

**ΤΕΧΝΟΛΟΓΙΚΟ ΕΚΠΑΙΔΕΥΤΙΚΟ ΙΔΡΥΜΑ ΔΥΤΙΚΗΣ ΕΛΛΑΔΑΣ  
ΣΧΟΛΗ ΤΕΧΝΟΛΟΓΙΚΩΝ ΕΦΑΡΜΟΓΩΝ  
ΤΜΗΜΑ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ Τ.Ε.**

**ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ**

**Η ΑΝΑΠΤΥΞΙΑΚΗ ΠΛΑΚΕΤΑ ARDUINO:  
ΠΙΘΑΝΕΣ ΧΡΗΣΕΙΣ ΤΗΣ ΑΠΟ  
ΗΛΕΚΤΡΟΛΟΓΟΥΣ ΜΗΧΑΝΙΚΟΥΣ**

**ΣΠΟΥΔΑΣΤΗΣ: ΚΑΡΑΝΙΚΟΛΑΣ ΠΡΟΚΟΠΙΟΣ**

**ΕΠΟΠΤΕΥΩΝ: ΦΩΤΟΠΟΥΛΟΣ ΒΑΣΙΛΕΙΟΣ**

**ΠΑΤΡΑ 2013**

# ΠΡΟΛΟΓΟΣ

Το παρόν τεύχος αποτελεί την πτυχιακή εργασία που εκπονήθηκε στο Τμήμα Ηλεκτρολογίας του Τεχνολογικού Εκπαιδευτικού Ιδρύματος Πατρών και αναφέρεται στην αναπτυξιακή πλακέτα ARDUINO, που βασίζεται στον μικροελεγκτή AtMega. Μας δίνεται η δυνατότητα να υλοποιήσουμε ένα μεγάλο εύρος εφαρμογών που αφορά τον τομέα της ηλεκτρολογίας, που στην σημερινή εποχή η εξέλιξη της τεχνολογίας αναπτύσσεται με γοργούς ρυθμούς.

Στην αρχή γίνεται μια θεωρητική εισαγωγή πάνω στους αισθητήρες που μπορούν να βασιστούν πάνω στο Arduino. Στην συνέχεια γίνεται περιγραφή και ανάλυση της πλακέτας Arduino, των επιμέρων στοιχείων της και τέλος δίνονται παραδείγματα με εφαρμογές που θα μπορούσαν να υλοποιηθούν από ηλεκτρολόγους μηχανικούς.

Ευχαριστώ θερμά τον εποπτεύων και καθηγητή κ. Βασίλειο Φωτόπουλο . Καθηγητής του Τμήματος Ηλεκτρολογίας για την πολύτιμη βοήθεια και καθοδήγηση που με προσέφερε για την πραγματοποίηση της εργασίας.

# Η ΑΝΑΠΤΥΞΙΑΚΗ ΠΛΑΚΕΤΑ ARDUINO: ΠΙΘΑΝΕΣ ΧΡΗΣΕΙΣ ΤΗΣ ΑΠΟ ΗΛΕΚΤΡΟΛΟΓΟΥΣ ΜΗΧΑΝΙΚΟΥΣ

Καρανικόλας Προκόπιος

Επιβλέπων  
Φωτόπουλος Βασίλειος

Μέλος 1  
Βλαχόπουλος Πέτρος

Μέλος 2  
Γιαννακόπουλος Κωνσταντίνος

## ΠΕΡΙΛΗΨΗ

Η αναπτυξιακή πλακέτα arduino κυκλοφόρησε πριν από 5 περίπου χρόνια, μαζί με άλλες αντίστοιχες υλοποιήσεις (wiring, beagleboard, raspberry pi κ.α.). Με πολύ μικρό κόστος, μπορεί κανείς να πειραματιστεί μαζί της και να υλοποιήσει ένα μεγάλο εύρος διαφορετικών εφαρμογών, βασισμένος στον μικροελεγκτή AtMega που περιέχει η πλακέτα και ένα απλό προγραμματιστικό περιβάλλον με μια γλώσσα που μοιάζει αρκετά στη C. Στην διπλωματική αυτή περιγράφεται η πλακέτα Arduino στις διάφορες παραλλαγές της, και υλοποιούνται μερικές χρήσιμες απλές εφαρμογές για ηλεκτρολόγους όπως ο έλεγχος των τιμών κάποιων αισθητήρων, χειρισμός ρελέ, σύστημα ανίχνευσης φωτεινής πηγής κ.α.

## **ABSTRACT**

In the current thesis a theoretical study of Arduino platform is developed focusing on the properties of the board and its capabilities. Arduino programming language is presented as well as certain techniques for application development. The applications developed in the framework of the thesis cover a wide range of fields.

# ΠΙΝΑΚΑΣ ΠΕΡΙΕΧΟΜΕΝΩΝ

ΕΙΣΑΓΩΓΗ .....	09
ΚΕΦΑΛΑΙΟ 1 – ΘΕΩΡΗΤΙΚΟ ΥΠΟΒΑΘΡΟ ΔΙΚΤΥΩΝ ΑΙΣΘΗΤΗΡΩΝ – ΠΛΑΤΦΟΡΜΕΣ .....	10
1.1 Αίσθηση και Αισθητήρες .....	10
1.2 Προδιαγραφές Αισθητήρων .....	11
1.3 Ασύρματα Δίκτυα Αισθητήρων .....	12
1.4 Πλατφόρμες Χαμηλού Επιπέδου .....	14
1.5 Ασύρματα modules Υψηλού Επιπέδου .....	15
1.6 Τυποποίηση .....	16
1.7 ZigBee .....	16
1.8 Λειτουργικό Σύστημα .....	18
1.9 Αρχιτεκτονική Ασύρματων Δικτύων Αισθητήρων και Στοιβα Πρωτοκόλλου.....	19
1.10 Εφαρμογές Ασύρματων Δικτύων Αισθητήρων .....	21
1.11 Εφαρμογές για έξυπνο σπίτι .....	21
1.12 Παρακολούθηση νερού.....	21
1.13 Περιβαλλοντολογικές Εφαρμογές.....	23
1.14 Εφαρμογές σε Υπηρεσίες Υγείας.....	24
ΚΕΦΑΛΑΙΟ 2 -Η ΠΛΑΤΦΟΡΜΑ ΑΝΑΠΤΥΞΗΣ ΕΦΑΡΜΟΓΩΝ ΔΙΚΤΥΩΝ ΑΙΣΘΗΤΗΡΩΝ ARDUINO .....	27
2.1 Arduino Uno.....	27
2.2 Τροφοδοσία.....	29
2.3 Μνήμη.....	30
2.4 Είσοδοι και Έξοδοι.....	30
2.5 Επικοινωνία .....	31
2.6 Προγραμματισμός .....	31
2.7 Arduino Ethernet Shield .....	32
2.8 Relay Shield.....	33
2.9 Διάγραμμα Μπλοκ.....	35
2.10 Πλακέτα XBEE .....	37
2.11 Module XBee .....	37
2.12 Τεχνικά Χαρακτηριστικά.....	38
ΚΕΦΑΛΑΙΟ 3- ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΣ ΠΛΑΤΦΟΡΜΑΣ ARDUINO .....	41

3.1 Γλώσσα Προγραμματισμού Arduino.....	41
3.2 Δομή.....	41
3.3 Συνάρτηση setup().....	41
3.4 Συνάρτηση loop().....	42
3.5 Σχεδιάγραμμα (Sketch).....	42
3.6 Σχόλια (Comments).....	43
3.7 Μεταβλητές (Variables).....	43
3.8 Εύρος Μεταβλητών.....	43
3.9 Είδη Μεταβλητών.....	44
3.10 Τελεστές.....	45
3.11 Συγκριτικοί τελεστές.....	46
3.12 Σταθερές.....	46
3.13 Συναρτήσεις (Functions).....	47
3.14 Κοινές συναρτήσεις (Common Functions).....	47
3.15 Παραδείγματα.....	48
3.16 Συνάρτηση pinMode(pin,mode).....	49
3.17 Συνάρτηση digitalRead(pin).....	50
3.18 Συνάρτηση digitalWrite(pin,Value).....	50
3.19 Συνάρτηση analogRead(pin).....	50
3.20 Συνάρτηση analogWrite(pin,Value).....	51
3.21 Συνάρτηση delay(ms).....	51
3.22 Συνάρτηση millis().....	51
3.23 Συναρτήσεις min(x,y) max(x,y).....	52
3.24 Συνάρτηση randomSeed(seed).....	52
3.25 Συναρτήσεις random(max), random(min,max).....	52
3.26 Συνάρτηση Serial.begin(rate).....	52
3.27 Συνάρτηση Serial.println(data).....	53
ΚΕΦΑΛΑΙΟ 4- ΕΦΑΡΜΟΓΕΣ ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΥ ΣΕ ΠΛΑΤΦΟΡΜΑ ARDUINO.....	54
4.1 Εφαρμογή 1 – Ενεργοποίηση LED.....	54
4.2 Εφαρμογή 2 – Ψηφιακή Είσοδος σε Arduino.....	55
4.3 Εφαρμογή 3 – Χειρισμός μεγάλων ρευμάτων εξόδου.....	56
4.4 Εφαρμογή 4 – Έξοδος PWM.....	57
4.5 Εφαρμογή 5 – Είσοδος από ποτενσιόμετρο.....	58

4.6 Εφαρμογή 6 – Σύνδεση Φωτοαντίστασης LDR.....	59
4.7 Εφαρμογή 7 – Έλεγχος διάταξης LEDs.....	61
4.8 Εφαρμογή 8 – Βρόχος Έλεγχου .....	64
4.9 Εφαρμογή 9 – Έλεγχος If.....	66
4.10 Εφαρμογή 10 Έλεγχος Σερβοκινητήρα.....	69
Βιβλιογραφία.....	72
Παράρτημα Α.....	73
Παράρτημα Β.....	78
Παράρτημα Γ.....	79
Παράρτημα Δ.....	82

## ΚΑΤΑΛΟΓΟΣ ΣΧΗΜΑΤΩΝ

Σχήμα 1.1: Βήματα διαδικασίας αίσθησης .....	10
Σχήμα 1.2: Παράδειγμα Δικτύου Ασύρματων Αισθητήρων. ....	12
Σχήμα 1.3: Στοιβα πρωτοκόλλων ZigBee και 802.15.4 .....	17
Σχήμα 1.4: Αρχιτεκτονική Δικτύου Ασύρματων Αισθητήρων .....	19
Σχήμα 1.5: Στοιβα Πρωτόκολλου Ασύρματων Δικτύων Αισθητήρων .....	20
Σχήμα 1.6:Δίκτυο Ασύρματων Αισθητήρων για την παρακολούθηση ροής όγκου..... ..... νερού.....	22
Σχήμα 1.7:Αρχιτεκτονική Δικτύου εφαρμογής Code Blue.....	25
Σχήμα 2.1:Διασυνδέσεις μικροελεγκτή ATmega328. ....	31
Σχήμα 2.2:Διάγραμμα μπλοκ του Relay Shield. ....	35
Σχήμα 3.1:Κυκλωματικό διάγραμμα ενεργοποίησης LED .....	48
Σχήμα 4.1:Κυκλωματικό διάγραμμα εφαρμογής 1.....	54
Σχήμα 4.2:Κυκλωματικό διάγραμμα εφαρμογής 2.....	55
Σχήμα 4.3:Κυκλωματικό διάγραμμα εφαρμογής 3.....	56
Σχήμα 4.4:Κυκλωματικό διάγραμμα εφαρμογής 4.....	57
Σχήμα 4.5:Κυκλωματικό διάγραμμα εφαρμογής 5.....	58
Σχήμα 4.6:Κυκλωματικό διάγραμμα εφαρμογής 6.....	60
Σχήμα 4.7:Κυκλωματικό διάγραμμα εφαρμογής 7.....	62
Σχήμα 4.8:Κυκλωματικό διάγραμμα εφαρμογής 8.....	65
Σχήμα 4.9:Κυκλωματικό διάγραμμα εφαρμογής 9.....	67
Σχήμα 4.10:Κυκλωματικό διάγραμμα εφαρμογής 10 .....	71

## ΚΑΤΑΛΟΓΟΣ ΠΙΝΑΚΩΝ

Πίνακας 1: Τεχνολογίες ασύρματων κόμβων αισθητήρων με έμφαση στους μικρο- ελεγκτές, πομποδέκτες, την χωρητικότητα μνήμης δεδομένων και την προγραμματιστι- κή γλώσσα.....	13
---	----

Πίνακας 2: Χαρακτηριστικά Arduino uno .....	29
Πίνακας 3: Χαρακτηριστικά λειτουργίας ανά καναλι.....	35
Πίνακας 4: Βασικά τεχνικά χαρακτηριστικά λειτουργίας του module XBee.....	38
Πίνακας 5: Σήματα ακροδεκτών του module XBee.....	40

## ΚΑΤΑΛΟΓΟΣ ΚΩΔΙΚΩΝ

Κώδικας 1 Ενεργοποίηση LED .....	54
Κώδικας εφαρμογής 2 .....	55
Κώδικας εφαρμογής 3 .....	56
Κώδικας εφαρμογής 4 .....	57
Κώδικας εφαρμογής 5 .....	58
Κώδικας εφαρμογής 6 .....	60
Κώδικας εφαρμογής 7 .....	63
Κώδικας εφαρμογής 8 .....	65-66
Κώδικας εφαρμογής 9 .....	68
Κώδικας εφαρμογής 10 .....	70

## ΚΑΤΑΛΟΓΟΣ ΕΙΚΟΝΩΝ

Εικόνα 1.1: Χρονολογική εξέλιξη των ασύρματων κόμβων αισθητήρων .....	14
Εικόνα 1.2:Υλοποιήσεις της εφαρμογής Code Blue με Telos mote και MicaZ mote....	24
Εικόνα 2.1:Τα κύρια μέρη του arduino uno .....	27
Εικόνα 2.2:Όψεις της πλακέτας arduino uno .....	28
Εικόνα 2.3:Όψεις της πλακέτας Ethernet.....	32
Εικόνα 2.4:Πλακέτα Relay.....	33
Εικόνα 2.5:Πλακέτα Relay ενσωματωμένη σε πλακέτα arduino.....	34
Εικόνα 2.6:Πλακέτα Relay με ενσωματωμένο 315 MHz RF για ασύρματη..... επικοινωνία .....	34
Εικόνα 2.7:Ένα RFBee ελέγχει δύο πλακέτες .....	34
Εικόνα 2.8:XBee Shield στα δεξιά και XBee module στα αριστερά.....	37
Εικόνα 2.9:Λογική λειτουργίας της μονάδας XBee module.....	39
Εικόνα 2.10:Διασυνδέσεις των ακροδεκτών του module XBee.....	39
Εικόνα 3.1:Προγραμματιστικό περιβάλλον του Arduino.....	42
Εικόνα 3.2:Ένα Arduino uno συνδεδεμένο με ένα led .....	48
Εικόνα 4.1:Σχηματικό διάγραμμα εφαρμογής 6.....	59
Εικόνα 4.2:Σχηματικό διάγραμμα εφαρμογής 7.....	61
Εικόνα 4.3:Σχηματικό διάγραμμα εφαρμογής 8.....	64
Εικόνα 4.4:Σχηματικό διάγραμμα εφαρμογής 9 .....	67
Εικόνα 4.5:Σχηματικό διάγραμμα εφαρμογής 10.....	69



## ΕΙΣΑΓΩΓΗ

Την σημερινή εποχή οι γοργοί ρυθμοί εξέλιξης της τεχνολογίας έχουν μεγάλο αντίκτυπο στην καθημερινή ζωή των ανθρώπων σε επίπεδο δυνατοτήτων και ευκολιών. Η ευχέρεια στην πρόσβαση πληροφοριών παντός τύπου και μορφής είναι κάτι που χαρακτηρίζει τις σημερινές μέρες λόγω της ευρείας διάδοσης του διαδικτύου. Ως αποτέλεσμα των ανωτέρω είναι και η συνεχής ανάπτυξη νέων τάσεων και εφαρμογών.

Συγκεκριμένα τα τελευταία χρόνια παρατηρείται η ανάπτυξη ενός κλάδου που αναφέρεται βιβλιογραφικά με τον όρο διαδίκτυο των πραγμάτων (Internet of Things). Αυτός ο κλάδος περιγράφει την διασύνδεση υλικών οντοτήτων και αντικειμένων σε μια μορφή διαδικτύου έτσι ώστε να παρέχουν στον χρήστη πληροφορίες είτε για την κατάσταση της λειτουργία τους, είτε για διάφορα γεγονότα που έχουν αξία, είτε για να παρέχεται η δυνατότητα ενεργειών πάνω σε αυτά ανεξάρτητα από την φυσική τοποθεσία του χρήστη. Σημαντικό κομμάτι αυτού του κλάδου βασίζεται στις τεχνολογίες δικτύων ασύρματων κόμβων αισθητήρων. Τα δίκτυα αυτά αποτελούνται από μικρούς σε διαστάσεις κόμβους, σχεδιασμένους με έμφαση στην χαμηλή κατανάλωση για μεγάλη αυτονομία λειτουργίας, οι οποίοι παρέχουν πληροφορίες σχετικά με διάφορα φυσικά φαινόμενα τα οποία δειγματοληπτούν μέσω μιας μεγάλης ποικιλίας αισθητήρων που έχουν αναπτυχθεί για το σκοπό αυτό.

Στην ανάλυση που ακολουθεί γίνεται η περιγραφή της υλοποίησης διαφόρων εφαρμογών που βασίζονται σε μια τεχνολογία υλοποίησης του Internet of Things και είναι η πλατφόρμα ARDUINO. Στα παρακάτω κεφάλαια περιλαμβάνονται:

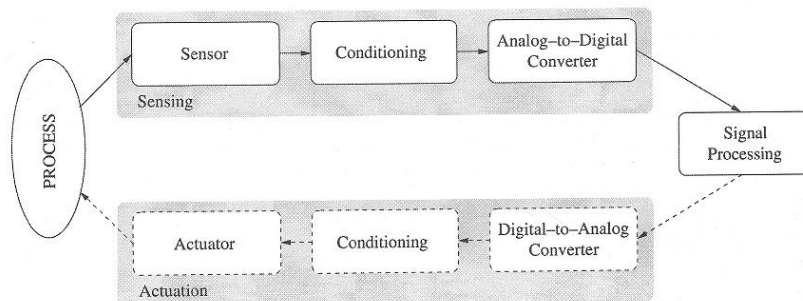
- Η ανάπτυξη του θεωρητικού υποβάθρου που σχετίζεται με δίκτυα αισθητήρων που βασίζονται στην τεχνολογική πλατφόρμα Arduino. Συγκεκριμένα καλύπτονται οι έννοιες της αίσθησης, της δειγματοληψίας, των ασύρματων δικτύων και της διαχείρισης δεδομένων.
- Η περιγραφή και ανάλυση των χαρακτηριστικών των υλικών που απαιτούνται για υλοποίηση εφαρμογών στην πλατφόρμα Arduino.
- Παρουσιάζονται παραδείγματα τέτοιων εφαρμογών στην πλατφόρμα Arduino με έμφαση σε εφαρμογές του κλάδου των ηλεκτρολόγων μηχανικών.

# ΚΕΦΑΛΑΙΟ 1– ΘΕΩΡΗΤΙΚΟ ΥΠΟΒΑΘΡΟ ΔΙΚΤΥΩΝ ΑΙΣΘΗΤΗΡΩΝ – ΠΛΑΤΦΟΡΜΕΣ

## 1.1 Αίσθηση και Αισθητήρες

Αίσθηση είναι η τεχνική που χρησιμοποιείται για να την συλλογή πληροφοριών που σχετίζονται με φυσικά αντικείμενα και φαινόμενα, με διαδικασίες και με την εμφάνιση-ύπαρξη γεγονότων. Ένα παράδειγμα είναι η συλλογή πληροφοριών για την αλλαγή στην θερμοκρασία ή την πίεση.

Το αντικείμενο που εκτελεί τέτοιου είδους διεργασίες ονομάζεται αισθητήρας. Από τεχνικής άποψης, ο αισθητήρας είναι μια συσκευή που μεταφράζει παραμέτρους ή γεγονότα από τον φυσικό κόσμο σε σήματα τα οποία μπορούν να μετρηθούν και να αναλυθούν. Ένας άλλος συχνά εμφανιζόμενος όρος είναι ο μετατροπέας ή μορφοτροπέας (transducer), που χρησιμοποιείται για να περιγράψει μια συσκευή η οποία μετατρέπει ενέργεια από μια μορφή σε μία άλλη. Ο αισθητήρας λοιπόν είναι ένας μετατροπέας που μετατρέπει ενέργεια που σχετίζεται με μια φυσική παράμετρο σε ηλεκτρική ενέργεια και μπορεί να εισαχθεί σε ένα υπολογιστικό σύστημα ή ελεγκτή. Ένα παράδειγμα για τα βήματα που ακολουθούνται στην διαδικασία της αίσθησης βρίσκεται στο σχήμα 1.1



**Σχήμα 1.1: Βήματα διαδικασίας αίσθησης**

Φαινόμενα που συμβαίνουν στον φυσικό κόσμο παρακολουθούνται από μια συσκευή αισθητήρα. Συνήθως, τα ηλεκτρικά σήματα που προκύπτουν δεν είναι έτοιμα για άμεση επεξεργασία από τις επόμενες βαθμίδες ενός ηλεκτρονικού συστήματος και απαιτείται ένα στάδιο προετοιμασίας. Σε αυτό το στάδιο συνήθως εφαρμόζεται μια βαθμίδα ενίσχυσης του λαμβανόμενου αναλογικού σήματος, έπειτα στο σήμα εφαρμόζονται κατάλληλες βαθμίδες φίλτρων για την ανάδειξη συγκεκριμένων χαρακτηριστικών, υλοποιείται κβάντιση με βάση τον ρυθμό της δειγματοληψίας και στο τελικό στάδιο υλοποιείται η απομόνωση των ωφέλιμων πληροφοριών καθώς και η βαθμονόμηση του ψηφιακού σήματος. Συγκεκριμένα, τα σήματα χρειάζονται ενίσχυση για να αλλάξει το εύρος πλάτους τους και να ταιριάζει με το εύρος σήματος του επόμενου σταδίου που είναι η μετατροπή της μορφής τους από αναλογική σε ψηφιακή. Επιπλέον τα εφαρμοζόμενα φίλτρα υλοποιούν την αφαίρεση του ανεπιθύμητου θορύβου σε συγκεκριμένο εύρος συχνοτήτων. Η κβάντιση με τον κατάλληλο ρυθμό δειγματοληψίας συντελεί στην υλοποίηση του μετατροπέα από αναλογική σε ψηφιακή μορφή

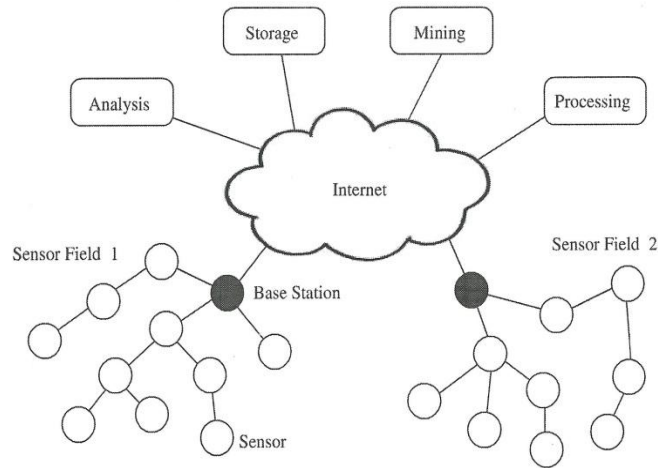
(ADC). Τέλος τα στάδια απομόνωσης και βαθμονόμησης ολοκληρώνουν την προετοιμασία του σήματος το οποίο βρίσκεται πλέον σε ψηφιακή μορφή και μπορεί να εφαρμοστεί σε αυτό επιπλέον επεξεργασία, αποθήκευση ή οπτικοποίηση.

Συνήθως τα ασύρματα δίκτυα αισθητήρων περιλαμβάνουν ενεργοποιητές που τους επιτρέπουν να ελέγχουν απευθείας παραμέτρους του φυσικού κόσμου. Για παράδειγμα ένας ενεργοποιητής μπορεί να είναι μια βαλβίδα που να ελέγχει την ροή ζεστού νερού, μια μηχανή που να ανοίγει ή να κλείνει μια πόρτα ή ένα παράθυρο, ή μια αντλία που να ελέγχει τα ποσοστά του καυσίμου σε μια μηχανή. Ένα τέτοιο δίκτυο ασύρματων αισθητήρων και ενεργοποιητών (WSAN) λαμβάνει εντολές από μια μονάδα ελέγχου και μετατρέπει τις εντολές αυτές σε σήματα εισόδου για τους ενεργοποιητές, οι οποίοι επενεργούν με μια φυσική διεργασία όπως φαίνεται στο σχήμα 1.1

## 1.2 Προδιαγραφές Αισθητήρων

Η επιλογή του είδους των αισθητήρων καθορίζεται από τις φυσικές παραμέτρους προς μέτρηση/παρακολούθηση, για παράδειγμα τέτοιες παράμετροι είναι η θερμοκρασία, η πίεση, το φως, η υγρασία. Εκτός από τις φυσικές παραμέτρους οι προδιαγραφές των αισθητήρων μπορούν να βασιστούν και σε άλλες μεθόδους, για παράδειγμα αν απαιτείται εξωτερική τροφοδοσία. Αν ένας αισθητήρας χρειάζεται εξωτερική τροφοδοσία ονομάζεται ενεργός αισθητήρας. Αυτό συμβαίνει όταν πρέπει να εκπέμψει κάποιου είδους ενέργεια (π.χ. μικροκύματα, φως, ήχο) για να ενεργοποιήσει μια αντίδραση ή να ανιχνεύσει μια αλλαγή στην ενέργεια το εκπεμπόμενου σήματος. Από την άλλη πλευρά υπάρχουν οι παθητικοί αισθητήρες που ανιχνεύουν την ενέργεια στο περιβάλλον και αντλούν την ισχύ τους από αυτή την ενέργεια εισόδου.

### 1.3 Ασύρματα Δίκτυα Αισθητήρων



**Σχήμα 1.2:** Παράδειγμα δικτύου ασύρματων αισθητήρων.

Με τον όρο ασύρματα δίκτυα αισθητήρων νοείται μια χωρική αρχιτεκτονική ασύρματων και αυτόνομων κόμβων που επικοινωνούν μεταξύ τους αλλά και με σταθμούς βάσης έτσι ώστε να ανταλλάσσουν πληροφορίες και μετρήσεις. Η ασύρματη επικοινωνία μεταξύ των κόμβων και των σταθμών βάσης διευκολύνει σε εφαρμογές που απαιτούν μεγάλο πλήθος κόμβων αλλά επιπλέον προσδίδει και βαθμούς ελευθερίας στην διάταξη του δικτύου λόγω της φορητότητας των κόμβων και της ευελιξίας των ασύρματων προτύπων επικοινωνιών.

