

Αγγελική Μπελεχάκη

## Προγραμματισμός με Arduino

---



ΑΘΗΝΑ 2020

---

## Φύλλα εργασίας για τον προγραμματισμό της πλατφόρμας Arduino

### ΕΙΣΑΓΩΓΗ

Τα φύλλα εργασίας που ακολουθούν, υλοποιούν βασικές διατάξεις κυκλωμάτων πάνω σε πλατφόρμα Arduino και δημιουργήθηκαν με σκοπό την εξοικείωση των μαθητών με την πλατφόρμα και τα υλικά, ώστε να μπορούν να δημιουργήσουν τις δικές τους κατασκευές. Απευθύνεται σε μαθητές της Γ' Γυμνασίου που έχουν αποκτήσει στοιχειώδη εμπειρία στον προγραμματισμό και μπορούν να συνδυάσουν την κατασκευή των κυκλωμάτων τους με τις γνώσεις από τα μαθήματα της Φυσικής, των Μαθηματικών και της Τεχνολογίας.

Οι ασκήσεις παρατίθενται με κλιμακούμενη δυσκολία.

## ΠΕΡΙΕΧΟΜΕΝΑ

<b>ΦΥΛΛΟ ΕΡΓΑΣΙΑΣ 1 - LED που αναβοσβήνει .....</b>	<b>4</b>
<b>ΦΥΛΛΟ ΕΡΓΑΣΙΑΣ 2 - Τρία LED που αναβοσβήνουν .....</b>	<b>7</b>
<b>ΦΥΛΛΟ ΕΡΓΑΣΙΑΣ 3 – Μέτρηση θερμοκρασίας και υγρασίας .....</b>	<b>9</b>
<b>ΦΥΛΛΟ ΕΡΓΑΣΙΑΣ 4 – Συλλογή τιμών έντασης φωτός με φωτοαντίσταση .....</b>	<b>12</b>
<b>ΦΥΛΛΟ ΕΡΓΑΣΙΑΣ 5 – Προγραμματισμός κινητήρα SERVO .....</b>	<b>14</b>
<b>ΦΥΛΛΟ ΕΡΓΑΣΙΑΣ 6 - Ποτενσιόμετρο και σχεδιογράφος processing .....</b>	<b>17</b>
<b>ΦΥΛΛΟ ΕΡΓΑΣΙΑΣ 7 - Κατασκευή ραντάρ με Arduino .....</b>	<b>24</b>
<b>ΦΥΛΛΟ ΕΡΓΑΣΙΑΣ 8 – Πρόγραμμα που κινεί έναν κινητήρα DC .....</b>	<b>29</b>
<b>ΦΥΛΛΟ ΕΡΓΑΣΙΑΣ 9 – Ρύθμιση έντασης LED με αισθητήρα BLUETOOTH .....</b>	<b>33</b>
<b>ΠΑΡΑΡΤΗΜΑ</b>	
Διαχείριση του αισθητήρα Bluetooth από τη σειριακή θύρα του Arduino .....	37
Βιβλιοθήκες .....	37
Σειριακή θύρα και σειριακή οθόνη .....	38
Βιβλιογραφία .....	40

## ΦΥΛΛΟ ΕΡΓΑΣΙΑΣ 1 - LED που αναβοσβήνει

Σε αυτή την άσκηση θα δημιουργήσουμε ένα κύκλωμα με ένα led που θα ανάβει και θα σβήνει ανά ένα δευτερόλεπτο.

**Βαθμός Δυσκολίας:** Εύκολο

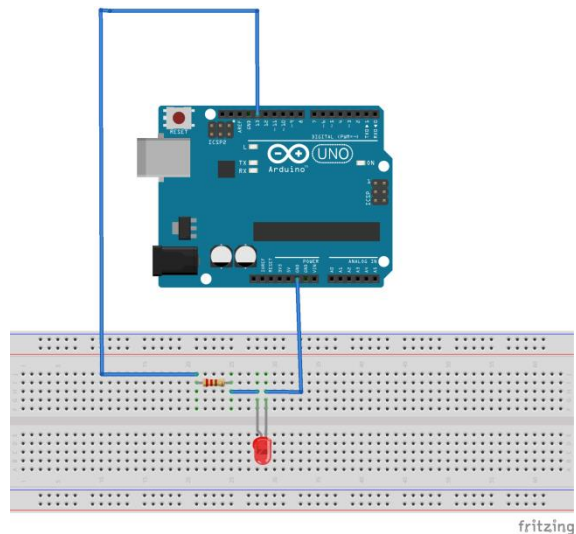
### Υλικά

1. Πλακέτα Arduino συνδεδεμένη στον υπολογιστή
2. Αντίσταση 220 Ω
3. LED
4. 2 jumper καλώδια
5. breadboard

### Περιγραφή

Χρησιμοποιούμε το LED και μια αντίσταση ώστε το ρεύμα που περνάει, να αντιστοιχεί στην τάση λειτουργίας του LED

Το κύκλωμα φαίνεται στην φωτογραφία.



Η είσοδος του ρεύματος είναι στο PIN 13 και η αντίσταση συνδέεται ανάμεσα στην έξοδο και στο LED ώστε να ρυθμίζεται η ποσότητα ρεύματος που περνάει από αυτό. Το LED συνδέεται με το μακρύτερο πόδι στο pin 8 (φορά +). Το κύκλωμα κλείνει με το άλλο πόδι του LED στην υποδοχή GND (φορά -) .

### Αλγόριθμος

1. Δήλωση μεταβλητής που αντιστοιχεί στο PIN ίση με 13
2. Ορίζουμε το PIN 13 ως Έξοδο
3. Δίνουμε τάση 5V στο PIN 13
4. Περιμένουμε 1 δευτερόλεπτο με το LED αναμμένο
5. Διακόπτουμε το ρεύμα στο PIN 13
6. Περιμένουμε 1 δευτερόλεπτο με το LED σβηστό.
7. Τα βήματα 3 έως 6 επαναλαμβάνονται συνεχώς.

### Πρόγραμμα

```
1. int led = 13;
2. void setup() {
3.   pinMode(led, OUTPUT);
4. }
5. void loop() {
6.   digitalWrite(led, HIGH);
7.   delay(1000);
8.   digitalWrite(led, LOW);
9.   delay(1000);
10. }
```

### Οδηγίες

Στο φάκελο Arduino της Επιφάνειας εργασίας του υπολογιστή σας θα βρείτε το αρχείο led.ino. Κάνοντας διπλό κλικ πάνω του το αρχείο θα ανοίξει μέσα από το προγραμματιστικό περιβάλλον του Arduino. Μεταγλωττίστε και ανεβάστε το πρόγραμμα στην πλακέτα του Arduino αφού τη συνδέσετε στον υπολογιστή σας μέσω της USB θύρας.

Για τη μεταγλώττιση επιλέξτε το Μενού: Σχέδιο > Επικύρωση / Μεταγλώττιση και για το ανέβασμα του εκτελέσιμου κώδικα από το Μενού: Σχέδιο > Ανέβασμα.

### Ερωτήσεις

1. Περιγράψτε τι κάνει το πρόγραμμα;
2. Σβήστε το ερωτηματικό από την εντολή digitalWrite(LED, HIGH); Μεταγλωττίστε και τρέξτε το πρόγραμμα. Τι παρατηρείτε; Τι μήνυμα εμφανίζεται;  
.....  
.....  
  
Τι είδους λάθος δημιουργήσατε; .....  
Διορθώστε το λάθος σας, εκτελέστε το πρόγραμμα.
3. Παρατηρήστε το Arduino και το Breadboard. Συμβουλευτείτε και τον κώδικα του προγράμματος. Σε ποιον αποδέκτη (pin) είναι συνδεδεμένο το LED; .....
4. Ποια είναι η εντολή με την οποία ανάβει το LED;  
.....
5. Με ποια εντολή σβήνει το LED; .....
6. Αλλάξτε την εντολή digitalWrite(LED, HIGH); με την digitalWrite(13, HIGH); και μεταγλωττίστε το πρόγραμμα. Τι συμβαίνει; Δικαιολογήστε την απάντησή σας.  
.....
7. Αλλάξτε το καλώδιο που συνδέει το LED από pin 13 στο pin 10. Τροποποιήστε το πρόγραμμα ώστε το LED να αναβοσβήνει.

8. Η εντολή `delay(1000);` δημιουργεί μια καθυστέρηση 1000 msec = ..... sec. Κάντε τις κατάλληλες τροποποιήσεις στο πρόγραμμα έτσι ώστε το LED να ανάβει για 2 sec και να σβήνει για 3 sec. Δοκιμάστε το πρόγραμμα.

## ΦΥΛΛΟ ΕΡΓΑΣΙΑΣ 2 - Τρία LED που αναβοσβήνουν

Σε αυτήν την άσκηση θα συνδέσουμε τρία led στο Arduino τα οποία θα αναβοσβήνουν ανά 3 δευτερόλεπτα.

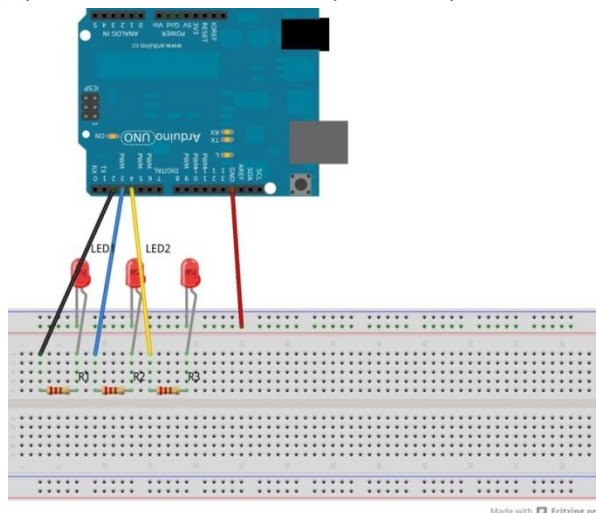
**Βαθμός δυσκολίας:** Εύκολο

### Υλικά

1. Πλακέτα Arduino συνδεδεμένη στον υπολογιστή
2. τρεις αντιστάσεις 220 Ω
3. τρία LED
4. 6 jumper καλώδια
5. breadboard

### Περιγραφή

Στο κύκλωμα του Φύλλου Εργασίας 1 προσθέστε δύο ακόμη led όπως φαίνεται στο πιο κάτω σχήμα. Τα leds είναι συνδεδεμένα στα pins 2, 3 και 4.



### Αλγόριθμος

1. Δίνουμε τάση 5V στο PIN 2
2. Περιμένουμε 3 δευτερόλεπτα με το LED αναμμένο
3. Διακόπτουμε το ρεύμα στο PIN 2
4. Επαναλαμβάνουμε τα βήματα 1 έως 3 για το pin 3 και μετά για το pin 4 και στη συνέχεια όλα τα πιο πάνω βήματα επαναλαμβάνονται συνεχώς .

