ΑΚΑΔΗΜΙΑ ΕΜΠΟΡΙΚΟΥ ΝΑΥΤΙΚΟΥ
ΜΑΚΕΔΟΝΙΑΣ
ΣΧΟΛΗ ΜΗΧΑΝΙΚΩΝ

ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ

ΘΕΜΑ :ΑΙΣΘΗΤΗΡΕΣ ΓΙΑ ARDUINO 2/5

ΣΠΟΥΔΑΣΤΗΣ :ΚΑΖΑΚΗΣ ΙΩΑΝΝΗΣ

ΕΠΙΒΛΕΠΟΥΣΑ
ΚΑΘΗΓΗΤΡΙΑ : ΠΕΡΙΒΟΛΗ ΠΑΣΧΑΛΙΝΑ

ΝΕΑ ΜΗΧΑΝΙΩΝΑ
2017

ΑΚΑΔΗΜΙΑ ΕΜΠΟΡΙΚΟΥ ΝΑΥΤΙΚΟΥ
ΜΑΚΕΔΟΝΙΑΣ
ΣΧΟΛΗ ΜΗΧΑΝΙΚΩΝ

ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ

ΘΕΜΑ :ΑΙΣΘΗΤΗΡΕΣ ΓΙΑ ARDUINO 2/5

ΣΠΟΥΔΑΣΤΗΣ :ΚΑΖΑΚΗΣ ΙΩΑΝΝΗΣ

ΑΜ : 5147

ΗΜΕΡΟΜΗΝΙΑ ΠΑΡΑΔΟΣΗΣ : ΙΟΥΝΙΟΣ 2017

Βεβαιώνεται η ολοκλήρωση της παραπάνω πτυχιακής εργασίας

Η καθηγήτρια

Περίληψη

Η πτυχιακή εργασία που παρουσιάζεται παρακάτω αποτελεί τμήμα ενός συλλογικού έργου που εκπονήθηκε από το εργαστήριο Συστημάτων Αυτομάτου Ελέγχου (Σ.Α.Ε.) της Σχολής Μηχανικών της Ακαδημίας Εμπορικού Ναυτικού Μακεδονίας. Το συλλογικό αυτό έργο είναι η καταγραφή, η διευθέτηση υλικών και η δημιουργία εφαρμογών σε επίπεδο κατασκευής και προγραμματισμού σε ότι αφορά την τεχνολογία Arduino. Η εργασία αποτελεί το δεύτερο μέρος από τα πέντε συνολικά μέρη του συνόλου του έργου. Ο στόχος της πτυχιακής είναι η οργάνωση των υλικών που υπάρχουν στο εργαστήριο των Σ.Α.Ε και αφορούν τμήμα των Arduino. Το εγχείρημα αυτό εστιάζει σε δυο άξονες. Ο πρώτος άξονας αφορά στη συλλογή πληροφοριών για τα υλικά Arduino του εργαστηρίου, όπως περιγραφή της λειτουργίας τους, κάποια τεχνικά χαρακτηριστικά τους και κώδικες προγραμματισμού. Ο δεύτερος άξονας αφορά στην διευθέτηση των υλικών Arduino, την τοποθέτηση τους μέσα σε ειδικά διαμορφωμένες θήκες και τέλος στην τοποθέτηση ετικετών με το εικονίδιο και το όνομα του κάθε υλικού ώστε αυτό να είναι άμεσα αναγνωρίσιμο. Ο σκοπός της πτυχιακής είναι να αποτελέσει σημαντικό εργαλείο στα χέρια των σπουδαστών και καθηγητών του εργαστήριου των Σ.Α.Ε. για την διδασκαλία του αντικειμένου. Παρακάτω στην πτυχιακή παρατίθενται οι περιγραφές και τα τεχνικά χαρακτηριστικά των υλικών Arduino καθώς και οπτικό υλικό αυτών.

Abstract

The dissertation presented below is part of a collaborative work done by the Automatic Control Systems Laboratory (SAE) of the School of Engineering of the Academy of Commercial Nautical Macedonia. This collective project is the recording, the arrangement of materials and the creation of applications at the level of construction and programming in terms of Arduino technology. Work is the second part of the five overall parts of the project as a whole. The aim of the dissertation is the organization of the materials in the laboratory of SAE and the part of Arduino. This project focuses on two axes. The first pillar concerns the collection of information about Arduino materials in the lab, including a description of their function, some technical features and programming codes. The second axis concerns the arrangement of Arduino materials, their placement in specially shaped cases and finally the placement of labels with the icon and the name of each material so that it is instantly recognizable. The purpose of the diploma thesis is to be an important tool in the hands of SAE students and professors, for teaching the subject. The following are the descriptions and technical specifications of Arduino materials and their optical materials.

Πρόλογος

Το Arduino είναι ένας μικροελεγκτής ο οποίος περιλαμβάνει ένα chip ATmega. Με λίγα λόγια διαθέτει εισόδους και εξόδους που αντιδρούν βάση του προγραμματισμού που κάναμε και που φορτώσαμε στο chip με τη βοήθεια του υπολογιστή. Το ολοκληρωμένο περιβάλλον ανάπτυξης (IDE) του Arduino είναι μία εφαρμογή γραμμένη σε Java, που λειτουργεί σε πολλές πλατφόρμες, και προέρχεται από το IDE για τη γλώσσα προγραμματισμού Processing και το σχέδιο Wiring. Έχει σχεδιαστεί για να εισάγει τον προγραμματισμό σε όσους δεν είναι εξοικειωμένοι με την ανάπτυξη λογισμικού. Περιλαμβάνει ένα πρόγραμμα επεξεργασίας κώδικα με χαρακτηριστικά όπως είναι η επισήμανση σύνταξης και ο συνδυασμός αγκύλων και είναι επίσης σε θέση να μεταγλωττίζει και να φορτώνει προγράμματα στην πλακέτα με ένα μόνο κλικ. Δεν υπάρχει συνήθως καμία ανάγκη να επεξεργασίας των αρχείων make . Ένα πρόγραμμα ή κώδικας που γράφτηκε για Arduino ονομάζεται σκίτσο (sketch).

ΚΕΦΑΛΑΙΟ 1 : ΕΙΣΑΓΩΓΗ ΣΤΟ ARDUINO

1.1 Γενικές πληροφορίες

Το Arduino είναι μια «ανοικτού κώδικα» πλατφόρμα «προτυποποίησης» ηλεκτρονικών, βασισμένη σε ευέλικτο και εύκολο στη χρήση hardware και software που προορίζεται για οποιονδήποτε έχει λίγη προγραμματιστική εμπειρία, στοιχειώδεις γνώσεις ηλεκτρονικών και ενδιαφέρεται να δημιουργήσει διαδραστικά αντικείμενα ή περιβάλλοντα.

Στην ουσία, πρόκειται για ένα ηλεκτρονικό κύκλωμα που βασίζεται στον μικροελεγκτή ATmega της Atmel και του οποίου όλα τα σχέδια, καθώς και το software που χρειάζεται για την λειτουργία του, διανέμονται μαζί με την αγορά του kit . Αφού κατασκευαστεί, μπορεί να συμπεριφερθεί σαν ένας μικροσκοπικός υπολογιστής, αφού ο χρήστης μπορεί να συνδέσει επάνω του πολλαπλές μονάδες εισόδου/εξόδου και να προγραμματίσει τον μικροελεγκτή να δέχεται δεδομένα από τις μονάδες εισόδου, να τα επεξεργάζεται και να στέλνει κατάλληλες εντολές στις μονάδες εξόδου. Λειτουργικά το Arduino μοιάζει πολύ με το NXT Brick των LegoMindstorms NXT. Άλλωστε η ρομποτική είναι μια από τις πολλές εφαρμογές στις οποίες το Arduino διαπρέπει.

Το Arduino κυκλοφορεί σε διάφορες εκδόσεις, επίσημες και ανεπίσημες. Από τις επίσημες εκδόσεις αυτές που ξεχωρίζουν για την αξιοπιστία είναι οι (Duemilanove, Diecimila, Nano, Mega, Bluetooth, LilyPad, Mini, Mini USB, Pro, ProMini, Serial και Serial SS) από τις οποίες συνιστάται κυρίως η έκδοση ArduinoDuemilanove ή τουλάχιστον των Diecimila ή Mega ,οι οποίες διαθέτουν υποδοχή USB και είναι συμβατές με τα shield. Όλες οι εκδόσεις εκτός από το ίδιο το Arduino, περιέχουν διάφορα άλλα εξαρτήματα και εργαλεία που μπορεί να χρειαστείτε για τις πρώτες σας εφαρμογές (όπως το απαραίτητο καλώδιο USB για την σύνδεση με τον υπολογιστή, ράστερ, καλώδια, LED, διακόπτες, ποτενσιόμετρα, αντιστάσεις, διόδους, τρανζίστορ κ.λπ.).

Στην συνέχεια θα δώσουμε περισσότερες πληροφορίες όσον αφορά τα βασικά κατασκευαστικά στοιχεία ενός Arduino Starter Kit,αναλύοντας τις έννοιες του μικροελεκτή (ο οποίος αποτελέι την καρδία του Arduino) , των ενσωματωμένων κουμπιών Led ,είσοδοι-έξοδοι της πλακέτας ,της τροφοδοσίας, και του Arduino IDE

με την με τον υπολογιστή .Σκοπός αυτής της ανάλυσης μας είναι να δούμε τα μέρη από τα οποία αποτελείται μια πλακέτα και να εμβαθύνουμε τις γνώσεις μας σχετικά με τον τρόπο λειτουργίας ενός Arduino.

1.2 Μικροελεγκτής – η καρδιά του Arduino

Το Arduino βασίζεται στον ATmega328, έναν 8-bit RISC μικροελεγκτή, τον οποίο χρονίζει στα 16MHz. Ο ATmega328 διαθέτει ενσωματωμένη μνήμη τριών τύπων:

- 2Kb μνήμης SRAM που είναι η ωφέλιμη μνήμη που μπορούν να χρησιμοποιήσουν τα προγράμματά σας για να αποθηκεύουν μεταβλητές, πίνακες κ.λπ. κατά το runtime. Όπως και σε έναν υπολογιστή, αυτή η μνήμη χάνει τα δεδομένα της όταν η παροχή ρεύματος στο Arduino σταματήσει ή αν γίνει reset.
- 1Kb μνήμης EEPROM η οποία μπορεί να χρησιμοποιηθεί για «ωμή» εγγραφή/ανάγνωση δεδομένων (χωρίς datatype) ανά byte από τα προγράμματά σας κατά το runtime. Σε αντίθεση με την SRAM, η EEPROM δεν χάνει τα περιεχόμενά της με απώλεια τροφοδοσίας ή reset οπότε είναι το ανάλογο του σκληρού δίσκου.
- 32Kb μνήμης Flash, από τα οποία τα 2Kb χρησιμοποιούνται από το firmware του Arduino που έχει εγκαταστήσει ήδη ο κατασκευαστής του. Το firmware αυτό που στην ορολογία του Arduino ονομάζεται bootloader είναι αναγκαίο για την εγκατάσταση των δικών σας προγραμμάτων στον μικροελεγκτή μέσω της θύρας USB, χωρίς δηλαδή να χρειάζεται εξωτερικός hardwareprogrammer. Τα υπόλοιπα 30Kb της μνήμης Flash χρησιμοποιούνται για την αποθήκευση αυτών ακριβώς των προγραμμάτων, αφού πρώτα μεταγλωττιστούν στον υπολογιστή σας. Η μνήμη Flash, όπως και η EEPROM δεν χάνει τα περιεχόμενά της με απώλεια τροφοδοσίας ή reset. Επίσης, ενώ η μνήμη Flash υπό κανονικές συνθήκες δεν προορίζεται για χρήση runtime μέσα από τα προγράμματά σας, λόγω της μικρής συνολικής μνήμης που είναι διαθέσιμη σε αυτά (2Kb SRAM + 1Kb EEPROM), έχει σχεδιαστεί μια βιβλιοθήκη που επιτρέπει την χρήση όσου χώρου περισσεύει (30Kb μείον το μέγεθος του προγράμματός σας σε μεταγλωττισμένη μορφή).

1.3 Είσοδοι – Έξοδοι

Καταρχήν το Arduino διαθέτει σειριακό interface. Ο μικροελεγκτήςATmega υποστηρίζει σειριακή επικοινωνία, την οποία το Arduino προωθεί μέσα από έναν ελεγκτή Serial-over-USB ώστε να συνδέεται με τον υπολογιστή μέσω USB. Η σύνδεση αυτή χρησιμοποιείται για την μεταφορά των προγραμμάτων που σχεδιάζονται από τον υπολογιστή στο Arduino αλλά και για αμφίδρομη επικοινωνία του Arduino με τον υπολογιστή μέσα από το πρόγραμμα την ώρα που εκτελείται. Επιπλέον, στην πάνω πλευρά του Arduino βρίσκονται 14 θηλυκά pin, αριθμημένα από 0 ως 13, που μπορούν να λειτουργήσουν ως ψηφιακές είσοδοι και έξοδοι. Λειτουργούν στα 5V και καθένα μπορεί να παρέχει ή να δεχτεί το πολύ 40mA.

Ως ψηφιακή έξοδος, ένα από αυτά τα pin μπορεί να τεθεί από το πρόγραμμά σας σε κατάσταση HIGH ή LOW, οπότε το Arduino θα ξέρει αν πρέπει να διοχετεύσει ή όχι ρεύμα στο συγκεκριμένο pin. Με αυτόν τον τρόπο μπορείτε λόγου χάρη να ανάψετε και να σβήσετε ένα LED που έχετε συνδέσει στο συγκεκριμένο pin. Αν πάλι ρυθμίσετε ένα από αυτά τα pin ως ψηφιακή είσοδο μέσα από το πρόγραμμά σας, μπορείτε με την κατάλληλη εντολή να διαβάσετε την κατάστασή του (HIGH ή LOW) ανάλογα με το αν η εξωτερική συσκευή που έχετε συνδέσει σε αυτό το pin διοχετεύει ή όχι ρεύμα στο pin (με αυτόν τον τρόπο λόγου χάρη μπορεί να «διαβαστεί» η κατάσταση ενός διακόπτη).

Μερικά από αυτά τα 14 pin, εκτός από ψηφιακές είσοδοι/έξοδοι έχουν και δεύτερη λειτουργία. Συγκεκριμένα:

- Τα pin 0 και 1 λειτουργούν ως RX και TX της σειριακής όταν το πρόγραμμά σας ενεργοποιεί την σειριακή θύρα. Έτσι, όταν λόγου χάρη το πρόγραμμά σας στέλνει δεδομένα στην σειριακή, αυτά προωθούνται και στην θύρα USB μέσω του ελεγκτή Serial-Over-USB αλλά και στο pin 0 για να τα διαβάσει ενδεχομένως μια άλλη συσκευή (π.χ. ένα δεύτερο Arduino στο δικό του pin 1). Αυτό φυσικά σημαίνει ότι αν στο πρόγραμμά ενεργοποιηθεί το σειριακό interface, χάνονται 2 ψηφιακές εισόδοι/εξόδοι.
- Τα pin 2 και 3 λειτουργούν και ως εξωτερικά interrupt (interrupt 0 και 1 αντίστοιχα). Με άλλα λόγια, ρυθμίζονται μέσα από το πρόγραμμά ώστε να λειτουργούν αποκλειστικά ως ψηφιακές είσοδοι στις οποίες όταν συμβαίνουν συγκεκριμένες αλλαγές, η κανονική ροή του προγράμματος σταματάει

άμεσα και εκτελείται μια συγκεκριμένη συνάρτηση. Τα εξωτερικά interrupt είναι ιδιαίτερα χρήσιμα σε εφαρμογές που απαιτούν συγχρονισμό μεγάλης ακρίβειας.

