***Εθνικό Μετσόβιο Πολυτεχνείο***

***Δ.Π.Μ.Σ. Συστήματα Αυτοματισμού***

***Κατεύθυνση Β:***

***Συστήματα Αυτομάτου Ελέγχου και Ρομποτικής***

***Μεταπτυχιακό Μάθημα:***

***Αισθητήρες***

***Ομάδα 3***

***Τρίτη και Τέταρτη Εργαστηριακή Άσκηση***

***Εισαγωγή στο Arduino***

***Μέλη Ομάδας – Α.Μ.:***

***Γεώργιος Κασσαβετάκης – 02121203***

***Γεώργιος Κρομμύδας – 02121208***

***Λάμπης Παπακώστας – 02121211***

ΑΘΗΝΑ

2023

# **1. Παράδειγμα - LED Blink**

Σε αυτό το παράδειγμα μας δόθηκε ο κώδικας ***blink.ino***. Σκοπός του συγκεκριμένου προγράμματος είναι να αναβοσβήνει το LED λαμπάκι που υπάρχει ενσωματωμένο στην πλακέτα του ***Arduino UNO REV3*** και βρίσκεται στο ***pin D13*** της πλακέτας.



Σχήμα 1. Πλακέτα Arduino UNO REV3

## **1.1 Ερώτημα 1**

Σε αυτό το ερώτημα έχουμε να τροποποιήσουμε τον δοθέντα κώδικα έτσι ώστε το ***LED*** να παραμένει αναμμένο για 1.5 sec και σβηστό για 0.5 sec. Αναλυτικά μπορούμε να δούμε τροποποιημένο τον κώδικα στο σχήμα 2.

Εικόνα που περιέχει κείμενο

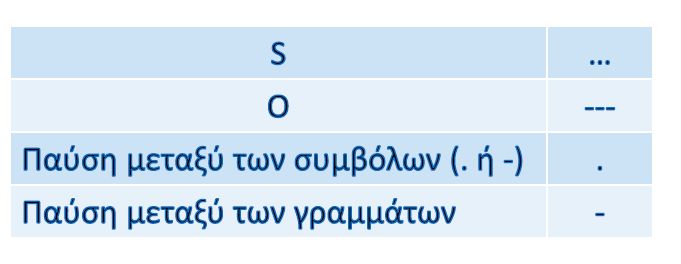
Περιγραφή που δημιουργήθηκε αυτόματα

Σχήμα 2. Κώδικας .ino πρώτου ερωτήματος

Με την έναρξη της εκτέλεσης της συνάρτησης του κώδικα, ξεκινάει η λειτουργία του προγράμματος. Η συνάρτηση θα δώσει σήμα στο ***LED*** ώστε να ανάψει. Αντίστοιχα θα δοθεί από την ίδια συνάρτηση σήμα για να σβήσει το ***LED***. Τροποποιούμε το όρισμα των συναρτήσεων . Για να παραμείνει το ***LED*** αναμμένο για 1.5 sec, το όρισμα θα αλλάξει από σε . Αντίστοιχα για να παραμείνει σβηστό για 0.5 sec, θα τροποποιηθεί το όρισμα από σε .

## **1.2 Ερώτημα 2**

Σε επέκταση του προηγούμενου ερωτήματος, τροποποιήθηκε ο κώδικας έτσι ώστε να εκπέμπεται από το ***LED*** η λέξη ***SOS*** σύμφωνα με τα σήματα *Morse*. Αναλυτικά στον πίνακα 1 παρουσιάζονται τα γράμματα που χρειαζόμαστε.



Πίνακας 1. Σήματα Morse Γραμμάτων

Καθώς το γράμμα ***S*** χρειάζεται τρεις τελείες για να υλοποιηθεί, τότε θα πρέπει να αναβοσβήσει τρεις φορές. Οπότε αυτό που θα κάνουμε είναι να φτιάξουμε ένα ***for loop*** στο οποίο θα αναβοσβήνουμε το λαμπάκι κάθε 0.5 sec. Με χρήση των συναρτήσεων θα δίνουμε σήμα στο λαμπάκι ώστε να παίρνει τιμή 1 και 0. Ενδιάμεσα από αυτές θα χρησιμοποιηθούν και δύο συναρτήσεις με όρισμα τα . Έτσι θα σχηματιστούν οι τρεις τελείες και το γράμμα ***S***. Με παρόμοιο τρόπο συντάσσεται και το block που αντιστοιχεί στο γράμμα ***Ο*** με την μόνη διαφορά ότι αλλάζει το όρισμα της συνάρτησης στα 1500 *msec*. Επιπλέον, ανάμεσα από τα σύμβολα έχουν τοποθετηθεί συναρτήσεις στα 500 msec. Τέλος, μετά την εμφάνιση κάποιου γράμματος έχουμε πάλι καθυστέρηση 1500 *msec* για να υπάρχει μια παύση. Στο σχήμα 3 μπορούμε να δούμε και το block κώδικα που αντιστοιχεί σε αυτή την διαδικασία.