### Πρόγραμμα

Προσθέστε στο πρόγραμμα τις κατάλληλες εντολές ώστε τα τρία led να ανάβουν διαδοχικά ανά τρία δευτερόλεπτα.

### Υπόδειξη

1. `Void loop() {`
2. `digitalWrite(2, HIGH);`
3. `delay(3000);`
4. `digitalWrite(2, LOW);`
5. `digitalWrite(3, HIGH);`
6. `delay(3000);`
7. `digitalWrite(...);`
8. `digitalWrite(4, HIGH);`
9. `delay(3000);`
10. `digitalWrite(...);`
11. `}`

### Οδηγίες

1. Μεταγλωττίστε το πρόγραμμα και ανεβάστε το στην πλακέτα Arduino,
2. Τροποποιείστε το πρόγραμμα ώστε το πρώτο LED να ανάβει για 3 δευτερόλεπτα, μετά να σβήνει, μετά το δεύτερο να ανάβει για 2 δευτερόλεπτα και μετά να σβήνει για 1 δευτερόλεπτο και στη συνέχεια να ανάβουν και τα τρία για 3 δευτερόλεπτα και μετά να σβήνουν.
3. Αποθηκεύστε το πρόγραμμά σας με το όνομα `flash_led.ino`.

### Ερωτήσεις

1. Περιγράψτε τι κάνει το πρόγραμμα  
.....  
.....  
.....
2. Κάντε τις κατάλληλες αλλαγές στο πρόγραμμά σας ώστε το πρώτο LED να ανάβει για 5 δευτερόλεπτα, το δεύτερο για 3 και το τρίτο για 1.
3. Δοκιμάστε το πρόγραμμά σας.



## ΦΥΛΛΟ ΕΡΓΑΣΙΑΣ 3 – Μέτρηση θερμοκρασίας και υγρασίας

Σε αυτήν την άσκηση θα χρησιμοποιήσουμε έναν αισθητήρα θερμοκρασίας και υγρασίας.

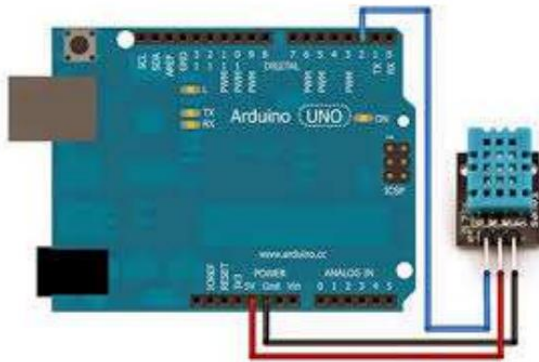
**Βαθμός δυσκολίας:** Εύκολο

### Υλικά

1. Πλακέτα Arduino συνδεδεμένη στον υπολογιστή
2. Αισθητήρας θερμοκρασίας και υγρασίας τύπου DHT11
3. Breadboard (μπορεί να μη χρησιμοποιηθεί και ο αισθητήρας να συνδεθεί απευθείας με την πλακέτα όπως φαίνεται στο σχήμα)

### Περιγραφή

Ο αισθητήρας συνδέεται στο κύκλωμα όπως φαίνεται στο πιο κάτω σχήμα. Για να λειτουργήσει ο αισθητήρας πρέπει να χρησιμοποιηθούν εντολές που βρίσκονται σε μια βιβλιοθήκη (DHT), η οποία πρέπει να ενσωματωθεί στο πρόγραμμα.



### Σύνδεση με το Arduino

Μπλέ καλώδιο (out): Pin

Μαύρο καλώδιο (-): Ground

Κόκκινο καλώδιο (+): 5V

### Αλγόριθμος

1. Διαβάζουμε την τιμή από τον αισθητήρα
2. Τυπώνουμε στην σειριακή οθόνη την τιμή της υγρασίας
3. Τυπώνουμε στην σειριακή οθόνη την τιμή της θερμοκρασίας
4. Περιμένουμε 5 δευτερόλεπτα
5. Επαναλαμβάνουμε τα βήματα 1 έως 4

### Βιβλιοθήκη

Για να λειτουργήσει ο αισθητήρας με το Arduino πρέπει να ενσωματωθεί η βιβλιοθήκη DHT στο πρόγραμμα όπως αναφέρεται στην εισαγωγή.

## Εντολές

**DHT.read11(dht\_apin);** Διαβάζεται η τιμή που αποδίδει ο αισθητήρας

**Serial.print(DHT.temperature);** Τυπώνεται η τιμή της θερμοκρασίας στο παράθυρο της σειριακής

**Serial.print(DHT.humidity);** Τυπώνεται η τιμή της υγρασίας στο παράθυρο της σειριακής

## Πρόγραμμα

```
1. #include "dht.h"
2. dht DHT
3. void setup() {
4.   Serial.begin(9600);
5.   delay(500);
6.   Serial.println("DHT11 Αισθητήρας Θερμοκρασίας και υγρασίας\n");
7.   delay(1000);
8. }
9. void loop() {
10.  DHT.read11(dht_apin); //διαβάζεται η τιμή από τον αισθητήρα
11.  Serial.print("Υγρασία= ")
12.  Serial.print(DHT.humidity); //τυπώνει την τιμή της υγρασίας
13.  Serial.println("%");
14.  Serial.print("Θερμοκρασία= ")
15.  Serial.print(DHT.temperature); //τυπώνει την τιμή της θερμοκρασίας
16.  Serial.println("°C");
17.  delay(5000); // Αναμονή πέντε δευτερολέπτων για νέα πρόσβαση στον
    αισθητήρα
18. }
```

## ΟΔΗΓΙΕΣ

1. Ανοίξετε το temp\_hum.ino, μεταγλωττίστε το και ανεβάστε τον εκτελέσιμο κώδικα στο Arduino
2. Το πρόγραμμα διαβάζει και εμφανίζει τις τιμές που δίνει η φωτοαντίσταση, Για να δείτε τις τιμές αυτές επιλέξτε από το Μενού: Εργαλεία > Παρακολούθηση Σειριακής.

## ΕΡΩΤΗΣΕΙΣ

1. Με ποια εντολή διαβάζουμε τις τιμές που δίνει ο αισθητήρας;  
.....
2. Ποια μεταβλητή μας δίνει τις τιμές της υγρασίας;  
.....
3. Ποια μεταβλητή μας δίνει τις τιμές της θερμοκρασίας  
.....

4. Με ποια εντολή μπορούμε να δούμε την τιμή της θερμοκρασίας στη παράλληλη οθόνη

## ΦΥΛΛΟ ΕΡΓΑΣΙΑΣ 4 – Συλλογή τιμών έντασης φωτός με φωτοαντίσταση

Σε αυτό το μάθημα θα χρησιμοποιήσουμε μια φωτοαντίσταση (LDR – Light Dependent Resistor) που θα χρησιμοποιηθεί σαν αισθητήρας φωτός.

**Βαθμός δυσκολίας:** Εύκολο

### Εφαρμογές

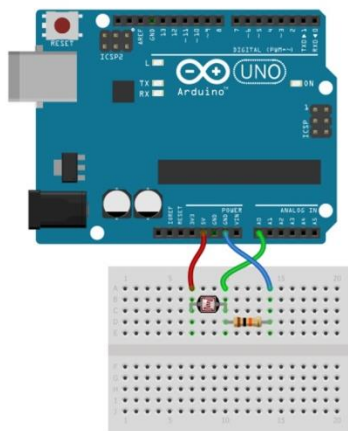
Η φωτοαντίσταση χρησιμοποιείται στα ηλεκτρονικά κυκλώματα φωτισμού σε περιπτώσεις που θέλουμε να ελέγξουμε το φωτισμό σε δρόμους ή σε ένα δωμάτιο.

Χρησιμοποιείται ακόμα σε αυτοματισμούς, συστήματα συναγερμού, πόρτες που ανοίγουν αυτόματα κ.λπ.

### Περιγραφή

Ο αισθητήρας φωτός (φωτοαντίσταση) λειτουργεί ως εξής: ανάλογα με την ένταση του φωτός που πέφτει πάνω του, περνάει ανάλογη ποσότητα ρεύματος από το κύκλωμα. Διαβάζοντας την ένδειξη που δίνει το pin πάνω στο οποίο βρίσκεται η φωτοαντίσταση όταν η τιμή είναι υψηλή, ο LDR δέχεται πολύ φως και χαμηλή όταν δεν υπάρχει φως.

Συνδέστε στο κύκλωμα του Arduino τη φωτοαντίσταση όπως φαίνεται στο πιο κάτω σχήμα. Η θύρα που συνδέεται η φωτοαντίσταση πρέπει να είναι αναλογική (εδώ είναι στο A0).



### Υλικά

1. Πλακέτα Arduino συνδεδεμένη στον υπολογιστή
2. Μία αντίσταση 220 Ω
3. Τρία jumper καλώδια
4. breadboard

### Αλγόριθμος

1. Διαβάζουμε την τιμή της έντασης του φωτός από τη φωτοαντίσταση
2. Αποθηκεύουμε την τιμή στη μεταβλητή R1
3. Τυπώνουμε την τιμή της R1 στην σειριακή οθόνη

4. Περιμένουμε 2 δευτερόλεπτα
5. Επαναλαμβάνουμε τα βήματα 1 έως 4

### Πρόγραμμα

```

1. int LDR1 = A0;      //συνδέεται η LDR1 στο Pin A0
2. void setup() {
3.   Serial.begin(9600);
4.   pinMode(LDR, INPUT);
5.   delay(2000);      }
6. void loop() {
7.   int R1 = analogRead(LDR); // Διαβάζει την τιμή της φωτοαντίστασης με όνομα LDR 1
8.   Serial.print("R1 = ");
9.   Serial.println(R1);
10.  delay(2000);
11. }
    
```

### Οδηγίες

1. Ανοίξτε το αρχείο ldr.ino
2. Μεταγλωττίστε το και ανεβάστε τον εκτελέσιμο κώδικα στο Arduino
3. Το πρόγραμμα διαβάζει και εμφανίζει τις τιμές που δίνει η φωτοαντίσταση, Για να δείτε τις τιμές αυτές επιλέξτε από το Μενού: Εργαλεία > Παρακολούθηση Σειριακής.

