Σχολική Μονάδα: 1ο ΕΠΑΛ Αγρινίου

Συμμετέχοντες Καθηγητές

ΔΗΜΑΚΗΣ ΦΩΤΙΟΣ
ΤΣΟΥΓΚΡΑΝΗΣ ΙΩΑΝΝΗΣ
ΤΣΑΦΑΣ ΑΠΟΣΤΟΛΟΣ
ΣΤΡΑΤΟΠΟΥΛΟΣ ΓΕΩΡΓΙΟΣ
ΜΠΟΛΙΑΡΗΣ ΠΡΟΚΟΠΗΣ

ΣΧΕΔΙΟ ΜΑΘΗΜΑΤΟΣ

ΜΕΛΕΤΗ ΤΟΥ ΨΗΦΙΑΚΟΥ ΒΑΡΟΜΕΤΡΙΚΟΥ ΑΙΣΘΗΤΗΡΑ ΠΙΕΣΗΣ BMP180
ΧΡΗΣΗ ΤΗΣ ΒΙΒΛΙΟΘΗΚΗΣ ΤΟΥ ΣΤΟ ARDUINO
ΕΦΑΡΜΟΓΗ ΣΤΗΝ ΚΑΤΑΣΚΕΥΗ ΒΑΡΙΟΜΕΤΡΟΥ

ΣΧΟΛΕΙΟ :	1ο ΕΠΑ.Λ. ΑΓΡΙΝΙΟΥ
ΜΑΘΗΜΑ:	ΗΛΕΚΤΡΟΝΙΚΕΣ ΔΙΑΤΑΞΕΙΣ (ΕΡΓΑΣΤΗΡΙΟ)
ΤΑΞΗ:	$B_{ΗΛΕΚΤΡΟΝΙΚΩΝ}$
ΔΙΑΡΚΕΙΑ :	2 ΩΡΕΣ
ΚΑΘΗΓΗΤΕΣ :	ΔΗΜΑΚΗΣ ΦΩΤΙΟΣ
	ΤΣΟΥΓΚΡΑΝΗΣ ΙΩΑΝΝΗΣ
	ΤΣΑΦΑΣ ΑΠΟΣΤΟΛΟΣ
	ΣΤΡΑΤΟΠΟΥΛΟΣ ΓΕΩΡΓΙΟΣ
	ΜΠΟΛΙΑΡΗΣ ΠΡΟΚΟΠΗΣ

1. Σκοποί :

- Να μελετήσουν τον αισθητήρα πίεσης BMP180 και τις εφαρμογές του με χρήση μικροεπεξεργαστή Arduino Uno.
- Να κατασκευάσουν διάταξη που να μετρά την ταχύτητα ανόδου – καθόδου από τις μεταβολές της ατμοσφαιρικής πίεσης (Βαριόμετρο).
- Να συνδυάζουν διαφορετικούς τρόπους προσέγγισης της γνώσης.

2. Διδακτικοί στόχοι :

Α. Γνωστικοί

1. Να κατανοήσουν οι μαθητές την σημασία των αισθητήρων (sensors) στην λήψη μετρήσεων
2. Να κατανοήσουν οι μαθητές την έννοια του θορύβου στις μετρήσεις των αισθητήρων (sensors) και τους τρόπους επεξεργασίας των μετρήσεων για την απαλλαγή από τον θόρυβο αυτό
3. Να μπορούν οι μαθητές να «διαβάζουν» δεδομένα από ένα ψηφιακό αισθητήρα με χρήση μικροεπεξεργαστή Arduino Uno δημιουργώντας το κατάλληλο κύκλωμα
4. Να κάνουν χρήση της γλώσσας προγραμματισμού του Arduino και να εφαρμόσουν γνώσεις που έχουν αποκτήσει από τον προγραμματισμό σε προηγούμενα μαθήματα.
5. Να μπορούν να εκτιμήσουν οι μαθητές τη σημασία οποιασδήποτε αλλαγής στον κώδικα του προγράμματος
6. Μέσω της βιωματικής μάθησης, οι μαθητές να μπορούν να κάνουν δοκιμές και

κατάλληλες προσαρμογές στον κώδικα του προγράμματος ώστε να πάρουν την επιθυμητή λειτουργία

7. Να συγκρίνουν τα χαρακτηριστικά , τις επιδόσεις και το κόστος της κατασκευής τους με μια αντίστοιχη συσκευή του εμπορίου.

Β. Εκπαιδευτικοί

1. Να συμμετέχουν ενεργά και να συνεργάζονται ομαδικά στην υλοποίηση της μαθησιακής διαδικασίας
2. Οι μαθητές να πειραματιστούν και να καταλήξουν σε συμπεράσματα μέσα από εποικοδομητικό διάλογο και να επιλέξουν την κατάλληλη λύση για την κατασκευή τους
3. Οι μαθητές να εξοικειωθούν με την ανάπτυξη στρατηγικών επίλυσης προβλημάτων

3. Μέθοδος Διδασκαλίας :

Ομαδοσυνεργατική διδασκαλία με οργανωμένη και φθίνουσα καθοδήγηση, σύμφωνα με την οποία οι μαθητές χωρίζονται σε ομάδες , τα μέλη των οποίων επιλέγονται από τον καθηγητή. Με αυτόν τον τρόπο θα εξασφαλιστεί η ενεργός συμμετοχή από την πλειοψηφία των μαθητών. Σε κάθε ομάδα ανατίθεται ένα τμήμα του έργου. Στην συνέχεια οι ομάδες θα συνεργαστούν για την τελική υλοποίηση της κατασκευής.

4. Διδακτική προσέγγιση :

- Οι μαθητές θα αναζητήσουν από το Διαδίκτυο το datasheet του αισθητήρα BMP180 και πληροφορίες σχετικά με την χρήση του και τα χαρακτηριστικά του.
- Θα αναζητήσουν επίσης πληροφορίες για το άρθρωμα που θα συνδεθεί με το ARDUINO, την βιβλιοθήκη που θα χρησιμοποιήσουν στον προγραμματισμό του μικροελεγκτή , τις διάφορες εντολές της βιβλιοθήκης αυτής καθώς και τι είδους μετρήσεις μπορούμε να πάρουμε
- Θα κατασκευάσουν το κύκλωμα συνδέοντας στο ARDUINO κατάλληλα το άρθρωμα του BMP180 και το Buzzer για την ηχητική σήμανση του Βαριομέτρου
- Θα προγραμματίσουν το παραπάνω κύκλωμα
- Θα ελέγξουν την συμπεριφορά του κυκλώματος αλλάζοντας τις παραμέτρους του κώδικα
- Θα μελετήσουν την επίδραση του θορύβου στις μετρήσεις ταχύτητας και την εφαρμογή κατάλληλων φίλτρων
- Θα αναζητήσουν πληροφορίες για εμπορικές συσκευές Βαριομέτρων , την χρήση τους τις επιδόσεις , τα χαρακτηριστικά και το κόστος
- Θα συγκρίνουν τις επιδόσεις των εμπορικών Βαριομέτρων με την κατασκευή προτείνοντας και δοκιμάζοντας κατάλληλες βελτιώσεις στο κύκλωμα και τον κώδικα

5. Μέσα διδασκαλίας – Υλικά :

- Οι υπολογιστές του εργαστηρίου στους οποίους έχει εγκατασταθεί η πλατφόρμα του ARDUINO με σύνδεση στο Διαδίκτυο
- Ένας μικροεπεξεργαστής Arduino Uno
- Ένα άρθρωμα με το αισθητήριο BMP180 κατάλληλο για σύνδεση με το Arduino , μέσω του διαύλου I^2C
- Ένα Buzzer για την ηχητική σήμανση
- Κατάλληλα καλώδια σύνδεσης
- Ο βιντεοπροβολέας του εργαστηρίου για προβολή διαφανειών από αρχείο power point

Ως προαπαιτούμενες γνώσεις θεωρούνται:

- Ο τρόπος προγραμματισμού του Arduino
- Η χρήση και εγκατάσταση βιβλιοθηκών στο Arduino
- Η χρήση του διαύλου I^2C

6. Σύντομη περιγραφή :

Οι μαθητές, χρησιμοποιώντας τον μικροελεγκτή Arduino Uno, θα μελετήσουν τις δυνατότητες του αισθητήρα BMP180 και θα κατασκευάσουν ένα βαριόμετρο. Μια συσκευή που μετρά ταχύτητες ανόδου – καθόδου από τις μεταβολές της ατμοσφαιρικής πίεσης. Θα χρησιμοποιήσουν σχέδιο κυκλώματος και πρόγραμμα που θα τους δοθεί. Θα ελέγξουν την λειτουργία της συσκευής βελτιώνοντάς την ως προς τον θόρυβο των μετρήσεων και την ακρίβειά της, πειραματιζόμενοι με το κύκλωμα και τον κώδικα που τους δόθηκε. Στο τέλος θα συγκρίνουν την κατασκευή τους με τις αντίστοιχες εμπορικές.

7. Δομή μαθήματος :

- **Αναλυτική περιγραφή επιμέρους βημάτων διδασκαλίας :**

  - **Παρουσίαση από τον καθηγητή**

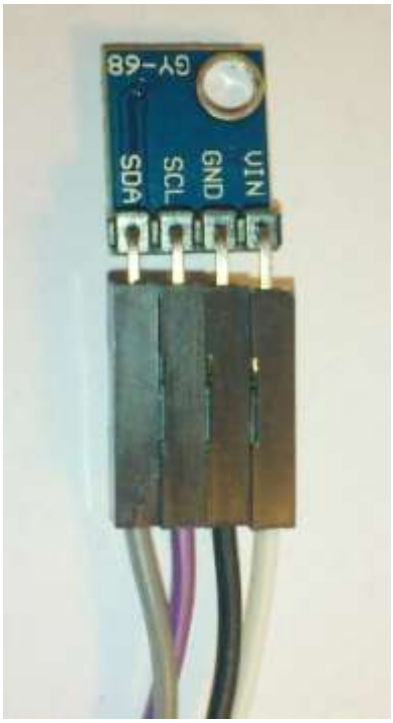
Ο καθηγητής, με την χρήση διαφανειών, θα κάνει πρώτα μια μικρή εισαγωγή για τον αισθητήρα BMP180 , τις δυνατότητές του και τους περιορισμούς του και θα παρουσιάσει την λειτουργία ενός βαριομέτρου. Θα δώσει στους μαθητές πληροφορίες για την αναζήτηση της αντίστοιχης βιβλιοθήκης στο Διαδίκτυο και θα παρουσιάσει τις εντολές της. Θα δώσει και θα εξηγήσει εν συντομία στους μαθητές σχέδιο κυκλώματος που θα υλοποιήσουν καθώς και κώδικα τον οποίο θα φορτώσουν στον μικροελεγκτή. Θα αναφερθεί τέλος εν συντομία στον θόρυβο που επιβαρύνει τις μετρήσεις και την αντιμετώπισή του.

- **Κατασκευή του κυκλώματος**

Η κατασκευή θα ξεκινήσει ετοιμάζοντας το BMP180 για σύνδεση με την πλακέτα του Arduino Uno. Το BMP180 επικοινωνεί με το Arduino μέσω του διαύλου I^2C. Οι μαθητές θα κόψουν τους ακροδέκτες σύνδεσης (pin headers) και θα τους κολλήσουν στις αντίστοιχες υποδοχές του αρθρώματος.

pin headers

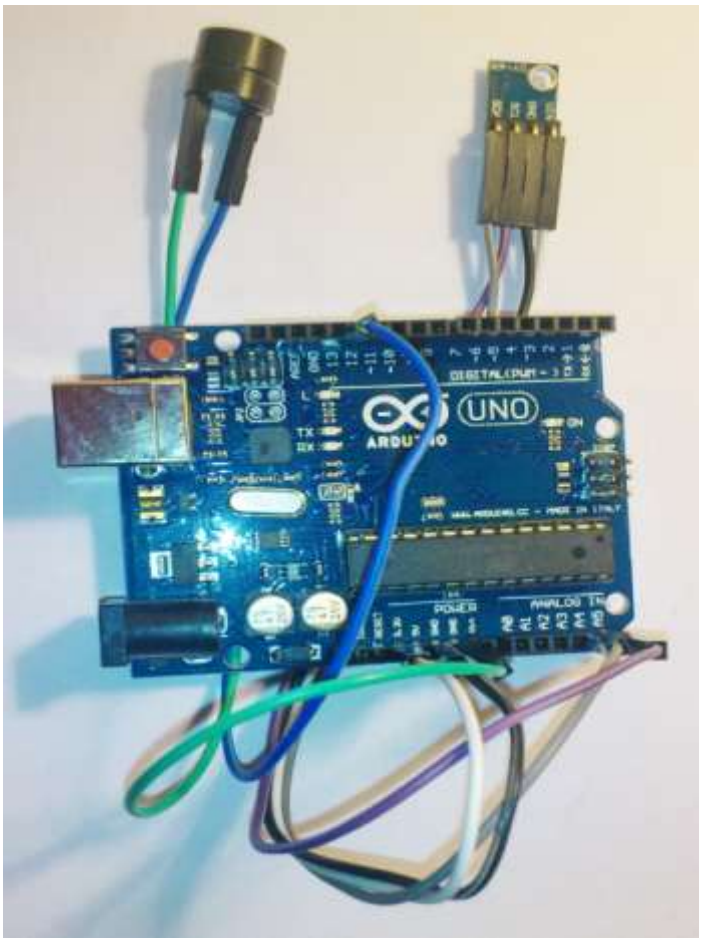
BMP180



Το BMP180 με
τους ακροδέκτες του

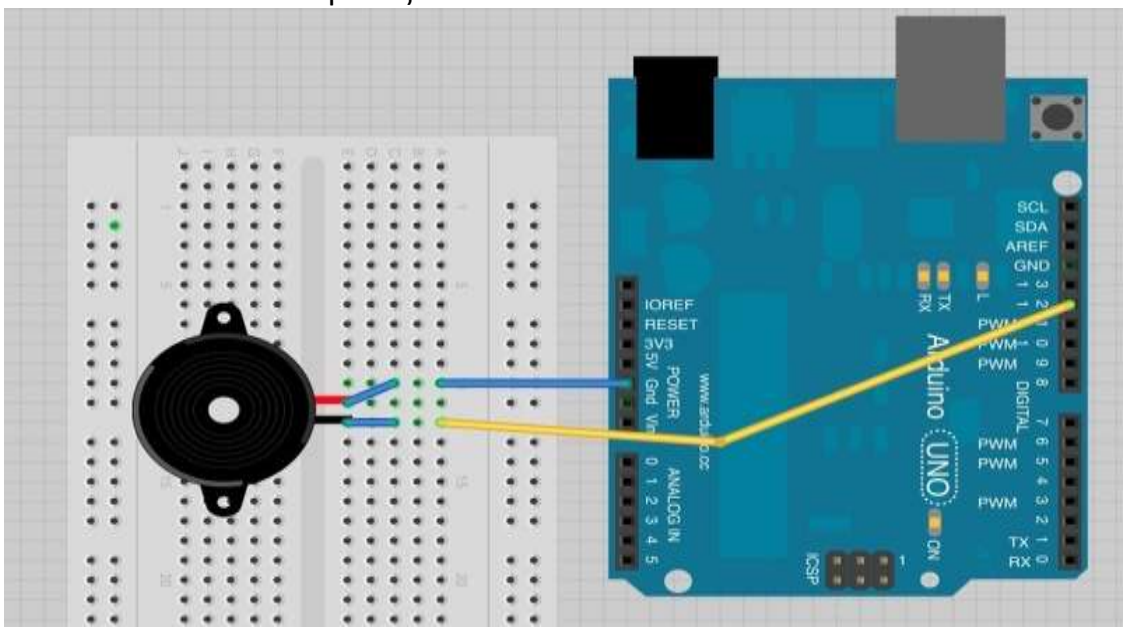
Jumper Cables

Buzzer

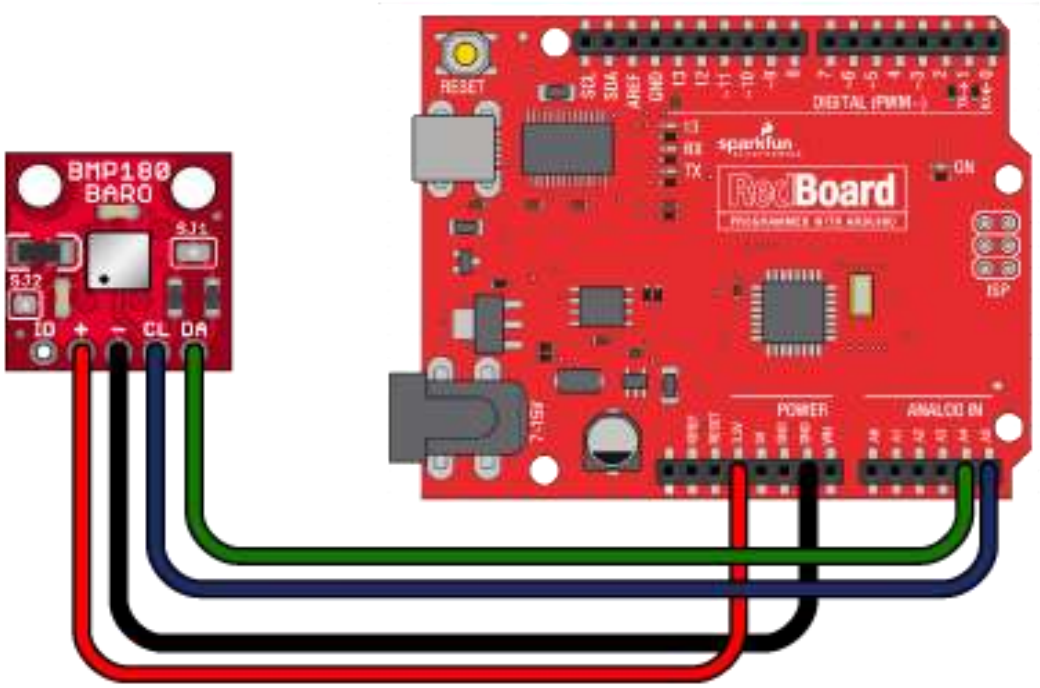


Το κύκλωμα έτοιμο

Έπειτα χρησιμοποιώντας Jamper cables θα συνδέσουν το άρθρωμα απευθείας πάνω στην πλακέτα του Arduino ακλουθώντας το σχέδιο κυκλώματος που θα τους δοθεί.
Στην πλακέτα θα συνδέσουν ομοίως και το Buzzer.



Σύνδεση του Buzzer



Σύνδεση του BMP180

Οι μαθητές πρέπει να προσέξουν ώστε, ο ακροδέκτης τροφοδοσίας του BMP180 να είναι συνδεδεμένος στα 3.3v της πλακέτας, αλλιώς υπάρχει κίνδυνος καταστροφής του. Ο ακροδέκτης SDA του I^2C στο A4 και ο SCL στο A5. Επίσης ο ακροδέκτης του buzzer θα συνδεθεί στο pin12 του Arduino.

- **Χρήση της βιβλιοθήκης του BMP180 – Γνωριμία με τις εντολές**

https://github.com/sparkfun/BMP180_Breakout_Arduino_Library

Από το πιο πάνω Link που θα δοθεί, οι μαθητές θα κατεβάσουν την βιβλιοθήκη για τον αισθητήρα , θα την εγκαταστήσουν στον υπολογιστή τους και θα γνωρίσουν τις εντολές που θα χρησιμοποιήσουν στο πρόγραμμα. Παρουσίαση των εντολών έχει ήδη κάνει ο καθηγητής στην αρχή του μαθήματος. Οι μαθητές μπορούν να «φορτώσουν» στο κύκλωμά τους τα έτοιμα παραδείγματα της βιβλιοθήκης και να παρατηρήσουν στην σειριακή οθόνη του Arduino τις διάφορες μετρήσεις που δίνει το αισθητήριο.

- **«Φόρτωμα» του προγράμματος βαριομέτρου**

Στην συνέχεια οι μαθητές θα μελετήσουν το πιο κάτω πρόγραμμα. Αυτό περιλαμβάνει εκτενή σχολιασμό για να διευκολύνει τους μαθητές στην κατανόηση του τρόπου λειτουργίας του.

Θα επισημανθούν οι διάφορες υπορουτίνες και η χρήση τους. Το πρόγραμμα θα φορτωθεί στο Arduino και οι μαθητές θα βλέπουν τις μετρήσεις στην σειριακή οθόνη. Ο κώδικας θα δοθεί σε έντυπη μορφή για μελέτη και σαν αρχείο απλού κειμένου (.txt) για «φόρτωμα».

ΑΚΟΛΟΥΘΕΙ Ο ΚΩΔΙΚΑΣ

```
/*
Στην εφαρμογή αυτή χρησιμοποιούμε τον αισθητήρα πίεσης BMP180
για να κατασκευάσουμε ένα Βαριόμετρο. Την συσκευή αυτή χρησιμοποιούμε στην
αεροπορία για να μετράμε την κατακόρυφη ταχύτητα ανόδου ή καθόδου από τις μεταβολές της ατμοσφαιρικής πίεσης.
Στο παράδειγμά μας θα έχουμε ακουστική σήμανση της ταχύτητας.
Θα γνωρίσουμε τις περισσότερες εντολές της βιβλιοθήκης του BMP180 ενώ θα χρησιμοποιήσουμε φίλτρο για να μειώσουμε το θόρυβο του αισθητηρίου.

Σε πραγματική υλοποίηση μπορούμε να αφαιρέσουμε τις εντολές που
εκτυπώνουν μηνύματα στην σειριακή θύρα.
*/

// Θα πρέπει να συμπεριλάβουμε την βιβλιοθήκη για το BMP180 και την Wire για το I2C
// Η Wire συμπεριλαμβάνεται στο ARDUINO
// Η βιβλιοθήκη για το BMP180 θα πρέπει να εγκατασταθεί πρώτα

#include <SFE_BMP180.h>
#include <Wire.h>

// Θα δημιουργήσουμε ένα αντικείμενο SFE_BMP180 το οποίο ονομάζουμε "pressure":
SFE_BMP180 pressure;

int speakerPin = 12;    //Ορίζεται το pin 12 για το buzzer
const int LI_filter = 48; //ορίζουμε τον συντελεστή του φίλτρου LI

/*
Ορίζουμε τους τόνους που θα επισημαίνουν τις διάφορες
ταχύτητες ανόδου ή καθόδου
*/

const int tone1=300;
const int tone2=400;
const int tone3=500;
const int tone4=100;

double P0  ; //αρχική τιμή πίεσης
long Time0 ; //αρχική χρονική στιγμή
```

```
void setup()
{
   Serial.begin(9600);  // Αρχικοποιούμε  την σειριακή θύρα
   Serial.println("REBOOT");

 /*
Η εντολή   «pressure.begin()»   αρχικοποιεί τον αισθητήρα
προκειμένου να ληφθούν τιμές αρχικοποίησης οι οποίες
και αποθηκεύονται στον αισθητήρα
*/

      if (pressure.begin())   {
      Serial.println("BMP180 init success");
      P0= getPressure();  // Με την υπορουτίνα  «getPressure» διαβάζουμε την αρχική πίεση
      Time0=millis();  // και σημειώνουμε την χρονική στιγμή της μέτρησης
     }
 else
         {
          // Ωχ ! Κάτι  πήγε λάθος. Συνήθως πρόβλημα συνδέσεων.
          // Ελέγξτε την σύνδεση του BMP180 στο ARDUINO

         Serial.println("BMP180 init fail\n\n");

          // Και με τρείς ηχητικούς τόνους από το Buzzer δηλώνουμε το σφάλμα

                  for (int i=0; i <= 2;  i++)  {
                  tone(speakerPin,tone1) ;
                  delay(500);
                  noTone(speakerPin);
                  delay(500);
                  }

         while(1); //  Ατέρμον βρόχος. Χρειάζεται reset.
         }
}
```

```
void loop()
{
  long Time; // χρονική στιγμή μέτρησης πίεσης
  double P , distance; //  Η πίεση και η υπολογισμένη υψομετρική απόσταση

      /*
      Με την υπορουτίνα  «getPressure» διαβάζουμε την  πίεση κάθε φορά
      και καταγράφουμε την χρονική στιγμή της μέτρησης , ώστε υπολογίζοντας
      από την διαφορά πιέσεων το ύψος κίνησης , μετράμε την ταχύτητα ανόδου ή καθόδου.
      */
  P = getPressure();
  Time = millis();

      /*
      Για να μειώσουμε το θόρυβο των μετρήσεων του αισθητηρίου
      χρησιμοποιούμε το φίλτρο LI που υλοποιείται με την πιο κάτω εξίσωση.
      Έτσι η μέτρηση  P κάθε φορά, αφαιρείται από τις προηγούμενες  P0,
       η διαφορά σταθμίζεται με τον συντελεστή  «LI_filter» και προστίθεται ξανά
      με τις προηγούμενες. Παίρνουμε έτσι μια διορθωμένη μέτρηση με λιγότερο θόρυβο.
      */
 P= P0 + (P-P0) / LI_filter;

        /*
      Η εντολή «pressure.altitude(P, P0)» της βιβλιοθήκης , μας δίνει την υψομετρική
      διαφορά σε μέτρα,  δύο σημείων που έχουν αντίστοιχα πιέσεις P (η μέτρηση κάθε
      φορά)  και P0 (η προηγούμενη μέτρηση).
      */
  distance = pressure.altitude(P, P0);

      /*
      Καλούμε την υπορουτίνα  «ring»  για να υπολογίσουμε την ταχύτητα ανόδου
      και να δημιουργήσουμε τις ηχητικές σημάνσεις
      */
ring( distance, Time);

      /*
      Η μετρούμενη πίεση  αποθηκεύεται για να χρησιμοποιηθεί στον επόμενο κύκλο
      όπως και η χρονική στιγμή της μέτρησης
      */
P0=P;
Time0 = Time;
 delay(60);  // Παύση 60 ms πριν την επόμενη μέτρηση
}
```

```
double getPressure()
{
char status; // Μεταβλητή για αποθήκευση της κατάστασης του αισθητηρίου
 double T,P; // Μεταβλητές για τις μετρήσεις θερμοκρασίας και πίεσης από το αισθητήριο

/*
Πριν την μέτρηση της πίεσης πρέπει να μετρήσουμε πρώτα την θερμοκρασία «T» για λόγους αντιστάθμισης. Αυτό γίνεται με την εντολή «pressure.getTemperature(T)».
Η μέτρηση της πίεσης «P» γίνεται με την εντολή «pressure.getPressure(P,T)» στην οποία χρησιμοποιούμε την θερμοκρασία «T».
Η εντολή «pressure.startTemperature()» ετοιμάζει τον αισθητήρα για την μέτρηση της θερμοκρασίας και επιστρέφει τον απαιτούμενο για την μέτρηση χρόνο σε ms. Αν αυτός είναι 0 , τότε υπάρχει σφάλμα και η μέτρηση δεν θα γίνει.
Η εντολή «pressure.startPressure(Z)» ετοιμάζει τον αισθητήρα για την μέτρηση της πίεσης και επιστρέφει τον απαιτούμενο για την μέτρηση χρόνο σε ms. Αν αυτός είναι 0 , τότε υπάρχει σφάλμα και η μέτρηση δεν θα γίνει. Η παράμετρος Z παίρνει τιμές 0,1,2,3 ανάλογα με την ακρίβεια της μέτρησης που επιθυμούμε. Μεγαλύτερη ακρίβεια σημαίνει περισσότερος χρόνος μέτρησης.
 */

/*
Εκκίνηση μέτρησης θερμοκρασίας. Εάν η εντολή εκτελεστεί , επιστρέφει τον απαιτούμενο για την μέτρηση χρόνο σε ms. Διαφορετικά επιστρέφει 0.
*/
 status = pressure.startTemperature();
        if (status != 0) {
        delay(status);  // Αναμονή για ολοκλήρωση μέτρησης θερμοκρασίας
```

```
/*
Ανάγνωση  της μετρούμενης θερμοκρασίας και αποθήκευση στην μεταβλητή «T»  .
Η εντολή επιστρέφει «1» αν εκτελεστεί επιτυχώς, διαφορετικά επιστρέφει «0» .
*/
        status = pressure.getTemperature(T);
                if (status != 0) {

/*
Εκκίνηση  μέτρησης πίεσης . Εάν η εντολή εκτελεστεί , επιστρέφει τον απαιτούμενο για την μέτρηση χρόνο σε ms.
Διαφορετικά επιστρέφει 0.
*/
                status = pressure.startPressure(0);
                        if (status != 0) {
                         delay(status); // Αναμονή για ολοκλήρωση μέτρησης πίεσης

/*
Ανάγνωση  της μετρούμενης πίεσης  και αποθήκευση στην μεταβλητή «P»  .
Η εντολή επιστρέφει «1» αν εκτελεστεί επιτυχώς, διαφορετικά επιστρέφει «0» .
Απαιτείται η προηγούμενη μέτρηση της θερμοκρασίας «T».
*/
                        status = pressure.getPressure(P,T);
                                if (status != 0) {
                                }
                                else Serial.println("Λάθος ανάκτησης πίεσης\n");
                        }
                        else Serial.println("Λάθος εκκίνησης μέτρησης πίεσης\n");
                }
                else Serial.println("Λάθος ανάκτησης θερμοκρασίας\n");
        }
        else Serial.println("Λάθος εκκίνησης μέτρησης θερμοκρασίας\n");

return(P); // Απόδοση από την υπορουτίνα της μετρούμενης πίεσης
}
```

```
void ring(double a , long  t )
{
double dt=t -Time0;  // Υπολογισμός του χρόνου ΔΤ μεταξύ μετρήσεων σε ms
double speed=1000 * a / dt ;    // Υπολογισμός ταχύτητας  σε m/sec

Serial.println();
Serial.print(speed,2);
Serial.print("  m/s");

/*
Αναλόγως της τιμής ταχύτητας καθόδου ή ανόδου αυτή  δηλώνεται ηχητικά με
ένα , δύο ή τρείς παλμούς ήχου, με διαφορετική συχνότητα   για κάθε περιοχή ταχυτήτων.
*/
if (speed<-0.4) {   //κίνηση προς τα κάτω
 tone(speakerPin,tone4) ;
 delay(300);
 noTone(speakerPin);
return;
}

if (speed>0.4 && speed<= 1) {
 tone(speakerPin,tone1) ;
 delay(300);
 noTone(speakerPin);
return;
}

if (speed>1 && speed <=2) {
 tone(speakerPin,tone2) ;
 delay(200);
 noTone(speakerPin);
 delay(200);
 tone(speakerPin,tone2) ;
 delay(200);
 noTone(speakerPin);
return;
}
```

```
if (speed>2) {
 tone(speakerPin,tone3) ;
 delay(100);
 noTone(speakerPin);
 delay(100);
 tone(speakerPin,tone3) ;
 delay(100);
 noTone(speakerPin);
 delay(100);
 tone(speakerPin,tone3) ;
 delay(100);
 noTone(speakerPin);
return;
}
}
```

ΤΕΛΟΣ ΤΟΥ ΚΩΔΙΚΑ

- **Εκτέλεση του προγράμματος – Διερεύνηση του κώδικα**

Οι μαθητές θα παρατηρήσουν στην σειριακή οθόνη του Arduino τις μετρήσεις ταχύτητας ανόδου – καθόδου , μετακινώντας κατακόρυφα την πλακέτα , καθώς και άλλα μηνύματα σχετικά με την λειτουργία της διάταξης. Αρχικά για να δούμε την επίδραση του θορύβου στις μετρήσεις αφήνουμε την κατασκευή ακίνητη και παρατηρούμε τις ενδείξεις. Η τιμή πρέπει να είναι 0.00 m/s. Λόγω θορύβου όμως οι τιμές έχουν μια διασπορά γύρω από αυτή την τιμή. Οι μαθητές θα σημειώσουν τις ακραίες τιμές που εμφανίζονται.
Στην συνέχεια επεμβαίνοντας στον κώδικα θα αφαιρέσουν από αυτόν την γραμμή :
P= P0 + (P-P0) / LI_filter; που βρίσκεται στην υπορουτίνα loop. Αυτή η γραμμή κώδικα , σε συνδυασμό με την : **delay(60);** στην ίδια υπορουτίνα , υλοποιούν στην ουσία ένα φίλτρο διέλευσης χαμηλών συχνοτήτων (Lossy Integrator) το οποίο αφαιρεί την επίδραση των έντονων διακυμάνσεων των τιμών με αντιστάθμισμα όμως την καθυστέρηση στην απόκριση από την αλλαγή της ταχύτητας (πίεσης). Το φίλτρο υλοποιείται υπολογίζοντας κάθε φορά το σταθμισμένο άθροισμα των παλιών τιμών και της νέας μέτρησης.
«Φορτώνοντας» τον τροποποιημένο κώδικα θα παρατηρήσουν , με ακίνητη πάλι την κατασκευή, την νέα διασπορά των μετρήσεων γύρω από την 0.00 m/s. Θα σημειώσουν ξανά τις ακραίες τιμές και θα συγκρίνουν με την προηγούμενη περίπτωση.
Ενδεικτικά πιο κάτω δίνονται μετρήσεις με χρήση του φίλτρου και χωρίς αυτό.

Οι μαθητές μπορούν να πειραματιστούν με την αποτελεσματικότητα του φίλτρου ρυθμίζοντας κάθε φορά τον συντελεστή **LI_filter** στην αρχή του κώδικα και αλλάζοντας την καθυστέρηση των 60 ms στο loop (συχνότητα δειγματοληψίας). Μπορούν επίσης , με την υπόδειξη του καθηγητή, να πειραματιστούν με την αποτελεσματικότητα και άλλων φίλτρων.

COM14 (Arduino/Genuino Uno)

```
BMP180 init success

20.92  m/s
0.74  m/s
-1.50  m/s
-1.05  m/s
1.62  m/s
0.58  m/s
0.38  m/s
0.00  m/s
-7.31  m/s
0.67  m/s
0.76  m/s
1.13  m/s
-0.16  m/s
-12.16  m/s
1.80  m/s
-1.01  m/s
1.13  m/s
-1.11  m/s
2.95  m/s
-1.84  m/s
0.95  m/s
0.96  m/s
0.67  m/s
-2.96  m/s
2.46  m/s
-0.99  m/s
0.00  m/s
2.97  m/s
1.11  m/s
-1.36  m/s
2.76  m/s
-1.30  m/s
0.00  m/s
13.80  m/s
-2.16  m/s
-0.50  m/s
1.25  m/s
0.79  m/s
0.00  m/s
-1.52  m/s
0.30  m/s
-6.80  m/s
-0.58  m/s
0.59  m/s
0.00  m/s
6.80  m/s
-1.41  m/s
-0.47  m/s
0.47  m/s
2.17  m/s
```

Αυτόματη κύλιση

Τιμές χωρίς φίλτρο

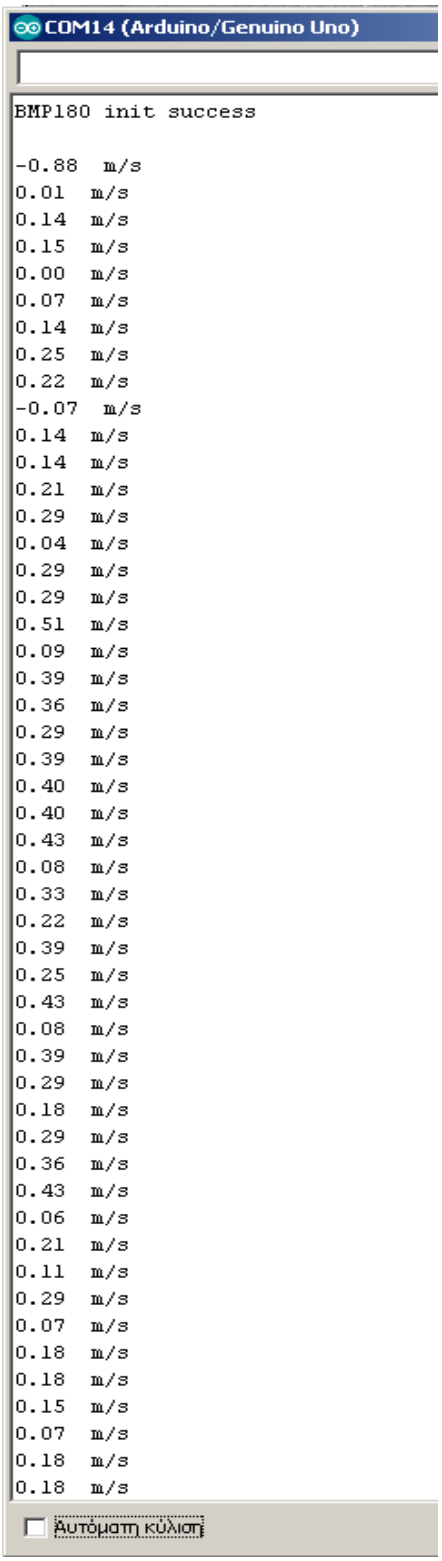
COM14 (Arduino/Genuino Uno)

```
BMP180 init success

-0.44  m/s
-0.01  m/s
-0.22  m/s
-0.15  m/s
0.07  m/s
0.07  m/s
-0.07  m/s
0.18  m/s
-0.07  m/s
-0.22  m/s
-0.07  m/s
0.18  m/s
0.11  m/s
-0.07  m/s
0.22  m/s
-0.14  m/s
0.04  m/s
0.07  m/s
0.11  m/s
0.07  m/s
0.07  m/s
-0.15  m/s
0.14  m/s
0.14  m/s
-0.15  m/s
0.14  m/s
-0.14  m/s
-0.14  m/s
-0.07  m/s
-0.29  m/s
-0.07  m/s
0.11  m/s
-0.14  m/s
0.04  m/s
0.04  m/s
0.11  m/s
-0.07  m/s
0.07  m/s
0.04  m/s
0.18  m/s
-0.14  m/s
0.04  m/s
-0.07  m/s
0.04  m/s
-0.07  m/s
0.04  m/s
-0.07  m/s
-0.22  m/s
-0.07  m/s
0.07  m/s
```

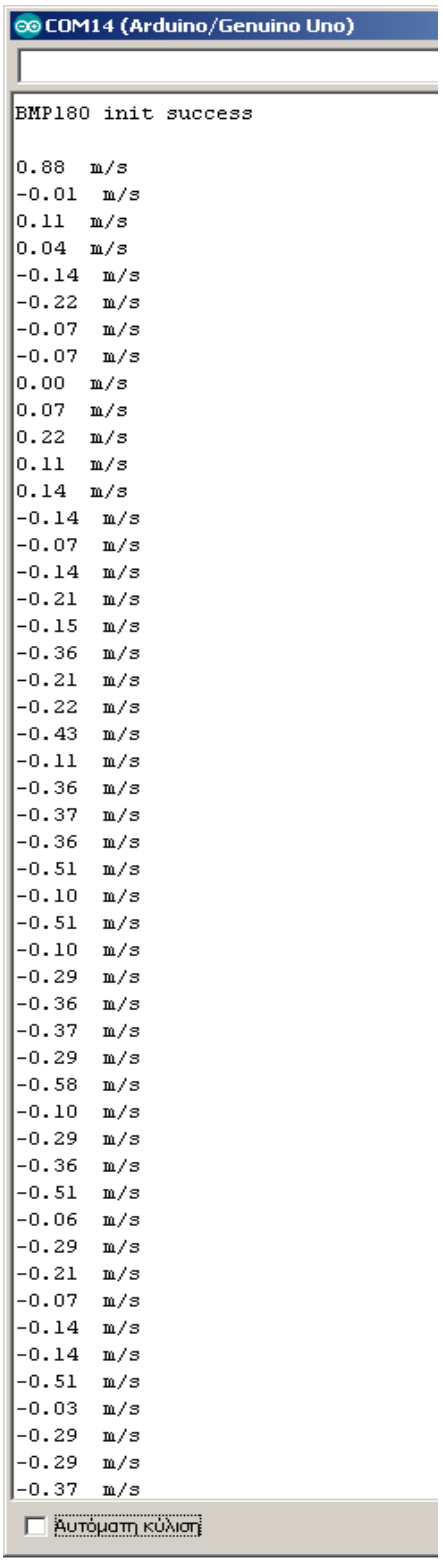
Αυτόματη κύλιση

Τιμές με χρήση φίλτρου

Στην συνέχεια οι μαθητές θα ελέγξουν την κατασκευή , αν ανταποκρίνεται στην ανοδική και καθοδική κίνηση . Θα ελέγξουν την απόκρισή της στις μεταβολές ταχύτητας σε σχέση με όσα ειπώθηκαν πιο πάνω για την χρήση φίλτρων. Μετακινώντας την πλακέτα κατακόρυφα , θα παρατηρήσουν στην σειριακή οθόνη τις μετρήσεις ταχύτητας και θα ελέγξουν βάση αυτών, την λειτουργία της ηχητικής σήμανσης από το buzzer. Ενδεικτικά πιο κάτω δίνονται μετρήσεις για κάθε περίπτωση.



Ανοδική κίνηση



Καθοδική κίνηση

Κατά την διάρκεια ελέγχου των μετρήσεων και λειτουργίας της διάταξης οι μαθητές θα πρέπει να τηρούν τις προδιαγραφές και περιορισμούς του κατασκευαστή για την χρήση του BMP180, όπως έχουν υποδειχθεί από τον καθηγητή στην αρχή του μαθήματος.

- **Συμπεράσματα**

Οι μαθητές συζητούν για την αποτελεσματική λειτουργία της διάταξης και προτείνουν τρόπους βελτίωσης της κατασκευής τους είτε βελτιώνοντας το κύκλωμα (θωρακίζοντάς το για το θόρυβο και τις επιδράσεις του περιβάλλοντος), είτε αλλάζοντας τον κώδικα (αύξηση της ακρίβειας μέτρησης από το BMP180).
Βρίσκουν στο Διαδίκτυο αντίστοιχα βαριόμετρα του εμπορίου και τα συγκρίνουν με την δική τους κατασκευή σε σχέση με τις επιδόσεις , τις προδιαγραφές το κόστος κ.λ.π. Προτείνουν τρόπους ώστε η κατασκευή αυτή να γίνει φορητή και πιο λειτουργική για τον χρήστη.

- **Δραστηριότητες μαθητών**

Οι δραστηριότητες που εκτέλεσαν οι μαθητές φαίνονται στην αναλυτική περιγραφή των παραπάνω επιμέρους βημάτων διδασκαλίας.

- **Αξιολόγηση μαθητών**

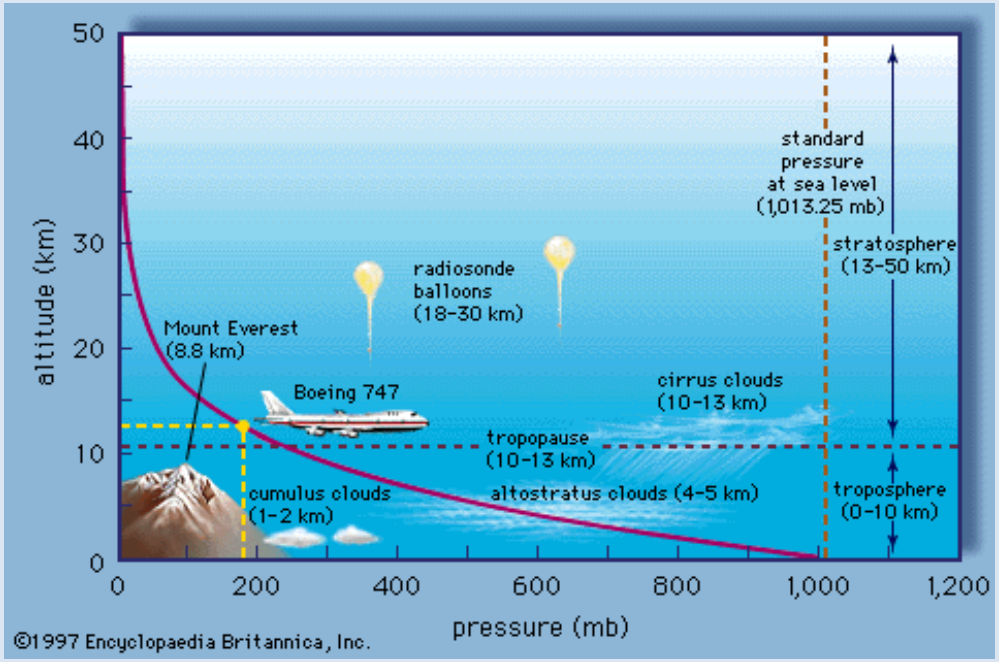
Δίνεται φύλλο αξιολόγησης. Επιλέγονται ερωτήσεις τις οποίες οι μαθητές πρέπει να είναι σε θέση να απαντήσουν.

ΜΕΛΕΤΗ ΤΟΥ ΨΗΦΙΑΚΟΥ ΒΑΡΟΜΕΤΡΙΚΟΥ ΑΙΣΘΗΤΗΡΑ ΠΙΕΣΗΣ BMP180
ΧΡΗΣΗ ΤΗΣ ΒΙΒΛΙΟΘΗΚΗΣ ΤΟΥ ΣΤΟ ARDUINO
ΕΦΑΡΜΟΓΗ ΣΤΗΝ ΚΑΤΑΣΚΕΥΗ ΒΑΡΙΟΜΕΤΡΟΥ

ΨΗΦΙΑΚΟΣ ΒΑΡΟΜΕΤΡΙΚΟΣ ΑΙΣΘΗΤΗΡΑΣ ΠΙΕΣΗΣ BMP180

https://learn.sparkfun.com/tutorials/bmp180-barometric-pressure-sensor-hookup-

- Είναι ένας βαρομετρικός αισθητήρας πίεσης με I ^{2}C ("Wire") interface
- Το BMP180 σχεδιάστηκε για να μετρήσει με ακρίβεια την ατμοσφαιρική πίεση
- Η ατμοσφαιρική πίεση, είναι η δύναμη που ο αέρας γύρω σας ασκεί για τα πάντα



$$\text{altitude} = 44330 * \left(1 - \left(\frac{p}{p_0}\right)^{\frac{1}{5.255}}\right)$$

ΨΗΦΙΑΚΟΣ ΒΑΡΟΜΕΤΡΙΚΟΣ ΑΙΣΘΗΤΗΡΑΣ ΠΙΕΣΗΣ BMP180

- Για να αντισταθμιστεί η θερμοκρασία, ο BMP180 περιλαμβάνει και **αισθητήρα θερμοκρασίας**

- **Ακρίβεια:** Η θεωρητική στάθμη θορύβου στο BMP180 είναι 0.25m αν και στην πράξη βλέπουμε θόρυβο της τάξης του 1m.

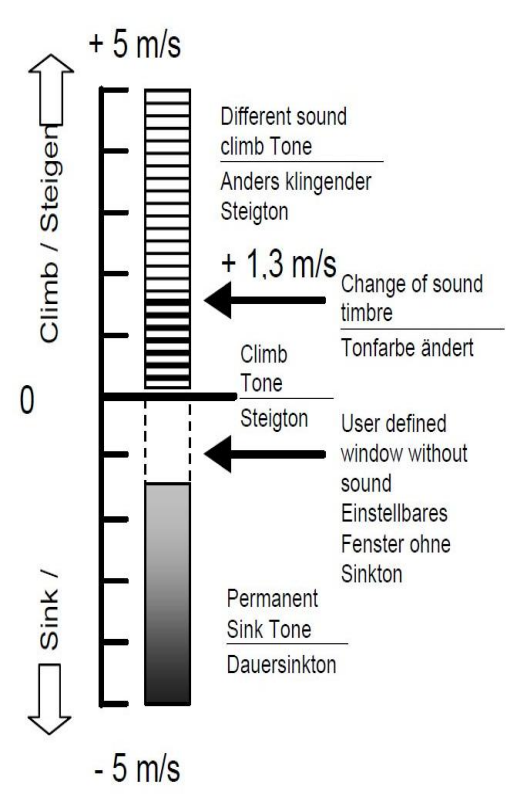
- **Μέγιστο υψόμετρο:** Ο BMP180 έχει κατώτερο όριο περίπου 300 hPa (mbar), το οποίο αντιστοιχεί σε υψόμετρο περίπου 3000μ ή 30.000 πόδια.

- **Ελάχιστο υψόμετρο:** Ο αισθητήρας έχει ανώτερο όριο 1100 hPa = mbar (ή 16 psi), η οποία είναι περίπου 500 πόδια κάτω από την επιφάνεια της θάλασσας.

Συμβουλές και κόλπα
Πράγματα που πρέπει να προσέξετε

- **Δώστε τη σωστή τάση:** Ο BMP180 θα λειτουργεί με τάσεις από 1.8V έως 3.6V.
- **Δώστε αέρα:** Να θυμάστε ότι ο BMP180 πρέπει να έχει πρόσβαση σε ατμοσφαιρικό αέρα για τη μέτρηση της πίεσής του, γι 'αυτό μην το βάλετε σε ένα σφραγισμένο θάλαμο.
- **Αλλά όχι πάρα πολύ αέρα:** Από την άλλη πλευρά, η έκθεση σε γρήγορη κίνηση του αέρα ή του ανέμου μπορεί να προκαλέσει στιγμιαίες διακυμάνσεις της πίεσης που θα επηρεάσουν τις μετρήσεις σας. Θωρακίστε τη συσκευή από ισχυρά ρεύματα αέρα.
- **Φροντίστε να είναι δροσερό:** Επειδή ακριβής μέτρηση της θερμοκρασίας απαιτείται για τη μέτρηση της πίεσης, προσπαθήστε να μην εκθέτετε τη συσκευή σε απότομες αλλαγές θερμοκρασίας, και κρατήστε το μακριά από πηγές θερμότητας.
- **Φροντίστε να είναι στεγνό:** Το BMP180 είναι ευαίσθητο στην υγρασία. Μην το βυθίζετε ή αφήνετε να έρθει σε επαφή με νερό σε υγρή μορφή.
- **Μην το εκθέτετε σε φώς:** Παραδόξως, το πυρίτιο εντός του BMP180 είναι ευαίσθητο στο φως, το οποίο μπορεί να εισέλθει στη συσκευή μέσω της οπής στην κορυφή του chip. Για μέγιστη ακρίβεια, θωρακίστε το τσιπ από το φως του περιβάλλοντος.

ΒΑΡΙΟΜΕΤΡΟ



Ο δείκτης κατακόρυφης ταχύτητας. Αυτός είναι ο πιο κοινός τύπος που χρησιμοποιείται σε αεροσκάφη, δείχνοντας ταχύτητα σε πόδια ανά λεπτό

Η ΒΙΒΛΙΟΘΗΚΗ ΤΟΥ BMP180

https://github.com/sparkfun/BMP180_Breakout_Arduino_Library

- **#include <SFE_BMP180.h>**
  Η βιβλιοθήκη για το BMP180 θα πρέπει να εγκατασταθεί πρώτα

- **SFE_BMP180 pressure;**
  Θα δημιουργήσουμε ένα αντικείμενο SFE_BMP180 το οποίο ονομάζουμε "pressure":

- **pressure.begin()**
  Η εντολή «pressure.begin()» αρχικοποιεί τον αισθητήρα προκειμένου να ληφθούν τιμές αρχικοποίησης οι οποίες και αποθηκεύονται στον αισθητήρα

- **pressure.altitude(P, P0)**
  Η εντολή «pressure.altitude(P, P0)» της βιβλιοθήκης , μας δίνει την υψομετρική διαφορά σε μέτρα, δύο σημείων που έχουν αντίστοιχα πιέσεις P (η μέτρηση κάθε φορά) και P0 (η προηγούμενη μέτρηση).

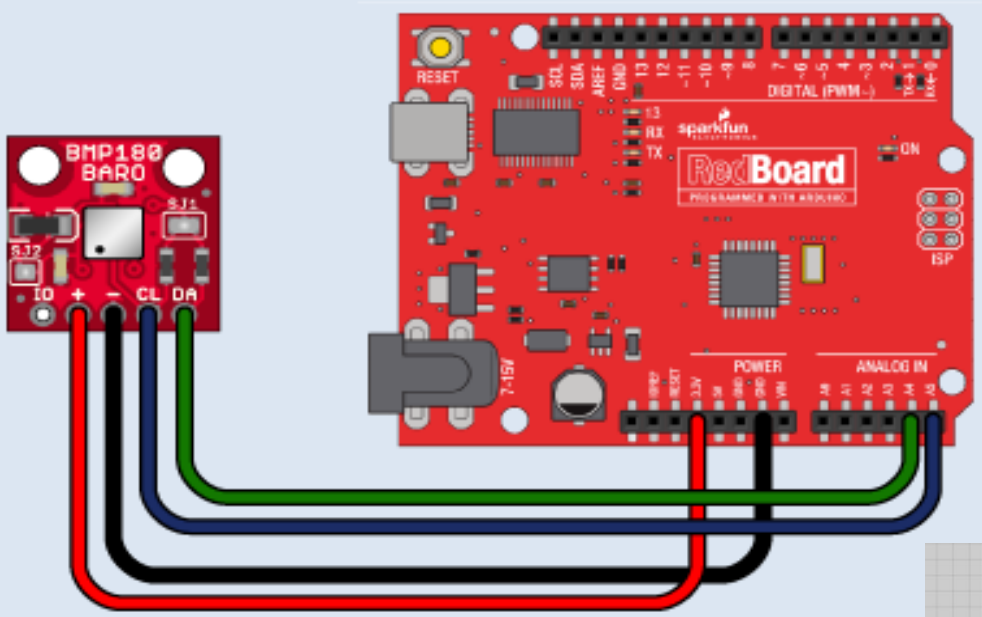
- **status = pressure.startTemperature()**
  Η εντολή «pressure.startTemperature()» ετοιμάζει τον αισθητήρα για την μέτρηση της θερμοκρασίας και επιστρέφει τον απαιτούμενο για την μέτρηση χρόνο σε ms. Αν αυτός είναι 0 , τότε υπάρχει σφάλμα και η μέτρηση δεν θα γίνει.

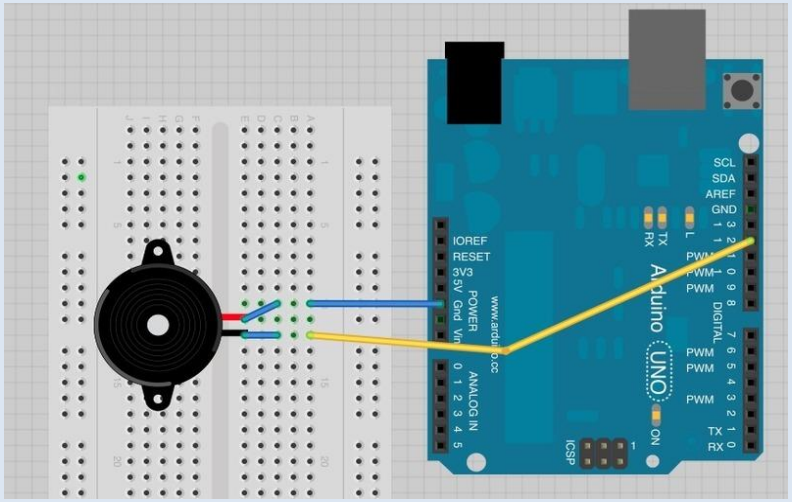
- **status = pressure.getTemperature(T)**
  Ανάγνωση της μετρούμενης θερμοκρασίας και αποθήκευση στην μεταβλητή «T» .Η εντολή επιστρέφει «1» αν εκτελεστεί επιτυχώς, διαφορετικά επιστρέφει «0» .

- **status = pressure.startPressure(0)**
  Η εντολή «pressure.startPressure(Z)» ετοιμάζει τον αισθητήρα για την μέτρηση της πίεσης και επιστρέφει τον απαιτούμενο για την μέτρηση χρόνο σε ms. Αν αυτός είναι 0 , τότε υπάρχει σφάλμα και η μέτρηση δεν θα γίνει. Η παράμετρος Z παίρνει τιμές 0,1,2,3 ανάλογα με την ακρίβεια της μέτρησης που επιθυμούμε. Μεγαλύτερη ακρίβεια σημαίνει περισσότερος χρόνος μέτρησης.

- **status = pressure.getPressure(P,T)**
  Η μέτρηση της πίεσης «P» γίνεται με την εντολή «pressure.getPressure(P,T)» στην οποία χρησιμοποιούμε την θερμοκρασία «T». Η εντολή επιστρέφει «1» αν εκτελεστεί επιτυχώς, διαφορετικά επιστρέφει «0» .
  Απαιτείται η προηγούμενη μέτρηση της θερμοκρασίας «T».

ΤΟ ΚΥΚΛΩΜΑ ΤΗΣ ΚΑΤΑΣΚΕΥΗΣ





Ο ΚΩΔΙΚΑΣ

```
//  Θα πρέπει να συμπεριλάβουμε την βιβλιοθήκη για το BMP180 και την Wire για το I2C
//  Η Wire συμπεριλαμβάνεται στο ARDUINO
//  Η βιβλιοθήκη για το BMP180 θα πρέπει να εγκατασταθεί πρώτα
#include <SFE_BMP180.h>
#include <Wire.h>
// Θα δημιουργήσουμε ένα αντικείμενο  SFE_BMP180 το οποίο ονομάζουμε  "pressure":
SFE_BMP180 pressure;
int speakerPin = 12;    //Ορίζεται το pin 12 για το buzzer
const int LI_filter = 48; //ορίζουμε τον συντελεστή του φίλτρου LI
const int tone1=300;
const int tone2=400;
const int tone3=500;
const int tone4=100;
double P0  ; //αρχική τιμή πίεσης
long Time0 ; //αρχική χρονική στιγμή

void setup()
{
    Serial.begin(9600);  // Αρχικοποιούμε  την σειριακή θύρα
    Serial.println("REBOOT");
     if (pressure.begin())   {
    Serial.println("BMP180 init success");
     P0= getPressure();  // Με την υπορουτίνα  «getPressure» διαβάζουμε την αρχική πίεση
     Time0=millis();  // και σημειώνουμε την χρονική στιγμή της μέτρησης
     }
  else
          {
           // Ωχ ! Κάτι  πήγε λάθος. Συνήθως πρόβλημα συνδέσεων.
           // Ελέγξτε την σύνδεση του BMP180 στο ARDUINO
          Serial.println("BMP180 init fail\n\n");
           // Και με τρείς ηχητικούς τόνους από το Buzzer δηλώνουμε το σφάλμα
                        for (int i=0; i <= 2;  i++)  {
                        tone(speakerPin,tone1) ;
                        delay(500);
                        noTone(speakerPin);
                        delay(500);
                        }
          while(1); //  Ατέρμον βρόχος. Χρειάζεται reset.
          }
}
```

```
void loop()
{
  long Time; // χρονική στιγμή μέτρησης πίεσης
  double P , distance; //  Η πίεση και η υπολογισμένη υψομετρική απόσταση
P = getPressure();
Time = millis();

 P= P0 + (P-P0) / LI_filter;

distance = pressure.altitude(P, P0);
ring( distance, Time);
P0=P;
Time0 = Time;
 delay(60);  // Παύση 60 ms πριν την επόμενη μέτρηση
}



double getPressure()
{
char status; // Μεταβλητή για αποθήκευση της κατάστασης του αισθητηρίου
 double T,P; // Μεταβλητές για τις μετρήσεις θερμοκρασίας και πίεσης από το αισθητήριο

status = pressure.startTemperature();
            if (status != 0) {
            delay(status);  // Αναμονή για ολοκλήρωση μέτρησης θερμοκρασίας
            status = pressure.getTemperature(T);
                                    if (status != 0) {
                                   status = pressure.startPressure(0);
                                                        if (status != 0) {
                                                         delay(status); // Αναμονή για ολοκλήρωση μέτρησης πίεσης
                                                        status = pressure.getPressure(P,T);
                                                                          if (status != 0) {
                                                                         }
                                                                         else Serial.println("Λάθος ανάκτησης πίεσης\n");
                                                        }
                                                        else Serial.println("Λάθος εκκίνησης μέτρησης πίεσης\n");
                                    }
                                   else Serial.println("Λάθος ανάκτησης θερμοκρασίας\n");
            }
            else Serial.println("Λάθος εκκίνησης μέτρησης θερμοκρασίας\n");
return(P); // Απόδοση από την υπορουτίνα της μετρούμενης πίεσης
}
```

```
void ring(double a , long  t )
{
double dt=t -Time0;  // Υπολογισμός του χρόνου ΔΤ μεταξύ μετρήσεων σε ms
double speed=1000 * a / dt ;    // Υπολογισμός ταχύτητας  σε m/sec
Serial.println();
Serial.print(speed,2);
Serial.print("  m/s");

if (speed<-0.4) {   //κίνηση προς τα κάτω
 tone(speakerPin,tone4) ;
 delay(300);
 noTone(speakerPin);
return;
}

if (speed>0.4 && speed<= 1) {
 tone(speakerPin,tone1) ;
 delay(300);
 noTone(speakerPin);
return;
}

if (speed>1 && speed <=2) {
 tone(speakerPin,tone2) ;
 delay(200);
 noTone(speakerPin);
 delay(200);
 tone(speakerPin,tone2) ;
 delay(200);
 noTone(speakerPin);
return;
}

if (speed>2) {
 tone(speakerPin,tone3) ;
 delay(100);
 noTone(speakerPin);
 delay(100);
 tone(speakerPin,tone3) ;
 delay(100);
 noTone(speakerPin);
 delay(100);
 tone(speakerPin,tone3) ;
 delay(100);
 noTone(speakerPin);
return;
}
}
```

Ο ΘΟΡΥΒΟΣ ΣΤΙΣ ΜΕΤΡΗΣΕΙΣ

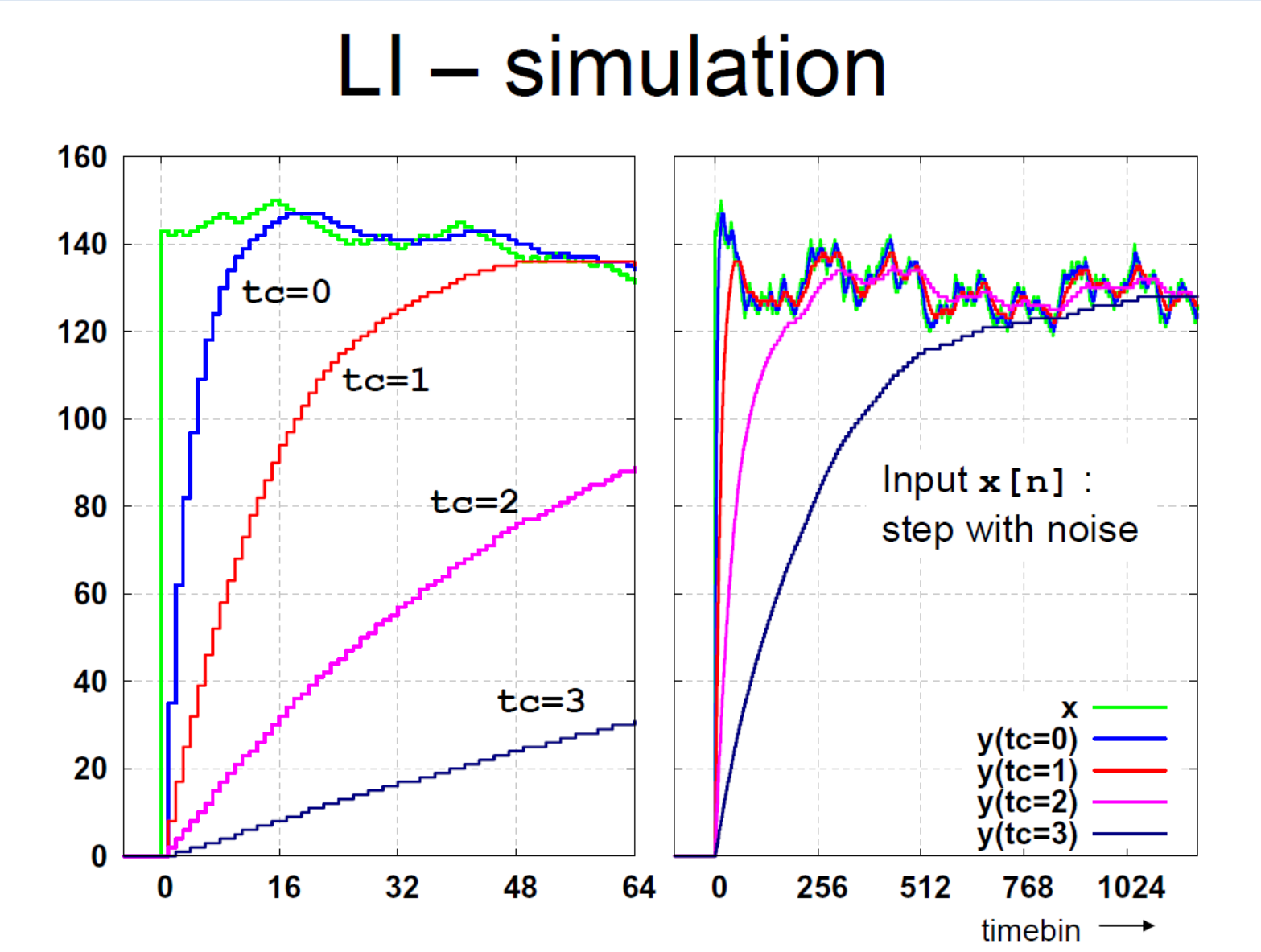
IIR – lossy integrator (LI)

- Simple example: in many cases the result of a single measurement is too noisy and an average over some time is preferred
- The lossy integrator (or relaxation filter) takes a weighted sum of the old output and the new input value:

$$y[n+1] = \frac{m-1}{m} y[n] + \frac{1}{m} x[n]$$

- In case of $m=2^K$ it can be easily realised without expensive multiplier or divider

Ο ΘΟΡΥΒΟΣ ΣΤΙΣ ΜΕΤΡΗΣΕΙΣ



ΣΥΜΠΕΡΑΣΜΑΤΑ

+

VS

Σχολική Μονάδα: 1ο ΕΚ Άργους

Συμμετέχοντες Καθηγητές

ΠΙΣΤΕΥΟΣ ΝΙΚΟΣ
ΤΣΙΓΚΑΣ ΗΛΙΑΣ
ΓΛΕΝΤΖΕΣ ΙΩΑΝΝΗΣ
ΔΕΔΕΣ ΚΩΣΤΑΣ

1° Εργαστηριακό Κέντρο Άργους

Σχέδιο Μαθήματος

Ομάδα Εργασίας: ***Πιστεύος Νικόλαος, Τσίγκας Ηλίας, Δέδες Κωνσταντίνος, Γλεντζές Ιωάννης***

Χρονικά περιθώρια τέλεσης της άσκησης: 3 (τρεις) εργαστηριακές ώρες

Άσκηση: Σύνδεση μίας πλακέτας Arduino Uno και μιας οθόνης LCD 20x4 μέσω του I2C

Στόχος της άσκησης είναι η εξοικείωση με τη σύνδεση μίας πλακέτας Arduino Uno και μιας οθόνης LCD 20x4 μέσω του I2C και την εμφάνιση κειμένου σε αυτή.
Για το χειρισμό της οθόνης θα γίνει χρήση της βιβλιοθήκης του Arduino, στη διεύθυνση:https://bitbucket.org/fmalpartida/new-liquidcrystal/downloads/

Το κείμενο που θα εμφανιστεί στην οθόνη θα είναι το παρακάτω:

1ο E.K ARGOUS

ELECTRONIC TEAM

ARDUINO

1. Εισαγωγή - Arduino Uno .