Εικόνα που περιέχει κείμενο

Περιγραφή που δημιουργήθηκε αυτόματα

Σχήμα 3. Κώδικας .ino δεύτερου ερωτήματος

# **2. Παράδειγμα – Έλεγχος LED μέσω σειριακής επικοινωνίας**

Σε αυτό το παράδειγμα μας δίνεται ο κώδικας ***PhysicalPixel.ino***. Η λειτουργία του συγκεκριμένου προγράμματος είναι να ανάβει το ***LED*** της πλακέτας μέσω μιας σειριακής εντολής. Οι σειριακές εντολές δίνονται μέσω του ***Serial Monitor*** του ***IDE***.

## **2.1 Ερώτημα 1**

Σε αυτό το ερώτημα είχαμε να τροποποιήσουμε τα ***if loops*** έτσι ώστε όταν διαβάζεται το επιθυμητό γράμμα από το ***Serial Monitor*** να δίνει σήμα στο ***LED*** για να ανάβει και να σβήνει. Ήδη, έχει υλοποιηθεί αυτό το κομμάτι με την χρήση των χαρακτήρων ***ASCII ‘H’*** και ***‘L’***. Εμείς επεκτείνουμε την λογική πράξη του ***if*** ώστε να δέχεται και τους χαρακτήρες ***‘h’*** και ***‘l’***. Έτσι, προσθέτοντας την λογική πράξη ***or*** που συμβολίζεται με || θα διαβάζονται σειριακά και οι δύο χαρακτήρες. Αναλυτικά μπορούμε να δούμε την διαδικασία στο σχήμα 4.

Εικόνα που περιέχει κείμενο

Περιγραφή που δημιουργήθηκε αυτόματα

Σχήμα 4. Κώδικας .ino πρώτου ερωτήματος

Έτσι, δίνοντας τον χαρακτήρα ***ASCII*** από το ***Serial Monitor***, θα διαβαστεί από την εντολή και θα αποθηκευτεί η τιμή στην μεταβλητή ***incomingByte***. Μέσα στο ***if loop*** θα χρησιμοποιηθεί η συνάρτηση ώστε να δώσε το απαραίτητο σήμα στο ***LED***.

## **2.2 Ερώτημα 2**

Σε αυτό το ερώτημα τροποποιήθηκε ο παραπάνω κώδικας έτσι ώστε κάθε φορά που διαβαστεί ένας χαρακτήρας ***ASCII*** από το ***Serial Monitor***  διαφορετικός από την επιθυμητή τετράδα να εμφανίζει ένα μήνυμα σφάλματος. Η διαδικασία αυτή φαίνεται αναλυτικά στο σχήμα 5.

Εικόνα που περιέχει κείμενο

Περιγραφή που δημιουργήθηκε αυτόματα

Σχήμα 5. Κώδικας .ino δεύτερου ερωτήματος

Τροποποιούμε το ***if loop*** έτσι ώστε να δημιουργηθεί μία δομή με ***if*** και ***else if***  και να κρατήσουμε την προηγούμενη δομή και κάθε φορά που διαβάζεται κάποιος άλλος χαρακτήρας, τότε θα εμφανίζει το μήνυμα που φαίνεται στο σχήμα 6. Για παράδειγμα δόθηκε ο χαρακτήρας και παράχθηκε το ***string*** που σηματοδοτεί το σφάλμα.



Σχήμα 6. Παράδειγμα εσφαλμένης εισόδου από το serial monitor

# **3. Παράδειγμα – Προσομοίωση φλόγας**

Σε αυτό το παράδειγμα μας δόθηκε ο κώδικας ***3\_LED\_Fire.ino***. Το συγκεκριμένο πρόγραμμα προσομοιώνει το φως που παράγει μια φωτιά. Αναλυτικά στο σχήμα 7 φαίνεται ο συνδεσμολογία του κυκλώματος. Παρατηρούμε ότι έχουν προστεθεί τρεις αντιστάσεις των . Οι αντιστάσεις αυτές χρειάζονται ώστε να υποβαθμιστεί η πτώση τάσης στο ***LED***, έτσι ώστε να μην υπάρχει κίνδυνος να καεί. Ένα ***LED*** έχει τάση λειτουργίας από τα 1.6 V έως τα 3V. Τα λαμπάκια δέχονται τυχαίες τιμές κάθε φορά με την χρήση της αναλογικής εισόδου από την συνάρτηση .



