

Συμμετοχή στον 7^ο Πανελλήνιο Διαγωνισμό Ανοικτών Τεχνολογιών στην Εκπαίδευση (ΕΛΛΑΚ)

Τίτλος: «Έξυπνοι αυτοματισμοί στον Δημόσιο Χώρο»

2^ο Δημοτικό Σχολείο Ευόσμου

Ομάδα Ρομποτικής: Churros (ΣΤ τάξη)

Υπεύθυνη Εκπαιδευτικός: Δώρα Τζιαμπάζη

ΕΙΣΑΓΩΓΗ

Με αφορμή τον διαγωνισμό <https://openedtech.ellak.gr/> σκεφτήκαμε και υλοποιήσαμε ιδέες που κάνουν τον δημόσιο χώρο έξυπνο, βιώσιμο, ευχάριστο, ασφαλή.

Το έργο μας περιλαμβάνει τα projects:

- Έξυπνο υπόστεγο και συλλογή νερού βροχής για δημόσια τουαλέτα
- Έξυπνος ανεμιστήρας
- Πυρασφάλεια
- Μουσικός διάδρομος - πιάνο

Βίντεο Παρουσίασης από την ομάδα: <https://youtu.be/bRXesyvyswI?si=9ubvV-g8wS6SaP8F>

Υλικά: Χρησιμοποιήσαμε τα πακέτα ρομποτικής S1 που στάλθηκαν από το υπουργείο, καθώς υπήρχε διαθεσιμότητα και έχουν και Arduino-based μικροελεγκτές. Εκτός από τα περιεχόμενα των πακέτων S1 χρειαστήκαμε επιπλέον έναν liquid level sensor, έναν flame sensor και καλώδια jumper wires. Για τη σύνδεση των τελευταίων δυο αισθητήρων αξιοποιήσαμε τα RJ11 extension units που υπάρχουν στο πακέτο για τους σερβοκινητήρες. Επίσης, τα μέλη της ομάδας χρησιμοποίησαν χαρτόκουτες, χαρτόνια και υλικά χειροτεχνίας για τις κατασκευές.

Πλαίσιο: Η ομάδα, αποτελούμενη από 4 κορίτσια, συστάθηκε στα τέλη Ιανουαρίου. Ο διαγωνισμός έδωσε έμπνευση στη θεματική των projects. Επίσης, οι συμμετοχές μας στο [Φεστιβάλ Εκπαιδευτικής Ρομποτικής και Τεχνολογίας: Προωθώντας την καινοτομία 2025](#) και στο [17^ο Μαθητικό Συνέδριο Πληροφορικής](#), εκτός από υπέροχες εμπειρίες, λειτούργησαν και ως «σκαλοπάτια» στον χρονοπρογραμματισμό μας. Θέσαμε ως στόχο να ολοκληρώσουμε και να παρουσιάσουμε κάποια έργα και μετά τη συμμετοχή μας τα εξελίξαμε και προχωρήσαμε στα επόμενα.

Έξυπνο υπόστεγο

Ο αυτοματισμός αυτός παρουσιάστηκε στο 17^ο Μαθητικό Συνέδριο Πληροφορικής από τα μέλη της ομάδας. Μπορείτε να δείτε την παρουσίαση [εδώ](https://docs.google.com/presentation/d/1b6wvV_ChIAj5bGumdMvQ_LpmBbx7QOixGRSLKSqDb9M/edit?usp=drive_link):

https://docs.google.com/presentation/d/1b6wvV_ChIAj5bGumdMvQ_LpmBbx7QOixGRSLKSqDb9M/edit?usp=drive_link

Λειτουργία: Έξυπνο στέγαστρο για περίπτωση βροχής. Ένα στέγαστρο μόνιμο μπορεί να μην είναι πάντα επιθυμητό, καθώς μπορεί να αποκρύπτει τον ουρανό/ήλιο. Το έξυπνο στέγαστρο είναι προσαρτημένο σε σερβοκινητήρα και «ξεδιπλώνεται» (αλλάζει γωνία) μόλις ο αισθητήρας στάθμης υγρού δώσει μια τιμή που θα ενημερώσει το πρόγραμμα ότι βρέχει.

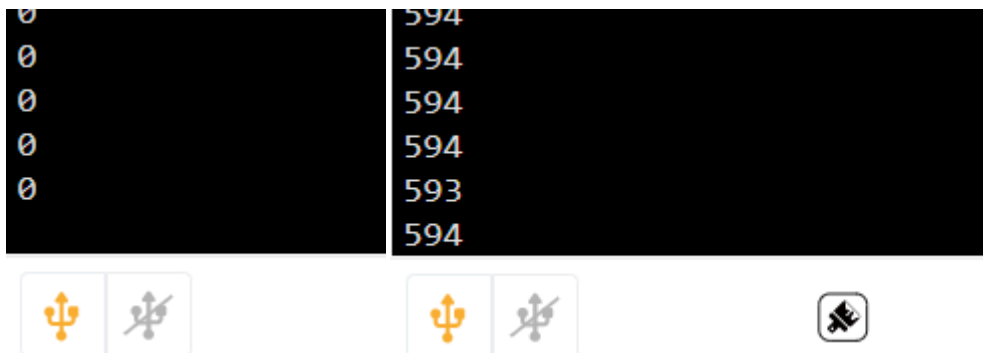
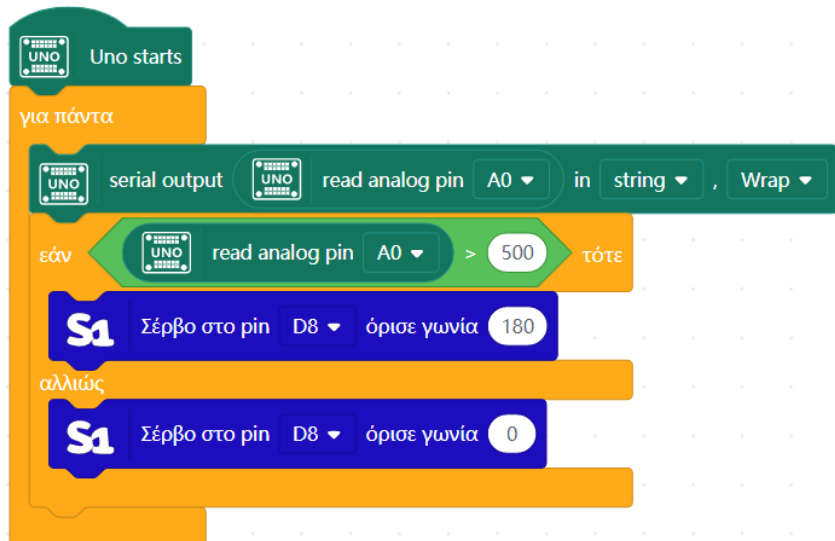
Υλικά: S1, microcontroller ARD:icon, 2 καλώδια RJ11, αισθητήρας στάθμης υγρού (liquid level sensor), servo motor, δυο επεκτάσεις για σύνδεση αισθητήρα και κινητήρα (RJ11 extension unit), jumper wires