Ένας κόμβος αισθητήρα συνήθως αποτελείται από τρία στοιχεία τα οποία μπορεί να είναι είτε σε μεμονωμένες πλακέτες είτε να είναι ενσωματωμένα σε ένα ενιαίο σύστημα. Τα στοιχεία αυτά είναι:

- Ασύρματα modules (wireless modules or motes): είναι τα βασικά στοιχεία του δικτύου αισθητήρων καθώς κατέχουν τις δυνατότητες επικοινωνίας και την προγραμματιζόμενη μνήμη στην οποία βρίσκεται ο κώδικας της εφαρμογής. Ένα τέτοιο module συνήθως αποτελείται από μικροελεγκτή, πομποδέκτη, πηγή ισχύος, μονάδα μνήμης, και μπορεί να περιέχει μερικούς αισθητήρες.
- Πλακέτα αισθητήρων (sensor board): είναι η πλακέτα που συνδέεται στο ασύρματο module η οποία έχει ενσωματωμένους αισθητήρες πολλών τύπων. Η πλακέτα αισθητήρων μπορεί να περιέχει και μια περιοχή προτυποποίησης για να συνδέονται και επιπλέον είδη αισθητήρων. Εναλλακτικά οι αισθητήρες μπορεί να είναι ενσωματωμένοι στο ασύρματο module.
- Πλακέτα προγραμματισμού (programming board) ή πλακέτα πύλης σύζευξης (gateway board): είναι η πλακέτα που παρέχει πολλαπλές διεπαφές συμπεριλαμβανομένων των Ethernet, Wi-Fi, USB, ή σειριακών θυρών για την σύνδεση διαφορετικών modules στο δίκτυο μιας επιχείρησης ή σε ένα βιομηχανικό

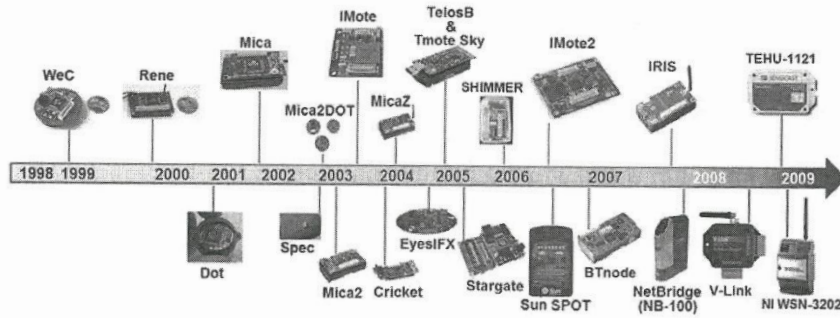
## ΚΕΦΑΛΑΙΟ 1 – ΘΕΩΡΗΤΙΚΟ ΥΠΟΒΑΘΡΟ ΔΙΚΤΥΩΝ ΑΙΣΘΗΤΗΡΩΝ

δίκτυο ή τοπικά σε έναν υπολογιστή. Αυτές οι πλακέτες χρησιμοποιούνται είτε για τον προγραμματισμό του module είτε για την συλλογή δεδομένων από αυτό. Συγκεκριμένα, ειδικές διατάξεις πρέπει να είναι συνδεδεμένες στην πλακέτα προγραμματισμού για να υπάρχει δυνατότητα φόρτωσης του κώδικα της εφαρμογής στην προγραμματιζόμενη μνήμη του ασύρματου module.

Παρακάτω ακολουθεί πίνακας με τα πιο δημοφιλή ασύρματα modules που έχουν κατασκευαστεί τα τελευταία χρόνια δίνοντας έμφαση στην ταχύτητα της επεξεργαστικής τους μονάδας, το μέγεθος της προγραμματιζόμενης και αποθηκευτικής τους μνήμης, της συχνότητας λειτουργίας και τους ρυθμούς μετάδοσής τους.

**Πίνακας 1: Τεχνολογίες ασύρματων κόμβων αισθητήρων με έμφαση στους μικροελεγκτές, τους πομποδέκτες, την χωρητικότητα μνήμης δεδομένων και την προγραμματιστική γλώσσα.**

Sensor Node Name	Microcontroller	Tranceiver	Program+Data Memory	External Memory	Programming
<b>Eyes</b>	MSP430F149	TR1001		8 Mbit	PeerOS Support
<b>EyesFX v1</b>	MSP430F149	TDA5250 (868 MHz) FSK		8 Mbit	TinyOS Support
<b>EyesFX v2</b>	MSP430F1611	TDA5250 (868 MHz) FSK		8 Mbit	TinyOS Support
<b>IMote</b>	ARM core 12 MHz	Bluetooth with the range of 30 m	64K SRAM	512K Flash	TinyOS Support
<b>IMote 1.0</b>	ARM 7TDMI 12-48 MHz	Bluetooth with the range of 30 m	64K SRAM	512K Flash	TinyOS Support
<b>IMote 2.0</b>	Marvell PXA271 ARM 11-400 MHz	TI CC2420 802.15.4/ZigBee compliant radio	32 MB SRAM	32 MB Flash	Microsoft .NET Micro, Linux, TinyOS Support
<b>Mica</b>	ATmega 103 4 MHz 8-bit CPU	RFM TR1000 radio 50 kbit/s	128+4K RAM	512K Flash	nesC Programming
<b>Mica2</b>	ATMEGA 128L	Chipcon 868/916 MHz	4K RAM	128K Flash	TinyOS, SOS and MantisOS Support
<b>Mica2Dot</b>	ATMEGA 128		4K RAM	128K Flash	
<b>MicaZ</b>	ATMEGA 128	TI CC2420 802.15.4/ZigBee compliant radio	4K RAM	128K Flash	nesC
<b>Telos</b>	MSP430		2K RAM		
<b>TelosB</b>	Texas Instruments MSP430 microcontroller	250 kbit/s 2.4 GHz IEEE 802.15.4 Chipcon Wireless Transceiver	10k RAM	48k Flash	Contiki, TinyOS, SOS and MantisOS Support
<b>T-Mote Sky</b>	Texas Instruments MSP430 microcontroller	250 kbit/s 2.4 GHz IEEE 802.15.4 Chipcon	10k RAM	48k Flash	Contiki, TinyOS, SOS and MantisOS Support
<b>Wasp mote</b>	Atmel ATmega 1281	ZigBee/802.15.4/DigiMesh/RF, 2.4 GHz/868/900 MHz	8K SRAM	128K FLASH ROM, 4K EEPROM, 2 GB SD card	C/Processing



Εικόνα 1.1: Χρονολογική εξέλιξη ασύρματων κόμβων αισθητήρων

## 1.4 Πλατφόρμες Χαμηλού Επιπέδου

Οι πλατφόρμες χαμηλού επιπέδου χαρακτηρίζονται από τις περιορισμένες τους δυνατότητες σε σχέση με την επεξεργασία, την μνήμη και την επικοινωνία. Σχεδιάστηκαν για να χρησιμοποιηθούν σε μεγάλους αριθμούς σε ασύρματα δίκτυα αισθητήρων για να παρέχουν τις δυνατότητες παρακολούθησης παραμέτρων (sensing) καθώς και την υποδομή για τις επικοινωνίες.

- Οικογένεια Mica: Η οικογένεια αυτή περιλαμβάνει τους κόμβους Mica, Mica2, MicaZ και IRIS. Κάθε κόμβος αποτελείται από 8-bit Atmel AVR μικροελεγκτές με ταχύτητα από 4-16MHz και προγραμματιζόμενη μνήμη flash από 128-256MB. Οι κόμβοι της οικογένειας Mica έχουν μεγάλο εύρος από πομποδέκτες. Ο κόμβος Mica έχει έναν πομποδέκτη 916MHz ή 433MHz στα 40kbps, ενώ ο κόμβος Mica2 έχει έναν πομποδέκτη 433/868/916MHz στα 40kbps. Οι κόμβοι MicaZ και IRIS έχουν εξοπλιστεί με πομποδέκτες συμβατούς με το πρότυπο IEEE 802.15.4 οι οποίοι λειτουργούν στα 2.4GHz με 250kbps ρυθμό μετάδοσης δεδομένων. Κάθε ασύρματο module έχει περιορισμένη μνήμη τύπου τυχαίας προσπέλασης (RAM) χωρητικότητας από 4-8kB και μνήμη δεδομένων 512kB. Επίσης κάθε έκδοση είναι εξοπλισμένη με σύνδεσμο 51 ακροδεκτών που χρησιμοποιείται για να συνδέει επιπλέον πλακέτα αισθητήρων και πλακέτα προγραμματισμού στο ασύρματο module.
- Οικογένεια Telos/Tmote: Η οικογένεια αυτή έχει παρεμφερή αρχιτεκτονική με τον κόμβο MicaZ της οικογένειας Mica. Ο πομποδέκτης είναι ακριβώς ο ίδιος με αυτόν του κόμβου MicaZ αλλά οι κόμβοι Telos/Tmote έχουν μεγαλύτερη μνήμη τυχαίας προσπέλασης (RAM) επειδή χρησιμοποιείται ένας 8MHz TI MSP430 μικροελεγκτής με 10kB RAM. Επιπλέον οι κόμβοι Telos/Tmote έχουν ενσωματωμένους αρκετούς αισθητήρες όπως αισθητήρα φωτός, αισθητήρα υπερύθρων, αισθητήρα υγρασίας και θερμοκρασίας καθώς επίσης διαθέτουν και θύρα USB που καταργεί την ανάγκη για επιπλέον πλακέτα αισθητήρων και πλακέτα προγραμματισμού. Τέλος διαθέτουν και συνδέσμους 6 και 10 ακροδεκτών για επιπλέον αισθητήρες [1].

- **EYE:** Η πλατφόρμα EYE σχεδιάστηκε σαν απόρροια μιας 3ετούς Ευρωπαϊκής έρευνας και είναι παρεμφερής με την αρχιτεκτονική των κόμβων της οικογένειας Telos/Tmote. Στους κόμβους EYE χρησιμοποιείται ένας 16-bit μικροελεγκτής με 60kB προγραμματιζόμενη μνήμη και 2kB μνήμη δεδομένων. Επιπλέον στους κόμβους EYE είναι ενσωματωμένοι οι ακόλουθοι αισθητήρες: πυξίδα, αισθητήρας επιτάχυνσης, αισθητήρας θερμοκρασίας, αισθητήρας φωτός και αισθητήρας πίεσης. Οι κόμβοι EYE διαθέτουν τον πομποδέκτη TR1001 που υποστηρίζει ρυθμούς μετάδοσης μέχρι 115,2kbps με ενεργειακή κατανάλωση της τάξης των 14.4mW, 16.0mW και 15.0mW κατά την διάρκεια λήψης, εκπομπής και αδράνειας αντίστοιχα. Οι κόμβοι τέλος διαθέτουν και μια RS232 σειριακή διεπαφή για προγραμματισμό [2].

## 1.5 Ασύρματα modules Υψηλού Επιπέδου

Εκτός των αισθητήρων, της τοπικής επεξεργασίας και της επικοινωνίας πολλαπλών βημάτων (multi-hop) τα ασύρματα δίκτυα αισθητήρων χρειάζονται επιπλέον λειτουργικότητες οι οποίες δεν μπορούν να μεταφερθούν αποδοτικά από τα ασύρματα module χαμηλού επιπέδου. Λειτουργίες υψηλού επιπέδου όπως η διαχείριση του δικτύου χρειάζονται μεγαλύτερη επεξεργαστική ισχύ και μνήμη σε σχέση με τις δυνατότητες που μπορούν να προσφέρουν τα ασύρματα modules χαμηλού επιπέδου. Επιπλέον η ολοκλήρωση των ασύρματων δικτύων αισθητήρων με τις ήδη υπάρχουσες υποδομές δικτύωσης χρειάζεται πολλαπλές τεχνικές επικοινωνίας να είναι ενσωματωμένες διαμέσου ενός module-δρομολογητή (gateway module). Επιπροσθέτως, στα δίκτυα όπου πλήμνες (hubs) επεξεργασίας ή αποθήκευσης είναι ενσωματωμένες με κόμβους αισθητήρων χρειάζονται κόμβοι με μεγαλύτερη χωρητικότητα. Για τους παραπάνω λόγους σχεδιάστηκαν τα ασύρματα modules υψηλού επιπέδου.

- **Stargate:** Η πλακέτα Stargate είναι μια πλατφόρμα υψηλών επιδόσεων που σχεδιάστηκε για να υλοποιεί αίσθηση (sensing), επεξεργασία σήματος, έλεγχο και διαχείριση δικτύωσης αισθητήρων. Η πλατφόρμα Stargate είναι βασισμένη στον Intel PXA-255 Xscale 400MHz RISC επεξεργαστή. Η Stargate διαθέτει flash μνήμη 32MB, 64MB SDRAM μνήμη τυχαίας προσπέλασης και ενσωματωμένους συνδέσμους για τα ασύρματα modules της οικογένειας Mica καθώς επίσης και κάρτες διασύνδεσης PCMCIA Bluetooth επικοινωνίας ή ασύρματης επικοινωνίας συμβατής με τα πρότυπα IEEE 802.11. Ως εκ τούτου μπορεί να λειτουργήσει σαν ασύρματος δρομολογητής και υπολογιστική πλήμνη (hub) για αλγορίθμους επεξεργασίας μέσα στο δίκτυο. Μπορεί να συνδεθεί με κάμερα ή άλλες οπτικοακουστικές συσκευές και να λειτουργήσει σαν αισθητήρας πολυμέσων μεσαίας ανάλυσης αν και έχει μεγάλη ενεργειακή κατανάλωση.
- **Imote:** Η Intel ανέπτυξε δύο πρότυπες γενιές ασύρματων αισθητήρων για υψηλών επιδόσεων εφαρμογές. Η πλατφόρμα Imote κατασκευάστηκε με βάση έναν ολοκληρωμένο ασύρματο μικροελεγκτή που περιελάμβανε έναν 8-bit 12MHz ARM7 επεξεργαστή, για ασύρματη επικοινωνία μια πλακέτα Bluetooth, 64kB RAM μνήμη τυχαίας προσπέλασης και 32kB μνήμη flash καθώς επίσης και πολλές επιλογές εισόδων/εξόδων. Η αρχιτεκτονική του λειτουργικού ήταν βασισμένη σε μια έκδοση του λειτουργικού συστήματος TinyOS για τον επεξεργαστή ARM.

- Η δεύτερη γενιά του Imote, το **Imote2** κατασκευάστηκε με βάση ένα νέο χαμηλής κατανάλωσης επεξεργαστή τον 32-bit PXA271 XScale χρονισμένο στα 320/416/520 MHz ο οποίος χρησιμοποιούσε λειτουργίες ψηφιακής επεξεργασίας σήματος (DSP) για αποθήκευση ή συμπίεση, και για ασύρματη επικοινωνία χρησιμοποιήθηκε μια πλακέτα ChipCon CC2420 συμβατή με το πρότυπο IEEE 802.15.4. Είχε μεγάλη ενσωματωμένη μνήμη τυχαίας προσπέλασης (RAM) και μνήμη flash, υποστήριξη για επιπλέον πομποδέκτες και μια ποικιλία από υψηλής ταχύτητας συνδέσμους εισόδου/εξόδου για να συνδέονται αισθητήρες ή κάμερες. Τέλος μπορούσε να χρησιμοποιεί σαν λειτουργικό σύστημα το Linux και να εκτελεί εφαρμογές υλοποιημένες στην γλώσσα προγραμματισμού Java.

## 1.6 Τυποποίηση

Η μεγάλη ποικιλία στις πλατφόρμες αισθητήρων είχε σαν αποτέλεσμα την δημιουργία προβλημάτων συμβατότητας στις εφαρμογές, ως εκ τούτου άρχισαν να γίνονται προσπάθειες τυποποίησης συγκεκριμένα για τα χαρακτηριστικά της επικοινωνίας. Δημιουργήθηκε το πρότυπο IEEE 802.15.4 που περιγράφει τις προδιαγραφές για τους ασύρματους πομποδέκτες χαμηλού ρυθμού δεδομένων με σκοπό την μεγάλη διάρκεια ζωής της μπαταρίας και την χαμηλή πολυπλοκότητα. Τρία φάσματα συχνοτήτων επιλέχθηκαν για την επικοινωνία, 2,4GHz (γενικό), 915MHz (για την Αμερική), 868MHz (για την Ευρώπη). Το φυσικό επίπεδο χρησιμοποιεί BPSK για τις συχνοτήτες 868/915MHz και OQPSK για την συχνότητα 2,4GHz, το επίπεδο ζεύξης δεδομένων παρέχει επικοινωνία για διάφορες τοπολογίες όπως αστέρα και δέντρου με χρήση ελεγκτών. Η εμβέλεια εκπομπής των κόμβων είναι 10-100 μέτρα με ρυθμούς μετάδοσης από 20-250kbps.

Πάνω από την τυποποίηση IEEE 802.15.4 βρίσκονται διάφορα άλλα “σώματα” τυποποιήσεων που εμπλουτίζουν τις δυνατότητες των δικτύων χαμηλής κατανάλωσης σε διάφορους τομείς. Είναι γενικά αποδεκτό ότι πρότυπα σαν το Bluetooth και το WiFi δεν ταιριάζουν καλά με εφαρμογές αισθητήρων χαμηλής κατανάλωσης για αυτό προτιμούνται πρότυπα όπως τα ZigBee, WirelessHART, WINA, SP100.11a, που ικανοποιούν ακριβώς ότι χρειάζονται τα δίκτυα χαμηλής κατανάλωσης [3].

## 1.7 ZigBee

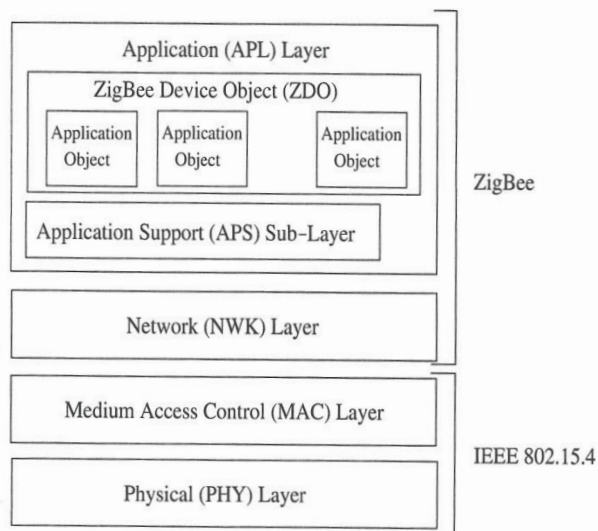
Το πρότυπο ZigBee σχεδιάστηκε για να ικανοποιήσει τις ανάγκες για αποδοτικότητα στις ασύρματες επικοινωνίες που θα υποστηρίζουν χαμηλούς ρυθμούς μετάδοσης δεδομένων, χαμηλή κατανάλωση ενέργειας, ασφάλεια και αξιοπιστία. Δόθηκε έμφαση σε πέντε βασικές περιοχές εφαρμογών: αυτοματισμοί σπιτιών, έξυπνη ενέργεια, αυτοματισμοί κτηρίων, υπηρεσίες τηλεπικοινωνιών, και προσωπική φροντίδα υγείας.

Το πρότυπο ZigBee σχεδιάστηκε συγκεκριμένα για να συνδυάζεται με το πρότυπο IEEE802.15.4. Στο σχήμα 1.3 που ακολουθεί φαίνεται η κατηγοριοποίηση των επιπέδων αναλόγως του προτύπου που ανήκουν. Το φυσικό επίπεδο και το επίπεδο ζεύξης δεδομένων ορίζονται από το πρότυπο IEEE 802.15.4, καθώς το πρότυπο ZigBee ορίζει το επίπεδο δικτύου και το πλαίσιο της εφαρμογής. Τα αντικείμενα της εφαρμογής ορίζονται από τον χρήστη. Για να υποστηρίζεται μεγάλο εύρος εφαρμογών τρεις τύποι κίνησης έχουν οριστεί. Αρχικά είναι η



περιοδική κίνηση δεδομένων για να παρακολουθεί τις εφαρμογές όπου αισθητήρες παρέχουν συνεχόμενες πληροφορίες σχετικά με διάφορα φυσικά φαινόμενα. Η ανταλλαγή δεδομένων ελέγχεται από ελεγκτές δικτύων ή με χρήση δρομολογητών. Έπειτα είναι η διαλειπόμενη κίνηση δεδομένων που εφαρμόζεται στις εφαρμογές που βασίζονται σε ανίχνευση γεγονότων και ενεργοποιείται είτε από την εφαρμογή είτε από κάποιον εξωτερικό παράγοντα. Αυτού του τύπου η κίνηση διεκπεραιώνεται από κάθε κόμβο δρομολόγησης. Για εξοικονόμηση ενέργειας η συσκευή μπορεί να λειτουργεί σε κατάσταση αποσύνδεσης από το υποσύστημα ραδιομετάδοσης για εξοικονόμηση ενέργειας, ενώ λειτουργεί σε κατάσταση ύπνου τις περισσότερες φορές. Οποτε οι πληροφορίες χρειάζεται να αποσταλούν, ο πομποδέκτης ενεργοποιείται και η συσκευή γνωστοποιεί την παρουσία της στο δίκτυο. Τέλος η επαναλαμβανόμενη χαμηλής καθυστέρησης δεδομένων κίνηση ορίζεται για συγκεκριμένες επικοινωνίες όπως ένα κλικ του ποντικιού που χρειάζεται να ολοκληρωθεί σε συγκεκριμένο χρόνο.

Το επίπεδο δικτύου ZigBee (NWK) προσφέρει λειτουργίες διαχείρισης δικτύου για τον χειρισμό του δικτύου. Καθορίζονται οι διαδικασίες για την δημιουργία ενός καινούργιου δικτύου και αν οι συσκευές θα αποκτήσουν δικαιώματα ή θα παραιτηθούν από το δίκτυο. Επιπλέον ανάλογα με τις λειτουργίες του δικτύου, η στοίβα επικοινωνίας της κάθε συσκευής μπορεί να παραμετροποιηθεί. Επειδή μια συσκευή ZigBee μπορεί να είναι μέρος από διαφορετικά δίκτυα κατά τον κύκλο ζωής της, η προτυποποίηση ορίζει έναν ευέλικτο μηχανισμό διευθυνσιοδότησης. Επομένως, ο συντονιστής δικτύου αναθέτει μια διεύθυνση στην συσκευή καθώς αυτή γίνεται μέλος του δικτύου. Σαν αποτέλεσμα, η μοναδική ταυτότητα της κάθε συσκευής δεν χρησιμοποιείται για επικοινωνία αλλά μια συντομότερη διεύθυνση ανατίθεται για να βελτιώσει την αποδοτικότητα κατά την διάρκεια της επικοινωνίας. Σε μια αρχιτεκτονική τύπου δένδρου, η διεύθυνση μιας συσκευής επιπλέον γνωστοποιεί και τον γονέα της, το οποίο είναι χρήσιμο για λόγους δρομολόγησης. Το επίπεδο δικτύου ZigBee προσφέρει επιπλέον συγχρονισμό μεταξύ συσκευών και ελεγκτών δικτύου. Τέλος οι διαδρομές εκπομπής πολλαπλών βημάτων δημιουργούνται από το επίπεδο δικτύου ZigBee σε συμφωνία με τα ορισμένα πρωτόκολλα.



Σχήμα 1.3: Στοίβα πρωτοκόλλων ZigBee και 802.15.4

Όπως παρουσιάζεται στο σχήμα 1.3 το πρότυπο ZigBee ορίζει επίσης συγκεκριμένα συστατικά στο επίπεδο εφαρμογής. Το επίπεδο αυτό αποτελείται από το APS υποεπίπεδο, το αντικείμενο συσκευή ZigBee, και τα αντικείμενα εφαρμογής που ορίζει ο κατασκευαστής. Οι εφαρμογές υλοποιούνται με χρήση των ορισμένων από τον κατασκευαστή αντικειμένων και η υλοποίηση είναι βασισμένη στις απαιτήσεις που ορίζει το πρότυπο. Το αντικείμενο συσκευή ZigBee ορίζει τις λειτουργίες που παρέχονται από την συσκευή για τις λειτουργίες του δικτύου. Πιο συγκεκριμένα, ο ρόλος συσκευών όπως ο συντονιστής δικτύου ή ο δρομολογητής ορίζονται στο αντικείμενο συσκευή ZigBee. Επιπλέον, όποτε μια συσκευή χρειάζεται να συνεργαστεί με το δίκτυο, τα αιτήματα σύνδεσης αντιμετωπίζονται από το αντικείμενο συσκευή ZigBee. Τέλος το υποεπίπεδο APS προσφέρει δυνατότητες ανακάλυψης στις συσκευές έτσι ώστε οι γείτονες μιας συσκευής και οι λειτουργικότητες που αυτοί παρέχουν να μπορούν αποθηκευτούν. Αυτές οι πληροφορίες χρησιμοποιούνται επίσης για να ταιριάζουν τα αιτήματα συνεργασίας των γειτόνων με συγκεκριμένες λειτουργίες [4].

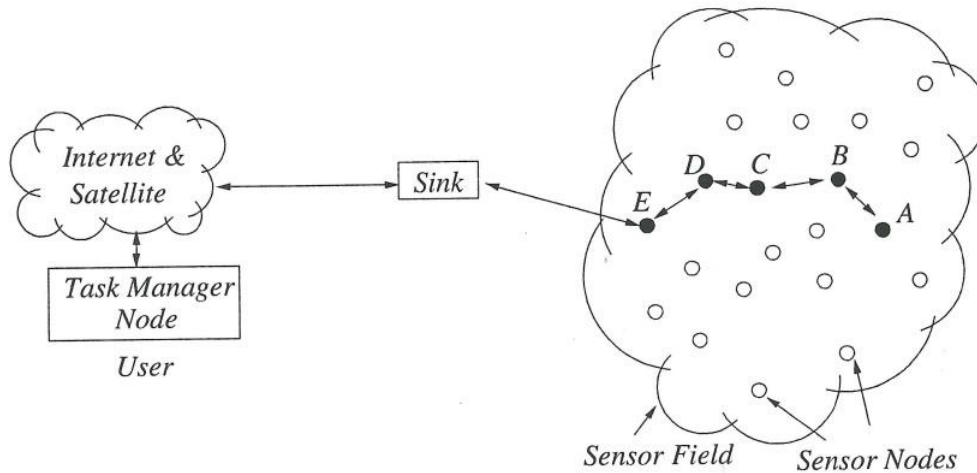
## 1.8 Λειτουργικό Σύστημα

Επιπλέον των πλατφόρμων υλικού και των προτυποποιήσεων, αρκετές πλατφόρμες λειτουργικού έχουν αναπτυχθεί συγκεκριμένα για τα ασύρματα δίκτυα αισθητήρων. Ανάμεσα σε αυτές η πιο διαδεδομένη είναι το TinyOS, το οποίο είναι ένα ανοικτού κώδικα λειτουργικό σύστημα σχεδιασμένο για ασύρματα δίκτυα ενσωματωμένων αισθητήρων.

Το TinyOS ενσωματώνει μια αρχιτεκτονική που στηρίζεται στα “στοιχεία” (components), η οποία ελαχιστοποιεί το μέγεθος του κώδικα και παρέχει μια ευέλικτη πλατφόρμα για την υλοποίηση νέων πρωτοκόλλων επικοινωνιών. Η βιβλιοθήκη του που βασίζεται στα “στοιχεία” περιέχει πρωτόκολλα δικτύων, παρεχόμενες υπηρεσίες, οδηγούς αισθητήρων, και εργαλεία απόκτησης δεδομένων οι οποίες μπορούν να τροποποιηθούν, και να βελτιστοποιηθούν με βάση τις απαιτήσεις των εκάστοτε εφαρμογών. Το TinyOS είναι βασισμένο σε μοντέλο εκτέλεσης οδηγούμενης από γεγονός (event driven) η οποία είναι μια καλή προσέγγιση από άποψη διαχείρισης ενέργειας [5].

Το Contiki είναι ένα λειτουργικό σύστημα ανοικτού κώδικα εκτέλεσης πολλαπλών εργασιών (multitasking) που αναπτύχθηκε για μια μεγάλη ποικιλία πλατφορμών που περιλαμβάνουν μικροελεγκτές όπως ο TI MSP430 και ο Atmel AVR, οι οποίοι χρησιμοποιούνται στις οικογένειες Telos, Tmote, και Mica. Το Contiki είναι κατασκευασμένο με βάση έναν πυρήνα (kernel) οδηγούμενο από γεγονός. Σαν αποτέλεσμα, σε σύγκριση με το TinyOS το οποίο είναι στατικά συνδεδεμένο με τον χρόνο μεταγλώττισης, το Contiki επιτρέπει στα προγράμματα και τους οδηγούς να μπορούν να αντικατασταθούν σε χρόνο εκτέλεσης χωρίς να κάνουν επανασύνδεση. Επιπλέον η υποστήριξη TCP/IP παρέχεται από την στοίβα μIP [6].

## 1.9 Αρχιτεκτονική Ασύρματων Δικτύων Αισθητήρων και Στοιβα Πρωτοκόλλου.

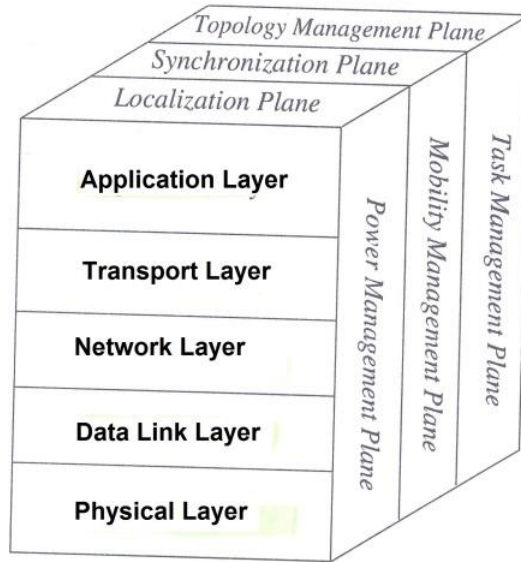


**Σχήμα 1.4:** Αρχιτεκτονική δικτύου ασύρματων αισθητήρων

Οι κόμβοι αισθητήρες είναι συνήθως διασκορπισμένοι σε ένα πεδίο αίσθησης όπως στο σχήμα 1.4. Καθένας από αυτούς τους διασκορπισμένους κόμβους αισθητήρες έχει την δυνατότητα να συλλέγει δεδομένα και να δρομολογεί τα δεδομένα πίσω στην πύλη και τον τελικό χρήστη. Τα δεδομένα δρομολογούνται πίσω στον τελικό χρήστη με εκπομπή πολλαπλών βημάτων μη δομημένης αρχιτεκτονικής διαμέσου της πύλης. Η πύλη μπορεί να επικοινωνεί με τον διαχειριστή διεργασιών/τελικό χρήστη μέσω του Internet ή δορυφόρου ή οποιασδήποτε άλλης μορφής ασύρματο δίκτυο, ή χωρίς κανένα δίκτυο όπου η πύλη μπορεί να είναι συνδεδεμένη απευθείας με τον τελικό χρήστη.

