

**ΤΕΧΝΟΛΟΓΙΚΟ ΕΚΠΑΙΔΕΥΤΙΚΟ ΙΔΡΥΜΑ ΔΥΤΙΚΗΣ ΕΛΛΑΔΑΣ**

**ΣΧΟΛΗ ΤΕΧΝΟΛΟΓΙΚΩΝ ΕΦΑΡΜΟΓΩΝ**

**ΤΜΗΜΑ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ ΤΕ**

**ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ**

**ΑΝΑΛΥΣΗ ΜΕΤΕΩΡΟΛΟΓΙΚΩΝ ΜΕΤΡΗΣΕΩΝ  
ΜΕΣΩ ΑΙΣΘΗΤΗΡΩΝ ΜΕ ΤΗ ΧΡΗΣΗ ΤΟΥ  
ΜΙΚΡΟΕΛΕΓΚΤΗ ARDUINO**

**1539**

**ΔΑΡΜΑ ΣΟΦΙΑ**

**ΕΠΟΠΤΕΥΩΝ ΚΑΘΗΓΗΤΗΣ : ΚΩΝΣΤΑΝΤΟΠΟΥΛΟΥ ΑΛΕΞΑΝΔΡΑ**

**ΠΑΤΡΑ 2015**



---

## ΠΕΡΙΛΗΨΗ

Η παρούσα πτυχιακή εργασία αφορά τη δημιουργία ενός συστήματος συλλογής, επεξεργασίας και αποθήκευσης μετεωρολογικών δεδομένων, με τη βοήθεια του κατάλληλου υλικού και λογισμικού.

Αρχικά γίνεται μια παρουσίαση της υπολογιστικής πλατφόρμας Arduino και των δυνατοτήτων της. Αναφέρονται τα είδη Arduino και τα χαρακτηριστικά τους και παρουσιάζονται τα βήματα για τον προγραμματισμό της πλατφόρμας.

Ακολουθεί η περιγραφή της πειραματικής διαδικασίας και των υλικών τα οποία χρησιμοποιήθηκαν για την υλοποίηση του συστήματος. Γίνεται αναλυτική περιγραφή των αισθητήρων που συνδέθηκαν στον μικροελεγκτή, της οθόνης LCD καθώς επίσης και της συνδεσμολογίας που πραγματοποιήθηκε. Οι αισθητήρες που χρησιμοποιήθηκαν ήταν για μέτρηση θερμοκρασίας, σχετικής υγρασίας, βαρομετρικής πίεσης, έντασης φωτισμού, ύψους βροχής και υγρασίας εδάφους. Για την ορθή παρουσίαση του πειραματικού μέρους της εργασίας κατασκευάστηκε μακέτα.

Μετά τη συλλογή των δεδομένων ακολουθεί η επεξεργασία τους. Στο κεφάλαιο αυτό δίνονται οι απαραίτητοι ορισμοί των μετεωρολογικών μεγεθών που μετρήθηκαν και η διαδικασία βαθμονόμησης των αισθητήρων για την οποία έγινε χρήση του προγράμματος ORIGIN. Στη συνέχεια περιγράφονται αναλυτικά τα βήματα για την αποστολή και αποθήκευση των δεδομένων που κατέγραψαν οι αισθητήρες σε αρχείο Excel.

Τέλος, προτείνονται δύο δυνατές εφαρμογές του συστήματος. Η πρώτη αφορά στη βελτίωση των μεθόδων καλλιέργειας και η δεύτερη στην πρόγνωση του καιρού με τη βοήθεια του αισθητήρα της βαρομετρικής πίεσης. Η εργασία ολοκληρώνεται με τα συμπεράσματα τα οποία προκύπτουν και πιθανές βελτιώσεις που θα μπορούσαν να γίνουν.



---

## ΠΕΡΙΕΧΟΜΕΝΑ

Περίληψη.....	Σελ.iii
Περιεχόμενα.....	Σελ.v
Εισαγωγή.....	Σελ.vii
<b>Κεφάλαιο 1<sup>ο</sup> ΤΟ ARDUINO.....</b>	<b>Σελ.1</b>
<b>1.1. Η πλατφόρμα Arduino.....</b>	<b>Σελ.1</b>
<b>1.1.1. Κύρια μέρη μικροελεγκτή Arduino.....</b>	<b>Σελ.2</b>
<b>1.1.2. Βασικές μνήμες.....</b>	<b>Σελ.3</b>
<b>1.1.3. Θύρες επικοινωνίας.....</b>	<b>Σελ.3</b>
<b>1.1.4. Πλεονεκτήματα Arduino.....</b>	<b>Σελ.4</b>
<b>1.2. Είδη Arduino.....</b>	<b>Σελ.5</b>
<b>1.3. Προγραμματισμός Arduino.....</b>	<b>Σελ.12</b>
<b>1.3.1. Εγκατάσταση προγράμματος.....</b>	<b>Σελ.12</b>
<b>1.3.2. Προγράμματα Arduino.....</b>	<b>Σελ.12</b>
<b>1.3.3. Βασικές δομές προγραμματισμού.....</b>	<b>Σελ.14</b>
<b>1.3.4. Ολοκληρωμένο περιβάλλον ανάπτυξης Arduino.....</b>	<b>Σελ.16</b>
<b>1.3.5. Ανέβασμα σχεδίου.....</b>	<b>Σελ.19</b>
<b>1.3.6. Βιβλιοθήκες.....</b>	<b>Σελ.22</b>
<b>1.4. Επικοινωνία με το διαδίκτυο μέσω Ethernet ή WiFi Shield.....</b>	<b>Σελ.26</b>
<b>1.4.1. Το Ethernet Shield.....</b>	<b>Σελ.26</b>
<b>1.4.2. Το WiFi Shield.....</b>	<b>Σελ.27</b>
<b>Κεφάλαιο 2<sup>ο</sup> ΠΕΙΡΑΜΑΤΙΚΟ ΜΕΡΟΣ.....</b>	<b>Σελ.28</b>
<b>2.1. Υλικά σύνδεσης.....</b>	<b>Σελ.28</b>
<b>2.2. Αισθητήρες.....</b>	<b>Σελ.31</b>
<b>2.3. LCD οθόνη.....</b>	<b>Σελ.36</b>
<b>2.4. Η μακέτα του συστήματος.....</b>	<b>Σελ.37</b>
<b>Κεφάλαιο 3<sup>ο</sup> ΕΠΕΞΕΡΓΑΣΙΑ ΜΕΤΡΗΣΕΩΝ.....</b>	<b>Σελ.38</b>
<b>3.1. Αισθητήρες.....</b>	<b>Σελ.38</b>
<b>3.2. ORIGIN.....</b>	<b>Σελ.40</b>
<b>3.3. Αποστολή δεδομένων σε αρχείο Excel.....</b>	<b>Σελ.45</b>
<b>Κεφάλαιο 4<sup>ο</sup> ΕΦΑΡΜΟΓΕΣ - ΣΥΜΠΕΡΑΣΜΑΤΑ.....</b>	<b>Σελ.57</b>
<b>4.1. Εφαρμογές στη γεωργία.....</b>	<b>Σελ.57</b>
<b>4.2. Πρόγνωση καιρού.....</b>	<b>Σελ.59</b>
<b>4.3. Συμπεράσματα.....</b>	<b>Σελ.63</b>
<b>Παράρτημα.....</b>	<b>Σελ.64</b>
<b>Βιβλιογραφία.....</b>	<b>Σελ.73</b>



## ΕΙΣΑΓΩΓΗ



Το ενδιαφέρον του ανθρώπου για την πρόγνωση του καιρού ξεκινά από πολύ παλιά. Οι αρχαίοι Έλληνες από τον 5<sup>ο</sup> αιώνα π.Χ. παρατηρούσαν τα διάφορα φαινόμενα προσπαθώντας να τα ερμηνεύουν και να τα κωδικοποιούν. Τα “Μετεωρολογικά” του Αριστοτέλη (350 π.Χ.) αποτέλεσαν το πρότυπο διδακτικό βιβλίο της Μετεωρολογίας για περισσότερο από δύο χιλιάδες χρόνια. Η συνέχεια δόθηκε από το Θεόφραστο, μαθητή του Αριστοτέλη, που με το βιβλίο των “Σημείων” συνδύασε επιστήμη και παράδοση, δίνοντας κανόνες για την πρόγνωση του καιρού. Τα έργα αυτά αποτελούν την απαρχή της επιστημονικής συστηματοποίησης των μετεωρολογικών παρατηρήσεων, επιτυγχάνοντας την επιστημονική ερμηνεία των καιρικών φαινομένων.

Η Μετεωρολογία[1] ανήκει στις Θετικές επιστήμες, με κύριο αντικείμενο την έρευνα της ατμόσφαιρας στο σύνολο της και των φαινομένων που συμβαίνουν σ’ αυτή. Τα σημαντικότερα αυτών των φαινομένων είναι η ατμοσφαιρική πίεση, οι μεταβολές της θερμοκρασίας και η υγρασία. Τα στοιχεία αυτά αποτελούν τις κύριες αιτίες κάθε ατμοσφαιρικής διατάραξης, αποτελέσματα της οποίας είναι τα μετεωρολογικά φαινόμενα.

Στην παρούσα πτυχιακή γίνεται καταγραφή των στοιχείων αυτών με τη χρήση της πλατφόρμας Arduino και κατάλληλων αισθητήρων, με σκοπό την ανάλυση και αποθήκευση τους για μελλοντική αξιοποίηση.





# ΚΕΦΑΛΑΙΟ 1<sup>ο</sup>

## ΤΟ ARDUINO

### 1.1 Η ΠΛΑΤΦΟΡΜΑ ARDUINO



*Εικ.1 Η πλατφόρμα Arduino*

Το Arduino[2] είναι ένας single-board μικροελεγκτής, δηλαδή μια απλή μητρική πλακέτα ανοικτού κώδικα, με ενσωματωμένο μικροελεγκτή και εισόδους/εξόδους, και η οποία μπορεί να προγραμματιστεί με τη γλώσσα Wiring (ουσιαστικά πρόκειται για τη γλώσσα προγραμματισμού C++ και ένα σύνολο από βιβλιοθήκες, υλοποιημένες επίσης στην C++). Το Arduino μπορεί να χρησιμοποιηθεί για την ανάπτυξη ανεξάρτητων διαδραστικών αντικειμένων αλλά και να συνδεθεί με υπολογιστή μέσω προγραμμάτων σε Processing, Max/MSP, Pure Data, SuperCollider. Οι περισσότερες εκδόσεις του Arduino μπορούν να αγοραστούν προ-συναρμολογημένες. Το διάγραμμα και πληροφορίες για το υλικό είναι ελεύθερα διαθέσιμα για αυτούς που θέλουν να συναρμολογήσουν το Arduino μόνοι τους. Κατασκευάζεται κυρίως από την εταιρεία Smart Project.

Το σχέδιο Arduino είναι μία διακλάδωση της πλατφόρμας Wiring για λογισμικό ανοικτού κώδικα και προγραμματίζεται χρησιμοποιώντας μια γλώσσα βασισμένη στο Wiring (σύνταξη και βιβλιοθήκες), παρόμοια με την C++ με απλοποιήσεις και αλλαγές, καθώς και ένα ολοκληρωμένο περιβάλλον ανάπτυξης (IDE).

Μία πλακέτα Arduino αποτελείται από ένα μικροελεγκτή Atmel AVR (ATmega328 και ATmega168 στις νεότερες εκδόσεις, ATmega8 στις παλαιότερες) και συμπληρωματικά εξαρτήματα για την διευκόλυνση του χρήστη στον προγραμματισμό και την ενσωμάτωση του σε άλλα κυκλώματα. Ο AVR διαθέτει μια τροποποιημένη Harvard αρχιτεκτονική, η οποία αναπτύχθηκε από την Atmel το 1996. Ο AVR ήταν ένας από τους πρώτους μικροελεγκτές που χρησιμοποίησε on-chip flash memory για αποθήκευση του προγράμματος, σε αντίθεση με την εφάπαξ προγραμματιζόμενη μνήμη ROM, EPROM, EEPROM που χρησιμοποιούσαν οι άλλοι μικροελεγκτές μέχρι τότε. Το πρόγραμμα και τα δεδομένα αποθηκεύονται σε ξεχωριστά συστήματα μνήμης τα οποία εμφανίζονται σε διαφορετικούς χώρους διεύθυνσης, αλλά υπάρχει η δυνατότητα ανάγνωσης των δεδομένων από τη μνήμη του προγράμματος χρησιμοποιώντας συγκεκριμένες οδηγίες. Ο μικροελεγκτής που χρησιμοποιήθηκε ανήκει στην οικογένεια των megaAVR και διαθέτει μνήμη 256 KB. Όλες οι πλακέτες περιλαμβάνουν ένα γραμμικό ρυθμιστή τάσης 5V και έναν κρυσταλλικό ταλαντωτή 16MHz (ή κεραμικό αντηχητή σε κάποιες παραλλαγές). Ο μικροελεγκτής είναι από κατασκευής προγραμματισμένος με ένα φορτωτή εκκίνησης (bootloader), έτσι ώστε να μην χρειάζεται εξωτερικός προγραμματιστής και συνδέεται μέσω USB θύρας με τον Η/Υ.

Οι χρήστες του μικροελεγκτή καλούνται να ορίσουν δύο λειτουργίες για να κάνουν ένα πρόγραμμα κυκλικής εκτέλεσης :

- setup() : μία συνάρτηση που τρέχει μία φορά στην αρχή του προγράμματος η οποία αρχικοποιεί τις ρυθμίσεις
- loop() : μία συνάρτηση η οποία καλείται συνέχεια μέχρι η πλακέτα να απενεργοποιηθεί

### **1.1.1 Κύρια μέρη μικροελεγκτή Arduino**

Ένας μικροελεγκτής Arduino αποτελείται από αρκετά μέρη. Τα κυριότερα από αυτά είναι:

- Μικροεπεξεργαστής
- Ψηφιακές είσοδοι/έξοδοι
- Αναλογικές είσοδοι
- Εξωτερική τροφοδοσία
- Θύρα USB
- Τάσεις τροφοδοσίας 3.3V, 5V, GND
- Reset button

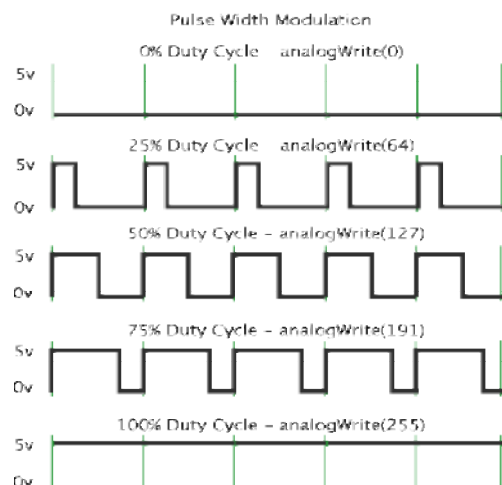
### 1.1.2 Βασικές μνήμες

Οι βασικές μνήμες που υπάρχουν στο Arduino είναι οι ακόλουθες :

- Flash memory : Σε αυτή τοποθετείται το πρόγραμμα προς εκτέλεση. Οι πληροφορίες παραμένουν και μετά την απενεργοποίηση του συστήματος.
- SRAM memory : Χρησιμοποιείται για την προσωρινή αποθήκευση των δεδομένων του προγράμματος που θα εκτελεστεί. Οι πληροφορίες χάνονται με την απενεργοποίηση του συστήματος.
- EEPROM memory : Σε αυτή αποθηκεύονται οι τιμές των μεταβλητών όταν σβήσει η πλατφόρμα. Και σε αυτή τη μνήμη οι πληροφορίες παραμένουν στο σύστημα μετά την απενεργοποίηση.

### 1.1.3 Θύρες επικοινωνίας

Η επικοινωνία του Arduino πραγματοποιείται μέσω των ψηφιακών και αναλογικών εισόδων/εξόδων. Το Arduino Mega 2560 διαθέτει 16 αναλογικές εισόδους (analog input) και 54 ψηφιακές εισόδους/εξόδους (digital input/output). Οι αναλογικές εισοδοί συμβολίζονται ως A0-A15, ενώ οι ψηφιακές με αριθμούς από το 0 έως το 53. Οι ψηφιακές θύρες 2-13 και 44 και 46 είναι PWM. Ως PWM[3] ή Pulse Width Modulation (Διαμόρφωση εύρους παλμών) ορίζεται η τεχνική η οποία δίνει αναλογικά αποτελέσματα με τη χρήση ψηφιακών μέσων. Ο ψηφιακός έλεγχος χρησιμοποιείται για τη δημιουργία ενός τετραγωνικού κύματος, δηλαδή ενός σήματος που εναλλάσσεται μεταξύ on και off ( ή αλλιώς μεταξύ 5V και 0V). Η διάρκεια του "on time" είναι το pulse width, δηλαδή το εύρος του παλμού (ή αλλιώς duty cycle).



Εικ. 2 Η μορφή της εξόδου PWM

#### 1.1.4 Πλεονεκτήματα Arduino

Οι εφαρμογές που υλοποιούνται με τη χρήση του Arduino είναι πολλές, γεγονός που οφείλεται σε συγκεκριμένα χαρακτηριστικά της πλατφόρμας.

- Οικονομική : Έχει αρκετά χαμηλό κόστος σε σχέση με άλλες πλατφόρμες και είναι αρχιτεκτονικά ανοιχτή, γεγονός που επιτρέπει στο χρήστη να την αναπτύξει μόνος του.
- Μεταφέρσιμη : Σε σύγκριση με αντίστοιχα προϊόντα μπορεί να μεταφερθεί οπουδήποτε και να προγραμματιστεί στα περισσότερα λειτουργικά συστήματα.
- Επεκτάσιμη : Το λογισμικό της είναι ανοιχτό σε όλους, γεγονός που διευκολύνει την ανάπτυξη και εξέλιξη της. Η επέκταση της γλώσσας προγραμματισμού μπορεί να γίνει μέσω μετατροπών των διαθέσιμων βιβλιοθηκών ή με την δημιουργία νέων.
- Συμβατή με πολλά λειτουργικά συστήματα : Το περιβάλλον προγραμματισμού της πλατφόρμας Arduino είναι συμβατό με τα λειτουργικά συστήματα Windows, Macintosh OS X και Linux.
- Απλό περιβάλλον προγραμματισμού : Το περιβάλλον προγραμματισμού της πλατφόρμας Arduino είναι αρκετά απλό και ενδείκνυται για νέους χρήστες, αλλά παράλληλα δίνει πολλές δυνατότητες στους πιο προχωρημένους.

## 1.2 ΕΙΔΗ ARDUINO

### 1.UNO

Το Arduino Uno βασίζεται στην τεχνολογία ATmega328. Έχει 14 pins ψηφιακής εισόδου/εξόδου και 6 αναλογικές εισόδους. Είναι η πλακέτα που εμφανίζεται με τη μεγαλύτερη συχνότητα, καθώς χρησιμοποιείται στις περισσότερες εφαρμογές.



*Εικ.3 Το Arduino Uno*

### 2.LEONARDO

Το Arduino Leonardo βασίζεται στην τεχνολογία ATmega32u4. Έχει 20 pins ψηφιακής εισόδου/εξόδου (εκ των οποίων τα 7 μπορούν να χρησιμοποιηθούν ως PWM έξοδοι) και 12 αναλογικές εισόδους.



*Εικ.4 Το Arduino Leonardo*

### 3.DUE

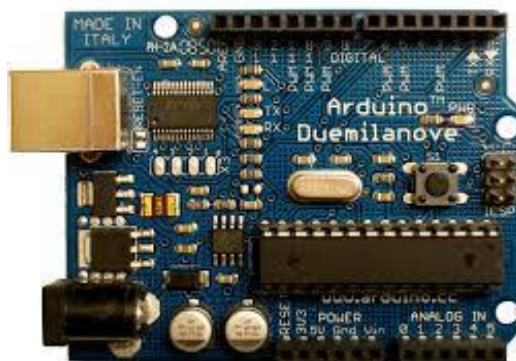
Το Arduino Due είναι βασισμένο στην τεχνολογία Atmel SAM3X8E ARM Cortex-M3 CPU και είναι η πρώτη πλακέτα που έχει βασιστεί σε 32-bit ARM core microcontroller. Σε αντίθεση με τα περισσότερα Arduino, το Due λειτουργεί στα 3,3V.



*Εικ.5 Το Arduino DUE*

### 4.DUEMILANOVE

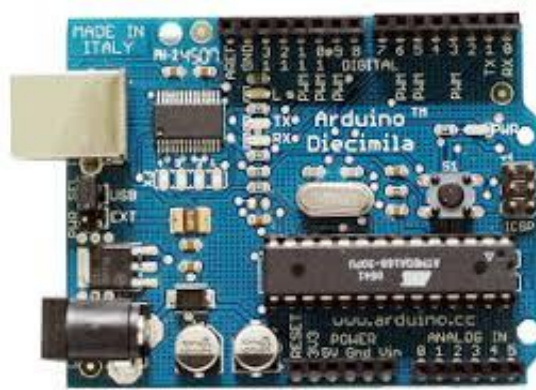
Το Arduino Duemilanove βασίζεται στην τεχνολογία ATmega168 (και στην ATmega328 στη νεότερη έκδοση). Αποτελείται από 14 pin ψηφιακής εισόδου/εξόδου (εκ των οποίων τα 6 μπορούν να χρησιμοποιηθούν ως PWM εξοδοί) και 6 αναλογικές εισόδους.



*Εικ.6 Το Arduino Duemilanove*

## 5.DIECIMILA

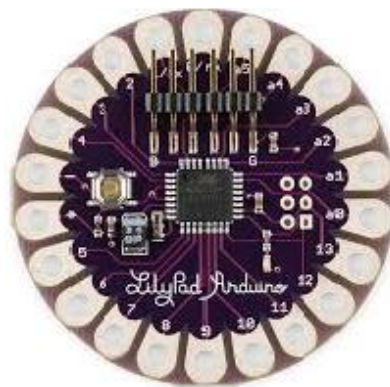
Το Arduino Diecimila είναι βασισμένο στην τεχνολογία ATmega168 και αποτελείται από 14 pins ψηφιακής εισόδου/εξόδου (εκ των οποίων τα 6 μπορούν να χρησιμοποιηθούν ως PWM έξοδοι) και 6 pins αναλογικής εισόδου.



*Εικ.7 Το Arduino Diecimila*

## 6.LILYPAD

Το Lilypad Arduino βασίζεται στην τεχνολογία ATmega168 ή ATmega328 και έχει αρκετές σχεδιαστικές διαφορές, σε σχέση με άλλες πλακέτες. Χρησιμοποιείται σε εφαρμογές ένδυσης και E-textiles.



*Εικ.8 Το Arduino Lilypad*

## 7.ESPLORA

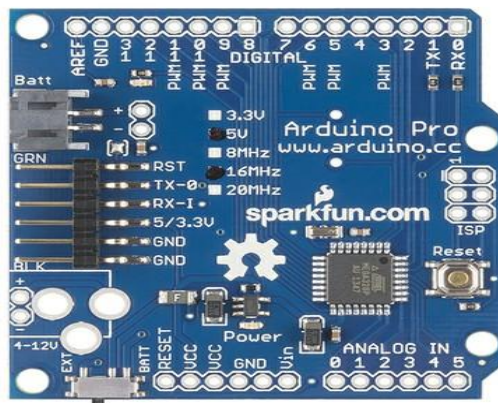
Το Arduino Esplora προέκυψε από το Leonardo. Χρησιμοποιεί την τεχνολογία ATmega32u4. Διαφοροποιείται από τα περισσότερα είδη Arduino καθώς περιλαμβάνει ένα σετ αισθητήρων έτοιμο για αλληλεπίδραση. Η μορφή του παραπέμπει σε χειριστήριο βιντεοπαιχνιδιών.



*Εικ.9 Το Arduino Esplora*

## 8.PRO

Το Arduino Pro είναι βασισμένο στην τεχνολογία ATmega168 ή ATmega328. Κυκλοφορεί σε δύο εκδοχές, μία στα 3,3V/8MHz και στα 5V/16MHz. Περιέχει 14 pins ψηφιακής εισόδου/εξόδου και 6 pins αναλογικής εισόδου.



*Εικ.10 Το Arduino Pro*



## 9.NANO

Το Arduino Nano είναι μια μικρή,πλήρης πλακέτα βασισμένη στις τεχνολογίες ATmega 328 ή ATmega168. Έχει περίπου την ίδια λειτουργικότητα με το Arduino Duemilanove και για τη σύνδεση του απαιτείται Mini-B USB καλώδιο.



*Εικ.11 Το Arduino Nano*

## 10.BLUETOOTH

Το Arduino Bluetooth χρησιμοποιεί τεχνολογία ATmega168 και παρέχει την δυνατότητα για Bluetooth interface.



*Εικ.12 Το Arduino Bluetooth*

## 11.ARDUINO MEGA

Το Arduino Mega χρησιμοποιεί τεχνολογία surface-mounted ATmega1280. Έχει 54 pins ψηφιακής εισόδου/εξόδου και 16 αναλογικές εισόδους και παρέχει περισσότερη μνήμη από άλλες πλακέτες (όπως για παράδειγμα το Uno).



*Εικ.13 Το Arduino Mega*

## 12.PRO MINI

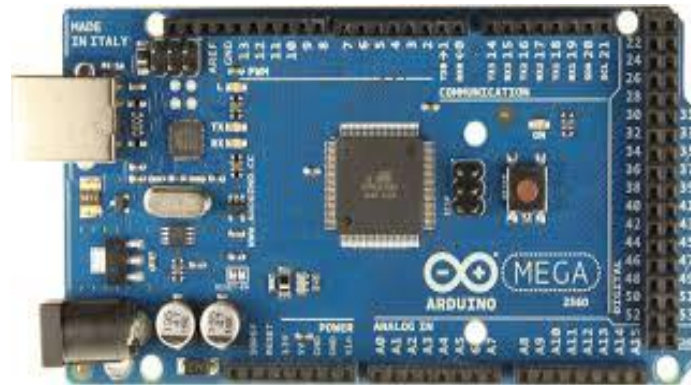
Το Arduino Pro Mini βασίζεται στην τεχνολογία ATmega328, έχει 14 pins ψηφιακής εισόδου/εξόδου και 6 αναλογικής εισόδου.



