

# **ΔΙΔΑΚΤΙΚΟ ΣΕΝΑΡΙΟ ΓΥΜΝΑΣΙΟΥ**

## **ΜΕ ΧΡΗΣΗ ARDUINO**

Στα πλαίσια του επιμορφωτικού προγράμματος ΟΗ4ΗΣ  
«Προγραμματισμός και εφαρμογές μικροελεγκτών στην διδακτική πράξη. Αξιοποίηση της  
αναπτυξιακής πλατφόρμας του Arduino, ως εκπαιδευτικό εργαλείο»

**ΦΩΤΙΑΔΟΥ ΓΛΥΚΕΡΙΑ**

Καθηγήτρια Πληροφορικής (ΠΕ19)

1<sup>ο</sup> Γυμνάσιο Κομοτηνής

# ΔΙΔΑΚΤΙΚΟ ΣΕΝΑΡΙΟ

---

## 1. ΤΙΤΛΟΣ ΣΕΝΑΡΙΟΥ

Προγραμματίζοντας το Arduino

## 2. ΕΚΤΙΜΩΜΕΝΗ ΔΙΑΡΚΕΙΑ ΤΟΥ ΣΕΝΑΡΙΟΥ

Προβλέπεται να διαρκέσει 2 διδακτικές ώρες στο σχολείο και απαιτεί από τον μαθητή και 1 ώρα επιπλέον εργασία στο σπίτι..

## 3. ΕΜΠΛΕΚΟΜΕΝΕΣ ΓΝΩΣΤΙΚΕΣ ΠΕΡΙΟΧΕΣ

Εμπλέκονται η έννοια του προγράμματος, του προγραμματισμού και των γλωσσών προγραμματισμού, που αποτελούν αντικείμενο διδασκαλίας της πληροφορικής καθώς και του μικροελεγκτή arduino.

## 4. ΕΝΤΑΞΗ ΣΤΟ ΠΡΟΓΡΑΜΜΑ ΣΠΟΥΔΩΝ

Το παρόν σενάριο εντάσσεται στο μάθημα «Πληροφορική» της Γ Γυμνασίου, στην Ενότητα1 «Γνωρίζω τον υπολογιστή ως ενιαίο σύστημα–Προγραμματισμός», στο Κεφάλαιο 2: «Ο Προγραμματισμός στην Πράξη».

## 5. ΠΡΟΑΠΑΙΤΟΥΜΕΝΕΣ ΓΝΩΣΕΙΣ ΤΩΝ ΜΑΘΗΤΩΝ

Το σενάριο υλοποιείται στο τέλος της σχολικής χρονιάς της Γ Τάξης Γυμνασίου. Οι μαθητές γνωρίζουν τις έννοιες του αλγορίθμου και του προγράμματος, που έχουν διδαχθεί στην αρχή της χρονιάς. **Ξέρουν τι είναι γλώσσες προγραμματισμού, πηγαίος και εκτελέσιμος κώδικας καθώς και την διαδικασία της μεταγλώττισης.**

Έχουν ήδη διδαχθεί και χρησιμοποιήσει μία γλώσσα προγραμματισμού οπότε γνωρίζουν **τι είναι εντολή, τις βασικές δομές προγραμματισμού, την έννοια της μεταβλητής.**

Το σενάριο προβλέπεται να υλοποιηθεί στο 3<sup>ο</sup> τρίμηνο της Γ Τάξης Γυμνασίου και αφού οι μαθητές **έχουν διδαχθεί την γλώσσα προγραμματισμού scratch**. Οι μαθητές δηλαδή **γνωρίζουν πώς να φτιάχνουν απλά προγράμματα** σε μια γλώσσα προγραμματισμού κι έχουν αρχίσει να αποκτούν αλγοριθμικό τρόπο σκέψης και ικανότητες επίλυσης προβλήματος.

## 6. ΣΚΟΠΟΙ ΚΑΙ ΣΤΟΧΟΙ ΤΟΥ ΔΙΔΑΚΤΙΚΟΥ ΣΕΝΑΡΙΟΥ

Σκοπός του διδακτικού σεναρίου είναι να χρησιμοποιήσουν οι μαθητές μια γλώσσα προγραμματισμού που δεν είναι οπτική, σε ένα περιβάλλον προγραμματισμού που είναι κοντά σε **αυτό που χρησιμοποιούν οι επαγγελματίες προγραμματιστές** γνωρίζοντας παράλληλα και την «τεχνολογία» arduino για την οποία τόσος λόγος γίνεται τα τελευταία χρόνια.

Ειδικότερα:

### ΓΝΩΣΕΙΣ

- Να περιγράφουν τι είναι το arduino
- Να περιγράφουν τι είναι breadboard
- Να απαριθμούν στοιχεία που μπορούν να συνδεθούν στο arduino
- Να αναφέρουν ενδεικτικές χρήσεις/projects του arduino
- Να κατονομάζουν την γλώσσα προγραμματισμού του arduino
- Να κατανοούν την βασική δομή προγράμματος του arduino
- Να κατανοούν κάποιες βασικές εντολές του arduino
- Να διαβάζουν ένα απλό προγραμματάκι για arduino
- Να συγκρίνουν μια οπτική γλώσσα προγραμματισμού (όπως η scratch) με μια μη οπτική (όπως η C++)
- Να εξηγούν την διαδικασία μεταγλώττισης και ανεβάσματος του προγράμματος στο arduino

### ΔΕΞΙΟΤΗΤΕΣ

- να αναγνωρίζουν την πλακέτα arduino και το breadboard
- να αναγνωρίζουν τα: LED, Piezo Buzzer, αισθητήρα φωτός (LDR)
- να συνδέουν στην πλακέτα arduino LEDs, ηχεία, αισθητήρες
- να συνδέουν το arduino στον υπολογιστή
- να εκκινούν το περιβάλλον προγραμματισμού IDE του arduino
- να μεταγλωττίζουν τον πηγαίο κώδικα
- να αναγνωρίζουν και να διορθώνουν συντακτικά λάθη
- να αναγνωρίζουν και να διορθώνουν λάθη συνδεσμολογίας
- να ανεβάζουν τον εκτελέσιμο κώδικα στο arduino
- να αναγνωρίζουν ένα αρχείο πηγαίου κώδικα arduino
- να κάνουν μικρές τροποποιήσεις σε ένα πρόγραμμα για να πάρουν το επιθυμητό αποτέλεσμα

## ΣΤΑΣΕΙΣ

- να μην θεωρούν τη διαδικασία του προγραμματισμού ως κάτι που είναι πέρα από τις δυνατότητές τους
- να θεωρούν ότι ο καθένας μπορεί να μάθει να προγραμματίζει, αρκεί να μάθει μια γλώσσα προγραμματισμού
- να θεωρούν δυνατή και εύκολη την εκμάθηση μιας γλώσσας προγραμματισμού (υψηλού επιπέδου) γιατί οι γλώσσες προγραμματισμού μοιάζουν με τη φυσική μας γλώσσα
- να θεωρούν τον υπολογιστή μηχανή χωρίς ευφυΐα που απλά εκτελεί τις εντολές που γράφει ο προγραμματιστής σε ένα πρόγραμμα
- να θεωρούν τον υπολογιστή ως ένα εργαλείο για να δημιουργήσουν (π.χ. προγράμματα και αυτοματισμούς) και να μετατραπούν από απλοί χρήστες σε δημιουργούς.