Σχήμα 7. Συνδεσμολογία κυκλώματος προσομοίωσης φλόγας

## **3.1 Ερώτημα 1**

Σε αυτό το ερώτημα είχαμε να τροποποιήσουμε τον κώδικα (βελτιστοποίηση) ώστε να μην αναγράφονται 3 φορές οι συναρτήσεις και **.** Δημιουργούμε δύο ***for loops*** στα οποία αναθέτουμε αντίστοιχα τα τιμές από το 9 έως το 11 που θα είναι οι έξοδοι του Arduino προς τα λαμπάκια. Είναι συνδεδεμένα σε PWM εξόδους για να μπορέσουν να πάρουν ενδιάμεσες τιμές και να παράγει η πλακέτα «αναλογικές τιμές». Αναλυτικά μπορούμε να δούμε τον κώδικα στο σχήμα 8. Επίσης, στο σχήμα 9 βλέπουμε και την πραγματική συνδεσμολογία του παραδείγματος.

Εικόνα που περιέχει κείμενο

Περιγραφή που δημιουργήθηκε αυτόματα

Σχήμα 8. Κώδικας .ino πρώτου ερωτήματος

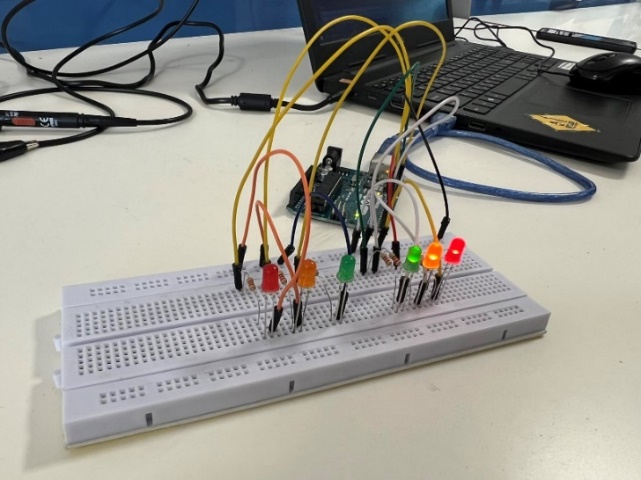
Εικόνα που περιέχει κείμενο, ηλεκτρονικές συσκευές

