



ΤΕΧΝΟΛΟΓΙΚΟ  
ΕΚΠΑΙΔΕΥΤΙΚΟ  
ΙΔΡΥΜΑ

ΤΕΙ ΗΠΕΙΡΟΥ

**ΣΧΟΛΗ: ΤΕΧΝΟΛΟΓΙΚΩΝ ΕΦΑΡΜΟΓΩΝ**  
**ΤΜΗΜΑ: ΜΗΧΑΝΙΚΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ Τ.Ε.**

## **ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ**

**ΘΕΜΑ:**

**«Δημιουργία ενός μετεωρολογικού σταθμού με χρήση μικροελεγκτή και εμφάνιση δεδομένων σε σελίδα.»**

---

**ΟΝΟΜΑΤΕΠΩΝΥΜΟ: Δημάκη Αθηνά**

**ΕΠΙΒΛΕΠΩΝ ΚΑΘΗΓΗΤΗΣ: Τζάλλας Αλέξανδρος**

**ΑΡΤΑ, ΦΕΒΡΟΥΑΡΙΟΣ 2017**

ΔΗΜΙΟΥΡΓΙΑ ΕΝΟΣ ΜΕΤΕΩΡΟΛΟΓΙΚΟΥ ΣΤΑΘΜΟΥ ΚΑΙ ΕΜΦΑΝΙΣΗ  
ΑΠΟΤΕΛΕΣΜΑΤΩΝ ΣΕ ΣΕΛΙΔΑ

ΔΗΜΑΚΗ ΑΘΗΝΑ

Α.Μ.: 10035

### Ευχαριστίες

Θα ήθελα να ευχαριστήσω θερμά τον καθηγητή μου κ. Αλέξανδρο Τζάλλα, για την ενθάρρυνση, την κατανόηση και την άψογη συνεργασία που είχαμε όλο το διάστημα της πτυχιακής μου εργασίας, αλλά και για τον σεβασμό και την ανθρώπινη κατανόηση προς το πρόσωπό μου. Ακόμη ήθελα να ευχαριστήσω και όλους τους καθηγητές που συνάντησα στην φοίτησή μου. Το τελευταίο, μα μεγάλο ευχαριστώ είναι για τους προστάτες και βοηθούς μου, την οικογένειά μου για την ανεκτίμητη παρουσία τους δίπλα μου και στους φίλους μου που ήταν δίπλα μου στην προσπάθεια αυτή.

Αφιερώνεται στην Προστάτιδά μου και Βοηθό μου.

***ΔΗΛΩΣΗ ΠΝΕΥΜΑΤΙΚΗΣ ΙΔΙΟΚΤΗΣΙΑΣ***

Η παρούσα εργασία αποτελεί προϊόν αποκλειστικά δικής μου προσπάθειας. Όλες οι πηγές που χρησιμοποιήθηκαν περιλαμβάνονται στη βιβλιογραφία και γίνεται ρητή αναφορά σε αυτές μέσα στο κείμενο όπου έχουν χρησιμοποιηθεί.

ΥΠΟΓΡΑΦΗ

**ΠΕΡΙΕΧΟΜΕΝΑ:**

**Περιεχόμενα**

Ευχαριστίες.....	1
<i>ΔΗΛΩΣΗ ΠΝΕΥΜΑΤΙΚΗΣ ΙΔΙΟΚΤΗΣΙΑΣ</i> .....	3
ΠΕΡΙΛΗΨΗ.....	10
ABSTRACT .....	11
1 ΕΙΣΑΓΩΓΗ .....	12
2 ΜΕΤΕΩΡΟΛΟΓΙΑ .....	13
2.1 Εισαγωγή στην μετεωρολογία.....	13
2.2 Ιστορική αναδρομή μετεωρολογίας .....	14
2.3 Εθνική Μετεωρολογική Υπηρεσία.....	17
2.4 Εθνικό Αστεροσκοπείο Αθηνών .....	18
2.5 Ινστιτούτο Ερευνών Περιβάλλοντος και Βιώσιμης Ανάπτυξης .....	20
3 ΜΕΤΕΩΡΟΛΟΓΙΚΑ ΦΑΙΝΟΜΕΝΑ.....	23
3.1 Θερμοκρασία.....	23
3.2 Υγρασία.....	23
3.3 Ατμοσφαιρική Πίεση.....	24
3.4 Βαρομετρική Πίεση.....	24
4 ΌΡΓΑΝΑ ΜΕΤΡΗΣΗΣ.....	26
4.1 Θερμοκρασία.....	26
4.2 Υγρασία.....	27
4.3 Βαρομετρική Πίεση.....	28
4.4 Εύχρηστα όργανα στην καθημερινότητα του ανθρώπου .....	29
5 ΜΕΤΕΩΡΟΛΟΓΙΚΟΣ ΣΤΑΘΜΟΣ.....	34
5.1 Ορισμός Μετεωρολογικού Σταθμού .....	34
5.2 Μετεωρολογικοί Σταθμοί στην Ελλάδα.....	34
5.3 Μετεωρολογικοί Σταθμοί στον κόσμο .....	39
6 ΕΝΣΩΜΑΤΩΜΕΝΑ ΣΥΣΤΗΜΑΤΑ.....	42
6.1 Ορισμός Ενσωματωμένων Συστημάτων .....	42
6.2 Χαρακτηριστικά Ενσωματωμένων Συστημάτων .....	43
6.3 Μικροεπεξεργαστές.....	44
6.4 Μικροελεγκτές .....	45
6.4.1 Γενικά .....	45

6.4.2	Κατηγορίες Μικροελεγκτών.....	47
6.4.3	Μικροελεγκτές 8,16 και 32-bit.....	47
6.4.4	Μικροελεγκτές Εσωτερικής και Εξωτερικής Μνήμης .....	48
6.4.5	Αρχιτεκτονική Von Neumann και Harvard.....	48
6.4.6	Αρχιτεκτονική CISC και RISC.....	49
6.4.7	Οι Μικροελεγκτές AVR.....	50
6.4.8	Τα χαρακτηριστικά των AVR της Atmel.....	51
6.4.9	Το Hardware ενός AVR μικροελεγκτή .....	51
6.4.10	Ο Πυρήνας του Μικροελεγκτή AVR .....	52
6.4.11	Περιφερειακά του μικροελεγκτή AVR .....	53
6.5	Σύγκριση μικροελεγκτή-μικροεπεξεργαστή .....	55
7	ARDUINO .....	57
7.1	Η Ιστορία του Arduino.....	57
7.2	Τι είναι το arduino.....	57
7.3	Η πλατφόρμα του .....	58
7.4	Το λογισμικό του.....	59
7.5	Πλεονεκτήματα του arduino .....	60
7.6	Arduino Mega on Mega2560 .....	61
7.7	Arduino Ethernet Shield.....	69
8	ΑΙΣΘΗΤΗΡΕΣ .....	73
8.1	Ορισμός αισθητήρων.....	73
8.2	Κατηγορίες αισθητήρων.....	73
8.3	Αισθητήρες της εργασίας .....	74
8.4	Αισθητήρας DHT11 .....	74
8.5	Αισθητήρας DS18B20 και DS18B20-Waterproof .....	76
8.6	Αισθητήρας BMP180.....	77
8.7	Αισθητήρας MPL115A2 .....	78
9	ΠΕΡΙΦΕΡΕΙΑΚΟΣ ΕΞΟΠΛΙΣΜΟΣ .....	79
9.1	Το εξωτερικό ρολόι Tiny Real_Time_Clock_DS1307 .....	79
9.2	Micro_SD Memory Card 4GB .....	80
10	ΤΟ SOFTWARE ΤΗΣ ΕΡΓΑΣΙΑΣ.....	81
10.1	HTML .....	81
10.2	XML.....	81
10.3	CSS.....	81

10.4	AJAX.....	82
10.5	PHP .....	83
10.6	SQL .....	83
10.7	JAVA.....	84
10.8	NOTEPAD++.....	84
10.9	XAMPP .....	85
11	ΤΟ HARDWARE ΤΗΣ ΕΡΓΑΣΙΑΣ .....	86
11.1	Το απαιτούμενο υλικό.....	86
11.2	Το κοστολόγιο του απαιτούμενου υλικού.....	91
12	ΚΑΤΑΣΚΕΥΗ /ΥΛΟΠΟΙΗΣΗ .....	92
12.1	Εγκατάσταση λογισμικού.....	92
12.2	Κούμπωμα των 2 arduino πλακετών.....	96
12.3	Συνδεσμολογία της πλακέτας με τον αισθητήρα DHT11 .....	96
12.4	Συνδεσμολογία της πλακέτας με τον αισθητήρα DS18B20.....	100
12.5	Συνδεσμολογία της πλακέτας με τον αισθητήρα DS18B20-Waterproof.....	106
12.6	Συνδεσμολογία της πλακέτας με τον αισθητήρα BMP180 .....	107
12.7	Συνδεσμολογία της πλακέτας με τον αισθητήρα MPL115A2 .....	112
12.8	Συνδεσμολογία της πλακέτας με το Tiny RTC_DS1307.....	116
12.9	Η τελική συνδεσμολογία της εργασίας .....	120
12.10	Ο προγραμματισμός του Web Server.....	130
12.11	Κατασκευή σελίδας.....	132
13	ΈΛΕΓΧΟΣ ΣΥΣΤΗΜΑΤΟΣ .....	135
14	ΣΥΜΠΕΡΑΣΜΑΤΑ.....	140
15	ΠΡΟΤΕΙΝΟΜΕΝΗ ΕΠΕΚΤΑΣΗ ΤΗΣ ΕΡΓΑΣΙΑΣ.....	141
	ΒΙΒΛΙΟΓΡΑΦΙΑ.....	143
	ΠΑΡΑΡΤΗΜΑ .....	146

**Περιεχόμενα Εικόνων:**

Εικόνα 1. EMY .....	17
Εικόνα 2. Ο Βαρόνος Γεώργιος Σίνας .....	18
Εικόνα 3. Θησείο .....	21
Εικόνα 4. Πεντέλη .....	22
Εικόνα 5. Υδραργυρικό Θερμόμετρο .....	26
Εικόνα 6. Ψυχρόμετρο August .....	27
Εικόνα 7. Υγρογράφος .....	28
Εικόνα 8. Βαρόμετρο .....	28
Εικόνα 9. Υδραργυρικό Θερμόμετρο .....	29
Εικόνα 10. Ο Avidsen .....	30
Εικόνα 11. Ο Avidsen εσωτερικά .....	30
Εικόνα 12. Ο TA-368 .....	31
Εικόνα 13. Ο TA-368 εσωτερικά .....	32
Εικόνα 14. Ο TA-298 .....	33
Εικόνα 15. Ο TA-298 εσωτερικά .....	33
Εικόνα 16. Η σελίδα της EMY .....	35
Εικόνα 17. Η σελίδα του meteo.gr .....	36
Εικόνα 18. A_Ενεργοί σταθμοί .....	36
Εικόνα 19. B_Ενεργοί σταθμοί .....	37
Εικόνα 20. Γ_Ενεργοί σταθμοί .....	37
Εικόνα 21. Απεικόνιση των σταθμών πάνω στον χάρτη της Ελλάδας .....	38
Εικόνα 22. Οι τιμές στην Πτολεμαΐδα_α .....	38
Εικόνα 23. Οι τιμές στην Πτολεμαΐδα_β .....	39
Εικόνα 24. Οι τιμές στην Πτολεμαΐδα_γ .....	39
Εικόνα 25. Οι επιλογές στις Χώρες στην σελίδα της EMY .....	40
Εικόνα 26. Οι επιλογές στις Χώρες στην σελίδα του meteo .....	40
Εικόνα 27. Ενδεικτικά οι τιμές από αεροδρόμια κάποιων χωρών .....	41
Εικόνα 28. Ο Atmega32 .....	50
Εικόνα 29. Η Inrea .....	57
Εικόνα 30. Διάφορες πλακέτες του Arduino .....	58
Εικόνα 31. Ο Arduino Mega2560 .....	58
Εικόνα 32. Η δομή ενός προγράμματος .....	60
Εικόνα 33. Ο Arduino Mega2560 .....	61
Εικόνα 34. Οι ακροδέκτες τροφοδοσίας .....	62
Εικόνα 35. Τα μέρη της πλακέτας .....	62
Εικόνα 36. Ο ATmega2560 .....	65
Εικόνα 37. Ο ATmega2560 .....	66
Εικόνα 38. Ο Arduino Ethernet Shield .....	69
Εικόνα 39. Winzet W5100 .....	69
Εικόνα 40. Το W5100 .....	71
Εικόνα 41. Ο αισθητήρας DHT11 .....	74
Εικόνα 42. Ο DHT11 εσωτερικά .....	75
Εικόνα 43. Οι ακροδέκτες του DHT11 .....	75
Εικόνα 44. Ο DS18B20 .....	76
Εικόνα 45. Ο DS18B20-Waterproof .....	76
Εικόνα 46. Ο BMP180 .....	77
Εικόνα 47. Το chip του bmp180 .....	77
Εικόνα 48. Ο MPL115A2 .....	78
Εικόνα 49. Το chip του MPL115A2 .....	78

Εικόνα 50. Ο DS1307.....	79
Εικόνα 51. Ο DS1307 εσωτερικά.....	79
Εικόνα 52. Η Micro_SD Card.....	80
Εικόνα 53. PHP.....	83
Εικόνα 54. SQL.....	83
Εικόνα 55. Μία function-συνάρτηση της JAVA.....	84
Εικόνα 56. NOTEPAD++.....	85
Εικόνα 57. XAMPP.....	85
Εικόνα 58. Επιλογή προγράμματος εγκατάστασης του Arduino.....	92
Εικόνα 59. Εγκατάσταση.....	92
Εικόνα 60. Εγκατάσταση των drivers.....	93
Εικόνα 61. Εγκατάσταση των βιβλιοθηκών.....	93
Εικόνα 62. Εγκατάσταση των βιβλιοθηκών.....	93
Εικόνα 63. Το περιβάλλον του IDE.....	94
Εικόνα 64. Επιλογή του Arduino Mega2560.....	94
Εικόνα 65. Επιλογή της σειριακής θύρας 4.....	95
Εικόνα 66. Τα εγκατεστημένα παραδείγματα.....	95
Εικόνα 67. Η σειριακή οθόνη.....	96
Εικόνα 68. Το κούμπωμα των 2 πλακετών.....	96
Εικόνα 69. Η συνδεσμολογία DHT11-Mega2560.....	98
Εικόνα 70. Κατά την κατασκευή DHT11.....	99
Εικόνα 71. Η συνδεσμολογία DS18B20-Mega2560.....	102
Εικόνα 72. Η συνδεσμολογία DS18B20-Mega2560.....	102
Εικόνα 73. Η κατασκευή του DS18B20.....	103
Εικόνα 74. Η κατασκευή του DS18B20.....	103
Εικόνα 75. Οι ακροδέκτες των DS18B20 DS18B20-Waterproof.....	106
Εικόνα 76. Κατά την κατασκευή του DS18B20-Waterproof.....	107
Εικόνα 77. Κατά την κατασκευή του DS18B20-Waterproof.....	107
Εικόνα 78. Η συνδεσμολογία BMP180-Mega2560.....	109
Εικόνα 79. Κατά την κατασκευή του BMP180.....	110
Εικόνα 80. Κατά την κατασκευή του BMP180.....	110
Εικόνα 81. Κατά την κατασκευή του BMP180.....	111
Εικόνα 82. Ο προγραμματισμός του BMP180.....	112
Εικόνα 83. Η συνδεσμολογία των MPL115A2-Mega2560.....	114
Εικόνα 84. Arduino UNO.....	115
Εικόνα 85. Κατά την κατασκευή του MPL115A2.....	115
Εικόνα 86. Η συνδεσμολογία των Tiny RTC_DS1307-Mega2560.....	118
Εικόνα 87. Κατά την κατασκευή των Tiny RTC_DS1307-Mega2560.....	119
Εικόνα 88. Κατά την κατασκευή των Tiny RTC_DS1307-Mega2560.....	119
Εικόνα 89. Η τελική συνδεσμολογία της εργασίας μου.....	122
Εικόνα 90. Κατά την κατασκευή της τελικής μου εργασίας.....	122
Εικόνα 91. Κατά την κατασκευή της τελικής μου εργασίας.....	123
Εικόνα 92. Η κατασκευή της τελικής μου εργασίας.....	123
Εικόνα 93. Η κατασκευή της τελικής μου εργασίας.....	124
Εικόνα 94. Ο έλεγχος του συστήματος στις 18/1/2017 το μεσημέρι.....	135
Εικόνα 95. Ο έλεγχος του συστήματος στις 18/1/2017 το μεσημέρι.....	136
Εικόνα 96. Ο έλεγχος του συστήματος στις 18/1/2017 το μεσημέρι.....	136
Εικόνα 97. Ο έλεγχος του συστήματος στις 18/1/2017 το βράδυ.....	137
Εικόνα 98. Ο έλεγχος του συστήματος με το φυσικό περιβάλλον στις 18/1/2017 το μεσημέρι.....	137
Εικόνα 99. Ο έλεγχος του συστήματος με το φυσικό περιβάλλον στις 18/1/2017 το μεσημέρι.....	138
Εικόνα 100. Ο έλεγχος του συστήματος με το φυσικό περιβάλλον στις 18/1/2017 το μεσημέρι.....	138
Εικόνα 101. Ο έλεγχος του συστήματος με το φυσικό περιβάλλον στις 18/1/2017 το βράδυ.....	139

Εικόνα 102. Ο έλεγχος του συστήματος με το φυσικό περιβάλλον στις 18/1/2017 το βράδυ..... 139

### Περιεχόμενα Πινάκων:

Πίνακας 1. Τα εργαλεία του IDE.....	59
Πίνακας 2. Η σύγκριση των Arduino Mega2560-Ethernet Shield.....	68
Πίνακας 3. Οι αισθητήρες της εργασίας .....	74
Πίνακας 4. Τα χαρακτηριστικά του DHT11.....	75
Πίνακας 5. Τα χαρακτηριστικά των DS18B20 και DS18B20-Waterproof.....	76
Πίνακας 6. Τα χαρακτηριστικά του BMP180.....	77
Πίνακας 7. Τα χαρακτηριστικά του MPL115A2. ....	78
Πίνακας 8. Τα χαρακτηριστικά του RTC_DS1307. ....	79
Πίνακας 9. Το κοστολόγιο των υλικών της κατασκευαστικής εργασίας .....	91
Πίνακας 10. Το κοστολόγιο της αγοράς των ψηφιακών θερμομέτρων. ....	91
Πίνακας 11. Συνδεσμολογία των ακροδεκτών του DHT11 με την πλακέτα. ....	98
Πίνακας 12. Συνδεσμολογία των ακροδεκτών του DS18B20 με την πλακέτα.....	102
Πίνακας 13. Συνδεσμολογία των ακροδεκτών του DS18B20-Waterproof με την πλακέτα μέσω των ακροδεκτών του DS18B20. ....	106
Πίνακας 14. Συνδεσμολογία των ακροδεκτών του BMP180 με τις πλακέτες. ....	109
Πίνακας 15. Συνδεσμολογία των ακροδεκτών του MPL115A2 με την πλακέτα. ....	114
Πίνακας 16. Συνδεσμολογία των ακροδεκτών του RTC_DS1307 με τις πλακέτες.....	118
Πίνακας 17. Καταγραφή των τιμών της θερμοκρασίας από όλα τα ψηφιακά όργανα και τους αισθητήρες. ....	140

### Περιεχόμενα Σχημάτων:

Σχήμα 1. Διαφορές μικροελεγκτή-μικροεπεξεργαστή.....	56
Σχήμα 2. Το σχηματικό του Arduino Mega2560.....	67
Σχήμα 3. Το σχηματικό του Ethernet Shield. ....	72
Σχήμα 4. Η δομή της CSS. ....	82
Σχήμα 5. Το ηλεκτρονικό σχέδιο DHT11 – Mega2560.....	97
Σχήμα 6. Το ηλεκτρικό σχέδιο των DS18B20 και DS18B20-Waterproof με τον Mega2560. ....	101
Σχήμα 7. Το ηλεκτρονικό σχέδιο των BMP180-Mega2560. ....	108
Σχήμα 8. Το ηλεκτρονικό σχέδιο των MPL115A2-Mega2560. ....	113
Σχήμα 9. Το ηλεκτρονικό σχέδιο των Tiny RTC_DS1307-Mega2560. ....	117
Σχήμα 10. Το ηλεκτρονικό σχέδιο της τελικής εργασίας μου. ....	121

### Περιεχόμενα Διαγραμμάτων:

Διάγραμμα 1. Χρονολογική Εξέλιξη της Μετεωρολογικής Υπηρεσίας στην Ελλάδα. ....	21
Διάγραμμα 2. Η δομή των Ενσωματωμένων Συστημάτων.....	42
Διάγραμμα 3. Κατηγορίες μικροελεγκτών. ....	47
Διάγραμμα 4. Οι αρχιτεκτονικές Von Neumann και Harvard. ....	49
Διάγραμμα 5. Διάγραμμα των pins του μικροελεγκτή AVR. ....	52
Διάγραμμα 6. Αρχιτεκτονική των μικροελεγκτών AVR. ....	52
Διάγραμμα 7. Αρχιτεκτονική AJAX.....	82

## ΠΕΡΙΛΗΨΗ

Στην παρούσα εργασία μελετώνται οι έννοιες σχετικά με τον καιρό, τα μετεωρολογικά φαινόμενα, την ιστορία της εθνικής μετεωρολογικής υπηρεσίας, αλλά και οι έννοιες του μικροεπεξεργαστή, του μικροελεγκτή, και πιο συγκεκριμένα οι μικροελεγκτές AVR της Atmel. Σκοπός της εργασίας είναι η δημιουργία ενός οικιακού μετεωρολογικού σταθμού με τη χρήση μιας πλακέτας Arduino και αισθητήρων μέτρησης της θερμοκρασίας, της υγρασίας και της ατμοσφαιρικής πίεσης. Καθώς και η εμφάνιση των τιμών αυτών σε ιστοσελίδα.

**Λέξεις κλειδιά:** καιρός, μικροελεγκτές, Arduino, αισθητήρες, ιστοσελίδα

### **ABSTRACT**

In this paper we study the concepts of weather, meteorological phenomena, the history of the National Weather Service, and the concepts of microprocessor, microcontroller, and more specifically the AVR microcontrollers of Atmel. The aim of this work is the creation of a home weather station using an Arduino board and temperature measuring sensors, humidity and atmospheric pressure and the appearance of these prices in website.

**Keywords:** weather, microcontrollers, Arduino, sensors, website

## 1 ΕΙΣΑΓΩΓΗ

Η παρούσα εργασία ασχολείται με ένα πολύ καθημερινό αλλά και ταυτόχρονα σημαντικό θέμα, τον καιρό. Σημαντικό καθώς από αυτόν εξαρτάται η «καλή μέρα» όταν ξυπνάμε το πρωί, τι ρούχα να φορέσουμε για βγούμε έξω, για το ταξίδι μας, για τους αγρότες πότε θα σπείρουν, για τους ναυτικούς εάν θα έχει «μπουνάτσα» και σε κάθε βήμα μας εξαρτόμαστε από αυτόν.

Έτσι, αποφάσισα να ασχοληθώ με αυτόν και να βρω έναν τρόπο μελέτης της θερμοκρασίας, της υγρασίας και της ατμοσφαιρικής πίεσης. Με αυτήν την αφορμή ανακάλυψα την αναπτυξιακή πλατφόρμα του Arduino και τους αισθητήρες DHT11, MPL115A2, BMP180, DS18B20 και DS18B20-Waterproof και προσπάθησα να δημιουργήσω έναν μετεωρολογικό σταθμό στο σπίτι μου, που θα μου δείχνει τα δεδομένα σε σελίδα στον υπολογιστή μου.

## 2 ΜΕΤΕΩΡΟΛΟΓΙΑ

Στο κεφάλαιο αυτό θα μελετήσουμε τις έννοιες του καιρού, της μετεωρολογίας καθώς και την εξέλιξη της μελέτης και διάγνωσής τους στο χρόνο. Θα ερευνήσουμε τους τομείς στους οποίους η συμβολή τους είναι πολύ σημαντική.

### 2.1 Εισαγωγή στην μετεωρολογία

Καιρός χαρακτηρίζεται η κατάσταση της ατμόσφαιρας πάνω από μια περιοχή και για σύντομη χρονική περίοδο, η οποία ορίζεται με διάφορα μετεωρολογικά φαινόμενα και στοιχεία, όπως είναι η θερμοκρασία, η υγρασία του αέρα, η ατμοσφαιρική πίεση, η συννεφιά, η βροχή, οι άνεμοι, η κατάσταση στη θάλασσα, η ορατότητα κτλ. Για να μελετήσουν τις διάφορες καιρικές καταστάσεις σημειώνουν τα διάφορα μετεωρολογικά στοιχεία των διαφόρων τόπων πάνω στο λεγόμενο «συνοπτικό χάρτη του καιρού». Τα στοιχεία αυτά τα συγκεντρώνουν από παρατηρήσεις ορισμένων ωρών. Οι μετεωρολογικοί σταθμοί της πατρίδας μας κάνουν παρατηρήσεις τρεις φορές την ημέρα: στις 8 το πρωί, στις 2 το μεσημέρι και στις 8 το βράδυ. (Αξιοτέλλης, 1993)

Η Μετεωρολογία αποτελεί κλάδο στις Φυσικές επιστήμες, που ασχολείται με την μελέτη της ατμόσφαιρας και των φαινομένων που συμβαίνουν μέσα στη μάζα της. Τα φαινόμενα που μελετά κατά κύριο λόγο η Μετεωρολογία είναι στο κατώτερο στρώμα της ατμοσφαιρας, που καλούνται ως τροπές, ως μεταβλητές αυτού του στρώματος, κατ' επέκταση το στρώμα αυτό ονομάστηκε τροπόσφαιρα. Οι αρχαίοι Έλληνες τα αποκαλούσαν μετέωρα που σημαίνει οτιδήποτε βρίσκεται στον ουρανό, γι' αυτό και η επιστήμη που τα μελετά ονομάστηκε Μετεωρολογία και τα φαινόμενα Μετεωρολογικά φαινόμενα. Η μετεωρολογία παράλληλα με την μελέτη αυτών των φαινομένων εξετάζει συγχρόνως τις επιδράσεις της ατμόσφαιρας πάνω στην επιφάνεια του εδάφους, στους ωκεανούς και στο ζωικό βασίλειο γενικά. Μελετά γενικά το κλίμα του κάθε τόπου ώστε να βρίσκεται σε θέση να προβλέπει τον καιρό.

Τα φαινόμενα που εξετάζει η Μετεωρολογία είναι: οι μεταβολές των ατμοσφαιρικών πιέσεων, η θερμοκρασία και η υγρασία του εδάφους και του αέρα, η κατεύθυνση και η δύναμη (ταχύτητα) των ανέμων, τα υδατώδη μετέωρα (βροχή, χιόνι, χαλάζι, πάχνη), η ηλιοφάνεια και η συννεφιά ενός τόπου, οι υπεριώδεις ακτίνες του ήλιου. Επίσης τα φαινόμενα που συμπεριλαμβάνουν τον ατμοσφαιρικό ηλεκτρισμό, δηλαδή οι καταγίδες, και εκείνα που οφείλονται στην ίδια την ατμόσφαιρα όπως διάθλαση, ανάκλαση κλπ. Στη μελέτη των φαινομένων, η Μετεωρολογία αναζητά τα αίτια που τα προκαλούν και τους παράγοντες που συμβάλουν, αλλά και τα όργανα που θα χρησιμοποιήσει ο μετεωρολόγος. Τελικός στόχος της μετεωρολογίας είναι η πρόγνωση του καιρού. Τα δελτία καιρού που μας ενημερώνουν από το ραδιόφωνο, την τηλεόραση και το διαδίκτυο συντάσσονται από την μετεωρολογική υπηρεσία. Τις πληροφορίες που χρειάζεται τις παίρνει από μετεωρολογικούς σταθμούς, οι οποίοι είναι σκόρπιοι σε διάφορα σημεία του εδάφους, είτε από μετεωρολογικά αερόστατα, είτε από πλοία και αεροπλάνα που είναι επανδρωμένα με πλήθος οργάνων (βαρόμετρα, υγρόμετρα, θερμόμετρα, ψυγρόμετρα, βροχόμετρα), που τα περισσότερα λειτουργούν αυτόματα. Σήμερα, με τη βοήθεια των μετεωρολογικών δορυφόρων δίνουν πιο αξιόπιστες πληροφορίες για τον καιρό και την εξέλιξή του στο άμεσο μέλλον. (Αξιοτέλλης, 1993)

## 2.2 Ιστορική αναδρομή μετεωρολογίας

Σύμφωνα με τις πληροφορίες που δίνει η εθνική μετεωρολογική υπηρεσία για τη σταδιακή εξέλιξη της μελέτης του καιρού και την αλληλεπίδραση με τους ανθρώπους αναφέρει τα εξής:

Όταν ο άνθρωπος εμφανίστηκε πάνω στη γη άρχισε να δέχεται στην καθημερινή του ζωή τις επιδράσεις των καιρικών φαινομένων. Μια ισχυρή καταιγίδα μπορούσε να προκαλέσει πλημμύρα, να καταστρέψει τη σοδειά ή να πνίξει ανθρώπους. Γι' αυτό και οι πρώτοι θεοί που λάτρευε ήταν θεοί που μπορούσαν να ελέγχουν τα βίαια καιρικά φαινόμενα.

Κατά την αρχαιότητα οι διάφοροι λαοί απέδιδαν τη δημιουργία των ατμοσφαιρικών και καιρικών φαινομένων στους θεούς. Η Ελληνική μυθολογία αποτελεί τον αδιάψευστο μάρτυρα σύμφωνα με τον οποίο στον Ελληνικό χώρο η δημιουργία τέτοιων φαινομένων αποδίδονταν στους θεούς, με κορυφαίο φυσικά το θεό Δία (Ζεός). Αξιοσημείωτες από την αρχαία Ελληνική μυθολογία είναι οι εκφράσεις “Σημεία των Καιρών” και οι “Αλκυονίδες ημέρες”. Ως Αλκυονίδες μέρες ονομάζονται από τους Έλληνες οι αίθριες ημέρες που επικρατούν από τη 15<sup>η</sup> Δεκεμβρίου έως και την 15<sup>η</sup> Φεβρουαρίου, τις οποίες ο Αριστοτέλης της χαρακτήρισε ως ημέρες ‘ευδίας’. Οι ημέρες αυτές έλαβαν το όνομά τους “αλκυόνια” από το μύθο της ελληνικής μυθολογίας με την Αλκυόνη, κόρη του Αιόλου που κυβερνούσε τους ανέμους. Η Αλκυόνη έπεσε σε σφάλμα και ο Δίας την μετέτρεψε σε πουλί για να την τιμωρήσει, επειδή όμως την λυπήθηκε διέταξε τον Αίολο να σταματάει για 14 μέρες τους ανέμους και να έχει καλοκαιρία ώστε να μπορεί να γεννήσει τα αυγά της και να μην σπάσουν από τους ανέμους του χειμώνα.

Τον 5<sup>ο</sup> π. Χ. αιώνα οι αρχαίοι Έλληνες φιλόσοφοι και αστρονόμοι μετά από συστηματική παρατήρηση των καιρικών φαινομένων διατύπωσαν αρχές για την ασφαλή εξαγωγή συμπερασμάτων σχετικά με τον καιρό. Προσπάθησαν μάλιστα να φτάσουν μέχρι τη στατιστική πρόγνωση του καιρού, κρατώντας στοιχεία για τα καιρικά φαινόμενα σε ένα ημερολόγιο που είναι γνωστό σαν ‘παράπηγμα’. Η σημαντική περίοδος της μετεωρολογίας ξεκινάει από την εποχή του Αριστοτέλη, καθώς ο ίδιος της έδωσε το κύρος που έπρεπε με τις θεωρίες και τις παρατηρήσεις του ίδιου και των μαθητών του στο σύγγραμμά του για την Μετεωρολογία, για αυτόν το λόγο ονομάστηκε Πατέρας της Μετεωρολογίας.

Μέχρι την εφεύρεση του θερμομέτρου (1503) και του βαρομέτρου (1643) καμία σημαντική πρόοδος δεν είχε γίνει στην επιστήμη της μετεωρολογίας. Η χρήση των οργάνων αυτών ήταν σταθμός γιατί για πρώτη φορά τα μετεωρολογικά στοιχεία εκφράστηκαν με αριθμητικά μεγέθη. Ακολούθησε και η εφεύρεση άλλων οργάνων όπως των βροχόμετρων, των υγρόμετρων και των ανεμόμετρων. Επίσης οι καιρικές παρατηρήσεις που κατέγραφαν οι καπετάνιοι των ιστιοφόρων πλοίων που άρχισαν πλέον μεγαλύτερα ταξίδια, έδωσαν ώθηση για διάφορες έρευνες τον 17<sup>ο</sup>, 18<sup>ο</sup> και 19<sup>ο</sup> αιώνα.

Η συγκέντρωση και η επεξεργασία των παρατηρήσεων αυτών από τα καράβια αλλά και από διάφορους σταθμούς στη στεριά, επέτρεψε τη συστηματική έρευνα των διαφόρων φαινομένων της ατμόσφαιρας και την προσπάθεια ερμηνείας των αιτίων που τα προκαλούν. Σταδιακά άρχισαν με την πάροδο του χρόνου να διατυπώνονται αρχές και νόμοι της Φυσικής Μετεωρολογίας.

Ο Γερμανός μετεωρολόγος Brandes το 1820 με μια πολύ σημαντική μελέτη προσπάθησε να βοηθήσει στην καλύτερη κατανόηση αλλά και απεικόνιση των καιρικών φαινομένων, δημιουργώντας τους πρώτους συνοπτικούς χάρτες καιρού, όπου γίνεται φανερό η ύπαρξη και η φύση των συστημάτων πίεσης (βαρομετρικά χαμηλά, βαρομετρικά ψηλά). Την εποχή εκείνη, λοιπόν γεννιέται η Συνοπτική μετεωρολογία, ενώ η περαιτέρω ανάπτυξή της έγινε χάρη σε ένα τυχαίο γεγονός που αφορούσε μια στρατιωτική επιχείρηση.

Κατά τη διάρκεια του Κριμαϊκού Πολέμου το Νοέμβριο του 1854 μια κακοκαιρία επηρέασε τη περιοχή του Εύξεινου Πόντου προκαλώντας μεγάλες ζημιές στον Αγγλικό και το Γαλλικό στόλο. Το γεγονός αυτό αποτέλεσε αφορμή για τον Γάλλο Υπουργό του Στρατού να προσεγγίσει το διευθυντή του Αστεροσκοπείου των Παρισίων La verrier και να του εκθέσει τις απορίες του όπως, εάν θα μπορούσε να έχει γίνει πρόβλεψη της κακοκαιρίας. Ξεκίνησε ο La verrier να συλλέγει παρατηρήσεις από διακόσιους και πλέον μετεωρολογικούς σταθμούς της Ευρώπης και διαπίστωσε ότι η μεγάλη ατμοσφαιρική διαταραχή που επηρέασε τον Εύξεινο Πόντο είχε προηγουμένως διατρέξει την Ευρώπη.

Το συμπέρασμα που εξήγαγε από τη συλλογή των δεδομένων ήταν ότι η πρόγνωση του καιρού εναπόκειται στη συχνή καταγραφή μετεωρολογικών παρατηρήσεων από πολλές περιοχές και στην άμεση αποστολή των συλλεγμένων στοιχείων στα ειδικά μετεωρολογικά κέντρα. Ο La verrier αναδεικνύεται ο θεμελιωτής των δικτύων μετεωρολογικών σταθμών και μετεωρολογικών υπηρεσιών. Μετά από αυτήν του την ανακάλυψη δημιουργείται στο Αστεροσκοπείο Παρισίων η Μετεωρολογική Υπηρεσία, που περιλάμβανε και ειδικό τμήμα προγνώσεων. Η πρόγνωση τώρα βασίζεται στις παρατηρήσεις που γίνονται στη Γαλλία και σε άλλες χώρες της Ευρώπης και αποστέλλονται μέσω του τηλεγράφου, του οποίου η εφεύρεση που έγινε το 1848 πρόσφερε μεγάλες υπηρεσίες στη μετεωρολογία.

Αποτέλεσε αφορμή να ιδρυθούν μετεωρολογικές υπηρεσίες και σε άλλες χώρες με πρώτες τις ΗΠΑ, την Αγγλία και την Ολλανδία. Πολύ σημαντική ήταν η συνεργασία μεταξύ τους, στον τρόπο εκτέλεσης, συγκέντρωσης και ανταλλαγής των μετεωρολογικών παρατηρήσεων για την καθημερινή σύνταξη των χαρτών καιρού, και για την ανάλυση και την πρόγνωση των καιρικών καταστάσεων, συζητήθηκαν και καθορίστηκαν στα διάφορα μετεωρολογικά συνέδρια. Γρήγορα διαπιστώθηκε ότι είναι απαραίτητη η διεθνής συνεργασία για τη σωστή πρόβλεψη του καιρού με αποτέλεσμα να συσταθεί το 1878 ο διεθνής Μετεωρολογικός Οργανισμός, από τον οποίο δημιουργήθηκε το 1950 ο Παγκόσμιος Μετεωρολογικός Οργανισμός (Π.Μ.Ο.).

Η σειρά της Ελλάδας έρχεται το 1839 οπότε και αρχίζουν να γίνονται οι πρώτες μετεωρολογικές παρατηρήσεις από το Αστεροσκοπείο Αθηνών ενώ από το 1890 αρχίζει να λειτουργεί ένα μικρό δίκτυο από 7 μετεωρολογικούς σταθμούς σε όλη τη χώρα. Στην αρχή η πρόγνωση του καιρού βάση συνοπτικών χαρτών γινόταν κατά προσέγγιση, εξαρτώμενη από την προσωπική πείρα και την κρίση των μετεωρολόγων. Τη βελτίωση της πρόγνωσης του καιρού επεδίωξαν να λύσουν οι μαθηματικοί και οι φυσικοί της εποχής οι οποίοι προσπαθούσαν να διερευνήσουν τη δυναμική των υφέσεων (βαρομετρικά χαμηλά) και άλλων διαταραχών καθώς και τη θερμοδυναμική λειτουργία της ατμόσφαιρας.

Ύστερα από πολλά χρόνια, τέλη του 19<sup>ου</sup> αιώνα στο πεδίο των θεωρητικών ερευνών γίνεται η σημαντική διατύπωση της θεωρίας της γενικής κυκλοφορίας της ατμόσφαιρας από τον V.Bjerknes που αποτελεί τον ακρογωνιαίο λίθο της θεωρητικής μετεωρολογίας. Παράλληλα οι μετεωρολόγοι εκφράζουν την ανάγκη να μελετηθεί η ατμόσφαιρα κατά ύψος και γίνονται προσπάθειες με αερόστατα και μετεωρολογικούς αετούς.

Στις αρχές του 20<sup>ου</sup> αιώνα σημειώνεται η μεγάλη ανάπτυξη της μετεωρολογίας με την εφεύρεση του ασύρματου και την πρόοδο της φυσικής, των μαθηματικών και της χημείας. Κατά τη διάρκεια του 1<sup>ου</sup> Παγκόσμιου Πολέμου η χρησιμοποίηση του αεροπλάνου και των ασφύξιογόνων αερίων, ανάγκασε τους εμπόλεμους να αναδιοργανώσουν τις Μετεωρολογικές Υπηρεσίες και να πυκνώσουν τις παρατηρήσεις που αφορούσαν, κυρίως, τους ανώτερους ανέμους. Η ύπαρξη του πλούσιου αυτού υλικού και η επεξεργασία του αλλά ιδιαίτερα το υλικό των μετεωρολογικών δικτύων των Σκανδιναβικών χωρών, οι οποίες ήταν εκτός πολέμου, προέκυψαν καινούργιες θεωρίες για τις ατμοσφαιρικές διαταραχές, μεταξύ των οποίων εξέχουσα θέση κατέχουν οι θεωρίες των V και J.Bjerknes, Solberg και του Bergeron, για τις μετωπικές επιφάνειες και τη δημιουργία και εξέλιξη των υφέσεων.

Στην περίοδο του Μεσοπολέμου διατυπώνονται θεωρίες για το σχηματισμό των αέριων μαζών, το σχηματισμό των νεφών, τα προβλήματα της γενικής κυκλοφορίας της ατμόσφαιρας, τη θερμοδυναμική της ατμόσφαιρας και των κινηματικών ιδιοτήτων των μετώπων και των υφέσεων. Μεγάλη ανακάλυψη είναι η χρήση της ραδιοβολίδας που επέτρεψε να μελετήσουμε την ατμόσφαιρα κατά ύψος. Σημειώθηκε ως ένας από τους σημαντικότερους σταθμούς στην εξέλιξη της Συνοπτικής και της Θεωρητικής Μετεωρολογίας. Στην Ελλάδα το 1840 ιδρύεται η Μετεωρολογική Υπηρεσία η οποία υπάγεται στο τότε Υπουργείο Αεροπορίας και αρχίζει να λειτουργεί με ένα πιο οργανωμένο τρόπο.

Ο πόλεμος του 2<sup>ου</sup> Παγκόσμιου Πολέμου ήταν σίγουρα μια περίοδος γόνιμων εξελίξεων και μεγάλης προόδου για τη Μετεωρολογία. Επειδή οι επιχειρήσεις στην ξηρά, στη θάλασσα και στον αέρα ήταν κατά τη διάρκεια του πολέμου αυτού πολύ μεγαλύτερες από ότι κατά τον 1<sup>ο</sup> Παγκόσμιο Πόλεμο ο ρόλος της μετεωρολογίας ήταν μεγάλος και οι υπηρεσίες τις οποίες πρόσφερε εξαιρετικής σπουδαιότητας. Καθόλου αμελητέος δεν ήταν ο ρόλος της Αγγλικής Μετεωρολογικής Υπηρεσίας στην επιλογή της χρονικής στιγμής για τη μεγάλη απόβαση στη Νορμανδία. Με τα δίκτυα των σταθμών και κυρίως των σταθμών ραδιοβολίσεων και ανωτέρας ατμόσφαιρας που πύκνωσαν οδήγησαν σε μια νέα κατά ύψος μελέτη της ατμόσφαιρας και σε μια καινούργια θεώρηση για την ανάλυση και πρόγνωση των καιρικών καταστάσεων. Οι παρατηρήσεις των α/φ που πετούσαν σε μεγάλα ύψη ήταν πολύτιμες, γιατί εξαιτίας αυτών εντοπίστηκε η ύπαρξη των πολύ ισχυρών ανέμων (Jet Stream) στην κατώτερη στρατόσφαιρα. Σπουδαίο ρόλο έπαιξαν οι ηλεκτρικές συσκευές, οι οποίες εφευρέθηκαν κατά τη διάρκεια του πολέμου και ιδιαίτερα το 'radar', η χρήση του οποίου ήταν άμεση στην καθημερινή υπηρεσία του καιρού.

Στη Μεταπολεμική Ελλάδα, η Μετεωρολογική Υπηρεσία ονομάζεται πλέον Εθνική Μετεωρολογική Υπηρεσία, και υπάγεται από τότε μέχρι και σήμερα στο Υπουργείο Εθνικής Άμυνας.

Η περίοδος μετά τον 2<sup>ο</sup> Παγκόσμιο Πόλεμο υπήρξε εξ αρχής επαναστατική στην ιστορία της Φυσικής της ατμόσφαιρας. Οι γερμανικοί πύραυλοι V2 οι οποίοι έφευγαν από τους Γερμανούς στους Άγγλους σαν ιπτάμενες βόμβες, άρχισαν από το 1946 να χρησιμοποιούνται από τους Αμερικάνους για τη μελέτη της ανώτερης ατμόσφαιρας. Τέτοιου τύπου πύραυλοι και ακόμα πιο εξελιγμένης μορφής ρίχθηκαν από Αμερικάνους και Ρώσους, οι οποίοι έδωσαν στοιχεία για τη μελέτη της ατμόσφαιρας μέχρι τα ανώτερα όριά της. Μετά από αυτό η χημική σύσταση, η φυσική υφή και ο ιονισμός της ατμόσφαιρας παύουν να είναι κρυμμένα μυστικά.

Στην πρόοδο της Μετεωρολογίας συνέβαλαν οι παρατηρήσεις και τα πειράματα κατά τη διάρκεια του Διεθνούς Γεωφυσικού Έτους (1 Ιουλίου 1957 έως 31 Δεκεμβρίου 1958) το οποίο υπήρξε η μεγαλύτερη μέχρι τότε σημειωθείσα επιστημονική εκστρατεία στην ιστορία της ανθρωπότητας. Παράλληλα την ίδια περίοδο μπήκαν σε τροχιά γύρω από τη γη οι πρώτοι τεχνητοί δορυφόροι για την έρευνα της γήινης ατμόσφαιρας και της περιοχής κοντά στα όρια του κοσμικού διαστήματος. Το αποτέλεσμα των τεχνητών δορυφόρων στη Μετεωρολογία ήταν τόσο επιτυχής, ώστε αποφασίστηκε να κατασκευαστούν και ειδικοί μετεωρολογικοί δορυφόροι τόσο από τους Αμερικάνους όσο και από τους Ρώσους. Ο TIROS 1 ήταν ο πρώτος που μπήκε σε τροχιά γύρω από τη γη την 1<sup>η</sup> Απριλίου του 1960 έπειτα ακολούθησε ο Ρωσικός COSMOS. Βρισκόμαστε στον 21<sup>ο</sup> αιώνα, πλέον οι μετεωρολογικοί δορυφόροι που είναι σε τροχιά και καλύπτουν την Ευρώπη είναι οι Αμερικανικοί NOAA (σειρά TIROS), οι Ευρωπαϊκοί METEOSAT IV και οι Ρωσικοί METEOR I και II.

Το 1960 λαμβάνεται η πρώτη φωτογραφία μετεωρολογικού δορυφόρου. Από εδώ και πέρα οι δορυφόροι στέλνουν σε Επίγειους Δορυφορικούς Σταθμούς δεδομένα μετρήσεων ακτινοβολία από την επιφάνεια της Γής και της Ατμόσφαιρας. Η σταδιακή επεξεργασία των δεδομένων των μετρήσεων από τον Επίγειο σταθμό σχηματοποιείται και δίδεται σε μορφή

εικόνων που απεικονίζονται τμήματα του πλανήτη στις οποίες διαχωρίζεται η στεριά από τη θάλασσα και τα νέφη.

Σημαντικό ρόλο στην συγκέντρωση των μετεωρολογικών παρατηρήσεων της ατμόσφαιρας έχουν και τα αεροσκάφη κατά τη διάρκεια της πτήσης των μεγάλων πολιτικών αεροσκαφών που είναι συχνά εξοπλισμένα με αυτόματους αισθητήρες, έτσι ώστε να αναμεταδίδουν εν πτήση δεδομένα πίεσης, θερμοκρασίας και ανέμου. Σημειώνεται ότι ο συνολικός αριθμός αναφορών από αεροσκάφη είναι ημερησίως άνω των 2000, λόγω του γεγονότος ότι είναι διαθέσιμα σε περιοχές με ελάχιστα άλλα δεδομένα όπως οι ωκεανοί. Ωστόσο, οι μετρήσεις των δεδομένων θερμοκρασίας και ανέμου δεν είναι πολύ αξιόπιστα, και είναι πολύ συγκεντρωμένα στα ύψη των 9 με 10 km (χιλιομέτρων) κατά μήκος των μεγάλων αεροδιαδρόμων, αδυνατώντας να παρέχουν μια τρισδιάστατη εικόνα της ατμόσφαιρας.

### 2.3 Εθνική Μετεωρολογική Υπηρεσία



Εικόνα 1. EMY

Στην Ελλάδα, η Εθνική Μετεωρολογική Υπηρεσία, γνωστή επίσης και με τα αρχικά EMY και HNMS ( Hellenic National Meteorological Services ), είναι ελληνική κρατική υπηρεσία που υπάγεται στις Ένοπλες δυνάμεις και ειδικότερα στο Γενικό Επιτελείο Αεροπορίας και έχει σαν αποστολή τον προσδιορισμό και την πρόβλεψη των μετεωρολογικών συνθηκών, δηλαδή του καιρού. Η υπηρεσία διαθέτει ένα δίκτυο από ειδικούς μετεωρολογικούς σταθμούς από όπου συγκεντρώνει παρατηρήσεις και με τις οποίες ενημερώνει τους μετεωρολογικούς χάρτες και εκδίδει μετεωρολογικά δελτία, τακτικά και έκτατα, ιδιαίτερα χρήσιμα στις αεροπορικές και θαλάσσιες μεταφορές. Έτος ίδρυσης του είναι το 1931 και εδρεύει σήμερα στις εγκαταστάσεις του πρώην ανατολικού αεροδρομίου Αθηνών στο Ελληνικό Αττική. Είναι επανδρωμένο τόσο από στρατιωτικό προσωπικό (κυρίως αξιωματικούς της Σχολής Ικάρων) όσο και από πολιτικό, που περιλαμβάνει φυσικούς, μαθηματικούς και ειδικούς πληροφορικής, με επιπρόσθετο στρατιωτικό και πολιτικό προσωπικό άλλων ειδικοτήτων (διοικητικό - οικονομικό - τεχνικό).

Ο συνολικός αριθμός που απασχολεί είναι 565 άτομα, όχι μόνο στο κεντρικό κτήριο της αλλά και σε παραρτήματα σε όλη την Ελλάδα, καθώς το μετεωρολογικό της δίκτυο καλύπτει σχεδόν όλη τη χώρα. Η EMY συνεργάζεται με τις αντίστοιχες υπηρεσίες άλλων χωρών σε συγκεκριμένους χρόνους του 24ώρου μεταξύ των οποίων και ανταλλάσσονται μετεωρολογικές παρατηρήσεις, με σκοπό τις αντικειμενικότερες τιμές των μετρήσεων.

Η EMY εκτός από την πρόγνωση του καιρού, δίνει πολύτιμες πληροφορίες για τον καιρό και το κλίμα, στις Κρατικές Υπηρεσίες, στις συγκοινωνίες, στη γεωργία, στον αθλητισμό, κ.λπ.

## 2.4 Εθνικό Αστεροσκοπείο Αθηνών



Εικόνα 2. Ο Βαρόνος Γεώργιος Σίνας.

Το Εθνικό Αστεροσκοπείο Αθηνών διερεύνει την ιστορία του, το οποίο μετρά πλέον πάνω από 170 χρόνια προσφοράς στην έρευνα και την κοινωνία και συνοπτικά παραθέτει τα εξής στοιχειώδη ιστορικά στοιχεία για την ενημέρωση των πολιτών της Ελλάδος:

Στα τέλη του 1840, ο Πρόξενος της Ελλάδας στη Βιέννη, ο Βαρόνος Γεώργιος Σίνας εξέφρασε την πρόθεσή του να κάνει μια δωρεά για την ανάπτυξη της επιστημονικής έρευνας σε ένα από τα ανώτατα εκπαιδευτικά ιδρύματα. Ο φίλος του κ. Prokesh-Osten, Πρεσβευτής της Αυστρίας στην Ελλάδα και ο γραμματέας του πρέσβη ο φυσικός και αστρονόμος, Καθηγητής Γεώργιος Βούρης πείθουν τον Γεώργιο Σίνα να δωρίσει στο Ελληνικό Κράτος 500.000 δραχμές για την ίδρυση του Αστεροσκοπείου στην Αθήνα.

Την ιστορική ημέρα της 26<sup>ης</sup> Ιουνίου του 1842 πραγματοποιείται η θεμελίωση του Αστεροσκοπείου Αθηνών κατά τη διάρκεια έκλειψης Ηλίου. Στην εκδήλωση παρευρέθηκαν με μεγάλη επισημότητα ο Βασιλιάς Όθωνας, μέλη της κυβέρνησης και της Ιεράς Συνόδου της Εκκλησίας της Ελλάδος. Η περιοχή όπου επιλέχτηκε για την ανέγερση του Αστεροσκοπείου, ήταν στο Λόφο Νυμφών στο Θησείο, απέναντι από την Ακρόπολη. Μπροστά σε χιλιάδες κόσμο ο καθηγητής κ. Βούρης εκφώνησε το πανηγυρικό λόγο της ημέρας. Την λαμπρή εκδήλωση συνόδευσαν μουσική και κανονιοβολισμοί μιας φρεγάτας, που ήταν αγκυροβολημένη στον Πειραιά, κατά την στιγμή τοποθέτησης του θεμέλιου λίθου.

Το πρώτο κτίριο του Αστεροσκοπείου, γνωστό και ως κτίριο Σίνα, οικοδομήθηκε με βάση τα σχέδια του νέου Δανού αρχιτέκτονα Theophil Hansen, τα οποία παρουσιάστηκαν από τον αρχιτέκτονα Edward Schaubert. Το τελικό σχέδιο εγκρίθηκε από τον καθηγητή κ. Βούρη και τον Schumacher, μεγάλου αστρονόμου της εποχής. Η μορφή του κτιρίου είναι σταυροειδή, προσανατολισμένη με βάση τα τέσσερα σημεία του ορίζοντα και στο κέντρο υπάρχει ένας μικρός θόλος. Η αποπεράτωση του κτιρίου ολοκληρώθηκε το 1846.

Ο εμπνευστής του Αστεροσκοπείου Αθηνών, ο κ. Βούρης ήταν και ο πρώτος του Διευθυντής. Είναι η πρώτη φορά, που Έλληνας αστρονόμος έχει τη δυνατότητα να διεκπεραιώνει αστρονομικές παρατηρήσεις και να παράγει σημαντικό επιστημονικό έργο βασιζόμενο σε αυτές. Δημοσιεύει αρκετές εργασίες στο περιοδικό Astronomische Nachrichten, άλλα μεγάλο μέρος του έργου του, όπως ο κατάλογος των θέσεων περίπου 1000 αστερών παρέμεινε αδημοσίευτο. Ένα σημαντικό έργο του κ. Βούρη αποτέλεσε ο προσδιορισμός των γεωγραφικών συντεταγμένων του Αστεροσκοπείου, οι οποίες αποτέλεσαν τη βάση για την χαρτογράφηση της Ελλάδας.

Μετά από μια επιτυχημένη περίοδο σηματοδοτείται μία σημαντική αλλαγή στις 4 Δεκεμβρίου του 1858, τη διεύθυνση του Αστεροσκοπείου αναλαμβάνει ο γερμανός επιστήμονας

Ιούλιος Σμιθ (Johann Friedrich Julius Schmidt). Ο κ. Σμιθ αναβαθμίζει την μετεωρολογική υπηρεσία του Αστεροσκοπείου εκτελώντας συστηματικές μετεωρολογικές παρατηρήσεις σε διάφορα μέρη της Αττικής και αλλού. Οι παρατηρήσεις του δημοσιεύονται στο δίτομο έργο με τίτλο "Beitrage zur Physikalischen von Griechenland". Ενώ, τακτικά αποστέλλει πίνακες με μετεωρολογικά στοιχεία στο Αστεροσκοπείο του Παρισιού.

Το 1884 ο καθηγητής κ. Κοκκίδης αναλαμβάνει την διεύθυνση του Αστεροσκοπείου Αθηνών, προσωρινά και χωρίς αμοιβή. Πετυχαίνει το στόχο του το να επεκτείνει το μικρό μετεωρολογικό δίκτυο με απομακρυσμένους από την Αθήνα σταθμούς όπως αρχικά στη Ζάκυνθο, στην Κέρκυρα και στη Λάρισα.

