

Τι είναι το Arduino?



Εξαρτήματα και Βασικές Εντολές

2018

Περιεχόμενα

Arduino	1
Hardware	3
Shields	4
Ελεγκτής moter	4
LAN/ ETHERNET Controller	5
Multifunctional Controller	5
Prototyping Shield	5
Προγραμματισμός	6
Δηλώσεις μεταβλητών, πινάκων.....	8
Ο Βρόγχος for	8
Ο βρόγχος while.....	9
Συνθήκη ελέγχου if	9
Εξαρτήματα.....	10
Δίοδοι.....	11
Αισθητήρας υγρασίας και θερμοκρασίας.....	12
Μοτέρ.....	13
Πυκνωτές.....	13
Photoresistor.....	14
Κουμπί.....	14

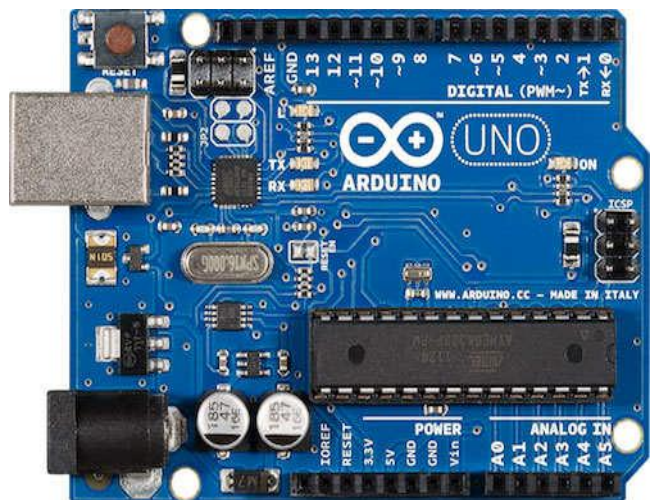
Ο μικροελεγκτής **Arduino** είναι ένας μικροελεγκτής μονής πλακέτας, δηλαδή μια απλή μητρική πλακέτα ανοικτού κώδικα με ενσωματωμένο μικροελεγκτή και εισόδους/εξόδους, η οποία μπορεί να προγραμματιστεί με τη γλώσσα Wiring (είναι η γλώσσα προγραμματισμού C++ με προσθήκη ειδικών βιβλιοθηκών). Το Arduino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

Το 2005 μια ομάδα φοιτητών ξεκίνησε μια προσπάθεια για τη δημιουργία μιας υπολογιστικής πλακέτας, η οποία θα ήταν η πιο φθηνή που θα είχε βγει έως τότε στην αγορά. Οι ιδρυτές Massimo Banzi και David Cueartielles ονόμασαν το σχέδιο από τον Arduino της Inrea και ξεκίνησαν να παράγουν πλακέτες σε ένα μικρό εργοστάσιο στην Ιβρέα, κωμόπολη της επαρχίας Τορίνο στην περιοχή Πεδεμόντιο της βορειοδυτικής Ιταλίας - την ίδια περιοχή στην οποία στεγαζόταν η εταιρία υπολογιστών Olivetti.

Το σχέδιο Arduino είναι μία διακλάδωση της πλατφόρμας Wiring για λογισμικό ανοικτού κώδικα και προγραμματίζεται χρησιμοποιώντας μια γλώσσα βασισμένη στο Wiring (σύνταξη και βιβλιοθήκες), παρόμοια με την C++ με απλοποιήσεις και αλλαγές, καθώς και ένα ολοκληρωμένο περιβάλλον ανάπτυξης (IDE).

Hardware

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 προσαρμογών chip USB-to-Serial. Πρακτικά, συνδέουμε μια USB θύρα του υπολογιστή μας με αυτή του Arduino.

