4ο Τρίωρο

Εισαγωγή στο Arduino

Το *Arduino* είναι μια μητρική πλακέτα υλικού (Hardware), ανερχόμενο τα τελευταία χρόνια στον χώρο της εκπαίδευσης. Χρησιμοποιείται σε ποικίλες εκπαιδευτικές δραστηριότητες και μαθητικούς διαγωνισμούς, για την μύηση των μαθητών στις αρχές του προγραμματισμού, της ηλεκτρονικής και της μηχανικής. Το *Arduino* παρέχει τη δυνατότητα στους μαθητές να δουν την εφαρμογή του προγραμματισμού σε ρεαλιστικές καταστάσεις και να κατανοήσουν ότι μέσω αυτού, έχουν τον κύριο έλεγχο της λειτουργίας ρομποτικών συστημάτων και κατασκευών. Μετά από κάθε κώδικα που σχεδιάζουν οι μαθητές, βλέπουν απτά και άμεσα αν εφαρμόζεται στην πράξη, εντοπίζοντας εύκολα τα λάθη που έκαναν για να τα διορθώσουν.

Το *Arduino* ενδείκνυται για εκπαιδευτικές εφαρμογές για τους παρακάτω λόγους:

- Έχει Χαμηλό Κόστος
- Είναι απλό, μέσα σε λίγες ώρες ο άπειρος χρήστης μπορεί να δημιουργήσει την πρώτη του κατασκευή
- Είναι Ανοικτού κώδικα (υλικό και λογισμικό)
- Είναι ιδανικό για Ιδιοκατασκευές εναλλακτικών, καινοτόμων πειραματικών διατάξεων σε σχέση με αυτές του εμπορίου
- Έχει πλούσιο διαδικτυακό υλικό υποστήριξης
- Υπάρχουν ειδικά forum όπου οι χρήστες μοιράζονται την εμπειρία και την εργασία τους με άλλους εκπαιδευτικούς, ερευνητές και χομπίστες
- Διαθέτει πολλαπλών ειδών αισθητήρες με χαμηλό κόστος
- Διαθέτει εύκολους τρόπους απεικόνισης των δεδομένων που λαμβάνονται από τους αισθητήρες
- Είναι μια πολύ οικονομική λύση για Εκπαιδευτική Ρομποτική, με χαμηλού κόστους αναλώσιμα
- Διαθέτει πολλές εκδόσεις με κάθε έκδοση να καλύπτει διαφορετικές ανάγκες
- Διαθέτει πλακέτες επέκτασης (*shield*) που δίνουν νέες δυνατότητες στις πλατφόρμες του *Arduino*
- Το *Arduino IDE* εκτελείται σε λογισμικά *Linux*, *Windows* και *Mac*

Οι Εφαρμογές *Arduino* που θα βρείτε στον παρόντα οδηγό, είναι ένας συνδυασμός θεωρίας και πράξης, όπου τα παιδιά διδάσκονται τις βασικές εντολές του κώδικα με εφαρμογές διαβαθμισμένης δυσκολίας και μαθαίνουν παράλληλα την βασική συνδεσμολογία μεταξύ των ηλεκτρονικών εξαρτημάτων. Παράλληλα με τις Εφαρμογές του *Arduino* οι μαθητές καλούνται να κατασκευάσουν μια έξυπνη πόλη, μια δραστηριότητα που θα υλοποιηθεί σε τρία τρίωρα και θα εμπλουτίζεται σταδιακά σχετικά συνδεδεμένες θεματικά με το εκάστοτε τρίωρο, για να μπουν στη λογική της υλοποίησης των ιδεών τους, μέσω της διαδικασίας «κατασκευάζω κάτι από το μηδέν και το κάνω να ζωντανέψει προγραμματίζοντας το να κάνει αυτό που έχω στο μυαλό μου».

Μητρική πλακέτα Arduino UNO

Το Arduino είναι μία ολοκληρωμένη πλατφόρμα ανάπτυξης έργων ηλεκτρονικής, αυτοματισμού και ρομποτικής, η οποία παρέχει ελεύθερα το αναγκαίο λογισμικό και υλικό. Το λογισμικό της πλατφόρμας είναι το Arduino IDE. Το υλικό περιλαμβάνει μία σειρά από πλακέτες μικροελεγκτή με διαφορετικά χαρακτηριστικά, ώστε ο χρήστης να μπορεί να επιλέξει την έκδοση που ταιριάζει καλύτερα στην εκάστοτε εφαρμογή του.

Στην ενότητα αυτή θα παρουσιαστούν το Arduino UNO και το Breadboard, που αποτελούν τα βασικά εξαρτήματα για όλες τις εφαρμογές που θα αναπτυχθούν στη συνέχεια. Το UNO είναι η βασική έκδοση πλακέτας της πλατφόρμας Arduino.

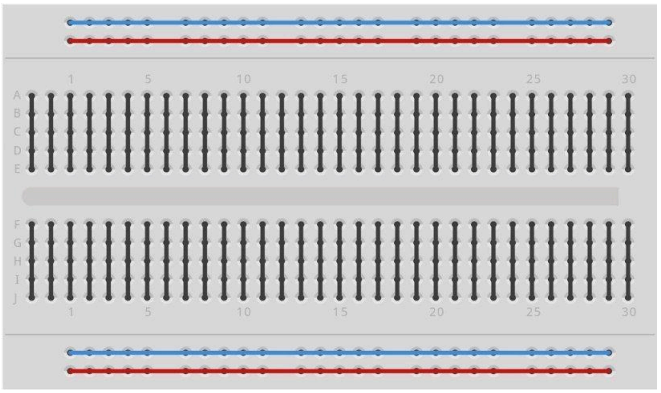
Εικόνα 1 Arduino UNO

Στην Εικόνα 1, στο κόκκινο πλαίσιο περικλείονται οι ψηφιακοί ακροδέκτες εισόδου/εξόδου, στο πράσινο οι ακροδέκτες αναλογικής εισόδου και στα πορτοκαλί οι ακροδέκτες τροφοδοσίας. Στα αριστερά της εικόνας μπορούμε να διακρίνουμε στο πάνω μέρος την υποδοχή USB που χρησιμοποιείται για φόρτωση προγράμματος, για σειριακή επικοινωνία με τον υπολογιστή αλλά και για τροφοδοσία (κίτρινο πλαίσιο). Στο κάτω μέρος υπάρχει η υποδοχή για την εξωτερική τροφοδοσία 7 - 12 V (κίτρινο πλαίσιο).

Breadboard

Το breadboard είναι μία διάταξη που επιτρέπει την εύκολη κατασκευή κυκλωμάτων χωρίς να απαιτούνται κολλήσεις. Συγκεκριμένα, το breadboard διαθέτει οπές πάνω στις οποίες μπορούν να συνδεθούν διάφορα ηλεκτρικά και ηλεκτρονικά στοιχεία. Εσωτερικά οι οπές αυτές διασυνδέονται μεταξύ τους όπως φαίνεται στην Εικόνα 2.

Οι 4 εξωτερικές σειρές οπών (2 πάνω και 2 κάτω) είναι συνδεδεμένες οριζόντια, ενώ στο κεντρικό του τμήμα οι διασυνδέσεις είναι κατακόρυφες, με κάθε στήλη να χωρίζεται σε δύο ανεξάρτητα τμήματα των 5 οπών το κάθε ένα (5άδες οπών).



Εικόνα 2 Breadboard

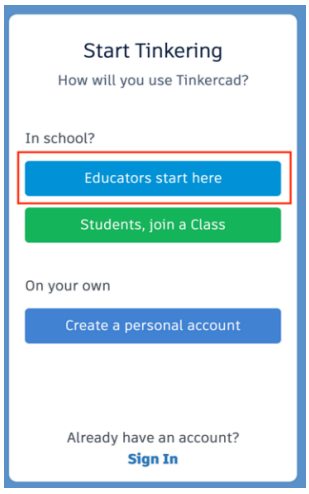
Έτσι, αν θέλουμε για παράδειγμα να ενώσουμε δύο καλώδια, αντί να τα κολλήσουμε με το κολλητήρι, αρκεί να τοποθετήσουμε από ένα άκρο τους σε δύο συνδεδεμένες οπές (π.χ. στην ίδια μισή στήλη). Οι οριζόντιες εξωτερικές σειρές, χρησιμοποιούνται συνήθως σε σύνθετες εφαρμογές, για να παρέχουμε τροφοδοσία και γείωση σε πολλά εξαρτήματα ταυτόχρονα.

2. Εισαγωγή στο TinkerCAD

Τι είναι το Tinkercad

Το Tinkercad είναι ένα λογισμικό τρισδιάστατου σχεδιασμού, το οποίο παρέχει τη δυνατότητα δημιουργίας projects (3D design και modeling) σε μια ποικιλία χρηστών, από αρχάριους μέχρι προχωρημένους. Στο πλαίσιο του Generation Next, θα το χρησιμοποιήσουμε ως περιβάλλον προσομοίωσης για τα project του Arduino. Το συγκεκριμένο εργαλείο μπορεί να χρησιμοποιηθεί είτε σε δια ζώσης διδασκαλία είτε σε διαδικτυακή.

Είσοδος στο TinkerCAD

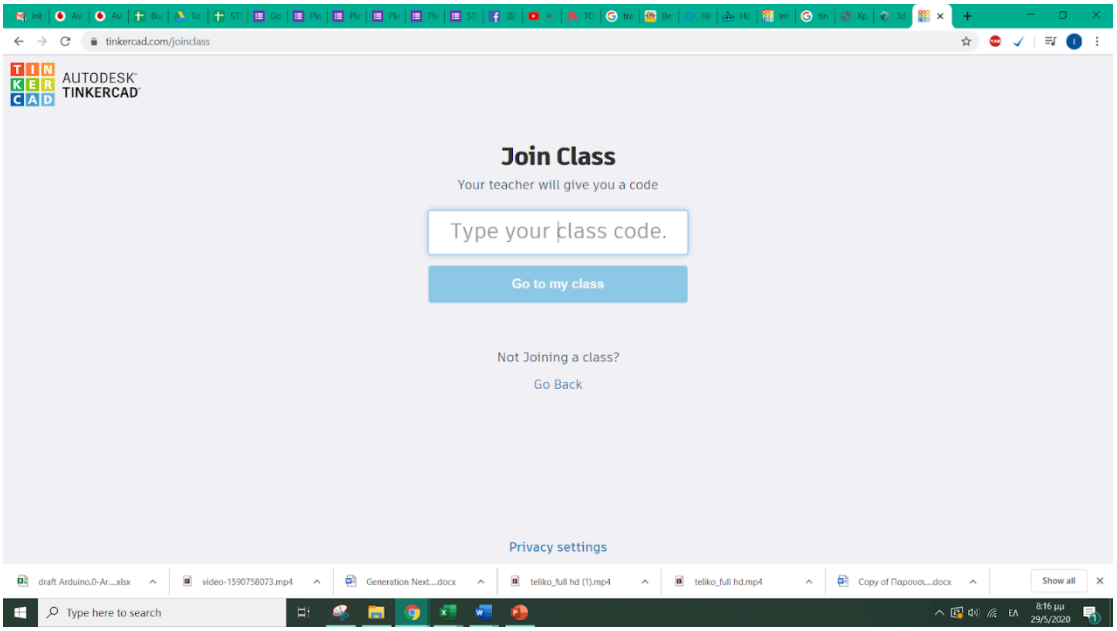


1. Για να ξεκινήσετε με το Tinkercad επισκεφθείτε την ιστοσελίδα Tinkercad.com και επιλέξτε “Start Tinkering” ή πάνω δεξιά “Join Now” και επιλέξτε “Educators start here”.
2. Οι μαθητές σας θα πρέπει να επιλέξουν “Students, join a Class”.

- Ως εκπαιδευτικοί θα πρέπει να δημιουργήσετε Teacher account, έτσι ώστε να μπορείτε να δημιουργήσετε τη δική σας τάξη για να συνδέονται οι μαθητές σας πολύ εύκολα με ένα link και να έχετε τη δυνατότητα να βλέπετε τα σχέδιά τους και να τα διαχειρίζεστε (TinkerCAD Classrooms).
- Αν έχετε ήδη προσωπικό λογαριασμό, αλλά όχι Teacher account, θα πρέπει να φτιάξετε νέο account και να σβήσετε τα προηγούμενα cookies ή να επικοινωνήσετε με την ομάδα του TinkerCAD για αλλαγή.

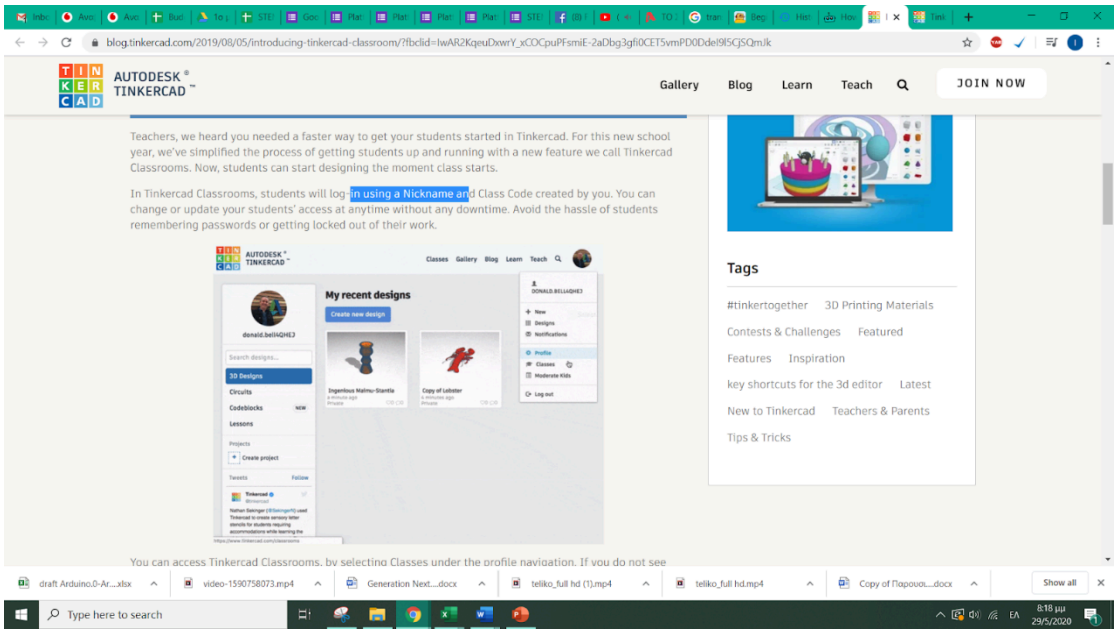
Tinkercad Classrooms

- **Σύνδεση**



Στα Tinkercad Classrooms, οι μαθητές συνδέονται χρησιμοποιώντας ένα username/ψευδώνυμο και έναν κωδικό που έχει δημιουργηθεί από εσάς. Μπορείτε να αλλάξετε ή να ανανεώσετε την πρόσβαση των μαθητών σας οποιαδήποτε στιγμή.

Για να μεταβείτε στα Tinkercad Classrooms, επιλέγετε Classes από το μενού του προφίλ σας. Αν δεν βλέπετε αυτή την επιλογή, βεβαιωθείτε ότι έχετε κάνει εγγραφή ως teacher στη σελίδα Teach.



Για να δημιουργήσετε μία τάξη, επιλέξτε Create new class. Θα σας ζητηθεί να ένα όνομα για την τάξη σας, καθώς επίσης και να προσθέσετε τη βαθμίδα και το θέμα της τάξης.

- **Προσθήκη μαθητών**

Για την εγγραφή των μαθητών σας στην τάξη, θα χρειαστεί να καταχωρήσετε μόνο το όνομά τους και ένα ψευδώνυμο που θα χρησιμοποιούν για την είσοδό τους στο Tinkercad. Το Tinkercad δημιουργεί αυτόματα ψευδώνυμα, βάσει του ονόματος του μαθητή που έχετε εισάγει, ωστόσο αν το επιθυμείτε εσείς ή οι μαθητές σας, μπορείτε να τα αλλάξετε (Εικόνα 1).

Εναλλακτικά, για μεγαλύτερη συντομία, μπορείτε να καταχωρήσετε μαζικά τη λίστα των μαθητών σας, επιλέγοντας το "Paste a list of students". Βεβαιωθείτε, ότι κάθε όνομα των μαθητών σας, είναι σε διαφορετική γραμμή (και όχι διαχωρισμένα με κόμμα ή tab). Κατόπιν, επιλέξτε "Add" και η λίστα των μαθητών θα προστεθεί στην τάξη σας, συμπεριλαμβανομένου των ψευδωνύμων τους

- **Σύνδεση μαθητών**

Αφού οι μαθητές σας επιλέξουν “Students, join a Class”, θα πρέπει να καταχωρήσουν το Ψευδώνυμό τους και τον κωδικό της τάξης, ο οποίος είναι μοναδικός για κάθε τάξη.

Για συντομία, παρέχεται και η εναλλακτική να καταχωρήσουν μόνο το Ψευδώνυμό τους.

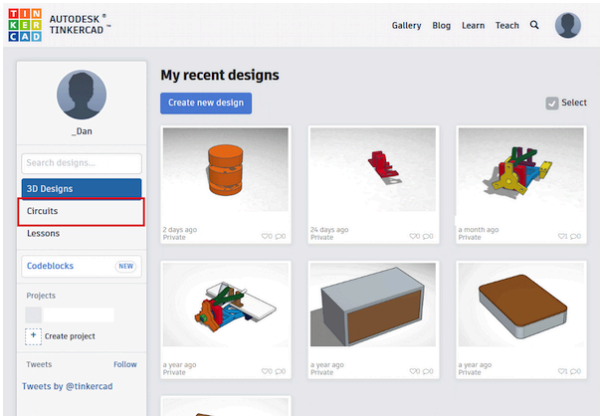
- **Επίβλεψη εργασιών των μαθητών**

Μέσω του TinkerCAD έχετε τη δυνατότητα να ελέγχετε και να επιβλέπετε τα projects των μαθητών σας. Επιλέγετε Class από πάνω δεξιά και εμφανίζεται η λίστα των μαθητών σας. Επιλέγετε το μαθητή και κατευθείαν έχετε πρόσβαση στα project του.

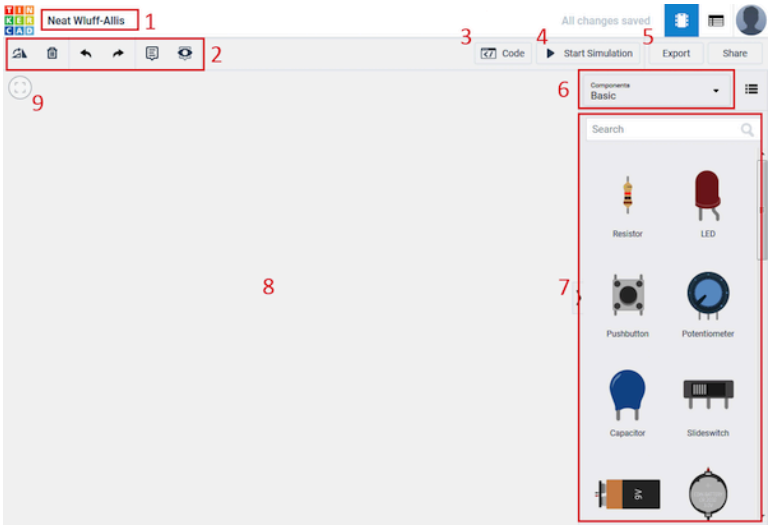
Δημιουργία κυκλώματος

- **Εισαγωγή**

Επιλέγετε “Circuits” στην αριστερή πλευρά της οθόνηςon the left side of the screen:



Έπειτα επιλέγετε “Create new Circuit” στην επόμενη σελίδα, όπου θα δείτε την παρακάτω οθόνη:



Παρακάτω, γίνεται επεξήγηση των βασικών εργαλείων/δυνατοτήτων, ούτως ώστε να εξοικειωθείτε με το περιβάλλον του TinkerCAD.

1. Αλλαγή ονόματος project.
2. Αυτή είναι η βασική περιοχή εργαλείων. Από αριστερά προς τα δεξιά: δεξιόστροφη περιστροφή επιλεγμένου αντικείμενου, διαγραφή επιλεγμένου τμήματος, αναίρεση και επανάληψη των τελευταίων ενεργειών, δημιουργία και εμφάνιση/απόκρυψη σημειώσεων.
3. Κάνοντας κλικ εδώ, μπορείς να αλλάξεις τον κώδικα, όταν αυτό επιτρέπεται από το σύστημα.

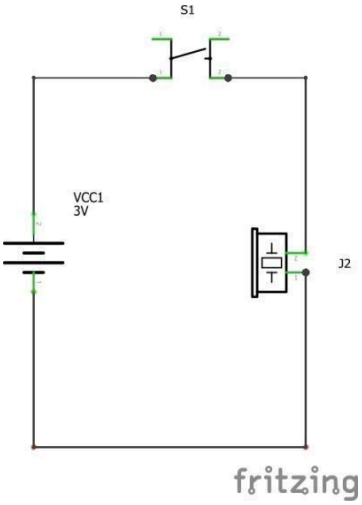
4. Έναρξη προσομοίωσης (σε περίπτωση τροποποίησης αν δεν ξεκινήσει κατευθείαν το ξαναπατάμε).
5. Εξαγωγή σχεδίου στο Autodesk EAGLE/δημιουργία αρχείου .BRD.
6. Λίστα στοιχείων που εμφανίζονται στο πεδίο από κάτω (προτείνεται να τα έχετε όλα εμφανή, οπότε επιλέξτε “All”).
7. Στοιχεία που έχουν επιλεχθεί.
8. Κύρια περιοχή σχεδίασης. Με “drag and drop” προσθέτετε στοιχεία στο σχέδιό σας.
9. Προσαρμογή όλων των στοιχείων στην περιοχή σχεδίασης.

Optional Δραστηριότητα εξοικείωσης με το περιβάλλον TinkerCad

Εισαγωγική Εφαρμογή: Κώδικας Morse

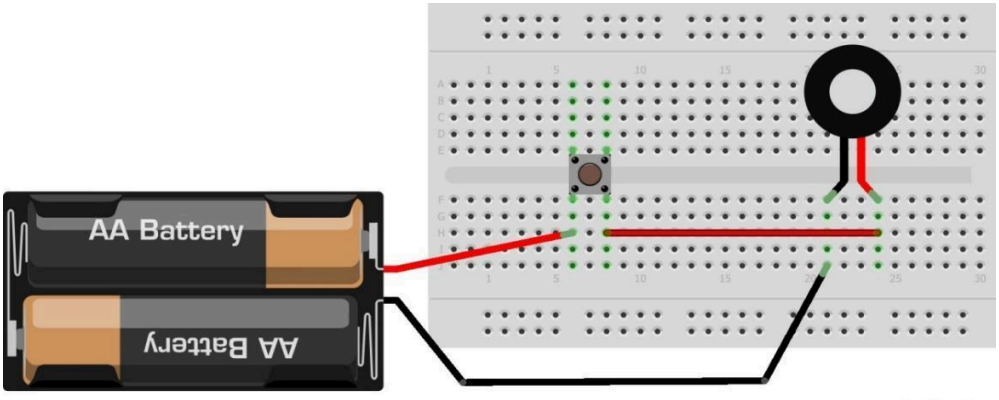
Χρόνος υλοποίησης: 20 λεπτά

Για να εξοικειωθούν οι μαθητές με το breadboard και τα ηλεκτρονικά κυκλώματα, έχουμε επιλέξει την παρακάτω δραστηριότητα που βασίζεται στην λογική του ανοιχτού και κλειστού κυκλώματος. Είναι ένα βομβητής για κώδικα Morse.



Υλικά που θα χρειαστείτε

Μπαταριοθήκη 3V (μπαταρίες AA ή AAA) , breadboard, καλώδια, buzzer (βομβητής), button (κουμπί πίεσης), LED κόκκινο (φωτοδίοδος) ή οποιουδήποτε άλλου χρώματος.



Δημιουργία Κυκλώματος με buzzer

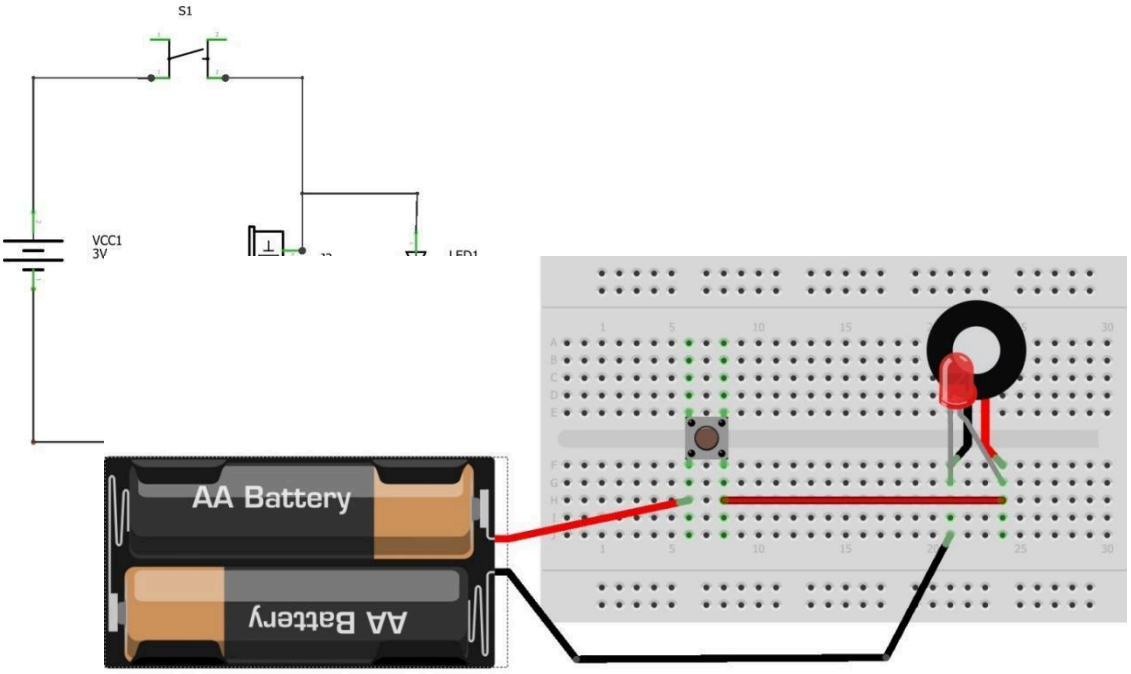
1. Φτιάχνουμε το κύκλωμα στο breadboard με το button, τον βομβητή και τα καλώδια όπως δείχνει το διάγραμμα και το σχήμα, με το θετικό πόλο του βομβητή στο καλώδιο που πάει στο + της μπαταρίας.

2. Συνδέουμε την μπαταρία όπως δείχνει το διάγραμμα και τεστάρουμε το κύκλωμα μας πιέζοντας το κουμπί και δίνοντας ρεύμα στο βομβητή.

Αναφέρουμε την παρόμοια χρήση που έχει ο διακόπτης για τα φώτα του σπιτιού μας και τις ιδιότητες του ανοιχτού - κλειστού κυκλώματος.

Δημιουργία Κυκλώματος με buzzer και led

Η δραστηριότητα μπορεί να εμπλουτιστεί με οπτικά, εκτός από ακουστικά μέσα! Η πρώτη επιλογή που έχετε είναι να αφαιρέσετε το buzzer και να το αντικαταστήσετε με το led, χωρίς να χαλάσετε το υπόλοιπο κύκλωμα. Η δεύτερη επιλογή που έχετε είναι να συνδέσετε παράλληλα το led με το buzzer όπως δείχνει το παρακάτω σχήμα.



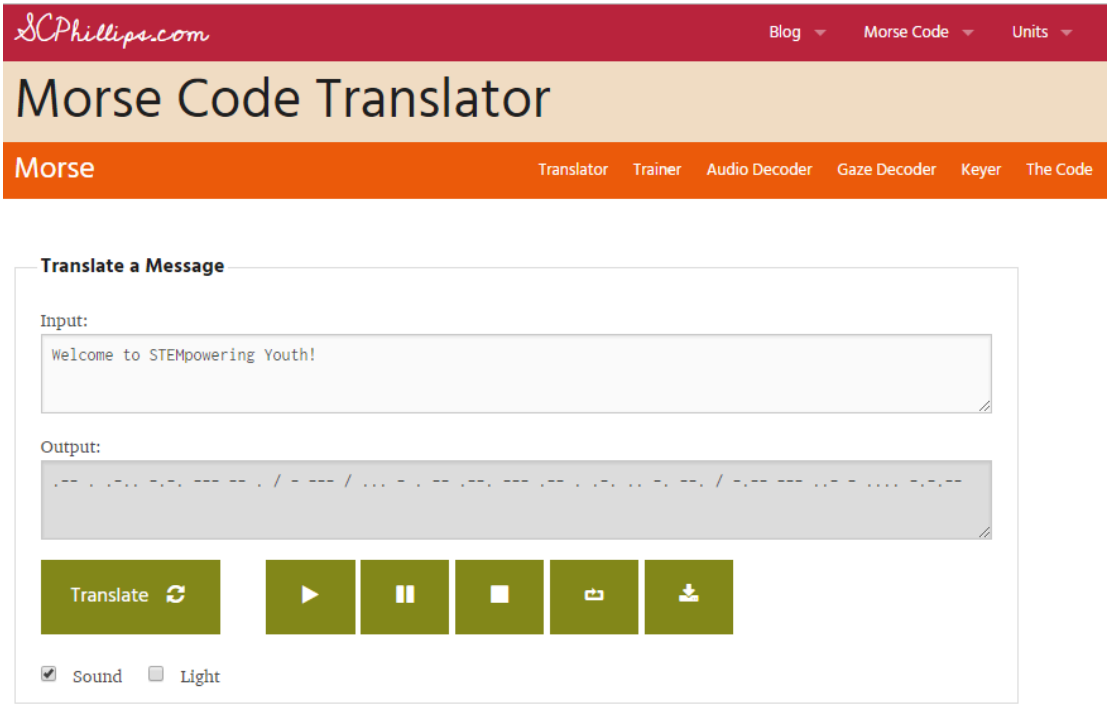
ΠΡΟΣΟΧΗ: Και το LED και το buzzer έχουν πολικότητα (+ και -). Το μακρύ πόδι του LED (+) να συνδεθεί με τον θετικό πόλο της μπαταρίας. Όμοια το + του βομβητή στο + της μπαταρίας.

Αφού φτιάξουμε τον τηλέγραφο, ήρθε η ώρα να στείλουμε το πρώτο μας μήνυμα! Για την δημιουργία του μηνύματος μας μπορούμε να επισκεφθούμε την παρακάτω ιστοσελίδα που μεταφράζει το μήνυμα μας σε κώδικα Morse:

https://morsecode.scphillips.com/translator.html

και για όσους μαθητές θέλουν να εξασκήσουν τις γνώσεις τους, προτείνουμε την παρακάτω ιστοσελίδα που έχει τον κώδικα Morse στα ελληνικά, ενδιαφέρουσες λεπτομέρειες και διάφορα μνημονικά εκμάθησης του κώδικα:

https://paroutsas.jmc.gr/different/morse.htm



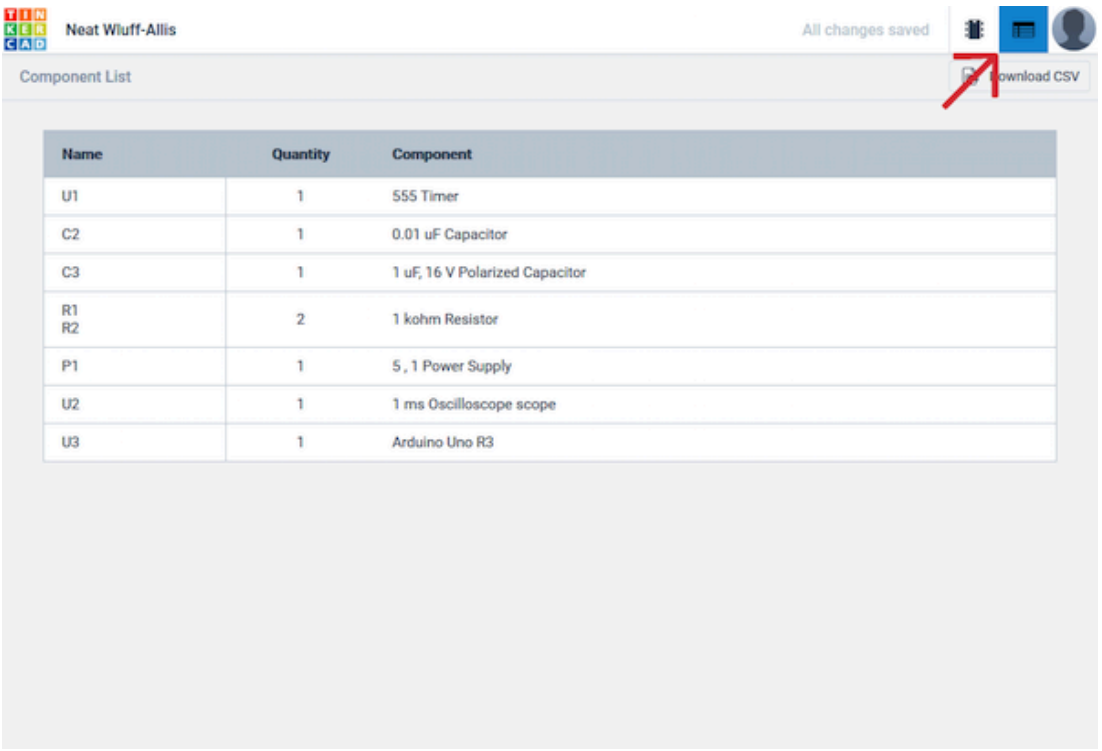
Η δραστηριότητα αυτή είναι ιδιαίτερα σημαντική για την **μετάβαση στην λογική του Arduino** μιας και περιλαμβάνει ένα απλό κύκλωμα με καλώδια, μπαταρία, buzzer και breadboard.

Breadboard: Μας δίνεται η ευκαιρία να μιλήσουμε στους μαθητές μας για τη λογική που κρύβετε κρύβεται από πίσω με τις συνδέσεις, εξηγώντας λέγοντας τους ότι το breadboardείναι φτιαγμένο για να μην χρειάζεται να κάνουμε κολλήσεις κάθε φορά μεταξύ του led και του καλωδίου.

Μπαταρία και κύκλωμα: Ανοιγοκλείνοντας το κουμπί ανοιγοκλείνουμε το κύκλωμα, δηλαδή ανοιγοκλείνουμε τη ροή του ρεύματος προς το λαμπάκι / buzzer. Μόλις οι μαθητές μας το κατανοήσουν, τους εξηγούμε ότι αντί για μπαταρία μπορούμε να τροφοδοτήσουμε το λαμπάκι / buzzer μέσω την πλακέτα arduino ... και όχι μόνο! Μπορούμε εκτός από το να δίνουμε ρεύμα να προγραμματίσουμε το λαμπάκι / buzzer να κάνει αυτό που θέλουμε!

Εξαγωγή

Αφού ολοκληρώσετε το σχέδιό σας, μπορείτε να το κάνετε export ή να δημιουργήσετε μία λίστα από στοιχεία που θα μπορείτε να τροποποιήσετε. Για να εμφανίσετε τη λίστα, κάντε κλικ στο κουμπί δίπλα από την εικόνα του προφίλ σας, όπως φαίνεται παρακάτω.



Name	Quantity	Component
U1	1	555 Timer
C2	1	0.01 uF Capacitor
C3	1	1 uF, 16 V Polarized Capacitor
R1 R2	2	1 kohm Resistor
P1	1	5 , 1 Power Supply
U2	1	1 ms Oscilloscope scope
U3	1	Arduino Uno R3

Arduino Basics

Εφαρμογή 1: LED που αναβοσβήνει (Blink)

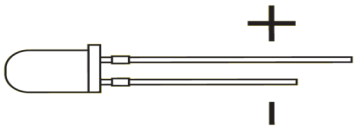
Στα πλαίσια της εφαρμογής θα παρουσιαστεί η χρήση των ψηφιακών ΠΙΝ της πλακέτας Arduino καθώς και η βασική δομή ενός προγράμματος Arduino. Η τελική κατασκευή θα είναι ένα LED που θα αναβοσβήνει περιοδικά.

