



Ανώτατο Τεχνολογικό Εκπαιδευτικό Ίδρυμα Πειραιά
Σχολή Τεχνολογικών Εφαρμογών
Τμήμα Ηλεκτρονικών Υπολογιστικών Συστημάτων

ΕΠΙΚΟΙΝΩΝΙΑ ANDROID ΜΕ ARDUINO ΕΦΑΡΜΟΓΕΣ

ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ

Μωρόγιαννη Βασιλική

Εισηγητής: Ιωάννης Έλληνας

Καθηγητής Α.Τ.Ε.Ι Πειραιά

ΒΙΒΛΙΟΘΗΚΗ
ΤΕΙ ΠΕΙΡΑΙΑ

ΑΙΓΑΛΕΩ, Μάιος 2014

ΕΠΙΚΟΙΝΩΝΙΑ ANDROID ΜΕ ARDUINO ΕΦΑΡΜΟΓΕΣ

ΕΠΙΚΟΙΝΩΝΙΑ ANDROID ΜΕ ARDUINO ΕΦΑΡΜΟΓΕΣ

ΕΥΧΑΡΙΣΤΙΕΣ

Η ολοκλήρωση της εργασίας αυτής δεν θα ήταν δυνατή αν όλα αυτά τα χρόνια δεν είχα την αμέριστη συμπαράσταση και καθοδήγηση ανθρώπων, οι οποίοι με αγκάλιασαν με την εμπιστοσύνη τους και με υποστήριξαν σε κάθε βήμα μου.

Θα ήθελα πρώτα να ευχαριστήσω τον επιβλέποντα καθηγητή μου κ. Έλληνα, ο οποίος με την επιμονή, υπομονή, τις γνώσεις του και την εμπειρία του συνετέλεσε τα μέγιστα στο να αισθανθώ προνομιούχα, ως μέλος του περιβάλλοντός του. Με απόλυτη ελευθερία στις κινήσεις μου και τις επιλογές μου, την σε κάθε βήμα αμέριστη εμπιστοσύνη προς το πρόσωπό μου, αλλά και διάφανη καθοδήγησή του, κατόρθωσα να επιτύχω τον πιο σημαντικό έως τώρα ακαδημαϊκό στόχο μιας μακράς και επίπονης πορείας.

Ολοκληρώνοντας, δεν θα μπορούσα να παραβλέψω φυσικά την ίδια μου την οικογένεια, η οποία στεκόταν πάντα δίπλα μου, διακριτικά, σε όλη την διάρκεια αυτής της προσπάθειάς. Ήταν οι άνθρωποι που με την αγάπη τους, την υπομονή τους και την διαρκή υποστήριξή τους έκαναν τις δύσκολες στιγμές να μοιάζουν ευκολότερες.

ΠΕΡΙΛΗΨΗ

Η παρούσα πτυχιακή ασχολείται με την εφαρμογή Amarino η οποία δίνει τη δυνατότητα σε χρήστες Android κινητών τηλεφώνων να συνδέονται μέσω Bluetooth με μια συσκευή Arduino, λαμβάνοντας στη τηλεφωνική τους συσκευή τα αποτελέσματα των εφαρμογών του Amarino. Θα εξηγήσουμε τη λειτουργία του κάθε στοιχείου που θα χρησιμοποιήσουμε (Arduino Uno, Bluetooth) καθώς και τον τρόπο εγκατάστασης των προγραμμάτων (Amarino, Arduino IDE, Eclipse). Τέλος θα παρουσιάσουμε το πρακτικό μέρος της εργασίας απεικονίζοντας κάποια παραδείγματα.

ΕΠΙΣΤΗΜΟΝΙΚΗ ΠΕΡΙΟΧΗ: Arduino

ΛΕΞΕΙΣ ΚΛΕΙΔΙΑ: Android, Amarino, Bluetooth

ΠΕΡΙΕΧΟΜΕΝΑ

1	Περιγραφή Υλικού.....	9
1.1	ANDROID	9
1.2	ARDUINO UNO	10
	Τι είναι το Arduino Uno	10
	Μνήμη Arduino.....	10
	Αναλυτικά η πλακέτα.....	11
1.3	JY-MCU BLUETOOTH	14
1.4	AMARINO	17
2	Περιγραφή Λογισμικού.....	22
2.1	Arduino IDE	22
2.2	Διαμόρφωση Bluetooth	26
2.3	Amarino.....	27
3	Παραδείγματα.....	34
3.1	Εγκατάσταση προγραμμάτων.....	34
3.2	Sensor Graph.....	38
3.3	Multi Color Lamp.....	42

ΚΑΤΑΛΟΓΟΣ ΣΧΗΜΑΤΩΝ

Εικόνα 1.1 : Arduino Uno	10
Εικόνα 1.2 :Αναλυτικά η πλακέτα	11
Εικόνα 1.3 : Bluetooth JY-MCU	14
Εικόνα 1.4 : Επικοινωνία τηλεφώνου-πλακέτας.....	17
Εικόνα 2.1 :Device Manager	23
Εικόνα 2.2 : Arduino IDE	24
Εικόνα 2.3 :Αρχικοποίηση περιβάλλοντος.....	24
Εικόνα 2.4 :Αρχικοποίηση περιβάλλοντος.....	25
Εικόνα 2.5 :Βασικό παράδειγμα	25
Εικόνα 2.6 : Εγκατάσταση Amarino ,Amarino Plug-in Bundle	27
Εικόνα 2.7 : Εγκατάσταση ως Package Installer	28
Εικόνα 2.8 :Αρχική οθόνη Amarino	28
Εικόνα 2.9 :Βασικές επιλογές Amarino.....	29
Εικόνα 2.10 :Έτοιμες εφαρμογές Amarino	30
Εικόνα 2.11 :Παράδειγμα Compass Sensor (text).....	30
Εικόνα 2.12 :Παράδειγμα Compass Sensor(graph)	31
Εικόνα 2.13 :Παράδειγμα Compass Sensor(bars).....	31
Εικόνα 2.14 :Επεξήγηση συντεταγμένων.....	32
Εικόνα 2.15 :Παράδειγμα Monitoring	33
Εικόνα 3.1 : Δημιουργία Android project.....	35
Εικόνα 3.2 : Εισαγωγή βιβλιοθήκης AmarinoLibrary.....	36
Εικόνα 3.3 : MAC Address όπως φαίνεται στο Amarino	36
Εικόνα 3.4 : Εγγραφή της MAC Address στον κώδικα.....	37
Εικόνα 3.5 : Επιλογές προγραμματιστή στην Android συσκευή.....	37
Εικόνα 3.6 : Εγκατάσταση προγραμμάτων ως Android Application	38
Εικόνα 3.7 : Συνδεσμολογία Sensor Graph	40
Εικόνα 3.8 : Γραφικά η μέγιστη τιμή του ποτενσιόμετρου.....	41
Εικόνα 3.9 : Γραφικά η ελάχιστη τιμή του ποτενσιόμετρου.....	41
Εικόνα 3.10 : Συνδεσμολογία Multi Color Lamp.....	43
Εικόνα 3.11 : Αρχική οθόνη Multi Color Lamp.....	44
Εικόνα 3.12 : Μέγιστη τιμή rgb led.....	45
Εικόνα 3.13 : Τυχαίο παράδειγμα(χρώμα) rgb led	45

ΚΑΤΑΛΟΓΟΣ ΠΙΝΑΚΩΝ

Πίνακας 1.1 : Πίνακας σφαλμάτων των AT Commands.....16

ΣΥΝΤΟΜΟΓΡΑΦΙΕΣ

ADC Analog to Digital Converter

PWM Pulse Width Modulation

IDE Integrated Development Environment

GUI Graphical User Interface

1 Περιγραφή Υλικού

ΕΙΣΑΓΩΓΗ

Σε αυτή την πτυχιακή εργασία θα παρουσιάσουμε το Arduino Uno, θα αναλύσουμε τις λειτουργίες του Arduino και θα εξηγήσουμε τον τρόπο σύνδεσης τους μέσω ενός Bluetooth το οποίο θα πρέπει να προσαρμόσουμε. Με αυτό το τρόπο θα μπορούμε να επικοινωνούμε με έναν μικροελεγκτή που θα μας εμφανίζει τα αποτελέσματα διάφορων εργασιών στη τηλεφωνική μας συσκευή.

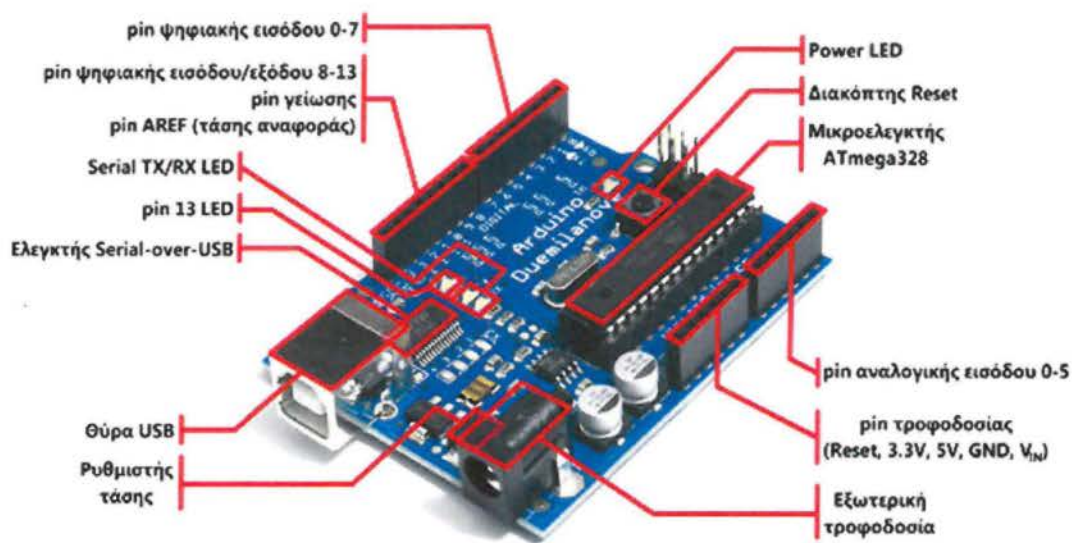
1.1 ANDROID

Το Android είναι ένα λειτουργικό σύστημα όπως τα Windows των PC, ή το OS-X των Macintosh, ένα πλαίσιο βασικού λογισμικού κι ένα γραφικό περιβάλλον μέσα στο οποίο οι κάτοχοι συμβατών συσκευών μπορούν να εγκαταστήσουν πολλές διαφορετικές εφαρμογές. Το Android ανήκει στη Google, στηρίζεται ωστόσο σε ειδική έκδοση του Linux, του γνωστού λειτουργικού συστήματος "ανοιχτού κώδικα". Κατά συνέπεια, είναι κι αυτό ελεύθερο προς αξιοποίηση με κάποιους περιορισμούς όπως οι κατασκευαστές να φέρουν και την πιστοποίηση της Google. Το Android μπορεί να στηρίξει τις λειτουργίες οποιουδήποτε είδους συσκευής (από το μικρότερο MP3 player έως τη μεγαλύτερη τηλεόραση υψηλής ευκρίνειας), έχει ωστόσο χρησιμοποιηθεί περισσότερο σε κινητά τηλέφωνα και υπολογιστές tablet. Αυτές είναι οι συσκευές που για την ώρα είναι καταλληλότερες να αξιοποιήσουν τις περίπου 200.00 εφαρμογές που η Google διαθέτει μέσα από το Δικτυακό κατάστημα του λειτουργικού της, το Android Market. Αντίπαλα του Android προϊόντα είναι το iOS της Apple, το WebOS της HP και το Windows Phone 7 της Microsoft.

1.2 ARDUINO UNO

Τι είναι το Arduino Uno

Το Arduino Uno είναι η παρακάτω έτοιμη πλακέτα.



Εικόνα 1.1 : Arduino Uno

Πρόκειται για ένα ηλεκτρονικό κύκλωμα που βασίζεται στον μικροελεγκτή ATmega της Atmel και του οποίου όλα τα σχέδια, καθώς και το software που χρειάζεται για την λειτουργία του, διανέμονται ελεύθερα και δωρεάν.