Στα ασύρματα δίκτυα αισθητήρων, οι κόμβοι αισθητήρων έχουν διπλή λειτουργικότητα συγκεκριμένα είναι ικανοί να δημιουργούν δεδομένα και να δρομολογούν δεδομένα. Ως εκ τούτου η επικοινωνία γίνεται για δύο λόγους:

- **Λειτουργία Πηγής:** Κόμβοι Πηγής με πληροφορίες γεγονότος εκτελούν λειτουργίες επικοινωνίας για να μεταδώσουν τα πακέτα τους στην πύλη.
- **Λειτουργία Δρομολόγησης:** Κόμβοι αισθητήρες συμμετέχουν στην προώθηση των πακέτων που λαμβάνουν από άλλους κόμβους στον επόμενο προορισμό.



**Σχήμα 1.5: Στοιβά πρωτοκόλλου ασύρματων δικτύων αισθητήρων**

Η στοιβά πρωτοκόλλου που χρησιμοποιείται από την πύλη και όλους τους κόμβους αισθητήρες φαίνεται στο σχήμα 1.5. Αυτή η στοιβά πρωτοκόλλου συνδυάζει ενημέρωση για θέματα ισχύος και δρομολόγησης, ενσωματώνει δεδομένα με πρωτόκολλα δικτύωσης, αναμεταδίδει πληροφορίες σχετικά με την αποδοτικότητα ισχύος μέσω του ασύρματου μέσου και προωθεί την συνεργατική προσπάθεια ανάμεσα στους ασύρματους κόμβους αισθητήρων. Η στοιβά πρωτοκόλλου αποτελείται από το φυσικό επίπεδο, το επίπεδο ζεύξης δεδομένων, το επίπεδο δικτύου, το επίπεδο μεταφοράς, το επίπεδο εφαρμογής, καθώς επίσης και από ένα επίπεδο συγχρονισμού, επίπεδο εντοπισμού, επίπεδο διαχείρισης τοπολογίας, επίπεδο διαχείρισης ενέργειας, επίπεδο διαχείρισης κινητικότητας, και ένα επίπεδο διαχείρισης διεργασιών.

Το φυσικό επίπεδο διευθετεί τις ανάγκες απλής αλλά ισχυρής διαμόρφωσης, τις τεχνικές εκπομπής και τεχνικές λήψης. Καθώς το περιβάλλον είναι θορυβώδες και οι κόμβοι αισθητήρων είναι φορητοί, το επίπεδο ζεύξης δεδομένων είναι υπεύθυνο στο να διασφαλίσει μια αξιόπιστη επικοινωνία με χρήση τεχνικών ελέγχου σφαλμάτων και να διαχειρίζεται την πρόσβαση στο κανάλι για να ελαχιστοποιεί τις συγκρούσεις λόγω γειτονικών εκπομπών. Το επίπεδο δικτύου φροντίζει την δρομολόγηση των δεδομένων που παρέρχονται από το επίπεδο μεταφοράς. Το επίπεδο μεταφοράς βοηθά στην συντήρηση της ροής δεδομένων αν η εφαρμογή του δικτύου αισθητήρων το χρειάζεται. Επιπρόσθετα τα επίπεδα ισχύος, κινητικότητας, και διαχείρισης διεργασιών παρακολουθούν την ισχύ, την κινητικότητα και την διανομή διεργασιών μεταξύ των κόμβων αισθητήρων. Αυτά τα επίπεδα συντονίζουν τις διεργασίες αίσθησης και χαμηλώνουν την κατανάλωση ισχύος.

Το επίπεδο διαχείρισης ενέργειας διαχειρίζεται με ποιο τρόπο ένας κόμβος αισθητήρα χρησιμοποιεί την ενέργεια του. Για παράδειγμα ο κόμβος αισθητήρα μπορεί να κλείσει τον δέκτη του όταν λάβει ένα μήνυμα από έναν γείτονα του. Αυτό γίνεται για την αποφυγή διπλών μηνυμάτων. Επίσης όταν τα επίπεδα ενέργειας ενός κόμβου είναι χαμηλά αυτός εκπέμπει στους γείτονες του ότι δεν μπορεί να συμμετάσχει στην διαδικασία δρομολόγησης. Η υπολειπόμενη ενέργεια που έχει είναι για να υλοποιεί διεργασίες αίσθησης.

Το επίπεδο διαχείρισης κινητικότητας ανιχνεύει και καταχωρεί την κίνηση των κόμβων αισθητήρα, έτσι ώστε μια δρομολόγηση προς τα πίσω στον χρήστη να είναι πάντα εφικτή και επίσης για να μπορούν οι κόμβοι να είναι ενημερωμένοι πάντα για τους γείτονες τους. Γνωρίζοντας τους γείτονες τους οι κόμβοι αισθητήρων μπορούν να ισορροπούν την χρήση της ισχύς τους και των διεργασιών τους.

Το επίπεδο διαχείρισης διεργασιών μπορεί να ισορροπεί και να οργανώνει τις διεργασίες αίσθησης για μια συγκεκριμένη περιοχή. Δεν χρειάζεται όλοι οι αισθητήρες σε μια περιοχή να εκτελούν διεργασίες αίσθησης την ίδια στιγμή. Σαν αποτέλεσμα κάποιοι κόμβοι αισθητήρες εκτελούν την διεργασία αίσθησης περισσότερες φορές ανάλογα με τα επίπεδα ενέργειας τους.

Αυτά τα επίπεδα διαχείρισης χρειάζονται για να δουλεύουν συνεργατικά οι κόμβοι αισθητήρων με ενεργειακά αποδοτικό τρόπο, να δρομολογούν δεδομένα σε κινητά δίκτυα αισθητήρων και να μοιράζονται πόρους μεταξύ τους. Χωρίς αυτά τα επίπεδα ο κάθε κόμβος θα δούλευε απομονωμένος. Σαν αποτέλεσμα των ανωτέρω είναι αποδοτικότερο οι κόμβοι αισθητήρων να συνεργάζονται για να επιτυγχάνεται η βέλτιστη διάρκεια ζωής ενός δικτύου αισθητήρων.

### **1.10 Εφαρμογές Ασύρματων Δικτύων Αισθητήρων**

Τα ασύρματα δίκτυα αισθητήρων μπορούν να αποτελούνται από πολλών διαφορετικών ειδών αισθητήρες όπως μαγνητικός, θερμικός, οπτικός, υπέρυθρος, ακουστικός, οι οποίοι μπορούν να παρακολουθήσουν ένα μεγάλο εύρος από περιβαλλοντικές συνθήκες όπως: θερμοκρασία, υγρασία, πίεση, ταχύτητα, κατεύθυνση, κίνηση, φως, επίπεδα θορύβου, την παρουσία ή την απουσία συγκεκριμένων αντικειμένων, την μηχανική αντοχή αντικειμένων. Σαν αποτέλεσμα είναι πολύ μεγάλο το εύρος των εφαρμογών που μπορούν να επιτευχθούν. Αυτό το φάσμα των εφαρμογών περιλαμβάνει εφαρμογές Εθνικής Ασφάλειας, καιρικές και κλιματολογικές αναλύσεις και προβλέψεις, παρακολούθηση σεισμικής επιτάχυνσης και δραστηριότητας, παρακολούθηση του περιβάλλοντος κ.α. [7].

### **1.11 Εφαρμογές για έξυπνο σπίτι**

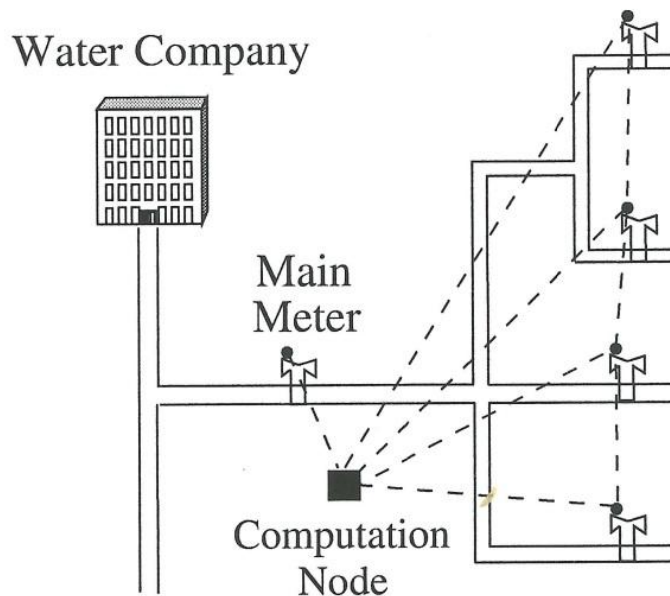
Καθώς η τεχνολογία εξελίσσεται, έξυπνοι κόμβοι αισθητήρων και ενεργοποιητές (actuators) μπορούν να ενσωματωθούν σε συσκευές και εφαρμογές όπως, ηλεκτρικές σκούπες, φούρνους μικροκυμάτων, ψυγεία και καταψύκτες, συσκευές αναπαραγωγής βίντεο καθώς και σε συστήματα παρακολούθησης νερού. Αυτοί οι κόμβοι αισθητήρων μέσα σε συσκευές μπορούν να αλληλεπιδρούν μεταξύ τους και με ένα εξωτερικό δίκτυο μέσω του διαδικτύου ή δορυφορικών επικοινωνιών. Επιτρέπουν στον τελικό χρήστη να διαχειρίζεται εύκολα οικιακές συσκευές είτε τοπικά είτε απομακρυσμένα.

### **1.12 Παρακολούθηση νερού**

Ο κύριος στόχος του συστήματος παρακολούθησης νερού για οικιακή χρήση είναι να εντοπίζει την σπατάλη στην χρήση του νερού και να ενημερώνει τους ενδιαφερομένους για μια πιο αποδοτική χρήση. Επειδή οι εταιρίες ύδρευσης παρέχουν μόνο την συνολική χρήση του νερού για ένα σπίτι δεν είναι εύκολο να καθορίσει κάποιος αναλυτικά τις ποσότητες που

συνεισφέρουν οι διαφορετικές πηγές στην συνολική κατανάλωση. Χρησιμοποιώντας όμως ένα ασύρματο δίκτυο αισθητήρων, τοποθετώντας σε κάθε σωλήνα ξεχωριστά έναν ασύρματο κόμβο αίσθησης μπορεί να γίνει παρακολούθηση του συστήματος ύδρευσης του σπιτιού με χαμηλό κόστος.

Η αρχή λειτουργίας του συστήματος αυτού είναι βασισμένη στο γεγονός ότι η ροή νερού σε συγκεκριμένο σωλήνα μπορεί να υπολογιστεί μετρώντας τις δονήσεις του σωλήνα λόγω τις αναλογικής σχέσης μεταξύ των δύο. Επομένως ασύρματοι αισθητήρες έχουν συνδεθεί στους σωλήνες νερού για να μετράνε τις δονήσεις με χρήση επιταχυνσιόμετρου. Λόγω της μη γραμμικής σχέσης μεταξύ των δονήσεων και της ροής του νερού ο κάθε κόμβος αισθητήρα θα πρέπει να είναι ρυθμισμένος να ανακαλύπτει τους βέλτιστους συνδυασμούς παραμέτρων που συνδέουν τις πληροφορίες επιτάχυνσης με την ροή του νερού. Αντί της χειροκίνητης ρύθμισης που μπορεί να εκτελεστεί με επιπλέον εγκατάσταση αισθητήρα μέτρησης ροής νερού σε κάθε σωλήνα στο παραπάνω σύστημα αυτές οι ρυθμίσεις υλοποιούνται αυτόματα με την βοήθεια του γενικού μετρητή νερού.



**Σχήμα 1.6: Δίκτυο ασύρματων αισθητήρων για την παρακολούθηση ροής όγκου νερού.**

Η αρχιτεκτονική του συστήματος φαίνεται στο σχήμα 1.6, και αποτελείται από τρία είδη στοιχείων. Ένας ασύρματος κόμβος αισθητήρα είναι προσαρμοσμένος στον κεντρικό μετρητή νερού για να συλλέγει πραγματική ροή νερού και να την στέλνει στο υπόλοιπο δίκτυο. Αφού ο κόμβος αυτός είναι ρυθμισμένος, λόγω του κεντρικού ρολογιού από την εταιρία ύδρευσης λειτουργεί σαν κατώφλι αλήθειας για το υπόλοιπο σύστημα. Το επόμενο είδος συστατικού του συστήματος είναι οι αισθητήρες κραδασμών που είναι ενσωματωμένοι σε κάθε σωλήνα για να παρακολουθούνται οι κραδασμοί. Οι πληροφορίες σχετικά με τους κραδασμούς στέλνονται στο κεντρικό κόμβο επεξεργασίας ο οποίος ρυθμίζει αυτόματα τις παραμέτρους των αισθητήρων και υπολογίζει την ροή νερού από κάθε σωλήνα του σπιτιού ξεχωριστά. Η παραμετροποίηση στηρίζεται σε έναν αλγόριθμο βελτιστοποίησης που βασίζεται στο γεγονός ότι το άθροισμα των ροών νερού σε κάθε μη παραμετροποιημένο σωλήνα θα πρέπει να είναι ίσο με την ένδειξη του

κεντρικού ρολογιού ροής νερού. Στη συνέχεια, οι κραδασμοί χρησιμοποιούνται σε ένα μοντέλο ροής για να υπολογιστεί ο ρυθμός ροής σε χωρικά διανεμημένες περιοχές.

Το σύστημα παρέχει πληροφορίες πραγματικού χρόνου σχετικά με την χρήση του νερού σε διαφορετικές τοποθεσίες του συστήματος ύδρευσης του σπιτιού, το οποίο μπορεί να χρησιμοποιηθεί για να βελτιώσει την αποδοτικότητα των σπιτιών στο μέλλον [8].

### 1.13 Περιβαλλοντολογικές Εφαρμογές

Τα ασύρματα δίκτυα αισθητήρων έχουν χρησιμοποιηθεί επίσης σε ακραία περιβάλλοντα όπου η συνεχόμενη παρουσία του ανθρώπου δεν είναι δυνατή. Η παρακολούθηση ηφαιστειών είναι ένα παράδειγμα εφαρμογής σε ακραίες περιβαλλοντικές συνθήκες όπου η παρουσία ανθρώπου δεν είναι δυνατή. Σε αυτή την εφαρμογή το, δίκτυο αισθητήρων τοποθετείται κοντά σε ενεργά ηφαίστεια για συνεχή παρακολούθηση της δραστηριότητάς τους.

Δύο μελέτες έχουν αναπτυχθεί σε δύο ηφαίστεια στο Εκουαδόρ κατά την περίοδο 2004-2005 ως απόδειξη της μελέτης για την παρακολούθηση ηφαιστειών με χρήση ασύρματων δικτύων αισθητήρων. Το 2004 ένα μικρό δίκτυο τριών κόμβων αισθητήρων με ενσωματωμένα μικρόφωνα χρησιμοποιήθηκε για να παρακολουθήσει ένα ηφαίστριο που ξεσπούσε στο κεντρικό Εκουαδόρ. Το 2005, 16 κόμβοι Tmote Sky εξοπλισμένοι με σεισμικούς και ακουστικούς αισθητήρες χρησιμοποιήθηκαν για 19 ημέρες για την παρακολούθηση ενός ενεργού ηφαιστείου στο Βόρειο Εκουαδόρ. Οι κόμβοι αισθητήρων ήταν εξοπλισμένοι με εξωτερικές κεραίες μεγάλου κέρδους για να βελτιώσουν την εμβέλεια της επικοινωνίας και τρεις κόμβοι επικοινωνίας μεγάλης απόστασης χρησιμοποιήθηκαν για να εκπέμπουν τα δεδομένα σε έναν κεντρικό ελεγκτή. Ένας φορητός υπολογιστής εξοπλισμένος με κατευθυντική κεραία χρησιμοποιήθηκε ως πύλη για να συλλέγει τις πληροφορίες και να διαχειρίζεται το δίκτυο απομακρυσμένα.

Ο κύριος στόχος της εφαρμογής ήταν να συλλέξει σεισμικές πληροφορίες που βασίζονταν σε σεισμούς που συνέβαιναν κοντά σε ηφαίστεια. Καθώς οι σεισμοί συνήθως κρατούν λιγότερο από 60 δευτερόλεπτα, χρησιμοποιήθηκε μεγάλος ρυθμός δειγματοληψίας στον σεισμικό αισθητήρα (π.χ. 100Hz), όμως αυτό περιόρισε την δυνατότητα τοπικής αποθήκευσης πληροφοριών σε 20 λεπτά ηφαιστειακής δραστηριότητας. Κάθε κόμβος αισθητήρα χρησιμοποιεί την τοπική μνήμη σαν κυκλικό ρυθμιστή που φιλτράρει την συλλεγόμενη πληροφορία με την χρήση ανιχνευτή καταωφλίου βραχυπρόθεσμου μέσου όρου/μακροπρόθεσμου μέσου όρου για να ξεχωρίζει τα γεγονότα που σχετίζονται με την ηφαιστειακή δραστηριότητα. Όταν ένα γεγονός δημιουργείται αυτό το γεγονός αναφέρεται στην πύλη. Αν ένα γεγονός υψηλής εμπιστοσύνης δημιουργηθεί με πολλαπλές αναφορές η πύλη στέλνει πίσω στο δίκτυο μια εντολή συλλογής δεδομένων. Σαν απάντηση σε αυτή την εντολή κάθε κόμβος εκπέμπει τις πληροφορίες που έχει αποθηκευμένες στην μνήμη του. Συνεπώς ένα μεγάλο κομμάτι από δεδομένα εκπέμπεται στην πύλη μόνο όταν ένα γεγονός ενδιαφέροντος ανιχνευθεί από ένα μικρό αριθμό κόμβων. Αυτό μειώνει την σπατάλη εύρους ζώνης και την κατανάλωση ενέργειας σε κάθε κόμβο.

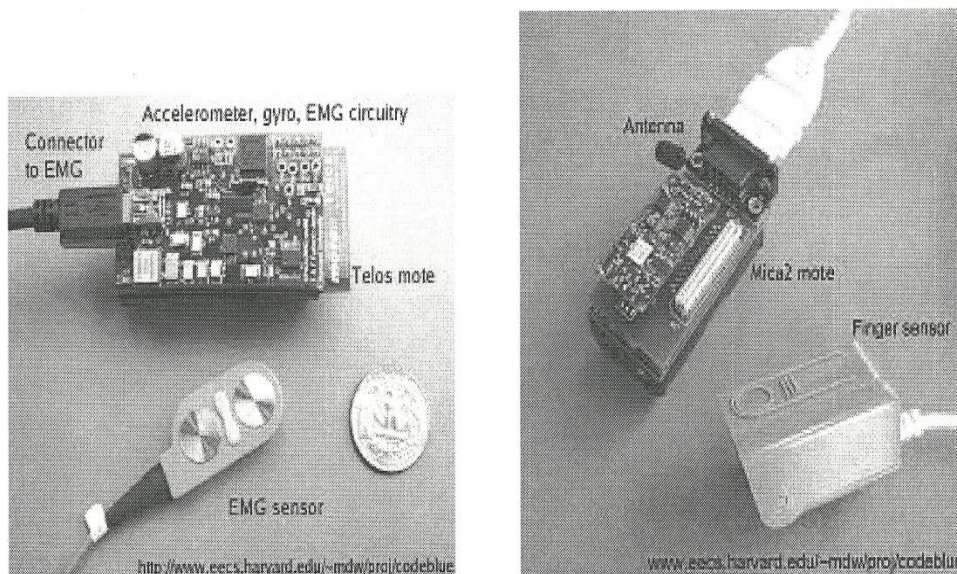
Οι κόμβοι αισθητήρων είναι συγχρονισμένοι χρησιμοποιώντας το flooding time synchronization protocol και μια μονάδα GPS με ενσωματωμένο κόμβο MicaZ παρέχει πληροφορίες σχετικά με την τοποθεσία. Τα αποτελέσματα αυτής της εφαρμογής παρέχουν σημαντικές πληροφορίες που σχετίζονται με φυσικές διεργασίες που συμβαίνουν στο εσωτερικό των ηφαιστειών [9].



## 1.14 Εφαρμογές σε Υπηρεσίες Υγείας

Εφαρμογή παρακολούθησης ασθενή:

Το σχέδιο CodeBlue του Πανεπιστημίου Χάρβαρντ εστιάζει σε φορετούς(wearable) αισθητήρες που παρακολουθούν τα ζωτικά σήματα ενός ασθενή στην καθημερινή ζωή του. Για να γίνει εφικτό αυτό πλακέτες αισθητήρων με κυκλώματα παλμικού οξύμετρου, ηλεκτροκαρδιογράφου, και ηλεκτρομυογράφου έχουν σχεδιαστεί για τους κόμβους MicaZ και Telos όπως φαίνονται στην εικόνα 1.2. Συνεπώς οι σφυγμοί, ο κορεσμός οξυγόνου στο αίμα, οι ηλεκτρικές δραστηριότητες της καρδιάς, η κίνηση του ασθενή και η μυϊκή δραστηριότητα μπορούν να καταγράφονται σε συνεχή βάση. Η πλατφόρμα CodeBlue χρησιμοποιεί τρεις κόμβους που λειτουργούν σε δικτυακή ρύθμιση και έτσι μπορεί το ιατρικό προσωπικό να παρακολουθεί τους ασθενείς με την χρήση PDA.

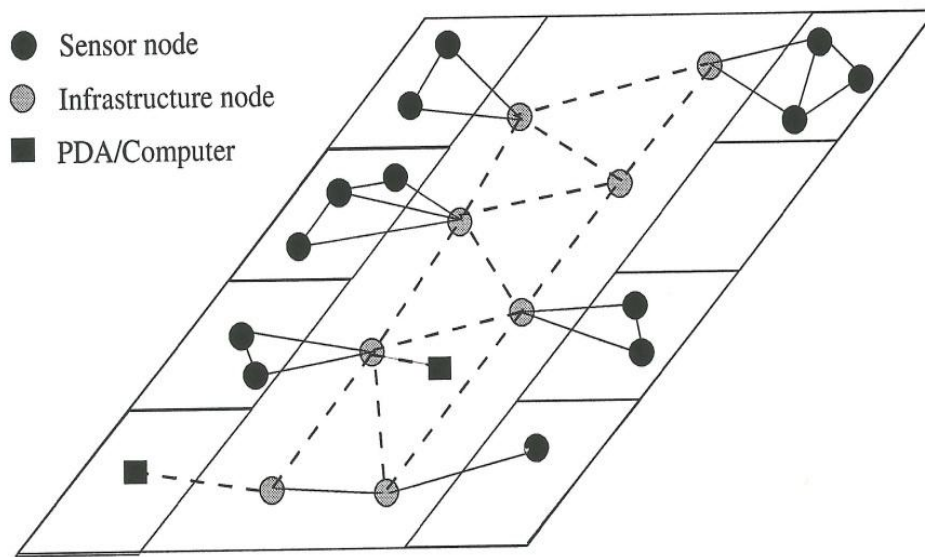


**Εικόνα 1.2: Υλοποιήσεις της εφαρμογής Code Blue με Telos mote και MicaZ mote.**

Η αρχιτεκτονική του δικτύου του Code Blue φαίνεται στο σχήμα 1.7 όπως θα ήταν οργανωμένο σε ένα νοσοκομείο. Συνεπώς κόμβοι αισθητήρων με αρκετούς αισθητήρες είναι τοποθετημένοι πάνω σε ασθενείς για παρακολούθηση. Επιπρόσθετοι κόμβοι είναι τοποθετημένοι μέσα στο νοσοκομείο για να παρέχουν την υποδομή για την παροχή μηνυμάτων από και προς τους κόμβους. Το ιατρικό προσωπικό μπορεί να αποκτήσει πρόσβαση στο δίκτυο μέσω υπολογιστών και PDA. Το δίκτυο είναι βασισμένο σε έναν μηχανισμό δημοσίευσης/εγγραφής, όπου οι κόμβοι αισθητήρων δημοσιοποιούν τα διαθέσιμα δεδομένα τους και το ιατρικό



προσωπικό εγγράφεται σε αυτά τα δεδομένα με βάση συγκεκριμένες συνθήκες. Ο μηχανισμός δημοσίευσης/εγγραφής είναι βασισμένος στο πρωτόκολλο δρομολόγησης προσαρμοστικής ζήτησης πολλαπλής διανομής (Adaptive Demand-Drive Multicast Routing (ADMR)) το οποίο εκτελεί πληροφορίες δρομολόγησης βασισμένο στις δημοσιευμένες πληροφορίες. Κάθε φορά που ένας κόμβος δημοσιεύει δεδομένα κάθε κόμβος τους δικτύου ενημερώνεται σχετικά με το βέλτιστο μονοπάτι για αυτόν τον αρχικό κόμβο. Η εύρεση του μονοπατιού ανάμεσα σε κόμβους γίνεται περιοδικά με περιορισμένη πλημμύρα για να προσαρμόζεται το ασύρματο κανάλι στις αλλαγές.



**Σχήμα1.7: Αρχιτεκτονική δικτύου εφαρμογής Code Blue.**

Όταν ένας κόμβος αισθητήρα είναι συνδεδεμένος στο δίκτυο οι δυνατότητες του εκπέμπονται με την χρήση ενός πρωτοκόλλου ανακάλυψης. Η εγγραφή σε δημοσιευμένα δεδομένα γίνεται μέσω του επίπεδου ερώτησης CodeBlue (CBQ) που προσδιορίζει την ερώτηση με χρήση μιας πλειάδας  $\{S, \tau, \text{chan}, \rho, C, p\}$  όπου  $S$  είναι η ομάδα εγγεγραμμένων κόμβων, το  $\tau$  είναι ο τύπος του εγγεγραμμένου αισθητήρα,  $\text{chan}$  είναι ο τύπος του καναλιού που χρησιμοποιείται για την δρομολόγηση, το  $\rho$  είναι ο ρυθμός δειγματοληψίας, και το  $C$  είναι ο βέλτιστος μέγιστος αριθμός που πρέπει να αναφερθούν. Για να μοντελοποιηθεί το ενδιαφέρον των χρηστών ένα φίλτρο  $p$  χρησιμοποιείται στην ερώτηση όπου φιλτράρει τις πληροφορίες ανάλογα με τους τύπους των αισθητήρων. Σύμφωνα με τα παραπάνω ένας γιατρός μπορεί να καθορίσει ένα συγκεκριμένο αισθητήρα να στέλνει πληροφορίες για έναν συγκεκριμένο ασθενή όταν οι ενδείξεις του υπερβούν τα φυσιολογικά όρια.