Υλικά

Τα υλικά που θα χρησιμοποιήσουμε για την εφαρμογή μας:

LED

Το LED (Light Emitting Diode) είναι ένα στοιχείο, το οποίο όταν διαρρέεται από ρεύμα φωτοβολεί. Όσο μεγαλύτερη είναι η ένταση του ρεύματος που διαρρέει το LED, τόσο εντονότερο είναι το φως που παράγεται.



Το LED

Ως δίοδος, το LED επιτρέπει τη διέλευση του ρεύματος μόνο προς μία φορά. Όπως φαίνεται και στην Εικόνα 3 οι δύο ακροδέκτες του LED έχουν διαφορετικό μήκος. Ο πιο μακρύς ονομάζεται άνοδος και συνδέεται στο θετικό πόλο της πηγής (+), ενώ ο πιο κοντός ονομάζεται κάθοδος και συνδέεται στον αρνητικό πόλο (- ή GND ή γείωση).

Αντιστάτης

Ανάμεσα στο LED και την πηγή πρέπει να παρεμβάλλεται ένας αντιστάτης, προκειμένου να περιοριστεί το ρεύμα που θα διαρρεύσει το κύκλωμα και να προστατευτεί το LED. Η πιο συνηθισμένη τιμή αντίστασης που χρησιμοποιείται με το LED στις εφαρμογές Arduino είναι 220 Ω. (Μπορούμε όμως να βάλουμε οποιαδήποτε τιμή από 1 kΩ (1000 Ω) ως 180 Ω. Μεγάλη αντίσταση = μικρότερη φωτεινότητα του LED, αλλά και μικρότερη κατανάλωση ενέργειας, η οποία βέβαια είναι αμελητέα στις δικές μας εφαρμογές.

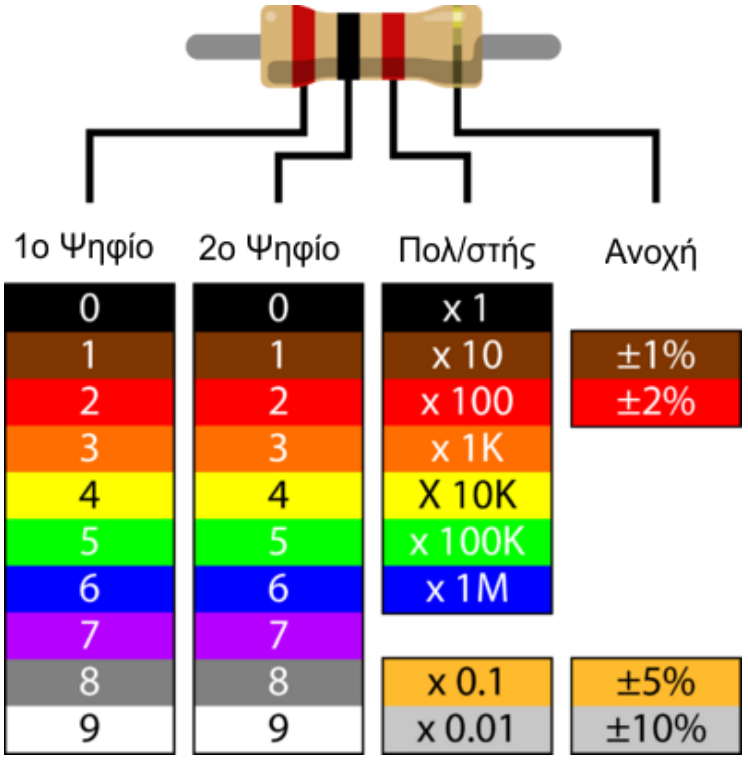
Αντιστάτης 220 Ω

Χρωματικός κώδικας αντιστατών

Η τιμή της αντίστασης ενός αντιστάτη δηλώνεται με χρωματιστές ζώνες επάνω στο εξάρτημα. Για τους αντιστάτες με μπεζ χρώμα, που είναι και οι πιο συνηθισμένοι, Η 1η και η 2η ζώνη είναι αριθμοί και η 3η ζώνη είναι ο πολλαπλασιαστής, δηλαδή μας λέει

πόσα μηδενικά να προσθέσουμε στο τέλος. Η 4η ζώνη, λιγότερο σημαντική για εμάς, δείχνει την ανοχή του αντιστάτη (ακρίβεια της τιμής του).

ΧΡΩΜΑΤΙΚΟΣ ΚΩΔΙΚΑΣ ΑΝΤΙΣΤΑΤΩΝ



Π.χ. για τον αντιστάτη του διπλανού σχήματος είναι:

1η ζώνη: Κόκκινο = 2

2η ζώνη: Κόκκινο = 2

3η ζώνη: Καφέ = 1 (Σημαίνει: Προσθέτω 1 μηδενικό)

Επομένως σχηματίζεται ο αριθμός: 2 2 0 δηλαδή 220 Ω.

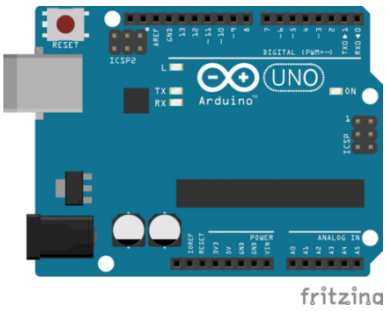
4η ζώνη: Χρυσό = Ανοχή 5%

Καλώδια

Για τη διασύνδεση των διαφόρων στοιχείων των κυκλωμάτων μας, θα χρησιμοποιηθούν ειδικά καλώδια, που είναι κατάλληλα για χρήση με breadboard και ονομάζονται jumper cables.

Arduino UNO

Το Arduino UNO διαθέτει 14 ψηφιακούς ακροδέκτες εισόδου/εξόδου (0 – 13). Όταν οι ψηφιακοί ακροδέκτες χρησιμοποιούνται ως έξοδοι, μπορούν να τεθούν σε μία από δύο καταστάσεις: HIGH (5V) και LOW (0V).

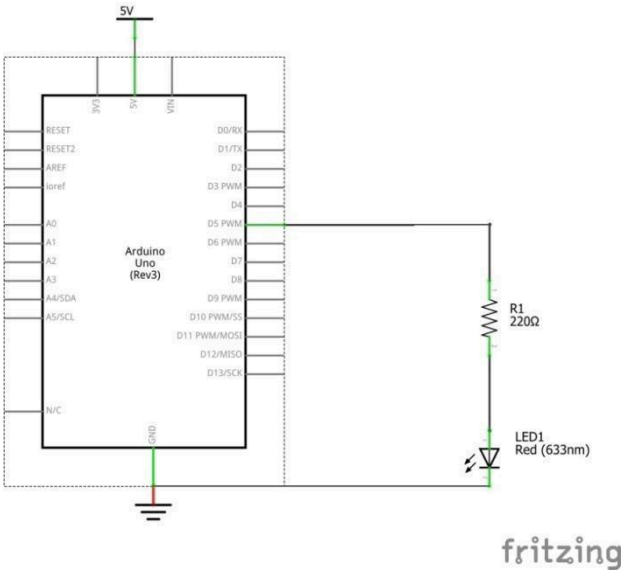


Arduino UNO

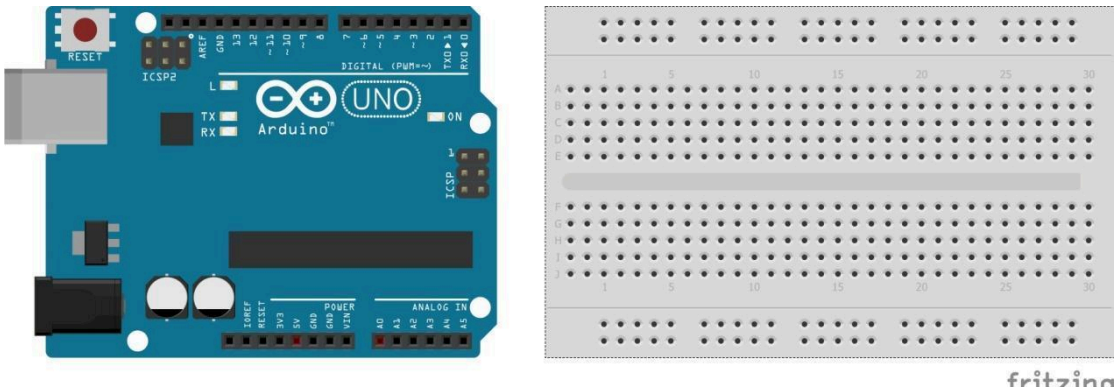
Στα πλαίσια της εφαρμογής μας θα χρησιμοποιήσουμε το Arduino ως μία προγραμματιζόμενη πηγή για την τροφοδοσία του κυκλώματος. Συγκεκριμένα θα αξιοποιήσουμε έναν από τους ψηφιακούς ακροδέκτες, ο οποίος όταν τίθεται σε κατάσταση HIGH το LED θα ανάβει, ενώ όταν τίθεται σε κατάσταση LOW το LED θα σβήνει. Ο ακροδέκτης γείωσης (GND) θα παίζει το ρόλο του αρνητικού πόλου της πηγής.

Κατασκευή κυκλώματος

Στη συνέχεια παρουσιάζεται βήμα προς βήμα η κατασκευή του κυκλώματος. Το ΣΧΗΜΑΤΙΚΟ ΔΙΑΓΡΑΜΜΑ του κυκλώματος που θα κατασκευάσουμε είναι:

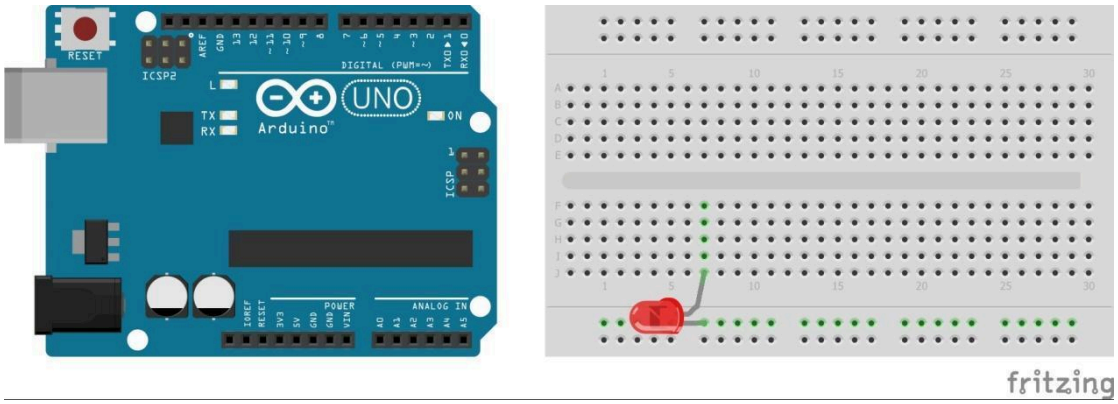


Ξεκινάμε με το Arduino UNO και ένα breadboard μισού μεγέθους όπως φαίνεται στην εικόνα. Ακολουθούμε τα βήματα που περιγράφονται στη συνέχεια.



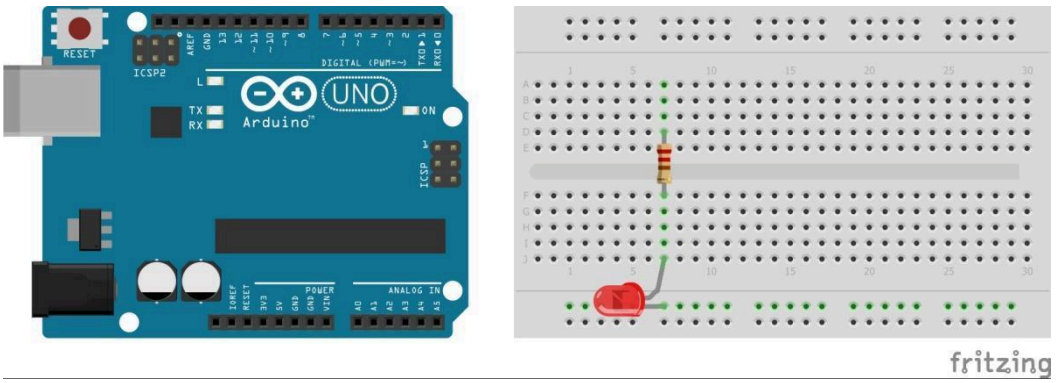
Βήμα 1

Συνδέουμε ένα LED στο breadboard όπως φαίνεται στο παρακάτω σχήμα. Προσέχουμε η ΑΝΟΔΟΣ (+) του LED (μακρύ πόδι) να είναι επάνω όπως βλέπουμε το σχήμα. Η ΚΑΘΟΔΟΣ (-) του LED (κοντό πόδι) συνδέεται κάτω, στη ράγα με τις πολλές οπές.



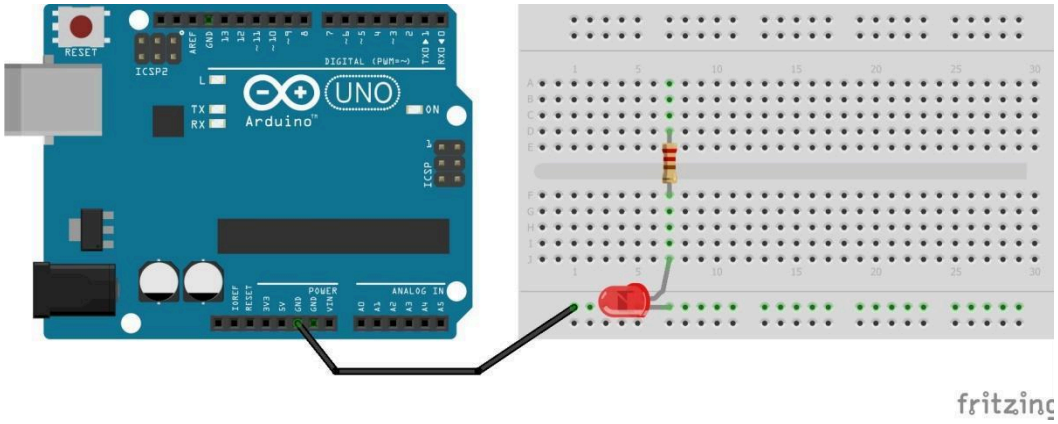
Βήμα 2

Συνδέουμε έναν αντιστάτη 220 Ω (κόκκινο, κόκκινο, καφέ, χρυσό) όπως φαίνεται στο σχήμα που ακολουθεί. Παρατηρούμε ότι συνδέεται η ΑΝΟΔΟΣ του LED με το ένα άκρο του αντιστάτη αφού μπήκαν στην ίδια πεντάδα οπών.



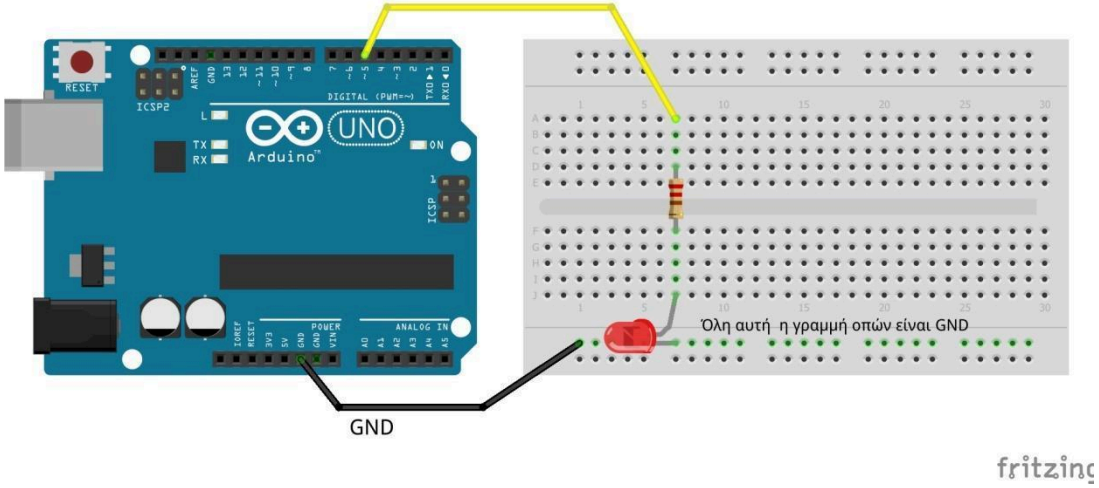
Βήμα 3

Συνδέουμε ένα καλώδιο από το πιν GND στην ακριανή οπή της γραμμής οπών όπου έχουμε συνδέσει την ΚΑΘΟΔΟ του LED. Έτσι, τώρα η ΚΑΘΟΔΟΣ του LED είναι συνδεμένη με τη γείωση (GND).



Βήμα 4

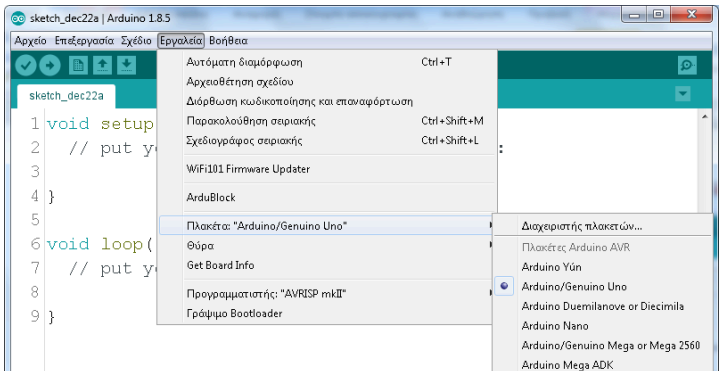
Συνδέουμε ένα καλώδιο από το ΨΗΦΙΑΚΟ ΠΙΝ 5 του Arduino στην πεντάδα οπών όπου είχαμε βάλει το ελεύθερο (επάνω) άκρο του αντιστάτη.



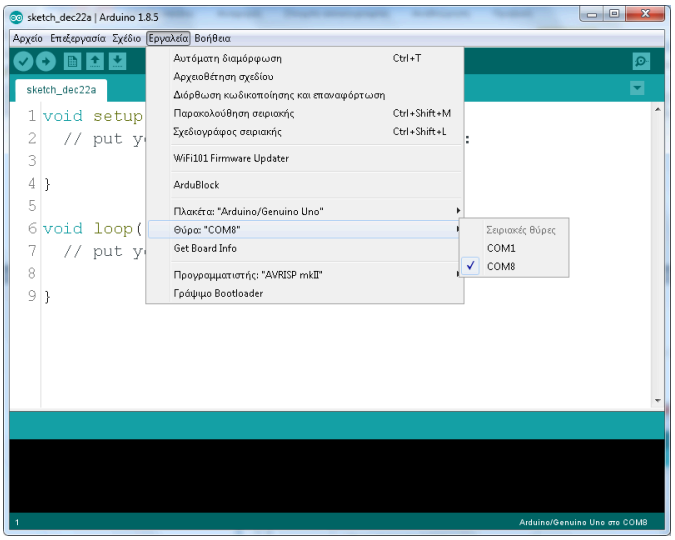
Τώρα το κύκλωμα είναι ολοκληρωμένο.

Σύνδεση του Arduino UNO με τον υπολογιστή

Συνδέουμε το Arduino UNO με το καλώδιο USB σε μία από τις USB θύρες του υπολογιστή. Στη συνέχεια, εκτελούμε το Arduino IDE. Στο παράθυρο που θα ανοίξει πηγαίνουμε στο μενού *Εργαλεία*, στην εγγραφή *Πλακέτα* και επιλέγουμε *Arduino/Genuino UNO*.



Ακολούθως, πάλι από το μενού *Εργαλεία*, πηγαίνουμε στο *Θύρα* και επιλέγουμε τη θύρα COM στην οποία έχει συνδεθεί το Arduino



Ανάπτυξη προγράμματος σε Arduino IDE

Τώρα βρισκόμαστε στο παράθυρο με το μεγάλο λευκό άδειο χώρο όπου γράφουμε το πρόγραμμα σε γλώσσα C (ακριβέστερα C++) του Arduino. Τα προγράμματα που γράφουμε ονομάζονται «σκίτσα» στην ορολογία του Arduino (sketches).

Κάθε πρόγραμμα έχει δύο διαδικασίες (υποπρογράμματα ή ομάδες εντολών) που προϋπάρχουν και ΕΙΝΑΙ ΑΠΑΡΑΙΤΗΤΕΣ στο πρόγραμμά μας: τις **setup()** και **loop().**

- Στη **setup** βάζουμε τις εντολές που θέλουμε να εκτελεστούν μία φορά μόνο.
- Στη **loop** βάζουμε τις εντολές που θέλουμε να επαναλαμβάνονται, αφού, όταν τελειώσει, η loop ξαναρχίζει από την αρχή της. Αυτό συνεχίζεται μέχρι να αποσυνδέσουμε το Arduino από την τάση τροφοδοσίας ή να πατήσουμε το κουμπί Reset.

Στο πρώτο παράδειγμα, γράφουμε ένα πρόγραμμα που αναβοσβήνει ένα LED. Το πρόγραμμά μας λέγεται «LED που αναβοσβήνει» ή "BLINK".

Το πρόγραμμα "BLINK" σε κώδικα C του Arduino:

```
void setup() {
  pinMode(5, OUTPUT);    // Όρισε το πιν 5 ως έξοδο
}

void loop() {
  digitalWrite(5, HIGH);   // Άναψε το LED
  delay(1000);                 // Περίμενε 1 δευτερόλεπτο εδώ
  digitalWrite(5, LOW);    // Σβήσε το LED
  delay(1000);                 // Περίμενε 1 δευτερόλεπτο εδώ
}
```

Οι εντολές που χρησιμοποιήθηκαν είναι:

- pinMode (… αριθμός του πιν , … INPUT ή OUTPUT);

Αυτή η εντολή ορίζει κάποιο πιν του Arduino ως ΕΙΣΟΔΟ (INPUT) ή ΕΞΟΔΟ (OUTPUT). ΣΗΜΕΙΩΣΗ: Τελειώνουμε την εντολή με το ελληνικό ερωτηματικό. Π.χ pinMode(5, OUTPUT);

- digitalWrite(… αριθμός του πιν, … HIGH ή LOW);

Αυτή η εντολή κάνει το αντίστοιχο πιν να βγάζει + (5V) ή - (GND). Ο αριθμός του πιν μπορεί να είναι από 0 ως 13 (αναφερόμαστε στα ψηφιακά πιν). Π.χ. digitalWrite(5, HIGH);

- delay(… χρόνος σε ms);

Αυτή η εντολή απλά σταματάει τη ροή του προγράμματος για ορισμένο χρόνο, τον οποίο δώσαμε σε χιλιοστά του δευτερολέπτου (millisecond ή ms). Π.χ. μπορούμε να γράψουμε: delay(2000); για καθυστέρηση 2 δευτερόλεπτα.

- // Σχόλια

Αν βάλουμε δύο κάθετες τη μια δίπλα στην άλλη, ό,τι ακολουθεί σε εκείνη τη σειρά αγνοείται από τον μεταγλωττιστή. Πρόκειται για ΣΧΟΛΙΟ. Είναι σημείωση για εμάς, για να μας υπενθυμίζει τι κάνει κάθε εντολή όταν διαβάζουμε μετά από καιρό το πρόγραμμα εμείς οι ίδιοι.

ΠΡΟΣΟΧΗ: Κάθε εντολή ΠΡΕΠΕΙ να τελειώνει με το ελληνικό ερωτηματικό. (Εξαιρούνται οι εντολές επανάληψης και επιλογής, που εδώ δεν χρησιμοποιήθηκαν).

Optional: Εφαρμογή 1: Βομβητής στη θέση του LED

Βομβητής (Active Buzzer)

Ο βομβητής ή buzzer είναι συσκευή που παράγει ήχο. Υπάρχουν δύο κατηγορίες buzzer, τα ενεργά (active) και τα παθητικά (passive). Τα active buzzer διαθέτουν εσωτερικό ταλαντωτή και όταν τροφοδοτούνται με συνεχή τάση, παρέχουν ένα τόνο συγκεκριμένης συχνότητας. Αντίθετα, τα passive buzzer δεν διαθέτουν εσωτερικό ταλαντωτή και για να παράγουν ήχο, πρέπει η τάση τροφοδοσίας τους να μεταβάλλεται (π.χ. HIGH□LOW□HIGH□LOW …). Η συχνότητα του ήχου που παράγει ένα passive buzzer είναι ίση με τη συχνότητα με την οποία μεταβάλλεται η τάση τροφοδοσίας του και άρα μπορεί να αλλάζει.

Βομβητής (Buzzer)

Πειραματισμός

Πραγματοποιούμε την δραστηριότητα με το λαμπάκι όπως περιεγράφηκε πριν, και μετά αντικαθιστούμε το λαμπάκι με το buzzer. Το πρόγραμμα που είναι περασμένο στην πλακέτα Arduino παραμένει το ίδιο. Απλά αλλάζουμε το LED με τον βομβητή και παρατηρούμε τον διακοπτόμενο ήχο που παράγεται.

Εφαρμογή 2: LED που ανάβει με το πάτημα κουμπιού

Οι ψηφιακοί ακροδέκτες του Arduino (ακροδέκτες 0 - 13 στο UNO) μπορούν να χρησιμοποιηθούν είτε ως ΕΞΟΔΟΙ είτε ως ΕΙΣΟΔΟΙ. Για να γνωρίσουμε τη χρήση των πιν ως ψηφιακών εισόδων του Arduino, θα προσθέσουμε στην 1η εφαρμογή ένα κουμπί συνδεμένο σε ένα άλλο πιν. Όσο το κουμπί παραμένει πατημένο, το LED θα είναι αναμμένο. Μόλις αφήνουμε το κουμπί, το LED θα σβήνει. Για το σκοπό αυτό, θα χρησιμοποιήσουμε μέσα στο πρόγραμμα μία εντολή ελέγχου: if(…) … else … (δηλαδή: *αν(…) …αλλιώς …)*, μέσα στην οποία θα ελέγχουμε την κατάσταση του πιν ΕΙΣΟΔΟΥ (πιν 2) και θα ενεργοποιούμε κατάλληλα το πιν ΕΞΟΔΟΥ (πιν 5).

Υλικά

Στη συνέχεια παρουσιάζονται τα επιπλέον υλικά που θα χρησιμοποιηθούν στη δεύτερη εφαρμογή.

Κουμπί

Στην εφαρμογή αυτή χρησιμοποιούμε για πρώτη φορά ένα κουμπί πίεσης (button).

(α) **(β)**

Όπως φαίνεται στην Εικόνα (α), το κουμπί πίεσης διαθέτει 4 ακροδέκτες. Οι ακροδέκτες αυτοί είναι ανά 2 συνδεδεμένοι μεταξύ τους (πάνω-κάτω), ενώ η διάταξη χωρίζεται σε δύο ανεξάρτητα κομμάτια (δεξί – αριστερό) όπως φαίνεται στο σχηματικό σύμβολο (Εικόνα (β)). Όταν πιέζουμε το κουμπί, κλείνει ο διακόπτης και συνδέεται το δεξί με το αριστερό του μέρος.

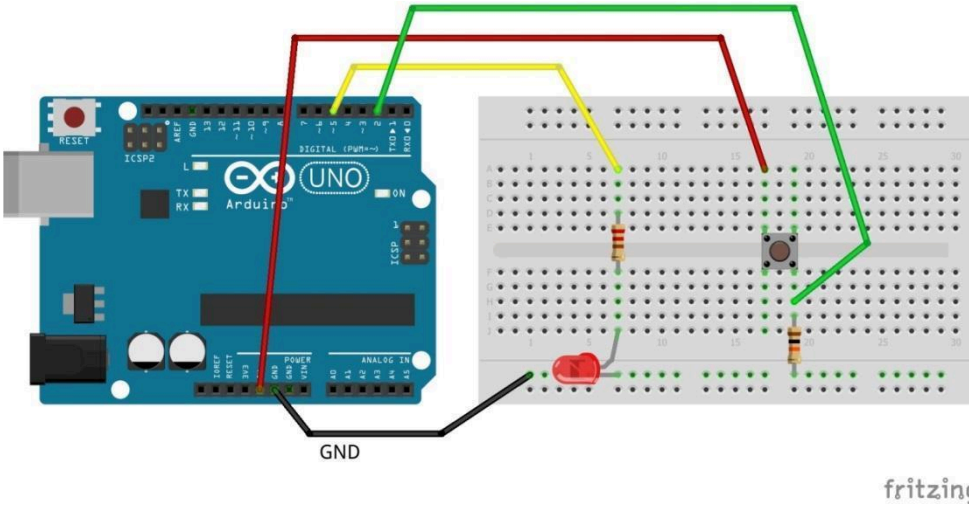
Αντιστάτης 10kΩ (χρώματα: καφέ, μαύρο, πορτοκαλί, χρυσό)

Για τη σύνδεση του κουμπιού με το Arduino θα χρησιμοποιήσουμε έναν αντιστάτη 10 kΩ.

Αντιστάτης 10kΩ

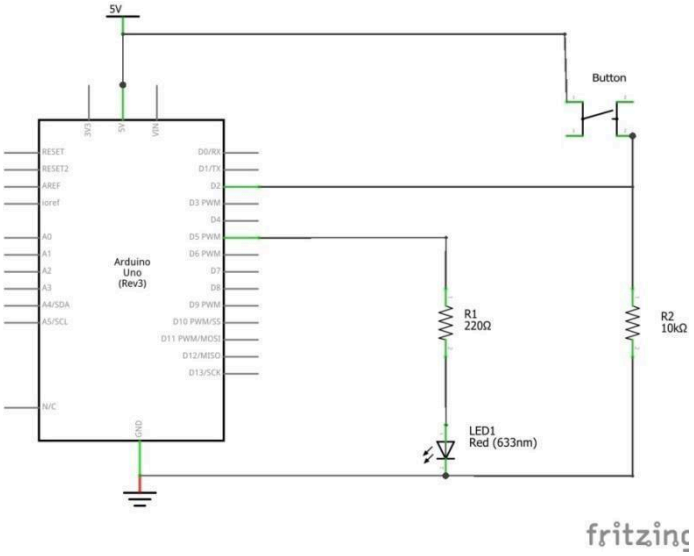
Κύκλωμα

Το κύκλωμα θα περιλαμβάνει ένα LED και ένα κουμπί. Όσον αφορά τη συνδεσμολογία του LED, δεν υπάρχει κάποια διαφοροποίηση σε σχέση με το κύκλωμα της 1ης εφαρμογής. Προσθέτουμε όμως το κουμπί (button) και έναν αντιστάτη 10kΩ. Κατασκευάζουμε το κύκλωμα όπως στο σχήμα:



Το αριστερό τμήμα του κουμπιού συνδέεται στην τροφοδοσία (pin 5 V) και το δεξί στη γείωση (GND) μέσω της αντίστασης 10 kΩ, για αποφυγή βραχυκυκλώματος όταν πατάμε το κουμπί. Για τον έλεγχο της κατάστασης του κουμπιού, συνδέουμε το δεξί του τμήμα στον ψηφιακό ακροδέκτη 2 του Arduino, που θα χρησιμοποιηθεί ως είσοδος. Όταν το κουμπί δεν είναι πατημένο, το κύκλωμα είναι ανοικτό, δεν υπάρχει ρεύμα, ούτε και πτώση τάσης στην αντίσταση των 10 kΩ. Άρα, ο ακροδέκτης 2 είναι συνδεμένος, μέσω της R2, στη γείωση και είναι σε δυναμικό 0V (κατάσταση LOW). Όταν πατηθεί το κουμπί, κλείνει το κύκλωμα και ο ακροδέκτης 2 βρίσκεται συνδεμένος σε δυναμικό 5 V, δηλαδή στον θετικό πόλο της τάσης τροφοδοσίας (κατάσταση HIGH).

Το σχηματικό διάγραμμα του κυκλώματος είναι:



Ανάπτυξη προγράμματος σε Arduino IDE

Το πρόγραμμα «LED και κουμπί» σε κώδικα C του Arduino:

```
void setup() {
  pinMode(5, OUTPUT);   // Όρισε το πιν 5 ως έξοδο  (LED)
  pinMode(2, INPUT);   // Όρισε το πιν 2 ως είσοδο  (Διακόπτης)
}

void loop() {
  if (digitalRead(2) == HIGH) {    // Αν ο διακόπτης είναι πατημένος…
    digitalWrite(5, HIGH);   // Άναψε το LED
  }
  else {                           // … αλλιώς …
    digitalWrite(5, LOW);   // Σβήσε το LED
  }
}
```

Η νέα εντολή είναι:

• digitalRead(… πιν);

Διαβάζει την κατάσταση του πιν που αναφέρουμε στην παρένθεση. Π.χ. η εντολή: a=digitalRead(2); διαβάζει την κατάσταση του πιν 2 και την αποθηκεύει στη μεταβλητή a.

ΠΑΡΑΤΗΡΗΣΗ: Μπορεί κάποιος να θέσει το ερώτημα: «Γιατί να μη συνδέσουμε απλά το κουμπί σε σειρά με τον αντιστάτη και το LED και να ξεφορτωθούμε και το Arduino; Πάλι θα λειτουργεί το κύκλωμα με τον ίδιο τρόπο.». Αυτό είναι σωστό, αλλά έχοντας ως «ενδιάμεσο» το Arduino, μπορούμε να κάνουμε αυτοματισμούς όπως; Να πατάμε το κουμπί στιγμιαία και το LED να ανάβει για ορισμένο χρόνο και μετά να σβήνει, όπως στο φως ενός κλιμακοστασίου πολυκατοικίας. Αυτό και πολλά άλλα μπορούν να γίνουν μόνο αν έχουμε στη διάθεσή μας ένα μικρό υπολογιστή όπως το Arduino. Χωρίς να αλλάξουμε το υλικό (hardware) μπορούμε να αλλάξουμε τον τρόπο λειτουργίας με αλλαγές στο λογισμικό (software).

Εφαρμογή 3: LED που αναβοσβήνει και LED με κουμπί

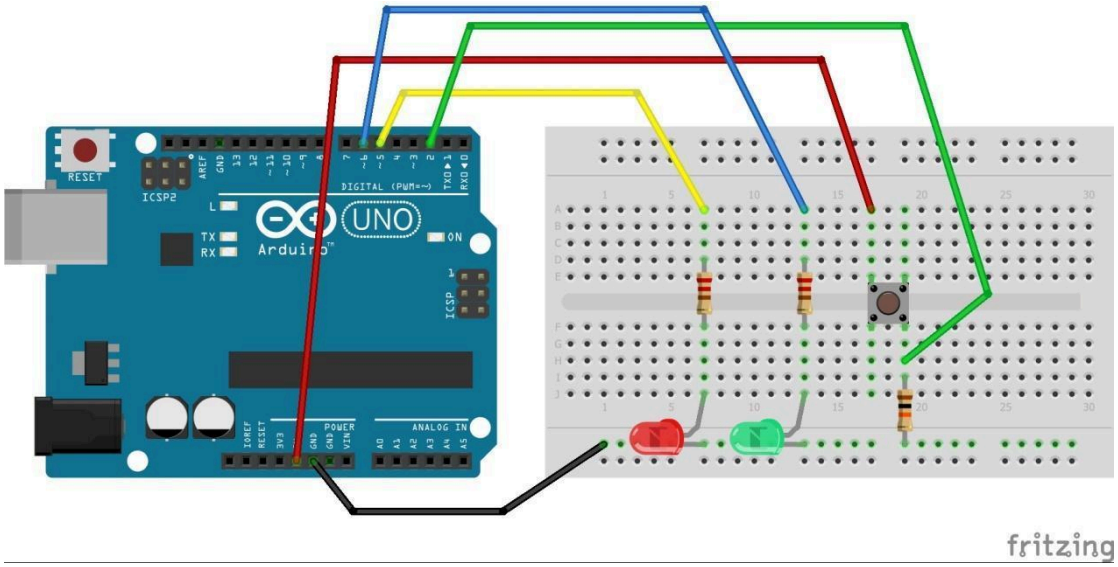
Αυτή η εφαρμογή αποτελεί έναν συνδυασμό των Εφαρμογών 1 και 2. Συγκεκριμένα, θα υπάρχουν δύο LED και ένα κουμπί. Το ένα LED θα αναβοσβήνει περιοδικά (όπως στην Εφαρμογή 1), ενώ το δεύτερο θα ανάβει όσο το κουμπί είναι πατημένο (Εφαρμογή 2). Στα πλαίσια της ανάπτυξης του προγράμματος, θα εντοπίσουμε ένα μειονέκτημα της εντολής *delay MILLIS* και θα προτείνουμε μία εναλλακτική υλοποίηση χρονοκαθυστέρησης.

