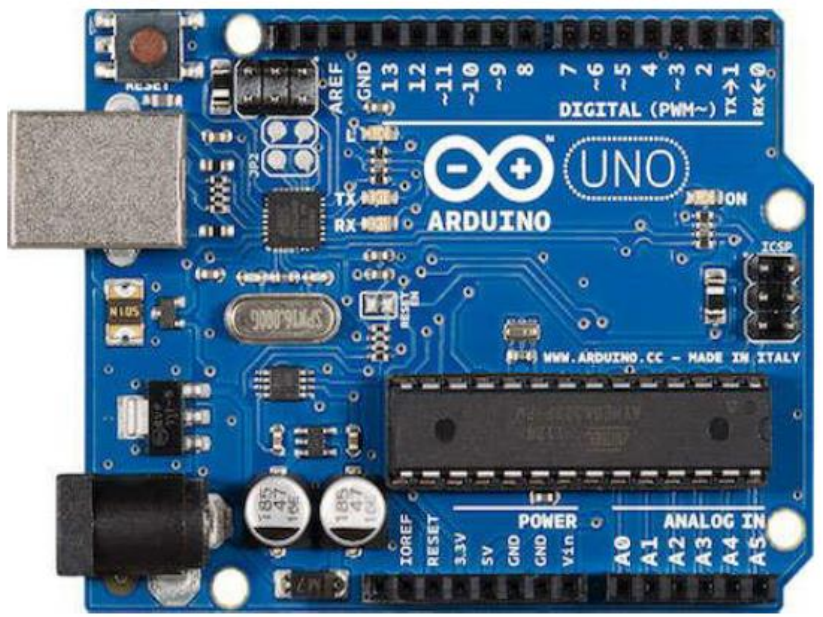
Εξαρτήματα και Βασικές Εντολές

Hardware

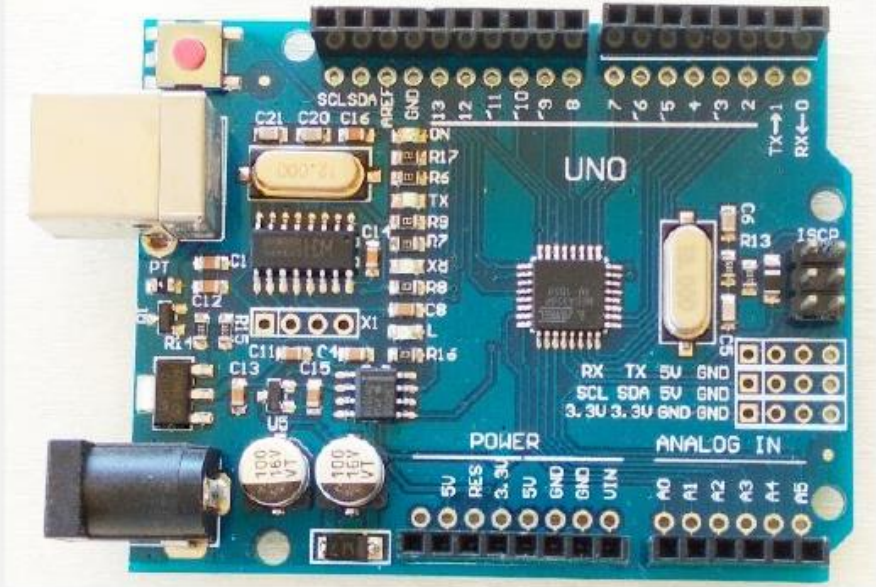
Ο μικροελεγκτής **Arduino** είναι ένας μικροελεγκτής μονής πλακέτας, δηλαδή μια απλή μητρική πλακέτα ανοικτού κώδικα με ενσωματωμένο μικροελεγκτή και εισόδους/εξόδους, η οποία μπορεί να προγραμματιστεί με τη γλώσσα Wiring (είναι η γλώσσα προγραμματισμού C++ με προσθήκη ειδικών βιβλιοθηκών). Το Arduino μπορεί να χρησιμοποιηθεί για την ανάπτυξη κυκλωμάτων και κατασκευών αλλά και να συνδεθεί και να επικοινωνήσει με υπολογιστή μέσω ειδικού λογισμικού.