Περιγραφή που δημιουργήθηκε αυτόματα

Σχήμα 9. Συνδεσμολογία κυκλώματος πρώτου ερωτήματος

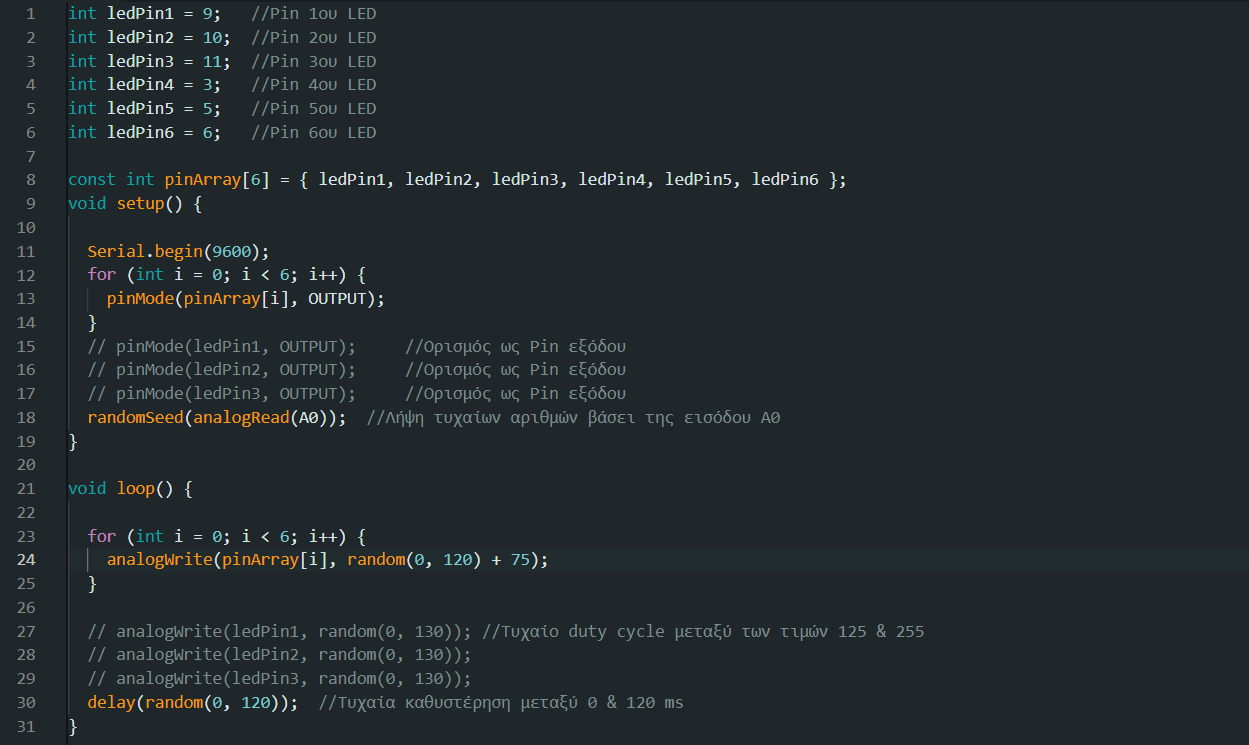
## **3.2 Ερώτημα 2**

Σε αυτό το ερώτημα ζητείται να επεκταθεί ο κώδικας για να διαχειρίζεται 6 ***LEDs***. Αρχικά, τροποποιούμε την συνδεσμολογία του κυκλώματος, προσθέτοντας 6 ***LED*** στο ***breadboard*** μαζί με 6 αντιστάσεις των . Τα ***LED*** είναι όλα συνδεδεμένα σε ***PWM(~)*** εξόδους, έτσι ώστε να μπορούν να πάρουν και ενδιάμεσες τάσεις. Αναλυτικότερα φαίνεται το κύκλωμα στο σχήμα 10.



Σχήμα 10. Συνδεσμολογία κυκλώματος δεύτερου ερωτήματος

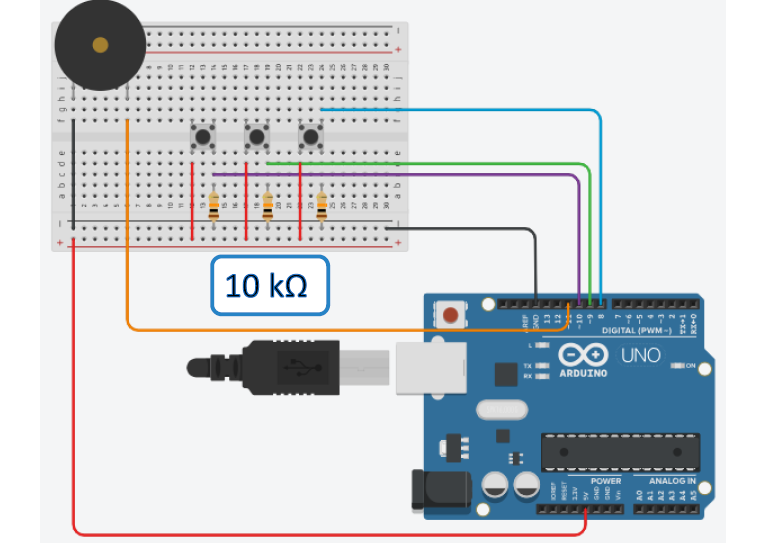
Το επόμενο βήμα είναι να τροποποιήσουμε τον κώδικα του προγράμματος, έτσι ώστε να λειτουργούν και τα 6 ***LED***. Αρχικά ορίζουμε τα ***pins*** ως ***int*** τιμές έτσι ώστε να χρησιμοποιήσουμε αυτές τις εξόδους του Arduino. Εν συνεχεία, δημιουργούμε μία στατική δομή δεδομένων (πίνακα), στον οποίο θα αποθηκεύσουμε τις μεταβλητές που περιέχουν την πληροφορία των εξόδων. Το επόμενο βήμα είναι να τροποποιήσουμε τα ***for loops***  έτσι ώστε να ορίζουμε τα και τα για τα παραπάνω ***pins***. Μέσα στις συνθήκες του ***for loop*** ορίζουμε έναν ***index***  ο οποίος θα τρέξει όλον τον πίνακα. Έτσι σε κάθε επανάληψη θα αναθέτουμε σε κάθε pin την ιδιότητα της εξόδου. Το επόμενο βήμα είναι να του αναθέτουμε τυχαία μια τιμή ώστε να ανάβει το κάθε λαμπάκι με κάποια τάση. Έτσι, αντίστοιχα ορίζουμε και ένα ***for loop***  όπου θα δίνουμε σε κάθε έξοδο μια τυχαία τιμή από το σύνολο τιμών . Το συγκεκριμένο διάστημα ορίστηκε, καθώς παρατηρήθηκε πως και τα 6 ***LED*** αναβοσβήνουν γρήγορα παίρνοντας διαφορετικές τάσεις. Αναλυτικότερα μπορούμε να δούμε τον κώδικα στο σχήμα 11.



