**Φωτοδίοδοι - Leds**
Οι φωτοδίοδοι ή τα leds όπως έχουν κυριαρχήσει, υπάρχουν σε διάφορα χρώματα και τάσεις λειτουργίας. Ανάλογα με την προβλεπόμενη τάση λειτουργίας του led που έχουμε, πρέπει να χρησιμοποιήσουμε μαζί και την κατάλληλη αντίσταση, ώστε να αποφύγουμε κάποια καταστροφή από υπέρταση.

Για να βρούμε την αντίσταση που χρειαζόμαστε αρκεί να θυμηθούμε τον τύπο **R = (Vπ – Vλ) / I**, όπου:

* **R** είναι η αντίσταση που χρειαζόμαστε
* **Vπ**η παρεχόμενη τάση από την πήγή μας (5V από το Arduino)
* **Vλ** η τάση λειτουργίας του led
* **I** το ρεύμα λειτουργίας του led

Τυπικά, μια αντίσταση γύρω στα 220 Ω καλύπτει τα περισσότερα led που έχουμε χρησιμοποιήσει. Τα leds έχουν πολικότητα, δηλαδή δουλεύουν μόνο αν συνδεθούν στην κατάλληλη φορά ρεύματος. Συνήθως το πόδι που πρέπει να συνδεθεί στη θετική κατεύθυνση (+) είναι πιο μακρύ από το αντίστοιχο για την αρνητική φορά (-).

Επιπλέον, το λαμπάκι από την αρνητική μεριά (–) είναι συνήθως επίπεδο κι όχι στρογγυλό όπως είναι από το άλλο πόδι (+). Μην ανησυχείτε όμως, μια ανάποδη σύνδεση δεν θα το καταστρέψει, απλά δεν θα ανάψει στην ανάποδη φορά (εκτός από την ιδιαίτερη περίπτωση να έχουμε δώσει πολύ μεγάλη τάση).

**Ψηφιακές ακίδες (Digital pins)**

Οι ακίδες αυτές στο Arduino μπορούν να ρυθμιστούν είτε ως είσοδοι είτε ως έξοδοι, όμως από προεπιλογή είναι ρυθμισμένες ως είσοδοι. Αξίζει να σημειωθεί, ότι η πλειοψηφία των αναλογικών ακίδων του Arduino (Atmega), μπορεί να ρυθμιστεί και να χρησιμοποιηθεί, με τον ίδιο ακριβώς τρόπο όπως οι ψηφιακές ακίδες. Οι συναρτήσεις ψηφιακής εισόδου και εξόδου είναι οι παρακάτω:

* **pinMode()**: Ρυθμίζει τη συγκεκριμένη ακίδα να συμπεριφέρεται ως είσοδος/ έξοδος.

Σύνταξη: *pinMode(pin, mode)*

Παράμετροι:
**pin**: Ο αριθμός της ακίδας της οποίας η λειτουργία είναι επιθυμητό να αλλάξει.
**mode**: INPUT/OUTPUT

* **digitalWrite()**: Γράφει μια υψηλή (HIGH) ή μια χαμηλή (LOW) τιμή σε μια ψηφιακή ακίδα. Αν η ακίδα έχει ρυθμιστεί ως έξοδος με την συνάρτηση *pinMode()*, τότε η τάση της θα καθορίσει στην αντίστοιχη τιμή: 5V για HIGH και 0V για LOW. Αν η ακίδα έχει ρυθμιστεί ως είσοδος, γράφοντας HIGH στην συνάρτηση *digitalWrite()*θα ενεργοποιήσει μια εσωτερική pullup-αντίσταση των 20 Κ ενώ γράφοντας LOW θα την απενεργοποιήσει.

Σύνταξη: *digitalWrite(pin,value)*

Παράμετροι:
**pin**: Ο αριθμός της ακίδας της οποίας η λειτουργία είναι επιθυμητό να αλλάξει.
**value**: HIGH/LOW

* **digitalRead()**: Διαβάζει την τιμή από μια συγκεκριμένη ψηφιακή ακίδα, που είναι είτε HIGH είτε LOW.

Σύνταξη: *digitalRead(pin)*

Παράμετροι:
**pin**: Ο αριθμός της ακίδας της οποίας η λειτουργία είναι επιθυμητό να αλλάξει.Επιστρέφει: **HIGH**/**LOW**

**Αναλογικές ακίδες εισόδου (Analog input pins)**

Οι ελεγκτές Atmega που χρησιμοποιούνται για την πλατφόρμα Arduino περιέχουν έναν ενσωματωμένο αναλογικό-σε-ψηφιακό μετατροπέα 6 καναλιών. Ο μετατροπέας διαθέτει ανάλυση 10 bits, επιστρέφοντας ακέραιους από 0 έως 1023. Ενώ η κύρια λειτουργία της αναλογικής ακίδας για τους περισσότερους χρήστες Arduino είναι να διαβάζει αναλογικούς αισθητήρες, οι αναλογικές ακίδες έχουν επίσης όλες τις λειτουργίες των γενικών ακίδων εισόδου/εξόδου.

