
**ΑΛΕΞΑΝΔΡΕΙΟ ΤΕΧΝΟΛΟΓΙΚΟ ΕΚΠΑΙΔΕΥΤΙΚΟ ΙΔΡΥΜΑ
ΘΕΣΣΑΛΟΝΙΚΗΣ
ΣΧΟΛΗ ΤΕΧΝΟΛΟΓΙΚΩΝ ΕΦΑΡΜΟΓΩΝ
ΤΜΗΜΑ ΗΛΕΚΤΡΟΝΙΚΩΝ ΜΗΧΑΝΙΚΩΝ ΤΕ**

**Κατασκευή κυκλώματος ελέγχου για έλεγχο Kilobot και
συλλογική συμπεριφορά**

**Πτυχιακή εργασία
του
Ελευθέριου Νικολαΐδη
ΚΑΣ 511076**

**Κωδικός πτυχιακής
16208Μ**

Ημερομηνία ανάληψης: 2/11/2016
Ημερομηνία περάτωσης: 23/05/2017

**Επιβλέπων: Δόκτωρ Τζέκης Παναγιώτης
Συν-επιβλέπων: Καθηγητής James A. R. Marshall και Δόκτωρ AndreaGiovanni Reina**

Θεσσαλονίκη, 2017

**ALEXANDER TECHNOLOGICAL EDUCATIONAL INSTITUTE
OF THESSALONIKI
SCHOOL OF TECHNOLOGICAL APPLICATIONS
DEPARTMENT OF ELECTRONICS ENGINEERING**

Creation of an overhead controller for the Kilobots and
collective behavior

**Thesis of
Eleftherios Nikolaidis
511076**

**Thesis number
16208M**

Start date: 2/11/2016

Submission date: 23/05/2017

Supervisor: Dr. Tzekis Panagiotis

Co-Supervisors: Prof. James A. R. Marshall and Dr. Andrea Giovanni Reina

Thessaloniki, 2017

Πρόλογος

Η συγκεκριμένη πτυχιακή εργασία πραγματοποιήθηκε στο Πανεπιστήμιο του Sheffield της Αγγλίας, υπό την επίβλεψη του αναπληρωτή καθηγητή Τζέκη Παναγιώτη. Θα ήθελα να ευχαριστήσω τον καθηγητή James A. R. Marshall και τον Andreagioanni Reina που με αποδέχτηκαν στο Πανεπιστήμιο του Sheffield, προκειμένου να πραγματοποιήσω το πρόγραμμα πρακτικής άσκησης και να εργαστώ με τα Kilobot, ενώ παράλληλα ήταν δίπλα μου σε ό,τι χρειάστηκα. Επίσης, θα ήθελα να ευχαριστήσω τον κ. Τζέκη για τη υποστήριξη και όλα όσα μου δίδαξε κατά την διάρκεια των σπουδών μου.

Foreword

This particular thesis took place in the University of Sheffield of the United Kingdom, under the supervision of associate professor Tzeki Panagioti. I would like to thank professor James A. R. Marshall and Andreagioanni Reina for accepting me at the University of Sheffield, in order to carry out my internship and to work with the Kilobots, while at the same time they were always willing to help me with anything I needed. In addition, I would like to thank professor Tzeki for the support and everything he taught me during the span of my undergraduate studies.

Περιεχόμενα

Περίληψη	6
Abstract	7
Εισαγωγή	8
Κεφάλαιο 1^ο - Τα ρομπότ στη ζωή μας	9
1.1 Τα πρώτα ρομπότ	9
1.2 Χρήσεις των ρομπότ	9
Κεφάλαιο 2^ο - Συλλογική συμπεριφορά	11
2.1 Ορισμός και παραδείγματα	11
Κεφάλαιο 3^ο - Το Kilobot	14
3.1 Ορισμός και χρήσεις	14
3.2 Τεχνικά χαρακτηριστικά	15
3.3 Επιφάνεια κίνησης και επικοινωνία μεταξύ Kilobot	15
Κεφάλαιο 4^ο - Over-Head Controller (OHC)	17
4.1 Πλακέτα ελέγχου	19
4.1.1 Διαδικασία προγραμματισμού της πλακέτας ελέγχου	20
4.2 Πλακέτα υπέρυθρων LED	25
4.2.1 Τεστ για την επιλογή των υπέρυθρων LED	26
4.3 Γραφικό περιβάλλον KiloGUI	28
4.4 Περιβάλλον προσομοίωσης	30
4.4.1 Εγκατάσταση του V-Rep και αρχικές ρυθμίσεις	30
4.4.2 Χειρισμός του V-Rep	31
4.5 Προγραμματισμός των Kilobot	32
4.5.1 Επικοινωνία OHC με Kilobot	32
4.6 Βελτίωση της υπέρυθρης επικοινωνίας	33
4.6.1 Τεστ – 1: Μέτρηση της φωτεινότητας στο τραπέζι	36
4.6.2 Τεστ – 2: Περιοχή κάλυψης της υπέρυθρης επικοινωνίας	37
4.6.3 Συμπεράσματα	38
4.7 Πρόσθετες πλακέτες υπέρυθρων LED	38
4.8 Αποτελέσματα	39

4.9 Προβλήματα που προέκυψαν	39
4.10 Μελλοντικά σχέδια	40
Παράρτημα 1 - Φωτογραφίες από το ανέβασμα του Bootloader	41
Παράρτημα 2 – Επεξήγηση όρων	45
Βιβλιογραφία	47

Περίληψη

Με τη ραγδαία εξέλιξη της τεχνολογίας τα ρομπότ μπαίνουν όλο και περισσότερο στην καθημερινότητά μας. Μας επιτρέπουν να παράγουμε προϊόντα ασταμάτητα, ακούραστα και με μικρό κόστος, ενώ παράλληλα διευκολύνουν τη ζωή μας με διαφορετικούς τρόπους (π.χ. στις βιομηχανίες, στην εκπαίδευση και στην ιατρική). Μπορούν να εργάζονται μεμονωμένα ή σαν μέλη μιας ομάδας (συλλογικά). Το Kilobot είναι ένα χαμηλού κόστους ρομποτικό σύστημα που χρησιμοποιείται για τη μελέτη των αλληλεπιδράσεων μεταξύ των ατόμων. Εμπνευσμένο από τα έντομα, και κυρίως από τις μέλισσες, τα Kilobot εργάζονται συλλογικά σε μεγάλες ομάδες για ένα κοινό σκοπό. Για τον έλεγχο και τον προγραμματισμό τους χρησιμοποιείται μια πλακέτα ελέγχου που ονομάζεται “Over-Head Controller (OHC)”. Αυτή η πλακέτα καλύπτει επαρκώς μία επιφάνεια περίπου ενός τετραγωνικού μέτρου, από ύψος ενός μέτρου, και μπορεί να ελέγξει συγχρόνως περίπου 100-200 ρομπότ. Συνήθως όμως, χρειάζεται να χειριστούμε περιοχές μεγαλύτερες από 1m^2 και από μεγαλύτερες αποστάσεις με αποτέλεσμα, ο χρήστης να πρέπει να μετακινεί τον OHC σε όλη την επιφάνεια, κάθε φορά που θέλει να στείλει μια καινούργια εντολή στα Kilobot. Αυτή η διαδικασία μπορεί να διαρκέσει αρκετό χρόνο, είναι κουραστική για το χρήστη και τα ρομπότ δεν είναι συγχρονισμένα. Στην παρούσα πτυχιακή εργασία παρουσιάζεται ένα νέο σύστημα ελέγχου των Kilobot που χωρίζεται σε δύο μέρη (πλακέτα ελέγχου και πλακέτες υπέρυθρων LED), μπορεί να καλύψει πολύ μεγαλύτερες επιφάνειες (πάνω από 2m^2) και έλυσε τα προβλήματα που υπήρχαν στην υπέρυθη επικοινωνία μεταξύ των Kilobot και της πλακέτας ελέγχου. Επίσης, περιγράφονται 3 πειράματα [1] που βοήθησαν στην επιλογή των κατάλληλων υπέρυθρων LED για την πλακέτα και στη λύση των προβλημάτων στην υπέρυθη επικοινωνία.

Abstract

With the rapid development of technology, robots are getting more and more into our day-to-day life. They allow us to produce products non-stop, effortlessly and with low cost while at the same time they make our lives easier in many different ways (e.g. in industries, in education, and in medicine). Robots can operate individually or as parts of a team (collectively). The Kilobot is a low-cost robotic system that is used for the study of the interactions between individuals. Inspired by insects, and specifically the honeybees, the Kilobots work collectively in large numbers for a common goal. In order to be controlled and programmed they need a control board that is called “Over-Head Controller (OHC)”. This board can cover sufficiently an area of approximately 1m^2 , from 1m height, and it can control simultaneously 100-200 robots. Usually though, it is needed to cover larger areas than 1m^2 and from larger heights. As a result, every time the user wants to send a command to the Kilobots, he has to move the OHC manually over the area, in order to cover the whole place. This process is time consuming, it is tiring for the user and the robots are not synchronized. In this thesis, we suggest a new control system for the Kilobots that is divided in two parts (control board and infrared LED boards), can cover much larger areas (larger than 2m^2) and solved the problems that existed in the infrared communication between the OHC and the Kilobots. In addition, we will describe 3 experiments [1] that helped us choose the appropriate infrared LEDs for the board and solve the problems in the infrared communication.