### Ερωτήσεις

1. Περιγράψτε τη λειτουργία της εντολής analogRead(LDR1)
2. Ποια η λειτουργία της εντολής pinMode(OUTPUT);
3. Ποιος ο ρόλος της μεταβλητής R1;
4. Τι τιμές δίνει ο αισθητήρας:

Στο φως;	
Όταν σκιάσετε λίγο;	
Όταν σκιάσετε αρκετά;	
Στο σκοτάδι;	

## ΦΥΛΛΟ ΕΡΓΑΣΙΑΣ 5 – Προγραμματισμός κινητήρα SERVO

**Βαθμός Δυσκολίας:** Μέτριας δύσκολο

### Περιγραφή κυκλώματος

Οι κινητήρες servo είναι μικρές συσκευές με έναν εξωτερικό άξονα. Αυτός ο άξονας περιστρέφεται σε διάφορες θέσεις όταν αποσταλεί στον servo ένα σήμα. Όσο υπάρχει το σήμα στην γραμμή εισόδου του Servo, αυτός θα διατηρεί τον άξονα του σε μια συγκεκριμένη θέση. Όταν αλλάζει το σήμα προκαλεί στο servo να μεταβάλλει την γωνία του άξονα.

**Εφαρμογές:** τα servo χρησιμοποιούνται σε τηλεχειριζόμενα αεροπλάνα, αυτοκίνητα και στην ρομποτική. Οι κινητήρες αυτοί έχουν ένα εξαιρετικά μικρό μέγεθος αλλά είναι αρκετά ισχυροί για το μέγεθος τους. Αυτό τους κάνει πολύ εύχρηστους.

Σε αυτήν την άσκηση, ο σερβοκινητήρας γυρίζει 15 μοίρες, 30 μοίρες, 45 μοίρες, 60 μοίρες, 75 μοίρες, 90 μοίρες, 75 μοίρες, 60 μοίρες, 45 μοίρες, 30 μοίρες, 15 μοίρες, 0 μοίρες και μετά 0 μοίρες έως 180 μοίρες και από 180 μοίρες σε 0 μοίρες.

### Σύνδεση με το Arduino

- Κόκκινο καλώδιο: 5V
- Πορτοκαλί: pin 9
- Μαύρο: ground

### Αλγόριθμος

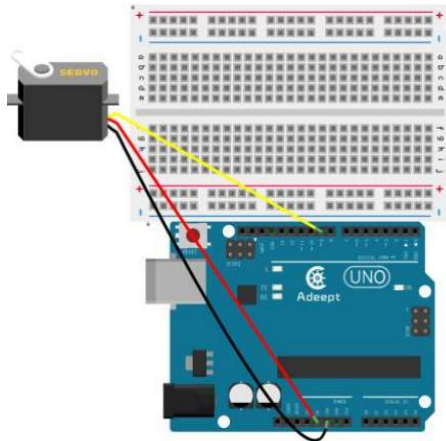
1. Ορίζουμε το τη σύνδεση του servo στο pin 9.
2. Στρέφουμε τον κινητήρα 15 μοίρες
3. Περιμένουμε 1 λεπτό
4. Στρέφουμε τον κινητήρα 30 μοίρες
5. Περιμένουμε 1 λεπτό
6. Τα Βήματα 2 έως 5 επαναλαμβάνονται ώστε ο κινητήρας να στρέφεται σε κάθε βήμα 15 μοίρες περισσότερο από το προηγούμενο μέχρι τις 90 μοίρες και στη συνέχεια να αλλάζει φορά και να μειώνει κατά 15 μοίρες τη στροφή.
7. Με την εντολή επαναληπτική εντολή for ο κινητήρας στρέφεται μια μοίρα κάθε 15 δευτερόλεπτα μέχρι τις 180 μοίρες και στη συνέχεια επιστρέφει στην αρχική του θέση με μια εντολή for και 15 δευτερόλεπτα ανάμεσα σε κάθε κίνηση.

### Βιβλιοθήκη

Η βιβλιοθήκη που υποστηρίζει τον σερβοκινητήρα στην επικοινωνία με το Arduino χρησιμοποιείται με την εντολή: `#include <Servo.h>`

### Μέθοδοι

- `attach()`: ορίζει το pin που είναι συνδεδεμένος ο κινητήρας
- `write()`: ορίζονται οι μοίρες που στρέφεται ο κινητήρας



### Πρόγραμμα

```
1. #include <Servo.h>
2. Servo myservo;//δημιουργεί το αντικείμενο myservo
3. void setup(){
4.   myservo.attach(9);//συνδέει τον κινητήρα στο pin 9
5.   myservo.write(0);// ο άξονας του κινητήρα τοποθετείται στις 0 μοίρες
6.   delay(1000);
7.   void loop(){
8.     myservo.write(15);//ο άξονας του κινητήρα τοποθετείται στις 15 μοίρες
9.     delay(1000);
10.    myservo.write(30);
11.    delay(1000);
12.    myservo.write(45);
13.    delay(1000);
14.    myservo.write(60);
15.    delay(1000);
16.    myservo.write(75);
17.    delay(1000);
18.    myservo.write(90);
19.    delay(1000);
20.    myservo.write(75);
21.    delay(1000);
22.    myservo.write(60);
23.    delay(1000);
24.    myservo.write(45);
25.    delay(1000);
26.    myservo.write(30);
27.    delay(1000);
28.    myservo.write(15);
29.    delay(1000);
30.    myservo.write(0);
31.    delay(1000);
```

```
32. for(int num=0;num<=180;num++)
33. {
34. myservo.write(num);
35. delay(10);
36. }
37. for(int num=180;num>=0;num--)
38. {
39. myservo.write(num);
40. delay(10);
41. }
42. }
```

#### ΟΔΗΓΙΕΣ

1. Συνδέστε τον κινητήρα με το Arduino όπως φαίνεται στο πιο πάνω σχήμα.
2. Ανοίξτε το πρόγραμμα servo.ino και μεταγλωττίστε το και ανεβάστε τον εκτελέσιμο κώδικα στο Arduino

#### ΕΡΩΤΗΣΕΙΣ

1. Τι αποτέλεσμα έχει η εντολή myservo.write(30);
2. Ποια η χρήση της εντολής attach();
3. Πως ονομάζεται ο σερβοκινητήρας στο πρόγραμμα
4. Ποια είναι η τιμή της μεταβλητής num στην εντολή for(int num=0;num<=180; num++)  
Πότε σταματάει να εκτελείται η εντολή for  
Πόσες φορές εκτελείται η εντολή myservo.write(num);
5. Αλλάξτε τις εντολές του προγράμματος των γραμμών 8 έως 19 με ένα μπλοκ εντολών for.
6. Αλλάξτε τις πιο κάτω εντολές του προγράμματος των γραμμών 20 έως 31 με ένα μπλοκ εντολών for.



## ΦΥΛΛΟ ΕΡΓΑΣΙΑΣ 6 - Ποτενσιόμετρο και σχεδιογράφος processing

Σε αυτό το μάθημα θα συνδέσουμε ένα ποτενσιόμετρο με το Arduino και αλλάζοντας την τιμή της τάσης, αυτή θα απεικονίζεται σε μια οθόνη LCD 1602 και θα αποτυπώνεται γραφικά στην οθόνη με έναν κύκλο που αλλάζει το μέγεθός του ανάλογα με την τιμή της τάσης. Αυτό γίνεται με τη χρήση του προγράμματος processing χρησιμοποιώντας τη σειριακή θύρα του Arduino.

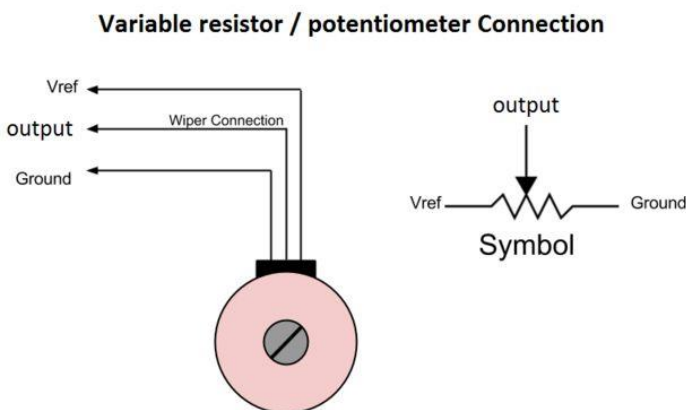
### Υλικά

- ποτενσιόμετρο
- οθόνη LCD 1602
- Arduino UNO
- Breadboard
- Διάφορα καλώδια σύνδεσης ηλεκτρονικών στοιχείων στο Breadboard
- Καλώδιο USB για τη σύνδεση του Arduino στον υπολογιστή

### Ποτενσιόμετρο

Τα ποτενσιόμετρα είναι μεταβλητές αντιστάσεις των οποίων η τιμή της αντίστασης μπορεί να αλλάξει γυρνώντας τον μοχλό που διαθέτουν στο πάνω μέρος τους. Σίγουρα έχετε χρησιμοποιήσει ένα στο παρελθόν, π.χ. για να αλλάξετε την ένταση σε ένα στερεοφωνικό.

Τα ποτενσιόμετρα υπάρχουν σε διάφορα μεγέθη αντίστασης. Μπορούν να δώσουν τιμές

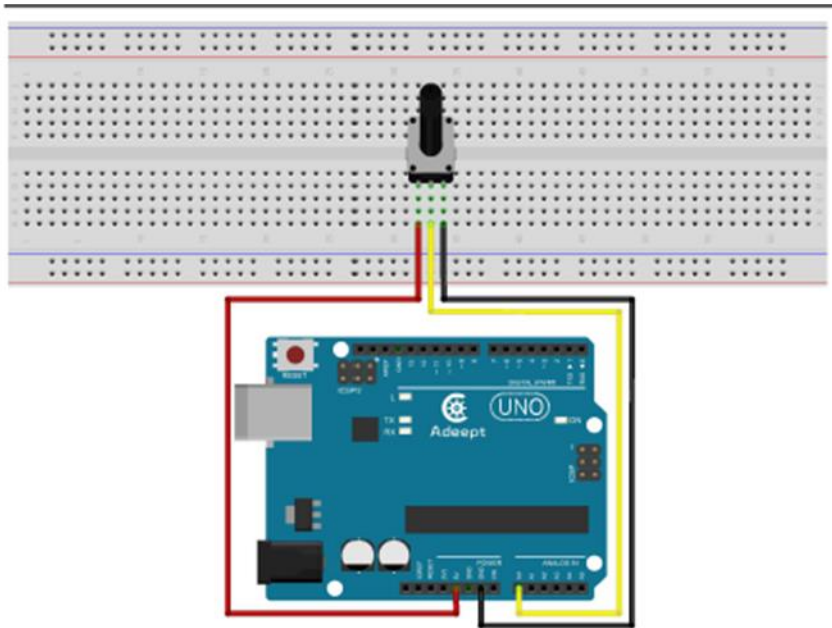


από 0  $\Omega$  μέχρι την μέγιστη τιμή για την οποία έχουν σχεδιαστεί. Για παράδειγμα, ένα ποτενσιόμετρο 10 K $\Omega$  μπορεί να δώσει τιμές από 0  $\Omega$  μέχρι μέγιστο 10 K $\Omega$ .

