



ΤΕΧΝΟΛΟΓΙΚΟ
ΕΚΠΑΙΔΕΥΤΙΚΟ
ΙΔΡΥΜΑ
ΤΕΙ ΗΠΕΙΡΟΥ

ΤΕΙ ΗΠΕΙΡΟΥ

ΣΧΟΛΗ ΤΕΧΝΟΛΟΓΙΚΩΝ ΕΦΑΡΜΟΓΩΝ

ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ ΤΕ

ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ

**ΜΕΛΕΤΗ ΚΑΙ ΚΑΤΑΣΚΕΥΗ ΑΝΙΧΝΕΥΣΗΣ ΕΜΠΟΔΙΟΥ ΚΑΙ
ΠΡΟΣΔΙΟΡΙΣΜΟΥ ΤΗΣ ΣΧΕΤΙΚΗΣ ΑΠΟΣΤΑΣΗΣ ΜΕ ΧΡΗΣΗ
ΜΙΚΡΟΕΛΕΓΚΤΗ**

Καλλιόπη Τσαχάκη

Επιβλέπων καθηγητής: Κωνσταντίνος Αγγέλης

**DESIGN AND IMPLEMENTATION OF OBSTACLE
DETECTION AND DISTANCE MEASUREMENT USING
MICROCONTROLLER TECHNOLOGY**

Περιεχόμενα

Περιεχόμενα	3
Πίνακας Εικόνων	5
1. Εισαγωγή.....	6
1.1. Εισαγωγή	6
1.2. Σκοπός	6
1.3. Μεθοδολογία	7
1.4. Δομή εργασίας	7
1.5. Βασικά κεφάλαια.....	7
2. Μικροελεγκτές.....	9
2.1. Εισαγωγή.....	9
2.2. Μικροελεγκτές.....	9
2.3. Πλατφόρμα Arduino	10
2.4. Ο μικροελεγκτής του Arduino	12
2.5. Θύρες I/O Arduino	12
2.6. Διάταξη πλακέτας Arduino	13
2.7. Τροφοδοσία	15
2.8. Ενσωματωμένα κουμπιά και LED	16
2.9. Προγραμματισμός σε Arduino	16
2.9.1. Περιβάλλον προγραμματισμού Arduino IDE	17
2.9.2. Εγκατάσταση περιβάλλοντος Arduino IDE.....	18
2.9.3. Δηλώσεις μεταβλητών	19
2.9.4. Συναρτήσεις διαχείρισης θυρών εισόδου – εξόδου (Pins)	19
3. Σχεδιασμός του συστήματος	23
3.1. Εισαγωγή.....	23
3.2. Αισθητήρας απόστασης	24
3.3. Συνδεσμολογία.....	26
3.4. Συγγραφή προγράμματος	28
3.5. Οθόνη LCD	30
3.6. Ασύρματη μετάδοση.....	34

3.6.1. Κύκλωμα ESP8266	34
3.6.2. Arduino web server.....	35
4. Συμπεράσματα.....	41
4.1. Αποτίμηση Arduino.....	41
4.2. Πλεονεκτήματα – Μειονεκτήματα	41
4.3. Αποτίμηση εργασίας.....	42
4.4. Μελλοντική εργασία.....	43
Βιβλιογραφία	44

Πίνακας Εικόνων

Εικόνα 1. Arduino Uno	10
Εικόνα 2. Περιεχόμενα Arduino starter kit	11
Εικόνα 3. Πλακέτα Arduino.....	13
Εικόνα 4. Τροφοδοσία.....	15
Εικόνα 5. Arduino	23
Εικόνα 6. Λειτουργία αισθητήρα	24
Εικόνα 7. HC-SR04 Ultrasonic	24
Εικόνα 8. Έξοδος Trig / Είσοδος Echo	25
Εικόνα 9. Υπολογισμός απόστασης.....	25
Εικόνα 10. Συνδεσμολογία	26
Εικόνα 11. Κατασκευή	27
Εικόνα 12. Arduino IDE	29
Εικόνα 13. Έλεγχος με την παρακολούθηση της σειριακής.....	29
Εικόνα 14. Συνδεσμολογία Οθόνης.....	30
Εικόνα 15. Ηλεκτρονικό σχέδιο	31
Εικόνα 16. Σύνδεση Αισθητήρα και Οθόνης.....	33
Εικόνα 17. ESP8266 (ESP-01).....	34
Εικόνα 18. Ακροδέκτες ESP8266 (ESP-01)	35
Εικόνα 19. Σύνδεση με τοπικό ασύρματο δίκτυο	40

1. Εισαγωγή

1.1. Εισαγωγή

Η συγκεκριμένη εργασία πραγματεύεται την κατασκευή ενός μηχανισμού ανίχνευσης εμποδίου και προσδιορισμού της σχετικής απόστασης με τη χρήση ενός μικροελεγκτή. Ως λύση για την κατασκευή επιλέχθηκε ο μικροελεγκτής Arduino σε συνδυασμό με έναν διαδεδωμένο αισθητήρα απόστασης με τη χρήση υπερήχων. Στην εργασία παρουσιάζεται τόσο ο τρόπος κατασκευής όσο και ο κώδικας προγραμματισμού του μικροελεγκτή.

1.2. Σκοπός

Σκοπός της συγκεκριμένης εργασίας είναι η ανάπτυξη μιας εφαρμογής ανίχνευσης εμποδίου με τη χρήση της πλατφόρμας Arduino. Στα πλαίσια της εργασίας αρχικά γίνεται μια προεπισκόπηση των βασικών συστατικών του Arduino και του σχεδιασμού του. Γίνεται αναλυτική επεξήγηση της λειτουργίας του και αρχικός σκοπός είναι να διερευνηθεί σε βάθος ο τρόπος κατασκευής ενός αυτοματισμού με τη χρήση της πλατφόρμας.

Ο βασικός σκοπός είναι επίσης ο σχεδιασμός και η ανάπτυξη του μηχανισμού που ανιχνεύει εμπόδια. Για το λόγο αυτό μελετάται ο τρόπος με τον οποίο συνδέονται διάφορες περιφερειακές συσκευές όπως αισθητήρες, οθόνη LCD κα, και πως αυτές οι συσκευές μπορούν να λειτουργούν διαδραστικά με τον μικροελεγκτή του Arduino στέλνοντας ή λαμβάνοντας δεδομένα. Επίσης επιλέγονται εκείνα τα ηλεκτρονικά στοιχεία που απαιτούνται για την ολοκλήρωση της κατασκευής.

Τέλος στόχος της εργασίας είναι η ολοκλήρωση ενός μηχανισμού με τη σύνδεση των εξωτερικών μονάδων στο Arduino και τον κατάλληλο προγραμματισμό του έτσι ώστε να γίνεται η καταγραφή της απόστασης ενός εμποδίου από τον αισθητήρα και εμφάνισή του με κάποιο τρόπο στον χρήστη. Επίσης γίνεται διερεύνηση της δυνατότητας να μεταδίδονται τα δεδομένα με ασύρματη μετάδοση μέσω ενός ασύρματου τοπικού δικτύου WiFi.

1.3. Μεθοδολογία

Η εργασία βασίστηκε αρχικά σε βιβλιογραφική έρευνα και σε έρευνα στο Διαδίκτυο για την εύρεση των κατάλληλων πηγών, δεδομένου ότι το Arduino υποστηρίζεται από μια μεγάλη κοινότητα χρηστών. Επίσης το εγχειρίδιο χρήσης του Arduino Starter kit αποτέλεσε την αρχική πηγή αναφοράς και εξοικείωσης στα πρώτα στάδια με την πλατφόρμα.

Στη συνέχεια η έρευνα ήταν κυρίως πρωτογενής με τη χρήση παραδειγμάτων και πειραματισμών χρησιμοποιώντας στην πράξη τον μικροελεγκτή, τα περιφερειακά εξαρτήματα και το περιβάλλον προγραμματισμού του Arduino. Ξεκινώντας από απλά παραδείγματα με τη χρήση λυχνιών LED και σύνδεση του Arduino με ένα breadboard, σταδιακά υλοποιήθηκαν ενδιάμεσα μικρά παραδείγματα μέχρι την υλοποίηση του μηχανισμού ανίχνευσης απόστασης.

Τέλος διερευνήθηκε η δυνατότητα εμφάνισης των αποτελεσμάτων σε οθόνη LCD καθώς και η αποστολή των δεδομένων μέσω της ασύρματης σύνδεσης του Arduino με ένα τοπικό δίκτυο σε άλλους σταθμούς εργασίας.

1.4. Δομή εργασίας

Στο πρώτο κεφάλαιο γίνεται μια γενική αναφορά στους μικροελεγκτές και στη συνέχεια ακολουθεί μια λεπτομερής περιγραφή των τεχνικών χαρακτηριστικών του Arduino. Περιγράφονται τα ηλεκτρονικά στοιχεία της πλακέτας καθώς και το λογισμικό του. Τέλος περιγράφεται η μέθοδος προγραμματισμού και βασικοί παράμετροι της γλώσσας προγραμματισμού του Arduino.

Στο επόμενο κεφάλαιο περιγράφεται βήμα προς βήμα ο σχεδιασμός και η κατασκευή του κυκλώματος καθώς και ο κώδικας για την αποστολή των δεδομένων του αισθητήρα στη σειριακή θύρα του Arduino. Παρουσιάζεται η επιτυχής μεταγλώττιση του προγράμματος και η εμφάνιση των αποτελεσμάτων στην κονσόλα παρακολούθησης της σειριακής θύρας επικοινωνίας. Επίσης γίνεται παρουσίαση της συνδεσμολογίας και του αντίστοιχου κώδικα για την εμφάνιση των αποτελεσμάτων σε οθόνη LCD. Τέλος διερευνάται η δυνατότητα αποστολής των δεδομένων ασύρματα.

Στο τελευταίο κεφάλαιο γίνεται μια αποτίμηση της πλατφόρμας με βάση τα πλεονεκτήματα και τα μειονεκτήματα της. Επίσης γίνεται ένας σύντομος απολογισμός της εργασίας όσον αφορά στην επίτευξη των στόχων της. Τέλος προτείνονται κάποια επόμενα στάδια για την επέκταση των δυνατοτήτων του αυτοματισμού ως μελλοντική εργασία.

1.5. Βασικά κεφάλαια

Τα βασικά κεφάλαια της συγκεκριμένης εργασίας περιλαμβάνουν τα εξής:

- Τεκμηρίωση της εργασίας με περιγραφή της κατασκευής βήμα προς βήμα καθώς και του τρόπου λειτουργίας του μηχανισμού.
- Την κατασκευή του κυκλώματος που αποτελείται από το Arduino και τον αισθητήρα απόστασης.
- Τον πρόγραμμα που φορτώθηκε στο Arduino για τη λειτουργία του κυκλώματος.

2. Μικροελεγκτές

2.1. Εισαγωγή

Η παρούσα εργασία αφορά την ανάπτυξη ενός αυτόνομου συστήματος ανίχνευσης απόστασης με τη χρήση ενός μικροελεγκτή όπως το Arduino. Οι μικροελεγκτές είναι μικρά υπολογιστικά συστήματα, σχεδιασμένα σε ένα και μόνο ολοκληρωμένο κύκλωμα ολοκλήρωσης υψηλής κλίμακας.

2.2. Μικροελεγκτές

Ένας μικροελεγκτής αποτελεί ένα ολοκληρωμένο υπολογιστικό σύστημα. Όπως κάθε υπολογιστικό σύστημα, περιέχει την κεντρική μονάδα επεξεργασίας, έναν σύνολο καταχωρητών, μνήμη και κύκλωμα ελέγχου περιφερειακών συσκευών. Ο μικροελεγκτής, εκτός του ότι μπορεί να εκτελέσει αριθμητικές και λογικές πράξεις, μπορεί καταχωρήσει τιμές σε μνήμη RAM, να επικοινωνήσει με συσκευές και με το εξωτερικό περιβάλλον. Έτσι τα βασικά στοιχεία ενός μικροελεγκτή είναι:

- αριθμητική και λογική μονάδα (ALU).
- μονάδα ελέγχου και αποκωδικοποίησης εντολών.
- καταχωρητές ειδικού σκοπού όπως συσσωρευτή, μετρητή προγράμματος, καταχωρητή εντολών.
- εσωτερικούς χρονιστές - απαριθμητές.

Πέραν των βασικών αυτών στοιχείων, ένας μικροελεγκτής περιέχει:

- η μνήμη τύπου (ROM ή EPROM) και
- η μνήμη τύπου RAM.