Σε μία πλακέτα Arduino υπάρχει ένας μικροελεγκτής Atmel AVR (ATmega328 και ATmega168 στις νεότερες εκδόσεις, ATmega8 στις παλαιότερες) και συμπληρωματικά κυκλώματα για την ομαλή λειτουργία της πλακέτας και της επικοινωνίας της με το περιβάλλον. Όλες οι πλακέτες περιλαμβάνουν ένα γραμμικό ρυθμιστή τάσης 5V και έναν κρυσταλλικό ταλαντωτή 32 MHz. Ο μικροελεγκτής είναι από κατασκευής προγραμματισμένος με ένα *bootloader*, έτσι ώστε να μην χρειάζεται εξωτερικός προγραμματιστής.



Σήμερα τα Arduino προγραμματίζονται μέσω USB· αυτό καθίσταται δυνατό μέσω της εφαρμογής προσαρμογέων chip USB-to-Serial. Πρακτικά, συνδέουμε μια USB θύρα του υπολογιστή μας με αυτή του Arduino.

Οι μικροελενκτές της σειράς UNO διαθέτουν 14 ψηφιακές θύρες I/O, εκ των οποίων οι 6 μπορούν να λειτουργήσουν και αναλογικά (~). Επιπλέων, διαθέτει 6 αναλογικές θύρες, 3 θύρες γείωσης 1 θύρα υποδοχής τάσης, 1 θύρα παροχής τάσης 5V και 1 θύρα παροχής τάσης 3.3V.



Αριστερά : Η γνήσια αναπτυξιακή πλακέτα Arduino Uno r3.

Φωτογραφία : https://store.arduino.cc/arduino-uno-rev3

Δεξιά: Ένας από τους κλώνους του Arduino Uno r3.

Είσοδοι και Έξοδοι.

Κάθε ένας από τους 14 ψηφιακούς ακροδέκτες και τους 6 αναλογικούς για το UNO μπορεί να χρησιμοποιηθεί ως είσοδος ή έξοδος εφαρμόζοντας τις εντολές pinMode(), digitalWrite(), και digitalRead(). Οι αναλογικοί ακροδέκτες δοκιμάζουν τα επίπεδα τάσης και αναφέρουν μια ψηφιακή τιμή χρησιμοποιώντας την εντολή analogRead(). Οι αναλογικοί ακροδέκτες και ένα υποσύνολο των ψηφιακών προσφέρουν Pulse Width Modulation (PWM) χρησιμοποιώντας την εντολή analogWrite().

Κάθε ακροδέκτης λειτουργεί σε ηλεκτρική τάση 0-5V DC, μπορεί να παρέχει ή να λάβει ηλεκτρικό ρεύμα με ανώτατο όριο 40mA (συνιστάται 20mA) και έχει εσωτερική αντίσταση pull-up 20-50 kOhms.

Καλώδια.

Υπάρχουν διάφορα είδη, αρσενικά – αρσενικά, θηλυκά – θηλυκά, αρσενικά – θηλυκά. Χωρίς αυτά, η επικοινωνία του μικροελενκτή με τα εξαρτήματα δεν υφίσταται.

Εξωτερικός, 9V-ολτος παροχέας τάσης.

Συνδέουμε μια μπαταρία των 9V επάνω του και χρησιμοποιούμε το κόκκινο καλώδιο σαν παροχή και το μαύρο σαν γείωση.

Απλά LED.

Δίνοντας την επιτρεπόμενη ένταση του ρεύματος, τα λαμπάκια φωτίζονται.

