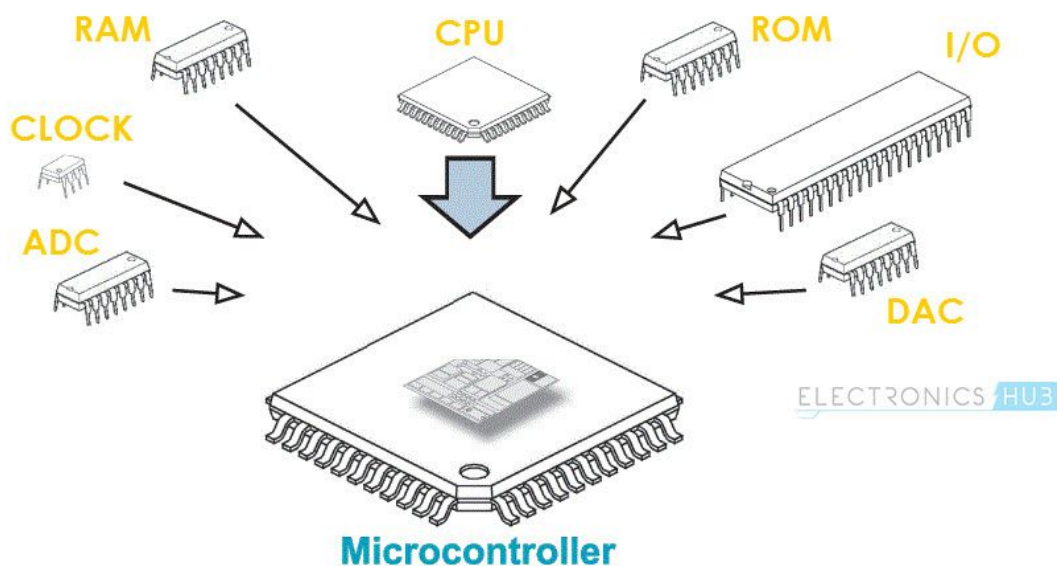


Εισαγωγή στους μικροελεγκτές και τα arduino

Ένας μικροελεγκτής είναι στην ουσία ένας πολύ βασικός υπολογιστής. Είναι απλούστερος και καταναλώνει πολύ μικρότερη ενέργεια από έναν τυπικό Η/Υ, αλλά αποτελείται από τα ίδια βασικά μέρη όπως CPU, μνήμη RAM και είσοδο/έξοδο, με επιπλέον κυκλώματα όπως μετατροπείς αναλογικού-ψηφιακού ρεύματος και άλλα όπως φαίνονται στην παρακάτω εικόνα. Όλα τα παραπάνω βρίσκονται σε ένα και μόνο chip.

Στην απλούστερη δυνατή εξήγηση, οι μικροελεγκτές είναι μικροί υπολογιστές οι οποίοι χειρίζονται και ελέγχουν ηλεκτρονικές συσκευές.



Οι μικροελεγκτές είναι υπολογιστικά συστήματα ειδικού σκοπού, δηλαδή εκτελούν περιορισμένες και συγκεκριμένες λειτουργίες, για αυτό και έχουν πολύ περιορισμένους υπολογιστικούς πόρους αλλά και πολύ μικρή κατανάλωση ενέργειας. Χρησιμοποιούνται σε κάθε είδους αυτοματοποιημένων συσκευών και προϊόντων, και η γνώση της δομής και προγραμματισμού τους είναι πολύτιμη για την κατανόηση και ενασχόληση με τεχνολογίες όπως τα ενσωματωμένα συστήματα και το Διαδίκτυο των Πραγμάτων.

