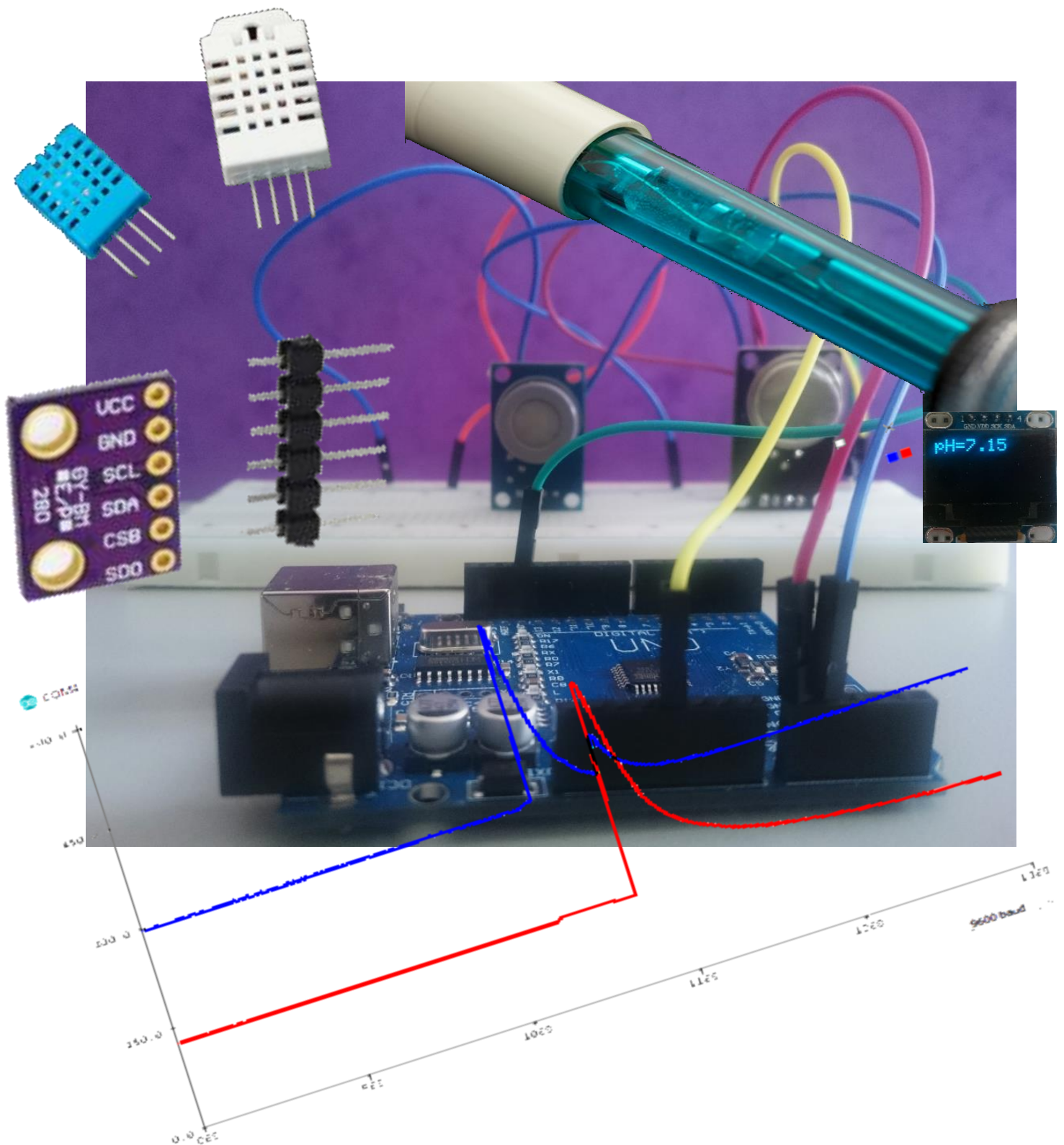


Κατασκευές με Arduino



Μαριάντζελα Κομνηνού

ΙΤΥΕ-ΔΙΟΦΑΝΤΟΣ

Η προσπάθεια αυτή έγινε στο πλαίσιο του προγράμματος πρακτικής άσκησης φοιτητών του Πανεπιστημίου Πατρών ως φοιτήτρια του Τμήματος της Επιστήμης των Υλικών. Η πρακτική έγινε στο Ινστιτούτο Τεχνολογίας Υπολογιστών με συνεργασία των Διευθύνσεων Τηλεματικής & Εφαρμογών Περιφερειακής Ανάπτυξης, Εκπαιδευτικής Τεχνολογίας και Τεχνικής Υποστήριξης Συστημάτων.

Η επίβλεψη της πρακτικής άσκησης έγινε από τον Διευθυντή της Διεύθυνσης Τηλεματικής και Εφαρμογών Περιφερειακής Ανάπτυξης καθηγητή του Τμ. Μηχανικών Η/Υ και Πληροφορικής κ. Γιάννη Γαροφαλάκη τον οποίο ευχαριστώ θερμά για την ευκαιρία που μου έδωσε να εκπαιδευτώ σε ένα δομημένο και επαγγελματικό περιβάλλον ανάμεσα σε εξαιρετικούς επιστήμονες.

Επίσης ευχαριστώ τα στελέχη των Διευθύνσεων του ΙΤΥΕ που με βοήθησαν στην μελέτη του υλικού, στην διόρθωση και μορφοποίηση των κειμένων, στην σύνδεση των υλικών, στην παροχή εξοπλισμού, και στην μεταφορά τεχνογνωσίας στην αξιοποίηση λογισμικού σχεδίασης, σε λογισμικό προσομοίωσης, σε πληθώρα βάσεων δεδομένων με αισθητήρες και μικροηλεκτρονικά.

Κατά τη διάρκεια της πρακτικής άσκησης ο σκοπός μας ήταν να δημιουργηθούν οι πρώτες σημειώσεις/οδηγοί για την αξιοποίηση των Edulabs που φτιάχτηκαν πιλοτικά για τα Δημόσια Σχολεία. Το υλικό θα αναρτηθεί στο Αποθετήριο Εφαρμογών του Υπουργείου Παιδείας και Θρησκευμάτων και των Εποπτευόμενων Φορέων (<https://github.com/itminedu>).

Πρώτα δημιουργήσαμε ένα οδηγό για τα Βασικά της γλώσσας C στον προγραμματισμό της πλατφόρμας μικροϋπολογιστών της οικογένειας των Arduino. Για τον προγραμματισμό και τα παραδείγματα χρησιμοποιήθηκαν ανοιχτά/ελεύθερα λογισμικά όπως ο compiler TDM-GCC MinGW (<https://sourceforge.net/projects/tdm-gcc/>) και το περιβάλλον Code::Blocks v.16.01 (<http://www.codeblocks.org/> Open source, cross platform, free C, C++ and Fortran IDE).

Κατόπιν με ένα δεύτερο οδηγό περιγράψαμε τις ιδιαίτερες πτυχές του προγραμματισμού σε Arduino IDE (www.arduino.cc), τα βασικά ηλεκτρονικά και ηλεκτρολογικά υλικά που χρησιμοποιούνται στην πλατφόρμα και δημιουργήσαμε παραδείγματα χρήσης/υλοποίησης με αυξανόμενο βαθμό δυσκολίας.

Έπειτα, δημιουργήθηκε ένας Οδηγός «Ξεκινήστε γρήγορα με το Arduino» στον οποίο είναι συγκεντρωμένες οι εντολές της C, του Arduino και βασικά ηλεκτρονικά.

Τέλος με τα στοιχεία αυτά φτιάχτηκαν οδηγίες υλοποίησης για 4 παραδείγματα / πειράματα στην πράξη.

Μέσα από την πρακτική άσκηση βελτίωσα τις γνώσεις μου, αξιοποιώντας αυτά που έμαθα στη σχολή μου υπό την οπτική του μηχανικού, του προγραμματιστή και του εκπαιδευτή.

Ευχαριστώ ιδιαίτερα τον καθηγητή μου κ. Δημήτρη Αλεξανδρόπουλο στο Τμήμα Επιστήμης των Υλικών και τον καθηγητή Γιάννη Γαροφαλάκη που με επέβλεπαν ως υπεύθυνοι της πρακτικής εργασία μου στο ΙΤΥΕ.

Μαρία-Αγγελική Κομνηνού

Πάτρα, Δεκέμβριος 2016-Φεβρουάριος 2017

ΠΕΡΙΕΧΟΜΕΝΑ

1.	Αισθητήρας Θερμοκρασίας-Υγρασίας.....	4
1.1	Στόχος.....	4
1.2	Αισθητήρες θερμοκρασίας-υγρασίας DHT11 vs DHT22	4
1.3	Υλικά που χρησιμοποιούνται	5
1.4	Συνδεσμολογία	6
1.5	Κώδικας	11
1.6	Εμφάνιση Αποτελεσμάτων.....	12
2.	Βαρομετρικός-Θερμοκρασιακός Αισθητήρας.....	13
2.1	Στόχος.....	13
2.2	Μορφή Αισθητήρα	13
2.2.1	Πρωτόκολλο I ² C.....	13
2.3	Υλικά που χρησιμοποιούνται	15
2.4	Συνδεσμολογία	16
2.5	Κώδικας.....	18
3.	Βαρομετρικός-Θερμοκρασιακός Αισθητήρας με OLED οθόνη.....	22
3.1	Στόχος.....	22
3.2	Μορφή OLED οθόνης.....	22
3.3	Υλικά που χρησιμοποιούνται	22
3.4	Συνδεσμολογία	23
3.5	Κώδικας.....	25
4.	Αισθητήρας Αερίων.....	26
4.1	Περίληψη.....	26
4.2	Είδη Αισθητήρων Αερίων	26
4.3	Υλικά που χρησιμοποιούνται	27
4.4	Συνδεσμολογία	28
4.5	Κώδικας.....	29
4.6	Εμφάνιση Αποτελεσμάτων.....	30
5.	Αισθητήρας pH.....	31
5.1	Στόχος.....	31
5.2	Χαρακτηριστικά αισθητήρα	31
5.3	Υλικά που χρησιμοποιήθηκαν.....	31
5.4	Συνδεσμολογία	32
5.5	Καλιμπράρισμα pHμετρου	33
5.6	Κώδικας.....	34

Στον παρόντα οδηγό περιγράφουμε τα βήματα υλοποίησης κατασκευών με χρήση αισθητήρων και μικροελεγκτών βασισμένων στα Arduino Boards. Για την κάθε κατασκευή αναφέρεται ο στόχος της κατασκευής, τα υλικά που θα χρειαστούν, η περιγραφή τους, η συνδεσμολογία τους, το πρόγραμμα που θα εκτελείται στο Arduino και το αποτέλεσμα του.

Οι κατασκευές είναι αυξανόμενης δυσκολίας και πολυπλοκότητας και περιέχουν πολλές από τις αρχές σχεδιασμού, προγραμματισμού και υλοποίησης κατασκευών με μικροελεγκτές/μικροεπεξεργαστές.

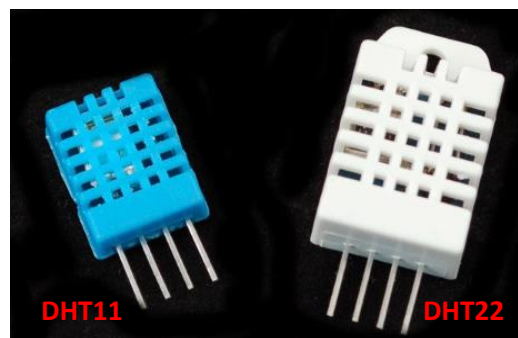
1. Αισθητήρας Θερμοκρασίας-Υγρασίας

1.1 Στόχος

Αυτό το παράδειγμα αφορά την δημιουργία κατασκευής ενός -χαμηλού κόστους- αισθητήρα θερμοκρασίας & υγρασίας. Για την κατασκευή υπάρχουν πολλών ειδών αισθητήρες με απλούστερους και οικονομικότερους τους αισθητήρες της οικογένειας DHT. Οι DHT αισθητήρες αποτελούνται από δύο τμήματα, ένα χωρητικό αισθητήρα υγρασίας κι ένα θερμίστορ (θερμικός αντιστάτης). Επιπλέον, περιέχουν ένα τσιπ που μετατρέπει το σήμα από αναλογικό σε ψηφιακό και δίνει ψηφιακά τη θερμοκρασία και την υγρασία. Το ψηφιακό σήμα είναι αρκετά εύκολο να διαβαστεί χρησιμοποιώντας οποιοδήποτε Arduino board.

1.2 Αισθητήρες θερμοκρασίας-υγρασίας DHT11 vs DHT22

Υπάρχουν δύο είδη DHT αισθητήρων, οι οποίοι είναι σχεδόν ίδιοι εξωτερικά αλλά διαφέρουν τα χαρακτηριστικά τους.



DHT11 Αισθητήρας

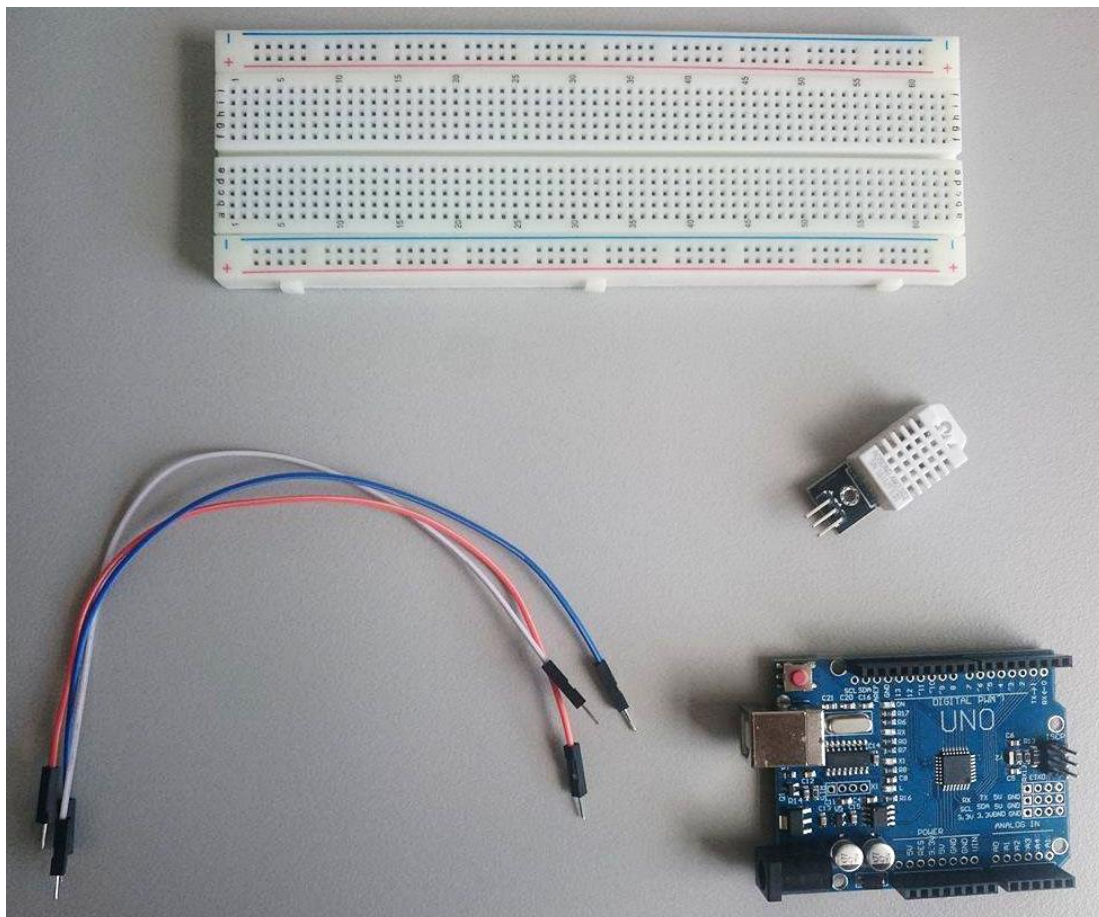
- Πάρα πολύ φθηνός (~1€)
- Τάση λειτουργίας 3V ως 5V
- Ρεύμα σε λειτουργία σε μέγιστη χρήση (όσο ζητά δεδομένα) 2.5mA
- Μέτρηση υγρασίας από 20% έως 80% με 5% ακρίβεια και ευαισθησία 1%
- Μέτρηση θερμοκρασίας από 0-50 °C με ακρίβεια ± 2 °C και ευαισθησία 1%
- Ρυθμός δειγματοληψίας ≤ 1 Hz (ένα δείγμα/δευτερόλεπτο)
- Μέγεθος 15.5mm x 12mm x 5.5mm
- Συνδεσμολογία με 4 pins

