

Φύλλο Εργαστηριακής Άσκησης:

Αισθητήρας Απόστασης και Ενημέρωση σε LCD Display

Όνοματεπώνυμο:

Ημερομηνία:

Ομάδα:

Σύνδεση με τα προηγούμενα

Την προηγούμενη φορά σας αφήσαμε κάποια ελευθερία στην επιλογή των κυκλωμάτων εξόδου που θα υλοποιούσατε (φως, ανεμιστήρας, σκίαστρο), σε σχέση με δυο εισόδους (θερμοκρασία, φωτισμός).

Σήμερα, αυξάνοντας την ελευθερία και την αυτενέργειά σας, θα σας ενθαρρύνουμε να αναζητήσετε μόνοι σας (διαδικτυακά) ιδέες, λύσεις και υλικό για την ολοκλήρωση της σημερινής δραστηριότητας.

Γνωριμία με τα νέα «εξαρτήματα»



Ο **αισθητήρας απόστασης HC-SR04** είναι μια συνηθισμένη λύση αισθητήρα απόστασης που στηρίζεται στην εκπομπή και την λήψη μετά από ανάκλαση υπερήχων 40kHz. Οι προδιαγραφές του αναφέρουν ότι λειτουργεί από 2-400cm. Αυτό μένει να το διερευνήσετε.

Τα 4 pins του αισθητήρα είναι Τάση (Vcc), Γείωση (GND), Πομπός (Trig) και Δέκτης (Echo).



Ένα **LCD Display** είναι μια οθόνη στην οποία έχουμε λίγες γραμμές λίγων χαρακτήρων. Δυο συνηθισμένες λύσεις είναι οθόνη με 2 γραμμές των 16 χαρακτήρων και οθόνη με 4 γραμμές των 20 χαρακτήρων.

Οι 16 ακίδες σύνδεσης και επικοινωνίας, της δικής μας οθόνης θα εξηγηθούν παρακάτω.

Δοκιμή και διερεύνηση λειτουργίας, του αισθητήρα απόστασης

Συνδεσμολογία

Το pin **Vcc** του αισθητήρα στην τάση (5V) και το pin **GND** στην γείωση.

Το pin **Trig** (που στέλνει το ηχητικό σήμα) και το pin **Echo** (που το λαμβάνει) συνδέονται σε δυο ψηφιακά I/O pin του Arduino



(**ΠΡΟΣΟΧΗ**, θυμηθείτε να δηλώσετε το ένα pin ως **input** και το άλλο pin ως **output**)

Εκπομπή ηχητικού παλμού

Η αποστολή του ηχητικού παλμού γίνεται με κώδικα της μορφής που ακολουθεί. Πειραματιστείτε και με άλλη διάρκεια (αντί για 10 μs).

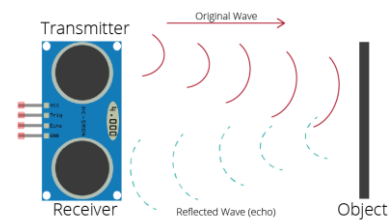
```
DigitalWrite(trigPin, HIGH); // έναρξη εκπομπής ηχητικού παλμού
DelayMicroseconds(10); // διάρκεια 10μs
DigitalWrite(trigPin, LOW); // λήξη εκπομπής ηχητικού παλμού
```

Υπολογισμός απόστασης

Ο υπολογισμός της απόστασης υπολογίζεται με κώδικα της μορφής

```
Duration = pulseIn(EchoPin, HIGH);
Distance_cm = Duration/58.138;
```

Με τη συνάρτηση `pulseIn()`, έχουμε την διάρκεια του ταξιδιού του ήχου, ενώ ο υπολογισμός σχετίζεται με την ταχύτητα του ήχου στον αέρα (343 m/sec [στους 20°C]) και τη συνολική διαδρομή (απόσταση X 2). Δοκιμάστε μικρές αλλαγές στην τιμή 58.138 (π.χ. 58.2).



Προσέξτε: ...

... για να δούμε την απόσταση χρησιμοποιήστε το serial monitor

... 1 microsecond = 1μs = 10⁻⁶ sec = 1/1.000.000 sec

... καλό είναι να αφήσουμε κάποια **milliseconds** πριν εκπέμψουμε ξανά νέο σήμα (η επανάληψη γίνεται διαρκώς μέσα στο void loop())

... σκεφτείτε πως είναι καλό να ξεκινήσετε «μηδενίζοντας» και να στείλετε αρχικά ένα σήμα `DigitalWrite(trigPin, LOW)` για 2 μs

... τα `Duration`, `Distance_cm`, `EchoPin`, `trigPin` είναι μεταβλητές, δηλ. εμείς διαλέξαμε ονόματα για αυτά, θα μπορούσε για τους μαθητές μας να είναι και σε greeklish για να είναι πιο κατανοητό πως δεν είναι δεσμευμένες λέξεις της γλώσσας προγραμματισμού.