1.1 Γενικά

Η καρδιά του Arduino Uno είναι φυσικά ένας μικροεπεξεργαστής. Αυτός είναι το «μυαλό» του Arduino και είναι προγραμματιζόμενος ώστε να ελέγχει τα 14 ψηφιακά input/output pins και τα 6 αναλογικά που υπάρχουν πάνω στην πλακέτα ανάπτυξης. Δια μέσου αυτών των 20 pins γίνονται όλες οι διασυνδέσεις με εξωτερικά στοιχεία (κινητήρες, LEDs, LCD οθόνες κλπ) και αισθητήρες (Ultrasonic, θερμόμετρα, accelometers κ.α). Στην πλακέτα ανάπτυξης υπάρχει μια θύρα USB. Μέσω αυτής γίνεται η μεταφορά δεδομένων από αυτήν προς κάποια άλλη συσκευή, συνήθως έναν υπολογιστή, και το αντίστροφο. Ωστόσο, η κύρια χρήση στα αρχικά στάδια εκμάθησης είναι η μεταφορά του προγράμματος από τον υπολογιστή στον μικροεπεξεργαστή αλλά και η οπτικοποίηση των δεδομένων που απορρέουν από την λειτουργία της συσκευής μετά από το προγραμματισμό.

1.2 Μέρη ενός Arduino Uno.

Το Arduino έχει 14 ψηφιακούς ακροδέκτες Εισόδου/Εξόδου οι οποίοι μπορούν να τεθούν ως είσοδοι ή ως έξοδοι με τις εντολές-συναρτήσεις pinMode(), digitalWrite(), και digitalRead() που θα αναλυθούν παρακάτω. Λειτουργούν στα 5 Volts και έχουν την δυνατότητα να παρέχουν ή να καταβυθίζουν ένταση της τάξεως των 40mA. Σε κάθε pin υπάρχει εσωτερικά ένας Pull-up αντιστάτης στα 20-50KΩ. Επιπλέον, έχει 6 αναλογικούς ακροδέκτες Εισόδου. Αυτοί μπορούν να διαβάσουν αναλογικές τιμές όπως η τάση μιας μπαταρίας κτλ και να τις μετατρέψουν σε έναν αριθμό από 0-1023. Η μέτρηση της τάσης γίνεται από προκαθορισμένα από 0 έως 5 volts. Εκτός αυτού, 6 εκ των 14 ψηφιακών ακροδεκτών οι P3, P5, P6, P9, P10 και P11 έχουν την δυνατότητα να προγραμματιστούν ώστε να λειτουργούν ως αναλογικές Έξοδοι.

1.3 Χαρακτηριστικά του Arduino

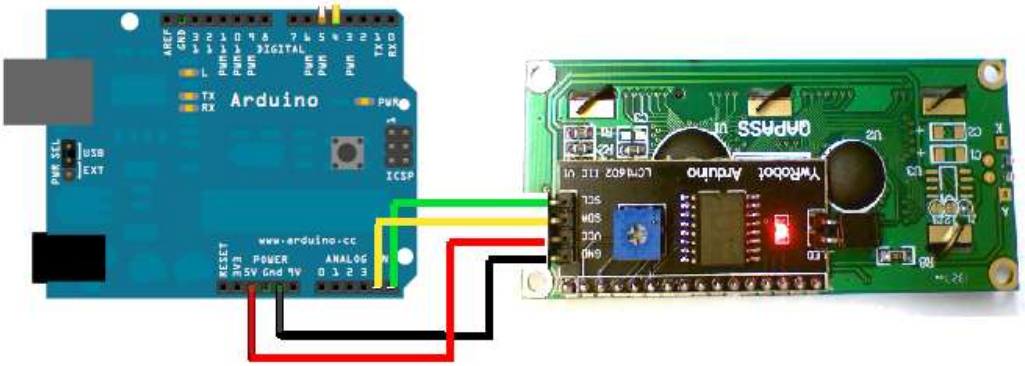
- Microcontroller: ATmega328
- Τάση λειτουργίας: 5V
- Τάση εισόδου: 7-12V
- Τάση εισόδου (όριο): 6-20V
- Digital I/O Pins: 14 (εκ των οποίων 6 περιέχουν PWM εξόδους)
- Analog Input Pins: 6
- DC ρεύματος I/O Pin: 40 mA
- DC τρέχουσα για 3.3V Pin: 50 mA
- Flash Memory: 32 KB εκ των οποίων 0,5 KB που χρησιμοποιούνται από τον bootloader
- SRAM: 2 KB
- EEPROM: 1 KB
- Clock Speed: 16 MHz

2. Εργαστηριακός Εξοπλισμός

- Πλακέτα Arduino Uno
- Καλώδιο USB,
- Καλώδια σύνδεσης πλακέτας – οθόνης *(Jumper wire pack)*
- Ηλεκτρονικός υπολογιστής και
- Οθόνη LCD 20x4 - I2C.

3. Πορεία Εργασίας

1. Συνδέουμε την πλακέτα Arduino Uno με την οθόνη LCD 20x4 - I2C και κατόπιν με τον υπολογιστή χρησιμοποιώντας το καλώδιο USB. Παρατηρούμε ότι το LED της πλακέτας ανάβει.



2. Περιβάλλον Arduino: Στον υπολογιστή έχει ήδη μεταφορτώθεί η τελευταία έκδοση Arduino από την ιστοσελίδα http://arduino.cc/en/Main/Software ενώ έχει γίνει και η εγκατάσταση του προγράμματος.

3. Έναρξη της εφαρμογής Arduino: Κάνουμε διπλό κλικ στην εφαρμογή Arduino.exe

4. Επιλογή Board και Σειριακής θύρας Από το περιβάλλον ανάπτυξης του, από το μενού Tools επιλέγουμε για Board το Arduino Uno και για Σειριακή θύρα το COM3.

5. Ολοκληρωμένο περιβάλλον ανάπτυξης (IDE) του Arduino: Πληκτρολογούμε το παρακάτω πρόγραμμα:

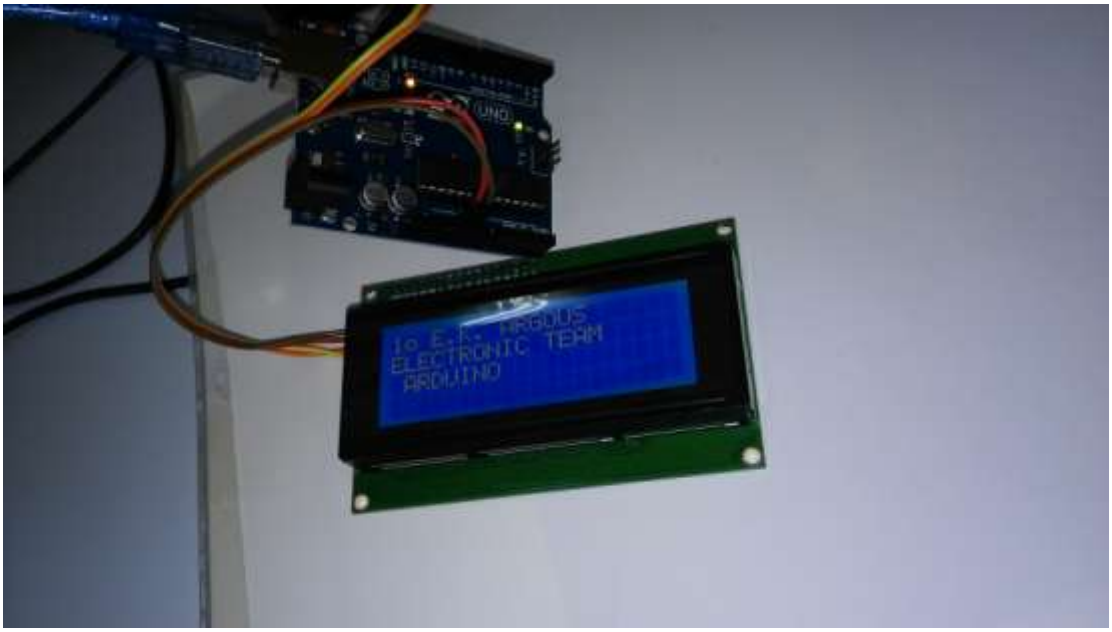
```
#include <Wire.h>

#include <LiquidCrystal_I2C.h>

#define BACKLIGHT_PIN     13
```

```
LiquidCrystal_I2C lcd(0x27,2,1,0,4,5,6,7,3,POSITIVE);  // Set the LCD I2C address


//LiquidCrystal_I2C lcd(0x27, 2, 1, 0, 4, 5, 6, 7, 3, POSITIVE);  // Set the LCD I2C address


// Creat a set of new characters
const uint8_t charBitmap[][8] = {
  { 0xc, 0x12, 0x12, 0xc, 0, 0, 0, 0 },
  { 0x6, 0x9, 0x9, 0x6, 0, 0, 0, 0 },
  { 0x0, 0x6, 0x9, 0x9, 0x6, 0, 0, 0x0 },
  { 0x0, 0xc, 0x12, 0x12, 0xc, 0, 0, 0x0 },
  { 0x0, 0x0, 0xc, 0x12, 0x12, 0xc, 0, 0x0 },
  { 0x0, 0x0, 0x6, 0x9, 0x9, 0x6, 0, 0x0 },
  { 0x0, 0x0, 0x0, 0x6, 0x9, 0x9, 0x6, 0x0 },
  { 0x0, 0x0, 0x0, 0xc, 0x12, 0x12, 0xc, 0x0 }

};
void setup()
{
  int charBitmapSize = (sizeof(charBitmap ) / sizeof (charBitmap[0]));


 // Switch on the backlight
 pinMode ( BACKLIGHT_PIN, OUTPUT );
 digitalWrite ( BACKLIGHT_PIN, HIGH );


 lcd.begin(20,4);            // initialize the lcd


  for ( int i = 0; i < charBitmapSize; i++ )
```

```
  {
    lcd.createChar ( i, (uint8_t *)charBitmap[i] );
  }

 lcd.home ();                // go home
 lcd.print("1o E.K. ARGOUS");
 delay ( 1000 );
 lcd.setCursor (0,1);       // go to the next line
 lcd.print ("ELECTRONIC TEAM");
 delay ( 1000 );
 lcd.setCursor (1,2);       // go to the next line
 lcd.print ("ARDUINO");

}
void loop()
{

}
```

6. Το κείμενο που θα εμφανιστεί στην οθόνη είναι το παρακάτω:

```
1o E.K ARGOUS
ELECTRONIC TEAM
  ARDUINO
```



7. Τέλος άσκησης

4. Βιβλιογραφία – Πηγές πληροφοριών

Ιστοσελίδες που χρησιμοποιήθηκαν:

http://arduino.cc/en/Guide/Environment?from=Tutorial.Bootloader

http://arduino.cc/en/Guide/Windows

http://arduino.cc/en/Reference/HomePage

http://arduino.cc/en/Main/ArduinoBoardUno

http://arduino.cc/en/Tutorial/Memor

https://bitbucket.org/fmalpartida/new-liquidcrystal/downloads/

http://arduino-info.wikispaces.com/LCD-Blue-I2C

Σχολική Μονάδα: 3ο ΓΕΛ Βέροιας

Συμμετέχοντες Καθηγητές

ΚΑΡΑΛΕΚΑΣ ΓΕΩΡΓΙΟΣ

ΕΣΑΓΩΓΗ ΣΤΟΝ ΑΥΤΟΜΑΤΙΣΜΟ

1.ΣΥΝΟΠΤΙΚΗ ΠΑΡΟΥΣΙΑΣΗ ΣΕΝΑΡΙΟΥ

1.1 Τίτλος διδακτικού σεναρίου

Εισαγωγή στον αυτοματισμό, προσομοίωση τμημάτων του νευρικού συστήματος του ανθρώπου με τη βοήθεια του ARDUINO.

1.2 Εμπλεκόμενες γνωστικές περιοχές

Βιολογία : Νευρικό σύστημα, Φυσική: Ηλεκτρισμός, Πληροφορική :προγραμματισμός σε γλώσσα **Wiring**, μια παραλλαγή της C/C++ για μικροελεγκτές.

1.3 Τάξεις στις οποίες μπορεί να απευθύνεται

Β΄ Λυκείου θετικής Κατεύθυνσης

1.4 Συμβατότητα με το αναλυτικό πρόγραμμα

Από το αναλυτικό Πρόγραμμα προβλέπεται η διδασκαλία του κεφαλαίου Κεντρικό νευρικό σύστημα στην πρώτη λυκείου , το κεφάλαιο του ηλεκτρισμού στη Δευτέρα λυκείου αλλά και το μάθημα της πληροφορικής στην πρώτη λυκείου

1.5 Οργάνωση της διδασκαλίας και απαιτούμενη υλικοτεχνική υποδομή

Οι μαθητές θα εργαστούν χωρισμένοι σε ομάδες των 2-3 ατόμων στο Εργαστήριο Πληροφορικής του σχολείου. Εναλλακτικά η διδασκαλία μπορεί να πραγματοποιηθεί στο Εργαστήριο των Φυσικών Επιστημών με τη χρήση των απαραίτητων υπολογιστών.

Λογισμικό: << Arduino Software (IDE) >>

1.6 Διδακτικοί στόχοι

Οι μαθητές στο τέλος της διδακτικής ώρας, εργαζόμενοι σε συνθήκες εργαστηρίου θα πρέπει να είναι σε θέση:

- να κατανοούν πως μπορεί να προσομοιωθεί μια ανθρώπινη λειτουργεία μέσα από ένα ηλεκτρονικό κύκλωμα.
- να διακρίνουν τη σχέση ανάμεσα στις εισόδους και τις εξόδους ενός κυκλώματος αυτοματισμού και την αντιστοίχισή τους με τις αισθητικές και κινητικές νευρικές οδούς στο ανθρώπινο σώμα.
- να διακρίνουν τις διαφορές ανάμεσα σε αισθητήρες και συσκευές εκτέλεσης και πώς αυτές ελέγχονται από των μικροελεγκτή, σε σχέση πάντα με την αντίστοιχη των ανθρωπίνων λειτουργειών.
- να κατανοήσουν πώς μία φυσική μεταβλητή μετατρέπεται σε ηλεκτρική τάση και πώς αυτή με τη σειρά της μέσα από τον AD μετατροπέα γίνεται αριθμός.

- να κατανοήσουν και να μπορούν μέσα από απλές πράξεις να επεξεργαστούν τα δεδομένα από τον αισθητήρα θερμοκρασίας ώστε να χρησιμοποιηθούν με τον κατάλληλο τρόπο από τον μικροελεγκτή.
- να κατανοήσουν τις βασικές εντολές και συναρτήσεις της γλώσσας προγραμματισμού και να μπορούν να φτιάξουν ένα δικό τους πρόγραμμα.
- να γνωρίσουν την έννοια της μεταγλώττισης Compile ενός skech από των υπολογιστή και την σύνδεσή του με το ARDUINO.
- Να συνειδητοποιήσουν πως προσομοιώνονται κάποιες ανθρώπινες λειτουργείες μέσα από τις μηχανές. Στο κύκλωμα που θα υλοποιήσουν, προσομοιώνονται η λειτουργεία της κόρης του οφθαλμού και η λειτουργεία της εφίδρωσης. Αντίστοιχα αυτές οι λειτουργείες σ' ένα ρομπότ μπορούν να είναι η μείωση της φωτεινότητας σε ένα αισθητήρα πχ κάμερα και η λειτουργεία ενός συστήματος ψύξης εάν ξεπεράσει η θερμοκρασία ένα όριο σε κάποιο κύκλωμα.
- Να κατανοήσουν ότι όπως στον άνθρωπο, το σήμα από το αισθητήριο όργανο, φτάνει μέσα από το αισθητικό νεύρο στον εγκέφαλο όπου και επεξεργάζεται και μετά από το κινητικό νεύρο φτάνει η εντολή του εγκεφάλου στο εκτελεστικό όργανο, έτσι και σ' ένα ρομπότ η όλη διαδικασία είναι παρόμοια. Μόνο που εδώ το ρόλο των νευρώνων τον παίζουν τα καλώδια και τον ρόλο του εγκεφάλου ο μικροελεγκτής.
- Τέλος μπορούν να καταλάβουν ότι όπως και ο άνθρωπος έχει τα αντανακλαστικά όπου το σήμα από ένα αισθητήριο δεν περνά κατ' ανάγκη από τον εγκέφαλο αλλά πάει κατ' ευθείαν σχεδόν στο εκτελεστικό όργανο έτσι και σ' ένα ρομπότ μπορεί αυτό να συμβεί.

1.7 Εκτιμώμενη διάρκεια

Δύο διδακτικές ώρες. Η πρώτη για τη απαραίτητη θεωρία για το ανθρώπινο Νευρικό σύστημα και τις γνώσεις για τα απαραίτητα ηλεκτρονικά για την κατανόηση του κυκλώματος που θα υλοποιηθεί στο εργαστήριο Πληροφορικής. Και η δεύτερη για την εξήγηση των απαιτούμενων εντολών για τον προγραμματισμό και η υλοποίηση του κυκλώματος.

2. Το προτεινόμενο σενάριο

Α. Η οργάνωση της διδασκαλίας που προτείνεται έχει ως εξής:

Πρώτη διδακτική ώρα:

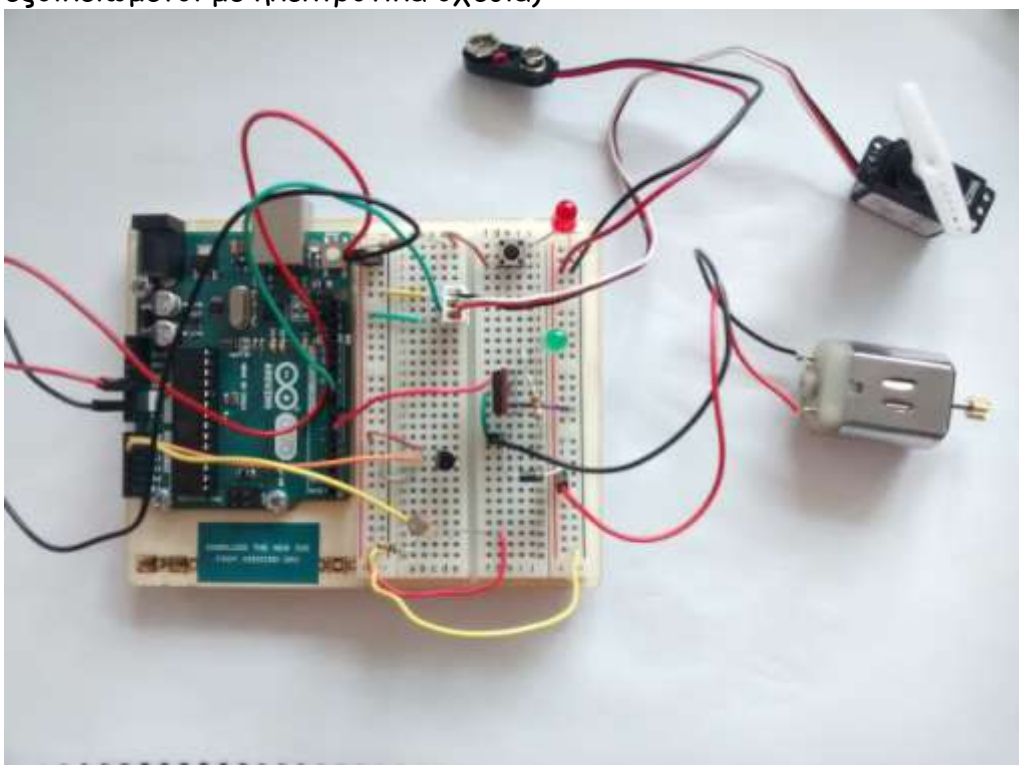
Ανάπτυξη συνοπτικά της θεωρίας για το περιφερειακό νευρικό σύστημα , για τα αισθητικά και κινητικά νεύρα ,για τα αισθητήρια όργανα και τα εκτελεστικά όργανα, πως ενώνονται αυτά με το Κεντρικό νευρικό σύστημα (εγκέφαλος) του ανθρώπου και τα αντανακλαστικά. Εκτιμώμενος χρόνος 20 λεπτά. Για τα επόμενα 15 λεπτά εξήγηση και παρουσίαση των απαραίτητων ηλεκτρονικών εξαρτημάτων για την ανάπτυξη του κυκλώματος πως συνδέονται στην πλακέτα

του Arduino και συσχέτισή του κάθε αισθητήρα με μία ανθρώπινη λειτουργεία.
Α)Την κόρη του οφθαλμού, η οποία διαστέλλεται στο λίγο φως και συστέλλεται στο πολύ με τον συνδυασμό του αισθητήρα φωτός και του σερβομηχανισμού.
Β)Την εφίδρωση του ανθρώπου (η οποία τον ψύχει όταν ζεσταίνεται) με αυτό του αισθητήρα θερμοκρασίας και τον κινητήρα με τη φτερωτή.
Γ) Και τα αντανακλαστικά με τον διακόπτη και το led του οποίου το σήμα δεν περνά από το μικροελεγτή.
Στο υπόλοιπο της ώρας χρόνος για τυχών ερωτήσεις εξηγήσεις.

Δεύτερη διδακτική ώρα:

Παρουσίαση και εξήγηση των απαραίτητων εντολών, υλικών που θα χρησιμοποιηθούν για την υλοποίηση του προγράμματος, κυκλώματος. Εκτιμώμενος χρόνος 15 λεπτά. Στα επόμενα 20 λεπτά δίνονται τα απαραίτητα υλικά στους μαθητές ανά ομάδα τριών και ξεκινούν την υλοποίηση του κυκλώματος και του προγράμματος. Κατά τη διάρκεια της υλοποίησης, στο βιντεοπροβολέα βρίσκεται αναρτημένο το σχέδιο (φωτογραφία) του κυκλώματος ,όπως και το πρόγραμμα που πρέπει να υλοποιήσουν, εάν δεν υπάρχει αυτή η δυνατότητα τότε δίνεται μία σελίδα Α4 στην κάθε ομάδα με το σχέδιο και το πρόγραμμα. Στα τελευταία 10 λεπτά της ώρας για τον έλεγχο της κατανόησης του κυκλώματος αλλά και του προγράμματος ζητείται από τους μαθητές να αλλάξουν τα δεδομένα της θερμοκρασίας για τα οποία ξεκινά το μοτέρ να περιστρέφεται και πάνω από μία θερμοκρασία να σταματά όπως και η λειτουργεία του αισθητήρα φωτός να αντιστραφεί.

Φωτογραφία κυκλώματος
(κρίνεται ποιο εύχρηστή, επειδή οι μαθητές της Β λυκείου δεν είναι εξοικειωμένοι με ηλεκτρονικά σχέδια):



Πρόγραμμα:
(Το πρόγραμμα είναι ενδεικτικό)

```
#include <Servo.h>

Servo myServo;  // create a servo object

int lightSensorValue = 0; // variable to hold the value from the light sensor
const int lightSensorPin = A1;   // pin with the photoresistor
// named constant for the pin the sensor is connected to
const int sensorPin = A0;
// room temperature in Celcius
const float baselineTemp = 20.0;

const int motorPin =  6; // the number of the motor pin
int angle;   // variable to hold the angle for the servo motor

void setup() {
   // initialize the motor pin as an output:
  pinMode(motorPin, OUTPUT);
  // initialize the switch pin as an input:

  myServo.attach(9); // attaches the servo on pin 9 to the servo object
```

```
  Serial.begin(9600); // open a serial connection to your computer
}

void loop() {
  lightSensorValue = analogRead(lightSensorPin);
  // read the value on AnalogIn pin 0
  // and store it in a variable
  int sensorVal = analogRead(sensorPin);
   // send the 10-bit sensor value out the serial port
  Serial.print("temprature sensor Value: ");
  Serial.print(sensorVal);
  //Serial.println("              ");
  // convert the ADC reading to voltage
  float voltage = (sensorVal / 1024.0) * 5.0;

  // Send the voltage level out the Serial port
  Serial.print(", Volts: ");
  Serial.print(voltage);
    // convert the voltage to temperature in degrees C
  // the sensor changes 10 mV per degree
  // the datasheet says there's a 500 mV offset
  // ((volatge - 500mV) times 100)
  Serial.print(", degrees C: ");
  float temperature = (voltage - .5) * 100;
  Serial.println(temperature);
  Serial.println("     ----------         ");

  if (temperature < baselineTemp + 2) {
     // turn motor off:
    digitalWrite(motorPin, LOW);
  }
  else {
    // turn motor on:
    digitalWrite(motorPin, HIGH);
  }

  // print out the value to the serial monitor from the light sensor
  Serial.print("lightSensorValue : ");
  Serial.print(lightSensorValue);

  // scale the numbers from the pot
```

```
  angle = map(lightSensorValue, 0, 1023, 179, 0);

  // print out the angle for the servo motor
  Serial.print(", angle: ");
  Serial.println(angle);
  Serial.println("      -------------      ");
  // set the servo position
  myServo.write(angle);

  // wait for the servo to get there
  delay(2000);
}
```

Καραλέκας Γεώργιος

3ο Γενικό Λύκειο Βέροιας
Οδός Πλουτάρχου ΒΕΡΟΙΑ Ν. ΗΜΑΘΙΑΣ 59100 ΕΛΛΑΔΑ
+30 2331028804
+30 2331075804
mail@3lyk-veroias.ima.sch.gr

Σχολική Μονάδα: 1ο ΕΠΑΛ Βόλου

Συμμετέχοντες Καθηγητές

ΓΚΙΚΑΣ ΑΘΑΝΑΣΙΟΣ

ΣΧΕΔΙΟ ΜΑΘΗΜΑΤΟΣ ΨΗΦΙΑΚΩΝ ΗΛΕΚΤΡΟΝΙΚΩΝ ΓΙΑ ΠΑΙΔΙΑ ΓΥΜΝΑΣΙΟΥ

ΚΑΤΑΣΚΕΥΗ ΑΥΤΟΜΑΤΟΥ ΦΩΤΟΣ ΝΥΧΤΟΣ ΜΕ ΤΟΝ ΜΙΚΡΟΕΛΕΓΚΤΗ ARDUINO

ΔΙΑΡΚΕΙΑ: 2 ώρες
ΚΑΘΗΓΗΤΗΣ: Αθανάσιος Γκίκας

1. **Σκοπός**
   Σκοπός της εργαστηριακής άσκησης είναι η δημιουργία ενός αυτόματου φωτός, χρησιμοποιώντας τον μικροελεγκτή Arduino, μια φωτοαντίσταση, ένα led 5mm, τις απαραίτητες αντιστάσεις και καλωδιώσεις.

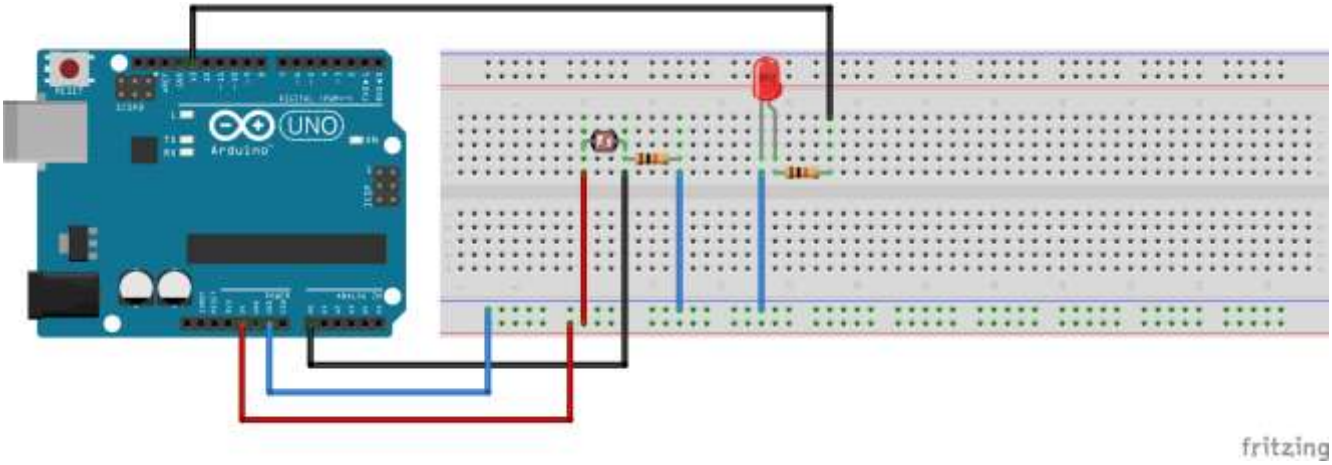
2. **Διδακτικοί στόχοι**
   Οι διδακτικοί στόχοι είναι:
   - Να κατανοήσουν οι μαθητές τις έννοιες της φωτοαντίστασης και της αντίστασης
   - Να κατανοήσουν οι μαθητές πώς δημιουργείται ένα κύκλωμα με τη χρήση του μικροελεγκτή Arduino
   - Να κάνουν οι μαθητές χρήση της γλώσσας προγραμματισμού και του περιβάλλοντος προγραμματισμού του Arduino
   - Να είναι σε θέση να εκτιμήσουν τη σημασία αλλαγών στον κώδικα του προγράμματος

3. **Εκπαιδευτικοί στόχοι**
   Οι εκπαιδευτικοί στόχοι είναι:
   - Να συνεργαστούν οι μαθητές και να δουλέψουν ομαδικά
   - Οι μαθητές να πειραματιστούν και μέσα από τις κατάλληλες μεταξύ τους συζητήσεις να καταλήξουν στην πιο αποδοτική λύση για την κατασκευή τους
   - Να εξοικειωθούν ο μαθητές με την ανάπτυξη στρατηγικών επίλυσης προβλημάτων

4. **Υλικά διδασκαλίας**
   - Μικροελεγκτής Arduino
   - Breadboard
   - Φωτοαντίσταση
   - Αντιστάσεις 220Ω και 10kΩ
   - Led 5 mm
   - Καλώδιο USB για σύνδεση με τον Η/Υ
   - Η/Υ με διαθέσιμη USB θύρα

5. **Εκπαιδευτική διαδικασία**
   Τα βήματα διδασκαλία που θα ακολουθηθούν είναι:

- Αρχικά πραγματοποιείται η παρουσίαση της φωτοαντίστασης και του τρόπου λειτουργίας της. Παρουσιάζεται επίσης η αρχή λειτουργίας των αντιστάσεων και εξηγείται ο ρόλος τους σε ένα κύκλωμα.
- Κατόπιν πραγματοποιείται η συνδεσμολογία, ως εξής: Συνδέουμε τη φωτοαντίσταση σε σειρά με την αντίσταση των 10kΩ. Τον ελεύθερο ακροδέκτη της φωτοαντίστασης τον συνδέουμε στα 5V του Arduino (κόκκινο καλώδιο) και τον ελεύθερο ακροδέκτη της αντίστασης τον συνδέουμε στη γείωση (GND – μπλε καλώδιο). Συνδέουμε τον κόμβο της φωτοαντίστασης με την αναλογική θύρα εισόδου A0 (μαύρο καλώδιο). Το Led συνδέεται στην ψηφιακή θύρα 13 μέσω της αντίστασης 220Ω. Η συνδεσμολογία παρουσιάζεται στο παρακάτω σχήμα:



- Στη συνέχεια γράφουμε τον κώδικα, τον οποίο φορτώνουμε στο Arduino, με διαδικασίες που έχουμε διδαχθεί. Προαιρετικά υπάρχει η δυνατότητα να βλέπουμε τις τιμές της φωτεινότητας στον Η/Υ, μέσω της σειριακής θύρας επικοινωνίας. Ο κώδικας δίνεται στη συνέχεια.
- Μόλις ολοκληρωθεί η κατασκευή και ο προγραμματισμός, οι μαθητές θα ελέγξουν την ορθή λειτουργία του κυκλώματος και θα πειραματιστούν με τις τιμές της έντασης φωτεινότητας, παρακολουθώντας παράλληλα τις τρέχουσες τιμές στη σειριακή οθόνη.

Seminar_Sketch | Arduino 1.6.7

Αρχείο Επεξεργασία Σχέδιο Εργαλεία Βοήθεια

Seminar_Sketch

```
/* Εκπαιδευτική άσκηση: Αυτόματο φωτάκι νύχτας
 * Σχέδιο μαθήματος στα πλαίσια Σεμιναρίου ΕΑΠ
 * Ιανουάριος 2016 */

/* Εκχώρηση τιμών στις μεταβλητές */

int PhotoResistorPin = A0;
int ledPin = 13;
int lightInt;


void setup() {
 /* Προαιρετικό τμήμα κώδικα: Άνοιγμα σειριακής επικοινωνίας
  * για να βλέπουμε τις τιμές της φωτεινότητας στον Η/Υ  */

Serial.begin(9600);

// Ορισμός της θύρας ledPin(13)ως θύρα εξόδου

pinMode(ledPin, OUTPUT);
}

void loop() {
// Διάβασμα αναλογικής θύρας

 lightInt = analogRead(PhotoResistorPin);

// Εκτύπωση στη σειριακή οθόνη

 Serial.println(lightInt);

// Αν η ένταση είναι <500 τότε άναψε το led, αλλιώς σβήστο

  if(lightINT < 500) {
    digitalWrite(ledPIN, HIGH);
  }
 else {
  digitalWrite(ledPIN, LOW);
  }

 delay(200);
  }
```

Σχολική Μονάδα: ΕΠΑΛ Διδυμότειχου

Συμμετέχοντες Καθηγητές

ΚΑΨΙΔΗΣ ΙΩΑΝΝΗΣ
ΤΣΙΟΡΑΛΗ ΘΕΟΠΟΥΛΑ
ΤΑΣΙΔΟΥ ΔΗΜΗΤΡΑ

ΕΠΑΛ Διδυμοτείχου

Σχέδιο μαθήματος

Κατασκευή προγραμματιζόμενου αναλογικού αρθρώματος για τη χρήση με Προγραμματιζόμενο Λογικό Ελεγκτή (PLC) ή αυτόνομα.

Το σχέδιο μαθήματος χωρίζεται σε δυο μέρη που φαίνονται στα δυο αρχεία που επισυνάπτονται.

Το ένα αρχείο έχει ένα σχέδιο μαθήματος 4 ωρών περίπου όπου μαθητές από τον τομέα ηλεκτρολογίας κατασκευάζουν ένα relay module και το συνδέουν με τον μικροελεγκτή εφαρμόζοντας γνώσεις που απέκτησαν σχετικά με τα υλικά που χρησιμοποιούνται για την κατασκευή.

Το άλλο αρχείο προέρχεται από τον τομέα πληροφορικής το οποίο έχει να κάνει με τον προγραμματισμό του μικροελεγκτή και εφαρμογή αντίστοιχης ύλης του τομέα.

Οι μαθητές έρχονται σε επαφή με διαφορετικά αντικείμενα από άλλους τομείς και μαθαίνουν να συνεργάζονται.

Επισυνάπτεται επίσης διεύθυνση με βίντεο από την προσομοίωση της κατασκευής
https://www.youtube.com/watch?v=rfBGm1xZb2c

ΣΧΕΔΙΟ ΜΑΘΗΜΑΤΟΣ

ΣΤΟΙΧΕΙΑ ΑΣΚΗΣΗΣ
Επώνυμο:Καψίδης_____ Όνομα: Δημήτριος Ιωάννης
Σχολείο: 1ο ΕΠΑ.Λ. Διδυμοτείχου_____Τμήμα:**Β ηλεκτρ**
Είδος Διδασκαλίας: ΘΕΩΡΙΑ και ΕΡΓΑΣΤΗΡΙΟ
Ημ/νία _____

ΣΤΟΙΧΕΙΑ ΜΑΘΗΜΑΤΟΣ

1. Μάθημα: **ΑΥΤΟΜΑΤΙΣΜΟΙ ΚΑΙ ΣΤΟΙΧΕΙΑ ΗΛΕΚΤΡΟΝΙΚΗΣ_**
2. Τίτλος Ενότητας: **Ηλεκτρονόμοι**
3. Σκοποί:

- Να κατανοήσουν τη βασική λειτουργία της διόδου σε ορθή και ανάστροφη τάση
- Να γνωρίσουν τη χαρακτηριστική καμπύλη της διόδου
- Να γνωρίσουν τις περιοχές λειτουργίας ενός τρανζίστορ
- Να αναφέρουν τα μέρη από τα οποία αποτελείται ένας ηλεκτρονόμος
- Να εξηγούν τη λειτουργία ενός ηλεκτρονόμου
- Να αναγνωρίζουν τα στοιχεία που αποτελούν ένα κύκλωμα για τη λειτουργία ενός ηλεκτρονόμου σε συνδυασμό με μια ψηφιακή έξοδο και εξηγούν

4. Βοηθήματα-Βιβλιογραφία:

- ΣΥΣΤΗΜΑΤΑ ΑΥΤΟΜΑΤΙΣΜΩΝ, Ζούλης Ν., Καφφετζάκης Π., Σούλτης Γ., 2000, ΛΙΒΑΝΗΣ, ΣΕΛ. 88-95
- ΣΤΟΙΧΕΙΑ ΗΛΕΚΤΡΟΝΙΚΗΣ, Βαρζάκας Π., Πάσχος Ι., Τσελέκας Πολυδ., 2008, ΟΕΔΒ, σελ. 49-55, 61-65, 160-165, 172-176.

ΠΟΡΕΙΑ ΜΑΘΗΜΑΤΟΣ

ΕΦΑΡΜΟΓΗ ( )
1. ΠΡΟΕΤΟΙΜΑΣΙΑ (περίπου 5 λεπτά)

1η ώρα: Γίνεται συζήτηση για στοιχεία που αφήνουν το ρεύμα να περάσει προς τη μια κατεύθυνση αλλά όχι προς την άλλη (σαν βαλβίδες).

2η ώρα: αναρωτιόμαστε αν υπάρχουν διακόπτες χωρίς κινητά μέρη

3η ώρα: Παρατηρούμε τι συμβαίνει όταν ένας ηλεκτρονόμος συνδέεται σε ψηφιακή έξοδο και αναρωτιόμαστε τι συμβαίνει και πώς λύνεται.

4η ώρα: Επανάληψη πειράματος και υπενθύμιση προβλήματος

2. ΠΑΡΟΥΣΙΑΣΗ (περίπου 30 λεπτά)

1η ώρα:
Περιεχόμενο – Βαθμίδες
Εξηγείται η λειτουργία της διόδου

Μέθοδοι
Προσομοίωση με κατάλληλο λογισμικό

Μέσα
Λογισμικό προσομοίωσης
Βιντεοπροβολέας
Δίοδοι διάφορων μορφών για επίδειξη του υλικού

2η ώρα:
Περιεχόμενο – Βαθμίδες
Εξηγείται η λειτουργία του τρανζίστορ

Μέθοδοι
Προσομοίωση κυκλωμάτων

Μέσα
Λογισμικό προσομοίωσης
Βιντεοπροβολέας

3η ώρα:
Περιεχόμενο – Βαθμίδες
Εξηγείται η λειτουργία του ηλεκτρονόμου.

Μέθοδοι
Επίδειξη των τμημάτων του υλικού και επίδειξη κυκλώματος για την κατανόηση της λειτουργίας του.
Επίδειξη του τρόπου λειτουργίας με πηγές τάσης.
Σύνδεση του πηνίου του ηλεκτρονόμου σε ψηφιακή έξοδο μικροεπεξεργαστή ή μικροελεγκτή.

Μέσα
Ηλεκτρονόμος, breadboard, πηγή τάσης (μπαταρίες), Διακόπτης, λαμπάκι ή LED και αντίσταση, πλακέτα μικροεπεξεργαστή

4η ώρα:
Περιεχόμενο – Βαθμίδες
Κατασκευή και δοκιμή του relay module στο εργαστήριο.

Μέσα
Υλικά για την κατασκευή του relay module όπως αντιστάσεις, τρανζίστορ, διόδοι, breadboard ή/και διάτρητες πλακέτες καθώς και τα σχέδια στα οποία θα καταλήξουν οι μαθητές μετά από έρευνα. Για την τελική κατασκευή απαιτούνται και σταθμοί συγκόλλησης (κολλητήρια και καλάι).

3. ΕΦΑΡΜΟΓΗ
1η ώρα: Δίνονται κατά τη διάρκεια της παρουσίασης το φύλλο εφαρμογής και οι μαθητές το συμπληρώνουν. Υπάρχουν και 1-2 ερωτήσεις στις οποίες οι μαθητές πρέπει να απαντήσουν αφού ανατρέξουν στο σχολικο εγχειρίδιο.

2η ώρα: Δίνονται στους μαθητές ερωτήσεις αντιστοίχισης και επιλογής ώστε να συζητηθούν τυχόν παρανοήσεις σχετικά με τη λειτουργία και τη χρήση του τρανζίστορ τουλάχιστον για τις περιπτώσεις που μας ενδιαφέρουν.

3η ώρα: Η εφαρμογή και η επανάληψη θα γίνει με την επίδοση του σχεδίου στο οποίο έχουμε καταλήξει μετά από μελέτη και την έναρξη της κατασκευής που θα ολοκληρωθεί και θα δοκιμαστεί την επόμενη ώρα.

4. ΕΛΕΓΧΟΣ (5)
1η ώρα: Ζητείται από τους μαθητές να υποδείξουν σε φύλλο ελέγχου αν τα κάποια διαγράμματα αναφέρονται σε δίοδο, και μεταξύ κυκλωμάτων σε ποιά από αυτά η δίοδος είναι ορθά και σε ποια ανάστροφα πολωμένη.

2η ώρα: Δίνονται ερωτήσεις πολλαπλής επιλογής και σχήματα στα οποία θα προστεθούν από τους μαθητές ετικέτες σχετικά με τις περιοχές λειτουργίας του τρανζίστορ.

3η ώρα: Ελέγχεται η κατανόηση του οπλισμού ενός ηλεκτρονόμου και η τροφοδοσία συσκευών μέσω των επαφών του.

Την 3η και 4η ώρα οι μαθητές θα κατασκευάσουν το module και θα ελεγχθεί η σωστή λειτουργία του. Ο έλεγχος θα γίνει με τη σύνδεση των modules στο Arduino και την εκτέλεση του ανάλογου λογισμικού. Ο προγραμματισμός του Arduino θα γίνει σε συνεργασία με τον τομέα πληροφορικής ώστε να έρθουν σε επαφή και να συνεργαστούν μαθητές από τις δυο ειδικότητες.

ΣΧΕΔΙΟ ΜΑΘΗΜΑΤΟΣ

****

ΟΝΟΜΑΤΕΠΩΝΥΜΑ : ΤΑΣΙΔΟΥ ΔΗΜΗΤΡΑ –ΤΣΙΟΡΑΛΗ ΘΕΟΠΟΥΛΑ

Σχολείο: **1ο ΕΠΑ.Λ. ΔΙΔΥΜΟΤΕΙΧΟΥ** Τμήμα: Γ Πληροφορικής

Είδος Διδασκαλίας: ΘΕΩΡΙΑ ΚΑΙ ΕΡΓΑΣΤΗΡΙΟ

Ημ/νία _____

ΣΤΟΙΧΕΙΑ ΜΑΘΗΜΑΤΟΣ

1. **Μάθημα**: ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΣ ΥΠΟΛΟΓΙΣΤΩΝ
2. **Τίτλος Ενότητας**: ΔΟΜΗ ΕΠΙΛΟΓΗΣ
3. **Σκοποί**:
   - Να κατανοήσουν οι μαθητές τις δυνατότητες της αλλαγής της διαδοχικής εκτέλεσης των εντολών ενός προγράμματος ανάλογα με τα αποτελέσματα συνθηκών που περιλαμβάνονται στις ίδιες τις εντολές
   - Να αναγνωρίζουν και να εφαρμόζουν λογικές εκφράσεις, απλές και σύνθετες.
   - Να κατανοήσουν τη σύνταξη και λειτουργία της εντολής if σε απλή και σύνθετη μορφή
   - Να εφαρμόσουν τις γνώσεις τους σε πραγματικές συνθήκες με τη χρήση του μικροελεγκτή ARDUINO

4. **Βοηθήματα-Βιβλιογραφία:**

   - Η Γλώσσα Προγραμματισμού C, Brian Kernighan, Dennis Ritchie, 1988, σελ. 85-91
   - Προγραμματίζοντας με τον μικροελεγκτή Arduino, Εμμανουήλ Πουλάκης, 2015, σελ. 16
   - http:// www.ebooks4greeks.gr/εισαγωγή-στη-c, Συγγραφέας: Γιάννης Τσιομπίκας, Κωνσταντίνος Μαργαρίτης , 2013, σελ. 8

5. **Εποπτικά Μέσα:**

Πίνακας
Βιντεοπροβολέας
Διαφάνειες
Ηλεκτρονικοί υπολογιστές

6. **Απαιτούμενο Λογισμικό**

Dev-C/C++
Arduino IDE
Δημιουργός Διαγραμμάτων Ροής

ΠΟΡΕΙΑ ΜΑΘΗΜΑΤΟΣ

1. **ΠΡΟΕΤΟΙΜΑΣΙΑ (περίπου 5 λεπτά)**
   1η ώρα: Καταιγισμός ιδεών σχετικά με παραδείγματα της καθημερινότητας όπου οι συνθήκες οδηγούν στη λήψη των ανάλογων αποφάσεων (πχ. φανάρια, βαθμολογία - χαρακτηρισμός επίδοσης κ.α.)
   2η ώρα: Επίδειξη διαγράμματος ροής και προβληματισμός για τα διαφορετικά σχήματα.
   3η ώρα: Υπενθύμιση των δομών προγραμματισμού και εστίαση στη δομή επιλογής.

2. **ΠΑΡΟΥΣΙΑΣΗ (περίπου 20 λεπτά)**

ΩΡΕΣ	ΠΕΡΙΕΧΟΜΕΝΟ-ΒΑΘΜΙΔΕΣ	ΜΕΘΟΔΟΙ	ΜΕΣΑ
1η	Εξηγούνται οι λειτουργίες και τα σύμβολα των σχεσιακών και λογικών τελεστών.	Προερωτήσεις Εισήγηση (λειτουργίες-παραδείγματα) Φύλλο εργασίας	Βιντεοπροβολέας
2η	Παρουσιάζεται η δομή επιλογής, αρχικά με ψευδοκώδικα και μετά μέσω διαγραμμάτων ροής	Ερωτήσεις - Απαντήσεις Διερεύνηση παραδειγμάτων από την καθημερινή ζωή Συζήτηση	Πίνακας Αντίστοιχο Λογισμικό
3η	Σύνταξη Δομής επιλογής if στη γλώσσα C	Ιστοεξερεύνηση - Πειραματισμός	Πίνακας Διαφάνειες
4η	Επίδειξη περιβάλλοντος και εντολών ARDUINO. Αναφορά στις απαραίτητες βιβλιοθήκες. Εφαρμογές της εντολής if με την τεχνική 'Μαύρο Κουτί'	Μελέτη Περίπτωσης. Πειραματισμός - Συνεργασία	Πίνακας

3. **ΕΦΑΡΜΟΓΗ (περίπου 10 λεπτά)**

1η ώρα: Δίνεται Φύλλο εργασιών σε κάθε μαθητή με ασκήσεις που αφορούν στους λογικούς - σχεσιακούς τελεστές.

2η ώρα: Οι μαθητές χωρίζονται σε ζευγάρια και τους ζητείται να πειραματιστούν, μετατρέποντας έτοιμο ψευδοκώδικα σε διάγραμμα ροής, στο λογισμικό 'Δημιουργός Διαγραμμάτων Ροής'.

3η ώρα: Κωδικοποίηση του προηγούμενου αλγορίθμου στη γλώσσα προγραμματισμού C, στο περιβάλλον DEV-C/C++. Παρακολούθηση ροής εκτέλεσης εισάγοντας εικονικά δεδομένα και εξάγοντας χειρωνακτικά τα αποτελέσματα. Εκτέλεση του κώδικα με τα ίδια δεδομένα στο περιβάλλον προγραμματισμού και σύγκριση αποτελεσμάτων για τυχόν λογικά λάθη . Εκσφαλμάτωση.

4η ώρα: Εφαρμογή παραδείγματος χρήσης της δομής στο Arduino IDE με έτοιμο αρχείο το οποίο οι μαθητές καλούνται να εκτελέσουν (μαύρο κουτί) στο λογισμικό προσομοίωσης.
Σχεδίαση του κώδικα για τη λειτουργία του relay module. Δοκιμή στο λογισμικό προσομοίωσης. Διασύνδεση του relay module (το οποίο έχει υλοποιηθεί από το τμήμα των Ηλεκτρολόγων) στον υπολογιστή και εκτέλεση του κώδικα.

4. **ΕΛΕΓΧΟΣ(περίπου 5 λεπτά)**

1η ώρα: Οι μαθητές ανταλλάσσουν τα φύλλα εργασιών μεταξύ τους ώστε να γίνει διόρθωση από τους ίδιους. Δίνονται οι απαραίτητες επεξηγήσεις.

2η ώρα: Δίνεται φύλλο ελέγχου όπου ζητείται από τους μαθητές να σχεδιάσουν αλγόριθμο της δομής επιλογής σε ψευδοκώδικα ή διάγραμμα ροής

3η ώρα: Συμπεράσματα - Αξιολόγηση αλγορίθμου.

4η ώρα: Έλεγχος της λειτουργίας του relay module σε συνεργασία με τον μικροελεγκτή.

Σχολική Μονάδα: Γυμνάσιο με Λυκειακές Τάξεις Καλλιθέας Ελασσόνας

Συμμετέχοντες Καθηγητές

ΤΑΝΟΣ ΗΛΙΑΣ
ΟΡΦΑΝΑΚΗΣ ΣΤΕΛΙΟΣ

Διδακτικό Σενάριο

Θεματική περιοχή: **Αέρια ατμόσφαιρας**
Τόπος εκτέλεσης σεναρίου: **Γυμνάσιο με Λυκειακές Τάξεις Καλλιθέας**
Τάξη εκτέλεσης σεναρίου: **Β Λυκείου Γενικής – (***στοιχεία διδάσκοντα:* **Ορφανάκης Στυλιανός)**

Δημιουργοί σεναρίου:
Ορφανάκης Στυλιανός ΠΕ 12.10 & ΠΕ 04.01
&
Τανός Ηλίας ΠΕ 12.10 & ΠΕ 04.01

Τίτλος διδακτικού σεναρίου:

Μέτρηση διοξειδίου του άνθρακα (CO_2), με τη χρήση του Arduino

Εκτιμώμενη διάρκεια διδακτικού σεναρίου

Η χρονική διάρκεια εξαρτάται από την βαθμίδα εκπαίδευσης όπου εφαρμόζεται. Η θεματολογία του σεναρίου μπορεί να υποστηρίξει δραστηριότητες από δύο (2) μέχρι και τρεις (3) διδακτικές ώρες, χωρίς να αποκλείεται υπέρβαση της παραπάνω διάρκειας, αν η φύση του μαθήματος, στο πλαίσιο του οποίου πραγματοποιείται, το επιτρέψει.

Ένταξη του διδακτικού σεναρίου στο πρόγραμμα σπουδών/προαπαιτούμενες γνώσεις

Το διδακτικό σενάριο αποτελεί μια διεπιστημονική προσέγγιση θεμάτων που εντάσσονται σε αρκετά γνωστικά αντικείμενα της δευτεροβάθμιας εκπαίδευσης. Δεν χρειάζονται ιδιαίτερες προαπαιτούμενες γνώσεις. Το σενάριο προσαρμόζεται ανάλογα με τη βαθμίδα εκπαίδευσης που εφαρμόζεται ή και το μάθημα στο πλαίσιο του οποίου πραγματοποιείται. Και φυσικά λαμβάνει υπόψη το γνωστικό επίπεδο των μαθητών/τριων που εμπλέκονται συνεργατικά για την υλοποίηση του.

Σκοποί και στόχοι του διδακτικού σεναρίου

Η συγκεκριμένη διδακτική πρακτική αποτελεί μέρος μιας ολοκληρωμένης διδακτικής προσέγγισης στη δευτεροβάθμια βαθμίδα εκπαίδευσης που στηρίζεται στην συνεργατική ανακαλυπτική μέθοδο με τη χρήση της ερευνητικής μεθοδολογίας. Το συγκεκριμένο μοντέλο – διδακτικό σενάριο φιλοδοξεί αφενός να χρησιμοποιηθεί από τους/ις μαθητές/ριες ως

ερευνητικό και ερμηνευτικό εργαλείο για τα φαινόμενα του φυσικού και τεχνολογικού κόσμου, αφετέρου να τους/ις βοηθήσει να κατανοήσουν τα προβλήματα που σχετίζονται με την καθημερινή ζωή και έχουν παγκόσμιο χαρακτήρα.
Οι διδακτικοί στόχοι που τίθενται αφορούν γνώσεις, δεξιότητες και στάσεις. Πιο συγκεκριμένα:
Σε επίπεδο γνώσεων οι μαθητές/τριες στο τέλος του σεναρίου θα πρέπει να :
- αναγνωρίζουν τα αέρια και να διατυπώνουν τις χημικές και φυσικές ιδιότητες τους.
- αναγνωρίζουν τις ανθρώπινες και φυσικές δραστηριότητες που συμβάλλουν στην ένταση της ύπαρξης τους στο φυσικό περιβάλλον
- αναγνωρίζουν και να διατυπώνουν τρόπους περιορισμού αυτών αερίων
- αναγνωρίζουν και να περιγράφουν τον τρόπο λειτουργίας των αισθητήρων μέτρησης των αερίων που στηρίζονται στον μικροελεγκτή Arduino.
- αναγνωρίζουν και να περιγράφουν τα βασικά ηλεκτρονικά στοιχεία που χρησιμοποιούνται στην κατασκευή των αισθητήρων.
- προγραμματίζουν σε βασικό επίπεδο τον μικροελεγκτή Arduino.

Σε επίπεδο δεξιοτήτων θα πρέπει να :
- αναζητούν αποτελεσματικά πληροφορίες από το διαδίκτυο
- συναρμολογούν τις απαραίτητες διατάξεις για την μέτρηση των αερίων.
- συνδέουν πειραματικά δεδομένα και συμπεράσματα,
- αναπτύσσουν δεξιότητες στη χρήση λογισμικών στατιστικής επεξεργασίας και παρουσίασης δεδομένων και συμπερασμάτων.
- βαθμονομούν τους διάφορους αισθητήρες
- αναπτύσσουν ικανότητες ελέγχου μεταβλητών κατά την πειραματική εργασία και
- εντάσσουν στην εισήγηση τους προς την ολομέλεια, μέσα σε ένα συγκεκριμένο χρόνο.

Περιγραφή του διδακτικού σεναρίου

Η δραστηριότητα και οι ενέργειες που τη συνθέτουν είναι ενδεικτικές γιατί μπορούν να εμπλουτιστούν ή να περιοριστούν, ανάλογα με τάξη εφαρμογής, το μάθημα εφαρμογής, το γνωστικό επίπεδο των μαθητών κ.λπ.

Χρήση Η.Υ., ΤΠΕ, καθώς και άλλων μέσων για το διδακτικό σενάριο

Στο προτεινόμενο σενάριο κυριαρχεί η χρήση του Η/Υ και του μικροελεγκτή Arduino. Πιο συγκεκριμένα, ο Η/Υ χρησιμοποιείται για την επεξεργασία και παρουσίαση των δεδομένων και αποτελεσμάτων, όπως επίσης και για την αναζήτηση πληροφοριών από το διαδίκτυο. Ακόμη μέσω του Η/Υ ανταλλάσσονται σχετικές πληροφορίες μεταξύ των μελών των ομάδων. Τέλος, η χρήση του μικροελεγκτή Arduino δίνει την ευκαιρία στους μαθητές να έρθουν σε επαφή με τον προγραμματισμό αλλά και να αναγνωρίσουν τον μετασχηματισμό των δεδομένων του φυσικού κόσμου σε ηλεκτρικά σήματα και τις πολλαπλές, μέσω του Η/Υ και του Arduino, αναπαραστάσεις τους.

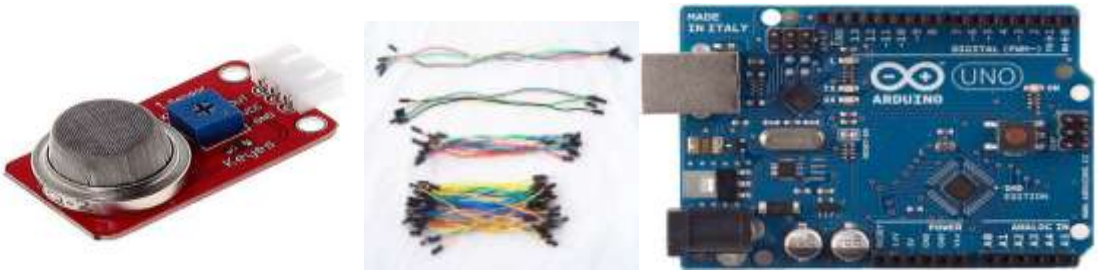
ΠΑΡΟΥΣΙΑΣΗ ΣΕΝΑΡΙΟΥ

ΔΡΑΣΤΗΡΙΟΤΗΤΑ

Α. Θα φτιάξουμε το βασικό κύκλωμα για την ρύθμιση του αισθητήρα MQ-2. Για το σκοπό αυτό θα χρειαστεί να γίνουν οι παρακάτω ενέργειες:

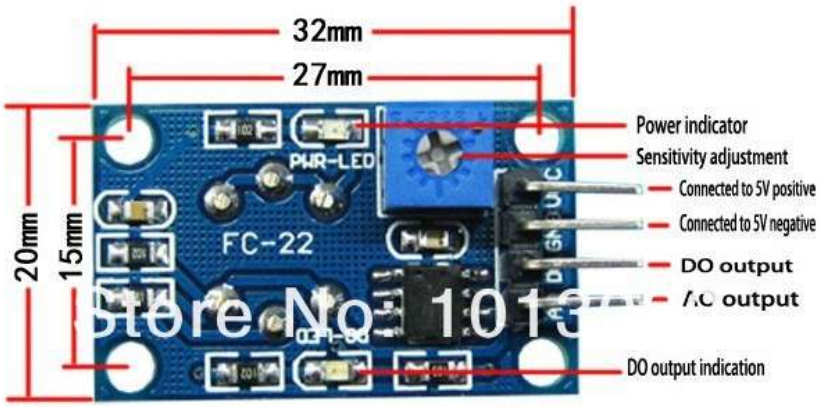
Α.1 Εντοπίστε στο πάγκο εργασίας
- **i)** τον αισθητήρα αερίων MQ-2,
- **ii)** τη πλακέτα Arduino και
- **iii)** καλώδια για τις σχετικές συνδέσεις.

Το αισθητήριο MQ-2 μπορεί να ανιχνεύσει τα εξής αέρια: υγραέριο, βουτάνιο, προπάνιο, μεθάνιο, υδρογόνο και καπνό. Μπορεί ακόμα να ανιχνεύσει και αλκοόλ. Σε κάποια απ' αυτά τα αέρια έχει μεγάλη ευαισθησία (μπορεί να τα εντοπίσει πολύ εύκολα) και σε κάποια άλλα χαμηλή.