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

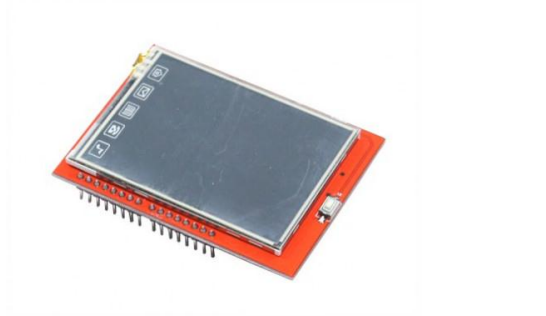
Το Arduino nano και το Arduino-Compatible Bare Bones Board και Boarduino Board ενδέχεται να παρέχουν male header pins στο κάτω μέρος του board προκειμένου να συνδέονται

σε Breadboards. Υπάρχουν πολλά boards συμβατά με και προερχόμενα από Arduino boards. Κάποια είναι λειτουργικά ισάξια με ένα Arduino και μπορεί να χρησιμοποιηθούν εναλλακτικά. Πολλοί είναι το βασικό Arduino με την προσθήκη καινοτόμων output drivers, συχνά για την χρήση σχολικής μόρφωσης για να απλοποιήσουν την κατασκευή buggies και μικρών robot. Άλλες είναι ηλεκτρικά ισάξια αλλά αλλάζουν τον παράγοντα μορφής, επιτρέποντας κάποιες φορές την συνεχόμενη χρήση των Shields ενώ κάποιες όχι. Κάποιες παραλλαγές είναι τελείως διαφορετικοί επεξεργαστές, με ποικίλα επίπεδα συμβατότητας.

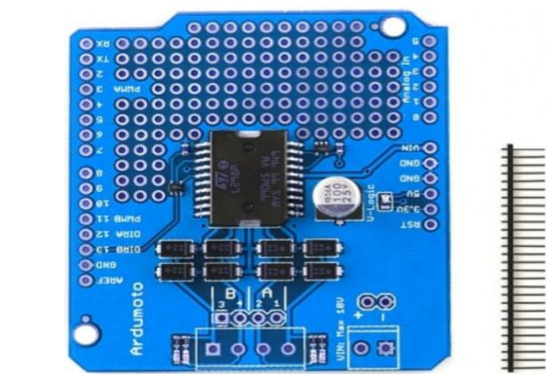
Shields

Τα Arduino και τα Arduino συμβατά boards χρησιμοποιούν την τεχνολογία των shields, τυπωμένων boards επεκτάσεων κυκλωμάτων που συνδέονται στα κανονικά παρεχόμενα Arduino pin-headers. Τα shields μπορούν να παρέχουν έλεγχο στα motors, GPS, Ethernet, LCD εικόνας ή breadboarding (προτυποποίησης). Με αυτόν τον τρόπο μπορούμε να επεκτείνουμε τη λειτουργικότητα της πλακέτας μας, και να δώσουμε τη δυνατότητα στο κύκλωμά μας να υλοποιήσει κάποιες συγκεκριμένες λειτουργίες. Χαρακτηριστικές shields :