Υλικά

Τα υλικά που θα χρησιμοποιηθούν για την εφαρμογή περιλαμβάνουν 2 LED (κόκκινο και πράσινο), 2 αντιστάσεις 220Ω (για τα LED), ένα κουμπί, 1 αντίσταση 10kΩ (για το κουμπί) και 5 καλώδια.

Κύκλωμα

Το τελικό κύκλωμα φαίνεται στην παρακάτω εικόνα



Όπως φαίνεται στην εικόνα, η άνοδος του δεύτερου LED έχει συνδεθεί, μέσω της αντίστασης 220Ω, στον ψηφιακό ακροδέκτη 6 του Arduino.

Σύνδεση πλακέτας Arduino UNO με τον υπολογιστή

Συνδέουμε την πλακέτα στον υπολογιστή και μέσα από το μενού *Εργαλεία* του Arduino IDE ορίζουμε τον τύπο της πλακέτας και τη θύρα σύνδεσης.

Ανάπτυξη προγράμματος σε Arduino IDE

Υλοποίηση_Α:

Το πρόγραμμα σε κώδικα C του Arduino:

```
void setup() {
  pinMode(5, OUTPUT);   // Όρισε το πιν 5 ως έξοδο  (κόκκινο LED)
  pinMode(6, OUTPUT);   // Όρισε το πιν 6 ως έξοδο  (πράσινο LED)
  pinMode(2, INPUT);    // Όρισε το πιν 2 ως είσοδο  (διακόπτης)
}


void loop() {
  if (digitalRead(2)== HIGH) {    // Αν ο διακόπτης είναι πατημένος…
    digitalWrite(5, HIGH);   // Άναψε το κόκκινο LED
  }
  else {                           // … αλλιώς …
    digitalWrite(5, LOW);    // Σβήσε το κόκκινο LED
  }
  digitalWrite(6, HIGH);   // Άναψε το πράσινο LED
  delay(3000);                  // Περίμενε 3 δευτερόλεπτα εδώ
  digitalWrite(6, LOW);    // Σβήσε το πράσινο LED
  delay(1000);                  // Περίμενε 3 δευτερόλεπτα εδώ
}
```

Υλοποίηση_Β:

Το πρόγραμμα σε κώδικα C του Arduino:

```
unsigned long start_time; // Δημιούργησε μια μεταβλητή τύπου unsigned long integer με όνομα start_time


 void setup() {
  pinMode(5, OUTPUT);   // Όρισε το πιν 5 ως έξοδο  (κόκκινο LED)
  pinMode(6, OUTPUT);   // Όρισε το πιν 6 ως έξοδο  (πράσινο LED)
  pinMode(2, INPUT);    // Όρισε το πιν 2 ως είσοδο  (διακόπτης)
}


void loop() {
  digitalWrite(6, HIGH);   // Άναψε το πράσινο LED
  start_time=millis();     // Βάλε στην start_time την τιμή της millis()
```

```
  while (millis() - start_time < 3000) {  // Εφόσον δεν πέρασαν 3 δευτ κάνε:
    if (digitalRead(2) == HIGH) {    // Αν ο διακόπτης είναι πατημένος…
      digitalWrite(5, HIGH);   // Άναψε το κόκκινο LED
    }
    else {                           // … αλλιώς …
      digitalWrite(5, LOW);   // Σβήσε το κόκκινο LED
    }
  }   // (τέλος της while)
  digitalWrite(6, LOW);   // Σβήσε το πράσινο LED
  start_time=millis();    // Βάλε στην start_time την τιμή της millis()
  while (millis() - start_time < 3000) {  // Εφόσον δεν πέρασαν 3 δευτ κάνε:
    if (digitalRead(2) == HIGH) {    // Αν ο διακόπτης είναι πατημένος…
      digitalWrite(5, HIGH);   // Άναψε το κόκκινο LED
    }
    else {                           // … αλλιώς …
      digitalWrite(5, LOW);   // Σβήσε το κόκκινο LED
    }
  }   // (τέλος της while)
}  // (τέλος της loop)
```

Οι εντολές που χρησιμοποιούνται είναι:

• unsigned long start_time

Εδώ δηλώνουμε μια ακέραια μεταβλητή τύπου unsigned long integer (32 bit). Μπορεί να αποθηκεύσει αριθμό από 0 ως 4.294.967.295 (ο απλός long integer, πάλι 32 bit, μπορεί να πάρει τιμές από -2.147.483.648 έως 2.147.483.647). Επειδή η τιμή της millis()πάει από 0 ως 4.294.967.295, χρειάζεται να έχουμε μια μεταβλητή αντίστοιχου τύπου για να χωράει να αποθηκεύσουμε την τιμή της.

ΣΗΜ.: Είναι απαραίτητο να δηλωθεί μια μεταβλητή πριν χρησιμοποιηθεί. Αλλιώς, όταν πάμε να μεταγλωττίσουμε το πρόγραμμα, θα σταματήσει η μεταγλώττιση και θα εμφανιστεί μήνυμα σφάλματος.

• millis()

Είναι ένας αριθμός που αυξάνεται συνέχεια σε ένα εσωτερικό «χρονόμετρο» του Arduino. Ξεκινάει να μετράει από το 0 όταν δώσουμε ρεύμα στο Arduino και αυξάνει την τιμή του κατά 1 κάθε χιλιοστό του δευτερολέπτου (κάθε 1 ms). Ξαναμηδενίζεται όταν φτάσει την τιμή 4.294.967.295 (σε περίπου 50 ημέρες) ή αν πατήσουμε το κουμπί reset.

• while(… συνθήκη){ … εντολές }

Αυτή είναι μια δομή επανάληψης με συνθήκη, γνωστή σε όσους έχουν κάνει προγραμματισμό. Η συνθήκη εξετάζεται και εφόσον είναι αληθής εκτελούνται οι εντολές που υπάρχουν μέσα στα άγκιστρα. Στη συνέχεια η εκτέλεση ξαναπάει στην αρχή της while και ελέγχεται πάλι η συνθήκη. Αν δεν ισχύει η συνθήκη κάποια στιγμή, οι εντολές που υπάρχουν στις αγκύλες δεν εκτελούνται και το πρόγραμμα προχωράει παρακάτω, μετά το άγκιστρο τέλους.

• Η δομή ελέγχου if(… συνθήκη) {… εντολές} else {… άλλες εντολές} είναι γνωστή από πριν.

Σύγκριση των δύο υλοποιήσεων (Α και Β)

Όπως μπορείτε να παρατηρήσετε, ενώ το LED που αναβοσβήνει λειτουργεί κανονικά, το LED που ελέγχεται από το κουμπί δεν έχει την αναμενόμενη συμπεριφορά.

Η δυσλειτουργία που παρατηρείται **στην Α υλοποίηση**, οφείλεται στο γεγονός ότι η κατάσταση του κουμπιού ελέγχεται στο πρόγραμμα μόνο μία στιγμή στην αρχή (εντολή if… else…), ενώ στα 6 δευτερόλεπτα που διαρκεί το αναβόσβημα του δεύτερου LED, το κουμπί δεν ελέγχεται και άρα οποιαδήποτε αλλαγή στην κατάστασή του αγνοείται. Για να μπορέσει λοιπόν το κουμπί να λειτουργήσει σωστά, θα έπρεπε η κατάστασή του να ελέγχεται συνέχεια κατά τη διάρκεια των δύο χρονοκαθυστερήσεων (εντολές delay). Αυτό δε μπορεί να γίνει αν χρησιμοποιήσουμε τις "delay" γιατί πολύ απλά το πρόγραμμα ΣΤΑΜΑΤΑΕΙ όταν εκτελείται μια εντολή delay.

Η λύση είναι η χρήση των millis(): **Στη Β υλοποίηση** παρακολουθούμε σε κάθε επανάληψη της loop αν πέρασε ο επιθυμητός χρόνος πριν κάνουμε την επόμενη ενέργεια. Αυτό έχει το πλεονέκτημα ότι το πρόγραμμα τρέχει συνεχώς (ή μάλλον «γυρίζει συνεχώς») μέσα στη loop και μπορεί να εξετάζει σε κάθε επανάληψη την κατάσταση του κουμπιού και να ανάψει αν χρειάζεται το LED. Επομένως το πρόγραμμα μπορεί να αντιδράσει σε πολύ σύντομο χρόνο αν πατήσουμε το κουμπί.

Εφαρμογή 4 : Φανάρι κυκλοφορίας

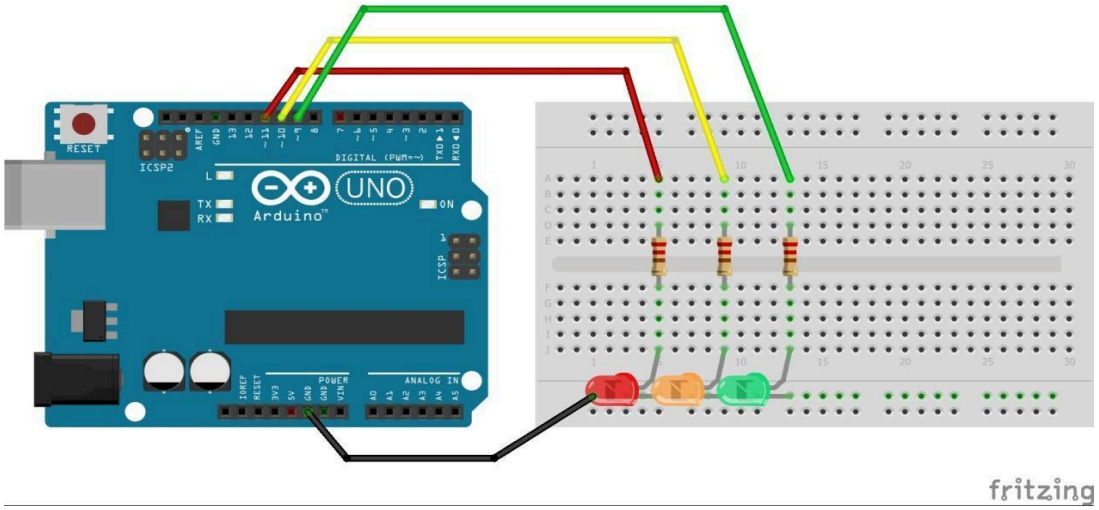
Η εφαρμογή που θα αναπτυχθεί θα προσομοιώνει τη λειτουργία ενός φαναριού κυκλοφορίας για αυτοκίνητα (μη ξεχνάτε ότι υπάρχουν και τα φανάρια πεζών). Συγκεκριμένα, θα χρησιμοποιηθούν τρία LED (κόκκινο, πορτοκαλί, πράσινο), τα οποία θα ανάβουν εναλλάξ με προκαθορισμένες διάρκειες. Στα πλαίσια της εφαρμογής, παρουσιάζεται η αξιοποίηση του breadboard σε πιο σύνθετα κυκλώματα και η έννοια του συγχρονισμού ενεργειών στο πρόγραμμα.

Υλικά

Θα χρησιμοποιηθούν: 1 κόκκινο LED, 1 πορτοκαλί ή κίτρινο LED, 1 πράσινο LED, 3 αντιστάσεις των 220 Ω και 4 καλώδια.

Κύκλωμα

Κατασκευάζουμε το κύκλωμα στο breadboard όπως δείχνει το παρακάτω σχήμα:



Στη συνέχεια, γράφουμε το παρακάτω πρόγραμμα στο Arduino IDE για να λειτουργήσουμε τα φανάρια:

Ανάπτυξη προγράμματος σε Arduino IDE

Το πρόγραμμα σε κώδικα C του Arduino:

```
int ledRed = 11;
int ledOrange = 10;
int ledGreen = 9;

void setup() {
pinMode(ledRed, OUTPUT);
pinMode(ledOrange, OUTPUT);
pinMode(ledGreen, OUTPUT);
}

void loop() {
// κόκκινο για 3 δευτερόλεπτα
digitalWrite(ledRed, HIGH);
digitalWrite(ledOrange, LOW);
digitalWrite(ledGreen, LOW);
delay(3000);

// πράσινο για 5 δευτερόλεπτα
digitalWrite(ledRed, LOW);
digitalWrite(ledOrange, LOW);
digitalWrite(ledGreen, HIGH);
delay(5000);
```

```
// πορτοκαλί για 1 δευτερόλεπτο
digitalWrite(ledRed, LOW);
digitalWrite(ledOrange, HIGH);
digitalWrite(ledGreen, LOW);
delay(1000);
}
```

Εντολές που χρησιμοποιούνται:

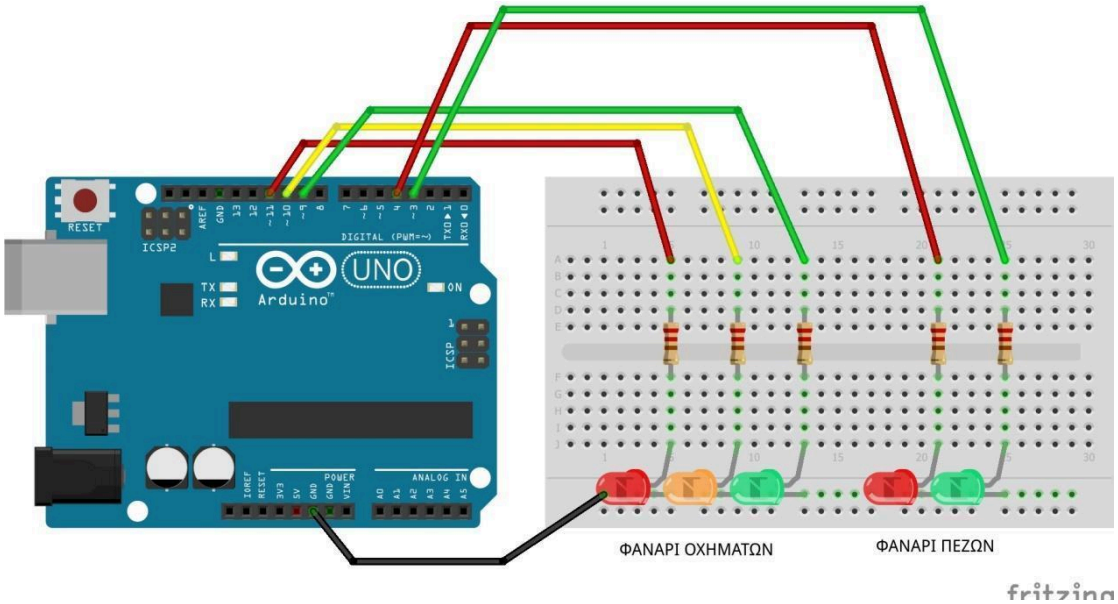
- Int …; Ορίζει μια ακέραιη μεταβλητή.

Optional: Εφαρμογή 2: Φανάρι κυκλοφορίας με φανάρι πεζών

Στην προηγούμενη εφαρμογή προσθέτουμε και ένα φανάρι για πεζούς.

Κύκλωμα

Κατασκευάζουμε το παρακάτω κύκλωμα βάζοντας επιπλέον 2 LED (κόκκινο και πράσινο) με αντιστάτες των 220Ω όπως δείχνει η παρακάτω εικόνα:



Στη συνέχεια, γράφουμε το παρακάτω πρόγραμμα στο Arduino IDE για να λειτουργήσουμε τα φανάρια πεζών:

Ανάπτυξη προγράμματος σε Arduino IDE

Το πρόγραμμα σε κώδικα C του Arduino:

```
int ledRed = 11;
int ledOrange = 10;
 int ledGreen = 9;
int pedRed = 4;
int pedGreen = 3;

void setup() {
pinMode(ledRed, OUTPUT);
pinMode(ledOrange, OUTPUT);
pinMode(ledGreen, OUTPUT);
 pinMode(pedRed, OUTPUT);
pinMode(pedGreen, OUTPUT);
}

void loop() {
// κόκκινο για 3 δευτερόλεπτα, πράσινο στους πεζούς
digitalWrite(ledRed, HIGH);
digitalWrite(ledOrange, LOW);
digitalWrite(ledGreen, LOW);
digitalWrite(pedRed, LOW);
 digitalWrite(pedGreen, HIGH);
 delay(3000);

// κόκκινο στους πεζούς, περιμένω για 1 δευτερόλεπτο πριν δώσω πράσινο στα αμάξια digitalWrite(pedRed, HIGH);
digitalWrite(pedGreen, LOW);
delay(1000);

// πράσινο κυκλοφορίας για 5 δευτερόλεπτα
digitalWrite(ledRed, LOW);
digitalWrite(ledOrange, LOW);
digitalWrite(ledGreen, HIGH);
delay(5000);

// πορτοκαλί για 1 δευτερόλεπτο
digitalWrite(ledRed, LOW);
digitalWrite(ledOrange, HIGH);
```

```
digitalWrite(ledGreen, LOW);
delay(1000);
}
```

Το Σειριακό Μόνιτορ (Serial Monitor) του Arduino

Το σειριακό μόνιτορ ή σειριακή οθόνη παρακολούθησης είναι μια επιπλέον δυνατότητα του Arduino IDE. Ανοίγει με το κουμπί που υπάρχει επάνω δεξιά στο IDE. Σε αυτό μπορούμε να λάβουμε και να δούμε γράμματα (χαρακτήρες) μέσω της θύρας USB (από το Arduino στον υπολογιστή), όπως επίσης και να στείλουμε γράμματα μέσω της θύρας USB (από τον υπολογιστή στο Arduino). Ένα απλό πρόγραμμα θα μας εξοικειώσει με τη χρήση του. Το πρόγραμμα απλά εκτυπώνει το μήνυμα Hello World ! στο σειριακό μόνιτορ και κάποιο άλλο μήνυμα που θέλουμε.

Οι εντολές που χρειαζόμαστε για να στείλουμε κάτι από το Arduino και να απεικονιστεί στο σειριακό μόνιτορ είναι βασικά δύο:

- Serial.print(….) ; Αυτή τυπώνει (ότι βάλαμε στην παρένθεση) αμέσως δίπλα σε ότι είχαμε εκτυπώσει πριν (δηλαδή ΚΟΛΛΗΜΕΝΟ).
- Serial.println(….) ; Αυτή τυπώνει δίπλα σε ότι είχαμε εκτυπώσει πριν και ΜΕΤΑ αλλάζει γραμμή (στέλνοντας αόρατους χαρακτήρες «αλλαγής γραμμής»)

Το πρόγραμμα για να δοκιμάσουμε τις εντολές είναι:

```
void setup(){
  Serial.begin(9600);   // ξεκίνησε το σειριακό μόνιτορ με ταχύτητα 9600 bps
  Serial.print("Hello World !");   // τύπωσε το μήνυμα στην οθόνη
  Serial.print("123456");   // τύπωσε κι αυτό (θα βγει ΚΟΛΛΗΜΕΝΟ με το προηγούμενο)
  Serial.println("Bye");   // τύπωσε αυτό και άλλαξε γραμμή μετά (πάλι ΚΟΛΛΗΜΕΝΟ)
 Serial.print("Hello again");   // τυπώνεται σε νέα γραμμή (προηγήθηκε: Serial.println)
}

void loop(){
}
```

ΠΑΡΑΤΗΡΗΣΗ: Η loop σε αυτό το πρόγραμμα δεν κάνει τίποτε. Είναι άδεια.

Ο φωτοαντιστάτης: Ένας αισθητήρας φωτός

Φωτοαντιστάτης

Ο φωτοαντιστάτης (Light Dependent Resistor, LDR) είναι ένας αντιστάτης, που η τιμή της αντίστασής του εξαρτάται από το φως που πέφτει πάνω του. Όσο πιο έντονο το φως, τόσο μικρότερη η αντίσταση. Ο φωτοαντιστάτης κατασκευάζεται από ειδικό φωτοευαίσθητο υλικό.

Εφαρμογή 1: LED που ανάβει όταν πέφτει το σκοτάδι

https://www.vodafonegenerationnext.gr/lessons/fota-poy-anaboyn-sto-skotadi

Στα πλαίσια της εφαρμογής αυτής οι μαθητές θα έχουν την ευκαιρία να κατασκευάσουν έναν αισθητήρα φωτός με τη χρήση ενός φωτοαντιστάτη (ή αλλιώς: φωτοαντίστασης). Στη συνέχεια, θα χρησιμοποιήσουν μία από τις αναλογικές εισόδους του Arduino για την ανάγνωση της τιμής του αισθητήρα. Η εφαρμογή που θα υλοποιηθεί, θα ανάβει αυτόματα ένα LED, όταν θα μειώνεται ο φωτισμός του χώρου.

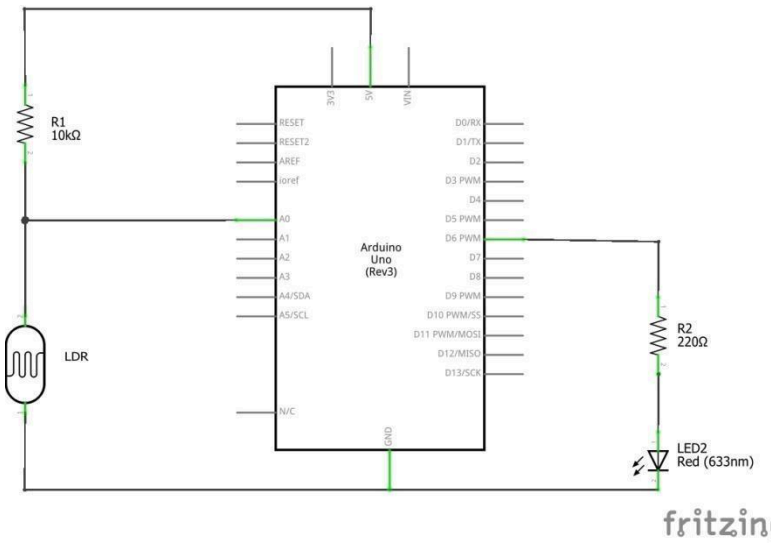
Υλικά

Τα νέα υλικά που θα εισάγουμε σε αυτή την εφαρμογή είναι ένας φωτοαντιστάτης και μία αντίσταση 10 kΩ. Επίσης χρησιμοποιούνται τα ήδη γνωστά: Κόκκινο LED, αντίσταση 220Ω, καλώδια.

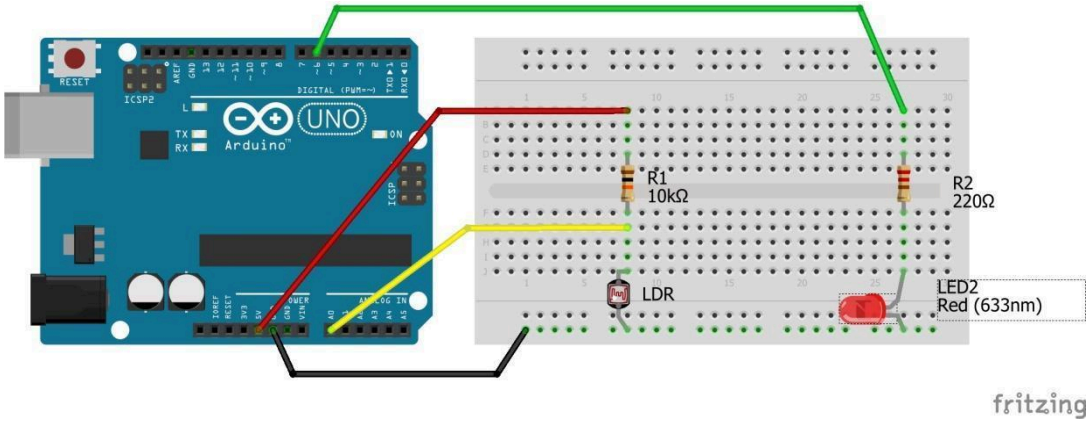
Κύκλωμα

Κατασκευάζουμε το κύκλωμα που δείχνουν οι επόμενες δύο εικόνες:

Σχηματικό διάγραμμα:



Πραγματική υλοποίηση:



Ανάπτυξη προγράμματος σε Arduino IDE

Το πρόγραμμα σε κώδικα C του Arduino:

```
// Πρόγραμμα για απλό πειραματισμό με φωτοαντιστάτη

int a;    // εδώ θα κρατάμε την τιμή που διαβάζουμε από τον φωτοαντιστάτη
float b;    // εδώ θα αποθηκεύουμε την τάση που έχει ο φωτοαντιστάτης στα άκρα του
float c;    // εδώ θα αποθηκεύουμε την τιμή της αντίστασης του φωτοαντιστάτη
int d;    // εδώ θα αποθηκεύσουμε την τιμή-όριο πάνω από την οποία θα ανάβουμε το LED

void setup() {
   Serial.begin(9600);     // ξεκινάμε το σειριακό μόνιτορ του Arduino IDE
   d=500; // τιμή πάνω από την οποία θα ανάβουμε το LED
  pinMode(6, OUTPUT);  // το πιν 6 θα είναι ή ΕΞΟΔΟΣ που θα συνδέσουμε το LED
   }

void loop() {
  a=analogRead(A0);     // διάβασε την αναλογική είσοδο A0 και βάλε την τιμή στην a
  Serial.print("Τιμή: ");   // απεικόνισε την τιμή της a στο σειριακό μόνιτορ του Arduino IDE
  Serial.print(a);
  Serial.print("\t");

    b=a*5.0/1023;         // υπολόγισε την τιμή της τάσης στα άκρα του φωτοαντιστάτη (αντιστοιχία: 1023 --> 5V)
```

```
 Serial.print("Τάση: ");
 Serial.print(b);    // απεικόνισε την τιμή της τάσης
 Serial.print(" V \t");

 c=b*10.0/(5-b);    // υπολόγισε την αντίσταση του φωτοαντιστάτη σε kΩ

 Serial.print(c);
 Serial.println(" kΩ");
 If(a>d)    // αν η τιμή του a είναι μεγαλύτερη από το d (όριο)
          digitalWrite(6, HIGH);   // άναψε το LED
else    // αλλιώς
          digitalWrite(6, LOW);   // σβήσε το LED
 delay(500); // περίμενε εδώ 0,5 δευτερόλεπτο
}
```

Εντολές που χρησιμοποιήθηκαν:

Χρησιμοποιήσαμε μια εντολή ανάγνωσης για αναλογική είσοδο:

- analogRead(A0): Αυτή διαβάζει την αναλογική είσοδο A0 και δίνει μια τιμή μεταξύ 0 και 1023, που αντιστοιχεί σε τάση από 0 ως 5V.

Χρησιμοποιήσαμε κάποιες εντολές για το Serial Monitor του Arduino IDE.

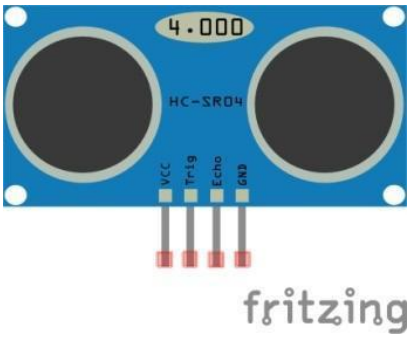
- Serial.begin(9600); Αυτή ξεκινάει την επικοινωνία μεταξύ Arduino και Serial Monitor στο IDE. Το «9600» είναι η ταχύτητα επικοινωνίας (9600 bps).
- Serial.print(….) ; Αυτή τυπώνει ότι έχουμε στην παρένθεση.
- Serial.println(….) ; Αυτή τυπώνει ότι έχουμε στην παρένθεση και ΜΕΤΑ αλλάζει γραμμή (στέλνοντας έναν αόρατο χαρακτήρα «αλλαγής γραμμής»)

ΠΑΡΑΤΗΡΗΣΗ: Το κύκλωμα χρησιμοποιεί έναν διαιρέτη τάσης αποτελούμενο από την αντίσταση 10kΩ και τη φωτοαντίσταση (βρίσκεται στην αριστερή άκρη του σχηματικού διαγράμματος). Χρησιμοποιώντας τον διαιρέτη τάσης πετυχαίνουμε να πάρουμε μια τάση που αλλάζει, εξαρτούμενη από το φως που πέφτει στη φωτοαντίσταση. Αυτή την τάση μετράει το Arduino μέσω της αναλογικής εισόδου A0.

Ο αισθητήρας απόστασης HC-SR04

Τι είναι

Ο HC-SR04 είναι ένας αισθητήρας απόστασης υπερήχων. Όπως φαίνεται και στην Εικόνα, διαθέτει έναν πομπό και ένα δέκτη υπερήχων, καθώς και 4 ακροδέκτες (επαφές σύνδεσης). Οι δύο ακριανοί ακροδέκτες VCC και GND, χρησιμοποιούνται για την τροφοδοσία του αισθητήρα και συνδέονται στην τάση (5V) και τη γείωση αντίστοιχα. Ο ακροδέκτης Trig χρησιμοποιείται για την εκκίνηση της διαδικασίας μέτρησης και ο ακροδέκτης Echo χρησιμοποιείται για την έξοδο του αποτελέσματος. Οι δύο αυτοί ακροδέκτες συνδέονται σε δύο ψηφιακές ακίδες (πιν) του Arduino.



Πώς λειτουργεί

Για να ξεκινήσει η διαδικασία της μέτρησης, πρέπει να στείλουμε στον ακροδέκτη Trig έναν παλμό High με διάρκεια τουλάχιστον 10 μsec. Μόλις ο αισθητήρας λάβει το σήμα ενεργοποίησης, στέλνει από τον πομπό μια ακολουθία υπερήχων. Όταν οι υπέρηχοι συναντήσουν κάποιο εμπόδιο αντανακλώνται και επιστρέφουν προς τον αισθητήρα, όπου και ανιχνεύονται από το δέκτη. Στη συνέχεια, ο αισθητήρας βγάζει ως έξοδο στον ακροδέκτη Echo έναν παλμό HIGH. Η διάρκεια του παλμού είναι ίση με το χρόνο που πέρασε από τη στιγμή της εκπομπής των υπερήχων, μέχρι τη λήψη της αντανάκλασης.

Πώς υπολογίζεται η απόσταση από το εμπόδιο

Το Arduino με κατάλληλες εντολές μετράει τη διάρκεια του παλμού που βγάζει ως έξοδο ο αισθητήρας, έστω *duration*. Με δεδομένο ότι οι υπέρηχοι ταξιδεύουν με την ταχύτητα του ήχου (340m/s = 0,034cm/μs) και με βάση τον τύπο της ταχύτητας (u=s/t), αν *distance* είναι η απόσταση από το εμπόδιο έχουμε:

$$0{,}034 = \frac{\square \times 2}{\square} \quad \square = \frac{0{,}034 \times \square}{2} \approx \frac{\square}{59}$$

Η διαίρεση με το 2, προκύπτει από το γεγονός ότι η διάρκεια του παλμού αντιστοιχεί στο χρόνο που έκαναν οι υπέρηχοι να πάνε μέχρι το εμπόδιο και να γυρίσουν πίσω στον αισθητήρα. Άρα η απόσταση που καλύπτουν οι υπέρηχοι σε αυτό το χρόνο, είναι η διπλάσια από αυτήν που θέλουμε να υπολογίσουμε.

Εφαρμογή 2: Μέτρηση απόστασης και απεικόνιση στο Serial Monitor

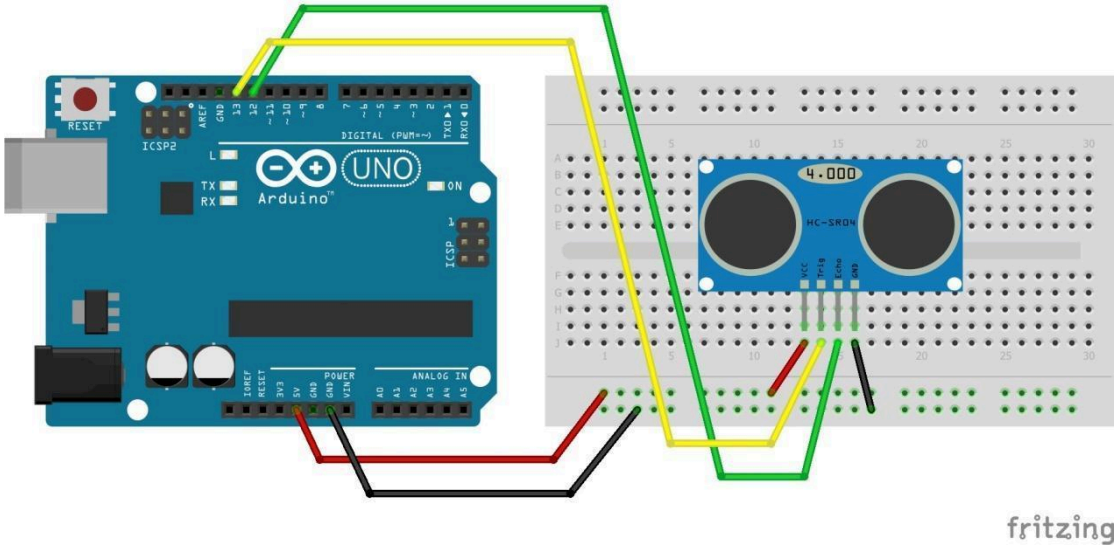
Στην εφαρμογή αυτή θα μετράμε απόσταση από κάποιο αντικείμενο με τον αισθητήρα υπερήχων και θα την εμφανίζουμε στο Serial Monitor του Arduino IDE.

Υλικά

Χρειαζόμαστε: την πλακέτα HC-SR04 (αισθητήρας απόστασης με υπερήχους), Arduino UNO, breadboard, Καλώδια jumper

Κύκλωμα

Κατασκευάζουμε το παρακάτω κύκλωμα:



Ανάπτυξη προγράμματος σε Arduino IDE

Το πρόγραμμα σε κώδικα C του Arduino:

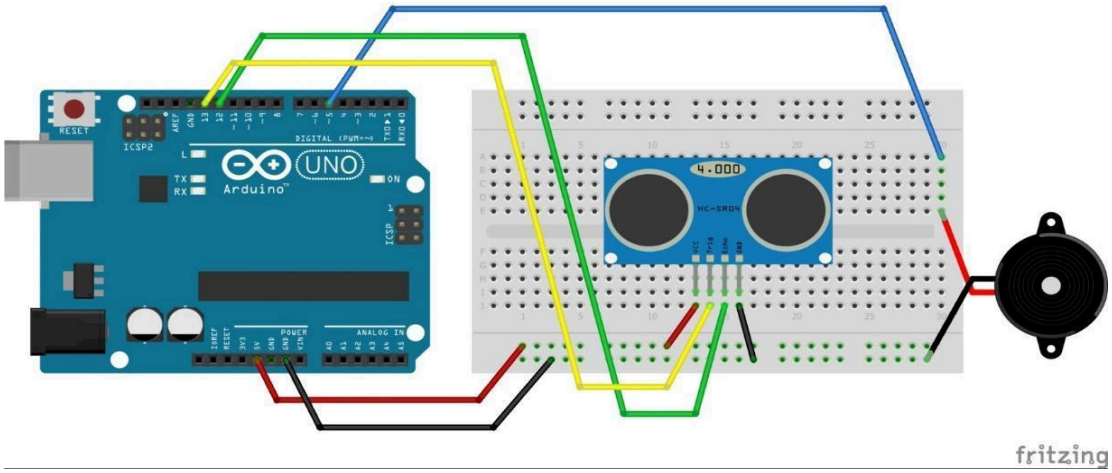
```
int echoPin = 12; // Echo Pin
int trigPin = 13; // Trigger Pin
int maximumRange = 200; // Maximum range needed
int minimumRange = 0; // Minimum range needed
long duration, distance; // Duration used to calculate distance

void setup() {
pinMode(trigPin, OUTPUT);
pinMode(echoPin, INPUT);
Serial.begin(9600);
}

void loop() {
digitalWrite(trigPin, LOW);
delayMicroseconds(2);
digitalWrite(trigPin, HIGH);
```

```
delayMicroseconds(10);
digitalWrite(trigPin, LOW);
duration = pulseIn(echoPin, HIGH);
//Υπολογισμός απόστασης (σε cm) βασιζόμενοι στην ταχύτητα του ήχου.
distance = duration/58.2;
if (distance >= maximumRange || distance <= minimumRange){
Serial.println("Εκτός ορίων");
delay(100);
}
else {
Serial.println(distance);
delay(50);
};

//Καθυστέρηση 50ms πριν την επόμενη ανάγνωση
delay(50);
}
```

Optional Εφαρμογή 1: Συναγερμός προσέγγισης με αισθητήρα υπερήχων και βομβητή (buzzer)

Στην εφαρμογή αυτή ο αισθητήρας υπερήχων ανιχνεύει την απόσταση από κάποιο αντικείμενο που πλησιάζει και ηχεί συναγερμός με τον βομβητή όταν το αντικείμενο πλησιάσει πολύ κοντά. Μπορεί να χρησιμοποιηθεί ως αισθητήρας προσέγγισης π.χ. σε κάποιο όχημα.