DHT22 Αισθητήρας

- Φθηνός (~2,5-3€ ανάλογα με το αν είναι σε πλακέτα)
- Τάση λειτουργίας 3V με 5V
- Ρεύμα με λειτουργία σε μέγιστη χρήση (όσο ζητά δεδομένα) 2.5mA
- Μέτρηση υγρασίας από 0 έως 100% με $\pm 2\%$ ακρίβεια και ευαισθησία 0.1%
- Θερμοκρασίες από -40-125 °C με ακρίβεια ± 0.5 °C και ευαισθησία 0.1 °C
- Ρυθμός δειγματοληψίας ≤ 0.5 Hz (ένα δείγμα / δύο δευτερόλεπτα)
- Μέγεθος 15.1mm x 25mm x 7.7mm
- Συνδεσμολογία με 4 pins

1.3 Υλικά που χρησιμοποιούνται

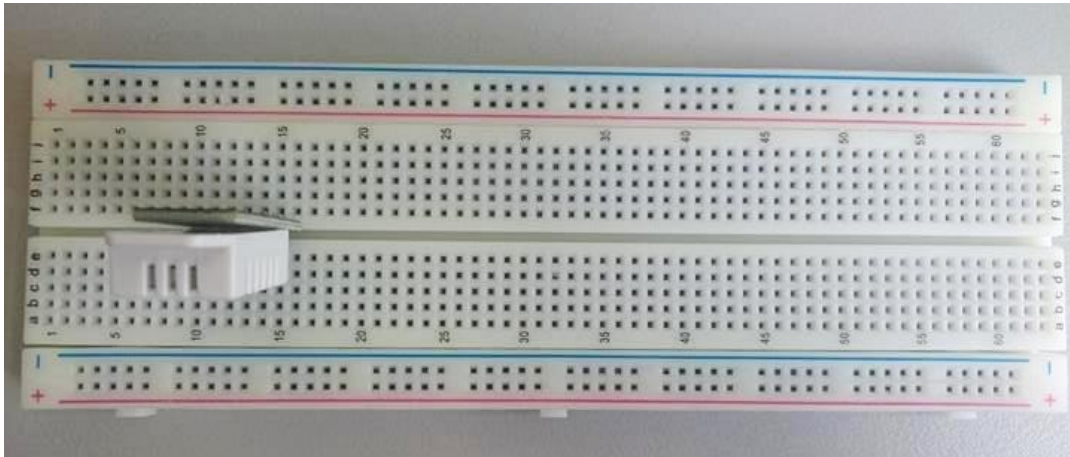
- Arduino Uno board
- Arduino Breadboard
- 3 καλώδια αρσενικό-αρσενικό
- Θα χρησιμοποιήσουμε τον αισθητήρα θερμοκρασίας & υγρασίας DHT22 σε πλακέτα που έχει 3 pins (+, data, -) και που για λίγο μεγαλύτερη τιμή παρέχει μεγαλύτερη ακρίβεια.



1.4 Συνδεσμολογία

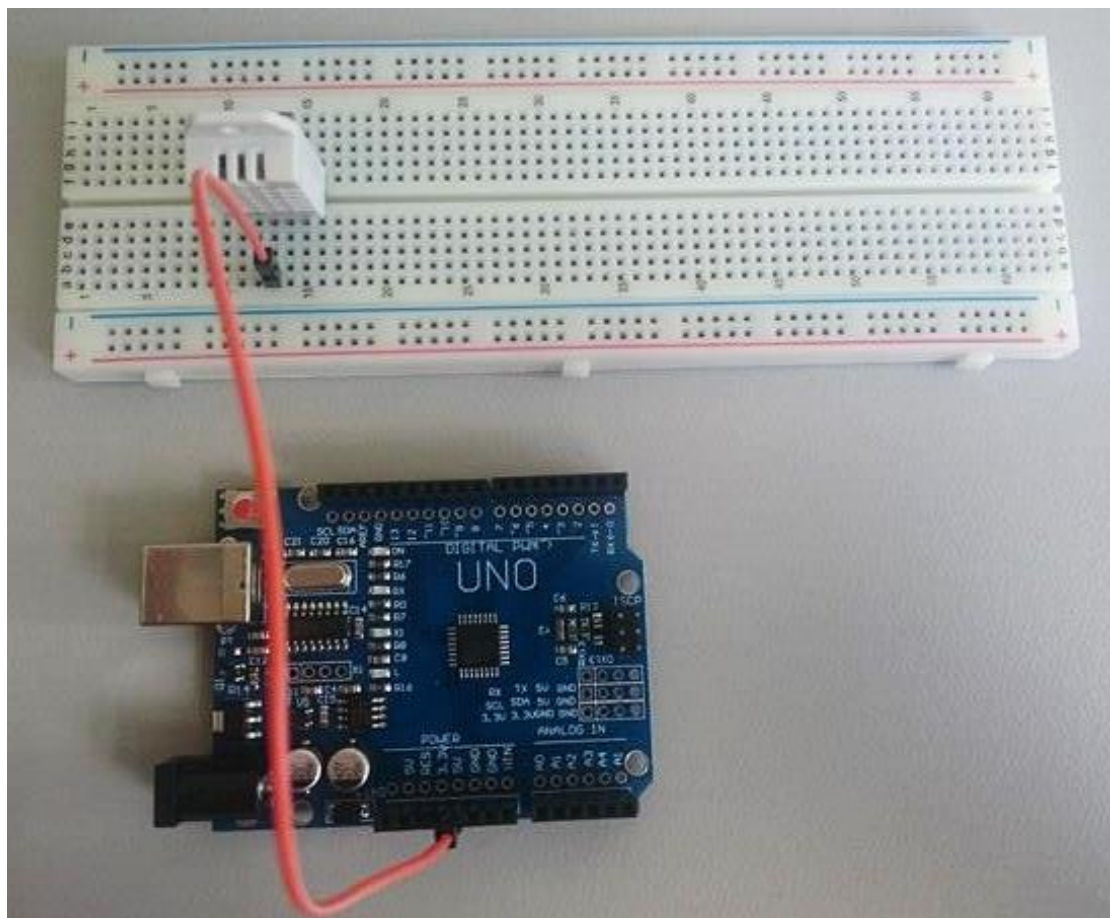
1^ο Βήμα

Αρχικά συνδέουμε τον αισθητήρα μας πάνω στην breadboard και προσέχουμε τα 3 pins να είναι στην ίδια σειρά τοποθετημένα (στην προκειμένη περίπτωση στην e).



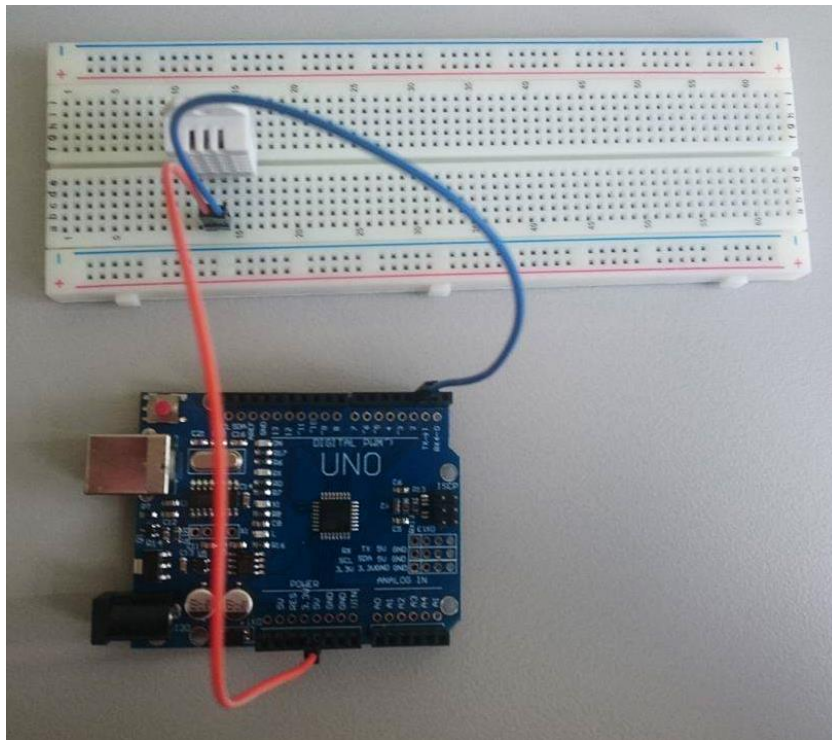
2ο Βήμα

Συνδέουμε το ένα άκρο του πρώτου καλωδίου στο 1^ο από τα 3 «ποδαράκια» (+) (pins) του αισθητήρα ενώ το άλλο στα 5V.



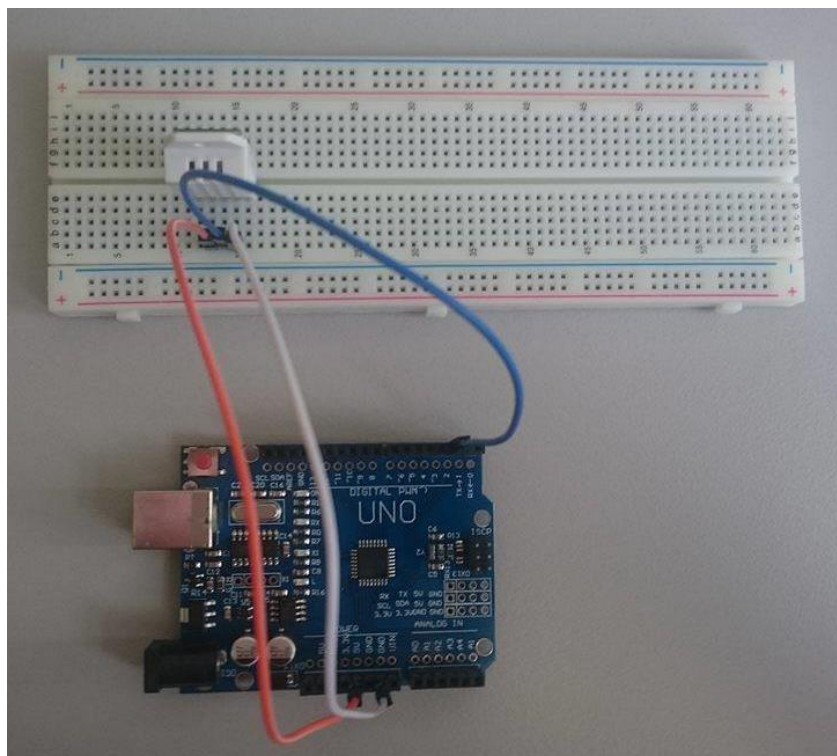
3^ο Βήμα

Προσθέτουμε το 2^ο καλώδιο με το ένα άκρο του να συνδέεται στο 2^ο «ποδαράκι» του αισθητήρα (data) και το άλλο στο ψηφιακό pin 2.



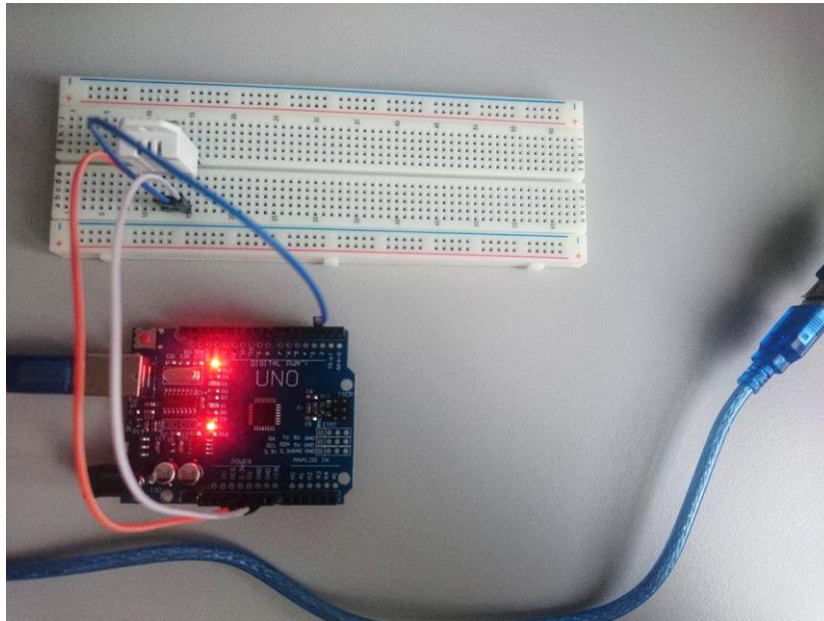
4^ο Βήμα

Συνδέουμε και το τελευταίο καλώδιο που ενώνει το 3^ο «ποδαράκι» του αισθητήρα (-) με τη γείωση (GND).



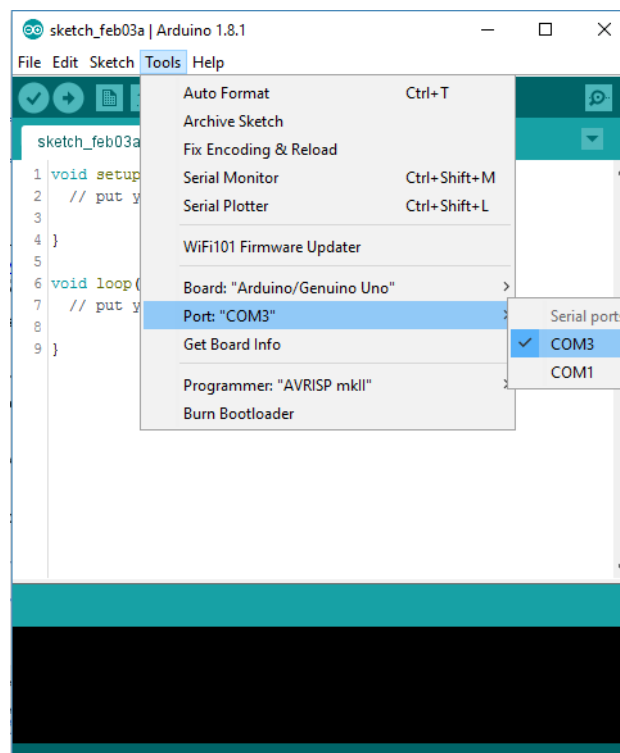
5ο Βήμα

Συνδέουμε τη διάταξη με τον υπολογιστή.