Η αναγνώριση της συμβολής που παρείχε η υπηρεσία και η αναβάθμισή της έρχεται στις 19 Ιουνίου 1890, με την ψήφιση του ειδικού νόμου στη Βουλή, με βάση του οποίου το Αστεροσκοπείο μετατρέπεται σε κρατικό ίδρυμα και ονομάζεται Εθνικό Αστεροσκοπείο Αθηνών (Ε.Α.Α.). Διευθυντής του Ε.Α.Α. διορίζεται ο καθηγητής κ. Δημήτριος Αιγινήτης. Εκτός από το Αστρονομικό τμήμα, δημιουργούνται ακόμη δύο τμήματα το Μετεωρολογικό και το Σεισμολογικό. Ο κ. Αιγινήτης αναζητούσε νέους οικονομικούς πόρους για την χρηματοδότηση των τμημάτων. Κατάφερε να κερδίσει μία πίστωση από το Πανεπιστήμιο την οποία και χρησιμοποίησε για την επισκευή και συντήρηση του κτιρίου και των οργάνων του Ε.Α.Α. Λόγω τις τότε οικονομικής κατάστασης στην Ελλάδα, η κρατική επιχορήγηση ήταν πολύ μικρή. Ο κ. Αιγινήτης δημιουργεί την σύσταση εθνικής επιτροπής, με την βοήθεια της οποίας πραγματοποίησε έρανο και μέσα σε λίγα χρόνια συνέλεξε σημαντικό ποσό από επιφανείς Έλληνες. Μερικά από τα ονόματα των μεγάλων δωρητών είναι τα: Δ. Δωρίδη, Α. Συγγρού, Μ. Κοργιαλένιο, Π. Στεφάνοβικ, Ε. Ζαρίφη, Κ. Μαυρομιχάλη, Α. Σκουζέ και Ν. Χρυσοβελώνη. Ο κ. Αιγινήτης έχοντας κατακτήσει το στόχο του, αναβαθμίζει το μετεωρολογικό δίκτυο, το οποίο προς το τέλος της θητείας του αριθμεί περίπου εκατό σταθμούς, και καταφέρνει να ιδρύσει την σεισμολογική υπηρεσία. Πριν την αποχώρησή του οργανώνει το επιστημονικό έργο του Ε.Α.Α. και φροντίζει την έκδοση του περιοδικού του "Annalles de l'Observatoire National d'Athenes". Οι 12 τόμοι του περιοδικού που δημοσιεύθηκαν κατά τη διάρκεια της θητείας του περιέχουν τις εργασίες διαφόρων τμημάτων του Αστεροσκοπείου. Αυτός ήταν ο άνθρωπος που συνέβαλε για την εισαγωγή στην Ελλάδα του παγκόσμιου χρονομετρικού συστήματος και της ώρας της ανατολικής Ευρώπης (1916), καθώς και του Γρηγοριανού ημερολογίου (1923).

Σημαντικός διορισμός ήταν του κ. Πλακίδης στο Αστρονομικό Τμήμα του ΕΑΑ το 1915. Το 1927 προάγεται σε έκτακτο αστρονόμο και το 1928, ύστερα από πρόταση του κ. Αιγινήτη, συνεχίζει για δύο χρόνια τις σπουδές του στα διάσημα αστεροσκοπεία Greenwich, Cambridge, Paris, Strasbourg, και Heidelberg. Με αφορμή την επιστροφή του στην Ελλάδα, το 1931 αναγορεύεται διδάκτωρ και προάγεται σε τακτικό αστρονόμο. Το 1935 ο κ. Πλακίδης εκλέγεται Καθηγητής στο Πανεπιστήμιο Αθηνών, διατηρώντας την θέση του ως προϊστάμενου του Αστρονομικού Τμήματος του Αστεροσκοπείου. Μέσα στα επόμενα δύο χρόνια το ΕΑΑ διαδέχονται τρεις διευθυντές, ο κ. Νικόλαος Κριτικός ανέλαβε προσωρινά έως το 1936, όποτε διορίστηκε ο κ. Ηλίας Μαρσιολόπουλος και τέλος, το 1937, Διευθυντής του ΕΑΑ διορίζεται ο πλοίαρχος Γεώργιος Χόρς.

Ακολουθεί το Νομοθετικό Διάταγμα υπ' αριθμ. 1378/1942 όπου ιδρύονται τρεις θέσεις διευθυντών για τα αντίστοιχα Τμήματα του Εθνικού Αστεροσκοπείου Αθηνών, Αστρονομίας, Μετεωρολογίας και Σεισμολογίας. Σύμφωνα με το οποίο: «Την προεδρία του συμβουλίου του Αστεροσκοπείου αναλαμβάνει αυτοδικαίως, εκ περιτροπής επί διετία, έκαστος των τριών Διευθυντών- Καθηγητών κατά σειράν αρχαιότητας ως τακτικού καθηγητού του Πανεπιστημίου». Με βάση τον παραπάνω νόμο, εναλλάσσονται στην προεδρία του Κέντρου πολλοί καθηγητές που συμμετείχαν στη διοίκησή του έως σήμερα.

## 2.5 Ινστιτούτο Ερευνών Περιβάλλοντος και Βιώσιμης Ανάπτυξης

Το Ινστιτούτο Ερευνών Περιβάλλοντος και Βιώσιμης Ανάπτυξης (ΙΕΠΒΑ) είναι ένα από τα τρία Ινστιτούτα του Εθνικού Αστεροσκοπείου Αθηνών (ΕΑΑ) το οποίο είναι το αρχαιότερο Ερευνητικό Κέντρο στην Ελλάδα και τα Βαλκάνια.

Το Μετεωρολογικό Ινστιτούτο, που ιδρύθηκε το 1946, ξεκίνησε το 1858 τη συστηματική πραγματοποίηση καθημερινών μετεωρολογικών παρατηρήσεων, η οποία συνεχίζεται ανελλιπώς μέχρι σήμερα στον ίδιο χώρο, στις εγκαταστάσεις του ΙΕΠΒΑ στο Θησείο, στο Λόφο των Νυμφών. Το ΙΕΠΒΑ διαθέτει την αρχαιότερη και πληρέστερη ιστορική βάση κλιματικής πληροφορίας στην Ελλάδα, η οποία καλύπτει χρονική περίοδο μεγαλύτερη του ενάμισι αιώνα.

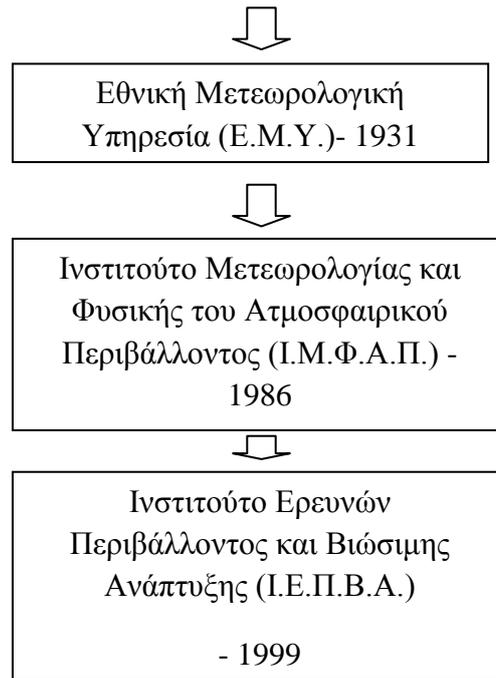
Το ΙΕΠΒΑ αποτέλεσε ουσιαστικά την πρώτη μετεωρολογική υπηρεσία της χώρας, ιδρύοντας και λειτουργώντας δίκτυο μετεωρολογικών σταθμών σε όλη την επικράτεια μέχρι το 1931, όταν το δίκτυο παραδόθηκε στην νεοσυσταθείσα τότε Εθνική Μετεωρολογική Υπηρεσία (ΕΜΥ).

Εκτός των κλιματικών ιστορικών αρχείων, το ΙΕΠΒΑ διαθέτει και ιστορικό αρχείο μετρήσεων ατμοσφαιρικού όζοντος για την περίοδο 1900-1940, το οποίο είναι και μοναδικό στην ευρύτερη περιοχή της ΝΑ Ευρώπης. Σημαντικό σημείο στην εξέλιξη του ΙΕΠΒΑ ήταν και η παρακολούθηση της ατμοσφαιρικής ρύπανσης στην Αθήνα, μέσω της συστηματικής καταγραφής των συγκεντρώσεων διάφορων ατμοσφαιρικών ρύπων από δίκτυο έξι σταθμών. Το δίκτυο αυτό πέρασε στη δικαιοδοσία του Υπουργείου Περιβάλλοντος το 1984 και αποτέλεσε τη βάση του εθνικού δικτύου καταγραφής ατμοσφαιρικής ρύπανσης.

Επόμενη σημαντική εξέλιξη σημειώνεται το 1986, με την μετονομασία σε 'Ινστιτούτο Μετεωρολογίας και Φυσικής του Ατμοσφαιρικού Περιβάλλοντος' (ΙΜΦΑΠ), τίτλος αντιπροσωπευτικότερος των νέων δραστηριοτήτων του – εκτός της Μετεωρολογίας και Κλιματολογίας- που σχετίζονται με τη Φυσική της ατμόσφαιρας, τις ήπιες μορφές ενέργειας κ.ά. Την περίοδο αυτήν, το Ινστιτούτο αναπτύσσει και επεκτείνει τις υποδομές του σε εργαστήρια και οργανολογικό εξοπλισμό, ενισχύει τις ερευνητικές του δραστηριότητες μέσα από συνεργασίες με Ερευνητικά Κέντρα και Πανεπιστήμια στην Ελλάδα και το εξωτερικό, ενώ συνεχίζει να παρέχει υπηρεσίες στην πολιτεία, στις τοπικές αρχές και σε ιδιώτες.

Η τελευταία αλλαγή του ονόματός του γίνεται το 1999, το Ινστιτούτο μετονομάζεται σε 'Ινστιτούτο Ερευνών Περιβάλλοντος και Βιώσιμης Ανάπτυξης' (ΙΕΠΒΑ) ανταποκρινόμενο στις νέες προκλήσεις και απαιτήσεις για ένα βιώσιμο περιβάλλον. Οι δραστηριότητές του επεκτείνονται περισσότερο, κυρίως προς την κατεύθυνση της μελέτης των επιπτώσεων της ανάπτυξης στο περιβάλλον, τις επιπτώσεις της κλιματικής αλλαγής, την υδρολογία και διαχείριση υδατικών πόρων, την ποιότητα του αέρα αλλά και την εξοικονόμηση ενέργειας στα κτίρια.





**Διάγραμμα 1. Χρονολογική Εξέλιξη της Μετεωρολογικής Υπηρεσίας στην Ελλάδα.**

Τα γραφεία του ΙΕΠΒΑ στεγάζονται πλέον λίγα στις ιστορικές εγκαταστάσεις του Θησείου, στο λόφο των Νυμφών σε μικρή απόσταση από την Ακρόπολη.



**Εικόνα 3. Θησείο.**

Από το 2001, τα περισσότερα γραφεία του ΙΕΠΒΑ μεταφέρθηκαν στις καινούργιες εγκαταστάσεις της Πεντέλης. Το νέο κτίριο του Ινστιτούτου βρίσκεται στο λόφο Κουφού, περίπου 15 χμ από το κέντρο της Αθήνας.



Εικόνα 4. Πεντέλη.

### 3 ΜΕΤΕΩΡΟΛΟΓΙΚΑ ΦΑΙΝΟΜΕΝΑ

Στο κεφάλαιο αυτό θα μελετήσουμε τους ορισμούς των βασικότερων μετεωρολογικών φαινομένων με τους οποίους ασχολείται το κατασκευαστικό μέρος της παρούσας εργασίας, καθώς και σε τι μονάδες μέτρησης μετριοούνται.

#### 3.1 Θερμοκρασία

Θερμοκρασία ατμόσφαιρας ονομάζεται η θερμοκρασία την οποία έχει ο ατμοσφαιρικός αέρας πάνω από μια περιοχή. Οι πηγές που θερμαίνουν τον ατμοσφαιρικό αέρα είναι ο ήλιος και το εσωτερικό της γης. Σπουδαιότερη όμως είναι ο ήλιος, που η ακτινοβολία του είναι η πρωταρχική αιτία των φαινομένων της γήινης ατμόσφαιρας. Η θερμοκρασία της ατμόσφαιρας παρουσιάζει ανώμαλες και κανονικές μεταβολές. Οι ανώμαλες οφείλονται στην εναλλαγή των αέριων μαζών, στον άνεμο, στη νέφωση, κτλ. και οι κανονικές στην περιστροφική κίνηση της γης. Από την άλλη, η θερμοκρασία της γης στους διάφορους τόπους της δεν είναι η ίδια. Η θερμοκρασία ενός τόπου εξαρτιέται από το πόσο ψυχρός ή θερμός είναι ο αέρας, που βρίσκεται πάνω σε αυτόν τον τόπο. Ο αέρας θερμαίνεται ή ψύχεται ανάλογα:

- Με τη θέση και το γεωγραφικό πλάτος του τόπου, αν δηλ., είναι κοντά στη θάλασσα, στον κάμφο ή στο βουνό, αν βρίσκεται κοντά στον ισημερινό ή στους πόλους.
- Με την εποχή· αν είναι άνοιξη, καλοκαίρι, φθινόπωρο ή χειμώνας.
- Με τις καιρικές συνθήκες, αν έχει συννεφιά, αν φυσούν ψυχροί άνεμοι.
- Από το ύψος. Η θερμοκρασία του αέρα ελαττώνεται, όσο ανυψωνόμαστε μέσα στην ατμόσφαιρα. Η πτώση (ελάττωση) της θερμοκρασίας ανά 100 μέτρα στην Ελλάδα είναι 0,6° K. (Αξιοτέλλης, 1993)

Η θερμοκρασία της ατμόσφαιρας μετριέται με τα θερμόμετρα και υπάρχουν διάφορες κλίμακες μέτρησης, με συνηθισμένες κλίμακες τις Κελσίου (Celsius, σύμβολο °C), Κέλβιν (Kelvin, σύμβολο °K) και Φαρενάιτ (Fahrenheit, σύμβολο °F). Στην Ελλάδα χρησιμοποιείται η κλίμακα Κελσίου και ορίζεται ως «Το σημείο βρασμού του νερού είναι στους 100 °C και το σημείο παγιοποίησης του, στους 0 °C». Η κλίμακα  $K$  σχετίζεται με την κλίμακα  $C$  ως εξής:  $K=C+273.16$ .

#### 3.2 Υγρασία

Ο ατμοσφαιρικός αέρας περιέχει και υδρατμούς. Η υγρασία είναι το φυσικό μέγεθος, το οποίο γενικά δείχνει την ποσότητα των υδρατμών, που υπάρχει στην ατμόσφαιρα ή στο συγκεκριμένο χώρο. Απόλυτη υγρασία είναι το πηλίκο της μάζας  $m$  των υδρατμών, που περιέχονται σε ορισμένο όγκο αέρα  $V$  δια του όγκου τούτου,

$$D_{\text{απόλ.}} = M_{\text{υδρατμών}} / V_{\text{αέρα}}$$

Σχετική υγρασία είναι το πηλίκο της μάζας των υδρατμών, που υπάρχουν σε κάποιον όγκο αέρα, δια της μάζας  $\mu$  των υδρατμών, τους οποίους αν είχε ο όγκος αυτός θα ήταν κορεσμένος,

$$D_{\text{σχετ.}} = m / \mu$$

Λέγοντας κορεσμένο αέρα εννοούμε τον αέρα εκείνον στον οποίον, αν προστεθεί οσηδήποτε μικρή ποσότητα υδρατμών, αυτή θα πάρει μορφή σταγονιδίων.

Άλλες εκφράσεις της υγρασίας, που χρησιμοποιούνται στη μετεωρολογία, είναι το σημείο δρόσου ή κόρου του ατμοσφαιρικού αέρα, η αναλογία μείγματος, η ειδική υγρασία και άλλα.

Η μέγιστη περιεκτικότητα του ατμοσφαιρικού αέρα σε υδρατμούς είναι συνάρτηση της πίεσης και της θερμοκρασίας του και καθορίζεται από το γνωστό νόμο Magnus-Tetens της θερμοδυναμικής. Ο νόμος Magnus-Tetens δίνει τη μερική πίεση (τάση) των υδρατμών που περιέχονται σε δείγμα ατμοσφαιρικού αέρα πίεσης  $P$  και θερμοκρασίας  $T$ .

$$e_s(T_s) = 6,1078 \exp\left(\frac{17,625 T}{T+243,04}\right)$$

όπου  $e_s$  η μέγιστη τάση υδρατμών σε hPa ως συνάρτηση της θερμοκρασίας  $T$  της ατμόσφαιρας μετρούμενης σε βαθμούς Κελσίου ( $^{\circ}\text{C}$ ).

Από το νόμο αυτό προκύπτει ότι η συγκέντρωση των υδρατμών σε δείγμα αέρα είναι ανάλογη της θερμοκρασίας. Άρα όσο ψυχρότερος είναι ο αέρας, τόσο λιγότερη μάζα υδρατμών μπορεί να συγκρατήσει. Όταν δείγμα αέρα περιέχει τη μέγιστη δυνατή ποσότητα υδρατμών τότε ονομάζεται κορεσμένο. Η υγρασία μεταβάλλεται από περιοχή σε περιοχή ανάλογα με το ύψος, που κάνουμε μετρήσεις και την εποχή του έτους.

Η φυσική σημασία της σχετικής υγρασίας είναι ότι δείχνει το πόσο απέχει το δείγμα αέρα από την κατάσταση κορεσμού. Σχετική υγρασία 100% σημαίνει ότι το δείγμα είναι κορεσμένο. Η σχετική υγρασία στους αυτόματους σταθμούς μετράται με ηλεκτρονικά υγρόμετρα. Η αρχή λειτουργίας τους στηρίζεται στη μεταβολή της χωρητικότητας συναρτήσει της υγρασίας πυκνωτή του οποίου το διηλεκτρικό είναι υγροσκοπικό. (Αξιοτέλλης, 1993)

### 3.3 Ατμοσφαιρική Πίεση

Η βαρύτητα είναι η αιτία που παραμένει το νερό και τα αέρια στην επιφάνεια της γης και δεν διαφεύγουν στο διάστημα. Η γη περιβάλλεται από ένα λεπτό στρώμα αερίων το οποίο ονομάζεται ατμόσφαιρα. Η ατμόσφαιρα έχει βάρος το οποίο πιέζει όλα τα σώματα που βρίσκονται μέσα σε αυτή. Η πίεση αυτή ονομάζεται ατμοσφαιρική πίεση.

Στην επιφάνεια της Γης η ατμοσφαιρική πίεση ισούται κατά μέσον όρο με το βάρος στήλης ύδατος ύψους 11 μέτρων (m) περίπου, ή 1 εκατομμύριοδυνες ανά  $\text{cm}^2$ . Στην μετεωρολογία 1000 δύνες/ $\text{cm}^2$  αντιστοιχούν σε ένα χιλιοστόμετρο (milibar). Κατά μέσο όρο στην επιφάνεια της γης η ατμοσφαιρική πίεση είναι 1000 milibar. Με απόφαση του Διεθνούς Μετεωρολογικού Οργανισμού (WMO) καθιερώθηκε ως μονάδα μέτρησης της ατμοσφαιρικής πίεσης το Εκτοπασκάλ (1 hPa) το οποίο η αντιστοιχία είναι 1 hPa=1 milibar.

Η πίεση μειώνεται ανάλογα με το υψόμετρο. Για τον λόγο αυτό οι μετρήσεις που γίνονται σε ύψος μεγαλύτερο από αυτό της επιφάνειας της θάλασσας, πρέπει να τροποποιηθούν ώστε να αντιστοιχούν σε μηδενικό ύψος. Η ενέργεια αυτή ονομάζεται ως **αναγωγή στην επιφάνεια της θάλασσας**. (Αξιοτέλλης, 1993)

### 3.4 Βαρομετρική Πίεση

Στη ατμοσφαιρική πίεση η βαρομετρική τάση ονομάζεται η τιμή της μεταβολής της ατμοσφαιρικής πίεσης εντός 3 ωρών προ εκάστης καθορισμένης κύριας ώρας παρατήρησης. Οι επίσημες μετεωρολογικές παρατηρήσεις γίνονται σε καθορισμένους χρόνους του 24ώρου.

Η βαρομετρική τάση λαμβάνεται από το διάγραμμα του βαρογράφου. Η κατανομή των βαρομετρικών πιέσεων και οι βαρομετρικές τάσεις αποτελούν σπουδαιότατα στοιχεία στις μεταβολές του καιρού. (Περογιαννάκης, 1974)

## 4 ΟΡΓΑΝΑ ΜΕΤΡΗΣΗΣ

Στο κεφάλαιο αυτό θα αναφερθούμε στα όργανα μετρήσεων των μετεωρολογικών φαινομένων που μελετήθηκαν στο προηγούμενο κεφάλαιο.

### 4.1 Θερμοκρασία

Η μέτρηση των τιμών της θερμοκρασίας γίνεται με τα κοινά υδραργυρικά θερμόμετρα ενώ η καταγραφή γίνεται με τους θερμογράφους. Η αντικειμενικότερη μέτρηση των τιμών τόσο στα θερμόμετρα όσο και στους θερμογράφους είναι να τοποθετημένα τα όργανα μέσα σε ένα μετεωρολογικό κλωβό και η μέτρηση της θερμοκρασίας να γίνεται πάντοτε υπό σκιά.

Στο υδραργυρικό θερμόμετρο δίνεται η τιμή της θερμοκρασίας του αέρα με ακρίβεια ενός δεκάτου του βαθμού Κελσίου. Ο μετεωρολόγος θα πρέπει να προσαρμόσει το μάτι του στην ελεύθερη επιφάνεια του υδραργύρου και να διαβάσει την τιμή της θερμοκρασίας σε βαθμούς Κελσίου στην κλίμακα που έχει πάνω του προσαρμοσμένο το θερμόμετρο.



Εικόνα 5. Υδραργυρικό Θερμόμετρο.

Άλλα όργανα μέτρησης της θερμοκρασίας είναι τα ακροβάθμια θερμόμετρα, τα οποία τοποθετούνται σε οριζόντια διάταξη και αποτελούνται από το μεγιστοβάθμιο και από το ελαχιστοβάθμιο θερμόμετρο. Στο μεγιστοβάθμιο θερμόμετρο ο εσωτερικός σωληνίσκος που περιέχει τον υδράργυρο παρουσιάζει μία στένωση έτσι ώστε να αποκλείεται η επιστροφή του υδραργύρου, ύστερα από συστολή λόγω ψύξης. Για να επανέλθει πρέπει να 'τινάξουμε' το θερμόμετρο. Η ελεύθερη άκρη της στήλης του υδραργύρου δείχνει την απολύτως μέγιστη τιμή της θερμοκρασίας του αέρα. Η μέτρηση αυτή γίνεται, συνήθως, κατά την παρατήρηση της 20:00 τοπικής ώρας.

Επίσης, το ελαχιστοβάθμιο θερμόμετρο διαφέρει από τα άλλα θερμόμετρα στο γεγονός ότι διαθέτει οινόπνευμα και όχι υδράργυρο, μέσα στο οινόπνευμα κινείται ένας μικρός μεταλλικός δείκτης, που έχει τη μορφή λεπτού αλτήρα μήκους περίπου 2cm. Με τη συστολή του οινόπνευματος, λόγω ψύξης, ο δείκτης αυτός παρασύρεται από την ελεύθερη άκρη της στήλης του οινόπνευματος, προς τη λεκάνη του θερμομέτρου. Όταν σταματάει η συστολή και αρχίζει η διαστολή του οινόπνευματος, λόγω αύξησης της θερμοκρασίας, ο δείκτης αυτός δεν παρασύρεται από τη διαστελλόμενη στήλη του οινόπνευματος αλλά παραμένει ακίνητος. Διαβάζοντας κανείς την ένδειξη της κλίμακας που συμπίπτει με το άκρο του δείκτη που είναι στη αντίθετη πλευρά από την λεκάνη έχει την απολύτως ελάχιστη τιμή της θερμοκρασίας του αέρα π.χ. κατά τη διάρκεια του 24ώρου. Είναι γνωστό ότι η ελάχιστη αυτή τιμή σημειώνεται λίγο μετά την ανατολή του ηλίου. Για το λόγο αυτό η μέτρηση αυτή γίνεται, συνήθως, κατά την παρατήρηση της 08:00 τοπικής ώρας. Η συνεχής καταγραφή των τιμών της θερμοκρασίας του αέρα επιτυγχάνεται με τη χρήση του θερμογράφου. Στο όργανο αυτό ένα τμήμα του λειτουργεί ως θερμογράφος και ένα άλλο ως υγρογράφος.

Όπως είναι φανερό η γραφίδα του θερμογράφου εφάπτεται συνεχώς μιας ταινίας η οποία είναι βαθμονομημένη σε βαθμούς Κελσίου (οριζόντιες γραμμές), ενώ οι κατακόρυφες γραμμές αναφέρονται στο χρόνο (χαραγμένες ανά 2ωρα). Η γραφίδα αυτή προσαρμόζεται σε ένα μεταλλικό κύλινδρο που είναι εφοδιασμένος με ωρολογιακό μηχανισμό ο οποίος περιστρέφει τον κύλινδρο με σταθερή συχνότητα, (μια πλήρη περιστροφή ανά εβδομάδα ή ανά 24ωρο). Η γραφίδα με τον τρόπο αυτό καταγράφει στην ταινία, μία συνεχή καμπύλη από την ανάγνωση της οποίας μπορεί να πάρει κανείς τις τιμές της θερμοκρασίας του αέρα ανά τακτά χρονικά διαστήματα, ώρα, δίωρο κλπ. (Γαλάνη, 2010)

## 4.2 Υγρασία

Η σχετική υγρασία μετριέται με τα εξής όργανα: τα υγρόμετρα, τα χημικά, συμπυκνωτικά, τα ψυχρόμετρα, τα απορροφήσεως και τα ηλεκτρικά. Στην μετεωρολογία αυτά που είναι πιο γνωστά είναι: τα ψυχρόμετρα August και ο υγρογράφος δια τριχών.



Εικόνα 6. Ψυχρόμετρο August.

Το ψυχρόμετρο August αποτελείται από ένα ζευγάρι όμοιων θερμομέτρων τα οποία στερεώνονται κατακόρυφα και με απόσταση μεταξύ τους 10 cm το ένα από το άλλο. Το ένα θερμοόμετρο διατηρείται γυμνό και ξηρό και παρέχει την θερμοκρασία αέρος κατά την παρατήρηση. Το δοχείο του άλλου θερμομέτρου καλύπτεται από λεπτό ύφασμα και το θερμοόμετρο αυτό καλείται υγρό. Το ύφασμα διατηρείται συνεχώς υγρό μέσω μιας ειδικής σύνδεσής του με δοχείο γεμάτο νερό.

Σε συνήθεις ατμοσφαιρικές συνθήκες το υγρόμετρο δείχνει θερμοκρασία χαμηλότερη του ξηρού και υπό ειδικές συνθήκες, ίση προς αυτή. Σε αυτή την περίπτωση θεωρείται ότι η σχετική υγρασία είναι 100 %. Η παρατήρηση είναι η ανάγνωση των δύο θερμομέτρων (ξηρό – υγρό) είναι περιοδική και γίνεται ανά 24ωρο. Η ακρίβεια των θερμομέτρων είναι 0,1 C.



Εικόνα 7. Υγρογράφος.

Ο υγρογράφος δια τριχών είναι όργανο όπου ευπαθές μέρος του αποτελείται από ειδικό σύστημα ανθρώπινων τριχών των οποίων η μεταβολή του μήκους τους, ανάλογα με την σχετική υγρασία, προκαλεί την κάθετη κίνηση γραφίδας πάνω σε χάρτινη ταινία που είναι περιτυλιγμένη πάνω σε περιστρεφόμενο κύλινδρο. Η ταινία είναι εβδομαδιαίας καταγραφής και καταγράφει σχετική υγρασία από 0 έως 100 %. (Μακρυνίτσας, 2007)

### 4.3 Βαρομετρική Πίεση

Όργανο μέτρησης της ατμοσφαιρικής πίεσης αποτελεί το βαρόμετρο. Το όργανο αυτό διακρίνεται σε υδραργυρικά, μεταλλικά και αυτογραφικά. Η βαρομετρική πίεση αποτελεί ένα από τα σημαντικότερα μετεωρολογικά στοιχεία και είναι αυτό που μπορεί να μετρηθεί ακριβέστερα.



Εικόνα 8. Βαρόμετρο.

Η κύρια χρήση του είναι για προβλέψεις μετεωρολογικών φαινομένων που πρόκειται να συμβούν στο άμεσο μέλλον (τις επόμενες 24 ώρες). Για παράδειγμα, μια υψηλή βαρομετρική πίεση υποδηλώνει αίθριο καιρό, ενώ μια χαμηλή σημαίνει ότι μια καταιγίδα πλησιάζει, μιλώντας πάντα για μεταβολές τοπικού επιπέδου.

Δίκτυα τέτοιων αισθητήρων δημιουργούν χάρτες βαρομετρικής πίεσης κάνοντας χρήση των ισοβαρικών γραμμών (γραμμές ίδιας πίεσης). Μέσω αυτών των γραμμών μπορούμε να προβλέψουμε την κίνηση μετεωρολογικών συστημάτων σε ευρύτερες περιοχές. Ένα υψηλό δυναμικό αποτελεί ένα φραγμό στην μετατόπιση ενός τέτοιου συστήματος, ενώ μια γραμμή χαμηλού δυναμικού αντιπροσωπεύει την πιθανότερη κίνηση του μετεωρολογικού συστήματος, αφού αποτελεί την διαδρομή με την μικρότερη αντίσταση.

Όμως, για την σωστή μέτρηση της βαρομετρικής πίεσης απαιτείται ακριβής μέτρηση της θερμοκρασίας αλλά και του υψομέτρου στο οποίο είναι τοποθετημένο το βαρόμετρο. Μια μικρή μεταβολή των δύο παραπάνω μεγεθών (θερμοκρασίας, υψομέτρου) θα επηρέαζε σημαντικά την προς μέτρηση βαρομετρική πίεση. (Καλωμενίδης, 2001)

#### 4.4 Εύχρηστα όργανα στην καθημερινότητα του ανθρώπου

Στην καθημερινότητά μας χρησιμοποιούμε είτε το απλό υδραργυρικό θερμόμετρο κρεμασμένο στον τοίχο μας, είτε κάποια ψηφιακά θερμόμετρα που πωλούνται στο εμπόριο και παρέχουν ταυτόχρονα την δυνατότητα μετρήσεων όπως η θερμοκρασία εσωτερικού/εξωτερικού χώρου καθώς και της υγρασίας.



Εικόνα 9. Υδραργυρικό Θερμόμετρο.

Στην παρούσα εργασία χρησιμοποιήθηκαν για μελέτη και σύγκριση τιμών τα εξής ψηφιακά όργανα μέτρησης :

- Το **Avidsen Lulea 107240**.

Ένας μικρός μετεωρολογικός σταθμός, ο οποίος ενημερώνει για την εσωτερική θερμοκρασία και υγρασία. Τα τεχνικά χαρακτηριστικά του είναι τα εξής:

- ✓ Εύρος θερμοκρασίας :  $-10^{\circ}\text{C}$  έως  $+50^{\circ}\text{C}$
- ✓ Εύρος υγραμέτρου : 20% - 95%
- ✓ Βάρος : 45 g
- ✓ Διαστάσεις : 80x65x13



Εικόνα 10. Ο Avidsen.

Η παρακάτω φωτογραφία δείχνει εσωτερικά από τι αποτελείται.



Εικόνα 11. Ο Avidsen εσωτερικά.

Παρατηρούμε ότι για την μέτρηση της θερμοκρασίας κάνει χρήση μίας αντίστασης, η οποία έχει την μορφή ενός κόκκου φακής. Για την μέτρηση της υγρασίας χρησιμοποιείται πυκνωτής με διηλεκτρικό τον αέρα. Υπάρχει ένας κρύσταλλος ταλάντωσης, ο οποίος είναι απαραίτητος για τον χρονισμό του τοπικού ταλαντωτή του ολοκληρωμένου. Ωστόσο, το ολοκληρωμένο που χρησιμοποιείται δεν αναγνωρίζεται, αφού έχει την μορφή μιας μεγάλης σταγόνας πάνω στην πλακέτα.

- Το **OEM TA-368** .

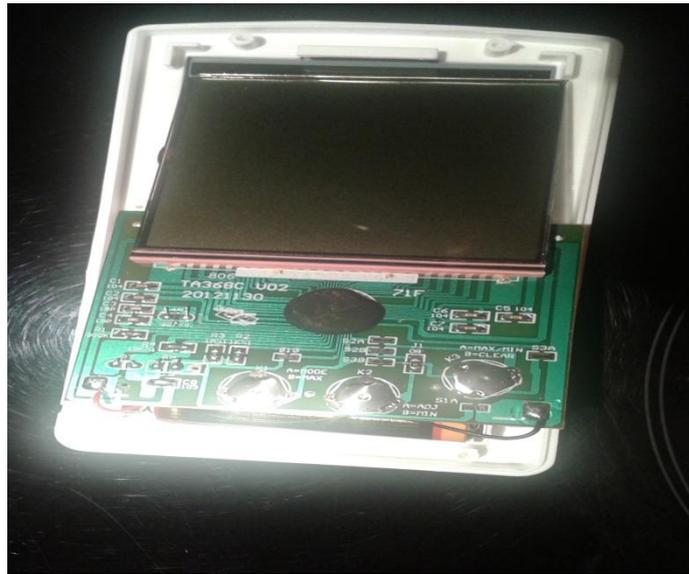
Το ψηφιακό όργανο μέτρησης διαθέτει υγρόμετρο, θερμόμετρο και ρολόι. Οι μετρήσεις της θερμοκρασίας και της υγρασίας είναι οι μέγιστες, οι ελάχιστες και οι μέσες τιμές τους. Έχει τα εξής τεχνικά χαρακτηριστικά:

- ✓ Εύρος θερμοκρασίας: -10~50°C
- ✓ Ακρίβεια θερμοκρασίας: ±1°C
- ✓ Εύρος υγρασίας: 10~99%RH
- ✓ Ακρίβεια υγρασίας: ±5%RH
- ✓ Μπαταρία: 1 × 1.5V AAA
- ✓ Βάρος: 94 g
- ✓ Διαστάσεις : 125x70x15



Εικόνα 12. Ο TA-368.

Στη φωτογραφία που ακολουθεί διαφαίνονται τα εσωτερικά στοιχεία από τα οποία αποτελείται το συγκεκριμένο όργανο μέτρησης. Όπως παρατηρούμε γίνεται χρήση των ίδιων στοιχείων με το προηγούμενο όργανο ένδειξης θερμοκρασίας-υγρασίας.



Εικόνα 13. Ο TA-368 εσωτερικά.

- Το OEM TA298

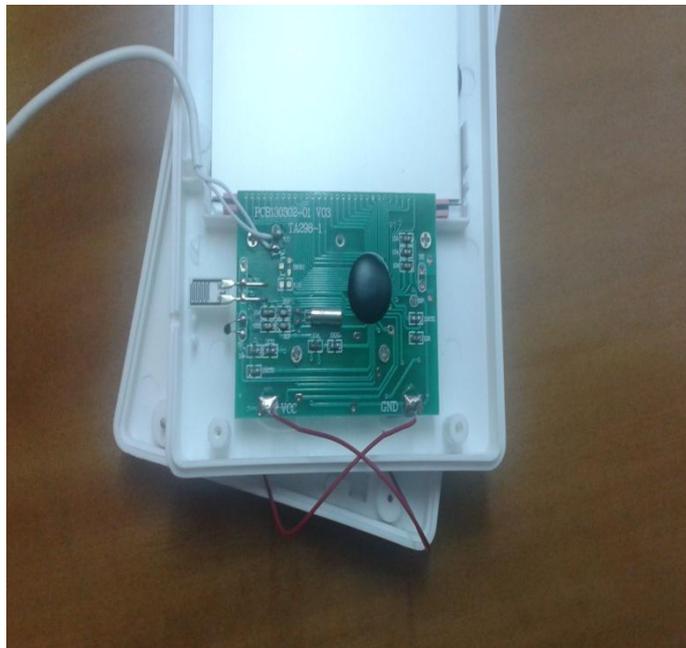
Η ψηφιακή αυτή συσκευή διαθέτει υγρασιόμετρο, θερμομέτρο με διπλό αισθητήρα για εσωτερική και εξωτερική θερμοκρασία, ρολόι, ξυπνητήρι και μεγάλη LCD οθόνη και με μεγάλα ψηφία. Τα τεχνικά χαρακτηριστικά του είναι:

- ✓ Μεγάλη οθόνη LCD αναγραφής υγρών κρυστάλλων
- ✓ Μεγάλη ακρίβεια στις μετρήσεις υγρασίας και θερμοκρασίας
- ✓ Ένδειξη εσωτερικής θερμοκρασίας
- ✓ Ένδειξη εξωτερικής θερμοκρασίας
- ✓ Ένδειξη υγρασίας εσωτερικού χώρου
- ✓ Επιλογή κλίμακας θερμοκρασίας °C /°F
- ✓ Εύρος υγρασίας : 10%RH ~ 99%RH
- ✓ Μνήμη μετρήσεων MIN/MAX για την υγρασία και την θερμοκρασία
- ✓ Σύνδεση έξτρα αισθητήρα θερμοκρασίας
- ✓ Αναγραφή ώρας ( 12 / 24 )
- ✓ Λειτουργία ξυπνητηριού
- ✓ Εύρος εξωτερικής θερμοκρασίας : -50°C ~ +70°C (-58°F ~ +158°F)
- ✓ Διαστάσεις : 150x90x20mm
- ✓ Βάρος : 166 g
- ✓ Τροφοδοσία : 1x1.5V (AAA) με χαμηλή κατανάλωση



Εικόνα 14. Ο TA-298.

Στην παρακάτω φωτογραφία απεικονίζονται τα εσωτερικά στοιχεία του θερμομέτρου-υγρομέτρου. Και αυτό μοιάζει με τα προηγούμενα.



Εικόνα 15. Ο TA-298 εσωτερικά.

## 5 ΜΕΤΕΩΡΟΛΟΓΙΚΟΣ ΣΤΑΘΜΟΣ

Σε αυτό το κεφάλαιο θα μελετήσουμε τον ορισμό του μετεωρολογικού σταθμού και θα ανακαλύψουμε τους μετεωρολογικούς σταθμούς στην Ελλάδα και στον κόσμο.

### 5.1 Ορισμός Μετεωρολογικού Σταθμού

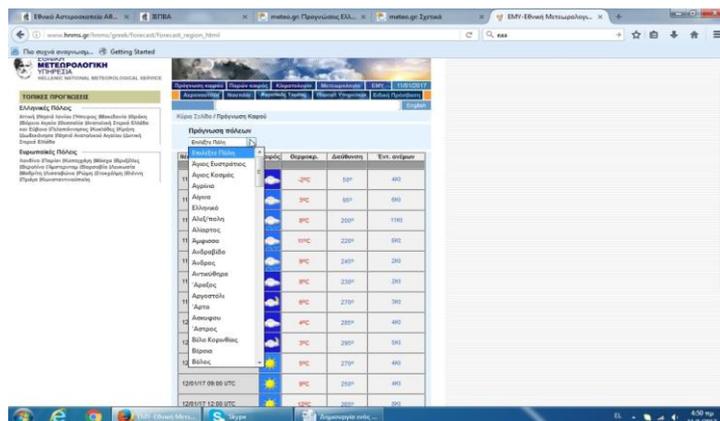
Ο Μετεωρολογικός σταθμός είναι ένα επίγειο σημείο στο οποίο πραγματοποιούνται τακτικές μετεωρολογικές παρατηρήσεις. Αποτελεί μία πλήρες εξοπλισμένη μόνιμη εγκατάσταση στην οποία βρίσκονται πολλά μετεωρολογικά όργανα, τόσο μέσα σε ένα μετεωρολογικό κλωβό είτε εκτός αυτού αλλά μέσα στον χώρο είτε και εντός αυτού, όπως επαναλήπτες μετεωρολογικών οργάνων. Η θέση τοποθέτησης αυτών των σταθμών αποφασίζεται από τη κεντρική Εθνική Μετεωρολογική Υπηρεσία της κάθε χώρας έτσι ώστε στο σύνολό τους αυτοί να αποτελούν ένα ενιαίο δίκτυο μετεωρολογικής παρατήρησης.

Πολύ σημαντικό είναι το γεγονός ότι κάθε Μετεωρολογικός σταθμός φέρει διεθνή αριθμό ταυτότητας με τον οποίο και απεικονίζεται στους μετεωρολογικούς χάρτες. Οι Μετεωρολογικοί σταθμοί επανδρώνονται από επιστημονικό προσωπικό ή ειδικά εκπαιδευμένο για τις ανάγκες των παρατηρήσεων. Στην Ελλάδα Μετεωρολογικοί σταθμοί υπάρχουν στις κυριότερες πόλεις, στους μεγάλους λιμένες και σε όλα τα αεροδρόμια της χώρας. Από τους Σταθμούς αυτούς μεταβιβάζονται σε τακτά χρονικά διαστήματα οι παρατηρούμενες ενδείξεις των φερομένων οργάνων με ειδικό κωδικοποιημένο τύπο σήματος. Οι σημαντικές αυτές αναφορές των μετεωρολογικών σταθμών στη κεντρική υπηρεσία καταχωρούνται στους υπό σύνταξη μετεωρολογικούς χάρτες της ευρύτερης περιοχής, από τη μελέτη των οποίων εξάγονται συμπεράσματα πρόβλεψης καιρού.

Επίσης μετεωρολογικοί σταθμοί θεωρούνται πέραν των ειδικών επιστημονικής έρευνας πλοίων και όλα εκείνα που φέρουν μετεωρολογικό κλωβό και καταγράφουν τακτές μετεωρολογικές παρατηρήσεις. Και αυτά φέρουν ιδιαίτερους κωδικούς αριθμούς με τους οποίους και υποβάλουν "μέτεο-σήματα" στη κεντρική μετεωρολογική υπηρεσία της εγγύτερης Χώρας του χώρου που διαπλέουν ή σε διεθνή μετεωρολογική υπηρεσία αν βρίσκονται σε Ωκεανούς.

### 5.2 Μετεωρολογικοί Σταθμοί στην Ελλάδα

Σύμφωνα με την ιστοσελίδα της Εθνικής Μετεωρολογικής Υπηρεσίας οι σταθμοί από τους οποίους λαμβάνει διάφορες μετρήσεις είναι εγκατεστημένοι στις πρωτεύουσες κάθε νομού και σε κάποιες επιλεγμένες μεγάλες πόλεις της χώρας.



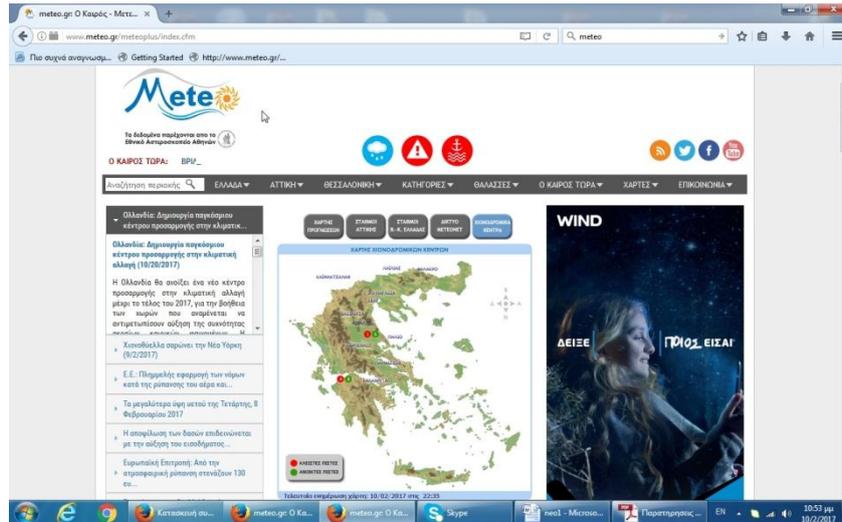
Εικόνα 16. Η σελίδα της EMY.

Το Εθνικό Αστεροσκοπείο Αθηνών έχει σχεδιάσει και αναπτύξει τον κόμβο **meteo.gr** με σκοπό την παροχή απλουστευμένων προγνώσεων καιρού για το ευρύ κοινό. Περιλαμβάνει προγνώσεις για συνολικά περίπου **500 περιοχές** και πόλεις της χώρας μας οι οποίες δίνονται σε μορφή πινάκων, ενώ πλήθος συμβόλων δίνουν στον χρήστη μια εύληπτη εικόνα του καιρού των επόμενων τριών ημερών. Η υπηρεσία μελών υποστηρίζει αυτή την στιγμή την τροποποίηση της αρχικής σελίδας ανάλογα με τις προτιμήσεις του χρήστη.

Η υποστήριξη των μετεωρολογικών μοντέλων που τροφοδοτούν τη σελίδα με προγνωστικά στοιχεία γίνεται από τη Βάσω Κοτρώνη, ερευνήτρια του Εθνικού Αστεροσκοπείου Αθηνών και τον Κώστα Λαγουβάρδο, ερευνητή του Εθνικού Αστεροσκοπείου Αθηνών. Ο σχεδιασμός, η ανάπτυξη και η συντήρηση του κόμβου αυτού γίνεται από τον Αλέξανδρο Μούστρη web developer/designer.

Καθημερινά δημιουργούνται προγνώσεις για τις επόμενες 144 ώρες ομαδοποιημένες σε 6ωρα (πρωί, μεσημέρι, απόγευμα και αργά την νύχτα). Επίσης παρέχονται προγνώσεις έντασης και διεύθυνσης ανέμου με την μορφή διαδραστικών χαρτών, προγνώσεις ύψους κύματος για όλες τις ελληνικές θάλασσες όπως και προγνώσεις υπερϊώδους ακτινοβολίας.

Το δίκτυο αυτόματων σταθμών του Ε.Α.Α. που αποτελείται από περίπου 240 σταθμούς τύπου Davis, οι οποίοι μετρούν όλες τις βασικές μετεωρολογικές παραμέτρους, δηλαδή πίεση, θερμοκρασία, υγρασία, βροχόπτωση, διεύθυνση και ένταση του ανέμου μεταδίδουν συνεχώς και σε πραγματικό χρόνο τις μετρήσεις από όλη σχεδόν την Ελλάδα. Ορισμένοι σταθμοί καταγράφουν επίσης και την ηλιακή και υπεριώδη ακτινοβολία. Όλες οι καταγραφές ανανεώνονται ανά 10 λεπτά, συλλέγονται σε πραγματικό χρόνο από το Ε.Α.Α. και αφού περάσουν από ποιοτικό έλεγχο, αρχειοθετούνται για μελλοντική χρήση.



Εικόνα 17. Η σελίδα του meteo.gr

Το meteo.gr χρησιμοποιεί ότι πιο σύγχρονο υπάρχει στον χώρο της μετεωρολογίας και διαρκώς αναζητά μέσω του επιστημονικού προσωπικού του την βελτίωση των παρεχόμενων δεδομένων. Οι τεχνολογίες που χρησιμοποιούνται είναι στην αιχμή του δόρατος της πληροφορικής επιστήμης. Σήμερα τροφοδοτεί με δεδομένα εκατοντάδες sites οργανισμών, επιχειρήσεων και ιδιωτών σε όλη την Ελλάδα.

Ενδεικτικά κάποιοι από τους ενεργούς σταθμούς φαίνονται παρακάτω:

Σταθμός	Προ θ	Ανεμος °	Προσ m/s	Υγρασία %	Μαξ άνεμος °	Low °C	High °C	Υγρος βροχής mm	
ΣΑΠΙΝΙΩΤΑ ΔΙΚ	02:40	-7.1	37	1017	40%	55	-10.9	0.0	
ΑΤ ΣΑΡΑΤΣΕ ΚΑΘΩΝ	02:40	-0.0	0	1017	94%	6	-1.0	0.0	
ΑΤ ΣΑΡΑΤΣΕ Η ΑΤΤ	02:40	3.7	5	1015	69%	13	1.3	0.0	
ΑΤ ΚΥΡΑΚΗ ΙΩΑΝ	02:20	-3.6	0	1017	94%	8	-4.4	-3.6	0.0
ΑΤΡΑ ΠΑΡΕΑΣ	02:40	-8.2	0	1016	80%	13	-8.2	-8.2	0.0
ΑΤΟΙ ΘΕΣΣΑΛΟΝΙΚΗΣ	02:40	2.2	8	1015	79%	16	0.4	2.1	0.0
ΑΤΟΣ ΜΗΔΙΑΔΕ	02:40	7.1	3	1016	50%	15	6.1	0.1	0.0
ΑΘΗΝΑ ΓΚΑΩ	02:40	2.4	0	1017	77%	5	1.9	3.9	0.0
ΑΓΓΙΑ	02:40	4.4	0	1016	64%	10	3.4	5.0	0.0
ΑΠΟΙΚΟ	02:40	3.6	0	1013	69%	8	1.0	4.0	0.0
ΑΛΑΤΩΝΑ ΝΕΣΣΩΝΑΣ	02:30	0.7	3	1014	67%	16	-2.4	0.9	0.0
ΑΛΕΞΑΝΔΡΗ	02:40	-2.9	3	1014	91%	11	-2.9	-1.7	0.0
ΑΛΙΒΑΝΟΣ ΚΑΘΩΝ	02:40	2.9	6	1017	87%	16	2.9	3.7	0.0
ΑΒΑΝΙΑΔΑ	02:40	6.1	15	1011	70%	31	2.8	6.1	0.0
ΑΙΓΙΟΥΣ	02:40	10.0	11	1017	54%	44	10.0	11.9	0.0
ΑΙΓΕΙΟΧΩΡΙΟ	02:30	3.3	3	1016	74%	10	3.1	4.6	0.0
ΑΙΓΙΝΤΑΔ	02:40	-16.9	3	1017	62%	6	-12.8	-11.1	0.0
ΑΙΘΑΛΕΙΑ	02:40	-6.4	0	1017	90%	5	-7.8	-6.7	0.0
ΑΙΘΑΛΩΝΑ	02:40	1.4	8	1014	91%	21	1.4	3.1	0.2
ΑΜΑΡΥΣΣΟΣ	02:30	2.6	0	1015	72%	19	3.0	7.3	0.0
ΑΜΑΡΩ	02:40	0.5	26	1017	61%	64	8.4	8.8	0.0
ΑΝΔΡΕΑΣΑΝΑ	02:40	1.6	6	1014	69%	23	-0.9	1.2	0.0
ΑΝΔΡΟΣ	02:40	5.1	9	1016	73%	19	4.8	6.3	0.0
ΑΝΩ ΙΚΩΤΑ	02:40	1.2	3	1016	77%	11	0.9	2.9	0.0
ΑΝΩΤΑ	02:30	1.2	6	1017	69%	15	-4.5	0.4	0.0
ΑΠΟΛΟΝΙΑΣ ΝΑΞΟΥ	02:40	7.9	6	1016	62%	34	7.9	8.8	0.0

Εικόνα 18. Α\_Ενεργοί σταθμοί.

# ΔΗΜΙΟΥΡΓΙΑ ΕΝΟΣ ΜΕΤΕΩΡΟΛΟΓΙΚΟΥ ΣΤΑΘΜΟΥ ΜΕ ΧΡΗΣΗ ΜΙΚΡΟΕΛΕΓΚΤΗ ΚΑΙ ΕΜΦΑΝΙΣΗ ΔΕΔΟΜΕΝΩΝ ΣΕ ΣΕΛΙΔΑ

Station Name	Time	X	Y	Z	W	S	T	P	Q
ΔΑΒΑΡΟΠΟΛΙΣ ΝΑΣΣΟΥ	03:40	3.4	0	1018	74%	21	3.6	5.8	0.0
ΔΕΡΒΕΝΑ ΚΟΡΝΘΙΑΣ	03:30	4.4	6	1014	64%	21	3.2	5.1	0.0
ΔΕΡΒΕΝΑ ΔΑΦΝΙΩΝ	03:40	-2.9	0	1015	94%	3	-4.9	-2.8	-1.0
ΔΕΝ	03:30	-2.3	2	1017	99%	10	-10.0	-2.4	0.0
ΔΩΚΟΣ	03:40	-6.0	2	1016	81%	11	-6.9	-6.1	0.0
ΔΟΛΟΥΣΑ	03:40	10.6	10	1017	51%	37	10.6	11.3	0.0
ΔΟΣΑΤΟ ΔΡΑΜΑΣ	03:40	-5.9	2	1015	99%	10	-6.3	-5.3	0.0
ΕΚΑΛΙΑ ΑΤΤΙΚΗΣ	03:40	2.3	2	1017	74%	13	1.4	2.4	0.0
ΕΛΙΕΥΒΕΡΙΟΥΠΟΛΗ	03:30	-5.0	0	1016	90%	10	-5.7	-5.3	0.0
ΕΠΙΣΚΑΛΩΝ	03:50	5.3	8	1014	82%	23	1.4	5.3	0.0
ΕΠΙΣΚΑΛΩΝ ΠΡΩΤΟΥ	03:40	6.8	15	1018	76%	34	5.1	7.1	0.4
ΖΑΓΟΡΑ ΠΗΛΙΟΥ	03:40	-3.0	3	1017	92%	11	-4.1	-3.1	0.0
ΖΑΚΥΝΘΟΣ	03:40	6.6	2	1012	93%	19	0.6	6.1	1.0
ΖΑΡΑΦΑΣ ΕΥΒΟΙΑΣ	03:40	0.1	0	1016	89%	18	-0.7	0.6	0.0
ΖΙΤΤΑ ΔΑΦΝΙΩΝ	03:40	-4.4	0	1016	92%	2	-4.8	-3.7	0.0
ΗΓΟΥΜΕΝΙΤΣΑ	03:20	-1.4	10	1014	92%	24	-1.3	0.4	0.0
ΗΡΑΚΛΕΙΑ ΚΥΚΛΑΔΩΝ	03:40	0.3	0	1017	61%	34	0.6	10.1	0.0
ΗΡΑΚΛΕΙΟ ΗΜΑΘΙΑΣ	03:30	5.7	10	1016	73%	24	0.0	5.7	0.0
ΗΡΑΚΛΕΙΟ ΗΜΑΘΙΑΣ	03:20	4.8	6	1019	76%	16	2.2	5.2	0.0
ΗΡΑΚΛΕΙΟ ΣΥΤΡΙΑΣ	03:40	5.9	0	1019	64%	26	4.0	6.1	0.0
ΘΑΛΟΣΣΑ ΗΜΑΘΙΑΣ	03:40	0.6	5	1014	85%	16	-0.3	0.8	0.0
ΘΕΟΔΩΡΟΠΟΛΗ ΑΡΦΑΣ	03:20	-5.6	0	1015	92%	6	-5.6	-5.6	0.0
ΘΕΟΔΩΡΟΣ ΦΩΚΙΑΣ	03:40	-2.1	0	1016	93%	5	-3.0	-2.4	0.0
ΘΕΣΣΑΛΩΝΗ	03:40	-3.8	13	1015	79%	26	-4.2	-3.2	0.0
ΘΥΡΑ	03:20	-4.9	5	1016	91%	15	-6.2	-4.7	0.0
ΚΑΡΑΙΣΤΑ	03:40	3.3	6	1019	76%	19	2.9	0.4	0.0
ΚΑΡΠΑΖ	03:50	4.8	3	1013	90%	11	2.4	4.7	1.6

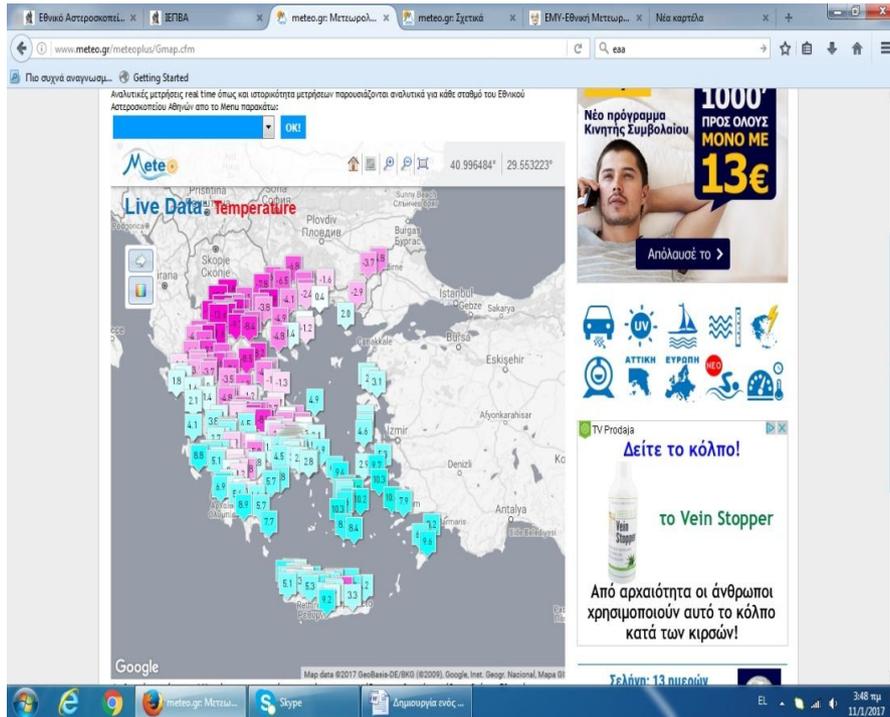
Εικόνα 19. Β\_Ενεργοί σταθμοί.