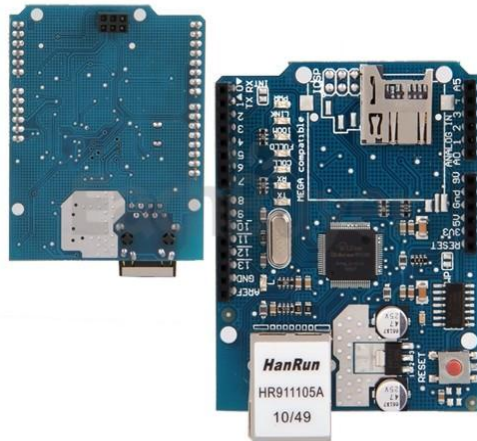
Οθόνη TFT



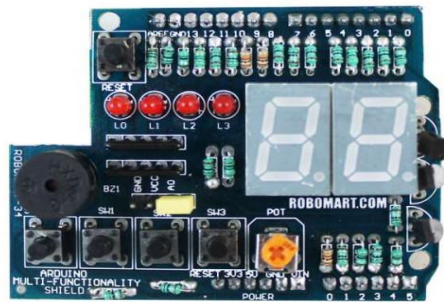
Ελεγκτής moter



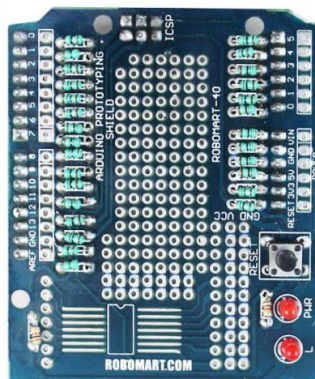
LAN/ ETHERNET Controller



Multifunctional Controller



Prototyping Shield



Προγραμματισμός

Το Arduino προγραμματίζεται μέσα από την ειδική εφαρμογή του. Μπορείτε να την κατεβάσετε από το επίσημο website: <https://www.arduino.cc/en/main/software>. Όπως αναφέρθηκε παραπάνω, 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.

Οι βασικές εντολές του Arduino είναι:

- `digitalRead(x)`

Διαβάζει την τιμή μιας ψηφιακής θύρας εξόδου x , 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)`

Προκαλεί καθυστέρηση στο πρόγραμμα για x ms

- `millis()`

Επιστρέφει το χρόνο από τότε που τρέχει το συγκεκριμένο πρόγραμμα (δεν μπορεί να υπερβεί τις 50 μέρες)

- `abs(x)`

Επιστρέφει την απόλυτη τιμή του x

- `max(x, y)`

Επιστρέφει τον μεγαλύτερο αριθμό μεταξύ του x και y

- `min(x, y)`

Επιστρέφει τον μικρότερο αριθμό μεταξύ του x και y

- `pow(x, y)`

Επιστρέφει το x^y

- `sq(x)`

Επιστρέφει το τετράγωνο του αριθμού x

- `sqrt(x)`

Επιστρέφει την τετραγωνική ρίζα του αριθμού x

- `isAlpha(x)`

Ελέγχει αν το x είναι χαρακτήρας (όχι αριθμός)

- `isAlphaNumeric(x)`

Ελέγχει αν το x είναι αλφαριθμητικό

- `isAscii(x)`

Ελέγχει αν το x ανήκει στον κώδικα ASCII

- `isControl(x)`

Ελέγχει αν είναι χαρακτήρας ελέγχου (/,\,|,κτλ)

- `isDigit(x)`

Ελέγχει αν το x είναι ψηφίο (αριθμός)

- `isGraph(x)`

Ελέγχει αν το x είναι εκτυπώσιμο

- `isLowerCase(x)`

Ελέγχει αν το x είναι πεζό

- `isUpperCase(x)`

Ελέγχει αν το x είναι κεφαλαίο

- `random(min, max)`

Επιστρέφει έναν ψευδοτυχαίο αριθμό μεταξύ min και max

- `Serial.begin(speed)`

Επιτρέπει την επικοινωνία του μικροελεγκτή με την οθόνη του υπολογιστή μας. Προτεινόμενη ταχύτητα τα 9600 bps

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, πινάκων

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, δεκαδικός -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
```

Όσον αφορά τους πίνακες, πρέπει να ξέρεται ότι μπορούμε να δηλώσουμε έναν πίνακα n διαστάσεων, με n έναν οποιοδήποτε αριθμό. Στο παρόν θα δείξουμε πως μπορούμε να δηλώσουμε έναν απλό μονοδιάστατο πίνακα ακεραίων:

```
int foo[4];
```

Δηλώσαμε έναν πίνακα ακεραίων (τον foo) 5 θέσεων (μετράμε και το 0). Σε κάθε θέση μπορούμε να βάλουμε και έναν ακέραιο για να τον αποθηκεύσουμε. π.χ.:

```
foo[0]=30;
foo[3]=2;
foo[1]=7;
foo[2]=88;
foo[4]=9;
```

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  
}
```

Ο βρόγχος for είναι ο σημαντικότερος επαναληπτικός βρόγχος, καθώς είναι και η μόνη εντολή με την οποία μπορούμε να ανατρέξουμε τα στοιχεία ενός ολόκληρου πίνακα.

