδυασμό χαρακτηριστικών από γλώσσες υψηλού και χαμηλού επιπέδου. Είναι μια μεταγλωττιζόμενη γλώσσα πολλαπλών παραδειγμάτων, με τύπους. Υποστηρίζει δομημένο, αντικειμενοστρεφή και γενικό προγραμματισμό.

Η γλώσσα αναπτύχθηκε από τον Μπιάρνε Στρούστρουπ το 1979 στα εργαστήρια Bell της AT&T, ως βελτίωση της ήδη υπάρχουσας γλώσσας προγραμματισμού C, και αρχικά ονομάστηκε "C with Classes", δηλαδή C με κλάσεις. Μετονομάστηκε σε C++ το 1983. Οι βελτιώσεις ξεκίνησαν με την προσθήκη κλάσεων, και ακολούθησαν, μεταξύ άλλων, εικονικές συναρτήσεις, υπερφόρτωση τελεστών, πολλαπλή κληρονομικότητα, πρότυπα κ.α.

Η γλώσσα ορίστηκε παγκοσμίως, το 1998, με το πρότυπο ISO/IEC 14882:1998. Η τρέχουσα έκδοση αυτού του προτύπου είναι αυτή του 2003, η ISO/IEC 14882:2003. Μια καινούρια έκδοση είναι υπό ανάπτυξη, γνωστή ανεπίσημα με την ονομασία C++0x.

Η C++ παρέχει περισσότερους από 30 τελεστές, που καλύπτουν τη βασική αριθμητική, το χειρισμό bit, αναφορά δεικτών, συγκρίσεις, λογικές πράξεις κ.α. Σχεδόν όλοι οι τελεστές μπορούν να υπερφορτωθούν για τύπους ορισμένους από το χρήστη, με λίγες εξαιρέσεις όπως πρόσβαση μέλους. Το πλούσιο σύνολο από τελεστές που μπορούν να υπερφορτωθούν είναι βασικό για τη χρήση της C++ ως γλώσσα ειδικού πεδίου (domain specific language). Οι υπερφορτώσιμοι τελεστές είναι ακόμα βασικό μέρος πολλών προχωρημένων τεχνικών προγραμματισμού της C++, όπως οι έξυπνοι δείκτες. Η υπερφόρτωση ενός τελεστή δεν αλλάζει την προτεραιότητα των υπολογισμών όπου χρησιμοποιείται, ούτε τον αριθμό των τελεστών που χρησιμοποιεί ο τελεστής (αν και οποιοσδήποτε τελεστέος μπορεί απλά να αγνοείται). [B3]



Εικόνα 12

Γλώσσα προγραμματισμού Java

Η γλώσσα προγραμματισμού Java είναι ένα πρόσφατο προϊόν της Sun Microsystems Inc.. Ξεκίνησε σαν μέρος ενός μεγαλύτερου σχεδίου που αφορούσε την ανάπτυξη λογισμικού για καταναλωτικά ηλεκτρονικά. Πρόκειται για μικρές, αξιόπιστες, φορητές, πραγματικού χρόνου συσκευές που στην αρχή βασιζόντουσαν στην C++. Αρκετά προβλήματα όμως παρουσιάστηκαν και η γλώσσα C++ δεν μπόρεσε να εφαρμοστεί τελικά. Χρειάστηκε να αναπτυχθεί μία νέα γλώσσα: η Java. Η Java, στην τελική της μορφή, βρήκε περαιτέρω εφαρμογή στην επίλυση μερικών προβλημάτων του σημερινού προγραμματισμού, όπως animation, την αλληλεπίδραση πραγματικού χρόνου (real-time interaction) και την εξερεύνηση του Web (Web browsing). Από την ημέρα της δημοσιοποίησης της τον Μάιο του 1995, η Java έχει εξαπλωθεί σε όλο το Internet. Η Java έχει κιόλας λύσει τα περισσότερα προβλήματα στο μοντέλο client/server και έχει προάγει την χρήση του World Wide Web. Παρ' όλα αυτά, επειδή η Java έχει σχεδιαστεί για να χρησιμοποιείται σε ανοικτά, δικτυωμένα περιβάλλοντα, έχει βαθιές συνέπειες στην ασφάλεια ενός συστήματος υπολογιστών. Επίσης, η δυνατότητα της να διανέμει αυτόματα εκτελέσιμα αρχεία μέσω του Internet επηρεάζει, με την σειρά της, την ασφάλεια των δικτύων. Τα εκτελέσιμα αρχεία από το Internet συχνά κρύβουν παγίδες και δύσκολα μπορεί να είναι κανείς σίγουρος ότι ο κώδικας είναι ελεύθερος από ιούς.

Η Java είναι δύο πράγματα:

- Γλώσσα προγραμματισμού
- Πλατφόρμα.

Η Java σαν γλώσσα προγραμματισμού, χαρακτηρίζεται από τα εξής: απλή, αντικειμενοστραφής, συμβατή με δικτυακά πρωτόκολλα, ουδέτερη της υποκείμενης αρχιτεκτονικής, φορητή ασφαλής, υψηλής απόδοσης, δυναμική, σταθερή, interpreted και multithreaded.

Στόχος της ομάδας της Sun που ανέπτυξε την Java, ήταν μια γλώσσα εύκολη στην χρήση, που δεν απαιτεί πολλή εξάσκηση και εκπαίδευση. Οι περισσότεροι προγραμματιστές στις μέρες μας δουλεύουν είτε με την C είτε με την C++. Έτσι, μολονότι η C++ δεν ήταν η κατάλληλη για το αρχικό σχέδιο, η Java σχεδιάστηκε βάσει της C++, με σκοπό να γίνει όσο το δυνατόν περισσότερο κατανοητή.

Η Java παραλείπει πολλά από τα σπανίως χρησιμοποιούμενα και δυσκολονόητα χαρακτηριστικά της C++, που δεν ωφελούν και πολύ την ευελιξία της γλώσσας. Προστέθηκαν διεργασίες, όπως η αυτόματη συλλογή των "σκουπιδιών" (automatic garbage collection), διευκολύνοντας τον προγραμματισμό σε Java. Μια κοινή πηγή πολυπλοκότητας της C++ και της C είναι η διαχείριση της μνήμης. Με την καινούργια διεργασία της αυτόματης συλλογής "σκουπιδιών", που συνιστάται από την περιοδική αποδέσμευση της μνήμης που δεν χρησιμοποιείται, μεγάλο μέρος από την δουλεία των προγραμματιστών αυτοματοποιείται και μειώνονται τα bugs.

Ένα πλεονέκτημα της Java που οφείλεται στην απλότητα της είναι ότι το μέγεθος των απαραίτητων εργαλείων. Ο Java interpreter και οι βασικές βιβλιοθήκες είναι μικρές και ο κώδικάς της Java είναι τόσο περιορισμένος σε μέγεθος που μπορεί άνετα να τρέξει σε οποιαδήποτε μικρή μηχανή και να κατέβει από το δίκτυο.

Λέγοντας ότι μία γλώσσα προγραμματισμού είναι αντικειμενοστραφής, εννοούμε η τεχνική σχεδιασμού ενός προγράμματος συγκεντρώνεται σε αντικείμενα. Ένα αντικείμενο είναι ο συνδυασμός δεδομένων, διαδικασιών και λειτουργιών με βασική ιδιότητα την απόκρυψη του συνδυασμού αυτού. Το κάθε αντικείμενο, δηλαδή, αντιμετωπίζεται σαν ένα "μαύρο κουτί". Τα αντικείμενα δεν είναι ανεξάρτητα μεταξύ τους, αλλά βρίσκονται σε σχέση αλληλεξάρτησης με τα υπόλοιπα. Υπάρχει η έννοια της κληρονομικότητας μεταξύ των αντικειμένων, δηλαδή ένα αντικείμενο μπορεί να κληρονομήσει δεδομένα από άλλα αντικείμενα.

Οι γλώσσες αντικειμενοστραφή προγραμματισμού είναι γλώσσες υψηλού επιπέδου, αφαιρετικές, αποτελεσματικές, γρήγορες και χρησιμοποιούνται για την δημιουργία μεγάλων και σημαντικών εφαρμογών. Οι αντικειμενοστραφής ευκολίες της Java είναι ίδιες με αυτές της C++, με επεκτάσεις από την Objectine C.

Η Java έχει μια μεγάλη βιβλιοθήκη από ρουτίνες για την επιτυχημένη συνεργασία με τα πρωτόκολλα HTTP και FTP. Κατ' αυτόν τον τρόπο, οι δικτυακές συνδέσεις δημιουργούνται ευκολότερα από ότι με την C ή την C++.

Τα προγράμματα σε Java μπορούν να έχουν πρόσβαση μέσω δικτύου σε αντικείμενα, με την ίδια άνεση που ένας χρήστης προσπελάζει ένα τοπικό σύστημα αρχείων.

Η Java προορίζεται για την σύνταξη προγραμμάτων που θα είναι αξιόπιστα από όλες τις πλευρές. Δίνεται έμφαση στον από νωρίς έλεγχο για πιθανά προβλήματα και στον έλεγχο σε πραγματικό χρόνο και στην εξάλειψη καταστάσεων που προκαλούν λάθη.

Η μεγαλύτερη διαφορά μεταξύ Java και C/C++ είναι το γεγονός ότι η Java έχει ένα μοντέλο δεικτών που εξαφανίζει την πιθανότητα της επαναχρησιμοποίησης της μνήμης και την καταστροφή των δεδομένων. Αντί για αριθμητικούς δείκτες (pointer arithmetic), η Java έχει πραγματικούς πίνακες (true arrays). Οι προγραμματιστές της Java δεν έχουν να φοβηθούν την ακούσια (ή μη) τροποποίηση της μνήμης, γιατί δεν υπάρχουν δείκτες (pointers). Εξάλλου, τα προγράμματα σε Java δεν μπορούν να αποκοτήσουν μη εγκεκριμένη πρόσβαση στην μνήμη.

Η Java προορίζεται για χρήση σε ανοικτά, δικτυωμένα περιβάλλοντα. Γι' αυτό το λόγο, ιδιαίτερη προσοχή έχει δοθεί στην ασφάλεια που παρέχει η γλώσσα. Η Java επιτρέπει την κατασκευή προγραμμάτων ελεύθερων από ιούς και η τροποποίηση τους είναι αδύνατη. Οι τεχνικές πιστοποίησης ταυτότητας βασίζονται στην ασύμμετρη κρυπτογραφία.

Υπάρχει μεγάλη σχέση μεταξύ του τρόπου διαχείρισης της μνήμης και της παρεχόμενης ασφάλειας. Αλλαγές στην σημασιολογία των δεικτών της μνήμης κάνουν αδύνατη την μη έγκυρη πρόσβαση στα δεδομένα της μνήμης ή της πρόσβασης των δεδομένων των αντικειμένων. Με αυτόν τον τρόπο καταπολεμούνται οι περισσότεροι ιοί.

Η Java έχει σχεδιαστεί για να υποστηρίζει δικτυακές εφαρμογές. Ένα δίκτυο, όμως, αποτελείται από ποικιλία διαφορετικών συστημάτων, με διαφορετικές CPU και λειτουργικά συστήματα. Για να μπορούν οι Java εφαρμογές να εκτελούνται παντού στο δίκτυο, το πρόγραμμα Java πρέπει να περάσει από δύο διαδικασίες ώστε να καταλήξει σε εκτελέσιμη μορφή. Πρώτα ο μεταγλωττιστής, μετατρέπει τον πηγαίο κώδικα του προγράμματος σε μία ενδιάμεση γλώσσα που καλείται Java bytecodes. Τα Java bytecodes είναι ανεξάρτητα της πλατφόρμας και με χρήση του ερμηνευτή (interpreter) κάθε bytecode εντολή μετατρέπεται σε κατάλληλη δυαδική μορφή για να τρέξει στον εκάστοτε υπολογιστή. Η μεταγλώττιση (compilation) συμβαίνει μόνο μια φορά για κάθε Java πρόγραμμα, η ερμηνεία (interpretation) γίνεται κάθε φορά που το πρόγραμμα εκτελείται. Το γεγονός ότι είναι ανεξάρτητη της υποκείμενης πλατφόρμας αποτελεί μεγάλο μέρος του ότι είναι φορητή, άλλα υπάρχουν και άλλα σημεία που χαρακτηρίζουν την φορητότητα της. Σε αντίθεση με την C/C++ δεν υπάρχουν καθόλου χαρακτηριστικά που εξαρτώνται από την CPU του υπολογιστή. Έτσι, τα μεγέθη των πρωταρχικών τύπων δεδομένων είναι καθορισμένα και η συμπεριφορά τους είναι παντού η ίδια. Για παράδειγμα, "int" σημαίνει πάντα έναν 32 bit ακέραιο και "float" πάντα αντιπροσωπεύει έναν 32 bit floating αριθμό.