## 7. ΑΠΑΙΤΟΥΜΕΝΗ ΥΛΙΚΟΤΕΧΝΙΚΗ ΥΠΟΔΟΜΗ

Το σενάριο θα υλοποιηθεί στο **εργαστήριο πληροφορικής** του σχολείου. Στους υπολογιστές θα πρέπει να είναι **εγκατεστημένο** το περιβάλλον προγραμματισμού του **arduino IDE (Intergrated Development Environment)**.

Οι υπολογιστές πρέπει να είναι συνδεδεμένοι στο **διαδίκτυο** γιατί είναι απαραίτητη η επίσκεψη συγκεκριμένων διαδικτυακών τόπων. Στο εργαστήριο πρέπει να υπάρχει **βιντεοπροβολέας** για την επίδειξη/επεξήγηση δύσκολων σημείων του μαθήματος.

**Για κάθε ζεύγος μαθητών θα πρέπει να διαθέτουμε μία πλακέτα arduino UNO με breadboard, 2-3 LEDs, 1 Piezzo Buzzer, μία φωτοαντίσταση LDR καθώς και καλώδια και αντιστάσεις για τις συνδέσεις των κυκλωμάτων.**

Θα διανεμηθούν στους μαθητές από τον καθηγητή **Φύλλα Εργασίας** που θα έχουν ετοιμαστεί από αυτόν.

## 8. ΠΕΡΙΓΡΑΦΗ ΤΟΥ ΔΙΔΑΚΤΙΚΟΥ ΣΕΝΑΡΙΟΥ-ΦΥΛΛΑ ΕΡΓΑΣΙΑΣ

Το διδακτικό σενάριο προβλέπεται να υλοποιηθεί σε **2 διδακτικές ώρες** προς το τέλος της σχολικής χρονιάς της Γ Γυμνασίου. Επειδή το μάθημα της πληροφορικής είναι **μονόωρο** στο Γυμνάσιο, δεν υπάρχει η δυνατότητα να αφιερώσουμε περισσότερο χρόνο. Από την άλλη είναι **πολύ δύσκολο για ένα παιδί γυμνασίου** με μόνη εμπειρία προγραμματισμού την οπτική γλώσσα scratch, να καταφέρει μέσα σε 2 ώρες να μάθει

όσα χρειάζονται για να συνδέσει και να προγραμματίσει το arduino και να πειραματιστεί με αυτό.

Για τον λόγο αυτό, τις **εισαγωγικές θεωρητικές γνώσεις** καλούμε τους μαθητές να τις αποκτήσουν **στο σπίτι παρακολουθώντας κάποια video/tutorials** που έχουν αναρτηθεί στο wiki του μαθήματος <http://pliroforikigym.pbworks.com/> . Έτσι μπορούμε να χρησιμοποιήσουμε τις **2 διδακτικές ώρες για πειραματισμό και ανακάλυψη**, στο χώρο του εργαστηρίου, με **φύλλα εργασίας** που απαιτούν **hands on** εργασία από τους μαθητές. **Αντιστρέφουμε δηλαδή τη συνηθισμένη εκπαιδευτική διαδικασία** που θέλει η θεωρία να παραδίνεται στο σχολείο και οι ασκήσεις να υλοποιούνται από τους μαθητές, μόνους στο σπίτι. Η Θεωρία θα κατακτηθεί στο σπίτι και η εξάσκηση θα γίνει σε συνεργασία με τους συμμαθητές, με την επίβλεψη και βοήθεια του καθηγητή στο σχολείο (**Flipped Classroom**).

#### **Στο σπίτι:**

Στα ολιγόλεπτα αυτά video, οι μαθητές θα δουν μια περιγραφή των συστατικών στοιχείων του arduino, τι είναι το breadboard, πως γίνονται οι συνδέσεις, κάποια arduino projects, το περιβάλλον προγραμματισμού IDE. Η συνολική διάρκεια των video δεν ξεπερνάει τα 30 λεπτά.

#### **Στο σχολείο:**

##### **1<sup>η</sup> διδακτική ώρα - Εξοικείωση πειραματισμός**

Δίνουμε σε κάθε ζεύγος μαθητών την πλακέτα arduino με το breadboard στο οποίο είναι ήδη συνδεδεμένο ένα LED και τους μοιράζουμε το φύλλο εργασίας. Κάθε ζευγάρι μαθητών μπορεί να ακολουθήσει το δικό του ρυθμό εργασίας.

**Αρχικά δεν θέλουμε οι μαθητές να έρθουν αντιμέτωποι με προβλήματα συνδεσμολογίας-ηλεκτρονικής, για αυτό κι έχουμε ετοιμάσει εμείς την πλακέτα.** Το φύλλο εργασίας τους καθοδηγεί να βρουν στον υπολογιστή τους και να ανοίξουν ένα πρόγραμμα για το arduino, το οποίο έχουμε ετοιμάσει εμείς. Το πρόγραμμα αναβοσβήνει το LED. Οι μαθητές καλούνται να εφαρμόσουν αυτά που είδαν στο σπίτι στα εκπαιδευτικά video και να μεταγλωττίσουν και να ανεβάσουν το πρόγραμμα στην πλακέτα και **να απαντήσουν στην ερώτηση του τι κάνει αυτό το πρόγραμμα.**

Είναι φυσικό οι μαθητές να αντιμετωπίσουν δυσκολίες την πρώτη φορά που χρησιμοποιούν το περιβάλλον προγραμματισμού του arduino τις οποίες μπορούν να ξεπεράσουν α) με την βοήθεια των συμμαθητών τους β) του καθηγητή τους και γ) ανατρέχοντας ίσως και στα video που είδαν στο σπίτι.

Στη συνέχεια ενημερώνονται από το φύλλο εργασίας ότι η γλώσσα προγραμματισμού στην οποία είναι γραμμένο το πρόγραμμα που βλέπουν και τρέχουν είναι ουσιαστικά η πολύ γνωστή στους προγραμματιστές C++. Οδηγούνται στο διαδίκτυο όπου μπορούν **να δουν μια κατάταξη των δημοφιλέστερων γλωσσών προγραμματισμού** για τον τρέχοντα μήνα και καλούνται να μελετήσουν αυτό τον πίνακα και να απαντήσουν σε κάποιες ερωτήσεις για να διαπιστώσουν και μόνοι τους ότι η C είναι μια πολύ παλιά γλώσσα προγραμματισμού που είναι στις 3 πρώτες θέσεις της κατάταξης όλα αυτά τα χρόνια. Επισημαίνουμε στους μαθητές ότι η C++ είναι μια γλώσσα που χρησιμοποιούν οι επαγγελματίες προγραμματιστές για την ανάπτυξη των διαφόρων εφαρμογών ενώ η scratch που διδάχθηκαν κατά την διάρκεια της σχολικής χρονιάς είναι μια οπτική γλώσσα προγραμματισμού (οι εντολές είναι πλακίδια που τοποθετούνται με σύρε κι άσε και δεν υπάρχει περίπτωση συντακτικού λάθους) που δημιουργήθηκε μόνο για εκπαιδευτικούς σκοπούς. Επιδιώκουμε να τους δημιουργήσουμε έντονο ενδιαφέρον να ασχοληθούν με την γλώσσα αυτή.