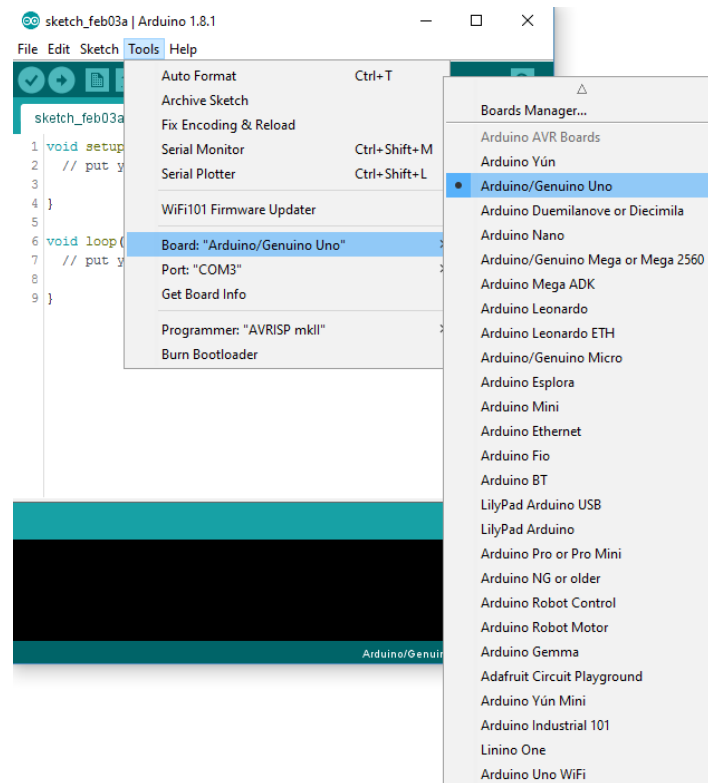
6ο Βήμα

Μετά τη σύνδεση του Arduino board στην USB θύρα του υπολογιστή και την εκκίνηση του Arduino IDE επιλέγουμε στο Tools-Port την νέα σειριακή θύρα που έχει εμφανιστεί π.χ. COM3





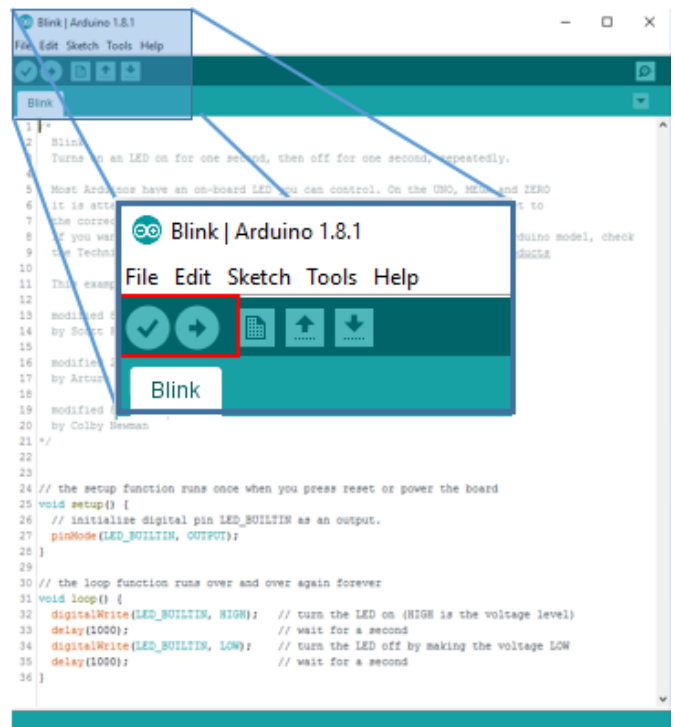
7ο Βήμα

Επιλέγουμε τον τύπο του Arduino board που έχουμε στη διάθεσή μας από το Tools-Board.



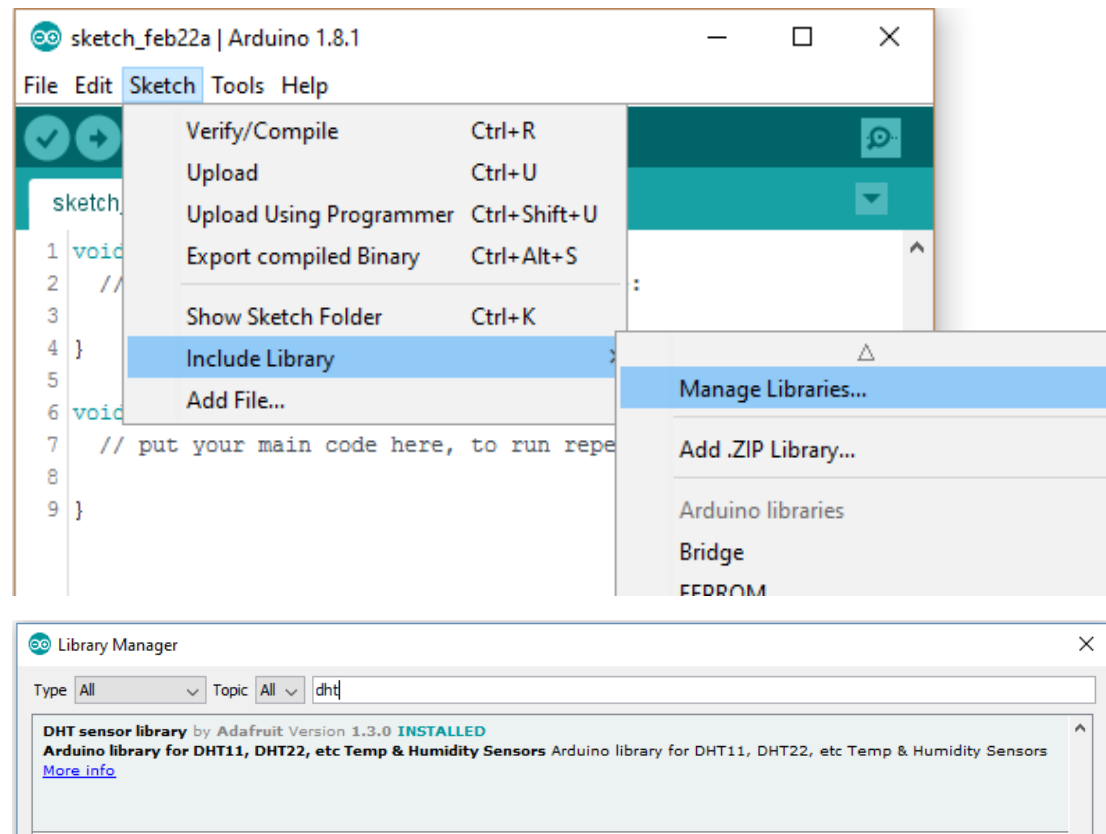
8ο Βήμα

Αφού γράψουμε τον κώδικα και είναι όλα έτοιμα για να τρέξει το πρόγραμμα, πρώτα πατάμε το κουμπί  για να γίνει ο έλεγχος σφαλμάτων και να «μεταγλωττιστεί» όπως λέμε το πρόγραμμα C σε γλώσσα που καταλαβαίνει ο επεξεργαστής του Arduino και μετά το κουμπί  για να «φορτωθεί» το πρόγραμμα στη μνήμη του Arduino και να αρχίσει να εκτελείται.



Σε περίπτωση που υπάρχει ένδειξη σφάλματος γιατί λείπει κάποια βιβλιοθήκη την εγκαθιστούμε με χρήση του menu **Sketch-Include Library-Manage Libraries...**

Κάνουμε αναζήτηση για DHT και εγκαθιστούμε την βιβλιοθήκη DHT sensor library by Adafruit ver 1.3.0



1.5 Κώδικας

```
// Παράδειγμα για DHT αισθητήρα θερμοκρασίας/υγρασίας

#include "DHT.h" // βιβλιοθήκη αισθητήρα

#define DHTPIN 2    // ψηφιακό pin στο οποίο συνδέουμε

#define DHTTYPE DHT22 // DHT 22 (AM2302), AM2321

// Αρχικοποιείται ο DHT αισθητήρας
DHT dht(DHTPIN, DHTTYPE);

void setup() {
    Serial.begin(9600); //αρχικοποιείται η επικοινωνία στα 9600 bytes
    Serial.println("DHTxx test!");

    dht.begin();
}

void loop() {
    // Αναμονή για 2 ms μεταξύ των μετρήσεων
    delay(2000);

    // Η ανάγνωση θερμοκρασίας ή υγρασίας παίρνει περίπου 250 milliseconds!

    float h = dht.readHumidity(); //
    // Διαβάζεται η θερμοκρασία σε Celsius
    float t = dht.readTemperature();

    // Εάν η εντολή "Διάβασε" αποτύχει σταματάει η επανάληψη
    if (isnan(h) || isnan(t)) {
        Serial.println("Failed to read from DHT sensor!");
        return;
    }

    //Υπολογίζεται ο δείκτης θερμότητας σε Celsius
    float hic = dht.computeHeatIndex(t, h, false);

    Serial.print("Humidity: ");
    Serial.print(h); //τυπώνεται η υγρασία ψηφιακά
    Serial.print(" %\t");
    Serial.print("Temperature: ");
    Serial.print(t); //τυπώνεται η θερμοκρασία ψηφιακά
    Serial.print(" *C ");

    Serial.print("Heat index: ");
    Serial.print(hic); //τυπώνεται ο δείκτης θερμότητας ψηφιακά
    Serial.println(" *C ");
}
```

1.6 Εμφάνιση Αποτελεσμάτων

Αφού ο κώδικας έχει γραφτεί και φορτωθεί στο Arduino με βάση τα βήματα του υποκεφαλαίου 1.4, πατώντας το κουμπί “Serial Monitor” εμφανίζεται η οθόνη με τη συλλογή των δεδομένων.

COM4

DHTxx test!

Humidity: 36.20 %	Temperature: 23.40 *C	Heat index: 22.74 *C
Humidity: 36.10 %	Temperature: 23.40 *C	Heat index: 22.74 *C
Humidity: 35.80 %	Temperature: 23.40 *C	Heat index: 22.73 *C
Humidity: 35.70 %	Temperature: 23.50 *C	Heat index: 22.84 *C
Humidity: 35.60 %	Temperature: 23.50 *C	Heat index: 22.83 *C
Humidity: 35.80 %	Temperature: 23.50 *C	Heat index: 22.84 *C
Humidity: 35.70 %	Temperature: 23.50 *C	Heat index: 22.84 *C
Humidity: 35.60 %	Temperature: 23.50 *C	Heat index: 22.83 *C
Humidity: 35.50 %	Temperature: 23.40 *C	Heat index: 22.72 *C
Humidity: 35.60 %	Temperature: 23.50 *C	Heat index: 22.83 *C
Humidity: 35.50 %	Temperature: 23.50 *C	Heat index: 22.83 *C
Humidity: 35.50 %	Temperature: 23.50 *C	Heat index: 22.83 *C
Humidity: 35.50 %	Temperature: 23.50 *C	Heat index: 22.83 *C
Humidity: 35.50 %	Temperature: 23.50 *C	Heat index: 22.83 *C
Humidity: 36.50 %	Temperature: 23.40 *C	Heat index: 22.75 *C
Humidity: 36.90 %	Temperature: 23.40 *C	Heat index: 22.76 *C
Humidity: 36.90 %	Temperature: 23.50 *C	Heat index: 22.87 *C
Humidity: 36.50 %	Temperature: 23.50 *C	Heat index: 22.86 *C
Humidity: 36.10 %	Temperature: 23.50 *C	Heat index: 22.85 *C
Humidity: 36.10 %	Temperature: 23.50 *C	Heat index: 22.85 *C
Humidity: 36.10 %	Temperature: 23.50 *C	Heat index: 22.85 *C
Humidity: 35.60 %	Temperature: 23.50 *C	Heat index: 22.83 *C
Humidity: 35.50 %	Temperature: 23.50 *C	Heat index: 22.83 *C
Humidity: 36.10 %	Temperature: 23.50 *C	Heat index: 22.85 *C
Humidity: 36.10 %	Temperature: 23.40 *C	Heat index: 22.74 *C
Humidity: 36.30 %	Temperature: 23.50 *C	Heat index: 22.85 *C
Humidity: 36.30 %	Temperature: 23.50 *C	Heat index: 22.85 *C
Humidity: 36.10 %	Temperature: 23.50 *C	Heat index: 22.85 *C
Humidity: 36.60 %	Temperature: 23.50 *C	Heat index: 22.86 *C
Humidity: 37.60 %	Temperature: 23.50 *C	Heat index: 22.89 *C

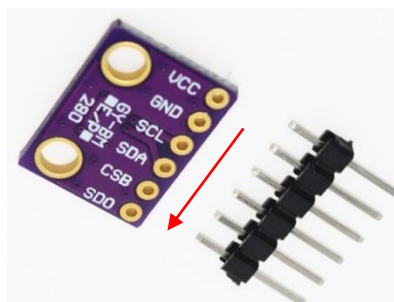
2. Βαρομετρικός-Θερμοκρασιακός Αισθητήρας

2.1 Στόχος

Στην κατασκευή αυτή θα φτιάξουμε ένα σύστημα μέτρησης βαρομετρικής πίεσης και θερμοκρασίας με ένα αισθητήρα. Όπως έχουμε δει υπάρχει πληθώρα αισθητήρων για να διαλέξει κάποιος και το τι θα επιλέξει εξαρτάται από την εφαρμογή, την ακρίβεια των μετρήσεων, το κόστος και την πολυπλοκότητα. Σε αυτή την κατασκευή θα επιλέξουμε τον αισθητήρα BMP280, ο οποίος συνδέεται με χρήση του I2C πρωτοκόλλου επικοινωνίας που θα δούμε παρακάτω. Ο αισθητήρας αυτός μπορεί να μετρήσει με μεγάλη ακρίβεια την βαρομετρική πίεση και θερμοκρασία. Δεδομένου ότι η βαρομετρική πίεση αλλάζει ανάλογα με το υψόμετρο, ο αισθητήρας αυτός μπορεί να μετρήσει και το ύψος με ακρίβεια ± 1 μέτρο! Αρκεί να γίνει μία μέτρηση στο επίπεδο της θάλασσας για αναφορά. Η ακρίβεια της βαρομετρικής πίεσης είναι στο ± 1 hPa και της θερμοκρασίας στους $\pm 1.0^\circ$ βαθμούς κελσίου.

2.2 Μορφή Αισθητήρα

Ο αισθητήρας BMP280 που χρησιμοποιούμε έχει έξι «ποδαράκια». Το 1^ο που ονομάζεται VCC συνδέεται με το pin 3V του Arduino, το 2^ο που είναι το GND συνδέεται με τη γείωση του Arduino, το 3^ο το οποίο λέγεται CSL συνδέεται με το αντίστοιχο του Arduino ενώ το 4^ο το οποίο είναι το SDA συνδέεται με το αντίστοιχο του Arduino. Τα υπόλοιπα δυο pin παραμένουν ελεύθερα.



Σε κάποια Arduino όπως παραδείγματος χάριν στα παλαιότερα Uno, δεν υπάρχουν ξεχωριστά CSL και SDA pins και το 3^ο και 4^ο pin συνδέεται στις θέσεις A4 και A5 αντίστοιχα. Το Arduino Uno ver 3 που χρησιμοποιούμε εμείς, διαθέτει τα pin CSL & SDA που υποστηρίζουν το πρωτόκολλο I²C (Inter-Integrated Circuit or “Two-Wire”, γράφεται και I2C).

2.2.1 Πρωτόκολλο I²C

Αυτό το πρωτόκολλο είναι πολύ χρήσιμο καθώς μας επιτρέπει να συνδέσουμε πολλαπλές συσκευές, σε δύο καλώδια και όλες μπορούν να μοιράζονται τα δύο ίδια pins: τα SCL και SDA. Η γραμμή SCL είναι η γραμμή ρολογιού, ενώ η SDA είναι η γραμμή δεδομένων. Οι γραμμές αυτές συνδέονται σε όλες τις συσκευές, που υπάρχουν πάνω στο δίαυλο PC. Προφανώς εκτός από τα παραπάνω καλώδια που μεταφέρουν δεδομένα, απαιτείται και δύο επιπλέον καλώδια για την τροφοδοσία με ρεύμα το VCC ή V+ και η γείωση (GND) ή V-. Η κάθε μία συσκευή έχει μια διεύθυνση που είναι ένας δεκαεξαδικός αριθμός που συνήθως δίνεται από τον κατασκευαστή και είναι χαρακτηριστική για τον κάθε αισθητήρα. Η διεύθυνση αυτή χρησιμοποιείται όταν πρέπει να ζητήσουμε δεδομένα από τον αισθητήρα και για να ξεχωρίσουμε από ποιον αισθητήρα παίρνουμε δεδομένα. Φυσικά ο τρόπος που διαβάζουμε δεδομένα ή αλληλοεπιδρούμε με τον αισθητήρα γίνεται απλούστερος με τη χρήση των κατάλληλων βιβλιοθηκών που απλοποιούν τον προγραμματισμό.