Κατασκευάζουμε το παρακάτω κύκλωμα με βάση το προηγούμενο. Προσθέτουμε μόνο τον βομβητή και κάποια καλώδια.



Κώδικας σε γλώσσα C του Arduino:

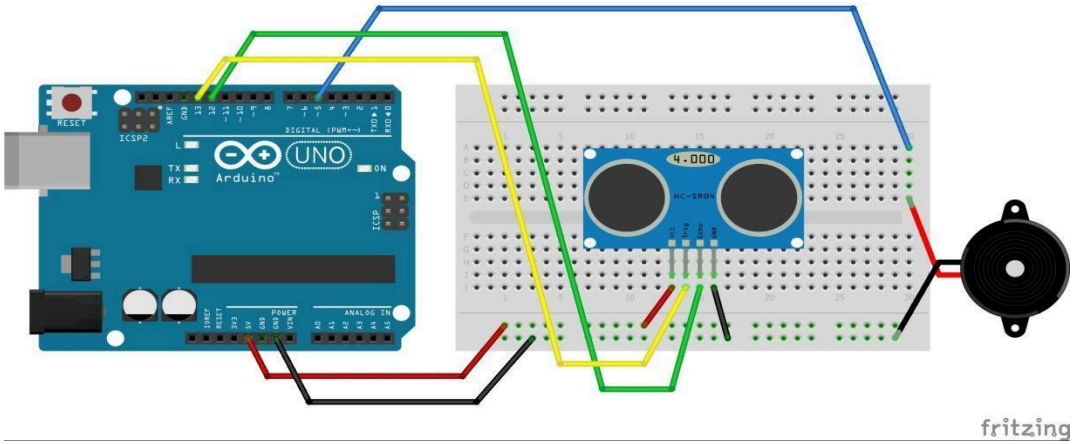
```
int echoPin = 12; // Πιν Echo της πλακέτας υπερήχων
int trigPin = 13; // Πιν Trigger της πλακέτας υπερήχων
int alarmPin = 5; // Βομβητής
int maximumRange = 200; // Μέγιστη επιτρεπόμενη εμβέλεια
int minimumRange = 0; // Ελάχιστη επιτρεπόμενη εμβέλεια
int alarmRange=10; // Η απόσταση στην οποία θα χτυπήσει συναγερμός
float duration, distance;
void setup() {
  pinMode(trigPin, OUTPUT);
  pinMode(echoPin, INPUT);
  pinMode(alarmPin, OUTPUT);
  Serial.begin(9600);
}

void loop() {
  digitalWrite(trigPin, LOW);
  delayMicroseconds(2);
  digitalWrite(trigPin, HIGH);
  delayMicroseconds(10);
  digitalWrite(trigPin, LOW);
  duration = pulseIn(echoPin, HIGH);
   //Υπολογισμός απόστασης (σε cm) βασιζόμενοι στην ταχύτητα του ήχου.
  distance = duration/58.2;
 if (distance >= maximumRange || distance <= minimumRange){
    Serial.println("Εκτός ορίων");
    delay(100);
  }
  else {
    Serial.println(distance,2);
    if(distance<alarmRange)
      digitalWrite(5,HIGH); // Άναψε το βομβητή
    else
      digitalWrite(5,LOW); // Σβήσε το βομβητή
   }
   delay(50);  //Καθυστέρηση 50ms πριν την επόμενη ανάγνωση
```

}

Optional Εφαρμογή 2: Βελτιωμένος συναγερμός προσέγγισης με αισθητήρα υπερήχων και βομβητή (buzzer)

Στην εφαρμογή αυτή χρησιμοποιούμε το ίδιο κύκλωμα με την Εφαρμογή 6α, αλλά βελτιωμένο λογισμικό για να αποφύγουμε περιστασιακούς παρασιτικούς συναγερμούς. Το λογισμικό παίρνει αρκετές μετρήσεις και εξάγει μέσο όρο πριν αποφασίσει να χτυπήσει συναγερμό.

Κατασκευάζουμε το παρακάτω κύκλωμα, ίδιο με το προηγούμενο.



Κώδικας σε γλώσσα C του Arduino:

```
// Πρόγραμμα που ηχεί βομβητή
// όταν πλησιάσει κάποιος κοντά
// στον αισθητήρα απόστασης με υπερήχους.
// Βελτίωση: Απορρίπτει περιστασιακές παρασιτικές τιμές.

int echoPin = 12; // Πιν Echo της πλακέτας υπερήχων
int trigPin = 13; // Πιν Trigger της πλακέτας υπερήχων
int alarmPin = 5; // Βομβητής  int a[20]; // Πίνακας-αποθήκη 20 τελευταίων τιμών απόστασης
int alarmRange=10; // Η απόσταση στην οποία θα χτυπήσει συναγερμός σε cm
unsigned long duration;
bool ignore; // σημαία για το αν θα αγνοηθεί μια τιμή
int i, sum, counter=0;
int mo, maxRange=500, minRange=6; // μέσος όρος, όρια παρασιτικών (παράλογων) τιμών
int distance; // απόσταση σε cm (ακέραιος. τα δεκαδικά δε μας ενδιαφέρουν)
```

```
void setup() {
  pinMode(trigPin, OUTPUT);
  pinMode(echoPin, INPUT);
  pinMode(alarmPin, OUTPUT);
  Serial.begin(9600);
}

void loop() {
  digitalWrite(trigPin, LOW);
  delayMicroseconds(2);
  digitalWrite(trigPin, HIGH);
  delayMicroseconds(10);
  digitalWrite(trigPin, LOW);
  duration = pulseIn(echoPin, HIGH);

  //Υπολογισμός απόστασης (σε cm) βασιζόμενοι στην ταχύτητα του ήχου.
  distance = duration/58.2;
  sum=0;
  for(i=0; i<19; i++) // ολίσθησε αριστερά τις 19 τιμές στον πίνακα
   {
     a[i]=a[i+1];
     sum=sum+a[i]; // πρόσθεσε την τιμή στο άθροισμα τιμών
   }
  if(distance>maxRange || distance<minRange) // αν η τιμή είναι εκτός λογικών ορίων...
    distance=mo; // βάλε το μέσο όρο στη θέση της

  sum=sum+distance; // τώρα το άθροισμα έχει όλες τις τιμές του πίνακα ενημερωμένου

    a[19]=distance; // αντικατάστησε την τελευταία θέση του a[] με την πρόσφατη τιμή απόστασης

  mo=sum/20; // μέσος όρος των 20 τιμών απόστασης
   if(distance>mo+mo/4 || distance<mo-mo/4) // αν η πρόσφατη τιμή απέχει πολύ από το μέσο όρο...
    ignore=true; // σημείωσε να μην ενεργοποιήσει τον συναγερμό
  else
    ignore=false;
```

```
  Serial.print("counter=");
  Serial.println(counter);
  Serial.print("mo=");
  Serial.println(mo);
  Serial.println(distance); // εκτύπωσε απόσταση στο μόνιτορ του IDE

  if(distance<alarmRange && counter>=20 && ignore==false)
  // δε χτυπάμε συναγερμό για τις 20 πρώτες μετρήσεις
  // ή αν η ignore είναι true
    digitalWrite(5,HIGH); // Άναψε το βομβητή
  else
    digitalWrite(5,LOW); // Σβήσε το βομβητή

  counter=counter+1; // αύξησε το μετρητή
  if(counter>20) counter=20; // αν πέρασε το 20, κράτα τον στο 20
  delay(20);  //Καθυστέρηση πριν την επόμενη επανάληψη
}
```

Η οθόνη LCD

Η οθόνη LCD 16x2 είναι μια οθόνη που απεικονίζει 2 σειρές των 16 χαρακτήρων. Εμείς χρησιμοποιούμε μια οθόνη LCD με επικοινωνία IIC ή I2C. Η οθόνη μας να έχει πίσω από την κεντρική πλακέτα της και μία μικρότερη πλακέτα που είναι το κύκλωμα για την επικοινωνία I2C μεταξύ οθόνης και Arduino. Η σύνδεση με αυτό το πρωτόκολλο επικοινωνίας απαιτεί μόνο 4 καλώδια μεταξύ των δύο συσκευών που αναφέραμε: +5V, GND, SDA, SCL.

Εφαρμογή 3 : Οθόνη LCD με σύνδεση I2C και “Hello world!”

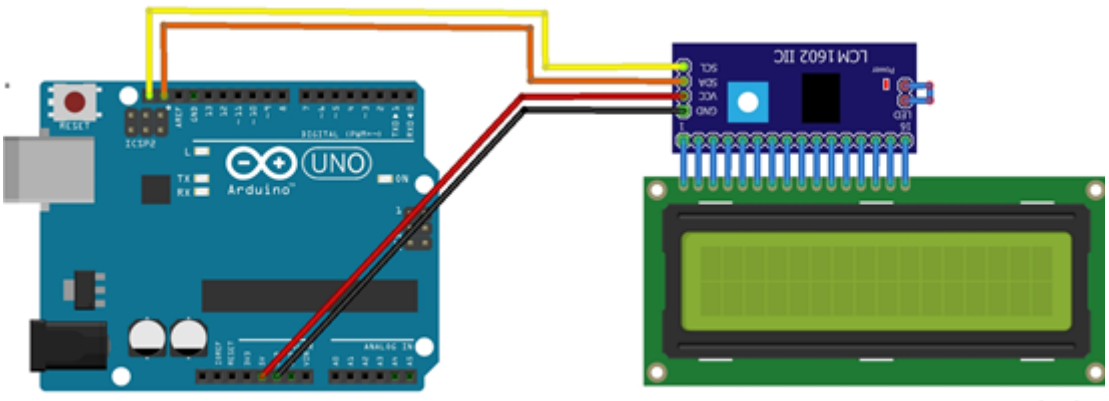
Στην εφαρμογή αυτή εμφανίζουμε στην οθόνη το γνωστό μήνυμα πρώτου πειραματισμού «Hello world!”.

Υλικά

Οθόνη LCD I2C, Arduino UNO, Καλώδια jumper

Κύκλωμα

Κατασκευάζουμε το κύκλωμα που δείχνει η παρακάτω εικόνα:



Ανάπτυξη προγράμματος σε Arduino IDE

Το πρόγραμμα “Hello World” για δοκιμή της οθόνης I2C LCD είναι:

```
// I2C LCD screen demo
//Compatible with the Arduino IDE 1.0
//Library version:1.1
#include <Wire.h>
#include <LiquidCrystal_I2C.h>

LiquidCrystal_I2C lcd(0x27,16 ,2);  // set the LCD address to 0x27, set 16 chars and 2 line display

void setup()
{
  lcd.init();                  // initialize the lcd
  // Print a message to the LCD.
  lcd.backlight();
  lcd.setCursor(1,0);
  lcd.print(" Hello, world! ");
  lcd.setCursor(0,1);
  lcd.print(" I2C LCD screen.");
}


void loop()
{
}
```

Αντιμετώπιση προβλημάτων

No1: Η οθόνη LCD δεν φωτίζεται

Δεν ανάβει η οθόνη LCD; (εννοείται ότι κάνατε σωστά τις 4 συνδέσεις με το Arduino). Γυρίστε την οθόνη σας από πίσω. Έχει ένα μαύρο πλαστικό βραχυκυκλωτήρα (jumper) στην άκρη της μικρής πλακέτας; ¨Η βλέπετε δύο γυμνά πιν στον αέρα; Στη δεύτερη περίπτωση πρέπει να τα ενώσετε με ένα κομμάτι καλώδιο ή και να τα κολλήσετε με κολλητήρι (ή να βάλετε ένα βραχυκυκλωτήρα, φυσικά). Αυτά τα πιν πρέπει να είναι ενωμένα για να ανάβει ο φωτισμός της οθόνης.

Παρατηρήθηκε από εκπαιδευτή σε προηγούμενο κύκλο να ΛΕΙΠΕΙ ο βραχυκυκλωτήρας από τη μικρή πλακέτα! Κανονικά πρέπει να υπάρχει στη θέση του από το εργοστάσιο. Μπορείτε να βρείτε βραχυκυκλωτήρες (πέρα από καταστήματα ηλεκτρονικών) από κάποια παλιά μητρική πλακέτα υπολογιστή (στα εργαστήρια πληροφορικής έχει κάποιες χαλασμένες που υπάρχουν για επίδειξη).

No2: Η οθόνη LCD δεν εμφανίζει κείμενο

1. Φορτώσατε την εφαρμογή στο Arduino αλλά δεν εμφανίζετε τίποτε στην οθόνη; Το πρώτο που πρέπει να κάνετε είναι να αποσυνδέσετε το Arduino από την τάση και να ελέγξετε ξανά τις συνδέσεις σας.
2. Αν οι συνδέσεις σας είναι εντάξει και δεν εμφανίζεται τίποτε στην οθόνη: Δοκιμάστε να στρέψετε το μικρό ποτενσιόμετρο τρίμερ που υπάρχει στην πίσω πλευρά της οθόνης, επάνω στη μικρή πλακέτα. Ταυτόχρονα ελέγχετε αν η εικόνα εμφανίζεται. Αυτό το τρίμερ ρυθμίζει το κοντράστ (αντίθεση) της εικόνας και σε κάποιες οθόνες δεν είναι σωστά ρυθμισμένο από το εργοστάσιο.
3. Αν και πάλι δεν μπορείτε να εμφανίσετε κείμενο στην οθόνη ενώ τρέξατε την εφαρμογή (και δοκιμάσατε να στρέψετε το τρίμερ ποτενσιόμετρο στην πίσω πλευρά και πάλι τίποτε), μπορεί η διεύθυνση επικοινωνίας οθόνης -Arduino μέσω I2C να είναι άλλη και όχι η 0x27 (που έχουν οι περισσότερες που στάλθηκαν από την SciCo). Τότε, απλά ανιχνεύουμε τη διεύθυνση που έχει η οθόνη με ένα πρόγραμμα I2C scanner. Υπάρχει το λινκ παρακάτω. Αντιγραφή - επικόλληση στο IDE και τρέξτε το. Θα σας πει ποια διεύθυνση πρέπει να χρησιμοποιείτε στα προγράμματά σας με την "δύσκολη" οθόνη LCD:

https://playground.arduino.cc/Main/I2cScanner?fbclid=IwAR2o_iTSEALTb0bk8iT9vDq-lnm9HXk5z7pUmWY-rHZrBdFq_VqNJcZnz_c

ή απλά κάντε έρευνα στο διαδίκτυο για ένα i2c scanner για Arduino.

Εφαρμογή 4: Γράφω κείμενο στο Serial monitor και εμφανίζεται την οθόνη LCD

Ένα ακόμη πρόγραμμα δοκιμής είναι αυτό, που μας επιτρέπει να γράφουμε στην οθόνη LCD ότι μήνυμα θέλουμε, πληκτρολογώντας το στο παράθυρο εισόδου του σειριακού μόνιτορ:

```
#include <Wire.h>
```

```
#include <LiquidCrystal_I2C.h>
LiquidCrystal_I2C lcd(0x27,16,2);
void setup()
{
 lcd.init();                  // initialize the lcd
 lcd.backlight();
 Serial.begin(9600);
}


void loop()
{
 // when characters arrive over the serial port...
 if (Serial.available()) {
  // wait a bit for the entire message to arrive
  delay(100);
  // clear the screen
  lcd.clear();
  // read all the available characters
  while (Serial.available() > 0) {
   // display each character to the LCD
   lcd.write(Serial.read());
  }
 }
}
```

Εφαρμογή 5: Εμφανίζω στην οθόνη LCD ένα μήνυμα για το Generation Next

Ένα τρίτο πρόγραμμα δοκιμής είναι αυτό, ειδικά για το «**Generation Next**»:

```
#include <Wire.h>
#include <LiquidCrystal_I2C.h>
//I2C pins declaration
LiquidCrystal_I2C lcd(0x3F, 16, 2);

void setup()
{
```

```
lcd.init();//Defining 16 columns and 2 rows of lcd display
lcd.backlight();//To Power ON the back light
//lcd.backlight();// To Power OFF the back light
}
void loop()
{
lcd.setCursor(0,0); //Defining positon to write from first row, first column .
lcd.print(" Generation Next"); // You can write 16 Characters per line .
delay(1000);//Delay used to give a dynamic effect
lcd.setCursor(0,1);  //Defining positon to write from second row, first column .
lcd.print("    Youth     ");
delay(8000);
lcd.clear();//Clean the screen
lcd.setCursor(0,0);
lcd.print(" I2C LCD Screen ");
lcd.setCursor(0,1);
lcd.print("     DEMO     ");
delay(8000);
}
```

Σερβομηχανισμός (servo): Ένας μηχανισμός που κινεί

Ο σερβομηχανισμός (servo) είναι μια μηχανική διάταξη που περιστρέφει ένα μικρό πλαστικό βραχίονα στην επιθυμητή από εμάς θέση. Η συνήθης χρήση σερβομηχανισμών είναι σε τηλεκατευθυνόμενα μοντέλα. Πλέον χρησιμοποιούνται και σε κατασκευές με Arduino και η τιμή τους είναι πολύ χαμηλή. Είναι ο πιο εύκολος τρόπος να κινήσουμε «εμπρός πίσω» κάποιο μηχανικό μέρος ή να περιστρέψουμε.

Σερβομηχανισμός (servo)

Ο σερβομηχανισμός έχει 3 ακροδέκτες για σύνδεση. Συνήθως είναι χρωματισμένα τα αντίστοιχα καλώδια ως εξής: καφέ (GND), κόκκινο (+5V), πορτοκαλί (σήμα χειρισμού, είσοδος του σερβομηχανισμού).

Ένα τυπικό servo συνοδεύεται από βραχίονες (λευκούς πλαστικούς συνήθως, σε διάφορα σχήματα: σταυροειδή, ραβδόμορφο με δύο σκέλη, ραβδόμορφο με ένα σκέλος), βίδες.

Για τον έλεγχο του σερβομηχανισμού πρέπει να στέλνουμε ένα σήμα με παλμούς μεταβαλλόμενου χρονικού πλάτους (PWM Pulse Width Modulation). Αυτό είναι αρκετά εύκολο να γίνει με το Arduino με τη βοήθεια της αντίστοιχης βιβλιοθήκης Servo.

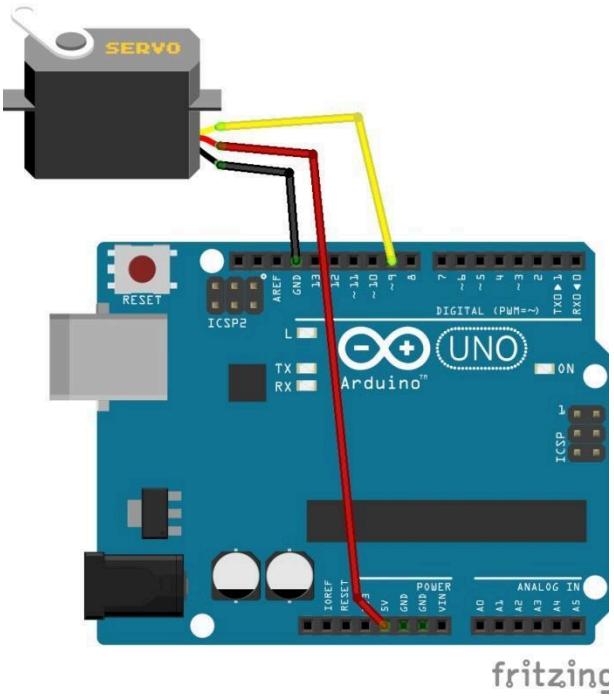
Εφαρμογή 6: Σερβομηχανισμός που περιστρέφεται δεξιά αριστερά

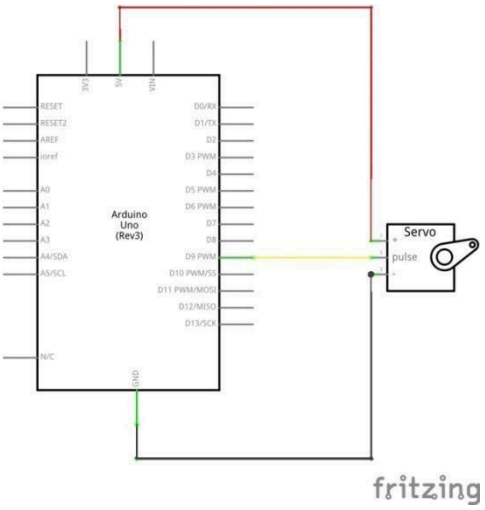
Υλικά

Χρειαζόμαστε: Arduino UNO, servo, καλώδια jumper.

Κύκλωμα

Το κύκλωμα που θα κατασκευάσουμε είναι το παρακάτω:





Το πρόγραμμα σε Arduino C για πειραματισμό με το servo είναι:

```
// Πειραματισμός με servo No1
// το servo περιστρέφεται από τη μία ακραία θέση του στην άλλη και πάλι πίσω

#include <Servo.h> // Συμπερίλαβε τη βιβλιοθήκη του σέρβο

Servo myservo; // Δημιούργησε ένα αντικείμενο τύπου Servo

int s=1; // Το βήμα (μοίρες) που θα αυξάνουμε τη θέση του σέρβο σε κάθε επανάληψη

void setup()
```

```
{
  myservo.attach(9);   //  Σχετίζει το σέρβο με το πιν 9
}

void loop()
{
  for(int i=0; i<=180; i=i+s){
    myservo.write(i); //  Γύρισε τον άξονα του servo στην τιμή i
    delay(15); // Περίμενε λίγο για να γυρίσει το servo στην τιμή που θέλουμε
  }
  for(int i=180; i>=0; i=i-s){
    myservo.write(i); //  Γύρισε τον άξονα του servo στην τιμή i
    delay(15); // Περίμενε λίγο για να γυρίσει το servo στην τιμή που θέλουμε
  }
}
```

Optional: Εφαρμογή 3: Αισθητήρας θερμοκρασίας LM35

Το LM35 είναι ένα ολοκληρωμένο κύκλωμα μέτρησης θερμοκρασίας. Είναι η πιο δημοφιλής επιλογή όταν θέλουμε να μετρήσουμε τη θερμοκρασία με το Arduino διότι είναι εύκολο στη σύνδεση και πολύ οικονομικό στην τιμή. Κυκλοφορεί σε κέλυφος πλαστικό μαύρο με τρεις ακροδέκτες (ή «πόδια» όπως κοινά τα αναφέρουμε).

Το LM35 έχει 3 ακροδέκτες που συνδέονται κατά σειρά (όπως βλέπουμε την πρόσοψη του εξαρτήματος): +5V, Έξοδος, GND.

Ο αισθητήρας θερμοκρασίας LM35

Το ολοκληρωμένο αυτό παράγει μια τάση που είναι ανάλογη της θερμοκρασίας. Επομένως πρέπει να συνδεθεί σε μια αναλογική είσοδο του Arduino.

Για να λειτουργήσει απλά συνδέουμε τους ακροδέκτες του αντίστοιχα στο: +5V, GND και ο τρίτος ακροδέκτης συνδέεται σε μια αναλογική είσοδο του Arduino (A0 ως A5).

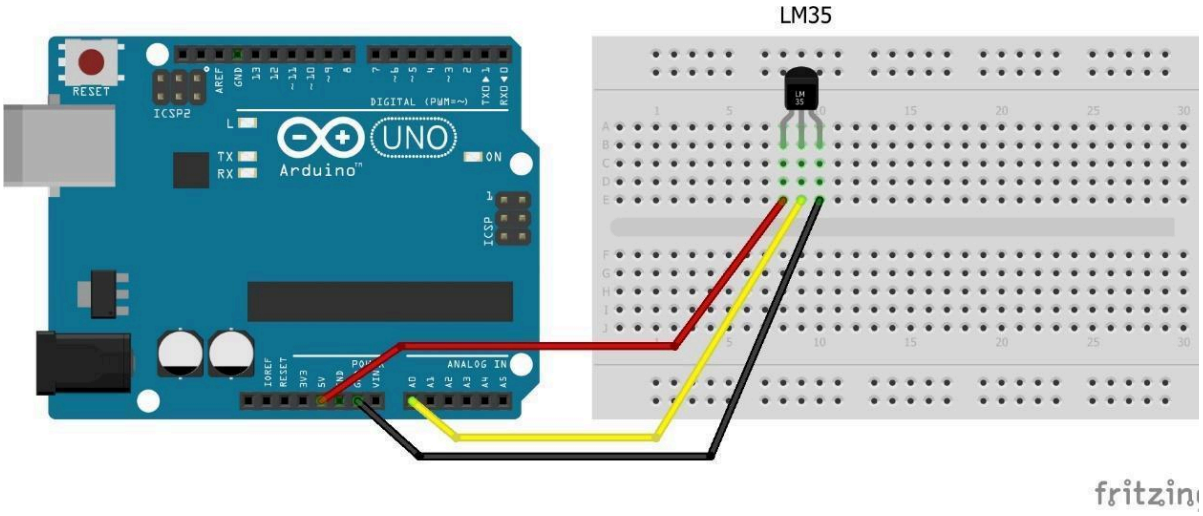
Σε αυτή την εφαρμογή θα μετρήσουμε τη θερμοκρασία με τον αισθητήρα LM35. Ο αισθητήρας αυτός βγάζει μια τάση που είναι αντίστοιχη με τη θερμοκρασία του. Με ένα απλό τύπο μπορούμε να υπολογίσουμε από αυτή την τάση την τιμή της θερμοκρασίας και να την απεικονίσουμε στο σειριακό μόνιτορ (ή σε κάποια οθόνη LCD).

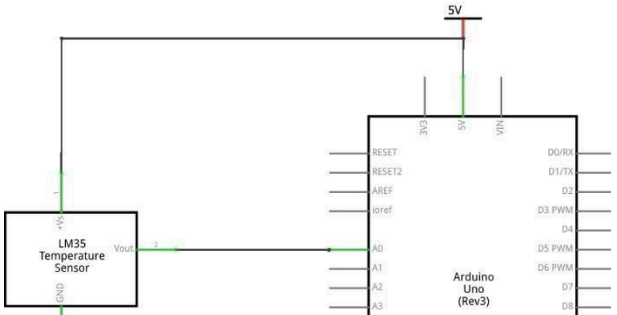
Υλικά

Arduino UNO, LM35, καλώδια jumper

Συνδεσμολογία

Κατασκευάζουμε το κύκλωμα όπως δείχνουν οι παρακάτω εικόνες:





Ανάπτυξη προγράμματος σε Arduino IDE

Το πρόγραμμα που θα χρησιμοποιήσουμε είναι:

```
int a;    // Εδώ αποθηκεύουμε την τιμή που διαβάσαμε από την αναλογική είσοδο A0 (από 0 ως 1023)
float temp;   // Εδώ αποθηκεύουμε την τιμή της θερμοκρασίας

void setup() {
  Serial.begin(9600);
}

void loop() {

  a=analogRead(A0);    // Διάβασε την τιμή από την A0

    temp = (5.0 * analogRead(A0) * 100.0) / 1023;      // Υπολόγισε την θερμοκρασία σε βαθμούς Κελσίου

   Serial.print(temp);      // Απεικόνισε τη θερμοκρασία στο σειριακό μόνιτορ του Arduino IDE
  Serial.println(" C");

  delay(1000);     // Περίμενε εδώ 1 δευτερόλεπτο

}
```

Εφαρμογή έμπνευσης από τους μαθητές των Χανίων

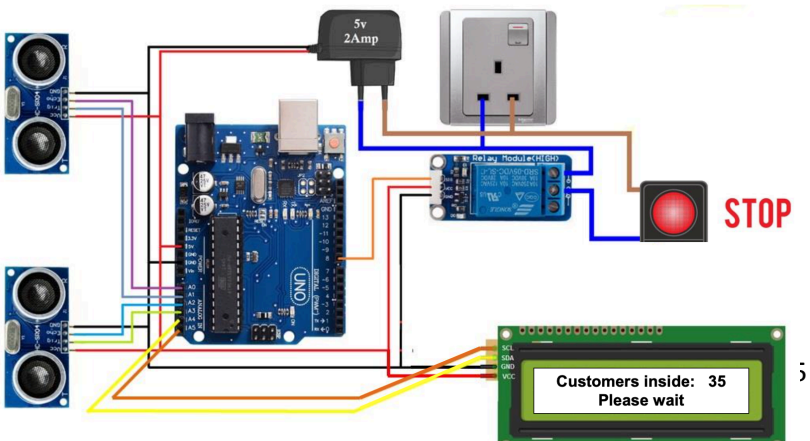
Σύστημα καταγραφής πελατών Safe Entry από τους Upcoming Scientists

Περιγραφή προβλήματος:
Το πρόβλημα το οποίο προσπαθούμε να επιλύσουμε είναι η μη ασφαλής είσοδος στα super markets λόγω συνωστισμού και παραβίασης των μέτρων που καθορίζουν τον αριθμό των πελατών που επιτρέπονται σε ένα κατάστημα ανάλογα με τα τετραγωνικά του. Ελπίζουμε ότι με μια συσκευή σαν τη δική μας θα περιορίσουμε την μετάδοση του COVID-19 σε πολυσύχναστους χώρους όπως τα super markets.

Περιγραφή προτεινόμενης λύσης:
Η πρόταση μας αποτελείται από δύο μέρη. Το 1ο και σημαντικότερο κομμάτι της είναι η συσκευή μας "Safe Entry" η οποία λειτουργεί σαν μετρητής των πελατών που βρίσκονται ανά πάσα στιγμή μέσα στο κατάστημα. Μετράει αυτούς που εισέρχονται και εξέρχονται κρατώντας πάντα λογαριασμό για το πόσοι βρίσκονται κάθε χρονική στιγμή μέσα στο κατάστημα. Το 2ο κομμάτι είναι ο αυτόματος μηχανισμός παροχής αντισηπτικού με το οποίο οι πελάτες θα βάζουν αντισηπτικό στα χέρια τους ανέπαφα, αποφεύγοντας έτσι τον κίνδυνο μετάδοσης μέσω των επαφών.

Κυκλωματική διάταξη

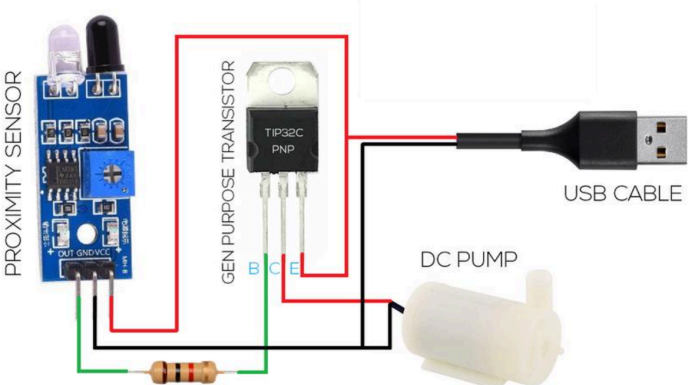
Safe Entry – 6ο Γυμνάσιο Χανίων



6ο Γυμνάσιο Χανίων

Upcoming Scientists

Safe Entry automatic alcohol dispenser

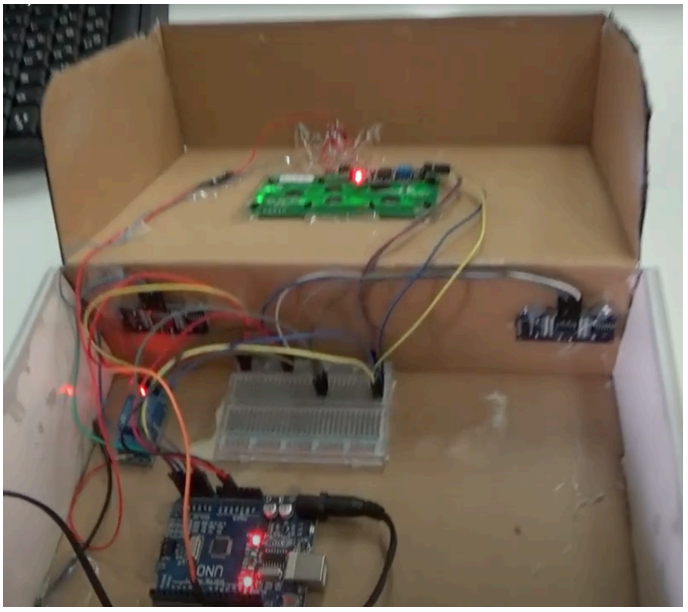


Κώδικας

```
// Safe Entry – 6ο Gymnasio –  Σταύρος, Φάνης, Κωνσταντίνος, Γιάννης
#include <Wire.h>
#include <LiquidCrystal_I2C.h>
LiquidCrystal_I2C lcd(0x27, 20, 4);

#define e_s1 A0 //echo pin
#define t_s1 A1 //Trigger pin
#define e_s2 A2 //echo pin
#define t_s2 A3 //Trigger pin
int relay = 8; // Out for light
long dis_a=0,dis_b=0;
int flag1=0, flag2=0;
int person = 0;
//**********************ultra_read*****************************
void ultra_read(int pin_t,int pin_e,long &ultra_time){
long time;
pinMode(pin_t,OUTPUT);
pinMode(pin_e,INPUT);
digitalWrite(pin_t,LOW);
delayMicroseconds(2);
digitalWrite(pin_t,HIGH);
delayMicroseconds(10);
time=pulseIn (pin_e,HIGH);
ultra_time =  time / 29 / 2;
}
void setup(){
Serial.begin(9600);// initialize serial communication at 9600 bits per second:
pinMode(relay, OUTPUT);
lcd.init();
lcd.backlight();
lcd.begin(20, 4);
lcd.setCursor(0, 0);
lcd.print("    6o Gymnasio   ");
lcd.setCursor(0, 1);
lcd.print(" Safe Entry Counter");
lcd.setCursor(0, 2);
lcd.print("     project by   ");
lcd.setCursor(0, 3);
lcd.print("Upcoming  Scientists");
delay(10000); // Waiting for a while
lcd.clear();
}
void loop(){
//************************
ultra_read(t_s1,e_s1,dis_a);delay(30);
ultra_read(t_s2,e_s2,dis_b);delay(30);
```

```
//************************
Serial.print("da:");Serial.println(dis_a);
Serial.print("db:");Serial.println(dis_b);
if(dis_a<90 && flag1==0){flag1=1;
if(flag2==0){person = person+1;}
}
if(dis_b<90 && flag2==0){flag2=1;
if(flag1==0){person = person-1;}
}
if(dis_a>90 && dis_b>90 && flag1==1 && flag2==1){
flag1=0, flag2=0;
delay(1000);
}
if(person<=0){
digitalWrite(relay,LOW);
lcd.setCursor(0, 0);
lcd.print("    6o Gymnasio     ");
lcd.setCursor(0, 1);
lcd.print("No Customers Inside");
lcd.setCursor(0,2);
lcd.print("      Welcome    ");
lcd.setCursor(0,3);
lcd.print("    Please Enter   ");
}
else if(person<10) {
digitalWrite(relay,LOW);
lcd.setCursor(0, 0);
lcd.print("    6o Gymnasio     ");
lcd.setCursor(0, 1);
lcd.print("Customers Inside: ");
lcd.print(person);
lcd.print("  ");
lcd.setCursor(0,2);
lcd.print("      Welcome    ");
lcd.setCursor(0,3);
lcd.print("    Please Enter   ");
}
else{digitalWrite(relay,HIGH);
lcd.setCursor(0, 0);
lcd.print("Customers Inside: ");
lcd.print(person);
lcd.print("  ");
lcd.setCursor(0,1);
lcd.print("       STOP        ");
lcd.setCursor(0,2);
lcd.print("    PLEASE WAIT    ");
lcd.setCursor(0,3);
lcd.print(" Too many customers");
```

}



Περιγραφή τεχνολογίας που χρησιμοποιήθηκε:

Για την συσκευή "μετρητή" χρησιμοποιήθηκαν: 1 x Arduino UNO, 2 x HC-SR04 distance sensors, 1x LCD display 20x4, 1x 5V relay module, 1 x Breadboard και καλώδια
Για τον αυτόματο μηχανισμό παροχής αντισηπτικού χρησιμοποιήθηκαν: 1 x medical tube, 1 x βάζο γυάλινο, 1 x 5V dc water pump, 1 x Transistor PNP 3A - TIP32, καλώδια και 1 Powerbank.