Τα Java bytecodes μεταφράζονται σε πραγματικό χρόνο σε εντολές μηχανής που εξαρτώνται από την εκάστοτε πλατφόρμα, και δεν αποθηκεύονται πουθενά. Η διαδικασία είναι γρήγορη και πιο αποτελεσματική. Μαζί με τα bytecodes μεταφέρονται πληροφορίες που μπορούν να χρησιμοποιηθούν κατά την εκτέλεση και παρέχουν την βάση για τους ελέγχους που πραγματοποιεί ο συνδετής (linker). Επίσης τα προγράμματα γίνονται πιο επιδεκτικά σε debugging διαδικασίες.

Η διαδικασία παραγωγής των εντολών μηχανής είναι απλή και γρήγορη. Ο κώδικας που προκύπτει είναι αποτελεσματικός. Ο μεταγλωττιστής από την μεριά του εφαρμόζει αυτόματη κατανομή των καταχωρητών (automatic register allocation) όταν παράγει τα bytecodes. Η τελική μορφή του κώδικα (εκτελέσιμη δυαδική μορφή) είναι μικρή σε μέγεθος και ταχύτατη στην εκτέλεση.

Τα προγράμματα σε Java έχουν την δυνατότητα να αντιμετωπίζουν πολλές καταστάσεις – διαδικασίες ταυτόχρονα. Σε αντίθεση, η C και C++ είναι single-threaded γλώσσες. Τα πλεονεκτήματα του multithreading είναι η καλύτερη πραγματικού χρόνου συμπεριφορά και η καλύτερη αλληλεπιδραστική ανταπόκριση.

Η Java είναι πιο δυναμική γλώσσα από την C ή C++. Έχει αναπτυχθεί για να προσαρμοστεί σε ένα εξελισσόμενο περιβάλλον. Οι βιβλιοθήκες εργαλείων αναπτύσσονται ελεύθερα με την πρόσθεση νέων μεθόδων και μεταβλητών, χωρίς να επηρεάζονται οι ήδη υπάρχουσες εφαρμογές.

Υπάρχουν δύο είδη προγραμμάτων στην Java: τα applets και οι applications.

- Applets είναι μικρά κομμάτια εκτελέσιμου κώδικα που απαιτούν ένα Web browser για να τρέξουν. Υποδηλώνεται στην HTML σελίδα με ένα <applet> tag. Όταν ο Web browser ενός χρήστη επεξεργαστεί μία ηλεκτρονική σελίδα που περιέχει ένα τέτοιο tag, αυτόματα κατεβάζει το applet και το τρέχει στην μηχανή του client. Λόγω του ότι διακινούνται στο δίκτυο, τα applets είναι μικρά. Τα applets επιτρέπουν τον εμπλουτισμό των σελίδων με αλληλεπιδραστικές εφαρμογές.
- Applications είναι προγράμματα που δεν απαιτούν Web browser για να τρέξουν και δεν διαθέτουν ενσωματωμένο μηχανισμό για μεταφορά στο δίκτυο. Βρίσκονται τοπικά αποθηκευμένες. Δεν διαφέρουν, δηλαδή, από τα προγράμματα σε άλλες γλώσσες. Μπορούν να πραγματοποιήσουν εργασίες όπως επεξεργασία κειμένου και επεξεργασία γραφικών. Όπως και τα applets, οι applications απαιτούν μία Java Virtual Machine για να τρέξουν. Η JVM μπορεί να είναι διαθέσιμη σαν ξεχωριστό πρόγραμμα, μπορεί να είναι ενσωματωμένη στο λειτουργικό σύστημα ή να είναι ενσωματωμένη στην

ίδια την application.

Τα δύο είδη έχουν πολλές κοινές δυνατότητες, όπως πρόσβαση σε απομακρυσμένη βάση δεδομένων, απόκτηση των ζητούμενων δεδομένων, τοπική επεξεργασία των δεδομένων και αποθήκευση των αποτελεσμάτων στον απομακρυσμένο υπολογιστή. Παρ' όλα αυτά, οι applications έχουν μεγαλύτερη ελευθερία όσον αναφορά τις υπηρεσίες του συστήματος. Για παράδειγμα, μια application, σε αντίθεση με ένα applet, έχει πρόσβαση γραφής και ανάγνωσης στα αρχεία οποιουδήποτε δίσκου. Ένα applet έχει πλήρη πρόσβαση μόνο στον σύστημα αρχείων του server από τον οποίο προήλθε. Επίσης, μια ακόμα αξιοσημείωτη διαφορά είναι ότι το applet χρειάζεται δίκτυο για να τρέξει, ενώ μια application όχι. [B2]



Διαφορές Java και C++

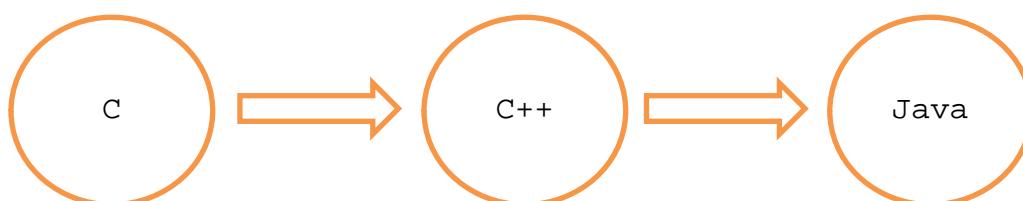
Η Java δημιουργήθηκε αρκετά αργότερα από την C++ και στηρίχθηκε σε μεγάλο βαθμό επάνω σε αυτήν. Για το λόγο αυτό το συντακτικό είναι σχεδόν παρόμοιο, ωστόσο υπάρχει εντυπωσιακά μεγάλος αριθμός διαφορών μεταξύ της C++ και της Java.

Η σύγκριση που ακολουθεί παρουσιάζει τις διαφορές ως σημαντικές βελτιώσεις της Java έναντι της C++, αφού δημιουργήθηκε για την βελτίωση και ευχρηστία της C++.

1. Η εμφανέστερη διαφορά είναι η ταχύτητα. Η Java που υφίσταται διερμηνεία καταλήγει να είναι περίπου 20 φορές πιο αργή από την C++. Ωστόσο αυτό δεν εμποδίζει την κατασκευή μεταγλωττιστών για αυτήν.
2. Στην Java οτιδήποτε που γράφεται πρέπει να βρίσκεται μέσα σε μία κλάση. Δεν υπάρχουν καθολικές συναρτήσεις ή καθολικές μεταβλητές. Αν χρειάζεται το ισοδύναμο των καθολικών, χρησιμοποιούνται static μέθοδοι ή δεδομένα μέσα σε μία κλάση. Δεν υπάρχουν επίσης structs ή enumerations ή unions. Βέβαια enumerations υπάρχουν στην Java στην έκδοση 1,5.
3. Όλες οι μέθοδοι ορίζονται μέσα στο σώμα της κλάσης, επομένως, αυτό θα φαινόταν στην C++ με το να δηλώσουμε όλες τις συναρτήσεις ως inline, αλλά στην πραγματικότητα δεν είναι. Σε αντίθεση, στη C++ μια member function που ορίζεται εξ' ολοκλήρου μέσα στην κλάση, θεωρείται inline.
4. Οι ορισμοί των κλάσεων είναι περίπου ίδιοι για την Java και τη C++, αλλά στην Java δεν υπάρχει ελληνικό ερωτηματικό στο τέλος. Δηλαδή δεν υπάρχουν δηλώσεις κλάσεων, μόνο ορισμοί κλάσεων.
5. Δεν υπάρχει τελεστής εμβέλεια « :: » στην Java, η Java χρησιμοποιεί τον τελεστή « . » για τα πάντα.
6. Η Java έχει όπως και η C++ ορισμένους στοιχειώδεις τύπους. Στην Java είναι οι: boolean, char, short, int, long, float, double. Το μέγεθος των τύπων αυτών είναι καθορισμένο και ανεξάρτητο από την πλατφόρμα. Επίσης ο έλεγχος των τύπων (type-checking) είναι πιο «σφιχτός» στην Java. Για παράδειγμα, στην Java δεν μπορούμε να έχουμε παραστάσεις που λειτουργούν ως συνθήκες ελέγχου και δεν είναι boolean.
7. Ο τύπος char στην Java χρησιμοποιεί το σύνολο χαρακτήρων 16-bit Unicode και κατά συνέπεια αναπαριστά τους περισσότερους εθνικούς χαρακτήρες αυτομάτως.
8. Αλφαριθμητικά μέσα σε “ “ μετατρέπονται αυτομάτως σε αντικείμενα τύπου String. Στην C τα αλφαριθμητικά δεν είναι αντικείμενα.
9. Παρόλο που φαίνονται παρόμοιοι, οι πίνακες στην Java έχουν διαφορετική δομή και συμπεριφορά από ότι στη C++. Υπάρχει μέλος δεδομένων (read-only) length που επιστρέφει το μέγεθος και run-time checking και παράγει εξαίρεση αν «βγούμε» εκτός ορίων, αντίθετα στην C++ που παράγουν σφάλμα.
10. Όλα τα αντικείμενα μη στοιχειωδών τύπων δημιουργούνται μόνο μέσω της δεσμευμένης λέξης new. Στη C++ υπάρχει η δυνατότητα δημιουργίας μη στοιχειωδών τύπων στη στοίβα (new Circle ());).
11. Δεν χρειάζεται στη Java να έχει δηλωθεί κάτι προτού χρησιμοποιηθεί και συνεπώς δεν χρειάζονται και forward class declarations. Αν θέλουμε να χρησιμοποιήσουμε μία κλάση ή μία μέθοδο προτού αυτή οριστεί, απλώς τη χρησιμοποιούμε. Ο μεταγλωττιστής θα βρει τον ορισμό, ακόμα και αν υπάρχει σε άλλο αρχείο.
12. Η Java δεν έχει preprocessor, αν θέλουμε να χρησιμοποιήσουμε κλάσεις μίας βιβλιοθήκης, χρησιμοποιούμε το import και το όνομα της βιβλιοθήκης.
13. Αναφορές σε αντικείμενα που είναι μέλη κλάσεων αρχικοποιούνται αυτόματα σε null. Η αρχικοποίηση στοιχειωδών τύπων δεδομένων είναι εγγυημένη στη Java.
14. Δεν υπάρχουν pointers στη Java με την έννοια της C/C++. Κατά τη δημιουργία ενός αντικειμένου

επιστρέφεται πάντοτε μία αναφορά. Οι αναφορές reference type στη C++ δεν είναι το ίδιο, πρέπει να αρχικοποιηθούν κατά τη δημιουργία τους και δεν μπορούν να «δείχνουν» αλλού στην πορεία. Στη C++ οι pointers μπορούν να δείχνουν οπουδήποτε στη μνήμη πράγμα επικίνδυνο. Στην Java δεν υπάρχει αριθμητική δεικτών.