Σε περίπτωση που δεν είναι γνωστή η διεύθυνση που χρησιμοποιεί ο αισθητήρας στο πρωτόκολλο I2C τρέχουμε ένα πρόγραμμα (μπορούμε να το βρούμε στο site <http://playground.arduino.cc/Main/I2cScanner>) το οποίο θα μας δώσει τη δεκαεξαδική διεύθυνση του αισθητήρα:

```
#include <Wire.h>

void setup()
{
  Wire.begin();

  Serial.begin(9600);
  while (!Serial);
  Serial.println("\nI2C Scanner");
}

void loop()
{
  byte error, address;
  int nDevices;

  Serial.println("Scanning...");

  nDevices = 0;
  for(address = 1; address < 127; address++)
  {

    Wire.beginTransmission(address);
    error = Wire.endTransmission();

    if (error == 0)
    {
      Serial.print("I2C device found at address 0x");
      if (address<16)
        Serial.print("0");
      Serial.print(address, HEX);
      Serial.println(" !");

      nDevices++;
    }
    else if (error==4)
    {
      Serial.print("Unknow error at address 0x");
      if (address<16)
        Serial.print("0");
      Serial.println(address, HEX);
    }
  }
  if (nDevices == 0)
    Serial.println("No I2C devices found\n");
  else
    Serial.println("done\n");

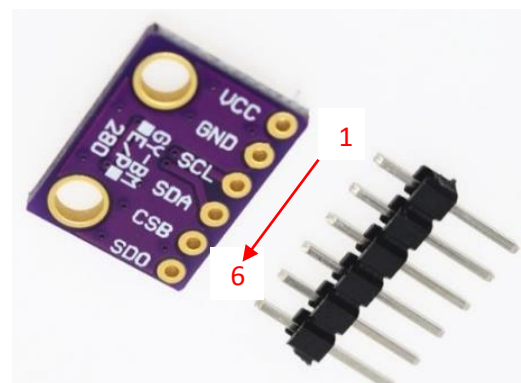
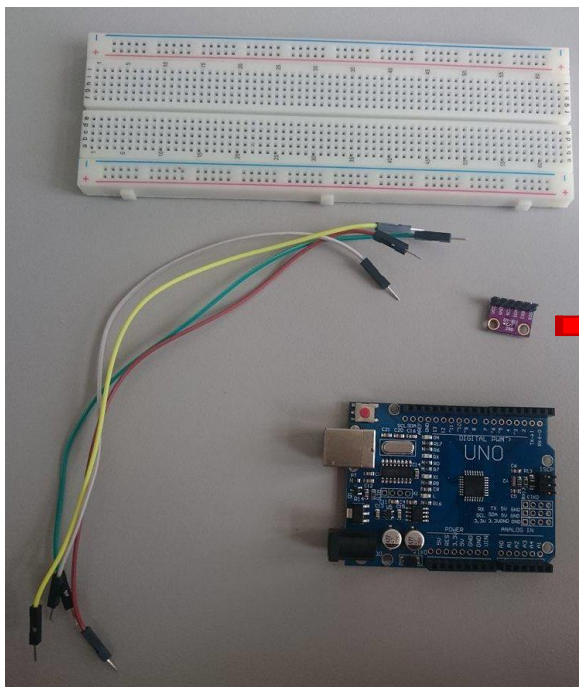
  delay(5000);
}
```

Έτσι στην έξοδο έχουμε την διεύθυνση του αισθητήρα (0x76).



2.3 Υλικά που χρησιμοποιούνται

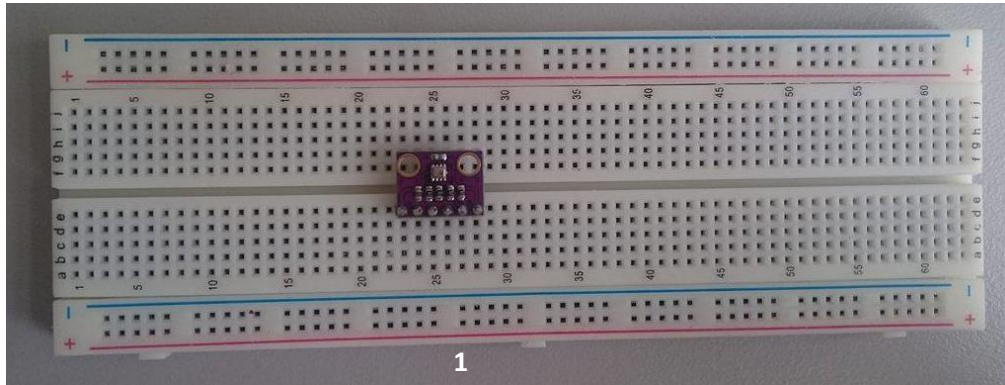
- Arduino Uno board
- Arduino Breadboard
- 4 καλώδια αρσενικό-αρσενικό
- bmp280 αισθητήρας θερμοκρασίας & βαρόμετρο



2.4 Συνδεσμολογία

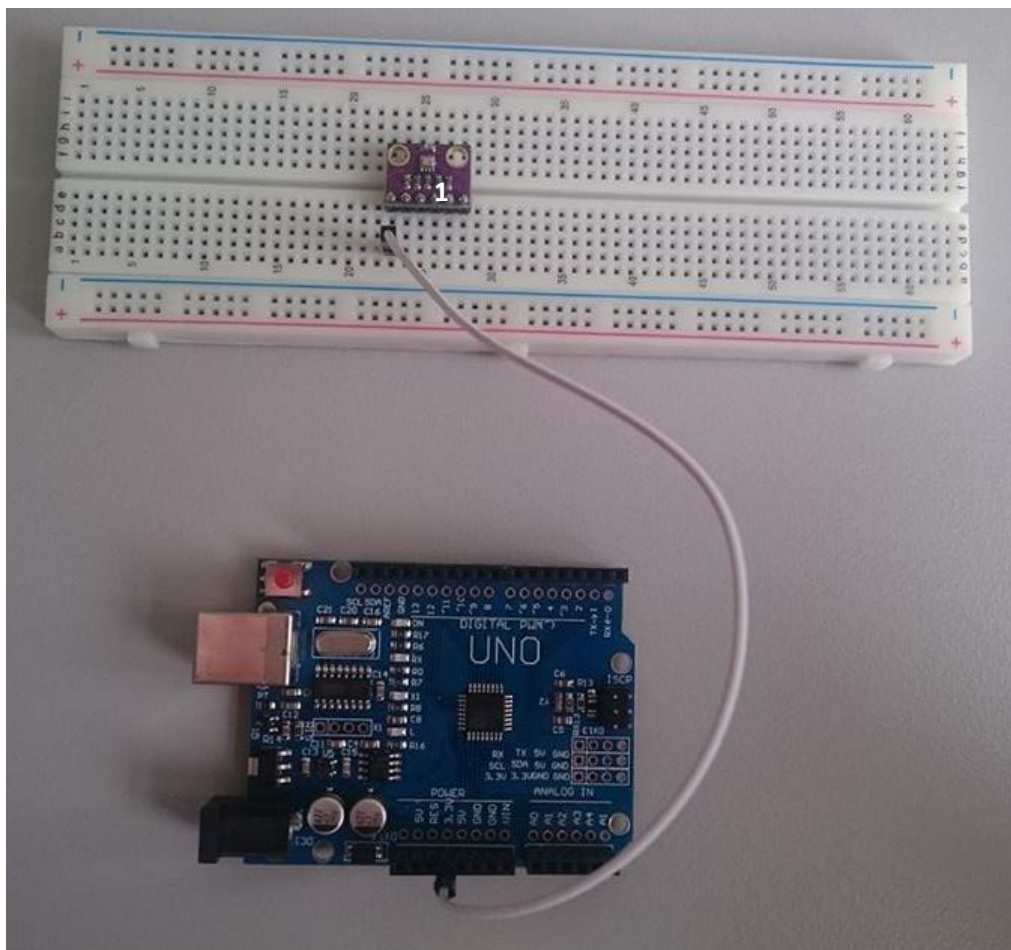
1^ο Βήμα

Αρχικά συνδέουμε τον αισθητήρα μας πάνω στην breadboard και προσέχουμε τα 6 pins να είναι στην ίδια σειρά τοποθετημένα (στην προκειμένη περίπτωση στην e).



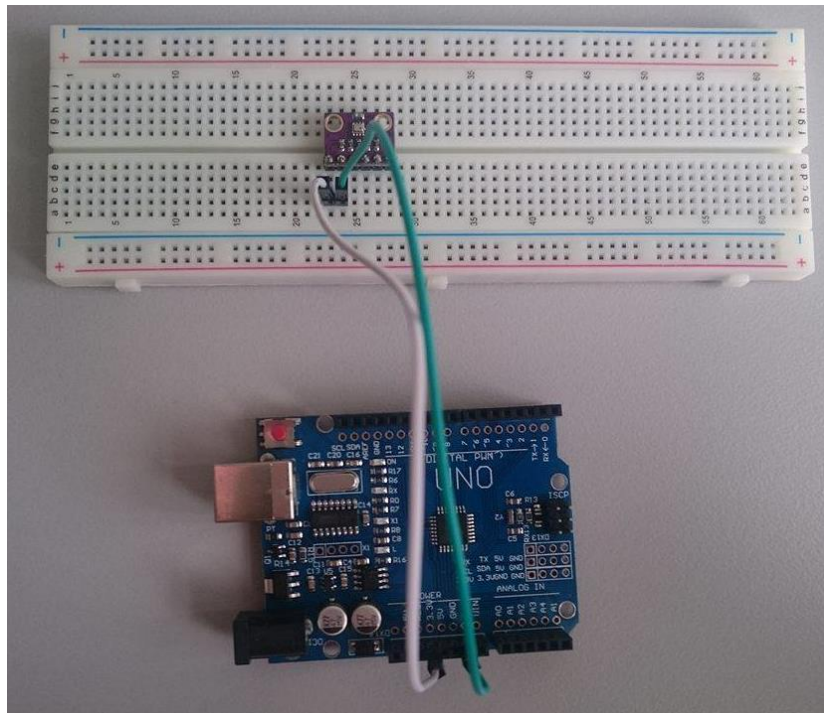
2^ο Βήμα

Συνδέουμε το ένα άκρο του πρώτου καλωδίου στο Vcc (1^ο από τα 6 pins) του αισθητήρα ενώ το άλλο στα 3.3V του Arduino.



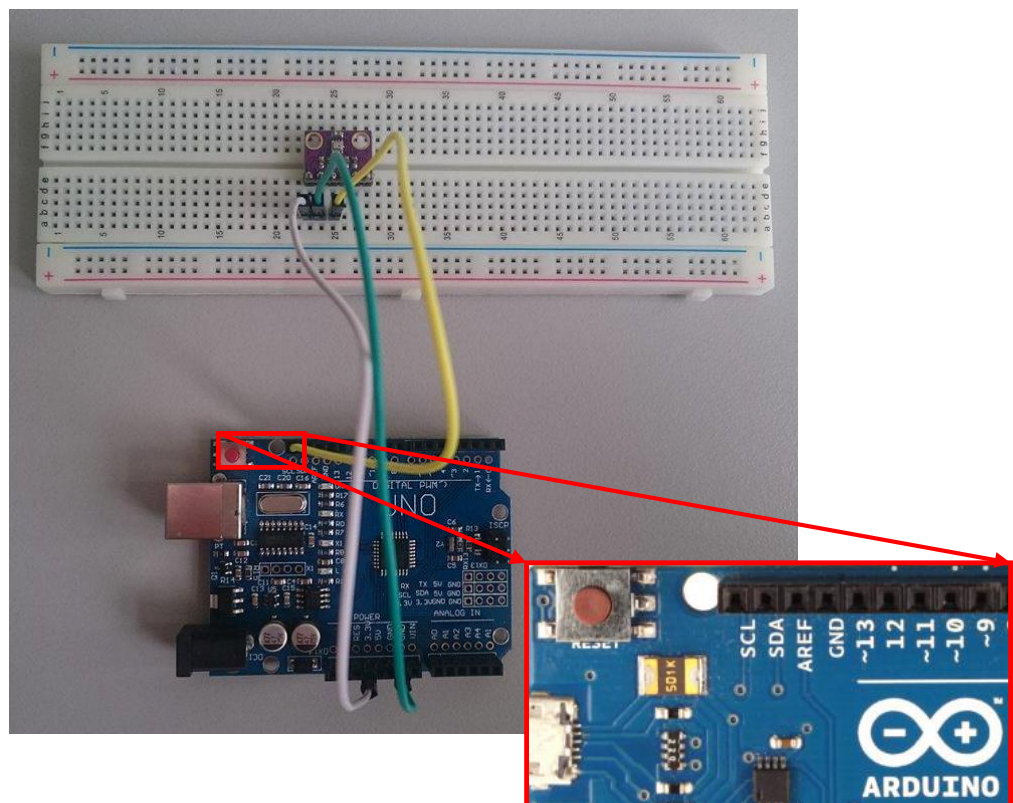
3^ο Βήμα

Συνδέουμε το 2^ο καλώδιο με το ένα άκρο του να συνδέεται στη γείωση (2^ο pin) του αισθητήρα και το άλλο στη γείωση (GND) του Arduino



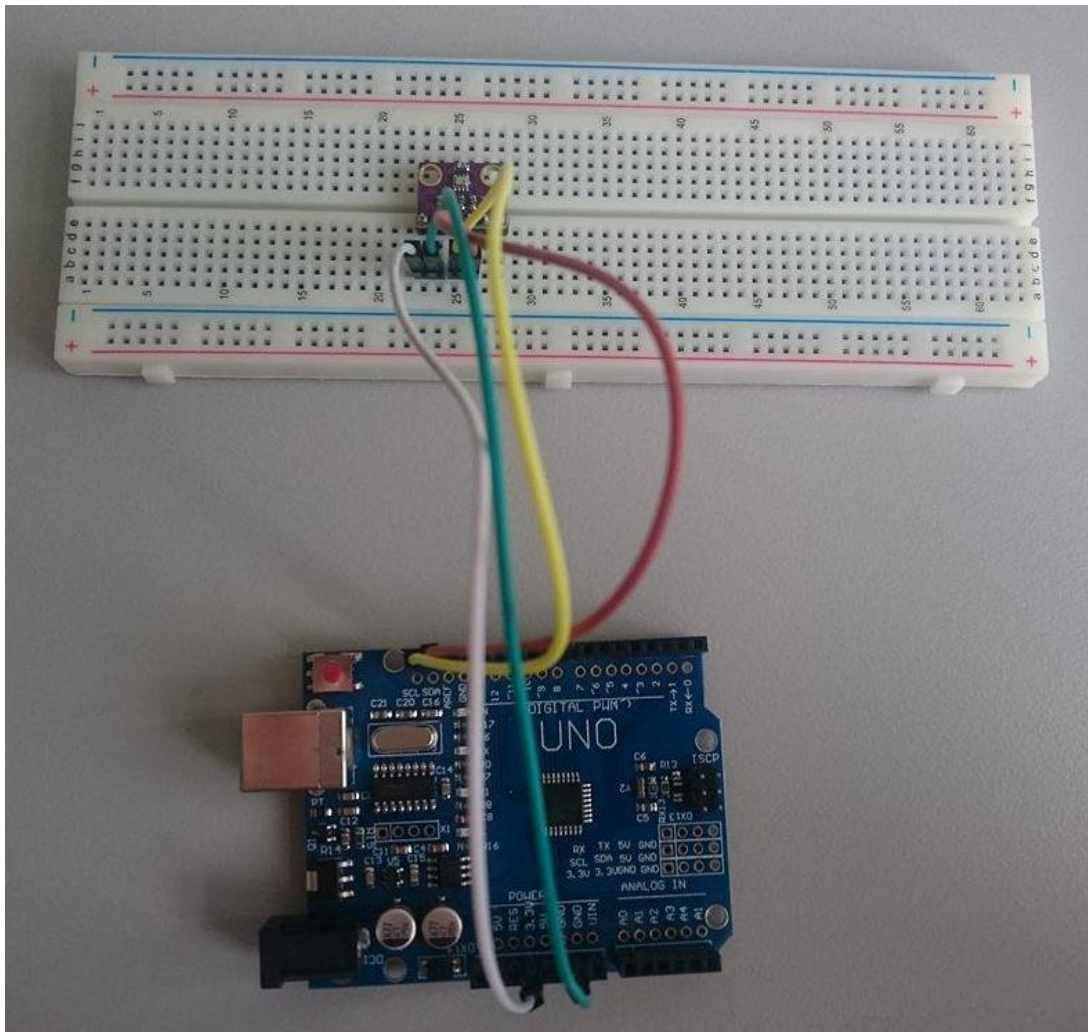
4^ο Βήμα

Συνδέουμε το 3^ο καλώδιο που ενώνει το SCL (3^ο pin) του αισθητήρα με το pin SCL του Arduino.