Station Name	Time	X	Y	Z	W	S	T	P	Q
ΠΕΡΙΟΥΛΗ	03:40	-7.6	3	1015	89%	28	-4.2	-4.4	0.4
ΠΕΤΡΑ ΛΕΙΒΟΣ	03:50	3.3	2	1014	83%	68	3.4	6.6	3.2
ΠΕΤΡΟΥΠΟΛΗ	03:00	2.7	0	1015	73%	15	2.6	3.6	0.0
ΠΗΜΕΛΑ ΗΜΑΘΙΑΣ	03:40	5.0	13	1012	69%	34	1.6	4.6	0.2
ΠΛΑΚΙΑΣ ΠΕΛΛΗΝΙΟΥ	03:40	5.7	0	1019	82%	21	4.6	7.6	0.0
ΠΛΑΤΑΝΙΑΣ ΚΑΙΝΩΝ	03:40	2.9	6	1017	86%	15	2.8	5.0	0.0
ΠΟΛΥΤΥΡΟΣ	03:10	-4.9	0	1016	86%	21	-5.6	-3.9	0.0
ΠΟΛΥΠΡΟΣΟΣ ΦΩΚΙΑΣ	03:40	-5.2	0	1017	92%	5	-7.4	-6.1	0.0
ΠΟΡΦΑΡΑ ΠΗΛΙΟΥ	03:40	-2.1	2	1015	66%	23	-2.8	-1.1	0.0
ΠΟΡΤΟ ΘΡΑΚΗΣ	03:40	2.9	3	1015	79%	21	3.1	5.3	0.0
ΠΟΤΑΜΟΣ ΠΕΛΛΗΝΙΟΥ	03:20	4.3	11	1019	96%	19	0.2	4.1	0.2
ΠΡΑΜΑΤΑ ΔΑΦΝΙΩΝ	03:40	-4.9	2	1017	93%	10	-5.5	-4.1	1.2
ΠΡΕΒΕΣΑ	03:40	0.9	15	1013	94%	32	0.9	2.8	0.4
ΠΤΟΛΕΜΑΙΔΑ	03:40	-14.3	0	1017	86%	2	-16.2	-14.4	0.0
ΠΤΟΛΕΜΑΙΔΑ 1η	03:40	-11.0	2	1018	83%	6	-14.5	-11.9	0.0
ΠΤΟΛΕΜΑΙΔΑ 2η	03:40	-11.1	3	1019	-	10	-12.9	-11.6	0.0
ΠΥΛΑΙΟΝ ΗΜΑΘΙΑΣ	03:40	3.3	0	1018	82%	8	2.5	5.1	0.0
ΠΥΡΡΟΣ ΗΜΑΘΙΑΣ	03:40	5.2	11	1012	65%	29	3.9	5.1	0.0
ΡΑΦΗΝΑ	03:40	-0.6	10	1015	94%	16	-0.7	1.1	0.0
ΡΕΦΥΜΙΟ	03:40	7.5	10	1015	67%	29	5.1	6.9	0.0
ΡΕΣΣΙΔΑ ΗΜΑΘΙΑΣ	03:40	-7.9	0	1017	84%	10	-12.4	-8.2	0.0
ΡΩ	03:40	4.8	0	1015	75%	19	3.7	5.2	0.0
ΡΩΣΣΟ ΠΟΛΗ	03:40	8.4	2	1020	76%	19	7.3	8.6	0.2
ΣΑΛΑΜΙΝΑ	03:20	4.3	2	1015	64%	24	4.3	7.5	0.0
ΣΑΜΟΝΑ	03:40	-0.6	11	1024	93%	27	-1.7	-0.7	0.0
ΣΑΜΟΝΑΚΟ	03:40	2.0	2	1014	86%	24	0.9	2.2	0.2
ΣΑΝΟΣ ΚΑΡΠΟΒΑΣΙ	03:40	4.9	10	1016	76%	27	5.3	8.7	0.0
ΣΑΠΤΟΡΡΗ	03:40	9.0	11	1016	65%	50	8.4	8.9	0.0
ΣΕΥΚΙΑ	03:40	-10.3	13	1018	76%	24	-12.2	-10.6	0.0

Εικόνα 20. Γ\_Ενεργοί σταθμοί.

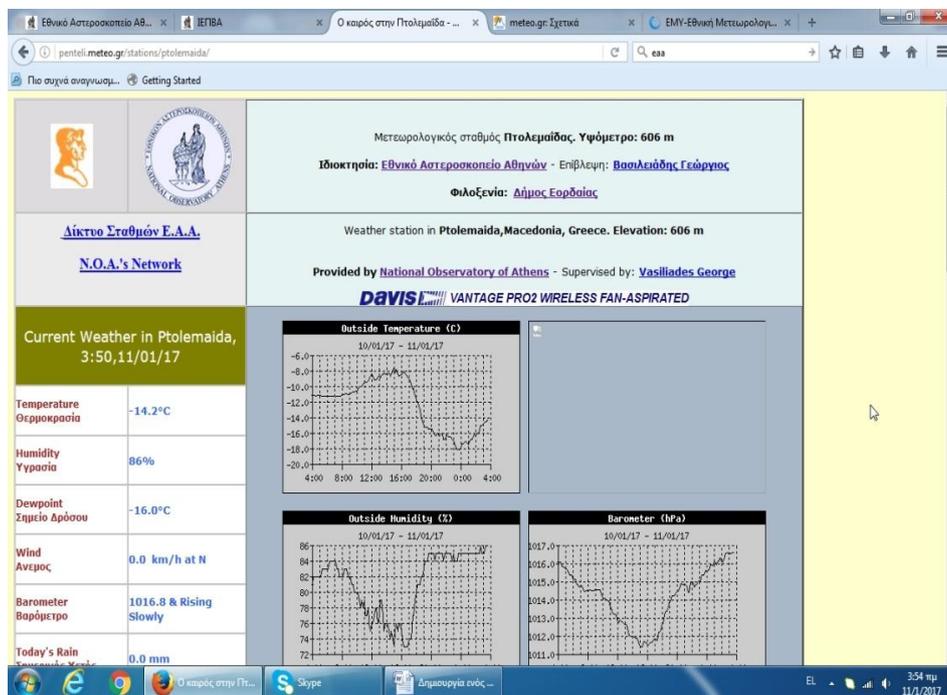
Επιλέγουμε την περιοχή που θέλουμε από τις επιλογές και εμφανίζονται τα στοιχεία. Στα στοιχεία εκτός από τις μετρήσεις, δίνονται και το γεωγραφικό πλάτος και μήκος της επιλεγμένης περιοχής.

# ΔΗΜΙΟΥΡΓΙΑ ΕΝΟΣ ΜΕΤΕΩΡΟΛΟΓΙΚΟΥ ΣΤΑΘΜΟΥ ΜΕ ΧΡΗΣΗ ΜΙΚΡΟΕΛΕΓΚΤΗ ΚΑΙ ΕΜΦΑΝΙΣΗ ΔΕΔΟΜΕΝΩΝ ΣΕ ΣΕΛΙΔΑ



Εικόνα 21. Απεικόνιση των σταθμών πάνω στον χάρτη της Ελλάδας.

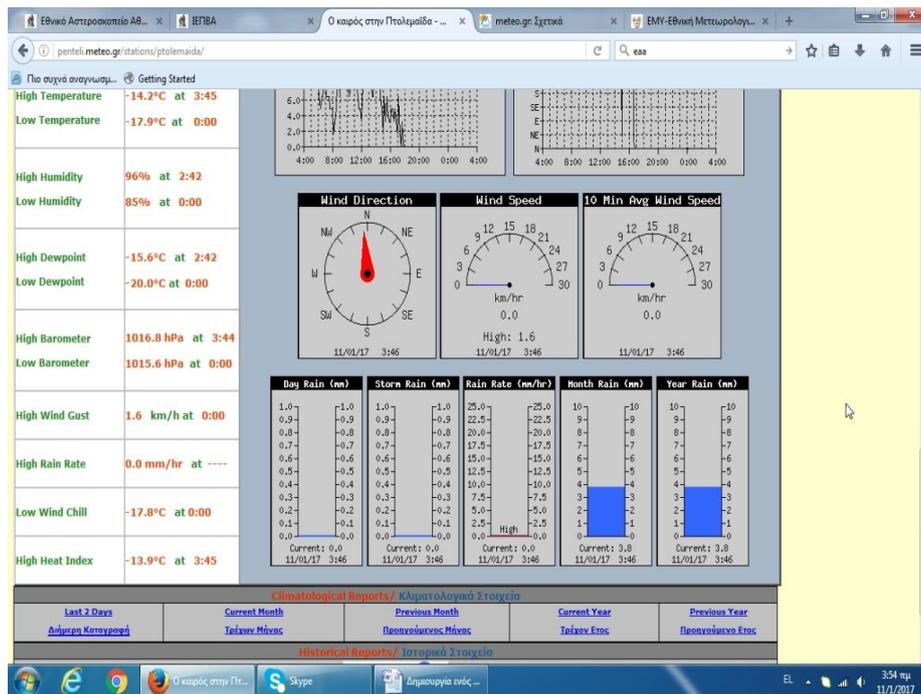
Στις παρακάτω εικόνες έγινε επιλογή της πόλης: ΠΤΟΛΕΜΑΪΔΑΣ.



Εικόνα 22. Οι τιμές στην Πτολεμαΐδα\_α.



Εικόνα 23. Οι τιμές στην Πτολεμαίδα\_β.

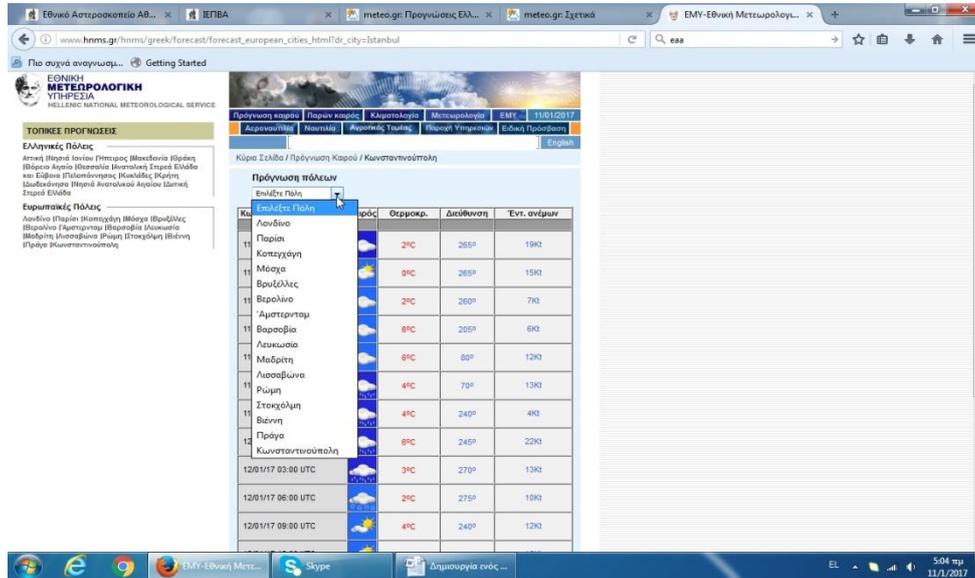


Εικόνα 24. Οι τιμές στην Πτολεμαίδα\_γ.

### 5.3 Μετεωρολογικοί Σταθμοί στον κόσμο

Η Εθνική Μετεωρολογική Υπηρεσία λαμβάνει μετρήσεις από ευρωπαϊκές πόλεις αλλά και από την Μόσχα και την Κωνσταντινούπολη. Οι επιλογές που δίνονται για μετρήσεις εκτός

Ελλάδος είναι πολύ περιορισμένες. Και παρατηρούμε ότι δεν δίνει πλήρεις πληροφορίες όπως δίνει η ιστοσελίδα του meteo.



Εικόνα 25. Οι επιλογές στις Χώρες στην σελίδα της EMY.

Η σελίδα του meteo δίνει σε παγκόσμιο χάρτη τις μετεωρολογικές μετρήσεις από σταθμούς εδάφους από διάφορες σημαντικές πόλεις του πλανήτη. Παρατηρούμε από τα στοιχεία που δίνονται, ότι όλοι οι σταθμοί είναι τοποθετημένοι σε αεροδρόμια, γεγονός που αναδεικνύει το σημαντικό ρόλο τους ώστε να πραγματοποιείται η ασφαλή μεταφορά του κόσμου που ταξιδεύει.



Εικόνα 26. Οι επιλογές στις Χώρες στην σελίδα του meteo.

# ΔΗΜΙΟΥΡΓΙΑ ΕΝΟΣ ΜΕΤΕΩΡΟΛΟΓΙΚΟΥ ΣΤΑΘΜΟΥ ΜΕ ΧΡΗΣΗ ΜΙΚΡΟΕΛΕΓΚΤΗ ΚΑΙ ΕΜΦΑΝΙΣΗ ΔΕΔΟΜΕΝΩΝ ΣΕ ΣΕΛΙΔΑ

Χώρα	Πόλη	Αεροδρόμιο	Ημερομηνία	Θερμότητα	Άνεμος	Πίεση	Φαινόμενα
Algeria	Tamanrasset	(Tamanrasset airport)	11/1/2017 04:00	3 °C	Calm	1012.9 mb	
American Samoa	Pago Pago	(Pago Pago Intl airport)	11/1/2017 03:50	32 °C	E (100 degrees) at 24 km/h	1009.8 mb	Few cumulus clouds at 520m, Scattered clouds at 5800m
Argentina	Ushuaia	(Ushuaia Malvinas Argentinas airport)	11/1/2017 04:00	4 °C	SW (200 degrees) at 43 km/h, 63 km/h gusts	989.2 mb	Drizzle Few clouds at 550m, Scattered clouds at 760m, Broken clouds at 1400m
Australia	Alice Springs	(Alice Springs airport)	11/1/2017 04:00	33 °C	E (90 degrees) at 11 km/h	1009.1 mb	
Australia	Hobart	(Hobart airport)	11/1/2017 04:00	21 °C	WNW (290 degrees) at 33 km/h, 52 km/h gusts	1002.0 mb	
Australia	Melbourne	(Melbourne Intl airport)	11/1/2017 04:00	22 °C	S (190 degrees) at 22 km/h	1010.8 mb	Scattered clouds at 1200m, Broken clouds at 1300m
Australia	Sydney	(Sydney Intl airport)	11/1/2017 04:00	36 °C	E (100 degrees) at 15 km/h	1004.1 mb	
Bahamas	Nassau	(Lynden Pindling Intl airport)	11/1/2017 04:00	21 °C	ENE (70 degrees) at 17 km/h	1026.4 mb	Few clouds at 610m, Scattered clouds at 1200m
Brazil	Belem	(Val De Cans Intl airport)	11/1/2017 04:00	23 °C	ESE (120 degrees) at 4 km/h	1010.8 mb	Scattered clouds at 300m, Broken clouds at 3000m

Εικόνα 27. Ενδεικτικά οι τιμές από αεροδρόμια κάποιων χωρών.

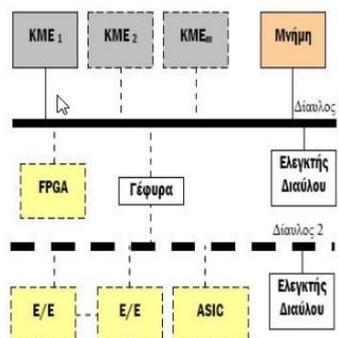
## 6 ΕΝΣΩΜΑΤΩΜΕΝΑ ΣΥΣΤΗΜΑΤΑ

Στο κεφάλαιο αυτό θα αναφερθούμε στα ενσωματωμένα συστήματα, στους μικροεπεξεργαστές, στους μικροελεγκτές, θα γίνει μία συνοπτική εξερεύνηση των μικροελεγκτών AVR που διαθέτει και ο Arduino αλλά και στις διαφορές που εντοπίζονται μεταξύ μικροελεγκτών και μικροεπεξεργαστών.

### 6.1 Ορισμός Ενσωματωμένων Συστημάτων

Ο σημερινός σύγχρονος τρόπος ζωής επιβάλλει την χρήση έξυπνων συσκευών, όπως τα κινητά τηλέφωνα, τους ελεγκτές αυτοκινήτων, τους φούρνους μικροκυμάτων και άλλες συσκευές, οι οποίες αποτελούνται από Ενσωματωμένα Συστήματα (embedded systems). Τα ενσωματωμένα αυτά συστήματα είναι υπολογιστικά συστήματα ειδικού σκοπού, με αρχιτεκτονική και αρχές λειτουργίας παρόμοιες με αυτές των συμβατικών υπολογιστών. Τα χαρακτηριστικά τους είναι το μικρό τους μέγεθος και η μικρή κατανάλωση ισχύος, η απόδοση σε συγκεκριμένες εφαρμογές και το χαμηλό τους κόστος. Η δομή τους βασίζεται σε έναν μικροεπεξεργαστή, ο οποίος είναι συνδεδεμένος μέσω μιας ιεραρχίας διαύλων με στοιχεία προσωρινής και μόνιμης αποθήκευσης, όπως μνήμες RAM, EEPROM, Flash. Δίνουν τη δυνατότητα να συνδέονται μαζί τους στοιχεία εξειδικευμένου υλικού, ώστε να επιτελούν καθορισμένες εργασίες για τις αντίστοιχες εφαρμογές. Ανάλογα με τις εφαρμογές μπορούν να χρησιμοποιηθούν ψηφιακά κυκλώματα που περιλαμβάνουν μία προγραμματιζόμενη λογική διάταξη (PLD, FPGA) ή ολοκληρωμένα κυκλώματα ειδικών εφαρμογών (ASIC, VLSI).

Για να γίνει καλύτερη η κατανόηση της έννοιας του ενσωματωμένου συστήματος θα πρέπει να σημειωθεί ότι είναι κάθε ψηφιακό σύστημα το οποίο εκτελεί μια συγκεκριμένη λειτουργία και δεν έχει καμία σχέση με τον υπολογιστή. Το ενσωματωμένο σύστημα χρησιμοποιείται για εξειδικευμένες εφαρμογές και έχει περιορισμένη χρήση.



Εικόνα 1 Γενικό διάγραμμα αρχιτεκτονικής δομής Ενσωματωμένου Συστήματος

### Διάγραμμα 2. Η δομή των Ενσωματωμένων Συστημάτων.

Η χρήση τους εντοπίζεται σε όλες τις ηλεκτρικές και ηλεκτρονικές συσκευές που χρησιμοποιούμε καθημερινά, από τον εκτυπωτή έως τα ηλεκτρονικά παιχνίδια και τα ηλεκτρονικά όργανα μετρήσεων ακρίβειας. Διαπιστώνεται η αναγκαιότητά τους στην βελτίωση της καθημερινότητάς μας που έχει σαν αποτέλεσμα την συνεχόμενη βελτίωση των ενσωματωμένων συστημάτων που εσωκλείονται μέσα στις συσκευές καθημερινής χρήσης. Τα ενσωματωμένα συστήματα της τελευταίας γενιάς ανταποκρίνονται στις αυξημένες αυτοματοποιημένες απαιτήσεις που υπάρχουν, για αυτό και καλούνται διαδίκτυο πραγμάτων (Internet of Things).

Ένα ενσωματωμένο σύστημα λοιπόν, έχει την δυνατότητα να αλληλεπιδρά με τον πραγματικό κόσμο και σε πραγματικό χρόνο, αφού απαιτείται η επικοινωνία μεταξύ τους να πραγματοποιείται σε καθορισμένα χρονικά διαστήματα, να συλλέγει πληροφορίες και να ανταποκρίνεται σε αυτές.

Η λειτουργία των ενσωματωμένων συστημάτων βασίζεται σε αισθητήρες από τους οποίους καταγράφει κάθε χρήσιμη πληροφορία για την σωστή λειτουργία τους. Έτσι, οι πληροφορίες που θα μπορούσε να συλλέξει ένα ενσωματωμένο σύστημα θα μπορούσαν να αφορούν τη θερμοκρασία, την πίεση, την υγρασία, την ταχύτητα ή την επιτάχυνση, αυτά θα αποτελούσαν μετρήσιμες τιμές κάποιων αισθητήρων. Υπάρχουν περιπτώσεις που γίνεται χρήση σύνθετων αισθητήρων, όπως κάμερες ή υπερηχητικοί αισθητήρες απόστασης. Η τεχνολογία επιτρέπει πλέον τη δημιουργία αισθητήρων με τη μορφή σύνθετων ηλεκτρομηχανικών συστημάτων (MEMS-micro-electro-mechanical systems) που αναπτύσσονται σε τσιπ πυριτίου και ενσωματώνουν και κυκλώματα μετατροπής του αναλογικού σήματος σε ψηφιακό ή και μονάδες επεξεργασίας. Τέτοια συστήματα ονομάζονται «έξυπνοι αισθητήρες».

Σήμερα, πλέον η 'επαφή' με τα ενσωματωμένα συστήματα μπορεί να γίνει και ερασιτεχνικά μέσα από μια μεγάλη ποικιλία εκπαιδευτικών μικροελεγκτών, όπως είναι ο Arduino ή με την χρήση των Lego Mindstorms. (Καλόμοιρος, 2015)

## 6.2 Χαρακτηριστικά Ενσωματωμένων Συστημάτων

Τα ενσωματωμένα συστήματα διαθέτουν τα εξής βασικά χαρακτηριστικά:

- ✓ Εκτελούνται συνεχώς και η εκτέλεσή τους δεν τερματίζεται ποτέ.
- ✓ Οι αλληλεπιδράσεις με το περιβάλλον του συστήματος είναι απρόβλεπτες και μη ελεγχόμενες.
- ✓ Σε κάποιες περιπτώσεις υπάρχουν φυσικοί περιορισμοί όπως, το ηλεκτρικό ρεύμα, το μέγεθος, το βάρος που επηρεάζουν το σχεδιασμό του συστήματος.
- ✓ Απαραίτητη κρίνεται η άμεση αλληλεπίδραση με το υλικό του συστήματος.
- ✓ Η ασφάλεια και η αξιοπιστία αποτελούν βασικά στοιχεία στο σχεδιασμό του συστήματος. (Δουμένης, 2015)

Μπορούμε να διακρίνουμε τα ενσωματωμένα συστήματα με κριτήριο την τεχνολογία τους. Ωστόσο, όλες οι τεχνολογίες αλληλοεπικαλύπτονται μεταξύ τους. Έτσι, υπάρχουν τρεις τεχνολογίες:

### 1. Τεχνολογία επεξεργαστών:

α. Επεξεργαστές γενικού σκοπού: Είναι προγραμματιζόμενες διατάξεις ικανές να ανταποκριθούν στις ανάγκες πολλών εφαρμογών. Έχουν μεγάλη RAM (εσωτερική ή εξωτερική), και συνήθως στηρίζονται σε αρχιτεκτονική Von-Neumann και ένα γενικό διάδρομο επεξεργασίας (data path). Παραδείγματα: Pentium, 8088, 68000...

β. Εξειδικευμένοι επεξεργαστές (application-specific): Είναι μία κατηγορία προγραμματιζόμενων επεξεργαστών που είναι βελτιστοποιημένοι για συγκεκριμένη κατηγορία εφαρμογών. Έχουν περιορισμένη RAM πάνω στο τσιπ, συνήθως υλοποιούν αρχιτεκτονική Harvard, η επεξεργασία τους είναι ειδικού σκοπού και έχουν ενσωματωμένα πολλά περιφερειακά

κυκλώματα. Επίσης, έχουν μνήμη προγράμματος. Παραδείγματα: microcontrollers, DSP processors.

γ. Επεξεργαστές μοναδικού σκοπού (single purpose): Είναι ψηφιακά κυκλώματα σχεδιασμένα να εκτελούν μόνον μία διεργασία. Περιλαμβάνουν μόνον τα απαραίτητα κυκλώματα που χρειάζονται ώστε να εκτελέσουν ένα μοναδικό πρόγραμμα. Δεν έχουν μνήμη προγράμματος, αλλά σε ορισμένες περιπτώσεις χρειάζονται λίγη RAM. Παραδείγματα: επιταχυντές υλικού, όπως video compressors, ψηφιακά φίλτρα, επεξεργαστές μηχανικής όρασης. Οι περιφερειακοί ελεγκτές (κυκλώματα επικοινωνίας με θύρες I/O και με ICs) θεωρούνται επίσης επεξεργαστές μοναδικού σκοπού.

## 2. Τεχνολογία Ολοκληρωμένων κυκλωμάτων:

α. Πλήρως προσαρμοσμένα (Full custom/VLSI). Όλα τα στρώματα του κυκλώματος σχεδιάζονται από την αρχή, από τα τραζίστορ μέχρι τις διασυνδέσεις.

β. Μερικώς προσαρμοσμένα (Semi-custom ASIC). Τα χαμηλότερα στρώματα είναι μερικώς σχεδιασμένα. Η σχεδίαση γίνεται με τοποθέτηση και διασύνδεση προσχεδιασμένων βαθμίδων.

γ. Διαμορφούμενες διατάξεις PLD (CPLDs/FPGAs). Όλα τα στρώματα υπάρχουν εξ αρχής μέσα στο τσιπ. Η σχεδίαση γίνεται με τη διαμόρφωση των πινάκων αναφοράς (LUT) και των προγραμματιζόμενων διασυνδέσεων.

## 3. Τεχνολογία σχεδίασης:

Για το λογισμικό διαθέτουμε ολοκληρωμένα περιβάλλοντα για ανάπτυξη λογισμικού, όπως assemblers, μεταγλωττιστές, προσομοιωτές και προγραμματιστές.

Για το υλικό μέρος διαθέτουμε εργαλεία σχεδίασης τύπου CAD και γλώσσες περιγραφής υλικού. Τα εργαλεία αυτά επιτρέπουν τη σχεδίαση σε διάφορα επίπεδα:

- στο επίπεδο του ολοκληρωμένου συστήματος (system level)
- Στο επίπεδο της συμπεριφοράς του κυκλώματος (behavioral level)
- Στο δομικό επίπεδο της μεταφοράς σημάτων ανάμεσα σε πύλες και καταχωρητές (Register - transfer level). (Καλόμοιρος, 2015)

### 6.3 Μικροεπεξεργαστές

Ένας μικροεπεξεργαστής αποτελεί μια μονάδα κεντρικής επεξεργασίας (CPU) υλοποιημένη μέσα σε ένα ολοκληρωμένο κύκλωμα. Τα δεδομένα επεξεργάζονται με μια σειρά από εντολές. Αυτές οι εντολές που εκτελεί ο μικροεπεξεργαστής είναι εντολές γραμμένες σε γλώσσα μηχανής και αποθηκευμένες στην κύρια μνήμη. Μια εντολή σε γλώσσα μηχανής είναι μια σειρά από δυαδικά ψηφία, όπου είναι κωδικοποιημένο το είδος της εντολής. Στο σύνολό τους οι εντολές αυτές χρησιμεύουν ως μία διασύνδεση ανάμεσα στο λογισμικό (software) και το υλικό (hardware), δηλαδή ανάμεσα στα προγράμματα και στους επεξεργαστές. Αρκετά χρόνια πριν, μια τέτοια μονάδα επεξεργασίας -CPU- σχεδιάζονταν με τη βοήθεια πολλών διακριτών ολοκληρωμένων κυκλωμάτων μεσαίας και μεγάλης κλίμακας ολοκλήρωσης (MSI, LSI). Η εταιρεία Intel, παρουσιάζοντας το 4004, έθεσε όλες τις απαραίτητες μονάδες που περιλαμβάνει τυπικά μια CPU, όπως την αριθμητική λογική μονάδα ή ALU, τον αποκωδικοποιητή εντολών, τους καταχωρητές, κυκλώματα ελέγχου διαύλων, εντός ενός απλού ολοκληρωμένου κυκλώματος (chip) και έτσι δημιουργήθηκε ο μικροεπεξεργαστής. Ο 4004 ήταν ένας απλός μικροεπεξεργαστής των 4 δυαδικών ψηφίων, δηλαδή είχε την δυνατότητα επεξεργασίας δεδομένων υπό την μορφή

λέξεων των 4-bits τη φορά, ο οποίος σχεδιάστηκε για να χρησιμοποιηθεί ως μονάδα επεξεργασίας αριθμών σε μια αριθμομηχανή.(Gadre, 2001)

Ο μικροεπεξεργαστής αποτελείται από τις ακόλουθες μονάδες:

- **Μονάδα αποκωδικοποίησης** (Decoding Unit)
- **Αριθμητική και Λογική Μονάδα** (Arithmetic and Logical Unit, ALU): Στη μονάδα αυτή εκτελούνται όλες μία προς μία οι αριθμητικές ή λογικές πράξεις, όπως έχουν γραφτεί σύμφωνα με τις εντολές που έχουν δοθεί στον υπολογιστή.
- **Καταχωρητές** (Registers): Αποτελούν μικρά κελιά μνήμης στο εσωτερικό του επεξεργαστή, που χρησιμοποιούνται για την προσωρινή αποθήκευση των δεδομένων, καθώς αυτά υπόκεινται σε επεξεργασία. Κατηγοριοποιούνται ανάλογα με τον τύπο του επεξεργαστή και τον κατασκευαστή, με βάση την οργάνωση και τη χωρητικότητά τους.
- **Μονάδα ελέγχου** (Control Unit): Η μονάδα αυτή στόχο έχει να ελέγξει τη ροή δεδομένων από και προς την ALU, τους καταχωρητές, τη μνήμη και τις περιφερειακές μονάδες εισόδου/εξόδου.
- **Μονάδα προσκόμισης** (Fetch Unit): Οι εντολές μεταφέρονται από τη μνήμη στον επεξεργαστή.
- **Μονάδα προστασίας** (Protection Unit): Ελέγχεται η κάθε διεργασία που εκτελεί ο μικροεπεξεργαστής, για να μην αλλάζουν δεδομένα που δεν πρέπει ή να μην εκτελούνται μη αποδεκτές εντολές, όπως, η διαίρεση αριθμού με το μηδέν.

## 6.4 Μικροελεγκτές

### 6.4.1 Γενικά

Όταν ένας μικροεπεξεργαστής, μαζί με τα απαραίτητα κυκλώματα υποστήριξης (περιφερειακές μονάδες εισόδου-εξόδου και μνήμη προγράμματος και δεδομένων), τοποθετηθούν μαζί στο ίδιο υλικό, έτσι ώστε να σχηματίσουν έναν στοιχειώδη υπολογιστή, ειδικότερα σε εφαρμογές ελέγχου και λήψης δεδομένων, η διάταξη που προκύπτει καλείται μικροϋπολογιστής. Όταν όλες οι παραπάνω διακριτές μονάδες που αποτελούν έναν μικροϋπολογιστή τοποθετηθούν μέσα στο σώμα του ίδιου ολοκληρωμένου κυκλώματος (chip), η διάταξη που προκύπτει καλείται μικροελεγκτής.

Ο όρος μικροελεγκτής αναφέρεται σε ένα ολοκληρωμένο κύκλωμα με ενσωματωμένες περιφερειακές μονάδες (θύρες, χρονιστές, σειριακές θύρες επικοινωνίας, κ.λπ.) και μονάδες μνήμης, εκτός από την κεντρική μονάδα επεξεργασίας (CPU).

Επομένως, διαπιστώνουμε ότι οι μικροελεγκτές αποτελούν την ‘καρδιά’ πολλών καθημερινών συσκευών. Κάθε μικροελεγκτής περιέχει μέσα σε ένα και μοναδικό ολοκληρωμένο κύκλωμα τα παρακάτω συστατικά:

- Μονάδα κεντρικής επεξεργασίας(CPU): Η ‘καρδιά’ ενός μικροελεγκτή.
- Μνήμη προγράμματος: Εδώ αποθηκεύονται οι εντολές που σχηματίζουν τον κορμό του προγράμματος(EEPROM,EPROM,ROM,Flash).
- Μνήμη RAM: Μνήμη τυχαίας προσπέλασης.
- Ταλαντωτής χρονισμού: Ο ρυθμός εκτέλεσης του προγράμματος.
- Σύστημα επανατοποθέτησης και Κύκλωμα ανίχνευσης βυθίσεων τάσης: Με το πρώτο υλοποιείται μηδενισμός(reset) ενώ στον δεύτερο διαπιστωθεί βύθιση τάσης θέτει τον μικροελεγκτή σε λειτουργία επανατοποθέτησης.
- Σειριακή θύρα επικοινωνίας: Ασύγχρονη σειριακή θύρα και Σύγχρονη σειριακή θύρα.
- Ψηφιακή θύρα εισόδου-εξόδου: Ανταλλαγή δεδομένων από και προς το εξωτερικό περιβάλλον με την μορφή ενός bit τη φορά.

- Αναλογική θύρα εισόδου-εξόδου: Μετατροπείς αναλογικού σήματος σε ψηφιακό, χρησιμοποιούνται για την ανάγνωση δεδομένων από αισθητήρες (θερμοκρασίας, κ.ά.). Μετατροπείς ψηφιακού σήματος σε αναλογικό, χρησιμοποιούνται για την οδήγηση κινητήρων, αναπαραγωγή σημάτων ήχου, κ.α.
- Χρονιστής επιτήρησης: Αποφυγή πιθανής κατάρρευσης του συστήματος.
- Ρολόι πραγματικού χρόνου: Σηματοδότηση συγκεκριμένων γεγονότων βάση την τρέχουσα ώρα (RTC).

Επίσης, αποτελείται και από τα εξής στοιχεία:

- έναν αριθμό από καταχωρητές ειδικού σκοπού (συσσωρευτή, καταχωρητή κατάστασης, μετρητή προγράμματος, καταχωρητή εντολών, καταχωρητή δείκτη).
- εσωτερικούς χρονιστές - απαριθμητές.
- αριθμητική και λογική μονάδα (ALU).
- μονάδα αποκωδικοποίησης εντολών.

Μια συγκεκριμένη οικογένεια μικροελεγκτών προσδιορίζει ένα σύνολο πολλών διαφορετικών μικροελεγκτών, οι οποίοι διαθέτουν τον ίδιο πυρήνα αλλά κάθε ένας από αυτούς διατίθεται σε διαφορετική συσκευασία, περιλαμβάνει διαφορετικές περιφερειακές μονάδες, λειτουργεί σε διαφορετικές ταχύτητες, κ.λπ. (Gadre, 2001)

Οι μικροελεγκτές βρίσκουν εφαρμογή στα παρακάτω πεδία:

- Σε συστήματα αυτοματισμών
- Σε κυκλώματα τηλεπικοινωνιών
- Στις ηλεκτρονικές συσκευές
- Στις ηλεκτρικές συσκευές
- Σε συστήματα τηλεματικής
- Σε συστήματα συλλογής δεδομένων (Data Acquisition)
- Σε εφαρμογές ηλεκτρονικών ισχύος
- Σε συστήματα διασύνδεσης
- Σε εφαρμογές δικτύων

Γενικότερα οι μικροελεγκτές χρησιμοποιούνται οπουδήποτε απαιτείται έλεγχος συστημάτων.

Υπάρχουν δεκάδες εταιρείες παγκοσμίως που κατασκευάζουν μικροελεγκτές. Οι πιο διαδεδομένες είναι:

- Microchip, [www.microchip.com](http://www.microchip.com)
- Atmel, [www.atmel.com](http://www.atmel.com)
- Texas Instruments, [www.ti.com](http://www.ti.com)
- Freescale (πρώην Motorola), [www.freescale.com](http://www.freescale.com)
- Intel, [www.intel.com](http://www.intel.com)
- Analog Devices, [www.analog.com](http://www.analog.com)

Οι περισσότερες εταιρείες παράγουν μεγάλη γκάμα μικροελεγκτών. Από πολύ μικρούς και φθηνούς για απλές εφαρμογές έως ιδιαίτερα προηγμένους για πολύ απαιτητικές εφαρμογές.

## 6.4.2 Κατηγορίες Μικροελεγκτών

Η ταξινόμηση των μικροελεγκτών μπορεί να γίνει με κριτήρια όπως, το εύρος διαύλου, το σετ εντολών, την αρχιτεκτονική μνήμης, την γλώσσα προγραμματισμού που χρησιμοποιείται κ.ά. Στο σχεδιάγραμμα φαίνεται η κατηγοριοποίηση των μικροελεγκτών.(Gadre, 2001)



Διάγραμμα 3. Κατηγορίες μικροελεγκτών.

## 6.4.3 Μικροελεγκτές 8,16 και 32-bit

Μία μνήμη αποτελείται από  $m$  λέξεις των  $k$  bit, αυτές οι λέξεις θα πρέπει να διευθυνσιοδοτηθούν ώστε να διαβάσει στην έξοδο δεδομένων της μνήμης, τη λέξη. Τα σήματα εισόδου διευθύνσεων  $\log_2(m)$  είναι απαραίτητα για καθορίσουν μι συγκεκριμένη λέξη. Όσες περισσότερες γραμμές διευθύνσεων έχει ένας διάυλος, τόσο περισσότερη μνήμη μπορεί να προσπελάσει άμεσα ο μικροεπεξεργαστής. Αν ο διάυλος έχει  $n$  γραμμές διευθύνσεων, τότε ο μικροεπεξεργαστής μπορεί να τον χρησιμοποιήσει για να απευθύνεται σε  $2^n$  διαφορετικές διευθύνσεις μνήμης. Για να μπορούν να χρησιμοποιηθούν μεγάλες μνήμες, οι διάυλοι χρειάζεται να έχουν πολλές γραμμές διευθύνσεων. Σύμφωνα με το εύρος του διαύλου διακρίνονται σε μικροελεγκτές των 8, 16 και 32-bit αντίστοιχα.

**Μικροελεγκτής 8-bit:** Όταν το εύρος του διαύλου είναι 8-bit και η μονάδα αριθμητικής λογικής (ALU) εκτελεί τις αριθμητικές και λογικές πράξεις σε ένα Byte (8 bit), τότε ο μικροελεγκτής ονομάζεται μικροελεγκτής 8-bit, όπως ο ATmega 328, ο Intel 8031/8051 και η σειρά MC68HC11 της Motorola.

**Μικροελεγκτής 16-bit:** Χαρακτηρίζεται από μεγάλη ακρίβεια και καλή απόδοση. Όταν το εύρος του διαύλου είναι 16-bit και η μονάδα αριθμητικής λογικής εκτελεί αριθμητικές και λογικές πράξεις μίας λέξης (16-bit) σε μία εντολή, τότε ο μικροελεγκτής ονομάζεται μικροελεγκτής 16-bit. Για παράδειγμα ένας μικροελεγκτής 8-bit μπορεί να προσπελάσει  $2^8$  θέσεις μνήμης, καταλήγοντας σε συνολικό εύρος  $0 \times 00 - 0 \times FF$  (0-255) για κάθε κύκλο. Αντίθετα, ένας μικροελεγκτής με μέγεθος 16-bit έχει εύρος  $2^{16}$ , δηλαδή  $0 \times 0000 - 0 \times FFFF$  (0-65535) ανά κύκλο. Παραδείγματα μικροελεγκτών 16-bit είναι το 8051XA, PIC2X, ο Intel8096,η σειρά MC68HC12 της Motorola.

Μικροελεγκτής 32-bit: Οι εντολές των 32-bit εκτελούν αριθμητικές και λογικές πράξεις. Είναι πιο βελτιωμένοι στην απόδοση και στην ακρίβεια. Χρησιμοποιούνται από συσκευές αυτόματου ελέγχου, όπως σε εφαρμογές συστημάτων ελέγχου ενός κινητήρα, σε μηχανές γραφείου, σε κινητά τηλέφωνα, ηχοσυστήματα MP3, αεροδιαστημικά συστήματα κ.α. Παραδείγματα μικροελεγκτών 32-bit, είναι η σειρά 32-bit AVR UC3 της Atmel, το PIC3x, το Motorola M683xx και η οικογένεια ARM 7, 9 και 11. (Βούλγαρης, 1982)

#### 6.4.4 Μικροελεγκτές Εσωτερικής και Εξωτερικής Μνήμης

Η μνήμη στους μικροελεγκτές μπορεί να διακριθεί σε εσωτερική και εξωτερική μνήμη. Τα ενσωματωμένα συστήματα που έχουν μία μονάδα μικροελεγκτή και όλα τα λειτουργικά τμήματα όπως η μνήμη για την αποθήκευση των δεδομένων αλλά και του προγράμματος διαθέσιμα σε ένα τσιπ, ονομάζονται μικροελεγκτές εσωτερικής μνήμης. Για παράδειγμα, ο 8051 έχει την μνήμη των δεδομένων και του προγράμματος, τις εισόδους και εξόδους, τη σειριακή επικοινωνία, τους μετρητές και timers και την ICL (Interrupt Control Logic) όλα σε ένα chip.

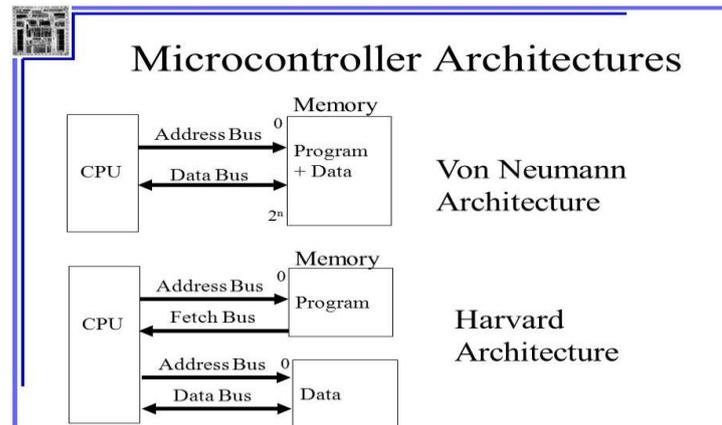
Ενσωματωμένα συστήματα που έχουν μία μονάδα μικροελεγκτή που δεν έχει όλα τα διαθέσιμα λειτουργικά τμήματα σε ένα chip, ονομάζεται μικροελεγκτής με εξωτερική μνήμη. Σε αυτούς, το σύνολο ή κάποιο μέρος των μονάδων μνήμης τους είναι διασυνδεδεμένα εξωτερικά χρησιμοποιώντας το λεγόμενο «glue Circuit». Για παράδειγμα ο 8031 που δεν έχει ενσωματωμένη μνήμη για την αποθήκευση του προγράμματος, είναι μικροελεγκτής εξωτερικής μνήμης. (Gadre, 2001)

#### 6.4.5 Αρχιτεκτονική Von Neumann και Harvard

Οι δύο αρχιτεκτονικές μικροελεγκτών είναι η Von Neumann και η Harvard, οι οποίες χρησιμοποιούν κοινή και ξεχωριστή μνήμη προγράμματος και δεδομένων αντίστοιχα.

Στην αρχιτεκτονική του Von Neumann διατίθεται μία κοινή μνήμη για την καταχώρηση των εντολών προγράμματος και δεδομένων, ενώ στην αρχιτεκτονική Harvard η κεντρική μονάδα επεξεργασίας ανακτά τις εντολές από τη μνήμη του προγράμματος και ανακτά ή καταχωρεί τα δεδομένα από τη μνήμη των δεδομένων. Στην Harvard η προσπέλαση εντολών-δεδομένων γίνεται ταυτόχρονα, αυξάνοντας έτσι την απόδοση του συστήματος.

Ο Von Neumann είναι η βάση όλων σχεδόν των ψηφιακών υπολογιστών, ακόμα και σήμερα. Τα βασικά του τμήματα είναι: η μνήμη, η αριθμητική λογική μονάδα, η μονάδα ελέγχου και ο εξοπλισμός εισόδου και εξόδου. Η μνήμη αποτελείται από 4096 λέξεις και η κάθε λέξη έχει 40-bit με τιμή 0 ή 1. Η κάθε λέξη περιέχει είτε δύο εντολές των 20-bit είτε ένα προσημασμένο ακέραιο των 40-bit. Οι εντολές χρησιμοποιούσαν 8-bit για τον προσδιορισμό του τύπου της εντολής και 12-bit για τον προσδιορισμό μίας από τις 4096 λέξεις της μνήμης. Στις αρχιτεκτονικές Von Neumann η μνήμη έχει έναν ενιαίο χώρο φυσικών διευθύνσεων. Η σειριακή αυτή προσπέλαση των δεδομένων και των εντολών δημιουργεί συμφόρηση και καθιστά τον ελεγκτή πιο αργό.



Διάγραμμα 4. Οι αρχιτεκτονικές Von Neumann και Harvard.

Η μνήμη που χωρίζεται σε επιμέρους κομμάτια μπορεί να επιτρέψει τις παράλληλες προσπελάσεις, ενώ μία ενοποιημένη δεν τις επιτρέπει. Επίσης αφού οι εντολές κανονικά δεν τροποποιούνται κατά την εκτέλεση, το περιεχόμενο της κρυφής μνήμης εντολών δεν χρειάζεται ποτέ να ξανά γράφεται πίσω στην κύρια μνήμη. Τα ολοκληρωμένα Atmega328 και Atmega2560 που χρησιμοποιούνται στην πλατφόρμες του Arduino βασίζονται στην αρχιτεκτονική Harvard. (Gadre, 2001)

#### 6.4.6 Αρχιτεκτονική CISC και RISC

Ένα κριτήριο για να διακρίνουμε τους μικροελεγκτές είναι η πολυπλοκότητα των εντολών που διαχειρίζεται η κεντρική μονάδα επεξεργασίας. Έχουμε λοιπόν, την αρχιτεκτονική RISC και CISC.

Η αρχιτεκτονική CISC (Αρχιτεκτονική Σύνθετου Ρεπερτορίου Εντολών) χαρακτηρίζει τις κεντρικές μονάδες επεξεργασίας με δυνατότητα εκτέλεσης πολύ πολύπλοκων λειτουργιών ανά εντολή μηχανής. Ενώ, η αρχιτεκτονική RISC (Αρχιτεκτονική Μειωμένου Ρεπερτορίου Εντολών) αφορά τις κεντρικές μονάδες επεξεργασίας με απλούστερες λειτουργίες ανά εντολή μηχανής. Τα τελευταία χρόνια οι εταιρείες προσανατολίζονται στην κατασκευή μικροελεγκτών τύπου RISC, σε γλώσσα υψηλού προγραμματισμού, όπως είναι οι PIC της Microchip και οι AVR της Atmel, οι οποίοι είναι αρχιτεκτονικής τύπου Harvard. Υπάρχουν όμως και οι HC08 της Motorola τύπου CISC και είναι αρχιτεκτονικής Von Neumann.

Κάποιες από τις διαφορές τους είναι οι εξής:

- Ο RISC χρησιμοποιεί πολλούς καταχωρητές σε αντίθεση με τους λίγους καταχωρητές του CISC.
- Ο RISC χρησιμοποιεί λίγες διευθύνσεις, ενώ ο CISC πολλές διευθύνσεις μνήμης.

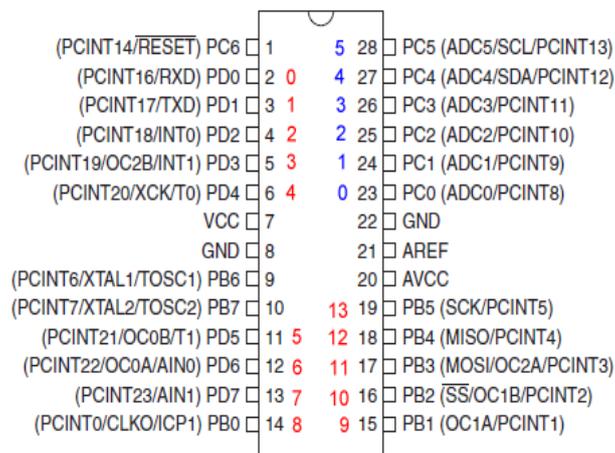
Ο RISC δίνει έμφαση στο software, ενώ ο CISC στο hardware.(Gadre, 2001)

### 6.4.7 Οι Μικροελεγκτές AVR

Δύο φοιτητές από τη Νορβηγία, οι A. Bogen και V. Wollan ανέπτυξαν την αρχιτεκτονική των AVR, κατά την διάρκεια της πρακτικής τους στο Νορβηγικό Ινστιτούτο Τεχνολογίας. Η δημιουργία τους αυτή ήταν γνωστή ως Micro RISC και την διέθεταν σε μορφή πυριτίου. Αυτό το γεγονός δεν πέρασε απαρατήρητο από τους ανθρώπους της μεγάλης εταιρείας Atmel και τους προσέγγισε για την εξαγορά της δημιουργίας τους. Από την στιγμή που αγοράστηκε η δημιουργία τους οι Νορβηγοί άρχισαν να δουλεύουν ακόμη πιο εντατικά για την βελτίωση της εσωτερικής αρχιτεκτονικής της AVR. Η εταιρία τους βοηθάει και τους συνδράμει σε αυτήν τους την προσπάθεια, παρέχοντάς τους όσα χρειάζονται για να επιτύχουν τον στόχο τους, ο οποίος ήταν ότι ο AVR μπορεί να προγραμματιστεί από ένα ρεπερτόριο εντολών γλώσσας υψηλού επιπέδου. Η πρώτη παρτίδα AVR ήταν ο AT90S8515, ο οποίος περιελάμβανε 40 ακίδες, είχε αρκετά στοιχεία ίδια με τον μικροελεγκτή 8051. Το 1997 εμφανίζεται ο μικροελεγκτής AVR 8-bit, από εκεί και μετά η εξέλιξή τους είναι συνεχόμενη. Το 2005 η πλατφόρμα του Arduino χρησιμοποιεί τον 8-bit μικροελεγκτή της Atmel, προωθώντας την εταιρεία να συνεχίζει η βελτίωση για την κάλυψη των όλο και περισσότερων αναγκών των πελατών της.

Ο μικροελεγκτής AVR περιλαμβάνει ένα 8-bit επεξεργαστή RISC αρχιτεκτονικής Harvard. Η αρχιτεκτονική αυτή δηλώνει ότι η κεντρική μονάδα επεξεργασίας (CPU) επικοινωνεί ταυτόχρονα με μία μνήμη προγράμματος και μία ξεχωριστή μνήμη δεδομένων. Όλοι οι μικροελεγκτές αυτής της σειράς διαθέτουν μία ιδιαιτερότητα που τους ξεχωρίζει η ενσωματωμένη μνήμη ταχείας αποθήκευσης, Flash λειτουργεί ως μνήμη προγράμματος. Οι μικροελεγκτές διακρίνονται σε μία ποικιλία από διαφορετικά μεγέθη και βαθμό πολυπλοκότητας, έτσι υπάρχουν οι 4, 8, 16, 32 ή 65-bit. Η σημαντική αυτή διάκριση αναφέρεται στο μήκος των εσωτερικών καταχωρητών και του εσωτερικού συσσωρευτή. Όταν γίνεται χρήση ενός 8-bit σημαίνει ότι η εσωτερική CPU συνδέεται με τις διάφορες μονάδες του συστήματος μέσα από έναν εσωτερικό δίαυλο δεδομένων μήκους 8-bit. Ακόμη και μεγαλύτερου μήκους δίαυλοι δουλεύουν σπάζοντας τα δεδομένα σε 8-bit.

Χαρακτηριστικό των chip της σειράς AVR, από το ATtiny15 μέχρι το Atmega328 περιλαμβάνουν το μέγεθος της μνήμης Flash στο όνομα τους. Αυτό σημαίνει ότι ο διαθέσιμος χώρος για το πρόγραμμα μπορεί να είναι από 1 KB έως 32 KB. Ο Arduino Uno R3 διαθέτει τον μικροελεγκτή Atmega328, ο οποίος έχει 32 KB (0.5 KB χρησιμοποιούνται από τον Bootloader) μνήμη Flash, το Arduino Mega2560 διαθέτει τον μικροελεγκτή Atmega2560, ο οποίος έχει 256 KB μνήμη flash (8 KB χρησιμοποιούνται από τον Bootloader). Ο περιορισμός αυτός ίσως εμφανίσει δυσκολία σε ιδιαίτερα πολύπλοκες εφαρμογές με πολλές γραμμές κώδικα. (Gadre, 2001)



Εικόνα 28. Ο Atmega32.

#### 6.4.8 Τα χαρακτηριστικά των AVR της Atmel

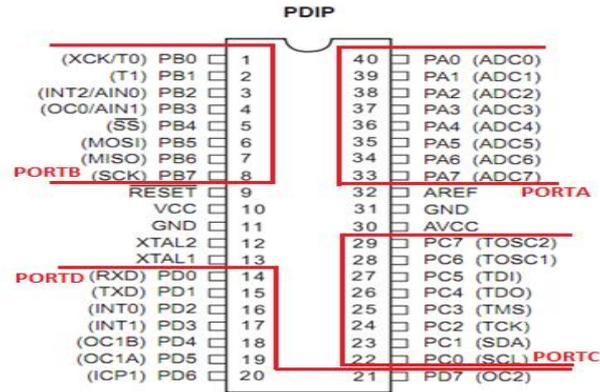
Οι μικροελεγκτές AVR της Atmel παρουσιάζουν τα ακόλουθα χαρακτηριστικά:

1. Συνδυάζουν την αρχιτεκτονική Risc με ως επί το πλείστον σταθερού μήκους εντολές, διαδικασίες αποθήκευσης – φόρτωσης στην μνήμη και 32 καταχωρητές γενικής χρήσης.
2. Μηχανισμό συνεχούς διοχέτευσης εντολών (instruction pipeline) σε δύο στάδια, που επιταχύνει σημαντικά την διαδικασία εκτέλεσης.
3. Οι περισσότερες από τις εντολές που περιλαμβάνει το ρεπερτόριό τους εκτελούνται στην διάρκεια μιας περιόδου του κεντρικού ρολογιού.
4. Λειτουργούν σε συχνότητες χρονισμού μέχρι 10 MHz.
5. Διαθέτουν μεγάλη ποικιλία σε ότι αφορά ενσωματωμένες περιφερειακές μονάδες όπως, ψηφιακές εισόδους/εξόδους (I/O) , μετατροπείς αναλογικού σήματος σε ψηφιακό ή ADC , μνήμη τύπου EEPROM, χρονιστές, μονάδες ασύγχρονης σειριακής επικοινωνίας ή UART (Universal Asynchronous Receiver Transmitter), ρολόγια πραγματικού χρόνου (RTC- Real Time Clock), μονάδες διαμόρφωσης εύρους παλμών (PWM- Pulse Width Modulation), κ.λπ.
6. Ενσωματωμένες μνήμες προγράμματος και δεδομένων.
7. Δυνατότητα προγραμματισμού εντός του συστήματος (ISP- In-System Programmable).
8. Διατίθενται σε συσκευασίες των 8 έως 64 ακροδεκτών οπότε κρίνονται κατάλληλοι για έναν μεγάλο αριθμό διαφορετικών εφαρμογών.
9. Είναι περίπου 12 φορές ταχύτεροι και πιο αποδοτικοί σε σχέση με τους ελεγκτές κλασικής αρχιτεκτονικής CISC (Complex Instruction Set Computer).
10. Ευρεία περιοχή τάσεων λειτουργίας από 2.7 έως 6.0 V.