Μνήμη Arduino

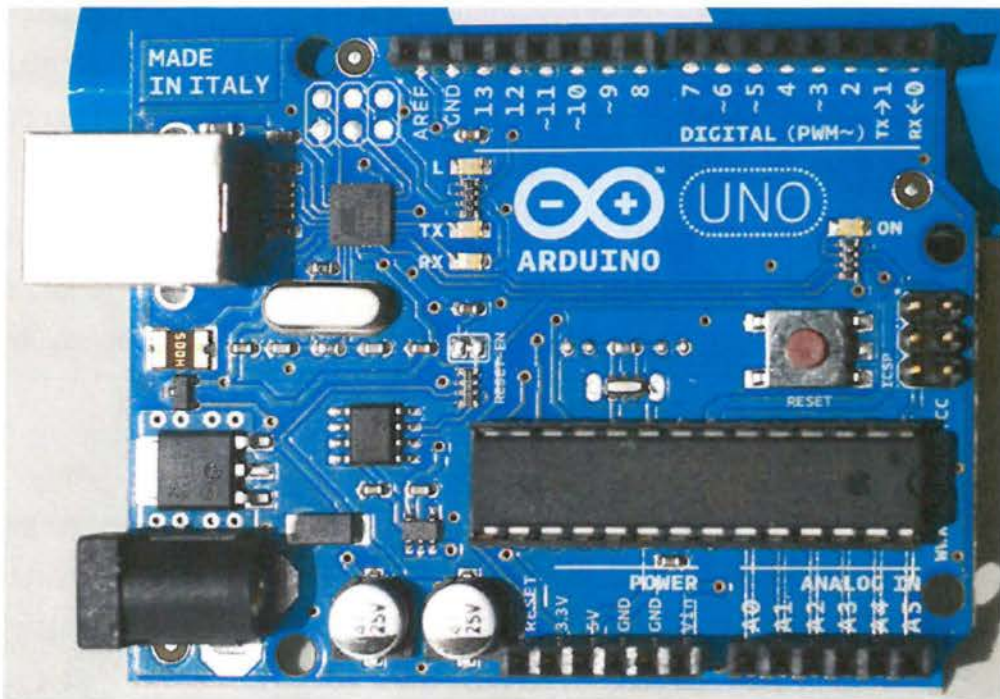
Το Arduino βασίζεται στον ATmega328, έναν 8-bit RISC μικροελεγκτή, τον οποίο χρονίζει στα 16MHz. Ο ATmega328 διαθέτει ενσωματωμένη μνήμη τριών τύπων:

- 2Kb μνήμης SRAM που είναι η ωφέλιμη μνήμη που μπορούν να χρησιμοποιήσουν τα προγράμματά μας για να αποθηκεύουν μεταβλητές, πίνακες κ.λπ. σε πραγματικό χρόνο (κατά το runtime). Όπως και σε έναν υπολογιστή, αυτή η μνήμη χάνει τα δεδομένα της όταν η παροχή ρεύματος στο Arduino σταματήσει ή αν γίνει reset.

ΕΠΙΚΟΙΝΩΝΙΑ ANDROID ΜΕ ARDUINO ΕΦΑΡΜΟΓΕΣ

- 1Kb μνήμης EEPROM η οποία μπορεί να χρησιμοποιηθεί για εγγραφή/ανάγνωση δεδομένων ανά byte από τα προγράμματά μας κατά το runtime. Σε αντίθεση με την SRAM, η EEPROM δεν χάνει τα περιεχόμενά της με απώλεια τροφοδοσίας ή reset οπότε είναι το ανάλογο του σκληρού δίσκου.
- 32Kb μνήμης Flash, από τα οποία τα 2Kb χρησιμοποιούνται από το υλικολογισμικό (firmware) του Arduino που έχει εγκαταστήσει ήδη ο κατασκευαστής του. Το firmware αυτό είναι αναγκαίο για την εγκατάσταση των προγραμμάτων μας στον μικροελεγκτή μέσω της θύρας USB. Τα υπόλοιπα 30Kb της μνήμης Flash χρησιμοποιούνται για την αποθήκευση αυτών ακριβώς των προγραμμάτων, αφού πρώτα μεταγλωττιστούν στον υπολογιστή μας. Η μνήμη Flash, όπως και η EEPROM δεν χάνει τα περιεχόμενά της με απώλεια τροφοδοσίας ή reset.

Αναλυτικά η πλακέτα



Εικόνα 1.2 :Αναλυτικά η πλακέτα

ΕΠΙΚΟΙΝΩΝΙΑ ANDROID ΜΕ ARDUINO ΕΦΑΡΜΟΓΕΣ

Στην πάνω πλευρά του Arduino βρίσκονται 14 pin, αριθμημένα από 0 ως 13, που μπορούν να λειτουργήσουν ως ψηφιακές είσοδοι και έξοδοι. Λειτουργούν στα 5V και καθένα μπορεί να παρέχει ή να δεχτεί το πολύ 40mA.

Ως ψηφιακή έξοδος, ένα από αυτά τα pin μπορεί να τεθεί από το πρόγραμμά σε κατάσταση HIGH ή LOW, οπότε το Arduino θα ξέρει αν πρέπει να διοχετεύσει ή όχι ρεύμα στο συγκεκριμένο pin.

Μερικά από αυτά τα 14 pin, εκτός από ψηφιακές είσοδοι/έξοδοι έχουν και δεύτερη λειτουργία. Συγκεκριμένα:

- Τα pin 0 και 1 λειτουργούν ως RX και TX της σειριακής όταν το πρόγραμμά μας ενεργοποιεί την σειριακή θύρα. Έτσι, όταν λόγω χάρη το πρόγραμμά στέλνει δεδομένα στην σειριακή, αυτά προωθούνται και στην θύρα USB μέσω του ελεγκτή Serial-Over-USB αλλά και στο pin 0 για να τα διαβάσει ενδεχομένως μια άλλη συσκευή .
- Τα pin 2 και 3 λειτουργούν και ως εξωτερικά interrupt (interrupt 0 και 1 αντίστοιχα). Μπορούμε να τα ρυθμίσουμε μέσα από το πρόγραμμά μας ώστε να λειτουργούν αποκλειστικά ως ψηφιακές είσοδοι στις οποίες όταν συμβαίνουν συγκεκριμένες αλλαγές, η κανονική ροή του προγράμματος σταματάει άμεσα και εκτελείται μια συγκεκριμένη συνάρτηση.
- Τα pin 3, 5, 6, 9, 10 και 11 μπορούν να λειτουργήσουν και ως ψευδοαναλογικές έξοδοι με το σύστημα PWM (Pulse Width Modulation), δηλαδή το ίδιο σύστημα που διαθέτουν οι μητρικές των υπολογιστών για να ελέγχουν τις ταχύτητες των ανεμιστήρων

Στην κάτω πλευρά του Arduino, με τη σήμανση ANALOG IN, υπάρχει μια ακόμη σειρά από 6 pin, αριθμημένα από το 0 ως το 5. Το καθένα από αυτά λειτουργεί ως αναλογική είσοδος κάνοντας χρήση του ADC (Analog to Digital Converter) που είναι ενσωματωμένο στον μικροελεγκτή.

ΕΠΙΚΟΙΝΩΝΙΑ ANDROID ΜΕ ARDUINO ΕΦΑΡΜΟΓΕΣ

Δίπλα από τα pin αναλογικής εισόδου, υπάρχει μια ακόμα συστοιχία από 6 pin με την σήμανση POWER. Η λειτουργία του καθενός έχει ως εξής:

- Το πρώτο, με την ένδειξη RESET, όταν γειωθεί έχει ως αποτέλεσμα την επανεκκίνηση του Arduino.
- Το δεύτερο, με την ένδειξη 3.3V, μπορεί να τροφοδοτήσει τα εξαρτήματά μας με τάση 3.3V. Η τάση αυτή δεν προέρχεται από την εξωτερική τροφοδοσία αλλά παράγεται από τον ελεγκτή Serial-over-USB και έτσι η μέγιστη ένταση που μπορεί να παρέχει είναι μόλις 50mA.
- Το τρίτο, με την ένδειξη 5V, μπορεί να τροφοδοτήσει τα εξαρτήματά μας με τάση 5V. Ανάλογα με τον τρόπο τροφοδοσίας του ίδιου του Arduino, η τάση αυτή προέρχεται είτε άμεσα από την θύρα USB, είτε από την εξωτερική τροφοδοσία αφού αυτή περάσει από ένα ρυθμιστή τάσης για να την «φέρει» στα 5V.
- Το τέταρτο και το πέμπτο pin, με την ένδειξη GND, είναι φυσικά γειώσεις.
- Το έκτο και τελευταίο pin, με την ένδειξη Vin έχει διπλό ρόλο. Σε συνδυασμό με το pin γείωσης δίπλα του, μπορεί να λειτουργήσει ως μέθοδος εξωτερικής τροφοδοσίας του Arduino, στην περίπτωση που δεν μας βολεύει να χρησιμοποιήσουμε την υποδοχή του φισ των 2.1mm. Αν όμως έχουμε ήδη συνδεδεμένη εξωτερική τροφοδοσία μέσω του φισ, μπορούμε να χρησιμοποιήσουμε αυτό το pin για να τροφοδοτήσουμε εξαρτήματα με την πλήρη τάση της εξωτερικής τροφοδοσίας (7~12V), πριν αυτή περάσει από τον ρυθμιστή τάσης όπως γίνεται με το pin των 5V.

Το Arduino μπορεί να τροφοδοτηθεί με ρεύμα είτε από τον υπολογιστή μέσω της σύνδεσης USB, είτε από εξωτερική τροφοδοσία που παρέχεται μέσω μιας υποδοχής φισ των 2.1mm (θετικός πόλος στο κέντρο) και βρίσκεται στην κάτω-αριστερή γωνία του Arduino.

Για να μην υπάρχουν προβλήματα, η εξωτερική τροφοδοσία πρέπει να είναι από 7 ως 12V και μπορεί να προέρχεται από ένα κοινό μετασχηματιστή του εμπορίου, από μπαταρίες ή οποιαδήποτε άλλη πηγή DC.

ΕΠΙΚΟΙΝΩΝΙΑ ANDROID ΜΕ ARDUINO ΕΦΑΡΜΟΓΕΣ

Πάνω στην πλακέτα του Arduino υπάρχει ένας διακόπτης micro-switch και 4 μικροσκοπικά LED επιφανειακής στήριξης. Η λειτουργία του διακόπτη (που έχει την σήμανση RESET) και του ενός LED με την σήμανση POWER είναι προφανής. Τα δύο LED με τις σημάνσεις TX και RX, χρησιμοποιούνται ως ένδειξη λειτουργίας του σειριακού interface, καθώς ανάβουν όταν το Arduino στέλνει ή λαμβάνει (αντίστοιχα) δεδομένα μέσω USB.

1.3 JY-MCU BLUETOOTH

Το Bluetooth που χρησιμοποιούμε στην εργασία είναι το JY-MCU το οποίο και πρέπει να προγραμματίσουμε ώστε να μπορεί να συγχρονιστεί με το Arduino μας.



