



**ΑΤΕΙ ΠΕΙΡΑΙΑ**  
**ΣΧΟΛΗ ΤΕΧΝΟΛΟΓΙΚΩΝ ΕΦΑΡΜΟΓΩΝ**  
**ΤΜΗΜΑ ΗΛΕΚΤΡΟΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΙΚΩΝ ΣΥΣΤΗΜΑΤΩΝ**



**ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ**  
**Μέτρηση Τάσης Δικτύου με arduino και εμφάνισή της σε android συσκευή μέσω bluetooth**

Κωνσταντίνος Σιάχρας

Εισηγητής: Ιωάννης Έλληνας, καθηγητής

90417

ΑΙΓΑΛΕΩ  
Μάιος 2014

ΒΙΒΛΙΟΘΗΚΗ  
ΤΕΙ ΠΕΙΡΑΙΑ

Μέτρηση Τάσης Δικτύου με arduino και εμφάνισή της σε android συσκευή μέσω bluetooth

Μέτρηση Τάσης Δικτύου με arduino και εμφάνισή της σε android συσκευή μέσω bluetooth

## **ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ**

**Μέτρηση Τάσης Δικτύου με arduino και εμφάνισής της σε android συσκευή μέσω bluetooth**

**Σιάχρας Κωνσταντίνος**

**A.M. 30417**

**Εισηγητής:**

**Ιωάννης Έλληνας , Καθηγητής**

**Εξεταστική Επιτροπή:**

**1ος , Καθηγητής**

**2ος , Καθηγητής**

**Ημερομηνία εξέτασης 21/05/2014**

Μέτρηση Τάσης Δικτύου με arduino και εμφάνισή της σε android συσκευή μέσω bluetooth

## **ΕΥΧΑΡΙΣΤΙΕΣ**

Η παρούσα πτυχιακή εργασία ήταν αποτέλεσμα δουλειάς πάνω σε τρία ενδιαφέρων γνωστικά αντικείμενα. Για την ολοκλήρωσή της χρειάστηκαν πολλές ώρες προγραμματισμού σε Java και σε μικροεπεξεργαστές καθώς και αρκετές γνώσεις των ηλεκτρονικών. Θέλω να ευχαριστήσω όλους τους καθηγητές και εκπαιδευτές της σχολής μας που τόσα χρόνια μας παρέχουν εκπαίδευση σε τόσα πολλά πεδία της πληροφορικής και των ηλεκτρονικών.

Μέτρηση Τάσης Δικτύου με arduino και εμφάνισή της σε android συσκευή μέσω bluetooth

## ΠΕΡΙΛΗΨΗ

Η παρούσα πτυχιακή ασχολείται με τις δυνατότητες από την ένωση δύο γνωστών εμπορικών πλατφορμών. Της ηλεκτρονικής πλατφόρμας arduino με την πλατφόρμα κινητών android (mobile platform). Θα ασχοληθούμε με την ανάγνωση σειριακών δεδομένων από ένα αισθητήρα ρεύματος (current sensor) συνδεδεμένο με έναν arduino και τη εμφάνιση του μέσα από μία εφαρμογή(application) σε μία ενεργή συσκευή android.(android enabled device). Η σύνδεσή τους θα γίνεται ασύρματα μέσω μιας σύνδεσης Bluetooth. Στο πρώτο κεφάλαιο θα αρχίσουμε με μία εισαγωγή για τις τεχνολογίες που χρησιμοποιούμε και στη συνέχεια θα δώσουμε έμφαση στις δύο βασικές πλατφόρμες της πτυχιακής καθώς και μια σύνοψη για τη τεχνολογία του Bluetooth, τέλος θα ενώσουμε αυτές τις τρεις τεχνολογίες ώστε να δούμε τις δυνατότητες που μας προσφέρουν.

ΕΠΙΣΤΗΜΟΝΙΚΗ ΠΕΡΙΟΧΗ: CS: Software engineering , electronics  
ΛΕΞΕΙΣ ΚΛΕΙΔΙΑ: android , arduino, bluetooth, current , open-source

Μέτρηση Τάσης Δικτύου με arduino και εμφάνισή της σε android συσκευή μέσω bluetooth



## Περιεχόμενα

1 : ΕΙΣΑΓΩΓΗ.....	13
1.1 Αντικείμενο Πτυχιακής.....	13
1.2 Ιστορική αναδρομή και γενικές πληροφορίες της πλατφόρμας android.....	13
1.3 Ιστορική αναδρομή και γενικές πληροφορίες πλατφόρμας arduino.....	14
1.4 Ιστορική αναδρομή και γενικές πληροφορίες τεχνολογίας Bluetooth.....	14
2: Arduino Electronic Platform.....	15
2.1 Arduino electronic platform.....	15
2.2 Arduino Hardware .....	18
2.3 Arduino Software.....	20
3 : Πλατφόρμα Android.....	21
3.1 Εισαγωγή.....	21
3.2 Η εξέλιξη του Android.....	22
3.3 Αρχιτεκτονική Android.....	24
3.4 Στο εσωτερικό μιας εφαρμογής του Android.....	29
3.5 Το αρχείο AndroidManifest.xml.....	29
3.6 Οι υπόλοιποι φάκελοι του project.....	30
3.7 Δομικά μέρη μιας εφαρμογής.....	30
4 : Τεχνολογία Bluetooth.....	33
4.1 Τι είναι το Bluetooth;.....	33
4.2 Bluetooth στο Android.....	34
4.3 Το Android.Bluetooth Package.....	35
5: Η υλοποίηση του κυκλώματος και ο Arduino κώδικας.....	36
5.1 Εισαγωγή.....	36
5.2 Συνδεσμολογία κυκλώματος.....	36
5.3 Ένωση όλων των παραπάνω.....	44
6 : ΥΛΟΠΟΙΗΣΗ ΤΟΥ ANDROID.....	49
6.1 Υλοποίηση της εφαρμογής στο Android.....	49
7: ΠΑΡΑΡΤΗΜΑ.....	53
7.1 Κώδικας της εφαρμογής.....	53
7.ΒΙΒΛΙΟΘΕΣΙΑ.....	69
ΥΠΟΔΕΙΓΜΑ ΡΑΧΗΣ.....	71
ΥΠΟΔΕΙΓΜΑ ΣΥΝΟΠΤΙΚΗΣ ΠΕΡΙΓΡΑΦΗΣ .....	73

## ΚΑΤΑΛΟΓΟΣ ΣΧΗΜΑΤΩΝ

- Σχήμα 1 : Arduino Uno
- Σχήμα 2 : Arduino Uno (κάτοψη)
- Σχήμα 3 : Arduino Χάρτης στοιχείων
- Σχήμα 4 : Atmega168/328 χάρτης ακίδων
- Σχήμα 5 : Arduino λογισμικό διεπαφής
- Σχήμα 6 : Android λογότυπο.
- Σχήμα 7 : Nexus One
- Σχήμα 8 : Android Εκδόσεις
- Σχήμα 9 : Ημερομηνίες έκδοσης των Android αναβαθμίσεων
- Σχήμα 10 : Αρχιτεκτονική Android
- Σχήμα 11 : Κύκλος ζωής μίας εφαρμογής
- Σχήμα 12 : Bluetooth λογότυπο
- Σχήμα 13 : Σφάλμα εξομείωσης του bluetooth στο Android
- Σχήμα 14 : Οι τάξεις του bluetooth
- Σχήμα 15 : Το κύκλωμα μας
- Σχήμα 15 : Διεπαφή της εφαρμογής μας
- Σχήμα 16 : Αισθητήρα ρεύματος ACS712
- Σχήμα 17: Συνδεσμολογία του αισθητήρα ρεύματος
- Σχήμα 18 : Ένταση προς τάση του αισθητήριου
- Σχήμα 19 : Θόρυβος ανα πυκνωτή του αισθητήριου
- Σχήμα 20 : Συνδεσμολογία του Bluetooth
- Σχήμα 21 : Στοιχεία εισαγωγής στον putty
- Σχήμα 22: Arduino Κώδικας που χρειάστηκε για bluetooth
- Σχήμα 23: LCD Button shield
- Σχήμα 24: Arduino Κώδικας που χρειάστηκε για LCD
- Σχήμα 25 : Κύκλωμα της πτυχιακής (schematic)
- Σχήμα 26 : Φωτογραφία κυκλώματος
- Σχήμα 27 : Διεπαφή της εφαρμογής μας κατά την εκκίνηση
- Σχήμα 28: Διεπαφή της εφαρμογής κατά τη χρήση της.
- Σχήμα 29 : Διεπαφή ανάπτυξης του android κώδικα(eclipse)

Μέτρηση Τάσης Δικτύου με arduino και εμφάνισή της σε android συσκευή μέσω bluetooth

## **ΚΑΤΑΛΟΓΟΣ ΠΙΝΑΚΩΝ**

Πίνακας 1 : Λίστα επίσημων arduino εκδόσεις και βασικά χαρακτηριστικά.

Πίνακας 2 : Κατάλογος ενώσεων του arduino(pins) με τα λοιπά εξαρτήματα.

Μέτρηση Τάσης Δικτύου με arduino και εμφάνισή της σε android συσκευή μέσω bluetooth

## ΣΥΝΤΟΜΟΓΡΑΦΙΕΣ

API – Application Programming Interface  
AVD Manager - Android Virtual Device Manager  
FTDI – Future Technology Devices International  
IEEE – Institute of Electrical and Electronics Engineers  
PWM – Pulse Width Modulation  
RX - Receiver  
RFCOMM – Radio Frequency Communication Port  
SDK – Software Development Kit  
SPP - Serial Port Profile  
TX – Transmitter  
TTL – Transistor Transistor Logic  
XMPP – Extensible Messaging and Presence Protocol  
WPAN - Wireless Personal Area Networks  
Wifi - Wireless Fidelity

## 1 : ΕΙΣΑΓΩΓΗ

Σε αυτό το κεφάλαιο αναλύεται το αντικείμενο της πτυχιακής καθώς γίνεται μία γενική πληροφόρηση και μια ιστορική αναδρομή για τις κύριες πλατφόρμες (android, arduino) καθώς για την τεχνολογία Bluetooth που χρησιμοποιήθηκαν για την επίτευξη της πτυχιακής.

### 1.1 Αντικείμενο Πτυχιακής

Αντικείμενο της πτυχιακής είναι ένα παράδειγμα των δυνατοτήτων της ένωση δύο γνωστών εμπορικών πλατφορμών. Της ηλεκτρονικής πλατφόρμας arduino με την πλατφόρμα κινητών android (mobile platform). Στο ηλεκτρονικό κομμάτι της πτυχιακής χρησιμοποιήσαμε ένα αισθητήρα ρεύματος(συγκεκριμένα τον ACS Breakout της sparkfun Electronics) ο οποίος διαβάζει σειριακά την ένταση της τάσης δικτύου και την αποστέλλει σε μία αναλογική είσοδο του arduino. Απο εκεί ο arduino αποστέλλει τα δεδομένα με μια bluetooth κεραία και τα αποδέχεται η android συσκευή μας για την εμφάνισή των δεδομένων στον χρήστη δια μέσω μιας εφαρμογής.

### 1.2 Ιστορική αναδρομή και γενικές πληροφορίες της πλατφόρμας android

Αρχικά αναπτύχθηκε από την Google και αργότερα από την Open Handset alliance. Η πρώτη παρουσίαση της πλατφόρμας Android έγινε στις 5 Νοεμβρίου 2007 παράλληλα με την ανακοίνωση της ίδρυσης του οργανισμού Open Handset Alliance, μιας κοινοπραξίας 48 τηλεπικοινωνιακών εταιριών, εταιριών λογισμικού καθώς και κατασκευής hardware, οι οποίες είναι αφιερωμένες στην ανάπτυξη και εξέλιξη ανοιχτών προτύπων στις συσκευές κινητής τηλεφωνίας. Παρόλο που το Android είναι ένα προϊόν ελεύθερου λογισμικού, ένα κομμάτι της ανάπτυξης του λογισμικού συνεχίζεται σε ιδιωτικό παρακλάδι. Για να έρθει αυτό το λογισμικό σε κοινή θέαση δημιουργήθηκε ένα παρακλάδι του μόνο ανάγνωσης, εν ονόματι "Cupcake". Το Cupcake συνήθως συγγέεται με τον τίτλο μιας ενημέρωσης, σε αντίθεση με όσα δηλώνει η ίδια η Google στην ιστοσελίδα ανάπτυξης του Android.

Μέτρηση Τάσης Δικτύου με arduino και εμφάνισή της σε android συσκευή μέσω bluetooth

### **1.3 Ιστορική αναδρομή και γενικές πληροφορίες πλατφόρμας arduino.**

Το **Arduino** είναι μια υπολογιστική πλατφόρμα βασισμένη σε μια απλή μητρική πλακέτα ανοικτού κώδικα, με ενσωματωμένο μικροελεγκτή και εισόδους/εξόδους, και η οποία μπορεί να προγραμματιστεί με τη γλώσσα Wiring (ουσιαστικά πρόκειται για τη γλώσσα προγραμματισμού C++ και ένα σύνολο από βιβλιοθήκες, υλοποιημένες επίσης στην C++)

Το 2005, ένα σχέδιο κίνησε προκειμένου να φτιαχτεί μία συσκευή για τον έλεγχο προγραμμάτων διαδραστικών σχεδίων από μαθητές, η οποία θα ήταν ποιο φθηνή από άλλα πρωτότυπα συστήματα διαθέσιμα εκείνη την περίοδο. Οι ιδρυτές Massimo Banzi και David Cueartielles ονόμασαν το σχέδιο από τον Arduino της Ivrea και ξεκίνησαν να παράγουν πλακέτες σε ένα μικρό εργοστάσιο στην Ιβρέα , κωμόπολη της επαρχίας Τορίνο της βορειοδυτικής Ιταλία .

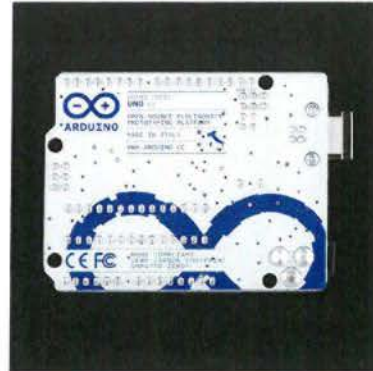
### **1.4 Ιστορική αναδρομή και γενικές πληροφορίες τεχνολογίας Bluetooth.**

Το Bluetooth είναι ένα βιομηχανικό πρότυπο για ασύρματα προσωπικά δίκτυα υπολογιστών(Wireless Personal Area Networks, WPAN). Πρόκειται για μια ασύρματη τηλεπικοινωνιακή τεχνολογία μικρών αποστάσεων, η οποία μπορεί να μεταδώσει σήματα μέσω μικροκυμάτων σε ψηφιακές συσκευές. Ιδρύθηκε το 1990 από την Ericsson η οποία έθεσε τις βάσεις για την ανάπτυξη μίας τεχνολογίας η οποία θα επέτρεπε τον σχηματισμό τοπικών δικτύων πολύ μικρής εμβέλειας. Το πρότυπο που προέκυψε υιοθετήθηκε στη συνέχεια από την IEEE ως το πρότυπο 802.15 για WPAN.

## 2: Arduino Electronic Platform



Εικόνα 1



Εικόνα 2

### 2.1 Arduino electronic platform

Οι πλατφόρμες Arduino κατασκευάζονται κυρίως από την εταιρία Smart Project. Ωστόσο, το Arduino ξεκίνησε ως έργο προς ανάπτυξη το 2005 στην Ιταλία, στο Ινστιτούτο Αλληλεπίδρασης Σχεδίασης Invea ώστε οι φοιτητές του Ινστιτούτου να αναπτύσσουν ενσωματωμένα συστήματα οικονομικά και αποδοτικά αξιοποιώντας τις δυνατότητες και τις ευκαιρίες που μπορεί να προσφέρει το ελεύθερο λογισμικό.

Γενικότερα, το Arduino θα λέγαμε ότι είναι ένα εργαλείο που μπορούμε να κατασκευάσουμε ένα υπολογιστικό σύστημα με την έννοια ότι αυτό θα ελέγχει συσκευές του φυσικού κόσμου, σε αντίθεση με τον κοινό Ηλεκτρονικό Υπολογιστή. Βασίζεται σε ευέλικτο, εύκολο στη χρήση υλικό και λογισμικό, σε μια αναπτυξιακή πλακέτα που ενσωματώνει επάνω έναν μικροελεγκτή και συνδέεται με τον Η/Υ για να προγραμματιστεί μέσα από ένα απλό περιβάλλον ανάπτυξης. Με το Arduino δημιουργούνται συσκευές οι οποίες εξυπηρετούν διάφορους σκοπούς έχοντας την δυνατότητα να δέχονται ερεθίσματα από το περιβάλλον τους (μέσω των αισθητήρων) και να αντιδρούν ανάλογα με το πως έχουν προγραμματιστεί. Τα παραπάνω δεν ακούγονται πρωτότυπα. Υπάρχουν και άλλες πλατφόρμες και υλοποιήσεις που μπορούν να κάνουν τα ίδια πράγματα. Ποια είναι η ειδοποιός διαφορά; Το Arduino βασίζεται σε τεχνολογίες ανοιχτού κώδικα. Μπορεί να κατασκευαστεί από τον καθένα, μπορεί να ενσωματωθεί σε συσκευές ακόμα και για εμπορικούς σκοπούς και το σημαντικότερο είναι ότι υπάρχει μια ολόκληρη κοινότητα

Μέτρηση Τάσης Δικτύου με arduino και εμφάνισή της σε android συσκευή μέσω bluetooth που χρησιμοποιεί το Arduino σε κατασκευές άρα υπάρχει μεγάλος όγκος ελεύθερης πληροφορίας. Γενικά, τα Projects στον εν λόγω μικροελεγκτή μπορούν να είναι αυτόνομα (σε επίπεδο hardware) ή να επικοινωνούν με κάποιο Software στον Η/Υ του προγραμματιστή (προγράμματα όπως τα Flash, Processing, MaxMSP). Το Arduino χρησιμοποιεί τώρα ένα ειδικά προγραμματιζόμενο Atmega382 αντί του chip FTDI ώστε αυτό να επιτρέπει τόσο την πιο γρήγορη ταχύτητα μεταφοράς όσο και τη γρήγορη σειριακή επικοινωνία. Ο μικροεπεξεργαστής ενός Arduino συνήθως προγραμματίζεται εκ των προτέρων ώστε να παρέχει κάποιο φορτωτή εκκίνησης (BootLoader). Ο φορτωτής εκκίνησης υπάρχει ώστε να απλοποιεί την διαδικασία της αποθήκευσης των προγραμμάτων στην Flash Memory του Arduino μέσω σειριακής USB θύρας. Επιπλέον, η γλώσσα προγραμματισμού, οι διάφορες βιβλιοθήκες και το ολοκληρωμένο περιβάλλον ανάπτυξης που υπάρχουν για τον προγραμματισμό της πλατφόρμας Arduino αποτελούν ανοιχτό λογισμικό προσφέροντας έτσι ανεκτίμητη γνώση σε όλους.