Όλα τα ποτενσιόμετρα έχουν τρεις ακροδέκτες. Οι εξωτερικοί χρησιμοποιούνται για την τροφοδοσία (Vred & GND). Ο μεσαίος ακροδέκτης αποδίδει την τιμή της μεταβλητής

αντίστασης ανάλογα με την θέση του μοχλού.

Για να λειτουργήσει το ποτενσιόμετρο με το Arduino χρειάζεται η βιβλιοθήκη WIRE που ενεργοποιείται με την εντολή: `#include <Wire.h>`

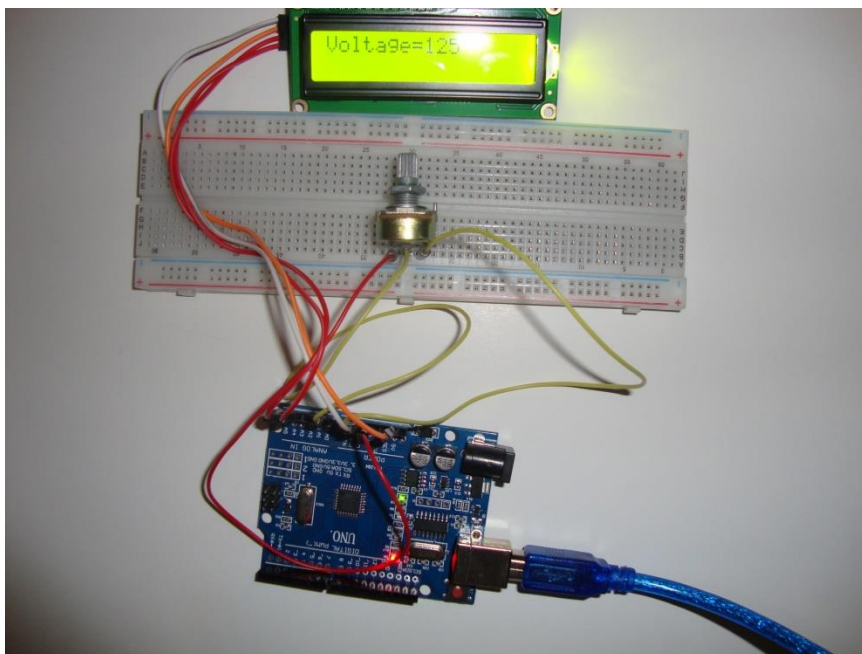


#### Σύνδεση

- Το πρώτο καλώδιο συνδέεται στη γείωση
- Το δεύτερο (μεσαίο) συνδέει το αναλογικό pin A0 του Arduino
- Το τρίτο συνδέεται με τα 5V στο Arduino.

#### Η οθόνη LCD 1602

Χρησιμοποιείται μια οθόνη LCD με 2 γραμμές των 16 χαρακτήρων. Σε αυτού του τύπου τις οθόνες μπορούμε να αποτυπώσουμε μόνο λατινικούς χαρακτήρες, σύμβολα και αριθμητικά ψηφία και όχι ελληνικούς χαρακτήρες. Για να επικοινωνήσει η οθόνη με το πρόγραμμα πρέπει να ενσωματωθεί στο πρόγραμμα η βιβλιοθήκη: LiquidCrystal\_I2C.h



### Απεικόνιση γραφικών με τη γλώσσα processing

Για να απεικονίσουμε γραφικά τις τιμές που δίνει το ποτενσιόμετρο χρησιμοποιούμε πρόγραμμα γραφικών σε γλώσσα processing. Το πρόγραμμα processing επικοινωνεί με το Arduino μέσω της σειριακής θύρας με την εντολή `Serial.read()`. Η εντολή αυτή διαβάζει την τιμή της τάσης από τη σειριακή θύρα. Στη γλώσσα του Arduino χρησιμοποιείται η εντολή `Serial.write()` που στέλνει την τιμή της τάσης στη σειριακή που παραλαμβάνει η `Serial.read()` (βλ. Παράρτημα, παρ. 2).

### Αλγόριθμος

1. Διάβασε την τιμή του ποτενσιόμετρου στη μεταβλητή `Sensor_Value`
2. Μετατροπή της τιμής που δίνει το ποτενσιόμετρο με αναλογία 0 έως 255.
3. Αποτυπώστε στην οθόνη και στείλτε την τιμή στη σειριακή θύρα.

### Εντολές

1. Διαβάζουμε την αναλογική τιμή απ' το μεσαίο pin του ποτενσιόμετρου

-> `sensorValue=analogRead(potPin)`

2. Κάνουμε αντιστοίχιση αναλογικών τιμών (0-1023) σε PWM (0-255)

-> `Voltage = map(value, 0, 1023, 0, 255);`

3. Για να αποσταλούν τα ψηφιακά δεδομένα σε σειριακή θύρα -συγκεκριμένα η τιμή της τάσης που διαβάζεται από το ποτενσιόμετρο - ώστε μετά να διαβαστεί από το πρόγραμμα σε γλώσσα processing που δημιουργεί τα γραφικά και στην LCD οθόνη.

-> `Serial.write(Voltage)`

4. Για την απεικόνιση των δεδομένων στην οθόνη LCD χρειάζεται η βιβλιοθήκη `LiquidCrystal_I2C.h` που περιέχει τις εντολές:

- `lcd.init();` : Για την αρχικοποίηση της οθόνης
- `lcd.backlight();` : Για να γίνει η οθόνη φωτεινή
- `lcd.clear();` : Καθαρίζει την οθόνη και τοποθετεί το δρομέα στην πάνω αριστερή γωνία
- `lcd.setCursor(15,0);` : Ο δρομέας της οθόνης τοποθετείται στη στήλη 15 και τη γραμμή 1
- `lcd.scrollDisplayLeft();` : Τα περιεχόμενα κυλάνε στην οθόνη κατά μια θέση προς τα αριστερά
- `lcd.print(array1[positionCounter]);` : Τυπώνει ένα μήνυμα στην οθόνη ξεκινώντας από τη θέση που βρίσκεται ο δρομέας. Για να γίνει αυτό δημιουργήθηκε ένας πίνακας με ονομασία `array1`.
- `lcd.print("Voltage=");` : Τυπώνει ένα σταθερό μήνυμα στην οθόνη

5. Με τη μέθοδο `serial.read()` το πρόγραμμα processing επικοινωνεί με το Arduino και με την `serial.write()` μεταφέρει την τιμή της τάσης στο πρόγραμμα.

## Πρόγραμμα

```

1. #include <LiquidCrystal_I2C.h>
2. #include <Wire.h>
3. int potPin = 0;
4. char array1[]="Counting the Voltage";          //the string to print on the LCD
5. LiquidCrystal_I2C lcd(0x27,16,2); // ορίζει το αντικείμενο LCD οθόνη για 16
   χαρακτήρες και 2 γραμμές
6. void setup()
7. {
8.   lcd.init();          // αρχικοποίηση της οθόνης
9.   Serial.begin(9600);
10.  lcd.backlight();
11.  lcd.clear(); //καθαρίζει την οθόνη από προηγούμενο περιεχόμενο
12.  lcd.setCursor(0,0); // τοποθετεί το δρομέα της οθόνης στη στήλη 0, γραμμή 0
13.  for (int positionCounter1 = 0; positionCounter1 < 16; positionCounter1++)
14.  {
15.    lcd.print(array1[positionCounter1]); // τυπώνει το μήνυμα στην LCD οθόνη.
16.    delay(1000);
17.  }
18. }
19. void loop() {
20.  lcd.clear();
21.  int sensorValue = analogRead(potPin);
22.  // επειδή η εντολή serial.read() αποδίδει τιμές μεταξύ του 0 και του 255 πρέπει
23.  // να αλλάξει κλίμακα με τιμές από 0 έως 1023
24.  int Voltage = map(sensorValue,0,1023,0,255);
25.  Serial.write(Voltage);
26.  lcd.print("Voltage = ");
27.  lcd.print(Voltage); //τυπώνει την πληροφορία στην LCD οθόνη
28.  Serial.print("Voltage = ");
29.  Serial.println(Voltage); //τυπώνει την πληροφορία στη σειριακή οθόνη
30.  delay(200);
31.  lcd.clear();
32. }

```

## Εντολές Processing:

### 1. size()

Ορίζει τη διάσταση σε pixels πάνω στο παράθυρο προβολής. Αυτή η συνάρτηση πρέπει να είναι στην πρώτη γραμμή του κώδικα ή η πρώτη μέσα στο setup().

Όποιο μέρος του κώδικα εμφανίζεται πριν τη εντολή size() μπορεί να τρέχει στον κώδικα περισσότερο από μια φορά και αυτό να οδηγήσει σε λάθος αποτελέσματα.

Οι μεταβλητές συστήματος width και height ορίζονται το μέγεθος του παραθύρου προβολής. Αν η size() δεν χρησιμοποιείται, τότε το παράθυρο έχει μέγεθος 100x100 pixels.

### 2. background()

Η συνάρτηση background() ορίζει το χρώμα που χρησιμοποιείται για το φόντο του παραθύρου του Processing. Εξ ορισμού το χρώμα είναι γκρι. Η συνάρτηση χρησιμοποιείται

μαζί με τη συνάρτηση `draw()` για να καθαρίζει ότι υπάρχει μέσα στο παράθυρο αλλά μπορεί και να χρησιμοποιείται μέσα στην `setup()` ώστε να ορίσει το φόντο.

### 3. `fill()`

Ορίζει το χρώμα που γεμίζει τα σχήματα.

### 4. `ellipse()`

Σχεδιάζει μια έλλειψη στην οθόνη.

### 5. `size()`

Ορίζει τη διάσταση σε pixels πάνω στο παράθυρο προβολής. Αυτή η συνάρτηση πρέπει να είναι στην πρώτη γραμμή του κώδικα ή η πρώτη μέσα στο `setup()`.

Όποιο μέρος του κώδικα εμφανίζεται πριν τη εντολή `size()` μπορεί να τρέχει στον κώδικα περισσότερο από μια φορά και αυτό να οδηγήσει σε λάθος αποτελέσματα.

Οι μεταβλητές συστήματος `width` και `height` ορίζονται το μέγεθος του παραθύρου προβολής. Αν η `size()` δεν χρησιμοποιείται, τότε το παράθυρο έχει μέγεθος 100x100 pixels.

### 6. `background()`

Η συνάρτηση `background()` ορίζει το χρώμα που χρησιμοποιείται για το φόντο του παραθύρου του Processing. Εξ ορισμού το χρώμα είναι γκρι. Η συνάρτηση χρησιμοποιείται μαζί με τη συνάρτηση `draw()` για να καθαρίζει ότι υπάρχει μέσα στο παράθυρο αλλά μπορεί και να χρησιμοποιείται μέσα στην `setup()` ώστε να ορίσει το φόντο.