5ο Βήμα

Συνδέουμε το 4^ο και τελευταίο καλώδιο το SDA pin (4^ο pin) του αισθητήρα και του SDA pin του Arduino μας.

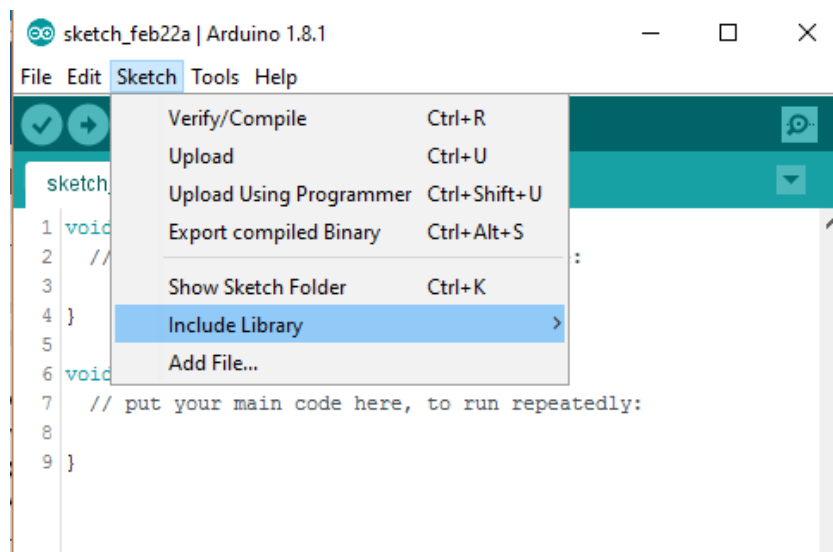


Ελέγχουμε τον κώδικα και τον «ανεβάζουμε» στον Arduino επεξεργαστή

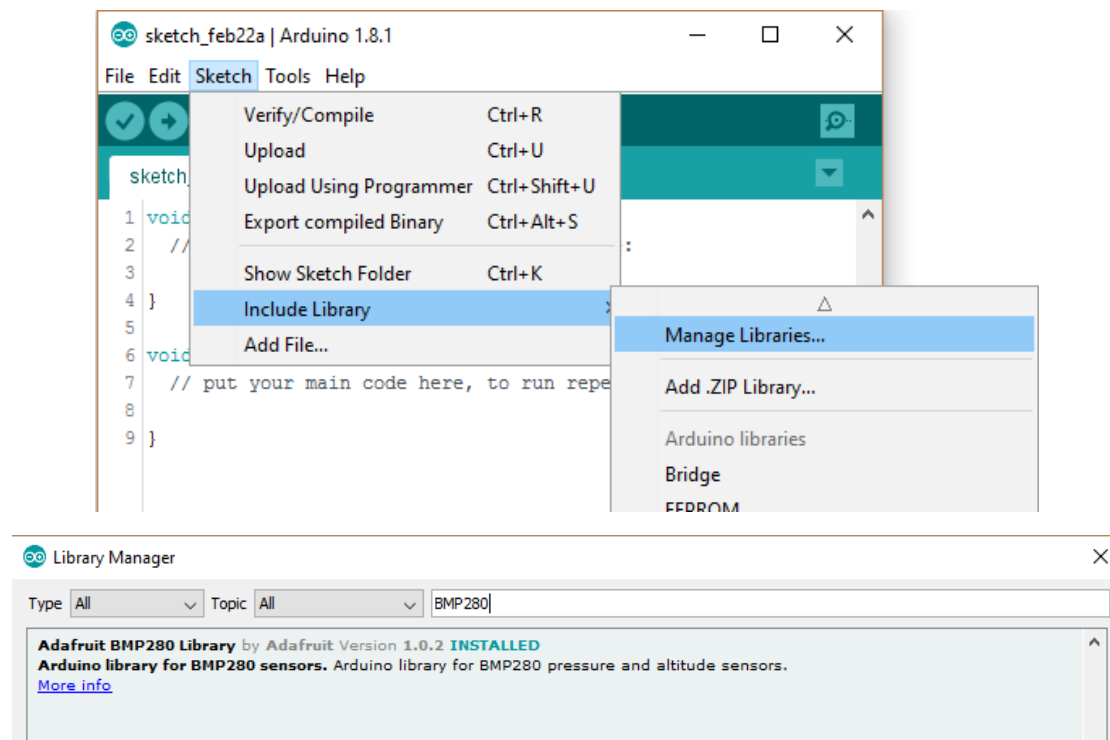
2.5 Κώδικας

Στον κώδικα βλέπουμε πως χρησιμοποιούνται κάποιες βιβλιοθήκες. Οι βιβλιοθήκες αυτές είναι οι κατάλληλες για τον αισθητήρα που χρησιμοποιούμε. Στον κώδικα για τη συγκεκριμένη κατασκευή χρησιμοποιούμε (δηλώσεις `#include`) την βιβλιοθήκη `Wire.h` για το πρωτόκολλο I2C και την βιβλιοθήκη `Adafruit_BMP280.h` για τον αισθητήρα ατμοσφαιρικής πίεσης και θερμοκρασίας.

Οι βιβλιοθήκες που χρειαζόμαστε εισάγονται στο πρόγραμμα με χρήση του menu στο Arduino IDE **Sketch-Include Library**.



Αν η βιβλιοθήκη δεν υπάρχει την αναζητούμε από το Sketch-Include Library-Manage Libraries... και αναζητούμε την βιβλιοθήκη BMP280. Αν δεν είναι εγκατεστημένη την εγκαθιστούμε.



```

#include <Wire.h> // I2C πρωτόκολλο
#include <Adafruit_BMP280.h>

Adafruit_BMP280 bmp; // ορίζεται ο αισθητήρας bmp

void setup() {
  Serial.begin(9600); // εκκινεί την επικοινωνία στα 9600 bytes
  Serial.println(F("BMP280 test")); // τυπώνει ψηφιακά τη φράση "BMP280 test"

  if (!bmp.begin(0x76)) { //δίνει τη διεύθυνση του I2C στην οποία είναι αποθηκευμένος ο αισθητήρας
    Serial.println("Could not find a valid BMP280 sensor, check wiring!");
    while (1);
  }
}

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

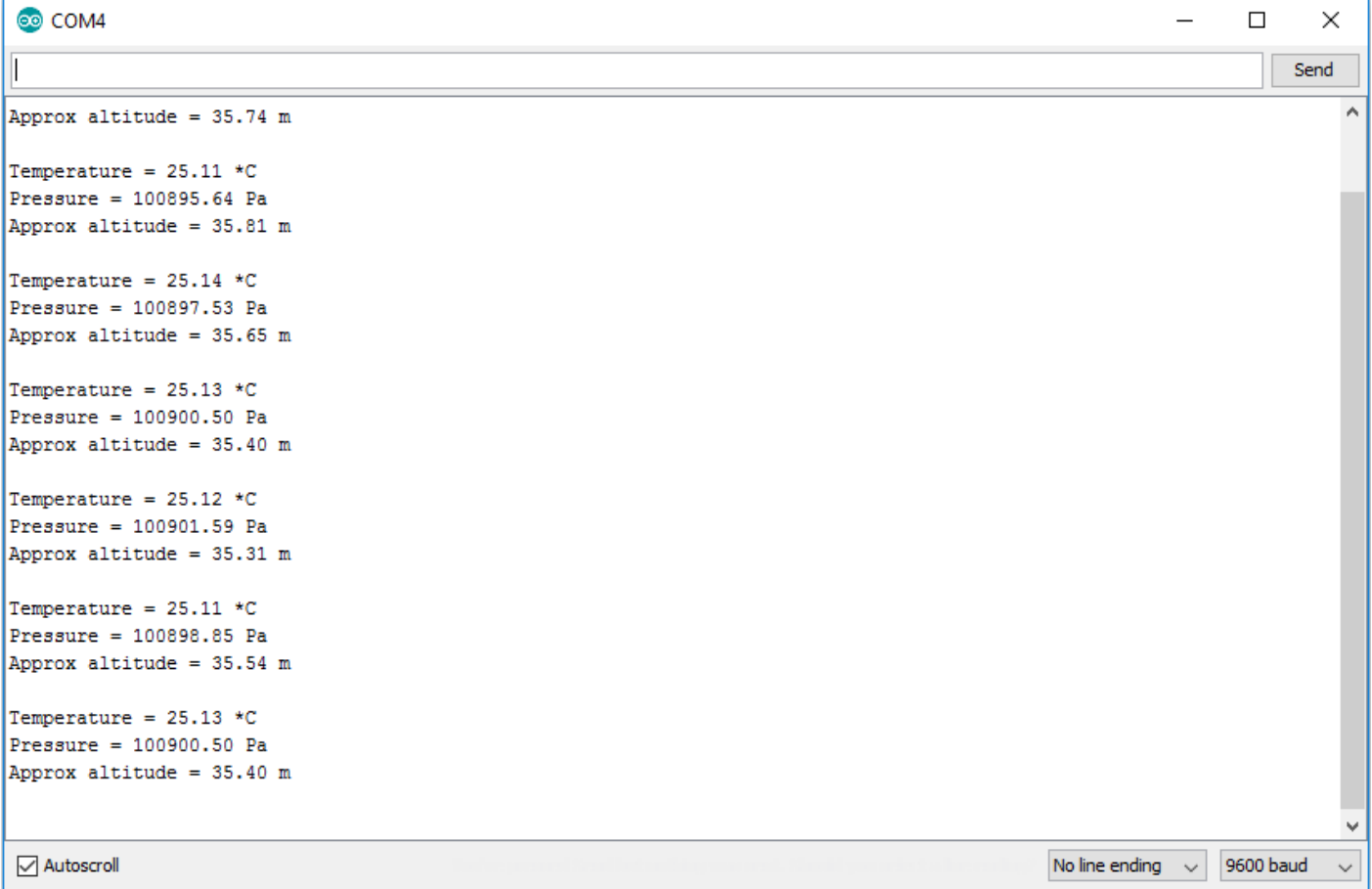
  Serial.print("Pressure = ");
  Serial.print(bmp.readPressure()); // τυπώνεται η πίεση
  Serial.println(" Pa");

  Serial.print("Approx altitude = ");
  Serial.print(bmp.readAltitude(1018.00)); // τυπώνεται το ύψος 1013.25
  Serial.println(" m");

  Serial.println();
  delay(2000);
}

```


Όταν εκτελείται το πρόγραμμα εμφανίζει στο serial monitor τα παρακάτω.



```
COM4

Approx altitude = 35.74 m

Temperature = 25.11 *C
Pressure = 100895.64 Pa
Approx altitude = 35.81 m

Temperature = 25.14 *C
Pressure = 100897.53 Pa
Approx altitude = 35.65 m

Temperature = 25.13 *C
Pressure = 100900.50 Pa
Approx altitude = 35.40 m

Temperature = 25.12 *C
Pressure = 100901.59 Pa
Approx altitude = 35.31 m

Temperature = 25.11 *C
Pressure = 100898.85 Pa
Approx altitude = 35.54 m

Temperature = 25.13 *C
Pressure = 100900.50 Pa
Approx altitude = 35.40 m
```

☒ Autoscroll No line ending 9600 baud

3. Βαρομετρικός-Θερμοκρασιακός Αισθητήρας με OLED οθόνη

3.1 Στόχος

Στην κατασκευή του προηγούμενου παραδείγματος προσθέτουμε μια OLED οθόνη, 25mm x 14mm (0.98in x 0.55in). Με τον τρόπο αυτό θα δούμε πως συνδέονται περισσότεροι αισθητήρες ή συσκευές στο I2C πρωτόκολλο και θα χρησιμοποιήσουμε την οθόνη για απεικόνιση της θερμοκρασίας και της ατμοσφαιρικής πίεσης που θα μας δίνει ο αισθητήρας BMP280.

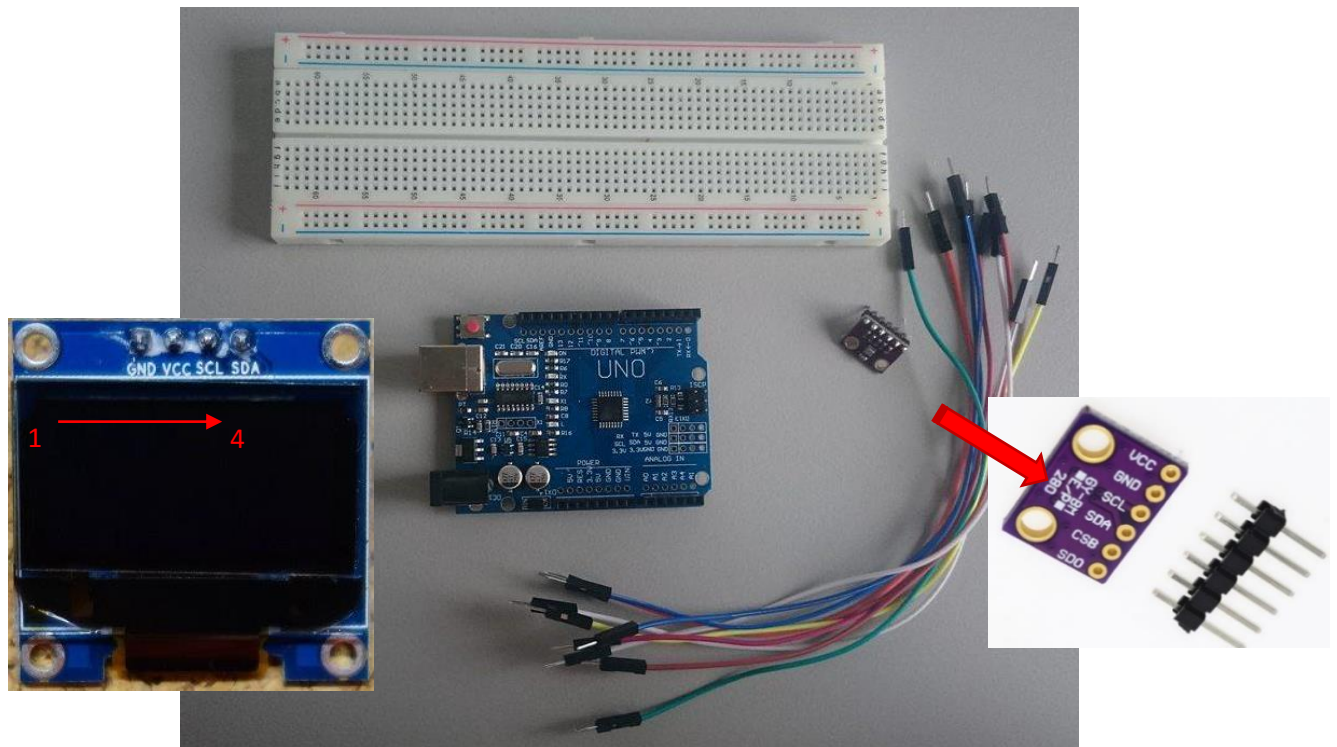
3.2 Μορφή OLED οθόνης

Η οθόνη που χρησιμοποιούμε είναι κατασκευασμένη από 128x64 ατομικά OLED pixels και δεν απαιτείται οπίσθιος φωτισμός. Η οθόνη OLED είναι μονόχρωμη (λευκό χρώμα), αλλά υπάρχουν και άλλα μοντέλα με διάφορα χρώματα, όπως και έγχρωμες. Για την επικοινωνία χρησιμοποιεί το πρωτόκολλο I2C. Αυτό σημαίνει ότι επικοινωνεί με το Arduino, χρησιμοποιώντας μόνο 2 pin, τα SCL & SDA και θα τη συνδέσουμε σε συνέχεια του αισθητήρα BMP280.