Επικοινωνία με Τοπικούς Φορείς:

Μέσω Email κάναμε αίτημα στην αλυσίδα Super Market "SYNKA" να επισκεφτούμε ένα κατάστημα και να μελετήσουμε τον τρόπο που γίνεται η καταμέτρηση των πελατών και με ποιον τρόπο τηρούνται όλα τα μέτρα ασφαλείας για την αποφυγή της διάδοσης του COVID-19. Στη συνέχεια και αφού ολοκληρώσαμε την κατασκευή μας πήγαμε σε ένα κατάστημα των Super Markets και δοκιμάσαμε τη συσκευή στον φυσικό χώρο. Οι υπεύθυνοι του Super Market ήταν πολύ ευγενικοί και συνεργάσιμοι βοηθώντας μας να πραγματοποιήσουμε τη δοκιμή.

Περιγραφή κοινωνικού αντικτύπου:

Με τη συσκευή αυτή πιστεύουμε ότι συμβάλλουμε στην καταπολέμηση της διασποράς του ιού.

6ο Τρίωρο

1. Arduino Αυτοματισμοί με App Inventor – IOT

Εφαρμογή 1: Αναβοσβήνουμε 2 LED στο Arduino από το κινητό μας τηλέφωνο

Πρόκειται για εφαρμογή «Διαδικτύου των Πάντων» (Internet of Things, IOT). Στην εφαρμογή αυτή συνδυάζουμε Arduino και App Inventor, μαθαίνοντας πώς να ελέγχουμε το Arduino από απόσταση μέσω του κινητού μας τηλεφώνου.

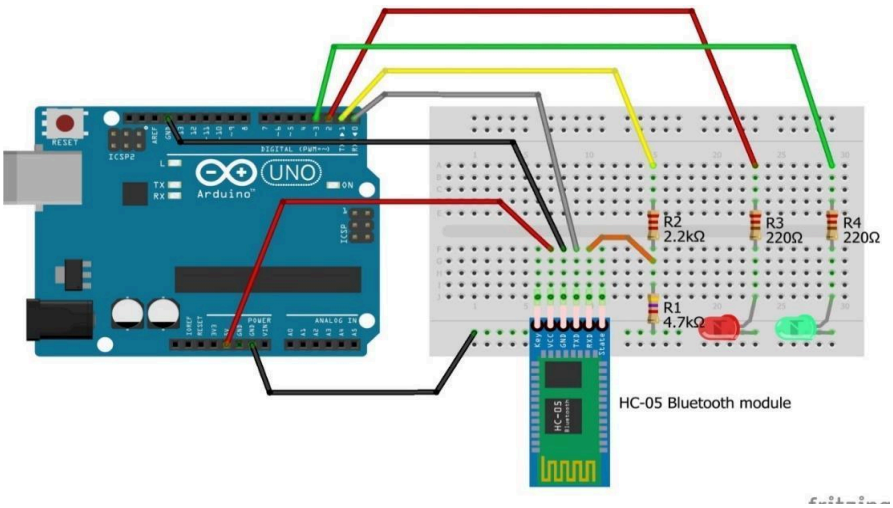
Στο πείραμα αυτό θα χρησιμοποιήσουμε την πλακέτα HC-05 Bluetooth για επικοινωνία του κινητού μας τηλεφώνου με τον Arduino. Συγκεκριμένα, θα κατασκευάσουμε ένα κύκλωμα με δύο LED στο Arduino το οποίο θα ελέγχουμε με το κινητό μας τηλέφωνο.

Το κύκλωμα με το Arduino χρειάζεται:

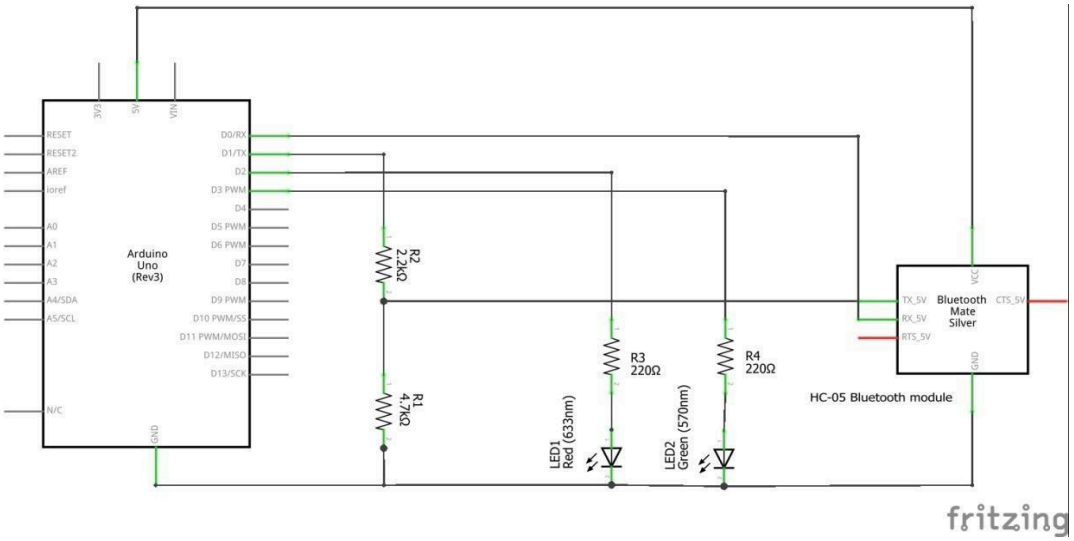
- Arduino Uno
- Πλακέτα Bluetooth HC-06
- Κόκκινο LED
- Πράσινο LED
- 2 αντιστάτες 220Ω
- 1 αντιστάτη 2,2kΩ
- 1 αντιστάτη 4,7kΩ
- Καλώδια jumper

Συνδεσμολογία

Πρώτα κατασκευάζουμε το κύκλωμα με το Arduino όπως δείχνει η παρακάτω εικόνα:



Το σχηματικό διάγραμμα του κυκλώματος είναι:



Ανάπτυξη προγράμματος σε Arduino IDE

Το πρόγραμμα σε κώδικα C του Arduino:

```
//Πρόγραμμα ελέγχου LED μέσω Bluetooth

char data = 0;              //Εδώ αποθηκεύεται το byte που λαμβάνουμε
void setup()
{
 Serial.begin(9600);       //Ρυθμός λειτουργίας της σειριακής επικοινωνίας
 pinMode(2, OUTPUT);       //Πιν του κόκκινου LED
 pinMode(3, OUTPUT);       //Πιν του πράσινου LED
}
void loop()
{
 if(Serial.available() > 0)  // Προχώρα παρακάτω μόνο αν έλαβες κάτι...
 {
  data = Serial.read();     //Διάβασε το byte που ήρθε και αποθήκευσέ το

  if(data == 11)           //Αν έλαβες τον αριθμό 11...
   digitalWrite(2, HIGH);  //Άναψε το πρώτο LED
  else if(data == 10)      //Αν έλαβες τον αριθμό 10...
   digitalWrite(2, LOW);   //Σβήσε το πρώτο LED
  else if(data == 21)      //Αν έλαβες τον αριθμό 21...
   digitalWrite(3, HIGH);  //Άναψε το δεύτερο LED
  else if(data == 20)      //Αν έλαβες τον αριθμό 20...
```

```
    digitalWrite(3, LOW);   //Σβήσε το δεύτερο LED
  }
}
```

ΠΡΟΣΟΧΗ ΕΔΩ ΠΩΣ ΘΑ ΑΝΕΒΑΣΟΥΜΕ ΤΟΝ ΚΩΔΙΚΑ: Πριν ανεβάσουμε τον κώδικα στο Arduino, αποσυνδέουμε τα καλώδια που πηγαίνουν στα ψηφιακά πιν 0 και 1 του Arduino (D0, D1). Αυτά στην εικόνα είναι τα καλώδια με το ΓΚΡΙ και το ΚΙΤΡΙΝΟ χρώμα. Ο λόγος είναι ότι αυτά τα πιν χρησιμοποιούνται για την επικοινωνία υπολογιστή – Arduino και το ανέβασμα του κώδικα στο Arduino, αλλά επίσης συνδέονται στην πλακέτα Bluetooth. Αφού ανεβάσουμε τον κώδικα στον Arduino, ξανασυνδέουμε τα καλώδια στα δύο αυτά πιν για να μπορεί να επικοινωνεί το Arduino με την πλακέτα Bluetooth. Αυτό μπορεί να γίνει και χωρίς να διακόψουμε την τροφοδοσία στο Arduino, αλλά με προσοχή για να συνδέσουμε στα σωστά πιν. Από εδώ και πέρα δεν θα χρειαστεί να αποσυνδέσουμε αυτά τα δύο καλώδια, εκτός από την περίπτωση που θα θελήσουμε και πάλι να ανεβάσουμε κώδικα στο Arduino, οπότε κάνουμε ξανά προσωρινή αποσύνδεση των καλωδίων.

ΚΙΝΗΤΟ ΤΗΛΕΦΩΝΟ και ζεύξη με την πλακέτα Bluetooth HC-06

Τώρα ήρθε η ώρα να κάνουμε κάποιες ρυθμίσεις στο κινητό μας τηλέφωνο:

1. Πάμε στις Ρυθμίσεις του κινητού μας (εικονίδιο γρανάζι).

2. Ενεργοποιούμε το *WiFi* και το *Bluetooth*.

Τώρα πρέπει να κάνουμε ζεύξη του κινητού μας με την πλακέτα Bluetooth HC-05.

3. Για να γίνει αυτό, θα πάμε στις Ρυθμίσεις του κινητού, θα πάμε στο *Bluetooth* και στις Διαθέσιμες συσκευές θα επιλέξουμε το HC-05 (μην ξεχνάτε επίσης ότι πρέπει να έχουμε συνδεμένο το ρεύμα στο Arduino μέσω του καλωδίου USB).

4. Επιλέγουμε Ζεύξη με το HC-06.

5. Τελειώσαμε. Από εδώ και πέρα μπορούμε να χρησιμοποιήσουμε με το κινητό μας, την πλακέτα αυτή. Φυσικά, για να γίνει αυτό, θα φτιάξουμε την κατάλληλη εφαρμογή στο App Inventor.

ΔΗΜΙΟΥΡΓΙΑ ΕΦΑΡΜΟΓΗΣ App Inventor στον Υπολογιστή

Στο App Inventor τώρα, δημιουργούμε μια εφαρμογή για να ελέγχουμε μέσω Bluetooth τα δύο LED που έχουμε συνδέσει στο Arduino.

1. Στον υπολογιστή μας ανοίγουμε ένα φυλλομετρητή ιστού (*Browser*) και πληκτρολογούμε στη γραμμή διευθύνσεων, επάνω, το: http://appinventor.mit.edu

2. Στη σελίδα που ανοίγει πατάμε το κουμπί επάνω δεξιά: *Create Apps*!

3. Βάζουμε τα στοιχεία του λογαριασμού μας *Google* για να συνδεθούμε ως χρήστης της *Google*.

4. Στην αρχική οθόνη του App Inventor επιλέγουμε στο μενού *Projects* *Start New Project*. Προς το παρόν έχουμε ως μόνο αντικείμενο την οθόνη του κινητού με όνομα *Screen1*.

5. Στο παράθυρο *Components* επιλέγουμε (αριστερό κλικ) την *Screen1* και πάμε δεξιά στις ιδιότητες *Properties*. Πάμε κάτω και κάνουμε την ιδιότητα *Title* να έχει κείμενο στο παράθυρο κειμένου «*Bluetooth Controler*».

6. Στη συνέχεια θα τοποθετήσουμε κάποια αντικείμενα στην οθόνη του κινητού.

Από την παλέτα User Interface σύρουμε και τοποθετούμε στην οθόνη τα εξής αντικείμενα και τα τοποθετούμε με την σειρά που ακολουθεί από πάνω προς τα κάτω:

- Label (Label1)
- ListPicker (ListPicker1)
- Label (Label2)
- Label (Label3)
- HorizontalArrangement (HorizontalArrangement1)

7. Τώρα, ΜΕΣΑ στο *HorizontalArrangement1* σύρουμε από την παλέτα User Interface και τοποθετούμε από αριστερά προς τα δεξιά τα εξής:

- Button (Button1), Label (Label4), Button (Button2)

8. Κάτω από το *HorizontalArrangement1* σύρουμε από την παλέτα *User Interface* και τοποθετούμε τα εξής, από πάνω προς τα κάτω:

- Label (Label5)
- Label (Label6)
- HorizontalArrangement (HorizontalArrangement2)

9. Τώρα, ΜΕΣΑ στο *HorizontalArrangement2* σύρουμε από την παλέτα *User Interface* και τοποθετούμε από αριστερά προς τα δεξιά τα εξής:

- Button (Button3), Label (Label7), Button (Button4)

10. Από την παλέτα *Connectivity* σύρουμε και ρίχνουμε στην οθόνη ένα *BluetoothClient*, το οποίο δεν εμφανίζεται μέσα στην οθόνη του κινητού αλλά πάει από κάτω και φαίνεται εκεί.

Τώρα πρέπει να αλλάξουμε μερικές **ιδιότητες** στα διάφορα αντικείμενα για να δώσουμε στην εικόνα αυτή της εφαρμογής την μορφή που θέλουμε:

1. Στο παράθυρο *Components* επιλέγουμε (κλικ) το **Label1** και πάμε δίπλα στο παράθυρο *Properties* του.

Εκεί κάνουμε Height 10pixels, Width Fill Parent, Text ΤΙΠΟΤΕ! (Διαγράφουμε ότι υπάρχει εκεί).

2. Στο παράθυρο *Components* επιλέγουμε (κλικ) το **ListPicker1** και πάμε δίπλα στο παράθυρο *Properties* του.

Εκεί κάνουμε BackgroundColor Blue, FontBold Τσεκαρισμένο, FontSize 14, Height Automatic, Width Fill Parent, Shape Rounded, Text PRESS to Connect with Bluetooth Device, TextAlignment Center, TextColor White, Title ΤΙΠΟΤΕ (Σβήνουμε ότι υπάρχει) .

3. Στο παράθυρο *Components* επιλέγουμε (κλικ) το **Label2** και πάμε δίπλα στο παράθυρο *Properties* του.

Εκεί κάνουμε Height 40pixels, Width Fill Parent, Text ΤΙΠΟΤΕ (Σβήνουμε ότι υπάρχει).

4. Στο παράθυρο *Components* επιλέγουμε (κλικ) το **Label3** και πάμε δίπλα στο παράθυρο *Properties* του.

Εκεί κάνουμε FontBold Τσεκαρισμένο, FontSize 16, Height Automatic, Width Fill Parent, Text Led 1 Control, TextAlignment Center.

5. Στο παράθυρο *Components* επιλέγουμε (κλικ) το **HorizontalArrangement1** και πάμε δίπλα στο παράθυρο *Properties* του.

Εκεί κάνουμε AlighHorizontal Center, Height Automatic, Width Fill Parent.

6. Στο παράθυρο *Components* επιλέγουμε (κλικ) το **Button1** και πάμε δίπλα στο παράθυρο Properties του.

Εκεί κάνουμε BackgroundColor Red, Height 50pixels, Width 100pixels, Shape Rounded, Text ON Led 1, TextAlignment Center.

7. Στο παράθυρο *Components* επιλέγουμε (κλικ) το **Label4** και πάμε δίπλα στο παράθυρο *Properties* του.

Εκεί κάνουμε Height FillParent, Width 20pixels, Text ΤΙΠΟΤΕ (Σβήνουμε ότι υπάρχει).

8. Στο παράθυρο *Components* επιλέγουμε (κλικ) το **Button2** και πάμε δίπλα στο παράθυρο Properties του.

Εκεί κάνουμε BackgroundColor Custom Στο πλαίσιο κειμένου επάνω γράφουμε: #850000ff και πατάμε Done, Height 50pixels, Width 100pixels, Shape Rounded, Text OFF Led 1, TextAlignment Center.

9. Στο παράθυρο *Components* επιλέγουμε (κλικ) το **Label5** και πάμε δίπλα στο παράθυρο *Properties* του.

Εκεί κάνουμε Height 20pixels, Width Fill Parent, Text ΤΙΠΟΤΕ (Σβήνουμε ότι υπάρχει).

10. Στο παράθυρο *Components* επιλέγουμε (κλικ) το **Label6** και πάμε δίπλα στο παράθυρο Properties του.

Εκεί κάνουμε FontBold Τσεκαρισμένο, FontSize 16, Height Automatic, Width Fill Parent, Text Led 2 Control, TextAlignment Center.

11. Στο παράθυρο *Components* επιλέγουμε (κλικ) το **HorizontalArrangement2** και πάμε δίπλα στο παράθυρο *Properties* του.

Εκεί κάνουμε AlighHorizontal Center, Height Automatic, Width Fill Parent.

12. Στο παράθυρο *Components* επιλέγουμε (κλικ) το **Button3** και πάμε δίπλα στο παράθυρο *Properties* του.

Εκεί κάνουμε BackgroundColor Green, Height 50pixels, Width 100pixels, Shape Rounded, Text ON Led 2, TextAlignment Center.

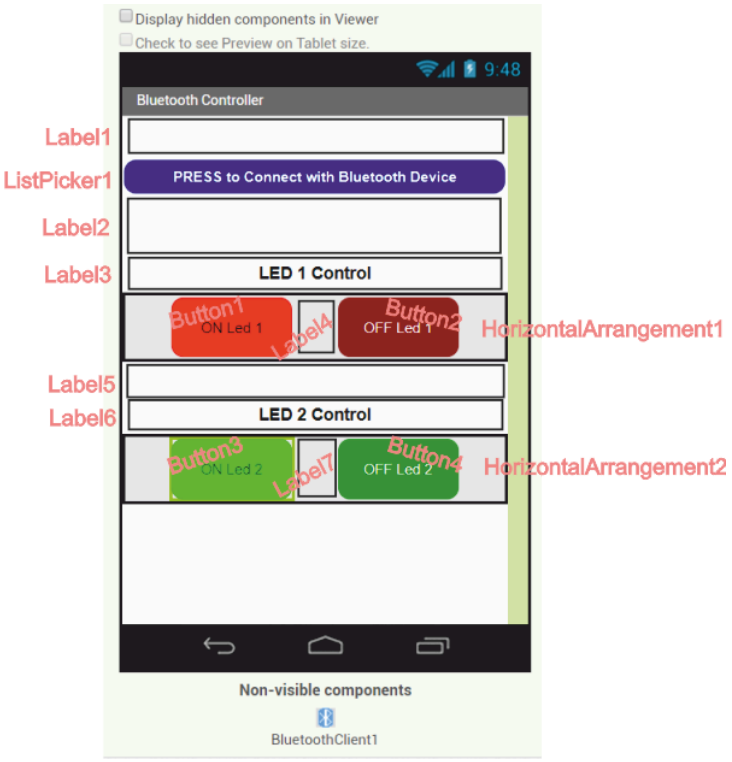
13. Στο παράθυρο *Components* επιλέγουμε (κλικ) το **Label7** και πάμε δίπλα στο παράθυρο *Properties* του.

Εκεί κάνουμε Height FillParent, Width 20pixels, Text ΤΙΠΟΤΕ (Σβήνουμε ότι υπάρχει).

14. Στο παράθυρο *Components* επιλέγουμε (κλικ) το **Button4** και πάμε δίπλα στο παράθυρο *Properties* του.

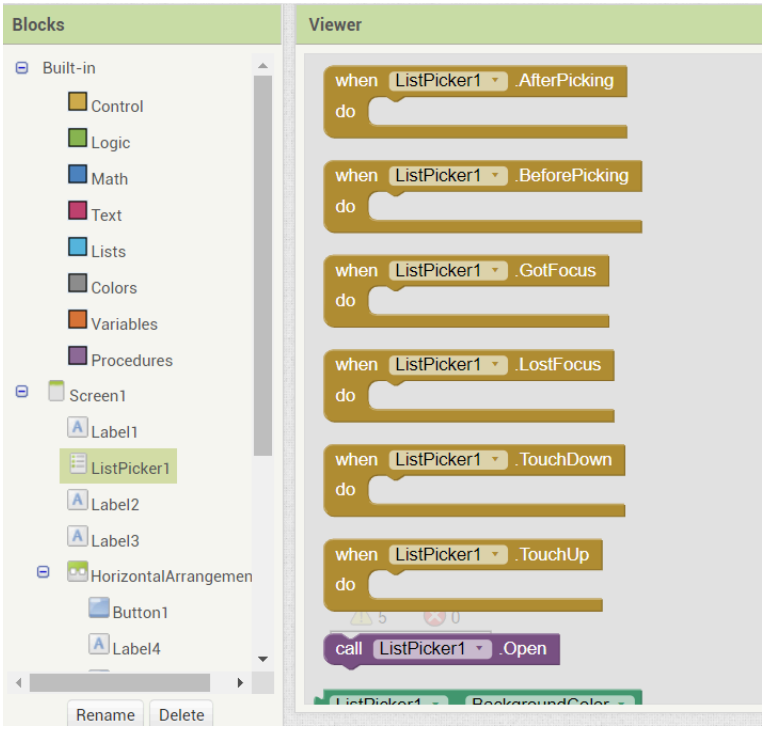
Εκεί κάνουμε BackgroundColor Custom στο πλαίσιο κειμένου επάνω βάζουμε: #008500ff και πατάμε Done, Height 50pixels, Width 100pixels, Shape Rounded, Text OFF Led2, TextAlignment Center.

Όταν τελειώσουμε αυτές τις ρυθμίσεις, η οθόνη του κινητού πρέπει να μοιάζει έτσι:

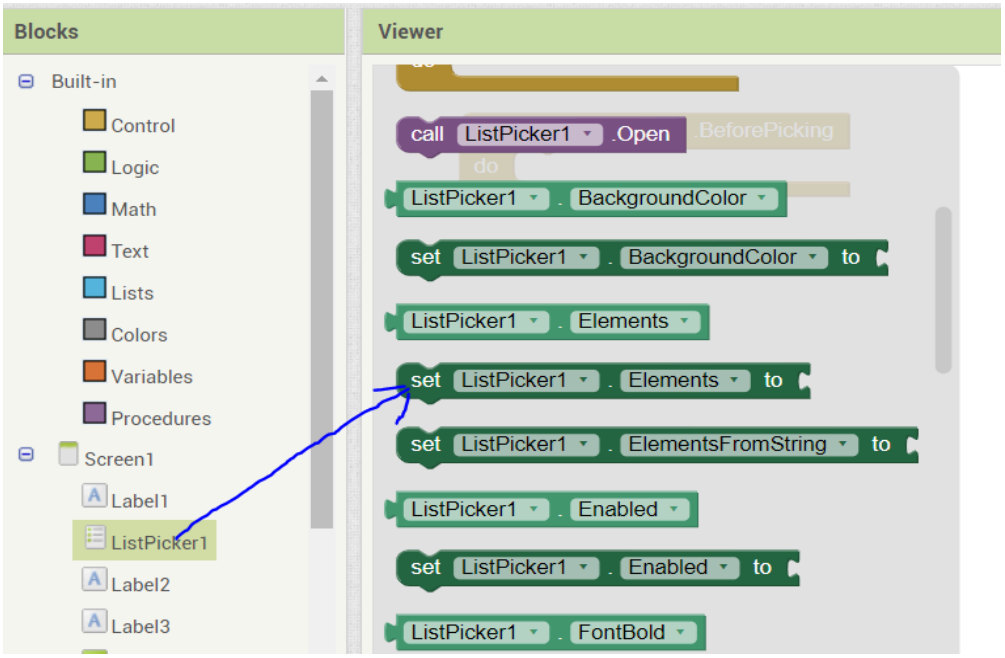


Στη συνέχεια, πάμε στην οθόνη του *Blocks AppInventor* για να γράψουμε τον κώδικα της εφαρμογής. Αυτός είναι ο εξής:

1. Από το παράθυρο *Blocks* αριστερά, κάνουμε αριστερό κλικ στο *ListPicker1* και στο διαφανές μενού με τα πλακίδια που ανοίγει, σύρουμε δεξιά στο παράθυρο κώδικα το πλακίδιο: *when ListPicker1.BeforePicking*.

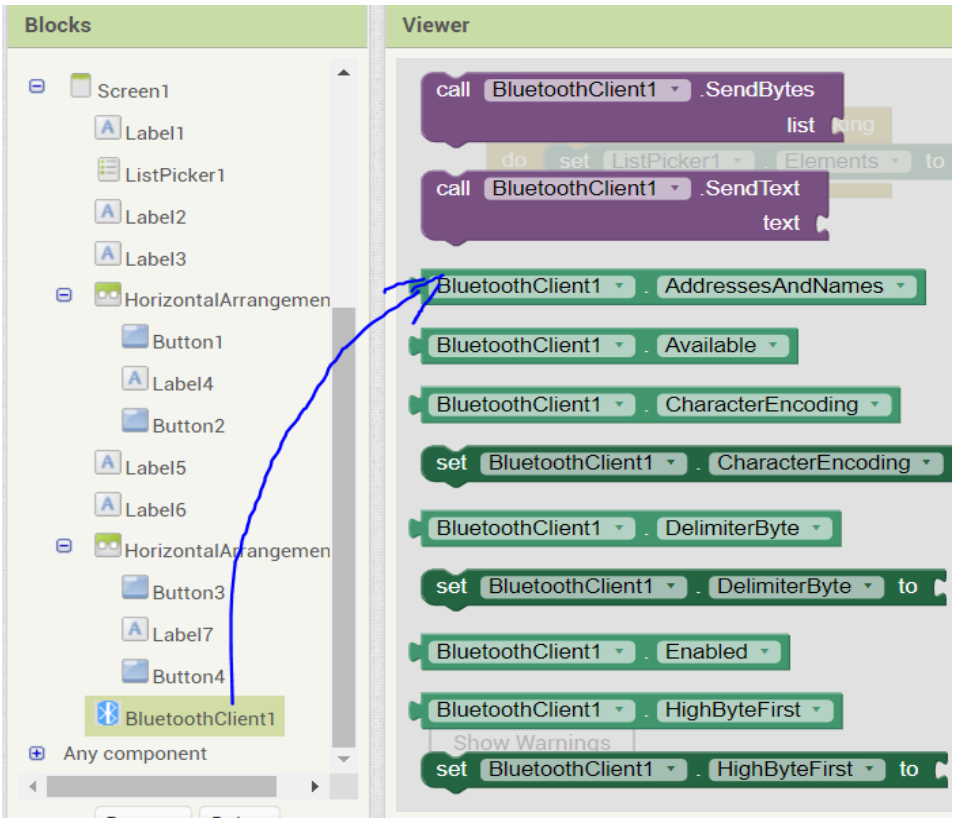


2. Από το παράθυρο *Blocks* αριστερά, κάνουμε αριστερό κλικ στο *ListPicker1* και

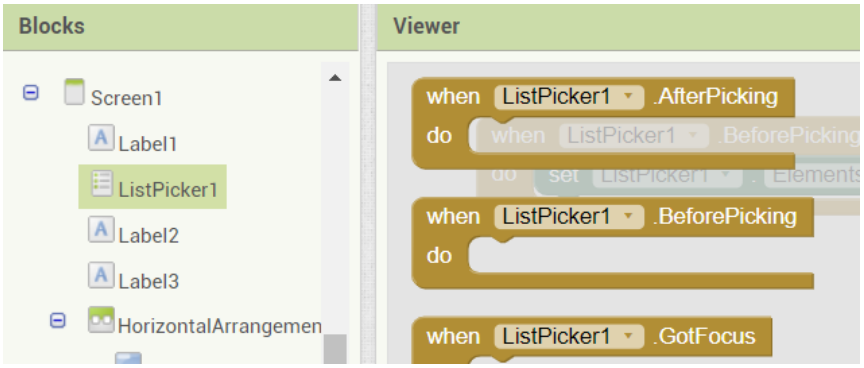


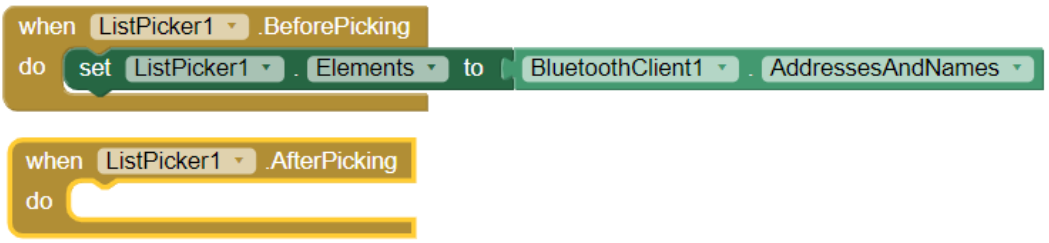
στο διαφανές μενού με τα πλακίδια που ανοίγει, σύρουμε δεξιά στο παράθυρο κώδικα το πλακίδιο: σετ *ListPicker1.Elements to*.

3. Από το παράθυρο *Blocks* αριστερά, κάνουμε αριστερό κλικ στο *BluetoothClient1* και στο διαφανές μενού με τα πλακίδια που ανοίγει, σύρουμε δεξιά στο παράθυρο κώδικα το πλακίδιο: *BluetoothClient1 .ddressesAndNames* και το βάζουμε στην υποδοχή του προηγούμενου πλακιδίου.

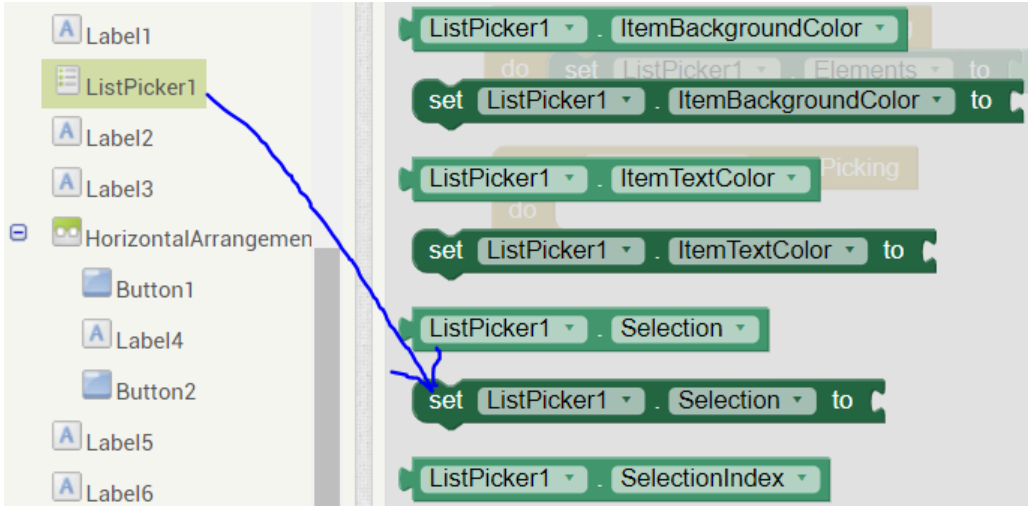


4. Από το παράθυρο *Blocks* αριστερά, κάνουμε αριστερό κλικ στο *ListPicker1* και στο διαφανές μενού με τα πλακίδια που ανοίγει, σύρουμε δεξιά στο παράθυρο κώδικα το μπεζ πλακίδιο: *when ListPicker1.AfterPicking*.



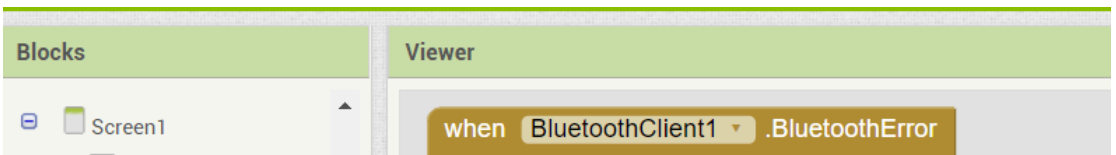


5.

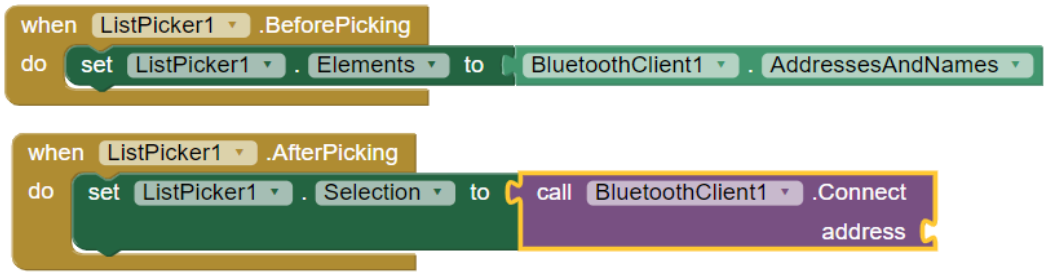


Από το παράθυρο *Blocks* αριστερά, κάνουμε αριστερό κλικ στο *ListPicker1* και στο διαφανές μενού με τα πλακίδια που ανοίγει, σύρουμε δεξιά στο παράθυρο κώδικα το πλακίδιο: *when ListPicker1.Selection to* και το βάζουμε στην υποδοχή του μπεζ παραθύρου *ListPicker1.AfterPicking*.

6. Από το παράθυρο *Blocks* αριστερά, κάνουμε αριστερό κλικ στο *BluetoothClient1* και στο διαφανές μενού με τα πλακίδια που ανοίγει, σύρουμε δεξιά στο παράθυρο κώδικα το μωβ πλακίδιο: *call BluetoothClient1.Connect*

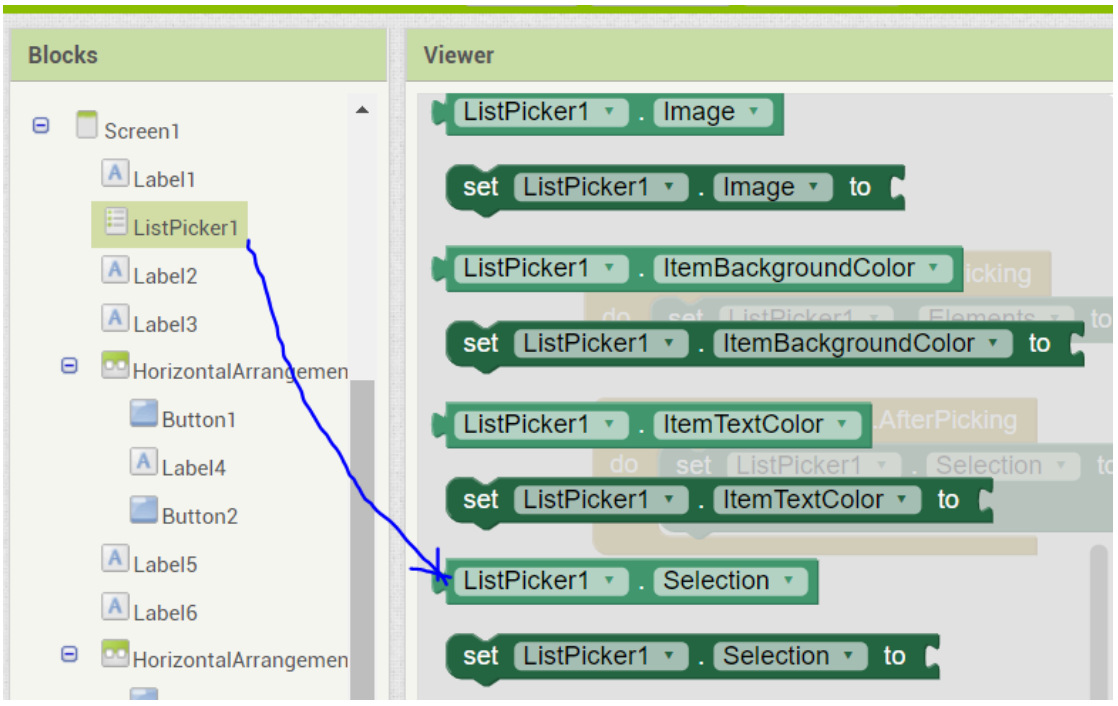


και το βάζουμε στην υποδοχή «*to*» του προηγούμενου πράσινου παραθύρου

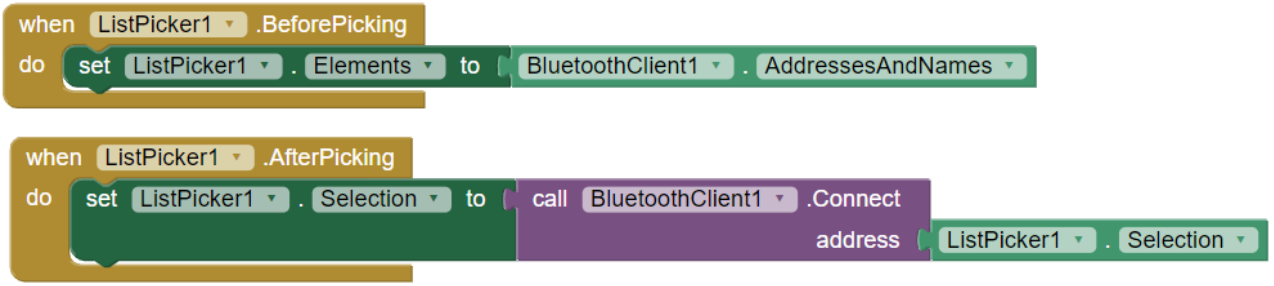


ListPicker1.Selection to.