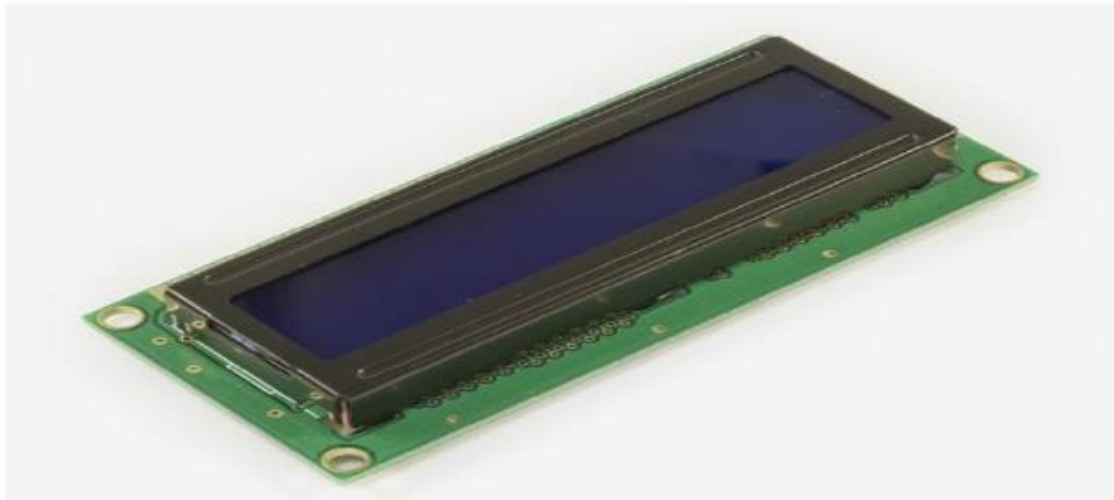
RGB LED.

Πρόκειτα για λαμπάκια τα οποία διαθέτουν 3 χρώματα (κόκκινο -Red, πράσινο -Green, μπλε -Blue). Αναλόγως την τιμή που θα δώσουμε στο κάθε χρώμα, θα εμφανιστεί και το ανάλογο χρώμα.

Αισθητήρας υγρασίας και θερμοκρασίας.

Γενικά υπάρχει μια πληθώρα αισθητήρων που μπορούν να συνδεθούν πάνω στον μικροελενκτή Arduino. Οι αισθητήρες αυτοί είναι υπεύθυνοι για το πιο σημαντικό έργο του Arduino: Την επικοινωνία με το εξωτερικό περιβάλλον.

Οθόνη LED. Οι περισσότερες οθόνες είναι μονόχρωμες ή δίχρωμες. Ο σκοπός τους είναι να απεικονίζουν δεδομένα.