Τους καθοδηγούμε μέσω του φύλλου εργασίας να **δημιουργήσουν και στη συνέχεια να διορθώσουν ένα συντακτικό λάθος**. Είναι σημαντικό να τους φέρουμε αντιμέτωπους με αυτό το είδος λαθών γιατί δεν το έχουν συναντήσει καθ' όλη την διάρκεια της σχολικής χρονιάς, λόγω της ενασχόλησης με την scratch.

Στη συνέχεια οι μαθητές **καλούνται να διαβάσουν τον κώδικα** και να **υποθέσουν/αναγνωρίσουν τις εντολές** που ανάβουν και σβήνουν το LED. Νομίζουμε ότι είναι σε θέση να το κάνουν αφού α) έχουν ήδη δει την συμπεριφορά του προγράμματος β) γνωρίζουν ήδη από την scratch ότι οι εντολές εκτελούνται σειριακά γ) οι εντολές του προγράμματος είναι πολύ λίγες δ) στο πρόγραμμα χρησιμοποιείται η μεταβλητή LED για τον αριθμό του ακροδέκτη, οπότε οι εντολές είναι πιο κοντά στην φυσική γλώσσα. Αποφεύγουμε να πούμε εμείς ποιες είναι εντολές και προτιμούμε να τις ανακαλύψουν οι μαθητές μόνοι τους.

Στη συνέχεια οι μαθητές **ενθαρρύνονται να πειραματιστούν με τις εντολές αυτές και τη συνδεσμολογία του LED**. Τους ζητάμε να κάνουν high άλλον ακροδέκτη από αυτό που είναι συνδεδεμένο το LED, να συνδέσουν το LED σε άλλο pin και να τροποποιήσουν το πρόγραμμα έτσι ώστε να δουλεύει για αυτή την περίπτωση. Τέλος καλούνται να **συνδέσουν μόνοι τους ένα δεύτερο LED**, βλέποντας την συνδεσμολογία από μία εικόνα και να προβούν στην τροποποίηση του κώδικα ώστε τα δύο LED να αναβοσβήνουν με κάποιο συγκεκριμένο ρυθμό.

Για τους μαθητές που θα είναι πολύ γρήγοροι και θα έχουν χρόνο για κάτι επιπλέον, έχουμε προβλέψει μια επιπλέον δραστηριότητα. Εδώ καλούνται **να προσθέσουν ένα**

**Piezo Buzzer** με συνδεσμολογία που δίνεται σε εικόνα και να προσθέσουν την εντολή για δημιουργία ήχου.

## 2<sup>η</sup> Διδακτική ώρα – School Bag Alarm

Συνεχίζεται ο πειραματισμός με το arduino. Οι μαθητές θα ασχοληθούν με **κάτι που έχει νόημα για αυτούς**. Καθοδηγούμενοι από ένα φύλλο εργασίας θα κατασκευάσουν και θα προγραμματίσουν έναν **συναγερμό σχολικής τσάντας**. Ο συναγερμός θα ενεργοποιείται μόλις κάποιος ανοίξει το φερμουάρ της τσάντας κι επιχειρήσει να πάρει κάτι μέσα από αυτή.

Ξεκινάμε ζητώντας από τους μαθητές να **σχολιάσουν την πρώτη εμπειρία τους με το arduino την προηγούμενη ώρα**. Σε ένα καταιγισμό ιδεών τους ζητάμε να μας πουν αν γνωρίζουν ή μπορούν να φανταστούν εφαρμογές/αυτοματισμούς στις οποίες θα μπορούσε να χρησιμοποιηθεί το arduino. Τους ενημερώνουμε ότι στο τέλος της ώρας θα έχουν κι αυτοί κατασκευάσει τον πρώτο αυτοματισμό τους. Το **SchoolBagAlarm!!!**

Μοιράζουμε το φύλλο εργασίας και τις πλακέτες με τα απαραίτητα υλικά (LED, Piezo Buzzer, φωτοαντίσταση (LDR)). Αυτή τη φορά **τα εξαρτήματα δεν είναι συνδεδεμένα στην πλακέτα** και η σύνδεση θα γίνει από τους μαθητές.

Στο φύλλο εργασίας υπάρχουν οι απαραίτητες πληροφορίες για την φωτοαντίσταση (τι είναι και πως λειτουργεί) και η συνδεσμολογία των εξαρτημάτων στο arduino.

Δίνεται έτοιμο πρόγραμμα που καλούνται να τρέξουν οι μαθητές και **να καταγράψουν τις μετρήσεις**. Το πρόγραμμα εμφανίζει στη σειριακή τις τιμές που επιστρέφει η φωτοαντίσταση (δειγματοληψία κάθε δευτερόλεπτο). Τους **ενθαρρύνουμε να πειραματιστούν για να καταλάβουν τι τιμές ο LDR στο φως και τι στο σκοτάδι**. Θα τις χρειαστούν, όταν αργότερα θα πρέπει να αποφασίσουν το threshold για την ενεργοποίηση του buzzer.

Στη συνέχεια τους ζητάμε **να γράψουν τον αλγόριθμο σε φυσική γλώσσα** που περιγράφει τις ενέργειες που θέλουμε να κάνει το πρόγραμμά μας. Οι μαθητές δεν γνωρίζουν την γλώσσα προγραμματισμού του arduino (C++). Άρα δεν είναι σε να μετατρέψουν τον αλγόριθμο σε πρόγραμμα. **Δεν θέλουμε όμως και να τους δώσουμε έτοιμη την εφαρμογή**. Διαλέγουμε έναν ενδιάμεσο τρόπο που περιέχει **ανακάλυψη και πειραματισμό**.

Τους δίνουμε τις εντολές που θα χρειαστούν και ζητάμε να τις αντιστοιχίσουν με την **ενέργεια που πιστεύουν ότι κάνει η κάθε μία από αυτές** (δίνουμε τις ενέργειες ανακατεμένες). Έτσι μπαίνουν στην διαδικασία να διαβάσουν τις διαθέσιμες εντολές και

να σκεφτούν τι μπορεί να κάνει η κάθε μία. Έχοντας τώρα μάθει τι κάνει η κάθε εντολή από αυτές που θα χρησιμοποιήσουν, τους **ζητάμε να μετατρέψουν τον αλγόριθμό τους στη γλώσσα προγραμματισμού του arduino.**

Το μόνο που μένει πια είναι να γράψουν τον κώδικα στο IDE και να προσπαθήσουν να τον κάνουν να δουλέψει σωστά, αν δεν δουλέψει με την πρώτη. Φυσικά θα υπάρξει πολύς πειραματισμός και συνεργασία μεταξύ των μαθητών.