Συνδεσμολογία: Συνδέουμε τον αισθητήρα στάθμης υγρού με jumper wires (female to female) στην επέκταση και την επέκταση με καλώδιο RJ11 σε μια αναλογική θύρα (εδώ στην A0). Συνδέουμε τον σερβοκινητήρα μέσω της επέκτασης σε μια ψηφιακή θύρα (εδώ D8).

Το πρόγραμμα:



Στην πρώτη εικόνα βλέπουμε ότι ο αισθητήρας είναι στεγνός, ενώ στη δεύτερη έχει νερό. Οι εμφανίσεις είναι από το serial output, καθώς υπάρχει στον βρόχο η αντίστοιχη εντολή από το πρόσθετο Arduino.

Περισσότερες πληροφορίες για τη βιωσιμότητα της λύσης αυτής:

- Το υπόστεγο ανοίγει μόνο όταν χρειάζεται. Έτσι:
 - Προστατεύει από τη βροχή, γίνεται καταφύγιο
 - Δεν αποκρύπτει τον ήλιο και τον ουρανό όταν έχει αίθριο καιρό
- Καλαίσθητος σχεδιασμός, χαρούμενα χρώματα και σχέδια
- Χρήσιμο για ανοιχτό χώρο όπως παιδική χαρά, πλατεία

Πρόσθετη λειτουργία ως βιώσιμη λύση:

- Όταν το υπόστεγο είναι ανοιχτό (άρα βρέχει) απελευθερώνει το πάνω μέρος μιας δεξαμενής που συλλέγει βρόχινο νερό για τις δημόσιες τουαλέτες
- Όταν είναι κλειστό λειτουργεί ως καπάκι της δεξαμενής για να επιβραδύνει την εξάτμιση
- Αυτοματισμός που μπορεί να έχει και οικιακή χρήση→ Προστασία μπουγάδας

Στις παρακάτω φωτογραφίες τα κορίτσια δοκιμάζουν το ηλεκτρομηχανικό και το κατασκευαστικό κομμάτι.



Έξυπνος ανεμιστήρας

Ο αυτοματισμός αυτός παρουσιάστηκε στην αρχική απλή του μορφή στο 17^ο Μαθητικό Συνέδριο Πληροφορικής στο NOESIS <https://youtu.be/IRixIs8aKpY?si=X2tBVcPl688wyosV> και στη συνέχεια ο κώδικας εμπλουτίστηκε.

Λειτουργία: Ο έξυπνος ανεμιστήρας προστατεύει τις ευπαθείς ομάδες τις ημέρες υψηλού κινδύνου, προσφέρει διάλειμμα δροσιάς και κλείνει όταν δεν υπάρχει ανάγκη για εξοικονόμηση ενέργειας. Δεν χρειάζεται χειροκίνητο έλεγχο, ενώ το όριο μπορεί να αλλάξει στον κώδικα.

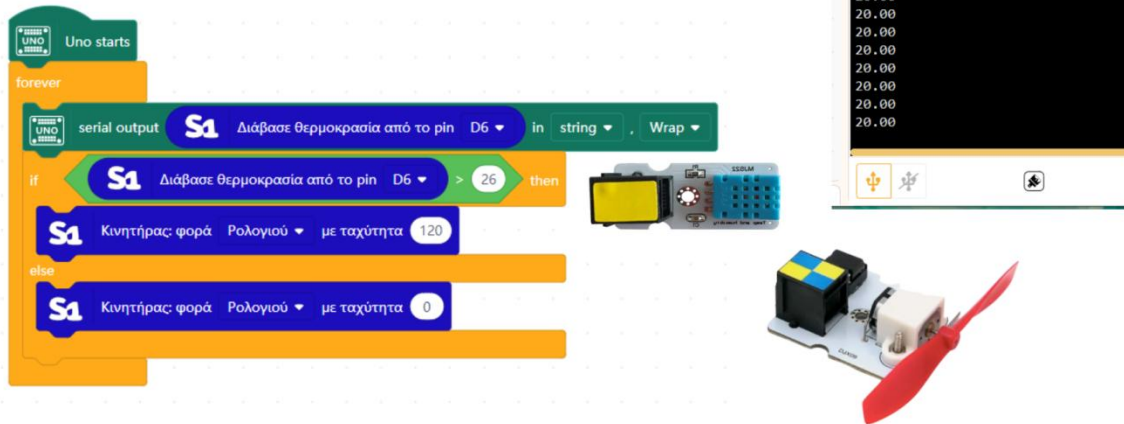
Σημείωση: Για λόγους επίδειξης ανεβάζουμε τη θερμοκρασία τρίβοντας τον αισθητήρα ή με ένα σεσουάρ. Την κατεβάζουμε με παγοκύστη, με κρύο αέρα ή περιμένοντας λίγο.