*Εικ.14 Το Arduino Pro Mini*

### 13.ARDUINO MEGA 2560

Το Arduino Mega 2560 είναι η νεότερη έκδοση του Mega, το οποίο και αντικατέστησε. Βασίζεται στην τεχνολογία ATmega2560, με μνήμη Flash στα 256KB και είναι σχεδιασμένο για πιο σύνθετα projects. Η συγκεκριμένη πλακέτα χρησιμοποιήθηκε στην παρούσα εργασία καθώς υπήρχε ανάγκη για μεγάλο αριθμό εισόδων, λόγω του πλήθους των αισθητήρων.



Εικ.15 Το Arduino Mega 2560

#### ΧΑΡΑΚΤΗΡΙΣΤΙΚΑ ΤΟΥ ΜΙΚΡΟΕΛΕΓΚΤΗ ΑΤΜΕΓΑ2560

Μικροελεγκτής	ATmega2560
Τάση λειτουργίας	5V
Τάση εισόδου (προτεινόμενη)	7-12V
Τάση εισόδου (οριακή)	6-20V
Ψηφιακές I/O	54 (εκ των οποίων 15 παρέχουν PWM έξοδο)
Αναλογικές εισοδοι	16
DC ρεύμα ανά I/O Pin	20 mA
DC ρεύμα για 3.3V Pin	50 mA
Μνήμη Flash	256 KB εκ των οποίων 8 KB χρησιμοποιούνται από ένα bootloader
Μνήμη SRAM	8 KB
Μνήμη EEPROM	4 KB
Ταχύτητα ρολογιού	16 MHz
Μήκος	101.52 mm
Πλάτος	53.3 mm
Βάρος	37 g

## 1.3 ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΣ ΤΗΣ ΠΛΑΤΦΟΡΜΑΣ ARDUINO

### 1.3.1 Εγκατάσταση προγράμματος

Το πρώτο βήμα που απαιτείται για τον προγραμματισμό του Arduino είναι η εγκατάσταση του προγράμματος Arduino IDE. Στο περιβάλλον του προγράμματος θα τοποθετηθεί ο κατάλληλος κώδικας, ο οποίος στη συνέχεια θα φορτωθεί στην πλατφόρμα. Οι διαθέσιμες εκδόσεις του προγράμματος είναι για Windows, Linux και Macintosh OSX και προσφέρονται δωρεάν στο site της εταιρείας και συγκεκριμένα στο link <https://www.arduino.cc/en/Main/Software>

### 1.3.2 Προγράμματα Arduino

Σε ένα ολοκληρωμένο πρόγραμμα Arduino διακρίνονται τα παρακάτω :

- Ø Μεταβλητές (values)
- Ø Συναρτήσεις (functions)
- Ø Δομή (structure)

#### Μεταβλητές (Values)

Οι μεταβλητές είναι ένας χώρος που αποθηκεύονται τα δεδομένα. Έχουν όνομα, τύπο και τιμή.

#### Συναρτήσεις (Functions)

Μια συνάρτηση είναι ένα κομμάτι κώδικα που του έχει δοθεί ένα όνομα και μπορεί να χρησιμοποιηθεί από οπουδήποτε μέσα στο σχεδιάγραμμα. Μερικές από τις κυριότερες είναι :

1. Συνάρτηση pinMode(pin,mode)

Η συνάρτηση pinMode χρησιμοποιείται στο μέρος του setup ώστε να καθορίσει ένα ορισμένο ακροδέκτη να συμπεριφέρεται ως είσοδος ή έξοδος.

2. Συνάρτηση digitalWrite(pin)

Διαβάζει την τιμή από ένα καθορισμένο ψηφιακό ακροδέκτη με το αποτέλεσμα να είναι είτε HIGH είτε LOW. Ο ακροδέκτης καθορίζεται είτε ως μεταβλητή είτε ως σταθερά.

3. Συνάρτηση digitalWrite(pin,value)

Βγάζει στην έξοδο του ακροδέκτη λογική τιμή HIGH ή LOW.

#### 4. Συνάρτηση analogRead(pin)

Διαβάζει την τιμή από ένα καθορισμένο αναλογικό ακροδέκτη, ο οποίος έχει συνδεθεί στο συγκεκριμένο pin. Οι αναλογικές εισοδοί του Arduino Mega είναι 16, από A0-A15.

#### 5. Συνάρτηση analogWrite(pin,value)

Γράφει μια ψευδο-αναλογική τιμή χρησιμοποιώντας PWM σε ένα ακροδέκτη εξόδου.

#### 6. Συνάρτηση delay(ms)

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

#### 7. Συνάρτηση Serial.begin(rate)

Ανοίγει μια σειριακή θύρα και θέτει το ρυθμό στη σειριακή μετάδοση δεδομένων.

#### 8. Συνάρτηση Serial.println(data)

Τυπώνει δεδομένα στη σειριακή θύρα που ολοκληρώνονται με τις ειδικές λειτουργίες του carriage return και line feed. Είναι η πιο κατάλληλη συνάρτηση για διάβασμα δεδομένων στο Serial Monitor για έλεγχο.

#### 9. Συναρτήσεις μετατροπής

int : ακέραιος με δυνατές τιμές από -32768 έως 32767

long : ακέραιος με δυνατές τιμές από -2147483648 έως 2147483647

float : δεκαδικός αριθμός

char : χαρακτήρας ενός byte

string : πίνακας χαρακτήρων

byte : με τιμές από 0-255

boolean : με τιμές 0 ή 1

### Δομή Προγράμματος (Structure)

Οι κύριοι βρόχοι που χρησιμοποιούνται για τον προγραμματισμό της πλατφόρμας Arduino είναι οι εξής :

#### **Void setup()**

Ο βρόχος αυτός εκτελείται μόνο μια φορά κατά την εκκίνηση του μικροελεγκτή. Σε αυτόν εισάγονται οι εντολές που θα εκτελεστούν μια φορά όπως οι αρχικοποιήσεις τιμών μεταβλητών ή ο χαρακτηρισμός των εισόδων/εξόδων που θα χρησιμοποιηθούν. Πρέπει να περιλαμβάνεται σε κάθε πρόγραμμα ακόμα και αν δεν υπάρχουν εντολές για εκτέλεση του.

## Void loop()

Αφού δημιουργηθεί ο βρόχος void setup(), σειρά έχει ο βρόχος void loop() ο οποίος εκτελείται συνεχώς και στον οποίο εισάγεται το κυρίως πρόγραμμα. Συνήθως περιλαμβάνει εντολές για είσοδο και έξοδο σημάτων από και προς το Arduino.

Η συνήθης δομή του κώδικα προγραμματισμού της πλατφόρμας Arduino έχει τη μορφή:

```
//δήλωση μεταβλητών
Void setup()
{
  //αρχικοποίηση μεταβλητών
}
Void loop()
{
  //κώδικας
}
```

Το μέρος του setup είναι το προπαρασκευαστικό στάδιο ενώ το loop είναι το μέρος της εκτέλεσης. Και τα δύο μέρη είναι απαιτούμενα για τη λειτουργία του προγράμματος. Το μέρος του setup πρέπει να δηλώνει τις μεταβλητές στην αρχή του προγράμματος και είναι η πρώτη συνάρτηση που εκτελείται στο πρόγραμμα. Αυτή η συνάρτηση εκτελείται μία φορά και χρησιμοποιείται για να θέσει την κατάσταση των ακροδεκτών στο arduino ή να αρχικοποιήσει θύρες επικοινωνίας.

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

### 1.3.3 Βασικές δομές προγραμματισμού

#### Τελεστές

Αριθμητικοί τελεστές

- = (τελεστής εκχώρησης)
- + (τελεστής πρόσθεσης)
- - (τελεστής αφαίρεσης)
- \* (τελεστής πολλαπλασιασμού)
- / (τελεστής διαίρεσης)
- % (τελεστής υπόλοιπου ακέραιας διαίρεσης)

Τελεστές αύξησης/μείωσης

- ++ (αύξηση κατά 1 ακέραια μονάδα)
- -- (μείωση κατά 1 ακέραια μονάδα)

### Τελεστές σύγκρισης

- == (ισότητα)
- != (ανισότητα)
- > (μεγαλύτερο)
- >= (μεγαλύτερο ή ίσο)
- < (μικρότερο)
- <= (μικρότερο ή ίσο)

### Λογικοί τελεστές

- ! (λογική άρνηση)
- && (λογική σύζευξη)
- || (λογική διάζευξη)

### Δυαδικοί τελεστές

- & (δυαδική σύζευξη)
- | (δυαδική διάζευξη)
- ~ (δυαδική άρνηση)
- ^ (δυαδική αποκλειστική διάζευξη)
- » (δυαδική δεξιά ολίσθηση)
- « (δυαδική αριστερή ολίσθηση)

### Σταθερές

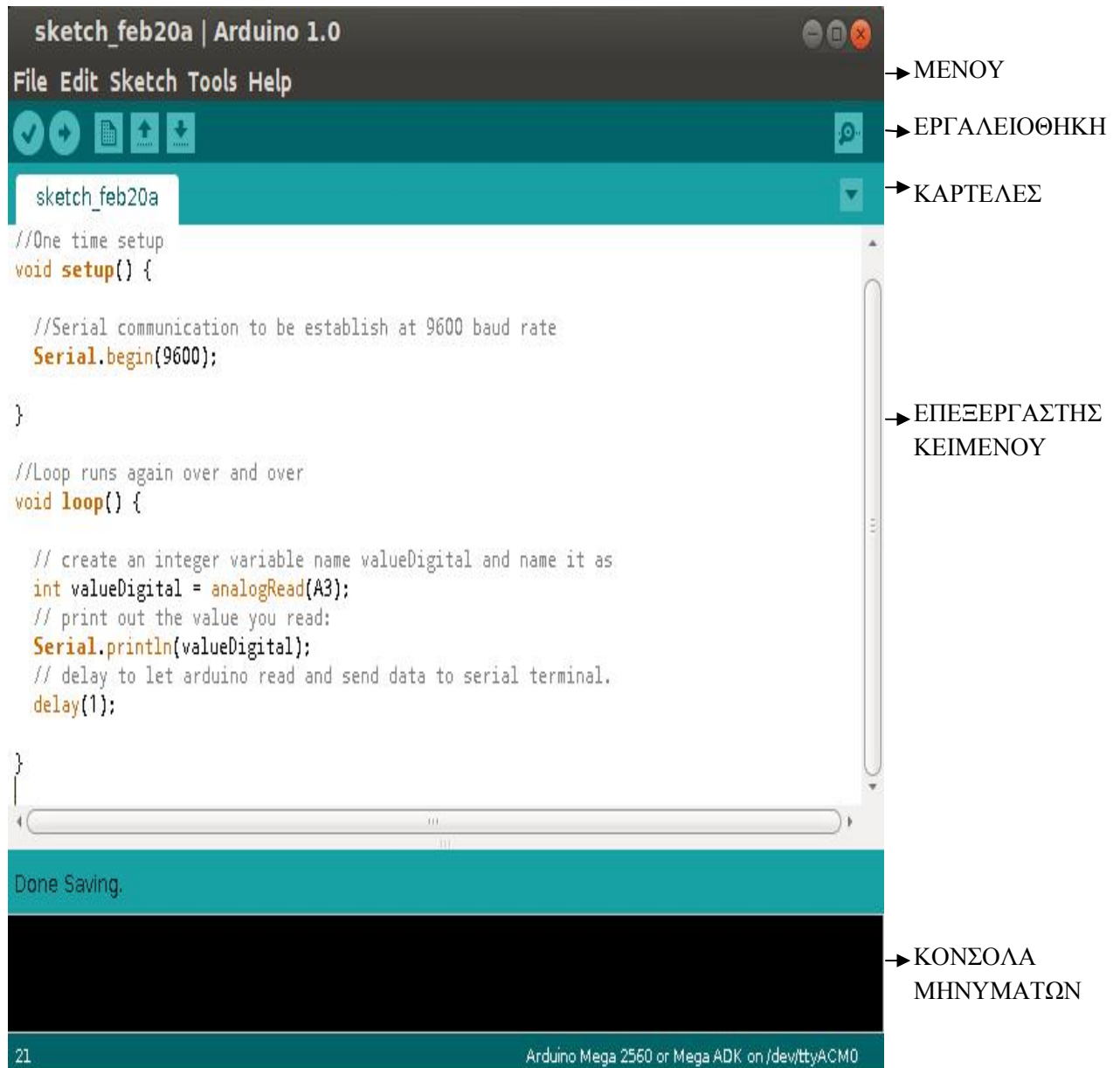
- HIGH (τιμή υψηλής στάθμης για μια επαφή εισόδου/εξόδου)
- LOW (τιμή χαμηλής στάθμης για μια επαφή εισόδου/εξόδου)
- false (λογικό επίπεδο ψεύδους μιας συνθήκης)
- true (λογικό επίπεδο αλήθειας μιας συνθήκης)
- INPUT (ορισμός επαφής ως είσοδος)
- OUTPUT (ορισμός επαφής ως έξοδος)
- A0-A15 (συμβολισμός αναλογικών επαφών εισόδου)

### Δομές ελέγχου ροής

- If (δομή ελέγχου συνθήκης)
- if...else (δομή ελέγχου πολλαπλών συνθηκών)
- for (δομή επαναληπτικού ελέγχου συνθήκης)
- while (δομή επαναληπτικού ελέγχου συνθήκης)
- do...while (δομή επαναληπτικού ελέγχου συνθήκης)
- goto (εντολή μετάβασης σε κάποιο σημείο του κώδικα)
- continue (εντολή παράλειψης της τρέχουσας επανάληψης)
- return (εντολή επιστροφής από μία συνάρτηση)
- break (εντολή διακοπής μιας επαναληπτικής δομής)
- switch...case (δομή ελέγχου περιπτώσεων)

### 1.3.4 Ολοκληρωμένο περιβάλλον ανάπτυξης Arduino

Το σχεδιάγραμμα (sketch) είναι το όνομα που χρησιμοποιεί το arduino για το εκάστοτε πρόγραμμα. Είναι η μονάδα κώδικα που φορτώνεται και εκτελείται στην πλακέτα arduino.



Εικ.16 Περιβάλλον ανάπτυξης του κώδικα Arduino IDE (Sketch)



## Εργαλειοθήκη

Στην εργαλειοθήκη του Sketch (κάτω από τη γραμμή μενού) βρίσκονται οι παρακάτω εντολές :



Verify/Compile (Έλεγχος/Μεταγλώττιση) : Έλεγχος για λάθη στον κώδικα



Upload (Ανέβασμα) : Ανέβασμα του κώδικα στον μικροελεγκτή



New (Νέο) : Δημιουργεί νέο sketch



Open (Άνοιγμα) : Παρουσιάζει ένα μενού με όλα τα sketch,κάνοντας κλικ σε ένα από αυτά θα ανοίξει μέσα στο τρέχον παράθυρο



Save (Αποθήκευση) : Αποθηκεύει το sketch




Serial monitor (σειριακή οθόνη) : Ανοίγει την σειριακή οθόνη ώστε να μπορούμε να δώσουμε δεδομένα από το πληκτρολόγιο



### 1.3.5 Ανέβασμα σχεδίου

Μετά την επιλογή του κατάλληλου κώδικα, ακολουθεί ο έλεγχος για συντακτικά λάθη. Αυτό γίνεται με την εντολή verify που βρίσκεται στην εργαλειοθήκη του sketch. Σε περίπτωση συντακτικού λάθους εμφανίζεται αντίστοιχο μήνυμα στην κονσόλα μηνυμάτων.



```
Classic_Blink_LED | Arduino 1.0.5-r2
File Edit Sketch Tools Help
Verify
Classic_Blink_LED $
const int LED = 13;

void setup()
{
  pinMode(LED, OUTPUT);
}

void loop()
{
  digitalWrite(LED, HIGH);
  delay(1000);
  digitalWrite(LED, LOW);
  delay(1000);
}

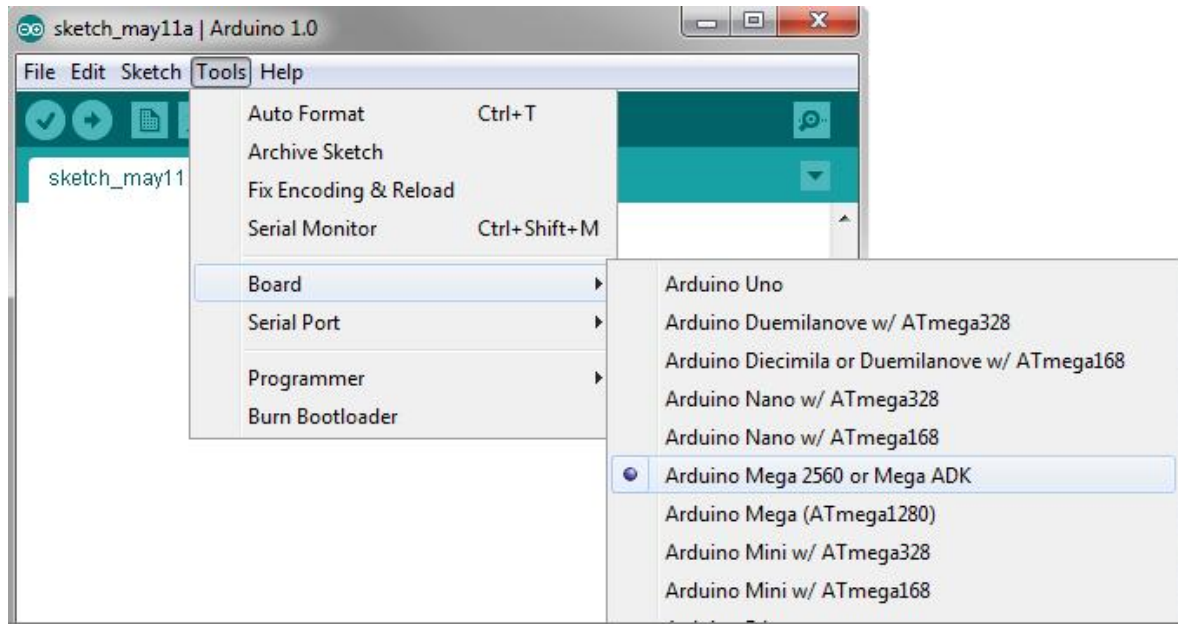
Done compiling.

Binary sketch size: 1,076 bytes (of a 32,256 byte maximum)
```

Εικ.19 Ανέβασμα σχεδίου

Αφού ελεγχθεί η ορθότητα του κώδικα ακολουθεί η επιλογή του τύπου του μικροελεγκτή που χρησιμοποιείται μέσω των εντολών :

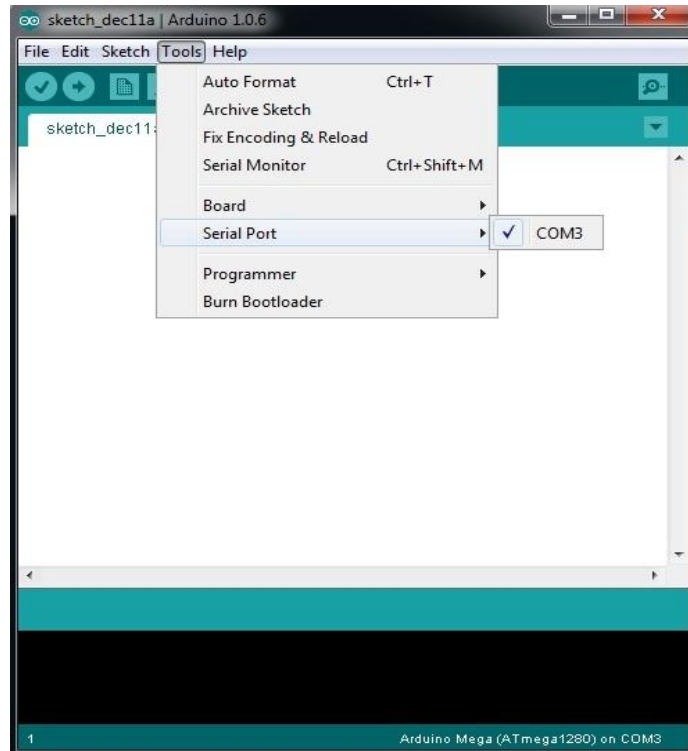
Tools → Board → Arduino Mega 2560



*Εικ.20 Επιλογή τύπου μικροελεγκτή*

Στη συνέχεια επιλέγεται η θύρα στην οποία έχει συνδεθεί το Arduino.

Tools → Serial Port → COM3

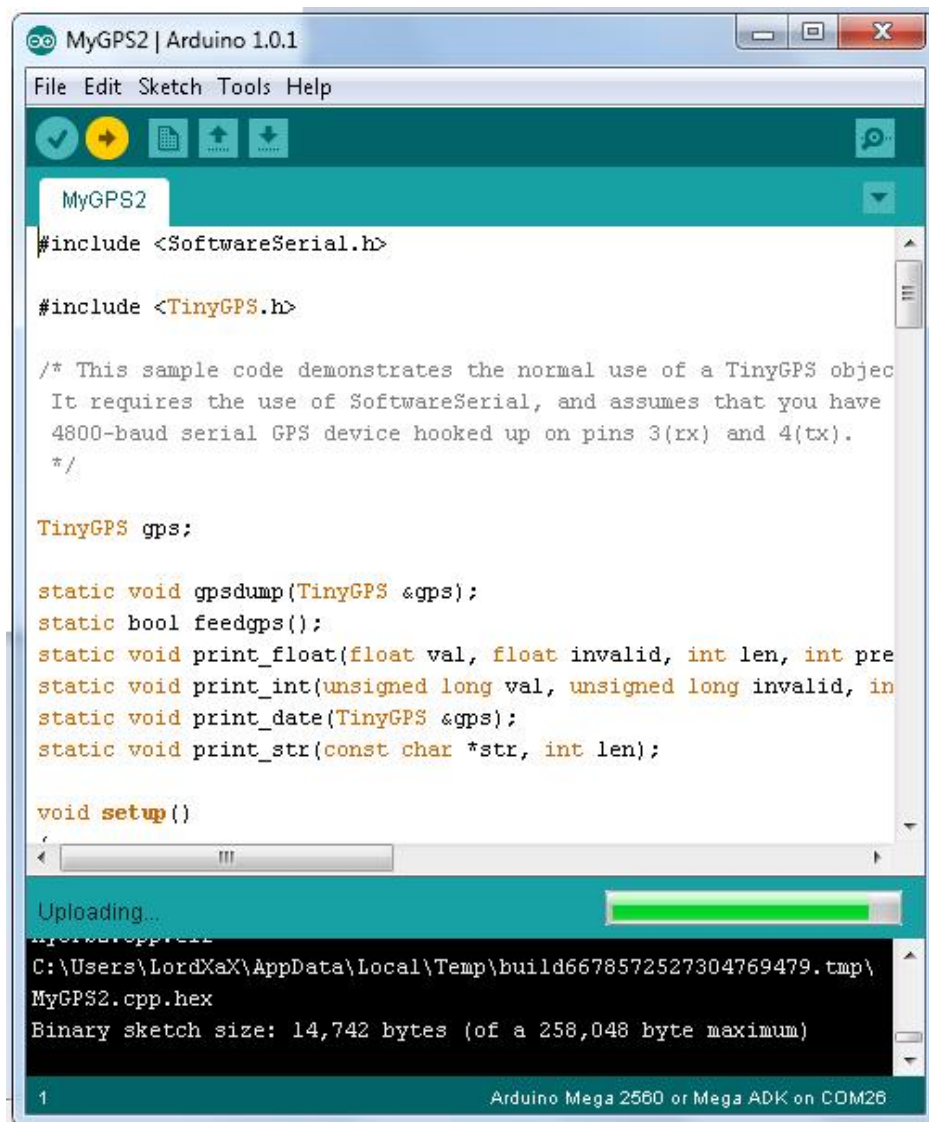


*Εικ.21 Επιλογή θύρας*

Στην περίπτωση που στο παράθυρο εμφανίζονται περισσότερες από μία θύρες, η διαδικασία εντοπισμού της κατάλληλης θύρας είναι η εξής :

Δεξί κλικ στο εικονίδιο «Ο υπολογιστής μου» που βρίσκεται στην επιφάνεια εργασίας → Ιδιότητες → Διαχείριση συσκευών → Θύρες COM → USB SERIAL