Σχήμα 11. Κώδικας .ino δεύτερου ερωτήματος

# **5. Παράδειγμα – Piano**

Σε αυτό το παράδειγμα μας δόθηκε ο κώδικας ***Piano.ino***. Ο κώδικας αυτός αναπαράγει τρεις νότες με το πάτημα τριών κουμπιών. Οι νότες αυτές έχουν διαφορετικές συχνότητες. Κατά το πάτημα των κουμπιών αναπαράγεται ήχος μέσω ενός ***buzzer***. Η συνδεσμολογία αυτού του κυκλώματος είναι η εξής:



Σχήμα 12. Κύκλωμα διάταξης πιάνου

## **5.1 Ερώτημα 1**

Καθώς έχουν αφαιρεθεί οι αντιστάσεις από τα κουμπιά, θα προκύψει το νέο κύκλωμα που φαίνεται στο σχήμα 13. Παρατηρούμε πως έχει προκύψει κάποιο πρόβλημα.

Εικόνα που περιέχει κείμενο, ηλεκτρονικές συσκευές

Περιγραφή που δημιουργήθηκε αυτόματα

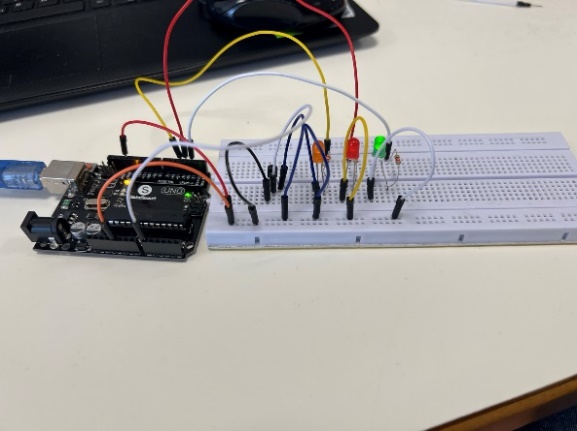
Σχήμα 13. Κύκλωμα διάταξης πιάνου χωρίς αντιστάσεις.

## **5.2 Ερώτημα 2**

Κατά την αφαίρεση των αντιστάσεων προκύπτει θόρυβος στο ***buzzer***. Καθώς πατιέται κάποιο κουμπί εφαρμόζεται η αντίστοιχη συχνότητα. Αυτό οφείλεται στην πτώση τάσης στα κουμπιά της συνδεσμολογίας. Καθώς οι αντιστάσεις χρησιμοποιούνται για την μείωση των ρευμάτων και του θορύβου στα ***buttons***.

# **6. Παράδειγμα – Αισθητήρας Θερμοκρασίας TMP36**

Σε αυτό το παράδειγμα μας δόθηκε ο κώδικας ***TMP36.ino***. Χρησιμοποιείται ένας αισθητήρας θερμοκρασίας ο οποίος ανιχνεύει την θερμοκρασία του χώρου. Το πρόγραμμα διαβάζει τα σήματα θερμοκρασίας του αισθητήρα και ανάβει το αντίστοιχο λαμπάκι το οποίο μας δείχνει το επίπεδο της θερμοκρασίας. Η συνδεσμολογία του κυκλώματος είναι η εξής:



Σχήμα 14. Κύκλωμα συνδεσμολογίας αισθητήρα θερμοκρασίας.

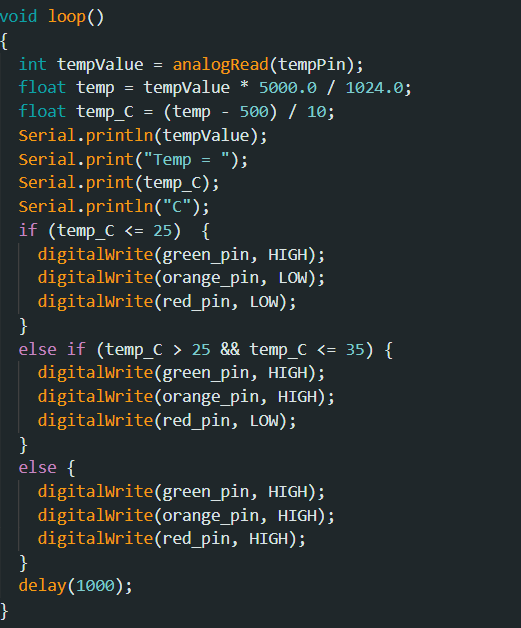
## **6.1 Ερώτημα 1**

Σε αυτό το ερώτημα είχαμε να τροποποιήσουμε τον κώδικα έτσι ώστε κάθε φορά που διαβάζεται μια θερμοκρασία, να ανάβει ένα ***LED***. Οι προδιαγραφές αυτές φαίνονται στον παρακάτω πίνακα.



Πίνακας 2. Αντιστοίχιση θερμοκρασιών και LED

Η τροποποίηση που θα κάνουμε στον κώδικα φαίνεται στο παρακάτω σχήμα.