Η οθόνη έχει τέσσερα «ποδαράκια». Το 1^ο που είναι το GND συνδέεται με τη γείωση του Arduino, το 2^ο που ονομάζεται VCC συνδέεται με το pin 5V του Arduino, το 3^ο το οποίο λέγεται CSL συνδέεται με το αντίστοιχο του Arduino ενώ το 4^ο το οποίο είναι το SDA συνδέεται με το αντίστοιχο του Arduino.

3.3 Υλικά που χρησιμοποιούνται

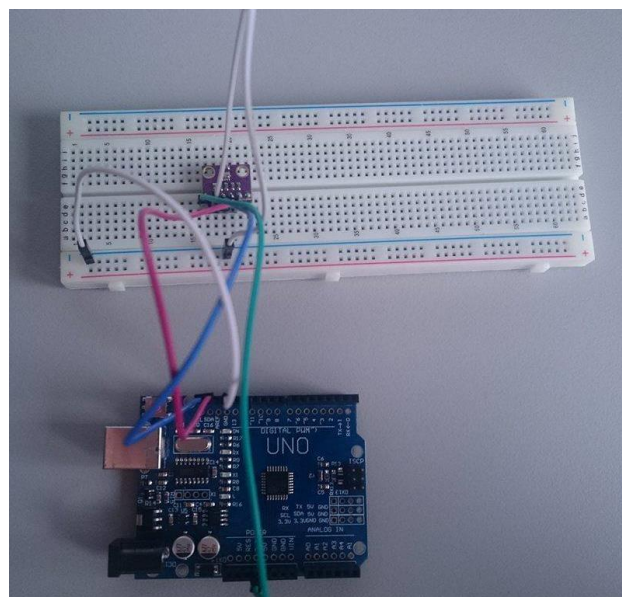
- Arduino Uno board
- Arduino Breadboard
- 9 καλώδια αρσενικό-αρσενικό
- bmp280 αισθητήρας θερμοκρασίας & βαρόμετρο
- SSD1306 OLED οθόνη



3.4 Συνδεσμολογία

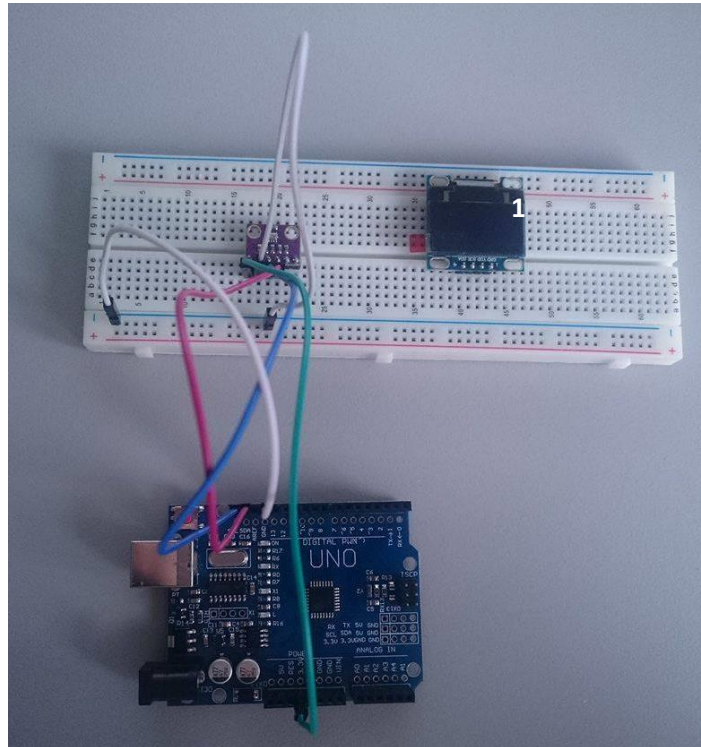
1^ο Βήμα

Αρχικά συνδέουμε τον αισθητήρα μας πάνω στην breadboard και προσέχουμε τα 6 pins να είναι στην ίδια σειρά τοποθετημένα (π.χ. στην ε). Έπειτα, προσθέτουμε τα τέσσερα καλώδια με τη σειρά που ήδη έχει αναφερθεί στο προηγούμενο παράδειγμα (υποκεφάλαιο 2.4).



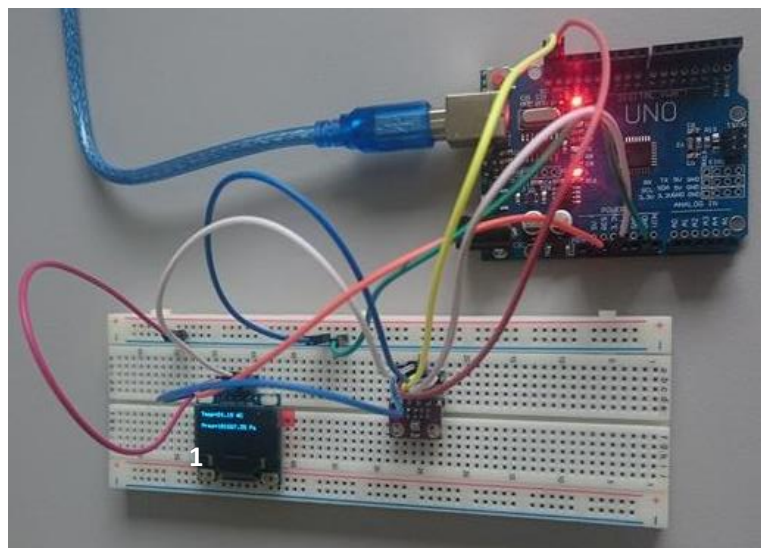
2^ο Βήμα

Συνδέουμε την OLED οθόνη μας. Το πρώτο pin βρίσκεται δεξιά όπως βλέπουμε την παρακάτω φωτογραφία.



3^ο Βήμα

Συνδέουμε με σειρά από το «1», το πρώτο καλώδιο μεταξύ GND OLED οθόνης και γείωσης, το δεύτερο μεταξύ VDD OLED οθόνης και του pin των 5V του Arduino, το τρίτο συνδέεται μεταξύ SCL OLED και SCL του βαρομετρικού αισθητήρα, ενώ το τελευταίο καλώδιο συνδέεται μεταξύ SDA OLED και SDA βαρομετρικού αισθητήρα.



Ελέγχουμε τον κώδικα και τον «ανεβάζουμε» στον Arduino επεξεργαστή.

3.5 Κώδικας

```
#include <Wire.h> //i2c βιβλιοθήκη
#include <Adafruit_BMP280.h> //BMP280 βιβλιοθήκη
#include <Adafruit_SSD1306.h> // lcd βιβλιοθήκη

#define OLED_RESET 4
Adafruit_SSD1306 display(OLED_RESET); //ορισμός του lcd στο I2C πρωτόκολλο

Adafruit_BMP280 bmp; // ορισμός του BMP280 στο I2C πρωτόκολλο

void setup() {
  Serial.begin(9600); // εκκινεί την επικοινωνία στα 9600 bytes
  Serial.println(F("BMP280 and lcd")); // τυπώνει στο σειριακό τη φράση "BMP280 and lcd"

  if (!bmp.begin(0x76)) { //βεβαιώνει τη διεύθυνση του I2C στην οποία βρίσκεται ο αισθητήρας
    Serial.println("Could not find a valid BMP280 sensor, check wiring!");
    while (1);
  }
  display.begin(SSD1306_SWITCHCAPVCC, 0x3C); // εκκίνηση με το πρωτόκολλο I2C στη διεύθυνση 0x3C (για 128x64)
  // init done

  display.setTextSize(1); //μέγεθος χαρακτήρων
  display.setTextColor(WHITE); // χρώμα κειμένου
}

void loop() {
  display.clearDisplay(); // καθαρίζει η οθόνη
  float temp=bmp.readTemperature();
  float pres=bmp.readPressure();

  display.setCursor(0,0); // ξεκινάει το τύπωνμα απο το πάνω μέρος της οθόνης
  display.print("Temp=");
  display.print(temp); // εμφανίζεται στην οθόνη η θερμοκρασία
  display.println(" *C");

  |
  display.setCursor(0,20); //ξεκινάει το τύπωνμα απο πιο κάτω στην οθόνη
  display.print("Pres=");
  display.print(pres); // εμφανίζεται στην οθόνη η πίεση
  display.println(" Pa");
  display.display(); // εμφανίζει ότι έχει εντολή να εμφανίσει

  delay(5000); //καθυστερήση 5 δευτερολέπτων
```

Ως έξοδο έχουμε την εμφάνιση της θερμοκρασίας και της πίεσης στην οθόνη μας.

4. Αισθητήρας Αερίων

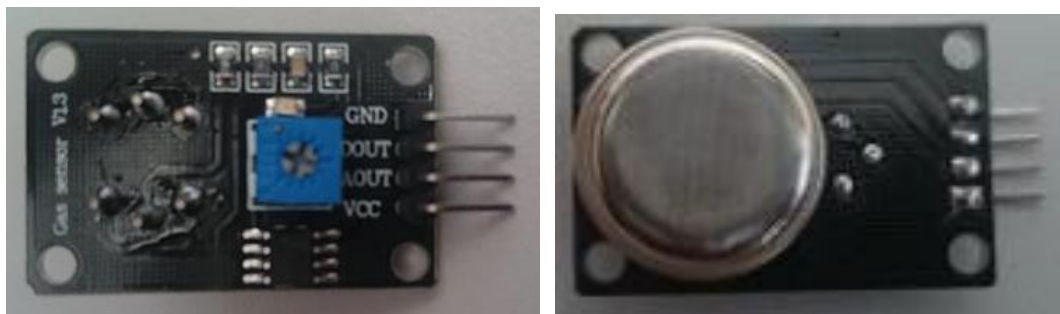
4.1 Περίληψη

Οι αισθητήρες αερίων χρησιμοποιούν ένα μικρό θερμαστή στο εσωτερικό με ένα ηλεκτρο-χημικό αισθητήρα. Είναι ευαίσθητοι για μια σειρά από καύσιμα αέρια και χρησιμοποιούνται σε εσωτερικούς χώρους σε θερμοκρασία δωματίου. Η έξοδος είναι ένα αναλογικό σήμα που μπορεί να διαβαστεί με μια αναλογική είσοδο του Arduino.

4.2 Είδη Αισθητήρων Αερίων

Αισθητήρας MQ-2

Ο MQ-2 αισθητήρας αερίου, χρησιμεύει για την ανίχνευση διαρροής αερίου στο σπίτι και τη βιομηχανία. Μπορεί να ανιχνεύσει υγραέριο, βουτάνιο, προπάνιο, μεθάνιο, αλκοόλ, υδρογόνο και καπνό. Μερικές μονάδες έχουν μια ενσωματωμένη μεταβλητή αντίσταση για να ρυθμίζεται το όριο πάνω από το οποίο θα δώσει ψηφιακό σήμα ο αισθητήρας για συναγερμό ορίου ή ενέργεια από το μικροελεγκτή.



Ο αισθητήρας έχει τέσσερα pins εκ των οποίων στο παράδειγμά μας θα χρησιμοποιήσουμε τα τρία:

- VCC στο pin των 5V του Arduino
- GND στο pin GND του Arduino
- Output στο αναλογικό pin A0 του Arduino
- Σε περίπτωση που χρειαζόμαστε και ειδοποίηση όταν ξεπερνά το όριο που έχει τεθεί με το ροοστάτη πάνω στον αισθητήρα συνδέουμε το D_Output στο pin 8 του Arduino



Ο αισθητήρας ζεσταίνεται μετά από λίγο, μην το αγγίζετε!

Αισθητήρας MQ-7 (CO sensor)

Το μονοξειδίο του άνθρακα (CO) είναι ένα πολύ επικίνδυνο αέριο καθώς είναι άοσμο και άχρωμο. Το CO παράγεται από την μερική οξείδωση του διοξειδίου του άνθρακα, όπως όταν χρησιμοποιούμε μια μηχανή εσωτερικής καύσης σε κλειστό χώρο.

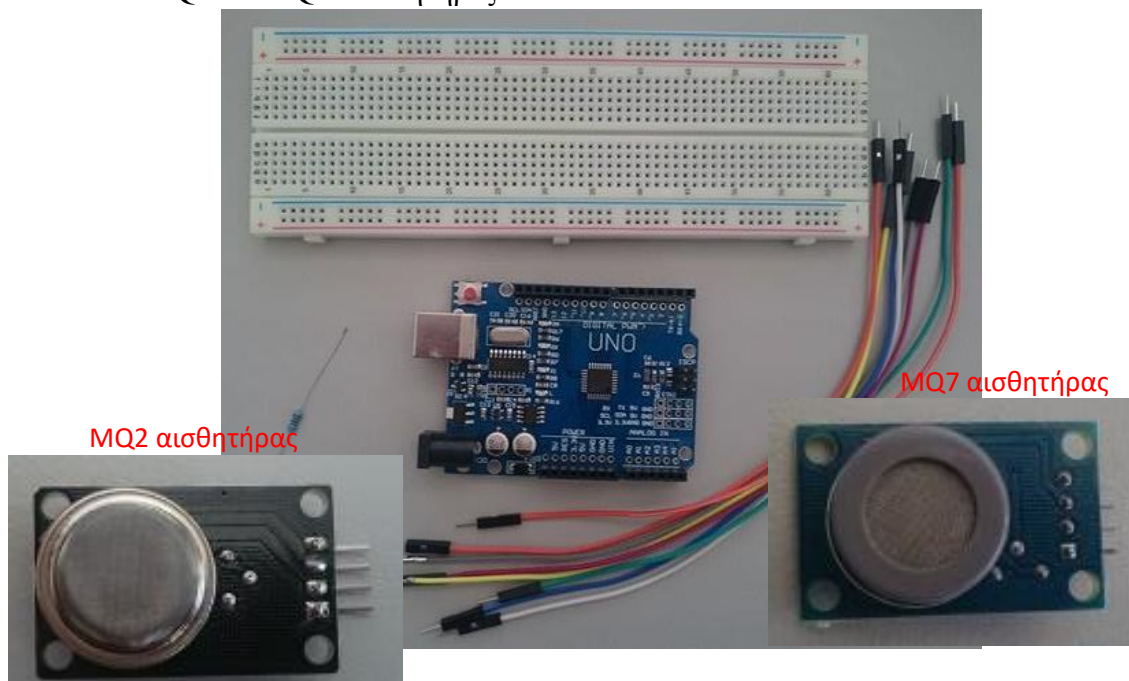


Ο αισθητήρας έχει τέσσερα pins τα οποία χρησιμοποιούνται ως εξής:

- VCC στο pin των 5V του Arduino
- GND στο pin GND του Arduino
- A_Output στο αναλογικό pin A1 του Arduino
- Σε περίπτωση που χρειαζόμαστε και ειδοποίηση όταν ξεπερνά το όριο που έχει τεθεί με το ροοστάτη πάνω στον αισθητήρα συνδέουμε το D_Output στο pin 8 του Arduino

4.3 Υλικά που χρησιμοποιούνται

- Arduino Uno board
- Arduino Breadboard
- 8 καλώδια αρσενικό-αρσενικό
- 1 αντίσταση 5 KΩ
- MQ-2 & MQ-7 αισθητήρες