### 7. `fill()`

Ορίζει το χρώμα που γεμίζει τα σχήματα.

### 8. `ellipse()`

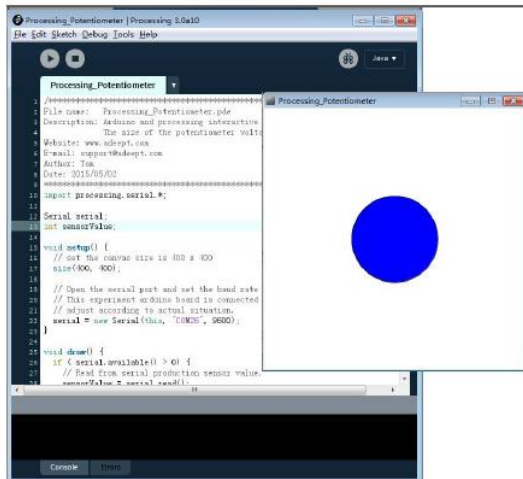
Σχεδιάζει μια έλλειψη στην οθόνη.

## Πρόγραμμα Processing

Για τη δημιουργία του σχεδίου χρησιμοποιήθηκε η [γλώσσα προγραμματισμού ανοικτού κώδικα Processing](#) που διαθέτει προγραμματιστικό περιβάλλον για προγραμματισμό εικόνων, animation και ήχων. Η γλώσσα αναπτύχθηκε στη Java, το συντακτικό της είναι απλουστευμένο και το προγραμματιστικό της μοντέλο βασίζεται στα γραφικά.


1. `import processing.serial.*;`
2. `Serial serial;`
3. `int sensorValue;`
4. `void setup() {`
5. `// ορίζει το μέγεθος του παραθύρου σε 400x400 pixels`
6. `size(400, 400);`
7. `// Ανοίγει η σειριακή θύρα με συχνότητα μετάδοσης 9600`
8. `// Εδώ χρησιμοποιούμε τη θύρα COM5`
9. `// Προσαρμόστε τις δικές σας συνθήκες`
10. `serial = new Serial(this, "COM5", 9600);`
11. `}`
12. `void draw() {`
13. `if ( serial.available() > 0) {`

14. `sensorValue = serial.read();`
15. `//Σχεδιάζει τον κύκλο`
16. `background(255); // ορίζεται χρώμα φόντου άσπρο`
17. `fill(0,0,255); // Ορίζεται το χρώμα του κύκλου μπλε`
18. `ellipse(200, 200, sensorValue, sensorValue); //Το σημείο (200,200) ορίζεται σαν το κέντρο του κύκλου`
19. `}`
20. `}}`



Τώρα αν γυρίσουμε τη λαβή του ποτενσιόμετρου στην οθόνη θα φαίνεται η μεταβολή του μεγέθους του μπλε κύκλου

## ΟΔΗΓΙΕΣ

1. Συνδέστε την πλακέτα Arduino στη θύρα USB του υπολογιστή σας.
2. Στον φάκελό σας θα βρείτε το αρχείο `current.ino`. Κάνοντας διπλό κλικ στο αρχείο ανοίγει το προγραμματιστικό περιβάλλον του Arduino και βλέπετε το περιεχόμενο του. Μεταγλωττίστε το και ανεβάστε το στην πλακέτα Arduino.
3. Στο φάκελό σας, επίσης, υπάρχει το αρχείο `cycle.pde`. Κάνοντας διπλό κλικ στο αρχείο ανοίγει το προγραμματιστικό περιβάλλον της γλώσσας προγραμματισμού Processing και βλέπετε το περιεχόμενό του. Πατήστε το πλήκτρο  για να εκτελέσετε το πρόγραμμα και δείτε στο νέο παράθυρο που ανοίγει τον κύκλο που εμφανίζεται.
4. Χρησιμοποιείτε το μοχλό του ποτενσιόμετρου για να αυξομειωθεί η τάση του ρεύματος (δεξιά για αύξηση και αριστερά για μείωση). Παρατηρείστε την αλλαγή στο μέγεθος του κύκλου που είναι ανάλογη με την τάση.

## ΕΡΩΤΗΣΕΙΣ

1. Περιγράψτε τι κάνουν τα προγράμματα:

.....

.....

.....  
Σβήστε το ερωτηματικό «;» από την εντολή `lcd.scrollDisplayLeft()`; μεταγλωττίστε και εκτελέστε το πρόγραμμα. Τι παρατηρείτε; Ποιο μήνυμα εμφανίζεται;

.....

.....

2. Τι είδους λάθος δημιουργήσατε;

.....

3. Διορθώστε το λάθος σας και επανεκτελέστε το πρόγραμμά σας.

4. Σε ποιόν ακροδέκτη είναι συνδεδεμένο το μεσαίο pin του ποτενσιόμετρου;

.....

5. Ποια εντολή του προγράμματος Processing χρησιμοποιείται ώστε το εσωτερικό του κύκλου να είναι μπλε;

.....

6. Ποια είναι η εντολή του προγράμματος Processing που ρυθμίζει το μέγεθος του κύκλου;

.....

7. Ποια είναι η χρήση της εντολής `Serial.write(Voltage)`;

.....

8. Ποια είναι η χρήση της εντολής `sensorValue = serial.read()`;

.....

9. Ποια η χρήση της εντολής: `lcd.clear()`;

.....

## ΦΥΛΛΟ ΕΡΓΑΣΙΑΣ 7 - Κατασκευή ραντάρ με Arduino

### Βαθμός δυσκολίας: Δύσκολο

Σε αυτό το μάθημα θα δημιουργήσουμε ένα ραντάρ που με τη βοήθεια ενός αισθητήρα υπερήχων, ένα RGB LED και ένα ηχείο θα μας ειδοποιεί όταν ένα εμπόδιο πλησιάζει προς αυτό. Το RGB LED αλλάζει το χρώμα του όπως ο αισθητήρας υπερήχων ανιχνεύει διαφορετικά αποτελέσματα απόστασης και το ηχείο παράγει ήχους σε διαφορετικές συχνότητες.

### Υλικά

- Arduino UNO Microcontrolier
- USB καλώδιο
- RGB LED
- αισθητήρας υπερήχων
- passive buzzer (ηχείο)
- 1 αντίσταση 1KΩ
- 1 NPN Transistor (S8050)
- 3 αντιστάσεις 220Ω
- 1 Breadboard
- Διάφορα καλώδια σύνδεσης ηλεκτρονικών στοιχείων στο Breadboard

### Περιγραφή

Χρησιμοποιούμε τον αισθητήρα Υπερήχων HC-SR04 για τον υπολογισμό απόστασης. Η απόσταση που μπορεί να υπολογίσει είναι από 2εκ. έως 400εκ. με ακρίβεια ενός εκατοστού σύμφωνα με τον κατασκευαστή του.



Ο αισθητήρας έχει 4 pins:

- Ground,
- VCC (5V),
- Trig (σε ένα ψηφιακό pin) και
- Echo (σε ένα ψηφιακό pin)

Εκπέμπει έναν υπέρηχο στα 40.000 Hz το οποίο διαδίδεται μέσω του αέρα και αν βρει εμπόδιο από κάποιο αντικείμενο στη διαδρομή του ανακλάται πίσω. Λαμβάνοντας υπόψη το χρόνο της διαδρομής και την ταχύτητα του ήχου μπορείτε να υπολογιστεί ην απόσταση.



Για τη λειτουργία του αισθητήρα πρέπει να τεθεί το Trig στο High για 10  $\mu$ s.

Χρησιμοποιείται η μέθοδος **pulseIn(echoPin, HIGH)** η οποία μετράει το χρόνο σε microseconds μέχρι το echoPin γίνει HIGH και το αποθηκεύει στη μεταβλητή duration.

Για παράδειγμα, αν το αντικείμενο είναι 1 εκατοστό μακριά από τον αισθητήρα και δεδομένου ότι η ταχύτητα του ήχου είναι 340 m/s ή 0.034 cm/ $\mu$ s το ηχητικό κύμα πρέπει να μετακινηθεί περίπου για 29,4  $\mu$ s.

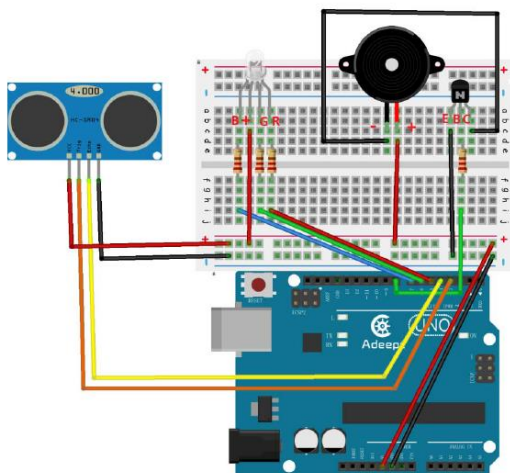
$$\text{Χρόνος}(t) = \text{διάστημα}(s) / \text{ταχύτητα}(v) = 1\text{cm} / 0,034 \text{ cm}/\mu\text{s} = 29,4 \mu\text{s}$$

Αυτή η τιμή που λαμβάνεται από το Echo pin διπλασιάζεται, γιατί το ηχητικό κύμα μετακινείται μπροστά και στη συνέχεια ανακλάται πίσω, άρα η απόσταση που διανύει το κύμα είναι διπλάσια.

Γι αυτό η διάρκεια που παίρνουμε από το Echo pin του αισθητήρα πρέπει να διαιρεθεί με 58,8 ( $\text{distance}=(\text{duration}/2)/29,4;$ ) για να υπολογιστεί η απόσταση.

### Υλικά

- Arduino UNO
- USB καλώδιο
- RGB LED
- αισθητήρας υπερήχων
- passive buzzer (ηχείο)
- 1 αντίσταση 1K $\Omega$
- 1 NPN Transistor (S8050)
- 3 αντιστάσεις 220 $\Omega$
- 1 Breadboard
- Διάφορα καλώδια σύνδεσης ηλεκτρονικών στοιχείων στο Breadboard



### Αλγόριθμος

1. Θέσε το trig στο high 10  $\mu$ s
2. Μέτρησε την απόσταση από το εμπόδιο
3. Μετάτρεψέ την απόσταση σε εκατοστά του μέτρου

4. Αν η απόσταση είναι μικρότερη από 20 εκατοστά, το led γίνεται κόκκινο και το ηχείο εκπέμπει ήχο στα 900 Hz  
 Διαφορετικά, αν η απόσταση είναι μικρότερη από 120 εκατοστά, το led γίνεται πράσινο και το ηχείο εκπέμπει ήχο στα 300 Hz  
 διαφορετικά το led γίνεται κίτρινο και το ηχείο εκπέμπει ήχο διακεκομμένα.