- Τα pin 3, 5, 6, 9, 10 και 11 μπορούν να λειτουργήσουν και ως ψευδοαναλογικές έξοδοι με το σύστημα PWM (PulseWidthModulation), δηλαδή το ίδιο σύστημα που διαθέτουν οι μητρικές των υπολογιστών για να ελέγχουν τις ταχύτητες των ανεμιστήρων. Έτσι, μπορεί να συνδεθεί λόγου χάρη ένα LED σε κάποιο από αυτά τα pin και να ελεγχθεί πλήρως η φωτεινότητά του με ανάλυση 8bit (256 καταστάσεις από 0-σβηστό ως 255-πλήρως αναμμένο) αντί να υπάρχει απλά η δυνατότητα αναμμένο-σβηστό που παρέχουν οι υπόλοιπές ψηφιακές έξοδοι. Είναι σημαντικό να γίνει σαφές ότι το PWM δεν είναι πραγματικά αναλογικό σύστημα και ότι θέτοντας στην έξοδο την τιμή 127, δεν σημαίνει ότι η έξοδος θα δίνει 2.5V αντί της κανονικής τιμής των 5V, αλλά ότι θα δίνει ένα παλμό που θα εναλλάσσεται με μεγάλη συχνότητα και για ίσους χρόνους μεταξύ των τιμών 0 και 5V.

Στην κάτω πλευρά του Arduino, με τη σήμανση ANALOG IN, υπάρχει μια ακόμη σειρά από 6 pin, αριθμημένα από το 0 ως το 5. Το καθένα από αυτά λειτουργεί ως αναλογική είσοδος κάνοντας χρήση του ADC (AnalogtoDigitalConverter) που είναι ενσωματωμένο στον μικροελεγκτή. Για παράδειγμα, μπορούμε να τροφοδοτήσουμε ένα από αυτά με μια τάση την οποία μπορούμε να τη μεταβάλουμε με ένα ποτενσιόμετρο από 0V ως μια τάση αναφοράς Vref η οποία, αν δεν κάνουμε κάποια αλλαγή, είναι προρυθμισμένη στα 5V. Τότε, μέσα από το πρόγραμμά μας μπορούμε να «διαβάσουμε» την τιμή του pin ως ένα ακέραιο αριθμό ανάλυσης 10-bit, από 0 (όταν η τάση στο pin είναι 0V) μέχρι 1023 (όταν η τάση στο pin είναι 5V). Η τάση αναφοράς μπορεί να ρυθμιστεί με μια εντολή στο 1.1V, ή σε όποια τάση επιθυμούμε (μεταξύ 2 και 5V) τροφοδοτώντας εξωτερικά με αυτή την τάση το pin με την σήμανση AREF που βρίσκεται στην απέναντι πλευρά της πλακέτας. Έτσι, αν τροφοδοτήσουμε το pin AREF με 3.3V και στην συνέχεια δοκιμάσουμε να διαβάσουμε κάποιο pin αναλογικής εισόδου στο οποίο εφαρμόζουμε τάση 1.65V, το Arduino θα μας επιστρέψει την τιμή 512. Τέλος, καθένα από τα 6 αυτά pin, με κατάλληλη εντολή μέσα από το πρόγραμμα μπορεί να μετατραπεί σε ψηφιακό pin εισόδου/εξόδου όπως τα 14 που βρίσκονται στην απέναντι πλευρά και τα οποία

περιγράφηκαν πριν. Σε αυτή την περίπτωση τα pin μετονομάζονται από 0~5 σε 14~19 αντίστοιχα.

1.4 Τροφοδοσία

Το Arduino μπορεί να τροφοδοτηθεί με ρεύμα είτε από τον υπολογιστή μέσω της σύνδεσης USB, είτε από εξωτερική τροφοδοσία που παρέχεται μέσω μιας υποδοχής φις των 2.1mm (θετικός πόλος στο κέντρο) και βρίσκεται στην κάτω-αριστερή γωνία του Arduino.Για να μην υπάρχουν προβλήματα, η εξωτερική τροφοδοσία πρέπει να είναι από 7 ως 12V και μπορεί να προέρχεται από ένα κοινό μετασχηματιστή του εμπορίου, από μπαταρίες ή οποιαδήποτε άλλη πηγή DC. Δίπλα από τα pin αναλογικής εισόδου, υπάρχει μια ακόμα συστοιχία από 6 pin με την σήμανση POWER.

Η λειτουργία του καθενός έχει ως εξής:

- Το πρώτο, με την ένδειξη RESET, όταν γειωθεί (σε οποιοδήποτε από τα 3 pin με την ένδειξη GND που υπάρχουν στο Arduino) έχει ως αποτέλεσμα την επανεκκίνηση του Arduino.
- Το δεύτερο, με την ένδειξη 3.3V, μπορεί να τροφοδοτήσει τα εξαρτήματά με τάση 3.3V. Η τάση αυτή δεν προέρχεται από την εξωτερική τροφοδοσία αλλά παράγεται από τον ελεγκτή Serial-over-USB και έτσι η μέγιστη ένταση που μπορεί να παρέχει είναι μόλις 50mA.
- Το τρίτο, με την ένδειξη 5V, μπορεί να τροφοδοτήσει τα εξαρτήματά με τάση 5V. Ανάλογα με τον τρόπο τροφοδοσίας του ίδιου του Arduino, η τάση αυτή προέρχεται είτε άμεσα από την θύρα USB (που ούτως ή άλλως λειτουργεί στα 5V), είτε από την εξωτερική τροφοδοσία αφού αυτή περάσει από ένα ρυθμιστή τάσης για να την «φέρει» στα 5V.

Το τέταρτο και το πέμπτο pin, με την ένδειξη GND, είναι φυσικά γειώσεις.

- Το έκτο και τελευταίο pin, με την ένδειξη Vin έχει διπλό ρόλο. Σε συνδυασμό με το pin γείωσης δίπλα του, μπορεί να λειτουργήσει ως μέθοδος εξωτερικής τροφοδοσίας του Arduino, στην περίπτωση που δεν βολεύει να χρησιμοποιήθεί η υποδοχή του φις των 2.1mm. Αν όμως υπάρχει ήδη συνδεδεμένη εξωτερική τροφοδοσία μέσω του φις, μπορούμε να χρησιμοποιήσουμε αυτό το pin για να τροφοδοτήσουμε εξαρτήματα με την

πλήρη τάση της εξωτερικής τροφοδοσίας (7~12V), πριν αυτή περάσει από τον ρυθμιστή τάσης όπως γίνεται με το pin των 5V.

1.5 Ενσωματωμένα κουμπιά και LED

Πάνω στην πλακέτα του Arduino υπάρχει ένας διακόπτης micro-switch και 4 μικροσκοπικά LED επιφανειακής στήριξης. Η λειτουργία του διακόπτη (που έχει την σήμανση RESET) και του ενός LED με την σήμανση POWER είναι μάλλον προφανής. Τα δύο LED με τις σημάνσεις TX και RX, χρησιμοποιούνται ως ένδειξη λειτουργίας του σειριακού interface, καθώς ανάβουν όταν το Arduino στέλνει ή λαμβάνει (αντίστοιχα) δεδομένα μέσω USB. Σημειώστε ότι τα LED αυτά ελέγχονται από τον ελεγκτή Serial-over-USB και συνεπώς δεν λειτουργούν όταν η σειριακή επικοινωνία γίνεται αποκλειστικά μέσω των ψηφιακών pin 0 και 1.

Τέλος, υπάρχει το LED με την σήμανση L. Η βασική δοκιμή λειτουργίας του Arduino είναι να του αναθέσετε να αναβοσβήνει ένα LED. Για να μπορείτε να το κάνετε αυτό από την πρώτη στιγμή, χωρίς να συνδέσετε τίποτα πάνω στο Arduino, οι κατασκευαστές του σκέφτηκαν να ενσωματώσουν ένα LED στην πλακέτα, το οποίο σύνδεσαν στο ψηφιακό pin 13. Έτσι, ακόμα και αν δεν έχουμε συνδέσει τίποτα πάνω στο φυσικό pin 13, αναθέτοντάς του την τιμή HIGH μέσα από το πρόγραμμά μας, θα ανάψει αυτό το ενσωματωμένο LED

1.6 Arduino IDE και σύνδεση με τον υπολογιστή

Ότι χρειάζεται για την διαχείριση του Arduino από τον υπολογιστή μας το παρέχει το ArduinoIDE.ToArduino IDE είναι βασισμένο σε Java και συγκεκριμένα παρέχει ένα πρακτικό περιβάλλον για την συγγραφή των προγραμμάτων μας (τα οποία ονομάζονται sketch στην ορολογία του Arduino) με συντακτική χρωματική σήμανση, αρκετά έτοιμα παραδείγματα, μερικές έτοιμες βιβλιοθήκες για προέκταση της γλώσσας και για εύκολο χειρισμό μέσα από τον κώδικά των εξαρτήματα που συνδέονται στο Arduino, τον compiler για την μεταγλώττιση των sketch, ένα serial monitor που παρακολουθεί τις επικοινωνίες της σειριακής (USB), αναλαμβάνει να στείλει αλφαριθμητικά της επιλογής στο Arduino μέσω αυτής και είναι ιδιαίτερα χρήσιμο για το debugging των sketch για να ανέβει το μεταγλωττισμένο sketch στο Arduino. Για τα δύο τελευταία χαρακτηριστικά βέβαια, το Arduino πρέπει να έχει συνδεθεί σε μια από τις θύρες USB του υπολογιστή και, λόγω του ελεγκτή

Sirial-over-USB , θα πρέπει να αναγνωριστεί από το λειτουργικό σύστημα ως εικονική σειριακή θύρα.

Για την σύνδεση θα χρειαστεί ένα καλώδιο USB από Type A σε Type B, όπως αυτό των εκτυπωτών. Για την αναγνώριση από το λειτουργικό θα χρειαστεί να εγκατασταθεί ο οδηγός του FTDI chip (δηλαδή του ελεγκτή Serial-over-USB), ο οποίος υπάρχει στον φάκελο drivers του Arduino IDE.

2 ΚΕΦΑΛΑΙΟ_Ο ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΣ ΤΟΥ ARDUINO

2.1 Γενικές πληροφορίες

Η γλώσσα του Arduino βασίζεται στη γλώσσα Wiring, μια παραλλαγή C/C++ για μικροελεγκτές αρχιτεκτονικής AVR όπως ο ATmega, και υποστηρίζει όλες τις βασικές δομές της C καθώς και μερικά χαρακτηριστικά της C++. Για compiler χρησιμοποιείται ο AVR gcc και ως βασική βιβλιοθήκη C χρησιμοποιείται η AVR libc.

Λόγω της καταγωγής της από την C, στην γλώσσα του Arduino μπορείτε να χρησιμοποιήσετε ουσιαστικά τις ίδιες βασικές εντολές και συναρτήσεις, με την ίδια σύνταξη, τους ίδιους τύπων δεδομένων και τους ίδιους τελεστές όπως και στην C. Πέρα από αυτές όμως, υπάρχουν κάποιες ειδικές εντολές, συναρτήσεις και σταθερές που βοηθούν για την διαχείριση του ειδικού hardware του Arduino.

Η βασική ρουτίνα setup() εκτελείται μια φορά μόνο κατά την εκκίνηση του προγράμματος ενώ η βασική ρουτίνα loop() περιέχει τον βασικό κορμό του προγράμματος και η εκτέλεσή της επαναλαμβάνεται συνέχεια σαν ένας βρόγχος while(true). Παρακάτω θα αναλύσουμε επαρκώς την δομή του προγράμματος σε ξεχωριστό κεφάλαιο.

2.2 Η Δομή του προγράμματος

Ένα τυπικό πρόγραμμα Arduino έχει την παρακάτω δομή:

```
//δήλωση μεταβλητών
voidsetup ()
        {
//αρχικοποιήσεις
        }
voidloop ()
        {
        //Κώδικας
       }
```

Υπάρχουν δυο ειδικές συναρτήσεις που είναι μέρος του κάθε sketch του Arduino οι οποίες είναι η setup() και η loop(). Η setup() καλείται μια φορά, όταν το

sketch ξεκινά ή όποτε κάνει επαναφορά (reset) η πλατφόρμα Arduino. Κυρίως, σε αυτήν γίνονται οι αρχικοποιήσεις των μεταβλητών, η ρύθμιση της κατάστασης των ακίδων (pins) και η προετοιμασία των βιβλιοθηκών. Αντιθέτως, η συνάρτηση loop() καλείται ξανά και ξανά επιτρέποντας έτσι στο πρόγραμμα να ανταποκριθεί σε εξωτερικά ερεθίσματα. Και οιδυο συναρτήσεις πρέπει να περιλαμβάνονται στο sketch, ακόμα και αν δεν περιέχουν κάτι και να είναι κενές.

2.3 Ψηφιακές ακίδες (Digitalpins)

Οι ακίδες αυτές στο Arduino μπορούν να ρυθμιστούν είτε ως είσοδοι είτε ως έξοδοι, όμως από προεπιλογή είναι ρυθμισμένες ως είσοδοι. Επίσης αξίζει να σημειωθεί, ότι η πλειοψηφία των αναλογικών ακίδων του Arduino (Atmega), μπορεί να ρυθμιστεί και να χρησιμοποιηθεί, με τον ίδιο ακριβώς τρόπο όπως οι ψηφιακές ακίδες.

Οι συναρτήσεις ψηφιακής εισόδου και εξόδου είναι οι παρακάτω:

- **pinMode():** Ρυθμίζει τη συγκεκριμένη ακίδα να συμπεριφέρεται ως είσοδος/ έξοδος

Σύνταξη: `pinMode(pin, mode)\`

- **digitalWrite():** Γράφει μια υψηλή (HIGH) ή μια χαμηλή (LOW) τιμή σε μια ψηφιακή ακίδα. Αν η ακίδα έχει ρυθμιστεί ως έξοδος με την συνάρτηση pinMode(), τότε η τάση της θα καθορίσει στην αντίστοιχη τιμή: 5V για HIGH και 0V για LOW. Αν η ακίδα έχει ρυθμιστεί ως είσοδος, γράφοντας HIGH στην συνάρτηση digitalWrite() θα ενεργοποιήσει μια εσωτερική pullup-αντίσταση των 20 K ενώ γράφοντας LOW θα την απενεργοποιήσει.

Σύνταξη: `digitalWrite(pin,value)`

- **digitalRead():**Διαβάζει την τιμή από μια συγκεκριμένη ψηφιακή ακίδα, που είναι είτε HIGH είτε LOW.