Οι συναρτήσεις αναλογικής εισόδου και εξόδου είναι οι παρακάτω:

* **analogWrite()**: Γράφει μια αναλογική τιμή (PWM κύμα) σε μια ακίδα. Μπορεί να χρησιμοποιηθεί, για παράδειγμα να ανάψει ένα LED σε διάφορες φωτεινότητες ή να οδηγήσει έναν κινητήρα σε διάφορες ταχύτητες. Μετά από μια κλήση της *analogWrite()*, η ακίδα θα δημιουργήσει ένα σταθερό τετραγωνικό κύμα του καθορισμένου κύκλου λειτουργίας μέχρι την επόμενη κλήση της *analogWrite()* (ή μια κλήση της *digitalWrite()*ή *digitalRead()* για την ίδια ακίδα). Η συχνότητα του σήματος PWM είναι περίπου 490 Hz. Στις περισσότερες πλατφόρμες Arduino η συνάρτηση αυτή λειτουργεί στις ακίδες 3, 5, 6, 9, 10, 11.

Σύνταξη: *analogWrite(pin, value)*

Παράμετροι:
**pin**: Ο αριθμός της ακίδας της οποίας θα γράψει επάνω.
**value**: ο κύκλος λειτουργίας μεταξύ 0 και 255.

* **analogRead()**: Διαβάζει την τιμή από την καθορισμένη αναλογική ακίδα.

Σύνταξη: *analogRead(pin)*

Παράμετροι:
**pin**: Ο αριθμός της αναλογικής ακίδας εισόδου από όπου θα διαβάζει. Επιστέφει: **ακέραιο από 0 έως 1023**.

**2η Εργασία : Έλεγχος πυλών εξόδου**

Σε αυτήν της εργασία καλούμαστε μέσα από την πιο συχνή εργασία ever, να συνδέσουμε ενεργοποιητές στις πύλες και στη συνέχεια να τους ελέγξουμε μέσα από τον δικό μας αλγόριθμο.

Ποιο συγκεκριμένα μετά την πρόσληψη μας από εταιρία κατασκευής φαναριών ελέγχου κυκλοφορίας μας ζητείται να φτιάξουμε ένα με το arduino. ( Το γνωστό φανάρι στους δρόμους, χωρίς αυτά των πεζών, μόνο κόκκινο, πορτοκαλί, πράσινο. Όποιος νιώθει δημιουργικός μπορεί σε δεύτερη φάση να το δοκιμάσει και με φανάρια πεζών χωρίς όμως να το ζητά η εργασία.)

* Δημιουργήστε το cirquit "**Edifanari Cyclades \_ username**" στο Τinkercad όπου username το username σας π.χ.  **Edifanari Cyclades \_ tsiastoudis**
* Εισάγετε ένα κόκκινο, ένα πορτοκαλί και ένα πράσινο led και **ένα** Arduino. Στην πραγματικότητα θα εισάγεται τρία κόκκινα και στη συνέχεια θα αλλάξετε το χρώμα των δύο, από την καρτέλα του που εμφανίζετε κάθε φορά που επιλέγετε ή εισάγετε κάποιο. (Τόσο λίγο ψάξιμο δεν είναι κακό)
* Στη συνέχεια δημιουργήστε τρεις ξεχωριστές συνδέσεις με τρεις αντιστάσεις σε τρεις εξόδους, τρία όμοια κυκλώματα με αυτό που δείξαμε στο βίντεο σε ένα Arduino΄
* Τέλος δημιουργήστε ένα κώδικα ελέγχου ώστε στην αρχή να έχουμε ανοιχτό το κόκκινο led και σβηστά τα άλλα, κ.ο.κ ώστε να προσομοιάσουμε ένα φανάρι κυκλοφορίας.\*
* **Ας υποθέσουμε ώστε να μην έχουμε χρονοβόρα διαδικασία ελέγχου των παραδοτέων ότι** στο φανταστικό φανάρι μας το κόκκινο διαρκεί 3 δευτερόλεπτα το πορτοκαλί 1 και το πράσινο 3.
* Στη συνέχεια δημοσιεύστε το και αντιγράψτε το link όπως δείξαμε Τέλος υποβάλετε το link του έργου σας για έλεγχο για να δούμε αν λειτουργεί σωστά.

\* Το συνηθέστερο λάθος είναι ...........  η σειρά ενεργοποίησης των φαναριών ! ! !

Καλή επιτυχία!