15. Δεν υπάρχει εντολή goto στην Java. Ο μόνος μηχανισμός άλματος χωρίς συνθήκη είναι η εντολή break label.
16. Η Java δεν έχει templates ή κάποια άλλη υλοποίηση παραμετροποιήσιμων τύπων. Υπάρχει ένα set συλλογών: Vector, Stack και Hashtable που λαμβάνουν αναφορές Object αλλά οι συλλογές αυτές δεν είναι σχεδιασμένες με βάση την απόδοση όπως η C++ Standard Template Library. Βέβαια στην Java 1.5 υπάρχουν generics, επιτρέπεται δηλαδή η δήλωση δομών, όπου καθορίζεται ο τύπος των στοιχείων της χωρίς να χρειάζεται casting για την προσπέλαση ενός στοιχείου.
17. Η Java κάνει αυτόματη περισυλλογή αχρησιμοποίητου χώρου μνήμης και αποδέσμευση του. Στην Java όλα τα αντικείμενα συλλέγονται από τον garbage collector όταν δοθεί η αναφορά null σε όλες τις αναφορές του αντικειμένου. Στη C++ πρέπει να καταμετρώνται όλες οι αναφορές (pointers) με τιμή null και ο καθαρισμός του χώρου μνήμης πρέπει να γίνεται ρητά με την εντολή delete . Ωστόσο τα memory leaks ενώ είναι δύσκολα στην Java, δεν είναι αδύνατα.
18. Η Java έχει ενσωματωμένη υποστήριξη για multithreading. Η κλάση Thread επιτρέπει τη δημιουργία νέων νημάτων, επικαλύπτοντας την run() μέθοδο.
19. Τα προσδιοριστικά ορατότητας public, private, protected δεν εφαρμόζονται στην Java σε μπλοκ κώδικα. Κάθε δήλωση μέλους μιας κλάσης λαμβάνει το δικό της προσδιοριστικό. Αν δεν υπάρχει προσδιοριστικό τα μέλη είναι προσπελάσιμα στο ίδιο πακέτο.
20. Ένα αντικείμενο εσωτερικής (inner) κλάσης στη Java διατηρεί αναφορά προς το «περιβάλλον» αντικείμενο και μπορεί να προσπελάσει τα μέλη της εξωτερικής κλάσης σαν να ήταν δικά του. Στη C++ χρειάζεται προσοχή ο έλεγχος πρόσβασης. Στο πρότυπο 2003 της C++ η εσωτερική κλάση έχει ειδικά δικαιώματα πρόσβασης στα μέλη της περικλείουσας κλάσης. Στο πρότυπο 1998 αντίθετα η εσωτερική κλάση δεν έχει ειδικά δικαιώματα πρόσβασης στην περικλείουσα κλάση.
21. Στην Java χρησιμοποιείται η δεσμευμένη λέξη extends για την κληρονομικότητα, και η super για τον καθορισμό μεθόδων που θα κληθούν από την γονική κλάση. Μόνο όμως μεθόδων που βρίσκονται ένα επίπεδο πιο πάνω.
22. Η Java παρέχει τη δυνατότητα διασυνδέσεων μέσω της δεσμευμένης λέξης interface ως ισοδύναμο μιας αφηρημένης κλάσης βάσης.
23. Στην Java δεν υπάρχει η αναγκαιότητα της δεσμευμένης λέξης virtual καθώς σε όλες τις κλήσεις μη-στατικών μεθόδων χρησιμοποιείται δυναμική διασύνδεση. Στη C++ η δυνατότητα αυτή υπάρχει για λόγους απόδοσης.
24. Η Java δεν υποστηρίζει πολλαπλή κληρονομικότητα. Υλοποίηση πολλαπλών διασυνδέσεων ωστόσο επιτρέπεται. [B4]



Εργαλεία ανάπτυξης εφαρμογών

Όπως έχουμε δει μέχρι τώρα οι εφαρμογές για το λειτουργικό σύστημα Android γράφονται στην γλώσσα προγραμματισμού Java. Απαραίτητο είναι, για όποιον επιθυμεί να αναπτύξει εφαρμογές για Android, να διαθέτει λειτουργικό σύστημα Windows, Linux ή Mac OS. Επιπρόσθετα, χρειάζονται τα παρακάτω εργαλεία, στις αντίστοιχες εκδόσεις του λειτουργικού στο οποίο θα πραγματοποιηθεί η διαδικασία ανάπτυξης:

1. Java Development Kit
2. Eclipse IDE για Java
3. Android SDK
4. ADT (Android Development Tools) Plug-in για το Eclipse

Όλα τα απαραίτητα εργαλεία είναι διαθέσιμα δωρεάν και η εγκατάστασή τους είναι αρκετά εύκολη και σύντομη διαδικασία, βέβαια η χρονική διάρκεια της εγκατάστασης εξαρτάται σημαντικά από την ταχύτητα της σύνδεσης στο διαδίκτυο.

❖ Εγκατάσταση του Java Development Kit

Για την συγγραφή οποιοσδήποτε εφαρμογής Java είναι απαραίτητη η ύπαρξη του Java Development Kit (JDK). Το JDK περιλαμβάνει ένα σύνολο εργαλείων ανάπτυξης προγραμμάτων για την γλώσσα προγραμματισμού Java όπως είναι ο Java (φορτωτής Java εφαρμογών), ο Java c (ο μεταφραστής του Java κώδικα σε Java bytecodes), ο Java h (δημιουργός C stubs για συγγραφή native μεθόδων), ο jar (ο γνωστός Java archiver) και άλλα. Το JDK είναι το υποσύνολο του Java SDK που είναι απαραίτητο για τον προγραμματισμό και την εκτέλεση εφαρμογών σε Java.

Το Java SDK περιλαμβάνει επιπλέον στοιχεία όπως application servers, επιπλέον debuggers και documentation. Για την ανάπτυξη εφαρμογών για το λειτουργικό σύστημα Android, το JDK είναι αρκετό. Η τρέχουσα έκδοσή του είναι η 7 (update 5) και το αρχείο εγκατάστασης είναι διαθέσιμο από τον ιστότοπο της Oracle.

❖ Εγκατάσταση του Eclipse IDE για Java

Το Eclipse IDE (Integrated Development Environment) είναι ένα ολοκληρωμένο περιβάλλον ανάπτυξης εφαρμογών. Δεν είναι απαραίτητο να έχει κάποιος ένα περιβάλλον ανάπτυξης προ κειμένου να αναπτύξει εφαρμογές, ένα απλό σημειωματάριο αρκεί. Οι τεράστιες, όμως, διευκολύνσεις που παρέχει ένα τέτοιο περιβάλλον το καθιστούν απαραίτητο για μεσαίου και μεγάλου μεγέθους εφαρμογές.

Το Eclipse παρέχει ένα περιβάλλον για την οργάνωση του κώδικα ενός ολοκληρωμένου project. Λειτουργίες όπως, διατήρηση εκδόσεων (versioning), διαμοιρασμός αλλαγών στον κώδικα μεταξύ των μελών της ομάδας ανάπτυξης, εικονικοί εξυπηρετητές για έλεγχο ορθότητας λειτουργίας web εφαρμογών και άλλες παρέχονται εντός του ίδιου του περιβάλλοντος. Επιπλέον, με την υποστήριξη των plug-in το Eclipse μπορεί να επεκτείνει τις λειτουργίες που προσφέρει ακόμα παραπέρα. Χαρακτηριστικό παράδειγμα είναι το ADT Plug-in για την ανάπτυξη εφαρμογών Android που θα δούμε παρακάτω.

Η εγκατάσταση του Eclipse είναι πολύ απλή υπόθεση. Το μόνο που χρειάζεται να γίνει είναι να αποθηκευθεί ο φάκελος του Eclipse που λαμβάνουμε από τον ιστότοπο www.eclipse.org σε μια επιθυμητή διαδρομή στον δίσκο μας. Από εκεί και πέρα το πρόγραμμά μας είναι πλήρως λειτουργικό. Λεπτομέρειες για τον τρόπο λειτουργίας του περιβάλλοντος παρατίθενται στο τέλος του κεφαλαίου όπου παρουσιάζεται η ανάπτυξη της πιο απλής εφαρμογής για το Android, της Hello World.

❖ Εγκατάσταση του Android SDK

Το Android SDK αποτελεί απαραίτητο εργαλείο για την ανάπτυξη εφαρμογών για Android. Αυτό περιλαμβάνει τις απαραίτητες βιβλιοθήκες για την ανάπτυξη εφαρμογών Android, παραδείγματα εφαρμογών με τον πηγαίο κώδικά τους, τον εξομοιωτή συσκευών καθώς και άλλες βιβλιοθήκες που έχουν να κάνουν με την τιμολόγηση των εφαρμογών και την πώλησή τους στο Google play. Το SDK μπορεί να το προμηθευτεί κανείς από τον ιστότοπο της Google developer google.com.

❖ Εγκατάσταση του ADT (Android Development Tools) Plug-in για το Eclipse

Χάρη στα Plug-in το Eclipse μπορεί να επεκτείνει τις λειτουργικότητες που προσφέρει. Εκμεταλλευόμενη το γεγονός αυτό η Google ανέπτυξε το ADT (Android Development Tools) Plugin για το Eclipse. Με το plug-in αυτό το Eclipse προσφέρει την δυνατότητα σχεδίασης διεπαφών χρήστη για τις εφαρμογές Android με γραφικό τρόπο (παρόμοια με το Visual Studio της Microsoft).

Επεκτείνει τις επιλογές των μενού εισαγωγής του Eclipse έτσι ώστε, για παράδειγμα, να μπορεί κανείς να δημιουργήσει ένα καινούριο project Android με την σωστή δομή και αρχεία αυτοματοποιημένα εντός του περιβάλλοντος. Επιπλέον, παρέχεται ειδικός editor για τα διάφορα XML αρχεία που περιέχονται σε ένα project Android καθώς και εργαλεία για την παραγωγή των τελικών πακέτων, της εφαρμογής, προς πώληση.

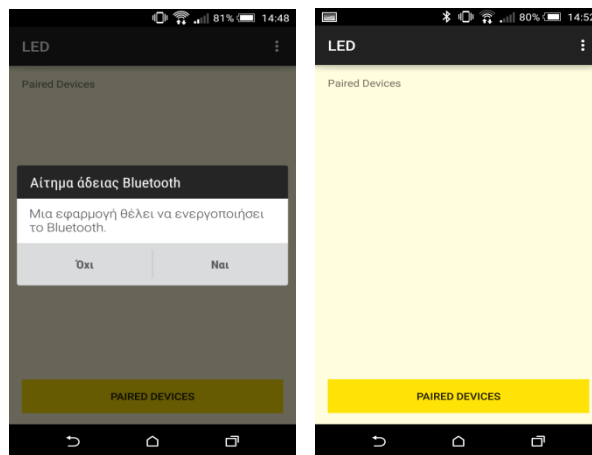
Η εγκατάσταση του plug-in μπορεί να γίνει εντός του ίδιου του Eclipse κάτι το οποίο είναι πολύ σημαντικό και εύχρηστο για τον χρήστη. [B1]



