

**Τεχνολογικό Εκπαιδευτικό Ίδρυμα Κρήτης**



**Σχολή Τεχνολογικών Εφαρμογών**

**Τμήμα Εφαρμοσμένης Πληροφορικής & Πολυμέσων**



**Πτυχιακή εργασία**

**Εφαρμογές ασύρματης τηλεμετρίας στην  
αναπτυξιακή πλατφόρμα Arduino**

**Ελευθεριάδης Μιχαήλ (ΑΜ:982)**

**Επιβλέπων καθηγητής : Κορνάρος Γεώργιος**

**Επιτροπή Αξιολόγησης:**

**Ημερομηνία παρουσίασης:**



# Ευχαριστίες

Ιδιαίτερο ευχαριστώ οφείλω στον Γεώργιο Κορνάρο και εισηγητή της πτυχιακής για την άριστη πληροφόρηση πάνω σε θέματα μικροϋπολογιστών, μικροελεγκτών και αρχιτεκτονικής υπολογιστών. Όπως επίσης και όλους τους διδάσκοντες του τμήματος Εφαρμοσμένης Πληροφορικής & Πολυμέσων που με τα εκάστοτε μαθήματα τους διαμόρφωσαν το απαραίτητο επίπεδο γνώσεων που ήταν η βάση της παρούσας εργασίας. Τέλος στην οικογένεια μου για την αμέριστη βοήθεια σε θέματα δακτυλογράφησης και ποιοτικού έλεγχου των κειμένων.



# Abstract

By using **open source hardware** and **sensors** that are widely available commercially, we can obtain **distant measurements** and recording of environmental conditions and checking for possible invasion at any given area. The **Arduino** and the **Wishield** are **development boards** suitable for these specific applications.

The Arduino board allows us to quickly record external events through the available **Atmel ATmega 328P** 20 Mhz processor. It has enough memory for calculations and for code storage.

- 2 KB **SRAM** type memory for calculations.
- 1 KB **EEPROM** type memory for permanent storage after deactivation of the system.
- 32 KB **Flash** type memory to store the executable code.

The Wishield provides the **802.11b** wireless protocol and the ability to serve simple **websites** that we can configure their content. It will also incorporate the relevant sensors on the platform:

- The temperature sensor **DS18B20** with a measuring range from -55 to 125 °C.
- The relative humidity sensor Honeywell **HIH-4030** with range from 0 to 100% RH.
- And the **Parallax 555-280227** PIR technology motion sensor with a coverage area of up to 6 meters.

The subject of this thesis is the design and study of special **telemetry** subsystem on the Arduino development environment which is based on C ++.



# Σύνοψη

Με την χρήση υλικού ανοιχτού κώδικα και αισθητήρες που κυκλοφορούν ευρέως στο εμπόριο, μπορούμε να επιτύχουμε την εξ αποστάσεως μέτρηση και καταγραφή των περιβαλλοντικών συνθηκών, όπως και τον έλεγχο για πιθανή εισβολή σε κάθε χώρο. Η **Arduino** και η **Wishield** είναι αναπτυξιακές πλακέτες κατάλληλες για τις συγκεκριμένες εφαρμογές.

Η πλακέτα **Arduino** μας επιτρέπει να έχουμε γρήγορη καταγραφή εξωτερικών γεγονότων μέσω του επεξεργαστή **Atmel ATmega 328P** των 20 Mhz που διαθέτει. Διαθέτει αρκετή μνήμη για υπολογισμούς και για αποθήκευση κώδικα.

- Μνήμη τύπου **SRAM** μεγέθους 2 KB για τους υπολογισμούς.
- Μνήμη τύπου **EEPROM** μεγέθους 1 KB για την μόνιμη αποθήκευση δεδομένων μετά την απενεργοποίηση του συστήματος.
- Μνήμη τύπου **Flash** μεγέθους 32 KB για την αποθήκευση του κώδικα προς εκτέλεση.

Η **Wishield** μας παρέχει ασύρματο πρωτόκολλο **802.11b** και την δυνατότητα να μας διακομίζει (serve) άπλες ιστοσελίδες που μπορούμε να διαμορφώσουμε το περιεχόμενο τους. Επίσης θα ενσωματωθούν στην πλατφόρμα σχετικοί αισθητήρες:

- Ο αισθητήρας θερμοκρασίας **DS18B20** με εύρος μέτρησης από -55 έως +125 °C.
- Ο αισθητήρας σχετικής υγρασίας **Honeywell HIH-4030** με εύρος 0 έως 100% RH.
- Όπως και ο **Parallax 555-280227** αισθητήρας κίνησης, τεχνολογίας PIR με περιοχή κάλυψης έως 6 μέτρα.

Θέμα αυτής της πτυχιακής εργασίας είναι η μελέτη και σχεδίαση ειδικού υποσυστήματος τηλεμετρίας και εφαρμογής στο περιβάλλον ανάπτυξης του **Arduino** που βασίζεται στην **C++**.





# Πίνακας Περιεχομένων

Abstract .....	v
Σύνοψη .....	vii
Πίνακας Περιεχομένων .....	ix
Πίνακας Εικόνων.....	xiv
Λίστα Πινάκων.....	xv
1. Εισαγωγή.....	1
1.1. Κίνητρο για την διεξαγωγή της εργασίας .....	1
1.2. Σκοπός και Στόχοι Εργασίας .....	2
1.3. Δομή Εργασίας .....	2
2. Γενική εισαγωγή στην τηλεμετρία .....	4
2.1. Απαρχές και η ανάπτυξη εφαρμογών τηλεμετρίας.....	4
2.2. Αρχή λειτουργίας.....	4
2.3. Οφέλη & Εφαρμογές .....	5
2.4. Άλλες εφαρμογές επιγραμματικά .....	5
3. Εισαγωγή στους μικροελεγκτές .....	7
3.1. Ψηφιακά ολοκληρωμένα κυκλώματα .....	7
3.2. Μονολιθικά Ο.Κ .....	7
3.3. Υβριδικά Ο.Κ.....	7
3.4. Χρήσιμα χαρακτηριστικά των Ο.Κ.....	8
3.5. Ταξινόμηση των Ο.Κ βάσει βαθμού ολοκλήρωσης.....	8
3.6. Μικροεπεξεργαστής.....	9
3.7. Μικροελεγκτής .....	9
3.8. Μικροελεγκτές σε σύγκριση με τους μικροεπεξεργαστές .....	9
3.9. Είδη μικροελεγκτών.....	10
3.9.1. Μικροελεγκτές 8, 16 και 32 bit .....	10
3.9.2. Ενσωματωμένος μικροελεγκτής .....	10
3.9.3. Μικροελεγκτής εξωτερικής μνήμης .....	10
3.9.4. Αρχιτεκτονική Von-Neuman (ή Princeton) .....	10
3.9.5. Αρχιτεκτονική Harvard.....	11
3.9.6. Αρχιτεκτονική CISC.....	11
3.9.7. Αρχιτεκτονική RISC.....	12

3.10.	Βασική δομή του μικροελεγκτή.....	12
3.11.	Μονάδα μνήμης.....	13
3.11.1.	Τα βασικότερα είδη μνήμης: .....	14
3.12.	Δίαυλος.....	14
3.13.	Μονάδα εισόδου/εξόδου (I/O).....	14
3.14.	Σειριακή επικοινωνία.....	15
3.15.	Μονάδα χρονισμού (Timer).....	15
3.16.	Μονάδα μετατροπής από αναλογικό σε ψηφιακό (ADC) .....	15
3.17.	Μονάδα εποπτείας χρονισμού (Watchdog Timer).....	15
3.18.	Προγραμματισμός μικροελεγκτών.....	16
4.	Παρουσίαση της αναπτυξιακής πλακέτας Arduino.....	18
4.1.	Τί είναι το Arduino; .....	18
4.2.	Γιατί Arduino; (έναντι άλλων).....	19
4.3.	Το hardware του Arduino .....	19
4.4.	Οι ακροδέκτες του Arduino .....	20
4.4.1.	Κάποιοι ακροδέκτες έχουν συγκεκριμένες λειτουργίες.....	20
4.4.2.	Ακροδέκτες για ειδικές λειτουργίες όπως:.....	21
4.5.	Μνήμη.....	21
4.6.	Τροφοδοσία.....	21
4.7.	Επικοινωνία .....	22
4.8.	Αυτόματη software επαναφορά .....	22
4.9.	USB προστασία από υπερτάση.....	22
4.10.	Περιβάλλον ανάπτυξης εφαρμογών.....	22
5.	Εισαγωγή στους αισθητήρες .....	25
5.1.	Αίσθηση και ερέθισμα .....	25
5.2.	Αισθητήρας και το αισθητήριο .....	25
5.3.	Βασικά είδη αισθητήρων .....	26
5.4.	Ταξινόμηση αισθητήρων .....	26
5.4.1.	Ο αισθητήρας μπορεί να είναι: παθητικός ή ενεργητικός.....	26
5.4.2.	Ανάλογα με το σύστημα αναφοράς, οι αισθητήρες μπορούν να ταξινομηθούν σε απόλυτους και σε σχετικούς.....	26
5.4.3.	Ένας άλλος τρόπος να ταξινομήσουμε έναν αισθητήρα είναι βάσει της ιδιαιτερότητας των χαρακτηριστικών ή των ιδιοτήτων του.....	27
5.5.	Στατικά χαρακτηριστικά των αισθητήρων.....	27
5.6.	Δυναμικά χαρακτηριστικά .....	27

5.7.	Συνοπτική παρουσίαση διαθέσιμων αισθητήρων: .....	28
6.	Θερμοκρασία, υγρασία και οι διαθέσιμοι αισθητήρες .....	30
6.1.	Θερμοκρασία .....	30
6.2.	Κλίμακα Κέλβιν .....	30
6.3.	Απόλυτη θερμοκρασία.....	31
6.4.	Σχετική θερμοκρασία.....	31
6.5.	Θερμοκρασία και αρχή λειτουργίας κάθε αισθητήρα θερμοκρασίας .....	31
6.6.	Η υγρασία .....	33
6.7.	Οι υγρομετρικές παράμετροι .....	33
6.7.1.	Απόλυτη Υγρασία.....	33
6.7.2.	Σχετική υγρασία.....	33
6.8.	Σημείο Δρόσου .....	34
6.9.	Υγρασία και διαθέσιμοι αισθητήρες.....	35
6.10.	Σφάλμα μέτρησης και αισθητήρες.....	36
6.11.	Η αβεβαιότητα μέτρησης.....	36
6.12.	Τυχαία σφάλματα και συστηματικά σφάλματα .....	36
6.13.	Προσδιορισμός τιμής.....	37
7.	Παρουσίαση των αισθητήρων του υποσυστήματος τηλεμετρίας.....	39
7.1.	Αισθητήρας θερμοκρασίας Dallas DS18B20 .....	39
7.2.	Χαρακτηριστικά:.....	39
7.3.	Περιγραφή ακροδεκτών.....	40
7.4.	Αισθητήρες θερμοκρασίας ολοκληρωμένου κυκλώματος.....	40
7.5.	Περιληπτική παρουσίαση του αισθητήρα DS18B20 .....	40
7.6.	Μοντέλο επικοινωνίας master/slave .....	41
7.7.	Τεχνολογία 1-Wire.....	41
7.8.	Ιδιαίτερα χαρακτηριστικά της τεχνολογίας 1-Wire .....	42
7.9.	Μέτρηση θερμοκρασίας.....	42
7.10.	Μνήμη.....	42
7.11.	Καταχωρητής θερμοκρασίας .....	43
7.12.	Καταχωρητής ρυθμίσεων.....	43
7.13.	Δημιουργία κώδικα CRC (κυκλικός έλεγχος πλεονασμού).....	44
7.14.	Σήμα συναγερμού .....	44
7.15.	Κώδικας χάραξης Laser Rom μνήμης 64bit .....	45
7.16.	Αρχικοποίηση .....	45

7.17.	Τροφοδοσία και παρασιτική τάση λειτουργίας.....	46
7.18.	Ρυθμίσεις.....	46
7.19.	Αλληλουχία ανταλλαγής δεδομένων .....	46
7.20.	Διαδικασία αρχικοποίησης - παλμοί reset και παρουσίας .....	46
7.21.	Σήματα πρωτοκόλλου I-Wire.....	47
7.22.	Εντολές διαχείρισης μνήμης ROM.....	47
7.23.	Εντολές λειτουργιών του DS18B20.....	48
7.24.	Αισθητήρας υγρασίας Honeywell HIH-4030 .....	49
7.25.	Χαρακτηριστικά:.....	49
7.26.	Αρχή λειτουργίας.....	49
7.27.	Χωρητικοί αισθητήρες υγρασίας .....	50
7.28.	Αντοχή κοντά στο σημείο δρόσου .....	50
7.29.	Ιδιαίτερα χαρακτηριστικά του HIH-4030 .....	50
7.30.	Θερμοσκληρυντικό περίβλημα .....	51
7.31.	Υψηλή γραμμικότητα και επιδόσεις.....	51
7.32.	Αισθητήρας PIR Parallax 555 - 280227 .....	53
7.33.	Χαρακτηριστικά:.....	53
7.35.	Ρύθμιση jumper.....	54
7.36.	Θεωρία λειτουργίας .....	54
7.37.	Ανάλυση της θεωρίας λειτουργίας.....	55
7.38.	Φακός Fresnel .....	56
7.39.	Φακός Fresnel τύπου NL-11NH .....	56
7.40.	Οπτικό πεδίο φακού.....	57
7.41.	Επιπλέον πληροφορίες.....	58
7.42.	Βαθμονόμηση .....	58
8.	Εισαγωγή στα ασύρματα τοπικά δίκτυα.....	60
8.1.	Το πρότυπο IEEE 802.11 .....	60
8.2.	Οι βασικές του μονάδες του 802.11 .....	60
8.3.	Η τοπολογία του 802.11.....	61
8.4.	Τύποι κίνησης σταθμών.....	61
8.5.	Μηχανισμοί λειτουργίας.....	62
8.6.	Η αρχιτεκτονική του 802.11 .....	62
8.7.	Το υπόστρωμα MAC (Medium Access Control Sublayer) του 802.11 .....	62
8.8.	Πρόσβαση στο ασύρματο μέσο μέσω της λειτουργίας CSMA/CA.....	62
8.9.	Χρονικά διαστήματα πρόσβασης.....	63

8.10.	Το φυσικό επίπεδο του 802.11.....	63
8.11.	Λειτουργίες του φυσικού στρώματος .....	63
8.12.	Υπηρεσίες που παρέχει το πρωτόκολλο 802.11 .....	64
8.12.1.	Station Services.....	64
8.12.2.	Distribution System Services.....	64
8.13.	Το πρωτόκολλο IEEE 802.11b .....	65
8.14.	Περιγραφή.....	65
8.15.	Direct Sequence Spread Spectrum (DSSS).....	65
8.16.	Κανάλια και συχνότητες εκπομπής.....	66
8.17.	Ανοχή στον θόρυβο .....	67
8.18.	Παρουσίαση της WiShield.....	68
8.19.	Χαρακτηριστικά.....	68
8.20.	Χαρακτηριστικά μονάδας Wi-Fi.....	69
8.21.	Πως χρησιμοποιούμε το κάθε pin.....	69
9.	Υλοποίηση του τηλεμετρικού υποσυστήματος.....	72
9.1.	Απαιτήσεις του τηλεμετρικού υποσυστήματος.....	72
9.2.	Απαιτήσεις των αισθητήρων.....	72
9.3.	Απαιτήσεις του μικροελεγκτή.....	73
9.4.	Απαιτήσεις του ασύρματου πρωτόκολλου.....	73
9.5.	Εγκατάσταση και ρύθμιση του Arduino IDE .....	73
9.6.	Παραγωγή εφαρμογών μέσω του Arduino IDE.....	75
9.7.	Οι στόχοι υλοποίησης του τηλεμετρικού υποσυστήματος .....	76
9.8.	Παρουσίαση και ανάλυση υλοποίησης.....	77
9.9.	Ανάλυση της συνάρτησης setup .....	77
9.10.	Ανάλυση της συνάρτησης loop.....	77
9.11.	Ανάλυση της συνάρτησης servePage.....	77
9.12.	Παρουσίαση των ιστοσελίδων της υλοποίησης.....	78
9.13.	Παρουσίαση του κυκλώματος της υλοποίησης .....	79
9.14.	Ρύθμιση και εγκατάσταση του τηλεμετρικού υποσυστήματος.....	80
10.	Συμπεράσματα.....	82
10.1.	Μελλοντικές επεκτάσεις.....	82

Βιβλιογραφία.....	84
Παραρτήματα .....	86
Παράρτημα Α κώδικας υλοποίησης σε C++ .....	86
Παράρτημα Β κώδικας html ιστοσελίδων.....	97
Πρώτη ιστοσελίδα:.....	97
Δεύτερη ιστοσελίδα:.....	98
Παράρτημα Γ το css style sheet των δύο ιστοσελίδων.....	99
Παράρτημα Δ παρουσίαση πτυχιακής.....	100

## Πίνακας Εικόνων

Εικόνα 2. 1: Παραδείγματα τηλεμετρίας.....	4
Εικόνα 3. 1: Μονολιθικά ολοκληρωμένα κυκλώματα σε διάφορα πλαστικά περιβλήματα .....	7
Εικόνα 3. 2: Ένα υβριδικό ολοκληρωμένο κύκλωμα.....	7
Εικόνα 3. 3: Ταξινόμηση μικροελεγκτών .....	10
Εικόνα 3. 4: Μπλοκ διάγραμμα αρχιτεκτονικής Von-Neuman.....	11
Εικόνα 3. 5: Μπλοκ διάγραμμα αρχιτεκτονικής Harvard .....	11
Εικόνα 3. 6: Θεμελιώδεις μονάδες του μικροελεγκτή .....	12
Εικόνα 3. 7: Διάγραμμα μπλοκ μικροελεγκτή .....	13
Εικόνα 4. 1: Κάτοψη του Arduino .....	18
Εικόνα 5. 1: Σχηματική παρουσίαση του σύνθετου αισθητήρα.....	26
Εικόνα 6. 1: Διάγραμμα μεταβολής σχετικής υγρασίας συναρτήσει της θερμοκρασίας .....	34
Εικόνα 6. 2: Διάγραμμα μεταβολής σημείο δρόσου συναρτήσει της θερμοκρασίας.....	35
Εικόνα 6. 3: Γραφική αναπαράσταση του συστηματικού σφάλματος.....	36
Εικόνα 6. 4: Γραφική αναπαράσταση του τυχαίου σφάλματος .....	37
Εικόνα 7. 1: Ο DS18B20 σε μορφή TO92 .....	39
Εικόνα 7. 2: Διάγραμμα block του DS18B20 .....	41
Εικόνα 7. 3: Βασικό διάγραμμα διαύλου 1-Wire .....	42
Εικόνα 7. 4: Σχηματική αναπαράσταση της μνήμης(scratchpad) .....	43
Εικόνα 7. 5: Διαμόρφωση bit του καταχωρητή θερμοκρασίας.....	43
Εικόνα 7. 6: Διάγραμμα θέσεων μνήμης καταχωρητή ρυθμίσεων.....	44
Εικόνα 7. 7: Υπάρχει άμεση αύξηση στον χρόνο μετατροπής αν η ανάλυση αυξηθεί. ....	44
Εικόνα 7. 8.: Σχηματική παρουσίαση υπολογισμού του CRC .....	44
Εικόνα 7. 9: Διαμόρφωση bit των TH και TL καταχωρητών.....	44
Εικόνα 7. 10: Σχηματική παρουσίαση του κώδικα χάραξης Laser.....	45
Εικόνα 7. 11: Τροφοδοσία με χρήση εξωτερικής τάσης.....	45

Εικόνα 7. 12: Τροφοδοσία με χρήση παρασιτικής τάσης. ....	45
Εικόνα 7. 13: Διάγραμμα αλληλουχίας αρχικοποίησης. ....	47
Εικόνα 7. 14: Ο ΗΙΗ-4030 σε breakout πλακέτα. ....	49
Εικόνα 7. 15: Βασική δομή αισθητήρα. ....	50
Εικόνα 7. 16: Τάση εξόδου σε συνάρτηση με την σχετική υγρασία στους 0 ° C, 70 ° C και 5 V. ....	51
Εικόνα 7. 17: Τάση εξόδου σε συνάρτηση με την σχετική υγρασία στους 25 ° C και 5 V. ....	51
Εικόνα 7. 18: Οι επιδόσεις σε διαφορετικές περιοχές θερμοκρασίας. ....	52
Εικόνα 7. 19: Ο αισθητήρας PIR στεγάζεται σε ένα ερμητικά σφραγισμένο μεταλλικό κέλυφος ώστε ο θόρυβος, θερμοκρασία και η υγρασία να μην το επηρεάζει. ....	53
Εικόνα 7. 20: Βασικό σχήμα της αρχής λειτουργίας. ....	54
Εικόνα 7. 21: Το φως που εισέρχεται στην κοιλότητα από μια μικρή οπή, έπειτα από πολλαπλές αντανακλάσεις απορροφάται σχεδόν ολοκληρωτικά από τα τοιχώματα. ....	55
Εικόνα 7. 22: Εδώ φαίνεται πως το κάθε πρίσμα επιφέρει σύγκλιση στο ίδιο σημείο. ....	56
Εικόνα 7. 23: Όψεις του φακού Fresnel NL-11NH. ....	56
Εικόνα 7. 24: Διαγράμματα της πλάγιας οπτικής και κάτοψης του φακού Fresnel NL-11NH. ....	57
Εικόνα 7. 25: Περιοχές και εύρος κάλυψης σε διάφορα σημεία προσάρτησης. ....	57
Εικόνα 8. 1: Ένα τυπικό WLAN. ....	60
Εικόνα 8. 2: Στοιβά πρωτοκόλλου IEEE 802.11. ....	61
Εικόνα 8. 3: Τα δεδομένα (data signal) κωδικοποιούνται με το σήμα «τεχνητού θορύβου» (PN-code) και παράγεται το τελικό σήμα που εκπέμπεται (coded signal). ....	65
Εικόνα 8. 4: Η ωφέλιμη πληροφορία (Narrowband Waveform) πλαισιώνεται με το σήμα «τεχνητού θορύβου» (Spread Waveform) και εξαπλώνεται στο φάσμα συχνοτήτων. ....	66
Εικόνα 8. 5: Διάγραμμα επικάλυψης καναλιών. ....	66
Εικόνα 8. 6: Κάτοψη της WiShield. ....	68
Εικόνα 8. 7: Η κεραία της WiShield και η προσάρτηση πάνω στο Arduino. ....	68
Εικόνα 9. 1: Δομή αρχείων εντός του Arduino IDE. ....	74
Εικόνα 9. 2: Επιλογή τύπου Arduino. ....	75
Εικόνα 9. 3: Λειτουργίες Compile/Verify και Upload στο Arduino IDE. ....	75
Εικόνα 9. 4: Η αρχική (πρώτη) ιστοσελίδα. ....	78
Εικόνα 9. 5: Η δεύτερη ιστοσελίδα. ....	79
Εικόνα 9. 6: Φωτογραφία του κυκλώματος υλοποίησης. ....	80

## Λίστα Πινάκων

Πίνακας 1: Χαρακτηριστικά Arduino. ....	20
Πίνακας 2: Είδη διαθέσιμων αισθητήρων. ....	28
Πίνακας 3: Ακρίβεια μέτρησης θερμοκρασίας σε συνάρτηση με την ακρίβεια σε βαθμούς °C. ....	32
Πίνακας 4: Ακροδέκτες του DS18B20. ....	40
Πίνακας 5: Αντιστοίχιση θερμοκρασίας και τιμής καταχωρητή. ....	42
Πίνακας 6: Εντολές διαχείρισης μνήμης ROM. ....	47
Πίνακας 7: Εντολές λειτουργίας του DS18B20. ....	48
Πίνακας 8: Τα pin του ΗΙΗ-4030. ....	53
Πίνακας 9: Οι θέσεις του jumper ρυθμίσεων του ΗΙΗ-4030. ....	54
Πίνακας 10: Οι συχνότητες των καναλιών του 802.11b. ....	66





# 1. Εισαγωγή

Στην εργασία αυτή, αναπτύξαμε ένα τηλεμετρικό υποσύστημα που βασίζεται στην αναπτυξιακή πλατφόρμα Arduino. Επιπλέον, προστεθήκαν αισθητήρες και μια ακόμη αναπτυξιακή πλακέτα που μας παρέχει το ασύρματο πρωτόκολλο 802.11b. Οι αισθητήρες αυτοί μας δίνουν την δυνατότητα να εκτελέσουμε μετρήσεις σε περιβαλλοντικές παραμέτρους όπως, την τιμή της θερμοκρασίας, την τιμή της σχετικής υγρασίας αλλά και την δυνατότητα ανίχνευσης κίνησης.

Για την επίτευξη του τηλεμετρικού υποσυστήματος μελετήθηκαν θέματα όπως τηλεμετρία, μικροελεγκτές, αισθητήρες και ασύρματα πρωτόκολλα. Η παραπάνω μελέτη μας επέτρεψε να κατανοήσουμε όλα τα γνωστικά πεδία που απαιτούνται για την ανάπτυξη ενός αποδοτικού τηλεμετρικού υποσυστήματος που θα συνδυάζει όλα τα εξαρτήματα που αναφέραμε πριν. Στην συνέχεια πραγματοποιήθηκε η υλοποίηση του τηλεμετρικού υποσυστήματος.

Η υλοποίηση είναι βασισμένη στην γλώσσα προγραμματισμού C++. Η διαδικασία διευκολύνθηκε από την ύπαρξη ενός εργαλείου ανάπτυξης εφαρμογών, το Arduino IDE. Το Arduino IDE μας επιτρέπει να επιταχύνουμε την διαδικασία υλοποίησης, λόγω των ιδιαίτερων χαρακτηριστικών του. Αυτά περιλαμβάνουν την ενσωμάτωση σ' αυτό, compiler, hardware programmer και code editor.

Η υλοποίηση του τηλεμετρικού υποσυστήματος, το οποίο μπορεί να τοποθετηθεί σε όποιο χώρο επιθυμούμε και να μελετήσουμε τις περιβαλλοντικές συνθήκες του αλλά και να είμαστε σε θέση να ανιχνεύσουμε την πιθανή εισβολή. Τέλος, τα παραπάνω μπορούμε να τα παρακολουθήσουμε από απόσταση μέσω των ιστοσελίδων το οποίο είναι σε θέση το υποσύστημα να μας διακομίζει.

## 1.1. Κίνητρο για την διεξαγωγή της εργασίας

Η εξέλιξη της τεχνολογίας τα τελευταία χρόνια έχει επιφέρει σημαντικές καινοτομίες στο χώρο της πληροφορικής. Καινοτομίες που επέτρεψαν την σημαντική αύξηση της επεξεργαστικής ισχύος των υπολογιστικών συστημάτων και ταυτόχρονα την σμίκρυνση τους αλλά και την μείωση του κόστους σε τέτοια επίπεδα ώστε η απόκτηση τους να μην αποτελεί πρόβλημα. Πλέον με ιδιαίτερα χαμηλό κόστος μπορούμε να έχουμε στην διάθεση μας ένα υπολογιστικό σύστημα με αξιοσέβαστες δυνατότητες που μας επιτρέπουν ένα ευρύ φάσμα εφαρμογών.

Τα εργαλεία ανάπτυξης λογισμικού ευνοήθηκαν ιδιαίτερα από το λογισμικό ανοιχτού κώδικα. Το λογισμικό ανοιχτού κώδικα (open source software) σαν φιλοσοφία και μέθοδος διανομής και επαναχρησιμοποίησης πηγαίου κώδικα (source code) μας δίνει την δυνατότητα να έχουμε δωρεάν πρόσβαση σε εφαρμογές αλλά και στον πηγαίο κώδικα τους. Οπότε η ανάπτυξη εφαρμογών γίνεται γρηγορότερα όπως και η βελτιστοποίηση τους είναι ευκολότερη λόγω της διαφάνειας που διαθέτει το λογισμικό ανοιχτού κώδικα.

Άμεσο αποτέλεσμα των παραπάνω είναι η δημιουργία εφαρμογών που μπορούν να εκτελεστούν σε οποιοδήποτε λειτουργικό σύστημα. Με αυτή την εξέλιξη καταλήγουμε στην δημιουργία εργαλείων ανεξάρτητου πλατφόρμας (platform independent). Δηλαδή τα εργαλεία ανάπτυξης για το λειτουργικό μας σύστημα έχουν τις ίδιες λειτουργίες, αλλά και πλήρη συμβατότητα μ' αυτήν σ' ένα άλλο λειτουργικό σύστημα.

Το κόστος απόκτησης ενός μικροελεγκτή θεωρείται πλέον ιδιαίτερα προσιτό σ' όσους θέλουν να τους χρησιμοποιήσουν ώστε να αναπτύξουν εφαρμογές. Οι δυνατότητες ενός μέσου μικροελεγκτή είναι αξιοσέβαστες και ικανές να παράγουν πλήθος εφαρμογών. Αλλά κι τελευταίες εξελίξεις στο χώρο ανάπτυξης νέων μικροελεγκτών μας πρόσφεραν μικροελεγκτές για κάθε πιθανή εφαρμογή.

Η ανάπτυξη εφαρμογών για μικροελεγκτές δεν αποτελεί έργο που μπορεί να ολοκληρωθεί μονό από έμπειρους και εξειδικευμένους προγραμματιστές. Τα εργαλεία ανάπτυξης είναι ιδιαίτερα εύχρηστα και μας επιτρέπουν να αναπτύξουμε πολύ γρήγορα κάθε είδους εφαρμογή. Μπορούμε επίσης να αναπτύξουμε εφαρμογές χωρίς να πρέπει να είμαστε άριστοι γνώστες του συστήματος ανάπτυξης εφαρμογών κι αυτό γιατί οι περισσότερες λειτουργίες του έχουν «κρυφτεί κάτω» από γραφικά περιβάλλοντα (graphical user interfaces) που αυτοματοποιούν την ανάπτυξη μέσω εύχρηστων μενού επίλογων.

Οι αισθητήρες που κυκλοφορούν στο εμπόριο καλύπτουν ένα ευρύ φάσμα μετρήσιμων ποσοτήτων. Η τάση που επικρατεί στις «νέες τεχνολογίες» υποτάσσει σμίκρυνση και πρακτικότητα

γεγονός που ευνόησε την ανάπτυξη καλύτερων αισθητήρων. Το κόστος τους και τα μεγέθη τους έχουν μειωθεί σημαντικά αλλά χωρίς να θυσιάσει η λειτουργικότητα. Με άμεσο αποτέλεσμα ακριβείς αισθητήρες με αξιόσέβαστες δυνατότητες διασύνδεσης (interfacing).

Οι τεχνολογίες που μας επιτρέπουν ασύρματη διαδίκτυωση έχουν γίνει ιδιαίτερα προσιτές και οικονομικές. Ο ασύρματος εξοπλισμός που υπάρχει στο εμπόριο είναι αποδοτικός και εύκολος στην χρήση. Ένα σημαντικό άλλο γεγονός όμως παραμένει, το ευρύ φάσμα εφαρμογών που το παρέχουν οι ασύρματες τεχνολογίες ακόμη δεν έχει κορέσει.

## 1.2. Σκοπός και Στόχοι Εργασίας

Οι στόχοι της εργασίας είναι οι εξής:

- Μελέτη και σύνθεση ενός τηλεμετρικού υποσυστήματος που να ενσωματώνει αποδοτικά τους αισθητήρες και τις αναπτυξιακές πλακέτες.
- Βαθμονόμηση (calibration) και ρύθμιση των αισθητήρων ώστε να παράγουν αξιόπιστες μετρήσεις αλλά και να διασυνδέονται άμεσα με το υπόλοιπο σύστημα.
- Συγγραφή κώδικα στο Arduino IDE, που θα μας παρέχει την δυνατότητα να καταγράφουμε και να μελετήσουμε τις συνθήκες περιβάλλοντος και να τις παρουσιάσουμε μέσω μιας ιστοσελίδας.
- Εξαγωγή συμπερασμάτων που αφορούν την πιστότητα των μετρήσεων και την βελτιστοποίηση του συστήματος.

## 1.3. Δομή Εργασίας

Η υπόλοιπη εργασία οργανώνεται ως εξής:

- Στο κεφάλαιο 2 δίνεται μια γενική εισαγωγή στην τηλεμετρία.
- Στο κεφάλαιο 3 παρουσιάζονται σύντομα οι μικροελεγκτές και οι βασικές τους ιδιότητες αλλά και τα είδη που υπάρχουν.
- Στο κεφάλαιο 4 παρουσιάζουμε την αναπτυξιακή πλακέτα Arduino.
- Στο κεφάλαιο 5 δίνεται μια σύντομη εισαγωγή στους αισθητήρες.
- Στο κεφάλαιο 6 μελετάμε τις δυνατότητες των διαθέσιμων αισθητήρων για τους περιβαλλοντικούς παράγοντες θερμοκρασία και σχετική υγρασία.
- Στο κεφάλαιο 7 συσχετίζουμε έννοιες όπως αβεβαιότητα και σφάλμα μέτρησης με την ακρίβεια των τιμών μέτρησης που παράγουν οι αισθητήρες.
- Στο κεφάλαιο 8 δίνεται μια σύντομη εισαγωγή στα ασύρματα πρωτόκολλα 802.11 και 802.11b, όπως επίσης παρουσιάζουμε την αναπτυξιακή πλακέτα WiShield.
- Στο κεφάλαιο 9 παρουσιάζουμε και αναλύουμε την υλοποίηση του τηλεμετρικού υποσυστήματος.
- Στο κεφάλαιο 10 εξάγουμε συμπεράσματα για την υλοποίηση και παραθέτουμε τις δυνατότητες μελλοντικής εξέλιξης της.



## 2. Γενική εισαγωγή στην τηλεμετρία

**Ορισμός:** Η λέξη προέρχεται από τις ελληνικές λέξεις *Τηλέ* = απομακρυσμένος και *Μέτρον* = μέτρο

*«Αρχικά τα συστήματα τηλεμετρίας χρησιμοποιούσαν ενσύρματη μετάδοση»*

### 2.1. Απαρχές και η ανάπτυξη εφαρμογών τηλεμετρίας

Τα αρχικά συστήματα τηλεμετρίας εισήχθησαν στις αρχές του 20ου αιώνα, χρησιμοποιούνταν για την εποπτεία φυσικών φαινομένων, αργότερα θα μπορέσουν να χρησιμοποιηθούν για την παρακολούθηση της διανομής του ηλεκτρικού ρεύματος. Στα πρώτα συστήματα ένα κέντρο ελέγχου χρησιμοποιούσε τηλεφωνικές γραμμές για να λάβει δεδομένα από απομακρυσμένες μονάδες παραγωγής ενέργειας. Έπειτα σε άλλους τομείς άρχισαν να εφαρμόζονται τα εν λόγω συστήματα, με βελτιώσεις που πραγματοποιούνται κατά τη διάρκεια των δεκαετιών που ακολούθησαν.

Την δεκαετία του 1930, μπαλόνια θα χρησιμοποιηθούν για τη συλλογή στοιχείων σχετικά με τις επικρατούσες ατμοσφαιρικές συνθήκες. Το 1950 δορυφόροι τίθενται σε τροχιά και χρησιμοποιούν την αρχή τηλεμετρίας για διάφορες εφαρμογές που περιλαμβάνουν την καταγραφή καιρικών συνθηκών και παρατήρηση φαινομένων. Οι εν λόγω δορυφόροι υπάρχουν και σήμερα και αρκετές εκατοντάδες από αυτούς βρίσκονται σε τροχιά γύρω από τη Γη.

### 2.2. Αρχή λειτουργίας

Τηλεμετρία είναι η εξ αποστάσεως μέτρηση και η απομακρυσμένη συλλογή δεδομένων που μπορούν να είναι φυσικής, περιβαλλοντικής ή βιολογικής σημασίας. Χρησιμοποιείται συνήθως για να συλλέγουμε δεδομένα από μακρινές, απρόσιτες περιοχές, ή όταν η συλλογή δεδομένων θα ήταν επικίνδυνο ή δύσκολο. Τα συστήματα επίσης που χρειάζονται εξωτερική καθοδήγηση για να λειτουργούν, απαιτούν την τηλεμετρία. Εξειδικευμένα όργανα εκτελούν τις μετρήσεις των φυσικών ποσοτήτων, και τις αποθηκεύουν ή τις μεταδίδουν μέσω σήματος που προκύπτει μετά από κάποια αρχική επεξεργασία σήματος ή μετατροπής.

Αν και ο όρος συνήθως αναφέρεται σε ασύρματους μηχανισμούς μεταφοράς δεδομένων (π.χ. χρησιμοποιώντας ραδιόφωνο, υπέρηχους ή υπέρυθρες ακτίνες), περιλαμβάνει επίσης τα δεδομένα που μεταφέρονται από άλλα μέσα, όπως τηλέφωνο, δίκτυο υπολογιστών ή άλλες ενσύρματες επικοινωνίες.

Ένα σύστημα τηλεμετρίας αποτελείται συνήθως από έναν αισθητήρα, όπως μια συσκευή εισόδου, ένα μέσο μετάδοσης με τη μορφή των ενσύρματων γραμμών ή ραδιοκύματα, συσκευές επεξεργασίας σήματος, καθώς και συσκευές για την εγγραφή ή εμφάνιση δεδομένων. Το αισθητήριο μετατρέπει μια φυσική ποσότητα, όπως η θερμοκρασία, σε ένα αντίστοιχο ηλεκτρικό σήμα, το οποίο στη συνέχεια μεταδίδεται με σκοπό την μέτρηση και καταγραφή.



Εικόνα 2. 1: Παραδείγματα τηλεμετρίας

### **2.3. Οφέλη & Εφαρμογές**

Τηλεμετρία είναι μια ιδιαίτερα αυτοματοποιημένη τεχνική επικοινωνίας, εξοικονομεί χρόνο, μειώνει γενικά τα έξοδα, παρέχει επιπλέον δυνατότητες μεταπώλησης και έχει ταχεία απόσβεση. Το χαμηλό κόστος ιδιοκτησίας σε συνδυασμό με την ελάχιστη τεχνογνωσία που απαιτεί εξασφαλίζει μια οικονομικώς αποδοτική και αξιόπιστη λύση.

Από τις πιο σημαντικές χρήσεις της τηλεμετρίας συμπεριλαμβάνουν την συλλογή δεδομένων καιρού, μονάδες παραγωγής ενέργειας, επανδρωμένων ή μη διαστημικών πτήσεων.

### **2.4. Άλλες εφαρμογές επιγραμματακά**

- Εξερεύνηση του διαστήματος
- Μηχανοκίνητοι αγώνες
- Γεωργία
- Διαχείριση υδάτων
- Διερεύνηση εδαφικών πόρων και άμυνας
- Τεχνολογία πυραύλων
- Πτητικές δοκιμές
- Στρατιωτικές πληροφορίες
- Παρακολούθηση πηγών ενέργειας
- Κατανομή & εκμετάλλευση πόρων
- Ιατρική
- Έρευνα και διαχείριση των προϊόντων αλιείας
- Εμπόριο
- Επιβολή του νόμου
- Πάροχοι ενέργειας
- Δοκιμές παντός τύπου



## 3. Εισαγωγή στους μικροελεγκτές

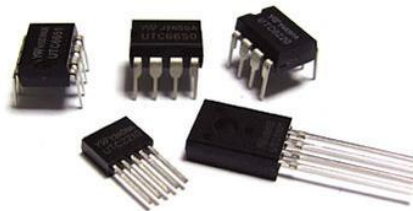
### 3.1. Ψηφιακά ολοκληρωμένα κυκλώματα

Τα ψηφιακά κυκλώματα κατασκευάζονται με ολοκληρωμένα κυκλώματα. Στην ηλεκτρονική, ένα ολοκληρωμένο κύκλωμα (**Intergrated circuit** ή **IC**, γνωστό και ως ψηφίδα πυριτίου, μικροτσίπ, τσιπ πυριτίου, τσιπ Η/Υ και μικροτσίπ ή Ο.Κ) είναι ένα **ηλεκτρονικό κύκλωμα<sup>1</sup> σε σμίκρυνση** (αποτελούμενο κυρίως από ημιαγώγιμα<sup>2</sup>, καθώς και παθητικά στοιχεία όπως τρανζίστορ, δίοδοι, αντιστάσεις) κατασκευασμένο **στην επιφάνεια** του λεπτού υποστρώματος ενός **ημιαγώγιμου υλικού**. Το chip τοποθετείται σε ένα κεραμικό ή πλαστικό περιβλήμα, ενώνονται και συγκολλούνται οι επαφές του chip με εξωτερικούς ακροδέκτες «pins» για να σχηματιστεί το ολοκληρωμένο κύκλωμα.

Σύμφωνα με τη διαδικασία κατασκευής και τη δομή τους, διακρίνουμε δύο βασικούς τύπους ολοκληρωμένων κυκλωμάτων που είναι ριζικά διαφορετικά μεταξύ τους: τα **μονολιθικά** και τα **υβριδικά** ολοκληρωμένα κυκλώματα.

### 3.2. Μονολιθικά Ο.Κ

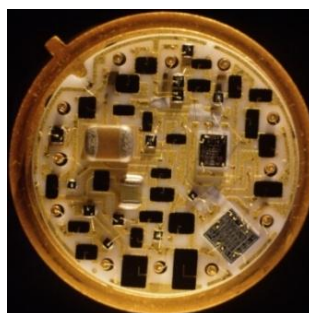
Είναι μικροκυκλώματα των οποίων τα στοιχεία έχουν πραγματοποιηθεί μέσα σε μια **στιβάδα γειτονική** στην επιφάνεια ενός **ημιαγώγιμου υποστρώματος**. Τα μονολιθικά ολοκληρωμένα κυκλώματα είναι η βάση της μοντέρνας μικροηλεκτρονικής.



Εικόνα 3. 1: Μονολιθικά ολοκληρωμένα κυκλώματα σε διάφορα πλαστικά περιβλήματα

### 3.3. Υβριδικά Ο.Κ

Ένα υβριδικό ολοκληρωμένο κύκλωμα είναι ένα ηλεκτρικό κύκλωμα σε σμίκρυνση που αποτελείται από **μεμονωμένα ημιαγώγιμα και παθητικά στοιχεία**, με τη μορφή διαφόρων **υμενίων** που αποτίθενται πάνω στην επιφάνεια ενός μονωτικού υποστρώματος ή σε έναν πίνακα κυκλωμάτων.



Εικόνα 3. 2: Ένα υβριδικό ολοκληρωμένο κύκλωμα

<sup>1</sup> αποτελείται από επιμέρους ηλεκτρονικά εξαρτήματα όπως, αντιστάσεις, τρανζίστορ, πυκνωτές, πηνία και διόδους, που συνδέονται με αγωγικά σύρματα μέσω των οποίων το ηλεκτρικό ρεύμα μπορεί να ρέει. Ο συνδυασμός τους επιτρέπει διάφορες λειτουργίες να εκτελούνται.

<sup>2</sup> ένα υλικό με ηλεκτρική αγωγιμότητα μεταξύ ενός αγωγού και μονωτικού υλικού.

### 3.4. Χρήσιμα χαρακτηριστικά των Ο.Κ

Επειδή το ολοκληρωμένο κύκλωμα, αντιπροσωπεύει από κατασκευαστική άποψη ένα σύνολο, το οποίο εκπληρώνει μια καθορισμένη λειτουργία και ικανοποιεί καθορισμένες απαιτήσεις, πρέπει να τοποθετηθεί στην κατηγορία των ηλεκτρονικών διατάξεων. Εν τούτοις, αν συγκριθεί με ένα τρανζίστορ ή με μία δίοδο κ.τ.λ., αποτελεί έναν τύπο διάταξης ριζικά καινούριο.

- Το πρώτο και κύριο χαρακτηριστικό του ολοκληρωμένου κυκλώματος είναι ότι **πραγματοποιεί, με ανεξάρτητο τρόπο, μια καθορισμένη λειτουργία**, συχνά πολύ πολύπλοκη, ενώ οι στοιχειώδεις ηλεκτρονικές διατάξεις (π.χ. αντιστάτες) δεν είναι ικανές να εκπληρώσουν μια ανάλογη λειτουργία παρά μόνο σε συνεργασία με άλλα ηλεκτρονικά εξαρτήματα.
- Ένα δεύτερο χαρακτηριστικό των ολοκληρωμένων κυκλωμάτων είναι ότι η **αύξηση της λειτουργικής πολυπλοκότητάς τους σε σχέση με τις θεμελιώδεις διατάξεις** δεν ακολουθείται από υποβάθμιση μιας οποιασδήποτε από τις κύριες παραμέτρους (πιστότητα, κόστος κτλ.). Αντίθετα όλες αυτές οι παράμετροι βελτιωμένες στα ολοκληρωμένα κυκλώματα.
- Ένα τρίτο χαρακτηριστικό των ολοκληρωμένων κυκλωμάτων είναι ότι έχουν **προτίμηση στα ενεργά στοιχεία** παρά στα παθητικά, στοιχειά δηλαδή που παράγουν ενέργεια (ενεργητικά π.χ. τρανζίστορ), έναντι αυτών που την αποθηκεύουν (παθητικά π.χ. αντιστάτες). Έτσι, ενδιαφερόμαστε να τοποθετήσουμε στο ίδιο τσιπ έναν, όσο το δυνατόν μεγαλύτερο, αριθμό στοιχείων ελάχιστης επιφάνειας. Η ελάχιστη επιφάνεια είναι χαρακτηριστικό των ενεργών στοιχείων
- Ένα τέταρτο χαρακτηριστικό των ολοκληρωμένων κυκλωμάτων συνδέεται με το γεγονός ότι **τα γειτονικά στοιχεία δεν απέχουν μεταξύ τους πάνω από 50 με 100μm** και επωφελούμαστε μειώνοντας την επίδραση των θερμικών μεταβολών και της διασποράς των παραμέτρων.