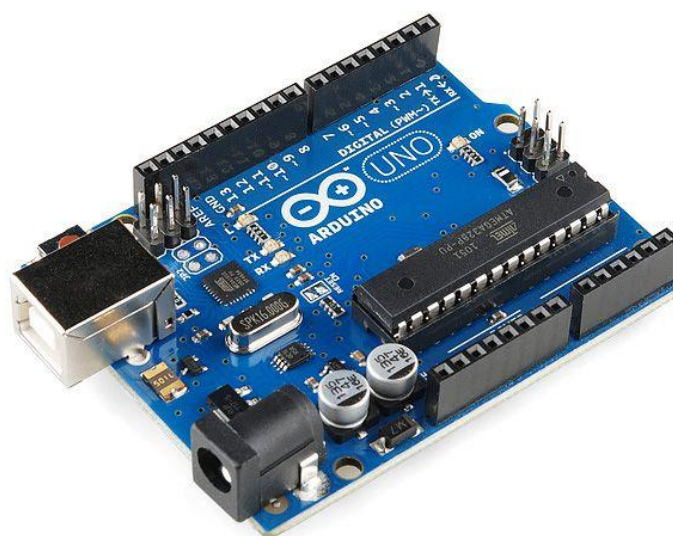
Τέλος, ένας μικροελεγκτής χρησιμοποιεί παράλληλες θύρες I/O αλλά και περιφερειακά κυκλώματα για την επικοινωνία με το εξωτερικό περιβάλλον.

Μέσα από τις θύρες εισόδου κι εξόδου ένας μικροελεγκτής μπορεί να δέχεται σήματα εισόδου με τη μορφή λογικών ψηφιακών καταστάσεων ή μπορεί να αποστέλλει σήματα σε άλλες συσκευές μέσα από θύρες εξόδου. Η ανταλλαγή δεδομένων με τις

συσκευές επιτυγχάνεται με ασύγχρονη ή σύγχρονη σειριακή επικοινωνία μέσα από ένα σύστημα σημάτων διακοπών. Πολλές περιφερειακές συσκευές χρησιμοποιούνται ως μετατροπείς αναλογικών σημάτων σε ψηφιακά για την χρησιμοποίησή τους από τον μικροελεγκτή, όπως αισθητήρες απόστασης, θερμοκρασίας, υγρασίας, φωτός. Επίσης ο μικροελεγκτής μπορεί να διαχειριστεί πολλές αναλογικές συσκευές όπως διόδους LED, ηλεκτρονόμους, ηλεκτρικούς κινητήρες και άλλα κυκλώματα, που συνήθως χρησιμοποιούνται σε αυτοματισμούς (Raj Kamal, 2009).

2.3. Πλατφόρμα Arduino

Η χρήση κι ο προγραμματισμός μικροελεγκτών αποτελούσε μια δύσκολη διαδικασία και απαιτούσε εξειδικευμένους προγραμματιστές. Επίσης η χρήση των μικροελεγκτών για την κατασκευή κάποιου αυτοματισμού προϋπέθετε γνώση ηλεκτρονικών και προγραμματισμό μικροεπεξεργαστών. Τα τελευταία χρόνια δημιουργήθηκαν πλατφόρμες κατασκευής αυτοματισμών πιο φιλικές προς αρχάριους χρήστες και προγραμματιστές. Έτσι το Arduino αποτέλεσε μια πλατφόρμα «ανοικτού κώδικα» για την κατασκευή αυτοματισμών χωρίς την απαίτηση ιδιαίτερων γνώσεων προγραμματισμού και ηλεκτρονικών. Προγραμματιστές χωρίς εξειδικευμένες προγραμματιστικές γνώσεις με βασικές γνώσεις ηλεκτρονικών μπορούν σχετικά να κατασκευάσουν αυτοματισμούς χρησιμοποιώντας πολύ φθηνά και τυποποιημένα υλικά όπως αισθητήρες, κινητήρες, λυχνίες LED και άλλα ηλεκτρονικά.



Εικόνα 1. Arduino Uno

Το Arduino μπορεί να μην είναι η πιο ισχυρή πλατφόρμα ανάπτυξης αυτοματισμών όμως η ευκολία στη χρήση του σε σχέση με τα άλλα συστήματα καθώς και ο βαθμός στον οποίο το συγκεκριμένο σύστημα είναι «ανοικτό», έχει συμβάλλει στο να δημιουργηθεί μια μεγάλη κοινότητα που το υποστηρίζει κατασκευάζοντας εφαρμογές και βιβλιοθήκες που διανέμονται online.



Ο επίσημος κατασκευαστής του πρωτότυπου Arduino διαθέτει μια έκδοση που απευθύνεται κατασκευαστές ηλεκτρονικών χωρίς ιδιαίτερη εμπειρία, το Arduino Starter Kit, το οποίο περιέχει το ίδιο το Arduino Uno συνοδευόμενο από άλλα ηλεκτρονικά εξαρτήματα (καλώδια, διακόπτες, αντιστάσεις, LED, ποτενσιόμετρα, κ.α.) και υλικά (πχ καλώδιο USB για την σύνδεση με υπολογιστή) απαραίτητα για βασικές εφαρμογές.

2.4. Ο μικροελεγκτής του Arduino

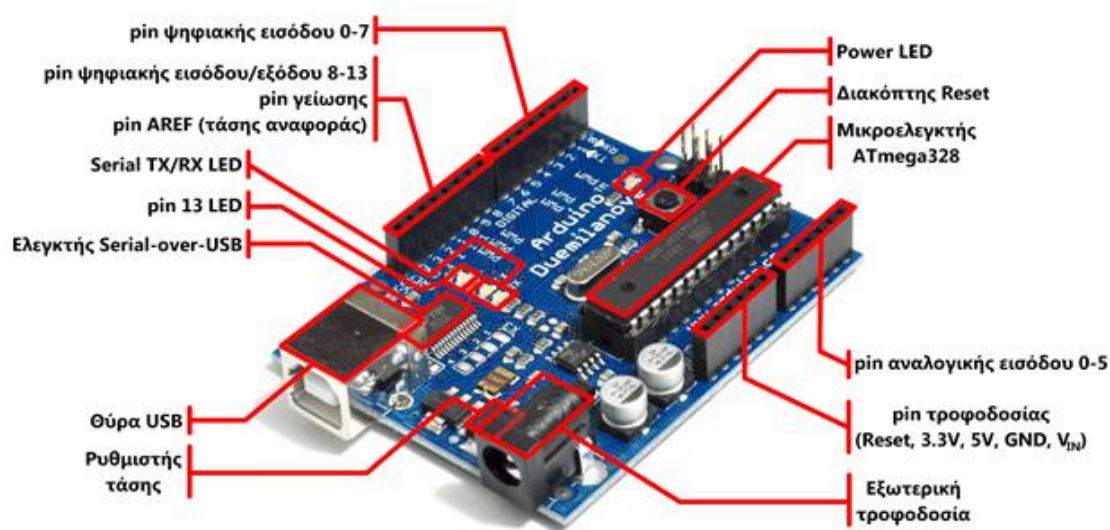
Το Arduino βασίζεται σε έναν μικροελεγκτή 8-bit τεχνολογίας RISC, τον ATmega328 με συχνότητα ρολογιού στα 16MHz. Ο επεξεργαστής διαθέτει ενσωματωμένη μνήμη της εξής μορφής:

- Μνήμη SRAM μεγέθους 2Kb. Χρησιμοποιείται για την αποθήκευση δεδομένων των μεταβλητών και των πινάκων κατά την εκτέλεση του κώδικα που έχει φορτωθεί στο Arduino. Όπως και με κάθε μνήμη RAM, το περιεχόμενό της χάνεται όταν διακοπεί η παροχή ρεύματος στο Arduino ή όταν γίνει reset.
- Μνήμη EEPROM μεγέθους 1Kb. Χρησιμοποιείται για εγγραφή και ανάγνωση δεδομένων εκτός τύπων και δομών δεδομένων και είναι προσπελάσιμη από τα προγράμματά κατά την εκτέλεσή τους. Η EEPROM αντίθετα με τη SRAM διατηρεί τα περιεχόμενά της και μετά την διακοπή της τροφοδοσίας ή το reset και συνεπώς λειτουργεί ως αποθηκευτικό μέσο.
- Μνήμη τύπου Flash 32Kb μεγέθους 32Kb. Τα 2Kb από τα 32 χρησιμοποιούνται από το bootloader του Arduino και έχουν δεσμευτεί από τον κατασκευαστή. Το bootloader είναι το firmware του κατασκευαστή για το Arduino και είναι απαραίτητο για την φόρτωση των προγραμμάτων του προγραμματιστή στον μικροελεγκτή μέσω της θύρας USB. Σε διαφορετική περίπτωση θα ήταν αναγκαία η φόρτωση των προγραμμάτων στο hardware από έναν εξειδικευμένο hardware προγραμματιστή. Για την αποθήκευση των προγραμμάτων που δημιουργεί ένας προγραμματιστής αφού φορτωθούν από τον bootloader χρησιμοποιούνται τα υπόλοιπα 30Kb της μνήμης Flash. Η μνήμη Flash διατηρεί τα δεδομένα της ακόμα και μετά από διακοπή τροφοδοσίας ή reset. Έτσι ένα πρόγραμμα που έχει φορτωθεί μια φορά στη μνήμη flash, μπορεί να επανεκκινήσει την εκτέλεσή του μετά από reset.

2.5. Θύρες I/O Arduino

Το σύστημα διεπαφής του Arduino είναι σειριακή. Ο μικροελεγκτής ATmega υποστηρίζει σειριακή επικοινωνία. Αυτή επιτυγχάνεται με τη χρήση ενός ελεγκτή Serial-over-USB ώστε το Arduino να συνδέεται μέσω USB με τον υπολογιστή. Η

σύνδεση αυτή έχει πολλές χρήσεις και λειτουργεί ανάλογα είτε ως θύρα εισόδου ή ως θύρα εξόδου. Μετά την μεταγλώττιση για παράδειγμα ενός προγράμματος, για να μπορέσει να φορτωθεί στη μνήμη του Arduino, χρησιμοποιείται η σύνδεση μ'εσω USB. Αντίστοιχα με την εντολή `Serial.print()` επιτυγχάνεται αμφίδρομη επικοινωνία του Arduino με τον υπολογιστή έτσι ώστε να εμφανίζονται στην κονσόλα που αντιστοιχεί στη θύρα σύνδεσης του υπολογιστή δεδομένα εξερχόμενα από το Arduino την ώρα που εκτελείται κάποιο πρόγραμμα.



Εικόνα 3. Πλακέτα Arduino

2.6. Διάταξη πλακέτας Arduino

Η πλακέτα του Arduino αποτελείται επίσης από 14 pin, αριθμημένα από 0 ως 13. Λειτουργούν ως ψηφιακές εισοδοι και έξοδοι. Η τάση λειτουργίας είναι στα 5V και η μέγιστη ένταση ρεύματος που μπορούν να δεχτούν είναι 40mA. Όταν ένα από αυτά τα pin λειτουργούν σαν ψηφιακή έξοδος, μπορεί να διοχετευθεί ρεύμα στο pin μέσω εντολής του προγράμματος που το θέτει σε κατάσταση HIGH. Για να διακοπεί η τροφοδοσία τότε τίθεται σε κατάσταση LOW. Έτσι μπορεί να δημιουργηθεί ένα κύκλωμα με τη σύνδεση κάποιου εξαρτήματος όπως είναι μια λυχνία LED σε ένα από τα pin. Συνδέοντας το ένα ακροδέκτη της λυχνίας στο pin και το άλλο στη γείωση, διοχετεύεται ρεύμα μέσω του προγράμματος στο pin με την αντίστοιχη εντολή που το θέτει σε κατάσταση HIGH και έτσι το LED ανάβει.