Βασικά Πλεονεκτήματα πλατφόρμας Arduino:

- ♣ Οικονομική: Η πλατφόρμα Arduino αποτελεί οικονομική λύση διότι είναι φθηνότερη. Επιπλέον, είναι αρχιτεκτονικά ανοιχτή και μπορεί ο οποιοσδήποτε να την αναπτύξει από μόνος του.
- ♣ Μεταφέρσιμη: Σε σχέση με τις υπάρχουσες πλατφόρμες στο εμπόριο η πλατφόρμα Arduino παρέχει πλήρη μεταφερισιμότητα με αποτέλεσμα να μπορεί να προγραμματιστεί στα περισσότερα λειτουργικά συστήματα.
- ♣ Επεκτάσιμη: Το υλικό και το λογισμικό της πλατφόρμας Arduino είναι ανοιχτά και ελεύθερα για όλους. Καθημερινά, χιλιάδες υποστηρικτές του ελεύθερου λογισμικού αναπτύσσουν διάφορες βιβλιοθήκες για την υποστήριξη της πλατφόρμας. Παράλληλα, τόσο η αρχιτεκτονική όσο και το υλικό της πλατφόρμας εξελίσσονται συνεχώς.



Παρακάτω ακολουθούν μερικές από τις πλατφόρμες Arduino που έχουν αναπτυχθεί και όπου η κάθε μία είτε αποτελεί εξέλιξη κάποιας άλλης, είτε έχει αναπτυχθεί για κάποιο συγκεκριμένο σκοπό, καθώς και τα βασικά χαρακτηριστικά τους:



Μέτρηση Τάσης Δικτύου με arduino και εμφάνισή της σε android συσκευή μέσω bluetooth

Όνομα	Processor	Processor Frequency	Digital I/O pins	Digital I/O with PWM	Analog Inputs pins	Release Date	Εικόνα πλακέτας
Arduino Pro	ATmega168 or ATmega328	16MHz	14	6	6		
LilyPad Arduino	ATmega168 V or ATmega328 V	8MHz	14	6	6	October 17, 2007	
Arduino Nano	ATmega168 V or ATmega328 V	16MHz	14	6	8	May 15, 2008	
Arduino (Pro) Mini	ATmega168	8MHz (3.3Vmodel) or 16MHz (5Vmodel)	14	6	6	August 23, 2008	
Arduino Fio	ATmega328 P	8MHz	14	6	8	March 18, 2010	
Arduino Uno	ATmega328 P	16MHz	14	6	6	September 24, 2010	
Arduino Mega2560	ATmega2560	16MHz	54	14	16	September 24, 2010	
Arduino Ethernet	ATmega328	16MHz	14	4	6	July 13, 2011	
Arduino Mega ADK	ATmega2560	16MHz	54	14	16	July 13, 2011	
Arduino Leonardo	Atmega32u4	16MHz	14	6	12	October 22, 2012	
Arduino Due	AT91SAM3X8E (ARM Cortex-M3)	84MHz	54	12	12	October 22, 2012	
Arduino Micro	ATmega32u4	16MHz	20	7	12	November 8, 2012	

## Μέτρηση Τάσης Δικτύου με arduino και εμφάνισή της σε android συσκευή μέσω bluetooth

Arduino Esplora	Atmega32u4	16MHz	-	-	-	December 10, 2012	
Arduino Yun	Atmega32u4, Atheros AR9331	16 MHz, 400 MHz	14	6	12	September 10, 2013	

Πίνακας 1

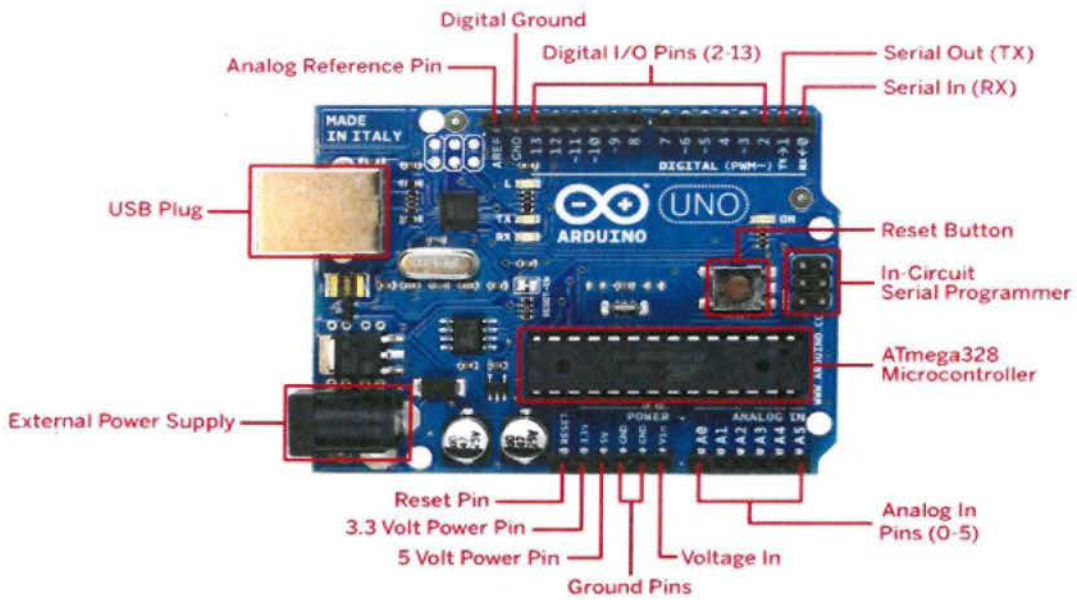
### 2.2 Arduino Hardware

Στην Συνδεσμολογία του κυκλώματος της πτυχιακής χρησιμοποιήσαμε την πλατφόρμα του Arduino Uno. Το Arduino Uno έχει 14 ψηφιακούς ακροδέκτες Εισόδου/Εξόδου οι οποίοι μπορούν να τεθούν ως είσοδοι ή ως έξοδοι με τις εντολές-συναρτήσεις `pinMode()`, `digitalWrite()`, και `digitalRead()` που θα αναλυθούν παρακάτω. Λειτουργούν στα 5 Volts και έχουν την δυνατότητα να παρέχουν ή να καταβυθίζουν ένταση της τάξεως των 40mA. Σε κάθε pin υπάρχει εσωτερικά ένας Pull-up αντιστάτης στα 20-50KΩ. Επιπλέον, έχει 6 αναλογικούς ακροδέκτες Εισόδου. Αυτοί μπορούν να διαβάσουν αναλογικές τιμές όπως η τάση μιας μπαταρίας κτλ και να τις μετατρέψουν σε έναν αριθμό από 0-1023. Η μέτρηση της τάσης γίνεται από προκαθορισμένα από 0 έως 5 volts. Εκτός αυτού, 6 εκ των 14 ψηφιακών ακροδεκτών οι P3, P5, P6, P9, P10 και P11 έχουν την δυνατότητα να προγραμματιστούν ώστε να λειτουργούν ως αναλογικές Έξοδοι.

Κάποιοι ακροδέκτες έχουν συγκεκριμένες λειτουργίες:

- Σειριακή Λειτουργία: 0 (RX) και 1 (TX). Χρησιμοποιούνται για λήψη (RX) και εκπομπή (TX) TTL σειριακών δεδομένων
- Εξωτερικές Διακοπές: 2 και 3. Αυτοί οι ακροδέκτες μπορούν να ενεργοποιούν διακοπές αν ανιχνευθεί παλμός χαμηλής τάσης. Με την συνάρτηση `attachInterrupt()`. Ο σκανδαλισμός των διακοπών μπορεί να γίνεται στο λογικό 0,1.
- PWM: 3, 5, 6, 9, 10, και 11 pins. Παρέχουν Έξοδο 8-bit PWM με την συνάρτηση `analogWrite()`.
- SPI: 10 (SS), 11 (MOSI), 12 (MISO), 13 (SCK). Αυτοί οι ακροδέκτες επιτρέπουν επικοινωνία SPI, η οποία αν και παρέχεται από το hardware δεν είναι ακόμα διαθέσιμη στην γλώσσα προγραμματισμού του Arduino
- LED: 13. Στον ακροδέκτη 13 υπάρχει ένα ενσωματωμένο LED. Όταν ο ακροδέκτης έχει τιμή HIGH, το LED ανάβει ενώ όταν το pin είναι LOW δεν ανάβει.

Μέτρηση Τάσης Δικτύου με arduino και εμφάνισή της σε android συσκευή μέσω bluetooth



Εικόνα 3

### ATmega168/328 Pin Mapping

Arduino function	ATmega Pin	ATmega Pin	Arduino function	
reset	(PCINT14/RESET) PC6	1	PC5 (ADC5/SCL/PCINT13)	analog input 5
digital pin 0 (RX)	(PCINT16/RXD) PD0	2	PC4 (ADC4/SDA/PCINT12)	analog input 4
digital pin 1 (TX)	(PCINT17/TXD) PD1	3	PC3 (ADC3/PCINT11)	analog input 3
digital pin 2	(PCINT18/INT0) PD2	4	PC2 (ADC2/PCINT10)	analog input 2
digital pin 3 (PWM)	(PCINT19/OC2B/INT1) PD3	5	PC1 (ADC1/PCINT9)	analog input 1
digital pin 4	(PCINT20/XCK/T0) PD4	6	PC0 (ADC0/PCINT8)	analog input 0
VCC	VCC	7	GND	GND
GND	GND	8	AREF	analog reference
crystal	(PCINT6/XTAL1/TOSC1) PB6	9	AVCC	VCC
crystal	(PCINT7/XTAL2/TOSC2) PB7	10	PB5 (SCK/PCINT5)	digital pin 13
digital pin 5 (PWM)	(PCINT21/OC0B/T1) PD5	11	PB4 (MISO/PCINT4)	digital pin 12
digital pin 6 (PWM)	(PCINT22/OC0A/AIN0) PD6	12	PB3 (MOSI/OC2A/PCINT3)	digital pin 11 (PWM)
digital pin 7	(PCINT23/AIN1) PD7	13	PB2 (SS/OC1B/PCINT2)	digital pin 10 (PWM)
digital pin 8	(PCINT0/CLKO/CP1) PB0	14	PB1 (OC1A/PCINT1)	digital pin 9 (PWM)

Digital Pins 11, 12 & 13 are used by the ICSP header for MISO, MOSI, SCK connections (Atmega168 pins 17, 18 & 19). Avoid low-impedance loads on these pins when using the ICSP header.

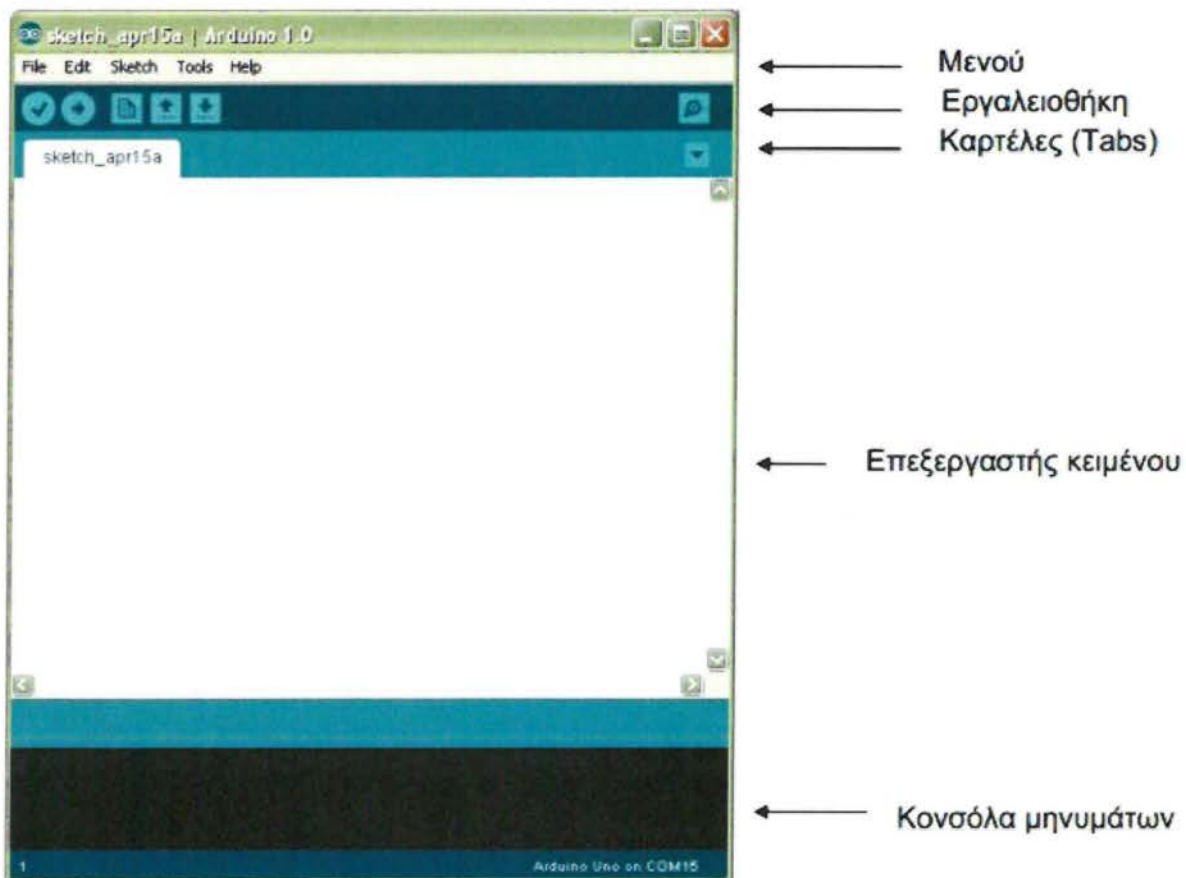
Mapping ATmega168/328 to Arduino pins

Εικόνα 4

### 2.3 Arduino Software

Η γλώσσα προγραμματισμού που χρησιμοποιεί είναι η : arduino programming language(βασισμένη σε Wiring) και το arduino development environment(βασισμένο στο processing) ουσιαστικά πρόκειται για τη γλώσσα προγραμματισμού C++ αλλά με επιπλέον βιβλιοθήκες σχεδιασμένες και αυτές σε C++.

Έρχεται με ένα φορτωτή εκκίνησης που μας επιτρέπει να ανεβάζουμε νέο κώδικα χωρίς τη χρήση εξωτερικού υλικού προγραμματιστή. Επικοινωνεί χρησιμοποιώντας το αρχικό πρωτόκολλο αναπτυξιακής κάρτας STK500. Είναι ανοιχτού κώδικα (Open- source) άρα το περιβάλλον ανάπτυξης μπορεί κάποιος να το "κατεβάσει δωρεάν".



Εικόνα 5

### 3 : Πλατφόρμα Android



Εικόνα 6

#### 3.1 Εισαγωγή

Το Android είναι ένα λειτουργικό σύστημα ανοιχτού κώδικα, βασισμένο στο Linux, για φορητές συσκευές όπως smartphones και tablets. Αναπτύχθηκε από την Google και αργότερα από την Open Handset Alliance η οποία είναι μια κοινοπραξία εταιριών λογισμικού, κατασκευής hardware και τηλεπικοινωνιών, οι οποίες είναι αφιερωμένες στην ανάπτυξη και εξέλιξη ανοιχτών προτύπων στις φορητές συσκευές. Η πρώτη παρουσίαση της πλατφόρμας Android έγινε στις 5 Νοεμβρίου 2007, παράλληλα με την ανακοίνωση της ίδρυσης του οργανισμού Open Handset Alliance. Η Google δημοσίευσε το μεγαλύτερο μέρος του κώδικα του Android υπό τους όρους της Apache License, μιας ελεύθερης άδειας λογισμικού.