## 9. ΑΝΑΠΑΡΑΣΤΑΣΕΙΣ ΤΩΝ ΜΑΘΗΤΩΝ/ΠΡΟΒΛΕΨΗ ΔΥΣΚΟΛΙΩΝ ΣΤΟ

### ΣΕΝΑΡΙΟ

Οι μαθητές της Γ Γυμνασίου, αν και έχουν εμπειρία προγραμματισμού (γλώσσα προγραμματισμού scratch) όταν βλέπουν **κώδικα σε C++**, το **θεωρούν ακαταλαβίστικο** και πιστεύουν ότι είναι αδύνατο να καταλάβουν τι κάνει, πολύ περισσότερο δε να γράψουν αυτοί ένα τέτοιο πρόγραμμα. Χρειάζεται να τους ενθαρρύνουμε να ασχοληθούν προσεκτικά με τις **ερωτήσεις των φύλλων εργασίας**, που τους βοηθούν **να διαβάσουν και να καταλάβουν μόνοι τους το κώδικα**. Όταν το καταφέρνουν αυτό νοιώθουν μεγάλη χαρά (κι εμείς φυσικά ακόμη μεγαλύτερη) και θέλουν να συνεχίσουν με μεγαλύτερη όρεξη. Πρέπει όμως να φροντίσουμε να μην εγκαταλείψουν την προσπάθεια πριν ακόμη ξεκινήσουν.

Πέρα από τον κώδικα της C++, αυτό που θα τους δυσκολέψει φυσικά είναι τα **ηλεκτρονικά κυκλώματα** (αντιστάσεις, καλώδια, τάσεις κτλ.) Αν και στην Γ Γυμνασίου ασχολούνται στην Φυσική με τα κυκλώματα και τον ηλεκτρισμό, αποφασίσαμε να μην επιμείνουμε στις γνώσεις ηλεκτρονικών και όπου είναι δυνατό να τους παρέχουμε έτοιμη την πληροφορία που χρειάζονται (συνδέσεις, μέγεθος αντιστάσεων κτλ.). Έτσι **επικεντρωνόμαστε στο κομμάτι που μας ενδιαφέρει περισσότερο, τον προγραμματισμό.**

Επίσης, μια επιπλέον δυσκολία θα αντιμετωπίσουν, όσοι δεν ακολουθήσουν τις οδηγίες και δεν **δουν τα tutorials στο σπίτι**.

Για την επίτευξη των στόχων του σεναρίου απαιτείται ενεργητική συμμετοχή και προσπάθεια από τους μαθητές με την ενθάρρυνση και συνεχή παρακολούθηση από τον καθηγητή.



## 10. ΔΙΔΑΚΤΙΚΟ ΣΥΜΒΟΛΑΙΟ-ΔΙΔΑΚΤΙΚΗ ΜΕΤΑΤΟΠΙΣΗ ΘΕΩΡΗΤΙΚΑ

### ΘΕΜΑΤΑ-ΔΙΔΑΚΤΙΚΟΣ ΘΟΡΥΒΟΣ

Γίνεται γνωστό στους μαθητές ότι στα επόμενα 2 μαθήματα θα ασχοληθούμε με το arduino και για να γίνει με επιτυχία αυτό θα χρειαστούν κάποιες γνώσεις που θα πρέπει να τις αποκτήσουν μέσω video-tutorials από το σπίτι. **Τους ενημερώνουμε ότι θα είναι αδύνατο να παρακολουθήσουν την πορεία των 2 επόμενων μαθημάτων αν δεν κάνουν την προετοιμασία στο σπίτι.** Τους ζητάμε να μας πουν την άποψή τους και αν συμφωνούν με αυτό. Πρέπει να βεβαιωθούμε ότι όλοι οι μαθητές κατάλαβαν το πόσο αναγκαίο είναι να γίνει αυτό, ειδικά αν δεν έχουμε ξαναχρησιμοποιήσει αυτή την εκπαιδευτική μέθοδο.

Επίσης τους **πληροφορούμε** ότι θα συμμετάσχουν ενεργά στην μαθησιακή διαδικασία ακολουθώντας τις οδηγίες 2 φύλλων εργασίας για την **κατασκευή αυτοματισμών** με arduino και ότι θα απαιτηθεί συνεργασία με τους συμμαθητές τους (ομάδες των 2 ατόμων).

Φυσικά δεν θα είναι εντελώς μόνοι τους σε αυτή την διαδικασία. Ο καθηγητής θα παρεμβαίνει όπου είναι απαραίτητο για να εισάγει ή εξηγήσει τις νέες έννοιες και να δώσει βοήθεια.

## 11. ΥΠΟΚΕΙΜΕΝΗ ΘΕΩΡΙΑ ΜΑΘΗΣΗΣ

Το σενάριο βασίζεται στη γνωστική **θεωρία μάθησης του εποικοδομητισμού (constructivism)** και στη **γνωστική θεωρία της ανακαλυπτικής μάθησης του (Bruner)**. Οι μαθητές καλούνται να πειραματιστούν με έτοιμα απλά προγράμματα, να ανακαλύψουν τι κάνουν οι εντολές, τι αποτελέσματα έχουν κάποια λάθη κοκ. Ο καθηγητής δεν είναι αυτός που τους μεταφέρει την γνώση αλλά μόνοι τους μέσα από τον πειραματισμό και την δοκιμή ανακαλύπτουν τον τρόπο με τον οποίο δημιουργούμε ένα πρόγραμμα χρησιμοποιώντας μια γλώσσα προγραμματισμού υψηλού επιπέδου. Η μάθηση έχει ενεργητικό χαρακτήρα. **Learning by Doing.** Ο μαθητής αναλαμβάνει ενεργό ρόλο στην οικοδόμηση της γνώσης του, δεν είναι παθητικός δέκτης των πληροφοριών που μεταδίδει ο καθηγητής. Οι προηγούμενες γνώσεις του μαθητή πρέπει να τροποποιηθούν και να επεκταθούν. Ο δάσκαλος είναι καθοδηγητής σε αυτή τη διαδικασία, αναλαμβάνει έναν υποστηρικτικό-συμβουλευτικό ρόλο στη δραστηριότητα των μαθητών. Σύμφωνα με τον εποικοδομισμό η μάθηση είναι μία υποκειμενική και εσωτερική διαδικασία οικοδόμησης νοημάτων και θεωρείται το αποτέλεσμα οργάνωσης και προσαρμογής των νέων πληροφοριών σε ήδη υπάρχουσες γνώσεις.