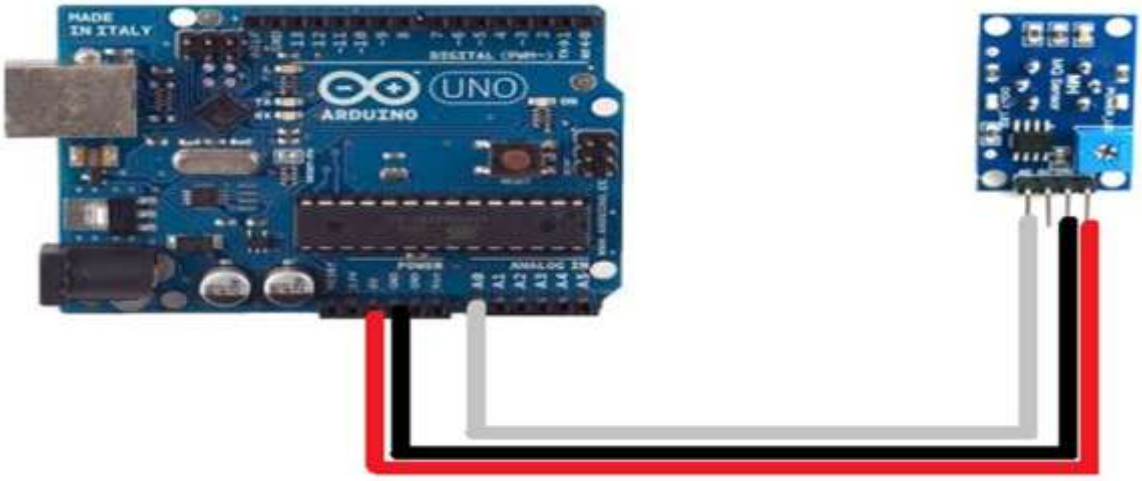
Αισθητήρας MQ-2, καλώδια και μικροελεγκτής Arduino (από αριστερά προς τα δεξιά)

Α.2 Μελετήστε την παρακάτω εικόνα σε σχέση με τον αισθητήρα MQ-2 που έχετε στα χέρια σας ώστε να ξέρετε τη λειτουργία του κάθε ακροδέκτη, να δείτε που είναι ο power indicator (δείκτης λειτουργίας), ο sensitivity adjustment (ρυθμιστικό ευαισθησίας) και DO output indication (δείκτης ψηφιακής εξόδου)



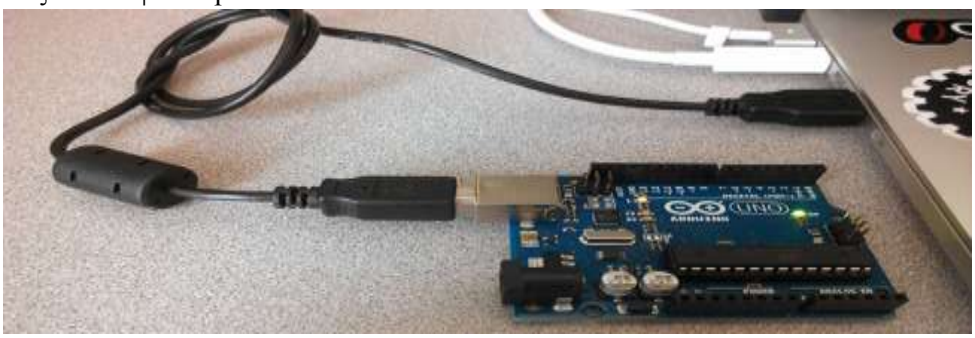
Κύκλωμα οδήγησης MQ-2

Α.3 Συνδέστε τον αισθητήρα MQ-2 με την πλακέτα Arduino όπως φαίνεται στην Εικόνα



Διασύνδεση μικροελεγκτή Arduino με αισθητήρα MQ-2

Α.4 Ανοίξτε το laptop όσοι διαθέτετε ή το desktop που έχετε μπροστά σας και συνδέστε τη USB θύρα του υπολογιστή σας με τη θύρα USB του Arduino για το σκοπό αυτό εντοπίστε το καλώδιο USB που είναι στο πάγκο εργασίας σας. Αν το κάνετε σωστά αμέσως θα ανάψει το power indication του Arduino.



Διασ

ύνδεση μικροελεγκτή Arduino με Η/Υ

Α5. Στην επιφάνεια εργασίας του υπολογιστή σας υπάρχει εικονίδιο με το σήμα του Arduino και την ετικέτα Scetches, πατήστε διπλό κλικ πάνω του και θα σας εμφανιστεί το περιβάλλον όπου μπορεί να γράψετε τον κώδικα που θέλετε να τρέξει. Αντιγράψτε τον παρακάτω κώδικα:

```
/* Testing MQ-2 GAS sensor with serial monitor
   Suitable for detecting of LPG, i-butane, propane, methane, alcohol, Hydrogen or smoke
*/
const int gasPin = A0; //GAS sensor output pin to Arduino analog A0 pin
void setup()
{
      Serial.begin(9600); //Initialize serial port - 9600 bps
}
void loop()
{
      Serial.println(analogRead(gasPin));
      delay(1000); // Print value every 1 sec.
}
```

Στη συνέχεια πατήστε το κουμπί με την ένδειξη verify για να δείτε αν έχει κάποια λάθη και στη συνέχεια το κουμπί upload ώστε να φορτωθεί στο Arduino. Αν δεν υπάρξει πρόβλημα το πράσινο led θα αναβοσβήνει. Στη συνέχεια πατήστε το εικονίδιο με το μεγεθυντικό φακό (Serial monitor) για να έχετε την καταγραφή των μετρούμενων τιμών.

Α6. Αφήστε τον αισθητήρα να καταγράφει για τέσσερα έως πέντε λεπτά τα αέρια που υπάρχουν στο υπόβαθρο. Μόλις συμπληρωθούν τα τέσσερα έως πέντε λεπτά καταγράψτε την ένδειξη που βλέπετε να επαναλαμβάνετε στο μόνιτορ του υπολογιστή σας.

Απάντηση:_____

Β. Θα ανιχνεύσουμε CO_2 για το σκοπό αυτό θα πρέπει να φέρεται στο πάγκο εργασία σας ένα μπουκάλι ισχυρού οξέος εν προκειμένω HCl, κι ένα μπουκάλι με ανθρακικό ορυκτό εν προκειμένω ασβεστόλιθο $CaCO_3$, ένα ψηφιακό χρονόμετρο κι ένα ογκομετρικό κύλινδρο. Αυτή είναι μια από τις πρώτες χημικές αντιδράσεις που διδάσκονται :

$$CaCO_3 + 2\,HCl \longrightarrow CaCl_2 + H_2O + CO_2$$

Β1. Στον ογκομετρικό κύλινδρο ρίχνεται 20ml $CaCO_3$ και 40ml HCl. Άμεσα τοποθετείται τον αισθητήρα MQ-2 πάνω από τον ογκομετρικό κύλινδρο και παρακολουθήστε τις

μετρήσεις όπως καταγράφονται στο μόνιτορ του υπολογιστή σας και συμπληρώστε τον παρακάτω πίνακα σύμφωνα με τις χρονικά διαστήματα σας αναφέρονται στη πρώτη στήλη:

Πίνακας Β1: Μετρήσεις για CO_2

Χρόνος (sec)	**Τιμή συγκέντρωσης CO_2**
0	
15	
30	
50	
70	
90	
120	
150	
180	
210	
240	
270	
300	
330	
360	

Β2. Αφού ανοίξετε τα παράθυρα του εργαστηρίου και ξεπλύνετε τα τον ογκομετρικό κύλινδρο, 10 λεπτά αργότερα επαναλάβετε τα προηγούμενα αλλά χρησιμοποιώντας 40ml $CaCO_3$ και 80ml HCl και παρακολουθήστε τις μετρήσεις όπως καταγράφονται στο μόνιτορ του υπολογιστή σας και συμπληρώστε τον παρακάτω πίνακα σύμφωνα με τις χρονικά διαστήματα σας αναφέρονται στη πρώτη στήλη:

Πίνακας Β2: Μετρήσεις για CO_2

Χρόνος (sec)	**Τιμή συγκέντρωσης CO_2**
0	
15	
30	
50	
70	
90	
120	
150	
180	
210	
240	
270	
300	
330	
360	

Β3. Τι παρατηρείται στις καταγεγραμμένες συγκεντρώσεις του CO2 . Οι μετρήσεις που έχουν προκύψει είναι οι αναμενόμενες με βάση και τη στοιχειομετρική θεωρία;

_____

_____

Βιβλιογραφία:
1. Πρακτικά 1ου Πανελλήνιου Συνεδρίου για την Προώθηση της Εκπαιδευτικής Καινοτομίας – Λάρισα – 2015, Τόμος ΙΙΙ, σελίδα 8, ISBN 978-618-82197-3-1, 978-618-82197-4-8, «Μέτρηση «θερμοκηπικών» αερίων με τη βοήθεια Arduino», συγγραφείς Πάλλας Αναστάσιος – Ορφανάκης Στυλιανός .
2. Fullan, M. (1991). The New Meaning of Educational Change. London: Cassell.
3. Kellett, M. (2005). How to Develop Children as Researchers. London: Paul Chapman.

Σχολική Μονάδα: ΕΚ Θήβας

Συμμετέχοντες Καθηγητές

ΓΙΑΝΝΑΚΟΣ ΝΙΚΟΛΑΟΣ
ΕΥΘΥΜΙΟΥ ΝΙΚΟΛΑΟΣ
ΜΙΧΑΗΛ ΤΡΙΑΔΑ
ΟΙΚΟΝΟΜΙΔΗΣ ΣΑΒΒΑΣ

ΕΡΓΑΣΤΗΡΙΑΚΟ ΚΕΝΤΡΟ ΘΗΒΑΣ

ΤΕΛΙΚΗ ΕΡΓΑΣΙΑ

στο Σεμινάριο εκπαίδευσης Εκπαιδευτικών στο Arduino από το Εργαστήριο Ψηφιακών Συστημάτων και Επεξεργασίας Ψηφιακών Μέσων του Ελληνικού Ανοικτού Πανεπιστημίου.

Συμμετέχοντες Εκπαιδευτικοί :

1. Γιαννακός Νικόλαος ΠΕ19 Πληροφορικής
2. Ευθυμίου Νικόλαος ΠΕ17.06 Μηχανολόγος
3. Μιχαήλ Τριάδα ΠΕ17.04 Ηλεκτρονικός ΑΣΕΤΕΜ
4. Οικονομίδης Σάββας ΠΕ17.02 Μηχανολόγος ΑΣΕΤΕΜ

ΣΧΕΔΙΟ ΜΑΘΗΜΑΤΟΣ

ΠΕΡΙΓΡΑΦΗ ΔΡΑΣΤΗΡΙΟΤΗΤΑΣ

Στο μάθημα αυτό οι μαθητές αντιμετωπίζουν το πρόβλημα της συλλογής φυσικών μεγεθών με την χρήση Αισθητήρων και της οπτικής απεικόνισής τους με τη βοήθεια Υπολογιστή . Θα αναπτύξουν πρόγραμμα ανάγνωσης τη Πίεσης με τη χρήση μιας πλακέτας Arduino και θα μεταφέρουν τα δεδομένα σε υπολογιστή για εμφάνιση σε μορφή γραφικών.

ΠΕΡΙΛΗΨΗ

Στο μάθημα αυτό διερευνάται το πώς μια συνεργασία λογισμικού και ηλεκτρονικού εξοπλισμού με τη χρήση Μικροελεγκτών μπορεί να αξιοποιήσει μετρήσεις Πίεσης και με τη χρήση Υπολογιστή να αξιοποιηθούν οι μετρήσεις και να οπτικοποιηθούν με τη χρήση γραφικών στοιχείων ώστε να είναι περισσότερο αντιληπτές και εύληπτες στο χρήστη.

Οι μαθητές εργάζονται σε ομάδες για να σχεδιάσουν τη διάταξη στο περιβάλλον Fritzing και προγραμματίσουν μια πλακέτα Arduino, αξιοποιώντας μια Αναλογική Είσοδο. Θα δημιουργήσουν το πρόγραμμα ανάγνωσης και εμφάνισης σε οθόνη LCD και Υπολογιστή. Παράλληλα θα διασυνδέσουν μέσω της **Processing** τα δεδομένα που θα συλλέξουν ώστε να εμφανίζονται με γραφικό τρόπο. Τέλος θα παρουσιάσουν την εργασία στη τάξη .

ΕΚΠΑΙΔΕΥΤΙΚΟΙ ΣΤΟΧΟΙ

Γνώσεις

- Να γνωρίσουν την χρήση των Αναλογικών Θυρών του Arduino και τον τρόπο ανάπτυξης κώδικα για την ανάγνωση των τιμών από αυτήν.
- Να περιγράφουν ένα ηλεκτρονικό κύκλωμα ανάγνωσης τιμών φυσικών μεγεθών.

Ικανότητες

- Να μάθουν να σχεδιάζουν κυκλώματα με τη χρήση Λογισμικού Σχεδίασης Ηλεκτρονικών διατάξεων.
- Να κατασκευάζουν το κύκλωμα χρησιμοποιώντας τα κατάλληλα εξαρτήματα και όργανα.
- Να συντάσσουν Κώδικα για την ανάγνωση και εμφάνιση δεδομένων σε οθόνη LCD
- Να συντάσσουν μια απλή εφαρμογή σε Processing.

Στάσεις

- Να διερωτώνται πώς η τεχνολογία μπορεί να βοηθήσει στην επίλυση καθημερινών προβλημάτων.
- Να υποστηρίζουν και ενθαρρύνουν στα πλαίσια της συνεργασίας και αλληλοβοήθειας τα άλλα μέλη της ομάδας.

ΑΝΑΜΕΝΟΜΕΝΑ ΜΑΘΗΣΙΑΚΑ ΑΠΟΤΕΛΕΣΜΑΤΑ

Ως Μαθησιακά Αποτελέσματα αυτής της δραστηριότητας, οι μαθητές πρέπει να αναπτύξουν δεξιότητες : Σχεδιασμού Ηλεκτρονικών Κυκλωμάτων, Ανάπτυξης λογισμικού για την ανάγνωση τιμών από την Αναλογική είσοδο και την εμφάνιση στην Οθόνη .Να καλλιεργήσουν ικανότητες Εργασίας σε Ομάδες.

ΔΙΑΡΚΕΙΑ

6 Εργαστηριακές Ώρες

ΔΡΑΣΤΗΡΙΟΤΗΤΕΣ ΜΑΘΗΜΑΤΟΣ

Οι μαθητές εργαζόμενοι σε ομάδες , θα αναζητήσουν πως οι μηχανικοί λογισμού και ηλεκτρονικοί μηχανικοί έχουν λύσει προβλήματα ανάγνωσης και παρουσίασης φυσικών μεγεθών (πχ. Πίεση Θερμοκρασία , Φωτεινότητα ,στάθμη δεξαμενών κλπ)

Θα εξετάσουν τη δυνατότητα αξιοποίησης του Arduino στην επίλυση των ανωτέρω προβλημάτων με τη χρήση των Αναλογικών Εισόδων.

Θα αναπτύξουν πρόγραμμα για την ανάγνωση των τιμών πίεσης από αισθητήρα πίεσης.

Θα αναπτύξουν πρόγραμμα για την εμφάνιση των ανωτέρω τιμών πίεσης σε οθόνη LCD και

Θα Χρησιμοποιήσουν τη **Processing** για την μεταφορά των τιμών από το Arduino σε Υπολογιστή και την απεικόνιση τους με γραφικά.

ΥΛΙΚΑ και ΜΕΣΑ - ΠΗΓΕΣ ΠΛΗΡΟΦΟΡΙΩΝ

Πλακέτα Arduino Uno, οθόνη i2c 4χ20, Breadboard ,Καλώδια Σύνδεσης , Αισθητήρας πίεσης MPX4250 της Freescale ,Ηλ. Υπολογιστής, Ίντερνετ ,Φύλλο Έργου , Φύλλο Πληροφοριών, Arduino IDE, Fritzing, Processing

Βιβλίο: Πουλάκης , Ε. (2015). *Προγραμματίζοντας με τον μικροελεγκτή Arduino.*
Ε. Πουλάκης: Ηράκλειο

ΧΡΗΣΗ INTERNET

Σύνδεση με Ιντερνέτ - Αναζήτηση Λημμάτων : “Data Acquisition “ και ”Data Acquisition using Arduino”

ΠΑΡΑΔΟΤΕΑ :

Σχέδιο Σύνδεσης εξαρτημάτων στο Fritzing, Κύκλωμα με συνδεδεμένο τον αισθητήρα Πίεσης και την οθόνη σε πλήρη λειτουργία ,Κώδικας για την ανάγνωση τιμών από την αναλογική είσοδο και προβολή σε LCD, αναπαράσταση σε γραφικό στοιχείο σε υπολογιστή των ανωτέρω τιμών.
Προαιρετικά : Χρήση αισθητήρων άλλων φυσικών μεγεθών.(πχ υγρασία ,ταχύτητα άνεμου ,θερμοκρασία κλπ)

ΚΡΙΤΗΡΙΑ ΑΠΟΤΙΜΗΣΗΣ :

Σχέδιο . Σκαρίφημα και Fritzing : 5 Βαθμοί
Ηλεκτρονικό Κύκλωμα Aduino , :5 Βαθμοί
Ανάπτυξη Κώδικα : 7 Βαθμοί
Ερωτήσεις έλεγχου διδασκαλίας : 3 Βαθμοί

Ακλουθούν :

- ΦΥΛΛΟ ΠΛΗΡΟΦΟΡΙΩΝ (Δίνεται από τον Εκπαιδευτικό)
- ΦΥΛΛΟ ΕΡΓΟΥ (Παραδοτέα Μαθητων)

ΕΡΓΑΣΤΗΡΙΑΚΟ ΚΕΝΤΡΟ ΘΗΒΑΣ

ΦΥΛΛΟ ΠΛΗΡΟΦΟΡΙΩΝ

(Γι α τον Μαθητή)

ΜΑΘΗΜΑ : Χρήση Αναλογικών Εισόδων, εμφάνιση σε LCD και διασύνδεση Arduino - PC με Processing

Στόχος της Εργασίας
Σκοπός είναι η Ανάπτυξη Κώδικα για την ανάγνωση τιμών από αισθητήρα συνδεδεμένο στην αναλογική είσοδο του Arduino και η εμφάνιση σε οθόνη LCD.

Οι τιμές στη συνέχεια μέσω της Processing διαβάζονται από υπολογιστή και απεικονίζονται σε μορφή γραφικού.

Υλικά

1. Arduino Uno
2. Breadboard
3. Καλώδια (Συνδετήρες)
4. Αισθητήρας πίεσης mpx4250
5. Οθόνη LCD 4x20 i2c

Πληροφορίες – Θεωρία

Βιβλίο : Πουλάκης , Ε. (2015). *Προγραμματίζοντας με τον μικροελεγκτή Arduino.*
Ε. Πουλάκης: Ηράκλειο

Οδηγίες Χρήσης Οθόνης LCD (Από προηγούμενη Ενότητα), https://bitbucket.org/fmalpartida/new-liquidcrystal/downloads η βιβλιοθήκη για την LCD,

https://www.arduino.cc/en/Tutorial/Graph

https://processing.org/reference/ πληροφορίες για απεικόνιση σε Processing και

https://www.arduino.cc/en/Tutorial/AnalogReadSerial πληροφορίες για ADC.

Το datasheet του αισθητήρα πίεσης.

Κώδικας για ανάγνωση Αναλογικής Εισόδου

Όπως γνωρίζετε η Πλακέτα του Arduino Uno έχει 6 αναλογικές εισόδους. Για να γίνει μια μετατροπή αναλογικού σε ψηφιακό, ο μετατροπέας χρειάζεται 100μs οπότε 10000 μετατροπές μπορούν να γίνουν σε 1s. Στην εντολή analogRead(pin) ορίζω το pin που είναι συνδεδεμένος ο αισθητήρας μου. Έτσι ο Κώδικας για την ανάγνωση μια τιμής και αποστολής της στο pc θα μπορούσε να είναι :

```
void loop() {

        // Η μεταβλητη  sensorValue θα οριστει  στην αρχη του Κωδικα

        sensorValue = analogRead(analogInPin);

        Serial.println(sensorValue);

}
```

Κώδικας για απεικόνιση σε Processing

Εδώ προσοχή πρέπει να δοθεί σε μερικά σημεία. Πρώτα πρέπει να συνδεθεί το Arduino στον υπολογιστή και μετά από την δημιουργία της θύρας στα windows να ανοιχτεί η εφαρμογή processing. Στη συνάρτηση setup θα πρέπει να αναγράφεται η εντολή println(serial.list()). Έτσι θα εμφανιστεί στη κονσόλα (το μαύρο παράθυρο κάτω από το παράθυρο του κώδικα) η λίστα με τις θύρες στα windows. Έπειτα στην εντολή myPort = new Serial(this, Serial.list()[0], 9600) το νούμερο μέσα στην αγκύλη, θα πρέπει να αντιστοιχεί στο νούμερο της θύρας που έχει το Arduino. Ο κώδικας θα μπορούσε να είναι

```
import processing.serial.*;
Serial myPort;      // The serial port
void setup(){
println(Serial.list());
myPort = new Serial(this, Serial.list()[0], 9600);
……................................
}
void draw(){
……..................................
}

void serialEvent (Serial myPort) {
String inString = myPort.readStringUntil('\n');
……................................................
}
```

ΕΡΓΑΣΤΗΡΙΑΚΟ ΚΕΝΤΡΟ ΘΗΒΑΣ

ΦΥΛΛΟ ΕΡΓΟΥ

(Γι α τον Μαθητή)

ΜΑΘΗΜΑ : Χρήση Αναλογικών Εισόδων, εμφάνιση σε LCD και διασύνδεση Arduino - PC με Processing

Στόχος της Εργασίας

Σκοπός είναι η Ανάπτυξη Κώδικα για την ανάγνωση τιμών από αισθητήρα συνδεδεμένο στην αναλογική είσοδο του Arduino και η εμφάνιση σε οθόνη LCD.

Οι τιμές στη συνέχεια μέσω της Processing διαβάζονται από υπολογιστή και απεικονίζονται σε μορφή γραφικού.

Πληροφορίες –Αναζήτηση

Διαβάστε το Φύλλο πληροφοριών που σας δόθηκε . Αναζητήστε στο διαδίκτυο μέσω της μηχανής Αναζήτησης Google τον όρο " Data Acquisition" (Συλλογή Δεδομένων) . Καταγράψτε τις χρήσεις και εφαρμογές που εντοπίσατε. (Συζήτηση στη Τάξη).

Στη συνέχεια αναζητήστε τον όρο " Data Acquisition using Arduino". Εδώ θα βρούμε εφαρμογές που μπορούμε να κατασκευάσουμε και εμείς.

Σχεδίαση - Κατασκευή του Κυκλώματος

Σε συνεργασία με τα μέλη της Ομάδας :

1. Σχεδιάστε το Κύκλωμα σε σκαρίφημα.
2. Στη συνέχεια το τελικό θα σχεδιαστεί στο Πρόγραμμα Fritzing.

3. Κατασκευάστε το κύκλωμα που σχεδιάσετε .Συνδέστε τον αισθητήρα **mpx4250** στην Αναλογική είσοδο **0.**

4. Αφού μελετήσετε τον Κώδικα στο φύλλο πληροφοριών και στα link που αναγράφονται, αναπτύξτε το δικό σας ώστε να διαβάζει τις τιμές από τον Αισθητήρα

5. Συνδέστε στον υπολογιστή το Arduino και φορτώστε το Κώδικα.

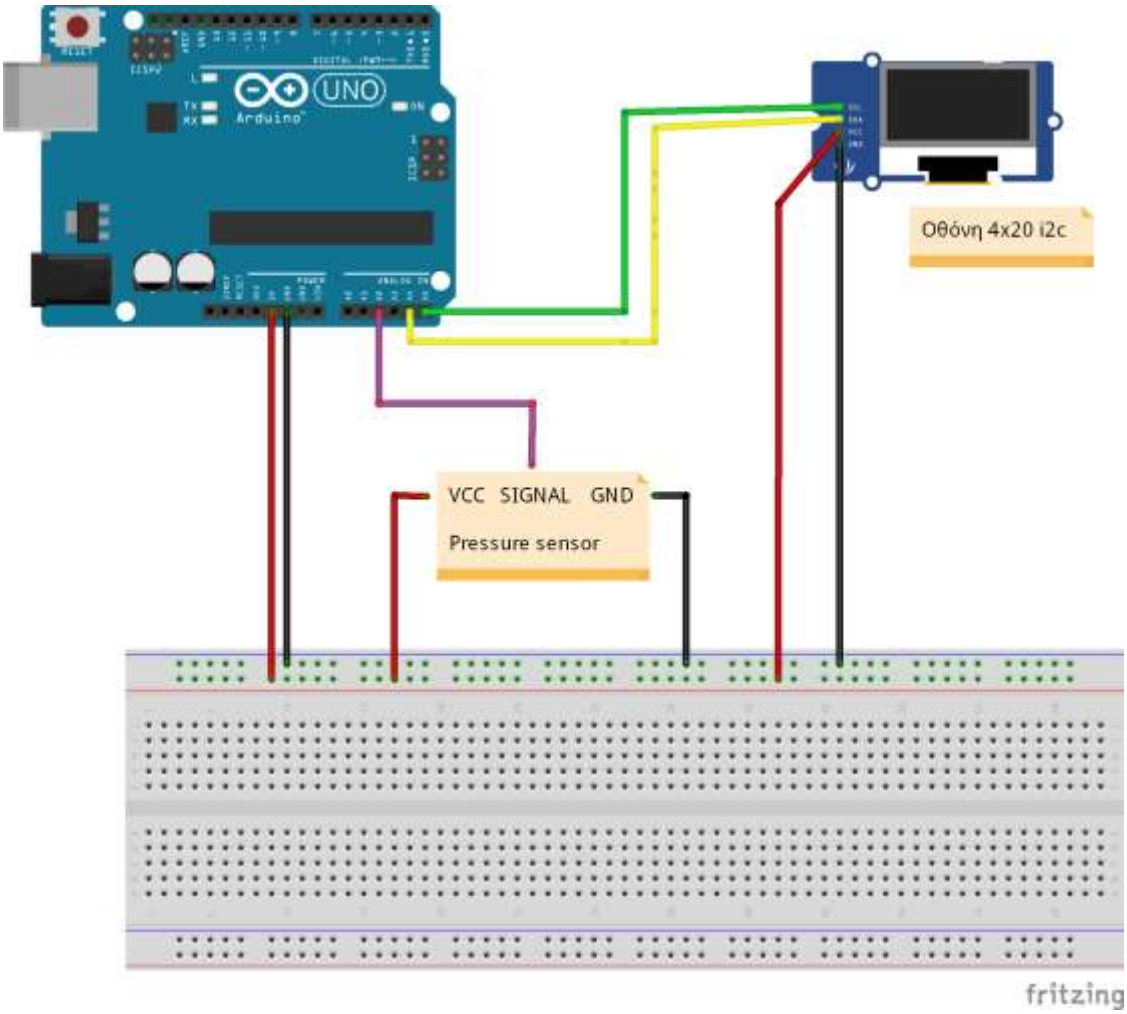
6. Εκκινήστε το Περιβάλλον της Processing

Ρυθμισεις : Δώστε προσοχή στα σημεία που αναφέρονται στο φύλλο πληροφοριών για τις πόρτες στα windows.

ΠΑΡΑΔΟΤΕΑ-ΑΠΟΤΕΛΕΣΜΑΤΑ

ΟΜΑΔΑ : …
Ονόματα Ομάδας :
1
2
3
4

ΣΧΗΜΑ ΣΥΝΔΕΣΕΩΝ (Από το Fritzing)



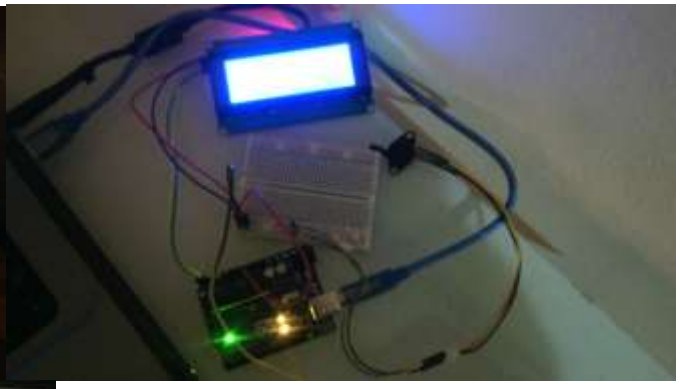
ΚΩΔΙΚΑΣ ARDUINO

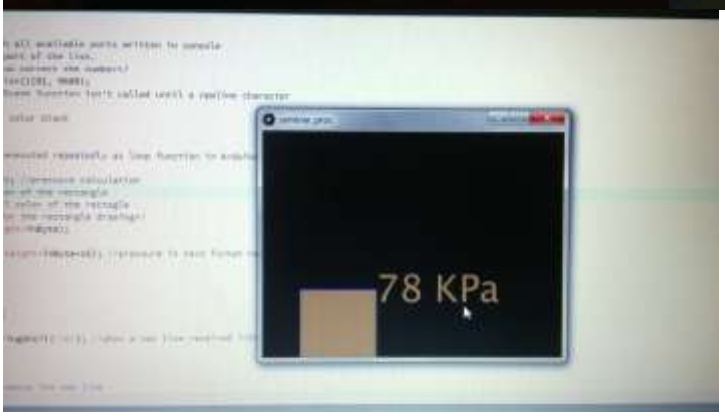
```
#include <Wire.h> //βιβλιοθήκη για i2c
#include <LiquidCrystal_I2C.h> //βιβλιοθήκη για i2c οθόνη

int value = 0; //τιμή μετατροπής adc
float pressure = 0; //τιμή πίεσης

LiquidCrystal_I2C lcd(0x27, 2, 1, 0, 4, 5, 6, 7, 3, POSITIVE);
//ορισμός αντικειμένου οθόνης lcd


void setup()
{
  Serial.begin(9600);//αρχικοποίηση της σειριακής πλατφόρμας για επικοινωνία με pc

  lcd.begin(20,4); //αρχικοποίηση οθόνης

 lcd.backlight(); //οπίσθιος φωτισμός
 lcd.clear(); //καθαρισμός οθόνης

}

void loop()
{
  value = analogRead(0); //ανάγνωση αναλογικής τιμής και μετατροπή
  delay(5);
  Serial.println(value); //αποστολή στο pc για χρήση στην processing
  delay(5);
  pressure = pres(value); //καλείται η συνάρτηση pres
  delay(10);

  lcd.setCursor(0,0);
  lcd.print("*System Pressure*"); //εμφάνιση μηνύματος
  lcd.setCursor(0,1);
  lcd.print(pressure); //εμφάνιση πίεσης
  lcd.setCursor(7,1);
  lcd.print("KPa"); //μονάδες

  delay(50);

}
/*συνάρτηση για τον υπολογισμό της πίεσης
 * όρισμα είναι η 10bit τιμή από τον μετατροπέα
 * από το datasheet του αισθητήρα εξάγεται η παρακάτω συνάρτηση μεταφοράς
  */
float pres(int q){
  return 0.259724069*q;
}
```

ΓΡΑΦΙΚΗ ΑΠΕΙΚΟΝΙΣΗ-ΚΩΔΙΚΑΣ PROCESSING

```
import processing.serial.*; //Serial communication library

Serial myPort;
int value = 0;
float inByte = 0;
int pressure = 0;

void setup(){
size(400, 300);// window size

println(Serial.list()); //list with all available ports written in console

/*myPort object handles the first port of the list.
Position might differ in your pc, so correct the number*/

myPort = new Serial(this, Serial.list()[0], 9600);
myPort.bufferUntil('\n'); //SerialEvent function isn't called until a newline character

background(0); //window background color black
}

void draw(){      //draw function executed repeatedly as loop function in Arduino
  background(0);
  pressure = int(0.259724069*value); //pressure calculation
  stroke(127, 34, 255);      //color of the rectangle
  fill(150,100,50);          //fill color of the rectangle

  /* the inByte variable is used for the rectangle drawing*/

  rect(50, height-inByte, 100, height-inByte);
  textSize(50);
  text(str(pressure)+" KPa", 150, height-inByte+15); //pressure in text format near the rectangle

}


void serialEvent (Serial myPort) {
  // get the ASCII string:
  String inString = myPort.readStringUntil('\n'); //when a new line received inString gets the value

  if (inString != null) {
    // trim off any whitespace:
    inString = trim(inString); //remove the new line
    // convert to an int and map to the screen height:
    value = int(inString);
    inByte = value;
    println(inByte);          //print inByte in console
    inByte = map(inByte, 0, 1023, 0, height); //mapping the inByte to window height
  }
}
```

System Pressure
96.887 KPa




78 KPa

113 KPa

ΕΡΩΤΗΣΕΙΣ

1. Πόσα αναλογικά pin διαθέτει το Arduino Uno ?
2. Ποιο είναι το πλεονέκτημα της i2c διασύνδεσης της οθόνης στον μικροελεγκτή;
3. Στην εντολή LiquidCrystal_I2C lcd(0x27, 2, 1, 0, 4, 5, 6, 7, 3, POSITIVE); το 0x27 παίζει συγκεκριμένο ρόλο ή μπορούμε να γράψουμε οποιαδήποτε τιμή;

Σχολική Μονάδα: ΕΠΑΛ Καρλοβασίου

Συμμετέχοντες Καθηγητές

ΜΑΡΤΙΝΗΣ ΣΤΕΛΙΟΣ

ΣΧΕΔΙΟ ΜΑΘΗΜΑΤΟΣ

Καθηγητής:	Μαρτίνης Στέλιος (ΠΕ17.04)
Σχολική Μονάδα:	ΕΠΑ.Λ Καρλοβάσου
Γνωστικό αντικείμενο:	Εκμάθηση Πλατφόρμας Arduino
Περιοχή γνωστικού αντικειμένου:	- Σύνδεση Οθόνης LCD - Διάβασμα αναλογικής εισόδου - Δομή επιλογής If,else if
Εκτιμώμενη διάρκεια:	Τρεις (3) ώρες Εργαστηρίου

Σενάριο :

1. Έχοντας αναλύσει σε προηγούμενα μαθήματα τις βασικές γνώσεις και δεξιότητες που θα πρέπει να έχει ένας μαθητής όσον αφορά:

α. τη σύνδεση ενός Arduino board μέσω usb σύνδεσης σε έναν υπολογιστή,

β. την αναγνώριση των pins του Arduino board, καθώς και

γ. την εξοικείωση με το Arduino IDE,

είμαστε πλέον σε θέση να κατασκευάσουμε μια ολοκληρωμένη εφαρμογή στο Arduino board.

Σε αυτό λοιπόν το εργαστηριακό μάθημα, διάρκειας 3 ωρών, θα υλοποιηθεί ολοκληρωμένη εφαρμογή (συνδεσμολογία κυκλώματος και συγγραφή κώδικα) που θα κάνει το Arduino board να λειτουργεί σαν ένα **"Έξυπνο Θερμόμετρο"**. Ένα θερμόμετρο δηλαδή όπου θα διαβάζει την θερμοκρασία του χώρου και θα ενημερώνει το χρήστη για την τιμή της μέσω ενός μηνύματος στην LCD οθόνη.

2. Σκοποί: Με την ολοκλήρωση της παρούσας διδακτικής ενότητας οι μαθητές θα πρέπει να είναι σε θέση να :

1. Συνδέουν κατάλληλα μια οθόνη LCD σε ένα Arduino board.
2. Χρησιμοποιούν την κατάλληλη βιβλιοθήκη και τις συναρτήσεις της, ώστε να διαχειρίζονται σωστά την LCD οθόνη μέσω του Arduino IDE.
3. Συνδέουν κατάλληλα τον αισθητήρα θερμοκρασίας LM35 σε μία αναλογική είσοδο του Arduino board.
4. Γράφουν κατάλληλο κώδικα στο Arduino IDE ώστε να διαβάζεται σωστά η θερμοκρασία από τον αισθητήρα θερμοκρασίας LM35.
5. Περιγράφουν τη δομή επιλογής if…else και if...else ιf για την επίλυση προβλημάτων.
6. Υλοποιούν κατάλληλο κώδικα if…else και if...else If στο Arduino IDE ώστε να εμφανίζονται κατάλληλα μηνύματα στο χρήστη μέσω της LCD οθόνης.

3. Προαπαιτούμενες γνώσεις : Η εξοικείωση των μαθητών με το περιβάλλον του Arduino IDE, η σύνταξη απλών προγραμμάτων, η σύνδεση εξαρτημάτων περιφερειακά του Arduino board.

4. Διδακτικές πρακτικές : Εισήγηση, Συζήτηση, Ομαδική εργασία, Επίδειξη

5. Πηγές: Παρουσιάσεις του σεμιναρίου Open Hardware 4 High Schools
Ιστοσελίδα: www.arduino.cc

6. Μέσα και υλικά : Πίνακας, Arduino board συνδεδεμένο σε Η/Υ, βιντεοπροβολέας, Φύλλα Εργασιών/Δραστηριοτήτων

7. Πορεία διδασκαλίας

α. 1η διδακτική ώρα (45’)

Μέθοδοι	**Περιεχόμενο**	**Διάρκεια**	**Μέσα διδασκαλίας**
Εισήγηση/ Επίδειξη	Υπενθύμιση βασικών εννοιών για Arduino board και τη σύνδεση του με τον Η/Υ: - Ψηφιακά/Αναλογικά pins - Σύνδεση μεσω usb σε Η/Υ - Arduino IDE	5’	- Πίνακας - Arduino board συνδεδεμένο σε Η/Υ
Επίδειξη/ Συζήτηση	Παρουσίαση της λειτουργίας και επίδειξη του τρόπου σύνδεσης μιας LCD οθόνης στο Arduino board. Παρουσίαση της βιβλιοθήκης LiquidCrystal για τον έλεγχο της οθόνης καθώς και των βασικών συναρτήσεων της.	10’	- Βιντεοπροβολέας Παρουσίαση (ppt) διαφάνειες 1-4
Ομαδική εργασία (4άδες)	Φύλλο Εργασίας 1	20’	- Φύλλο Εργασίας
Συζήτηση	Ανατροφοδότηση/ έλεγχος της δραστηριότητας	5’	
Συζήτηση	Συζήτηση των δυσκολιών που αντιμετώπισαν οι ομάδες στην διασύνδεση της LCD οθόνης	5’	

β. 2η διδακτική ώρα (45’)

Μέθοδοι	**Περιεχόμενο**	**Διάρκεια**	**Μέσα διδασκαλίας**
Εισήγηση/ Επίδειξη	Αφού έχει ολοκληρωθεί η σύνδεση της LCD οθόνης στο Arduino board, γίνεται μια μικρή επανάληψη για τις αναλογικές εισόδους που έχει το Arduino board	5’	- Πίνακας - Arduino board συνδεδεμένο σε Η/Υ
Επίδειξη/ Συζήτηση	Παρουσίαση του αισθητήρα θερμοκρασίας LM35. Επίδειξη του κυκλώματος για την σύνδεση του στο Arduino board και παρουσίαση ανάγνωσης της θερμοκρασίας στη σειριακή είσοδο του Arduino board.	10’	- Βιντεοπροβολέας Παρουσίαση (ppt) διαφάνειες 5-7
Ομαδική εργασία (4άδες)	Φύλλο Εργασίας 2	20’	- Φύλλο Εργασίας
Συζήτηση	Ανατροφοδότηση/ έλεγχος της δραστηριότητας	5’	
Συζήτηση	Συζήτηση/ επίλυση των δυσκολιών που αντιμετώπισαν οι ομάδες στη συνδεσμολογία και στην ανάγνωση της θερμοκρασίας.	5’	

γ. 3η διδακτική ώρα (45’)

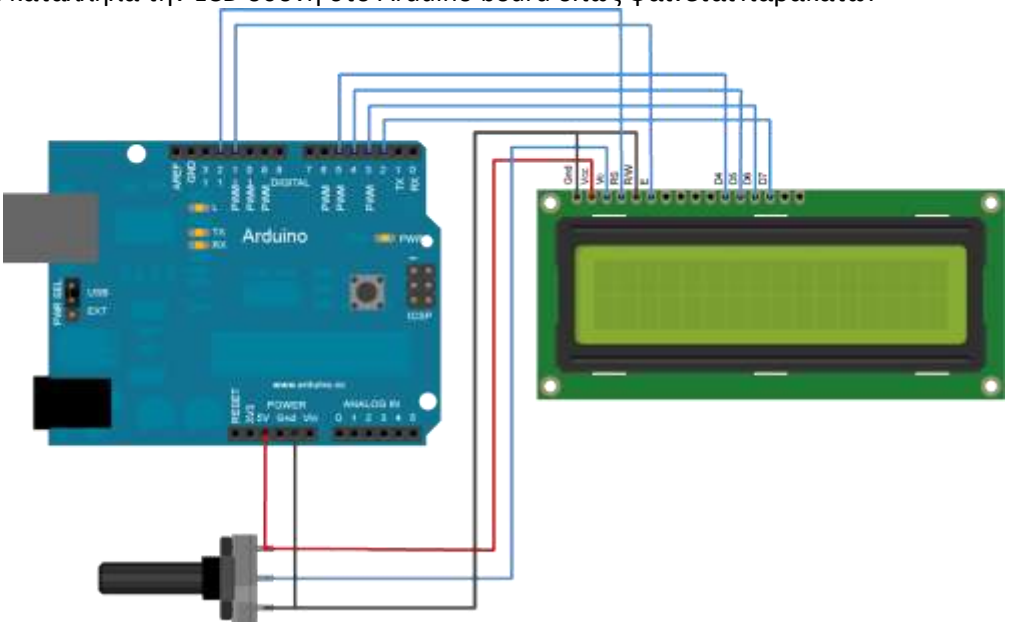
Μέθοδοι	**Περιεχόμενο**	**Διάρκεια**	**Μέσα διδασκαλίας**
Εισήγηση/ Επίδειξη	Τα δεδομένα τώρα θα τα προβάλουμε κατάλληλα στην LCD οθόνη που συνδέσαμε στο Arduino board (Φύλλο Εργασίας 1) Εισαγωγή της δομής ελέγχου if...	5’	- Πίνακας
Επίδειξη/ Συζήτηση	Παρουσίαση της δομής ελέγχου if..else, else if. Επίδειξη παραδειγμάτων.	10’	- Βιντεοπροβολέας Παρουσίαση (ppt) διαφάνειες 8,9
Ομαδική εργασία (4άδες)	Φύλλο Εργασίας 3	20’	- Φύλλο Εργασίας
Συζήτηση	Ανατροφοδότηση/ έλεγχος της δραστηριότητας	5’	
Συζήτηση	Ανακεφαλαίωση βασικών δεξιοτήτων: -Σύνδεση LCD Οθόνης -Σύνδεση και ανάγνωση από αναλογική είσοδο -Σωστή χρησιμοποίηση της δομής ελέγχου if..else, else if	5’	- Βιντεοπροβολέας Παρουσίαση (ppt) διαφάνειες 10

Φύλλο Εργασίας 1

Ομάδα εργασίας (Ονοματεπώνυμα):
.....
.....
.....
.....
.....

Απαιτούμενα Υλικά
- 1 Arduino board συνδεδεμένο σε Η/Υ
- 1 Οθόνη LCD παράλληλης σύνδεσης
- 1breadboard
- 1 ποτενσιόμετρο
- καλώδια για συνδέσεις

Πορεία εργασίας
α. Συνδέστε κατάλληλα την LCD οθόνη στο Arduino board όπως φαίνεται παρακάτω:



β. Συνδέστε το Arduino board με τον Η/Υ και ανοίξτε το Arduino IDE. Ποιος είναι ο ρόλος του ποτενσιόμετρου στη συνδεσμολογία μας;
.....
.....
.....

γ. Έχει κάποια ένδειξη η οθόνη αυτή τη στιγμή;
.....

δ. Δημιουργήστε ένα νέο sketch δίνοντας του το όνομα "lcd_connect" και εισάγεται την κατάλληλη βιβλιοθήκη ώστε να μπορείτε να διαχειριστείτε την οθόνη LCD.
ε. Πριν τη συνάρτηση void setup() γράψτε την παρακάτω συνάρτηση:

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

Εξηγήστε τη χρησιμότητα των αριθμών μέσα στη συνάρτηση
.....
.....
.....

στ. Μέσα στη συνάρτηση void setup() γράψτε τις παρακάτω εντολές

```
lcd.begin(16, 2);
lcd.clear();
lcd.print("εδώ γράψτε το όνομα σας");
```

Τι παρατηρείτε; Τι κάνει καθεμιά από αυτές τις εντολές;

.....

.....

.....

.....

.....

.....

ζ. Δοκιμάστε και καταγράψτε τη λειτουργία των παρακάτω συναρτήσεων της βιβλιοθήκης

```
setCursor()
noCursor()
```

.....

.....

.....

.....

.....

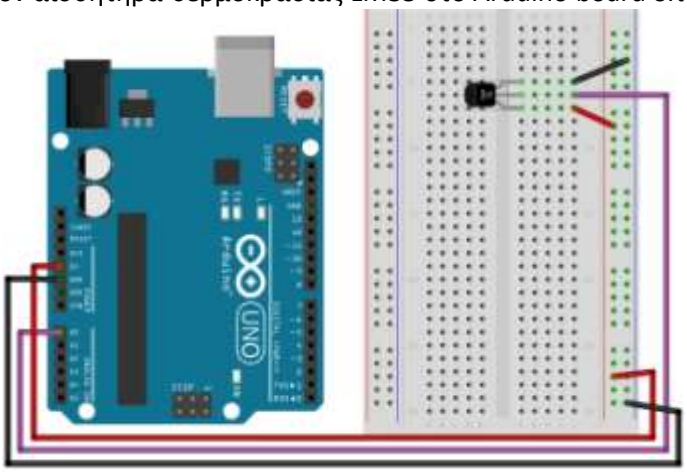
.....

Φύλλο Εργασίας 2

Ομάδα εργασίας (Ονοματεπώνυμα):
.....
.....
.....
.....
.....

Απαιτούμενα Υλικά
- 1 Arduino board συνδεδεμένο σε Η/Υ
- 1 αισθητήρα θερμοκρασίας LM35
- καλώδια για συνδέσεις

Πορεία εργασίας
α. Συνδέστε κατάλληλα τον αισθητήρα θερμοκρασίας LM35 στο Arduino board όπως φαίνεται παρακάτω:



β. Συνδέστε το Arduino board με τον Η/Υ και ανοίξτε το Arduino IDE *(χωρίς να αποσυνδέσετε την LCD οθόνη από την προηγούμενη δραστηριότητα)*. Σε ποια αναλογική είσοδο είναι συνδεδεμένος ο αισθητήρας;
.....
.....
.....

γ. Δημιουργήστε ένα νέο sketch δίνοντας του το όνομα "lm35_connect".
δ. Υλοποιήστε τον παρακάτω ψευδοκώδικα σε γλώσσα Arduino.

```
Δήλωση pin εισόδου
void setup(){
  Ξεκίνημα της σειριακής εισόδου του Arduino σε ταχύτητα 9600
  }
void loop(){
  Διάβασε την τιμή της αναλογικής εισόδου
  Αποθήκευσε αυτήν την τιμή σε μια μεταβλητή
  Τύπωσε την τιμή που διάβασες στη σειριακή κονσόλα
  }
```

Κάνε upload του κώδικα που γράψατε στο Arduino board και καταγράψτε ενδεικτικά κάποιες τιμές
.....
.....
.....
.....

Τι δηλώνουν αυτές οι τιμές που τυπώνονται στη σειριακή κονσόλα;

.....

.....

ε. Για τη μετατροπή των τιμών σε κατανοητή τιμή θερμοκρασίας γράψτε τον παρακάτω κώδικα:

```
 int temperature = (5.0 * "τιμή που διαβάσατε πριν") * 100.0) / 1024;
Serial.print(", degrees C: ");
Serial.println(temperature);
```

στ. Τι κάνει ακριβώς ο παραπάνω κώδικας;

.....

.....

.....

.....

.....

.....

.....

ζ. Κάντε εκ νέου upload του sketch και δείτε τα αποτελέσματα στην κονσόλα.

η. Τι πρέπει να διορθώσετε στον κώδικα του sketch ώστε η ανάγνωση να γίνεται κάθε 5sec;

.....

.....

.....

.....

.....

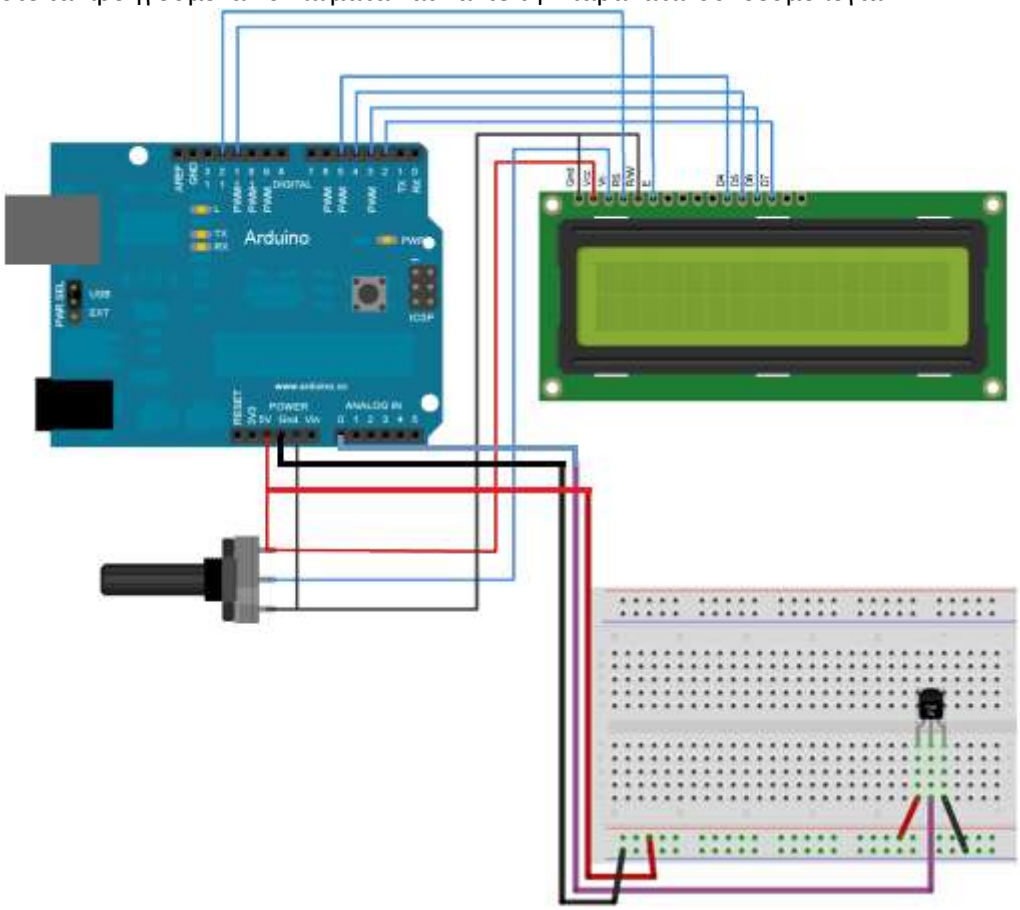
.....

Φύλλο Εργασίας 3

Ομάδα εργασίας (Ονοματεπώνυμα):
.....
.....
.....
.....
.....

Απαιτούμενα Υλικά
- 1 Arduino board συνδεδεμένο σε Η/Υ
- 1 Οθόνη LCD παράλληλης σύνδεσης
- 1breadboard
- 1 ποτενσιόμετρο
- 1 αισθητήρα θερμοκρασίας TMP36
- καλώδια για συνδέσεις

Πορεία εργασίας
α. Συνδυάστε τα προηγούμενα κυκλώματα και κάντε την παρακάτω συνδεσμολογία:



β. Συνδέστε το Arduino board με τον Η/Υ και ανοίξτε το Arduino IDE.
γ. Δημιουργήστε ένα νέο sketch δίνοντας του το όνομα "smart_temp".
δ. Συνδυάστε τα προηγούμενα δύο sketches ώστε να εμφανίζεται στην lcd οθόνη η θερμοκρασία που διαβάζει ο αισθητήρας.

ε. Υλοποιήστε την παρακάτω λογική ώστε το θερμόμετρο μας να γίνει λιγάκι "έξυπνο":

Όταν η θερμοκρασία του δωματίου είναι πάνω από 28 βαθμούς Κελσίου τότε να εμφανίζεται μήνυμα στην οθόνη "Hot". Ενώ όταν η θερμοκρασία είναι κάτω από τους 10 βαθμούς Κελσίου να εμφανίζεται μήνυμα στην οθόνη "Cold". Σε ενδιάμεσες θερμοκρασίες να εμφανίζεται μήνυμα στην οθόνη "Normal".

Γράψτε εδώ τον κώδικά που σκεφτήκατε

.....
.....
.....
.....
.....
.....
.....
.....
.....
.....
.....
.....

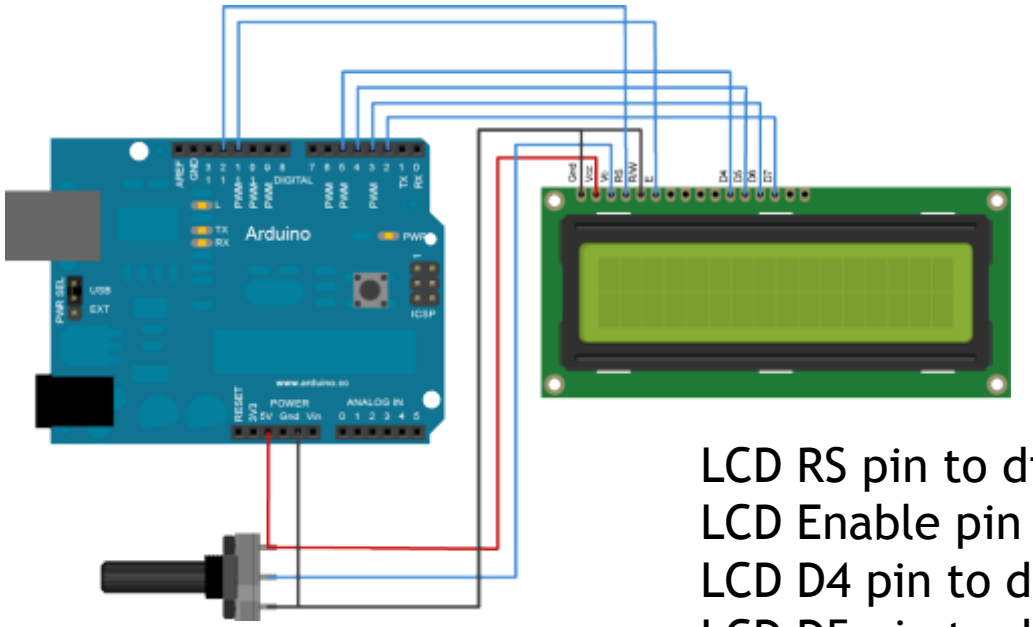
στ. Κάντε upload του sketch και δείτε τα αποτελέσματα στην οθόνη.
ζ. Εντοπίστε προβλήματα της υλοποίησης προτείνετε βελτιώσεις

.....
.....
.....
.....
.....
.....
.....
.....
.....
.....
.....
.....

ΜΑΡΤΙΝΗΣ ΣΤΕΛΙΟΣ Π.Ε 17.04 ΕΚΜΑΘΗΣΗ ΠΛΑΤΟΡΜΑΣ ARDUINO

- Σύνδεση Οθόνης LCD
- Διάβασμα αναλογικής εισόδου
- Δομή επιλογής If,else if

ΣΥΝΔΕΣΗ LCD ΟΘΟΝΗΣ ΣΤΟ ARDUINO



LCD RS pin to digital pin 12
LCD Enable pin to digital pin 11
LCD D4 pin to digital pin 5
LCD D5 pin to digital pin 4
LCD D6 pin to digital pin 3
LCD D7 pin to digital pin 2

- Αρκεί μόνο η σωστή σύνδεση της οθόνης;

ΒΙΒΛΙΟΘΗΚΗ LIQUIDCRYSTAL.H

- Θα πρέπει να φορτώσουμε στο sketch μας την βιβλιοθήκη <LiquidCrystal.h>
- Συναρτήσεις της βιβλιοθήκης:

  - LiquidCrystal()
  - begin()
  - clear()
  - home()
  - setCursor()
  - write()
  - print()
  - cursor()
  - noCursor()
  - blink()
  - noBlink()
  - display()
  - noDisplay()
  - scrollDisplayLeft()
  - scrollDisplayRight()
  - autoscroll()
  - noAutoscroll()
  - leftToRight()
  - rightToLeft()
  - createChar()

ΠΑΡΑΔΕΙΓΜΑ

```
#include <LiquidCrystal.h>

LiquidCrystal lcd(12, 11, 10, 5, 4, 3, 2);

void setup(){
  lcd.begin(16,1);
  lcd.print("hello, world!");
}

void loop() {}
```

ΑΙΣΘΗΤΗΡΑΣ LM35

- Features
- Calibrated directly in ° Celsius (Centigrade)
- Linear + 10.0 mV/°C scale factor
- 0.5°C accuracy guaranteeable (at +25°C)
- Rated for full -55° to +150°C range
- Suitable for remote applications
- Low cost due to wafer-level trimming
- Operates from 4 to 30 volts
- Less than 60 μA current drain
- Low self-heating, 0.08°C in still air
- Nonlinearity only ±¼°C typical
- Low impedance output, 0.1 Ohm for 1 mA load

Typical Applications

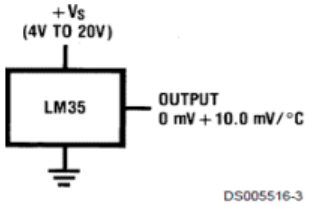
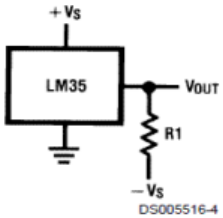


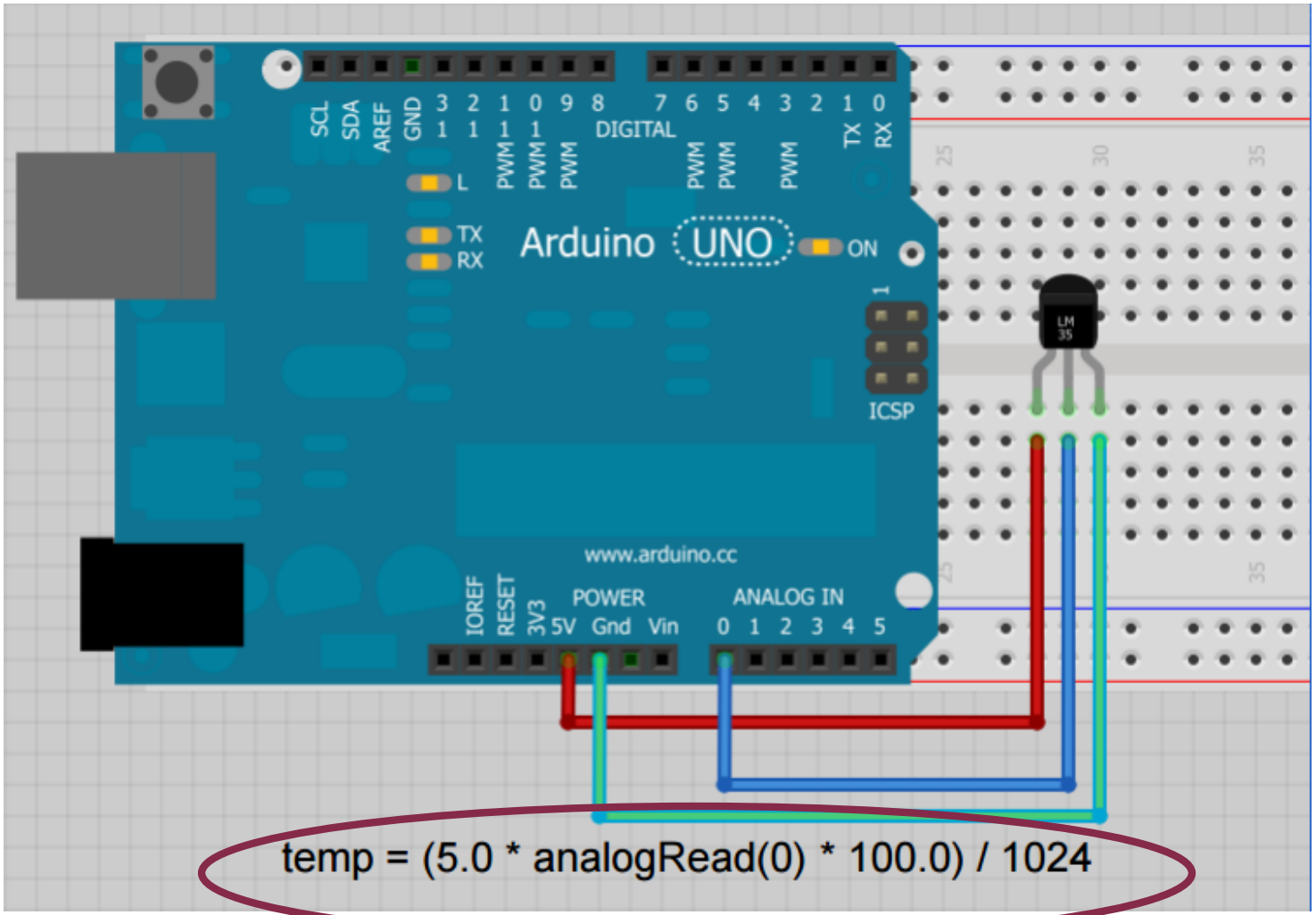
FIGURE 1. Basic Centigrade Temperature Sensor (+2°C to +150°C)



Choose $R_1 = -V_S/50\ \mu A$
V_{OUT} = +1,500 mV at +150°C
= +250 mV at +25°C
= −550 mV at −55°C

FIGURE 2. Full-Range Centigrade Temperature Sensor

ΣΥΝΔΕΣΜΟΛΟΓΙΑ



temp = (5.0 * analogRead(0) * 100.0) / 1024

ΠΑΡΑΔΕΙΓΜΑ

```
float tempC;
int reading;
int tempPin = A0;

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

void loop()
{
reading = analogRead(tempPin);
tempC = (5.0 * analogRead(tempPin) * 100.0) / 1024;
Serial.println(tempC);
delay(1000);
}
```

ΔΟΜΗ ΕΛΕΓΧΟΥ IF…

- Πολλές φορές κατά τη διάρκεια εκτέλεσης του προγράμματός μας, θέλουμε να εκτελείται ένα συγκεκριμένο σύνολο εντολών, ανάλογα με το αν ισχύει κάποια συνθήκη. Για να το επιτύχουμε αυτό, να ελέγξουμε δηλαδή τη ροή του προγράμματος, χρησιμοποιούμε τις **δομές ελέγχου**:
  - **if**
  - **if..else if**

```
if (<συνθήκη>)
{
        // εάν ισχύει η <συνθήκη>
        // <εντολές>
}
```

if..else if

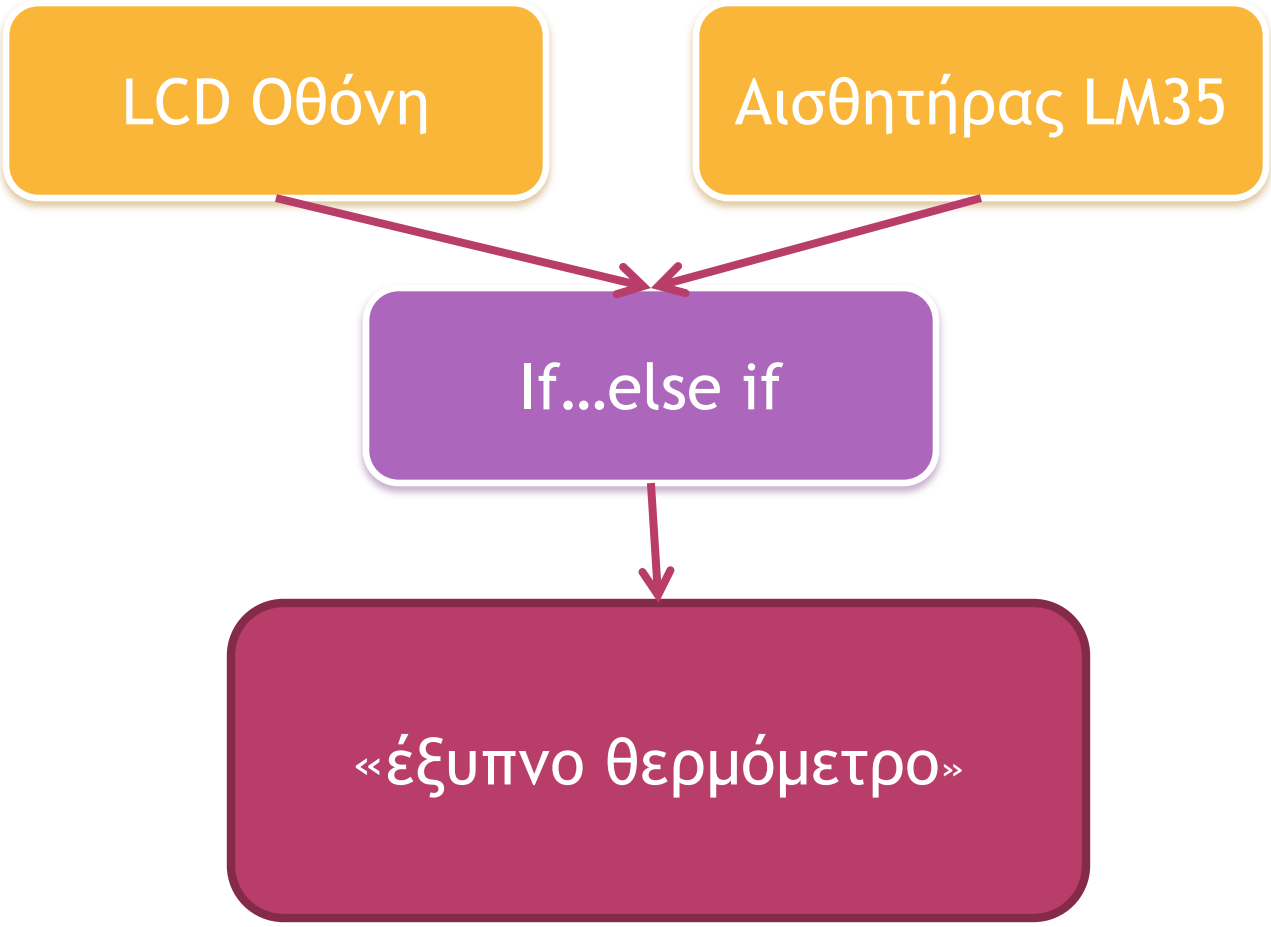
```
if (<συνθήκη 1>)
{
        // εάν ισχύει η <συνθήκη 1>
        // <εντολές 1>
}
else if (<συνθήκη 2>)
{
        // εάν ισχύει η <συνθήκη 2>
        // <εντολές 2>
}
else // η else μπαίνει προαιρετικά
{
        // σε κάθε άλλη περίπτωση
        // <εντολές 3>
}
```

ΠΑΡΑΔΕΙΓΜΑ

```
const int analogPin = A0;
const int ledPin = 13;
const int threshold = 400;
```

```
void setup() {
    pinMode(ledPin, OUTPUT);
   Serial.begin(9600);
}

void loop() {
    int analogValue = analogRead(analogPin);

  if (analogValue > threshold) {
    digitalWrite(ledPin, HIGH);
  } else {
    digitalWrite(ledPin, LOW);
  }
  // print the analog value:
  Serial.println(analogValue);
  delay(1);
}
```

ΑΝΑΚΕΦΑΛΑΙΩΣΗ



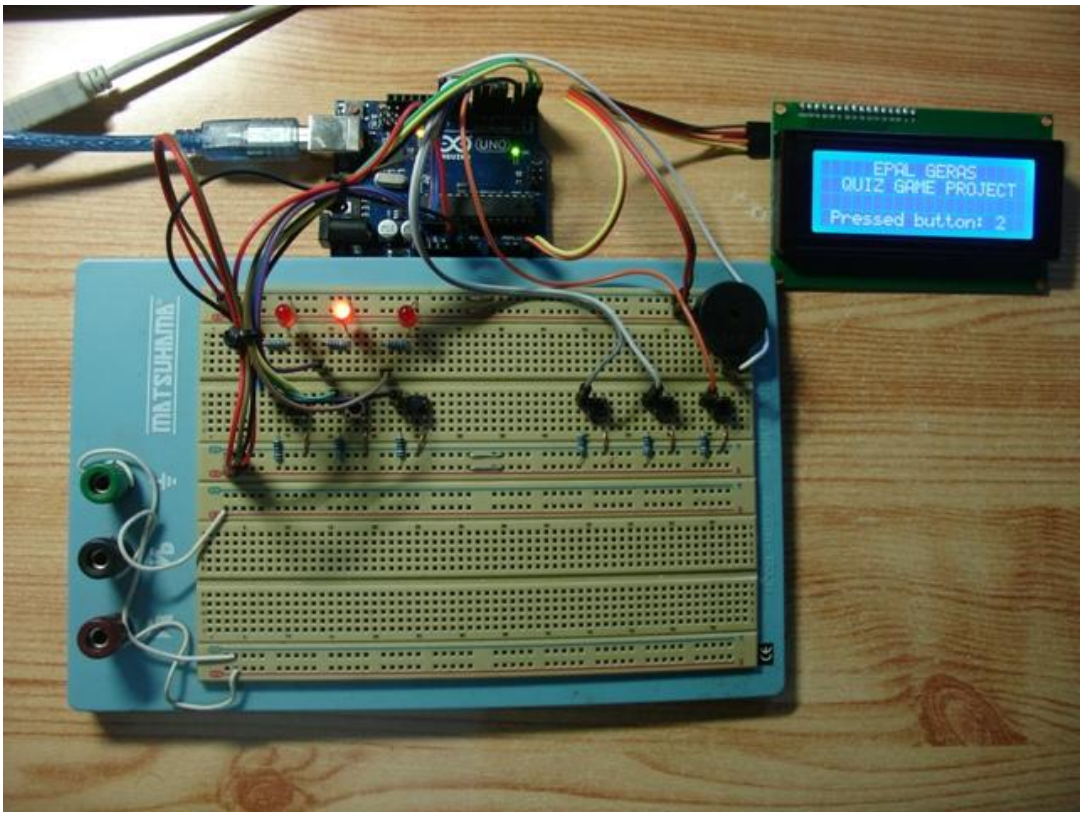
Σχολική Μονάδα: ΕΠΑΛ Γέρα Λέσβου

Συμμετέχοντες Καθηγητές

ΚΟΥΡΤΙΔΗΣ ΕΥΑΓΓΕΛΟΣ
ΦΩΤΙΑΔΗΣ ΚΩΝΣΤΑΝΤΙΝΟΣ
ΨΩΜΟΣ ΒΑΣΙΛΕΙΟΣ

ΔΙΔΑΚΤΙΚΟ ΣΕΝΑΡΙΟ

«Δημιουργία συστήματος διεξαγωγής παιχνιδιού ερωτήσεων/απαντήσεων με χρήση Arduino – QUIZ GAME»



Εργασία στα πλαίσια του σεμιναρίου:
OPEN HARDWARE FOR HIGH SCHOOL (OH4HS) COURSE του Ε.Α.Π.