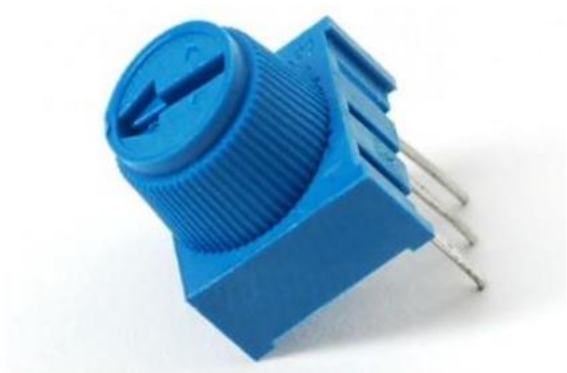


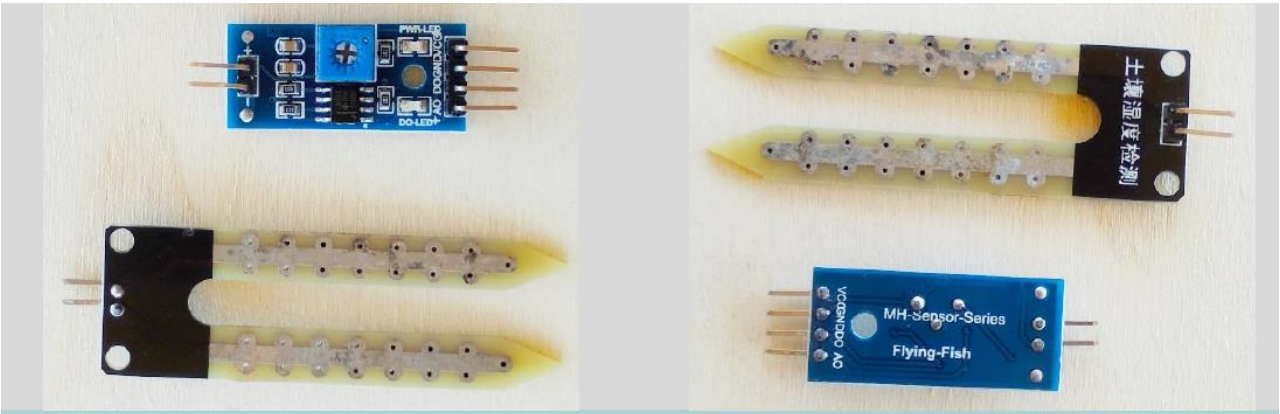
Ηχεία.

Οι συσκευές εξόδου του Arduino για τα ηχητικά σήματα.

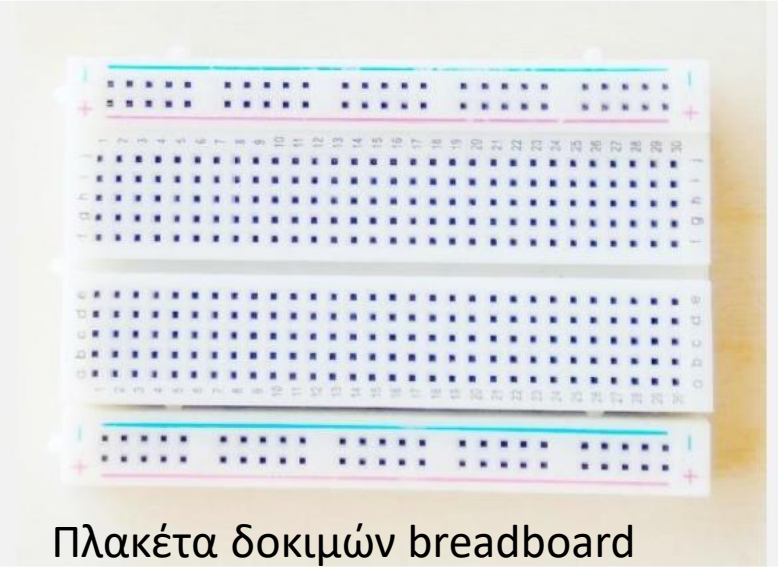
Ποτενσιόμετρο.

Πρόκειται για μια ρυθμιζόμενη μεταβλητή αντίσταση. Εμείς διαλέγουμε το ποσό της αντίστασης που θέλουμε να ασκηθεί στο κύκλωμά μας με απότερο σκοπό να επιτρέψουμε λιγότερη ή περισσότερη ροή ρεύματος.





Ο Αισθητήρας Υγρασίας Εδάφους FC-28.



Πλακέτα δοκιμών breadboard

Προγραμματισμός

Το Arduino προγραμματίζεται μέσα από την ειδική εφαρμογή του. Μπορείτε να την κατεβάσετε απο το επίσημο website: https://www.arduino.cc/en/main/software.

