



ΤΕΧΝΟΛΟΓΙΚΟ  
ΕΚΠΑΙΔΕΥΤΙΚΟ  
ΙΔΡΥΜΑ  
ΔΥΤΙΚΗΣ ΕΛΛΑΔΑΣ

ΣΧΟΛΗ ΤΕΧΝΟΛΟΓΙΚΩΝ ΕΦΑΡΜΟΓΩΝ

ΤΜΗΜΑ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ Τ.Ε.

ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ 1703

# ΠΑΡΑΚΟΛΟΥΘΗΣΗ ΘΕΡΜΟΚΡΑΣΙΑΣ- ΥΓΡΑΣΙΑΣ ΜΕ ARDUINO



ΣΠΟΥΔΑΣΤΕΣ : ΑΡΒΑΝΙΤΗΣ ΑΝΑΣΤΑΣΙΟΣ

ΠΕΣΜΑΤΖΟΓΛΟΥ ΒΑΣΙΛΕΙΟΣ

ΕΠΟΠΤΕΥΟΝ ΚΑΘΗΓΗΤΗΣ : ΚΑΡΕΛΗΣ ΔΗΜΗΤΡΗΣ

ΠΑΤΡΑ 2018



## Ευχαριστίες

*Με την ολοκλήρωση της πτυχιακή μας εργασίας θα θέλαμε να ευχαριστήσουμε όσους συνέβαλαν στην υλοποίησή της .*

*Πρωτίστως θα θέλαμε να ευχαριστήσουμε τον επιβλέπων καθηγητή μας κ. Καρέλη Δημήτριο, για την άψογη συνεργασία και την εμπιστοσύνη που μας έδειξε καθ' όλη τη διάρκεια της προσπάθειάς μας .*

*Τέλος θα θέλαμε να αφιερώσουμε την πτυχιακή μας εργασία στις οικογένειες μας και τους φίλους μας για την ηθική συμπαράσταση που μας παρείχαν καθ' όλη την διάρκεια των σπουδών μας .*

## ΠΕΡΙΛΗΨΗ

Το αντικείμενο της παρούσας εργασίας είναι η δημιουργία ενός μετρητή θερμοκρασίας και υγρασίας του περιβάλλοντος και την εμφάνισή τους σε μια οθόνη υγρών κρυστάλλων μέσω του μικροελεγκτή Arduino που είναι μία πλακέτα "ανοικτού κώδικα" με την οποία κάποιος χωρίς ιδιαίτερες γνώσεις προγραμματιστή μπορεί εύκολα να την χρησιμοποιήσει και να κατασκευάσει εφαρμογές ρομποτικής και συστήματα αυτοματισμού.. πιο συγκεκριμένα της έκδοσης Uno. Ο προγραμματισμός του Arduino γίνεται σε C ή C++ ,το Arduino IDE έρχεται με μια βιβλιοθήκη λογισμικού που ονομάζεται "Wiring". Το ολοκληρωμένο περιβάλλον ανάπτυξης (IDE) του Arduino είναι μία εφαρμογή γραμμένη σε Java, που λειτουργεί σε πολλές πλατφόρμες και προέρχεται από το IDE για τη γλώσσα προγραμματισμού Processing και το σχέδιο Wiring. Επίσης χρησιμοποιήθηκε και ένα επιπλέον λογισμικό λεγόμενο και ως Fritzing μέσω του οποίου μπόρεσα να κάνω την αναπαράσταση των υλικών που χρησιμοποιήσα για την εργασία αλλά και των αποτελεσμάτων της. Το Fritzing είναι ένα δωρεάν εργαλείο σχεδίασης κυκλωμάτων. Είναι πολύ απλό στην χρήση του και έχει ενσωματωμένα όλα τα ηλεκτρονικά εξαρτήματα που θα χρειαστείτε για την σχεδίαση των κυκλωμάτων σας. Μέσα σε αυτό θα βρείτε και όλες τις πλακέτες Arduino καθώς και διάφορα αισθητήρια γι' αυτό. Θα γίνει ανάλυση των υλικών και των προγραμμάτων που χρησιμοποιήθηκαν στην εργασία καθώς και παράθεση εικόνων για την καλύτερη κατανόηση τους.

## **ABSTRACT**

The subject of this work is to create a temperature and humidity meter of the environment and display it on a liquid crystal display through the Arduino microcontroller, an open source board with which someone without a knowledgeable developer can easily use it and to build robotics applications and automation systems .. more specifically the Uno version. Arduino's programming is done in C or C ++, the Arduino IDE comes with a software library called "Wiring". Arduino's integrated development environment (IDE) is a Java-based, multi-platform application that comes from the IDE for the programming language and the Wiring design. I also used an additional software called Fritzing through which I was able to make the representation of the materials I used for the work and its results. Fritzing is a free circuit design tool. It is very simple to use and incorporates all the electronic components you will need to design your circuits. Inside this you will find all the Arduino boards as well as various sensors for that. An analysis will be made of the materials and programs used in the work, as well as the quoting of images for their better understanding.

ΠΕΡΙΛΗΨΗ .....	4
ABSTRACT.....	5
ΚΕΦΑΛΑΙΟ 1: ΜΙΚΡΟΕΛΕΓΚΤΕΣ .....	8
1.1 Τι είναι ο Μικροελεγκτής.....	8
1.1.1 Δομή Μικροελεγκτή .....	8
1.1.2 Δίαυλοι επικοινωνίας μικροελεγκτή.....	9
1.2 Εφαρμογές μικροελεγκτών .....	10
1.3 Κατασκευαστές μικροελεγκτών .....	11
1.4 Γλώσσα προγραμματισμού μικροελεγκτών .....	11
1.5 Γλώσσες χαμηλού επιπέδου .....	11
1.5.1 Πλεονεκτήματα γλωσσών χαμηλού επιπέδου .....	11
1.5.2 Μειονεκτήματα γλωσσών χαμηλού επιπέδου .....	12
1.6 Γλώσσες υψηλού επιπέδου.....	12
1.6.1 Πλεονεκτήματα γλωσσών υψηλού επιπέδου.....	12
1.6.2 Μειονεκτήματα γλωσσών υψηλού επιπέδου.....	12
1.7 Διαδεδομένες κατηγορίες μικροελεγκτών .....	12
ΚΕΦΑΛΑΙΟ 2: ΜΙΚΡΟΕΛΕΓΚΤΗΣ ARDUINO .....	14
2.1 Ιστορική αναδρομή Arduino .....	14
2.2 Τι είναι ο ARDUINO .....	14
2.3 Πλεονεκτήματα ARDUINO.....	16
2.4 Πλακέτα ARDUINO .....	17
2.5 Επίσημες πλακέτες .....	17
2.6 Λογισμικό- Προγραμματισμός.....	19
2.6.1 Λογισμικό ARDUINO.....	19
2.6.2 Προγραμματισμός- Βασικές λειτουργίες.....	20
2.7 Επεκτάσεις ARDUINO (ARDUINO shields).....	27
2.7.1 Αναλυτική Περιγραφή.....	29
ΚΕΦΑΛΑΙΟ 3: ARDUINO UNO .....	31
3.1 Τι είναι το Arduino Uno.....	31
3.2 Ανάλυση πλακέτας Arduino Uno.....	32
3.3. Τεχνικά Χαρακτηριστικά .....	35
3.4 Τεχνικά χαρακτηριστικά Arduino Uno .....	36

3.4.1 FlashMemory .....	36
3.4.2 SRAMMemory .....	36
3.4.3 EEPROMMemory .....	36
3.4.4 FTDI .....	37
ΚΕΦΑΛΑΙΟ 4 : SENSORS (ΑΙΣΘΗΤΗΡΕΣ) .....	38
4.1 DHT-11 αισθητήρας θερμοκρασία και υγρασία .....	38
4.1.2 Σχετικά με το DHT-11 .....	38
4.2 Αισθητήρας θερμοκρασίας DS18B20 .....	39
4.2.1 Βασικά χαρακτηριστικά DS18B20 .....	39
4.3 Αισθητήρας θερμοκρασίας LM35 .....	40
4.4 Αισθητήρας Αερίου MQ-7 .....	41
4.5 Ηλεκτρονικά εξαρτήματα .....	42
4.5.1 Ποτενσιόμετρο – Potentiometer .....	42
4.5.2 Φωτοдиодοι – Leds .....	42
4.6 Δυνατοτητες του ARDUINO .....	43
ΚΕΦΑΛΑΙΟ 5 : Πειραματική διαδικασία .....	45
5.1 Σκοπός .....	45
5.2 Τα υλικά της κατασκευής .....	45
5.3 Εγκατάσταση του Arduino IDE στο PC .....	46
5.4 Δοκιμαστικός κώδικας .....	52
5.5 LCD Οθόνη .....	52
5.5.1 Συνδεσμολογία οθόνης LCD .....	53
5.6 Κώδικας μέτρησης DHT-11 .....	54
5.7 Κώδικας μέτρησης DS18B20 .....	56
5.8 ΣΥΝΔΕΣΜΟΛΟΓΙΑ ΚΥΚΛΩΜΑΤΟΣ .....	58
ΒΙΒΛΙΟΓΡΑΦΙΑ .....	59

# ΚΕΦΑΛΑΙΟ 1: ΜΙΚΡΟΕΛΕΓΚΤΕΣ

## 1.1 Τι είναι ο Μικροελεγκτής

Μικροελεγκτής είναι ένα προγραμματιζόμενο ολοκληρωμένο κύκλωμα το οποίο διαθέτει επεξεργαστή, μνήμη, διάφορα περιφερειακά κυκλώματα καθώς επίσης και θύρες εισόδου/εξόδου για επικοινωνία με εξωτερικές συσκευές. Θα μπορούσε να παρομοιαστεί με έναν μικροϋπολογιστή. Όπως ακριβώς ένας μικροϋπολογιστής έχει επεξεργαστή, μνήμη, περιφερειακές συσκευές και εκτελεί προγράμματα έτσι κι ένας μικροελεγκτής διαθέτει τα παραπάνω χαρακτηριστικά και μάλιστα ολοκληρωμένα σε ένα μόνο chip. Το πρόγραμμα που εκτελεί ο μικροελεγκτής αποθηκεύεται μόνιμα στη μνήμη προγράμματος.

### 1.1.1 Δομή Μικροελεγκτή

Η οργάνωση των μικροελεγκτών είναι παρόμοια με εκείνη των κλασικών υπολογιστών. Αποτελούνται από τις παρακάτω λειτουργικές μονάδες.

#### *Μονάδες εισόδου/εξόδου*

Με αυτές το σύστημα επικοινωνεί με το εξωτερικό του περιβάλλον.

#### Τον *μικροεπεξεργαστή* (ΜΚΕ)

Περιλαμβάνει:

- Την αριθμητική και λογική μονάδα, που επεξεργάζεται τα δεδομένα
- Τη μονάδα ελέγχου, που είναι υπεύθυνη για τον έλεγχο και τον συντονισμό όλων των μονάδων του συστήματος.
- Τους καταχωρητές, που χρησιμεύουν για προσωρινή αποθήκευση.

#### Την *κύρια μνήμη*

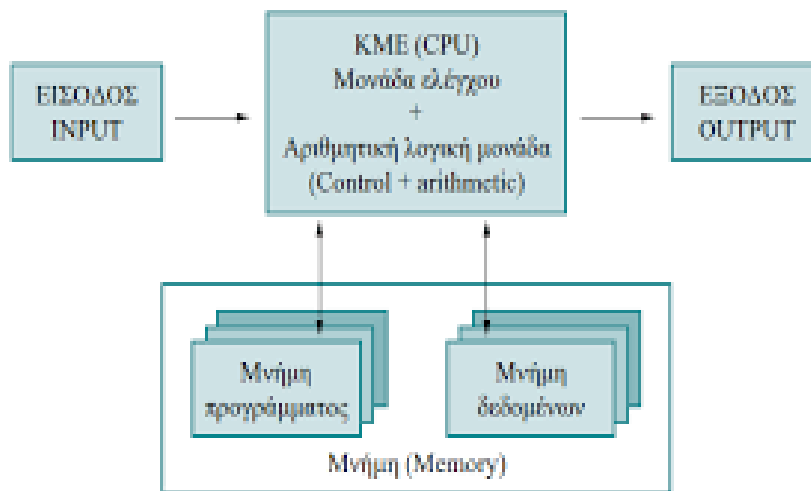
Χρησιμεύει για την αποθήκευση των εντολών του προγράμματος, των αρχικών δεδομένων και των ενδιάμεσων αποτελεσμάτων.

Το ιδιαίτερο χαρακτηριστικό σε σχέση με άλλα υπολογιστικά συστήματα είναι ότι ολόκληρη η μονάδα επεξεργασίας περιέχεται σε ένα ολοκληρωμένο κύκλωμα, που κατασκευάζεται σε ένα μικρό κομμάτι πυριτίου και αναφέρεται σαν μικροεπεξεργαστής.

Η ενσωμάτωση όλων των στοιχείων της κεντρικής μονάδας επεξεργασίας σε ένα μόνο ολοκληρωμένο κύκλωμα, συνδυάζει τα πλεονεκτήματα του μικρού μεγέθους, της υψηλής αξιοπιστίας και του χαμηλού κόστους.

Ο μικροεπεξεργαστής συνδέεται κατάλληλα με τα ολοκληρωμένα κυκλώματα της μνήμης και των μονάδων εισόδου/εξόδου για να αποτελέσει το υπολογιστικό σύστημα που ονομάζουμε *μικροελεγκτή*.





Σχήμα 1. Τυπικό σχήμα μικροελεγκτή.

### 1.1.2 Δίαυλοι επικοινωνίας μικροελεγκτή

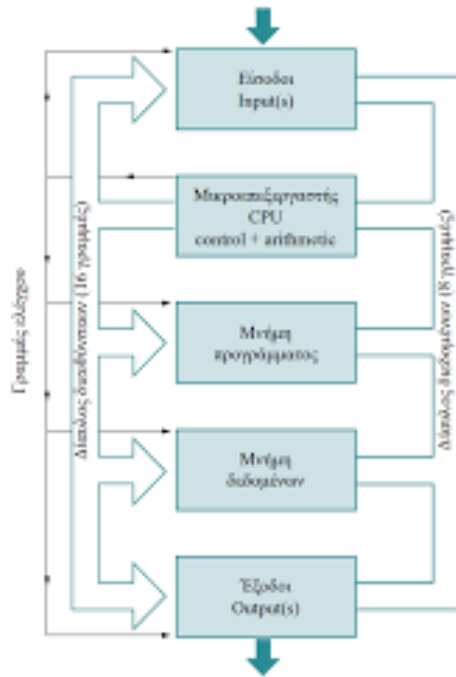
Η μεταφορά της δυαδικής πληροφορίας ανάμεσα στις σίαφορες μονάδες του μικροελεγκτή γίνεται παράλληλα από ένα σύνολο γραμμών, που αναφέρονται σαν **δίαυλος δεδομένων** (data bus).

Οι γραμμές αυτές αναφέρονται σαν **γραμμές δεδομένων** (data lines).

Ο δίαυλος δεδομένων δε λύνει όλα τα προβλήματα μεταφοράς της πληροφορίας. Ο μικροεπεξεργαστής θα πρέπει να έχει τη δυνατότητα επιλογής της μονάδας, με την οποία θα επικοινωνήσει και να μπορεί να την ειδοποιήσει ότι θα στείλει ή θα πάρει δεδομένα από αυτή.

Για το λόγο αυτό διαθέτει δύο ακόμα διαύλους, το **δίαυλο διευθύνσεων** (address bus) και το **δίαυλο ελέγχου** (control bus). Οι γραμμές των διαύλων αυτών λέγονται **γραμμές διευθύνσεων** (address lines), με τις οποίες ο μικροεπεξεργαστής στέλνει τη δυαδική διεύθυνση της θέσης μνήμης ή της μονάδας εισόδου/εξόδου, με την οποία θέλει να επικοινωνήσει. και **γραμμές ελέγχου** (control lines) με τις οποίες ο μικροεπεξεργαστής στέλνει τα κατάλληλα ηλεκτρικά σήματα για την ενεργοποίηση των επιθυμητών λειτουργιών της μνήμης ή των μονάδων εισόδου/εξόδου.

Η απαίτηση της ενεργοποίησης στοιχειωδών λειτουργιών σε προκαθορισμένα χρονικά διαστήματα δηλιουργεί την ανάγκη ύπαρξης μιας βάσης χρόνου, που αναφέρεται ως **κύκλωμα χρονισμού** (clock). Αποτελείται συνήθως από ένα κρυσταλλικό ταλαντωτή, που παράγει τετραγωνικούς παλμούς σταθερής συχνότητας. Η συχνότητα αυτή του ταλαντωτή καθορίζει και τη συχνότητα λειτουργίας του μικροεπεξεργαστή.



## 1.2 Εφαρμογές μικροελεγκτών

Οι μικροελεγκτές βρίσκουν εφαρμογή στα παρακάτω πεδία:

1. Σε συστήματα αυτοματισμών.
2. Σε κυκλώματα τηλεπικοινωνιών.
3. Στις ηλεκτρονικές συσκευές.
4. Στις ηλεκτρικές συσκευές.
5. Σε συστήματα τηλεματικής.
6. Σε συστήματα συλλογής δεδομένων (Data Acquisition).
7. Σε εφαρμογές ηλεκτρονικών ισχύος.
8. Σε συστήματα διασύνδεσης.
9. Σε εφαρμογές δικτύων Γενικότερα οι μικροελεγκτές χρησιμοποιούνται οπουδήποτε απαιτείται έλεγχος συστημάτων. Όταν λέμε ενσωματωμένα συστήματα (Embedded Systems) εννοούμε συστήματα τα οποία είναι βασισμένα σε μικροεπεξεργαστή (ή επίσης FPGA ή DSP).

### 1.3 Κατασκευαστές μικροελεγκτών

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.

### 1.4 Γλώσσα προγραμματισμού μικροελεγκτών

Οι μικροελεγκτές γενικά προγραμματίζονται σε γλώσσες χαμηλού επιπέδου. Τελευταία όλο και περισσότεροι προγραμματιστές επιλέγουν γλώσσες υψηλότερο επιπέδου. Ως γλώσσα χαμηλού επιπέδου ονομάζεται μια γλώσσα η οποία βρίσκεται πιο κοντά στο υλικό (γλώσσα μηχανής, assembly) Ως γλώσσα υψηλού επιπέδου ονομάζεται μια γλώσσα η οποία είναι αυστηρά δομημένη και υπάρχει συγκεκριμένος compiler ο οποίος μετατρέπει το πρόγραμμα σε γλώσσα μηχανής για το συγκεκριμένο μικροελεγκτή.