Επίσης βασιζόμαστε και στις **κοινωνιογνωστικές θεωρίες μάθησης** (συγκεκριμένα στον κοινωνικό εποικοδομισμό του Vygotsky) Σύμφωνα με τις θεωρίες αυτές, η μάθηση συντελείται με την αλληλεπίδραση του ατόμου με άλλα άτομα, σε συγκεκριμένες επικοινωνιακές περιστάσεις και μέσω της υλοποίησης κοινών δραστηριοτήτων (activities). Παροτρύνουμε τους μαθητές μας να εργαστούν συνεργατικά συμπληρώνοντας τα φύλλα εργασίας τους ως ομάδα, μετατρέποντας τις κοινωνικές σχέσεις σε νοητικές λειτουργίες. Οι μαθητές οικοδομούν την γνώση δουλεύοντας με τον δικό τους ρυθμό μαθαίνοντας να συνεργάζονται και να διαπραγματεύονται.

## 12. ΟΡΓΑΝΩΣΗ ΤΗΣ ΤΑΞΗΣ-ΕΦΙΚΤΟΤΗΤΑ ΣΧΕΔΙΑΣΗΣ

Το σενάριο υλοποιείται στο **εργαστήριο πληροφορικής** του σχολείου. Οι μαθητές δουλεύουν σε **ομάδες των 2 ατόμων**. Οι μαθητές συνεργάζονται για την συμπλήρωση των φύλλων εργασίας.

Κάθε ομάδα διαθέτει **ηλεκτρονικό υπολογιστή, πλακέτα arduino με breadboard και τα εξαρτήματα που είναι απαραίτητα για την υλοποίηση των δραστηριοτήτων (LED, Buzzer, LDR, καλώδια κτλ.)**

Ο προβλεπόμενος χρόνος κρίνεται ότι είναι οριακά επαρκής για την υλοποίηση του σεναρίου. Είναι **απαραίτητη η προετοιμασία των μαθητών στο σπίτι** (πρώτη επαφή με το arduino, tutorials). Αν δεν γίνει αυτό χρειάζεται μια επιπλέον ώρα για να τους μεταφέρουμε αυτές τις βασικές γνώσεις. Στην περίπτωση που κάποιοι μόνο μαθητές δεν κάνουν την απαραίτητη προετοιμασία στο σπίτι, φροντίζουμε να τους βάλουμε σε ζεύγη έτσι ώστε τουλάχιστον ο ένας να είναι προετοιμασμένος.

## 13. ΔΙΔΑΚΤΙΚΕΣ ΠΡΟΣΕΓΓΙΣΕΙΣ ΚΑΙ ΣΤΡΑΤΗΓΙΚΕΣ

Για την επίτευξη των διδακτικών μας στόχων χρησιμοποιούμε κυρίως τις διδακτικές στρατηγικές του **πειραματισμού** και **διερεύνησης**. Επιδιώκουμε όσο το δυνατό να οδηγήσουμε τους μαθητές στην **ανακάλυψη**. Οι μαθητές χωρίς να γνωρίζουν τίποτε για την γλώσσα προγραμματισμού του arduino καλούνται να τροποποιήσουν έτοιμα προγράμματα, να πειραματιστούν με αυτά, να κάνουν υποθέσεις και να τις επαληθεύσουν. Τους καθοδηγούμε έτσι ώστε να ανακαλύψουν μόνοι τους τις εντολές. Χρησιμοποιώντας την μέθοδο του **μαύρου κουτιού** (παρατηρούν πρώτα το αποτέλεσμα του προγράμματος, χωρίς να γνωρίζουν τον κώδικα και στη συνέχεια γνωρίζοντας το

αποτέλεσμα, διαβάζουν τον κώδικα και προσπαθούν να αντιστοιχίσουν σε αυτό που παρατήρησαν την εντολή που το δημιουργεί).

Επίσης χρησιμοποιούμε την καινοτόμο μέθοδο της **Flipped Classroom**. Αντιστρέφουμε δηλαδή τη συνηθισμένη εκπαιδευτική διαδικασία και επιλέγουμε να γίνει **στο σπίτι η «παράδοση» της θεωρίας** (η οποία όταν γίνεται στην αίθουσα είναι συνήθως δασκαλοκεντρική και βαρετή για τους μαθητές) και αφήνουμε για **την τάξη την εξάσκηση και την πρακτική εφαρμογή** όσων είδανε/μάθανε στο σπίτι σε συνεργασία με τους άλλους μαθητές (μαθητοκεντρική διαδικασία).

Ακόμη χρησιμοποιούμε τις διδακτικές τεχνικές των **ερωτοαποκρίσεων και εργασίας των μαθητών σε ομάδες** για την συμπλήρωση φύλλων εργασίας. Χρησιμοποιούμε μαθητοκεντρικές τεχνικές διδασκαλίας θέτοντας τον μαθητή πρωταγωνιστή της μαθησιακής διαδικασίας.

Με τη εργασία των μαθητών σε ομάδες (**συνεργατική μάθηση**) εξασφαλίζουμε την ενεργό συμμετοχή όλων των μαθητών. Δημιουργούμε το κατάλληλο μαθησιακό περιβάλλον ώστε ο μαθητής μέσω της αλληλεπίδρασης με τους συμμαθητές του, επιλύοντας προβλήματα να καταφέρει να οικοδομήσει τη νέα γνώση συνδέοντας την με την πρότερη, πάντα με τον καθηγητή αρωγό και καθοδηγητή.

## 14. ΑΞΙΟΛΟΓΗΣΗ ΤΩΝ ΜΑΘΗΤΩΝ

Δεν προβλέπεται κάποια ξεχωριστή διαδικασία αξιολόγησης των μαθητών. Η αξιολόγηση των μαθητών γίνεται ουσιαστικά από την σωστή συμπλήρωση των φύλλων εργασίας.

# Arduino-Φύλλο Εργασίας 1

## Lets LED

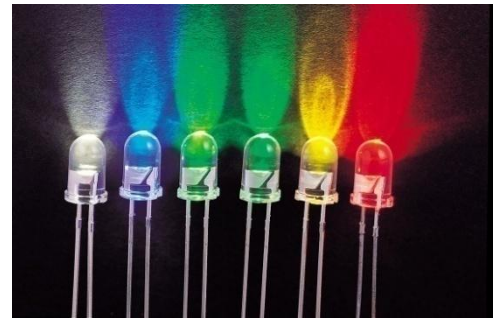
Θα κάνουμε τα φωτάκια LED να αναβοσβήνουν στους ρυθμούς που εμείς θέλουμε....

Τμήμα: .....

Ημερομηνία: .....

Μαθητής/τρια1: .....

Μαθητής/τρια2: .....



- 1) Στον φάκελο «Κοινόχρηστα Έγγραφα→Arduino→FlashingLED» θα βρείτε το αρχείο/πρόγραμμα «FlashingLed.ino». Κάνοντας διπλό κλικ στο αρχείο θα ανοίξει το προγραμματιστικό περιβάλλον του arduino και θα δείτε το περιεχόμενο του «FlashingLed.ino». Μεταγλωττίστε και ανεβάστε τον εκτελέσιμο κώδικα στην πλακέτα arduino που σας έχει δοθεί από τον καθηγητή σας. (Θα πρέπει φυσικά να την συνδέσετε στον υπολογιστή σας μέσω της USB θύρας).