Κάποια από τα pin, εκτός από ψηφιακές θύρες I/O μπορούν να χρησιμοποιηθούν και με άλλους τρόπους:

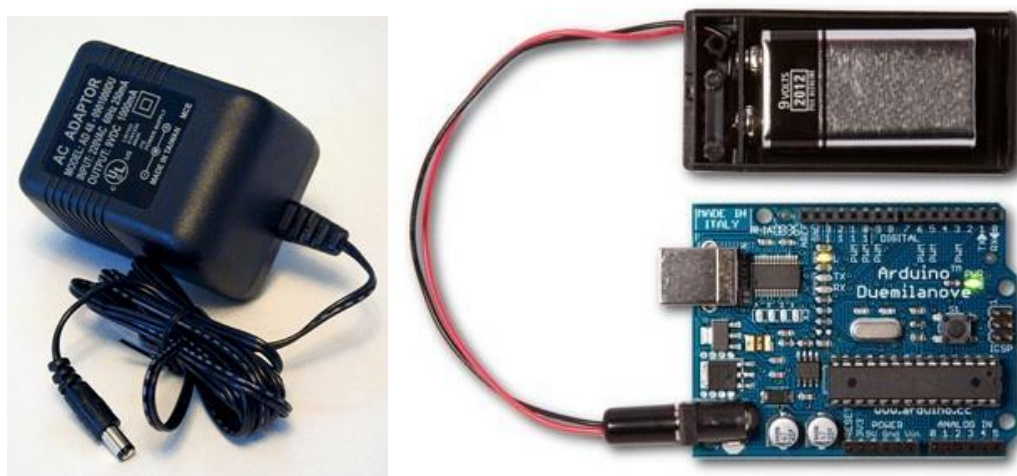
- Τα pin 0 και 1 μπορούν να χρησιμοποιηθούν για την αποστολή και λήψη αντίστοιχα δεδομένων στη θύρα USB μέσω του ελεγκτή Serial-Over-USB ως RX και TX. Το πρόγραμμα θα πρέπει να ενεργοποιήσει τη λειτουργία και τότε τα δεδομένα που στέλνονται στην σειριακή, διοχετεύονται και στο pin 0 για να τα λάβει κάποια άλλη συσκευή όπως ένας δεύτερος μικροελεγκτής.
- Τα pin 2 και 3 μπορούν να ρυθμιστούν μέσω εντολών του προγράμματος έτσι ώστε να λειτουργήσουν ως interrupts δίνοντας την δυνατότητα διακοπής της ροής του προγράμματος για να εκτελεστεί μια συγκεκριμένη συνάρτηση.
- Κάποια pin λειτουργούν και ως αναλογικές έξοδοι. Έτσι τα pin 3, 5, 6, 9, 10 και 11 φέρουν την ένδειξη ANALOG και με το σύστημα PWM (Pulse Width Modulation), μπορεί να μεταβάλλεται η τάση που διοχετεύεται. Το σύστημα PWM δεν είναι πραγματικά αναλογικό. Σε μια κλίμακα από τιμή 0 ως 255, θέτοντας στην έξοδο του pin την τιμή 127, στην πραγματικότητα δεν διοχετεύεται τάση 2.5V αντί τάσης 5V. Στην πραγματικότητα αυτό που συμβαίνει είναι ότι δίνεται ένας παλμός που εναλλάσσει με μεγάλη συχνότητα την τάση μεταξύ των τιμών 0 και 5V. Έτσι αν συνδεθεί ένα LED σε αυτό το pin, δίνεται η εντύπωση της χαμηλότερης ή υψηλότερης φωτεινότητας σε μια κλίμακα 256 καταστάσεων από 0 που σημαίνει καθόλου φωτεινότητα μέχρι 255 που δίνει την εντύπωση της πλήρους φωτεινότητας.

Εκτός από τα 14 pin που λειτουργούν ως ψηφιακές I/O, υπάρχουν επίσης έξι αναλογικά pin εισόδου με την σήμανση ANALOG IN, με αρίθμηση από το 0 ως το 5. Η λειτουργία τους ως αναλογική είσοδος επιτυγχάνεται μέσω ενός ενσωματωμένου μετατροπέα από αναλογικό σε ψηφιακό σήμα (Analog to Digital Converter ADC) ο οποίος μπορεί να τροφοδοτήσει ένα από αυτά τα pin με μια τάση από 0 Volt ως μια τάση αναφοράς Vref η οποία προεπιλεγμένα είναι ίση με 5Volt. Στον κώδικα με την αντίστοιχη εντολή μπορεί να ανιχνευθεί η τιμή του pin ως ακέραιος αριθμός μήκους 10-bit, με μικρότερη τιμή 0 μέχρι 1023 για τάση στο pin 5Volt. Επίσης τα έξι αυτά pin, με την αντίστοιχη εντολή στον κώδικα μπορεί να μετατραπεί σε ψηφιακό pin εισόδου/εξόδου και να λειτουργούν με παρόμοιο τρόπο όπως τα υπόλοιπα 14 που

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

2.7. Τροφοδοσία

Η τροφοδοσία του Arduino είναι από 7 ως 12V DC. Η πηγή τροφοδοσίας μπορεί να είναι ένας μετασχηματιστής που συνδέεται στο οικιακό ρεύμα, ή μπαταρίες ή οποιαδήποτε άλλη πηγή DC. Το Arduino μπορεί να τροφοδοτηθεί επίσης από τον υπολογιστή μέσω της σύνδεσης USB.



Εικόνα 4. Τροφοδοσία

Στην πλακέτα υπάρχει μια σειρά που αποτελείται από 6 pin με την ετικέτα POWER. Η λειτουργία του καθενός έχει ως εξής:

- Το πρώτο pin έχει την ένδειξη RESET και αν γειωθεί σε ένα από τα pin με ένδειξη GND (Ground – γείωση) γίνεται επανεκκίνηση του Arduino.
- Το δεύτερο pin φέρει την ένδειξη 3.3V. Αν κάποιο περιφερειακό εξάρτημα συνδεθεί με αυτό το pin, τότε τροφοδοτείται με τάση 3.3V από τον ελεγκτή Serial-over-USB μέσω του Arduino με μέγιστη ένταση ρεύματος 50mA.
- Το τρίτο pin φέρει την ένδειξη 5V. Αν κάποιο περιφερειακό εξάρτημα συνδεθεί με αυτό το pin, τότε τροφοδοτείται με τάση 5V είτε από τον ελεγκτή Serial-over-USB μέσω του Arduino ή από την εξωτερική τροφοδοσία αφού αυτή περάσει από έναν μετασχηματιστή τάσης στα 5V.
- Το τέταρτο και το πέμπτο pin είναι γειώσεις και φέρουν τη σήμανση GND.

- Το έκτο pin, με την ένδειξη Vin μπορεί να χρησιμοποιηθεί με δύο τρόπους. Μπορεί να χρησιμοποιηθεί ως μέθοδος εξωτερικής τροφοδοσίας του Arduino, όταν δεν είναι συνδεδεμένη η υποδοχή εξωτερικής τροφοδοσίας, σε συνδυασμό με το pin γείωσης.
- Από την άλλη αν είναι ήδη συνδεδεμένη η εξωτερική τροφοδοσία τότε το pin λειτουργεί ως πηγή τροφοδοσίας για περιφερειακά εξαρτήματα με την πλήρη τάση της εξωτερικής τροφοδοσίας (7~12V) κάτι που δεν επιτυγχάνεται με το pin των 5V και τον μετασχηματιστή τάσης.

2.8. Ενσωματωμένα κουμπιά και LED

Στην πλακέτα του Arduino διακρίνονται κάποια στοιχεία όπως ο ενσωματωμένος διακόπτης RESET ο οποίος και προφανώς χρησιμοποιείται για τη λειτουργία της επανεκκίνησης. Επίσης υπάρχουν κάποια επιφανειακά LED. Το LED με την ένδειξη POWER είναι αναμμένο όταν η πλακέτα τροφοδοτείται με ρεύμα. Τα LED που φέρουν τις ενδείξεις TX και RX λειτουργούν κατά τη λειτουργία της σειριακής επικοινωνίας του Arduino κατά την αποστολή ή λήψη δεδομένων μέσω της θύρας USB και ελέγχονται από τον αντίστοιχο ελεγκτή σειριακής επικοινωνίας.

Τέλος, υπάρχει ένα ενσωματωμένο LED με την ένδειξη L. Το LED αυτό είναι συνδεδεμένο με το pin 13 και χρησιμοποιείται κυρίως ως ενδεικτικό της λειτουργίας του Arduino ακόμα κι όταν δεν έχει συνδεθεί κάποιο άλλο εξάρτημα πάνω σε αυτό. Έτσι μπορεί να γραφτεί κάποιο βασικό πρόγραμμα που θα μπορεί να ανάψει το LED αυτό αναθέτοντάς την τιμή HIGH στο φυσικό pin 13. Θα μπορούσε κάποιος να παρομοιάσει τη λειτουργία του με εκείνα τα βασικά προγράμματα σε οποιαδήποτε πλατφόρμα προγραμματισμού που εμφανίζουν τη φράση “Hello World”, μετά την πρώτη επιτυχή μεταγλώττιση (Massimo Banzi,2011).

2.9. Προγραμματισμός σε Arduino

Το κύκλωμα των μονάδων του Arduino βασίζεται όπως αναφέραμε στην ανοικτή τεχνολογία, δηλαδή ο σχεδιασμός και τα συστατικά του μέρη περιγράφονται από τους κατασκευαστές του, με σκοπό να μπορεί να αναπαραχθεί από άλλους σχεδιαστές.

Έτσι, εκτός από την πρωτότυπη μονάδα με την ονομασία Arduino που προέρχεται από τους δημιουργούς και επίσημους κατασκευαστές του στην Ιταλία, υπάρχουν πάρα πολλές ακόμα υλοποιήσεις μονάδων του, απόλυτα συμβατές με τα προγράμματα και κυκλώματα που ενδεχομένως ήδη. Η μοναδική δέσμευση είναι να αναφέρονται με άλλη ονομασία οι κατασκευές τρίτων, γι αυτό και κυκλοφορούν πολλές άλλες εκδόσεις οι οποίες συνήθως έχουν ονόματα που καταλήγουν σε -ino, όπως μια από τις κινέζικες εκδοχές του το Funduino.

2.9.1 Περιβάλλον προγραμματισμού Arduino IDE

Όπως θα δούμε στη συνέχεια, η εφαρμογή συγγραφής προγραμμάτων και διαχείριση του Arduino είναι το Arduino IDE, το οποίο και διατίθεται από την επίσημη ιστοσελίδα και για τις 3 πλατφόρμες Windows, MacOS και Linux. Μπορεί να γίνει καταβίβασή του εντελώς δωρεάν από την επίσημη ιστοσελίδα (<http://arduino.cc/en/Main/Software>). Έχει αναπτυχθεί σε Java αν και η γλώσσα προγραμματισμού που χρησιμοποιείται βασίζεται στη C. Πρόκειται για ένα περιβάλλον που αποτελείται:

- Τον συντάκτη των προγραμμάτων. Δεν προσφέρει κάποια συντακτική αναγνώριση και διόρθωση αλλά διαφορετική χρωματική σήμανση. Τα προγράμματα τα οποία εξάγονται είναι απλά script κειμένου που ονομάζονται sketch.
- Τον μεταγλωττιστή των προγραμμάτων scetch ο οποίος κι ελέγχει την ορθότητα των προγραμμάτων. Επίσης υπάρχει και οθόνη εμφάνισης των σφαλμάτων ή της επιτυχούς μεταγλώττισης ή φόρτωσης των προγραμμάτων.
- Έτοιμες βιβλιοθήκες που έχουν προ-εγκατασταθεί και μπορούν να συμπεριληφθούν από την επιλογή Συμπερίληψη. Πέραν αυτών υπάρχει η δυνατότητα προσθήκης πολλών βιβλιοθηκών που έχουν αναπτυχθεί από προγραμματιστές Arduino και κυκλοφορούν είτε επίσημα ή ανεξάρτητα από την επίσημη ιστοσελίδα στο Διαδίκτυο. Οι βιβλιοθήκες προσφέρουν και έτοιμα παραδείγματα κώδικα με σχολιασμό των εντολών.

- Κονσόλα παρακολούθησης (serial monitor) της σειριακής θύρας USB, η οποία σε συνδυασμό με την εντολή serial.print μπορεί να χρησιμοποιηθεί για την παρακολούθηση της εκτέλεσης του κώδικα.
- Μενού επιλογών με διάφορες ρυθμίσεις ανάλογα με την έκδοση του Arduino.

2.9.1 Εγκατάσταση περιβάλλοντος Arduino IDE

Για τον προγραμματισμό της μονάδας Arduino απαιτείται όπως αναφέραμε το περιβάλλον προγραμματισμού Arduino IDE. Στο περιβάλλον αυτό συντάσσονται τα προγράμματα σε γλώσσα προγραμματισμού που βασίζεται στη γλώσσα C/C++. Ο κώδικας αρχικά μεταγλωττίζεται και στη συνέχεια μεταφορτώνεται στη μονάδα Arduino. Υπάρχει επίσης εξελληνισμένο μενού, καθώς και αρκετά έτοιμα παραδείγματα χρήσης βασικών λειτουργιών ενσωματωμένα μέσα στην εγκατάσταση (Αρχείο => Παραδείγματα).