Παρατίθεται και ένα παράδειγμα προσπέλασης πίνακα:

```
for(i=0;i<5;i++){  
    x= foo[i];  
}
```

Στο παράδειγμά μας χρησιμοποιήσαμε τον πίνακα foo που ορίσαμε στο πάνω υποκεφάλαιο. Προσπελάσαμε όλον τον πίνακα με τη βοήθεια του βρόγχου for και για κάθε θέση του πίνακα, θέταμε την τιμή της στην μεταβλητή x.

Ο βρόγχος while

Ο βρόγχος while είναι άλλος ένας βρόγχος επανάληψης. Ουσιαστικά, η εντολή αυτή επαναλαμβάνει ένα σώμα εντολών όσο ισχύει μια συνθήκη. Για παράδειγμα:

```
while (i<3) {  
    //σώμα εντολών  
}
```

Προσοχή! Τόσο στο βρόγχο while όσο και στον βρόγχο for πρέπει να προσέξουμε να μην υποπέσουμε στο σφάλμα του αέναου βρόγχου. Δηλαδή, θα πρέπει να υπάρχει κάποια εντολή, η οποία κάποια στιγμή να θέτει την ελεγχόμενη συνθήκη σε ψευδή, ώστε να σταματήσει να εκτελείτε ο βρόγχος. Διαφορετικά, θα εκτελείτε ασταμάτητα.