Σύνταξη: `digitalRead(pin)`

2.4 Αναλογικές ακίδες εισόδου (Analog inputp ins)

Οι ελεγκτές Atmega που χρησιμοποιούνται για την πλατφόρμα Arduino περιέχουν έναν ενσωματωμένο αναλογικό-σε-ψηφιακό μετατροπέα 6 καναλιών. Ο μετατροπέας διαθέτει ανάλυση 10 bit, επιστρέφοντας ακέραιους από 0 έως 1023. Ενώ η κύρια λειτουργία της αναλογικής ακίδας για τους περισσότερους χρήστες Arduino είναι να διαβάζει αναλογικούς αισθητήρες, οι αναλογικές ακίδες έχουν επίσης όλες τις λειτουργίες των γενικών ακίδων εισόδου/εξόδου. Οι συναρτήσεις αναλογικής εισόδου και εξόδου είναι οι παρακάτω:

- **analogWrite():**Γράφει μια αναλογική τιμή (PWM κύμα) σε μια ακίδα. Μπορεί ναχρησιμοποιηθεί για παράδειγμα να ανάψει ένα LED σε διάφορες φωτεινότητες ήνα οδηγήσει ένα κινητήρα σε διάφορες ταχύτητες. Μετά από μια κλήση της analogWrite(), η ακίδα θα δημιουργήσει ένα σταθερό τετραγωνικό κύμα τουκαθορισμένου κύκλου λειτουργίας μέχρι την επόμενη κλήση της analogWrite() (ήμια κλήση της digitalWrite() ή digitalRead() για την ίδια ακίδα). Η συχνότητα τουσήματος PWM είναι περίπου 490 Hz. Στις περισσότερες πλατφόρμες Arduino ησυνάρτηση αυτή λειτουργεί στις ακίδες 3, 5, 6, 9, 10, 11.

  Σύνταξη: `analogWrite(pin, value)`

- **analogRead():**Διαβάζει την τιμή από την καθορισμένη αναλογική ακίδα.

  Σύνταξη: `analogRead(pin)`

2.5 Βασικές δομές και λειτουργίες προγραμματισμού

Παρακάτω, ακολουθούν μερικές από τις πιο βασικές δομές και λειτουργίες που μπορεί να αξιοποιηθούν ως εργαλεία κατά την συγγραφή ενός προγράμματος Arduino :

ADVANCED I/O (Προηγμένες συναρτήσεις εισόδου και εξόδου)

- **tone()** (παράγει ένα τετραγωνικό σήμα ορισμένης συχνότητας)

- **noTone()** (διακόπτει την παραγωγή μιας συχνότητας σε ένα καθορισμένο πείρο)

  Σύνταξη: noTone(PIN);

- **shiftOut()** (ολισθαίνει τα ψηφία μιας τιμής σε μία επαφή εξόδου)

  Σύνταξη: shiftOut(dataPin,clockPin,bitOrder,data);
  shiftOut(dataPin,clockPin,bitOrder,data,count);
  shiftOut(dataPin,clockPin,bitOrder,data,count,delayTime);

- **pulseIn()** (επιστρέφει την διάρκεια σε μs ενός παλμού HIGH ή LOW)

  Σύνταξη: pulseIn(PIN, value);
  pulseIn(PIN, value, timeout);

TIME (Συναρτήσεις χρόνου)

- **millis()** (Επιστρέφει την διάρκεια εκτέλεσης του προγράμματος σε ms)

  Σύνταξη : millis();

- **micros()** (Επιστρέφει την διάρκεια εκτέλεσης του προγράμματος σε μs.)

  Σύνταξη : micros();

- **delay()** (Επέρχεται παύση του προγράμματος - η διάρκεια παύσης δίδεται σε ms)

  Σύνταξη: delay(milliseconds);

- **delayMicroseconds()** (Επέρχεται παύση του προγράμματος - η διάρκεια παύσης δίδεται σε μs)

  Σύνταξη : delayMicroseconds(microseconds);

BITS AND BYTES (Συναρτήσεις επεξεργασίας δυαδικών αριθμών)

- **lowByte()** (επιστρέφει το χαμηλό(το δεξιότερο) byte μίας μεταβλητής)

  Σύνταξη: lowByte(wordValue)

- **highByte()** (επιστρέφει το υψηλότερο(δηλαδή το αριστερότερο) byte μίας μεταβλητής σε 2byte(16bit))

  Σύνταξη: highByte(wordValue)

- **bitRead()** (διαβάζει ένα συγκεκριμένο ψηφίο μίας μεταβλητής)

  Σύνταξη: var=bitRead(byteValue,n)

- **bitWrite()** (γράφει σε ένα συγκεκριμένο ψηφίο μιας μεταβλητής)

  Σύνταξη: var=bitWrite(byteVal,n,bitVal)

- **bitSet()** (θέτει την τιμή 1 σε ένα καθορισμένο ψηφίο μίας μεταβλητής)

  Σύνταξη: var=bitSet(byteVal,n)

- **bitClear()** (θέτει την τιμή 0 σε ένα καθορισμένο ψηφίο μιας μεταβλητής)

  Σύνταξη: var=bitClear(byteValue,n)

- **bit()** (υπολογίζει ενα συγκεκριμένο bit με βάση το 2)

  Σύνταξη: bit(n)

EXTERNAL INTERRUPTS

(Συναρτήσεις χρήσης ρουτινών εξυπηρέτησης διακοπών)

- **attachInterrupt()** (ενεργοποιεί μία ρουτίνα εξυπηρέτησης διακοπής υπηρεσίας(ISR)

  Σύνταξη: attachInterrupt(digitalPinToInterrupt(pin),ISR,mode);
  attachInterrupt(interrupt,function,mode)
  attachInterrupt(pin,ISR,mode)(Arduino Due only)

- **detachInterrupt()** (απενεργοποιεί μία ρουτίνα εξυπηρέτησης διακοπής)

  Σύνταξη: detachInterrupt(interrupt)

  detachInterrupt(pin)Arduino Due only)

INTERRUPTS

(Συναρτήσεις ενεργοποίησης και απενεργοποίησης διακοπών)

- **interrupts()** (ενεργοποιεί τα σήματα διακοπής)

  Σύνταξη: interrupts()

- **noInterrupts()** (απενεργοποιεί τα σήματα διακοπής)

  Σύνταξη: nointerrupts()

DATA TYPES (Τύποι δεδομένων)

- **boolean** (λογική δυαδική τιμή -> true or false)

  Σύνταξη: booleanvar

  booleanvar=booleanvalue

- **char** (προσημασμένος χαρακτήρας 8 ψηφίων)

  Σύνταξη: charvar

  Charvar =value

- **unsignedchar** (μη προσημασμένος χαρακτήρας 8 ψηφίων)

  Σύνταξη: unsigned char var

  Unsigned char var =value

- **byte** (μη προσημασμένος χαρακτήρας 8 ψηφίων)

  Σύνταξη: byte var

  bytevar = value

- **int** (προσημασμένος ακέραιος αριθμός 16 ψηφίων)

  Σύνταξη : intvar

  intvar = value

- **unsignedint**(μη προσημασμένος ακέραιος αριθμός 16 ψηφίων)

  Σύνταξη: unsigned intvar

  unsignedintvar = value

- **word** (μη προσημασμένος ακέραιος αριθμός 16 ψηφίων)

  Σύνταξη: word var

  wordvar = value

- **long** (προσημασμένος ακέραιος αριθμός 32 ψηφίων)

  Σύνταξη: long var

  longvar = value

- **unsignedlong** (μη προσημασμένος ακέραιος αριθμός 32 ψηφίων)

  Σύνταξη: unsigned long var

  unsigned long var = value

- **float, double**(αριθμός κινητής υποδιαστολής απλής ακρίβειας)

  Σύνταξη: float foo

  float foo = value

- **String** (αντικείμενο αλφαριθμητικού με χρήσιμες μεθόδους)

  Σύνταξη: String(val)String(val, base)

CONVERSION (Συναρτήσεις μετατροπής τύπων)

- **char(), byte()**

- **int(), word(), long()**

- **float(), double()**

CONTROL STRUCTURES (Δομές ελέγχου ροής)

- **if** (δομή ελέγχου μίας συνθήκης)

  Σύνταξη: if (expression) {
  statements
  }

- **if ... else** (δομή ελέγχου πολλαπλών συνθηκών)

  Σύνταξη: if (expression) {
  statements1
  }else{
  statements2
  }

- **for** (δομή επαναληπτικού ελέγχου συνθήκης)

  Σύνταξη: for(initialization;test;increment) {
  //statements
  }

- **while** (δομή επαναληπτικού ελέγχου συνθήκης)

  Σύνταξη: while(expression){
  //statemen }

- **do ... while** (δομή επαναληπτικού ελέγχου συνθήκης)

  Σύνταξη: do{
  //statement
  }while(test condition);

- **switch ... case** (δομή ελέγχου περιπτώσεων)

  Σύνταξη switch (expression) {
  case label:
  // statements
  break;
  case label:

```
// statements
break;
default:
// statements
}
```

- **break** (εντολή διακοπής μιας επαναληπτικής δομής)

  Σύνταξη: `break;`

- **continue** (εντολή παράλειψης της τρέχουσας επανάληψης)

  Σύνταξη: `continue;`

- **return** (εντολή επιστροφής από μία συνάρτηση)

  Σύνταξη: `return [value];`

- **goto** (εντολή μετάβασης σε κάποιο σημείο του κώδικα)

  Σύνταξη:
  ```
  label: //some name
  goto label; //jump to label
  ```

COMPARISON OPERATORS (Τελεστές σύγκρισης)

- == (ισότητα)
- **!**=(ανισότητα)
- <(μικρότερο)
- >(μεγαλύτερο)
- <=(μικρότερο ή ίσο)
- >= (μεγαλύτερο ή ίσο)

POINTER ACCESS OPERATORS (Τελεστέςδεικτών)

- * (τελεστής απόκτησης περιεχομένου)
- **&**(τελεστής απόκτησης διεύθυνσης)

MATHANDTRIGONOMETRY

(Μαθηματικές και Τριγωνομετρικές συναρτήσεις)

- **max()** (βρίσκει τον μεγαλύτερο ανάμεσα σε δύο αριθμούς)

  Σύνταξη: max(x,y)

- **min()** (βρίσκει τον μικρότερο ανάμεσα σε δύο αριθμούς)

  Σύνταξη: min(x, y)

- **abs()** (επιστρέφει την απόλυτη τιμή ενός αριθμού)

  Σύνταξη: abs(x)

- **constrain()** (ελέγχει για υπερχείλιση ή υποχείλιση ορίων)

  Σύνταξη: constrain(x, a, b)

- **map()** (πραγματοποιεί γραμμικό μετασχηματισμό ορίων)

  Σύνταξη: map(value, fromLow, fromHigh, toLow, toHigh)

- **pow()** (επιστρέφει το αποτέλεσμα μίας δύναμης)

  Σύνταξη: x=pow(base,exponent)

- **sqrt()** (επιστρέφει την ρίζα ενός αριθμού)

  Σύνταξη: y=sqrt(x)

- **sin()** (υπολογίζει το ημίτονο ενός αριθμού)

  Σύνταξη: sin(rad)

- **cos()** (υπολογίζει το συνημίτονο ενός αριθμού)

  Σύνταξη: cos(rad)

- **tan()** (υπολογίζει την εφαπτομένη ενός αριθμού)

  Σύνταξη: tan(rad)

RANDOM NUMBERS

(Συναρτήσεις γεννήτριας ψευδοτυχαίων αριθμών)

- **random()** (δίδεται ένας νέος αριθμός από την γεννήτρια)

  Σύνταξη:random(value1);

  random(value1, value2);

- **randomSeed()** (θέτει τον σπόρο της γεννήτριας παραγωγής)

  Σύνταξη: randomSeed(value);

ARITHMETIC OPERATORS (Αριθμητικοί τελεστές)

- = (τελεστής εκχώρησης)

  Σύνταξη:var = value;

- + (τελεστής πρόσθεσης)

  Σύνταξη: //variablesorconstants

  sum=var1+var2; //sum

- -(τελεστής αφαίρεσης)

  Σύνταξη: sub=var1-var2; //difference

- *(τελεστής πολλαπλασιασμού)

  Σύνταξη: mul=var1*var2; //product

- / (τελεστής διαίρεσης)

  Σύνταξη: div=var1/var2 //quotient

- % (τελεστής υπόλοιπου ακεραίας διαίρεσης)

  Σύνταξη: res=dividend%divisor;

BOOLEAN OPERATORS (Λογικοί τελεστές)

- **&&** (λογική σύζευξη)

  Σύνταξη:expression1 && expression2

- || (λογική διάζευξη)

  Σύνταξη: expression1 || expression2

- **!** (λογική άρνηση)

  Σύνταξη: !expression

BITWISE OPERATORS(Δυαδικοί τελεστές)

- & (δυαδική σύζευξη)

  Σύνταξη: expression1 & expression2

- | (δυαδική διάζευξη)

  Σύνταξη: expression1 | expression2

- ^ (δυαδική αποκλειστική διάζευξη)

  Σύνταξη: expression1 ^ expression2

- ~ (δυαδική άρνηση)

  Σύνταξη: ~expression

- << (δυαδική αριστερή ολίσθηση)

  Σύνταξη: variable <<number_of_bits

- >> (δυαδική δεξιά ολίσθηση)

  Σύνταξη: variable >>number_of_bits

COMPOUNDOPERATORS (Τελεστές αύξησης και μείωσης)

- ++ (αύξηση κατά μία ακέραιη μονάδα)
- --(μείωση κατά μία ακέραιη μονάδα)

COMPOUND (Σύνθετοι τελεστές)

- +=, -=, *=, /=, %= (σύνθετοι αριθμητικοί τελεστές)
- &=, |=, ^=, ~=, <<=, >>= (σύνθετοι δυαδικοί τελεστές)

2.6 Υποστήριξη Βιβλιοθηκών (Libraries)

Η χρήση βιβλιοθηκών προσφέρουν περισσότερο λειτουργικότητα σε συνεργασία με το υλικό και τον χειρισμό των δεδομένων. Για να χρησιμοποιηθεί μια βιβλιοθήκη σε ένα sketch, μπορεί να επιλεγεί από το μενού Sketch → Import Library. Αυτό θα εισάγει μια ή περισσότερες βιβλιοθήκες #include δηλώσεις στην κορυφή του sketch. Επειδή οι βιβλιοθήκες φορτώνονται στην πλακέτα με το sketch, αυξάνουν το μέγεθος του χώρου που καταλαμβάνεται. Εάν ένα sketch δεν χρειάζεται πλέον μια βιβλιοθήκη, απλά μπορούμε να την διαγράψουμε από την κορυφή του κώδικα.

Για την εγκατάσταση των βιβλιοθηκών που δεν υπάρχουν ήδη στο λογισμικό, μπορεί να δημιουργηθεί ένας κατάλογος με την ονομασία libraries (βιβλιοθήκες), μέσα στον κατάλογο του sketchbook. Στην συνέχεια αποσυμπιέζουμε τη βιβλιοθήκη εκεί

Παρακάτω ακολουθούν μερικές από τις βιβλιοθήκες που υποστηρίζονται από το Arduino:

2.6.1 Standard Arduino Libraries

EEPROM: Για την ανάγνωση και εγγραφή σε ‘’μόνιμη’’αποθήκευση.