Υλικά: Ard:icon μικροελεγκτής, αισθητήρας θερμοκρασίας-υγρασίας, fan motor, 2 Καλώδια RJ11

Συνδεσμολογία: Συνδεουμε με τα γαλάζια καλώδια τον αισθητήρα θερμοκρασίας/υγρασίας σε μια ψηφιακή θύρα και τον βηματικό κινητήρα (fan motor) στη θύρα D3/4.

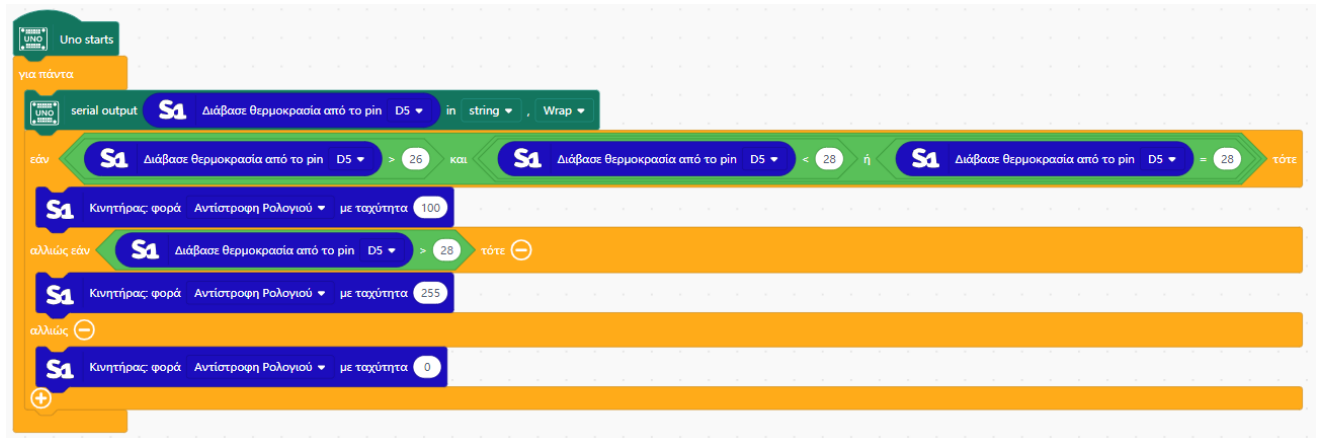
1^η εκδοχή: Απλή δομή επιλογής εάν-αλλιώς με απλό τελεστή στη συνθήκη

Κώδικας σε Mind+ με πρόσθετο για το S1



2^η εκδοχή: Όπως υλοποιήθηκε τελικά. Προχωρήσαμε σε πιο περίπλοκη δομή επιλογής (εάν, αλλιώς εάν, αλλιώς), κατηγοριοποίηση διαβάθμιση με thresholds, πιο περίπλοκη συνθήκη και ανάλυση σε ενδεχόμενα με AND/OR στους τελεστές.

Οι ανεμιστήρες έχουν διαβαθμισμένη ένταση/σκάλα με ταχύτητες, καθώς οι ανάγκες δροσιάς επίσης αυξομειώνονται. Έτσι θα μπορούσαμε να δώσουμε την εντολή στον ανεμιστήρα όχι απλά να τεθεί σε λειτουργία, αλλά να περιστραφεί με μικρότερη ή μεγαλύτερη ταχύτητα. Διακρίναμε 3 περιπτώσεις με τις τιμές να είναι ενδεικτικές:



Εάν η $26 > \text{θερμοκρασία} \geq 28$ τότε

ο ανεμιστήρας περιστρέφεται με μικρή ταχύτητα

αλλιώς εάν $\text{θερμοκρασία} > 28$ τότε

ο ανεμιστήρας περιστρέφεται με τη μέγιστη ταχύτητα

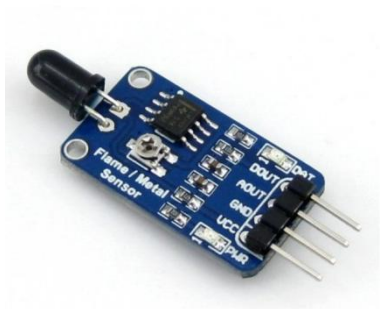
αλλιώς (δηλαδή το ενδεχόμενο που έμεινε είναι $\text{θερμοκρασία} \leq 26$) τότε

ο ανεμιστήρας παραμένει κλειστός

Πυρασφάλεια

Λειτουργία: Αν ο αισθητήρας φωτιάς ή αισθητήρας φλόγας (flame sensor) εντοπίσει φλόγα σε κοντινή απόσταση τότε θα ηχήσει ο βομβητής και θα αναβοσβήνει το λαμπάκι. Σε μια εναλλακτική εκδοχή, αφήσαμε το λαμπάκι να φωτοβολεί σταθερά αντί να αναβοσβήνει.