Συνθήκη ελέγχου 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.  
}
```

Εξαρτήματα

Εδώ θα δούμε κάποια χαρακτηριστικά κυκλώματα – εξαρτήματα που μπορούμε να συνδέσουμε στον μικροελεγκτή και να δημιουργήσουμε τα δικά μας κυκλώματα.

Καλώδια.

Υπάρχουν διάφορα είδη, αρσενικά – αρσενικά, θηλυκά – θηλυκά, αρσενικά – θηλυκά. Χωρίς αυτά, η επικοινωνία του μικροελεγκτή 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.



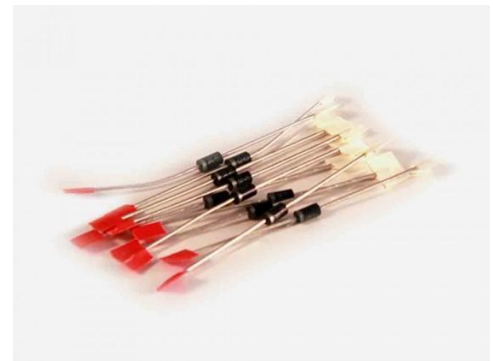
Αντιστάσεις,

1.2 Kohm. Απαραίτητες για την προστασία υπέρτασης των διαφόρων κυκλωμάτων.



Δίοδοι.

Οι διατάξεις αυτές, έχουν ένα και μοναδικό σκοπό: Να επιτρέψουν το ρεύμα να διαρρεύσει από το ένα κύκλωμα στο άλλο.



Micro Sd

Κάρτα αποθήκευσης δεδομένων. Ο σκοπός της είναι προφανής: Αποθηκεύουμε διάφορα δεδομένα που θα θέλαμε κατά τη διάρκεια που τρέχει το πρόγραμμά μας. Σε ποιο προχωρημένες κατασκευές, το Arduino τρέχει ένα λειτουργικό πρόγραμμα μέσα από την κάρτα αποθήκευσης.

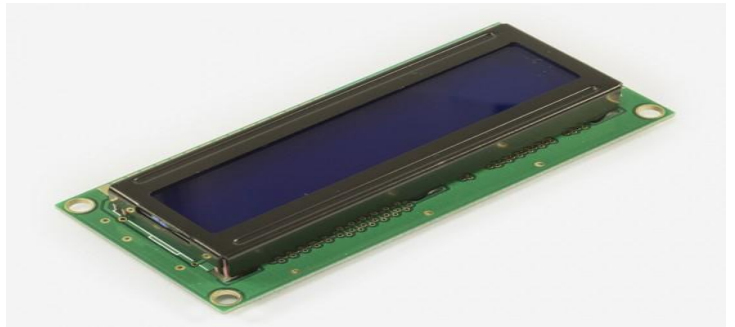


Αισθητήρας υγρασίας και θερμοκρασίας.

Γενικά υπάρχει μια πληθώρα αισθητήρων που μπορούν να συνδεθούν πάνω στον μικροελεγκτή 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.



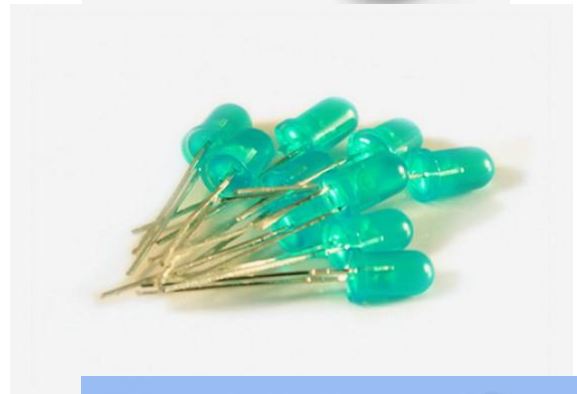
Ηχεία.

Οι συσκευές εξόδου του Arduino για τα ηχητικά σήματα.



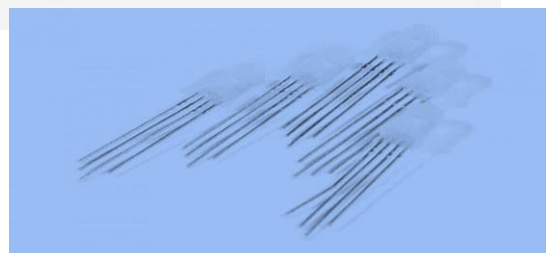
Απλά LED.

Δίνοντας την επιτρεπόμενη ένταση του ρεύματος, τα λαμπάκια φωτίζονται.



RGB LED.

Πρόκειται 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.



Μοτέρ.

Χρησιμεύει για διάφορες κατασκευές, π.χ. Ρόδες, κάτι που να γυρίζει, κτλ



Σέρβο.

Το ίδιο με το μοτέρ, αλλά το σέρβο μπορεί να κάνει μόνο μια κίνηση έως 180 μήρες, απο τα δεξιά προς τα αριστερά ή το αντίστροφο.



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.



Πυκνωτές.

Αυτές οι διατάξεις λειτουργούν σαν μπαταρίες. Συσσωρεύουν ρεύμα και μπορούμε να το χρησιμοποιήσουμε για διάφορες εξεζητημένες περιπτώσεις, όπως π.χ. Για να εξομαλύνουμε την κυμάτωση, κτλ.



Thermistor.

Πρόκειται για διάταξη που αυξομειώνει την αντίσταση αναλόγως τη θερμοκρασία της.



Photoresistor.

Διάταξη που αυξομειώνει την αντίσταση ανάλογα με το φως που εκπέμπεται επάνω της.



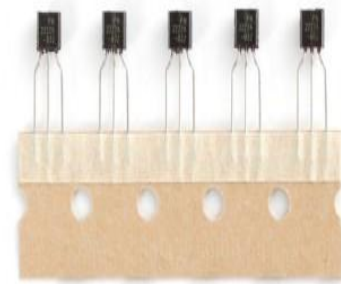
Κουμπί.

Το απλό κουμπάκι λειτουργεί σαν διακόπτης. Αναλόγως, αν είναι ή όχι πατημένο ανοίγει ή κλείνει το κύκλωμα.