Τέλος, με την εντολή upload ο κώδικας φορτώνεται στον μικροελεγκτή και εμφανίζεται το μήνυμα ολοκλήρωσης της διαδικασίας προγραμματισμού (Done uploading).



```
MyGPS2 | Arduino 1.0.1
File Edit Sketch Tools Help
MyGPS2
#include <SoftwareSerial.h>

#include <TinyGPS.h>

/* This sample code demonstrates the normal use of a TinyGPS object.
   It requires the use of SoftwareSerial, and assumes that you have
   a 4800-baud serial GPS device hooked up on pins 3(rx) and 4(tx).
   */

TinyGPS gps;

static void gpstdump(TinyGPS &gps);
static bool feedgps();
static void print_float(float val, float invalid, int len, int pre);
static void print_int(unsigned long val, unsigned long invalid, int len);
static void print_date(TinyGPS &gps);
static void print_str(const char *str, int len);

void setup()

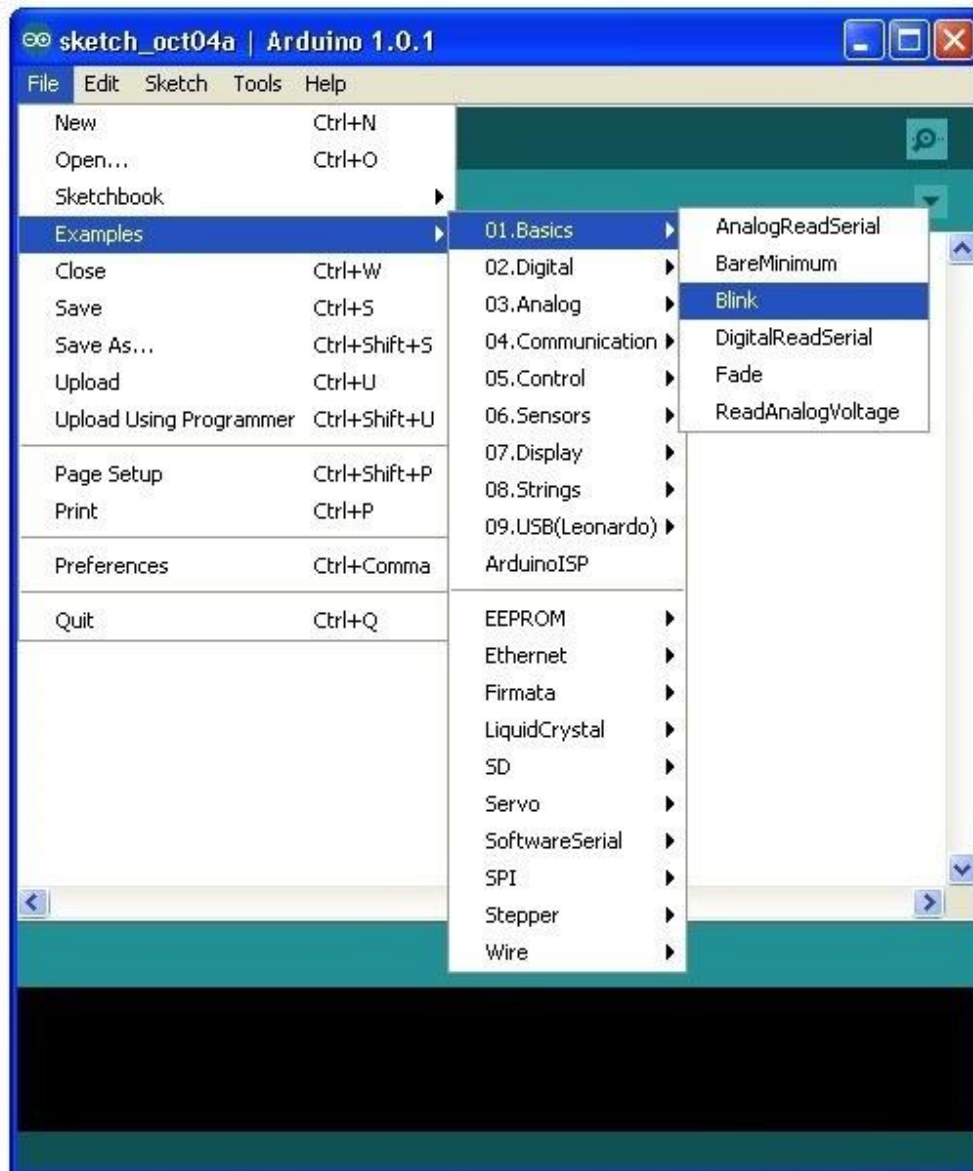
Uploading...
C:\Users\LordXaX\AppData\Local\Temp\build6678572527304769479.tmp\
MyGPS2.cpp.hex
Binary sketch size: 14,742 bytes (of a 258,048 byte maximum)

1 Arduino Mega 2560 or Mega ADK on COM26
```

Εικ.22 Ανέβασμα του κώδικα στον μικροελεγκτή

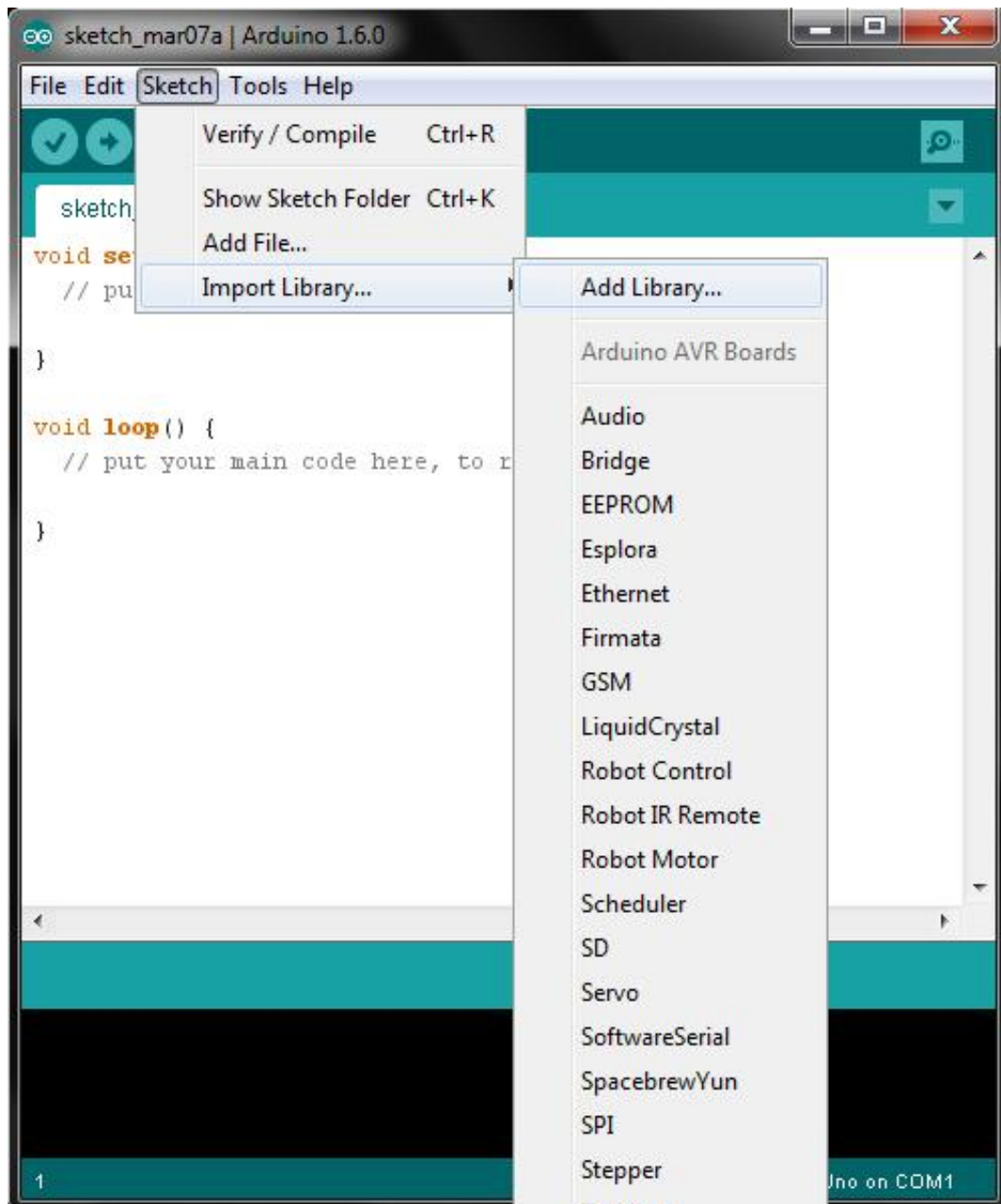
### 1.3.6 Βιβλιοθήκες

Το περιβάλλον του Arduino μπορεί να επεκταθεί μέσω της χρήσης βιβλιοθηκών, όπως συμβαίνει με τις περισσότερες πλατφόρμες. Οι βιβλιοθήκες παρέχουν επιπλέον λειτουργικότητα στα διάφορα σχεδιαγράμματα. Υπάρχουν διάφορα παραδείγματα κώδικα (examples), τα οποία παρέχονται από το πρόγραμμα, όμως δίνεται επιπλέον η δυνατότητα στο χρήστη είτε να κατεβάσει έτοιμες βιβλιοθήκες είτε να δημιουργήσει τις δικές του.



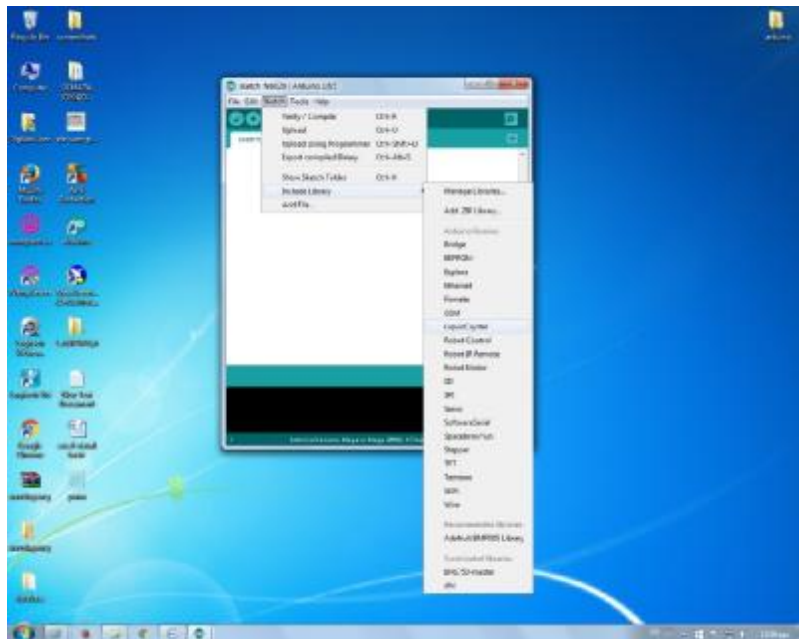
Εικ.23 Τα παραδείγματα του Arduino

Η διαδικασία εισαγωγής μιας βιβλιοθήκης στο πρόγραμμα του Arduino γίνεται με τη χρήση του **Sketch** που βρίσκεται στο μενού και στη συνέχεια με την επιλογή **Import Library** (εισαγωγή βιβλιοθήκης). Εκεί εμφανίζεται η επιλογή **Add Library** (προσθήκη βιβλιοθήκης).



Εικ.24 Εισαγωγή βιβλιοθήκης

Στην παρούσα πτυχιακή, εκτός από τη δημιουργία νέων βιβλιοθηκών, έγινε και χρήση έτοιμων βιβλιοθηκών οι οποίες παρέχονταν από το πρόγραμμα ή ήταν διαθέσιμες στο διαδίκτυο. Η διαδικασία εισαγωγής των βιβλιοθηκών παρουσιάζεται στη συνέχεια. Ο κώδικας κάθε βιβλιοθήκης είναι διαθέσιμος στο παράρτημα.

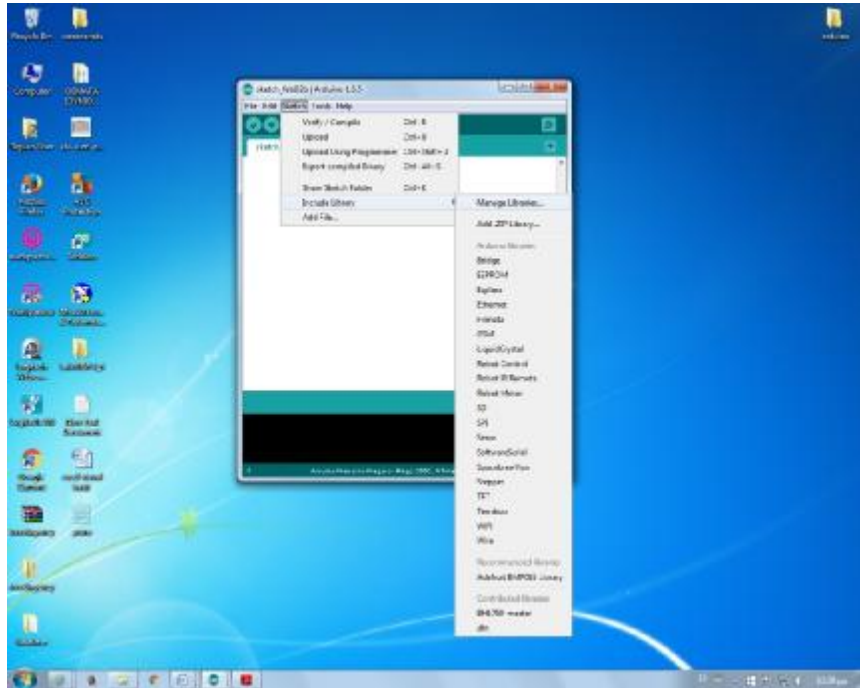


*Εικ.25 Χρήση της βιβλιοθήκης LiquidCrystal*

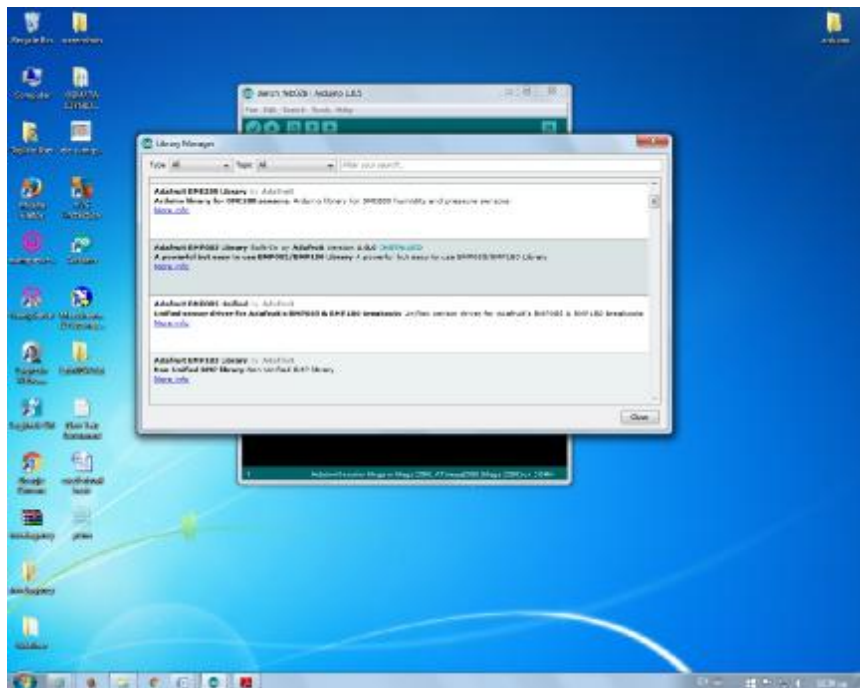


*Εικ.26 Χρήση της βιβλιοθήκης Wire*





Εικ.27 Εισαγωγή βιβλιοθηκών από το πρόγραμμα



Εικ.28 Εισαγωγή της βιβλιοθήκης Adafruit BMP085

## 1.4 ΕΠΙΚΟΙΝΩΝΙΑ ΜΕ ΤΟ ΔΙΑΔΙΚΤΥΟ ΜΕΣΩ ΕΤHERNET Ή WIFI SHIELD

### 1.4.1 Το Ethernet Shield

Το Arduino έχει τη δυνατότητα επικοινωνίας με το διαδίκτυο, η οποία πραγματοποιείται μέσω του Ethernet shield[5]. Το Arduino Ethernet shield έχει ως βάση το chip W5100 της Wiznet. Το shield είναι επέκταση του Arduino και συνδέεται με τον μικροελεγκτή με τη βοήθεια κατάλληλων συρμάτων ακροδεκτών. Η τροφοδοσία παρέχεται από το Arduino και η σύνδεση με τον υπολογιστή πραγματοποιείται μέσω καλωδίου Ethernet. Επιπλέον, υπάρχει υποδοχή κάρτας microSD για την αποθήκευση αρχείων. Στο κομμάτι του προγραμματισμού, με τη χρήση του βρόχου Ethernet.begin() και τις κατάλληλες ρυθμίσεις, το Ethernet shield αποκτά άμεσα μια διεύθυνση MAC και μια IP. Για να μπορέσει να συνδεθεί με το internet είναι απαραίτητη η βιβλιοθήκη Ethernet, η οποία πρέπει να προστεθεί στο πρόγραμμα. Επιπλέον, απαιτείται και η προσθήκη της κατάλληλης βιβλιοθήκης για την λειτουργία του αναγνώστη της κάρτας microSD.



*Εικ.29 Το Arduino Ethernet Shield*

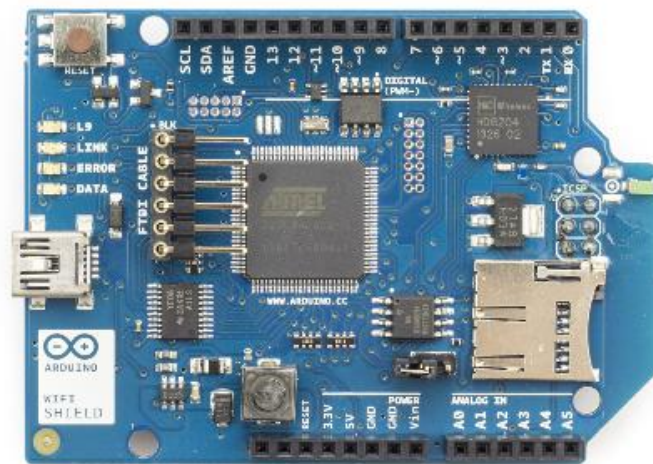
Παράμετροι :

Mac (Media access control) : πίνακας από 6 bytes

Ip : η IP διεύθυνση της συσκευής, πίνακας από 4 bytes

### 1.4.2 To Wifi Shield

Το WiFi Shield[6] επιτρέπει στην πλατφόρμα Arduino να συνδέεται με το διαδίκτυο κάνοντας χρήση της βιβλιοθήκης WiFi library και να αποθηκεύει αρχεία σε μία κάρτα microSD με τη βιβλιοθήκη SD library. Η επικοινωνία με το διαδίκτυο στην περίπτωση αυτή γίνεται ασύρματα. Ο κώδικας που χρησιμοποιεί η βιβλιοθήκη WiFi έχει πολλά κοινά στοιχεία με τη βιβλιοθήκη Ethernet. Το WiFi Shield διευρύνει σημαντικά το εύρος των εφαρμογών που μπορεί να υλοποιήσει ο χρήστης με το Arduino, καθώς καταργεί την ενσύρματη επικοινωνία.



*Εικ.30 Το Arduino WiFi Shield*

## ΚΕΦΑΛΑΙΟ 2<sup>ο</sup>

### ΠΕΙΡΑΜΑΤΙΚΟ ΜΕΡΟΣ

#### 2.1 Υλικά Σύνδεσης

##### Ø Arduino Mega 2560 με καλώδιο USB

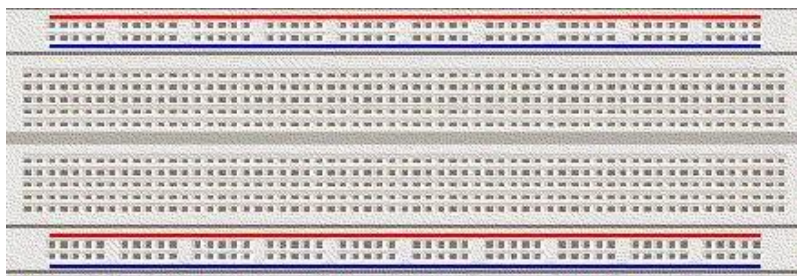
Στην παρούσα εργασία χρησιμοποιήθηκε το Arduino Mega 2560, το οποίο είναι η νεότερη έκδοση του Mega. Βασίζεται στην τεχνολογία ATmega2560, με μνήμη Flash στα 256 KB και τάση λειτουργίας 5 V. Η σύνδεση με τον υπολογιστή έγινε μέσω καλωδίου USB.



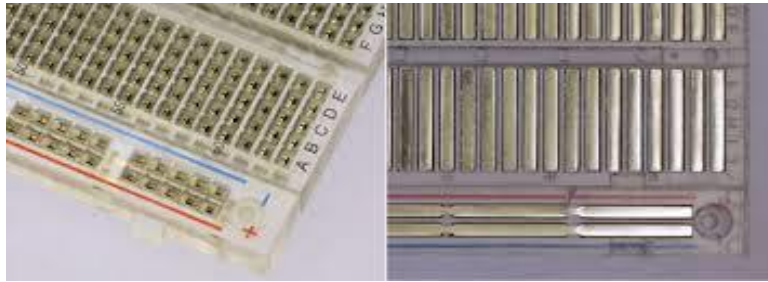
*Εικ.31 Το Arduino Mega 2560*

##### Ø Solderless Breadboard

Το breadboard είναι μια πλαστική πλακέτα με υποδοχές, οι οποίες συνδέονται μεταξύ τους με αγώγιμο υλικό δημιουργώντας με τον τρόπο αυτό κοινά σημεία. Η χρήση της πλακέτας αυτής διευκολύνει τη σύνδεση πολλών αισθητήρων στο Arduino, καθώς αυξάνεται σημαντικά ο αριθμός των υποδοχών. Για την εργασία χρησιμοποιήθηκε το breadboard MB-102 της εταιρείας EVER-MUSE.



*Εικ.32 Το breadboard*



*Εικ.33 Το εσωτερικό του breadboard*

### **Ø Καλώδια**

Αφού γίνει η επιλογή των κατάλληλων αισθητήρων και της πλατφόρμας ακολουθεί η υλοποίηση του κυκλώματος. Για τη σύνδεση των αισθητήρων στο Arduino και το breadboard χρησιμοποιήθηκαν τρία είδη καλωδίων και συγκεκριμένα Αρσενικό-Αρσενικό (κίτρινα), Αρσενικό-Θηλυκό (γκρι) και Θηλυκό-Θηλυκό (μαύρα).



*Εικ.34 Καλώδια σύνδεσης αισθητήρων*

### **Ø Αντιστάσεις**

Για την υλοποίηση των κυκλωμάτων χρησιμοποιήθηκαν διάφορες αντιστάσεις (1KΩ,500Ω κλπ)



*Εικ.35 Αντιστάσεις*

### **Ø Ποτενσιόμετρο**

Το ποτενσιόμετρο συνδέθηκε με την LCD οθόνη για να υπάρχει δυνατότητα ρύθμισης της φωτεινότητας. Διαθέτει τρεις υποδοχές εκ των οποίων μία συνδέεται στην πηγή (5V), μία στη γείωση (GND) και μία σε κάποια από τις αναλογικές εισόδους (A0-A15).

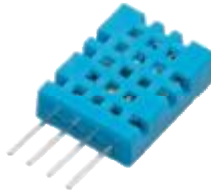


*Εικ.36 Το ποτενσιόμετρο*

## 2.2 Αισθητήρες

Για τη διεξαγωγή των πειραματικών μετρήσεων έγινε χρήση των παρακάτω αισθητήρων :

### 1. ΑΙΣΘΗΤΗΡΑΣ ΘΕΡΜΟΚΡΑΣΙΑΣ/ΥΓΡΑΣΙΑΣ ΠΕΡΙΒΑΛΛΟΝΤΟΣ (DHT11)[7]

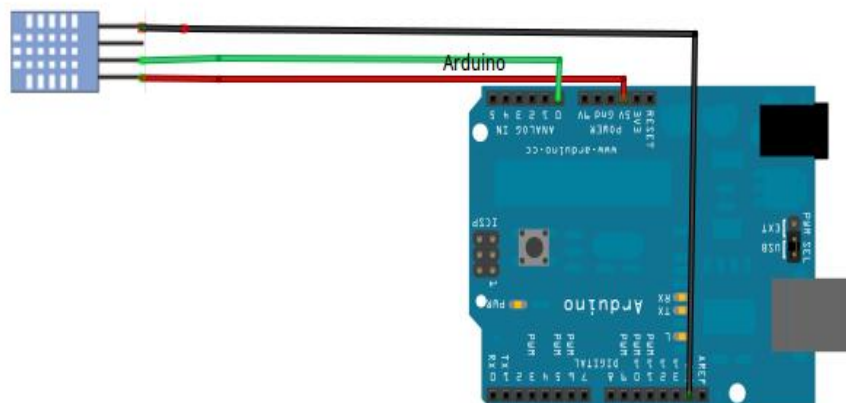


Ο αισθητήρας θερμοκρασίας/υγρασίας διαθέτει 4 pins. Ο τρόπος σύνδεσης είναι ο εξής :

- Pin 1 → Vcc (Τάση τροφοδοσίας 5V)
- Pin 2 → A0 (Μία εκ των αναλογικών εισόδων A0-A15)
- Pin 3 → N/C (Δεν συνδέεται-Not Connected)
- Pin 4 → GND (Γείωση-Ground)

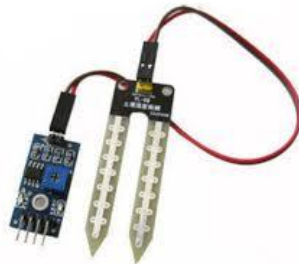
Τα τεχνικά χαρακτηριστικά του αισθητήρα δίνονται παρακάτω.

- Τάση λειτουργίας : 3-5V και I/O
- Μέγιστο ρεύμα : 2.5mA μέγιστο ρεύμα κατά τη διάρκεια μετατροπής
- Ακρίβεια : Καλή για 20-80% αναγνώσεις υγρασίας με 5% ακρίβεια
- Ακρίβεια : Καλή για 0-50°C αναγνώσεις θερμοκρασίας  $\pm 2^{\circ}\text{C}$  ακρίβεια
- Ρυθμός δειγματοληψίας : Όχι μεγαλύτερος του 1 Hz (μία φορά ανά δευτερόλεπτο)
- Διαστάσεις αισθητήρα : 15.5mm x 12mm x 5.5mm
- 4 pins με 0.1" απόσταση



Εικ.37 Σύνδεση του DHT11 με το Arduino

## 2. ΑΙΣΘΗΤΗΡΑΣ ΥΓΡΑΣΙΑΣ ΕΔΑΦΟΥΣ (SOIL HYGROMETER)[8]



Ο αισθητήρας υγρασίας εδάφους διαθέτει 4 pins τα οποία συνδέονται ως εξής :

Pin 1 → A0 (Μία εκ των αναλογικών εισόδων A0-A15)

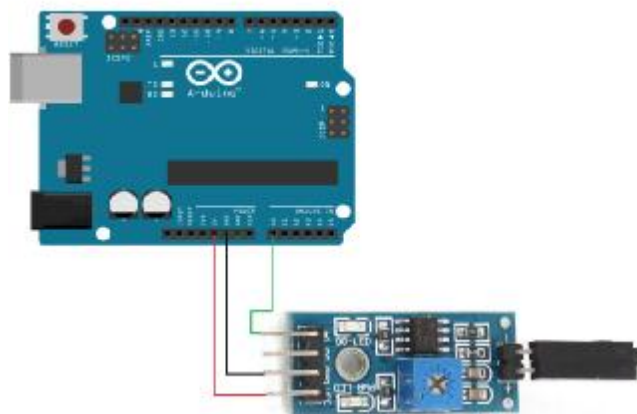
Pin 2 → D0 (Δεν συνδέεται-N/C)

Pin 3 → GND (Γείωση-GROUND)

Pin 4 → VCC (Τάση τροφοδοσίας 5V)

Τα τεχνικά χαρακτηριστικά του αισθητήρα είναι τα ακόλουθα.