Τον Ιούλιο του 2005, η Google εξαγόρασε την Android Inc, μια μικρή εταιρεία με έδρα το Palo Alto στην California των ΗΠΑ. Εκείνη την εποχή ελάχιστα ήταν γνωστά για τις λειτουργίες της Android Inc, εκτός του ότι ανέπτυσαν λογισμικό για κινητά τηλέφωνα. Αυτή ήταν η αρχή της φημολογίας περί σχεδίων της Google για να διεισδύσει στην αγορά κινητής τηλεφωνίας. Στην Google, η ομάδα με επικεφαλής τον Andy Rubin ανέπτυξε μια κινητή πλατφόρμα που στηρίζεται στον πυρήνα του Linux, την οποία προώθησαν με την παροχή ενός ευέλικτου, αναβαθμίσιμου συστήματος. Έχει αναφερθεί ότι η Google είχε ήδη συγκεντρώσει μια σειρά από εταιρους hardware και Software και επισήμανε στους παρόχους ότι ήταν ανοικτή σε διάφορους βαθμούς συνεργασίας εκ μέρους της.

Μέτρηση Τάσης Δικτύου με arduino και εμφάνισή της σε android συσκευή μέσω bluetooth

Έντυπα και ηλεκτρονικά μέσα ενημέρωσης σύντομα ανέφεραν φήμες ότι η Google ανέπτυξε μια Google-branded συσκευή. Περισσότερες φήμες ακολούθησαν, αναφέροντας ότι η Google καθόριζε τις τεχνικές προδιαγραφές και έδειχνε πρωτότυπα στους κατασκευαστές κινητών τηλεφώνων και τους φορείς δικτύων. Τελικά η Google παρουσίασε το smartphone της Nexus One που χρησιμοποιεί το open-source λειτουργικό σύστημα Android. Η συσκευή κατασκευάστηκε από την HTC , και έγινε διαθέσιμη στις 5 Ιανουαρίου 2010.



Εικόνα 7

### **3.2 Η εξέλιξη του Android**

Όπως αναφέραμε παραπάνω, το Android είναι ένα λειτουργικό σύστημα ανοιχτού κώδικα. Η εξέλιξη του λόγω της open-source φύσης του είναι ραγδαία και αυτό αντικατοπτρίζεται στο γεγονός ότι οι 9 κύριες εκδόσεις του έχουν κυκλοφορήσει σε διάστημα 4.5 ετών, από τον Απρίλη του 2009 μέχρι τον Νοέμβριο του 2014 .

Στην πληροφορική συνηθίζεται τα προϊόντα hardware και Software να κυκλοφορούν εκτός από τον αριθμό έκδοσης τους, και με μία κωδική ονομασία. Η κωδική ονομασία μπορεί να είναι πχ ονόματα πόλεων (Windows Viena, Chicago), ονόματα ζώων (OSX Leopard, Lion), στην περίπτωση όμως του Android τα κώδικα ονόματα έρχονται στη μορφή επιδορπίου!

Μέτρηση Τάσης Δικτύου με arduino και εμφάνισή της σε android συσκευή μέσω bluetooth



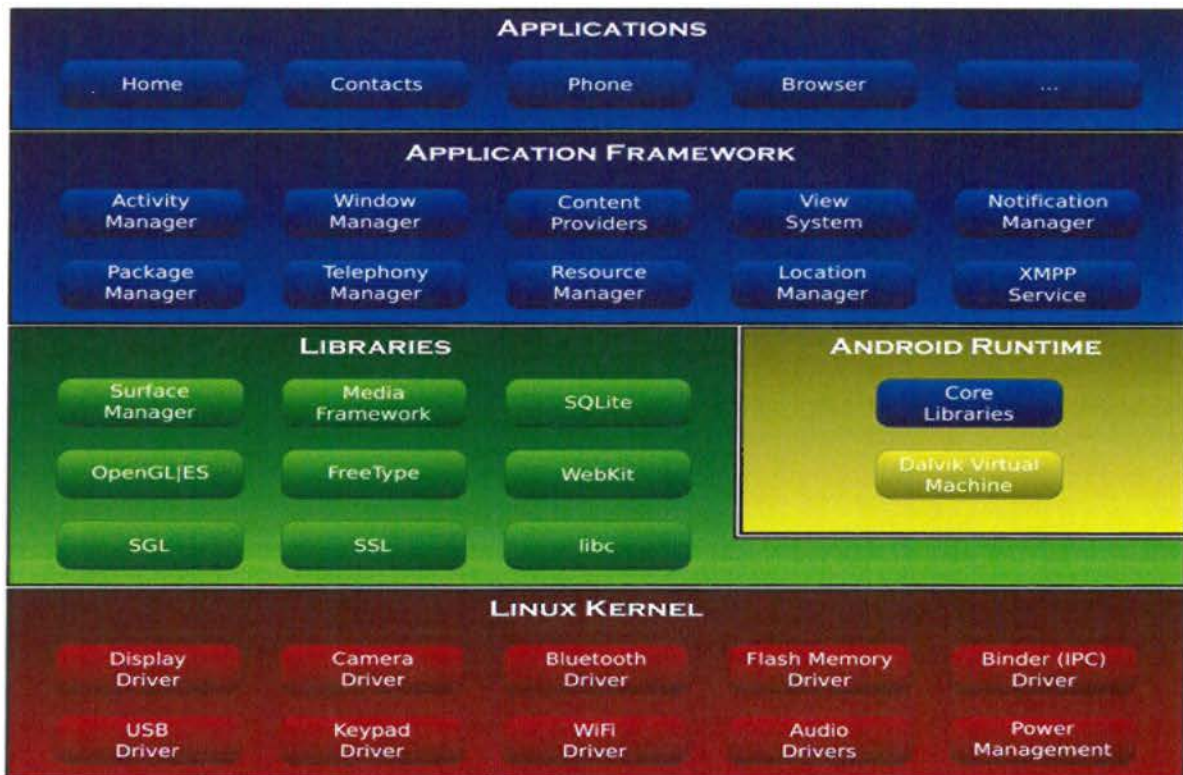
Εικόνα 8



Εικόνα 9

### 3.3 Αρχιτεκτονική Android

Το Android δεν είναι μόνο ένα λειτουργικό σύστημα. Είναι μια στοίβα λογισμικού η οποία αποτελείται από το λειτουργικό σύστημα, τις υπηρεσίες διασύνδεσης με τις εφαρμογές (middleware) και τέλος από τις κύριες (core) εφαρμογές, μεταξύ αυτών, ενός email client, μιας εφαρμογής διαχείρισης SMS, ενός ημερολογίου, ενός browser, εφαρμογή διαχείρισης επαφών, και άλλες οι οποίες έρχονται δεμένες με την υπόλοιπη στοιβάδα λογισμικού του Android. Στο επίσημο σχεδιάγραμμα που ακολουθεί (Εικόνα 10) θα δούμε οπτικά την αρχιτεκτονική αυτή



Εικόνα 10



Απο ότι βλέπουμε η αρχιτεκτονική αποτελείται από 4 βασικά επίπεδα

- Επίπεδο Linux Kernel
- Επίπεδο που εμπεριέχει τις βιβλιοθήκες (Libraries) καθώς και το android runtime(Dalvik Virtual Machine)
- Το πλαίσιο εφαρμογής (Application Framework)
- Επίπεδο διαδραστικών εφαρμογών

### 3.3.1 Επίπεδο Linux Kernel

Η βάση της στοίβας λογισμικού του Android είναι ο πυρήνας Linux. Ο τροποποιημένος πυρήνας του συστήματος βασίζεται στην έκδοση 2.6 (και στην έκδοση 3.0.1 για το Android 4.0) του Linux Kernel, η οποία υποστηρίζει όλες τις κύριες λειτουργίες του λειτουργικού συστήματος. Οι λειτουργίες αυτές αφορούν διαχείριση μνήμης, διαχείριση διεργασιών, λειτουργίες δικτύου, ασφάλεια του λειτουργικού, και ένα σύνολο οδηγών υλικού (hardware drivers). Οι οδηγοί αυτοί είναι υπεύθυνοι για την επικοινωνία του software με το hardware της συσκευής. Ενδεικτικά ο πυρήνας του Android περιέχει:

- Οδηγό προβολής οθόνης
- Οδηγό Wifi και Bluetooth
- Οδηγό κάμερας
- κλπ

Ο πυρήνας του Android μπορεί να βασίζεται στον πυρήνα του Linux, αλλά διαφέρει αρκετά από αυτόν. Ο λόγος είναι οι αλλαγές στην αρχιτεκτονική που έχει κάνει η Google για να είναι ελαφρύτερος και βελτιστοποιημένος για χρήση σε κινητές συσκευές. Αυτό σημαίνει ότι παρότι το Android είναι κατά βάση Linux, επί της ουσίας είναι αρκετά δύσκολο να τρέξουν εφαρμογές ή να χρησιμοποιηθούν βιβλιοθήκες από τη μία πλατφόρμα στην άλλη. Ο Linus Torvalds έχει αναφέρει ότι τελικά στο μέλλον το Android και το Linux θα μοιράζονται έναν κοινό πυρήνα, αλλά αυτό θα αργήσει 3-4 χρόνια ακόμα.

### 3.3.2 Επίπεδο που εμπεριέχει τις βιβλιοθήκες (Libraries) καθώς και το android runtime(Core Libraries & Dalvik Virtual Machine)

Στο δεύτερο επίπεδο της στοίβας έχουμε τις βιβλιοθήκες του Android. Αυτές ουσιαστικά αποτελούν τα APIs που είναι διαθέσιμα στους προγραμματιστές για την ανάπτυξη των εφαρμογών. Οι βιβλιοθήκες από μόνες τους δεν αποτελούν εφαρμογές αλλά ενσωματώνονται και χρησιμοποιούνται από τις εφαρμογές για τις διάφορες λειτουργίες που παρέχει η καθεμία από αυτές. Ουσιαστικά αποτελούν ένα από τα δομικά υλικά των εφαρμογών, και άρα είναι αναπόσπαστο κομμάτι τους. Οι δυνατότητες των βιβλιοθηκών του Android γίνονται εμφανείς στους προγραμματιστές στην στοίβα του πλαισίου εφαρμογής. Οι βιβλιοθήκες του android είναι οι περιγραμματικά οι εξής:

- **BiomC libc** : υπεύθυνο για διεργασίες , δημιουργία νημάτων , μαθηματικές πράξεις κτλ
- **Surface manager**: υπεύθυνο για το τι δείχνει το interface της συσκευής(πχ κατά την ανανέωση)
- **Media framework** : υπεύθυνο για την εκτέλεση αρχείων εικόνας και ήχου
- **Webkit** : υπεύθυνο για την εμφάνιση ιστοσελίδων
- **OpenGL** : υπεύθυνο για την διαχείριση υψηλής ανάλυσης γραφικά
- **SQLite** : υπεύθυνο για τη διαχείριση βάσεων δεδομένων

Σε αυτό το επίπεδο αρχιτεκτονικής βρίσκεται και το Android Runtime. Το οποίο αποτελείται από δύο κατηγορίες. Τις βασικές Java κλάσεις και το Dalvik Virtual Machine.

Σχεδόν το σύνολο των APIs του Android βασίζονται στη γλώσσα προγραμματισμού Java. Στην Java ως γνωστόν υπάρχει η λεγόμενη Java Virtual Machine στην οποία εκτελείτε ο κώδικας bytecode των εφαρμογών. Στο Android υπάρχει κάτι παρόμοιο και δεν είναι άλλο από την εικονική μηχανή Dalvik. Η **DVM(Dalvik Virtual Machine)** λοιπόν είναι η εικονική μηχανή μέσω της οποίας τρέχουν οι εφαρμογές του Android. Η κάθε εφαρμογή τρέχει μέσω τις δικής της εικονικής μηχανής στη δικιά της διεργασία και για αυτό το λόγο καμία εφαρμογή δεν έχει επαφή με την άλλη, ενώ εκτελούνται ταυτόχρονα. Η Dalvik δεν υποστηρίζει τον κώδικα bytecode, αντί αυτού οι κλάσεις της Java γίνονται compile σε αρχεία .dex ώστε να τρέξουν στην VM. Τα αρχεία dex ουσιαστικά αποτελούν συμπιεσμένα δεδομένα για εξοικονόμηση χώρου κατά την εκτέλεση. Το Android είναι από τη φύση του multitasking λειτουργικό σύστημα και για αυτό

Μέτρηση Τάσης Δικτύου με arduino και εμφάνισή της σε android συσκευή μέσω bluetooth επιτρέπει στις εφαρμογές του να τρέχουν σε πολλά νήματα ταυτόχρονα και να απασχολούν πολλές διαδικασίες εάν αυτό είναι αναγκαίο.

Για να γίνει αυτό εφικτό η μηχανή Dalvik είναι σχεδιασμένη για να έχει ελάχιστο αντίκτυπο στη χρήση της μνήμης. Χάρη στον λιτό της σχεδιασμό, το σύστημα είναι σε θέση να τρέχει πολλές εικονικές μηχανές ταυτόχρονα.

### 3.3.3 Το πλαίσιο εφαρμογής (Application Framework)

Το Android παρέχει στους developers μια ανοιχτού κώδικα πλατφόρμα ανάπτυξης και τη δυνατότητα να αναπτύξουν με αυτή ιδιαίτερα καινοτόμες και πλούσιες σε υλικό, εφαρμογές. Οι developers έχουν στην διάθεση τους τη δυνατότητα ελέγχου του υλικού της συσκευής και μέσω αυτής μπορούν να αποκτήσουν πρόσβαση σε υπηρεσίες εντοπισμού, εκτέλεση διεργασιών παρασκηνίου, και πάρα πολλές ακόμη δυνατότητες οι οποίες βασίζονται στα APIs που είναι διαθέσιμα. Τα διαθέσιμα APIs είναι τα εξής :

1. Σύστημα Προβολών(View System): Περιέχει κοινά στοιχεία γραφικών (buttons , labels κτλ )
2. Διαχειριστής Πακέτων(Package Manager) : Είναι μια βάση δεδομένων η οποία περιέχει ποιες εφαρμογές είναι εγκατεστημένες στη συσκευή
3. Διαχειριστής Παραθύρων(Window Manager) : Χειρίζεται τα παράθυρα των εφαρμογών, όπως για παράδειγμα το notification bar, το κύριο παράθυρο της συσκευής καθώς και τα υποπαράθυρα(menus and dialogs)
4. Διαχειριστής Πόρων(Resource Manager) : Διαχειρίζεται τους μη compiled πόρους της εφαρμογής όπως για παράδειγμα strings , user interface (πχ αλλαγή γλώσσας)
5. Διαχειριστής Δραστηριοτήτων(Activity Manager) : Διαχειρίζεται το κύκλο ζωής της εφαρμογής και την πλοήγηση των δραστηριοτήτων της εφαρμογής.
6. Διαχειριστής Τοποθεσίας(Location Manager) : Παρέχει πληροφορίες σχετικά με την τοποθεσία και την κίνηση της συσκευής.
7. Διαχειριστής Ειδοποιήσεων(Notification Manager) : Επιτρέπει στις εφαρμογές να χρησιμοποιούν το notification bar και να αφήνουν πληροφορίες σε αυτό.
8. Διαχειριστής Τηλεφωνίας(Telephony Manager) : Παρέχει προσβασιμότητα σχετικά με τις τηλεφωνικές υπηρεσίες του κινητού
9. Πάροχος Περιεχομένων(Content Provider) : Λειτουργεί σαν βάση δεδομένων για

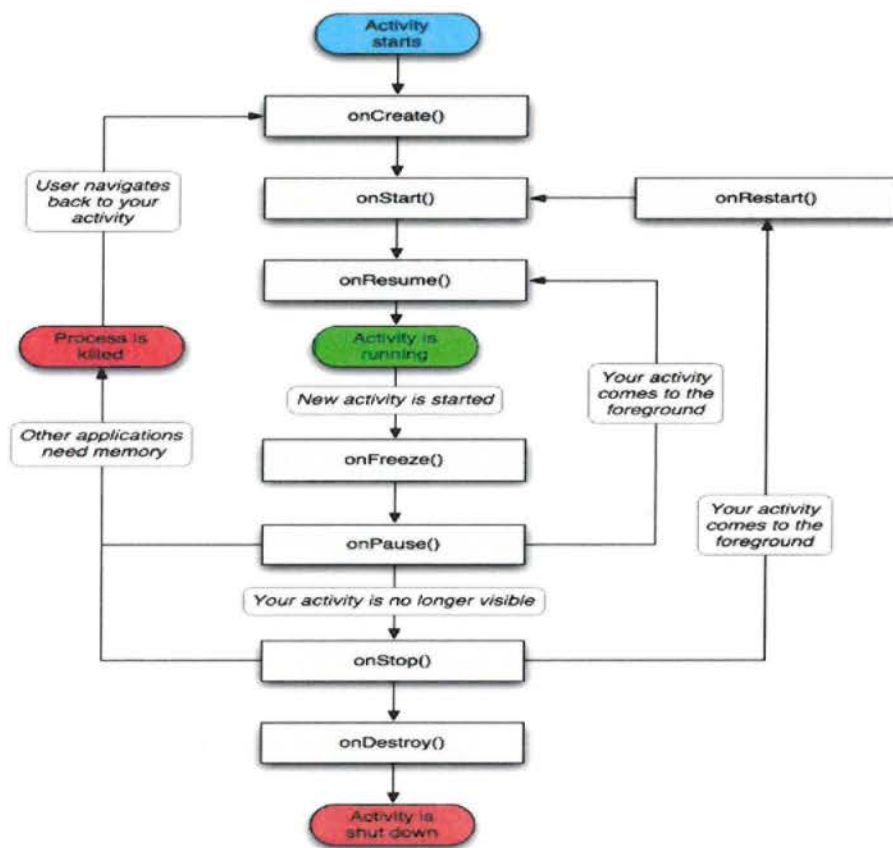
Μέτρηση Τάσης Δικτύου με arduino και εμφάνισή της σε android συσκευή μέσω bluetooth μεταφορά και αποθήκευση πληροφοριών

10. Υπηρεσία XMPP(XMPP Service) : Επιτρέπει τις εφαρμογές να συνδεθούν με τους Google servers μέσω μιας σύνδεσης TCP(Transmission Control Protocol)

### 3.3.4 Επίπεδο διαδραστικών εφαρμογών

Στο τελευταίο επίπεδο της αρχιτεκτονικής του Android λοιπόν, συναντάμε το πλαίσιο των εφαρμογών. Οι developers έχουν πρόσβαση σε όλα τα APIs μεταξύ αυτών και στα κύρια APIs που χρησιμοποιούν οι ενσωματωμένες εφαρμογές. Η δομή των εφαρμογών είναι τέτοια που ευνοείται η επαναχρησιμοποίηση δομικών συστατικών, και επίσης επιτρέπεται η χρήση των δυνατοτήτων τις μίας εφαρμογής από άλλες εφαρμογές, βέβαια κάτω από τις προδιαγραφές ασφάλειας του Android.