Εισαγωγή

Μέχρι πριν από μερικές δεκαετίες τα ρομπότ υπήρχαν μόνο σε ταινίες και στη φαντασία μας. Σήμερα, τα ρομπότ έχουν αρχίσει να παίζουν σημαντικό ρόλο στη ζωή μας και στο κοντινό μέλλον προβλέπεται ότι θα κυκλοφορούν ανάμεσά μας.

Καθημερινά η τεχνολογία αναπτύσσεται δίνοντας μας βελτιωμένες μεθόδους στο να προσεγγίζουμε προβλήματα και να ξεπερνάμε δυσκολίες. Τα ηλεκτρονικά εξαρτήματα γίνονται όλο και μικρότερα, οι μπαταρίες γίνονται μικρότερες σε μέγεθος με μεγαλύτερη χωρητικότητα, η αντοχή των υλικών βελτιώνεται και η ταχύτητα των υπολογισμών στους υπολογιστές αυξάνεται ραγδαία (σχεδόν τριπλασιάστηκε μέσα στα τελευταία 3 χρόνια [2] [3]) φτάνοντας τα 93.000 τρισεκατομμύρια υπολογισμούς το δευτερόλεπτο.

Όλη αυτή εξέλιξη μας επιτρέπει να φτιάχνουμε συσκευές πιο γρήγορες, πιο μικρές, με μεγαλύτερη ακρίβεια, με αντοχή στο χρόνο και ανεξάρτητα από τεχνητές πηγές ενέργειας (π.χ. με ηλιακούς συλλέκτες). Η βελτίωση της αντοχής και η μείωση του βάρους των υλικών μας δίνει τη δυνατότητα να έχουμε ιπτάμενες συσκευές (π.χ. drones) που μπορούν να χρησιμοποιηθούν για μεταφορές προϊόντων, δημοσιογραφία, αναζήτηση και διάσωση ανθρώπων σε συντρίμια ή απομακρυσμένες περιοχές, στη γεωργία για παρακολούθηση των σοδιών κ.α..

Μέρος αυτής της εξέλιξης οφείλεται στην μελέτη της φύσης. Παρατηρώντας πώς τα ζώα ανταπεξέρχονται και προσαρμόζονται στο περιβάλλον τους μπορούμε να κατασκευάσουμε πιο αεροδυναμικά σχήματα, να φτιάξουμε ρομπότ που περπατάνε και τρέχουν, βρίσκονται κάτω από το νερό, σκαρφαλώνουν σε δέντρα ή ανώμαλες επιφάνειες ή ακόμα και να προσομοιώσουμε τη συμπεριφορά ενός πλήθους ανθρώπων.

Τα περισσότερα ζώα για να επιβιώσουν συνηθίζουν να ζουν σε ομάδες και αγέλες, αυτό τους επιτρέπει να αποτρέπουν εχθρούς από το να τους επιτεθούν, να αναζητούν πιο αποτελεσματικά για την τροφή τους, να βρίσκουν στέγη και να μεταναστεύουν.

Με τη μελέτη της συλλογικής συμπεριφοράς, προσπαθούμε να καταλάβουμε πώς παίρνουν αποφάσεις τα ζώα και οι άνθρωποι, αλλά και γιατί παίρνουν αυτές τις αποφάσεις.

Κεφάλαιο 1^ο - Τα ρομπότ στη ζωή μας

1.1 Τα πρώτα ρομπότ

Με τον όρο “ρομπότ” (robot) ονομάζουμε οποιαδήποτε μηχανική συσκευή που μπορεί να εκτελέσει μια περίπλοκη σειρά ενεργειών αυτόνομα ή κάτω από τον έλεγχο ενός ανθρώπου.

Η ονομασία “ρομπότ” προέρχεται από την Τσέχικη λέξη “ρομπότα” (robota) που σημαίνει εξαναγκαστική εργασία. Καθιερώθηκε ως όρος με τη σημερινή του έννοια το 1920 από τον Τσέχο θεατρικό συγγραφέα Καρέλ Τσάπεκ, στο έργο του “R.U.R.” (Rossum’s Universal Robots), όπου σατιρίζει την εξάρτηση της κοινωνίας από τους μηχανικούς εργάτες (ρομπότ) που τελικά εξοντώνουν τους δημιουργούς τους [Wikipedia].

Η σημερινή έννοια των ρομπότ άρχισε να αναπτύσσεται στο ξεκίνημα της βιομηχανικής επανάστασης, με την ανάπτυξη της πολύπλοκης μηχανικής και την ακόλουθη άφιξη του ηλεκτρισμού (μετά το 1831). Χάρης αυτήν, έγινε δυνατή η τροφοδοσία μηχανών με μικρούς συμπαγής κινητήρες. Με την εξέλιξη της τεχνολογίας, ξεκίνησε η κατασκευή μηχανών που χειρίζονταν από τον άνθρωπο για την αποφυγή της επαφής με επικίνδυνα υλικά (π.χ. ραδιενεργά) και για περισσότερη ακρίβεια (π.χ. σε συναρμολογήσεις προϊόντων ή αυτοκινήτων). Μετά το 1970 με την άφιξη του μικροελεγκτή (microcontroller), ξεκίνησε η κατασκευή ρομπότ που μπορούσαν να επαναπρογραμματιστούν ταχύτατα για την παραγωγή νέων ή διαφοροποιημένων προϊόντων. Έτσι, ξεκίνησε και η κατασκευή ανθρωποειδών ρομπότ (android) που είναι σχεδιασμένα για να μοιάζουν και να κινούνται σαν τον άνθρωπο. Τέτοιου είδους ρομπότ είναι εφοδιασμένα με διάφορους αισθητήρες, για να αντλούν πληροφορίες από το περιβάλλον, με ταχύτατους επεξεργαστές πληροφοριών και με κινητήριο σύστημα εκτέλεσης πληροφοριών.

1.2 Χρήσεις των ρομπότ

Καθημερινά τα ρομπότ εντάσσονται όλο και περισσότερο στη ζωή μας με κάθε τρόπο.

Στη βιομηχανία μπορούν να χρησιμοποιηθούν για αποφυγή της άμεσης επαφής του ανθρώπου με επικίνδυνες ή επίπονες εργασίες (π.χ. ηλεκτροσυγκολήσεις [4], συναρμολογήσεις [5], βαφές με ψεκασμό [6], ταξινόμηση προϊόντων σε αποθήκες [7] κ.α.), αλλά κυρίως χρησιμοποιούνται για να εκτελέσουν εργασίες φθηνότερα και πιο γρήγορα από ότι ένας άνθρωπος. Έτσι, είναι δυνατόν μία παραγωγική μονάδα να παράγει μεγαλύτερη ποσότητα κάποιων προϊόντων, πιο γρήγορα και πιο οικονομικά (π.χ. σε μια αλυσίδα παραγωγής).

Στην εκπαίδευση μπορούν να χρησιμοποιηθούν για την εκμάθηση προγραμματισμού, ώστε τα παιδιά να αποκτήσουν ενδιαφέρον για τις νέες τεχνολογίες και να κατανοήσουν παραπάνω τις έννοιες του προγραμματισμού μέσω διαδραστικών ρομπότ [8]. Επίσης, έχουν δείξει ότι η ένταξη των ρομπότ στο σχολείο μπορεί να συνδεθεί με την εφαρμογή “project based learning” προγραμμάτων, όπου τα παιδιά αναπτύσσουν τόσο δεξιότητες επίλυσης προβλημάτων, αλλά και τις κοινωνικές τους δεξιότητες. Ακόμα, η αλληλεπίδραση μαθητών που βρίσκονται στο φάσμα του αυτισμού με τα ρομπότ, μπορεί να συμβάλλει στην ανάπτυξη των κοινωνικών τους δεξιοτήτων [9]. Τέλος, το μάθημα γίνεται πιο ευχάριστο και ενδιαφέρον για τους μαθητές.

Στην ιατρική μπορούν να χρησιμοποιηθούν ως χειρουργικά ρομπότ που δίνουν τη δυνατότητα στους γιατρούς να εκτελούν εγχειρήσεις με μεγαλύτερη ακρίβεια και εξ αποστάσεως, ειδικά όταν δεν υπάρχουν άλλα εκπαιδευμένα φυσικά πρόσωπα για βοήθεια. Μία άλλη σχετικά καινούργια εφαρμογή είναι η χρήση ρομπότ στην αποκατάσταση ασθενών. Γίνεται κυρίως με τη χρήση εξωσκελετικών μηχανισμών που επιτρέπουν στους ασθενείς να αποκτήσουν πιο γρήγορα κινητικότητα σε μέλη του σώματός τους, μετά από κάποιο ατύχημα ή εγκεφαλική δυσλειτουργία.

Εκτός από τα ρομπότ που χειρίζονται από τον άνθρωπο υπάρχουν και αυτά που δρουν αυτόνομα. Αυτά είναι προγραμματισμένα ρομπότ, τα οποία εκτελούν μια σειρά ενεργειών που έχουν αποθηκευμένα στη μνήμη και για τη λειτουργία τους δεν είναι απαραίτητη κάποια ενέργεια από τον άνθρωπο. Μπορούν να λειτουργούν ατομικά (π.χ. σε μία βιομηχανία για να εκτελούν μία σειρά ενεργειών ασταμάτητα και με απόλυτη ακρίβεια) ή συλλογικά (σαν μέρος μιας ομάδας, βλέπε Κεφάλαιο 2 για περισσότερες πληροφορίες).