Ethernet: Για την σύνδεση με το διαδίκτυο χρησιμοποιώντας το Arduino Ethernet Shield.

Ethernet 2: Για την σύνδεση με το διαδίκτυο χρησιμοποιώντας το Arduino Ethernet Shield2 και toArduino Leonardo Eth.

Firmata:Για την επικοινωνία με τις εφαρμογές στον υπολογιστή χρησιμοποιώντας ένα τυπικό σειριακό πρωτόκολλο.

GSM: Για την σύνδεση σε δίκτυο GSM/GRPS με την ασπίδα GSM.

GSM2: Η βιβλιοθήκη GSM Shield 2 έχει νέες εντολές ήχου για τις φωνητικές κλήσεις και τη θέση εντολών.

Liquid Crystal: Για τον έλεγχο οθονών υγρών κρυστάλλων.

9 Axes Motion: Για τον έλεγχο της ασπίδας 9 AxesMotion.

SD: Για τον έλεγχο και την εγγραφή καρτών SDs.

Servo: Για τον έλεγχο σερβοκινητήρων.

SPI: Για την επικοινωνία με συσκευές που χρησιμοποιούν το πρόγραμμα Serial Peripheral Interface (SPI) Bus.

SoftwareSerial: Για την σειριακή επικοινωνία με οποιαδήποτε ψηφιακά Pins.

Stepper: Για τον έλεγχο κινητήρων Stepper.

TFT: Για την εισαγωγή κειμένου, εικόνων και σχημάτων στην οθόνη του Arduino TFT.

WiFi: Για την σύνδεση με το διαδίκτυο χρησιμοποιώντας την ασπίδα Arduino WiFi.

Wire: Με την διασύνδεση δύο αγωγών TWI/I2C με σκοπό την αποστολή και λήψη δεδομένων μέσω συσκευών και αισθητήρων.

Braccio: Για τον έλεγχο του Arduino Tinker Kit Braccio.

Lucky: Για τον έλεγο του Arduino Lucky Shield.

2.6.2 ArduinoLoRaShieldLibraries

LoRa Node: Για τον έλεγχο της ασπίδας του Arduino LoRa Node.

2.6.3 Arduino Primo Libraries

ArduinoLowPower: Για όσο το δυνατόν καλύτερη διαχείριση του Arduino Primo.

BLE: Για την διαχείριση Bluetooth στο Arduino Primo.

CLR: Για την διαχείριση IR στο Arduino Primo.

Comparator: Για την σύγκριση μιας αναλογικής εισόδου με ένα σήμα αναφοράς.

NFC: Για την διαχείριση της ετικέτας NFC στο Arduino Primo.

PPI: Προγραμματιζόμενη Περιφερειακή Διασύνδεση για Arduino Primo.

WiFi Link: Για την σύνδεση της πλακέτας με το WiFi.

2.6.4 Arduino Star OTTO Libraries

Wifi Link: Για την σύνδεση της πλακέτας με το WiFi.

Audio: Για την αναπαραγωγή και εγγραφή αρχείων ήχου από τις κάρτες μνήμης SD.

Arduino Graphics: Για την ανάπτυξη μιας βιβλιοθήκης γραφικών που θα χρησιμοποιηθεί στο Arduino Star OTTO.

2.6.5 Linino boards (Yun/ Yun Mini/ Industrial 101/ Tian) Libraries

Ciao: Μια εύκολη στην χρήση και ισχυρή τεχνολογία που απλοποιεί την αλληλεπίδραση μεταξύ του μικροελεκτη και του ‘’έξω κόσμου΄΄.

Bridge: Για την επικοινωνίασμεταξύ του επεξεργαστή Linux και των πινάκων Linino.

2.6.6 ATSAMD21G18 boards (M0/ M0 Pro/ Zero Pro/ Tian) Libraries

RTC: Για τον εσωτερικό ελεγχο του RTC στους πίνακες ATSAMD21G18.

Energy Saving: Αυτή η βιβλιοθήκη μπορεί να θέσει σε κατάσταση αναμονής τον μικροελεκτή και να τον αφυπνίσει μέσω εξωτερικού διακόπτη προειδοποίησης ,μόνο στους πίνακες ATSAMD21G18.

2.6.7 Due Only Libraries

Audio: Για την αναπαραγωγή αρχείων ήχου μέσω καρτών SD.

Scheduler : Για την διαχέιριση πολλαπών non-blocking εργασιών.

USB Host: Για την επικοινωνία περιφερειακά μέσω θυρών USB όπως ποντίκια, πληκτρολόγια.

2.6.8 Esplora Only Library

Esplora: Αυτή η βιβλιοθήκη σου δίνει την δυνατότητα να έχεις εύκολη πρόσβαση σε διάφορους αισθητήρες ενεργοποιητές οι οποίοι είναι τοποθετημένοι στον πίνακα Esplora Board .

2.6.9 Arduino Robot Library

Robot: Αυτή η βιβλιοθήκη επιτρέπει την εύκολη πρόσβαση στις λειτουργίες του Arduino Robot.

2.6.10 Arduino USB Libraries

Keyboard: Για την αποστολή των πληκτρολογήσεων στον συνδεδεμένο υπολογιστή.

Mouse: Για τον έλεγχο κίνησης του δρομέα σε ένα συνδεδεμένο υπολογιστή.

2.6.11 Contributed Libraries

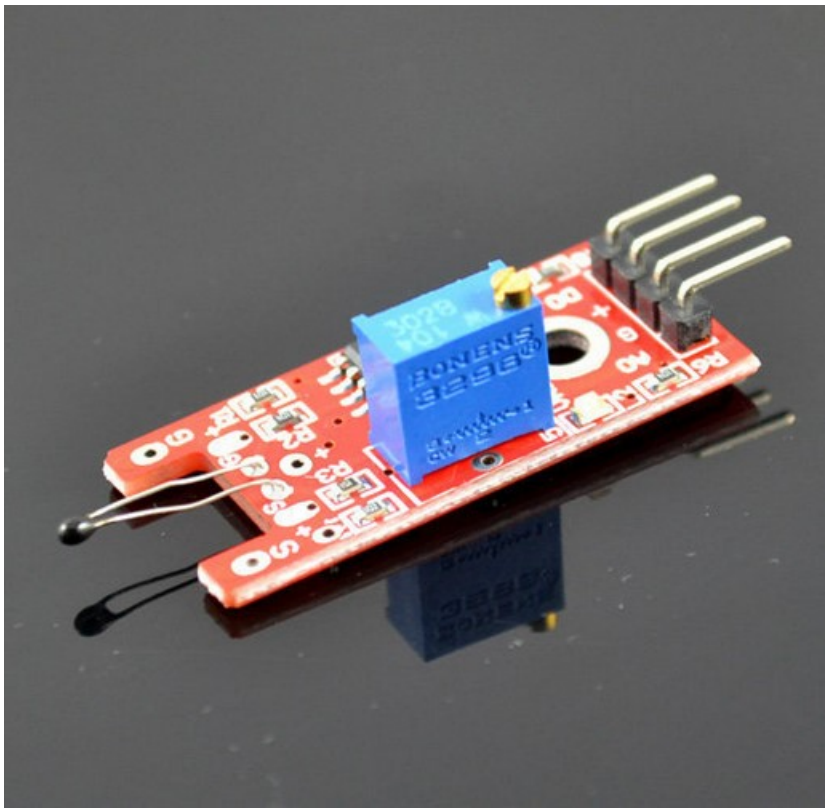
CapacitiveSensingLibrary :Για την μετατροπή δύο ή περισσότερων Pins του Arduino σε ένα χωρητικό αισθητήρα.

Τέλος μετά την εκτενή αναφορά στον προγραμματισμό ενός Arduino καθώς και στην περιγραφή των βασικών δομών και λειτουργιών αυτού, θα συνεχίσουμε την πτυχιακή με την περιγραφή μερικών αισθητήρων από το Starter SensorKit V1.0 for Arduino, αναλύοντας τον τρόπο λειτουργίας αυτών καθώς και τρόπο κατασκευής τους.

3 ΚΕΦΑΛΑΙΟ ΑΙΣΘΗΤΗΡΕΣ ΓΙΑ ARDUINO

3.1 Ψηφιακή μονάδα θερμοκρασίας

Η ψηφιακή μονάδα θερμοκρασίας συνδέεται στην θέση 13 LED της πλακέτας του Arduino δημιουργώντας έτσι ένα απλό κύκλωμα. Ο ψηφιακός αισθητήρας θερμοκρασίας συνδέεται ψηφιακά με τρείς διασυνδέσεις και λειτουργεί ως μεταβλητή αντίσταση (NTC/PTC). Πιο συγκεκριμένα δουλεύει ως εξής: καθώς αυξάνεται η θερμοκρασία, μειώνεται η τάση στη έξοδο του αισθητήρα. Αφού λάβει ένα εύρος τιμών της τάσης εξόδου, ρυθμίζεται κατάλληλα ώστε σύμφωνα με τις τιμές αυτές να μετατρέπει την τιμή της θερμοκρασία σε κατάλληλη τάση εξόδου. Όταν ο ψηφιακός αισθητήρας θερμοκρασίας ανιχνεύσει –αισθανθεί το κατάλληλο σήμα - κλειδί, τότε το Led ανάβει,σε διαφορετική περίπτωση παραμένει σβηστό.



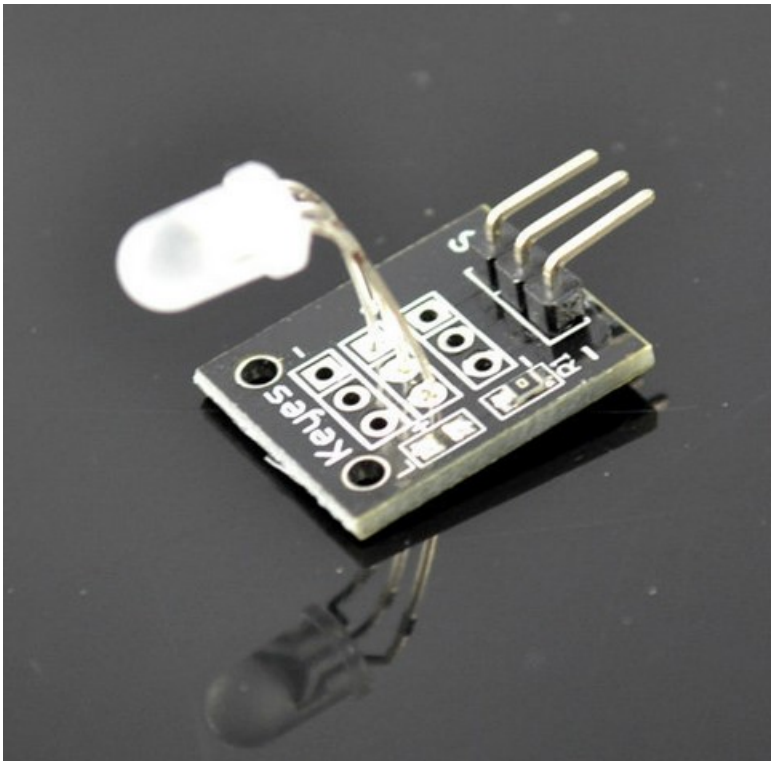
Εικόνα 3.1: Ψηφιακή μονάδα θερμοκρασίας

Ο κώδικας προγραμματισμού για τη ψηφιακή μονάδα επεξεργασίας είναι ο παρακάτω:–

```
intLed = 13 ;                          // Ορίστε την διεπαφήLed
intbuttonpin = 3;                      // Καθορίστε την διεπαφή του ψηφιακού αισθητήρα θερμοκρασίας
intval ;                               // Ορίστε αριθμητικές μεταβλητές val
voidsetup ()                           // Δήλωση λειτουργίας που ισχύει για όλο το πρόγραμμα
{
pinMode (Led, OUTPUT) ;                // ΟρίστετοLedωςδιεπαφήεξόδου
pinMode (buttonpin, INPUT) ;           // Καθορίστε την διεπαφή εξόδου του ψηφιακού αισθητήρα
}
voidloop ()                            // Δήλωση λειτουργίας που εκτελείται συνέχεια
{
val = digitalRead (buttonpin) ;        // Στηψηφιακήδιεπαφήθαδοθεί μια αρχική τιμή 3 για να διαβαστεί η τιμή val
if (val =HIGH)                         // Όταν ο ψηφιακός αισθητήρας θερμοκρασίας ανιχνεύσει ένα σήμα,η λυχνία Ledαναβοσβήνει.
{
digitalWrite (Led, HIGH);
}
else
{
digitalWrite (Led, LOW);
}
}
```

3.2 Κόκκινη και πράσινη μονάδα Led (κοινής καθόδου)

Τα προϊόντα αυτά χρησιμοποιούνται ευρέως σε ηλεκτρονικές συσκευές, PDA, MP3, στα ακουστικά, στις ψηφιακές φωτογραφικές μηχανές, στα DVD, στις επικοινωνίες, στους φορτιστές, στους υπολογιστές και στα κινητά τηλέφωνα καθώς και σε πολλά άλλα πεδία εφαρμογών. Αυτή η λυχνία Led περιέχει δύο διόδους εκπομπής φωτός. Αποτελείται από 3 ακροδέκτες από τους οποίους ο ένας είναι της γείωσης GND, ο ένας του πράσινου φωτός και ο άλλος του κόκκινου. Αν τροφοδοτήσουμε τη γείωση (GND) και έναν από τους ακροδέκτες μπορούμε να δούμε μόνο μια δέσμη φωτός (π.χ μόνο πράσινο ή μόνο κόκκινο χρώμα). Σε περίπτωση που τροφοδοτήσουμε τη γείωση και ταυτόχρονα και τους δύο ακροδέκτες αναμειγνύουμε το χρώμα και παίρνουμε μία κίτρινο/πορτοκαλί απόχρωση. Η συγκεκριμένη μονάδα λειτουργεί ανάλογα με την τάση που δίνεται.Η τάση του κόκκινου χρώματος κυμαίνεται από 1.4-1.8Vκαι του πράσινου χρώματος από 2.0-2.8V.