Ομάδα Εργασίας: Φωτιάδης Κωνσταντίνος εκπ/κος, ΠΕ17.07
Ψώμος Βασίλειος, εκπ/κος ΠΕ19
Κουρτίδης Ευάγγελος, εκπ/κος ΠΕ20

Σχολείο: ΕΠΑ.Λ. Γέρας

Επόπτης : Δρ. Φωτόπουλος Βασίλειος, ΣΕΠ Ψ.Σ. Ε.Α.Π.

Παπάδος Γέρας, 22 Ιανουαρίου 2016

Δημιουργία συστήματος διεξαγωγής παιχνιδιού ερωτήσεων/απαντήσεων με χρήση Arduino – QUIZ GAME

Δημιουργοί: Φωτιάδης Κωνσταντίνος, εκπ/κός ΠΕ17.07
Ψώμος Βασίλειος, εκπ/κός ΠΕ19
Κουρτίδης Ευάγγελος, εκπ/κός ΠΕ20

Σχολείο: 1ο ΕΠΑΛ ΓΕΡΑΣ, Δ/ΝΣΗ ΔΕΥΤΕΡΟΒΑΘΜΙΑΣ ΕΚΠ/ΣΗΣ Ν. ΛΕΣΒΟΥ

1. ΠΡΟΚΑΤΑΡΚΤΙΚΑ ΣΤΟΙΧΕΙΑ

Σκοποί και Στόχοι:
Στο τέλος της εφαρμογής του σεναρίου στην τάξη, οι μαθητές θα πρέπει

- Να οργανώνονται σε ομάδες και να υλοποιούν μεθοδικά την εργασία μιας εφαρμογής μικροελεγκτή.
- Να κατασκευάζουν το πλήρες κύκλωμα της εφαρμογής.
- Να αναγνωρίζουν τις εισόδους και τις εξόδους της.
- Να ακολουθούν οδηγίες και να γράφουν κώδικα για την λειτουργία της.
- Να ανακατασκευάζουν/παραμετροποιούν το σύστημα (κύκλωμα+κώδικας), σύμφωνα με τα νέα δεδομένα της εφαρμογής.
- Να χρησιμοποιούν την εφαρμογή στα πλαίσια της διδασκαλίας ενός μαθήματος.

Αριθμός φάσεων διδακτικού σεναρίου: Τέσσερις (4)

Εκτιμώμενη διάρκεια:

	Διδακτικές ώρες
Φάση 1	3
Φάση 2	3
Φάση 3	3
Φάση 4	2
Σύνολο	**11**

Προαπαιτούμενες γνώσεις μαθητών:
Οι μαθητές έχουν διδαχθεί εισαγωγικά μαθήματα και έχουν υλοποιήσει απλές εφαρμογές με χρήση Arduino.

Συσχετισμός με το Αναλυτικό Πρόγραμμα / Διαθεματικότητα:
Το σενάριο αυτό μπορεί να συσχετιστεί με την διδασκαλία Αρχών Ψηφιακής Τεχνολογίας, Μικροελεγκτών, Δομημένου Προγραμματισμού, Αντικειμενοστραφούς Προγραμματισμού, Συστημάτων Αυτοματισμού, Βασικών Ηλεκτρονικών ή Βασικών Ηλεκτρολογικών Γνώσεων των Τομέων Πληροφορικής, Αυτοματισμού, Ηλεκτρονικών και Ηλεκτρολόγων των ΕΠΑΛ.
Μπορεί επίσης να ενταχθεί στο περιεχόμενο ενός project.

Διδακτικό υλικό και απαιτούμενη υλικοτεχνική υποδομή – Προετοιμασία:
Το σχολείο θα πρέπει να διαθέτει το Arduino Starter Kit καθώς και ένα άρθρωμα σειριακής LCD Οθόνης. Οι υπολογιστές του εργαστηρίου να έχουν εγκατεστημένο το Arduino IDE με την βιβλιοθήκη New Liquid Crystal. Θα πρέπει να είναι διαθέσιμα και τα αρχεία **QuizGame.ino** και

QuizGameCodeExercise.ino μέσα στους αντίστοιχους φακέλους τους **QuizGame** και **QuizGameCodeExercise**.

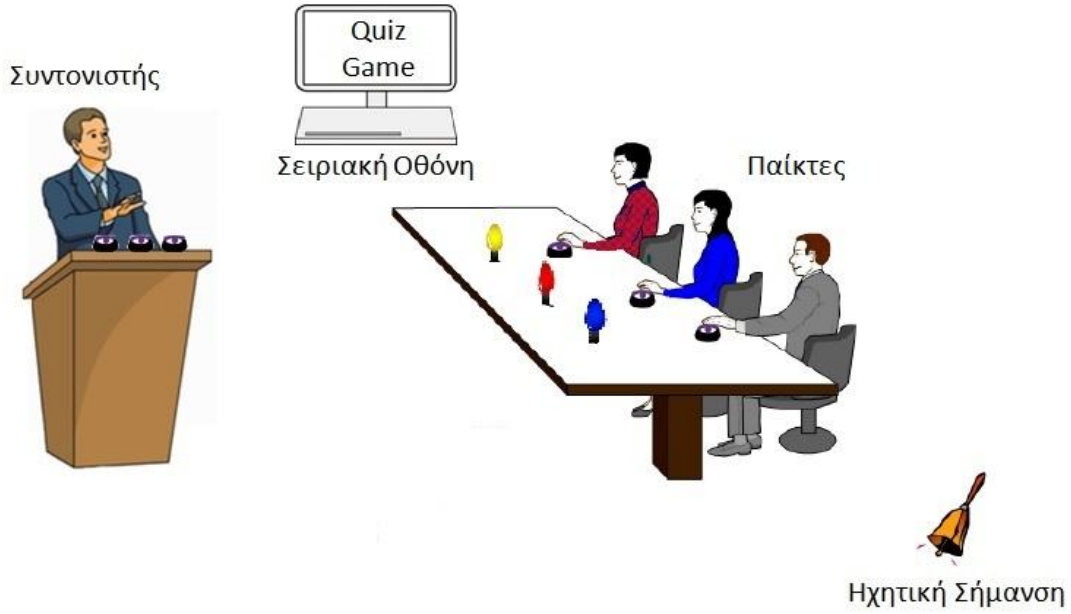
Πρότερες αναπαραστάσεις μαθητών:
Κάποιοι μαθητές να θεωρούν εκ των προτέρων ότι η δημιουργία ενός πάνελ για Quiz αποτελεί ένα εγχείρημα δύσκολο και υπερβολικά σύνθετο.

Οργάνωση τάξης:
Οι μαθητές οργανώνονται σε ομάδες των τριών – τεσσάρων ατόμων. Ο καθηγητής έχει βοηθητικό – συμβουλευτικό ρόλο.

Διδακτικές προσεγγίσεις:
Ενεργητική συμμετοχή - Διερευνητική μάθηση - Εργασία σε ομάδες – Παιχνίδι ρόλων.

2. ΓΕΝΙΚΗ ΠΕΡΙΓΡΑΦΗ

Στόχος της ομάδας των μαθητών είναι να κατασκευάσουν ένα σύστημα διεξαγωγής παιχνιδιού ερωτήσεων/απαντήσεων (Quiz Game), συγκεκριμένα ένα επιτραπέζιο πάνελ αποτελούμενο από κουμπιά (buttons), ηχείο (buzzer), φωτεινές σημάνσεις (leds), που θα υπηρετεί τη λειτουργία, τους όρους και τους κανόνες του παιχνιδιού (***Σχήμα 1***).



Σχήμα 1: Γενική Διαρρύθμιση παιχνιδιού Quiz Game

Το πάνελ θα το χειρίζονται οι συμμετέχοντες του παιχνιδιού (3 ή 4 παίκτες και ένας συντονιστής). Για κάθε παίκτη αντιστοιχεί ένα button, ενώ ο συντονιστής έχει στη διάθεσή του 3 buttons. Αρχικά, ο συντονιστής απευθύνεται στο σύνολο των παικτών διατυπώνοντας μια ερώτηση. Όποιος παίκτης γνωρίζει την απάντηση πατάει το button του και το led του ταχύτερου ανάβει. Το σύστημα αποκλείει προσωρινά τους άλλους παίκτες από οποιαδήποτε ενέργεια. Πατώντας το πρώτο του button (1/3) ο συντονιστής προετοιμάζει την έγκριση ή την απόρριψη της απάντησης με την φράση «ο παίκτης …. απάντησε...». Αν ο παίκτης απαντήσει σωστά στην ερώτηση τότε ο συντονιστής πατάει το button του (2/3) που αντιστοιχεί στην πριμοδότηση του παίκτη με 10 πόντους. Αν απαντήσει λάθος, ο

συντονιστής πατάει το κουμπί «ξεκλειδώματος» (3/3) των άλλων παικτών για να απαντήσουν κ.ο.κ. Κάθε ενέργεια ή αποτέλεσμα συνοδεύεται από ανάλογο ηχητικό εφέ. Επίσης κάθε απόκριση του συστήματος στη ροή του παιχνιδιού γίνεται μέσω μηνυμάτων στη σειριακή οθόνη ή – μετά την επέκταση της εφαρμογής - στην οθόνη LCD. Στο τέλος αφού έχουν ολοκληρωθεί όλες οι ερωτήσεις, ο παίκτης με τους περισσότερους πόντους είναι ο νικητής.

Το διδακτικό σενάριο προβλέπει την κατασκευή του συστήματος σε τέσσερις φάσεις μέσα από τις οποίες οι μαθητές θα έχουν την δυνατότητα να αποκτήσουν προοδευτικά δεξιότητες κατασκευής και προγραμματισμού ηλεκτρονικών διατάξεων με χρήση μικροελεγκτή. Επίσης η εφαρμογή μπορεί να χρησιμοποιηθεί στην ολοκληρωμένη της μορφή, στα πλαίσια ενός άλλου μαθήματος ως εργαλείο-παιχνίδι μάθησης.

Φάση 1η (Φύλλο Εργασίας 1): Οι μαθητές κατασκευάζουν το πρωτότυπο του υλικού μέρους και εξασκούνται στον προγραμματισμό των βασικών λειτουργιών του κυκλώματος.

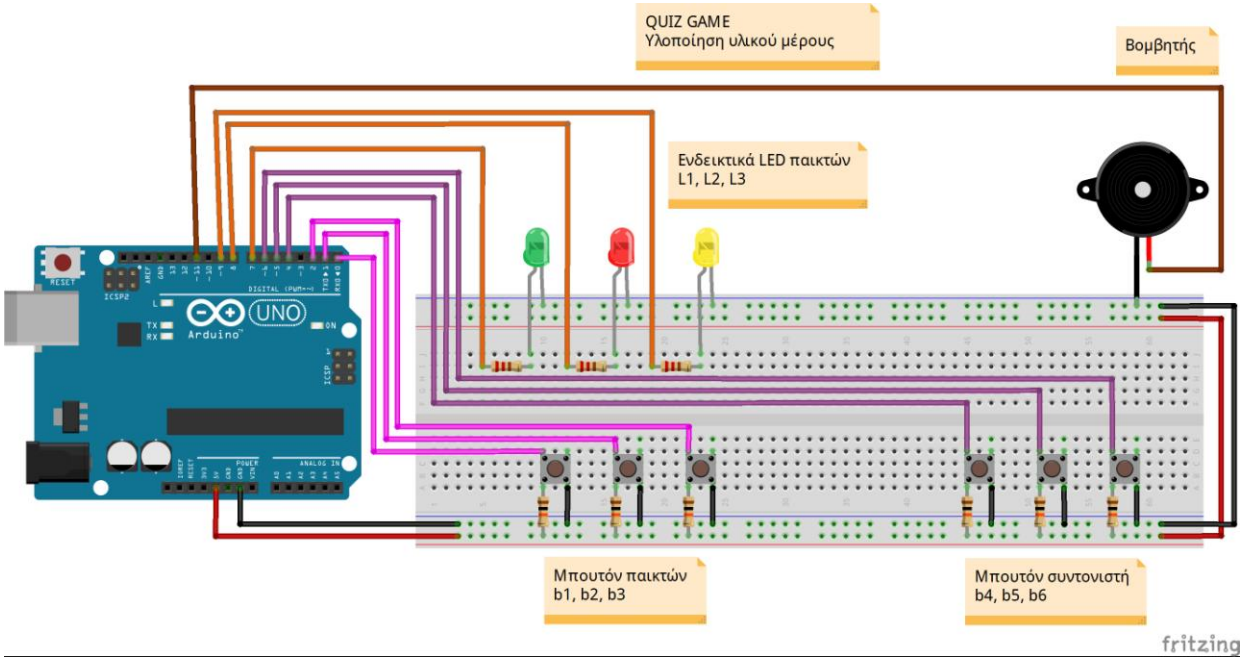
Φάση 2η (Φύλλο Εργασίας 1): Οι μαθητές επεκτείνουν την λειτουργία του Quiz Game από τρεις σε τέσσερις παίκτες. Παράλληλα, εξασκούνται στον προγραμματισμό του τροποποιημένου κυκλώματος.

Φάση 3η (Φύλλο Εργασίας 3): Οι μαθητές τροποποιούν την λειτουργία του συστήματος (κύκλωμα + κώδικας), προσθέτοντας στο κύκλωμα μια σειριακή LCD οθόνη.

Φάση 4η (Φύλλο Εργασίας 4): Οι μαθητές κάνουν χρήση του τελικού συστήματος παιχνιδιού Quiz Game σε πραγματικές συνθήκες μέσα σε μια τάξη στα πλαίσια ενός μαθήματος.

3. ΠΕΡΙΓΡΑΦΗ ΥΛΙΚΟΥ ΜΕΡΟΥΣ

Για την υλοποίηση του υλικού μέρους του συστήματος, οι μαθητές αρχικά θα κατασκευάσουν το πρωτότυπο κύκλωμα σύμφωνα με το ***Σχέδιο 1***, που θα τους δοθεί. Η κατασκευή υλοποιείται με το Φύλλο Εργασίας 1.



Σχέδιο 1: Πλήρες κύκλωμα παιχνιδιού Quiz Game

Το κύκλωμα του ***Σχεδίου 1*** θα τροποποιηθεί από τους μαθητές κατά την 2η και 3η φάση (Φύλλα Εργασίας 2 και 3) με την προσθήκη ενός επιπλέον παίκτη και μιας σειριακής LCD οθόνης (***Σχήμα 2***).

Σχήμα 2: Άρθρωμα σειριακής LCD οθόνης

4. ΠΕΡΙΓΡΑΦΗ ΛΟΓΙΣΜΙΚΟΥ ΜΕΡΟΥΣ

Το λογισμικό μέρος του συστήματος αποτελείται από δύο προγράμματα αποθηκευμένα στα αρχεία:

1) **QuizGame.ino**

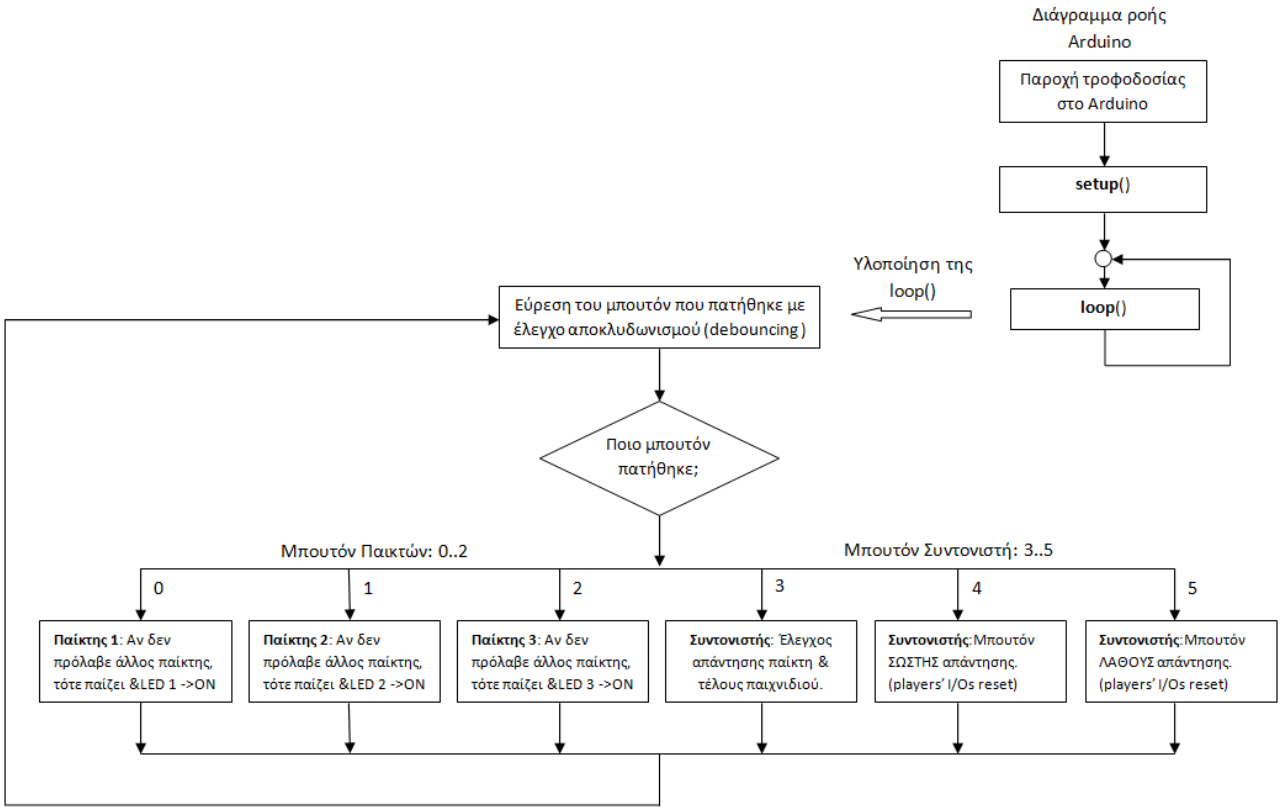
   Είναι ένα πρόγραμμα με το οποίο υλοποιείται η πλήρης λειτουργικότητα του παιχνιδιού Quiz Game.

2) **QuizGameCodeExercise.ino**

   Είναι ένα πρόγραμμα που έχει την βασική δομή του πλήρους προγράμματος, από το οποίο όμως έχουν αφαιρεθεί οι πιο σύνθετες λειτουργίες του. Είναι δηλαδή ένα απλοποιημένο πρόγραμμα του κυκλώματος του παιχνιδιού, το οποίο έχει διαμορφωθεί ώστε να εξυπηρετήσει την μαθησιακή διαδικασία στις διάφορες φάσεις του διδακτικού σεναρίου.

   Οι μαθητές δουλεύοντας πάνω στο **QuizGameCodeExercise.ino** εκπαιδεύονται σταδιακά στον προγραμματισμό του πλήρους συστήματος.

Ένα συνοπτικό διάγραμμα ροής της δομής του προγράμματος του Quiz Game δίνεται στο ***Σχήμα 3.***



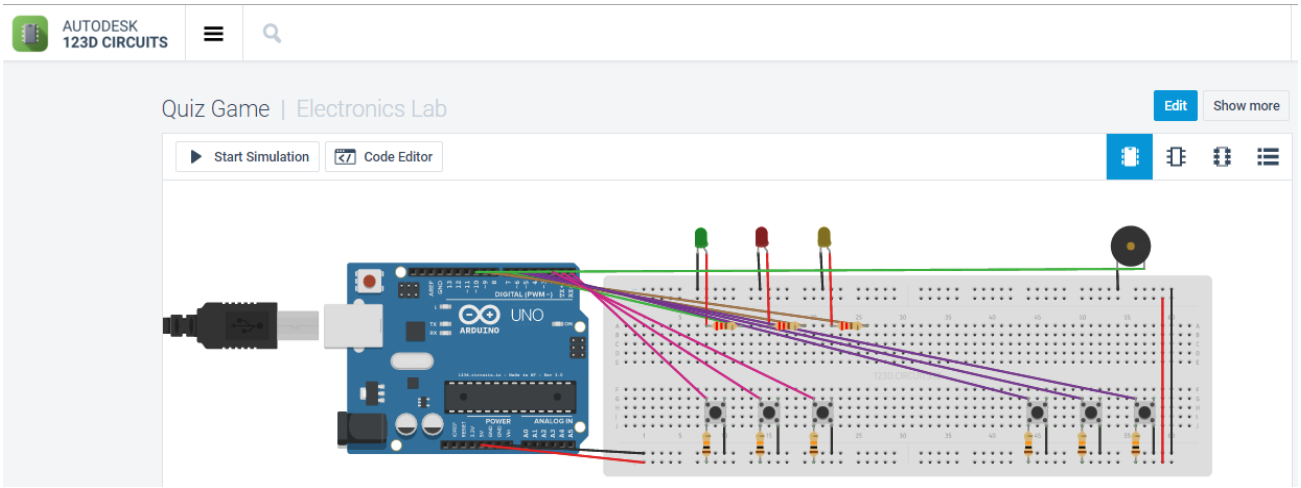
Σχήμα 3: Συνοπτικό διάγραμμα ροής προγράμματος Quiz Game

Για το άνοιγμα των παραπάνω αρχείων και τον προγραμματισμό του συστήματος, οι μαθητές θα εργαστούν στο περιβάλλον προγραμματισμού Arduino IDE.

Τα μηνύματα του παιχνιδιού στέλνονται σε μια σειριακή οθόνη, μέσω της σύνδεσης του Arduino με μια θύρα USB ενός Η/Υ. Ένα παράδειγμα μιας σειράς μηνυμάτων του Quiz Game δίνεται παρακάτω:

```
VOCATIONAL HIGH SCHOOL OF GERA, LESVOS
**********  THE QUIZ GAME  ***********

The Answer of Player 1 is: RIGHT, 10pts in total
 The Answer of Player 2 is: RIGHT, 10pts in total
 The Answer of Player 2 is: RIGHT, 20pts in total
The Answer of Player 3 is: WRONG, 0pts in Total
The Answer of Player 1 is: RIGHT, 20pts in total

The total players' points are:
Player 1: 20pts
Player 2: 20pts
Player 3: 0pts

****  The players points go back to zero  ****

*** NEW ROUND OF THE QUIZ GAME IS STARTING ***
```

Οι μαθητές μπορούν να πειραματιστούν με την λειτουργία QUIZ GAME και μέσα από την εξομοίωση του συστήματος (***Σχήμα 4***) που είναι διαθέσιμη στην διεύθυνση: https://123d.circuits.io/circuits/1423275-quiz-game του ιστότοπου της AUTODESK 123D CIRCUITS.



Σχήμα 4: Εξομοίωση συστήματος Quiz Game

ΔΙΔΑΚΤΙΚΟ ΣΕΝΑΡΙΟ

"ΔΗΜΙΟΥΡΓΙΑ ΣΥΣΤΗΜΑΤΟΣ ΔΙΕΞΑΓΩΓΗΣ ΠΑΙΧΝΙΔΙΟΥ ΕΡΩΤΗΣΕΩΝ/ΑΠΑΝΤΗΣΕΩΝ ΜΕ ΧΡΗΣΗ ARDUINO - QUIZ GAME"

1^{ο} ΕΠΑ.Λ. ΓΕΡΑΣ

Σχολικό Έτος:

Τάξη: ……..
Τμήμα: ……..
Ομάδα: ……..
Ημερομηνία: ……..

Καθηγητής:

Ονοματεπώνυμο Μαθητών:

ΦΥΛΛΟ ΕΡΓΑΣΙΑΣ 1

Τίτλος: **Κατασκευή πλήρους κυκλώματος του παιχνιδιού Quiz Game και εξάσκηση στον προγραμματισμό των βασικών λειτουργιών του.**

Σκοπός: - Η κατασκευή του κυκλώματος του παιχνιδιού Quiz Game και η εξοικείωση με τις βασικές λειτουργίες του συνοδευτικού κώδικα.

Στόχοι: - Στο τέλος αυτής της εργασίας θα είστε ικανοί:

1. Να κατασκευάζετε το πλήρες κύκλωμα ενός παιχνιδιού ερωτήσεων/ απαντήσεων με χρήση Arduino.
2. Να αναγνωρίζετε τις εισόδους και τις εξόδους σε ένα κύκλωμα Arduino αξιολογώντας τη συνδεσμολογία και τον κώδικά που το συνοδεύει.
3. Να υλοποιείτε κώδικα χρησιμοποιώντας τη δομή επιλογής switch.
4. Να γράφετε τον κώδικα του κυκλώματος έτσι ώστε να αναβοσβήνουν τα Led όταν πατάμε τα μπουτόν.

- ✓ *Διδακτικό σενάριο δημιουργίας συστήματος διεξαγωγής παιχνιδιού ερωτήσεων/απαντήσεων με χρήση Arduino – Quiz Game.*
- ✓ *Σχόλια κώδικα με την περιγραφή λειτουργίας του προγράμματος στο αρχείο QuizGame.ino*
- ✓ "Προγραμματίζοντας με τον μικροελεγκτή Arduino", Πουλάκης Εμ., Ηράκλειο, Ιανουάριος 2015.

Α. ΠΛΗΡΟΦΟΡΙΕΣ

Για την γενική περιγραφή του Quiz Game, θα ανατρέξετε στο σχετικό φύλλο πληροφοριών του διδακτικού σεναρίου.

Αρχικά θα κατασκευάσετε το πλήρες κύκλωμα του παιχνιδιού Quiz Game, σύμφωνα με το **Σχέδιο 1**.

Στη συνέχεια θα εξασκηθείτε στον προγραμματισμό των βασικότερων λειτουργιών του πλήρους παιχνιδιού, μέσα από απλά βήματα που περιγράφονται στις επόμενες ενότητες του φύλλου εργασίας. Σκοπός της εξάσκησης αυτής είναι η κατανόηση της λειτουργίας του πλήρους κώδικα του παιχνιδιού.

Μια από τις βασικότερες δομές που χρησιμοποιείται στο κώδικα του Quiz Game είναι η δομή επιλογής **switch**, η λειτουργία της οποίας περιγράφεται ως εξής:

Δομή επιλογής switch

Όπως η **if** έτσι και η **switch** μας επιτρέπει την διακλάδωση του προγράμματος σε διαφορετικά block κώδικα ανάλογα με την περίπτωση που ισχύει. Επειδή όμως οι περιπτώσεις της **switch** είναι σταθερές και όχι λογικές εκφράσεις, μπορεί να ισχύει μόνο μια περίπτωση, και άρα να εκτελεστεί μόνο ένα τμήμα του κώδικα. Η **switch** συντάσσεται ως εξής:

Σύνταξη:

```
switch (μεταβλητή){
        case σταθερά1:
εντολή_Α;

        break;
        case σταθερά2:
εντολή_Β;

        break;
        case σταθερά3:
εντολή_Γ;

        break;
        default:  εντολή_Δ;
}
```

Παράδειγμα:

```
switch (pin){
        case 1: digitalWrite(pin, HIGH);
                break;
        case 2: digitalWrite(pin, HIGH);
                break;
        case 3: digitalWrite(pin, LOW);
         digitalWrite(pin, LOW);
                 break;
        default: Serial.print("Δεν υπάρχει διαθέσιμο pin");
}
```

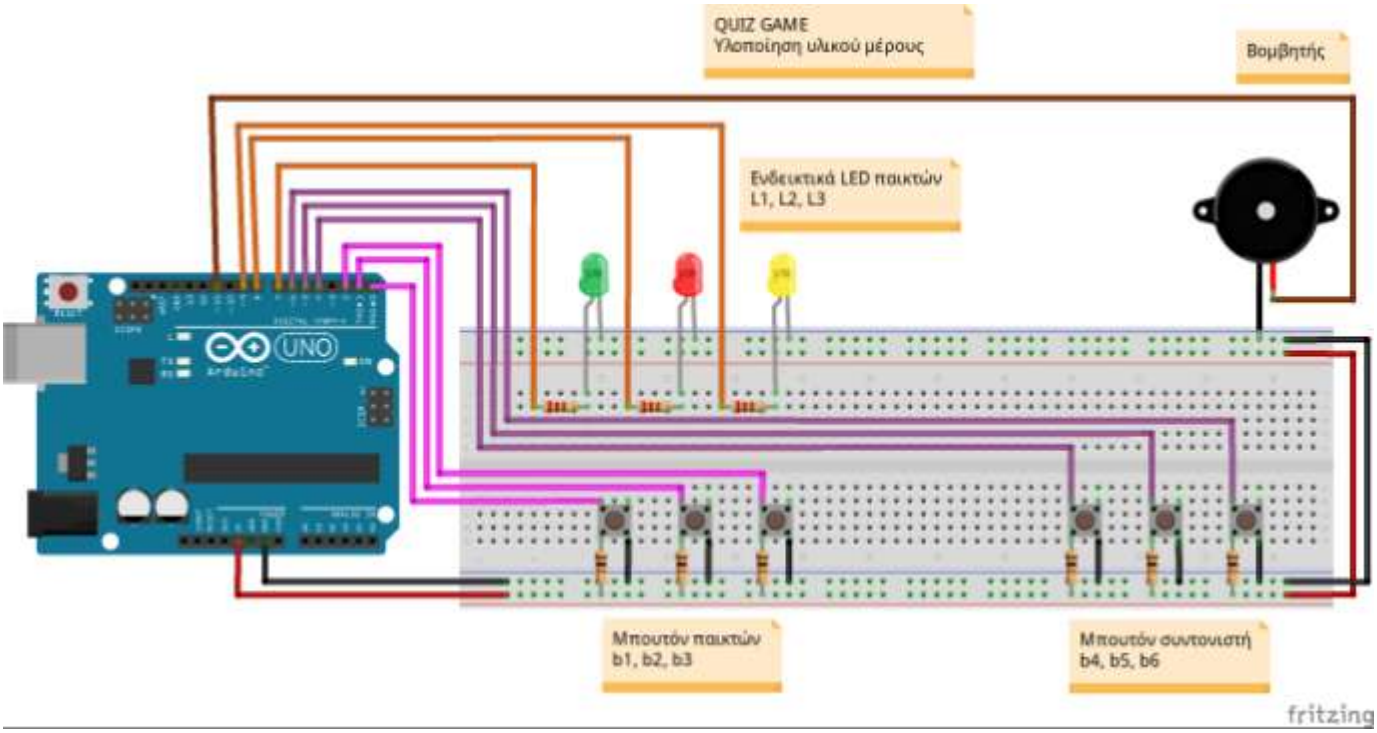
Όπως φαίνεται και στο παράδειγμα, αν το **pin** έχει π.χ. την τιμή 2 τότε εκτελείται η εντολή **digitalWrite(pin, HIGH)** και επειδή μετά βάλαμε την δεσμευμένη λέξη **break**; η ροή του προγράμματος διακόπτεται και συνεχίζει μετά το τέλος της **switch**. Αν η **break** δεν υπήρχε τότε ο κώδικας θα εκτελούσε την εντολή και μετά όλες τις παρακάτω, δηλαδή αν αντιστοιχούσε στην 2, τότε θα εκτελούνταν και η 3 και η **default**. Σημειώστε ότι η **default** εκτελείται αν δεν προκύψει εγκυρότητα κάποιας άλλης περίπτωσης.

Β. ΑΠΑΙΤΟΥΜΕΝΑ ΥΛΙΚΑ

α/α	Υλικά	Ποσότητα	Παρατηρήσεις
1	Μονάδα Arduino Uno R3	1 τεμ.	
2	Πλακέτα Κατασκευών -Breadboard	1 τεμ.	
3	Μπουτόν - Button	6 τεμ.	
4	LED	3 τεμ.	Κατά προτίμηση διαφορετικού χρώματος
5	Ηχείο - Buzzer	1 τεμ.	
6	Αντίσταση 10kΩ	6 τεμ.	
7	Αντίσταση 220Ω	3 τεμ.	
8	USB καλώδιο	1 τεμ.	Για την σύνδεση του Arduino με Η/Υ
9	Καλώδια		Για την πραγματοποίηση των συνδεσμολογιών στο Breadbord

Γ. ΠΟΡΕΙΑ ΕΡΓΑΣΙΑΣ

1. Υλοποιήστε την συνδεσμολογία του πλήρους κυκλώματος του παιχνιδιού Quiz Game, όπως φαίνεται στο **Σχέδιο 1**:



Σχέδιο 1: Πλήρες κύκλωμα παιχνιδιού Quiz Game

- Σύμφωνα με το κύκλωμα αυτό, συμπληρώστε τα κενά των παρακάτω προτάσεων:

α) Ο αριθμός των ψηφιακών εισόδων του Arduino που χρησιμοποιείται στο κύκλωμα είναι.....

β) Στις ψηφιακές εισόδους είναι συνδεδεμένα..... *(συμπληρώστε τον συνολικό αριθμό και το όνομα του/των εξαρτήματος/ων)*

γ) Ο αριθμός των ψηφιακών εξόδων του Arduino που χρησιμοποιείται στο κύκλωμα είναι .....

δ) Στις ψηφιακές εξόδους είναι συνδεδεμένα ..... *(συμπληρώστε τον συνολικό αριθμό και το όνομα του/των εξαρτήματος/ων)*

ε) Ο αριθμός των αναλογικών εξόδων τύπου PWM του Arduino που χρησιμοποιείται στο κύκλωμα είναι .....

στ) Στις αναλογικές εξόδους τύπου PWM είναι συνδεδεμένα..... *(συμπληρώστε τον συνολικό αριθμό και το όνομα του/των εξαρτήματος/ων)*

2. Μέσα από το **Arduino IDE**, ανοίξτε το αρχείο **QuizGameCodeExercise.ino** που βρίσκεται στον φάκελο **QuizGameCodeExercise**. Στη συνέχεια, αναλύστε τον κώδικα με τη βοήθεια του καθηγητού σας, και συμπληρώστε τα παρακάτω στοιχεία:

ΣΤΟΙΧΕΙΑ ΛΕΙΤΟΥΡΓΙΑΣ ΕΙΣΟΔΩΝ/ΕΞΟΔΩΝ		
Εξάρτημα	**Αριθμός pin σύνδεσης**	**Κατάσταση αρχικής λειτουργίας (INPUT ή OUTPUT)**
Μπουτον 1		
Μπουτον 2		
Μπουτον 3		
Μπουτον 4		
Μπουτον 5		
Μπουτον 6		
Led 1		
Led 2		
Led 3		
Βομβητής		

- Το όνομα της συνάρτησης που καλείται στην **setup()** για την αρχικοποίηση των ψηφιακών εισόδων είναι:.....
- Το όνομα της συνάρτησης που καλείται στην **setup()** για την αρχικοποίηση των ψηφιακών εξόδων είναι:.....
- Το όνομα της συνάρτησης που καλείται στην **setup()** για την αρχικοποίηση της συριακής θύρας επικοινωνίας είναι:.....

3. Βρείτε την δομή επιλογής **switch** που χρησιμοποιείται μέσα στην **loop()** και εντοπίστε τις τρεις συναρτήσεις (εντολές):

   α) pressedButtonMsg( )
   β) setLedPinON( ) και,
   γ) setLedPinOFF( )

Οι παραπάνω συναρτήσεις, αν και έχουν οριστεί μέσα στον κώδικα, δεν πραγματοποιούν καμία ενέργεια.
Γράψτε τον κώδικα των συναρτήσεων αυτών έτσι ώστε:

- Όταν πατάμε ένα από τα τρία πρώτα μπουτόν (button 1,2,3):
  α) Καλώντας την συνάρτηση **pressedButtonMsg( )**, να τυπώνεται στην σειριακή οθόνη το μήνυμα με τον αριθμό του μπουτόν που πατήθηκε, π.χ. «The pressed button is: 1»

β) Καλώντας την συνάρτηση **setLedPinON( )**, να ανάβει το Led 1 όταν πατιέται το button 1, το Led 2 όταν πατιέται το button 2 και το Led 3 όταν πατιέται το button 3.

- Όταν πατάμε ένα από τα επόμενα τρία μπουτόν (button 4,5,6):
  α) Καλώντας την συνάρτηση **pressedButtonMsg()**, να τυπώνεται στην σειριακή οθόνη το μήνυμα με τον αριθμό του μπουτόν που πατήθηκε, π.χ. «The pressed button is: 4»
  β) Καλώντας την συνάρτηση **setLedPinOFF ()**, να σβήνει, το Led 1 όταν πατιέται το button 4, το Led 2 όταν πατιέται το button 5 και το Led 3 όταν πατιέται το button 6.

4. Φορτώστε τον κώδικα που γράψατε στο προηγούμενο βήμα και ελέγξτε την λειτουργία του παρουσία του καθηγητή σας.

ΠΑΡΑΤΗΡΗΣΕΙΣ

ΔΙΔΑΚΤΙΚΟ ΣΕΝΑΡΙΟ

"ΔΗΜΙΟΥΡΓΙΑ ΣΥΣΤΗΜΑΤΟΣ ΔΙΕΞΑΓΩΓΗΣ ΠΑΙΧΝΙΔΙΟΥ ΕΡΩΤΗΣΕΩΝ/ΑΠΑΝΤΗΣΕΩΝ ΜΕ ΧΡΗΣΗ ARDUINO - QUIZ GAME"

1^{ο} ΕΠΑ.Λ. ΓΕΡΑΣ

Σχολικό Έτος:

Τάξη: ……..
Τμήμα: ……..
Ομάδα: ……..
Ημερομηνία: ……..

Καθηγητής:

Ονοματεπώνυμο Μαθητών:

ΦΥΛΛΟ ΕΡΓΑΣΙΑΣ 2

Τίτλος: **Προσθήκη 4^{ου} παίκτη στο Quiz Game - Τροποποίηση του πλήρους κυκλώματος και διαδικασία επέκτασης του προγράμματος.**

Σκοπός: - Η κατασκευή του κυκλώματος του παιχνιδιού Quiz Game και η εξοικείωση με τις βασικές λειτουργίες του συνοδευτικού κώδικα.

Στόχοι: - Στο τέλος αυτής της εργασίας θα είστε ικανοί:

1. Να τροποποιείτε το κύκλωμα του Quiz Game για την προσθήκη και 4^{ου} παίκτη στο παιχνίδι.
2. Να τροποποιείτε τον κώδικα του κυκλώματος, έτσι ώστε να ανταποκρίνεται στις αλλαγές που κάνατε στο κύκλωμα.

- ✓ *Διδακτικό σενάριο δημιουργίας συστήματος διεξαγωγής παιχνιδιού ερωτήσεων/απαντήσεων με χρήση Arduino – Quiz Game.*
- ✓ *Σχόλια κώδικα με την περιγραφή λειτουργίας του προγράμματος στο αρχείο QuizGame.ino*
- ✓ "Προγραμματίζοντας με τον μικροελεγκτή Arduino", Πουλάκης Εμ., Ηράκλειο, Ιανουάριος 2015.

Α. ΠΛΗΡΟΦΟΡΙΕΣ

Για την γενική περιγραφή του Quiz Game, θα ανατρέξετε στο σχετικό φύλλο πληροφοριών του διδακτικού σεναρίου.

Σκοπός της σημερινής μας εργασίας είναι να προσθέσουμε στο παιχνίδι έναν ακόμη παίκτη. Αυτό σημαίνει ότι θα πρέπει να εργαστούμε τόσο στο υλικό μέρος (hardware) όσο και στο λογισμικό μέρος (software) του παιχνιδιού.

Για τις αλλαγές στο υλικό μέρος, θα πρέπει να προσδιορίσουμε:

- τα εξαρτήματα που θα χρειαστούμε και
- το τμήμα του κυκλώματος στο οποίο θα γίνουν οι συνδεσμολογίες.

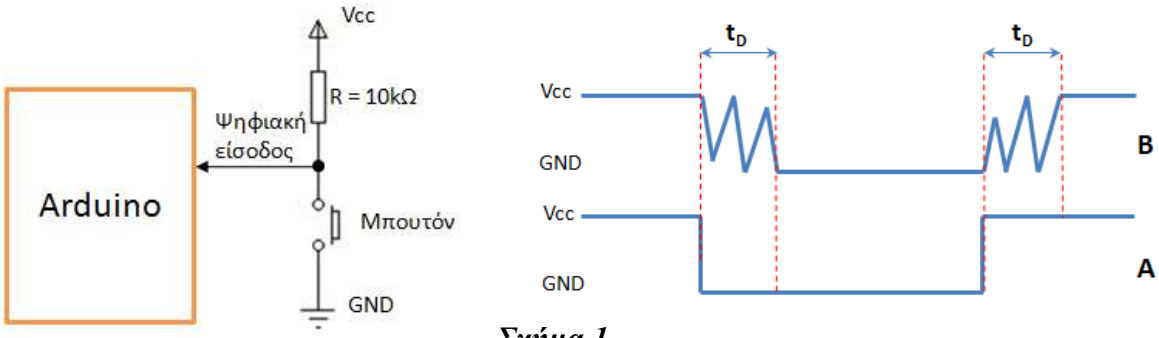
Για τις αλλαγές στο λογισμικό μέρος θα εντοπίσουμε:

- τα σημεία που ορίζονται οι είσοδοι (μπουτόν) και οι έξοδοι (LED) για τον κάθε παίκτη
- το τμήμα του κώδικα που επαναλαμβάνεται για όλους τους παίκτες.

Παρακάτω περιγράφεται το θεωρητικό μέρος της συνδεσμολογίας των ψηφιακών εισόδων και εξόδων του Quiz Game, το οποίο θα μας βοηθήσει περεταίρω στην επιτυχή ολοκλήρωση των αλλαγών που πρέπει να πραγματοποιήσουμε στο κύκλωμα.

ΣΥΝΔΕΣΜΟΛΟΓΙΑ ΤΩΝ ΨΗΦΙΑΚΩΝ ΕΙΣΟΔΩΝ ΤΟΥ QUIZ GAME ΜΕ ΜΠΟΥΤΟΝ

Οι συνδέσεις των εισόδων υλοποιήθηκαν σύμφωνα με το κύκλωμα στο ***Σχήμα 1***. Επομένως, τα μπουτόν του κυκλώματος (3 των παικτών + 3 του παρουσιαστή) είναι συνδεδεμένα στις εισόδους του Arduino μέσω μιας pull-up αντίστασης.



Σχήμα 1

Όσο το μπουτόν δεν είναι πατημένο, η ψηφιακή είσοδος του Arduino συνδέεται στα **Vcc = +5V**, δηλαδή είναι **HIGH**. Όταν το μπουτόν πατηθεί (και όσο αυτό διαρκεί), η είσοδος του Arduino συνδέεται στην γείωση του κυκλώματος **GND**, δηλαδή γίνεται **LOW** (***Σχήμα 1***, Κυματομορφή **A**) . Αυτή η αλλαγή της κατάστασης της εισόδου από **HIGH** σε **LOW** θα έπρεπε ιδανικά να είναι ακαριαία. Αυτό όμως δεν συμβαίνει στην πραγματικότητα. Αυτό που συμβαίνει είναι ότι με το πάτημα του μπουτόν έχουμε αρκετές μεταβάσεις από **LOW** σε **HIGH**. Μεσολαβεί δηλαδή ένα μικρό χρονικό διάστημα $\mathbf{t_D}$ της τάξης μερικών **ms** (συνήθως 5 ως 20ms) όπου η κατάσταση είναι απροσδιόριστη (***Σχήμα 1***, Κυματομορφή **B**). Το φαινόμενο αυτό ονομάζεται **κλυδωνισμός** (Bouncing) και είναι ένα σύνηθες φαινόμενο σε εφαρμογές που χρησιμοποιούνται μπουτόν και διακόπτες. Αυτό μπορεί να είναι η αιτία για μια ανεπιθύμητη ενεργοποίηση ή και απενεργοποίση της εισόδου στην οποία είναι συνδεδεμένο το μπουτόν.

Η διαδικασία αντιμετώπισης του φαινομένου αυτού ονομάζεται **αποκλυδωνισμός** (Debouncing) και μπορεί να γίνει με δύο τρόπους:

α) Με hardware, με χρήση δηλαδή ειδικών ηλεκτρονικών διατάξεων για τον σκοπό αυτό στο κύκλωμα του διακόπτη.

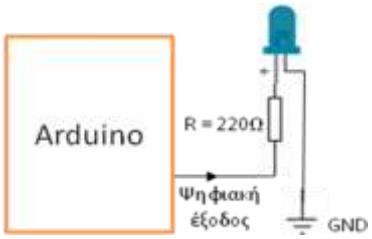
β) Με software, με την προσθήκη δηλαδή μιας ειδικής ρουτίνας-συνάρτησης στον κώδικα του κυκλώματος.

Στο κώδικα του Quiz Game, ο κλυδωνισμός αντιμετωπίζεται με την χρήση των συναρτήσεων **thisSwitch_justPressed( )** και **check_switches( ).** Έτσι, καλώντας μέσα από την **loop( )** την συνάρτηση **thisSwitch_justPressed( )**, αυτή μας επιστρέφει τον αριθμό του μπουτόν που έχει πατηθεί, αφού πρώτα

ελέγχθηκε αν υπάρχει κλυδωνισμός στην είσοδος που αντιστοιχεί σε αυτό. Βασική παράμετρος για την λειτουργία των συναρτήσεων αυτών είναι το χρονικό διάστημα που εξετάζεται για τον αποκλυδωνισμό του μπουτόν. Το χρονικό αυτό διάστημα ονομάζεται **DEBOUNCING TIME** ή απλά **DEBOUNCE** και όπως αναφέραμε προηγουμένως μετριέται σε **ms**.

ΣΥΝΔΕΣΜΟΛΟΓΙΑ ΤΩΝ ΨΗΦΙΑΚΩΝ ΕΞΟΔΩΝ ΤΟΥ QUIZ GAME ΜΕ LED

Η σύνδεση των τριών LED του κυκλώματος στις εξόδους του Arduino είναι αρκετά απλή και πραγματοποιείται σύμφωνα με το ***Σχήμα 2***.



Σχήμα 2

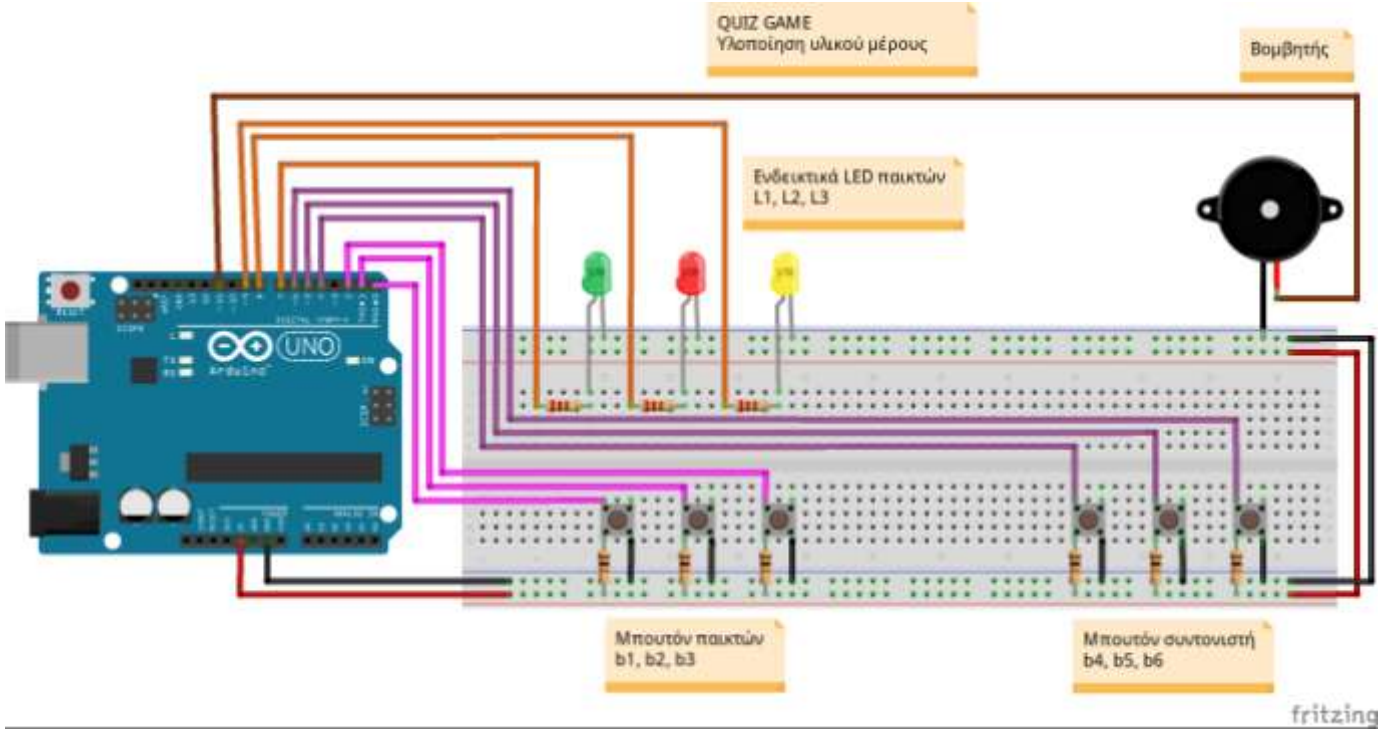
Β. ΑΠΑΙΤΟΥΜΕΝΑ ΥΛΙΚΑ

α/α	Υλικά	Ποσότητα	Παρατηρήσεις
1	Πλήρες κύκλωμα Quiz Game	1 τεμ.	Βλ. Φύλλο Εργασίας 1
2	USB καλώδιο	1 τεμ.	Για την σύνδεση του Arduino με Η/Υ
3	Καλώδια		Για την πραγματοποίηση των συνδεσμολογιών στο Breadbord
4			
5			
6			
7			

Γ. ΠΟΡΕΙΑ ΕΡΓΑΣΙΑΣ

1. Συμβουλευτείτε το ***Σχέδιο 1*** του πλήρους κυκλώματος του **Quiz Game** και τις πληροφορίες της προηγούμενης ενότητας για να προσδιορίσετε τις αλλαγές που πρέπει να κάνετε ώστε να προστεθεί ένας επιπλέον παίκτης στο παιχνίδι. Στην συνέχεια συμπληρώστε τον Πίνακα με τα <u>απαιτούμενα υλικά</u> της ενότητας **Β** με τα εξαρτήματα που θα χρειαστείτε για να πραγματοποιήσετε τις αλλαγές αυτές.

2. Για τη σύνδεση με το Arduino θα χρησιμοποιήσετε ως είσοδο το **pin 3** και ως έξοδο το **pin 10** που είναι ελεύθερα. Για διευκόλυνσή σας, σημειώστε με μολύβι τη συνδεσμολογία που θα κάνετε πάνω στο εικονιζόμενο ***Σχέδιο 1***.



Σχέδιο 1: Πλήρες κύκλωμα παιχνιδιού Quiz Game

3. Αφού προμηθευτείτε τα υλικά που θα χρειαστείτε, πραγματοποιήστε πάνω στο breadboard του Quiz Game τη συνδεσμολογία που σκεφτήκατε. Όταν ολοκληρώσετε όλες τις συνδέσεις, ελέγξτε με την βοήθεια του καθηγητή σας αν τις κάνατε σωστά.

4. Από το **Arduino IDE**, ανοίξτε το αρχείο **QuizGameCodeExercise.ino** που βρίσκεται στον φάκελο **QuizGameCodeExercise**.

5. Τροποποιείστε τον κώδικα του **QuizGameCodeExercise.ino** έτσι ώστε να ανταποκρίνεται στις αλλαγές που κάνατε στο κύκλωμα του Quiz Game, Στον κώδικα θα πρέπει:

   - Όταν πατάμε το μπουτόν **4** (του 4^{ου} παίχτη που προσθέσαμε), να γίνονται οι ίδιες ακριβώς ενέργειες με αυτές που γίνονται όταν πατάμε τα button **1**, **2** και **3**.

   - Όταν πατάμε το μπουτόν **7** (τα μπουτόν του συντονιστή τώρα είναι τα 5, 6 και7):
     α) Η σειριακή οθόνη να τυπώνει το μήνυμα «The pressed button is: 7»
     β) Τα Led **<u>3 και 4</u>** να σβήνουν.

6. Με βάση τις αλλαγές που κάνατε στον κώδικα, συμπληρώστε τα παρακάτω στοιχεία:

   α) Το όνομα και οι νέες τιμές της δομής του πίνακα (array) όπου έχουν ομαδοποιηθεί όλα τα pins εισόδων του κυκλώματος είναι *(Γράψτε τον κώδικα που αντιστοιχεί στην αλλαγή αυτή):* …….....

   β) Το όνομα και οι νέες τιμές της δομής του πίνακα (array) όπου έχουν ομαδοποιηθεί τα pins εξόδων των Led του κυκλώματος είναι *(Γράψτε τον κώδικα που αντιστοιχεί στην αλλαγή αυτή):*…….....

   γ) Για να αλλάξετε τον χρόνο αποκλυδωνισμού (debounce) των μπουτόν από 10ms που είναι σε 20ms, ο κώδικας που πρέπει να γράψετε είναι:…….....

   δ) Κάνατε αλλαγή στην δομή επιλογής **switch** (ΝΑΙ/ΟΧΙ);……

   Αν ναι, πόσες είναι οι επιλογές της δομής τώρα; ……...

   Καταγράψετε τον κώδικα εφόσον έχετε προσθέσει νέα επιλογή:

   case ….:

   break;

7. Φορτώστε στο **Arduino** τον κώδικα που τροποποιήσατε στο βήμα (5) και ελέγξτε την λειτουργία του, παρουσία του καθηγητή σας.

ΠΑΡΑΤΗΡΗΣΕΙΣ

ΔΙΔΑΚΤΙΚΟ ΣΕΝΑΡΙΟ

"ΔΗΜΙΟΥΡΓΙΑ ΣΥΣΤΗΜΑΤΟΣ ΔΙΕΞΑΓΩΓΗΣ ΠΑΙΧΝΙΔΙΟΥ ΕΡΩΤΗΣΕΩΝ/ΑΠΑΝΤΗΣΕΩΝ ΜΕ ΧΡΗΣΗ ARDUINO - QUIZ GAME"

1^{ο} ΕΠΑ.Λ. ΓΕΡΑΣ

Σχολικό Έτος:

Τάξη: ……..
Τμήμα: ……..
Ομάδα: ……..
Ημερομηνία: ……..

Καθηγητής:

Ονοματεπώνυμο Μαθητών:

ΦΥΛΛΟ ΕΡΓΑΣΙΑΣ 3

Τίτλος: **Προσθήκη σειριακής LCD οθόνης στο παιχνίδι Quiz Game – Τροποποίηση κυκλώματος και κώδικα.**

Σκοπός: - Η τροποποίηση του κυκλώματος και του κώδικα του Quiz Game, ώστε τα μηνύματα του παιχνιδιού να εμφανίζονται σε μια σειριακή LCD οθόνη.

Στόχοι: - Στο τέλος αυτής της εργασίας θα είστε ικανοί:

1. Να τροποποιείτε το κύκλωμα του Quiz Game με σκοπό την προσθήκη μιας σειριακής LCD οθόνης.
2. Να τροποποιείτε τον κώδικα του κυκλώματος, έτσι ώστε να ανταποκρίνεται στις αλλαγές που κάνατε στο κύκλωμα.
3. Να γράφετε κώδικα χρησιμοποιώντας την βιβλιοθήκη ***LiquidCrystal_I2C***

- ✓ *Διδακτικό σενάριο δημιουργίας συστήματος διεξαγωγής παιχνιδιού ερωτήσεων/απαντήσεων με χρήση Arduino – Quiz Game.*
- ✓ *Σχόλια κώδικα με την περιγραφή λειτουργίας του προγράμματος στο αρχείο QuizGame.ino*
- ✓ "Προγραμματίζοντας με τον μικροελεγκτή Arduino", Πουλάκης Εμ., Ηράκλειο, Ιανουάριος 2015.
- ✓ Ιστότοπος LCD-Blue-I2C: https://arduino-info.wikispaces.com/LCD-Blue-I2C

Α. ΠΛΗΡΟΦΟΡΙΕΣ

Για την γενική περιγραφή του Quiz Game, θα ανατρέξετε στο σχετικό φύλλο πληροφοριών του διδακτικού σεναρίου.

Ως τώρα, τα μηνύματα του Quiz Game στέλνονται μέσω της θύρας USB του Arduino για να εμφανίζονται σε μια σειριακή οθόνη ενός Υ/Η. Σκοπός της σημερινής μας εργασίας είναι να αλλάξουμε τον τρόπο αυτό έτσι ώστε τα μηνύματα πλέον να εμφανίζονται σε μια σειριακή LCD οθόνη, όπως αυτή που εικονίζεται στο ***Σχήμα 1***.

Σχήμα 1: Άρθρωμα σειριακής LCD οθόνης:

Άρθρωμα σειριακής οθόνης LCD (I²C +LCD)

Η επικοινωνία με την σειριακή LCD οθόνη θα πραγματοποιείται μέσω του διαύλου **I2C** τον οποίο διαθέτει το Arduino.