Η σχετικά απλή αρχιτεκτονική τους δίνει το πλεονέκτημα του μικρού σε απαίτηση κύκλου εκμάθησης στους αρχαίους.(Gadre, 2001)

#### 6.4.9 Το Hardware ενός AVR μικροελεγκτή

Ο μικροελεγκτής AVR διαθέτει ακίδες, για κάθε μία από τις οποίες έχει ένα μοναδικό όνομα. Για να γίνει καλύτερα αντιληπτό αυτό ας υποθέσουμε ότι συνδέουμε ένα Led στο pin14, μπορούμε μετά να κάνουμε το pin αυτό HIGH ή LOW, δίνοντας μία τάση 5V ή 0V αντίστοιχα, μιλώντας σε αυτό ως PD 0. Οι περισσότερες ακίδες σε ένα AVR έχουν και δευτερεύουσες λειτουργίες, οι οποίες αναφέρονται μέσα σε παρένθεση. Παρακάτω στο διάγραμμα φαίνονται οι RXD και TXD που χρησιμοποιούνται στην σειριακή επικοινωνία, βρίσκονται στις ακίδες PD 0 και PD 1 αντίστοιχα. Αυτό βοηθάει στην κατανόηση της εσωτερικής λειτουργίας τους, έτσι οι ακίδες είναι χωρισμένες σε ομάδες το πολύ των οκτώ ακίδων. Κάθε ομάδα περιλαμβάνει το πολύ 8 pin, μπορούμε να αναφερόμαστε σε αυτά χρησιμοποιώντας ένα δυαδικό αριθμό των 8 bit, ώστε να ενεργοποιήσουμε ή όχι την τάση τους. .(Gadre, 2001)

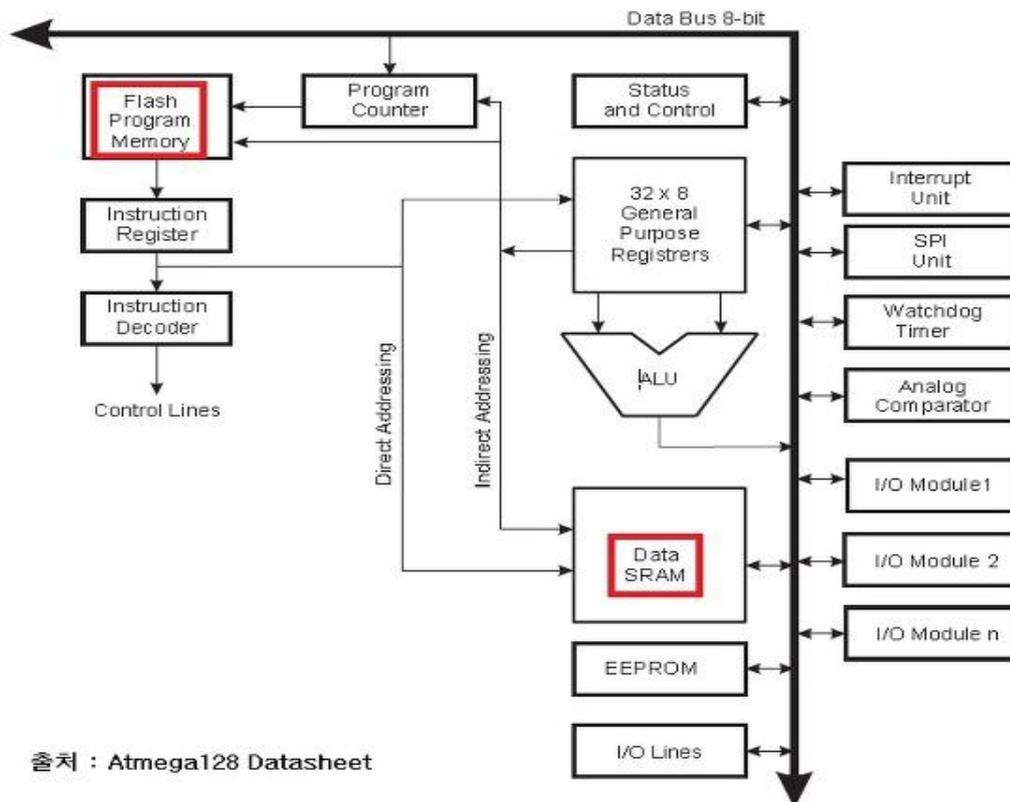


Διάγραμμα 5. Διάγραμμα των pins του μικροελεγκτή AVR.

#### 6.4.10 Ο Πυρήνας του Μικροελεγκτή AVR

Η οικογένεια των AVR μικροελεγκτών διαθέτει τα ακόλουθα συστατικά:

- CPU : Η κεντρική μονάδα επεξεργασίας (CPU) ενός AVR είναι η ‘καρδιά’ του μικροελεγκτή. Η κύρια ‘δουλειά’ της είναι να εκτελεί ανάκτηση δεδομένων από την μνήμη flash υπό μορφή εντολών, να τις αποκωδικοποιεί και μετά να τις εκτελεί. Διαθέτει 32 συνολικά καταχωρητές των 8-bit όπου αποθηκεύονται διάφορες μεταβλητές, την αριθμητική λογική μονάδα καθώς και τον αποκωδικοποιητή εντολών. (Melear, Kilbane, n.d.)



출처 : Atmega128 Datasheet

Διάγραμμα 6. Αρχιτεκτονική των μικροελεγκτών AVR.

- ΜΝΗΜΗ : Η μνήμη στους μικροελεγκτές AVR διακρίνεται σε τρεις διαφορετικούς τύπους μνήμης, κάθε ένας έχει διαφορετική χρήση:
- Μνήμη Flash : Η ενσωματωμένη μνήμη ταχείας αποθήκευσης ή Μνήμη Προγράμματος αποθηκεύει τον κώδικα του προγράμματος χωρίς την ανάγκη εξωτερικών μνήμων, η οποία σε περίπτωση διακοπής ρεύματος τα δεδομένα δεν χάνονται. Η μνήμη αυτή μπορεί να προγραμματιστεί σειριακά ή παράλληλα. Το μέγεθος της μνήμης του προγράμματος αναφέρεται συνήθως στην ονοματολογία της ίδια της συσκευής (για παράδειγμα ο Atmega328 έχει Flash 32 KB).
- Μνήμη RAM : Η Μνήμη Τυχαίας Προσπέλασης χρησιμοποιείται από την CPU για την προσωρινή αποθήκευση μεταβλητών και διευθύνσεων επιστροφής κατά την εκτέλεση του προγράμματος μας.
- Μνήμη EEPROM : Η Ηλεκτρικά Επαναπρογραμματιζόμενη Μνήμη Ανάγνωσης Μόνο αποτελεί μνήμη δεδομένων όπως σταθερές τιμές και μεταβλητές. Είναι αργή στην εγγραφή. Όπως και η Flash Memory, η EEPROM μπορεί να διατηρήσει το περιεχόμενό της ακόμη και με απουσία τάσης.(1 KB)
- CLOCK : Στα ψηφιακά κυκλώματα η σειρά με την οποία πραγματοποιούνται τα γεγονότα είναι πολύ σημαντική, καθώς κάποιο γεγονός πρέπει να προηγείται κάποιου άλλου, ενώ μερικές φορές δύο γεγονότα πρέπει να πραγματοποιούνται ταυτόχρονα. Για τη σηματοδότηση συγκεκριμένων γεγονότων, πολλά ψηφιακά κυκλώματα χρησιμοποιούν ρολόγια για την επίτευξη του χρονισμού.

Τέσσερις τύποι ρολογιών χρησιμοποιούνται στους μικροελεγκτές, κρύσταλλα (crystals), κεραμικά αντηχεία (ceramic resonators), RC (resistor , capacitor) ταλαντωτές και ταλαντωτές πυριτίου (silicon oscillators). Πολλοί από τους μικροελεγκτές της σειράς AVR διαθέτουν ενσωματωμένο ταλαντωτή τύπου RC ως βασικό ρολόι. Τρέχει στα 8 Mhz περίπου. Στη συνέχεια το ρολόι της CPU διαιρείται από το βασικό ρολόι, και τρέχει περίπου στο 1 MHz από προεπιλογή, αλλά μερικές φορές όταν χρειαζόμαστε επιπλέον ταχύτητα, το ανεβάζουμε στα 8 MHz. Μετά από το ρολόι της CPU έρχονται όλα τα άλλα περιφερειακά ρολόγια. Υπάρχουν ρολόγια για το υποσύστημα εισόδου εξόδου, για τον ADC (μετατροπέα από αναλογικό σε ψηφιακό), RAM, FLASH και EEPROM.

- OUTPUTS : Κάποιες ακίδες ενός AVR chip μπορούν χρησιμοποιηθούν ως ψηφιακές έξοδοι, δηλαδή μπορούν να δεχθούν εντολή, μέσω του προγράμματος, ώστε να εξάγει είτε το επίπεδο της τάσης τροφοδοσίας είτε το επίπεδο της τάσης γείωσης.
- INPUTS : Μπορούν να οριστούν και ως ψηφιακοί είσοδοι, όπου ελέγχουν αν η τάση που εφαρμόζεται εξωτερικά στην ακίδα είναι high η low. Εάν η τάση στην ακίδα είναι πάνω από το μισό της τάσης τροφοδοσίας, το chip θέτει ένα bit μίας εσωτερικής μεταβλητής ίσο με 1. Αν η τάση είναι κάτω από το μισό της τάσης τροφοδοσίας, το ίδιο bit θα γίνει ίσο με 0.(Gadre, 2001)

#### 6.4.11 Περιφερειακά του μικροελεγκτή AVR

Παρακάτω αναφέρονται κάποιες από τις περιφερειακές μονάδες για τους μικροελεγκτές AVR.

ΣΕΙΡΙΑΚΗ ΘΥΡΑ ΕΠΙΚΟΙΝΩΝΙΑΣ : Η θύρα αυτή χρησιμοποιείται για την επικοινωνία του ελεγκτή με εξωτερικές διατάξεις με την μορφή σειριακής μετάδοσης δεδομένων σε

οποιαδήποτε ταχύτητα μετάδοσής τους. Οι σειριακές θύρες διακρίνονται αρχικά στις ασύγχρονες σειριακές θύρες, USART (Universal Synchronous and Asynchronous Receiver and Transmitter), οι οποίες απαιτούν 1 bit εκκίνησης και 1 bit διακοπής και στις σύγχρονες σειριακές θύρες, UART, (Universal Asynchronous Receive and Transmit), οι οποίες απαιτούν ένα πρόσθετο σήμα χρονισμού.

Η επικοινωνία ενός μικροελεγκτή AVR μπορεί να γίνει με δύο μορφές είτε σε επίπεδο τερματικού είτε μέσω διαύλων. Παρακάτω αναφέρονται κάποιες μονάδες επικοινωνίας των AVR:

Η Μονάδα Σειριακής Επικοινωνίας, SPI, Serial Peripheral Interface, η οποία είναι καλή για γρήγορη επικοινωνία σε μικρές αποστάσεις, όπως μνήμες, ADCs και DACs. Το SPI είναι να πρωτόκολλο σύγχρονης σειριακής επικοινωνίας μεταξύ δύο συσκευών ή δύο ολοκληρωμένων. Το SPI υποστηρίζει έναν μεγάλο ρυθμό μετάδοσης δεδομένων περίπου μέχρι 3 MHz.

Μικροελεγκτές με ενσωματωμένο SPI μπορούν να επικοινωνήσουν με περιφερειακές συσκευές που διαθέτουν ανάλογη θύρα επικοινωνίας SPI. Σε περίπτωση όμως που ο μικροελεγκτής δεν περιέχει το συγκεκριμένο πρωτόκολλο, μπορεί να χρησιμοποιήσει τις γραμμές I/O και να μιμηθεί τους ακροδέκτες που χρησιμοποιούνται για το SPI, με μείωση όμως στην ταχύτητα μεταφοράς δεδομένων.

Μια θύρα που χρησιμοποιείται για το SPI διαθέτει τα εξής σήματα :

- MISO: Σήμα εξόδου σειριακών δεδομένων
- MOSI: Σήμα εισόδου σειριακών δεδομένων
- SCK: Σήμα χρονισμού
- SS: Σήμα επιλογής.

Έστω ότι έχουμε δυο μικροελεγκτές που επικοινωνούν μεταξύ τους με το SPI.

Τότε, όταν ο ένας έχει τον χαρακτήρα Master ο άλλος έχει τον χαρακτήρα Slave. Η μόνη διαφορά μεταξύ του Master και του Slave είναι ποιος παράγει το ρολόι χρονισμού. Ο Master παράγει το ρολόι χρονισμού και τα δεδομένα που εξέρχονται από τον ακροδέκτη MISO του Slave εισέρχονται σε αυτόν (τον Master) από τον δικό του ακροδέκτη MISO, ενώ την ίδια στιγμή δεδομένα εξέρχονται από αυτόν από τον ακροδέκτη MOSI και εισέρχονται στον Slave από τον δικό του ακροδέκτη MOSI.

Ο Δίαυλος Επικοινωνίας Ολοκληρωμένων Κυκλωμάτων, I<sup>2</sup>C, Inter Integrated Circuit, ο οποίος δημιουργεί ένα μικρό δίκτυο επιτρέποντας τη διασύνδεση αισθητήρων στο ίδιο ζεύγος καλωδίων. Συσκευές που διαχειρίζονται μικρή ποσότητα δεδομένων συνήθως χρησιμοποιούν τον I<sup>2</sup>C. Είναι ένα πρωτόκολλο επικοινωνίας μέσω δύο γραμμών οι οποίες καλούνται αντίστοιχα, γραμμή σειριακών δεδομένων (Serial Data Line - SDA) και γραμμή σειριακού ρολογιού (Serial Clock Line -SCL). Κάθε ξεχωριστή διάταξη που συνδέεται στον δίαυλο διευθυνσιοδοτείται από το πρόγραμμα οδήγησης με μια μοναδική διεύθυνση, η οποία αντιστοιχεί αποκλειστικά στην διάταξη αυτή. Τα δεδομένα μεταφέρονται στον δίαυλο με την μορφή λέξεων των 8 bits. Για να είναι εφικτή η σύνδεση ενός μικροελεγκτή AVR σε διάφορες εξωτερικές περιφερειακές διατάξεις μπορεί να γίνει χρήση ειδικών ολοκληρωμένων κυκλωμάτων επέκτασης θυρών εισόδου – εξόδου, τα οποία διαθέτουν μονάδα διασύνδεσης I<sup>2</sup>C.

Ο Σειριακός Δίαυλος Γενικής Χρήσης, USB, Universal Serial Bus, χρησιμοποιείται για την διασύνδεση του προσωπικού υπολογιστή με όλους σχεδόν τους τύπους των περιφερειακών διατάξεων όπως, πληκτρολόγιο, συσκευές ήχου, κ.λπ. Ένα σύστημα επικοινωνίας USB αποτελείται από τρία βασικά στοιχεία:

- Συνδεσμολογία USB.
- Μονάδα εξυπηρέτησης USB.

- Διατάξεις USB. Οι οποίες συνδέονται πάνω στον δίαυλο USB καθώς επίσης και κάποιες συσκευές διακλάδωσης (hubs) που συμπεριφέρονται και ως κόμβοι επέκτασης διαφόρων συστημάτων.

Επειδή κάθε μία από τις περιφερειακές μονάδες επικοινωνίας είναι ξεχωριστή μέσα στον AVR, μπορούμε να χρησιμοποιήσουμε κάθε μία από αυτές ταυτόχρονα.

**ΜΕΤΑΤΡΟΠΕΑΣ ΑΝΑΛΟΓΙΚΟΥ ΣΗΜΑΤΟΣ ΣΕ ΨΗΦΙΑΚΟ:** Είναι μία διάταξη που μετατρέπει την αναλογική τάση σε ψηφιακό αριθμό. Υπάρχουν δύο τεχνικές που μπορούν να μας αποφέρουν μεγαλύτερη ακρίβεια και μείωση του θορύβου στον ADC, αυτές είναι η oversampling και exponential smoothing.

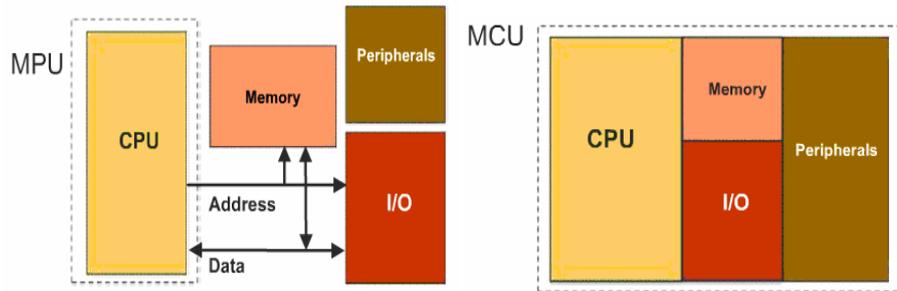
**ΔΙΑΚΟΠΤΕΣ :** Είναι ένας μηχανισμός ροής ελέγχου δεδομένων των περιφερειακών μονάδων. Είναι μία συνάρτηση που μπορούμε να γράψουμε στο πρόγραμμα μας, η οποία θα εκτελείτε αυτόματα κάθε φορά που συναντάται η συνθήκη ενός interrupt. Λέγεται ρουτίνα διακοπής (interrupt) επειδή ο επεξεργαστής σταματάει οτιδήποτε έκανε στην κύρια ροή του προγράμματος και τρέχει την κατάλληλη συνάρτηση. Μόλις τελειώσει με την ρουτίνα διακοπής, δηλαδή μετά την εκτέλεση της συνάρτησης του interrupt, ο επεξεργαστής ξεκινά και πάλι από το σημείο που είχε σταματήσει. Μπορεί να πραγματοποιηθεί και πατώντας το κουμπί reset ή αλλάζοντας μία τιμή εισόδου.

**ΧΡΟΝΙΣΤΗΣ :** Ο χρονιστής στους μικροεπεξεργαστές AVR μπορεί να χρησιμοποιηθεί και ως μετρητής. Ως χρονιστής δέχεται στην είσοδο χρονισμού του το κεντρικό σήμα χρονισμού του συστήματος. Σαν μετρητής δέχεται παλμούς ενός εξωτερικού σήματος από συγκεκριμένο ακροδέκτη εισόδου ή εξόδου. Οι μετρητές μπορούν επίσης να συνδυαστούν και με ρολόγια, γι' αυτό και συχνά αναφέρονται ως timers/counters. Με ένα ρολόι και ένα timer, μπορείς να μετρήσεις είτε πόσο χρόνο διαρκεί κάποιο γεγονός, ή τη συχνότητα του γεγονότος αυτού. (Gadre, 2001)

## 6.5 Σύγκριση μικροελεγκτή-μικροεπεξεργαστή

Σύμφωνα με τα παραπάνω διαπιστώνονται κάποιες διαφορές μεταξύ των μικροελεγκτών και των μικροεπεξεργαστών. Ο μικροελεγκτής αποτελεί ένα μικρό αυτόνομο υπολογιστικό σύστημα, προγραμματισμένο με τέτοιο τρόπο ώστε να εκτελεί μία καθορισμένη λογική ακολουθία εντολών, οι οποίες έχουν καταχωρηθεί στην μόνιμη μνήμη του. Κάθε φορά που θα κάνει επανεκκίνηση ο μικροελεγκτής, θα ακολουθεί την ίδια διαδικασία, δηλαδή θα ανακαλεί τα δεδομένα, θα τα επεξεργάζεται και με βάση τα αποτελέσματα της επεξεργασίας θα ελέγχει το περιβάλλον του. Αποτελεί, λοιπόν ένα σύστημα ειδικού σκοπού, που στόχο έχει τον έλεγχο και την εξυπηρέτηση ενός συγκεκριμένου αυτοματισμού.

Σε αντίθεση με τον μικροεπεξεργαστή, οποίος μετά την εκκίνησή του δεν είναι από μόνος του σε θέση να εκτελέσει κάποια λογική ακολουθία. Έχει την δυνατότητα σύνδεσης με τις μνήμες RAM και ROM, όμως αυτές αποτελούν ξεχωριστές μονάδες, που συνήθως δεν ολοκληρώνονται μέσα στον ίδιο τον μικροεπεξεργαστή. (Αλεβυζάκης, Μαρράκης, 2007)



Σχήμα 1. Διαφορές μικροελεγκτή-μικροεπεξεργαστή.

Η αρχιτεκτονική των μικροελεγκτών, που δεν υλοποιείται στους μικροεπεξεργαστές, είναι ότι έχουν διαφορετικό διάδρομο για τις εντολές (instructions bus) και διαφορετικό διάδρομο δεδομένων (data bus). Η αρχιτεκτονική αυτή αναφέρεται και ως αρχιτεκτονική Harvard.

Οι μικροϋπολογιστές ακολουθούν την αρχιτεκτονική Von Neumann. Η εκτέλεση μιας λογικής ακολουθίας είναι να καταχωρηθούν στη μνήμη RAM οι εντολές του προγράμματος, μέσω μίας περιφερειακής μονάδας. Το πρόγραμμα, δηλαδή, καταλαμβάνει ένα μέρος στην ίδια μνήμη που διατίθεται επίσης για τα δεδομένα και τα αποτελέσματα της επεξεργασίας. Όταν διακοπεί η τροφοδοσία, τα δεδομένα της μνήμης RAM χάνονται, μαζί με το πρόγραμμα. Την επόμενη φορά ο μικροεπεξεργαστής μπορεί να φορτώσει στη μνήμη RAM και να επεξεργαστεί ένα διαφορετικό πρόγραμμα. Οι διάφορες μονάδες του υπολογιστικού συστήματος επικοινωνούν παράλληλα μέσω των ίδιων διαδρόμων δεδομένων, διευθύνσεων και ελέγχου. (Gadre, 2001)

Οι μικροελεγκτές διαθέτουν αρκετά πλεονεκτήματα, όπως:

- Αυτονομία. Η οποία επιτυγχάνεται μέσω της ενσωμάτωσης σύνθετων περιφερειακών υποσυστημάτων όπως, μνήμες και θύρες επικοινωνίας. Καταντώντας έτσι τους μικροελεγκτές να μην χρειάζονται κανένα άλλο ολοκληρωμένο κύκλωμα για να λειτουργήσουν.
- Εύκολη υλοποίηση εφαρμογών. Η ενσωμάτωση των περιφερειακών πραγματοποιείται ευκολότερα χάρη στις απλούστερες διασυνδέσεις. Τα οφέλη αυτής της ενέργειας είναι πολλά όπως, η χαμηλότερη κατανάλωση ισχύος, μεγιστοποιώντας τη φορητότητα και ελαχιστοποιώντας έτσι το κόστος της συσκευής στην οποία ενσωματώνεται ο μικροελεγκτής.
- Είναι αρκετά οικονομικοί.
- Διαθέτουν μεγάλη αξιοπιστία, λόγω των λιγότερων διασυνδέσεων.
- Ο μικρότερος αριθμός και μήκος εξωτερικών διασυνδέσεων καθώς και οι χαμηλότερες ταχύτητες λειτουργίας οδηγούν σε μειωμένες εκπομπές ηλεκτρομαγνητικών παρεμβολών και μειωμένη ευαισθησία σε αντίστοιχες παρεμβολές από άλλες ηλεκτρικές και ηλεκτρονικές συσκευές.
- Διαθέτουν περισσότερους ακροδέκτες για ψηφιακές εισόδους-εξόδους, εξαιτίας της μη δέσμευσής τους για τη σύνδεση εξωτερικών περιφερειακών.
- Το μέγεθος του υπολογιστικού συστήματος τους είναι μικρό. (Gadre, 2001)

## 7 ARDUINO

Στο κεφάλαιο αυτό θα γίνει μια μικρή ανασκόπηση στην ανακάλυψη του Arduino, θα μελετηθεί τόσο το hardware όσο και το software υλικό του, αλλά και ιδιαίτερα οι εκδόσεις πλακετών του Arduino που χρησιμοποιούνται στην παρούσα εργασία.

### 7.1 Η Ιστορία του Arduino



Εικόνα 29. Η Ivrea.

Η ιστορία του Arduino ξεκινάει σε μια πόλη της βορειοδυτικής Ιταλίας, στην Ivrea το 2005 ξεκίνησε ένα σχέδιο προκειμένου να κατασκευαστεί μια συσκευή για τον έλεγχο προγραμμάτων και διαδραστικών σχεδίων από μαθητές, με χαμηλότερο κόστος από άλλα πρωτότυπα συστήματα που ήταν διαθέσιμα εκείνη την περίοδο. Εμπνευστής του σχεδίου αυτού ήταν ο καθηγητής Massimo Banzi, ο οποίος θέλησε να κατασκευάσει ευκολότερη τη μάθηση των ηλεκτρονικών για τους μαθητές του. Για το λόγο αυτό ζήτησε βοήθεια από τον David Cuatrecasas, μηχανικό από το πανεπιστήμιο του Malmo. Οι ιδρυτές Massimo Banzi και David Cuatrecasas ξεκίνησαν να παράγουν πλακέτες σε ένα μικρό εργοστάσιο στην Ivrea, στην ίδια περιοχή στην οποία στεγαζόταν η εταιρία υπολογιστών Olivetti. Ονόμασαν το έργο τους Adruin of Ivrea «Arduino» που μεταφράζεται ελεύθερα ως «γενναίος φίλος», με σκοπό να δημιουργήσουν έναν μικροελεγκτή προσιτό ως προς τη χρήση του. Την ανάπτυξη του λογισμικού για τον μικροελεγκτή την ανέθεσαν σε δύο φοιτητές του πανεπιστημίου Malmo. Η πρώτη παρτίδα που παράχθηκε αποτελούνταν από 200 μικροελεγκτές, υπό την εποπτεία του ηλεκτρολόγου μηχανικού Gianluca Martino. Οι μικροελεγκτές αυτοί ονομάστηκαν Serial Arduino και περιελάμβαναν μία ATmega8 με άμεση σύνδεση RS-232 με το μικροελεγκτή και όλα τα επιμέρους περιφερειακά του. Έκτοτε έχουν δημιουργηθεί δεκάδες πλακέτες Arduino παγκοσμίως, χάρη στο ευέλικτο και εύκολο στη χρήση hardware και software. Το Arduino μπορεί να χρησιμοποιηθεί από τον οποιοδήποτε με στοιχειώδεις γνώσεις ηλεκτρονικών που ενδιαφέρεται να δημιουργήσει διαδραστικά αντικείμενα ή περιβάλλοντα, με μόνο περιορισμό τη φαντασία του.

### 7.2 Τι είναι το arduino

Σύμφωνα με τον, Massimo Banzi, το Arduino είναι μία ανοικτού κώδικα πλατφόρμα «πρωτοτυποποίησης» ηλεκτρικών κυκλωμάτων βασισμένη σε ευέλικτο και εύκολο στη χρήση υλικό και λογισμικό, που προορίζεται για οποιονδήποτε διαθέτει μερική εμπειρία στον προγραμματισμό, στοιχειώδεις γνώσεις ηλεκτρονικών και ενδιαφέρεται να δημιουργήσει διαδραστικά αντικείμενα ή περιβάλλοντα.

Δύο επιμέρους μέρη απαρτίζουν το Arduino, η πλακέτα Arduino η οποία αποτελεί το κομμάτι του hardware και το δεύτερο τμήμα που είναι το Arduino IDE, το κομμάτι του λογισμικού που τρέχει στον υπολογιστή.



Εικόνα 30. Διάφορες πλακέτες του Arduino.

### 7.3 Η πλατφόρμα του

Μία πλακέτα Arduino αποτελείται από ένα μικροελεγκτή AVR όπως, ATmega2560, ATmega328 και ATmega168 στις νεότερες εκδόσεις και ATmega8 στις παλαιότερες και συμπληρωματικά εξαρτήματα για την διευκόλυνση του χρήστη στον προγραμματισμό και την ενσωμάτωσή του σε άλλα κυκλώματα. Οι περισσότερες πλακέτες περιλαμβάνουν ένα γραμμικό ρυθμιστή τάσης 5V και έναν κρυσταλλικό ταλαντωτή 16MHz. Ο μικροελεγκτής είναι προγραμματισμένος με ένα bootloader, έτσι ώστε να μην χρειάζεται εξωτερικός προγραμματιστής. Ο φορτωτής εκκίνησης διευκολύνει την διαδικασία αποθήκευσης των προγραμμάτων στην μνήμη flash μέσω USB.

Η πλακέτα του Arduino δίνει την δυνατότητα για πολλές εισόδους ή εξόδους για χρήση με άλλα κυκλώματα μέσω των pin που υπάρχουν τοποθετημένα πάνω του σχεδόν σε όλες τις πλευρές ανάλογα την έκδοσή του. Η πλακέτα μπορεί να τροφοδοτηθεί είτε με τροφοδοτικό AC/DC Adapter των 7-12V είτε απευθείας από τη USB θύρα του ηλεκτρονικού υπολογιστή, είτε μέσω μιας μπαταρίας, όπου τα καλώδιά της θα τοποθετηθούν στα pin GND και Vin. Η επιλογή τροφοδοσίας USB ή εξωτερικής πηγής γίνεται αυτόματα. Τέλος, περιλαμβάνει ένα κουμπί επανεκκίνησης Reset σε περίπτωση που θέλουμε να το επαναφέρουμε στις εργοστασιακές ρυθμίσεις.



Εικόνα 31. Ο Arduino Mega2560.

#### 7.4 Το λογισμικό του

Το ολοκληρωμένο περιβάλλον ανάπτυξης (IDE) του Arduino έχει γραφτεί σε γλώσσα προγραμματισμού Java, επιτρέποντάς του να μπορεί να μεταφερθεί στα περισσότερα λειτουργικά συστήματα. Το περιβάλλον ανάπτυξης του είναι βασισμένο στην Processing, το οποίο είναι ένα περιβάλλον ανάπτυξης σχεδιασμένο να εισάγει στον προγραμματισμό μη εξοικειωμένους με την ανάπτυξη λογισμικού. Η συγκεκριμένη γλώσσα προγραμματισμού προέρχεται από την Wiring, μία γλώσσα που μοιάζει με την C η οποία παρέχει παρόμοια λειτουργικότητα για μια πιο περιορισμένης σχεδίασης πλακέτα, της οποίας το περιβάλλον ανάπτυξης βασίζεται επίσης στην Processing.

Ένα ολοκληρωμένο περιβάλλον ανάπτυξης (Integrated Development Environment – IDE) περιλαμβάνει κάποια παραδείγματα και βιβλιοθήκες με σκοπό να βοηθήσει και τους αρχάριους στον προγραμματισμό στην ανάπτυξη προγραμμάτων. Το περιβάλλον ανάπτυξης του Arduino περιέχει έναν επεξεργαστή κειμένου για τη σύνταξη κώδικα, μια περιοχή μηνυμάτων, μια γραμμή εργαλείων με κουμπιά για κοινές λειτουργίες, καθώς και μια σειρά από μενού. Μέσω ενός κουμπιού του επιτρέπεται η σύνδεση με την πλακέτα του Arduino για να φορτωθεί το πρόγραμμα και να υπάρξει αλληλοεπικοινωνία μεταξύ χρήστη και πλακέτας.

Με το άνοιγμα του Arduino IDE αυτόματα δημιουργείται η δομή ενός σκίτσου (sketch), το οποίο είναι ένα μικρό πρόγραμμα στον υπολογιστή, που φορτώνεται στον μικροελεγκτή της πλακέτας Arduino και καλείται να το εκτελέσει. Το σκίτσο γράφεται στον χώρο επεξεργασίας κειμένου του IDE, δίνει την δυνατότητα αποκοπής ή επικόλλησης και αναζήτησης ή αντικατάστασης κειμένου. Έχει μία περιοχή «μηνυμάτων» για ανατροφοδότηση με παράλληλη εμφάνιση σφαλμάτων. Η κονσόλα απεικονίζει την έξοδο του κειμένου από το περιβάλλον Arduino συμπεριλαμβανομένων πλήρων μηνυμάτων λάθους και άλλες πληροφορίες. Διαθέτει μία σειρά από επιλογές εργαλείων που επιτρέπουν τον έλεγχο, την μεταγλώττιση και τη φόρτωση του προγράμματος, αλλά και την αποθήκευση ενός σκίτσου, ενώ ακόμη δίνει την δυνατότητα μέσω ενός κουμπιού να ανοιχτεί η σειριακή οθόνη.

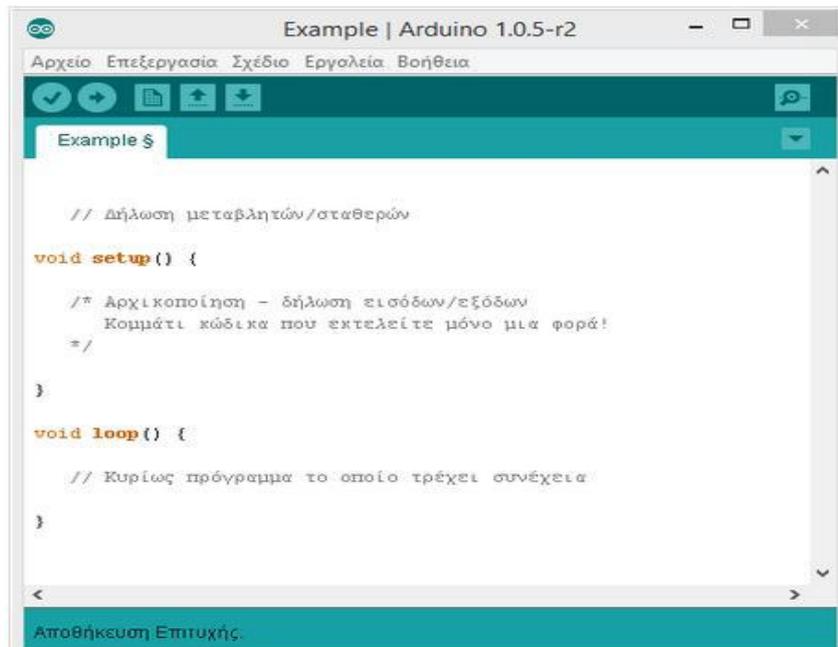
Στον παρακάτω πίνακα παρουσιάζονται τα εργαλεία του περιβάλλοντος ανάπτυξης.

Εργαλείο	Περιγραφή
 Verify	Ελέγχει για συντακτικά λάθη στον κώδικα.
 Upload	Μεταγλωττίζει τον κώδικα και τον φορτώνει στο Arduino. Αν δεν είναι συντακτικά σωστός δεν μπορεί να γίνει η φόρτωση.
 New	Δημιουργεί ένα νέο sketch.
 Open	Παραθέτει ένα menu με όλα τα sketch. Ενεργοποιώντας ένα από αυτά, θα ανοίξει αυτόματα στο τρέχον παράθυρο.
 Save	Αποθηκεύει ένα sketch.
 Serial Monitor	Ανοίγει την σειριακή οθόνη και μέσω αυτής παρακολουθείται η ανταλλαγή δεδομένων που γίνεται στην σειριακή θύρα.

Πίνακας 1. Τα εργαλεία του IDE.

Η σύνταξη ενός προγράμματος χωρίζεται σε τρία μέρη:

- 1<sup>ο</sup> : Οι δηλώσεις μεταβλητών και ενσωματώσεις βιβλιοθηκών.
- 2<sup>ο</sup> : Μέσα στην **setup()** γράφονται οι συναρτήσεις και κομμάτι του κώδικα που εκτελείται μία μόνο φορά στην αρχή του προγράμματος, η οποία αρχικοποιεί τις ρυθμίσεις. Αρχικοποίηση καταστάσεων και μεταβλητών.
- 3<sup>ο</sup> : Μέσα στην **loop()** γράφεται το κυρίως πρόγραμμα και η εκτέλεση επαναλαμβάνεται συνέχεια μέχρι να απενεργοποιηθεί (και να βγει ο Arduino από το ρεύμα).



```
Example | Arduino 1.0.5-r2
Αρχείο Επεξεργασία Σχέδιο Εργολεία Βοήθεια
Example $
// Δήλωση μεταβλητών/σταθερών
void setup() {
  /* Αρχικοποίηση - δήλωση εισόδων/εξόδων
  Κομμάτι κώδικα που εκτελείτε μόνο μια φορά!
  */
}
void loop() {
  // Κυρίως πρόγραμμα το οποίο τρέχει συνέχεια
}
```

Εικόνα 32. Η δομή ενός προγράμματος.

## 7.5 Πλεονεκτήματα του arduino

Μελετώντας το Arduino διαπιστώθηκε ότι είναι ένα εργαλείο πολύ εύχρηστο, τόσο σε αρχάριους προγραμματιστές, πόσο μάλλον στους προχωρημένους, καθώς έχει έναν μεγάλο αριθμό πλεονεκτημάτων που δικαιολογούν την χρήση της συγκεκριμένης πλατφόρμας, όπως:

- Είναι οικονομικό. Οι πλακέτες του Arduino είναι σχετικά φθηνές σε σχέση με άλλους μικροελεγκτές. Το κόστος αγοράς δεν ξεπερνάει τα 50 €.
- Είναι μεταφέσιμο και σε πολλά λειτουργικά συστήματα. Οι μηχανικοί λογισμικού, ανέπτυξαν το περιβάλλον προγραμματισμού του Arduino για Windows, Macintosh OSX και για λειτουργικά συστήματα Linux. Τα περισσότερα συστήματα ανάπτυξης μικροελεγκτών περιορίζονται στα Windows.
- Το προγραμματιστικό περιβάλλον του Arduino γίνεται εύκολα κατανοητικό και μπορούν να το χρησιμοποιήσουν πολύ εύκολα όλοι.
- Είναι ανοιχτού λογισμικού. Το λογισμικό του Arduino διανέμεται με την μορφή εργαλείων ανοιχτού λογισμικού και μπορεί να επεκταθεί από έμπειρους προγραμματιστές. Αποτελεί το μεγαλύτερο πλεονέκτημα, αφού δίνει την δυνατότητα να ασχοληθούν όσοι ενδιαφέρονται με την περαιτέρω επέκταση του κώδικα, να προσθέσουν βιβλιοθήκες και όλα αυτά μπορούν να τα μοιράζονται μεταξύ τους.

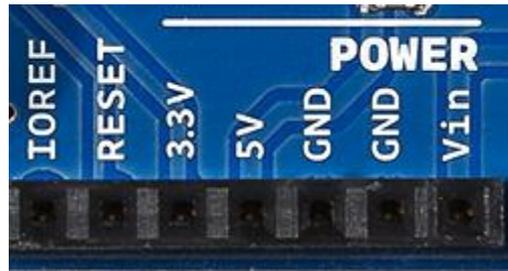
- Είναι ανοιχτού υλικού. Το Arduino βασίζεται στους μικροελεγκτές της Atmel ATMEGA8, ATMEGA168, ATMEGA328. Τα σχηματικά για τα αναπτυξιακά είναι κάτω από την άδεια της Creative Commons, επιτρέποντας σε έμπειρους σχεδιαστές να κατασκευάσουν το δικό τους αναπτυξιακό, εξελίσσοντάς το ήδη υπάρχον χωρίς να έχουν νομικά προβλήματα. Η ακόμη καλύτερα όχι τόσο έμπειροι χρήστες μπορούν να επιδιώξουν την αντιγραφή και κατασκευή της πλακέτας σε ράστερ για να καταλάβουν την λειτουργία ενός Arduino.
- Διαθέτει πολλούς τρόπους διασύνδεσης. Μερικοί από αυτούς είναι η SPI, η I<sup>2</sup>C, η σειριακή διεπαφή, οι PWM (Pulse Width Modulation), οι ADC, οι ψηφιακές εισοδοι/έξοδοι και δυνατότητα αξιοποίησης των χρονιστών και των interrupts του AVR.
- Μεγάλη ποικιλία στις πλατφόρμες. Η πληθώρα του υλικού που μπορεί να χρησιμοποιήσει κάποιος για να εξοικειωθεί και να αποκτήσει εμπειρία σε πρώτο στάδιο και μετέπειτα μπορεί να διαλέξει την ανάλογη έκδοση που του επιτρέπει να υλοποιήσει την κατασκευή του.
- Είναι επεκτάσιμο. Οι διάφορες εκδόσεις των Shields δίνουν την δυνατότητα να κουμπώσουν πάνω στο Arduino και να επεκτείνουν την λειτουργικότητά του, είτε με την πρόσβαση στο διαδίκτυο, είτε μετατρέποντάς το σε GPS, είτε να παράγει διάφορους ήχους, ή να αποτελεί ένα από τα βασικά εξαρτήματα σε μία ρομποτική κατασκευή.

## 7.6 Arduino Mega on Mega2560



Εικόνα 33. Ο Arduino Mega2560.

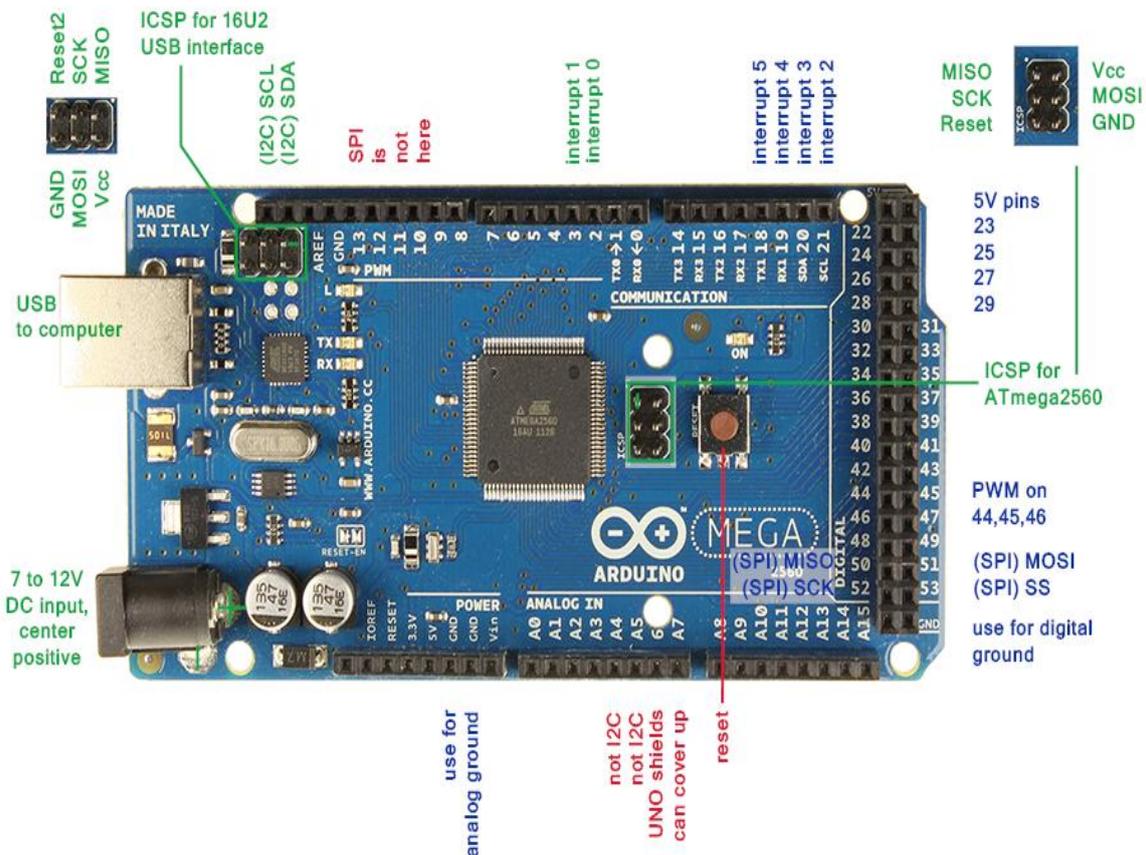
Το Arduino Mega 2560 είναι μια αναπτυξιακή πλατφόρμα βασισμένη στον μικροεπεξεργαστή ATmega2560. Έχει 54 ψηφιακές εισόδους και εξόδους (εκ των οποίων οι 15 μπορούν να χρησιμοποιηθούν ως αποτέλεσμα PWM), 16 αναλογικές εισόδους, έναν ταλαντωτή κρυστάλλου των 16 MHz, μια σειριακή σύνδεση μέσω USB, μια είσοδο παροχής ενέργειας και ένα κουμπί επανατοποθέτησης (reset button). Η τροφοδοσία μπορεί να είτε με την επικοινωνία με έναν υπολογιστή διαμέσου ενός καλωδίου USB ή με έναν AC-to-DC προσαρμογέα ή με μία μπαταρία. Το Arduino Mega μπορεί να τροφοδοτηθεί μέσω USB ή με εξωτερικό τροφοδοτικό.



Εικόνα 34. Οι ακροδέκτες τροφοδοσίας.

Οι ακροδέκτες τροφοδοσίας είναι οι εξής:

- VIN: Η τάση εισόδου της πλακέτας, όταν χρησιμοποιείται εξωτερική πηγή ενέργειας. Εάν χρησιμοποιηθεί μπαταρία, τότε η τροφοδοσία τάσης γίνεται μέσω αυτού του ακροδέκτη.
- 5V: Η τάση που χρησιμοποιείται από τον μικροελεγκτή και από άλλα εξωτερικά στοιχεία συνδεδεμένα με την πλακέτα. Η τάση αυτή δίνεται μέσω της σύνδεσης με το USB.
- 3.3V: Η τάση αυτή παράγεται από τον ελεγκτή Serial-over-USB και η μέγιστη ένταση ρεύματος είναι τα 50mA.
- GND: Υπάρχουν δύο ακροδέκτες για την γείωση.



Εικόνα 35. Τα μέρη της πλακέτας.

**Υποδοχές εισόδου/εξόδου:** Κάθε μία από τις 54 ψηφιακές ακίδες του Mega μπορούν να χρησιμοποιηθούν ως είσοδο ή έξοδο, χρησιμοποιώντας τις συναρτήσεις `pinMode`, `digitalWrite`, `digitalRead`. Λειτουργούν σε 5V. Κάθε είσοδος μπορεί να παρέχει ή να λάβει ένα μέγιστο ρεύμα έντασης 20 mA ενώ προστατεύεται από υπερτροφοδοσία μέσω pull-up αντιστάσεων.

Οι ακίδες με τις επιπλέον λειτουργίες είναι οι ακόλουθες:

**Σειριακοί δίαυλοι:**

- Σειριακή επικοινωνία 0 (RX) και 1 (TX)
- Σειριακή επικοινωνία 1: 19 (RX) και 18 (TX)
- Σειριακή επικοινωνία 2: 17 (RX) και 16 (TX)
- Σειριακή επικοινωνία 3: 15 (RX) και 14 (TX)

**Εξωτερικοί διακόπτες ροής προγράμματος:**

Οι ακίδες με τους αριθμούς 2 (`interrupt 0`), 3 (`interrupt 1`), 18 (`interrupt 5`), 19 (`interrupt 4`), 20 (`interrupt 3`), 21 (`interrupt 2`).

Χρησιμοποιείται η συνάρτηση: `attachInterrupt(interrupt, function, mode)`

Το mode είναι μία από τις καταστάσεις:

- LOW: έχει χαμηλό ρεύμα
- CHANGE: κάθε φορά που αλλάζει η κατάσταση ρεύματος στην ακίδα
- RISING: το ρεύμα από χαμηλό γίνεται υψηλό
- FALLING: το ρεύμα από υψηλό γίνεται χαμηλό

Ρυθμίζονται από το πρόγραμμα, ώστε να λειτουργούν αποκλειστικά ως ψηφιακές είσοδοι στις οποίες όταν συμβαίνουν συγκεκριμένες αλλαγές, η ροή του προγράμματος σταματάει άμεσα και εκτελείται η παραπάνω συνάρτηση. Είναι χρήσιμα σε εφαρμογές που απαιτούν συγχρονισμό μεγάλης ακρίβειας.

**Ακίδες PWM:**

Οι ακίδες με τους αριθμούς 2 έως 13 και 44 έως 46. Λειτουργούν και ως ψευδο-αναλογικές εξοδοι.

Χρησιμοποιείται η συνάρτηση: `analogRead(analogPin)`,

όπου `analogPin` ο αριθμός της ακίδας.

**Αναλογικοί είσοδοι:**

Οι ακίδες με τους αριθμούς A0 έως και A15, που βρίσκονται πάνω στην πλακέτα μετά τις ακίδες τροφοδοσίας, χρησιμοποιούνται για την ανάγνωση αναλογικών σημάτων.

### **Επικοινωνία SPI:**

Αντιστοιχεί στις ακίδες 50 (MISO), 51 (MOSI), 52 (SCK), 53 (SS).

Υποστηρίζουν την επικοινωνία SPI, η οποία αν και παρέχεται δεν χρησιμοποιείται.

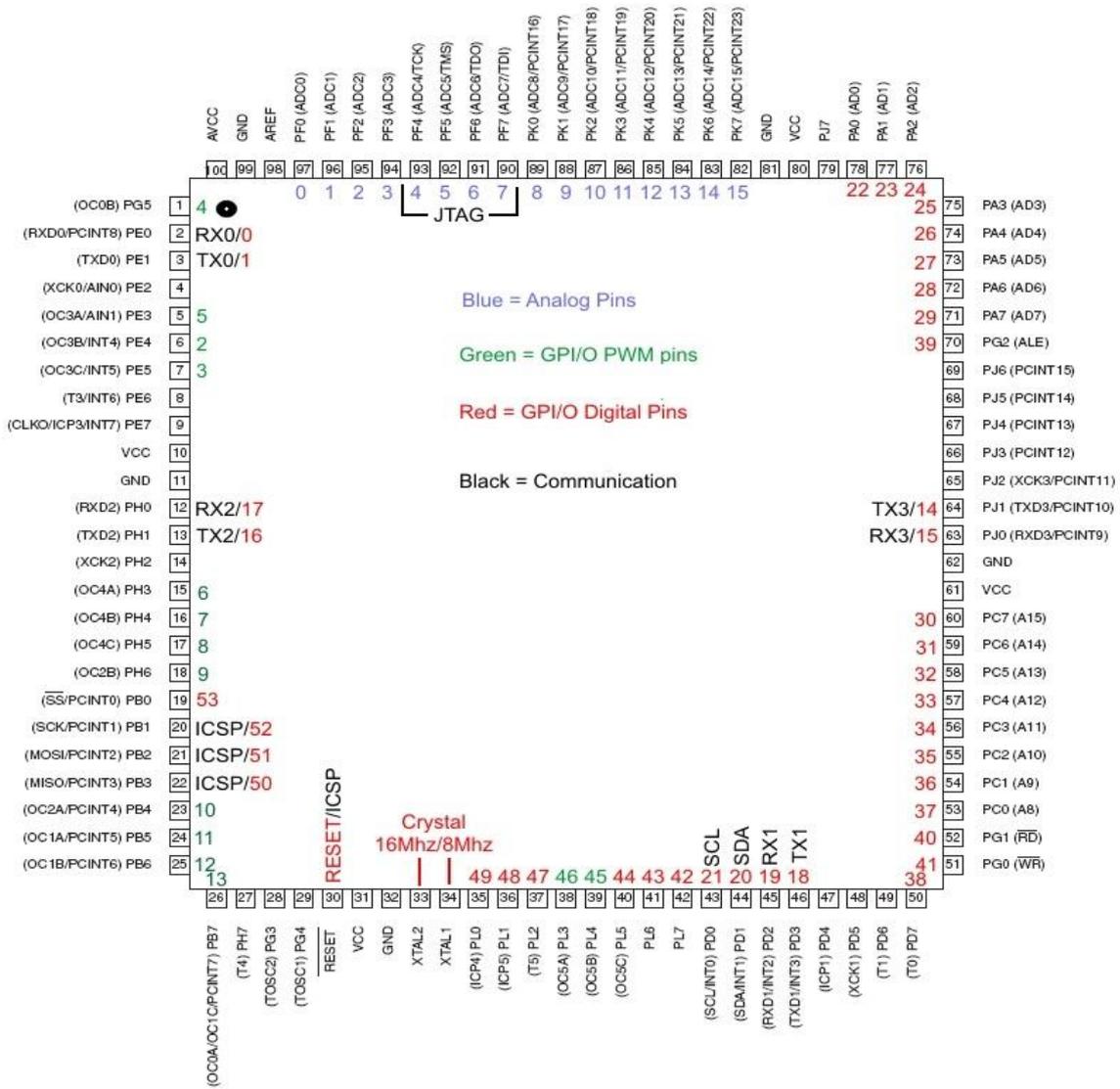
### **LED:**

Στην ακίδα 13 υπάρχει ενσωματωμένο led που συνδέεται με την ψηφιακή είσοδο 13. Όταν ανάβει σημαίνει ότι η πλακέτα έχει επαρκές ρεύμα για να λειτουργήσει η πλακέτα.

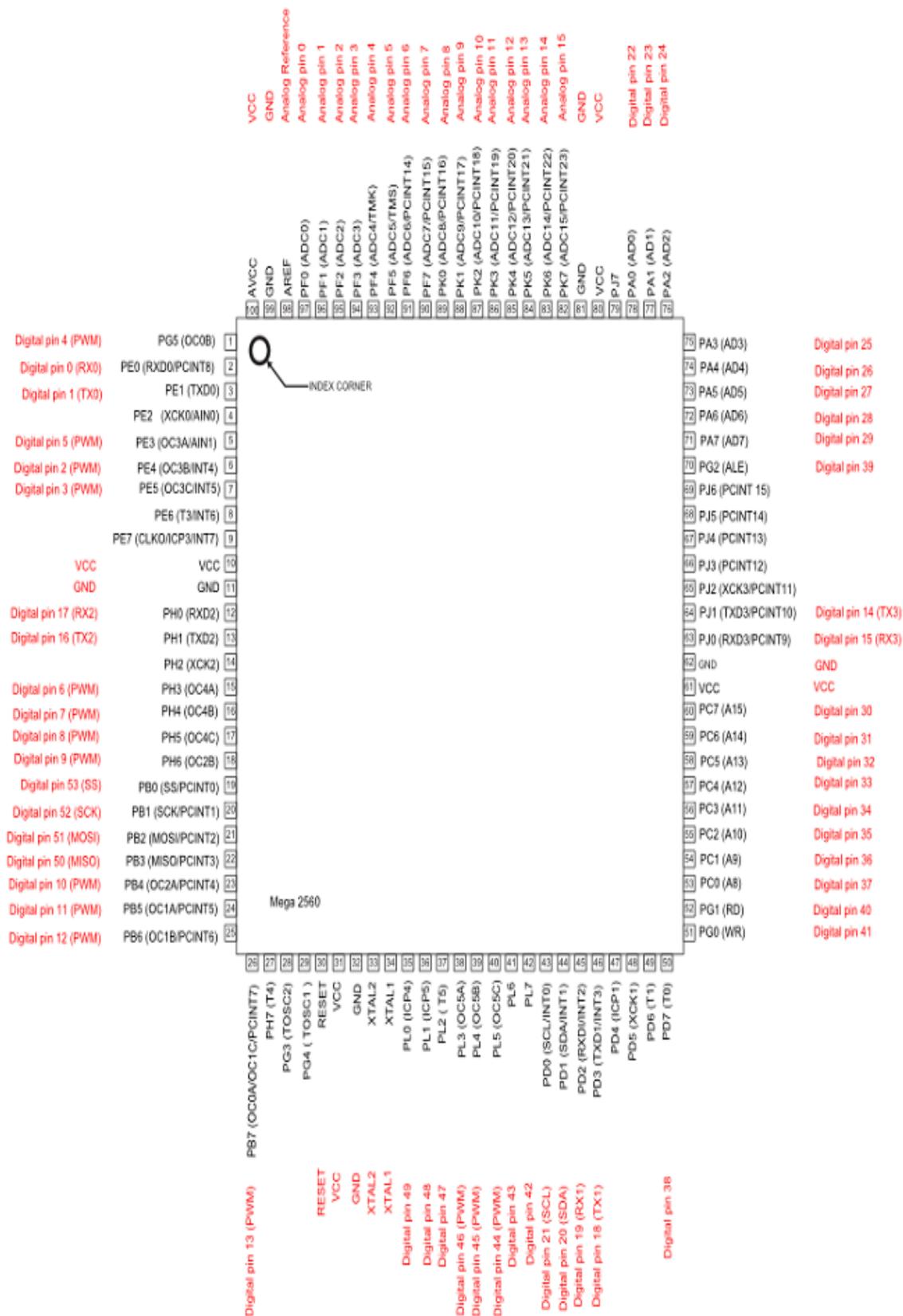
### **Επικοινωνία I<sup>2</sup>C:**

Οι ακίδες 20 (SDA) και 21 (SCL). Υποστηρίζει σειριακή επικοινωνία μεταξύ ολοκληρωμένων κυκλωμάτων. Υπάρχει η δυνατότητα να συνδεθούν μέσω αυτού του πρωτοκόλλου επικοινωνίας πολλοί αισθητήρες χωρίς να προκληθεί πρόβλημα, καθώς η μετάδοση γίνεται για τον κάθε αισθητήρα ξεχωριστά, χωρίς να παρεμβάλλει ο ένας τα δεδομένα του στον άλλο. Ο Arduino καταφέρνει να υλοποιήσει αυτόν τον συγχρονισμό.