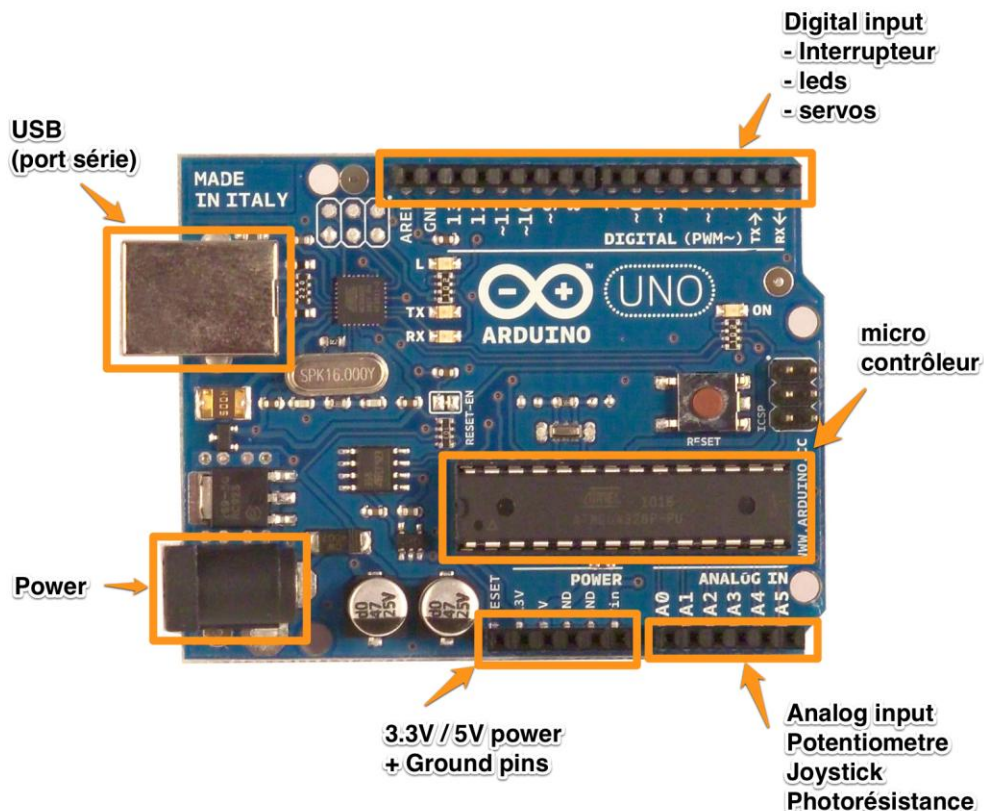
Επιπροσθέτως του μηχανισμού δημοσίευσης/εγγραφής το πρόγραμμα CodeBlue παρέχει υπηρεσίες τοποθεσίας με την βοήθεια μιας ομάδας κόμβων που εκπέμπουν περιοδικά πιλοτικά μηνύματα. Ένας κινητός κόμβος μπορεί να υπολογίσει την θέση του με βάση ένα λαμβανόμενο πιλοτικό σήμα. Αυτή η υπηρεσία είναι βασική σε νοσοκομεία για τον εντοπισμό ιατρών, ασθενών, νοσοκόμων και σημαντικών μηχανημάτων. Τέλος παρέχεται και ένα γραφικό περιβάλλον στους χρήστες και στο ιατρικό προσωπικό έτσι ώστε να μπορούν εύκολα να χρησιμοποιούν το σύστημα [10].

## ΚΕΦΑΛΑΙΟ 2 - Η ΠΛΑΤΦΟΡΜΑ ΑΝΑΠΤΥΞΗΣ ΕΦΑΡΜΟΓΩΝ ΔΙΚΤΥΩΝ ΑΙΣΘΗΤΗΡΩΝ ARDUINO

### 2.1 Arduino Uno

Το Arduino Uno είναι μια πλακέτα μικροελεγκτή βασισμένο στον επεξεργαστή ATmega 328. Έχει 14 συνδέσμους ψηφιακής εισόδου/εξόδου (έξι από τους οποίους μπορούν να χρησιμοποιηθούν για έξοδο PWM), 6 αναλογικές εισόδους, έναν κρυσταλλικό ταλαντωτή στα 16MHz, μια σύνδεση USB, ένα βύσμα τροφοδοσίας, βύσμα ICSP και ένα κουμπί επανεκκίνησης. Περιέχει ότι χρειάζεται για να υπάρχει η δυνατότητα ελέγχου του μικροελεγκτή.

Το Arduino Uno διαφέρει από τις άλλες αντίστοιχες πλακέτες διότι δεν χρησιμοποιεί τον οδηγό FTDI για μετατροπή του USB σε σειριακή σύνδεση, αλλά έχει το ολοκληρωμένο ATmega16U2 που είναι προγραμματισμένο να λειτουργεί σαν μετατροπέας σύνδεσης από USB σε σειριακή μορφή [11]. Στην εικόνα 2.1 φαίνονται τα κύρια μέρη του Arduino και στην εικόνα 2.2 τα Arduino uno r3, Arduino uno r2 και Arduino uno SMD.

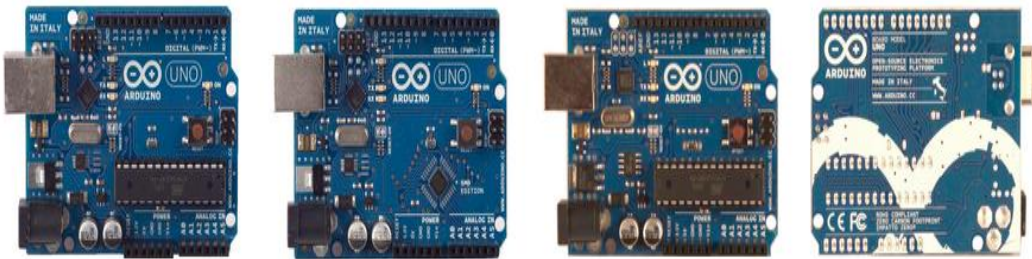


Εικόνα 2.1: Τα κύρια μέρη του Arduino uno.



Arduino Uno R3 Front

Arduino Uno R3 Back



Arduino Uno R2 Front

Arduino Uno SMD

Arduino Uno Front

Arduino Uno Back

**Εικόνα 2.2: Όψεις της πλακέτας Arduino Uno.**

**Πίνακας 2: Χαρακτηριστικά Arduino Uno.**

Microcontroller	ATmega328
Operating Voltage	5V
Input Voltage (recommended)	7-12V
Input Voltage (limits)	6-20V
Digital I/O Pins	14 (of which 6 provide PWM output)
Analog Input Pins	6
DC Current per I/O Pin	40 mA
DC Current for 3.3V Pin	50 mA
Flash Memory	32 KB (ATmega328) of which 0.5 KB used by bootloader
SRAM	2 KB (ATmega328)
EEPROM	1 KB (ATmega328)
Clock Speed	16 MHz

## 2.2 Τροφοδοσία

Το Arduino Uno μπορεί να τροφοδοτηθεί είτε από την θύρα USB είτε με εξωτερική τροφοδοσία. Η εξωτερική τροφοδοσία μπορεί να είναι ή με την χρήση μετασχηματιστή από εναλλασσόμενη τάση σε συνεχή ή με την χρήση μπαταριών. Ο μετασχηματιστής συνδέεται μέσω βύσματος 2,1 χιλιοστών με τον θετικό πόλο στο κέντρο, στο βύσμα που βρίσκεται ενσωματωμένο στο Arduino. Αν δεν υπάρχει μετασχηματιστής, μπορεί να συνδεθεί μπαταρία στο ενσωματωμένο βύσμα 2,1 χιλιοστών.

Η πλακέτα μπορεί να λειτουργήσει με εξωτερική τροφοδοσία από 6 έως 20 V. Αν όμως τροφοδοτηθεί με λιγότερο από 7 V, ο σύνδεσμος τροφοδοσίας 5 V που διαθέτει η πλακέτα ίσως να μην μπορέσει να παράξει 5 V και η πλακέτα να είναι ασταθής. Αν χρησιμοποιηθούν περισσότερα από 12 V για τροφοδοσία ο ρυθμιστής τάσης μπορεί να υπερθερμανθεί και να χαλάσει την πλακέτα. Οπότε προκύπτει ότι η προτεινόμενη τροφοδοσία κυμαίνεται από 7 έως 12 V.

Οι σύνδεσμοι τροφοδοσίας είναι όπως ακολουθούν:

- **Vin**: Είναι ίση με την τάση τροφοδοσίας που έχει εφαρμοστεί στο Arduino. Μπορεί να τροφοδοτήσει εξωτερικές πλακέτες.
- **5V**: Προσφέρει τάση 5V που έχει προκύψει από τον ρυθμιστή τάσης της πλακέτας σε περίπτωση που η τάση τροφοδοσίας είναι μεγαλύτερη των 5V.
- **3,3V**: Προσφέρει 3,3 V που προέρχονται από τον ρυθμιστή τάσης της πλακέτας. Το μεγαλύτερο ρεύμα που τραβάει είναι 50mA.
- **GND**: Σύνδεσμος γείωσης.

## 2.3 Μνήμη

Ο μικροελεγκτής ATmega328 διαθέτει 32KB μνήμη (όπου τα 0,5kB χρησιμοποιούνται από τον φορτωτή εκκίνησης (bootloader). Διαθέτει επίσης 2kB μνήμη τυχαίας προσπέλασης τύπου SRAM και 1kB μνήμη τύπου EEPROM.

## 2.4 Είσοδοι και Έξοδοι

Κάθε ένας από τους 14 συνδέσμους του Arduino Uno μπορεί να χρησιμοποιηθεί σαν είσοδος ή σαν έξοδος κάνοντας χρήση της συνάρτησης `pinMode()`, `digitalWrite()` και `digitalRead()`. Λειτουργούν στα 5V. Κάθε σύνδεσμος μπορεί να προσφέρει ή να τραβήξει το μέγιστο 40mA και διαθέτει μια εσωτερική αντίσταση της τάξης των 20-50kΩ. Επιπρόσθετα μερικοί σύνδεσμοι έχουν κάποιες ξεχωριστές δυνατότητες.

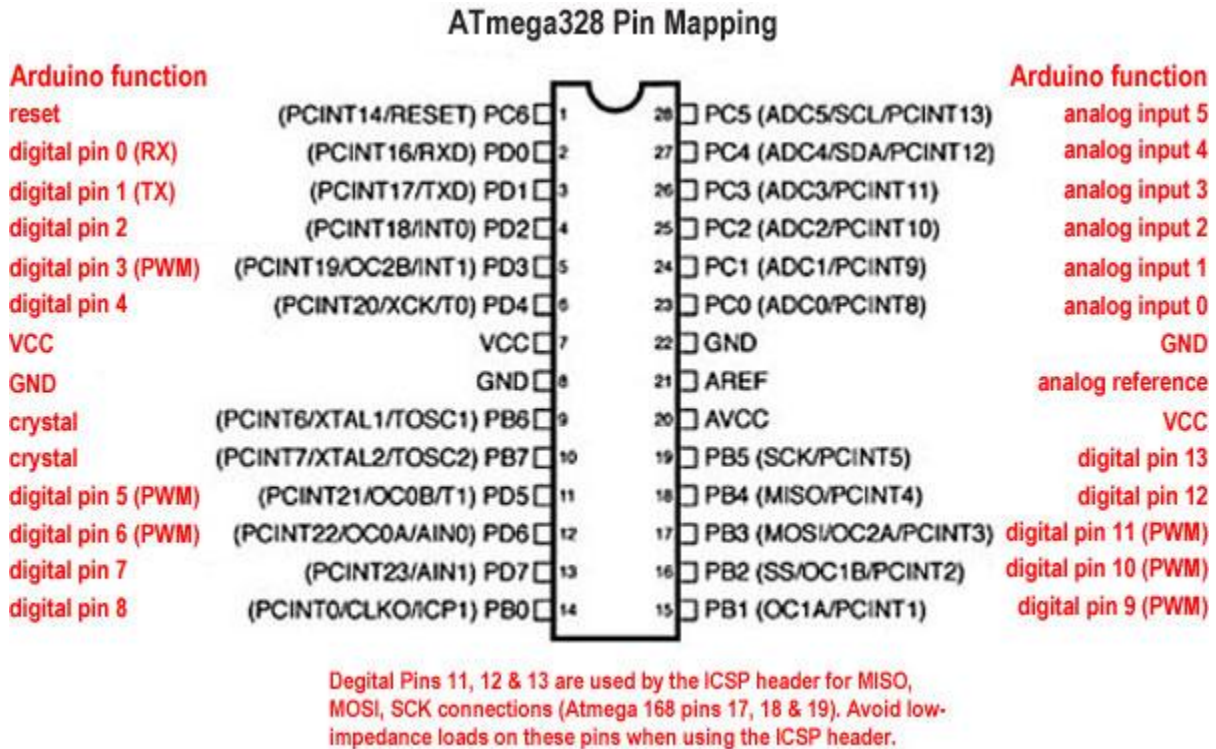
- **Serial**: 0 (RX) και 1 (TX). Αυτοί οι σύνδεσμοι μπορούν να χρησιμοποιηθούν για να λάβουν ή να εκπέμψουν σειριακά δεδομένα και συνδέονται στους αντίστοιχους του ATmega8U2.
- **PWM**: Οι σύνδεσμοι 3,5,6,9,10 και 11 παρέχουν 8-bit PWM έξοδο κάνοντας χρήση της συνάρτησης `analogWrite()`.
- **SPI**: Οι σύνδεσμοι 10(SS), 11(MOSI), 12(MISO) και 13(SCK) χρησιμοποιούνται για επικοινωνίες τύπου SPI κάνοντας χρήση της αντίστοιχης βιβλιοθήκης SPI.
- **LED**: Ο σύνδεσμος 13 έχει συνδεδεμένο ένα led και όταν η κατάσταση της θύρας είναι HIGH αυτό ανοίγει και όταν είναι LOW αυτό σβήνει.

Το Arduino Uno έχει και 6 αναλογικές θύρες με ονόματα από A0 έως A5 που η κάθε μια είναι σε θέση να προσφέρει 10 bit ανάλυση. Σαν προεπιλογή μπορούν να μετρήσουν από 0 μέχρι 5V, όμως μπορεί να αλλάξει το επάνω αυτό όριο του εύρους μέτρησης χρησιμοποιώντας την θύρα AREF και την συνάρτηση `analogReference()`.

- **I2C**: Οι θύρες A4 ή SDA και A5 ή SCL υποστηρίζουν επικοινωνία τύπου I2C με χρήση της βιβλιοθήκης `Wire`.



- **AREF:** Η θύρα αυτή προσφέρει την τάση αναφοράς και μπορεί να χρησιμοποιηθεί μαζί με την συνάρτηση analogReference().
- **Reset:** Αυτή η θύρα όταν βρεθεί στην κατάσταση του λογικού 0 επανακκινεί τον μικροελεγκτή.



Σχήμα 2.1: Διασυνδέσεις μικροελεγκτή ATmega328.

## 2.5 Επικοινωνία

Το Arduino Uno διαθέτει μια σειρά από υποδομές για να μπορεί να επικοινωνεί με τον υπολογιστή, άλλα Arduino καθώς και με άλλους εξωτερικούς μικροελεγκτές. Ο μικροελεγκτής ATmega328 προσφέρει σειριακή επικοινωνία με χρήση του UART TTI(5V), το οποίο είναι διαθέσιμο στις ψηφιακές θύρες 0 και 1 (RX) και (TX) αντίστοιχα. Ο μικροελεγκτής ATmega16U2 μετατρέπει την σειριακή επικοινωνία σε μορφή USB και έτσι προκύπτει η εμφάνιση μιας εικονικής θύρας επικοινωνίας στον υπολογιστή. Τα led Tx και Rx αναβοσβήνουν όποτε η πλακέτα του Arduino δέχεται ή στέλνει δεδομένα διαμέσου της θύρας USB.

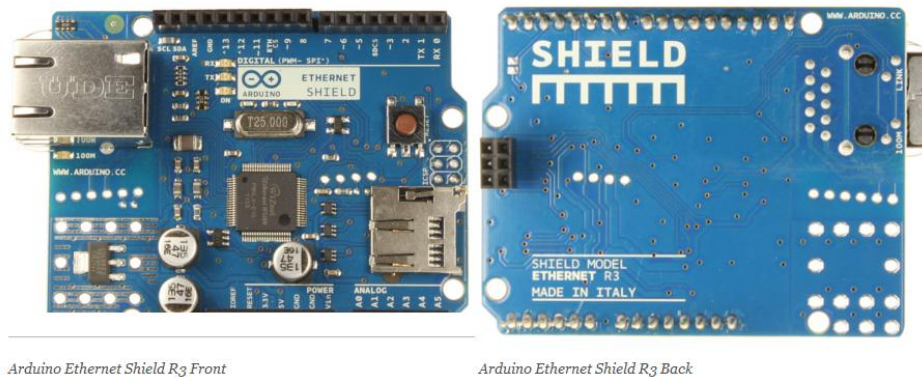
## 2.6 Προγραμματισμός

Το Arduino μπορεί να προγραμματιστεί με χρήση του προγράμματος Arduino IDE. Ο μικροελεγκτής ATmega328 έχει προεγκατεστημένο ένα πρόγραμμα εκκίνησης που επιτρέπει να

φορτώνονται κώδικες στην πλακέτα χωρίς να χρειάζεται εξωτερική πλακέτα προγραμματισμού. Η επικοινωνία με τον μικροελεγκτή γίνεται με χρήση του πρωτοκόλλου STK500.

## 2.7 Arduino Ethernet Shield

Η πλακέτα Arduino Ethernet συνδέει το Arduino με το Internet. Η πλακέτα αυτή συνδέεται στην πλακέτα του Arduino μέσω των θυρών SPI. Η τάση λειτουργίας της είναι 5V που τα προσφέρει η πλακέτα του Arduino. Η πλακέτα Ethernet διαθέτει έναν ελεγκτή Ethernet W5100 με ενσωματωμένη 16kB μνήμη. Η ταχύτητα επικοινωνίας είναι της μορφής 10/100 Mb. Το ολοκληρωμένο W5100 της Wiznet παρέχει την στοίβα δικτύου (IP) που είναι ικανή για υλοποίηση προτύπων TCP και UDP. Υποστηρίζει μέχρι και 5 ταυτόχρονες συνδέσεις τύπου socket. Με την χρήση της βιβλιοθήκης Ethernet μπορούν να γραφτούν προγράμματα που θα έχουν πρόσβαση στο Internet.



Arduino Ethernet Shield R3 Front

Arduino Ethernet Shield R3 Back

### Εικόνα 2.3: Όψεις πλακέτας Ethernet .

Η πλακέτα Ethernet διαθέτει βύσμα τύπου RJ-45 για καλώδιο ethernet και κουμπί για επανεκκίνηση της λειτουργίας της. Επίσης διαθέτει ενσωματωμένη υποδοχή για κάρτα μνήμης τύπου micro-SD η οποία μπορεί να χρησιμοποιηθεί για αποθήκευση αρχείων που θα χρησιμοποιούνται ή θα διακινούνται στο δίκτυο. Η υποδοχή της κάρτας μνήμης micro-SD μπορεί επίσης να διαβάσει την κάρτα μνήμης με χρήση της βιβλιοθήκης SD.

Το Arduino επικοινωνεί με την πλακέτα Ethernet μέσω των θυρών SPI. Αυτές οι θύρες είναι οι 11, 12 και 13. Η πλακέτα επίσης διαθέτει και μια σειρά από led που ο ρόλος τους είναι να παράγουν βοηθητικές ενδείξεις.

- **PWR:** Το led αυτό δείχνει ότι η πλακέτα Ethernet έχει τροφοδοσία.
- **LINK:** Το Led αυτό δείχνει αν υπάρχει συνδεδεμένο καλώδιο δικτύου στην πλακέτα και επίσης αναβοσβήνει όταν η πλακέτα δέχεται ή στέλνει δεδομένα.
- **FULLD:** Αυτό το led δείχνει αν η σύνδεση είναι πλήρως αμφίδρομη.
- **100Mb:** Αυτό το led δείχνει την ταχύτητα της σύνδεσης στο δίκτυο, αν είναι αναμμένο τότε υποστηρίζεται ταχύτητα 100Mb.
- **RX:** Αυτό το led αναβοσβήνει όταν η πλακέτα λαμβάνει δεδομένα.



- **TX:** Αυτό το led αναβοσβήνει όταν η πλακέτα στέλνει δεδομένα.
- **COLL:** Το led αυτό αναβοσβήνει όταν εντοπιστεί κάποια σύγκρουση στο δίκτυο.

## 2.8 Relay Shield

Η πλακέτα ρελέ (Relay Shield) είναι μια πλακέτα με τέσσερα ρελέ που παρέχει ευκολία στις εφαρμογές που απαιτούν έλεγχο υψηλής και χαμηλής τάσης. Η μέγιστη ισχύς που μπορεί να χρησιμοποιηθεί είναι 90W για συνεχή τάση ή 360V για εναλλασσόμενη τάση. Η πλακέτα μπορεί να ελεγχθεί απευθείας από το Arduino διαμέσου των ψηφιακών συνδέσμων εισόδου/εξόδου που παρέχει το Arduino με εξωτερική τροφοδοσία 9V όπως φαίνεται στην εικόνα 2.5. Με την υποδοχή RFBee και την πλακέτα 315MHZ RF όπως εμφανίζεται στην εικόνα 2.6 η πλακέτα ρελέ μπορεί να ελεγχθεί απομακρυσμένα, γεγονός που την κάνει ιδανική για εφαρμογές ρομποτικής, έλεγχο βιομηχανικής φύσεως και εφαρμογές έξυπνων σπιτιών.



**Εικόνα 2.4: Πλακέτα Relay**



**Εικόνα 2.5: Πλακέτα Relay ενσωματωμένη σε πλακέτα Arduino.**



**Εικόνα 2.6: Πλακέτα Relay με ενσωματωμένο 315MHZ RF για ασύρματη επικοινωνία.**

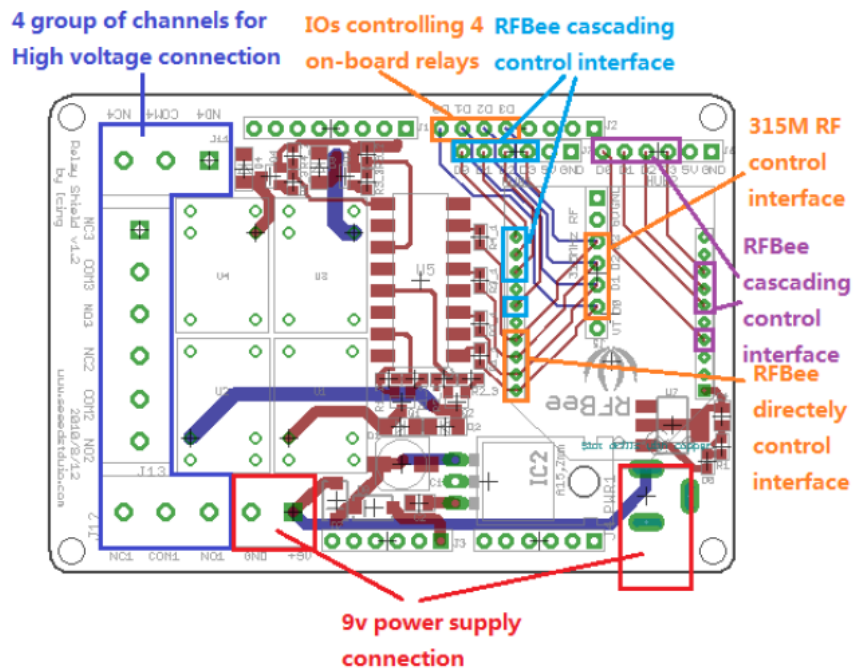
Τα κύρια χαρακτηριστικά της πλακέτας Relay συνοψίζονται ως εξής:

- Τέσσερα ρελέ με κύκλωμα φωτοσύζευξης
- Εύκολη εγκατάσταση με βίδες
- Μικρό βάρος
- Μικρό μέγεθος
- Συμβατή με Arduino

**Πίνακας 3: Χαρακτηριστικά λειτουργίας ανά κανάλι.**

Contact Rating	3A AC 120V / DC 24V
Maximal Switching Voltage	AC 240V / DC 60V
Max Switching Current	5A
Max Switching Power	AC 360VA / DC 90W
Electrical Life Expectancy	100,000 Operations
Mechanical Life Expectancy	10,000,000 Operations
Safety Standard(relay)	UL cUL TUV CQC
Working Voltage	9VDC
Weight	165g
Working temperature	-30°C to +85°C
Working Humidity	40% - 85%

## 2.9 Διάγραμμα Μπλοκ



**Σχήμα 2.2: Διάγραμμα μπλοκ του Relay Shield.**

Στο ανωτέρω σχήμα αναδεικνύεται η διασύνδεση των modules της πλακέτα Relay σε λογικό επίπεδο. Διακρίνονται τα εξής στοιχεία:

### Ομάδα 1

- **COM1**: Κοινός ακροδέκτης
- **NC1**: Κανονικά κλειστός, στην περίπτωση που ο NC1 είναι συνδεδεμένος με τον COM1 τότε στην είσοδο D0 εφαρμόζεται λογικό μηδέν αλλιώς αν εφαρμόζεται λογικό ένα ο NC1 δεν είναι συνδεδεμένος με τον COM1.
- **NO1**: Κανονικά ανοικτός, στην περίπτωση που εφαρμόζεται λογικό μηδέν στην είσοδο D0 τότε ο NO1 δεν είναι συνδεδεμένος με τον COM1. Αν στην είσοδο D0 εφαρμόζεται λογικό ένα τότε το NO1 είναι συνδεδεμένος με τον COM1.

Η πλακέτα Relay διαθέτει άλλες τρεις ομάδες που έχουν ακριβώς την ίδια λογική με την παραπάνω μόνο που αλλάζει η είσοδος ελέγχου και γίνεται αντίστοιχα D1,D2,D3.

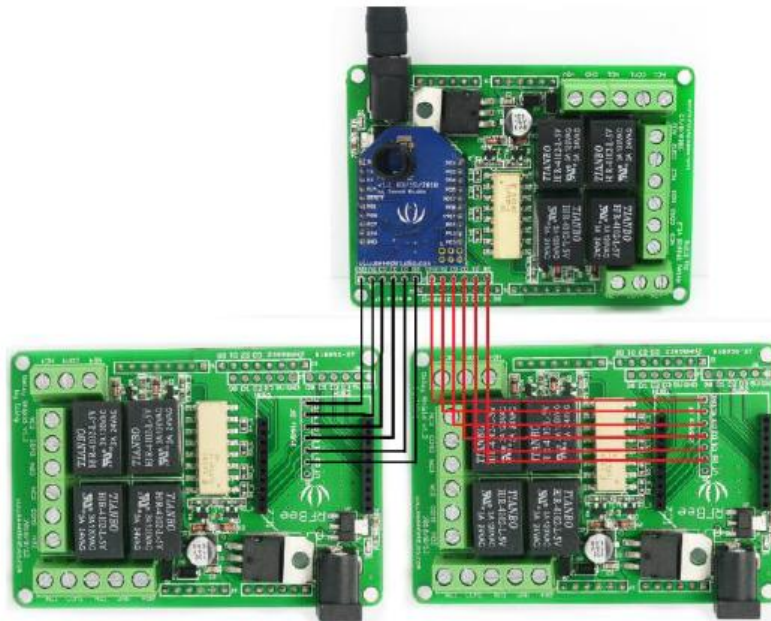
Βύσμα τροφοδοσίας για 9V πηγή τάσης

**Είσοδοι/Εξοδοι ελέγχου ρελέ:** Στην πλακέτα υπάρχουν τέσσερις ακροδέκτες ελέγχου D0, D1, D2, D3 που με αυτούς ελέγχονται τα ρελέ και πρέπει να είναι συνδεδεμένοι στους ακροδέκτες του Arduino 4 έως 7, για να υπάρχει δυνατότητα να ελέγχονται τα ρελέ από το Arduino.

**Διεπαφή Ελέγχου 315MHz RF:** Μαζί με τους ακροδέκτες της γείωσης GND και των 5V υπάρχει και η διεπαφή για την πλακέτα 315MHz RF για να ελέγχονται τα ρελέ απομακρυσμένα.

**Διεπαφή απευθείας ελέγχου RFBee:** Τα τέσσερα ρελέ μπορούν να ελεγχθούν απευθείας από την πλακέτα RFBee με την χρήση των 4 DIO συνδέσμων του RFBee.

**RFBee cascading διεπαφή ελέγχου:** Καθώς το RFBee έχει 12 συνδέσμους DIO ένα RFBee μπορεί να ελέγξει δύο πλακέτες Relay όπως υποδεικνύεται στην εικόνα 2.7.



**Εικόνα 2.7:** Ένα RFBee ελέγχει δύο πλακέτες Relay .

## 2.10 XBee Shield



**Εικόνα2.8: XBee Shield στα δεξιά και XBee module στα αριστερά.**

Η πλακέτα XBEE επιτρέπει στο Arduino να επικοινωνεί ασύρματα με άλλους κόμβους κάνοντας χρήση του πρωτοκόλλου ZigBee. Επίσης επιτρέπει τον προγραμματισμό του module XBee κάνοντας χρήση της θύρας USB του Arduino.