### 3.5. Ταξινόμηση των Ο.Κ βάσει βαθμού ολοκλήρωσης

*«Η λειτουργική πολυπλοκότητα των ολοκληρωμένων κυκλωμάτων χαρακτηρίζεται, κατά συνθήκη, από τον βαθμό ολοκλήρωσης δηλ. από τον αριθμό στοιχείων (συνήθως τρανζίστορ) που περιέχονται μέσα σε ένα τσιπ.»*

- Μικρή κλίμακα ολοκλήρωσης ή **SSI** (Small Scale Integration): περιέχουν έως 10 τρανζίστορ ή λίγες πύλες σε ένα ενιαίο πλαστικό περίβλημα, πχ AND, OR, NOT πύλες.
- Μεσαίας κλίμακα ολοκλήρωσης ή **MSI** (Medium Scale Integration): μεταξύ 10 και 100 τρανζίστορ ή δεκάδες πύλες σε ένα ενιαίο πακέτο πλαστικό περίβλημα. Εκτελούν ψηφιακές λειτουργίες, όπως πχ αθροιστές, αποκωδικοποιητές, μετρητές, flip-flops και πολυπλέκτες.
- Μεγάλης κλίμακα ολοκλήρωσης ή **LSI** (Large Scale Integration): μεταξύ 100 και 1.000 τρανζίστορ ή εκατοντάδες από τις πύλες. Πραγματοποιούν ειδικές ψηφιακές λειτουργίες όπως πχ την αριθμητική και λογική μονάδα, I / O συστήματα, μνήμη.
- Πολύ μεγάλη κλίμακα ολοκλήρωσης ή **VLSI** (Very-Large Scale Integration): μεταξύ 1.000 και 10.000 τρανζίστορ ή χιλιάδες πύλες. Ικανές για υπολογιστικές εργασίες, όπως επεξεργαστές, μεγάλες συστοιχίες μνήμης και προγραμματιζόμενες διατάξεις λογικής.
- Πάρα πολύ μεγάλη κλίμακα ολοκλήρωσης ή **SLSI** (Super-Large Scale Integration): μεταξύ 10.000 και 100.000 τρανζίστορ σε ένα ενιαίο πλαστικό περίβλημα. Μεγαλύτερων απαιτήσεων υπολογιστικές εργασίες όπως πχ, τα τσιπ μικροεπεξεργαστή PC, μικροελεγκτές και αριθμομηχανές.



- Εξαιρετικά μεγάλη κλίμακα ολοκλήρωσης ή **ULSI** (Ultra-Large Scale Integration): πάνω από 1 εκατομμύριο τρανζίστορ. Χρησιμοποιούνται σε CPU υπολογιστών, GPU, επεξεργαστές βίντεο, μικροελεγκτές, FPGA και πολύπλοκες PIC.

Εκτός από το βαθμό ολοκλήρωσης χρησιμοποιούμε και έναν άλλο όρο, την **πυκνότητα στοιχείων**, δηλ. τον αριθμό των στοιχείων (συνήθως τρανζίστορ) ανά μονάδα επιφάνειας του τσιπ. Αυτό το μέγεθος χαρακτηρίζει κυρίως τη στάθμη της τεχνολογίας.

### 3.6. Μικροεπεξεργαστής

*«Ίσως η σπουδαιότερη εφαρμογή ολοκληρωμένων κυκλωμάτων είναι ο μικροεπεξεργαστής»*

Ένας **μικροεπεξεργαστής** («μικρο-» λόγω της απόστασης των τρανζίστορ εντός του IC μετρούμενη σε  $\mu\text{m}$  ή  $1 \times 10^{-6} \text{ m}$ ) περιλαμβάνει τις περισσότερες ή όλες τις λειτουργίες μιας κεντρικής μονάδας επεξεργασίας (CPU) ενός H/Y σε ένα ενιαίο ολοκληρωμένο κύκλωμα. Ένας σύγχρονος μικροεπεξεργαστής αποτελείται από τις ακόλουθες μονάδες.

- **Μονάδα αποκωδικοποίησης εντολών** (Instruction Decoding Unit): Μονάδα που μετατρέπει τα προγράμματα σε εντολές Assembly.
- **Αριθμητική και Λογική Μονάδα** (Arithmetic and Logical Unit, ALU): Η μονάδα στην οποία εκτελούνται μία προς μία οι αριθμητικές ή λογικές πράξεις, όπως υπαγορεύονται από τις εντολές που έχουν δοθεί στον H/Y.
- **Καταχωρητές** (Registers): Μικρά κελιά μνήμης στο εσωτερικό του επεξεργαστή, που χρησιμοποιούνται για την προσωρινή αποθήκευση των δεδομένων, καθώς αυτά υφίστανται επεξεργασία. Οι καταχωρητές διαφέρουν ανάλογα με τον τύπο του επεξεργαστή και τον κατασκευαστή, τόσο ως προς την οργάνωση όσο και ως προς τη χωρητικότητά τους.
- **Μονάδα ελέγχου** (Control Unit): Ελέγχει τη ροή δεδομένων από και προς την ALU, τους καταχωρητές, τη μνήμη και τις περιφερειακές μονάδες εισόδου/εξόδου.
- **Μονάδα προσκόμισης** (Fetch Unit): Μεταφέρει τις εντολές από τη μνήμη στον επεξεργαστή.
- **Μονάδα προστασίας** (Protection Unit): Εξασφαλίζει το αποδεκτό της κάθε διεργασίας που εκτελεί ο επεξεργαστής, ώστε να μη τροποποιούνται δεδομένα που δεν πρέπει ή να μην εκτελούνται μη αποδεκτές εντολές, όπως π.χ. διαίρεση αριθμού με το μηδέν.

### 3.7. Μικροελεγκτής

Ο μικροελεγκτής είναι ένα πλήρες υπολογιστικό σύστημα βελτιστοποιημένο για τον έλεγχο hardware (υλικού) και ενσωματώνει μικροεπεξεργαστή, μνήμη και I/O περιφερειακές μονάδες σε μια μονό **ψηφίδα πυριτίου**. Η ύπαρξη των παραπάνω σε μια και μόνο ψηφίδα πυριτίου σημαίνει ότι η ταχύτητα ενισχύεται, διότι οι I/O περιφερειακές μονάδες απαιτούν λιγότερο χρόνο να διαβάσουν ή να γράψουν από τις εξωτερικές συσκευές. Επίσης ο επεξεργαστής και η μνήμη ανταλλάσσουν δεδομένα γρηγορότερα.

Όμως ένας μικροεπεξεργαστής είναι συνήθως βελτιστοποιημένος για να συντονίζει την ροή των δεδομένων μεταξύ των μονάδων μνήμης και των περιφερικών συσκευών εκτός του περιβάλλοντος του (ψηφίδα πυριτίου). Οι συνδέσεις ενός μικροεπεξεργαστή περιλαμβάνουν τη διευθυνσιοδότηση και τους διαύλους δεδομένων, που του επιτρέπουν να επιλέξει ένα από τα περιφερειακά του και να στείλει ή να ανακτήσετε δεδομένα από αυτά. Επειδή ο επεξεργαστής του μικροελεγκτή και τα περιφερειακά του είναι ενσωματωμένα στην ίδια ψηφίδα πυριτίου, οι μονάδες που περιέχει είναι αυτόνομες και σπάνια έχουν δομές διαύλων που εκτείνονται έξω από τα αυτούς.

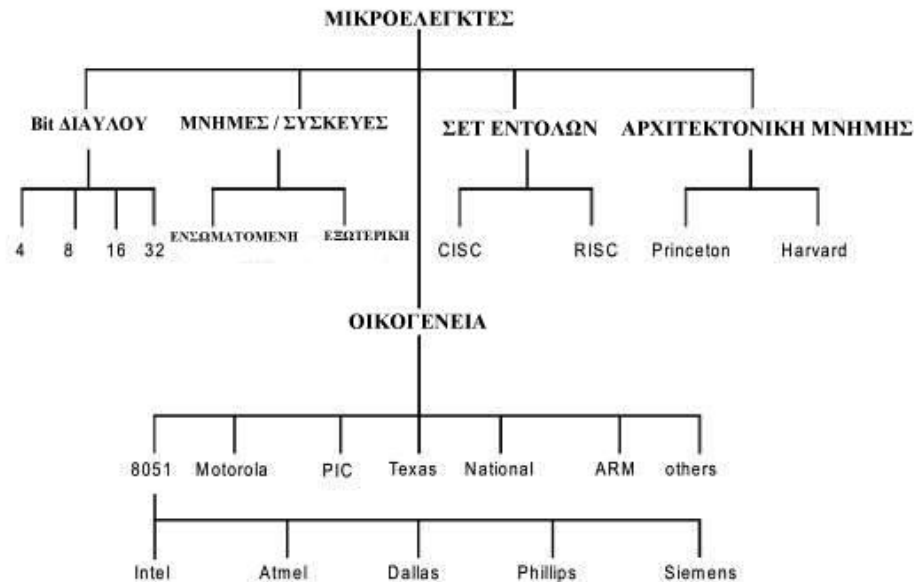
### 3.8. Μικροελεγκτές σε σύγκριση με τους μικροεπεξεργαστές

Ο μικροελεγκτής διαφέρει από έναν μικροεπεξεργαστή σε πολλά σημεία. Πρώτο και το πιο σημαντικό είναι η λειτουργικότητά του. Για έναν μικροεπεξεργαστή, θα πρέπει να του προστεθούν επιπλέον μονάδες όπως π.χ. μνήμη, για να είναι σε θέση για λήψη ή αποστολή δεδομένων. Με λίγα λόγια ο μικροεπεξεργαστής είναι η καρδιά του H/Y. Από την άλλη, ο μικροελεγκτής έχει σχεδιαστεί

για να είναι όλα αυτά σε ένα. Δεν απαιτούνται άλλα εξωτερικά εξαρτήματα για την πλήρη χρήση του, επειδή όλα τα απαραίτητα περιφερειακά ήδη ενσωματωμένα σε αυτόν.

### 3.9. Είδη μικροελεγκτών

Μικροελεγκτές μπορούν να ταξινομηθούν με βάση το εύρος των διαύλων (σε bit), την αρχιτεκτονική, τη μνήμη και το σετ εντολών.



Εικόνα 3. 3: Ταξινόμηση μικροελεγκτών

#### 3.9.1. Μικροελεγκτές 8, 16 και 32 bit

Όταν η μονάδα ALU εκτελεί αριθμητικές και λογικές λειτουργίες μιας εντολής, σε έναν 8-bit διάυλο, ο μικροελεγκτής είναι ένα 8-bit. Ομοίως ισχύει για τους μικροελεγκτές 16 και 32 bit.

#### 3.9.2. Ενσωματωμένος μικροελεγκτής

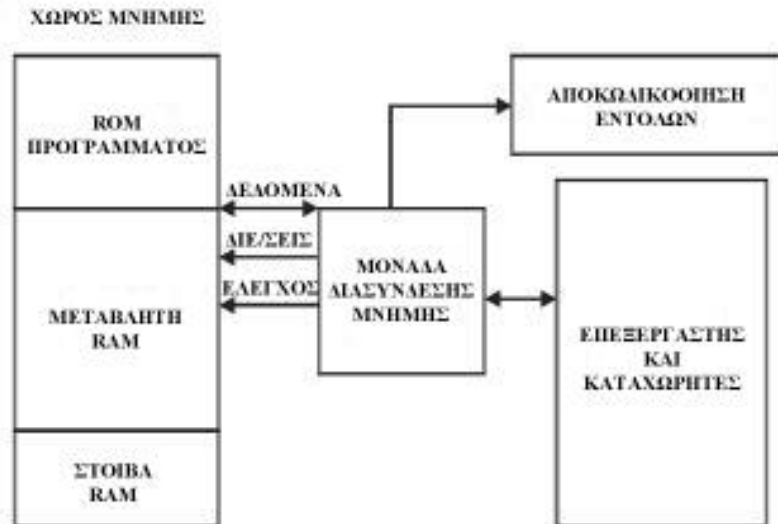
Όταν ένα σύστημα διαθέτει μονάδα μικροεπεξεργαστή η οποία έχει όλα τα λειτουργικά τμήματα (συμπεριλαμβανομένων μνήμη προγράμματος καθώς και δεδομένων) που είναι τοποθετημένα σε ένα τσιπ, το σύστημα αυτό ονομάζεται ενσωματωμένος μικροελεγκτής.

#### 3.9.3. Μικροελεγκτής εξωτερικής μνήμης

Όταν ένα ενσωματωμένο σύστημα διαθέτει μονάδα μικροελεγκτή που δεν έχει όλα τα λειτουργικά τμήματα τοποθετημένα σε ένα τσιπ ονομάζεται μικροελεγκτής εξωτερικής μνήμης. Το σύνολο ή μέρος των μονάδων μνήμης είναι διασυνδεδεμένα εξωτερικά χρησιμοποιώντας ένα κύκλωμα διασύνδεσης που ονομάζεται glue circuit (κύκλωμα συγκόλλησης).

#### 3.9.4. Αρχιτεκτονική Von-Neuman (ή Princeton)

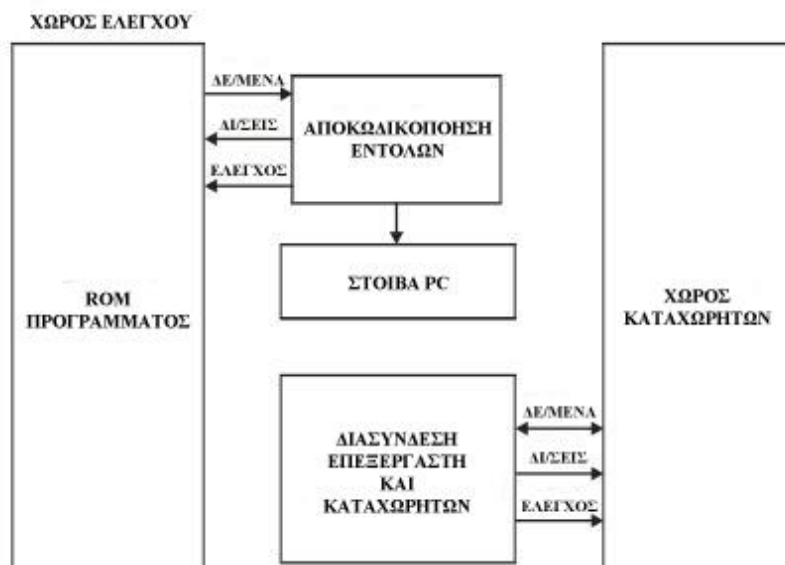
Μικροελεγκτές με βάση την Von-Neuman αρχιτεκτονική έχουν μόνο έναν διάυλο για «δεδομένα», που χρησιμοποιείται για να μεταφέρει και εντολές και δεδομένα. Οι εντολές του προγράμματος και τα δεδομένα είναι αποθηκευμένα σε κοινή μνήμη. Όταν ο μικροελεγκτής χρησιμοποιεί την κύρια μνήμη, πρώτα εκτελεί αυτή την εντολή, και στη συνέχεια ανακαλεί τα δεδομένα για την εκτέλεση της εντολής. Οι λειτουργίες επειδή είναι ξεχωριστές επιβραδύνουν τη λειτουργία του μικροελεγκτή.



Εικόνα 3. 4: Μπλοκ διάγραμμα αρχιτεκτονικής Von-Neuman

### 3.9.5. Αρχιτεκτονική Harvard

Μικροελεγκτές με βάση την αρχιτεκτονική Harvard έχουν ξεχωριστό δίαυλο δεδομένων και δίαυλο εντολών. Αυτό επιτρέπει στις εντολές να εκτελούνται παράλληλα. Καθώς μια εντολή «προ-φέρνεται» (pre-fetch), εκτελείται στον δίαυλο δεδομένων. Μόλις η τρέχουσα εντολή εκτελεστεί, η επόμενη εντολή είναι έτοιμη προς εκτέλεση. Το pro-fetch θεωρητικά επιτρέπει την ταχύτερη εκτέλεση των εντολών σε σχέση με την Von-Neuman αρχιτεκτονική αλλά σε βάρος της πολυπλοκότητας (υψηλότερη). Η αρχιτεκτονική Harvard μπορεί να εκτελεί τις εντολές σε λιγότερους κύκλους εντολών (instruction cycles) από την Von-Neuman αρχιτεκτονική.



Εικόνα 3. 5: Μπλοκ διάγραμμα αρχιτεκτονικής Harvard

### 3.9.6. Αρχιτεκτονική CISC

Σχεδόν το σύνολο των μικροελεγκτών σήμερα βασίζονται στην τεχνολογία CISC (Complex Instruction Set Computer - Υπολογιστής Σύνθετου Σει Εντολών). Όταν ένα μικροελεγκτής διαθέτει ένα σετ εντολών που υποστηρίζει σύνθετες λειτουργίες για την εκτέλεση αριθμητικών και λογικών εντολών, μεταφοράς δεδομένων και πρόσβασης μνήμης, τότε λέγεται ότι είναι CISC αρχιτεκτονικής.

Τα πλεονεκτήματα της αρχιτεκτονικής CISC είναι ότι πολλές από τις εντολές της λειτουργούν ως μακροεντολές (macros), επιτρέποντας στον προγραμματιστή να χρησιμοποιήσει μια εντολή στη θέση πολλών απλούστερων.

### 3.9.7. Αρχιτεκτονική RISC

Η τάση της βιομηχανίας για το σχεδιασμό μικροεπεξεργαστών ή RISC (RISC Reduced Instruction Set Computers - Υπολογιστές Απλούστερου Σετ Εντολών). Όταν ένας μικροελεγκτής έχει ένα σετ εντολών που υποστηρίζει απλούστερους τρόπους εκτέλεσης αριθμητικών και λογικών εντολών και μεταφοράς δεδομένων, τότε είναι αρχιτεκτονικής RISC.

Τα οφέλη από την απλότητα του σχεδιασμού RISC είναι μικρότερα τσιπ, αισθητή μείωση ο αριθμού pin και πολύ χαμηλή κατανάλωση ενέργειας.

### 3.10. Βασική δομή του μικροελεγκτή

Για να είναι εμφανής η λειτουργία και η διασύνδεση των δομικών στοιχείων του μικροελεγκτή, θα αναλυθούν οι βασικές μονάδες του ξεχωριστά. Λεπτές γραμμές που οδηγούν από το κέντρο προς τις πλευρές του μικροελεγκτή αντιπροσωπεύουν καλώδια που συνδέουν εσωτερικές μονάδες με τις ακίδες στο περίβλημα του μικροελεγκτή τις λεγόμενες γραμμές συγκόλλησης.

Για την υλοποίηση μιας εφαρμογής, ένας μικροελεγκτής δεν αρκεί. Εκτός από ένα μικροελεγκτή, χρειαζόμαστε ένα πρόγραμμα που θα εκτελεστεί, καθώς και μερικά ακόμη στοιχεία που συνθέτουν μια λογική διασύνδεσης. Η **εικόνα 3.6** που ακολουθεί δείχνει ένα μικροελεγκτή στο εσωτερικό του.

Συνήθως ένας μικροελεγκτής ενσωματώνει πάνω στην ψηφίδα πυριτίου τα ακόλουθα:

- Την CPU (ΚΜΕ)
- Μονάδες Μνήμης
- I/O σε παράλληλη συνδεσμολογία

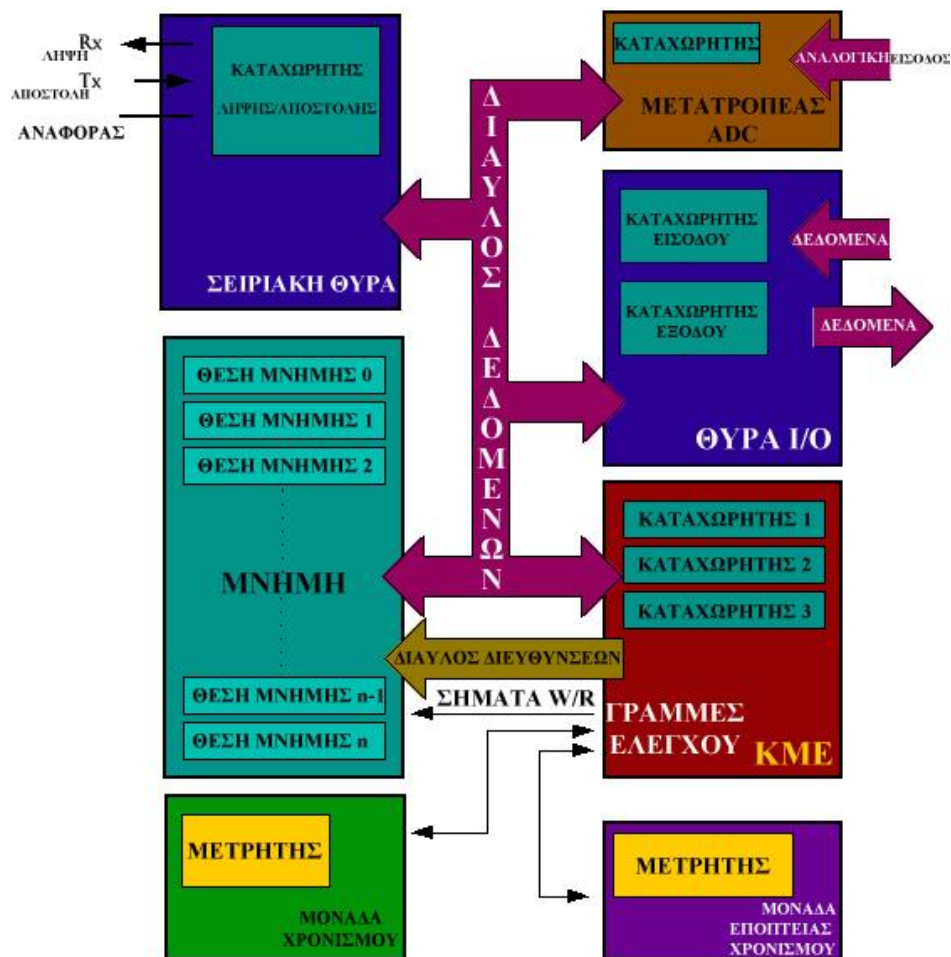


Εικόνα 3. 6: Θεμελιώδεις μονάδες του μικροελεγκτή

Ο μικροελεγκτής συνδυάζει και άλλες μονάδες , όπως:

- Μονάδα χρονισμού (Timer) που καθιστά δυνατό στον μικροελεγκτή την εκτέλεση εργασιών για συγκεκριμένες χρονικές περιόδους και στιγμές.
- Σειριακή θύρα I / O για να επιτρέψει τη ροή δεδομένων μεταξύ του μικροελεγκτή και άλλες συσκευές, όπως ένα PC ή ένα άλλο μικροελεγκτή.
- Μονάδα ADC (Analog-to-Digital Converter - Αναλογικό σε Ψηφιακό Μετατροπέα) ώστε να είναι δυνατό ο μικροελεγκτής να δέχεται είσοδο αναλογικών δεδομένων για επεξεργασία.

- Μονάδες μνήμης διαφόρων ειδών που του επιτρέπουν την αποθήκευση και ανάκληση δεδομένων.
- Διαύλους που επιτρέπουν την αποδοτική μεταφορά δεδομένων μεταξύ της CPU και των υπολοίπων μονάδων.
- Μονάδα εποπτείας χρονισμού (Watchdog Timer) που μας επιτρέπει να ελέγχουμε την σωστή εκτέλεση του προγράμματος.



Εικόνα 3. 7: Διάγραμμα μπλοκ μικροελεγκτή

### 3.11. Μονάδα μνήμης

Η μνήμη είναι μέρος του μικροελεγκτή της οποίας η λειτουργία είναι η αποθήκευση δεδομένων. Ο ευκολότερος τρόπος για να την παρουσιάσουμε είναι να την περιγράψουμε ως ένα μεγάλο ντουλάπι με πολλά συρτάρια. Αν υποθέσουμε ότι έχουμε κατασκευάσει τα συρτάρια με τέτοιο τρόπο ώστε να μην μπορούν να συγχέονται, κανένα από τα περιεχόμενά τους στη συνέχεια δεν θα είναι δύσκολο προσβάσιμο. «Αρκεί να είναι γνωστή η ονομασία του κάθε συρταριού και έτσι το περιεχόμενό του θα είναι γνωστό σε μας στα σίγουρα».

Τα στοιχεία της μονάδας μνήμης λειτουργούν ως εξής. Για μια συγκεκριμένη είσοδο παίρνουμε τα περιεχόμενα μιας θέσης μνήμης που έχει την δική της διεύθυνση. Δύο νέες έννοιες μας χρειάζονται: διεθυνσιοδότηση και θέση μνήμης. Η μνήμη αποτελείται από όλες τις πιθανές θέσεις της και η διεθυνσιοδότηση μνήμης δεν είναι τίποτα άλλο, από την επιλογή μιας θέσης από αυτήν.

Αυτό σημαίνει ότι για να επιλέξουμε την επιθυμητή θέση μνήμης, πρέπει να περιμένουμε για το περιεχόμενο αυτής της τοποθεσίας. Εκτός από την ανάγνωση μιας θέσης μνήμης, θα πρέπει επίσης να υπάρχει και η εγγραφή πάνω σε αυτήν τη θέση.

### 3.11.1. Τα βασικότερα είδη μνήμης:

- **ROM (μνήμη μόνο για ανάγνωση)**  
Η μνήμη μόνο για ανάγνωση (ROM Read Only Memory) είναι ένας τύπος μνήμης που χρησιμοποιείται για να αποθηκεύσει μόνιμα το πρόγραμμα που εκτελείται. Το μέγεθος του προγράμματος που μπορεί να αποθηκευθεί, εξαρτάται από το μέγεθος αυτής της μνήμης. Ανάλογα με τον τύπο του μικροελεγκτή η ROM μπορεί να ενσωματωθεί στο μικροελεγκτή ή να προστεθεί ως εξωτερικό τσιπ.
- **RAM (μνήμη τυχαίας προσπέλασης)**  
Η μνήμη τυχαίας προσπέλασης (RAM Random Access Memory) είναι ένας τύπος μνήμης που χρησιμοποιείται για την προσωρινή αποθήκευση δεδομένων που δημιουργούνται και χρησιμοποιούνται κατά τη διάρκεια της λειτουργίας των μικροελεγκτών. Το περιεχόμενο αυτής της μνήμης διαγράφεται όταν η παροχή ρεύματος διακόπτεται.
- **EEPROM (ηλεκτρικά διαγράψιμη και προγραμματίσιμη μνήμη μόνο για ανάγνωση)**  
Η ηλεκτρικά διαγράψιμη και προγραμματίσιμη μνήμη μόνο για ανάγνωση (EEPROM Electrically Erasable Programmable ROM) είναι ένας ειδικός τύπος μνήμης που δεν περιλαμβάνονται σε όλους τους μικροελεγκτές. Το περιεχόμενό της μπορεί να αλλάξει κατά τη διάρκεια της εκτέλεσης του προγράμματος (παρόμοια με την μνήμη RAM), αλλά παραμένει μόνιμα αποθηκευμένο, ακόμη και μετά την διακοπή της παροχής ρεύματος (παρόμοια με ROM). Συχνά χρησιμοποιείται για την αποθήκευση δεδομένων τα οποία πρέπει να αποθηκεύονται πριν την διακοπή της παροχής ρεύματος

### 3.12. Δίαυλος

Σε τεχνικό επίπεδο, αυτό αντιπροσωπεύεται από μια ομάδα 8, 16 ή περισσότερων καλωδίων. Υπάρχουν δύο τύποι διαύλων: διευθύνσεως και δεδομένων. Στον πρώτο αποτελείται από όσες γραμμές θέλουμε να για να χωρίσουμε σε διευθύνσεις την μνήμη, και στον άλλο όσο είναι το εύρος των δεδομένων, στην περίπτωσή μας 8 bits ή της μιας άλλης γραμμής σύνδεσης. Ο πρώτος χρησιμεύει για τη μεταφορά δεδομένων από μια διεύθυνση μνήμης της CPU, και ο δεύτερος για να συνδεθούν όλες οι υπόλοιπες μονάδες στο εσωτερικό του μικροελεγκτή.

Ο δίαυλος όσον αφορά τη λειτουργικότητα την βελτιώνει, αλλά ένα νέο πρόβλημα επίσης έχει εμφανιστεί: έχουμε μια μονάδα που είναι σε θέση να εργαστεί από μόνη της, αλλά δεν έχει καμία επαφή με τον έξω κόσμο, ή μαζί μας! Για την άρση αυτής της ανεπάρκειας, ας προσθέσουμε μια μονάδα η οποία περιέχει αρκετές θέσεις μνήμης εκ των οποίων το ένα άκρο είναι συνδεδεμένο με τον δίαυλο δεδομένων και στο άλλο να είναι συνδεδεμένο με τις γραμμές εξόδου του μικροελεγκτή.

### 3.13. Μονάδα εισόδου/εξόδου (I/O)

Αυτές οι θέσεις που έχουμε μόλις προστεθήκαν ονομάζονται «θύρες». Υπάρχουν διάφορων τύπων θύρες: εισόδου, εξόδου ή αμφίδρομες. Κατά την εργασία με θύρες, πρωτίστως είναι απαραίτητο να επιλεχτεί ποια θύρα θα χρησιμοποιηθεί, και στη συνέχεια να σταλούν δεδομένα σε αυτήν ή να πάρουμε δεδομένα απ' αυτή.

Όταν χρησιμοποιείται με αυτό τον τρόπο με μια θύρα αυτή λειτουργεί σαν μια θέση μνήμης. Δεδομένα εγγράφονται σε αυτήν ή διαβάζονται από αυτήν, το γεγονός αυτό παρατηρείται επάνω στις ακίδες του μικροελεγκτή.

### 3.14. Σειριακή επικοινωνία

Εκτός αυτών που έγιναν αναφορά παραπάνω πρέπει να προστεθεί μια μονάδα που να προσφέρει τη δυνατότητα επικοινωνίας με τον έξω κόσμο. Για να λειτουργήσει αυτό, πρέπει να θέσουμε τους κανόνες ανταλλαγής δεδομένων. Οι κανόνες αυτοί ονομάζονται πρωτόκολλο. Πρωτόκολλο είναι αυτό που ορίζεται εκ των προτέρων έτσι ώστε δεν θα υπάρξει καμία παρεξήγηση μεταξύ των πλευρών που επικοινωνούν μεταξύ τους.

Με βάση το πρωτόκολλο μπορεί κάθε bit να παραμείνει στην γραμμή για χρονικό διάστημα το οποίο είναι ίσο με T και στο τέλος, ή μετά το 8<sup>ο</sup> bit, να δοθεί η λογική ποσότητα «1» στη γραμμή η οποία θα σηματοδοτήσει το τέλος της μετάδοσης των δεδομένων. Το πρωτόκολλο αυτό ονομάζεται NRZ (Non-Return to Zero). Τα δεδομένα μετακινούνται από τη μνήμη μέσα από διάλυο προς την τοποθεσία που θα αποσταλούν και στη συνέχεια στη μονάδα που θα τα λάβει σύμφωνα με το πρωτόκολλο.

Πρέπει όμως να έχουμε ξεχωριστές γραμμές για τη λήψη και την αποστολή, για να είναι δυνατή η λήψη και αποστολή δεδομένων την ίδια στιγμή. Η λειτουργία full-duplex επιτρέπει αυτόν τον τρόπο επικοινωνίας και η μονάδα επικοινωνίας που μας το επιτρέπει ονομάζεται σειριακή μονάδα.

### 3.15. Μονάδα χρονισμού (Timer)

Δεδομένου ότι έχουμε εξηγήσει τη σειριακή επικοινωνία, μπορούμε να λάβουμε, να αποστείλουμε και να επεξεργαστούμε δεδομένα. Ωστόσο, προκειμένου να χρησιμοποιηθεί, χρειαζόμαστε μια επιπλέον μονάδα. Αυτή είναι η μονάδα χρονισμού η οποία είναι ιδιαίτερα σημαντική, διότι μπορεί να μας παρέχει πληροφορίες για το χρόνο και τη χρονική διάρκεια.

Το στοιχείο στη μονάδα χρονισμού είναι ένας μετρητής που «τρέχει συνεχώς» και στην πραγματικότητα είναι ένας καταχωρητής του οποίου η αριθμητική τιμή αυξάνεται κατά ένα σε ισόποσα διαστήματα, έτσι ώστε με τη ανάγνωση της τιμής του κατά τις περιόδους T1 και T2 και με βάση τη διαφορά τους να μπορούμε να καθορίσουμε πόσο χρονικό διάστημα έχει παρέλθει. Ένα πολύ σημαντικό μέρος της λειτουργίας του μικροελεγκτή.

### 3.16. Μονάδα μετατροπής από αναλογικό σε ψηφιακό (ADC)

Δεδομένου ότι τα σήματα των περιφερικών συνήθως διαφέρουν ουσιαστικά από αυτά που ο μικροελεγκτής μπορεί να καταλάβει (μηδέν και ένα), θα πρέπει να μετατραπούν σε ένα πρότυπο που μπορεί να «κατανοηθεί» από τον μικροελεγκτή. Το έργο αυτό εκτελείται από μια μονάδα που μετατρέπει την αναλογική τιμή σε ψηφιακή ή Analog-to-Digital Converter (μετατροπέα αναλογικό σε ψηφιακό) γνωστό και ως ADC. Αυτό η μονάδα είναι υπεύθυνη για τη μετατροπή δεδομένων που έχουν αναλογική τιμή σε δυαδικό αριθμό και στη συνέχεια να τα μεταφέρει από στην CPU ώστε η CPU να μπορεί να τα επεξεργαστεί περαιτέρω.

### 3.17. Μονάδα εποπτείας χρονισμού (Watchdog Timer)

Ένα ακόμα πράγμα που απαιτεί την προσοχή μας είναι η άψογη λειτουργία του μικροελεγκτή καθ' όλη τη διάρκεια του χρόνου λειτουργίας του. Ας υποθέσουμε ότι, ως αποτέλεσμα κάποιας παρέμβασης (το οποίο συχνά συμβαίνει) ο μικροελεγκτής μας σταματά την εκτέλεση του προγράμματος, ή χειρότερα, αρχίζει να μην λειτουργεί σωστά.

Φυσικά, όταν αυτό συμβαίνει με έναν H/Y, απλά «πατάμε το κουμπί reset» και θα συνεχίσουμε να δουλεύουμε. Ωστόσο, δεν υπάρχει κουμπί reset που μπορούμε να «πατήσουμε» στον μικροελεγκτή και έτσι να λυθεί το πρόβλημά μας. Για να ξεπεραστεί αυτό το εμπόδιο, θα πρέπει να εισαχθεί μια ακόμη μονάδα που ονομάζεται μονάδα εποπτείας χρονισμού. Αυτό η μονάδα είναι στην πραγματικότητα ένα άλλος μετρητής που «τρέχει συνεχώς», όπου το πρόγραμμα πρέπει να γράφει ένα μηδέν κάθε φορά που εκτελείται μια εντολή σωστά. Σε περίπτωση που το πρόγραμμα «κολλήσει», το μηδέν δεν θα γραφτεί και από μόνη της (η μονάδα εποπτείας χρονισμού) θα προκαλέσει «reset» στον μικροελεγκτή κατά την επίτευξη της μέγιστης τιμής της (μετά από πολλές αυξήσεις της τιμής του). Αυτό θα έχει ως αποτέλεσμα την εκτέλεση του προγράμματος και πάλι, και αυτή τη φορά

σωστά. Αυτό είναι ένα σημαντικό στοιχείο κάθε προγράμματος ώστε να είναι αξιόπιστο, χωρίς την εποπτεία του ανθρώπου.

### 3.18. Προγραμματισμός μικροελεγκτών

Ο προγραμματισμός μπορεί να γίνει σε αρκετές γλώσσες, όπως η Assembly<sup>3</sup>, Basic, C και C++ που είναι από τις πλέον διαδεδομένες γλώσσες. Η Assembly ανήκει σε γλώσσες χαμηλότερου επίπεδου όπου ο προγραμματισμός γίνεται με αργούς ρυθμούς, αλλά καταλαμβάνουν τον ελάχιστο χώρο στη μνήμη και δίνει τα καλύτερα αποτελέσματα, όπου η ταχύτητα της εκτέλεσης του προγράμματος είναι το ζητούμενο. Προγράμματα σε γλώσσα C++ είναι πιο εύκολο να αναπτυχθούν, πιο εύκολο να γίνουν κατανοητά διότι οι εντολές σε C++ είναι πλησιέστερες στην ανθρώπινη λογική, αλλά είναι πιο αργή η εκτέλεση τους από τα προγράμματα σε Assembly.

Μετά που θα αναπτύξουμε το πρόγραμμα, θα μεταφερθεί στον μικροελεγκτή και θα εκτελεστεί. Για να γίνει αυτό πρέπει να προσθέσουμε μερικά ακόμη εξωτερικά εξαρτήματα που απαιτούνται για αυτές τις εργασίες. Πρώτα πρέπει να δώσουμε ενέργεια στον μικροεπεξεργαστή με τη σύνδεσή του σε παροχή ρεύματος (ισχύς που απαιτείται για τη λειτουργία όλων των ηλεκτρονικών εξαρτημάτων) και ύστερα ο ταλαντωτής ο ρόλος του οποίου είναι παρόμοιος με το ρολό που η καρδιά παίζει σε ένα ανθρώπινο σώμα. Τα ρολόγια χρονισμού του ταλαντωτή (συχνότητα ταλάντωσης) με βάση τα οποία ο μικροελεγκτής εκτελεί τις εντολές του προγράμματος. Όσο δέχεται ο μικροελεγκτής εντολές, θα αναζητήσει την έναρξη του προγράμματος και θα αρχίσετε να το εκτελεί. Πώς το πρόγραμμα θα λειτουργήσει/αποδώσει εξαρτάται από πολλές παραμέτρους, η σημαντικότερη των οποίων είναι η δεξιοτεχνία και η εμπειρία του προγραμματιστή.

---

<sup>3</sup> μια γλώσσα προγραμματισμού πολύ κοντά στη γλώσσα μηχανής και στο υλικό του υπολογιστή.





## 4. Παρουσίαση της αναπτυξιακής πλακέτας Arduino

### 4.1. Τί είναι το Arduino;

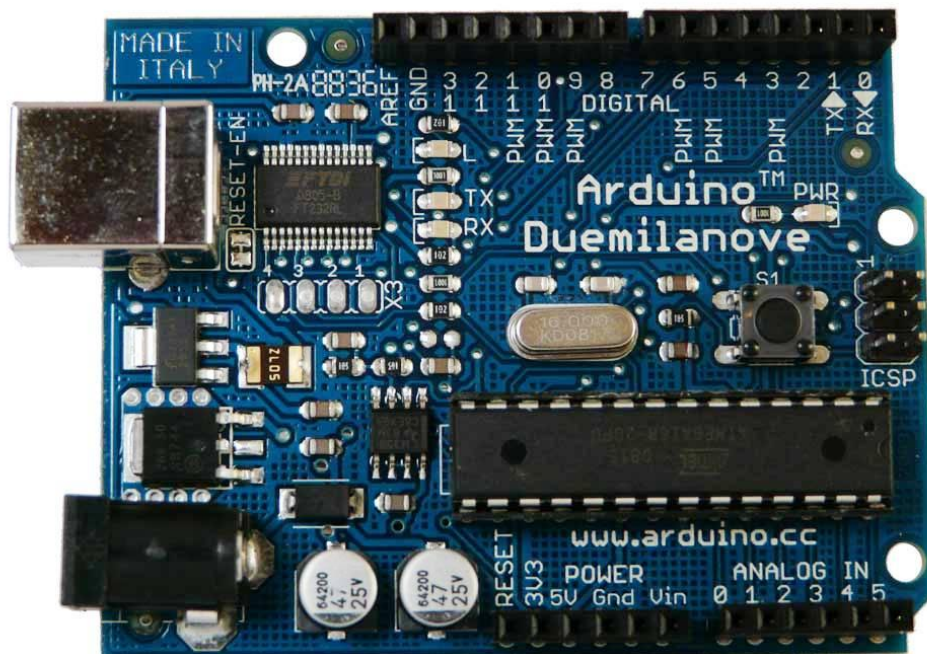
Το **Arduino** είναι αναπτυξιακή πλακέτα μικροελεγκτή, ανοιχτού τύπου υλικού μονής πλακέτας (open-source hardware single-board microcontroller) όπως και το λογισμικό ανάπτυξης εφαρμογών (IDE).

Είναι παράγωγο της αναπτυξιακής πλατφόρμας Wiring, που μας επιτρέπει να αναπτύξουμε ένα υπολογιστικό σύστημα το οποίο θα ελέγχει συσκευές του φυσικού κόσμου. Είναι ανοιχτού κώδικα το λογισμικό του περιβάλλοντος ανάπτυξης (open source IDE). Η αναπτυξιακή πλακέτα που τον αποτελεί, ενσωματώνει έναν μικροελεγκτή με βασικό στοιχείο τον επεξεργαστή Atmel AVR ATMega328P και I/O system (σύστημα εισόδου-εξόδου). Επίσης συνδέεται με H/Y μέσω θύρας USB για να το προγραμματίσουμε μέσα από ένα απλό περιβάλλον ανάπτυξης.

Το λογισμικό του περιβάλλοντος ανάπτυξης βασίζεται στην γλώσσα προγραμματισμού Wiring και αυτής που χρησιμοποιεί το αναπτυξιακό περιβάλλον του Processing, οι οποίες είναι ανοιχτού κώδικα (open source) και μπορεί κάποιος να τις «κατεβάσει δωρεάν».

Η γλώσσα προγραμματισμού του Arduino είναι παρόμοια με την C++ με κάποιες απλοποιήσεις και μετατροπές και αποτελεί μια εφαρμογή σε επίπεδο software της καλωδίωσης που συναντάται στο Arduino. Εξομοιώνει θα λέγαμε απόλυτα το hardware (υλικό) του μικροελεγκτή.

Το Arduino μπορεί να χρησιμοποιηθεί για να αναπτύξουμε διαδραστικά αυτοματοποιημένα αντικείμενα, ικανά να δεχτούν εισόδους από πληθώρα αισθητηρίων οργάνων και διακόπτες, αλλά και να ελέγχουν διάφορα φώτα, κινητήρες και άλλες συσκευές εξόδου του φυσικού κόσμου. Τα Projects στον εν λόγω μικροελεγκτή μπορούν να είναι αυτόνομα (σε επίπεδο hardware) ή να επικοινωνούν με κάποιο software σε κάποιο H/Y.



Εικόνα 4. 1: Κάτοψη του Arduino

## 4.2. Γιατί Arduino; (έναντι άλλων)

Υπάρχει πληθώρα μικροελεγκτών και αναπτυξιακών στο εμπόριο για να ασχοληθεί κάποιος και να παράγει εφαρμογές. Όλα τα εργαλεία που μας παρέχουν αυτές είναι απλοποιημένα καθώς «κρύβουν» τις δύσκολες λεπτομέρειες της αρχιτεκτονικής και επιτρέπουν τον άμεσο προγραμματισμό του μικροελεγκτή, ακόμη και στον αρχάριο χρήστη προσφέροντας τα πάντα σε ένα και μόνο «πακέτο «έτοιμο για χρήση». Το Arduino διαφέρει γιατί απλοποιεί ιδιαίτερα την διαδικασία ανάπτυξης εφαρμογών με μικροελεγκτές, και παρέχει επιπλέον πλεονεκτήματα που ενδείκνυνται για χρήση από δασκάλους, μαθητές, hobbyistes και τα οποία είναι:

- **Χαμηλό κόστος:** οι πλακέτες του Arduino είναι εξαιρετικά φθηνές σε σχέση με άλλες πλατφόρμες μικροελεγκτών. Ειδικά δε με τα schematics (σχηματικά) που βρίσκονται δωρεάν στο Internet μπορεί να κατασκευάσει οποιοσδήποτε την φθηνότερη εκδοχή ενός Arduino.
- **Τρέχει σε διάφορα λειτουργικά συστήματα:** το περιβάλλον προγραμματισμού του Arduino είναι διαθέσιμο για Windows, Macintosh OSX και για λειτουργικά συστήματα Linux. Τα περισσότερα συστήματα ανάπτυξης μικροελεγκτών περιορίζονται στα Windows.
- **Απλό, ξεκάθαρο προγραμματιστικό περιβάλλον:** το περιβάλλον προγραμματισμού ενός Arduino ενδείκνυται για αρχάριους, αλλά είναι ταυτόχρονα και ευέλικτο και για πιο προχωρημένους χρήστες.
- **Ανοιχτού λογισμικού και λογισμικού που επεκτείνεται και παραμετροποιείται:** το software του Arduino διανέμεται με την μορφή εργαλείων ανοιχτού λογισμικού και είναι διαθέσιμο προς επέκταση. Η γλώσσα προγραμματισμού του μπορεί να επεκταθεί διαμέσου βιβλιοθηκών της C++ και της AVR C που είναι για τον προγραμματισμό των Atmel μικροελεγκτών, γλώσσα στην οποία επίσης βασίστηκε το λογισμικό του Arduino. Ομοίως μπορεί να προστεθεί κώδικας AVR-C που έχει ήδη γραφτεί, σε πρόγραμμα για Arduino.
- **Ανοιχτού υλικού το οποίο μπορεί να επεκταθεί:** το Arduino βασίζεται στον μικροελεγκτή ATmega328P της Atmel. Τα σχηματικά για τα αναπτυξιακή πλακέτα βρίσκονται υπό την άδεια χρήσης Creative Commons, επιτρέποντας σε σχεδιαστές να κατασκευάσουν το δικό τους, εξελίσσοντας το ήδη υπάρχον χωρίς να έχουν νομικά προβλήματα.

