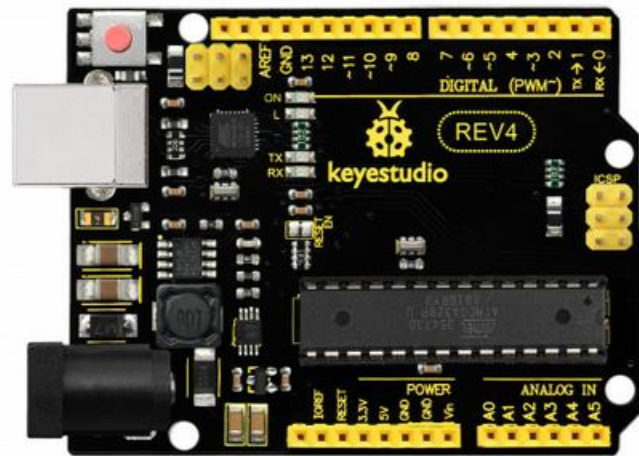


3.4 Ο μικροελεγκτής Arduino

Ο μικροελεγκτής συνιστά ένα ολοκληρωμένο κύκλωμα, στο οποίο ενσωματώνονται ο μικροεπεξεργαστής, η μνήμη και οι περιφερειακές διατάξεις. Είναι ουσιαστικά ένας μικρός υπολογιστής με περιορισμένες δυνατότητες και πόρους συστήματος (μικρή υπολογιστική ισχύς, μνήμη), ο οποίος έχει σχεδιαστεί σε ένα και μόνο ολοκληρωμένο σύστημα. Συνήθως ένας μικροελεγκτής είναι φτιαγμένος για να επιτελεί συγκεκριμένες λειτουργίες. Οι μικροελεγκτές χρησιμοποιούνται ευρύτατα, π.χ. στις μεταφορές (αεροπλάνα, αυτοκίνητα), στα ιατρικά μηχανήματα, σε αναρίθμητες συσκευές, στο IoT. Εμείς θα χρησιμοποιήσουμε το Plus Starter Kit for Arduino το οποίο είναι μια πλακέτα η οποία αποτελείται από τον μικροελεγκτή ATMEGA328P-AU, με Clock speed 16MHz.

Επίσης έχει και τα ακόλουθα χαρακτηριστικά: EEPROM: 1 KB, SRAM: 2 KB, Flash Memory: 32KB. Σημειώνεται ότι η τάση λειτουργίας ορίζεται στα DC 5V ή στα 3.3V, το οποίο καθορίζεται από διακόπτη/dip switch. Αποτελείται από **ψηφιακές θύρες εισόδου/εξόδου I/O (2 – 13)**, καθώς και από **αναλογικές (analog input)**. Επίσης, με το σύμβολο **GND** αναφερόμαστε στη γείωση ή αλλιώς στον αρνητικό πόλο ενός κυκλώματος ή εξαρτήματος. Το κόκκινο πλήκτρο εκτελεί τη λειτουργία **reset** του μικροεπεξεργαστή εφόσον το πατήσουμε. Η πλακέτα ενσωματώνει και **USB type C connector** για σύνδεση του kit με τον υπολογιστή, μέσω καλωδίου USB.

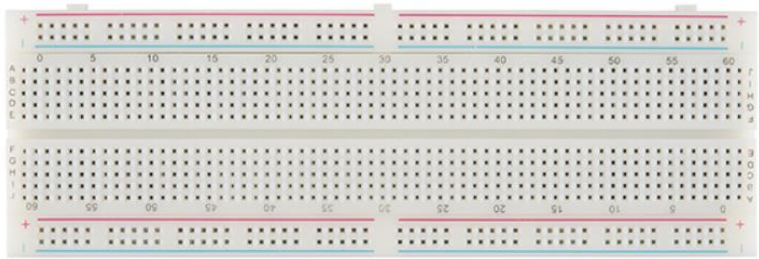


Εικόνα 3.10. Το Plus Starter Kit for Arduino (Keyestudio REV4)

3.4.1 Το breadboard

Για να χρησιμοποιήσουμε την πλακέτα χρειαζόμαστε έναν εξωτερικό πίνακα, **το breadboard**, πάνω στον οποίο θα αναπτύσσουμε τα κυκλώματά μας (θα συνδέουμε αντιστάσεις, λαμπάκια, πλήκτρα κ.ά.). Ονομάζεται έτσι καθώς πολύ παλιά τα ηλεκτρονικά κυκλώματα ήταν ογκώδη και όποιος ήθελε να φτιάξει ένα κύκλωμα έπαιρνε την πρώτη «**σανίδα**» που έβρισκε μπροστά του και αυτή ήταν συνήθως το ξύλο που κόβουμε το ψωμί! Έτσι έμεινε η ονομασία αυτή. **Το breadboard είναι ένας πίνακας ανάπτυξης κυκλωμάτων, ο οποίος είναι γεμάτος οπές.** Εσωτερικά έχει οριζόντιες και κάθετες συνδέσεις των οπών. Συγκεκριμένα στο επάνω και κάτω μέρος του, εντοπίζουμε δύο οριζόντιες γραμμές από οπές, η μία είναι κόκκινη και η άλλη μπλέ. Αυτές οι οπές επικοινωνούν οριζόντια, όπως δείχνουν οι γραμμές, και μάλιστα μέχρι εκεί που διακόπτονται οι χρωματιστές γραμμές. Η κόκκινη γραμμή συμβολίζει το ρεύμα, δηλαδή την τροφοδοσία, ενώ η μπλε συμβολίζει τη γείωση. Συνεπώς, το πρώτο πράγμα που κάνουμε στον πίνακά μας είναι να δώσουμε σε μια **κόκκινη γραμμή τροφοδοσία (Vcc)** και στην αντίστοιχη **μπλέ τη γείωση (GND)**, ώστε να μπορούμε να μοιράσουμε εύκολα αυτά τα δύο συστατικά στις συνδέσεις του κυκλώματος.