**Βοήθεια:** για την μεταγλώττιση επιλέξτε από το Μενού **Σχέδιο→Επικύρωση/Μεταγλώττιση**, για το ανέβασμα του εκτελέσιμου κώδικα στην πλακέτα επιλέξτε από το Μενού **Σχέδιο→Ανέβασμα**. Τι κάνει το πρόγραμμα;:.....

- 2) Η γλώσσα προγραμματισμού που χρησιμοποιούμε για τον προγραμματισμό του arduino είναι η C++. Στο διαδίκτυο μπορείτε να βρείτε μια κατάταξη των δημοφιλέστερων γλωσσών προγραμματισμού για τον τρέχοντα μήνα..

a) Ποια είναι η θέση της C++ στην κατάταξη;:.....

b) Ποια γλώσσα προγραμματισμού είναι στην πρώτη θέση;: .....

c) Σε ποια θέση ήταν η C++ το 1986: .....

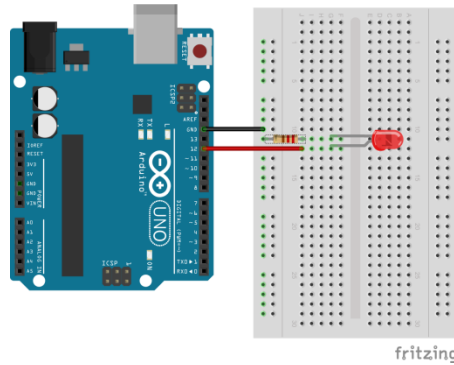
d) Σε ποια θέση ήταν η Java το 1986: .....

- 3) Σβήστε το ερωτηματικό από την εντολή `digitalWrite(LED, HIGH)` ; Μεταγλωττίστε και τρέξτε το πρόγραμμα. Τι παρατηρείτε; Τι μήνυμα εμφανίζεται;:

.....  
.....

Πως ονομάζεται το λάθος που μόλις δημιουργήσατε;:.....

Διορθώστε το λάθος σας, προσθέστε το ερωτηματικό που το συντακτικό της C++ απαιτεί στο τέλος κάθε εντολής.



- 4) Παρατηρήστε το arduino σας και το breadboard. Συμβουλευτείτε και τον κώδικα του προγράμματός. Σε ποιον ακροδέκτη(pin) είναι συνδεδεμένο το LED;:.....

**Βοήθεια:** Σε ποιον ακροδέκτη είναι συνδεδεμένο το κόκκινο καλώδιο;

Μέσω αυτού του ακροδέκτη δίνουμε «εντολή» στο LED να ανάψει ή να σβήνει.

- 5) Διαβάστε το πρόγραμμα. Με ποια εντολή του προγράμματος πιστεύετε ότι ανάβει το LED;:

.....

- 6) Διαβάστε το πρόγραμμα. Με ποια εντολή του προγράμματος πιστεύετε ότι σβήνει το LED;:

.....

- 7) Στο πρόγραμμα αλλάξτε την εντολή `digitalWrite(LED, HIGH)` ; σε `digitalWrite(11, HIGH)` ; Μεταγλωττίστε και τρέξτε το πρόγραμμα. Τι παρατηρείτε;

.....

Γιατί πιστεύετε ότι έγινε αυτό; :.....

- 8) Αλλάξτε το καλώδιο που συνδέει το LED από το pin 12 στο pin 11. Κάντε τις κατάλληλες τροποποιήσεις στο πρόγραμμα έτσι ώστε το LED να αναβοσβήνει.

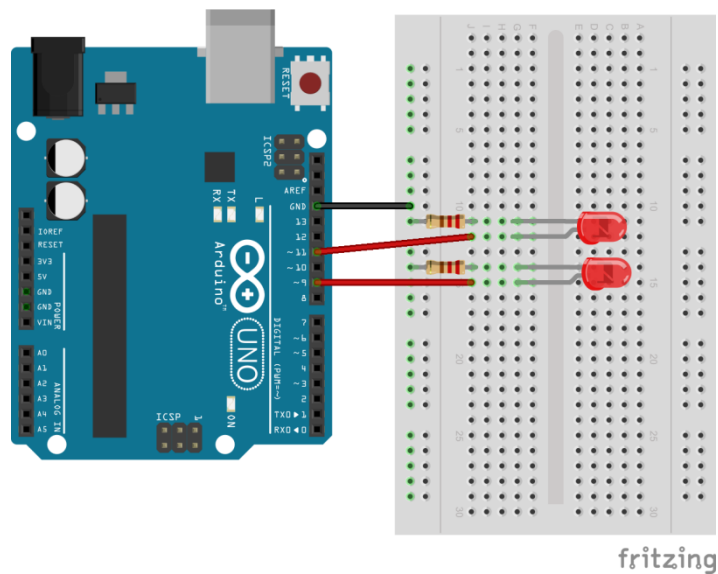
- 9) Η εντολή `delay(3000)`; Δημιουργεί μια καθυστέρηση 3000 msec = ..... sec (είναι αντίστοιχη της εντολής περίμενε της scratch). Κάντε τις κατάλληλες τροποποιήσεις στο πρόγραμμα έτσι ώστε το LED να **ανάβει για 4 sec και να σβήνει για 6 sec**. Δοκιμάστε το πρόγραμμά σας. Οι τροποποιημένες εντολές:

```
void loop() {
    digitalWrite(11, HIGH)
    delay(.....);
    digitalWrite(11, LOW);
    delay(.....);
}
```

- 10) Αποθηκεύστε το πρόγραμμά σας στο φάκελο σας με όνομα «**MyFlashingLED.ino**».

**Υπόδειξη:** Από το μενού Αρχείο→ΑποθήκευσηΩς

11) Προσθέστε ακόμη ένα LED στο breadboard, όπως βλέπετε στο παρακάτω σχήμα.