Εικόνα 14

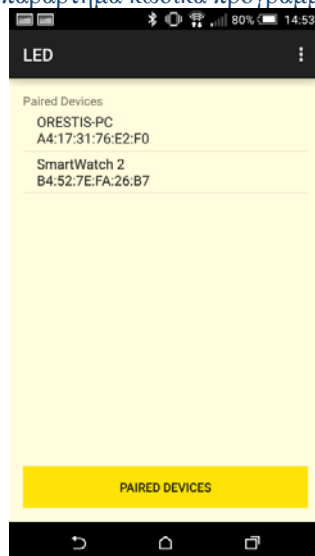
Πληροφορίες Εφαρμογής

Για να μπορεί η εφαρμογή που θα κατασκευάσουμε να πληροί τις προϋποθέσεις της πτυχιακής θα πρέπει αρχικά να μας δίνει την δυνατότητα να ενεργοποιούμε την κεραία Bluetooth της συσκευής στην οποία είναι εγκατεστημένη η εφαρμογή μας. Αυτό θα πραγματοποιείται κατά την εκκίνηση της εφαρμογής, δηλαδή όταν το Bluetooth δεν είναι ενεργοποιημένο, θα εμφανίζεται στο άνοιγμα της εφαρμογής ένα μήνυμα που θα ρωτάει τον χρήστη αν θέλει να ενεργοποιήσει την σύνδεση. Η απάντηση δίνεται πατώντας ένα από τα δύο κουμπιά που εμφανίζονται. Οι επιλογές είναι «Ναι» ή «Όχι». Στην περίπτωση που ο χρήστης επιλέξει το κουμπί «Όχι» τότε η εφαρμογή τερματίζει ενώ διαφορετικά πραγματοποιείται έναρξη της εφαρμογής. Αφού ο χρήστης θα έχει επιλέξει την ενεργοποίηση της διασύνδεσης Bluetooth η εφαρμογή θα είναι έτοιμη να αναζητήσει τις συσκευές που βρίσκονται στον χώρο. Αυτή η ενέργεια πραγματοποιείται με το πάτημα του κουμπιού που υπάρχει στην πρώτη οθόνη και αναγράφει, «Paired Devices». Για την δημιουργία αυτής της οθόνης χρησιμοποιήσαμε ένα Button, ένα Text view και ένα Lis view. Το list view μας επιτρέπει να παρουσιάζει τις συσκευές που έχει ανιχνεύσει η εφαρμογή. Το button, ενεργοποιεί την ανίχνευση των συσκευών όταν το επιλέγει ο χρήστης, ενώ το text view χρησιμοποιείται για να εμφανίζει το μήνυμα «Paired Devices». Στις παρακάτω φωτογραφίες απεικονίζονται οι οθόνες που αναλύθηκαν παραπάνω. Σε αυτό το κομμάτι θα πρέπει να ενεργοποιήσουμε την κατασκευή που περιέχει τον μικροελεγκτή για να μπορέσει η εφαρμογή να συνδεθεί με αυτήν.



Εικόνα 15

Αφού έχουν πραγματοποιηθεί οι παραπάνω ενέργειες η εφαρμογή θα μας παρουσιάσει τις ανιχνεύσιμες συσκευές που υπάρχουν στον χώρο που βρισκόμαστε. Εκτός από αυτές θα υπάρχουν και οι συσκευές που έχει πραγματοποιηθεί πρόσφατα σύνδεση. Σε αυτό το σημείο ο χρήστης θα πρέπει να επιλέξει την συσκευή με την οποία θέλει να πραγματοποιήσει σύνδεση πατώντας επάνω στο το όνομα της, η παρουσίαση των συσκευών πραγματοποιείται στο List View. (βλέπε παράρτημα κώδικα προγραμματισμού I, σελ.: 46)



Εικόνα 16

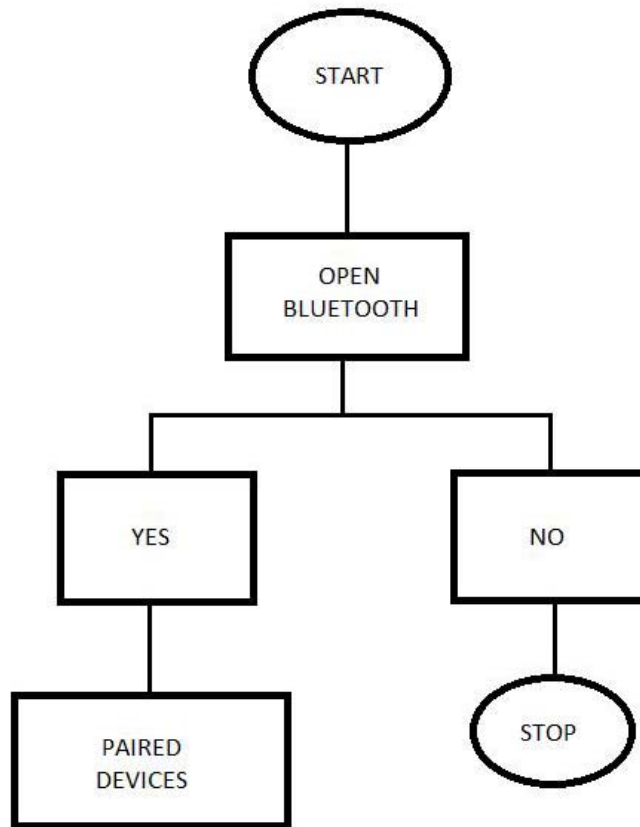
Αφού πραγματοποιηθεί με επιτυχία η σύνδεση με την συσκευή, η εφαρμογή θα μεταβεί σε μια οθόνη που θα μας επιτρέπει να χειριστούμε μέσω αυτής την συσκευή που είναι συνδεδεμένη στον μικροελεγκτή, είτε αυτή είναι κάμερα ασφαλείας, είτε μια τηλεκατευθυνόμενο κατασκευή. Εμείς επιλέξαμε να διαχειριζόμαστε δύο led. Μέσο της εφαρμογής θα μπορούμε να το ενεργοποιούμε και να το απενεργοποιούμε αλλά και να ελέγχουμε την φωτεινότητα τους. Για αυτό τον λόγο η επόμενη οθόνη στην οποία θα μεταβαίνουμε ύστερα από την σύνδεση περιέχει 4 Buttons για την ενεργοποίηση και απενεργοποίηση των led, καθώς και δύο seek Bar για την διαχείριση της φωτεινότητας, σε αυτό το σημείο της εφαρμογής έχουμε προσθέσει και δύο Lumn (text view) στα οποία θα παρουσιάζεται το ποσοστό φωτεινότητας του led. Εκτός από αυτά περιέχει 2 Text View τα οποία βοηθούν τον χρήστη να κατανοεί τις λειτουργίες που πραγματοποιούνται σε αυτήν την οθόνη. Επίσης έχουμε προσθέσει άλλο ένα Button για την άμεση αποσύνδεση της εφαρμογής από τον συσκευή. Στην παρακάτω εικόνα παρουσιάζεται η δεύτερη οθόνη για την διαχείριση των led. (βλέπε παράρτημα κώδικα προγραμματισμού II, σελ.: 47)



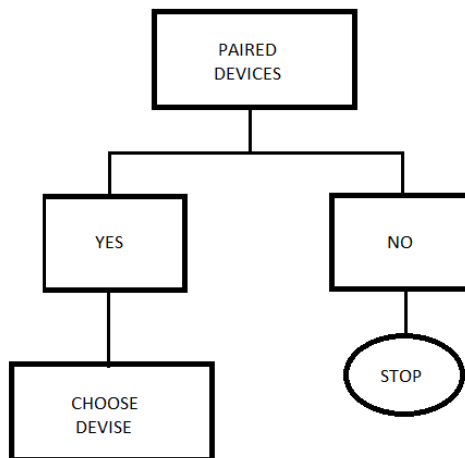
Εικόνα 17

Στην παραπάνω ανάλυση, περιγράφεται η εμφάνιση της εφαρμογής. Η εφαρμογή ακόμα δεν πραγματοποιεί τις ενέργειες που πρέπει. Για να το πετύχουμε αυτό, πρέπει να γράψουμε κώδικα για την κάθε οθόνη της εφαρμογής μας που θα προσδιορίζει στην εφαρμογή, τι ενέργειες θα πραγματοποιεί όταν ο χρήστης θα πατάει τις επιλογές στην κάθε οθόνη.

Θα ξεκινήσουμε με την πρώτη, στην οποία γίνεται ανίχνευση των συσκευών και μετά η επιλογή της συσκευής με την οποία επιθυμούμε να συνδεθούμε. Η εγγραφή του κώδικα πραγματοποιείται στην καρτέλα DeviceList.java. Ο κώδικας αυτός αφορά την εύρεση αλλά και την σύνδεση της εφαρμογής με την κατασκευή η οποία συνδέεται μέσω Bluetooth. (βλέπε παράρτημα κώδικα προγραμματισμού ΙΙΙ, σελ.: 50)

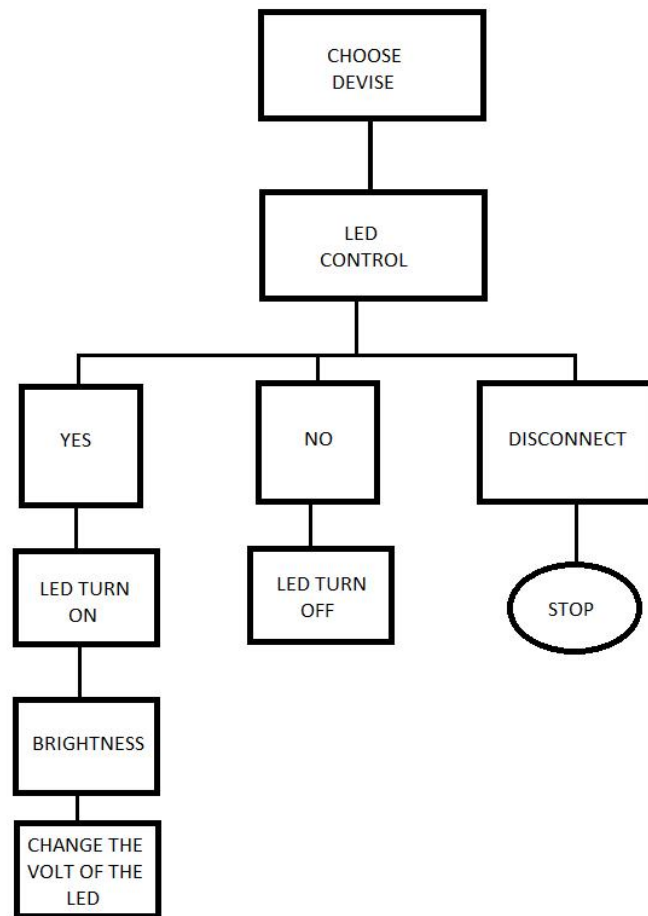


Λογικό διάγραμμα για την ενεργοποίηση της κεραίας Bluetooth



Λογικό διάγραμμα για την σύνδεση με την κατασκευή μέσω Bluetooth

Η ίδια διαδικασία πρέπει να πραγματοποιηθεί και για την δεύτερη οθόνη η οποία αφορά την διαχείριση των led. Σε αυτόν τον κώδικα προγραμματίζουμε τα κουμπιά On, Off, On1, Off1 και της αποσύνδεσης. Επίσης και τις μπάρες κύλισης που ρυθμίζουν την φωτεινότητα. Ο κώδικας αυτός γράφεται στην καρτέλα LedControl.java. (βλέπε παράρτημα κώδικα προγραμματισμού IV, σελ.: 53)



Λογικό διάγραμμα για την διαχείριση του LED

Αφού έχουμε πραγματοποιήσει αυτά τα στάδια πλέον η εφαρμογή χρειάζεται έναν τελευταίο κώδικα για να είναι έτοιμη να χρησιμοποιηθεί από τον χρήστη. Βασικό κομμάτι στον προγραμματισμό μιας εφαρμογής είναι και πραγματοποίηση του κώδικα για τις παραμέτρους, στην καρτέλα String.xml, όπως και οι παράμετροι στο AndroidManifest.xml. Στο τομέα των παραμέτρων πραγματοποιούμε τις κατάλληλες ρυθμίσεις για την κεραία Bluetooth της συσκευής στην οποία είναι εγκατεστημένη η εφαρμογή. (βλέπε παράρτημα κώδικα προγραμματισμού V, σελ.: 58)



Μικροελεγκτής

Πριν αναλύσουμε την πλακέτα με την οποία συνεργάζεται η εφαρμογή μας, πρέπει να αναλύσουμε το πιο σημαντικό κομμάτι αυτής, τον μικροελεγκτή.