Παρακάτω παραθέτετε ο μικροελεγκτής ATmega2560 καθώς και το σχηματικό του Arduino Mega2560.



Εικόνα 36. Ο ATmega2560.

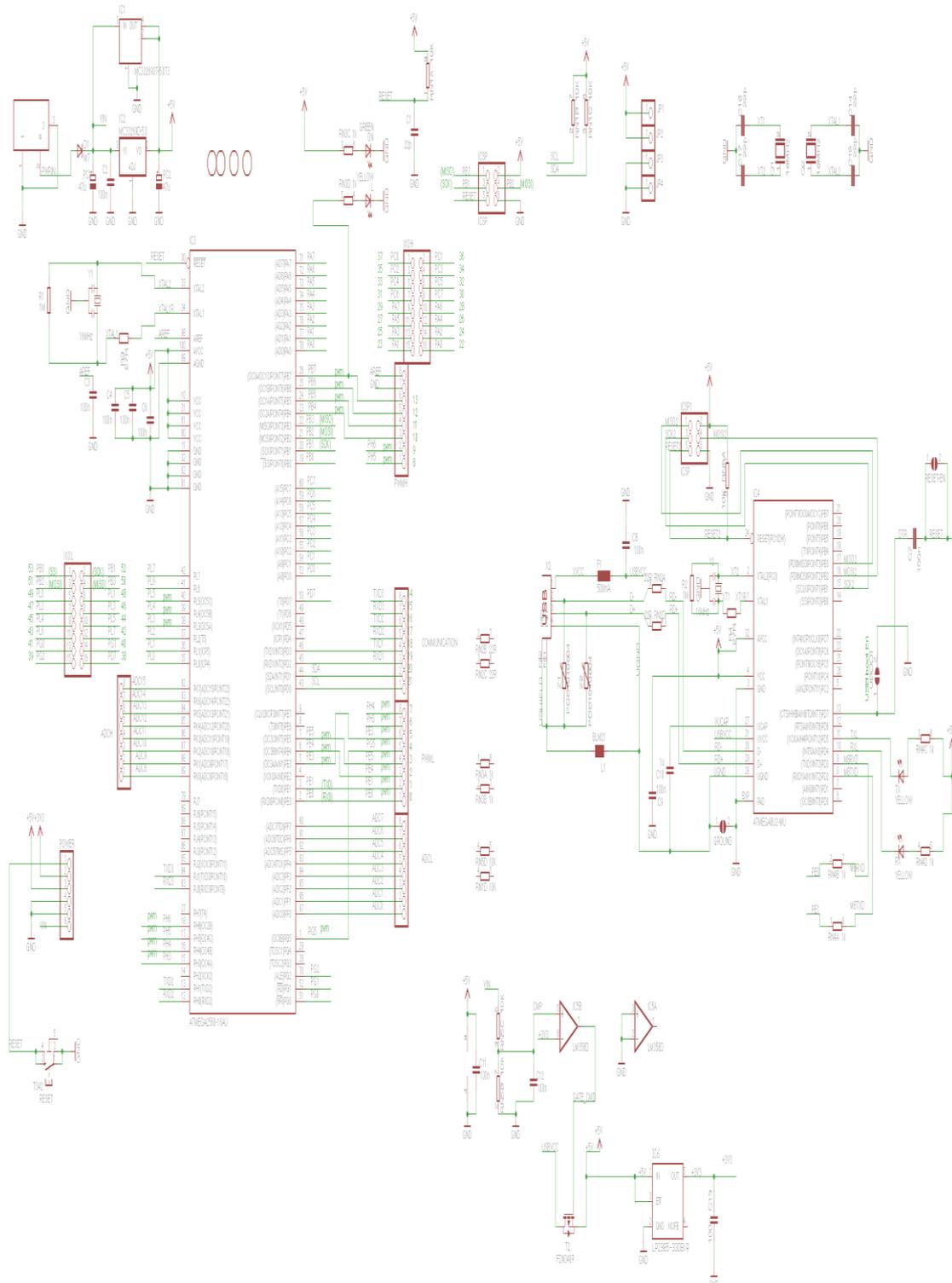


Εικόνα 37. Ο ATmega2560.

## Arduino Mega 2560 Reference Design

Reference Design for ARDUINO MEGA 2560 WITH ALL FAILSAFE AND OTHER WARRANTIES, EXPRESS OR IMPLIED, REGARDING PRODUCTS, INCLUDING BUT NOT LIMITED TO, ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

Atmel may make changes to specifications and product descriptions at any time without notice. The Customer must not rely on the absence or stated details of any features or instructions marked "reserved" or "undefined". Atmel reserves the right to discontinue or alter availability of products or specifications without notice, and is not responsible for any consequences thereof. The product information on the Web Site or Release is subject to change without notice. Do not fabricate a design with this information.



Σχήμα 2. Το σχηματικό του Arduino Mega2560.

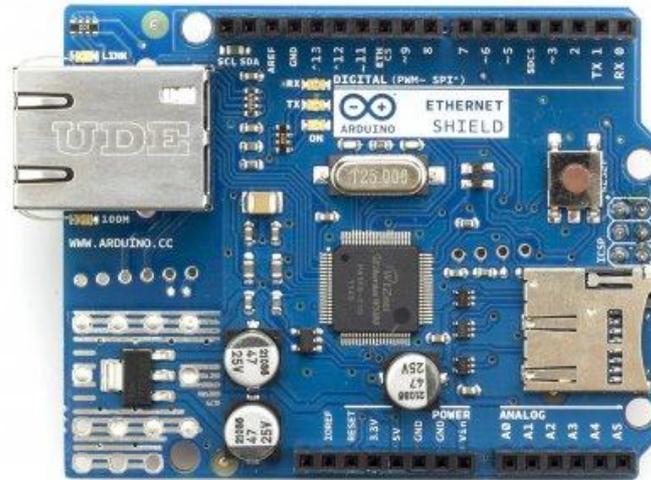
Όταν κάποιος ξεκινήσει να ασχοληθεί με το Arduino, συνήθως η πρώτη του επιλογή είναι μία έκδοση όπως το UNO. Εάν στην πορεία διαπιστώσει ότι οι ανάγκες του δεν καλύπτονται από αυτήν την πλακέτα, τότε στρέφεται στον Arduino Mega 2560, αφού μπορεί να καλυφθεί σε ανάγκες μνήμης και δέσμευση εισόδων/εξόδων. Το οποίο έχει την ίδια επεξεργαστική ισχύ με το UNO, αλλά διαθέτει περισσότερη μνήμη σε περίπτωση που χρειάζεται να συνδεθούν πολλοί αισθητήρες ή να φορτωθούν πολλές βιβλιοθήκες. Ένα μειονέκτημα του Mega2560 είναι εάν και ο μικροελεγκτής, δεν μπορούμε να τον βγάλουμε και να τοποθετήσουμε καινούργιο.

Στον πίνακα που ακολουθεί συγκεντρώθηκαν όλα τα χαρακτηριστικά των πλακετών που θα χρησιμοποιηθούν στην υλοποίηση της εργασίας.

Μοντέλα	Arduino Mega 2560	Arduino Ethernet Shield
Μικροελεγκτής	ATmega2560	ATmega328
Τάση Λειτουργίας	5V	5V
Τάση Εισόδου	7-12V	7-12V
Όρια Τάσης	6-20V	6-20V
Ψηφιακοί ακροδέκτες εισόδου / εξόδου	54 (από τα οποία 15 παρέχουν έξοδο PWM)	14 (από τα οποία 4 παρέχουν Έξοδο PWM)
Κανάλια PWM	15	4
Αναλογικοί ακροδέκτες εισόδου	16	6
DC ρεύμα ανά είσοδο/έξοδο ακροδέκτη	20 mA	40 mA
DC ρεύμα για 3.3V ακροδέκτη	50 mA	50 mA
Μνήμη Flash	256 KB (8 KB χρησιμοποιείται από τον bootloader)	32 KB (0.5 KB χρησιμοποιούνται από τον bootloader)
SRAM	8 KB	2 KB
EEPROM	4 KB	1 KB
Ταχύτητα Ρολογιού	16 MHz	16 MHz
Διαστάσεις	101.52 x 53.3 mm	68.6 x 53.3 mm
Βάρος	32 g	28 g
Τιμή (23% ΦΠΑ)	45 €	30 €

Πίνακας 2. Η σύγκριση των Arduino Mega2560-Ethernet Shield.

## 7.7 Arduino Ethernet Shield



Εικόνα 38. Ο Arduino Ethernet Shield.

Το Arduino Ethernet Shield είναι μία ολοκληρωμένη πλακέτα σχεδιασμένη ώστε να κουμπώνει πάνω στο Arduino επιτρέποντάς του να συνδεθεί στο Internet μέσω ενός τυπικού καλωδίου Ethernet. Η συγκεκριμένη έκδοση βασίζεται στο Ethernet chip Winzet W5100. Το Winzet W5100 παρέχει την δυνατότητα να συνδεθεί σε Ethernet δίκτυο (IP), για TCP και UDP επικοινωνία. Όπως και οι άλλες εκδόσεις του Arduino διατίθεται μία έτοιμη βιβλιοθήκη Ethernet που επιτρέπει τη προγραμματισμό προγραμμάτων ώστε να συνδεθεί στο Internet χρησιμοποιώντας το Ethernet Shield. Το Ethernet Shield συνδέεται-κουμπώνει στο Arduino χρησιμοποιώντας τις συρμάτινες προεξοχές που εκτείνονται από την Shield, έτσι κρατά τη διάταξη του άθικτη. Η πλακέτα παρέχει ένα πρότυπο RJ45 υποδοχής για Ethernet καλώδιο. Ένα κουμπί επαναφοράς που επαναφέρει τόσο τον W5100 αλλά και την πλακέτα Arduino σε αρχικές ρυθμίσεις.



Εικόνα 39. Winzet W5100.

Υπάρχει μια ενσωματωμένη υποδοχή κάρτας Micro-SD, το οποίο μπορεί να χρησιμοποιηθεί για να αποθηκεύσει τα αρχεία για την εξυπηρέτηση μέσω του δικτύου. Είναι συμβατό με το Arduino Uno και Mega (χρησιμοποιώντας τη βιβλιοθήκη Ethernet). Η ενσωματωμένη συσκευή ανάγνωσης καρτών microSD είναι προσβάσιμη μέσω της βιβλιοθήκης SD.

#### **Ακίδες τροφοδοσίας:**

Όπως και στις πλακέτες των Arduino έχουν τις ίδιες επιλογές, VIN, 5V, 3.3V, GND.

#### **Αναλογικοί είσοδοι:**

Έπονται από τις ακίδες της τροφοδοσίας και κυμαίνονται μεταξύ A0 και A5.

Η ακίδα A0 προστατεύει την εγγραφή της κάρτας. Η ακίδα A1 ελέγχει εάν έχει τοποθετηθεί η κάρτα θέση υποδοχής της.

#### **Ψηφιακοί είσοδοι/έξοδοι:**

Οι ακίδες PWM είναι από 0 έως 13.

Η ακίδα D2 σαν διακόπτης του Ethernet. Η ακίδα D4 χρησιμοποιείται για την SD κάρτα και μπορεί να χρησιμοποιηθεί σαν γενική είσοδος/έξοδος ένα δεν έχει τοποθετηθεί κάρτα. Η ακίδα D10 χρησιμοποιείται για την Ethernet SPI CS και δεν μπορεί να δεσμευτεί σαν είσοδος/έξοδος. Η ακίδα D11 αντιστοιχεί σε SPI MOSI. Η ακίδα D12 σε MISO SPI. Και η ακίδα D13 σε SPI SCK.

#### **Επικοινωνία I<sup>2</sup>C:**

Μετά τις ψηφιακές ακίδες υπάρχουν δύο ακίδες οι SDA και SCL που υποστηρίζουν την σειριακή επικοινωνία.

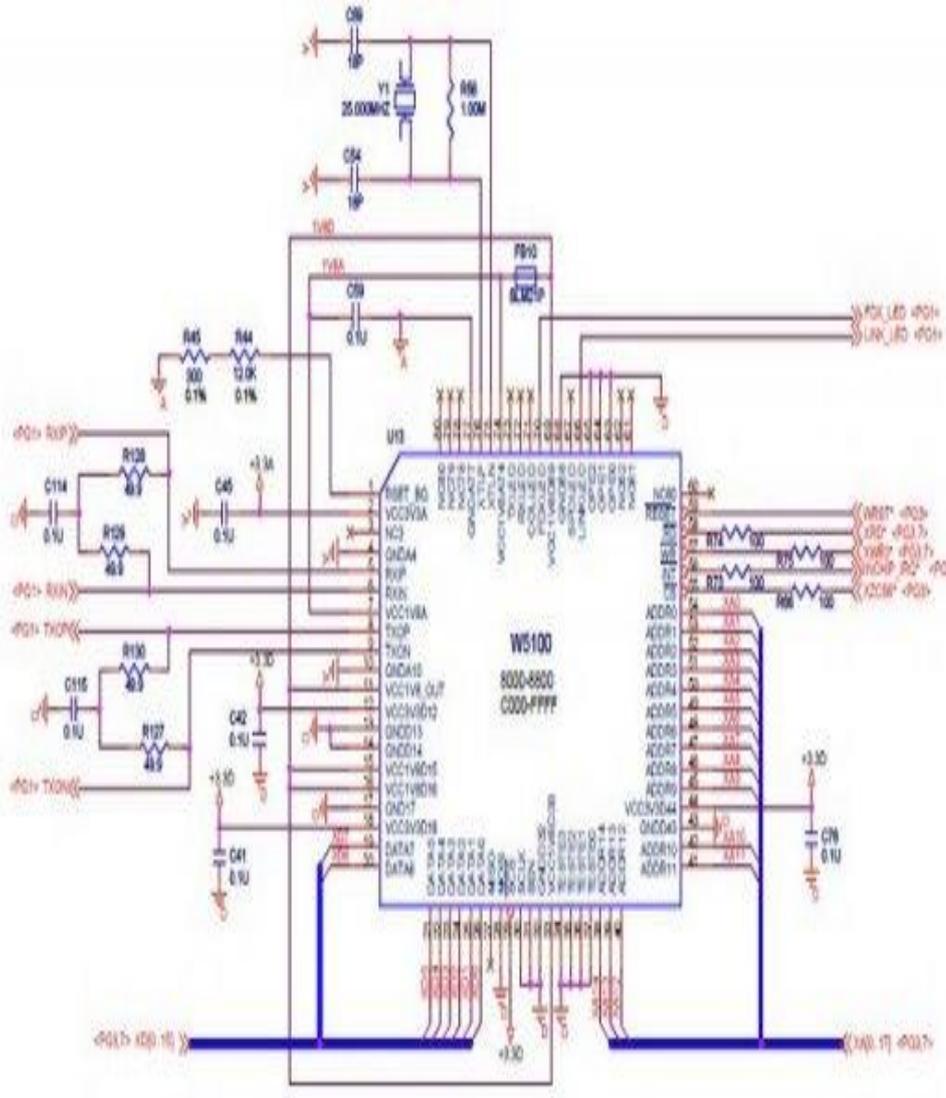
#### **Επικοινωνία SPI:**

Ο Arduino επικοινωνεί τόσο με τον W5100 όσο και με την SD κάρτα χρησιμοποιώντας τη λεωφόρο που προσφέρει η επικοινωνία SPI. Η ακίδα D10 χρησιμοποιείται για την Ethernet SPI CS και δεν μπορεί να δεσμευτεί σαν είσοδος/έξοδος. Η ακίδα D11 αντιστοιχεί σε SPI MOSI. Η ακίδα D12 σε MISO SPI. Και η ακίδα D13 σε SPI SCK. Αυτή η επικοινωνία πραγματοποιείται στις ακίδες D50, D51, D52 του Mega. Στον Mega η ακίδα D53 (SS) δεν χρησιμοποιείται για την επιλογή της SD κάρτας, αλλά πρέπει να μείνει ως έξοδος διαφορετικά η SPI επικοινωνία δεν μπορεί να λειτουργήσει.

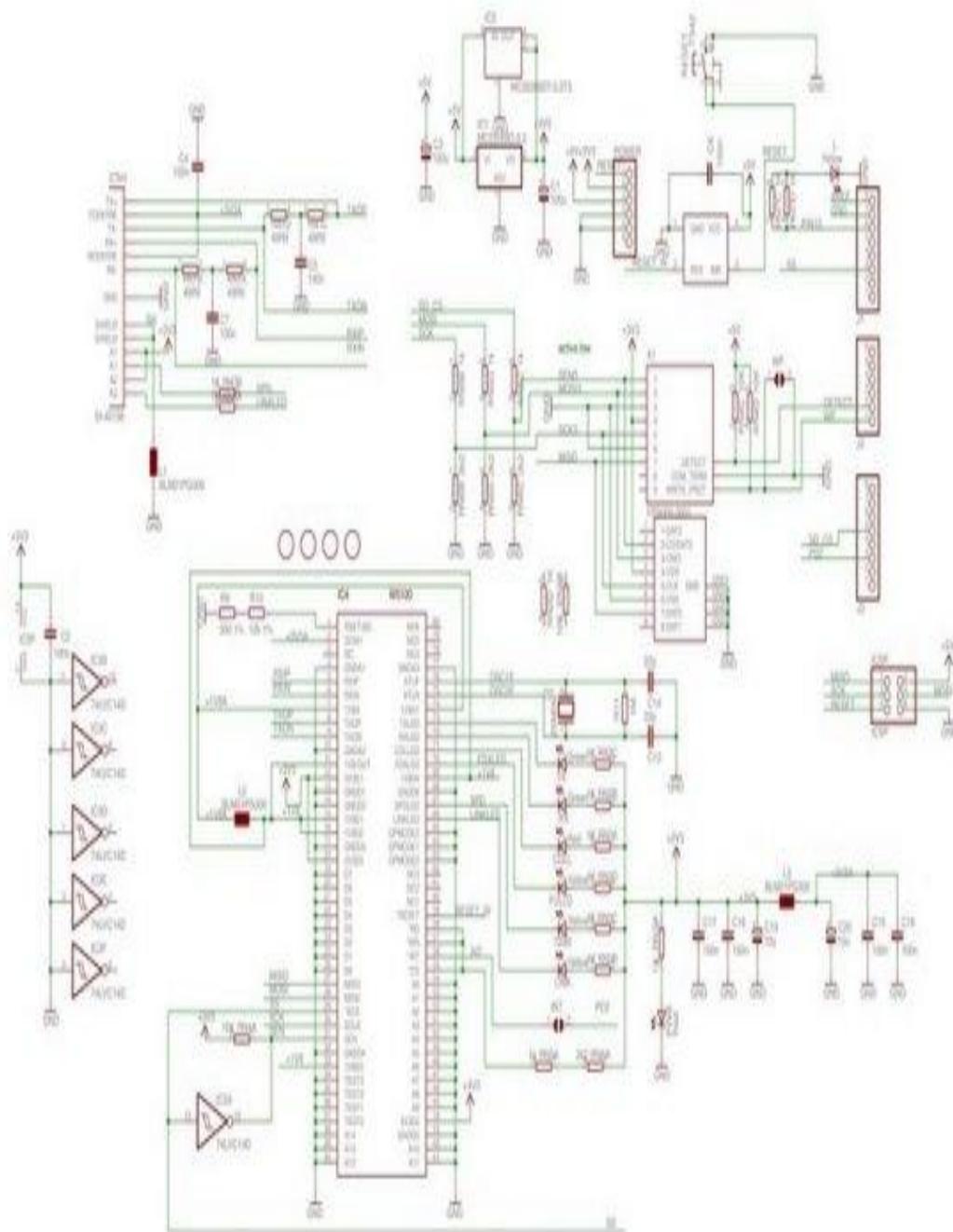
#### **Leds:**

Η πλακέτα περιέχει μια σειρά από ενημερωτικές Leds:

- PWR : Στις ψηφιακές ακίδες δείχνει ότι η πλακέτα τροφοδοτείται με ρεύμα.
- LINK: Δείχνει την παρουσία σύνδεσης δικτύου και αναβοσβήνει όταν η υλοποιείται η αποστολή ή λήψη δεδομένων.
- FULLD: Υποδεικνύει ότι η σύνδεση δικτύου είναι αμφίδρομη (full duplex).
- 100M: Επιβεβαιώνει την παρουσία της σύνδεσης με το δίκτυο ενός 100 Mb/s (σε αντίθεση με τα Mb/s).
- RX: Αναβοσβήνει όταν η πλακέτα λαμβάνει δεδομένα.
- TX: Αναβοσβήνει όταν η πλακέτα στέλνει δεδομένα.
- COLL: Αναβοσβήνει όταν ανιχνεύονται συγκρούσεις δεδομένων.



Εικόνα 40. Το W5100.



### Arduino ETHERNET - shield V5

Copyright (c) 2010 Arduino  
Released under the Creative Commons Attribution-Share Alike 3.0 License  
<http://creativecommons.org/licenses/by-sa/3.0/>

Σχήμα 3. Το σχηματικό του Ethernet Shield.

## 8 ΑΙΣΘΗΤΗΡΕΣ

Σε αυτό το κεφάλαιο θα μελετήσουμε τους αισθητήρες, τα είδη των αισθητήρων και θα αναφερθούμε στους αισθητήρες που θα χρησιμοποιηθούν στην υλοποίηση της παρούσας εργασίας.

### 8.1 Ορισμός αισθητήρων

Με τον όρο αισθητήρα εννοούμε ένα σύστημα το οποίο λαμβάνει ως είσοδο ένα φυσικό μέγεθος και το μετατρέπει, ανάλογα, σε ένα άλλο φυσικό μέγεθος που είναι αναγνωρίσιμο στον παρατηρητή ή στην ηλεκτρική/ μηχανική συσκευή. Σχεδόν σε κάθε συσκευή υπάρχουν αισθητήρες, καθημερινά οι συσκευές παίρνουν μικρές αποφάσεις με βάση την κατάσταση του περιβάλλοντός τους.

### 8.2 Κατηγορίες αισθητήρων

Οι αισθητήρες μπορεί να είναι: μηχανικοί, ηλεκτρονικοί ή και βιολογικοί. Χαρακτηριστικά μεγέθη που καθορίζουν την ποιότητα και τα όρια του κάθε αισθητήρα δίνονται παρακάτω:

- 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.
- Καθυστερήση: 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.

Οι αισθητήρες μπορούν να χωριστούν σε δύο κατηγορίες ανάλογα με το είδος του ηλεκτρονικού σήματος που παρέχουν στην έξοδό τους. Έτσι έχουμε τους:

#### 1. Αισθητήρες αναλογικής εξόδου.

Το σήμα στην έξοδο είναι συνεχές και ανάλογο του πλάτους τάσης του αισθητήρα.

#### 2. Αισθητήρες ψηφιακής εξόδου.

Το σήμα εξόδου είναι ψηφιακό, παίρνει δηλαδή διακριτές τιμές υπό την μορφή δυαδικού αριθμού ή σειράς παλμών.

Η βασική διαφορά ανάμεσα σε αυτές τις δύο κατηγορίες αισθητήρων είναι ότι στην πρώτη κατηγορία έχουμε μεγαλύτερο ποσοστό σφάλματος επειδή το ηλεκτρικό σήμα που δίνουν μπορεί να αλλοιωθεί από τον ηλεκτρικό θόρυβο. (Elgar, 2003)

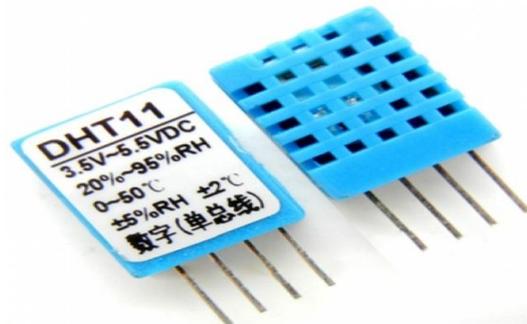
### 8.3 Αισθητήρες της εργασίας

Οι αισθητήρες που χρησιμοποιούνται στη παρούσα εργασία είναι οι ακόλουθοι:

Αισθητήρες	DHT11	DS18B20	DS18B20- Waterproof	BMP180	MPL115A2
Θερμοκρασία	NAI	NAI	NAI	NAI	NAI
Υγρασία	NAI	OXI	OXI	OXI	OXI
Ατμοσφαιρική Πίεση	OXI	OXI	OXI	NAI	NAI

Πίνακας 3. Οι αισθητήρες της εργασίας.

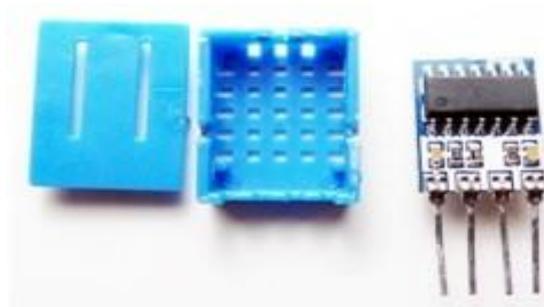
### 8.4 Αισθητήρας DHT11



Εικόνα 41. Ο αισθητήρας DHT11.

Ο αισθητήρας DHT11 είναι ένας φθηνός και αξιόπιστος αισθητήρας. Έχει μικρό μέγεθος, μικρή κατανάλωση ρεύματος και μπορεί να μεταδίδει δεδομένα στην πλακέτα κάθε 2 sec. Το εξωτερικό περίβλημα που προστατεύει τον αισθητήρα είναι από πλαστικό. Λόγω αυτών των ιδιοτήτων του έχει χρησιμοποιηθεί σε πολλές εφαρμογές και υπάρχει πληθώρα βιβλιοθηκών στο διαδίκτυο. Ο αισθητήρας αυτός ακολουθεί το πρωτόκολλο επικοινωνίας του 1-Wire. Η μονόδρομη επικοινωνία γίνεται μέσω ενός μόνο καλωδίου, ανά τακτά χρονικά διαστήματα, στέλνει μόνο τις τιμές των μετρήσεων. Δεν ελέγχει την πληροφορία.

Στη φωτογραφία που ακολουθεί φαίνεται εσωτερικά ο DHT11 και διακρίνεται και το chip που χρησιμοποιείται.

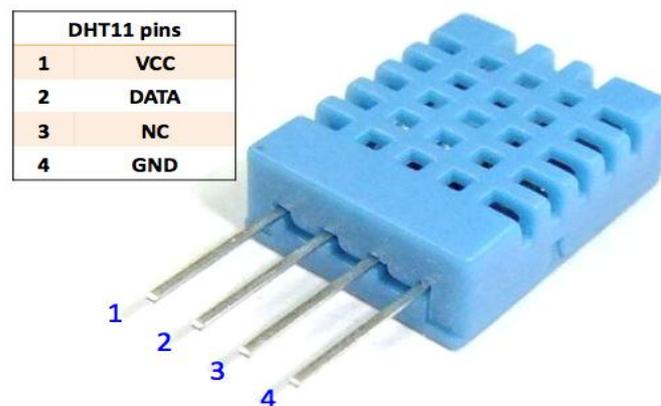


Εικόνα 42. Ο DHT11 εσωτερικά.

Τα τεχνικά χαρακτηριστικά του είναι:

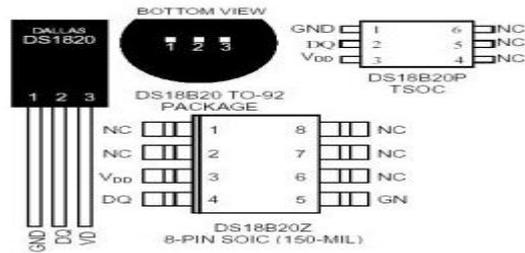
Διασθητήρας DHT11	Υγρασία	Θερμοκρασία
Τάση	3.5-5.5V	3.3-5.5V
Έξοδος	Ψηφιακή	Ψηφιακή
Διαστάσεις	15.5 x 12 x 5.5	15.5 x 12 x 5.5
Εύρος	20% - 95% RH	-40 – 125 °C
Ακρίβεια	± 5 % RH	± 2 °C
Δειγματοληψία	0,5Hz	0,5 Hz

Πίνακας 4. Τα χαρακτηριστικά του DHT11.



Εικόνα 43. Οι ακροδέκτες του DHT11.

## 8.5 Αισθητήρας DS18B20 και DS18B20-Waterproof



Εικόνα 44. Ο DS18B20.

Το ψηφιακό αυτό αισθητήριο ανήκει στην κατηγορία των έξυπνων αισθητήρων, διότι μετατρέπει τη θερμότητα σε ψηφιακή μορφή. Ένα κύριο χαρακτηριστικό του είναι ο μοναδικός 8-Byte ROM αριθμός ταυτότητας που το κάνει να ξεχωρίζει από τα υπόλοιπα αισθητήρια της ίδιας οικογένειας την στιγμή που κατασκευάζεται, δηλαδή το προσωπικό τους ID. Ακολουθεί το πρωτόκολλο επικοινωνίας 1-Wire, το οποίο επιτρέπει να συνδεθούν δύο ή περισσότεροι αισθητήρες πάνω στον ίδιο δίαυλο. Επειδή, λοιπόν η επικοινωνία γίνεται μέσω ενός μοναδικού καλωδίου, η επικοινωνία είναι μονόδρομη και για αυτό ο ίδιος ο αισθητήρας παίρνει πρωτοβουλία και στέλνει τις τιμές. Το πρωτόκολλο 1-Wire δεν επιτρέπει να στέλνουν όλοι μαζί οι αισθητήρες τα δεδομένα τους για να μην αλλοιωθούν. Ο κώδικας μπορεί να διαβάσει και να ξεχωρίσει τους αισθητήρες από τα ID τους και να εμφανίσει τις τιμές μέτρησης στον αντίστοιχο αισθητήρα.

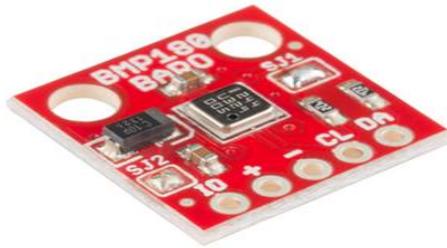
Αισθητήρας	DS18B20	DS18B20-Waterproof
Τάση	3-5.5V	3-5.5V
Εύρος	-55°C – 125°C	-55°C – 125°C
Ακρίβεια	+/- 0.5 °C	+/- 0.5 °C
Διαστάσεις	1 x 0.5 mm	~1m

Πίνακας 5. Τα χαρακτηριστικά των DS18B20 και DS18B20-Waterproof.



Εικόνα 45. Ο DS18B20-Waterproof.

## 8.6 Αισθητήρας BMP180



Εικόνα 46. Ο BMP180.

Ο αισθητήρας βαρομετρικής πίεσης BMP180 της εταιρίας Bosch, έχει πάνω του ένα chip που του επιτρέπει να μετράει την βαρομετρική πίεση της ατμόσφαιρας και την θερμοκρασία. Η πίεση αλλάζει σύμφωνα με το υψόμετρο, έτσι ο αισθητήρας αυτός δίνει την δυνατότητα να υπολογιστεί και το υψόμετρο. Ο αισθητήρας είναι αρκετά μικρός σε μέγεθος και είναι κολλημένος σε PCB με ρυθμιστή τάσης 3.3V, I<sup>2</sup>C level shifter και pull-up αντίσταση στο I<sup>2</sup>C ακροδέκτη. Στο εσωτερικό του υπάρχει ένας αισθητήρας πιεζοηλεκτρικής αντίστασης, ένας αναλογικός σε ψηφιακός μετατροπέας και μία μονάδα ελέγχου με EEPROM και σειριακή I<sup>2</sup>C διεπαφή. Η EEPROM αποθηκεύει 176 bit από ξεχωριστά δεδομένα που υπάρχουν για το setup. (UP = δεδομένα πίεσης 16 έως 19 bit, UT= δεδομένα θερμοκρασίας 16 bit). Η I<sup>2</sup>C είναι μία αμφίδρομη επικοινωνία, υπάρχει ένας πομπός και ένας δέκτης όπου στέλνονται και λαμβάνονται δεδομένα για αυτό αυτή η επικοινωνία δεν έχει σφάλματα όπως μπορεί να διαπιστώνονται σε άλλα πρωτόκολλα επικοινωνίας.

Το chip που έχει πάνω του ο αισθητήρας BMP180 είναι ο παρακάτω.



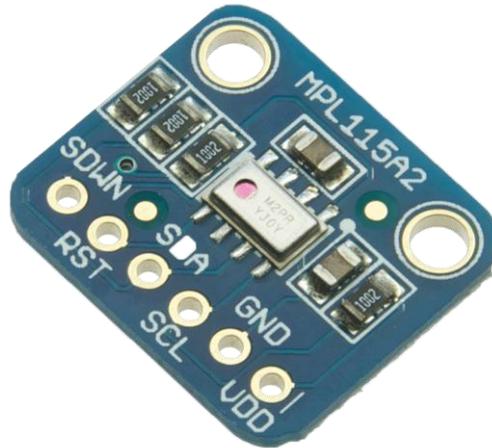
Εικόνα 47. Το chip του bmp180.

Τεχνικές προδιαγραφές είναι:

Αισθητήρας	BMP180
Τάση	1.8V – 5V
Συμβατική Τάση	3.3V
Εύρος Πίεσης	300 – 1100 hPa
Απόκλιση Πίεσης	0.02 hPa
Εύρος Θερμοκρασίας	-40 - 85 °C
Απόκλιση θερμοκρασίας	+/- 2 °C
Διαστάσεις	1.5mm x 1.5mm

Πίνακας 6. Τα χαρακτηριστικά του BMP180.

## 8.7 Αισθητήρας MPL115A2



Εικόνα 48. Ο MPL115A2.

Ο αισθητήρας MPL115A2 έχει σχεδιαστεί για την μέτρηση ειδικά της ατμοσφαιρικής πίεσης αλλά και της θερμοκρασίας της ατμόσφαιρας. Επίσης, λόγω της συσχέτισης της ατμοσφαιρικής πίεσης και του ύψους στην οποία γίνεται η μέτρηση, η βιβλιοθήκη παρέχει και την επιπλέον λειτουργία, της ένδειξης του τρέχοντος υψομέτρου της περιοχής του αισθητήρα. Ακολουθεί το πρωτόκολλο επικοινωνίας I<sup>2</sup>C.

Το chip του MPL115A2 φαίνεται στην ακόλουθη φωτογραφία.



Εικόνα 49. Το chip του MPL115A2.

Τεχνικές προδιαγραφές είναι:

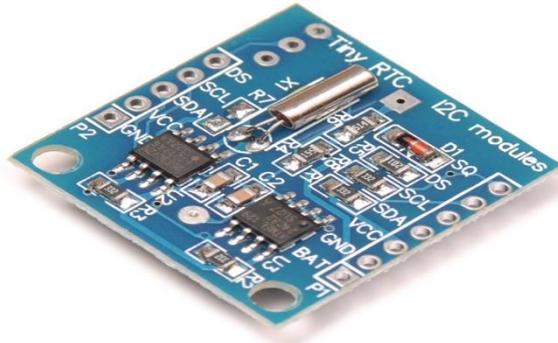
Αισθητήρας	MPL115A2
Τάση (Συνεχές ρεύμα)	2.4 – 5 V
Τάση	3 – 5 V
Εύρος Πίεσης	500 – 1150 hPa
Εύρος Θερμοκρασίας	-40 °C – 105 °C
Ευαισθησία	+/- 1 kPa /0.25m
Διαστάσεις	1.5mm x 1.5mm

Πίνακας 7. Τα χαρακτηριστικά του MPL115A2.

## 9 ΠΕΡΙΦΕΡΕΙΑΚΟΣ ΕΞΟΠΛΙΣΜΟΣ

Σε αυτό το κεφάλαιο παρουσιάζονται τα επιπλέον εξαρτήματα που χρησιμοποιούνται στην υλοποίηση, καθώς η παρουσία τους είναι πολύ σημαντική.

### 9.1 Το εξωτερικό ρολόι Tiny Real Time Clock\_DS1307



Εικόνα 50. Ο DS1307.

Το Tiny Real Time Clock\_DS1307 είναι ένα χαμηλού κόστους και ρολόι ακριβείας πραγματικού χρόνου (RTC). Η συσκευή με την βοήθεια μπαταρίας μπορεί να συγκρατήσει τον χρόνο όταν ακόμα και η τροφοδοσία στην κύρια συσκευή σταματήσει. Η ενσωμάτωση του κρυσταλλικού αντηχείου εγγυάται την μακροχρόνια ακρίβεια της συσκευής καθώς και την μείωση του κόστους.

Το RTC δίνει πληροφορίες για δευτερόλεπτα, λεπτά, ώρες, ημέρα, ημερομηνία, μήνα και έτος. Η ημερομηνία ρυθμίζεται αυτόματα για μήνες με λιγότερες από 31 ημέρες. Το ρολόι λειτουργεί σε μορφή είτε 12 είτε 24 ωρών με MM/ΠΜ δείκτη. Οι διευθύνσεις και τα δεδομένα μεταφέρονται σειριακά μέσω του I<sup>2</sup>C διαύλου. Όταν προγραμματιστεί και τρέξει το πρόγραμμα, τότε ο DS1307 εναρμονίζεται με το ρολόι του υπολογιστή και δείχνει την ίδια ώρα.

Τεχνικές διαγραφές είναι:

Εξωτερικό Ρολόι	Real Time Clock_DS1307
Τάση	4.5 – 5.5 V
Διαστάσεις	3mm x 3mm

Πίνακας 8. Τα χαρακτηριστικά του RTC\_DS1307.



Εικόνα 51. Ο DS1307 εσωτερικά.

## 9.2 Micro\_SD Memory Card 4GB

Η Ethernet Shield χρησιμοποιείται για τη διασύνδεση του Arduino με το δίκτυο, καθώς περιλαμβάνει έναν αναγνώστη καρτών Micro-SD (card reader). Αυτό μας επιτρέπει να εφοδιάσουμε τη μονάδα με μια τεράστια ποσότητα περιφερειακής μνήμης απλά τοποθετώντας μια κάρτα Micro-SD στην υποδοχή του αναγνώστη, ενώ σε συνδυασμό με τις κατάλληλες βιβλιοθήκες ο Arduino μπορεί να αποκτήσει πρόσβαση στο σύστημα αρχείων (file system) της κάρτας. Οι σύγχρονες κάρτες Micro-SD διαθέτουν χωρητικότητες της τάξης των μερικών GB ενώ κοστίζουν μόλις μερικά ευρώ.



Εικόνα 52. Η Micro\_SD Card.

Για να χρησιμοποιήσουμε την εξωτερική Micro-SD κάρτα μέσα από ένα πρόγραμμα για Arduino θα πρέπει να συμπεριλάβουμε πρώτα την βιβλιοθήκη ETHERNET.h, η οποία βρίσκεται προεγκατεστημένη στο IDE του Arduino και περιέχει όλες τις συναρτήσεις για την διασύνδεση της κάρτας με το Arduino. Στην συνέχεια ορίζουμε την διεύθυνση IP του δικτύου, το υποδίκτυο στο οποίο ανήκει ο σταθμός, την διεύθυνση MAC της κάρτας, αρχικοποιούμε το αντικείμενο και ορίζουμε την πόρτα στην οποία θα ανταποκρίνεται ο εξυπηρετητής. Εξ' ορισμού η πόρτα στην οποία ανταποκρίνονται όλες οι αιτήσεις του HTTP πρωτοκόλλου είναι η 80. Στη συνάρτηση setup() γίνεται η αρχικοποίηση και έναρξη λειτουργίας της κάρτας.

## 10 TO SOFTWARE ΤΗΣ ΕΡΓΑΣΙΑΣ

Στο κεφάλαιο αυτό θα αναφέρουμε τις γλώσσες προγραμματισμού που χρησιμοποιήθηκαν καθώς και τα εργαλεία ανάπτυξης λογισμικού.

### 10.1 HTML

Η HTML (Hyper Text Markup Language) είναι μία γλώσσα ειδικής χρήσης για τη δημιουργία ιστοσελίδων στο χώρο του διαδικτύου. Η σύνταξη αυτής της γλώσσας δεν χρειάζεται εξειδικευμένο εργαλείο ανάπτυξης, μπορεί να δημιουργηθεί σε ένα σημειωματάριο ή ένα notepad++ και τα αρχεία που δημιουργούνται αποθηκεύονται αρχικά ως .txt. Ο κώδικας που έχει γραφτεί για να μπορεί να διαβαστεί από ένα πρόγραμμα περιήγησης, όπως ο Internet Explorer, ώστε να εμφανιστεί στον χρήστη ως η επιθυμητή πληροφορία, πρέπει να αποθηκευτεί το αρχείο κειμένου ως .htm ή .html.

Η γλώσσα HTML χρησιμοποιεί ετικέτες (tags), οι οποίες αποτελούν λέξεις-κλειδιά και εσωκλείονται ανάμεσα στα σύμβολα '<' και '>', μία από αυτές είναι η <html> και </html>, που δηλώνουν την αρχή και το τέλος του πραγματικού κώδικα. Οι ετικέτες της HTML γλώσσας λειτουργούν ανά ζεύγη όπως ήδη είδαμε. Ανάμεσα στις ετικέτες, οι σχεδιαστές ιστοσελίδων μπορούν να τοποθετήσουν κείμενο, πίνακες, γραφικά, ήχο, βίντεο και να διαμορφώσουν την σελίδα όπως θέλουν αυτοί.

Για να προβληθεί το τελικό αποτέλεσμα από διάφορους χρήστες, θα πρέπει να εγκατασταθεί ένας Web Server σε έναν υπολογιστή, ώστε να φιλοξενούνται οι ιστοσελίδες. Αυτό συνήθως γίνεται τοπικά στον υπολογιστή του προγραμματιστή και έπειτα ανεβάζονται οι σελίδες στον Web Server για να δημοσιευτούν στο διαδίκτυο.

Εύστοχη και απαραίτητη κρίνεται και η αναφορά του πρωτοκόλλου επικοινωνίας HTTP (Hyper Text Transfer Protocol) που χρησιμοποιείται από τα προγράμματα περιήγησης για να επιτευχθεί η ασφαλής προσπέλαση των ιστοσελίδων από τους χρήστες του διαδικτύου. Το πρωτόκολλο αυτό είναι το πιο διαδεδομένο πρωτόκολλο στα προγράμματα περιήγησης. (Τσουροπλής, Κλημόπουλος, 2005)

### 10.2 XML

Η γλώσσα XML (eXtensive Markup Language) είναι μία καινούργια μέθοδος για το σχεδιασμό των ιστοσελίδων. Η XML ασχολείται με την δόμηση των δεδομένων της σελίδας, ενώ η HTML με την παρουσίαση των δεδομένων.

Η XML στοχεύει στην ανταλλαγή των δεδομένων. Σε αυτό βοηθάει και η σύνταξη της γλώσσας αυτής, η οποία χρησιμοποιεί ετικέτες (tags) που επίσης εσωκλείονται στα σύμβολα '<' και '>'. Τα στοιχεία που περιέχουν αυτές οι ετικέτες μπορούν να χρησιμοποιηθούν και από άλλα προγράμματα και να υλοποιηθεί έτσι μία ανταλλαγή δεδομένων μέσω του διαδικτύου. (Τσουροπλής, Κλημόπουλος, 2005)

### 10.3 CSS

Η CSS (Cascading Style Sheets) είναι μια γλώσσα που δημιουργήθηκε για την διευκόλυνση της μορφοποίησης των σελίδων, για αυτό και ονομάστηκε φύλλα επάλληλων στυλ. Ο συνδυασμός της με τις γλώσσες HTML και XHTML κάνουν τον έλεγχο της εμφάνισης μιας



## 10.5 PHP



Εικόνα 53. PHP.

Η PHP (Hypertext Preprocessor) είναι μια γλώσσα προγραμματισμού για την ανάπτυξη δυναμικών ιστοσελίδων. Ο κώδικας PHP πρέπει να ελέγχεται πρώτα από έναν Apache, ώστε να μεταγλωττίζει τον κώδικα αυτόν σε HTML. (Τσουροπλής, Κλημόπουλος, 2005)

Η PHP μπορεί:

1. Να παράγει δυναμικό περιεχόμενο σελίδων.
2. Να δημιουργήσει, να ανοίξει, να διαβάσει, να γράψει και να κλείσει αρχεία στο διακομιστή.
3. Να συλλέξει δεδομένα φόρμας.
4. Να στείλει και να πάρει cookies.
5. Να προσθέσει, να διαγράψει, να τροποποιήσει τα δεδομένα στη βάση δεδομένων.
6. Να περιορίσει τους χρήστες να έχουν πρόσβαση σε ορισμένες σελίδες στην ιστοσελίδα.
7. Να κρυπτογραφήσει δεδομένα.

## 10.6 SQL



Εικόνα 54. SQL.

Η SQL (Structured Query Language) είναι μια δομημένη γλώσσα επερωτήσεων για τη διαχείριση βάσεων δεδομένων. Διαθέτει εντολές ορισμό δεδομένων, ερωτήσεις και ενημερώσεις. Είναι μία γλώσσα ορισμού και διαχείρισης δεδομένων.

Τα δεδομένα στην SQL τοποθετούνται μέσα σε πίνακες, οι οποίοι αποτελούνται από στήλες και γραμμές. Οι βάσεις δεδομένων είναι χρήσιμες για την αποθήκευση πληροφοριών σε κατηγορίες. (Elsmani, Navathe, 2007) Ορισμένες από τις βασικότερες εντολές SQL αναφέρονται παρακάτω:

1. **SELECT**: Εξάγει δεδομένα από μια βάση δεδομένων.
2. **UPDATE**: Ενημερώνει δεδομένα σε μια βάση δεδομένων.

3. **DELETE**: Διαγράφει δεδομένα από μια βάση δεδομένων.
4. **INSERT INTO**: Εισάγει δεδομένα σε μια βάση δεδομένων.
5. **CREATE DATABASE**: Δημιουργεί μια βάση δεδομένων.
6. **ALTER DATABASE**: Τροποποιεί μια βάση δεδομένων.
7. **CREATE TABLE**: Δημιουργεί έναν νέο πίνακα.
8. **ALTER TABLE**: Τροποποιεί έναν πίνακα.
9. **DROP TABLE**: Διαγράφει έναν πίνακα.
10. **CREATE INDEX**: Δημιουργεί έναν δείκτη (κλειδί αναζήτησης).
11. **DROP INDEX**: Διαγράφει έναν δείκτη.

## 10.7 JAVA

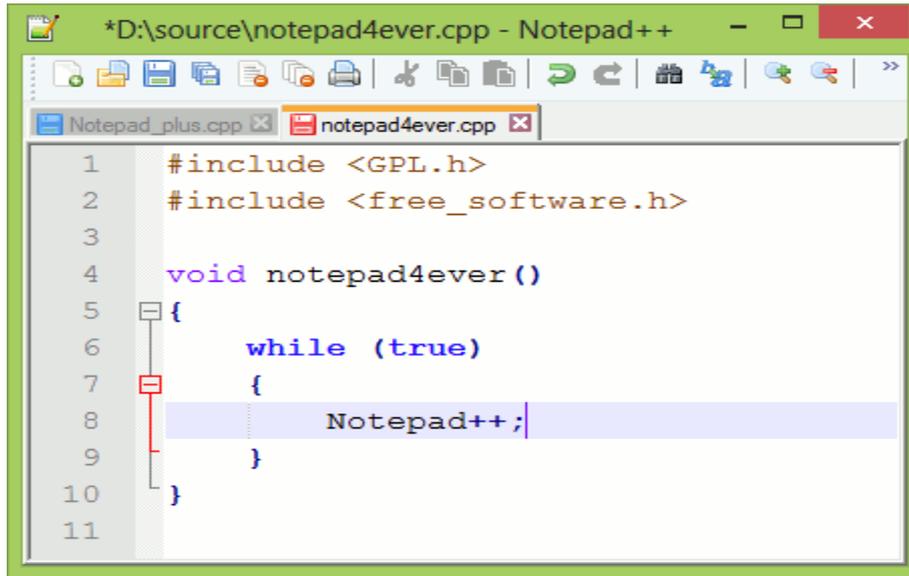
Η JAVA είναι μία αντικειμενοστραφής γλώσσα προγραμματισμού, η οποία βασίζεται στη γλώσσα C++ και σκοπός της είναι να τρέχει τα προγράμματά της σε οποιονδήποτε υπολογιστή, ανεξάρτητα από το λειτουργικό σύστημα που αυτός υποστηρίζει. Η JAVA μπορεί να διαχειριστεί δεδομένα όπως, κείμενο, γραφικά, ήχο ή βίντεο από το ίδιο πρόγραμμα. Επιτρέπει ακόμη, την ανάπτυξη εφαρμογών μέσω του διαδικτύου. (Τσουροπλής, Κλημόπουλος, 2005)

```
function myFunction()
{
  var x="";
  var time=new Date().getHours();
  if (time<20)
  {
    x="Good day";
  }
  document.getElementById("demo").innerHTML=x;
}
```

Εικόνα 55. Μία function-συνάρτηση της JAVA.

## 10.8 NOTEPAD++

Το notepad++ είναι ένα πρόγραμμα επεξεργασίας κειμένου ανοικτού κώδικα. Μπορεί να αντικαταστήσει το απλό σημειωματάριο και να επεξεργαστεί αρχεία σε πολλές γλώσσες προγραμματισμού παρέχοντας διάφορα εργαλεία και διευκολύνσεις στον προγραμματιστή. Διανέμεται ελεύθερα στο διαδίκτυο και η χρήση του είναι εύκολη.



```
*D:\source\notepad4ever.cpp - Notepad++
Notepad_plus.cpp x notepad4ever.cpp x
1 #include <GPL.h>
2 #include <free_software.h>
3
4 void notepad4ever ()
5 {
6     while (true)
7     {
8         Notepad++;
9     }
10 }
11
```

Εικόνα 56. NOTEPAD++.

## 10.9 XAMPP



Εικόνα 57. XAMPP.

Το XAMPP είναι μία σύνθεση από προγράμματα ελεύθερου λογισμικού. Λειτουργεί για οποιαδήποτε πλατφόρμα. Περιέχει έναν εξυπηρετητή ιστοσελίδων HTTP Apache, ένα σύστημα διαχείρισης των βάσεων δεδομένων MySQL και ένα διερμηνέα για PHP και Perl. Σκοπός του είναι η ανάπτυξη και δοκιμή ιστοσελίδων στον τοπικό υπολογιστή.

## 11 ΤΟ HARDWARE ΤΗΣ ΕΡΓΑΣΙΑΣ

Αυτό το κεφάλαιο αναφέρεται στο hardware υλικό που χρησιμοποιήθηκε για την υλοποίηση της παρούσας εργασίας και το κοστολόγιο για την αγορά των υλικών.

### 11.1 Το απαιτούμενο υλικό

Τα σημαντικότερα μέρη για τη δημιουργία του μετεωρολογικού σταθμού αποτελούνται από τις 2 αναπτυξιακές πλακέτες Arduino, τους αισθητήρες και το Real Time Clock.

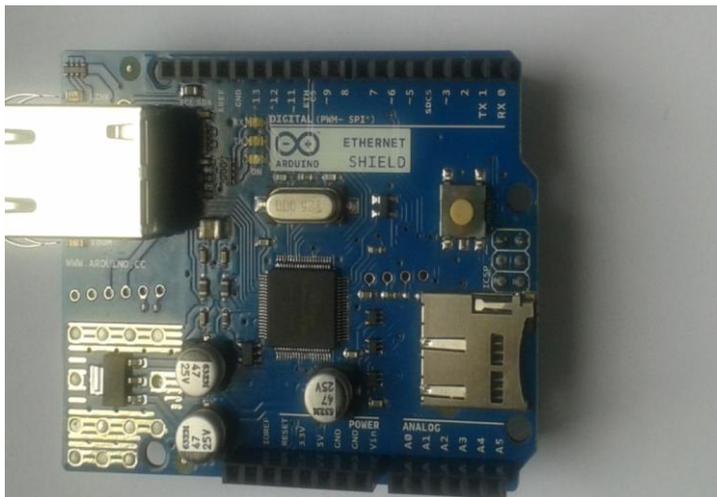
Αναλυτικότερα, τα υλικά είναι:

- **1 Πλακέτα Arduino Mega 2569 Rev3.**



- **1 Πλακέτα Arduino Ethernet Shield.**

Η πλακέτα αυτή κουμπώνει πάνω στην πλακέτα του Arduino Mega 2560.

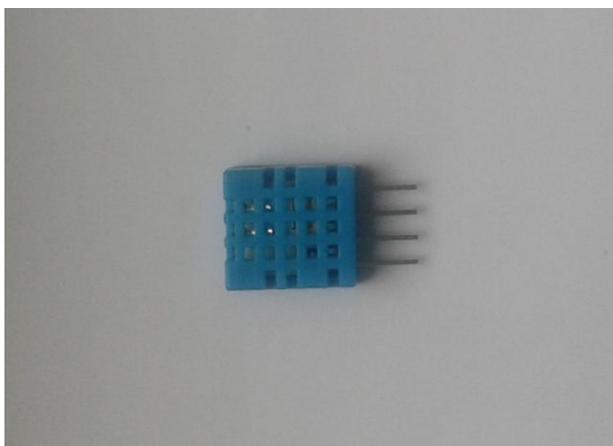


- **1 USB 2.0 καλώδιο 5 μέτρων (A to B).**

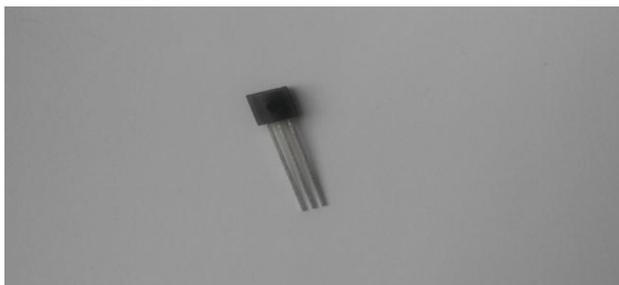
Ο δίαυλος επικοινωνίας USB είναι μια σύνδεση του υπολογιστή με τις περιφερειακές συσκευές.