Τι είναι όμως το I2C;

Ο δίαυλος I**2C**, είναι ένας σειριακός δίαυλος που δημιουργήθηκε από τη **Philips** και χρησιμοποιείται για την σύνδεση περιφερειακών μικρής ταχύτητας σε διάφορα συστήματα μικρουπολογιστών και μικροελεγκτών με εφαρμογές στα αυτοκίνητα, στα κινητά τηλέφωνα και σε πολλές άλλες ηλεκτρονικές συσκευές. Ο δίαυλος **I2C** μπορεί να χρησιμοποιηθεί τόσο για την επικοινωνία συσκευών που βρίσκονται πάνω σε ένα τυπωμένο κύκλωμα όσο και για την επικοινωνία συσκευών που συνδέονται με καλώδια.

Το **I2C** ακολουθεί το πρωτόκολλο **TWI** (two wire interface). Απαιτεί δηλαδή μόνο δύο καλώδια για την επικοινωνία μεταξύ 2 ή περισσότερων συσκευών που είναι συνδεδεμένες στον δίαυλο.

Η επικοινωνία είναι αμφίδρομης κατεύθυνσης και γίνεται μέσω των γραμμών **SCL** και **SDA**.

Η γραμμή **SCL** είναι η γραμμή ρολογιού, ενώ η **SDA** είναι η γραμμή δεδομένων. Στις γραμμές αυτές συνδέουμε όλες τις συσκευές με τις οποίες θέλουμε να επικοινωνήσουμε μέσω του **I2C.** Προφανώς εκτός από τα παραπάνω καλώδια, απαιτείται ένα τρίτο καλώδιο, το οποίο είναι η γείωση (**GND**) ή **0V**. Επίσης, απαιτείται και ένα τέταρτο καλώδιο το οποίο είναι η γραμμή τροφοδοσίας, με την οποία τροφοδοτούνται με ισχύ οι διάφορες συσκευές που συνδέονται στον δίαυλο. Έτσι καταφέρνουμε να μειώσουμε τα καλώδια σύνδεσης π.χ. μιας LCD οθόνης, από 10 που χρειάζονται αρχικά χωρίς τη χρήση του διαύλου **I2C**, σε μόνο 2 (+2 για **Vcc** και GND).

Στο Arduino, ο **I2C** δίαυλος είναι διαθέσιμος μέσα από τα αναλογικά **pin A4** και **A5**. Στο **pin A4** είναι η γραμμή **SDA** και στο **pin A5**, η γραμμή **SCL**.

Πως θα προγραμματίσω την σειριακή LCD Οθόνη;

Βασική προϋπόθεση για να πραγματοποιηθούν τα παραπάνω, μέσω του κώδικα στο Arduino, είναι η χρησιμοποίηση της βιβλιοθήκης **LiquidCrystal_I2C** και της βιβλιοθήκης **Wire** οι οποίες πρέπει να βρίσκονται στο φάκελο **libraries** του προγράμματος ώστε να είναι διαθέσιμες στο κύριο πρόγραμμα.

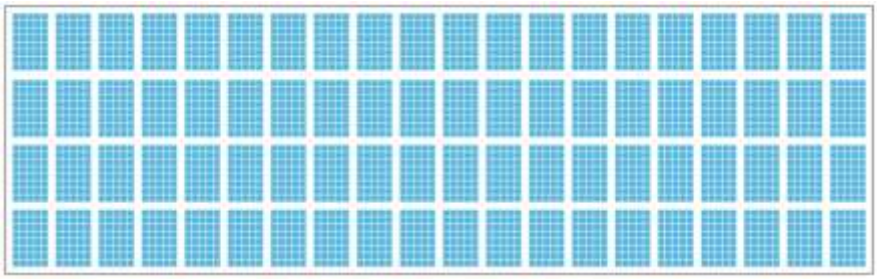
Το πρώτο βήμα μας στον κώδικα είναι να δηλώσουμε τις παραπάνω βιβλιοθήκες και να ορίσουμε τις παραμέτρους της **I2C** επικοινωνίας με την συσκευή που έχουμε συνδέσει στον δίαυλο. Στον κώδικά μας αυτό γίνεται γράφοντας:

```
#include <Wire.h>
#include <LiquidCrystal_I2C.h>

LiquidCrystal_I2C lcd(0x27, 2, 1, 0, 4, 5, 6, 7, 3, POSITIVE); // set the LCD address to 0x27
```

Κάθε συσκευή που συνδέεται στον δίαυλο **I2C** έχει την δικιά της διεύθυνση. Έτσι, στην εφαρμογή μας η σειριακή **LCD** οθόνη που θα συνδέσουμε έχει την διεύθυνση **0x27** (το πρώτο όρισμα στην παραπάνω δήλωση της ***lcd***).

Το επόμενο βήμα είναι να αρχικοποιήσουμε την **I2C** επικοινωνία, λαμβάνοντας υπόψη τα διάφορα χαρακτηριστικά που έχει η συσκευή που συνδέσαμε. Στην περίπτωση μας, η αρχικοποίηση αυτή γίνεται μέσα στην **setup( )**, καλώντας την συνάρτηση **SetupI2cLcdInterfaceForMsg()**. Ανάμεσα στις γραμμές του κώδικα της συνάρτησης αυτής είναι και η **lcd.begin(20,4)**, η οποία αρχικοποιεί την LCD οθόνη, λαμβάνοντας υπόψη ότι αυτή αποτελείται από χαρακτήρες των **τεσσάρων (4) γραμμών** και **είκοσι (20) στηλών** (βλ. ***Σχήμα 2***).

Σχήμα 2: LCD οθόνη 4x20:



Με την δήλωση της βιβλιοθήκης **LiquidCrystal_I2C** στον κώδικα, γίνονται διαθέσιμες και οι παρακάτω συναρτήσεις:

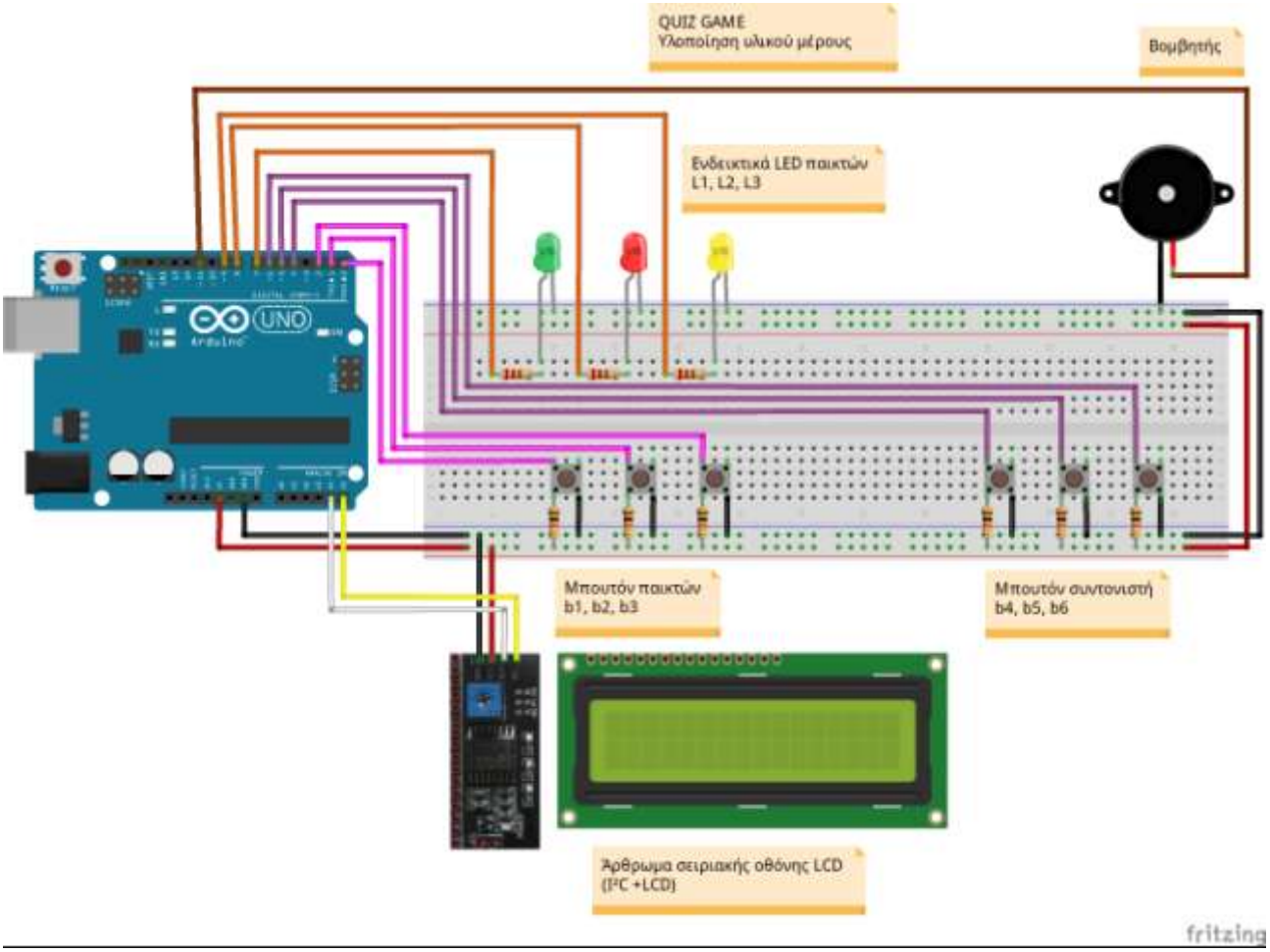
Συνάρτηση βιβλιοθήκης LiquidCrystal_I2C	**Περιγραφή λειτουργίας**
lcd.print(“Κείμενο”)	Τυπώνει το “κείμενο”
lcd.clear( )	Καθαρίζει την οθόνη
lcd.setCursor(x, y)	Τοποθετεί τον κερσορα στις συντεταγμένες x, y
lcd.blink( )	Αναβοσβήνει το κείμενο της οθόνης
lcd.scrollDisplayLeft( )	Μετατοπίζει το κείμενο αριστερά
lcd.scrollDisplayRight( )	Μετατοπίζει το κείμενο δεξιά
lcd.autoscroll( )	Ενεργοποιεί την αυτόματη μετατόπιση του κειμένου μια θέση αριστερά κάθε φορά που στέλνουμε ένα γράμμα στην οθόνη.
lcd noAutoscroll (“Κείμενο”)	Απενεργοποιεί την αυτόματη μετατόπιση
lcd.setBacklight( )	Ενεργοποιεί τον πίσω φωτισμό
lcd.noBacklight( )	Απενεργοποιεί τον πίσω φωτισμό
lcd.create( )	Δημιουργεί ένα νέο χαρακτήρα

Β. ΑΠΑΙΤΟΥΜΕΝΑ ΥΛΙΚΑ

α/α	Υλικά	Ποσότητα	Παρατηρήσεις
1	Πλήρες κύκλωμα Quiz Game	1 τεμ.	Βλ. Φύλλα Εργασίας 1 και 2
2	Άρθρωμα σειριακής LCD Οθόνης	1 τεμ.	Η LCD οθόνη του εργαστηρίου με I2C διεύθυνση 0x27 και 4x20 χαρακτήρες.
3	USB καλώδιο	1 τεμ.	Για την σύνδεση του Arduino με Η/Υ
4	Καλώδια		Για την πραγματοποίηση των συνδεσμολογιών στο Breadbord

Γ. ΠΟΡΕΙΑ ΕΡΓΑΣΙΑΣ

1. Συμβουλευτείτε τις πληροφορίες της προηγούμενης ενότητας και στην συνέχεια πραγματοποιείστε την συνδεσμολογία της σειριακής LCD οθόνης στο κύκλωμα του Quiz Game σύμφωνα με το ***Σχέδιο 2.***



Σχέδιο 2: Πλήρες κύκλωμα παιχνιδιού Quiz Game με προσθήκη σειριακής LCD οθόνης

Εντοπίστε την **I2C** μονάδα της σειριακής LCD οθόνης που βρίσκεται στο πίσω της μέρος (βλ. ***Σχήμα 2)*** και πραγματοποιείστε τις συνδέσεις μεταξύ των **pin** του Arduino και της οθόνης σύμφωνα με τον παρακάτω πίνακα:

Pin Arduino	*Pin I2C μονάδας LCD*
A4	*SDA*
A5	*SCL*
5V	*Vcc*
GND	*GND*

Σχήμα 3: I2C μονάδα σειριακής LCD οθόνης

2. Από το **Arduino IDE**, ανοίξτε το αρχείο **QuizGameCodeExercise.ino** που βρίσκεται στον φάκελο **QuizGameCodeExercise**.

3. Στον κώδικα του **QuizGameCodeExercise.ino** για την αποστολή μηνυμάτων έχουν οριστεί οι εξής συναρτήσεις:

Συνάρτηση μηνυμάτων μέσω USB και σειριακής οθόνης Y/H	**Περιγραφή λειτουργίας**	**Συνάρτηση μηνυμάτων μέσω I2C και σειριακής LCD οθόνης**	**Περιγραφή λειτουργίας**
SetupInterfaceForMsg( )	Αρχικοποιεί την σειριακή πόρτα με ταχύτητα 9600 bps	SetupI2cLcdInterfaceForMsg( )	Αρχικοποιεί την LCD οθόνη των 4x20 χαρακτήρων
pressedButtonMsg(x)	Στέλνει το μήνυμα με τον αριθμό (x) του μπουτόν που πατήθηκε	pressedButtonLCDMsg(x)	Στέλνει το μήνυμα με τον αριθμό (x) του μπουτόν που πατήθηκε

Στην αριστερή στήλη είναι οι συναρτήσεις που χρησιμοποιούνται για την αποστολή μηνυμάτων στην σειριακή οθόνη του Η/Υ ενώ στην δεξιά οι αντίστοιχες συναρτήσεις που θα χρειαστούμε για την αποστολή των μηνυμάτων στην σειριακή LCD οθόνη που συνδέσαμε στο κύκλωμα.

- Αρχικά εντοπίστε μέσα στην loop() την συνάρτηση **SetupInterfaceForMsg()** και στην θέση της καλέστε την **SetupI2cLcdInterfaceForMsg( )**

- Στη συνέχεια συμπληρώστε παρακάτω τον κώδικα της συνάρτησης **pressedButtonLCDMsg**, έτσι ώστε στην **πρώτη γραμμή** της οθόνης να εμφανίζεται ο <u>αριθμός του μπουτόν που πατήθηκε</u> (όμοια με την **pressedButtonMsg**). Ενημερώστε ανάλογα και τον κώδικα στο αρχείο **QuizGameCodeExercise.ino**:

```
void pressedButtonLCDMsg(int buttonNumber){



}
```

4. Εντοπίστε μέσα στην **loop( )** την συνάρτηση **pressedButtonMsg** και στην θέση της καλέστε αυτή που γράψατε στο **βήμα 3**. Αποθηκεύστε τις αλλαγές και φορτώστε στο **Arduino** τον κώδικα για να ελέγξτε την λειτουργία του, παρουσία του καθηγητή σας.

5. Συμπληρώστε παρακάτω τον κώδικα της συνάρτησης **pressedButtonLCDMsg** και στην συνέχεια μεταφέρτε τον στο αρχείο **QuizGameCodeExercise.ino**, έτσι ώστε όταν πατιούνται τα τρία μπουτόν του συντονιστή να πραγματοποιούνται τα εξής:

   - να υλοποιείται χρονοκαθυστέρηση **5** δευτερολέπτων και
   - στην τρίτη γραμμή των χαρακτήρων της οθόνης να τυπώνεται το μήνυμα της χρονοκαθυστέρησης αυτής : "*The time is: **x** sec*", όπου x είναι ο χρόνος από **0** ως **5** sec.

   *void **pressedButtonLCDMsg**(int buttonNumber){*

   }

6. Αποθηκεύστε τις αλλαγές και φορτώστε στο **Arduino** τον κώδικα για να ελέγξτε την λειτουργία του, παρουσία του καθηγητή σας.

ΠΑΡΑΤΗΡΗΣΕΙΣ

ΔΙΔΑΚΤΙΚΟ ΣΕΝΑΡΙΟ

“ΔΗΜΙΟΥΡΓΙΑ ΣΥΣΤΗΜΑΤΟΣ ΔΙΕΞΑΓΩΓΗΣ ΠΑΙΧΝΙΔΙΟΥ ΕΡΩΤΗΣΕΩΝ/ΑΠΑΝΤΗΣΕΩΝ ΜΕ ΧΡΗΣΗ ARDUINO - QUIZ GAME”

1^{ο} ΕΠΑ.Λ. ΓΕΡΑΣ

Σχολικό Έτος:

Τάξη: ……..
Τμήμα: ……..
Ομάδα: ……..
Ημερομηνία: ……..

Καθηγητής:

Ονοματεπώνυμο Μαθητών:

ΦΥΛΛΟ ΕΡΓΑΣΙΑΣ

Τίτλος: **Εφαρμογή του παιχνιδιού Quiz Game στο μάθημα των Αγγλικών σε επαναληπτικό μάθημα ανακεφαλαίωσης λεξιλογίου.**

Σκοπός: - Η επανάληψη και εμπέδωση του αγγλικού λεξιλογίου που έχουν διδαχθεί τα παιδιά σε προηγούμενα μαθήματα.

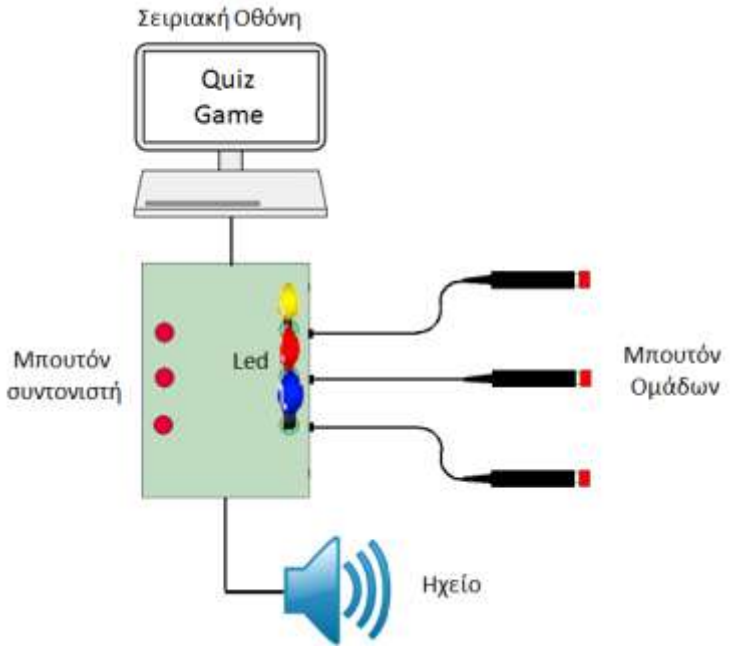
Στόχοι: - Στο τέλος αυτής της εργασίας θα είστε ικανοί:

1. Να δίνετε μετάφραση των αγγλικών λέξεων που διδαχθήκατε σε προηγούμενα μαθήματα.
2. Να χρησιμοποιείτε τις λέξεις που διδαχθήκατε σε προηγούμενα μαθήματα και να διατυπώνετε προτάσεις από αυτές.
3. Να κάνετε χρήση του κυκλώματος του παιχνιδιού Quiz Game.

- ✓ *Διδακτικό σενάριο δημιουργίας συστήματος διεξαγωγής παιχνιδιού ερωτήσεων/απαντήσεων με χρήση Arduino – Quiz Game.*
- ✓ “Προγραμματίζοντας με τον μικροελεγκτή Arduino”, Πουλάκης Εμ., Ηράκλειο, Ιανουάριος 2015.

Α. ΠΛΗΡΟΦΟΡΙΕΣ

Για την αξιοποίηση του παιχνιδιού Quiz Game στη διδακτική διαδικασία σε κάποιο μάθημα π.χ. των Αγγλικών θα χρειαστεί να έχει προσαρμοστεί το παιχνίδι για συνθήκες τάξης. Θα έχει διαμορφωθεί ένα πάνελ αποτελούμενο από κουμπιά χειρισμού (arcade buttons). Η συνολική διάταξη θα έχει την μορφή του ***Σχήματος 1.***



Σχήμα 1: Διάταξη Quiz Game προσαρμοσμένη σε συνθήκες τάξης

Εξηγούμε στους μαθητές ότι το μάθημα θα επικεντρωθεί στην επανάληψη του λεξιλογίου αγγλικών των προηγούμενων μαθημάτων και ότι αυτό θα γίνει με τη βοήθεια του παιχνιδιού Quiz Game.

Τα παιδιά χωρίζονται σε 3 ομάδες των 4 ατόμων και μοιράζουν μεταξύ τους, τους παρακάτω ρόλους μέσα στην κάθε ομάδα:

1. Όλοι θα γίνουν εναλλάξ χειριστές button ομάδας, οπότε καθορίζουν τη σειρά των παικτών τους. Παίκτης 1, Παίκτης 2, Παίκτης 3 και Παίκτης 4.
2. Ένας παίκτης ορίζεται από την ομάδα σαν αρχηγός και μπορεί να κάνει ένσταση αν κάτι δεν λειτουργήσει σωστά στη διαδικασία του παιχνιδιού.
3. Ο εκπαιδευτικός θα έχει αρχικά το ρόλο του συντονιστή του παιχνιδιού με σκοπό, σταδιακά ο ρόλος του συντονιστή να δοθεί στους μαθητές.

Αρχικά γίνεται εξήγηση των κανόνων του παιχνιδιού:

- Το παιχνίδι θα αποτελείται από 3 γύρους των τεσσάρων ερωτήσεων (όσοι είναι και οι παίκτες των ομάδων).
- Σε κάθε γύρο η κάθε ομάδα έχει από έναν ενεργό παίκτη που θα χειρίζεται το arcade button της ομάδας και θα πρέπει να απαντήσει στην ερώτηση.
- Σε κάθε γύρο ο συντονιστής απευθύνεται στο σύνολο των παικτών-χειριστών buttons διατυπώνοντας μια ερώτηση/λέξη.

- Ο παίκτης που θα πατήσει πρώτος το button του, έχει το δικαίωμα να απαντήσει. Η σωστή απάντηση κερδίζει πόντους για την ομάδα του. Για να είναι σωστή μια απάντηση θα πρέπει ο παίκτης να δώσει την ερμηνεία της λέξης και να την χρησιμοποιήσει σε μια πρόταση στα αγγλικά.
- Ο συντονιστής αποφασίζει αν είναι σωστή η απάντηση και σε περίπτωση λάθους δίνει μια ολοκληρωμένη σωστή απάντηση.
- Επαναλαμβάνονται τα βήματα μέχρι να συμπληρωθούν 12 ερωτήσεις (για να παίξουν τις ίδιες φορές όλοι οι παίκτες).

Οι λέξεις/ερωτήσεις προέρχονται από μια λίστα ερωτήσεων/ πίνακα ερωτήσεων.

	A	B	C	D	E	F	G
1	detect	transmit	purchase	precise	frequency	initiative	rumbling
2	evolve	effort	advance	hire	earn	stable	harm
3	treat	cosmetic	benefit	spare	opponent	eager	order
4	donate	salary	proud	party	sympathy	gesture	land

Οι κανόνες μπορούν να αναπροσαρμοστούν ανάλογα με το μέγεθος της τάξης.

Σχολική Μονάδα: 1ο ΕΠΑΛ Μούδρου Λήμνου

Συμμετέχοντες Καθηγητές

ΖΩΗ ΑΝΑΣΤΑΣΙΑ

ΣΧΕΔΙΟ ΜΑΘΗΜΑΤΟΣ

THEREMIN ΜΕ ΦΩΤΟΑΝΤΙΣΤΑΣΗ ΚΑΙ ARDUINO

ΣΧΟΛΕΙΟ: 1^{Ο} ΕΠΑ.Λ ΜΟΥΔΡΟΥ

ΕΚΠΑΙΔΕΥΤΙΚΟΣ: ΑΝΑΣΤΑΣΙΑ ΖΩΗ

ΤΟΜΕΑΣ – ΤΑΞΗ : ΗΛΕΚΤΡΟΝΙΚΏΝ – Β

ΜΑΘΗΜΑ: ΤΕΧΝΟΛΟΓΙΑ ΔΙΑΤΑΞΕΩΝ ΗΛΕΚΤΡΟΝΙΚΗΣ

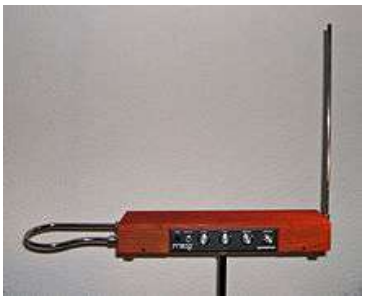
ΔΙΑΡΚΕΙΑ:2 ΔΙΔΑΚΤΙΚΕΣ ΩΡΕΣ

ΣΚΟΠΟΣ Να κατασκευαστεί ένα οπτικό Theremin, δηλ. ένα ηλεκτρονικό κύκλωμα το οποίο να παράγει μουσική καθώς αλλάζει το φως που πέφτει πάνω σε μια φωτοαντίσταση.

ΕΙΔΙΚΟΤΕΡΟΙ ΣΤΟΧΟΙ:

- Να γνωρίσουν οι μαθητές τη λειτουργία της φωτοαντίστασης
- Να γνωρίσουν οι μαθητές τη λειτουργία του πιεζοηλεκτρικού βομβητή
- Να γνωρίσουν οι μαθητές την λειτουργία της αναλογικής εισόδου του Arduino
- Να εξασκηθούν οι μαθητές στη χρήση της σειριακής οθόνης του Arduino IDE
- Να συναρμολογήσουν ένα απλό κύκλωμα στο Arduino
- Να φορτώσουν τον κώδικα και να ελέγξουν την λειτουργία του
- Να εξασκηθούν στην λειτουργία της εντολής tone

ΕΙΣΑΓΩΓΗ



Theremin: Το **θέρεμιν** (αλλιώς θερεμίνη, αρχικά αιθερόφωνο) είναι ένα ηλεκτρονικό μουσικό όργανο που εφευρέθηκε το 1919. Είναι το μόνο διαδεδομένο μουσικό όργανο το οποίο μπορεί να παιχτεί εκ του μακρόθεν, δηλ. «στον αέρα», χωρίς να το αγγίξεις. Η θερεμίνη είναι ένα κουτί, από το οποίο προεξέχουν δύο κεραίες, που δημιουργούν ένα ηλεκτρομαγνητικό πεδίο. Με την κίνηση του χεριού του ανάμεσα στις κεραίες και το ηλεκτρικό δυναμικό του σώματός του, ο μουσικός επιδρά στο ηλεκτρομαγνητικό πεδίο. Οι μεταβολές αυτές του ηλεκτρομαγνητικού πεδίου αναγνωρίζονται, ενισχύονται και αναπαράγονται από τις κεραίες ως ήχος.

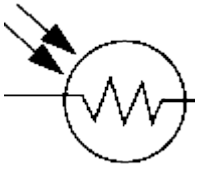
https://www.youtube.com/watch?v=K6KbEnGnymk

Στο μάθημα αυτό θα κατασκευάσουμε ένα <οπτικό> theremin, το οποίο θα παράγει διαφορετικό ήχο όταν αλλάζει η ένταση του φωτός που πέφτει σε μια φωτοαντίσταση.

ΘΕΩΡΗΤΙΚΟ ΜΕΡΟΣ

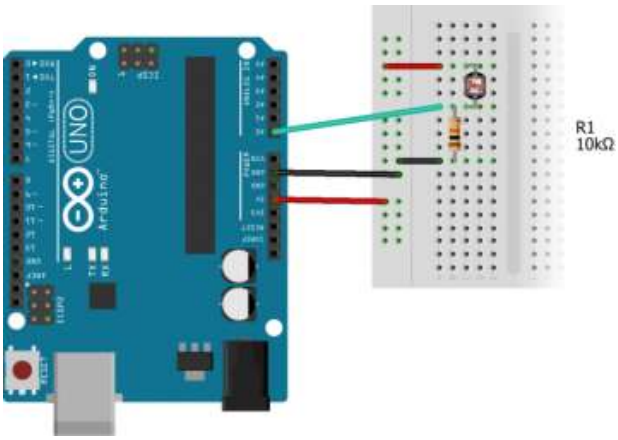
Λειτουργία φωτοαντίστασης (LDR – Light Depended Resistors): Οι φωτοαντιστάσεις είναι αντιστάσεις που κατασκευάζονται από ειδικούς ημιαγωγούς και στις οποίες ελαττώνεται η αντίσταση όταν στην επιφάνειά τους πέσει φως. Η ονομαστική τιμή τους στο σκοτάδι είναι αρκετά μεγάλη (ΜΩ) και ονομάζεται τιμή σκότους. Η τιμή αυτή μειώνεται όταν στην επιφάνειά τους πέσει φως. Κατασκευάζεται από ημιαγωγά υλικά όπως θειούχο κάδμιο (CdS), σελήνιο (Se), ενώσεις μολύβδου και θείου κ.α.

Φωτοαντίσταση



Σύμβολο

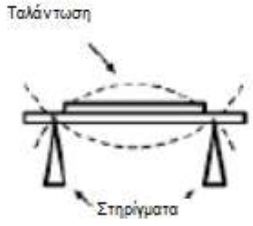
Διαιρέτης τάσης με φωτοαντίσταση –Σύνδεση στην αναλογική είσοδο του Arduino



Στο διπλανό κύκλωμα η φωτοαντίσταση είναι συνδεδεμένη σε σειρά με μία αντίσταση 10ΚΩ σχηματίζοντας έναν διαιρέτη τάσης. Η μία άκρη του κυκλώματος (μαύρο καλώδιο) είναι ενωμένο στα 0V ενώ η άλλη (κόκκινο καλώδιο) είναι ενωμένη σε τάση 5V. Κατ' αυτό τον τρόπο σχηματίζεται ένας διαιρέτης τάσης, όπου στο στο σημείο σύνδεσης των δύο εξαρτημάτων (γαλάζιο καλώδιο) μπορούμε να μετρήσουμε οποιαδήποτε τιμή τάσης από 0 έως 5V, ανάλογα με την ωμική τιμή της φωτοαντίστασης που με τη σειρά της εξαρτάται από το φως που πέφτει πάνω της. Το γαλάζιο καλώδιο οδηγείται στην αναλογική είσοδο A0 του Arduino όπου ο ενσωματωμένος στο Arduino μετατροπέας από αναλογικό σε ψηφιακό (ADC – Analog to Digital Converter) μετατρέπει την αναλογική τάση σε έναν ακέραιο αριθμό από 0 έως 1023. Όταν η φωτοαντίσταση εκτίθεται στο φως τότε η ωμική της τιμή πέφτει επομένως η τάση που οδηγείται στην είσοδο A0 είναι μεγαλύτερη και αντίστροφα.

Πιεζοηλεκτρικός βομβητής: Ο πιεζοηλεκτρικός βομβητής (piezo buzzer) είναι ένα εξάρτημα το οποίο κατασκευάζεται από ειδικά κρυσταλλικά υλικά και η λειτουργία του βασίζεται στο πιεζοηλεκτρικό φαινόμενο. Όταν στα άκρα του εφαρμόζεται τάση τότε ταλαντώνεται και παράγει ηχητικό κύμα.

Piezo buzzer



Εσωτερική κατασκευή

ΠΡΑΚΤΙΚΟ ΜΕΡΟΣ

ΥΛΙΚΑ & ΟΡΓΑΝΑ

- Πλακέτα Arduino Uno
- Φωτοαντίσταση
- Αντίσταση 10ΚΩ
- Πιεζοηλεκτρικός βομβητής (piezo buzzer)
- Καλώδια σύνδεσης
- Breadboard
- Ωμόμετρο
- Γεννήτρια συχνοτήτων

ΠΟΡΕΙΑ ΕΡΓΑΣΙΑΣ

1. Συνδέστε την φωτοαντίσταση σε ένα ωμόμετρο και μετρήστε την ωμική της αντίσταση για διάφορες φωτεινές καταστάσεις. Καταγράψτε τις τιμές που μετρήσατε στον παρακάτω πίνακα. (Σημ. η φωτοαντίσταση δεν έχει πολικότητα)

Κατάσταση	**Καλυμμένη με το δάκτυλο**	**Καλυμμένη με χαρτί**	**Σε φυσικό φως**	**Φωτισμένη με φακό**
Ωμική Αντίσταση				

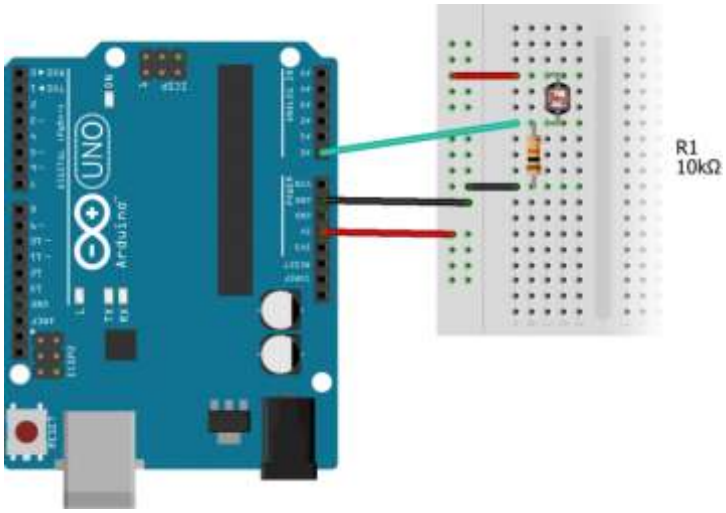
Τι παρατηρείτε; ……

……

…….

2. Συνδέστε τον πιεζοηλεκτρικό βομβητή στην έξοδο γεννήτριας συχνοτήτων και δώστε διάφορες τιμές συχνοτήτων από 100 Hz έως 8 KHz, τετραγωνικής μορφής, κρατώντας την τάση εξόδου της γεννήτριας σταθερή. Ακούστε τους διάφορους τόνους που παράγονται. Σε ποια συχνότητα περίπου παράγεται ο δυνατότερος σε ένταση ήχος; ……..

3. Συναρμολογήστε το παρακάτω κύκλωμα χρησιμοποιώντας την πλακέτα Arduino Uno



Φορτώστε στο Arduino το παρακάτω πρόγραμμα

```
int lightPin = 0;

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

}

void loop()
{
   Serial.println(analogRead(lightPin));
   delay(10);
}
```

Ανοίξτε την σειριακή οθόνη του Arduino IDE και σημειώστε την τιμή που διαβάζετε στην σειριακή οθόνη για διάφορες τιμές φωτισμού.

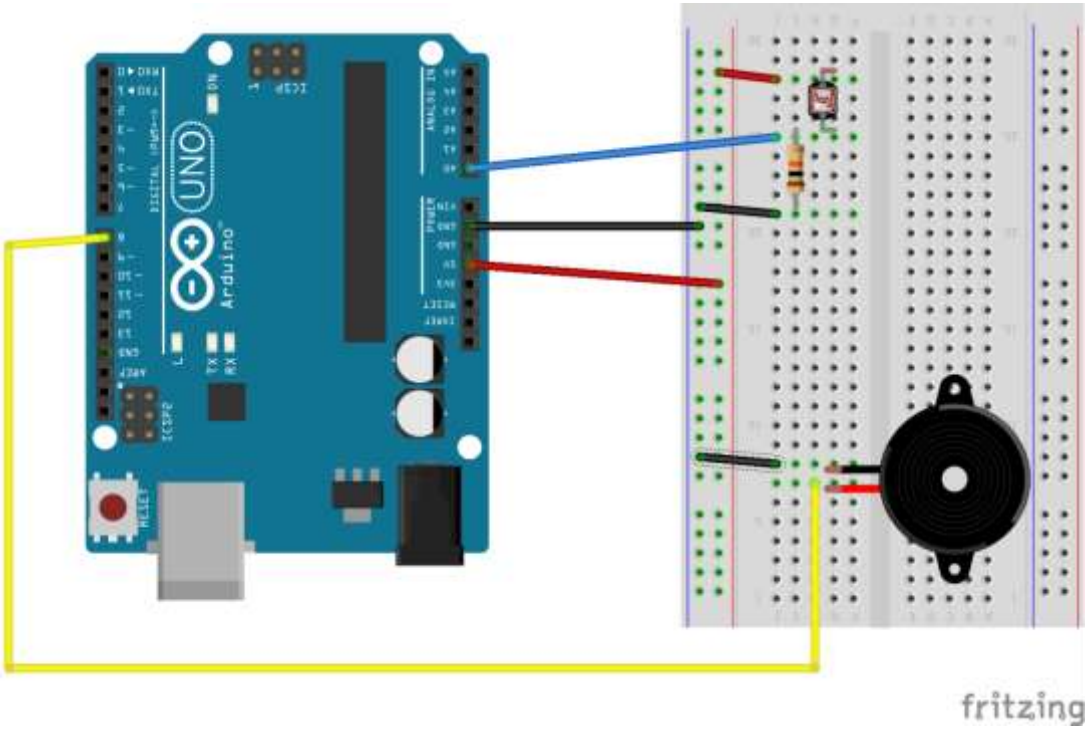
Κατάσταση	**Καλυμμένη με το δάκτυλο**	**Καλυμμένη με χαρτί**	**Σε φυσικό φως**	**Φωτισμένη με φακό**
Τιμή στην σειριακή οθόνη				

Τι παρατηρείτε; ……

……

……

4. Επεκτείνατε το παραπάνω κύκλωμα συνδέοντας τον πιεζοηλεκτρικό βομβητή όπως στο παρακάτω σχήμα



5. Φορτώστε τον παρακάτω κώδικα στο Arduino

```
int sensorValue;
int sensorLow = 1023;
int sensorHigh = 0;
const int ledPin = 13;
void setup() {
  pinMode(ledPin, OUTPUT);
  digitalWrite(ledPin, HIGH);
  while (millis() < 5000) {
    sensorValue = analogRead(A0);
    if (sensorValue > sensorHigh) {
      sensorHigh = sensorValue;
    }
    if (sensorValue < sensorLow) {
      sensorLow = sensorValue;
    }
  }
  digitalWrite(ledPin, LOW);
}
```

```
void loop() {
  sensorValue = analogRead(A0);
  int pitch = map(sensorValue, sensorLow, sensorHigh, 50, 4000);
  tone(8, pitch, 20);
  delay(10);
}
```

6. Ήρθε η ώρα να διασκεδάσετε!!!! Κινήστε τα χέρια σας πάνω από την φωτοαντίσταση και ακούστε τους ήχους που παράγονται.

7. Η λειτουργία της εντολής tone είναι να: ....
8. Η σύνταξη της εντολής tone είναι: .....
9. Τι θα πρέπει να αλλάξουμε στη σύνταξη της εντολης αυτής στο παραπάνω πρόγραμμα αν ο βομβητής είναι συνδεδεμένος στην ψηφιακή έδοδο D9 ; .....
10. Με ποια συχνότητα λειτουργεί ο πιεζοηλεκτρικός βομβητής στο παραπάνω πρόγραμμα; .....
11. Ποια είναι η σημασία της τρίτης παραμέτρου (20) στην εντολή tone στο παραπάνω παράδειγμα; (`tone(8, pitch, 20);`

ΒΙΒΛΙΟΓΡΑΦΙΑ –ΔΙΚΤΥΟΓΡΑΦΙΑ

Μπρακατσούλας Ε., Παπαϊωάνου Γ., Παπαδάκης Ι.(2003) *Γενικά Ηλεκτρονικά, Μέρος Α' Θεωρία*, ΤΕΕ, ΟΕΔΒ, Αθήνα

Γκιόκας Α., Φριλίγκος Στ., (2002), *Τεχνολογία Διατάξεων Ηλεκτρονικής*, ΤΕΕ, ΟΕΔΒ, Αθήνα

https://el.wikipedia.org/wiki/%CE%98%CE%AD%CF%81%CE%B5%CE%BC%CE%B9%CE%BD

http://playground.arduino.cc/Learning/PhotoResistor

http://arduino.cc/starterKit

https://www.arduino.cc/

Σχολική Μονάδα: 1ο ΕΠΑΛ Λειβαδιάς

Συμμετέχοντες Καθηγητές

ΚΑΛΟΒΡΕΚΤΗΣ ΚΩΝΣΤΑΝΤΙΝΟΣ
ΜΕΛΑΣ ΚΩΝΣΤΑΝΤΙΝΟΣ

Σχέδιο μαθήματος
Έλεγχος σερβομηχανισμού

Κωνσταντίνος Καλοβρέκτης
ΠΕ 17.08 – Ηλεκτρονικός, 1ο ΕΠΑΛ Λιβαδειάς,
Δελφών 24-26, Λιβαδειά, ΤΚ 32100

Μάθημα: Ηλεκτρονικές Διατάξεις της Β΄ Τάξης ΕΠΑΛ

Αντικείμενο Εργαστηριακής Άσκησης: Έλεγχος σερβομηχανισμού

Βαθμίδα: Β΄ ΕΠΑΛ της ειδικότητας: Τεχνικός Ηλεκτρονικών και Υπολογιστικών Συστημάτων, Εγκαταστάσεων

Διάρκεια: 2 Διδακτικές ώρες.

Γνωστικό υπόβαθρο μαθητών: Με βάση το αναλυτικό πρόγραμμα σπουδών του μαθήματος Ηλεκτρονικές Διατάξεις της Β΄ Τάξης ΕΠΑΛ της ειδικότητας Τεχνικός Ηλεκτρονικών και Υπολογιστικών Συστημάτων, Εγκαταστάσεων οι μαθητές γνωρίζουν να υλοποιούν απλά κυκλώματα σε πλακέτα breadboard.

Γενικός διδακτικός στόχος του μαθήματος: Οι μαθητές να κατανοούν τον τρόπο λειτουργία του σερβομηχανισμού με προγραμματισμό Arduino.

Αντικειμενικοί στόχοι μαθήματος:

- Οι μαθητές να μπορούν να αναγνωρίζουν τη μορφή ενός σήματος PWM
- Οι μαθητές να είναι σε θέση να κατασκευάζουν ένα απλό κύκλωμα σε πλακέτα breadboard και να το διασυνδέουν με τις εισόδους και τις εξόδους του arduino.
- Οι μαθητές να είναι σε θέση να χρησιμοποιούν το περιβάλλον Arduino

IDE και να συντάσσουν ένα απλό πρόγραμμα.

Μέθοδος διδασκαλίας: Εισήγηση, Ερωτοαποκρίσεις, Ομάδες Εργασίας

Μέσα διδασκαλίας – Υλικά: Η/Υ με εγκατεστημένο λογισμικό Arduino IDE, Arduino UNO, καλωδίο USB, breadboard, καλώδια, σεβομηχανισμός

Βοήθηματα: Σημειώσεις μαθήματος

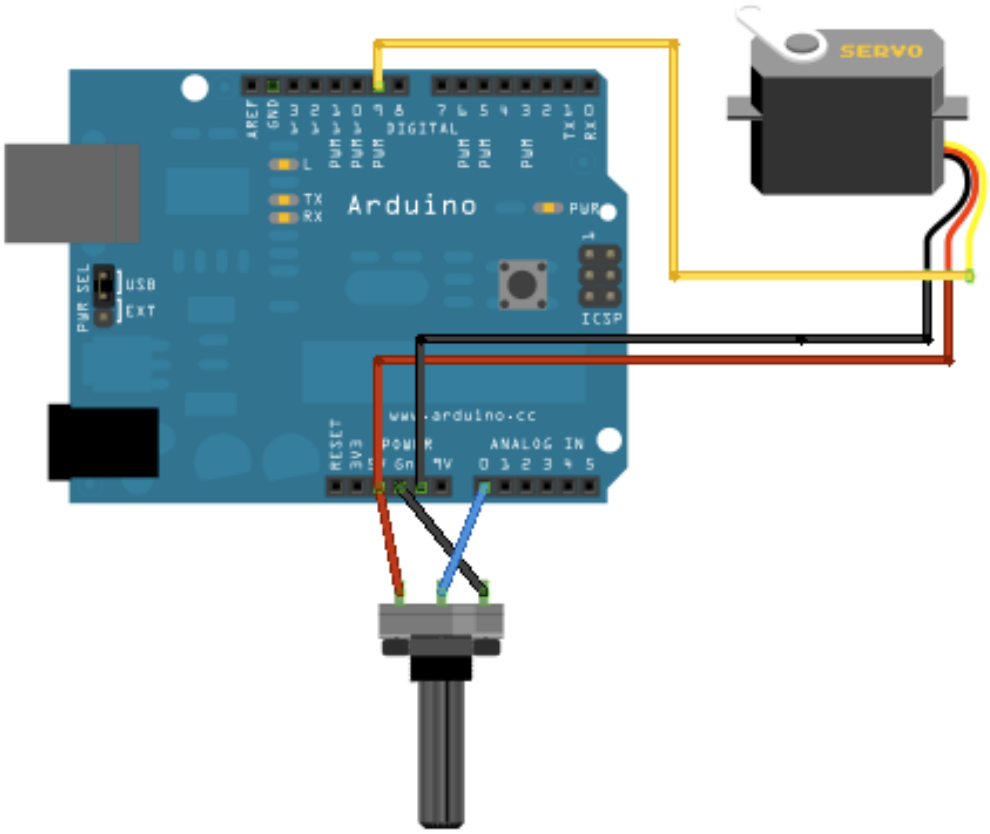
10. Χρονοπρογραμματισμός Εργαστηρίου

1η διδακτική ώρα

1. Περιγραφή άσκησης (5 λεπτά)
2. Περιγραφή κυκλώματος (5 λεπτά)
3. Θεωρητική των εξόδων PWM . (10 λεπτά)
4. Επεξήγηση ελέγχου του σερβομηχανισμού. (15 λεπτά)

2η διδακτική ώρα

1. Πραγματοποίηση της παρακάτω συνδεσμολογίας. (10 λεπτά)



Πρόγραμμα

```
#include <Servo.h>
Servo myservo;
int potpin = 0;
  int val;
void setup() {
  myservo.attach(9);
}
void loop() {
  val = analogRead(potpin);
 val = map(val, 0, 1023, 0, 180);
 myservo.write(val);
  delay(15)
}
```

11. Πλαίσιο αξιολόγησης των μαθητών

Προφορικές ερωτήσεις:

1. Που συνδέεται ο σερβομηχανισμός;
2. Ποιος είναι ο ρόλος του ποτενσιόμετρου;
3. Πως μεταβάλλουμε τη γωνία στον σερβομηχανισμό;
4. Τι αναφέρουμε στο τμήμα void setup;

12. Εργασία για το σπίτι

Τροποποιήστε το παραπάνω πρόγραμμα, έτσι ώστε να χρησιμοποιεί δύο σερβομηχανισμούς ελεγχόμενους από το ίδιο ποτενσιόμετρο με καθυστέρηση χρόνου

Σχολική Μονάδα: 1ο ΕΠΑΛ Λουτρακίου

Συμμετέχοντες Καθηγητές

ΚΥΡΟΠΟΥΛΟΣ ΣΤΑΥΡΟΣ
ΘΩΜΑΣ ΘΕΟΔΩΡΟΣ
ΓΡΑΜΜΑΤΙΚΟΣ ΒΑΣΙΛΕΙΟΣ
ΤΣΑΚΟΣ ΑΝΤΩΝΙΟΣ

Σχολείο : 1^{ο} ΕΠΑΛ ΛΟΥΤΡΑΚΙΟΥ

Τάξη : Β’ **Τομέας** : Ηλεκτρονικής

Καθηγητές : Κυρόπουλος Σταύρος, Θωμάς Θεόδωρος, Γραμματικός Βασίλειος, Τσάκος Αντώνιος.

Μάθημα : Εργαστήριο Ηλεκτρονικών Διατάξεων.

Άσκηση : ‘’ Απεικονίζοντας την μέτρηση της θερμοκρασίας με χρώματα, Arduino uno.’’

Στόχοι:

1) Να γνωρίζουν οι μαθητές την συνδεσμολογία του αισθητήρα θερμοκρασίας **LM35** στις αναλογικές εισόδους του μικροελεγκτή **ARDUINO.**
2) Να μπορούν να συνδέουν ένα **RGB Led** στις **PWM** εξόδους του μικροελεγκτή**.**
3) Να μπορούν να διαβάζουν τα δεδομένα εισόδου από τον αισθητήρα και με το κατάλληλο λογισμικό να τα μετατρέπουν σε αντίστοιχη έγχρωμη απεικόνιση.
4) Να συνδέουν στη θύρα USB του Η/Υ το ARDUINO και με το κατάλληλο IDE να προγραμματίζουν τον μικροελεγκτή.

Υλικά:

- Arduino uno
- Breadboard
- RGB LED
- 3x 330 Ω αντιστάσεις
- LM35
- Καλώδιο συνδέσεων.

Α. Θεωρητικό μέρος

Στην παρούσα άσκηση χρησιμοποιούμε τον αισθητήρα θερμοκρασίας LM35.

Η σειρά LM35 είναι μια σειρά ολοκληρωμένων κυκλωμάτων ακριβείας, με τάση εξόδου γραμμικά ανάλογη προς τη θερμοκρασία σε βαθμούς Κελσίου. Η συσκευή LM35 έχει ένα πλεονέκτημα έναντι άλλων γραμμικών αισθητήρων θερμοκρασίας, βαθμονομημένων σε Kelvin, καθώς ο χρήστης δεν απαιτείται να αφαιρέσει μια μεγάλη σταθερή τάση από την έξοδο, ώστε να ληφθεί Κελσίου κλιμάκωση.

Από το LM35 παίρνουμε 10mV ανά 1 ^{o}C . Έτσι για να έχουμε την μέτρηση σε ^{o}C, αν χρησιμοποιήσουμε τάση αναφοράς για τον ADC τα 1.1 volts και ο ADC έχει διακριτική ικανότητα 10 bits, τότε θα έχουμε :

2^10=1024 στάθμες με (1.1 / 1024)=1.0742mV διακριτική ικανότητα του ADC.

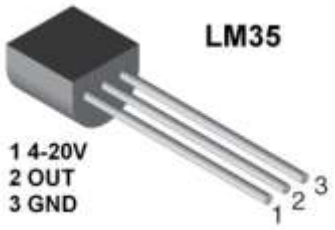
Από το

LM35 έχουμε 10mV ανά 1 βαθμό Κελσίου. Έτσι κάθε αλλαγή στην στάθμη του ADC (1.0742mV) αντιστοιχεί σε (1.0742 mV / 10 mV)=0.10742=(1 / 9.31) ^{o}C.

Η συσκευή LM35 δεν απαιτεί κανένα καλιμπράρισμα ή άλλη μικρορύθμιση, για να παρέχει τυπικά ακρίβεια της τάξης του ± ¼ ° C σε θερμοκρασία δωματίου και ± ¾ ° C κατά τη διάρκεια μιας πλήρους εύρους (-55 ° C έως 150 ° C) περιοχής θερμοκρασιών. Η χαμηλή αντίσταση εξόδου, γραμμική απόδοση και ακριβή εγγενή βαθμονόμηση της συσκευής LM35 κάνει τη διασύνδεση της με κύκλωμα ελέγχου ιδιαίτερα εύκολη.

Η συσκευή απαιτεί μονή τροφοδοσία και αντλεί μόλις 60 μΑ από αυτήν. Η συσκευή LM35 έχει βαθμολογηθεί να λειτουργούν σε ένα -55 ° C έως 150 ° C περιβάλλον

θερμοκρασιών, ενώ η έκδοση LM35C είναι σχεδιασμένη για -40 ° C και 110 ° C .



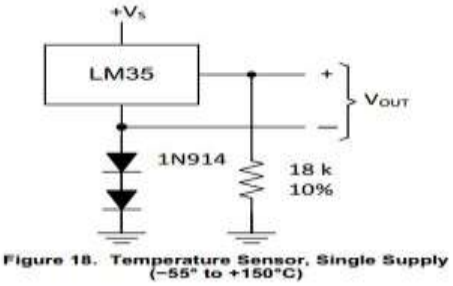


Figure 18. Temperature Sensor, Single Supply (−55° to +150°C)

Typical Applications

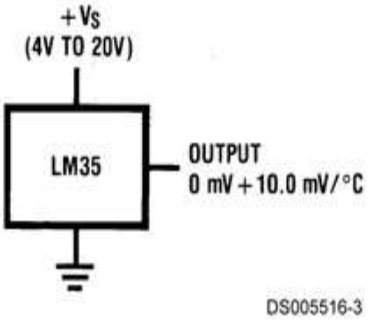
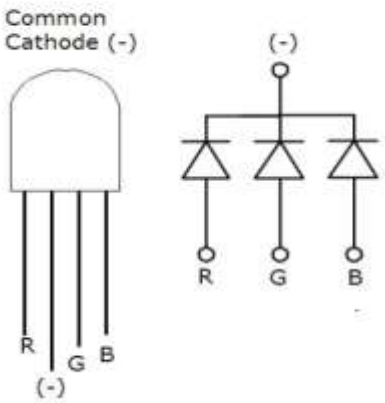
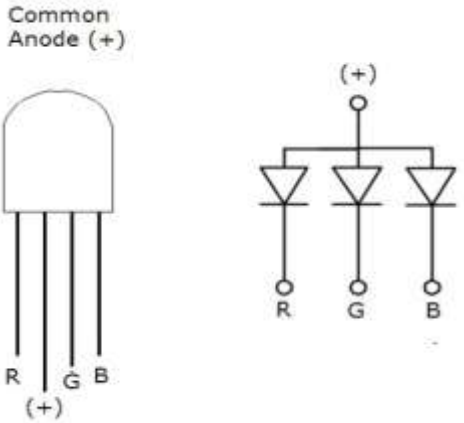


FIGURE 1. Basic Centigrade Temperature Sensor (+2˚C to +150˚C)

Στα ποιο πάνω σχήματα βλέπουμε την διάταξη των ακροδεκτών του LM35, καθώς και συνδέσεις αυτού για διαφορετικές περιοχές θερμοκρασιών.

Το RGB LED (Red, Green, Blue) διαθέτει 4 ακροδέκτες, έναν για κάθε χρώμα και έναν κοινής καθόδου ή κοινής ανόδου. Στο εσωτερικό του κρύβει τρία led, ένα για το κάθε χρώμα, τα οποία μπορούν να συνδυαστούν για να δώσουν ένα μεγάλο φάσμα χρωμάτων στον χρήστη. Ουσιαστικά πρόκειται για 3 σε 1 led! Το σχηματικό του διάγραμμα με την διάταξη των ακροδεκτών έχει ως εξής:





Όταν έχω RGB led κοινής καθόδου τότε ο κοινός ακροδέκτης συνδέεται στο GND (-) και στέλνουμε στα pin R,G,B τιμές (0 για off – 255 για full bright) ρυθμίζοντας την φωτεινότητα του κάθε led στα κατάλληλα επίπεδα για το κάθε χρώμα.

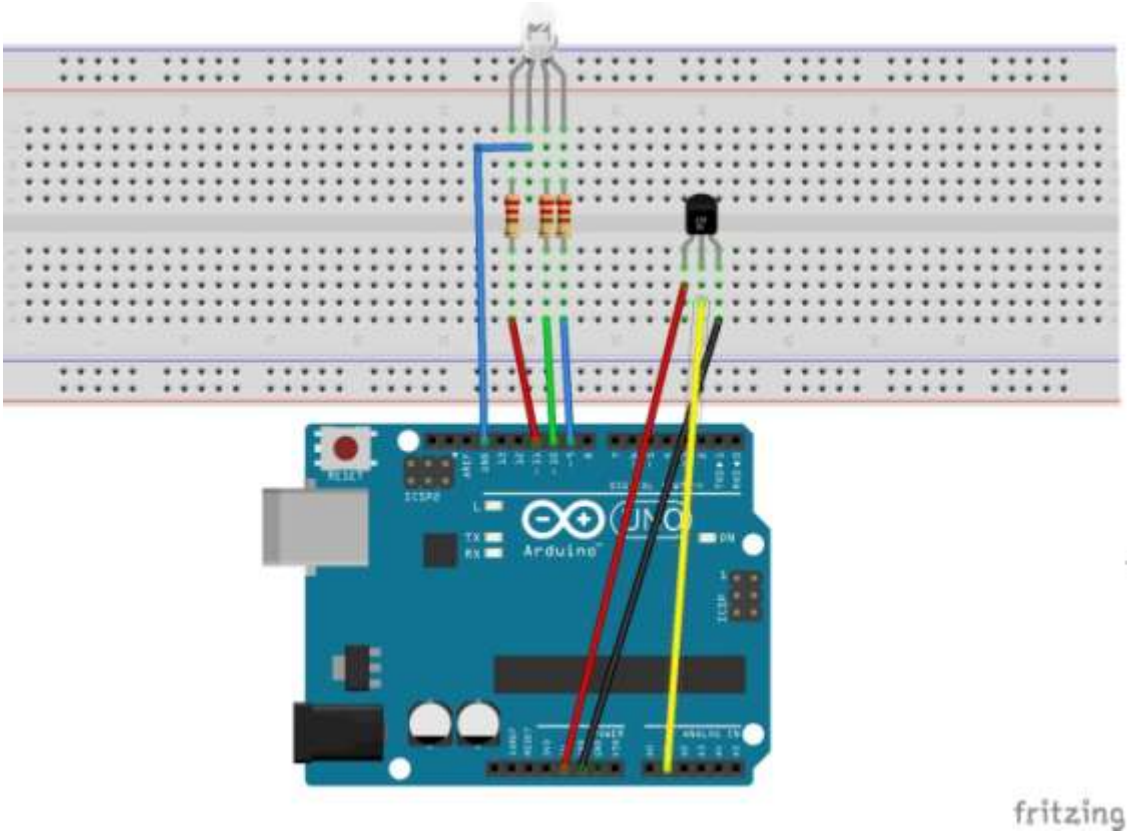
Εάν όμως έχω RGB led κοινής ανόδου τότε ο κοινός ακροδέκτης συνδέεται στο (+) και θα ισχύει το αντίθετο για τις τιμές που στέλνουμε στα pin R,G,B, ώστε να ρυθμίσουμε κατάλληλα την φωτεινότητα, δηλαδή (255 για off – 0 για full bright). Στην περίπτωση αυτή θα πρέπει να κάνουμε το εξής για το RED π.χ. : RED = 255 – RED.

Θα χρειαστούμε τον παρακάτω πίνακα, όπου είναι καταχωρημένες οι τιμές που θα στέλνουμε στα pin εξόδου για τον έλεγχο της φωτεινότητας των led, μέσω της PWM λειτουργίας των pin.

ΧΡΩΜΑ	**BRIGHTNESS**		
	RED	**GREEN**	**BLUE**
ΛΕΥΚΟ	225	225	225
ΜΩΒ	106	90	205
ΜΠΛΕ	0	0	225
ΚΥΑΝΟ	65	105	225
ΚΙΤΡΙΝΟ	225	105	0
ΠΟΡΤΟΚΑΛΙ	225	65	0
ΚΟΚΚΙΝΟ	225	0	0

Β. Πειραματικό μέρος:

1) Υλοποιούμε την συνδεσμολογία που φαίνεται στην παρακάτω εικόνα.



Στην συγκεκριμένη άσκηση το RGB led που χρησιμοποιούμε είναι κοινής καθόδου, οπότε ο κοινός ακροδέκτης του συνδέεται στο GND (-).

Σε περίπτωση που χρησιμοποιηθεί RGB led κοινής ανόδου τότε ο κοινός ακροδέκτης θα συνδεθεί στο (+) 5 Volt και για να μην μετατρέπουμε συνεχώς τον κώδικα προγραμματισμού ανάλογα με τον τύπο που χρησιμοποιούμε, θα συμπεριλάβουμε αρχικά το εξής όρισμα :

```
#define COMMON_ANODE
```

Το οποίο, για την πρώτη περίπτωση, το βάζουμε σε σχόλια για να γίνει ανενεργό.

Κατόπιν, πριν γράψουμε στα pin εξόδου τις τιμές για τα R,G,B, led, το ελέγχουμε με ένα:

```
#ifdef COMMON_ANODE
        red = 255 - red;
        green = 255 - green;
        blue = 255 - blue;
#endif
```

ώστε να πετύχουμε την αντίστοιχη μετατροπή των τιμών αυτών.

Τα pin εξόδου που θα χρησιμοποιήσουμε είναι τα : 9, 10, 11 των DIGITAL PWM OUTPUTS.

2) Στο προγραμματιστικό περιβάλλον για το ARDUINO γράφουμε τον παρακάτω κώδικα:

Πρόγραμμα

//----------**ΟΡΙΣΜΟΣ ΜΕΤΑΒΛΗΤΩΝ**----------

```
/*   #define COMMON_ANODE  /*  Χρησιμοποιείται αν εχω led κοινής ανόδου, διαφορετικά την γράφουμε ως σχολια. */

int redPin = 11;   // Red LED,   connected to digital pin 11

int greenPin = 10; // Green LED, connected to digital pin 10

int bluePin = 9;   // Blue LED,  connected to digital pin 9

float temp_c;      //  Αποθήκευση της τιμής της Θερμοκρασίας

int sensor_val;    //  Αποθήκευση της τιμής της Τάσης από τον αισθητήρα (0-1023)

int temp_pin = 1;  //  Κανάλι εισόδου του ADC.


void setup()

  {

  pinMode(redPin, OUTPUT);                    // sets the pins as output

  pinMode(greenPin, OUTPUT);

  pinMode(bluePin, OUTPUT);

  analogReference(INTERNAL);              /* Ορισμός της τάσης αναφοράς του ADC στα 1,1 Volts με χρήση εσωτερικής στάθμης αναφοράς του μικροελεγκτή ( μείωση από τα 5 στα 1,1 Volts  για καλύτερη διακριτική ακρίβεια – 4,9 mV σε 1 mV) */

  }

void loop()

  {
```

```
  sensor_val = analogRead(temp_pin);          /* Διάβασμα της τάσης που δίνει το LM35 από
τον ADC (κανάλι εισόδου 1 – pin A1) και απόδοση της τιμής (0 – 1023) στην μεταβλητή
sensor_val.  */

  temp_c = (sensor_val / 9.31);                    // Μετατροπή σε βαθμούς Κελσίου.


 // Επιλογή χρώματος ανάλογα με την τιμή της temp_c


 if (temp_c >= 40.00)

 { setColor(225, 0, 0);                        // red

 }

  else if (temp_c >= 32.00 && temp_c < 40.00)

 { setColor(225, 35, 0);                     // orange

 }

  else if (temp_c >= 25.00 && temp_c < 32.00)

 { setColor(225, 105, 0);                    // yellow

 }

  else if (temp_c >= 18.00 && temp_c < 25.00)

 { setColor(0, 225, 225);                    // cyan

 }

  else if (temp_c >= 11.00 && temp_c < 18.00)

 { setColor(0, 0, 225);                     // blue

 }

  else if (temp_c >= 4.00 && temp_c < 11.00)

 { setColor(225, 0, 100);                    // purple

 }

  else

 { setColor(225, 225, 225);                  // white

 }
```

```
delay(1000);


}


void setColor(int red, int green, int blue)
{
  #ifdef COMMON_ANODE
    red = 255 - red;
    green = 255 - green;
    blue = 255 - blue;
  #endif
analogWrite(redPin, red);
analogWrite(greenPin, green);
analogWrite(bluePin, blue);
}
```

3) Γράφοντας στο μέρος της αρχικοποίησης (setup()) τον κώδικα

```
Serial.begin(9600);
```

δηλώνουμε την σειριακή επικοινωνία και εισάγοντας στο μέρος του (loop()) σε κατάλληλα σημεία τις εντολές:

```
Serial.println(μεταβλητή);
```

Μπορούμε να βλέπουμε στην σειριακή οθόνη τις τιμές των διάφορων μεταβλητών και ειδικά της **temp_c** ώστε να ελέγχουμε την απόκριση του αισθητήρα (LM35) στις θερμοκρασιακές μεταβολές και να προσαρμόζουμε κατάλληλα τις περιοχές θερμοκρασίας για τον αντίστοιχο χρωματισμό του RGB Led.