Εικόνα3. 2: Κόκκινη και πράσινη μονάδα Led (κοινής καθόδου)

-Διάμετρος	: 5mm	-Ρεύμα(mA) : 20mA
-Χρώμα Περιβλήματος	: Κανένα	-Τάση (V): Πράσινο :2.3-2 .6V
-Τύπος Πακέτου	: Διάχυση	-Τάση (V): Κόκκινο :1.9-2 .2V.
-Χρώμα	: Πράσινο +Κόκκινο	

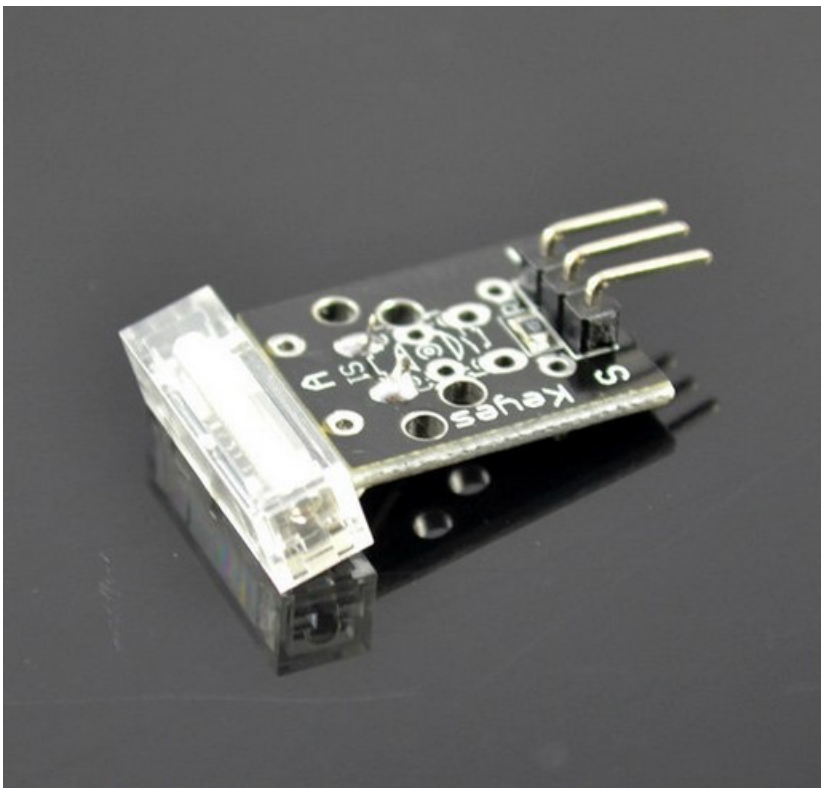
Ο κώδικας προγραμματισμού είναι ο παρακάτω:

```
intredpin = 11;          / / Επιλέξτε το pin για την κόκκινη λυχνία LED
intbluepin = 10;         / / Επιλέξτε το pin για την μπλε λυχνία LED
intval;
void setup () {
pinMode (redpin, OUTPUT);
pinMode (bluepin, OUTPUT);
Serial.begin (9600);
}
void loop ()
{
for (val = 255; val> 0; val -)
{
analogWrite (11, val);
analogWrite (10, 255-val);
SunFounder
delay (15);
}
for (val = 0; val<255; val + +)
{
analogWrite (11, val);
analogWrite (10, 255-val);
delay (15);
}Serial.println (val, DEC);
}
```

3.3 Αισθητήρας κρούσης

Ένας αισθητήρας κρούσεων, είναι ένας ηλεκτρονικός διακόπτης που ανιχνεύει ένα πλάτος κραδασμών και μεταφέρει τα σήματα που παράγονται στο κύκλωμα, ενεργοποιώντας το. Αποτελείται από ένα αγώγιμο ελατήριο κραδασμών, από έναν διακόπτη και από τον πείρο ενεργοποίησης.

Ο αισθητήρας κρούσεων λειτουργεί ως εξής: το αγώγιμο ελατήριο κραδασμών και ο πείρος ενεργοποίησης τοποθετούνται με ακρίβεια στον διακόπτη και στερεώνονται με κόλλα. Κανονικά, το ελατήριο και ο πείρος ενεργοποίησης δεν βρίσκονται σε επαφή. Μόλις όμως ο αισθητήρας ανιχνεύσει-δεχτεί την κρούση, το ελατήριο θα υποστεί μια δόνηση με αποτέλεσμα να έρθει σε επαφή με τον πείρο ενεργοποίησης, κλείνοντας έτσι κύκλωμα. Αυτό έχει σαν αποτέλεσμα να παράγεται ένα σήμα εξόδου ενεργοποιώντας μία εντολή όπως πχ να ανάψει μία λυχνία Led. Ο αισθητήρας κρούσης είναι πολύ ευαίσθητος, αφού μπορεί να αισθάνεται ακόμη και πολύ μικρής εντάσεως δόνηση.



Εικόνα 3.3: Αισθητήρας κρούσης

Ο κώδικας προγραμματισμού είναι ο παρακάτω:

```
intLed = 13 ;                          / / Καθορίστε την διεπαφήLed
intShock = 3                           / / Καθορίστε την διεπαφή του
                                       αισθητήρα κρούσης
intval ;                               / / Ορίστε αριθμητικές μεταβλητές val
SunFounder
voidsetup ()
{
pinMode (Led, OUTPUT) ;                / / ΟρίστετοLedως διεπαφή εξόδου
pinMode (Shock, INPUT) ;               / / Ορίστε την διεπαφή εξόδου του
                                       αισθητήρα κρούσης
}
voidloop ()
{
val = digitalRead (Shock) ;            / / Αναγνώριση της ψηφιακήςδιεπαφής
                                       αποδίδοντας την τιμή 3 στην θέσηval
if (val == HIGH)                       / / Όταν θα έχουμε κρούση τότε ο
                                       αισθητήρας θα ανιχνεύσει ένα σήμα και
                                       το Ledθα αναβοσβήσει
{
digitalWrite (Led, LOW);
}
else
{
digitalWrite (Led, HIGH);
}
}
```

3.4 Υπέρυθρος αισθητήρας αποφυγής εμποδίων

Ο υπέρυθρος αισθητήρας αποφυγής εμποδίων έχει σχεδιαστεί για την κατασκευή ενός τροχοφόρου ρομπότ που να μπορεί να αποφεύγει εμπόδια από μακριά. Ένας αισθητήρας αποφυγής εμποδίων χρησιμοποιεί την αρχή υπέρυθρης ανάκλασης για την ανίχνευση εμποδίων, έχοντας μεγάλη ακρίβεια. Όταν δεν υπάρχει κανένα αντικείμενο μπροστά, ο υπέρυθρος δέκτης δεν λαμβάνει σήματα. Ενώ όταν υπάρχει ένα αντικείμενο μπροστά, το υπέρυθρο φως θα μπλοκαριστεί από αυτό και θα αντικατοπτριστεί προς τον υπέρυθρο δέκτη ο οποίος θα λάβει σήμα, αναγνωρίζοντας έτσι ότι υπάρχει αντικείμενο κοντά του. Ένας αισθητήρας αποτελείται κυρίως από έναν υπέρυθρο πομπό, έναν υπέρυθρο δέκτη και ένα ποτενσιόμετρο. Σύμφωνα με τον αντανακλαστικό χαρακτήρα ενός αντικειμένου, αν δεν υπάρχει κανένα εμπόδιο, η υπέρυθρη ακτίνα που εκπέμπεται από τον πομπό θα αποδυναμωθεί από την απόσταση διάδοσης και τελικά θα εξαφανιστεί. Αντίθετα εάν υπάρχει εμπόδιο, όταν η υπέρυθρη ακτινοβολία συναντήσει το εμπόδιο, θα αντανακλαστεί πίσω στον υπέρυθρο δέκτη. Στη συνέχεια, ο υπέρυθρος δέκτης ανιχνεύει αυτό το σήμα και επιβεβαιώνει ότι υπάρχει εμπόδιο μπροστά του. Η απόσταση αποφυγής εμποδίων του αισθητήρα υπερύθρων είναι ρυθμιζόμενη η οποία ρυθμίζεται περιστρέφοντας το ποτενσιόμετρο.

Εικόνα 3.4: Υπέρυθρος αισθητήρας αποφυγής εμποδίων

Χαρακτηριστικά:

- ΣήμαΕξόδου	:ΕπίπεδοTTL(στο χαμηλό επίπεδο υπάρχει εμπόδιο, δεν υπάρχει μεγάλο εμπόδιο)
-Τάση λειτουργίας	: DC 3.3V-5V
-Θερμοκρασία λειτουργίας	: -10 ℃ - +50 ℃
-Απόσταση ανίχνευσης	:2-40cm
-ΙΟδιεπαφή	: 4-διεπαφές καλωδίων (- / + / S / EN)
-Βάρος	: 9g
- Μέγεθος	: 28mm × 23mm
- Αποτελεσματική γωνία	: 35 °
- Ρεύμα εργασίας	≥ 20mA

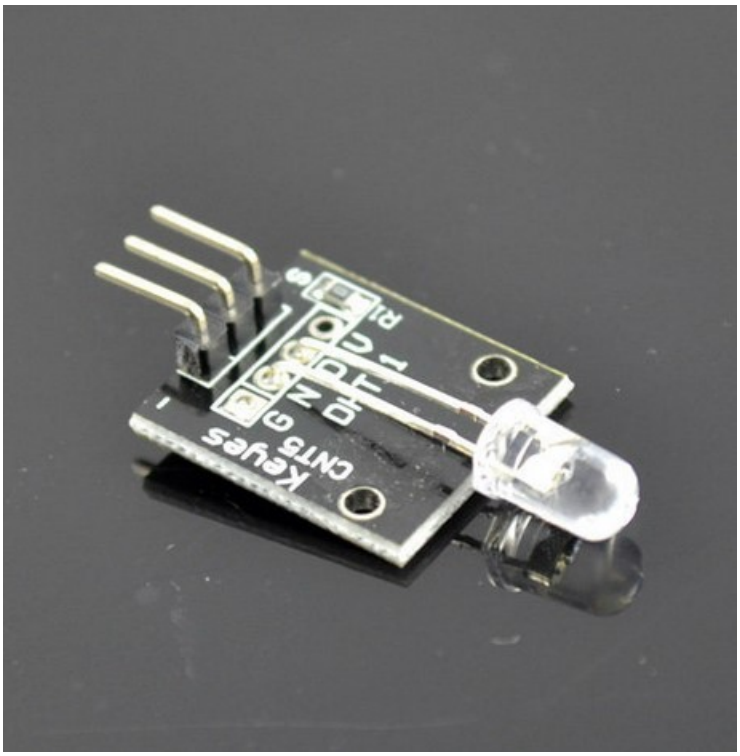
Ο κώδικας προγραμματισμού είναι ο παρακάτω:

```
intLed = 13 ;                    / / Καθορίστε την διεπαφήLed
intbuttonpin = 3;                / / Καθορίστε την διεπαφή του αισθητήρα αποφυγής
                                 εμποδίων
intval   ;                       / / Ορίστε τις αριθμητικές μεταβλητές val
voidsetup ()
{
pinMode (Led, OUTPUT) ;          / / Καθορίστε την επαφή εξόδου του Led
pinMode (buttonpin, INPUT) ;/ / Καθορίστε τηνεπαφή εξόδου του αισθητήρα
                                 αποφυγήςεμποδίων
}
voidloop ()
{
val = digitalRead (buttonpin) ;/ / Στην ψηφιακή διεπαφή θα δοθεί μια τιμή 3 για να
                                 διαβαστεί η τιμή val
if (val == HIGH)                 // Όταν ο αισθητήρας αποφυγής εμποδίων εντοπίσει ένα
                                 σήμα,η λυχνία Led θα αναβοσβήσει
{
digitalWrite (Led, HIGH);
}
else
{
digitalWrite (Led, LOW);
}
```

3.5 Μονάδα Led χρωμάτων που αναβοσβήνει αυτόματα

ΗμονάδαLed 7 χρωμάτων αναβοσβήνει αυτόματα χρησιμοποιώντας μια δίοδο εκπομπής φωτός υψηλής φωτεινότητας με διάμετρο 5mm και έχει τα ακόλουθα χαρακτηριστικά:

Τύπος Προϊόντος	: LED
Σχήμα	: ΣτρογγυλόLED 5mmDIPtype
Χρώμα	: ροζ, κίτρινο, πράσινο (υψηλής φωτεινότητας)
Τύπος φακού	: λευκή ομίχλη
Συγκεκριμένη τάση λειτουργίας	:3.0-4 .5 V



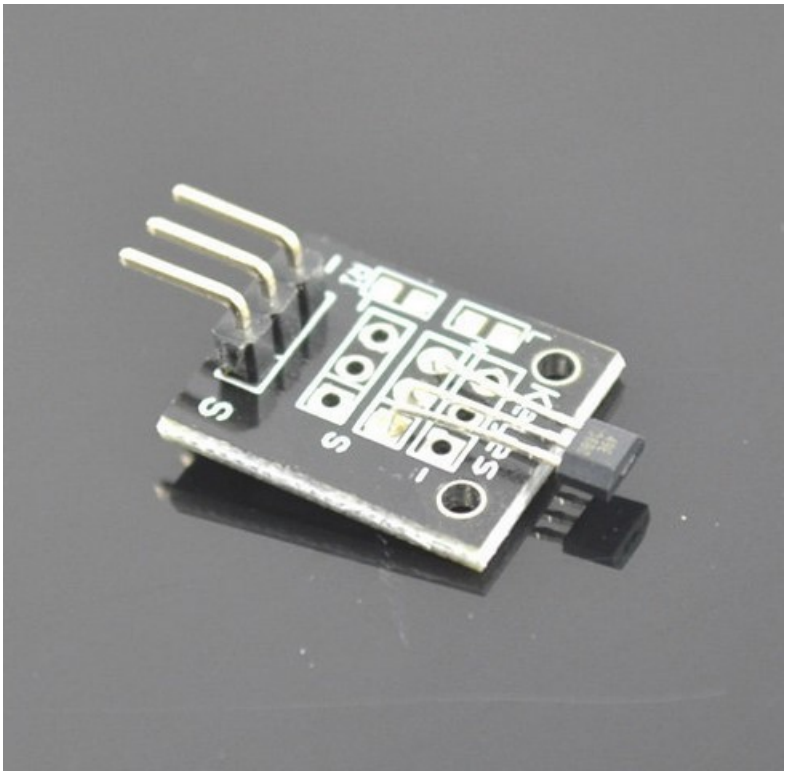
Εικόνα 3.5: Μονάδα Led χρωμάτων που αναβοσβήνει αυτόματα

Ο κώδικας προγραμματισμού είναι ο παρακάτω:

```
/ *

Blink

Ενεργοποιεί μια ενδεικτική λυχνία για δυο δευτερόλεπτα,στη συνέχεια απενεργοποιείται
για δυο δευτερόλεπτα, επανειλημμένα.

Αυτός ο κωδικας είναι .

* /

voidsetup () {

SunFounder                                  / / Αρχικοποιήστε τον ψηφιακό ακροδέκτη
                                            ως έξοδο

                                  / / Η επαφή-ακίδα 13 εχει ένα Led συνδεδεμένο
                          στις περισσότερες πλακέτες Arduino

pinMode (13, OUTPUT);

}

void loop () {

digitalWrite (13, HIGH);                    / / ανάβει η λυχνίαLed

delay (2000);                               / / περιμένουμε για ένα δευτερόλεπτο

digitalWrite (13, LOW);                     / / σβήνει η λυχνίαLed

delay (2000);                               / / περιμένουμε για ένα δευτερόλεπτο

}
```

3.6 Αναλογικός αισθητήρας μαγνητικού πεδίου

Ο παραπάνω αισθητήρας αποτελεί ένα μαγνητικό διακόπτη. Αν δεν υπάρχει ανιχνεύσιμο μαγνητικό πεδίο, η γραμμή σήματος του αισθητήρα βρίσκεται στην θέση HIGH (στα 3,5 V).Όταν παρουσιαστεί ένα μαγνητικό πεδίο κοντά στον αισθητήρα ,η γραμμή σήματος οδηγείται στη θέση LOW και ταυτόχρονα ανάβει η ενδεικτική λυχνία ledτου αισθητήρα. Η πολικότητα του μαγνητικού πεδίου είναι αυτή που επηρεάζει την δράση μεταγωγής {Μεταγωγή: είναι η διαδικασία μετάδοσης της πληροφορίας μέσω διαδοχικών ενδιάμεσων κόμβων προκειμένου να επικοινωνήσουν δύο τερματικοί κόμβοι δηλαδή πομπού και δέκτη και χρησιμοποιείται σε δίκτυα επικοινωνίας}. Η εμπρόσθια πλευρά του αισθητήρα χρησιμοποιεί την αντίστροφη πολικότητα καθώς το πίσω μέρος του αισθητήρα ενεργοποιείται. Στο παρακάτω κώδικα προγραμματισμού το LED στο ARDUINO (pin 13) θα ανάψει όταν υπάρχει ένα μαγνητικό πεδίο. Στην κατάσταση ηρεμίας η κατανάλωση ισχύος είναι 3 mA ενώ όταν το Led ενεργοποιείται η κατανάλωση φτάνει στα 8mA .