Εικόνα 1.3 : Bluetooth JY-MCU

Για να προγραμματίσουμε το Bluetooth υπάρχουν οι AT Commands όπου ανάλογα την έκδοση της συσκευής διαφέρουν. Μερικές από αυτές είναι:

- AT+VERSION?, ως απάντηση θα έχουμε την έκδοση του
- AT+BAUDx, ορίζει το baud rate του Bluetooth (όπου x = 1, 2, 3, 4, 5, 6, 7, 8 → 1200, 2400, 4800, 9600, 19200, 38400, 57600, 115200)
- AT+NAMEname, ορίζει το καινούργιο όνομα της συσκευής μας, μέχρι 20 χαρακτήρες

ΕΠΙΚΟΙΝΩΝΙΑ ANDROID ΜΕ ARDUINO ΕΦΑΡΜΟΓΕΣ

- AT+PINxxxx, ορίζουμε καινούργιο κωδικό πρόσβασης (by default είναι 1234)
- AT+ROLE=x, θέτουμε ρόλο στη συσκευή (Default=0, Slave=0, Master=1, Slave-Loop role=2)
- AT+PIO=10, θέτουμε το PIO10 σε high level
- AT+ORGL, επαναφορά εργοστασιακών ρυθμίσεων
- AT+ADDR?, μας επιστρέφει την διεύθυνση του Bluetooth σε Hex
- AT+PIO=<x>,<y>, όπου x=πόρτα και y=κατάσταση αυτής. Οπότε αν είναι 1 τότε θα είναι σε high level αλλιώς με 0 σε low
- AT+RMAAD, διαγράφει όλες τις πιστοποιημένες συσκευές που υπάρχουν στη λίστα του Bluetooth
- AT+FSAD=<x>, το Bluetooth αναζητά τη συσκευή με τη συγκεκριμένη διεύθυνση x ώστε να συνδεθεί
- AT+ADCN?, επιστρέφει το συνολικό αριθμό των αποθηκευμένων συσκευών που υπάρχουν στη λίστα του Bluetooth
- AT+MRAD?, επιστρέφει τη διεύθυνση της συσκευής που συνδέθηκε τελευταία
- AT+STATE?, απαντάει με την κατάσταση του Bluetooth όπως INITIALIZED, READY, PAIRABLE, PAIRED, INQUIRING, CONNECTING, CONNECTED, DISCONNECTED

Εάν ως απάντηση επιστρέφει Hex αριθμούς τότε πρέπει να συμβουλευτούμε τον παρακάτω πίνακα

ΕΠΙΚΟΙΝΩΝΙΑ ANDROID ΜΕ ARDUINO ΕΦΑΡΜΟΓΕΣ

Πίνακας 1.1 : Πίνακας σφαλμάτων των AT Commands

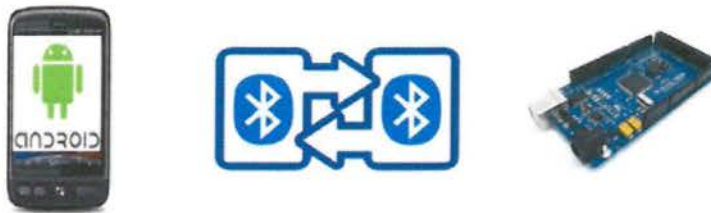
error_code (Hexadecimal)	Πληροφορίες
0	AT command error
1	Default result
2	PSKEY write error
3	Too long length of device name (more than 32 bytes).
4	No device name
5	Bluetooth address: NAP is too long
6	Bluetooth address: UAP is too long
7	Bluetooth address: LAP is too long
8	No PIO number's mask
9	No PIO number
A	No Bluetooth devices.
B	Too length of devices
C	No inquire access code
D	Too long length of inquire access code
E	Invalid inquire access code
F	The length of passkey is 0.
10	Too long length of passkey (more than 16 bytes)
11	Invalid module role
12	Invalid baud rate
13	Invalid stop bit
14	Invalid parity bit
15	Authentication device is not at the pair list.
16	SPP lib hasn't been initialized
17	SPP lib has been repeated initialization
18	Invalid inquire mode
19	Too long inquire time
1A	No Bluetooth address
1B	Invalid safe mode

1C	Invalid encryption mode
----	-------------------------

1.4 AMARINO

Το Amarino είναι μια εφαρμογή Android που δημιουργεί ένα κανάλι επικοινωνίας μεταξύ ενός κινητού τηλεφώνου και ενός εξωτερικού κυκλώματος (στην προκείμενη περίπτωση του Arduino) μέσω Bluetooth.

Το Amarino αποτελείται από δύο συστατικά στοιχεία: μια Android εφαρμογή για το κινητό και μια βιβλιοθήκη για τον Arduino. Τα δύο συστατικά στοιχεία επικοινωνούν μέσω Bluetooth. Ο χρήστης μπορεί να γράψει εφαρμογές για το κινητό και τον Arduino εύκολα χρησιμοποιώντας το κανάλι επικοινωνίας του Amarino.



Εικόνα 1.4 : Επικοινωνία τηλεφώνου-πλακέτας

Όταν μια εφαρμογή εγκαθίσταται σε μια συσκευή, το λειτουργικό αναλύει το αρχείο AndroidManifest.xml για πληροφορίες σχετικά με τις ενσωματωμένες συνιστώσες, τις προτιμήσεις και τις λοιπές πληροφορίες. Το Amarino Manifest είναι:

ΕΠΙΚΟΙΝΩΝΙΑ ANDROID ΜΕ ARDUINO ΕΦΑΡΜΟΓΕΣ

```
<manifest xmlns:android="http://schemas.android.com/apk/res/android" package="at.abraxas.amarino"
android:versionCode="11" android:versionName="0.55">

    <application android:icon="@drawable/icon" android:label="@string/app_name">

        <activity android:name="MainScreen" android:screenOrientation="portrait">

            <intent-filter>

                <action android:name="android.intent.action.MAIN"/>

                <category android:name="android.intent.category.LAUNCHER"/>

            </intent-filter>

        </activity>

        <activity android:name="DeviceDiscovery" android:screenOrientation="portrait"/>
        <activity android:name="EventListActivity" android:screenOrientation="portrait"/>
        <activity android:name="Monitoring"/>
        <service android:name="AmarinoService"/>
        <receiver android:name="RemoteControl">

            <intent-filter>

                <action android:name="amarino.intent.action.CONNECT"/>
                <action android:name="amarino.intent.action.DISCONNECT"/>
                <action
                    android:name="amarino.intent.action.ACTION_GET_CONNECTED_DE
                    VICES"/>

            </intent-filter>

        </receiver>

    </application>

    <uses-sdk android:targetSdkVersion="8" android:minSdkVersion="4"/>
    <uses-permission android:name="android.permission.BLUETOOTH"/></uses-permission>
    <uses-permission android:name="android.permission.BLUETOOTH_ADMIN"/> </uses-
    permission>

    <supports-screens android:anyDensity="true" android:resizeable="true"
    android:smallScreens="true" android:largeScreens="true" android:normalScreens="true">

    </supports-screens>

</manifest>
```

ΕΠΙΚΟΙΝΩΝΙΑ ANDROID ME ARDUINO ΕΦΑΡΜΟΓΕΣ

Κοιτώντας το application παρατηρούμε τα εξής:

- Το `Amarino` αποτελείται από 4 δραστηριότητες, μια υπηρεσία παρασκηνίου και ένα δέκτη εκπομπής.
- Μετά τη δήλωση του εικονιδίου και του τίτλου της εφαρμογής, δηλώνεται ότι η `MAINSCREEN` της εφαρμογής λειτουργεί μόνο σε `portrait mode`.
- Το `intent filter` καθορίζει ποιες ενέργειες (`actions`) θα ενεργοποιήσουν τη συνιστώσα. Η ενέργεια `android.intent.action.MAIN` καθορίζει ότι αυτή η δραστηριότητα θα ενεργοποιηθεί όταν ο χρήστης επιλέξει το εικονίδιο της και η `android.intent.category.LAUNCHER` ότι η δραστηριότητα θα είναι ορατή μόλις ενεργοποιηθεί.
- Οι επόμενες τρεις δραστηριότητες δηλώνουν τις τρεις επιπλέον οθόνες που υπάρχουν στο `Amarino`, αλλά επειδή δεν έχουν `intent filters` δεν μπορούν να ενεργοποιηθούν από κάποια συνιστώσα έξω από το `Amarino`.
- Μετά δηλώνεται η υπηρεσία για να είναι ενημερωμένο το λειτουργικό για την παρουσία της.
- Μετά δηλώνεται ένας δέκτης εκπομπής ο οποίος θα ενεργοποιείται μόλις ληφθούν μηνύματα του τύπου `CONNECT`, `DISCONNECT`, `ACTION_GET_CONNECTED_DEVICES`. Οι δέκτες εκπομπής επιτρέπεται να καθορίζουν ένα `intent filter` κατά τη διάρκεια της εκτέλεσής τους. Όμως, όταν το `intent filter` δηλώνεται στο αρχείο `manifest`, το λειτουργικό σύστημα θα φροντίσει για την έναρξη της συνιστώσας του δέκτη και θα καλέσει την `onReceive callback method` μόλις δεχτεί ένα μήνυμα που ταιριάζει με το `intent filter`.
- Το `uses-sdk` λέει στο λειτουργικό ποιες εκδόσεις του αναπτυξιακού υποστηρίζονται.
- Το `supports-screens` καθορίζει ποια μεγέθη οθονών υποστηρίζονται.
- Το `uses-permission` δηλώνει ότι η εφαρμογή χρειάζεται από το λειτουργικό δύο Bluetooth άδειες τις οποίες χρειάζεται το `Amarino` για

ΕΠΙΚΟΙΝΩΝΙΑ ANDROID ΜΕ ARDUINO ΕΦΑΡΜΟΓΕΣ

να έχει πρόσβαση στο υλικό του Bluetooth.

ΕΠΙΚΟΙΝΩΝΙΑ ANDROID ΜΕ ARDUINO ΕΦΑΡΜΟΓΕΣ

Το Amarino toolkit έχει δύο συστατικά στοιχεία: Μια εφαρμογή Android και μια βιβλιοθήκη για τον Arduino. Το Amarino λειτουργεί σαν ένα κανάλι επικοινωνίας επιτρέποντας την ανταλλαγή δεδομένων μεταξύ ενός κινητού τηλεφώνου και ενός μικροελεγκτή μέσω του Bluetooth. Δημιουργεί μια αόρατη στο χρήστη ανταλλαγή δεδομένων η οποία βασίζεται στη δημιουργία γεγονότων (events). Από την άλλη μεριά, ο Arduino με τη βιβλιοθήκη του έχει ένα μηχανισμό κλήσης μεθόδων μόλις υπάρξει είσοδος ενός δεδομένου (callbackmethods). Ο χρήστης χρησιμοποιεί ξεχωριστές μεθόδους ανάλογα με τον τύπο των δεδομένων εισόδου (π.χ. δεδομένα που προέρχονται από το επιταχυνσιόμετρο του κινητού, από το πληκτρολόγιο, κλπ). Επίσης διαθέτει μεθόδους για την εξαγωγή των δεδομένων αφού διαχωρίσει τα γεγονότα εισόδου και οδηγήσει στην κατάλληλη συνάρτηση. Επίσης διαθέτει μεθόδους για την αποστολή δεδομένων στο κινητό.

Η εφαρμογή Android είναι εφοδιασμένη με ένα GUI (Graphical User Interface) που επιτρέπει στο χρήστη να χειριστεί τις διάφορες συνδέσεις Bluetooth, να δημιουργήσει μια σειρά γεγονότων και να στείλει δεδομένα για κάθε γεγονός στον Arduino. Έτσι μπορεί να διαχειριστεί όλους τους αισθητήρες ενός κινητού, όπως επιταχυνσιόμετρο, αισθητήρα φωτός, πυξίδα, κλπ.

2 Περιγραφή Λογισμικού

2.1 Arduino IDE

Για την διαχείριση του Arduino από τον υπολογιστή μας, παρέχεται ολοκληρωμένο περιβάλλον ανάπτυξης (IDE) το οποίο μπορούμε να βρούμε στο επίσημο site, <http://arduino.cc/en/Main/Software> .