Εδώ εμείς έχουμε χρησιμοποιήσει τις εξής:

α) temp_c >= 40.00 ------→ RED

β) 32.00 <= temp_c < 40.00 ------→ ORANGE

γ) 25.00 <= temp_c < 32.00 ------→ YELLOW

δ) 18.00 <= temp_c < 25.00 ------→ CYAN

ε) 11.00 <= temp_c < 18.00 ------→ BLUE

στ) 4.00 <= temp_c < 11.00 ------→ PURPLE

ζ) temp_c < 4.00 ------→ WHITE

Τις διάφορες θερμοκρασίες μπορούμε να τις πετύχουμε χρησιμοποιώντας πότε ένα ψυκτικό σπρέι και πότε σεσουάρ μαλλιών για θερμό αέρα.

Γ. Βιβλιογραφία:

https://www.arduino.cc/

http://www.ti.com/lit/ds/symlink/lm35.pdf

https://solarbotics.com/product/led-rgb-8cd5kca/

http://fritzing.org/home/

http://users.sch.gr/manpoul/docs/arduino/ProgrammingArduino.pdf

https://el.wikipedia.org/wiki/Arduino

Σχολική Μονάδα: 1ο ΕΠΑΛ Μεσολογγίου

Συμμετέχοντες Καθηγητές

ΠΑΠΑΖΑΧΑΡΙΑΣ ΑΘΑΝΑΣΙΟΣ
ΚΡΗΤΑΣ ΟΘΩΝ

Εκπαιδευτικό Σενάριο

Διασύνδεση οθονών χαρακτήρων LCD με την πλατφόρμα Arduino

Αντικείμενο Σεναρίου

Μέσω του εκπαιδευτικού σεναρίου, οι μαθητές θα εξοικειωθούν με τη διασύνδεση οθονών χαρακτήρων LCD με μικροελεγκτές (εδώ το Arduino) με 2 διαφορετικούς τρόπους. Επίσης θα κάνουν χρήση των δυνατοτήτων που τους παρέχει ο ελεγκτής της οθόνης. Τέλος θα εξοικοιωθούν με τη έννοια και τη χρήση των βιβλιοθηκών λογισμικού στον προγραμματισμό.

Τάξη / Τομέας /Μάθημα

Β' ΕΠΑΛ, Τομέας Ηλεκτρονικής.

Μάθημα: Ηλεκτρονικές Διατάξεις (Εργαστήριο)

Άσκηση 17 του εργαστηρίου

Γνωστικό Υπόβαθρο μαθητών - απαιτούμενες γνώσεις

Οι μαθητές θα πρέπει να είναι σε θέση:

- να αναγνωρίζουν τους διάφορους ακροδέκτες του Arduino και να μπορούν να πραγματοποιούν συνδέσεις.
- να γνωρίζουν τη χρήση του breadboard και τον τρόπο τοποθέτησης ηλεκτρονικών εξαρτημάτων σε αυτό
- να χειρίζονται το προγραμματιστικό περιβάλλον Arduino IDE και να γνωρίζουν τη διαδικασία ελέγχου ενός προγράμματος και φόρτωσής του στο Arduino.
- να γνωρίζουν τη δομή ενός προγράμματος φτιαγμένου στο Arduino IDE και να αναγνωρίζουν τα βασικά στοιχεία του (συναρτήσεις, δηλώσεις κ.λ.π.)

Γενικός στόχος Σεναρίου

Να γνωρίσουν οι μαθητές τους τρόπους διασύνδεσης μιας οθόνης χαρακτήρων LCD (και κατ΄επέκταση και άλλων συσκευών) με το Arduino καθώς και να εμπλουτίσουν τις γνώσεις τους στον προγραμματισμό γενικά και στη χρήση βιβλιοθηκών στο περιβάλλον του Arduino IDE.

Ειδικότεροι στόχοι

Μετά από την ολοκλήρωση της διδασκαλίας, οι μαθητές θα μπορούν να:

- διασυνδέουν μια οθόνη χαρακτήρων LCD στο Arduino με παράλληλο τρόπο
- χρησιμοποιούν το σειριακό I2C module για την σειριακή διασύνδεση της οθόνης
- αναφέρουν τα πλεονεκτήματα και μειονεκτήματα της παράλληλης και της σειριακής διασύνδεσης συσκευών
- περιγράφουν τον δίαυλο I2C
- προσθέτουν βιβλιοθήκες στο Arduino IDE
- εισάγουν βιβλιοθήκες σε ένα πρόγραμμα
- εκτυπώνουν ειδικούς χαρακτήρες στη οθόνη LCD
- δημιουργούν τους δικούς τους χαρακτήρες για την οθόνη LCD

Διδακτικές Τεχνικές/Προσεγγίσεις

Εργαστηριακή άσκηση σε ομάδες των 2 ατόμων, Εισήγηση, Συζήτηση, Ερωταποκρίσεις.

Χρονοπρογραμματισμός Διδασκαλίας

Διάρκεια διδασκαλίας: 4 διδακτικές ώρες μοιρασμένες σε 2 δίωρα εργαστήρια (160-180 λεπτά)

Εποπτικά και Διδακτικά μέσα
Υπολογιστές, Arduino boards με τα παρελκόμενα (breadboard, οθόνη, I2C module, jumper wires, trimmer 10K), Διαδίκτυο, Βιντεοπροβολέας
Παρουσιάσεις, Εργαστηριακές Ασκήσεις, Ερωτηματολόγια.

Καθώς η το Σενάριο αυτό προβλέπει τη διδασκαλία σε 2 διαφορετικά δίωρα, οι διάφορες φάσεις θα παρουσιαστούν για το κάθε δίωρο ξεχωριτά

1ο δίωρο (διασύνδεση LCD οθόνης με παράλληλο τρόπο)

Φάση 1. Προσέλκυση της προσοχής και του ενδιαφέροντος των μαθητών - Διάρκεια 10 λεπτά
(Εισήγηση, Ερωταποκρίσεις, Καταιγισμός ιδεών).

Στη συγκεκριμένη φάση της διδασκαλίας μέσω μιας σύντομης εισήγησης του εκπαιδευτικού γνωστοποιούνται στους μαθητές το θέμα και οι στόχοι του μαθήματος. Μέσω ερωταποκρίσεων οι μαθητές καλούνται να αναγνωρίσουν συσκευές της καθημερινότητάς τους ή του εργαστηρίου όπου χρησιμοποιούνται οθόνες χαρακτήρων.
Μοιράζεται στους μαθητές η **εργαστηριακή άσκηση 17α**. (Ελλείψει βιβλίου οι μαθητές θα πρέπει να λάβουν όλοι από ένα αντίτυπο των εργαστηριακών ασκήσεων που ετοιμάστηκαν για το μάθημα αυτό)

Φάση 2. Εισαγωγικά σημεία της άσκησης - Διάρκεια 10 λεπτά
(Εισήγηση)

Ξεκινά το κυρίως κομμάτι της διδασκαλίας με τις αρχικές διαφάνειες της **παρουσίασης της Άσκησης 17α**. Θα πρέπει να χρησιμοποιηθούν οι διαφάνειες 1-8. Η φάση ολοκληρώνεται με την συγκέντρωση από τους μαθητές των υλικών που απαιτεί η άσκηση.

Φάση 3. Εκτέλεση της εργαστηριακής άσκησης (ανα δυάδες) - πρώτο μέρος (υλοποίηση κυκλώματος και φόρτωση έτοιμου παραδείγματος)- Διάρκεια 25 λεπτά
(Εισήγηση, Εργαστηριακή άσκηση)

Έχοντας πλέον στα χέρια τους οι μαθητές την οθόνη LCD επανερχόμαστε στη διαφάνεια 7 για να τονίσουμε την αρίθμηση των ακροδεκτών. Η εργαστηριακή άσκηση εκτελείται από τους μαθητές σύμφωνα με την εκφώνηση που έχουν στη διάθεσή τους. Ταυτόχρονα προβάλλονται οι διαφάνειες 9-11 της **παρουσίασης της Άσκησης 17α**. Αυτό βοηθά στο οι μαθητές να προχωρούν στην εκτέλεση της άσκηση σχεδόν παράλληλα. Αρχικά γίνεται ο έλεγχος του backlight. Στη συνέχεια οι μαθητές προχωρούν στην υλοποίηση του τελικού κυκλώματος, την φόρτωση του έτοιμου παραδείγματος. Σε αυτό το σημείο οι μαθητές θα πρέπει να κάνουν αλλαγές στον κώδικα για να λειτουργήσει η οθόνη και να συμπληρώσουν ένα πίνακα στο έντυπο της **εργαστηριακής άσκησης 17α**.

Φάση 4. Επεξήγηση του κώδικα που χρησιμοποιήθηκε – Εισήγηση σχετικά με τις βιβλιοθήκες στο Arduino IDE - Διάρκεια 15 λεπτά
(Εισήγηση, Ερωταποκρίσεις)

Ο διδάσκοντας επεξηγεί τον κώδικα γραμμή γραμμή (με χρήση των διαφανειών 12-15 της **παρουσίασης της Άσκησης 17α**) Θα πρέπει να επιμείνει σε οτιδήποτε νέο ενώ οι μαθητές θα πρέπει να επεξηγούν τμήματα του κώδικα τα οποία έχουν ξανασυναντήσει. Για παράδειγμα θα πρέπει να αναγνωρίσουν τις 2 βασικές συναρτήσεις ενός προγράμματος του Arduino IDE, καθώς και δηλώσεις που έχουν ξανασυναντήσει.

Φάση 5. Εκτέλεση της εργαστηριακής άσκησης (ανα δυάδες) - δεύτερο μέρος (εκτύπωση ειδικών χαρακτήρων)- Διάρκεια 15 λεπτά
(Εισήγηση, Εργαστηριακή άσκηση)

Οι μαθητές καλούνται να εκτυπώσουν στην οθόνη μηνύματα που περιέχουν ειδικούς χαρακτήρες. Όταν αυτό δεν γίνεται, ο διδάσκων τους εξηγεί τη μνήμη χαρακτήρων του HD44780 και τους δίνει οδηγίες για το πώς θα τυπώσουν τους επιθυμητούς χαρακτήρες. Χρησμοποιούνται οι υπόλοιπες διαφάνειες της **παρουσίασης της Άσκησης 17α**

Φάση 6 Αξιολόγηση Διάρκεια 10 λεπτά.:
(Συμπλήρωση ερωτηματολογίου)

Ζητείται από τους μαθητές να συμπληρώσουν το ερωτηματολόγιο **Άσκηση 17a - LCD Display (parallel) ερωτηματολόγιο**.

Εργασία για το σπίτι

Βρείτε τις συναρτήσεις που έχει η βιβλιοθήκη LiquidCrystal και καταγράψτε τις. Επιλέξτε πέντε από αυτές (εκτός από όσες είδαμε εδώ) και γράψτε εν συντομία τι κάνει η κάθε μια από αυτές. Όπως πάντα, αν δεν τα πάτε και τόσο καλά με τα αγγλικά, για την μετάφραση μπορεί να σας βοηθήσει κάποιος γνωστός σας ή η καθηγήτρια των αγγλικών σας.

2ο δίωρο (διασύνδεση LCD οθόνης με σειριακό τρόπο – δημιουργία χαρακτήρων)

Φάση 1. Σύνδεση με τα προηγούμενα - Διάρκεια 10 λεπτά
(Εισήγηση, Ερωταποκρίσεις).

Στη συγκεκριμένη φάση της διδασκαλίας ο εκπαιδευτικός προσπαθεί να δει ποια στοιχεία από το προηγούμενο μάθημα οι μαθητές έχουν κάνει κτήμα τους και ποια μπορεί να είναι ακόμα θολά και ίσως πρέπει να τα διαλευκάνει. Ελέγχει και κατα πόσο οι μαθητές έχουν κάνει την εργασία για το σπίτι της προηγούμενης εβδομάδας.
Στη συνέχεια γνωστοποιούνται στους μαθητές το θέμα και οι στόχοι του μαθήματος. Μέσω ερωταποκρίσεων οι μαθητές καλούνται να ανακαλύψουν που μπορεί να χρησιμοποιείται σειριακή μετάδοση δεδομένων (USB, ethernet κ.α.)
Μοιράζεται στους μαθητές η **εργαστηριακή άσκηση 17β**. (Ελλείψει βιβλίου οι μαθητές θα πρέπει να λάβουν όλοι από ένα αντίτυπο των εργαστηριακών ασκήσεων που ετοιμάστηκαν για το μάθημα αυτό).

Φάση 2. Εισαγωγικά σημεία της άσκησης - Διάρκεια 10 λεπτά
(Εισήγηση)

Ξεκινά το κυρίως κομμάτι της διδασκαλίας με τις αρχικές διαφάνειες της **παρουσίασης της Άσκησης 17β**. Θα πρέπει να χρησιμοποιηθούν οι διαφάνειες 1-9. Η φάση ολοκληρώνεται με την συγκέντρωση από τους μαθητές των υλικών που απαιτεί η άσκηση. Έμφαση θα πρέπει να δωθεί:
Α) στις διαφορές (πλεονεκτήματα – μειωνεκτήματα) της παράλληλης και της σειριακής μεταφοράς δεδομένων
Β) στην αρχιτεκτονική και τα σήματα του διάυλου I2C
Γ) στον τρόπο που οι διάφορες συσκευές πάνω σε ένα δίαυλο I2C ξεχωρίζουν μεταξύ τους
Δ) στο πού συνδέονται οι I2C συσκεύές στο Arduino
Ο χρονισμός του διαύλου I2C είναι ελαφρώς πιο προχωρημένο θέμα και ο διδάσκων θα πρέπει να αποφασίσει το βάθος στο οποίο θα προχωρήσει ανάλογα με το γνωστικό υπόβαθρο των μαθητών.

Φάση 3. Εκτέλεση της εργαστηριακής άσκησης (ανα δυάδες) - πρώτο μέρος (υλοποίηση κυκλώματος, εισαγωγή βιβλιοθηκών και συγγραφή ή φόρτωση κώδικα)- Διάρκεια 25 λεπτά
(Εισήγηση, Εργαστηριακή άσκηση)

Οι μαθητές καλούνται να συνδέσουν την οθόνη LCD με το LCD I2C module (το τελευταίο έχει κολλημένους πάνω του κατάλληλους ακροδέκτες ώστε να μπορεί η οθόνη να συνδέεται ή να αφαιρείται ανάλογα) και στη συνέχεια να υλοποιήσουν την σύνδεση με το Arduino.
Έμφαση πρεπει να δωθεί σε ποια pin του Arduino πρέπει να συνδεθούν τα SCL και SDA.
Κατόπιν, με τις οδηγίες του φυλλαδίου της **εργαστηριακής άσκησης 17β** και του διδάσκοντα κατεβάζουν την απαραίτητη βιβλιοθήκη για να λειτουργήσει το LCD I2C module και την εγκαθιστούν.
Τέλος γράφουν τον απαραίτητο κώδικα ή τον φορτώνουν από το αρχείο HelloWorld_i2c_17.ino που βρίσκουν στο τοπικό δίκτυο.
Η εργαστηριακή άσκηση εκτελείται από τους μαθητές σύμφωνα με την εκφώνηση που έχουν στη διάθεσή τους. Ταυτόχρονα προβάλλονται οι διαφάνειες 10 -14 της **παρουσίασης της Άσκησης 17β**. Αυτό βοηθά στο οι μαθητές να προχωρούν στην εκτέλεση της άσκηση σχεδόν παράλληλα.

Φάση 4. Επεξήγηση του κώδικα που χρησιμοποιήθηκε – Τρόπος δημιουργίας χαρακτήρων Διάρκεια 15 -30 λεπτά
(Εισήγηση, Ερωταποκρίσεις)

Ο διδάσκοντας επεξηγεί τον κώδικα γραμμή γραμμή (με χρήση της **παρουσίασης της Άσκησης 17β**, διαφάνειες 15-21) Θα πρέπει να επιμείνει σε οτιδήποτε νέο ενώ οι μαθητές θα πρέπει να επεξηγούν τμήματα του κώδικα τα οποία έχουν ξανασυναντήσει. Για παράδειγμα θα πρέπει να αναγνωρίσουν τις 2 βασικές συναρτήσεις ενός προγράμματος του Arduino IDE, καθώς και δηλώσεις που έχουν ξανασυναντήσει. Επίσης θα πρέπει να γνωρίζουν τι κάνουν όσες συναρτήσεις βρήκαν κα ανέλυσαν στη εργασία για το σπίτι της προηγούμενης εβδομάδας.
Επεξηγείται τέλος και ο τρόπος με τον οποίο ο μαθητής μπορεί να δημιουργήσει ένα δικό του χαρακτήρα με ένα παράδειγμα που υπάρχει στον κώδικα.

Φάση 5. Εκτέλεση της εργαστηριακής άσκησης (ανα δυάδες) - δεύτερο μέρος (δημιουργία και εκτύπωση χαρακτήρων)- Διάρκεια 15 λεπτά
(Εισήγηση, Εργαστηριακή άσκηση)

Οι μαθητές καλούνται να εκτυπώσουν στην οθόνη χαρακτήρες που έχουν δημιουργήσει οι ίδιοι. Στο τέλος της **εργαστηριακής άσκησης 17β** υπάρχουν πίνακες που μπορούν να χρησιμοποιηθούν για το σχεδιασμό.
Η φάση αυτή είναι προαιρετική και υλοποιείται μόνο εφόσον ο διδάσκων κρίνει ότι υπάρχει χρόνος (αν η φάση 4 ολοκληρώθηκε σε λιγότερο από 20 λεπτά)

Φάση 6 Αξιολόγηση - Διάρκεια 10 λεπτά
(Συμπλήρωση ερωτηματολογίου)

Ζητείται από τους μαθητές να συμπληρώσουν το ερωτηματολόγιο **Άσκηση 17β - LCD Display (I2C) ερωτηματολόγιο**.

Εργασία για το σπίτι

Ψάξτε στο διαδίκτυο και βρείτε πέντε (5) ακόμα συσκευές/αισθητήρες που μπορούμε να συνδέσουμε στο Arduino και οι οποίες/οι χρησιμοποιούν τον δίαυλο I2C. Για κάθε μια από αυτές να βρείτε και τη διεύθυνση που συνήθως χρησιμοποιεί στον δίαυλο I2C.

Και αν έχετε διάθεση……

Τροποποιήστε τον πιο πάνω κώδικα ώστε στην οθόνη να εμφανίζονται στο ίδιο σημείο με τη σειρά οι χαρακτήρες:

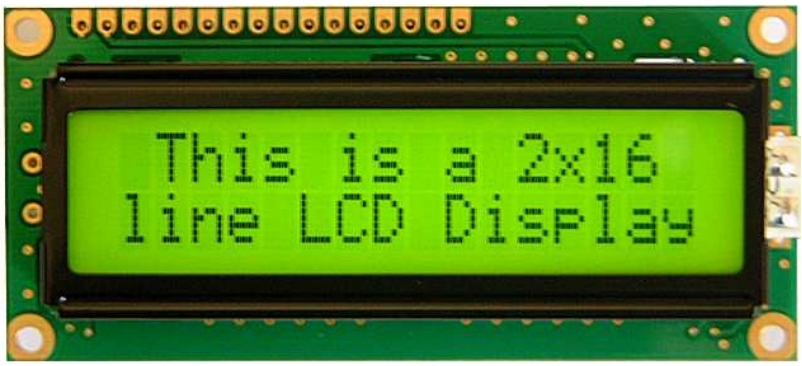
Μεταξύ δυο διαδοχικών χαρακτήρων να περνά 1/10 του δευτερολέπτου περίπου.

1^{ο} ΕΠΑ.Λ. Μεσολογγίου	**Σχολικό έτος : ____ – ____**
Τάξη : *Β΄$_{ΗΝ}$*	**Μάθημα : *Ηλεκτρονικές Διατάξεις***
Ημερομηνία: **…….**	**Μαθητές:** **……** **……** **……**

Άσκηση 17(α)
Θέμα : Απεικόνιση σε LCD display 16x2 (με παράλληλη σύνδεση)

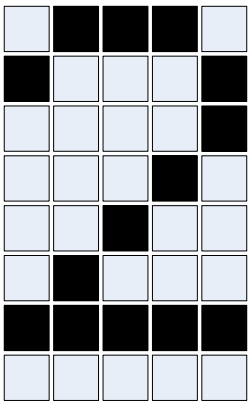
Σύντομη θεωρία

Όταν κανείς σχεδιάζει μια εφαρμογή είναι συνήθως χρήσιμο να εμφανίζει κάποια δεδομένα ή μηνύματα σε μια οθόνη (π.χ. την θερμοκρασία ή ένα string). Πολλά είδη οθονών είναι διαθέσιμα ανάλογα με το μέγεθος ή το χρώμα, όπως 8x2, 16x2, 16x4, 20x4 κ.α., άλλες με πράσινο background, άλλες με μπλε, άλλες με κόκκινο. Εμείς θα χρησιμοποιήσουμε μια οθόνη LCD 16x2, όπως στο σχήμα 17.1.



(α)
Σχήμα 17.1. 16x2 LCD display

Το όνομά της δηλώνει ότι πρόκειται για μια οθόνη υγρών κρυστάλλων (Liquid Crystal Display) όπου μπορούν να απεικονιστούν 2 γραμμές χαρακτήρων με 16 χαρακτήρες κάθε γραμμή. Κάθε χαρακτήρας σχηματίζεται από ένα πίνακα (matrix) 5x8 κουκίδων. Για παράδειγμα, για τον αριθμό 2 μαυρίζουν οι κουκίδες που φαίνονται στο επόμενο σχήμα.



Σχήμα 17.2. Απεικόνιση του αριθμού 2 σε matix 5x8 dots

Να σημειωθεί ότι οι 5 κουκίδες στη κάτω γραμμή χρησιμοποιούνται συνήθως για τον κέρσορα. Για να γράψει κανείς στην οθόνη ένα χαρακτήρα θα πρέπει να προσδιορίσει τη θέση στην οποία αυτός θα γραφεί δίνοντας τη γραμμή και την στήλη. Έτσι η πρώτη πάνω αριστερά θέση είναι η (0,0) ενώ η τελευταία κάτω αριστερά είναι η (1,15).

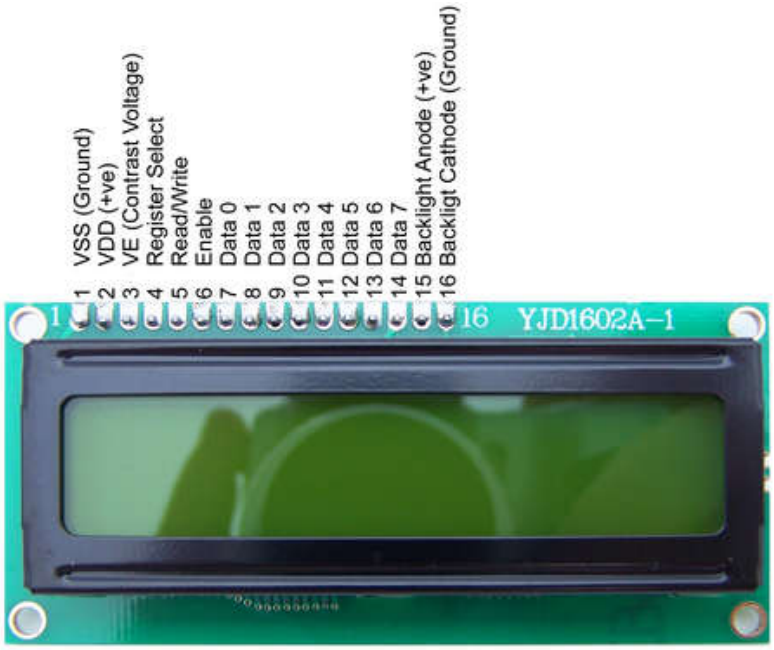
Η οθόνες του εργαστηρίου χρησιμοποιούν τον ελεγκτή HD44780 της Hitachi. Ο ελεγτής αυτός χρησιμοποιείται σε μονόχρωμες οθόνες χαρακτήρων εδώ και 30 χρόνια. Η ευρεία του αποδοχή και το γεγονός ότι παράγονται ελεγτές συμβατοί με τον HD44780 τον έχουν κάνει το de facto standard για τις οθόνες LCD.

Οι LCD οθόνες χαρακτήρων μπορούν να έχουν οπίσθιο φωτισμό (backlight) (π.χ. με LED) αν και κάτι τέτοιο δεν είναι απαραίτητο. Η διεπαφή που χρησιμοποιούν αποτελείται από 16 επαφές όπου μπορεί κανείς να προσαρτήσει, αν δεν υπάρχουν ήδη, ακροδέκτες των 2.54mm. Στις περιπτώσεις που δεν υπάρχει οπίσθιος φωτισμός απαιτούνται μόνο 14 ακροδέκτες.

Οι ακροδέκτες είναι οι ακόλουθοι:

pin	όνομα	Επεξήγηση	Περιγραφή
1	GND	Ground	
2	VCC	Τροφοδοσία	3.3V – 5V
3	VE ή VO	Contrast Voltage	
4	RS	Register Select	0 = command (εντολή) 1 = data (δεδομένα)
5	R/W	Read/Write	0 = write (εγγραφή) 1 = read (ανάγνωση) (*1)
6	EN	Enable	Αρνητικά ακμοπυροδήτητο ρολόι
[7 – 14]	[D0 – D7]	Data bits	Bits δεδομένων (*2)
15	Backlight Anode (+)		Άνοδος LED οπίσθιου φωτισμού
16	Backlight Cathode (-)		Κάθοδος LED οπίσθιου φωτισμού

***1** Συνήθως δεν διαβάζουμε από την οθόνη και γι'αύτό το λόγο το συγκεκριμένο pin συνδέεται στη γείωση*
***2** Σε 4-bit λειτουργία, δεν χρησιμοποιούνται τα [Bit 0 – Bit 3]*



Σχήμα 17.3. Οι ακροδέκτες μιας 16x2 LCD οθόνης

Οι LCD οθόνες του εργαστηρίου έχουν ήδη κολλημένους πάνω τους τους ακροδέκτες των 2.54mm (να ευχαριστείτε τους παλαιότερους μαθητές γι' αυτό). Είμαστε πλέον έτοιμοι να συνδέσουμε τις οθόνες στο arduino για να τις δοκιμάσουμε και κατόπιν να προβάλουμε εκεί τα μηνύματά μας.

Υλικά:

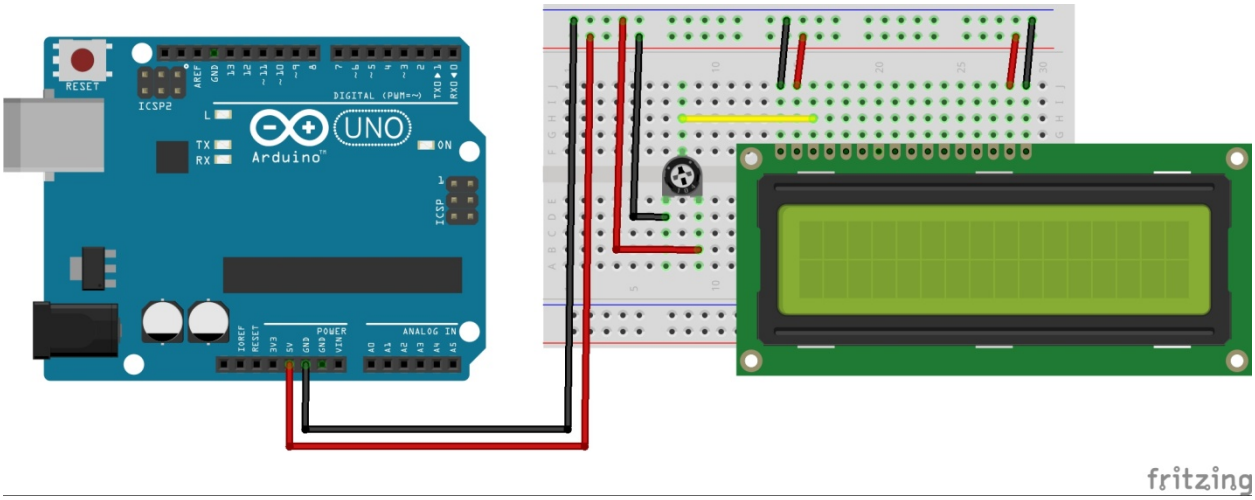
1 arduino uno board
1 breadboard
16 Jumper wires
(1 αντιστάτης 220Ω)
1 trimmer 10KΩ
1 16x2 LCD display

Λογισμικό – Βιβλιοθήκες :

Βιβλιοθήκη **LiquidCrystal** (ήδη ενσωματωμένη στο Arduino IDE)

Υλοποίηση:

Αρχικά θα δοκιμάσουμε αν η οθόνη μας ανάβει και τον τρόπο με τον οποίο θα ελέγχουμε αντίθεση. Το κύκλωμα που θα υλοποιήσουμε είναι αυτό του επόμενου σχήματος:

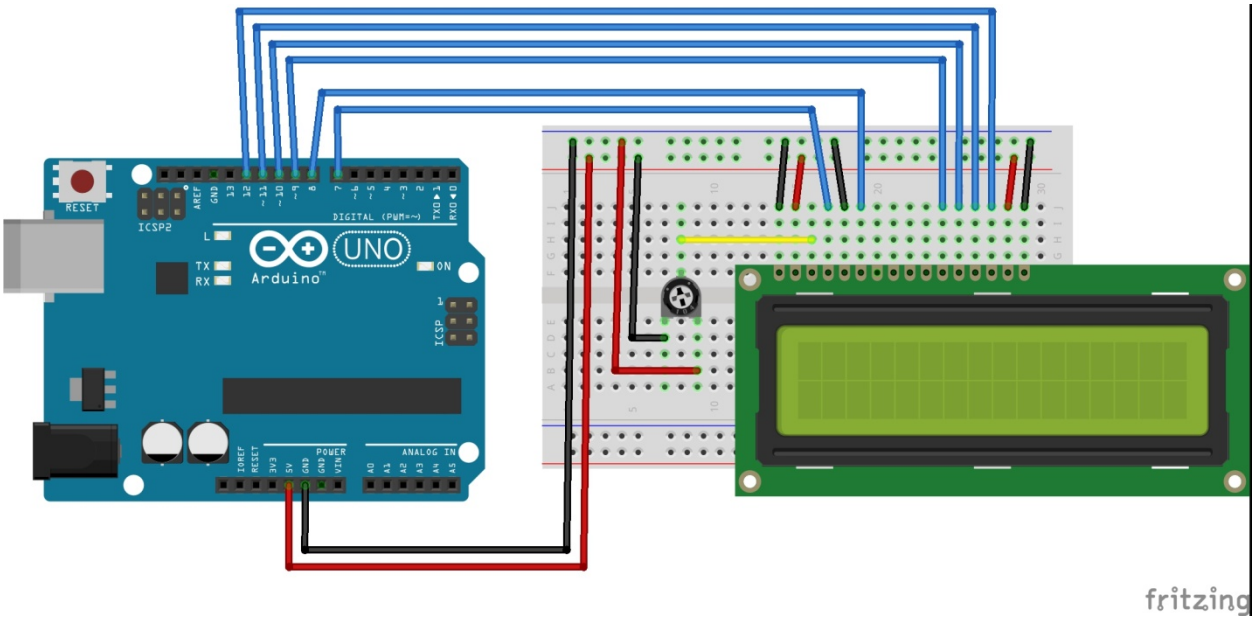


Σχήμα 17.4. Συνδεσμολογία ελέγχου του οπίσθιου φωτισμού

Στο κύκλωμα αυτό έχουμε συνδέσει την τροφοδοσία και τη γείωση της LCD οθόνης (pins 2 και 1) καθώς και την άνοδο και την κάθοδο του LED οπίσθιου φωτισμού (pins 15 και 16). Για να ελέγχουμε την αντίθεση (contrast) της οθόνης έχουμε συνδέσει και ένα trimmer στο pin 3. Ρυθμίζοντας την τάση στο pin 3 (με τη βοήθεια του trimmer) θα πρέπει να παρατηρούμε να γίνονται ποιο έντονα ή ποιο αχνά τα ορθογώνια τα οποία είναι ουσιαστικά πίνακες απο κουκίδες (5x8 κουκίδες σε κάθε θέση). Από τη στιγμή που το πετύχαμε αυτό θα προχωρήσουμε στις υπόλοιπες διασυνδέσεις.

Σημαντική παρατήρηση: οι περισσότερες οθόνες LCD με οπίσθιο φωτισμό έχουν ενσωματωμένη μια αντίσταση συνδεδεμένη στο pin 15. Σε περίπτωση που αυτό δεν ισχύει θα πρέπει να χρησιμοποιηθεί εξωτερική αντίσταση μεταξύ της τροφοδοσίας και του pin 15. Η τιμή της αντίστασης θα εξαρτάται από τα ηλεκτρικά χαρακτηριστικά της οθόνης και θα κυμαίνεται από 100Ω έως 220Ω περίπου.

Στη συνέχεια θα προσθέσουμε τα απαραίτητα καλώδια στο κύκλωμά μας ώστε να είναι δυνατή η αποστολή δεδομένων από το arduino στην οθόνη. Το κύκλωμα που θα υλοποιηθεί είναι το επόμενο.



Παρατηρήστε ότι από τα 8 pins για δεδομένα που έχει η οθόνη εμείς χρησιμοποιούμε μόνο τα 4. Αυτό σημαίνει ότι η διαδικασία να στείλουμε στην οθόνη ένα byte (8-bit), το οποίο θα αντιστοιχεί σε ένα χαρακτήρα, θα είναι λίγο περισσότερο πολύπλοκη απ' ότι αν χρησιμοποιούσαμε τα και τα 8 bit. Βέβαια αυτό δεν μας πειράζει καθόλου γιατί εμείς δεν θα μπούμε σε τέτοια λεπτομέρεια. Θα αφήσουμε άλλον να το κάνει αυτό. Και αυτός δεν είναι άλλος από την έτοιμη βιβλιοθήκη LiquidCrystal ή οποία είναι ενσωματωμένη στο Arduino IDE και τις συναρτήσεις που αυτή έχει.

Κώδικας:

Σε αυτή την άσκηση δεν θα χρειαστεί να κοπιάσουμε πολύ για την συγγραφή του κώδικα, θα πάρουμε έτοιμο κώδικα από τα παραδείγματα του Arduino IDE.

Από το περιβάλλον του Arduino IDE επιλέγουμε Files -> Examples ->LiquidCrystal - >HelloWorld και έχουμε τον ακόλουθο κώδικα:

```
/*
  LiquidCrystal Library - Hello World

 Demonstrates the use a 16x2 LCD display.  The LiquidCrystal
 library works with all LCD displays that are compatible with the
 Hitachi HD44780 driver. There are many of them out there, and you
 can usually tell them by the 16-pin interface.

 This sketch prints "Hello World!" to the LCD
 and shows the time.

  The circuit:
 * LCD RS pin to digital pin 12
 * LCD Enable pin to digital pin 11
 * LCD D4 pin to digital pin 5
 * LCD D5 pin to digital pin 4
 * LCD D6 pin to digital pin 3
 * LCD D7 pin to digital pin 2
 * LCD R/W pin to ground
 * LCD VSS pin to ground
 * LCD VCC pin to 5V
 * 10K resistor:
 * ends to +5V and ground
 * wiper to LCD VO pin (pin 3)

 Library originally added 18 Apr 2008 by David A. Mellis
 library modified 5 Jul 2009 by Limor Fried (http://www.ladyada.net)
 example added 9 Jul 2009 by Tom Igoe
 modified 22 Nov 2010 by Tom Igoe

 This example code is in the public domain.

 http://www.arduino.cc/en/Tutorial/LiquidCrystal
 */

// include the library code:
#include <LiquidCrystal.h>

// initialize the library with the numbers of the interface pins
LiquidCrystal lcd(12, 11, 5, 4, 3, 2);

void setup() {
  // set up the LCD's number of columns and rows:
  lcd.begin(16, 2);
  // Print a message to the LCD.
  lcd.print("hello, world!");
}

void loop() {
  // set the cursor to column 0, line 1
  // (note: line 1 is the second row, since counting begins with 0):
```

```
  lcd.setCursor(0, 1);
  // print the number of seconds since reset:
  lcd.print(millis() / 1000);
}
```

Αν δοκιμάσετε να τρέξετε τον παραπάνω κώδικα (είμαι σίγουρος ότι κάποιοι από εσάς ήδη το κάνατε ☺) δεν θα δείτε κανένα μήνυμα στην οθόνη. Ο λόγος; Απλός. Το πρόγραμμα αυτό θεωρεί διαφορετική διασύνδεση μεταξύ LCD οθόνης και Arduino από αυτή που έχουμε υλοποιήσει. Αν προσέξετε στα σχόλια στην αρχή του scetch, περιγράφεται η διασύνδεση αυτή. Στον επόμενο πίνακα φαίνονται οι ακροδέκτες του Arduino που χρησιμοποιεί το παράδειγμα. **Συμπληρώστε** τους ακροδέκτες που έχετε συνδέσει στην υλοποίησή μας:

pin	Έτοιμο παράδειγμα	Δική μας υλοποίηση
RS	12	
EN	11	
D4	5	
D5	4	
D6	3	
D7	2	

Και τώρα που είδαμε τη διαφορά θα πρέπει να τροποποιήσουμε τον κώδικα ώστε να αναταποκρίνεται στη δική μας διασύνδεση. Η δήλωση των παραπάνω ακροδεκτών γίνεται με την εντολή:

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

η οποία πρέπει να γίνει **(συμπληρώστε)**:

```
LiquidCrystal lcd(__,__,__,___,___,__);
```

Θα εξηγήσουμε πιο κάτω τι συμαίνει αυτή η εντολή, όπως και ο υπόλοιπος κώδικας.

Αν τρέξετε τώρα το πρόγραμμα θα πρέπει στην οθόνη να εμφανιστεί το μήνυμα Hello World! στην πρώτη γραμμή της οθόνης και τα δευτερόπλεπτα από το τελευταίο reset στη δεύτερη γραμμή.

Όλος ο κώδικας αποτελείται ουσιαστικά από 8 γραμμές. Ας δούμε λίγο αναλυτικότερα τι κάνει κάθε εντολή. Ξεκινάμε με τη δήλωση

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

η οποία δηλώνει στον compiler ότι θα πρέπει να συμπεριλάβει και την έτοιμη βιβλιοθήκη LiquidCrystal. Η βιβλιοθήκη αυτή περιλαμβάνει έτοιμες μεθόδους (συναρτήσεις) που βοηθούν στο χειρισμό μιας LCD οθόνης. Θα μπορούσαμε να κάνουμε τη δουλειά μας και χωρίς αυτή αλλά θα ήταν πολύ πιο κοπιώδες και ξεφεύγει από τους στόχους του μαθήματος.

Ακολουθεί η εντολή:

```
LiquidCrystal lcd(7, 8, 9, 10, 11, 12);
```

Η LiquidCrystal lcd() δημιουργεί ένα αντικείμενο τύπου LiquidCrystal, με όνομα lcd. Οι παράμετροι που δίνονται μέσα στην παρένθεση θα πρέπει να είναι 6 ή 7 ή 10 ή 11, σύμφωνα με τα παρακάτω πρότυπα:

LiquidCrystal(rs, enable, d4, d5, d6, d7)
LiquidCrystal(rs, rw, enable, d4, d5, d6, d7)
LiquidCrystal(rs, enable, d0, d1, d2, d3, d4, d5, d6, d7)
LiquidCrystal(rs, rw, enable, d0, d1, d2, d3, d4, d5, d6, d7)

Εμείς χρησιμοποιούμε την πρώτη μορφή (ή όπως αλλιώς μπορούμε να πούμε, τον πρώτο constructor). Έτσι η πρώτη παράμετρος είναι το pin 7 του Arduino που είναι συνδεδεμένο με το pin RS της οθόνης, η δεύτερη είναι το pin 8 του Arduino που είναι συνδεδεμένο με το pin EN της οθόνης κ.ο.κ.

Ακολουθεί η συνάρτηση setup όπου με τη δήλωση :

```
lcd.begin(16, 2);
```

γίνεται αρχικοποίηση του αντικείμένου lcd, όπου δηλώνεται το μέγεθος της οθόνης, ενώ με την δήλωση

```
lcd.print("hello, world!");
```

τυπώνεται στην οθόνη και στη θέση που είναι ο κέρσορας το μήνυμα hello, world!. Μετά την αρχικοποίηση ο κέρσορας βρίσκεται στη θέση (0,0) δηλαδή στον πρώτο χαρακτήρα της πρώτης γραμμής. Να σημειώσουμε ότι με την συνάρτηση `print` μπορούμε να τυπώσουμε ένα χαρακτήρα ή ένα string ή την τιμή μιας μεταβλητής (ή μιας έκφρασης) μόνο κάθε φορά. Με μια εντολή print δεν μπορούμε να τυπώσουμε ενα string και μια μεταβλητή μαζί. Η `println` θα τύπωνε το μήνυμά μας και μετά θα άλλαζε γραμμή.

Συνεχίζουμε με τη συνάρτηση loop όπου με τη δήλωση :

```
lcd.setCursor(0, 1);
```

ο κέρσορας τοποθετείται στην πρώτη θέση (στήλη 0) της δεύτερης γραμμής (γραμμή 1) της οθόνης.

Τέλος η

```
lcd.print(millis() / 1000);
```

τυπώνει στην οθόνη τα δευερόλεπτα που πέρασαν από το τελευταίο reset.

<u>Και κάτι επιπλέον ....</u>

Ας δοκιμάσουμε τώρα να τυπώσουμε και άλλα μηνύματα. Τι λέτε για το ¨π=3.14159¨ ή το ¨R=10ΚΩ¨ ή το ¨SORRY!. YOU LOSE ☹¨. Δοκιμάστε οποιοδήποτε από τα δυο πρώτα στη θέση του `lcd.print("hello, world!");` και δείτε το αποτέλεσμα. Τι παρατηρείτε;

Οι χαρακτήρες π και Ω δεν τυπώνονται σωστά στην οθόνη. Γιατί άραγε; Μπορούμε να κάνουμε κάτι γι'αυτό;

Λοιπόν, οι χαρακτήρες που μπορούν να τυπωθούν στην οθόνη είναι αποθηκευμένοι σε μια μνήμη ROM του ελεγκτή HD44780. Αν διαβάσουμε το data sheet του ελεγκτή θα βρούμε ποιοι είναι αυτοί οι χαρακτήρες. Για ευκολία ο πίνακας αυτός παρατίθεται στη επόμενη σελίδα.

Από τον πίνακα βλέπουμε ότι ο χαρακτήρας ¨π¨ βρίσκεται στην στήλη 1111 (περισσότερο σημαντικά bit) και στη γραμμή 0111 (λιγότερο σημαντικά bit). Έτσι αντιστοιχεί στην λέξη 11110111_2 ή 247_{10}. Για να τυπώσουμε τον χαρακτήρα αυτόν θα πρέπει να χρησιμοποιήσουμε την εντολή:

```
lcd.print(char(247));
```

και οπότε ολόκληρη η έκφραση θα ήταν:

```
lcd.print(char(247));
lcd.print("=3.14159");
```

Το ίδιο συμβαίνει και με το σύμβολο ¨Ω¨ που αντιστοιχεί στη λέξη 11110100_2 ή 244_{10}. Τι θα γίνει όμως με τη λυπημένη φατσούλα; Δεν υπάρχει τέτοιο σύμβολο. Μπορεί όμως κάτι να γίνει. Θα το δούμε την επόμενη φορά.

Εργασία για το σπίτι

Βρείτε τις συναρτήσεις που έχει η βιβλιοθήκη LiquidCrystal και καταγράψτε τις. Επιλέξτε πέντε από αυτές (εκτός από όσες είδαμε εδώ) και γράψτε εν συντομία τι κάνει η κάθε μια από αυτές. Όπως πάντα, αν δεν τα πάτε και τόσο καλά με τα αγγλικά, για την μετάφραση μπορεί να σας βοηθήσει κάποιος γνωστός σας ή η καθηγήτρια των αγγλικών σας.

Πίνακας 17.2 Το σετ των χαρακτήρων του ελεγκτή HD44780
(πηγή: http://mil.ufl.edu/4744/docs/lcdmanual/characterset.html)

1^{ο} ΕΠΑ.Λ. Μεσολογγίου	**Σχολικό έτος : ____ – ____**
Τάξη : *Β΄$_{ΗΝ}$*	**Μάθημα : *Ηλεκτρονικές Διατάξεις***
Ημερομηνία: **…….**	**Μαθητής:** **……**

Άσκηση 17(α)
Θέμα : Απεικόνιση σε LCD display 16x2 (παράλληλη σύνδεση)

Φύλλο Εργασίας

Ποιες από τις παρακάτω προτάσεις θεωρείτε ότι είναι σωστές;

(μπορεί κάθε ερώτηση να έχει περισσότερες από μια σωστές απαντήσεις)

Α. Οι οθόνες χαρακτήρων LCD:

1. Παράγονται μόνο σε διαστάσεις 2x16 και 4x20
2. Έρχονται με διάφορα χρώματα οπίσθιου φωτισμού
3. Μπορούν να χρησιμοποιηθούν και για την προβολή γραφικών

Β. Ο ελεγκτής HD44780

1. Έχει ευρεία αποδοχή
2. Είναι σχετικά καινούργιος ελεγκτής
3. Χρησιμοποιείται τόσο σε οθόνες χαρακτήρων όσο και σε οθόνες γραφικών
4. Αποτελεί το de facto standard

Γ. Μια οθόνη χαρακτήρων που χρησιμοποιεί τον HD44780

1. Έχει 16 ακροδέκτες
2. Η φωτεινότητά της δεν μπορεί να αλλάξει
3. Τα δεδομένα που δέχεται είναι μόνο 8-bit
4. Δεν μπορεί να είναι μεγαλύτερη από 4x20

Δ. Οι οθόνες LCD έρχονται πάντα με μια αντίσταση συνδεδεμένη στο pin 15

1. Σωστό
2. Λάθος

Ε. Μια βιβλιοθήκη

1. περιλαμβάνει δηλώσεις συναρτήσεων αλλά και αντικειμένων ή τύπων δεδομένων
2. απλοποιεί τη συγγραφή του κώδικα
3. πρέπει να γραφτεί από τον χρήστη
4. Απαιτούνται οι βιβλιοθήκες Wire και η LiquidCrystal

ΣΤ. Για πετύχουμε εύκολα την επικοινωνία του Arduino με την LCD οθόνη

1. Δεν απαιτούνται βιβλιοθήκες
2. Απαιτείται μόνο η βιβλιοθήκη Wire
3. Απαιτείται μόνο η βιβλιοθήκη LiquidCrystal
4. Απαιτούνται οι βιβλιοθήκες Wire και η LiquidCrystal

Ζ. Η δήλωση lcd.begin(16, 2); σημαίνει ότι η οθόνη που θα χρησιμοποιήσουμε έχει:

1. 16 ακροδέκτες εκ των οποίων η τροφοδοσία είναι ο 2ος
2. 2 γραμμές και 16 στήλες

Η. Τι εντολές θα γράφατε για να πετύχετε την εκτύπωση σε μια οθόνη LCD χαρακτήρων 2x16 το παρακάτω μήνυμα:

Temperature = x °C

Όπου x η τιμή μιας μεταβλητής (με όνομα x). Το σύμβολο ° βρίσκεται στη στήλη 1101 και στη γραμμή xxxx1111 του πίνακα χαρακτήρων.

Ηλεκτρονικές Διατάξεις

Άσκηση 17

LCD Display (παράλληλη σύνδεση)

1ο ΕΠΑ.Λ. Μεσολογγίου
Τομέας Ηλεκτρονικής
Σχ. Έτος : 2015-2016
Διδάσκων: Παπαζαχαρίας Αθανάσιος

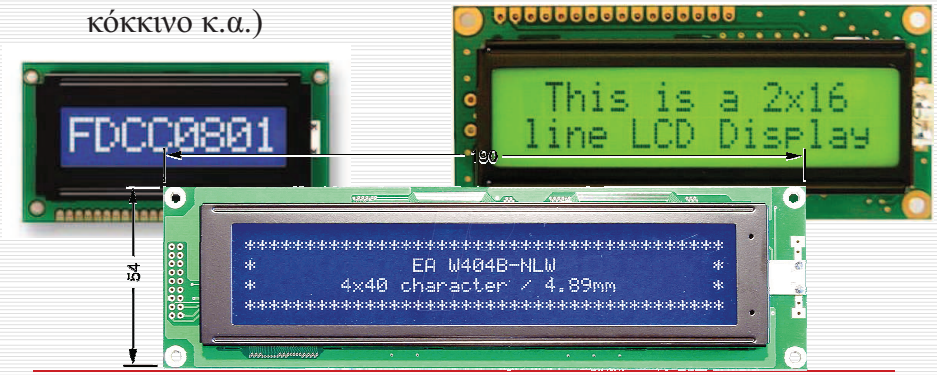
Εισαγωγή

- Αυτή την εβδομάδα θα μάθουμε:

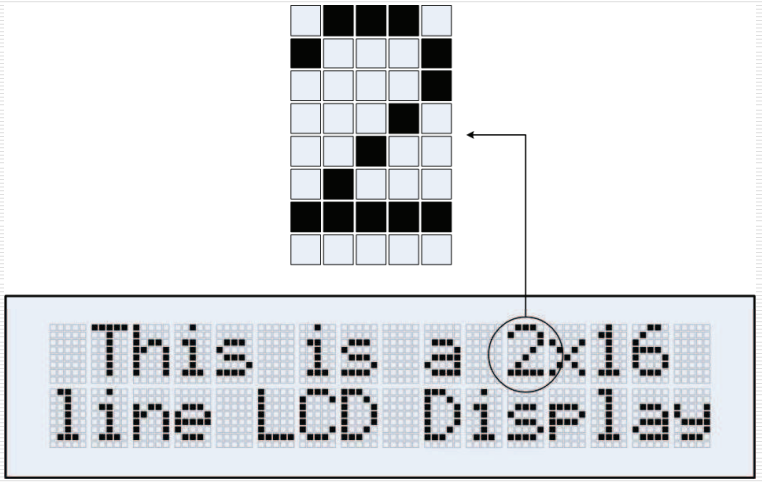
- Για τις οθόνες χαρακτήρων τύπου LCD
- Για τη διασύνδεσή τους με μικροελεγκτές
- Για τη χρήση ενσωματωμένων βιβλιοθηκών του Arduino IDE

Είδη οθονών

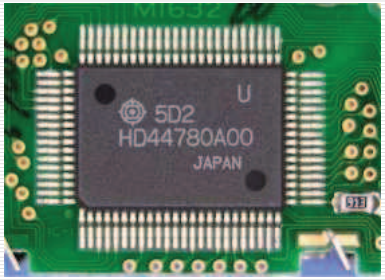
- Διάφορα μεγέθη (8x1, 8x2, 16x1, 16x2, 20x2, 20x4, 24x2, 24x4, 40x2, 40x4)
- Διάφορα χρώματα οπίσθιου φωτισμού (πράσινα, μπλέ, κόκκινο κ.α.)



Δομή χαρακτήρα



HD44780

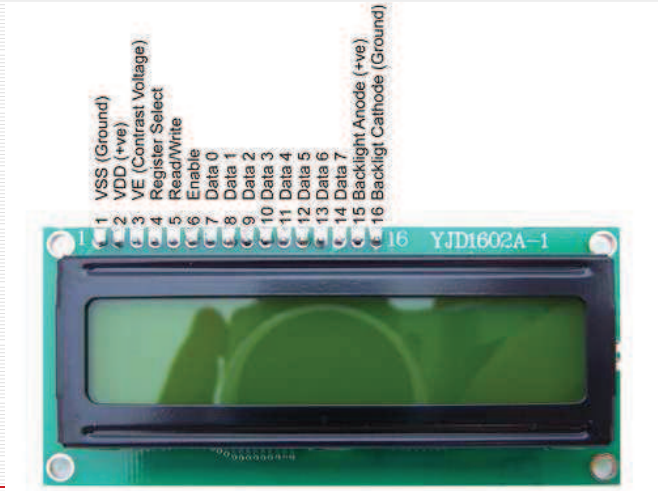


- για μονόχρωμες οθόνες χαρακτήρων
- 30 χρόνια στην πιάτσα
- ευρεία αποδοχή
- de facto standard

LCD 16x2 pins

pin	όνομα	Επεξήγηση	Περιγραφή
1	GND	Ground	
2	VCC	Τροφοδοσία	3.3V – 5V
3	VE ή VO	Contrast Voltage	
4	RS	Register Select	0 = command (εντολή) 1 = data (δεδομένα)
5	R/W	Read/Write	0 = write (εγγραφή) 1 = read (ανάγνωση)
6	EN	Enable	Αρνητικά ακμοπυροδήτητο ρολόι
[7 – 14]	[D0 – D7]	Data bits	Bits δεδομένων
15	Backlight Anode (+)		Άνοδος LED οπίσθιου φωτισμού
16	Backlight Cathode (-)		Κάθοδος LED οπίσθιου φωτισμού

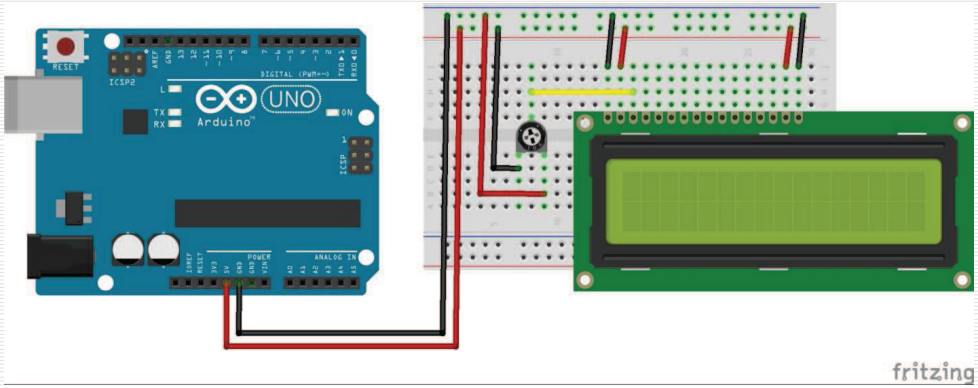
LCD 16x2 pins



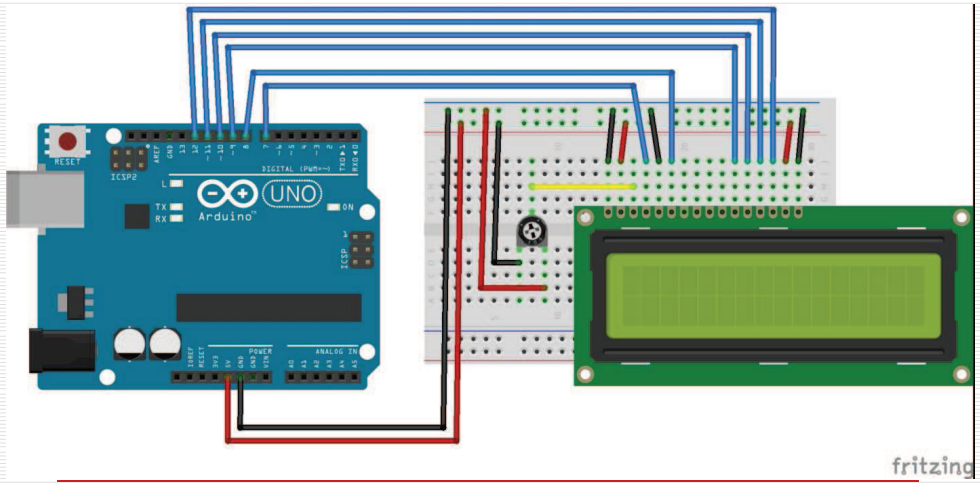
Υλικά

- 1 arduino uno board
- 1 breadboard
- 16 Jumper wires
- *(1 αντιστάτης 220Ω)*
- 1 trimmer 10KΩ
- 1 16x2 LCD display

Κύκλωμα ελέγχου backlight



Τελικό κύκλωμα



Κώδικας

- Files -> Examples ->LiquidCrystal - >HelloWorld

```
// include the library code:
#include <LiquidCrystal.h>
// initialize the library with the numbers of the interface pins
LiquidCrystal lcd(12, 11, 5, 4, 3, 2);
void setup() {
  // set up the LCD's number of columns and rows:
  lcd.begin(16, 2);
  // Print a message to the LCD.
  lcd.print("hello, world!");
}
void loop() {
  // set the cursor to column 0, line 1
lcd.setCursor(0, 1);
  // print the number of seconds since reset:
  lcd.print(millis() / 1000);
}
```


Κώδικας (επεξήγηση)

`#include <LiquidCrystal.h>`

- Με τη δήλωση `#include` δίνουμε οδηγία στον compiler να συμπεριλάβει στο αρχείο ο που δημιουργεί και τον κώδικα της συγκεκριμένης βιβλιοθήκης
- Η βιβλιοθήκη περιλαμβάνει δηλώσεις συναρτήσεων αλλά και αντικειμένων ή τύπων δεδομένων
- Ο σκοπός προφανής: δεν χρειάζεται να ξανανακαλύψουμε τον τροχό.

Κώδικας (επεξήγηση – συν.)

```
LiquidCrystal lcd(7, 8, 9, 10, 11, 12);
```

- Με τη δήλωση αυτή δημιουργούμε ένα αντικείμενο τύπου `LiquidCrystal` με όνομα `lcd`. Το αντικείμενο αυτό αντιστοιχεί στην φυσική LCD οθόνη μας.
- Ταυτόχρονα δίνουμε και κάποιες παραμέτρους.

```
LiquidCrystal(rs, enable, d4, d5, d6, d7)
LiquidCrystal(rs, rw, enable, d4, d5, d6, d7)
LiquidCrystal(rs, enable, d0, d1, d2, d3, d4, d5, d6, d7)
LiquidCrystal(rs, rw, enable, d0, d1, d2, d3, d4, d5, d6, d7)
```


Κώδικας (επεξήγηση – συν.)

```
void setup() {
  lcd.begin(16, 2);
  lcd.print("hello, world!");
}
```

- Η συνάρτηση `begin()` υπάρχει στην βιβλιοθήκη `LiquidCrystal` και τη χρησιοποιούμε για να αρχικοποιήσουμε την οθόνη, όπως στην πιο πάνω δήλωση.
- Η συνάρτηση `print()` της βιβλιοθήκης `LiquidCrystal` τυπώνει ένα μήνυμα στην τρέχουσα θέση του κέρσορα. Το μήνυμα αυτό θα είναι είτε string (ή χαρακτήρας) είτε η τιμή μιας μεταβλητής

Κώδικας (επεξήγηση – συν.)

```
void loop() {
  lcd.setCursor(0, 1);
  lcd.print(millis() / 1000);
}
```

- Η συνάρτηση `setCursor()` τοποθετεί τον κέρσορα στη θέση που ορίζουμε εμείς.
- Η `lcd.print(millis()/1000);` τυπώνει ένα αριθμό. Τι πιστεύετε ότι δείχνει αυτός ο αριθμός;

κάτι επιπλέον...

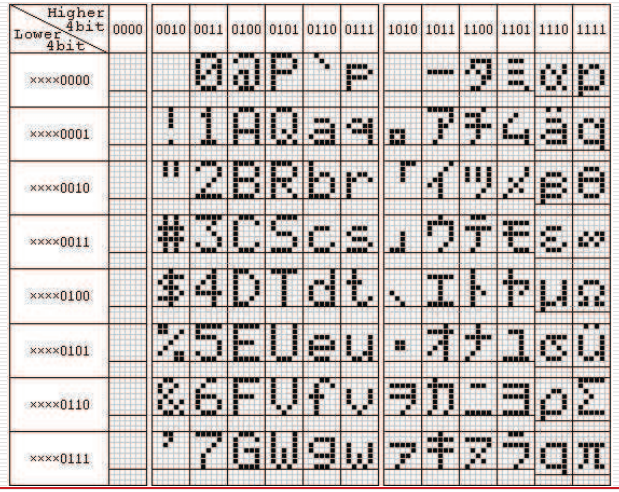
Δοκιμάστε το :

```
void setup() {
  lcd.begin(16, 2);
  lcd.print("π=3.14159");
}
```

ή το :

```
void setup() {
  lcd.begin(16, 2);
  lcd.print("R=10KΩ");
}
```

Πίνακας χαρακτήρων (πάνω τμήμα)



κάτι επιπλέον ...

Δοκιμάστε τώρα το :

```
void setup() {
  lcd.begin(16, 2);
  lcd.print(char(247));
  lcd.print("=3.14159");
}
```

? Πώς θα είναι η εντολή για εκτύπωση του ¨ R=10KΩ ¨;

Άσκηση για το σπίτι

1. **Βρείτε τις συναρτήσεις που έχει η βιβλιοθήκη LiquidCrystal και καταγράψτε τις. Επιλέξτε πέντε από αυτές (εκτός όσων είδαμε εδώ) κα γράψτε εν συντομία τι κάνει η κάθε μια από αυτές. Όπως πάντα, αν δεν τα πάτε και τόσο καλά με τα αγγλικά, για την μετάφραση μπορεί να σας βοηθήσει κάποιος γνωστός σας ή η καθηγήτρια των αγγλικών σας.**

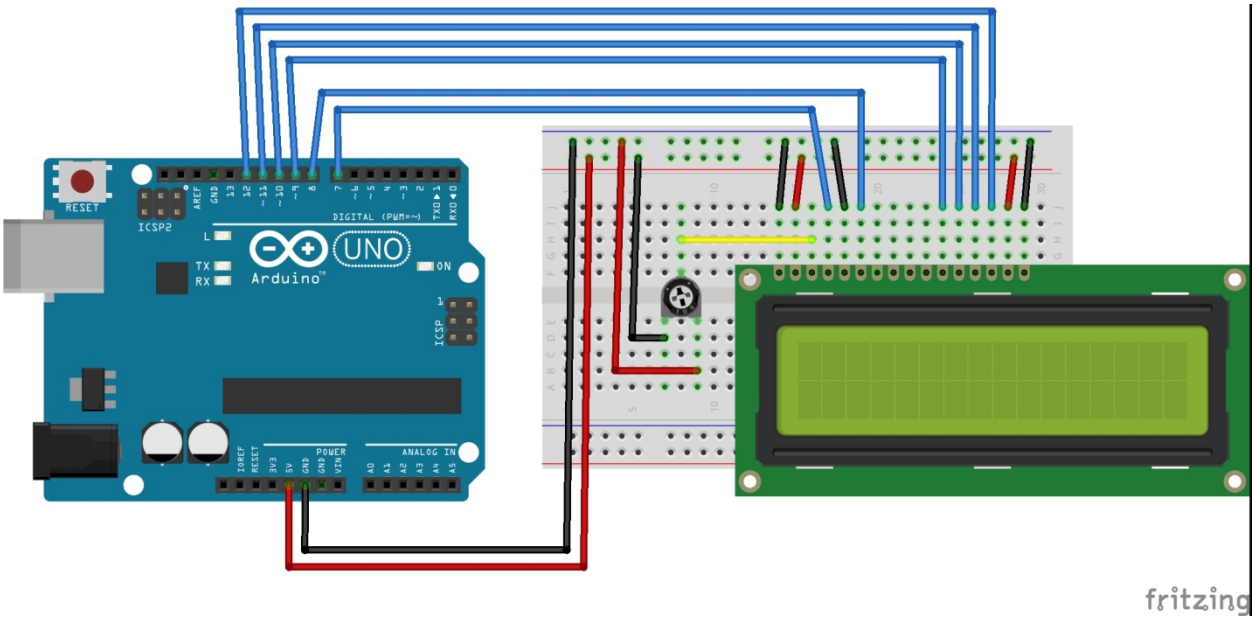
1^{ο} ΕΠΑ.Λ. Μεσολογγίου	**Σχολικό έτος : ____ – ____**
Τάξη : *Β΄$_{ΗΝ}$*	**Μάθημα : *Ηλεκτρονικές Διατάξεις***
Ημερομηνία: **…….**	**Μαθητές:** **……** **……** **……**

Άσκηση 17(β)
Θέμα : Απεικόνιση σε LCD display 16x2 (με σύνδεση I2C)

Σύντομη θεωρία

Την προηγούμενη εβδομάδα είδαμε πώς να συνδέουμε μια οθόνη χαρακτήρων LCD 16x2 στο Arduino και να προγραμματίζουμε το τελευταίο να εμφανίζει ένα μήνυμα στην οθόνη. Επίσης είδαμε το πως δηλώνουμε και χρησιμοποιούμε μια ενσωματωμένη βιβλιοθήκη στο Arduino IDE. Αυτή την εβδομάδα θα συνεχίσουμε να δουλεύουμε με τις οθόνες LCD αλλά θα χρησιμοποιήσουμε έναν διαφορετικό τρόπο διασύνδεσης.