Εικόνα 3.6: Αναλογικός αισθητήρας μαγνητικού πεδίου

Γνωρίζουμε ότι η ηλεκτρική ενέργεια που μεταφέρεται μέσω ενός αγωγού παράγει ένα μαγνητικό πεδίο, το οποίο ποικίλλει ανάλογα με την ένταση του ρεύματος. Ο συγκεκριμένος αισθητήρας μπορεί να χρησιμοποιηθεί για την μέτρηση

του ρεύματος αυτού χωρίς να διακόψει το κύκλωμα και αυτό γίνεται μέσω του αναλογικού σήματος που μεταδίδει μετατρέποντας την μαγνητική ενέργεια του ηλεκτρικού ρεύματος σε τάση εξόδου. Συνήθως αυτός ο αισθητήρας είναι ενσωματωμένος με ένα πυρήνα τυλίγματος ή αλλιώς ένας μόνιμος μαγνήτης περιβάλλει τον προς μέτρηση αγωγό.

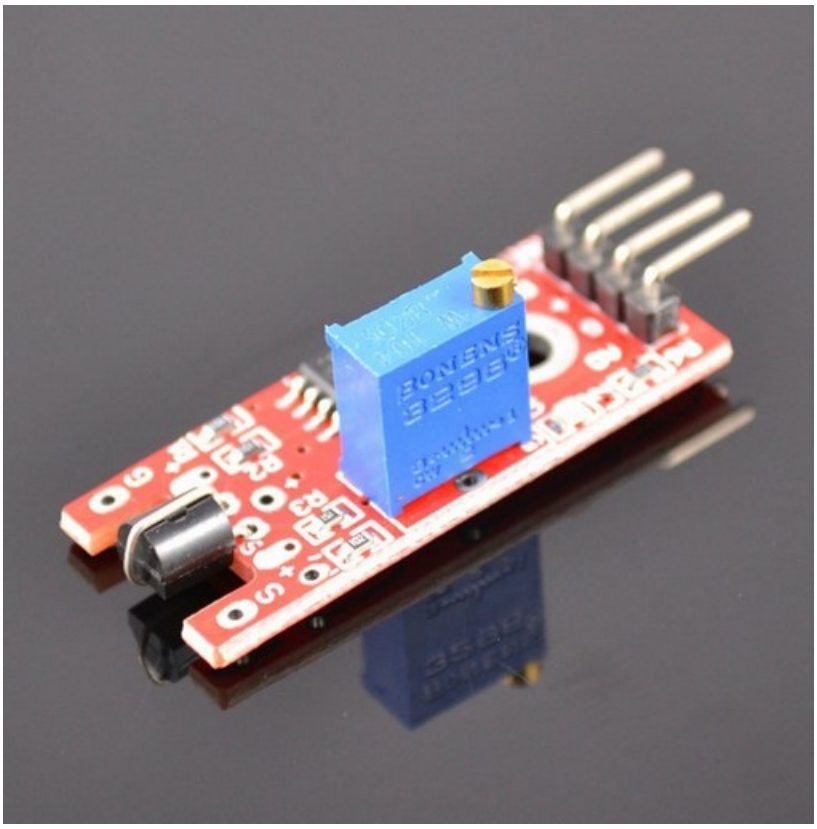
Ο κώδικας προγραμματισμού είναι ο παρακάτω:

```
IntsensorPin = A5;      // Επιλέξτε τον ακροδέκτη εισόδου

intledPin = 13;          // Επιλέξτε τον ακροδέκτη για την λυχνία Led

intsensorValue = 0;     // Αποθηκεύστε την τιμή της μεταβλητής που προέρχεται από τον αισθητήρα

void setup () {

SunFounder

pinMode (ledPin, OUTPUT);

Serial.begin (9600);

}

void loop () {

sensorValue = analogRead (sensorPin);

digitalWrite (ledPin, HIGH);

delay (sensorValue);

digitalWrite (ledPin, LOW);

delay (sensorValue);

Serial.println (sensorValue, DEC);

}
```

3.7 Μεταλλικός αισθητήρας επαφής

Ο μεταλλικός αισθητήρας αφής είναι ένας τύπος διακόπτη, ο οποίος λειτουργεί μόνο όταν έρθει σε επαφή η καμπυλωτή ακίδα του τρανζίστορ με ένα φορτισμένο σώμα. Το τρανζίστορ υψηλής συχνότητας μεταφέρει ηλεκτρισμό όταν λαμβάνει ηλεκτρομαγνητικά σήματα. Πιο συγκεκριμένα λειτουργεί ως εξής:

Όταν αγγίξουμε το ηλεκτρόδιο βάσης του τρανζίστορ με τα δάκτυλα μας ή με οποιοδήποτε άλλο μέρος του σώματος (όπως είναι γνωστό το ανθρώπινο σώμα είναι ένα είδος αγωγού) μία κεραία μπορεί να λάβει τα ηλεκτρομαγνητικά σήματα μέσω του αέρα. Αυτά τα σήματα των ηλεκτρομαγνητικών κυμάτων που συλλέγονται στο τρανζίστορ από το ανθρώπινο σώμα, ενισχύονται μέσω αυτού και στην συνέχεια επεξεργάζονται από την συγκριτή της μονάδας, για να παραχθούν εν τέλει σταθερά σήματα εξόδου. Με το Led να είναι συνδεδεμένο στον ακροδέκτη 13 του Arduino και ταυτόχρονα ο ακροδέκτης D0 του αισθητήρα στο D7 της πλακέτας, έχει ως αποτέλεσμα η λυχνία Led να ανάψει όταν ο μεταλλικός αισθητήρας επαφής εντοπίσει σήματα αφής, διαφορετικά θα είναι εκτός λειτουργίας



Εικόνα 7 : Μεταλλικός αισθητήρας επαφής

Ο κώδικας προγραμματισμού είναι ο παρακάτω:

```
intLed = 13 ;                          / / Καθορίστε την επαφή Led
intbuttonpin = 3;                       / / Καθορίστε τη διεπαφή του αισθητήρα
intval ;                               / / Ορίστε αριθμητικές μεταβλητές val
voidsetup ()
{
pinMode (Led, OUTPUT) ;                / / ΟρίστετοLedωςεπαφήεξόδου
pinMode (buttonpin, INPUT) ;           / / Ορίστετηνδιεπαφήεξόδου του μεταλλικού
                                       αισθητήραεπαφής
SunFounder
}
voidloop ()
{
val = digitalRead (buttonpin) ;        / / Στηψηφιακήδιεπαφήδόθηκετιμή 3
                                       γιαναδιαβαστείτιμήval
if (val == HIGH)                       / /
                                       Όταναμεταλλικόςαισθητήραςαφήςεντοπίσειένασ
                                       ήμα ,λυχνίαLedαναβοσβήνει.
{
digitalWrite (Led, HIGH);
}
else
{
digitalWrite (Led, LOW);
}
}
```

3.8 Μονάδα ανίχνευσης ήχου υψηλής ευαισθησίας

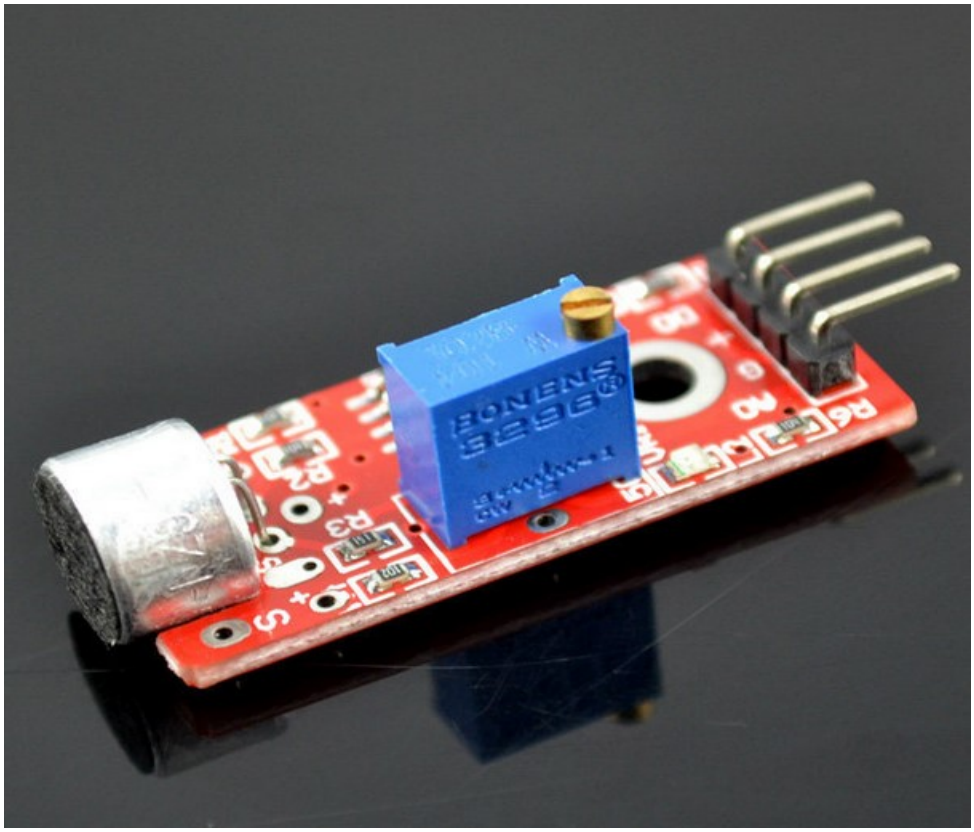
Ο παραπάνω αισθητήρας μπορεί να ανιχνεύσει τον ήχο και έχει δύο εξόδους:

Η 1η εκπέμποντας από το μικρόφωνο του Sun Founder ένα αναλογικό σήμα σε πραγματικό χρόνο

Η 2η όσο και ένα ψηφιακό σήμα όταν η ένταση του ήχου φτάσει ένα συγκεκριμένο όριο παράγοντας αναλόγως σήμα υψηλής ή χαμηλής τάσης εξόδου (Το Οριο-αντιπροσωπεύει την ευαισθησία του αισθητήρα η οποία μπορεί να ρυθμιστεί μέσω ποτενσιόμετρου.

Για παράδειγμα η ψηφιακή έξοδος μπορει να ενεργοποιηθεί μέσω ενός χτυπήματος των δυο χεριών μας (παλαμάκια) με σκοπο να ανάψουμε ή να σβήσουμε τα φώτα του σπιτού.

Αυτός ο αισθητήρας είναι πολύ χρήσιμος σε συνδιασμό με μια μονάδα ρελέ,αφου μπορούν να συνδιαστούν σε ηλεκτρονικά κυκλώματα κάνοντας την ζωή μας πιο εύκολη.



Εικόνα 3.8 : Μονάδα ανίχνευσης ήχου υψηλής ευαισθησίας

Τα χαρακτηριστικά της μονάδος :

-Υπάρχει μια οπή στερέωσης διαμέτρου 3 mm

-Χρησιμοποιεί εναλλασσόμενο ρεύμα τάσεως 5 Vμε αναλογική έξοδο

-Υπάρχουν flip εξόδου με συγκεκριμένα επίπεδα ορίων

-Χρησιμοποιεί υψηλής ευαισθησίας μικρόφωνο και ενδεικτικών λυχνιών

Και τέλος η τάση εξόδουτου σήματος από τον συγκριτή είναι μικρής εντάσεως.

Ο κώδικας προγραμματισμού είναι ο παρακάτω:

```
1 Ψηφιακή έξοδος:
intLed = 13 ;                          / Καθορίστε την διεπαφήLed
intbuttonpin = 3                       / Καθορίστε την διεπαφή του αισθητήρα D0
; Intval ;                             / / Ορίστε αριθμητικές μεταβλητές val
voidsetup ()
{
pinMode (Led, OUTPUT) ;                / / Ορίστε το Ledως διεπαφή εξόδου
pinMode (buttonpin, INPUT) ;           / / Η διεπαφή εξόδου D0 είναι ο καθορισμένος
αισθητηρας
}
voidloop ()
{
val = digitalRead (buttonpin) ;        / / Στη ψηφιακή διεπαφή θαδ οθεί μια τιμή 3
                                       ώστε να διαβαστεί η τιμή val
if (val == HIGH)                       / / ; Όταν η μονάδα ανίχνευσης ήχου  ανιχνεύσει
                                       ένα σήμα, η λυχνία Led αναβοσβήνει
{
digitalWrite (Led, HIGH)
;}
else
{
```

```
digitalWrite (Led, LOW)
;}
}
2 Αναλογική έξοδος:
intsensorPin = A5;                    // Επιλέξτε τον ακροδέκτη εισόδου για το ποτενσιόμετρο
intledPin = 13;                       // Επιλέξτε τον ακροδέκτη για την λυχνία Led
intsensorValue = 0;                   // Αποθηκεύστε τις μεταβλητές τιμές που
                                      προέρχονται από τον αισθητήρα
voidsetup () {
pinMode (ledPin, OUTPUT);
Serial.begin (9600);
}
void loop () {
sensorValue = analogRead (sensorPin);
digitalWrite (ledPin, HIGH);
SunFounder
delay (sensorValue);
digitalWrite (ledPin, LOW);
delay (sensorValue);
Serial.println (sensorValue, DEC);
}
```

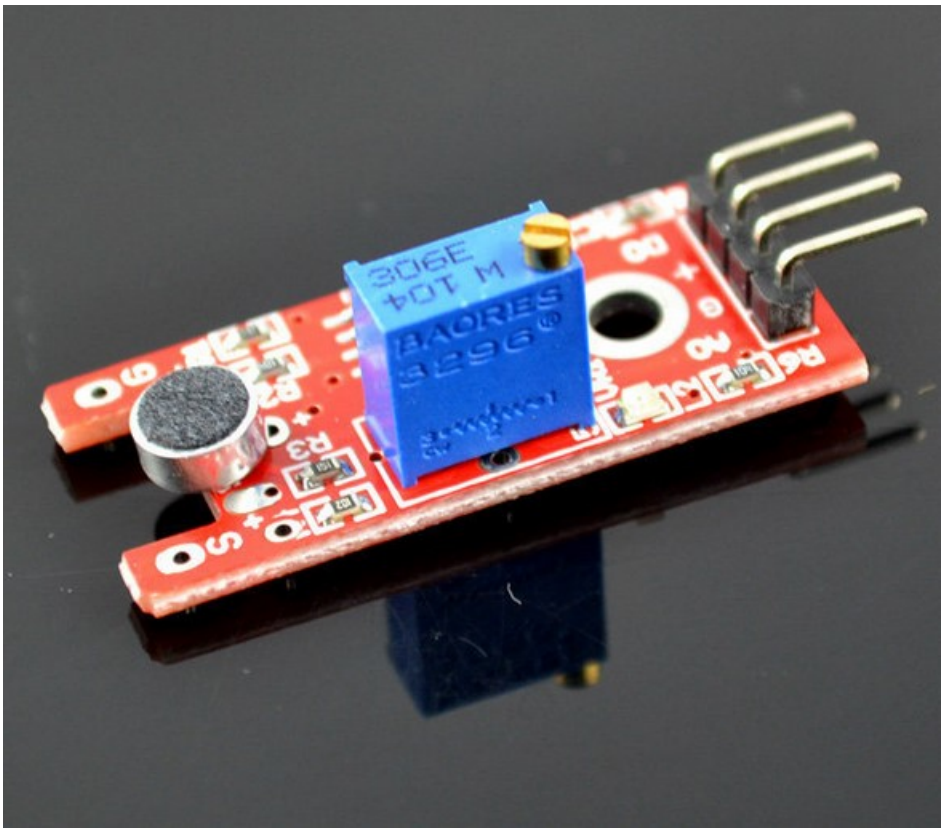
3.9 Μονάδα μικροφώνου για ανίχνευση ήχου

Γενικά υπάρχουν δύο είδη αισθητήρων ανίχνευσης ήχου αυτού που περιγράψαμε παραπάνω ο οποίος είναι ο αισθητήρας ανίχνευσης ήχου υψηλής ευαισθησίας και αυτός που θα περιγράφουμε τώρα που είναι ο αισθητήρας μικροφώνου. Η μόνη διαφορά αυτών των δύο αισθητήρων αφορά την ευαισθησία, όπου ο αισθητήρας μικροφώνου είναι χαμηλότερης ευαισθησίας.