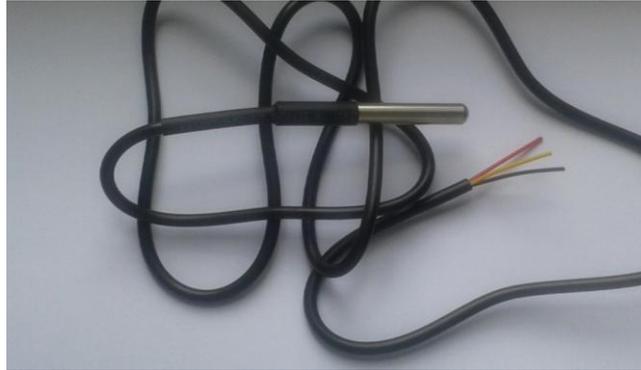
- **1 Humidity Sensor DHT11.**



- **1 Temperature Sensor DS18B20.**



- **1 Temperature Sensor DS18B20- Waterproof**



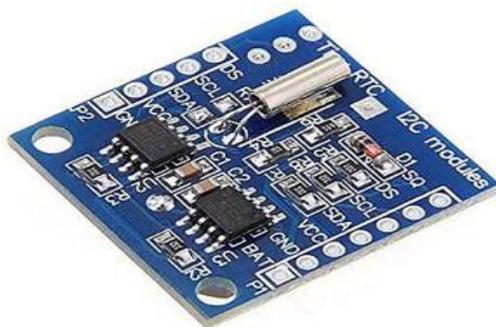
- **Barometric Pressure Sensor- BMP180**



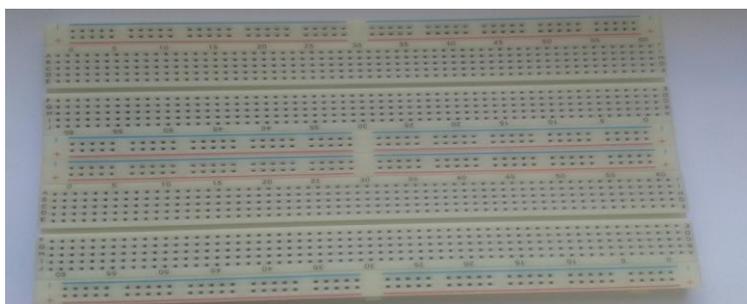
- **MPL115A2 – I<sup>2</sup>C Barometric pressure/Temperature Sensor**



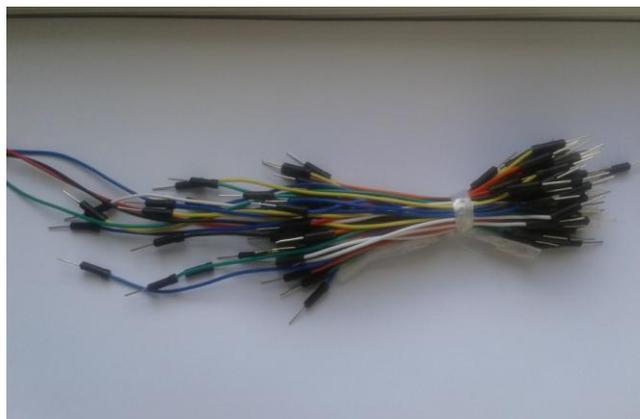
- 1 Real Time Clock\_DS1307.



- 1 Breadboard 830 Tie.



- 1 Breadboard Jumper Cables 65 κομματιών M/M.



- **1 Resistor Carbon ¼ 10 Kohm.**



- **1 Resistor Carbon ¼ 4.7 Kohm.**



- **1 Micro SD Memory Card 4GB.**

Θα τοποθετηθεί στην Ethernet Shield.



- **Pin Header 1x40 Male 2.54 mm black.**

Αυτά τα pin χρειάζονται να συγκολληθούν με τους MPL115A2 και BMP180.



• **1 Μπαταρία CR2032 3V.**

Θα τοποθετηθεί στο RTC\_DS1307.



## 11.2 Το κοστολόγιο του απαιτούμενου υλικού

Στον παρακάτω πίνακα παρουσιάζεται το κοστολόγιο της αγοράς των υλικών.

A/A	ΠΕΡΙΓΡΑΦΗ ΥΛΙΚΟΥ	ΠΟΣΟΤΗΤΑ	ΤΙΜΗ (23%ΦΠΑ)
1	Arduino Mega 2560 Rev3	1	45,000 €
2	Arduino Ethernet Shield	1	30,000 €
3	Humidity Sensor DHT11	1	3,250 €
4	Temperature Sensor DS18B20	1	2,440 €
5	Temperature Sensor DS18B20-Waterproof	1	6,500 €
6	Barometric Sensor – BMP180	1	7,320 €
7	MPL115A2 – I <sup>2</sup> C BAROM./TEMP. SEN.	1	6,500 €
8	Breadboard 830 Tie	2	9,020 €
9	Breadboard Jumper Cables 65 pcs m/m	1	3,900 €
10	Resistor Carbon ¼ 10Kohm	10	0,010 €
11	Resistor Carbon ¼ 4.7Kohm	10	0,010 €
12	USB Cable 2.0 A-B Male 5m	1	2,030 €
13	Micro SD Memory Card 4GB	1	6,500 €
14	RTC_DS1307	1	3,000 €
15	CR2032	1	2,000 €
16	Pin Header 1X40 Male 2.54 mm black	1	0,400 €

Πίνακας 9. Το κοστολόγιο των υλικών της κατασκευαστικής εργασίας.

**Συνολικό Ποσό: 112,36 €**

Αυτό το ποσό χρειάστηκε για την αγορά των υλικών της εργασίας μου.

Υπάρχει, όμως και το κόστος αγοράς των ψηφιακών θερμομέτρων για να πραγματοποιηθεί η σύγκριση τιμών.

A/A	ΕΙΔΗ ΘΕΡΜΟΜΕΤΡΩΝ	ΠΟΣΟΤΗΤΑ	ΤΙΜΗ (23%ΦΠΑ)
1	Avidsen	1	6 €
2	TA-298	1	9,5 €
3	T2-368	1	8 €
<b>ΣΥΝΟΛΙΚΟ ΠΟΣΟ:</b>		<b>3</b>	<b>13,5 €</b>

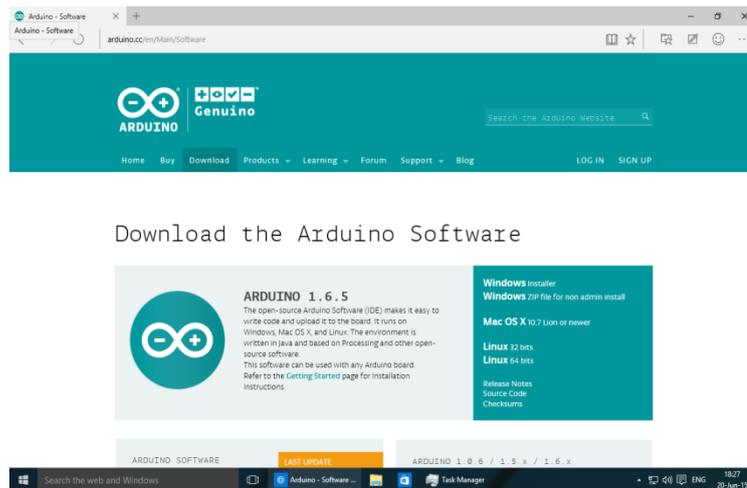
Πίνακας 10. Το κοστολόγιο της αγοράς των ψηφιακών θερμομέτρων.

## 12 ΚΑΤΑΣΚΕΥΗ /ΥΛΟΠΟΙΗΣΗ

Σκοπός αυτού του κεφαλαίου είναι να παρουσιαστεί η σταδιακή εξέλιξη της κατασκευής της εργασίας μου με το hardware υλικό που αναφέρθηκε στο προηγούμενο κεφάλαιο.

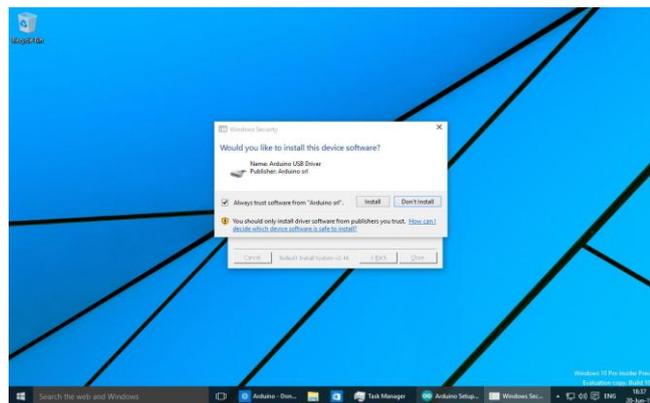
### 12.1 Εγκατάσταση λογισμικού

Πηγαίνουμε στο <http://arduino.cc/download> και κατεβάζουμε τον Windows Installer :



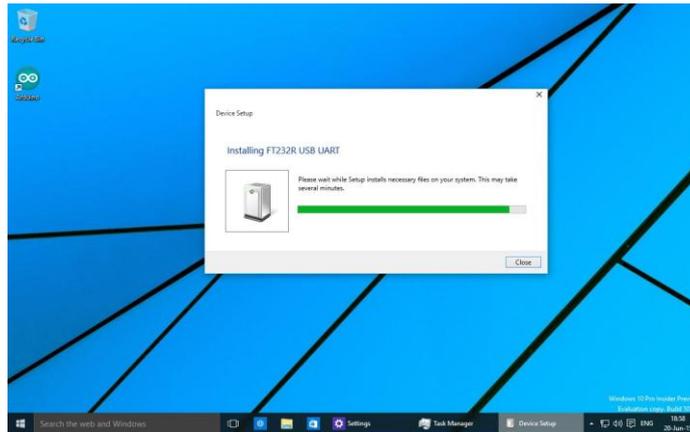
Εικόνα 58. Επιλογή προγράμματος εγκατάστασης του Arduino.

Όταν εγκαθίσταται, εμφανίζεται ένα παράθυρο στο οποίο πρέπει να επιλέξουμε 'I Agree' στο 'License Agreement' και μετά next, next, χωρίς να αλλάξουμε κάποια από τις επιλογές που μας δίνονται, μέχρι να ολοκληρωθεί η εγκατάσταση. Κάποια στιγμή θα μας ρωτήσει αν θέλουμε να εγκαταστήσουμε τον USB Driver του Arduino, επιλέγουμε 'Install' και συνεχίσουμε.



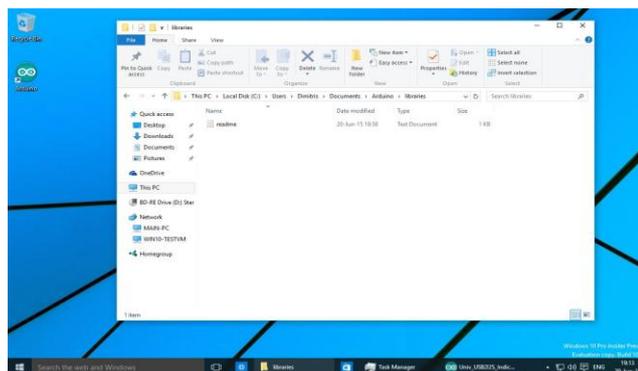
Εικόνα 59. Εγκατάσταση.

Έχει ολοκληρωθεί η εγκατάσταση και ένα εικονίδιο Arduino εμφανίζεται στην επιφάνεια εργασίας του υπολογιστή. Στη συνέχεια συνδέουμε με USB το Arduino μας. Βλέπουμε να εγκαθίστανται οι drivers:



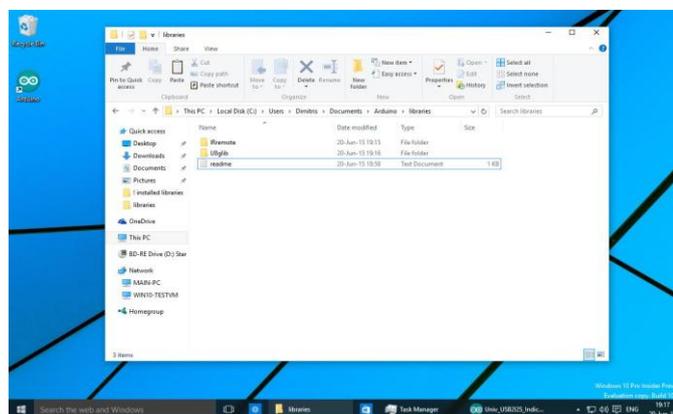
Εικόνα 60. Εγκατάσταση των drivers.

Όταν ολοκληρωθεί η διαδικασία είμαστε έτοιμοι να ανοίξουμε το περιβάλλον του Arduino. Πριν ξεκινήσουμε χρειάζεται να αποθηκεύσουμε στο μηχάνημα τις βιβλιοθήκες που μας είναι απαραίτητες. Οι εν λόγω libraries πρέπει να περαστούν στον φάκελο C:\Users\xxxxxxx\Documents\Arduino\libraries, όπου «xxxxxxx» το user name μας στον υπολογιστή.



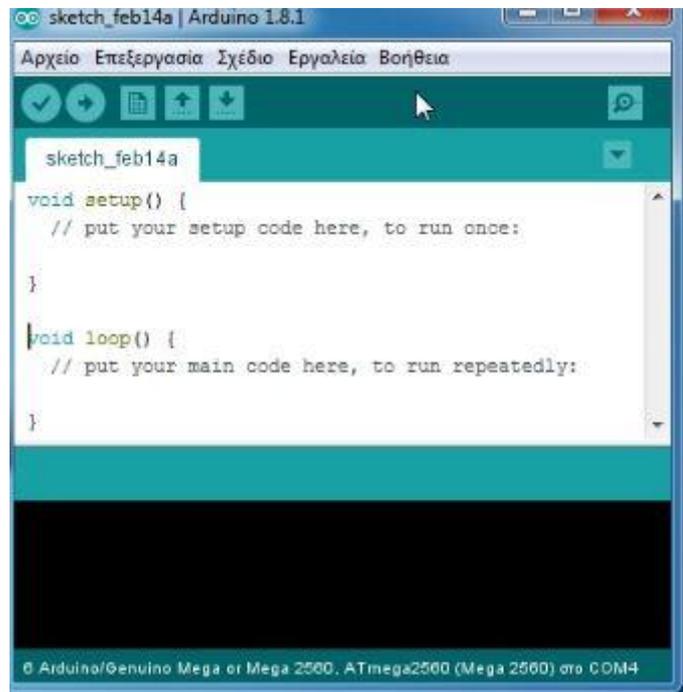
Εικόνα 61. Εγκατάσταση των βιβλιοθηκών.

Παίρνουμε λοιπόν τους φακέλους που χρειαζόμαστε και τους αποθηκεύουμε εκεί:



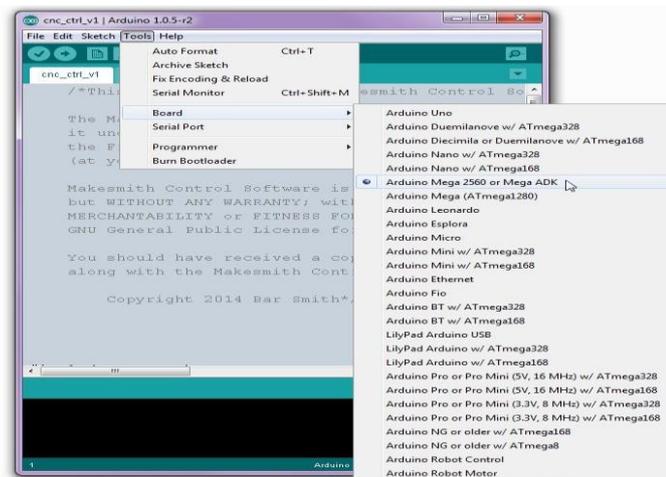
Εικόνα 62. Εγκατάσταση των βιβλιοθηκών.

Μπορούμε πλέον να ανοίξουμε το πρόγραμμα του Arduino με ένα διπλό κλικ στο εικονίδιο του και εμφανίζεται μπροστά μας το περιβάλλον του όπως φαίνεται παρακάτω.

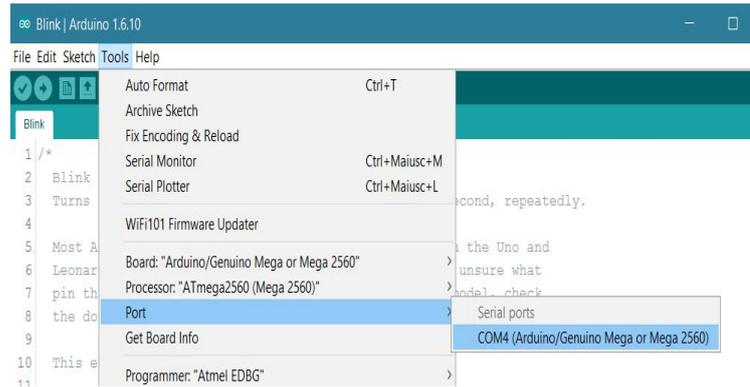


Εικόνα 63. Το περιβάλλον του IDE.

Τώρα πρέπει να επιλέξουμε το Arduino Mega on Mega 2560 καθώς και την σειριακή θύρα COM port στην οποία πρέπει να συνδεθεί για να λειτουργήσει. Πηγαίνουμε από το μενού Tools -> Board και επιλέγουμε το Mega2560 και Tools -> Serial Port -> COM 4.

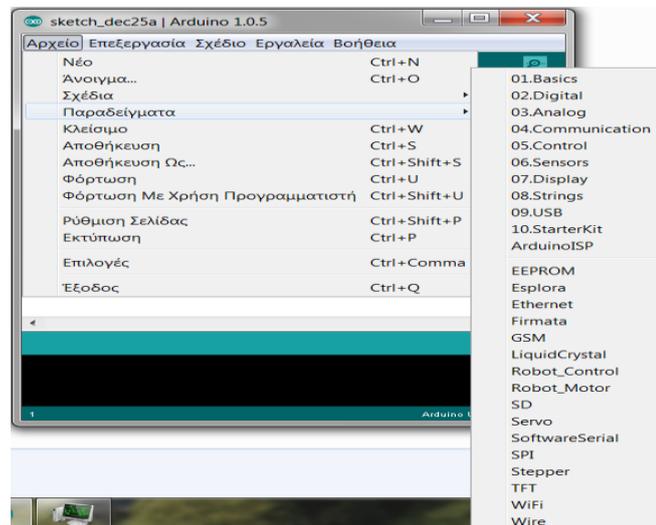


Εικόνα 64. Επιλογή του Arduino Mega2560.



Εικόνα 65. Επιλογή της σειριακής θύρας 4.

Τώρα μπορούμε να δοκιμάσουμε να τρέξουμε κάποιο από τα παραδείγματα που έχει ήδη. Από Αρχείο -> Παραδείγματα.



Εικόνα 66. Τα εγκατεστημένα παραδείγματα.

Έχοντας επιλέξει το παράδειγμα που θέλουμε, πατάμε από την εργαλειοθήκη τα verify και upload για να αποθηκευτεί στον μικροελεγκτή ο διορθωμένος κώδικας και πατώντας το κουμπί serial monitor, εμφανίζεται δίπλα στην οθόνη μας μία ακόμη οθόνη που εμφανίζονται τα σειριακά δεδομένα που αποστέλλονται από την πλακέτα του Arduino. Στο κάτω μέρος της σειριακής οθόνης υπάρχει ένα κουμπί που επιτρέπει να επιλεγεί ο ρυθμός μετάδοσης των δεδομένων (baud), σύμφωνα με ότι έχει δηλωθεί μέσα στην παρένθεση της γραμμής Serial.begin() του προγράμματος.



Εικόνα 67. Η σειριακή οθόνη.

## 12.2 Κούμπωμα των 2 arduino πλακετών

Αφού, ολοκληρώθηκε η εγκατάσταση του λογισμικού για τον προγραμματισμό των arduino, τώρα κουμπώνουμε τα 2 arduino ,καθώς τα pins που διαθέτει ο Ethernet shield arduino τα αντίστοιχα διαθέτει και ο Arduino Mega on Mega2560.



Εικόνα 68. Το κούμπωμα των 2 πλακετών.

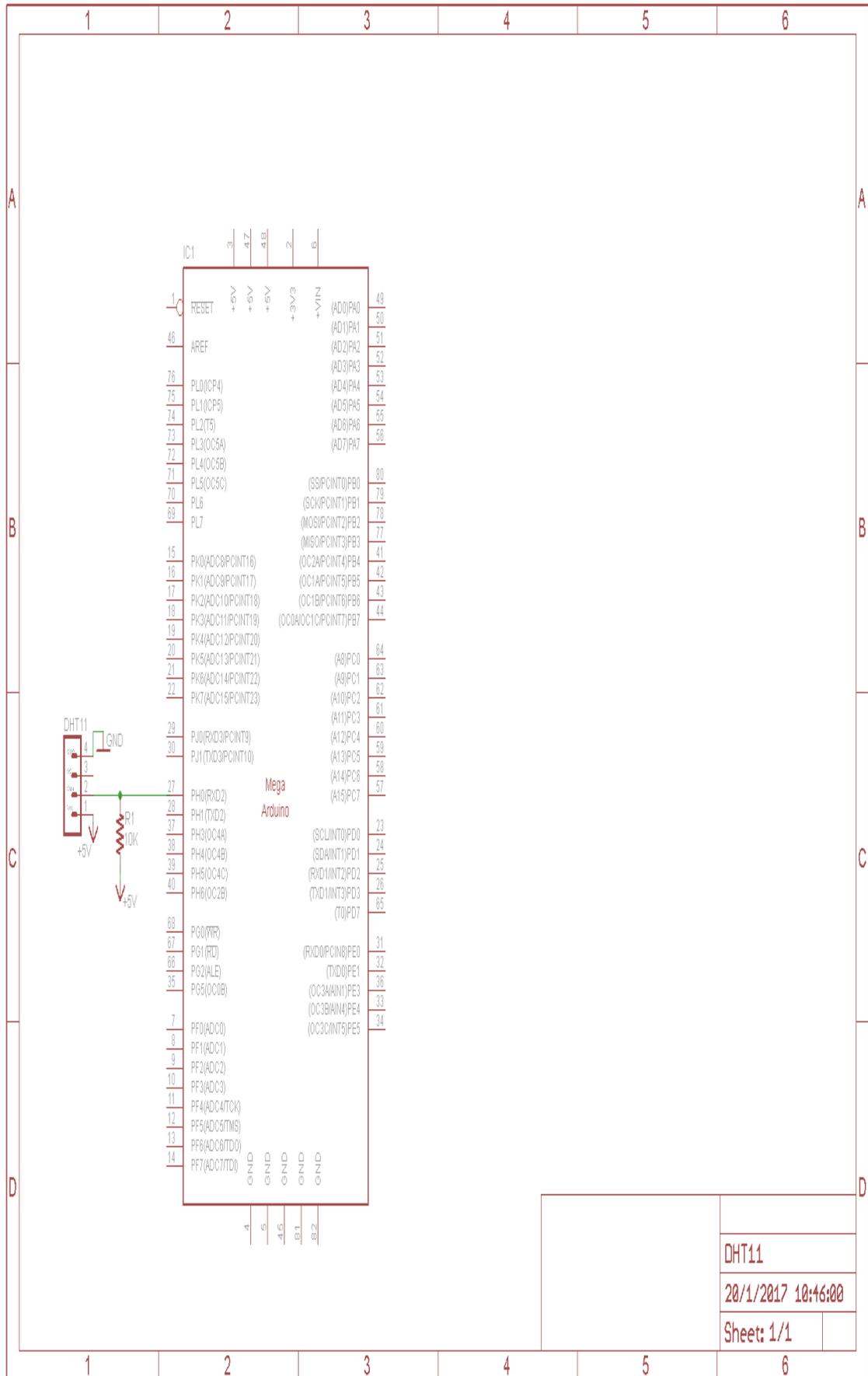
## 12.3 Συνδεσμολογία της πλακέτας με τον αισθητήρα DHT11

Ξεκινάει η σταδιακή κατασκευή της τελικής μου εργασίας.

Μια πολύ σημαντική σημείωση για την μέθοδο συνδεσμολογίας οποιονδήποτε ηλεκτρονικών υλικών είναι να συνδέουμε **πάντα πρώτα την γείωση**.

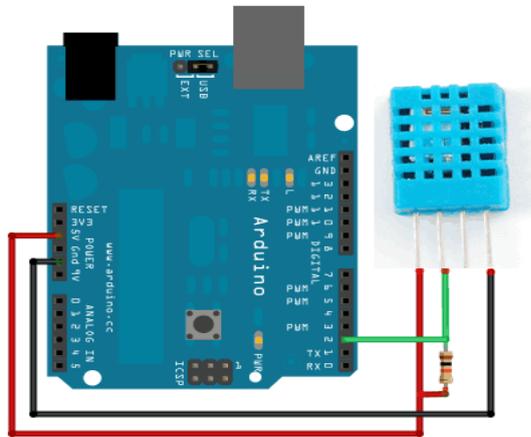
Πρώτα θα κουμπώσουμε τον αισθητήρα DHT11 στον breadboard, έπειτα θα ακολουθήσουμε την συνδεσμολογία που απαιτείται με το arduino σύμφωνα με το ηλεκτρονικό κύκλωμα που έχουμε σχεδιάσει στο πρόγραμμα eagle.

- Ηλεκτρονικό Σχέδιο του DHT11.



Σχήμα 5. Το ηλεκτρονικό σχέδιο DHT11 – Mega2560.

- Σχέδιο συνδεσμολογίας του DHT11.



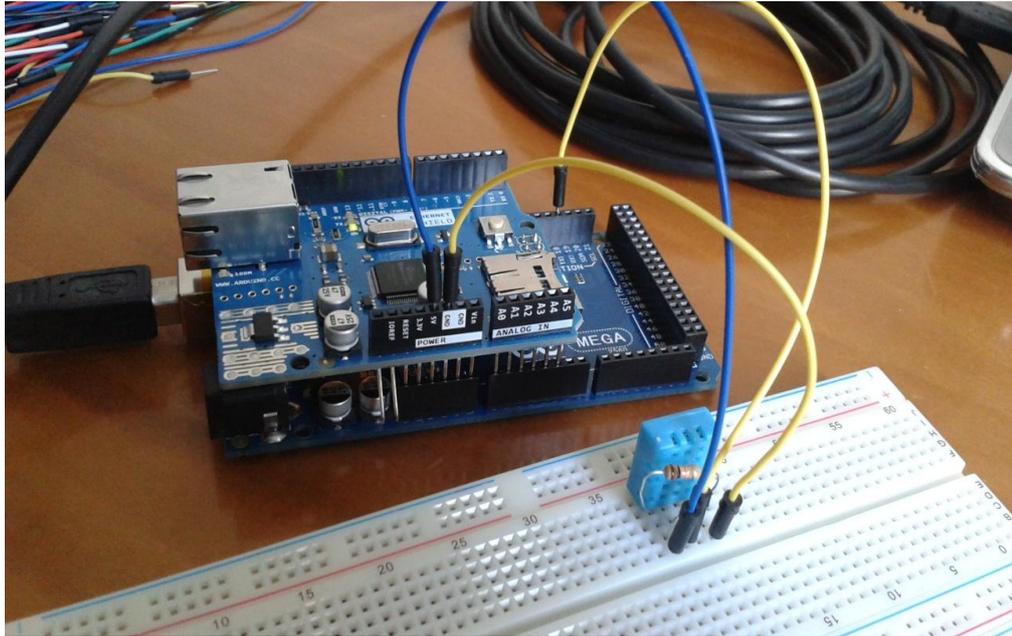
Εικόνα 69. Η συνδεσμολογία DHT11-Mega2560.

Συνδέουμε το 1<sup>ο</sup> pin του αισθητήρα από αριστερά, από την μεριά που βλέπουμε τις εγχοπές του, που είναι το VCC πρώτα με την αντίσταση των 10KΩhm και μετά για τροφοδοσία με το pin 5V του arduino Ethernet shield. Το 2<sup>ο</sup> pin, που είναι για την αποστολή των δεδομένων, συνδέεται με την αντίσταση των 10 KΩhm στον breadboard και έπειτα στο pin D17- RX 2 (receiver- δέκτης) του Mega. Τέλος συνδέουμε το 4<sup>ο</sup> pin του DHT11 , που είναι η γείωση, με το GND του arduino Ethernet shield.

DHT11	Αντίσταση KΩhm	10	Arduino Shield	Ethernet	Arduino Mega2560
PIN 1 (VCC)	Πρώτα συνδέεται με την αντίσταση και μετά με τον arduino (pull-up)		5V		-
PIN 2 (DATA)	Πρώτα συνδέεται με την αντίσταση και μετά με τον arduino (pull-up)		-		D17 (RX2)
PIN 3 (NORMAL CLOSED)	-		-		-
PIN 4 (GND)	-		GND		-

Πίνακας 11. Συνδεσμολογία των ακροδεκτών του DHT11 με την πλακέτα.

- Φωτογραφία από την υλοποίηση του DHT11.



Εικόνα 70. Κατά την κατασκευή DHT11.

- Ο κώδικας του DHT11.

```
#include <dht11.h>
dht11 DHT;
#define DHT11_PIN 17

void setup(){
  Serial.begin(9600);
  Serial.println("DHT TEST PROGRAM ");
  Serial.print("LIBRARY VERSION: ");
  Serial.println(DHT11LIB_VERSION);
  Serial.println();
  Serial.println("Type,\tstatus,\tHumidity (%),\tTemperature (C)");
}

void loop(){
  int chk;
  Serial.print("DHT11, \t");
  chk = DHT.read(DHT11_PIN); // READ DATA
  switch (chk){
    case DHTLIB_OK:
      Serial.print("OK,\t");
      break;
    case DHTLIB_ERROR_CHECKSUM:
      Serial.print("Checksum error,\t");
      break;
    case DHTLIB_ERROR_TIMEOUT:
      Serial.print("Time out error,\t");
      break;
    default:
```

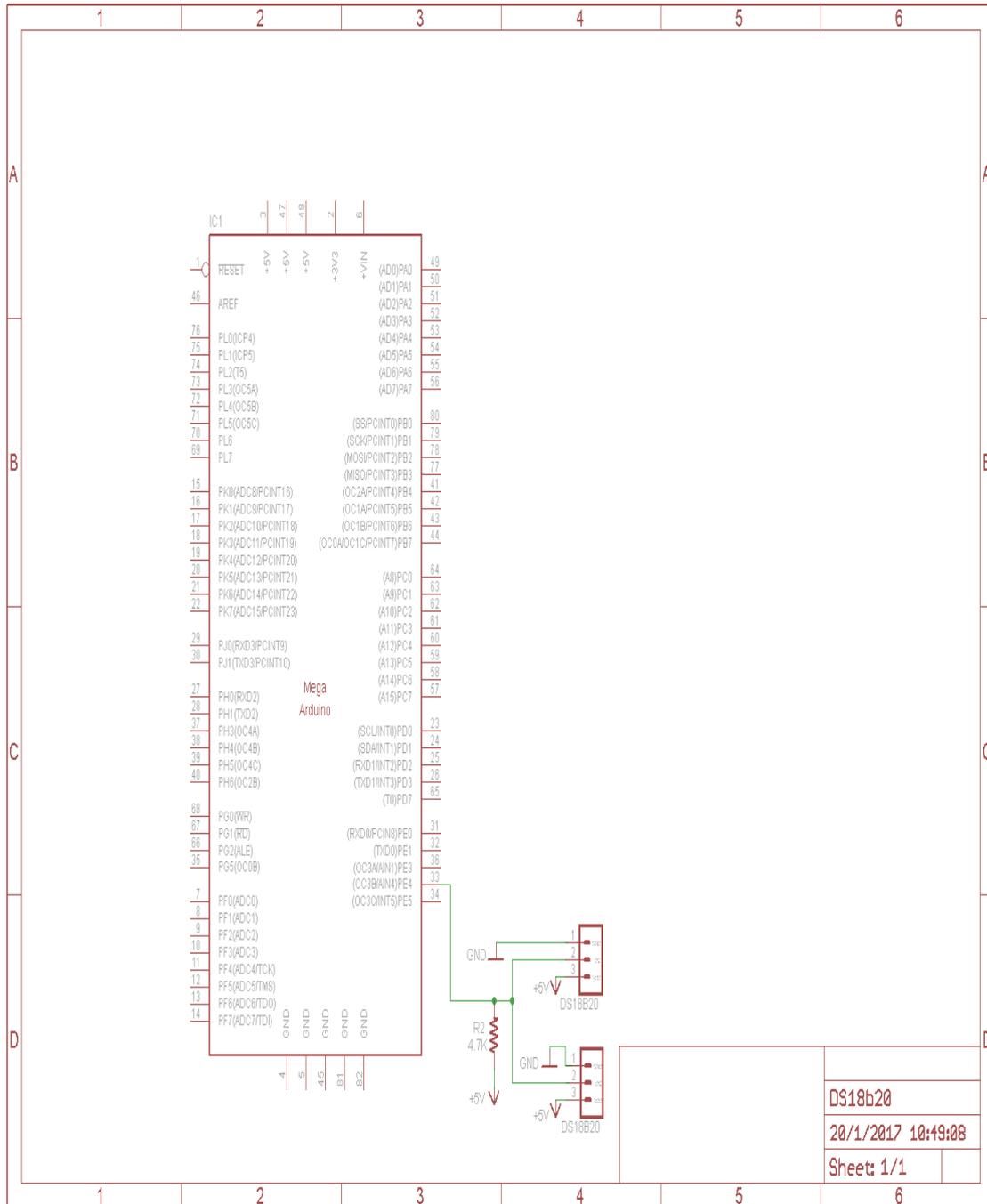
```
        Serial.print("Unknown error,\t");
        break;
    }
    // DISPLAT DATA
    Serial.print(DHT.humidity,1);
    Serial.print(",\t");
    Serial.println(DHT.temperature,1);

    delay(1000);
}
```

#### 12.4 Συνδεσμολογία της πλακέτας με τον αισθητήρα DS18B20

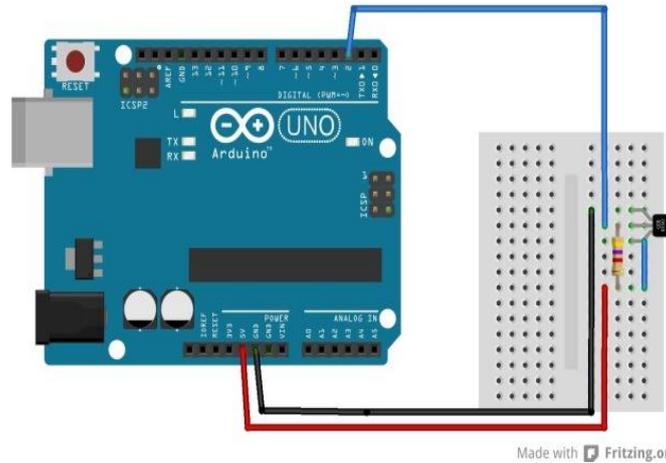
Επόμενος αισθητήρας που χρησιμοποιείται είναι ο DS18B20. Τοποθετείται πάνω στον breadboard. Η συνδεσμολογία που ακολουθείται φαίνεται παρακάτω, σύμφωνα με το ηλεκτρονικό σχέδιό του.

- Ηλεκτρονικό Σχέδιο του DS18B20.

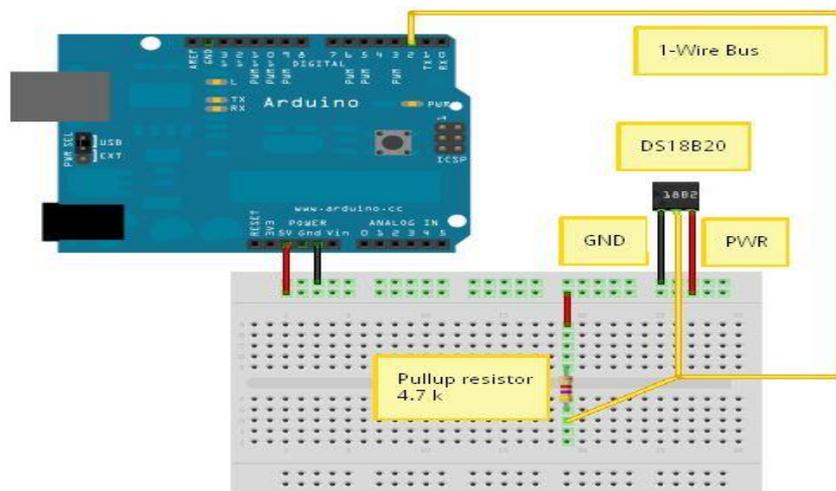


Σχήμα 6. Το ηλεκτρικό σχέδιο των DS18B20 και DS18B20-Waterproof με τον Mega2560.

- Σχέδιο συνδεσμολογίας του DS18B20.



Εικόνα 71. Η συνδεσμολογία DS18B20-Mega2560.



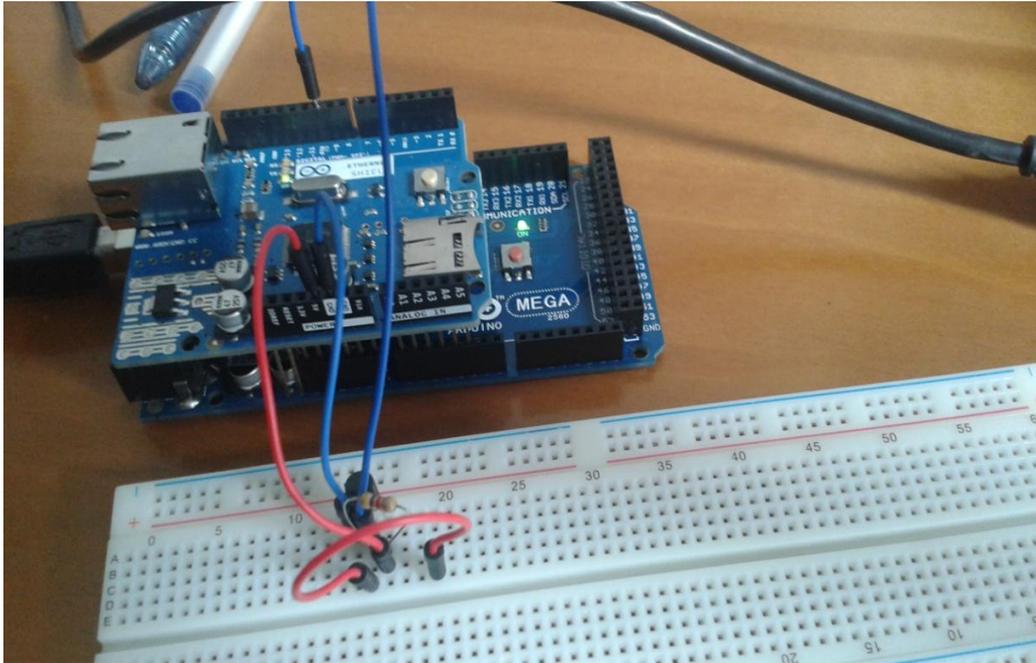
Εικόνα 72. Η συνδεσμολογία DS18B20-Mega2560.

Τοποθετούμε τον αισθητήρα μπροστά μας, ώστε να βλέπουμε την πλακέ μεριά του και όχι την κοίλη καμπύλη του. Το 1<sup>ο</sup> pin από αριστερά είναι το GND (γείωση) και συνδέεται με καλώδιο με το GND του arduino. Συνδέεται μία αντίσταση των 4.7 KΩhm μεταξύ 2<sup>ου</sup> και 3<sup>ου</sup> pin του αισθητήρα. Το 2<sup>ο</sup> pin του αισθητήρα συνδέεται με το Digital 10 pin του arduino. Τέλος συνδέεται το 3<sup>ο</sup> pin του DS18B20 με τα 5V της τροφοδοσίας του arduino.

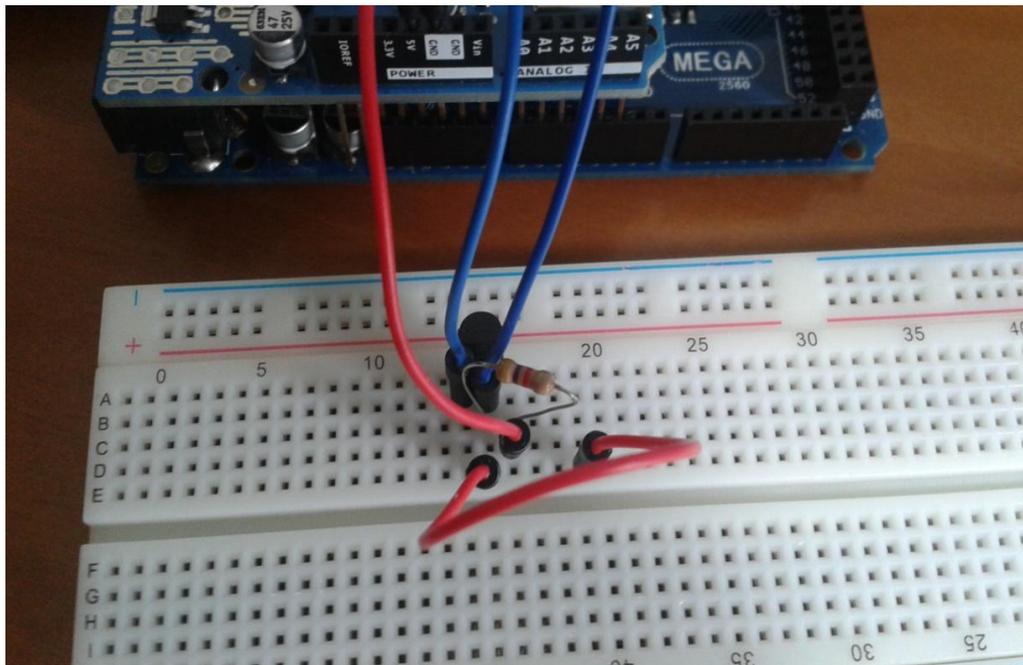
DS18B20	Αντίσταση 4.7 KΩhm	Ethernet Shield
1 (GND)	-	GND
2 (DATA)	Συνδέεται με το pin 3 και έπειτα με την πλακέτα	Digital pin10
3 (VCC)	Συνδέεται με το pin 2 και έπειτα με την πλακέτα	5V

Πίνακας 12. . Συνδεσμολογία των ακροδεκτών του DS18B20 με την πλακέτα.

- Φωτογραφίες από την υλοποίηση του DS18B20.



Εικόνα 73. Η κατασκευή του DS18B20.



Εικόνα 74. Η κατασκευή του DS18B20.

- Ο κώδικας του DS18B20.

```
#include <OneWire.h>
```

```
// OneWire DS18S20, DS18B20, DS1822 Temperature Example
```

```
OneWire ds(10); // on pin 10 (a 4.7K resistor is necessary)
```

```
void setup(void) {
  Serial.begin(9600);
}

void loop(void) {
  byte i;
  byte present = 0;
  byte type_s;
  byte data[12];
  byte addr[8];
  float celsius, fahrenheit;

  if ( !ds.search(addr) ) {
    Serial.println("No more addresses.");
    Serial.println();
    ds.reset_search();
    delay(250);
    return;
  }

  Serial.print("ROM =");
  for( i = 0; i < 8; i++) {
    Serial.write(' ');
    Serial.print(addr[i], HEX);
  }

  if (OneWire::crc8(addr, 7) != addr[7]) {
    Serial.println("CRC is not valid!");
    return;
  }
  Serial.println();

  // the first ROM byte indicates which chip
  switch (addr[0]) {
    case 0x10:
      Serial.println("  Chip = DS18S20"); // or old DS1820
      type_s = 1;
      break;
    case 0x28:
      Serial.println("  Chip = DS18B20");
      type_s = 0;
      break;
    case 0x22:
      Serial.println("  Chip = DS1822");
      type_s = 0;
      break;
    default:
      Serial.println("Device is not a DS18x20 family device.");
      return;
  }

  ds.reset();
  ds.select(addr);
```

```
ds.write(0x44, 1);    // start conversion, with parasite power on at the end

delay(1000);    // maybe 750ms is enough, maybe not
// we might do a ds.depower() here, but the reset will take care of it.

present = ds.reset();
ds.select(addr);
ds.write(0xBE);    // Read Scratchpad

Serial.print(" Data = ");
Serial.print(present, HEX);
Serial.print(" ");
for ( i = 0; i < 9; i++) {    // we need 9 bytes
  data[i] = ds.read();
  Serial.print(data[i], HEX);
  Serial.print(" ");
}
Serial.print(" CRC=");
Serial.print(OneWire::crc8(data, 8), HEX);
Serial.println();

// Convert the data to actual temperature
// because the result is a 16 bit signed integer, it should
// be stored to an "int16_t" type, which is always 16 bits
// even when compiled on a 32 bit processor.
int16_t raw = (data[1] << 8) | data[0];
if (type_s) {
  raw = raw << 3; // 9 bit resolution default
  if (data[7] == 0x10) {
    // "count remain" gives full 12 bit resolution
    raw = (raw & 0xFFF0) + 12 - data[6];
  }
} else {
  byte cfg = (data[4] & 0x60);
  // at lower res, the low bits are undefined, so let's zero them
  if (cfg == 0x00) raw = raw & ~7; // 9 bit resolution, 93.75 ms
  else if (cfg == 0x20) raw = raw & ~3; // 10 bit res, 187.5 ms
  else if (cfg == 0x40) raw = raw & ~1; // 11 bit res, 375 ms
  //// default is 12 bit resolution, 750 ms conversion time
}
celsius = (float)raw / 16.0;
fahrenheit = celsius * 1.8 + 32.0;
Serial.print(" Temperature = ");
Serial.print(celsius);
Serial.print(" Celsius, ");
Serial.print(fahrenheit);
Serial.println(" Fahrenheit");
}
```

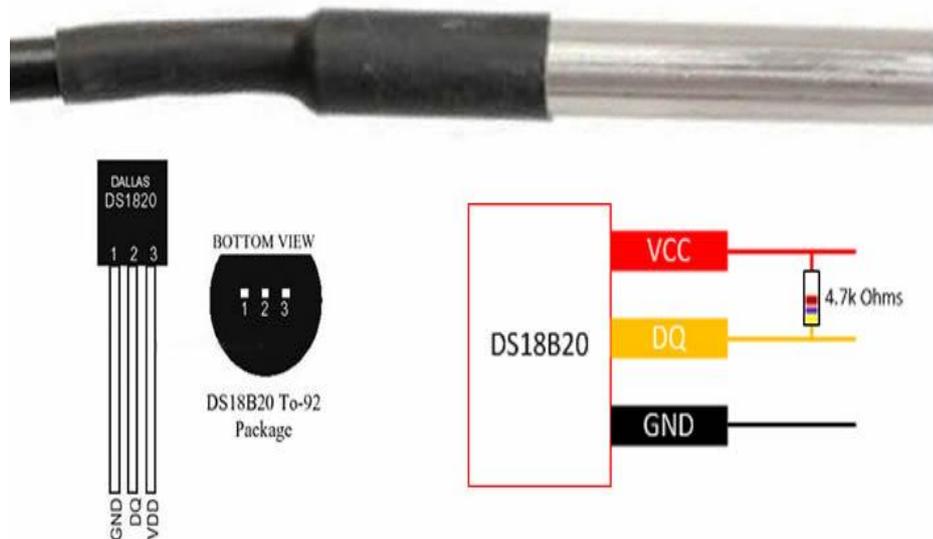
## 12.5 Συνδεσμολογία της πλακέτας με τον αισθητήρα DS18B20-Waterproof.

Τρίτος αισθητήρας που χρησιμοποιείται είναι ο DS18B20-Waterproof.

Αυτός ο αισθητήρας όπως και ο DS18B20 χρησιμοποιούν το ίδιο πρωτόκολλο επικοινωνίας One Wire, το οποίο τους δίνει την δυνατότητα να χρησιμοποιούν δύο ή περισσότεροι αισθητήρες πάνω στον ίδιο δίαυλο.

Για αυτόν τον λόγο και θα συνδέσουμε τον αδιάβροχο DS18B20 παράλληλα με τον παραπάνω αισθητήρα.

- Ηλεκτρονικό Σχέδιο του DS18B20-Waterproof.  
Το σχέδιο αυτού του αισθητήρα είναι στο αντίστοιχο σχέδιο του DS18B20.
- Σχέδιο συνδεσμολογίας του DS18B20-Waterproof.



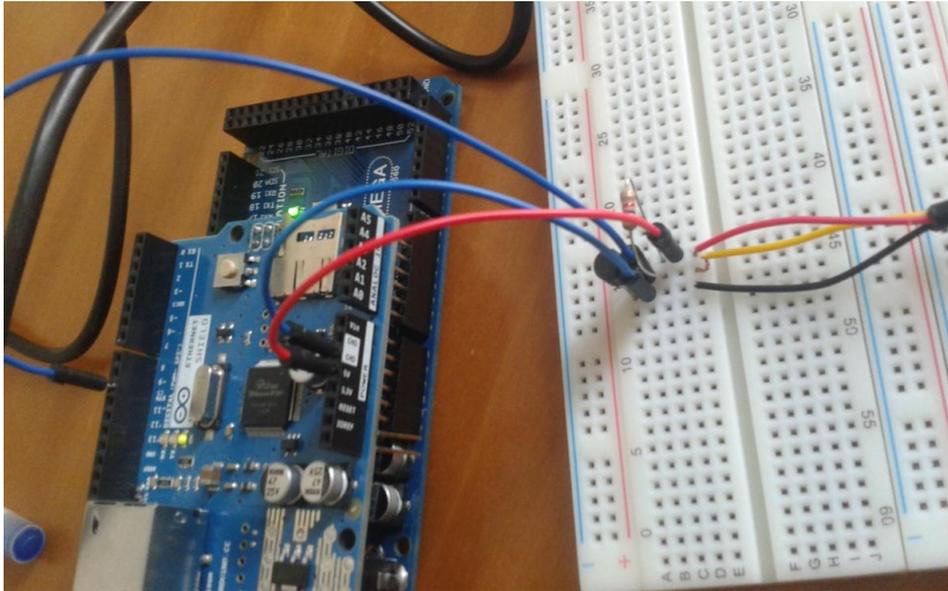
Εικόνα 75. Οι ακροδέκτες των DS18B20 DS18B20-Waterproof.

Συνδέεται στο ήδη υπάρχον κύκλωμα του DS18B20 και του ARDUINO. Άρα, αφού πρώτα κόψουμε το πλαστικό περίβλημα των συρμάτινων δεσμών με ένα κοπτηράκι, τις απογυμνωμένες δέσμες τις συνδέουμε ως εξής:

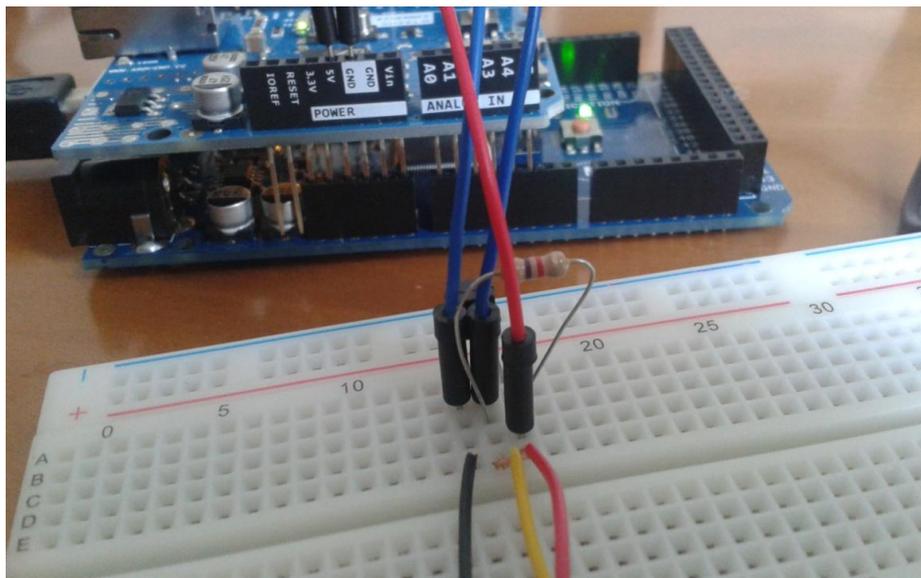
DS18B20-Waterproof	DS18B20
Μαύρο καλώδιο (GND)	1 (GND)
Κίτρινο (DATA)	2 (DATA)
Κόκκινο (VCC)	3 (VCC)

Πίνακας 13. Συνδεσμολογία των ακροδεκτών του DS18B20-Waterproof με την πλακέτα μέσω των ακροδεκτών του DS18B20.

- Φωτογραφίες από την υλοποίηση του DS18B20-Waterproof.



Εικόνα 76. Κατά την κατασκευή του DS18B20-Waterproof.



Εικόνα 77. Κατά την κατασκευή του DS18B20-Waterproof.

- Ο κώδικας του DS18B20-Waterproof.

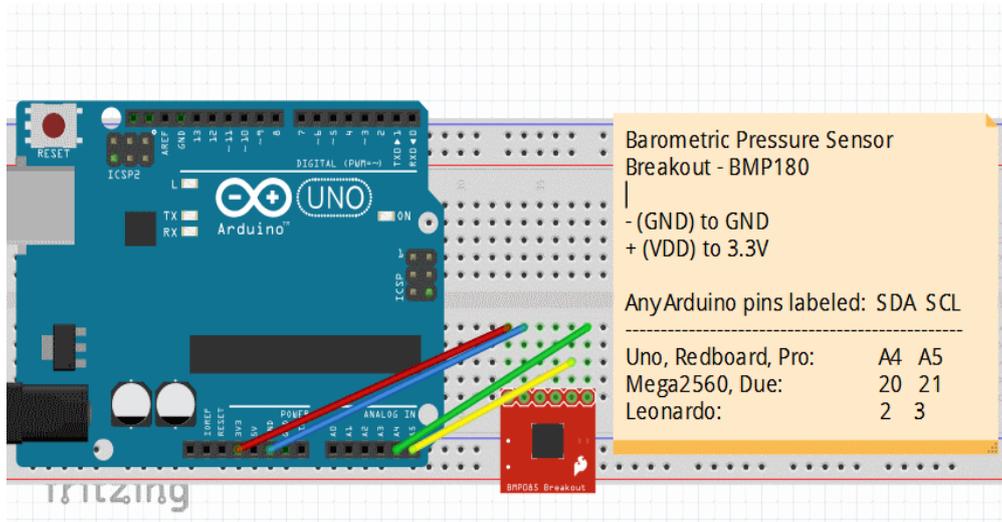
Είναι ο ίδιος κώδικας που τρέχουμε και για τον DS18B20.

## 12.6 Συνδεσμολογία της πλακέτας με τον αισθητήρα BMP180

Τέταρτος αισθητήρας που χρησιμοποιείται είναι ο BMP180. Αυτός ο αισθητήρας χρησιμοποιεί για τη μεταφορά των δεδομένων (0 ή 1) μόνο δύο καλώδια (τα οποία είναι αμφίδρομης κατεύθυνσης): Τα SCL και SDA. Η γραμμή SCL είναι η γραμμή ρολογιού, ενώ η SDA είναι η γραμμή δεδομένων. Οι γραμμές αυτές συνδέονται σε όλες τις συσκευές, που



- Σχέδιο συνδεσμολογίας του BMP180.