Το σημαντικότερο είναι ότι καμία απο αυτές τις εφαρμογές δεν είναι αναγκαστικά εγκατεστημένες , αν κάποιος θελήσει να αλλάξει μία από αυτές μπορεί πολύ απλά να την εγκαταστήσει, ωστόσο πρέπει να έχει υπόψη του την βασική δομή ζωής των εφαρμογών όπως φαίνεται και στην παρακάτω εικόνα.



### **3.4 Στο εσωτερικό μιας εφαρμογής του Android**

Κάθε εφαρμογή αποτελείται από ένα σύνολο αρχείων και φακέλων δομημένα σε μορφή project, τα οποία αφού γίνουν compiled μέσω του Android SDK μας δίνουν το αρχείο .apk. Το αρχείο αυτό αποτελεί την εφαρμογή και μπορούμε να εγκαταστήσουμε στις συσκευές μας. Ξεκινώντας, η κάθε εφαρμογή αποτελείται όπως είπαμε από πολλά αρχεία δομημένα σε φακέλους. Όλες οι εφαρμογές πρέπει να έχουν ένα μοναδικό όνομα πακέτου (Package name) το οποίο χρησιμοποιείτε από το λειτουργικό σύστημα για αναγνώριση της εφαρμογής. Μια εφαρμογή μπορεί να αποτελείται από πολλά υποπακέτα, εφόσον αυτό είναι απαραίτητο λόγω της πολυπλοκότητας της εφαρμογής, αλλά μόνο από ένα κύριο πακέτο.

### **3.5 Το αρχείο AndroidManifest.xml**

Κάθε project εφαρμογής περιέχει ένα αρχείο στο οποίο βρίσκονται καταχωρημένες οι σημαντικότερες πληροφορίες της εφαρμογής, και το αρχείο αυτό ονομάζεται AndroidManifest.xml. Πρόκειται όπως λέει και το όνομα του για ένα αρχείο xml μέσα στο οποίο ο προγραμματιστής καταχωρεί τις σημαντικότερες πληροφορίες της εφαρμογής για χρήση από το λειτουργικό σύστημα. Κάποιες από αυτές τις πληροφορίες είναι:

- Το όνομα του πακέτου της εφαρμογής
- Το κανονικό της όνομα που φαίνεται στον χρήστη
- Η έκδοση των APIs που χρησιμοποιούνται
- Ο αριθμός έκδοσης της εφαρμογής
- Οι άδειες χρήσης που ζητάει η εφαρμογή
- Όλες οι δραστηριότητες, πάροχοι περιεχομένου, υπηρεσίες, κλπ, που περιέχει και χρησιμοποιεί η εφαρμογή.

Όπως αντιλαμβανόμαστε πρόκειται για πολύ σημαντικό αρχείο και αποτελεί κύριο συστατικό κάθε εφαρμογής. Στην δικιά μας περίπτωση για παράδειγμα όπως θα αναφέρουμε και στο κεφάλαιο Bluetooth, για να επιτραπεί η χρήση του bluetooth απο την εφαρμογή χρειάζεται την παρακάτω εντολή στο manifest.xml

```
<manifest ... >
  <uses-permission android:name="android.permission.BLUETOOTH" />
  ...
</manifest>
```

### 3.6 Οι υπόλοιποι φάκελοι του project

Ένα project αποτελείται από περισσότερους από τους 3 βασικούς φακέλους, κάποιοι από τους οποίους μπορεί να θεωρηθούν και περιττοί αναλόγως την περίπτωση. Στο project λοιπόν περιλαμβάνονται και ο φάκελος με τα διαθέσιμα APIs αναλόγως την έκδοση που έχουμε επιλέξει να δουλέψουμε, ο φάκελος με τις διαθέσιμες βιβλιοθήκες που έχουμε εισάγει στο build path του project μας, και επίσης περιλαμβάνει και τις διαβαθμίσεις του φακέλου res, όπως είναι οι φάκελοι drawable-hdpi, drawable-mdpi, layout-port, menu, κλπ. Σε αυτούς περιλαμβάνονται τα ειδικά διαμορφωμένα αρχεία πόρων που έχουμε τοποθετήσει ώστε να είναι διαθέσιμα από το λειτουργικό σύστημα, αναλόγως την περίπτωση.

### 3.7 Δομικά μέρη μιας εφαρμογής

Παραπάνω αναφέραμε ότι όλα τα δομικά μέρη της εφαρμογής πρέπει να αναφέρονται αναλυτικά στο αρχείο AndroidManifest.xml, πια είναι όμως αυτά τα δομικά μέρη και πια η λειτουργία του καθενός;

- Δραστηριότητες (Activities) – Πρόκειται ίσως για το κύριο δομικό στοιχείο μιας εφαρμογής. Δραστηριότητα είναι μια οθόνη διεπαφής χρήστη (GUI) και προβολής πληροφοριών. Κάθε εφαρμογή έχει τόσες Activities όσες και οι διαφορετικές οθόνες οι οποίες εμφανίζονται στον χρήστη. Όλες οι δραστηριότητες συνεργάζονται μεταξύ τους για να δώσουν στον χρήστη μια συνολική εμπειρία χρήσης της εφαρμογής.

- Προθέσεις (Intents) – Οι δραστηριότητες επικοινωνούν και εναλλάσσουν την λειτουργία τους μέσω των Intents. Ουσιαστικά τα Intents εξασφαλίζουν την μετάβαση από την μία

Μέτρηση Τάσης Δικτύου με arduino και εμφάνισή της σε android συσκευή μέσω bluetooth δραστηριότητα σε μια άλλη και επίσης χρησιμοποιούνται για ανταλλαγή δεδομένων. Η ανταλλαγή δεδομένων, μπορεί να γίνει είτε μεταξύ των Activities μιας εφαρμογής, είτε από τη μία εφαρμογή στην άλλη. Παραδείγματος χάρη μπορούμε μέσω ενός Intent να εκκινήσουμε έναν browser ώστε να μας ανοίξει απευθείας ένα url το οποίο έχουμε παρέχει εμείς μέσω ενός Intent.

- Υπηρεσίες (Services) – Πρόκειται για λειτουργίες της εφαρμογής οι οποίες είναι σχεδιασμένες να τρέχουν στο παρασκήνιο και να επιστρέφουν αποτελέσματά ακόμη και όταν η εφαρμογή δεν είναι στο προσκήνιο. Πχ μια εφαρμογή media player μπορεί μέσω μιας υπηρεσίας να συνεχίσει να παίζει μουσική ακόμη και όταν το κύριο παράθυρο της εφαρμογής δεν βρίσκεται στο προσκήνιο

- Πάροχος Περιεχόμενων (Content Providers) - Η ανταλλαγή δεδομένων από μια εφαρμογή στην άλλη όπως είπαμε παραπάνω μπορεί να γίνει μέσω ενός Intent, ένας πάροχος περιεχομένου όμως έχει πιο σύνθετη λειτουργία. Οι content providers μιας εφαρμογής διαχειρίζονται συγκεκριμένα δεδομένα της εφαρμογής τα οποία έχει ορίσει ο προγραμματιστής κατά την κατασκευή του. Συνηθισμένα δεδομένα τα οποία μοιράζονται μέσω Content Providers, είναι οι βάσεις δεδομένων SQLite μιας εφαρμογής, και οι επαφές του χρήστη.

- Δέκτες Μετάδοσης (Broadcast Receivers) – Πρόκειται για ένα είδους υπηρεσία η οποία αντιλαμβάνεται κάποια γεγονότα του συστήματος και αναλαμβάνει να ενημερώσει το σύστημα η τις υπόλοιπες εφαρμογές. Ο σκοπός τους είναι διπλός καθότι μπορούν και να ενημερωθούν για κάποιο συμβάν από άλλες εφαρμογές, αλλά και να ειδοποιήσουν τις υπόλοιπες εφαρμογές και το σύστημα για κάποιο συμβάν που τις ενεργοποίησε. Δεν έχουν γραφικό περιβάλλον αλλά μπορούν να προβάλουν ειδοποίηση στον χρήστη μέσω της μπάρας ειδοποιήσεων. Συνήθως χρησιμοποιούνται ως διαμεσολαβητές μεταξύ των Activities και των Services μιας εφαρμογής.

Μέτρηση Τάσης Δικτύου με arduino και εμφάνισή της σε android συσκευή μέσω bluetooth



## 4 : Τεχνολογία Bluetooth



Εικόνα 12

Αφού αναφέραμε τις δύο βασικές πλατφόρμες της πτυχιακής. Ήρθε η ώρα να αναφέρουμε και το μέσω επικοινωνίας που χρησιμοποιήσαμε στην εφαρμογή μας, την τεχνολογία Bluetooth. Παλαιότερα ο κύριος τρόπος να ενώνουμε υπολογιστικές συσκευές ήταν τα καλώδια, και βασικά τα σειριακά RS232 serial cables, σήμερα όπως ήτανε και αναμενόμενο τα περισσότερα έχουνε αντικατασταθεί από τις νέες τεχνολογίες όπως στο παράδειγμά μας το Bluetooth.

### 4.1 Τι είναι το Bluetooth;

Η ασύρματη τεχνολογία Bluetooth είναι μια μικρής εμβέλειας ασύρματη επικοινωνία, τεχνολογίας RF με προδιαγραφές το χαμηλό κόστος και ισχύ. Το οποίο ενεργοποιεί peer-to-peer διαδικτυακές συνδέσεις. Κάθε συσκευή Bluetooth έχει τη μοναδική 48 bit IEEE MAC Bluetooth διεύθυνση (BD\_ADDR), η οποία την προσδιορίζει στις άλλες συσκευές. Οι διευθύνσεις είναι παρέχεται από την SIG1 (Special Interest Group) και χρειάζεται να τοποθετηθεί σε κάθε συσκευή από τον κατασκευαστή. Οι συσκευές ενός κατασκευαστή έχουν όλες την ίδια Bluetooth διεύθυνση η οποία πρέπει στην συνέχεια να αλλαχτεί είτε από τον κατασκευαστή είτε από τον χρήστη.

Το Bluetooth Serial Port Profile (SPP) επίσης αναφέρεται και σαν RFCOMM όπου το RF σημαίνει Radio Frequency και το COMM σαν Communication Port, και μας παρέχει σειριακή μεταφορά δεδομένων χρησιμοποιώντας Sockets.

Μέτρηση Τάσης Δικτύου με arduino και εμφάνισή της σε android συσκευή μέσω bluetooth

## 4.2 Bluetooth στο Android

Τα Bluetooth APIs είναι διαθέσιμα στο λειτουργικά των android με την ελευθέρωση της έκδοσης android 2.0 SDK . Οι συσκευές android που έχουνε bluetooth hardware θα πρέπει να έχουνε τις εξής λειτουργίες:

- Να σαρώνουνε και να ανακαλύπτουν άλλες Bluetooth συσκευές
- Να θεσπίσουνε μία RFCOMM σύνδεση και να μπορούνε να μεταφέρουνε δεδομένα μέσω αυτής
- Να κρατάνε σταθερή μια point-to-point και multipoint σύνδεση με άλλες Bluetooth συσκευές

Μία επίσης πολύ σημαντική λεπτομέρεια είναι ότι η πλατφόρμα Android SDK δεν υποστηρίζει στον AVD Manager (Android Virtual Device Manager) την τεχνολογία bluetooth, άρα για να κάνεις εξομείωση πρέπει αναγκαστικά να συνδέσης στο SDK πραγματική android enabled device.

Σε περίπτωση που προσπαθήσετε να το τρέξετε μέσα απο Virtual machine θα σας εμφανιστεί η παρακάτω εικόνα σφάλματος.



Εικόνα 13

### 4.3 To Android.Bluetooth Package

Τα Bluetooth APIs είναι κομμάτι του πακέτου android.Bluetooth και το android επιτρέπει μόνο κρυπτογραφημένες συνδέσεις. Οι δύο συνδεόμενες συσκευές θα πρέπει να είναι bonded ή paired ώστε να μπορέσουν να συνδεθούν. Όταν γίνει το pairing τότε μπορείς να συνδεθείς χωρίς κανένα πρόβλημα ασφαλείας.

Τα Bluetooth APIs classes που χρησιμοποιούνται είναι τα εξής όπως αναφέρονται στο πίνακα.

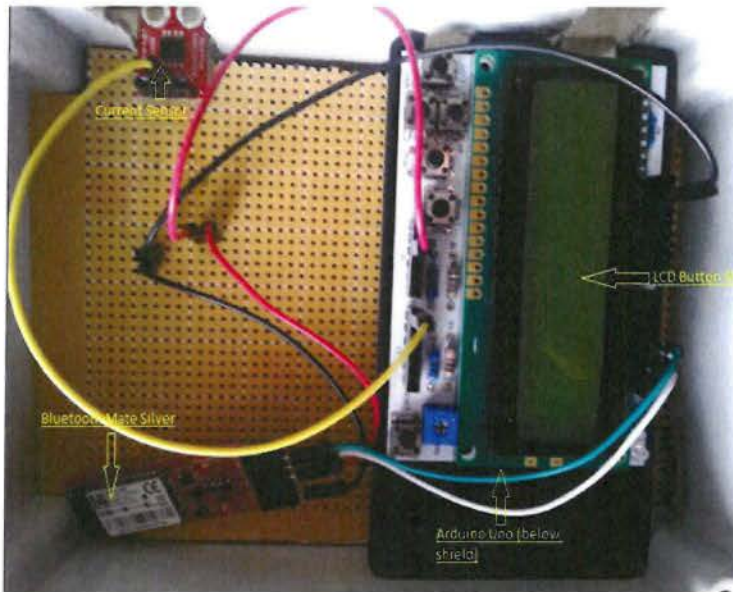
Class	Description
<b>BluetoothAdapter</b>	<ul style="list-style-type: none"><li>• Represents the Bluetooth radio hardware on the device</li></ul>
<b>BluetoothDevice</b>	<ul style="list-style-type: none"><li>• Represents a remote Bluetooth device</li></ul>
<b>BluetoothServerSocket</b>	<ul style="list-style-type: none"><li>• Used to open a socket to listen for incoming connections.</li><li>• Provides a <b>BluetoothSocket</b> object when a connection is made</li></ul>
<b>BluetoothSocket</b>	<ul style="list-style-type: none"><li>• Used by the client to establish a connection to a remote device</li><li>• After the device is connected, a <b>BluetoothSocket</b> object is used by both sides to handle the connection and retrieve the input and output streams</li></ul>

Εικόνα 14

Για να μπορέσουμε να χρησιμοποιήσουμε Bluetooth στην εφαρμογή μας πρέπει να το προσθέσουμε στο android.manifest.xml αρχείο της πλατφόρμας SDK. Αυτό γίνεται με τον εξής τρόπο :

```
<manifest ... >
  <uses-permission android:name="android.permission.BLUETOOTH" />
  ...
</manifest>
```

## 5: Η υλοποίηση του κυκλώματος και ο Arduino κώδικας



Εικόνα 15

### 5.1 Εισαγωγή

Σε αυτό το κεφάλαιο θα αναφέρουμε τα βήματα που χρειάστηκαν για την υλοποίηση του ηλεκτρονικού τμήματος της πτυχιακής. Αρχικά θα αναλύσουμε την συνδεσμολογία του κυκλώματος που εμπεριέχει κυρίως τον arduino καθώς και τα κομμάτια κώδικα του arduino ώστε να καταλάβει ο αναγνώστης πως ακριβώς δουλεύει το κύκλωμα μας.