## 4.3. Το hardware του Arduino

Ο μικροελεγκτής Arduino ενσωματώνει τον επεξεργαστή **ATMEL ATmega328P**, όπως επίσης κι όλα τα απαραίτητα εξαρτήματα για την τροφοδοσία και την διασύνδεση του μικροελεγκτή με H/Y. Τα οποία περιλαμβάνουν **υποσυστήματα μνήμης, εισόδου/εξόδου, γραμμικό ρυθμιστή τάσης, ταλαντωτής κρυστάλλου, bootloader** (μικροκώδικα εκκίνησης υλικού), **προσαρμογέα USB σε FTDI Serial**. Η πλακέτα μπορεί να τροφοδοτηθεί είτε με τροφοδοτικό των 9Volt, είτε απευθείας από την USB θύρα του H/Y.

## Τα χαρακτηριστικά εν συντομία:

Μικροελεγκτής	ATmega328P
Τάση Λειτουργίας	5V
Τάση Εισόδου	7-12V
Όρια Τάσης	6-20V
Ψηφιακοί Ακροδέκτες I/O	14 (εκ των οποίων 6 παρέχουν PWM <sup>4</sup> έξοδο)
Ψηφιακοί Ακροδέκτες Εισόδου	6
DC ρεύμα ανά I/O Ακροδέκτη	40 mA
DC ρεύμα για 3.3V Ακροδέκτη	50 mA
Μνήμη Flash <sup>5</sup>	32 KB ( 2 KB χρησιμοποιούνται από τον bootloader)
SRAM <sup>6</sup>	2 KB
EEPROM	1KB
Ταχύτητα Ρολογιού	16 MHz

Πίνακας 1: Χαρακτηριστικά Arduino.

## 4.4. Οι ακροδέκτες του Arduino

Ο Arduino έχει **14 ψηφιακούς ακροδέκτες** (pin). Αυτοί μπορούν να τεθούν ως είσοδοι ή ως έξοδοι. Λειτουργούν στα 5 Volt και έχουν την δυνατότητα να παρέχουν ή να δέχονται ένταση της τάξεως των 40mA. Σε κάθε pin υπάρχει εσωτερικά ένας pullup<sup>7</sup> αντιστάτης στα 20-50KΩ. Επιπλέον έχει **5 αναλογικούς ακροδέκτες** εισόδου. Αυτοί μπορούν να «διαβάσουν» αναλογικές τιμές όπως η τάση μιας μπαταρίας και να τις μετατρέψουν σε μια τιμή από το 0 έως το 1023 (10 bit). Η μέτρηση της τάσης γίνεται από προκαθορισμένα επίπεδα από 0 έως 5 Volt. Εκτός αυτών, 6 εκ των 14 ψηφιακών ακροδεκτών οι 3, 5, 6, 9, 10 και 11 έχουν την δυνατότητα να προγραμματιστούν ώστε να λειτουργούν ως αναλογικές έξοδοι.

### 4.4.1. Κάποιοι ακροδέκτες έχουν συγκεκριμένες λειτουργίες

- **Σειριακή Λειτουργία:** 0 (RX) and 1 (TX). Χρησιμοποιούνται για λήψη (RX) και εκπομπή (TX) TTL σειριακών δεδομένων. Αυτοί οι ακροδέκτες είναι συνδεδεμένοι με τους αντίστοιχους του μετατροπέα FTDI USB σε TTL.
- **Εξωτερικές Διακοπές (interrupts):** 2 και 3. Αυτοί οι ακροδέκτες μπορούν να ενεργοποιούν διακοπές αν ανιχνευθεί παλμός χαμηλής τάσης.

<sup>4</sup> τεχνική μετατροπής αναλογικών σημάτων σε ψηφιακά μέσω χρήσης τετραγωνικών παλμών που είναι κατάλληλα διαμορφωμένοι.

<sup>5</sup> μη πτητικό μέσο αποθήκευσης που αναπτύχθηκε από την EEPROM και μπορεί να διαγραφεί και να αναπρογραμματιστεί ηλεκτρικά όπως και η EEPROM.

<sup>6</sup> είδος μνήμης ημιαγωγών, όπου η λέξη static (στατική) δείχνει ότι δεν χρειάζεται να ανανεωθεί τακτικά για την αποθήκευση κάθε bit.

<sup>7</sup> αντιστάσεις που χρησιμοποιούνται στα ηλεκτρονικά κυκλώματα για να διασφαλιστεί ότι η τάση θα παραμείνει στα αναμενόμενα επίπεδα.

- **PWM:** 3, 5, 6, 9, 10 και 11. Παρέχουν Έξοδο 8-bit PWM.
- **SPI<sup>8</sup>:** 10 (SS), 11 (MOSI), 12 (MISO), 13 (SCK). Αυτοί οι ακροδέκτες επιτρέπουν επικοινωνία SPI, η οποία και παρέχεται από το hardware.
- **LED:** 13. Στον ακροδέκτη 13 υπάρχει ένα ενσωματωμένο LED. Όταν ο ακροδέκτης έχει τιμή HIGH, το LED φωτοβολεί..

#### 4.4.2. Ακροδέκτες για ειδικές λειτουργίες όπως:

- **I<sup>2</sup>C<sup>9</sup>:** 4 (SDA) και 5 (SCL). Υποστηρίζει το πρωτόκολλο I<sup>2</sup>C (TWI) χρησιμοποιώντας βιβλιοθήκες τις γλώσσας προγραμματισμού Wiring.
- **AREF:** Τάσης αναφοράς για την αναλογικές εισόδους.
- **Reset:** Αν τεθεί σε κατάσταση LOW τότε επαννεκινεί τον μικροελεγκτή. Σε αυτή τη γραμμή τοποθετείται ένας διακόπτης.

## 4.5. Μνήμη

Ο ATmega328P έχει 32KB μνήμης flash για την αποθήκευση κώδικα (2 KB εκ των οποίων χρησιμοποιούνται από τον bootloader). Έχει επίσης 2 KB SRAM για την εκτέλεση υπολογισμών και 1 KB μνήμης EEPROM για την μόνιμη αποθήκευση δεδομένων μετά την διακοπή τροφοδοσίας.

## 4.6. Τροφοδοσία

Το αναπτυξιακή πλακέτα Arduino τροφοδοτείται είτε από εξωτερική τροφοδοσία είτε απευθείας από θύρα USB. Η επιλογή της πηγής γίνεται αυτόματα. Ως εξωτερική τροφοδοσία ορίζεται είτε μια μπαταρία, είτε μετασχηματιστής των 9 Volt. Η μπαταρία μπορεί να συνδεθεί στις υποδοχές του Arduino Vin και GND όπου τοποθετούνται ο θετικός πόλος και ο αρνητικός αντίστοιχα. Αν τροφοδοτήσουμε με μετασχηματιστή τοποθετούμε το βύσμα στην υποδοχή που υπάρχει, με τον θετικό πόλο στο κέντρο στο αντίστοιχο βύσμα.

Η πλακέτα μπορεί να λειτουργήσει με εξωτερική πηγή από 6 έως 20 Volt. Αν ωστόσο τροφοδοτηθεί με λιγότερα από 7 Volt τα pin εξόδου 5 Volt δεν θα καταφέρουν να εξάγουν τάση 5 Volt. Αν από την άλλη δώσουμε πάνω από 12 Volt θα υπερθερμανθεί ο σταθεροποιητής τάσης στην πλακέτα και ενδεχόμενος να καταστραφεί. Συνεπώς μια ιδανική τάση είναι τα 9 Volt.

Οι ακροδέκτες τροφοδοσίας είναι οι εξής:

- **VIN:** ακροδέκτης για μη σταθεροποιημένη τάση. Συνήθως εδώ συνδέεται μια εξωτερική πηγή τροφοδοσίας.
- **5V:** ακροδέκτης σταθεροποιημένης τάσης 5 Volt. Χρησιμοποιείται για την τροφοδοσία του μικροελεγκτή ή άλλων ηλεκτρονικών στοιχείων.
- **3V3:** το ολοκληρωμένο κύκλωμα FTDI που βρίσκεται στην πλακέτα του Arduino παράγει τάση των 3.3V με μέγιστο ρεύμα 50mA.
- **GND:** ακροδέκτες γείωσης

<sup>8</sup> πρότυπο σύγχρονης σειριακής σύνδεσης δεδομένων που υποστηρίζει αμφίδρομη ροή μέσω συστήματος master / slave.

<sup>9</sup> multi-master σειριακός διάυλος υπολογιστή ενός άκρου που χρησιμοποιείται για να συνδέσει χαμηλής ταχύτητας περιφερειακά με μια άλλη ηλεκτρονική συσκευή.

## 4.7. Επικοινωνία

Το Arduino έχει την δυνατότητα να επικοινωνεί με H/Y, ένα άλλο Arduino ή άλλους μικροελεγκτές. Ο ATmega 328P παρέχει σειριακή επικοινωνία TTL<sup>10</sup> στα 5 Volt τύπου UART<sup>11</sup>, η οποία είναι διαθέσιμη από τους ακροδέκτες RX 0 (λήψη) και TX 1 (εκπομπή) του. Επιπλέον είναι ενσωματωμένο ένα FTDI<sup>12</sup> FT232RL IC το οποίο παρέχει σειριακή επικοινωνία με H/Y για προγραμματισμό μέσω της θύρας USB με την βοήθεια των ανάλογων FTDI drivers. Οι drivers αυτοί περιλαμβάνονται στο software για τον Arduino και παρέχουν μια virtual port (ιδεατή θύρα) επικοινωνίας στον H/Y για τους σκοπούς της επικοινωνίας.

## 4.8. Αυτόματη software επαναφορά

Αντί να απαιτεί το πάτημα του κουμπιού επαναφοράς (reset) πριν από την αποστολή κώδικα, το Arduino έχει σχεδιαστεί με τρόπο που να επιτρέπει την επαναφορά μέσω λογισμικού (software reset) ενώ είναι συνδεδεμένο σε έναν H/Y. Μία από τις γραμμές ελέγχου ροής υλικού του FT232RL συνδέεται με την reset γραμμή του ATmega328P μέσω ενός πυκνωτή 100 nF. Όταν αυτή η γραμμή είναι σε χρήση (χαμηλή), η γραμμή επαναφοράς «πέφτει» για αρκετά μεγάλο χρονικό διάστημα ώστε να προκαλέσει επαναφορά στο τσιπ. Το λογισμικό Arduino χρησιμοποιεί αυτή την δυνατότητα για να μας επιτραπεί να προσθέσουμε κώδικα πατώντας απλά το κουμπί upload στο περιβάλλον ανάπτυξης. Αυτό σημαίνει ότι ο bootloader μπορεί να έχει μικρότερο χρόνο εκκίνησης, όπως μπορεί και η γραμμή ελέγχου ροής υλικού να είναι συντονισμένη με την έναρξη της μεταφόρτωσης (upload).

Είναι προγραμματισμένο να αγνοεί ακατάλληλα δεδομένα (δηλαδή οτιδήποτε εκτός από την αποστολή νέου κώδικα) και θα παρακολουθήσει τα πρώτα byte δεδομένων που του αποστέλλονται και μετά ανοίγει την σύνδεση. Το Arduino περιέχει ένα trace (ίχνος χαλκού) που μπορεί να κοπεί για να απενεργοποιηθεί η λειτουργία αυτόματης επαναφοράς. Οι ακροδέκτες του ίχνους (χαλκού) πρέπει να είναι συγκολλημένοι για να την απενεργοποιήσουμε. Είναι η ένδειξη «RESET-EL». Επίσης είμαστε σε θέση να ακυρώσουμε την επιλογή αυτόματης επαναφοράς με τη σύνδεση ενός 110 ohm αντιστάτη μεταξύ της τροφοδοσίας των 5V και της γραμμής επαναφοράς (reset pin).

## 4.9. USB προστασία από υπερτάση

Το Arduino διαθέτει μια ασφάλεια που προκαλεί reset και προστατεύει τις θύρες USB του H/Y από τα βραχυκυκλώματα και τις υψηλές τιμές τάσης. Αν και οι περισσότεροι υπολογιστές παρέχουν τέτοια προστασία εσωτερικά, η ασφάλεια αυτή παρέχει ένα επιπλέον επίπεδο προστασίας. Σε περίπτωση που περισσότερο από 500 mA εφαρμοστούν στη θύρα USB, η ασφάλεια θα διακόψει αυτόματα τη σύνδεση μέχρι το βραχυκύκλωμα ή υπερτάση να σταματήσει.

## 4.10. Περιβάλλον ανάπτυξης εφαρμογών

Το Arduino χρησιμοποιεί ένα ιδιαίτερα απλό σύστημα ανάπτυξης εφαρμογών το Arduino IDE. Μια εφαρμογή με γραφικό περιβάλλον που προσφέρει δυνατότητες επεξεργασίας κώδικα (code editor), compiler (μεταφραστή κώδικα), code uploader (μεταφορά κώδικα στο Arduino) όπως και δυνατότητες διαχείρισης βιβλιοθηκών και αρχείων κώδικα.

Στο Arduino IDE η διαδικασία ανάπτυξης γίνεται αρχικά με την συγγραφή του κώδικα στο ειδικό πεδίο επεξεργασίας και στην συνέχεια αποστέλλεται στο Arduino μέσω της επιλογής upload. Ο ATmega328P παρέχεται preburned με έναν bootloader που μας επιτρέπει να φορτώσουμε τον νέο κώδικα χωρίς τη χρήση εξωτερικού programmer (προγραμματιστή υλικού). Επικοινωνεί

<sup>10</sup> κατηγορία ψηφιακών κυκλωμάτων κατασκευασμένα από διπολικά τρανζίστορ και αντιστάσεις.

<sup>11</sup> που μεταφράζει δεδομένα μεταξύ παράλληλης και σειριακής μορφής.

<sup>12</sup> κύκλωμα υπεύθυνο μετατροπή RS-232 ή TTL σειριακή μετάδοση σημάτων σε USB.

χρησιμοποιώντας το πρωτόκολλο STK 500<sup>13</sup>. Μπορεί επίσης να παρακάμψει τον bootloader και το πρόγραμμα του μικροελεγκτή μέσω της ICSP<sup>14</sup> (In-Circuit Serial Programming).

**Σημείωση: το Arduino IDE θα παρουσιαστεί εκτενεστέρα παρακάτω.**

---

<sup>13</sup> η επικοινωνία μεταξύ της αναπτυξιακής πλακέτας και του υπολογιστή γίνεται μέσω μιας εικονικής RS232 (θύρα COM PC) και παρέχει ρυθμούς διαμεταγωγής δεδομένων έως 115.2kbps, 8 bit δεδομένων, 1 stop bit και χωρίς ιστοιμία.

<sup>14</sup> τεχνική όπου μια προγραμματιζόμενη συσκευή προγραμματίζεται εφόσον τοποθετηθεί σε πλακέτα κυκλώματος.





## 5. Εισαγωγή στους αισθητήρες

*«Ο αισθητήρας είναι μια συσκευή,*

*δέχεται ένα ερέθισμα και ανταποκρίνεται μ' ένα ηλεκτρικό σήμα»*

### 5.1. Αίσθηση και ερέθισμα

Ο αισθητήρας έρχεται να γεφυρώσει ένα σημαντικό χάσμα. Το χάσμα μεταξύ του φυσικού κόσμου και των οργάνων που πρέπει να κατασκευάσει ο άνθρωπος ώστε να έχει πλήρη εικόνα τού και των μεταβολών του. Το πρόβλημα συνιστάται στο γεγονός ότι τα όργανα που κατασκευάζει ο άνθρωπος παράγουν έξοδο σε μορφή ηλεκτρικών σημάτων. Άρα πρέπει να κατασκευάσουμε μια νέα συσκευή που να μπορεί να παρακολουθεί τις μεταβολές του φυσικού κόσμου και να παράγει μια έξοδο ηλεκτρικού σήματος, ανάλογη των μεταβολών.

Το ερέθισμα είναι είτε η ποσότητα, είτε η ιδιότητα, είτε η κατάσταση που λαμβάνεται από τον αισθητήρα και το μετατρέπει σε ηλεκτρικό σήμα. Μερικές πηγές βιβλιογραφίας<sup>15</sup> χρησιμοποιούν ένα διαφορετικό όρο, «το μετρήσιμο» που έχει την ίδια έννοια, ωστόσο, με έμφαση σε ποσοτικά χαρακτηριστικά της αίσθησης. Ο αισθητήρας συχνά ορίζεται ως «η συσκευή που λαμβάνει και ανταποκρίνεται σε ένα σήμα ή ερέθισμα».

Ο σκοπός του αισθητήρα είναι να ανταποκριθεί σε κάποιο είδος εισόδου (ερέθισμα) με φυσικές ιδιότητες και να το μετατρέψει σε ένα ηλεκτρικό σήμα που να είναι συμβατό με τα ηλεκτρονικά κυκλώματα. Μπορούμε γενικά να πούμε ότι ο αισθητήρας είναι ένας μεταφραστής μιας ποσότητας μη ηλεκτρικής σε μια ηλεκτρική. Με τον όρο «ηλεκτρική ποσότητα», εννοούμε ένα σήμα, το οποίο μπορεί να αναγνωρίσιμο και επεξεργάσιμο από ηλεκτρονικές συσκευές.

Κάθε αισθητήρας είναι ένας μετατροπέας ενέργειας. Δεν έχει σημασία τι μέτρηση κάνουμε, διότι πάντα ασχολούμαστε με τη μεταφορά ενέργειας, από το «ερέθισμα» που θέλουμε να μετρήσουμε, στον αισθητήρα. Η διαδικασία της «αίσθησης» είναι μια ιδιαίτερη περίπτωση μεταφοράς πληροφοριών, επειδή κάθε μεταφορά πληροφοριών απαιτεί την μεταφορά ενέργειας. Ως εκ τούτου, ένας αισθητήρας διαθέτει συγκεκριμένες ιδιότητες ηλεκτρικές εισόδου και εξόδου.

Ο αισθητήρας δεν λειτουργεί αυτόνομα, είναι πάντα μέρος ενός ευρύτερου συστήματος που μπορούν να ενσωματώσουν άλλους ανιχνευτές, βελτιωτικά σήματος, επεξεργαστές σήματος, συσκευές μνήμης, καταγραφείς δεδομένων, και ενεργοποιητές. Η τοποθέτησή του σε ένα σύστημα είναι είτε εγγενής είτε εξωγενής.

- **Εγγενής:** μπορεί να είναι τοποθετημένος στο εσωτερικό μέρος μιας συσκευής και να ελέγχει την κατάσταση της, ώστε η συσκευή να επιτυγχάνει επιθυμητή απόδοση.
- **Εξωγενής:** μπορεί να τοποθετηθεί στην είσοδο μιας συσκευής για να αντιληφθεί τις εξωτερικές συνθήκες και να επισημάνει στο σύστημα σχετικά με τις διακυμάνσεις εξωτερικού περιβάλλοντος.

*«Ο αισθητήρας είναι πάντα μέρος κάποιου συστήματος συλλογής δεδομένων και οι επιδόσεις του πρέπει να εκτιμηθούν με βάση τις απαιτήσεις αυτού του συστήματος.»*

### 5.2. Αισθητήρας και το αισθητήριο

Ο αισθητήρας δεν θα πρέπει να συγχέεται με το αισθητήριο (ή μορφοτροπέας - transducer). Το αισθητήριο είναι ένας μετατροπέας ενέργειας, η οποία μπορεί να είναι οποιονδήποτε μορφής και να την μετατρέπει σε μια οποιονδήποτε άλλη. Ο αισθητήρας αντίθετα, μετατρέπει οποιοδήποτε μορφή ενέργειας σε ηλεκτρική. Ένα παράδειγμα αισθητηρίου είναι ένα ηχείο, το οποίο μετατρέπει ένα ηλεκτρικό σήμα σε ένα μεταβαλλόμενο μαγνητικό πεδίο και στη συνέχεια, σε ηχητικά κύματα. Το

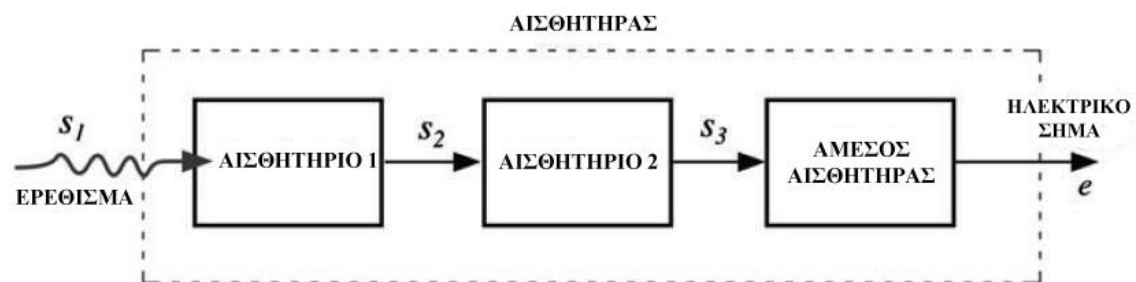
<sup>15</sup> Norton HN(1989) Handbook of transducers. Prentice Hall, Englewood Cliffs, NJ

αισθητήριο χρησιμοποιείται ως actuator (ενεργοποιητής) σε διάφορα συστήματα. Ένας actuator μπορεί να περιγραφεί ως το αντίθετο του αισθητήρα, μετατρέπει ένα ηλεκτρικό σήμα σε κάποιο είδος μη ηλεκτρικής ενέργειας (π.χ. ηλεκτρικός κινητήρας). Επίσης το αισθητήριο μπορεί να είναι εξάρτημα ενός πιο πολύπλοκου αισθητήρα.

### 5.3. Βασικά είδη αισθητήρων

Υπάρχουν δύο είδη αισθητήρων: **άμεσος** και **σύνθετος**.

- Ο **άμεσος** μετατρέπει ένα ερέθισμα σε ηλεκτρικό σήμα ή τροποποιεί ένα ηλεκτρικό σήμα χρησιμοποιώντας το κατάλληλο μέσον.
- Ο **σύνθετος** αισθητήρας έχει ένα ή περισσότερα αισθητήρια, που τοποθετούνται πριν από έναν άμεσο (αισθητήρα), ώστε να παράγει ηλεκτρικό σήμα.



Εικόνα 5. 1: Σχηματική παρουσίαση του σύνθετου αισθητήρα

### 5.4. Ταξινόμηση αισθητήρων

*«Τα συστήματα ταξινόμησης των αισθητήρων κυμαίνονται από πολύ απλά μέχρι ιδιαίτερα σύνθετα.»*

#### 5.4.1. Ο αισθητήρας μπορεί να είναι: παθητικός ή ενεργητικός.

- **Παθητικός αισθητήρας:** δεν χρειάζεται καμία πρόσθετη πηγή ενέργειας και ανταποκρίνεται με το να παράγει άμεσα ένα ηλεκτρικό σήμα. Δηλαδή, η ενέργεια ερεθίσματος στην είσοδο μετατρέπεται από τον αισθητήρα σε σήμα εξόδου.
- **Ενεργητικός αισθητήρας:** απαιτεί εξωτερική τροφοδοσία ρεύματος για τη λειτουργία του, η οποία ονομάζεται σήμα διέγερσης. Αυτό το σήμα τροποποιείται από τον αισθητήρα ώστε να παράγει το σήμα εξόδου. Οι ενεργοί αισθητήρες ονομάζονται μερικές φορές και παραμετρικοί, επειδή οι ιδιότητες τους αλλάζουν, λόγω των μεταβολών στην τιμή του ρεύματος τροφοδοσίας.

#### 5.4.2. Ανάλογα με το σύστημα αναφοράς, οι αισθητήρες μπορούν να ταξινομηθούν σε απόλυτους και σε σχετικούς.

- **Απόλυτος αισθητήρας:** ανιχνεύει ένα ερέθισμα σε σχέση με την απόλυτη φυσική κλίμακα, που είναι ανεξάρτητη από τις συνθήκες μέτρησης.
- **Σχετικός αισθητήρας:** παράγει ένα σήμα που σχετίζεται με κάποια ειδική περίπτωση που σε κάθε περίπτωση δεν ανταποκρίνεται στην απόλυτη φυσική κλίμακα.

### 5.4.3. Ένας άλλος τρόπος να ταξινομήσουμε έναν αισθητήρα είναι βάσει της ιδιαιτερότητας των χαρακτηριστικών ή των ιδιοτήτων του.

- **Προδιαγραφές:** ευαισθησίας, εύρος ερεθίσματος
- **Υλικό κατασκευής:** οργανικό, ανόργανο
- **Μέσα ανίχνευσης:** ηλεκτρικά, μαγνητικά
- **Φαινόμενα μετατροπής:** χημικά, βιολογικά
- **Πεδία εφαρμογών:** γεωργία, αυτοκινητοβιομηχανία
- **Τύπος ερεθίσματος:** ακτινοβολία, θερμότητα

### 5.5. Στατικά χαρακτηριστικά των αισθητήρων

Τα στατικά χαρακτηριστικά των αισθητήρων αναφέρονται στην κατάσταση κατά την οποία έχει επέλθει ισορροπία μεταξύ αισθητήρα και του μετρούμενου ερεθίσματος. Για να επιτευχθεί κάτι τέτοιο πρέπει το μετρούμενο ερέθισμα είτε να είναι σταθερό, είτε να μεταβάλλεται πολύ αργά σε σχέση με τη δυνατότητα του αισθητήρα να αντιληφθεί τη μεταβολή αυτή. Τα παραπάνω συνοψίζονται στα εξής:

- **Εύρος:** τα όρια στα οποία η συσκευή λειτουργεί αξιόπιστα.
- **Ακρίβεια:** η εγγύτητα της τιμής εξόδου προς τη τιμή εισόδου.
- **Σφάλμα:** η διαφορά ανάμεσα στη μετρούμενη τιμή και τη πραγματική τιμή.
- **Ανοχή:** το μέγιστο σφάλμα που μπορεί να δημιουργήσει ο αισθητήρας.
- **Διακριτική ικανότητα:** η μικρότερη αλλαγή τιμής εισόδου που μπορεί να ανιχνεύσει.
- **Ευαισθησία:** η σχέση της αλλαγής εξόδου προς τη αλλαγή εισόδου, είναι ίση με τη διαφορά των τιμών της εξόδου προς τη διαφορά των αντίστοιχων τιμών εισόδου.
- **Βαθμονόμηση:** η βαθμολόγηση της κλίμακας σε μονάδες.
- **Νεκρή ζώνη:** το μέγιστο ποσό αλλαγής της εισόδου που δεν επιφέρει αλλαγή στην έξοδο.
- **Γραμμικότητα:** ο βαθμός στον οποίο η γραφική παράσταση της εξόδου προσεγγίζει ευθεία ως προς την είσοδο του αισθητήρα.
- **Απόκριση:** ο χρόνος που απαιτείται για να λάβει τη τελική τιμή η έξοδος.
- **Καθυστέρηση:** η καθυστέρηση της αλλαγής της εξόδου ως προς την είσοδο.
- **Ευστάθεια:** η μεταβολή της εξόδου σε μεγάλη χρονική περίοδο, χωρίς μεταβολή της εισόδου και των συνθηκών.
- **Υστέρηση:** η διαφορά στην έξοδο όταν η κατεύθυνση της μεταβολής της εισόδου αντιστραφεί.
- **Επαναληψιμότητα:** η παραγωγή του ίδιου αποτελέσματος, σε διαφορετικές χρονικές στιγμές, με την ίδια είσοδο.
- **Ολίσθηση:** η μεταβολή των χαρακτηριστικών του αισθητήρα με το χρόνο και το περιβάλλον.
- **Στατικό σφάλμα:** σταθερό σφάλμα σε όλο το εύρος λειτουργίας, το οποίο μπορεί να αντισταθμιστεί.
- **Χρόνος λειτουργίας:** ο εκτιμώμενος χρόνος λειτουργίας στα πλαίσια των προδιαγραφών του.

### 5.6. Δυναμικά χαρακτηριστικά

Εκτός από τα στατικά χαρακτηριστικά τα οποία αναφέρονται στην περίπτωση όπου έχει επέλθει ισορροπία μεταξύ σήματος εισόδου και αισθητήρα (δηλαδή όταν πλέον τα σήματα εισόδου και εξόδου δεν μεταβάλλονται με το χρόνο), κάθε αισθητήρας διαθέτει και δυναμικά χαρακτηριστικά. Τα δυναμικά χαρακτηριστικά περιγράφουν τη συμπεριφορά του αισθητήρα μεταξύ της στιγμής κατά την οποία το σήμα εισόδου μεταβάλλεται έως τη στιγμή κατά την οποία το σήμα εξόδου θα σταθεροποιηθεί εκ νέου. Τα δυναμικά χαρακτηριστικά ισχύουν, όπως και τα στατικά, εντός συγκεκριμένου εύρους συνθηκών λειτουργίας του αισθητήρα. Αν ο αισθητήρας βρεθεί να λειτουργεί εκτός αυτού του εύρους, τότε τα δυναμικά χαρακτηριστικά αναμένεται να μεταβληθούν.

## 5.7. Συνοπτική παρουσίαση διαθέσιμων αισθητήρων:

- **Ακουστικός:**
  - Πλάτος κύματος
  - Φάση, πόλωση,
  - Φάσμα , μετρό ταχύτητας
  - Άλλο
- **Βιολογικός:**
  - Βιομάζα (τύπος, συγκέντρωση, κατάσταση)
  - Άλλο
- **Ηλεκτρικός:**
  - Φορτίο, ρεύμα
  - Τάση, διαφορά δυναμικού
  - Ηλεκτρικό πεδίο (πλάτος, φάση, πόλωση)
  - Φάσμα
  - Αγωγιμότητα
  - Διαπερατότητα
  - Άλλο
- **Μαγνητικός:**
  - Μαγνητικό πεδίο (πλάτος, φάση, πόλωση, φάσμα)
  - Μαγνητική διακύμανση
  - Διαπερατότητα
  - Άλλο
- **Ακτινοβολίας:**
  - Τύπου
  - Ενέργεια
  - Ένταση
  - Άλλο
- **Θερμικός:**
  - Θερμοκρασία
  - Μεταβολές
  - Ειδική θερμότητα
  - Θερμική αγωγιμότητα
  - Άλλο
- **Οπτικός:**
  - Πλάτος κύματος
  - Φάση , πόλωση,
  - Δείκτης διάθλασης
  - Δείκτης εκπομπής, αντανάκλαστικότητα, απορρόφηση
  - Άλλο
- **Μηχανικός:**
  - Θέση (γραμμική, γωνιακή)
  - Επιτάχυνση
  - Δύναμη
  - Πίεση καταβολής
  - Καταπόνηση
  - Μάζα, πυκνότητα
  - Αδράνεια, ροπή
  - Ταχύτητα ροής, ρυθμός μάζας, μεταφορά
  - Σχήμα, αντοχή, προσανατολισμός
  - Σκληρότητα, συμμόρφωση
  - Ιξώδες
  - Κρυσταλλικότητα, δομική ακεραιότητα
  - Άλλο
- **Χημικός:**
  - Στοιχεία (τύπος, συγκέντρωση, κατάσταση)

**Πίνακας 2: Είδη διαθέσιμων αισθητήρων.**



## 6. Θερμοκρασία, υγρασία και οι διαθέσιμοι αισθητήρες

### 6.1. Θερμοκρασία

*«θερμότητα και θερμοκρασία είναι διαφορετικές έννοιες. Η μεν θερμότητα είναι μορφή ενέργειας, η δε θερμοκρασία ιδιότητα και μέγεθος.»*

Η **θερμοκρασία** είναι η φυσική ιδιότητα που βασικά προσδιορίζει τις έννοιες του ζεστού και του κρύου. Για παράδειγμα, το σώμα με την μεγαλύτερη θερμοκρασία έναντι άλλου ή άλλων λέγεται θερμότερο (πιο ζεστό). Η θερμοκρασία στη πράξη είναι ακριβώς το μέτρο εκείνο με το οποίο προσδιορίζεται η «θερμική κατάσταση» των διαφόρων σωμάτων, είναι δηλαδή ένα φυσικό μέγεθος που συνδέεται με την μέση κινητική ενέργεια των σωματιδίων ενός συστατικού, το οποίο και χαρακτηρίζει πόσο θερμό ή πόσο ψυχρό είναι αυτό. Το αίτιο που δημιουργεί το αίσθημα του θερμού ή ψυχρού είναι η θερμότητα που όταν χορηγείται (απορροφάται) ή αφαιρείται (εκλύεται) από ένα σώμα προκαλεί «μεταβολή θερμοκρασίας» (ύψωση ή υποβίβαση).

Η θερμοκρασία μετριέται με ειδικά όργανα που λέγονται θερμόμετρα, την μεταβολή της οποίας διαβάζουμε στη κατάλληλα βαθμολογημένη κλίμακα σε βαθμούς θερμοκρασίας του θερμομέτρου. Τα θερμόμετρα διακρίνονται σε «κοινά» ή «υδραργυρικά», σε «θερμόμετρα οινόπνευματος» (για χαμηλότερες θερμοκρασίες), χρησιμοποιούνται επίσης και «ηλεκτρικά θερμόμετρα» που βασίζονται στην αρχή του θερμοηλεκτρικού στοιχείου, όπως επίσης και τα «οπτικά» ή ηλεκτρικά «πυρόμετρα» καθώς και άλλα ειδικών κατηγοριών. Η βαθμονόμηση των θερμομέτρων γίνεται σε βαθμούς Celsius (Κελσίου) °C, στο μετρικό σύστημα, και σε βαθμούς Fahrenheit (Φαρενάιτ) °F, στο αγγλικό σύστημα.

- Στο θερμόμετρο Κελσίου το μηδέν της κλίμακας (0° C) αντιστοιχεί στη θερμοκρασία τήξεως του πάγου, το δε 100 (100° C) στη θερμοκρασία βρασμού του ύδατος. Η ενδιάμεση αυτών απόσταση υποδιαιρείται σε 100 ίσα μέρη που καλούνται «βαθμοί Κελσίου».
- Στο θερμόμετρο Φαρενάιτ η θερμοκρασία τήξεως του πάγου αντιστοιχεί στους 32° F, η δε θερμοκρασία βρασμού στους 212° F. Το ενδιάμεσο αυτών διάστημα υποδιαιρείται σε 180 ίσα μέρη που καλούνται «βαθμοί Φαρενάιτ».

### 6.2. Κλίμακα Κέλβιν

Ο **Βαθμός Κέλβιν** (° K) είναι μονάδα μέτρησης της θερμοκρασίας στο Διεθνές Σύστημα Μονάδων (SI). Η αναφορά σε θερμοκρασίες στην κλίμακα Kelvin γίνεται όταν μια ένδειξη θερμοκρασίας συνοδεύεται με το σύμβολο «K», π.χ. Θερμοκρασία 77 K. Ο Βαθμός Κέλβιν φέρεται κατά την ομώνυμη κλίμακα όπου το απόλυτο μηδέν (-273,15° C βαθμοί Κελσίου) αντιστοιχεί στο 0° K. Η θερμοκρασία 0 K είναι η χαμηλότερη δυνατή θερμοκρασία και ονομάζεται **απόλυτο μηδέν**. Η κλίμακα επινοήθηκε από τον Ουίλιαμ Τόμσον (William Thomson), 1ο Βαρόνο του Kelvin.

- Η Κλίμακα Κέλβιν **δεν έχει αρνητικές τιμές**.
- Ο Βαθμός Κέλβιν είναι η βασική μονάδα μέτρησης θερμοκρασιών σε προβλήματα που έχουν σχέση με τη Θερμοδυναμική, τη μηχανική ρευστών κ.α.
- Ένας βαθμός κλίμακας Κέλβιν ισοδυναμεί με ένα βαθμό Κελσίου.
- Η **κλίμακα Kelvin** μετράει την λεγόμενη **απόλυτη θερμοκρασία**. Η διαφορά της με την κλίμακα Κελσίου έγκειται στους 273,15 βαθμούς. Δηλαδή:  $T_k = T_c + 273,15$ .

### 6.3. Απόλυτη θερμοκρασία

Κάθε θερμοκρασία που έχει ως αρχή μέτρησης το απόλυτο μηδέν της όποιας κλίμακας (Κελσίου ή Φαρενάιτ) χαρακτηρίζεται απόλυτη θερμοκρασία. Το απόλυτο μηδέν είναι η θερμοκρασία από την οποία ξεκινά η κλίμακα Κέλβιν και που προσδιορίζεται για μεν την κλίμακα Κελσίου στους  $-273,15^{\circ}\text{C}$  και για δε την κλίμακα Φαρενάιτ στους  $-459,67^{\circ}\text{F}$ . Θεωρητικά είναι η κατάσταση εκείνη στην οποία ένα υλικό δεν έχει καμία άλλη ενέργεια παρά αυτή από τις κβαντομηχανικές ταλαντώσεις των ατόμων που το αποτελούν (ενέργεια μηδενικού σημείου). Η απόλυτη θερμοκρασία συμβολίζεται με το γράμμα  $T$ .

### 6.4. Σχετική θερμοκρασία

Κάθε θερμοκρασία που μετριέται αρχίζοντας από το  $0^{\circ}$  της κλίμακας Κελσίου ή της κλίμακας Φαρενάιτ ονομάζεται *σχετική θερμοκρασία* και καλείται *θετική* όταν είναι άνω του μηδενός και *αρνητική* όταν είναι κάτω.

Η *σχετική θερμοκρασία* έχει ιδιαίτερα ευρύτατη χρήση τόσο στη καθημερινή ζωή όσο και στις διάφορες τεχνικές και μηχανολογικές εφαρμογές. Συμβολίζεται με το λατινικό γράμμα  $t$ .

### 6.5. Θερμοκρασία και αρχή λειτουργίας κάθε αισθητήρα θερμοκρασίας

Θερμοκρασία είναι η φυσική ιδιότητα που καθορίζει κατά πόσο ένα σύστημα ισορροπεί θερμικά ή όχι με άλλα. Για να λάβουμε την τιμή της θερμοκρασίας ενός σώματος απαιτείται ουσιαστικά η μετάδοση ενός μικρού μέρους της θερμικής ενέργειας του στον αισθητήρα του οποίου η λειτουργία είναι να μετατρέψει αυτή την ενέργεια σε ηλεκτρικό σήμα. Όταν ένας αισθητήρας τοποθετείται στο εσωτερικό ή επάνω στο αντικείμενο, η θερμική αγωγιμότητα εμφανίζεται στο όριο ανάμεσα στο αντικείμενο και αισθητήρα. Το αισθητήριο στοιχείο του αισθητήρα οπότε θερμαίνεται ή ψύχεται ανταλλάσει θερμότητα με το αντικείμενο. Τα σύγχρονα θερμόμετρα είναι συνήθως ηλεκτρονικά και χωρίζονται σε δύο κατηγορίες: στα θερμόμετρα επαφής και στα εξ αποστάσεως.

- **Τα θερμόμετρα επαφής:** μετρούν ερχόμενα σε επαφή με το σύστημα του οποίου ζητείται η θερμοκρασία, δηλαδή μετρούν ουσιαστικά την ίδια τους τη θερμοκρασία. Άρα, η ακρίβεια της μέτρησης εξαρτάται, μεταξύ άλλων, από το κατά πόσο έχει επέλθει θερμική ισορροπία μεταξύ θερμομέτρου και συστήματος.
- **Τα θερμόμετρα εξ αποστάσεως:** μετρούν τη θερμική ακτινοβολία του συστήματος και εμμέσως υπολογίζουν τη θερμοκρασία. Ως εκ τούτου δεν απαιτείται φυσική επαφή μεταξύ θερμομέτρου και μέτρηση συστήματος.

Το είδος του αισθητήρα που θα χρησιμοποιηθεί εξαρτάται από διάφορους παράγοντες, όπως το εύρος διακύμανσης της προς μέτρηση θερμοκρασίας, την απαιτούμενη ακρίβεια και πιστότητα, το περιβάλλον στο οποίο θα τοποθετηθεί ο αισθητήρας – αν δηλαδή θα υποστεί μηχανικές ή άλλες καταπονήσεις κ.λπ. Η δυσκολία ή ευκολία της μέτρησης εξαρτάται από την τιμή της θερμοκρασίας, το μέσο του οποίου θέλουμε να προσδιορίσουμε τη θερμοκρασία, καθώς και από τις απαιτήσεις για ακρίβεια και πιστότητα. Στον **πίνακα 3**, συνοψίζεται η δυσκολία μέτρησης της θερμοκρασίας, για διάφορες περιπτώσεις.

Θερμοκρασία	Απαίτηση σε ακρίβεια			
	±5°C	±1°C	±0.5°C	±0.1 °c
-200 °C	Απαιτείται προσοχή	Δύσκολα	Δύσκολα	Πολύ δύσκολα
0°C - 50 °C	Εύκολα	Απαιτείται προσοχή	Δύσκολα	Πολύ δύσκολα
1000 °C	Απαιτείται προσοχή	Πολύ δύσκολα	Εξαιρετικά δύσκολα	Σχεδόν δύνατον
2000 °C	Πολύ δύσκολα	Εξαιρετικά δύσκολα	Σχεδόν αδύνατον	Αποκλείεται

**Πίνακας 3: Ακρίβεια μέτρησης θερμοκρασίας σε συνάρτηση με την ακρίβεια σε βαθμούς °C.**



## 6.6. Η υγρασία

Ο όρος υγρασία αναφέρεται στον υδρατμό, ο οποίος είναι αέριο, είναι το νερό σε αέρια μορφή. Η υγρασία είναι παρούσα παντού στην ατμόσφαιρα. Ακόμη και στις πιο ξηρές περιοχές. Η μέτρηση της υγρασίας είναι πιο δύσκολη από την μέτρηση των περισσότερων άλλων μονάδων, όπως η θερμοκρασία και η πίεση. Ένας λόγος γι' αυτό είναι ότι διαθέτει εξαιρετικά δυναμική περιοχή τιμών.

Ένας άλλος λόγος που οι μετρήσεις είναι δύσκολες είναι ότι γίνονται σε πολύ διαφορετικές ατμοσφαιρικές συνθήκες, για παράδειγμα από θερμοκρασίες από  $-60\text{ }^{\circ}\text{C}$  έως  $1000\text{ }^{\circ}\text{C}$  με την παρουσία διαβρωτικών ή μη διαβρωτικών στοιχείων, καθώς και με ποικιλία προσμειξέων σωματιδίων ή χημικών.

## 6.7. Οι υγρομετρικές παράμετροι

Υπάρχουν δυο ομάδες υγρομετρικών παραμέτρων :

α) εκείνες που εξαρτώνται μόνο από το ποσό των υδρατμών στον αέρα και είναι γνωστές σαν απόλυτες μετρήσεις

β) εκείνες που εξαρτώνται όχι μόνο από το ποσό των υδρατμών, αλλά επίσης και από τη θερμοκρασία του αέρα. Αυτές οι τελευταίες είναι γνωστές σαν σχετικές μετρήσεις.

### A Ομάδα

#### 6.7.1. Απόλυτη Υγρασία

Είναι η αναλογία μάζας υδρατμών προς τον όγκο του αέρα ή αερίου. Συνήθως εκφράζεται σε γραμμάρια ανά κυβικό μέτρο και καθορίζει το ποσό των υδρατμών (σε γραμμάρια) που περιέχεται σε δοσμένο όγκο υγρού ατμοσφαιρικού αέρα. Οι μεταβολές της σχετίζονται με τις αλλαγές της ατμοσφαιρικής πίεσης. Για κάθε μια τιμή θερμοκρασίας του αέρα υπάρχει και μια αντίστοιχη μέγιστη τιμή απόλυτης υγρασίας.