### 1.5 Γλώσσες χαμηλού επιπέδου

#### 1.5.1 Πλεονεκτήματα γλωσσών χαμηλού επιπέδου

- Ο προγραμματιστής έχει τον απόλυτο έλεγχο της συμπεριφοράς του μικροελεγκτή.
- Μπορεί να επιτύχει με απόλυτη ακρίβεια διάφορους χρονισμούς.
- Δεν απαιτείται η δαπάνη για την αγορά assembler καθώς συνήθως διατίθεται δωρεάν από την κατασκευάστρια εταιρεία.

### **1.5.2 Μειονεκτήματα γλωσσών χαμηλού επιπέδου**

- Απαιτείται μεγαλύτερος κόπος για την εκμάθηση της συμβολικής γλώσσας του εκάστοτε μικροελεγκτή.
- Τα προγράμματα που δημιουργούνται σε συμβολική γλώσσα δεν είναι ευανάγνωστα και ο προγραμματιστής δυσκολεύεται να θυμηθεί τη λογική που έχει εφαρμόσει όταν χρειάζεται να κάνει τροποποιήσεις εκ των υστέρων.
- Είναι δυσκολότερο να δουλέψουν πολλοί προγραμματιστές στο ίδιο πρόγραμμα.

### **1.6 Γλώσσες υψηλού επιπέδου**

#### **1.6.1 Πλεονεκτήματα γλωσσών υψηλού επιπέδου**

- Είναι ευκολότερη η ανάπτυξη μεγάλων και σύνθετων προγραμμάτων.
- Μπορούν να δουλέψουν πιο εύκολα πολλοί προγραμματιστές στο ίδιο πρόγραμμα.

#### **1.6.2 Μειονεκτήματα γλωσσών υψηλού επιπέδου**

- Σε εφαρμογές με κρίσιμους χρονισμούς είναι δυσκολότερη η συγγραφή κώδικα που ανταποκρίνεται στους χρονισμούς αυτούς
- Μερικές φορές η δαπάνη για την αγορά compiler δεν αποτελεί αμελητέο μέγεθος
- Σε παλιότερους compilers ο κώδικας μηχανής που παραγόταν δεν ήταν βελτιστοποιημένος με αποτέλεσμα να απαιτείται μικροελεγκτής με πολύ περισσότερη μνήμη. Οι compilers που κυκλοφορούν σήμερα διαθέτουν εξελιγμένα εργαλεία για βελτιστοποίηση (optimization) του κώδικα και έχουν κερδίσει την εμπιστοσύνη ακόμα και των πιο δύσπιστων προγραμματιστών.

### **1.7 Διαδεδομένες κατηγορίες μικροελεγκτών**

Λόγω του ισχυρότατου ανταγωνισμού αλλά και της τάσης ενσωμάτωσης των μικροελεγκτών σε κάθε ηλεκτρική και ηλεκτρονική συσκευή, η βιομηχανία μικροελεγκτών έχει καταλήξει στην παραγωγή ανταγωνιστικών μοντέλων μαζικής παραγωγής καθώς και μικροελεγκτών για πιο εξειδικευμένες εφαρμογές. Έτσι διακρίνονται οι εξής κυρίως κατηγορίες:

- Μικροελεγκτές (καμμιά φορά 4-bit αλλά συνήθως 8-bit) πολύ χαμηλού κόστους, γενικής χρήσης, με πολύ μικρό αριθμό ακροδεκτών (ακόμη και λιγότερους από 8). Σχεδιάζονται με έμφαση στη χαμηλή κατανάλωση ισχύος και την αυτάρκεια, ώστε

να χρειάζονται ελάχιστα ή και καθόλου εξωτερικά εξαρτήματα και να μη μπορεί να αντιγραφεί εύκολα το εσωτερικό λογισμικό τους. Απουσιάζει η δυνατότητα επέκτασης της μνήμης τους. Μερικά μοντέλα είναι ευρέως γνωστά στους ερασιτέχνες ηλεκτρονικούς, όπως πχ οι περισσότεροι μικροελεγκτές των σειρών PIC (Microchip), AVR (Atmel) και [8051](#) (Intel, Atmel, Dallas κτ).

- Μικροελεγκτές (συνήθως 8-bit αλλά και 16 ή 32-bit) χαμηλού κόστους, γενικής χρήσης, με μέτριο έως σχετικά μεγάλο αριθμό ακροδεκτών. Διαθέτουν μεγάλο αριθμό κοινών περιφερειακών, όπως θύρες UART, I<sup>2</sup>C, SPI ή CAN, μετατροπείς αναλογικού σε ψηφιακό και ψηφιακού σε αναλογικό. Στους κατασκευαστές της Άπω Ανατολής (Ιαπωνία, Κορέα), συνηθίζεται η ενσωμάτωση ελεγκτών οθόνης υγρών κρυστάλλων και πληκτρολογίου. Μερικές φορές παρέχουν δυνατότητα εξωτερικής επέκτασης της μνήμης τους.
- Μικροελεγκτές (κυρίως 32-bit) μέσου κόστους, γενικής χρήσης, με μεγάλο αριθμό ακροδεκτών. Χαρακτηρίζονται από έμφαση στην ταχύτητα εκτέλεσης εντολών, υψηλή αυτάρκεια περιφερειακών και μεγάλες δυνατότητες εσωτερικής ή εξωτερικής μνήμης προγράμματος (FLASH) και RAM. Στο χώρο αυτό έχουν ισχυρή παρουσία οι αρχιτεκτονικές με υψηλή μεταφερσιμότητα λογισμικού (portability) από τον ένα στον άλλο κατασκευαστή. Πχ μεταξύ των μικροελεγκτών τύπου [ARM](#) ή MIPS, το σύνολο των βασικών εντολών που αναγνωρίζει η ALU είναι ακριβώς το ίδιο, μειώνοντας έτσι τις μεγάλες αλλαγές στο λογισμικό, όταν στο μέλλον ο πελάτης υιοθετήσει ένα μικροελεγκτή άλλου κατασκευαστή (αρκεί, φυσικά, να υποστηρίζει κι αυτός το σύνολο εντολών ARM ή MIPS, αντίστοιχα).
- Μικροελεγκτές εξειδικευμένων εφαρμογών, οι οποίοι ενσωματώνουν συνήθως κάποιο εξειδικευμένο πρωτόκολλο επικοινωνίας το οποίο υλοποιείται πάντοτε σε hardware. Τέτοιοι μικροελεγκτές χρησιμοποιούνται σε τηλεπικοινωνιακές συσκευές όπως τα [μόντεμ](#).

Η μεγάλη μερίδα πωλήσεων των μικροελεγκτών εξακολουθεί να αφορά αυτούς των 8-bit, καθώς είναι η κατηγορία με το χαμηλότερο κόστος και το μικρότερο μέγεθος λογισμικού για το ίδιο αποτέλεσμα, ιδίως επειδή οι σύγχρονες οικογένειες μικροελεγκτών 8-bit έχουν πολύ βελτιωμένες επιδόσεις σε σχέση με το παρελθόν.

## ΚΕΦΑΛΑΙΟ 2: ΜΙΚΡΟΕΛΕΓΚΤΗΣ ARDUINO

### 2.1 Ιστορική αναδρομή Arduino

Το 2005 ξεκίνησε ένα σχέδιο προκειμένου να φτιαχτεί μια συσκευή για τον έλεγχο προγραμμάτων διαδραστικών σχεδίων από μαθητές, η οποία θα ήταν ποιο φθηνή από άλλα πρωτότυπα συστήματα διαθέσιμα εκείνη την περίοδο.

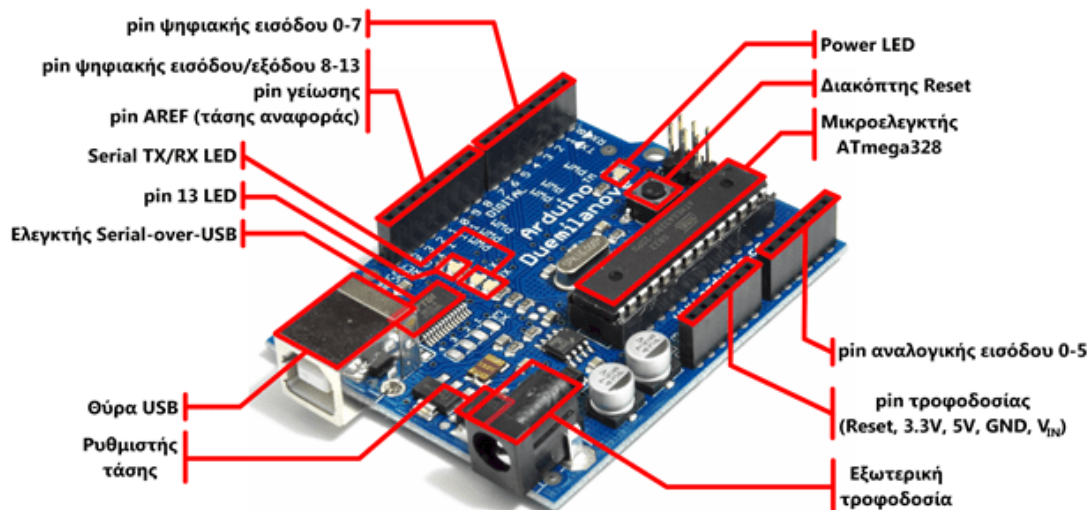
Οι ιδρυτές Massimo Banzi και David Cueartielles ονόμασαν το σχέδιο από τον Arduino της Inrea και ξεκίνησαν να παράγουν πλακέτες σε ένα μικρό εργοστάσιο στην Ιβρέα, κωμόπολη της επαρχίας Τορίνο στην περιοχή Πεδεμόντιο της βορειοδυτικής Ιταλίας - την ίδια περιοχή στην οποία στεγαζόταν η εταιρία υπολογιστών Olivetti.

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).

Τον Σεπτέμβριο του 2006 ανακοινώθηκε το Arduino Mini. Τον Οκτώβρη του 2008 ανακοινώθηκε το Arduino Duemilanove. Αρχικά βασίστηκε στο Atmel Atmega168, αλλά μετά στάλθηκε με το ATmega328. Τον Μάρτιο του 2009 ανακοινώθηκε το Arduino Mega. Είναι βασισμένο στο Atmel Atmega1280. Από τον Μάιο του 2011 πάνω από 30.000 Arduino ήταν σε χρήση σε όλο τον κόσμο. Τον Ιούλιο του 2012 ανακοινώθηκε το Arduino Leonardo. Είναι βασισμένο στο Atmel ATmega32u4. Τον Οκτώβριο του 2012 ανακοινώθηκε το Arduino Due. Είναι βασισμένο στο Atmel SAM3X8E, που είχε πυρήνα ARM CORTEX R3. Τον Νοέμβριο του 2012 ανακοινώθηκε το ArduinoMicro. Είναι βασισμένο στο AtmelATmega32u4. Τον Μάιο του 2013 ανακοινώθηκε το Arduino Robot. Είναι βασισμένο στο Atmel ATmega32u4 και ήταν το πρώτο επίσημο Arduino με ρόδες. Τον Μάιο του 2013 ανακοινώθηκε το ArduinoYun. Είναι Βασισμένο στο ATmega32u4 και στο Atheros AR9331 και ήταν το πρώτο προϊόν wifi που συνδύαζε το Arduino με το Linux.

### 2.2 Τι είναι ο ARDUINO

Ο Arduino είναι μία πλακέτα "ανοικτού κώδικα" με την οποία κάποιος χωρίς ιδιαίτερες γνώσεις προγραμματιστή μπορεί εύκολα να την χρησιμοποιήσει και να κατασκευάσει εφαρμογές ρομποτικής και συστήματα αυτοματισμού.



Ο Arduino αποτελείται από έναν μικροεπεξεργαστή, τον ATmega της Atmel και έχει την δυνατότητα να δεχθεί μονάδες εισόδου / εξόδου. Οι μονάδες εισόδου / εξόδου χωρίζονται σε Ψηφιακές & Αναλογικές. Υπάρχει μεγάλη πληθώρα συσκευών, συμβατές με τις πλακέτες Arduino. Κάποιες απο αυτές είναι: Αισθητήρες θερμοκρασίας, υγρασίας, δύναμης, απόστασης, γυροσκόπια, αξελερόμετρα, πίεσης κ.α. Επίσης με τον Arduino μπορείτε να ελέγξετε μοτοράκια DC, βηματικά (stepper) και servo ακόμη και leds, φώτα (220v), ρελέ και ότι άλλο μπορείτε να σκεφτείτε ώστε να δημιουργήσετε το δικό σας project.

Τον Arduino μπορείτε να τον προγραμματίσετε απο τον υπολογιστή σας μέσω της σειριακής θύρας που υποστηρίζει ο μικροεπεξεργαστής ATmega απο την θύρα USB του υπολογιστή. Η σειριακή αυτή σύνδεση (Serial over Usb) χρησιμοποιείται για την μεταφορά προγραμμάτων απο τον υπολογιστή προς την πλακέτα Arduino αλλά και το αντίστροφο για την μεταφορά των δεδομένων, που λαμβάνει ο Arduino απο της συσκευές, προς τον υπολογιστή.

Οι περισσότερες πλακέτες Arduino μπορούν να τροφοδοτηθούν από μπαταρία ή τροφοδοτικό, ακόμα και απο το ίδιο USB που χρησιμοποιούμε όπως είπαμε νωρίτερα για την μεταφορά του προγράμματος. Το voltage που μπορούμε να βάλουμε στον Arduino είναι από 9-12V χωρίς να υπάρχει απολύτως κανένα πρόβλημα ώστε να καεί η πλακέτα.



### 2.3 Πλεονεκτήματα ARDUINO

Το Arduino μπορεί να χρησιμοποιηθεί για τη δημιουργία προσωποποιημένων ηλεκτρονικών project, παίρνοντας εισερχόμενα δεδομένα από μια ποικιλία από εξερχόμενων δεδομένων. Τα προγράμματα μπορεί να είναι ανεξάρτητα (να εκτελούνται ηλαδή μόνο πάνω στην πλατφόρμα) ή μπορεί να χρησιμοποιηθούν για να προωθούν ή να λαμβάνουν πληροφορίες από εφαρμογές που τρέχουν στον υπολογιστή. Το λογισμικό που χρησιμοποιείται για να προγραμματιστεί η πλατφόρμα είναι ανοιχτού κώδικα και μπορεί να ληφθεί δωρεάν. Τα πλεονεκτήματα του μικροελεγκτή Arduino είναι τα παρακάτω:

- Χαμηλό κόστος: Τα Arduino είναι σχετικά φθηνά σε σύγκριση με άλλες πλατφόρμες. Σχεδόν όλα τα εξαρτήματα μπορούν να συναρμολογηθούν με το χέρι και ακόμα και τα έτοιμα δεν έχουν πολύ μεγάλο κόστος.
- Ξεκάθαρο προγραμματιστικό περιβάλλον: Το περιβάλλον του Arduino είναι εύκολο στην χρήση για αρχάριους, αλλά παρ' όλα αυτά αρκετά ευέλικτο για να το εκμεταλλευτούν οι μνημόνοι στον προγραμματισμό χρήστες.
- Ανεξάρτητο λειτουργικού συστήματος: Το λογισμικό τους τρέχει σε Windows, Macintosh OSX και Linux



- Ανοιχτού κώδικα και αναπτυσσόμενο λογισμικό: Το λογισμικό του **Arduino** εκδίδεται σαν ανοιχτού κώδικα εργαλεία, διαθέσιμα για ανάπτυξη από έμπειρους προγραμματιστές. Η γλώσσα μπορεί επίσης να επεκταθεί μέσω βιβλιοθηκών C++.

## 2.4 Πλακέτα ARDUINO

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 ενσωμάτωσή του σε άλλα κυκλώματα. 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 *bootloader*, έτσι ώστε να μην χρειάζεται εξωτερικός προγραμματιστής.

Σε εννοιολογικό επίπεδο, στην χρήση του Arduino software stack, όλα τα boards προγραμματίζονται με μία RS-232 σειριακή σύνδεση, αλλά ο τρόπος που επιτυγχάνεται αυτό διαφέρει σε κάθε hardware εκδοχή. Οι σειριακές πλάκες Arduino περιέχουν ένα απλό level shifter κύκλωμα για την μετατροπή του σήματος επιπέδου RS-232 σε TTL. Τα σημερινά Arduino προγραμματίζονται μέσω USB· αυτό καθίσταται δυνατό μέσω της εφαρμογής προσαρμογέων chip USB-to-Serial όπως το FTDI FT232. Κάποιες παραλλαγές, όπως το Arduino mini και το ανεπίσημο Boarduino, χρησιμοποιούν ένα αφαιρούμενο USB-to-Serial καλώδιο ή board, Bluetooth ή άλλες μεθόδους. (Όταν χρησιμοποιείται με παραδοσιακά εργαλεία microcontroller αντί για το Arduino IDE, χρησιμοποιείται πρότυπος προγραμματισμός AVR ISP).












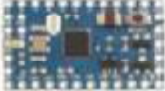


Ο πίνακας Arduino εκθέτει τα περισσότερα microcontroller I/O pins για χρήση από άλλα κυκλώματα. Τα Diecimila, Duemilanove και το τρέχον Uno παρέχουν 14 ψηφιακά I/O pins, έξι από τα οποία μπορούν να παράγουν pulse-width διαμορφωμένα σήματα, και έξι αναλογικά δεδομένα. Αυτά τα pins βρίσκονται στην κορυφή του πίνακα μέσω female headers 0.1 ιντσών (2,2mm). Διάφορες εφαρμογές ασπίδων plug-in είναι εμπορικά διαθέσιμες.

## 2.5 Επίσημες πλακέτες

Το πρωτότυπο υλικολογισμικό του Arduino κατασκευάζεται από την Ιταλική εταιρία Smart Projects.<sup>[9]</sup> Κάποιες πλακέτες με την μάρκα του Arduino έχουν σχεδιαστεί από την Αμερικάνικη εταιρία SparkFun Electronics.<sup>[10]</sup> Δεκαέξι εκδοχές του Arduino Hardware έχουν χρησιμοποιηθεί εμπορικά μέχρι τώρα:

1. Το Serial Arduino, προγραμματισμένο με μία σειριακή DE-9 σύνδεση χρησιμοποιώντας τεχνολογία ATmega8.
2. Το Arduino Extreme, με ένα USB interface για προγραμματισμό χρησιμοποιώντας τεχνολογία ATmega8.
3. Το Arduino Mini, μία έκδοση μινιατούρας του Arduino χρησιμοποιώντας τεχνολογία surface-mounted ATmega168.