### 5.2 Συνδεσμολογία κυκλώματος

Συνδεσμολογικά η πτυχιακή είχε τρία κομμάτια:

1. Σύνδεση του αισθητήρα ρεύματος με τον arduino
2. Σύνδεση του Bluetooth με τον arduino
3. Σύνδεση της LCD εικόνας με τον arduino
4. Ένωση όλων των παραπάνω

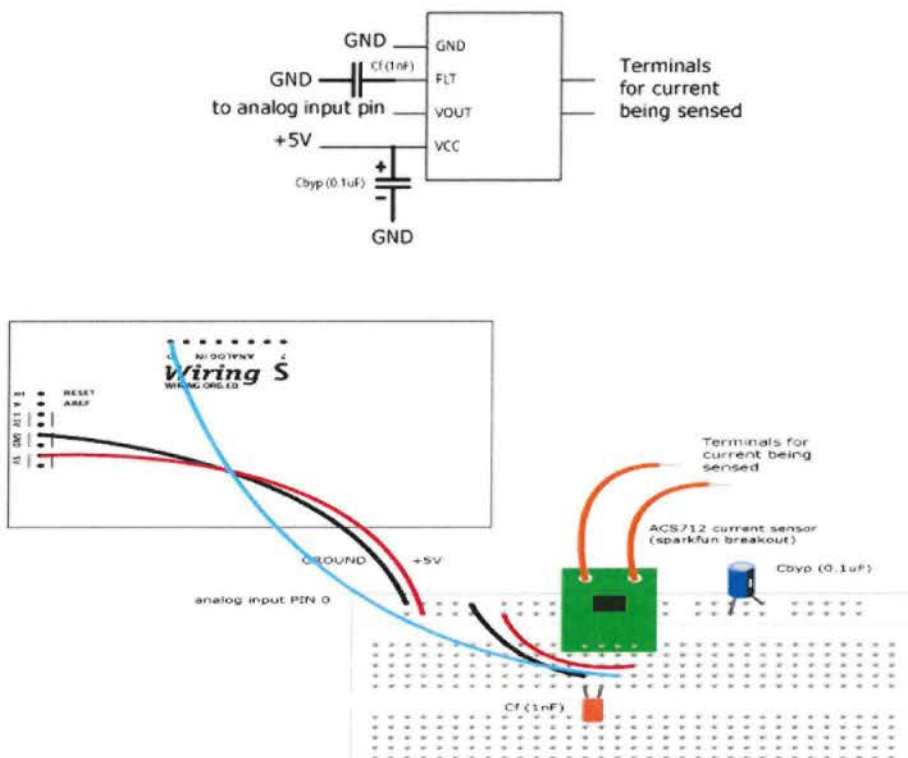
### 5.2.1 Σύνδεση αισθητήρα ρεύματος

Η συνδεσμολογία ενός αισθητήρα με τον arduino είναι σχετικά πολύ πολύπλοκο ζήτημα, η βασική ουσία είναι να πάρουμε την αναλογική έξοδο του αισθητήρα και να την ενώσουμε σε ένα από τα αναλογικά pin του arduino. Παρόλο που ακούγεται αρκετά εύκολο, τα προβλήματα που προκύπτουν είναι πολλά. Σε περίπτωση που συναντήσουμε κάποιο πρόβλημα τρέχουμε στο σχετικό datasheet του εξαρτήματος. Στην δική μας περίπτωση χρησιμοποιήσαμε τον αισθητήρα ρεύματος ACS712 breakout (<https://www.sparkfun.com/products/8882>) ο οποίος παρέχει λεπτομερή μέτρηση DC και AC σήματος, χρειάζεται μόνο μία τροφοδοσία 5VDC for VCC και παρέχει αναλογική έξοδο ώστε να το χρησιμοποιήσουμε σε κάποιο αναλογικό pin του arduino.



Εικόνα 16

Η συνδεσμολογία του δίνεται από την εταιρεία κατασκευής του (Allegro) είναι η εξής :



Εικόνα 17

Μέτρηση Τάσης Δικτύου με arduino και εμφάνισή της σε android συσκευή μέσω bluetooth

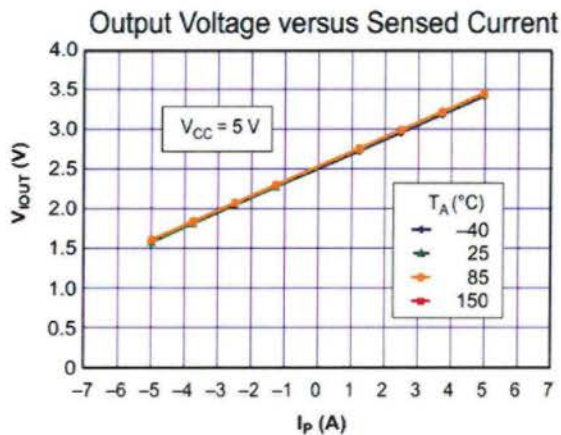
Ο προτεινόμενος κώδικας της arduino που δίνεται για σειριακή ανάγνωση δεδομένων απο αναλογικό αισθητήρα είναι ο εξής :

```
int sensorValue;

void setup()
{
  Serial.begin(9600); // sets the serial port to 9600
}

void loop()
{
  sensorValue = analogRead(A0); // read analog input pin 0
  Serial.println(sensorValue, DEC); // prints the value read
  delay(100); // wait 100ms for next reading
}
```

Όπως ξέρουμε ο arduino στα analog pins του διαβάζει τιμές από 0-5Volts και τα μετατρέπει σε 0-1023 counters. Στην δική μας περίπτωση , επειδή ο ACS712 μετράει ρεύμα αλλά αποστέλλει στον arduino και αυτός μία τάση 0-5Volts, πρέπει να μετατρέψουμε αυτά τα δεδομένα με κώδικα σε ρεύμα. Ανοίγοντας το datasheet του αισθητήρα βλέπουμε ότι για 0 ένταση ρεύματος το αισθητήριο αποστέλλει 2.5Volts στον arduino.



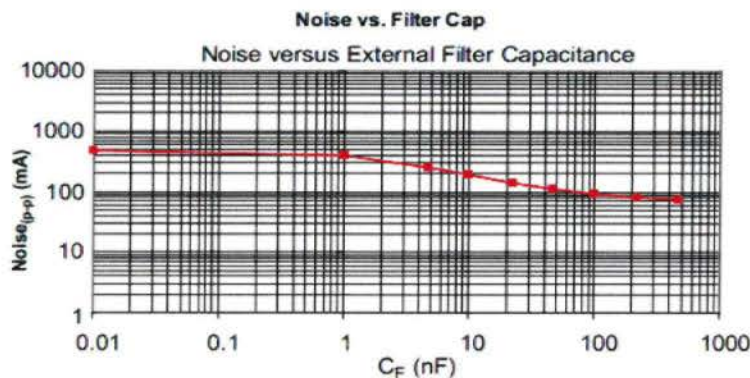
Εικόνα 18

Άρα ο arduino μετατρέπει τα 2.5Volts σε 512 counters. Για να εμφανίζονται σε εμάς 0 πρέπει με κώδικα να αφαιρέσουμε την αναλογική τιμή που θα διαβάζουμε με το νούμερο 512. Στη συνέχεια πρέπει να μετατρέψουμε τη τάση αυτή να εμφανίζετε σαν ρεύμα, για να επιτευχθεί αυτό πολλαπλασιάζουμε την παραπάνω τιμή (512 -analogRead(A0) με το μέγιστο φορτίο ρεύματος που μπορεί να διέρχεται τον αισθητήρα (πληροφορία που παρέχετε στο datasheet) και να το διαιρέσουμε με τον μέγιστο αριθμό των counters (1023) που μπορεί να επεξεργαστεί ο arduino.

Μέτρηση Τάσης Δικτύου με arduino και εμφάνισή της σε android συσκευή μέσω bluetooth

```
void setup (){\n  Serial.begin(9600);\n  void loop()\n  {\n    int sensorValue = analogRead(A0); //ανάγνωση της αναλογικής εισόδου\n    serial.println((512 - sensorValue)*27.03 /1023); //τύπωση του αποτελέσματος στην usb\n    σύνδεση (για debugging σκοπούς)\n    delay(100);\n  }\n}
```

Παρόλο που ο κώδικας ευσταθεί όπως τον γράψαμε, αντιμετωπίζουμε ένα μεγάλο πρόβλημα, ότι το συγκεκριμένο αισθητήριο δημιουργεί αρκετό θόρυβο. Ανατρέχοντας ξανά στο datasheet βλέπουμε το παρακάτω πίνακα.



Εικόνα 19

Βλέπουμε ότι όσο πιο μεγάλης χωρητικότητας πυκνωτή χρησιμοποιήσουμε τόσο λιγότερο θόρυβο θα έχουμε, σύμφωνα με το παραπάνω πίνακα, ένας πυκνωτής 470nF είναι η ιδανικότερη λύση. Για την εξακρίβωση του πυκνωτή μπορούμε να τον εφαρμόσουμε πρακτικά στη συσκευή μας και να μετρήσουμε το θόρυβο. Άρα τρέχοντας τον παρακάτω κώδικα εντοπίζουμε και τι πυκνωτής μας δίνει το μικρότερο θόρυβο.

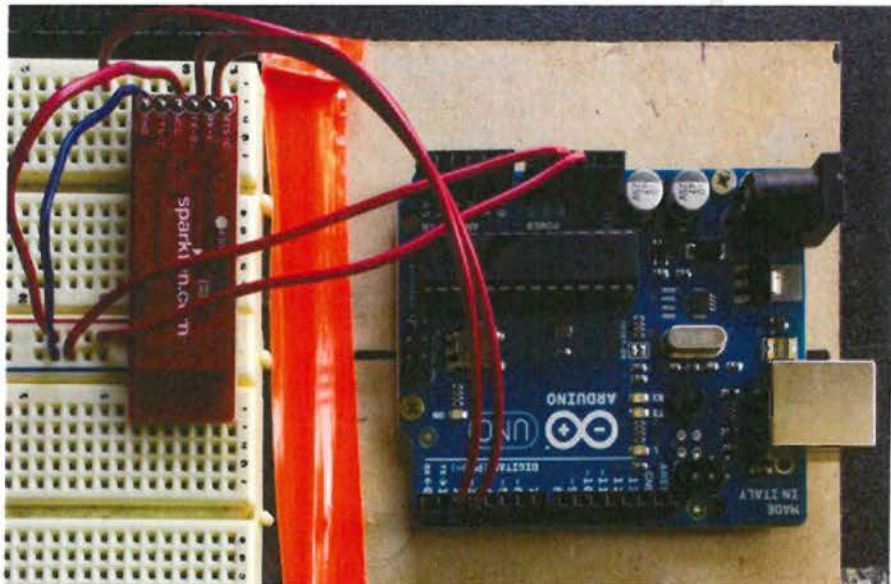
```
amps = (510 - analogRead(A0)) * 27.03 / 1023; //υπολογισμός ρεύματος\nmaxAmps = max(maxAmps, amps); //μέγιστη τιμή ρεύματος\nminAmps = min(minAmps , amps); //μικρότερη τιμή ρεύματος\nnoise= maxAmps - minAmps; //διαφορά max - min
```

Μέτρηση Τάσης Δικτύου με arduino και εμφάνισή της σε android συσκευή μέσω bluetooth

### 5.2.2 Σύνδεση του Bluetooth με τον arduino.

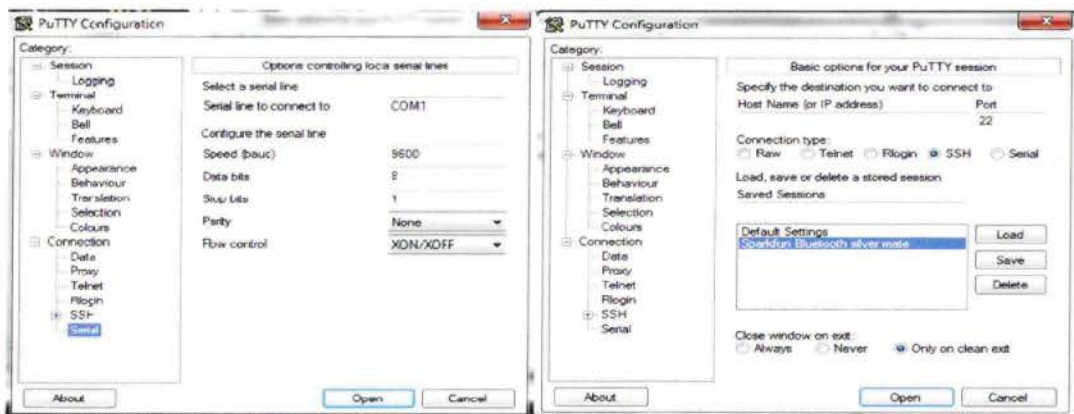
Για την πτυχιακή μας χρησιμοποιήσαμε το Silver Bluetooth Mate από την sparkfun. Η συνδεσμολογία του είναι σχετικά απλή και η sparkfun έχει ένα πολύ καλό οδηγό για την καθοδήγηση του εξαρτήματος. Παρακάτω είναι ένας πίνακας κ εικόνα με τη συνδεσμολογία του Bluetooth.

Bluetooth Mate	Arduino
CTS-I	No connection (leave floating)
VCC	5V
GND	GND
TX-O	D2
RX-I	D3
RTS-O	No connection (leave floating)



Εικόνα 20

Το επόμενο βήμα ήταν να δοκιμάσουμε αν το bluetooth device δούλευε σωστά. Για το βήμα αυτό



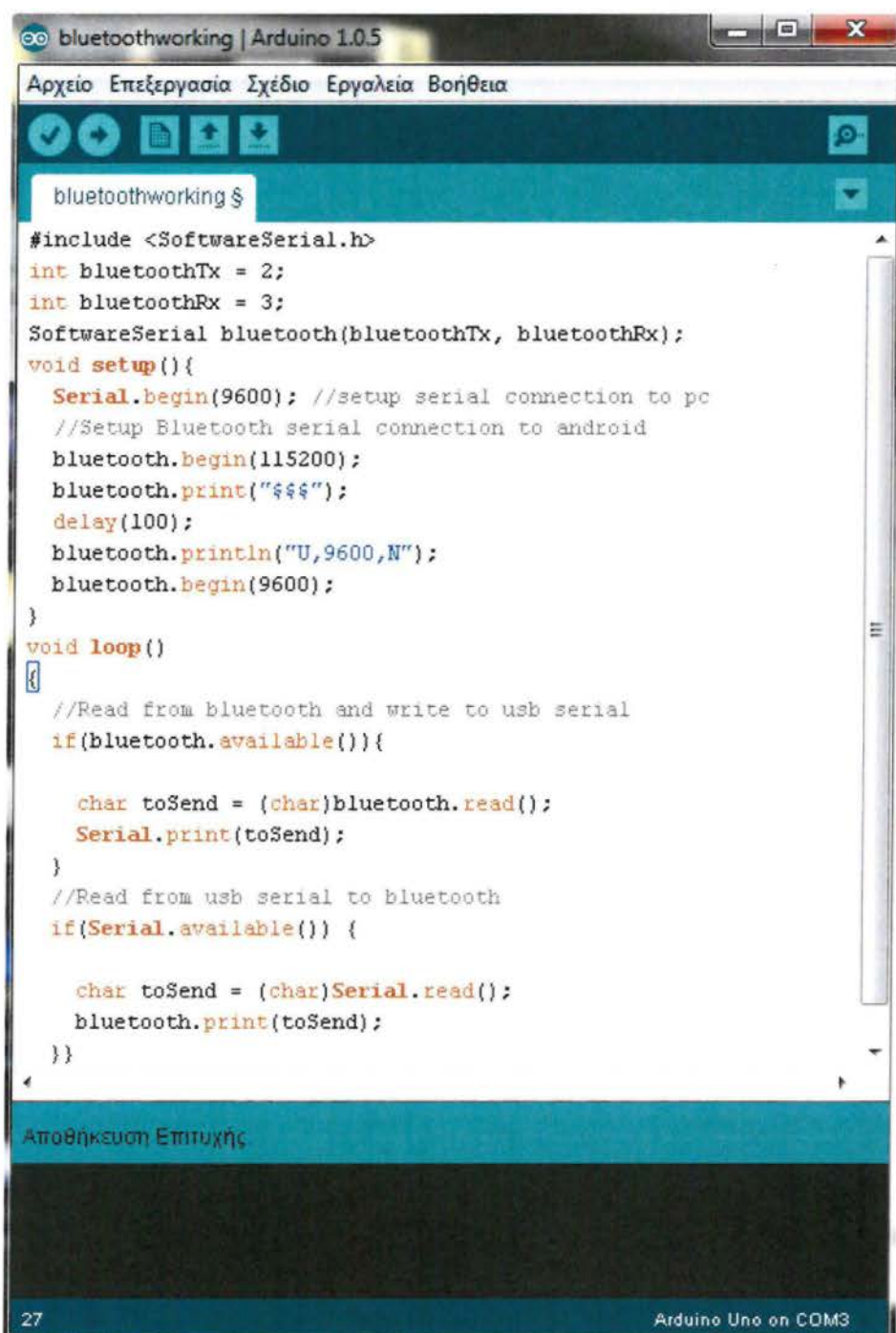
Εικόνα 21

συνδέσαμε τον arduino δια μέσω του λογισμικού putty ασύρματα με τον υπολογιστή. Τα στοιχεία εισαγωγής για τον putty είναι όπως την παρακάτω εικόνα.