## 2.11 Module XBee

Το module XBee κατασκευάστηκε με βάση τα χαρακτηριστικά του προτύπου IEEE 802.15.4 για να πληρεί τις ανάγκες για χαμηλή κατανάλωση και χαμηλό κόστος. Η συχνότητα λειτουργίας του είναι 2.4GHz όπως προβλέπεται από το πρότυπο IEEE 802.15.4.

Το XBee προσφέρει ασύρματη συνδεσιμότητα σε συσκευές. Χρησιμοποιεί το πρωτόκολλο IEEE 802.15.4 για γρήγορη δικτύωση σημείου σε πολλαπλά σημεία και σημείο σε σημείο. Σχεδιάστηκε για εφαρμογές που απαιτούν μεγάλη διαπερατότητα, χαμηλή καθυστέρηση και προβλεπόμενο χρόνο επικοινωνιών.

## 2.12 Τεχνικά Χαρακτηριστικά

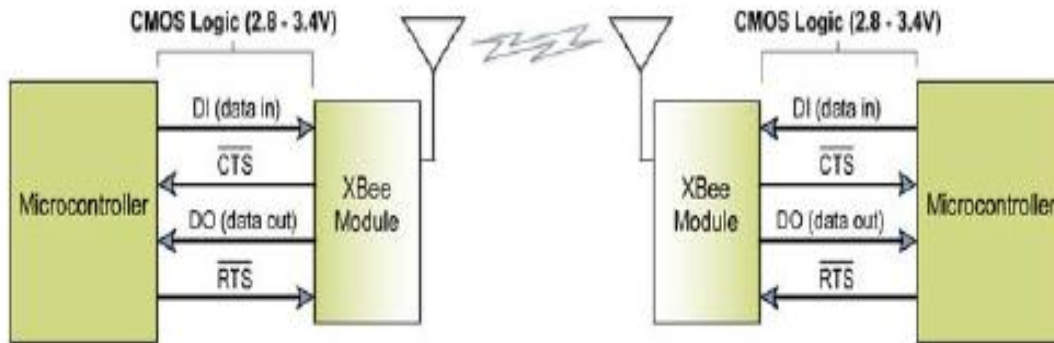
**Πίνακας 4: Βασικά τεχνικά χαρακτηριστικά λειτουργίας του module XBee.**

Specification	XBee	XBee-PRO
<b>Performance</b>		
Indoor/Urban Range	up to 100 ft. (30 m)	Up to 300' (100 m)
Outdoor RF line-of-sight Range	up to 300 ft. (100 m)	Up to 1 mile (1500 m)
Transmit Power Output (software selectable)	1mW (0 dBm)	60 mW (18 dBm) conducted, 100 mW (20 dBm) EIRP*
RF Data Rate	250,000 bps	250,000 bps
Serial Interface Data Rate (software selectable)	1200 - 115200 bps (non-standard baud rates also supported)	1200 - 115200 bps (non-standard baud rates also supported)
Receiver Sensitivity	-92 dBm (1% packet error rate)	-100 dBm (1% packet error rate)
<b>Power Requirements</b>		
Supply Voltage	2.8 – 3.4 V	2.8 – 3.4 V
Transmit Current (typical)	45mA (@ 3.3 V)	If PL=0 (10dBm): 137mA(@3.3V), 139mA(@3.0V) PL=1 (12dBm): 155mA (@3.3V), 153mA(@3.0V) PL=2 (14dBm): 170mA (@3.3V), 171mA(@3.0V) PL=3 (16dBm): 188mA (@3.3V), 195mA(@3.0V) PL=4 (18dBm): 215mA (@3.3V), 227mA(@3.0V)
Idle / Receive Current (typical)	50mA (@ 3.3 V)	55mA (@ 3.3 V)
Power-down Current	< 10 $\mu$ A	< 10 $\mu$ A
<b>General</b>		
Operating Frequency	ISM 2.4 GHz	ISM 2.4 GHz
Dimensions	0.960" x 1.087" (2.438cm x 2.761cm)	0.960" x 1.297" (2.438cm x 3.294cm)
Operating Temperature	-40 to 85° C (industrial)	-40 to 85° C (industrial)
Antenna Options	Integrated Whip, Chip or U.FL Connector	Integrated Whip, Chip or U.FL Connector
<b>Networking &amp; Security</b>		
Supported Network Topologies	Point-to-point, Point-to-multipoint & Peer-to-peer	
Number of Channels (software selectable)	16 Direct Sequence Channels	12 Direct Sequence Channels
Addressing Options	PAN ID, Channel and Addresses	
<b>Agency Approvals</b>		
United States (FCC Part 15.247)	OUR-XBEE	OUR-XBEEPRO
Industry Canada (IC)	4214A XBEE	4214A XBEEPRO
Europe (CE)	ETSI	ETSI (Max. 10 dBm transmit power output)*
Japan	n/a	005NYCA0376 (Max. 10 dBm transmit power output)**

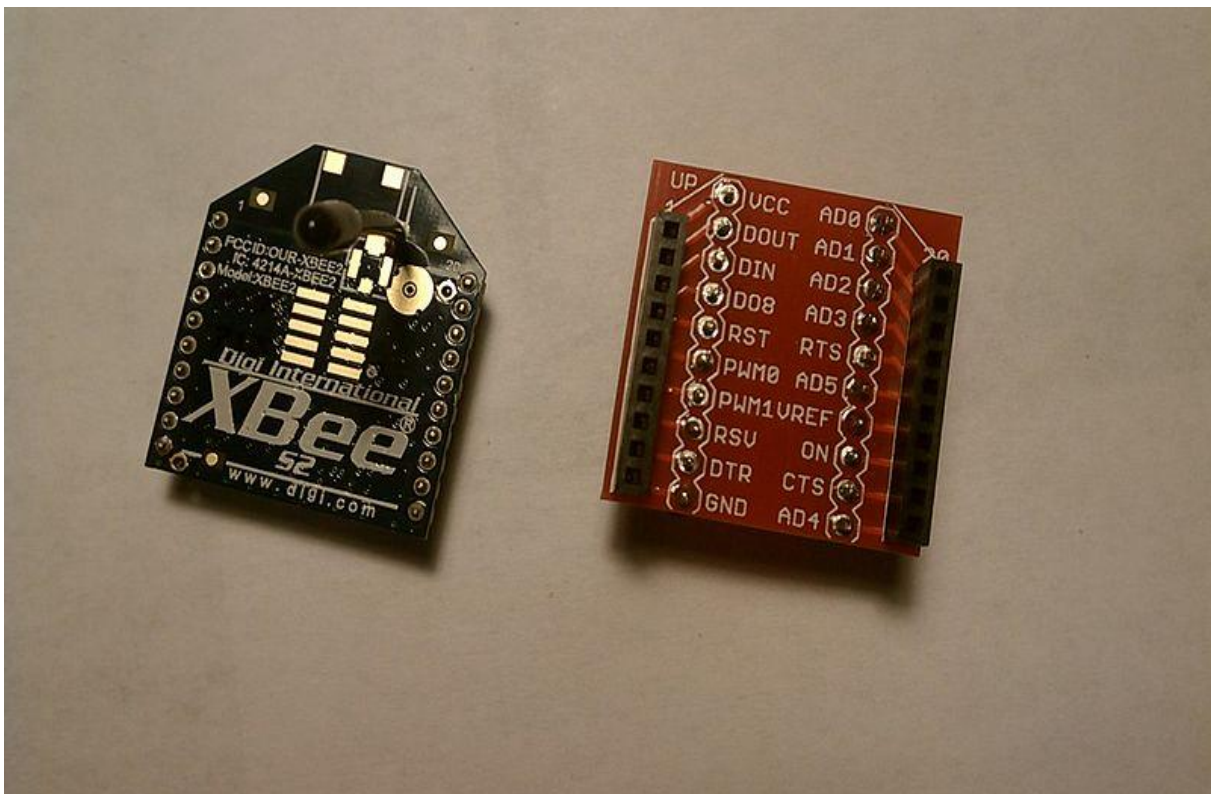
Το XBee έχει ισχύ εξόδου 1mW και η ακτίνα εκπομπής και λήψης του κυμαίνεται από τα 30 μέτρα για εσωτερικούς χώρους και 100 μέτρα για εξωτερικούς χώρους. Ο ρυθμός μετάδοσης δεδομένων στο ασύρματο κανάλι είναι 250kbps και η ευαισθησία του δέκτη του είναι -91dBm. Υποστηρίζει έλεγχο σφαλμάτων με χρήση επαναλήψεων ή επιβεβαιώσεων. Έχει χωρητικότητα 16 καναλιών με το κάθε κανάλι να μπορεί να έχει μέχρι και 65.000 διευθύνσεις. Η τάση τροφοδοσίας του κυμαίνεται από 2,8V έως 3,4V. Ο πομπός του απαιτεί 45mA ρεύμα και ο δέκτης του 50mA ρεύμα, αλλά το XBee σε κατάσταση ύπνου τραβάει μόλις 10 $\mu$ A.



Το XBee συνδέεται με το Arduino με χρήση της πλακέτας XBee Shield και έτσι τροφοδοτείται αυτόματα από το Arduino και δεν χρειάζεται εξωτερική πηγή ισχύος. Επίσης κάνει χρήση των ακροδεκτών 0 και 1 του Arduino με αποτέλεσμα να μπορεί να εκπέμπει ασύρματα τα δεδομένα που παράγει το Arduino για σειριακή επικοινωνία καθώς επίσης και να λαμβάνει τα αντίστοιχα δεδομένα. Κάνει δηλαδή μετατροπή της σειριακής επικοινωνίας σε ασύρματη, ενσωματώνοντας τα δεδομένα που λαμβάνονται και αποστέλλονται στην στοίβα του προτύπου IEEE 802.15.4 και ZigBee. Στην εικόνα 2.9 φαίνεται η λογική λειτουργίας του XBee module με έναν μικροελεγκτή και στην εικόνα 2.10 η διασυνδέσεις των ακροδεκτών του.



Εικόνα 2.9: Λογική λειτουργίας της μονάδας XBee module.



Εικόνα 2.10: Διασυνδέσεις των ακροδεκτών του module XBee.

**Πίνακας 5: Σήματα ακροδεκτών του module XBee.**

Pin #	Name	Direction	Description
1	VCC	-	Power supply
2	DOUT	Output	UART Data Out
3	DIN / CONFIG	Input	UART Data In
4	DO8*	Output	Digital Output 8
5	RESET	Input	Module Reset (reset pulse must be at least 200 ns)
6	PWM0 / RSSI	Output	PWM Output 0 / RX Signal Strength Indicator
7	PWM1	Output	PWM Output 1
8	[reserved]	-	Do not connect
9	DTR / SLEEP_RQ / DI8	Input	Pin Sleep Control Line or Digital Input 8
10	GND	-	Ground
11	AD4 / DIO4	Either	Analog Input 4 or Digital I/O 4
12	CTS / DIO7	Either	Clear-to-Send Flow Control or Digital I/O 7
13	ON / SLEEP	Output	Module Status Indicator
14	VREF	Input	Voltage Reference for A/D Inputs
15	Associate / AD5 / DIO5	Either	Associated Indicator, Analog Input 5 or Digital I/O 5
16	RTS / AD6 / DIO6	Either	Request-to-Send Flow Control, Analog Input 6 or Digital I/O 6
17	AD3 / DIO3	Either	Analog Input 3 or Digital I/O 3
18	AD2 / DIO2	Either	Analog Input 2 or Digital I/O 2
19	AD1 / DIO1	Either	Analog Input 1 or Digital I/O 1
20	AD0 / DIO0	Either	Analog Input 0 or Digital I/O 0



## ΚΕΦΑΛΑΙΟ 3- ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΣ ΠΛΑΤΦΟΡΜΑΣ ARDUINO

### 3.1 Γλώσσα Προγραμματισμού Arduino

Η γλώσσα προγραμματισμού για την πλακέτα Arduino βασίζεται στην τεχνολογία Wiring η οποία είναι ένα πλαίσιο προγραμματισμού ανοικτού κώδικα για μικροελεγκτές. Η διάλεκτος και οι εντολές της είναι παρόμοιες με την γλώσσα προγραμματισμού C. Παρακάτω ακολουθούν μερικές από τις βασικές αρχές για τον προγραμματισμό της πλακέτας Arduino.

### 3.2 Δομή

Η βασική δομή ενός προγράμματος σε Arduino είναι σχετικά απλή και αποτελείται από δύο τουλάχιστον μέρη. Αυτά τα δύο μέρη ή συναρτήσεις περικλείουν μπλοκ εντολών στο σώμα τους και είναι

```
void setup()
{
  statements;
}

void loop()
{
  statements;
}
```

Το μέρος του setup() είναι το προπαρασκευαστικό στάδιο ενώ το loop() είναι το μέρος της εκτέλεσης. Και τα δύο μέρη είναι απαιτούμενα για τη λειτουργία του προγράμματος.

Το μέρος του setup() πρέπει να δηλώνει τις μεταβλητές στην αρχή του προγράμματος και είναι η πρώτη συνάρτηση που εκτελείται στο πρόγραμμα. Αυτή η συνάρτηση εκτελείται μία φορά και χρησιμοποιείται για να θέσει την κατάσταση των ακροδεκτών στο arduino ή να αρχικοποιήσει θύρες επικοινωνίας (π.χ serial).

Το μέρος της συνάρτησης loop() ακολουθεί στη συνέχεια και περιλαμβάνει τον κώδικα που εκτελείται συνεχώς για την είσοδο σημάτων, την έξοδο κτλ. Αυτή η συνάρτηση είναι στον πυρήνα κάθε προγράμματος Arduino και κάνει την κυρίως δουλειά.

### 3.3 Συνάρτηση setup()

Η συνάρτηση setup() καλείται μια φορά στην αρχή του προγράμματος. Αρχικοποιεί τους ακροδέκτες του arduino ή ξεκινά τη σειριακή σύνδεση. Πρέπει να περιλαμβάνεται σε κάθε πρόγραμμα ακόμα κι αν δεν υπάρχουν εντολές για εκτέλεσή του.

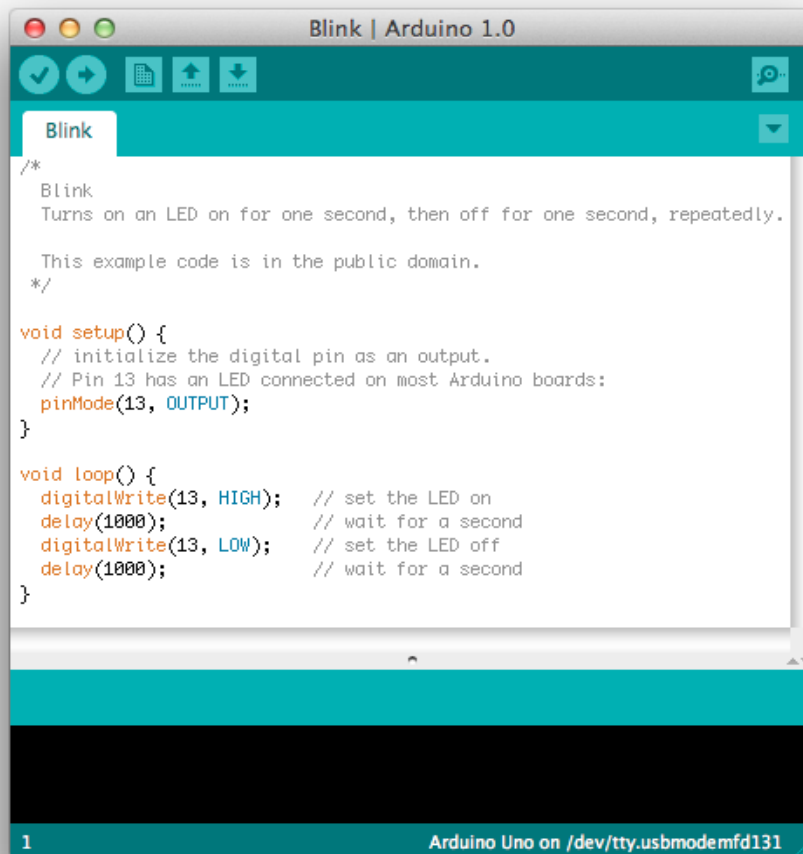
### 3.4 Συνάρτηση loop()

Η συνάρτηση loop() επαναλαμβάνεται συνεχώς εκτελώντας με βρόχο τις εντολές στο σώμα της. Συνήθως περιλαμβάνει εντολές για είσοδο και έξοδο σημάτων από και προς το arduino.

```
void loop()
{
  digitalWrite(pin, HIGH); // ενεργοποίηση ακροδέκτη
  delay(1000);             // παύση για ένα δευτερόλεπτο
  digitalWrite(pin, LOW); // απενεργοποίηση ακροδέκτη
  delay(1000);             // παύση για ένα δευτερόλεπτο
}
```

### 3.5 Σχεδιάγραμμα (Sketch)

Το σχεδιάγραμμα είναι το όνομα που χρησιμοποιεί το Arduino για το εκάστοτε πρόγραμμα. Είναι η μονάδα κώδικα που φορτώνεται και εκτελείται στην πλακέτα Arduino. Στην παρακάτω εικόνα φαίνεται το περιβάλλον εργασίας του Arduino IDE.



The image shows a screenshot of the Arduino IDE interface. The window title is "Blink | Arduino 1.0". The code editor displays the following code:

```
/*
  Blink
  Turns on an LED on for one second, then off for one second, repeatedly.

  This example code is in the public domain.
  */

void setup() {
  // initialize the digital pin as an output.
  // Pin 13 has an LED connected on most Arduino boards:
  pinMode(13, OUTPUT);
}

void loop() {
  digitalWrite(13, HIGH); // set the LED on
  delay(1000);            // wait for a second
  digitalWrite(13, LOW);  // set the LED off
  delay(1000);            // wait for a second
}
```

At the bottom of the IDE, the status bar shows "1" on the left and "Arduino Uno on /dev/tty.usbmodemfd131" on the right.

Εικόνα 3.1: Προγραμματιστικό περιβάλλον του Arduino IDE.

### 3.6 Σχόλια (Comments)

```

/*
 * Blink
 *
 * The basic Arduino example. Turns on an LED on for one second,
 * then off for one second, and so on... We use pin 13 because,
 * depending on your Arduino board, it has either a built-in LED
 * or a built-in resistor so that you need only an LED.
 *
 * http://www.arduino.cc/en/Tutorial/Blink
 */

```

Ότι βρίσκεται μεταξύ /\* και \*/ αγνοείται από το Arduino όταν τρέχει τον κώδικα που βρίσκεται στο σχεδιάγραμμα που του έχει φορτωθεί. Αυτό χρησιμοποιείται για τους ανθρώπους που διαβάζουν τον κώδικα μιας εφαρμογής για να μπορούν να καταλάβουν τι κάνει ο κώδικας.

### 3.7 Μεταβλητές (Variables)

Οι μεταβλητές είναι ένας χώρος που αποθηκεύονται τα δεδομένα. Έχουν όνομα, τύπο και τιμή. Για παράδειγμα στην παρακάτω γραμμή κώδικα παρατηρείται η δήλωση μεταβλητής με όνομα ledPin με τύπο int και τιμή 13. Χρησιμοποιείται για να δείξει σε ποιον ακροδέκτη του Arduino συνδέεται ένα led.

```
int ledPin = 13; // σύνδεση LED με τον ακροδέκτη 13
```

Με αυτόν τον τρόπο κάθε φορά που αναφέρεται το όνομα ledPin ανακαλείται και η τιμή που της έχει ανατεθεί.

### 3.8 Εύρος Μεταβλητών

Μια μεταβλητή μπορεί να δηλωθεί στην αρχή του προγράμματος πριν το void setup(), τοπικά μέσα σε συναρτήσεις και μερικές φορές μέσα σε μπλοκ κώδικα όπως το for. Η θέση όπου δηλώνεται μια μεταβλητή καθορίζει και το εύρος της χρήσης της, ή τη δυνατότητα συγκεκριμένων μερών του προγράμματος να χρησιμοποιήσουν τις μεταβλητές αυτές.

Μια global μεταβλητή δηλώνεται με σκοπό να μπορεί να προσπελαθεί και χρησιμοποιηθεί από κάθε συνάρτηση και εντολή παντού στο πρόγραμμα. Για να είναι μια μεταβλητή global πρέπει να δηλωθεί στην αρχή του προγράμματος πριν τη συνάρτηση setup().

Μια τοπική μεταβλητή ορίζεται μέσα σε μια συνάρτηση ή σε μπλοκ κώδικα. Είναι μόνο ορατή μέσα στη συνάρτηση επομένως είναι δυνατόν στο ίδιο πρόγραμμα να υπάρχουν δύο μεταβλητές με το ίδιο όνομα και διαφορετικό εύρος που να περιέχουν διαφορετικές τιμές. Η ανάθεση μεταβλητών σε κάθε συνάρτηση απλοποιεί το πρόγραμμα και μειώνει τα προγραμματιστικά λάθη.

```

int value;          // η «τιμή» είναι διαθέσιμη
                   //σε κάθε συνάρτηση

void setup()
{
  // no setup needed
}

void loop()
{
  for (int i=0; i<20;) // η “i” είναι διαθέσιμη μόνο στην for-loop
  {
    i++;
  }
  float f;
}

```

### 3.9 Είδη Μεταβλητών

Τα είδη μεταβλητών που χρησιμοποιούνται στον προγραμματισμό Arduino είναι οι κλασσικοί τύποι όπως το byte (8bit αριθμητική τιμή χωρίς δεκαδικά με εύρος 0-255), int (ακέραιοι αριθμητικοί χωρίς δεκαδικά ψηφία αποθηκεύονται ως 16bit αριθμοί με εύρος 32767 μέχρι -32768), long (επεκτεταμένο εύρος ακεραίων αποθηκεύονται ως 32bit με εύρος 2147483647 μέχρι -2147483648), float (για κινητής υποδιαστολής αριθμούς με δεκαδικό σημείο, αποθηκεύονται ως 32bit με εύρος 3.4E+38 μέχρι -3.4<sup>E</sup>+38).

Ένας τύπος μεταβλητής που μπορεί επίσης να χρησιμοποιηθεί σε προγραμματισμό Arduino είναι οι διατάξεις (arrays) που αποτελούν συλλογή δεδομένων που προσπελαύνονται με δείκτες. Κάθε τιμή της διάταξης προσπευλαύνεται με το όνομα της διάταξης και μια τιμή ενός δείκτη. Η πρώτη τιμή κάθε διάταξης αντιστοιχεί σε μηδενική τιμή δείκτη, επομένως το εύρος του δείκτη είναι από 0 μέχρι N-1, όπου N το πλήθος των στοιχείων της διάταξης.

Η δήλωση διάταξης και αρχικοποίηση γίνεται όπως ο παρακάτω κώδικας

```
int myArray[] = {value0, value1, value2...}
```

Ή εναλλακτικά

```
int myArray[5];          // array έξι θέσεων
myArray[3] = 10;        //τοποθετεί στον τέταρτο δείκτη την τιμή 10
```

Για την προσπέλαση τιμής μια διάταξης χρησιμοποιείται όπως παρακάτω:

```
x = myArray[3];        // το x είναι ίσο με 10
```

Οι διατάξεις χρησιμοποιούνται σε βρόχους for όπου χρησιμοποιείται ένας δείκτης αύξησης για κάθε διάταξη.

```
int ledPin = 10; //το LED είναι στον ακροδέκτη 10
byte flicker[] = { 180, 30, 255, 200, 10, 90, 150, 60};
//array με 8 διαφορετικές τιμές

void setup() //αποθηκεύει την έξοδο του ακροδέκτη
{
  pinMode(ledPin, OUTPUT);
}
void loop()
{
  for (int i=0; i<7; i++) //επανάληψη σε array 8 θέσεων

  {
    analogWrite(ledPin, flicker[i]); // εγγραφή στο δείκτη
    delay(200); // καθυστέρηση 200ms
  }
}
```

### 3.10 Τελεστές

Οι αριθμητικοί τελεστές επιτρέπουν την εκτέλεση πράξεων όπως πρόσθεση, αφαίρεση, πολλαπλασιασμός και διαίρεση. Η αριθμητική πράξη εκτελείται χρησιμοποιώντας τους τύπους δεδομένων των τελεστών, επομένως η πράξη  $9 / 4$  επιστρέφει ως αποτέλεσμα το 2 αντί του 2.25 αφού οι τελεστές 9 και 4 είναι ορισμένοι ως τύποι δεδομένων int. Αν οι τελεστές είναι διαφορετικού τύπου, χρησιμοποιείται για το αποτέλεσμα της αριθμητικής πράξης ο μεγαλύτερος τύπος. Πχ στο προηγούμενο παράδειγμα αν ο τελεστής 9 έχει οριστεί ως float τότε το αποτέλεσμα θα είναι 2.25.

Υπάρχει επιπλέον η δυνατότητα να χρησιμοποιείται η τεχνική cast που μετατρέπει ένα τύπο μεταβλητής σε άλλο σε πραγματικό χρόνο. Πχ  $I = (int)3.6$  επιστρέφει αποτέλεσμα στη μεταβλητή I ίσο με 3.

Οι σύνθετες αναθέσεις συνδυάζουν έναν αριθμητικό τελεστή με μια ανάθεση μεταβλητής. Οι πιο κοινές σύνθετες αναθέσεις είναι

```
x ++ // αυξάνει το x κατά +1
x - - // μειώνει το x κατά -1
x += y // αυξάνει το x με την τιμή +y
x -= y // μειώνει το x κατά -y
x *= y // πολλαπλασιάζει το x κατά y
x /= y // διαιρεί το x με το y
```

### 3.11 Συγκριτικοί τελεστές

Συγκριτικοί τελεστές χρησιμοποιούνται για τον έλεγχο μιας μεταβλητής ή σταθεράς σε βρόχους ελέγχου. Οι τελεστές σύγκρισης που χρησιμοποιούνται σε γλώσσα προγραμματισμού Arduino είναι

```
x == y    // το x ισούται με y
x != y    // το x είναι διάφορο του y
x < y     // το x είναι μικρότερο του y
x <= y    // το x είναι μικρότερο ή ίσο του y
x >= y    // το x είναι μεγαλύτερο ή ίσο του y
```

Οι λογικοί τελεστές είναι ένας συνήθης τρόπος για σύγκριση δύο εκφράσεων και επιστροφή αποτελέσματος TRUE ή FALSE.

Logical AND:

```
If (x > 0 && x < 5)    // αληθές μόνο εάν και οι δυο εκφράσεις είναι αληθείς
```

Logical OR:

```
If (x > 0 || y > 0)    // αληθές μόνο εάν μία έκφραση είναι αληθής
```

Logical NOT:

```
If (!x > 0)            // αληθές μόνο μια έκφραση είναι ψευδής
```

### 3.12 Σταθερές

Η γλώσσα προγραμματισμού Arduino έχει ορισμένες προκαθορισμένες τιμές που αποκαλούνται σταθερές και κάνουν πιο ευκολοανάγνωστα τα προγράμματα. Υπάρχουν οι λογικές σταθερές που καθορίζουν λογικά επίπεδα. Το FALSE ορίζεται ως το 0 και το TRUE ως το 1. Επιπλέον υπάρχουν σταθερές που ορίζουν τα επίπεδα σε ένα ακροδέκτη του Arduino ως HIGH και LOW και χρησιμοποιούνται όταν διαβάζεται ή γράφεται μια πληροφορία σε ψηφιακούς ακροδέκτες του Arduino. Το HIGH ορίζεται ως το λογικό επίπεδο 1, το ON ή ως τάση αντιστοιχεί στα 5volt ενώ το LOW είναι το λογικό επίπεδο 0, το OFF και αντιστοιχεί σε τάση 0 volts. Επιπλέον σταθερές που χρησιμοποιούνται με τη συνάρτηση pinMode() είναι η INPUT και OUTPUT.

### 3.13 Συναρτήσεις (Functions)

Μια συνάρτηση είναι ένα κομμάτι κώδικα που του έχει δοθεί ένα όνομα και μπορεί να χρησιμοποιηθεί από οπουδήποτε μέσα στο σχεδιάγραμμα. Για παράδειγμα ακολουθεί ο ορισμός της συνάρτησης `setup()`:

```
void setup()
{
  pinMode(ledPin, OUTPUT);    // αρχικοποίηση ακροδέκτη
}
```

Η πρώτη γραμμή προσφέρει πληροφορίες σχετικά με την συνάρτηση όπως το όνομα της. Οι λέξεις πριν και μετά το όνομα της συνάρτησης μας δείχνουν τον τύπο του αποτελέσματος της συνάρτησης και τις παραμέτρους που μπορεί να περνάμε σε αυτήν. Ο κώδικας μεταξύ των `{ }` καλείται σώμα της συνάρτησης και δείχνει τι κάνει αυτή η συνάρτηση. Υπάρχουν συναρτήσεις οι οποίες είναι έτοιμες και παρέχονται από την ομάδα σχεδιασμού του Arduino αλλά υπάρχει και η δυνατότητα ο χρήστης της γλώσσας να δημιουργεί τις δικές του συναρτήσεις ανάλογα με το τι θέλει να υλοποιήσει μέσα σε μια εφαρμογή.