Παράλληλα, παρατηρούμε και τις κάθετες γραμμές των οπών, οι οποίες επικοινωνούν κατακόρυφα ανά πέντε (π.χ. οι στήλες A,B, C, D, E και F,G,H,I,J). Στη μέση του breadboard παρατηρούμε μια εσοχή. Αυτή δείχνει την αλλαγή τμήματος στον πίνακα.



Εικόνα 3.11. Το breadboard

Δηλαδή οι επάνω κατακόρυφες στήλες δεν επικοινωνούν με τις κάτω κατακόρυφες στήλες. Αυτό είναι πολύ χρήσιμο, όταν έχουμε να συνδέσουμε ολοκληρωμένα κυκλώματα ή ακόμη και πλήκτρα με επαναφορά (push buttons). Έτσι συνδέουμε τα επάνω ποδαράκια (pin) του ολοκληρωμένου κυκλώματος στις επάνω οπές και τα κάτω pin του ολοκληρωμένου κυκλώματος στις κάτω οπές.

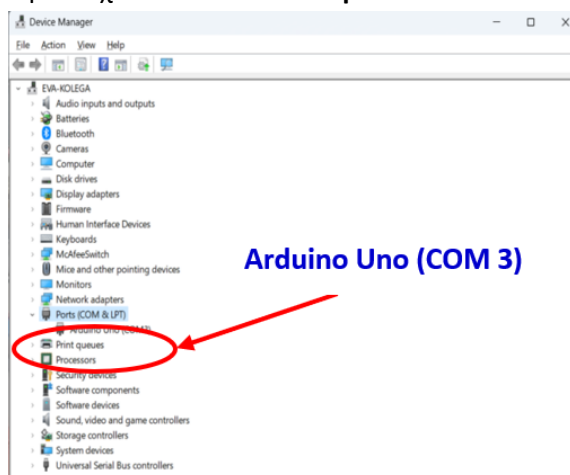


Δραστηριότητα 1

1. Παρατηρήστε το breadboard και συζητήστε στην ομάδα σας σχετικά με τις εσωτερικές συνδέσεις του.
2. Παρατηρήστε την πλακέτα **Plus Starter Kit for Arduino** και αναγνωρίστε τις ψηφιακές θύρες, τη γείωση, το reset button και την USB διεπαφή.
3. Συνδέστε την πλακέτα με το καλώδιο USB στην αντίστοιχη θύρα του ηλεκτρονικού υπολογιστή.

3.4.2 Διαδικασία Σύνδεσης της πλακέτας με τον υπολογιστή

Αρχικά συνδέουμε το kit μέσω καλωδίου USB στην USB θύρα του υπολογιστή. Στη συνέχεια θα πρέπει να κατεβάσουμε και να εγκαταστήσουμε το ολοκληρωμένο περιβάλλον ανάπτυξης του **Arduino - (Arduino Integrated Development Environment - IDE)**. Από τη σελίδα <https://www.arduino.cc>. επιλέγουμε το **μενού Software** και στη συνέχεια το **Download Options**.



Εικόνα 3.12. Η διαχείριση συσκευών των Windows



Εικόνα 3.13. Το Arduino IDE

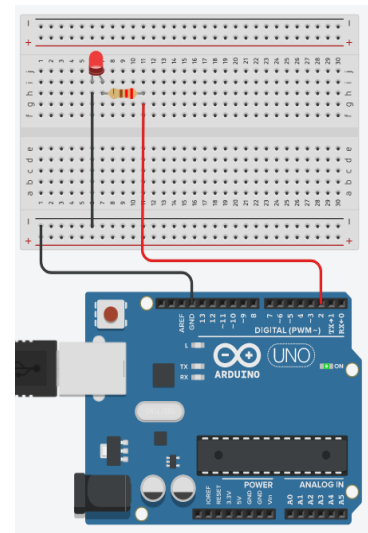
Εκτελούμε το πρόγραμμα που μόλις κατεβάσαμε και πραγματοποιούμε κανονικά όλα τα βήματα εγκατάστασης. Επίσης κατά την εγκατάσταση ορίζουμε ότι όλοι οι χρήστες θα μπορούν να χρησιμοποιούν αυτό το πρόγραμμα. Ολοκληρώνεται η εγκατάσταση και πατάμε **finish**. Στη συνέχεια στον **device manager** θα πρέπει να δούμε ότι στα Ports (COM and LPT) μάς βγάζει το Arduino Uno στην COM 3. Στην περίπτωση μας, ο driver έχει εγκατασταθεί στη σειριακή θύρα (COM Port 3).