- Τάση λειτουργίας : 3.3V~5V
- Ρυθμιζόμενη ευαισθησία (ρύθμιση με το μπλε ψηφιακό ποτενσιόμετρο)
- Λειτουργία διπλής εξόδου, η αναλογική έξοδος είναι πιο ακριβής
- Σταθερή βιδωτή τρύπα για εύκολη εγκατάσταση
- Με ενδεικτική λυχνία λειτουργίας ( κόκκινο) και η ένδειξη της ψηφιακής εξόδου μεταγωγής ( πράσινο )
- Έχει LM393 chip συγκριτή, σταθερό
- Διαστάσεις Panel PCB : 3cm x 1.5cm
- Διαστάσεις Probe εδάφους : 6cm x 2cm
- Μήκος καλωδίου : 21cm



Εικ.38 Σύνδεση του αισθητήρα υγρασίας εδάφους με το Arduino



### 3. ΑΙΣΘΗΤΗΡΑΣ ΥΨΟΥΣ ΒΡΟΧΗΣ (RAIN OR WATER LEVEL SENSOR)[9]



Ο αισθητήρας ύψους βροχής διαθέτει 3 pins. Αυτά συνδέονται με το Arduino ως εξής :

Pin 1 → S (Μία εκ των αναλογικών εισόδων A0-A15)

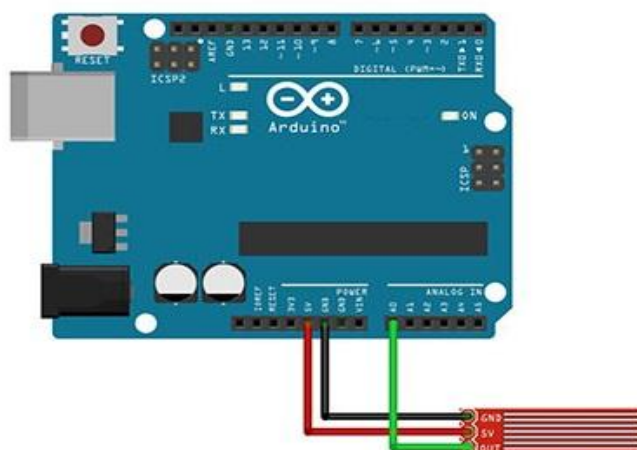
Pin 2 → + (Τάση τροφοδοσίας 5V)

Pin 3 → - (Γείωση-GROUND)

Τα τεχνικά χαρακτηριστικά του αισθητήρα είναι τα παρακάτω.

- Τάση λειτουργίας : DC 3-5 V
- Ρεύμα λειτουργίας : μικρότερο από 20mA
- Τύπος αισθητήρα : Αναλογικός
- Περιοχή ανίχνευσης : 40mm x 16mm
- Θερμοκρασία λειτουργίας : 10°C – 30°C
- Υγρασία : 10% - 90% χωρίς συμπύκνωση
- Διαστάσεις αισθητήρα : 62mm x 20mm x 8mm

Για τον αισθητήρα αυτό έγινε βαθμονόμηση με τη χρήση του προγράμματος ORIGIN.



*Εικ.39 Σύνδεση του αισθητήρα βροχής με το Arduino*

#### 4. ΑΙΣΘΗΤΗΡΑΣ ΒΑΡΟΜΕΤΡΙΚΗΣ ΠΙΕΣΗΣ (BMP085 BAROMETER)[10]

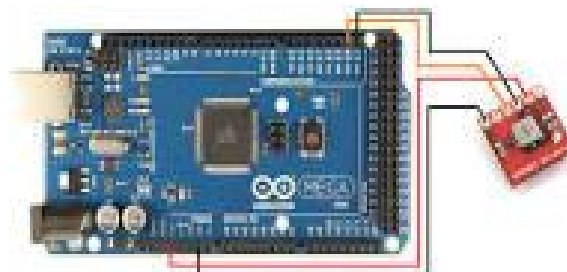


Ο αισθητήρας BMP085 διαθέτει 7 pins τα οποία συνδέονται στο Arduino ως εξής :

- Pin 1 → 5V (Δεν συνδέεται - N/C)
- Pin 2 → 3,3V (Τάση τροφοδοσίας 3,3V)
- Pin 3 → GND (Γείωση-GROUND)
- Pin 4 → SDA ( Communication SDA 20)
- Pin 5 → SCL (Communication SCL 21)
- Pin 6 → XCLR (Δεν συνδέεται – N/C)
- Pin 7 → EOC (Δεν συνδέεται – N/C)

Τα τεχνικά χαρακτηριστικά του αισθητήρα είναι τα εξής :

- Εύρος πίεσης : 300 – 1100 hPa (9000m πάνω από το επίπεδο της θάλασσας - 500m)
- Τάση τροφοδοσίας : 1.8V - 3.6V (το VDDA) 1.62V - 3.6V (VDDR)
- LCC8 πακέτο: κεραμικό πακέτο χωρίς μόλυβδο (LCC)
- Χαμηλή κατανάλωση ενέργειας : 5μΑ σε κανονική λειτουργία
- Ακρίβεια : Κατάσταση χαμηλής κατανάλωσης, με ανάλυση 0.06 hPa (0.5 m)
- Υψηλή γραμμική λειτουργία με ανάλυση 0.03 hPa (0.25m)
- I<sup>2</sup>C (Inter-Integrated Circuit) : Το I<sup>2</sup>C είναι ένα interface επικοινωνίας με το οποίο επιτρέπεται αμφίδρομη (half duplex) επικοινωνία ανάμεσα σε μία master και μία slave συσκευή οι οποίες είναι συνδεδεμένες πάνω στο bus.
- Με έξοδο θερμοκρασίας
- Αντιστάθμιση θερμοκρασίας
- Χωρίς μόλυβδο, RoHS συμβατό
- Χρόνος αντίδρασης : 7.5ms
- Ρεύμα αναμονής : 0.1μΑ



*Εικ.40 Σύνδεση του αισθητήρα βαρομετρικής πίεσης με το Arduino*

## 5. ΑΙΣΘΗΤΗΡΑΣ ΕΝΤΑΣΗΣ ΦΩΤΙΣΜΟΥ (LIGHT SENSOR)[11]

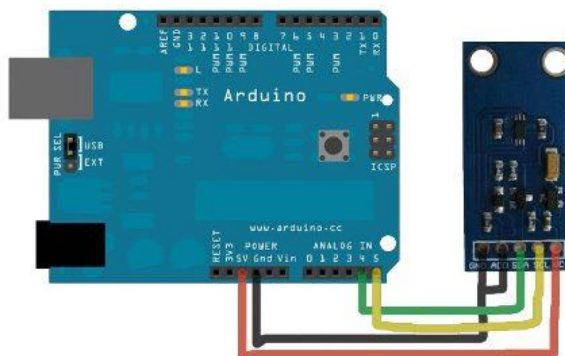


Ο αισθητήρας φωτός BH1750 διαθέτει 5 pins τα οποία συνδέονται στο Arduino ως εξής :

- Pin 1 → GND (Γείωση-GROUND)
- Pin 2 → ADD
- Pin 3 → SDA
- Pin 4 → SCL
- Pin 5 → VCC (Τάση τροφοδοσίας 5V)

Τα τεχνικά χαρακτηριστικά του αισθητήρα είναι τα εξής :

- Ψηφιακή μονάδα ανίχνευσης έντασης του φωτός: GY-30
- ROHM το αρχικό τσιπ BH1750FVI
- Παροχή ρεύματος: 3-5 V
- Εύρος φωτισμού : 0-65535 lx
- Αισθητήρας με ενσωματωμένο μετατροπέα 16bit AD
- Άμεση ψηφιακή έξοδος, πολύπλοκοι υπολογισμοί παραλείπονται
- Δεν κάνει διάκριση μεταξύ του φωτός περιβάλλοντος
- Κοντά στην οπτική ευαισθησία των φασματικών χαρακτηριστικών
- 1 lux μέτρηση υψηλής ακρίβειας από ένα ευρύ φάσμα φωτεινότητας
- I<sup>2</sup>C (Inter-Integrated Circuit) : Το I<sup>2</sup>C είναι ένα interface επικοινωνίας με το οποίο επιτρέπεται αμφίδρομη (half duplex) επικοινωνία ανάμεσα σε μία master και μία slave συσκευή οι οποίες είναι συνδεδεμένες πάνω στο bus.



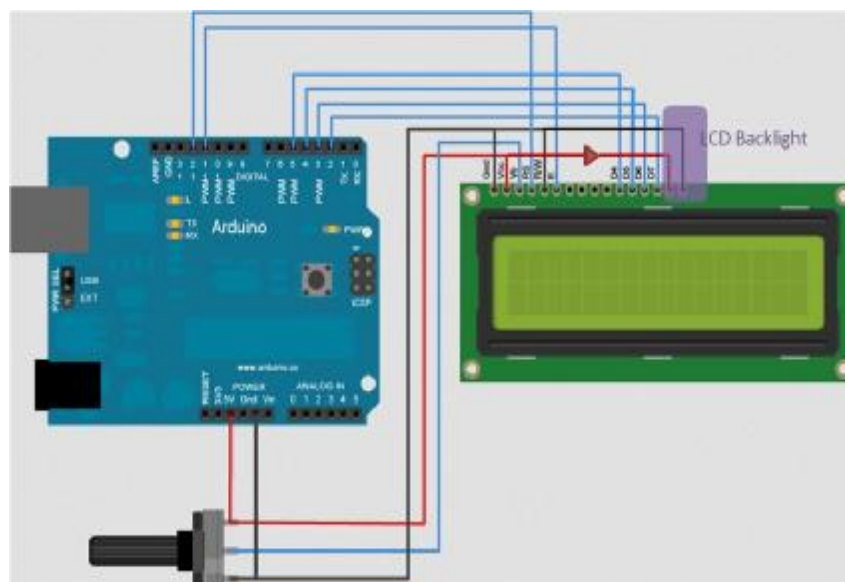
*Εικ.41 Σύνδεση του αισθητήρα φωτός με το Arduino*

## 2.3 LCD οθόνη

Για την απεικόνιση των τιμών που μετρήθηκαν μέσω των αισθητήρων, έγινε χρήση της οθόνης LCD[12] η οποία διαθέτει 2 γραμμές και 16 στήλες. Για τη σωστή λειτουργία της έγινε χρήση της βιβλιοθήκης LiquidCrystal (βλ. παράρτημα), η οποία είναι αποθηκευμένη στο sketch του Arduino. Στην οθόνη συνδέθηκε ένα ποτενσιόμετρο με το οποίο ρυθμίστηκε η φωτεινότητα της.

Η LCD οθόνη που χρησιμοποιήθηκε διαθέτει 16 pins τα οποία συνδέονται με το Arduino ως εξής :

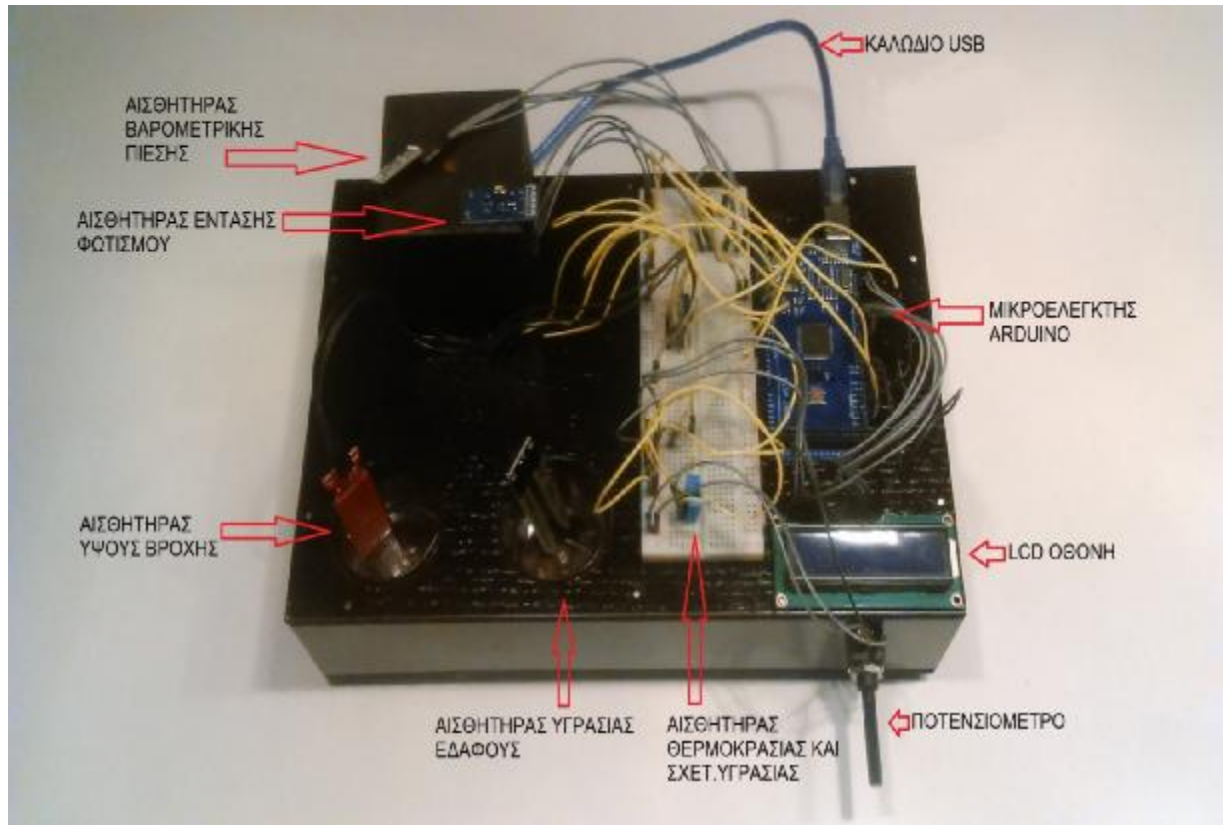
- Pin 1 → VSS (Γείωση-GROUND)
- Pin 2 → VDD (Τάση τροφοδοσίας 5V)
- Pin 3 → VO (Συνδέεται με το pin 2 του ποτενσιόμετρου)
- Pin 4 → RS (Συνδέεται με την PWM 12 έξοδο του Arduino)
- Pin 5 → RW (Γείωση-GROUND)
- Pin 6 → E (Συνδέεται με την PWM 11 έξοδο του Arduino)
- Pin 7 → D0 (Δεν συνδέεται-N/C)
- Pin 8 → D1 (Δεν συνδέεται-N/C)
- Pin 9 → D2 (Δεν συνδέεται-N/C)
- Pin 10 → D3 (Δεν συνδέεται-N/C)
- Pin 11 → D4 (Συνδέεται με την PWM 5 έξοδο του Arduino)
- Pin 12 → D5 (Συνδέεται με την PWM 4 έξοδο του Arduino)
- Pin 13 → D6 (Συνδέεται με την PWM 3 έξοδο του Arduino)
- Pin 14 → D7 (Συνδέεται με την PWM 2 έξοδο του Arduino)
- Pin 15 → A (Συνδέεται με μια αντίσταση σε σειρά και τα 5V)
- Pin 16 → K (Γείωση-GROUND)



Εικ.42 Σύνδεση της οθόνης LCD με το Arduino

## 2.4 Η μακέτα του συστήματος

Στην παρακάτω εικόνα παρουσιάζεται η μακέτα που κατασκευάστηκε για τις ανάγκες της εργασίας. Διακρίνονται οι αισθητήρες που χρησιμοποιήθηκαν, η οθόνη LCD, το Arduino, το breadboard και το ποτενσιόμετρο.



Εικ.43 Η μακέτα

## ΚΕΦΑΛΑΙΟ 3<sup>ο</sup>

### ΕΠΕΞΕΡΓΑΣΙΑ ΜΕΤΡΗΣΕΩΝ

#### 3.1 ΑΙΣΘΗΤΗΡΕΣ

Αισθητήρες[13] ονομάζονται οι συσκευές οι οποίες ανιχνεύουν στην είσοδο ένα φυσικό μέγεθος και παράγουν από αυτό μια μετρήσιμη έξοδο. Τα φυσικά μεγέθη που μετρήθηκαν στην παρούσα εργασία είναι :

Ø Θερμοκρασία ατμόσφαιρας[14]

Θερμοκρασία ατμόσφαιρας ονομάζεται το φυσικό μέγεθος που περιγράφει τη θερμοκρασία την οποία έχει ο ατμοσφαιρικός αέρας πάνω από μια περιοχή. Είναι ένας από τους βασικούς παράγοντες που καθορίζουν την πρόγνωση του καιρού σε μία περιοχή. Τα όργανα μέτρησης της θερμοκρασίας ονομάζονται θερμόμετρα και οι κυριότερες κλίμακες μέτρησης είναι οι Κελσίου (°C), Kelvin (K), Fahrenheit (F).

Ø Υγρασία[15]

Στον ατμοσφαιρικό αέρα περιέχονται και υδρατμοί που προέρχονται από την εξάτμιση υγρών επιφανειών, κυρίως των θαλασσών. Η παρουσία αυτών των υδρατμών στον αέρα ονομάζεται υγρασία και διακρίνεται σε "απόλυτη" και σε "σχετική υγρασία".

Απόλυτη υγρασία ονομάζεται η μάζα των υδρατμών (σε γραμμάρια) που περιέχεται σε 1m<sup>3</sup>, δηλαδή η πυκνότητα του αέρα σε υδρατμούς.

Σχετική υγρασία ονομάζεται ο λόγος της ποσότητας ή της μάζας των υδρατμών, που περιέχει ο αέρας, προς εκείνη την ποσότητα ή το βάρος των υδρατμών τους οποίους μπορεί να συμπεριλάβει (υπό την αυτή θερμοκρασία και πίεση) μέχρι να κορεσθεί. Η σχετική υγρασία εκφράζεται επί τοις %.

Όταν ο αέρας περιέχει τη μέγιστη δυνατή ποσότητα υδρατμών ονομάζεται κορεσμένος. Η θερμοκρασία στην οποία ο ακόρεστος αέρας φτάνει στον κορεσμό (καθώς ψύχεται), ονομάζεται σημείο δρόσου. Η μέτρηση της υγρασίας της ατμόσφαιρας γίνεται με τα υγρόμετρα.

Ø Βαρομετρική ή ατμοσφαιρική πίεση[16]

Βαρομετρική ή ατμοσφαιρική πίεση ονομάζεται το φυσικό μέγεθος που περιγράφει την πίεση που ασκεί η ατμόσφαιρα με το βάρος της στην επιφάνεια της γης. Η ατμοσφαιρική ή βαρομετρική πίεση μεταβάλλεται «οριζόντια» και «κατακόρυφα». Οι «οριζόντιες μεταβολές» είναι πολύ μικρότερες από τις «κατακόρυφες μεταβολές» όμως έχουν εξαιρετική σημασία στη δημιουργία των καιρικών φαινομένων, όπως για παράδειγμα οι άνεμοι. Η βαρομετρική ή ατμοσφαιρική πίεση στην επιφάνεια της γης, ισούται περίπου με το βάρος της στήλης υδραργύρου ύψους 760mm.

Ø Βαρομετρική τάση[16]

Στην ατμοσφαιρική πίεση ως βαρομετρική τάση ορίζεται η μεταβολή της ατμοσφαιρικής πίεσης εντός τριών ωρών προ εκάστης καθορισμένης κύριας ώρας παρατήρησης. Η τιμή της βαρομετρικής τάσης αποτελεί βασικό στοιχείο στις μεταβολές του καιρού.

Ø Ένταση φωτισμού[17]

Ένταση φωτισμού ή επιφωτισμός ονομάζεται το φυσικό μέγεθος που περιγράφει το φως που προσπίπτει σε μια επιφάνεια. Ορίζεται ως η φωτεινή ροή που προσπίπτει κάθετα σε μια επιφάνεια προς το εμβαδόν της επιφάνειας αυτής. Αν η φωτεινή ροή δεν προσπίπτει κάθετα στην επιφάνεια, τότε αναλύεται σε δύο συνιστώσες, μία κάθετη στην επιφάνεια και μία παράλληλη. Για τον υπολογισμό λαμβάνεται υπόψιν μόνο η κάθετη συνιστώσα. Μονάδα μέτρησης της έντασης φωτισμού είναι το  $\text{lux} = \text{lm/m}^2$ .

Οι αισθητήρες συνδέονται, ανάλογα με τα χαρακτηριστικά τους, στις ψηφιακές ή τις αναλογικές εισόδους του Arduino. Για κάθε αισθητήρα έγινε χρήση της κατάλληλης βιβλιοθήκης για την μετατροπή των τιμών στην είσοδο σε μετρήσιμες τιμές στην έξοδο. Οι αισθητήρες που χρησιμοποιήθηκαν στην εργασία, παρουσιάστηκαν αναλυτικά σε προηγούμενο κεφάλαιο.

## ΜΕΤΡΗΣΗ

Μέτρηση είναι μια διαδικασία, μέσω της οποίας δίνεται η δυνατότητα καταγραφής πληροφοριών, χρησιμοποιώντας υλικά και διάφορες μεθόδους επεξεργασίας. Οι μετρήσεις πραγματοποιούνται με τη χρήση συσκευών που ονομάζονται αισθητήρες. Όταν οι αισθητήρες συνδεθούν με όργανα κατάλληλα για την επεξεργασία των μετρήσεων, προκύπτει ένα σύνολο που ονομάζεται όργανα μέτρησης.

## ΒΑΘΜΟΝΟΜΗΣΗ

Ως βαθμονόμηση ορίζεται η διαδικασία μέσω της οποίας γνωστές τιμές εισόδου εφαρμόζονται σε ένα μετρητικό σύστημα ή αισθητήρα με σκοπό την παρατήρηση της εξόδου του συστήματος. Η βαθμονόμηση καθορίζει ουσιαστικά την κλίμακα του οργάνου. Τυπικά, ένας αισθητήρας βαθμονομείται με την παροχή ενός γνωστού σήματος στην είσοδο και καταγραφή της εξόδου. Η διαδικασία της βαθμονόμησης εφαρμόστηκε στην παρούσα εργασία στον αισθητήρα ύψους βροχής, καθώς έδινε στη σειριακή οθόνη τιμές τάσης οι οποίες έπρεπε να μετατραπούν στις κατάλληλες μονάδες, δηλαδή σε mm.

Για τις μετατροπές αυτές έγινε χρήση του προγράμματος ORIGIN.

### 3.2 ORIGIN



*Εικ.44 Το λογότυπο του ORIGIN*

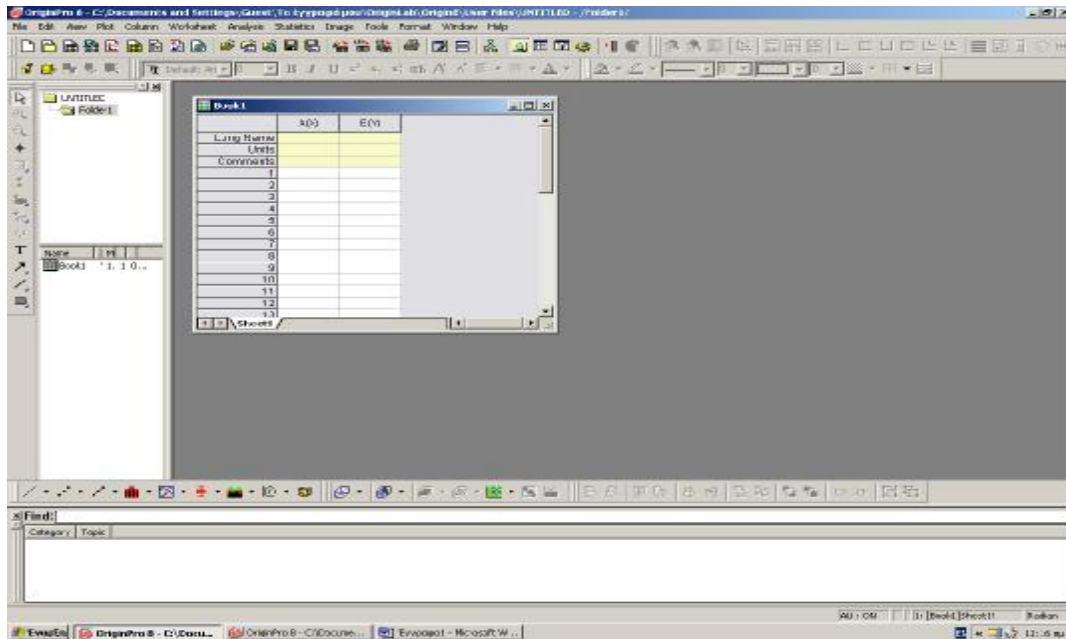
Το ORIGIN[18] είναι μια εφαρμογή για ανάλυση δεδομένων και δημιουργία γραφικών παραστάσεων, προσαρμοσμένη στις ανάγκες επιστημόνων και μηχανικών. Η εφαρμογή αυτή χρησιμοποιήθηκε με στόχο την ανάλυση των δεδομένων που συγκεντρώθηκαν κατά τη διάρκεια του πειραματικού μέρους. Σκοπός είναι η εύρεση της βέλτιστης πολυωνυμικής συνάρτησης (δηλαδή του πολυωνύμου με την μεγαλύτερη δυνατή ακρίβεια), ώστε κατά την μετατροπή να ελαχιστοποιείται το σφάλμα. Όσο μικρότερο είναι το σφάλμα, τόσο πιο κοντά στις πραγματικές θα είναι οι τιμές που καταγράφονται από τους αισθητήρες. Ο αισθητήρας για τον οποίο εφαρμόστηκε η διαδικασία της βαθμονόμησης ήταν ο αισθητήρας ύψους βροχής. Τα βήματα που ακολουθήθηκαν για τη βαθμονόμηση του, περιγράφονται αναλυτικά στη συνέχεια.