### 3.14 Κοινές συναρτήσεις (Common Functions)

Μερικές χρήσιμες έτοιμες συναρτήσεις είναι οι ακόλουθες: `pinMode()`, `digitalWrite()`, και `delay()`. Η `PinMode()` ρυθμίζει αν ένας ακροδέκτης θα χρησιμοποιηθεί σαν είσοδος ή σαν έξοδος. Για να γίνει αυτό θα πρέπει η συνάρτηση να έχει ως παραμέτρους τον αριθμό του ακροδέκτη που θα χρησιμοποιηθεί καθώς επίσης και μια εκ των δύο σταθερών `INPUT` ή `OUTPUT`. Η συνάρτηση `digitalWrite()` δίνει σαν έξοδο σε έναν ακροδέκτη μια τιμή για παράδειγμα:

```
digitalWrite(ledPin, HIGH);
```

η οποία θέτει τον ακροδέκτη 13 του Arduino σε κατάσταση λογικού 1 οπότε αυτό θα παρέχει 5V.

Η συνάρτηση `delay()` προκαλεί στο Arduino μια καθυστέρηση για έναν συγκεκριμένο αριθμό `milliseconds` που έχει δηλωθεί σε αυτήν σαν παράμετρος.

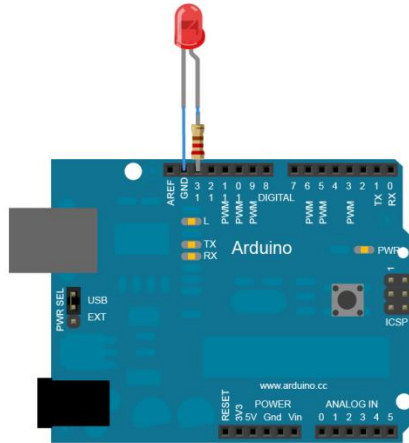
```
delay(1000);
```

Το παραπάνω παράδειγμα προκαλεί αναμονή ενός δευτερολέπτου.



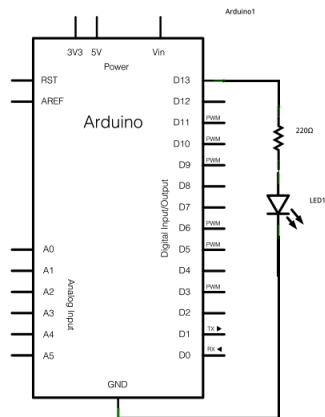
### 3.15 Παραδείγματα

Παρακάτω ακολουθεί ένα παράδειγμα σχεδιαγράμματος για το Arduino που περιλαμβάνει όλες τις παραπάνω βασικές έννοιες. Στο σχεδιάγραμμα αυτό υλοποιείται μια εφαρμογή που ανάβει και σβήνει ένα led κάθε ένα δευτερόλεπτο. Τα αντικείμενα που είναι απαραίτητα για την υλοποίηση του παραδείγματος είναι μια πλακέτα Arduino ένα led και μια αντίσταση.



Εικόνα 3.2: Ένα Arduino Uno συνδεδεμένο με ένα led

Το κυκλωματικό διάγραμμα της συνδεσμολογίας είναι το ακόλουθο:



Σχήμα 3.1: Το κυκλωματικό διάγραμμα της συνδεσμολογίας

Ο κώδικας που υλοποιεί την παραπάνω εφαρμογή είναι:

/\*

**Blink**

*Ενεργοποιεί το LED για ένα sec, έπειτα το απενεργοποιεί για ένα sec, επαναληπτικά.*

*Ο κώδικας του παραδείγματος είναι στον τομέα public.*

\*/

```

void setup() {
  //αρχικοποιούμε τον ακροδέκτη εξόδου
  //Ο ακροδέκτης 13 συνδέεται με το LED στην πλακέτα Arduino
  pinMode(13, OUTPUT);
}

void loop() {
  digitalWrite(13, HIGH);      // ενεργοποίηση LED
  delay(1000);
  digitalWrite(13, LOW);     // απενεργοποίηση του LED
  delay(1000);
}

```

Εδώ παρατηρείται ότι έχει ορισθεί ο ακροδέκτης 13 για να χρησιμοποιηθεί ως έξοδος του κυκλώματος. Αυτό συμβαίνει μέσα στην συνάρτηση setup() η οποία όπως αναφέρθηκε τρέχει μια φορά όταν ξεκινάει το Arduino. Παρακάτω ορίζεται η συνάρτηση loop() στην οποία παρατηρείται ανάθεση στον ακροδέκτη 13 της τάση των 5V. Με την χρήση της συνάρτησης delay(1000) διατηρείται η τάση στον ακροδέκτη 13 για ένα δευτερόλεπτο. Με την συνάρτηση digitalWrite(13, LOW) αφαιρείται από τον ακροδέκτη η τάση 5V για ένα επίσης δευτερόλεπτο. Η συνάρτηση loop() τρέχει ξανά και ξανά για όση ώρα είναι σε λειτουργία η πλακέτα του Arduino.

Η γλώσσα προγραμματισμού σε Arduino υποστηρίζει όλες τις λειτουργίες στον προγραμματισμό C όπως τα σημεία ελέγχου if, if...else τους βρόχους for, while και do ... while.

### 3.16 Συνάρτηση pinMode(pin,mode)

Η συνάρτηση pinMode χρησιμοποιείται στο μέρος του setup() ώστε να καθορίσει ένα ορισμένο ακροδέκτη να συμπεριφέρεται ως INPUT ή OUTPUT. Η προκαθορισμένη ρύθμιση είναι INPUT οπότε αν επιθυμούμε να συμπεριφέρεται έτσι δεν χρειάζεται να το καθορίσουμε ρητά. Όταν τίθεται σε κατάσταση INPUT τότε υποδηλώνεται ότι η είσοδος είναι σε κατάσταση υψηλής εμπέδησης. Επιπλέον υπάρχουν αντιστάσεις των 20KΩ στο Arduino που μπορούν να προσπελαστούν από το λογισμικό. Αυτές οι αντιστάσεις προσπελούνται με τον ακόλουθο τρόπο

```

pinMode(pin, INPUT); //τοποθέτηση του ακροδέκτη στην είσοδο
digitalWrite(pin, HIGH); // σύνδεση αντιστάσεων

```

Όταν το ακροδέκτη καθορίζεται ως OUTPUT τότε τίθεται σε κατάσταση χαμηλής εμπέδησης και παρέχει 40 mA ρεύμα σε άλλες συσκευές ή κυκλώματα που συνδέονται σε αυτό. Είναι αρκετό ρεύμα για να ενεργοποιήσει ένα LED αλλά όχι μεγάλο ρεύμα για να τρέξει ρελέ, κινητήρες, κτλ. Όταν ο ακροδέκτης δουλεύει σε κατάσταση OUTPUT τότε είναι καλή ιδέα να συνδέεται σε σειρά με αντιστάσεις 470Ω ή 1KΩ.

### 3.17 Συνάρτηση `digitalRead(pin)`

Διαβάζει την τιμή από ένα καθορισμένο ψηφιακό ακροδέκτη με το αποτέλεσμα να είναι είτε HIGH είτε LOW. Το ακροδέκτη καθορίζεται είτε ως μεταβλητή είτε ως σταθερά (0-13).

```
value = digitalRead(Pin); // η «τιμή» είναι ίση με τον ακροδέκτη εισόδου
```

### 3.18 Συνάρτηση `digitalWrite(pin, Value)`

Βγάζει στην έξοδο του ακροδέκτη λογική τιμή HIGH ή LOW (on/off). Το ακόλουθο παράδειγμα διαβάζει ένα pushbutton που είναι συνδεδεμένο με την ψηφιακή είσοδο και ανάβει ένα LED που συνδέεται με την ψηφιακή έξοδο όταν το κουμπί ενεργοποιείται.

```
int led = 13; // σύνδεση LED με ακροδέκτη 13
int pin = 7; // σύνδεση διακόπτη με τον ακροδέκτη 7
int value = 0; // μεταβλητή αποθήκευσης της τιμής read
void setup()
{
  pinMode(led, OUTPUT); // αρχικοποίηση ακροδέκτη 13 εξόδου
  pinMode(pin, INPUT); // αρχικοποίηση ακροδέκτη 7 εισόδου
}
void loop()
{
  value = digitalRead(pin); // αρχικοποίηση τιμής με τον ακροδέκτη εισόδου

  digitalWrite(led,value); //αρχικοποίηση του LED με την τιμή διακόπτη
}
```

### 3.19 Συνάρτηση `analogRead(pin)`

Διαβάζει την τιμή από ένα καθορισμένο αναλογικό ακροδέκτη με ανάλυση 10bit. Αυτή η συνάρτηση δουλεύει μόνο για τα αναλογικά ακροδέκτη (0-5). Το αποτέλεσμα της ανάγνωσης είναι ένας αριθμός από 0 – 1023.

```
Value = analogRead(pin); // αρχικοποίηση τιμής με αυτή του ακροδέκτη
```

### 3.20 Συνάρτηση `analogWrite(pin, Value)`

Γράφει μια ψευδο-αναλογική τιμή χρησιμοποιώντας pulse width modulation (PWM) σε ένα ακροδέκτη εξόδου. Στο Arduino με ATmega168 αυτή η συνάρτηση δουλεύει στα ακροδέκτη 3,5,6,9,10,11. Μια τιμή 0 παράγει μια σταθερή τιμή 0 volts στο καθορισμένο ακροδέκτη, μια

τιμή 255 παράγει μια σταθερή τιμή 5 volt στο καθορισμένο ακροδέκτη και για μια τιμή ενδιάμεση στο 0-255 η έξοδος αλλάζει γρήγορα από τα 0-5volts. Μια τιμή ίση με 64 η έξοδος θα είναι στα  $\frac{3}{4}$  του χρόνου ίση με 0 και στο  $\frac{1}{4}$  ίση με 5volt. Για μια τιμή 128 στο  $\frac{1}{2}$  χρόνο θα είναι 0 και στον άλλο  $\frac{1}{2}$  θα είναι 5.

Το ακόλουθο παράδειγμα διαβάζει μια αναλογική τιμή από το αναλογικό ακροδέκτη εισόδου, μετατρέπει την τιμή διαιρώντας την με 4 και βγάζει ως έξοδο ένα σήμα PWM στο PWM pin.

```
int led = 10;           //LED που συνδέεται με τον ακροδέκτη 10
int pin = 0;           //ποτενσιόμετρο που συνδέεται με τον αναλογικό ακροδέκτη 0
int value;             // τιμή για ανάγνωση

void setup() { }      // χωρίς αρχικοποίηση

void loop()
{
  value = analogRead(pin); // αρχικοποίηση «τιμή» με αυτή του ακροδέκτη
  value /= 4;              // διαιρεί τις τιμές 0-1023
  analogWrite(led, value); //βγάζει έξοδο το σήμα PWM στο LED
}
```

### 3.21 Συνάρτηση delay(ms)

Προκαλεί παύση της εκτέλεσης του προγράμματος για χρόνο ίσο με το όρισμα της συνάρτησης σε msec

```
delay(1000); // αναμονή 1 sec
```

### 3.22 Συνάρτηση millis()

Επιστρέφει τον αριθμό των msec από τη στιγμή που το Arduino άρχισε την εκτέλεση του προγράμματος. Θα γίνει overflow και θα μηδενιστεί ο αριθμός σε διάστημα περίπου 9 ώρες μετά την εκτέλεση.

```
value = millis ();
```

### 3.23 Συναρτήσεις min(x,y) max(x,y)

Είναι συναρτήσεις που επιστρέφουν το μέγιστο και τον ελάχιστο ανάμεσα σε δύο τιμές

```
value = min(value, 100); //επιστρέφει την τιμή 100 ή μικρότερο
```

### 3.24 Συνάρτηση randomSeed(seed)

Θέτει μια τιμή ως το αρχικό σημείο για τη συνάρτηση random(). Συνήθως ως όρισμα χρησιμοποιείται η έξοδος της συνάρτησης millis() ή ακόμα και η analogRead().

### 3.25 Συναρτήσεις random(max), random(min,max)

Η συνάρτηση random επιτρέπει την παραγωγή ψευδο-τυχαίων αριθμών μέσα σε ένα καθορισμένο εύρος ανάμεσα στις τιμές min, max. Χρησιμοποιείται μετά την κλήση της συνάρτησης randomSeed(seed).

Το ακόλουθο παράδειγμα δημιουργεί μια τυχαία τιμή μεταξύ 0-255 και βγάζει ως έξοδο ένα PWM σήμα ίσο με την τυχαία τιμή.

```
int randomNumber; //μεταβλητή που θα αποθηκεύσει την τυχαία τιμή

int led = 10; // LED που συνδέεται στον ακροδέκτη 10

void setup () { }

void loop ()
{
  randomSeed(millis()); //δημιουργεί τυχαίες τιμές
  randomNumber = random(255); //στο εύρος 0..255
  analogWrite(led, randomNumber); // βγάζει έξοδο ένα σήμα PWM με μια τυχαία τιμή
  delay(500);
}
```

### 3.26 Συνάρτηση Serial.begin(rate)

Ανοίγει μια σειριακή θύρα και θέτει το ρυθμό στην σειριακή μετάδοση δεδομένων. Ο τυπικός ρυθμός για επικοινωνία είναι 9600. Με τη χρήση της συνάρτησης Serial.begin(rate) δεν μπορούν να χρησιμοποιηθούν ταυτόχρονα τους ψηφιακούς ακροδέκτες 0 (RX) και 1(TX).

```
void setup()
{
  Serial.begin(9600); // ανοίγει την σειριακή θύρα στα 9600bps
}
```

### 3.27 Συνάρτηση `Serial.println(data)`

Τυπώνει δεδομένα στη σειριακή θύρα που ολοκληρώνονται με τις ειδικές λειτουργίες του carriage return και line feed. Είναι η πιο κατάλληλη συνάρτηση για διάβασμα δεδομένων στο Serial Monitor για έλεγχο. Το ακόλουθο παράδειγμα κάνει μια ανάγνωση από το αναλογικό ακροδέκτη 0 και στέλνει τα δεδομένα στον υπολογιστή κάθε 1 δευτερόλεπτο.

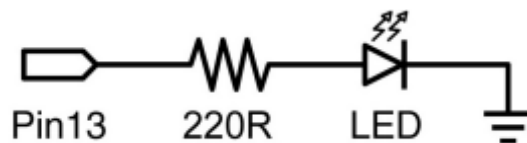
```
void setup()  
{  
  Serial.begin(9600);           // άνοιγμα σειριακής θύρας  
}  
  
void loop()  
{  
  Serial.println(analogRead(0) ); //τυπώνει τα δεδομένα  
  delay(1000);  
}
```

## ΚΕΦΑΛΑΙΟ 4-ΕΦΑΡΜΟΓΕΣ ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΥ ΣΕ ΠΛΑΤΦΟΡΜΑ ARDUINO

Στο κεφάλαιο αυτό παρουσιάζεται ο σχεδιασμός και η υλοποίηση εφαρμογών σε πλατφόρμα Arduino με τις κατάλληλες ηλεκτρονικές διατάξεις και τον προγραμματισμό της πλατφόρμας.

### 4.1 Εφαρμογή 1 – Ενεργοποίηση LED.

Το κυκλωματικό διάγραμμα για την υλοποίηση ενός ηλεκτρονικού κυκλώματος που θα ενεργοποιεί ένα LED ανά συγκεκριμένα χρονικά διαστήματα παρουσιάζεται παρακάτω



Σχήμα 4.1: Κυκλωματικό διάγραμμα ενεργοποίησης LED.

Το LED συνδέεται μέσω αντίστασης 220Ω στο ακροδέκτη 13 και αναβοσβήνει ανά ένα δευτερόλεπτο. Ο κώδικας σε Arduino είναι

```
int ledPin = 13;           //το LED συνδέεται με τον ακροδέκτη 13

void setup( )             //τρέχει μια φορά
{
  pinMode(ledPin, OUTPUT); //αρχικοποιεί τον ακροδέκτη 13 ως έξοδο
}

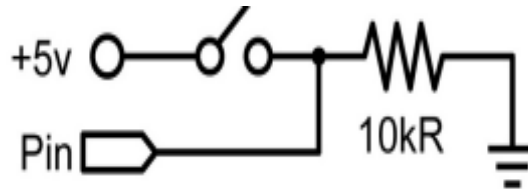
void loop( )
{
  digitalWrite(ledPin, HIGH); //ενεργοποίηση του LED
  delay(1000);                //καθυστέρηση 1sec
  digitalWrite(ledPin, LOW);  //απενεργοποίηση του LED
  delay(1000);                //καθυστέρηση 1sec
}
```

Κώδικας 1 Ενεργοποίηση LED.



## 4.2 Εφαρμογή 2 – Ψηφιακή Είσοδος σε Arduino.

Στην εφαρμογή αυτή δοκιμάζεται η ψηφιακή είσοδος στο Arduino που μπορεί να πάρει δύο πιθανές τιμές (ON/OFF). Ο ακροδέκτης ψηφιακής εισόδου 2 διαβάζει την είσοδο μέσω ενός διακόπτη ή pushbutton και εφόσον ενεργοποιηθεί τότε βγάζει την έξοδο σε ένα LED που ανάβει.



Σχήμα 4.2: Κυκλωματικό διάγραμμα εφαρμογής 2.

Και ο κώδικας στην πλατφόρμα Arduino δίνεται παρακάτω:

```
int ledPin = 13;           // ο ακροδέκτης εξόδου στο LED
int inPin = 2;            // ο ακροδέκτης εισόδου στον διακόπτη

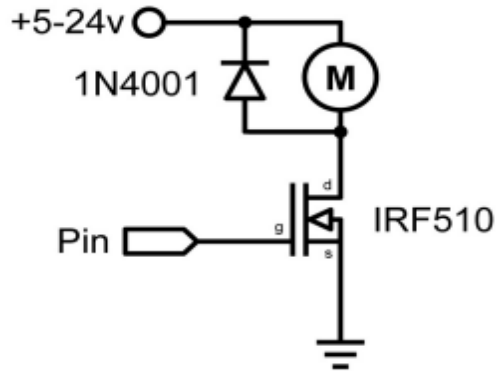
void setup()
{
  pinMode(ledPin, OUTPUT); // δήλωση του LED για έξοδο
  pinMode(inPin, INPUT);  // ορισμός του διακόπτη για είσοδο
}

void loop()
{
  if (digitalRead(inPin) == HIGH) // ελέγχει αν η είσοδος είναι ενεργή
  {
    digitalWrite(ledPin, HIGH); // ενεργοποίηση του LED
    delay(1000);
    digitalWrite(ledPin, LOW);  // απενεργοποίηση του LED
    delay(1000);
  }
}
```

Κώδικας εφαρμογής 2.

### 4.3 Εφαρμογή 3 – Χειρισμός μεγάλων ρευμάτων εξόδου.

Όπως έχει αναφερθεί και στο κεφάλαιο 3 το ρεύμα εξόδου από το Arduino είναι 40mA μέγιστο. Συχνά όμως είναι σημαντικό να χειρίζεται το Arduino μεγαλύτερες τιμές ρευμάτων σε φορτία. Για αυτό το λόγο στα κυκλώματα χρησιμοποιείται ένα τρανζίστορ. Στην ακόλουθη εφαρμογή το Arduino ανοιγοκλείνει γρήγορα ένα MOSFET 5 φορές το δευτερόλεπτο. Στο κύκλωμα υπάρχει κι ένας κινητήρας για να επιδεικνύεται η συμπεριφορά που έχει περιγραφεί.



Σχήμα 4.3: Κυκλωματικό διάγραμμα εφαρμογής 3.

Ο κώδικας που υλοποιεί την εφαρμογή είναι:

```
int outPin = 5;           //σύνδεση ακροδέκτη με το τρανζίστορ

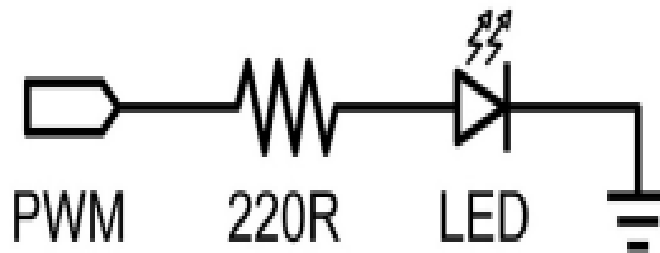
void setup ( )
{
  pinMode(outPin, OUTPUT); // ο ακροδέκτης 5 δηλώνεται ως έξοδο
}

void loop( )
{
  for (int i=0; i<=5; i++) //επανάληψη 5 φορές
  {
    digitalWrite(outPin, HIGH); //ενεργοποίηση τρανζίστορ
    delay(250);                 //καθυστέρηση ¼ sec
    digitalWrite(outPin, LOW);  //απενεργοποίηση τρανζίστορ
    delay(250);                 //καθυστέρηση ¼ sec
  }
  delay(1000);                 //καθυστέρηση 1 sec
}
```

Κώδικας εφαρμογής 3.

## 4.4 Εφαρμογή 4 – Έξοδος PWM

Η διαμόρφωση PWM αποτελεί τον τρόπο που παρέχει η πλατφόρμα Arduino για να παράγει ένα ψευδο-αναλογικό σήμα εξόδου. Με αυτό τον τρόπο μπορούμε να ελέγχουμε τη φωτεινότητα ενός LED ή να ελέγχουμε ένα σερβοκινητήρα συνδεδεμένο στο κυκλωματικό διάγραμμα της εξόδου.



Σχήμα 4.4: Κυκλωματικό διάγραμμα εφαρμογής 4.

```

int ledPin = 9;           //ο ακροδέκτης PWM στο LED
void setup() { }

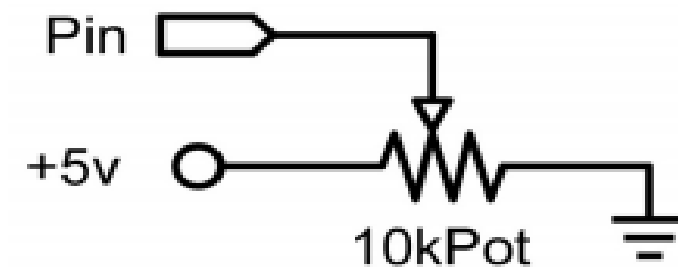
void loop()
{
  for (int i=0; i<=255; i++) // αύξηση ψευδο-αναλογικού σήματος
  {
    analogWrite(ledPin, i); //ενεργοποίηση φωτεινότητας κατά i
    delay(100);             //καθυστέρηση 100ms
  }
  for (int i=255; i>=0; i--) //ελάττωση ψευδο-αναλογικού σήματος
  {
    analogWrite(ledPin, i); // ενεργοποίηση φωτεινότητας κατά i
    delay(100);             //καθυστέρηση 100ms
  }
}

```

Κώδικας εφαρμογής 4.

## 4.5 Εφαρμογή 5 – Είσοδος από ποτενσιόμετρο.

Με τη χρήση ενός ποτενσιόμετρου και την είσοδο αναλογικού σε ψηφιακό σήμα του Arduino μπορούμε να διαβάζουμε την αναλογική τιμή από το ποτενσιόμετρο και να τη μετατρέπουμε σε ψηφιακή 0-1023.



Σχήμα 4.5: Κυκλωματικό διάγραμμα εφαρμογής 5.

```

int potPin = 0;           //ο ακροδέκτης εισόδου στο ποτενσιόμετρο
int ledPin = 13;        //ο ακροδέκτης εξόδου στο LED

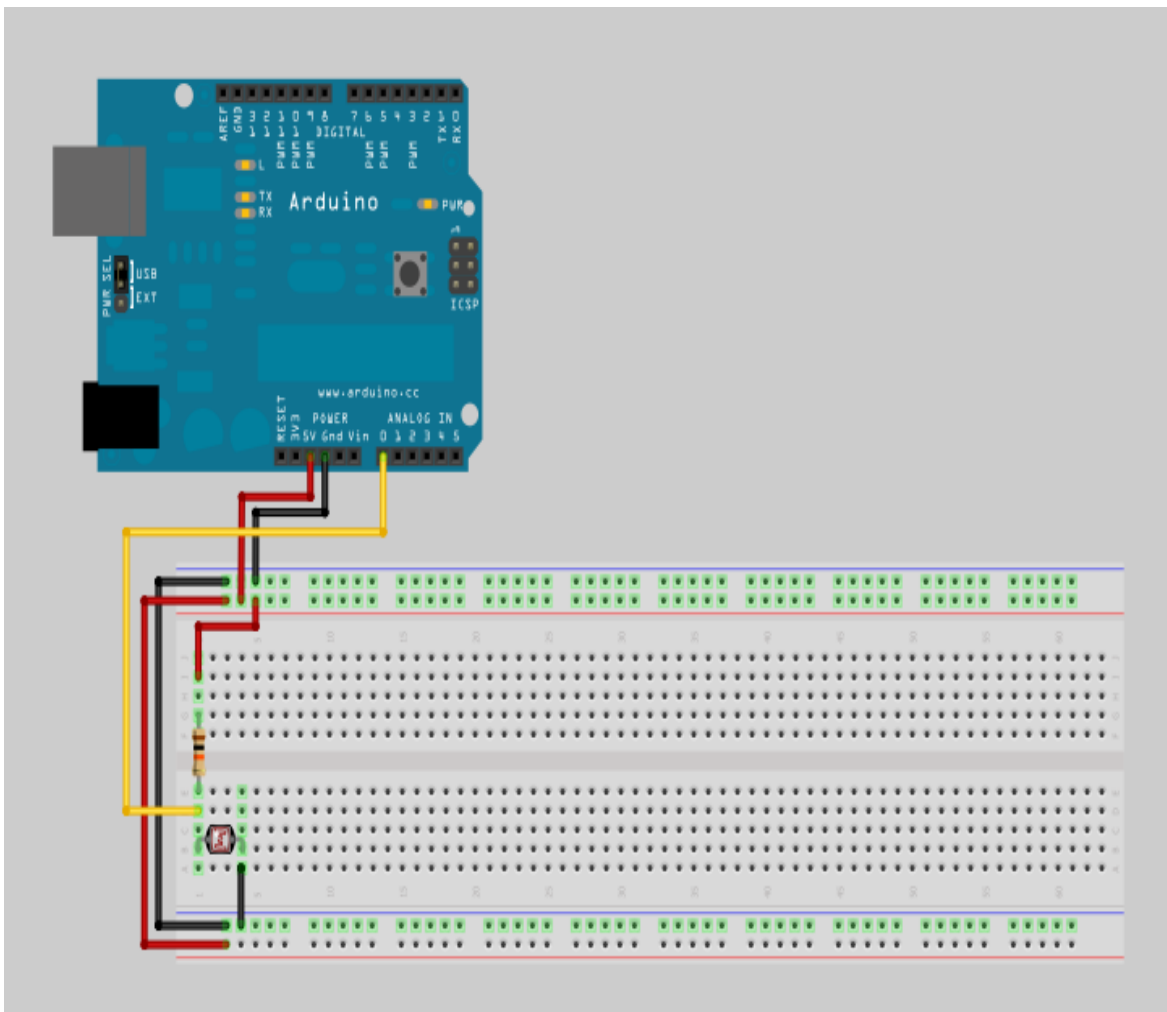
void setup()
{
  pinMode(ledPin, OUTPUT); //δήλωση του ακροδέκτη ledPin ως έξοδο
}
void loop()
{
  digitalWrite(ledPin, HIGH); //ενεργοποίηση του ledPin
  delay(analogRead(potPin)); //παύση προγράμματος
  digitalWrite(ledPin, LOW); //απενεργοποίηση ledPin
  delay(analogRead(potPin)); //διακοπή προγράμματος
}

```

Κώδικας εφαρμογής 5.