Transistors.

Οι διατάξεις αυτές έχουν την ικανότητα να ελέγχουν ένα μεγάλο σε ένταση ρεύμα με μόλις λίγα mA.



Το σεμινάριο αυτό απευθύνεται σε

- Σε εκπαιδευτές Δημοσίων ή Ιδιωτικών Δημοτικών, Γυμνασίων, Λυκείων, ΙΕΚ, ΣΕΚ, ΣΔΔ, ΕΠΑΛ, ΚΔΒΜ κτλ.
- Σε μαθητές, γονείς και όλους όσους ανεξάρτητα από την ειδικότητά και την ηλικία τους επιθυμούν να αποκτήσουν τις απαραίτητες γνώσεις Προγραμματίζοντας με τον μικροελεγκτή Arduino
- Σε όλους όσους επιθυμούν την υιοθέτηση των εκπαιδευτικών διαδικασιών STEM (Science Technology Engineering Mathematics)

Ο Σκοπός του σεμιναρίου "Εισαγωγή στην εκπαιδευτική ρομποτική με Arduino" είναι:

- Η απόκτηση των απαραίτητων γνώσεων Προγραμματίζοντας με τον μικροελεγκτή Arduino.
- Η εισαγωγή των αρχάριων χρηστών στον προγραμματισμό του μικροελεγκτή Arduino.
- Η διευκόλυνση των εκπαιδευτικών που επιθυμούν την υιοθέτηση των διαδικασιών STEM (δάσκαλοι και καθηγητές όχι απαραίτητα Πληροφορικής) με τη διάθεση οργανωμένου εκπαιδευτικού υλικού και την εφαρμογή του στη σχολική τάξη.

Το σεμινάριο αυτό στοχεύει στο να επιτευχθεί:

1. Η εισαγωγή του χρήστη στις βασικές λειτουργίες προγραμματισμού του Arduino.
2. Διευκόλυνση των εκπαιδευτών με διάθεση Ασκήσεων Βήμα-Βήμα ώστε να υλοποιούνται σε ορισμένο χρονικό διάστημα σχολικών εκπαιδευτικών ωρών.
3. Υιοθέτηση της εκπαιδευτικής διαδικασίας STEM (Science Technology Engineering Mathematics) με συγκεκριμένα παραδείγματα.

Το σεμινάριο «Εισαγωγή στην εκπαιδευτική ρομποτική με Arduino» αποτελείται από έξι (6) διδακτικές ενότητες, οι οποίες είναι οι ακόλουθες :

ΠΡΩΤΗ ΔΙΔΑΚΤΙΚΗ ΕΝΟΤΗΤΑ : α) Τι είναι το arduino; β) Η Γλώσσα προγραμματισμού του arduino και οι βασικές εντολές

ΔΕΥΤΕΡΗ ΔΙΔΑΚΤΙΚΗ ΕΝΟΤΗΤΑ : α) Δημιουργία κυκλώματος με φωτοδίοδο. β) Δημιουργία κυκλώματος με ηχείο και κουμπί

ΤΡΙΤΗ ΔΙΔΑΚΤΙΚΗ ΕΝΟΤΗΤΑ : Δημιουργία κυκλώματος με αισθητήρα απόστασης

ΤΕΤΑΡΤΗ ΔΙΔΑΚΤΙΚΗ ΕΝΟΤΗΤΑ : Δημιουργία κυκλώματος έξυπνης κλειδαριάς με δύο ποτενσιόμετρα

ΠΕΜΠΤΗ ΔΙΔΑΚΤΙΚΗ ΕΝΟΤΗΤΑ : Δημιουργία κυκλώματος με LED display

ΕΚΤΗ ΔΙΔΑΚΤΙΚΗ ΕΝΟΤΗΤΑ: α) Δημιουργία κυκλώματος για να ανίχνευση κίνησης β) Δημιουργία κυκλώματος συναγερμού

Ακλουθώντας τις παραπάνω ασκήσεις γίνεται εισαγωγή στις βασικές έννοιες της εκπαιδευτικής ρομποτικής με Arduino και οι χρήστες μπορούν άμεσα να χρησιμοποιήσουν στην πράξη το υλικό αυτό.