Στις μέρες μας, ιδιαίτερη προσοχή δίνεται και στην τεχνητή νοημοσύνη (artificial intelligence, AI) η οποία δίνει νοημοσύνη σε μηχανές. Χαρακτηρίζονται αλλιώς ως έξυπνες μηχανές, επειδή αντιλαμβάνονται το περιβάλλον τους και δρουν βάσει των πληροφοριών που πήραν, με αποτέλεσμα να αυξάνουν την πιθανότητα επιτυχίας.

Με τη χρήση της τεχνητής νοημοσύνης και της ρομποτικής έχουν αρχίσει να εμφανίζονται τα πρώτα οικιακά ρομπότ. Είτε για οικιακές δουλειές όπως καθάρισμα [10] και μαγείρεμα [11], είτε για συντροφιά και φροντίδα ηλικιωμένων ή παιδιών [12] [13] [14] [15]. Τα ρομπότ φαίνεται πως ξεκίνησαν να μπαίνουν όλο και περισσότερο στη ζωή μας.

Κεφάλαιο 2^ο - Συλλογική συμπεριφορά



Εικόνα 2.1 Παραδείγματα συλλογικής συμπεριφοράς. **α)** Τα περισσότερα ψάρια ταξιδεύουν σε σχολεία για να προστατευτούν από θηρευτές, να ταξιδεύουν πιο αποτελεσματικά και να βελτιώσουν την αναζήτηση της τροφής τους. [Flickr: Adam Rifkin] **β)** Η είσοδος μιας φωλιάς μυρμηγκιών. Τα μυρμήγκια αφήνουν ίχνη φερομόνης στη διαδρομή από την τροφή μέχρι τη φωλιά τους ώστε τα υπόλοιπα μυρμήγκια να μπορούν να ακολουθήσουν και να βρουν την τροφή [43]. **γ)** Φίλαθλοι αναπαριστούν “κύμα” σε αθλητικό γεγονός [Saurabh Deshpande] **δ)** Ροή κυκλοφορίας στην Ισπανία [44] **ε)** Σμήνος πουλιών μεταναστεύει σε ομάδες [Leah Keshet & Ryan Lukeman] **ζ)** Μία ομάδα ακρίδων [iStockphoto/Ruvan Boshoff].

2.1 Ορισμός και παραδείγματα

Η συλλογική συμπεριφορά είναι ένας όρος που χρησιμοποιούν οι κοινωνιολόγοι, για να αναφερθούν σε μια αυθόρμητη σειρά συμπεριφορών που αποκτά ένα μεγάλο σύνολο ανθρώπων.

Πιο συγκεκριμένα, η συλλογική συμπεριφορά αναφέρεται στη σχετικά αυθόρμητη και μη δομημένη συμπεριφορά ενός μεγάλου συνόλου ατόμων που δρουν μαζί ή επηρεάζονται από άλλα άτομα. Την ονομάζουμε σχετικά αυθόρμητη και μη δομημένη, γιατί η συμπεριφορά μπορεί να είναι αυθόρμητη, αλλά εν μέρει προγραμματισμένη ή μπορεί να είναι απρόβλεπτη και εν μέρει οργανωμένη και προβλέψιμη. Κάποια είδη συλλογικής συμπεριφοράς συμβαίνουν πιο αυθόρμητα και τυχαία από κάποια άλλα, και κάποια είδη είναι πιο πιθανό να περιλαμβάνουν άτομα που δρουν ομαδικά, αντί να επηρεάζουν το ένα το άλλο. Στο γενικό πλαίσιο όμως, η συλλογική συμπεριφορά θεωρείται περισσότερο αυθόρμητη, παρά δομημένη, σε σχέση με τις συμβατικές συμπεριφορές μιας τάξης, ενός χώρου εργασίας ή μια συνηθισμένη καθημερινή εργασία που μας είναι γνώριμη.

Γενικά, οι συλλογικές συμπεριφορές έχουν πολύ λίγα κοινά στοιχεία μεταξύ τους, παρά το γεγονός ότι τα βασικά χαρακτηριστικά που τα αποτελούν τους επιτρέπουν να κατηγοριοποιούνται σε αυτές.

Τα συλλογικά φαινόμενα παρατηρούνται πολύ συχνά στη ζωή μας, χωρίς να τα δίνουμε ιδιαίτερη προσοχή: από τη συσσωμάτωση των κυττάρων για την κατασκευή του σώματός μας, τη δραστηριότητα των νευρώνων στον εγκέφαλό μας, ένα σμήνος πουλιών που παίρνει σχηματισμούς στον ουρανό και από το “παλλόμενο” πλήθος ανθρώπων που μας περιτριγυρίζουν ένα Σάββατο βράδυ.

Η κατανόηση αυτών των φαινομένων είναι ένα σημαντικό μέρος του τομέα της αναπτυξιακής βιολογίας, νευροεπιστήμης, οικολογίας, και κοινωνιολογίας (όπως ήδη εξηγήσαμε), αλλά όχι μόνο αυτών. Για παράδειγμα, η μελέτη της νευροεπιστήμης και του ανοσοποιητικού συστήματος περιλαμβάνει την κατανόηση του πώς οι νευρώνες και τα κύτταρα αλληλοεπιδρούν για να πάρουν αποφάσεις. Μελετώντας τη συμπεριφορά των ζώων μας δίνει ένα παράδειγμα του πώς τα πειράματα μπορούν να συνδυαστούν με μια θεωρία περίπλοκων συστημάτων για να καταλάβουμε καλύτερα τον κόσμο που μας περιβάλλει.

Η μελέτη των συλλογικών φαινομένων στηρίζεται στην ιδέα, ότι μια μεθοδολογία μπορεί να εφαρμοστεί για την κατανόηση συστημάτων σε διάφορες φυσικές κλίμακες. Ενώνοντας διαφορετικά επίπεδα οργάνωσης, κατανοούμε φαινόμενα στα οποία επαναλαμβανόμενες αλληλεπιδράσεις μεταξύ μεμονωμένων ατόμων παράγουν αποτελέσματα σε μεγαλύτερες κλίμακες από τον εαυτό τους.

Η συμπεριφορά των ζώων, μας παρέχει πολλά ενδιαφέροντα και προσβάσιμα παραδείγματα συλλογικών φαινομένων και περίπλοκων συστημάτων. Οι περισσότεροι άνθρωποι είναι οικείοι με τα ίχνη των μυρμηγκιών, τη συγκέντρωση των κατσαρίδων, τα σχολεία ψαριών, τις μεταναστεύσεις των πουλιών, τα σμήνη μελισσών, την κατασκευή των ιστών από τις αράχνες και την μαζική μετακίνηση των ακρίδων· ακόμα και αν δεν τα έχουν παρατηρήσει από κοντά.

Τα ζώα είναι ουσιαστικά πιο περίπλοκα από τις πρωτεΐνες ή τα κύτταρα και μπορεί να είναι δύσκολο να αποδώσουμε μια καθαρή περιγραφή της συμπεριφοράς μεμονωμένων ζώων. Οι ιδιαιτερότητες του κάθε ζώου και η δυσκολία στη συλλογή δεδομένων για τη συμπεριφορά τους σημαίνει ότι σπάνια μπορούμε να γράψουμε συνοπτικά μαθηματικά μοντέλα για τις ενέργειες και αλληλεπιδράσεις των ζώων.

Μεμονωμένα ζώα είναι εγγενώς δύσκολο να προβλεφθούν, αλλά πολλές φορές μπορούμε να κάνουμε ισχυρές προβλέψεις για την συλλογική συμπεριφορά τους. Με μαθηματικά μοντέλα, όταν εφαρμοστούν σωστά, είναι δυνατόν να κάνουμε προβλέψεις για τη συμπεριφορά ακόμα και του πιο περίπλοκου ζώου, συμπεριλαμβανομένου και του ανθρώπου [16].

Με τη βοήθεια των προσομοιώσεων (βλέπε Κεφάλαιο 4.4) και των πειραμάτων με ρομπότ, όπως το Kilobot (βλέπε Κεφάλαιο 3) [17] και το e-ruck [9] μπορούμε να αναπαράγουμε μαθηματικά μοντέλα και να βγάλουμε κάποια συμπεράσματα για τη συμπεριφορά των ζώων, αλλά και των ανθρώπων.

Κεφάλαιο 3^ο - Το Kilobot

3.1 Ορισμός και χρήσεις

Το Kilobot είναι ένα χαμηλού κόστους, εύκολο στη χρήση ρομποτικό σύστημα που χρησιμοποιείται για την προώθηση της ανάπτυξης ενός πληθυσμού ρομπότ που μπορεί να προγραμματιστεί για να αποδώσει χρήσιμες πληροφορίες από την καθοδήγηση των αλληλεπιδράσεων μεταξύ των ατόμων. Αυτοί οι πληθυσμοί, εμπνεύστηκαν από αποικίες εντόμων, όπως οι μέλισσες και τα μυρμήγκια, τα οποία μπορούν αποτελεσματικά να ψάξουν και να βρουν πηγές τροφής σε μεγάλα και περίπλοκα περιβάλλοντα, να μεταφέρουν συλλογικά μεγάλα αντικείμενα και να συντονίσουν το χτίσιμο φωλιών και γεφυρών σε τέτοια περιβάλλοντα.