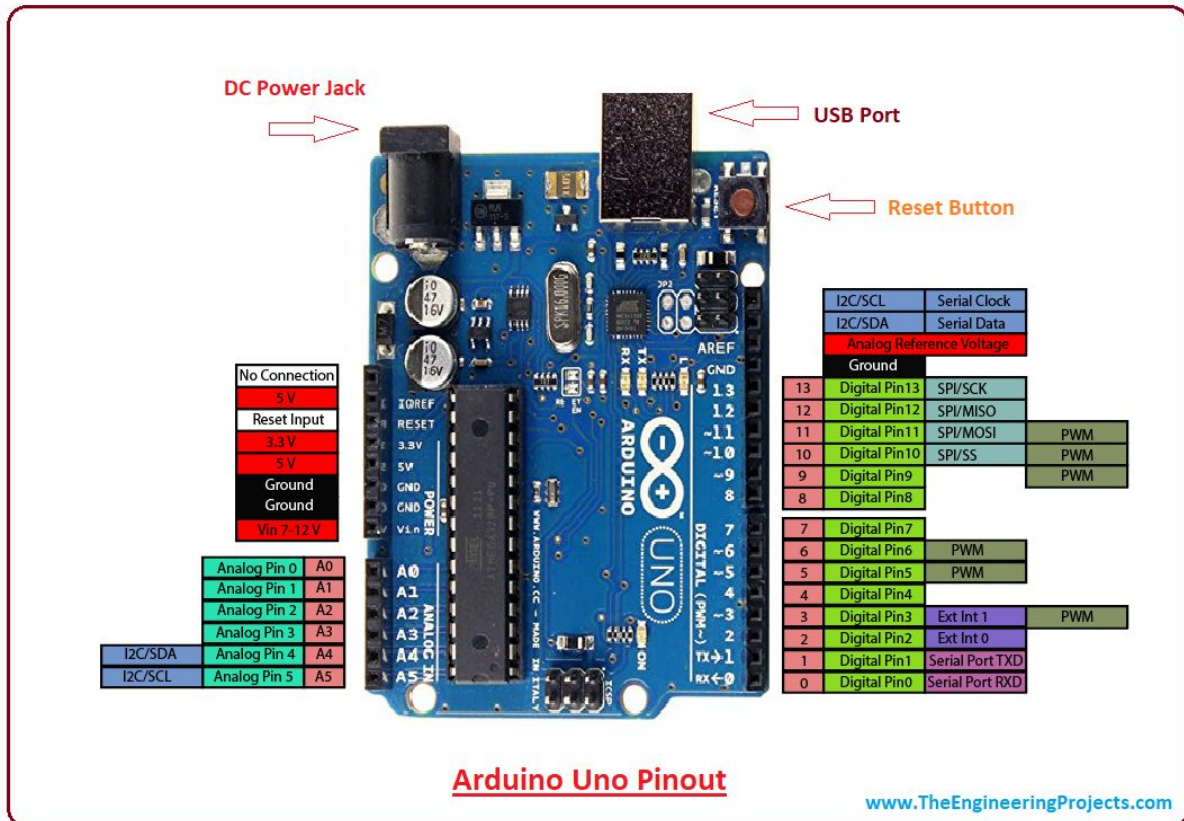
Τα Arduino είναι open source ηλεκτρονικές πλακέτες που περιέχουν έναν μικροελεγκτή καθώς και τις απαραίτητες διασυνδέσεις όπως τα GPIO pins, επιτρέποντας την υλοποίηση αναρίθμητων ηλεκτρονικών συστημάτων και συσκευών. Στο εργαστήριο θα χρησιμοποιήσουμε την πιο δημοφιλή από αυτές τις πλακέτες, το Arduino UNO.

GPIO PINS

Κάθε μικροελεγκτής, και κατ' επέκταση κάθε Arduino Development Board, διαθέτει κάποια GPIO (General Purpose Input-Output) Pins. Αυτά είναι επαφές, πάνω στο τσιπ ή την πλακέτα, των οποίων την τάση ο μικροελεγκτής μπορεί να διαβάσει ή διαμορφώσει ώστε να

αλληλεπιδρά με της περιφερειακές ηλεκτρονικές συσκευές (αυτό άλλωστε τους κάνει και “ελεγκτές”).

Τα pins χωρίζονται σε δύο βασικές κατηγορίες, τα **αναλογικά** και **ψηφιακά**. Όλα τα pins δεν είναι όμοια μεταξύ τους, καθώς μερικά είναι σχεδιασμένα να τελούν και κάποιες επιπλέον λειτουργίες, όπως PWM έξοδο (λεπτομέρειες παρακάτω).



Ξεκινώντας με τα **αναλογικά** pins. Αποτελούν μόνο εισόδους (εξαιρουμένων των A4, A5 στο Uno), και συνδέονται με κύκλωμα ADC (Analog to Digital Converter), δηλαδή μετρούν ψηφιακά την τάση. Συγκεκριμένα, στο Arduino Uno, το κύκλωμα ADC έχει 10-bit ανάλυση (τιμές 0-1023) και κάνει αναφορά στα (περίπου) 5V που είναι η τάση του μικροελεγκτή. Αυτό σημαίνει πως η τάση υπολογίζεται ως εξής:

$$V = (\text{value} / 1023) * 5 \text{ ή γενικότερα}$$

$$V = (\text{value} / \text{maxValue}) * \text{referenceVoltage}$$

Η τιμή της εισόδου (value) δίνεται καλώντας την συνάρτηση `int analogRead(pin);`

Τα **ψηφιακά** pins μπορούν να είναι είτε είσοδος, είτε έξοδος, πράγμα που ορίζουμε καλώντας την συνάρτηση `pinMode(pin, mode);`. Όντας ψηφιακά μπορούν να έχουν δύο μόνο καταστάσεις, HIGH και LOW, όπου HIGH το pin έχει τάση υψηλή (5V για το Arduino Uno) και LOW τάση χαμηλή, δηλαδή 0V.