4. Το Arduino Nano, ένα ακόμα πιο μικρό, USB τροφοδοτούμενη εκδοχή του Arduino χρησιμοποιώντας τεχνολογία surface-mounted ATmega168 (ATmega328 για την νεότερη έκδοση).
5. Το LilyPad Arduino, ένα μινιμαλιστικό σχέδιο για εφαρμογές ένδυσης και E-textiles χρησιμοποιώντας τεχνολογία surface-mounted ATmega328.
6. Το Arduino NG, με ένα USB interface για προγραμματισμό και χρησιμοποιώντας τεχνολογία ATmega8.
7. Το Arduino NG plus, με ένα USB interface για προγραμματισμό και χρησιμοποιώντας τεχνολογία ATmega168.
8. Το Arduino Bluetooth, με Bluetooth interface για προγραμματισμό χρησιμοποιώντας τεχνολογία ATmega168.
9. Το Arduino Diecimila, με ένα USB interface και χρησιμοποιεί τεχνολογία ATmega168 σε ένα DIP28 πακέτο.
10. Το Arduino Duemilanove (“2009”), χρησιμοποιεί τεχνολογία ATmega168 (ATmega328 για την καινούργια έκδοση) και τροφοδοτείται μέσω ενέργειας USB/DC, αυτόματα εναλλασσόμενης.
11. Το Arduino Mega, χρησιμοποιώντας τεχνολογία surface-mounted ATmega1280 για περαιτέρω I/O και μνήμη<sup>[11]</sup>.
12. Το Arduino Uno, χρησιμοποιώντας την ίδια τεχνολογία ATmega328 όπως το τελευταίο μοντέλο Duemilanove, αλλά ενώ το Duemilanove χρησιμοποιεί ένα FTDI chipset για το USB, το Uno χρησιμοποιεί τεχνολογία ATmega8U2 προγραμματισμένο ως σειριακός μετατροπέας.
13. Το Arduino Mega2560, χρησιμοποιεί τεχνολογία surface-mounted ATmega2560 φέρνοντας την ολική μνήμη στα 256kB. Επίσης ενσωματώνει τη νέα τεχνολογία ATmega8U2 (ATmega16U2 σε αναθεώρηση τύπου 3) USB chipset.
14. Το Arduino Leonardo, με ένα ATmega32U4 chip που εξαλείφει την ανάγκη για συνδεσιμότητα μέσω USB και μπορεί να χρησιμοποιηθεί ως ψηφιακό πληκτρολόγιο ή ποντίκι. Κυκλοφόρησε στο Maker Faire Bay Area το 2012.
15. Το Arduino Esplora, με εμφάνιση που παραπέμπει σε χειριστήριο κονσόλας βιντεοπαιχνιδιών με joystick και ενσωματωμένους αισθητήρες για ήχο, φως, θερμοκρασία και επιτάχυνση.
16. Το Arduino Due είναι ένα μικροχειριστήριο board βασισμένο στην τεχνολογία Atmel SAM3X8E ARM Cortex-M3 CPU. Είναι το πρώτο board της Arduino βασισμένη σε επεξεργαστή 32-bit ARM microcontroller<sup>[12][6]</sup>.

				
Arduino Uno	Arduino Leonardo	Arduino Mega 2560	Arduino LilyPad	Arduino Mega ADK
				
Arduino Fio	Arduino Ethernet	Arduino Pro	Arduino BT	Arduino Nano
				
USB/Serial Light Adapter	Arduino Mini	Mini USB/Serial Adapter	Arduino Pro Mini	

## 2.6 Λογισμικό- Προγραμματισμός

### 2.6.1 Λογισμικό ARDUINO

```

Blink | Arduino 1.0.2
Αρχείο Επεξεργασία Σχέδιο Εργαλεία Βοήθεια
✓ ↻ 📄 ⬆️ ⬇️
Blink
/*
 * Blink
 * Turns on an LED on for one second, then off for one second, repeatedly.
 *
 * This example code is in the public domain.
 */

// Pin 13 has an LED connected on most Arduino boards.
// give it a name:
int led = 13;

// the setup routine runs once when you press reset:
void setup() {
  // initialize the digital pin as an output.
  pinMode(led, OUTPUT);
}

// the loop routine runs over and over again forever:
void loop() {
  digitalWrite(led, HIGH); // turn the LED on (HIGH is the voltage level)
  delay(1000);             // wait for a second
  digitalWrite(led, LOW);  // turn the LED off by making the voltage LOW
  delay(1000);             // wait for a second
}

```

Το συγκεκριμένο πρόγραμμα που βλέπετε στην εικόνα δεξιά το οποίο μπορείτε να βρείτε στον Compiler της Arduino (File -> Sketchbook -> Examples -> Digital -> Blink) αναβοσβήνει ένα Led. Αναλυτικά δηλώνει την μεταβλητή "led" ότι αντιστοιχεί στο pin 13 του Arduino. Στην συνέχεια μέσα στην setup δηλώνεται ότι η μεταβλητή "led" είναι OUTPUT. Και τέλος

στο main προγραμμα δηλαδή στην loop , με την μεταβλητή digitalWrite(led, HIGH) στέλνει στο pin 13 ρεύμα 5V οπότε και το Led ανάβει. Αμέσως μετά κάνει μια διακοπή ενός δευτερολέπτου και πάλι με την εντολή digitalWrite(led, LOW) σβήνει το Led και κάνει μια παύση ενός δευτερολέπτου. Ο κώδικας που βρίσκεται μέσα στην loop εκτελείται συνέχεια με αποτέλεσμα το Led να αναβοσβήνει.

Το ολοκληρωμένο περιβάλλον ανάπτυξης (IDE) του Arduino είναι μία εφαρμογή γραμμένη σε Java, που λειτουργεί σε πολλές πλατφόρμες και προέρχεται από το IDE για τη γλώσσα προγραμματισμού Processing και το σχέδιο Wiring. Έχει σχεδιαστεί για να εισαγάγει στον προγραμματισμό τους καλλιτέχνες και τους νέους που δεν είναι εξοικειωμένοι με την ανάπτυξη λογισμικού. Περιλαμβάνει ένα πρόγραμμα επεξεργασίας κώδικα με χαρακτηριστικά όπως είναι η επισήμανση σύνταξης και ο συνδυασμός αγκύλων και είναι επίσης σε θέση να μεταγλωττίζει και να φορτώνει προγράμματα στην πλακέτα με ένα μόνο κλικ. Δεν υπάρχει συνήθως καμία ανάγκη να επεξεργαστείτε αρχεία make ή να τρέξετε προγράμματα σε ένα περιβάλλον γραμμής εντολών. Ένα πρόγραμμα ή κώδικας που γράφτηκε για Arduino ονομάζεται σκίτσο (sketch).

Τα Arduino προγράμματα είναι γραμμένα σε C ή C++. Το Arduino IDE έρχεται με μια βιβλιοθήκη λογισμικού που ονομάζεται "Wiring", από το πρωτότυπο σχέδιο Wiring, γεγονός που καθιστά πολλές κοινές λειτουργίες εισόδου/εξόδου πολύ πιο εύκολες.

## 2.6.2 Προγραμματισμός- Βασικές λειτουργίες

### Δήλωση Μεταβλητών

Όπως σε όλες τις γλώσσες προγραμματισμού, μπορούμε να δηλώσουμε ονόματα μεταβλητών. Οι τύποι μεταβλητών που υποστηρίζονται στο Arduino είναι αρκετοί:

- byte, με τιμές από 0 έως και 255
- float, δεκαδικοί αριθμοί
- boolean, με τιμές το 0 και (True-False)
- int, ακέραιος με δυνατές τιμές από -32768 έως και 32767
- char, ένας χαρακτήρας (μέγεθος ένα Byte)
- long, ακέραιος με δυνατές τιμές από -2147483648 έως και 2147483647
- string, πίνακας χαρακτήρων

### Θύρες εισόδου/εξόδου

Το Arduino Uno R3 έχει 14 ψηφιακές θύρες εισόδου ή εξόδου (digital input/output pins) και έξι αναλογικές εισόδους (analog input pins). Οι 14 ψηφιακές θύρες ονομάζονται με νούμερα από το 0 έως το 13, ενώ οι έξι αναλογικές με το γράμμα A ακολουθούμενο από ένα νούμερο από 0 μέχρι το 5 (π.χ. A3). Στην έξοδο τα pins μπορούν να δώσουν 0 έως και 5V τάση. Από τις 14 ψηφιακές θύρες οι έξι, και ειδικότερα οι 3, 5, 6, 9, 10, 11, είναι και PWM

θύρες (Pulse Width Modulation), δηλαδή μπορούν να προσομοιώσουν αναλογικές εξόδους.

Έτσι, συνοπτικά για την είσοδο και έξοδο έχουμε:

- Για ψηφιακή είσοδο, χρησιμοποιούμε τις 14 ψηφιακές 0..13. Όταν δουλεύουν ψηφιακά, η είσοδος μπορεί να είναι ή 0 ή 5V, με τον χαρακτηρισμό LOW ή HIGH όπως θα δούμε παρακάτω.
- Για ψηφιακή έξοδο, χρησιμοποιούμε τις 14 ψηφιακές 0..13. Όταν δουλεύουν ψηφιακά, η έξοδος μπορεί να είναι 0 ή 5V, με τον χαρακτηρισμό LOW ή HIGH όπως θα δούμε παρακάτω.
- Για αναλογική είσοδο, δηλαδή να διαβάσουμε τιμές ρεύματος στο διάστημα 0 έως 5V, χρησιμοποιούμε τις έξι αναλογικές θύρες A0..A5.
- Για αναλογική έξοδο, μπορούμε να χρησιμοποιήσουμε τις έξι PWM ψηφιακές θύρες (3, 5, 6, 9, 10, 11), οι οποίες θα μας δώσουν ρεύμα εξόδου όποιας τιμή θέλουμε στο διάστημα από 0 έως 5V.

Γράφοντας κώδικα θα πρέπει να αρχικοποιήσουμε τις θύρες που χρησιμοποιούμε με τη συνάρτηση `pinMode()`, δηλαδή να δίνουμε την πληροφορία για όποιες χρησιμοποιήσουμε αν θα είναι για είσοδο ή για έξοδο.

Η συνάρτηση αυτή αναλύεται στην επόμενη ενότητα. Όταν χρησιμοποιείται η σειριακή οθόνη παρακολούθησης της επικοινωνίας με τον υπολογιστή, χρησιμοποιούνται τα pins 0 και 1 για αυτό, οπότε προτείνουμε να μην τα χρησιμοποιείτε στις εφαρμογές σας, εκτός αν αυτό είναι απαραίτητο (π.χ. δεν μας φτάνουν τα υπόλοιπα 12 pins για την εφαρμογή μας). Επίσης, στη θύρα 13 υπάρχει συνήθως συνδεδεμένο ήδη ένα Led πάνω στην πλακέτα Arduino Uno, κι έτσι μπορούμε να το χρησιμοποιούμε για σχετικές λειτουργίες.

Οι χρήστες πρέπει μόνο 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 που καλείται συνέχεια μέχρι η πλακέτα να απενεργοποιηθεί.

#### Σχόλια

Μπορούν να χρησιμοποιηθούν δύο κάθετες `//` για να γράψουμε κάποιο σχόλιο σε μία γραμμή (ό,τι ακολουθεί τις `//` αγνοείται), ή τα `/* */` που περικλείουν τα σχόλια που γράφονται σε περισσότερες γραμμές (ό,τι υπάρχει ανάμεσα στο `/*` και στο `*/` αγνοείται).

## Διαχείριση pins

Η κύρια λειτουργία του μικροελεγκτή βασίζεται στο να ελέγχει τις θύρες που διαθέτει και είτε να δίνει ρεύμα είτε να παίρνει ρεύμα από αυτές. Στην αρχικοποίηση κάθε προγράμματος (μέσα στη συνάρτηση `setup`) θα χρειαστεί να χαρακτηρίσουμε τα Pins που χρησιμοποιούμε ως είσοδο ή ως έξοδο. Η συνάρτηση `pinMode` (Pin, Mode) χρησιμοποιείται με το όνομά της και ορίσματα α) τον αριθμό Pin και β) την κατάσταση λειτουργίας που χαρακτηρίζεται με τη λέξη `INPUT` (είσοδος) ή `OUTPUT` (έξοδος). Όπως έχουμε αναφέρει έχουμε 14 ψηφιακά Pins, 6 εκ των οποίων είναι PWM, με ονόματα 0-13 και έξι αναλογικά με ονόματα A0-A5.

## Ψηφιακή είσοδος

Και τα 14 ψηφιακά pins του Arduino μπορούν να δουλεύουν ως ψηφιακές εισόδους, δηλαδή να “διαβάσουν” ως είσοδο τάση με τιμή είτε 0V είτε 5V. Αυτό γίνεται με χρήση της συνάρτησης `digitalRead`(Pin), όπου το όρισμα Pin αναφέρεται στο νούμερο της θύρας για την οποία θα πάρουμε είσοδο, ενώ η συνάρτηση επιστρέφει με το όνομά της την τιμή εισόδου. Η τάση εισόδου μπορεί να είναι 0V ή 5V, οι οποίες αναπαρίστανται με προκαθορισμένες τιμές στην τιμή που διαβάζουμε:

-LOW : όταν λάβει τάση 0 V στην είσοδο (pin)

-HIGH : όταν λάβει τάση 5 V στην είσοδο (pin)

## Ψηφιακή έξοδος

Και τα 14 pins του Arduino μπορούν να δουλεύουν ως ψηφιακές εξόδους, δηλαδή δίνουν έξοδο 0V ή 5V. Αυτό γίνεται με χρήση της συνάρτησης `digitalWrite`(Pin, Value), όπου το όρισμα Pin αναφέρεται στο νούμερο της θύρας για την οποία θα δώσουμε τάση εξόδου, ενώ η τάση εξόδου μπορεί να είναι 0V ή 5V, οι οποίες αναπαρίστανται με προκαθορισμένες τιμές στην παράμετρο `value`

-LOW : θα δώσει 0 V στην έξοδο (pin)

-HIGH : θα δώσει 5 V στην έξοδο (pin)

## Αναλογική είσοδος

Το Arduino έχει 6 αναλογικές εισόδους, οι οποίες χαρακτηρίζονται με τα σύμβολα A0, A1, A2, A3, A4, A5. Μπορούμε να συνδέσουμε κάποιο αναλογικό εξάρτημα (π.χ. αναλογικός αισθητήρας LM 35) και να το διαβάσουμε ως είσοδο. Αυτό γίνεται με χρήση της συνάρτησης `analogRead` (Pin), όπου το όρισμα Pin αναφέρεται στο νούμερο της θύρας για την οποία θα πάρουμε είσοδο, ενώ η συνάρτηση επιστρέφει με το όνομά της την τιμή εισόδου. Η τιμή εισόδου κυμαίνεται από 0 μέχρι και 1023. Συνήθως

χρησιμοποιούμε μια μεταβλητή για να καταχωρήσουμε την τιμή.

#### Αναλογική έξοδος

Κάποια από τα 14 Pins του Arduino έχουν την ένδειξη PWM, δηλαδή μπορούν να προσομοιώσουν την αναλογική έξοδο μέσω παλμοκωδικής διαμόρφωσης.

Έτσι, με τιμές από το 0 μέχρι το 255 προσομοιώνουμε (αναλογικά) το διάστημα από 0 έως 5V. Αυτό γίνεται με χρήση της συνάρτησης `analogWrite(Pin, Value)`, όπου το όρισμα Pin αναφέρεται στο νούμερο της θύρας για την οποία θα δώσουμε ρεύμα εξόδου, ενώ η τάση εξόδου κυμαίνεται από 0 V μέχρι και 5 V, οι οποίες τιμές της τάσης αναλογικά αναπαρίστανται με τιμές στη μεταβλητή value. Τιμή 0 δίνει 0V στην έξοδο (pin), τιμή 255 δίνει τάση 5V στην έξοδο (pin), ενώ αναλογικά μπορούμε να δώσουμε ενδιάμεσες τάσεις.

#### Συνάρτηση delay(καθυστέρησης)

Στο πρόγραμμά μας μπορούμε να ορίσουμε μια καθυστέρηση ώστε να διαρκέσει για το χρόνο που εμείς ορίζουμε ένα γεγονός. Αυτό το επιτυγχάνουμε με χρήση της συνάρτησης `delay(time)` όπου στη θέση time δίνουμε το χρόνο σε ms (1/1000 sec). Η εντολή `delay(time)` σημαίνει ότι σταματά στο σημείο αυτό η εκτέλεση του προγράμματός μας για το χρόνο time που εμείς ορίζουμε. Έχοντας υπόψιν όσα αναφέρθηκαν θα αναλυθεί στην επόμενη υπό ενότητα ο τρόπος με τον οποίο χρησιμοποιήθηκε στην εργασία αυτή ο Arduino UNO.