Η λογική σύνταξης των προγραμμάτων του Arduino είναι απλή. Απαιτείται η ύπαρξη δύο βασικών συναρτήσεων:

setup() και η loop()

Οι συναρτήσεις αυτές λειτουργούν ως εξής:

setup(): Στη συνάρτηση αυτή μπαίνουν όλες οι εντολές που πρέπει να τρέξουν μία φορά κατά την ενεργοποίηση της μονάδας (όταν δηλαδή τροφοδοτηθεί με ρεύμα ή όταν πατηθεί το πλήκτρο reset που υπάρχει). Συνήθως μπαίνουν αρχικοποιήσεις τιμών μεταβλητών και οπωσδήποτε ο χαρακτηρισμός των εισόδων/εξόδων που θα χρησιμοποιηθούν (αν δηλαδή ένα συγκεκριμένο Pin θα είναι είσοδος ή έξοδος).

loop(): Στη συνάρτηση αυτή εισάγεται το βασικό πρόγραμμά το οποίο και θα εκτελείται επαναλαμβανόμενα. Αφού εκτελεσθούν οι εντολές θα ενεργοποιηθεί ξανά η loop(), επαναλαμβάνοντας από την αρχή το πρόγραμμα. Αυτό θα συμβαίνει συνεχώς, όσο έχει ρεύμα το Arduino ή μέχρι να πατηθεί το πλήκτρο reset.

Έτσι, η βασική λειτουργία του Arduino είναι ότι τρέχει η συνάρτηση setup() μία φορά στην αρχή και ακολούθως η loop() ξανά και ξανά μέχρι να μην τροφοδοτείται με ρεύμα ή να πατηθεί το πλήκτρο reset. Στην περίπτωση του Reset ξανατρέχει η συνάρτηση setup() μία φορά και ακολούθως η loop(), όπως δηλαδή ακριβώς και όταν

αρχικά ενεργοποιείται με ρεύμα ο μικροελεγκτής. Στην περίπτωση που έχουν γίνει αλλαγές στο πρόγραμμά, εφόσον αρχικά φορτωθεί στον μικροελεγκτή, αρκεί το πλήκτρο Reset ώστε να εκτελεσθεί το πρόγραμμά από την αρχή.

Ένα τυπικό πρόγραμμα έχει την παρακάτω δομή:

```
void setup() {  
  /* οι εντολές που θα τρέξουν μια φορά κατά την ενεργοποίηση ή μετά από Reset */  
}  
void loop()  
{  
  /* οι εντολές που τρέχουν ξανά και ξανά, μέχρι να απενεργοποιηθεί ή να πατηθεί το  
  Reset */  
}
```

2.9.1. Δηλώσεις μεταβλητών

Όπως σε όλες τις γλώσσες προγραμματισμού, υπάρχουν διάφοροι τύποι μεταβλητών.

Μερικοί τύποι μεταβλητών που υποστηρίζονται στο Arduino είναι:

- boolean, λογικός τύπος με τιμές 0 και 1 (ή True– False)
- byte, ακέραιος με τιμές από 0 έως και 255
- int, ακέραιος με δυνατές τιμές από -32768 έως και 32767
- long, ακέραιος με δυνατές τιμές από -2147483648 έως και 2147483647
- float, δεκαδικοί αριθμοί
- char, χαρακτήρας (μέγεθος ένα Byte)
- string, ακολουθία χαρακτήρων

2.9.2 Συναρτήσεις διαχείρισης θυρών εισόδου – εξόδου (Pins)

Κύρια λειτουργία του μικροελεγκτή είναι να ελέγχει τις θύρες που διαθέτει. Αυτές τις διαχειρίζεται είτε ως έξοδο ή ως είσοδο και είτε στέλνει ή λαμβάνει ηλεκτρικό ρεύμα σε αυτές. Κατά την αρχικοποίηση του προγράμματος στη συνάρτηση setup, θα χρειαστεί να χαρακτηριστούν τα Pins ως είσοδοι ή ως έξοδοι.

Για τη λειτουργία αυτή χρησιμοποιείται η συνάρτηση:

```
pinMode(Pin, Mode)
```

η οποία καλείται με το όνομά της και ορίσματα α) τον αριθμό Pin και β) την κατάσταση λειτουργίας που χαρακτηρίζεται με τη λέξη INPUT (είσοδος) ή OUTPUT(έξοδος).

Όπως έχουμε αναφέρει υπάρχουν 14 ψηφιακά Pins, 6 εκ των οποίων είναι PWM, με ονόματα 0..13. Για παράδειγμα η κλήση της συνάρτησης:

```
pinMode(11, OUTPUT);
```

Ορίζει το pin 11 ως έξοδο. Ομοίως μπορεί να χρησιμοποιηθεί μια μεταβλητή `pin` αντί συγκεκριμένου αριθμού για το Pin:

```
pinMode(ledPin, OUTPUT);
```

Και τα 14 pins του Arduino μπορούν να λειτουργήσουν και ως ψηφιακές έξοδοι, δηλαδή δίνουν έξοδο 0 ή 5V. Αυτό γίνεται με χρήση της συνάρτησης

```
digitalWrite(Pin, Value)
```

όπου το όρισμα Pin αναφέρεται στο νούμερο της θύρας για την οποία θα δοθεί τάση εξόδου, ενώ η τάση εξόδου μπορεί να είναι μία από τις τιμές 0V ή 5V. Οι τιμές αυτές αναπαρίστανται με προκαθορισμένες σταθερές στην παράμετρο `value`:

LOW : θα δώσει 0 V στην έξοδο (pin)

HIGH : θα δώσει 5 V στην έξοδο (pin)

Για παράδειγμα η εντολή:

```
digitalWrite(ledPin, HIGH);
```

Θα στείλει ένα ηλεκτρικό σήμα τάσεως 5 volt στη θύρα που έχει οριστεί ως `ledPin`. Η αντίστοιχη θύρα θα πρέπει να έχει οριστεί ως θύρα εξόδου στη διαδικασία `setup()`, με χρήση της συνάρτησης `pinMode`. Για παράδειγμα:

```
pinMode(ledPin, OUTPUT)
```

Επίσης ανάμεσα στις εντολές μπορεί να οριστεί μια καθυστέρηση. Αυτό επιτυγχάνεται με χρήση της συνάρτησης:

```
delay(time)
```

όπου στη θέση `time` καθορίζεται ο χρόνος σε ms (millisecond = 1/1000 sec).

Η εντολή `delay(time)` σημαίνει ότι σταματά στο σημείο αυτό η εκτέλεση του προγράμματός για χρόνο `time`. Για παράδειγμα:

```
delay(1000); //σταματά την εκτέλεση του προγράμματος για 1000 ms = 1 sec
```

(Εγχειρίδιο Arduino Starter Kit)

Στη συνέχεια παρατίθεται ένας πίνακας με τις πιο σημαντικές εντολές και τις επεξηγήσεις τους:

Όρισμα	Είδος	Τύπος	Παράμετροι	Περιγραφή
LOW	Σταθερά	int	—	Σταθερά με την τιμή 0.
HIGH	Σταθερά	int	—	Σταθερά με την τιμή 1
INPUT	Σταθερά	int	—	Σταθερά με την τιμή 0
OUTPUT	Σταθερά	int	—	Σταθερά με την τιμή 1
pinMode	Εντολή	—	(pin, mode)	Συνάρτηση καθορισμού αν το συγκεκριμένο ψηφιακό pin θα είναι pin εισόδου ή pin εξόδου ανάλογα με την τιμή που δίνεται στην παράμετρο mode (INPUT ή OUTPUT αντίστοιχα).
digitalWrite	Εντολή	—	(pin, pinstatus)	Συνάρτηση που θέτει την κατάσταση pinstatus (HIGH ή LOW) στο συγκεκριμένο ψηφιακό pin.
digitalRead	Συνάρτηση	int	(pin)	Συνάρτηση που επιστρέφει την κατάσταση του συγκεκριμένου ψηφιακού pin (0 για LOW και 1 για HIGH) εφόσον αυτό είναι pin εισόδου.
analogReference	Εντολή	—	(type)	Συνάρτηση που δέχεται τις τιμές DEFAULT, INTERNAL ή EXTERNAL στην παράμετρο type για να καθορίσει την τάση αναφοράς (V_{ref}) των αναλογικών εισόδων (5V, 1.1V ή η εξωτερική τάση με την οποία τροφοδοτείται το pin AREF αντίστοιχα)
analogRead	Συνάρτηση	int	(pin)	Συνάρτηση που επιστρέφει έναν ακέραιο από 0 έως 1023, ανάλογα με την τάση που τροφοδοτείται το συγκεκριμένο pin.
analogWrite	Εντολή	—	(pin, value)	Συνάρτηση που θέτει το συγκεκριμένο ψηφιακό pin σε κατάσταση ψευδοαναλογικής εξόδου (PWM). Η παράμετρος value καθορίζει το πλάτος του παλμού σε σχέση με την περίοδο του παραγόμενου σήματος στην κλίμακα από 0 ως 255 (π.χ. με value 127, το πλάτος του παλμού είναι ίσο με μισή περίοδο).
millis	Συνάρτηση	unsigned long	()	Μετρητής που επιστρέφει το χρονικό διάστημα σε ms από την στιγμή που άρχισε η εκτέλεση του προγράμματος. Λόγω του τύπου

				μεταβλητής (32bit) έχει όριο την τιμή 2^{32}ms
delay	Εντολή	—	(time)	Συνάρτηση που διακόπτει προσωρινά την ροή του προγράμματος για χρόνο time σε ms. Η παράμετρος time είναι unsigned long (από 0 ως 2^{32}).
Serial.begin	Μέθοδος κλάσης	—	(datarate)	Συνάρτηση που θέτει τον ρυθμό μεταφοράς δεδομένων του σειριακού interface (σε baud)
Serial.println	Μέθοδος κλάσης	—	(data)	Συνάρτηση που διοχετεύει τα δεδομένα data για αποστολή στη σειριακή θύρα.

(deltahacker.gr, 2009)

3 Σχεδιασμός του συστήματος

3.9 Εισαγωγή

Το πρώτο στάδιο κατασκευής του αυτοματισμού ήταν η προμήθεια των απαραίτητων υλικών. Το πακέτο Arduino Starter kit παρείχε την πλακέτα του Arduino Uno μαζί με τα απαραίτητα καλώδια για τη σύνδεση με τις υπόλοιπες συσκευές.

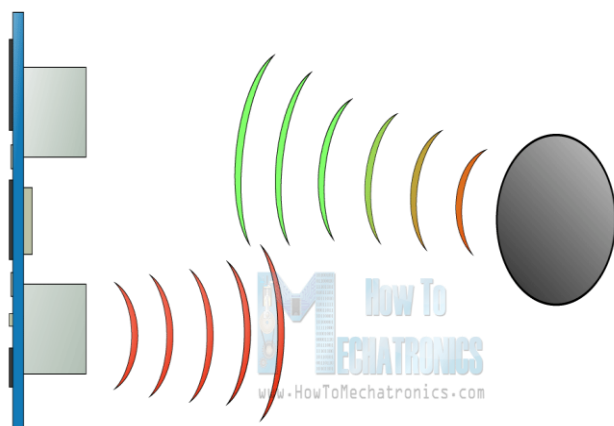


Εικόνα 5. Arduino

Για την κατασκευή επίσης χρησιμοποιήθηκαν ένα breadboard για τη σύνδεση του Arduino με τον αισθητήρα και ο αισθητήρας απόστασης HC-SR04 Ultrasonic. Η επιλογή του συγκεκριμένου αισθητήρα έγινε με γνώμονα τη μεγάλη υποστήριξη από την κοινότητα χρηστών του Arduino, την σχετική αξιοπιστία του αλλά και το πολύ λογικό κόστος αγοράς του.

3.10 Αισθητήρας απόστασης

Ο αισθητήρας αποστέλει ένα υπερηχητικό σήμα σε συχνότητα 40KHz το οποίο μεταδίδεται στον αέρα μέχρι να συναντήσει κάποιο εμπόδιο το οποίο και ανακλά το σήμα πίσω στον αισθητήρα. Υπολογίζοντας τον χρόνο μετάδοσης και λήψης του σήματος αλλά και την ταχύτητα του ήχου, ο επεξεργαστής μπορεί να υπολογίσει την απόσταση.