Σχήμα 15. Κώδικας πρώτου ερωτήματος.

Κάθε φορά που διαβάζεται μια τιμή τάσης από το ***tempPin*** μέσω της εντολής , αυτή, τότε υπολογίζεται η τάση εξόδου σύμφωνα με τον τύπο της δεύτερης εντολής του σχήματος. Εν συνεχεία υπολογίζεται η θερμοκρασία, σύμφωνα με την τάση που διάβασε ο μικροελεγκτής από τον αισθητήρα. Τέλος, κάθε θερμοκρασία που έχει υπολογιστεί, ελέγχεται σε ένα if loop. Εάν είναι μικρότερη ή ίση με τους , τότε θα πρέπει να ανάψει μόνο το πράσινο ***LED***. Οπότε εισάγεται στον πρώτο βρόχο το πρόγραμμα και μέσω της εντολής θα ανάψει μόνο το πρώτο ***LED***. Στην περίπτωση που έχει διαβάσει μια θερμοκρασία μεταξύ των τιμών και, τότε θα εκτελεστεί το δεύτερο ***loop***. Τότε θα ανάψει το πράσινο και πορτοκαλί ***LED*** μέσω των εντολών . Σε περίπτωση που διαβαστεί θερμοκρασία μεγαλύτερη των , τότε αυτό που συμβαίνει είναι το πρόγραμμα να εισέρχεται στον τρίτο βρόχο και ανάβει και τα τρία ***LED***, μέσω της εντολής .

## **6.2 Ερώτημα 2**

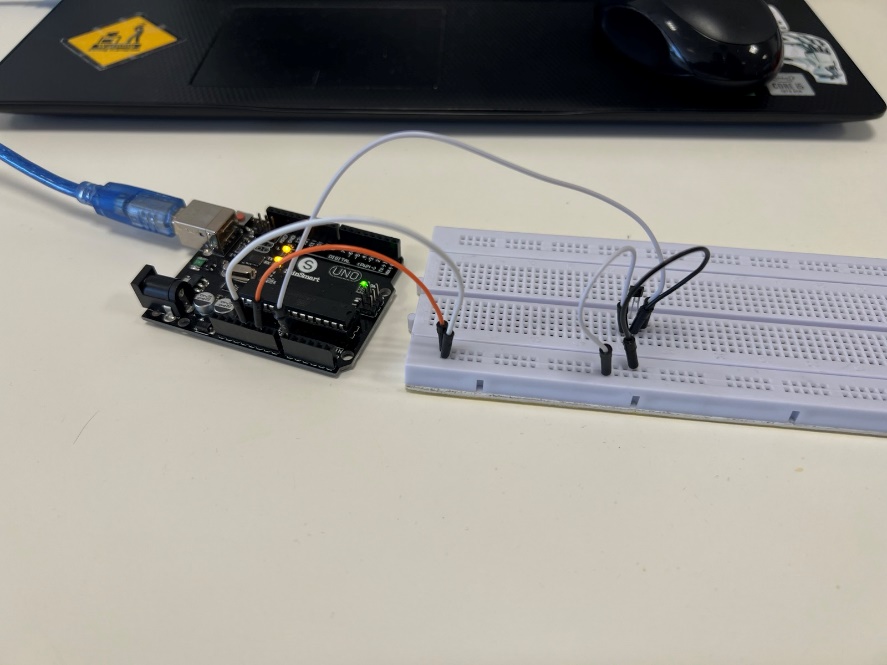
Στην περίπτωση που έχουμε τάση τροφοδοσίας και η ανάλυση του ***ADC*** του ***Arduino*** είναι , τότε θα έχουμε δείγματα . Έτσι, η εντολή που υπολογίζει την τάση εξόδου του αισθητήρα θα τροποποιηθεί ως εξής:



Σχήμα 16. Τύπος υπολογισμού τάσης εξόδου.

# **7. Παράδειγμα – Hall SS49**

Σε αυτό το παράδειγμα μας δίνεται ο κώδικας ***7\_Hall\_SS49.ino***. Το πρόγραμμα αυτό είναι υπεύθυνο να δέχεται μετρήσεις από τον αισθητήρα Hall SS49, ώστε να ανιχνεύεται το μαγνητικό πεδίο ενός μαγνήτη. Η συνδεσμολογία του κυκλώματος φαίνεται στο σχήμα 17.



Σχήμα17. Συνδεσμολογία κυκλώματος αισθητήρα Hall SS49.