#### **ΠΕΡΙΓΡΑΦΗ ΒΑΘΜΟΝΟΜΗΣΗΣ**

Για τη βαθμονόμηση του αισθητήρα ύψους βροχής υλοποιήθηκε η παρακάτω πειραματική διαδικασία :

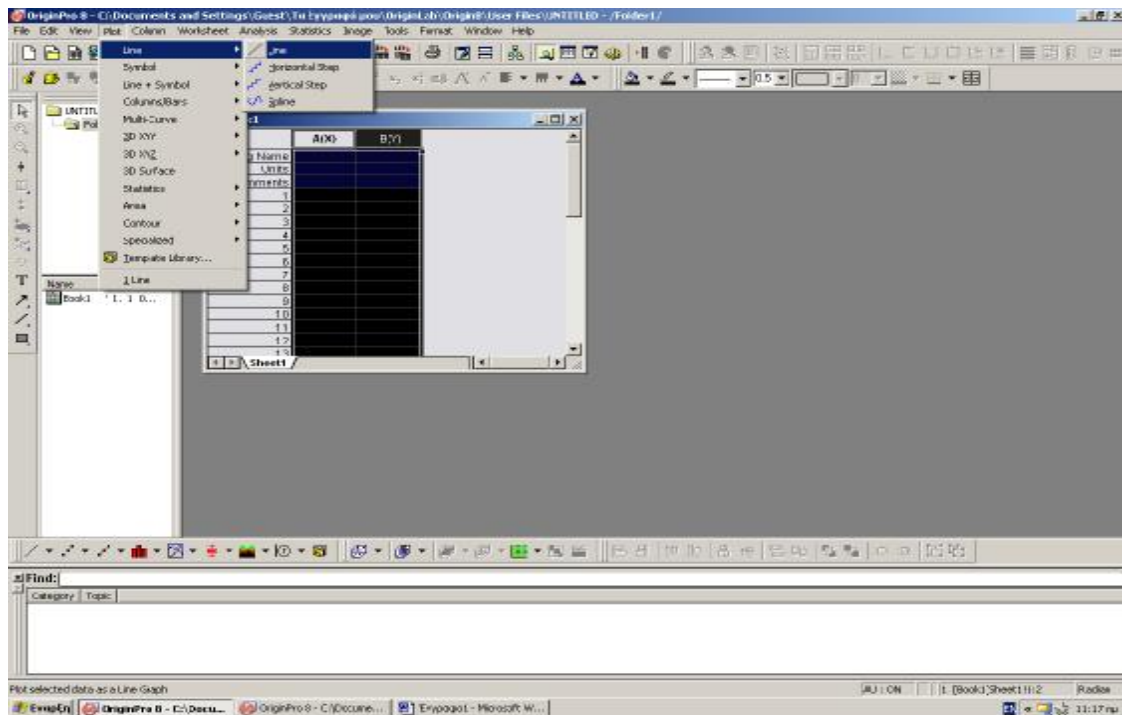
Χρησιμοποιήθηκε ο αισθητήρας βροχής και ένα πλαστικό ποτήρι, στο οποίο δημιουργήθηκε κλίμακα από κάτω προς τα πάνω ανά 5 mm. Αρχικά καταγράφηκε η τιμή που μετρούσε ο αισθητήρας όταν το ποτήρι ήταν άδειο (η τιμή αυτή εμφανίζεται στη σειριακή οθόνη του προγράμματος Arduino). Στη συνέχεια προστέθηκε νερό μέχρι τα 5 mm και καταγράφηκε η νέα τιμή που μετρούσε ο αισθητήρας. Τα βήματα επαναλήφθηκαν ανά 5 mm μέχρι τα 45 mm που είχε τη δυνατότητα να μετρήσει ο αισθητήρας. Τα ζεύγη τιμών που προέκυψαν από το πείραμα, μεταφέρθηκαν στο Book1 του προγράμματος ORIGIN στις στήλες A(X) και B(Y) αντίστοιχα. Στην A(X) τοποθετήθηκαν οι τιμές της αναλογικής εισόδου A0, δηλαδή η τιμή της τάσης που κατέγραψε ο αισθητήρας, ενώ στην B(Y) οι τιμές που μετρήθηκαν στην πειραματική διαδικασία, δηλαδή τα mm βροχής. Μετά την ολοκλήρωση της καταγραφής των τιμών, ακολούθησε η διαδικασία εύρεσης του βέλτιστου πολυωνύμου.





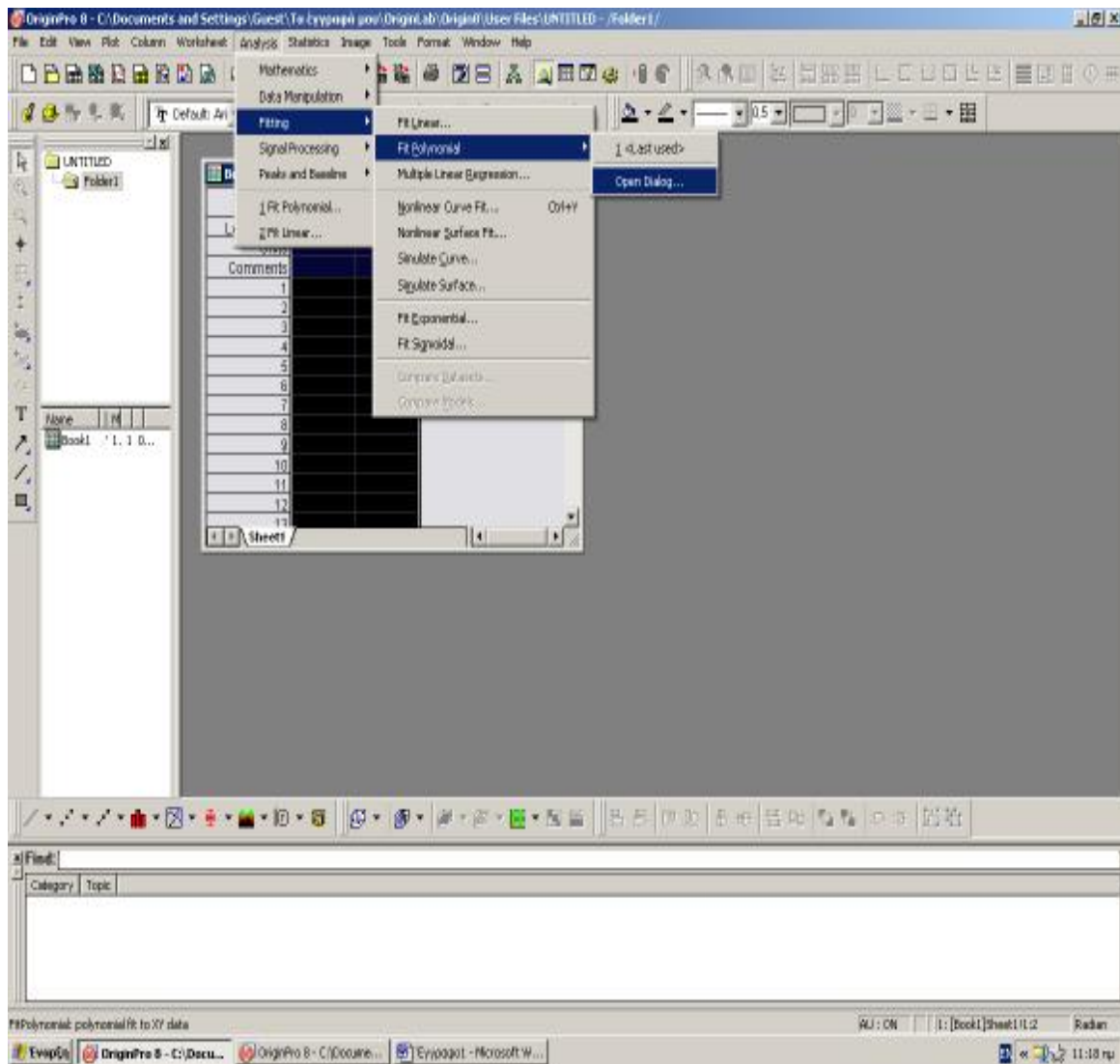
Εικ.45 Το περιβάλλον του προγράμματος ORIGIN

Για την δημιουργία γραφικής παράστασης επιλέχθηκαν οι δύο στήλες και στη συνέχεια Plot → Line → Line.



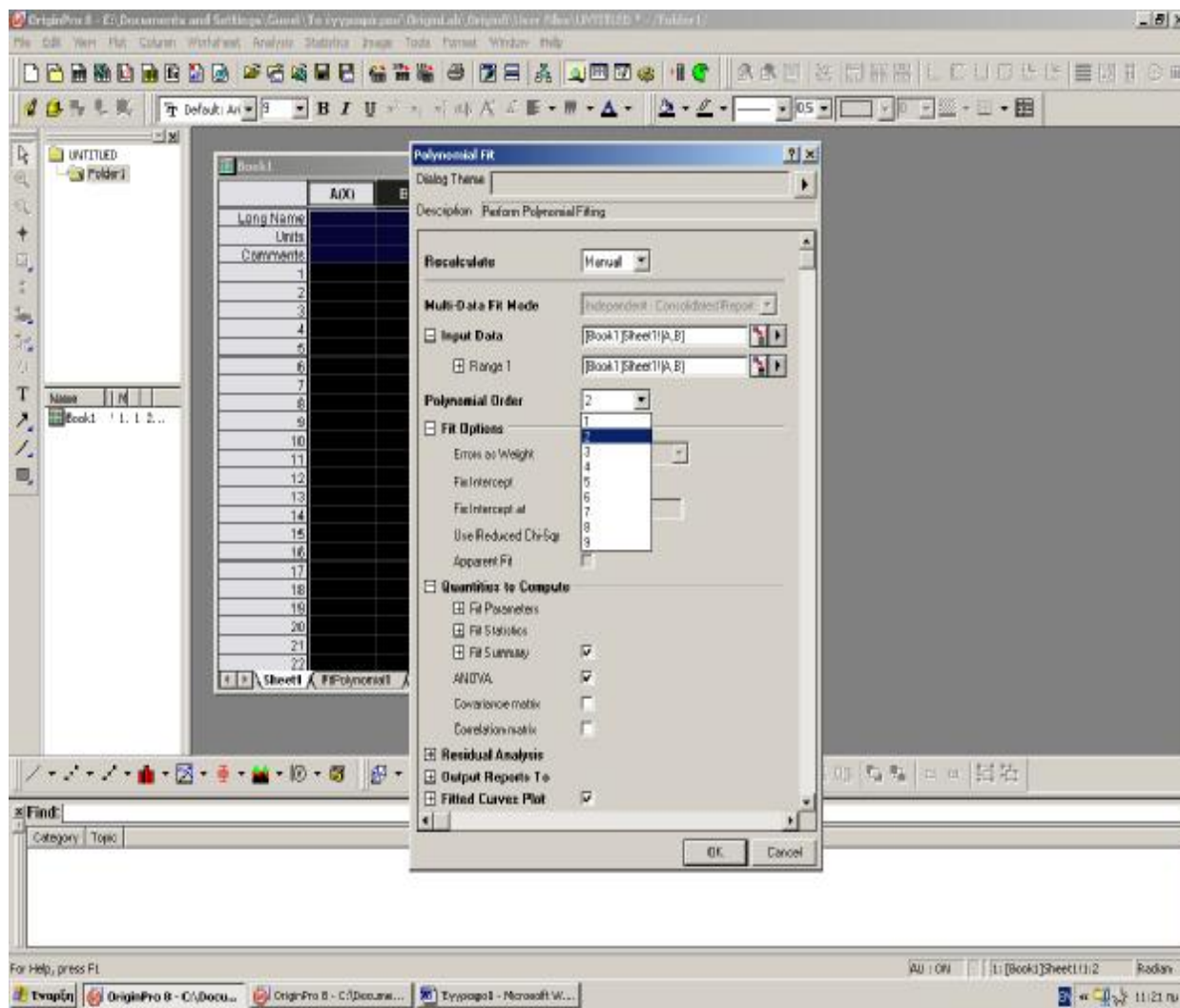
Εικ.46 Επιλογή Plot

Στη συνέχεια πραγματοποιήθηκε η εύρεση του βέλτιστου πολωνύμου ακολουθώντας τα βήματα Analysis → Fitting → Fit polynomial → Open dialog.



Εικ.47 Επιλογή Analysis

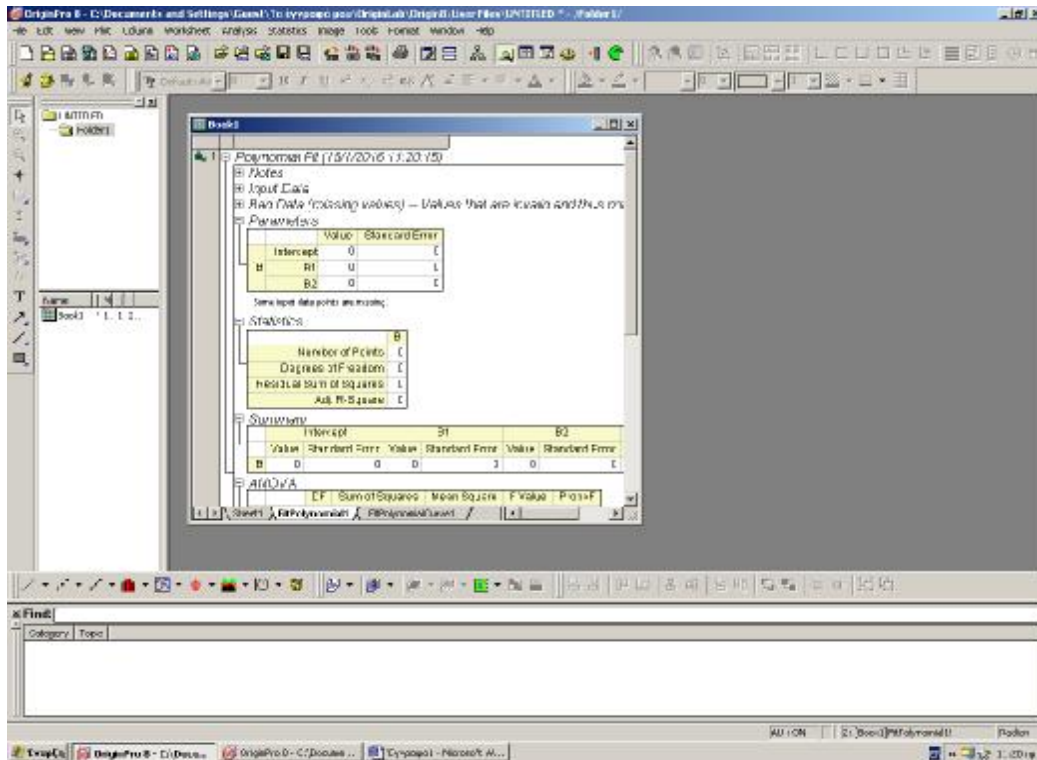
Ο βαθμός του πολωνύμου είναι αυτός που καθορίζει την ακρίβεια που θα έχει η πολυωνυμική συνάρτηση. Η επιλογή βαθμού γίνεται με το Polynomial Order.



Εικ.48 Επιλογή βαθμού του πολυωνύμου

Επιλέγοντας OK ανοίγει η σελίδα FitPolynomial (μέσα στο Book1). Ο πρώτος πίνακας που εμφανίζεται είναι ο πίνακας Parameters. Στη στήλη Value δίνονται οι συντελεστές του πολυωνύμου (B1,B2 αν είναι δευτέρου βαθμού), καθώς επίσης ο σταθερός όρος (intercept). Στη στήλη Standard Error δίνεται το σφάλμα που έχει υπολογίσει το πρόγραμμα για τους συντελεστές του πολυωνύμου.

Στον επόμενο πίνακα με το όνομα Statistics και στη γραμμή Adj.R-Square δίνεται η ακρίβεια του πολυωνύμου. Βέλτιστο θα είναι το πολυώνυμο που η ακρίβεια του τείνει στη μονάδα. Μεταβάλλοντας διαδοχικά το βαθμό του πολυωνύμου με το Polynomial Order, μπορούμε να βρούμε ποιο είναι το καταλληλότερο πολυώνυμο για κάθε αισθητήρα.



Εικ.49 Οι υπολογισμοί του προγράμματος ORIGIN

Αφού ολοκληρωθεί η διαδικασία εύρεσης της βέλτιστης πολυωνμικής συνάρτησης το πολυώνυμο που προκύπτει θα περιληφθεί στον κώδικα του αισθητήρα στο sketch του Arduino.

```

if( vroxi <= 490){
  mmvroxi=0.01*vroxi;
}

if( vroxi >= 490 && vroxi < 600){
mmvroxi= -
0.000000333631*vroxi*vroxi*vroxi*vroxi+0.0000772554*vroxi*vroxi*vroxi-
0.06474*vroxi*vroxi+23.63446*vroxi-3197.62725;
}

if( vroxi >= 600){
  mmvroxi=40;
}

```

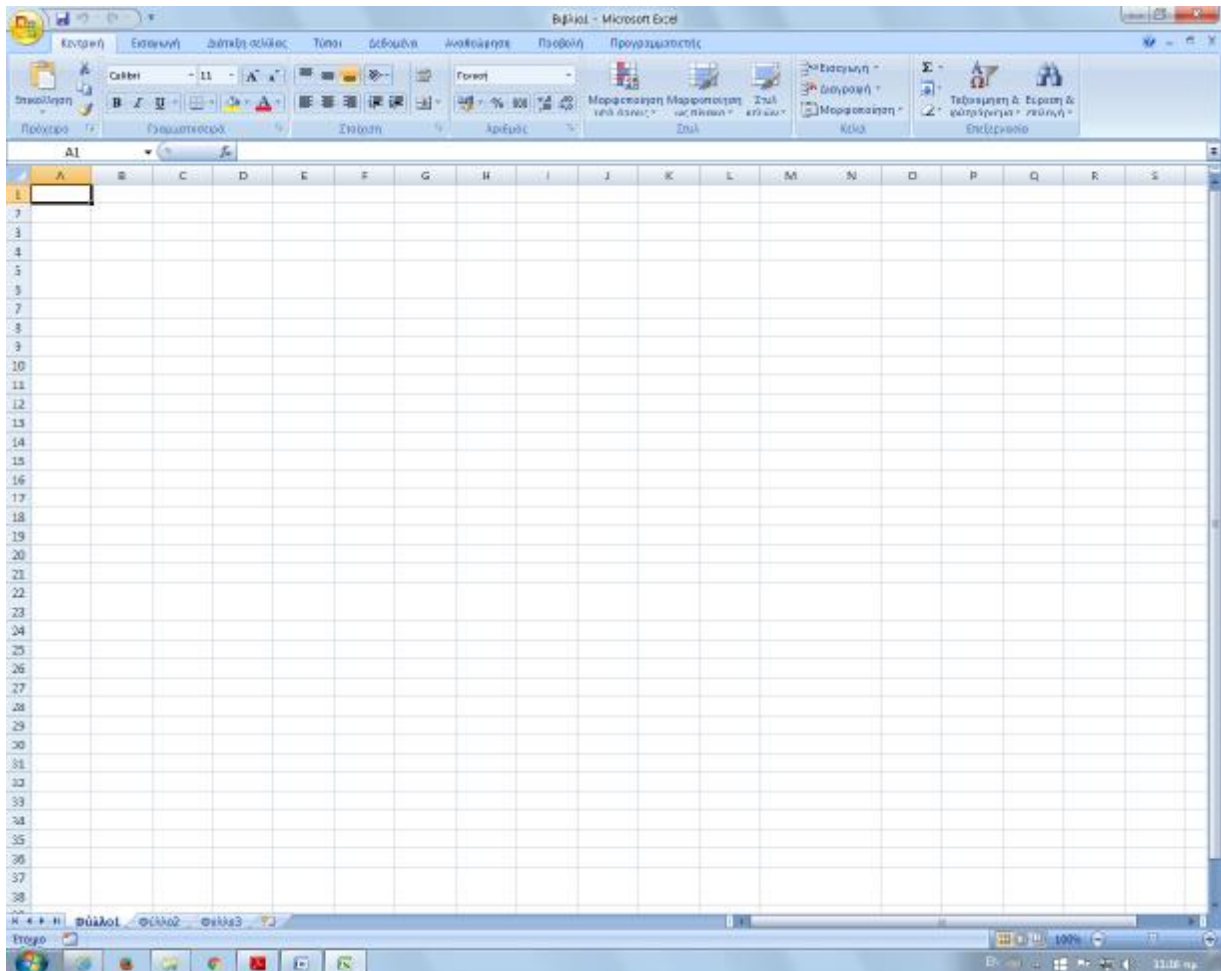
Εικ.50 Το πολυώνυμο για τον υπολογισμό του ύψους βροχής (4<sup>ο</sup> βαθμού)

### 3.3 ΑΠΟΣΤΟΛΗ ΔΕΔΟΜΕΝΩΝ ΣΕ ΑΡΧΕΙΟ EXCEL

Μετά την ολοκλήρωση της διαδικασίας προγραμματισμού του Arduino, ακολουθεί η αποστολή των δεδομένων σε αρχείο Excel[19]. Η αποστολή γίνεται σειριακά, με τη χρήση κώδικα σε Visual Basic. Το Excel είναι ένα μέρος του πακέτου Microsoft Office το οποίο αποτελεί ένα διαδεδομένο πρόγραμμα λογιστικών φύλλων. Παρέχει δυνατότητες αριθμητικών υπολογισμών, γραφικών παραστάσεων, διαγραμμάτων καθώς επίσης και δυνατότητες προγραμματισμού. Η έκδοση που χρησιμοποιήθηκε είναι το Microsoft Excel 2007.

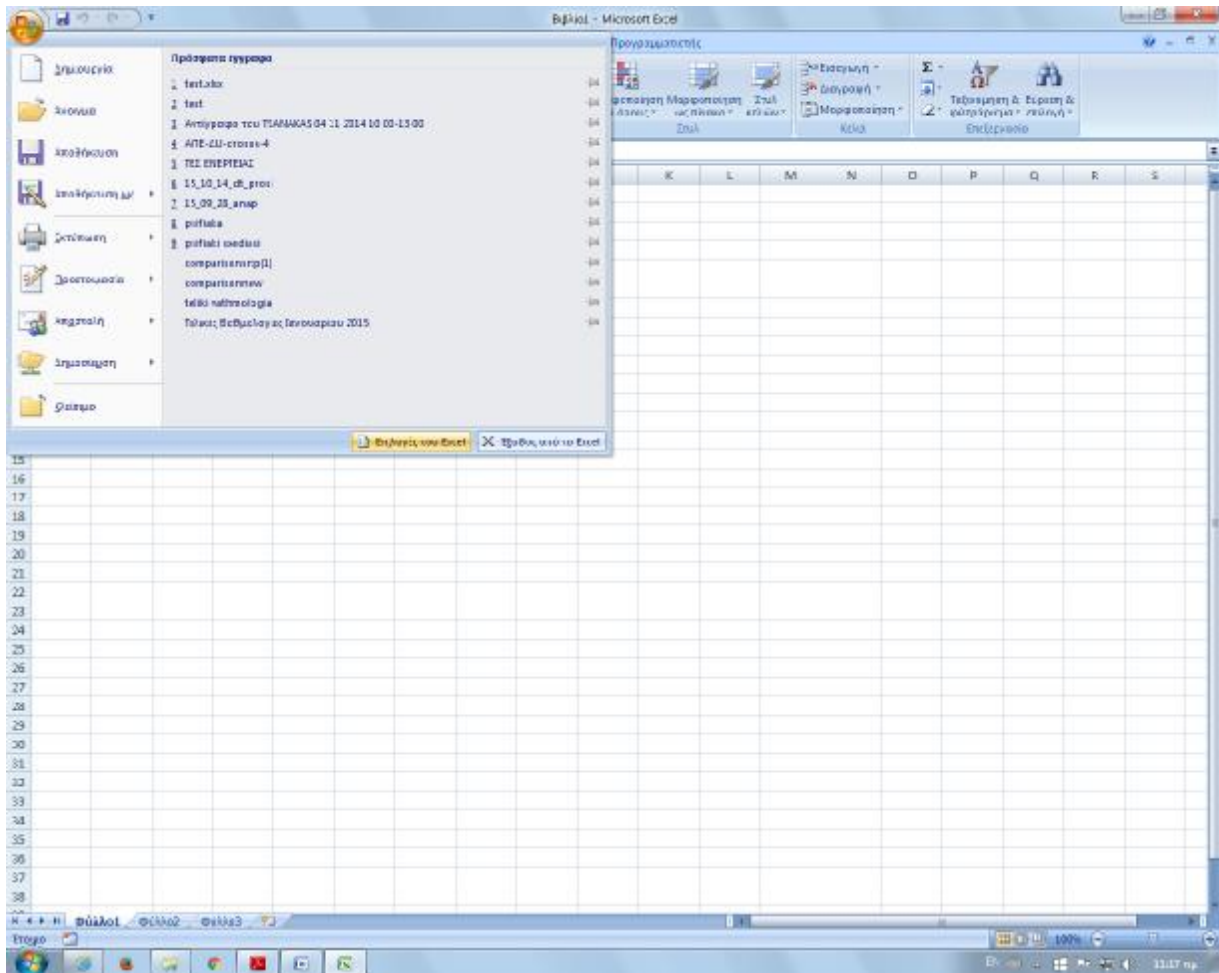
Τα βήματα για την αποστολή των δεδομένων από το Arduino στο Excel είναι η ακόλουθη :

Αρχικά ανοίγουμε το Excel και επιλέγουμε το Office Button το οποίο βρίσκεται πάνω αριστερά στο παράθυρο.