Ο μικροελεγκτής (microcontroller) είναι ένας τύπος επεξεργαστή, ουσιαστικά μια παραλλαγή μικροεπεξεργαστή, ο οποίος μπορεί να λειτουργήσει με ελάχιστα εξωτερικά εξαρτήματα, λόγω των πολλών ενσωματωμένων υποσυστημάτων που διαθέτει. Χρησιμοποιείται ευρύτατα σε όλα τα ενσωματωμένα συστήματα (embedded systems) ελέγχου χαμηλού και μεσαίου κόστους, όπως αυτά που χρησιμοποιούνται σε αυτοματισμούς, ηλεκτρονικά καταναλωτικά προϊόντα (από ψηφιακές φωτογραφικές μηχανές έως παιχνίδια), ηλεκτρικές συσκευές και κάθε είδους αυτοκινούμενα τροχοφόρα οχήματα.

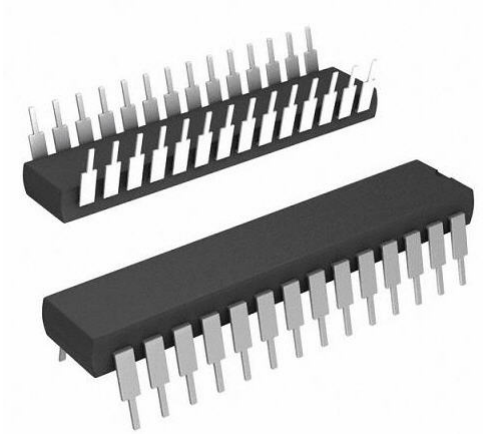
Πλεονεκτήματα Μικροελεγκτών

- Αυτονομία, μέσω της ενσωμάτωσης σύνθετων περιφερειακών υποσυστημάτων όπως μνήμες και θύρες επικοινωνίας. Έτσι πολλοί μικροελεγκτές δεν χρειάζονται κανένα άλλο ολοκληρωμένο κύκλωμα για να λειτουργήσουν.
- Η ενσωμάτωση περιφερειακών σημαίνει ευκολότερη υλοποίηση εφαρμογών λόγω των απλούστερων διασυνδέσεων. Επίσης, οδηγεί σε χαμηλότερη κατανάλωση ισχύος, μεγιστοποιώντας τη φορητότητα και ελαχιστοποιεί το κόστος της συσκευής στην οποία ενσωματώνεται ο μικροελεγκτής.
- Χαμηλό κόστος.
- Μεγαλύτερη αξιοπιστία, και πάλι λόγω των λιγότερων διασυνδέσεων.
- Μειωμένες εκπομπές ηλεκτρομαγνητικών παρεμβολών και μειωμένη ευαισθησία σε αντίστοιχες παρεμβολές από άλλες ηλεκτρικές και ηλεκτρονικές συσκευές. Το πλεονέκτημα αυτό προκύπτει από το μικρότερο αριθμό και μήκος εξωτερικών διασυνδέσεων καθώς και τις χαμηλότερες ταχύτητες λειτουργίας.
- Περισσότεροι διαθέσιμοι ακροδέκτες για ψηφιακές εισόδους-εξόδους, λόγω της μη δέσμευσής τους για τη σύνδεση εξωτερικών περιφερειακών
- Μικρό μέγεθος συνολικού υπολογιστικού συστήματος.

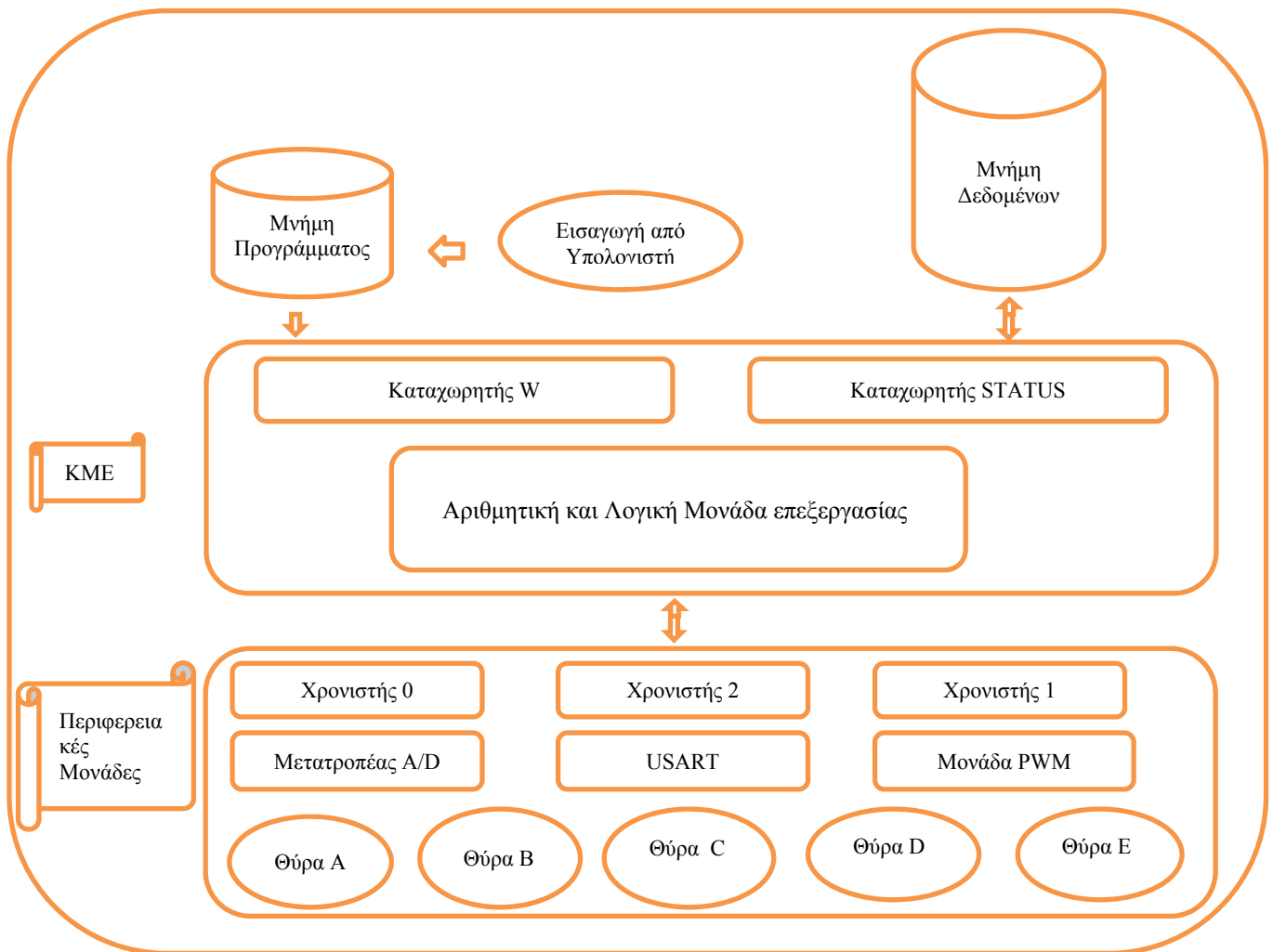
Ανάλογα με την εφαρμογή για την οποία προορίζεται ένας μικροελεγκτής, μπορεί να περιέχει και:

- Μία ή περισσότερες ασύγχρονες σειριακές θύρες επικοινωνίας (Universal Asynchronous Receiver Transmitter, UART).
Σύγχρονες σειριακές θύρες επικοινωνίας (πχ I2C, SPI, Ethernet).
- Ολόκληρα υποσυστήματα για την άμεση υποστήριξη από υλικολογισμικό (firmware) των πιο σύνθετων πρωτοκόλλων επικοινωνίας όπως CAN, HDLC, ISDN, ADSL.
- Μονάδα άμεσης εκτέλεσης πράξεων κινητής υποδιαστολής (Floating Point Processing Unit, FPU), η οποία είναι πάντοτε πιο γρήγορη από την ALU του επεξεργαστή. Τέτοιες μονάδες χαρακτηρίζουν τους μικροελεγκτές με δυνατότητες ψηφιακής επεξεργασίας σήματος (Digital Signal Processing, DSP). Τα τελευταία χρόνια, με την ευρύτατη διάδοση των φορητών συσκευών ήχου και εικόνας, παρατηρείται μια τάση σύγκλισης των μικροελεγκτών με τους DSP.[4]
- Περισσότερες από μία εισόδους για μετατροπή αναλογικού σήματος σε ψηφιακό (Analog to Digital converter, ADC).
- Μετατροπέα ψηφιακού σε αναλογικό σήμα (Digital to Analog converter, DAC).
- Ελεγκτή οθόνης υγρών κρυστάλλων (Liquid Crystal Display, LCD).
- Υποσύστημα προγραμματισμού πάνω στο κύκλωμα (τύπου ISP, βλ. παραπάνω). Χάρη σε αυτό το κύκλωμα, είναι δυνατός ο επαναπρογραμματισμός (αναβάθμιση λογισμικού) της εφαρμογής, συνδέοντας στη συσκευή μια εξωτερική συσκευή προγραμματισμού (συνήθως σε θύρα UART RS-232) ή ακόμη και από το διαδίκτυο. Αυτή η δυνατότητα απαιτεί την προϋπαρξη λογισμικού υποδοχής (bootstrap) μέσα στη μνήμη προγράμματος και επομένως δεν μπορεί να γίνει σε τελείως άδεια μνήμη προγράμματος.
- Υποσύστημα προγραμματισμού (τύπου ISP) και διάγνωσης (συνήθως είναι το καθιερωμένο πρότυπο JTAG). Χάρη σε αυτό, είναι δυνατός ο προγραμματισμός της μνήμης προγράμματος χωρίς να προαπαιτείται κάποιο πρόγραμμα υποδοχής. Γι αυτό το λόγο, είναι ιδιαίτερα χρήσιμο στον αρχικό προγραμματισμό, πχ κατά τη συναρμολόγηση, ή σε περίπτωση σφάλματος (bug) στο λογισμικό υποδοχής το οποίο να καθιστά αδύνατη την κανονική αναβάθμιση.

Όπως μπορούμε να διακρίνουμε από την παρακάτω εικόνα ένας μικροελεγκτής αποτελείται από πολλές συνδέσεις (ποδαράκια μικροελεγκτή). Σε αυτές πραγματοποιείται η σύνδεση του με τα επιμέρους εξαρτήματα.



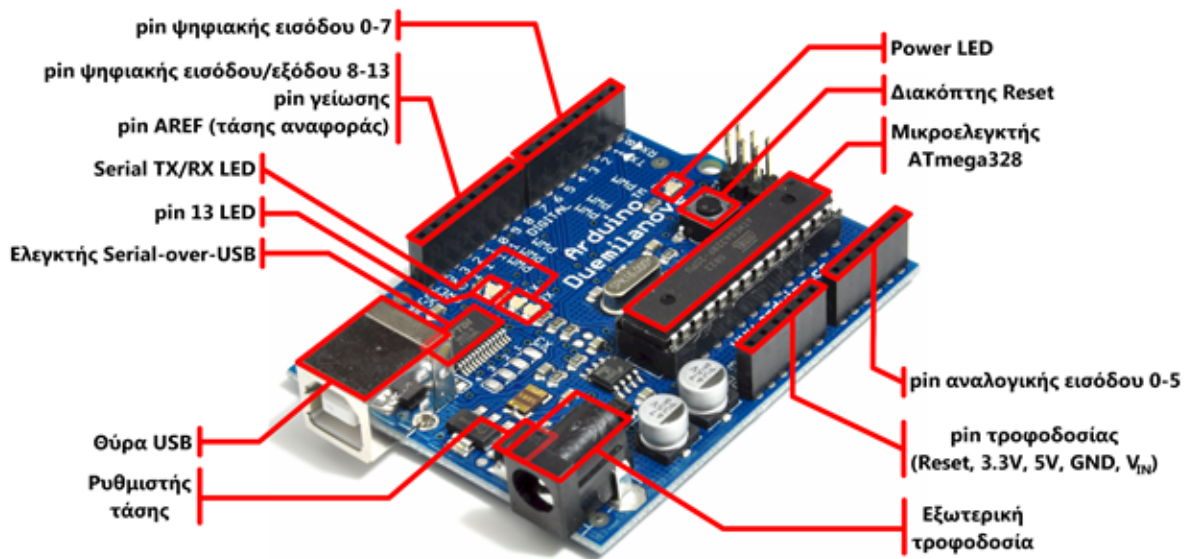
Στο παρακάτω σχήμα απεικονίζεται η αρχιτεκτονική του μικροελεγκτή.