### B Ομάδα

#### 6.7.2. Σχετική υγρασία

Η σχετική υγρασία του αέρα αποτελεί την πιο κοινή έκφραση της ατμοσφαιρικής υγρασίας. Αυτή χαρακτηρίζει το λόγο των υδρατμών που υπάρχουν στην ατμόσφαιρα σε μια δεδομένη τιμή θερμοκρασίας και πίεσης σε σχέση με την μέγιστη ποσότητα των υδρατμών την οποία ο αέρας είναι ικανός να κρατήσει στις ίδιες συνθήκες πίεσης<sup>16</sup> και θερμοκρασίας μέχρις ότου αυτός κορεσθεί. Η σχετική υγρασία εκφράζεται **επί τοις %**. Έτσι, το 50 τοις εκατό RH σημαίνει ότι το 50 τοις εκατό του αέρα στο δείγμα το κατέχουν υδρατμοί επί του συνόλου σε αυτή τη θερμοκρασία. Αν το δείγμα του αέρα θερμανθεί, η σχετική υγρασία θα μειωθεί, έστω και αν η ποσότητα των υδρατμών δεν έχει αλλάξει.

Και ορίζεται από τον τύπο:

$$\bullet \quad \Sigma_v = B'/B \times 100$$

**B'**: ποσότητα υπαρχόντων υδρατμών

**B**: ποσότητα που καθιστά τον αέρα κεκορεσμένο ή μέγιστη τάση υδρατμών

**«Ο κεκορεσμένος αέρας έχει σχετική υγρασία 100%, ενώ ο τελείως ξηρός αέρας έχει 0%»**

---

<sup>16</sup> το πηλίκιο της ασκούμενης ατμοσφαιρικής πίεσης σε μία επιφάνεια δια του εμβαδού της επιφάνειας αυτής

Μια μεταβολή στη σχετική υγρασία του ατμοσφαιρικού αέρα μπορεί να προκληθεί με ένα από τους δυο παρακάτω τύπους:

α) Αν υπάρχει μια ελεύθερη επιφάνεια νερού, η υγρασία μπορεί να αυξηθεί με την εξάτμιση. Αυτή η διεργασία είναι βαθμιαία και προϋποθέτει τη διάχυση των υδρατμών μέσα στην ατμόσφαιρα.

β) Όταν αυξάνει η θερμοκρασία του αέρα, παράλληλα αυξάνει και η ικανότητα του να συγκρατεί περισσότερη υγρασία. Σε μια τέτοια περίπτωση εφόσον δεν προστίθενται καινούρια ποσά υδρατμών η σχετική υγρασία θα ελαττωθεί, αντίθετα με τη μείωση της θερμοκρασίας η σχετική υγρασία αυξάνεται γιατί μειώνεται η ικανότητα κατακράτησης υδρατμών από τον αέρα



Εικόνα 6. 1: Διάγραμμα μεταβολής σχετικής υγρασίας συναρτήσει της θερμοκρασίας

### Χρήσιμοι ορισμοί

- Ο αέρας που περιλαμβάνει την μέγιστη ποσότητα υδρατμών ονομάζεται **κεκορεσμένος**.
- Η μεταβολή της αέριας κατάστασης των υδρατμών σε υγρή ή στερεά καλείται **συμπύκνωση**<sup>17</sup>.
- Η τάση των υδρατμών στο «σημείο του κορεσμού» ονομάζεται «μέγιστη τάση υδρατμών». Αυτή είναι συνάρτηση μόνο της θερμοκρασίας της ατμόσφαιρας και αυξάνει με αυτήν.

### Επίσης:

Όταν επικρατεί ομίχλη ο αέρας είναι συνήθως κεκορεσμένος. Όταν η σχετική υγρασία είναι 100% τότε η ατμόσφαιρα είναι κεκορεσμένη δηλαδή πλήρης υδρατμών μη δυνάμενη να συγκρατήσει άλλους. Αντίθετα όταν είναι π.χ. 50% και η θερμοκρασία αέρος 20° C για τον Χειμώνα, και 26° C για το Καλοκαίρι, τότε αισθανόμαστε ευχάριστα.

## 6.8. Σημείο Δρόσου

Το σημείο δρόσου, εκφράζεται σε °C ή °F, είναι η θερμοκρασία και η πίεση στην οποία ένα αέριο αρχίζει να συμπυκνώνεται σε υγρό. Είναι η θερμοκρασία στην οποία παρουσιάζεται συμπύκνωση των υδρατμών της ατμόσφαιρας όταν αυτή αρχίζει να ψύχεται κάτω από σταθερή πίεση. Αφού ο θερμός αέρας είναι ικανός να συγκρατεί περισσότερους υδρατμούς από τον ψυχρό αέρα, το σημείο δρόσου, κατά μέσο όρο, είναι μεγαλύτερο το καλοκαίρι παρά τον χειμώνα, και γενικότερα αυξάνει με την αύξηση της θερμοκρασίας.

<sup>17</sup> η μετατροπή αερίου σε υγρό ή στερεό



Εικόνα 6. 2: Διάγραμμα μεταβολής σημείο δρόσου συναρτήσει της θερμοκρασίας

## 6.9. Υγρασία και διαθέσιμοι αισθητήρες

*«Η μέτρηση της υγρασίας δεν είναι ένα ασήμαντο έργο, ούτε είναι διαισθητικά κατανοητό»*

Θέματα όπως διάρκεια ζωής προϊόντων, ποιότητα, κόστος, ασφάλεια εγκαταστάσεων, και προστασία υγείας εργαζομένων επηρεάζονται σημαντικά από τις μεταβολές της υγρασίας. Πολλές μελέτες έχουν γίνει για την βελτίωση των αισθητήρων υγρασίας, αλλά παρόλα αυτά δεν μπορούν να μετρήσουν και να καταγράψουν με ακρίβεια 100% την υγρασία. Λόγω των παραπάνω έχουμε οδηγηθεί στην κατασκευή αισθητήρων υγρασίας πολλών ειδών όπου ο καθένας μπορεί να μας παρέχει μόνο ένα είδος μέτρησης κάθε φορά. Οπότε με βάση τις εκάστοτε ανάγκες θα πρέπει να γίνει και η επιλογή του κατάλληλου αισθητήρα.

Δεν είναι εύκολο να μετρηθεί η υγρασία με ακρίβεια. Πολλοί μέθοδοι έχουν χρησιμοποιηθεί για την μέτρηση της σχετικής υγρασίας: χρήση ανθρώπινης τρίχας ή νάιλον φιλμ, μέθοδος υγρού και ξηρού βολβού, αισθητήρες ημιαγωγών, και ούτω καθεξής. Όλες αυτές οι μέθοδοι έχουν τα πλεονεκτήματά τους αλλά και τους περιορισμούς τους και κανείς από αυτούς δεν μπορεί να χαρακτηριστεί τέλειος. Επιπλέον, υπάρχουν πολύ λίγες συσκευές υψηλής ακρίβειας βαθμονόμησης, παραγωγής και μέτρησης υγρασίας, καθιστώντας σχετικά δύσκολο να επιβεβαιώσουμε την ακρίβεια της μέτρησης.

## 6.10. Σφάλμα μέτρησης και αισθητήρες

«Κάθε φορά που προσπαθούμε να ποσοτικοποιήσουμε μια φυσική διαδικασία, εμφανίζεται η αβεβαιότητα»

Ένα χαρακτηριστικό που συνήθως παρατηρούμε σ' όλους σχεδόν τους τύπους αισθητήρων είναι ο τρόπος με τον οποίο εκφράζουμε την ακρίβεια τους. Μπορούμε να την εκφράσουμε με κάποιο ποσοστό απόκλισης τις εκατό (%) ή ένα καθορισμένο εύρος τιμών (π.χ. απόκλιση  $\pm 5$ ). Ο λόγος που εμφανίζονται αυτές οι αποκλίσεις είναι η ύπαρξη του σφάλματος μέτρησης όπως και η αβεβαιότητα μέτρησης.

## 6.11. Η αβεβαιότητα μέτρησης

Στην μετρολογία (η επιστήμη των μετρήσεων), η αβεβαιότητα μέτρησης είναι μια μη-αρνητική παράμετρος που χαρακτηρίζει τη διασπορά των τιμών που αποδίδονται σε μια μετρούμενη ποσότητα. Η αβεβαιότητα βασίζεται στην θεωρία των πιθανοτήτων και δείχνει την ελλιπή κατανόηση της ποσότητας. Όλες οι μετρήσεις υπόκεινται σε αβεβαιότητα και η τιμή που μετρήθηκε είναι μόνο πλήρης εάν συνοδεύεται από την δήλωση αβεβαιότητας.

## 6.12. Τυχαία σφάλματα και συστηματικά σφάλματα

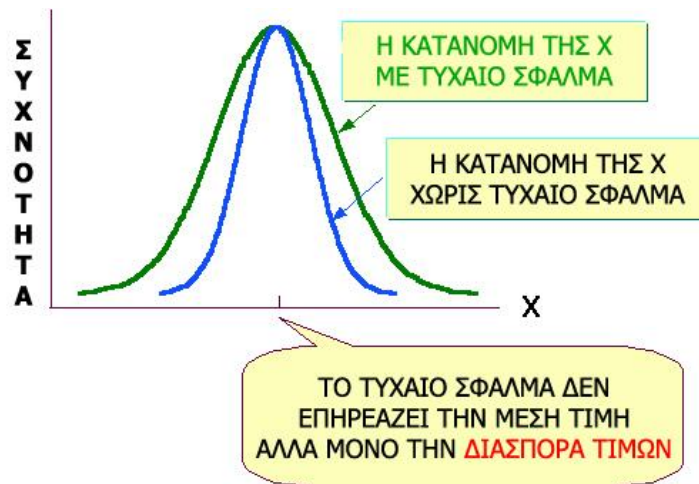
«Υπάρχουν δύο τύποι σφάλματος μέτρησης, το συστηματικό σφάλμα και το τυχαίο σφάλμα»

Ένα **συστηματικό σφάλμα** (εκτίμηση του οποίου είναι γνωστή ως μεροληψία στη μέτρηση ή bias) συνδέεται με το γεγονός ότι η μετρούμενη τιμή περιλαμβάνει ένα offset (απόκλιση). Σε γενικές γραμμές, ένα συστηματικό σφάλμα, θεωρείται ως μια ποσότητα. Αποτελεί στοιχείο του σφάλματος μέτρησης που παραμένει σταθερό ή εξαρτάται με έναν συγκεκριμένο τρόπο με κάποια άλλη ποσότητα.



Εικόνα 6. 3: Γραφική αναπαράσταση του συστηματικού σφάλματος

Ένα **τυχαίο σφάλμα** συνδέεται με το γεγονός ότι όταν η μέτρηση επαναλαμβάνεται θα υπάρξει μια μετρημένη τιμή που είναι διαφορετική από την προηγούμενη. Είναι τυχαίο το ότι η επόμενη μετρούμενη τιμή δεν μπορεί να προβλεφθεί με ακρίβεια από τις προηγούμενες της.



Εικόνα 6. 4: Γραφική αναπαράσταση του τυχαίου σφάλματος

«Σε γενικές γραμμές μπορεί ένα σφάλμα μέτρησης να οφείλεται ταυτόχρονα και στους δυο παραπάνω τύπους σφαλμάτων»

### 6.13. Προσδιορισμός τιμής

Η διαπίστωση και εκτίμηση της ακρίβειας μέτρησης μπορεί να γίνει με την επανάληψη της μέτρησης. Όσο λιγότερο αποκλίνουν οι μετρούμενες τιμές, τόσο ακριβέστερο είναι το αποτέλεσμα που θα καταγράψουμε ως τιμή της μέτρησης. Ταυτόχρονα, όσο περισσότερο επαναλαμβάνεται μια τιμή, τόσο πιο πιθανό είναι η συγκεκριμένη τιμή να είναι η πραγματική τιμή.

Το φαινόμενο της επαναλαμβανόμενης μέτρησης της ίδιας τιμής ονομάζεται **επαναληψιμότητα της τιμής**. Έτσι, η τιμή με τη μεγαλύτερη επαναληψιμότητα μάλλον είναι η πραγματική τιμή. Πριν όμως καταλήξουμε σε αυτό το συμπέρασμα πρέπει να ελέγξουμε αν κάποιο σφάλμα επαναλαμβάνεται συστηματικά (συστηματικό σφάλμα), για παράδειγμα σε μια μέτρηση μήκους δεν βλέπουμε ότι το μέτρο που χρησιμοποιούμε δεν είναι εκατοστά αλλά ίντσες. Με βάση τα παραπάνω, αν γίνουν πολλές μετρήσεις και δείχνουν την ίδια τιμή, τότε σχεδόν σίγουρα η πραγματική τιμή του μεγέθους είναι αυτή η τιμή.

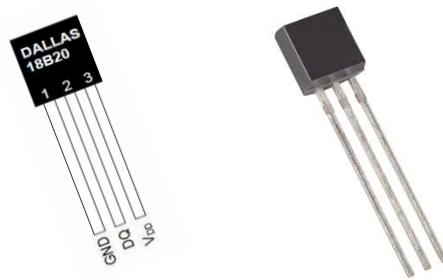
Αντίθετα, αν σε πολλές επαναλήψεις μετρηθούν πολλές τιμές και μερικές από αυτές αποκλίνουν αρκετά, τότε χρησιμοποιούμε ορισμένες μεθόδους, για την απαλοιφή των ακραίων τιμών και τον υπολογισμό της κεντρικής τιμής των μετρήσεων. Ως κεντρική τιμή των εγκεκριμένων μετρήσεων συνήθως λαμβάνουμε τη μέση τιμή τους (τον αριθμητικό μέσον όρο). Θεωρούμε ότι αυτή η κεντρική τιμή είναι η πραγματική, γιατί σύμφωνα με τη θεωρία των πιθανοτήτων αυτή η τιμή είναι πιο πιθανό να είναι η πραγματική τιμή.



## 7. Παρουσίαση των αισθητήρων του υποσυστήματος τηλεμετρίας

### 7.1. Αισθητήρας θερμοκρασίας Dallas DS18B20

Το ψηφιακό θερμόμετρο DS18B20 παρέχει μετρήσεις στην κλίμακα Κελσίου εύρους 9 έως 12 bit και λειτουργία συναγερμού σε σημεία χαμηλής και υψηλής ενεργοποίησης (trigger) που δεν μπορούν να μεταβληθούν κατά την λειτουργία αλλά μπορούν να ορισθούν από τον χρήστη. Ο DS18B20 επικοινωνεί μέσω δίαυλου ενός καλωδίου (1-Wire bus) με τον μικροελεγκτή, και απαιτεί μόνο γραμμή δεδομένων (π.χ. καλώδιο) και μία γείωση. Μπορεί να λειτουργήσει σε θερμοκρασίες από  $-55^{\circ}\text{C}$  έως  $+125^{\circ}\text{C}$  και παρέχει ακρίβεια  $\pm 0,5^{\circ}\text{C}$  στο εύρος  $-10^{\circ}\text{C}$  έως  $+85^{\circ}\text{C}$ . Επιπλέον ο DS18B20 μπορεί να αποκομίσει ρεύμα από την γραμμή δεδομένων μέσω παρασιτικού ρεύματος, άρα δεν απαιτείται εξωτερική πηγή ρεύματος. Τέλος κάθε DS18B20 έχει έναν μοναδικό κώδικα 64 bit, ο οποίος επιτρέπει σε πολλαπλά DS18B20 να λειτουργούν στον ίδιο δίαυλο ενός καλωδίου (1-Wire bus). Κατά συνέπεια, είναι απλό μ' έναν μικροελεγκτή να ελέγξουμε πολλά DS18B20 που κατανέμονται σε μια μεγάλη περιοχή.



Εικόνα 7. 1: Ο DS18B20 σε μορφή TO92

### 7.2. Χαρακτηριστικά:

- 1-Wire<sup>®</sup> Interface που απαιτεί μόνο ένα Pin για επικοινωνία.
- κάθε συσκευή έχει έναν μοναδικό 64-Bit σειριακό κωδικό αποθηκευμένο σε on-board ROM.
- δυνατότητα multidrop που απλοποιεί τις καταναεμημένες εφαρμογές μέτρησης Θερμοκρασίας
- δεν απαιτεί επιπλέον εξαρτήματα.
- μπορεί να τροφοδοτηθεί από τη γραμμή τροφοδοσίας 3.0V έως 5.5V.
- μέτρα θερμοκρασίες από  $-55^{\circ}\text{C}$  έως  $+125^{\circ}\text{C}$  ( $-67^{\circ}\text{F}$  έως  $257^{\circ}\text{F}$ ).
- $\pm 0,5^{\circ}\text{C}$  Ακρίβεια από  $-10^{\circ}\text{C}$  έως  $+85^{\circ}\text{C}$ .
- ρυθμιζόμενη ακρίβεια μέτρησης εύρους 9 έως 12 bit.
- χρόνος μέτρησης θερμοκρασίας 750ms (Max).
- λειτουργία συναγερμού.

### 7.3. Περιγραφή ακροδεκτών

Αριθμός Ακροδέκτη ( Pin No)	Ονομασία Ακροδέκτη	Λειτουργία Ακροδέκτη
1	GND	Γείωση
2	DQ	Είσοδος / Έξοδος δεδομένων. Ακροδέκτης διαύλου ενός καλωδίου ( 1-Wire® bus pin )
3	VDD	Προαιρετικός. Ο VDD πρέπει να γειωθεί για να λειτουργήσει μέσω παρασιτικού ρεύματος

Πίνακας 4: Ακροδέκτες του DS18B20.

### 7.4. Αισθητήρες θερμοκρασίας ολοκληρωμένου κυκλώματος

Οι αισθητήρες θερμοκρασίας ημιαγωγών παράγονται με τη μορφή IC. Οφείλουν την ύπαρξη τους στο γεγονός ότι οι δίοδοι ημιαγωγών έχουν ευαισθησία στην θερμοκρασία και το εκδηλώνουν με διαφορά τάσης στα άκρα τους. Όταν τα δύο πανομοιότυπα τρανζίστορ λειτουργούν σε σταθερή αναλογία πυκνότητας συλλέκτη ρεύματος<sup>18</sup>, η διαφορά τάσης μεταξύ βάσης και εκπομπού είναι ευθέως ανάλογη προς την απόλυτη θερμοκρασία.

Η χρήση αισθητήρων θερμοκρασίας ολοκληρωμένου κυκλώματος περιορίζεται σε εφαρμογές όπου η θερμοκρασία είναι εντός του εύρους -55° έως 150 ° C. Το εύρος μέτρησης τους μπορεί να είναι μικρό σε σύγκριση με εκείνη των θερμοστοιχείων<sup>19</sup> και των RTD<sup>20</sup>, αλλά έχουν πολλά πλεονεκτήματα: είναι μικροί σε διαστάσεις, ακριβείς και ανέξοδοι.

Οι διαθέσιμοι αισθητήρες θερμοκρασίας σε μορφή IC είναι είτε σε αναλογική ή ψηφιακή μορφή εξόδου. Μπορούν επικοινωνούν μέσω μιας ψηφιακής γραμμής, όπως ένα καλώδιο PWM, ή SPI σύνδεσης.

### 7.5. Περιληπτική παρουσίαση του αισθητήρα DS18B20

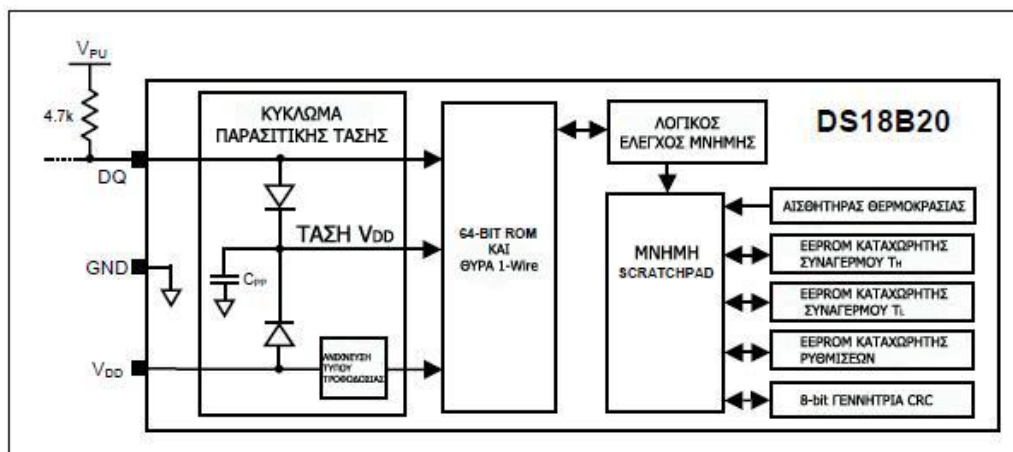
Το 64-bit ROM έχει αποθηκευμένο το μοναδικό σειριακό κωδικό της συσκευής. Η μνήμη (scratchpad memory) περιέχει τον καταχωρητή θερμοκρασίας μεγέθους 2-byte που προέρχεται από την ψηφιακή έξοδο του αισθητήρα θερμοκρασίας. Επιπλέον, η μνήμη παρέχει πρόσβαση στους καταχωρητές μεγέθους 1-byte της άνω και κάτω στάθμης (TH και TL) ενεργοποίησης του συναγερμού και καταχωρητή μεγέθους 1-byte για τις ρυθμίσεις. Ο καταχωρητής ρυθμίσεων επιτρέπει στο χρήστη να ρυθμίσει την ανάλυση σε 9, 10, 11, ή 12 bit κατά την ψηφιακή μετατροπή θερμοκρασίας. Οι καταχωρητές TH, TL, και των ρυθμίσεων είναι EEPROM, έτσι θα διατηρούν τα δεδομένα όταν η συσκευή είναι εκτός λειτουργίας.

<sup>18</sup> Συλλέγει τα φορτία που καταφθάνουν από τον εκπομπού του τρανζίστορ.

<sup>19</sup> Η λειτουργία των θερμοστοιχείων βασίζεται στο θερμοηλεκτρικό φαινόμενο Seebeck, κατά το οποίο όταν δύο μεταλλικοί αγωγοί διαφορετικών υλικών έρθουν σε επαφή στα άκρα τους, αναπτύσσεται ηλεκτρεγερτική δύναμη, που οφείλεται σε καθαρά θερμικά αίτια, και της οποίας το μέγεθος εξαρτάται από την θερμοκρασία του σημείου επαφής.

<sup>20</sup> Θερμοαντιστάτες όπου η μέτρηση της θερμοκρασίας προέρχεται από την συσχέτιση της αντίστασης του με τη θερμοκρασία.





Εικόνα 7. 2: Διάγραμμα block του DS18B20

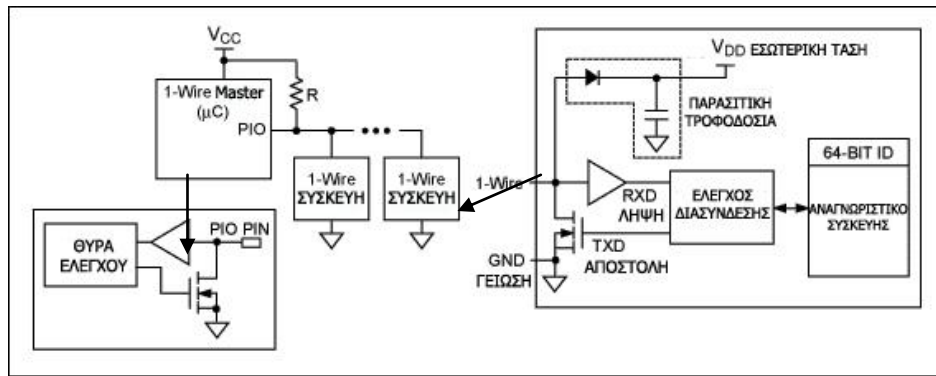
Ο DS18B20 χρησιμοποιεί πρωτόκολλο διαύλου ενός καλωδίου(1-Wire bus protocol) που υλοποιεί επικοινωνία μόνο μ' ένα σήμα ελέγχου. Η γραμμή ελέγχου απαιτεί μια αδύναμη αντίσταση pullup. Σε αυτό το σύστημα διαύλου, ο μικροελεγκτής (η συσκευή master) προσδιορίζει και ελέγχει τους αισθητήρες με βάση τον μοναδικό κωδικό 64-bit κάθε αισθητήρα. Επειδή ο κάθε αισθητήρας έχει μοναδικό κωδικό, ο αριθμός των αισθητήρων που μπορούν να υπάρχουν σε ένα δίαυλο είναι ουσιαστικά απεριόριστος.

## 7.6. Μοντέλο επικοινωνίας master/slave

Το master / slave είναι ένα μοντέλο επικοινωνίας, όπου μία συσκευή ή διαδικασία (γνωστή ως master), έχει μονόδρομο έλεγχο ενός ή περισσότερων άλλων συσκευών (γνωστών ως slave). Μόλις αυτή αρχικοποιηθεί, η κατεύθυνση ελέγχου είναι πάντα από τον master στον slave. Σε μερικά συστήματα ο master επιλέγεται από μια ομάδα συσκευών, με τις υπόλοιπες συσκευές στο ρόλο του slave. Ο master ξεκινά την επικοινωνία και στέλνει αιτήσεις. Ένας slave λαμβάνει μόνο τα αιτήματα, εκτελεί αναγνώριση του master και στέλνει απαντήσεις. Μόνος του ο σκλάβος δεν μπορεί να εκτελεί το πρωτόκολλο επικοινωνίας.

## 7.7. Τεχνολογία 1-Wire

Η τεχνολογία 1-Wire είναι ένα σειριακό πρωτόκολλο. Χρησιμοποιεί μια ενιαία γραμμή δεδομένων συν μία γείωση για την επικοινωνία. Ο 1-Wire master αρχικοποιεί και ελέγχει την επικοινωνία με μία ή περισσότερες slave στο δίαυλο (1-Wire). Κάθε slave έχει έναν μοναδικό και αναλλοίωτο, εργοστασιακά προγραμματιζόμενο 64-bit αναγνωριστικό (αριθμός αναγνώρισης), το οποίο χρησιμεύει ως διεύθυνση της συσκευής στο δίαυλο 1-Wire. Οι συσκευές λειτουργούν σε εύρος τάσης 2.8V (min) έως 5.25V (max). Δεν απαιτούν πάντοτε pin για την τροφοδοσία, την αντλούν αν χρειαστεί από τον δίαυλο 1-Wire μέσω παρασιτικής τροφοδοσίας. Ο DS18B20 είναι πάντα τύπου slave. Όταν υπάρχει μόνο ένας slave στο δίαυλο, τότε το σύστημα που αναφέρεται ως «single-drop». Το σύστημα είναι «multidrop» αν υπάρχουν πολλοί slave στο δίαυλο. Όλα τα δεδομένα και οι εντολές που διαβιβάζουν πρώτα το λιγότερο σημαντικό bit στον δίαυλο 1-Wire.



Εικόνα 7. 3: Βασικό διάγραμμα διαύλου 1-Wire

## 7.8. Ιδιαίτερα χαρακτηριστικά της τεχνολογίας 1-Wire

Σε αντίθεση με άλλα συστήματα σειριακής επικοινωνίας, όπως το I<sup>2</sup>C ή SPI, το 1-Wire είναι ειδικά σχεδιασμένη για χρήση σε περιβάλλοντα επικοινωνίας. Είτε η αποσύνδεση από το δίαυλο ή απώλεια επικοινωνίας θέτει τον/τους slave σε μια καθορισμένη κατάσταση, reset. Όταν η τάση τροφοδοσίας επανέλθει, οι slave αποστέλλουν σήμα (παλμό) παρουσίας. Εξαιρετικά υψηλή προστασία σε στατικό ηλεκτρισμό (ESD).

## 7.9. Μέτρηση θερμοκρασίας

Η βασική λειτουργία του αισθητήρα DS18B20 είναι η άμεση μετατροπή της θερμοκρασίας σε ψηφιακή τιμή. Η ανάλυση (σε bit και °C) του αισθητήρα μπορεί να ρυθμιστεί από τον χρήστη, σε διατιμήσεις της τάξης των 0,5 °C, 0,25 °C, 0,125 °C και 0,0625 °C αντίστοιχα. Η αρχική τιμή ανάλυσης είναι 12-bit. Μετά τη μετατροπή, τα προκύπτοντα θερμοκά δεδομένα αποθηκεύονται στον καταχωρητή 2-byte θερμοκρασίας και εγγράφονται στην μνήμη (scratchpad). Η έξοδος των δεδομένων θερμοκρασίας είναι βαθμονομημένη σε βαθμούς Κελσίου. Για Φαρενάιτ τιμές, πρέπει να χρησιμοποιηθεί ένας πίνακας μετατροπής.

Θερμοκρασία σε °C	Ψηφιακή Έξοδος Δυαδική	Ψηφιακή Έξοδος Δεκαεξαδική
+ 125	0000 0111 1101 0000	07D0h
+85*	0000 0101 0101 0000	0550h
+25.0625	0000 0001 1001 0001	0191h
+ 10.125	0000 0000 1010 0010	00A2h
+0.5	0000 0000 0000 1000	0008h
0	0000 0000 0000 0000	0000h
-0.5	1111 1111 1111 1000	FFF8h
-10.125	1111 1111 0101 1110	FF5Eh
-25.0625	1111 1110 0110 1111	FH6Fh
-55	1111 1100 1001 0000	FC90h

Πίνακας 5: Αντιστοίχιση θερμοκρασίας και τιμής καταχωρητή.

\* Η τιμή του καταχωρητή θεοκρασίας είναι +85°C.

## 7.10. Μνήμη

Η μνήμη του DS18B20 είναι οργανωμένη όπως φαίνεται στην **εικόνα 7.4**. Αποτελείται από μια μνήμη τύπου SRAM (scratchpad) με μη μεταβλητό αποθηκευτικό χώρο (nonvolatile) EEPROM για την αποθήκευση των υψηλών και χαμηλών τιμών καταχωρητών που ενεργοποιούν τον συναγερμό (TH και TL) και του καταχωρητές ρυθμίσεων.

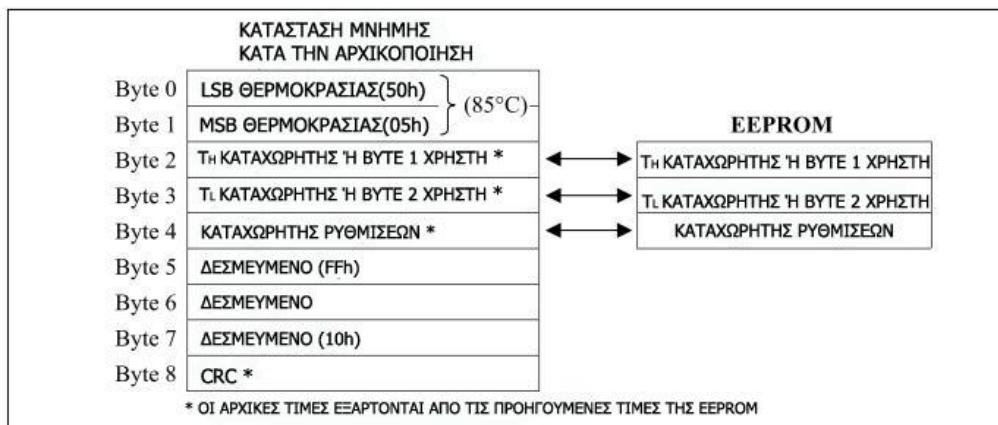
Τα byte 0 και 1 της scratchpad περιέχει τα LSB και το MSB του καταχωρητή που περιέχει την τιμή της θερμοκρασίας. Αυτά τα byte είναι μόνο για ανάγνωση. Τα byte 2 και 3 παρέχουν πρόσβαση

στους TH και TL καταχωρητές. Το byte 4 περιλαμβάνει τα δεδομένα διαμόρφωσης των καταχωρητών, Τα byte 5, 6, και 7 είναι για αποκλειστική εσωτερική χρήση από τη συσκευή και δεν μπορούν να αλλαχτούν οι τιμές τους. Το byte 8 της μνήμης είναι μόνο για ανάγνωση και περιέχει τον κώδικα CRC για τα bytes 0 έως 7 της μνήμης.

Τα δεδομένα εγγράφονται στα byte 2, 3, και 4 της μνήμης με την εντολή **Write Scratchpad [4Eh]**. Τα δεδομένα πρέπει να διαβιβάζονται στον DS18B20 ξεκινώντας από το λιγότερο σημαντικό bit του byte 2. Για να επαληθεύει η ακεραιότητα των δεδομένων, η μνήμη μπορεί να διαβάσει (χρησιμοποιώντας την εντολή **Read Scratchpad [BEh]**) αφού τα δεδομένα είναι να εγγραφούν. Κατά την ανάγνωση της μνήμης, τα δεδομένα μεταφέρονται μέσω του διαύλου 1-Wire ξεκινώντας από το λιγότερο σημαντικό bit του byte 0. Για να μεταφερθούν τα TH, TL και τα δεδομένα διαμόρφωσης από τη μνήμη στην EEPROM, ο master του καναλιού πρέπει να δώσει την εντολή **Copy Scratchpad [48h]**.

Τα δεδομένα των καταχωρητών της EEPROM διατηρούνται όταν η συσκευή είναι εκτός λειτουργίας. Κατά αρχικοποίηση της EEPROM τα δεδομένα ξαναφορτώνονται στις αντίστοιχες θέσεις μνήμης. Τα δεδομένα μπορούν επίσης να ξαναφορτωθούν από την EEPROM στην μνήμη ανά πάσα στιγμή, χρησιμοποιώντας την εντολή **Recall E2[B8h]** εντολή.

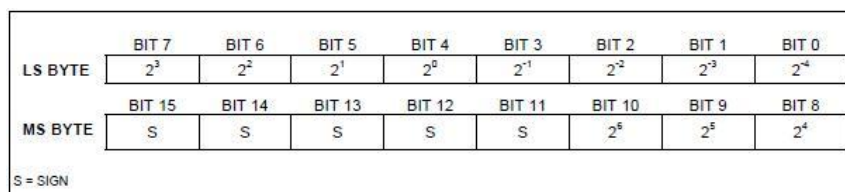
**Σημείωση:** αν ο δεν χρησιμοποιείται στον DS18B20 η λειτουργία συναγερμού, οι καταχωρητές TH και TL μπορούν να χρησιμοποιηθούν ως μνήμη γενικής χρήσης.



Εικόνα 7. 4: Σχηματική αναπαράσταση της μνήμης (scratchpad).

## 7.11. Καταχωρητής θερμοκρασίας

Τα δεδομένα θερμοκρασίας αποθηκεύονται σ' έναν 16-bit προσημασμένων τιμών καταχωρητή που έχει δεχτεί συμπλήρωμα κατά 2. Τα bits πρόσημου (S) μας δείχνουν αν η θερμοκρασία είναι θετική ή αρνητική: S = 0 αν θετική και S = 1 για αρνητική.

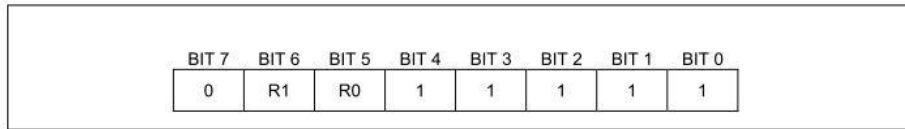


Εικόνα 7. 5: Διαμόρφωση bit του καταχωρητή θερμοκρασίας.

## 7.12. Καταχωρητής ρυθμίσεων

Το byte 4 της μνήμης περιέχει τις ρυθμίσεις των καταχωρητών, είναι οργανωμένο όπως φαίνεται στην **εικόνα 7.6**. Ο χρήστης μπορεί να ρυθμίσει την ανάλυση της μετατροπής του DS18B20 χρησιμοποιώντας το R0 και R1 bit σε αυτό τον καταχωρητή, όπως φαίνεται στην **εικόνα 7.7**. Η προεπιλεγμένη τιμή των R0 και R1 είναι 1 (12-bit μέγεθος). Το bit 7 και bit 0 έως 4 στον καταχωρητή

διαμόρφωσης προορίζονται για εσωτερική χρήση από τη συσκευή και δεν μπορούν να αλλαχτούν οι τιμές τους.



Εικόνα 7. 6: Διάγραμμα θέσεων μνήμης καταχωρητή ρυθμίσεων.

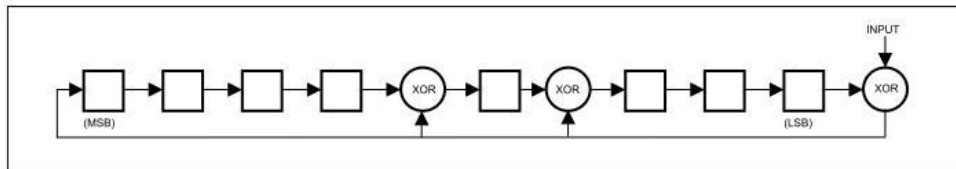
R1	R0	ΑΝΑΛΥΣΗ ΣΕ ΒΙΤ	ΜΕΓΙΣΤΟΣ ΧΡΟΝΟΣ ΜΕΤΑΤΡΟΠΗΣ	
0	0	9	93.75ms	( $t_{CONV}/8$ )
0	1	10	187.5ms	( $t_{CONV}/4$ )
1	0	11	375ms	( $t_{CONV}/2$ )
1	1	12	750ms	( $t_{CONV}$ )

Εικόνα 7. 7: Υπάρχει άμεση αύξηση στον χρόνο μετατροπής αν η ανάλυση αυξηθεί.

### 7.13. Δημιουργία κώδικα CRC (κυκλικός έλεγχος πλεονασμού)

Τα CRC<sup>21</sup> byte παρέχονται στο πλαίσιο του 64-bit κώδικα ROM του DS18B20 στο 9ο byte της μνήμης. Ο CRC κωδικός της ROM υπολογίζεται από τα πρώτα 56 bit της και περιέχεται στο πιο σημαντικό byte της ROM. Το CRC της μνήμης υπολογίζεται από τα δεδομένα που είναι αποθηκευμένα στη μνήμη και ως εκ τούτου αλλάζουν όταν τα δεδομένα διαβάζονται από την μνήμη. Το CRC παρέχει μια μέθοδο επικύρωσης των δεδομένων.

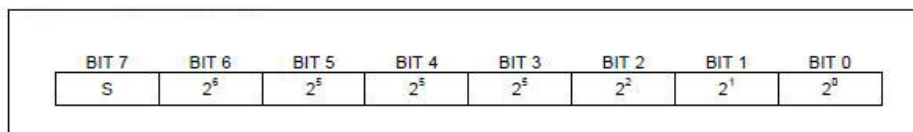
**Σημείωση:** για να βεβαιωθούμε ότι τα δεδομένα έχουν διαβάσει σωστά, πρέπει να υπολογίσει εκ νέου το CRC από τα δεδομένα. Αν η υπολογισμένη τιμή του CRC ταιριάζει με την τιμή του CRC που γνωρίζουμε, τα δεδομένα δεν έχουν λάθη.



Εικόνα 7. 8.: Σχηματική παρουσίαση υπολογισμού του CRC .

### 7.14. Σήμα συναγερμού

Μόνον τα bits 11 έως και 4 του καταχωρητή θερμοκρασίας χρησιμοποιούνται για τη σύγκριση τιμών των  $T_H$  και  $T_L$  (οι  $T_H$  και  $T_L$  είναι 8-bit καταχωρητές). Αν η θερμοκρασία είναι χαμηλότερη ή ίση με  $T_L$  ή μεγαλύτερη ή ίση με  $T_H$ , μια κατάσταση συναγερμού, ενεργοποιείται και μια σημαία συναγερμού ορίζεται. Η σημαία αυτή ενημερώνεται μετά από κάθε μέτρηση θερμοκρασίας. Ως εκ τούτου, αν η κατάσταση συναγερμού υποχωρήσει, η σημαία θα πρέπει να απενεργοποιηθεί μετά την επόμενη μέτρηση θερμοκρασίας

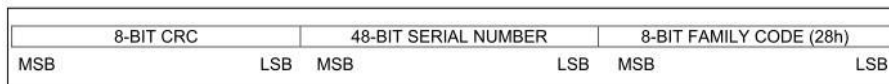


Εικόνα 7. 9: Διαμόρφωση bit των  $T_H$  και  $T_L$  καταχωρητών.

<sup>21</sup> τεχνική ανίχνευσης σφαλμάτων κατά τη διάρκεια μετάδοσης ή αποθήκευσης δεδομένων.

## 7.15. Κώδικας χάραξης Laser Rom μνήμης 64bit

Κάθε DS18B20 περιέχει ένα μοναδικό 64-bit κώδικα αποθηκευμένο στη μνήμη ROM. Τα λιγότερο σημαντικά 8 bits του ROM κώδικα περιέχουν τον 1-Wire κωδικό: 28h. Τα επόμενα 48 bits περιέχουν ένα μοναδικό σειριακό αριθμό. Τα πιο σημαντικά 8 bit περιέχουν ένα ελέγχου κυκλικού πλεονασμού (CRC) μεγέθους 1 byte που υπολογίζεται από τα πρώτα 56 bit του κώδικα ROM. Ο 64-bit κώδικας ROM και οι συναφείς λειτουργίες ROM λογικής ελέγχου επιτρέπουν τον DS18B20 να λειτουργεί ως μια συσκευή 1-Wire.

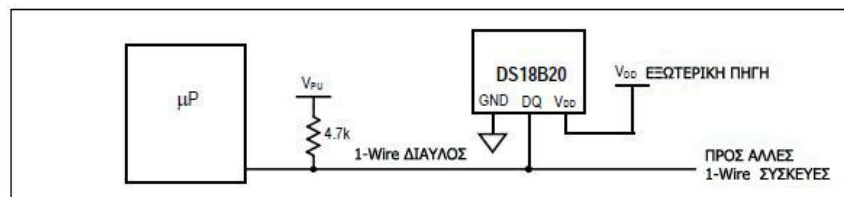


Εικόνα 7. 10: Σχηματική παρουσίαση του κώδικα χάραξης Laser.

## 7.16. Αρχικοποίηση

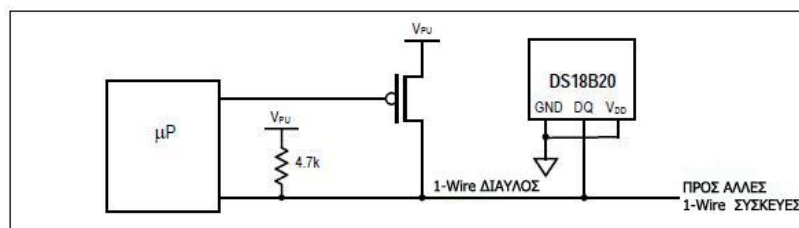
Ο DS18B20 μπορεί να τροφοδοτείται από εξωτερική πηγή στο VDD pin, ή να μπορεί να λειτουργήσει με «παρασιτική τάση», το οποίο του επιτρέπει να λειτουργήσει χωρίς εξωτερική τροφοδοσία. Η παρασιτική τάση είναι πολύ χρήσιμη για εφαρμογές που απαιτούν τηλεμετρία θερμοκρασίας ή όταν είναι πολύ περιορισμένος ο χώρος. Στην **εικόνα 7.11** δείχνει το κυκλώματα το οποίο «κλέβει» ενέργεια από το δίαυλο 1-Wire μέσω του pin DQ όταν ο δίαυλος είναι υψηλή σε υψηλή διακριτή στάθμη. Αποθηκεύεται ενέργεια στον παρασιτικό πυκνωτή ισχύος (CPP) για την παροχή ρεύματος όταν ο δίαυλος είναι σε χαμηλή διακριτή τιμή. Όταν ο DS18B20 χρησιμοποιεί παρασιτική τάση, το VDD pin πρέπει να συνδεθεί με την γείωση.

Η παρασιτική λειτουργία, μπορεί να παρέχει ικανοποιητική τάση για να τον DS18B20 για τις περισσότερες εργασίες όσο οι απαιτήσεις τάσης πληρούνται. Ωστόσο, όταν εκτελεί μετατροπές θερμοκρασίας ή αντιγραφή δεδομένων από τη μνήμη στην EEPROM, το ρεύμα λειτουργίας πρέπει να είναι 1.5mA. Για να είναι βέβαιο ότι ο DS18B20 έχει επαρκή παροχή ρεύματος, είναι απαραίτητο να υπάρχει μια ισχυρή pullup αντίσταση στο δίαυλο 1-Wire όποτε μετατροπές θερμοκρασίας λαμβάνουν χώρα ή δεδομένα αντιγράφονται από την μνήμη στην EEPROM. Αυτό μπορεί να επιτευχθεί με τη χρήση ενός MOSFET. Καμία άλλη δραστηριότητα δεν μπορεί να λάβει χώρα ενώ το η pullup αντίσταση είναι ενεργοποιημένη.



Εικόνα 7. 11: Τροφοδοσία με χρήση εξωτερικής τάσης.

Ο DS18B20 μπορεί επίσης να τροφοδοτείται από τη συμβατική εξωτερική παροχή ρεύματος για το VDD pin, όπως φαίνεται στην **εικόνα 7.12**. Το πλεονέκτημα αυτής της μεθόδου είναι ότι η pullup MOSFET δεν είναι απαραίτητη.



Εικόνα 7. 12: Τροφοδοσία με χρήση παρασιτικής τάσης.