Το Arduino IDE είναι βασισμένο σε Java και συγκεκριμένα παρέχει:

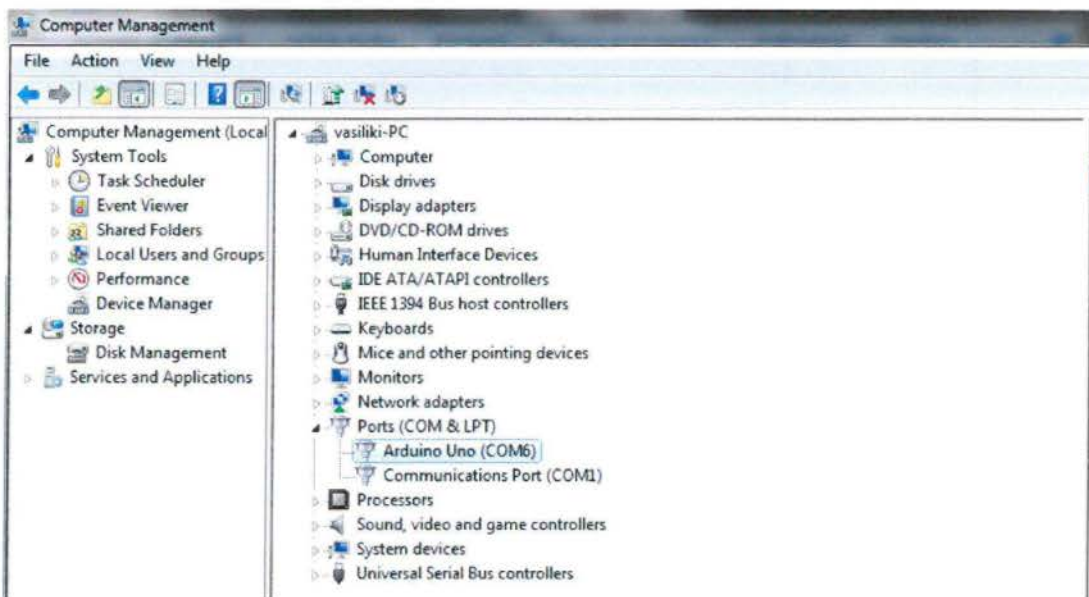
- ένα πρακτικό περιβάλλον για την συγγραφή των προγραμμάτων μας
- αρκετά έτοιμα παραδείγματα,
- μερικές έτοιμες βιβλιοθήκες για προέκταση της γλώσσας και για να χειριζόμαστε εύκολα μέσα από τον κώδικά μας τα εξαρτήματα που συνδέουμε στο Arduino,
- τον compiler για την μεταγλώττιση των προγραμμάτων (sketch) μας,
- ένα serial monitor που παρακολουθεί τις επικοινωνίες της σειριακής (USB), αναλαμβάνει να στείλει αλφαριθμητικά της επιλογής μας στο Arduino μέσω αυτής και είναι ιδιαίτερα χρήσιμο για το debugging των sketch μας
- και την επιλογή να ανεβάσουμε το μεταγλωττισμένο sketch στο Arduino.

Για τα δύο τελευταία χαρακτηριστικά βέβαια, το Arduino πρέπει να έχει συνδεθεί σε μια από τις θύρες USB του υπολογιστή και, λόγω του ελεγκτή Serial-over-USB, θα πρέπει να αναγνωριστεί από το λειτουργικό μας σύστημα ως εικονική σειριακή θύρα.

ΕΠΙΚΟΙΝΩΝΙΑ ANDROID ΜΕ ARDUINO ΕΦΑΡΜΟΓΕΣ

Για την σύνδεση θα χρειαστούμε ένα καλώδιο USB από Type A σε Type B, όπως αυτό των εκτυπωτών.

Επιλέγουμε το κατάλληλο αρχείο από το επίσημο site: <http://arduino.cc/en/Main/Software> το, για Windows στην περίπτωση μας, και το σώζουμε στον υπολογιστή μας. Αντιγράφουμε το φάκελο στο Program Files, και από εκεί δημιουργούμε συντόμευση για δική μας ευκολία. Συνδέουμε την πλακέτα, μέσω usb, με τον υπολογιστή μας αλλά η αυτόματη εγκατάσταση θα είναι ανεπιτυχής αφού δεν θα υπάρχουν οι κατάλληλοι drivers. Ανοίγουμε το Device Manager και με δεξί κλικ πάνω στη συσκευή επιλέγουμε Update device drivers. Το επόμενο βήμα είναι να επιλέξουμε το μονοπάτι που βρίσκονται οι drivers, και αυτό είναι 'c:/Program Files/arduino'. Στο τέλος της διαδικασίας θα φανεί η συσκευή μας 'Arduino Uno(comx)'.



Εικόνα 2.1 :Device Manager

Ανοίγοντας το Arduino πρέπει να επιλέξουμε με ποια πόρτα θα επικοινωνούμε με την πλακέτα. Πηγαίνουμε «Tools-Board» και επιλέγουμε Arduino Uno (Εικόνα 2.3) καθώς και «Tools-Serial Port» και επιλέγουμε το "comx" (Εικόνα 2.4) που μπήκε αυτόματα κατά την εγκατάσταση της καινούργιας συσκευής, που φαίνεται στο Device Manager.

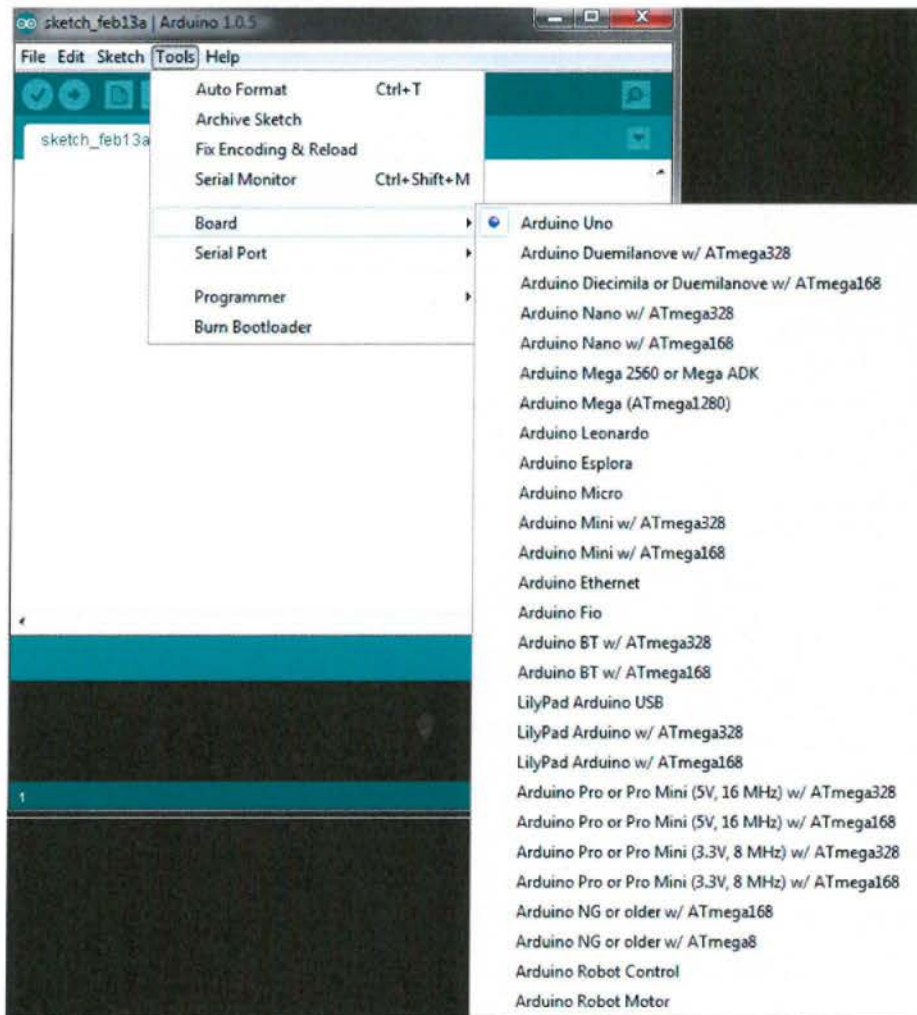
ΕΠΙΚΟΙΝΩΝΙΑ ANDROID ΜΕ ARDUINO ΕΦΑΡΜΟΓΕΣ

Τα εργαλεία του Arduino Uno είναι τα παρακάτω



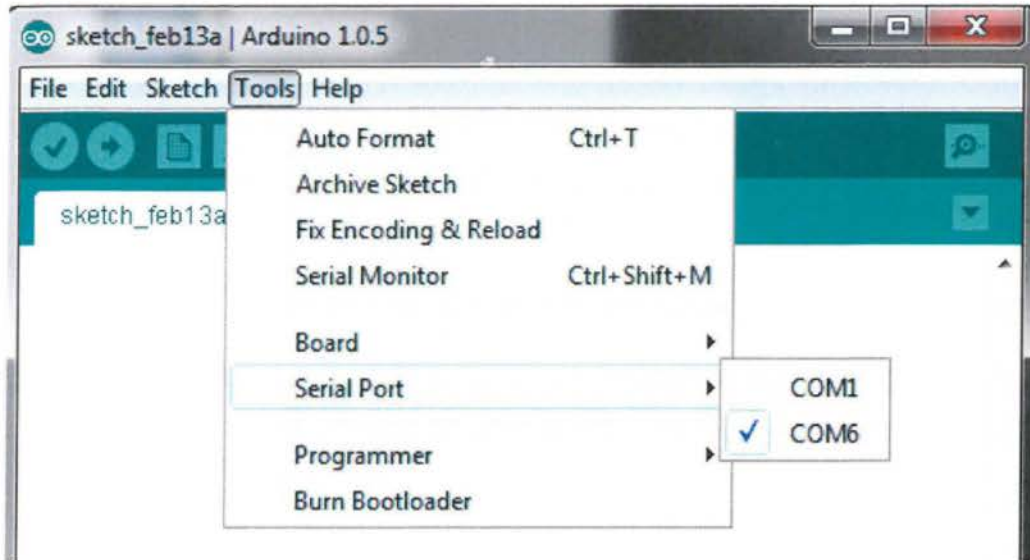
Εικόνα 2.2 : Arduino IDE

όπου το καθένα από τα αριστερά είναι για επαλήθευση του προγράμματος, φόρτωση αυτού στη πλακέτα μας, άνοιγμα νέου παράθυρου για συγγραφή κώδικα, άνοιγμα αποθηκευμένου κώδικα και τέλος για αποθήκευση του εγγεγραμμένου κώδικα.



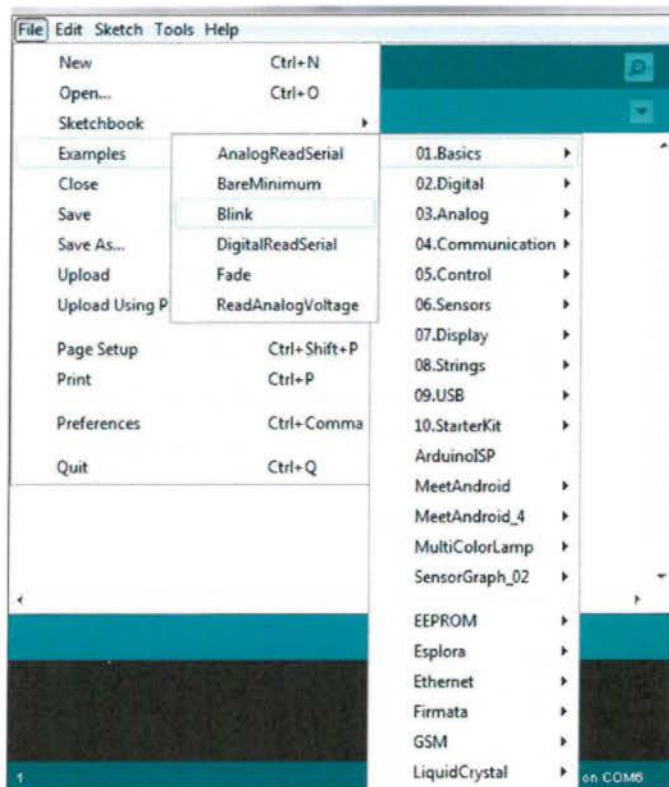
Εικόνα 2.3 : Αρχικοποίηση περιβάλλοντος

ΕΠΙΚΟΙΝΩΝΙΑ ANDROID ΜΕ ARDUINO ΕΦΑΡΜΟΓΕΣ



Εικόνα 2.4 : Αρχικοποίηση περιβάλλοντος

Για να δούμε αν υπάρχει επικοινωνία, μεταξύ υπολογιστή και πλακέτας, θα κάνουμε ένα έτοιμο παράδειγμα ώστε να δούμε και την άμεση ανταπόκριση της πλακέτας μας. Επιλέγουμε «File-Examples-Basics-Blink». Αυτόματα ανοίγει καινούργιο παράθυρο με το έτοιμο παράδειγμα.



Εικόνα 2.5 : Βασικό παράδειγμα

ΕΠΙΚΟΙΝΩΝΙΑ ANDROID ΜΕ ARDUINO ΕΦΑΡΜΟΓΕΣ

Αφού έχουμε ανοίξει το παράδειγμα «Blink», το φορτώνουμε στην πλακέτα κι αν όλα πήγαν καλά, στην μαύρη οθόνη όπου φαίνεται η πρόοδος, θα εμφανιστεί μήνυμα πως έγινε επιτυχώς. Αμέσως θα δούμε στην πλακέτα μας να αναβοσβήνει συνέχεια ένα λαμπάκι. Άρα η επικοινωνία μεταξύ υπολογιστή και πλακέτας είναι σωστή.