Ακολουθώντας αυτήν την έμπνευση από τη φύση, ένας πληθυσμός ρομπότ μπορεί μια μέρα να κατασκευάσει σήραγγες σε συντρίμια για να βρει επιζώντες, να σαρώσει ένα περιβάλλον και να αφαιρέσει μολυσμένα αντικείμενα, να βοηθήσει το μειωμένο αριθμό των μελισσών στην επικοινωνία των σοδιών και να σχηματίζουν αυτόνομα βοηθήματα στήριξης σε ετοιμόρροπα κτήρια. Το Kilobot σχεδιάστηκε για να παρέχει στους επιστήμονες μια φυσική πλατφόρμα για ανάπτυξη και κατανόηση της συλλογικής συμπεριφοράς και τη συνειδητοποίηση των δυνατοτήτων για απόδοση λύσεων σε ένα μεγάλο εύρος προκλήσεων.

Η πρώτη εμφάνιση των Kilobot είχε γίνει στο Harvard το 2011 από τον Michael Rubenstein και την ομάδα του [18]. Η ονομασία προέρχεται από την ιδέα του να έχεις “κιλά” ρομπότ να δουλεύουν μαζί, αλλά και επειδή η παρουσίαση τους είχε γίνει με 1024 ρομπότ, που είναι προγραμματιστική μονάδα του “kilo”.

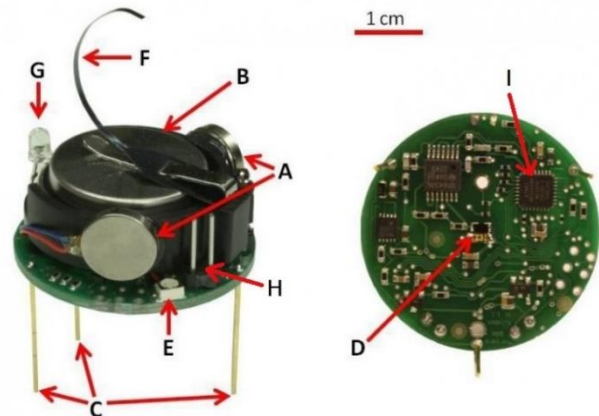


Εικόνα 3.1 Παρουσίαση 1024 Kilobot. [Michael Rubenstein, Harvard University]

3.2 Τεχνικά χαρακτηριστικά

Κάθε Kilobot έχει:

- Ένα μικροελεγκτή (Atmega328p, 8bit @ 8Mhz) για επεξεργαστική μονάδα, ο οποίος έχει 32KB μνήμη Flash που χρησιμοποιείται για τον Bootloader και για τον προγραμματισμό του, 1KB μνήμη EEPROM για τις τιμές βαθμονόμησής του και 2KB μνήμη SDRAM.
- Επαναφορτιζόμενη μπαταρία ιόντων λιθίου 3.7V και ενσωματωμένο κύκλωμα φορτιστή, που φορτίζει την μπαταρία όταν δοθούν +6V σε ένα από τα ποδαράκια του και γείωση στην ακίδα φόρτισης. Κάθε πλήρης φόρτιση έχει αυτονομία 3 μήνες σε αναστολή ή 2,5 ώρες σε κίνηση.
- Επικοινωνία με γειτονικά Kilobot σε ακτίνα 7cm με αντανάκλαση των υπέρυθρων στο έδαφος που πατάνε και καθορισμό της απόστασής τους από την ισχύ του σήματος που λάβανε από το άλλο Kilobot.
- Ανίχνευση της έντασης του περιβάλλοντος φωτισμού.
- 2 κινητήρες δονήσεων, οι οποίοι χειρίζονται ανεξάρτητα και επιτρέπουν στο Kilobot να κινείται στο χώρο. Κάθε κινητήρας έχει 255 διαφορετικές βαθμίδες ισχύος.
- Ένα LED (Light-Emitting Diode) τριών καταστάσεων (RGB) με κατεύθυνση προς τα πάνω και κάθε χρώμα έχει 3 επίπεδα φωτεινότητας.
- Διάμετρο 33mm και ύψος 34mm (μαζί με τα ποδαράκια)
- Μία κεφαλή σειριακής εξόδου, για αντιμετώπιση προβλημάτων στον κώδικα μέσω του τερματικού του υπολογιστή.



Εικόνα 3.2 Ισομετρική προβολή (Αριστερά) και κάτω μέρος (Δεξιά) ενός Kilobot. Κάποια από τα κύρια χαρακτηριστικά είναι: (A) Κινητήρες δονήσεων, (B) Μπαταρία λιθίου, (C) Ακαμπτα ποδαράκια στήριξης, (D) Υπέρυθρος πομπός και δέκτης, (E) LED τριών καταστάσεων (RGB), (F) Ακίδα φόρτισης, (G) Αισθητήρας Περιβάλλοντος φωτισμού, (H) Κεφαλή σειριακής εξόδου και (I) Μικροελεγκτής. Παρατηρήστε τη γραμμή του 1cm για κλίμακα. [Michael Riihenstein Harvard University]

3.3 Επιφάνεια κίνησης και επικοινωνία μεταξύ Kilobot

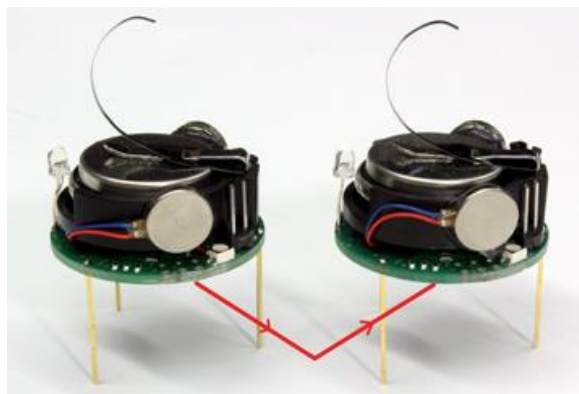
Επιφάνεια και κίνηση

Τα Kilobot τοποθετούνται πάνω σε μία λεία αντανεκλαστική επιφάνεια, γυαλί ή μελαμίνη (ασπροπίνακας), η οποία πρέπει να μην έχει σκόνη ή άλλα εμπόδια, γιατί δημιουργεί προβλήματα στη μετακίνησή τους και στην επικοινωνία τους με τα άλλα Kilobot και την πλακέτα ελέγχου. Τα Kilobot, λόγω του μικρού τους βάρους και των λεπτών ακίδων για ποδαράκια τους επιτρέπει να

κινούνται με δονήσεις. Με τους 2 κινητήρες δόνησης, που διαθέτουν, μπορούμε να ελέγξουμε την κατεύθυνση της κίνησής τους. Για ευθύγραμμη κίνηση, οι 2 κινητήρες έχουν την ίδια τιμή (με ταχύτητα έως 1 εκατοστό/δευτερόλεπτο). Όταν θέλουμε να περιστραφούν δεξιά, ο δεξιός κινητήρας είναι σταματημένος και ο αριστερός στο μέγιστο (με ταχύτητα έως 45 μοίρες/δευτερόλεπτο), ενώ για αριστερή περιστροφή το αντίστροφο. Αυτός ο τρόπος κίνησης όμως, δεν τα επιτρέπει να κινηθούν προς τα πίσω, να ανιχνεύσουν ότι έχουν κολλήσει κάπου και να προσπεράσουν εμποδία, όπως κόκκους σκόνης ή γρατζουνιές στην επιφάνεια της κίνησής τους, αλλά διατηρεί το κόστος τους πολύ μικρό.

Επικοινωνία ρομπότ

Τα Kilobot επικοινωνούν μεταξύ τους με υπέρυθρα σήματα. Στο κάτω μέρος τους έχουν έναν αισθητήρα και ένα πομπό υπέρυθρων σημάτων με οπτική γωνία 120°. Έτσι, όταν θέλουν να στείλουν ένα μήνυμα σε κάποιο άλλο Kilobot, ο πομπός στέλνει ένα κωδικοποιημένο μήνυμα προς όλες τις κατευθύνσεις. Για να το λάβει αυτό το μήνυμα ένα άλλο Kilobot πρέπει να είναι στην κατάσταση της “παρατήρησης”, δηλαδή να περιμένει, να έχει τον αισθητήρα του ανοιχτό και να μην στέλνει κάποιο μήνυμα εκείνη τη στιγμή. Υπάρχει μια μικρή καθυστέρηση στη διαδικασία, αλλά επειδή κινούνται πολύ αργά και έχουν μία ακτίνα επικοινωνίας, περίπου στα 7cm, όταν 2 Kilobot έρθουν σε αυτή την απόσταση είναι θέμα δευτερολέπτων μέχρι να λάβουν το μήνυμα.