## 4.6 Εφαρμογή 6 - Σύνδεση Φωτοαντίστασης LDR.

Στόχος της εφαρμογής 6 είναι η εξοικείωση με την σύνδεση μιας φωτοαντίστασης LDR για τη μέτρηση των επιπέδων φωτεινότητας του περιβάλλοντος χώρου. Χρησιμοποιώντας την αντίσταση  $R=10\text{ K}\Omega$  να φτιαχτεί με τέτοιο τρόπο ο διαιρέτης τάσης ώστε όταν μειώνεται η ποσότητα φως που προσπίπτει στην επιφάνεια της φωτοαντίστασης LDR τότε να αυξάνεται η τάση εξόδου που θα συνδεεται στον αναλογικο ακροδεκτη 0 του arduino. Το αποτέλεσμα της μέτρησης θα είναι ο αριθμός από 0 μέχρι 1023 και να εκτυπώνεται στη σειριακή θύρα του Arduino ανά sec. Η σειριακή θύρα να ρυθμιστεί σε ταχύτητα 57600 bps.



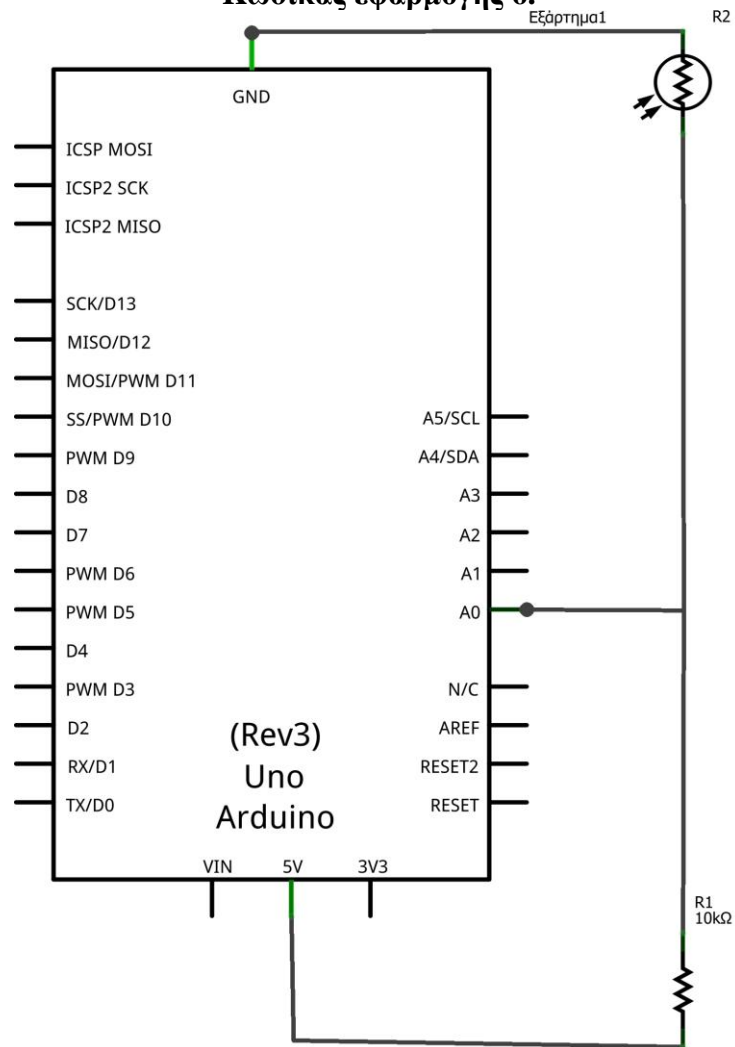
Εικόνα 4.1: Σχηματικό διάγραμμα εφαρμογής 6.

```

int LDR_sensor = 0; // αρχικοποίηση μεταβλητής LDR
void setup()
{
  Serial.begin(57600); // αρχικοποίηση σειριακής θύρας
}
void loop()
{
  LDR_sensor = analogRead(0); // διάβασμα τάσης εισόδου
  Serial.print("Epipedo fwtnothtas: "); // εκτύπωση στην σειριακή θύρα
  Serial.println(LDR_sensor); // εκτύπωση στην σειριακή θύρα
  delay(1000); // καθυστέρηση
}

```

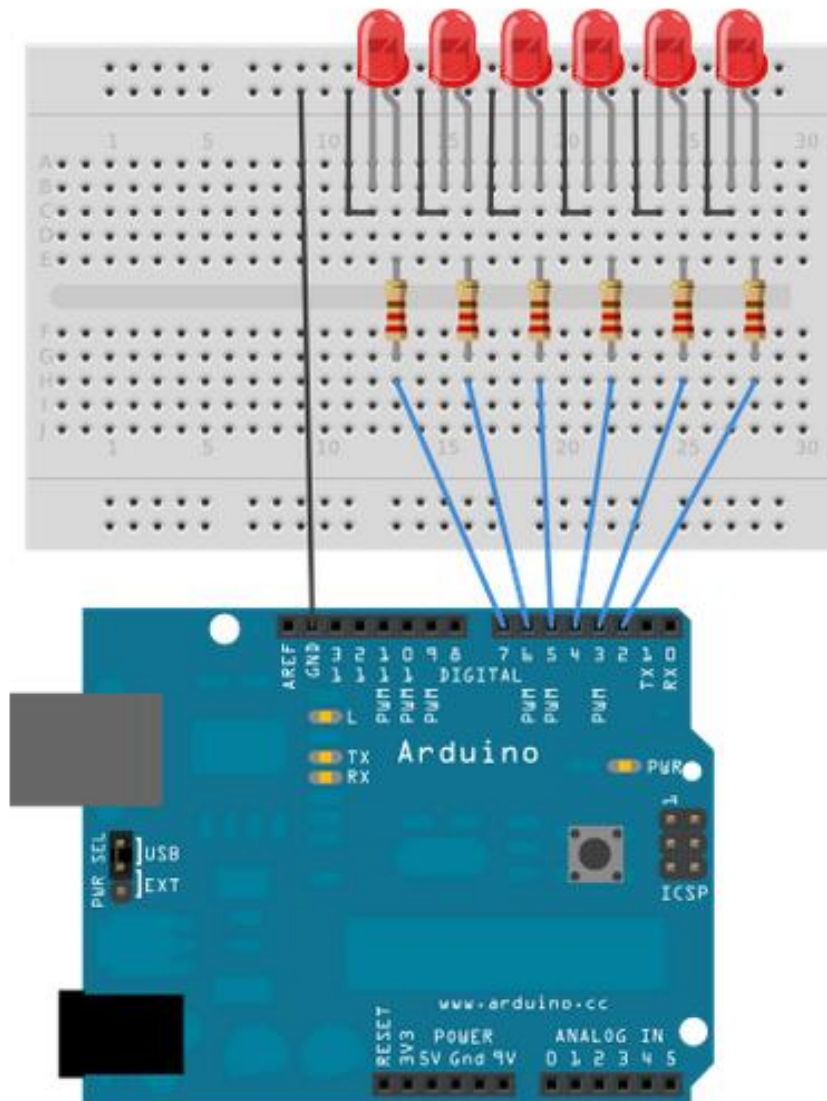
Κώδικας εφαρμογής 6.



Σχήμα 4.6: Κυκλωματικό διάγραμμα εφαρμογής 6.

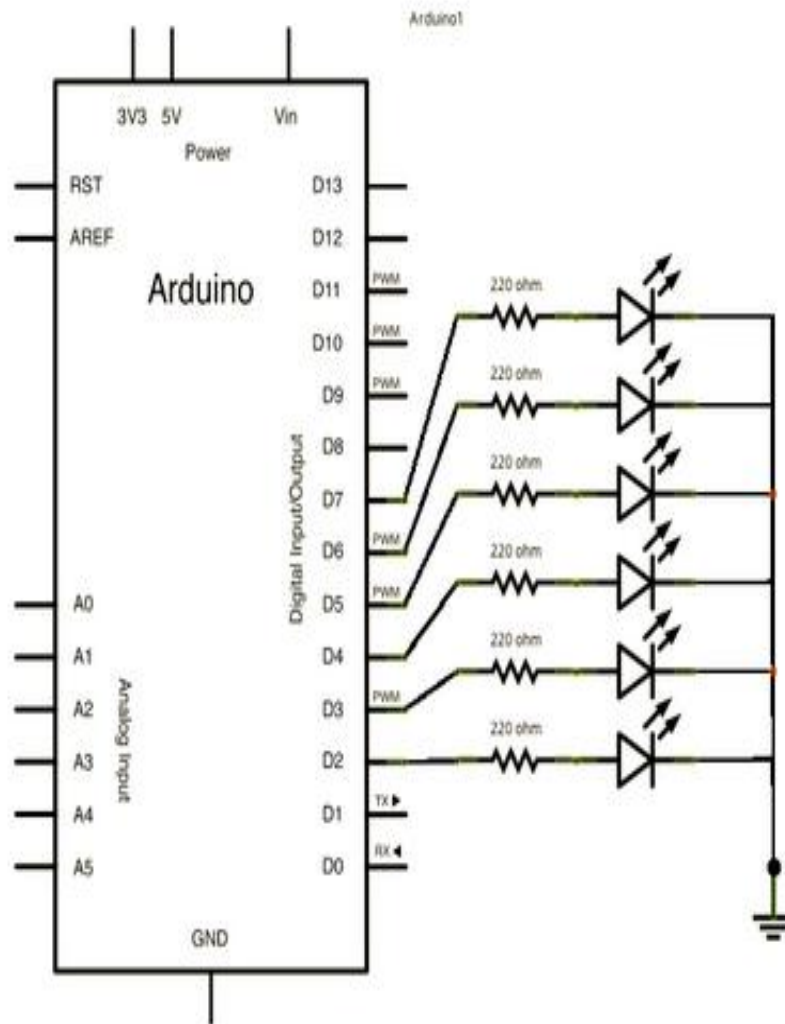
## 4.7 Εφαρμογή 7 – Έλεγχος διάταξης LEDs.

Η έβδομη εφαρμογή αφορά την ενεργοποίηση μιας σειράς LEDs που έχουν συνδεθεί στα ακροδέκτη 2 έως 7 του Arduino και τα οποία είναι αριθμημένα με συνεχόμενο τρόπο ενώ τα LEDs ανάβουν ακολουθιακά. Κάθε LED είναι σε σειρά με αντίσταση 220Ω



Εικόνα 4.2: Σχηματικό διάγραμμα εφαρμογής 7.





Σχήμα 4.7: Κυκλωματικό διάγραμμα εφαρμογής 7.

```

int timer = 100;           //Όσο μεγαλύτερη η τιμή τόσο πιο αργό το χρονόμετρο
int ledPins[ ] = { //array με αριθμούς ακροδεκτών στους οποίους τα LED's είναι
    2,7,4,6,5,3 };           //συνδεδεμένα
int pinCount = 6;

void setup() {
    //τα στοιχεία array είναι αριθμημένα από το 0 στο (pinCount - 1)
    //χρησιμοποίηση ενός βρόχου για να αρχικοποιήσουμε κάθε ακροδέκτη ως
    έξοδο
    for (int thisPin = 0; thisPin < pinCount; thisPin++) {
        pinMode(ledPins[thisPin], OUTPUT);
    }
}

void loop() {
    //loop από το μικρότερο ακροδέκτη στο μεγαλύτερο
    for (int thisPin = 0; thisPin < pinCount; thisPin++) {
        //ενεργοποίηση ακροδέκτη
        digitalWrite(ledPins[thisPin], HIGH);
        delay(timer);
        //απενεργοποίηση ακροδέκτη
        digitalWrite(ledPins[thisPin], LOW);

    }

    //loop από το μεγαλύτερο στο μικρότερο
    for (int thisPin = pinCount - 1; thisPin >= 0; thisPin--) {
        //ενεργοποίηση ακροδέκτη
        digitalWrite(ledPins[thisPin], HIGH);
        delay(timer);
        //απενεργοποίηση ακροδέκτη
        digitalWrite(ledPins[thisPin], LOW);
    }
}

```

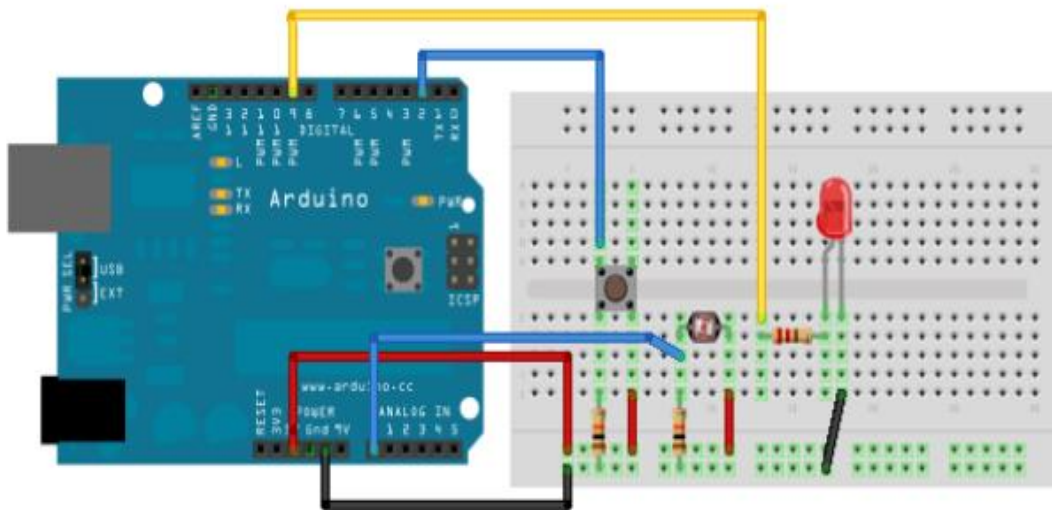
Κώδικας εφαρμογής 7.

## 4.8 Εφαρμογή 8 – Βρόχος Έλεγχου.

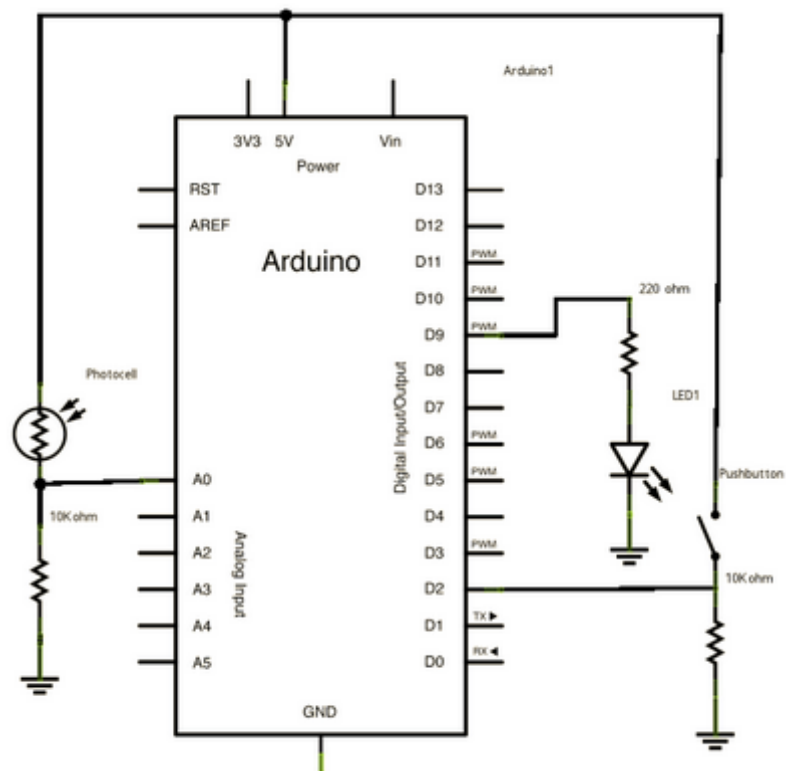
Με την εφαρμογή 8 δημιουργείται ένα πρόγραμμα ελέγχου ενός αισθητήρα συνδεδεμένου στην αναλογική είσοδο του Arduino που ανάλογα με το αν πατιέται ένα κουμπί το πρόγραμμα κάνει βαθμονόμηση της τιμής του αισθητήρα.

Στο κύριο μέρος του προγράμματος διαβάζεται μια τιμή από ένα αισθητήρα (φωτοαντίσταση) στο αναλογικό ακροδέκτη 0 και η τιμή αυτή χρησιμοποιείται για να οδηγείται το LED στον ακροδέκτη 9. Το πρόγραμμα συνεχίζει να κάνει αυτή τη λειτουργία μέχρι να πατηθεί ένα κουμπί συνδεδεμένο στο ψηφιακό ακροδέκτη 2 οπότε καλείται η συνάρτηση `calibrate()` που υλοποιείται μέσα στο πρόγραμμα, που ψάχνει για την μέγιστη και την ελάχιστη τιμή του αναλογικού αισθητήρα. Όταν το κουμπί αφήνεται, τότε το πρόγραμμα συνεχίζει την προηγούμενη λειτουργία.

Με αυτόν τον τρόπο μπορούμε να ανανεώνουμε τη μέγιστη και ελάχιστη τιμή της φωτοαντίστασης παρακολουθώντας έτσι τις συνθήκες φωτισμού.



Εικόνα 4.3: Σχηματικό διάγραμμα εφαρμογής 8.



Σχήμα 4.8: Κυκλωματικό διάγραμμα εφαρμογής 8.

//Αυτές οι σταθερές δεν θα αλλάξουν

```
const int sensorPin = A2;           //ακροδέκτης πάνω στον αισθητήρα που είναι συνδεδεμένος
const int ledPin = 9;               //ακροδέκτης που είναι συνδεδεμένος στο LED
const int indicatorLedPin = 13;     // ακροδέκτης στο συνδεδεμένος LED
const int buttonPin = 2;           //ακροδέκτης στον διακόπτη
```

```
int sensorMin = 1023;              //η μικρότερη τιμή του αισθητήρα
int sensorMax = 0;                 //η μεγαλύτερη τιμή του αισθητήρα
int sensorValue = 0;               //η τιμή του αισθητήρα
```

```
void setup() {
```

//τοποθετούμε τους ακροδέκτες του LED ως έξοδο και τους ακροδέκτες του διακόπτη ως  
//είσοδο

```
pinMode(indicatorLedPin, OUTPUT);
pinMode(ledPin, OUTPUT);
pinMode(buttonPin, INPUT);
}
```

```

void loop() {
  //ενώ ο διακόπτης είναι πατημένος ξεκίνησε την συνάρτηση calibrate()
  while (digitalRead(buttonPin) == HIGH) {
    calibrate();
  }
  // δίνει το τέλος της συνάρτησης callibrate
  digitalWrite(indicatorLedPin , LOW);
  //ανάγνωση αισθητήρα
  sensorValue = analogRead(sensorPin);

  //εφάρμοσε το καλιμπράρισμα στον αισθητήρα ανάγνωσης
  sensorValue = map(sensorValue, sensorMin, sensorMax, 0, 255);

  //σε περίπτωση που η τιμή του αισθητήρα είναι έξω από το εύρος που έχουμε δει στο
  //καλιμπράρισμα
  sensorValue = constrain(sensorValue, 0, 255);
  //μειώνουμε την τιμή του LED χρησιμοποιώντας την τιμή του καλιμπραρίσματος
  analogWrite(ledPin, sensorValue);
}

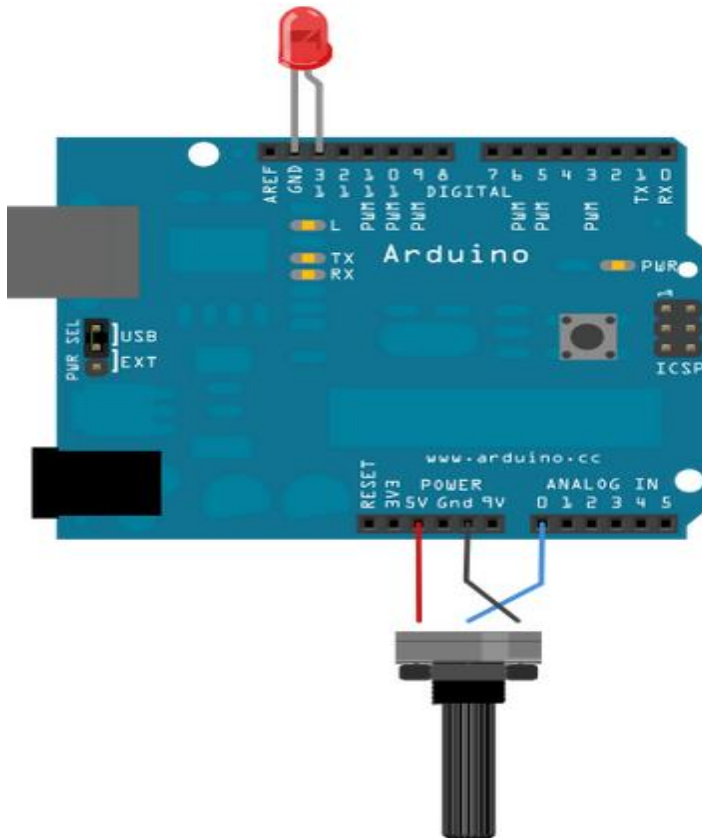
void calibrate() {
  //ενεργοποίηση του LED ώστε να δείχνει ότι το καλιμπράρισμα έχει ξεκινήσει
  digitalWrite(indicatorLedPin, HIGH);
  sensorValue = analogRead(sensorPin);
  // καταχώρησε την μέγιστη τιμή του αισθητήρα
  if (sensorValue > sensorMax) {
    sensorMax = sensorValue;
  }
  //καταχώρησε την μικρότερη τιμή
  if (sensorValue < sensorMin) {
    sensorMin = sensorValue;
  }
}

```

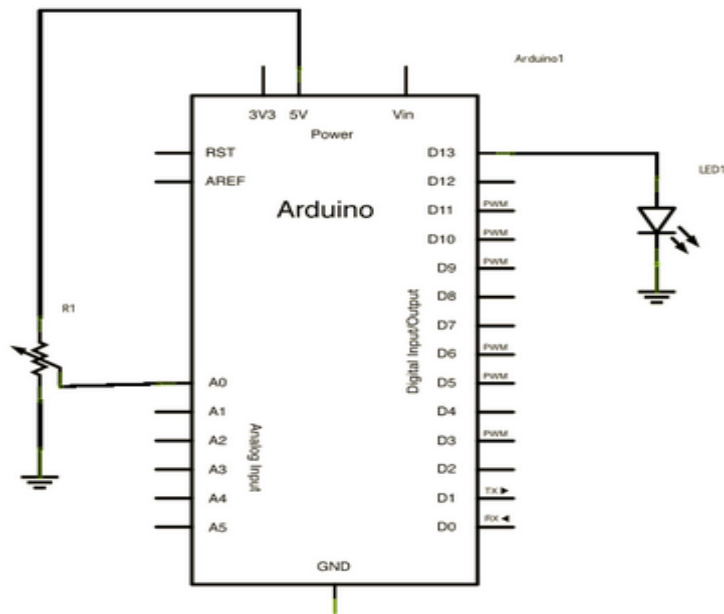
Κώδικας εφαρμογής 8.

## 4.9 Εφαρμογή 9 – Έλεγχος If.

Ο έλεγχος συνθηκών με το if() αποτελεί σημαντικό στοιχείο του προγραμματισμού Arduino. Στην εφαρμογή που παρατίθεται ένα LED που συνδέεται στον ακροδέκτη 13 ανάβει όταν η τιμή που διαβάζεται στην αναλογική είσοδο ξεπερνά ένα ορισμένο κατώφλι.



Εικόνα 4.4: Σχηματικό διάγραμμα εφαρμογής 9.



Σχήμα 4.9: Κυκλωματικό διάγραμμα εφαρμογής 9.

Ένα ποτενσιόμετρο συνδέεται στην αναλογική είσοδο και η τιμή που διαβάζεται από αυτό αποθηκεύεται στη μεταβλητή analogValue. Αυτή η μεταβλητή συγκρίνεται με ένα κατώφλι. Αν η τιμή είναι μεγαλύτερη του κατωφλίου τότε το LED ανάβει αλλιώς παραμένει σβηστό.

```

const int analogPin = A0; //ακροδέκτης που ο αισθητήρας είναι συνδεδεμένος
const int ledPin = 13;    //ακροδέκτης που το LED είναι συνδεδεμένο
const int threshold = 400; //τυχαίο επίπεδο που βρίσκεται το εύρος της
αναλογικής εισόδου

void setup() {
//ενεργοποίηση του LED ως είσοδος
pinMode(ledPin, OUTPUT);
//ενεργοποίηση της σειριακής επικοινωνίας
Serial.begin(9600);
}

void loop() {
//διάβασε την τιμή του ποτενσιόμετρου
int analogValue = analogRead(analogPin);

//αν η αναλογική τιμή είναι αρκετά υψηλή ενεργοποίησε το LED
if (analogValue > threshold) {
digitalWrite(ledPin, HIGH);
}
else
{
digitalWrite(ledPin, LOW);
}

//τύπωσε την αναλογική τιμή
Serial.println(analogValue);
delay(1); //καθυστέρηση
}

```

**Κώδικας εφαρμογής 9.**

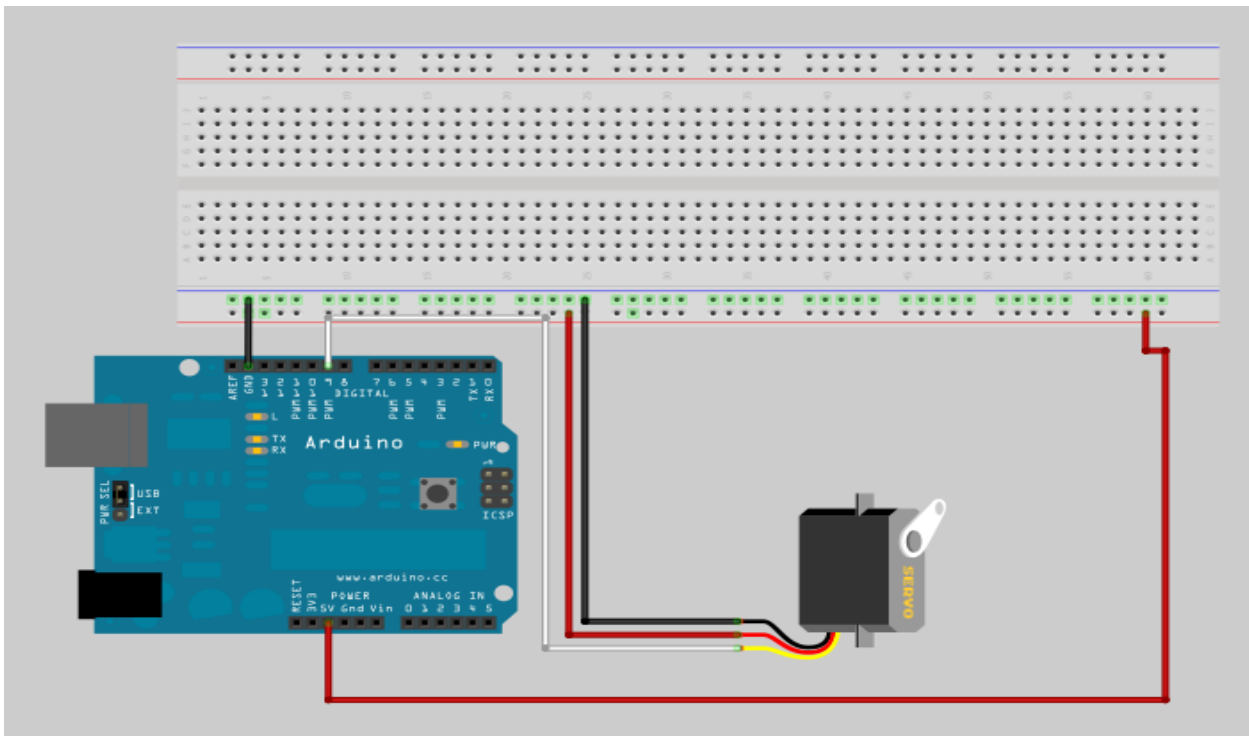


## 4.10 Εφαρμογή 10 - Έλεγχος Σερβοκινητήρα.

Οι σερβοκινητήρες έχουν τρία καλώδια, ισχύος, γείωσης και σήματος. Το καλώδιο ισχύος είναι το κόκκινο και συνδέεται με το ακροδέκτη 5V του Arduino. Το καλώδιο γείωσης είναι το μαύρο ή καφέ και συνδέεται με το ακροδέκτη γείωσης του Arduino. Το καλώδιο σήματος είναι τυπικά το κίτρινο ή πορτοκαλί και συνδέεται με το ακροδέκτη 9.

Στόχος της εφαρμογής 10 είναι η εξοικείωση με τη σύνδεση και χειρισμό ενός σερβοκινητήρα ο οποίος μπορεί να περιστρέψει τον άξονά του κατά 180 μοίρες. Για το χειρισμό του θα γίνει χρήση της ενσωματωμένης βιβλιοθήκης του Arduino, Servo.