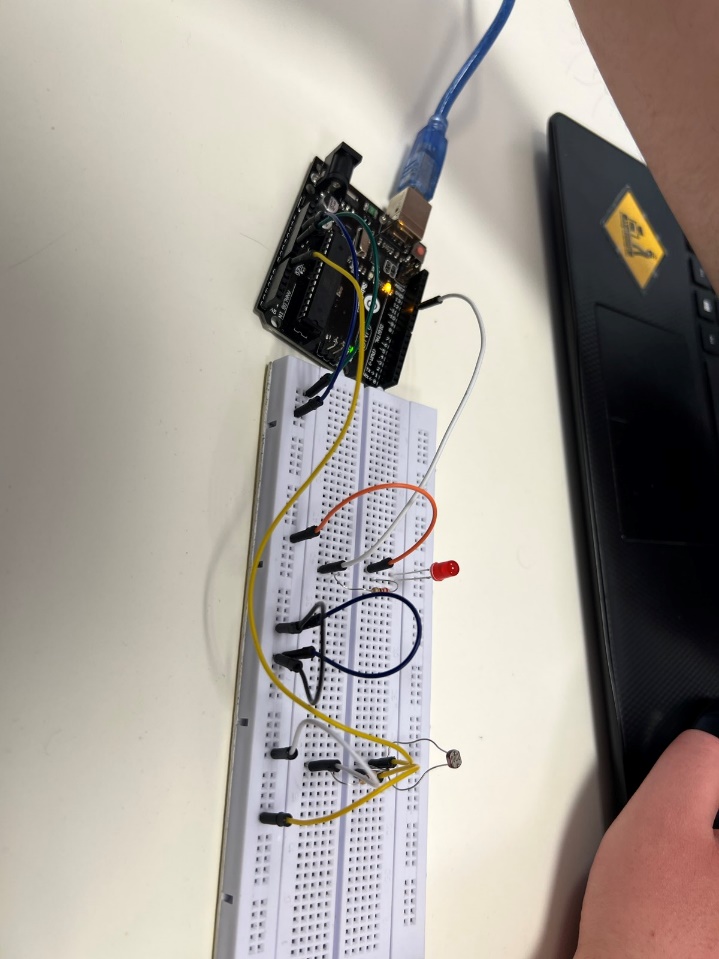
## **7.1 Ερώτημα 1**

## **7.2 Ερώτημα 2**

## **7.3 Ερώτημα 3**

# **8. Παράδειγμα – Αισθητήρας φωτεινότητας**

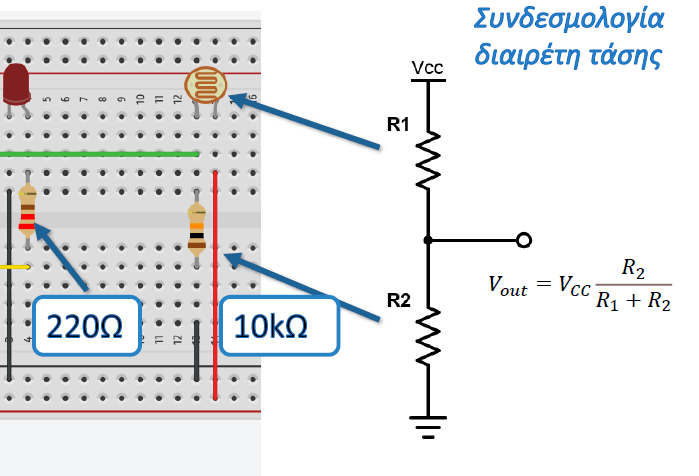
Σε αυτό το παράδειγμα μας δίνεται ο κώδικας ***8\_Light\_Sensor.ino***. Ο κώδικας αυτός είναι υπεύθυνος να διαβάζει είτε ψηφιακά είτε αναλογικά την έξοδο μιας φωτοαντίστασης και αναλόγως ανάβει ή σβήνει ένα ***LED***. Η συνδεσμολογία φαίνεται στο σχήμα



Η αναλογική λειτουργία του προγράμματος επιτρέπει την φωτεινότητα του ***LED*** να ανάβει και να σβήνει ανάλογα με την τάση που διαβάζεται από την φωτοαντίσταση. Έτσι, θα προσομοιώνει όλες τις τιμές της τάσης. Η ψηφιακή λειτουργία του προγράμματος επιτρέπει στο ***LED*** να ανάβει και να σβήνει μια φορά. Όταν εκπέμπεται φως στην αντίσταση τότε το ***LED*** θα ανάβει. Όταν δεν εκπέμπεται φως στην αντίσταση τότε το λαμπάκι δεν θα ανάψει.