**Σημείωση:** η χρήση της παρασιτικής τάσης δεν συνιστάται για θερμοκρασίες άνω των 100 ° C. Ο DS18B20 μπορεί να μην είναι σε θέση να διατηρήσει επικοινωνία λόγω της υψηλής διαρροής ρεύματος που μπορεί να υπάρχει σε αυτές τις θερμοκρασίες.

## 7.17. Τροφοδοσία και παρασιτική τάση λειτουργίας

Ένα άλλο χαρακτηριστικό του DS18B20 είναι η δυνατότητα να λειτουργεί χωρίς εξωτερική τροφοδοσία. Η τάση παρέχεται διάμεσο μιας αντίστασης pullup στο DQ pin όταν ο δίαυλος είναι σε υψηλή διακριτή στάθμη τάσης. Η τάση υψηλής διακριτής στάθμης φορτίζει επίσης έναν εσωτερικό πυκνωτή ( $C_{PP}$ ), ο οποίος παρέχει στη συνέχεια ισχύ όταν ο δίαυλος βρίσκεται σε χαμηλή στάθμη διακριτής τάσης. Αυτή η μέθοδος άντλησης τάσης από τον δίαυλο ονομάζεται «παρασιτική τάση». Ως εναλλακτική λύση, μπορεί να τροφοδοτείται κι από εξωτερική τροφοδοσία.

## 7.18. Ρυθμίσεις

Ο 1-Wire διαύλου έχει εξ ορισμού μόνο μία γραμμή δεδομένων. Αυτό επιτρέπει την «απελευθέρωση» της γραμμής δεδομένων όταν η συσκευή δεν μεταδίδει δεδομένα, ώστε ο δίαυλος να είναι διαθέσιμος για χρήση από άλλη συσκευή. Η 1-Wire θύρα του DS18B20 (DQ pin) είναι ένας ανοιχτός βρόχος. Ο 1-Wire δίαυλος απαιτεί μια εξωτερική αντίσταση pullup περίπου 5kΩ. Έτσι, η τιμή αναμονής για το δίαυλο 1-Wire είναι υψηλής στάθμης. Αν ο δίαυλος είναι στην χαμηλή στάθμη για περισσότερο από 480μs, στο δίαυλο θα γίνει επαναφορά (reset).

## 7.19. Αλληλουχία ανταλλαγής δεδομένων

Η αλληλουχία ανταλλαγής δεδομένων για την πρόσβαση στον DS18B20 έχει ως εξής:

- Βήμα 1. Αρχικοποίηση
- Βήμα 2. Εντολή ROM (απαιτούμενη σε ανταλλαγή δεδομένων)
- Βήμα 3. Εντολή λειτουργίας DS18B20 (απαιτούμενη σε ανταλλαγή δεδομένων)

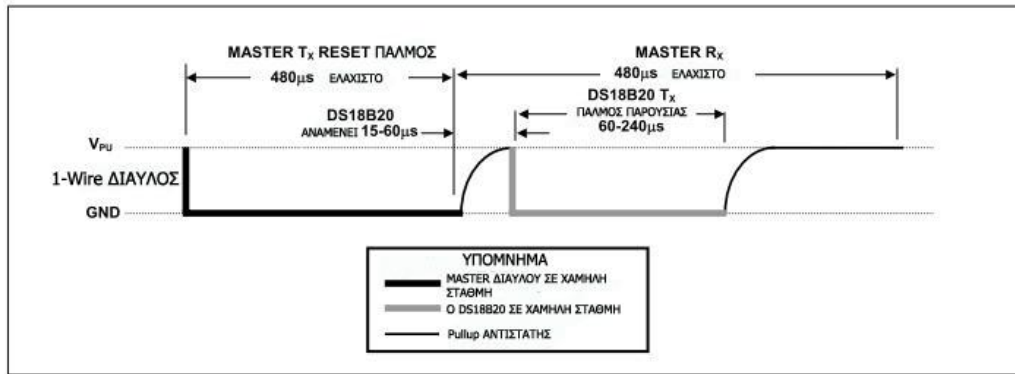
Είναι πολύ σημαντικό να ακολουθηθεί αυτήν την αλληλουχία κάθε φορά που στον DS18B20 γίνεται πρόσβαση. Αλλιώς ο DS18B20 δεν θα λειτουργήσει σωστά αν τα βήματα δεν πραγματοποιηθούν με τη σειρά αυτή ή λείπουν κάποια. Όλες οι ανταλλαγές δεδομένων αρχίζουν στο δίαυλο 1-Wire με μια αλληλουχία αρχικοποίησης. Η αλληλουχία αρχικοποίησης αποτελείται από έναν παλμό reset που μεταδίδεται από τον master μονάδα στον δίαυλο και ακολουθείται από παλμό παρουσίας που μεταδίδεται από τις μονάδες slave.

**Σημείωση:** ο παλμός παρουσίας δίνει τη δυνατότητα στον master του διαύλου να γνωρίζει ότι οι συσκευές slave είναι στο δίαυλο και είναι έτοιμες να λειτουργήσουν.

## 7.20. Διαδικασία αρχικοποίησης - παλμοί reset και παρουσίας

Κατά τη διάρκεια αλληλουχίας αρχικοποίησης μεταδίδει (TX) ο master του διαύλου, παλμό επαναφοράς θέτοντας τον 1-Wire δίαυλο σε χαμηλή διακριτή στάθμη για τουλάχιστον 480μs. Στη συνέχεια απελευθερώνεται ο δίαυλος και αλλάζει σε λειτουργία λήψης (RX). Όταν απελευθερώνεται, η 5kΩ αντίσταση pullup θέτει τον 1-Wire δίαυλο σε υψηλή διακριτή στάθμη. Όταν ο DS18B20 ανιχνεύει αυτό την επερχόμενη αλλαγή, περιμένει 15μs με 60μs και στη συνέχεια μεταδίδει έναν παλμό παρουσίας θέτοντας τον 1-Wire δίαυλο σε χαμηλή διακριτή στάθμη για 60μs με 240μs.





Εικόνα 7. 13: Διάγραμμα αλληλουχίας αρχικοποίησης.

## 7.21. Σήματα πρωτοκόλλου 1-Wire

Ο DS18B20 χρησιμοποιεί το 1-Wire πρωτόκολλο επικοινωνίας για την εξασφάλιση της ακεραιότητας των δεδομένων. Τα είδη σήματος που ορίζονται από το παρόν πρωτόκολλο: παλμός reset, παλμός παρουσίας, γράψε 0, γράψε 1, διάβασε 0, και να διάβασε 1 (reset pulse, presence pulse, write 0, write 1, read 0, read 1). Ο master ξεκινά όλα αυτά τα σήματα, με εξαίρεση τον παλμό παρουσίας.

## 7.22. Εντολές διαχείρισης μνήμης ROM

Εφόσον η μονάδα master έχει εντοπίσει έναν παλμό παρουσίας, μπορεί να δώσει μια εντολή ROM. Οι εντολές αυτές λειτουργούν σε συνεργασία με τους μοναδικούς 64-bit κώδικες ROM της κάθε μονάδας slave και επιτρέπει στον master να ξεχωρίσει μια συγκεκριμένη συσκευή αν πολλές είναι παρόντες στο δίαυλο 1-Wire. Οι εντολές αυτές επιτρέπουν επίσης να καθοριστεί αν κάποια συσκευή έχει κατάσταση συναγερμού. Υπάρχουν πέντε εντολές ROM, και κάθε εντολή έχει 8 bits μήκος.

Όνομασία	HEX Τιμή	Περιγραφή	Ανάλυση
<b>SEARCH ROM</b>	<b>F0h</b>	<b>Αναζήτηση</b>	Με αυτήν την εντολή η μονάδα master σαρώνει τον δίαυλο 1-Wire καταγράφει τους μοναδικούς κωδικούς 64 bit κάθε slave μονάδας. Όποτε η χρήση των μονάδων αποκτάει πιο κατανεμημένη δομή.
<b>READ ROM</b>	<b>33h</b>	<b>Ανάγνωση</b>	Αυτή η εντολή μπορεί να χρησιμοποιηθεί μόνον όταν υπάρχει ένας slave στο δίαυλο. Αυτό επιτρέπει στον master του διαύλου να διαβάσει τον 64-bit ROM κώδικα του slave, χωρίς να χρησιμοποιεί τη διαδικασία ROM αναζήτησης.
<b>MATCH ROM</b>	<b>55h</b>	<b>Ταυτοποίηση</b>	Η εντολή αυτή ακολουθείται από έναν 64-bit κώδικα ROM η ακολουθία αυτή επιτρέπει στον master του διαύλου για την ταυτοποίηση μιας συγκεκριμένης συσκευής slave σε ένα multidrop ή single-drop δίαυλο. Μόνο ο slave που ταιριάζει ακριβώς στον 64-bit κώδικα ROM ταυτοποιείται και εν τέλει δέχεται εκ νέου εντολές.
<b>SKIP ROM</b>	<b>CCh</b>	<b>Παράληψη μνήμης Rom</b>	Ο master μπορεί να χρησιμοποιήσει αυτή την εντολή για να ταυτοποιήσει όλες τις συσκευές στο δίαυλο ταυτόχρονα, χωρίς να στείλει οποιαδήποτε δεδομένα κωδικού ROM.
<b>ALARM SEARCH</b>	<b>ECh</b>	<b>Αναζήτηση συναγερμού</b>	Η λειτουργία αυτής της εντολής είναι ταυτόσημη με τη λειτουργία της εντολής SEARCH ROM [F0h] εκτός από το ότι μόνο οι slave με ειδικής ένδειξη συναγερμού θα ανταποκριθούν. Αυτή η εντολή επιτρέπει στην master συσκευή να προσδιορίσετε εάν κάποια slave υπέστη κατάσταση συναγερμού κατά τη διάρκεια της πιο πρόσφατης μετατροπής θερμοκρασίας.

Πίνακας 6: Εντολές διαχείρισης μνήμης ROM.

## 7.23. Εντολές λειτουργιών του DS18B20

Αφ' ότου ο master του διαύλου έχει χρησιμοποιήσει μια εντολή ROM για την ταυτοποίηση μιας slave με τον οποία επιθυμεί να επικοινωνήσει, μπορεί να δώσει μια από τις εντολές λειτουργίας. Οι εντολές αυτές επιτρέπουν στον master να γράψει και να διαβάσει από τη μνήμη της slave, να εκτελέσει μετατροπές θερμοκρασίας όπως και τον καθορισμό του τύπου παροχής ρεύματος.

Όνομασία	HEX Τιμή	Περιγραφή	Ανάλυση
<b>CONVERT T</b>	<b>44h</b>	<b>Μετατροπή (Θερμοκρασίας) T</b>	Ξεκινά την μετατροπή θερμοκρασίας. Μετά τη μετατροπή, τα προκύπτοντα θερμικά δεδομένα αποθηκεύονται στον καταχωρητή 2 byte θερμοκρασίας στη μνήμη και η μονάδα επιστρέφει σε κατάσταση αδράνειας χαμηλής ισχύος.
<b>READ SCRATCHPAD</b>	<b>BEh</b>	<b>Ανάγνωση μνήμης</b>	Αυτή η εντολή επιτρέπει στον master να διαβάσει το περιεχόμενο της μνήμης. Η μεταφορά των δεδομένων ξεκινά με το λιγότερο σημαντικό bit του byte 0 και συνεχίζει μέχρι το 9ο byte (byte 8 - CRC).
<b>WRITE SCRATCHPAD</b>	<b>4Eh</b>	<b>Εγγραφή μνήμης</b>	Αυτή η εντολή επιτρέπει στον master να γράψει 3 byte δεδομένων στην μνήμη του. Το πρώτο byte δεδομένων που εγγράφεται στον καταχωρητή TH (byte 2 της μνήμης), το δεύτερο byte στον TL καταχωρητή (byte 3 μνήμης), και το τρίτο byte γράφεται στον καταχωρητή διαμόρφωσης (byte 4 μνήμης). Στα δεδομένα το λιγότερο σημαντικό bit πρέπει να διαβιβάζεται πρώτο.
<b>COPY SCRATCHPAD</b>	<b>48h</b>	<b>Αντιγραφή μνήμης</b>	Αντιγράφει τα περιεχόμενα των καταχωρητών TH, TL, και της διαμόρφωσης δεδομένων από την μνήμη στην EEPROM μνήμη.
<b>RECALL E<sup>2</sup></b>	<b>B8h</b>	<b>Ανάκληση E<sup>2</sup></b>	Ανακαλεί τις τιμές των καταχωρητών TH, TL, και της διαμόρφωσης των δεδομένων από την EEPROM στην μνήμη.
<b>READ POWER SUPPLY</b>	<b>B4h</b>	<b>Ανάγνωση τάσης τροφοδοσίας</b>	Η συσκευή master αποστέλλει αυτή την εντολή ακολουθούμενη από μια χρονοσχισμή ανάγνωσης για να κριθεί εάν στο δίαυλο χρησιμοποιούν κάποιες slave μονάδες παρασιτική τροφοδοσία

Πίνακας 7: Εντολές λειτουργίας του DS18B20.



## 7.24. Αισθητήρας υγρασίας Honeywell HIH-4030

Ο HIH-4030 μετρά την σχετική υγρασία (% RH) και θα παρέχει, ως έξοδο αναλογικού εύρους τάση. Χρησιμοποιεί χωρητικά θερμοσκληρυντικά στοιχεία υψηλής ακρίβειας και ενσωματώνει κύκλωμα επεξεργασίας σήματος. Βασίζεται στην ηλεκτρική χωρητικότητα, η ικανότητα δύο κοντινών ηλεκτρικών αγωγών να δημιουργούν ένα ηλεκτρικό πεδίο μεταξύ τους. Ο αισθητήρας αποτελείται από δύο μεταλλικές πλάκες με μη αγώγιμο φιλμ πολυμερούς<sup>22</sup> μεταξύ τους. Το φιλμ συλλέγει την υγρασία από τον αέρα και η υγρασία που ανιχνεύει προκαλεί μικρές αλλαγές στην τάση μεταξύ των δύο πλακών. Οι αλλαγές στην τάση θα μετατραπούν σε ψηφιακές ενδείξεις δείχνει το ποσοστό της υγρασίας στον αέρα.

### 7.25. Χαρακτηριστικά:

- Χυτό θερμοσκληρυντικό πλαστικό περίβλημα
- Σχεδόν γραμμική απόκριση τάσης εξόδου ανά μονάδα %RH (% σχετικής υγρασίας)
- Χαμηλή κατανάλωση των 200  $\mu$ A,
- Αυξημένη ακρίβεια ( $\pm 3,5\%$  και 0 έως 100 %RH )
- Χαμηλός χρόνος απόκρισης ( 5s ) με υστέρηση 3%
- Σταθερή επίδοση , μηδαμινή ανάγκη για αναρρύθμιση
- Ανθεκτικός σε διάφορα χημικά
- Συμβατός με πολλούς μικροελεγκτές
- Απαιτούμενη τροφοδοσία: 5 V
- Θερμοκρασία λειτουργίας: -40 έως 90 ° C
- Διαστάσεις: 4,14 x 2,67 x 8,59 mm



Εικόνα 7. 14: Ο HIH-4030 σε breakout πλακέτα.

### 7.26. Αρχή λειτουργίας

Αυτό το σύστημα βασίζεται στην ηλεκτρική χωρητικότητα, η ικανότητα δύο κοντινών ηλεκτρικών αγωγών (μεταλλικές πλάκες) να δημιουργούν ηλεκτρικό πεδίου ανάμεσα τους. Ο αισθητήρας αποτελείται από δύο μεταλλικές πλάκες με μη αγώγιμο φιλμ πολυμερούς μεταξύ τους. Η ταινία συλλέγει την υγρασία από τον αέρα. Η υγρασία προκαλεί μικρές αλλαγές στην τάση μεταξύ των δύο πλακών. Οι αλλαγές στην τάση θα μετατραπούν σε ψηφιακές ενδείξεις που δείχνουν το ποσό της υγρασίας στον αέρα.

<sup>22</sup> είδος πλαστικής μεμβράνης με υψηλή αντοχή στις διακυμάνσεις θερμοκρασίας και χαμηλή απορροφητικότητα στο νερό.

## 7.27. Χωρητικοί αισθητήρες υγρασίας

Ένα χωρητικός αισθητήρας υγρασίας λειτουργεί όπως και οι πλάκες του πυκνωτή. Πάνω στο χαμηλότερο ηλεκτρόδιο είναι απλωμένο αγωγίμο υπόστρωμα, συχνά ένα κεραμικό υλικό. Ένα λεπτό πολυμερές στρώμα υγροσκοπικό ενεργεί ως διηλεκτρικό, και πάνω από αυτό είναι η πάνω πλάκα, η οποία ενεργεί ως δεύτερο ηλεκτρόδιο, αλλά η οποία επίσης επιτρέπει στους υδρατμούς να περάσουν μέσα από αυτό, στο πολυμερές. Τα μόρια υδρατμών του νερού εισέρχονται ή εξέρχονται από το υγροσκοπικό πολυμερές μέχρι την περιεκτικότητα σε υδρατμούς να είναι σε ισορροπία με τον αέρα του περιβάλλοντος ή του αερίου. Η διηλεκτρική αντοχή του πολυμερούς είναι ανάλογη με την περιεκτικότητα σε υδρατμούς. Με τη σειρά της η διηλεκτρική αντοχή επηρεάζει την χωρητικότητα, η οποία μετράται δέχεται επεξεργασία για να δώσει την τιμή της σχετικής υγρασίας.

Η αλλαγή στη χωρητικότητα είναι συνήθως 0,2 με 0,5 pF για 1% RH αλλαγή. Οι χωρητικοί αισθητήρες χαρακτηρίζονται από χαμηλό συντελεστή θερμοκρασίας (temperature coefficient)<sup>23</sup>, ικανότητα να λειτουργούν σε υψηλές θερμοκρασίες (έως 200 ° C), η πλήρης ανάκαμψη από τη συμπίκνωση, καθώς και εύλογη ανθεκτικότητα σε ατμούς χημικών. Οι χωρητικοί αισθητήρες περιορίζονται από την απόσταση. Λόγω της χωρητικής επίδρασης του καλωδίου σύνδεσης σε σχέση με τις σχετικά μικρές μεταβολές χωρητικότητας του αισθητήρα. Ένα πρακτικό όριο είναι <3 μέτρα.

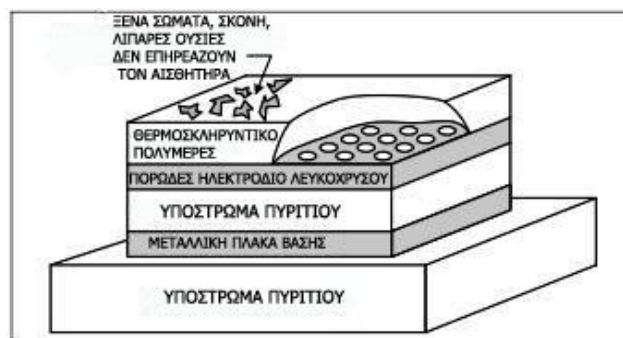
## 7.28. Αντοχή κοντά στο σημείο δρόσου

Οι χωρητικοί αισθητήρες παρέχουν διακριτές μεταβολές σήματος σε χαμηλές τιμές σχετικής υγρασίας, παραμένουν σταθεροί σε μακροχρόνια χρήση, και έχουν ελάχιστη ολίσθηση. Οι χωρητικοί αισθητήρες σχετικής υγρασίας ανιχνεύουν άμεσα της αλλαγές στον σχετικό κορεσμό που εμφανίζεται ως μεταβολή της χωρητικότητας με γρήγορη απόκριση υψηλή γραμμικότητα χαμηλή καθυστέρηση και άριστη σταθερότητα στο βάθος του χρόνου. Ο σχετικός κορεσμός είναι ο ίδιος όσο και η σχετική υγρασία όταν ο αισθητήρας βρίσκεται σε θερμοκρασία περιβάλλοντος. Και επειδή αυτό συμβαίνει συνήθως η καταμέτρηση της χωρητικότητας του αισθητήρα μας παρέχει τη μεταβολή στην σχετική υγρασία.

Οι χωρητικοί αισθητήρες σχετικής υγρασίας είναι κυρίαρχοι σε ατμοσφαιρικές και βιομηχανικές μετρήσεις γιατί είναι οι μοναδικοί οι οποίοι μπορούν να μετρήσουν το πλήρες φάσμα των μεταβολών της υγρασίας, με χαμηλές απώλειες λόγω θερμικών φαινομένων.

## 7.29. Ιδιαίτερα χαρακτηριστικά του ΗΙΗ-4030

Αποτελείται από ένα υπόστρωμα πάνω στο οποίο ένα λεπτό στρώμα του πολυμερούς που έχει τοποθετηθεί ανάμεσα σε δύο ηλεκτρόδια αγωγιμότητας. Η πάνω μεταλλική επιφάνεια είναι επικαλυμμένη με ένα πορώδες μεταλλικό ηλεκτρόδιο για να το προστατεύσει από τη μόλυνση και την έκθεση σε συμπίκνωση. Το υπόστρωμα είναι πυριτίου. Η σταδιακή αλλαγή της διηλεκτρικής σταθεράς ενός χωρητικό αισθητήρα υγρασίας είναι σχεδόν ευθέως ανάλογη προς την σχετική υγρασία του περιβάλλοντος χώρου.



Εικόνα 7. 15: Βασική δομή αισθητήρα.

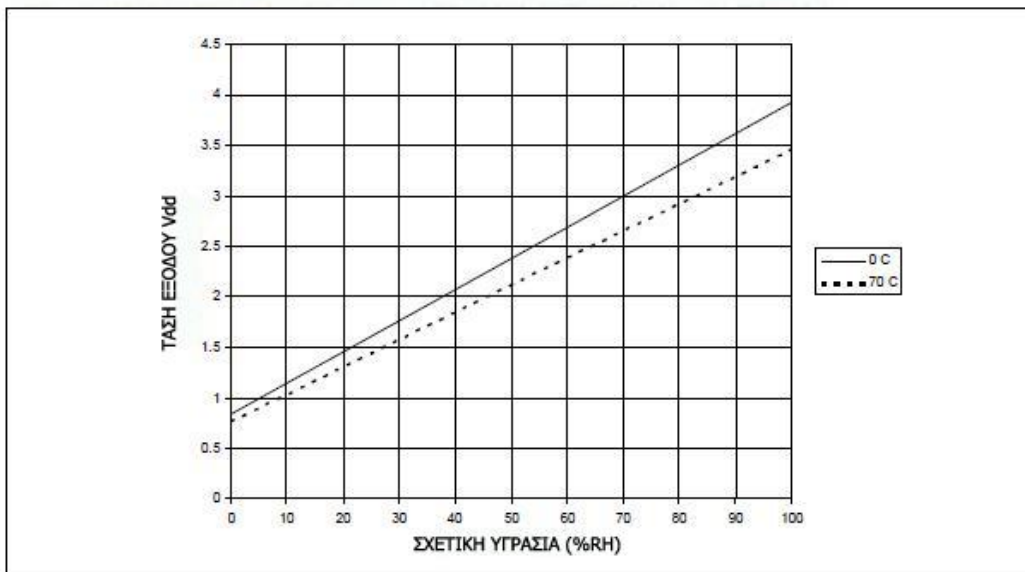
<sup>23</sup> συντελεστής μεταβολής αντίστασης ανά βαθμό μεταβολής θερμοκρασίας.

### 7.30. Θερμοσκληρυντικό περίβλημα

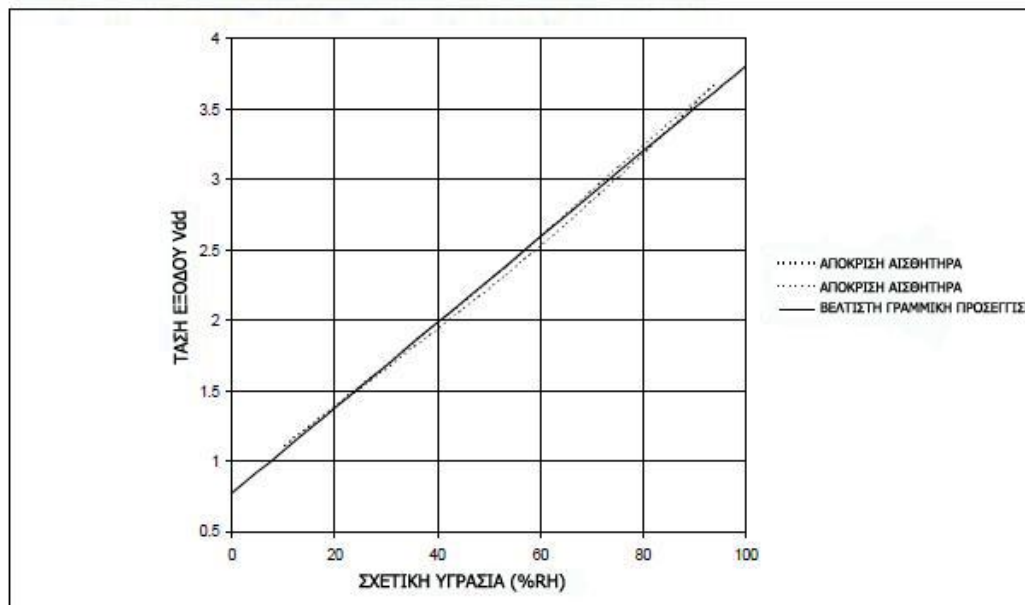
Το θερμοσκληρυντικό πλαστικό είναι ένα είδους πολυμερούς υλικού τ' οποίο η σκλήρυνση του είναι μη αναστρέψιμη. Τα θερμοσκληρυντικά είναι γενικά ανθεκτικότερα από τα θερμοπλαστικά υλικά λόγω της τρισδιάστατης δομής των μορίων του , που συνθέτουν ένα «δίκτυο δεσμών». Επίσης είναι καταλληλότερα για εφαρμογές με υψηλές θερμοκρασίες περιβάλλοντος.

### 7.31. Υψηλή γραμμικότητα και επιδόσεις

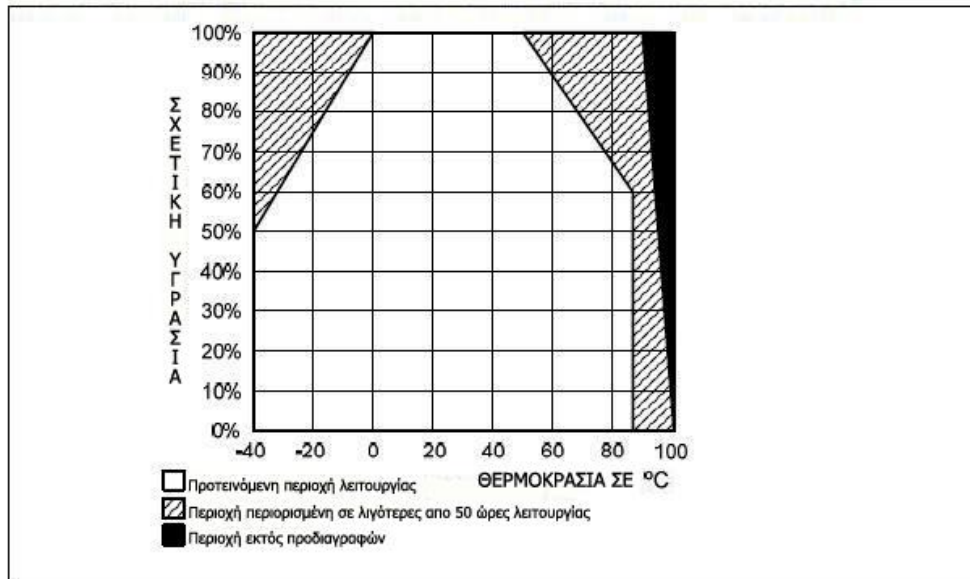
Ο αισθητήρας HIH-4030 παρουσιάζει ιδιαίτερα γραμμική απόκριση στη τάση εξόδου, ιδιότητα που μας εγγυάται ακρίβεια στις μετρήσεις λόγω χαμηλών διακυμάνσεων. Επίσης η απόδοση του παραμένει σταθερή για θερμοκρασίες μεταξύ 0 και 50 °C.



Εικόνα 7. 16: Τάση εξόδου σε συνάρτηση με την σχετική υγρασία στους 0 °C, 70 °C και 5 V.



Εικόνα 7. 17: Τάση εξόδου σε συνάρτηση με την σχετική υγρασία στους 25 °C και 5 V.



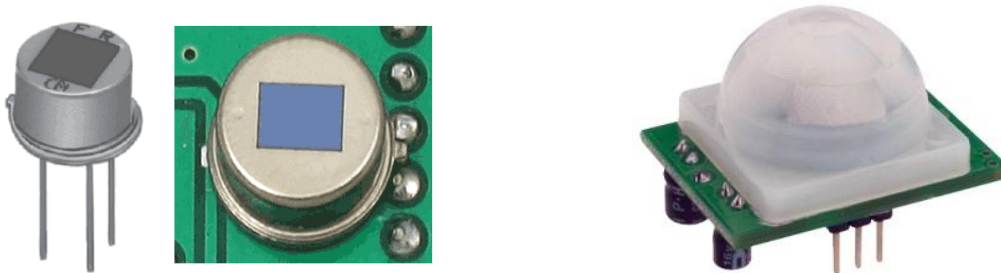
Εικόνα 7. 18: Οι επιδόσεις σε διαφορετικές περιοχές θερμοκρασίας.

## 7.32. Αισθητήρας PIR Parallax 555 - 280227

Ο PIR αισθητήρας (Passive Infra-Red μετ: Υπερύθρου Παθητικός) είναι μια πυροηλεκτρική συσκευή<sup>24</sup> που ανιχνεύει κίνηση από την μέτρηση των μεταβολών στα επίπεδα της υπέρυθρης ακτινοβολίας (θερμότητα) που εκπέμπονται από τα γύρω αντικείμενα. Η κίνηση μπορεί να ανιχνευτεί ελέγχοντας για μια ξαφνική αλλαγή στα μοτίβα της υπέρυθρης ακτινοβολίας γύρω από τον αισθητήρα. Όταν ανιχνευτεί κίνηση από τον αισθητήρα, παράγει έξοδο υψηλής διακριτής στάθμης<sup>25</sup> στο pin εξόδου. Αυτό το σήμα διακριτής στάθμης μπορεί να διαβαστεί από ένα μικροελεγκτή και περαιτέρω να αναλυθεί και να χρησιμοποιηθεί.

### 7.33. Χαρακτηριστικά:

- Εύρος ανίχνευσης μέχρι 6 μέτρα
- Έξοδος ενός bit υψηλή / χαμηλή διακριτής τιμής
- Jumper επιλογής μονής ή συνεχόμενης εξόδου (σε bit)
- 3-pin SIP προσαρμογέα που είναι έτοιμο για breadboard
- Μικρό μέγεθος
- Συμβατό με πολλούς μικροελεγκτές
- Απαιτούμενη τροφοδοσία: 3,3 - 5 VDC > 3 mA
- Θερμοκρασία λειτουργίας: 0 έως 50 ° C
- Διαστάσεις: 32,2 x 24,3 x 25,4 mm



Εικόνα 7. 19: Ο αισθητήρας PIR στεγάζεται σε ένα ερμητικά σφραγισμένο μεταλλικό κέλυφος ώστε ο θόρυβος, θερμοκρασία και η υγρασία να μην το επηρεάζει.

### 7.34. Περιγραφή των pin

Pin	Όνομα	Λειτουργία
-	<b>GND</b>	Γείωση 0 V
+	<b>Vin</b>	Είσοδος 3.3 to 5 VDC
<b>OUT(Signal)</b>	<b>Έξοδος</b>	Έξοδος σε μικροελεγκτή

Πίνακας 8: Τα pin του PIR Parallax 555 - 280227.

<sup>24</sup> η ικανότητα ορισμένων υλικών να δημιουργούν προσωρινή τάση όταν θερμαίνονται ή ψύχονται

<sup>25</sup> ακολουθία στο πεδίο του χρόνου διακριτών ακεραίων τιμών που καθένα απ' αυτά ονομάζεται δείγμα

### 7.35. Ρύθμιση jumper

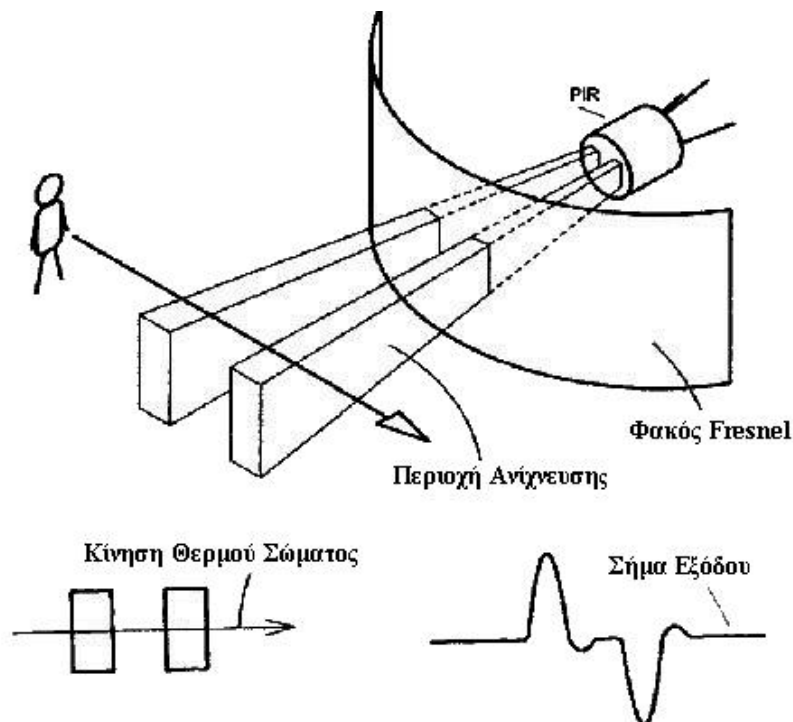
Σύμβολο	Περιγραφή
<b>H</b>	Έξοδος παραμένει σε υψηλή διακριτή στάθμη όταν έχουμε συνεχόμενη ανίχνευση κίνησης. Έξοδος σε χαμηλή διακριτή στάθμη ( απουσία κίνησης ).
<b>L</b>	Έξοδος εναλλάσσεται μεταξύ υψηλής και χαμηλής διακριτής στάθμης όταν έχουμε ανίχνευση κίνησης. Συνεχόμενη ανίχνευση κίνησης προκαλεί επαναλαμβανόμενους παλμούς υψηλής \ χαμηλής στάθμης. Έξοδος σε χαμηλή διακριτή στάθμη.

Πίνακας 9: Οι θέσεις του jumper ρυθμίσεων του H1H-4030.

### 7.36. Θεωρία λειτουργίας

Πυροηλεκτρικές συσκευές, όπως ο αισθητήρας PIR, περιέχουν στοιχεία που κατασκευάζονται από κρυσταλλικό υλικό που παράγει ηλεκτρικό φορτίο όταν εκτίθεται σε υπέρυθη ακτινοβολία. Οι αλλαγές στο ποσό της υπέρυθρης που προσπίπτουν στο στοιχείο αλλάζουν την τάση που δημιουργείται και που μετρώνται από έναν ενισχυτή που υπάρχει στην πλακέτα. Η συσκευή περιέχει ένα ειδικό φίλτρο που ονομάζεται φακός Fresnel, ο οποίος επικεντρώνει τα υπέρυθρα σήματα πάνω στο στοιχείο. Καθώς τα υπέρυθρα σήματα του περιβάλλοντος αλλάζουν γρήγορα, ο ενισχυτής που υπάρχει στην πλακέτα τροφοδοτεί την έξοδο για να δείξει κίνηση.

Ο αισθητήρας PIR έχει δύο υποδοχές, κάθε υποδοχή είναι κατασκευασμένη από ειδικό υλικό που είναι ευαίσθητα στην υπέρυθη ακτινοβολία. Όταν ο αισθητήρας είναι σε κατάσταση αδράνειας, οι δύο υποδοχές ανιχνεύουν το ίδιο ποσό υπέρυθρης ακτινοβολίας που ακτινοβολείται από το δωμάτιο ή τοίχους ή σε εξωτερικούς χώρους. Όταν ένα θερμό σώμα όπως ένας άνθρωπος ή ένα ζώο περάσει, το πρώτο μισό του αισθητήρα PIR ( η μια υποδοχή ), θα ανιχνεύσει μια θετική ( τάσης ) αλλαγή διαφορά μεταξύ των υποδοχών. Όταν το θερμό σώμα απομακρυνθεί από την περιοχή ανίχνευσης, το αντίθετο συμβαίνει, σύμφωνα με το οποίο ο αισθητήρας παράγει μια αρνητική μεταβολή τάσης. Αυτές οι αλλαγές στις υποδοχές είναι το στοιχείο που μας αποδεικνύει ότι έχει εντοπίσει κίνηση ο αισθητήρας.



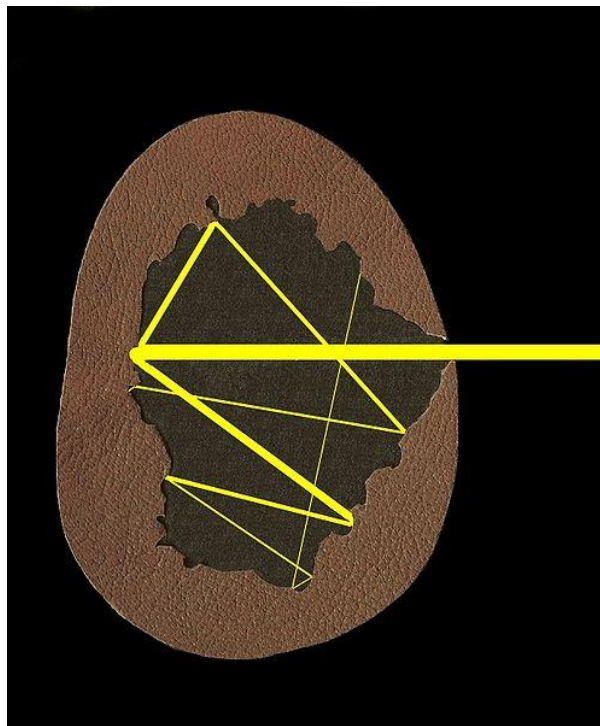
Εικόνα 7. 20: Βασικό σχήμα της αρχής λειτουργίας.

### 7.37. Ανάλυση της θεωρίας λειτουργίας

*«Όλα τα αντικείμενα εκπέμπουν ακτινοβολία, γνωστή ως ακτινοβολία μέλαν σώματος.»*

**Μέλαν σώμα:** ένα ιδανικό σώμα που δεν αντανακλά καθόλου φως (ή άλλης μορφής ηλεκτρομαγνητική ακτινοβολία) ούτε αφήνει το φως να το διαπεράσει. Ωστόσο το ίδιο το σώμα εκπέμπει κάποια ακτινοβολία, η οποία εξαρτάται από την θερμοκρασία στην οποία βρίσκεται.

Το φυσικό «αντικείμενο» που προσεγγίζει καλύτερα το μέλαν σώμα, δεν είναι καν σώμα, αλλά μια μικρή οπή σε ένα κοίλο σώμα (όπως π.χ. η είσοδος μιας σπηλιάς). Το φως που μπαίνει μέσα στην κοιλότητα από την οπή θα ανακλαστεί πολλές φορές πάνω στα τοιχώματα της κοιλότητας και κάθε φορά ένα μέρος του θα απορροφάται από αυτά. Η πιθανότητα για ένα τμήμα της ακτινοβολίας που μπήκε μέσα στην κοιλότητα από την οπή να ξαναβγεί από αυτήν είναι πολύ μικρή, αν η οπή είναι αρκετά μικρή σε σχέση με την κοιλότητα, πράγμα που σημαίνει ότι μόνο ένα πολύ μικρό μέρος από το προσπίπτον φως «ανακλάται» από την οπή, ενώ το υπόλοιπο έχει απορροφηθεί. Αυτό συμβαίνει ανεξάρτητα από το υλικό των τοιχωμάτων και το μήκος κύματος της προσπίπτουσας ακτινοβολίας, διότι, καθώς τα στερεά σώματα έχουν συνεχές φάσμα εκπομπής και απορρόφησης, όλα τα μήκη κύματος σταδιακά θα απορροφηθούν. Δεδομένου ότι το φως που παίρνουμε πίσω είναι αμελητέο, η μόνη ακτινοβολία που θα παίρνουμε από την οπή είναι η θερμική ακτινοβολία που παράγεται στο εσωτερικό της κοιλότητας και εξαρτάται μόνο από την θερμοκρασία της, υπό την προϋπόθεση ότι αυτή βρίσκεται σε θερμική ισορροπία.



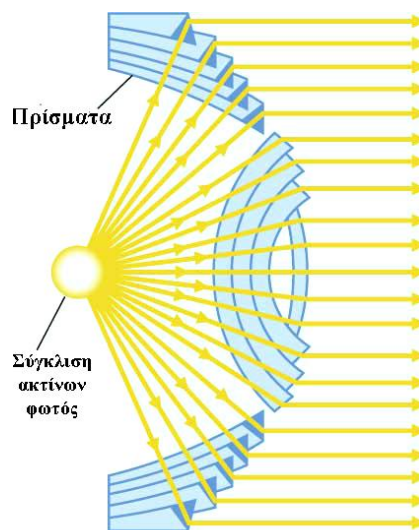
**Εικόνα 7. 21:** Το φως που εισέρχεται στην κοιλότητα από μια μικρή οπή, έπειτα από πολλαπλές αντανακλάσεις απορροφάται σχεδόν ολοκληρωτικά από τα τοιχώματα.

Ο παθητικός αισθητήρας υπέρυθρων (Passive InfraRed) είναι μια ηλεκτρονική συσκευή που μετρά το φως στο φάσμα του υπέρυθρου που ακτινοβολείται από τα αντικείμενα στο οπτικό πεδίο της. Η υπέρυθρη ακτινοβολία είναι αόρατη στο ανθρώπινο μάτι αλλά μπορεί να ανιχνευθεί από κατάλληλες ηλεκτρονικές συσκευές. Η σημασία του όρου παθητικός σε αυτήν την περίπτωση σημαίνει ότι ο αισθητήρας PIR δεν εκπέμπει υπέρυθρη ακτίνα αλλά μόνο παθητικά δέχεται την εισερχόμενη υπέρυθρη ακτινοβολία. Δεν έχουμε την δυνατότητά να το ανιχνεύσουμε οπτικά πέρα από το κόκκινο επειδή αυτό το χρώμα αντιπροσωπεύει το χαμηλότερο ενεργειακό επίπεδο που τα μάτια μας μπορούν να αισθανθούν προτού να γίνει αόρατο. Κατά συνέπεια, τα υπέρυθρα σήματα κάτω από το ενεργειακό επίπεδο του κόκκινου χρώματος είναι αόρατα και χρειαζόμαστε τον PIR.



### 7.38. Φακός Fresnel

Για να γίνει πιο αποδοτική η ανίχνευση της κίνησης γίνεται χρήση ενός ειδικού φακού του Fresnel. Ο φακός Fresnel μειώνει την ποσότητα του υλικού που απαιτείται σε σύγκριση με ένα συμβατικό σφαιρικό φακό διαιρώντας το φακό σε ένα σύνολο ομόκεντρων δακτυλιοειδών τμημάτων γνωστά ως «ζώνες Fresnel», τα οποία είναι θεωρητικά απεριόριστες. Σε κάθε μία από αυτές τις ζώνες, το συνολικό πάχος του φακού μειώνεται, διαιρώντας τη συνεχή επιφάνεια ενός κανονικού φακού σε μια σειρά επιφανειών της ίδιας καμπυλότητας, δημιουργώντας έτσι σταδιακή ασυνέχεια μεταξύ τους. Στην πραγματικότητα ένας φακός Fresnel μπορεί να θεωρηθεί ως μια σειρά από πρίσματα τοποθετημένα σε κύκλο, με το πιο απότομο από τα πρίσματα στις άκρες και μια σχεδόν επίπεδη κυρτότητα στο κέντρο.



Εικόνα 7. 22: Εδώ φαίνεται πως το κάθε πρίσμα επιφέρει σύγκλιση στο ίδιο σημείο.

Οι φακοί Fresnel είναι συνήθως από γυαλί ή πλαστικό: το μέγεθός τους ποικίλλει από μεγάλους (παλιούς ιστορικούς φάρους, στο μέγεθος ενός μέτρου), μέσο μέγεθος (προβολείς OHP), μικρό μέγεθος (οθόνες TLR / SLR φωτογραφικής μηχανής). Σε πολλές περιπτώσεις είναι πολύ λεπτοί και επίπεδοι, σχεδόν ευλύγιστοι, με πάχος από 1 έως 5 χιλιοστά φάσμα.

### 7.39. Φακός Fresnel τύπου NL-11NH

Ο φακός *Fresnel* που υπάρχει στον PIR αισθητήρα είναι τύπου *NL-11NH* είναι κατασκευασμένος από υψηλής πυκνότητας πολυαιθυλένιο έχει 17 οπτικές ζώνες, εστιακό μήκος<sup>26</sup> 12,9 χιλιοστά και θερμοκρασία λειτουργίας: από -20 έως +70 ° C.