η γλώσσα σύνταξης είναι το Wiring, ουσιαστικά πρόκειται για τη γλώσσα C++ με κάποιες επιπλέον βιβλιοθήκες που κάνουν ευκολότερες τις λειτουργίες I/O. Ο προγραμματιστής αρχικά θα πρέπει να θέσει ποιες θύρες θα χρησιμοποιήσει σαν εισόδους και ποιες σαν εξόδους μέσα στη built-in συνάρτηση setup(). Στη συνέχεια, γράφει το πρόγραμμα του μέσα στη built-in συνάρτηση loop(). Όπως αφήνει να εννοηθεί και το όνομα της, πρόκειται για μια επαναληπτική διαδικασία που σκοπό έχει τη συνεχή επανάληψη του προγράμματος μας όταν αυτό τελειώνει.

Μεταφόρτωση προγράμματος στο Arduino

Προκειμένου να μεταφορτωθεί ένα πρόγραμμα (σκίτσο ή sketch, όπως το ονομάζει το Arduino IDE) σε μια πλακέτα Arduino UNO (όπως αυτή που χρησιμοποιείται στις κατασκευές) μέσω λειτουργικού συστήματος windows, πρέπει να ακολουθηθεί μια απλή διαδικασία βήμα προς βήμα:

1. Συνδέστε το Arduino με το καλώδιο USB.

2. Το τετράγωνο άκρο του καλωδίου USB συνδέστε το με το Arduino και την επίπεδη άκρη συνδέστε τη με μια θύρα USB στον υπολογιστή σας.

3. Εκκινήστε την εφαρμογή Arduino IDE από τον Η/Υ σας.

4. Δημιουργήστε ένα νέο σκίτσο.

Η λογική του Arduino είναι πολύ απλή - στην ουσία υπάρχουν δύο βασικές συναρτήσεις, η setup() και η loop() οι οποίες δουλεύουν ως εξής:

- **setup()** - εδώ βάζουμε όλες τις εντολές που πρέπει να τρέξουν μία φορά, όταν ενεργοποιείται η μονάδα μας (όταν δηλαδή δίνουμε ρεύμα ή όταν πατηθεί το πλήκτρο reset που υπάρχει). Συνήθως μπαίνουν αρχικοποιήσεις τιμών μεταβλητών και οπωσδήποτε ο χαρακτηρισμός των εισόδων/εξόδων που θα χρησιμοποιήσουμε (αν δηλαδή ένα συγκεκριμένο Pin θα είναι είσοδος ή εξοδος).
- **loop()** - εδώ γράφουμε το πρόγραμμά μας. Οι εντολές που υπάρχουν θα τρέξουν κι όταν φτάσει στο τέλος θα ενεργοποιηθεί ξανά η loop(), συνεχίζοντας από την αρχή της, και ξανά. Αυτό θα συμβαίνει συνεχώς, όσο έχει ρεύμα το Arduino ή μέχρι να πατηθεί το πλήκτρο reset.

Έτσι, η βασική λειτουργία του Arduino είναι ότι τρέχει η συνάρτηση setup() μία φορά στην αρχή και ακολούθως η loop() ξανά και ξανά μέχρι να το κλείσουμε (να μην τροφοδοτείται με ρεύμα) ή να πατήσουμε το πλήκτρο reset.

Στην περίπτωση του Reset ξανατρέχει η συνάρτηση setup() μία φορά και ακολούθως η loop() ξανά και ξανά, όπως δηλαδή ακριβώς και όταν αρχικά ενεργοποιείται με ρεύμα ο μικροελεγκτής. Στην περίπτωση που έχουμε κάνει αλλαγές στο πρόγραμμά μας και το φορτώσουμε στον μικροελεγκτή (θα δούμε παρακάτω τη διαδικασία αυτή) αρκεί να πατήσουμε το πλήκτρο Reset ώστε να φορτώσει το πρόγραμμά μας από την αρχή με τον τρόπο που περιγράφηκε.

Οι βασικές εντολές του Arduino είναι:

- `digitalRead(χ)`

Διαβάζει την τιμή μιας ψηφιακής θύρας εξόδου χ, αν είναι HIGH ή LOW

- `digitalWrite(pin, y)`

Θέτει τη θύρα pin σε ψηφιακό HIGH ή LOW

- `pinMode(pin, mode)`

Θέτει τη θύρα pin, σαν είσοδο ή έξοδο. Πάντα τη χρησιμοποιούμε στη συνάρτηση setup().