### **Βασικές δομές και λειτουργίες προγραμματισμού**

Παρακάτω, ακολουθούν μερικές από τις πιο βασικές δομές και λειτουργίες που μπορεί να αξιοποιηθεί ως εργαλεία κατά την συγγραφή ενός προγράμματος Arduino :

#### **Δομές ελέγχου ροής**

- `if` (δομή ελέγχου μίας συνθήκης)
- `if ... else` (δομή ελέγχου πολλαπλών συνθηκών)
- `for` (δομή επαναληπτικού ελέγχου συνθήκης)
- `while` (δομή επαναληπτικού ελέγχου συνθήκης)
- `do ... while` (δομή επαναληπτικού ελέγχου συνθήκης)
- `switch .. case` (δομή ελέγχου περιπτώσεων)
- `break` (εντολή διακοπής μιας επαναληπτικής δομής)
- `continue` (εντολή παράλειψης της τρέχουσας επανάληψης)
- `return` (εντολή επιστροφής από μία συνάρτηση)

goto (εντολή μετάβασης σε κάποιο σημείο του κώδικα)

### **Αριθμητικοί τελεστές**

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

### **Λογικοί τελεστές**

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

### **Δυαδικοί τελεστές**

- & (δυαδική σύζευξη)
- | (δυαδική διάζευξη)
- ^ (δυαδική αποκλειστική διάζευξη)
- ~ (δυαδική άρνηση)
- << (δυαδική αριστερή ολίσθηση)
- >> (δυαδική δεξιά ολίσθηση)

### **Τελεστές αύξησης και μείωσης**

- ++ (αύξηση κατά μία ακέραιη μονάδα)
- (μείωση κατά μία ακέραιη μονάδα)

### **Σύνθετοι τελεστές**

- +=, -=, \*=, /=, %= (σύνθετοι αριθμητικοί τελεστές)
- &=, |=, ^=, ~=, <<=, >>= (σύνθετοι δυαδικοί τελεστές)

### **Τελεστές σύγκρισης**

- == (ισότητα)
- != (ανισότητα)
- < (μικρότερο)
- > (μεγαλύτερο)
- <= (μικρότερη ή ίση)
- >= (μεγαλύτερη ή ίση)



## Τελεστές δεικτών

- \* (τελεστής απόκτησης περιεχομένου)
- & (τελεστής απόκτησης διεύθυνσης)

## Σταθερές

- HIGH (τιμή υψηλής στάθμης για μία επαφή εισόδου ή εξόδου)
- LOW (τιμή χαμηλής στάθμης για μία επαφή εισόδου ή εξόδου)
- false (λογικό επίπεδο ψεύδους σε μία συνθήκη)
- true (λογικό επίπεδο αλήθειας σε μία συνθήκη)
- INPUT (χρησιμοποιείται για τον ορισμό μίας επαφής ως είσοδο)
- OUTPUT (χρησιμοποιείται για τον ορισμό μίας επαφής ως έξοδο)
- A0, ..., A5 (συμβολοσταθερές για τις αναλογικές επαφές εισόδου)

## Τύποι δεδομένων

- boolean (λογική δυαδική τιμή)
- char (προσημασμένος χαρακτήρας 8 ψηφίων)
- unsigned char (μη προσημασμένος χαρακτήρας 8 ψηφίων)
- byte (μη προσημασμένος χαρακτήρας 8 ψηφίων)
- int (προσημασμένος ακέραιος αριθμός 16 ψηφίων)
- unsigned int (μη προσημασμένος ακέραιος αριθμός 16 ψηφίων)
- word (μη προσημασμένος ακέραιος αριθμός 16 ψηφίων)
- long (προσημασμένος ακέραιος αριθμός 32 ψηφίων)
- unsigned long (μη προσημασμένος ακέραιος αριθμός 32 ψηφίων)
- float, double (αριθμός κινητής υποδιαστολής απλής ακρίβειας)
- String (αντικείμενο αλφαριθμητικού με χρήσιμες μεθόδους)
- Ωλφαριθμητικό μπορεί να θεωρηθεί και ο πίνακας χαρακτήρων

## Συναρτήσεις μετατροπής τύπων

- char(), byte()
- int(), word(), long()
- float(), double()

## Συναρτήσεις εισόδου και εξόδου

- pinMode() (ορίζει μια επαφή ως είσοδο ή έξοδο)

Συναρτήσεις ψηφιακής εισόδου και εξόδου

- digitalWrite() (γράφει σε μία ψηφιακή επαφή εξόδου)
- digitalWrite() (διαβάζει από μία ψηφιακή επαφή εισόδου)

### **Συναρτήσεις αναλογικής εισόδου και εξόδου**

- analogReference()  
(ορίζει την τάση αναλογικής αναφοράς)
- analogWrite() (γράφει PWM σήματα σε μία επαφή εξόδου)
- analogRead() (διαβάζει από μία αναλογική επαφή εισόδου)

### **Προηγμένες συναρτήσεις εισόδου και εξόδου**

- tone() (παράγει ένα τετραγωνικό σήμα ορισμένης συχνότητας)
- noTone() (διακόπτει την παραγωγή τετραγωνικών σημάτων)
- shiftOut() (ολισθαίνει τα ψηφία μιας τιμής σε μία επαφή εξόδου)
- pulseIn() (επιστρέφει την διάρκεια σε ms ενός παλμού HIGH ή LOW)

### **Συναρτήσεις χρόνου**

- millis() (διάρκεια εκτέλεσης του προγράμματος σε ms)
- micros() (διάρκεια εκτέλεσης του προγράμματος σε μs)
- delay() (παύση προγράμματος - η διάρκεια δίδεται σε ms)
- delayMicroseconds() (παύση προγράμματος - η διάρκεια δίδεται σε μs)

### **Μαθηματικές και Τριγωνομετρικές συναρτήσεις**

- max() (βρίσκει τον μεγαλύτερο ανάμεσα σε δύο αριθμούς)
- min() (βρίσκει τον μικρότερο ανάμεσα σε δύο αριθμούς)
- abs() (επιστρέφει την απόλυτη τιμή ενός αριθμού)
- constrain() (ελέγχει για υπερχείλιση ή υποχείλιση ορίων)
- map() (πραγματοποιεί γραμμικό μετασχηματισμό ορίων)
- pow() (επιστρέφει το αποτέλεσμα μίας δύναμης)
- sqrt() (επιστρέφει την ρίζα ενός αριθμού)
- sin() (υπολογίζει το ημίτονο ενός αριθμού)
- cos() (υπολογίζει το συνημίτονο ενός αριθμού)
- tan() (υπολογίζει την εφαπτομένη ενός αριθμού)

### **Συναρτήσεις γεννήτριας ψευδοτυχαίων αριθμών**

- random() (δίδεται ένας νέος αριθμός από την γεννήτρια)
- randomSeed() (θέτει τον σπόρο της γεννήτριας παραγωγής)

## Συναρτήσεις επεξεργασίας δυαδικών αριθμών

- lowByte() (επιστρέφει το δεξιότερο byte μίας μεταβλητής)
- highByte() (επιστρέφει το αριστερότερο byte μίας μεταβλητής)
- bitRead() (διαβάζει ένα συγκεκριμένο ψηφίο μίας μεταβλητής)
- bitWrite() (γράφει σε ένα συγκεκριμένο ψηφίο μιας μεταβλητής)
- bitSet() (γράφει την τιμή 1 σε κάποιο ψηφίο μίας μεταβλητής)
- bitClear() (γράφει την τιμή 0 σε κάποιο ψηφίο μιας μεταβλητής)
- bit() (υπολογίζει μία συγκεκριμένη δύναμη με βάση το 2)

## Συναρτήσεις χρήσης ρουτινών εξυπηρέτησης διακοπών

- attachInterrupt() (ενεργοποιεί μία ρουτίνα εξυπηρέτησης διακοπής)
- detachInterrupt() (απενεργοποιεί μία ρουτίνα εξυπηρέτησης διακοπής)

## Συναρτήσεις ενεργοποίησης και απενεργοποίησης διακοπών

- interrupts() (ενεργοποιεί τα σήματα διακοπής)
- noInterrupts() (απενεργοποιεί τα σήματα διακοπής)

## Υποστήριξη σειριακής επικοινωνίας

- Serial (αντικείμενο σειριακής επικοινωνίας με χρήσιμες μεθόδους)

## 2.7 Επεκτάσεις ARDUINO (ARDUINO shields)

ArduinoShields είναι κάποιες επεκτάσεις από την αρχική πλακέτα όπου μας δίνουν επιπλέον δυνατότητες στον μικροεπεξεργαστή μας. Κάποιες από τις εκδόσεις του Arduino έχουν προεγκατεστημένες κάποιες από αυτές τις επεκτάσεις κάτι που τα κάνει ακόμα πιο εύρηστα. Οι επεκτάσεις αυτές είναι εύκολο να τοποθετηθούν και είναι σχετικά φθηνές στην παραγωγή τους. Υπάρχουν πάρα πολλές επεκτάσεις, εγώ θα αναφέρω τις πιο βασικές, οι οποίες είναι οι εξής :

### ArduinoWiFi Shield

Το ArduinoWifi Shield συνδέει το arduino στο διαδίκτυο ασύρματα.

### Arduino Ethernet Shield

Το Arduino Ethernet Shield συνδέει το arduino στο διαδίκτυο μέσω ενός καλωδίου RJ-45.

### Wireless SD Shield





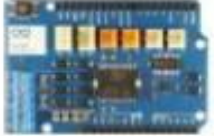
Το WirelessSDShield επιτρέπει σε μια πλακέτα Arduino να επικοινωνεί ασύρματα με μια ασύρματη μονάδα. Η μονάδα μπορεί να επικοινωνήσει έως και 100 πόδια σε εσωτερικούς χώρους και 300 πόδια σε εξωτερικούς χώρους. Η μονάδα αυτή περιλαμβάνει επίσης μια θύρα υποδοχής SD.

□ **Wireless Proto Shield**

Το WirelessProtoShield επιτρέπει σε μια πλακέτα Arduino να επικοινωνεί ασύρματα με μια ασύρματη μονάδα. Η μονάδα μπορεί να επικοινωνήσει έως και 100 πόδια σε εσωτερικούς χώρους και 300 πόδια σε εξωτερικούς χώρους.

□ **Arduino Motor Shield**

Το ArduinoMotorShield επιτρέπει την διαχείριση δυο DC κινητήρων από την ίδια συσκευή, ελέγχοντας την ταχύτητα και την κατεύθυνση του καθενός ξεχωριστά.

Arduino WiFi Shield	
Arduino Ethernet Shield	
Wireless SD Shield	
Wireless Proto Shield	
Arduino Motor Shield	

### 2.7.1 Αναλυτική Περιγραφή

Όπως είπαμε και προηγουμένως η Ethernet Shield μας παρέχει την δυνατότητα να ενώσουμε το Arduino μας με το διαδίκτυο. Βασίζετε στο μικροτσιπ της Wiznet W5100. Αυτό το τσιπ μας παρέχει στοίβα δικτύου IP ικανή να ανταποκριθεί τόσο σε TCP όσο και σε UDP αλλά και Pv4, ICMP, ARP, IGMP, PPPoE. Υποστηρίζει τέσσερις ταυτόχρονες συνδέσεις.

Η Ethernet Shield έχει στάνταρ RJ-45 σύνδεση, με την καινούρια έκδοση της πλακέτας να παρέχει και POE, power over Ethernet. Στην ασπίδα που θα χρησιμοποιήσουμε εμείς δεν υπάρχει αυτή η λειτουργία. Υπάρχει και ένας διακόπτης επανεκκίνησης (reset button) για να διασφαλίσει ότι το Ethernet τσιπ της πλακέτας λειτουργεί σωστά κατά την εκκίνηση. Αυτός ο διακόπτης εφόσον η πλακέτας μας λειτουργεί και είναι συνδεδεμένη με το Arduino σε περίπτωση χρήσης του θα κάνει reset και το δεύτερο.

Εικόνα 26 Wiznet Ethernet W5100 chip



Τεχνικά η σύνδεση μεταξύ των πλακετών γίνεται με μεγάλους συρμάτινους ακροδέκτες που έχει στο κάτω μέρος η Ethernet Shield και ενώνετε με τις υποδοχές Εφαρμογή Arduino των ακροδεκτών της από κάτω πλακέτας. Έτσι η διάταξη παραμένει ίδια και κατά συνέπεια μπορούμε να συνδέσουμε εκ νέου πλακέτα από πάνω.

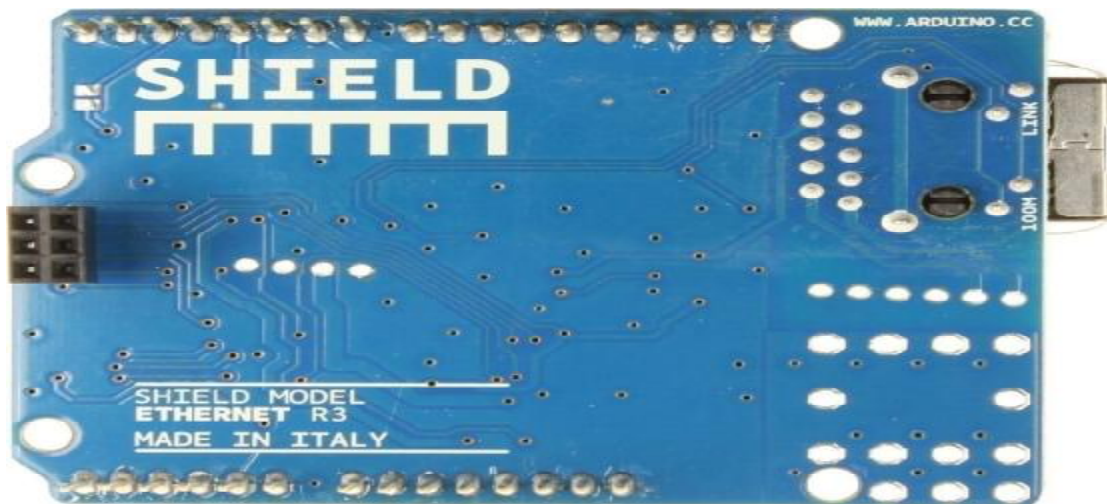
Επίσης υπάρχει πάνω στο shield υποδοχή για κάρτα αποθήκευσης δεδομένων micro sd την οποία μπορούμε να την χρησιμοποιήσουμε για να αποθηκεύσουμε δεδομένα που μεταφέρουμε στο διαδίκτυο. Στην συγκεκριμένη εργασία θα δούμε παρακάτω ότι θα το χρησιμοποιήσουμε ως datalogger για να αποθηκεύουμε πληροφορίες από τους αισθητήρες μας.

Το Arduino επικοινωνεί με το Wiznet W510 ethernet chip τόσο και με την SD κάρτα από τον δίαυλο SPI διαμέσου της ICSP επικεφαλίδας.

Η πρώτη σύνδεση μας δεσμεύει τους ψηφιακούς ακροδέκτες 10,11,12 και 13 για το UNO και αυτή της SD κάρτας τον ακροδέκτη 4. Έτσι αυτοί οι ακροδέκτες δεν μπορούν να είναι διαθέσιμοι για χρήση γενικού σκοπού. Για αυτό το λόγο πρέπει μέσα στο πρόγραμμα μας με τις κατάλληλες εντολές και βιβλιοθήκες να επιλέξουμε τους συγκεκριμένους ακροδέκτες για να υποδηλώσουμε ότι χρησιμοποιούμε το Ethernet τσιπ αλλά και την λειτουργία Εφαρμογή Arduino της κάρτας sd. (Θα αναφερθούμε σε ξεχωριστό κεφάλαιο για βιβλιοθήκες και συναρτήσεις-εντολές που υπάρχουν και κάθε ένα από τους σκοπούς αυτούς).

Τέλος να αναφέρουμε ότι η ασπίδα μας έχει διάφορα λαμπάκια για συγκεκριμένες ενδείξεις:

- PWR: Μας δείχνει ότι η ασπίδα και το Arduino έχουν τροφοδοσία
- LINK: Υποδεικνύει την ύπαρξη σύνδεσης με το δίκτυο και αναβοσβήνει όταν η ασπίδα δέχεται ή αποστέλνει δεδομένα.
- FULLD: Μας αναφέρει ότι η σύνδεση στο δίκτυο είναι πλήρης αμφίδρομη ταυτόχρονη (full duplex).
- 100M: Υποδεικνύει την ύπαρξη 100Mb/s σύνδεσης στο δίκτυο.
- RX: Λάμπει όταν η ασπίδα δέχεται δεδομένα.
- TX: Λάμπει όταν η ασπίδα στέλνει δεδομένα.
- COLL: Λάμπει όταν υπάρχουν συγκρούσεις μέσα στο δίκτυο.



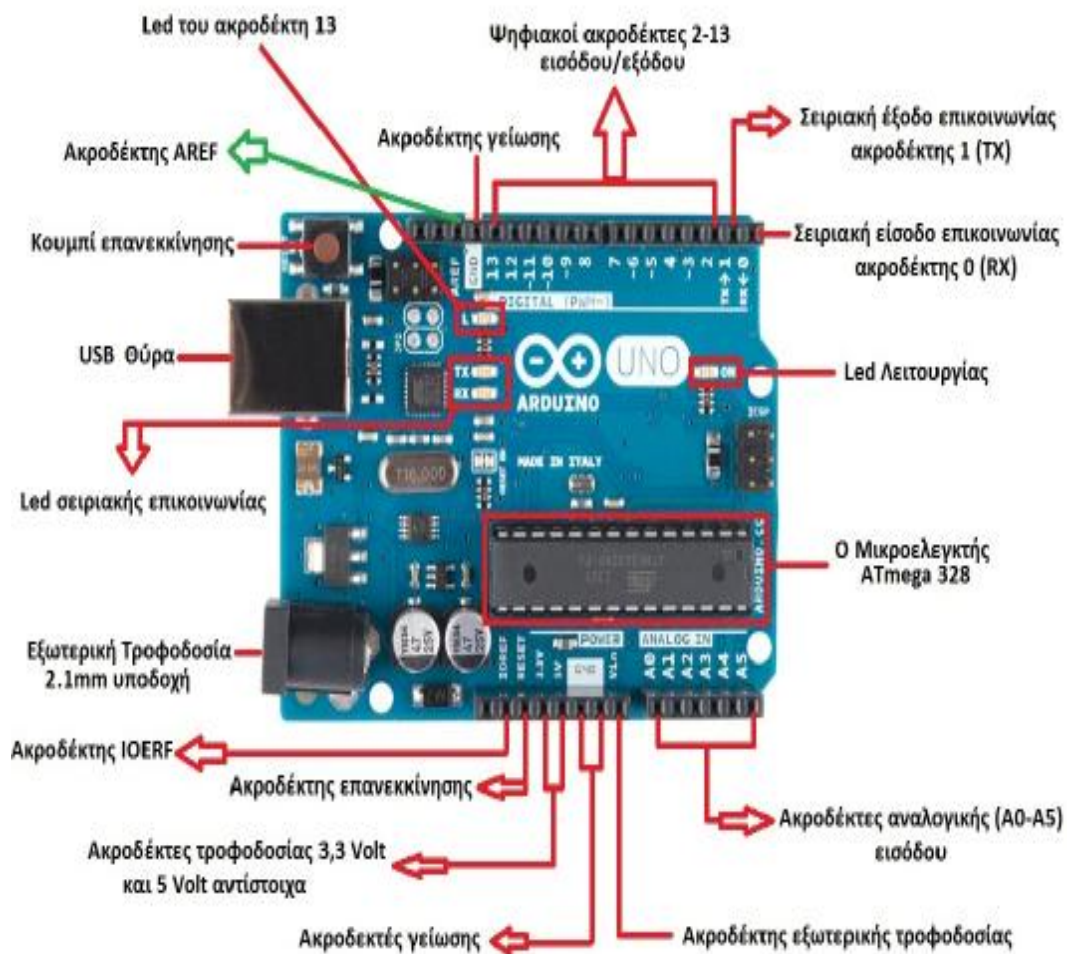
**Εικόνα 27 Arduino Ethernet Shield κάτω όψη**

## ΚΕΦΑΛΑΙΟ 3: ARDUINO UNO

### 3.1 Τι είναι το Arduino Uno

Το Arduino Uno είναι μια πλακέτα μικροελεγκτή βασισμένη στο chip Atmega328p. Έχει 14 ψηφιακές εισόδους ( 6 από αυτές μπορούν να χρησιμοποιηθούν ως έξοδοι PWM), 6 αναλογικές εισόδους, ένα 16 MHz χαλαζία, μια σύνδεση USB, μια υποδοχή ρεύματος, μια κεφαλίδα ICSP και ένα κουμπί επαναφοράς. "Uno" σημαίνει ένα στα ιταλικά και επιλέχτηκε για να σηματοδοτήσει την έναρξη του Arduino λογισμικού (IDE) 1.0. Πλέον υπάρχουν νεότερες εκδόσεις του λογισμικού.