Μέτρηση Τάσης Δικτύου με arduino και εμφάνισή της σε android συσκευή μέσω bluetooth

Ο κώδικας που χρειαστήκαμε ήταν ο εξής :



```
bluetoothworking | Arduino 1.0.5
Αρχείο Επεξεργασία Σχέδιο Εργαλεία Βοήθεια
bluetoothworking $
#include <SoftwareSerial.h>
int bluetoothTx = 2;
int bluetoothRx = 3;
SoftwareSerial bluetooth(bluetoothTx, bluetoothRx);
void setup(){
  Serial.begin(9600); //setup serial connection to pc
  //Setup Bluetooth serial connection to android
  bluetooth.begin(115200);
  bluetooth.print("$$$");
  delay(100);
  bluetooth.println("U,9600,N");
  bluetooth.begin(9600);
}
void loop()
{
  //Read from bluetooth and write to usb serial
  if(bluetooth.available()){
    char toSend = (char)bluetooth.read();
    Serial.print(toSend);
  }
  //Read from usb serial to bluetooth
  if(Serial.available()) {
    char toSend = (char)Serial.read();
    bluetooth.print(toSend);
  }
}
Αποθήκευση Επιτυχής
27 Arduino Uno on COM3
```

Εικόνα 22

Μέτρηση Τάσης Δικτύου με arduino και εμφάνισή της σε android συσκευή μέσω bluetooth

### 5.2.3 Συνδεσιμότητα της LCD εικόνας με τον arduino

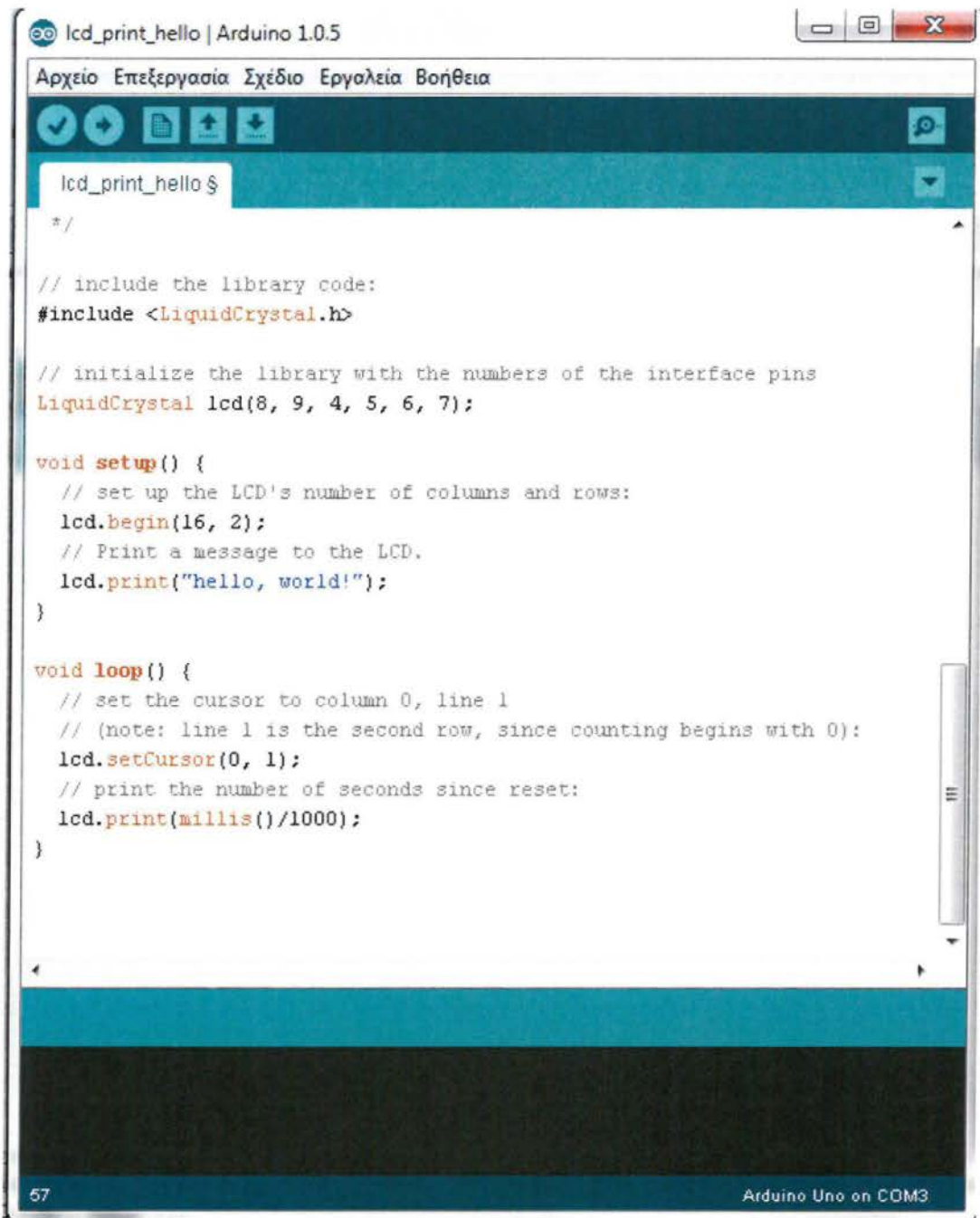
Όπως και τις παραπάνω συνδεσμολογίες, ο arduino και εδώ έχει έτοιμα libraries και οδηγούς για διευκόλυνση της κοινότητας. Στο κύκλωμά μας συνδέσαμε την οθόνη LCD : [LCD Button shield : DEV 11851](#). Είναι ένα LCD 16X2 με μαύρους χαρακτήρες και πράσινο φως για χρώμα φόντου, καθώς και εμπεριέχει και πέντε κουμπιά σε περίπτωση που ο κώδικας σου περιλαμβάνει μενού,



Εικόνα 23

Η Συνδεσμολογία του είναι πάρα πολύ απλή μιας και είναι ένα shield του arduino. Δηλαδή απλά το τοποθετείς πάνω στο arduino. Όσο αφορά τον λογισμικό τμήμα της σύνδεσης , δηλώνεις το παρακάτω κώδικα αλλάζοντας μόνο τις παραμέτρους των pins, σύμφωνα με τις οδηγίες που παραγράφονται στο datasheet του shield. Ο κώδικας που δίνεται για την εμφάνιση του Lcd είναι ο παρακάτω.

Μέτρηση Τάσης Δικτύου με arduino και εμφάνισή της σε android συσκευή μέσω bluetooth



```
lcd_print_hello | Arduino 1.0.5
Αρχείο Επεξεργασία Σχέδιο Εργαλεία Βοήθεια
lcd_print_hello $
*/
// include the library code:
#include <LiquidCrystal.h>

// initialize the library with the numbers of the interface pins
LiquidCrystal lcd(8, 9, 4, 5, 6, 7);

void setup() {
  // set up the LCD's number of columns and rows:
  lcd.begin(16, 2);
  // Print a message to the LCD.
  lcd.print("hello, world!");
}

void loop() {
  // set the cursor to column 0, line 1
  // (note: line 1 is the second row, since counting begins with 0):
  lcd.setCursor(0, 1);
  // print the number of seconds since reset:
  lcd.print(millis()/1000);
}
```

Εικόνα 24

Μέτρηση Τάσης Δικτύου με arduino και εμφάνισή της σε android συσκευή μέσω bluetooth

### 5.3 Ένωση όλων των παραπάνω.

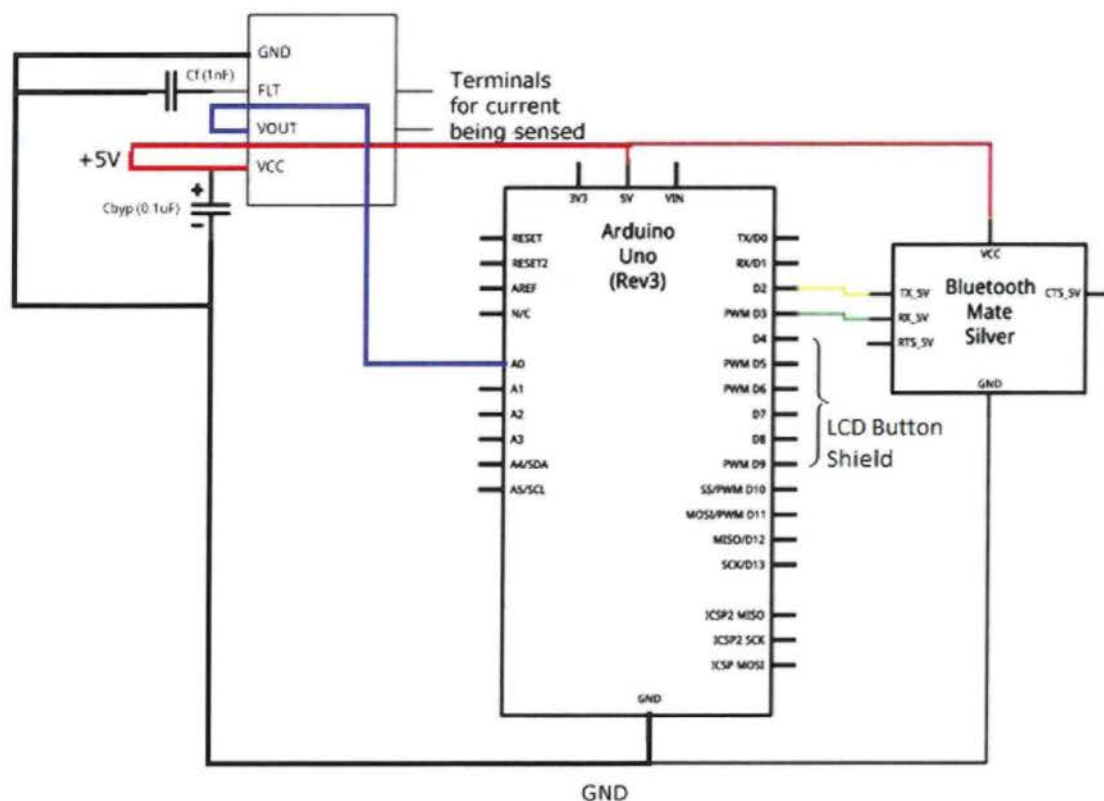
Η συνδεσμολογία του κυκλώματος φαίνεται στην παρακάτω εικόνα. Στο analog1 του arduino συνδέουμε το αισθητήρα ρεύματος και στο pin D2 τοποθετούμε το Tx του bluetooth. Και τέλος στο

D3 το  $T_R$  Το LCD όπως αναφέραμε απλά τοποθετείται πάνω στον arduino.

Τα digital pins του arduino χρησιμοποιούνται για το LCD Shield

- D4: LCD bit 4
- D5: LCD bit 5
- D6: LCD bit 6
- D7: LCD bit 7
- D8: LCD RS
- D9: LCD Enable
- D10: LCD backlight brightness adjustment

Το σχηματικό του ολοκληρωμένου κυκλώματος είναι το εξής :

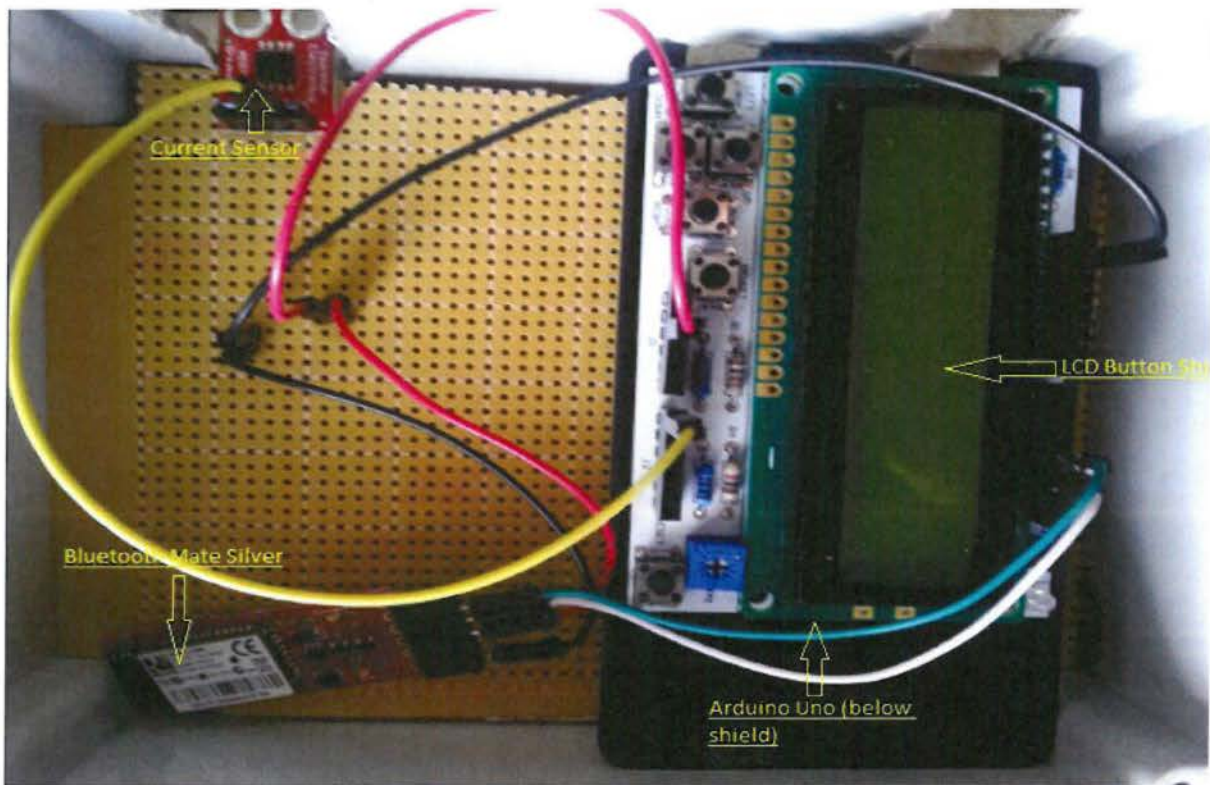


Εικόνα 25

Μέτρηση Τάσης Δικτύου με arduino και εμφάνισή της σε android συσκευή μέσω bluetooth

Arduino Pins	Ένωση με :
Vcc 5V	Vcc (bluetooth) & Vcc (Current Sensor)
Gnd	Gnd(bluetooth) & Gnd (Current Sensor)
D4	Lcd shield bit 4
D5	Lcd shield bit 5
D6	Lcd shield bit 6
D7	Lcd shield bit 7
D8	Lcd shield Rs
D9	Lcd shield Enable
A0	Vout (στον current Sensor)
D2	Tx (Bluetooth Mate Silver )
D3	Rx (Bluetooth Mate silver)

Πίνακας 2



Εικόνα 26

Αφού η υλοποίηση του Hardware ολοκληρώθηκε, η τελική μορφή του arduino είναι όπως

Μέτρηση Τάσης Δικτύου με arduino και εμφάνισή της σε android συσκευή μέσω bluetooth φαίνεται στην παραπάνω εικόνα, καθώς και η τελική μορφή του κώδικα του arduino είναι η παρακάτω :

```
#include <LiquidCrystal.h>
#include <SoftwareSerial.h>
//αρχικοποίηση μεταβλητών
float minAmps;
float maxAmps;
float amps;
float lastAmps;
float noise;
int bluetoothTx = 2;
int bluetoothRx = 3;
//επιλογή pin 2-3 για το bluetooth transmit/receive
SoftwareSerial bluetooth(bluetoothTx, bluetoothRx);

//αρχικοποίηση της βιβλιοθήκης για τα pin του LCD
LiquidCrystal lcd(8, 9, 4, 5, 6, 7);
void setup() {
// set up the LCD's number of columns and rows:
lcd.begin(16, 2);
// Print a message to the LCD.
lcd.print("Siahras K 30417");
//Setup usb serial connection to computer
Serial.begin(9600);
//Setup Bluetooth serial connection to android
bluetooth.begin(115200);
bluetooth.print("$$$");
delay(100);
bluetooth.println("U,9600,N");
bluetooth.begin(9600);
}
```

Μέτρηση Τάσης Δικτύου με arduino και εμφάνισή της σε android συσκευή μέσω bluetooth

```
void loop() {  
  //υπολογισμός του ρεύματος  
  amps = (510 - analogRead(A0)) * 27.03 / 1023;  
  if (amps>0) {  
    //ψηφιακό φίλτρο για τη μείωση του θορύβου  
    amps = (amps + lastAmps) / 2;  
    lastAmps = amps;  
    maxAmps = max(maxAmps, amps);  
    minAmps = min(minAmps , amps);  
    noise= maxAmps – minAmps;  
    //εκτύπωση στον υπολογιστή για debuggin σκοπούς  
    Serial.print(((510 - analogRead(A0))*27.03/1023));  
    Serial.print(" ");  
    Serial.print(amps);  
    Serial.print(" ");  
    Serial.print(noise);  
    Serial.println(" ");  
    if (Serial.read() != -1) {maxAmps = amps; minAmps = amps;}  
    delay(100);  
    // wait 100ms for next reading  
    // cursor 2h grammh  
    lcd.setCursor(0, 1);  
    // print to apotelesma sto lcd  
    lcd.print(((512 - analogRead(A0))*27.03/1023));  
    lcd.print(" ");  
    lcd.print(amps);  
    lcd.print(" ");  
    lcd.print(noise);  
    delay(250);  
    //stelnei sto bluetooth  
    bluetooth.print(amps);  
  }  
}
```

Μέτρηση Τάσης Δικτύου με arduino και εμφάνισή της σε android συσκευή μέσω bluetooth