Η σύνδεση που χρησιμοποιήσαμε στην προηγούμενη άσκηση ήταν παράλληλη. Δηλαδή τα δεδομένα μεταφέρονταν από το Arduino στην οθόνη μας κατα ομάδες των 4-bits (αυτό ονομάζεται και nibble και είναι ουσιαστικά μισό byte). Θα μπορούσαμε φυσικά να μεταφέρουμε τα δεδομένα ανά byte αλλά αυτό θα απαιτούσε 4 επιπλέον γραμμές δεδομένων. Στο επόμενο σχήμα φαίνεται η διασύνδεση που είχαμε χρησιμοποιήσει.

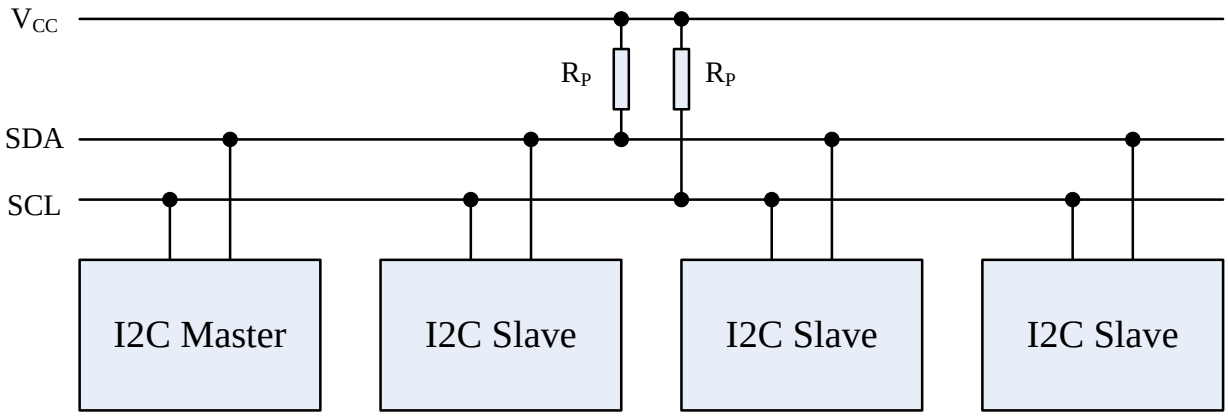


Σχήμα 17.1 Παράλληλη σύνδεση LCD οθόνης για τη μεταφορά nibbles

Παρατηρούμε ότι πέρα από την τροφοδοσία και τη γείωση, απαιτούνται 6 επιπλέον γραμμές για τη μεταφορά των δεδομένων στην οθόνη. Αυτό, αν και είναι ορθό και λειτουργικό, είναι μια τεράστια σπατάλη ακροδεκτών του Arduino. Μπορεί να μην φαίνεται τώρα αλλά αν ετοιμάζετε ένα μεγαλύτερο project (όπως στην πραγματικότητα κάνουμε και εμείς), ο πεπερασμένος αριθμός των ακροδεκτών του Arduino Uno που χρησιμοποιούμε είναι μια περιοριστική παράμετρος.

Για να μην ξοδεύουμε άσκοπα ακροδέκτες λοιπόν θα πρέπει να βρεθεί ένας διαφορετικός τρόπος διασύνδεσης. Και αυτός είναι η διασύνδεση με δίαυλο I2C (ή IIC ή I^2C). Ο δίαυλος I2C είναι ένας **σειριακό**ς δίαυλος που δημιουργήθηκε από τη Philips και χρησιμοποιείται για την σύνδεση περιφερειακών μικρής ταχύτητας σε ηλεκτρονικές συσκευές. Χρησιμοποιεί μόνο δύο καλώδια, τα οποία είναι αμφίδρομης κατεύθυνσης: Τα SCL και SDA. Η γραμμή SCL (serial clock) είναι η γραμμή ρολογιού, ενώ η SDA (serial data) είναι η γραμμή δεδομένων. Οι γραμμές αυτές συνδέονται σε όλες τις συσκευές στον δίαυλο I2C.

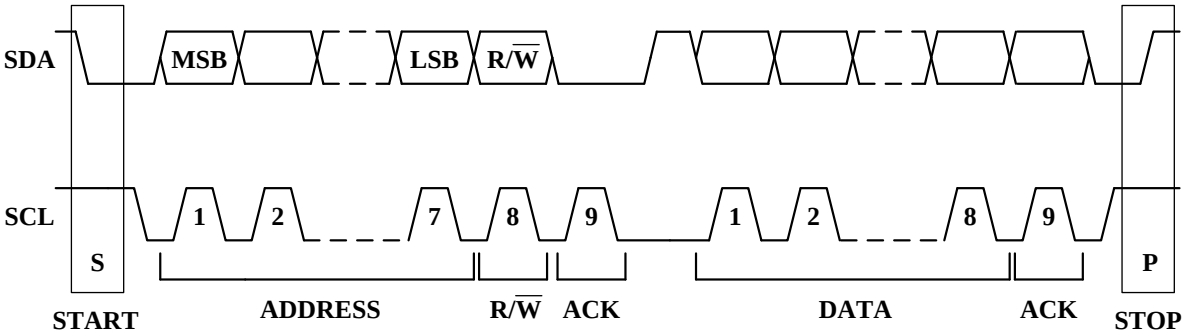
Μια τυπική διασύνδεση σε δίαυλο I2C φαίνεται στο επόμενο σχήμα. I2C Master είναι ο κόμβος που δημιουργεί το σειριακό ρολόι SCL και ξεκινά την επικοινωνία ενώ I2C Slave είναι ο κόμβος που δέχεται το ρολόι και απαντά στον Master. Κάθε φορά μπορούν να υπάρχουν περισσότεροι από ένας Masters ενώ οι ρόλοι Master και Slave μπορούν να εναλλάσσονται.



Σχήμα 17.2 Τυπική διασύνδεση σε δίαυλο I2C

Οι αντιστάσεις R_P είναι αντιστάσεις pull up και διατηρούν την στάθμη στις γραμμές SCL και SDA σε λογικό high όσο δεν υπάρχει επικοινωνία.

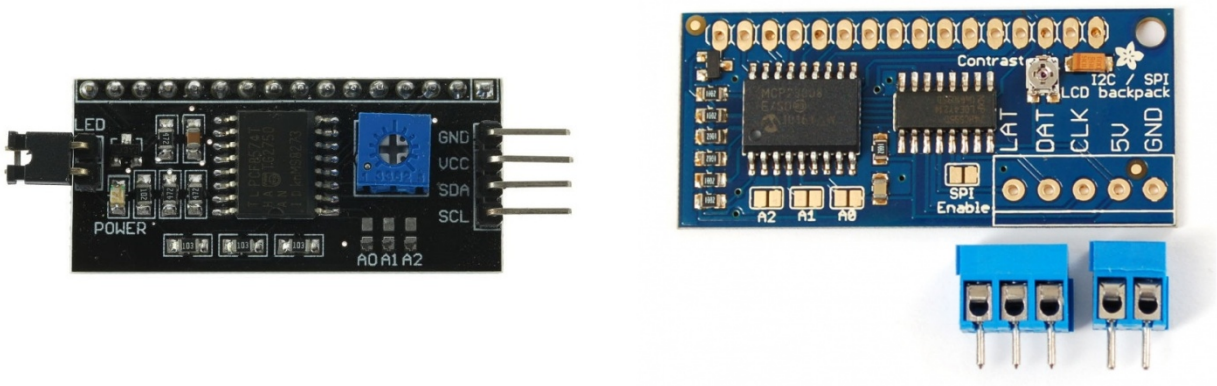
Για να μπορούν να ξεχωρίσουν οι διάφοροι κόμβοι μεταξύ τους και να ξέρουν σε ποιον απευθύνεται ο Master, ο καθένας έχει μια διαφορετική διεύθυνση (7-bit ή 10-bit) την οποία τοποθετεί ο Master στον δίαυλο SDA όταν ξεκινά μια επικοινωνία.



Σχήμα 17.3 Παράδειγμα χρονισμού σε δίαυλο I2C

Όταν ο σε ένα δίαυλο I2C ο Master θέλει να ξεκινήσει μια επικοινωνία, το δηλώνει με τη συνθήκη S (Start). Όπου ουσιαστικά οδηγή την γραμμή SDA στο λογικό 0 και μετά ενεργοποιεί το ρολόι SCL. Στη συνέχεια τοποθετεί την επιθυμητή διεύθυνση στο δίαυλο (με το περισσότερο σημαντικό ψηφίο πρώτο). Στο παράδειγμα του σχήματος 17.3 η διεύθυνση αυτή είναι μήκους 7-bit. Ένα 8° bit ($R/\overline{W}$) δηλώνει αν ο Master θέλει να διαβάσει από ή να γράψει στην συσκευή Slave. Η συσκευή Slave που αναγνώρισε τη διεύθυνσή της θα πρέπει αν απαντήσει στον επόμενο παλμό με ένα σήμα acknowledgment (ACK). Κατόπιν στη γραμμή SDA θα τοποθετηθούν τα δεδομένα (είτε από τον Master είτε από τον Slave). Ένα ακόμα σήμα ACK θα δηλώσει την επιτυχή λήψη για κάθε byte δεδομένων. Όταν ολοκληρωθεί η μεταφορά των δεδομένων ο Master στέλνει στο δίαυλο τη συνθήκη P (Stop).

Μέχρι εδώ όλα καλά, αλλά πώς θα συνδέσω μια οθόνη LCD με παράλληλο interface σε ένα σειριακό δίαυλο; Η απάντηση είναι ότι θα χρειαστώ ένα επιπλέον ηλεκτρονικό εξάρτημα. Στην επόμενη εικόνα φαίνονται 2 τέτοια κοινά modules.



(α) (β)

Σχήμα 17.4 Δυο κοινά modules για διασύνδεση οθόνης LCD 16x2 σε δίαυλο I2C

Τα modules αυτά αναλαμβάνουν την επικοινωνία μεταξύ μικροελεγκτή (Arduino) και οθόνης μέσω του I2C διαύλου. Πριν δούμε τον τρόπο με τον οποίο συνδέονται Arduino, LCD backpack module και LCD οθόνη να αναφέρουμε ότι η διεύθυνση των παραπάνω modules στο δίαυλο I2C είναι η 0. Αν

θέλουμε να την αλλάξουμε θα πρέπει να βραχυκυκλώσουμε τα jumpers A2, A1 και A0 που φαίνονται στην επιθυμητή διεύθυνση, π.χ. αν θέλουμε την διεύθυνση 2 (010_2) θα πρέπει να βραχυκυκλώσουμε το A1 ενώ για την 5 (101_2) θα πρέπει να βραχυκυκλώσουμε το A2 και το A0.

Σημαντική παρατήρηση: μερικά αρθρώματα I2C για LCD οθόνες δεν έχουν τέτοια jumpers. Η διεύθυνσή τους είναι συνήθως η 0x27. Για να βρούμε την διεύθυνση θα πρέπει να ανατρέχουμε στο σχετικό datasheet. Καλό θα ήταν αυτό να γίνεται σε κάθε περίπτωση. Στο διαδίκτυο υπάρχουν και προγράμματα για Arduino που ελέγχουν (σκανάρουν) ποια διεύθυνση έχει κάθε I2C συσκευή. Ένα τέτοιο βρίσκεται εδώ: http://arduino-info.wikispaces.com/LCD-Blue-I2C

Υλικά:

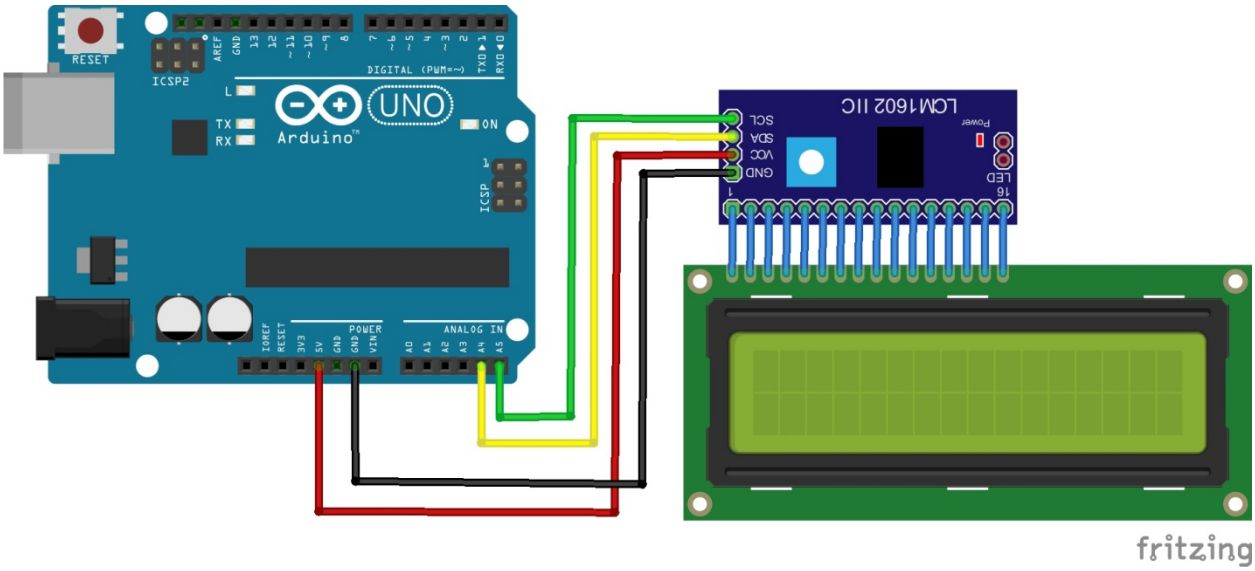
1 arduino uno board
1 breadboard
4 Jumper wires
1 16x2 LCD display
1 LCD display Backpack με 16-pin long 0.1" female header

Λογισμικό – Βιβλιοθήκες :

Βιβλιοθήκη **Wire** ενσωματωμένη στο arduino IDE.
Βιβλιοθήκη **LiquidCrystal_I2C** από τον F. Malpartida. Η βιβλιοθήκη είναι διαθέσιμη εδώ:
https://bitbucket.org/fmalpartida/new-liquidcrystal/downloads, το πιο νέο αρχείο (τον Νοέμβριο 2015 ήταν το NewliquidCrystal_1.3.4.zip).
(Η βιβλιοθήκη υπάρχει και στον τοπικό δίκτυο, στον κοινόχρηστο φάκελο στο server, σε περίπτωση που δεν είναι δυνατόν να την κατεβάσετε)

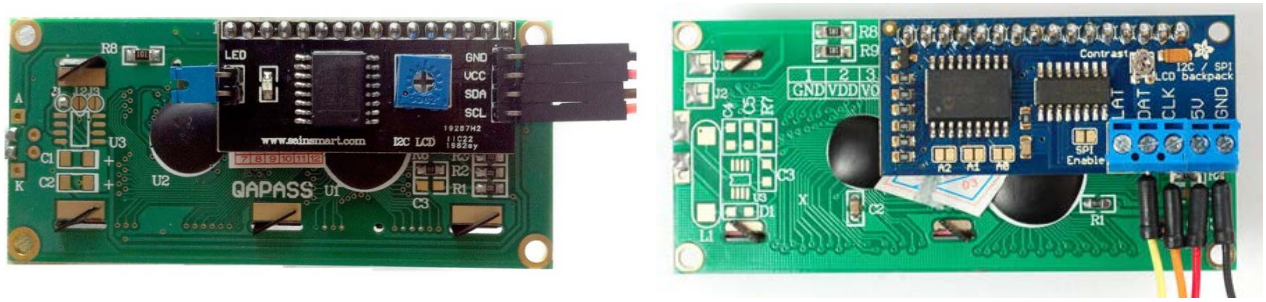
Υλοποίηση:

Το κύκλωμα που θα υλοποιήσουμε είναι το ακόλουθο:



Σχήμα 17.5. Συνδεσμολογία arduino και I2C LCD module

Βέβαια η LCD οθόνη συνδέεται απευθείας πάνω στο I2C LCD άρθρωμα όπως στην επόμενη εικόνα:



(α) (β)

Σχήμα 17.6. Σύνδεση LCD οθόνης και I2C LCD module για τα 2 modules του σχήματος 17.3

Στο κύκλωμα αυτό έχουμε συνδέσει την τροφοδοσία και τη γείωση του I2C LCD αρθρώματος στα αντίστοιχα του Arduino, ενώ το σεριακό ρολόι (SCL) συνδέεται στον ακροδέκτη A5 του Arduino και τα σειριακά δεδομένα στον A4. Θυμηθείτε ότι οι ακροδέκτες A0-A5 είναι αναλογικές είσοδοι αλλά μπορούν να χρησιμοποιηθούν και σαν ψηφιακές είσοδοι/έξοδοι. Επιπλέον, η βιβλιοθήκη wire του arduino (θα χρησιμοποιηθεί και αυτή) υλοποιεί το δίαυλο I2C και χρησιμοποιεί συγκεκριμένα τον ακροδέκτη A5 για SCL ενώ τον A4 για SDA.

<u>Κώδικας:</u>

Αφού υλοποιήσουμε το κύκλωμα (δεν είναι δύσκολο, μόνο 4 καλώδια) θα πρέπει να εγκαταστήσουμε και την βιβλιοθήκη που κατεβάσαμε. Ανοίξτε το περιβάλλον Arduino IDE και πηγαίνετε στο menu Σχέδιο -> Εισαγωγή Βιβλιοθήκης… -> Add library…. Στο παράθυρο που αναδύθυκε βρείτε και επιλέξτε το αρχείο NewliquidCrystal_1.3.4.zip που κατεβάσατε. Επιλέξτε open και η βιβλιοθήκη θα εγκατασταθεί. Εναλλακτικά βρείτε το αρχείο NewliquidCrystal_1.3.4.zip και αποσυμπιέστε το στην επιφάνεια εργασίας. Ανοίξτε το φάκελο όπου έγινε η αποσυμπίεση και αντιγράψτε το φάκελο NewliquidCrystal στο φάκλεο Libraries εκεί που είναι εγκατεστημένο το Arduino IDE (στο εργαστήριο είναι στο C:\arduino-1.6.6\libraries\).

Αν όλα έγιναν σωστά, όταν ανοίγετε το Arduino IDE θα πρέπει στα παραδείγματα (File -> Examples) να υπάρχει και νέα καταχώρηση, η NewLiquidCrystal, όπου θα βρείτε έτοιμα παραδείγματα για το I2C LCD module. Εμείς όμως θα χρησιμοποιήσουμε τον παρακάτω κώδικα. Μπορείτε να τον πληκτρολογήσετε ή να τον βρείτε στη κοινόχρηστο φάκελο του server του τοπικού δικτύου με όνομα HelloWorld_i2c_17.ino

```
/*
 16x2 LCD με I2C backpack
```

```
 Κώδικας γραμμένος από τον Παπαζαχαρία Αθανάσιο το Δεκ 2015
 Βασισμένος σε κώδικα από τα παραδείγματα της βιλιοθήκης NewLiquidCrystal

 Περιγράφει τον τρόπο για να δημιουργήσετε το δικούς σας χαρακτήρες

  Το κύκλωμα:
 * LCD I2C VCC pin στην τροφοδοσία (5V)
 * LCD I2C GND pin στη γείωση (GND)
 * LCD I2C SDA pin στο αναλογικό pin A4
 * LCD I2C SCL pin στο αναλογικό pin A5
 */

#include <Wire.h>
#include <LiquidCrystal_I2C.h>

LiquidCrystal_I2C lcd(0x27, 2, 1, 0, 4, 5, 6, 7, 3, POSITIVE);

// δημιουργία πίνακα για ένα νέο χαρακτήρα
byte frownie[8] = {
  0b00000,
  0b01010,
  0b00000,
  0b00000,
  0b01110,
  0b10001,
  0b00000,
  0b00000
};

void setup()
{
  lcd.begin(16,2);                // αρχικοποίηση της lcd οθόνης
  lcd.createChar (0, frownie);    // φόρτωση του χαρακτήρα 0 στον ελεγκτή της
οθόνης
  lcd.home ();                    // πήγαινε τον κέρσορα στη θέση (0,0)
  lcd.print("Hello World! (I2C) ");
}

void loop()
{
  lcd.setCursor ( 0, 1 ); // πήγαινε στην επόμενη γραμμή
  lcd.print("SORRY! YOU LOSE ");
  lcd.print (char(0));
  }
```

Αν τρέξετε τον κώδικα θα δείτε στην οθόνη σαν να εμφανίζεται το μήνυμα ¨ SORRY! YOU LOSE ☹ ¨ που είχαμε αναφέρει στο προηγούμενο μάθημα (εντάξει, χρειάζεται και λίγη φαντασία για να δεις τη θλιμένη φατσούλα). Επεξηγώντας τον κώδικα βήμα βήμα θα δούμε πώς φτιάξαμε τη φατσούλα και θα πρέπει και εσείς να φτιάξετε ένα δικό σας χαρακτήρα (ίσως τη χαρούμενη φάτσα).

Αρχικά συμπεριλαμβάνουμε στο σχέδιό μας 2 βιβλιοθήκες.

```
#include <Wire.h>
#include <LiquidCrystal_I2C.h>
```

Η μια είναι η Wire, η οποία είναι ήδη ενσωματωμένη στο Arduino IDE και υλοποιεί (μεταξύ άλλων) τον δίυλο I2C και η άλλη είναι η LiquidCrystal_I2C η οποία θα μας βοηθήσει να χρησιμοποιήσουμε το I2C LCD module.

Για να μπορέσει να λειτουργήσει η οθόνη μας θα πρέπει να δηλώσουμε τη διεύθυνσή της στον κώδικα. **Προσοχή** καθώς πολλά παραδείγματα περιλαμβάνουν την εντολή

```
LiquidCrystal_I2C lcd(0x38);
```

όπου θεωρούν τη διεύθυνση της οθόνης να είναι η 0x38. Αν συμβαίνει αυτό θα πρέπει να αλλάζουμε τη δήλωση αυτή με την εντολή:

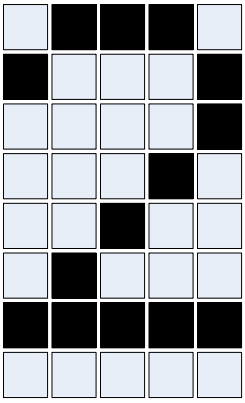
```
LiquidCrystal_I2C lcd(0x27, 2, 1, 0, 4, 5, 6, 7, 3, POSITIVE);
```

Σε αυτή την περίπτωση ο constructor LiquidCrystal_I2C() δέχεται σαν ορίσματα τα:

```
addr, EN, RW, RS, D4, D5, D6, D7, Backlight, POLARITY
```

δηλαδή τη διεύθυνση του I2C module, καθώς και τα pins του module που συνδέονται στα EN, RW, RS, D4, D5, D6, D7 και Backlight της LCD οθόνης. Ο οπίσθιος φωτισμός της οθόνης ελέγχεται από το trimmer που βρίσκεται πάνω στο άρθρωμα I2C.

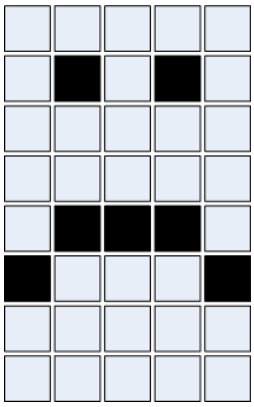
Στη συνέχεια δημιουργούμε έναν πίνακα από 8 bytes ο οποίος θα χρησιμοποιηθεί για την δημιουργία ενός χαρακτήρα. Για να το κατανοήσουμε κλύτερα θα πρέπει να θυμηθούμε ότι, όπως είπαμε και στο προηγούμενο μάθημα, ο κάθε χαρακτήρας της LCD οθόνης σχηματίζεται από ένα πίνακα (matrix) 5x8 κουκίδων. Για παράδειγμα, για τον αριθμό 2 μαυρίζουν οι κουκίδες που φαίνονται στο επόμενο σχήμα.



Σχήμα 17.7. Απεικόνιση του αριθμού 2 σε matix 5x8 dots

Είχαμε δει επίσης ότι οι χαρακτήρες είναι αποθηκευμένοι σε μια μνήμη ROM του ελεγκτή HD44780 και υπάρχει και σχετικός πίνακας (δες προηγούμενη άσκηση, πίνακας 17.2). Παρατηρήστε ότι δεν υπάρχουν χαρακτήρες σε όλες τις θέσεις του πίνακα. Οι πρώτες θέσεις του πίνακα είναι κενές. Σε αυτές τις θέσεις μπορεί κανείς να αποθηκεύσει δικούς του χαρακτήρες και εδώ θα μάθουμε πως.

Καταρχάς θα πρέπει να ξέρουμε πως θέλουμε να είναι ο χαρακτήρας μας. Πρέπει λοιπόν πρώτα να τον σχεδιάσουμε σαν ένα πίνακα 5x8 κουκίδων. Στο σχήμα 17.8α φαίνεται η σχεδίαση που κάναμε για τη θλιμμένη φατσούλα (frownie).



(α)

0	0	0	0	0
0	1	0	1	0
0	0	0	0	0
0	0	0	0	0
0	1	1	1	0
1	0	0	0	1
0	0	0	0	0
0	0	0	0	0

(β)

Σχήμα 17.8 Απεικόνιση του frownie σε matix 5x8 dots

Στο σχήμα 17.8β βλέπουμε την αντιστοιχία του σχήματος 17.8α σε bits όπου χρησιμοποιούμε το 1 για τις κουκκίδες που θα πρέπει να μαυρίσουν και το 0 για τις υπόλοιπες. Έτσι λοιπόν, για να μπορέσουμε να δημιουργήσουμε τον χαρακτήρα θα πρέπει πρώτα να ορίσουμε ποιες κουκίδες θα μαυρίσουν και ποιες όχι. Στον κώδικά μας αυτό γίνεται με την δήλωση:

```
byte frownie[8] = {
  0b00000,
  0b01010,
  0b00000,
  0b00000,
  0b01110,
  0b10001,
  0b00000,
  0b00000
};
```

Εκεί ορίζουμε ένα πίνακα από 8 bytes (δηλαδή ένα πίνακα 8x8 bit) και δίνουμε τις κατάλληλες τιμές. Δεν χρειάζεται να δηλώσουμε και τα 8 bits του κάθε byte, μόνο τα 5 λιγότερο σημαντικά καθώς τα περισσότερο σημαντικά θα πάρουν αυτόματα την τιμή 0. Η δήλωση όμως του πίνακα δεν σημαίνει και δημιουργία του χαρακτήρα. Αυτό θα γίνει παρακάτω, στη συνάρτηση **setup**()

Στη setup πρώτα αρχικοποιούμε την οθόνη με την δήλωση

```
lcd.begin(16,2);
lcd.print("Hello World! (I2C) ");
```

με την οποία αρχικοποιούμε την οθόνη μας. Στη συνέχεια δημιουργούμε τον χαρακτήρα μας με την εντολή

```
lcd.createChar(0, frownie);
```

όπου δίνουμε τη θέση όπου θα δημιουργηθεί ο χαρακτήρας (θέση 0) και τον πίνακα των bytes που αντιστοιχεί στις κουκίδες (frownie).

Με την εντολή

```
lcd.home ();
```

ο κέρσορας επιστρέφει στην αρχή της οθόνης όπου τυπώνουμε ένα μήνυμα με την:

```
lcd.print("Hello World! (I2C) ");
```

Στη συνάρτηση **loop()** πάμε στην επόμενη γραμμή

```
lcd.setCursor ( 0, 1 );
```

Όπου τυπώνουμε το μήνυμα “SORRY! YOU LOSE ” ακολουθούμενο από τον χαρακτήρα μας.

```
lcd.print("SORRY! YOU LOSE ");
lcd.print char(0));
```

Σημείωση: για την εκτύπωση του χαρακτήρα 0 καλό είναι να χρησιμοποιούμε τη πιο πάνω δήλωση. Για τους χαρακτήρες σε άλλες θέσεις (π.χ. την 1) μπορούμε να χρησιμοποιούμε και τη δήλωση

```
lcd.print(1);
```

Αν έχετε χρόνο σχεδιάστε και δικούς σας χαρακτήρες και εμφανίστε τους στην οθόνη. Χρησιμοποιείστε τους παρακάτω πίνακες για να σχεδιάσετε τους χαρακτήρες.

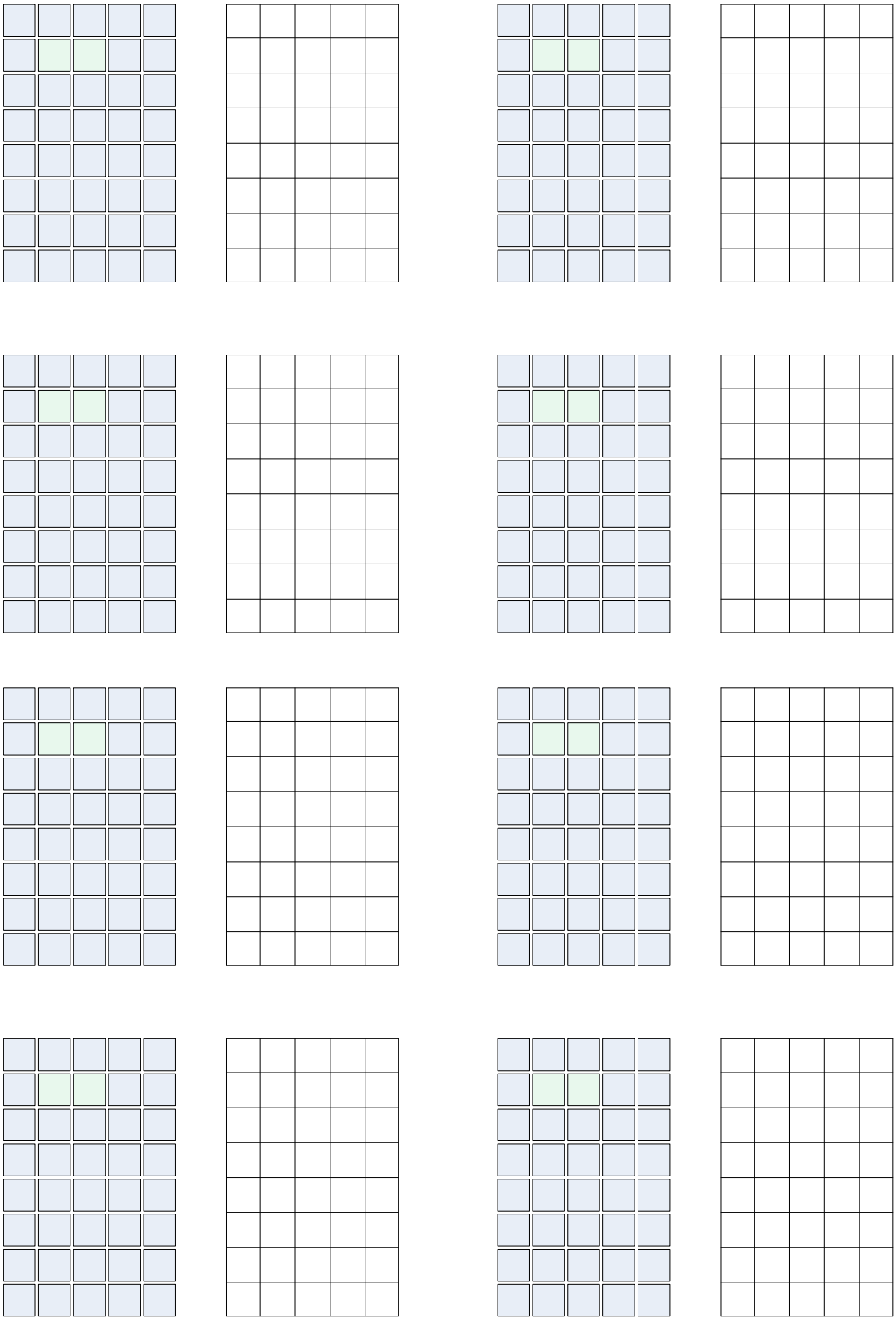
Εργασία για το σπίτι

Ψάξτε στο διαδίκτυο και βρείτε πέντε (5) ακόμα συσκευές/αισθητήρες που μπορούμε να συνδέσουμε στο Arduino και οι οποίες/οι χρησιμοποιούν τον δίαυλο I2C. Για κάθε μια από αυτές να βρείτε και τη διεύθυνση που συνήθως χρησιμοποιεί στον δίαυλο I2C.

Και αν έχετε διάθεση……

Τροποποιήστε τον πιο πάνω κώδικα ώστε στην οθόνη να εμφανίζονται στο ίδιο σημείο με τη σειρά οι χαρακτήρες:

Μεταξύ δυο διαδοχικών χαρακτήρων να περνά 1/10 του δευτερολέπτου περίπου.



1^{ο} ΕΠΑ.Λ. Μεσολογγίου	**Σχολικό έτος : ____ – ____**
Τάξη : *Β΄$_{ΗΝ}$*	**Μάθημα : *Ηλεκτρονικές Διατάξεις***
Ημερομηνία: **…….**	**Μαθητής:** **……**

Άσκηση 17(β)
Θέμα : Απεικόνιση σε LCD display 16x2 (με σύνδεση I2C)

Φύλλο Εργασίας

Ποιες από τις παρακάτω προτάσεις θεωρείτε ότι είναι σωστές;

(μπορεί κάθε ερώτηση να έχει περισσότερες από μια σωστές απαντήσεις)

Α. Τα πλεονεκτήματα της σειριακής επικοινωνίας έναντι της παράλληλης είναι:

1. Η μεγαλύτερη ταχύτητα αποστολής δεδομένων
2. Ο μικρότερος αριθμός γραμμών
3. Η εγκυρότητα των δεδομένων που μεταφέρονται
4. Η οικονομία ακροδεκτών στις συνδεδεμένες συσκευές

Β. Ο δίαυλος I2C αποτελείται από τις γραμμές

1. SDA και SCL
2. SCL, MOSI και MISO
3. VCC, GND, SDA και SCL
4. VCC και GND

Γ. Στο δίαυλο I2C

1. το ρολόι δημιουργείται μόνο από τον master
2. τα δεδομένα στέλνονται μόνο από τον slave
3. Οι ρόλοι master και slave είναι σαφώς ορισμένοι και δεν αλλάζουν
4. Η διευθύνσεις στέλνονται μόνο από τον master

Δ. Ατο Arduino σαν SCL και SDA χρησιμοποιούνται :

1. Οι ψηφιακοί ακροδέκτες 1 και 0
2. Οι ψηφιακοί ακροδέκτες 5 και 4
3. Οποιοιδήποτε αναλογικοί ακροδέκτες
4. Οι αναλογικοί ακροδέκτες 5 και 4

Ε. Οι βιβλιοθήκες που χρειάζονται για την επικοινωνία του Arduino με το σειριακό module που διαθέτουμε είναι οι :

1. Μόνο η Wire
2. Η Wire και η LiquidCrystal
3. Η Wire και η LiquidCrystal_I2C
4. Μονο η LiquidCrystal_I2C

ΣΤ. Η διεύθυνση του I2C LCD module που χρησιμοποιήσαμε:

1. Ορίζεται από το πρόγραμμα (όποια θέλουμε)
2. Είναι η 0x38
3. Είναι η 0x27
4. Από το πρόγραμμα επιλέγουμε μια από τις 0x38 και 0x27

Ζ. Οι χαρακτήρες σε μια οθόνη LCD χαρακτήρων, απεικονίζονται σαν πίνακας από

1. 5x5 κουκίδες
2. 5x8 κουκίδες
3. 8x8 κουκίδες
4. 7x8 κουκίδες

Η. Η συνάρτηση της LiquidCrystal η οποία δημιουργεί τον χαρακτήρα είναι η:

1. begin
2. setCursor
3. print
4. createChar

Ηλεκτρονικές Διατάξεις

Άσκηση 17

LCD Display (σειριακή σύνδεση)

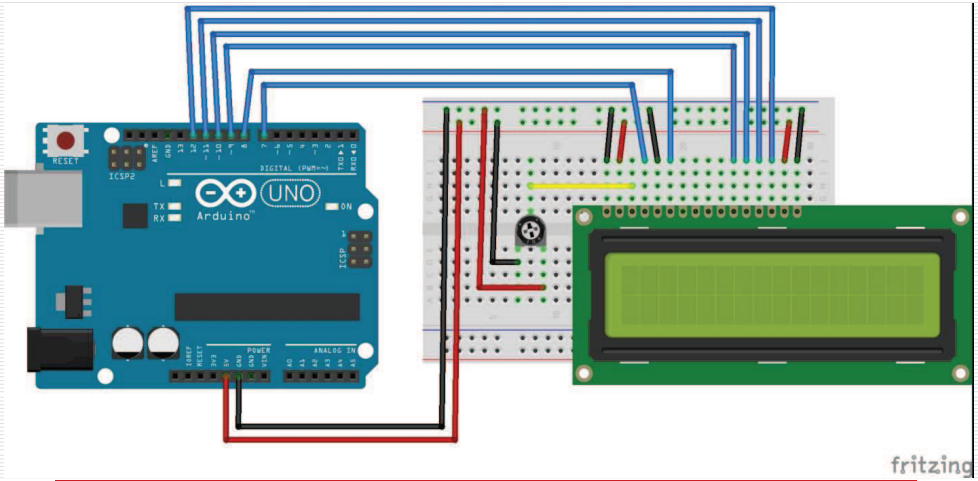
1ο ΕΠΑ.Λ. Μεσολογγίου
Τομέας Ηλεκτρονικής
Σχ. Έτος : 2015-2016
Διδάσκων: Παπαζαχαρίας Αθανάσιος

Εισαγωγή

Αυτή την εβδομάδα θα μάθουμε:

- Για τον δίαυλο σειριακής επικοινωνίας I2C
- Για το άρθρωμα I2C LCD module
- Πώς να εισάγουμε βιβλιοθήκες στο Arduino IDE
- Πώς να σχεδιάζουμε δικούς μας χαρακτήρες

Κύκλωμα παράλληλης σύνδεσης Arduino – LCD οθόνης (4-bit)



Παράλληλη σύνδεση

+ Μεγάλη ταχύτητα μεταφοράς δεδομένων
+ Δεν απαιτείται επιπλέον κύκλωμα

— Μεγάλος αριθμός γραμμών
— Μεγάλη κατανάλωση ακροδεκτών του Arduino

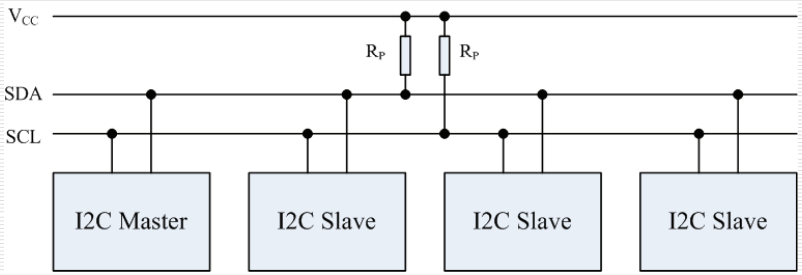
Σειρακή σύνδεση

+ Μικρός αριθμός γραμμών
+ Μικρή κατανάλωση ακροδεκτών του Arduino

— Μικρότερη ταχύτητα μεταφοράς δεδομένων
— Απαιτείται επιπλέον κύκλωμα

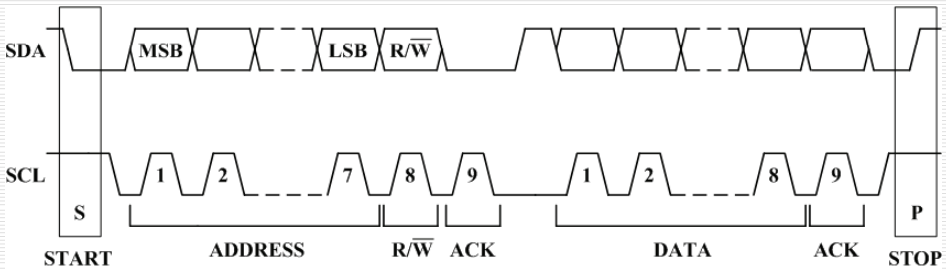
- I2C ή I²C ή IIC
- SPI

Δίαυλος I2C

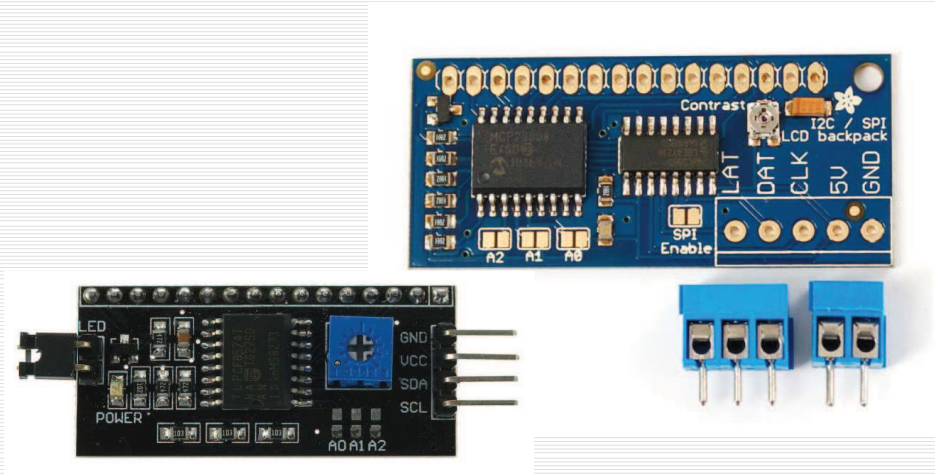


- Ο master παράγει το ρολόι (SCL) και ξεκινά μια
- Οποιοσδήποτε αριθμός από masters και slaves
- Οι ρόλοι master και slave μπορούν να εναλλάσσονται

Δίαυλος I2C - χρονισμός



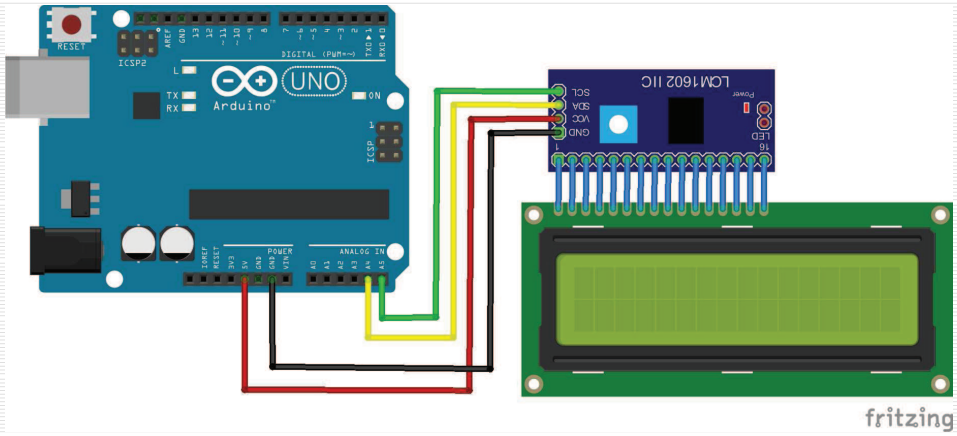
LCD 16x2 I2C modules



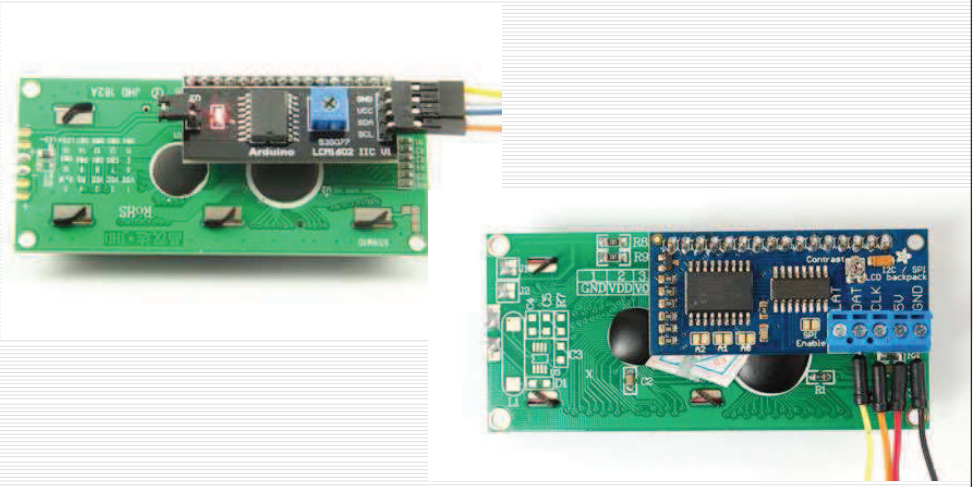
Υλικά

- 1 arduino uno board
- 1 breadboard
- 4 Jumper wires
- 1 16x2 LCD display
- 1 LCD display Backpack με 16-pin long 0.1" female header

Κύκλωμα



Σύνδεση οθόνης και module



Βιβλιοθήκες

- Wire (ενσωματωμένη στο Arduino IDE)
- LiquidCrystal_I2C (από τον F. Malpartida)

url:
https://bitbucket.org/fmalpartida/new-liquidcrystal/downloads

αρχείο:
NewliquidCrystal_1.3.4.zip

Εγκατάσταση βιβλιοθήκης

μέσω Arduino IDE

- Σχέδιο -> Εισαγωγή Βιβλιοθήκης… -> Add library….
- NewliquidCrystal_1.3.4.zip
- open

Απευθείας
- NewliquidCrystal_1.3.4.zip -> αποσυμπίεση
- αντιγραφή του φακέλου NewliquidCrystal στο C:\arduino-1.6.6\libraries\

Κώδικας

```
#include <Wire.h>
#include <LiquidCrystal_I2C.h>
LiquidCrystal_I2C lcd(0x27, 2, 1, 0, 4, 5, 6, 7, 3, POSITIVE);
// create new character
byte frownie[8] = {
  0b00000,
  0b01010,
  0b00000,
  0b00000,
  0b01110,
  0b10001,
  0b00000,
  0b00000
};
void setup()
{
  lcd.begin(16,2);                // initialize the lcd
  lcd.createChar (0, frownie);    // load character 0 to the LCD
  lcd.home ();                    // go home (line 0, column 0)
  lcd.print("Hello World! (I2C) ");
}
void loop()
{
  lcd.setCursor ( 0, 1 ); // go to next line
  lcd.print("SORRY! YOU LOSE ");
  lcd.print (char(0));
  }
```


Κώδικας (επεξήγηση)

```
#include <Wire.h>
#include <LiquidCrystal_I2C.h>
```

- Η βιβλιοθήκη wire είναι ενσωματωμένη στο Arduino IDE
- Η βιβλιοθήκη LiquidCrystal_I2C πρέπει να εισαχθεί με τον τρόπο που περιγράψαμε πιο πριν
- Χρησιμοποιεί την Wire

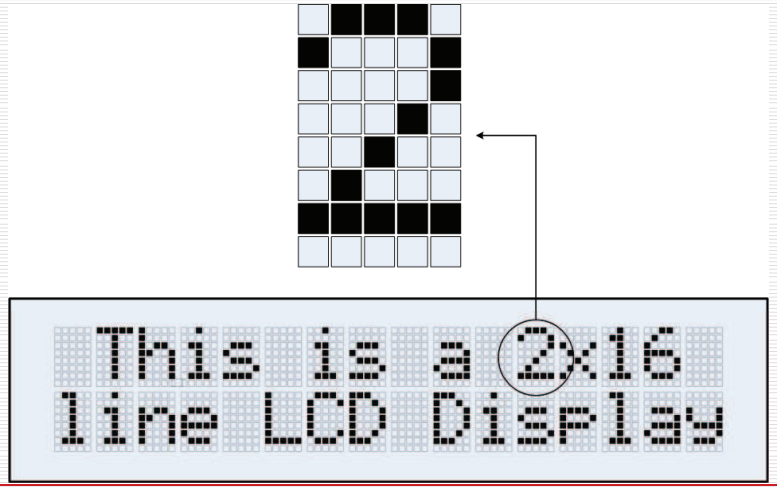
Κώδικας (επεξήγηση – συν.)

```
LiquidCrystal_I2C lcd(0x27, 2, 1, 0, 4,
5, 6, 7, 3, POSITIVE);
```

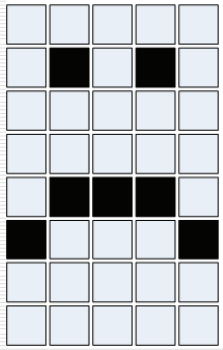
- Με τη δήλωση αυτή δημιουργούμε ένα αντικείμενο τύπου `LiquidCrystal_I2C` με όνομα `lcd`. Το αντικείμενο αυτό αντιστοιχεί στην φυσική LCD οθόνη μας.
- Ταυτόχρονα δίνουμε και κάποιες παραμέτρους.

```
LiquidCrystal(addr, EN, RW, RS, D4, D5, D6, D7, Backlight,
POLARITY)
```

Απεικόνιση χαρακτήρων



Απεικόνιση χαρακτήρων



0	0	0	0	0
0	1	0	1	0
0	0	0	0	0
0	0	0	0	0
0	1	1	1	0
1	0	0	0	1
0	0	0	0	0
0	0	0	0	0

Κώδικας (επεξήγηση – συν.)

```
byte frownie[8] = {
  0b00000,
  0b01010,
  0b00000,
  0b00000,
  0b01110,
  0b10001,
  0b00000,
  0b00000
};
```

- Πίνακας 8 bytes για τη δημιουργία ενός χαρακτήρα.

Κώδικας (επεξήγηση – συν.)

```
void setup()
{  lcd.begin(16,2);
   lcd.createChar (0, frownie);
   lcd.home ();
   lcd.print("Hello World! (I2C) "); }
```

- Η συνάρτηση `createChar` υπάρχει στην βιβλιοθήκη `LiquidCrystal_I2C` και τη χρησιοποιούμε για να δημιουργήσουμε ένα χαρακτήρα από ένα πίνακα bytes.
- Η συνάρτηση `home` τοποθετεί τον κέρσορα στην αρχή της γραμμής.

Κώδικας (επεξήγηση – συν.)

```
void loop() {
 lcd.setCursor ( 0, 1 );
 lcd.print("SORRY! YOU LOSE ");
 lcd.print (char(0)); }
```

- Η `lcd.print (char(0));` τυπώνει τον χαρακτήρα 0 που δημιουργήσαμε.

? Δημιουργήστε και τυπώστε ένα smiley.

Άσκηση για το σπίτι

1. **Ψάξτε στο διαδίκτυο και βρείτε πέντε (5) ακόμα συσκευές/αισθητήρες που μπορούμε να συνδέσουμε στο Arduino και οι οποίες/οι χρησιμοποιούν τον δίαυλο I2C. Για κάθε μια από αυτές να βρείτε και τη διεύθυνση που συνήθως χρησιμοποιεί στον δίαυλο I2C.**

Και αν έχετε διάθεση….

1. **Τροποποιήστε τον πιο πάνω κώδικα ώστε στην οθόνη να εμφανίζονται στο ίδιο σημείο με τη σειρά οι χαρακτήρες:**

Μεταξύ δυο διαδοχικών χαρακτήρων να περνά 1/10 του δευτερολέπτου περίπου.

Σχολική Μονάδα: 1ο ΕΠΑΛ Ναυπλίου

Συμμετέχοντες Καθηγητές

ΛΙΓΟΜΙΤΗΣ ΓΕΡΑΣΙΜΟΣ
ΡΟΥΜΕΛΙΩΤΗΣ ΠΑΝΤΕΛΗΣ

ΕΠΑΛ: ΝΑΥΠΛΙΟΥ
ΚΑΘΗΓΗΤΗΣ: ΛΙΓΟΜΙΤΗΣ ΓΕΡΑΣΙΜΟΣ

ΣΧΕΔΙΟ ΜΑΘΗΜΑΤΟΣ

ΜΑΘΗΜΑ: ΨΗΦΙΑΚΑ ΣΥΣΤΗΜΑΤΑ
ΕΝΟΤΗΤΑ: ΜΙΚΡΟΕΠΕΞΕΡΓΑΣΤΕΣ
ΤΑΞΗ: Γ΄
ΤΙΤΛΟΣ: Εισαγωγή στους Μικροελεγκτές (Arduino)

Χαρακτηριστικά της ενότητας: **Περιγραφή και λειτουργία των Μικροελεγκτών**

Σκοπός: **Η μελέτη κατασκευή και λειτουργία συστημάτων με τη χρήση Μικροελεγκτών**

Στόχοι: Όταν ολοκληρωθεί αυτή η ενότητα οι μαθητές θα πρέπει να:

- ***Εξηγούν την διαφορά μεταξύ Μικροεπεξεργαστή και Μικρολεγκτή.***
- ***Περιγράφουν τη λειτουργία ενός μικρολεγκτή.***
- ***Αναφέρουν τις λειτουργικές μονάδες ενός μικρολεγκτή.***
- ***Περιγράφουν την εσωτερική του αρχιτεκτονική.***
- ***Αναφέρουν ποια είδη μνήμης χρησιμοποιούν οι Μικροελεγκτές.***
- ***Αποτιμούν την αξία των αυτοματισμών και των αυτόματων συστημάτων.***

Μεθοδολογικές επιλογές:

- ***Μετωπική Διδασκαλεία.***
- ***Συνδιάλεξη.***

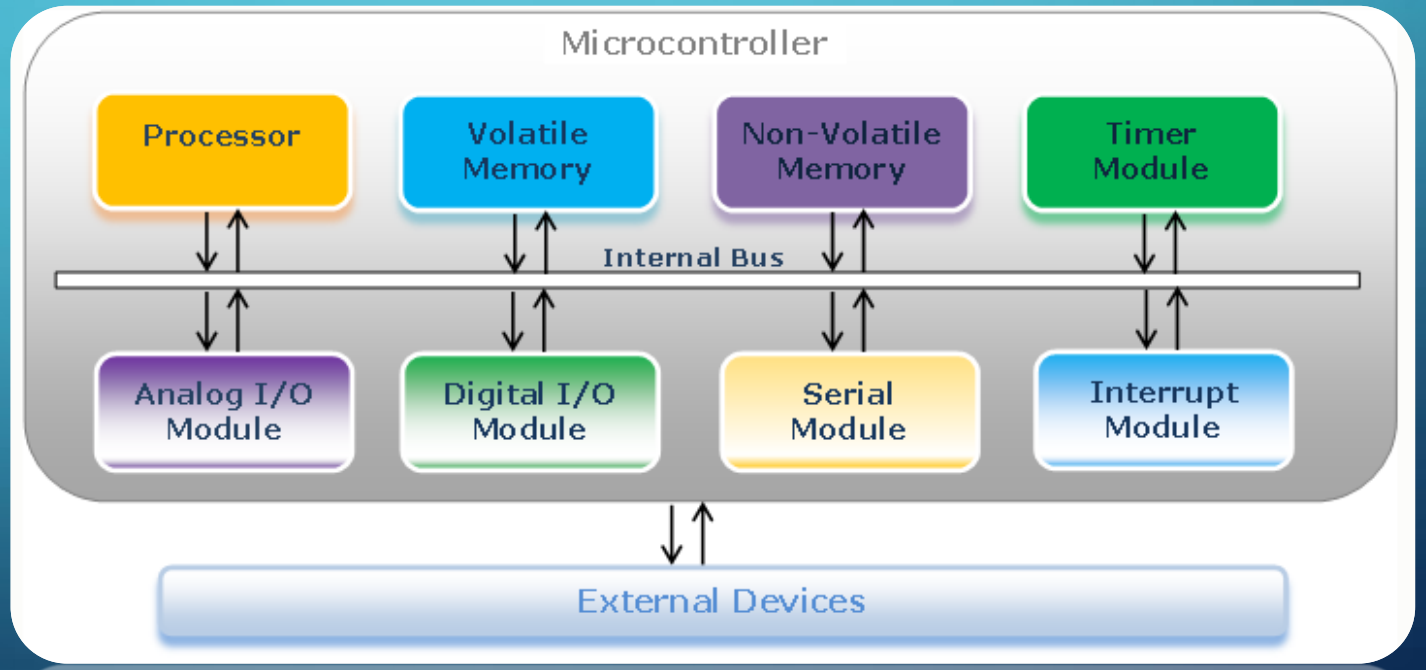
Διδακτικά Υλικά:

- ***Παρουσίαση με Βιντεοπροβολέα.***
- ***Επίδειξη του Αντικειμένου.***

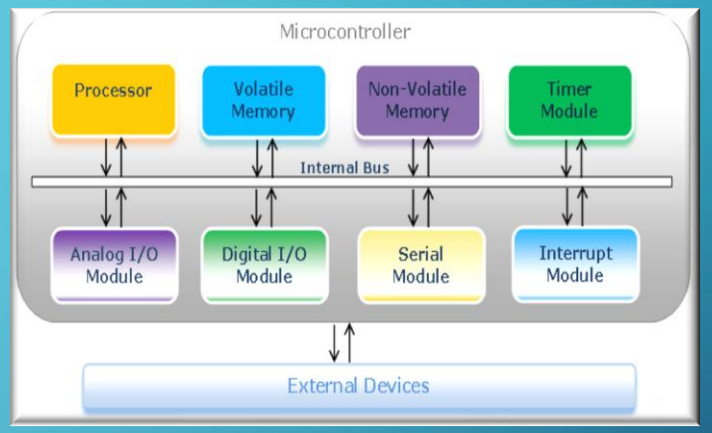
Μικροελεγκτής (Microcontroller)

Είναι ένας υπολογιστής σε μέγεθος Ολοκληρωμένου Κυκλώματος (Chip), που εκτελεί ένα πρόγραμμα σε πραγματικό χρόνο, για την επίτευξη ειδικού σκοπού.

Το πρόγραμμα παράγει μια αλληλουχία σημάτων εισόδου-εξόδου, και αυτή την συνάρτηση καλείται να ελέγχει ο Μικροελεγκτής.



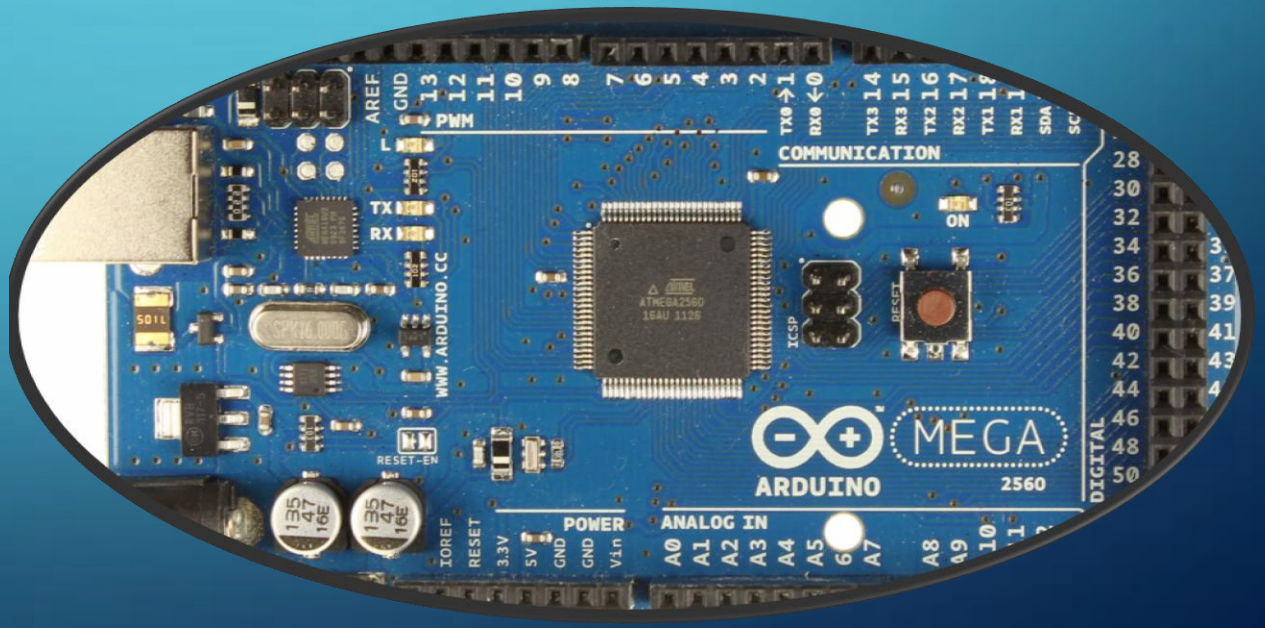
Το Ολοκληρωμένο περιλαμβάνει:



- Εσωτερικό Δίαυλο Επικοινωνίας (Internal Bus)
- Κεντρική Μονάδα Επεξεργασίας (CPU - Processor).
- Μνήμη ROM (Non-Volatile Memory ROM, PROM, EPROM, EEPROM and FLASH)
- Μνήμη RAM (Volatile Memory SRAM and DRAM)
- Ρολόι (Clock).
- Κυκλώματα Χρονισμού (Timer Module).
- Αναλογικές Μονάδες Εισόδου/Εξόδου (Analog I/O Module Interface).
- Ψηφιακές Μονάδες Εισόδου/Εξόδου (Digital I/O Module Interface).
- Μονάδες για Σειριακή Επικοινωνία (Serial Module).
- Σύστημα Διακοπών (Interrupt Module).
- Δυνατότητα Επέκτασης με άλλες Εξωτερικές Συσκευές (External Devices).

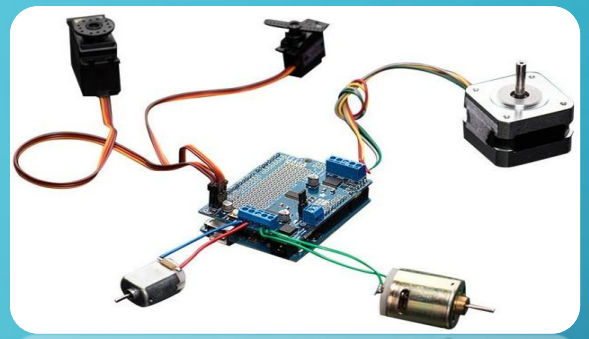
Εφαρμογές Μικροελεγκτών

- Κυκλώματα Επικοινωνιών
- Συστήματα Βιομηχανικών Αυτοματισμών
- Βιοϊατρική Τεχνολογία
- Αυτοκινητοβιομηχανία



Τι είναι το Arduino;

- Είναι ένα open hardware σύστημα
- Σχεδιάστηκε το 2005 στην Ιταλία
- Βασίζεται σε μcs της Atmel
- Βγαίνει σε πολλές παραλλαγές
- Στοχεύει σε μια μεγάλη γκάμα ανθρώπων που δεν έχουν απαραίτητα πολύ εξειδικευμένες γνώσεις



Κατασκευές Με Arduino !!!