Εικόνα 3: Arduino Uno 15



## 3.2 Ανάλυση πλακέτας Arduino Uno

Η πλακέτα arduino uno διαθέτει :

- 14 ψηφιακές I/O θύρες (εισόδου & εξόδου). Κυρίως με το πρόγραμμα που θα φορτωθεί στον μικροελεγκτή αυτές οι θύρες μπορούν να εργαστούν σαν είσοδοι ή έξοδοι I/O ψηφιακών σημάτων.
- Οι ψηφιακές θύρες 3, 5, 6, 9, 10, 11 μπορούν να λειτουργήσουν και ως ψευδοαναλογικές θύρες εξόδου με το σύστημα PWM (Pulse Width Modulation), δηλαδή το ίδιο σύστημα που διαθέτουν οι μητρικές των υπολογιστών για να ελέγχουν τις ταχύτητες των ανεμιστήρων και γενικότερα παλμικά σήματα. Το PWM παίρνει ένα εύρος τιμών από το 0 έως το 255. Δεν είναι πραγματικά αναλογικό σύστημα, έτσι θέτοντας στην έξοδο την τιμή 127, δεν σημαίνει ότι η έξοδος θα παρέχει 2.5V αντί της κανονικής τιμής των 5V, αλλά ότι θα δίνει έναν παλμό που η τάση του θα εναλλάσσεται με μεγάλη συχνότητα και για ίσα χρονικά διαστήματα μεταξύ των τιμών 0V και 5V με σκοπό η μέση τιμή να ισούται με 2,5V.
- Οι θύρες 0 και 1 χρησιμοποιούνται επίσης και για να λαμβάνουν (RX) και να μεταδίδουν (TX) TTL σειριακά δεδομένα. Έτσι, όταν για παράδειγμα το πρόγραμμα στέλνει δεδομένα σειριακά, τότε αυτά προωθούνται στην θύρα USB μέσω του ελεγκτή Serial-Over-USB (FTDI) όπως επίσης και στο pin 0 για να τα διαβάσει ενδεχομένως μία άλλη συσκευή (π.χ. ένα δεύτερο arduino στη δικιά του θύρα 1, η επικοινωνία είναι πολύ σημαντική). Αυτό φυσικά σημαίνει ότι αν στο πρόγραμμα ενεργοποιηθεί το σειριακό interface, καταλαμβάνονται δύο ψηφιακές θύρες εισόδου/εξόδου.
- Οι θύρες 2 και 3 λειτουργούν και ως εξωτερικά interrupt (interrupt 0 και 1 αντίστοιχα), δηλαδή σαν διακόπτες αν συμβεί κάτι που έχουμε θέσει εμείς. Με άλλα λόγια, μπορούν να ρυθμιστούν μέσα από το πρόγραμμα ώστε να λειτουργούν αποκλειστικά ως ψηφιακές είσοδοι στις οποίες όταν συμβαίνουν συγκεκριμένες αλλαγές τάσης, η κανονική ροή του προγράμματος να σταματάει άμεσα και να εκτελείται μία συγκεκριμένη συνάρτηση, ή αλλαγή στην ροή του ρεύματος. Τα εξωτερικά interrupt είναι ιδιαίτερα χρήσιμα σε εφαρμογές που απαιτούν συγχρονισμό μεγάλης ακρίβειας.
- 6 αναλογικές θύρες εισόδου αριθμημένες από το 0 έως το 5 μια προς μία. Το καθένα από αυτά λειτουργεί ως αναλογική είσοδος κάνοντας χρήση του ADC (Analog to Digital Converter). Για παράδειγμα, αν τροφοδοτηθεί ένα από αυτά τα pin με μία τάση η οποία μπορεί να κυμανθεί με ένα ποτενσιόμετρο ακριβείας από 0V ως μία τάση αναφοράς Vref (η οποία αν δεν γίνει κάποια αλλαγή είναι προρυθμισμένη στα 5V ή την τάση που θέλουμε να έχουμε εμείς), τότε μέσα από το πρόγραμμα μπορεί να «διαβαστεί» η τιμή της θύρας ως ένας ακέραιος αριθμός χωρητικότητας 10-bit, από 16 το 0 (όταν η τάση στο pin είναι 0V) μέχρι το 1023 (όταν η τάση στο pin είναι 5V). Η τάση αναφοράς Vref μπορεί να ρυθμιστεί με μία εντολή όπως για παράδειγμα στα 1.1V. Ένας άλλος τρόπος όπου η τάση αναφοράς μπορεί να δηλωθεί από τον προγραμματιστή είναι τροφοδοτώντας με μία εξωτερική τάση αναφοράς τη θύρα με την σήμανση AREF που βρίσκεται στην απέναντι πλευρά της πλακέτας. Έτσι, αν τροφοδοτηθεί η θύρα AREF με 3.3V και στην συνέχεια εκτελεσθεί η εντολή να διαβαστεί κάποιο pin αναλογικής εισόδου στο οποίο εφαρμόζετε τάση 1.65V, το Arduino θα επιστρέψει την τιμή 512.



- Δίπλα από τις θύρες αναλογικής εισόδου, υπάρχει μία ακόμα συστοιχία από 6 pin με την σήμανση POWER. Η λειτουργία του καθενός pin έχει ως εξής:
- Το πρώτο, με την ένδειξη RESET, όταν γειωθεί (με οποιοδήποτε από τα 3 pin με την ένδειξη GND που υπάρχουν στο arduino) έχει ως αποτέλεσμα την επανεκκίνηση του arduino , υπάρχει βέβαια και κουμπί που κάνει αυτή την λειτουργία.
- Το δεύτερο με την ένδειξη 3.3V, μπορεί να τροφοδοτήσει διατάξεις, συσκευές ή αισθητήρες με τάση 3.3V και είναι σημαντική τάση. Η τάση αυτή δεν προέρχεται από την εξωτερική τροφοδοσία αλλά παράγεται από τον ελεγκτή Serial-over-USB και έτσι η μέγιστη ένταση που μπορεί να παρέχει είναι μόλις 50mA για αυτό θέλει προσοχή αν αυτό που θέλουμε να τροφοδοτήσουμε θέλει περισσότερο πρέπει να γίνει με ξεχωριστή τάση.
- Η τρίτη θύρα με την ένδειξη 5V, μπορεί να χρησιμοποιηθεί και αυτή για την τροφοδότηση διαφόρων εξαρτημάτων, συσκευών ή αισθητήρων με τάση 5V(π.χ. άλλο Arduino). Ανάλογα με τον τρόπο τροφοδοσίας του ίδιου του Arduino, η τάση αυτή προέρχεται είτε άμεσα από την θύρα USB (που ούτως ή άλλως παρέχει τάση 5V), είτε από την εξωτερική τροφοδοσία αφού αυτή περάσει από ένα ρυθμιστή τάσης για να την «σταθεροποιήσει» στα 5V, η εξωτερική τροφοδοσία κυμένεται από 7v μέχρι 12v.
- Το τέταρτο και το πέμπτο pin με την ένδειξη GND είναι οι γειώσεις.
- Το έκτο και τελευταίο pin, με την ένδειξη Vin(τάση εισόδου) έχει διπλό ρόλο. Σε συνδυασμό με το pin γείωσης δίπλα του, μπορεί να λειτουργήσει ως μέθοδος εξωτερικής τροφοδοσίας του Arduino στην περίπτωση που δεν βολεύει να χρησιμοποιηθεί η υποδοχή του φισ των 2.1mm(JACK). Αν όμως υπάρχει ήδη συνδεδεμένη εξωτερική τροφοδοσία μέσω του φισ(jack), τότε μπορεί να χρησιμοποιηθεί αυτό το pin για να τροφοδοτήσει εξαρτήματα και συσκευές με την πλήρη τάση της εξωτερικής τροφοδοσίας (7~12V), πριν αυτή περάσει από τον ρυθμιστή τάσης όπως γίνεται με το pin των 5V.
- Η φόρτωση του sketch πραγματοποιείται μέσω μίας USB θύρας που διαθέτει η πλακέτα arduinouno. Έτσι οι πληροφορίες που προέρχονται από την USB θύρα του υπολογιστή εισέρχονται στην USB θύρα του arduinouno και στην συνέχεια οδηγούνται στο FDTI ολοκληρωμένο για να διαμορφωθούν σε μία κατάλληλη μορφή ώστε ο μικροελεγκτής να μπορέσει να τις διαβάσει επιτυχώς.
- Πάνω στην πλακέτα του arduinouno υπάρχει ένας μικροδιακόπτης (micro-switch) και 4 smd(επιφανειακής στήριξης) LED. Η λειτουργία του διακόπτη (που έχει την σήμανση RESET) και του ενός LED με την σήμανση POWER είναι προφανής στην πλακέτα. Τα δύο LED με τις σημάνσεις TX και RX, χρησιμοποιούνται ως ένδειξη λειτουργίας του σειριακού interface, καθώς ανάβουν όταν το arduino στέλνει ή λαμ-

βάνει (αντίστοιχα) δεδομένα μέσω της USB θύρας. Τα LED αυτά ελέγχονται από τον ελεγκτή Serial-over-USB και συνεπώς δεν λειτουργούν όταν η σειριακή επικοινωνία γίνεται αποκλειστικά μέσω των ψηφιακών pin 0 και 1.

- Επίσης, υπάρχει το LED με τη σήμανση L. Η βασική λειτουργία του LED στην πλακέτα Arduino είναι για να αναβοσβήνει συνήθως για δοκιμαστικό σκοπό. Οι κατασκευαστές σκέφτηκαν να ενσωματώσουν ένα LED στην πλακέτα το οποίο το σύνδεσαν στη ψηφιακή θύρα 13 μέσω μιας αντίστασης. Έτσι ακόμα και αν δεν έχει συνδεθεί τίποτα πάνω στο φυσικό pin 13, αναθέτοντας του την τιμή HIGH μέσα από το πρόγραμμα, θα ανάψει το ενσωματωμένο LEDL επιτυχώς.

- Προστασία υπερέντασης USB

Το Arduino Uno έχει μία επανεκκινήσιμη αντίσταση η οποία προστατεύει τις USB θύρες του υπολογιστή μας από υπερένταση. Παρόλο που κάθε υπολογιστής παρέχει την δική του εσωτερική προστασία, η αντίσταση παρέχει μία επιπλέον προστασία. Αν περισσότερα από 500mA εφαρμοστούν στην USB πόρτα, η αντίσταση αυτόματα θα σπάσει την σύνδεση μέχρι να απομακρυνθεί η υπερφόρτωση.

- Φυσικά Χαρακτηριστικά

Το μέγιστο μήκος και πλάτος της πλακέτας του Uno είναι 6.858 και 5.334 εκατοστά αντίστοιχα, με την USB πόρτα και την υποδοχή ρεύματος να εκτείνονται πέρα των αρχικών διαστάσεων. Τέσσερις τρύπες για βίδες επιτρέπουν στην πλακέτα να μπορεί να τοποθετηθεί σε μία ξεχωριστή επιφάνεια ή ένα κουτί.

- Αυτόματη Επανεκκίνηση Λογισμικού

Αντί να απαιτείτε το πάτημα του πλήκτρου της επανεκκίνησης πριν από ένα ανέβασμα προγράμματος, το Arduino Uno είναι σχεδιασμένο με τέτοιο τρόπο που επιτρέπει την επανεκκίνηση από το λογισμικό που τρέχει στον συνδεδεμένο υπολογιστή. Μία από τις γραμμές ελέγχου ροής υλικού (DTR) του ATmega8U2/16U2 συνδέεται με τη γραμμή επαναφοράς των ATmega328 μέσω ενός πυκνωτή 100 nanofarad. Όταν αυτή η γραμμή είναι χαμηλή, η γραμμή επανεκκίνησης πέφτει τόσο όσο το chip να κάνει επανεκκίνησης. Το λογισμικό μας επιτρέπει να ανεβάσουμε κώδικα απλώς πατώντας ένα κουμπί στο περιβάλλον του λογισμικού. Αυτό σημαίνει ότι ο bootloader μπορεί να έχει μικρότερο χρονικό όριο, όπως η μείωση του DTR μπορεί να είναι καλά συντονισμένη με την έναρξη του ανεβάσματος του κώδικα. Αυτή η επιλογή έχει και άλλες επιπτώσεις. Όταν η πλακέτα μας είναι ενωμένη με έναν υπολογιστή που τρέχει Mac OS X ή Linux, κάνει επανεκκίνηση κάθε φορά που γίνεται μια σύνδεση σε αυτό από το λογισμικό (μέσω USB). Για το επόμενο μισό δευτερόλεπτο ο bootloader τρέχει στο Uno. Αν και είναι προγραμματισμένο να αγνοεί τα άσχετα δεδομένα (οτιδήποτε εκτός του κώδικα που ανεβάζουμε) θα παρακολουθήσει τα πρώτα bytes δεδομένων που στέλνονται στην πλακέτα αφότου έχει γίνει σύνδεση. Αν ένα πρόγραμμα τρέχει στην πλακέτα λαμβάνει μια απλή διαμόρφωση ή δέχεται νέα δεδομένα όταν αρχικά ανοίγει, πρέπει να σιγουρευτούμε ότι το λογισμικό με το οποίο επικοινωνεί

περιμένει ένα δευτερόλεπτο για να ανοίξει η επικοινωνία προτού στείλει αυτά τα δεδομένα.

### 3.3. Τεχνικά Χαρακτηρηστικά

Microcontroller	ATmega328P
Operating Voltage	5V
Input Voltage (recommended)	7-12V
Input Voltage (limit)	6-20V
Digital I/O Pins	14 (of which 6 provide PWM output)
PWM Digital I/O Pins	6
Analog Input Pins	6
DC Current per I/O Pin	20 mA
DC Current for 3.3V Pin	50 mA
Flash Memory	32 KB (ATmega328P) of which 0.5 KB used by bootloader
SRAM	2 KB (ATmega328P)
EEPROM	1 KB (ATmega328P)
Clock Speed	16 MHz
LED_BUILTIN	13
Length	68.6 mm
Width	53.4 mm
Weight	25 g

Στην παραπάνω εικόνα παρατίθενται τα τεχνικά χαρακτηριστικά του Arduino Uno.

## 3.4 Τεχνικά χαρακτηριστικά Arduino Uno

### 3.4.1 FlashMemory

Η μνήμη flash έχει χωρητικότητα 32Kb, από τα οποία τα 2Kb χρησιμοποιούνται από το firmware του arduino που έχει εγκαταστήσει ήδη ο κατασκευαστής του στο χώρο μνήμης. Το firmware αυτό που στην ορολογία του arduino ονομάζεται bootloader είναι αναγκαίο για την εγκατάσταση των προγραμμάτων στον μικροελεγκτή μέσω της θύρας USB από το FTDI chip, χωρίς δηλαδή να χρειάζεται εξωτερικός hardware programmer. Τα υπόλοιπα 30Kb της μνήμης Flash χρησιμοποιούνται για την αποθήκευση αυτών ακριβώς των προγραμμάτων που θα επιχειρήσετε να αναπτύξετε από μόνοι σας αν ασχοληθείτε, αφού πρώτα μεταγλωττιστούν στον υπολογιστή. Η μνήμη Flash δεν χάνει τα περιεχόμενά της με την απώλεια της τροφοδοσίας ή κάνοντας reset το μικροελεγκτή αλλιώς θα είχαμε πρόβλημα στο να ξαναπρογραμματίζουμε ξανά. Επίσης, ενώ η μνήμη Flash υπό κανονικές συνθήκες δεν προορίζεται για χρήση runtime, μέσα από τα προγράμματα λόγω της μικρής συνολικής μνήμης που είναι διαθέσιμη σε αυτά (2Kb SRAM + 1Kb EEPROM), έχει σχεδιαστεί μία βιβλιοθήκη από τις πολλές, που επιτρέπει την χρήση runtime στον χώρο που περισσεύει από την αποθήκευση των sketch (30Kb πλην το μέγεθος του προγράμματος σε μεταγλωτισμένη μορφή).

### 3.4.2 SRAMMemory

Η μνήμη SRAM (static random access memory) είναι η ωφέλιμη μνήμη που μπορούν να χρησιμοποιήσουν τα προγράμματα για να αποθηκεύουν μεταβλητές, πίνακες, ορίσματα, τιμές κ.λπ. κατά το runtime. Όπως και σε έναν υπολογιστή, αυτή η μνήμη χάνει τα δεδομένα της όταν η παροχή ρεύματος στο arduino σταματήσει ή αν γίνει reset, για αυτό λέγεται και στατική. Στο ATmega328 η SRAM μνήμη καταλαμβάνει χώρο 2048 bytes κατά την διάρκεια μίας κανονικής λειτουργίας και όλες οι μεταβλητές φορτώνονται σε αυτή καθ' όλη την διάρκεια της λειτουργίας του μικροελεγκτή.

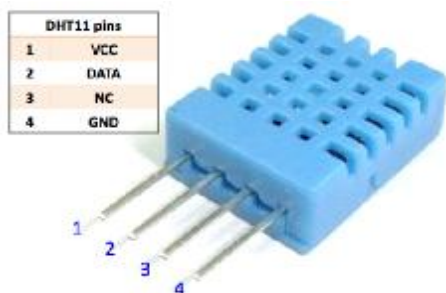
### 3.4.3 EEPROMMemory

Το τελευταίο μέρος της μνήμης είναι η EEPROM και καταλαμβάνει 1024 bytes, αρκετά μικρή για μνήμη που χρησιμοποιείται μόνο για ανάγνωση (read-only). Η 20 EEPROM έχει όριο ζωής καθώς δε μπορεί να επαναπρογραμματιστεί για περισσότερες από 100.000 φορές. Είναι μία byte addressable μνήμη, γεγονός που καθιστά λίγο δυσκολότερο να τεθεί σε χρήση αφού απαιτείται ειδική βιβλιοθήκη ώστε να μπορέσει κάποιος να έχει πρόσβαση σε αυτή.



## ΚΕΦΑΛΑΙΟ 4 : SENSORS (ΑΙΣΘΗΤΗΡΕΣ)

### 4.1 DHT-11 αισθητήρας θερμοκρασία και υγρασία



Το DHT-11 είναι ένα αισθητήριο που χρησιμοποιείτε για την εύρεση της σχετικής υγρασίας και θερμοκρασίας στον χώρο.

Η υγρασία και θερμοκρασία του χώρου θα εκτυπώνονται στην σειριακή οθόνη.

#### 4.1.2 Σχετικά με το DHT-11

Το DHT-11 είναι ένας βασικός, χαμηλού κόστους, αισθητήρας για την εύρεση υγρασίας και θερμοκρασίας στον χώρο. Στο εσωτερικό του κρύβει έναν αισθητήρα υγρασίας και ένα θερμίστορ (μεταβλητή αντίσταση που η τιμή της αλλάζει σε σχέση με την θερμοκρασία) 'διαβάζοντας' έτσι τον αέρα που το περιβάλλει.