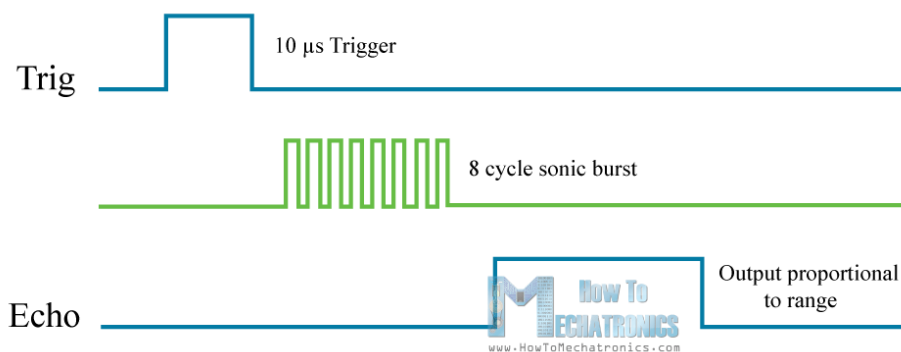
Εικόνα 6. Λειτουργία αισθητήρα

Ο αισθητήρας που χρησιμοποιείται στην εργασία είναι ο HC-SR04 Ultrasonic ο οποίος έχει 4 ακροδέκτες (4 pins) που αντιστοιχούν στη γείωση (Ground), στην παροχή ρεύματος 5 volt (VCC), στον πομπό (Trig) και στον δέκτη του σήματος (Echo). Οι ακροδέκτες Ground και VCC συνδέονται με τους αντίστοιχους του Arduino ενώ οι ακροδέκτες trig και echo μπορούν να συνδεθούν με οποιοδήποτε ακροδέκτη I/O στο Arduino.



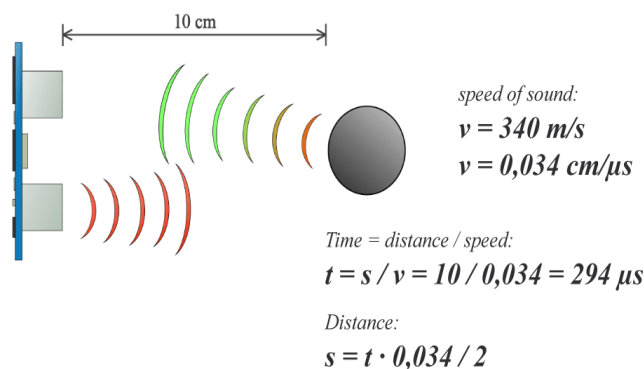
Εικόνα 7. HC-SR04 Ultrasonic

Για την παραγωγή του υπερηχητικού σήματος θα πρέπει να τεθεί ο ακροδέκτης Trig ως έξοδος (output) σε κατάσταση High για 10 μ s. Έτσι από τον πομπό θα σταλούν 8 κύκλοι υπερηχητικού κύματος τα οποία θα ταξιδέψει με την ταχύτητα του ήχου μέχρι να ληφθεί σε περίπτωση ανάκλασης από τον δέκτη (Echo). Ο δέκτης θα στείλει τη μέτρηση στον ακροδέκτη Echo ο οποίος και θα λάβει σε microsecond το χρόνο διάδοσης του κύματος.



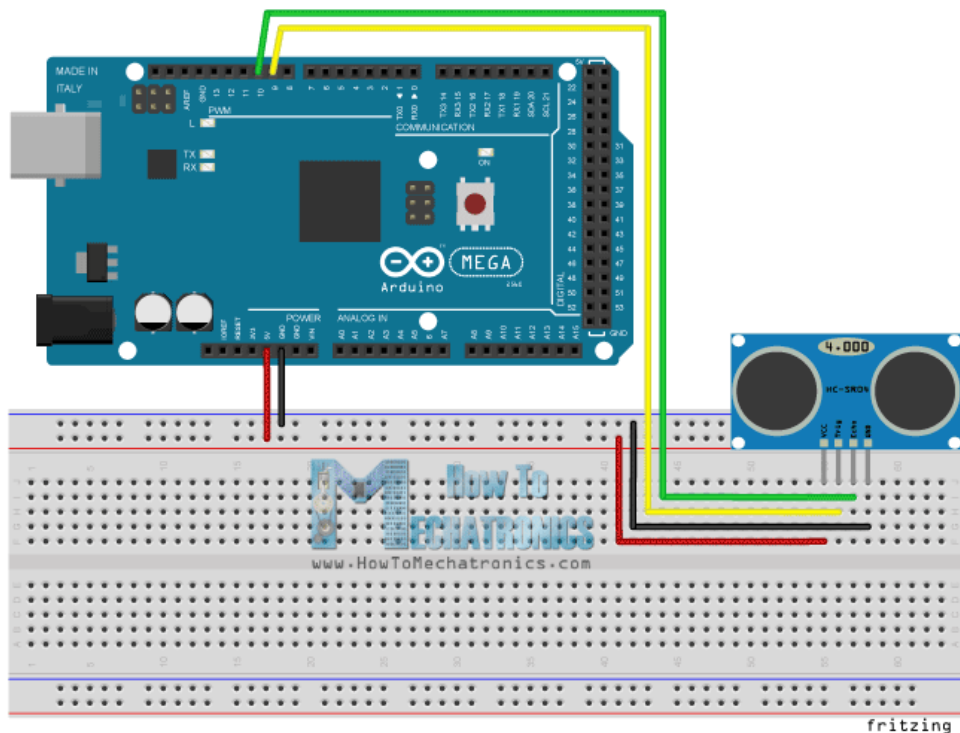
Εικόνα 8. Έξοδος Trig / Είσοδος Echo

έτσι αν για παράδειγμα υπάρχει ένα αντικείμενο σε απόσταση 10 cm από τον αισθητήρα, δεδομένου ότι η ταχύτητα του ήχου είναι 340 m/s ή 0.034 cm/ μ s, το ηχητικό κύμα θα χρειαστεί περίπου 294 μ s. Ο χρόνος όμως θα είναι ο διπλάσιος διότι το κύμα χρειάστηκε να ταξιδέψει μέχρι το αντικείμενο και να επιστρέψει πίσω. Συνεπώς θα πρέπει η απόσταση που θα προκύψει να διαιρεθεί δια 2. τελικά η απόσταση σε cm προκύπτει από τον πολλαπλασιασμό του χρόνου διάδοσης με την ταχύτητα του ήχου σε cm/microsecond και κατόπιν τη διαίρεση με το 2.



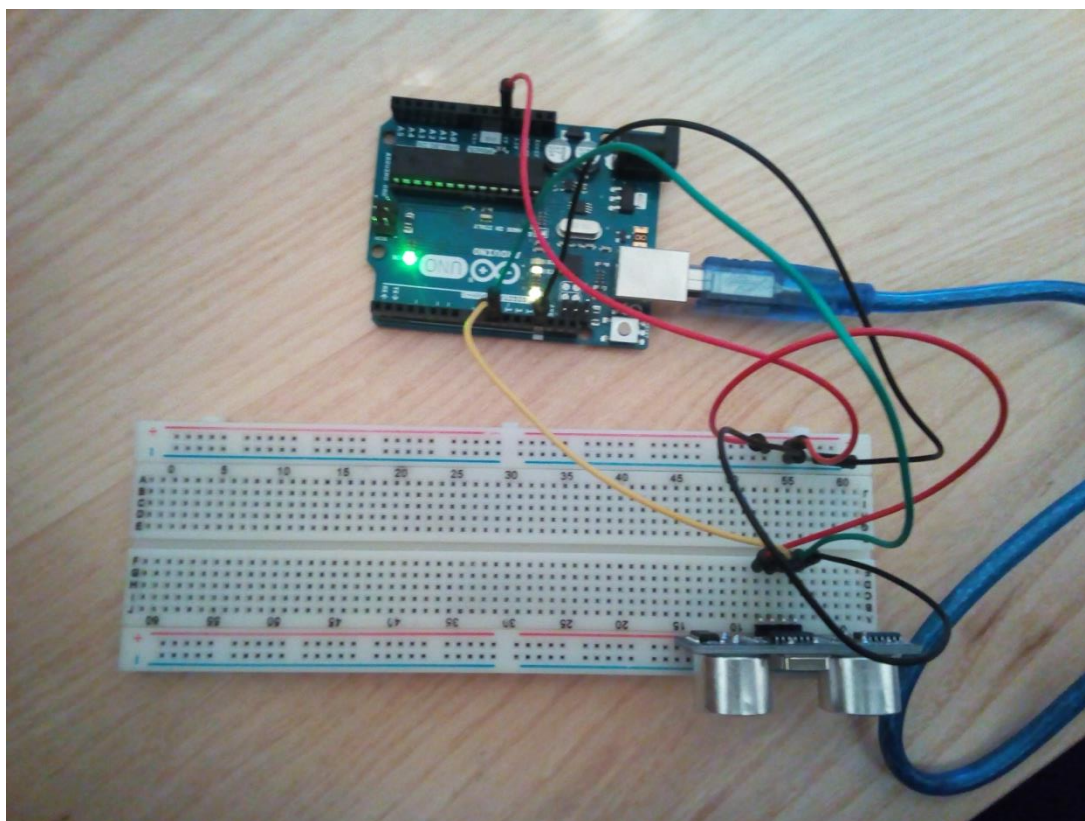
Εικόνα 9. Υπολογισμός απόστασης

3.11 Συνδεσμολογία



Εικόνα 10. Συνδεσμολογία

Η συνδεσμολογία ακολουθεί το Σχήμα. Στο κύκλωμα έχει συνδεθεί ο ακροδέκτης του πομπού (trig) με τον ακροδέκτη 9 του Arduino ενώ ο ακροδέκτης του δέκτη (Echo) με τον ακροδέκτη 10 του Arduino. Στον κώδικα αρχικά ορίζεται οι ακροδέκτες Trig και Echo. Στη συγκεκριμένη περίπτωση οι μεταβλητές που θα χρησιμοποιηθούν ονομάζονται trigPin και echoPin αντίστοιχα. Επίσης χρησιμοποιείται μια μεταβλητή τύπου Long με όνομα “duration” για την αποθήκευση του διαστήματος μετάδοσης του κύματος και μια μεταβλητή τύπου Integer και όνομα distance για την αποθήκευση της απόστασης (create.arduino.cc/projecthub).



Εικόνα 11. Κατασκευή

Στη συνάρτηση `setup` ορίζεται η μεταβλητή `trigPin` ως `output` ενώ η μεταβλητή `echoPin` ως `Input`.

Στη συνάρτηση `loop` αρχικά θα πρέπει να διασφαλιστεί ότι η μεταβλητή `trigPin` και ο ακροδέκτης δεν έχουν προηγούμενη τιμή για αυτό και τίθεται σε κατάσταση `LOW` για 2 μs . Για την παραγωγή του υπέρηχου το `trigPin` τίθεται σε κατάσταση `HIGH` για 10 μs . Χρησιμοποιώντας την συνάρτηση `pulseIn()` του αισθητήρα, γίνεται λήψη του χρόνου που έχει διαβάσει ο δέκτης του αισθητήρα την οποία και αναθέτει στη μεταβλητή `"duration"`. Η συνάρτηση έχει δύο παραμέτρους εκ των οποίων η πρώτη είναι το όνομα του `echo pin` και η δεύτερη η κατάσταση `HIGH` ή `LOW`. Στη συγκεκριμένη περίπτωση, η τιμή τίθεται σε `HIGH` έτσι ώστε η συνάρτηση `pulseIn()` να περιμένει από τον ακροδέκτη να γίνει `HIGH` από την αποστολή του κύματος και να