#### Συναρτήσεις για το χρωματισμό του LED και της συχνότητας του ηχείου

1. Για τη ρύθμιση των χρωμάτων του RGB LED χρησιμοποιείται η συνάρτηση **setColor** με ορίσματα 3 ακέραιους αριθμούς που παίρνουν τιμές από 0 έως 255. Όταν ο πρώτος παίρνει την τιμή 255 και οι άλλοι δύο το 0, τότε το LED γίνεται κόκκινο. Όταν το μεσαίο παίρνει τιμή 255 και τα άλλα δύο 0, το LED γίνεται πράσινο και όταν οι δύο πρώτοι είναι 0 και ο τρίτος 255 το LED είναι κίτρινο. Με διαφορετικούς συνδυασμούς μπορούμε να επιτύχουμε διαφορετικούς χρωματισμούς.
2. Για τη ρύθμιση της συχνότητας του ηχείου χρησιμοποιούνται οι συναρτήσεις **buzzer1()**, **buzzer2()** και **buzzer3()**. Σε αυτές τις συναρτήσεις χρησιμοποιείται η εντολή **tone(beepPin, <συχνότητα>)**. Το beepPin αντιπροσωπεύει το pin στο οποίο είναι συνδεδεμένο το ηχείο. Για παράδειγμα αν θέλουμε το ηχείο να ηχήσει στα 600 Hz η εντολή πρέπει να είναι: **tone(beepPin, 600);**

#### Πρόγραμμα

```

1. const int TringPin=3;
2. const int EchoPin=4;
3. int ledPin1 = 5;
4. int ledPin2 = 6;
5. int ledPin3 = 7;
6. int i = 0;
7. int beepPin = 8;
8. void setup()
9. {
10. Serial.begin(9600);
11. pinMode(EchoPin,INPUT);
12. pinMode(TringPin,OUTPUT);
13. pinMode(ledPin1,OUTPUT);
14. pinMode(ledPin2,OUTPUT);
15. pinMode(ledPin3,OUTPUT);
16. pinMode(beepPin,OUTPUT);
17. }
18. void loop()
19. {
20. digitalWrite(TringPin,LOW);
21. delayMicroseconds(2);
22. digitalWrite(TringPin,HIGH);
23. delayMicroseconds(10);
24. int distance = pulseIn(EchoPin,HIGH);
25. distance = distance/58;
26. Serial.println(distance);
27. if (distance<20)
28. {
29. setColor(255,0,0);
30. buzzer1();

```

```

31. }
32. else if(distance<=120)
33. {
34. setColor(0,0,255);
35. buzzer2();
36. }
37. else
38. {
39. setColor(0,255,0);
40. buzzer3();
41. }
42. }
43. void buzzer1()
44. {
45. tone (beepPin,900);
46. delay(300);
47. noTone(beepPin);
48. }
49. void buzzer2()
50. {
51. tone (beepPin,600);
52. delay(300);
53. noTone(beepPin);
54. }
55. void buzzer3()
56. {
57. if (i%100 == 1){
58. tone(beepPin,300);
59. }
60. else{
61. noTone(beepPin);
62. }
63. i++;
64. }
65. void setColor(int red,int green,int blue)
66. {
67. analogWrite(ledPin1,red);
68. analogWrite(ledPin2,green);
69. analogWrite(ledPin3,blue);
70. }

```

### Οδηγίες

Συνδέστε την πλακέτα Arduino στη θύρα USB του υπολογιστή σας.

Στον φάκελό σας θα βρείτε το αρχείο radar.ino. Κάνοντας διπλό κλικ στο αρχείο ανοίγει το προγραμματιστικό περιβάλλον του Arduino και βλέπετε το περιεχόμενο του. Μεταγλωττίστε το και ανεβάστε το στην πλακέτα Arduino.

### Ερωτήσεις

1. Περιγράψτε τι κάνει το πρόγραμμα: .....

.....

.....

2. Σβήστε το ερωτηματικό «;» από την εντολή `digitalWrite(TringPin, HIGH);` μεταγλωττίστε και εκτελέστε το πρόγραμμα. Τι παρατηρείτε; Ποιο μήνυμα εμφανίζεται;

.....

.....

3. Πως ονομάζεται το λάθος που μόλις δημιουργήσατε;

.....

.....

4. Διορθώστε το λάθος σας και επανεκτελέστε το πρόγραμμά σας.

5. Σε ποιούς ακροδέκτες είναι συνδεδεμένο το RGB LED; Που φαίνεται αυτό στο πρόγραμμα;

.....

.....

6. Ποιες εντολές του προγράμματος χρησιμοποιείται για να ανάψει το RGB LED σε χρώμα πράσινο;

.....

.....

7. Πόση πρέπει να είναι η απόσταση του εμποδίου από τον αισθητήρα ώστε το χρώμα του RGB LED να γίνεται πράσινο;

.....

8. Ποια είναι η χρήση της εντολής `int distance = pulseIn(EchoPin,HIGH);`;

.....

9. Αλλάξτε το καλώδιο που συνδέει το ηχείο από το pin 8 σε 9. Αλλάξτε το πρόγραμμά σας ώστε το ηχείο να λειτουργεί.

10. Ποια η χρήση της εντολής: `tone(beepPin,300);`

.....

## ΦΥΛΛΟ ΕΡΓΑΣΙΑΣ 8 – Πρόγραμμα που κινεί έναν κινητήρα DC

**Βαθμός Δυσκολίας:** Δύσκολο

Σε αυτό το πείραμα ελέγχεται η κατάσταση του κινητήρα με τη βοήθεια μιας πλατφόρμας Arduino. Οι καταστάσεις που μπορεί να βρίσκεται ο κινητήρας είναι: κίνηση, σταμάτημα, αντίθετη κίνηση, επιτάχυνση και επιβράδυνση.

### Περιγραφή

Ο DC κινητήρας είναι μια συσκευή που μετατρέπει την ηλεκτρική ενέργεια σε μηχανική. Επειδή εύκολα ελέγχεται, μπορεί να χρησιμοποιηθεί σαν ανεμιστήρας, ηλεκτρονικό παιχνίδι κ.λπ.

Ο κινητήρας DC δουλεύει με συνεχές ρεύμα. Για να συνδεθεί με το Arduino χρειάζονται τέσσερα καλώδια, ένα στη θύρα των 5V, ένα στη γείωση και τα άλλα δύο συνδέονται σε δύο pins (στο δικό μας παράδειγμα είναι το 9 και το 10). Ανάλογα με τη σύνδεση η κίνηση γίνεται δεξιόστροφα ή αριστερόστροφα.

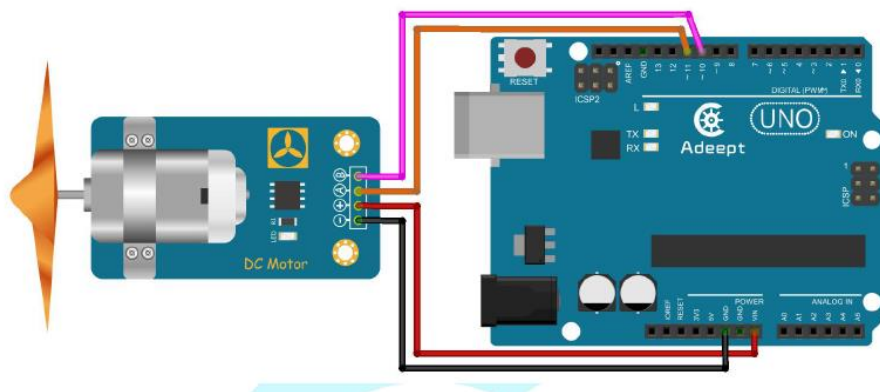
### Υλικά

- 1 Arduino UNO R3 Board
- 1 DC Motor Module
- 4 jumper καλώδια

Σημείωση: Πρέπει πρώτα να συνδέουμε πρώτα την τροφοδοσία. Αφού δώσουμε ενέργεια στη συσκευή, συνδέουμε τα pins σήματος. Διαφορετικά ο κινητήρας μπορεί να ζεσταθεί και να μη δουλεύει.

### Σύνδεση με το Arduino

- Το καλώδιο A συνδέεται στο pin 9
- Το καλώδιο B συνδέεται στο pin 10
- Το καλώδιο - συνδέεται στο GROUND
- Το καλώδιο + συνδέεται στο 5V



Όταν φορτωθεί το πρόγραμμα στο Arduino ο ανεμιστήρας, ανάλογα με τις εντολές, γυρίζει, επιταχύνεται, επιβραδύνει, σταματά, ξεκινά κ.ο.κ. Την τιμή της ταχύτητας μπορείτε να τη βλέπετε στη σειριακή οθόνη.

### Άσκηση 1η

Σε αυτήν την άσκηση ο κινητήρας κινείται προς τα δεξιά για 5 δευτερόλεπτα, μετά από δύο δευτερόλεπτα κινείται αριστερά και περιμένει δύο δευτερόλεπτα. Η εντολή που χρησιμοποιείται για τη κίνηση είναι η: `DigitalWrite(<pin>, <τιμές ταχύτητας από 0 έως 255>);`

### Αλγόριθμος

1. Κινείστε τον κινητήρα δεξιά στη μέγιστη ταχύτητα για 5 δευτερόλεπτα
2. Περιμένε 2 δευτερόλεπτα
3. Κινείστε τον κινητήρα αριστερά στη μέγιστη ταχύτητα για 5 δευτερόλεπτα
4. Περιμένε 2 δευτερόλεπτα
5. Σταμάτησέ τα όλα για 2 δευτερόλεπτα
6. Επανάλαβε τα βήματα 1 έως 5

### Πρόγραμμα

```

1. int INA = 9;
2. int INB = 10;
3. void setup() {
4.   pinMode(INA,OUTPUT);
5.   pinMode(INB,OUTPUT);      }
6. void loop() {
7.   digitalWrite(INA,255);
8.   digitalWrite(INB,LOW);
9.   delay(5000);
10.  digitalWrite(INA,LOW);
11.  digitalWrite(INB,LOW);
12.  delay(200);
13.  digitalWrite(INA,LOW);
14.  digitalWrite(INB,255);
15.  delay(5000);
16.  digitalWrite(INA,LOW);
17.  digitalWrite(INB,LOW);
18.  delay(200);      }
```

### Οδηγίες

Συνδέστε την πλακέτα Arduino στη θύρα USB του υπολογιστή σας.

Στον φάκελό σας θα βρείτε το αρχείο `dc_motor1.ino`. Κάνοντας διπλό κλικ στο αρχείο ανοίγει το προγραμματιστικό περιβάλλον του Arduino και βλέπετε το περιεχόμενο του. Μεταγλωττίστε το και ανεβάστε το στην πλακέτα Arduino.