Οι συνδέσεις είναι απλές, το πρώτο pin από αριστερά στα 3-5V, το δεύτερο pin (data) σε μια ψηφιακή είσοδο και το τέρμα δεξιά στην γείωση.

**Σημείωση:** Μπορείτε να χρησιμοποιήσετε και τον αισθητήρα DHT22, ίδιος με τον DHT11 με την διαφορά ότι έχει καλύτερες ακρίβειες στην μέτρηση.

#### Τεχνικές πληροφορίες:

- Πηγή : 3-5V
- Μέγιστο ρεύμα: 2.5mA
- Υγρασία: 20-80%, ακρίβεια 2-5%
- Θερμοκρασία: 0 to 50°C, ακρίβεια  $\pm 0.5^{\circ}\text{C}$

## 4.2 Αισθητήρας θερμοκρασίας DS18B20

Το DS18B20 επικοινωνεί μέσω ενός διαύλου 1-Wire που εξ ορισμού απαιτεί μόνο μία γραμμή δεδομένων (και γείωση) για επικοινωνία με το Arduino. Έχει ένα εύρος θερμοκρασίας λειτουργίας από  $-55^{\circ}\text{C}$  έως  $+125^{\circ}\text{C}$  και είναι ακριβής  $\pm 0,5^{\circ}\text{C}$  στην περιοχή  $-10^{\circ}\text{C}$  έως  $+85^{\circ}\text{C}$ . Επιπλέον, το DS18B20 μπορεί να παράγει ενέργεια απευθείας από τη γραμμή δεδομένων ("παράσιτο"), εξαλείφοντας την ανάγκη για εξωτερική τροφοδοσία.

Κάθε DS18B20 διαθέτει έναν μοναδικό σειριακό κώδικα 64 bit, ο οποίος επιτρέπει πολλαπλές συσκευές DS18B20 να λειτουργούν στον ίδιο δίαυλο 1-Wire. Έτσι, είναι απλό να χρησιμοποιηθεί ένας μικροεπεξεργαστής για τον έλεγχο πολλών DS18B20 που διανέμονται σε μια μεγάλη περιοχή. Οι εφαρμογές που μπορούν να επωφεληθούν από αυτό το χαρακτηριστικό περιλαμβάνουν περιβαλλοντικούς ελέγχους HVAC, συστήματα παρακολούθησης της θερμοκρασίας μέσα σε κτίρια, εξοπλισμό ή μηχανήματα και συστήματα παρακολούθησης και ελέγχου διαδικασιών.

### 4.2.1 Βασικά χαρακτηριστικά DS18B20

Η μοναδική διασύνδεση 1-καλωδίου απαιτεί μόνο έναν ακροδέκτη θύρας για επικοινωνία

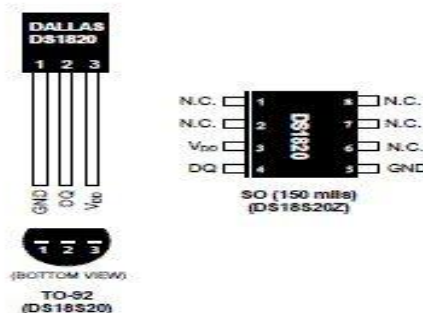
Κάθε συσκευή έχει έναν μοναδικό 64-bit σειριακό κώδικα αποθηκευμένο σε ενσωματωμένο ROM

Δεν απαιτούνται εξωτερικά στοιχεία

Μπορεί να τροφοδοτείται από γραμμή δεδομένων? Το εύρος τροφοδοσίας ρεύματος είναι 3.0V έως 5.5V

Μέτρα Θερμοκρασίες από  $-55^{\circ}\text{C}$  έως  $+125^{\circ}\text{C}$  ( $-67^{\circ}\text{F}$  έως  $+257^{\circ}\text{F}$ )

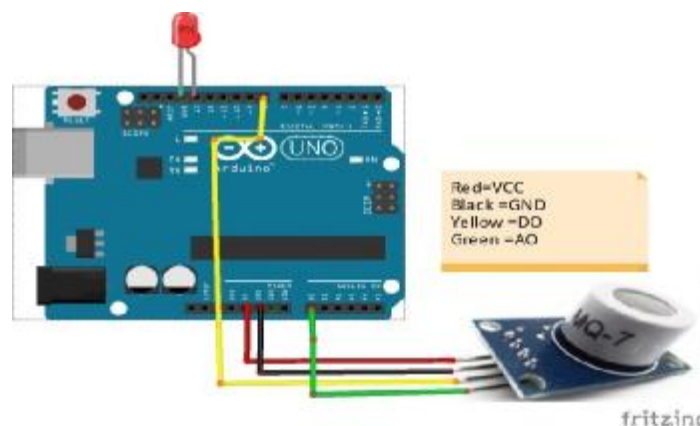
$\pm 0,5^{\circ}\text{C}$  Ακρίβεια από  $-10^{\circ}\text{C}$  έως  $+85^{\circ}$







#### 4.4 Αισθητήρας Αερίου MQ-7



Σημειώσεις: Λόγω της διαφορετικής παρτίδας παραγωγής, το χρώμα του προϊόντος θα είναι διαφορετικό, αλλά εγγυόμαστε ότι όλο το προϊόν είναι ίδιας ποιότητας. Αυτός ο πίνακας διαθέτει LED TX και RX που το καθιστούν λίγο καλύτερο για χρήση μέσω του καλωδίου FTDI. Μπορείτε να δείτε πραγματική κυκλοφορία στις λυχνίες LED για να επαληθεύσετε εάν η κάρτα λειτουργεί.

Χαρακτηριστικό γνώρισμα:

Μέγεθος μονάδας:  $30 \times 19 \times 15\text{mm}$  /  $1,18 * 0,74 * 0,59$  "

Τάση εισόδου: DC5V

Απώλεια ισχύος: 150mA

DO Έξοδος: 0,1-5V

AO Έξοδος: 0,1-0,3V

Ανιχνεύσιμη συγκέντρωση: 10-1000ppmCO

Χαρακτηριστικά μονάδας:

Υψηλής ποιότητας διπλός πίνακας σχεδίασης

Ισχύς και ένδειξη εξόδου σήματος TTL

Έχει καλύτερη ευαισθησία ανίχνευσης για το μονοξείδιο του άνθρακα

Έχει μεγάλη διάρκεια ζωής και αξιόπιστη σταθερότητα

Κατάλληλο για την ανίχνευση μονοξειδίου του άνθρακα, αερίου κ.λπ.

## 4.5 Ηλεκτρονικά εξαρτήματα

### 4.5.1 Ποτενσιόμετρο – Potentiometer

Ένα ποτενσιόμετρο μπορεί να περιγραφεί ως μια διάταξη με κύκλωμα μεταβλητής αντίστασης. Γυρίζοντας το χειριστήριο που έχει, αυξάνει ή μειώνει την αντίστασή του, οπότε περνάει λιγότερο ή περισσότερο ρεύμα από αυτό

Έχει τρία σημεία σύνδεσης – ένα πόδι συνδέεται στην πηγή (5V), ένα στη γείωση (GND) και το μεσαίο στο κύκλωμα που θέλουμε να πάρει τη μεταβλητή τιμή ρεύματος (αναλογικό pin στο Arduino).

Το ποτενσιόμετρο δηλαδή δουλεύει ως ένας διαιρέτης τάσης, ο οποίος όμως μεταβάλλεται κάθε φορά που γυρίζουμε το χειριστήριό του. Ποτενσιόμετρα χρησιμοποιούνται σε πολλές εφαρμογές, όπως στην αυξομείωση του ήχου στα στερεοφωνικά ή την αυξομείωση της έντασης μιας λάμπας.

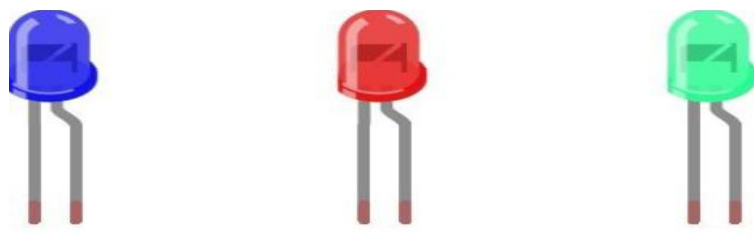


### 4.5.2 Φωτοдиодοι – Leds

Οι φωτοдиодοι, ή τα leds όπως έχουν κυριαρχήσει, υπάρχουν σε διάφορα χρώματα και τάσεις λειτουργίας. Ανάλογα με την προβλεπόμενη τάση λειτουργίας του led που έχουμε πρέπει να χρησιμοποιήσουμε μαζί και την κατάλληλη αντίσταση, ώστε να αποφύγουμε κάποια καταστροφή από υπέρταση. Για να βρούμε την αντίσταση που χρειαζόμαστε αρκεί να θυμηθούμε τον τύπο  $R = (V_{\pi} - V_{\lambda}) / I$  όπου R είναι η αντίσταση που χρειαζόμαστε,  $V_{\pi}$  η παρεχόμενη τάση από την πηγή μας (5V από το Arduino),  $V_{\lambda}$  η τάση λειτουργίας του led και I το ρεύμα λειτουργίας του led. Τυπικά, μια αντίσταση γύρω στα 220 Ω καλύπτει τα περισσότερα led που έχουμε χρησιμοποιήσει.

Επίσης, τα leds έχουν πολικότητα, δηλαδή δουλεύουν μόνο αν συνδεθούν στην κατάλληλη φορά ρεύματος. Συνήθως το πόδι που πρέπει να συνδεθεί στη θετική κατεύθυνση (+) είναι πιο μακρύ από το αντίστοιχο για την αρνητική φορά (-). Επίσης, το λαμπάκι από την αρνητική μεριά (-) είναι συνήθως επίπεδο κι όχι στρογγυλό όπως

είναι από το άλλο πόδι (+). Μην ανησυχείτε όμως, μια ανάποδη σύνδεση δεν θα το καταστρέψει, απλά δεν θα ανάψει στην ανάποδη φορά (εκτός από την ιδιαίτερη περίπτωση να έχετε δώσει πολύ μεγάλη τάση).



#### 4.6 Δυνατότητες του ARDUINO

Αν και μικροσκοπικό ( 7x5 cm ) οι δυνατότητες που προσφέρει είναι πάρα πολλές .Μπορούμε να το χρησιμοποιήσουμε σε εφαρμογές ρομποτικής , περιβαλλοντικές μετρήσεις (π.χ. μετεωρολογία ) , σε αυτοματισμούς.καταφέροντας έτσι πάρα πολλά: την κίνηση servo, stepper και Dcκινητήρων , τη λήψη πληροφοριών απο διάφορους αισθητήρες (θερμοκρασίας, υγρασίας , υπέρυθρων κ.α.) , τν αμφίδρομη σειριακή επικοινωνία μεταξύ Arduino και PC χρησιμοποιώντας γλώσσες προγραμματισμού ( όπως Java και Python), όπως επίσης την αναπαραγωγή και αντίληψη ήχων .Η πλακέτα Arduino μέχρι αυτή τη στιγμή διατίθεται σε 16 βασικές παραλλαγές οι οποίες αναφέρονται σε διαφορετικές χρήσεις η κάθε μια , ανάλογα με τις ανάγκες της εφαρμογής μας.

Παρακάτω μια λίστα με διάφορων ειδών εξαρτήματα τα οποία θα μπορούμε να συνδέσουμε στο Arduino ανάλογα με την περίπτωση και τον σκοπό που επιδιώκεται να επιτύχουμε :

- [Αερίου](#)
- [Θερμοκρασίας](#)
- [Ήχου](#)
- [Βάρους](#)
- [Υγρασίας](#)
- [Ραδιενέργειας](#)
- [Υπερύθρων](#)
- [GPS](#)
- [Κάμερες](#)
- [Keypad & Joystick](#)
- [RFID](#)
- [RTC](#)
- [Τάσης/Ρεύματος](#)
- [Πίεσης](#)

- [Αφής](#)
- [Φως/Χρώμα](#)
- [Ultrasonic](#)
- [Κίνησης](#)
- [Γυροσκόπια/Αξελερόμετρα/Μαγνητόμετρα](#)
- [Heart](#)
- [Μαγνητικοί](#)
- [Conductive](#)
- [Εγγύτητας](#)
- [Πρόσκρουσης](#)
- [3D Gesture](#)
- [Anemometer](#)
- [Hydro Generator](#)
- [Ehealth](#)
- [Lidar](#)
- [Isolator](#)



Logo της εταιρίας ARDUINO

## ΚΕΦΑΛΑΙΟ 5 : Πειραματική διαδικασία

### 5.1 Σκοπός

Σκοπός της συγκεκριμένης εφαρμογής είναι ο συνδυασμός ενός αναπτυξιακού συστήματος μαζί με αισθητήρες για την δημιουργία μιας ‘έξυπνης’ εφαρμογής.

Υλοποιήθηκε ένα μοντέλο, μικρού μεγέθους, μετεωρολογικού σταθμού με συγκεκριμένους αισθητήρες που προαναφέραμε στα παραπάνω κεφάλαια και αναπτυξιακό Arduino.

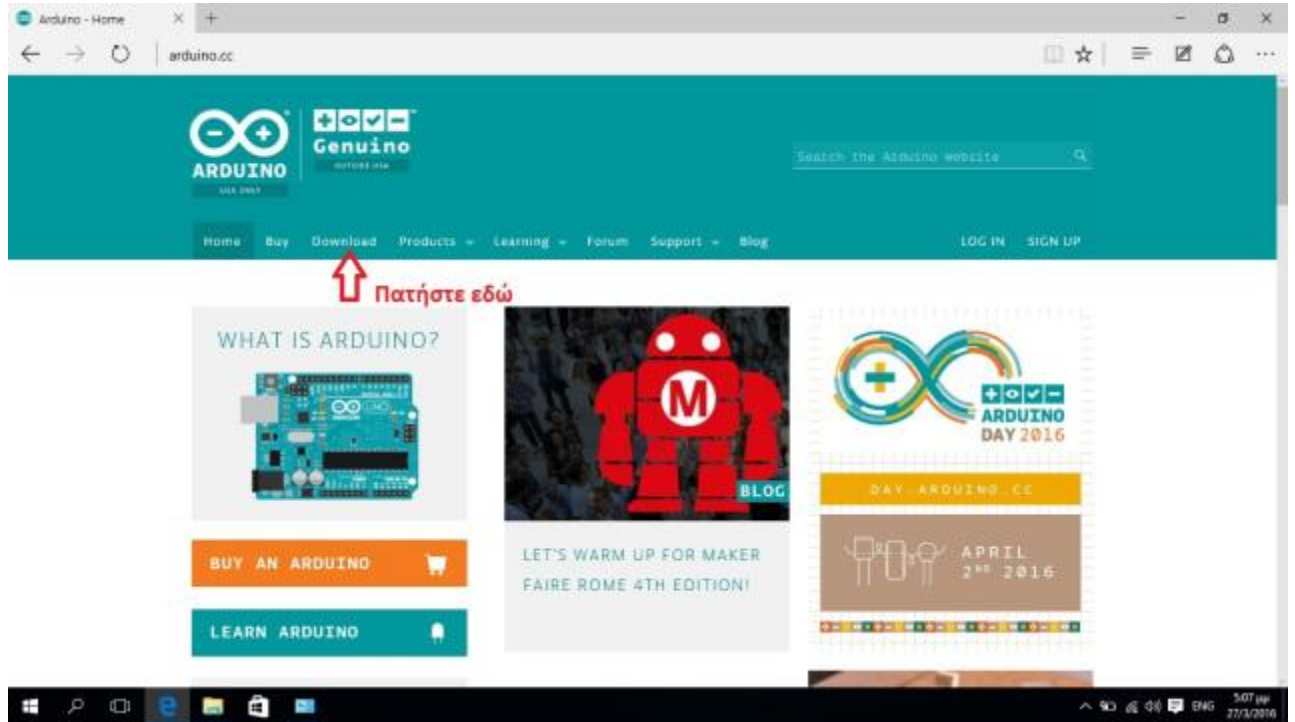
Απώτερος στόχος της όλης εφαρμογής είναι ο προγραμματισμός του Arduino, η συνδεσμολογία αισθητήρων και η μέτρηση-καταγραφή real-time τιμών που αφορούν μετεωρολογικά φαινόμενα καθώς και η γραφική παρουσίασή τους.

### 5.2 Τα υλικά της κατασκευής

- 1) Μία πλακέτα arduino uno.
- 2) Καλώδιο usb.
- 3) Υπολογιστή με εγκατεστημένο το arduino ide.
- 4) οθόνη LCD (16x2).
- 5) ποτενσιόμετρο.
- 6) καλώδια.
- 7) σένσορας **DHT-11** θερμοκρασία ς-υγρασίας.
- 8) Πλακέτα τοποθέτησης εξαρτημάτων (breadboard) .
- 9) Καλώδιο τροφοδοσίας .

### 5.3 Εγκατάσταση του Arduino IDE στο PC

Για να προγραμματίσετε την μονάδα σας θα χρειαστεί να κατεβάσετε από το ίντερνετ, και να εγκαταστήσετε το προγραμματιστικό περιβάλλον του arduino (IDE). Το λογισμικό διατίθεται δωρεάν από την επίσημη σελίδα του arduino.