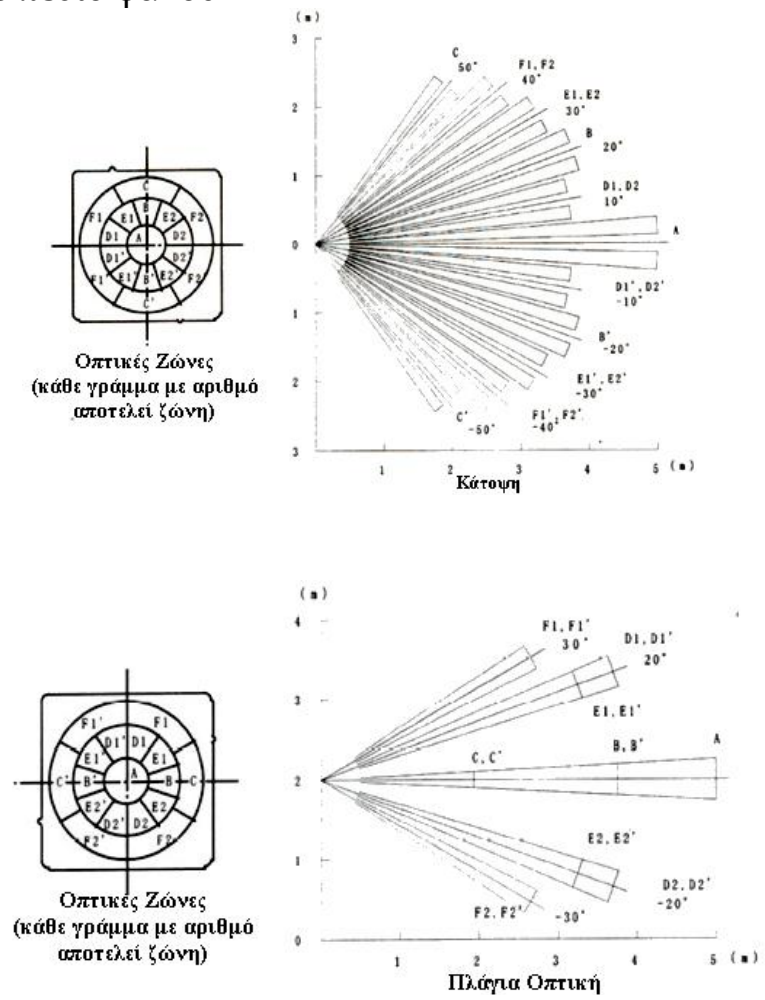


Εικόνα 7. 23: Όψεις του φακού Fresnel NL-11NH.

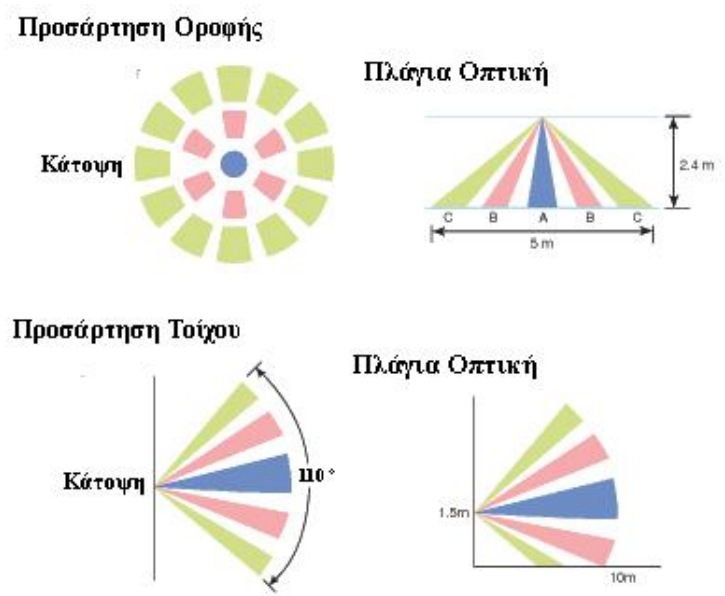
<sup>26</sup> μέτρο για το πόσο έντονα συγκλίνει (εστίαση) ή αποκλίνει (διασκορπίζει) το φως ένας φακός. Ένας φακός με μικρό εστιακό μήκος έχει μεγαλύτερη οπτική δύναμη από έναν με ένα μεγάλο εστιακό μήκος



### 7.40. Οπτικό πεδίο φακού



Εικόνα 7. 24: Διαγράμματα της πλάγιας οπτικής και κάτοψης του φακού Fresnel NL-11NH.



Εικόνα 7. 25: Περιοχές και εύρος κάλυψης σε διάφορα σημεία προσάρτησης.

#### **7.41. Επιπλέον πληροφορίες**

Πυροηλεκτρικές συσκευές, όπως ο αισθητήρας PIR, περιέχουν στοιχεία που κατασκευάζονται από κρυσταλλικό υλικό που παράγει ηλεκτρικό φορτίο όταν εκτίθεται σε υπέρυθρη ακτινοβολία. Οι αλλαγές στο ποσό της υπέρυθρης που προσπίπτουν στο στοιχείο αλλάζουν την τάση που δημιουργείται και που μετρώνται από έναν ενισχυτή που υπάρχει στον αισθητήρα PIR. Η συσκευή περιέχει ένα ειδικό φίλτρο που ονομάζεται φακός Fresnel, ο οποίος επικεντρώνει τα υπέρυθρα σήματα πάνω στο στοιχείο. Καθώς τα υπέρυθρα σήματα του περιβάλλοντος αλλάζουν γρήγορα, ο ενισχυτής που υπάρχει στην πλακέτα τροφοδοτεί την έξοδο για να μας δείξει την αντίχρευση κίνησης.

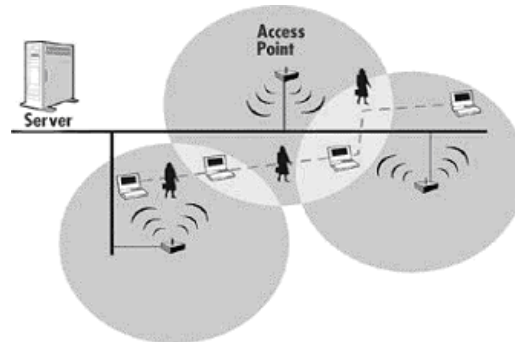
#### **7.42. Βαθμονόμηση**

Ο αισθητήρας PIR απαιτεί «χρόνο προθέρμανσης», προκειμένου να λειτουργήσει σωστά. Αυτό οφείλεται στο χρόνο που απαιτείται για την «μάθηση», του περιβάλλοντος του. Αυτό μπορεί να διαρκέσει από 10 έως 60 δευτερόλεπτα. Σε αυτό το διάστημα θα πρέπει να υπάρχει όσο το δυνατόν λιγότερη κίνηση στον οπτικό πεδίο αισθητήρα.



## 8. Εισαγωγή στα ασύρματα τοπικά δίκτυα

Ένα ασύρματο τοπικό δίκτυο (WLAN: Wireless Local Area Network) είναι ένα σύστημα επικοινωνίας, το οποίο καθιστά δυνατή την διασύνδεση κινητών ή ακίνητων χρηστών (και την μεταφορά δεδομένων) μέσω ηλεκτρομαγνητικών κυμάτων. Η συνηθισμένη ακτίνα δράσης ενός τέτοιου δικτύου μπορεί να εκτείνεται σε αρκετά μέτρα, είναι επίσης ικανή να συνδέσει τους ορόφους μιας εταιρίας μέχρι και τα κτίρια μιας πανεπιστημιούπολης.



Εικόνα 8. 1: Ένα τυπικό WLAN.

Αποτελεί ένα ασύρματο τρόπο διασύνδεσης, ενώ δίνει την δυνατότητα σύνδεσης και με το Internet. Οι ασύρματες τεχνολογίες πρόσβασης χρησιμοποιούνται για να αντικαταστήσουν ή να επεκτείνουν ένα κοινό ενσύρματο δίκτυο (Ethernet) και επιτρέπουν στον κινητό χρήστη την ασύρματη μετάδοση και λήψη δεδομένων.

### 8.1. Το πρότυπο IEEE 802.11

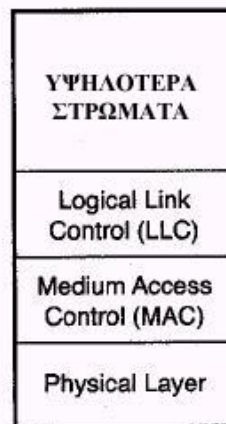
IEEE 802.11 είναι ένα σύνολο προτύπων που υλοποιεί επικοινωνία υπολογιστών μέσω ασύρματων τοπικών δικτύων (WLAN) στις περιοχές των 2,4, 3,6 και 5 Ghz. Έχουν δημιουργηθεί και συντηρούνται από την επιτροπή προτύπων IEEE LAN / MAN (IEEE 802). Στη βασική έκδοση του προτύπου IEEE 802.11-2007 υπάρχουν και μεταγενέστερες τροποποιήσεις. Τα πρότυπα αυτά παρέχουν τη βάση για την ανάπτυξη προϊόντων ασύρματης δικτύωσης, χρησιμοποιώντας την Wi-Fi εμπορική επωνυμία.

### 8.2. Οι βασικές του μονάδες του 802.11

Τα ασύρματα δίκτυα 802.11 αποτελούνται από τις παρακάτω τέσσερις βασικές μονάδες:

- **Σημείο πρόσβασης (Access Point - AP):** Το AP είναι η μονάδα που παίζει το ρόλο γέφυρας μεταξύ του ενσύρματου και του ασύρματου δικτύου, μετατρέποντας κατάλληλα τα πλαίσια που ανταλλάσσονται μεταξύ αυτών.
- **Σύστημα διανομής (Distribution System):** Το σύστημα διανομής ενώνει τα διάφορα AP του ίδιου δικτύου, επιτρέποντάς τους να ανταλλάσσουν πλαίσια. Το 802.11 δεν προσδιορίζει τον τρόπο που θα γίνεται αυτό.
- **Ασύρματο μέσο μετάδοσης (Wireless Medium):** Έχουν οριστεί διάφορα φυσικά στρώματα που χρησιμοποιούν, είτε ραδιοσυχνότητες, είτε υπέρυθρες ακτίνες για τη μετάδοση των πλαισίων μεταξύ των σταθμών του ασύρματου δικτύου.
- **Σταθμοί (Stations):** Οι σταθμοί που ανταλλάσσουν πληροφορία μέσω του ασυρμάτου δικτύου συνήθως είναι φορητές συσκευές (για παράδειγμα laptops ή PDAs) χωρίς όμως αυτό να είναι απαραίτητο.

«Το πρότυπο 802.11 μέσω στρώμας πρωτοκόλλων παρέχει τη λειτουργία MAC και PHY για την ασύρματη σύνδεση σταθερών, φορητών, και κινούμενων σταθμών σε μια δεδομένη περιοχή.»



Εικόνα 8. 2: Στοιβά πρωτοκόλλου IEEE 802.11.

### 8.3. Η τοπολογία του 802.11

Η βασική δομική μονάδα κάθε 802.11 δικτύου αποκαλείται Basic Service Set (BSS) και αποτελείται από μία ομάδα σταθμών που επικοινωνούν μεταξύ τους. Τα όρια του BSS καθορίζονται από την περιοχή ραδιοκάλυψης, που ονομάζεται Basic Service Area (BSA). Ένας σταθμός σε ένα BSS μπορεί να επικοινωνεί με οποιονδήποτε άλλο σταθμό στο ίδιο BSS.

Η τοπολογία του 802.11 αποτελείται από στοιχεία που αλληλεπιδρούν ώστε να παρέχουν ένα ασύρματο τοπικό δίκτυο, που να παρέχει τη δυνατότητα μετακίνησης των σταθμών, η οποία να μην γίνεται αντιληπτή από τα ανώτερα στρώματα. Ένας σταθμός (station), είναι κάθε συσκευή η οποία εμπεριέχει τις λειτουργίες του 802.11, δηλαδή το επίπεδο MAC, το φυσικό στρώμα και μια διασύνδεση (interface) με το ασύρματο μέσο.

- **BSS:** Το βασικό δομικό στοιχείο ενός IEEE 802.11 LAN είναι το BSS (Basic Service Set βασικό σύνολο υπηρεσιών). Το (BSS) είναι το σύνολο όλων των σταθμών που μπορούν να επικοινωνούν μεταξύ τους.
- **IBSS:** Ο πιο βασικός τύπος ενός 802.11 LAN είναι το IBSS (Independent BSS), όπου δύο ή περισσότεροι σταθμοί μπορούν να επικοινωνούν απευθείας μεταξύ τους.
- **ESS:** Όταν οι ανάγκες της διαδίκτυωσης ξεπερνούν τα όρια του IBSS, το 802.11 καθορίζει τη δομή ενός πιο σύνθετου τοπικού δικτύου που ονομάζεται ESS (Extended Service Set) και στο οποίο είναι δυνατή η διασύνδεση και η επικοινωνία πολλών BSS μεταξύ τους. Το στοιχείο που χρησιμοποιείται για την διασύνδεση των BSS ονομάζεται DS (Distributed System). Η πρόσβαση στο DS γίνεται με την βοήθεια ενός σταθμού που καλείται AP (Access Point) και ο οποίος παρέχει ουσιαστικά τη διασύνδεση των σταθμών που βρίσκονται σε διάφορα BSS στο DS.

### 8.4. Τύποι κίνησης σταθμών

Το 802.11 αναγνωρίζει τους παρακάτω τύπους κίνησης σταθμών:

- **Απουσία μετακίνησης:** Ο τύπος αυτός αναφέρεται σε σταθμούς που δεν μετακινούνται και σε αυτούς που μετακινούνται μέσα σε ένα τοπικό BSS.

- **BSS μετακίνηση:** Ο τύπος αυτός αναφέρεται σε σταθμούς που μετακινούνται από ένα BSS σε ένα άλλο BSS μέσα στο ίδιο ESS.
- **ESS μετακίνηση:** Αυτός ο τύπος μετακίνησης αναφέρεται σε σταθμούς που μετακινούνται από ένα BSS σε ένα άλλο BSS το οποίο ανήκει σε διαφορετικό ESS.

Αξίζει να αναφέρουμε ότι το 802.11 ενώ υποστηρίζει ξεκάθαρα τους δύο πρώτους τύπους μετακίνησης, δεν εγγυάται την διατήρηση της σύνδεσης κατά την μετακίνηση σε διαφορετικό ESS.

## 8.5. Μηχανισμοί λειτουργίας

Το 802.11 υποστηρίζει δύο τρόπους λειτουργίας:

- **Ομότιμα ή Ad Hoc:** όπου δεν υπάρχει κάποιος κεντρικός σταθμός βάσης ή σημείο πρόσβασης, οι κόμβοι είναι ισότιμοι και η πρόσβαση στο κοινό μέσο (τον κενό χώρο) ρυθμίζεται από κάποιο καταναμημένο πρωτόκολλο όπως το CSMA/CA.
- **Με σημείο πρόσβασης:** έναν κεντρικό κόμβο τοπικού δικτύου δηλαδή, συνήθως συνδεδεμένο σε ενσύρματο δίκτυο κορμού (π.χ. στο Internet ή σε κάποιο μεγάλο Ethernet LAN).

Και στις δύο τεχνικές υποστηρίζονται ρυθμοί μετάδοσης 1 και 11Mbps στην ζώνη συχνοτήτων 2.4 - 2.4835GHz σε μικρές αποστάσεις. Ένα άλλο ενδιαφέρον σημείο στην περίπτωση των ασύρματων δικτύων είναι η συμβατότητα των διαφόρων συσκευών.

## 8.6. Η αρχιτεκτονική του 802.11

Ενώ η τοπολογία καθορίζει τα αναγκαία μέσα για τη φυσική διασύνδεση του ασύρματου δικτύου, η αρχιτεκτονική καθορίζει τον τρόπο λειτουργίας του δικτύου. Έτσι, η αρχιτεκτονική του 802.11 η οποία εφαρμόζεται σε κάθε σταθμό, αποτελείται από ένα υπόστρωμα **MAC** και το στρώμα **PHY** (φυσικό).

## 8.7. Το υπόστρωμα MAC (*Medium Access Control Sublayer*) του 802.11

Σκοπός του υποστρώματος MAC είναι να παρέχει λειτουργίες ελέγχου πρόσβασης (στις οποίες συμπεριλαμβάνονται η διευθυνσιοδότηση, ο έλεγχος της σωστής σειράς των πλαισίων κ.ά.) στο μοιραζόμενο φυσικό κανάλι, όπως αυτό καθορίζεται από το πρότυπο.

Το υπόστρωμα MAC του IEEE 802.11 λειτουργεί ως διεπαφή μεταξύ του φυσικού επιπέδου και της ασύρματης κάρτας δικτύου. Υποστηρίζει τόσο δομημένη όσο και λειτουργία Ad Hoc. Δύο επιπλέον χαρακτηριστικά αξιοπιστίας στο IEEE 802.11 MAC υποεπίπεδο είναι ο κυκλικός έλεγχος πλεονασμού (CRC - Cyclic Redundancy Check) και ο κατακερματισμός πακέτων. Κάθε πακέτο έχει το δικό του CRC υπολογισμένο και αυτό επισυνάπτεται, ώστε να εξασφαλίσει ότι τα δεδομένα δεν έχουν καταστραφεί κατά τη μεταφορά.

Ο κατακερματισμός πακέτων θα στείλει μεγάλα πακέτα σε μικρά κομμάτια όταν αποστέλλονται ασύρματα. Αυτό έχει δύο πλεονεκτήματα. Το πρώτο είναι η μείωση της ανάγκης για αναμετάδοση, διότι η πιθανότητα να αλλοιωθεί ένα πακέτο αυξάνεται με το μέγεθος του πακέτου. Το δεύτερο είναι ότι στην περίπτωση αλλοίωσης πακέτου, ο κόμβος πρέπει να αναμεταδώσει μόνο ένα μικρό κομμάτι, επομένως αυτό είναι πιο γρήγορο.

## 8.8. Πρόσβαση στο ασύρματο μέσο μέσω της λειτουργίας CSMA/CA

Το CSMA/CA (Carrier Sense Multiple Access with Collision Avoidance). Η λειτουργία Carrier Sense πραγματοποιείται με παρακολούθηση του καναλιού πριν την έναρξη εκπομπής. Αν

κάποιος άλλος client εκείνη την ώρα τύχει να εκπέμπει, τότε ο πρώτος περιμένει, έως ότου να βρεθεί στιγμή που το κανάλι να είναι ελεύθερο. Για να επιτευχθεί ένα καλό ποσοστό συγχρονισμού, που θα εξασφαλίσει την εύρυθμη λειτουργία του δικτύου, πρέπει οι περισσότεροι σταθμοί να βρίσκονται σε θέση να «ακούν» τις εκπομπές όλων των άλλων.

Όταν δηλαδή ένας σταθμός ελέγχει το μέσο για να δει αν είναι σε χρήση, μπορεί εσφαλμένα να αποφασίσει ότι είναι ελεύθερο, μιας και δεν είναι σε θέση να λαμβάνει τις εκπομπές όλων των άλλων σταθμών του Access Point. Σε αυτήν την περίπτωση, το αποτέλεσμα θα είναι συνεχείς συγκρούσεις. Συνήθως το Access Point τείνει να ευνοεί τον εκπομπό με το καλύτερο σήμα, καθώς λαμβάνει το σήμα του ασθενέστερου σαν θόρυβο και το απορρίπτει. Δεδομένων λοιπόν των συνθηκών, μια και μόνο συσκευή μπορεί να μονοπωλήσει ολόκληρο το εύρος ζώνης του AP.

## 8.9. Χρονικά διαστήματα πρόσβασης

Το IEEE 802.11 καθορίζει την ύπαρξη χρονικών διαστημάτων για την μεσολάβηση μεταξύ των διαφόρων λειτουργιών αποστολής και λήψης πλαισίων ενός σταθμού. Το χρονικό διάστημα μεταξύ των πλαισίων (frames) καλείται **IFS** (Inter Frame Space).

## 8.10. Το φυσικό επίπεδο του 802.11

Το 802.11 καθορίζει διάφορα φυσικά στρώματα, καθώς γίνονται διαθέσιμες νέες τεχνολογίες. Έτσι, ενώ το αρχικό Standard υποστήριζε ρυθμούς μετάδοσης έως 2 Mbps, τα τρέχοντα Standard της οικογένειας 802.11 καθορίζουν φυσικά στρώματα με ρυθμούς μετάδοσης μέχρι και 54 Mbps, χρησιμοποιώντας κατάλληλες τεχνικές διαμόρφωσης.

Το IEEE 802.11 φυσικό στρώμα (Physical Layer) είναι χωρισμένο σε δύο μέρη, το φυσικό στρώμα πρωτοκόλλου σύγκλισης (PLCP - Physical Medium Dependent) και το φυσικό μέσον εξαρτήσεως (PMD - Physical Medium Dependent). Το PMD φροντίζει για την ασύρματη κωδικοποίηση. Το PLCP παρουσιάζει μια κοινή διεπαφή για το MAC υπόστρωμα ώστε να επεμβαίνει σε αυτό και να του παρέχει ανίχνευση φέροντος όπως και CCA (Clear Channel Assessment Σαφής Εκτίμηση Καναλιού).

PLCP έχει δύο δομές, ένα μακρύ και ένα μικρό προοίμιο. Όλα τα συμβατά με το 802.11 συστήματα πρέπει να υποστηρίζουν το μακρύ προοίμιο. Το μικρό προοίμιο παρέχεται στο πρωτόκολλο για να βελτιωθεί η αποτελεσματικότητα της απόδοσης ενός δικτύου κατά τη μετάδοση ειδικών δεδομένων, όπως φωνής, VoIP (Voice-over IP) και streaming video.

## 8.11. Λειτουργίες του φυσικού στρώματος

Για την εκτέλεση των λειτουργιών του υποστρώματος PLCP, το 802.11 καθορίζει την χρήση των μηχανών κατάστασης (state machines). Κάθε μηχανή κατάστασης εκτελεί μία από τις παρακάτω λειτουργίες:

- **Ανίχνευση φέροντος:** Η λειτουργία αυτή αφορά τον καθορισμό της κατάστασης του μέσου δηλαδή αν το μέσο είναι απασχολημένο ή αν έχουμε εισερχόμενα σήματα (δεδομένα).
- **Μετάδοση:** Η λειτουργία αυτή αναφέρεται στην αποστολή των διαδοχικών bytes ενός πλαισίου δεδομένων.
- **Λήψη:** Η λειτουργία αυτή αναφέρεται στην λήψη διαδοχικών bytes ενός πλαισίου δεδομένων.

Το φυσικό στρώμα (PHY) καλύπτει τη φυσική διεπαφή μεταξύ των συσκευών και ασχολείται με τη μετάδοση ακατέργαστων (raw) bits μέσω του καναλιού επικοινωνίας. Το φυσικό επίπεδο του προτύπου IEEE 802.11 είναι μια επέκταση του IEEE 802.11 φυσικού επιπέδου το οποίο υποστηρίζει 1 και 2 Mbps. Το IEEE 802.11 μπορεί να υποστηρίξει υψηλότερες ταχύτητες μετάδοσης δεδομένων της τάξης του 5,5 και 11 Mbps με τη χρήση Complementary Code Keying (CCK) με Quadrature Phase Shift Keying (QPSK) διαμόρφωση και τεχνολογία Direct-Sequence Spread-Spectrum (DSSS).

## 8.12. Υπηρεσίες που παρέχει το πρωτόκολλο 802.11

Το 802.11 καθορίζει τις υπηρεσίες που παρέχουν τις απαιτούμενες λειτουργίες για την αποστολή των δεδομένων ανάμεσα σε δύο ομότιμα στρώματα LLC<sup>27</sup>. Αυτές οι υπηρεσίες, που υλοποιεί το στρώμα MAC, χωρίζονται σε δύο κατηγορίες:

- **Station Services:** Authentication, Deauthentication και Privacy.
- **Distribution System Services:** Association, Disassociation, Distribution, Integration και Reassociation.

### 8.12.1. Station Services

Το 802.11 καθορίζει υπηρεσίες για την παροχή λειτουργιών μεταξύ των σταθμών:

- **Authentication:** κάθε σταθμός, είτε είναι μέρος ενός IBSS ή ενός ESS δικτύου, πρέπει να χρησιμοποιήσει την υπηρεσία της ‘επικύρωσης’ (authentication) πριν την εγκατάσταση μιας σύνδεσης (association).
- **Deauthentication:** όταν ένας σταθμός θέλει να αποσυνδεθεί (disassociate) από έναν άλλον σταθμό χρησιμοποιεί την υπηρεσία που καλείται ‘deauthentication’.
- **Privacy:** η υπηρεσία αυτή βασίζεται στον αλγόριθμο WEP. Ο αλγόριθμος αυτός κρυπτογραφεί τα μηνύματα (με την χρήση του αλγορίθμου κρυπτογράφησης RC4) που στέλνονται δια μέσου του ασύρματου δικτύου.

### 8.12.2. Distribution System Services

- **Association:** κάθε σταθμός πρέπει αρχικά να θέσει σε λειτουργία την υπηρεσία της σύνδεσης (association) με ένα AP πριν στείλει οποιαδήποτε πληροφορία.
- **Disassociation:** η υπηρεσία αυτή τερματίζει μια υπάρχουσα σύνδεση. Οι σταθμοί πρέπει να αποσυνδέονται όταν εγκαταλείπουν ένα δίκτυο και τα AP όταν χρειάζονται συντήρηση.
- **Distribution:** Ένας σταθμός χρησιμοποιεί την υπηρεσία αυτή κάθε φορά που θέλει να στείλει MAC πλαίσια δια μέσου του DS. Το 802.11 δεν καθορίζει τον τρόπο με τον οποίο το DS διανέμει τα δεδομένα. Η μόνη πληροφορία που δίνει η υπηρεσία στο DS είναι ο καθορισμός του BSS για το οποίο προορίζεται το πλαίσιο.
- **Integration:** η υπηρεσία της ενοποίησης (integration) κάνει εφικτή την διανομή των MAC πλαισίων μέσω μιας πύλης (portal) μεταξύ ενός DS και ενός LAN που δεν ανήκει στην οικογένεια 802.11.
- **Reassociation:** η υπηρεσία αυτή της επανασύνδεσης (reassociation) καθιστά ικανό ένα σταθμό να αλλάζει την τρέχουσα κατάσταση σύνδεσης από ένα AP σε ένα άλλο. Με τον τρόπο αυτό υποστηρίζεται η μετάβαση μεταξύ διαφορετικών BSS.

Το 802.11 υποστηρίζει, μεταξύ των άλλων υπηρεσιών, την περιαγωγή (roaming) ενός σταθμού μεταξύ πολλών APs, τα οποία χρησιμοποιούν το ίδιο ή διαφορετικό κανάλι. Για την υποστήριξη της λειτουργίας αυτής, κάθε AP μεταδίδει σε συγκεκριμένα χρονικά διαστήματα (συνήθως κάθε 100 ms) ένα σήμα (που καλείται beacon signal) και το οποίο ενημερώνει τον κάθε σταθμό για την τρέχουσα ισχύ της σύνδεσής του με το ανάλογο AP.

---

<sup>27</sup> παρέχει μηχανισμούς πολυπλεξίας που καθιστούν δυνατή τη δυνατότητα σε διαφορετικά δικτυακά πρωτόκολλα να συνυπάρχουν.



## 8.13. Το πρωτόκολλο IEEE 802.11b

Το 802.11b αποτελεί συμπλήρωμα στο πρότυπο IEEE 802.11 και έχει την εμπορική ονομασία Wi-Fi. Το Wi-Fi προέρχεται από τα αρχικά των «Wireless Fidelity» (Ψηφιακή Πιστότητα). Η οικογένεια αυτή περιλαμβάνει και το **802.11b** που χρησιμοποιεί και η Wishield.

## 8.14. Περιγραφή

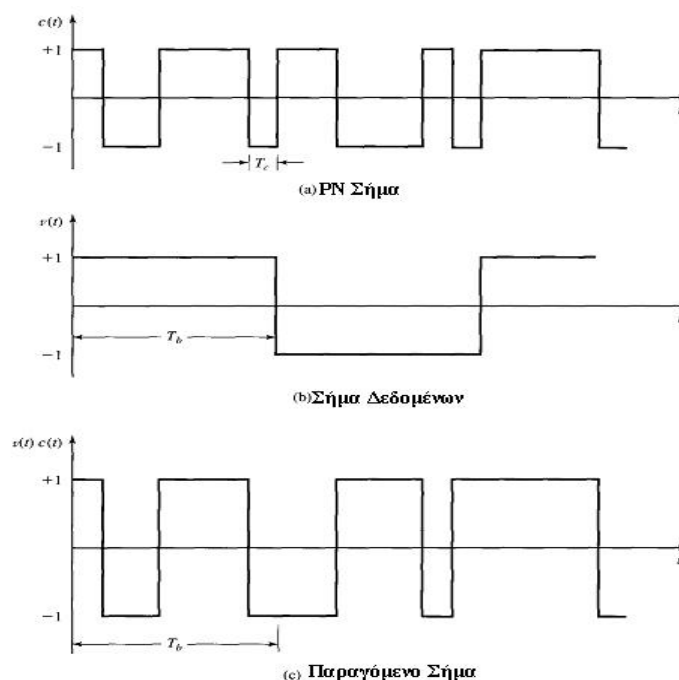
Το 802.11b έχει μέγιστο ρυθμό δεδομένων 11 Mbit/s και χρησιμοποιεί την CSMA/CA μέθοδο πρόσβασης μέσου που ορίζεται στο αρχικό πρότυπο (IEEE 802.11). Λόγω των επιπλέον δεδομένων έλεγχου που απαιτεί η CSMA/CA (protocol overhead), στην πράξη, το 802.11b έχει μέγιστη απόδοση περίπου **5,9 Mbit/s** όταν χρησιμοποιούμε **TCP** και **7,1 Mbit/s**, με **UDP**.

Το 802.11b είναι μια άμεση επέκταση της τεχνικής διαμόρφωσης DSSS (Direct Sequence Spread Spectrum) που ορίζεται στο αρχικό πρότυπο. Τα WLAN λειτουργούν στη μη αδειοδοτημένη περιοχή συχνοτήτων **ISM** (Industrial, Scientific and Medical) στα 2,4 - 2.4835 Ghz. **Στην Ελλάδα χρησιμοποιείται η ζώνη των 2.4 GHz.**

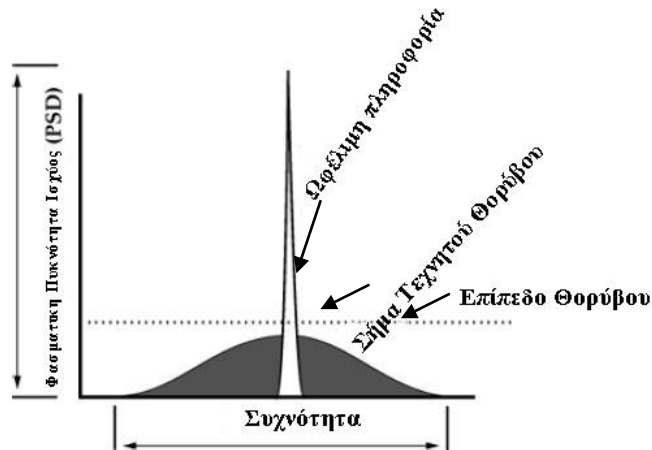
Οι 802.11b συσκευές υφίστανται παρεμβολές από άλλα προϊόντα που λειτουργούν στη ζώνη 2.4 GHz. Συσκευές που λειτουργούν στο εύρος συχνότητας 2,4 GHz περιλαμβάνουν: φούρνους μικροκυμάτων, συσκευές Bluetooth, σύστημα ενδοεπικοινωνίας και ασύρματα τηλέφωνα.

## 8.15. Direct Sequence Spread Spectrum (DSSS)

Στην τεχνική DSSS το μεταδιδόμενο σήμα κωδικοποιείται κατά φάση με **ψευδό-τυχαία τμήματα «τεχνητού θορύβου»** τα οποία ονομάζονται **chips**. Ουσιαστικά κάθε μεταδιδόμενο bit δεδομένων πολλαπλασιάζεται με αυτό το σήμα του τεχνητού θορύβου το οποίο είναι εναλλαγή των τιμών -1 και 1 σε συχνότητα πολύ μεγαλύτερη του αρχικού σήματος. Με αυτό τον τρόπο η ενέργεια της μετάδοσης εξαπλώνεται σε μεγαλύτερο εύρος ζώνης. Η ακολουθία των σημάτων του ψευδό-θορύβου είναι εκ των προτέρων γνωστή τόσο στον πομπό όσο και στον δέκτη. Ο δέκτης επειδή γνωρίζει την ακολουθία του τεχνητού θορύβου μπορεί να αποκωδικοποιήσει το σήμα και να ανακτήσει την αρχική πληροφορία.



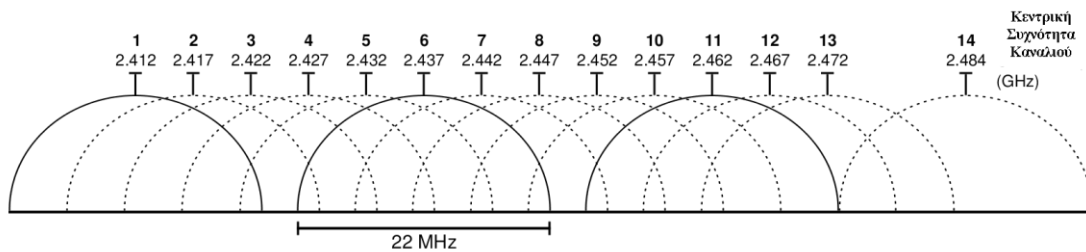
**Εικόνα 8. 3:** Τα δεδομένα (data signal) κωδικοποιούνται με το σήμα «τεχνητού θορύβου» (PN-code) και παράγεται το τελικό σήμα που εκπέμπεται (coded signal).



Εικόνα 8. 4: Η ωφέλιμη πληροφορία (Narrowband Waveform) πλαισιώνεται με το σήμα «τεχνητού θορύβου» (Spread Waveform) και εξαπλώνεται στο φάσμα συχνοτήτων.

### 8.16. Κανάλια και συχνότητες εκπομπής

Το 802.11b πρωτόκολλο χρησιμοποιεί **14 κανάλια εύρους 22MHz τα οποία επικαλύπτονται** μεταξύ τους. Αυτό δίνει την δυνατότητα σε πολλούς σταθμούς να λειτουργούν ταυτόχρονα **χωρίς παρεμβολές**.



Εικόνα 8. 5: Διάγραμμα επικάλυψης καναλιών.

A/A Καναλιού	Συχνότητα σε Ghz	Delta Συχνότητας σε MHz	Εύρος Καναλιού σε GHz	Επικαλύπτει το κανάλι με A/A
1	2.412 GHz	5 MHz	2.401-2.423 GHz	2
2	2.417 GHz	5 MHz	2.406-2.428 GHz	1,3
3	2.422 GHz	5 MHz	2.411-2.433 GHz	2,4
4	2.427 GHz	5 MHz	2.416-2.438 GHz	3,5
5	2.432 GHz	5 MHz	2.421-2.443 GHz	4,6
6	2.437 GHz	5 MHz	2.426-2.448 GHz	5,7
7	2.442 GHz	5 MHz	2.431-2.453 GHz	6,8
8	2.447 GHz	5 MHz	2.436-2.458 GHz	7,9
9	2.452 GHz	5 MHz	2.441-2.463 GHz	8,10
10	2.457 GHz	5 MHz	2.446-2.468 GHz	9,11
11	2.462 GHz	5 MHz	2.451-2.473 GHz	10,12
12	2.467 GHz	5 MHz	2.456-2.478 GHz	11,13
13	2.472 GHz	5 MHz	2.461-2.483 GHz	12

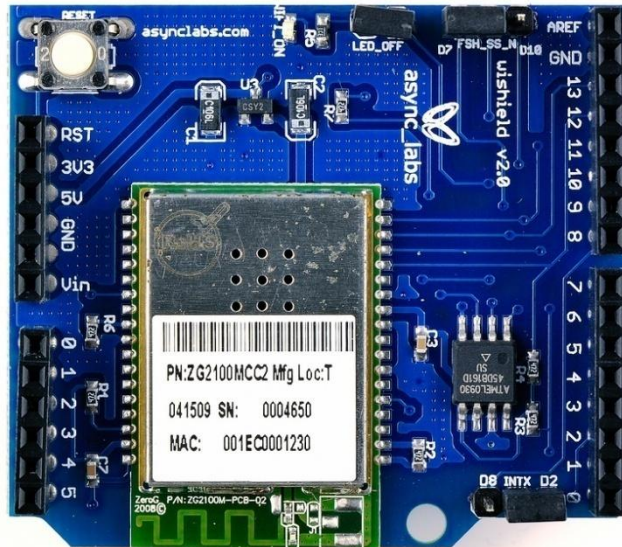
Πίνακας 10: Οι συχνότητες των καναλιών του 802.11b.

## 8.17. Ανοχή στον θόρυβο

Επιπλέον, IEEE 802.11b καθορίζει έναν **δυναμικό ρυθμό διαμεταγωγής**, επιτρέποντας στους ρυθμούς δεδομένων να προσαρμόζονται αυτόματα **σε θορυβώδεις συνθήκες**. Αυτό σημαίνει ότι οι συσκευές IEEE 802.11b θα μεταδώσουν σε χαμηλότερες ταχύτητες, 5.5 Mbps, 2 Mbps, και 1 Mbps κάτω από συνθήκες θορύβου. Όταν οι συσκευές κινηθούν πίσω, εντός περιοχών με υψηλότερη ταχύτητα μετάδοσης, η σύνδεση αυτόματα θα επιταχυνθεί και πάλι.

## 8.18. Παρουσίαση της WiShield

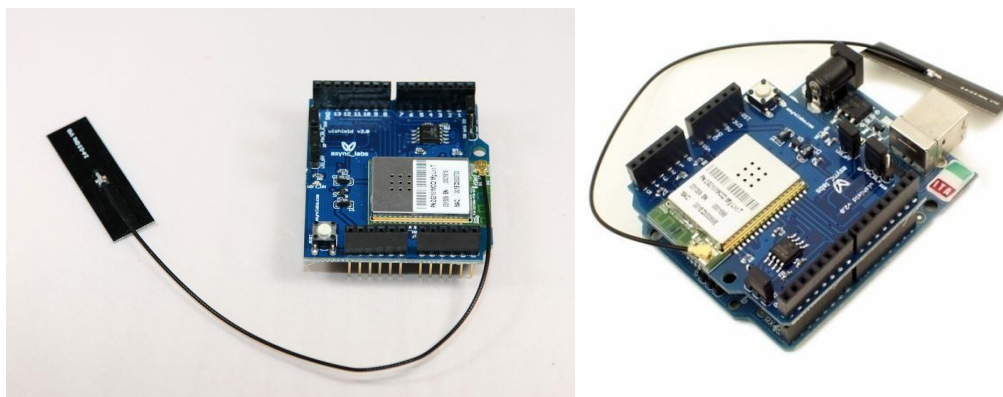
Η Wisshield μας παρέχει ασύρματο πρωτόκολλο 802.11b όπως επίσης και την δυνατότητα να μας διακομίζει (serve) άπλες ιστοσελίδες που μπορούμε να διαμορφώσουμε το περιεχόμενό τους.



Εικόνα 8. 6: Κάτοψη της WiShield.

## 8.19. Χαρακτηριστικά:

- Επιπρόσθετη πλακέτα για το Arduino με απολυτή συμβατότητα διαστάσεων
- 9 pin επιπρόσθετα για κάθε επιπλέον χρήση
- SPI πρωτόκολλο επικοινωνίας (μεγίστης συχνότητας 25MHz)
- Όλα τα pin του Arduino είναι προσβάσιμα μέσω αυτής
- Εύκολη πρόσβαση στο κουμπί reset
- Ενσωματωμένη κεραία
- Επιλογή είδους interrupt μέσω pin, INT0 (port D, pin 2) και pin 8 port B, pin 0)
- LED κατάστασης σύνδεσης WiFi στο pin 9 (port B, pin 1)
- Μονάδα DataFlash AT45DB161D με χωρητικότητα 16 Mbit (2MB)



Εικόνα 8. 7: Η κεραία της WiShield και η προσάρτηση πάνω στο Arduino.

## 8.20. Χαρακτηριστικά μονάδας Wi-Fi

- Πιστοποιημένη 802.11b Wi-Fi συσκευή με ταχύτητες διεκπεραιωτικότητας:
  - 1Mbps και 2 Mbps
- Υποστήριξη των προτύπων υποδομής (BSS) και ad hoc (IBSS) ασύρματης δικτύωσης
- Ασφάλεια ασύρματων δικτύων μέσω προτύπων:
  - WEP (64-bit και 128-bit)
  - WPA/WPA2 (TKIP και AES) PSK
- Χαμηλή κατανάλωση ενέργειας :
  - Αναμονή 10mA (Standby)
  - Μετάδοση 230mA (Transmit)
  - Λήψη 85mA (Receive)
  - Αναμονή χαμηλής κατανάλωσης 250μA (Sleep)

## 8.21. Πως χρησιμοποιούμε το κάθε pin

### ➤ Μονάδα WiFi μέσω διαύλου SPI

- Λειτουργία Slave Select (SS) : στο pin 10 του Arduino (port B, pin 2), καθορίζει την slave συσκευή
- Λειτουργία Clock (SCK) : στο pin 13 του Arduino (port B, pin 5), προσθέτει επιπλέον λειτουργίες χρονισμού
- Λειτουργία Master In, Slave Out (MISO) : στο pin 12 του Arduino (port B, pin 4), η master συσκευή δέχεται είσοδο ενώ η slave παράγει την έξοδο
- Λειτουργία Master Out, Slave In (MOSI) : στο pin 11 του Arduino (port B, pin 3), η master συσκευή παράγει την έξοδο ενώ η slave δέχεται είσοδο

### ➤ Επιλογή είδους interrupt (μονό μέσω επιλογέα)

- INT0 : στο pin 2 του Arduino (port D, pin 2)
- DIG8 : στο pin 8 του Arduino (port B, pin 0)

### ➤ Interrupt επιλογέας για το WiFi

- Επιτρέπει την επιλογή μεταξύ των τύπων INT0 or DIG8

### ➤ LED κατάστασης WiFi σύνδεσης: στο pin 9 του Arduino (port B, pin 1)

- Μόλις η WiShield συνδεθεί με ένα σημείο πρόσβασης (AP) η οποιαδήποτε WiFi συσκευή μέσω ενός AdHoc δικτύου, αυτό το LED ανάβει

### ➤ Επιλογέας (jumper) LED

- Η ανάκτηση του pin 9 για χρήση, με αφαίρεση του επιλογέα LED

### ➤ DataFlash λειτουργία μέσω διαύλου SPI

- Λειτουργία Slave Select (SS) : στο pin 7 του Arduino (port D, pin 7), καθορίζει την slave συσκευή
- Λειτουργία Clock (SCK) : στο pin 13 του Arduino (port B, pin 5), προσθέτει επιπλέον λειτουργίες χρονισμού
- Λειτουργία Master In, Slave Out (MISO) : στο pin 12 του Arduino (port B, pin 4), η master συσκευή δέχεται είσοδο ενώ η slave παράγει την έξοδο

- Λειτουργία Master Out, Slave In (MOSI) : στο pin 11 του Arduino (port B, pin 3), η master συσκευή παράγει την έξοδο ενώ η slave δέχεται είσοδο
- **Κουμπί Reset**
  - Το ίδιο κουμπί χρησιμοποιεί το Arduino (ίδια θέση) για να προκαλέσει Reset. Προκαλεί reset στην Wishield και στο Arduino



## 9. Υλοποίηση του τηλεμετρικού υποσυστήματος

Έχοντας κατανοήσει θέματα όπως τηλεμετρία, μικροελεγκτές, αισθητήρες και ασύρματα πρωτοκολλά επικοινωνίας, είμαστε σε θέση να προχωρήσουμε στην υλοποίηση του τηλεμετρικού υποσυστήματος. Θα μελετήσουμε τις απαιτήσεις του και βάσει αυτών, θα παραμετροποιήσουμε τους αισθητήρες και θα υλοποιήσουμε εφαρμογή σε γλώσσα προγραμματισμού C++, που θα συνδυάζει όλα τα μέρη του υποσυστήματος με απώτερο στόχο την ικανοποίηση των απαιτήσεων.

**Σημείωση:** Σε καμία περίπτωση αντικείμενο αυτής της εργασίας δεν αποτελούν θέματα όπως προγραμματισμός σε C++, AVR C, Html όπως και η σύνθεση css style sheets. Επίσης, δεν αποτελούν θέμα αυτής της εργασίας τεχνικά θέματα όπως εγκατάσταση λογισμικού και ρύθμιση υλικού σε οποιοδήποτε λειτουργικό σύστημα.