Arduino

Το Arduino είναι μια μητρική πλακέτα ανοικτού κώδικα, με ενσωματωμένο μικροελεγκτή και αναλογικές και ψηφιακές εισόδους και εξόδους. Την συγκεκριμένη πλακέτα μπορούμε να την προγραμματίσουμε με τη γλώσσα Wiring, η οποία βασίζεται στην γλώσσα προγραμματισμού C++. Το Arduino μπορεί να χρησιμοποιηθεί για την ανάπτυξη ανεξάρτητων διαδραστικών αντικειμένων αλλά και να συνδεθεί με υπολογιστή. Το πρόγραμμα με το οποίο προγραμματίζεται η πλακέτα Arduino, διατίθεται δωρεάν στο διαδίκτυο στην ιστοσελίδα www.arduino.cc.

Μία πλακέτα Arduino αποτελείται από ένα μικροελεγκτή Atmel AVR και συμπληρωματικά εξαρτήματα για την διευκόλυνση του χρήστη στον προγραμματισμό και την ενσωμάτωση του σε άλλα κυκλώματα. Όλες οι πλακέτες περιλαμβάνουν ένα γραμμικό ρυθμιστή τάσης 5V και έναν κρυσταλλικό ταλαντωτή 16MHz. Ο μικροελεγκτής είναι από κατασκευής προγραμματισμένος με ένα bootloader, έτσι ώστε να μην χρειάζεται εξωτερικός προγραμματιστής. Η πλακέτα Arduino διαθέτει microcontroller I/O pins για χρήση από άλλα κυκλώματα. Το Arduino Uno που χρησιμοποιούμε εμείς μας παρέχει 14 ψηφιακά I/O pins, έξι από τα οποία μπορούν να παράγουν διαμορφωμένα σήματα, και έξι αναλογικά δεδομένα.



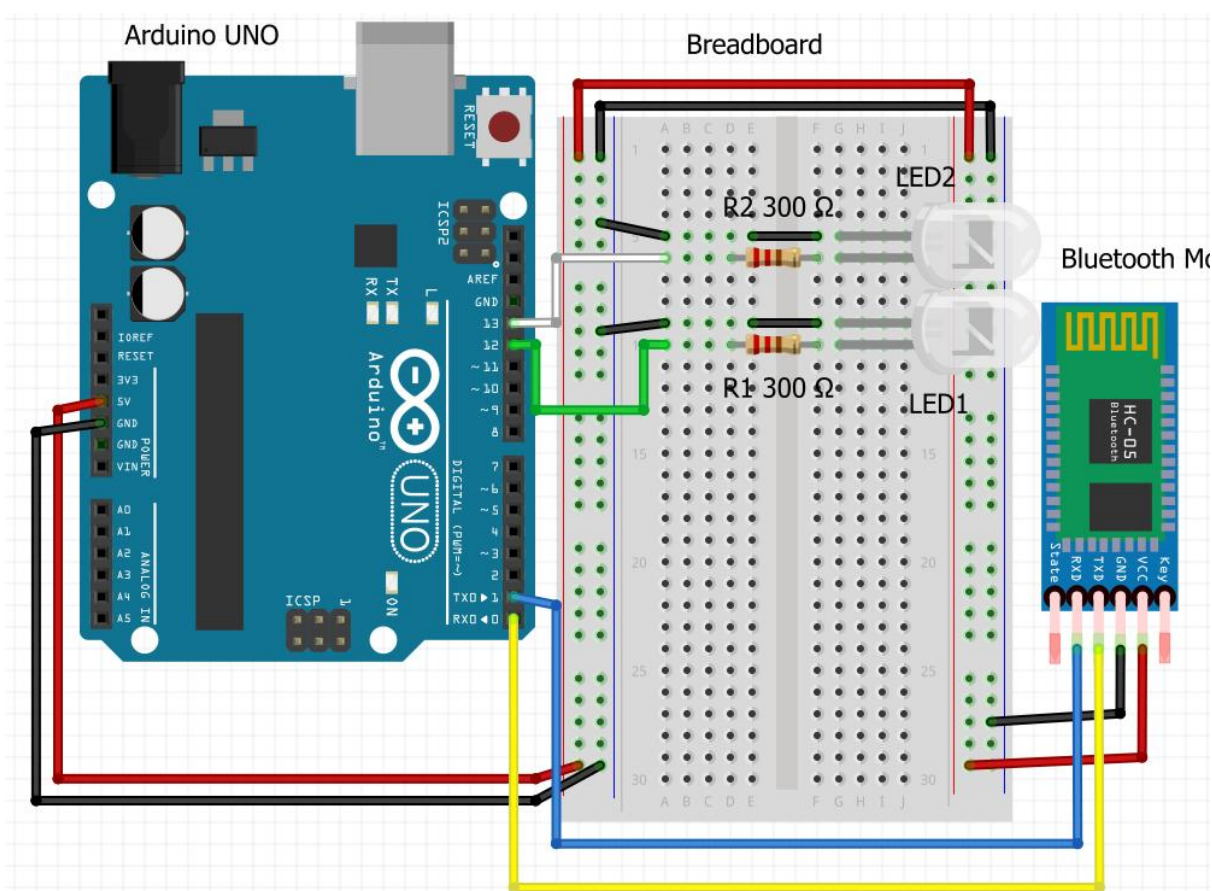
Εικόνα 18

Συνδεσμολογία

Για την κατασκευή της πλακέτας μας χρησιμοποιήσαμε ένα breadboard, δύο λυχνίες led, 2 αντιστάσεις 300 Ω, ένα Bluetooth Module και μια πλακέτα Arduino Uno.

Οι δύο λυχνίες led συνδέθηκαν στις ψηφιακές εξόδους 13 και 12 αφού παρεμβάλαμε μεταξύ τους από μία αντίσταση 300 Ω για την προστασία της λυχνίας. Από τις υποδοχές +5V και GND του Arduino, τροφοδοτήσαμε το Breadboard με τάση 5 volt. Το Bluetooth Module το τροφοδοτούμε με τάση 5 volt, από το breadboard και με 2 καλώδια συνδέουμε την υποδοχές του RX και TX με τις υποδοχές του Arduino TX και RX αντίστοιχα. Βασικό κομμάτι για την κατασκευή είναι η τροφοδοσία της πλακέτας. Αυτό μπορούμε να το επιτύχουμε με την σύνδεση ενός καλωδίου USB στην αντίστοιχη θύρα της πλακέτας ή με την σύνδεση της με τροφοδοτικό το οποίο θα της παρέχει τροφοδοσία 5 volt.

Στην παρακάτω εικόνα παρουσιάζεται η συνδεσμολογία της κατασκευής μας.



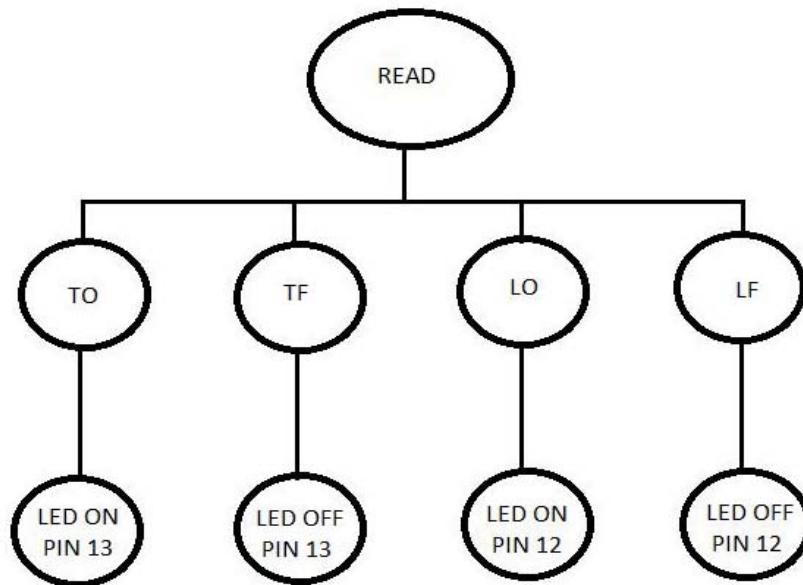
Εικόνα 19

Προγραμματισμός Πλακέτας Arduino

Για να μπορεί η κατασκευή μας να λειτουργεί με την εφαρμογή που έχουμε δημιουργήσει για κινητές συσκευές θα πρέπει να προγραμματίσουμε την πλακέτα Arduino, για τις ενέργειες που θα πραγματοποιεί όταν θα πατάμε τα κουμπιά της εφαρμογής. Όταν στην εφαρμογή πατάμε τα κουμπιά ON-LED1 και OFF-LED1 απευθυνόμαστε στο Led το οποίο είναι συνδεδεμένο στην ψηφιακή έξοδο 12. Δηλαδή η εφαρμογή στέλνει μέσω Bluetooth τους χαρακτήρες LO και LF για ενεργοποίηση και απενεργοποίηση αντίστοιχα. Όταν ο μικροελεγκτής που βρίσκεται στην πλακέτα Arduino διαβάσει τους χαρακτήρες αυτούς με την βοήθεια του Bluetooth Module, τότε θα ενεργοποιεί ή θα απενεργοποιεί την έξοδο 12 στην οποία έχουμε συνδεδεμένο το Led 1. Αντίστοιχα λειτουργεί και το Led 2, όμως σε αυτή την περίπτωση όταν πατάμε τα κουμπιά ON-LED2 και OFF-LED2 η εφαρμογή στέλνει τους χαρακτήρες TO και TF για ενεργοποίηση και απενεργοποίηση αντίστοιχα, και ο μικροελεγκτής τότε θα ενεργοποιεί ή θα απενεργοποιεί την έξοδο 13 στην οποία έχουμε συνδεδεμένο το Led 2.

Για τον προγραμματισμό της πλακέτας Arduino, χρησιμοποιούμε την εφαρμογή που διατίθεται δωρεάν στο διαδίκτυο και είναι βασισμένη στην γλώσσα προγραμματισμού C++. (βλέπε παράρτημα κώδικα προγραμματισμού VI, σελ.: 59)

Στην παρακάτω εικόνα απεικονίζεται το λογικό διάγραμμα, της διαδικασίας που θα πραγματοποιεί ο μικροελεγκτής της κατασκευής.



Λογικό διάγραμμα για τον μικροελεγκτή



Εικόνα 20

Σκοπός

Ο σκοπός που επιλέξαμε το συγκεκριμένο θέμα είναι η ραγδαία ανάπτυξη του αυτοματισμού σήμερα. Όμως σημαντική εξέλιξη παρουσιάζεται στους αυτοματισμούς οικιακής χρήσης για αυτό και ο σκοπός λειτουργίας της παραπάνω πλακέτας βασίζεται πάνω σε αυτούς τους αυτοματισμούς. Δηλαδή η πλακέτα συνδέεται, μέσω Bluetooth, με την εφαρμογή που «τρέχει» σε συσκευή με λειτουργικό Android και μας επιτρέπει τον έλεγχο των δύο Led τα οποία είναι συνδεδεμένα στην πλακέτα. Στο κύκλωμα της πλακέτας υπάρχει ο μικροελεγκτής ο οποίος είναι προγραμματισμένος αλλά και συνδεδεμένος έτσι ώστε να μπορεί να δίνει τις κατάλληλες εντολές στα Led. Μέσω της συσκευής ο χρήστης μπορεί να ενεργοποιήσει και να απενεργοποιήσει τα led, αλλά και να ελέγξει την φωτεινότητα του. Η επιλογή της σύνδεσης Bluetooth μας επιτρέπει τον χειρισμό από κοντινή επαφή. Σε περίπτωση που θέλουμε, ο έλεγχος να πραγματοποιείται από μεγαλύτερη απόσταση, υπάρχει η δυνατότητα σύνδεσης του μικροελεγκτή μέσω ιντερνέτ, με την χρήση κεραίας wi-fi και να μπορεί ο χρήστης να μπαίνει σε server όπου και αν είναι, για τον έλεγχο της συσκευής ή των συσκευών που ελέγχει ο μικροελεγκτής. Το κύκλωμα αυτό μας δίνει την δυνατότητα να συνδέσουμε στην πλακέτα διάφορα εξαρτήματα τα οποία θα έχουμε την δυνατότητα να τα ελέγχουμε μέσω κάποιας κινητής συσκευής. Τέτοια εξαρτήματα μπορεί να είναι κάμερες ασφαλείας, ρολά παραθύρων, γκαραζόπορτες κ.λπ..