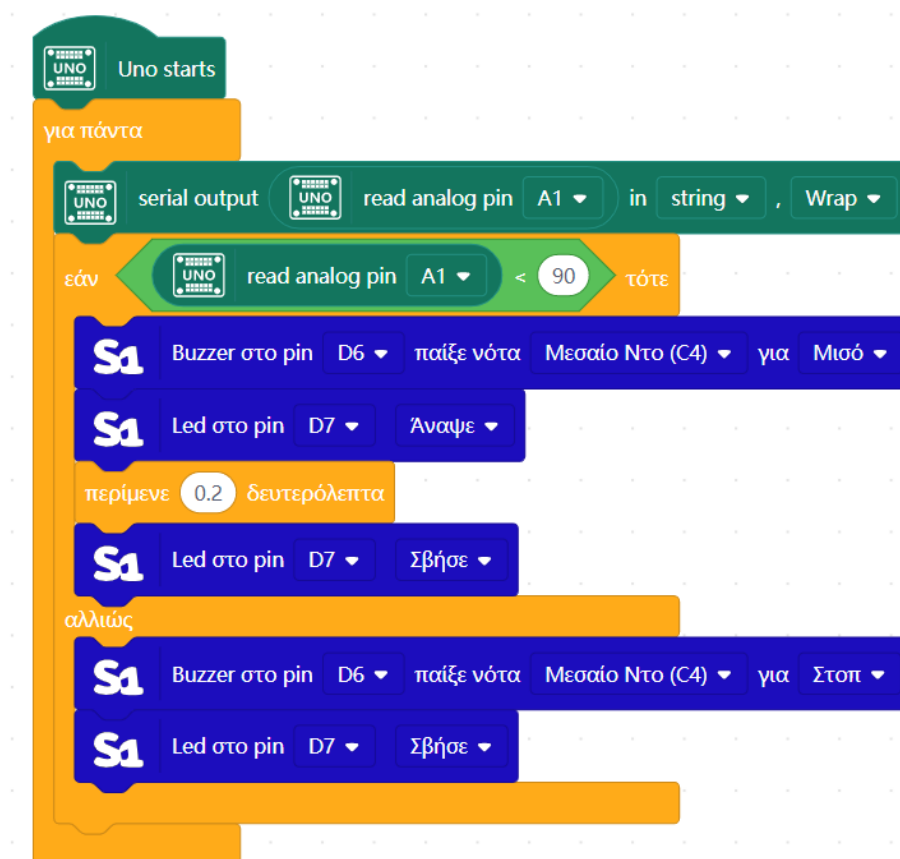
Υλικά: Microcontroller ARD:icon, αισθητήρας φλόγας, extension S1 (ν,γ,ς στις ακίδες), 3 καλώδια RJ11, buzzer, κόκκινο led.



Συνδεσμολογία: Σε δυο ψηφιακές θύρες (εδώ D6,D7) συνδέουμε τον βομβητή και το λαμπάκι. Σε μια αναλογική θύρα (εδώ A1) συνδέουμε τον αισθητήρα φλόγας με τη βοήθεια της επέκτασης που το κουτί διαθέτει για τον σερβοκινητήρα. Συνδέουμε την ακίδα S της επέκτασης στην ακίδα A(analog) του αισθητήρα. Ο αισθητήρας συνοδεύεται από καλωδιάκια κατά την αγορά, διαφορετικά χρησιμοποιούμε jumper wires.

Σημείωση: Ο αισθητήρας αυτός λειτουργεί και ως ψηφιακός με το D(digital) από τις ακίδες του.

Πρόγραμμα:



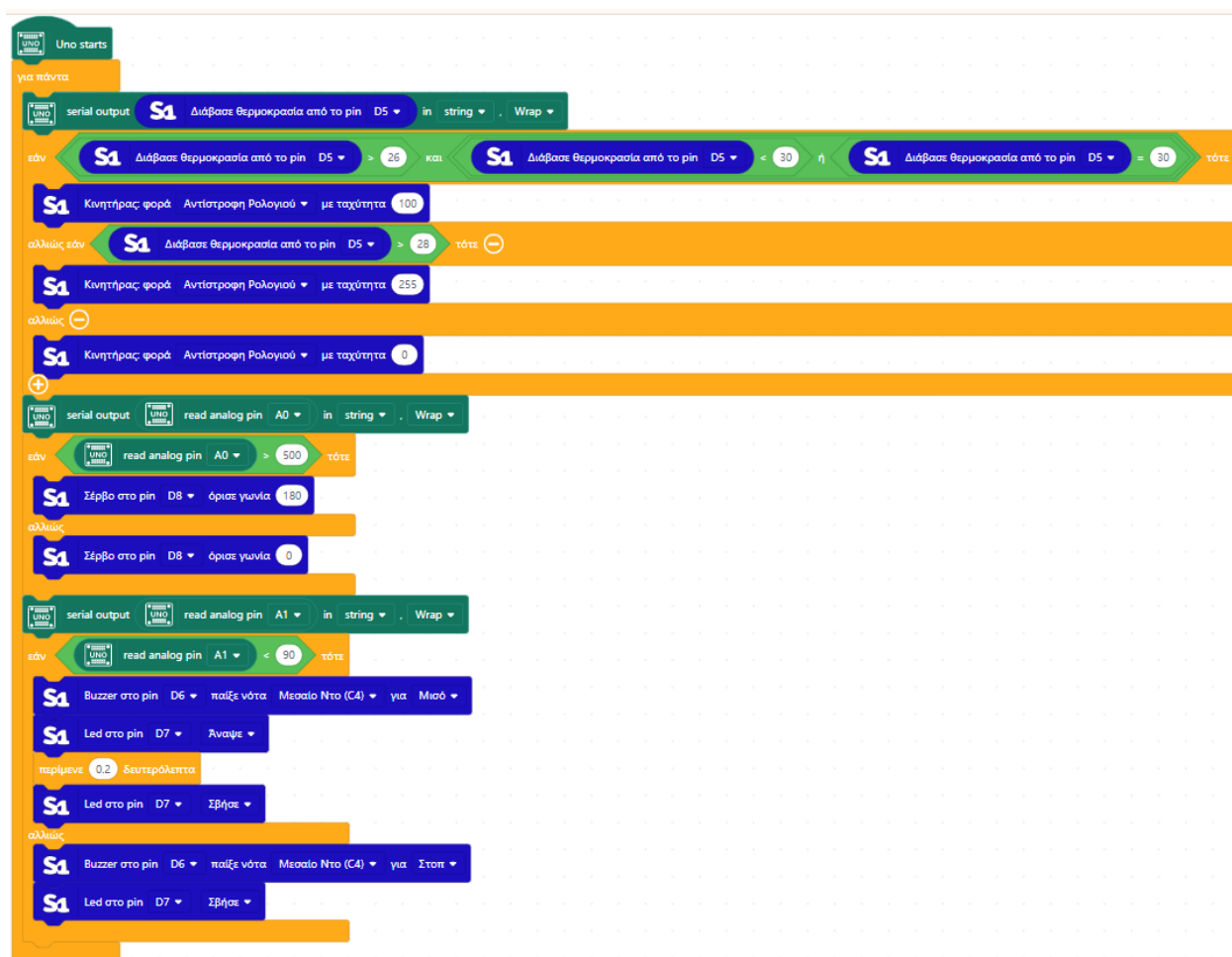
Σχετικά με τη συνθήκη: Πώς θα ξέρουμε πότε ο αισθητήρας πιάνει τη φλόγα;