### 9.1. Απαιτήσεις του τηλεμετρικού υποσυστήματος

Σ' αυτή την ενότητα θα αναλύσουμε τις απαιτήσεις του τηλεμετρικού υποσυστήματος και θα τις χωρίσουμε σε υποενότητες με βάση το αντικείμενο που πραγματεύονται και θα τις χρησιμοποιήσουμε ως οδηγό στην μετέπειτα ανάπτυξη. Οι απαιτήσεις κατηγοριοποιούνται βάση των επιμέρους κομματιών του υποσυστήματος. Αυτά περιλαμβάνουν τους αισθητήρες, το μικροελεγκτή και τα ασύρματα πρωτόκολλο που χρησιμοποιεί.

### 9.2. Απαιτήσεις των αισθητήρων

Στο κεφάλαιο 7 παρουσιάστηκαν οι επιμέρους αισθητήρες που θα χρησιμοποιηθούν και συντέθηκε μια ποσότητα πληροφοριών που μας βοηθήσει να κατανοήσουμε τους αισθητήρες και τις απαιτήσεις τους, ώστε να λειτουργήσουν μέσα στα πλαίσια που ορίζει ο κατασκευαστής τους.

Συγκεκριμένα ο κάθε αισθητήρας, απαιτεί συγκεκριμένες συνθήκες περιβάλλοντος για να λειτουργήσει στα πλαίσια προδιαγραφών του κατασκευαστή. Ακόμη, οι αισθητήρες έχουν συγκεκριμένο εύρος τιμών εξόδου (τιμές μετρήσεων), γεγονός που πρέπει να του δώσουμε επίσης ιδιαίτερη έμφαση. Τέλος, πρέπει να δώσουμε έμφαση σε τυχόν ρυθμίσεις που απαιτούνται, ώστε οι μετρήσεις (των αισθητήρων) να είναι αξιόπιστες.

#### Για τον αισθητήρα θερμοκρασίας Dallas DS18B20:

- Οι μετρήσεις μας να είναι στο εύρος -55 έως 125 ° C.
- Τα διαστήματα χρόνου που απαιτούμε μετρήσεις να μην μικρότερα από 93,75ms.
- Απαιτείται μια αντίσταση pull-up τουλάχιστον 4.7kΩ.
- Να παρέχουμε τάση τροφοδοσίας εντός του εύρους 3 με 5.5 Volt .

#### Για τον αισθητήρα υγρασίας Honeywell HHH-4030:

- Οι μετρήσεις μας να είναι στο εύρος 0 έως 100 %RH.
- Τα διαστήματα χρόνου που απαιτούμε μετρήσεις να μην μικρότερα από 5 sec.
- Θερμοκρασία περιβάλλοντος λειτουργίας από -40 έως 90 ° C.
- Να παρέχουμε τάση τροφοδοσίας 5 Volt .

#### Για τον αισθητήρα PIR Parallax 555 - 280227:

- Η περιοχή ανίχνευσης να μην είναι μεγαλύτερη από έναν κύκλο διαμέτρου 6 μέτρων.
- Χρονικό διάστημα ενός λεπτού το οποίο ο αισθητήρας απαιτεί για την αυτορρύθμιση του.
- Θερμοκρασία περιβάλλοντος λειτουργίας από 0 έως 50 ° C.
- Να παρέχουμε τάση τροφοδοσίας 3.3 έως 5 Volt.



### 9.3. Απαιτήσεις του μικροελεγκτή

Αναφέραμε στο κεφάλαιο 3 τι συνθέτει έναν μικροελεγκτή και ως εκ τούτου έγιναν κατανοητές οι ιδιότητες του. Προκύπτει λοιπόν, ότι πρέπει να τον χειριστούμε ως μια μονάδα προγραμματίσιμη, με περιορισμούς στις δυνατότητες της και συγκεκριμένες απαιτήσεις όπως, π.χ. επιπλέον απαιτούμενο υλικό για την χρήση του.

Από την στιγμή που ο μικροελεγκτής είναι ενσωματωμένος στην αναπτυξιακή πλακέτα Arduino, οι περισσότερες λειτουργικές απαιτήσεις του καλύπτονται απ' αυτήν. Απομένει να αναλύσουμε τις απαιτήσεις της αναπτυξιακής πλακέτας Arduino και ν' αναφέρουμε επιγραμματικά τις απαιτήσεις του ATmega 328P.

#### Για τον ATmega 328P:

Ο κώδικας που θα αναπτύξουμε να μην απαιτεί περισσότερο από 32 KB χώρο μνήμης για την αποθήκευση του, ούτε περισσότερο από 2 KB για τους υπολογισμούς που θα εκτελέσει αλλά και να μην απαιτεί περισσότερο από 1 KB για μόνιμη αποθήκευση δεδομένων.

#### Για την αναπτυξιακή πλακέτα Arduino:

- Καλώδιο USB τύπου (A σε B) για τον προγραμματισμό.
- Τροφοδοσία 9 Volt και 500 mA ή μέσω θύρας USB.
- Η/Υ με λειτουργικό σύστημα που να υποστηρίζει το Arduino IDE.

### 9.4. Απαιτήσεις του ασύρματου πρωτόκολλου

Στο κεφάλαιο 8 αναλύθηκαν τα ασύρματα πρωτόκολλα επικοινωνίας 802.11 και 802.11b. Στην υλοποίησή μας το ασύρματο πρωτόκολλο 802.11b το παρέχει η αναπτυξιακή πλακέτα WiShield. Απομένει να ικανοποιήσουμε τις λειτουργικές απαιτήσεις της WiShield και να λάβουμε υπόψη τους περιορισμούς που μας επιβάλλει.

#### Για την αναπτυξιακή πλακέτα WiShield:

- Τάση τροφοδοσίας 5 Volt.
- Παροχή AP (Access Point) συμβατό με 802.11b.
- Το AP να είναι ρυθμισμένο σε infrastructure ή Ad hoc λειτουργία.
- Να παρέχουμε στην WiShield έγκυρες IP address, Subnet mask, Gateway IP και SSID ώστε να μπορεί να συνδεθεί στο AP και να επικοινωνεί σωστά μ' αυτό.

### 9.5. Εγκατάσταση και ρύθμιση του Arduino IDE

Για να αναπτύξουμε την υλοποίηση είναι απαραίτητο να εγκαταστήσουμε το Arduino IDE και να ρυθμίσουμε τις παραμέτρους που απαιτούνται για την χρήση του. Το Arduino IDE δεν παρέχεται με installer (αυτόματο εγκαταστάτη λογισμικού) και πρέπει να το εγκαταστήσουμε εμείς στην επιθυμητή τοποθεσία στον Η/Υ μας. Αρχικά θα κατεβάσουμε το Arduino IDE από την διεύθυνση <sup>28</sup> στο διαδίκτυο απ' όπου και μας παρέχεται δωρεάν, στην αντίστοιχη έκδοση του λειτουργικού συστήματος του Η/Υ μας.

Το Arduino IDE έρχεται σε συμπιεσμένο αρχείο (zip file) και θα πρέπει να το αποσυμπέσουμε στον Η/Υ μας στην επιθυμητή τοποθεσία με αντίστοιχη εφαρμογή αποσυμπίεσης (π.χ. rar). Μετά θα πρέπει να εγκαταστήσουμε τους αντίστοιχους drivers για το FTDI chip ώστε να μπορούμε να έχουμε επικοινωνία μεταξύ του Arduino IDE και του Arduino μέσω σειριακής θύρας και του πρωτοκόλλου STK 500. Και τέλος, να εγκαταστήσουμε τις απαραίτητες C++ βιβλιοθήκες για την υλοποίηση.

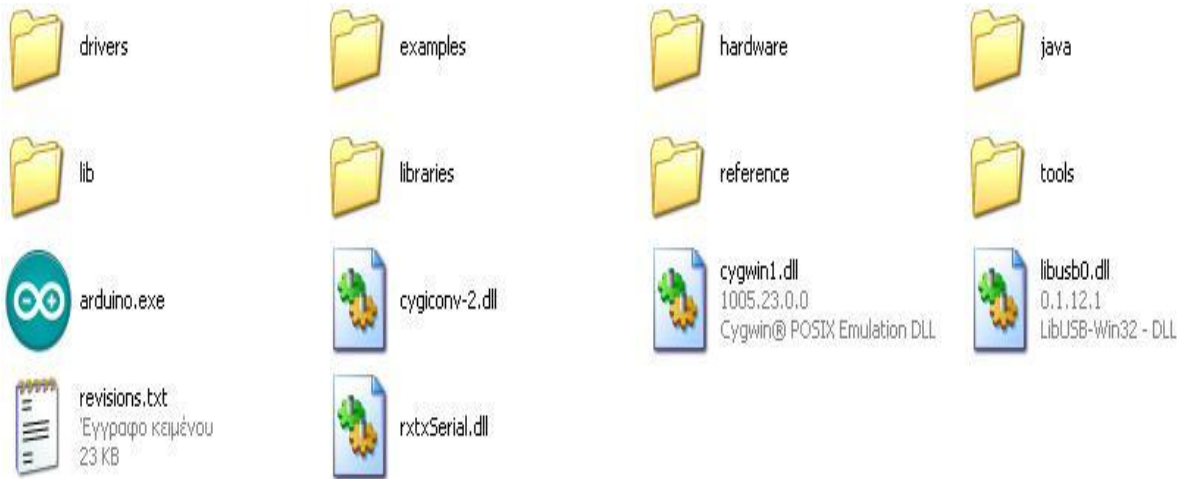
---

<sup>28</sup> <http://arduino.cc/en/Main/Software>

**Σημείωση:** οι παραπάνω διαδικασίες διαφοροποιούνται ανάλογα το λειτουργικό σύστημα του Η/Υ μας. Θα δείξουμε (σε περιβάλλον *Windows XP*) μόνο την εγκατάσταση των C++ βιβλιοθηκών που είναι κοινή για όλα τα λειτουργικά συστήματα. Πληροφορίες για τα υπόλοιπα, παρέχονται στην διεύθυνση<sup>29</sup> του διαδικτύου.

### Εγκατάσταση των C++ βιβλιοθηκών:

Έχοντας αποσυμπίσει το Arduino IDE στην επιθυμητή τοποθεσία έχουμε την εξής δομή αρχείων και φακέλων:



**Εικόνα 9. 1: Δομή αρχείων εντός του Arduino IDE.**

Μετακινούμαστε στον φάκελο **libraries** και δημιουργούμε τους εξής φακέλους εντός αυτού: *OneWire*, *DS18B20*, *WiShield* και *Dataflash*. Έπειτα αποσυμπιέζουμε τις C++ βιβλιοθήκες που θα τις κατεβάσουμε από τις διευθύνσεις<sup>30 31 32 33</sup> του διαδικτύου. Έπειτα κάνουμε διπλό κλικ στο εκτελέσιμο αρχείο *arduino.exe* ώστε οι βιβλιοθήκες να αναγνωριστούν από το Arduino IDE και να είμαστε σε θέση να τις χρησιμοποιήσουμε.

### Ρύθμιση του Arduino IDE:

Έχοντας ολοκληρώσει τις παραπάνω ενέργειες το Arduino IDE είναι έτοιμο προς χρήση και το μόνο που απομένει είναι να ρυθμίσουμε τον τύπο αναπτυξιακής πλακέτας Arduino που θα χρησιμοποιήσουμε, στην περίπτωση μας Arduino Duemilanove. Πατώντας την επιλογή Tools→Board→ Arduino Duemilanove or Nano w/Atmega 328 έχουμε επιλέξει το τύπο Arduino που επιθυμούμε.

**Σημείωση:** η παραπάνω ρύθμιση κρίνεται απαραίτητη για την σωστή λειτουργία του Arduino και την αποφυγή μόνιμης βλάβης στο hardware του.

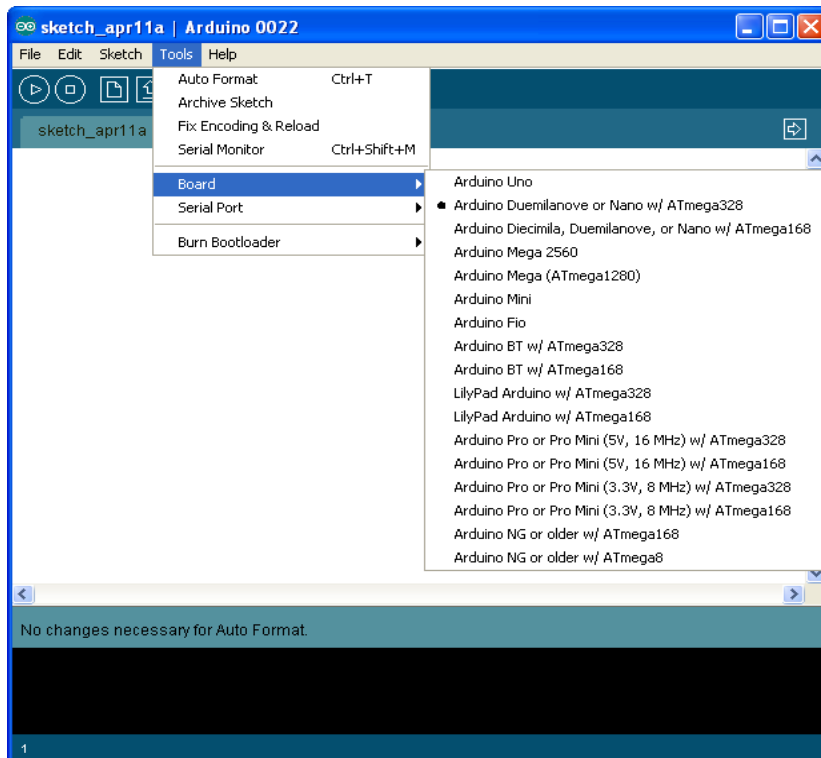
<sup>29</sup> <http://arduino.cc/en/Guide/HomePage>

<sup>30</sup> <https://github.com/asynclabs/WiShield> φάκελος WiShield

<sup>31</sup> <https://github.com/asynclabs/dataflash> φάκελος Dataflash

<sup>32</sup> [http://download.milesburton.com/Arduino/MaximTemperature/DallasTemperature\\_372Beta.zip](http://download.milesburton.com/Arduino/MaximTemperature/DallasTemperature_372Beta.zip) φάκελος DS18B20

<sup>33</sup> [http://www.pjrc.com/teensy/arduino\\_libraries/OneWire.zip](http://www.pjrc.com/teensy/arduino_libraries/OneWire.zip) φάκελος OneWire



Εικόνα 9. 2: Επιλογή τύπου Arduino.

## 9.6. Παραγωγή εφαρμογών μέσω του Arduino IDE

Έχοντας ολοκληρώσει την εγκατάσταση και παραμετροποίηση του Arduino IDE, είμαστε σε θέση να αναπτύξουμε την υλοποίηση του τηλεμετρικού υποσυστήματος. Θα αναλύσουμε πολύ σύντομα τη διαδικασία ανάπτυξης εφαρμογών στο Arduino IDE, όπως επίσης και τα απαιτούμενα δομικά στοιχεία αυτής.

### Η διαδικασία παραγωγής εφαρμογών:

Η διαδικασία παραγωγής εφαρμογών στο Arduino IDE είναι ιδιαίτερα απλή. Κατά την διάρκεια ανάπτυξης της υλοποίησης, θα χρησιμοποιήσουμε μόνο δυο λειτουργίες απ' αυτές που μας παρέχει το Arduino IDE. Αυτές είναι αρκετές για την παράγωγη κάθε εφαρμογής. Είναι οι λειτουργίες **Verify** ή **Compile** και **Upload**.



*Verify ή Compile*

*Upload*

Εικόνα 9. 3: Λειτουργίες Compile/Verify και Upload στο Arduino IDE.

**Verify ή Compile:** με την λειτουργία αυτή το Arduino IDE εκτελεί την διαδικασία **compile**<sup>34</sup> για τον κώδικα που αναπτύξαμε και αν δεν επισημάνει λάθη κατά την διαδικασία compile τότε δημιουργεί τα εξής αρχεία:

<sup>34</sup> μετατρέπει/μεταφράζει κείμενο γραμμένο σε μια γλώσσα προγραμματισμού (γλώσσα πηγαίου κώδικα) σε μια άλλη γλώσσα προγραμματισμού (τη γλώσσα στόχο π.χ. Assembly).

\*.cpp → πηγαίος κώδικας C++ της εφαρμογής μας.

\*.cpp.hex → τα περιεχόμενα της EEPROM (αν έχει ήδη ή απαιτεί η εφαρμογή δεδομένα εντός της)

\*.cpp.elf → executable linkable format αρχείο για την περαιτέρω ανάλυση της εφαρμογής.

\*.cpp.hex → δεδομένα σε δεκαεξαδική μορφή προς εκτέλεση που θα μεταφερθούν στο Arduino.

\*.cpp.o → object αρχείο που συνδυάζει πηγαίο κώδικα με τις απαραίτητες βιβλιοθήκες (C++).

**Σημείωση:** όπου \* το όνομα του αρχείου της εφαρμογής π.χ. test.cpp.

**Upload:** μέσω της λειτουργίας Upload το εκτελέσιμο αρχείο σε hex μορφή μεταφέρεται στο Arduino διάμεσο της USB θύρας και του πρωτοκόλλου STK 500.

### Τα απαραίτητα δομικά στοιχεία μιας εφαρμογής:

Για να θεωρηθεί μια εφαρμογή έγκυρη στο Arduino IDE θα πρέπει να περιέχει υποχρεωτικά δυο συναρτήσεις, την **setup** και την **loop**. Η συνάρτηση **setup** είναι υπεύθυνη για τις απαραίτητες αρχικοποιήσεις παραμέτρων σε software και hardware επίπεδο. Η συνάρτηση **loop** εκτελείται συνεχώς, ενώ ο κώδικας της εφαρμογής μας εκτελείται.

**Σημείωση:** η χρήση των συναρτήσεων **setup** και **loop** θα αναλυθεί περαιτέρω κατά την διαδικασία ανάλυσης του κώδικα υλοποίησης του τηλεμετρικού υποσυστήματος.

## 9.7. Οι στόχοι υλοποίησης του τηλεμετρικού υποσυστήματος

Έχοντας ολοκληρώσει την ανάλυση απαιτήσεων και την εγκατάσταση του Arduino IDE, είναι πλέον σημαντικό να καθορίσουμε τους στόχους της υλοποίησης του τηλεμετρικού υποσυστήματος. Οι στόχοι της υλοποίησης θα περιγραφθούν σύντομα σ' αυτή την ενότητα και θα αναλυθούν εκτενώς στις επόμενες.

Συνοπτικά οι στόχοι που θα υλοποιηθούν μέσω της γλώσσας προγραμματισμού C++ είναι οι εξής:

- Υλοποίηση ενός software timer που θα μετρά τον χρόνο λειτουργίας σε δευτερόλεπτα.
- Κάθε 6 δευτερόλεπτα θα μετρά και θα καταγράφει την θερμοκρασία και την υγρασία όπως επίσης και θα ελέγχει αν έχει ανιχνευτεί κίνηση στον χώρο που είναι τοποθετημένο το τηλεμετρικό υποσύστημα.
- Κάθε 30 δευτερόλεπτα θα υπολογίζεται η μέση θερμοκρασία και υγρασία, επιπλέον θα υπολογίζεται η μέγιστη και η ελάχιστη τιμή θερμοκρασίας και υγρασίας.
- Κάθε 60 δευτερόλεπτα θα αποθηκεύει στην μνήμη EEPROM τις τιμές μέγιστης και ελάχιστης υγρασίας και θερμοκρασίας.
- Κάθε μια ώρα ή 3600 δευτερόλεπτα αποθηκεύει στην Dataflash μνήμη της WiShield την μέση τιμή θερμοκρασίας και υγρασίας που έχει ήδη υπολογιστεί.
- Δημιουργία δυο ιστοσελίδων δυναμικού περιεχομένου βάσει των εκάστοτε μετρήσεων.

## 9.8. Παρουσίαση και ανάλυση υλοποίησης

Σε προγραμματιστικό επίπεδο η υλοποίηση αποτελείται από τρία βασικά μέρη, τις συναρτήσεις `setup`, `loop`, `servePage`. Κάθε μια εξυπηρετεί συγκεκριμένη λειτουργία και όλες μαζί υλοποιούν το σύνολο των λειτουργιών. Συγκεκριμένα η `setup` μας επιτρέπει να ρυθμίσουμε όλες τις απαραίτητες παραμέτρους που απαιτεί η υλοποίηση πριν την κάθε εκτέλεση του κώδικα. Η `loop` εκτελείται συνεχώς και μας παρέχει τις μετρήσεις και τις καταγραφές που έχουμε ορίσει στους στόχους της υλοποίησης όπως και επίσης την δυνατότητα να διακομίσουμε τις δυο ιστοσελίδες που έχουμε αναπτύξει για την παρουσίαση των μετρήσεων. Τέλος, η `servePage` μέσω ειδικών συνδυασμών `html strings` (αλφαριθμητικά ιστοσελίδας) μας παρέχει τις ιστοσελίδες που απαιτούνται.

## 9.9. Ανάλυση της συνάρτησης `setup`

Μέσω της συνάρτησης `setup` θα ρυθμίσουμε `software` και `hardware` παραμέτρους. Συγκεκριμένα απενεργοποιούμε προσωρινά το AT45DB161D (IC μνήμης τύπου Dataflash) και το g2100 (IC για το πρωτόκολλο 802.11b) της WiShield. Τα απενεργοποιούμε για να αποφύγουμε τα `hardware conflicts` (συγκρούσεις υλικού) μεταξύ τους μέχρι την ολοκλήρωση της `setup`. Επίσης απενεργοποιούμε τα `interrupt` για το g2100 τα οποία είναι ενεργοποιημένα εξ αρχής. Αν δεν τα απενεργοποιήσουμε, κάθε φορά που γίνεται κλήση της συνάρτησης `servePage` το `interrupt` που θα παραχθεί θα μας αποτρέψει να εκτελούμε μετρήσεις. Τέλος, αρχικοποιούμε τον αισθητήρα DS18B20 ώστε να είναι σε θέση να μας παρέχει μετρήσεις μέσω του διαύλου 1 Wire.

## 9.10. Ανάλυση της συνάρτησης `loop`

Στην συνάρτηση `loop` υλοποιούμε ένα `software timer` (χρονόμετρο μέσω λογισμικού) το οποίο μας παρέχει την δυνατότητα να μετράμε τον χρόνο σε δευτερόλεπτα. Χρησιμοποιώντας το μας δίνεται η δυνατότητα να καθορίσουμε χρονικά διαστήματα στα οποία μπορούμε να εκτελούμε μετρήσεις και υπολογισμούς. Έχοντας ήδη καθορίσει τα χρονικά διαστήματα που απαιτούνται στους στόχους της υλοποίησης απομένει να αναφέρουμε περιληπτικά τις λειτουργίες που εκτελεί η συνάρτηση `loop`.

Στο πρώτο δευτερόλεπτο (και μόνο σε αυτό) ανακτάμε από την μνήμη EEPROM όπου ήταν αποθηκευμένες από προηγούμενες εκτελέσεις του κώδικα (μεταξύ διαδοχικών `reset` ή `power off`), την μέγιστη και ελάχιστη τιμή θερμοκρασίας και της υγρασίας. Κάθε 6 δευτερόλεπτα μετράμε την τιμή της θερμοκρασίας και της υγρασίας κι ελέγχουμε αν έχουμε ανιχνεύσει κίνηση. Κάθε 30 δευτερόλεπτα υπολογίζουμε μέσω των μετρήσεων που εκτελούμε ανά 6 δευτερόλεπτα (5 μετρήσεις σε 30 δευτερόλεπτα), την μέση θερμοκρασία και υγρασία όπως και την ελάχιστη αλλά και την μέγιστη τιμή τους. Κάθε λεπτό αποθηκεύουμε την μέγιστη και ελάχιστη τιμή θερμοκρασίας και υγρασίας στην μνήμη EEPROM ώστε να μπορούμε να τις διατηρήσουμε σε περίπτωση `reset` ή `power off`. Τέλος σε κάθε εκτέλεση της `loop` αν επιθυμούμε μας παρέχεται η δυνατότητα να μας διακομιστεί κάποια από τις δυο ιστοσελίδες.

## 9.11. Ανάλυση της συνάρτησης `servePage`

Με την συνάρτηση `servePage` συνθέτουμε τις δυο ιστοσελίδες που έχουμε αναπτύξει. Συνοπτικά η μέθοδος σύνθεσης τους λειτουργεί ως εξής: μέσω των `html strings` που έχουμε αποθηκεύσει στην μνήμη που είναι αποθηκευμένος ο κώδικας της υλοποίησης (ονομασία `PROGMEM` και μέρος της Flash μνήμης) του ATmega328P παράγουμε με κατάλληλους συνδυασμούς αυτών πλήρεις ιστοσελίδες, επίσης με παρόμοια μέθοδο προσθέτουμε ένα `css style sheet`. Τέλος προσθέτουμε στην ιστοσελίδα ένα αρχείο εικόνας που είναι κι αυτό αποθηκευμένο στην `PROGMEM` μνήμη.

### Πως η `servePage` συνθέτει τα `html strings` και τις τιμές μετρήσεων σε ιστοσελίδες

Για να συνθέσουμε ένα τμήμα κειμένου σε `html` το περικλείουμε με τα ανάλογα `tags` μορφοποίησης π.χ. αν επιθυμούμε `bold` μορφοποίηση ο κώδικας θα 'ναι ο εξής: `<b>κείμενο</b>`. Το

πρόβλημα όμως εμφανίζεται όταν επιθυμούμε να συνθέσουμε ταυτόχρονα κείμενο με μορφοποίηση και τις τιμές μετρήσεων. Οι τιμές των μετρήσεων είναι ουσιαστικά μεταβλητές, που σημαίνει ότι οι τιμή τους ανά πάσα στιγμή ενδέχεται να αλλάξει. Εν ολίγοις η html αδυνατεί να μορφοποιήσει κείμενο το οποίο δεν έχει σταθερή τιμή και για να επιτευχτεί η διαμόρφωση που επιθυμούμε απαιτείται μια νέα μεθοδολογία μορφοποίησης.

Η μεθοδολογία που χρησιμοποιείται από την υλοποίηση αλλάζει το προηγούμενο παράδειγμα κώδικα html ως εξής: `<b>κείμενο τιμή μέτρησης κείμενο</b>` η διάφορα σε σχέση με πριν είναι ότι ο κώδικας τώρα αποτελείται από 3 ξεχωριστά μέρη το πρώτο είναι `<b>κείμενο`, το δεύτερο `τιμή μέτρησης` και το τρίτο `κείμενο</b>`. Σε κάθε περίπτωση όμως και σύμφωνα με τις ανάγκες, συνθέτουμε κατάλληλα με την παραπάνω μεθοδολογία τις δυο ιστοσελίδες της υλοποίησης.

Είναι σημαντικό να αναφέρουμε ότι η συνάρτηση `servePage` «τεμαχίζει» τις ιστοσελίδες της υλοποίησης σε τμήματα html string και τιμών μετρήσεων σύμφωνα με την παραπάνω μεθοδολογία. Υστερά αυτά τα τμήματα αποστέλλονται μέσω της WiShield στον H/Y που αιτείται τις ιστοσελίδες και ο web browser (του H/Y) τις ανασυνθέτει σε ιστοσελίδες.

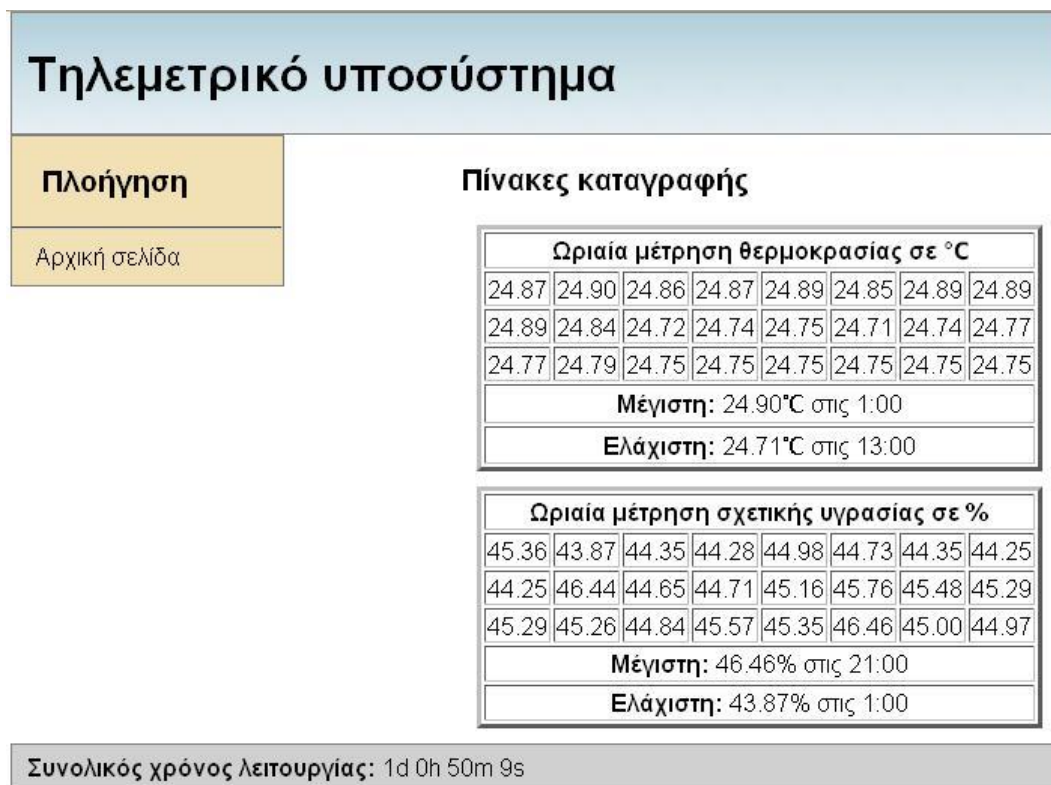
## 9.12. Παρουσίαση των ιστοσελίδων της υλοποίησης

Στην πρώτη ιστοσελίδα παρουσιάζεται η παρούσα, μέση, μέγιστη και ελάχιστη τιμή της θερμοκρασίας και της υγρασίας. Ακόμη παρουσιάζεται αν ανιχνεύτηκε κίνηση. Στην περίπτωση που έχει ανιχνευτεί, μας παρουσιάζεται το πλήθος των ανιχνεύσεων, πριν πόση ώρα ανιχνεύτηκε η τελευταία αλλά και αν έχουμε κίνηση σε εξέλιξη.

Τηλεμετρικό υποσύστημα	
Πλοήγηση	<b>Θερμοκρασία</b>
	Τρέχουσα: 24.31°C Ελάχιστη: 22.00°C Μέση: 24.31°C Μέγιστη: 26.25°C
Καταγραφές 24ωρου	<b>Σχετική υγρασία</b>
	Τρέχουσα: 42.93% Ελάχιστη: 17.92% Μέση: 42.45% Μέγιστη: 55.67%
	<b>Ανίχνευση κίνησης</b>
	Πλήθος ανιχνεύσεων κίνησης: 8 Η τελευταία πριν: 0h 0m 3s Κίνηση σε εξέλιξη: ΟΧΙ
Συνολικός χρόνος λειτουργίας: 1d 0h 50m 4s	

Εικόνα 9. 4: Η αρχική (πρώτη) ιστοσελίδα.

Στην δεύτερη ιστοσελίδα παρουσιάζεται σε δυο πίνακες η καταγραφή της θερμοκρασίας και της υγρασίας ανά ώρα για μια περίοδο 24 ωρών. Το εικοσιτετράωρο αυτό χωρίζεται σε τρία οκτάωρα και η ώρα 0:00 είναι το πρώτο από τα αριστερά κελί της πρώτης σειράς. Επίσης σε κάθε μια από τις ιστοσελίδες υπάρχει ένα χρονόμετρο στο κάτω μέρος που μας δείχνει το συνολικό χρόνο λειτουργίας του υποσυστήματος.



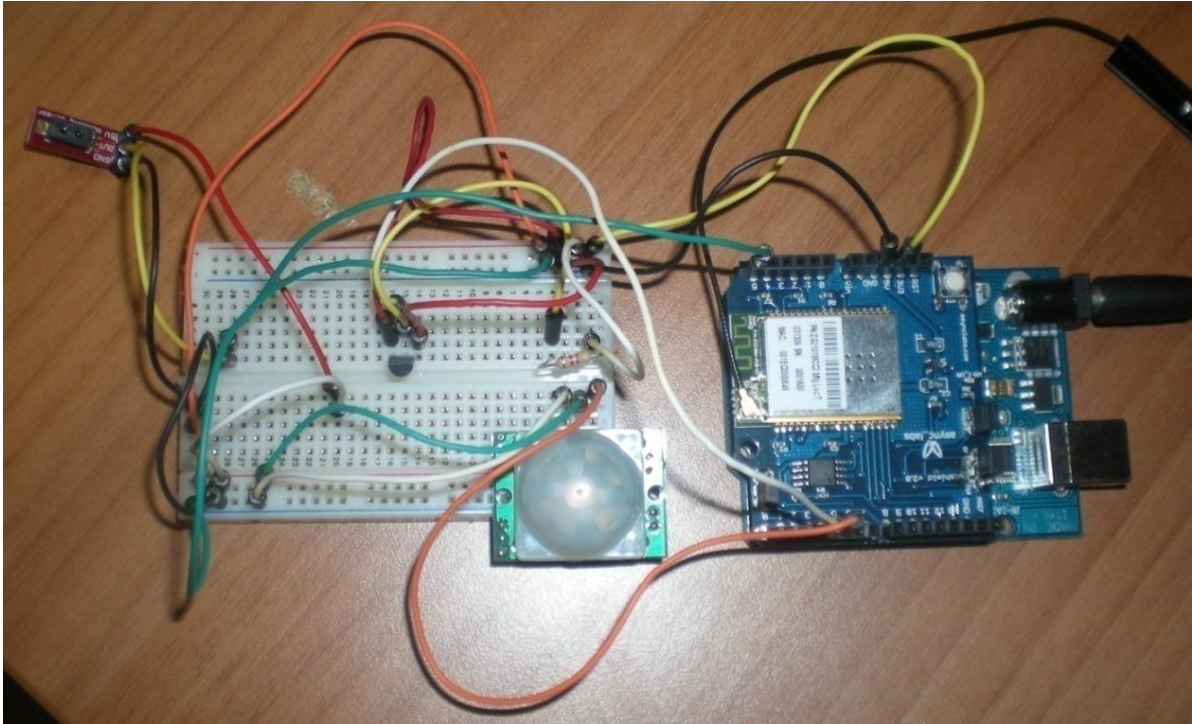
Εικόνα 9. 5: Η δεύτερη ιστοσελίδα.

### 9.13. Παρουσίαση του κυκλώματος της υλοποίησης

Έχοντας ολοκληρώσει σε προγραμματιστικό επίπεδο την υλοποίηση πρέπει να συνθέσουμε το κατάλληλο κύκλωμα που να περιλαμβάνει όλους τους αισθητήρες και τις αναπτυξιακές πλακέτες Arduino και WiShield. Στο κύκλωμα αυτό υπάρχουν κάποιες ιδιαίτερες απαιτήσεις, όπως τιμή εφαρμοζόμενης τάσης, επιπλέον εξαρτήματα αλλά και η σύνδεση τους σε συγκεκριμένα pin του Arduino. Οι απαιτήσεις συγκεκριμένα είναι οι εξής:

- Παροχή τάσης 5 Volt μέσω του pin 5V του Arduino.
- Γείωση μέσω του pin GND του Arduino.
- Σύνδεση του αισθητήρα Dallas DS18B20 στο ψηφιακό pin 4 του Arduino.
- Σύνδεση του αισθητήρα Honeywell HHH-4030 στο αναλογικό pin 5 του Arduino.
- Σύνδεση του αισθητήρα PIR Parallax 555 - 280227 στο ψηφιακό pin 5 του Arduino.
- Τοποθέτηση αντίστασης 4,7 kΩ μεταξύ του ψηφιακού pin 4 και του pin εξόδου του Dallas DS18B20.
- Προσαρμογή της WiShield στο Arduino (όπως στην εικόνα 8.7).
- Παροχή συνεχούς τάσης 9 Volt στο Arduino μέσω μετασχηματιστή (220V AC σε 9V DC).





Εικόνα 9. 6: Φωτογραφία του κυκλώματος υλοποίησης.

**Σημείωση:** Το παραπάνω κύκλωμα το παραθέτουμε μόνο για λόγους επίδειξης. Η μορφή του μπορεί να μεταβληθεί, μοναδικός περιορισμός στην υλοποίηση είναι οι απαιτήσεις της προηγούμενης ενότητας.

#### 9.14. Ρύθμιση και εγκατάσταση του τηλεμετρικού υποσυστήματος

Εφόσον αποφασίσουμε την τοποθεσία εγκατάστασης του τηλεμετρικού υποσυστήματος οφείλουμε να εκτελέσουμε μια σειρά ρυθμίσεων πριν το θέσουμε σε λειτουργία. Στην ενότητα **Απαιτήσεις του ασύρματου πρωτόκολλου**, αναφέραμε τι απαιτήσεις πρέπει να ικανοποιήσουμε ώστε το τηλεμετρικό υποσύστημα να μπορεί να έχει ασύρματη επικοινωνία με τον Η/Υ μας. Συγκεκριμένα, στον κώδικα C++ της υλοποίησης θα πρέπει να τροποποιήσουμε τα πεδία που αναφέρονται σε παραμέτρους όπως διεύθυνση IP, Gateway IP, subnet mask, SSID, security passphrase και τοπολογία σύνδεσης (adhoc ή infrastructure). Τροποποιούμε τα παραπάνω, στις τιμές που έχουν οριστεί στο AP, στο οποίο θα συνδεθεί το τηλεμετρικό υποσύστημα.

**Σημείωση:** Λόγω ότι η WiShield δεν διαθέτει ενσωματωμένη την λειτουργία DHCP οι παραπάνω ρυθμίσεις κρίνονται απαραίτητες.





## 10. Συμπεράσματα

Η υλοποίηση του τηλεμετρικού αποστήματος, μας παρέχει την δυνατότητα να μελετήσουμε εύκολα τις περιβαλλοντικές συνθήκες του χώρου που είναι εγκατεστημένο αρκεί, αυτός να καλύπτεται από ασύρματη υποδομή πρωτοκόλλου 802.11b. Τα οφέλη χρήσης του σημαντικά, τα κυριότερα το μέγεθος του, το οποίο είναι ιδιαίτερα μικρό, επίσης οι απαιτήσεις του σε ενέργεια αλλά και σε τεχνογνωσία που απαιτείται για την διαχείριση του είναι επίσης χαμηλές. Τέλος, σημαντικό όφελος αποτελεί το κόστος του, το οποίο είναι αρκετά χαμηλό και το οποίο μάλιστα είναι ιδιαίτερα προσιτό σε σχέση με τα αντίστοιχα ολοκληρωμένα τηλεμετρικά συστήματα που είναι διαθέσιμα στο εμπόριο.

### 10.1. Μελλοντικές επεκτάσεις

Ως επέκταση του τηλεμετρικού υποσυστήματος μπορούμε να ορίσουμε εναλλακτικά σενάρια χρήσης, τα οποία όμως, πλήρως καλύπτονται από τις δυνατότητες του υπάρχοντος hardware. Συγκεκριμένα:

- Χρησιμοποιώντας μια επιπλέον ομάδα ίδιων αισθητήρων, η ακρίβεια του αλλά και η περιοχή κάλυψης θα αυξηθούν αισθητά.
- Η μετατροπή του σε σταθμό πρόβλεψης καιρού, με μόνη αλλαγή του PIR αισθητήρα, με έναν συμβατό αισθητήρα ατμοσφαιρικής πίεσης.
- Λόγω ότι τα schematics του Arduino αλλά και της WiShield είναι διαθέσιμα υπό την άδεια Creative Commons μπορούμε να τροποποιήσουμε τα κυκλώματα τους ώστε να καταλαμβάνουν ακόμη λιγότερο χώρο ή να τους προσθέσουμε επιπλέον hardware.



# Βιβλιογραφία

- [1] Frank Carden, Robert Henry, Russ Jedlicka, Telemetry Systems Engineering, Artech House Telecommunications Library 2002
- [2] Yuan Taur, Tak H. Ning, Fundamentals of Modern VLSI Devices, Cambridge University Press 2009
- [3] John Crisp, Introduction to Microprocessors and Microcontrollers, Elsevier 2004
- [4] Dale Wheat, Arduino Internals, Apress 2011
- [5] Jacob Fraden, Handbook of Modern Sensors, Springer 2010
- [6] Robert P. Benedict, Fundamentals of Temperature, Pressure and Flow Measurements, John Wiley & Sons 1984
- [7] John M. Castillo, Relative Humidity: Sensors, Management and Environmental Effects, Nova Science 2011
- [8] William Trochim, James P. Donnelly, The Research Methods Knowledge Base, Atomic Dog 2007
- [9] Maxim DS18B20, <http://datasheets.maxim-ic.com/en/ds/DS18B20.pdf>
- [10] Honeywell HIH-4030, [http://sensing.honeywell.com/index.php?ci\\_id=51625&la\\_id=1](http://sensing.honeywell.com/index.php?ci_id=51625&la_id=1)
- [11] Parallax PIR 555-280227, <http://www.parallax.com/Portals/0/Downloads/docs/prod/sens/555-28027-PIRSensor-v2.1.pdf>
- [12] Andrew S. Tanenbaum, Computer Networks, Prentice Hall 2010
- [13] WiShield, [http://asynclabs.com/wiki/index.php?title=WiShield\\_2.0](http://asynclabs.com/wiki/index.php?title=WiShield_2.0)



# Παραρτήματα

## Παράρτημα Α κώδικας υλοποίησης σε C++

```
/******
```

```
Filename:      Thesis_code.cpp
Description:   κώδικας υλοποίησης του τηλεμετρικού υποσυστήματος σε C++
```

```
*****/
```

Στην παρούσα υλοποίηση ο κώδικας χωρίζεται σε τρία βασικά μέρη `setup`, `servePage` και `loop`.

Στο κομμάτι `setup` ουσιαστικά αρχικοποιούμε τις παραμέτρους `hardware` και `software` που απαιτούνται. Ιδιαίτερη προσοχή πρέπει να δοθεί στην σωστή αρχικοποίηση της `Wishield` ώστε να μην έχει `hardware conflicts` με το `AT45DB161D`. Για αυτό το λόγο απενεργοποιούμε το `AT45DB161D` μέχρι να το χρειαστούμε. Επίσης θέτουμε το `pin 8` σε `HIGH` κατάσταση ώστε το `g2100 802.11b chip` της `Wishield` να μην παράγει `interrupt` και να σταματήσει την `loop`.

Στο κομμάτι `loop` υλοποιούμε έναν `software timer` που μετρά δευτερόλεπτα και κάνουμε μετρήσεις και υπολογισμούς. Αρχικά διαβάζουμε την μέγιστη και ελάχιστη τιμή θερμοκρασίας αλλά όπως και της υγρασίας από την `EEPROM` που ήταν αποθηκευμένες εκεί από προηγούμενες εκτελέσεις κώδικα (μεταξύ διαδοχικών `reset`). Κάθε 6 δευτερόλεπτα μετράμε θερμοκρασία και υγρασία κι αν έχουμε ανιχνεύσει κίνηση. Κάθε 30 δευτερόλεπτα υπολογίζουμε την μέση θερμοκρασία και υγρασία όπως και την ελάχιστη αλλά και την μέγιστη τους. Κάθε λεπτό αποθηκεύουμε την μέγιστη και ελάχιστη τιμή θερμοκρασίας και υγρασίας στην `EEPROM` ώστε να μπορούμε να τις διατηρούμε σε περίπτωση `reset` ή `power off`. Τέλος μέσω της `WiServer.server_task` που εκτελείται σε κάθε εκτέλεση της `loop`, ξεκινάμε την διακώμηση ιστοσελίδων.

Στο κομμάτι `servePage` συνθέτουμε τις δυο ιστοσελίδες που έχουμε αναπτύξει. Η μέθοδος σύνθεσης τους είναι η έξης: μέσω των `html strings` που έχουμε αποθηκεύσει στην `PROGMEM` μνήμη παράγουμε πλήρεις ιστοσελίδες με κατάλληλους συνδυασμούς αυτών. Επίσης προσθέτουμε `css stylesheet` και μια εικόνα που είναι επίσης αποθηκευμένα στην `PROGMEM` μνήμη.

Στην πρώτη ιστοσελίδα παρουσιάζεται η παρούσα, μέση, μέγιστη και ελάχιστη τιμή της θερμοκρασίας και της υγρασίας. Ακόμη παρουσιάζεται αν ανιχνεύτηκε κίνηση αν ναι, το πλήθος τους και αν έχουμε κίνηση σε εξέλιξη.

Στην δεύτερη ιστοσελίδα παρουσιάζεται η καταγραφή της θερμοκρασίας και της υγρασίας ανά ώρα για ένα εικοσιτετράωρο. Κάθε εικοσιτετράωρο χωρίζεται σε τρία οκτώωρα και η ώρα 0:00 είναι στο πρώτο αριστερά κελί.

Σε κάθε μια από τις ιστοσελίδες υπάρχει ένα χρονόμετρο στο `footer` που μας δείχνει το συνολικό χρόνο λειτουργίας του υποσυστήματος.