Εικόνα 21

Παράρτημα Κώδικα Προγραμματισμού

I. Προγραμματισμός Εμφάνισης Πρώτης Οθόνης Εφαρμογής

Παρακάτω παρουσιάζεται ο κώδικας που χρησιμοποιούμε στην πρώτη οθόνη για την διαμόρφωση της. Δηλαδή ο προσανατολισμός των ListView, TextView και Button, επίσης οι ονομασίες τους, τα κείμενα που παρουσιάζονται, καθώς και οι χρωματισμοί της οθόνης αλλά και του κουμπιού.

```
<RelativeLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:paddingLeft="@dimen/activity_horizontal_margin"
    android:paddingRight="@dimen/activity_horizontal_margin"
    android:paddingTop="@dimen/activity_vertical_margin"
    android:paddingBottom="@dimen/activity_vertical_margin"
    tools:context=".DeviceList"
    android:background="#ffffffdd">

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Paired Devices"
        android:id="@+id/textView"
        android:layout_alignParentTop="true"
        android:layout_alignParentLeft="true"
        android:layout_alignParentStart="true" />

    <Button
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Paired Devices"
        android:id="@+id/button"
        android:layout_alignParentBottom="true"
        android:layout_alignParentLeft="true"
        android:layout_alignParentStart="true"
        android:layout_alignParentRight="true"
        android:layout_alignParentEnd="true"
        android:background="#ffffe306" />

    <ListView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:id="@+id/listView"
        android:layout_centerHorizontal="true"
        android:layout_above="@+id/button"
        android:layout_below="@+id/textView" />
</RelativeLayout>
```

II. Προγραμματισμός εμφάνισης Δεύτερης Οθόνης εφαρμογής

Παρακάτω παρουσιάζεται ο κώδικας που χρησιμοποιούμε στην δεύτερη οθόνη για την διαμόρφωση της. Δηλαδή ο προσανατολισμός των δύο (2) TextView, των δύο (2) seek Bar, των δύο (2) Lumn (text view) και των τριών (5) Buttons, επίσης οι ονομασίες τους, τα κείμενα που παρουσιάζονται, καθώς και οι χρωματισμοί της οθόνης αλλά και του κουμπιού.

```
<RelativeLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:paddingLeft="@dimen/activity_horizontal_margin"
    android:paddingRight="@dimen/activity_horizontal_margin"
    android:paddingTop="@dimen/activity_vertical_margin"
    android:paddingBottom="@dimen/activity_vertical_margin"
    tools:context="com.led.led.ledControl"
    android:background="#ffffffdd">

<TextView
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="LED Control"
    android:id="@+id/textView2"
    android:layout_alignParentTop="true"
    android:layout_alignParentLeft="true"
    android:layout_alignParentStart="true" />

<SeekBar
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:id="@+id/seekBar"
    android:max="255"
    android:progress="1"
    android:indeterminate="false"
    android:layout_below="@+id/textView3"
    android:layout_alignParentLeft="true"
    android:layout_alignParentStart="true"
    android:layout_alignParentRight="true"
    android:layout_alignParentEnd="true" />

<TextView
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:textAppearance="?android:attr/textAppearanceLarge"
    android:text="Brightness"
    android:id="@+id/textView3"
    android:layout_marginTop="42dp"
    android:layout_below="@+id/button3"
    android:layout_alignParentLeft="true"
    android:layout_alignParentStart="true" />
```

```

<Button
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="ON"
    android:id="@+id/button2"
    android:layout_marginTop="30dp"
    android:layout_below="@+id/textView2"
    android:layout_alignParentLeft="true"
    android:layout_alignParentStart="true"
    android:layout_alignRight="@+id/seekBar"
    android:layout_alignEnd="@+id/seekBar"
    android:background="#ffffe306" />

```

```

<Button
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="OFF"
    android:id="@+id/button3"
    android:layout_marginTop="15dp"
    android:layout_below="@+id/button2"
    android:layout_alignParentLeft="true"
    android:layout_alignParentStart="true"
    android:layout_alignRight="@+id/button2"
    android:layout_alignEnd="@+id/button2"
    android:background="#ffffe306" />

```

```

<Button
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Disconnect"
    android:id="@+id/button4"
    android:layout_alignParentBottom="true"
    android:layout_alignParentLeft="true"
    android:layout_alignParentStart="true"
    android:background="#ffffe306" />

```

```

<TextView
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:id="@+id/lumn"
    android:layout_alignTop="@+id/textView3"
    android:layout_toRightOf="@+id/button4"
    android:layout_marginLeft="6dp"
    android:layout_marginStart="6dp"
    android:layout_alignParentRight="true"
    android:layout_alignParentEnd="true"
    android:layout_above="@+id/seekBar" />

```

```

<TextView
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:textAppearance="?android:attr/textAppearanceSmall"
    android:text="Πτυχιακή Ορέση Νικόπουλου"
    android:id="@+id/textView4"
    android:layout_alignBottom="@+id/button4"
    android:layout_alignParentRight="true"
    android:layout_alignParentEnd="true" />

```

```

<Button
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="ON LED-2-"
    android:id="@+id/button5"
    android:layout_below="@+id/seekBar"
    android:background="#ffffe306"
    android:layout_alignRight="@+id/seekBar"
    android:layout_alignEnd="@+id/seekBar"
    android:layout_alignParentLeft="true"
    android:layout_alignParentStart="true" />

<Button
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="OFF LED-2-"
    android:id="@+id/button6"
    android:background="#ffffe306"
    android:layout_below="@+id/button5"
    android:layout_alignParentLeft="true"
    android:layout_alignParentStart="true"
    android:layout_marginTop="48dp"
    android:layout_alignRight="@+id/button5"
    android:layout_alignEnd="@+id/button5" />

<SeekBar
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:id="@+id/seekBar2"
    android:layout_below="@+id/textView5"
    android:layout_alignParentLeft="true"
    android:layout_alignParentStart="true"
    android:layout_alignRight="@+id/button6"
    android:layout_alignEnd="@+id/button6" />

<TextView
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:id="@+id/lumn1"
    android:layout_above="@+id/seekBar2"
    android:layout_alignParentRight="true"
    android:layout_alignParentEnd="true" />

<TextView
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:textAppearance="?android:attr/textAppearanceLarge"
    android:text="Brightness LED-2-"
    android:id="@+id/textView5"
    android:layout_below="@+id/button6"
    android:layout_alignParentLeft="true"
    android:layout_alignParentStart="true" />

</RelativeLayout>

```

III. Προγραμματισμός Λειτουργίας Πρώτης Οθόνης Εφαρμογής

```
package com.led.led;

import android.support.v7.app.ActionBarActivity;
import android.os.Bundle;
import android.view.Menu;
import android.view.MenuItem;
import android.content.Intent;
import android.view.View;
import android.widget.AdapterView;
import android.widget.AdapterView.OnItemClickListener;
import android.widget.ArrayAdapter;
import android.widget.Button;
import android.widget.ListView;
import android.bluetooth.BluetoothAdapter;
import android.bluetooth.BluetoothDevice;
import android.widget.TextView;
import android.widget.Toast;

import java.util.ArrayList;
import java.util.Set;

public class DeviceList extends ActionBarActivity
{
    //widgets
    Button btnPaired;
    ListView devicelist;
    //Bluetooth
    private BluetoothAdapter myBluetooth = null;
    private Set<BluetoothDevice> pairedDevices;
    public static String EXTRA_ADDRESS = "device_address";

    @Override
    protected void onCreate(Bundle savedInstanceState)
    {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_device_list);

        //Calling widgets
        btnPaired = (Button)findViewById(R.id.button);
        devicelist = (ListView)findViewById(R.id.listView);

        //if the device has bluetooth
        myBluetooth = BluetoothAdapter.getDefaultAdapter();

        if(myBluetooth == null)
        {
            //Show a msg. that the device has no bluetooth adapter
            Toast.makeText(getApplicationContext(), "Bluetooth Device Not
Available", Toast.LENGTH_LONG).show();

            //finish apk
            finish();
        }
        else if(!myBluetooth.isEnabled())
        {
            //Ask to the user turn the bluetooth on
            Intent turnBTON = new
Intent(BluetoothAdapter.ACTION_REQUEST_ENABLE);
            startActivityForResult(turnBTON, 1);
        }
    }
}
```



```

    }

    btnPaired.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View v)
        {
            pairedDevicesList();
        }
    });
}

private void pairedDevicesList()
{
    pairedDevices = myBluetooth.getBondedDevices();
    ArrayList list = new ArrayList();

    if (pairedDevices.size()>0)
    {
        for(BluetoothDevice bt : pairedDevices)
        {
            list.add(bt.getName() + "\n" + bt.getAddress()); //Get the
device's name and the address
        }
    }
    else
    {
        Toast.makeText(getApplicationContext(), "No Paired Bluetooth
Devices Found.", Toast.LENGTH_LONG).show();
    }

    final ArrayAdapter adapter = new
ArrayAdapter(this, android.R.layout.simple_list_item_1, list);
    devicelist.setAdapter(adapter);
    devicelist.setOnItemClickListener(myListClickListener); //Method
called when the device from the list is clicked
}

private AdapterView.OnItemClickListener myListClickListener = new
AdapterView.OnItemClickListener()
{
    public void onItemClick (AdapterView<?> av, View v, int arg2, long
arg3)
    {
        // Get the device MAC address, the last 17 chars in the View
String info = ((TextView) v).getText().toString();
String address = info.substring(info.length() - 17);

        // Make an intent to start next activity.
Intent i = new Intent(DeviceList.this, ledControl.class);

        //Change the activity.
i.putExtra(EXTRA_ADDRESS, address); //this will be received at
ledControl (class) Activity
startActivity(i);
    }
};
}

```

```

@Override
public boolean onCreateOptionsMenu(Menu menu)
{
    // Inflate the menu; this adds items to the action bar if it is
present.
    getMenuInflater().inflate(R.menu.menu_device_list, menu);
    return true;
}

@Override
public boolean onOptionsItemSelected(MenuItem item) {
    // Handle action bar item clicks here. The action bar will
    // automatically handle clicks on the Home/Up button, so long
    // as you specify a parent activity in AndroidManifest.xml.
    int id = item.getItemId();

    //noinspection SimplifiableIfStatement
    if (id == R.id.action_settings) {
        return true;
    }

    return super.onOptionsItemSelected(item);
}
}