2.2 Διαμόρφωση Bluetooth

Για να μπορέσουμε να δουλέψουμε στην εργασία μας, θα πρέπει να του αλλάξουμε baud rate, από 9600 που είναι εργοστασιακή ρύθμιση, σε 57600. Η διαδικασία είναι η εξής:

- Τροφοδοτούμε το Arduino μέσω usb και συνδέουμε το Bluetooth στα 3,3V και Ground του Arduino
- Συνδέουμε το Rx και Tx του Bluetooth με το Rx και Tx του Arduino αντίστοιχα
- Για να βάλουμε το Bluetooth σε programming mode, συνδέουμε στο pin KEY του Bluetooth μια αντίσταση 10 KΩ προς την Vcc.
- Ανοίγουμε το περιβάλλον Arduino IDE και επιλέγουμε "Tools-Serial Monitor" και θέτουμε "No line ending" και "9600".
- Εισάγουμε AT και πατάμε Send. Το Bluetooth απαντά με OK που σημαίνει πως έχουμε σωστή επικοινωνία.
- Πληκτρολογούμε «AT+NAMEvasiliki» και πατάμε Send. Η απάντηση που περιμένουμε από το Bluetooth είναι Oksetname.
- Επίσης πληκτρολογούμε «AT+BAUD7» και πατάμε Send. Η απάντηση που περιμένουμε από το Bluetooth είναι OKbaudrate
- Απομακρύνουμε τη τροφοδοσία από το pin KEY και κάνοντας reset στο Bluetooth είναι έτοιμο για χρήση με νέα χαρακτηριστικά

Θα πρέπει να δώσουμε προσοχή στη συνδεσιμότητα του Bluetooth και του Arduino. Κατά τη διάρκεια του προγραμματισμού, όπως είπαμε πιο πάνω, τα Tx και Rx του ενός συνδέονται αντίστοιχα με τα Tx και Rx του άλλου. Όμως κατά τη διάρκεια της φόρτωσης ενός προγράμματος στο Arduino, θα πρέπει τα Tx/Rx του Bluetooth να είναι

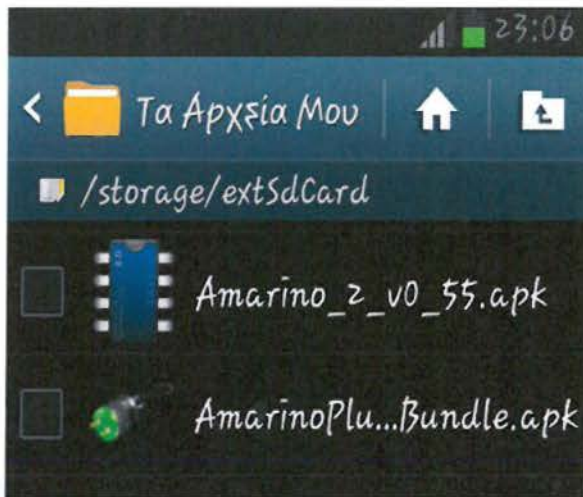
ΕΠΙΚΟΙΝΩΝΙΑ ANDROID ΜΕ ARDUINO ΕΦΑΡΜΟΓΕΣ

αποσυνδεδεμένα. Στην περίπτωση βέβαια που το πρόγραμμα μας τρέχει, για να υπάρξει επικοινωνία μεταξύ Android συσκευής και Arduino θα πρέπει το Tx/Rx του Bluetooth να είναι συνδεδεμένο με το Rx/Tx αντίστοιχα του Arduino.

2.3 Amarino

Για να εγκαταστήσουμε το πρόγραμμα στο τηλέφωνο μας, 'κατεβάζουμε' την εφαρμογή (Amarino_2.apk) από την επίσημη ιστοσελίδα του Amarino, <http://www.amarino-toolkit.net/index.php/download.html>, Amarino Plug-in Bundle καθώς και τις έτοιμες βιβλιοθήκες του, MeetAndroid - Arduino Library και Amarino Library .

Τα μεταφέρουμε στην Android συσκευή μας μέσω usb και τα εγκαθιστούμε ξεκινώντας από το Amarino_2.apk και μετά το Amarino Plug-in Bundle με τον ίδιο τρόπο.



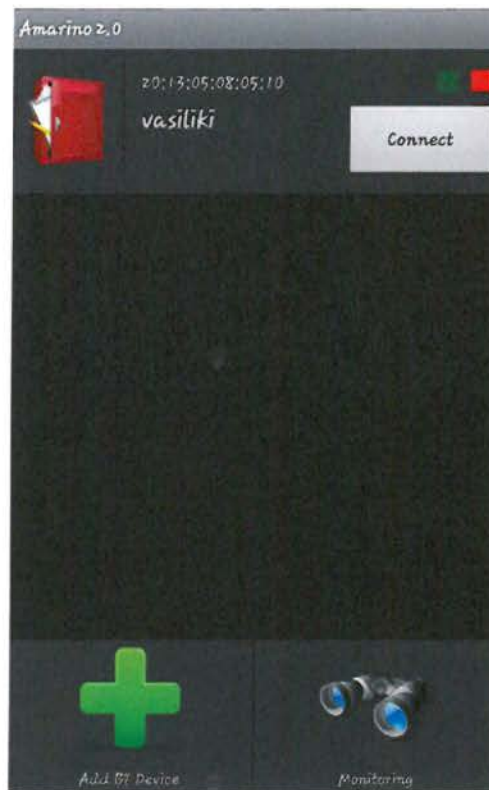
Εικόνα 2.6 : Εγκατάσταση Amarino ,Amarino Plug-in Bundle

ΕΠΙΚΟΙΝΩΝΙΑ ANDROID ΜΕ ARDUINO ΕΦΑΡΜΟΓΕΣ



Εικόνα 2.7 : Εγκατάσταση ως Package Installer

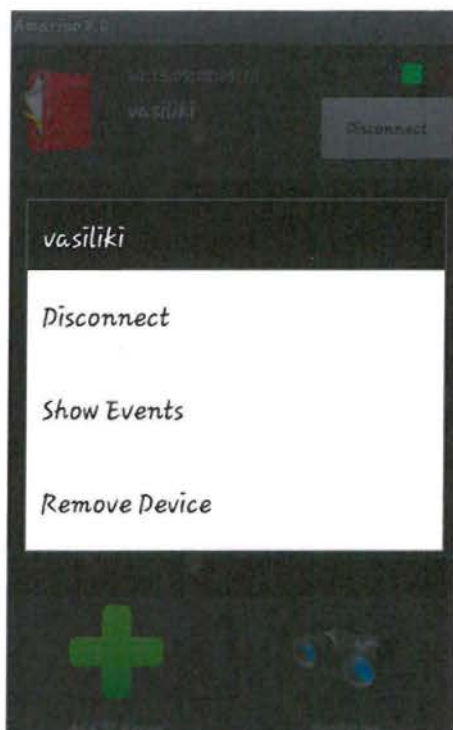
Ανοίγοντας το Amarino επιλέγουμε το “Add BT Device” και αφού βρούμε τη συσκευή, vasiliki στην περίπτωση μας, την επιλέγουμε, βάζουμε τον κωδικό 123 και πατάμε Connect. Πλέον στην αρχική οθόνη του Amarino θα υπάρχει αποθηκευμένη η συσκευή μας αναγράφοντας πάνω από το όνομα της την διεύθυνση Mac της.



Εικόνα 2.8 : Αρχική οθόνη Amarino

ΕΠΙΚΟΙΝΩΝΙΑ ANDROID ΜΕ ARDUINO ΕΦΑΡΜΟΓΕΣ

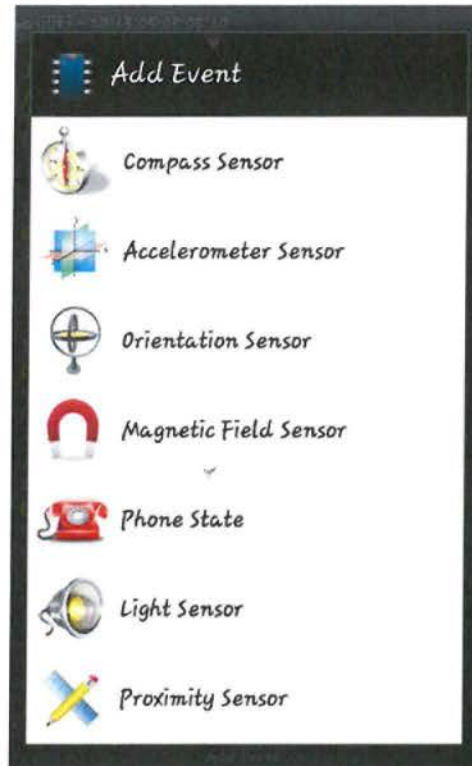
Πατώντας το Connect ανοίγει αυτόματα το Bluetooth του κινητού τηλεφώνου μας και συνδεόμαστε με την συσκευή μας. Κάθε φορά που θα κλείνουμε και θα ανοίγουμε ξανά το Amarino η συσκευή αυτή θα υπάρχει. Ο μόνος τρόπος για να την διαγράψουμε οριστικά από τη λίστα μας είναι πατώντας παρατεταμένα στο όνομα της, και επιλέγοντας "Remove Device".



Εικόνα 2.9 :Βασικές επιλογές Amarino

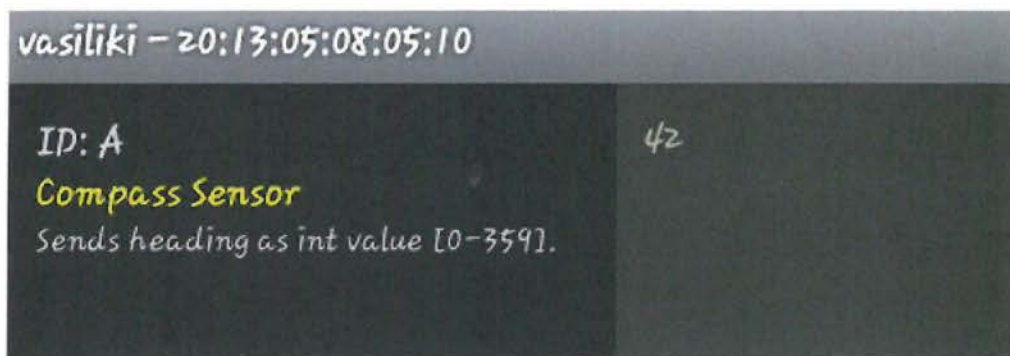
Η επιλογή "Show Events" ανοίγει ένα καινούργιο παράθυρο με έτοιμες εφαρμογές του Amarino όπως φαίνεται στην παρακάτω εικόνα. Για κάθε εφαρμογή μπορούμε να επιλέξουμε την ταχύτητα που θα μας στέλνει τα δεδομένα καθώς και τον τρόπο απεικόνισης τους (text, graph, bars).

ΕΠΙΚΟΙΝΩΝΙΑ ANDROID ΜΕ ARDUINO ΕΦΑΡΜΟΓΕΣ



Εικόνα 2.10 :Έτοιμες εφαρμογές Amarino

Αναλυτικά το Compass Sensor στέλνει μια τιμή από 0-359 όπου το 0 αντιστοιχεί στο βορρά, το 90 στην ανατολή, το 180 στο νότο και το 270 στη δύση. Ένα παράδειγμα αφήνοντας σταθερή τη τηλεφωνική μας συσκευή είναι το παρακάτω.