7. Από το παράθυρο *Blocks* αριστερά, κάνουμε αριστερό κλικ στο *ListPicker1* και στο διαφανές μενού με τα πλακίδια που ανοίγει, σύρουμε δεξιά στο παράθυρο κώδικα το πλακίδιο: *ListPicker1.Selection*

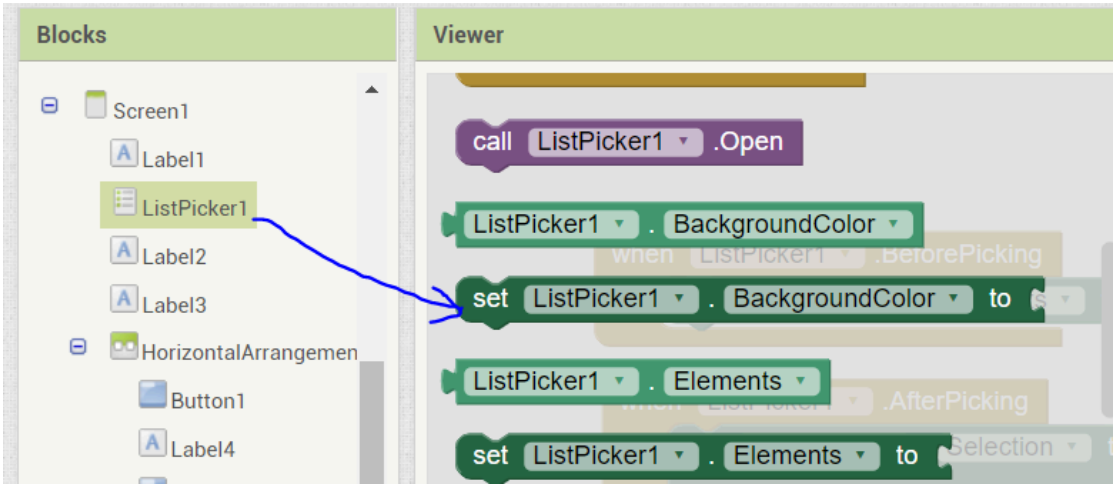


και το βάζουμε στην υποδοχή του προηγούμενου μωβ παραθύρου *call*

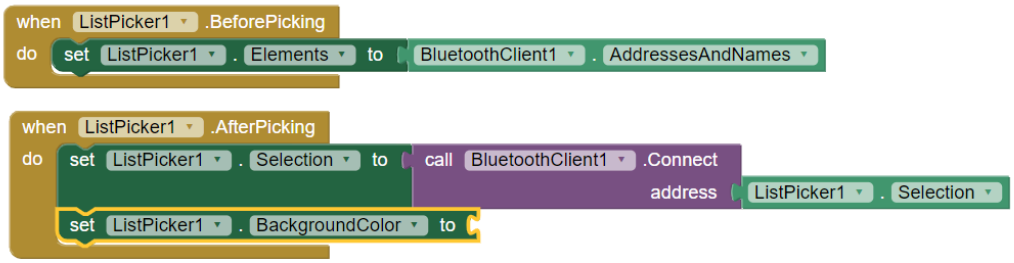


BluetoothClient1.Connect.

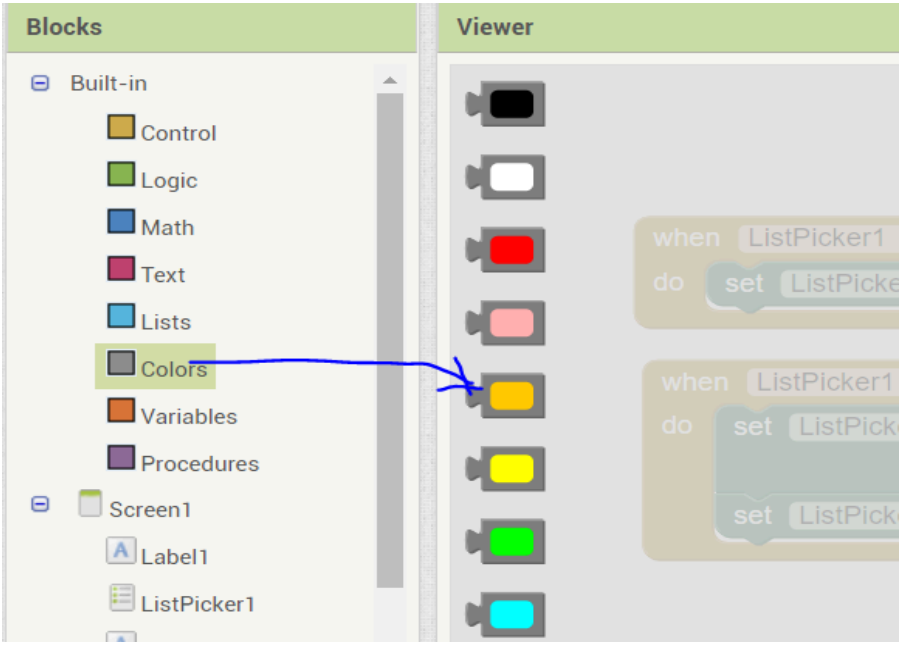
8. Από το παράθυρο *Blocks* αριστερά, κάνουμε αριστερό κλικ στο *ListPicker1* και στο διαφανές μενού με τα πλακίδια που ανοίγει, σύρουμε δεξιά στο παράθυρο κώδικα το πλακίδιο: set*ListPicker1.BackgroundColor to*



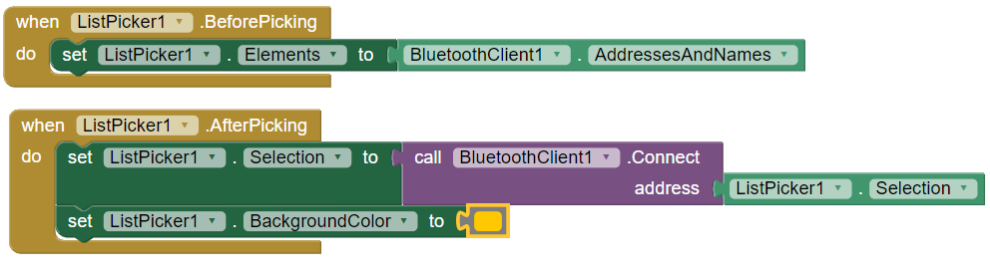
και το βάζουμε κάτω από το *set Picker1.Selection to*



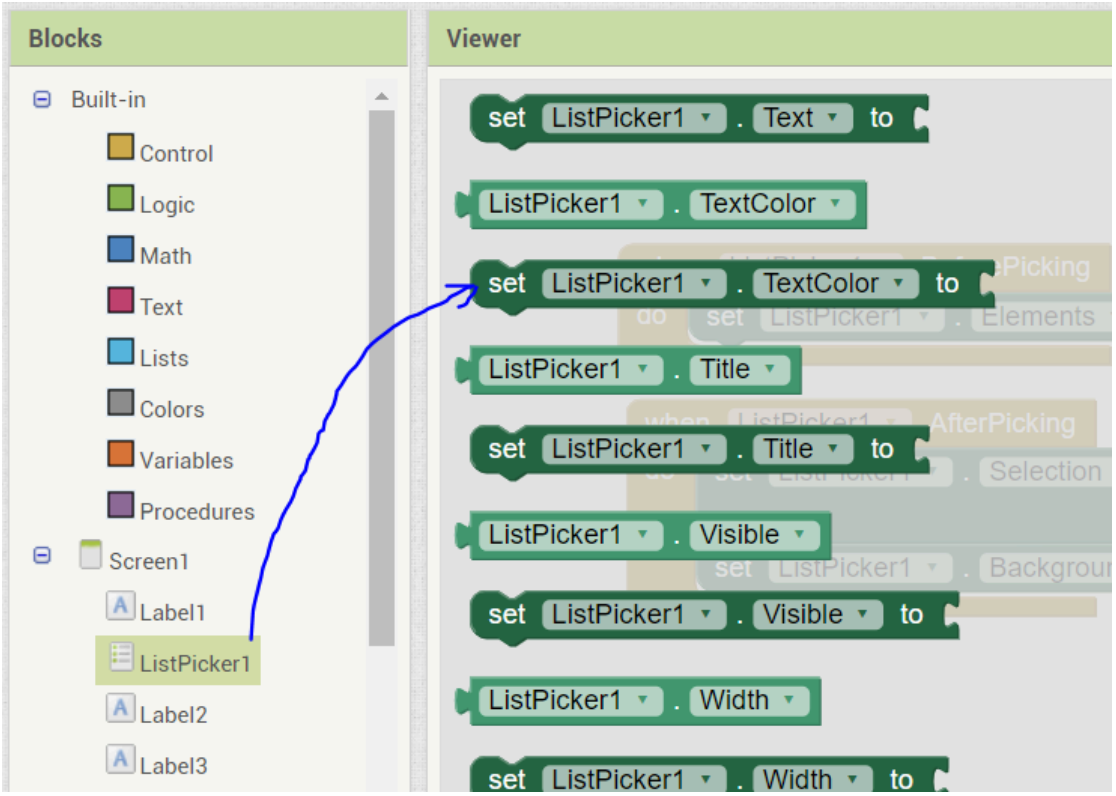
9. Από το παράθυρο *Blocks* αριστερά, κάνουμε αριστερό κλικ στο *Colors* και στο διαφανές μενού με τα πλακίδια που ανοίγει, σύρουμε δεξιά στο παράθυρο κώδικα το πλακίδιο με το πορτοκαλί χρώμα



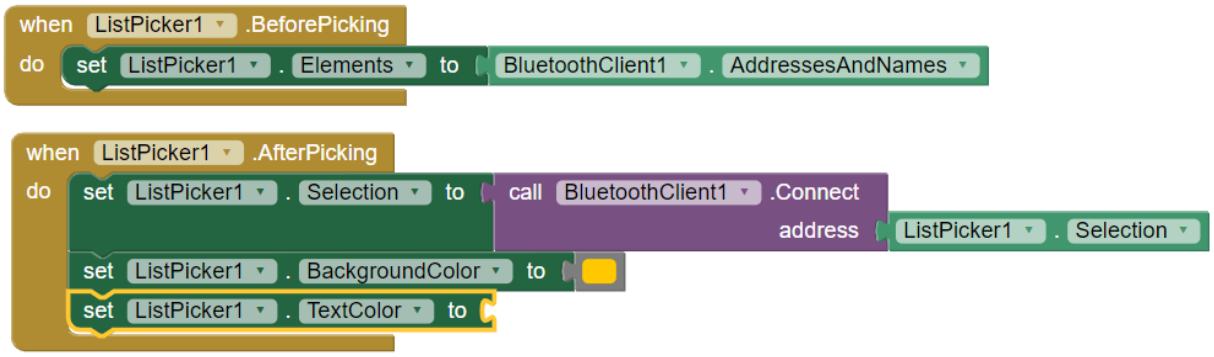
και το τοποθετούμε στην υποδοχή του προηγούμενου πλακιδίου:



10. Από το παράθυρο *Blocks* αριστερά, κάνουμε αριστερό κλικ στο *ListPicker1* και στο διαφανές μενού με τα πλακίδια που ανοίγει, σύρουμε δεξιά στο παράθυρο κώδικα το πλακίδιο: set*ListPicker1.TextColor to*

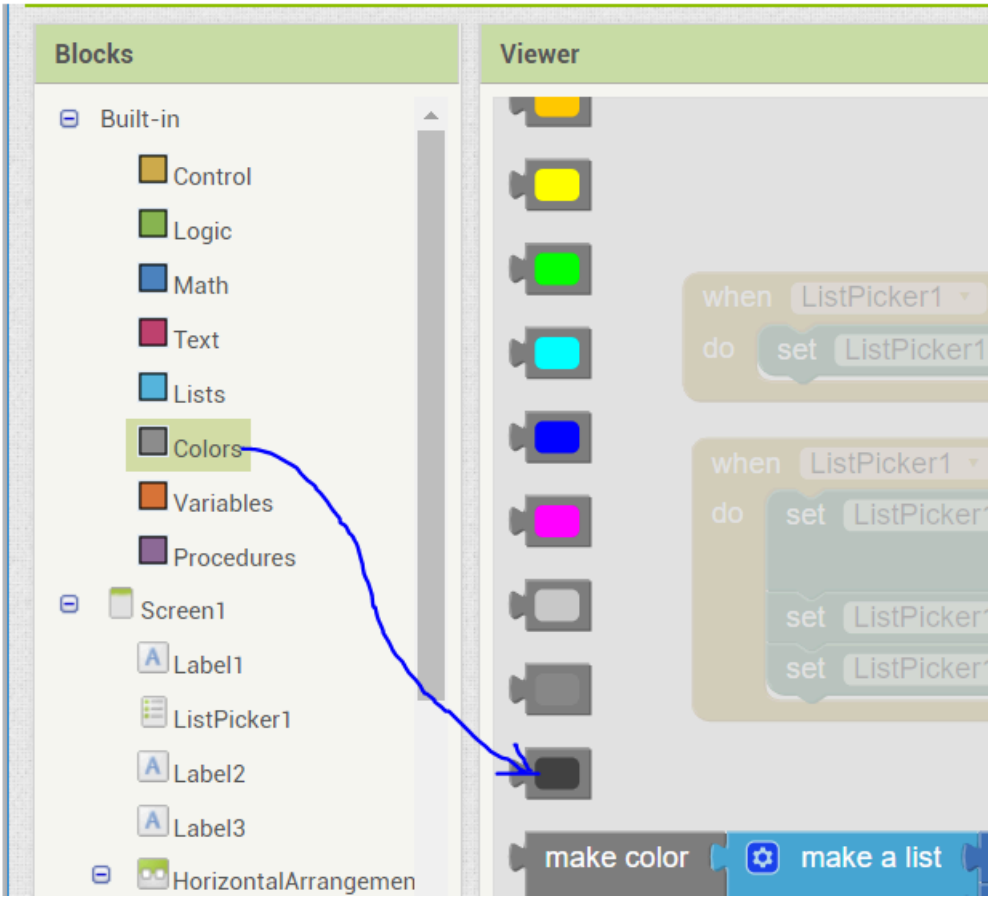


και το τοποθετούμε κάτω από το σκούρο πράσινο πλακίδιο που λέει: set*ListPicker1.BackgroundColor.*

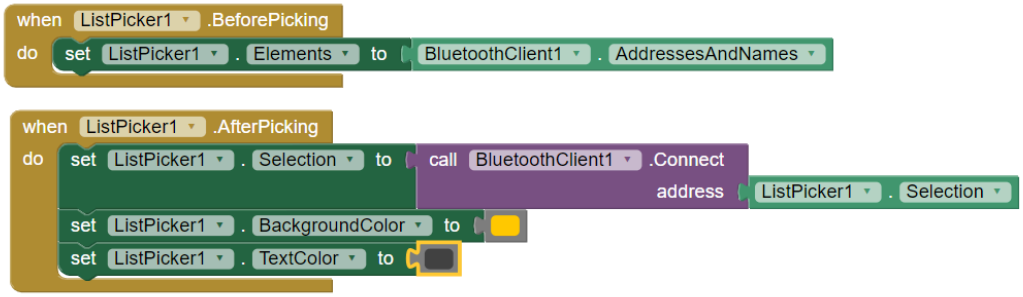


11. Από το παράθυρο *Blocks* αριστερά, κάνουμε αριστερό κλικ στο *Colors* και στο διαφανές μενού με τα πλακίδια που ανοίγει, σύρουμε δεξιά στο παράθυρο κώδικα το πλακίδιο με το μαύρο χρώμα

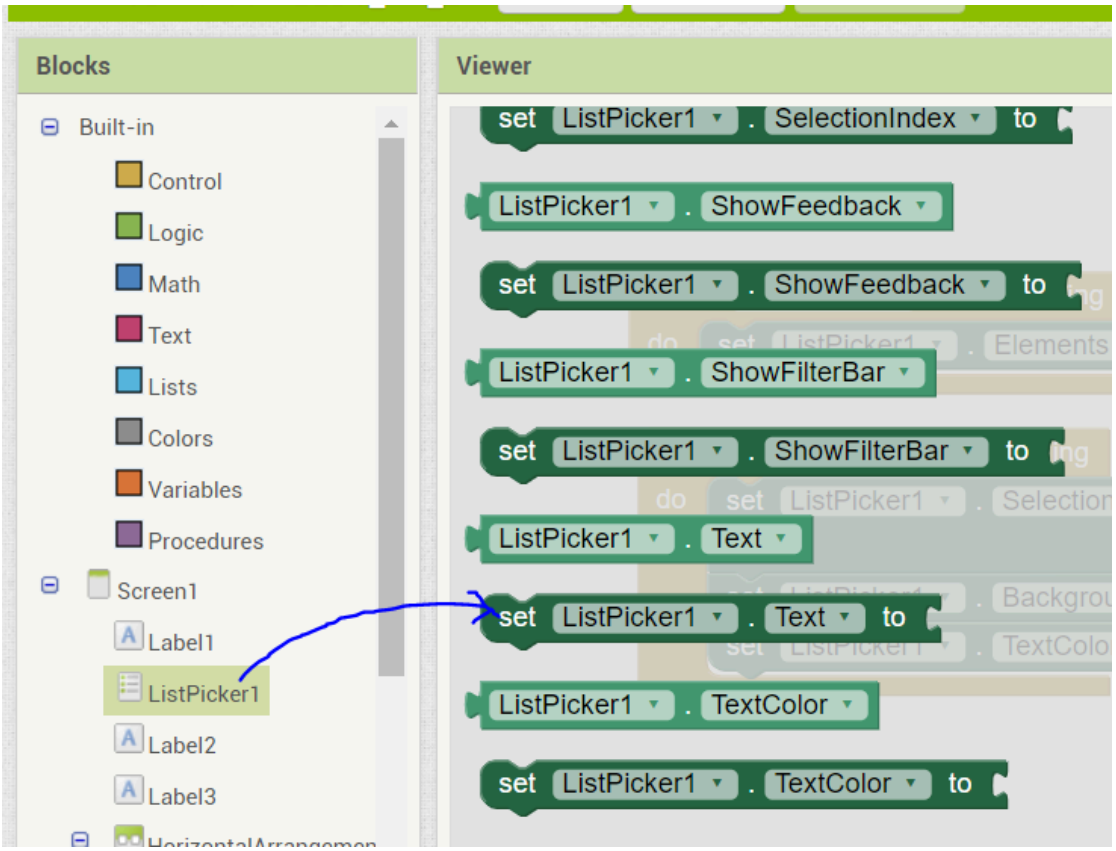
και το τοποθετούμε στην υποδοχή του προηγούμενου πλακιδίου: set



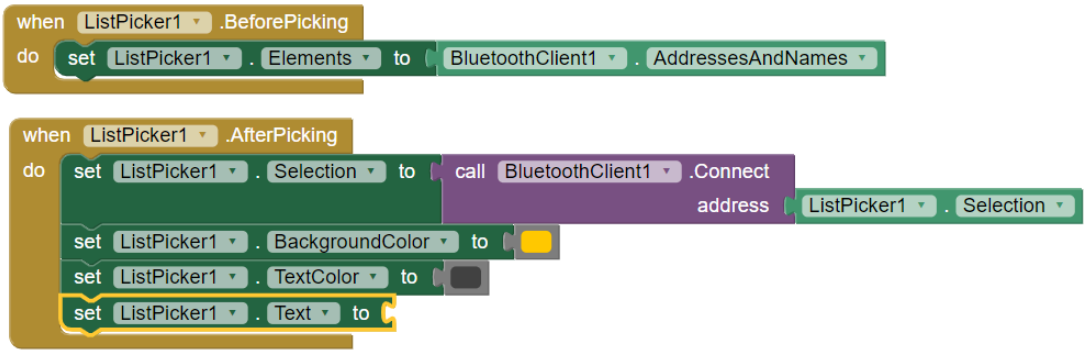
ListPicker1.TextColor to.



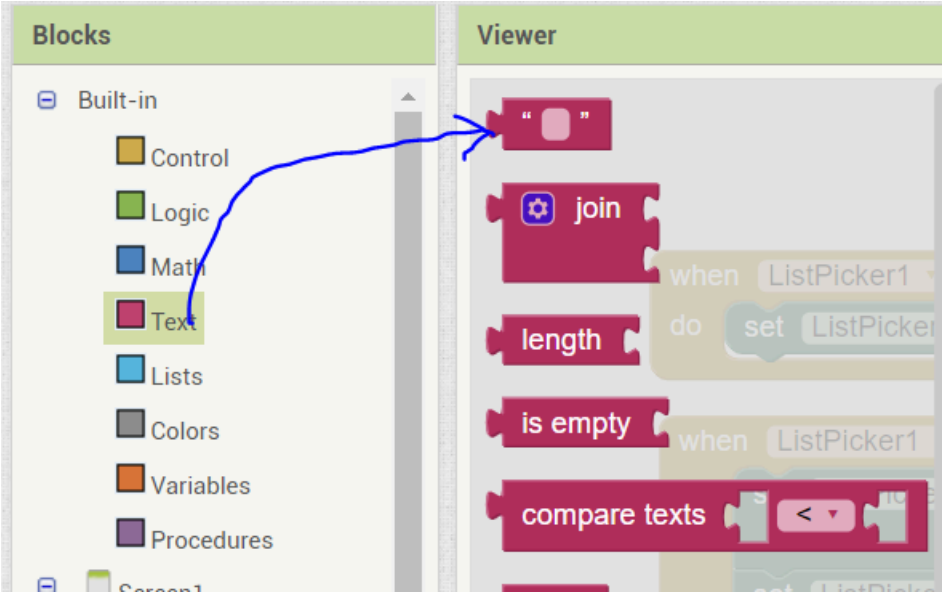
12. Από το παράθυρο *Blocks* αριστερά, κάνουμε αριστερό κλικ στο *ListPicker1* και στο διαφανές μενού με τα πλακίδια που ανοίγει, σύρουμε δεξιά στο παράθυρο κώδικα το πλακίδιο: set*ListPicker1.Text to*

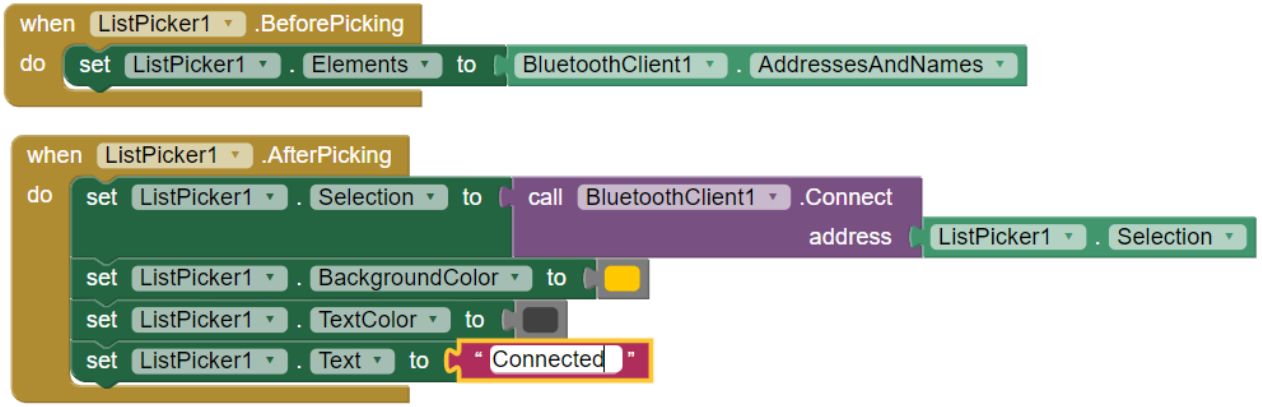


και το τοποθετούμε κάτω από το πλακίδιο: set ListPicker1.TextCo



13. Από το παράθυρο *Blocks* αριστερά, κάνουμε αριστερό κλικ στο *Text* και στο διαφανές μενού με τα πλακίδια που ανοίγει, σύρουμε δεξιά στο παράθυρο κώδικα το πρώτο του πλακίδιο

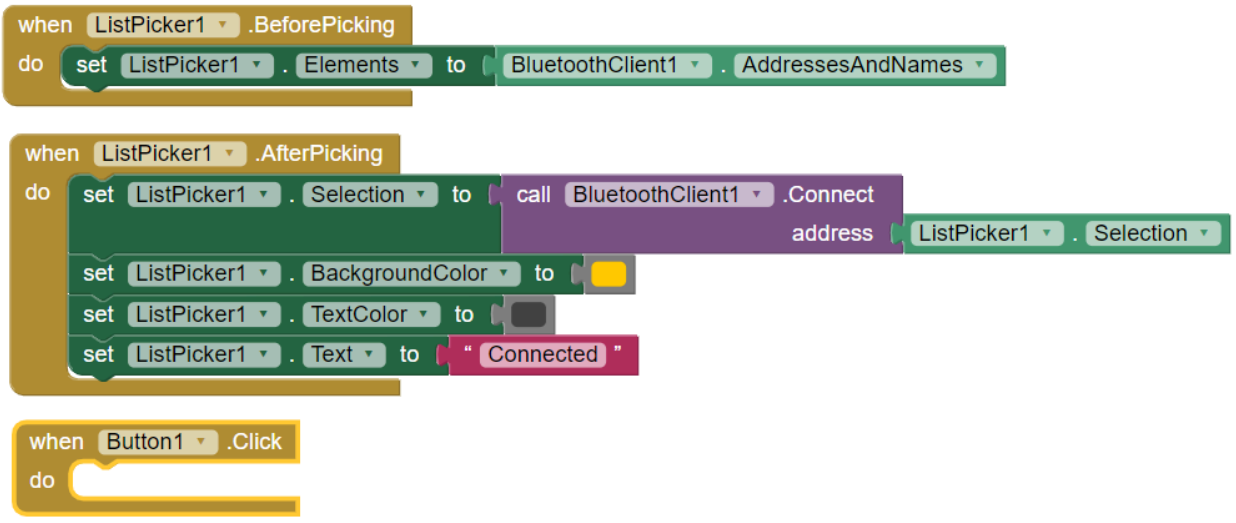




και το τοποθετούμε στην υποδοχή του προηγούμενου:

14. Τώρα κάνουμε κλικ μέσα στο ροζ τετράγωνο του κόκκινου πλακιδίου και γράφουμε εκεί μέσα «Connected».

15. Από το παράθυρο *Blocks* αριστερά, κάνουμε αριστερό κλικ στο *Button1* και στο

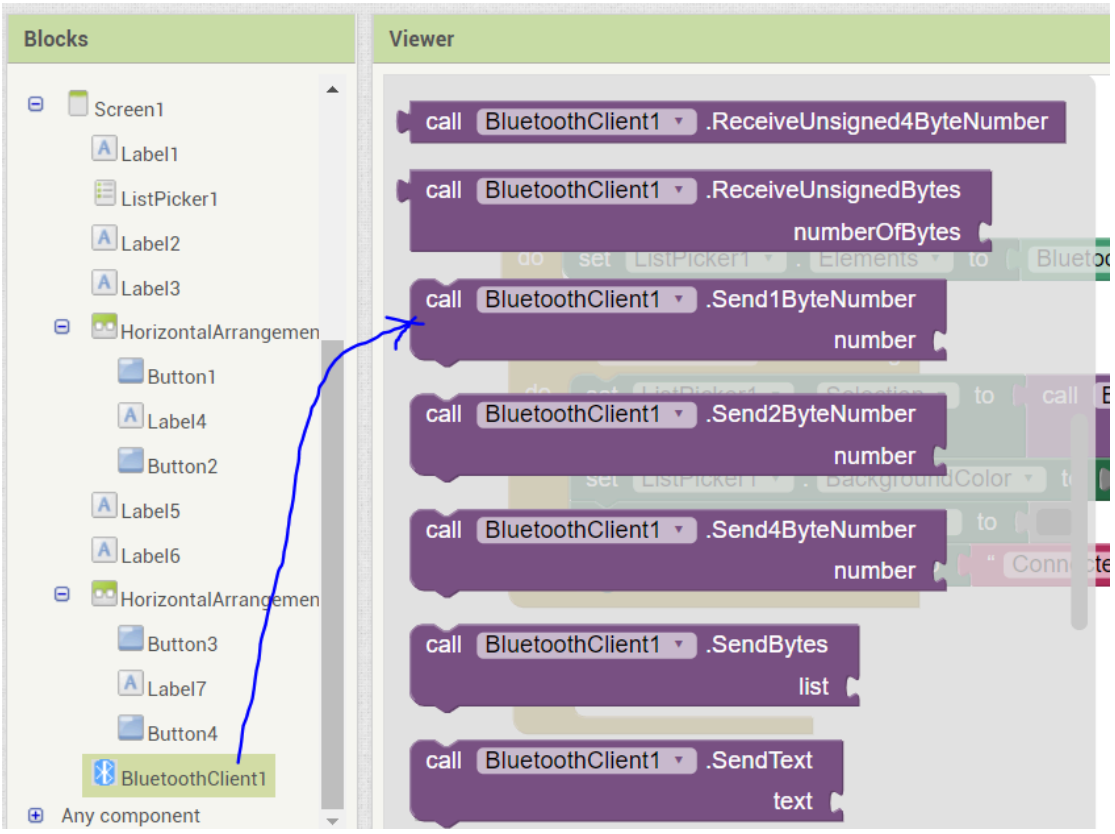


2 ⊗ 0

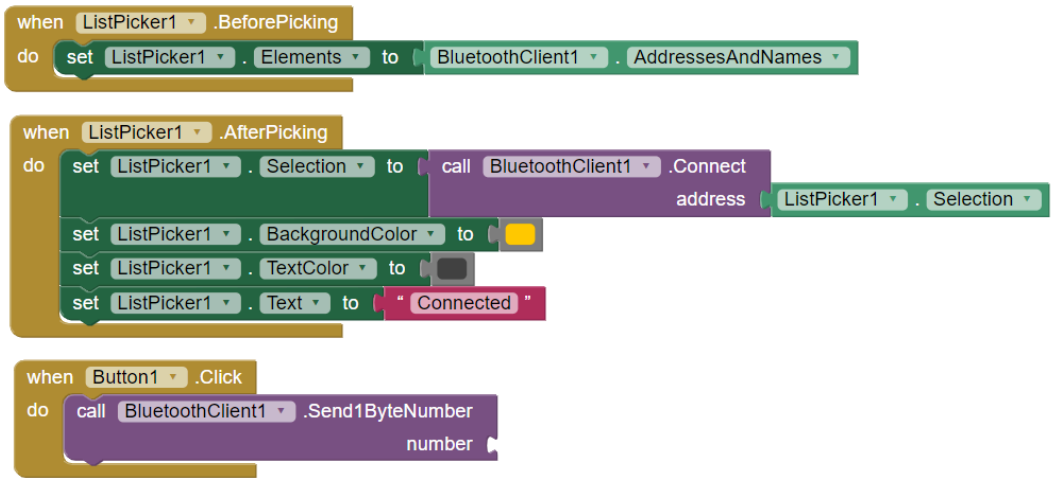
διαφανές μενού με τα πλακίδια που ανοίγει, σύρουμε δεξιά στο παράθυρο κώδικα το

πλακίδιο “When Button1.Click και το τοποθετούμε οπουδήποτε στον άδειο λευκό χώρο της σχεδιαστικής επιφάνειας:

16. Από το παράθυρο *Blocks* αριστερά, κάνουμε αριστερό κλικ στο BluetoothClient*1* και στο διαφανές μενού με τα πλακίδια που ανοίγει, σύρουμε δεξιά στο παράθυρο κώδικα το μωβ πλακίδιο: call BluetoothClient1. Send1ByteNumber number:

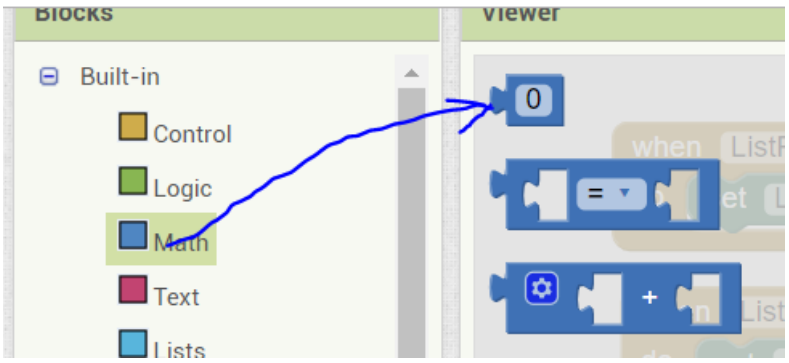


και το τοποθετούμε μέσα στο μπεζ πλακίδιο που μόλις αφήσαμε στην επιφάνεια σχεδίασης:

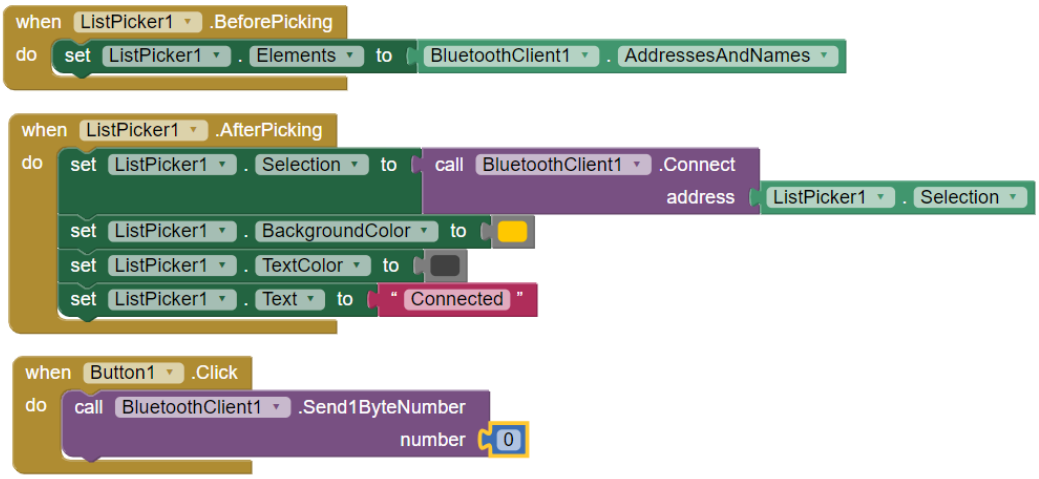


17. Από το παράθυρο *Blocks* αριστερά, κάνουμε αριστερό κλικ στο *Math* και στο διαφανές μενού με τα πλακίδια που ανοίγει, σύρουμε δεξιά στο παράθυρο κώδικα το

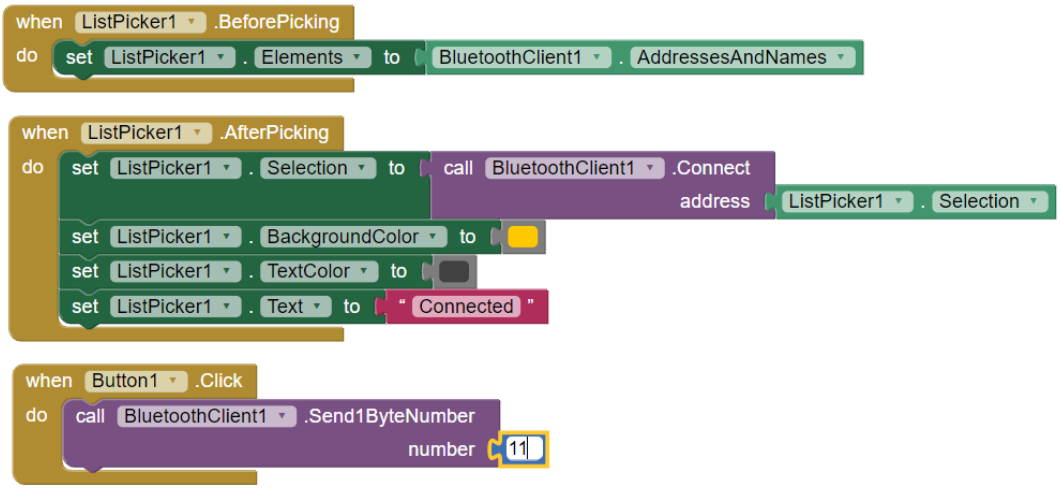
πρώτο πλακίδιο



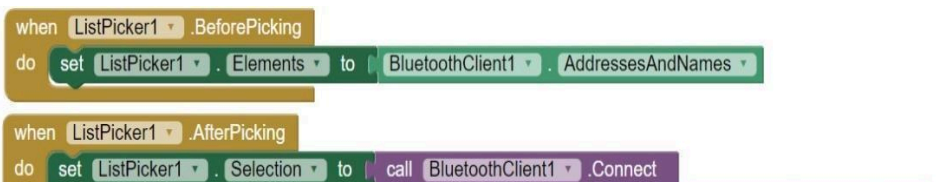
και το τοποθετούμε μέσα στην υποδοχή του προηγούμενου πλακιδίου:



Μέσα στο τετράγωνο παράθυρο αυτού του μικρού πλακιδίου πληκτρολογούμε: 11



18…Συνεχίζουμε κατά τον ίδιο τρόπο σύροντας και τοποθετώντας τα πλακίδια για τα *Button2*, *Button3*, *Button4*, όπως κάναμε στα βήματα 14 ως και 16. Το τελικό αποτέλεσμα πρέπει να είναι όπως υποδεικνύεται στην παρακάτω φωτογραφία.