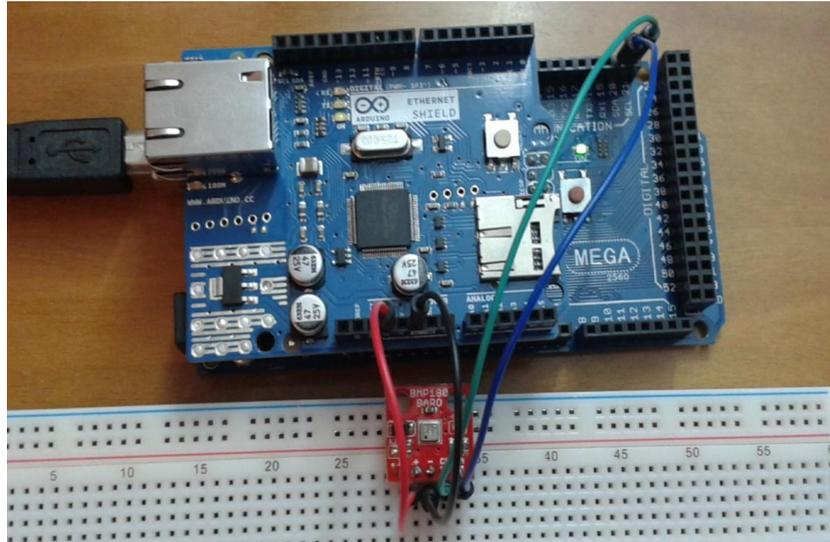
Εικόνα 78. Η συνδεσμολογία BMP180-Mega2560.

Τοποθετούμε τον αισθητήρα πάνω στον breadboard και η πλευρά των pins του να είναι παράλληλη ως προς τις πλακέτες για να μπορέσει να πιο εύκολα η συνδεσμολογία των καλωδίων της μεταξύ τους σύνδεσης. Κοιτάμε τον BMP180 έτσι ώστε να διαβάζουμε τα pins του. Από αριστερά προς τα δεξιά διαβάζουμε και συνδέουμε. Το pin '+' του αισθητήρα, που είναι η VDD, τροφοδοσία καλωδιώνεται με τα 3.3V της πλακέτας του Arduino. Το pin '-' του BMP180 συνδέεται με την γείωση GND της πλακέτας. Το pin SDA συνδέεται με την ακίδα 20 του Arduino Mega και το SCL με την ακίδα 21 του Mega.

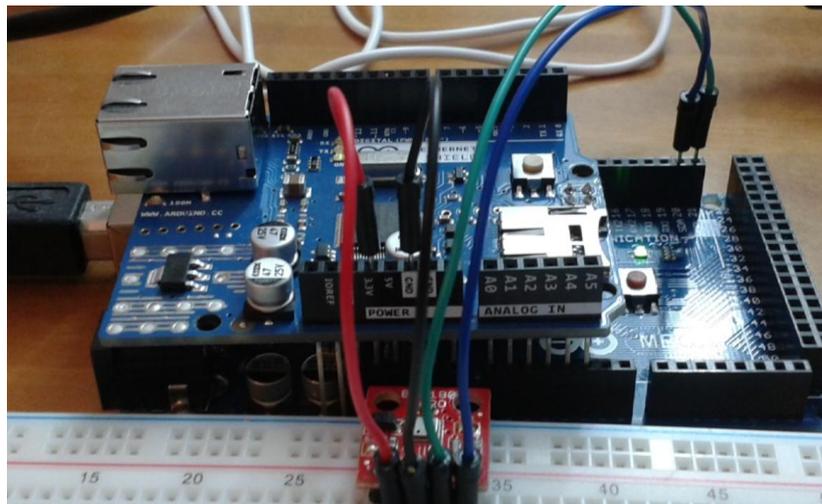
BMP180	ARDUINO MEGA	ETHERNET SHIELD
1 (+)	-	3.3V
2 (-)	-	GND
3 (SCL)	21 PIN (SCL)	-
4 (SDA)	20 PIN (SDA)	-

Πίνακας 14. Συνδεσμολογία των ακροδεκτών του BMP180 με τις πλακέτες.

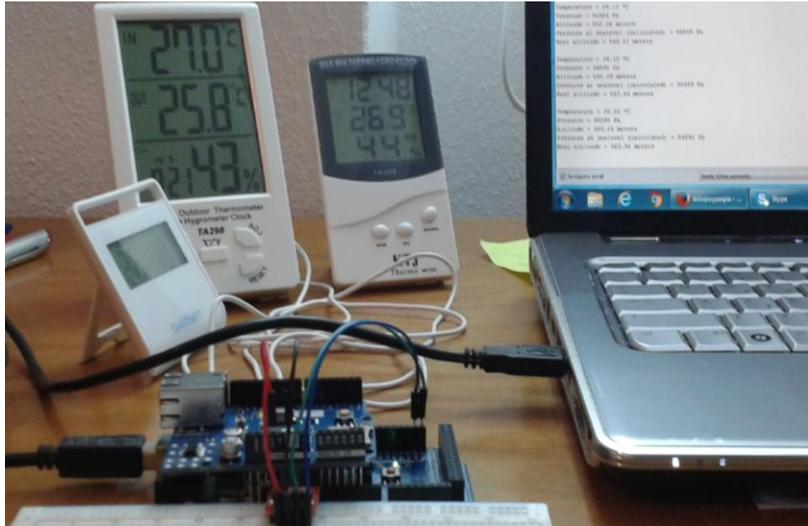
- Φωτογραφίες από την υλοποίηση του BMP180.



Εικόνα 79. Κατά την κατασκευή του BMP180.



Εικόνα 80. Κατά την κατασκευή του BMP180.



Εικόνα 81. Κατά την κατασκευή του BMP180.

- Ο κώδικας του BMP180.

```
#include <Wire.h>
#include <Adafruit_BMP085.h>

Adafruit_BMP085 bmp;

void setup() {
  Serial.begin(9600);
  if (!bmp.begin()) {
    Serial.println("Could not find a valid BMP085 sensor, check wiring!");
    while (1) {}
  }
}

void loop() {
  Serial.print("Temperature = ");
  Serial.print(bmp.readTemperature());
  Serial.println(" *C");

  Serial.print("Pressure = ");
  Serial.print(bmp.readPressure());
  Serial.println(" Pa");

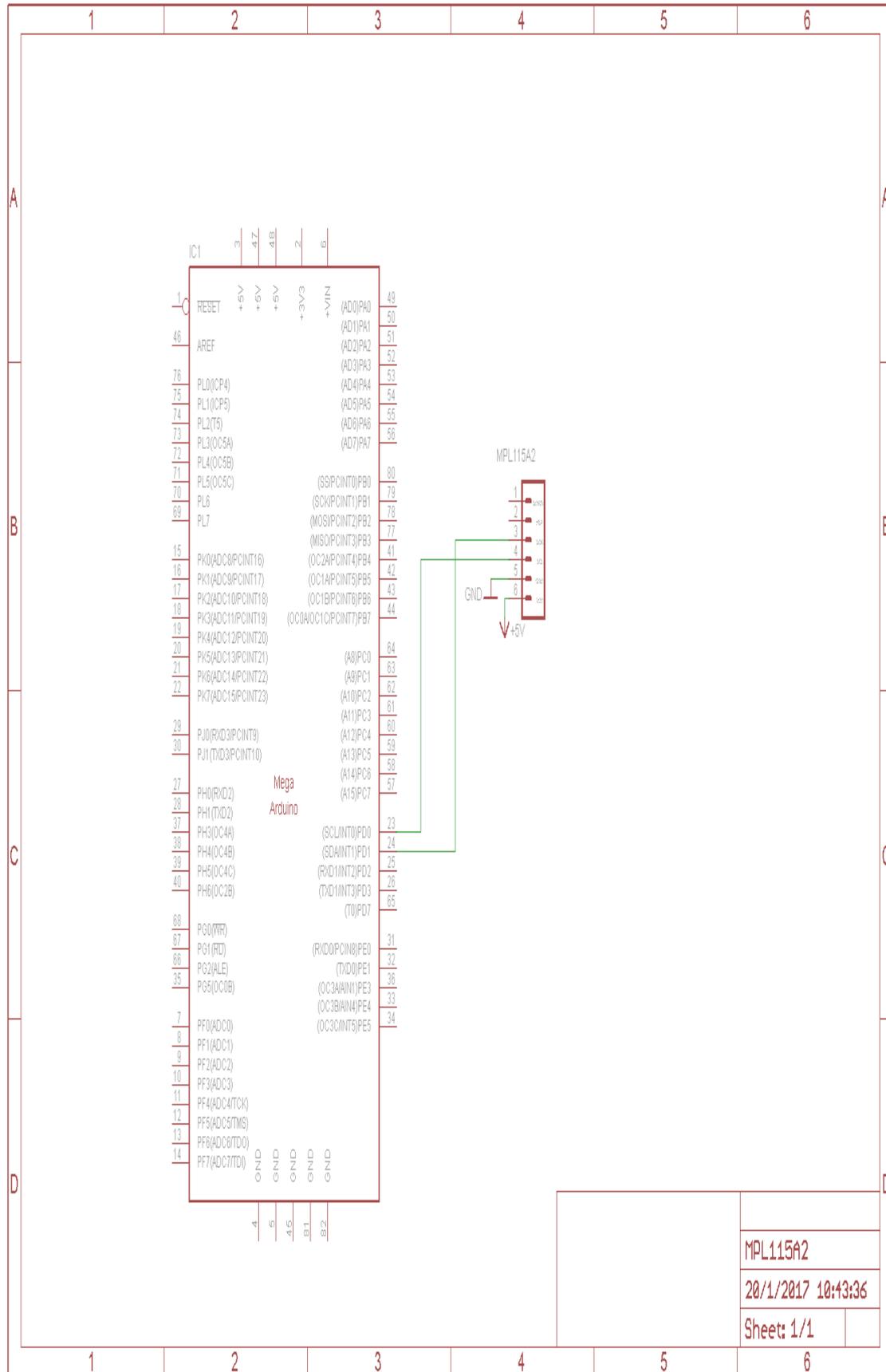
  // Calculate altitude assuming 'standard' barometric
  // pressure of 1013.25 millibar = 101325 Pascal
  Serial.print("Altitude = ");
  Serial.print(bmp.readAltitude());
  Serial.println(" meters");

  Serial.print("Pressure at sealevel (calculated) = ");
  Serial.print(bmp.readSealevelPressure());
  Serial.println(" Pa");

  // you can get a more precise measurement of altitude
  // if you know the current sea level pressure which will
```

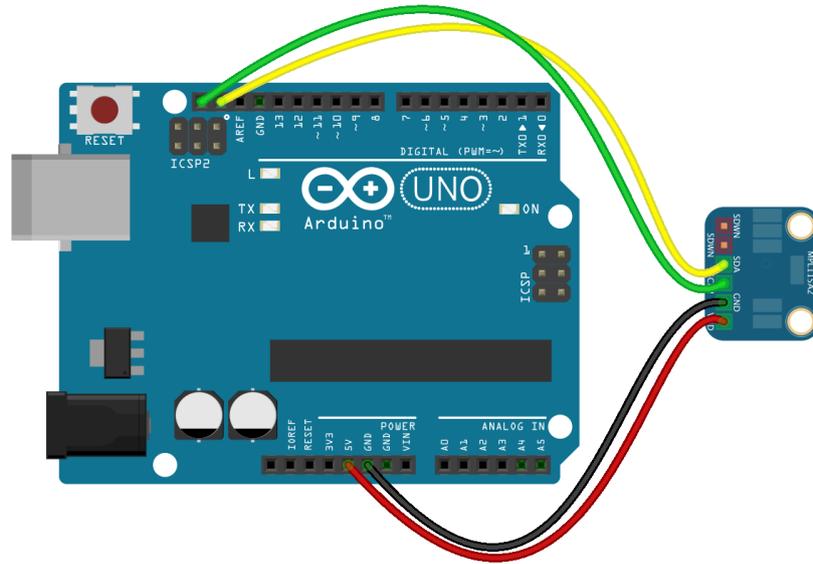


- Ηλεκτρονικό Σχέδιο του MPL115A2 .



Σχήμα 8. Το ηλεκτρονικό σχέδιο των MPL115A2-Mega2560.

- Σχέδιο συνδεσμολογίας του MPL115A2 .



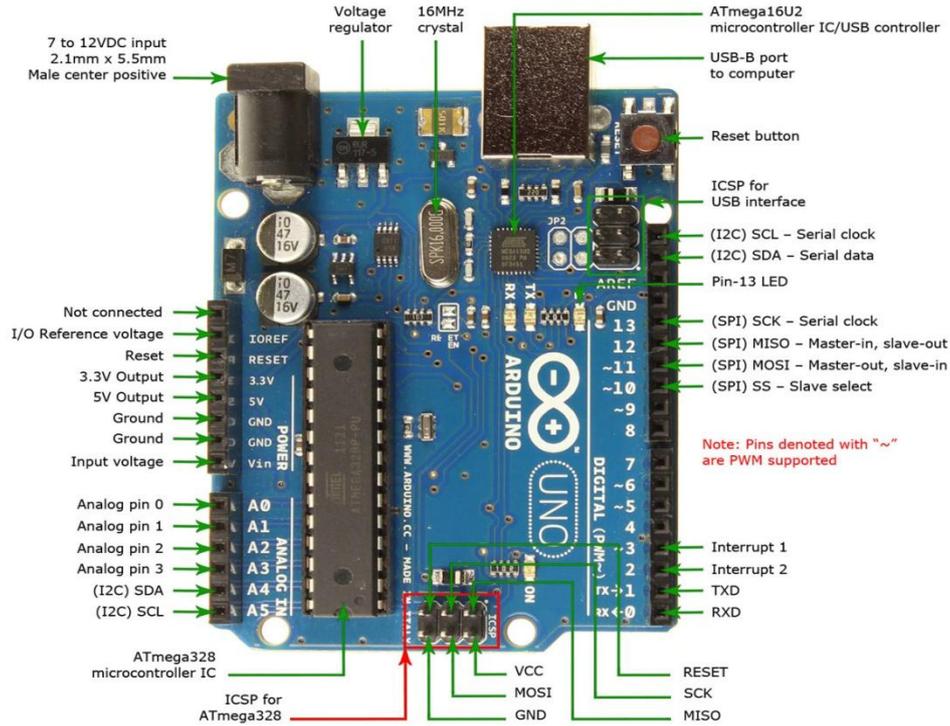
Εικόνα 83. Η συνδεσμολογία των MPL115A2-Mega2560.

Τοποθετούμε τον αισθητήρα στον breadboard έτσι ώστε να διαβάζουμε τα στοιχεία και το breadboard παράλληλα με την πλακέτα για ευκολία στην μεταξύ τους συνδεσμολογία. Από αριστερά προς τα δεξιά ξεκινάμε. Το pin SDA συνδέεται με την ψηφιακή ακίδα 20 του Mega. Το pin SCL συνδέεται με την ψηφιακή ακίδα 21 του Mega. Το GND με το GND και το VDD με τα 5V της Ethernet shield.

MPL115A2	Arduino Mega	Ethernet Shield
1 (SDA)	Digital 20 pin (SDA)	-
2 (SCL)	Digital 21 pin (SCL)	-
3 (GND)	-	GND
4 (VDD)	-	5V

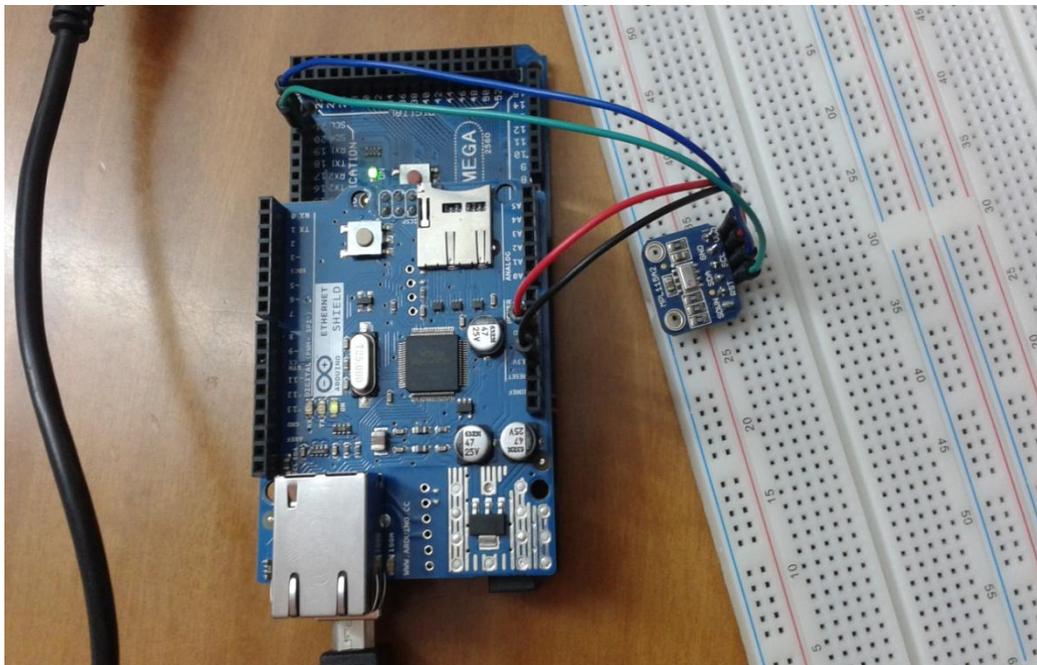
Πίνακας 15. Συνδεσμολογία των ακροδεκτών του MPL115A2 με την πλακέτα.

Παρατήρηση, επειδή έχει γίνει χρήση του Arduino UNO για να παρουσιαστεί η συνδεσμολογία παραθέτεται παρακάτω ο UNO.



Εικόνα 84. Arduino UNO.

- Φωτογραφίες από την υλοποίηση του MPL115A2 .



Εικόνα 85. Κατά την κατασκευή του MPL115A2.

- Ο κώδικας του MPL115A2.

```
#include <Wire.h>
#include <Adafruit_MPL115A2.h>
```

```
Adafruit_MPL115A2 mpl115a2;

void setup(void)
{
  Serial.begin(9600);
  Serial.println("Hello!");

  Serial.println("Getting barometric pressure ...");
  mpl115a2.begin();
}

void loop(void)
{
  float pressureKPA = 0, temperatureC = 0;

  mpl115a2.getPT(&pressureKPA,&temperatureC);
  Serial.print("Pressure (kPa): "); Serial.print(pressureKPA, 4); Serial.print(" kPa ");
  Serial.print("Temp (*C): "); Serial.print(temperatureC, 1); Serial.println(" *C both measured
together");

  pressureKPA = mpl115a2.getPressure();
  Serial.print("Pressure (kPa): "); Serial.print(pressureKPA, 4); Serial.println(" kPa");

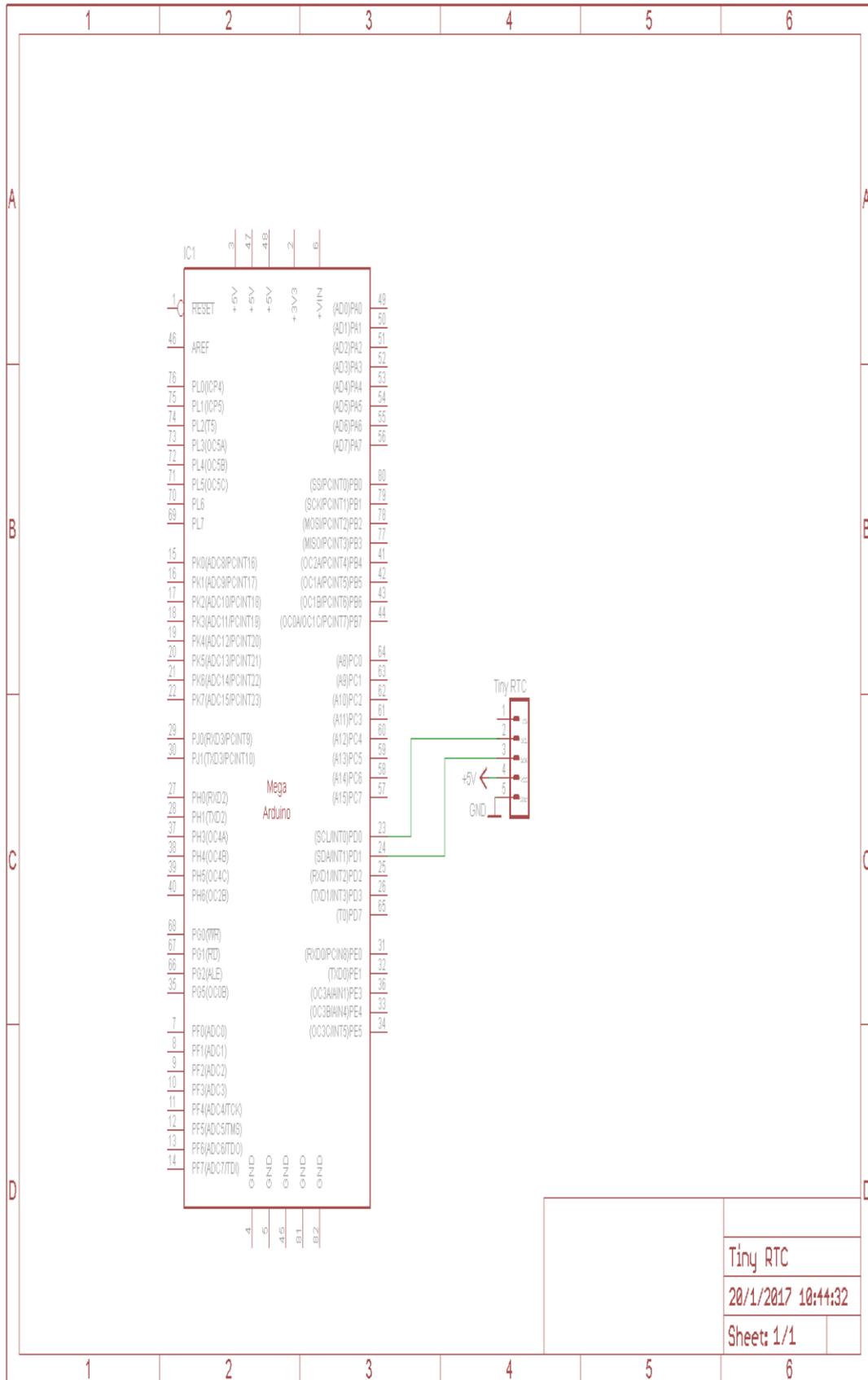
  temperatureC = mpl115a2.getTemperature();
  Serial.print("Temp (*C): "); Serial.print(temperatureC, 1); Serial.println(" *C");

  delay(1000);
}
```

## 12.8 Συνδεσμολογία της πλακέτας με το Tiny RTC\_DS1307

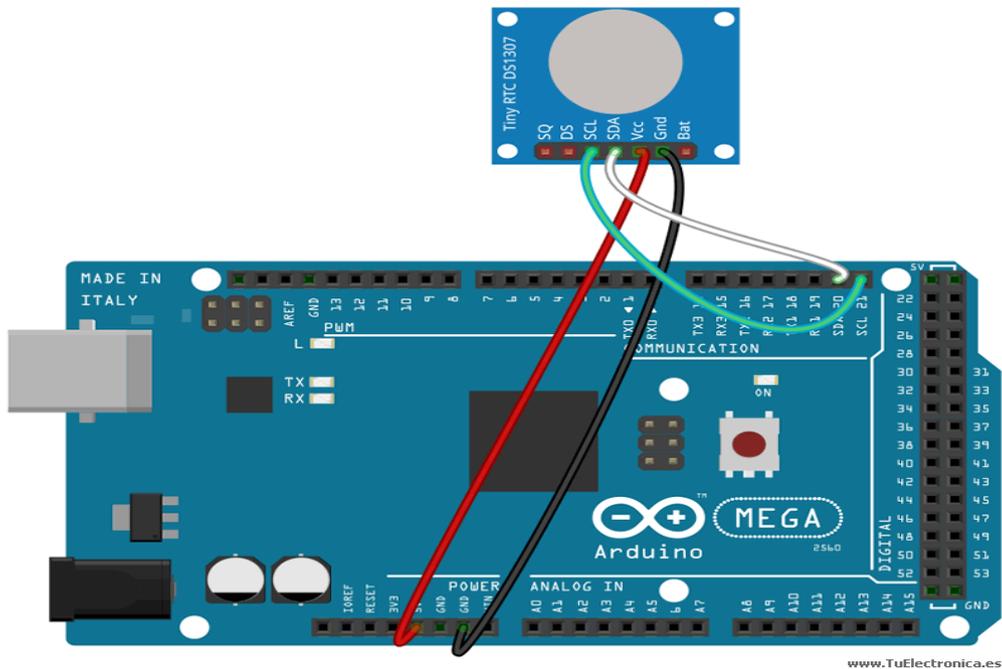
Ο Tiny RTC\_DS1307 χρησιμοποιείται για να εμφανίζεται η τρέχον ημερομηνία και ώρα που γίνονται οι λήψεις των μετρήσεων από τους παραπάνω αισθητήρες.

- Ηλεκτρονικό Σχέδιο του Tiny RTC\_DS1307.



Σχήμα 9. Το ηλεκτρονικό σχέδιο των Tiny RTC\_DS1307-Mega2560.

- Σχέδιο συνδεσμολογίας του Tiny RTC\_DS1307.



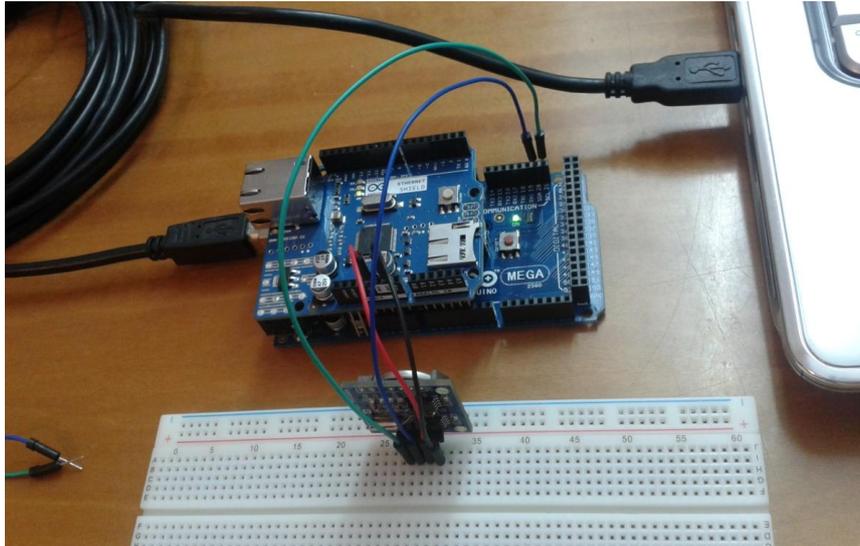
Εικόνα 86. Η συνδεσμολογία των Tiny RTC\_DS1307-Mega2560.

Αρχικά, τοποθετούμε τον DS1307 στον breadboard με την μπαταρία να είναι παράλληλα με την πλακέτα. Ξεκινάμε την καλωδίωση από την άλλη πλευρά του εξωτερικού ρολογιού από αριστερά προς τα δεξιά. Το SCL συνδέεται με την ψηφιακή ακίδα 21 του Mega και το SDA με την 20 του Mega. Ο ακροδέκτης VCC με την τροφοδοσία 5V και το GND με την γείωση του Ethernet.

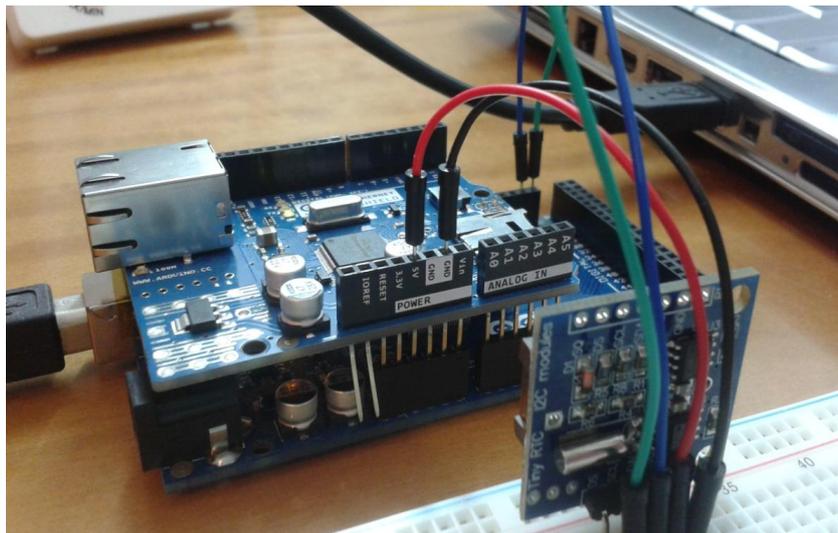
RTC_DS1307	Arduino Mega	Ethernet Shield
1 (SCL)	21 (SCL)	-
2 (SDA)	20 (SDA)	-
3 (VCC)	-	5V
4 (GND)	-	GND

Πίνακας 16. Συνδεσμολογία των ακροδεκτών του RTC\_DS1307 με τις πλακέτες.

- Φωτογραφίες από την υλοποίηση του Tiny RTC\_DS1307.



Εικόνα 87. Κατά την κατασκευή των Tiny RTC\_DS1307-Mega2560.



Εικόνα 88. Κατά την κατασκευή των Tiny RTC\_DS1307-Mega2560.

- Ο κώδικας του Tiny RTC\_DS1307.

```
#include <Wire.h>  
#include "RTClib.h"
```

```
RTC_DS1307 RTC;
```

```
void setup () {  
  Serial.begin(9600);  
  Wire.begin();  
  RTC.begin();
```

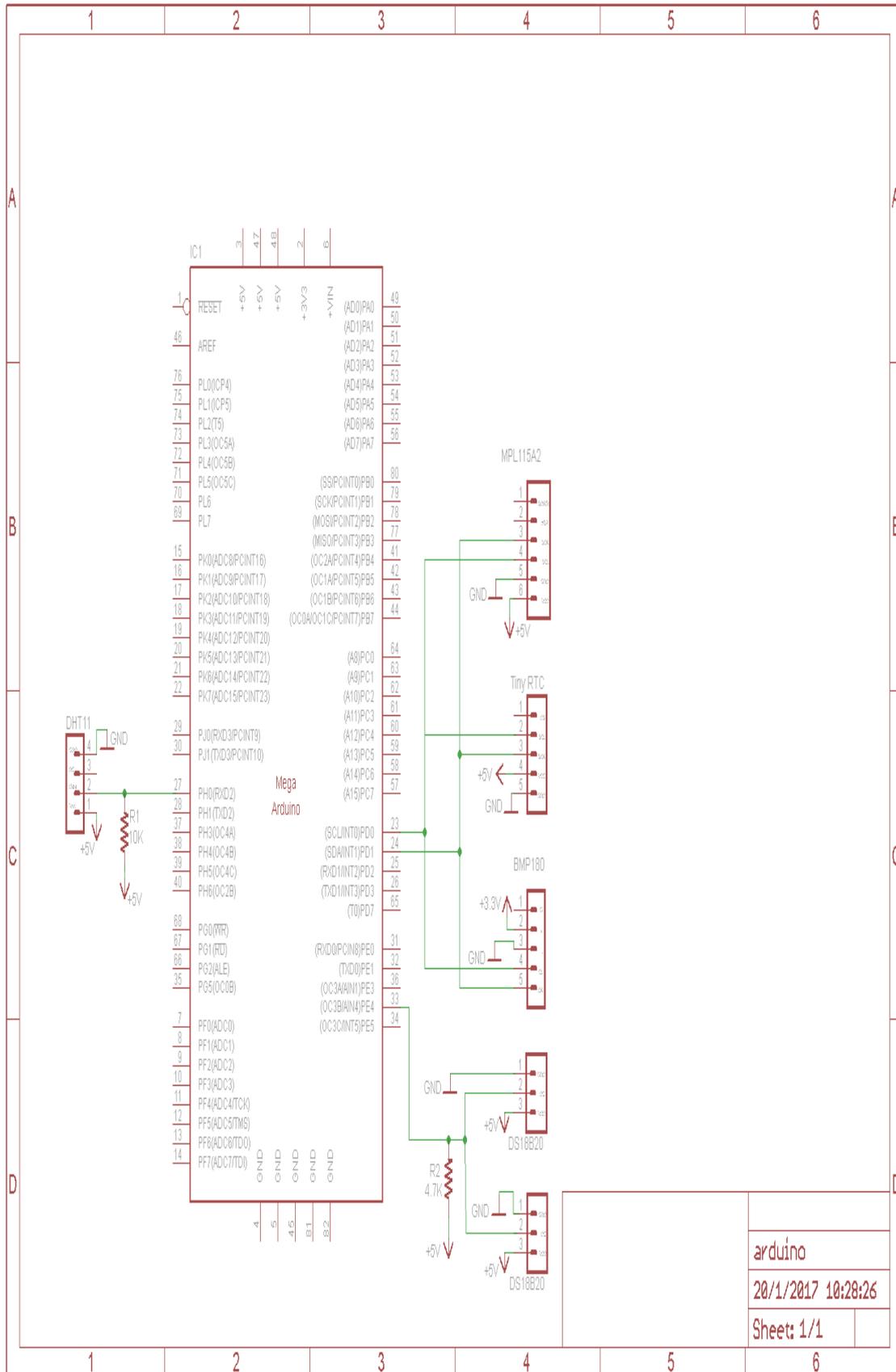
```
  // following line sets the RTC to the date & time this sketch was compiled
```

```
RTC.adjust(DateTime(__DATE__, __TIME__));  
}  
  
void loop () {  
  DateTime now = RTC.now();  
  Serial.print(now.year(), DEC);  
  Serial.print('/');  
  Serial.print(now.month(), DEC);  
  Serial.print('/');  
  Serial.print(now.day(), DEC);  
  Serial.print(' ');  
  Serial.print(now.hour(), DEC);  
  Serial.print(':');  
  Serial.print(now.minute(), DEC);  
  Serial.print(':');  
  Serial.print(now.second(), DEC);  
  Serial.println();  
  delay(3000);  
}
```

## 12.9 Η τελική συνδεσμολογία της εργασίας

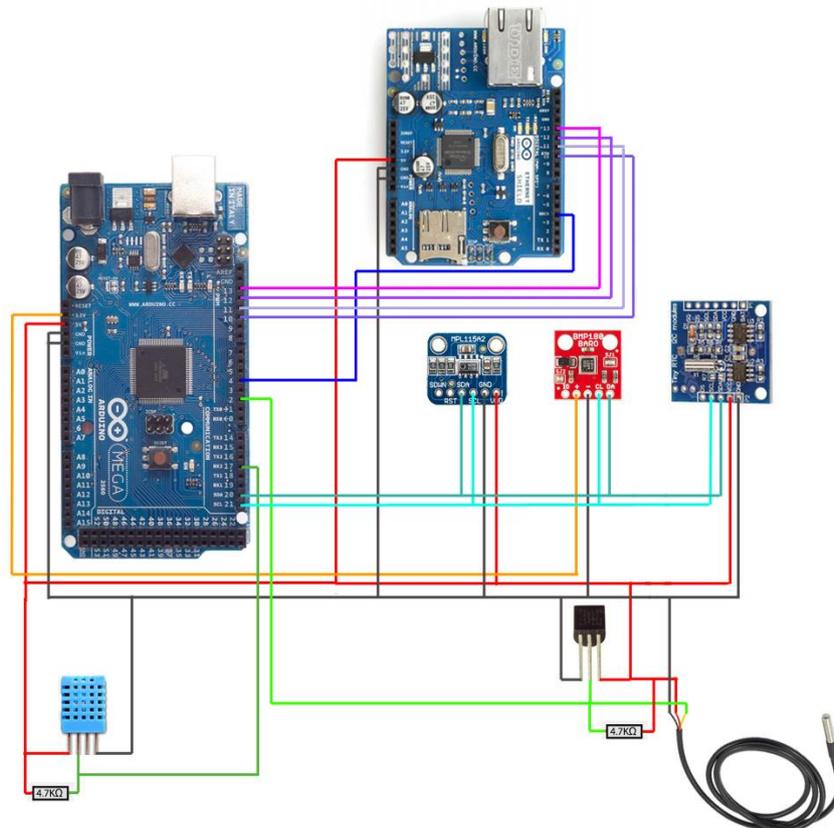
Αφού έχει γίνει η πρώτη μελέτη των αισθητήρων και έχει γίνει κατανοητό πως λειτουργούν, επόμενο βήμα της εργασίας είναι να τοποθετηθούν όλοι μαζί και να προγραμματιστούν έτσι ώστε να φαίνονται οι τιμές όλων τους. Η συνδεσμολόγηση των αισθητήρων γίνεται σταδιακά λόγω του μεγάλου αριθμού τους. Έτσι πρώτα συνδεσμολογούνται τα πιο περίπλοκα κομμάτια της τελικής εργασίας και έπειτα τα υπόλοιπα.

- Ηλεκτρονικό Σχέδιο της τελικής εργασίας.



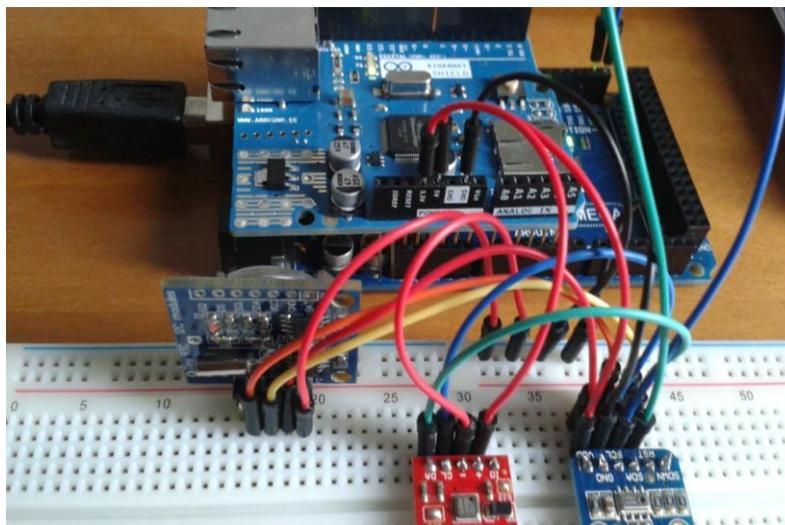
Σχήμα 10. Το ηλεκτρονικό σχέδιο της τελικής εργασίας μου.

- Σχέδιο συνδεσμολογίας της τελικής εργασίας.

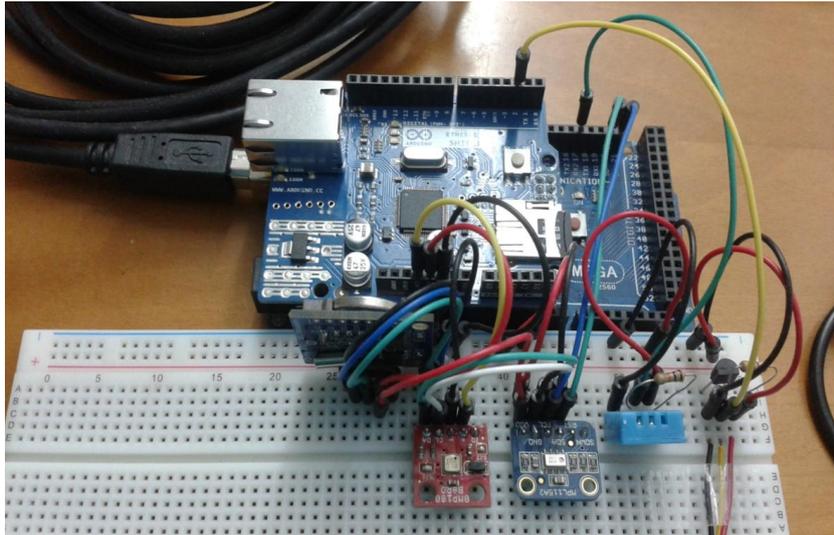


Εικόνα 89. Η τελική συνδεσμολογία της εργασίας μου.

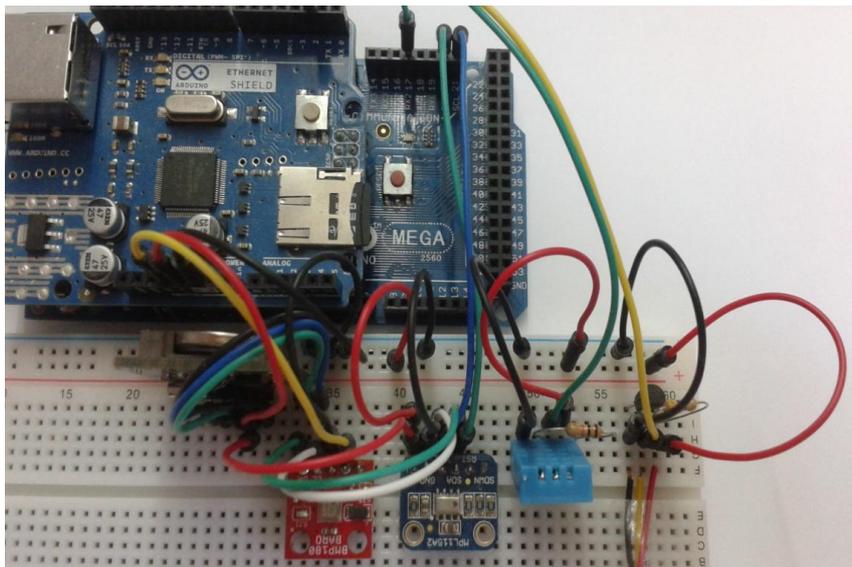
- Φωτογραφίες από την υλοποίηση της τελικής εργασίας.



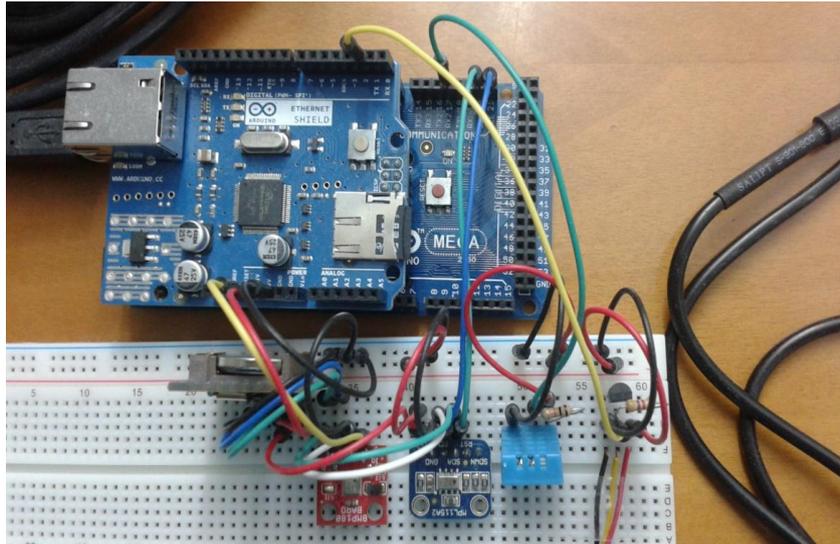
Εικόνα 90. Κατά την κατασκευή της τελικής μου εργασίας.



Εικόνα 91. Κατά την κατασκευή της τελικής μου εργασίας.



Εικόνα 92. Η κατασκευή της τελικής μου εργασίας.



Εικόνα 93. Η κατασκευή της τελικής μου εργασίας.

- Ο κώδικας της τελικής εργασίας.

```
// Ethernet
#include <SPI.h>
#include <Ethernet.h>
#include <SD.h>
// RTC Time Clock
#include <Wire.h>
#include "RTClib.h"
RTC_DS1307 RTC;
// ONEWire Sensors
#include <OneWire.h>
#include <DallasTemperature.h>
#define ONE_WIRE_BUS 2
OneWire oneWire(ONE_WIRE_BUS);
DallasTemperature sensors(&oneWire);
// DHT11
#include <dht11.h>
dht11 DHT;
#define DHT11_PIN 17
// BMP180 Barometric Pressure & Temp Sensor
#include <Adafruit_BMP085.h>
Adafruit_BMP085 bmp;
// MPL115A2
#include <Adafruit_MPL115A2.h>
Adafruit_MPL115A2 mpl115a2;

// size of buffer used to capture HTTP requests
#define REQ_BUF_SZ 60

// MAC address from Ethernet shield sticker under board
byte mac[] = { 0xDE, 0xAD, 0xBE, 0xEF, 0xFE, 0xED };
IPAddress ip(192, 168, 1, 123); // IP address
EthernetServer server(80); // port 80
File webFile; // the web page file on the SD card
```

```
char HTTP_req[REQ_BUF_SZ] = {0}; // buffered HTTP
char req_index = 0;           // index into HTTP_req buffer

void setup()
{
  Serial.begin(9600);
  sensors.begin();
  //Adding time
  Wire.begin();
  RTC.begin();
  //If we remove the comment from the following line, we will set up the module time and
  date with the computer one
  RTC.adjust(DateTime(__DATE__, __TIME__));

  // Ethernet chip (PIN)
  pinMode(10, OUTPUT);
  digitalWrite(10, HIGH);

  // initialize SD card
  Serial.println("Initializing SD card...");
  if (!SD.begin(4)) {
    Serial.println("ERROR - SD card initialization failed!");
    return; // init failed
  }
  Serial.println("SUCCESS - SD card initialized.");
  // check for index.htm file
  if (!SD.exists("index.htm")) {
    Serial.println("ERROR - Can't find index.htm file!");
    return; // can't find index file
  }
  Serial.println("SUCCESS - Found index.htm file.");

  // Check BMP180 sensor
  if (!bmp.begin()) {
    Serial.println("Could not find a valid BMP180 sensor, check wiring!");
    while (1) {}
  }

  Ethernet.begin(mac, ip); // initialize Ethernet device
  server.begin();         // start to listen for clients

  mpl115a2.begin();
}

void loop()
{
  // Read DHT11 Sensors
  int chk;
  chk = DHT.read(DHT11_PIN); // READ DATA
  switch (chk){
    case DHTLIB_OK:
      // Serial.println("DHT11, \t OK,\t");
      break;
    case DHTLIB_ERROR_CHECKSUM:
```

```
        Serial.println("DHT11, \t Checksum error,\t");
        break;
    case DHTLIB_ERROR_TIMEOUT:
        Serial.println("DHT11, \t Time out error,\t");
        break;
    default:
        Serial.println("DHT11, \t Unknown error,\t");
        break;
}
delay(500);

EthernetClient client = server.available(); // try to get client

if (client) { // got client?
    boolean currentLineIsBlank = true;
    while (client.connected()) {
        if (client.available()) { // client data available to read
            char c = client.read(); // read 1 byte (character) from client
            // limit the size of the stored received HTTP request
            // buffer first part of HTTP request in HTTP_req array (string)
            // leave last element in array as 0 to null terminate string (REQ_BUF_SZ - 1)
            if (req_index < (REQ_BUF_SZ - 1)) {
                HTTP_req[req_index] = c; // save HTTP request character
                req_index++;
            }
            // last line of client request is blank and ends with \n
            // respond to client only after last line received
            if (c == '\n' && currentLineIsBlank) {
                // send a standard http response header
                client.println("HTTP/1.1 200 OK");
                // remainder of header follows below, depending on if
                // web page or XML page is requested
                // Ajax request - send XML file
                if (StrContains(HTTP_req, "ajax_inputs")) {
                    // send rest of HTTP header
                    client.println("Content-Type: text/xml");
                    client.println("Connection: keep-alive");
                    client.println();
                    // send XML file containing input states
                    XML_response(client);
                }
                else { // web page request
                    // send rest of HTTP header
                    client.println("Content-Type: text/html");
                    client.println("Connection: keep-alive");
                    client.println();
                    // send web page
                    webFile = SD.open("index.htm"); // open web page file
                    if (webFile) {
                        while(webFile.available()) {
                            client.write(webFile.read()); // send web page to client
                        }
                        webFile.close();
                    }
                }
            }
        }
    }
}
```

```
    }
    // display received HTTP request on serial port
    Serial.print(HTTP_req);
    // reset buffer index and all buffer elements to 0
    req_index = 0;
    StrClear(HTTP_req, REQ_BUF_SZ);
    break;
}
// every line of text received from the client ends with \r\n
if (c == '\n') {
    // last character on line of received text
    // starting new line with next character read
    currentLineIsBlank = true;
}
else if (c != '\r') {
    // a text character was received from client
    currentLineIsBlank = false;
}
} // end if (client.available())
} // end while (client.connected())
delay(1); // give the web browser time to receive the data
client.stop(); // close the connection
} // end if (client)

sensors.requestTemperatures();

// Print All Data
Serial.println("-----");
// Print Date - Time
DateTime now = RTC.now();
SerialDigits(now.day());
Serial.print('/');
SerialDigits(now.month());
Serial.print('/');
Serial.print(now.year(), DEC);
Serial.print(" ");
SerialDigits(now.hour());
Serial.print(':');
SerialDigits(now.minute());
Serial.println("");
// Print Sensors Data...
Serial.print(sensors.getTempCByIndex(0));
Serial.println(" *C [18B20]");
Serial.print(sensors.getTempCByIndex(1));
Serial.println(" *C [18B20]");
Serial.print(DHT.humidity,1);
Serial.println(" % [DHT11]");
Serial.print(DHT.temperature,1);
Serial.println(" *C [DHT11]");
Serial.print(bmp.readTemperature());
Serial.println(" *C [BMP180]");
Serial.print(bmp.readPressure());
Serial.println(" Pa [BMP180]");
Serial.print(bmp.readAltitude());
```

```
Serial.println(" meters [BMP180]");
Serial.print(bmp.readAltitude(101500));
Serial.println(" meters (Real) [BMP180]");
Serial.print(mpl115a2.getPressure(), 4);
Serial.println(" kPa [MPL115A2]");
Serial.print(mpl115a2.getTemperature(), 1);
Serial.println(" *C [MPL115A2]");
Serial.println("-----");
}

// send the XML file with Sensors values
void XML_response(EthernetClient cl)
{
  cl.print("<?xml version = \"1.0\" ?>");
  cl.print("<inputs>");
  // Read time and date
  DateTime now = RTC.now();
  cl.print("<date>");
  cl.print(now.day());
  cl.print("/");
  cl.print(now.month());
  cl.print("/");
  cl.print(now.year(), DEC);
  cl.print(" ");
  cl.print(now.hour());
  cl.print(":");
  cl.print(now.minute());
  cl.println("</date>");
  // Read Sensors DS18B20
  cl.print("<DS18B20>");
  cl.print(sensors.getTempCByIndex(0));
  cl.println("</DS18B20>");
  cl.print("<DS18B20>");
  cl.print(sensors.getTempCByIndex(1));
  cl.println("</DS18B20>");
  // Read Sensors DHT11
  cl.print("<DHT11>");
  cl.print(DHT.humidity,1);
  cl.println("</DHT11>");
  cl.print("<DHT11>");
  cl.print(DHT.temperature,1);
  cl.println("</DHT11>");
  // Read BMP180
  cl.print("<BMP180>");
  cl.print(bmp.readTemperature());
  cl.println("</BMP180>");
  cl.print("<BMP180>");
  cl.print(bmp.readPressure());
  cl.println("</BMP180>");
  cl.print("<BMP180>");
  cl.print(bmp.readAltitude());
  cl.println("</BMP180>");
  cl.print("<BMP180>");
  cl.print(bmp.readAltitude(101500));
```

```
    cl.println("</BMP180>");
// Read MPL115A2
    cl.print("<MPL115A2>");
    cl.print(mpl115a2.getPressure(), 4);
    cl.println("</MPL115A2>");
    cl.print("<MPL115A2>");
    cl.print(mpl115a2.getTemperature(), 1);
    cl.println("</MPL115A2>");
    cl.print("</inputs>");
}

// sets every element of str to 0 (clears array)
void StrClear(char *str, char length)
{
    for (int i = 0; i < length; i++) {
        str[i] = 0;
    }
}

// searches for the string sfind in the string str
// returns 1 if string found
// returns 0 if string not found
char StrContains(char *str, char *sfind)
{
    char found = 0;
    char index = 0;
    char len;
    len = strlen(str);
    if (strlen(sfind) > len) {
        return 0;
    }
    while (index < len) {
        if (str[index] == sfind[found]) {
            found++;
            if (strlen(sfind) == found) {
                return 1;
            }
        }
        else {
            found = 0;
        }
        index++;
    }
    return 0;
}

void SerialDigits(int digits){
    if(digits < 10)
        Serial.print('0');
    Serial.print(digits);
}
```

## 12.10 Ο προγραμματισμός του Web Server

Με τον παρακάτω κώδικα μετατρέπεται ο Arduino σε Web Server, έτσι ώστε όταν καλείς την ip του arduino μέσω του browser ενός ηλεκτρονικού υπολογιστή του ίδιου δικτύου να απαντάει και να στέλνει το αποτέλεσμα που έχουμε ορίσει μέσω του κώδικα.

```
// Ethernet
#include <SPI.h>
#include <Ethernet.h>
#include <SD.h>
// size of buffer used to capture HTTP requests
#define REQ_BUF_SZ 60

// MAC address from Ethernet shield sticker under board
byte mac[] = { 0xDE, 0xAD, 0xBE, 0xEF, 0xFE, 0xED };
IPAddress ip(192, 168, 1, 123); // IP address
EthernetServer server(80); // port 80
File webFile; // the web page file on the SD card
char HTTP_req[REQ_BUF_SZ] = {0}; // buffered HTTP
char req_index = 0; // index into HTTP_req buffer

void setup()
{
  Serial.begin(9600);
  // Ethernet chip (PIN)
  pinMode(10, OUTPUT);
  digitalWrite(10, HIGH);

  // initialize SD card
  Serial.println("Initializing SD card...");
  if (!SD.begin(4)) {
    Serial.println("ERROR - SD card initialization failed!");
    return; // init failed
  }
  Serial.println("SUCCESS - SD card initialized.");
  // check for index.htm file
  if (!SD.exists("index.htm")) {
    Serial.println("ERROR - Can't find index.htm file!");
    return; // can't find index file
  }
  Serial.println("SUCCESS - Found index.htm file.");
}

void loop()
{
  EthernetClient client = server.available(); // try to get client

  if (client) { // got client?
    boolean currentLineIsBlank = true;
    while (client.connected()) {
      if (client.available()) { // client data available to read
        char c = client.read(); // read 1 byte (character) from client
        // limit the size of the stored received HTTP request
        // buffer first part of HTTP request in HTTP_req array (string)

```

```
// leave last element in array as 0 to null terminate string (REQ_BUF_SZ - 1)
if (req_index < (REQ_BUF_SZ - 1)) {
    HTTP_req[req_index] = c;    // save HTTP request character
    req_index++;
}
// last line of client request is blank and ends with \n
// respond to client only after last line received
if (c == '\n' && currentLineIsBlank) {
    // send a standard http response header
    client.println("HTTP/1.1 200 OK");
    // remainder of header follows below, depending on if
    // web page or XML page is requested
    // Ajax request - send XML file
    if (StrContains(HTTP_req, "ajax_inputs")) {
        // send rest of HTTP header
        client.println("Content-Type: text/xml");
        client.println("Connection: keep-alive");
        client.println();
        // send XML file containing input states
        XML_response(client);
    }
    else { // web page request
        // send rest of HTTP header
        client.println("Content-Type: text/html");
        client.println("Connection: keep-alive");
        client.println();
        // send web page
        webFile = SD.open("index.htm");    // open web page file
        if (webFile) {
            while(webFile.available()) {
                client.write(webFile.read()); // send web page to client
            }
            webFile.close();
        }
    }
    // display received HTTP request on serial port
    Serial.print(HTTP_req);
    // reset buffer index and all buffer elements to 0
    req_index = 0;
    StrClear(HTTP_req, REQ_BUF_SZ);
    break;
}
// every line of text received from the client ends with \r\n
if (c == '\n') {
    // last character on line of received text
    // starting new line with next character read
    currentLineIsBlank = true;
}
else if (c != '\r') {
    // a text character was received from client
    currentLineIsBlank = false;
}
} // end if (client.available())
} // end while (client.connected())
```

```
    delay(1);    // give the web browser time to receive the data
    client.stop(); // close the connection
  } // end if (client)
}
```

## 12.11 Κατασκευή σελίδας

Επειδή ο κώδικας είναι μεγάλος, δημιουργήθηκε ένα αρχείο .html που αποθηκεύθηκε στην Micro SD Card της Ethernet Shield και προγραμματίστηκε το Arduino να διαβάζει από τον κώδικα το αρχείο αυτό, ώστε να μην επιβαρυνθεί η εσωτερική μνήμη του Arduino.