Εικόνα 3.3 Υπέρυθρη επικοινωνία μεταξύ 2 Kilobot. Για να επιτευχθεί επικοινωνία μεταξύ 2 Kilobot πρέπει το ένα να είναι σε λειτουργία αποστολής και το άλλο σε λειτουργία λήψης δεδομένων. Ένα Kilobot δεν μπορεί να λαμβάνει και να στέλνει δεδομένα ταυτόχρονα. [Michael Rubenstein, Harvard University]

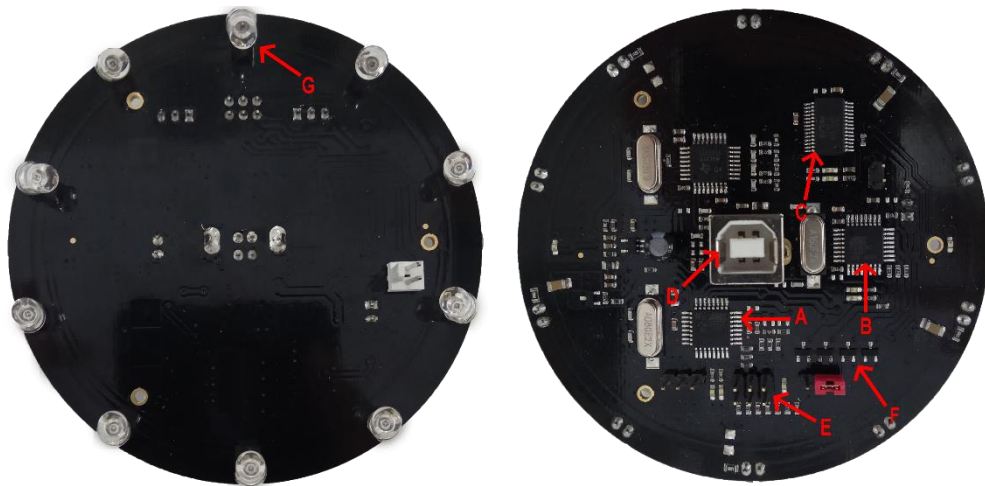
Έλεγχος

Τα Kilobot προγραμματίζονται και ελέγχονται από την πλακέτα ελέγχου, η οποία μπορεί να βρίσκεται πάνω (Over-Head Controller, OHC) ή κάτω από την επιφάνεια κίνησης [19], και από το γραφικό περιβάλλον KiloGUI (βλέπε Κεφάλαιο 4.3). Η πλακέτα που θα κατασκευάσουμε εμείς θα είναι ένας OHC (βλέπε Κεφάλαιο 4).

Κεφάλαιο 4^ο - Over-Head Controller (OHC)

Για τον προγραμματισμό και τον έλεγχο των Kilobots χρησιμοποιείται ο OHC (Εικόνα 4.1) που διανέμεται από την εταιρία κατασκευής τους, την K-Team [20].

Ο OHC σε συνδυασμό με το γραφικό περιβάλλον KiloGUI (βλέπε Κεφάλαιο 4.3), μπορεί να χειριστεί ένα μεγάλο αριθμό Kilobot ταυτόχρονα.



Εικόνα 4.1 Κάτω όψη (Αριστερά) και πάνω όψη (Δεξιά) του OHC της εταιρίας K-Team. Τα κύρια στοιχεία του OHC είναι: (A) Μικροελεγκτής Atmega328, (B) Μικροελεγκτής AT90USB162, (C) Μετατροπέας USB σε σειριακή επαφή FT232R, (D) Υποδοχή USB τύπου-B για σύνδεση με τον υπολογιστή, (E) Κεφαλή για ανέβασμα του Bootloader σε ένα Kilobot μέσω του ISP (In-System Programmer), (F) MOSFET (ZXMN2B01F) ως διακόπτες και (G) 10 υπέρυθρα LED.

Ο OHC επικοινωνεί με τα Kilobot που βρίσκονται από κάτω του με υπέρυθρα σήματα επιτρέποντας στο χρήστη να:

- Στείλει ταυτόχρονα σε όλα μαζί καινούργιο πρόγραμμα.
- Ελέγξει τη συμπεριφορά τους (π.χ. ξεκίνημα, παύση και απενεργοποίηση).
- Κάνει βαθμονόμηση των κινητήρων τους.
- Ανιχνεύσει και να διορθώσει λάθη στο πρόγραμμά τους χρησιμοποιώντας τη σειριακή θύρα.

Αυτές οι ενέργειες μπορούν να πραγματοποιηθούν για κάθε Kilobot που βρίσκεται σε διάμετρο ενός μέτρου (όταν ο OHC βρίσκεται σε ύψος ενός μέτρου παράλληλα με την επιφάνεια), εκτός από τη σύνδεση με τη σειριακή θύρα που γίνεται μόνο ένα Kilobot τη φορά..

Ο OHC μπορεί να προγραμματίσει και ατομικά Kilobot χρησιμοποιώντας τον ενσωματωμένο προγραμματιστή ISP (In-System Programmer). Με τον ISP μπορούμε να προγραμματίσουμε ένα μικροελεγκτή επάνω στην πλακέτα που βρίσκεται (αυτή η τεχνική είναι ιδιαίτερα χρήσιμη σε υλικά επιφανειακής στήριξης όπου η μεταχείρισή τους είναι πιο δύσκολη), χωρίς να χρειάζεται να τον αφαιρέσουμε και να τον προγραμματίσουμε σε μια αναπτυξιακή πλακέτα. Επίσης, μπορεί να συνδεθεί με τη σειριακή θύρα του Kilobot και να πάρει δεδομένα από το ρομπότ και να τα εμφανίσει στην οθόνη του υπολογιστή.

Ο OHC της K-Team είναι πολύ καλός για σχετικά μικρές επιφάνειες, αλλά στα περισσότερα πειράματα συλλογικής συμπεριφοράς χρειάζεται να καλύψουμε επιφάνειες μεγαλύτερες από $1m^2$ (που είναι η επιφάνεια που μπορεί να καλύψει επαρκώς ο συγκεκριμένος OHC). Οπότε για να καλύψουμε μια μεγάλη περιοχή χρειάζεται είτε να μετακινεί ο χρήστης με ένα μεγάλο καλώδιο τον OHC, κάτι το οποίο παίρνει χρόνο, είναι κουραστικό για το χρήστη και τα Kilobot δεν βρίσκονται σε συγχρονισμό εξαιτίας της διαφοράς στη χρονική στιγμή της έναρξης του προγράμματος τους, είτε να τοποθετηθούν περισσότεροι OHC σε μία βάση πάνω από την επιφάνεια κίνησης των Kilobot. Ωστόσο, το κόστος του OHC και η δυσκολία χειρισμού πολλών πλακετών μαζί, είναι ένας αποθαρρυντικός παράγοντας για τη δεύτερη λύση.

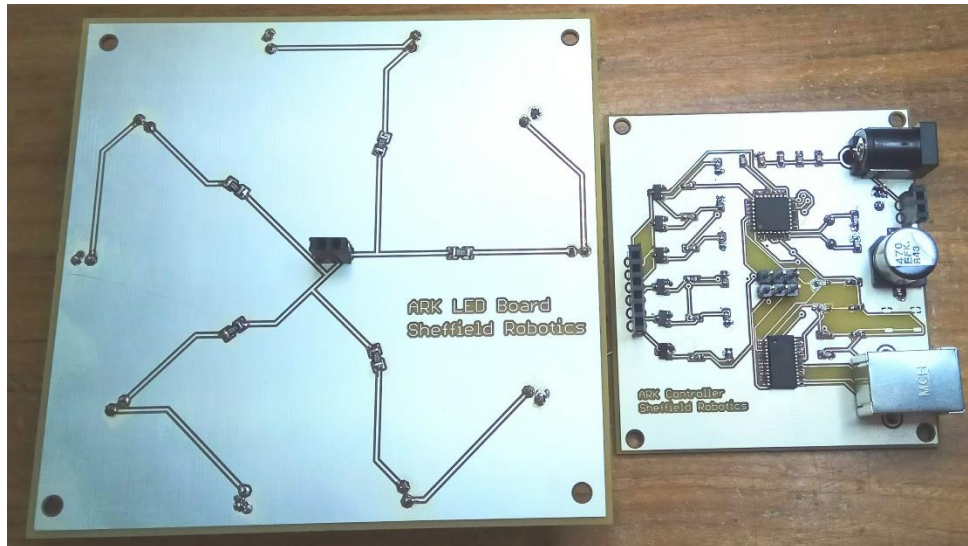
Για την αντιμετώπιση των παραπάνω προβλημάτων θα κατασκευάσουμε ένα νέο σύστημα OHC (βλέπε Εικόνα 4.2), τροποποιώντας τα τρέχοντα σχέδια και λογισμικό, τα οποία είναι διαθέσιμα για μη εμπορική χρήση [21].

Τα βήματα που θα ακολουθήσουμε για την κατασκευή του νέου συστήματος είναι :

1. Κατασκευή σχηματικών.
2. Τεστ για την επιλογή των υπέρυθρων LED.
3. Ανέβασμα του Bootloader.
4. Αλλαγές στον κώδικα, compile και ανέβασμα.
5. Τεστ για την βελτίωση της υπέρυθρης επικοινωνίας μεταξύ Kilobot και OHC.