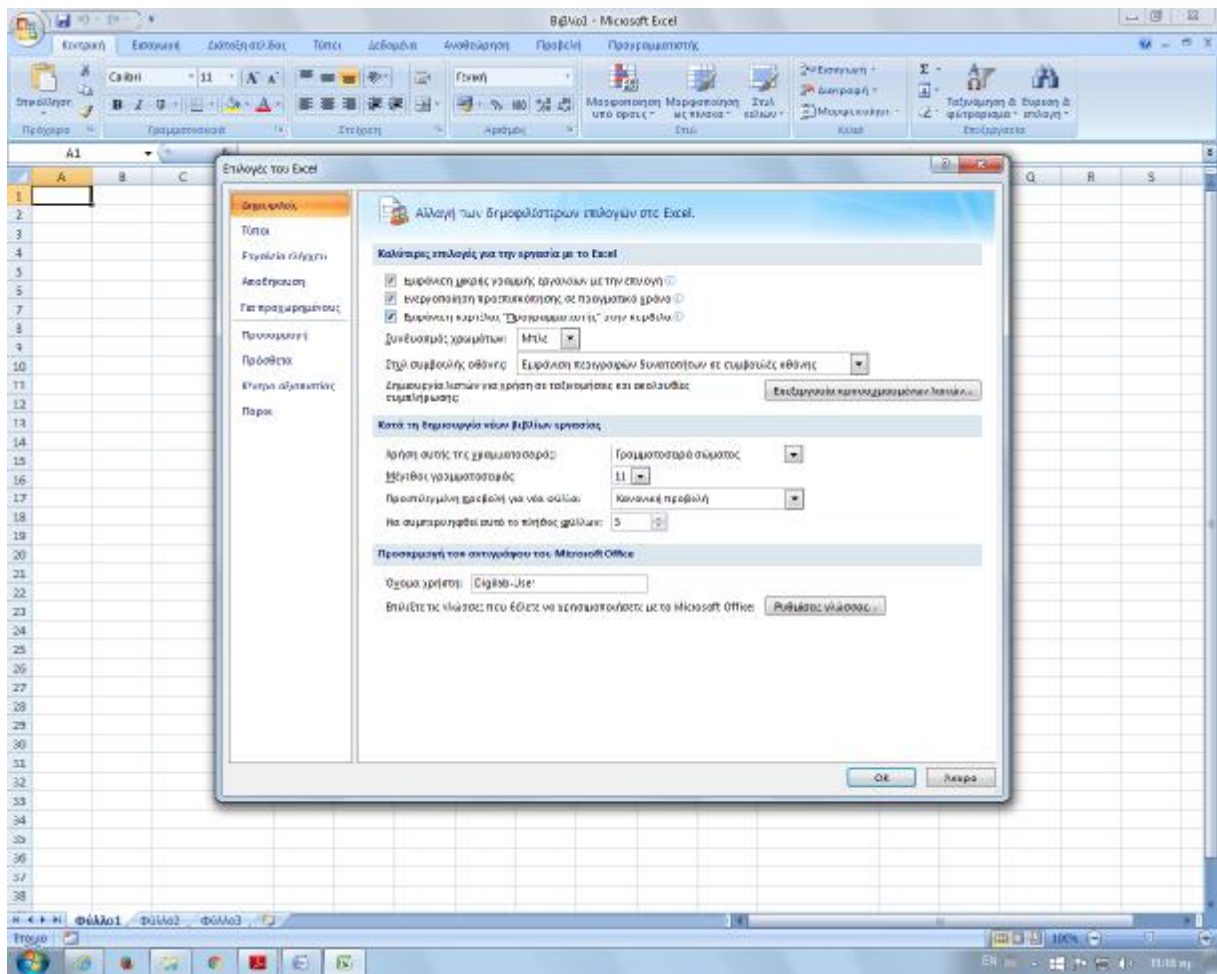
*Εικ.51 Το περιβάλλον του Excel*

Στο παράθυρο που ανοίγει πηγαίνουμε στο «Επιλογές του Excel» (ή Excel Options) που βρίσκεται στο κάτω μέρος του νέου παραθύρου και το επιλέγουμε.



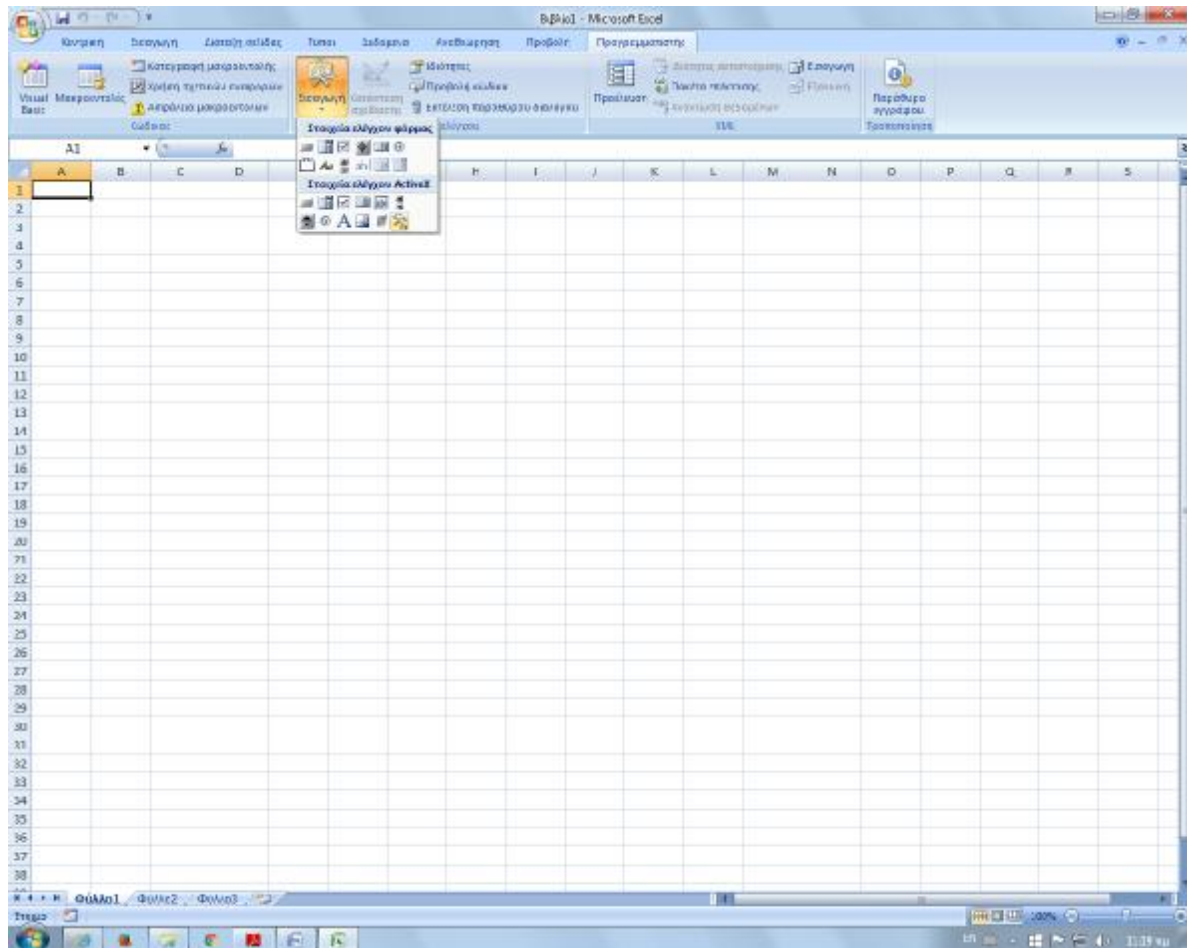
Εικ.52 Επιλογές του Excel

Τότε ανοίγει το παράθυρο Επιλογές του Excel και στην καρτέλα Δημοφιλείς (ή Popular) επιλέγουμε το «Εμφάνιση καρτέλας "Προγραμματιστής" στην κορδέλα» (ή Show Developer Tab) και πατάμε το OK.



Εικ.53 Εμφάνιση της καρτέλας Προγραμματιστής

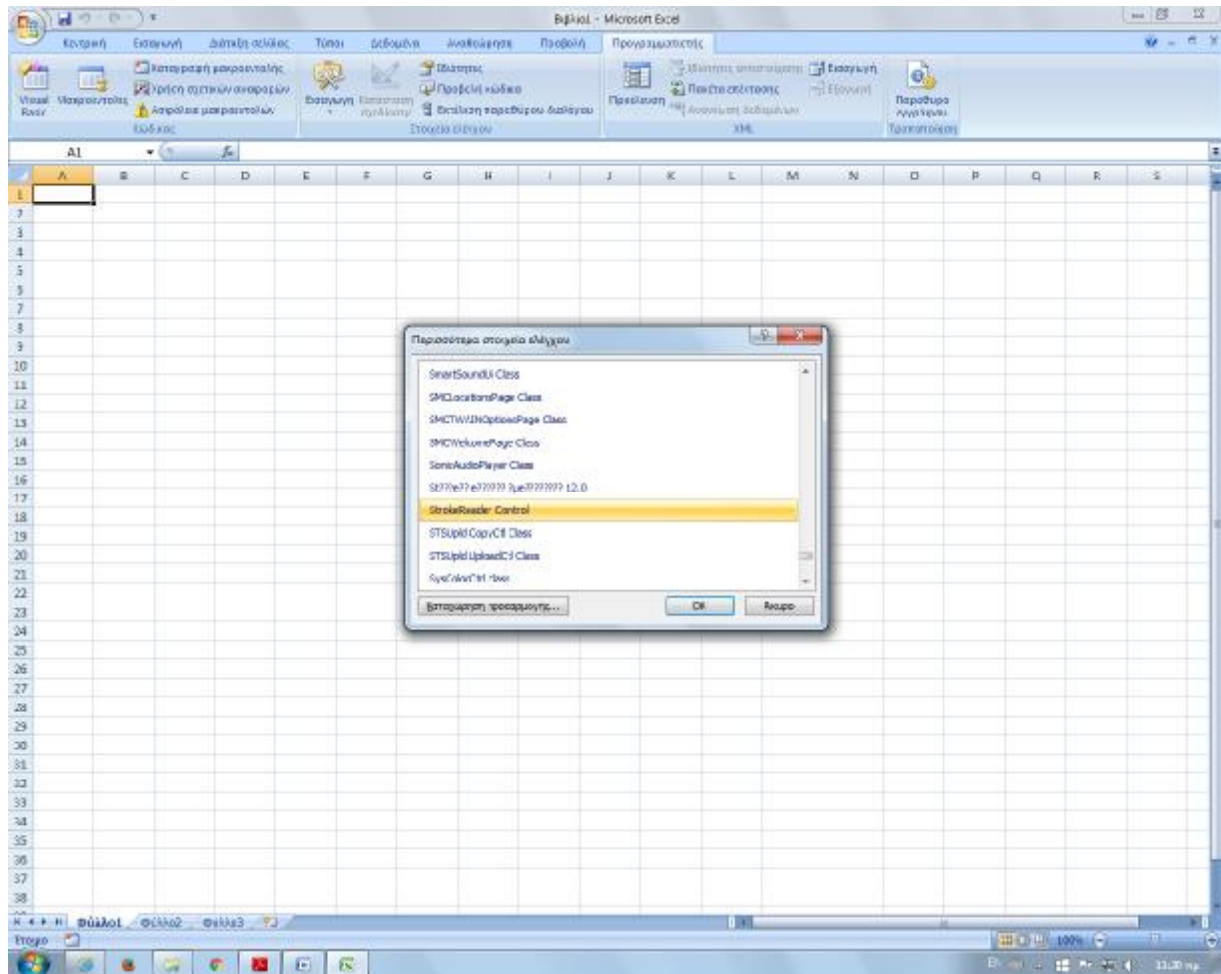
Στη συνέχεια πηγαίνουμε στην καρτέλα Προγραμματιστής → Εισαγωγή → Στοιχεία ελέγχου ActiveX → Περισσότερα στοιχεία ελέγχου



Εικ.54 Εισαγωγή στοιχείου ελέγχου ActiveX

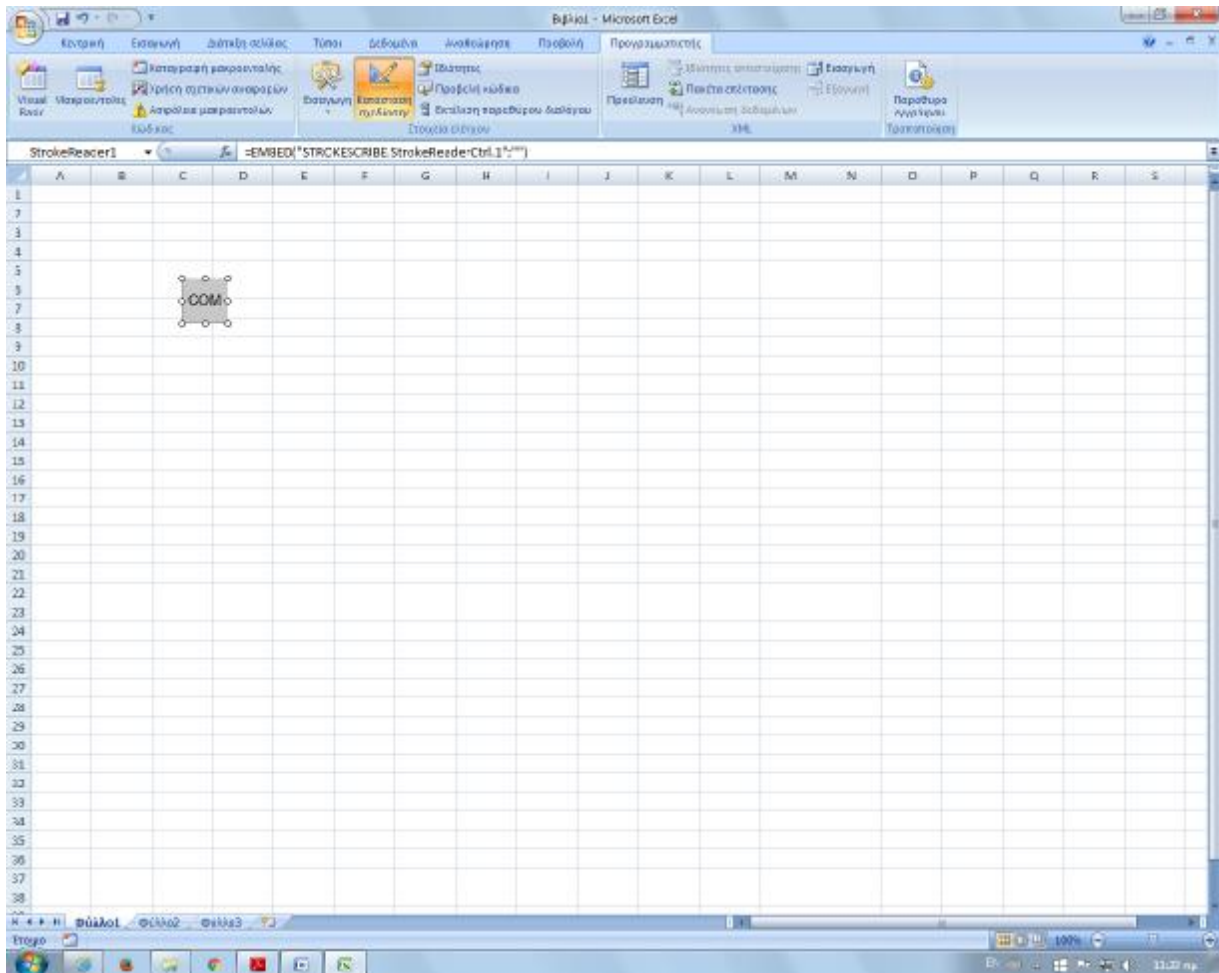


Στο παράθυρο Περισσότερα στοιχεία ελέγχου επιλέγουμε StrokeReader Control και πατάμε OK. Σε περίπτωση που δεν εμφανίζεται η επιλογή StrokeReader Control θα πρέπει να την κατεβάσουμε. Παρέχεται δωρεάν σε διάφορες ιστοσελίδες όπως για παράδειγμα στην ιστοσελίδα <http://strokereader.soft112.com/>



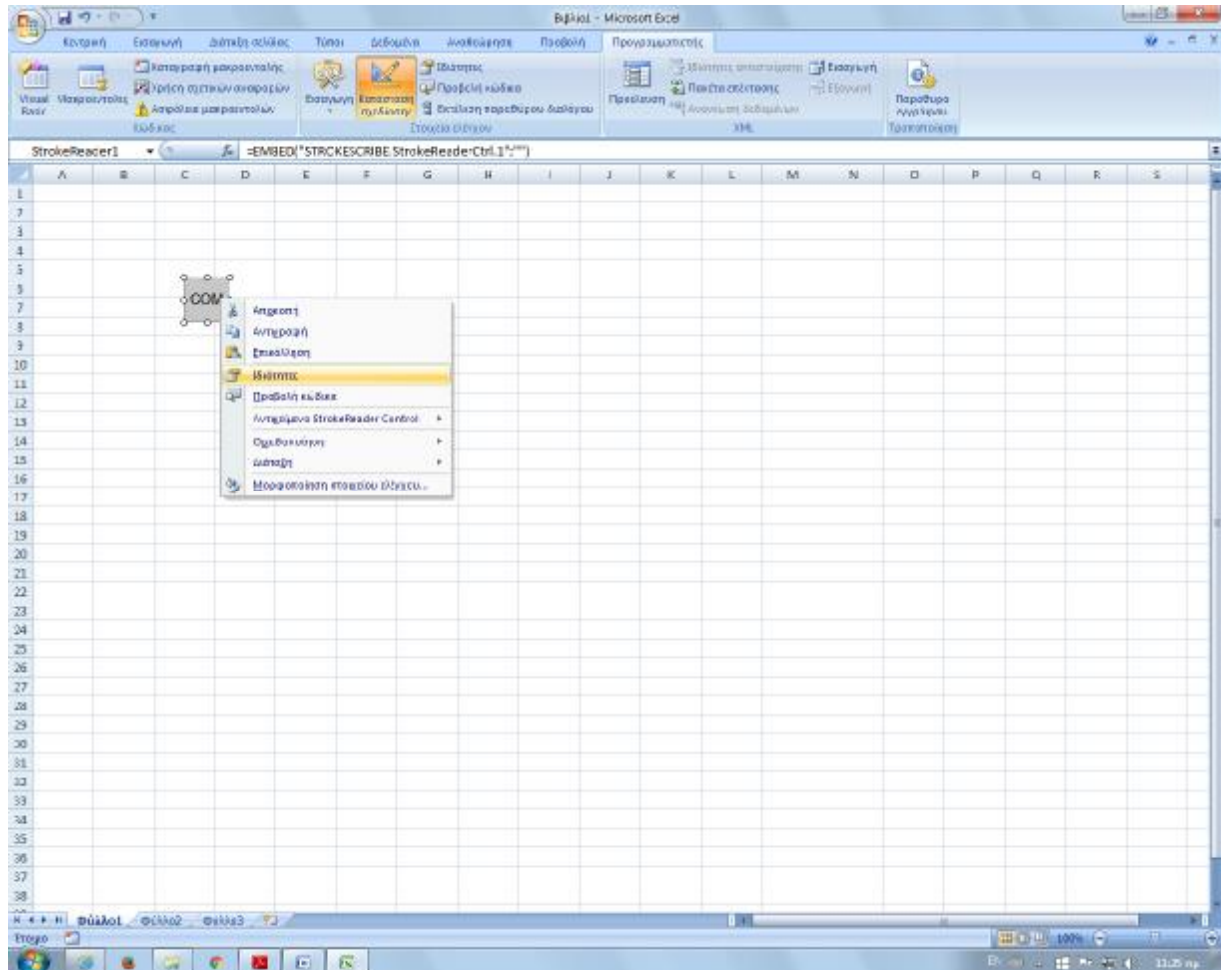
Εικ.55 Επιλογή StrokeReader Control

Κάνουμε αριστερό κλικ σε οποιοδήποτε σημείο του φύλλου εργασίας για να τοποθετηθεί το serial port ActiveX. Αυτό εμφανίζεται με τη μορφή ενός τετραγώνου με το όνομα COM.



Εικ.56 Εισαγωγή στοιχείου ActiveX

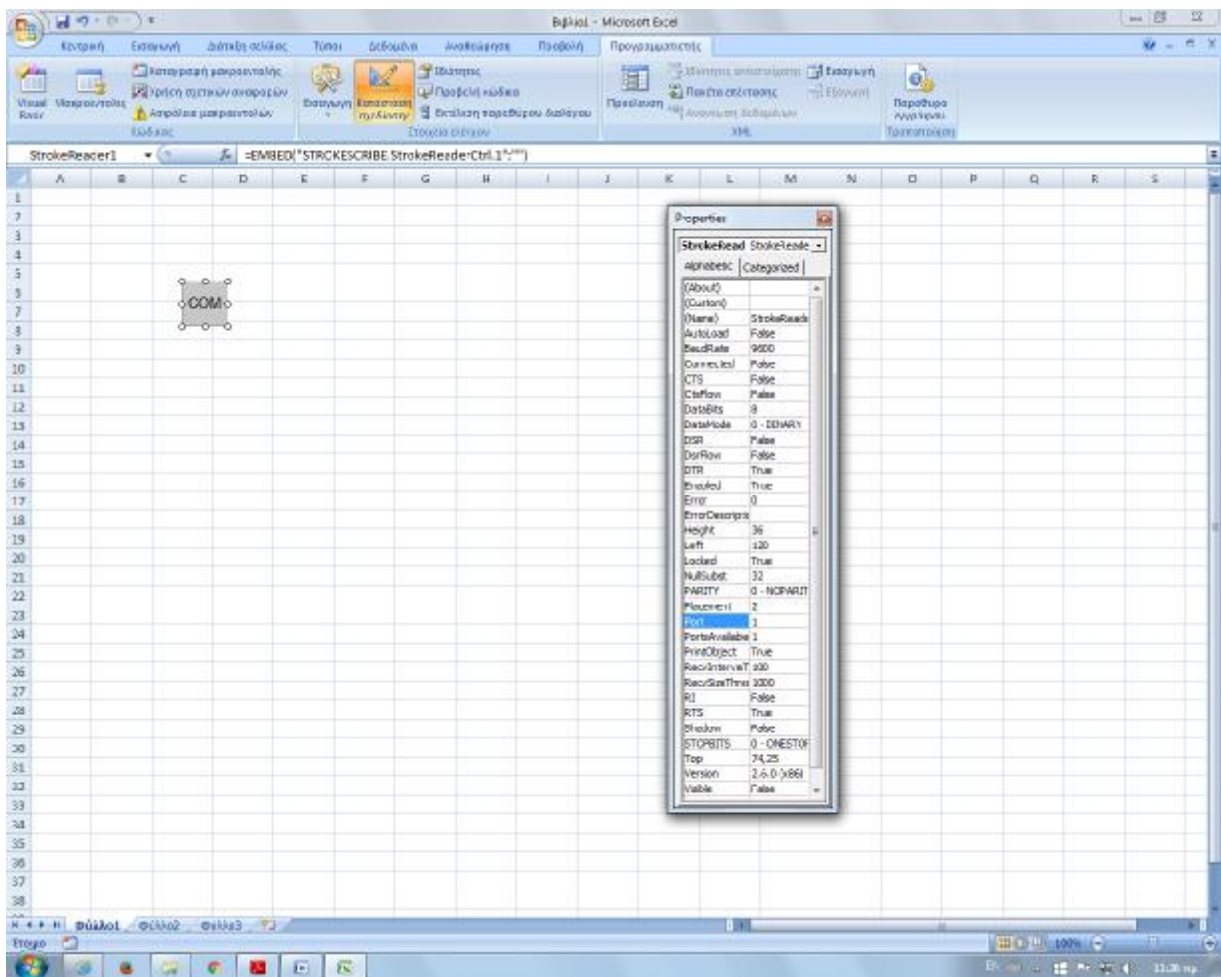
Κάνοντας δεξί κλικ στο serial port και επιλέγοντας Ιδιότητες (ή Properties) ανοίγει το αντίστοιχο παράθυρο στο οποίο μπορούμε να κάνουμε τις απαραίτητες ρυθμίσεις.