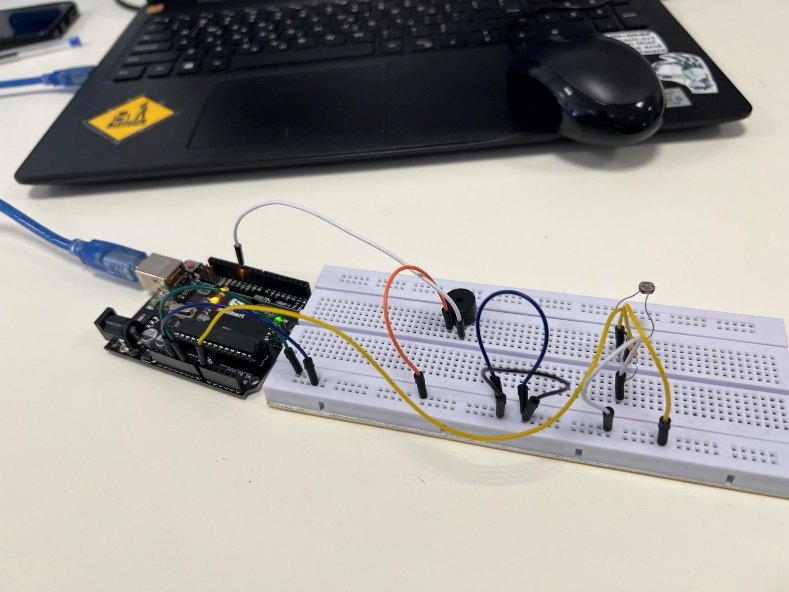
## **8.1 Ερώτημα 1**

Στο παρών κύκλωμα υπάρχει ένας διαιρέτης τάσης. Η ύπαρξη του διαιρέτη τάσης είναι να υποβαθμίζει την τάση εισόδου σε μία μικρότερη τιμή που θα εφαρμόζεται στο ***LED***. Επειδή το ***LED*** είναι μια δίοδος πρακτικά, σε μεγάλες εφαρμογές τάσης μπορεί να καταστραφεί.

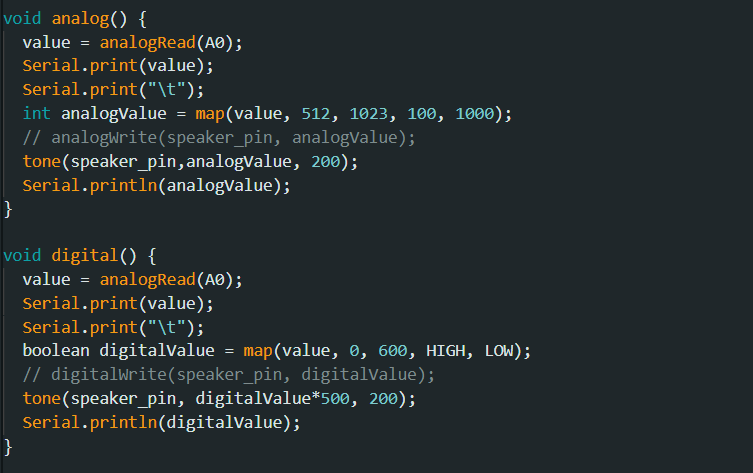


## **8.2 Ερώτημα 2**

Σε αυτό το ερώτημα είχαμε να τροποποιήσουμε τον δοθέν κώδικα και να τροποποιηθεί και το κύκλωμα της διάταξης, με την αφαίρεση του ***LED*** και την εισαγωγή ενός ***buzzer***. Το νέο κύκλωμα παρουσιάζεται στο παρακάτω σχήμα .



Γνωρίζουμε πως το ***buzzer*** λειτουργεί σε συχνότητες από 100 έως 1000. Οπότε θα τροποποιήσουμε την συνάρτηση και στην συνάρτηση και στην συνάρτηση . Έτσι, οι τιμές εξόδου που θα έχει η τιμή ***analogValue*** θα κυμαίνονται σε αυτό το εύρος. Στη συνέχεια με την εντολή θα εισάγουμε αυτή την τιμή από το Arduino στην έξοδο 11 που είναι η είσοδος του buzzer. Αυτό θα γίνει με την εντολή . Έτσι, θα εφαρμόζεται η αντίστοιχη συχνότητα στο buzzer με διάρκεια .



Συνεπώς, με την τροποποίηση του κυκλώματος και του κώδικα θα έχουμε την αντίστοιχη λειτουργία. Εάν επιλεχθεί ο χαρακτήρας από το Serial Monitor, τότε ανάλογα με την εκπομπή του φωτός στην φωτοαντίσταση, θα δέχεται το buzzer όλο το εύρος συχνοτήτων της συνάρτησης . Στην περίπτωση που έχουμε εισάγει την τιμή από το Serial Monitor, τότε το buzzer θα διαβάζει μία μόνο συγκεκριμένη συχνότητα από την φωτοαντίσταση και θα παράγει ήχο. Ένα παράδειγμα λειτουργίας φαίνεται και στο σχήμα που ακολουθεί.