## Άσκηση 2η

Σε αυτήν την άσκηση ο κινητήρας αυξάνει σταδιακά την ταχύτητά του και μετά από ένα δευτερόλεπτο την μειώνει επίσης σταδιακά.

## Αλγόριθμος

1. Σταμάτησε τον κινητήρα
2. Όρισε την ταχύτητα στο 60
3. Αύξησε την ταχύτητα του κινητήρα κατά 20
4. Περίμενε 1 δευτερόλεπτο
5. Επανάλαβε 7 φορές τα βήματα 3 και 4
6. Μείωσε την ταχύτητα του κινητήρα κατά 20
7. Περίμενε 1 δευτερόλεπτο
8. Επανάλαβε 6 φορές τα βήματα 5 και 6

## Εντολές

**digitalWrite(Apin, speed):** η εντολή αυτή κινεί τον κινητήρα με ταχύτητα ίση με την τιμή της μεταβλητής speed προς τα δεξιά.

**digitalWrite(Bpin, speed):** η εντολή αυτή κινεί τον κινητήρα με ταχύτητα ίση με την τιμή της μεταβλητής speed προς τα αριστερά.

## Πρόγραμμα

```

1. void setup()
2. {
3.   pinMode(Apin,OUTPUT);
4.   pinMode(Bpin,OUTPUT);
5.   Serial.begin(9600);           //ανοίγει η σειριακή θύρα με ρυθμό στα 9600 bps
6. }
7. void loop()
8. {
9.   digitalWrite(Apin,0);
10.  delay(1000);
11.  DCmotorspeed = 60;
12.  for(int i=0;i<7;i++){
13.    DCmotorspeed = DCmotorspeed + 20;
14.    digitalWrite(Apin,DCmotorspeed);
15.    digitalWrite(Bpin,0);
16.    Serial.println(DCmotorspeed); //στέλνει δεδομένα στη σειριακή οθόνη
17.    delay(1000);
18.  }
19.  for(int i=0;i<6;i++){
20.    DCmotorspeed = DCmotorspeed - 20;

```

```
21. digitalWrite(Apin,0);
22. digitalWrite(Bpin,DCmotorspeed);
23. Serial.println(DCmotorspeed);    //στέλνει δεδομένα στη σειριακή οθόνη
24. delay(1000);
25. DCmotorspeed = DCmotorspeed - 20;
26. }
27. digitalWrite(Apin,0);
28. digitalWrite(Bpin,0);
29. }
```



### Οδηγίες

Συνδέστε την πλακέτα Arduino στη θύρα USB του υπολογιστή σας.

Στον φάκελό σας θα βρείτε το αρχείο dc\_motor2.ino. Κάνοντας διπλό κλικ στο αρχείο ανοίγει το προγραμματιστικό περιβάλλον του Arduino και βλέπετε το περιεχόμενό του. Μεταγλωττίστε το και ανεβάστε το στην πλακέτα Arduino.

### Ερωτήσεις

1. Με ποια εντολή κινείται ο κινητήρας DC;
2. Ποιο το αποτέλεσμα της εντολής: digitalWrite(Apin, 0);



## ΦΥΛΛΟ ΕΡΓΑΣΙΑΣ 9 – Ρύθμιση έντασης LED με αισθητήρα BLUETOOTH

**Βαθμός Δυσκολίας:** Δύσκολο

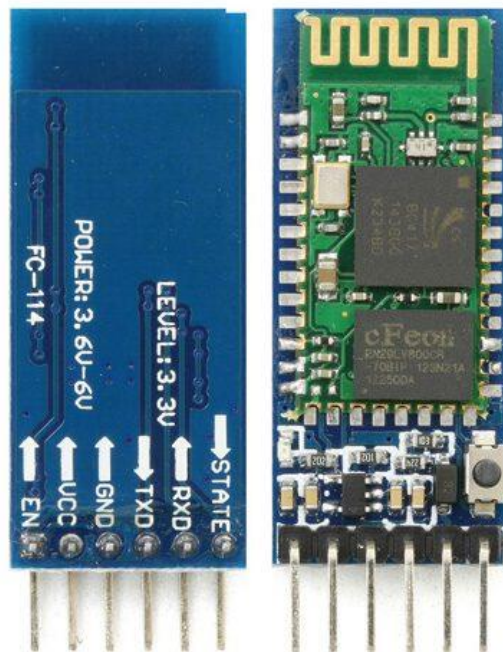
### Περιγραφή

Σε αυτήν την άσκηση θα αναβοσβήνουμε ένα led από smartphone ή tablet χρησιμοποιώντας έναν αισθητήρα Bluetooth συνδεδεμένο με Arduino. Εκτός από τα υλικά του Arduino, θα κατεβάσουμε μια εφαρμογή Android που ονομάζεται BlueControl για να ελεγχθεί το led μέσω αυτής.

**Εφαρμογές:** Με τους αισθητήρες Bluetooth μπορούμε να επικοινωνούμε απομακρυσμένα με άλλες συσκευές όπως smartphones ή tablets και να διαχειριζόμαστε συσκευές όπως οικιακές συσκευές, οχήματα κ.λπ.

Οι δυο πιο δημοφιλείς μονάδες Bluetooth της αγοράς είναι:

- Το **HC-05** με το οποίο παρέχεται η δυνατότητα του να λειτουργεί ως master και slave δηλαδή και στέλνει και δέχεται αιτήσεις και,
- το **HC-06** που διαθέτει 2 παραπάνω pins και λειτουργεί μόνο ως slave (λαμβάνει μόνο αιτήσεις σύνδεσης)



Η μονάδα HC-05 έχει 6 pins:

- VCC (2), στο οποίο συνδέουμε την τροφοδοσία από το Arduino.
- GND (3), στο οποίο συνδέουμε την γείωση.
- RX (5), στο οποίο έρχονται τα δεδομένα από το Arduino.
- TX (4), από το οποίο αποστέλλονται τα δεδομένα που λαμβάνει η μονάδα Bluetooth προς το Arduino.
- STATE (6), το οποίο είναι απλά συνδεδεμένο με το LED της μονάδας Bluetooth και όταν αυτό ανάβει βγάζει έξοδο HIGH αλλιώς βγάζει έξοδο LOW.

- ENABLED ή KEY (1) το οποίο μας επιτρέπει να αλλάζουμε την κατάσταση της μονάδας μεταξύ δύο:
  - Κατάσταση δεδομένων, όπου το HC-05 λειτουργεί κανονικά για να μεταφέρει δεδομένα.
  - Κατάσταση ρυθμίσεων, όπου μπορούμε να δίνουμε εντολές στο HC-05 για να αλλάζουμε τις ρυθμίσεις του.

Επίσης διαθέτει μια ένδειξη LED (8) η οποία ανάλογα με τον ρυθμό που αναβοσβήνει μας πληροφορεί για την κατάσταση της μονάδας, καθώς και ένα πλήκτρο (7) το οποίο μας επιτρέπει να αλλάζουμε την κατάσταση της μονάδας.

Η επικοινωνία του Bluetooth με το Arduino γίνεται μέσω της σειριακής επικοινωνίας που περιλαμβάνονται στην βιβλιοθήκη Serial.

**Σημείωση:** Όταν θέλουμε να ανεβάσουμε το πρόγραμμα στο Arduino από τον υπολογιστή μας χρησιμοποιώντας την θύρα USB θα πρέπει να προσέξουμε να απενεργοποιήσουμε προσωρινά την μονάδα Bluetooth (απλά αποσυνδέστε το καλώδιο στο pin VCC, ή τα καλώδια στο RX και TX pins) για να μπορέσει να ανέβει το πρόγραμμά μας. Αν ξεχάσουμε να αποσυνδέσουμε το Bluetooth, το Arduino IDE θα μας ειδοποιήσει ότι η σειριακή θύρα του Arduino είναι απασχολημένη (αφού την έχει δεσμεύσει το Bluetooth) και δεν θα ανεβάσει το πρόγραμμα. Αφού ανεβάσουμε το πρόγραμμά μας στο Arduino μπορούμε να συνδέσουμε και πάλι την μονάδα HC-05 ώστε να λειτουργήσει κανονικά.

#### Υλικά

- Arduino
- Bluetooth συσκευή/αισθητήρας – HC05
- Καλώδια
- LEDs
- Μία συσκευή ANDROID

#### Η διαδικασία επικοινωνίας

Κατεβάστε την εφαρμογή Android που ονομάζεται **BlueControl** (δωρεάν). Αυτόματα ενεργοποιεί το Bluetooth της συσκευής σας.

- Συνδέστε τον αισθητήρα όπως φαίνεται στην πιο πάνω εικόνα. Μη συνδέσετε τα RX & TX pins πριν ανεβάσετε τον κώδικα!
- Ανεβάστε στο Arduino τον πιο πάνω κώδικα.
- Την πρώτη φορά που η συσκευή σας δει το Bluetooth αισθητήρα, σας ζητάει password που είναι ή 1234 ή 0000.
- Όταν η συσκευή συνδεθεί με τον αισθητήρα, το LED θα ξεκινήσει να αναβοσβήνει με πιο αργό ρυθμό από το συνηθισμένο.

## Αλγόριθμος

1. Διάβασε τα δεδομένα από τον αισθητήρα Bluetooth
2. Τύπωσε τα δεδομένα στη σειριακή θύρα
3. Αν το σήμα είναι 1 (δηλαδή έχει πατηθεί ο χαρακτήρας A στην εφαρμογή BlueControl)  
Αναψε το led
4. Αν το σήμα είναι 0 (δηλαδή έχει πατηθεί ο χαρακτήρας B στην εφαρμογή BlueControl)  
Σβήσε το led

## Εντολές

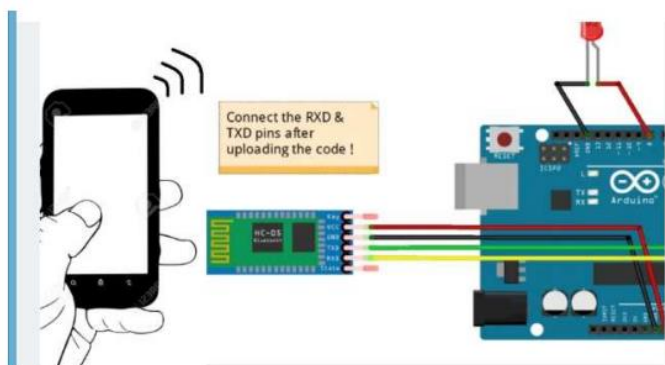
Χρησιμοποιείται η εντολή switch που ανήκει στην κατηγορία των εντολών επιλογής όπως και η if.