Εικόνα 2.11 :Παράδειγμα Compass Sensor (text)

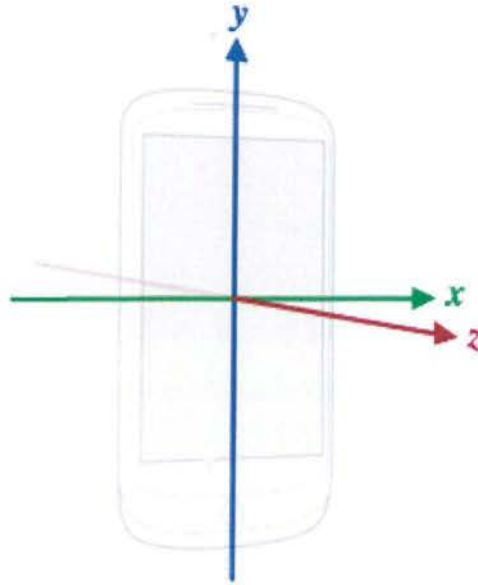


Εικόνα 2.12 :Παράδειγμα Compass Sensor(graph)



Εικόνα 2.13 :Παράδειγμα Compass Sensor(bars)

Το Accelerometer Sensor μας στέλνει σε μορφή κειμένου τις συντεταγμένες που αλλάζουν καθώς κινούμε το κινητό μας τηλέφωνο αλλιώς σε γραφική με τρία διαφορετικά χρώματα που αντιστοιχούν στους άξονες x, y, z όπως φαίνεται στην παρακάτω εικόνα.



Εικόνα 2.14 :Επεξήγηση συντεταγμένων

Το Orientation Sensor δείχνει το τον προσανατολισμό, το πρώτο ψηφίο αντιστοιχεί στις συντεταγμένες και τα άλλα δυο στην μετακίνηση του τηλεφώνου στους άξονες x, y.

Το Magnetic Field Sensor μετράει το μαγνητικό πεδίο του περιβάλλοντος και επιστρέφει τρεις τιμές σε micro-Tesla.

Το Phone State στέλνει έναν αριθμό που αντιστοιχεί στην κατάσταση που βρίσκεται το τηλέφωνο την συγκεκριμένη χρονική στιγμή. Αν το τηλέφωνο βρίσκεται σε αδράνεια επιστρέφει τον αριθμό 0, αν χτυπάει τον αριθμό 1 και όταν μιλάμε τον αριθμό 2.

Το Light Sensor επιστρέφει έναν αριθμό που αντιστοιχεί στο επίπεδο φωτισμού του περιβάλλοντος εργασίας μας.

Το Proximity Sensor επιστρέφει μια τιμή από το 0-5 από default . Αυτή τη λειτουργία δεν την έχουν όλα τα κινητά τηλέφωνα που διαθέτουν touch screen. Συγκεκριμένα είναι ένας αισθητήρας που διατάσσει την οθόνη να σβήσει όταν την ακουμπάμε στο αυτί μας και να ανοίξει όταν την απομακρύνουμε από αυτό. Η τιμή αυτή που επιστρέφει είναι η απόσταση του αισθητήρα αυτού από το αυτί μας.

Το Battery Level επιστρέφει έναν αριθμό από το 0-100 που αντιστοιχεί στο ποσοστό της μπαταρίας που έχει απομείνει για χρήση.

Το Time Tick στέλνει κάθε λεπτό έναν αριθμό που αντιστοιχεί στα λεπτά της

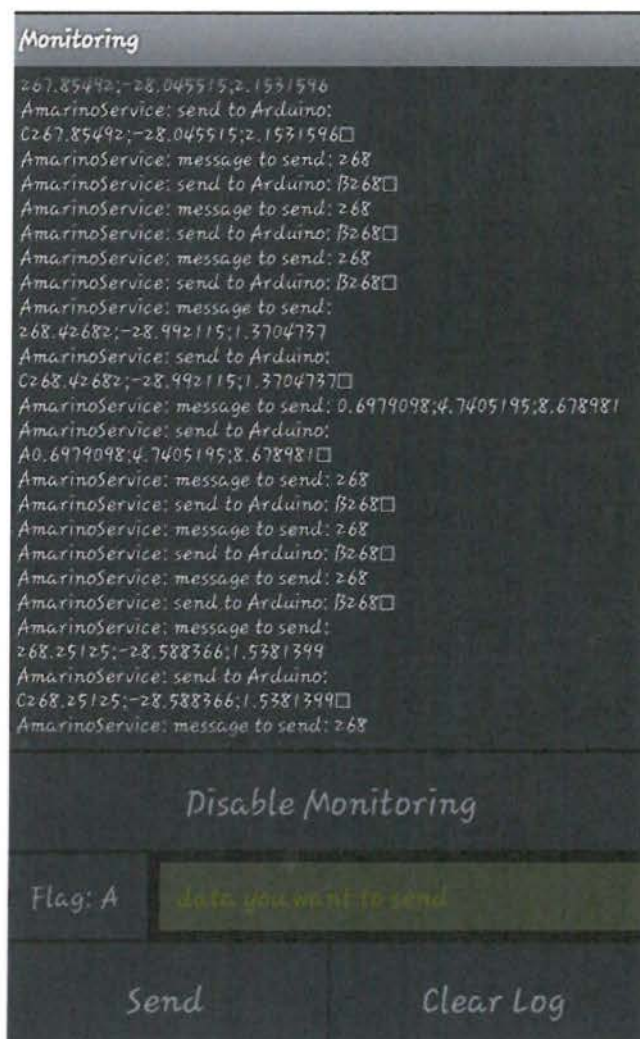
ΕΠΙΚΟΙΝΩΝΙΑ ANDROID ΜΕ ARDUINO ΕΦΑΡΜΟΓΕΣ

πραγματικής ώρας.

Το Test Event στέλνει τυχαίους αριθμούς από το 0-255 στο Arduino για απλές δοκιμές.

Το Receive SMS στέλνει στο Arduino τους 30 πρώτους χαρακτήρες ενός γραπτού μηνύματος σε μορφή String μήκους από 0-30

Από την αρχική οθόνη, ο χρήστης μπορεί να επιλέξει το πλήκτρο Monitoring, οπότε εμφανίζεται μια οθόνη η οποία δείχνει τα δεδομένα εξόδου ή εισόδου και μηνύματα για λάθη ή debugging. Επιλέγοντας τη σημαία ενός γεγονότος (event ID) και εισάγοντας ένα μήνυμα, ο χρήστης μπορεί να το στείλει στον Arduino.



Εικόνα 2.15 :Παράδειγμα Monitoring

3 Παραδείγματα

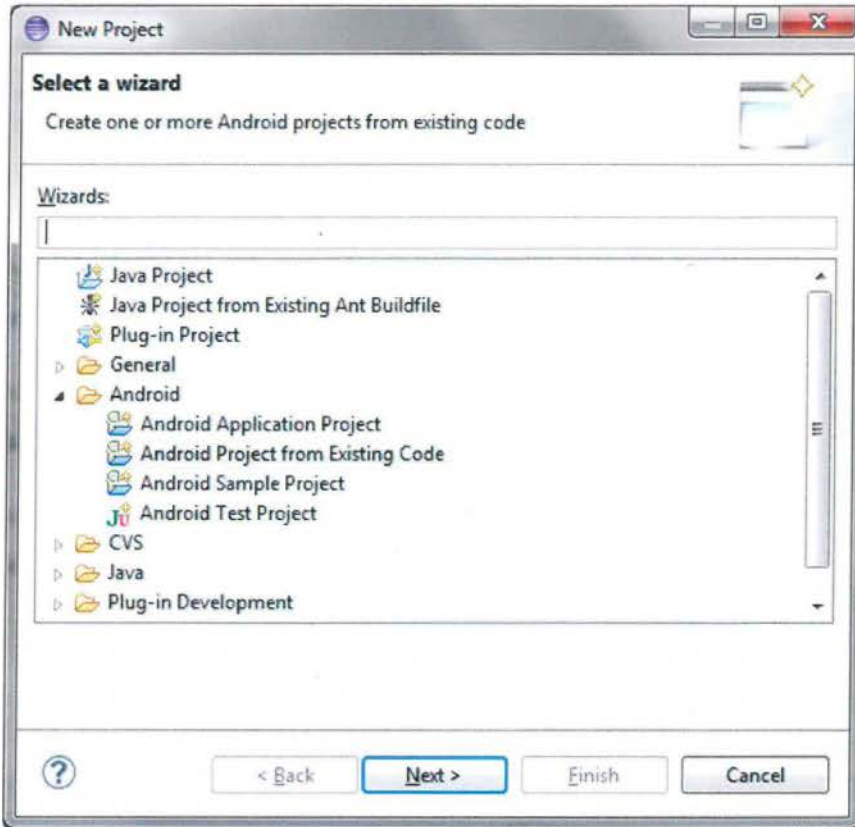
3.1 Εγκατάσταση προγραμμάτων

Για να εγκαταστήσουμε τις εφαρμογές Sensor Graph και Multi Color Lamp στη τηλεφωνική μας συσκευή θα πρέπει να εγκαταστήσουμε στον υπολογιστή μας το Eclipse. Βέβαια θα χρειαστούμε κι άλλα εργαλεία ώστε να τρέξει το πρόγραμμα μας ως Android εφαρμογή. Τα βήματα είναι απλά, όμως θέλει προσοχή στη σειρά των εγκαταστάσεων,

Από το <http://www.eclipse.org/downloads/> κάνουμε λήψη το Eclipse Standar 4.3.1 (στην περίπτωση μας) και μετά το τέλος της εγκατάστασης συνεχίζουμε με το Android SDK από το http://developer.android.com/sdk/index.html?utm_source=weibolife . Παραγοντοποιούμε το eclipse ώστε να περάσουμε το ADT plugin for Eclipse IDE, όπου αναλυτικά τα βήματα αυτά θα τα βρούμε στο <http://developer.android.com/sdk/installing/bundle.html>, προσοχή θέλει να εγκαταστήσουμε τη σωστή έκδοση android, αυτή του κινητού μας τηλεφώνου. Τέλος θα χρειαστεί να 'κατεβάσουμε' τον κώδικα των Sensor Graph και MultiColorLamp από το <http://www.amarino-toolkit.net/index.php/download.html> .

Αφού έχει γίνει αυτή η χρονοβόρα διαδικασία, ανοίγουμε το Eclipse και πατάμε δεξί κλικ στο Project Panel και επιλέγουμε New > Project. Ανοίγει ένα νέο παράθυρο και εκεί επιλέγουμε Android Project from Existing Code και πατάμε Next

ΕΠΙΚΟΙΝΩΝΙΑ ANDROID ΜΕ ARDUINO ΕΦΑΡΜΟΓΕΣ

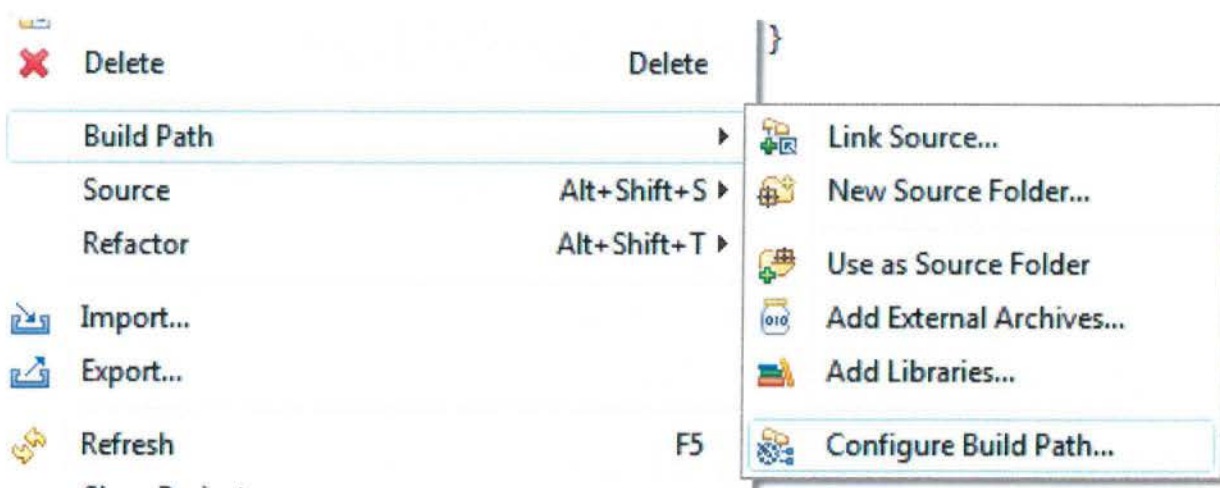


Εικόνα 3.1 : Δημιουργία Android project

Στο επόμενο παράθυρο επιλέγουμε τη διαδρομή που είναι αποθηκευμένος στον υπολογιστή μας ο κώδικας του Sensor Graph και πατάμε Finish. Αμέσως θα φορτωθεί ο κώδικας όπου χρειάζεται μικρές αλλαγές για να το τρέξουμε.