Παρακολουθούμε με την εντολή serial output πώς αλλάζουν οι τιμές όταν πλησιάζουμε και απομακρύνουμε μια πηγή φλόγας, π.χ. ένα κεράκι (θέλει προσοχή). Όταν βρούμε ένα σημείο πάνω ή κάτω από το οποίο έχουμε φλόγα κοντά, φτιάχνουμε τη συνθήκη. Κατά τον έλεγχο φάνηκε ότι όσο πιο κοντά φέρναμε τη φλόγα, τόσο μίκραιναν οι τιμές. Επομένως στον τελεστή είπαμε ότι κάτω από αυτό το όριο (το 90) υπάρχει φωτιά (σε μια λογική απόσταση).

Σημείωση για συμπερίληψη: Ο συναγερμός αυτός έχει οπτική ένδειξη και ηχητική ειδοποίηση για να συμπεριλάβει και άτομα με προβλήματα ακοής/όρασης. Υπάρχει κ παιδί στο σχολείο με βαρηκοΐα που δεν άκουγε τον βομβητή.

Συνολικό πρόγραμμα (3 ΣΕ 1)

Στα έργα προγραμματισμού και ρομποτικής «συμμαζεύουμε» ότι μπορούμε σε γενικότερα σχήματα. Οι τρεις αυτοί αυτοματισμοί (κλιματισμός, υπόστεγο, πυρασφάλεια) ελέγχονται όλοι από μια πλακέτα που έχει το πρόγραμμα:



Εδώ είναι ο συγκεντρωτικός κώδικας, αλλά παραπάνω γίνεται διαχωρισμός σε ξεχωριστές και περιγραφές και ξεχωριστά προγράμματα για διδακτικό σκοπό ή/και σε περίπτωση που αναπαραχθούν μεμονωμένα σε άλλο πλαίσιο.

Διάδρομος Πιάνο

Λειτουργία: Η ομάδα πρότεινε ένα διασκεδαστικό διαδραστικό μονοπάτι, κατά το πρότυπο ενός αντίστοιχου που έχουν δει σε πόλη του εξωτερικού. Όταν ο περαστικός βαδίζει παίζει με το βήμα του μια μουσική κλίμακα. Κάθε αισθητήρας/κουμπί αφής αντιστοιχεί σε μια νότα. Εμείς επιλέξαμε την κλίμακα Ντο (μια οκτάβα).

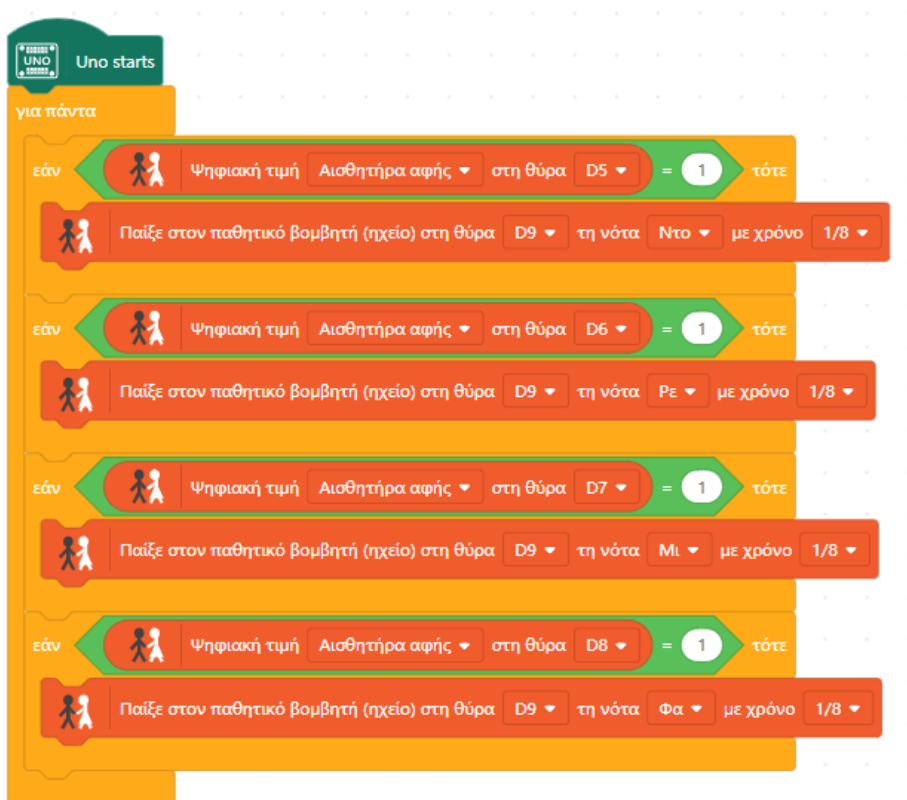
Υλικά: Δυο Microcontrollers ARD:icon, 2 buzzers, 8 αισθητήρες αφής (touch switches), 10 καλώδια RJ11