Εικ.57 Άνοιγμα παραθύρου «Ιδιότητες»

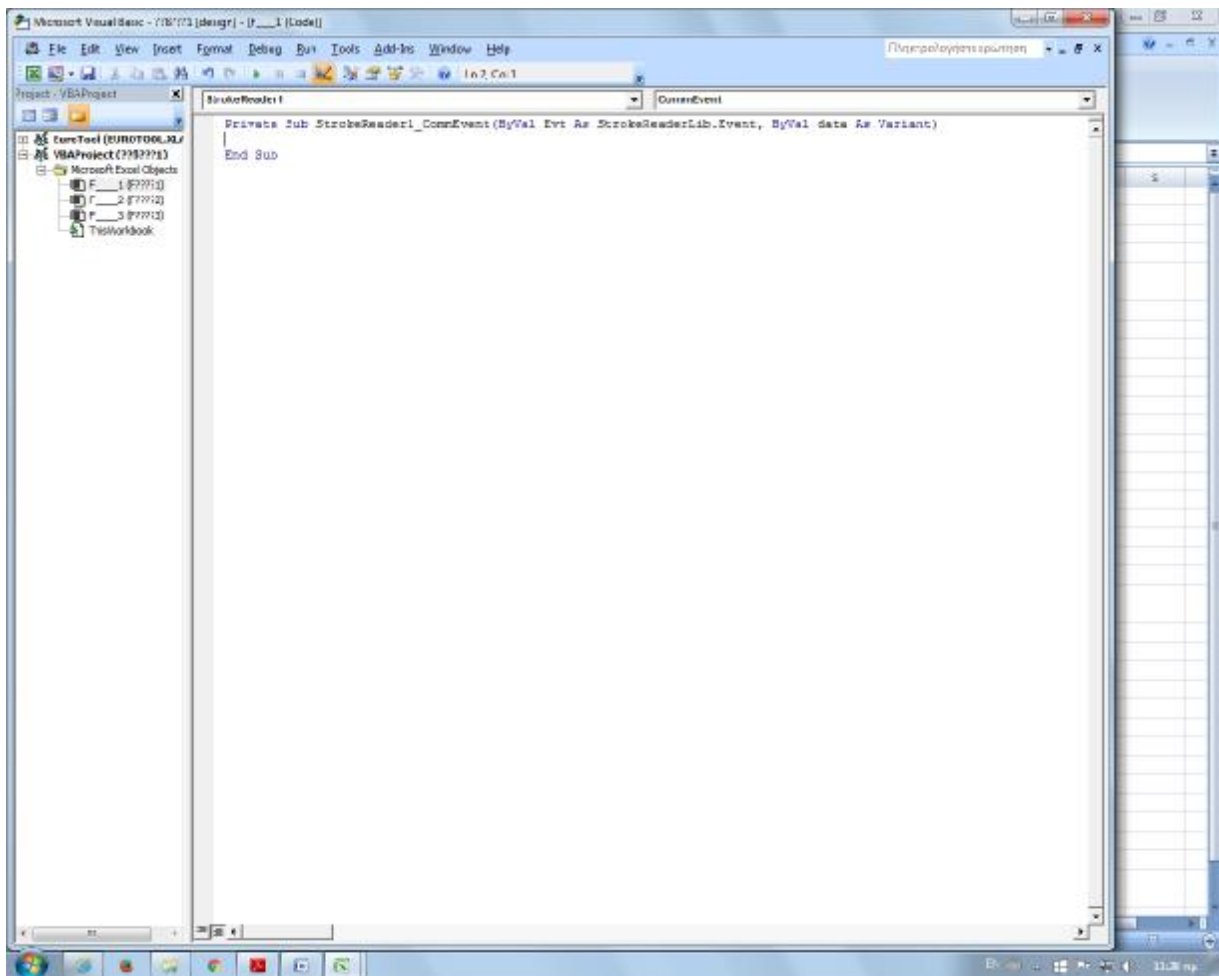
Οι παράμετροι που πρέπει να ρυθμίσουμε είναι οι εξής : Port, BaudRate, DataBits, PARITY, STOPBITS, Connected.

- Port : Ο αριθμός της θύρας που χρησιμοποιείται από το Arduino
- BaudRate : Ο ρυθμός μετάδοσης των δεδομένων (συνήθως παίρνει την τιμή 9600)
- DataBits : Ο αριθμός των bits ανά byte (θέτουμε την τιμή 8)
- PARITY : Συνήθως ρυθμίζεται στο NOPARITY
- STOPBITS : Το πλήθος των stop bits που αποστέλλονται μετά από κάθε byte (θέτουμε την τιμή 1)
- Connected : Το ρυθμίζουμε στο True για να συνδεθεί με τη σειριακή θύρα. Το ActiveX θα συνδεθεί αυτόματα με το USB όταν αυτό τοποθετηθεί στη θύρα του υπολογιστή.



Εικ.58 Ρύθμιση παραμέτρων

Κάνοντας διπλό κλικ στο τετράγωνο COM ανοίγει το παράθυρο Microsoft Visual Basic[20] στο οποίο θα γράψουμε τον κώδικα για να ολοκληρωθεί η διαδικασία αποστολής δεδομένων στο Excel από το Arduino. Η Visual Basic είναι μία τρίτης γενιάς γλώσσα προγραμματισμού και πρόθεση της Microsoft ήταν να δημιουργήσει μία γλώσσα εύκολη στη χρήση. Προήλθε από την BASIC και επιτρέπει την ταχεία ανάπτυξη εφαρμογών (RAD), πρόσβαση σε βάσεις δεδομένων με τη χρήση των Data Access Objects και την δημιουργία αντικειμένων ActiveX.



Εικ.59 Το περιβάλλον της Microsoft Visual Basic

Ο κώδικας που χρησιμοποιήθηκε για την αποστολή των δεδομένων από το Arduino στο serial port του Excel δίνεται αναλυτικά στο παράρτημα. Πληροφορίες για την δημιουργία του κώδικα δίνονται στην ιστοσελίδα <https://strokescribe.com/en/read-serial-port-excel-2007.html>. Η μετατροπή που έγινε στον κώδικα είναι η προσθήκη της ημερομηνίας και ώρας στην πρώτη στήλη του φύλλου Excel με τη χρήση της εντολής Cells(cell\_idx + 1, 1) = Now(). Τα δεδομένα που αποστέλλονται σειριακά περνάνε στην επόμενη στήλη του φύλλου εργασίας.

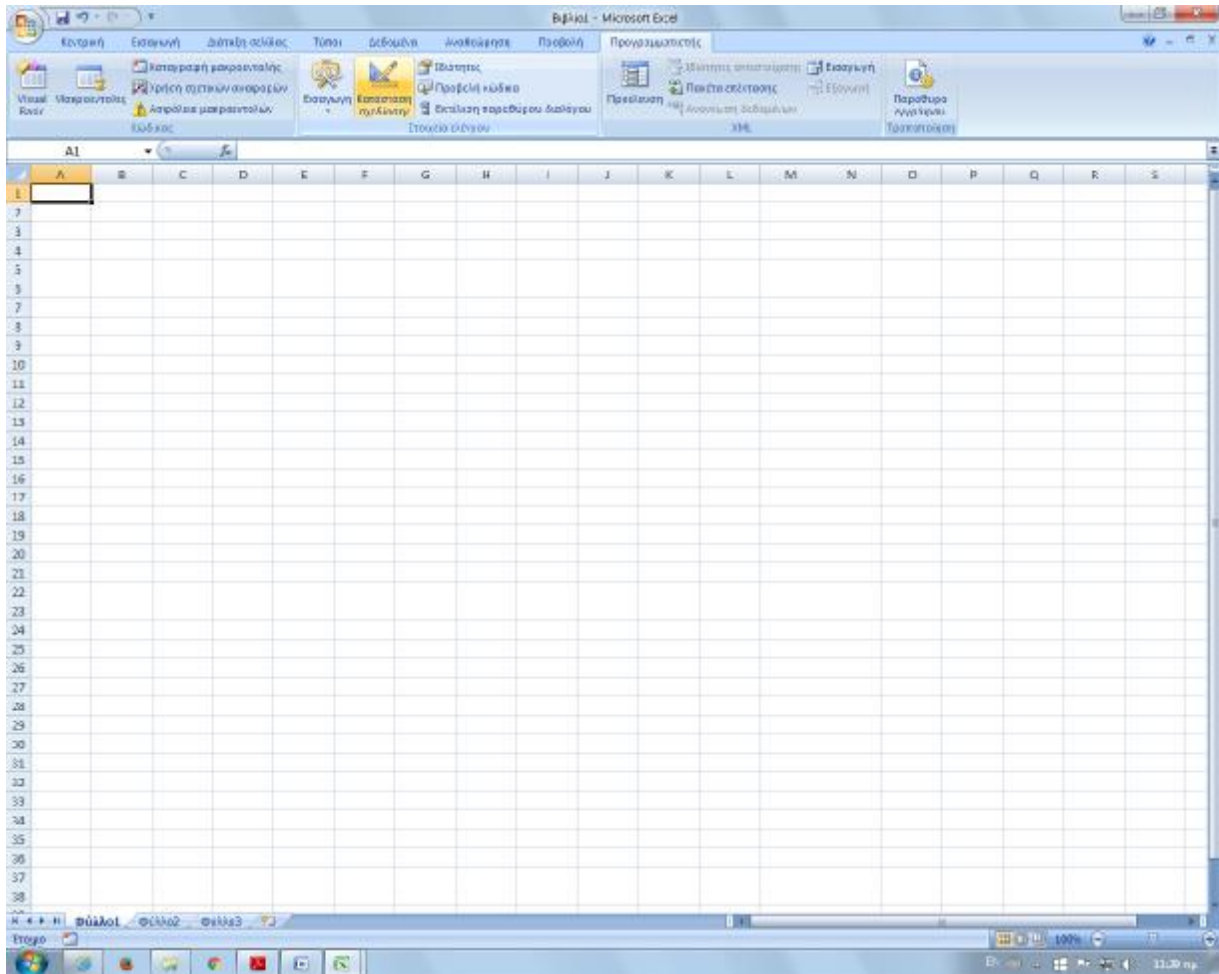
```

Dim buf As String 'Buffer for incoming serial data
Dim cell_idx As Integer 'The row number cell to store a received barcode
Private Sub StrokeReader1_ConnEvent(ByVal Evt As StrokeReaderLib.Event, _
    ByVal data As Variant)
    Select Case Evt 'Can be EVT_DISCONNECT or EVT_DATA or EVT_SERIAL_READY
        Case EVT_DISCONNECT 'if USB serial port adapter is just disconnected from the PC
            MsgBox "Serial port converter disconnected"
        Case EVT_CONNECT
            MsgBox "Connected" 'if RS232 or RS485 port adapter is just connected to USB bus
        Case EVT_DATA 'Incoming serial data
            buf = buf + StrokeReader1.Read(Text) 'Incoming serial data accumulated in buf
            Do
                LF = InStr(buf, Chr(10)) 'Most of scanners send ASCII <LF> after each barcode
                If LF = 0 Then Exit Do 'Waiting for ASCII <LF> to be received
                s = Left(buf, LF - 1) 'Copy the data before <LF> code as the barcode text
                s = Replace(s, Chr(13), "") 'Some scanners could send <CR><LF>. Remove <CR> if present
                buf = Right(buf, Len(buf) - LF) 'Cut the parsed data from the buffer
                'This is useful if the external hardware sends multiple barcodes in a time.
                cell_idx = cell_idx + 1 'increment the row number of cell where the barcode will be stored
                Cells(cell_idx + 1, 2) = s
                Cells(cell_idx + 1, 1) = Now()
            Loop
        End Select
    End Sub

```

Εικ.60 Ο κώδικας που χρησιμοποιήθηκε σε Visual Basic

Το τελευταίο βήμα της διαδικασίας είναι η επιστροφή στο παράθυρο του Excel και η επιλογή της Κατάστασης σχεδίασης (ή Design Mode). Με τον τρόπο αυτό απενεργοποιείται η Κατάσταση σχεδίασης και εκκρεμεί η αποστολή των μετεωρολογικών δεδομένων στο φύλλο εργασίας.



*Εικ.61 Απενεργοποίηση της επιλογής «Κατάσταση Σχεδίασης»*





## ΚΕΦΑΛΑΙΟ 4<sup>ο</sup>

### ΕΦΑΡΜΟΓΕΣ – ΣΥΜΠΕΡΑΣΜΑΤΑ

Το σύστημα που υλοποιήθηκε στην παρούσα εργασία δίνει τη δυνατότητα στο χρήστη να καταγράφει και να αποθηκεύει τα μετεωρολογικά δεδομένα που συλλέγονται από τους αισθητήρες. Στη συνέχεια παρουσιάζονται κάποιες πιθανές εφαρμογές του συστήματος, καθώς και τα συμπεράσματα στα οποία καταλήξαμε.

#### 4.1 ΕΦΑΡΜΟΓΕΣ ΣΤΗ ΓΕΩΡΓΙΑ

Το σύστημα που υλοποιήθηκε μπορεί να βρει εφαρμογή σε διάφορους τομείς. Ένας από αυτούς είναι ο τομέας της γεωργίας. Το σύστημα παρέχει τη δυνατότητα στο χρήστη να συλλέγει σε καθημερινή βάση όλες τις απαραίτητες πληροφορίες (όπως η θερμοκρασία, η υγρασία του εδάφους κλπ) με πολύ χαμηλό κόστος. Επιπλέον, δίνεται η δυνατότητα αποθήκευσης των δεδομένων αυτών τα οποία μπορούν να αξιοποιηθούν για την πρόβλεψη ασθενειών στις καλλιέργειες. Με τον τρόπο αυτό και με τις κατάλληλες ενέργειες από την πλευρά του χρήστη, μπορεί να βελτιωθεί σημαντικά η διαχείριση των καλλιεργειών και να ελαχιστοποιηθεί η κατανάλωση νερού και ενέργειας. Η χρήση των αισθητήρων και των νέων τεχνολογιών γενικότερα, βρίσκουν σταδιακά εφαρμογή στη γεωργία. Η γεωργία ακριβείας[21] θεωρείται το μελλοντικό σύστημα παραγωγής προϊόντων, αν και στην Ελλάδα βρίσκεται ακόμη σε ερευνητικό επίπεδο. Είναι μία σύγχρονη μέθοδος καλλιέργειας η οποία συνδυάζει την συλλογή δεδομένων και πληροφοριών με τις πραγματικές ανάγκες της καλλιέργειας, τόσο χρονικά όσο και χωρικά. Αποτέλεσμα των διεργασιών αυτών είναι η αύξηση της απόδοσης της παραγωγής των αγροτικών προϊόντων, η βελτίωση της ποιότητας τους και παράλληλα η ελαχιστοποίηση της περιβαλλοντικής επιβάρυνσης και του κόστους των λειτουργικών εξόδων. Η τεχνολογία στην οποία βασίζεται κυρίως η γεωργία ακριβείας είναι τα ασύρματα δίκτυα αισθητήρων.



Εικ.63 Γεωργία ακριβείας



*Εικ.64 Μετεωρολογικός σταθμός δίπλα σε καλλιέργεια*

Μία ακόμη εφαρμογή σχετική με τη γεωργία που θα μπορούσε να βρει το σύστημα είναι σε θερμοκήπιο. Οι τιμές που καταγράφονται από τους αισθητήρες μπορούν να δώσουν μια ακριβή εικόνα για τις ανάγκες του χώρου και με κατάλληλο προγραμματισμό και χρήση αυτοματισμών μπορούμε να προγραμματίσουμε εντολές όπως το πότισμα ανάλογα με την σχετική υγρασία του εδάφους, η ρύθμιση της θερμοκρασίας με κατάλληλο σύστημα θέρμανσης, η προσαρμογή της έντασης φωτισμού με τη χρήση περσίδων και η ρύθμιση της σχετικής υγρασίας με τη χρήση αεραγωγών.



*Εικ.65 Θερμοκήπιο με αισθητήρες*

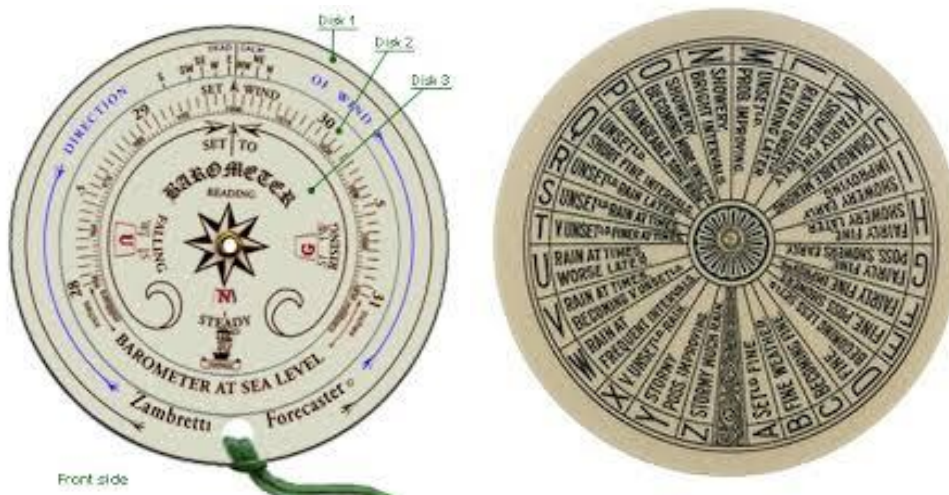
## 4.2 ΠΡΟΓΝΩΣΗ ΚΑΙΡΟΥ

### Ο ΑΛΓΟΡΙΘΜΟΣ ZAMBRETTI

Ο αλγόριθμος Zambretti[22] χρησιμοποιείται για την πρόγνωση του καιρού με τη βοήθεια ενός βαρομέτρου. Σχεδιάστηκε το 1915 από τους Άγγλους Νεργέτι και Ζάμπρα. Η ακρίβεια του οργάνου ξεπερνά το 90% για χρονική διάρκεια 12 ωρών και αφορά την πρόγνωση του καιρού στο Βόρειο Ημισφαίριο.

Αποτελείται από τρεις πλαστικούς δίσκους με φθίνουσα ακτίνα τοποθετημένους ομοκεντρικά, εκ των οποίων ο πρώτος σχετίζεται με τη διεύθυνση του ανέμου, ο δεύτερος έχει βαθμονομημένη κλίμακα για την βαρομετρική πίεση σε ίντσες υδραργύρου ή mbars και τη διεύθυνση του ανέμου και ο τρίτος διαθέτει τρία παράθυρα για την επιλογή του κατάλληλου γράμματος που εξαρτάται από την βαρομετρική πίεση το χειμώνα ή το καλοκαίρι. Η πρόγνωση που εμφανίζεται στην πίσω όψη του δίσκου συνδέεται με τα γράμματα του αγγλικού αλφάβητου.

Κάνοντας λοιπόν χρήση του αισθητήρα βαρομετρικής πίεσης, παρέχεται η δυνατότητα βραχυπρόθεσμης πρόγνωσης του καιρού με αρκετά μεγάλη ακρίβεια. Εφαρμογή του αλγορίθμου Zambretti με τη βοήθεια εξομοιωτή Java μπορεί να γίνει στην ιστοσελίδα <http://weather.vouhead.gr/wxforecast-zambretti.php>[23].



Εικ.66 Ο δίσκος Zambretti

Πρόγνωση καιρού
A : Αίθριος αμετάβλητος
B : Αίθριος
C : Καιρός σε βελτίωση
D : Αίθριος μεταβαλλόμενος
E : Αίθριος, πιθανές βροχοπτώσεις
F : Σχεδόν αίθριος, σε βελτίωση
G : Σχεδόν αίθριος, νωρίς πιθανές βροχοπτώσεις
H : Σχεδόν αίθριος, αργότερα πιθανές βροχοπτώσεις
I : Βροχοπτώσεις νωρίς, σε βελτίωση
J : Μεταβλητός, σε βελτίωση
K : Σχεδόν αίθριος, πιθανές βροχοπτώσεις
L : Μάλλον άστατος, αργότερα αίθριος
M : Άστατος, πιθανή βελτίωση
N : Βροχοπτώσεις, κατά περιόδους αίθριος
O : Βροχοπτώσεις, στη συνέχεια πιο άστατος
P : Μεταβλητός, σποραδικές βροχές
Q : Άστατος, κατά μικρές περιόδους αίθριος
R : Άστατος, αργότερα βροχή
S : Άστατος, σποραδικές βροχές
T : Πολύ άστατος, κατά περιόδους βελτίωση
U : Σποραδικές βροχές, αργότερα επιδείνωση
V : Σποραδικές βροχές, στη συνέχεια πολύ άστατος
W : Βροχή κατά τακτές περιόδους
X : Πολύ άστατος, βροχοπτώσεις
Y : Θυελλώδης, πιθανή βελτίωση
Z : Θυελλώδης, έντονη βροχή

*Εικ.67 Η πρόγνωση του καιρού με βάση το αγγλικό αλφάβητο*

Ανάλογα με την κατάσταση της βαρομετρικής τάσης χρησιμοποιείται και η αντίστοιχη συνάρτηση :

1. Για φθίνουσα βαρομετρική τάση  $Z_f = 130 - P/81$
2. Για σταθερή βαρομετρική τάση  $Z_s = 147 - 5P/376$
3. Για αύξουσα βαρομετρική τάση  $Z_r = 179 - 2P/129$

όπου P η τιμή της βαρομετρικής πίεσης.

Η τιμή του Z που προκύπτει είναι ένας ακέραιος αριθμός, ο οποίος δίνει την πρόβλεψη του καιρού με τη βοήθεια του παρακάτω πίνακα.

Αριθμός πρόγνωσης Z	Γράμμα πρόγνωσης δίσκου Zambretti	Πρόγνωση καιρού	Ατμοσφ. πίεση
1	A	Αίθριος αμετάβλητος	1050
2	B	Αίθριος	1040
3	D	Αίθριος μεταβαλλόμενος	1024
4	H	Σχεδόν αίθριος, αργότερα πιθανές βροχοπτώσεις	1018
5	O	Βροχοπτώσεις, στη συνέχεια πιο άστατος	1010
6	R	Άστατος, αργότερα βροχή	1004
7	U	Σποραδικές βροχές, αργότερα επιδείνωση	998
8	V	Σποραδικές βροχές, στη συνέχεια πολύ άστατος	991
9	X	Πολύ άστατος, βροχοπτώσεις	985
10	A	Αίθριος αμετάβλητος	1033
11	B	Αίθριος	1023
12	E	Αίθριος, πιθανές βροχοπτώσεις	1014
13	K	Σχεδόν αίθριος, πιθανές βροχοπτώσεις	1008
14	N	Βροχοπτώσεις, κατά περιόδους αίθριος	1000
15	P	Μεταβλητός, σποραδικές βροχές	994
16	S	Άστατος, σποραδικές βροχές	989
17	W	Βροχή κατά τακτές περιόδους	981
18	X	Πολύ άστατος, βροχοπτώσεις	974
19	Z	Θυελλώδης, έντονη βροχή	960
20	A	Αίθριος αμετάβλητος	1030
21	B	Αίθριος	1022
22	C	Καιρός σε βελτίωση	1012
23	F	Σχεδόν αίθριος, σε βελτίωση	1007
24	G	Σχεδόν αίθριος, νωρίς πιθανές βροχοπτώσεις	1000
25	I	Βροχοπτώσεις νωρίς, σε βελτίωση	995
26	J	Μεταβλητός, σε βελτίωση	990
27	L	Μάλλον άστατος, αργότερα αίθριος	984
28	M	Άστατος, πιθανή βελτίωση	978
29	Q	Άστατος, κατά μικρές περιόδους αίθριος	970
30	T	Πολύ άστατος, κατά περιόδους βελτίωση	965
31	Y	Θυελλώδης, πιθανή βελτίωση	959
32	Z	Θυελλώδης, έντονη βροχή	947

### Γενικές παρατηρήσεις

Σημειώνεται πως σε χαμηλές θερμοκρασίες και με το κατάλληλο σημείο δρόσου, ο υετός ενδέχεται να είναι χιόνι.

Γενικά εκτιμάται πως οι βόρειοι άνεμοι βελτιώνουν τις προοπτικές του καιρού μέχρι 1 μονάδα, η οποία αφαιρείται από την τιμή του  $Z$ .

Αντίθετα, οι νότιοι άνεμοι επιδεινώνουν τις προοπτικές του καιρού μέχρι 2 μονάδες συνεπώς θεωρείται απαραίτητη η αντίστοιχη προσαρμογή του  $Z$ .

Απαραίτητη είναι επίσης η ρύθμιση για μετρήσεις το χειμώνα ή το καλοκαίρι. Γενικά, ο χειμώνας αντιστοιχεί σε μείωση του  $Z$  κατά 1 μονάδα, ενώ το καλοκαίρι σε αύξηση κατά 1 μονάδα.

Από τα παραπάνω συμπεραίνουμε πως για την πρόγνωση του καιρού με τη χρήση βαρομέτρου σύμφωνα με τον αλγόριθμο Zambretti, απαιτούνται τα παρακάτω βήματα :

1. Ο καθορισμός της βαρομετρικής τάσης
2. Η χρήση της κατάλληλης συνάρτησης  $Z_f$ ,  $Z_s$ ,  $Z_r$
3. Μέτρηση της τρέχουσας τιμής της πίεσης
4. Υπολογισμός του  $Z$  με τη βοήθεια της κατάλληλης εξίσωσης
5. Ρύθμιση του  $Z$  ανάλογα με την κατεύθυνση του ανέμου (αν υπάρχουν στοιχεία)
6. Ρύθμιση του  $Z$  ανάλογα με την εποχή
7. Χρήση του ρυθμισμένου  $Z$  για πρόγνωση του καιρού με τη βοήθεια του πίνακα.

### 4.3 ΣΥΜΠΕΡΑΣΜΑΤΑ

Στην παρούσα πτυχιακή εργασία έγινε περιγραφή της υλοποίησης και λειτουργίας ενός συστήματος ανάλυσης μετεωρολογικών μετρήσεων μέσω αισθητήρων με τη χρήση της πλατφόρμας Arduino.

Τα συμπεράσματα που προκύπτουν είναι πως με πολύ χαμηλό κόστος μπορεί να υλοποιηθεί ένα σύστημα το οποίο δίνει στο χρήστη τη δυνατότητα να συλλέγει, να επεξεργάζεται και να αποθηκεύει δεδομένα για μελλοντική χρήση.

Το σύστημα αυτό μπορεί να βρει εφαρμογή στη γεωργία και την πρόγνωση του καιρού, καθώς έχει τη δυνατότητα να συλλέγει σημαντικές πληροφορίες, χωρίς να έχει ιδιαίτερες απαιτήσεις σε κατανάλωση ενέργειας και σε γνώσεις προγραμματισμού.