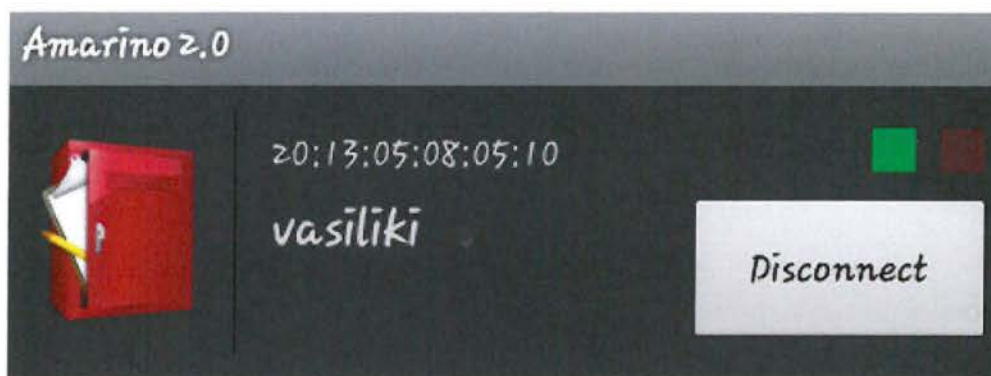
Από το <http://www.amarino-toolkit.net/index.php/download.html> κάνουμε λήψη το AmarinoLibrary όπου εμφανίζεται ως AmarinoLibrary_v0_55.jar. Αυτή είναι μια βιβλιοθήκη που πρέπει να ενσωματωθεί και στα δυο προγράμματα μας. Οπότε κάνουμε δεξί κλικ στο Sensor Graph project και επιλέγουμε Build Path > Configure Build Path

ΕΠΙΚΟΙΝΩΝΙΑ ANDROID ΜΕ ARDUINO ΕΦΑΡΜΟΓΕΣ



Εικόνα 3.2 : Εισαγωγή βιβλιοθήκης AmarinoLibrary

Στο καινούργιο παράθυρο που θα ανοίξει, στην καρτέλα Libraries, με το Add External Jars επιλέγουμε το AmarinoLibrary_v0_55.jar που 'κατεβάσαμε' πιο πριν. Στην καρτέλα Order and Export επιλέγουμε το Android 4.1.2 (στην δική μας περίπτωση) καθώς και τη βιβλιοθήκη που μόλις προσθέσαμε. Επίσης κάνουμε διπλό κλικ στο AndroidManifest.xml και αλλάζουμε την κατάσταση του Debuggable σε True. Τέλος στον κώδικα πρέπει να βάλουμε την MAC Address του Bluetooth που θα χρησιμοποιήσουμε στην εργασία μας ώστε ανοίγοντας το πρόγραμμα να συνδέεται κατευθείαν με τη συσκευή μας. Αυτή τη διεύθυνση την ξέρουμε ήδη από το Amarino όπως φαίνεται παρακάτω



Εικόνα 3.3 : MAC Address όπως φαίνεται στο Amarino

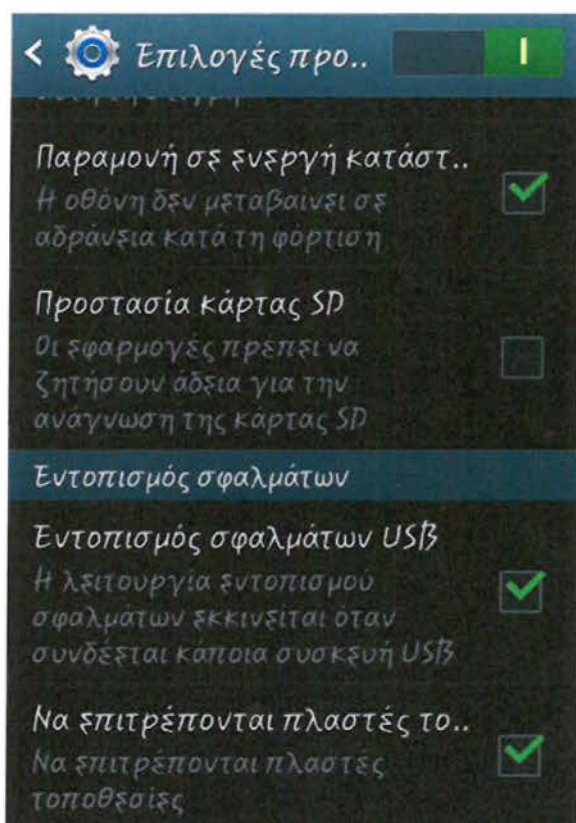
ΕΠΙΚΟΙΝΩΝΙΑ ANDROID ΜΕ ARDUINO ΕΦΑΡΜΟΓΕΣ

Οπότε πάμε στον κώδικα και το μόνο που αλλάζουμε είναι το DEVICE_ADDRESS

```
private static final String DEVICE_ADDRESS = "20:13:05:08:05:10";
```

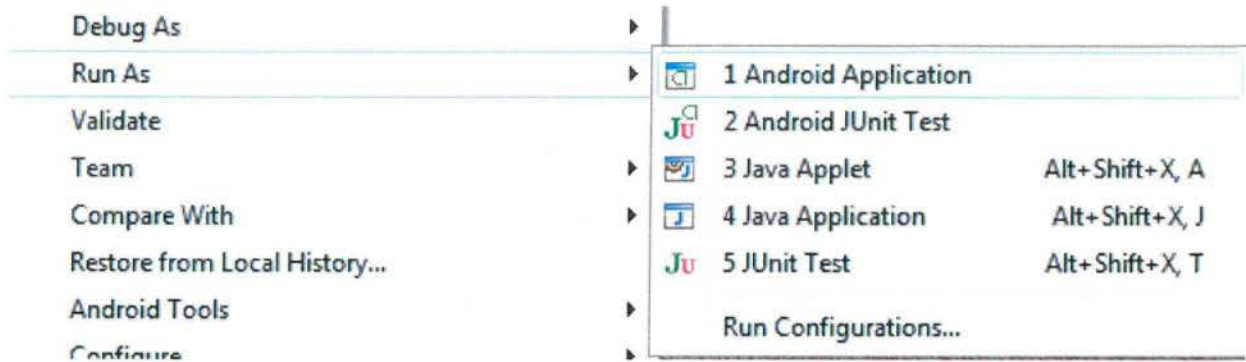
Εικόνα 3.4 : Εγγραφή της MAC Address στον κώδικα

Αφού αποθηκεύσουμε τις αλλαγές, συνδέουμε τη τηλεφωνική μας συσκευή μέσω usb στον υπολογιστή, πηγαίνουμε Ρυθμίσεις > Επιλογές Προγραμματιστή και αφού το ενεργοποιήσουμε, επιλέγουμε τα παρακάτω



Εικόνα 3.5 : Επιλογές προγραμματιστή στην Android συσκευή

Τέλος, στο Eclipse κάνουμε δεξί κλικ στο φάκελο του προγράμματος και επιλέγουμε Run As > Android Application.



Εικόνα 3.6 : Εγκατάσταση προγραμμάτων ως Android Application

Έτσι έχουμε εγκαταστήσει στο τηλέφωνο μας το Sensor Graph και είναι έτοιμο για χρήση. Την ίδια ακριβώς διαδικασία ακολουθούμε και για το Multi Color Lamp.

3.2 Sensor Graph

Αρχίζοντας από αυτό το παράδειγμα, ανοίγουμε το Arduino και φορτώνουμε τον κώδικα του SensorGraph

```
#include <MeetAndroid.h>

MeetAndroid meetAndroid;

int sensor = A5;

void setup() {
    Serial.begin(57600);
    pinMode(sensor, INPUT);
}

void loop() {
    meetAndroid.receive();
    meetAndroid.send(analogRead(sensor));
    delay(100);
}
```

ΕΠΙΚΟΙΝΩΝΙΑ ANDROID ΜΕ ARDUINO ΕΦΑΡΜΟΓΕΣ

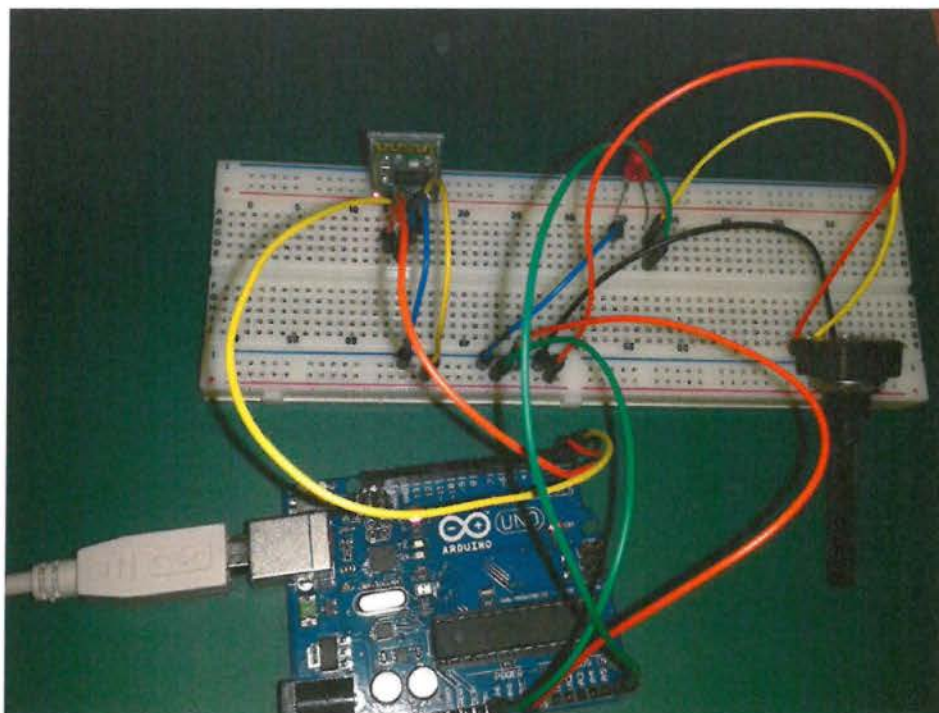
Ίσως κατά την επαλήθευση του προγράμματος να μας βγάλει ένα error 'meetandroid' does not name a type. Σε αυτή την περίπτωση πηγαίνουμε Sketch > Import Library και ψάχνουμε να βρούμε το MeetAndroid. Εάν δεν υπάρχει σημαίνει πως δεν το έχουμε περασμένο σωστά στις βιβλιοθήκες μας, και η διαδικασία είναι Sketch > Import Library > Add Library και επιλέγουμε το αρχείο όπως το κατεβάσαμε, συμπιεσμένο. Με το μήνυμα επιτυχίας σημαίνει πως αν κοιτάξουμε πάλι στο Sketch > Import Library, η βιβλιοθήκη μας θα υπάρχει. Κάνοντας ξανά επαλήθευση του κώδικα δεν θα υπάρχει κανένα λάθος και μπορούμε να το φορτώσουμε στο Arduino μας.

Πριν το φορτώσουμε στον Arduino μας θα πρέπει να συνδέσουμε τα στοιχειά μας. Θα χρειαστούμε ένα απλό led και ένα ποτενσιόμετρο, ώστε σε κάθε αλλαγή που θα κάνουμε σε αυτό, να φαίνεται στην οθόνη του κινητού μας. Κατ' αρχήν συνδέουμε το Bluetooth με το Arduino μας ως εξής:

- VCC Bluetooth με 5V του Arduino Uno
- GND Bluetooth με GND Arduino Uno
- Tx Bluetooth με Rx Arduino Uno
- Rx Bluetooth με Tx Arduino Uno
- Ο αρνητικός πόλος του led στο GND και ο θετικός στο A5 του Arduino
- Συνδέουμε επίσης το ποτενσιόμετρο σε VCC και GND. Το μεσαίο ποδαράκι επίσης στο θετικό πόλο του led ώστε να αυξομειώνουμε την ένταση του.

ΕΠΙΚΟΙΝΩΝΙΑ ANDROID ΜΕ ARDUINO ΕΦΑΡΜΟΓΕΣ

Το κύκλωμα είναι το ακόλουθο.