Σημείωση: Το πρότζεκτ υλοποιείται και με άλλη συνδεσμολογία και άλλο υλισμικό, ωστόσο εμείς τα αξιοποιήσαμε καθώς υπήρχε διαθεσιμότητα στο σχολείο και εμπειρία της ομάδας στο S1 και το Mind+.

Συνδεσμολογία: Σε κάθε microcontroller συνδέουμε 4 αισθητήρες αφής με τα γαλάζια καλώδια (D5, D6, D7, D8) και ένα Buzzer (D9).

Πρόγραμμα:

Στον πρώτο μικροελεγκτή φορτώνουμε το ακόλουθο πρόγραμμα που παίζει τις νότες Ντο-ρε-μι-φα.



Σχετικά με τον τελεστή: Θα μπορούσαμε, αντί για τον πράσινο τελεστή με το πορτοκαλί πλακίδιο δίπλα στο εάν, να χρησιμοποιήσουμε αυτό το πρόσθετο που έχει έτοιμο εξάγωνο πλακίδιο:



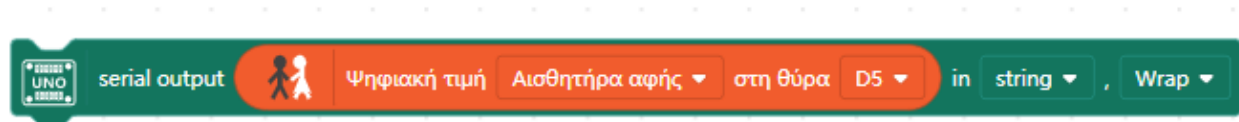
Στην «πορτοκαλί» επέκταση δεν υπήρχε κάτι τέτοιο για το κουμπί αφής όπως υπήρχε για τον μαγνητικό διακόπτη για παράδειγμα το εξάγωνο πλακίδιο «εντόπισε μαγνητικό πεδίο ο αισθητήρας στη θύρα D5;».

Επομένως, θα αναλύσουμε λίγο τι συμβαίνει.

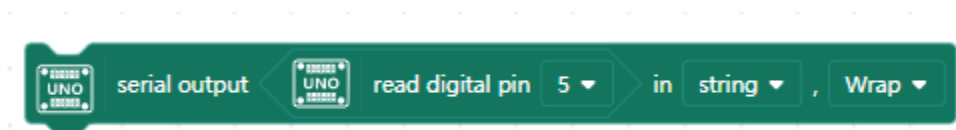
Ο αισθητήρας αφής ή κουμπί αφής είναι ψηφιακός και δίνει δυο τιμές, 0 και 1. Όταν τον αγγίζουμε παίρνει την τιμή 1. Γι' αυτόν τον λόγο έχει μπει αυτή η συνθήκη. Επομένως πρέπει να βάλουμε αυτόν τον γενικό τελεστή, αφού βρούμε σε ποια κατάσταση αντιστοιχεί το 0 και το 1 (δεν είναι πάντα το αναμενόμενο). Πώς το κάνουμε αυτό;

A) Με δοκιμή. Δοκιμάζουμε το 0 και το 1 εναλλάξ για μια νότα και βλέπουμε τι γίνεται.

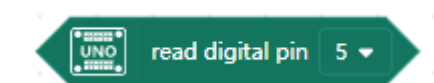
B) Ανοίγουμε το serial output και βλέπουμε ποια τιμή δείχνει όταν ακουμπάμε τον αισθητήρα και ποια όταν τον αφήνουμε. Για να εμφανιστούν εκεί τα 0 ή 1 πρέπει να σύρουμε στον κώδικα, κάτω από το «για πάντα» το παρακάτω πλακίδιο



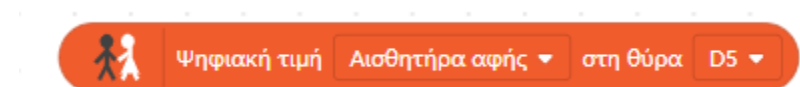
Ή εναλλακτικά



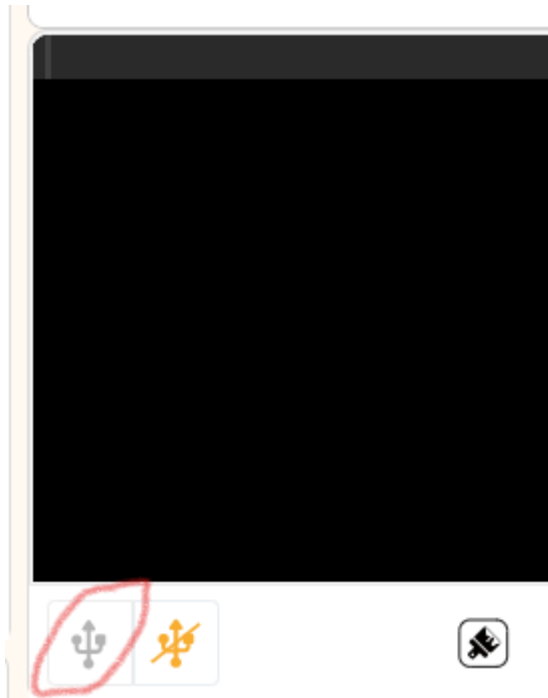
Καθώς σε πιο γενική μορφή ο κώδικας θα μπορούσε εντός του τελεστή να έχει το