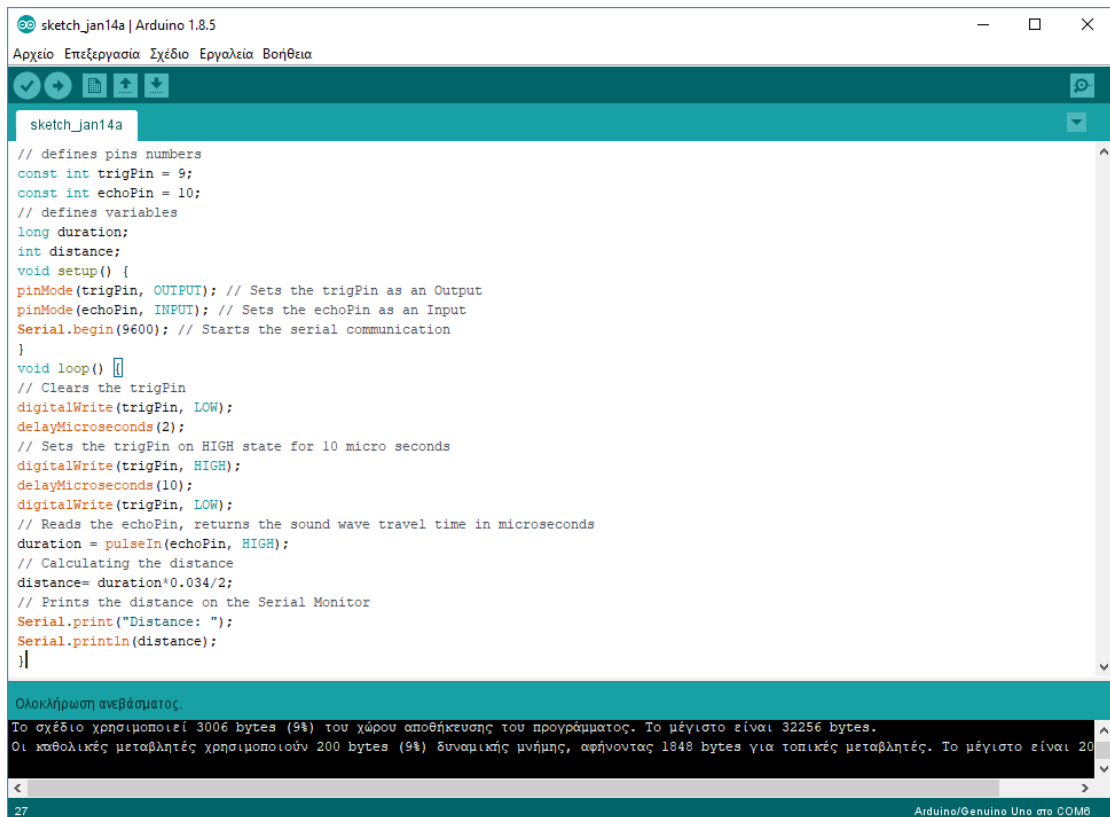
ξεκινήσει την μέτρηση του χρόνου μέχρι ο ακροδέκτης να περάσει σε κατάσταση LOW όταν το κύμα θα ληφθεί από τον δέκτη και θα σταματήσει η καταμέτρηση. Η συνάρτηση θα επιστρέψει τη διάρκεια σε microseconds.

3.12 Συγγραφή προγράμματος

Με βάση τα παραπάνω ο κώδικας που γράφτηκε στο Arduino IDE χρησιμοποιεί την είσοδο που λαμβάνει από το pin 10 και το αποστέλλει στην σειριακή για προεπισκόπηση:

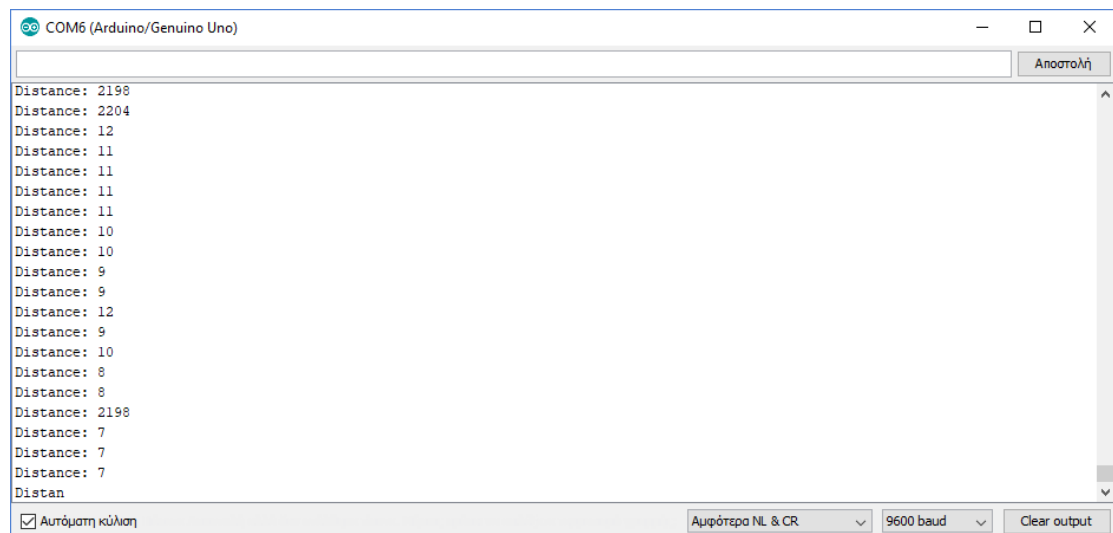
```
// defines constants as pin numbers
const int trigPin = 9;
const int echoPin = 10;
// defines variables for duration and distance
long duration;
int distance;
void setup() {
  pinMode(trigPin, OUTPUT); // Sets the pin 9 (trigPin) as an Output
  pinMode(echoPin, INPUT); // Sets the pin 10 (echoPin) as an Input
  Serial.begin(9600); // Starts the serial communication
}

void loop() {
  // Sends low voltage to clear the trigPin
  digitalWrite(trigPin, LOW);
  delayMicroseconds(2);
  // Sets the trigPin on HIGH state for 10 micro seconds
  digitalWrite(trigPin, HIGH);
  delayMicroseconds(10);
  // Sets the trigPin on LOW state
  digitalWrite(trigPin, LOW);
  // Reads the sound wave travel time in microseconds from the echoPin
  duration = pulseIn(echoPin, HIGH);
  // Calculates the distance as described in theory
  distance= duration*0.034/2;
  // Prints the distance on the Serial Monitor
  Serial.print("Distance: ");
  Serial.println(distance);
}
```



Εικόνα 12. Arduino IDE

Μέσω της επιλογής Εργαλεία -> Παρακολούθηση σειριακής ο προγραμματιστής παρακολουθεί τις τιμές που αποστέλλονται στη σειριακή θύρα επικοινωνίας.



Εικόνα 13. Έλεγχος με την παρακολούθηση της σειριακής

<https://randomnerdtutorials.com/complete-guide-for-ultrasonic-sensor-hc-sr04/>

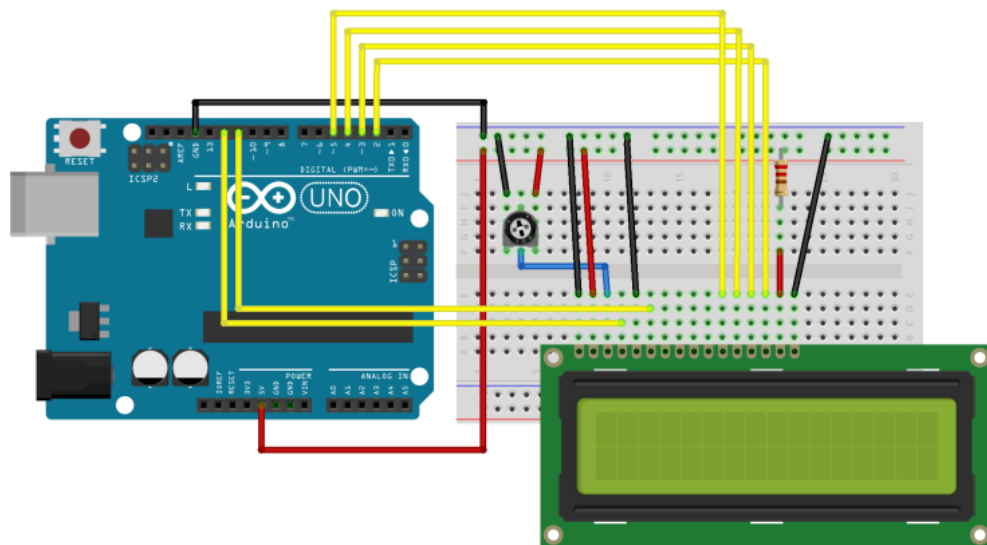
3.13 Οθόνη LCD

Για την εμφάνιση της απόστασης χρησιμοποιείται μια standard LCD οθόνη του Arduino. Επιπρόσθετα χρησιμοποιείται ένα ποτενσιόμετρο 10 Ω και μια αντίσταση 220 Ω.

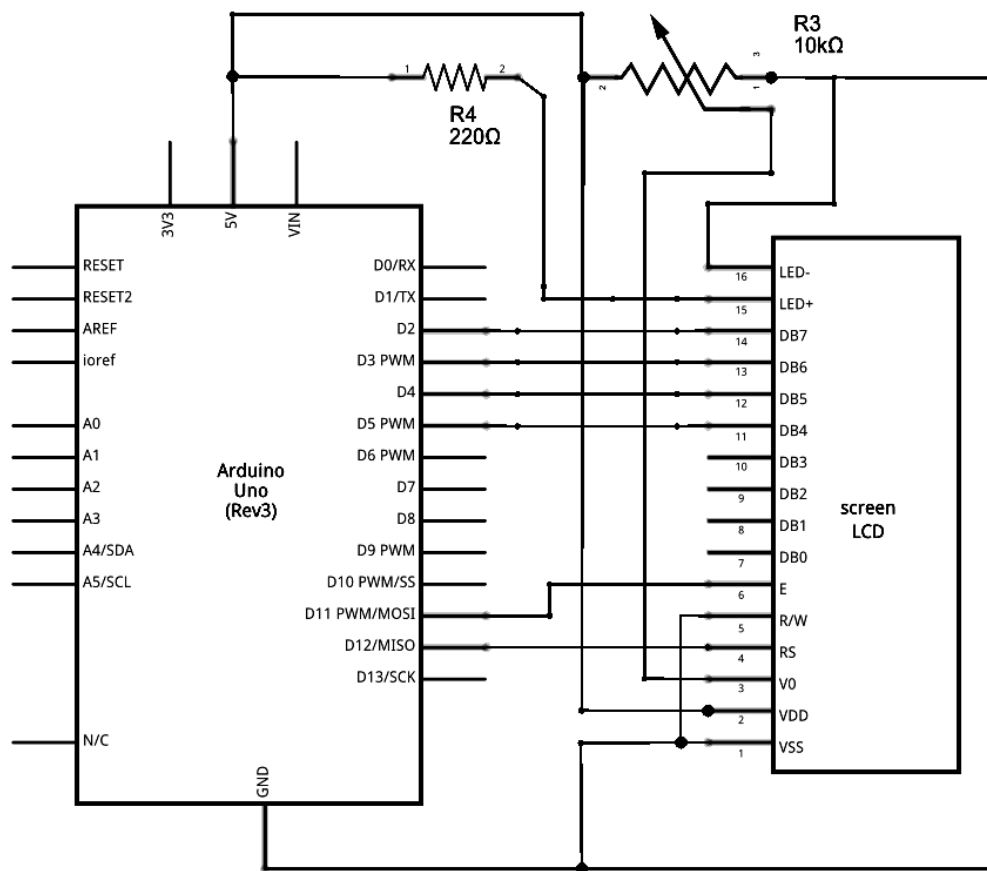
Όπως φαίνεται και στο σχήμα η συνδεσμολογία που χρησιμοποιείται είναι η εξής:

- Ο ακροδέκτης RS του LCD στον ψηφιακό ακροδέκτη 12 του Arduino
- Ο ακροδέκτης Enable του LCD στον ψηφιακό ακροδέκτη 11 του Arduino
- Ο ακροδέκτης D4 του LCD στον ψηφιακό ακροδέκτη 5 του Arduino
- Ο ακροδέκτης D5 του LCD στον ψηφιακό ακροδέκτη 4 του Arduino
- Ο ακροδέκτης D6 του LCD στον ψηφιακό ακροδέκτη 3 του Arduino
- Ο ακροδέκτης D7 του LCD στον ψηφιακό ακροδέκτη 2 του Arduino

Επιπρόσθετα συνδέεται το ποτενσιόμετρο των 10k στην παροχή των +5V και τη γείωση (GND) και την έξοδο στον ακροδέκτη VO του LCD. Μια αντίσταση των 220 Ω χρησιμοποιείται για την τροφοδοσία του φόντου του LCD , στον ακροδέκτη 15 και 16 του LCD .