Ο παραπάνω αισθητήρας μπορεί να ανιχνεύσει τον ήχο και να έχει δύο εξόδους:

1. Εκπέμποντας απο το μικρόφωνο του Sun Founder ένα αναλογικό σήμα σε πραγματικό χρόνο

2. Όσο και ένα ψηφιακό σήμα, όταν η ένταση του ήχου φτάσει ένα συγκεκριμένο όριο παράγοντας αναλόγως σήμα υψηλής ή χαμηλής τάσης εξόδου (Το Οριο- αντιπροσωπεύει την ευαισθησία του αισθητήρα η οποία μπορεί να ρυθμιστεί μέσω ποτενσιόμετρου).



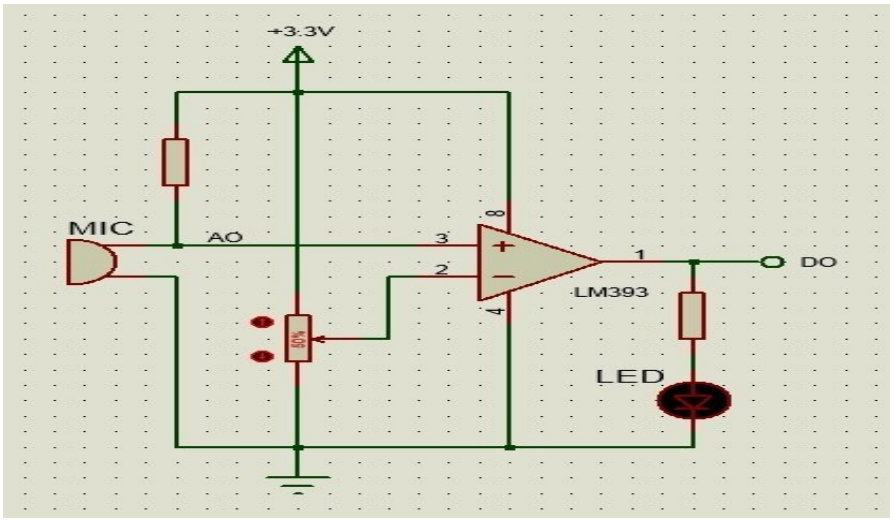
Εικόνα 3.9 : Μονάδα μικροφώνου για ανίχνευση ήχου

Μονάδα μικροφώνου για ανίσχευση ήχου

Η αρχή λειτουργίας του αισθητήρα είναι:

Το μικρόφωνο μετατρέπει το ηχητικό σε ηλεκτρικό σήμα (σε αναλογική ποσότητα),στην συνέχεια την αναλογική ποσότητα σε ψηφιακή τιμή με την βοήθεια του ADCκαι να το μεταφέρει σε MCUμε σκοπό την επεξεργασία του. Για να γίνει πιο κατανοητή ο τρόπος λειτουργίας χρησιμοποιώ το παρακάτω παραδειγμα

Στο παρακάτω σχήμα απεικονίζεται ο συγκριτής τάσης LM393. Όταν η τάση του ακροδέκτη εντός φάσης (ακροδέκτης 3) είναι υψηλότερη από αυτή του ακροδέκτη αναστροφής (ακροδέκτη 2),ο ακροδέκτης εξόδου 1 θα αποδώσει υψηλή τάση εξόδου, διαφορετικά εξέρχεται χαμηλή τάση εξόδου. Αρχικά όμως πρέπει να ρυθμίσουμε το ποτενσιόμετρο ώστε η τάση για τον ακροδέκτη 2 του LM393 να είναι μικρότερη από 5V.Όταν δεν υπάρχει φωνητική είσοδος, η αντίσταση του μικροφώνου είναι μεγάλη. Η τάση σε αυτή την φάση στον ακροδέκτη 3 του LM393 είναι κοντά στην τάση τροφοδοσίας (5V) με αποτέλεσμα ο ακροδέκτης 1 να αποδώσει υψηλή τάση εξόδου και η ενδεικτική λυχνία να είναι αναμμένη. Όταν τώρα υπάρχει φωνητική είσοδος, η αντίσταση του μικροφώνου μειώνεται, η τάση εξόδου του ακροδέκτη 1 γίνεται χαμηλής τιμής και η λυχνία Led σβήνει .Τέλος συνδέουμε τον ακροδέκτη 1 στην πλακέτα του Arduino στην θέση IO για να ανιχνεύσουμε τους ήχους αφού πρώτα το προγραμματίσουμε μέσω του παρακάτω κώδικα



Ο κώδικας προγραμματισμού είναι ο παρακάτω:

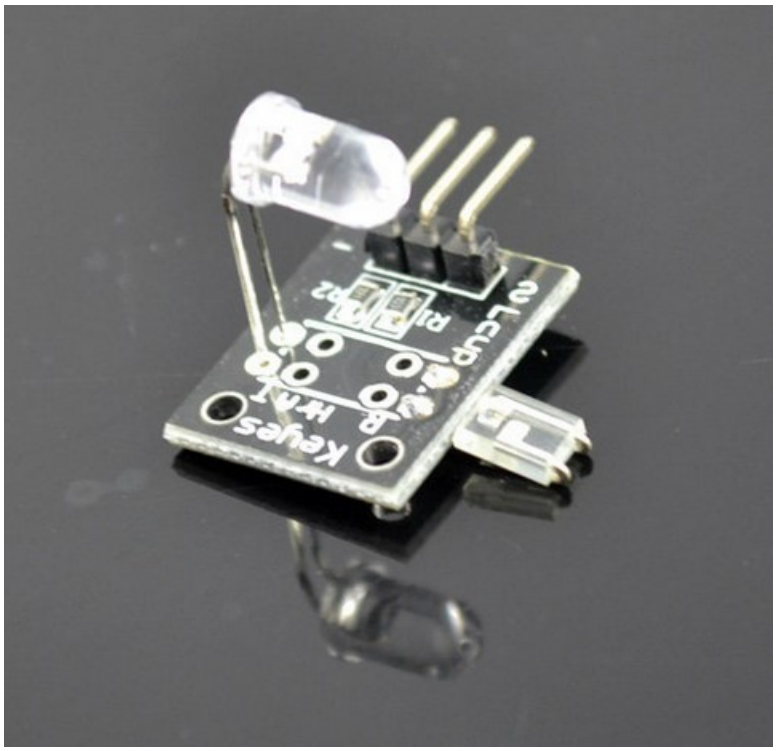
```
1 Ψηφιακή έξοδος
intLed = 13 ;                    / / Καθορίστε την διεπαφήLed
intbuttonpin = 3                 / / Καθορίστε την διεπαφή του αισθητήρα DO
intval ;                         / / Ορίστε αριθμητικές μεταβλητές val
voidsetup ()
SunFounder
{
pinMode (Led, OUTPUT)            ;/ / Ορίστετο  Led  ως διεπαφή εξόδου
pinMode (buttonpin, INPUT) ;     // η διεπαφή εξόδου DO είναι καθορισμένος
        αισθητήρας}
voidloop ()
{
val = digitalRead (buttonpin) ;  / / Στην ψηφιακή διεπαφή θα δοθεί μια τιμή 3
        ώστε να διαβαστεί η τιμή val
if (val == HIGH)                 / / ; Όταν η μονάδα ανίχνευσης ήχου  ανιχνεύσει
                                 ένα σήμα,η λυχνία Led αναβοσβήνει {
digitalWrite (Led, HIGH)
}
else
{
digitalWrite (Led, LOW)
}
}
2)Αναλογική έξοδος:
intsensorPin = A5;          / / Επιλέξτε τον ακροδέκτη εισόδου για το ποτενσιόμετρο
intledPin = 13;             / / Επιλέξτε τον ακροδέκτη για την λυχνία Led
```

```
  intsensorValue = 0;              / / Αποθηκεύστε τις μεταβλητές τιμές που προέρχονται
                                     από τον αισθητήρα

void setup () {

pinMode (ledPin, OUTPUT);

Serial.begin (9600);

}

void loop () {

sensorValue = analogRead (sensorPin);

digitalWrite (ledPin, HIGH);

delay (sensorValue);

digitalWrite (ledPin, LOW);

delay (sensorValue);

Serial.println (sensorValue, DEC);

}
```

3.10 Μονάδα δακτυλικής μέτρησης καρδιακών παλμών

Η μονάδα δακτυλικής μέτρησης καρδιακών παλμών χρησιμοποιεί ένα υπέρυθρο φωτεινό IR Led και ένα φωτοτρανζίστορ για να ανιχνεύσει τον παλμό του δακτύλου. Ένα κόκκινο Led αναβοσβήνει σε κάθε παλμό.
Πιο αναλυτικά η παρακολούθηση των παλμών λειτουργεί ως εξής:

Το Led αποτελεί την ‘’φωτεινή πλευρά του δακτύλου‘’ και το φωτοτρανζίστορ, το οποίο χρησιμοποιείται για να λάβει του παλμούς της αρτηριακής πίεσης που εκπέμπονται από το δάκτυλο. Όταν η αντίσταση του φωτοτρανζίστορ αλλάξει ελαφρώς , η φωτεινή ένδειξη Ledθα μεταβληθεί.



Εικόνα 3.10 :Μονάδα δακτυλικής μέτρησης καρδιακών παλμών

Ο κώδικας προγραμματισμού είναι ο παρακάτω:

```
ARDUINO test code:
/ / Pulse Monitor Test Script
intledPin = 13;
intsensorPin = 0;
double alpha = 0.75;
```

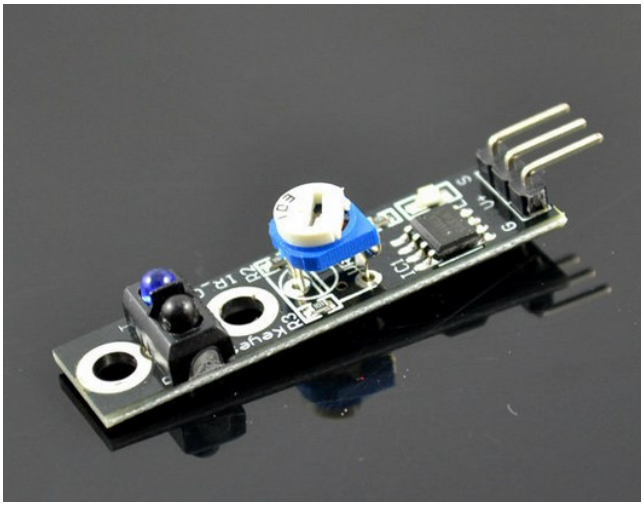
```
int period = 20;
double change = 0.0;
void setup ()
{
pinMode (ledPin, OUTPUT);
Serial.begin (115200);
}
SunFounder
void loop ()
{
static double oldValue = 0;
static double oldChange = 0;
intrawValue = analogRead (sensorPin);
double value = alpha * oldValue + (1 - alpha) * rawValue;
Serial.print (rawValue);
Serial.print (",");
Serial.println (value);
oldValue = value;
delay (period);
}
```

3.11 Αισθητήρας παρακολούθησης

Ένας αισθητήρας παρακολούθησης έχει την ίδια αρχή με έναν αισθητήρα αποφυγής εμποδίων, αλλά έχει εκπομπή μικρής ισχύος. Πιο αναλυτικά η αρχή λειτουργίας του παραπάνω αισθητήρα είναι:

Όταν ο υπέρυθρος πομπός εκπέμψει ακτίνες σε ένα κομμάτι χαρτί ,διότι το χαρτί έχει λευκή επιφάνεια, αυτές θα λάμψουν πάνω σε αυτό, θα αντανακλαστούν και θα ληφθούν από τον δέκτη, κάνοντας τον ακροδέκτη Sνα εξάγει χαμηλή τάση εξόδου. Αντίθετα εάν οι ακτίνες συναντήσουν μαύρες γραμμές, θα απορροφηθούν από αυτές και έτσι ο δέκτης δεν λάβει τίποτα πίσω, κάνοντας τον ακροδέκτη S να εξάγει υψηλή τάση εξόδου.

Η λειτουργία του θα γίνει κατανοητή μέσω της περιγραφής ενός πειράματος. Σε αυτό το πείραμα θα χρησιμοποιήσουμε έναν αισθητήρα αποφυγής εμποδίων και ένα Ledσυνδεδεμένο στον ακροδέκτη 13 της πλακέτας του Arduino Sun Founder ώστε να δημιουργήσουμε ένα απλό κύκλωμα με σκοπό να φτιάξουμε ένα φως παρακολούθησης. Δεδομένου ότι έχει συνδεθεί μία ενδεικτική λυχνία στον ακροδέκτη 13,συνδέουμε την ακίδα στην θέση D8 της πλακέτας. Έτσι όταν ο αισθητήρας παρακολούθησης ανιχνεύσει σήματα αντανάκλασης (λευκό) η λυχνία Ledθα ανάψει, διαφορετικά όταν ανιχνεύει μαύρη γραμμή η λυχνία Led θα απενεργοποιηθεί.