Το νέο σύστημα του OHC (βλέπε Εικόνα 4.2) μπορεί να χωριστεί σε 2 μέρη:

- Πλακέτα ελέγχου (Control board)
- Πλακέτες υπέρυθρων LED (LED board)



Εικόνα 4.2 (Αριστερά) Πλακέτα υπέρυθρων LED. (Δεξιά) Πλακέτα ελέγχου.

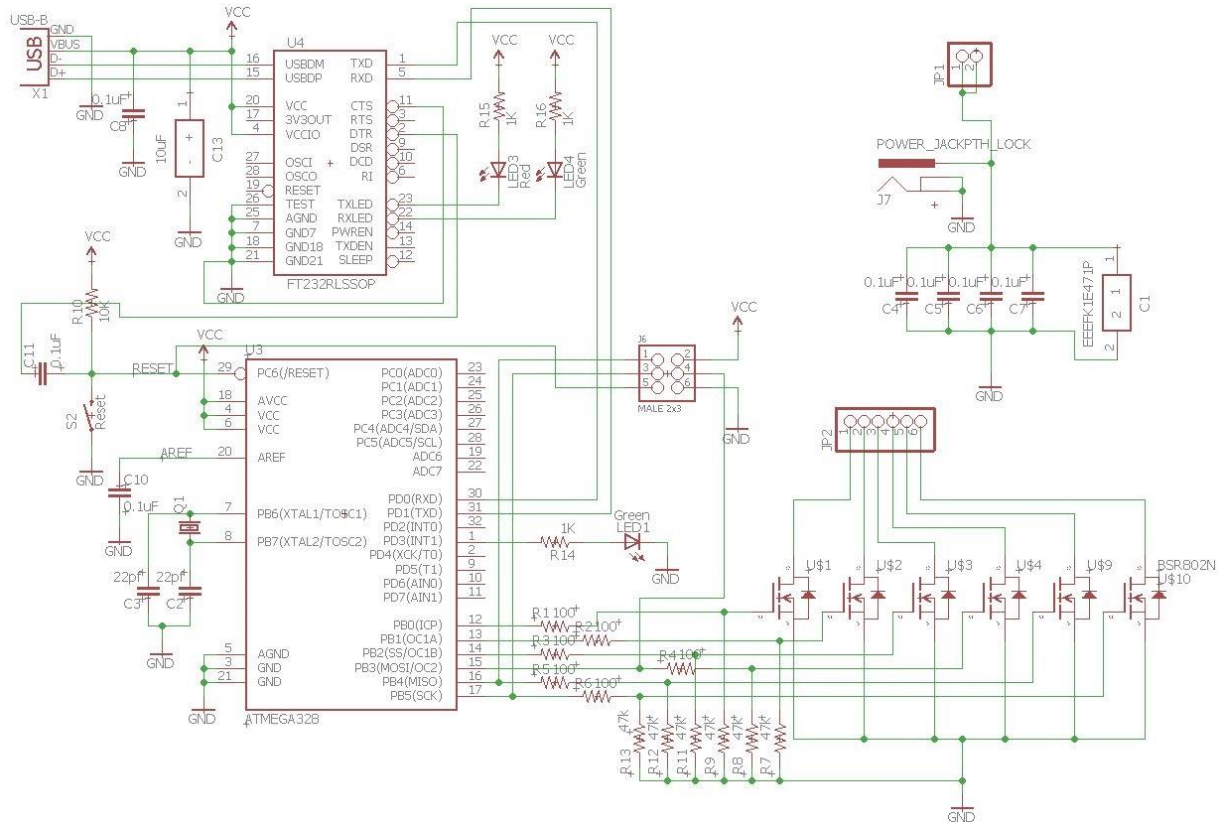
4.1 Πλακέτα ελέγχου

Σκοπός της πλακέτας ελέγχου είναι να δέχεται τις εντολές του υπολογιστή, να τις επεξεργάζεται και έπειτα να τις στέλνει στα Kilobot μέσω της πλακέτας υπέρυθρων LED (βλέπε Κεφάλαιο 4.2).

Η πλακέτα ελέγχου (Σχήμα 4.3) αποτελείται από:

- Ένα μετατροπέα σημάτων από USB σε σειριακή διεπαφή (FT232RL), η οποία επιτρέπει στον μικροελεγκτή να αναγνωρίζει τα δεδομένα που στέλνει ο υπολογιστής.
- Ένα μικροελεγκτή της ATMEL (Atmega328-AU), ο οποίος χρησιμοποιεί τον Bootloader του Arduino pro mini και χρησιμοποιείται για την επεξεργασία όλων των εντολών του υπολογιστή προς τα Kilobot.
- 4 διακόπτες MOSFET (BSR802N), που ελέγχουν την γείωση της πλακέτας υπέρυθρων LED.
- Ένα σύμπλεγμα πυκνωτών συνδεδεμένο στην τροφοδοσία των 12V, ώστε να ομαλοποιούν το σήμα από τις γρήγορες εναλλαγές των MOSFET.
- 2 LED (κόκκινο και πράσινο, LED3 και LED4), τα οποία μας δείχνουν πότε στέλνουμε εντολές ή ανεβάζουμε ένα πρόγραμμα από τον υπολογιστή στο μικροελεγκτή.
- Ένα πράσινο LED (LED1), το οποίο μένει αναμμένο για όσο χρόνο στέλνουμε μία εντολή.
- Μία κεφαλή 2x3, που χρησιμοποιείται για το ανέβασμα του Bootloader στον μικροελεγκτή.
- Μία κεφαλή (JP1), στην οποία συνδέεται η τροφοδοσία της πλακέτας υπέρυθρων LED.

- Μία κεφαλή (JP2), η οποία χρησιμοποιείται για να γειώνει την πλακέτα υπέρυθρων LED μέσω των διακοπών (MOSFET).



Σχήμα 4.3 Σχηματικό πλακέτας ελέγχου. Τα βασικά στοιχεία είναι: ο μετατροπέας USB σε σειριακή διεπαφή (FT232RL), ο μικροελεγκτής Atmega328 που χρησιμοποιεί τον Bootloader του Arduino pro mini και διαχειρίζεται όλες τις εντολές του υπολογιστή, η επαφή 2x3 για το ανέβασμα του bootloader στον μικροελεγκτή, οι διακόπτες MOSFET (BSR802N) που χρησιμοποιούνται για να γειώνουν τις πλακέτες υπέρυθρων LED, την επαφή JP1 που χρησιμοποιείται για την τροφοδοσία των υπέρυθρων LED και την επαφή JP2 που συνδέονται οι γειώσεις των πλακετών με τα υπέρυθρα LED.

4.1.1 Διαδικασία προγραμματισμού της πλακέτας ελέγχου

Bootloader

Πρώτα πρέπει να ανεβάσουμε στον μικροελεγκτή τον Bootloader (πρόγραμμα το οποίο τρέχει πρώτο και μας δίνει την δυνατότητα να ελέγξουμε τον μικροελεγκτή μέσω της σειριακής θύρας, χωρίς να χρειάζεται κάθε φορά εξωτερικό προγραμματιστή για να ανεβάσουμε κάποιο πρόγραμμα) του Arduino pro mini μέσω της κεφαλής 2x3 (βλέπε Εικόνα 4.3) που είναι ένας ICSP (In-Circuit System Programmer). Για να το κάνουμε αυτό θα χρειαστούμε ένα Arduino ή έναν προγραμματιστή AVR ISP. Για τη δική μας πλακέτα θα χρησιμοποιήσουμε τον Pocket AVR Programmer [22], της Sparkfun, συνδέοντας το καλώδιο ταινίας (ribbon cable) που παρέχεται, στην κεφαλή της πλακέτας μας.

Το ανέβασμα του Bootloader μπορεί να γίνει είτε από περιβάλλον Windows, είτε από Linux. Η μόνη προϋπόθεση είναι να είναι εγκατεστημένο και ρυθμισμένο το πρόγραμμα WinAVR [23] για την ταυτότητα του μικροελεγκτή μας. Ο πιο εύκολος τρόπος είναι από περιβάλλον Linux, επειδή έχουν ήδη εγκατεστημένη την ταυτότητα του μικροελεγκτή που θα χρησιμοποιήσουμε.

Διαδικασία για το ανέβασμα του Bootloader στο μικροελεγκτή:

1. Κατέβασμα του κώδικα από το GitHub [24].
2. Άνοιγμα ενός τερματικού (terminal) στη διαδρομή του αρχείου του bootloader.
3. Ανάθεση τιμών στις 3 ασφάλειες του μικροελεγκτή (hfuse, lfuse και efuse).
4. Ανέβασμα του αρχείου που περιέχει τον κώδικα του Bootloader.

Για τα βήματα 3 και 4 χρησιμοποιούμε 2 εντολές στο τερματικό.