αντί για το



Αφού, λοιπόν, βάλουμε αυτή την εντολή σε μια επαναληπτική δομή «για πάντα» και φορτώσουμε τη λούπα στο UNO, μένει να πατήσουμε αυτό το κουμπί (το κυκλωμένο στην ακόλουθη εικόνα) για να ανοίξουμε τη σειριακή οθόνη:



Θα βλέπουμε εδώ τις τιμές του αισθητήρα Live! Έτσι πειραματιζόμαστε και ελέγχουμε και άλλους αισθητήρες, κυρίως αναλογικούς όταν θέλουμε να βρούμε μια τιμή που θα γίνει όριο ή, όπως λέμε, κατώφλι (threshold).

Στον δεύτερο μικροελεγκτή απέμειναν οι υπόλοιπες 4 νότες: Σολ-Λα-Σι-Ντο.

Παρατηρούμε ότι οι επεκτάσεις δεν έχουν την ψηλότερη Ντο με την οποία κλείνει η οκτάβα.

Θα μπορούσαμε να μην συμπεριλάβουμε την 8^η νότα, αλλά έχουμε μια καλή ευκαιρία να παρατηρήσουμε και να επιλύσουμε το πρόβλημα, μεταβαίνοντας από τα πλακίδια στις γραπτές εντολές.

Σχετικά με την ήδη έτοιμη τετράδα (ντο-ρε-μι-φα) βλέπουμε δίπλα τον Auto Generated code

```
/*!  
 * MindPlus  
 * uno  
 *  
 */  
  
// Main program start  
void setup() {  
  pinMode(5, INPUT);  
  pinMode(9, OUTPUT);  
  pinMode(6, INPUT);  
}
```

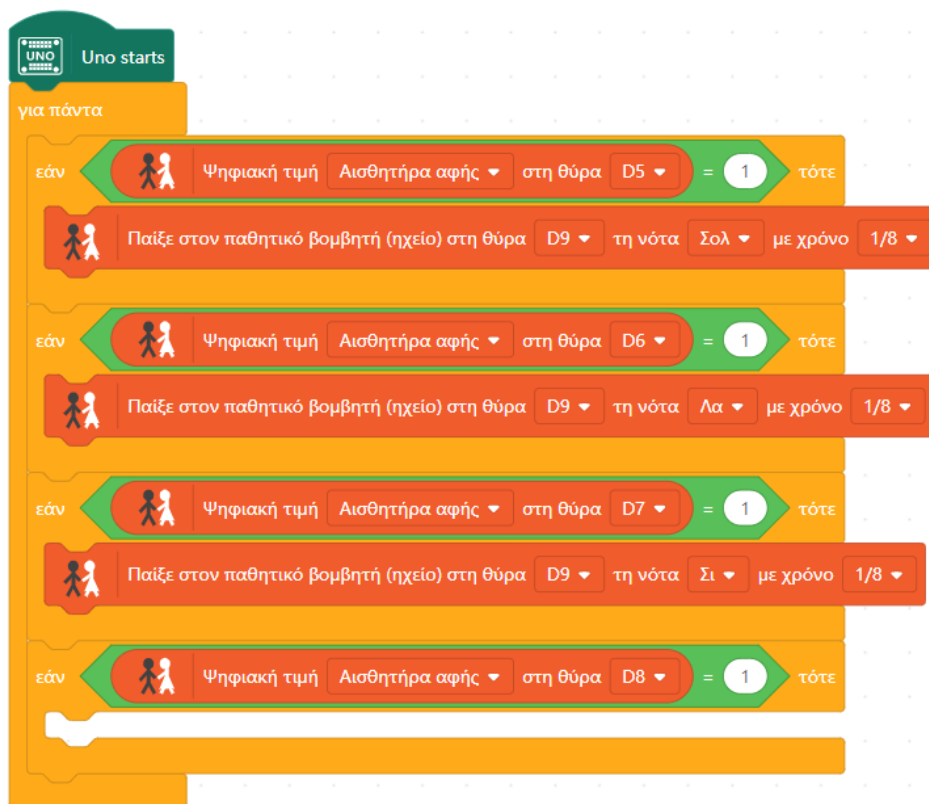
```

    pinMode(7, INPUT);
    pinMode(8, INPUT);
}
void loop() {
    if ((digitalRead(5)==1)) {
        tone(9, 262, 125);
    }
    if ((digitalRead(6)==1)) {
        tone(9, 294, 125);
    }
    if ((digitalRead(7)==1)) {
        tone(9, 330, 125);
    }
    if ((digitalRead(8)==1)) {
        tone(9, 349, 125);
    }
}

```

Σχετικά με τη δεύτερη τετράδα (σολ-λα-σι-ντο)

Ο κώδικας είναι ο παρακάτω, ωστόσο, το πρόσθετο δεν έχει την τελευταία νότα.