Στη συνέχεια είμαστε έτοιμοι να εκκινήσουμε το πρόγραμμα **Arduino IDE**. Σημειώνεται ότι στο περιβάλλον αυτό **ξεκινάμε** με **δύο βασικές συναρτήσεις**, την **setup()** και την **loop()**. Στη setup έχουμε την αρχικοποίηση των παραμέτρων του κυκλώματος και στη loop έχουμε την ατέρμονη επανάληψη εντολών, αυτές δηλαδή που θέλουμε να εκτελούνται συνέχεια. Η γλώσσα προγραμματισμού που χρησιμοποιούμε είναι η C σε βασικό επίπεδο. Σε επόμενη φάση μπορούμε να φτιάξουμε κι εμείς τις δικές μας συναρτήσεις.

3.4.3 Η πρώτη μου εφαρμογή

Ένα απλό παράδειγμα είναι η ενεργοποίηση μιας λυχνίας - **LED**. Πριν προγραμματίσουμε το κύκλωμα, θα πρέπει να το υλοποιήσουμε. Συνεπώς, συνδέουμε μία αντίσταση 220Ω και στη συνέχεια ένα λαμπάκι οποιουδήποτε χρώματος επιθυμούμε. Η χρήση αντίστασης είναι υποχρεωτική, ώστε να μην καεί το λαμπάκι. Σημειώνεται ότι το (+) στο λαμπάκι είναι το μακρύ ποδαράκι, η **άνοδος**, σε αντίθεση με το (-), την **κάθοδο**, το οποίο είναι το κοντό. Δηλαδή, μακρύ ποδαράκι - άνοδος στην τροφοδοσία (+), κοντό ποδαράκι - κάθοδος στην γείωση (-).

Συγκεκριμένα χρησιμοποιούμε την εντολή/συνάρτηση **pinMode (pinID, INPUT/OUTPUT)**, ώστε να καθορίσουμε την κατάσταση ή λειτουργία του pin, αν θα είναι δηλαδή για έξοδο από τον υπολογιστή ή για είσοδο στον υπολογιστή.



Εικόνα 3.14. Κύκλωμα με ένα Λαμπάκι - LED

Ορίζουμε λοιπόν ότι το pin 2 θα είναι για έξοδο – OUTPUT. Στη συνέχεια χρησιμοποιούμε την εντολή/συνάρτηση **digitalWrite(pinID, value)** με την οποία στέλνουμε στο εν λόγω pin την τιμή που επιθυμούμε, δηλαδή HIGH ή LOW. Όταν στείλουμε HIGH, το λαμπάκι ανάβει και όταν στείλουμε LOW, το λαμπάκι σβήνει. Στη συνέχεια με την εντολή **delay(ms)** ορίζουμε το χρονικό διάστημα καθυστέρησης σε ms. Εκτελούμε το πρόγραμμα και παρατηρούμε ότι το λαμπάκι ανάβει για ένα δευτερόλεπτο και σβήνει επίσης για ένα δευτερόλεπτο. Η λειτουργία αυτή βρίσκεται μέσα σε loop, πράγμα που σημαίνει ότι εκτελείται διαρκώς.

```
void setup() {  
  // put your setup code here,  
  // to run once:  
  pinMode(2, OUTPUT); // sets I/O  
  // PIN to "output"  
  digitalWrite(2, LOW); //sets I/O  
  // PIN to "low"  
  delay(200); // delay  
}  
  
void loop() {  
  // put your main code here,  
  // to run repeatedly:  
  digitalWrite(2, HIGH);  
  delay(1000); // wait for 1000ms  
  digitalWrite(2, LOW);  
  delay(1000); // wait for 1000ms  
}
```

Εικόνα 3.15. Κώδικας για ένα λαμπάκι



Δραστηριότητα 2

Υλοποιήστε το παραπάνω κύκλωμα στο breadboard της ομάδας σας, συνδέστε το με την πλακέτα Arduino και γράψτε τον αντίστοιχο κώδικα. Εκτελέστε το πρόγραμμα. Τι παρατηρείτε;

3.4.4 Τρεχαντήρι με δύο λαμπάκια

Έστω τώρα ότι θέλουμε να δημιουργήσουμε ένα **τρεχαντήρι με δύο λαμπάκια**, δηλαδή να ανάβει το πρώτο για ένα χρονικό διάστημα, να σβήνει και να ανάβει το επόμενο. Σημειώνεται ότι για τις καταστάσεις ON και OFF θα χρησιμοποιήσουμε το ίδιο χρονικό διάστημα, για να υπάρχει ομοιομορφία. Υλοποιούμε το κύκλωμα που φαίνεται στην εικόνα 3.3.8.