Ο κινητήρας στα πλαίσια της παρούσας άσκησης να εκτελεί μια περιστροφή από τις 0 μοίρες έως τις 180 μοίρες και αντίστροφα. Στη συνέχεια, και μετά από παύση 10 sec, να επαναλαμβάνει την παραπάνω διαδικασία.



Εικόνα 4.5: Σχηματικό διάγραμμα εφαρμογής 10.

```

#include <Servo.h> // ενσωμάτωση βιβλιοθήκη Servo
Servo servo; // δημιουργία ενός στιγμιότυπου servo
int servo_pos = 0; // μεταβλητή για αποθήκευση θέσης της servo μηχανής
void setup()
{
servo.attach(9); // προσάρτηση της servo μηχανής στον ακροδέκτη 9
}
void loop()
{
for(servo_pos = 0; servo_pos < 180; servo_pos++) // περιστροφή από 0 μέχρι και
180 μοίρες,
ανά 1 μοίρα
{
servo.write(servo_pos); // μετακίνηση της servo μηχανής στη θέση servo_pos
delay(15); // καθυστέρηση για 15 ms
}
for(servo_pos = 180; servo_pos >= 1; servo_pos--) // περιστροφή από 180 μέχρι
και 0
μοίρες , ανά 1 μοίρα
{
servo.write(servo_pos); // μετακίνηση servo μηχανής στη θέση servo_pos
delay(15); /
}
delay(10000); /
}

```

**Κώδικας εφαρμογής 10.**



## BIBΛΙΟΓΡΑΦΙΑ

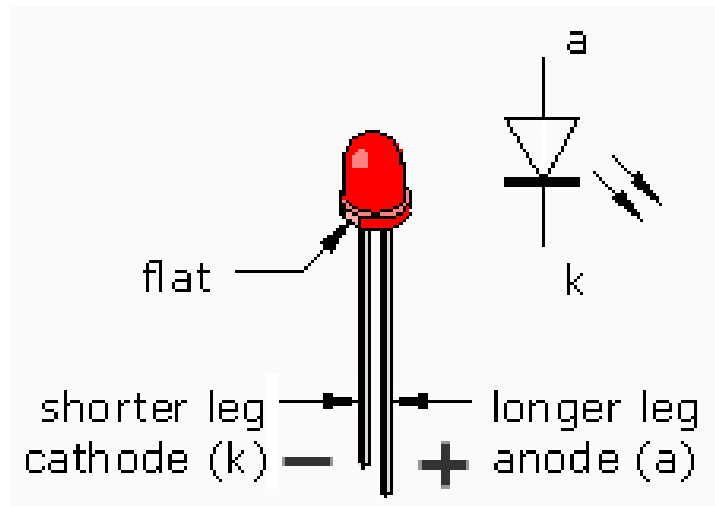
- [1] [http://playground.arduino.cc/uploads/Main/arduino\\_notebook\\_v1-1.pdf](http://playground.arduino.cc/uploads/Main/arduino_notebook_v1-1.pdf)  
(προσπέλαση 2013)
- [2] <http://arduino.cc/> (προσπέλαση 2013)
- [3] IECON 2007. 33rd Annual Conference of the IEEE, A Comparative Study of Wireless Protocols: Bluetooth, UWB, ZigBee, and Wi-Fi, Jin-Shyan Lee Ind. Technol. Res. Inst., Hsinchu Yu-Wei Su, Chung-Chou Shen.
- [4] ZigBee Alliance. <http://www.zigbee.org/>. (προσπέλαση 2013)
- [5] TinyOS. <http://www.tinyos.net/>. (προσπέλαση 2013)
- [6] A. Dunkels, B. Gronvall, and T. Voigt. Contiki – a lithgweight and flexible operating system for tiny networked sensors. In Proceedings of IEEE EmNets 2004. Tampa. Florida, USA, November 2004.
- [7] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci. Wireless sensor networks: a survey. Computer Networks, 38(4): 393-422, March 2002.
- [8] Y. Kim, T. Schmid, Z. M. Charbiwala, J. Friedman, and M. B. Srivastava. NAWMS: Nonintrusive Autonomous Water Monitoring System. In Proceedings of ACM SenSys 2008, pp. 309-322, Raleigh, NC, USA, November 2008.
- [9] G. Werner-Allen, K. Lorinez, M. Ruiz, O.Marcillo, J. Johnson, J. Lees, and M. Welsh. Deploying a wireless sensor network on an active volcano. IEEE Internet Computing, 10(2): 18-25, March/April 2006.
- [10] D. Malan, T. Fulford-Jones, M. Welsh, and S. Moulton. CodeBlue: an Ad Hoc sensor network infrastructure for emerging medical care. In Proceedings of Workshop on Applications of Mobile Embedded Systems, Boston MA, USA, June 2004.
- [11] Arduino Project. <http://www.arduino.cc/>. (προσπέλαση 2013)
- [12] Η χρήση της πλατφόρμας Arduino στη δημιουργία εργαστηρίων από απόσταση. Σπηλιόπουλος Αναστάσιος, απόφοιτος του Ελληνικού Ανοικτού Πανεπιστημίου.

## ΠΑΡΑΡΤΗΜΑ Α΄

### Α-1 Φωτοдиодος (LED)

Μία φωτοδιόδος (Light Emitting Diode, LED) είναι ένας ημιαγωγός ο οποίος έχει την ιδιότητα να εκπέμπει φως όταν τον διαπερνά ηλεκτρικό ρεύμα. Ανάλογα από τη χημική σύσταση του ημιαγωγίμου υλικού μπορεί να εκπέμπει φως σε διαφορετικά μήκη κύματος με αποτέλεσμα να υπάρχουν LED που εκπέμπουν φως διαφόρων χρωμάτων όπως κόκκινο, πράσινο, μπλε, κίτρινο ή ακόμα και υπεριώδες ή υπέρυθρο. Τα πιο συνηθισμένα είναι κυλινδρικού σχήματος με διάμετρο από 2 mm μέχρι και 10 mm ενώ υπάρχουν και σε άλλα διάφορα σχήματα ανάλογα με τη χρήση που προορίζονται.

Στο παρακάτω σχήμα φαίνεται ένα απλό κυλινδρικό LED καθώς και το αντίστοιχο σύμβολο που χρησιμοποιείται σε ένα σχέδιο κυκλώματος.



Το LED έχει δύο πόδια τα οποία ξεχωρίζουν μεταξύ τους με δύο τρόπους. Το πόδι στο οποίο συνδέουμε την τάση καλείται άνοδος και ξεχωρίζει επειδή συνήθως είναι μεγαλύτερου μήκους από το άλλο πόδι. Το πόδι στο οποίο συνδέουμε τη γείωση καλείται κάθοδος και είναι μικρότερου μήκους από το άλλο πόδι. Στην περίπτωση που για κάποιο λόγο δεν μπορούμε να διακρίνουμε το μεγαλύτερο ή μικρότερο πόδι τότε στην πλευρά του LED που βρίσκεται το πόδι της καθόδου υπάρχει μια επίπεδη επιφάνεια (flat) που υποδηλώνει ότι το πόδι αυτό είναι η κάθοδος.

Για τη λειτουργία ενός LED απαιτείται η σύνδεση της κατάλληλης αντίστασης σε σειρά είτε με το πόδι της καθόδου είτε με το πόδι της ανόδου προκειμένου αυτό να μην καταστραφεί.

**A-2 Datasheet 5mm Basic LED**

Color	Forward Voltage (V) @ 20mA	Wavelength (nm)	Luminous (mcd) 5mm	View Angle (deg) 5mm	Type
White	3.2-3.4		14000-16000	160-180	Clear
Warm White	3.2-3.4		13000-14000	160-180	Clear
Red	1.8-2.0	615-625	4000-5000	160-180	Clear
Orange	1.8-2.0	515-520	5000-6000	160-180	Clear
Yellow	1.8-2.0	587-595	4000-5000	160-180	Clear
Green	3.2-3.4	515-520	12000-14000	160-180	Clear
Blue	3.2-3.4	460-465	5000-6000	160-180	Clear
Pink	3.2-3.4		8000-9000	160-180	Clear

**Electrical / Optical Characteristics at TA=25°C**

Symbol	Parameter	Device	Typ.	Max.	Units	Test Conditions
$\lambda_{peak}$	Peak Wavelength	Super Bright Red	660		nm	$I_f=20mA$
$\lambda_D$ [1]	Dominant Wavelength	Super Bright Red	640		nm	$I_f=20mA$
$\Delta\lambda/2$	Spectral Line Half-width	Super Bright Red	20		nm	$I_f=20mA$
C	Capacitance	Super Bright Red	45		pF	$V_f=0V; f=1MHz$
$V_f$ [2]	Forward Voltage	Super Bright Red	1.85	2.5	V	$I_f=20mA$
$I_R$	Reverse Current	Super Bright Red		10	$\mu A$	$V_R = 5V$

Notes:

1. Wavelength: +/-1nm.

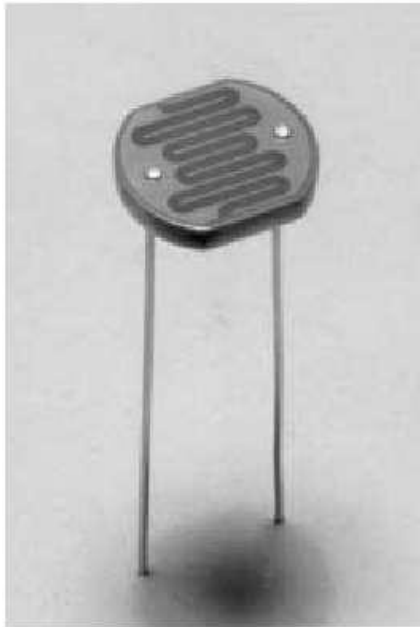
2. Forward Voltage: +/-0.1V.

**A-3 Υπολογισμός τιμής αντίστασης για χρήση με μια φωτοδίοδο (LED)**

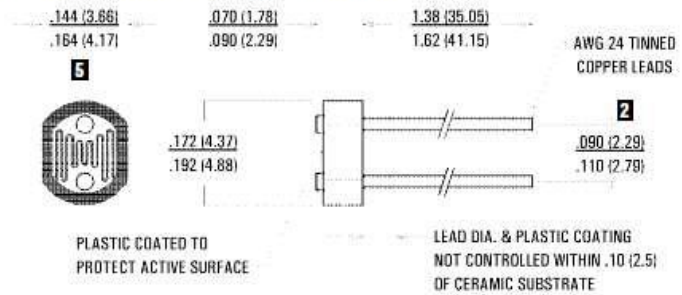
Από το datasheet του LED εντοπίζουμε τις τιμές  $V_f$  (forward voltage σε volt) και  $I_f$  (forward current σε Ampere). Στη συνέχεια υπολογίζουμε την τιμή της αντίστασης σε Ohm. Σύμφωνα με τον τύπο  $R_{led} = V_{cc} - V_f / I_f$ . Όπου  $V_{cc}$  η τάση τροφοδοσίας σε volt. Έτσι για παράδειγμα στο Arduino Uno, με τάση λειτουργίας +5 volt εάν θέλουμε να συνδέσουμε στον έναν ακροδέκτη του, ένα LED, με  $V_f=2,5$  volt και  $I_f=20$  mA=0.02A. Τότε η αντίσταση που προκύπτει θα είναι  $R_{led}=125$  Ω. Εάν δεν έχουμε την συγκεκριμένη αντίσταση, επιλέγουμε την αμέσως μεγαλύτερη που έχουμε διαθέσιμη.

**A-4 Datasheet VT90N2 (500KOhm 80mW LDR)**

**Photoconductive Cell** **VT900 Series**



**PACKAGE DIMENSIONS** inch (mm)



**ABSOLUTE MAXIMUM RATINGS**

Parameter	Symbol	Rating	Units
Continuous Power Dissipation Derate Above 25°C	$P_D$ $\Delta P_D / \Delta T$	80 1.6	mW mW/°C
Temperature Range Operating and Storage	$T_A$	-40 to +75	°C

**ELECTRO-OPTICAL CHARACTERISTICS @ 25°C (16 hrs. light adapt, min.)**

Part Number	Resistance (Ohms) <b>5 6</b>						Material Type	Sensitivity ( $\gamma$ , typ.) <small>LOG (R10R100) LOG (R0R10)</small>	Maximum Voltage (V, pk)	Response Time @ 1 fc (ms, typ.)		
	10 lux 2850 K			2 fc 2850 K	Dark					Rise (1-1/e)	Fall (1/e)	
	Min.	Typ.	Max.	Typ.	Min.	sec.						
VT90N1	6 k	12 k	18 k	6 k	200 k	5	∅	0.80	100	78	8	
VT90N2	12 k	24 k	36 k	12 k	500 k	5	∅	0.80	100	78	8	
VT90N3	25 k	50 k	75 k	25 k	1 M	5	∅	0.85	100	78	8	
VT90N4	50 k	100 k	150 k	50 k	2 M	5	∅	0.90	100	78	8	
VT93N1	12 k	24 k	36 k	12 k	300 k	5	3	0.90	100	35	5	
VT93N2	24 k	48 k	72 k	24 k	500 k	5	3	0.90	100	35	5	
VT93N3	50 k	100 k	150 k	50 k	500 k	5	3	0.90	100	35	5	
VT93N4	100 k	200 k	300 k	100 k	500 k	5	3	0.90	100	35	5	
VT935G												
<b>1</b>	Group A	10 k	18.5 k	27 k	9.3 k	1 M	5	3	0.90	100	35	5
	Group B	20 k	29 k	38 k	15 k	1 M	5	3	0.90	100	35	5
	Group C	31 k	40.5 k	50 k	20 k	1 M	5	3	0.90	100	35	5

## A-5 Datasheet Servo Motor Large



Here is a simple, low-cost, high quality servo for all your mechatronic needs. Large servo with a standard 3 pin power and control cable. Includes hardware as shown.

### Features:

- 180 degree rotation
  - 3 pole ferrite, all nylon gear
  - Top ball bearing
  - Operating Voltage: 4.8V~6.0V
  - Operating speed:
    - 0.20sec/60degree (4.8V)
    - 0.16sec/60degree (6.0V)
  - Stall torque:
    - 5.2kg\*cm (4.8V)
    - 6.5kg\*cm (6.0V)
  - Temperature Range: -20°C~60°C
    - Dead band width: 4μs
- Dimension:
- 41 x 20 x 38mm
  - Weight: 41g
  - Connector wire length: 32cm



## A-6 Datasheet microcontroller Atmega 328p

### Features

- High Performance, Low Power AVR® 8-Bit Microcontroller
- Advanced RISC Architecture
  - 131 Powerful Instructions – Most Single Clock Cycle Execution
  - 32 x 8 General Purpose Working Registers
  - Fully Static Operation
  - Up to 20 MIPS Throughput at 20 MHz
  - On-chip 2-cycle Multiplier
- High Endurance Non-volatile Memory Segments
  - 4/8/16/32K Bytes of In-System Self-Programmable Flash program memory (ATmega48PA/88PA/168PA/328P)
  - 256/512/512/1K Bytes EEPROM (ATmega48PA/88PA/168PA/328P)
  - 512/1K/1K/2K Bytes Internal SRAM (ATmega48PA/88PA/168PA/328P)
  - Write/Erase Cycles: 10,000 Flash/100,000 EEPROM
  - Data retention: 20 years at 85°C/100 years at 25°C(1)
- Optional Boot Code Section with Independent Lock Bits
- In-System Programming by On-chip Boot Program
- True Read-While-Write Operation
  - Programming Lock for Software Security
- Peripheral Features
  - Two 8-bit Timer/Counters with Separate Prescaler and Compare Mode
  - One 16-bit Timer/Counter with Separate Prescaler, Compare Mode, and Capture Mode
  - Real Time Counter with Separate Oscillator
  - Six PWM Channels
  - 8-channel 10-bit ADC in TQFP and QFN/MLF package
- Temperature Measurement
  - 6-channel 10-bit ADC in PDIP Package
- Temperature Measurement
  - Programmable Serial USART
  - Master/Slave SPI Serial Interface
  - Byte-oriented 2-wire Serial Interface (Philips I2C compatible)
  - Programmable Watchdog Timer with Separate On-chip Oscillator
  - On-chip Analog Comparator
  - Interrupt and Wake-up on Pin Change
- Special Microcontroller Features
  - Power-on Reset and Programmable Brown-out Detection
  - Internal Calibrated Oscillator
  - External and Internal Interrupt Sources
  - Six Sleep Modes: Idle, ADC Noise Reduction, Power-save, Power-down, Standby, and Extended Standby
- I/O and Packages
  - 23 Programmable I/O Lines
  - 28-pin PDIP, 32-lead TQFP, 28-pad QFN/MLF and 32-pad QFN/MLF
- Operating Voltage:
  - 1.8 - 5.5V for ATmega48PA/88PA/168PA/328P
- Temperature Range:
  - -40°C to 85°C
- Speed Grade:
  - 0 - 20 MHz @ 1.8 - 5.5V
- Low Power Consumption at 1 MHz, 1.8V, 25°C for ATmega48PA/88PA/168PA/328P:
  - Active Mode: 0.2 mA
  - Power-down Mode: 0.1 µA

## ΠΑΡΑΡΤΗΜΑ Β΄

## Χρωματικός Πίνακας Αντιστάσεων

Χρησιμοποιώντας τον παρακάτω χρωματικό πίνακα μπορούμε να διαβάσουμε μια αντίσταση από τα χρώματά της και να υπολογίσουμε την τιμή της σε Ohms. Η αντίσταση μπορεί να έχει είτε 4 μπάντες χρωμάτων είτε 5 μπάντες χρωμάτων.

**Resistor Color Code**

Color	1 <sup>st</sup> Band	2 <sup>nd</sup> Band	3 <sup>rd</sup> Band	Multiplier	Tolerance
Black	0	0	0	x 1 $\Omega$	
Brown	1	1	1	x 10 $\Omega$	+/- 1%
Red	2	2	2	x 100 $\Omega$	+/- 2%
Orange	3	3	3	x 1K $\Omega$	
Yellow	4	4	4	x 10K $\Omega$	
Green	5	5	5	x 100K $\Omega$	+/- 5%
Blue	6	6	6	x 1M $\Omega$	+/- .25%
Violet	7	7	7	x 10M $\Omega$	+/- .1%
Grey	8	8	8		+/- .05%
White	9	9	9		
Gold				x .1 $\Omega$	+/- 5%
Silver				x .01 $\Omega$	+/- 10%

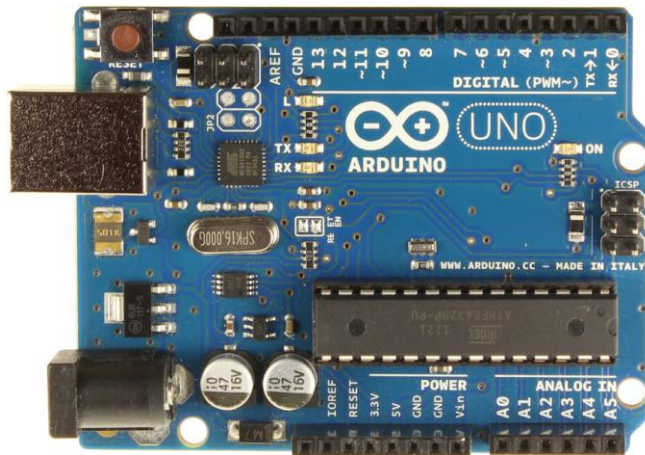
Έτσι, για παράδειγμα, έστω ότι έχουμε μια αντίσταση με 4 χρώματα και συγκεκριμένα πράσινο, μπλε, κίτρινο και χρυσό. Σύμφωνα με τον παραπάνω πίνακα θα έχουμε για το πρώτο χρώμα «5», για το δεύτερο χρώμα «6», για το τρίτο χρώμα που είναι ο πολλαπλασιαστής «10k $\Omega$ » και για το τέταρτο χρώμα που είναι η ανοχή « $\pm 5\%$ ». Επομένως θα έχουμε μια αντίσταση των 560 k $\Omega$  με ανοχή  $\pm 5\%$ .

Έστω τώρα ότι έχουμε μια αντίσταση με 5 χρώματα και συγκεκριμένα κόκκινο, πορτοκαλί, βιολετί, μαύρο και καφέ. Σύμφωνα με τον παραπάνω πίνακα θα έχουμε για το πρώτο χρώμα «2», για το δεύτερο χρώμα «3», για το τρίτο χρώμα «7», για το τέταρτο χρώμα που είναι ο πολλαπλασιαστής «1 $\Omega$ » και για το πέμπτο χρώμα που είναι η ανοχή « $\pm 1\%$ ». Επομένως θα έχουμε μια αντίσταση 237 $\Omega$  με ανοχή  $\pm 1\%$ .

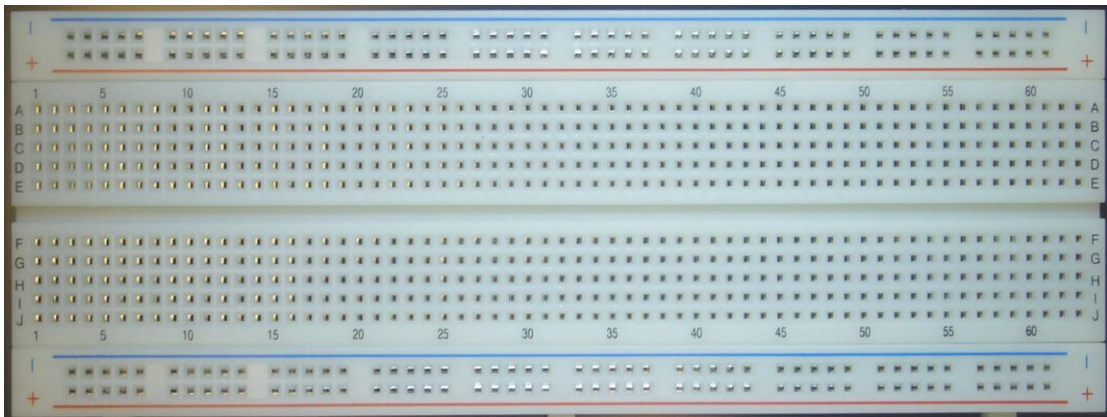
## ΠΑΡΑΡΤΗΜΑ Γ΄

Υλικά που χρησιμοποιήθηκαν στις εφαρμογές:

### 1. Arduino Uno R2



### 2. Breadboard



Το board είναι μια κατασκευή στην οποία υλοποιείται η συνδεσμολογία των στοιχείων των κυκλωμάτων και δεν έχει άλλη λειτουργία εκτός αυτής. Όσον αφορά τον τρόπο σύνδεσης και διαμόρφωσης των κυκλωμάτων, θα πρέπει να έχουμε υπόψη ότι το board αποτελείται από σειριακά βραχυκυκλωμένα σημεία (τρηπούλες που συνδέουμε τα στοιχεία ή τα καλώδια).

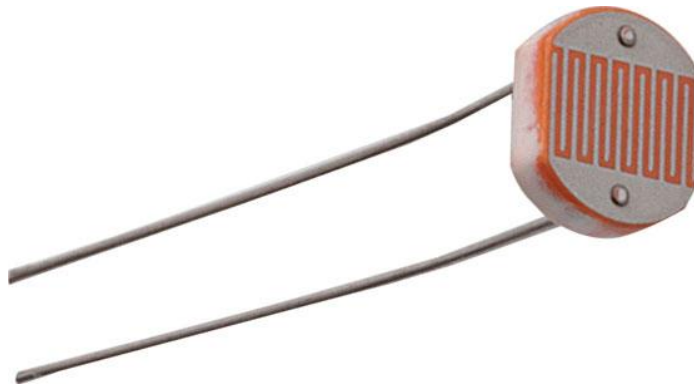
## ΠΑΡΑΡΤΗΜΑΤΑ

**3. Δίοδος Εκπομπής Φωτός, (*LED, Light Emitting Diode*),** αποκαλείται ένας ημιαγωγός ο οποίος εκπέμπει φωτεινή ακτινοβολία στενού φάσματος όταν του παρέχεται μία ηλεκτρική τάση κατά τη φορά ορθής πόλωσης (forward-biased)

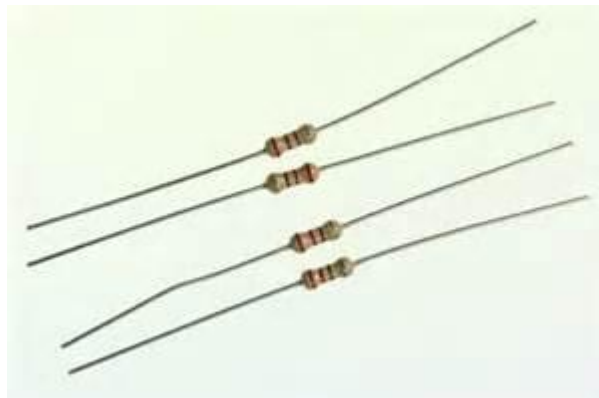


## **4. Φωτοαντίσταση (*Light Dependent Resistor, LDR*)**

Πρόκειται για ένα είδος αντίστασης, η αντίσταση της οποίας μειώνεται όταν αυξάνεται το φως που προσπίπτει στην επιφάνειά της. Ενδεικτικά, βρίσκει εφαρμογές σε μέτρηση των επιπέδων φωτός στον περιβάλλοντα χώρο καθώς και σε συστήματα συναγερμού.



## **5. Resistors (αντιστάσεις) R=330 Ω / R=1 ΚΩ / R=10 ΚΩ**



## ΠΑΡΑΡΤΗΜΑΤΑ

### 6. Servomotor (σερβοκινητήρας).



### 7. Potentiometer (ποτενσιόμετρο).



### 8. Καλώδια για την συνδεσμολογία των υλικών.



### 9. Διακόπτες (button)



## ΠΑΡΑΡΤΗΜΑ Δ΄

### Προγραμματιστικό περιβάλλον του Arduino IDE

Το Arduino IDE, είναι ένα ολοκληρωμένο περιβάλλον ανάπτυξης (integrated development environment), που περιέχει έναν επεξεργαστή πηγαίου κώδικα, έναν μεταγλωττιστή, εργαλεία αυτόματης παραγωγής κώδικα, αποσφαλματωτή, συνδετή, σύστημα ελέγχου εκδόσεων και εργαλεία κατασκευής γραφικών διασυνδέσεων χρήστη για τις υπό ανάπτυξη εφαρμογές. Συνδέεται με το hardware μέρος του Arduino για να φορτώσει προγράμματα και να επικοινωνεί μαζί τους. Ο κώδικας που έχει γραφτεί για το Arduino ονομάζεται sketch.

Το Arduino IDE είναι βασισμένο σε Java και συγκεκριμένα παρέχει:

- Ένα πρακτικό περιβάλλον για τη συγγραφή των προγραμμάτων ,με συντακτική χρωματική σήμανση.
- Βιβλιοθήκες για προέκταση της, εύκολες στην πρόσβαση μέσω διαδικτύου.
- Compiler (μεταγλωττιστής) για την μεταγλώττιση των sketch.
- Μια σειριακή οθόνη (serial monitor) που παρακολουθεί τις επικοινωνίες της σειριακής (USB), αναλαμβάνει να στείλει αλφαριθμητικά στο Arduino μέσω αυτής και είναι ιδιαίτερα χρήσιμο για τον προσδιορισμό τυχόν λαθών των sketch.
- Την επιλογή για μεταφορά των μεταγλωττισμένων sketch στο Arduino.

Η γλωσσά συγγραφής κώδικα του Arduino, βασίζεται στη γλώσσα Wiring, μια παραλλαγή C/C++ για μικροελεγκτές αρχιτεκτονικής AVR όπως ο AtMega και υποστηρίζει όλες τις βασικές δομές της C, καθώς και μερικά χαρακτηριστικά της C++.

Λόγω της καταγωγής της από τη C, στη γλώσσα συγγραφής κώδικα του Arduino, μπορούν να χρησιμοποιηθούν ουσιαστικά οι ίδιες βασικές εντολές και συναρτήσεις, με την ίδια σύνταξη, τους ίδιους τύπους δεδομένων και τους ίδιους τελεστές όπως και στη C. Πέρα από αυτές όμως υπάρχουν κάποιες ειδικές εντολές, συναρτήσεις και σταθερές που βοηθούν για τη διαχείριση του ειδικού hardware του Arduino. Στην παρακάτω εικόνα αναδεικνύεται το περιβάλλον εργασίας του Arduino IDE.

## ΠΑΡΑΡΤΗΜΑΤΑ