## Πρόγραμμα

```
void setup() {
  Serial.begin(9600);
  pinMode(13, OUTPUT);
}

void loop() {
  if(Serial.available()>0) {
    char data= Serial.read();
    Serial.print(data);
    switch(data) {
      case '1': digitalWrite(13, HIGH);break; // όταν πατηθεί το a στην εφαρμογή BlueControl
                                                    //το led ανάβει
      case '0': digitalWrite(13, LOW);break; // όταν πατηθεί το d στην εφαρμογή BlueControl
                                                    // το led σβήνει
      default : break;    } // στις υπόλοιπες περιπτώσεις δεν συμβαίνει κάτι
    Serial.println(data);  }
  delay(50);  }
```

#### ΕΛΕΓΧΟΣ LED ΜΕ BLUETOOTH



### Οδηγίες

Συνδέστε την πλακέτα Arduino στη θύρα USB του υπολογιστή σας.

Στον φάκελό σας θα βρείτε το αρχείο blth.ino. Κάνοντας διπλό κλικ στο αρχείο ανοίγει το προγραμματιστικό περιβάλλον του Arduino και βλέπετε το περιεχόμενό του. Μεταγλωττίστε το και ανεβάστε το στην πλακέτα Arduino.

## ΠΑΡΑΡΤΗΜΑ

### 1. Διαχείριση του αισθητήρα Bluetooth από τη σειριακή θύρα του Arduino

Μόλις ανοίξουμε την σειριακή οθόνη θα εμφανιστούν αρχικά οι πληροφορίες που έχουμε βάλει στην συνάρτηση setup του προγράμματος.

#### Έλεγχος Λειτουργίας

Για να ελέγξουμε αν όλα έχουν συνδεθεί σωστά και μπορεί το HC-05 να δεχτεί εντολές μπορούμε να στείλουμε την εντολή AT, η οποία θα πρέπει να μας επιστρέψει OK που σημαίνει ότι η σύνδεση είναι σωστή.

Αν όλα πάνε καλά η εντολή AT θα μας επιστρέψει OK.

#### Διαχείριση ονόματος της συσκευής

Για να δούμε το όνομα που έχει ήδη η μονάδα HC-05 πληκτρολογούμε την εντολή AT+NAME.

Το όνομα της μονάδας είναι αυτή τη στιγμή HC-05

Για να αλλάξουμε το όνομα γράφουμε την εντολή AT+NAME=Νέο όνομα. Έτσι, αν για παράδειγμα φτιάχνουμε έναν έξυπνο φωτισμό όπου θα συνδέεται με το κινητό μας μέσω Bluetooth για να στέλνει δεδομένα, μπορούμε να δώσουμε την εντολή AT+NAME= Light Manager.

Αλλάζουμε το όνομα του HC-05 σε Light Manager.

Για να ελέγξουμε αν όντως έγινε η αλλαγή, μπορούμε να δώσουμε ξανά την εντολή AT+NAME για να δούμε το όνομα της μονάδας.

Δίνοντας την εντολή AT+NAME βλέπουμε πως το όνομα του HC-05 έχει πλέον αλλάξει.

Για να αλλάξουμε τον κωδικό της μονάδας όταν γίνεται η σύζευξη με άλλες συσκευές Bluetooth χρησιμοποιούμε την εντολή AT+PSWD=<Νέος κωδικός>

Δίνοντας την εντολή AT+PSWD βλέπουμε τον τρέχοντα κωδικό της μονάδας

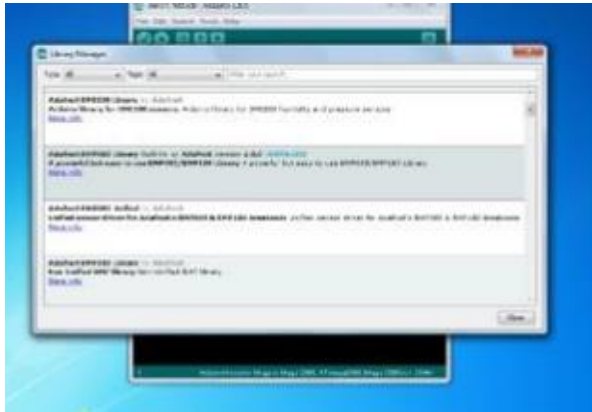
Αλλάζουμε τον κωδικό σε 0000 με την εντολή AT+PSWD=0000

### 2. Βιβλιοθήκες

Οι βιβλιοθήκες επεκτείνουν το περιβάλλον του Arduino και παρέχουν επιπλέον λειτουργικότητα στα προγράμματα. Οι περισσότεροι αισθητήρες για να προγραμματιστούν πρέπει να ενσωματώσουμε την αντίστοιχη βιβλιοθήκη.

Για να ενσωματωθεί μια έτοιμη βιβλιοθήκη πρέπει ο χρήστης να την κατεβάσει από το διαδίκτυο ή να δημιουργήσει τη δική του.

Η διαδικασία εισαγωγής μιας βιβλιοθήκης στο πρόγραμμα Arduino γίνεται με τη χρήση του «Σχέδιο» στο μενού και μετά επιλέγεται Import Library (εισαγωγή βιβλιοθήκης). Εμφανίζεται η επιλογή Add Library (προσθήκη βιβλιοθήκης).



### 3. Σειριακή θύρα και σειριακή οθόνη

Το Arduino επικοινωνεί με τον υπολογιστή μέσω ενός καλωδίου USB, για να το τροφοδοτηθεί με ρεύμα και για να ανέβει το πρόγραμμα σε αυτό, αλλά και για να ανταλλάσσουμε δεδομένα μέσω σειριακής επικοινωνίας.

Μπορούμε να χρησιμοποιήσουμε την σειριακή οθόνη (Serial Monitor) που περιλαμβάνεται στο Arduino IDE ή να γράψουμε τα δικά μας προγράμματα τα οποία θα στέλνουν και θα δέχονται δεδομένα από την σειριακή θύρα για να χρησιμοποιηθούν είτε σε κάποιον αισθητήρα είτε σε κάποιο πρόγραμμα όπως το processing (παράδειγμα θα δούμε στο φύλλο εργασίας 6 και στο φύλλο εργασίας 9).

Η σειριακή οθόνη εμφανίζει τα σειριακά δεδομένα που στέλνονται από το Arduino. Μπορούμε να επιλέξουμε το ρυθμό μετάδοσης δεδομένων (baud rate) ανάλογα με την τιμή που θα επιλέξουμε στο πρόγραμμά μας με την εντολή `Serial.begin()`. Συνήθως χρησιμοποιείται η τιμή 9600 εξ ορισμού `<Serial.begin(9600)>;`.

Για να δούμε τις τιμές που καταγράφει ο αισθητήρας θα χρησιμοποιήσουμε τη λειτουργία της παρακολούθησης της σειριακής από το περιβάλλον του προγράμματος.

#### Εντολές σειριακής θύρας

**`Serial.begin(9600);`** Η εντολή αυτή ενεργοποιεί τη σειριακή θύρα και έτσι μπορεί να εμφανιστεί η πληροφορία στο παράθυρο της σειριακής.

**`Serial.println("<κείμενο>");`** Η εντολή αυτή τυπώνει την πληροφορία στο παράθυρο της σειριακής και δημιουργεί μια νέα γραμμή. Παράδειγμα: Η εκτέλεση της εντολής `Serial.println("Καλημέρα!");` τυπώνει το λεκτικό Καλημέρα! στη σειριακή οθόνη ακολουθούμενη από το Enter.

**`Serial.print("<κείμενο>");`** Η εντολή αυτή τυπώνει την πληροφορία στο παράθυρο της σειριακής. Παράδειγμα: Η εκτέλεση της εντολής `Serial.print("Καλημέρα!");` τυπώνει το λεκτικό Καλημέρα!.

**`Serial.write();`** : στέλνει δεδομένα στη σειριακή θύρα

**Serial.read();** : παραλαμβάνει δεδομένα από τη σειριακή θύρα

Παραδείγματα

- `serial.write(45);` ⇒ στέλνει την τιμή 45 στη σειριακή θύρα
- `Serial.write(val);` ⇒ στέλνει την τιμή της μεταβλητής `val` στη σειριακή θύρα
- `serial.write("hello");` ⇒ στέλνει τη λέξη `hello` στη σειριακή θύρα
- `serial.read(45);` ⇒ η σειριακή θύρα παραλαμβάνει την τιμή 45
- `serial.read(45);` ⇒ η σειριακή θύρα παραλαμβάνει την τιμή 45

`serial.read(val);` ⇒ η σειριακή θύρα παραλαμβάνει την τιμή της μεταβλητής `val`

## Βιβλιογραφία

Brousseau, G. (1984). The crucial role of the didactical contract in the analysis and construction of situations in teaching and learning mathematics. In H.-G. Steiner (Ed.), *Theory of mathematics education: ICME 5 – topic area and miniconference: Adelaide, Australia*. Bielefeld, Germany: Institut fuer Didaktik der Mathematik der Universitaet Bielefeld.

Gonzalez, H. B., & Kuenzi, J. J. (2012). *Science, technology, engineering, and mathematics (STEM) education: A primer*. Congressional Research Service. Ανακτήθηκε 14 Σεπτεμβρίου 2019 από <http://www.fas.org/sgp/crs/misc/R42642.pdf>.

Psycharis, S. (2015). 'The Impact of Computational Experiment and Formative Assessment in Inquiry Based Teaching and Learning Approach in STEM Education ; *Journal of Science Education, and Technology* 25(2),316-326 (JOST) DOI 10.1007/s10956-015-9595-z

Psycharis, S., (2016). 'Inquiry Based- Computational Experiment, Acquisition of Threshold Concepts and Argumentation in Science and Mathematics Education (Accepted for publication at *Journal "Educational Technology & Society"*- Volume 19, Issue 3, 2016.

Wing, J. M. (2006). Computational thinking. *Communications of the ACM*, 49, 33-35.

Ψυχάρης, Σ. (2016). *Η Καινοτομία-Αριστεία στα Πρότυπα Σχολεία ως συνάρτηση του STEM και της Διαφοροποιημένης Μάθησης*. Ανακτήθηκε 14 Σεπτεμβρίου, 2019 από <http://goo.gl/RsgWzV>.

Κοτίνη Ι., & Τζελέπη Σ. (2013). Σχεδιάζοντας ένα μάθημα Πληροφορικής που βασίζεται στις αρχές του Εποικοδομητισμού, της Υπολογιστικής Σκέψης και της Παιχνιδοποίησης. Στο 5ο Συνέδριο «Η Πληροφορική στην εκπαίδευση». 11-13/10/2013. Πειραιάς: Πανεπιστήμιο Πειραιώς.

Χιονίδου-Μοσκοφόγλου, Μ (2005). Βασικές Μέθοδοι Ομαδο-συνεργατικής Διδασκαλίας και Μάθησης στα Μαθηματικά. [http://users.auth.gr/~lemonidi/sde\\_yliko/Omadosingatik i%20didaskalia%20.pdf](http://users.auth.gr/~lemonidi/sde_yliko/Omadosingatik%20didaskalia%20.pdf)