Στο αρχείο .html περιέχει τον κώδικα προβολής των τιμών. Η προβολή των τιμών γίνεται σε πραγματικό χρόνο και ανά 1 δευτερόλεπτο οι τιμές της σελίδας ανανεώνονται.

```
<!DOCTYPE html>
<html>
  <head>
    <title>Arduino Weather Station</title>
    <script>
      function GetArduinoIO()
      {
        nocache = "&nocache=" + Math.random() * 1000000;
        var request = new XMLHttpRequest();
        request.onreadystatechange = function()
        {
          if (this.readyState == 4) {
            if (this.status == 200) {
              if (this.responseXML != null) {
                // XML file received - contains Sensors
                values
                var count;
                // Read DATE - TIME
                var num_an =
                this.responseXML.getElementsByTagName('date').length;
                for (count = 0; count < num_an;
                count++) {
                  document.getElementsByClassName("date")[count].innerHTML =
                  this.responseXML.getElementsByTagName('date')[count].childNodes[0].nodeValue;
                  }
                  // Read Sensor 18B20
                  var num_an =
                  this.responseXML.getElementsByTagName('DS18B20').length;
                  for (count = 0; count < num_an;
                  count++) {
                    document.getElementsByClassName("DS18B20")[count].innerHTML =
                    this.responseXML.getElementsByTagName('DS18B20')[count].childNodes[0].nodeValue;
                    }
                    // Read Sensor DHT11
```

```
var num_an =
this.responseXML.getElementsByTagName('DHT11').length;
for (count = 0; count < num_an;
count++) {

    document.getElementsByClassName("DHT11")[count].innerHTML =
    this.responseXML.getElementsByTagName('DHT11')[count].childNodes[0].nodeValue;
}
// Read Sensor BMP180
var num_an =
this.responseXML.getElementsByTagName('BMP180').length;
for (count = 0; count < num_an;
count++) {

    document.getElementsByClassName("BMP180")[count].innerHTML =
    this.responseXML.getElementsByTagName('BMP180')[count].childNodes[0].nodeValue;
}
// Read Sensor MPL115A2
var num_an =
this.responseXML.getElementsByTagName('MPL115A2').length;
for (count = 0; count < num_an;
count++) {

    document.getElementsByClassName("MPL115A2")[count].innerHTML =
    this.responseXML.getElementsByTagName('MPL115A2')[count].childNodes[0].nodeValue;
}
}
}
}
}
request.open("GET", "ajax_inputs" + nocache, true);
request.send(null);
setTimeout('GetArduinoIO()', 1000);
}
</script>
<style>
.IO_box {
    float: left;
    margin: 0 20px 20px 0;
    border: 1px solid blue;
    padding: 0 5px 0 5px;
    width: 20%;
}
h1 {
    font-size: 120%;
    color: blue;
    margin: 0 0 10px 0;
}
h2 {
    font-size: 85%;
    color: #5734E6;
    margin: 5px 0 5px 0;
}
p, form, button {
```

```
        font-size: 80%;
        color: #252525;
    }
    span {
        font-weight: bold;
        color: #000;
    }
</style>
</head>
<body onload="GetArduinoIO()">
    <h1>Arduino Weather Station</h1>
    <hr><br>
    <h2>Date / Time</h2>
    <p><span class="date">...</span></p>
    <br>
    <div class="IO_box">
        <h2>Sensor 18B20</h2>
        <p>Temperature 18B20 1 : <span class="DS18B20">...</span> *C</p>
        <p>Temperature 18B20 2 : <span class="DS18B20">...</span> *C</p>
    </div>
    <div class="IO_box">
        <h2>Sensor DHT11</h2>
        <p>DHT11 Humidity : <span class="DHT11">...</span> %</p>
        <p>DHT11 Temperature : <span class="DHT11">...</span> *C</p>
    </div>
    <div class="IO_box">
        <h2>BMP180 Sensor</h2>
        <p>Temperature : <span class="BMP180">...</span> *C</p>
        <p>Pressure : <span class="BMP180">...</span> Pa</p>
        <p>Altitude : <span class="BMP180">...</span> meters</p>
        <p>Real altitude : <span class="BMP180">...</span> meters</p>
    </div>
    <div class="IO_box">
        <h2>Sensor MPL115A2</h2>
        <p>Pressure : <span class="MPL115A2">...</span> kPa</p>
        <p>Temperature : <span class="MPL115A2">...</span> *C</p>
    </div>
</body>
</html>
```

### 13 ΈΛΕΓΧΟΣ ΣΥΣΤΗΜΑΤΟΣ

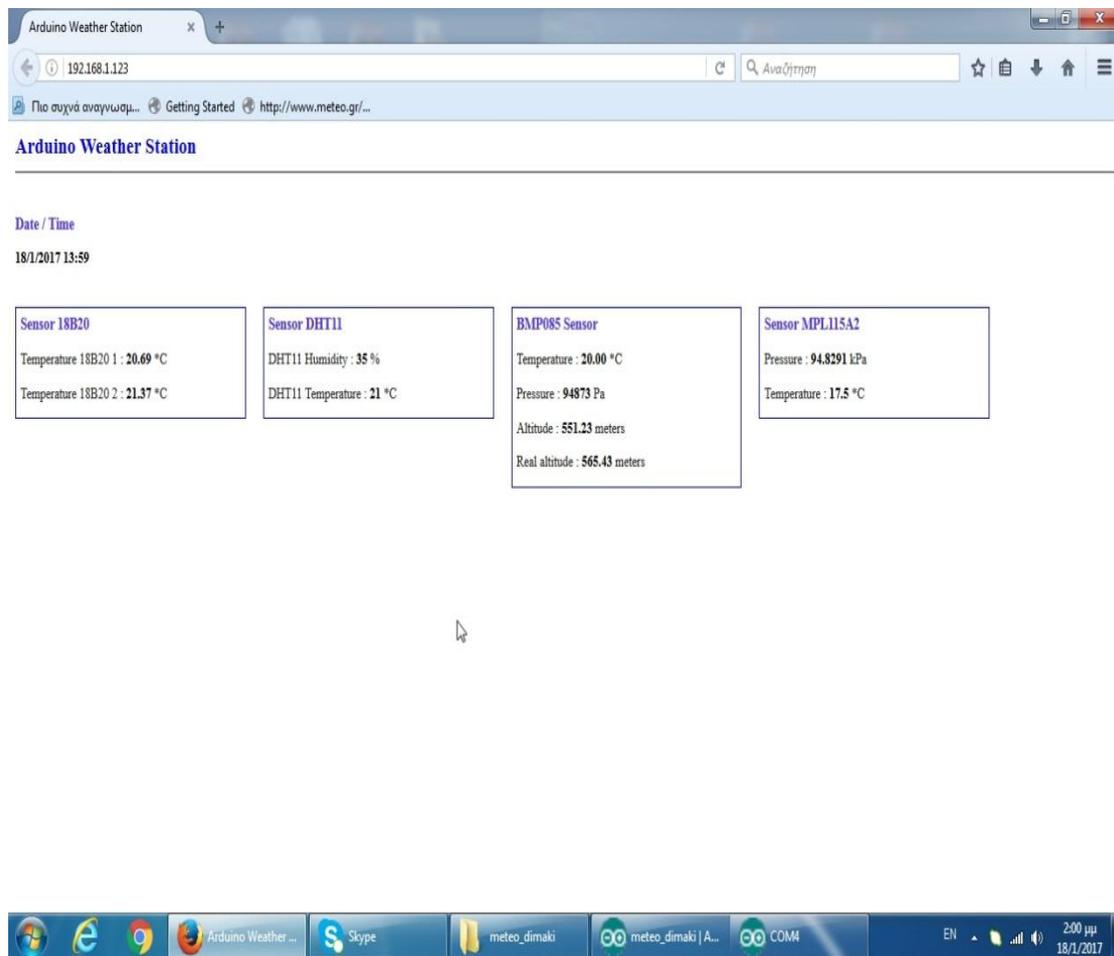
Σε αυτό το κεφάλαιο θα γίνει έλεγχος για το εάν επικοινωνούν οι αισθητήρες με την σελίδα του local host και εμφανίζονται τα αποτελέσματα των αισθητήρων καθώς και για την σωστή ένδειξη της ημερομηνίας και της ώρας.

- Δοκιμαστικός έλεγχος που υλοποιήθηκε στις 18/1/2017.

Στις παρακάτω εικόνες φαίνεται η επικοινωνία του συστήματος με την σελίδα και διαπιστώνεται η σωστή επικοινωνία μεταξύ τους.

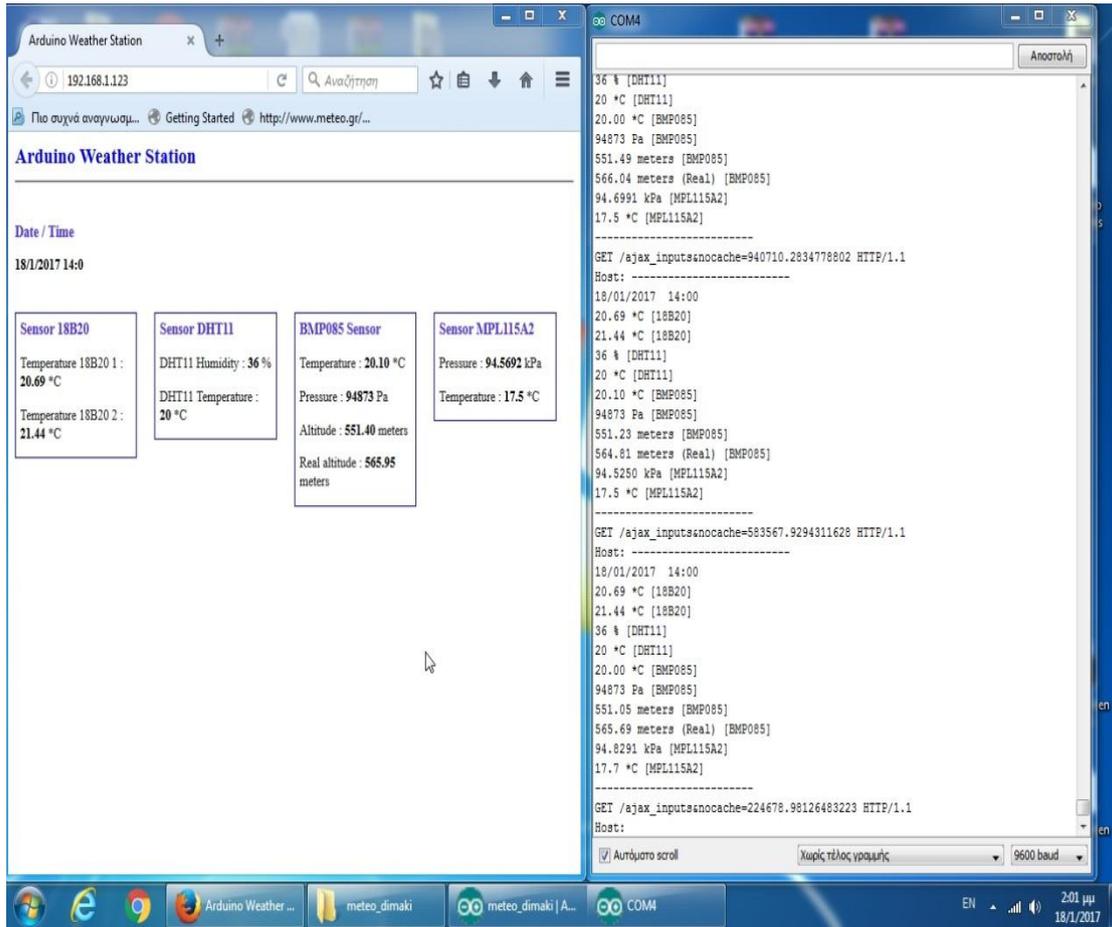
#### A. Η εμφάνιση των τιμών στην σελίδα.

Στις 2 το μεσημέρι.

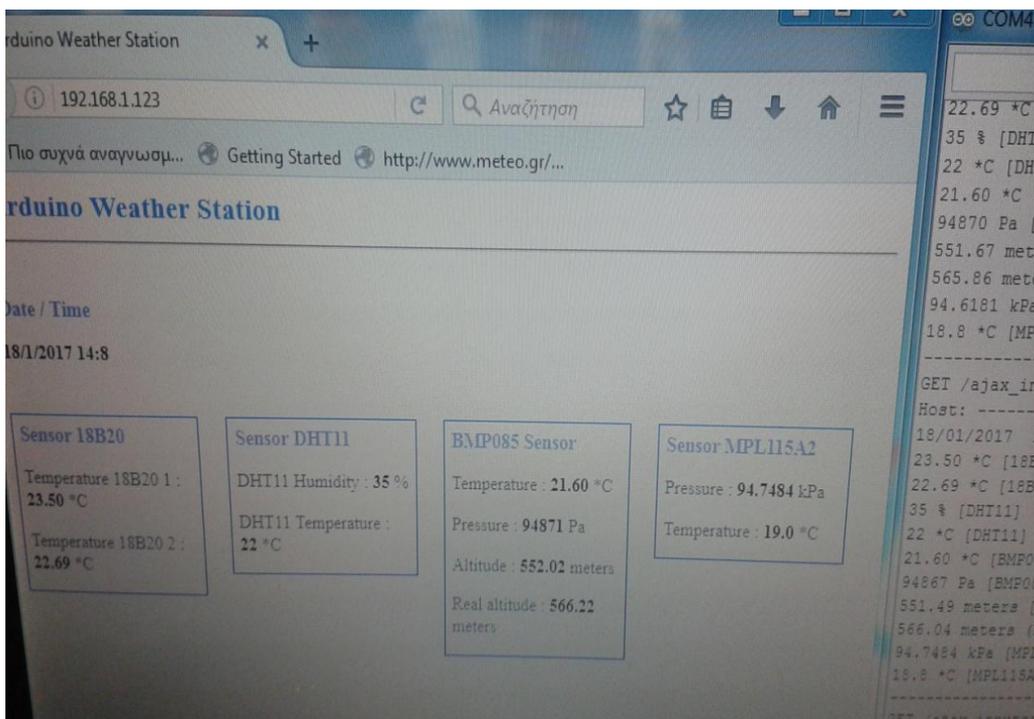


Εικόνα 94. Ο έλεγχος του συστήματος στις 18/1/2017 το μεσημέρι.

ΔΗΜΙΟΥΡΓΙΑ ΕΝΟΣ ΜΕΤΕΩΡΟΛΟΓΙΚΟΥ ΣΤΑΘΜΟΥ ΜΕ ΧΡΗΣΗ ΜΙΚΡΟΕΛΕΓΚΤΗ ΚΑΙ ΕΜΦΑΝΙΣΗ ΔΕΔΟΜΕΝΩΝ ΣΕ ΣΕΛΙΔΑ

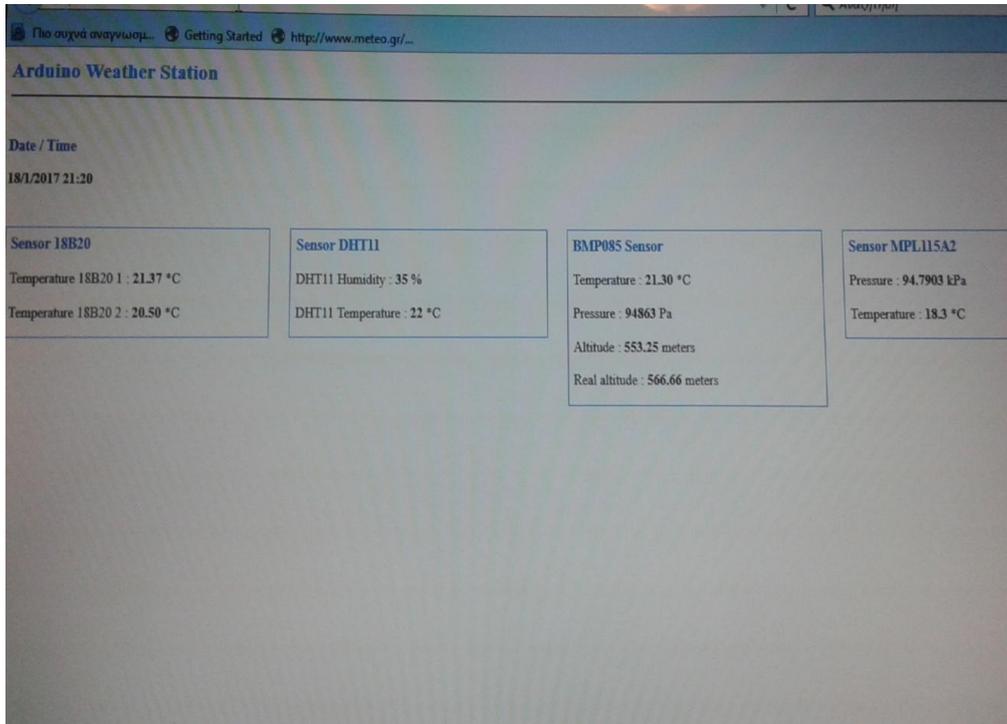


Εικόνα 95. Ο έλεγχος του συστήματος στις 18/1/2017 το μεσημέρι.



Εικόνα 96. Ο έλεγχος του συστήματος στις 18/1/2017 το μεσημέρι.

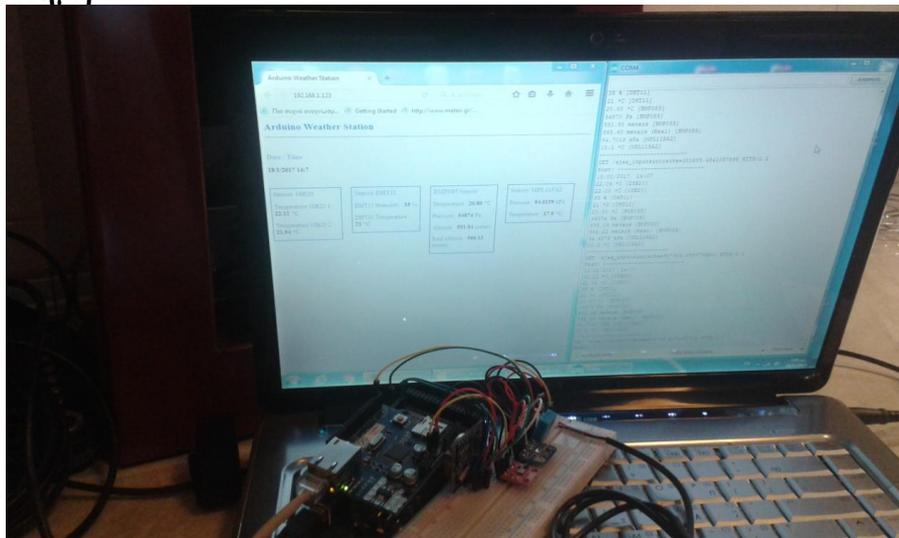
Στις 9 το βράδυ.



Εικόνα 97. Ο έλεγχος του συστήματος στις 18/1/2017 το βράδυ.

**Β. Ο έλεγχος στο φυσικό περιβάλλον παρουσία και άλλων θερμομετρικών συστημάτων.**

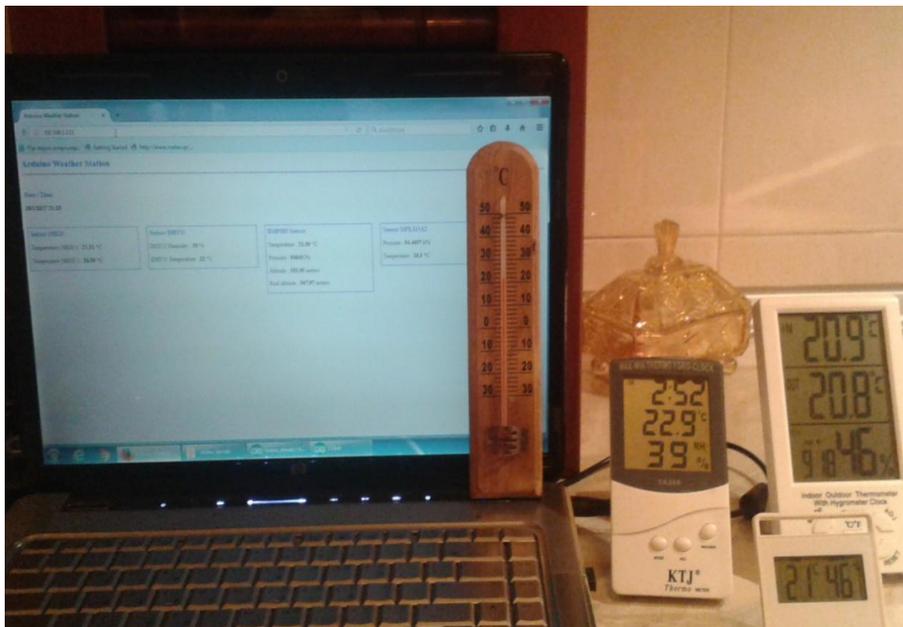
Στις 2 το μεσημέρι.



Εικόνα 98. Ο έλεγχος του συστήματος με το φυσικό περιβάλλον στις 18/1/2017 το μεσημέρι.

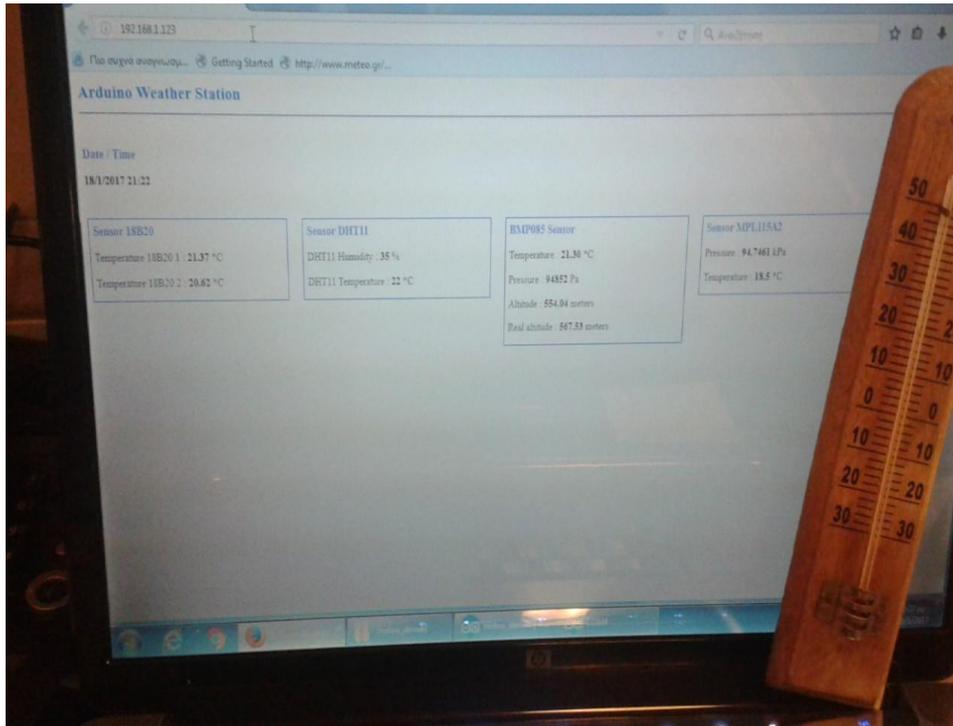


Εικόνα 99. Ο έλεγχος του συστήματος με το φυσικό περιβάλλον στις 18/1/2017 το μεσημέρι.

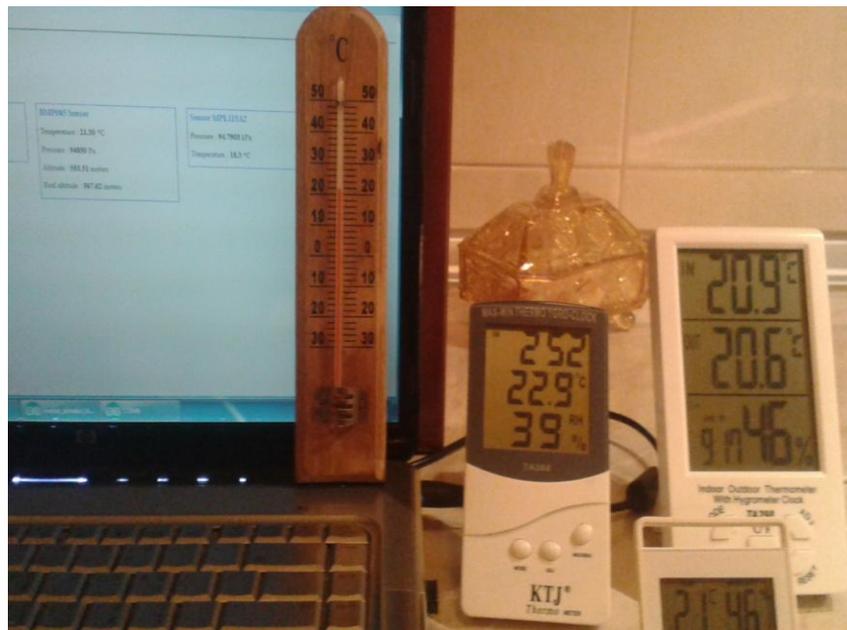


Εικόνα 100. Ο έλεγχος του συστήματος με το φυσικό περιβάλλον στις 18/1/2017 το μεσημέρι.

Στις 9 το βράδυ.



Εικόνα 101. Ο έλεγχος του συστήματος με το φυσικό περιβάλλον στις 18/1/2017 το βράδυ.



Εικόνα 102. Ο έλεγχος του συστήματος με το φυσικό περιβάλλον στις 18/1/2017 το βράδυ.

## 14 ΣΥΜΠΕΡΑΣΜΑΤΑ

Μετά την αποπεράτωση της εργασίας, διαπίστωσα τελικά ότι το θέμα της πτυχιακής μου είναι πολύ ενδιαφέρον και ο οποιοσδήποτε έχει κάποιες στοιχειώδεις γνώσεις προγραμματισμού μπορεί να ασχοληθεί ευχάριστα και με συνεχή αναζήτηση μπορεί να βελτιωθεί. Το τελευταίο διάστημα παρατηρείται σε κάποιες πόλεις της χώρας μας μια προσπάθεια απόκτησης μίας αρχικής γνωριμίας των μαθητών των τελευταίων τάξεων δημοτικού με το Arduino από καθηγητές πληροφορικής. Οι εντυπώσεις των μαθητών είναι καταπληκτικές, καθώς τους είναι εύχρηστο και κατανοήσιμο.

Όσον αναφορά τις μετρίσιμες τιμές των αισθητήρων κάποιοι είναι εντός ορίων και άλλοι κυμαίνονται μέσα στα πλαίσια της κατασκευαστικής τους απόκλισης λάθους.

Εάν θέλαμε να μελετήσουμε την μεταξύ τους διακύμανση στις τιμές θερμοκρασίας, μιας και αυτή η μέτρηση λαμβάνεται από όλους τους αισθητήρες, ίσως να έπρεπε να λάβουμε υπόψη μας και ορισμένα άλλα στοιχεία, όπως:

- i. Τον τύπο του αισθητήρα.  
Ο DS18B20 είναι ένα πάρα πολύ μικρό chip σε μέγεθος, σε αυτό συμβάλλει και το πρωτόκολλο επικοινωνίας του.  
Ομοίως και για τον DS18B20-Waterproof, ο οποίος διαθέτει και καλώδιο περίπου 1 μέτρο.  
Ο BMP180 είναι ένα chip το οποίο όμως βρίσκεται τοποθετημένο πάνω σε μία μικρή πλακέτα, που ενδεχομένως να επηρεάζει στην τιμή λήψης.  
Ο MPL115A2 ακολουθεί και ο ίδιος την ίδια διάταξη με τον BMP180.  
Ο DHT11 είναι και αυτός ένα μικρό chip που ακολουθεί το ίδιο απλό πρωτόκολλο επικοινωνίας με τους DS18B20, ωστόσο η διάταξή του είναι λίγο πιο μεγάλη από την δική τους.
- ii. Το υλικό κατασκευής που περιβάλλει τους αισθητήρες.  
Ο DS18B20 περικλείεται μέσα σε ένα μάλλον κεραμικό υλικό, ενώ ο DS18B20-Waterproof είναι μέσα σε ένα κυλινδρικό μεταλλικό περίβλημα. Οι αισθητήρες αυτοί είναι ιδιαίτερα ανθεκτικοί στις μηχανικές επιδράσεις και μπορούν να δουλέψουν χωρίς φθορές.  
Ο BMP180 αποτελεί ένα στοιχείο μέτρησης πιεζοαντίστασης πυριτίου, που όμως είναι εξίσου καλό με τους αισθητήρες που διέθεταν κεραμικό στοιχείο.  
Ο MPL115A2 είναι ένα στοιχείο μέτρησης πάνω σε μία μικρή πλακέτα.  
Ο DHT11 είναι εξωτερικά προστατευμένος από ένα πλαστικό περίβλημα.

Από τον ταυτόχρονο έλεγχο που έγινε στις 18/1/2017 στις 9 η ώρα το βράδυ στις τιμές των αισθητήρων και στα ψηφιακά θερμομέτρα που αγοράστηκαν για την συγκεκριμένη εργασία, καταγράφηκαν οι εξής τιμές:

AVIDSEN	TA-298	TA-368	DS18B20	DS18B20-Waterproof	DHT11	BMP180	MPL115A2
21 °C	20.9°C	22.9°C	21.37°C	20.5°C	22°C	21.3°C	18.3°C

**Πίνακας 17. Καταγραφή των τιμών της θερμοκρασίας από όλα τα ψηφιακά όργανα και τους αισθητήρες.**

Δυσκολίες που προέκυψαν κατά την διάρκεια της υλοποίησης της εργασίας ήταν αρκετές είτε λόγω της κακής επαφής των υλικών είτε λόγω προβλημάτων στον κώδικα

## 15 ΠΡΟΤΕΙΝΟΜΕΝΗ ΕΠΕΚΤΑΣΗ ΤΗΣ ΕΡΓΑΣΙΑΣ

Στο κεφάλαιο αυτό παραθέτετε μια προσπάθεια επέκταση της εργασίας με την δημιουργία γραφημάτων. Με την βοήθεια της κάρτας SD αποθηκεύεται κώδικας σε αυτήν, ώστε να «ελαφρυνθεί» η μνήμη του Arduino.

- Στο παρακάτω κομμάτι δηλώνονται οι βιβλιοθήκες της MySQL μέσα στον κώδικα του Arduino για να σταλούν τα στοιχεία στον server.

```
// *** MySQL Connection START***
#include <MySQL_Connection.h>
#include <MySQL_Cursor.h>
IPAddress server_addr(192,168,1,113); // Mysql Server IP
char user[] = "arduino"; // Mysql Username
char password[] = "root"; // Mysql Password
EthernetClient client;
MySQL_Connection conn((Client *)&client);
unsigned long currentTime;
unsigned long loopTime;
// *** MySQL Connection END***
```

- Στο παρακάτω κομμάτι του κώδικα γίνεται η αποστολή των δεδομένων στην MySQL μέσα στον κώδικα του Arduino.

```
// *** Send Data MySQL START ***
currentTime = millis();
if(currentTime >= (loopTime + 60000)){
  loopTime = currentTime;
  char INSERT_DATA[] = "INSERT INTO weather_stasion.data
(18b20_0,18b20_1,dht_hum,dht_temp,bmp_temp,bmp_pres,bmp_alt,bmp_real_alt,mp_pres,
mp_temp) VALUES ('%s','%s','%s','%s','%s','%s','%s','%s','%s','%s')";
  char query[256];
  MySQL_Cursor *cur_mem = new MySQL_Cursor(&conn);
  float d_18b20_0 = sensors.getTempCByIndex(0);
  float d_18b20_1 = sensors.getTempCByIndex(1);
  float d_dht_hum = DHT.humidity;
  float d_dht_temp = DHT.temperature;
  float d_bmp_temp = bmp.readTemperature();
  float d_bmp_pres = bmp.readPressure();
  float d_bmp_alt = bmp.readAltitude();
  float d_bmp_real_alt = bmp.readAltitude(101500);
  float d_mp_pres = mpl115a2.getPressure();
  float d_mp_temp = mpl115a2.getTemperature();
  if (d_mp_temp < -50){d_mp_temp = 0;}
  char data_18b20_0[10];
  char data_18b20_1[10];
  char data_dht_hum[10];
  char data_dht_temp[10];
  char data_bmp_temp[10];
  char data_bmp_pres[10];
  char data_bmp_alt[10];
```

```
char data_bmp_real_alt[10];
char data_mp_pres[10];
char data_mp_temp[10];
dtostrf(d_18b20_0, 1, 1, data_18b20_0);
dtostrf(d_18b20_1, 1, 1, data_18b20_1);
dtostrf(d_dht_hum, 1, 1, data_dht_hum);
dtostrf(d_dht_temp, 1, 1, data_dht_temp);
dtostrf(d_bmp_temp, 1, 1, data_bmp_temp);
dtostrf(d_bmp_pres, 1, 1, data_bmp_pres);
dtostrf(d_bmp_alt, 1, 1, data_bmp_alt);
dtostrf(d_bmp_real_alt, 1, 1, data_bmp_real_alt);
dtostrf(d_mp_pres, 1, 1, data_mp_pres);
dtostrf(d_mp_temp, 1, 1, data_mp_temp);
sprintf(query, INSERT_DATA, data_18b20_0, data_18b20_1, data_dht_hum,
data_dht_temp, data_bmp_temp, data_bmp_pres, data_bmp_alt, data_bmp_real_alt,
data_mp_pres, data_mp_temp);
cur_mem->execute(query);
delete cur_mem;
}
// *** Send Data MySQL END ***
```

- Με τη βοήθεια του XAMPP δημιουργείται η βάση δεδομένων της MySQL, ώστε να αποθηκεύονται τα δεδομένα και να βλέπουμε τα γραφήματα.

Στο παρακάτω κομμάτι κώδικα φαίνεται περιγράφεται η δημιουργία της βάσης δεδομένων και ο πίνακας weather\_station.data που είναι αποθηκευμένο μέσα σε ένα αρχείο .sql .

Βάση δεδομένων: weather\_station

```
CREATE TABLE `data` (
`id` int(11) NOT NULL,
`18b20_0` text NOT NULL,
`18b20_1` text NOT NULL,
`dht_hum` text NOT NULL,
`dht_temp` text NOT NULL,
`bmp_temp` text NOT NULL,
`bmp_pres` text NOT NULL,
`bmp_alt` text NOT NULL,
`bmp_real_alt` text NOT NULL,
`mp_pres` text NOT NULL,
`mp_temp` text NOT NULL,
`real_date` timestamp NOT NULL DEFAULT CURRENT_TIMESTAMP
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

## ΒΙΒΛΙΟΓΡΑΦΙΑ

Αξιοτέλλης, Γ. (1993). *Σχολική Υδρία*, Αθήνα: εκδ. ιδίου.

Εθνική Μετεωρολογική Υπηρεσία (χ.χ.). *Εισαγωγή στη Μετεωρολογία από την αρχαία Ελλάδα μέχρι σήμερα*. Ανακτήθηκε στις 12 Δεκεμβρίου 2016 από: [http://www.hnms.gr/hnms/greek/meteorology/full\\_story\\_html?dr\\_url=%2Fhnms%2Fdocrep%2Fdocs%2Fmisc%2FArxaia](http://www.hnms.gr/hnms/greek/meteorology/full_story_html?dr_url=%2Fhnms%2Fdocrep%2Fdocs%2Fmisc%2FArxaia)

Εθνική Μετεωρολογική Υπηρεσία (χ.χ.). *Σχετικά με την ΕΜΥ*. Ανακτήθηκε στις 12 Δεκεμβρίου 2016 από: [http://www.meteo.noa.gr/GR/history\\_gr.html](http://www.meteo.noa.gr/GR/history_gr.html)

Εθνικό Αστεροσκοπείο Αθηνών (χ.χ.). *Ιστορία του Εθνικού Αστεροσκοπείου Αθηνών*. Ανακτήθηκε στις 12 Δεκεμβρίου 2016 από: [http://www.noa.gr/index.php?option=com\\_content&view=article&id=79&Itemid=525&lang=el](http://www.noa.gr/index.php?option=com_content&view=article&id=79&Itemid=525&lang=el)

Ινστιτούτο Εθνικού Περιβάλλοντος και Βιώσιμης Ανάπτυξης (2013). *Το αρχαιότερο Ερευνητικό Κέντρο στην Ελλάδα και τα Βαλκάνια*. Ανακτήθηκε στις 12 Δεκεμβρίου 2016 από: [http://www.meteo.noa.gr/GR/iersd\\_facilities\\_gr.htm](http://www.meteo.noa.gr/GR/iersd_facilities_gr.htm)

Περογιαννάκης, Χ. (1974). *Ναυτική Μετεωρολογία*. Αθήνα: Ίδρυμα Ευγενίδου  
Ανακτήθηκε στις 12 Δεκεμβρίου 2016 από: [https://el.wikipedia.org/wiki/%CE%91%CF%84%CE%BC%CE%BF%CF%83%CF%86%CE%B1%CE%B9%CF%81%CE%B9%CE%BA%CE%AE\\_%CF%80%CE%AF%CE%B5%CF%83%CE%B7](https://el.wikipedia.org/wiki/%CE%91%CF%84%CE%BC%CE%BF%CF%83%CF%86%CE%B1%CE%B9%CF%81%CE%B9%CE%BA%CE%AE_%CF%80%CE%AF%CE%B5%CF%83%CE%B7)

Γαλάνη, Δ. (2010). *Καταγραφή στοιχείων και επεξεργασία μετεωρολογικών δεδομένων Χανίων περιόδου 1933-1940*, Ανώτατο Εκπαιδευτικό Ίδρυμα Κρήτης, Τμήμα Φυσικών Πόρων και Περιβάλλοντος Τομέας Μετεωρολογίας, Πτυχιακή Εργασία. Χανιά.

Μακρυνίτσας, Κ. (2007). *Εισαγωγή στην μετεωρολογία*. Πήλιο: εκδ. ιδίου.

Σταματόπουλος, Σ. (2015). *Ανάπτυξη αυτόνομου σταθμού καταγραφής μετεωρολογικών φαινομένων*. Κρήτη: εκδ. ιδίου.

Καλωμενίδης, Φ. (2001). *Μετεωρολογία*. Ξάνθη.

Εθνική Μετεωρολογική Υπηρεσία (χ.χ.). *Πρόγνωση πόλεων*. Ανακτήθηκε στις 12 Δεκεμβρίου 2016 από: [http://www.hnms.gr/hnms/greek/forecast/forecast\\_region\\_html](http://www.hnms.gr/hnms/greek/forecast/forecast_region_html)

Εθνική Μετεωρολογική Υπηρεσία (χ.χ.). *Σχετικά με το Εθνικό Αστεροσκοπείο Αθηνών*. Ανακτήθηκε στις 12 Δεκεμβρίου 2016 από: <http://www.meteo.gr/meteoplus/about-meteo.cfm>

Εθνική Μετεωρολογική Υπηρεσία (χ.χ.). *Πρόγνωση πόλεων*. Ανακτήθηκε στις 12 Δεκεμβρίου 2016 από: [http://www.hnms.gr/hnms/greek/forecast/forecast\\_european\\_cities\\_html?dr\\_city=Istanbul](http://www.hnms.gr/hnms/greek/forecast/forecast_european_cities_html?dr_city=Istanbul)

Εθνική Μετεωρολογική Υπηρεσία (χ.χ.). *Πρόγνωση πόλεων*. Ανακτήθηκε στις 12 Δεκεμβρίου 2016 από: <http://www.meteo.gr/meteoplus/world.cfm>

Καλόμοιρος, Ι. (2015). *Εισαγωγή στους μικροελεγκτές PIC*. Ανώτατο Τεχνολογικό Ίδρυμα Κεντρικής Μακεδονίας, Σχολή Τεχνολογικών Εφαρμογών, Τμήμα Μηχανικών Πληροφορικής. [πανεπιστημιακές σημειώσεις]. Ανακτήθηκε στις 20 Δεκεμβρίου 2016 από: [http://teachers.teicm.gr/kalomiros/Mtptx/e-books/Embedded\\_PIC\\_new.pdf](http://teachers.teicm.gr/kalomiros/Mtptx/e-books/Embedded_PIC_new.pdf)

Δουμένης, Γ. (2015) *Ενσωματωμένα Συστήματα* Τεχνολογικό Ίδρυμα Ηπείρου, Σχολή Τεχνολογικών Εφαρμογών Τμήμα Μηχανικών Πληροφορικής. [πανεπιστημιακές σημειώσεις]. Άρτα.

Garde, D. (2001). *Προγραμματίζοντας τον μικροελεγκτή AVR*. Αθήνα: Εκδόσεις Τζιόλα.

Βούλγαρη, Ν. (1982). *ΕΙΣΑΓΩΓΗ ΣΤΟΥΣ ΜΙΚΡΟΕΠΕΞΕΡΓΑΣΤΕΣ ΚΑΙ ΣΤΟΝ ΠΡΟΓΡΑΜΜΑΤΙΣΜΟ ΤΟΥΣ*. Ανακτήθηκε στις 15 Δεκεμβρίου 2016 από: <https://el.wikipedia.org/wiki/%CE%9C%CE%B9%CE%BA%CF%81%CE%BF%CE%B5%CF%80%CE%B5%CE%BE%CE%B5%CF%81%CE%B3%CE%B1%CF%83%CF%84%CE%AE%CF%82>

Αλεβυζάκης, Α. & Μαράκης Γ. (2007). *Έλεγχος και προγραμματισμός Μικροϋπολογιστών Atmel*. Ανώτατο τεχνολογικό Εκπαιδευτικό Ίδρυμα Κρήτης, Σχολή Τεχνολόγων Εφαρμογών, Τμήμα Ηλεκτρολογίας, Πτυχιακή Εργασία. Ανακτήθηκε στις 20 Δεκεμβρίου 2016 από: <http://nefeli.lib.teicrete.gr/browse/stef/hle/2007/MarakisGiannis,AlevyzakisAristotelis/attached-document/2007Marakis.pdf>

Melear, C. & Kilbane, K. (n.d.) *Problems solved by FLASH microcontrollers* Ανακτήθηκε στις 20 Δεκεμβρίου 2016 από: [https://el.wikipedia.org/wiki/%CE%9C%CE%B9%CE%BA%CF%81%CE%BF%CE%B5%CE%BB%CE%B5%CE%B3%CE%BA%CF%84%CE%AE%CF%82#cite\\_note-arnold4-1](https://el.wikipedia.org/wiki/%CE%9C%CE%B9%CE%BA%CF%81%CE%BF%CE%B5%CE%BB%CE%B5%CE%B3%CE%BA%CF%84%CE%AE%CF%82#cite_note-arnold4-1)

Arduino. (n.d.). *Home* Ανακτήθηκε στις 20 Δεκεμβρίου 2016 από: [www.arduino.cc/en](http://www.arduino.cc/en)

Arduino. (n.d.). *Home* Ανακτήθηκε στις 20 Δεκεμβρίου 2016 από: <https://www.arduino.cc/>

Arduino. (n.d.). *Arduino Board Mega2560*. Ανακτήθηκε στις 20 Δεκεμβρίου 2016 από: <https://www.arduino.cc/en/Main/ArduinoBoardMega2560>

Arduino. (n.d.). *arduino-mega2560-schematic*. Ανακτήθηκε στις 20 Δεκεμβρίου 2016 από: <https://www.arduino.cc/en/uploads/Main/arduino-mega2560-schematic.pdf>

Arduino. (n.d.). *Reference*. Ανακτήθηκε στις 20 Δεκεμβρίου 2016 από: <https://www.arduino.cc/en/Reference/HomePage>

Arduino. (n.d.). *Forum*. Ανακτήθηκε στις 20 Δεκεμβρίου 2016 από: <http://forum.arduino.cc/index.php?action=forum>

Arduino. (n.d.). *Arduino Ethernet Shield V1*. Ανακτήθηκε στις 20 Δεκεμβρίου 2016 από: <https://www.arduino.cc/en/Main/ArduinoEthernetShieldV1>

Arduino. (n.d.). *Ethernet*. Ανακτήθηκε στις 20 Δεκεμβρίου 2016 από: <https://www.arduino.cc/en/Reference/Ethernet>

Arduino. (n.d.). *Guide Arduino Ethernet Shield*. Ανακτήθηκε στις 20 Δεκεμβρίου 2016 από: <https://www.arduino.cc/en/Guide/ArduinoEthernetShield>

Elgar, P. (2003) *Αισθητήρες Μέτρησης και Ελέγχου*. Αθήνα: Εκδόσεις ΤΖΙΟΛΑ.

Grobotronics. (n.d.). *DHT11*. Ανακτήθηκε στις 20 Δεκεμβρίου 2016 από: <https://grobotronics.com/dht11.html>

Wikispace. (n.d.). *DS18B20*. Ανακτήθηκε στις 20 Δεκεμβρίου 2016 από: <https://arduino-info.wikispaces.com/Brick-Temperature-DS18B20>

Maximintegrated. (n.d.). *DS18B20*. Ανακτήθηκε στις 20 Δεκεμβρίου 2016 από: <https://www.maximintegrated.com/en/products/analog/sensors-and-sensor-interface/DS18B20.html>

Github. (n.d.). *Library*. Ανακτήθηκε στις 20 Δεκεμβρίου 2016 από: <https://github.com/milesburton/Arduino-Temperature-Control-Library>

Hobbytronic. (n.d.). *DS18B20*. Ανακτήθηκε στις 20 Δεκεμβρίου 2016 από: <http://www.hobbytronics.co.uk/ds18b20-arduino>

Adafruit. (n.d.). *Product*. Ανακτήθηκε στις 20 Δεκεμβρίου 2016 από: <https://www.adafruit.com/product/992>

Grobotronics. (n.d.). *BMP180*. Ανακτήθηκε στις 20 Δεκεμβρίου 2016 από: <https://grobotronics.com/barometric-pressure-sensor-bmp180-breakout.html>

Adafruit. (n.d.). *MPL115A2*. Ανακτήθηκε στις 20 Δεκεμβρίου 2016 από: [https://github.com/adafruit/Adafruit\\_MPL115A2](https://github.com/adafruit/Adafruit_MPL115A2)

Maximintegrated. (n.d.). *DS1307*. Ανακτήθηκε στις 20 Δεκεμβρίου 2016 από: <https://www.maximintegrated.com/en/products/digital/real-time-clocks/DS1307.html>

Github. (n.d.). *RTClib*. Ανακτήθηκε στις 20 Δεκεμβρίου 2016 από: <https://github.com/adafruit/RTClib>

Τσουροπλής, Α. & Κλημόπουλος, Κ. (2005). *Εισαγωγή στην Πληροφορική*. Αθήνα: Εκδόσεις Νέων Τεχνολογιών.

Castro, E. (2003). *Εισαγωγή στην HTML για τον Παγκόσμιο Ιστό* (5<sup>η</sup> έκδ.). Αθήνα: Κλειδάριθμος

Elsmari, R. & Navathe, S.B. (2007). *Θεμελιώδεις Αρχές Συστημάτων Βάσεων Δεδομένων* (5<sup>η</sup> έκδ.). Τόμος Α'. Αθήνα: Εκδόσεις Δίαυλος



```
Serial.print(bmp.readAltitude());  
Serial.println(" meters");  
  
Serial.print("Pressure at sealevel (calculated) = ");  
Serial.print(bmp.readSealevelPressure());  
Serial.println(" Pa");  
  
// you can get a more precise measurement of altitude  
// if you know the current sea level pressure which will  
// vary with weather and such. If it is 1015 millibars  
// that is equal to 101500 Pascals.  
Serial.print("Real altitude = ");  
Serial.print(bmp.readAltitude(101500));  
Serial.println(" meters");  
  
Serial.println();  
delay(500);  
}
```

Για τον MPL115A2.

```
#include <Wire.h>  
#include <Adafruit_MPL115A2.h>  
  
Adafruit_MPL115A2 mpl115a2;  
  
void setup(void)  
{  
  Serial.begin(9600);  
  Serial.println("Hello!");  
  
  Serial.println("Getting barometric pressure ...");  
  mpl115a2.begin();  
}  
  
void loop(void)  
{  
  float pressureKPA = 0, temperatureC = 0;  
  
  mpl115a2.getPT(&pressureKPA,&temperatureC);  
  Serial.print("Pressure (kPa): "); Serial.print(pressureKPA, 4); Serial.print(" kPa ");  
  Serial.print("Temp (*C): "); Serial.print(temperatureC, 1); Serial.println(" *C both measured  
together");  
  
  pressureKPA = mpl115a2.getPressure();  
  Serial.print("Pressure (kPa): "); Serial.print(pressureKPA, 4); Serial.println(" kPa");  
  
  temperatureC = mpl115a2.getTemperature();  
  Serial.print("Temp (*C): "); Serial.print(temperatureC, 1); Serial.println(" *C");  
  
  delay(1000);  
}
```

Για τον dht11.

```
#include <dht11.h>
dht11 DHT;
#define DHT11_PIN 4

void setup(){
  Serial.begin(9600);
  Serial.println("DHT TEST PROGRAM ");
  Serial.print("LIBRARY VERSION: ");
  Serial.println(DHT11LIB_VERSION);
  Serial.println();
  Serial.println("Type,\tstatus,\tHumidity (%),\tTemperature (C)");
}

void loop(){
  int chk;
  Serial.print("DHT11, \t");
  chk = DHT.read(DHT11_PIN); // READ DATA
  switch (chk){
    case DHTLIB_OK:
      Serial.print("OK,\t");
      break;
    case DHTLIB_ERROR_CHECKSUM:
      Serial.print("Checksum error,\t");
      break;
    case DHTLIB_ERROR_TIMEOUT:
      Serial.print("Time out error,\t");
      break;
    default:
      Serial.print("Unknown error,\t");
      break;
  }
  // DISPLAT DATA
  Serial.print(DHT.humidity,1);
  Serial.print(",\t");
  Serial.println(DHT.temperature,1);

  delay(1000);
}
```

Για τους DS18B20 DS18B20-Waterproof.

```
#include <OneWire.h>
#include <DallasTemperature.h>

#define ONE_WIRE_BUS_1 2
#define ONE_WIRE_BUS_2 4

OneWire oneWire_in(ONE_WIRE_BUS_1);
OneWire oneWire_out(ONE_WIRE_BUS_2);

DallasTemperature sensor_inhouse(&oneWire_in);
DallasTemperature sensor_outhouse(&oneWire_out);
```

```
void setup(void)
{
  Serial.begin(9600);
  Serial.println("Dallas Temperature Control Library Demo - TwoPin_DS18B20");

  sensor_inhouse.begin();
  sensor_outhouse.begin();
}

void loop(void)
{
  Serial.print("Requesting temperatures...");
  sensor_inhouse.requestTemperatures();
  sensor_outhouse.requestTemperatures();
  Serial.println(" done");

  Serial.print("Inhouse: ");
  Serial.println(sensor_inhouse.getTempCByIndex(0));

  Serial.print("Outhouse: ");
  Serial.println(sensor_outhouse.getTempCByIndex(0));
}
```

Για τον RTC\_DS1307.

```
#include <Wire.h>
#include "RTClib.h"
```

```
RTC_DS1307 RTC;
```

```
void setup () {
  Serial.begin(57600);
  Wire.begin();
  RTC.begin();

  if (! RTC.isrunning()) {
    Serial.println("RTC is NOT running!");
    // following line sets the RTC to the date & time this sketch was compiled
    RTC.adjust(DateTime(__DATE__, __TIME__));
  }
}

void loop () {
  DateTime now = RTC.now();

  Serial.print(now.year(), DEC);
  Serial.print('/');
  Serial.print(now.month(), DEC);
  Serial.print('/');
  Serial.print(now.day(), DEC);
  Serial.print(' ');
  Serial.print(now.hour(), DEC);
  Serial.print(':');
  Serial.print(now.minute(), DEC);
```

```
Serial.print(':');
Serial.print(now.second(), DEC);
Serial.println();

Serial.print(" since midnight 1/1/1970 = ");
Serial.print(now.unixtime());
Serial.print("s = ");
Serial.print(now.unixtime() / 86400L);
Serial.println("d");

// calculate a date which is 7 days and 30 seconds into the future
DateTime future (now.unixtime() + 7 * 86400L + 30);

Serial.print(" now + 7d + 30s: ");
Serial.print(future.year(), DEC);
Serial.print('/');
Serial.print(future.month(), DEC);
Serial.print('/');
Serial.print(future.day(), DEC);
Serial.print(' ');
Serial.print(future.hour(), DEC);
Serial.print(':');
Serial.print(future.minute(), DEC);
Serial.print(':');
Serial.print(future.second(), DEC);
Serial.println();

Serial.println();
delay(3000);
}
```

Για τον WebServer.

```
#include <SPI.h>
#include <Ethernet.h>

// Enter a MAC address and IP address for your controller below.
// The IP address will be dependent on your local network:
byte mac[] = {
  0xDE, 0xAD, 0xBE, 0xEF, 0xFE, 0xED };
IPAddress ip(192,168,1,177);

// Initialize the Ethernet server library
// with the IP address and port you want to use
// (port 80 is default for HTTP):
EthernetServer server(80);

void setup() {
  // Open serial communications and wait for port to open:
  Serial.begin(9600);
  while (!Serial) {
    ; // wait for serial port to connect. Needed for Leonardo only
  }
}
```

```
// start the Ethernet connection and the server:
Ethernet.begin(mac, ip);
server.begin();
Serial.print("server is at ");
Serial.println(Ethernet.localIP());
}

void loop() {
  // listen for incoming clients
  EthernetClient client = server.available();
  if (client) {
    Serial.println("new client");
    // an http request ends with a blank line
    boolean currentLineIsBlank = true;
    while (client.connected()) {
      if (client.available()) {
        char c = client.read();
        Serial.write(c);
        // if you've gotten to the end of the line (received a newline
        // character) and the line is blank, the http request has ended,
        // so you can send a reply
        if (c == '\n' && currentLineIsBlank) {
          // send a standard http response header
          client.println("HTTP/1.1 200 OK");
          client.println("Content-Type: text/html");
          client.println("Connection: close"); // the connection will be closed after completion of the
response
          client.println("Refresh: 5"); // refresh the page automatically every 5 sec
          client.println();
          client.println("<!DOCTYPE HTML>");
          client.println("<html>");
          // output the value of each analog input pin
          for (int analogChannel = 0; analogChannel < 6; analogChannel++) {
            int sensorReading = analogRead(analogChannel);
            client.print("analog input ");
            client.print(analogChannel);
            client.print(" is ");
            client.print(sensorReading);
            client.println("<br />");
          }
          client.println("</html>");
          break;
        }
        if (c == '\n') {
          // you're starting a new line
          currentLineIsBlank = true;
        }
        else if (c != '\r') {
          // you've gotten a character on the current line
          currentLineIsBlank = false;
        }
      }
    }
  }
}
```

```
}  
}  
// give the web browser time to receive the data  
delay(1);  
// close the connection:  
client.stop();  
Serial.println("client disconnected");  
}  
}
```