- `sudo avrdude -b 57600 -c usbtiny -p m328 -v -e -U efuse:w:0xFD:m -U hfuse:w:0xDA:m -U lfuse:w:0xFF:m`
- `sudo avrdude -b 57600 -c usbtiny -p m328 -v -e -U flash:w:ATmegaBOOT_168_atmega328.hex -U lock:w:0x0F:m`

Επεξήγηση εντολών:

- `sudo`: Εντολή των Linux για εκτέλεση της εντολής που ακολουθεί με τα δικαιώματα του διαχειριστή.
- `avrdude`: Εντολή του προγράμματος WinAVR που μας επιτρέπει να εκτελέσουμε κάποιες λειτουργίες σε ένα μικροελεγκτή μέσω της γραμμής εντολών.
- `-b 57600`: Με την παράμετρο `-b` ορίζουμε το ρυθμό μετάδοσης συμβόλων (Baud rate). Αυτός ο ρυθμός είναι διαφορετικός για κάθε σύστημα. Με αυτό το νούμερο δηλώνουμε σε ποιο ρυθμό ανιχνεύει τα δεδομένα ο μικροελεγκτής, ώστε να μπει σε λειτουργία “bootloader” και να έχουμε πρόσβαση στη μνήμη του μικροελεγκτή και να τον προγραμματίσουμε.
- `-c usbtiny`: Με την παράμετρο `-c` ορίζουμε τον προγραμματιστή που θα χρησιμοποιήσουμε για να ανεβάσουμε το πρόγραμμα στον μικροελεγκτή.
- `-p m328`: Με την παράμετρο `-p` ορίζουμε τον μικροελεγκτή που θα χρησιμοποιήσουμε.
- `-v`: Ενεργοποιούμε την αναλυτική εκτύπωση των ενεργειών που πρόκειται να εκτελεστούν (συνήθως ονομάζεται με τον Αγγλικό όρο “verbose output”).
- `-e`: Προκαλεί καθαρισμό της μνήμης ROM του μικροελεγκτή ώστε να μπορεί να προγραμματιστεί ξανά.

- -U: Πραγματοποιεί μια ενέργεια στη μνήμη. Στην περίπτωση της πρώτης εντολής (-U efuse:w:0xFD:m -U hfuse:w:0xDA:m -U lfuse:w:0xFF:m) έχουμε τον προγραμματισμό τριών ασφαλειών (efuse, hfuse και lfuse). Στη δεύτερη εντολή (-U Flash:w:ATmegaBOOT_168_atmega328.hex -U lock:w:0x0F:m), έχουμε το ανέβασμα του αρχείου με επέκταση “.hex” (που πρέπει να βρίσκεται στη διαδρομή της γραμμής εντολών) και μια δεύτερη εντολή (-U lock:w:0x0F:m) που κλειδώνει τον κώδικα που έχει μέσα, ώστε να απομονώσει τον Bootloader και να μην υπάρχει πρόσβαση του κώδικα με απλές τεχνικές.

Για στιγμιότυπα από τη διαδικασία ανεβάσματος του Bootloader παρακαλώ ανατρέξτε στο Παράρτημα 1.

Ανέβασμα του κώδικα

Με τον Bootloader έτοιμο στον μικροελεγκτή μπορούμε να ανεβάσουμε το πρόγραμμα του OHC. Το πρόγραμμα που παρέχεται από τους δημιουργούς [25] είναι για μία πλακέτα LED, αλλά εμείς έχουμε 6 (βλέπε Κεφάλαιο 4.2). Οι μόνες αλλαγές που πρέπει να γίνουν είναι στις γραμμές 33 έως 36 του αρχείου “ohc.c” (βλέπε Εικόνα 4.4).

Αρχικό	Επεξεργασμένο
23 <code>#if defined(BIGOHC)</code>	23 <code>#if defined(BIGOHC)</code>
24 <code>#define ir_port PORTD</code>	24 <code>#define ir_port PORTD</code>
25 <code>#define ir_ddr DDRD</code>	25 <code>#define ir_ddr DDRD</code>
26 <code>#define ir_mask (1<<4)</code>	26 <code>#define ir_mask (1<<4)</code>
27 <code>#define led_port PORTC</code>	27 <code>#define led_port PORTC</code>
28 <code>#define led_ddr DDRC</code>	28 <code>#define led_ddr DDRC</code>
29 <code>#define led_mask (1<<0)</code>	29 <code>#define led_mask (1<<0)</code>
30 <code>#elif defined(ARDUINO)</code>	30 <code>#elif defined(ARDUINO)</code>
31 <code>#define ir_port PORTB</code>	31 <code>#define ir_port PORTB</code>
32 <code>#define ir_ddr DDRB</code>	32 <code>#define ir_ddr DDRB</code>
33 <code>#define ir_mask (1<<1)</code>	33 <code>#define ir_mask 0b00111111 PORTB</code>
34 <code>#define led_port PORTB</code>	34 <code>#define led_port PORTD</code>
35 <code>#define led_ddr DDRB</code>	35 <code>#define led_ddr DDRD</code>
36 <code>#define led_mask (1<<5)</code>	36 <code>#define led_mask 0b00001000 PORTD</code>
37 <code>#else</code>	37 <code>#else</code>
38 <code>#define ir_port PORTD</code>	38 <code>#define ir_port PORTD</code>
39 <code>#define ir_ddr DDRD</code>	39 <code>#define ir_ddr DDRD</code>
40 <code>#define ir_mask (1<<3)</code>	40 <code>#define ir_mask (1<<3)</code>
41 <code>#define led_port PORTB</code>	41 <code>#define led_port PORTB</code>
42 <code>#define led_ddr DDRB</code>	42 <code>#define led_ddr DDRB</code>
43 <code>#define led_mask (1<<1)</code>	43 <code>#define led_mask (1<<1)</code>
44 <code>#endif</code>	44 <code>#endif</code>

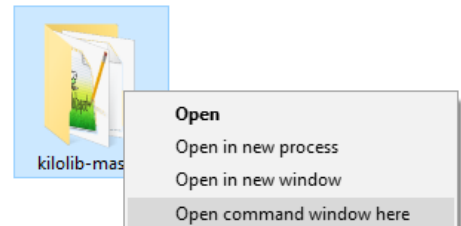
Εικόνα 4.4 Αρχικό και επεξεργασμένο κομμάτι του κώδικα του αρχείου “ohc.c”. Στη γραμμή 33 αλλάζουμε την έξοδο του σήματος από την πόρτα B και το κάνουμε να στέλνει το σήμα στα πρώτα 6 bit αντί για μόνο στο πρώτο. Στις γραμμές 34-36 αλλάζουμε την πόρτα και το bit εξόδου που ελέγχει το LED1 της πλακέτας ελέγχου.

Στη γραμμή 33 με την νέα εντολή στέλνουμε την έξοδο του σήματος στα πρώτα 6 bit της πόρτας B (PORTB) αντί μόνο για το πρώτο και στις γραμμές 34-36 αλλάζουμε την πόρτα και το bit εξόδου που ελέγχει το LED1 (βλέπε Σχήμα 4.3).

Για να ανεβάσουμε το πρόγραμμα πρέπει πρώτα να το κάνουμε compile, ώστε να παράγουμε το αρχείο μορφής “.hex”, το οποίο αναγνωρίζει ο μικροελεγκτής. Για να γίνει αυτό πρέπει να έχουμε εγκατεστημένο το πρόγραμμα WinAVR όπως στο ανέβασμα του bootloader. Αυτό το βήμα θα το πραγματοποιήσουμε από περιβάλλον Windows.

Διαδικασία compile και ανέβασμα του προγράμματος στον μικροελεγκτή:

1. Πατώντας το πλήκτρο “SHIFT” στο πληκτρολόγιο κάνουμε δεξί κλικ στο φάκελο που βρίσκεται το πρόγραμμά μας με τις βιβλιοθήκες του Kilobot και στη συνέχεια πατάμε “Ανοιγμα παραθύρου εντολών εδώ” ή “Open command window here” (Εικόνα 4.5).



Εικόνα 4.5 Απεικόνιση του ανοίγματος παραθύρου εντολών σε συγκεκριμένο φάκελο.

2. Μόλις ανοίξει το παράθυρο εντολών γράφουμε την εντολή “make ohc-arduino-16mhz” και πατάμε “ENTER”, ώστε να ξεκινήσει η διαδικασία του compile (Εικόνα 4.6α). Μόλις τελειώσει η διαδικασία (διατηρούμε το παράθυρο ανοιχτό) θα πρέπει να υπάρχει ένας φάκελος με το όνομα “build” (μέσα στο φάκελο “kilolib-master”) και μέσα να υπάρχει το αρχείο με το όνομα που του δώσαμε (“ohc-arduino-16mhz”) με την επέκταση “.hex” (Εικόνα 4.6β), το οποίο περιέχει δυαδική πληροφορία σε μορφή κειμένου ASCII (American Standard Code for Information Interchange).

Name	Type	Size
ohc-arduino-16mhz.elf	ELF File	15 KB
ohc-arduino-16mhz.hex	HEX File	11 KB

Εικόνα 4.6 (α)(Πάνω) Διαδικασία compile του προγράμματός μας στο παράθυρο εντολών. (β)(Κάτω) Το αρχείο με επέκταση “.hex”.

3. Πριν προχωρήσουμε με το ανέβασμα του κώδικα, πρέπει να συνδέσουμε με ένα καλώδιο USB τύπου B την πλακέτα ελέγχου με τον υπολογιστή.

4. Επιστρέφουμε τώρα στο παράθυρο εντολών, γράφουμε “cd build” και πατάμε “ENTER”, ώστε να έχουμε πρόσβαση στα αρχεία αυτού του φακέλου.