4.4 Συνδεσμολογία

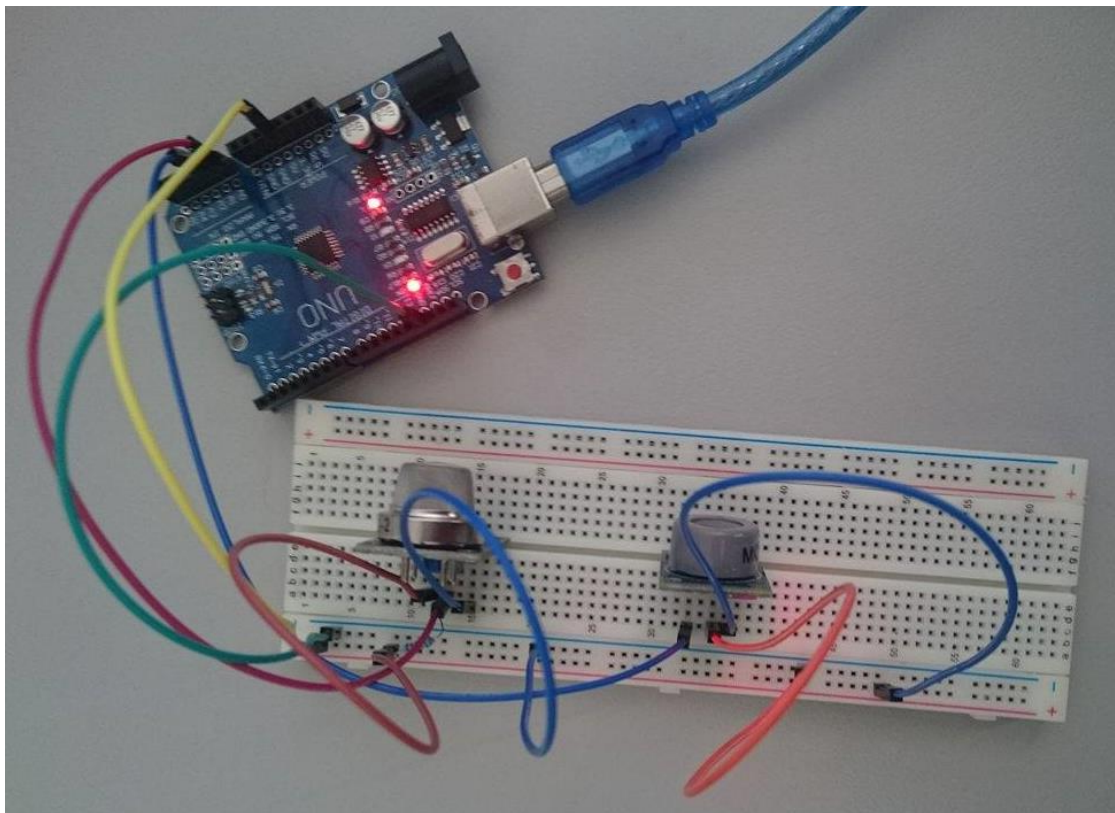
Η συνδεσμολογία που ακολουθείται είναι η εξής:

Ο αισθητήρας MQ-2 έχει τέσσερα pins εκ των οποίων χρησιμοποιούνται τα τρία:

- VCC στο pin των 5V του Arduino
- GND στο pin GND του Arduino
- Output στο αναλογικό pin A0 του Arduino
- Σε περίπτωση που χρειαζόμαστε και ειδοποίηση όταν ξεπερνά το όριο που έχει τεθεί με το ροοστάτη πάνω στον αισθητήρα συνδέουμε το D_Output στο pin 8 του Arduino

Ενώ ο αισθητήρας MQ-7 έχει τέσσερα pins τα οποία χρησιμοποιούνται τα 3 ως εξής:

- VCC στο pin των 5V του Arduino
- GND στο pin GND του Arduino
- A_Output στο αναλογικό pin A1 του Arduino
- Σε περίπτωση που χρειαζόμαστε και ειδοποίηση όταν ξεπερνά το όριο που έχει τεθεί με το ροοστάτη πάνω στον αισθητήρα συνδέουμε το D_Output στο pin 8 του Arduino



4.5 Κώδικας

```
#define SAMPLES 20
#define CO_THRESS 200 // ορίζεται το όριο πάνω από το οποίο θα ειδοποιημαστε για ύπαρξη CO
#define GAS_THRESS 400 // ορίζεται το όριο πάνω από το οποίο θα ειδοποιημαστε για ύπαρξη αερίων

const int gasPin = A0; //ο αισθητήρας GAS συνδέεται στο αναλογικό pin A0
const int COpin=A1;//ο αισθητήρας CO συνδέεται με το αναλογικό pin A1

void setup() {
  Serial.begin(9600); //εκκινείται η επικοινωνία στα 9600 bytes
}

void loop() {
  int COvalue=0; //δίνεται η τιμή 0 στη μέτρηση του CO
  int Gasvalue=0; //δίνεται η τιμή 0 στη μέτρηση του αερίου
  for (int i=0; i<= SAMPLES; i++){
    COvalue += analogRead(COpin); //διαβάζει την αναλογική τιμή από το AOUT pin του αισθητήρα CO
    Gasvalue += analogRead(gasPin); //διαβάζει την αναλογική τιμή από το AOUT pin του αισθητήρα αερίων (gas)
  }
  COvalue=COvalue/SAMPLES; //υπολογίζεται ο μέσος όρος για 10 μετρήσεις CO
  Gasvalue=Gasvalue/SAMPLES; //υπολογίζεται ο μέσος όρος για 10 μετρήσεις αερίου (gas)

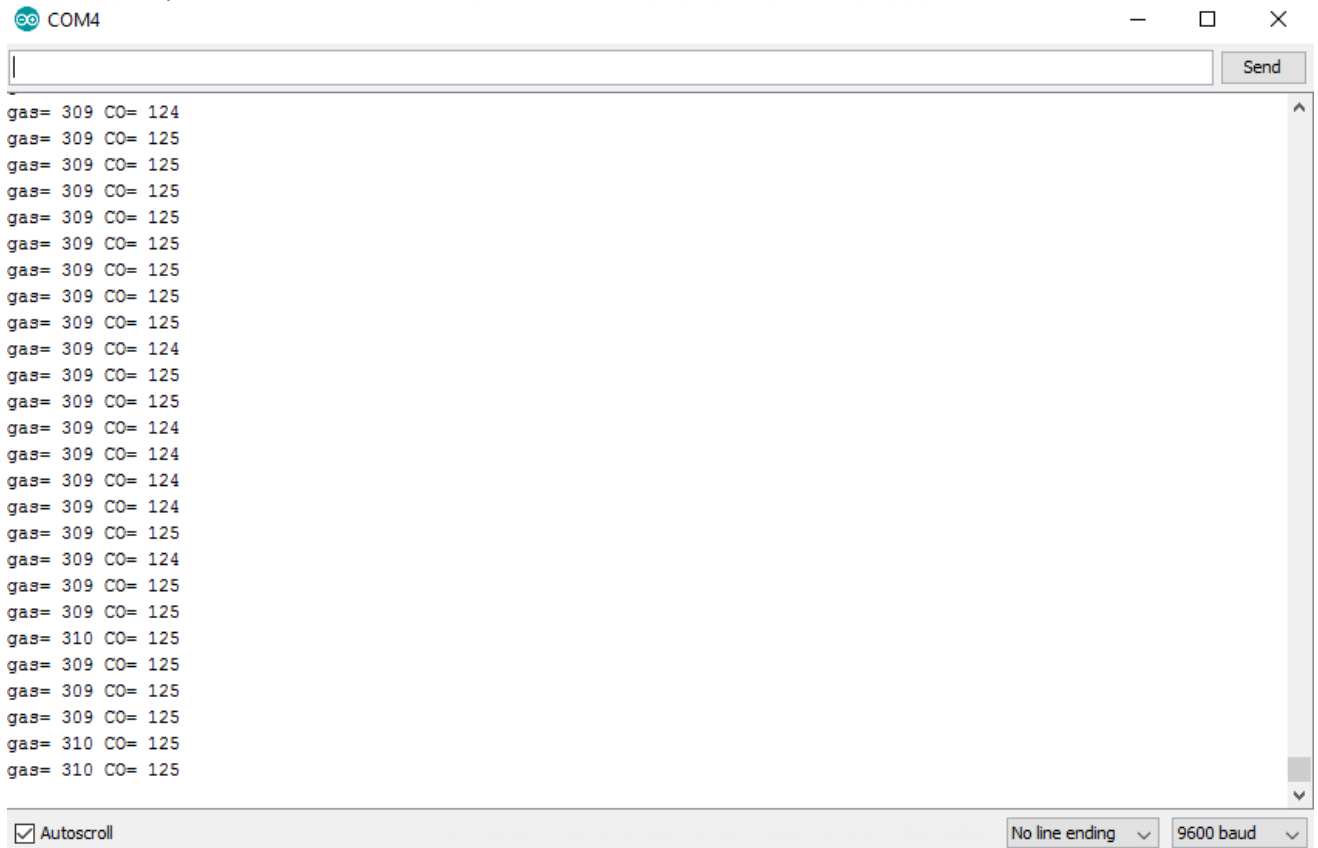
  Serial.print("gas= ");
  Serial.print(Gasvalue); //τυπώνεται η τιμή gas
  Serial.print(" CO= ");
  Serial.println(COvalue); //τυπώνεται η τιμή CO

  delay(1000); //καθυστερεί 1 δευτερόλεπτο
  if(COvalue > CO_THRESS)
    Serial.println("CO exeeded limit"); //εάν η μετρήσιμη τιμή CO ξεπερνάει την κρίσιμη τιμή τότε τυπώνεται η φράση "Το CO ξεπέρασε το όριο"
  if(Gasvalue > GAS_THRESS)
    Serial.println("Gas exeeded limit"); //εάν η μετρήσιμη τιμή gas ξεπερνάει την κρίσιμη τιμή τότε τυπώνεται η φράση "Το αέριο (gas) ξεπέρασε το όριο"
}
```

Όταν δοκιμάσουμε με αναπτήρα τους αισθητήρες έχουμε ενδείξεις ανάλογα με το αν έχουμε διαρροή υγραερίου ή φωτιά όπως φαίνεται στο διάγραμμα παρακάτω.

4.6 Εμφάνιση Αποτελεσμάτων

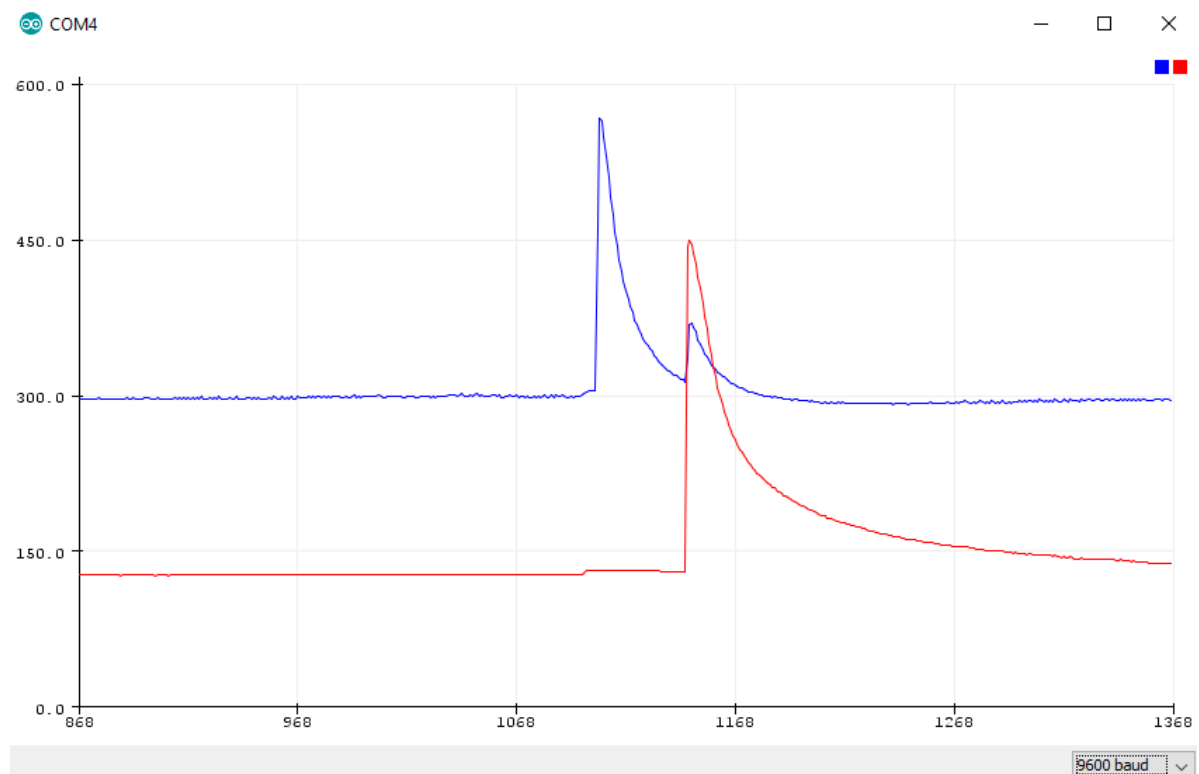
Εάν πατήσουμε tools=> serial monitor (Ctrl + Shift +M) η έξοδος είναι η εξής:



```
gas= 309 CO= 124
gas= 309 CO= 125
gas= 309 CO= 125
gas= 309 CO= 125
gas= 309 CO= 125
gas= 309 CO= 125
gas= 309 CO= 125
gas= 309 CO= 125
gas= 309 CO= 125
gas= 309 CO= 124
gas= 309 CO= 125
gas= 309 CO= 125
gas= 309 CO= 124
gas= 309 CO= 124
gas= 309 CO= 124
gas= 309 CO= 124
gas= 309 CO= 125
gas= 309 CO= 125
gas= 309 CO= 125
gas= 310 CO= 125
gas= 309 CO= 125
gas= 309 CO= 125
gas= 309 CO= 125
gas= 310 CO= 125
gas= 310 CO= 125
```

☒ Autoscroll No line ending 9600 baud

Ενώ εάν πατήσουμε tools=> serial plotter (Ctrl + Shift +L) η έξοδος είναι ένα διάγραμμα:



5. Αισθητήρας pH

5.1 Στόχος

Η κατασκευή αυτή έχει σκοπό να φτιάξουμε ένα αισθητήρας μέτρησης pH.

Ουσιαστικά είναι μια απλή μονοκύτταρη μπαταρία με πολύ υψηλή αντίσταση, όπου παράγεται τάση ανάλογη της συγκέντρωσης των ιόντων υδρογόνου που υπάρχει μέσα στον ανιχνευτή. Επομένως, όταν η συγκέντρωση είναι μεγαλύτερη σε κάποια πλευρά του ανιχνευτή (pH μέτρου), τότε η ροή των ιόντων προκαλεί μια ελαφριά τάση που ταλαντεύεται μεταξύ +/- και δείχνει αν έχουμε οξύ ή βάση.

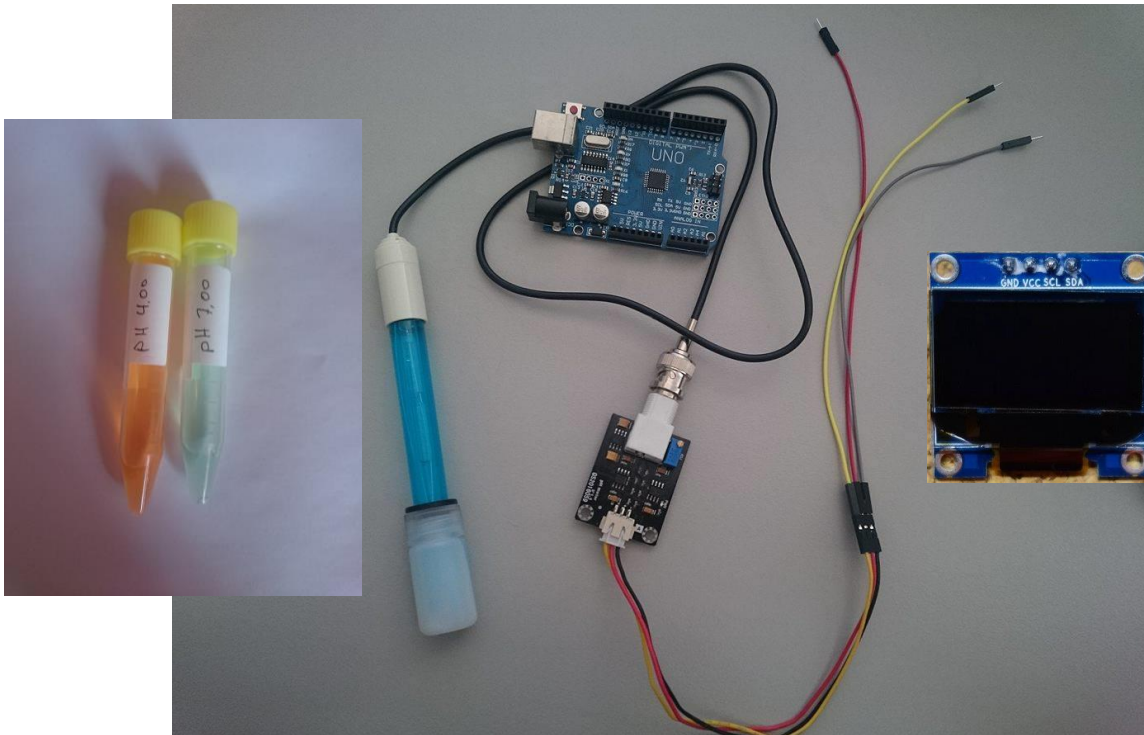
Η δημιουργία ενός ανιχνευτή pH αποτελείται από 2 ηλεκτρόδια, ένα σε διάλυμα KCl και το άλλο σε χαμηλή συγκέντρωση HCl. Το κέντρο "αίσθησης" του ηλεκτροδίου (άνοδος) περιβάλλεται από ένα ειδικό βολβό γυαλιού που επιτρέπει τα ιόντα να περάσουν, το άλλο ηλεκτρόδιο (κάθοδος) είναι σφραγισμένο από το ηλεκτρόδιο "αίσθησης" και θεωρείται το ηλεκτρόδιο αναφοράς. Αυτό στη συνέχεια συνδέεται με ένα πορώδες κεραμικό βύσμα (ή ίνες χαλαζία) και σχηματίζει μια γέφυρα άλατος με το διάλυμα δοκιμής. Αυτή η γέφυρα αποτελεί ένα γαλβανικό στοιχείο και δημιουργεί την απλή μπαταρία. Το δυναμικό μεταξύ του ηλεκτροδίου στη συνέχεια, μας δίνει το pH του διαλύματος δοκιμής.