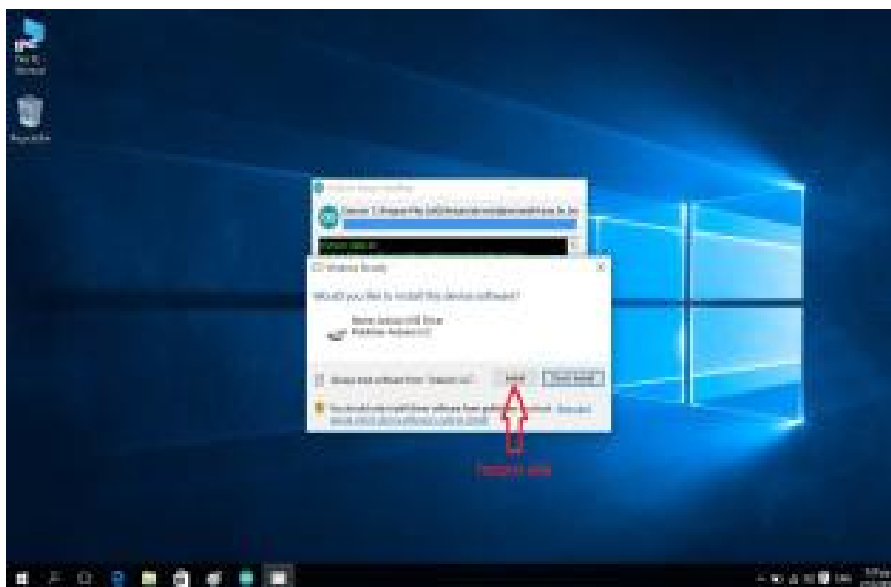
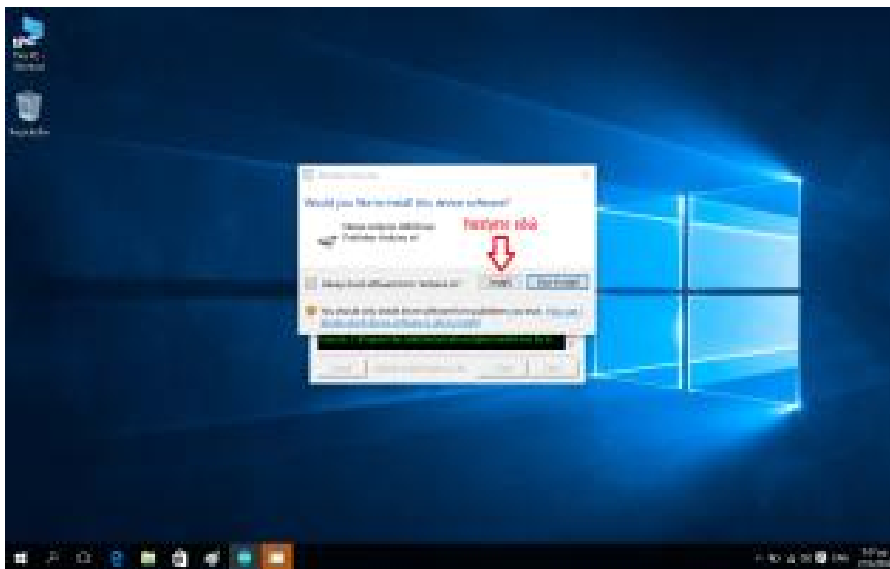


### Πατήστε στην επιλογή Download

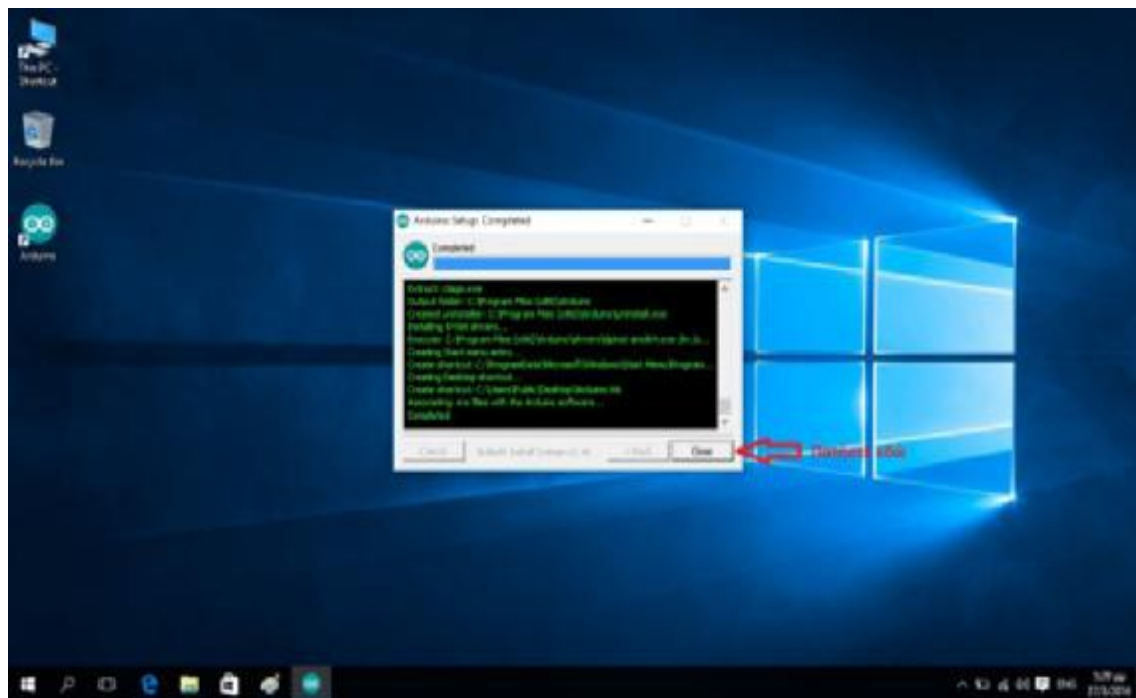


Στην σελίδα αυτή μπορείτε να επιλέξετε στο να κατεβάσετε το πρόγραμμα εγκατάστασης ή το συμπιεσμένης μορφής αρχείο(Zip). Αν δεν γνωρίζεται τι είναι τα zip αρχεία είναι καλύτερα να κατεβάσετε το πρόγραμμα εγκατάστασης το οποίο γράφει στον υπολογιστή όλα όσα χρειάζεστε για να χρησιμοποιήσετε το προγραμματιστικό περιβάλλον του arduino συμπεριλαμβανομένων και των οδηγών αναγνώρισης (drivers) του από τα windows.

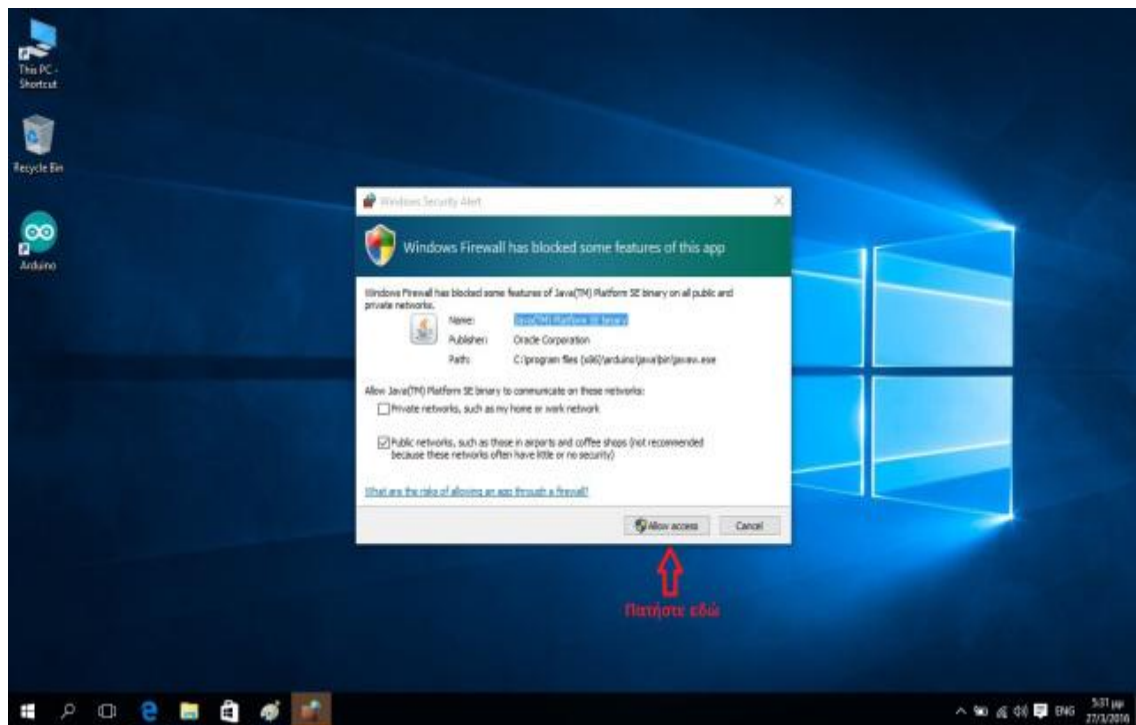
Όταν κατεβάσετε το πρόγραμμα εγκατάστασης διπλό-κλικάρετε το. Η διαδικασία εγκατάστασης είναι απλή, άπλα πατήστε στο «i agree» μετά στο επόμενο στάδιο το «next» και τέλος το «install». Τα windows θα βγάλουν ένα δυο εικονίδια, άπλα επιβεβαιώστε ότι θέλετε να εγκαταστήσετε το πρόγραμμα.



Όταν τελειώσει η εγκατάσταση πατήστε στο «close» ώστε να κλείσει το παράθυρο της εγκατάστασης.



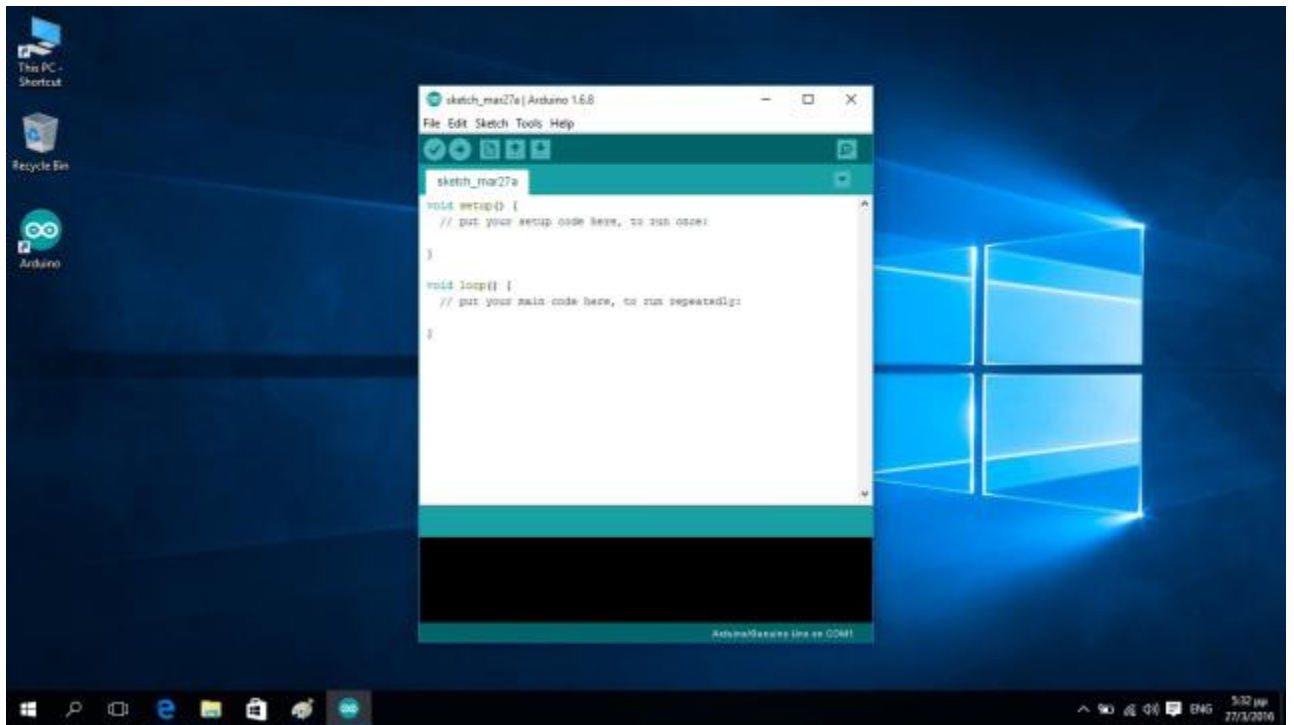
Στο desktop σας θα είναι τώρα πια το πρόγραμμα του arduino. Κλικάρετε πάνω του δυο φορές και θα ανοίξει. Με το που ανοίξει τα windows θα σας ρωτήσουν αν επιτρέπετε την χρήση. Πατήστε στο «allow access».



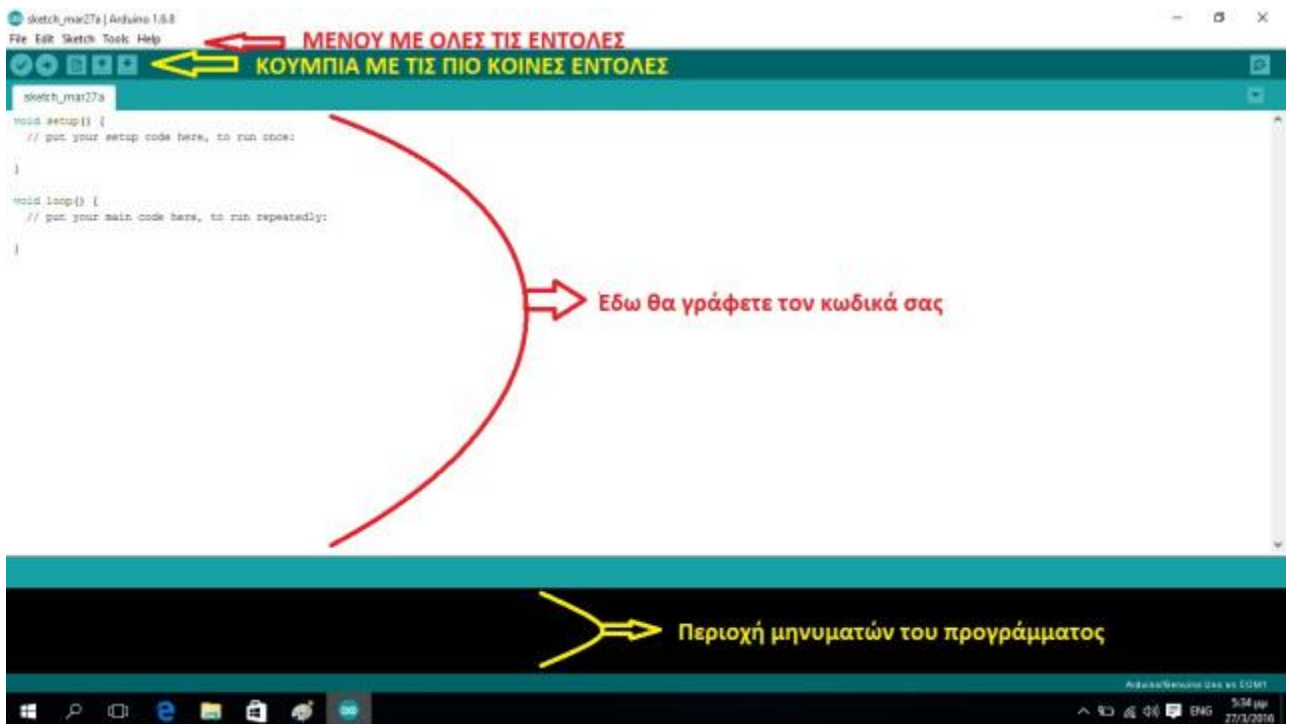
Είστε έτοιμοι.....



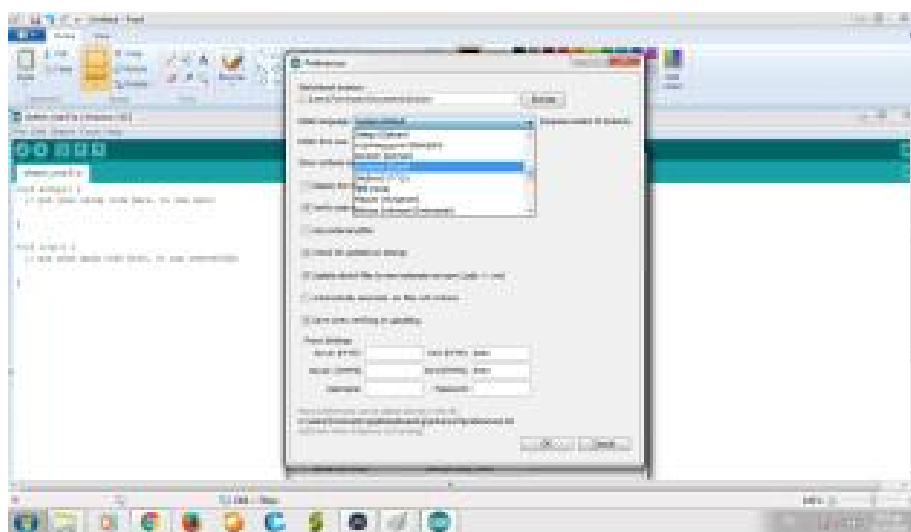
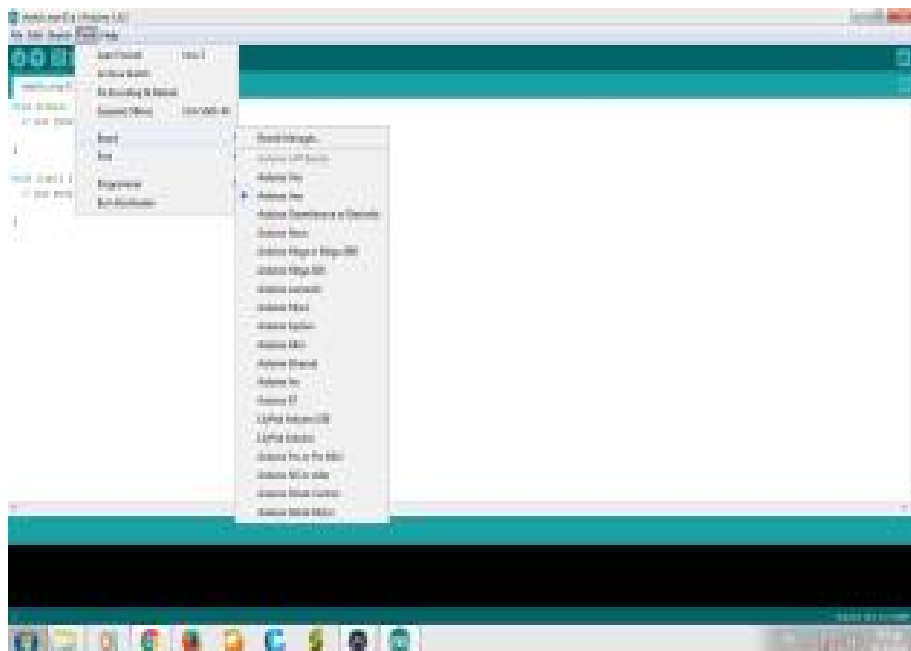
Θα εμφανιστεί μπροστα σας τώρα πια η κλασική και λιτή προγραμματιστική πλατφόρμα του arduino.



Ας δούμε λίγο πιο αναλυτικά το arduino IDE.



A) Πάνω πάνω έχουμε το μενού με όλες τις επιλογές. Ένα από τα πρώτα πράγματα που μπορείτε να κάνετε είναι να αλλάξετε την γλώσσά σε ελληνικά. Η διαδικασία είναι απλή πατήστε στο file και επιλέξτε την κατηγορία preferences έπειτα πατήστε στην επιλογή editor language και επιλέξτε τα ελληνικά. Πατήστε το Ok ,το μόνο που έχετε να κάνετε τώρα πια είναι να επαννεκινήσετε το πρόγραμμα.