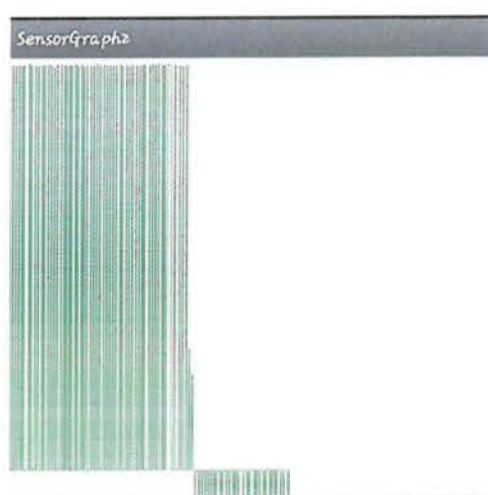
Εικόνα 3.7 : Συνδεσμολογία Sensor Graph

Αυξομειώνοντας το ροοστάτη μας, αυξομειώνεται αντιστρόφως ανάλογα η ένταση του led και παίρνουμε τα παρακάτω αποτελέσματα στο κινητό μας τηλέφωνο

ΕΠΙΚΟΙΝΩΝΙΑ ANDROID ΜΕ ARDUINO ΕΦΑΡΜΟΓΕΣ



Εικόνα 3.8 : Γραφικά η μέγιστη τιμή του ποτενσιόμετρου



Εικόνα 3.9 : Γραφικά η ελάχιστη τιμή του ποτενσιόμετρου

3.3 Multi Color Lamp

Σε αυτή την εφαρμογή, το Bluetooth συνδέεται όπως παραπάνω και οι θετικοί πόλοι των led συνδέονται με τα digital pin 9, 10 και 11. Όπως φαίνεται και στον κώδικα το κόκκινο led στο 9, το πράσινο στο 10 και το μπλε στο 11. Για κάθε led χρησιμοποιούμε μια αντίσταση 220Ω. Βέβαια θα μπορούσαμε να χρησιμοποιήσουμε στην εφαρμογή αυτή ένα led rgb αλλά οι αλλαγές δεν θα ήταν έντονα ορατές.

```
#include <MeetAndroid.h>

MeetAndroid meetAndroid;

int redLed = 9;

int greenLed = 10;

int blueLed = 11;

void setup() {

    Serial.begin(57600);

    meetAndroid.registerFunction(red, 'r');

    meetAndroid.registerFunction(green, 'g');

    meetAndroid.registerFunction(blue, 'b');

    pinMode(redLed, OUTPUT);

    pinMode(greenLed, OUTPUT);

    pinMode(blueLed, OUTPUT);

    digitalWrite(redLed, HIGH);

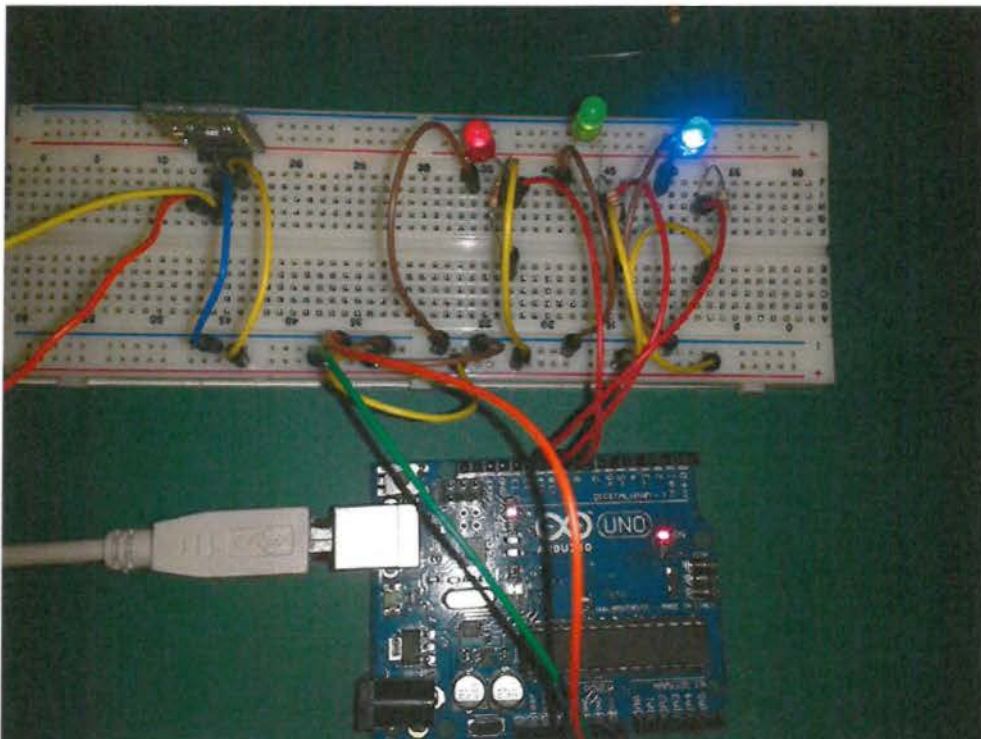
    digitalWrite(greenLed, HIGH);

    digitalWrite(blueLed, HIGH);

}
```


ΕΠΙΚΟΙΝΩΝΙΑ ANDROID ΜΕ ARDUINO ΕΦΑΡΜΟΓΕΣ

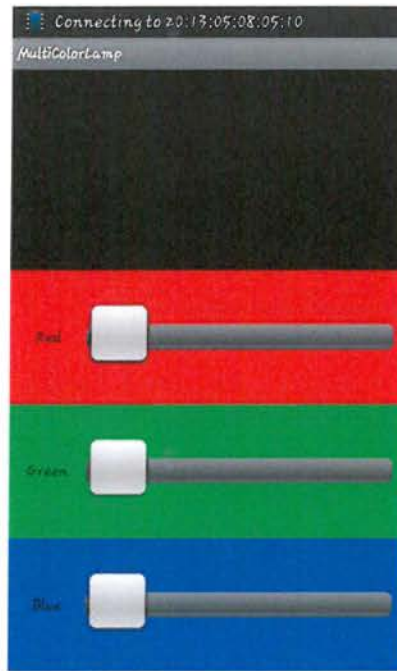
```
void loop() {  
    meetAndroid.receive();  
}  
  
void red(byte flag, byte numofValues) {  
    analogWrite(redLed, meetAndroid.getInt());  
}  
  
void green(byte flag, byte numofValues) {  
    analogWrite(greenLed, meetAndroid.getInt());  
}  
  
void blue(byte flag, byte numofValues) {  
    analogWrite(blueLed, meetAndroid.getInt());  
}  
}
```



Εικόνα 3.10 : Συνδεσμολογία Multi Color Lamp

ΕΠΙΚΟΙΝΩΝΙΑ ANDROID ΜΕ ARDUINO ΕΦΑΡΜΟΓΕΣ

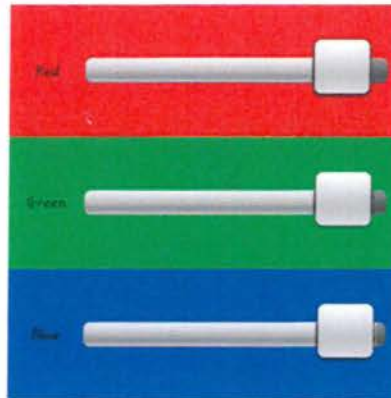
Η αρχική οθόνη του MultiColorLamp στην τηλεφωνική μας συσκευή είναι η παρακάτω.



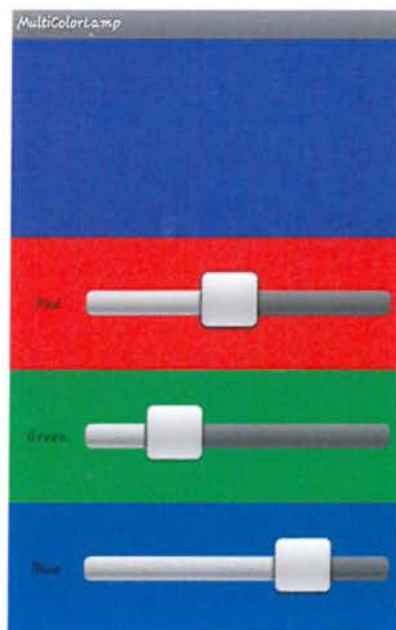
Εικόνα 3.11 : Αρχική οθόνη Multi Color Lamp

Πάνω από την κόκκινη μπάρα απεικονίζεται το χρώμα που θα αντιστοιχεί στο rgb led ανάλογα με τη ένταση που έχουμε δώσει σε κάθε χρώμα. Για παράδειγμα αν είναι όλα κλειστά, το led δεν θα φωτίζει (Εικόνα 3.11) ενώ αν είναι όλα στη μέγιστη τιμή, το rgb led θα είναι λευκό (Εικόνα 3.12). Σε ένα τυχαίο παράδειγμα, θα είναι μώβ όπως φαίνεται παρακάτω (Εικόνα 3.13).

ΕΠΙΚΟΙΝΩΝΙΑ ANDROID ΜΕ ARDUINO ΕΦΑΡΜΟΓΕΣ



Εικόνα 3.12 : Μέγιστη τιμή rgb led



Εικόνα 3.13 : Τυχαίο παράδειγμα(χρώμα) rgb led

ΕΠΙΛΟΓΟΣ

Η παρούσα πτυχιακή εργασία είναι αποτέλεσμα της επιτυχούς ολοκλήρωσης πολλαπλών τομέων της σύγχρονης τεχνολογίας, αυτών της πληροφορικής, της ηλεκτρονικής και του αυτοματισμού. Η χρήση αισθητηρίων οργάνων και προγραμματιζόμενων ελεγκτών από το επίπεδο του ολοκληρωμένου κυκλώματος έως τη συγγραφή προγραμμάτων σε γλώσσες ανώτερου επιπέδου, αποτελεί μια από τις σημαντικότερες κατευθύνσεις που μπορεί κάποιος σπουδαστής ή σπουδάστρια του τμήματος Μηχανικών Ηλεκτρονικών Υπολογιστικών Συστημάτων να ακολουθήσει τόσο κατά τη διάρκεια των σπουδών, όσο και κατά τη μετέπειτα επαγγελματική σταδιοδρομία.

Η ενασχόληση με ένα τόσο μεγάλο εύρος αντικειμένων, που αρμονικά ολοκληρώνουν σύγχρονες τεχνολογικές υπηρεσίες και παρέχονται στο χρήστη τόσο για την καλύτερευση των συνθηκών διαβίωσης, όσο και για την ψυχαγωγία και εκπαίδευση του, είναι μια ιδιαίτερη εμπειρία. Η πολυπλοκότητα του εγχειρήματος, ωθεί στην εκ νέου κατανόηση και πρακτική χρήση παλαιών γνώσεων καθώς και στην αναζήτηση νέων. Οι πολλές στιγμές που βρέθηκα αντιμέτωπη με δύσκολα προβλήματα κατά τη διάρκεια της πτυχιακής αυτής εργασίας συνοδεύτηκαν από στιγμές χαράς καθώς τα προβλήματα αυτά με κόπο και ανακούφιση σιγά-σιγά αντιμετωπίστηκαν.

Η πτυχιακή αυτή εργασία αν και καλύπτει ένα μικρό κομμάτι του αντικειμένου, δύναται να χρησιμοποιηθεί από τον αναγνώστη προκειμένου να κάνει με τη σειρά του, τα πρώτα βήματα στη δημιουργία κυκλωμάτων και ανάπτυξη εφαρμογών για την πλατφόρμα του κυκλώματος Arduino αλλά και τη γενικότερη κατανόηση παρόμοιων αλλά και διαφορετικών τεχνολογιών.

ΒΙΒΛΙΟΓΡΑΦΙΑ

- Arduino Uno, <http://arduino.cc/en/Main/Software>
- Bluetooth JY-MCU, <http://ecno92.blogspot.gr/2012/11/jy-mcu-linvor-at-commands-change-name.html>
- Amarino, <http://www.amarino-toolkit.net/>
- Sensor Graph, <http://www.instructables.com/id/Sensor-Graph-experiment-using-Amarino-shield-and-A/>
- Multi Color Lamp, <http://www.buildcircuit.com/amarino-based-sensor-graph-with-custom-bluetooth-id-and-led-controller/>