## 6 : ΥΛΟΠΟΙΗΣΗ ΤΟΥ ANDROID

### 6.1 Υλοποίηση της εφαρμογής στο Android

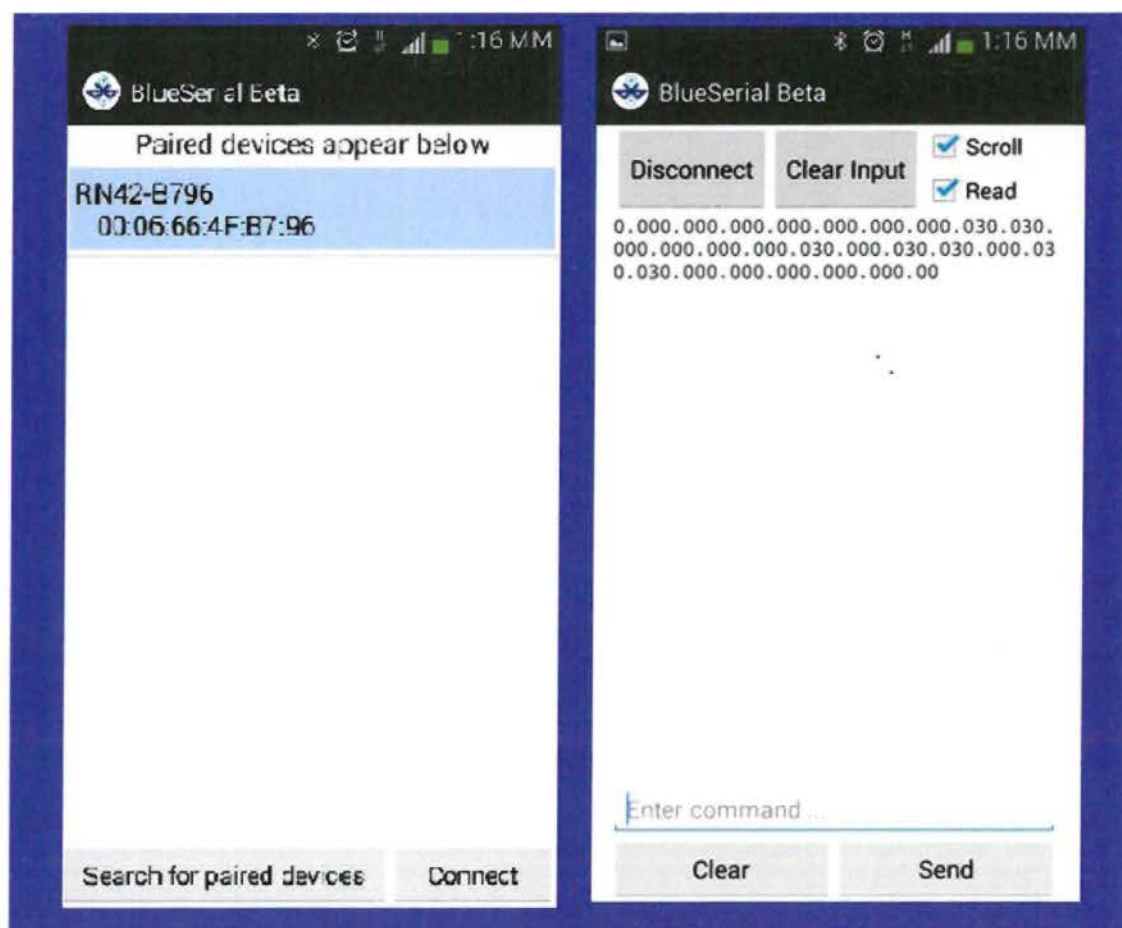
Μιας και το Arduino κομμάτι τελείωσε, το μόνο που απέμεινε είναι η υλοποίηση της εφαρμογής, παρόλο που χρησιμοποιήθηκε μία έτοιμη εφαρμογή απο το google play (Blue Serial Beta ), θα ακολουθήσει στο επόμενο κεφάλαιο η ανάλυση του κώδικα του android. Στο κεφάλαιο 3 παρουσιάσαμε πως λειτουργεί η πλατφόρμα android και τη δομή μιας εφαρμογής. Παρακάτω παρουσιάζονται η λειτουργία της εφαρμογής μας, . Κατά την ενεργοποίηση της εφαρμογής , η αρχική οθόνη παρουσιάζεται όπως την παρακάτω εικόνα.



Εικόνα 27

Η οποία αποτελείτε από τρία βασικά Views . Μία λίστα , η οποία εμφανίζει τα Bluetooth

Μέτρηση Τάσης Δικτύου με arduino και εμφάνισή της σε android συσκευή μέσω bluetooth devices τα οποία έχουν συνδεθεί στο παρελθόν με την συσκευή μας.(δεν κάνει scan απλά κοιτάει τα Settings-->bluetooth για προηγούμενες συσκευές). Τα επόμενα Views που εμφανίζονται στην εφαρμογή μας είναι δύο κουμπιά (buttons) από τα οποία το ένα είναι για να ψάξει για bluetooth devices και το άλλο για να ξεκινήσει η σύνδεση , εφόσον έχουμε επιλέξει μία συσκευή απο τη λίστα μας. Με το που πατήσουμε το κουμπί Search for paired devices θα εμφανιστούν στη λίστα τα devices που έχουν το bluetooth ενεργοποιημένο. Στη συνέχεια πατώντας το κουμπί Connect θα ξεκινήσει η εμφάνιση των δεδομένων που λαμβάνει η android συσκευή από τον arduino(δια μέσω bluetooth). Η διαδικασία προβάλλεται στην παρακάτω εικόνα.



Εικόνα 28

Για να επιτευχθεί σωστά η επικοινωνία των δύο συσκευών , πρώτα πρέπει να τις συζεύξουμε

Μέτρηση Τάσης Δικτύου με arduino και εμφάνισή της σε android συσκευή μέσω bluetooth μεταξύ τους. Αυτό γίνεται με δύο τρόπους. Πρώτος είναι ο στάνταρ τρόπο τις συσκευής. Δηλαδή να πάμε στο μενού bluetooth ->scan for devices και να κάνουμε σύζευξη. Δεύτερος τρόπος είναι μέσω κώδικα, δηλαδή να ενεργοποιούμε το bluetooth με εντολές.(στην δική μας περίπτωση και τα δύο είναι προσβάσιμα). Αν ζητηθεί pin πρόσβασης και το pairing τοποθετούμε 1234, το οποίο είναι το σύνθημα για τις ηλεκτρονικές συσκευές. Από την πλευρά του arduino δεν χρειάζεται να κάνουμε κάτι ξεχωριστό εκτός από το να το έχουμε αναμμένο.

Στο προγραμματιστικό κομμάτι ξεκινάμε πρώτα χρειάζοντας μία αναφορά για τον bluetooth adaptor του android. Αυτή παίρνεται καλώντας :

```
mBTAdapter = BluetoothAdapter.getDefaultAdapter();
```

Η τιμή επιστροφής αυτής της μεθόδου θα είναι null αν η συσκευή δεν έχει bluetooth δυνατότητες.

Στην συνέχεια έχοντας τον adapter, μπορούμε να δούμε αν το bluetooth είναι ενεργό η όχι, και να το ενεργοποιήσουμε. Ο κώδικας για αυτό είναι :

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

Τώρα που ξέρουμε ότι έχουμε bluetooth adapter και ξέρουμε ότι λειτουργεί παίρνουμε πρόσβαση στη bluetooth συσκευή. Το επόμενο βήμα μας είναι να εδραιώσουμε την επικοινωνία μας δημιουργώντας τα socket για το bluetooth connection.

```
if (mBTSocket == null || !mIsBluetoothConnected) {  
    mBTSocket = mDevice.createInsecureRfcommSocketToServiceRecord(mDeviceUUID);  
    BluetoothAdapter.getDefaultAdapter().cancelDiscovery();  
    mBTSocket.connect();  
}
```

Και τέλος καθώς έχουμε και το socket έτοιμο το χρησιμοποιούμε για τη μεταφορά των δεδομένων με τις παρακάτω εντολές.

```
inputStream = mBTSocket.getInputStream();  
mBTSocket.getOutputStream().write(mEditSend.getText().toString().getBytes());
```

Μέτρηση Τάσης Δικτύου με arduino και εμφάνισή της σε android συσκευή μέσω bluetooth

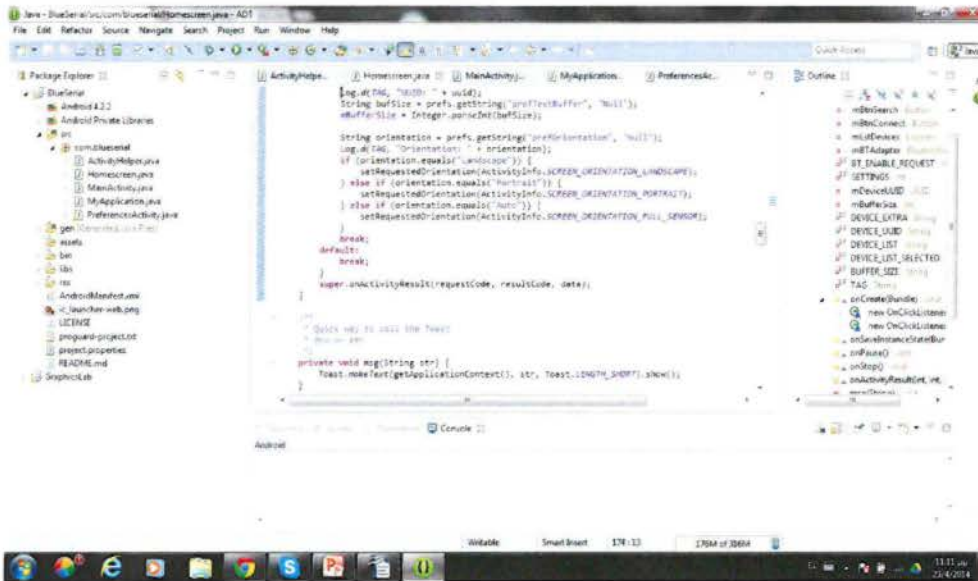
Ο κώδικας του android χωρίζεται σε 4 κυρίως δραστηριότητες , πρώτη είναι η MainActivity.class η οποία είναι υπεύθυνη για τη εμφάνιση των δεδομένων στον χρήστη και τη σωστή διεπαφή του προγράμματος(MainActivity.java), η συγκεκριμένη παίρνει τα στοιχεία της από υπόλοιπες κλάσεις. Μας παρέχει το σχεδιασμό της οθόνης κατά τη διάρκεια εμφάνισης των δεδομένων , δηλαδή όπως φαίνεται στο δεξιό κομμάτι της εικόνα 28.

Η κλάση Homescreen.java χρησιμοποιείτε για την σχεδίαση της διεπαφής της αρχικής οθόνης, δηλαδή την οθόνη που θα εμφανιστεί κατά το άνοιγμα της εφαρμογής, είναι επίσης η κλάση η οποία αρχικοποιεί την bluetooth σύνδεση

Η κλάση ActivityHelper.java χρησιμοποιείτε κυρίως από την MainActivity.java για την αποστολή δεδομένων για τον προσανατολισμό(orientation) που έχει τη δεδομένο στιγμή η android συσκευή.

Και τέλος η κλάση PreferenceActivity.java είναι υπεύθυνη για την σωστή προβολή της εφαρμογής μετά από επιλογές του χρήστη όπως το να απαντήσει μία κλήση, δηλαδή πράξεις του χρήστη που θα χρειαστούν να επιβάλλουν στην εφαρμογή μας pause η για παράδειγμα orientation (στο οποίο η συσκευή αποθηκεύει τα δεδομένα, κλείνει τη διεπαφή του χρήστη και την ξαναφτιάχνει από την αρχή με βάση το καινούργιο προσανατολισμό).

## 7: ΠΑΡΑΡΤΗΜΑ



Εικόνα 29

### 7.1 Κώδικας της εφαρμογής

#### MainActivity.java

```
package com.blueserial;

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

Μέτρηση Τάσης Δικτύου με arduino και εμφάνισή της σε android συσκευή μέσω bluetooth

```
public class MainActivity extends Activity {
    //αρχικοποίηση μεταβλητών
    private static final String TAG = "BlueTest5-MainActivity";
    private int mMaxChars = 50000;//Default
    private UUID mDeviceUUID;
    private BluetoothSocket mBTSocket;
    private ReadInput mReadThread = null;

    private boolean mIsUserInitiatedDisconnect = false;

    // Όλα τα στοιχεία της διεπαφής(κουμπιά κτλ )
    private TextView mTxtReceive;
    private EditText mEditSend;
    private Button mBtnDisconnect;
    private Button mBtnSend;
    private Button mBtnClear;
    private Button mBtnClearInput;
    private ScrollView scrollView;
    private CheckBox chkScroll;
    private CheckBox chkReceiveText;

    private boolean mIsBluetoothConnected = false;

    private BluetoothDevice mDevice;

    private ProgressDialog progressDialog;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        ActivityHelper.initialize(this);
        //φτιάχνουμε το layout χρησιμοποιώντας τον activityHelper ο οποίος έχει
        το προσανατολισμό της οθόνης κτλ(screen orientation)
        Intent intent = getIntent();
        Bundle b = intent.getExtras();
        //τα intents του activityHelper (δλδ τα δεδομένα που ήρθανε)
        //κώδικας να διαβάσει από το Homescreen τα δεδομένα για το bluetooth
        mDevice = b.getParcelable(Homescreen.DEVICE_EXTRA);
        mDeviceUUID = UUID.fromString(b.getString(Homescreen.DEVICE_UUID));
        mMaxChars = b.getInt(Homescreen.BUFFER_SIZE);

        Log.d(TAG, "Ready");
        //αρχικοποίηση των στοιχείων απο το xml file του android
        mBtnDisconnect = (Button) findViewById(R.id.btnDisconnect);
        mBtnSend = (Button) findViewById(R.id.btnSend);
        mBtnClear = (Button) findViewById(R.id.btnClear);
        mTxtReceive = (TextView) findViewById(R.id.txtReceive);
        mEditSend = (EditText) findViewById(R.id.editSend);
        scrollView = (ScrollView) findViewById(R.id.viewScroll);
        chkScroll = (CheckBox) findViewById(R.id.chkScroll);
        chkReceiveText = (CheckBox) findViewById(R.id.chkReceiveText);
        mBtnClearInput = (Button) findViewById(R.id.btnClearInput);

        mTxtReceive.setMovementMethod(new ScrollingMovementMethod());
    }
}
```

Μέτρηση Τάσης Δικτύου με arduino και εμφάνισή της σε android συσκευή μέσω bluetooth

```
mBtnDisconnect.setOnClickListener(new OnClickListener() {

    @Override
    public void onClick(View v) {
        mIsUserInitiatedDisconnect = true;
        new DisconnectBT().execute();
    }
});

mBtnSend.setOnClickListener(new OnClickListener() {

    @Override
    public void onClick(View arg0) {
        try {
mBTSocket.getOutputStream().write(mEditSend.getText().toString().getBytes());
        } catch (IOException e) {
            // TODO Auto-generated catch block
            e.printStackTrace();
        }
    }
});
//καθαρίζει στο κλικ του κουμπιού το textView για αποστολή μηνυμάτων
mBtnClear.setOnClickListener(new OnClickListener() {

    @Override
    public void onClick(View arg0) {
        mEditSend.setText("");
    }
});
//καθαρίζει τα input μηνύματα που έχουνε τυπωθεί
mBtnClearInput.setOnClickListener(new OnClickListener() {

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

}

private class ReadInput implements Runnable {

    private boolean bStop = false;
    private Thread t;
    //διαβάζει το input του bt σε νέο νήμα
    public ReadInput() {
        t = new Thread(this, "Input Thread");
        t.start();
    }

    public boolean isRunning() {
        return t.isAlive();
    }

    @Override
    public void run() {
```

Μέτρηση Τάσης Δικτύου με arduino και εμφάνισή της σε android συσκευή μέσω bluetooth

```
InputStream inputStream;

try {
    inputStream = mBTSocket.getInputStream();
    while (!bStop) {
        byte[] buffer = new byte[256];
        if (inputStream.available() > 0) {
            inputStream.read(buffer);
            int i = 0;
            /*αυτό είναι αναγκαίο επειδή κάθε νέος string
buffer χρησιμοποιεί ολόκληρο το buffer i.e 256 χαρακτήρες στο android 2.3.4
http://stackoverflow.com/a/8843462/1287554
            */
            for (i = 0; i < buffer.length && buffer[i] != 0; i++) {
            }
            final String strInput = new String(buffer, 0, i);

            /*
            *αν είναι checked τότε κάνε receive text. Για
καλύτερο σχεδιασμό θα έπρεπε να σταματάει το thread και να ελευθερώνει τα resources
            */

            if (chkReceiveText.isChecked()) {
                mTxtReceive.post(new Runnable() {
                    @Override
                    public void run() {
                        mTxtReceive.append(strInput);
                        int txtLength =
                            mTxtReceive.getEditableText().length();
                        if(txtLength > mMaxChars){
                            mTxtReceive.getEditableText().delete(0, txtLength- mMaxChars);
                        }
                    }
                });

                if (chkScroll.isChecked()) { // χρησιμοποίησε το schroll μόνο αν είναι
checked.
                    scrollView.post(new Runnable() { // Snippet from
http://stackoverflow.com/a/4612082/1287554

                    @Override

                    public void run() {

                        scrollView.fullScroll(View.FOCUS_DOWN);
                    }
                });
            }
        });
    }

    Thread.sleep(500);
}
} catch (IOException e) {
    // TODO Auto-generated catch block
    e.printStackTrace();
} catch (InterruptedException e) {
```



Μέτρηση Τάσης Δικτύου με arduino και εμφάνισή της σε android συσκευή μέσω bluetooth

```
        // TODO Auto-generated catch block
        e.printStackTrace();
    }

}

public void stop() {
    bStop = true;
}

}

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

    @Override
    protected void onPreExecute() {
    }

    @Override
    protected Void doInBackground(Void... params) {

        if (mReadThread != null) {
            mReadThread.stop();
            while (mReadThread.isRunning())
                ; // Wait until it stops
            mReadThread = null;
        }

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

        return null;
    }

    @Override
    protected void onPostExecute(Void result) {
        super.onPostExecute(result);
        mIsBluetoothConnected = false;
        if (mIsUserInitiatedDisconnect) {
            finish();
        }
    }
}

private void msg(String s) {
    Toast.makeText(getApplicationContext(), s, Toast.LENGTH_SHORT).show();
}

@Override
//αν γίνει πάυση προγράμματος
protected void onPause() {
```

Μέτρηση Τάσης Δικτύου με arduino και εμφάνισή της σε android συσκευή μέσω bluetooth

```
        if (mBTSocket != null && mIsBluetoothConnected) {
            new DisconnectBT().execute();
        }
        Log.d(TAG, "Paused");
        super.onPause();
    }

    @Override
    //στην επαναφορά του προγράμματος
    protected void onResume() {
        if (mBTSocket == null || !mIsBluetoothConnected) {
            new ConnectBT().execute();
        }
        Log.d(TAG, "Resumed");
        super.onResume();
    }