Εικόνα 3.11 :Αισθητήρας παρακολούθησης

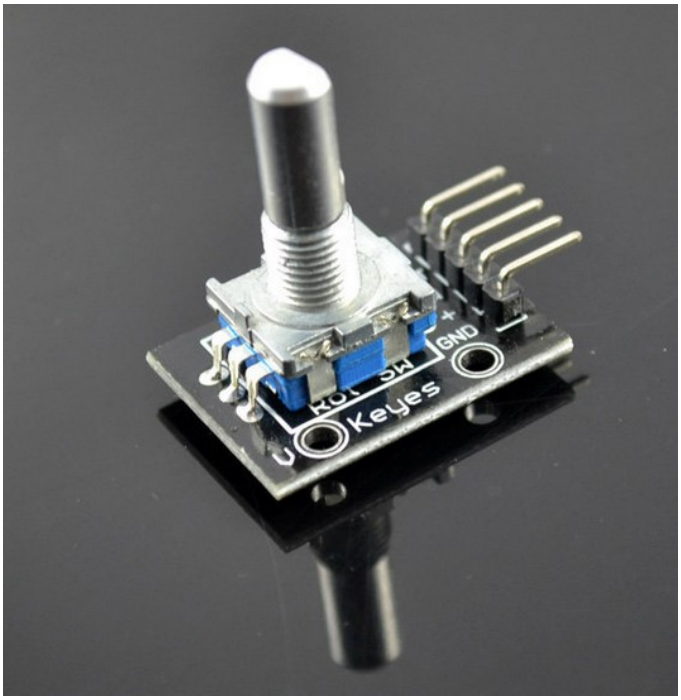
Ο κώδικας προγραμματισμού είναι ο παρακάτω:

```
Routines source code:

intLed = 13 ;                    / / Όρισε την επαφή Led

intbuttonpin = 3;                / / Όρισε την επαφή του αισθητήρα παρακολούθησης

intval ;                         / / Όρισε τις αριθμητικές μεταβλητές val

void setup ()

{

pinMode (Led, OUTPUT) ;    / / Καθόρισε τοLedως επαφή εξόδου

pinMode (buttonpin, INPUT) ;/ / Καθόρισε τον αισθητήρα παρακολούθησης ως
επαφή εξόδο

}

void loop ()

{

val = digitalRead (buttonpin) ;/ / digital interface will be assigned a value of 3 to read
val

SunFounder

if (val == HIGH)                 / / Όταν ο αισθητήρας παρακολούθησης ανιχνεύσει ένα
                                                                  σήμα, το Ledθα ανάψει

{

digitalWrite (Led, HIGH);

}

else

{

digitalWrite (Led, LOW);

}

}
```

3.12 Περιστροφικός κωδικοποιητής

Ένας περιστροφικός κωδικοποιητής είναι μια ηλεκτρομηχανική συσκευή που μετατρέπει την κίνηση ενός άξονα ή την γωνιακή του θέση,σε ένα αναλογικό ή ψηφιακό κώδικα. Οι περιστροφικοί κωδικοποιητές τοποθετούνται συνήθως στην πλευρά που είναι κάθετη προς τον άξονα.Λειτουργούν ως αισθητήρες για την ανίχνευση της γωνίας, της ταχύτητας, του μήκους, της θέσης και της επιτάχυνσης στο πεδίο αυτοματισμού.

Υπάρχουν κυρίως δύο τύποι περιστροφικού κωδικοποιητή: οι απόλυτοι και οι αυξητικοί (ή αλλιώς σχετικοί). Η έξοδος των απόλυτων κωδικοποιητών δείχνει την τρέχουσα θέση του άξονα, καθιστώντας τους ως μεταγωγείς γωνίας. Η έξοδος των αυξητικών κωδικοποιητών παρέχει πληροφορίες σχετικά με την κίνηση του άξονα, η οποία τυπικά επεξεργάζεται περαιτέρω σε άλλες πληροφορίες, όπως η ταχύτητα, η απόσταση και η θέση.

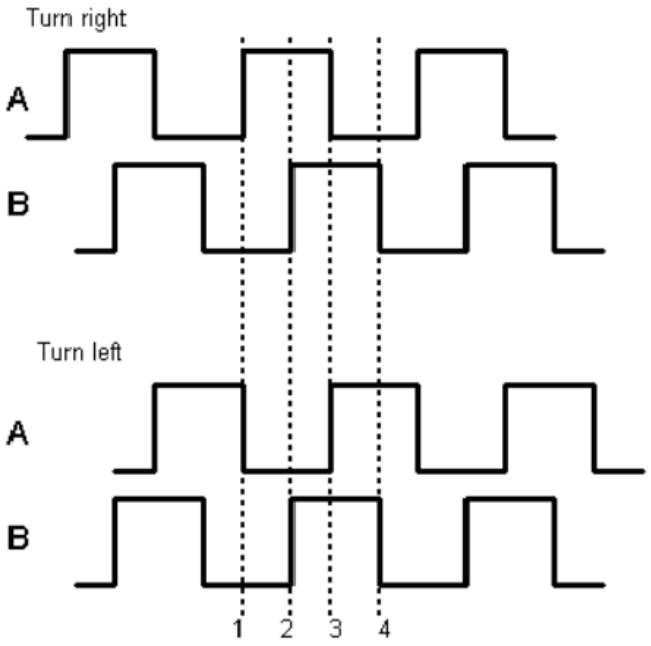


Εικόνα 3.12 : Περιστροφικός κωδικοποιητής

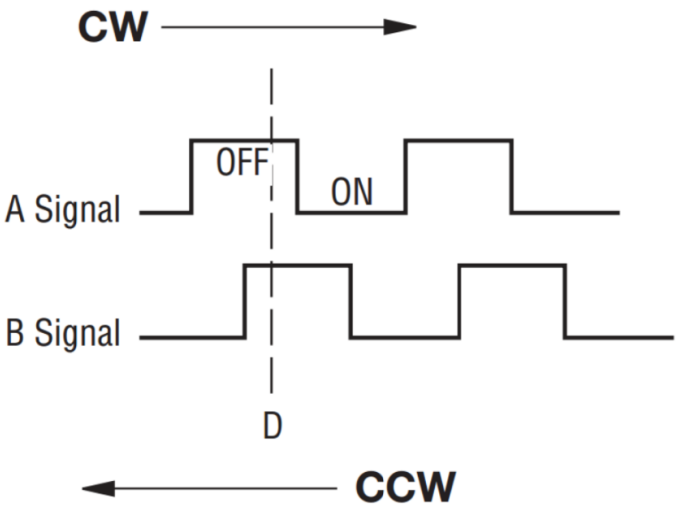
Για να γίνει πιο κατανοητή η λειτουργία ενός περιστροφικού αισθητήρα μπορούμε να παρακολουθήσουμε το παρακάτω πείραμα:

Χρησιμοποιούμε έναν βαθμιδωτό κωδικοποιητή ο οποίος ανήκει στην κατηγορία του περιστροφικού αισθητήρα, για να μετατρέψουμε την περιστροφική μετατόπιση σε μια σειρά ψηφιακών παλμών. Στη συνέχεια οι παλμοί θα

χρησιμοποιηθούν για τον έλεγχο της γωνιακής μετατόπισης. Ο κωδικοποιητής δημιουργεί τετράγωνα κύματα δύο φάσεων των οποίων η διαφορά φάσης είναι 90 °. Συνήθως τα τετράγωνα κύματα ονομάζονται κανάλι Α και κανάλι Β. Για λεπτομέρειες, υπάρχει το παρακάτω σχήμα.



Είναι δύσκολο να γίνει διάκριση ανάμεσα στην αριστερή και τη δεξιά στροφή κατά τον προγραμματισμό. Για αυτό το λόγο, χρησιμοποιώντας έναν παλμογράφο ο οποίος παρατηρεί την αριστερή και τη δεξιά στροφή ενός διακόπτη, βρίσκουμε μια διαφορά φάσης μεταξύ των σημάτων των δύο ακίδων εξόδου όπως φαίνεται παρακάτω.



Έτσι αν η έξοδος 1 και η έξοδος 2 είναι υψηλή, τότε ο διακόπτης περιστρέφεται δεξιόστροφα. Αν η έξοδος 1 είναι υψηλή και η έξοδος 2 χαμηλή, τότε ο διακόπτης περιστρέφεται αριστερόστροφα. Ως αποτέλεσμα, κατά τη διάρκεια του προγραμματισμού SCM, αν η έξοδος 1 είναι υψηλή, τότε μπορούμε να πούμε αν ο περιστροφικός κωδικοποιητής στρέφεται αριστερά ή δεξιά όσο γνωρίζουμε την κατάσταση της εξόδου 2.

Ο κώδικας προγραμματισμού είναι ο παρακάτω:

```
int redPin = 2;
int yellowPin = 3;
int greenPin = 4;
int aPin = 6;
int bPin = 7;
int buttonPin = 5;
int state = 0;
intlongPeriod = 5000;                    / / Time at green or red
intshortPeriod = 700;                    / / Time period when changing
int targetCount = shortPeriod;
int count = 0;
void setup ()
{
pinMode (aPin, INPUT);
pinMode (bPin, INPUT);
pinMode (buttonPin, INPUT);
pinMode (redPin, OUTPUT);
pinMode (yellowPin, OUTPUT);
pinMode (greenPin, OUTPUT);
}
```

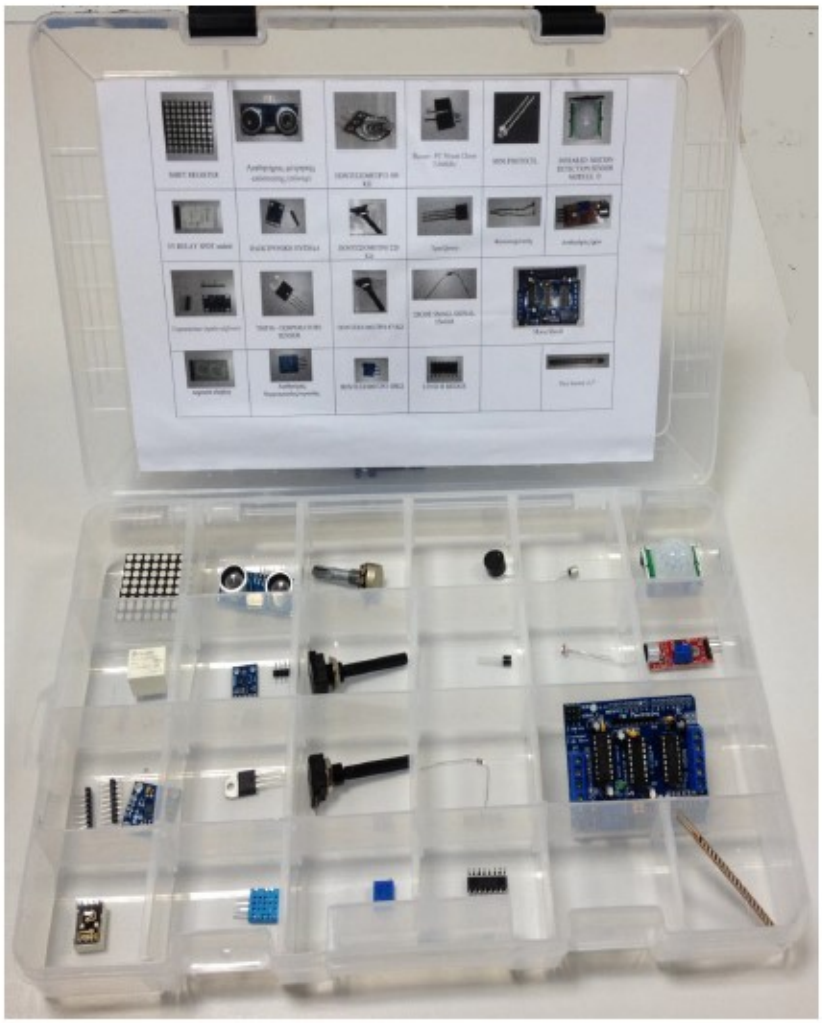
```
void loop ()
{
count + +;
if (digitalRead (buttonPin))
{
setLights (HIGH, HIGH, HIGH);
}
else
{
int change = getEncoderTurn ();
int newPeriod = longPeriod + (change * 1000);
if (newPeriod> = 1000 && newPeriod <= 10000)
{
longPeriod = newPeriod;
SunFounder
}
if (count> targetCount)
{
setState ();
count = 0;
}
}
delay (1);
}
int getEncoderTurn ()
{
/ / Return -1, 0, or +1
```

```
static int oldA = LOW;
static int oldB = LOW;
int result = 0;
int newA = digitalRead (aPin);
int newB = digitalRead (bPin);
if (newA! = oldA | | newB! = oldB)
{
/ / Something has changed
if (oldA == LOW && newA == HIGH)
{
result = - (oldB * 2 - 1);
}
}
oldA = newA;
oldB = newB;
return result;
}
int setState ()
{
if (state == 0)
{
setLights (HIGH, LOW, LOW);
targetCount = longPeriod;
state = 1;
}
else if (state == 1)
{
```

```
setLights (HIGH, HIGH, LOW);
targetCount = shortPeriod;
state = 2;
}
else if (state == 2)
SunFounder
{
setLights (LOW, LOW, HIGH);
targetCount = longPeriod;
state = 3;
}
else if (state == 3)
{
setLights (LOW, HIGH, LOW);
targetCount = shortPeriod;
state = 0;
}
}
void setLights (int red, int yellow, int green)
{
digitalWrite (redPin, red);
digitalWrite (yellowPin, yellow);
digitalWrite (greenPin, green);
}
```

4 ΚΕΦΑΛΑΙΟ: ΔΙΕΥΘΕΤΗΣΗ ΥΛΙΚΩΝ ARDUINO

Σε αυτό το κεφάλαιο παρουσιάζεται η διευθέτηση των υλικών Arduino, η τοποθέτηση τους μέσα σε ειδικά διαμορφωμένες θήκες και η τοποθέτηση ετικετών με το εικονίδιο και το όνομα του κάθε υλικού ώστε αυτό να είναι άμεσα αναγνωρίσιμο.



Εικόνα 4.1 : Διευθέτηση των υλικών Arduino

Επίλογος - Συμπεράσματα

Η παρούσα πτυχιακή εργασία από την μία αποτελεί χρήσιμο βοήθημα στους σπουδαστές της Α.Ε.Ν ΜΑΚΕΔΟΝΙΑΣ που θα θελήσουν να ασχοληθούν με το αντικείμενο του arduino, μιας και τα αισθητήρια που περιέχει καλύπτουν μεγάλο εύρος του υπάρχοντος υλικού και από την άλλη συμβάλλει στην διευκόλυνση λειτουργίας του εργαστηριού των Συστημάτων Αυτομάτου Ελέγχου. Το σίγουρο είναι πως παρέχει ένα μεγάλο αριθμό πληροφοριών που αφορούν στους αισθητήρες και αποτελεί ένα καλό οδηγό μελέτης.

Βιβλιογραφία

1. http://www.arduino.org/learning/reference
2. https://www.sunfounder.com/learn/category/sensor_kit_v1_for_Arduino.html
3. https://deltahacker.gr/arduino-intro/
4. https://www.youtube.com/
5. https://el.wikipedia.org/wiki/
6. https://t-h.wikispaces.com/file
7. Εγχειρίδιο ArduinoStartersSensorsKit37 in 1 box.

Περιεχόμενα