Θα στραφούμε στον κώδικα Arduino C που μας δίνει περισσότερες δυνατότητες. Πάνω δεξιά στο παράθυρο που εμφανίζεται ο κώδικας (Auto generated), τον αντιγράφουμε και τον παρατηρούμε:

```

/*!
 * MindPlus
 * uno
 *
 */

// Main program start
void setup() {
  pinMode(5, INPUT);
  pinMode(9, OUTPUT);
  pinMode(6, INPUT);
  pinMode(7, INPUT);
  pinMode(8, INPUT);
}
void loop() {
  if ((digitalRead(5)==1)) {
    tone(9, 392, 125);
  }
  if ((digitalRead(6)==1)) {
    tone(9, 440, 125);
  }
  if ((digitalRead(7)==1)) {
    tone(9, 494, 125);
  }
  if ((digitalRead(8)==1)) {
  }
}

```

πατάμε το manual, κάνουμε επικόλληση και παρεμβαίνουμε αντιγράφοντας και τροποποιώντας τμήμα του κώδικα που είναι για τις άλλες νότες. Παρατηρούμε ότι στις παρενθέσεις, το πρώτο στοιχείο είναι η θύρα του βομβητή, το τρίτο στοιχείο είναι η αξία (διάρκεια νότας), άρα αυτό που απομένει, το δεύτερο στοιχείο (είναι αυτό που αλλάζει, ενώ τα άλλα είναι κοινά), αντιστοιχεί στη συχνότητα της νότας. Η νότα που μας ενδιαφέρει (C5) είναι στα 523 Hz

Άρα η τελευταία εντολή διαμορφώνεται ως εξής κάτω από το if

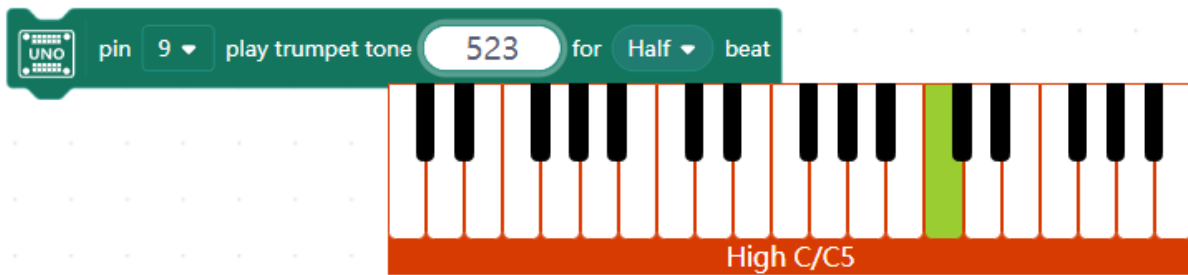
```

    if ((digitalRead(8)==1)) {
      tone(9, 523, 125);
    }
}

```

Πώς βρήκαμε τον αριθμό που αντιστοιχεί σε αυτή τη νότα;

Πηγαίνοντας στο πρόσθετο Arduino του Mind+, τραβώντας την εντολή



και παρατηρώντας το πιανάκι. Αν η νότα Λα (A4) είναι στα 440 όπως φαίνεται σε προηγούμενη σειρά του κώδικα, ανεβαίνοντας στην επόμενη Ντο βλέπουμε την τιμή 523.

Πληκτρολογούμε αυτή την τιμή και κάνουμε Upload τον κώδικα αυτόν και όχι κώδικα με πλακίδια.

```

/*!
 * MindPlus
 * uno
 *
 */

// Main program start
void setup() {
  pinMode(5, INPUT);
  pinMode(9, OUTPUT);
  pinMode(6, INPUT);
  pinMode(7, INPUT);
  pinMode(8, INPUT);
}
void loop() {
  if ((digitalRead(5)==1)) {
    tone(9, 392, 125);
  }
  if ((digitalRead(6)==1)) {
    tone(9, 440, 125);
  }
  if ((digitalRead(7)==1)) {
    tone(9, 494, 125);
  }
  if ((digitalRead(8)==1)) {
    tone(9, 523, 125);
  }
}

```

```
Auto Generated Manual Editing

8 // Main program start
9 void setup() {
10   pinMode(5, INPUT);
11   pinMode(9, OUTPUT);
12   pinMode(6, INPUT);
13   pinMode(7, INPUT);
14   pinMode(8, INPUT);
15 }
16 void loop() {
17   if ((digitalRead(5)==1)) {
18     tone(9, 392, 125);
19   }
20   if ((digitalRead(6)==1)) {
21     tone(9, 440, 125);
22   }
23   if ((digitalRead(7)==1)) {
24     tone(9, 494, 125);
25   }
26   if ((digitalRead(8)==1)) {
27     tone(9, 523, 125);
28   }
29 }
30 }
```

Τώρα έχουμε δυο μικροελεγκτές στους οποίους έχουμε φορτώσει στον καθένα από ένα πρόγραμμα. Βάζοντάς τους τον έναν δίπλα στον άλλον και τις δυο ομάδες με τους αισθητήρες αφής (τα «πλήκτρα») στη σειρά, έχουμε έναν διάδρομο στον οποίο οι πεζοί βαδίζουν και παίζουν μουσική.

Περισσότερα με το R2

Εκτός από το S1 πειραματιστήκαμε με το όχημα R2. Στον δημόσιο χώρο θα μπορούσαν να υπάρχουν και ρομποτικά οχήματα για μεταφορά και διασκέδαση για μικρούς και μεγάλους, κάτι σαν εναλλακτικά «κουρσάκια». Με τα R2 δοκιμάσαμε να χορέψουμε [zumba](#), με κάθε μέλος της ομάδας να τηλεχειρίζεται ένα ρομπότ με το mobile app. Επίσης προγραμματίσαμε στο Mind+ [μια χορογραφία](#) με βάση ένα μουσικό κομμάτι. Μπορούμε να φανταστούμε τα R2 είτε σαν οχήματα που μπορούν να δεχτούν αναβάτη είτε σαν μικρά ρομπότ με τα οποία ο κόσμος θα μπορεί να παίζει και να μαθαίνει, να τα τηλεχειρίζεται/προγραμματίζει/φορτώνει ή «πειράζει» έτοιμα προγράμματα.

Πηγές για εκπαιδευτικούς

[Το σεμινάριο eTwinning για το S1](#)

Πρόσθετα/extensions:

<https://ale3andro.gr/blog/>

<https://portal.stem.edu.gr/>