- `analogRead(x)`

Επιστρέφει την τιμή της τάσης της αναλογικής θύρας

- `analogWrite(pin, value)`

Θέτει στη θύρα pin την τιμή value (0-255)

- `delay(x)`

Βασικά στοιχεία της γλώσσας

Κάθε εντολή τελειώνει με το ελληνικό ερωτηματικό (;). Τα σχόλια είναι οι γραμμές που αγνοεί ο compiler και γράφονται μέσα στα /* αυτά είναι σχόλια */ ή για κάθε σειρά ξεχωριστά

// οτιδήποτε μπροστά από τις δύο γραμμές είναι σχόλια.

Δηλώσεις μεταβλητών

Το κάθε πρόγραμμα για να λειτουργήσει και να έχει μια στοιχειώδη λειτουργικότητα και διαδραστικότητα χρειάζεται να διαθέτει κάποιες μεταβλητές. Τις μεταβλητές τις δηλώνει ο χρήστης στην αρχή του προγράμματος. Κάθε μεταβλητή διαθέτει έναν τύπο και ένα όνομα. Ο τύπος της μεταβλητής καθορίζει το είδος της και μπορεί να είναι χαρακτήρας -char, λέξη -string, ακέραιος -int, δεδκαδικός -float. Για παράδειγμα:

```
char x;
//δηλώνουμε μια μεταβλητή χ, τύπου χαρακτήρα
int x;
//δηλώνουμε μια μεταβλητή χ, τύπου ακεραίου
float  x;
//δηλώνουμε μια μεταβλητή χ, τύπου δεκαδικού
string x;
//δηλώνουμε μια μεταβλητή χ, τύπου λέξης
```

Ο Βρόγχος for

Ο επαναληπτικός βρόγχος for είναι μια εντολή η οποία μας βοηθάει να κάνουμε μια επαναληπτική διαδικασία. Η σύνταξή του είναι η εξής:

for (αρχική τιμή; όσο η αρχική τιμή είναι μικρότερη από κάποια άλλη τιμή ; με το ανάλογο βήμα).

Για παράδειγμα:

```
for (i=0;i<5;i++){
//σώμα εντολών
}
```

Συνθήκη ελένχου if

Είναι ο πιο συχνός και συνηθισμένος έλεγχος. Ελέγχουμε αν ισχύει μια συνθήκη. π.χ

```
if(x>4){
 //σώμα εντολών
}
```

Ένας πιο “advanced” έλεγχος είναι να ελέγξουμε αν ισχύει η συνθήκη και αν δεν ισχύει τότε να εκτελέσουμε κάποιο συγκεκριμένο κώδικα. π.χ.

```
if(x>4){
//σώμα εντολών
}else{
//σώμα εντολών
}
```

Τέλος, η πιο ακραία μορφή της εντολής if είναι η εξής: Ελέγχουμε αν ισχύουν διαφορετικές συνθήκες, αλλιώς υλοποιούμε κάποιο συγκεκριμένο κώδικα. Π.χ.:

```
if(x>3){
     //σώμα εντολών
}
else if(x>5){
     //σώμα εντολών
}
else{
     //σώμα εντολών.
}
```

Βιβλιογραφία

- Για την παρουσίαση χρησιμοποιήθηκε υλικό από:

  - Εκπαιδευτική ρομποτική με τον μικροελεγκτή Arduino. Ένας πρακτικός οδηγός κατασκευών για μαθητές και εκπαιδευτικούς. Νικολάου Α. Φανουράκη (ISBN: 978-618-00-1548-5)
  - Πουλάκης , Ε. (2015). Προγραμματίζοντας με τον μικροελεγκτή Arduino (ISBN: 978-960-93-6760-8)