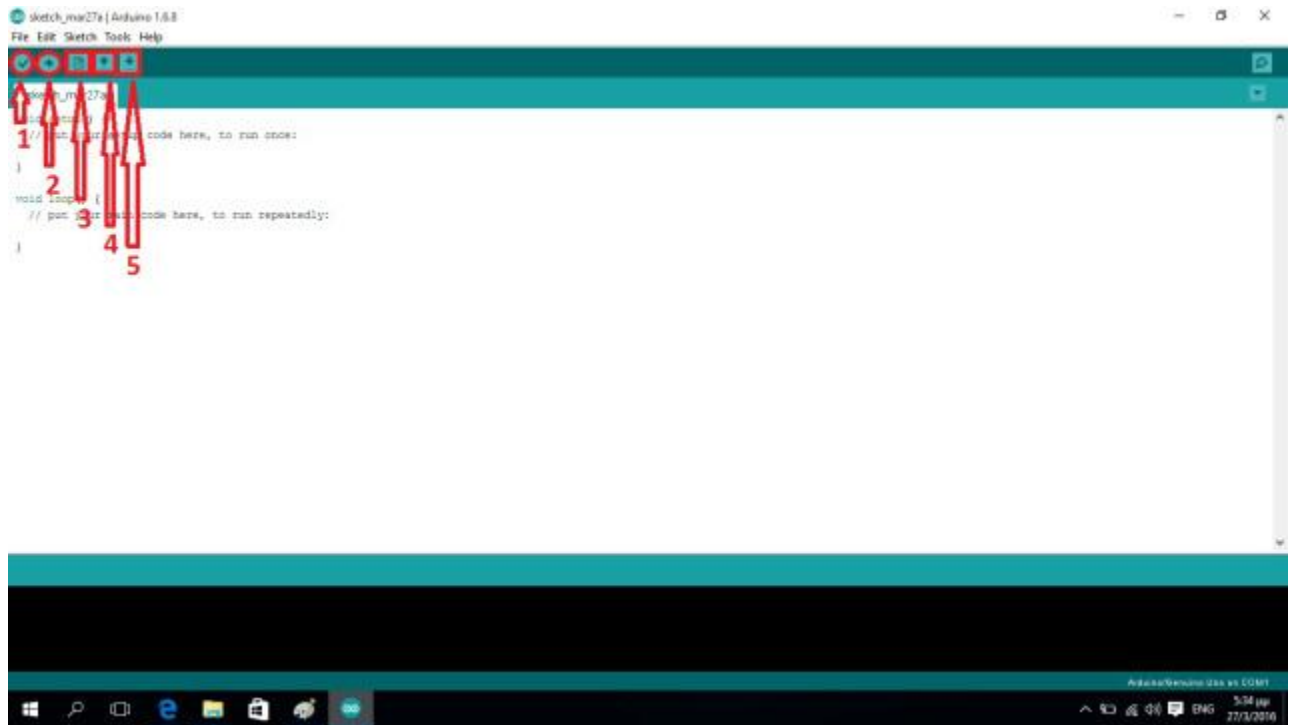


Επίσης στο μενού αυτό πρέπει να είναι επιλεγμένη σωστά η εκδοσή του arduino που έχετε καθώς και η θύρα που χρησιμοποιεί. Συνήθως τώρα πια αυτά γίνονται αυτόματα. Αν δεν έχει γίνει όμως πατήστε το tools και πηγαίνετε στην επιλογή board. Εκεί υπάρχει μια λίστα με όλες τις εκδόσεις του arduino, επιλέξτε την δικιά σας. Έπειτα πηγαίνετε στην επιλογή Port (θύρα) και επιλέξτε την θύρα που χρησιμοποιεί το arduino σας.

B) Κουμπιά με τις πιο κοινές επιλογές. Οι εντολές αυτές είναι οι εξής: Verify(επικύρωση), Upload(εξαγωγή),New(Νέο), Open(Άνοιξε),Save(Σώσε)

Γ) Τον text-editor. Εδώ γράφουμε τα προγράμματά (sketch) μας. Όλα τα προγράμματά μας πρέπει να εμπεριέχουν μέσα τους τις εντολές void setup() και το void loop() για να τα επικυρώσει το πρόγραμμα.

Δ) Περιοχή μηνυμάτων του προγράμματος. Σε αυτό το μέρος το πρόγραμμα μας ενημερώνει για το αποτέλεσμα από διάφορες ενέργειες μας. Για παράδειγμα μας πληροφορεί ότι το πρόγραμμα που γράψαμε είναι συντακτικά ορθό ή όχι όταν του πούμε να μας το επικυρώσει (verify) ή μας ενημερώνει για την επιτυχημένη εξαγωγή (upload) του προγράμματος μας στο arduino μας.



1. **Verify(επικύρωση):** Αυτή η εντολή κοιτάει γραμμή γραμμή το πρόγραμμα που έχουμε γράψει και μας ενημερώνει αν είναι γραμμένο συντακτικά σωστά. Αυτό δεν σημαίνει ότι το πρόγραμμα είναι και σωστό σε σχέση με το αποτέλεσμα που θέλουμε να πετύχουμε άπλα σημαίνει ότι δεν έχουμε ξεχάσει κάποιο σύμβολο ή ότι έχουμε ορίσει σωστά όλες τις μεταβλητές μας.
2. **Upload (εξαγωγή):** Αυτή η εντολή μεταφέρει το πρόγραμμα μας στην πλακέτα (στο arduino μας). Αν πριν δεν έχουμε κάνει επικύρωση ή αν από τότε που κάναμε επικύρωση έχουμε αλλάξει κάτι, τότε το πρόγραμμα θα κάνει επικύρωση και μετά θα στείλει το πρόγραμμα σε περίπτωση που είναι σωστό, στο arduino μας.
3. **New(Νέο):** Αυτή η εντολή άπλα μας δίνει την δυνατότητα να ξεκινήσουμε από την αρχή να γράφουμε ένα νέο πρόγραμμα.
4. **Open (άνοιξε):** Αυτή η επιλογή μας επιτρέπει να ανοίξουμε ένα πρόγραμμα που έχουμε σώσει στο παρελθόν.

5. **Save (σώσε):** Με αυτή την επιλογή μπορούμε να σώσουμε το πρόγραμμα που έχουμε γράψει για να το δουλέψουμε ή χρησιμοποιήσουμε κάποια άλλη στιγμή.

#### 5.4 Δοκιμαστικός κώδικας

```
void setup() { pinMode(13, OUTPUT);
}

void loop() {
digitalWrite(13, HIGH);
delay(1000);
digitalWrite(13, LOW);
delay(1000);
}
```

#### 5.5 LCD Οθόνη

Εδώ θα δούμε πως συνδέεται μια οθόνη LCD (LiquidCrystalDisplay) με το Arduino uno και πως μπορούμε προγραμματιστικά να γράψουμε σε αυτή.

Το μέγεθος της οθόνης το μετράμε σε γραμμές επί στήλες (π.χ. 2x16, 4x20). Οι χαρακτήρες που μπορούν να χωρέσουν σε αυτή είναι ίσοι με το γινόμενο τους, έτσι για παράδειγμα αν έχουμε μια οθόνη 2x16 μπορούν να χωρέσουν σε αυτή 32 χαρακτήρες. Οι χαρακτήρες μπορεί να είναι νούμερα, γράμματα αγγλικού αλφάβητου, κενά καθώς και σύμβολα.

Η παραπάνω εικόνα δείχνει μια LCD οθόνη μεγέθους 2x16 χαρακτήρων μπλε φωτισμού. Μπορείτε να διακρίνετε τα blocks-τετραγωνάκια που αχνοφαίνονται τα οποία μπορούν να απεικονίσουν χαρακτήρες. Αν τα μετρήσετε θα δείτε πως είναι 32.

### 5.5.1 Συνδεσμολογία οθόνης LCD

**Κάθε οθόνη διαθέτει τα εξής pins:**

- Register Select (RS) pin

Καθορίζει σε πιο κομμάτι της μνήμης της οθόνης γίνεται η εγγραφή

- Read/Write (R/W) pin

Καθορίζει αν γίνεται εγγραφή ή ανάγνωση (Read/Write)

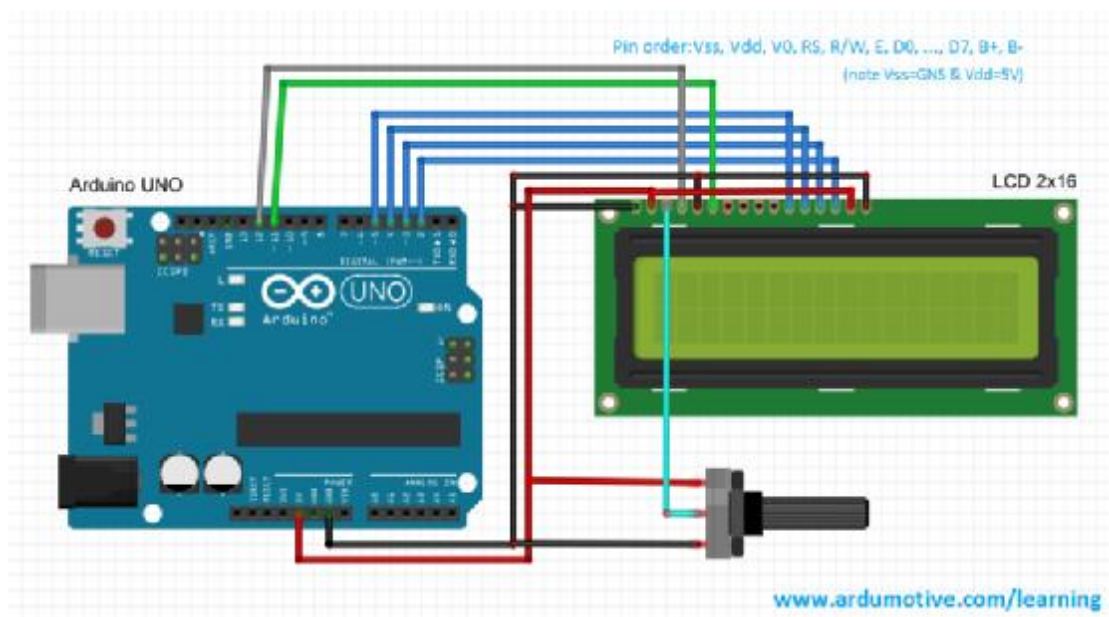
- Enable pin

Επιτρέπει να γίνει η χρήση της

- 8 data pins (D0 -D7)

Από αυτά περνάει η πληροφορία στην οθόνη

**Μεταξύ αυτών υπάρχουν και τα pins της τροφοδοσίας 5V και GND, το pin που ρυθμίζει την αντίθεση και συνήθως το συνδέουμε με ποτενσιόμετρο και τα pins για τον φωτισμό Bkl+ και Bkt- (ή B+, B-). Απ' τα 8 data pins εμείς θα χρησιμοποιούμε τα D4, D5, D6 και D7 (4bit mode).**





## 5.6 Κώδικας μέτρησης DHT-11

```
#include <dht.h>
```

```
#include <LiquidCrystal.h>
```

```
LiquidCrystal lcd(12, 11, 5, 4, 3, 2);
```

```
dht DHT;
```

```
#define DHT11_PIN 7
```

```
void setup(){  
  
    lcd.begin(16, 2);  
  
    }  
  
void loop()  
  
    {  
  
        int chk = DHT.read11(DHT11_PIN);  
  
        lcd.setCursor(0,0);  
  
        lcd.print("Thermokr:");  
  
        lcd.print(DHT.temperature);  
  
        lcd.print((char)223);  
  
        lcd.print("C");  
  
        lcd.setCursor(0,1);  
  
        lcd.print("ugrasia: ");  
  
        lcd.print(DHT.humidity);
```

```
lcd.print('%');  
  
delay(5000);  
  
}
```

## 5.7 Κώδικας μέτρησης DS18B20

```
#include <OneWire.h>  
  
#include <DallasTemperature.h>  
  
#define ONE_WIRE_BUS 2  
  
OneWire oneWire(ONE_WIRE_BUS);  
  
DallasTemperature sensors(&oneWire);  
  
void setup(void)  
  
{  
  
Serial.begin(9600);  
  
Serial.println("Dallas Temperature IC Control Library Demo");
```



```
sensors.begin();

}

void loop(void)

{

    Serial.print(" Requesting temperatures...");

    sensors.requestTemperatures();

    Serial.println("DONE");

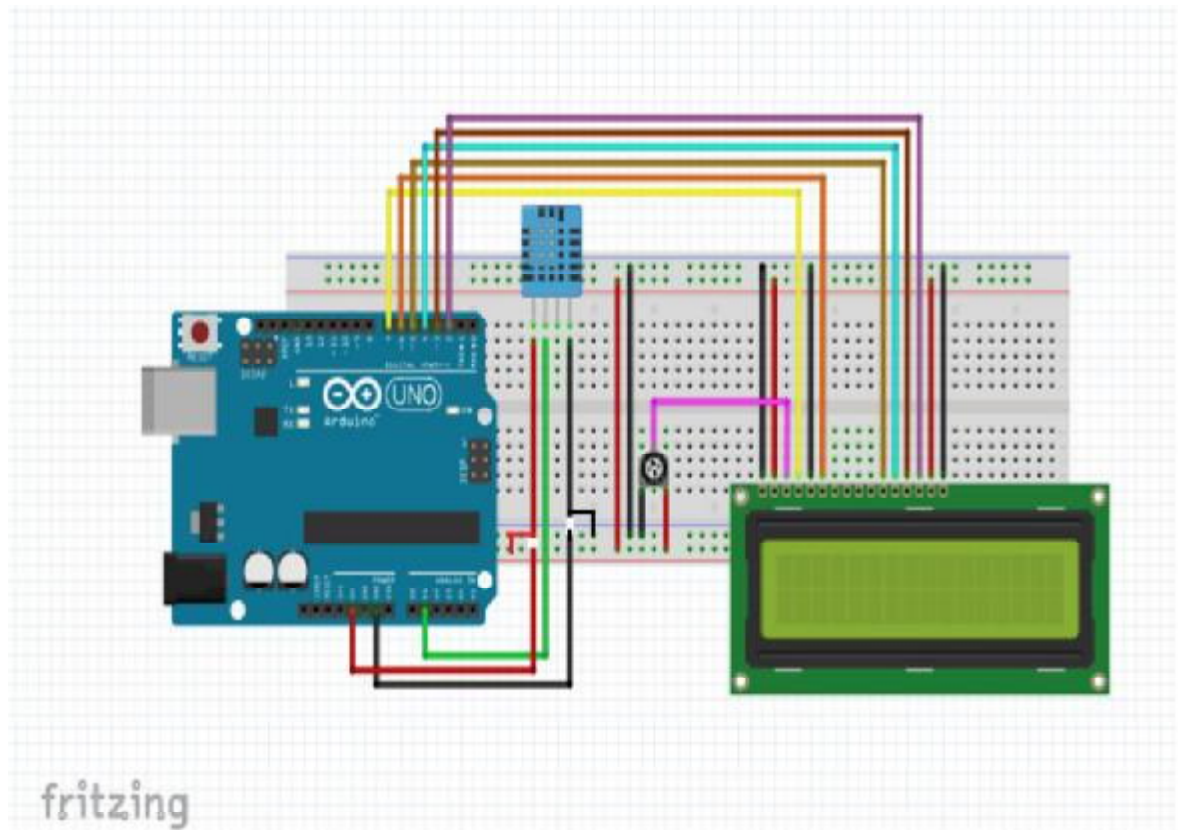
    Serial.print("Temperature is: ");

    Serial.print(sensors.getTempCByIndex(0));

    delay(1000);

}
```

## 5.8 ΣΥΝΔΕΣΜΟΛΟΓΙΑ ΚΥΚΛΩΜΑΤΟΣ



## **ΒΙΒΛΙΟΓΡΑΦΙΑ**

### Ελληνική Βιβλιογραφία

Άννα Μαστοράκη, (2014) Project Maniacs Εισαγωγή στο Arduino.

Βασιλική Παυλή, (2013) Η Διδασκαλία εκπαιδευτικής ρομποτικής με τη χρήση μικροελεγκτών (Arduino, PIC), Πτυχιακή Εργασία, Τμ Τεχνολογίας Πληροφορικής και Τηλεπικοινωνιών, ΑΤΕΙ Λάρισας.

Διομήδης Παναγιώτης, Μέγα Αναστασία (2015), Εφαρμογή Arduino μετρήσεων και καταγραφής μετεωρολογικών μεγεθών Πτυχιακή Εργασία, Τμ Μηχανικών Πληροφορικής ΤΕ, ΑΤΕΙ Θεσσαλίας.

Δρ. Βολογιαννίδης Σταύρος, (2009). Ευφυής Έλεγχος, Θεωρία και Εφαρμογής.

Μπαρδάκης Δημήτρης- Άγγελος, Μαρίνος Βαγγέλης (2014). Ιστορική αναδρομή στο Arduino.

### Ξένη Βιβλιογραφία

Enrique Ramos Melgar and Ciriaco Castro Diez ,Arduino And Kinect Projects

Hardware by Jonathan Oxeer and Hugh Blemings( 2009 )Beginning Arduino Practical Arduino cool Projects for Ope Source

Massino Banzi (2009), Getting Started With Arduino

### Διαδικτυακή βιβλιογραφία

<http://aesop.iep.edu.gr/node/18509> (Ανάκτηση 10/10/2018).

<https://el.wikipedia.org/wiki/Arduino> (Ανάκτηση 20/10/2018).

<https://deltahacker.gr/arduino-intro/> (Ανάκτηση 15/09/2018).

<https://grobotronics.com/arduino-boards-el.html> (Ανάκτηση 25/10/2018).

<http://www.ardumotive.com/how-to-use-dht-11-sensor-gr.html> (Ανάκτηση 03/10/2018).

<http://arduino.cc/en/Main/arduinoBoardUno> (Ανάκτηση 10/11/2018).

<http://arduino.cc/en/Guide/Environment?from=Tutorial.Bootloader> (Ανάκτηση 18/11/2018).

<http://arduino.cc/en/Guide/Windows> (Ανάκτηση 05/11/2018).

[http:// arduino.cc/en/Reference/HomePage](http://arduino.cc/en/Reference/HomePage) (Ανάκτηση 03/11/2018)

[http:// arduino.cc/en/Main/ArduinoBoardUno](http://arduino.cc/en/Main/ArduinoBoardUno) (Ανάκτηση 29/10/2018).

[http:// arduino.cc/en/Tutorial/Memory](http://arduino.cc/en/Tutorial/Memory) (Ανάκτηση 15/11/2018).

<http://users.ionio.gr/~floros/thesis/Thesis%20Triantafylloy.pdf> (Ανάκτηση 30/09/2018).

<http://www.eln.teilam.gr/sites/default/files/Lesson03.pdf> (Ανάκτηση 20/09/2018).