- **Με απλά ηλεκτρονικά εξαρτήματα.**
- **Από έτοιμα κυκλώματα του εμπορίου** shields.
- **Ολοκληρωμένες κατασκευές δικής μας έμπνευσης.**

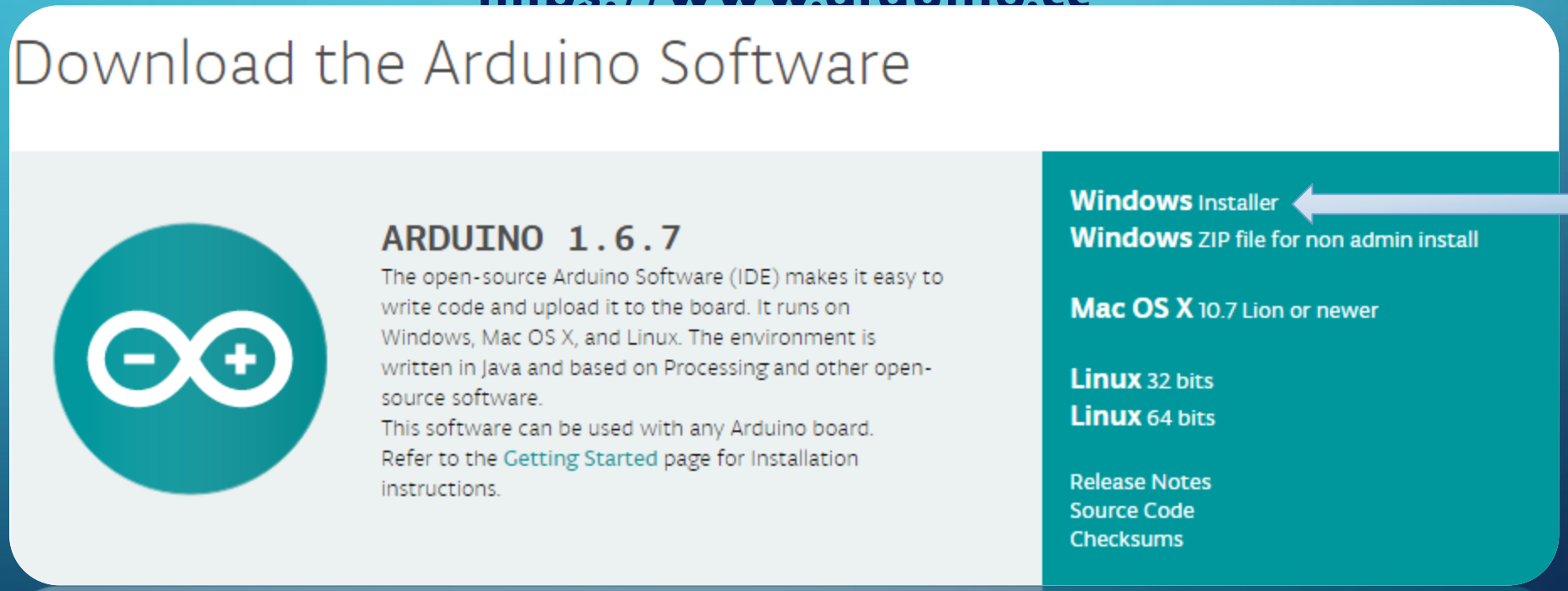
Arduino UNO

- Μικροελεγκτής: ATMEGA328
- Τάση λειτουργίας: 5V
- Τάση εισόδου: 7-12V
- Όρια τάσης εισόδου: 6-20V
- Ψηφιακοί ακροδέκτες I/O: 14, (6 PWM έξοδοι)
- Αναλογικοί ακροδέκτες εισόδου: 6
- IDC ανά pin: 40mA
- IDC για τάσης: 3.3V 50mA
- Μνήμη flash: 32KB (2KB για το BOOTLOADER)
- Μνήμη SRAM: 2KB
- Μνήμη EEPROM: 1KB

Arduino IDE

Εγκατάσταση του Λογισμικού για το Προγραμματισμό του Arduino

https://www.arduino.cc



Arduino IDE

Το πρώτο μου πρόγραμμα Arduino

demo_01 §

```
int WLED = 13; // Δήλωση της Καθολικής Μεταβλητής WLED ως Ακέραιος Αριθμός
               // Και απόδοση της αριθμητικής τιμής 13
void setup() {  // Αρχή της void setup.
                // Ρουτίνα αρχικοποίησης μεταβλητών.
                // Εκτελείται μια και μόνο φορά.
pinMode(WLED,OUTPUT); // Το PIN 13 γίνεται Ψηφιακή Έξοδος.
}               // Τέλος της void setup.

void loop()     // Αρχή της void loop.
                // Η ρουτίνα Loop δεν έχει τέλος,
                // οι εντολές της εκτελούνται συνεχώς.
  digitalWrite(WLED,HIGH);  // Θέσε την Ψηφιακή Έξοδο σε Υψηλό Δυναμικό
                            // Το Led σε λειτουργία
  delay(2000);              // Καθυστέρησε για δύο δευτερόλεπτα
  digitalWrite(WLED,LOW);   // Θέσε την Ψηφιακή Έξοδο σε Χαμηλό Δυναμικό
  delay(2000);              // Καθυστέρησε για δύο δευτερόλεπτα
}
```

https://www.arduino.cc

Σχολική Μονάδα: 1ο ΕΠΑΛ Πύργου

Συμμετέχοντες Καθηγητές

ΚΑΟΥΤΣΗΣ ΒΑΣΙΛΕΙΟΣ
ΚΑΠΡΟΥΛΙΑΣ ΔΗΜΗΤΡΙΟΣ

1ο ΕΠΑΛ ΠΥΡΓΟΥ / ΕΚ ΠΥΡΓΟΥ

Σχέδιο μαθήματος:

Εισαγωγή στην χρήση και προγραμματισμό του arduino board.

Εκπαιδευτικοί

Καούτσης Βασίλης, ηλεκτρονικός

Καπρούλιας Δημήτρης, ηλεκτρολόγος

Περιεχόμενο μαθήματος

Εισαγωγικά στοιχεία.

Το Arduino είναι ένας μικροελεγκτής ο οποίος περιλαμβάνει ένα ολοκληρωμένο κύκλωμα (chip) ATmega της εταιρίας Atmel, το chip αυτό μοιάζει σαν μικροσκοπικός «υπολογιστής» που αποτελείται από CPU, RAM, Flash Memory, εισόδους και εξόδους όπου αντιδρούν ανάλογα με τον προγραμματισμό που κάνουμε και φορτώνουμε στο chip με την βοήθεια του υπολογιστή ή μιας συσκευής android (κινητό ή tablet)..

Το Arduino ξεκίνησε το 2005 στο Ινστιτούτο Διαδραστικής Σχεδίασης (Interaction Design Institute Ivrea(I.D.I.I.)) στην πόλη της Ιβρέα (κωμόπολη της επαρχίας Τορίνο στην βορειοδυτική Ιταλία) όταν ο καθηγητής Massimo Banzi έψαχνε τρόπους να διευκολύνει τους φοιτητές του τμήματος του να δουλέψουν με την τεχνολογία. Μαζί με τον David Cuartielles, μηχανικό από το πανεπιστήμιο του Malmö της Σουηδίας συζήτησαν το πρόβλημα, και αποφάσισαν να φτιάξουν έναν μικροελεγκτή όπου οι σχεδιαστές θα μπορούσαν να συμπεριλάβουν στην δουλειά τους.

Πώς μπορούμε να προγραμματίσουμε το Arduino ;

Μέσω του περιβάλλοντος ανάπτυξής του που ονομάζεται Arduino IDE (σε windows, σε συσκευές android ονομάζεται arduinodroid, για συντομία θα χρησιμοποιούμε γενικά την ονομασία Arduino IDE και για τις δύο πλατφόρμες.Το Arduino χρησιμοποιεί την γλώσσα Wiring, η οποία είναι μια παραλλαγή της γλώσσας C/C++ και είναι αρκετά εύκολη στην σύνταξη της. Όλες οι βιβλιοθήκες της Wiring ειναι υλοποιημένες σε C++.

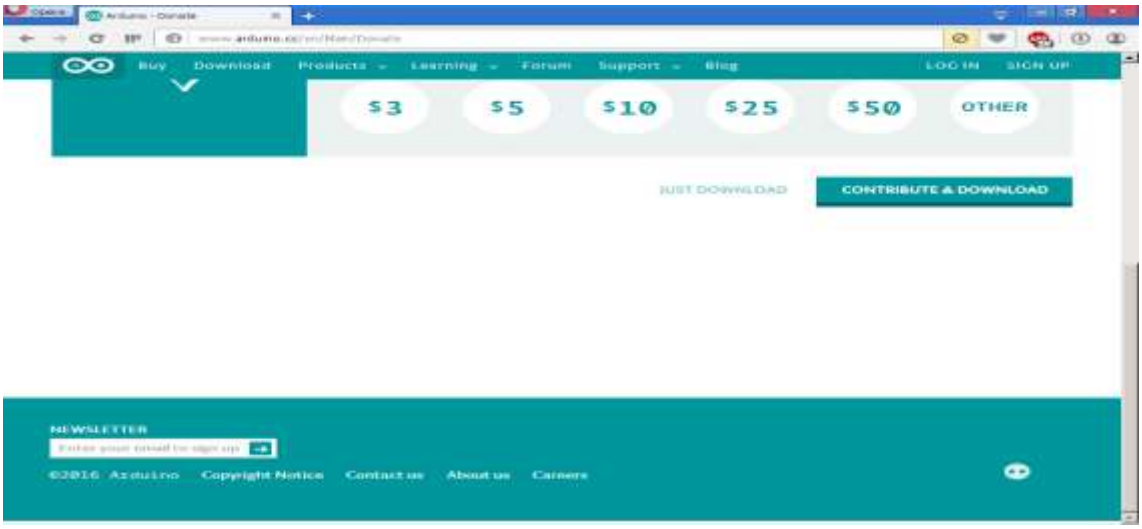
Το υλικό μέρος μιας πλακέτας Arduino αποτελείται απο έναν μικροελεγκτή ATMEL AVR, έναν κρυσταλλικό ταλαντωτή που ειναι χρονισμένος στα 16MHz, έναν σταθεροποιητή τάσης στα 5V, αναλογικές και ψηφιακές εισόδους/εξόδους (Ε/Ε), μια θύρα USB για την σύνδεση με τον υπολογιστή για τον προγραμματισμό του, και διάφορα άλλα εξαρτήματα.

Εγκατάσταση Arduino (Windows)

Το λογισμικό του Arduino που θα χρησιμοποιήσουμε για να προγραμματίσουμε το Arduino είναι διαθέσιμο για Windows. Πηγαίνουμε στη σελίδα:

https://www.arduino.cc/en/Main/Software

και κανουμε κλικ στο Windows zip file

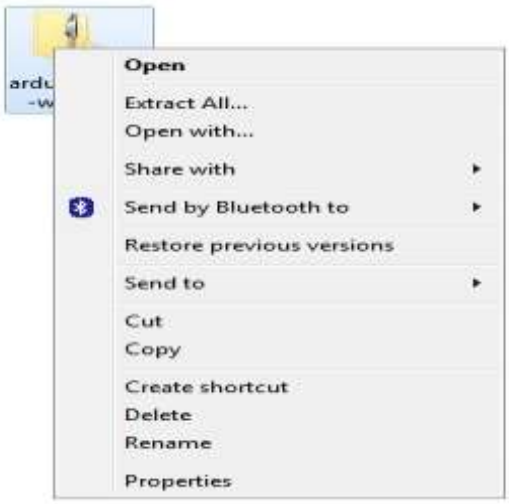


Επιλέγουμε 'just download' ή contribute & download εάν θέλουμε να ενισχύσουμε οικονομικά τους δημιουργούς.

Θα πρέπει να αποσυμπιέσουμε το συμπιεσμένο αρχείο που θα δώσει έναν φάκελο Arduino που περιέχει το πρόγραμμα του Arduino και κάποια άλλα στοιχεία.

Σε ένα ξεχωριστό στάδιο, πρέπει μετά να εγκαταστήσουμε τους USB drivers.

Ξεκινάμε το κατέβασμα του συμπιεσμένου αρχείου για Windows. Υπάρχει μόνο μια έκδοση λογισμικού, ανεξάρτητα από το αν χρησιμοποιούμε Windows XP ή Windows 7, 8 η 10.



Όταν το συμπιεσμένο αρχείο έχει κατέβει, κάνουμε αποσυμπίεση του περιεχομένου στην επιφάνεια εργασίας, με δεξί κλικ στο αρχείο επιλέγοντας ‘Extract all…’ από το μενού.

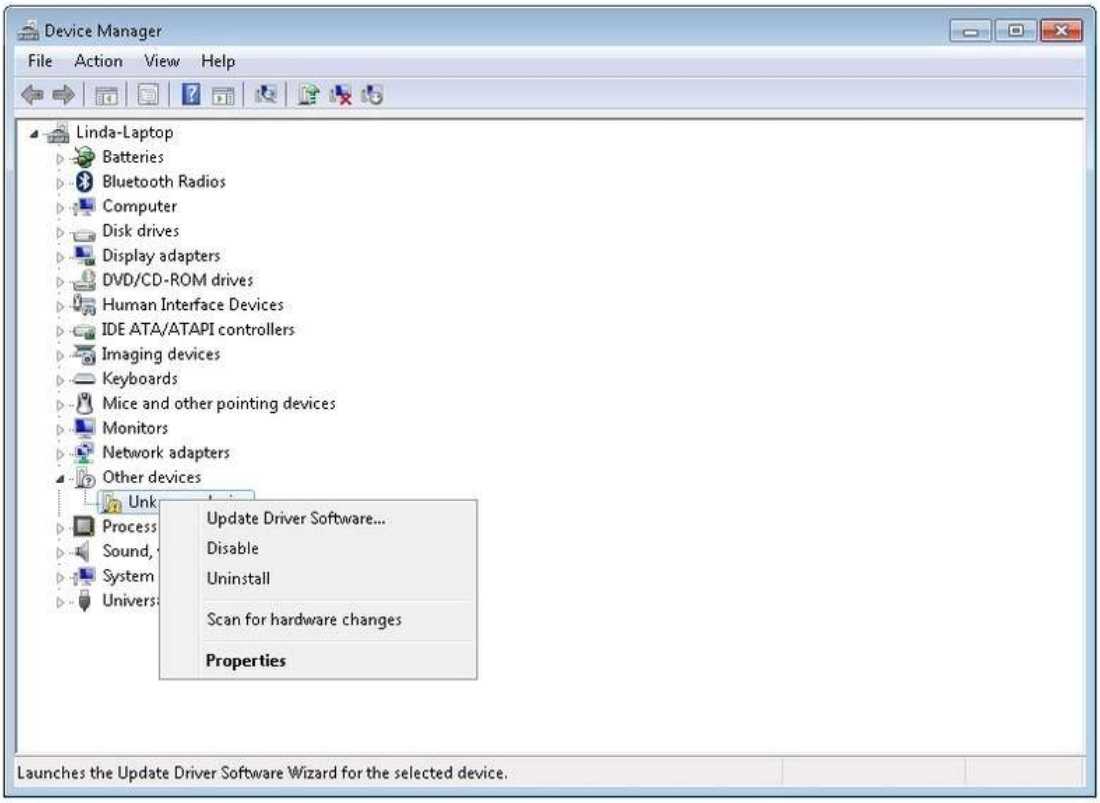
Τώρα επιλέγουμε την επιφάνεια εργασίας και κάνουμε κλικ στο ‘Extract’. Αργότερα μπορούμε να το μετακινήσουμε κάπου αλλού στον υπολογιστή, απλά μετακινώντας τον φάκελο.

Ο φάκελος του Arduino περιέχει το πρόγραμμα του Arduino και επιπλέον τους drivers που αφήνουν τον Arduino να συνδέεται με τον υπολογιστή με ένα καλώδιο USB. Πριν όμως ξεκινήσουμε με το λογισμικό του Arduino, πρέπει να εγκαταστήσουμε τους USB drivers.

Συνδέουμε το ένα άκρο του καλωδίου USB στο Arduino και το άλλο σε μια υποδοχή USB στον υπολογιστή. Η ενδεικτική λυχνία LED στην οθόνη θα ανάψει και μπορεί να πάρουμε ένα μήνυμα από τα Windows ‘Found New Hardware’. Αγνοούμε το μήνυμα

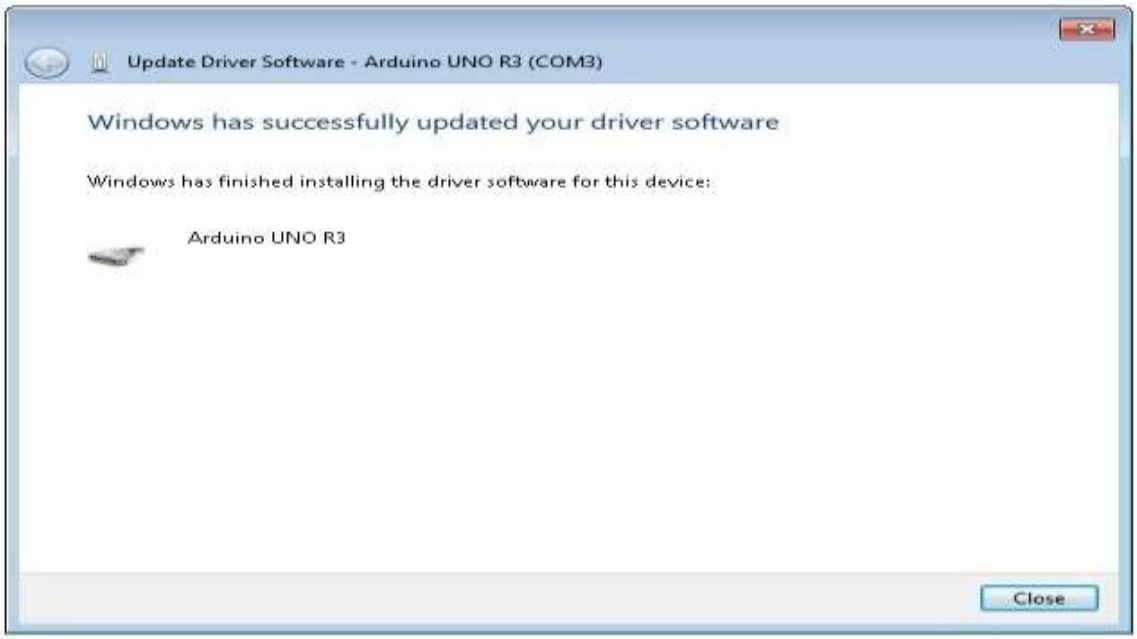
και ακυρώνουμε κάθε προσπάθεια που κάνουν τα Windows για να εγκαταστήσουν τους drivers αυτόματα.

Η πιο αξιόπιστη μέθοδος για την εγκατάσταση των USB drivers είναι να χρησιμοποιήσουμε τη διαχείριση συσκευών. Αυτή είναι προσβάσιμη με διαφορετικούς τρόπους ανάλογα με την έκδοση των Windows. Στα Windows 7, πρέπει πρώτα να ανοίξουμε τον Πίνακα Ελέγχου->Σύστημα και Ασφάλεια->Διαχείριση Συσκευών. Κάτω από το τομέα «Άλλες συσκευές» υπάρχει ένα εικονίδιο «άγνωστη συσκευή» με ένα μικρό κίτρινο τρίγωνο δίπλα του. Αυτό είναι το Arduino.



Δεξί κλικ στη συσκευή και επιλέγουμε από το μενού «ενημέρωση προγράμματος οδήγησης». Από το πλαίσιο που θα ανοίξει επιλέγουμε το «αναζήτηση λογισμικού προγράμματος από τον υπολογιστή μου» και μετά επιλέγουμε το φάκελο arduino-1.0.2-windows\arduino1.0.2\drivers.

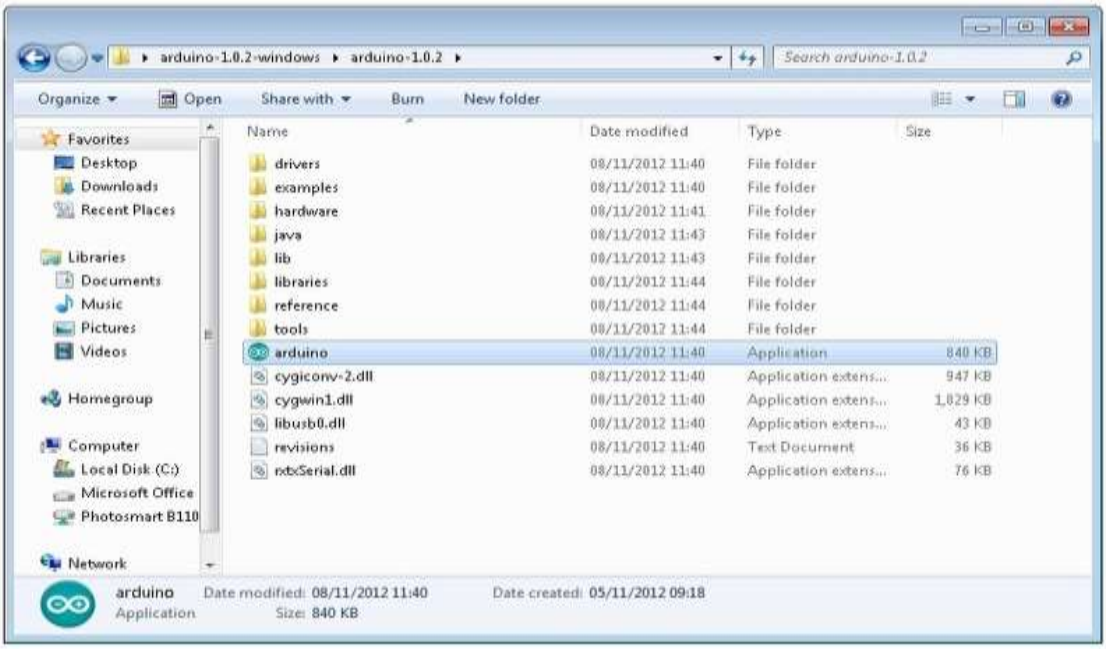
Κάνουμε κλικ στο 'Next' και αν εμφανιστεί στην οθόνη προειδοποίηση ασφαλείας, επιλέγουμε "ναι", αφήνοντας το λογισμικό να εγκατασταθεί. Μόλις το λογισμικό



εγκατασταθεί, θα εμφανιστεί το μήνυμα επιβεβαίωσης:

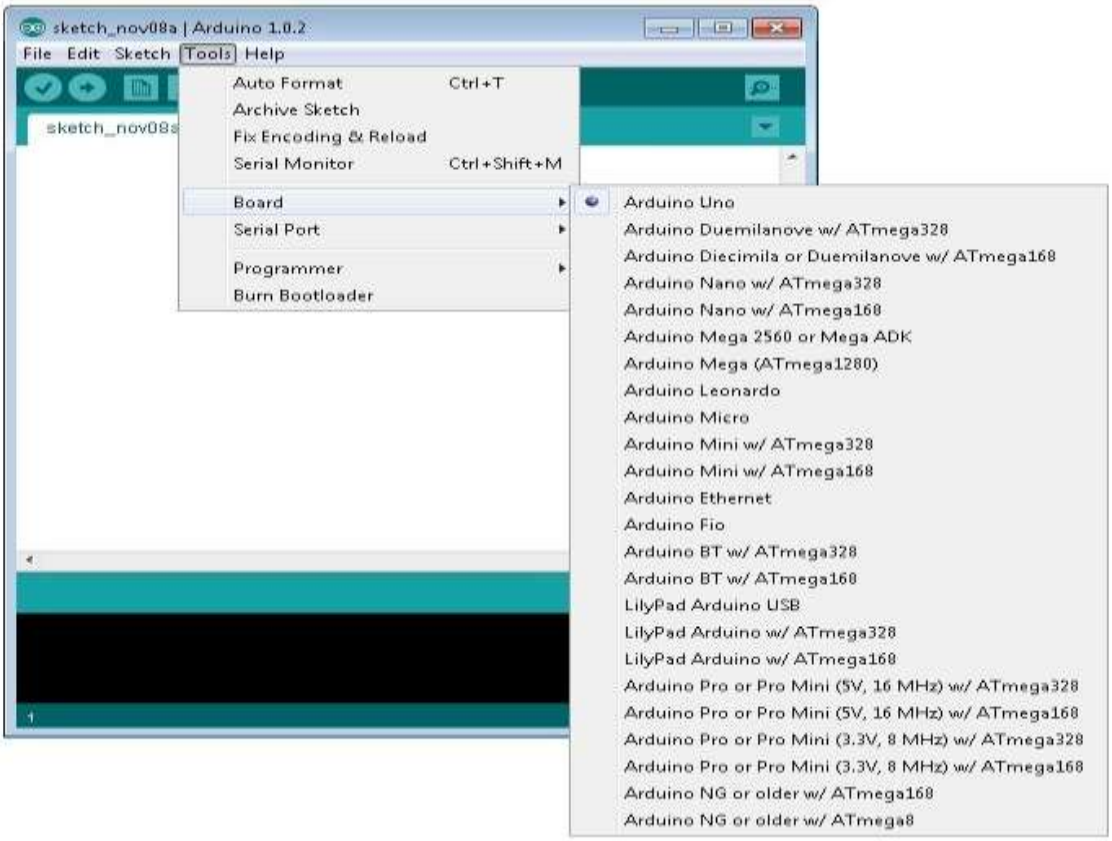
Εκκίνηση του ardouino ide.

Τώρα είμαστε έτοιμοι να ξεκινήσουμε το λογισμικό του Arduino, ανοίγουμε το φάκελο Arduino κα μετά ανοίγουμε την εφαρμογή Arduino που περιέχεται σε αυτόν.

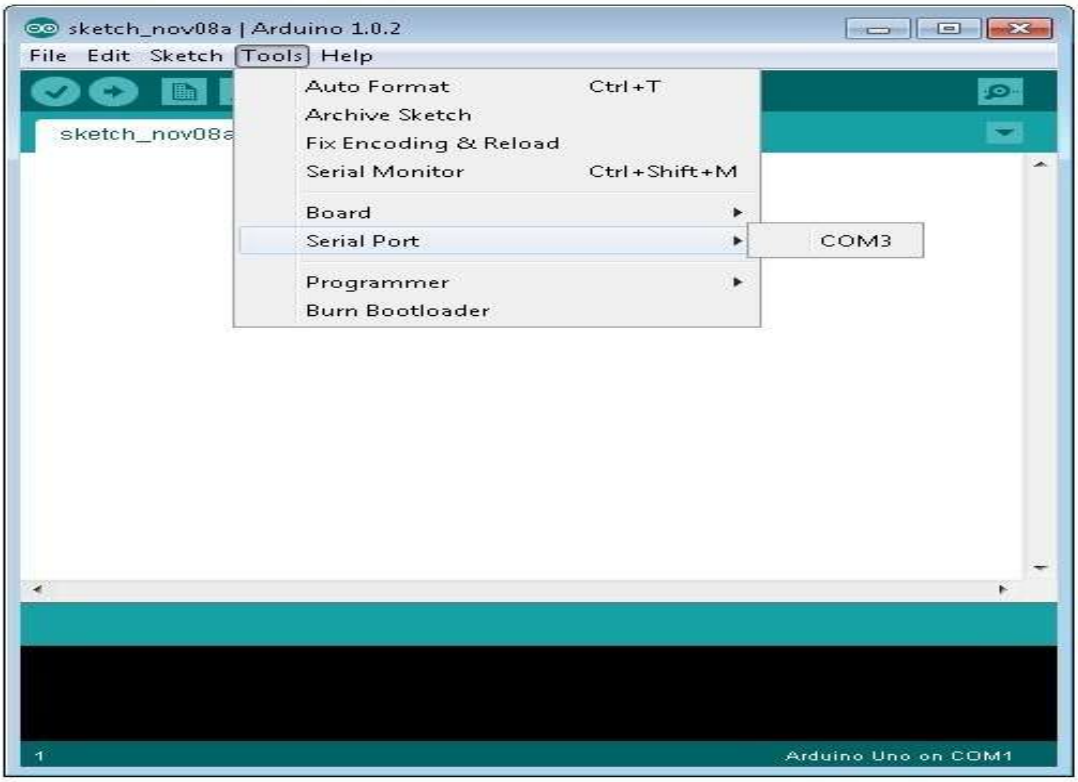


Ξεκινάμε το Arduino IDE, αλλά για να μπορούμε να προγραμματίσουμε πρέπει πριν από αυτό, να πούμε στο Arduino, ποιον τύπο πλακέτας χρησιμοποιούμε και επιπλέον να επιλέξουμε την θύρα που είναι συνδεδεμένο.

Για να πούμε στο Arduino IDE ποιον τύπο πλακέτας χρησιμοποιούμε, από το μενού



εργαλείων, επιλέγουμε Board και μετά ‘Arduino UNO ‘ ή ‘Leonardo’ είτε όποιον άλλο τύπο διαθέτουμε.



Επιπλέον στο μενού εργαλείων , θα βρούμε την ‘Serial Port’ επιλογή.

Θα υπάρχει πιθανότατα μόνο μια επιλογή εδώ και θα λέει COM3 ή COM4. Ακόμα και αν υπάρχει μόνο μια επιλογή, θα πρέπει να επιλεγεί.

Εισαγωγή στον προγραμματισμό του arduino.

Το LED ‘L’

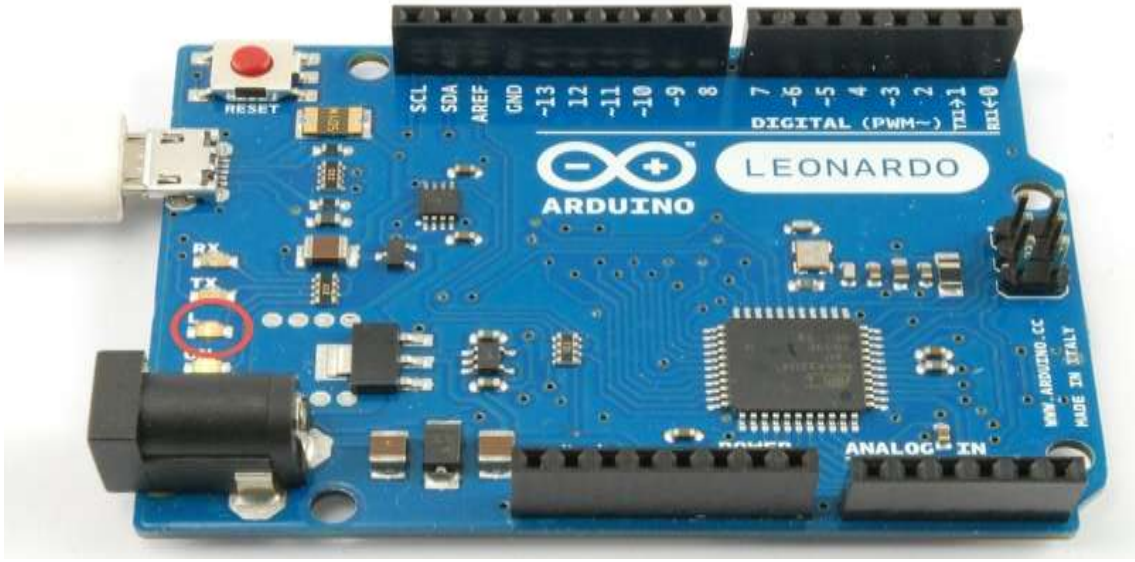
Το Arduino έχει σειρές από υποδοχές κατά μήκος των δύο πλευρών, οι οποίες χρησιμοποιούνται για την σύνδεση με τις ηλεκτρονικές συσκευές που επιτρέπουν στο Arduino να κάνει περισσότερα.

Ωστόσο, το Arduino έχει επίσης ένα μονό LED που μπορούμε να ελέγξουμε από τα σχεδιαγράμματα. Αυτό το LED είναι ενσωματωμένο πάνω στην πλακέτα του Arduino

και συχνά αναφέρεται ως 'L' LED, όπως σημειώνεται πάνω στην πλακέτα.

Η θέση του είναι κυκλωμένη με κόκκινο στην εικόνα του Arduino Uno παρακάτω:

Στο arduino uno:



και στο arduino leonardo:

Φόρτωση του παραδείγματος ‘Blink’

Θα καταλάβουμε ότι το LED “L” στην πλακέτα του Arduino ήδη αναβοσβήνει όταν το συνδέσουμε σε ένα βύσμα USB. Αυτό συμβαίνει γιατί οι πλακέτες Arduino έχουν προεγκατεστημένο τον κώδικα του ‘Blink’.

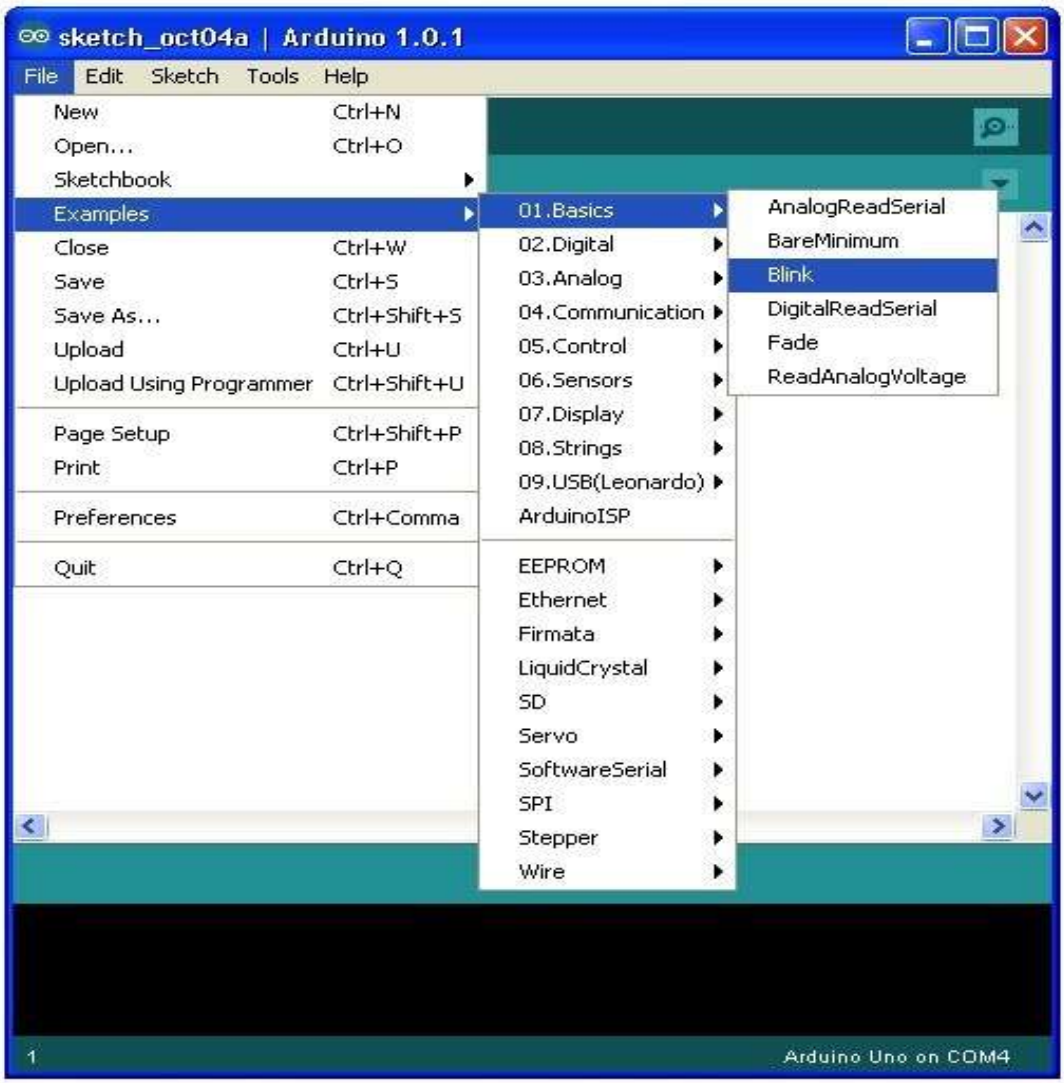
Σε αυτό το μάθημα , θα επαναπρογραμματίσουμε το Arduino με τον δικό μας κώδικα ‘Blink’ και έπειτα θα αλλάξουμε τον ρυθμό με τον οποίο αναβοσβήνει.

Στην αρχή εγκαταστήσαμε το Arduino IDE και φροντίσαμε ώστε να μπορούμε να βρούμε την σωστή σειριακή θύρα του για να το συνδέσουμε στην πλακέτα του Arduino. Τώρα θα αρχίσουμε τον προγραμματισμό του.

Το Arduino IDE περιλαμβάνει μια μεγάλη συλλογή από παραδείγματα κώδικες τα οποία μπορούμε να φορτώσουμε και να χρησιμοποιήσουμε. Περιλαμβάνει επίσης τον κώδικα για να κάνουμε το LED “L” να αναβοσβήνει.

Φορτώνουμε τον κώδικα του 'Blink' που θα βρούμε στο σύστημα μενού του IDE όπως παρακάτω: File → Examples → 01.Basics

Όταν το παράθυρο προγραμματισμού ανοίξει, κάνουμε μεγέθυνση ώστε να μπορούμε να δούμε όλο τον κώδικα στο παράθυρο:



Blink | Arduino 1.0.1

File Edit Sketch Tools Help

Blink

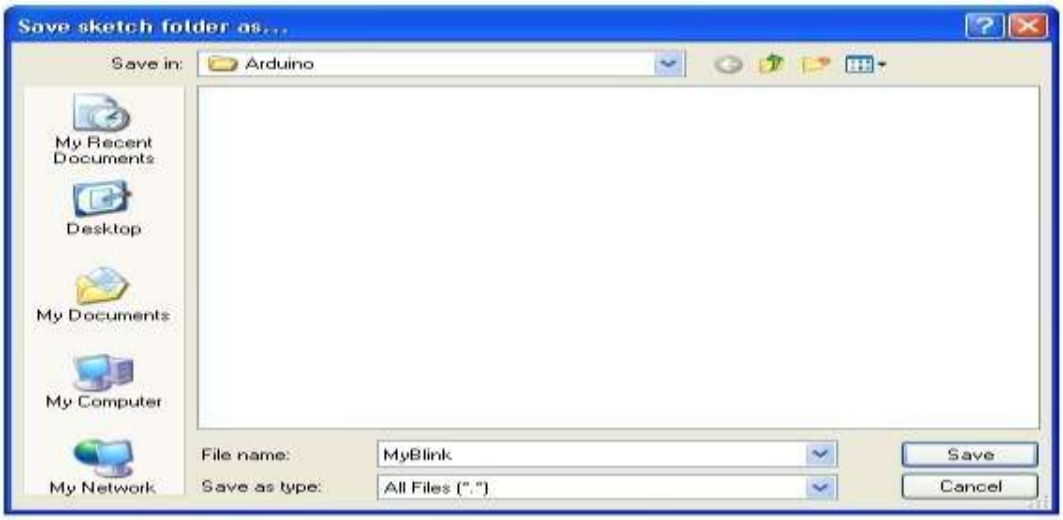
```
/*
  Blink
  Turns on an LED on for one second, then off for one second, repeatedly.

  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
}
```

1 Arduino Uno on COM4

Αποθηκεύοντας ένα αντίγραφο του 'Blink'

Τα παραδείγματα-κώδικες που περιλαμβάνονται στο Arduino IDE είναι 'μόνο για ανάγνωση'. Μπορούμε να τα φορτώσουμε σε μια πλακέτα Arduino, αλλά αν κάνουμε αλλαγές, δεν μπορούμε να τις αποθηκεύσουμε στο ίδιο αρχείο.

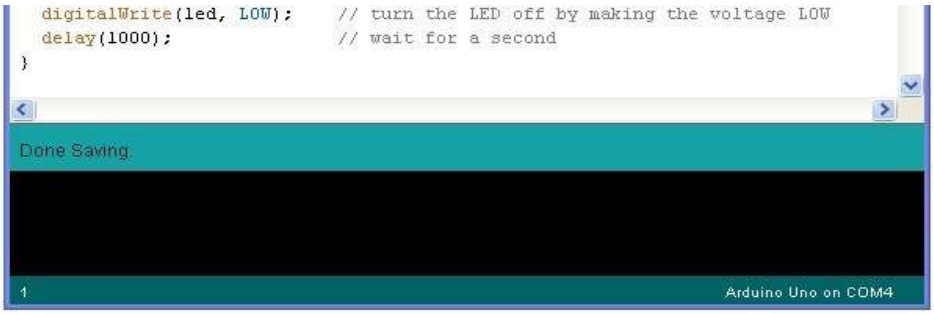
Πρόκειται να αλλάξουμε αυτόν τον κώδικα, ώστε το πρώτο πράγμα που χρειάζεται να



κάνουμε είναι να αποθηκεύσουμε το δικό μας αντίγραφο, για να το αλλάξουμε όποτε θέλουμε.

Από το μενού του File στο Arduino IDE επιλέγουμε 'Save as' και μετά αποθηκεύουμε τον κώδικα με το όνομα 'MyBlink'.

Αποθήκευσαμε το δικό μας αντίγραφο 'Blink' στο sketchbook. Αν κάποτε θελήσουμε να το ξαναβρούμε, θα χρησιμοποιήσουμε την επιλογή στο μενού File → Sketchbook



Μεταφόρτωση Blink στην πλακέτα.

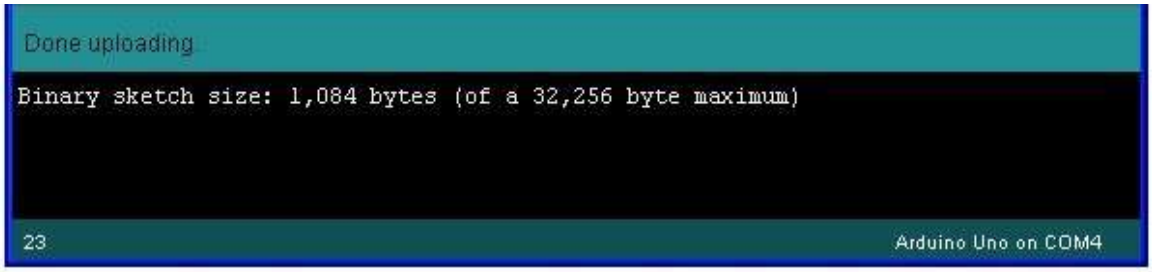
Συνδέουμε την πλακέτα Arduino με τον υπολογιστή με ένα καλώδιο USB και ελέγχουμε ότι το 'Board Type' και το 'Serial Port' έχουν ρυθμιστεί σωστά.

Το Arduino IDE θα δείξει τις τρέχουσες ρυθμίσεις για την πλακέτα στο τέλος του παραθύρου.

Κάνουμε κλικ στο κουμπί 'upload'. Το δεύτερο κουμπί από τα αριστερά στην γραμμή εργαλείων.

Αν κοιτάξουμε την περιοχή κατάστασης του IDE, θα δούμε μια γραμμή προόδου και μια σειρά μηνυμάτων. Στην αρχή θα λέει ''Compiling Sketch..'. Αυτό μετατρέπει τον κώδικα σε μια μορφή κατάλληλη για φόρτωμα στη πλακέτα.

Στη συνέχεια, η κατάσταση θα αλλάξει σε ‘Uploading’. Σε αυτό το σημείο, τα λαμπάκια στο Arduino θα αρχίσουν να τρεμοπαίζουν όσο ο κώδικας μεταφέρεται.



Στο τέλος, η κατάσταση θα αλλάξει σε ‘Done’.

Το άλλο μήνυμα μας λέει ότι ο κώδικας χρησιμοποιεί 1.084 bytes από τα 32.256 bytes διαθέσιμα. Μετά το 'Compiling Sketch..' στάδιο θα μπορούσαμε να πάρουμε το επόμενο μήνυμα λάθους:

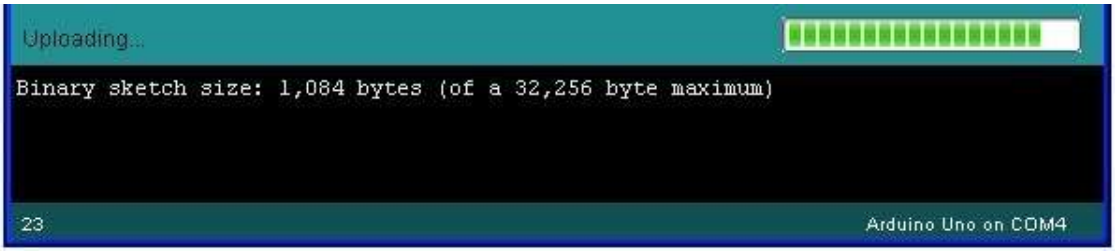
Η ένδειξη είναι στην κορυφή στο κόκκινο πλαίσιο, αυτό πιθανώς σημαίνει ότι η πλακέτα δεν συνδέεται καθόλου, ή οι οδηγοί δεν έχουν εγκατασταθεί σωστά ή ότι έχει επιλεχθεί λάθος σειριακή θύρα.

Αν συμβεί αυτό, ελέγχουμε ξανά την εγκατάσταση.

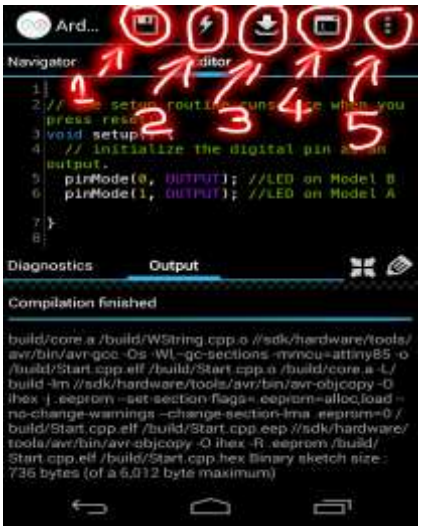
Μόλις η μεταφόρτωση ολοκληρωθεί η πλακέτα θα πρέπει να κάνει επανεκκίνηση και θα αρχίσει να αναβοσβήνει το led.

<u>Σύνδεση και προγραμματισμός arduino</u>

<u>με συσκευές android (κινητά, tablets)</u>



Είναι πλέον δυνατόν να συνδέσουμε και να προγραμματίσουμε το arduino με μία συσκευή android. Όπως φαίνεται και στην φωτογραφία το μόνο επιπλέον εξάρτημα



είναι ένα καλώδιο USB - OTG.

Κατεβάζουμε το πρόγραμμα arduinodroid από το google play store:

http://play.google.com/store/apps/details?id=name.antonsmirnov.android.arduinodroid2

Το πρόγραμμα μοιάζει αρκετά με το αντίστοιχο των windows τουλάχιστον όσο αναφορά τις λειτουργίες, όπως φαίνεται και πιο κάτω :

1) κουμπί για αποθήκευση

2) κουμπί για μεταγλώττιση

3) κουμπί για μεταφορά του

εκτελέσιμου αρχείου στο arduino (upload)

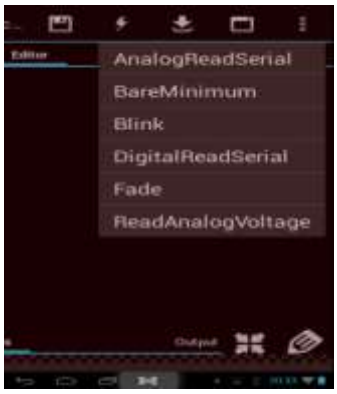
4) κουμπί για εμφάνιση της σειριακής κονσόλας

5) κουμπί μενού για διάφορες λειτουργίες

Πάμε τώρα να δούμε μερικές βασικές λειτουργίες όπως να ανοίξουμε το πρόγραμμα blink, να το μεταγλωττίσουμε και να το μεταφέρουμε στο arduino.

Πρώτα συνδέουμε το arduino με την συσκευή android. Κάθε android συσκευή έχει μια micro usb port. Σε αυτή συνδέουμε το καλωδίο usb-otg και αυτό το συνδέουμε

στο καλώδιο usb που καταλήγει στο arduino.



Πάμε να δούμε πως θα ανοίξουμε το βασικό πρόγραμμα blink. Εκτελούμε το

arduinodroid και πατάμε το κουμπί μενού (στη φωτογραφία με τον αριθμό 5)

Πατάμε Sketch και μετά Examples και 01. Basics

και τέλος επιλέγουμε Blink.

Τώρα είμαστε έτοιμοι να ξεκινήσουμε την μεταγλώττιση. Πατάμε το κουμπί της μεταγλώττισης (στη φωτογραφία το πλήκτρο με την αρίθμηση 2).

Αφού ολοκληρωθεί η διαδικασία της μεταγλώττισης θα παραχθεί το αρχείο blink.cpp.hex και το οποίο θα πρέπει να μεταφερθεί στο arduino.

Για να ξεκινήσει η διαδικασία upload πατάμε το πλήκτρο με την αρίθμηση 3.

Στο παράθυρο που ανοίγει επιλέγουμε την συσκευή σύνδεσης πατώντας πάνω στη λέξη "arduino", και στο επόμενο παράθυρο μας ζητά επιβεβαίωση παραχώρησης δικαιωμάτων πρόσβασης στην συσκευή usb, πατάμε "ok". Η διαδικασία ολοκληρώνεται και το arduino "τρέχει" τώρα το sketch που έχουμε κάνει upload.

Στη διασύνδεση του arduino με συσκευές android πρέπει να προσέξουμε δύο σήμεία: Α) αν και οι συσκευές android μπορούν να τροφοδοτήσουν το arduino μέσω την θύρας micro-usb, η συνεχής πολύωρη χρήση μπορεί να εξαντλήσει την μπαταρία, ειδικά αν πρόκειται για κινητό, πολύ γρήγορα.

Β) Όσοι χρήστες έχουν απενεργοποιήσει στις συσκευές android τη λειτουργία "systemui" για λόγους βελτίωσης της απόδοσης, η διαδικασία "upload" του sketch θα αποτύχει, και ο λόγος είναι ότι μέσα στην λειτουργία "systemui" βρίσκεται ο κώδικας του προγράμματος που παραχωρεί δικαιώματα χρήσης της συσκευής usb. Το πρόβλημα λύνεται εάν η λειτουργία "systemui" ενεργοποιηθεί ξανά (enabled).

Πως δουλεύει το 'Blink'

Εδώ είναι ο κώδικας για το 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 πρώτο πράγμα που σημειώνουμε είναι ότι το μεγαλύτερο μέρος από αυτόν τον κώδικα ονομάζεται 'comments', δηλαδή σχόλια. Τα σχόλια δεν είναι πραγματικές οδηγίες του προγράμματος, είναι απλά σχόλια για το πώς δουλεύει το πρόγραμμα. Είναι εκεί για όφελός μας, έτσι ώστε να υπάρχει κάποια εξήγηση που θα συνοδεύει τον κώδικα.

Οτιδήποτε βρίσκεται ανάμεσα σε /* και */ στην κορυφή του κώδικα είναι ένα σχόλιο για το μπλοκ, που επεξηγεί το θέμα του κώδικα.

Υπάρχουν όμως επίσης σχόλια ανά γραμμή που αρχίζουν με // και οτιδήποτε μέχρι το τέλος της γραμμής, θεωρείται ένα σχόλιο.

Η πρώτη πραγματική γραμμή του κώδικα είναι:

```
int led =13;
```

Όπως εξηγεί το σχόλιο παραπάνω, εδώ δίνεται ένα όνομα στον ακροδέκτη που συνδέεται με το LED. Αυτός είναι ο 13 στα περισσότερα Arduino, συμπεριλαμβανομένων των Uno και Leonardo.

Στη συνέχεια, έχουμε την 'setup' λειτουργία. Ξανά όπως λένε τα σχόλια, αυτό τρέχει όταν το κουμπί reset είναι πιεσμένο. Επίσης τρέχει οποτεδήποτε η πλακέτα κάνει reset για οποιονδήποτε λόγο, όπως μετά την φόρτωση ενός κώδικα.

```
void setup(){

// initialize the digital pin as an output.

 pinMode(led, OUTPUT);

}
```

Κάθε κώδικας Arduino πρέπει να έχει την επιλογή 'setup', και μέσα στο κομμάτι αυτό μπορούμε να προσθέσουμε οδηγίες-εντολές από τον δικό μας κώδικα πάντα ανάμεσα σε { }.

Τώρα στην περίπτωση του παραδείγματος που εξετάζουμε, υπάρχει μια εντολή, η οποία όπως λένε τα σχόλια πρόκειται να χρησιμοποιήσουμε το LED σαν έξοδο.

Είναι επίσης υποχρεωτικό για έναν κώδικα να έχει μια 'loop' λειτουργία. Αντίθετα με την λειτουργία του 'setup' που τρέχει μόνο μια φορά μετά από την επαναφορά, η λειτουργία της 'loop' , αφού έχει τελειώσει με το τρέξιμο των εντολών, αμέσως αρχίζει ξανά.

```
void loop(){

 digitalWrite(led, HIGH);// turn the LED on (HIGH is the voltage level)
```

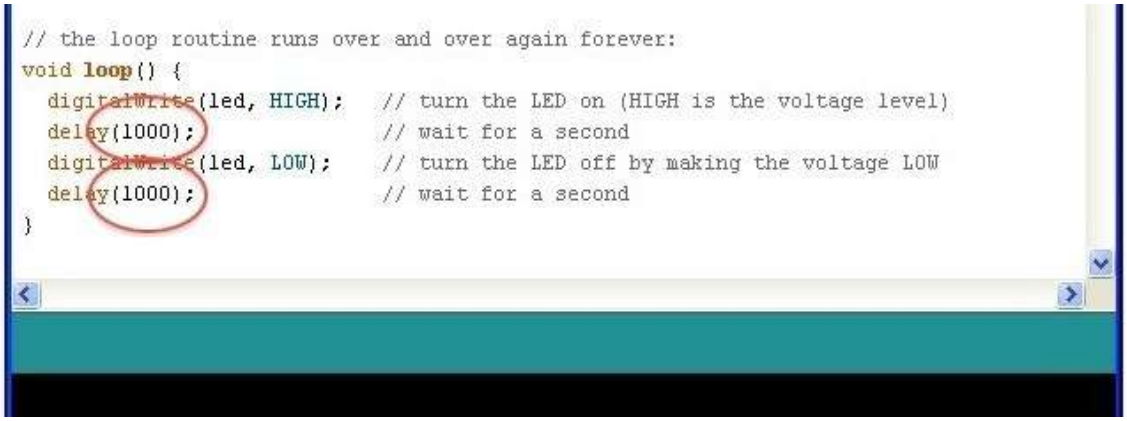
delay(1000);// wait for a second

digitalWrite(led, LOW);// turn the LED off by making the voltage LOW

delay(1000);// wait for a second

}

Μέσα στη loop λειτουργία, οι εντολές πρώτα απ' όλα πάνε το LED pin στην κατάσταση HIGH (ανάβει το λαμπάκι), μετά κάνει 'delay' για 1000 milliseconds (1 second), έπειτα



μετατρέπει το LED pin στην κατάσταση LOW (σφήνει το λαμπάκι) και σταματάει για άλλο 1 second.

Πως το LED αναβοσβήνει γρηγορότερα

Τώρα πρόκειται να κάνουμε το LED να αναβοσβήνει γρηγορότερα. Όπως θα έχετε καταλάβει, η λύση σε αυτό βρίσκεται στη παράμετρο μέσα στην παρένθεση () για την λειτουργία της καθυστέρησης 'delay'.

Αυτή η καθυστέρηση της περιόδου είναι σε milliseconds – χιλιοστά του δευτερολέπτου, έτσι λοιπόν αν θέλουμε το LED να αναβοσβήνει 2 φορές γρηγορότερα, αλλάζουμε την τιμή από 1000 σε 500. Αυτό θα σταματάει για μισό δευτερόλεπτο σε κάθε καθυστέρηση και όχι σε ένα ολόκληρο.

Φορτώνουμε τον κώδικα ξανά και θα δούμε το LED να αρχίζει να λάμπει πιο γρήγορα.

Διδακτικοί στόχοι

Οι μαθητές θα πρέπει:

1) Να γνωρίζουν την προέλευση, τι είναι και ποια μέρη αποτελούν ένα ardouino board.

2) Να μπορούν να κατεβάζουν από το διαδίκτυο το arduino IDE και τους usb drivers.

3) Να μπορούν να εγκαθιστούν στον υπολογιστή και σε μία συσκευή android το αντίστοιχο arduino IDE

4) Να συνδέουν σωστά το arduino με τον υπολογιστή και με συσκευή android μέσω του καλωδίου usb.

5) Να φορτώνουν και να παραλλάσουν το εισαγωγικό πρόγραμμα blink.

Αξιολόγηση της επίτευξης των διδακτικών στόχων

Να γνωρίζουν την προέλευση, τι είναι και ποια μέρη αποτελούν ένα ardouino board	2 γραπτές ερωτήσεις-τεστ: α) ανοιχτού τύπου και β) επιλογής σωστού – λάθους
Να μπορούν να κατεβάζουν από το διαδίκτυο το arduino IDE και τους usb drivers	Θα ζητείται από κάθε μαθητή να πηγαίνει στην ιστιοσελίδα του arduino και να κατεβάζει το arduino IDE και τους usb drivers.
Να μπορούν να εγκαθιστούν στον υπολογιστή και σε συσκευή android το λογισμικό arduino IDE.	Θα ζητείται από κάθε μαθητή να εγκαθιστά στον υπολογιστή και σε συσκεή android το αντίστοιχο λογισμικό arduino IDE
Να συνδέουν σωστά το arduino με τον υπολογιστή και με μία συσκευή android μέσω usb.	Θα ζητείται από κάθε μαθητή να συνδέσει το arduino με τον υπολογιστή και με μία συσκευή android.
Να φορτώνουν και να παραλλάσουν το εισαγωγικό πρόγραμμα.	Θα ζητείται από κάθε μαθητή να φορτώνει το εισαγωγικό πρόγραμμα blink καθώς και μία τουλάχιστον παραλλαγή του.

Γραπτές ερωτήσεις - τέστ

1) ανοικτού τύπου
α) Τι είναι το arduino, τι περιλαμβάνει και από ποιά μέρη αποτελείται;
β) Πως μπορούμε να προγραμματίσουμε το arduino; Ποιά γλώσσα χρησιμοποιούμε για τον προγραμματισμό;

2) επιλογής σωστού - λάθους (κυκλώνουμε το Σ η Λ)
α) Το Arduino είναι ένας μικροελεγκτής ο οποίος περιλαμβάνει ένα ολοκληρωμένο κύκλωμα (chip) ATmega της εταιρίας Atmel.
Σωστό Λάθος

β) Το Arduino ξεκίνησε το 2015 στο Ινστιτούτο Διαδραστικής Σχεδίασης (Interaction Design Institute Ivrea(I.D.I.I.)) στην πόλη της Ιβρέα.
Σωστό Λάθος

Κριτήριο αξιολόγησης

Τα εμπόδια που θα υπάρξουν στην επίτευξη των διδακτικών στόχων, από πλευράς μαθητών και ο τρόπος αντιμετώπιση τους.

Τα βασικά προβλήματα που αναμένεται να υπάρξουν είναι: α) οι μαθητές, στην μεγάλη πλειοψηφία τους, έχουν αρκετά περιορισμένη γνώση της αγγλικής γλώσσας, έτσι οι έννοιες που αντιστοιχούν στους αγγλικούς τεχνικούς όρους παρουσιάζουν αυξημένη δυσκολία κατανόησης. Η αντιμετώπιση του προβλήματος γίνεται με επεξήγηση από την πλευρά του διδάσκοντος σε κρίσιμους όρους του μαθήματος.
β) οι μαθητές έχουν περιορισμένη γνώση προγραμματισμού και πρέπει η μετάβαση σε λειτουργικό κώδικα της C/C++ να γίνει σταδιακά και με προσοχή για να μην χαθεί η ουσία της ικανότητας του arduino να προσφέρει έξυπνες λύσεις αυτοματισμού.
γ) παρατηρείται αυξημένο ενδιαφέρον των μαθητών σε συσκευές android (κινητά, tablets). Αυτό το ενδιαφέρον μπορεί να αξιοποιηθεί για την μάθηση νέων τεχνολογιών και εφαρμογών μικροελεγκτών όπως του arduino.

Ανάπτυξη το σχεδίου μαθήματος και των σταδίων του.

Φάσεις	Διδακτικός Στόχος που επιδιώκεται	Ενέργειες μαθητών	Ενέργειες διδάσκοντος
1.	Να γνωρίσουν την προέλευση, τι είναι και ποια μέρη αποτελούν ένα ardouino board.	Παρακολούθηση της αφήγησης του διδάσκοντος.	Αφήγηση της προέλευσης και της σύστασης του arduino board, επίδειξη ενός arduino uno.
2.	Να μπορούν να κατεβάζουν από το διαδίκτυο το arduino IDE και τους usb drivers.	Επανάληψη της διαδικασίας που παρουσίασε ο διδάσκων από τον κάθε μαθητή.	Επίδειξη της διαδικασίας.
3.	Να μπορούν να εγκαθιστούν στον υπολογιστή και σε συσκευή android, το arduino IDE και τους	Επανάληψη της διαδικασίας που παρουσίσε ο διδάσκων από τον κάθε μαθητή.	Επίδειξη της διαδικασίας.

	usb drivers.		
4.	Να συνδέουν σωστά το arduino με τον υπολογιστή και με μια συσκευή android, μέσω του καλωδίου usb και του προγράμματος arduino IDE.	Επανάληψη της διαδικασίας που παρουσίασε ο διδάσκων από τον κάθε μαθητή.	Επίδειξη της διαδικασίας.
5.	Να φορτώνουν και να παραλλάσουν το εισαγωγικό πρόγραμμα.	Επανάληψη της διαδικασίας που παρουσίασε ο διδάσκων από τον κάθε μαθητή.	Επίδειξη της διαδικασίας.
6.	Ανακεφαλαίωση -επανάληψη	Ερωτήσεις, κρίσεις και σχόλια για την διαδικασία.	Συζήτηση, απαντήσεις σε απορίες.

<u>Πίνακας υλικοτεχνικής υποδομής</u>

Όργανα
Ηλεκτρονικός υπολογιστής
Συσκευή android
Ardouino board
Καλώδιο σύνδεσης usb
Καλώδιο usb-otg

Μετά την αξιολόγηση του αποτελέσματος που επιτεύχθηκε καταγράφονται τα αποτελέσματα και σκεφτόμαστε τι θα πρέπει να αλλάξει.

Αξιολόγηση του σχεδίου μαθήματος

	Διδακτικοί στόχοι	Βαθμός επίτευξης (υψηλός, μέτριος, χαμηλός)
Αξιολόγηση επίτευξης διδακτικών στόχων	1ος Διδακτικός στόχος 2ος Διδακτικός στόχος 3ος Διδακτικός στόχος 4ος Διδακτικός στόχος 5ος Διδακτικός στόχος	
Αντιμετώπιση μαθησιακών εμποδίων (επιτυχής, ανεπιτυχής)	Προτάσεις	
Λειτουργικότητα οπτικοακουστικών και εργαστηριακού εξοπλισμού	Σχόλια	
Άλλες παρατηρήσεις		

Πηγές πληροφόρησης:

- https://el.m.wikipedia.org/wiki/Arduino
- https://www.arduino.cc/en/Guide/HomePage

- https://www.arduino.cc/en/Main/Software

- http://kastoria.teiwm.gr/robotics/?p=308

- http://learn.robotstore.gr/arduino/lesson-0.-getting-started/page-985/

- http://learn.robotstore.gr/arduino/lesson-1.-blink/

- http://m.instructables.com/id/Program-your-Arduino-with-a-Android-device/

- http://www.arduinodroid.info/p/introduction.html