Περισσότερες πληροφορίες, θα βρείτε στο help του επίσημου ιστότοπου του Arduino (και αλλού φυσικά). Το πλήρες reference manual, υπάρχει και offline στην e-class (Υλικό Αναφοράς).

Πρόταση: Αναζητήστε διαδικτυακά πληροφορίες, κυκλώματα, λύσεις, προγράμματα και σχετικά projects κατανοήστε τα και εκμεταλλευτείτε τα για την δική σας διερεύνηση.

Παρατηρήσεις και συμπεράσματα από την διερεύνησή σας

Καταγράψτε τα ευρήματά σας από την δοκιμή λειτουργίας, του αισθητήρα απόστασης σε σχέση με αναζήτηση ορίων, περιορισμών, τυχόν ιδιαιτεροτήτων και ό,τι άλλο έχετε εντοπίσει. Τι θα συμβουλευάτε τους μαθητές σας αν τον χρησιμοποιήσουν;



Προέκταση: Γιατί κάποιος προτείνει να τοποθετηθεί ένας αισθητήρας υπερήχων πάνω στον άξονα ενός servo; Τι θα μπορούσε να έχει στο μυαλό του;

Απορία: Τι θα γίνει αν υπάρχει επίπεδο εμπόδιο (τοιχος) μπροστά από τον αισθητήρα με κλίση 45 μοιρών και όχι κάθετα;

Δοκιμή και διερεύνηση λειτουργίας, της LCD οθόνης

Η τεχνολογία της οθόνης είναι αυτή των υγρών κρυστάλλων. **Μην πιέζετε την επιφάνεια της οθόνης**, ακόμη και αν σας φαίνεται χαριτωμένο.

Σημαντικό: Συμβουλευτείτε την τεκμηρίωση του εξαρτήματος που έχετε κάθε στιγμή μπροστά σας, υπάρχει περίπτωση να αλλάζει η διάταξη των ακίδων (pin)

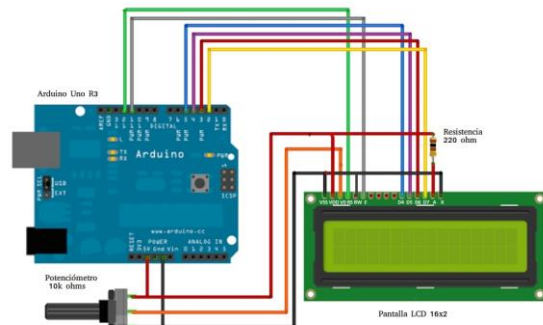
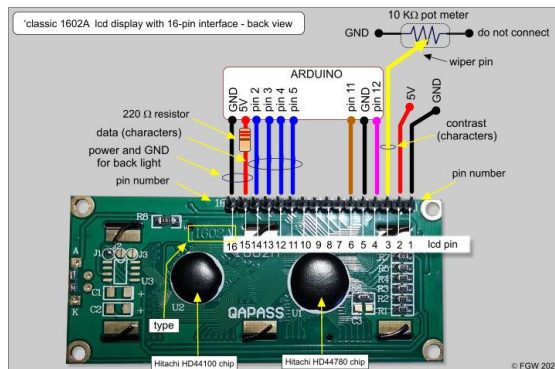
Στην περίπτωση μας ισχύει (επαληθεύτε το όμως):

- Pin 1: GND (0V) για λειτουργία
- Pin 2: Vcc (5V) για λειτουργία
- Pin 3: V0 για ρύθμιση αντίθεσης (έλεγχος με ποτενσιόμετρο)
- Pin 4: RS (επιλογή register, επιλογή μεταξύ εντολών και δεδομένων)
- Pin 5: R/W (επιλογή για εγγραφή ή ανάγνωση για δεδομένα και εντολές)
- Pin 6: E (Enable) συγχρονισμός ανάγνωσης και εγγραφής
- Pin 7- Pin 14: 8-bit για τη μετάδοση των πληροφοριών που θα εμφανίζονται στην οθόνη
- Pin 15: Vcc (5V) για οπίσθιο φωτισμό
- Pin 16: GND (0V) για οπίσθιο φωτισμό

ΣΥΝΔΕΣΜΟΛΟΓΙΑ με Arduino

Δεν θα χρειαστεί να συνδέσουμε όλα τα pin. Εντοπίστε σχετικές εικόνες με τη συνδεσμολογία, παρατηρήστε μικροδιαφορές, κάντε υποθέσεις, δοκιμάστε, γίνετε «σοφότεροι».