Όταν ολοκληρωθεί το πρόγραμμά μας είμαστε έτοιμοι να το φορτώσουμε στο κινητό και να το δοκιμάσουμε.

1. Πατάμε στο παράθυρο του App Inventor επάνω, στο μενού: Connect AI Companion, κατά τα γνωστά. Επίσης στο κινητό μας τηλέφωνο ανοίγουμε την εφαρμογή *AI Companion*. Επιλέγουμε στον υπολογιστή να ανεβάσουμε το πρόγραμμα στο κινητό με τη βοήθεια *QR Code* και κάνουμε το ανέβασμα.

2. Όταν η εφαρμογή ανοίξει, βρισκόμαστε στην οθόνη με τα 4 κουμπιά κάτω και ένα μπλε επίμηκες κουμπί επάνω.

3. Πρώτα πρέπει να πατήσουμε στο επίμηκες μπλε κουμπί *«PRESS to Connect with Bluetooth Device»* και να επιλέξουμε το HC-05 στη λίστα που βγαίνει με τις συζευγμένες συσκευές. Στη συνέχεια η εφαρμογή μας ξαναγυρίζει στην αρχική οθόνη με τα κουμπιά, όπου μπορούμε να ανάβουμε και να σβήνουμε τα δύο LED κατά βούληση.

ΠΡΟΣΟΧΗ: Αν μας εμφανιστεί σφάλμα ότι δεν μπορεί να γίνει η σύνδεση του κινητού με το HC-06 ή δεν φαίνεται το HC-06 στη λίστα των συσκευών, τότε:

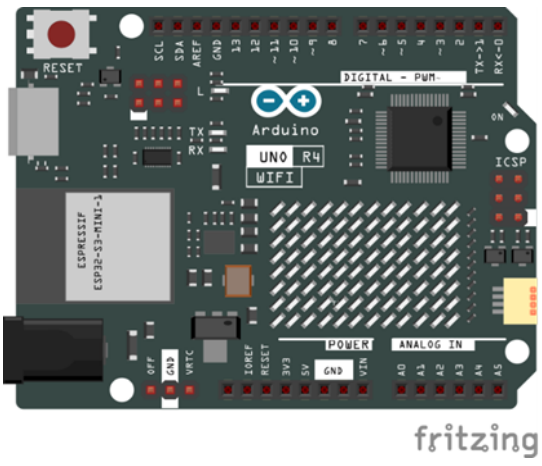
1. Ίσως να μην κάναμε πιο πριν ζεύξη του HC-06 με το κινητό μας. Την κάνουμε (από τις ρυθμίσεις Bluetooth του κινητού) και ξαναδοκιμάζουμε.

2. Ακόμη, ίσως ξεχάσαμε να ενεργοποιήσουμε το Bluetooth στο κινητό μας, οπότε το ενεργοποιούμε και ξαναδοκιμάζουμε.

2. Δημιουργία εφαρμογής AI με το Arduino

Σε αυτό το project θα φτιάξουμε μία εφαρμογή Arduino που θα επικοινωνεί με το ChatGPT. Για το σκοπό αυτό, θα χρησιμοποιήσουμε την πλακέτα Arduino Uno R4 WiFi, η οποία διαθέτει δυνατότητα σύνδεσης στο Internet.

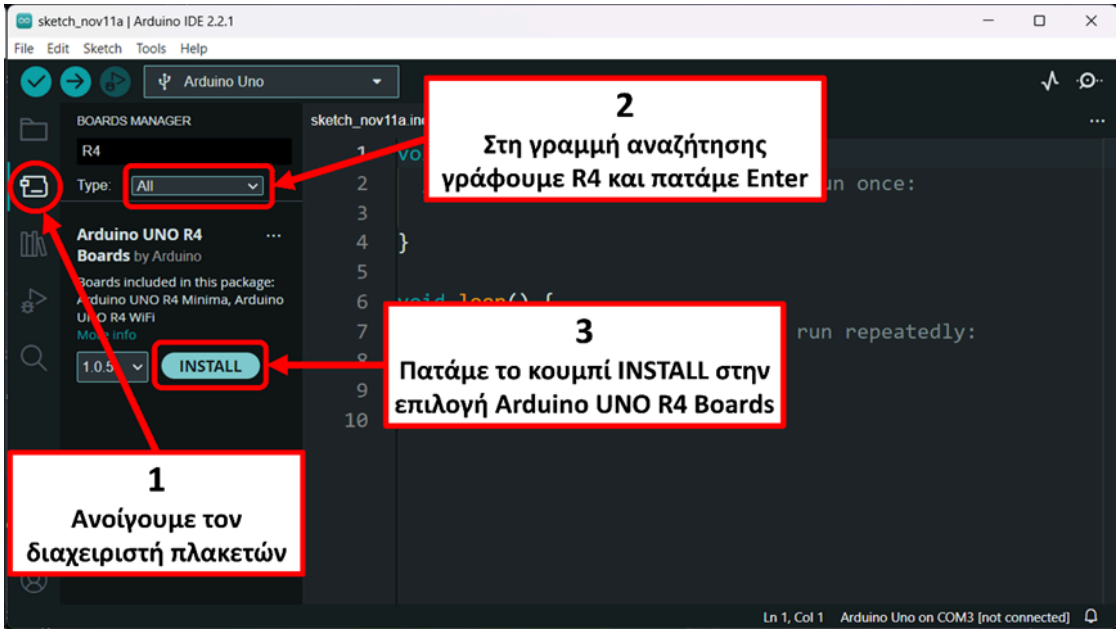
Το Arduino Uno R4 WiFi



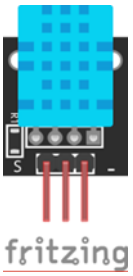
Το Arduino Uno R4 WiFi είναι μία νέα έκδοση πλακέτας. Έχει τις ίδιες διαστάσεις, την ίδια διάταξη ακροδεκτών και την ίδια τάση λειτουργίας (5V) με το Arduino Uno R3,

αλλά διαθέτει αυξημένες δυνατότητες. Ο βασικός μικροελεγκτής της πλακέτας, ο RA4M1, είναι ισχυρότερος με περισσότερη μνήμη και αποθηκευτικό χώρο. Επιπλέον, η πλακέτα περιλαμβάνει έναν δεύτερο μικροελεγκτή ESP32-S3, ο οποίος παρέχει δυνατότητες ασύρματης συνδεσιμότητας WiFi και BLE. Ένα ακόμα σημαντικό χαρακτηριστικό είναι το ενσωματωμένο LED Matrix, το οποίο αποτελείται από μία συστοιχία 12x8 LED.

Για να μπορέσουμε να προγραμματίσουμε την πλακέτα με το Arduino IDE θα πρέπει πρώτα να την προσθέσουμε μέσω του Boards Manager με τη διαδικασία που φαίνεται στην εικόνα.



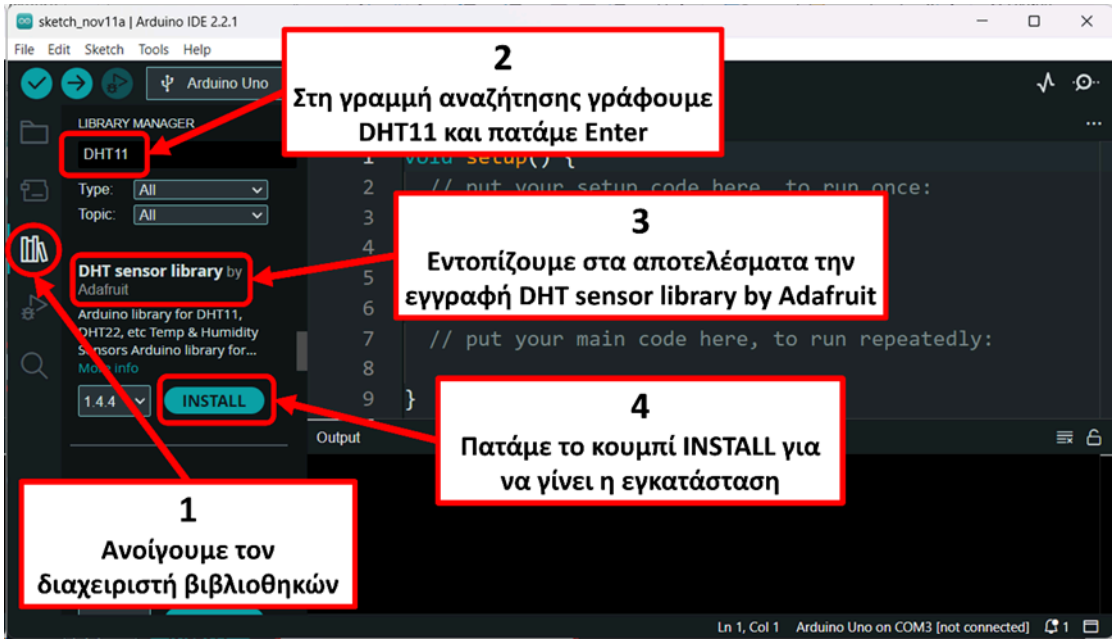
Ο Αισθητήρας DHT11



Ο DHT11 είναι ένας ψηφιακός αισθητήρας θερμοκρασίας και υγρασίας. Διατίθεται σκέτος ή ως “module” τοποθετημένος πάνω σε ειδική πλακέτα. Όπως φαίνεται στην εικόνα, η έκδοση “module” διαθέτει τρεις ακροδέκτες. Ο δεξιός με την ένδειξη – και ο μεσαίος, χρησιμοποιούνται για τροφοδοσία. Ο ακροδέκτης με την ένδειξη S είναι η έξοδος όπου ο αισθητήρας βγάζει τις μετρήσεις του. Ο DHT11 μετράει τη θερμοκρασία σε βαθμούς Κελσίου με ακρίβεια ενός βαθμού. Η υγρασία μετριέται ως ποσοστό επί τοις εκατό.

Ο αισθητήρας DHT11 είναι αρκετά διαδεδομένος και έτσι υπάρχουν πολλές βιβλιοθήκες που υποστηρίζουν τη χρήση του. Εμείς θα χρησιμοποιήσουμε το DHT

sensor library της εταιρείας Adafruit. Η προσθήκη της βιβλιοθήκης στο περιβάλλον του Arduino IDE γίνεται με το Library Manager.



Κατά τη διάρκεια της εγκατάστασης, το Library Manager ζητάει την άδειά μας για να εγκαταστήσει και το Adafruit Unified Sensor, που είναι απαραίτητο για τη λειτουργία του DHT sensor library.

Εφαρμογή 2 : Χαρακτηρισμός καιρικών συνθηκών με το ChatGPT

Το ChatGPT είναι ένα σύστημα τεχνητής νοημοσύνης της OpenAI, που χρησιμοποιεί αλγορίθμους επεξεργασίας φυσικής γλώσσας, για να μπορεί να επικοινωνεί με τους χρήστες/ριές του, με τρόπο που μοιάζει ανθρώπινος. Στην εφαρμογή μας θα δημιουργήσουμε έναν απλό μετεωρολογικό σταθμό με το Arduino R4 WiFi και τον αισθητήρα DHT11. Το Arduino θα μετράει τη θερμοκρασία και την υγρασία, και θα στέλνει τις μετρήσεις στο μοντέλο ChatGPT3.5-instruct, προκειμένου να λάβει έναν σύντομο λεκτικό χαρακτηρισμό των καιρικών συνθηκών. Στη συνέχεια θα προβάλει στο ενσωματωμένο LED Matrix ένα μήνυμα, που θα περιλαμβάνει τις αρχικές μετρήσεις και τον λεκτικό χαρακτηρισμό.

Δημιουργία API Key για την επικοινωνία με το ChatGPT

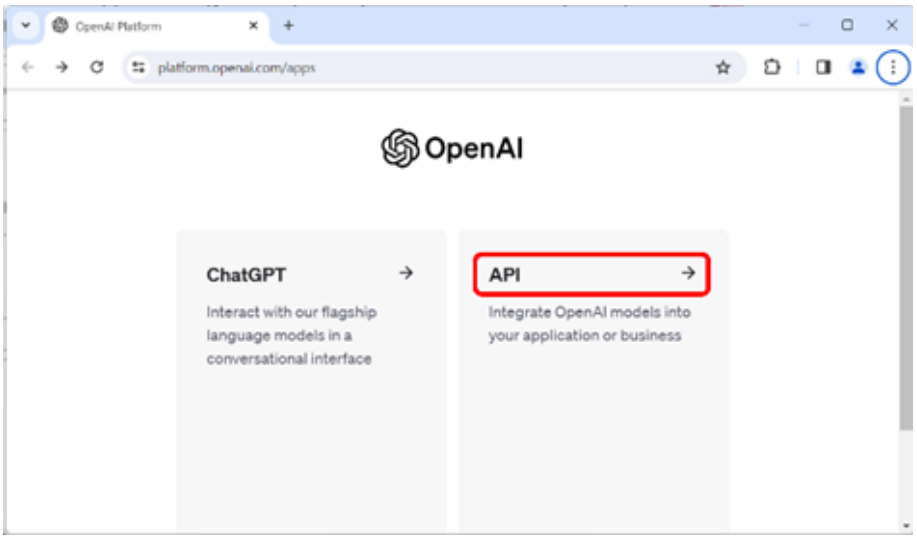
Η χρήση των υπηρεσιών του ChatGPT μέσα από το πρόγραμμα του Arduino, γίνεται με τη χρήση κατάλληλου API (Application Programming Interface). Η χρήση των API παρέχεται από την OpenAI ως συνδρομητική υπηρεσία και χρεώνεται με βάση τον όγκο των δεδομένων που ανταλλάσσονται. Ο όγκος των δεδομένων μετριέται σε tokens (ενδεικτικά μία αγγλική λέξη αντιστοιχεί κατά μέσο όρο σε 1.3 token ενώ για το

μοντέλο gpt-3.5-turbo-instruct η χρέωση είναι 0.0015$ ανά 1000 εισερχόμενα tokens και 0.0020$ ανά 1000 εξερχόμενα). Με βάση την τρέχουσα τιμολογιακή πολιτική της OpenAI, σε κάθε νέο/α χρήστη/ρια της υπηρεσίας παρέχεται δωρεάν πίστωση 5 δολαρίων με διάρκεια 3 μηνών. Λεπτομερείς πληροφορίες για τη χρέωση της χρήσης των API μπορείτε να βρείτε στη διεύθυνση https://openai.com/pricing.

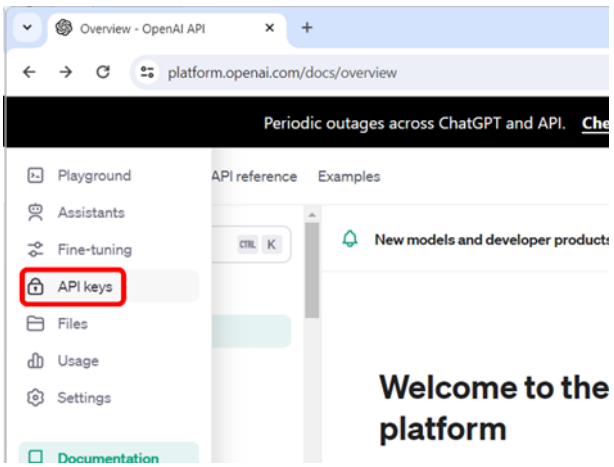
Για να μπορεί ή εφαρμογή να επικοινωνεί με το μοντέλο ChatGPT, θα πρέπει μαζί με κάθε αίτημα να στέλνει και ένα API key, με βάση το οποίο θα γίνεται η πιστοποίηση και η χρέωση. Η έκδοση του API key γίνεται μέσα από την ιστοσελίδα της OpenAI:

1. Ανοίγουμε τη σελίδα https://openai.com και συνδεόμαστε πατώντας το link Log in. Η σύνδεση μπορεί να γίνει είτε με εξειδικευμένο λογαριασμό OpenAI, είτε χρησιμοποιώντας έναν λογαριασμό Google, Microsoft ή Apple.

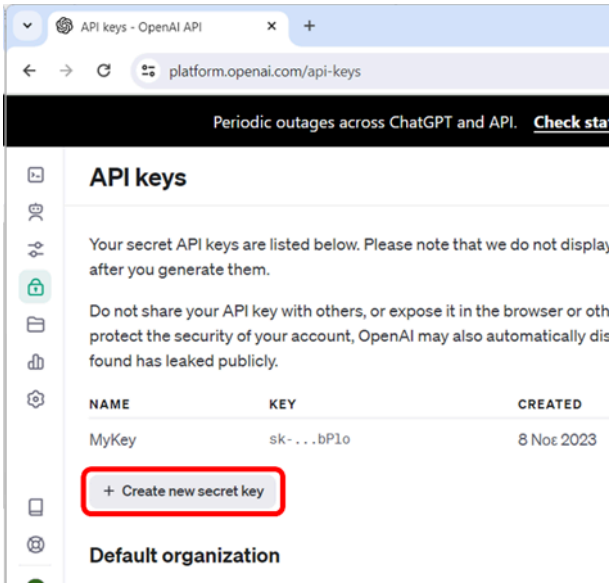
2. Στη νέα σελίδα επιλέγουμε API.



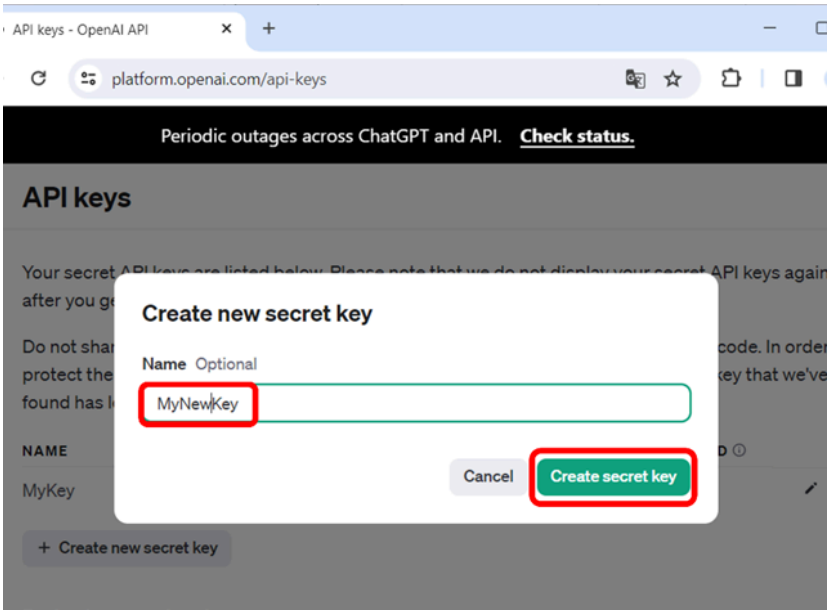
3. Στο μενού στην αριστερή πλευρά πατάμε API keys.



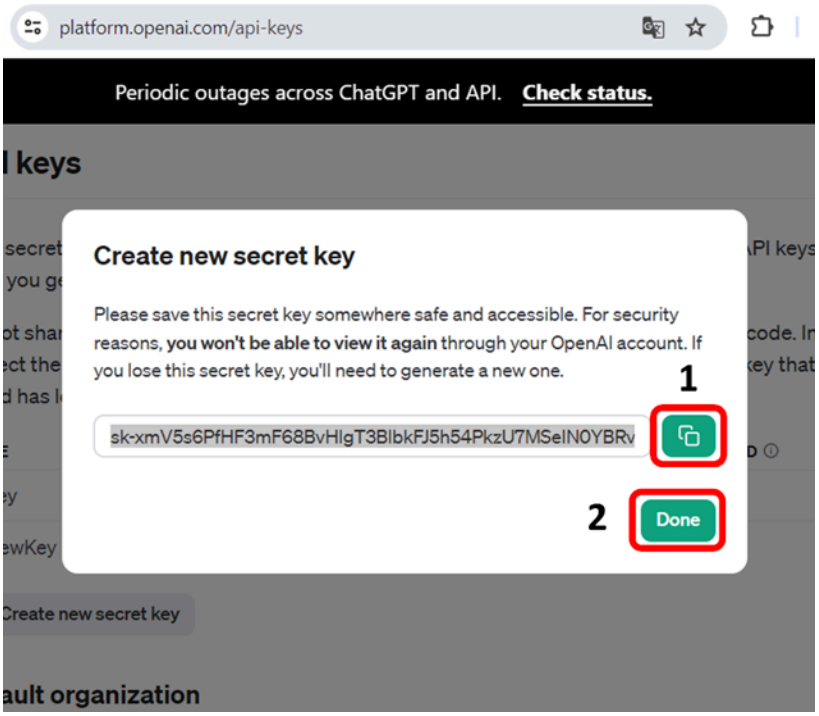
4. Επιλέγουμε Create new secret key.



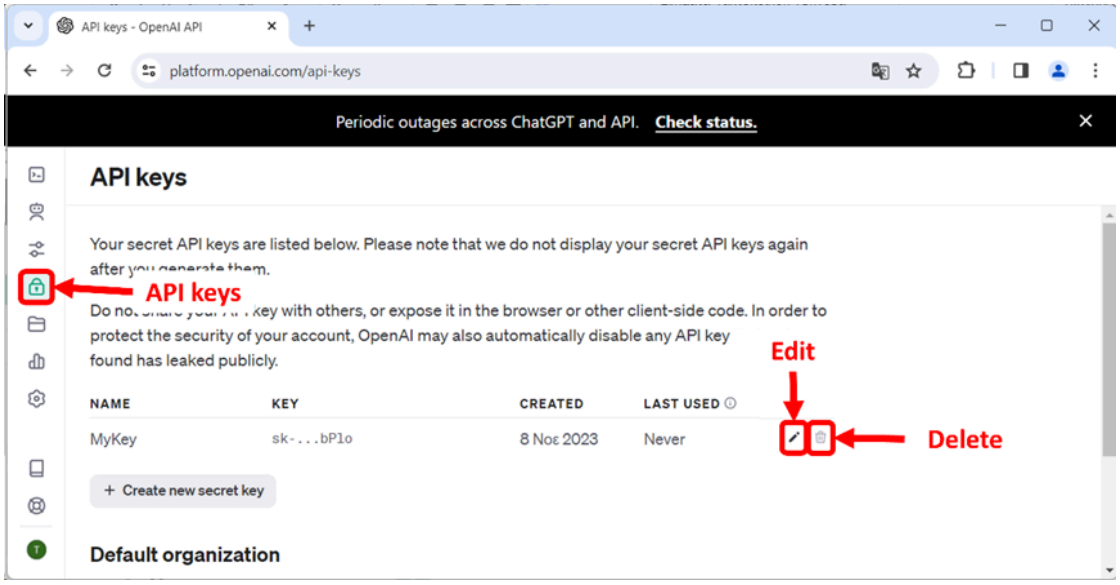
5. Προαιρετικά ορίζουμε ένα όνομα για το κλειδί και πατάμε το κουμπί Create secret key.



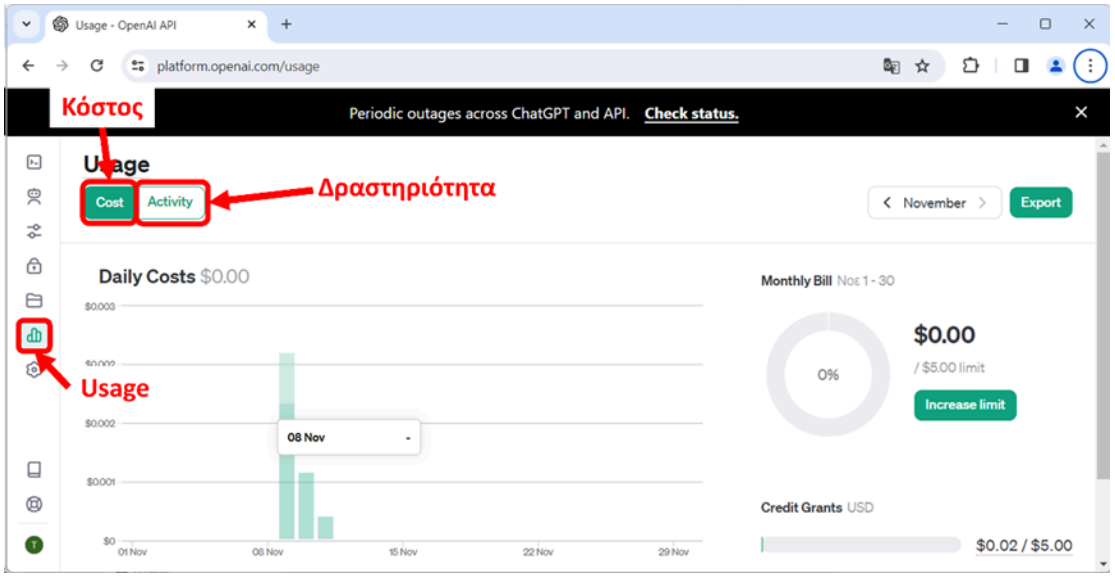
6. Αντιγράφουμε το κλειδί πατώντας το σχετικό κουμπί, το κάνουμε επικόλληση σε ένα κενό έγγραφο και το αποθηκεύουμε. Μετά πατάμε Done.



7. Επιλέγοντας από το αριστερό μενού API keys μπορούμε να διαχειριζόμαστε τα κλειδιά που έχουμε δημιουργήσει.



8. Πατώντας στο Usage βρίσκουμε χρήσιμες πληροφορίες για τη δραστηριότητα και τη χρέωση των APIs.



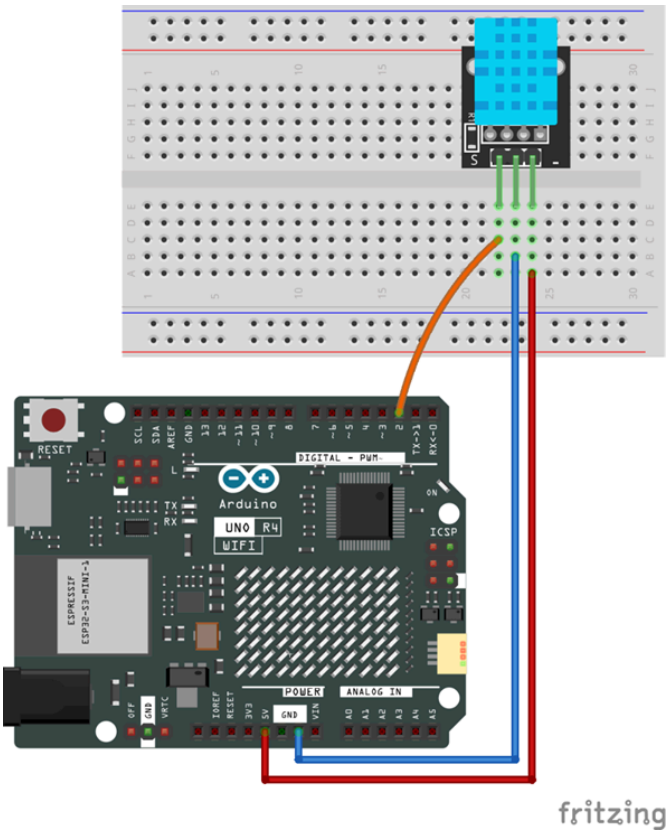
Υλικά

Θα χρησιμοποιήσουμε: Το Arduino Uno R4 WiFi, τον αισθητήρα DHT11, το breadboard και μερικά καλώδια jumper.

Κύκλωμα

Τοποθετούμε τον DHT11 πάνω στο breadboard και στη συνέχεια με τα καλώδια κάνουμε τις συνδέσεις που φαίνονται στον ακόλουθο πίνακα και στο κύκλωμα:

Pin DHT11	Pin Arduino
S	D2
Μεσαίος	5V
-	GND



Ανάπτυξη προγράμματος σε Arduino IDE

Στο πρόγραμμα, η λειτουργία της λήψης μετρήσεων, της επικοινωνίας με το ChatGPT και της εμφάνισης των αποτελεσμάτων εκτελούνται μέσα από τη setup. Με τον τρόπο αυτό, όταν συνδέετε την πλακέτα στο ρεύμα, η διαδικασία θα εκτελείται μόνο μία φορά και αν θέλετε να την επαναλάβετε θα πρέπει να πατήσετε το κουμπί reset του Arduino.

Για να λειτουργήσει η εφαρμογή, θα πρέπει να εγκατασταθούν στο Arduino IDE οι παρακάτω βιβλιοθήκες:

- DHT sensor library
- ArduinoHttpClient
- Arduino_JSON
- ArduinoGraphics

Η εγκατάσταση των βιβλιοθηκών πραγματοποιείται με τη χρήση του Library Manager, σύμφωνα με τη διαδικασία που παρουσιάστηκε προηγούμενα.
Στον κώδικα της εφαρμογής που παρατίθεται στη συνέχεια, θα πρέπει να γίνουν οι παρακάτω αλλαγές:

1. Στην εντολή
const char* ssid = "Your_WiFi_SSID";
να αντικαταστήσετε το Your_WiFi_SSID με το SSID του ασύρματου δικτύου σας.

2. Στην εντολή
const char* password = "Your_WiFi_Password";
να αντικαταστήσετε το Your_WiFi_Password με το Password του ασύρματου δικτύου.

3. Στην εντολή
String OpenAI_key = "Your_API_key";
να αντικαταστήσετε το Your_API_key με το API key που δημιουργήσατε στο OpenAI.

Για τη διευκόλυνσή σας οι εντολές εμφανίζονται τονισμένες

```
#include "WiFiS3.h"
#include <ArduinoHttpClient.h>
#include <Arduino_JSON.h>
#include "ArduinoGraphics.h"
#include <Arduino_LED_Matrix.h>
#include <Wire.h>
#include "DHT.h"

#define DHTPIN 2  //Pin σύνδεσης αισθητήρα από κύκλωμα
#define DHTTYPE DHT11
DHT dht(DHTPIN, DHTTYPE);

const char* ssid = "Your_WiFi_SSID";  //SSID ασύρματου δικτύου
const char* password = "Your_WiFi_Password"; //Password ασύρματου δικτύου
int status = WL_IDLE_STATUS;
WiFiSSLClient wifi;

HttpClient client = HttpClient(wifi, "api.openai.com", 443); //AI server
String OpenAImodel = "gpt-3.5-turbo-instruct";  //Ορισμός μοντέλου AI
String OpenAI_key = "Your_API_key";  //Το κλειδί API που δημιουργήσατε

ArduinoLEDMatrix matrix;

void setup() {
  Serial.begin(115200);
  matrix.begin(); //Αρχικοποίηση LED Matrix
  dht.begin(); //Αρχικοποίηση αισθητήρα
  connectToWiFi(); //Σύνδεση στο δίκτυο
  delay(1000);

  int humidity = dht.readHumidity(); //Λήψη μετρήσεων
  int temperature = dht.readTemperature();
  String weather = "Temp " + String(temperature) +
          "C Hum " + String(humidity) + "% "; //Μήνυμα με μετρήσεις
  String query = "Given a temperature of " + String(temperature) +
                 " degrees Celsius and humidity of " + String(humidity) +
```

```
            "%, characterize the weather in a few words:"; //Ερώτημα για ChatGPT
  weather = weather + Characterization(query); //Προσθήκη απόκρισης στο μήνυμα
  displayOnMatrix(weather); //Προβολή μηνύματος
}

void loop() {
}

void connectToWiFi() { //Σύνδεση στο ασύρματο δίκτυο
  unsigned long start = millis();
  while (status != WL_CONNECTED) {//Όσο δεν έχουμε συνδεθεί
        Serial.print("Attempting to connect to SSID: ");
        Serial.println(ssid);
        status = WiFi.begin(ssid, password); //Προσπάθεια σύνδεσης
        while ((millis() - start <= 10000) &&
        status != WL_CONNECTED); //Αναμονή μέχρι την επόμενη προσπάθεια
  }
  Serial.println("Connected");
}

String Characterization(String query) { //Επικοινωνία με το ChatGPT
  String jsonPayload = "{\"model\":\"" + OpenAImodel + "\",\"prompt\":\"" +
          query + "\",\"max_tokens\":10}"; //Προετοιμασία αιτήματος
  client.beginRequest(); //Αποστολή αιτήματος HTTP με Headers
  client.post("/v1/completions");
  client.sendHeader("Content-Type", "application/json");
  client.sendHeader("Authorization", "Bearer " + OpenAI_key);
  client.sendHeader("Content-Length", jsonPayload.length());
  client.beginBody();
  client.print(jsonPayload);
  client.endRequest();

  int httpResponseCode = client.responseStatusCode(); //Λήψη απόκρισης
  String response = client.responseBody();
  if (httpResponseCode == 200) { //Έλεγχος για σφάλμα
        JSONVar jsonData = JSON.parse(response); //Ανάκτηση χαρακτηρισμού
        String message = jsonData["choices"][0]["text"];
    message.trim();
        message = message + "        ";
        return message; //Επιστροφή χαρακτηρισμού
  } else {
        return "Failed to get characterization    "; //Επιστροφή μηνύματος σφάλματος
  }
}

void displayOnMatrix(String characterization) { //Εμφάνιση κειμένου στο LED Matrix
  for (int i = 0; i < 3; i++) { //Το μήνυμα εμφανίζεται 3 φορές
    matrix.beginDraw();
    matrix.stroke(0xFFFFFFFF);
    matrix.textScrollSpeed(70); //Ταχύτητα κύλισης
    matrix.textFont(Font_4x6); //Μέγεθος γραμματοσειράς
    matrix.beginText(10, 1, 0xFFFFFF);
    matrix.println(characterization); //Εμφάνιση κειμένου
    matrix.endText(SCROLL_LEFT);
    matrix.endDraw();
  }
}
```

Συζήτηση για Project

Στο 6ο τρίωρο συζητάμε με τους μαθητές για την τελική μορφή των project τους, ώστε να γίνει η παραγγελία όσων επιπλέον υλικών χρειάζονται και δεν έχουν ήδη παραγγελθεί. Ξεκινάει η διαδικασία υλοποίησης των project, πάντα σε συνεννόηση με τους υπεύθυνους του προγράμματος.