Επίσης υλοποιήθηκαν τέσσερις επιπλέον συναρτήσεις. Η `get_temperature` υπολογίζει την θερμοκρασία. Η `get_humidity` υπολογίζει την υγρασία με όρισμα την παρούσα θερμοκρασία.

Η `uptime` παίρνει όρισμα τον αριθμό των δευτερολέπτων που πέρασαν από την αρχή εκτέλεσης του κώδικα και υπολογίζει τις μέρες, ώρες, λεπτά, και δευτερόλεπτα λειτουργίας.

Και τέλος η `FloatFromByteArray` συνθέτει έναν `float` από 4 `byte`.

```
*****/
```

```
#include <dataflash.h> //βιβλιοθήκη για χρήση του DataFlash AT45DB161D
```

```

#include <WiServer.h> //βιβλιοθήκη για την διακώμιση ιστοσελίδων
#include "gradient.h" //η εικόνα gradient.jpg έχει μετατραπεί σε πίνακα PROGMEM
για χρήση ως header
extern "C"
{
#include "g2100.h" //βιβλιοθήκη για low level control του g2100 chip
}
#include <OneWire.h> //βιβλιοθήκη για low level control του One Wire bus
#include <DallasTemperature.h> //βιβλιοθήκη για low level control του DS18B20

#define ONE_WIRE_BUS 4 //σύνδεση του DS18B20 στο pin 4
OneWire oneWire(ONE_WIRE_BUS); //δημιουργία αντικειμένου κλάσης OneWire με όρισμα
το pin 4
DallasTemperature sensors(&oneWire); //δημιουργία αντικειμένου κλάσης
DallasTemperature με όρισμα το oneWire

#define FLASH_SLAVE_SELECT 7 //το pin 7 ως FLASH_SLAVE_SELECT
#define WIFI_SLAVE_SELECT 10 //το pin 10 ως WIFI_SLAVE_SELECT

//---ομάδα macro που χειρίζεται την EEPROM ως ενιαίο κομμάτι μνήμης μέσω μίας
struct---//
#include <avr/eeprom.h>
#define eeprom_read_to(dst_p, eeprom_field, dst_size) eeprom_read_block(dst_p,
(void *)offsetof(__eeprom_data, eeprom_field), MIN(dst_size,
sizeof((__eeprom_data*)0)->eeprom_field))
#define eeprom_read(dst, eeprom_field) eeprom_read_to(&dst, eeprom_field,
sizeof(dst))
#define eeprom_write_from(src_p, eeprom_field, src_size) eeprom_write_block(src_p,
(void *)offsetof(__eeprom_data, eeprom_field), MIN(src_size,
sizeof((__eeprom_data*)0)->eeprom_field))
#define eeprom_write(src, eeprom_field) { typeof(src) x = src;
eeprom_write_from(&x, eeprom_field, sizeof(x)); }
#define MIN(x,y) ( x > y ? y : x )
//-----//

//---global μεταβλητές-----//
float
humidity_loop,average_temperature,max_temperature,average_humidity,max_humidity,hum
idity_now,temperature_now,temperature_loop,temperature_min=99.9,humidity_min=99.9,t
emperature_array[5],humidity_array[5];
volatile unsigned long seconds,minutes,hours,days; //μεταβλητές που μας δείχνουν
το συνολικό χρόνο λειτουργίας
static unsigned long lastTick = 0; //προσωρινή μεταβλητή για τον υπολογισμό ενός
δευτερολέπτου
volatile static unsigned long timer,pir_timer; //κυρίως χρονόμετρο & χρονόμετρο
του pir αισθητήρα
int pir=0; //πλήθος ανιχνεύσεων κίνησης
//-----//

//---struct που καταγράφει τις μέγιστες και ελάχιστες τιμές της υγρασίας &
θερμοκρασίας στην EEPROM---//
struct __eeprom_data {
const float eeprom_max_temp;
const float eeprom_min_temp;
const float eeprom_min_hum;
const float eeprom_max_hum;
};

//---Παράμετροι ασύρματης επικοινωνίας με το AP-----//

U8 local_ip[] = {xxx,xxx,xxx,xxx}; //IP της WiShield
U8 gateway_ip[] = {xxx,xxx,xxx,xxx}; //Gateway IP
U8 subnet_mask[] = {xxx,xxx,xxx,xxx}; //Subnet mask
const prog_char ssid[] PROGMEM = {"xxx"}; //SSID, Max 32 bytes
U8 security_type = 0; //τύπος ασφάλειας 0==open, 1==WEP, 2==WPA, 3==WPA2
U8 wireless_mode = 1; //τύπος δικτύου 1==Infrastructure, 2==Ad-hoc
const prog_char security_passphrase[] PROGMEM = {""}; //AP password
prog_uchar wep_keys[] = {}; //wep keys

```

```

U8 security_passphrase_len, ssid_len; //αριθμός χαρακτήρων των keys

//-----//

//---html strings για την δυναμική δημιουργία ιστοσελίδων-----//
const prog_char css[] PROGMEM= {"html,body{margin:0;padding:0}body{font: 100%
arial,sans-serif;text-align:center}p{margin:0 10px 10px} a{padding:5px; text-
decoration:none; color:#000000;} div#header{border:1px solid gray;background-
color:#CBE9F2;background-image:url('gradient.jpg');background-repeat:repeat-
x;}div#header h1{height:80px;line-height:80px;margin:0;padding-
left:10px;}div#container{text-align:left}div#content p{line-
height:1.4}div#navigation{background:#F6F0E0;}div#navigation ul{margin:15px 0;
padding:0; list-style-type:none;}div#navigation li{margin-
bottom:5px;}div#extra{background:#F1E1B5;}div#footer{background:#D0D0D0;}div#footer
p{margin:0;padding:5px 10px}div#container{width:700px;margin:0
auto}div#wrapper{float:left;width:100%}div#content{margin-left:
300px}div#navigation{float:left;width:150px;margin-left:-550px}div#extra{border:1px
solid gray;float:left;width:180px;margin-left:-700px}div#footer{border:1px solid
gray;clear:left;width:100%}";
const prog_char header[] PROGMEM= {"<html><head><meta
content=\"text/html;charset=utf-8\" http-equiv=\"content-type\"/><title>Ιστοσελίδα
παρακολούθησης τηλεμετρικού υποσυστήματος</title><link rel=\"stylesheet\"
href=\"styles.css\" type=\"text/css\" /></head><body><div id=\"container\"><div
id=\"header\"><h1>Τηλεμετρικό υποσύστημα</h1></div><div id=\"wrapper\"><div
id=\"content\">";
const prog_char home_nav[] PROGMEM= {"</div></div><div
id=\"extra\"><p><strong><h3><p>Πλοήγηση</p></h3></strong><hr/><p><a
href=\"/log\">Καταγραφές 24ωρου</a></p>";
const prog_char log_nav[] PROGMEM= {"</div></div><div
id=\"extra\"><p><strong><h3><p>Πλοήγηση</p></h3></strong><hr/><p><a
href=\"/\">Αρχική σελίδα</a></p>";
const prog_char log_caption[] PROGMEM= {"<p><h3>Πίνακες καταγραφής</h3></p>";
const prog_char footer[] PROGMEM= {"</div><div id=\"footer\"><p><b>Συνολικός χρόνος
λειτουργίας: </b>";
const prog_char footer_end[] PROGMEM= {"</p></div></div></body></html>";
const prog_char temperature_banner[] PROGMEM= {"<p><h4>Θερμοκρασία</h4></p>";
const prog_char celsius[] PROGMEM= {"&#8451;";
const prog_char percentage[] PROGMEM= {"&#37;";
const prog_char text_now[] PROGMEM= {"<p>Τρέχουσα: ";
const prog_char text_max[] PROGMEM= {"<p>Μέγιστη: ";
const prog_char text_med[] PROGMEM= {"<p>Μέση: ";
const prog_char text_min[] PROGMEM= {"<p>Ελάχιστη: ";
const prog_char humidity_banner[] PROGMEM= {"<p><h4>Σχετική υγρασία</h4></p>";
const prog_char pir_table[] PROGMEM= {"<p><table border=\"0\">";
const prog_char home_pir[] PROGMEM= {"<p><h4>Ανίχνευση κίνησης</h4></p><hr/>";
const prog_char pir_now[] PROGMEM= {"<td>Κίνηση σε εξέλιξη:</td>";
const prog_char pir_msg_2[] PROGMEM= {"<td>Πλήθος ανιχνεύσεων κίνησης:</td>";
const prog_char log_table[] PROGMEM= {"<p><table border=\"3\">";
const prog_char log_caption_1[] PROGMEM= {"<td colspan=\"8\"><center><b>Ωριαία
μέτρηση θερμοκρασίας σε &#8451;</b></center></td>";
const prog_char log_caption_2[] PROGMEM= {"<td colspan=\"8\"><center><b>Ωριαία
μέτρηση σχετικής υγρασίας σε &#37;</b></center></td>";
const prog_char log_caption_max[] PROGMEM= {"<td colspan=\"8\"
rowspan=\"1\"><center><b>Μέγιστη:</b> ";
const prog_char log_caption_min[] PROGMEM= {"<td colspan=\"8\"
rowspan=\"1\"><center><b>Ελάχιστη:</b> ";
const prog_char log_caption_at[] PROGMEM= {"στις ";
const prog_char log_caption_end[] PROGMEM= {":00</center>";
const prog_char yes[] PROGMEM= {"<td>ΝΑΙ</td>";
const prog_char no[] PROGMEM= {"<td>ΟΧΙ</td>";
const prog_char last[] PROGMEM= {"<td>Η τελευταία πριν:</td>";
const prog_char hr[] PROGMEM= {"<hr/>";
const prog_char p_end[] PROGMEM= {"</p>";
const prog_char space[] PROGMEM= {"&nbsp;";
const prog_char tr[] PROGMEM= {"<tr>";
const prog_char tr_end[] PROGMEM= {"</tr>";
const prog_char td[] PROGMEM= {"<td>";
const prog_char td_end[] PROGMEM= {"</td>";

```



```

const prog_char table_end[] PROGMEM= {"</table></p>"};
const prog_char d[] PROGMEM= {"d "};
const prog_char h[] PROGMEM= {"h "};
const prog_char m[] PROGMEM= {"m "};
const prog_char s[] PROGMEM= {"s"};
//-----//

//----η συνάρτηση setup είναι υπεύθυνη για την αρχικοποίηση software & hardware
παραμέτρων πριν την εκτέλεση της loop----//

void setup()
{
  pinMode(FLASH_SLAVE_SELECT,OUTPUT); //θέτει το pin 7 ως έξοδο
  digitalWrite(FLASH_SLAVE_SELECT,HIGH); //προσωρινή απενεργοποίηση του AT45DB161D
για αποφυγή conflicts
  pinMode(WIFI_SLAVE_SELECT,OUTPUT); //θέτει το pin 10 ως έξοδο
  digitalWrite(WIFI_SLAVE_SELECT,HIGH); //προσωρινή απενεργοποίηση του g2100 για
αποφυγή conflicts
  WiServer.init(servePage); //αρχικοποίηση της ιστοσελίδας που θα διακομίσουμε
  dflash.init(FLASH_SLAVE_SELECT); //AT45DB161D ως slave
  digitalWrite(8, HIGH); //το pin 8 ανεβαίνει σε κατάσταση HIGH ώστε να μην
παράγει interrupt το g2100 και σταματάει την loop()
  sensors.begin(); //αρχικοποίηση του DS18B20
}
//-----//

//---η συνάρτηση αυτή συνθέτει την κύρια ιστοσελίδα που θα διακομίσουμε---//

boolean servePage(char* URL) {

  if (strcmp(URL, "/") == 0){
    WiServer.print_P(header);
    WiServer.print_P(hr);
    WiServer.print_P(temperature_banner);
    WiServer.print_P(hr);
    WiServer.print_P(text_now);
    WiServer.print(temperature_now=get_temperature()); //θερμοκρασία τώρα
    WiServer.print_P(celsius);
    WiServer.print_P(p_end);
    WiServer.print_P(text_min);
    WiServer.print(temperature_min); //ελάχιστη θερμοκρασία
    WiServer.print_P(celsius);
    WiServer.print_P(p_end);
    WiServer.print_P(text_med);
    WiServer.print(average_temperature); //μέση θερμοκρασία
    WiServer.print_P(celsius);
    WiServer.print_P(p_end);
    WiServer.print_P(text_max);
    WiServer.print(max_temperature); //μέγιστη θερμοκρασία
    WiServer.print_P(celsius);
    WiServer.print_P(hr);
    WiServer.print_P(humidity_banner);
    WiServer.print_P(hr);
    WiServer.print_P(text_now);
    WiServer.print(humidity_now=get_humidity(temperature_now)); //υγρασία τώρα
    WiServer.print_P(percentage);
    WiServer.print_P(p_end);
    WiServer.print_P(text_min);
    WiServer.print(humidity_min); //ελάχιστη υγρασία
    WiServer.print_P(percentage);
    WiServer.print_P(p_end);
    WiServer.print_P(text_med);
    WiServer.print(average_humidity); //μέση υγρασία τώρα
    WiServer.print_P(percentage);
    WiServer.print_P(p_end);
    WiServer.print_P(text_max);
    WiServer.print(max_humidity); //μέγιστη υγρασία τώρα
    WiServer.print_P(percentage);
  }
}

```

```

WiServer.print_P(p_end);
WiServer.print_P(hr);
WiServer.print_P(home_pir);
WiServer.print_P(pir_table);
WiServer.print_P(tr);
WiServer.print_P(pir_msg_2);
WiServer.print_P(td);
WiServer.print(pir); //πόσες κινήσεις είχαμε
WiServer.print_P(td_end);
WiServer.print_P(tr_end);
WiServer.print_P(tr);
WiServer.print_P(last);
if(pir>0)
{
    uptime(timer-pir_timer); //η τελευταία κίνηση πότε ήταν
    WiServer.print_P(td);
    WiServer.print(hours); //ώρες μετά την τελευταία κίνηση
    WiServer.print_P(h);
    WiServer.print(minutes); //λεπτά μετά την τελευταία κίνηση
    WiServer.print_P(m);
    WiServer.print(seconds); //δευτερόλεπτα μετά την τελευταία κίνηση
    WiServer.print_P(s);
    WiServer.print_P(td_end);
}
else
{
    WiServer.print_P(td);
    WiServer.print("--");
    WiServer.print_P(td_end);}
    WiServer.print_P(tr_end);
    WiServer.print_P(tr);
    WiServer.print_P(pir_now);
    if(digitalRead(5)==1) //κίνηση σε εξέλιξη
    WiServer.print_P(yes);
    if(digitalRead(5)==0) //απουσία κίνησης
    WiServer.print_P(no);
    WiServer.print_P(tr_end);
    WiServer.print_P(table_end);
    WiServer.print_P(home_nav); //σελίδα καταγραφής 24ωρου
    WiServer.print_P(footer);
    uptime(timer); //uptime συστήματος
    WiServer.print(days); //συνολικές μέρες λειτουργίας
    WiServer.print_P(d);
    WiServer.print(hours); //συνολικές ώρες λειτουργίας
    WiServer.print_P(h);
    WiServer.print(minutes); //συνολικά λεπτά λειτουργίας
    WiServer.print_P(m);
    WiServer.print(seconds); //συνολικά δευτερόλεπτα λειτουργίας
    WiServer.print_P(s);
    WiServer.print_P(footer_end);

    return true;
}
//-----τέλος βασικής ιστοσελίδας-----//

//---όταν ο web browser κάνει αίτημα για συγκεκριμένα αρχεία τα αποστέλλουμε με τον
παρακάτω κώδικα---//

if (strcmp(URL, "/styles.css") == 0){ //αποστολή του stylesheet styles.css
    WiServer.print_P(css); //δεδομένα στην PROGMEM

    return true;
}

if (strcmp(URL, "/gradient.jpg") == 0){ //αποστολή εικόνας gradient.jpg
    WiServer.write_P((char*)gradient,912); //δεδομένα στην PROGMEM

```

```

    return true;
}
//-----//

//---εδώ συνθέτουμε την ιστοσελίδα καταγραφών 24ωρου---//

if (strcmp(URL, "/log") == 0){

    float v,max_temp=0.0,min_temp=99.9,max_hum=0.0,min_hum=99.9; //προσωρινές
μεταβλητές για τις τιμές μέγιστων και ελάχιστων στους πίνακες
    int max_pos=0,min_pos=0,max_pos_2=0,min_pos_2=0; //προσωρινές μεταβλητές για
την θέση μέγιστων και ελάχιστων στους πίνακες
    byte array_byte[4]; //πίνακας για την αποθήκευση των 4 byte ενός float

    dflash.Page_To_Buffer(0,1); //μεταφορά δεδομένων από την page 0 στον buffer 1
    dflash.Page_To_Buffer(1,2); //μεταφορά δεδομένων από την page 1 στον buffer 2
    WiServer.print_P(header);

//---πίνακας καταγραφής θερμοκρασίας---//

    WiServer.print_P(log_caption);
    WiServer.print_P(log_table);
    WiServer.print_P(tr);
    WiServer.print_P(log_caption_1);
    WiServer.print_P(tr_end);
    WiServer.print_P(tr);
//---1η γραμμή πίνακα τιμών θερμοκρασίας---//
    for (int j=0,i=0;i< 33; i++) //επεξεργασία 32 byte ή 8 float
    {
        if((i%4==0)&&(i!=0)) //έλεγχος αν είμαστε στο 4ο byte
        {
            WiServer.print_P(td);
            v=FloatFromByteArray(array_byte); //δόμηση τιμής από 4 byte σ' ένα float
            WiServer.print(v);

            if(v>max_temp){ //εύρεση μέγιστης τιμής
                max_temp=v;
                max_pos=i/4;} //θέση μέγιστης τιμής
            if(v<min_temp){ //εύρεση ελάχιστης τιμής
                min_temp=v;
                min_pos=i/4;} //θέση ελάχιστης τιμής
            WiServer.print_P(td_end);
            j=0; //μόλις έγινε ανάγνωση του 4ου byte
        }
        array_byte[j]=dflash.Buffer_Read_Byte(1,i); //ανάγνωση 1 byte από τον buffer 1
        j++; //θέση επόμενου byte
    }
    WiServer.print_P(tr_end);
//---2η γραμμή πίνακα τιμών θερμοκρασίας---//
    for (int j=0,i=36;i<65; i++) //επεξεργασία 32 byte ή 8 float offset 32 byte
    {
        if((i%4==0)&&(i!=0)) //έλεγχος αν είμαστε στο 4ο byte
        {
            WiServer.print_P(td);
            v=FloatFromByteArray(array_byte); //δόμηση τιμής από 4 byte σ' ένα float
            WiServer.print(v);

            if(v>max_temp){ //εύρεση μέγιστης τιμής
                max_temp=v;
                max_pos=i/4;} //θέση μέγιστης τιμής
            if(v<min_temp){ //εύρεση ελάχιστης τιμής
                min_temp=v;
                min_pos=i/4;} //θέση ελάχιστης τιμής
            WiServer.print_P(td_end);
            j=0; //μόλις έγινε ανάγνωση του 4ου byte
        }
        array_byte[j]=dflash.Buffer_Read_Byte(1,i); //ανάγνωση 1 byte από τον buffer 1
    }
}

```

```

        j++; //θέση επόμενου byte
    }
    WiServer.print_P(tr_end);
//---3η γραμμή πίνακα τιμών θερμοκρασίας---//
    WiServer.print_P(tr);

    for (int j=0,i=68;i< 97; i++) //επεξεργασία 32 byte ή 8 float offset 32 byte
    {
        if((i%4==0)&&(i!=0)) //έλεγχος αν είμαστε στο 4ο byte
        {
            WiServer.print_P(td);
            v=FloatFromByteArray(array_byte); //δόμηση τιμής από 4 byte σ' ένα float
            WiServer.print(v);

            if(v>max_temp){ //εύρεση μέγιστης τιμής
                max_temp=v;
                max_pos=i/4;} //θέση μέγιστης τιμής
            if(v<min_temp){ //εύρεση ελάχιστης τιμής
                min_temp=v;
                min_pos=i/4;} //θέση ελάχιστης τιμής
            WiServer.print_P(td_end);
            j=0; //μόλις έγινε ανάγνωση του 4ου byte
        }
        array_byte[j]=df.flash.Buffer_Read_Byte(1,i); //ανάγνωση 1 byte από τον buffer 1
        j++; //θέση επόμενου byte
    }
    WiServer.print_P(tr_end);
    WiServer.print_P(tr);
    WiServer.print_P(log_caption_max);
    WiServer.print(max_temp); //μέγιστη τιμη θερμοκρασίας
    WiServer.print_P(celsius);
    WiServer.print_P(space);
    WiServer.print_P(log_caption_at);
    WiServer.print(max_pos-1); //τι ώρα
    WiServer.print_P(log_caption_end);
    WiServer.print_P(td_end);
    WiServer.print_P(tr_end);
    WiServer.print_P(tr);
    WiServer.print_P(log_caption_min);
    WiServer.print(min_temp); //ελάχιστη τιμη θερμοκρασίας
    WiServer.print_P(celsius);
    WiServer.print_P(space);
    WiServer.print_P(log_caption_at);
    WiServer.print(min_pos-1); //τι ώρα
    WiServer.print_P(log_caption_end);
    WiServer.print_P(td_end);
    WiServer.print_P(tr_end);
    WiServer.print_P(table_end);

//---πίνακας καταγραφής υγρασίας---//
    WiServer.print_P(log_table);
    WiServer.print_P(tr);
    WiServer.print_P(log_caption_2);
    WiServer.print_P(tr_end);
//---1η γραμμή πίνακα τιμών υγρασίας---//
    for (int j=0,i=0;i< 33; i++) //επεξεργασία 32 byte ή 8 float
    {
        if((i%4==0)&&(i!=0)) //έλεγχος αν είμαστε στο 4ο byte
        {
            WiServer.print_P(td);
            v=FloatFromByteArray(array_byte); //δόμηση τιμής από 4 byte σ' ένα float
            WiServer.print(v);

            if(v>max_hum){ //εύρεση μέγιστης τιμής
                max_hum=v;
                max_pos_2=i/4;} //θέση μέγιστης τιμής
            if(v<min_hum){ //εύρεση ελάχιστης τιμής
                min_hum=v;

```

```

        min_pos_2=i/4;} //θέση ελάχιστης τιμής
        WiServer.print_P(td_end);
        j=0; //μόλις έγινε ανάγνωση του 4ου byte
    }
    array_byte[j]=dflash.Buffer_Read_Byte(2,i); //ανάγνωση 1 byte από τον buffer 2
    j++; //θέση επόμενου byte
}
WiServer.print_P(tr_end);
//---2η γραμμή πίνακα τιμών υγρασίας---//
WiServer.print_P(tr);

for (int j=0,i=36;i<65; i++) //επεξεργασία 32 byte ή 8 float offset 32 byte
{
    if((i%4==0)&&(i!=0)) //έλεγχος αν είμαστε στο 4ο byte
    {
        WiServer.print_P(td);
        v=FloatFromArray(array_byte); //δόμηση τιμής από 4 byte σ' ένα float
        WiServer.print(v);

        if(v>max_hum){ //εύρεση μέγιστης τιμής
            max_hum=v;
            max_pos_2=i/4;} //θέση μέγιστης τιμής
        if(v<min_hum){ //εύρεση ελάχιστης τιμής
            min_hum=v;
            min_pos_2=i/4;} //θέση ελάχιστης τιμής
        WiServer.print_P(td_end);
        j=0; //μόλις έγινε ανάγνωση του 4ου byte
    }
    array_byte[j]=dflash.Buffer_Read_Byte(2,i); //ανάγνωση 1 byte από τον buffer 2
    j++; //θέση επόμενου byte
}
WiServer.print_P(tr_end);
//---3η γραμμή πίνακα τιμών υγρασίας---//
WiServer.print_P(tr);

for (int j=0,i=68;i< 97; i++) //επεξεργασία 32 byte ή 8 float offset 32 byte
{
    if((i%4==0)&&(i!=0)) //έλεγχος αν είμαστε στο 4ο byte
    {
        WiServer.print_P(td);
        v=FloatFromArray(array_byte); //δόμηση τιμής από 4 byte σ' ένα float
        WiServer.print(v);

        if(v>max_hum){ //εύρεση μέγιστης τιμής
            max_hum=v;
            max_pos_2=i/4;} //θέση μέγιστης τιμής
        if(v<min_hum){ //εύρεση ελάχιστης τιμής
            min_hum=v;
            min_pos_2=i/4;} //θέση ελάχιστης τιμής
        WiServer.print_P(td_end);
        j=0; //μόλις έγινε ανάγνωση του 4ου byte
    }
    array_byte[j]=dflash.Buffer_Read_Byte(2,i); //ανάγνωση 1 byte από τον buffer 2
    j++; //θέση επόμενου byte
}
WiServer.print_P(tr_end);
WiServer.print_P(tr);
WiServer.print_P(log_caption_max);
WiServer.print(max_hum); //μέγιστη τιμη υγρασίας
WiServer.print_P(percentage);
WiServer.print_P(space);
WiServer.print_P(log_caption_at);
WiServer.print(max_pos_2-1); //τι ώρα
WiServer.print_P(log_caption_end);
WiServer.print_P(td_end);
WiServer.print_P(tr_end);
WiServer.print_P(log_caption_min);
WiServer.print(min_hum); //ελάχιστη τιμη υγρασίας

```

```

WiServer.print_P(percentage);
WiServer.print_P(space);
WiServer.print_P(log_caption_at);
WiServer.print(min_pos_2-1); //τι ώρα
WiServer.print_P(log_caption_end);
WiServer.print_P(td_end);
WiServer.print_P(tr_end);
WiServer.print_P(table_end);

WiServer.print_P(log_nav); //πίσω στην αρχική σελίδα
WiServer.print_P(footer);
uptime(timer); //υπολογισμός uptime
WiServer.print(days); //συνολικές μέρες λειτουργίας
WiServer.print_P(d);
WiServer.print(hours); //συνολικές ώρες λειτουργίας
WiServer.print_P(h);
WiServer.print(minutes); //συνολικά λεπτά λειτουργίας
WiServer.print_P(m);
WiServer.print(seconds); //συνολικά δευτερόλεπτα λειτουργίας
WiServer.print_P(s);
WiServer.print_P(footer_end);

return true;
}
//-----τέλος ιστοσελίδας log-----//

return false; //αν επιστρέψει ψευδές -->URL not found --> 404 error---//
}
//-----τέλος servePage()-----//

//---η loop εκτελείται συνεχώς---//
void loop(){

static unsigned i,j=0,k=1,timer_2,timer_3,timer_4,timer_5; //αν δεν είναι τύπου
static θα χάσουν τις τιμές τους ανα iteration
float temperature_sum,average_temperature_loop,humidity_sum,average_humidity_loop;

WiServer.server_task(); //βασική συνάρτηση που ελέγχει αν ζητάμε κάποια ιστοσελίδα

if(!zg_get_conn_state()) //αν ανιχνευθεί αποσύνδεση --> low level reset στο g2100
{
zg_chip_reset();
WiServer.init(servePage);
}

//---υλοποίηση ενός software timer που ελέγχει αν η διαφορά της millis() με την
lastTick είναι 1000 msec ή 1 sec

if (millis() - lastTick >= 1000)
{
lastTick = millis(); //η millis() μας επιστρέφει τα millisecond από την αρχή
εκτέσης του κώδικα
timer++; //μετρήθηκε ενα sec
timer_2++; //βοηθητικοί timers
timer_3++; //<--|
timer_4++; //<--|
timer_5++; //<--|
}

//---εγγραφή μέγιστων & ελάχιστων τιμών από την EEPROM που καταγράφηκαν σε
προηγούμενες μετρήσεις --> παραμένουν μεταξύ των reset---//
if(timer==1)
{
float temp_t_max,temp_t_min,temp_h_max,temp_h_min;
eeprom_read(temp_t_max, eeprom_max_temp);
eeprom_read(temp_t_min, eeprom_min_temp);
eeprom_read(temp_h_max, eeprom_max_hum);
}

```

```

    eeprom_read(temp_h_min, eeprom_min_hum);
    max_temperature=temp_t_max;
    temperature_min=temp_t_min;
    max_humidity=temp_h_max;
    humidity_min=temp_h_min;
}

//---ελέγχουμε για κίνηση & κρατάμε 5 δείγματα αν 6 sec και βρίσκουμε επίσης
μέγιστη & ελάχιστη τιμή---//
if(timer_2==6) //ελέγχεται κάθε 6 sec
{
    temperature_loop=get_temperature(); //μέτρηση θερμοκρασίας
    humidity_loop=get_humidity(temperature_loop); //μέτρηση υγρασίας
    if(i>4)
        i=0;
    temperature_array[i]=temperature_loop; //πίνακας θερμορασίας
    humidity_array[i]=humidity_loop; //πίνακας υγρασίας
    i++;
    timer_2=0; //μηδενισμός timer
    if(temperature_loop>max_temperature) //εύρεση μέγιστης θερμοκρασίας
        max_temperature=temperature_loop;
    if(temperature_loop<temperature_min) //εύρεση ελάχιστης θερμοκρασίας
        temperature_min=temperature_loop;
    if(humidity_loop>max_humidity) //εύρεση μέγιστης υγρασίας
        max_humidity=humidity_loop;
    if(humidity_loop<humidity_min) //εύρεση ελάχιστης υγρασίας
        humidity_min=humidity_loop;
    if (digitalRead(5) == HIGH) //PIR στο pin 5 αν δώσει HIGH --> κίνηση
    {
        pir++; //σύνολο ανιχνευθέντων κινήσεων
        pir_timer=timer; //χρονική στιγμή που ανιχνεύτηκε κίνηση
    }
}

//---υπολογίζουμε την μέση τιμή θερμοκρασίας & υγρασίας με 5 δείγματα που τα
παίρνουμε ανά 6 sec---//
if(timer_3==30) //ελέγχεται κάθε μισό λεπτό
{
    temperature_sum=humidity_sum=0;
    for(int i=0;i<5;i++)
    {
        temperature_sum+=temperature_array[i];
        humidity_sum+=humidity_array[i];
    }
    average_temperature_loop=temperature_sum/5;
    average_humidity_loop=(humidity_sum/5);
    average_temperature=average_temperature_loop;
    average_humidity=average_humidity_loop;
    timer_3=0; //μηδενισμός timer
}

if(timer_4==60) //ελέγχεται κάθε λεπτό
{
    float temp_t_max,temp_t_min,temp_h_max,temp_h_min;
    temp_t_max=max_temperature;
    temp_t_min=temperature_min;
    temp_h_max=max_humidity;
    temp_h_min=humidity_min;
    eeprom_write(temp_t_max, eeprom_max_temp); //εγγραφή μέγιστων & ελάχιστων στην
EEPROM
    eeprom_write(temp_t_min, eeprom_min_temp); //<--|
    eeprom_write(temp_h_max, eeprom_max_hum); //<--|
    eeprom_write(temp_h_min, eeprom_min_hum); //<--|
    timer_4=0; //μηδενισμός timer
}

if(timer_5==3600) //ελέγχεται κάθε ώρα --> 3600 sec
{

```

```

    if(k)
    {
        float t_t=average_temperature;
        float t_h=average_humidity;
        dflash.Buffer_Write_Float(1,j,t_t); //μεταφορά της μέσης θερμοκρασίας στο
buffer
        dflash.Buffer_Write_Float(2,j,t_h); //μεταφορά της μέσης υγρασίας στο buffer
        dflash.Buffer_To_Page(1,0); //μεταφορά δεδομένων στην page 0
        dflash.Buffer_To_Page(2,1); //μεταφορά δεδομένων στην page 1
        j+=4; //μετακίνηση στην επόμενη θέσης μήμης 1 float --> 4 byte
        if(j==96)
        {
            k=0;
        }
    }
    timer_5=0; //μηδενισμός timer
}
}
//---τέλος της loop-----//

//---η get_temperature υπολογίζει την θερμοκρασία ---//
float get_temperature()
{
    sensors.waitForConversion(true); //αναμονή έως ο DS18B20 επιστρέψει την
θερμοκρασία
    sensors.requestTemperatures(); //request για θερμοκρασία στον δίαυλο 1 Wire
    return sensors.getTempCByIndex(0); //η θερμοκρασία της πρώτης συσκευής στον
δίαυλο
}

//---η get_humidity υπολογίζει την υγρασία με όρισμα την παρούσα θερμοκρασία ---//
float get_humidity(float temperature)
{
    float humidity,input;
    input=analogRead(5); //η τιμή που μας δίνει ο αισθητήρας HIH-4030
    humidity=(((input/1024)*5)+0.017)-0.958)/30.68)/(1.0546 -
(0.00216*(temperature))*1000; //υπολογισμός υγρασίας βάση επίδρασης θερμοκρασίας
    return humidity;
}

//---η uptime πέρνει όρισμα αριθμό δευτερολέπτων και υπολογίζει τις days hours
minutes seconds
void uptime(unsigned long timer)
{
    days=timer/86400; //δευτερόλεπτα ανά μέρα
    hours=(timer%86400)/3600; //δευτερόλεπτα ανά ώρα
    minutes=((timer%86400)%3600)/60; //δευτερόλεπτα ανά λεπτό
    seconds=((timer%86400)%3600)%60; //δευτερόλεπτα υπόλοιπο
}

//---η FloatFromByteArray συνθέτει έναν float απο 4 byte
float FloatFromByteArray (const unsigned char * received_data)
{
    float f;
    memcpy (&f, received_data, sizeof (float)); //παίρνει 4 byte και αντιγράφει τα
δεδομένα τους ως ενιαίο κομμάτι μήμης μέσω της memcpy στο περιεχόμενο ενός δείκτη
float
    return f;
}

```



## Παράρτημα Β κώδικας html ιστοσελίδων

**Σημείωση:** ο κώδικας html περιέχει και τιμές μετρήσεων.

### Πρώτη ιστοσελίδα:

```
<html>
<head>
<meta content="text/html; charset=UTF-8" http-equiv="content-type">
<title>Ιστοσελίδα παρακολούθησης τηλεμετρικού υποσυστήματος</title>
<link rel="stylesheet" href="thesis_home_files/styles.css" type="text/css">
</head>
<body>
<div id="container">
<div id="header">
<h1>Τηλεμετρικό υποσύστημα</h1></div>
<div id="wrapper"><div id="content">
<hr>
<p></p><h4>Θερμοκρασία</h4><p></p>
<hr><p>Τρέχουσα: 24.31°C</p>
<p>Ελάχιστη: 22.00°C</p>
<p>Μέση: 24.31°C</p>
<p>Μέγιστη: 26.25°C</p>
<hr><p></p>
<h4>Σχετική υγρασία</h4><p></p>
<hr>
<p>Τρέχουσα: 42.93%</p>
<p>Ελάχιστη: 17.92%</p>
<p>Μέση: 42.45%</p>
<p>Μέγιστη: 55.67%</p>
<hr><p></p>
<h4>Ανίχνευση κίνησης</h4><p></p>
<hr>
<p><table border="0"><tbody>
<tr><td>Πλήθος ανιχνεύσεων κίνησης:</td><td>8</td></tr>
<tr><td>Η τελευταία πριν:</td><td>0h 0m 3s</td></tr>
<tr><td>Κίνηση σε εξέλιξη:</td><td>OXI</td>
</tr></tbody></table></p></div></div>
<div id="extra">
<p><strong><h3><p>Πλοήγηση</p></h3></strong></p>
<hr>
<p><a href="http://192.168.1.10/log">Καταγραφές 24ωρου</a></p></div>
<div id="footer"><p>
<b>Συνολικός χρόνος λειτουργίας: </b>1d 0h 50m 4s</p></div></div>
</body>
</html>
```



## Παράρτημα Γ το css style sheet των δύο ιστοσελίδων

### Css style sheet:

```
html,body{margin:0;padding:0}
body{font: 100% arial,sans-serif;text-align:center}
p{margin:0 10px 10px}
a{padding:5px; text-decoration:none; color:#000000;}
div#header{border:1px solid gray;background-color:#CBE9F2;background-
image:url('gradient.jpg');background-repeat:repeat-x;}
div#header h1{height:80px;line-height:80px;margin:0;padding-left:10px;}
div#container{text-align:left}
div#content p{line-height:1.4}
div#navigation{background:#F6F0E0;}
div#navigation ul{margin:15px 0; padding:0; list-style-type:none;}
div#navigation li{margin-bottom:5px;}
div#extra{background:#F1E1B5;}
div#footer{background:#D0D0D0;}
div#footer p{margin:0;padding:5px 10px}
div#container{width:700px;margin:0 auto}
div#wrapper{float:left;width:100%}
div#content{margin-left: 300px}
div#navigation{float:left;width:150px;margin-left:-550px}
div#extra{border:1px solid gray;float:left;width:180px;margin-left:-700px}
div#footer{border:1px solid gray;clear:left;width:100%}
```

# Εφαρμογές ασύρματης τηλεμετρίας στην αναπτυξιακή πλατφόρμα Arduino

Ελευθεριάδης Μιχαήλ      ΑΜ: 982  
Επιβλέπων καθηγητής: Κορνάρος Γεώργιος

1

## Συνδυάζουμε:

3 αισθητήρες

- Θερμοκρασίας
- Υγρασίας
- Κίνησης

Αναπτυξιακές πλακέτες

- Arduino
- WiShield

2

## Με σκοπό:

### Τηλεμετρικό υποσύστημα

- Εκτελεί μετρήσεις & υπολογισμούς
  - Καταγράφει
  - Διακομίζει ιστοσελίδες

3

## Αρχικά θα μελετήσουμε θέματα :

- Τηλεμετρία
- Μικροελεγκτές
- Αισθητήρες
- Ασύρματα πρωτόκολλα

4

# Τηλεμετρία

- Τι είναι
- Πως λειτουργεί
- Οφέλη

5

# Μικροελεγκτές

## Τι είναι;

- Τι προσφέρουν;
  - Ταξινόμηση
  - Βασική δομή
- Προγραμματισμός

6

# Αισθητήρες

## Είδη:

- Άμεσος
- Έμμεσος

## Ταξινόμηση:

- Παθητικός
  - Ενεργητικός
  - Απόλυτος
  - Σχετικός
- Ιδιαίτερων χαρακτηριστικών & ιδιοτήτων

7

# Θερμοκρασία, υγρασία & αισθητήρες

## Τι είναι:

- Θερμοκρασία
  - Υγρασία
  - Απόλυτη
  - Σχετική
- Κλίμακες μέτρησης
  - Είδη

8

## Θερμοκρασία, υγρασία & αισθητήρες (2)

- Αισθητήρες θερμοκρασίας
  - Επαφής
  - Εξ αποστάσεως
- Αισθητήρες υγρασίας
- Δυσκολίες στις μέτρησεις

9

## Σφάλμα μέτρησης και αισθητήρες

Κατά τις μετρήσεις σημαντικό να έχουμε υπόψην :

- Σφάλμα μέτρησης
  - Αβεβαιότητα
- Εμφάνιση τυχαίων & στατιστικών σφαλμάτων
  - Ενίσχυση επαναληψιμότητας

10



# Ασύρματο πρωτόκολλο 802.11b

- Επέκταση του 802.11
- Ad hoc ή Infrastructure
  - Στα 2.4 GHz
- Απόδοση έως 5.5 Mbps
- 14 επικαλυπτόμενα κανάλια των 22MHz
- Αποφυγή συγκρούσεων με CSMA/CA
  - Ανοχή στον θόρυβο

11

# Arduino

- Αναπτυξιακή πλακέτα μικροελεγκτή
  - Open source hardware
    - Arduino IDE
- Ενσωματώνει απαραίτητα εξαρτήματα
  - Κόστος

12

## Arduino (2)

- 14 ψηφιακή pin
- 5 αναλογικά pin
- FTDI σε USB

### Επεξεργαστής ATmega 328P:

- 16 MHz
- 32 KB Flash
- 2 KB SRAM
- 1 KB EEPROM

•Προστασία από υψηλή τάση & λανθασμένο προγραμματισμό

13

## WiShield

- Ασύρματο πρωτόκολλο 802.11b
- Όλα τα pin του Arduino προσβάσιμα & απολυτή συμβατότητα διαστάσεων
  - Ενσωματωμένη κεραία
- DataFlash με χωρητικότητα 16 Mbit (2MB)

14

## Dallas DS18B20

- 1-Wire interface που απαιτεί μόνο ένα pin
- Μοναδικός 64-Bit κωδικός στο on-board ROM
- -55 °C έως +125 °C (-67 °F έως 257 °F)
- Ρυθμιζόμενη ακρίβεια μέτρησης
  - Χρόνος μέτρησης 750ms max
  - «Παρασιτική τάση»
  - Λειτουργία συναγερμού

15

## Honeywell HIH-4030

- Γραμμική απόκριση τάσης εξόδου
  - Αυξημένη ακρίβεια 3,5%
- Σταθερή επίδοση , μηδαμινή ανάγκη για αναρρύθμιση
- Ανθεκτικός σε διάφορα χημικά

16

## PIR Parallax 555 – 280227

- Εύρος ανίχνευσης μέχρι 6 μέτρα
- Έξοδος ενός bit υψηλή / χαμηλή διακριτής τιμής
- Jumper μονής ή συνεχόμενης εξόδου σε bit
- 3-pin SIP

17

## Υλοποίηση

- Υλοποίηση ενός software timer
- Τι εκτελείται σε καθε χρονικό διαστημα:
- Κάθε 6 δευτερόλεπτα
  - Κάθε 30 δευτερόλεπτα
  - Κάθε 60 δευτερόλεπτα
  - Κάθε μια ώρα
- Δημιουργία δυο ιστοσελίδων δυναμικού περιεχομένου

18

## Υλοποίηση (2)

Βασική δομή κώδικα C++ οι συναρτήσεις:

- setup
- loop
- servePage

19

## Υλοποίηση (3)

setup:

Ρύθμιση Software & hardware παραμέτρων

loop:

- Συνεχής εκτέλεση
- Μετρήσεις & καταγραφές
- Έλεγχος αν επιθυμούμε ιστοσελίδα

20

## Υλοποίηση (4)

servePage:

- Σύνθεση ιστοσελίδων
- Στατικότητα Html → `<b>κείμενο</b>`
- Μεθοδολογία δυναμικών ιστοσελίδων  
→ `<b>κείμενο μετρήσεις κείμενο</b>`

21

## Υλοποίηση (5)

Οι ιστοσελίδες της υλοποίησης

- Στην πρώτη  
Βασικές μετρήσεις
- Στην δεύτερη  
Καταγραφές 24 ωρου  
Το κύκλωμα της υλοποίησης:
  - Ρύθμιση και εγκατάσταση

22

## Συμπεράσματα

- Μελέτη περιβαλλοντικών συνθηκών
  - Ιδιαίτερα μικρό μέγεθος
- Χαμηλές απαιτήσεις σε ενέργεια και τεχνογνωσία
- Αρκετά χαμηλό κόστος

23

## Μελλοντικές επεκτάσεις

- Ομάδα ίδιων αισθητήρων, η ακρίβεια και η περιοχή κάλυψης θα αυξηθούν αισθητά
- Η μετατροπή του σε σταθμό πρόβλεψης καιρού, με μόνη αλλαγή του PIR αισθητήρα, με έναν συμβατό αισθητήρα ατμοσφαιρικής πίεσης
- Schematics του Arduino αλλά και της WiShield είναι διαθέσιμα υπό την άδεια Creative Commons, τροποποιούμε ώστε σμίκρυνση ή επιπλέον hardware

24

## Βιβλιογραφία

- [1] Frank Carden, Robert Henry, Russ Jedlicka, Telemetry Systems Engineering, Artech House Telecommunications Library 2002
- [2] Yuan Taur, Tak H. Ning, Fundamentals of Modern VLSI Devices, Cambridge University Press 2009
- [3] John Crisp, Introduction to Microprocessors and Microcontrollers, Elsevier 2004
- [4] Dale Wheat, Arduino Internals, Apress 2011
- [5] Jacob Fraden, Handbook of Modern Sensors, Springer 2010
- [6] Robert P. Benedict, Fundamentals of Temperature, Pressure and Flow Measurements, John Wiley & Sons 1984
- [7] John M. Castillo, Relative Humidity: Sensors, Management and Environmental Effects, Nova Science 2011
- [8] William Trochim, James P. Donnelly, The Research Methods Knowledge Base, Atomic Dog 2007
- [9] Maxim DS18B20, <http://datasheets.maxim-ic.com/en/ds/DS18B20.pdf>
- [10] Honeywell HIH-4030, [http://sensing.honeywell.com/index.php?ci\\_id=51625&la\\_id=1](http://sensing.honeywell.com/index.php?ci_id=51625&la_id=1)
- [11] Parallax PIR 555-280227, <http://www.parallax.com/Portals/0/Downloads/docs/prod/sens/555-28027-PIRSensor-v2.1.pdf>
- [12] Andrew S. Tanenbaum, Computer Networks, Prentice Hall 2010
- [13] WiShield, [http://asynclabs.com/wiki/index.php?title=WiShield\\_2.0](http://asynclabs.com/wiki/index.php?title=WiShield_2.0)

25

Σας ευχαριστω πολύ!

Ερωτήσεις;

26