Άλλα βασικά pins είναι το **5V**, που παρέχει σταθερή τάση 5 Volt, και το **GND** pin που παρέχει γείωση στα κυκλώματά μας (ένα κύκλωμα πρέπει πάντα να επιστρέφει σε γείωση οπότε αυτό το pin θα είναι πρακτικά και αυτό που θα χρησιμοποιούμε περισσότερο).

Βασικές functions:

- void setup()
- void loop()
- void pinMode(pin, mode);
- void digitalWrite(pin, value);
- bool digitalRead(pin);
- int analogRead(pin);

Για λεπτομερή αναφορά στις functions επισκευθείτε:

<https://www.arduino.cc/reference/en/#functions>

setup

Είναι η μία από τις δύο προϋπάρχουσες function κάθε προγράμματος arduino. Η setup καλείται μία και μόνο φορά αμέσως μόλις ξεκινάει το πρόγραμμα και εκτελεί τον κώδικα που της ορίζουμε. Μέσα σε αυτήν εκτός των άλλων ορίζουμε τις ιδιότητες των pins.

loop

Είναι η δεύτερη προϋπάρχουσα function κάθε προγράμματος arduino. Όπως φαίνεται και από το όνομά της, ο κώδικας που γράφουμε μέσα σε αυτήν θα εκτελείται συνεχώς από τη στιγμή που ξεκινάει το πρόγραμμά μας μέχρις ότου τερματιστεί.

pinMode

Η function void pinMode(pin, mode); ορίζει την ιδιότητα των ψηφιακών pins. Αυτά μπορούν να είναι είτε είσοδος είτε έξοδος. Συγκεκριμένα, η παράμετρος pin παίρνει ως τιμή τον αριθμό του pin που θέλουμε να χειριστούμε και η παράμετρος mode τις εξής τρεις τιμές:

OUTPUT, για έξοδο

INPUT, για είσοδο

INPUT_PULLUP, για είσοδο, με το pin να συνδέεται με έναν αντιστάτη 20-50KΩ με τον πόλο των 5V.

Παράδειγμα: pinMode(**11**, OUTPUT); //Ορίζουμε το pin 11 να είναι έξοδος

digitalRead

Με την function bool digitalRead(pin) μας επιστρέφεται η τιμή true ή false, ανάλογα με την τάση εισόδου, εάν αυτή είναι κοντά στα 0V ή 5V αντίστοιχα.

Παράδειγμα: digitalRead(**13**); //Διαβάζουμε την είσοδο του pin 13

analogRead

Η `int analogRead(pin)` επιστρέφει την τάση που εφαρμόζουμε σε ένα `analog pin` με βάση τον κανόνα που αναφέρεται παραπάνω.

Παράδειγμα: `analogRead(3);` *//Διαβάζουμε την είσοδο του αναλογικού pin 3*

digitalWrite

Την έξοδο ενός `pin` με την ιδιότητα `OUTPUT` διαχειριζόμαστε με την function `void digitalWrite(pin, value);`, όπου η παράμετρος `pin` ορίζει το `pin` που διαχειριζόμαστε και η `value` παίρνει τιμή `HIGH` ή `LOW`. Επίσης μπορεί να πάρει τιμές `true/false` ή `0/1`. Έτσι ορίζουμε την τάση τους.

Παράδειγμα: `digitalWrite(11, HIGH);` *//Θέτουμε την έξοδο του ψηφιακού pin 11 σε HIGH*

analogWrite

Παρόμοια λειτουργεί και η `analogWrite(pin, value)`, με διαφορά πως η παράμετρος `value` παίρνει μια ακέραια τιμή 8-bit (0-255). Τη μέθοδο αυτή μπορούμε να την καλέσουμε μόνο για `pins` τα οποία υποστηρίζουν `PWM` έξοδο.

Παράδειγμα: `analogWrite(9, 127);` *//Θέτουμε την έξοδο του ψηφιακού pin με PWM 9 σε 127, παρέχουμε δηλαδή τη μισή τάση*

Το `PWM` (`Pulse Width Modulation`) είναι μία τεχνική με την οποία μπορούμε να παίρνουμε **αναλογική έξοδο χρησιμοποιώντας ψηφιακή είσοδο**, και υποστηρίζεται από μερικά μόνο ψηφιακά `pins` τα οποία φαίνονται παραπάνω.

Αν θέλετε να διαβάσετε πως επιτυγχάνεται η αναπαράσταση αναλογικών τιμών από ψηφιακό σήμα επισκευθείτε: <https://www.arduino.cc/en/tutorial/PWM>