Επιπλέον βελτιώσεις και επεκτάσεις που θα μπορούσαν να γίνουν στο σύστημα αυτό είναι :

- Επικοινωνία με το διαδίκτυο : Με τη χρήση Ethernet Shield ή Wifi Shield θα μπορούσαμε να έχουμε τη δυνατότητα επικοινωνίας με το διαδίκτυο είτε με τη χρήση καλωδίου είτε ασύρματα.
- Αισθητήρες : Με τη χρήση αισθητήρων υψηλότερου κόστους θα είχαμε μετρήσεις με μεγαλύτερη ακρίβεια.
- Προγραμματισμός : Πιθανές βελτιώσεις στο κομμάτι του προγραμματισμού θα μείωναν τις ανάγκες του συστήματος σε μνήμη.
- Αισθητήρας ανέμου : Αν στο σύστημα είχε περιληφθεί ο συγκεκριμένος αισθητήρας, θα είχαμε τη δυνατότητα δημιουργίας ενός ανεξάρτητου μετεωρολογικού σταθμού.
- Εφαρμογή για smartphone : Θα μπορούσαμε να υλοποιήσουμε κατάλληλη εφαρμογή για αποστολή των δεδομένων απ' ευθείας σε κινητό τηλέφωνο.
- Στατιστικά στοιχεία : Καταγράφοντας σε καθημερινή βάση τις μετρήσεις του συστήματος για μεγάλο χρονικό διάστημα θα είχαμε τη δυνατότητα να συγκεντρώσουμε στατιστικά στοιχεία για το κλίμα και τις ανάγκες της περιοχής, όπως για παράδειγμα η μέση μηνιαία θερμοκρασία ή η μέση μηνιαία βροχόπτωση.

## ΠΑΡΑΡΤΗΜΑ

### ΚΩΔΙΚΕΣ ΑΙΣΘΗΤΗΡΩΝ

#### ΚΩΔΙΚΑΣ ΑΙΣΘΗΤΗΡΑ ΘΕΡΜΟΚΡΑΣΙΑΣ/ΣΧΕΤΙΚΗΣ ΥΓΡΑΣΙΑΣ (ΒΙΒΛΙΟΘΗΚΗ <dht.h>)

```
#include <dht.h>           // περιλαμβάνεται η βιβλιοθήκη <dht.h>

#define dht_dpin A0

dht DHT;

void setup(){
  Serial.begin(9600);
  delay(300);             // καθυστέρηση 300 ms
  Serial.println("Humidity and temperature\n\n"); // εμφάνισε στη σειριακή οθόνη το μήνυμα
                                                                "Humidity and temperature"
  delay(700);            // καθυστέρηση 700 ms
}                          // τέλος του "setup()"

void loop(){              // ξεκινά το κύριο πρόγραμμα

  DHT.read11(dht_dpin);

  Serial.print("Current humidity = "); // εμφάνισε στην οθόνη το μήνυμα "Current humidity="
  Serial.print(DHT.humidity);         // εμφάνισε την τιμή της υγρασίας
  Serial.print("% ");                 // εμφάνισε το σύμβολο %
  Serial.print("temperature = ");    // εμφάνισε στην οθόνη το μήνυμα "temperature="
  Serial.print(DHT.temperature);     // εμφάνισε την τιμή της θερμοκρασίας
  Serial.println("C ");               // εμφάνισε το σύμβολο C
  delay(5000);                        // καθυστέρηση 500 ms
}                                      // τέλος του "loop()"
```



## ΚΩΔΙΚΑΣ ΔΙΣΘΗΤΗΡΑ ΒΑΡΟΜΕΤΡΙΚΗΣ ΠΙΕΣΗΣ

(BIBΛΙΟΘΗΚΗ <Adafruit\_BMP085.h>)

```
#include <Wire.h>                // περιλαμβάνεται η βιβλιοθήκη <Wire.h>
#include <Adafruit_BMP085.h>     // περιλαμβάνεται η βιβλιοθήκη <Adafruit_BMP085.h>

Adafruit_BMP085 bmp;

void setup() {                  // αρχή του "setup()"

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

void loop() {                   // αρχή του "loop()"

  //Serial.print("Temperature = "); // εμφάνισε στη σειριακή οθόνη το μήνυμα "Temperature="
  //Serial.print(bmp.readTemperature()); // εμφάνισε στη σειριακή οθόνη την τιμή της θερμοκρασίας
  //Serial.println(" *C");           // εμφάνισε στη σειριακή οθόνη το σύμβολο *C

  Serial.print("Pressure = ");      // εμφάνισε στη σειριακή οθόνη το μήνυμα "Pressure="
  Serial.print(bmp.readPressure()); // εμφάνισε στη σειριακή οθόνη την τιμή της πίεσης
  Serial.println(" Pa");           // εμφάνισε στη σειριακή οθόνη το σύμβολο Pa

  // Serial.print("Altitude = ");   // εμφάνισε στη σειριακή οθόνη το μήνυμα "Altitude="
  // Serial.print(bmp.readAltitude()); // εμφάνισε στη σειριακή οθόνη την τιμή του υψόμετρου
  // Serial.println(" meters");     // εμφάνισε στη σειριακή οθόνη το σύμβολο meters

  // Serial.print("Pressure at sealevel (calculated) = "); // εμφάνισε στη σειριακή οθόνη το μήνυμα
  //                                                         // "Pressure at sealevel (calculated) = "
  // Serial.print(bmp.readSealevelPressure()); // εμφάνισε στη σειριακή οθόνη την τιμή του
  //                                                         // υψόμετρου στο επίπεδο της θάλασσας
  // Serial.println(" Pa");         // εμφάνισε στη σειριακή οθόνη το σύμβολο Pa

  //Serial.print("Real altitude = "); // εμφάνισε στη σειριακή οθόνη το μήνυμα "Real altitude="
  //Serial.print(bmp.readAltitude(101500)); // εμφάνισε στη σειριακή οθόνη την τιμή του υψόμετρου
  //Serial.println(" meters");       // εμφάνισε στη σειριακή οθόνη το σύμβολο meters

  Serial.println();
  delay(500); // καθυστέρηση 500 ms

  // }                            // τέλος του "loop()"
}
```

## **ΚΩΔΙΚΑΣ ΑΙΣΘΗΤΗΡΑ ΕΝΤΑΣΗΣ ΦΩΤΙΣΜΟΥ**

(BIBΛΙΟΘΗΚΗ <BH1750.h>)

```
#include <Wire.h>           // περιλαμβάνεται η βιβλιοθήκη <Wire.h>
#include <BH1750.h>        // περιλαμβάνεται η βιβλιοθήκη <BH1750.h>

BH1750 lightMeter;        // αισθητήρας έντασης φωτισμού BH1750

void setup(){              // αρχή του "setup()"
  Serial.begin(9600);
  lightMeter.begin();
  Serial.println("Running...");
}                            // τέλος του "setup()"

void loop() {              // αρχή του "loop()"

  uint16_t lux = lightMeter.readLightLevel(); // δήλωση της μεταβλητής

  Serial.print("Light: "); // εμφάνισε στη σειριακή οθόνη το μήνυμα "Light"
  Serial.print(lux);       // εμφάνισε στη σειριακή οθόνη την τιμή της έντασης φωτισμού
  Serial.println(" lx");   // εμφάνισε στη σειριακή οθόνη το σύμβολο lx

  delay(1000);             // καθυστέρηση 1000 ms
}                            // τέλος του "loop()"
```

## **ΚΩΔΙΚΑΣ ΑΙΣΘΗΤΗΡΑ ΥΨΟΥΣ ΒΡΟΧΗΣ**

(BIBΛΙΟΘΗΚΗ <AnalogReadSerial.h>)

```
void setup() {             // αρχή του "setup"
  Serial.begin(9600);      // εκκίνηση σειριακής επικοινωνίας στα 9600 bits ανά δευτερόλεπτο
}                            // τέλος του "setup"

void loop() {              // αρχή του "loop"

  int vrox1 = analogRead(A1); // διάβασε την τιμή της αναλογικής εισόδου

  Serial.println(vrox1);    // εμφάνισε στη σειριακή οθόνη την τιμή της αναλογικής εισόδου

  delay(1000);             // καθυστέρηση 1000 ms
}                            // τέλος του "loop"
```

### **ΚΩΔΙΚΑΣ ΔΙΣΘΗΤΗΡΑ ΥΓΡΑΣΙΑΣ ΕΛΑΦΟΥΣ**

(BIBΛΙΟΘΗΚΗ <AnalogReadSerial.h>)

```
void setup() {      // αρχή του "setup"  
  Serial.begin(9600); // εκκίνηση σειριακής επικοινωνίας στα 9600 bits ανά δευτερόλεπτο  
}                  // τέλος του "setup"  
  
void loop() {      // αρχή του "loop"  
  
  int xoma = analogRead(A2); // διάβασε την τιμή της αναλογικής εισόδου  
  
  Serial.println(xoma);      // εμφάνισε στη σειριακή οθόνη την τιμή της αναλογικής εισόδου  
  
  delay(1000); // καθυστέρηση 1000 ms  
  
}                  // τέλος του "loop"
```

### **ΚΩΔΙΚΑΣ ΟΘΟΝΗΣ LCD**

(BIBΛΙΟΘΗΚΗ <LiquidCrystal.h>)

```
#include <LiquidCrystal.h> // περιλαμβάνεται η βιβλιοθήκη <LiquidCrystal.h>  
  
LiquidCrystal lcd(12, 11, 5, 4, 3, 2); // αρχικοποίηση της βιβλιοθήκης με τα interface pins  
  
void setup() { // αρχή του setup()  
  
  lcd.begin(16, 2); // ρύθμιση του αριθμού στηλών και γραμμών της οθόνης LCD  
  
  // lcd.print("hello, world!"); // εμφάνισε στην οθόνη το μήνυμα "hello, word"  
}  
  
void loop() { // αρχή του loop()  
  
  lcd.noDisplay(); // απόκρυψη των μηνυμάτων της οθόνης  
  delay(500); // καθυστέρηση 500 ms  
  
  lcd.display(); // εμφάνιση των μηνυμάτων της οθόνης  
  delay(500); // καθυστέρηση 500 ms  
} // τέλος του loop()
```

## ΚΥΡΙΟΣ ΚΩΔΙΚΑΣ ΕΚΤΕΛΕΣΗΣ ΤΟΥ ΠΡΟΓΡΑΜΜΑΤΟΣ

```
#include <Wire.h> // περιλαμβάνεται η βιβλιοθήκη <Wire.h>
#include <Adafruit_BMP085.h> // περιλαμβάνεται η βιβλιοθήκη <Adafruit_BMP085.h>

Adafruit_BMP085 bmp;

int vroxix=0; // αρχικοποίηση μεταβλητών
int mmvroxix=0; // αρχικοποίηση μεταβλητών
int xomax=0; // αρχικοποίηση μεταβλητών

#include <LiquidCrystal.h> // περιλαμβάνεται η βιβλιοθήκη <LiquidCrystal.h>

LiquidCrystal lcd(12, 11, 5, 4, 3, 2); // αρχικοποίηση της βιβλιοθήκης με τους αριθμούς των
// interface pins

#include <dht.h> // περιλαμβάνεται η βιβλιοθήκη <dht.h>

#define dht_dpin A0

dht DHT;

#include <Wire.h> // περιλαμβάνεται η βιβλιοθήκη <Wire.h>
#include <BH1750.h> // περιλαμβάνεται η βιβλιοθήκη <BH1750.h>

BH1750 lightMeter; // αισθητήρας έντασης φωτισμού BH1750

void setup() { // αρχή του "setup()"

  lcd.begin(16, 2); // ρύθμιση του αριθμού στηλών και γραμμών της οθόνης LCD

  lightMeter.begin();
  Serial.println("Running...");
  Serial.begin(9600);

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

} // τέλος του "setup()"

void loop() { // αρχή του "loop()"
```

```

DHT.read11(dht_dp1n);           // ξεκινά το κύριο μέρος του προγράμματος

vroxi = analogRead(A1);        // ο αισθητήρας βροχής συνδέεται στην αναλογική είσοδο A1
xoma = analogRead(A2);        // ο αισθητήρας υγρασίας εδάφους στην αναλογική είσοδο A2

lcd.setCursor(0, 0);           // πήγαινε στη θέση (0,0) της οθόνης LCD
lcd.print("Temperature= ");    // εμφάνισε στην οθόνη το μήνυμα "Temperature"
lcd.setCursor(0, 1);           // πήγαινε στη θέση (0,1) της οθόνης LCD
lcd.print(DHT.temperature);    // εμφάνισε στην οθόνη την τιμή της τρέχουσας θερμοκρασίας
lcd.print(" oC");              // εμφάνισε στην οθόνη το σύμβολο oC

Serial.print("Temperature:");  // εμφάνισε στη σειριακή οθόνη το μήνυμα "Temperature:"
Serial.print(DHT.temperature); // εμφάνισε στη σειριακή οθόνη την τιμή της θερμοκρασίας
Serial.print(" oC ");          // εμφάνισε στη σειριακή οθόνη το σύμβολο oC

delay(1000);                   // καθυστέρηση 1000 ms
lcd.clear();                    // καθάρισε την οθόνη LCD

lcd.setCursor(0, 0);           // πήγαινε στη θέση (0,0) της οθόνης LCD
lcd.print("Current humidity= "); // εμφάνισε στην οθόνη το μήνυμα "Current humidity="
lcd.setCursor(0, 1);           // πήγαινε στη θέση (0,1) της οθόνης LCD
lcd.print(DHT.humidity);       // εμφάνισε στην οθόνη την τιμή της υγρασίας
lcd.print(" %");               // εμφάνισε στην οθόνη το σύμβολο %

Serial.print("Humidity:");     // εμφάνισε στη σειριακή οθόνη το μήνυμα "Humidity:"
Serial.print(DHT.humidity);    // εμφάνισε στη σειριακή οθόνη την τιμή της υγρασίας
Serial.print(" % ");           // εμφάνισε στη σειριακή οθόνη το σύμβολο %

delay(1000);                   // καθυστέρηση 1000 ms
lcd.clear();                    // καθάρισε την οθόνη LCD

uint16_t lux = lightMeter.readLightLevel(); // δήλωση της μεταβλητής lux

lcd.setCursor(0, 0);           // πήγαινε στη θέση (0,0) της οθόνης LCD
lcd.print("Light= ");          // εμφάνισε στην οθόνη το μήνυμα "Light="
lcd.setCursor(0, 1);           // πήγαινε στη θέση (0,1) της οθόνης LCD
lcd.print(lux);                // εμφάνισε στην οθόνη την τιμή της έντασης φωτισμού
lcd.print(" lx");

Serial.print("Light:");        // εμφάνισε στη σειριακή οθόνη το μήνυμα "Light:"
Serial.print(lux);             // εμφάνισε στη σειριακή οθόνη την τιμή της έντασης φωτισμού
Serial.print(" lx ");          // εμφάνισε στη σειριακή οθόνη το σύμβολο lx

delay(1000);                   // καθυστέρηση 1000 ms
lcd.clear();                    // καθάρισε την οθόνη LCD

```

```

int32_t piesi = bmp.readPressure();    // δήλωση της μεταβλητής piesi

lcd.setCursor(0, 0);                  // πήγαινε στη θέση (0,0) της οθόνης LCD
lcd.print("Pressure= ");              // εμφάνισε στην οθόνη το μήνυμα "Pressure="
lcd.setCursor(0, 1);                  // πήγαινε στη θέση (0,1) της οθόνης LCD
lcd.print(piesi);                      // εμφάνισε στην οθόνη την τιμή της πίεσης
lcd.print(" Pa");                     // εμφάνισε το σύμβολο Pa

Serial.print("Pressure:");            // εμφάνισε στη σειριακή οθόνη το μήνυμα "Pressure:"
Serial.print(bmp.readPressure());     // εμφάνισε στη σειριακή οθόνη την τιμή της πίεσης
Serial.print(" Pa ");                 // εμφάνισε στη σειριακή οθόνη το σύμβολο Pa

delay(1000);                          // καθυστέρηση 1000 ms
lcd.clear();                           // καθάρισε την οθόνη LCD

if( vroxi <= 490){                    // αν η τιμή της μεταβλητής vroxi είναι <=490 τότε υπολόγισε τα mmvroxi
  mmvroxi=0.01*vroxi;                 // από αυτή τη σχέση
}

if( vroxi >= 490 && vroxi < 600){     // αν η τιμή είναι >=490 και <=600 υπολόγισε
                                     // τα mmvroxi από αυτή τη σχέση
  mmvroxi= -0.0000000333631*vroxi*vroxi*vroxi*vroxi+0.0000772554*vroxi*vroxi*vroxi-
  0.06474*vroxi*vroxi+23.63446*vroxi-3197.62725;
}

if( vroxi >= 600){                    // αν η τιμή της μεταβλητής είναι >600 τότε τα mmvroxi παίρνουν την
  mmvroxi=40;                          // από αυτή τη σχέση
}

delay(1000);                          // καθυστέρηση 1000 ms
lcd.clear();                           // καθάρισε την οθόνη LCD

lcd.setCursor(0, 0);                  // πήγαινε στη θέση (0,0) της οθόνης LCD
lcd.print("Rain= ");                  // εμφάνισε στην οθόνη το μήνυμα "Rain="
lcd.setCursor(0, 1);                  // πήγαινε στη θέση (0,1) της οθόνης LCD
lcd.print(mmvroxi);                   // εμφάνισε στην οθόνη την τιμή του ύψους βροχής
lcd.print(" mm");                     // εμφάνισε στην οθόνη το σύμβολο mm

Serial.print("Rain:");                // εμφάνισε στη σειριακή οθόνη το μήνυμα "Rain:"
Serial.print(mmvroxi);                 // εμφάνισε στη σειριακή οθόνη την τιμή του ύψους βροχής
Serial.print(" mm ");                  // εμφάνισε στη σειριακή οθόνη το σύμβολο mm

delay(1000);                          // καθυστέρηση 1000 ms
lcd.clear();                           // καθάρισε την οθόνη LCD
lcd.setCursor(0, 0);                  // πήγαινε στη θέση (0,0) της οθόνης LCD

```

```

lcd.print("Soil = ");           // εμφάνισε στην οθόνη το μήνυμα "Soil="
lcd.setCursor(0, 1);           // πήγαινε στη θέση (0,1) της οθόνης LCD

if( xoma <= 490){               // αν η τιμή της μεταβλητής xoma είναι <=490 τότε
  lcd.print("YGRO");           // εμφάνισε στην οθόνη το μήνυμα "YGRO"
  Serial.print("Soil: YGRO"); // εμφάνισε στη σειριακή οθόνη το μήνυμα "Soil:YGRO"
}

if( xoma >= 490 && xoma < 600){ // αν η τιμή είναι >=490 και <= 600 τότε
  lcd.print("NORM");           // εμφάνισε στην οθόνη το μήνυμα "NORM"
  Serial.print("Soil: NORM"); // εμφάνισε στη σειριακή οθόνη το μήνυμα "Soil:NORM"
}

  if( xoma >= 600){            // αν η τιμή της μεταβλητής xoma είναι >=600 τότε
  lcd.print("XERO");           // εμφάνισε στην οθόνη το μήνυμα "XERO"
  Serial.print("Soil: XERO"); // εμφάνισε στη σειριακή οθόνη το μήνυμα "Soil:XERO"
}

Serial.println(" ");           // αλλαγή γραμμής στη σειριακή οθόνη

delay(1000);                   // καθυστέρηση 1000 ms
lcd.clear();                   // καθάρισε την οθόνη LCD

}

```

## ΚΩΔΙΚΑΣ VISUAL BASIC

```
Dim buf As String           // buffer για εισερχόμενα σειριακά δεδομένα
Dim cell_idx As Integer     // ο αριθμός γραμμής του κελιού που αποθηκεύεται ένα barcode που
                             έχει ληφθεί

Private Sub StrokeReader1_CommEvent(ByVal Evt As StrokeReaderLib.Event, _
                                     ByVal data As Variant)
    Select Case Evt // μπορεί να είναι EVT_DISCONNECT ή EVT_DATA ή EVT_SERIALEVENT
        Case EVT_DISCONNECT // αυτό το μήνυμα εμφανίζεται όταν ο adapter της σειριακής
                             θύρας USB έχει αποσυνδεθεί από τον υπολογιστή
            MsgBox "Serial port converter disconnected"
        Case EVT_CONNECT
            MsgBox "Connected" // αν ο RS232 ή ο RS485 adapter μόλις συνδέθηκε στο διάλογο USB

        Case EVT_DATA // εισερχόμενα σειριακά δεδομένα
            buf = buf + StrokeReader1.Read(Text) // εισερχόμενα σειριακά δεδομένα που
                                                  συσσωρεύονται στο buf

            Do // η δομή Do.....Loop επιτρέπει να ελέγξουμε μια συνθήκη είτε στην αρχή είτε στο τέλος
                LF = InStr(buf, Chr(10)) // οι περισσότεροι σαρωτές στέλνουν ASCII <LF> μετά από
                                         κάθε barcode

                If LF = 0 Then Exit Do // αναμονή για λήψη ASCII <LF>

                s = Left(buf, LF - 1) // αντιγραφή των δεδομένων πριν τον κώδικα <LF> ως
                                     το κείμενο του barcode
                s = Replace(s, Chr(13), "") // κάποιιοι σαρωτές μπορεί να στέλνουν <CR><LF>. Αφαίρεσε
                                           το <CR> αν εμφανίζεται

                buf = Right(buf, Len(buf) - LF) // αφαίρεσε τα αναλυμένα δεδομένα από τον buffer
                // αυτό είναι χρήσιμο όταν το εξωτερικό hardware στέλνει πολλαπλά barcodes στον ίδιο χρόνο
                cell_idx = cell_idx + 1 // αυξάνεται κατά 1 ο αριθμός γραμμής του κελιού όπου θα
                                       αποθηκευτεί το barcode

                Cells(cell_idx + 1, 2) = s
                Cells(cell_idx + 1, 1) = Now() // συνάρτηση που επιστρέφει την τρέχουσα ημερομηνία
                                               και ώρα

            Loop // τέλος του Loop
    End Select
End Sub
```



## **ΒΙΒΛΙΟΓΡΑΦΙΑ-ΠΗΓΕΣ**

- [1]<https://el.wikipedia.org/wiki/%CE%9C%CE%B5%CF%84%CE%B5%CF%89%CF%81%CE%BF%CE%BB%CE%BF%CE%B3%CE%AF%CE%B1>
- [2] <https://www.arduino.cc/>
- [3] <https://www.arduino.cc/en/Tutorial/PWM>
- [4] [http://users.sch.gr/asal1/material/seminaria/teliko24\\_1.pdf](http://users.sch.gr/asal1/material/seminaria/teliko24_1.pdf)
- [5] <https://www.arduino.cc/en/Main/ArduinoEthernetShield>
- [6] <https://www.arduino.cc/en/Guide/ArduinoWiFiShield>
- [7] <http://www.ebay.com/itm/DHT11-DHT-11-Digital-Temperature-and-Humidity-Sensor-for-Arduino-/321969658244?hash=item4af6e31984:g:bPOAAOSwyQtV3BHp>
- [8] <http://www.ebay.com/itm/Soil-Hygrometer-Detection-Module-Soil-Moisture-Sensor-For-arduino-Smart-car-/400364471802?hash=item5d379505fa:g:JpgAAOxyyjpRxuot>
- [9] <http://www.ebay.com/itm/2pcs-Water-Level-Sensor-Depth-of-Detection-Water-Sensor-for-Arduino-/381374758015?hash=item58cbb4fc7f:g:V80AAMXQyY1TV-CR>
- [10] <http://www.ebay.com/itm/Neu-SainSmart-BMP085-Barometric-Digital-Pressure-Sensor-Modul-Board-for-Arduino-/221135850501?hash=item337cb94c05:g:b4AAAoxyDoZSIabT>
- [11] <http://www.ebay.com/itm/New-BH1750FVI-Digital-Light-intensity-Sensor-Module-For-Arduino-3V-5V-power-/261831720173?hash=item3cf662fced:g:e7wAAOSwstxVGppz>
- [12] <http://www.ebay.com/itm/ESUMIC-LCD-Module-for-Arduino-UNO-R3-MEGA2560-16-X-2-1602-White-on-Blue-/131681147105?hash=item1ea8ced8e1:g:Mx4AAOSwhkRWdvz->
- [13]<https://el.wikipedia.org/wiki/%CE%91%CE%B9%CF%83%CE%B8%CE%B7%CF%84%CE%AE%CF%81%CE%B1%CF%82>
- [14][https://el.wikipedia.org/wiki/%CE%98%CE%B5%CF%81%CE%BC%CE%BF%CE%BA%CF%81%CE%B1%CF%83%CE%AF%CE%B1\\_%CE%B1%CF%84%CE%BC%CF%8C%CF%83%CF%86%CE%B1%CE%B9%CF%81%CE%B1%CF%82](https://el.wikipedia.org/wiki/%CE%98%CE%B5%CF%81%CE%BC%CE%BF%CE%BA%CF%81%CE%B1%CF%83%CE%AF%CE%B1_%CE%B1%CF%84%CE%BC%CF%8C%CF%83%CF%86%CE%B1%CE%B9%CF%81%CE%B1%CF%82)
- [15][https://el.wikipedia.org/wiki/%CE%A5%CE%B3%CF%81%CE%B1%CF%83%CE%AF%CE%B1\\_%CE%B1%CF%84%CE%BC%CF%8C%CF%83%CF%86%CE%B1%CE%B9%CF%81%CE%B1%CF%82](https://el.wikipedia.org/wiki/%CE%A5%CE%B3%CF%81%CE%B1%CF%83%CE%AF%CE%B1_%CE%B1%CF%84%CE%BC%CF%8C%CF%83%CF%86%CE%B1%CE%B9%CF%81%CE%B1%CF%82)

[16] [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)

[17] [https://el.wikipedia.org/wiki/%CE%88%CE%BD%CF%84%CE%B1%CF%83%CE%B7\\_%CF%86%CF%89%CF%84%CE%B9%CF%83%CE%BC%CE%BF%CF%8D](https://el.wikipedia.org/wiki/%CE%88%CE%BD%CF%84%CE%B1%CF%83%CE%B7_%CF%86%CF%89%CF%84%CE%B9%CF%83%CE%BC%CE%BF%CF%8D)

[18] <http://www.originlab.com/>

[19] <https://strokescribe.com/en/read-serial-port-excel-2007.html>

[20] [https://en.wikipedia.org/wiki/Visual\\_Basic](https://en.wikipedia.org/wiki/Visual_Basic)

[21] [http://users.auth.gr/silleos/new\\_page\\_6.htm](http://users.auth.gr/silleos/new_page_6.htm)

[22] <http://www.meteormetrics.com/zambretti.htm>

[23] <http://weather.vouhead.gr/wxforecast-zambretti.php>