    @Override
    //κλείσιμο εφαρμογής
    protected void onStop() {
        Log.d(TAG, "Stopped");
        super.onStop();
    }

    @Override
    //αποθήκευση προσωρινών δεδομένων
    protected void onSaveInstanceState(Bundle outState) {
        // TODO Auto-generated method stub
        super.onSaveInstanceState(outState);
    }

    private class ConnectBT extends AsyncTask<Void, Void, Void> {
        private boolean mConnectSuccessful = true;

        @Override
        //σύνδεση με bluetooth
        protected void onPreExecute() {
            progressDialog = ProgressDialog.show(MainActivity.this, "Hold
on", "Connecting");// http://stackoverflow.com/a/11130220/1287554
        }

        @Override
        //δημιουργία των socket για τη σύνδεση του bt τη συσκευή
        //αρχικοποίηση του adapter και discoverability.
        protected void doInBackground(Void... devices) {
            try {
                if (mBTSocket == null || !mIsBluetoothConnected) {
                    mBTSocket =
mDevice.createInsecureRfcommSocketToServiceRecord(mDeviceUUID);

BluetoothAdapter.getDefaultAdapter().cancelDiscovery();
                    mBTSocket.connect();
                }
            } catch (IOException e) {
                // Unable to connect to device
                e.printStackTrace();
                mConnectSuccessful = false;
            }
        }
    }
}
```

Μέτρηση Τάσης Δικτύου με arduino και εμφάνισή της σε android συσκευή μέσω bluetooth

```
        }
        return null;
    }
    @Override
    protected void onPostExecute(Void result) {
        super.onPostExecute(result);

        if (!mConnectSuccessful) {
            Toast.makeText(getApplicationContext(), "Could not connect
to device. Is it a Serial device? Also check if the UUID is correct in the settings",
Toast.LENGTH_LONG).show();
            finish();
        } else {
            msg("Connected to device");
            mIsBluetoothConnected = true;
            mReadThread = new ReadInput(); // Kick off input reader
        }

        progressDialog.dismiss();
    }
}}
```

## Homescreen.java

```
package com.blueserial;

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

import com.blueserial.R;

import android.app.Activity;
import android.bluetooth.BluetoothAdapter;
import android.bluetooth.BluetoothDevice;
import android.content.Context;
import android.content.Intent;
import android.content.SharedPreferences;
import android.content.pm.ActivityInfo;
import android.graphics.Color;
import android.os.AsyncTask;
import android.os.Bundle;
import android.preference.PreferenceManager;
import android.util.Log;
import android.view.LayoutInflater;
import android.view.Menu;
import android.view.MenuItem;
import android.view.View;
import android.view.View.OnClickListener;
import android.view.ViewGroup;
import android.widget.AdapterView;
import android.widget.AdapterView.OnItemClickListener;
import android.widget.ArrayAdapter;
import android.widget.Button;
import android.widget.ListView;
```

Μέτρηση Τάσης Δικτύου με arduino και εμφάνισή της σε android συσκευή μέσω bluetooth

```
import android.widget.TextView;
import android.widget.Toast;

public class Homescreen extends Activity {

    private Button mBtnSearch;
    private Button mBtnConnect;
    private ListView mLstDevices;

    private BluetoothAdapter mBTAdapter;

    private static final int BT_ENABLE_REQUEST = 10; // This is the code we use
for BT Enable
    private static final int SETTINGS = 20;

    private UUID mDeviceUUID = UUID.fromString("00001101-0000-1000-8000-
00805F9B34FB"); // Standard SPP UUID . Οποιαδήποτε ηλεκτρονική συσκευή θέλουμε να
ενώσουμε με το android αυτό το private UUID χρησιμοποιούμε πάντα
    //
(http://developer.android.com/reference/android/bluetooth/BluetoothDevice.html#create
InsecureRfcommSocketToServiceRecord%28java.util.UUID%29)

    private int mBufferSize = 50000; //Default
    public static final String DEVICE_EXTRA = "com.blueserial.SOCKET";
    public static final String DEVICE_UUID = "com.blueserial.uuid";
    private static final String DEVICE_LIST = "com.blueserial.devicelist";
    private static final String DEVICE_LIST_SELECTED =
"com.blueserial.devicelistselected";
    public static final String BUFFER_SIZE = "com.blueserial.buffersize";
    private static final String TAG = "BlueTest5-Homescreen";

    @Override

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

        setContentView(R.layout.activity_homescreen);
        ActivityHelper.initialize(this); //Αυτό το κομμάτι κώδικα το έχουμε για
να εξασφαλίσουμε ότι screen rotation μένει σε όλα τα activities και όχι μονο σε αυτό
εδώ //
        Log.d(TAG, "Created");

        mBtnSearch = (Button) findViewById(R.id.btnSearch);
        mBtnConnect = (Button) findViewById(R.id.btnConnect);

        mLstDevices = (ListView) findViewById(R.id.LstDevices);
        /*
        *Τσεκάρει αν υπάρχει κάποιο savedInstanceState. Αν ναι , αυτό σημαίνει
ότι το onCreate ήταν πιθανών ενεργοποιημένο απο configuration change
*όπως screen rotate etc. Αν ισχύει αυτή η περίπτωση εδώ
ανακτασκευάζουμε τα views που θα αναγκαστικά θα χρειαστούμε
*/
        if (savedInstanceState != null) {
            ArrayList<BluetoothDevice> list =
savedInstanceState.getParcelableArrayList(DEVICE_LIST);
            if(list!=null){
                initList(list);
            }
        }
    }
}
```

Μέτρηση Τάσης Δικτύου με arduino και εμφάνισή της σε android συσκευή μέσω bluetooth

```
MyAdapter adapter = (MyAdapter)mLstDevices.getAdapter();
int selectedIndex =
savedInstanceState.getInt(DEVICE_LIST_SELECTED);
    if(selectedIndex != -1){
        adapter.setSelectedIndex(selectedIndex);
        mBtnConnect.setEnabled(true);
    }
} else {
    initList(new ArrayList<BluetoothDevice>());
}

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

mBtnSearch.setOnClickListener(new OnClickListener() {

    @Override
    public void onClick(View arg0) {
        mBTAdapter = BluetoothAdapter.getDefaultAdapter();

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

mBtnConnect.setOnClickListener(new OnClickListener() {

    @Override
    public void onClick(View arg0) {
        BluetoothDevice device = ((MyAdapter)
(mLstDevices.getAdapter())).getSelectedItem();
        Intent intent = new Intent(getApplicationContext(),
MainActivity.class);

        intent.putExtra(DEVICE_EXTRA, device);
        intent.putExtra(DEVICE_UUID, mDeviceUUID.toString());
        intent.putExtra(BUFFER_SIZE, mBufferSize);
        startActivity(intent);
    }
});

}

/**
 *το χρησιμοποιούμε όταν η οθόνη περιστρέφεται , αν αυτό δεν χειρίζεται σωστά
τα δεδομένα κατά την περιστροφή της οθόνης θα χαθούν
 */
@Override
protected void onSaveInstanceState(Bundle outState) {
```

Μέτρηση Τάσης Δικτύου με arduino και εμφάνισή της σε android συσκευή μέσω bluetooth

```
        super.onSaveInstanceState(outState);
        MyAdapter adapter = (MyAdapter) (mLstDevices.getAdapter());
        ArrayList<BluetoothDevice> list = (ArrayList<BluetoothDevice>)
adapter.getEntireList();

        if (list != null) {
            outState.putParcelableArrayList(DEVICE_LIST, list);
            int selectedIndex = adapter.selectedIndex;
            outState.putInt(DEVICE_LIST_SELECTED, selectedIndex);
        }
    }

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

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

    @Override
    protected void onActivityResult(int requestCode, int resultCode, Intent data)
{
        switch (requestCode) {
            case BT_ENABLE_REQUEST:
                if (resultCode == RESULT_OK) {
                    msg("Bluetooth Enabled successfully");
                    new SearchDevices().execute();
                } else {
                    msg("Bluetooth couldn't be enabled");
                }
                break;
            case SETTINGS: //If the settings have been updated
                SharedPreferences prefs =
PreferenceManager.getDefaultSharedPreferences(this);
                String uuid = prefs.getString("prefUuid", "Null");
                mDeviceUUID = UUID.fromString(uuid);
                Log.d(TAG, "UUID: " + uuid);
                String bufSize = prefs.getString("prefTextBuffer", "Null");
                mBufferSize = Integer.parseInt(bufSize);

                String orientation = prefs.getString("prefOrientation", "Null");
                Log.d(TAG, "Orientation: " + orientation);
                if (orientation.equals("Landscape")) {

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

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

                    setRequestedOrientation(ActivityInfo.SCREEN_ORIENTATION_FULL_SENSOR);
                }
            }
        }
    }
}
```

Μέτρηση Τάσης Δικτύου με arduino και εμφάνισή της σε android συσκευή μέσω bluetooth

```
        }
        break;
    default:
        break;
    }
    super.onActivityResult(requestCode, resultCode, data);
}

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

/**
 * Initialize the List adapter
 * @param objects
 */
private void initList(List<BluetoothDevice> objects) {
    final MyAdapter adapter = new MyAdapter(getApplicationContext(),
R.layout.list_item, R.id.lstContent, objects);
    mLstDevices.setAdapter(adapter);
    mLstDevices.setOnItemClickListener(new OnItemClickListener() {

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

/**
 * Ψάχνει για paired devices , δεν ξανακάνει scan απλά κοιτάει τα Setubgs-
>Bluetooth και ελέγχει αν αυτές οι συσκευές που συνεννοηθήκανε παλιότερα είναι
τριγύρω.
 */
private class SearchDevices extends AsyncTask<Void, Void,
List<BluetoothDevice>> {

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

Μέτρηση Τάσης Δικτύου με arduino και εμφάνισή της σε android συσκευή μέσω bluetooth

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

}

/**
 * Ο προσαρμογέας για να δείξει τις συσκευές σε μία λίστα
 *
 */
private class MyAdapter extends ArrayAdapter<BluetoothDevice> {
    private int selectedIndex;
    private Context context;
    private int selectedColor = Color.parseColor("#abcdef");
    private List<BluetoothDevice> myList;

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

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

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

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

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

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



Μέτρηση Τάσης Δικτύου με arduino και εμφάνισή της σε android συσκευή μέσω bluetooth

```
private class ViewHolder {
    TextView tv;
}

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

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

@Override
//κάνουμε εισαγωγή της λίστας του bt με τη μέθοδο inflate
public View getView(int position, View convertView, ViewGroup parent) {
    View vi = convertView;
    ViewHolder holder;
    if (convertView == null) {
        vi =
LayoutInflater.from(context).inflate(R.layout.list_item, null);
        holder = new ViewHolder();

        holder.tv = (TextView) vi.findViewById(R.id.lstContent);

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

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

    return vi;
}

@Override
public boolean onCreateOptionsMenu(Menu menu) {
    // Με τη μέθοδο Inflate the menu; προσθέτει αντικείμενα στην action bar
    getMenuInflater().inflate(R.menu.homescreen, menu);
    return true;
}

@Override
public boolean onOptionsItemSelected(MenuItem item) {
    switch (item.getItemId()) {
        case R.id.action_settings:
            Intent intent = new Intent(Homescreen.this,
PreferencesActivity.class);
```

Μέτρηση Τάσης Δικτύου με arduino και εμφάνισή της σε android συσκευή μέσω bluetooth

```
        startActivityForResult(intent, SETTINGS);
        break;
    }
    return super.onOptionsItemSelected(item);
}
}
```

## ActivityHelper.java

```
package com.blueserial;

import android.app.Activity;
import android.content.SharedPreferences;
import android.content.pm.ActivityInfo;
import android.preference.PreferenceManager;

public class ActivityHelper {
    public static void initialize(Activity activity) {
        SharedPreferences prefs =
        PreferenceManager.getDefaultSharedPreferences(activity);
        //Διαμορφώνει το προσανατολισμό της οθόνης
        String orientation = prefs.getString("prefOrientation", "Null");
        if ("Landscape".equals(orientation)) {

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

            activity.setRequestedOrientation(ActivityInfo.SCREEN_ORIENTATION_PORTRAIT);
        } else {

            activity.setRequestedOrientation(ActivityInfo.SCREEN_ORIENTATION_FULL_SENSOR);
        }
    }
}
```

## PreferencesActivity.java

```
package com.blueserial;

import java.util.Map;

import android.content.SharedPreferences;
import android.content.SharedPreferences.OnSharedPreferenceChangeListener;
import android.os.Bundle;
import android.preference.EditTextPreference;
import android.preference.ListPreference;
import android.preference.Preference;
import android.preference.PreferenceActivity;
import android.preference.PreferenceManager;

public class PreferencesActivity extends PreferenceActivity implements
OnSharedPreferenceChangeListener {
```

Μέτρηση Τάσης Δικτύου με arduino και εμφάνισή της σε android συσκευή μέσω bluetooth

```
@Override
public void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    ActivityHelper.initialize(this);
    addPreferencesFromResource(R.xml.preferences);
    // Χρησιμοποιούμε αυτή την συνδεσιμότητα απο συσκευές android 2.2 +
}

public void onSharedPreferenceChanged(SharedPreferences sharedPreferences,
String key) {
    Preference pref = findPreference(key);

    if (pref instanceof ListPreference) {
        ListPreference listPref = (ListPreference) pref;
        pref.setSummary(listPref.getEntry());
        ActivityHelper.initialize(this);
    }

    if (pref instanceof EditTextPreference) {
        EditTextPreference editPref = (EditTextPreference) pref;
        pref.setSummary(editPref.getText());
    }
}

@Override
protected void onPause() {

PreferenceManager.getDefaultSharedPreferences(this).unregisterOnSharedPreferenceChange
Listener(this);
    super.onPause();
}

@Override
protected void onResume() {

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

    for (Map.Entry<String, ?> entry : keys.entrySet()) {
        // Log.d("map values", entry.getKey() + ": " +
entry.getValue().toString());

        Preference pref = findPreference(entry.getKey());
        (pref != null) {
            pref.setSummary(entry.getValue().toString());
        }
    }

    super.onResume();
}
}
```

Μέτρηση Τάσης Δικτύου με arduino και εμφάνισή της σε android συσκευή μέσω bluetooth

## MyApplication.java

```
package com.blueserial;
import android.app.Application;
public class MyApplication extends Application{
    @Override
    public void onCreate() {
        // TODO Auto-generated method stub
        super.onCreate();
    }
}
```

## 7.ΒΙΒΛΙΟΘΕΣΙΑ

[1] Wei-Meng Lee 2012 “Beginning Android 4 Application Development” Wrox

[2] Reto Meier 2012 “Professional Android 4 Application Development” Wrox

[3] Massimo Banzi 2012 “Getting Started with Arduino 2<sup>nd</sup> Edition” O'REILLY

[4] Επίσημη ιστοσελίδα του android .  
<http://developer.android.com/index.html>

[5] Επίσημη ιστοσελίδα του arduino.  
<http://www.arduino.cc/>

[6] Θεωρία για την κατανόηση του android  
[https://www.coursera.org/specialization/mobilecloudcomputing/2?utm\\_medium=dashboard](https://www.coursera.org/specialization/mobilecloudcomputing/2?utm_medium=dashboard)

[7] Κοινότητα ερωτήσεων και πληροφόρησης.  
<http://stackoverflow.com/>  
<https://www.coursera.org/>

[8] Υποστήριξη θεωρίας για τη χρήση bluetooth-android  
<http://bellcode.wordpress.com/2012/01/02/android-and-arduino-bluetooth-communication/>

[9] Ιστοσελίδα αγοράς προϊόντων της πτυχιακής και εύρεσης των datasheets.  
<https://www.sparkfun.com/>

[10] Ιστοσελίδα εύρεσης της εφαρμογής που χρησιμοποιήσαμε.  
[www.googleplay.com](http://www.googleplay.com)

Μέτρηση Τάσης Δικτύου με arduino και εμφάνισή της σε android συσκευή μέσω bluetooth

## ΥΠΟΔΕΙΓΜΑ ΡΑΧΗΣ



Μέτρηση Τάσης Δικτύου με arduino και εμφάνισή της σε android συσκευή μέσω bluetooth



## ΥΠΟΔΕΙΓΜΑ ΣΥΝΟΠΤΙΚΗΣ ΠΕΡΙΓΡΑΦΗΣ

### Title

*Measuring amperage of electricity network with arduino and display it through android device via bluetooth.*

### Abstract

In this paper, we are representing the construction of a circuit that measures the ampere Voltage and display them to android running mobile devices, such as Cellphones and tablets.

### Intoduction

This current Thesis is reffering towards the marriage of 2 major free open source platforms, the arduino free electronic platform and the android free software platform. The connection between those two platforms is managed with the Bluetooth techonology.

### Overview

The aim of the thesis is to measure the amberage of a circuit or the electric network, and there post the readings inside our arduino platform. From arduino we connect the Bluetooth adaptor and we just broadcast the results at the air as a structure of serial data, for debuggin purposes we display the results serial also via a LCD screen . From the side of android , we have installed our application, which scans for bluetooth enabled devices and ask permission to pair with them. After the pairing is completed , whatever serial data the arduino platform sends, get received from the android devices and get displayed towards the user.

### Conclusion

The conclusions of this thesis isn't the specific job that our circuit does, but the possibilities that the marriage of those two free source platforms can produce. As all know the growth of the android enabled devices atm in world is enormous. Over than 6billion devices produced every few years so the new generations learn to live with them and enjoy them. Connecting the android devices with electronic circuits and getting the data from sensors, its the ideal way to making our life easier.