5.2 Χαρακτηριστικά αισθητήρα

- Λειτουργία στα 5V
- Μέγεθος: 43mm x 32mm
- Εύρος μετρήσεων: 0-14 pH
- Εύρος θερμοκρασιακών μετρήσεων: 0-60 °C
- Ακρίβεια: ± 0.1 pH (25 °C)
- Χρόνος απόκρισης: ≤ 1 λεπτό

5.3 Υλικά που χρησιμοποιήθηκαν

- Arduino Uno board
- 6 καλώδια αρσενικό-αρσενικό
- μετρητής pH
- οθόνη OLED
- υγρά καλιμπραρίσματος (4 & 7 pH)



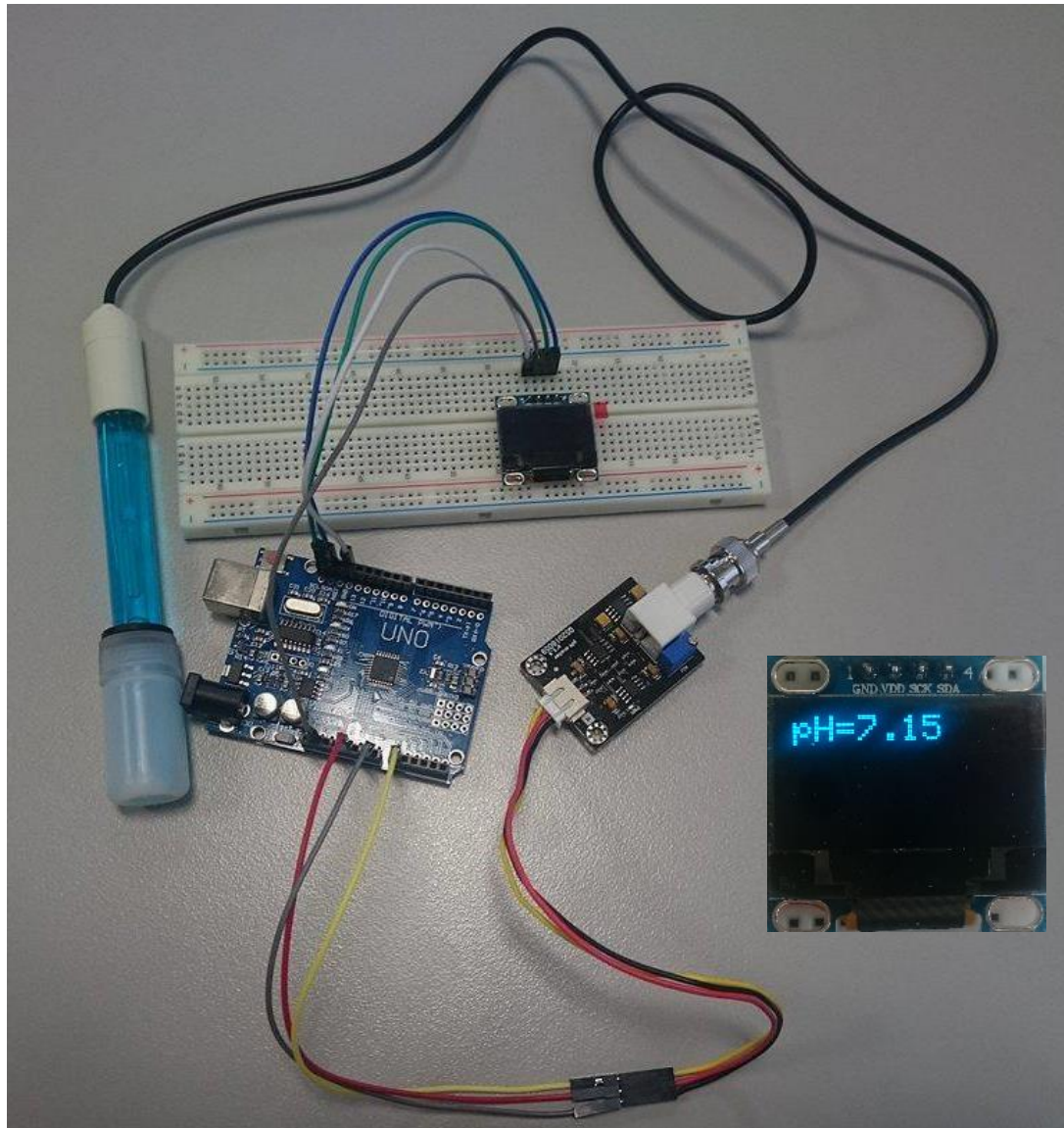
5.4 Συνδεσμολογία

Βήμα 1°

Το πρώτο καλώδιο (μαύρο) του μετρητή pH συνδέεται μεταξύ της GND(-) του pHμετρου και της γείωσης του Arduino, το δεύτερο καλώδιο (κόκκινο) μεταξύ VDD οθόνης και του pin των 5V του Arduino, ενώ το τρίτο (κίτρινο) συνδέει το A_output του μετρητή pH με το αναλογικό pin A0.

Βήμα 2°

Συνδέουμε με σειρά, το πρώτο καλώδιο (λευκό) μεταξύ GND OLED οθόνης και γείωσης, το δεύτερο (γκρι) μεταξύ VDD OLED οθόνης και του pin των 5V του Arduino, το τρίτο (πράσινο) συνδέεται μεταξύ SCL OLED και SCL Arduino, ενώ το τελευταίο καλώδιο (μπλε) συνδέεται μεταξύ SDA OLED και SDA Arduino.



5.5 Καλιμπράρισμα pHμετρου

Θα χρειαστούμε δύο ρυθμιστικά διαλύματα. Ένα με pH 7 κι ένα με pH 4. Για την ρύθμιση χρειαζόμαστε να ανοίξουμε την σειριακή πόρτα και να απαντήσουμε στην ερώτηση αν θέλουμε να καλιμπράρουμε τη διάταξη (1-Yes) ή όχι (0-No).

Πατάμε 1 και Send για έναρξη καλιμπραρίσματος.

- Ξεπλένουμε το μετρητή pH με απεσταγμένο ή απιονισμένο νερό.
- Στεγνώνουμε το μετρητή pH , ειδικά την άκρη, με μια καθαρή χαρτοπετσέτα.
- Τοποθετούμε το μετρητή pH στο ρυθμιστικό διάλυμα με pH 7 με το άκρο του να βυθίζεται σε αυτό.
- Όταν η τιμή σταθεροποιηθεί, σημειώνουμε την τιμή που μας δείχνει στο serial monitor για να την βάλουμε στη θέση float K=3.89 του κώδικα.

- Ξεπλένουμε και στεγνώνουμε και πάλι τον ανιχνευτή.
- Τοποθετούμε τον ανιχνευτή στο ρυθμιστικό διάλυμα με pH 4 και περιμένουμε 2 λεπτά για σταθεροποίηση.
- Στέλνουμε στη σειριακή θύρα ένα χαρακτήρα για να προχωρήσει στο επόμενο στάδιο που μας δείχνει συνεχώς την μετρούμενη τιμή.
- Ρυθμίζουμε το ποτενσιόμετρο με μικρές κινήσεις ώστε να δείξει 4.00 στο serial monitor.
- «Ανεβάζουμε» τον κώδικα με τις δικές μας τιμές στο Arduino.



Βεβαιωνόμαστε ότι έχουμε ξεπλύνει και στεγνώσει τον ανιχνευτή μετά από κάθε ανάγνωση

5.6 Κώδικας

Το Arduino διαβάζει στην αναλογική είσοδο A0 την τάση που δίνει ο μετρητής pH και την κατανέμει σε 1024 βαθμίδες.

Όταν μετράμε το ρυθμιστικό διάλυμα με pH 7, τότε η τιμή Cal_pH στον κώδικα είναι ίση με 7.

Ανάλογα με το μήκος των καλωδίων που χρησιμοποιούμε, το ρεύμα που τροφοδοτείται το Arduino, καθώς και την παλαιότητα του αισθητήρα υπάρχει μία μικρή απόκλιση στη μέτρηση και μπορεί η ένδειξη να μην είναι 7 ακριβώς. Για το λόγο αυτό κάνουμε το καλιμπράρισμα της διάταξης και το πρόγραμμα μας προτείνει να αλλάξουμε την σταθερά K με την προτεινόμενη, ώστε η μέτρηση να είναι ακριβώς 7. Η σταθερά αυτή προκύπτει από την απόκλιση των x mV που μετρά ο αισθητήρας ως προς την τιμή των mV για την τιμή 7. Έτσι με τη απλή μέθοδο στην οποία τα 5V είναι στην βαθμίδα 1024, υπολογίζουμε το συντελεστή K ως $1024x/5$.

Κατόπιν με το ρυθμιστικό διάλυμα pH 4 ρυθμίζουμε την ενίσχυση του σήματος ώστε να είναι ακριβώς η ένδειξη που βλέπουμε στην οθόνη όσο το ρυθμιστικό μας διάλυμα, δηλαδή 4.

Μετά το καλιμπράρισμα στην κανονική λειτουργία της διάταξης χρησιμοποιείται ο συντελεστής διόρθωσης που έχουμε βάλει στο πρόγραμμα και έχουμε ανεβάσει στο Arduino, ώστε να γίνονται ακριβείς μετρήσεις.


```

1 #include <Wire.h> //i2c βιβλιοθήκη
2 #include <Adafruit_SSD1306.h> // lcd βιβλιοθήκη για το SSD1306
3 #define OLED_RESET 4
4 #define SAMPLES 20.0 //Αριθμός μετρήσεων για υπολογισμό μέσου όρου
5 Adafruit_SSD1306 display(OLED_RESET); //ορισμός του lcd
6
7 int pHPin =A0; // ο μετρητής pH συνδέεται με το αναλογικό pin του Arduino A0
8 float K=3.89; //Ορίζεται η σταθερά διόρθωσης καλιμπραρίσματος για το μετρητή pH
9 int Cal_pH=7; //το pH στο οποίο καλιμπράρουμε αρχικά
10
11
12 //***** Μεταβλητές για τη χρήση του μετρητή pH *****/
13
14 float pHValue=0;
15 float average=0;
16
17 //*****Διαβάζονται οι τιμές pH και υπολογίζεται ο μέσος όρος *****/
18 void ReadpH(){
19     int i=0;
20     long reading=0;
21     long sum=0;
22
23     for(i=0;i<SAMPLES;i++){
24         reading=analogRead(pHPin);
25         sum=sum+reading;
26         delay(20);
27     }
28     average=sum/SAMPLES;
29     // μετατρέπεται ο μέσος όρος του pH από mv σε ph
30     pHValue=average*K*5/1024;
31 }
32
33
34 //*****Χρήσιμες οδηγίες για το καλιμπράρισμα *****/
35 void usinginfo(){
36     Serial.println(F("pH Probe Script for arduino"));
37     Serial.println(F("How to Use:"));
38     Serial.println(F("1:Place Probe into pH7 calibration fluid, open serial "));
39     Serial.println(F("2:Take Recomened cell constand and change it in the top of code"));
40     Serial.println(F("3:Rinse Probe and place in pH4 calibration fluid"));
41     Serial.println(F("4:Adjust potentiometer on pH meter shield until ph reading in serial is 4"));
42     Serial.println(F(" "));
43     Serial.println(F("Thats it your calibrated and your readings are accurate!"));
44 }
45
46 //***** Υπολογισμός σταθεράς K *****/
47 void Calibrate(){
48     float Kn=0;
49     ReadpH(); //διαβάζονται οι τιμές του pH
50     Kn=((Cal_pH*1024)/(average*5));
51     Serial.println(F("1:Place Probe into pH7 calibration fluid, open serial "));
52     Serial.print(F("2:Change K constand in the top of code with = "));
53     Serial.println(Kn);
54     Serial.println(F("Then change calibration fluid to pH4 and send a key when ready"));

```

```

55 do{
56   } while(Serial.available()<=0);
57   Serial.read();
58   Serial.println(F("Adjust potentiometer to display 4pH.));
59   Serial.println(F("Send a key when done"));
60
61 do{
62   ReadpH();
63   Serial.println(pHValue);
64   } while(Serial.available()<=0); // επαναλαμβάνεται μέχρι να σταλεί ένας χαρακτήρας στη σειριακή θύρα
65   Serial.println(F("End of setup"));
66 }
67
68 void setup() {
69   char calibration=0;
70   Serial.begin(9600); // εκκινεί την επικοινωνία στα 9600 bytes
71
72   // εκκινεί την επικοινωνία μεταξύ πρωτοκόλλου I2C με τη διεύθυνση 0x3C (για οθόνη 128x64)
73
74   display.begin(SSD1306_SWITCHCAPVCC, 0x3C);
75   display.clearDisplay(); // "καθαρίζει" η οθόνη
76   Serial.println(F(" LCD ready" )); // τυπώνει στο serial "LCD ready"
77   display.setTextSize(1); //Μέγεθος χαρακτήρων στο LCD
78   display.setTextColor(WHITE); // χρώμα χαρακτήρων οθόνης
79   display.setCursor(0,0);// τυπώνει στο πάνω μέρος της οθόνης
80   display.println("LCD ready.");
81   display.setCursor(0,20); // τυπώνει 20 Pixels πιο κάτω στην οθόνη
82   display.println("To calibrate open serial monitor");
83   display.display(); // εμφανίζεται στην οθόνη ότι εμπεριέχεται μέσα στη μνήμη του display
84
85   if(Serial){
86     Serial.println(F("Calibrate pH meter? 0-for No 1-for Yes"));
87     do{
88       } while(Serial.available()<=0); //επαναλαμβάνεται μέχρι να σταλεί χαρακτήρας στη σειριακή θύρα
89     calibration=Serial.read();
90     if (calibration=='1') { // ΠΡΟΣΟΧΗ: το 1 μπαίνει σε μονά εισαγωγικά '1' επειδή είναι χαρακτήρας
91       Serial.println(F("Performing calibration..."));
92       usinginfo(); // καλείται η συνάρτηση usinginfo
93       Calibrate(); // καλείται η συνάρτηση Calibrate
94     }
95     else Serial.println(F("End of setup"));
96   }
97   display.setTextSize(2); //Μέγεθος χαρακτήρων
98 }
99
100 void loop() {
101   ReadpH(); //διαβάζεται το pH
102   display.clearDisplay(); // καθαρίζει η οθόνη
103   display.setCursor(0,0);
104   display.print("pH=");
105   display.println(pHValue); //τυπώνει την τιμή του pH στην οθόνη
106   display.display();
107   delay(2000);
108 }

```