ΠΑΡΑΡΤΗΜΑ: Αισθητήρες και πλακέτες για Arduino

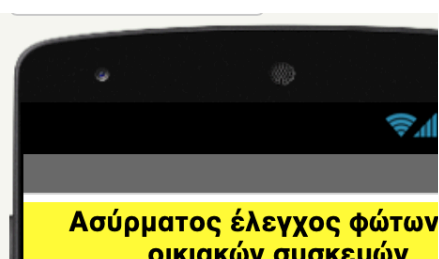
Η πλακέτα Arduino είναι ένας μικροελεγκτής που δέχεται πολλούς αισθητήρες και εξαρτήματα εισόδου (INPUT) και εξόδου (OUTPUT), καθιστώντας το ικανό να πραγματοποιήσει ακόμα και τις πιο τρελές ιδέες που έχουμε στο μυαλό μας.

Παρακάτω σας παραθέτουμε μια λίστα με μερικά από τα εξαρτήματα που είναι συμβατά με το Arduino, τα 10 πρώτα τα χρησιμοποιήσαμε στα μαθήματα.

- Διακόπτης
- LED απλό
- Buzzer
- Ποτενσιόμετρο
- Φωτοαντίσταση
- LCD 16x2
- Σερβομηχανισμός
- Αισθητήρας υπερήχων
- LM35 (Θερμοκρασίας)
- Bluetooth
- NRF24L01 (Ασύρματη επικοινωνία)
- Αισθητήρας κίνησης με υπέρυθρες ή «Υπέρυθρο ραντάρ»
- ADXL345 (Επιτάχυνσης σε x, y, z άξονα ή βαρύτητας)
- DHT11 (Υγρασίας αέρα και θερμοκρασίας αέρα)
- GSM module (Για κλήσεις και μηνύματα με το δίκτυο κινητής τηλεφωνίας)
- Ήχου ανίχνευση- ψηφιακός (με μικρόφωνο)
- UV (έντασης υπεριώδους ακτινοβολίας, αναλογικός)
- Αερίων (CO, CO2, CH4 κ.λ.π.)
- Πίεσης ατμόσφαιρας
- Μαύρου/άσπρου (με υπέρυθρο φως π.χ. για ανίχνευση εμποδίων ή ακολούθηση μαύρης γραμμής στο πάτωμα)
- Δύναμης
- Κάμψης (παραμόρφωσης)
- Κλίσης (γυάλινη αμπούλα με υδράργυρο)
- Reed switch (διακόπτης γλωσσίδας – κλείνει με τη βοήθεια μαγνήτη)
- Μετάλλων
- Βροχής (αγωγιμότητα νερού)
- Φλόγας (με υπέρυθρες)
- Laser module (παράγει δέσμη laser)
- Δονήσεων
- Υγρασίας εδάφους (αγωγιμότητας εδάφους)

- RTC module (Ρολόι πραγματικού χρόνου DS1307 ή DS3231)
- Μαγνητικού πεδίου σε x,y,z άξονες (μαγνητόμετρο)
- RGB LED module
- Πλακέτα για οδήγηση μικρών μοτέρ (με L293D) π.χ. για αυτοκινητάκια
- Θερμοκρασίας ψηφιακός (DS18B20)
- Πλακέτα ρελέ (για οδήγηση μεγάλων φορτίων και μεγάλης τάσης)
- Joystick αναλογικό

Εφαρμογή έμπνευσης από τους μαθητές της Ρόδου



ΕΛΕΓΧΟΣ ΦΩΤΙΣΜΟΥ ΚΑΙ ΟΙΚΙΑΚΩΝ ΣΥΣΚΕΥΩΝ ΜΕΣΩ BLUETOOTH

Περιγραφή προβλήματος:
Οι νέες τεχνολογίες προσφέρουν ολοένα και περισσότερες δυνατότητες για να έχουμε αυτοματισμούς στην καθημερινότητά μας. Όλο και περισσότερες συσκευές συνδέονται στο διαδίκτυο και μας επιτρέπουν να ελέγχουμε τα πάντα εξ αποστάσεως. Θα μπορούσαμε λοιπόν να ελέγχουμε από το κινητό τηλέφωνό μας , το φωτισμό και τις διάφορες συσκευές του σπιτιού μας.

Περιγραφή προτεινόμενης λύσης:
Με αυτή την εφαρμογή , μπορούμε μέσω του κινητού τηλεφώνου μας , να ανάψουμε και να σβήσουμε τα φώτα του δωματίου και της κουζίνας, να ανάψουμε και να σβήσουμε το φούρνο, να εκκινήσουμε και να σταματήσουμε τον ανεμιστήρα. Μπορούμε ακόμη να ανοίξουμε και να κλείσουμε την πόρτα του σπιτιού. Η εφαρμογή επίσης μας πληροφορεί για την τρέχουσα θερμοκρασία του χώρου

Κώδικας Arduino

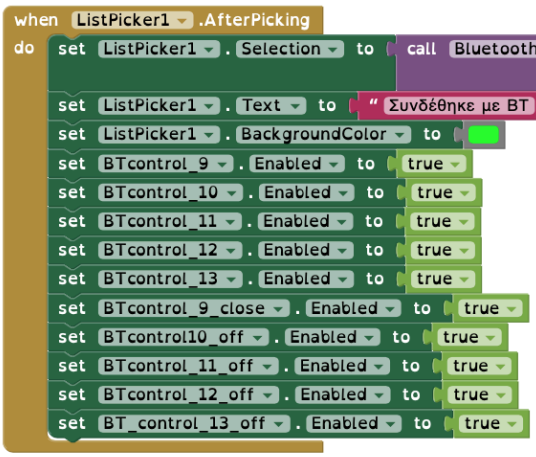
```
#include <Servo.h>
#include <SoftwareSerial.h>

SoftwareSerial mySerial(2,3);
Servo myservo; // Δημιούργησε ένα αντικείμενο τύπου Servo

int door_close=0;
int door_open=90;

void setup() {

 analogReference(INTERNAL);

 Serial.begin(9600);
 mySerial.begin(9600);
 myservo.attach(9); // Σχετίζει το σερβο με το πιν 9

 myservo.write(door_close); // door close
 delay(2000);
 myservo.detach();// απενεργοποιώ το σερβο

pinMode(10,OUTPUT);
pinMode(11,OUTPUT);
pinMode(12,OUTPUT);
pinMode(13,OUTPUT);
}

void loop() {

byte readmyserial;
readmyserial='a';
```

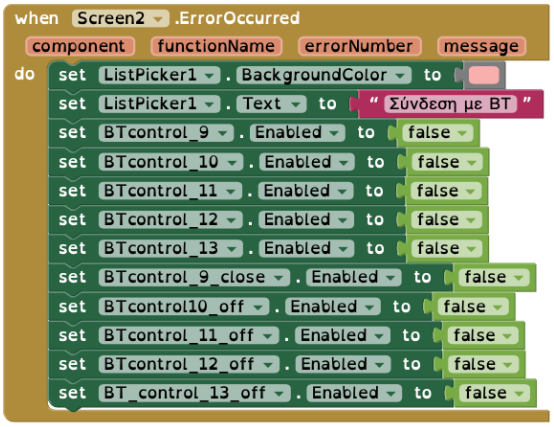
```
myservo.detach();

 if (mySerial.available()>0)                    //
   {
      readmyserial=mySerial.read();
   }
       if (readmyserial==10 )
            digitalWrite(10,HIGH);
       if (readmyserial==101 )
            digitalWrite(10,LOW);

       if (readmyserial==11 )
            digitalWrite(11,HIGH);
       if (readmyserial==111 )
            digitalWrite(11,LOW);

       if (readmyserial==12 )
            digitalWrite(12,HIGH);
       if (readmyserial==121 )
            digitalWrite(12,LOW);

       if (readmyserial==13 )
            digitalWrite(13,HIGH);
       if (readmyserial==131 )
            digitalWrite(13,LOW);

        if (readmyserial==9 )
              {
                myservo.attach(9);
                myservo.write(door_close);
                delay(2000);
                myservo.detach();
              }

        if (readmyserial==91 )
              {
                myservo.attach(9);
                myservo.write(door_open);
                delay(2000);
                myservo.detach();
              }

byte temperature;
 temperature=(analogRead(A0)*1.1*100/1024);
 delay(2000);
 mySerial.write(temperature);
 Serial.println(temperature);
}
```

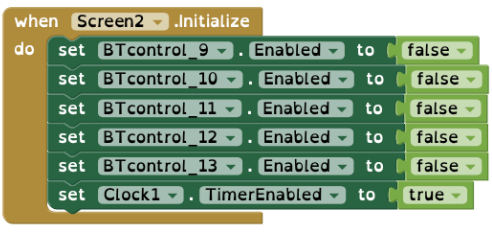

9:48
Πιέστε για σύνδεση με bluetooth
Άνοιγμα της πόρτας
Κλείσιμο πόρτας
Άνοιγμα λαμπήρων καθιστικού
Κλείσιμο λαμπήρων καθιστικού
Άνοιγμα λαμπήρων κουζίνας
Κλείσιμο λαμπήρων κουζίνας
Άνοιγμα φούρνου
Κλείσιμο φούρνου
Άνοιγμα ανεμιστήρα
Κλείσιμο ανεμιστήρα
Non-visible components
BluetoothClient1 Clock1


9:48
Screen3
Προβολή της υπάρχουσας κατάστασης των συσκευών
Η θερμοκρασία του χώρου είναι: Text C
Η πόρτα είναι: Text for Label6
Οι λαμπήρες του καθιστικού είναι: Text for Label8
Οι λαμπήρες της κουζίνας είναι: Text for Label10
Ο φούρνος είναι: Text for Label12
Ο ανεμιστήρας είναι: Text for Label14
Επιστροφή
Non-visible components
Clock1 BluetoothClient1

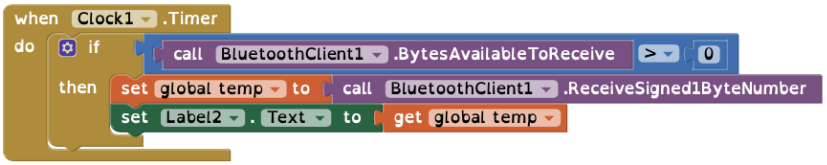

when ListPicker1 .BeforePicking
do set ListPicker1 . Elements to BluetoothClient1 . AddressesAndNames

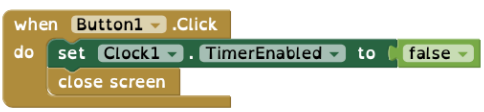

when ListPicker1 .AfterPicking
do set ListPicker1 . Selection to call BluetoothClient1 .Connect
address ListPicker1 . Selection
set ListPicker1 . Text to " Συνδέθηκε με BT "
set ListPicker1 . BackgroundColor to
set BTcontrol_9 . Enabled to true
set BTcontrol_10 . Enabled to true
set BTcontrol_11 . Enabled to true
set BTcontrol_12 . Enabled to true
set BTcontrol_13 . Enabled to true
set BTcontrol_9_close . Enabled to true
set BTcontrol10_off . Enabled to true
set BTcontrol_11_off . Enabled to true
set BTcontrol_12_off . Enabled to true
set BT_control_13_off . Enabled to true

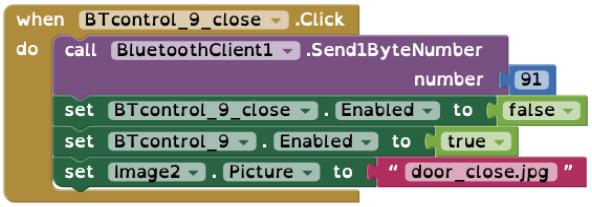

when Screen2 .ErrorOccurred
component functionName errorNumber message
do set ListPicker1 . BackgroundColor to
set ListPicker1 . Text to " Σύνδεση με BT "
set BTcontrol_9 . Enabled to false
set BTcontrol_10 . Enabled to false
set BTcontrol_11 . Enabled to false
set BTcontrol_12 . Enabled to false
set BTcontrol_13 . Enabled to false
set BTcontrol_9_close . Enabled to false
set BTcontrol10_off . Enabled to false
set BTcontrol_11_off . Enabled to false
set BTcontrol_12_off . Enabled to false
set BT_control_13_off . Enabled to false

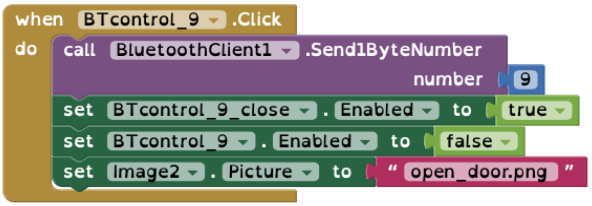

when Screen2 .Initialize
do set BTcontrol_9 . Enabled to false
set BTcontrol_10 . Enabled to false
set BTcontrol_11 . Enabled to false
set BTcontrol_12 . Enabled to false
set BTcontrol_13 . Enabled to false
set Clock1 . TimerEnabled to true

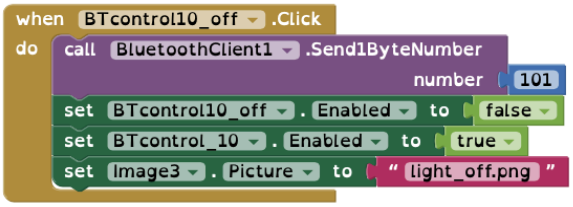

initialize global temp to 0

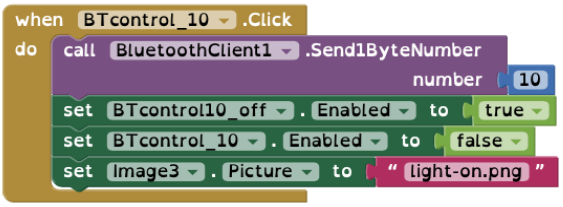

when Clock1 .Timer
do if call BluetoothClient1 .BytesAvailableToReceive > 0
then set global temp to call BluetoothClient1 .ReceiveSigned1ByteNumber
set Label2 . Text to get global temp

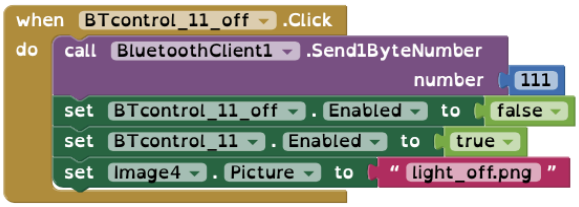

when Button1 .Click
do set Clock1 . TimerEnabled to false
close screen

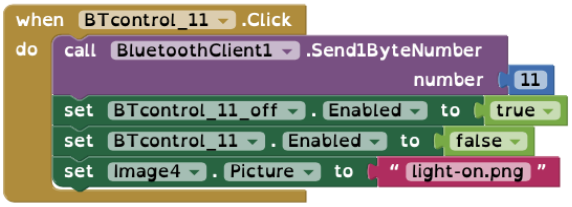


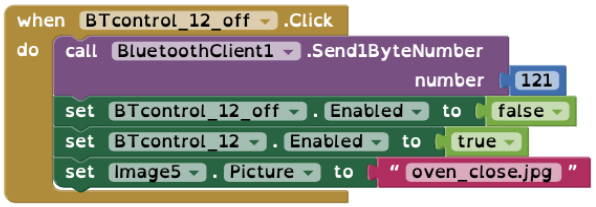


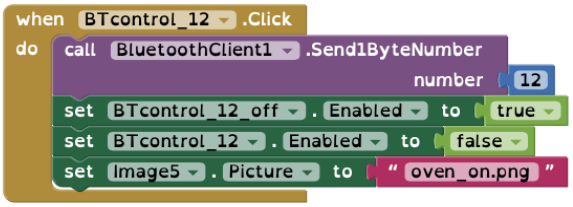


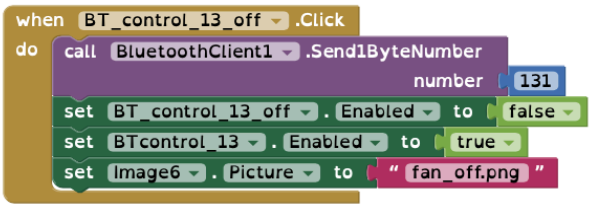


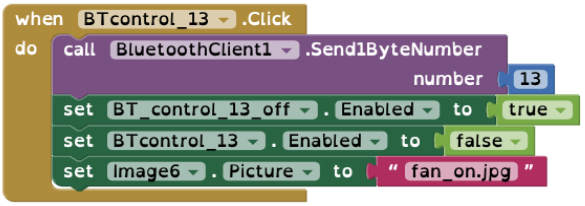






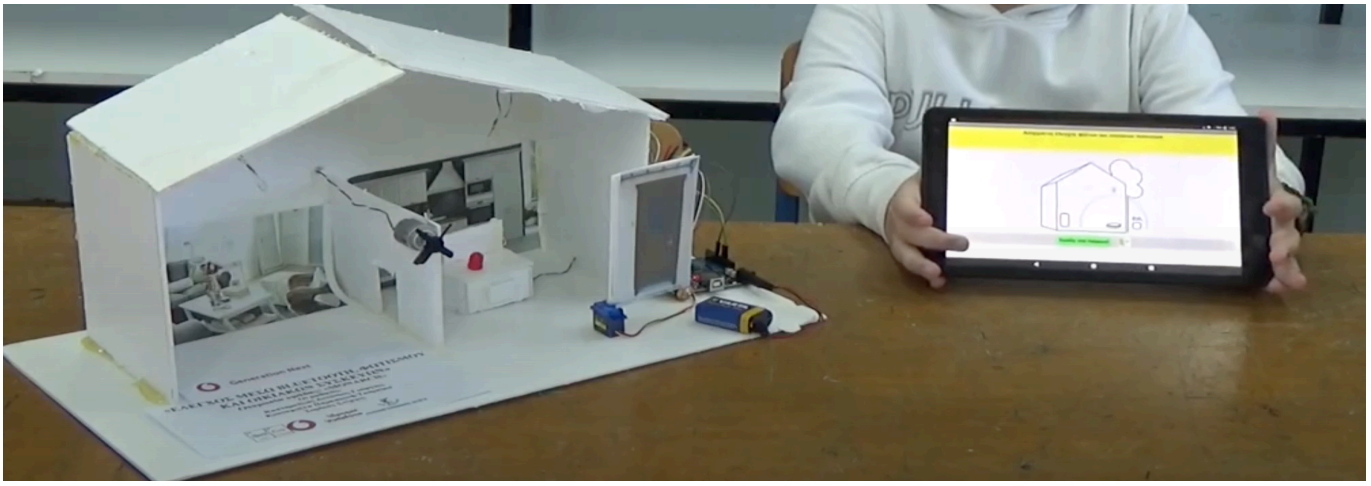






Περιγραφή τεχνολογίας που χρησιμοποιήθηκε:

Arduino Uno
Bluetooth HC-05
Αισθητήρας θερμοκρασίας LM35
Servo
Moter



7ο, 8ο, τρίωρο

Αφιερωμένα στην κατασκευή και καταγραφή των μαθητικών πρότζεκτ.

Οι μαθητές με την ολοκλήρωση των μαθημάτων πρέπει να συμπληρώσουν τα **ερωτηματολόγια λήξης του προγράμματος μας.**

1. Διαγωνισμός Καινοτομίας

Παρουσίαση των Πρότζεκτ και του Διαγωνισμού

Ο τελικός στόχος του προγράμματος είναι οι μαθητές να χρησιμοποιήσουν έμπρακτα τις γνώσεις και δεξιότητες που απέκτησαν, ενεργοποιώντας τη φαντασία τους και έχοντας ως στόχο την **επίλυση κάποιου τοπικού προβλήματος**. Με την υλοποίηση του τελικού πρότζεκτ, οι μαθητές ολοκληρώνουν τον κύκλο μαθημάτων και παίρνουν την βεβαίωση συμμετοχής τους στο πρόγραμμα. Μέσα από αυτή τη διαδικασία, οι μαθητές έχουν την ευκαιρία να φτιάξουν την δική τους **ερευνητική ομάδα** και να βιώσουν τα στάδια ενός **πρότζεκτ**, από την έρευνα μέχρι τον πειραματισμό και την υλοποίησή του. Εξοπλισμένοι με τεχνολογικά εργαλεία και ακολουθώντας την διεπιστημονική μέθοδο, οι μαθητές συνεργάζονται για να αποτυπώσουν την ιδέα τους με τον πιο δημιουργικό τρόπο. Η SciCo υποστηρίζει τη διαδικασία των κατασκευών τόσο με τη βοήθεια κάποιου εκπαιδευτικού όσο και με τη παροχή των υλικών που θα χρειαστούν για την υλοποίηση του.

Κάθε ομάδα μαθητών έχει το δικαίωμα να συμμετέχει στον **διαγωνισμό καινοτομίας**, ο οποίος έχει σχεδιαστεί ως επιβράβευση- τόσο για τους μαθητές όσο και για τους εκπαιδευτικούς των ομάδων που θα βραβευτούν- για την ευρηματικότητα και το αντίκτυπο που έχουν τα πρότζεκτ τους στο κοινωνικό σύνολο

2. Εισαγωγή στην διερευνητική μάθηση (Project Based Learning)

Η διερευνητική μάθηση ή αλλιώς η διεπιστημονική μέθοδος συνδέεται άμεσα με την επίλυση προβλημάτων (*problem solving*) και οι μαθητές του «Generation Next» θα έχουν την ευκαιρία να λύσουν ένα ρεαλιστικό πρόβλημα (*real world problem*) ακολουθώντας τα διερευνητικά βήματα, ακριβώς όπως μια ομάδα ερευνητών!

Μετά την ολοκλήρωση των μαθημάτων, η κάθε ομάδα μαθητών διαλέγει ένα πρόβλημα προς επίλυση και κάνει το δικό της πρότζεκτ το οποίο θα δουλέψει τα τρία τελευταία τρίωρα του προγράμματος.

Στο διάστημα του Generation Next οι μαθητές διδάσκονται ένα εύρος ομαδικών δραστηριοτήτων STEM και Coding, εμπνέονται από τα μαθήματα και στο τέλος κάθε κύκλου διαλέγουν ένα πρόβλημα (κατά προτίμηση τοπικό) για να δώσουν μια τεχνολογική λύση. Οι μαθητικές κατασκευές που επιλύουν κάποιο τοπικό πρόβλημα αποτελεί το κλείσιμο του προγράμματος μας, όπου τα παιδιά συνθέτουν τις νέες γνώσεις που απέκτησαν με την φαντασία και την δημιουργικότητα τους και μπαίνουν στον ρόλο του ερευνητή, μηχανικού και εφευρέτη.

1. Επιλογή θέματος προς επίλυση

Το πρώτο βήμα κάθε ερευνητικής διαδικασίας είναι η επιλογή του θέματος. Αυτή μπορεί να πηγάζει από κάποια προσωπική εμπειρία κάποιου θέματος το οποίο έχουμε συναντήσει στην προσωπική μας ζωή, ή από την παρατήρηση προβλημάτων που αντιμετωπίζουν οι άνθρωποι στο οικογενειακό μας περιβάλλον, στο σχολείο ή στην τοπική μας κοινωνία. Το πρόγραμμα ενθαρρύνει τους μαθητές να αναγνωρίσουν κάποιο τοπικό πρόβλημα που επηρεάζει την ευρύτερη σχολική ή τοπική τους κοινότητα.

Αυτό προϋποθέτει ότι οι μαθητές θα μπουν σε μια διαδικασία έρευνας ώστε να αναγνωρίσουν τις ανάγκες και τις ελλείψεις που παρατηρούν γύρω τους. Η διαδικασία αυτή, εκτός από το ότι τους μαθαίνει τα βήματα της ερευνητικής διαδικασίας, αναπτύσσει την παρατηρητικότητά τους αλλά και τις συναισθηματικές τους δεξιότητες και τη διάθεση προσφοράς προς το κοινωνικό σύνολο. Χωρίζουμε το πρώτο βήμα της αναγνώρισης του προβλήματος σε δύο επιμέρους διεργασίες:

Α) Διερεύνηση της τοπικής κοινωνίας: ενθαρρύνουμε τους μαθητές να συζητήσουν και να πραγματοποιήσουν μια διερεύνηση των ιδιαίτερων χαρακτηριστικών της τοπικής κοινωνίας ή της πόλης τους. Με αυτό τον τρόπο θα μπορέσουν να συνειδητοποιήσουν τις ιδιαίτερες ανάγκες ή ελλείψεις οι οποίες θα μπορούσαν να επιλυθούν μέσω μιας τεχνολογικής λύσης. Κατευθύνουμε τα παιδιά ώστε μέσω της συζήτησης να αναγνωρίσουν:

- Τα βασικά χαρακτηριστικά της περιοχής τους, όπως είναι η γεωγραφική θέση, το κλίμα, ιδιαίτερα καιρικά φαινόμενα που επηρεάζουν τη ζωή των κατοίκων ή χαρακτηριστικά που θα μπορούσαν να υποστηρίξουν την ανάπτυξη ανανεώσιμων πηγών ενέργειας, όπως μεγάλη ηλιοφάνεια.
- Τις κύριες και παραδοσιακές πηγές εισοδήματος στην τοπική κοινωνία, για παράδειγμα αν πρόκειται για μια περιοχή με αγροτική και κτηνοτροφική παραγωγή, τουριστική ανάπτυξη ή άλλα παραδοσιακά επαγγέλματα που χαρακτηρίζουν την τοπική κοινότητα
- Ιδιαίτερα χαρακτηριστικά της τοπικής κοινότητας, ανάγκες και ελλείψεις που επηρεάζουν τους ίδιους ή συνανθρώπους τους και οι οποίες θα μπορούσαν να αντιμετωπιστούν μέσω μιας τεχνολογικής λύσης.
- Τοπικά πλεονεκτήματα, όπως μνημεία ιστορικής ή αρχαιολογικής αξίας τα οποία θα μπορούσαν να αναδειχθούν.
- Τοπικούς φορείς οι οποία θα μπορούσαν να εμπλακούν στην διερευνητική διαδικασία, όπως δημαρχεία, σχολεία, συλλόγους (ορειβατικούς, αστρονομικούς, μουσικούς, ποδηλατικούς) κλπ.

Η διερευνητική συζήτηση θα μπορούσε να διευκολυνθεί με τη δημιουργία ενός φύλλου εργασίας, στο οποίο οι μαθητές θα συμπληρώνουν τα χαρακτηριστικά τα οποία αναγνωρίζουν και θα απαντούν σε σχετικές ερωτήσεις που τους κατευθύνουν για να επιλέξουν ένα πρόβλημα προς επίλυση:

«Από όλα τα παραπάνω, πού εντοπίζετε κάποιο πρόβλημα που θα σας ενδιέφερε να επιλύσετε;»

«Σε ποιόν τοπικό φορέα θα μπορούσατε να απευθυνθείτε για να μάθετε περισσότερα σε σχέση με το θέμα που σας ενδιαφέρει;»

Β) Επιλογή του θέματος προς επίλυση και διερεύνηση λύσεων

Μετά την ολοκλήρωση της αρχικής διερεύνησης, παροτρύνουμε τα παιδιά να διερευνήσουν ατομικά και σαν ομάδα, να έρθουν με κάποια εναλλακτικά προβλήματα προς επίλυση και προτεινόμενες λύσεις και να τις συζητήσουμε. Ως πηγές έμπνευσης,

στην πλατφόρμα του Generation Next στην ενότητα Discover και στην ενότητα Διαγωνισμός, όπου παρουσιάζονται τα projects των Πανελληνίων Διαγωνισμών.

2. Υλοποίηση του project

Στη συνέχεια παρουσιάζουμε στους μαθητές τα βήματα υλοποίησης του ερευνητικού project που θα οδηγήσει στην επίλυση του προβλήματος που θα επιλέξουν. Αυτά περιλαμβάνουν το σχεδιασμό της λύσης, τον καθορισμό των απαραίτητων υλικών, την υλοποίηση, την παραγωγή και την παρουσίαση της εργασίας.

Α. Απλοποίηση του προβλήματος. Οι μαθητές πρέπει να αναγνωρίσουν ποια είναι τα βασικά στοιχεία που αποτελούν το πρόβλημα, και πώς μια τεχνολογική λύση θα μπορούσε να αντιμετωπίσει ένα ή περισσότερα από αυτά. Για παράδειγμα, εάν το πρόβλημα είναι οι πλημμύρες, θα μπορούσε να σχεδιαστεί ένας συναγερμός που ειδοποιεί για την άνοδο της στάθμης του νερού, μια αυτόματη αντλία απομάκρυνσης υδάτων, μια αυτόματη ιρλανδική διάβαση για ευάλωτα σημεία του οδικού δικτύου ή κάποια άλλη τεχνολογική λύση που αντιμετωπίζει κάποια πτυχή του προβλήματος. Οι μαθητές μπορούν να χωρίσουν το πρόβλημα στα επιμέρους στοιχεία του, και στη συνέχεια κατά τη διαδικασία της έρευνας να καταλήξουν με ποιο τρόπο θέλουν να το προσεγγίσουν.

Β. Δημιουργία ομάδας και ανάθεση ρόλων. Αφού έχει επιλεγεί το πρόβλημα προς επίλυση, κάθε μέλος της ομάδας πρέπει να αναλάβει έναν από τους διερευνητικούς ρόλους, με βάση την προσωπική του κλίση και τις δεξιότητές του. Ένα ή περισσότερα μέλη της ομάδας μπορούν να αναλάβουν το κομμάτι της αρχικής έρευνας στο διαδίκτυο για να εντοπίσουν ποιές τεχνολογικές λύσεις έχουν προταθεί για παρόμοια προβλήματα και πώς θα μπορούσαν να κατασκευαστούν με τα υλικά και τις δυνατότητες που υπάρχουν. Ένα άλλο μέλος μπορεί να αναλάβει την κατασκευή, άλλο το κομμάτι του προγραμματισμού, ένα τρίτο την καταγραφή της διαδικασίας μέσω φωτογραφιών και βίντεο κ.ο.κ.

Γ. Έρευνα. Στη φάση αυτή οι μαθητές έχουν επιλέξει το πρόβλημα που θέλουν να επιλύσουν, και διερευνούν τις πιθανές λύσεις. Βασικό κομμάτι σε κάθε διερευνητική διαδικασία είναι να εντοπιστούν οι υπάρχουσες λύσεις που έχουν προτείνει άλλοι ερευνητές, και πώς αυτές θα μπορούσαν να αξιοποιηθούν, να βελτιωθούν ή να αποτελέσουν πηγή έμπνευσης για την επίλυση του δικού μας προβλήματος. Η έρευνα δεν στηρίζεται στην παρθενογένεση ιδεών, αλλά στη βελτίωση και την ανάπτυξη της υπάρχουσας γνώσης Μέρος της έρευνας αυτής είναι και η επιλογή των τεχνολογικών εργαλείων και των υλικών που θα χρειαστούν. Για την επιλογή αυτή οι μαθητές πρέπει να λάβουν υπόψη τις δυνατότητες κάθε εργαλείου, με βάση την εμπειρία και την έρευνά τους, τις δεξιότητες που οι ίδιοι έχουν αποκτήσει στη χρήση του και τον διαθέσιμο χρόνο που έχουν για την υλοποίηση του project.. Η ερευνητική διαδικασία αυτή προτείνουμε να πραγματοποιηθεί μεταξύ του 6^{ου} και 7^{ου} τρίωρου μαθήματος, ώστε στο 7^{ο} μάθημα ο εκπαιδευτικός να συζητήσει και να συμβουλεύσει τους μαθητές σχετικά με την τελική λύση που θα επιλέξουν να εφαρμόσουν. Παράλληλα ο εκπαιδευτικός συμβουλεύεται τον εκπρόσωπο της SciCo για την καλύτερη δυνατή λύση και μετέπειτα αγορά των απαιτούμενων υλικών.

Δ. Κατασκευή και πειραματισμός. Στη φάση αυτή οι μαθητές κατασκευάζουν τη λύση που έχουν αποφασίσει, συμβάλλοντας ο καθένας με το ρόλο που του έχει ανατεθεί από την ομάδα. Είναι πολύ πιθανό να συναντήσουν προβλήματα στη διαδικασία κατασκευής. Ενθαρρύνουμε και βοηθάμε τους μαθητές να προσπαθήσουν να προσπαθήσουν να επιλύσουν τα προβλήματα αυτά και να προχωρήσουν σε εναλλακτικές λύσεις με τους διαθέσιμους πόρους.

Ε. Εμπλοκή της τοπικής κοινωνίας. Εφόσον πρόκειται για τεχνολογική λύση πάνω σε ένα τοπικό πρόβλημα, είναι πολύ σημαντικό να υπάρξει διάδραση με τους τοπικούς φορείς τους οποίους αυτή αφορά. Οι μαθητές ενθαρρύνονται να παρουσιάσουν την ιδέα και τη λύση τους στους σχετικούς φορείς, τόσο κατά τη διάρκεια όσο και μετά την ολοκλήρωση του project, και να λάβουν υπόψη τις προτάσεις τους. Μετά την δημιουργία της λύσης, η συνεργασία με τους φορείς αυτούς (τη διεύθυνση του σχολείου, το δήμο, την πυροσβεστική κλπ) μπορεί να οδηγήσει στην πρακτική εφαρμογή της σε μεγάλη κλίμακα. Έτσι οι μαθητές θα δουν τη λύση τους να έχει αποτελέσματα στη ζωή των συνανθρώπων τους.

ΣΤ. Καταγραφή, αποτύπωση και παρουσίαση της διαδικασίας και του αποτελέσματος. Η καταγραφή της πειραματικής διαδικασίας είναι αναπόσπαστο βήμα κάθε ερευνητικού project. Τονίζουμε στους μαθητές τη σημασία του να καταγράφουν τα βήματα που εκτελούν, τα υλικά που χρησιμοποιούν και την διαδικασία κατασκευής. Πολύ σημαντική είναι επίσης η συλλογή φωτογραφικού και οπτικοακουστικού υλικού (βίντεο). Η καταγραφή είναι σημαντική όχι μόνο για την τελική παρουσίαση του project, αλλά και γιατί αποτελεί βασικό στοιχείο της επιστημονικής μεθόδου και είναι απαραίτητη για την αναπαραγωγή, λειτουργία και βελτίωση του project. Η παρουσίαση και ο διαμοιρασμός της διαδικασίας και των αποτελεσμάτων είναι το τελικό στάδιο κάθε επιστημονικού, ερευνητικού ή κατασκευαστικού έργου.

Με την ολοκλήρωση του 6ου τρίωρου μαθήματος θα πρέπει οι μαθητές να έχουν δημιουργήσει τις ομάδες και να έχουν επιλέξει το πρόβλημα προς επίλυση. Αφού αποφασίσουν το θέμα, μπορούν να κάνουν ένα προσχέδιο της ιδέας τους, το πλάνο τους και να προχωρήσουν στην κατανομή των ρόλων (κατασκευή, έρευνα, αισθητική, καταγραφή, φωτογραφικό υλικό κλπ). Το έγγραφο αυτό μπορεί να έχει ερωτήσεις και ξεχωριστά πλαίσια για να ζωγραφίσουν το logo και το όνομα της ομάδας τους , να γράψουν τα ονόματα τους, τους ρόλους τους, τις κοινότητες εμπλοκής κλπ.

Ευχαριστούμε τους συνεργάτες μας:

Ιωάννη Μαλαμίδη

Σπύρο Πολυχρόνη Λιωνή

Και

Το Τμήμα Έρευνας και Ανάπτυξης, Ελληνογερμανική Αγωγή

Για τη συμβολή τους στη συγγραφή του παρόντος οδηγού

Η γενική σύνταξη και επιμέλεια του οδηγού

έγινε από

τον Μη Κερδοσκοπικό Οργανισμό

«SciCo – Επιστήμη Επικοινωνία»

στο πλαίσιο του προγράμματος "Generation Next"

Αθήνα, Σεπτέμβριος 2024