Δυο ενδεικτικά σχήματα φαίνεται παρακάτω (αριστερά είναι η πίσω όψη), προσέξτε τη διάταξη των pin.



Φροντίστε να καταλαβαίνετε τι ρόλο παίζει κάθε «καλώδιο» από τα αρκετά που θα χρησιμοποιήσετε. Προσοχή στη συνδεσμολογία, πηγαίνετε βήμα-βήμα.

ΠΡΟΣΟΧΗ στη διάταξη των pin, στα δυο σχήματα που δίνουμε είναι με αντίθετη σειρά (εμπρός-πίσω όψη)

Σχόλια:

Χρησιμοποιούμε 4 pin για μεταφορά δεδομένων, χρησιμοποιείται η τεχνική των 4 bit τη φορά, είναι αρκετό για ότι κάνουμε συνήθως, και θα το βρείτε έτσι στα περισσότερα έτοιμα παραδείγματα.

Με το ποτενσιόμετρο ρυθμίζουμε την αντίθεση (contrast) της οθόνης.

ΚΩΔΙΚΑΣ

Στον προγραμματισμό θα ενσωματώσουμε (include) την κατάλληλη βιβλιοθήκη, LiquidCrystal

Αυτή περιλαμβάνει κάποιες συναρτήσεις (functions). **Αφού πετύχετε μια βασική επικοινωνία**, καλό είναι να διερευνήσετε όσες περισσότερες μπορείτε.

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

Προσέξτε κάποια σημεία του κώδικα: ...

```
#include <LiquidCrystal.h>    // include the library
...
LiquidCrystal lcd(rs, en, d4, d5, d6, d7); // δηλώνετε τα αντίστοιχα pin που έχετε συνδέσει
...
void setup() {
  lcd.begin(16, 2);           // 2 γραμμές, 16 χαρακτήρες
  lcd.print("hello, world!"); // τυπώνω κάτι με την συνάρτηση lcd.print()
  ...
void loop() {
  ...
  lcd.setCursor(0, 1);      // set the cursor to column 0, line 1
  // (note: line 1 is the second row, since counting begins with 0)
  lcd.clear();              // ClearScreen
  ..lcd.write(4);
  ...
  lcd.blink();              // !!??!??
  lcd.noblink();           // !!??!??
  ...
```

Περισσότερες πληροφορίες, θα βρείτε στο help του επίσημου ιστότοπου του Arduino (και αλλού φυσικά). Το πλήρες reference manual, υπάρχει και offline στην e-class (Υλικό Αναφοράς).

Πρόταση: Όπως είναι η βασική ιδέα της σημερινής δραστηριότητας, αναζητήστε διαδικτυακά πληροφορίες, κυκλώματα, λύσεις, προγράμματα και σχετικά projects κατανοήστε τα και εκμεταλλευτείτε τα για την δική σας διερεύνηση.

Παρατηρήσεις και συμπεράσματα από την διερεύνησή σας

Καταγράψτε τα ευρήματά σας από την δοκιμή λειτουργίας, του LCD Display σε σχέση με αναζήτηση ορίων, περιορισμών, τυχόν ιδιαιτεροτήτων και ό,τι άλλο έχετε εντοπίσει. Τι θα συμβουλευάτε τους μαθητές σας αν χρειαστεί να το χρησιμοποιήσουν;

Απορίες: Μπορώ να γράψω ελληνικά; Τι γίνεται όταν στείλω μήνυμα μεγαλύτερο από 16 χαρακτήρες;

Σύνθεση του αισθητήρα απόστασης και της LCD οθόνης

Χρησιμοποιήστε τον αισθητήρα απόστασης προσομοιώνοντας μια πραγματική ανάγκη, που θα μας διατυπώσετε, με ταιριαστή ενημέρωση στην LCD οθόνη. Σχεδιάστε πριν αρχίσετε να υλοποιείτε.



Περιγραφή του «πραγματικού» προβλήματος

ΧΩΡΟΣ ΓΙΑ ΣΗΜΕΙΩΣΕΙΣ ΚΑΙ ΟΤΙ ΑΛΛΟ ΧΡΕΙΑΖΕΤΑΙ ΓΙΑ ΤΗΝ ΔΡΑΣΤΗΡΙΟΤΗΤΑ ΣΑΣ

ΧΩΡΟΣ ΓΙΑ ΣΗΜΕΙΩΣΕΙΣ ΚΑΙ ΟΤΙ ΑΛΛΟ ΧΡΕΙΑΖΕΤΑΙ ΓΙΑ ΤΗΝ ΔΡΑΣΤΗΡΙΟΤΗΤΑ ΣΑΣ