Εικόνα 14.Συνδεσμολογία Οθόνης



Εικόνα 15. Ηλεκτρονικό σχέδιο

Ακολουθεί ο κώδικας που συνδυάζει τις δύο συσκευές:

/*

The circuit:

- * LCD RS pin to digital pin 12
- * LCD Enable pin to digital pin 11
- * LCD D4 pin to digital pin 5
- * LCD D5 pin to digital pin 4
- * LCD D6 pin to digital pin 3
- * LCD D7 pin to digital pin 2
- * LCD R/W pin to ground
- * LCD VSS pin to ground

```

* LCD VCC pin to 5V
* 10K resistor:
* ends to +5V and ground
* wiper to LCD VO pin (pin 3)

*/

// include the library code:
#include <LiquidCrystal.h>

// initialize the library by associating any needed LCD interface pin
// with the arduino pin number it is connected to
const int rs = 12, en = 11, d4 = 5, d5 = 4, d6 = 3, d7 = 2;
LiquidCrystal lcd(rs, en, d4, d5, d6, d7);

// defines pins numbers for distance sensor
const int trigPin = 9;
const int echoPin = 10;
// defines variables
long duration;
int distance;

void setup() {
// lcd begins
lcd.begin(16,2);
pinMode(trigPin, OUTPUT); // Sets the trigPin as an Output
pinMode(echoPin, INPUT); // Sets the echoPin as an Input

}

void loop() {
// Clears the trigPin
digitalWrite(trigPin, LOW);
delayMicroseconds(2);
// Sets the trigPin on HIGH state for 10 micro seconds
digitalWrite(trigPin, HIGH);
delayMicroseconds(10);
digitalWrite(trigPin, LOW);
// Reads the sound wave travel time in microseconds from the echoPin
duration = pulseIn(echoPin,HIGH,50000);
// Calculates the distance

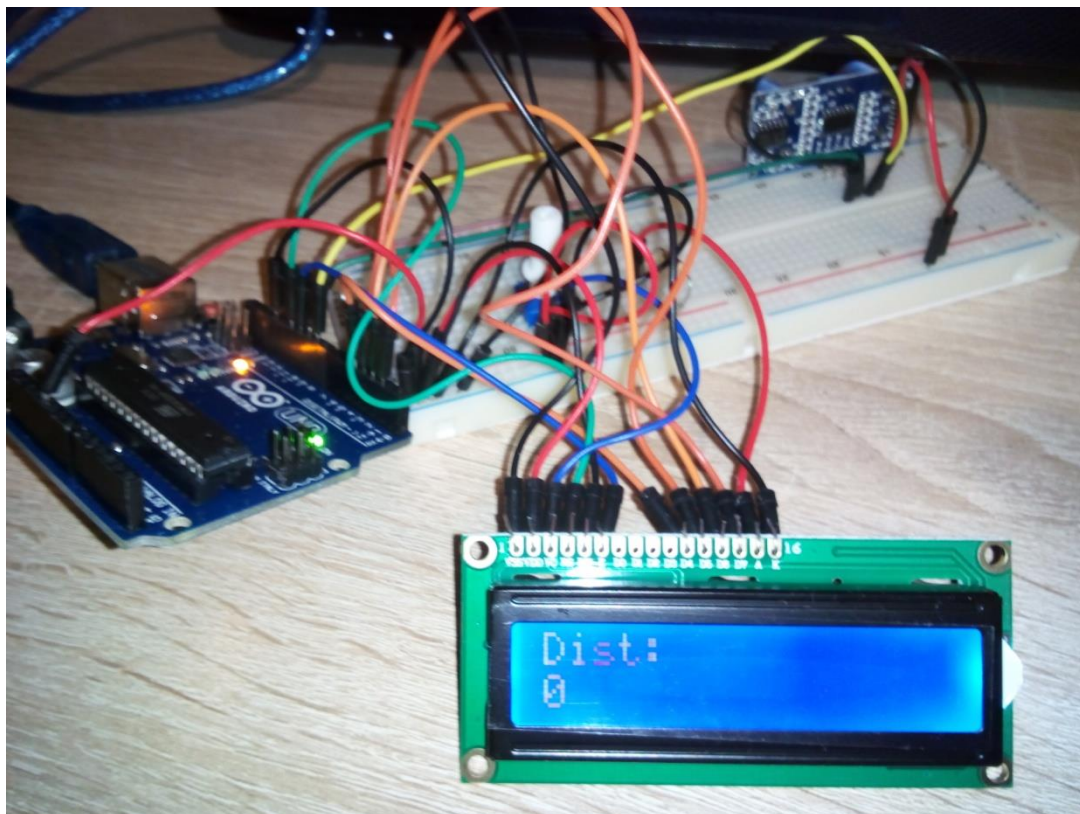
```

```
distance= duration*0.034/2;
```

```
delay(1000);  
lcd.setCursor(0,0);  
lcd.print("Distance:");  
lcd.setCursor(0,1);  
lcd.print(distance);
```

```
delay(1000);  
lcd.clear();  
}
```

Η κατασκευή στην παρούσα φάση φαίνεται στην Εικόνα.



Εικόνα 16.Σύνδεση Αισθητήρα και Οθόνης

Στις παραπάνω υλοποιήσεις, τα αποτελέσματα διοχετεύονταν ενσύρματα μέσω της σειριακής σύνδεσης USB στην θύρα (COM) επικοινωνίας στην οποία είναι συνδεδεμένο το Arduino (www.arduino.cc/en/Tutorial/HelloWorld).

3.14 Ασύρματη μετάδοση

Κατά τη διάρκεια της κατασκευής του μηχανισμού έγινε προσπάθεια να συνδεθεί ο μηχανισμός ασύρματα με έναν υπολογιστή με τη χρήση ενός ασύρματου τοπικού δικτύου Wifi. Για το λόγο αυτό χρησιμοποιήθηκε το κύκλωμα ESP8266.

3.14.1 Κύκλωμα ESP8266

Το ESP8266 είναι ένα από τα πιο δημοφιλή εξαρτήματα για τη διασύνδεση του Arduino σε ασύρματο δίκτυο.



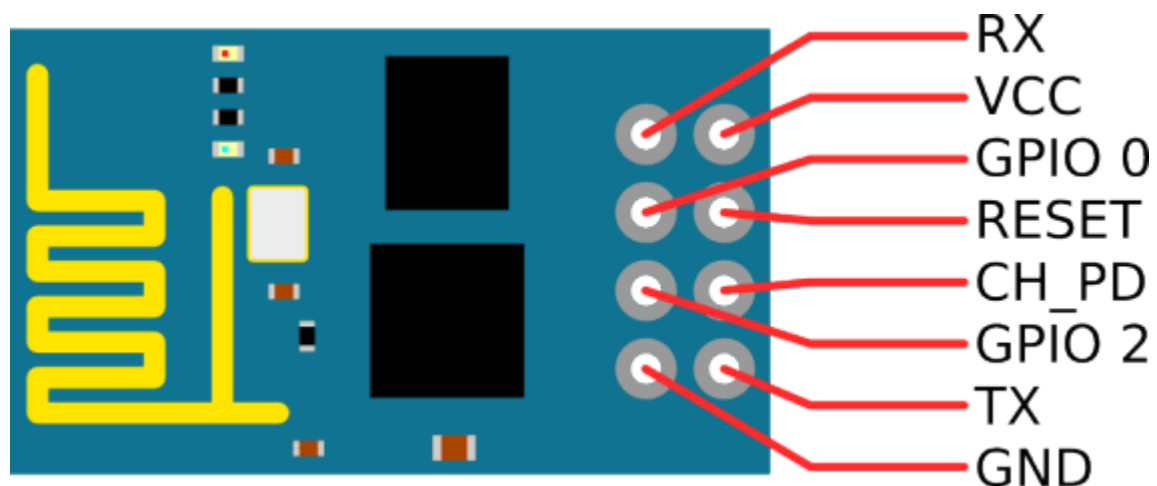
Εικόνα 17. ESP8266 (ESP-01)

Το ESP8266 είναι ένα Wi-Fi module με επεξεργαστή ARM και αποτελεί συγκριτικά με τα άλλα modules μια καλή λύση εξαιτίας του χαμηλού κόστους αλλά και της πληθώρας των εφαρμογών που διατίθενται στο Διαδίκτυο.

Οι δυνατότητες που μπορεί να έχει ένας προγραμματιστής χρησιμοποιώντας το συγκεκριμένο module είναι:

- Δημιουργία ενός web server
- Αποστολή αιτημάτων HTTP (requests)
- Έλεγχος εξόδων
- Διάβασμα εισόδων και interrupts
- Αποστολή email

Στη συγκεκριμένη εργασία χρησιμοποιείται το ESP8266 (ESP-01) V090, το οποίο και διαθέτει δύο LED δίπλα στην κεραία του. Στο παρακάτω σχήμα αποτυπώνονται οι ακροδέκτες του κυκλώματος.



Εικόνα 18. Ακροδέκτες ESP8266 (ESP-01)

3.14.2 Arduino web server

Ο παρακάτω κώδικας χρησιμοποιεί το module ESP8266 και την βιβλιοθήκη ESP8266WebServer.h για να μετατρέψει το Arduino σε web server. Αρχικά ορίζει

ένα αντικείμενο ESP8266WebServer στην http θύρα 80 και συνδέεται στο τοπικό ασύρματο δίκτυο με όνομα SSID και κωδικό PASSWORD. (Στην περίπτωση σύνδεσης σε οποιοδήποτε WIFI τοπικό δίκτυο οι συγκεκριμένες τιμές θα πρέπει να παραμετροποιηθούν). Στη συνέχεια μετά την αναμονή της σύνδεσης, αν η σύνδεση έγινε επιτυχώς, στην κονσόλα εμφανίζεται η IP διεύθυνση που αποδόθηκε στον web server (ESP8266 Web Server Step-By-Step Using Arduino IDE).

Οι υπόλοιπες συσκευές του δικτύου μπορούν να συνδεθούν με τον web server στη συγκεκριμένη IP στη θύρα 80 (http). Ο server με την μέθοδο server.on χειρίζεται τα requests από τις client εφαρμογές των υπόλοιπων συσκευών και απαντάει ανάλογα με το request. Στην περίπτωση του request "/Distance", επιστρέφει την τιμή από τη συνάρτηση getDistance():

```
#include <ESP8266WiFi.h>
```

```
#include <WiFiClient.h>
```

```
#include <ESP8266WebServer.h>
```

```
// defines variables
```

```
//variable of duration of echo
```

```
long duration;
```

```
//variable for calculated distance
```

```
int distance;
```

```
//ssid of the wifi network
```

```
const char* ssid = "SSID";
```

```
//password of the wifi network
```

```

const char* password = "password";

ESP8266WebServer server(80); // uses ESP8266WebServer class to create HTTP server on
port 80

// setup function

void setup() {

// defines pins numbers for distance sensor

const int trigPin = 9;

const int echoPin = 10;

Serial.begin(115200);

//disconnects any wifi connection

WiFi.disconnect(); // Disconnect AP

//turns wificonnection to WIFI_STA mode

WiFi.mode(WIFI_STA);

//starts the connection

WiFi.begin(ssid, password); // Connect to WIFI network


// Wait for connection to start and displays dots

while (WiFi.status() != WL_CONNECTED) {

    delay(500);

    Serial.println(".");

}

```

```

//when connected shows ip address

Serial.print("Connected to ");

Serial.println(ssid);

Serial.print("IP address: ");

Serial.println(WiFi.localIP());


//creates responses depending on browser request

//on simple / request , responds "Distance Server"

server.on("/", [](){

    server.send(200, "text/plain", "Distance Server");

});

//on /Distance request , calls SendDistance()

server.on("/Distance", SendDistance);

server.begin(); // Start HTTP server

Serial.println("HTTP server started.");

}


void loop() {

    // Clears the trigPin

    digitalWrite(trigPin, LOW);

    delayMicroseconds(2);

    // Sets the trigPin on HIGH state for 10 micro seconds

    digitalWrite(trigPin, HIGH);