12) Προσθέστε στο πρόγραμμά σας τις κατάλληλες εντολές έτσι ώστε:

α) το δεύτερο LED να ανάβει αμέσως μόλις σβήσει το πρώτο LED και να παραμένει αναμμένο για 4 sec (δηλαδή τα LED θα ανάβουν εναλλάξ για 4 sec το καθένα). Εντολές:









```
void loop() {  
    digitalWrite(11, HIGH)  
    delay(4000);  
    digitalWrite(11, LOW);  
    digitalWrite(....., HIGH);  
    delay(.....);  
    digitalWrite(....., .....);  
}
```

Αποθηκεύστε το πρόγραμμά σας στο φάκελό σας με όνομα «**2LEDs.ino**» (Αρχείο→ΑποθήκευσηΩς)

13) Τροποποιήστε το πρόγραμμά σας έτσι ώστε να συμβαίνει το εξής:

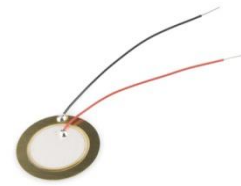
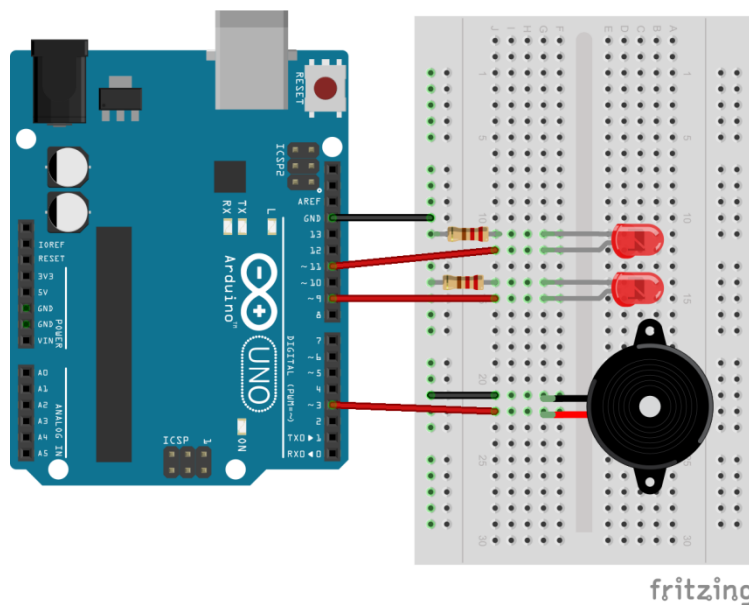
το πρώτο LED ανάβει για 5 sec, στη συνέχεια σβήνει και ανάβει το 2<sup>ο</sup> LED για άλλα 5 sec, στη συνέχεια ανάβουν και τα 2 LEDs για άλλα 5 sec και τέλος σβήνουν και τα 2 LEDs και παραμένουν σβηστά για 5sec.

Αποθηκεύστε το πρόγραμμά σας με το όνομα «**2LEDsCount.ino**»

LOW	HIGH		
			
.....	.....		
.....	.....		

## ΕΠΕΚΤΑΣΗ:

Προσθέστε στο arduino-breadboard ένα ηχείο (piezo Buzzer) με την συνδεσμολογία που βλέπετε παρακάτω:



Η εντολή **tone(3, 550, 3000)** δημιουργεί έναν ήχο στο ηχείο που είναι συνδεδεμένο στο Pin 3, συχνότητας 550Hz και διάρκειας 3000 msec.

α) Εισάγετε την εντολή στο πρόγραμμά σας έτσι ώστε όταν ανάβουν και τα 2 LED **ταυτόχρονα** να ακούγεται ο παραπάνω ήχος.

β) Κάντε τις απαραίτητες τροποποιήσεις στο πρόγραμμα έτσι ώστε να διατηρηθούν οι χρόνοι του 13ου ερωτήματος (κάθε συνδυασμός διατηρείται για 5 sec)

γ) Κάντε όσες τροποποιήσεις και δοκιμές. Προσθέστε και σε άλλα σημεία του προγράμματος ήχους, παίζετε με τις συχνότητες. Έτσι κάνουν όλοι οι προγραμματιστές άλλωστε. Σπάνια κάτι δουλεύει με την... πρώτη!!!

Καλή διασκέδαση....

## Arduino-Φύλλο Εργασίας 2

### SchoolBag Alarm

Τμήμα: .....

Ημερομηνία: .....

Μαθητής/τρια1: .....

Μαθητής/τρια2: .....



Έχει παρατηρηθεί μερικές φορές να εξαφανίζονται κάποια πράγματα από τις σχολικές τσάντες των μαθητών. Τι θα λέγατε να κατασκευάσαμε με arduino ένα εύκολο συναγερμό σχολικής τσάντας, που θα σφυρίζει μόλις κάποιος ανοίξει το φερμουάρ της τσάντας;

Θα χρησιμοποιήσουμε μία φωτοαντίσταση ([LDR](#) - Light Dependent Resistor) που θα χρησιμοποιηθεί ως αισθητήρας φωτός, ένα LED και ένα piezo Buzzer (ηχείο).

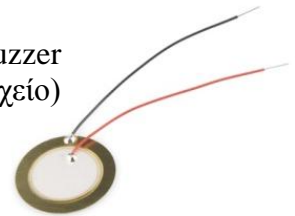
Φωτοαντίσταση



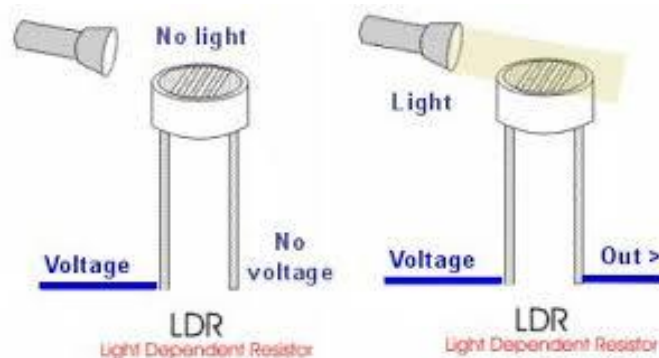
LED



Piezo Buzzer  
(Ηχείο)



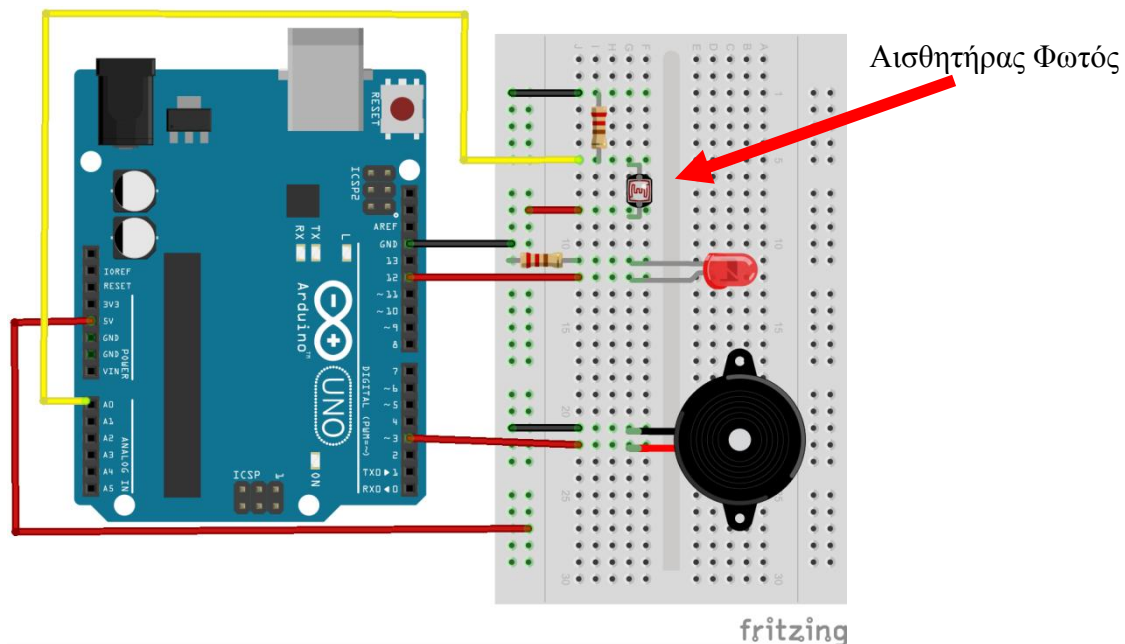
**Πως λειτουργεί η φωτοαντίσταση:** Ανάλογα με την ένταση του φωτός που πέφτει πάνω της, περνάει περισσότερο ή λιγότερο ρεύμα από το κυκλωμάκι μας. Έτσι αν διαβάσουμε την ένδειξη που δίνει το pin στο οποίο είναι συνδεδεμένη η φωτοαντίσταση (A0 στο σχήμα παρακάτω, κίτρινο καλώδιο) θα έχουμε υψηλή τιμή όταν ο LDR δέχεται πολύ φως και χαμηλή τιμή στο σκοτάδι.