```

IV. Προγραμματισμός Λειτουργίας Δεύτερης Οθόνης Εφαρμογής

```
package com.led.led;

import android.support.v7.app.ActionBarActivity;
import android.os.Bundle;
import android.view.Menu;
import android.view.MenuItem;
import android.bluetooth.BluetoothSocket;
import android.content.Intent;
import android.view.View;
import android.widget.Button;
import android.widget.SeekBar;
import android.widget.TextView;
import android.widget.Toast;
import android.app.ProgressDialog;
import android.bluetooth.BluetoothAdapter;
import android.bluetooth.BluetoothDevice;
import android.os.AsyncTask;

import java.io.IOException;
import java.util.UUID;

public class ledControl extends ActionBarActivity {

    Button btnOn, btnOff, btnDis, btnOn1, btnOff1;
    SeekBar brightness;
    TextView lumn;
    TextView lumn;
    String address = null;
    private ProgressDialog progress;
    BluetoothAdapter myBluetooth = null;
    BluetoothSocket btSocket = null;
    private boolean isBtConnected = false;
    //SPP UUID. Look for it
    static final UUID myUUID = UUID.fromString("00001101-0000-1000-8000-00805F9B34FB");

    @Override
    protected void onCreate(Bundle savedInstanceState)
    {
        super.onCreate(savedInstanceState);

        Intent newint = getIntent();
        address = newint.getStringExtra(DeviceList.EXTRA_ADDRESS);
        //receive the address of the bluetooth device

        //view of the ledControl
        setContentView(R.layout.activity_led_control);

        //call the widgtes
        btnOn = (Button)findViewById(R.id.button2);
        btnOff = (Button)findViewById(R.id.button3);
        btnDis = (Button)findViewById(R.id.button4);
        btnOn1 = (Button)findViewById(R.id.button5);
        btnOff1 = (Button)findViewById(R.id.button6);
        brightness = (SeekBar)findViewById(R.id.seekBar);
        brightness1 = (SeekBar)findViewById(R.id.seekBar2)
```

```

    lumn = (TextView)findViewById(R.id.lumn);
    lumn1 = (TextView)findViewById(R.id.lumn2);

    new ConnectBT().execute(); //Call the class to connect

    //commands to be sent to bluetooth
    btnOn.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View v) {
            turnOnLed(); //method to turn on led 1
        }
    });

    btnOff.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View v) {
            turnOffLed(); //method to turn off led 1
        }
    });

    btnOn1.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View v) {
            turnOnLed1(); //method to turn on led 2
        }
    });

    btnOff1.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View v) {
            turnOffLed1(); //method to turn off led 2
        }
    });
    btnDis.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View v) {
            Disconnect(); //close connection
        }
    });

    brightness.setOnSeekBarChangeListener(new
SeekBar.OnSeekBarChangeListener() {
        @Override
        public void onProgressChanged(SeekBar seekBar, int progress,
boolean fromUser) {
            if (fromUser == true) {
                lumn.setText(String.valueOf(progress));
                try {

    btSocket.getOutputStream().write(String.valueOf(progress).getBytes());
                } catch (IOException e) {

                }
            }
        }

        @Override
        public void onStartTrackingTouch(SeekBar seekBar) {

```

```

    }

    @Override
    public void onStopTrackingTouch(SeekBar seekBar) {

    }

    });

    brightness1.setOnSeekBarChangeListener(new
SeekBar.OnSeekBarChangeListener() {
    @Override
    public void onProgressChanged(SeekBar seekBar, int progress,
boolean fromUser) {
        if (fromUser == true) {
            lumn1.setText(String.valueOf(progress));
            try {

btSocket.getOutputStream().write(String.valueOf(progress).getBytes());
                } catch (IOException e) {

                }

            }

        }

    }

    @Override
    public void onStartTrackingTouch(SeekBar seekBar) {

    }

    @Override
    public void onStopTrackingTouch(SeekBar seekBar) {

    }

    });
}

private void Disconnect() {
    if (btSocket != null) //If the btSocket is busy
    {
        try {
            btSocket.close(); //close connection
        } catch (IOException e) {
            msg("Error");
        }
    }
    finish(); //return to the first layout
}

private void turnOffLed() {
    if (btSocket != null) {
        try {

btSocket.getOutputStream().write("LF".toString().getBytes());
            } catch (IOException e) {
                msg("Error");
            }

        }

    }

    private void turnOnLed() {
        if (btSocket != null) {

```

```

        try {
btSocket.getOutputStream().write("LO".toString().getBytes());
        } catch (IOException e) {
            msg("Error");
        }
    }

    private void turnOffLed1() {
        if (btSocket != null) {
            try {

btSocket.getOutputStream().write("TF".toString().getBytes());
                } catch (IOException e) {
                    msg("Error");
                }
            }
        }

    private void turnOnLed1() {
        if (btSocket != null) {
            try {

btSocket.getOutputStream().write("TO".toString().getBytes());
                } catch (IOException e) {
                    msg("Error");
                }
            }
        }

        // fast way to call Toast
        private void msg(String s) {
            Toast.makeText(getApplicationContext(), s,
Toast.LENGTH_LONG).show();
        }

        @Override
        public boolean onCreateOptionsMenu(Menu menu) {
            // Inflate the menu; this adds items to the action bar if it is
present.
            getMenuInflater().inflate(R.menu.menu_led_control, menu);
            return true;
        }

        @Override
        public boolean onOptionsItemSelected(MenuItem item) {
            // Handle action bar item clicks here. The action bar will
            // automatically handle clicks on the Home/Up button, so long
            // as you specify a parent activity in AndroidManifest.xml.
            int id = item.getItemId();

            //noinspection SimplifiableIfStatement
            if (id == R.id.action_settings) {
                return true;
            }

            return super.onOptionsItemSelected(item);
        }

        private class ConnectBT extends AsyncTask<Void, Void, Void> // UI

```

```

thread
{
    private boolean ConnectSuccess = true; //if it's here, it's
almost connected

    @Override
    protected void onPreExecute() {
        progress = ProgressDialog.show(ledControl.this,
"Connecting...", "Please wait!!!"); //show a progress dialog
    }

    @Override
    protected Void doInBackground(Void... devices) //while the
progress dialog is shown, the connection is done in background
    {
        try {
            if (btSocket == null || !isBtConnected) {
                myBluetooth =
BluetoothAdapter.getDefaultAdapter();//get the mobile bluetooth device
                BluetoothDevice dispositivo =
myBluetooth.getRemoteDevice(address);//connects to the device's address and
checks if it's available
                btSocket =
dispositivo.createInsecureRfcommSocketToServiceRecord(myUUID);//create a
RFCOMM (SPP) connection

BluetoothAdapter.getDefaultAdapter().cancelDiscovery();
                btSocket.connect();//start connection
            }
        } catch (IOException e) {
            ConnectSuccess = false;//if the try failed, you can
check the exception here
        }
        return null;
    }

    @Override
    protected void onPostExecute(Void result) //after the
doInBackground, it checks if everything went fine
    {
        super.onPostExecute(result);

        if (!ConnectSuccess) {
            msg("Connection Failed. Is it a SPP Bluetooth? Try
again.");
            finish();
        } else {
            msg("Connected.");
            isBtConnected = true;
        }
        progress.dismiss();
    }
}
}
}

```

V. Προγραμματισμός Παραμέτρων Εφαρμογής

String.xml

```
<?xml version="1.0" encoding="utf-8"?>
<resources>
<string name="app_name">LED</string>

<string name="hello_world">Hello world!</string>
<string name="action_settings">Settings</string>
<string name="title_activity_led_control">ledControl</string>
</resources>
```

AndroidManifest.xml

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.led.led" >
    <uses-permission android:name="android.permission.BLUETOOTH_ADMIN"/>
    <uses-permission android:name="android.permission.BLUETOOTH"/>

    <application
        android:allowBackup="true"
        android:label="@string/app_name"
        android:theme="@style/AppTheme" >
        <activity
            android:name=".DeviceList"
            android:label="@string/app_name" >
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />

                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>
        <activity
            android:name=".ledControl"
            android:label="LED Control" >
        </activity>
    </application>

</manifest>
```


VI. Προγραμματισμός Arduino

```
char data = 0;
char command;
String string;
#define led 12
#define led1 13

void setup()
{
  Serial.begin(9600);
  pinMode(led, OUTPUT);
  pinMode(led1, OUTPUT);
}
void loop()
{
  if (Serial.available()>0)
  {string= "";}
  while(Serial.available()>0)
  {
    command = ((byte)Serial.read());
    if(command== ':')
    {break;
    }
    else
    {
      string+= command;
    } }
  if(string=="TO")
  {
    LEDOn();
  }
  if(string=="TF")
  {
    LEDOff();
  }
  if(string=="LO")
  {
    LEDOn1();
  }
  if(string=="LF")
  {
    LEDOff1();
  }
}
void LEDOn()
{
  digitalWrite(led, HIGH);
}
void LEDOff()
{
  digitalWrite(led, LOW);
}
void LEDOn1()
{
  digitalWrite(led1, HIGH);
}
void LEDOff1()
{
  digitalWrite(led1, LOW);
}
```

Κατάλογος Εικόνων

- **EIKONA 1** : Ποσοστό προτίμησης σύμφωνα με το λειτουργικό.
- **EIKONA 2** : Λογότυπο του λειτουργικού android.
- **EIKONA 3** : Λογότυπο του λειτουργικού Android σύμφωνα με την έκδοση.
- **EIKONA 4** : Λογότυπο του λειτουργικού της Apple σύμφωνα με την έκδοση.
- **EIKONA 5** : Αρχική οθόνη κινητού με λογισμικό Android και Iphone.
- **EIKONA 6** : Λογότυπο των ηλεκτρονικών καταστημάτων αγοράς εφαρμογών App Store και Google Play
- **EIKONA 7** : Προτίμηση εφαρμογών σύμφωνα με την κατηγορία.
- **EIKONA 8** : Λογότυπο σύνδεσης Bluetooth.
- **EIKONA 9** : Λογότυπο Bluetooth 4.
- **EIKONA 10** : Bluetooth Module για συσκευές της Apple.
- **EIKONA 11** : Λογότυπο της σύνδεσης NFC.
- **EIKONA 12** : Λογότυπο της γλώσσας προγραμματισμού C++.
- **EIKONA 13** : Λογότυπο της γλώσσας προγραμματισμού Java.
- **EIKONA 14** : Λογότυπο της εφαρμογής Eclipse.
- **EIKONA 15** : Απεικόνιση εφαρμογής κατά το άνοιγμα της.
- **EIKONA 16** : Απεικόνιση εφαρμογής για σύνδεση με συσκευές.
- **EIKONA 17** : Απεικόνιση εφαρμογής στην οθόνη διαχείρισης του LED.
- **EIKONA 18** : Απεικόνιση της πλακέτας Arduino Uno.
- **EIKONA 19** : Απεικόνιση της συνδεσμολογίας της κατασκευής.
- **EIKONA 20** : Λογότυπο Arduino Wiring
- **EIKONA 21** : Έξυπνα σπίτια

Βιβλιογραφία

1. Ιστοσελίδες

- 1 [I1]: <http://el.dbpedia.org/page/Λογισμικο>
- 2 [I2]: <http://el.wikipedia.org/wiki/Λογισμικο>
- 3 [I3]: <http://el.wikipedia.org/wiki/NFC>

2. Άρθρα

- 1 [A1]: <http://pcmagazino.blogspot.gr/2016/03/android-1/html>
- 2 [A2]: www.myphone.gr/library/article-39.html
- 3 [A3]: <http://techmaniacs.gr/android-vs-ios/>

3. Βιβλία

- 1 [B1]: Lauren Darcey, Shane Conder. «*Μάθετε την ανάπτυξη εφαρμογών για το Android*» 2^η Έκδοση. Εκδόσεις Γιάννης Β. Σαμαράς.
- 2 [B2]: Paul Deitel, Harney Deitel. «*Java Προγραμματισμός*» 8^η Έκδοση. Εκδόσεις Μ.Γκούρδας.
- 3 [B3]: Δρ. Δ Κυτάγιας. «*C++ Προγραμματισμός*». Εκδόσεις Σύγχρονη Εκδοτική.
- 4 [B4]: Ιωάννης Χατζηλυγερούδης και Χρήστος Μακρής. «*Οντοκεντρικός Προγραμματισμός*». Διάλεξη στο Πανεπιστήμιο Πατρών, Τμήμα Μηχανικών Η/Υ και Πληροφορικής.

Ηλεκτρονική Μορφή Εργασίας