```

```

delayMicroseconds(10);

digitalWrite(trigPin, LOW);

server.handleClient();

}

void SendDistance()

{

//calls getDistance() to retrieve and calculate distance and creates stringOne for respond

String stringOne = String(getDistance(), 3);

server.send(200, "text/plain",stringOne);

}

double getDistance()

{

// defines variables

long duration;

int distance;

// Reads the echoPin, returns the sound wave travel time in microseconds

duration = pulseIn(echoPin,HIGH,50000);

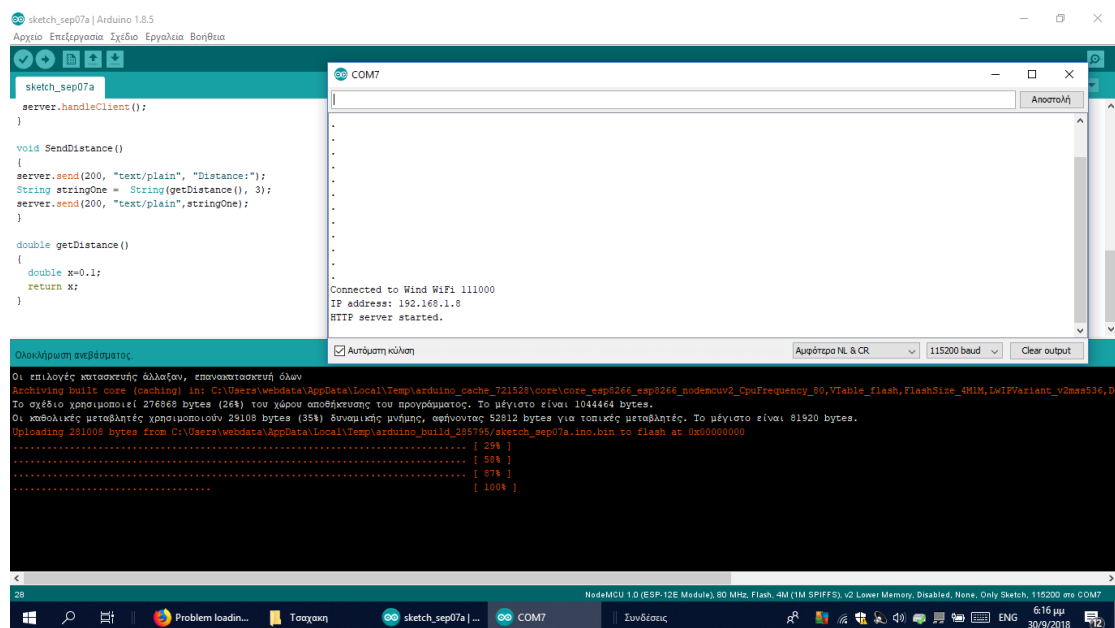
// Calculating the distance from type  $S=v*t \Rightarrow \text{distance}=\text{SOUNDvelocity}*duration$ 

distance= duration*0.034/2;

return distance;

}

```



Εικόνα 19. Σύνδεση με τοπικό ασύρματο δίκτυο

Η μεταφόρτωση του προγράμματος έγινε επιτυχώς και η συσκευή συνδέθηκε επιτυχώς μέσω WiFi με το τοπικό δίκτυο. Χρησιμοποιώντας την κονσόλα παρακολούθησης της σειριακής, έγινε γνωστή η τοπική IP διεύθυνση που πήρε ο μηχανισμός. Στο παράδειγμα της εικόνας η επιστρεφόμενη IP είναι η 192.168.1.8.

Οι υπολογιστές που είναι επίσης συνδεδεμένοι στο τοπικό δίκτυο, μέσω κάποιας client εφαρμογής όπως είναι ένας browser μπορούν να αποστείλουν στον μηχανισμό ένα http αίτημα. Αν το αίτημα είναι “http://192.168.1.8\”, απάντηση που στέλνει ο μηχανισμός είναι η φράση “Distance Server” ενώ αν αποστείλουν το αίτημα “http://192.168.1.8/Distance”, η απάντηση που θα αποσταλεί θα είναι η απόσταση που λαμβάνει από τον αισθητήρα.

Η επιστροφή της τιμής του αισθητήρα δεν έγινε δυνατή στην ασύρματη υλοποίηση και παραμένει αντικείμενο μελλοντικής εργασίας.

4 Συμπεράσματα

4.9 Αποτίμηση Arduino

Το Arduino αποτελεί μια φιλική και εύχρηστη πλατφόρμα ανάπτυξης και κατασκευής αυτοματισμών. Ένας χρήστης με μικρή προγραμματιστική εμπειρία μπορεί άμεσα, ανατρέχοντας σε βασικές οδηγίες και τεκμηρίωση για αρχάριους να δημιουργήσει την πρώτη του κατασκευή. Απαιτείται απλά η εγκατάσταση του Arduino IDE και γνώση των βασικών εντολών προγραμματισμού. Ήδη το πακέτο Starter kit που περιέχει το Arduino περιέχει και εγχειρίδιο χρήσης με πολλά παραδείγματα χρησιμοποιώντας διάφορες συσκευές εισόδου/εξόδου όπως είναι λυχνίες, αισθητήρες κα.

Είναι γεγονός βέβαια πως καθώς οι κατασκευές γίνονται πιο πολύπλοκες, η καλύτερη γνώση ηλεκτρονικών καθίσταται απαραίτητη. Σε κάποιες εφαρμογές απαιτείται η σύνδεση αντιστάσεων, ροοστατών και άλλων εξαρτημάτων που δεν είναι απαραίτητα γνωστά σε κάποιον αρχάριο χρήστη ή κάποιον προγραμματιστή. Παρόλα αυτά ακόμα και σε αυτή την περίπτωση ένας αρχάριος χρήστης, ακολουθώντας κατά γράμμα τις οδηγίες κατασκευής των μηχανισμών μπορεί να δημιουργήσει με επιτυχία έναν πολύπλοκο μηχανισμό. Σε αυτό συμβάλλει η τεράστια βιβλιοθήκη παραδειγμάτων που υπάρχουν είτε στη βιβλιογραφία ή κυρίως στο Διαδίκτυο, από την επίσημη ιστοσελίδα του Arduino αλλά κυρίως από τα χιλιάδες ιστολόγια και wikis εθελοντών οι οποίοι ανεβάζουν τα έργα τους και τα διανέμουν ελεύθερα.

4.10 Πλεονεκτήματα – Μειονεκτήματα

Θα μπορούσαμε να συνοψίσουμε τα πλεονεκτήματα του Arduino ως εξής:

- Πλατφόρμα “ανοικτού λογισμικού”
- Φιλική πλατφόρμα ανάπτυξης προς όλους τους χρήστες
- Εύκολη και φιλική προς αρχάριους χρήστες
- Εύκολη γλώσσα προγραμματισμού

- Συμβατή με διαδεδομένα και φθηνά ηλεκτρονικά εξαρτήματα
- Χιλιάδες βιβλιοθήκες υποστήριξης του υλικού(hardware)
- Χιλιάδες ελεύθερες βιβλιοθήκες με κώδικα
- Μεγάλη υποστήριξη στο Διαδίκτυο

Από την άλλη θα μπορούσαμε να αναφερθούμε σε δύο μειονεκτήματα. Το πρώτο έχει να κάνει με το κόστος αγοράς του πρωτότυπου Arduino. Αν και παραμένει σε λογικά επίπεδα, το κόστος των 80 ευρώ για το Starter kit (<https://store.arduino.cc/genuino-starter-kit>) είναι αισθητά μεγαλύτερο από αυτό που μπορεί να προϋπολογίσει ένας προγραμματιστής αυτοματισμών για μια απλή κατασκευή, ειδικά όταν αναφερόμαστε για πλατφόρμες ανοικτού λογισμικού και ανοικτής τεχνολογίας.

Το δεύτερο μειονέκτημα έχει να κάνει με το βαθμό ευκολίας που κάποιος αρχάριος μπορεί να δημιουργήσει μια κατασκευή. Το γεγονός ότι το Arduino του προσφέρει μια πολύ φιλική κι εύκολη πλατφόρμα ανάπτυξης αυτοματισμών, τον αποτρέπει από το να εμβαθύνει σε βασικές έννοιες ανάπτυξης εφαρμογών με πιο σύνθετα εργαλεία.

4.11 Αποτίμηση εργασίας

Βασικός σκοπός της συγκεκριμένης εργασίας όπως αναφέραμε στην εισαγωγή ήταν η ανάπτυξη μιας εφαρμογής ανίχνευσης της απόστασης ενός εμποδίου από τη συσκευή με τη χρήση της πλατφόρμας Arduino. Ο βασικός αυτός στόχος επιτεύχθηκε αφού μελετήθηκε ο τρόπος λειτουργίας και προγραμματισμού του Arduino. Στην επίτευξη του στόχου αυτού συνέβαλε σημαντικά η πληθώρα αναφορών και πηγών υποστήριξης της πλατφόρμας όσον αφορά στο σχεδιασμό, την ανάπτυξη και τον προγραμματισμό της πλατφόρμας.

Έτσι μια από τις βασικές προτεραιότητες, η εξοικείωση δηλαδή με την πλατφόρμα έγινε σταδιακά αλλά χωρίς ιδιαίτερες δυσκολίες, δεδομένου ότι, όπως αναφέραμε και στα πλεονεκτήματα της πλατφόρμας, το Arduino προσφέρει ένα πολύ φιλικό περιβάλλον ανάπτυξης αυτοματισμών στους αρχάριους χρήστες του. Επίσης αυτό

μελετήθηκαν παραδείγματα για τον τρόπο με τον οποίο συνδέονται διάφορες περιφερειακές συσκευές όπως αισθητήρες, LED και οθόνη LCD με τον μικροελεγκτή του Arduino στέλνοντας ή λαμβάνοντας δεδομένα. Έγινε επίσης προσπάθεια υλοποίησης ενός μηχανισμού για ασύρματη επικοινωνία της συσκευής ανίχνευσης με κάποιο σταθμό εργασίας. Ο στόχος αυτός επιτεύχθηκε όσον αφορά στο σκέλος της επικοινωνίας μέσω ασύρματου δικτύου του Arduino με ένα σταθμό εργασίας, αλλά δεν έγινε εφικτή η αποστολή των δεδομένων από τον αισθητήρα και αποτελεί αντικείμενο μελλοντικής εργασίας η ολοκλήρωσή του.

Κάνοντας μια συνολική αποτίμηση της εργασίας, η ευκολία χρήσης και προγραμματισμού της πλατφόρμας ανάπτυξης Arduino σε συνδυασμό με τη μεγάλη υποστήριξή της από τους χρήστες της, την καθιστούν μια από τις πιο ενδεδειγμένες λύσεις για την εισαγωγή των αρχάριων χρηστών στον τομέα της κατασκευής αυτοματισμών. Σίγουρα η ανάπτυξη πιο εξελιγμένων εφαρμογών απαιτεί μεγαλύτερη εμβάθυνση σε έννοιες ηλεκτρονικής, η οποία μπορεί να γίνει σταδιακά πάλι με την υλοποίηση πιο δύσκολων έργων μέσω του Arduino και στη συνέχεια στη μετάβαση σε κάποια άλλη τεχνολογία μικροελεγκτών.

4.12 Μελλοντική εργασία

Η βασικοί στόχοι της εργασίας που ήταν η διερεύνηση των δυνατοτήτων κατασκευής αυτοματισμών με την χρήση της πλατφόρμας Arduino και η δημιουργία ενός μηχανισμού ανίχνευσης εμποδίων και η εμφάνιση των αντίστοιχων τιμών στον χρήστη ολοκληρώθηκαν επιτυχώς. Επίσης διερευνήθηκε η δυνατότητα σύνδεσης με κάποιον άλλον σταθμό εργασίας μέσω κάποιου ασύρματου δικτύου. Ενώ υλοποιήθηκε επιτυχώς η ασύρματη σύνδεση σε ένα τοπικό δίκτυο, δεν ήταν δυνατός ο συγχρονισμός των συσκευών ανίχνευσης της απόστασης ενός εμποδίου και του μηχανισμού ασύρματης μετάδοσης των αποτελεσμάτων. Ο κώδικας που παρουσιάστηκε, μεταφορτώθηκε επιτυχώς στο Arduino το οποίο και συνδέθηκε επιτυχώς σε ασύρματο δίκτυο. Συνεπώς απαιτείται περεταίρω διερεύνηση και αποτελεί αντικείμενο μελλοντικής εργασίας η ολοκλήρωση μιας εφαρμογής που θα ανιχνεύει την απόσταση και θα αποστέλλει ασύρματα τα δεδομένα σε κάποιο σταθμό εργασίας.

Βιβλιογραφία

1. Raj Kamal, 2009, “Microcontrollers: Architecture, Programming, Interfacing and System Design”, Pearson Education
2. Massimo Banzi, 2011, “Getting Started with Arduino” O'Reilly Media
3. Εγχειρίδιο Arduino Starter Kit
4. “Εισαγωγή στο Arduino” προσπελάσθηκε στις 10/9/2018
URL: <https://deltahacker.gr/arduino-intro/>”
5. “Distance Measurement using Ultrasonic Sensor and Arduino”
προσπελάσθηκε στις 2/5/2018 URL :
“https://create.arduino.cc/projecthub/silicon-ripley-10/distance-measurement-using-ultrasonic-sensor-and-arduino-9bacd4?ref=search&ref_id=distance&offset=2”
6. “Tutorials > Examples from Libraries > LiquidCrystal > HelloWorld “
προσπελάσθηκε στις 10/6/2018
URL:”<https://www.arduino.cc/en/Tutorial/HelloWorld>”
7. “ESP8266 Web Server Step-By-Step Using Arduino IDE” ” προσπελάσθηκε στις 5/5/2018 URL: <https://www.youtube.com/watch?v=m2fEXhl70OY>
8. “Arduino Store” προσπελάσθηκε στις 10/9/2018 URL:
“<https://store.arduino.cc/genuino-starter-kit>”