Η φωτοαντίσταση χρησιμοποιείται σε ηλεκτρονικά κυκλώματα ελέγχου του φωτισμού, όπως π.χ ο φωτοδιακόπτης για τον έλεγχο του οδικού ή άλλου εξωτερικού φωτισμού. Μια άλλη χρήση είναι για τον έλεγχο λειτουργίας της λάμπας νυκτός που συνήθως έχουμε σε διαδρόμους ή στα υπνοδωμάτια. Χρησιμοποιείται ακόμα σε αυτοματισμούς, συστήματα συναγερμού, πόρτες που ανοίγουν αυτόματα κ.λ.π.







1) Συνδέστε στο arduino σας τον αισθητήρα φωτός (LDR) , όπως στο παραπάνω σχήμα.

2) Αρχικά θα μελετήσουμε τις τιμές που δίνει ο αισθητήρας.

Στον φάκελο «Κοινόχρηστα Έγγραφα→Arduino→CheckLDRValues» θα βρείτε το αρχείο/πρόγραμμα «CheckLDRValues.ino». Κάνοντας διπλό κλικ στο αρχείο θα ανοίξει το προγραμματιστικό περιβάλλον του arduino και θα δείτε το περιεχόμενο του «CheckLDRValues.ino». Μεταγλωττίστε και ανεβάστε τον εκτελέσιμο κώδικα στην πλακέτα arduino που σας έχει δοθεί από τον καθηγητή σας.

**Βοήθεια:** για την μεταγλώττιση επιλέξτε από το Μενού **Σχέδιο→Επικύρωση/Μεταγλώττιση**, για το ανέβασμα του εκτελέσιμου κώδικα στην πλακέτα επιλέξτε από το Μενού **Σχέδιο→Ανέβασμα**

Το πρόγραμμα διαβάζει και εμφανίζει στην οθόνη τις τιμές που δίνει ο αισθητήρας φωτός. Για να δείτε τις τιμές στην οθόνη σας επιλέξτε από το Μενού **Εργαλεία→Παρακολούθηση Σειριακής**.

**Τι τιμές δίνει ο αισθητήρας:**

- α) Στο φως:
- β) Όταν τον σκιάσετε λίγο:
- γ) Όταν τον σκιάσετε λίγο περισσότερο:
- δ) Στο σκοτάδι:


3) Πως θα χρησιμοποιήσουμε τον αισθητήρα φωτός για το SBA (SchoolBagAlarm):

Η κατασκευή μας θα τοποθετηθεί μέσα στην σχολική μας τσάντα. Όταν κάποιος ανοίξει το φερμουάρ της τσάντας μας, ο αισθητήρας φωτός θα ανιχνεύσει το φως. Τότε ο piezzo Buzzer θα δημιουργήσει ήχο και το λαμπάκι LED θα ανάψει. Στη αντίθετη περίπτωση (σκοτάδι) ο piezzo Buzzer θα σταματήσει να παράγει ήχο και το λαμπάκι LED θα σβήσει.

**Συμπληρώστε** στην πρώτη στήλη του παρακάτω πίνακα τον αλγόριθμο που περιγράφει τις παραπάνω ενέργειες. Επιλέξτε από τις φράσεις: (άναψε LED, σβήσε LED, ενεργοποίησε buzzer, απενεργοποίησε buzzer)

Αλγόριθμος	Κώδικας C
<p>Αν τιμή αισθητήρα &gt;.....</p> <p>Τότε</p> <p>.....;</p> <p>.....;</p> <p>Αλλιώς</p> <p>.....;</p> <p>.....;</p>	<pre> if ( ..... &gt; ..... ) { .....; .....; } else { .....; .....; } </pre>

4) Αντιστοιχίστε τις εντολές της πρώτης στήλης με τις ενέργειες της δεύτερης στήλης :

- `tone(SPEAKER,500);`

Δίνει εντολή να σταλεί υψηλή τάση ρεύματος στο pin με αριθμό όσο η μεταβλητή LED. Αν υπάρχει εκεί συνδεδεμένο LED τότε αυτό θα ανάψει.

- `noTone(SPEAKER);`

Δίνει εντολή να σταλεί χαμηλή τάση ρεύματος στο pin με αριθμό όσο η μεταβλητή LED. Αν υπάρχει εκεί συνδεδεμένο LED τότε αυτό θα σβήσει.

- `digitalWrite(LED, HIGH);`

Διαβάζει την τιμή που δίνει ο αισθητήρας που είναι συνδεδεμένος στο pin με αριθμό όσο η μεταβλητή LDR

- `digitalWrite(LED, LOW);`

Σταματάει τον buzzer, που είναι συνδεδεμένος στο pin με αριθμό όσο η τιμή της μεταβλητής SPEAKER, από το να δημιουργεί ήχο

- `analogRead(LDR)`

Ο buzzer, που είναι συνδεδεμένος στο pin με αριθμό όσο η τιμή της μεταβλητής SPEAKER, δημιουργεί συνέχεια, ήχο με συχνότητα 500HZ

5) Τώρα που γνωρίσατε τις εντολές που πρέπει να χρησιμοποιήσετε για να υλοποιήσετε τον παραπάνω αλγόριθμο σε C, συμπληρώστε την δεύτερη στήλη του πίνακα της άσκησης 3. Τοποθετείστε με την κατάλληλη σειρά τις παραπάνω εντολές μέσα σε μία if...else (Η if ...else είναι η αντίστοιχη εντολή της εάν...τότε της scratch)..

6) Προσθέστε τον κώδικα που μόλις γράψατε μέσα στην loop() του προγράμματος «**CheckLDRValues.ino**». Δοκιμάστε το πρόγραμμά σας.