5. Ελέγχουμε αν υπάρχει επικοινωνία με τον μικροελεγκτή με την εντολή “avrdude -c arduino -p atmega328p -P COM9 -b 57600”. Στην εντολή αυτή ορίζουμε ότι:

- Ο προγραμματιστής που θα χρησιμοποιήσουμε είναι το Arduino
- Ο μικροελεγκτής που θα προγραμματίσουμε είναι ο Atmega328p.
- Η θύρα του υπολογιστή που θα χρησιμοποιήσουμε στη δική μας περίπτωση είναι η COM9 (βλέπε Εικόνα 4.7). Για να ελέγξουμε ποια θύρα χρησιμοποιούμε πρέπει να πάμε στη “διαχείριση συσκευών” (device manager) που βρίσκεται μέσα στον “πίνακα ελέγχου” (control panel).
- Ο ρυθμός μετάδοσης συμβόλων (Baud Rate) είναι 57600 Bd(baud).



Εικόνα 4.7 Παράθυρο διαχείρισης συσκευών με κυκλωμένη τη θύρα που χρησιμοποιεί η πλακέτα ελέγχου.

6. Μόλις ολοκληρωθεί ο έλεγχος μπορούμε να ανεβάσουμε το πρόγραμμα με την εντολή “avrdude -c Arduino -p atmega328p -P COM9 -b 57600 -U flash:w:ohc-arduino-16mhz.hex:i” (βλέπε Εικόνα 4.8).

Τα παραπάνω βήματα μπορούν να πραγματοποιηθούν και από περιβάλλον Linux αλλάζοντας την σειριακή θύρα, επειδή στα Linux έχει διαφορετική ονομασία.

Για να βρούμε ποια θύρα χρησιμοποιούμε γράφουμε σε ένα τερματικό την εντολή “dmesg | grep tty”. Στη συνέχεια θα μας εμφανίσει όλες τις συσκευές USB που χρησιμοποιούν σειριακή θύρα. Το όνομα θα ξεκινάει από “tty” και θα τελειώνει με ένα νούμερο. Στη δικιά μας περίπτωση οι εντολές του βήματος 5 και 6 γίνονται:

- avrdude -c Arduino -p atmega328p -P ttyUSB0 -b 57600
- avrdude -c Arduino -p atmega328p -P ttyUSB0 -b 57600 -U flash:w:ohc-arduino-16mhz.hex:i


```
C:\Windows\system32\cmd.exe

C:\Users\...\Desktop\build>avrdude -c Arduino -p atmega328p -P COM9 -b 57600

avrdude: AVR device initialized and ready to accept instructions

Reading | ##### | 100% 0.02s

avrdude: Device signature = 0x1e950f
avrdude: safemode: Fuses OK
avrdude done. Thank you.

C:\Users\...\Desktop\build>avrdude -c Arduino -p atmega328p -P COM9 -b 57600 -U flash:w:ohc-arduino-16mhz.hex:i

avrdude: AVR device initialized and ready to accept instructions

Reading | ##### | 100% 0.00s

avrdude: Device signature = 0x1e950f
avrdude: NOTE: FLASH memory has been specified, an erase cycle will be performed
To disable this feature, specify the -D option.
avrdude: erasing chip
avrdude: reading input file "ohc-arduino-16mhz.hex"
avrdude: writing flash (3736 bytes):

Writing | ##### | 100% 1.88s

avrdude: 3736 bytes of flash written
avrdude: verifying flash memory against ohc-arduino-16mhz.hex:
avrdude: load data flash data from input file ohc-arduino-16mhz.hex:
avrdude: input file ohc-arduino-16mhz.hex contains 3736 bytes
avrdude: reading on-chip flash data:

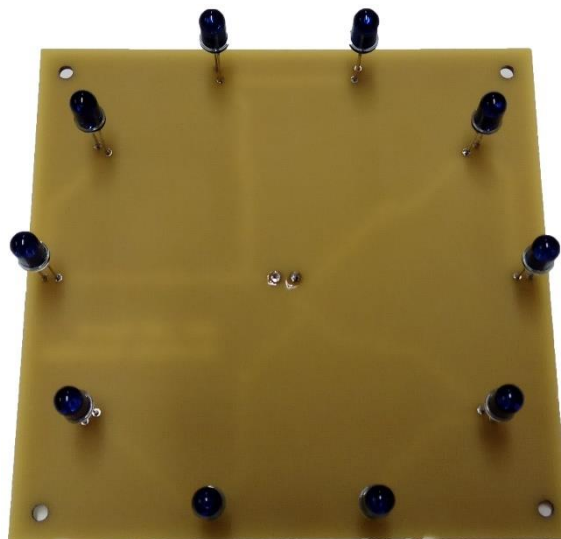
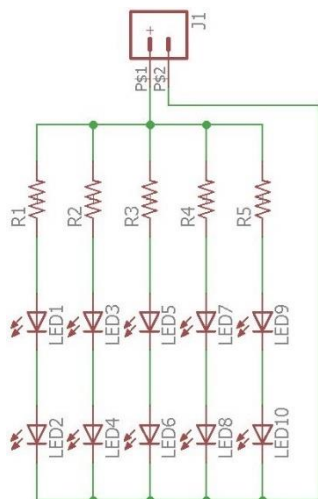
Reading | ##### | 100% 1.62s

avrdude: verifying ...
avrdude: 3736 bytes of flash verified
avrdude: safemode: Fuses OK
avrdude done. Thank you.
```

Εικόνα 4.8 Παράθυρο εντολών με τον έλεγχο της πλακέτας και το ανέβασμα του αρχείου.

4.2 Πλακέτα υπέρυθρων LED

Στην πλακέτα ελέγχου (Εικόνα 4.3) η επαφή JP1 και JP2 βρίσκονται για να συνδέουμε τις πλακέτες υπέρυθρων (IR) LED, οι οποίες στέλνουν το κωδικοποιημένο σήμα στα Kilobot (περισσότερες πληροφορίες στο Κεφάλαιο 4.5).



Εικόνα 4.9 Πλακέτα υπέρυθρων LED. (Αριστερά) Το σχηματικό της πλακέτας. (Δεξιά) Η πρωτότυπη πλακέτα.

Η κάθε πλακέτα αποτελείται από 10 υπέρυθρα LED, 5 αντιστάσεις και μία επαφή (βλέπε Εικόνα 4.9). Τα LED είναι τοποθετημένα σε κυκλική διάταξη με διάμετρο 9cm και σε ύψος 2,5cm, ώστε να μπορούμε να τα κατευθύνουμε πιο εύκολα και να διανέμουμε το φως τους καλύτερα. Από την

αριστερή υποδοχή της επαφής J1 (P\$1) παίρνουν 12V από την τροφοδοσία (JP1) και από την δεξιά υποδοχή της J1 (P\$2) συνδέονται στην επαφή JP2 στην πλακέτα ελέγχου.

4.2.1 Τεστ για την επιλογή των υπέρυθρων LED

Για την επιλογή των κατάλληλων LED εκτελέσαμε ένα τεστ στο οποίο δοκιμάσαμε 4 σετ υπέρυθρων LED με διαφορετικά χαρακτηριστικά (Πίνακας 4.10).

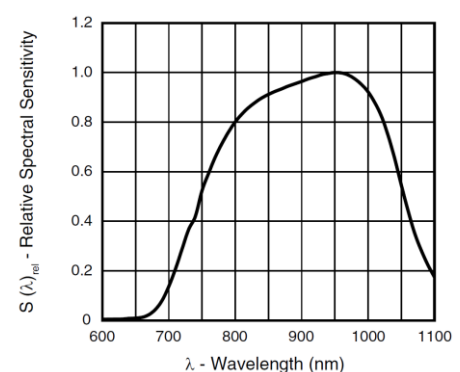
Τα κύρια χαρακτηριστικά ενός LED είναι:

- Το μήκος κύματος (wavelength)
- Η ακτινοβολούμενη ισχύς (radiant flux)
- Η ένταση ακτινοβολίας (radiant intensity)
- Η μισή γωνία (half-angle)

LED	Μήκος κύματος	Μισή-Γωνία	Ένταση Ακτινοβολίας	Ακτινοβολούμενη Ισχύς
SFH4554	860nm	+10°	>=250mW/sr	75mW
TSFF5510	870nm	+ 38°	32mW/sr	55mW
TSHF6410	890nm	+ 22°	70mW/sr	50mW
TSAL6200	940nm	+ 17°	72 mW/sr	40mW
OED-EL-1L1	940nm	+ 10°	120mW/sr	N/A
SFH4544	950nm	+10°	>=250mW/sr	75mW

Πίνακας 4.10 Τα LED που χρησιμοποιήθηκαν στο τεστ και τα χαρακτηριστικά

Το μήκος κύματος ορίζεται από την συχνότητα του κύματος (στην περίπτωση του φωτός το ηλεκτρομαγνητικό κύμα) και είναι καθοριστικό στοιχείο στην υπέρυθρη επικοινωνία. Το υπολογίζουμε από τον τύπο: $\lambda = u/f$, όπου το u είναι η ταχύτητα διάδοσης του κύματος (στην περίπτωσή μας η ταχύτητα του φωτός) και το f είναι η συχνότητα του κύματος. Επίσης, καθορίζει την περιοχή του ηλεκτρομαγνητικού φάσματος που μπορούμε να χρησιμοποιήσουμε, ώστε να έχουμε την καλύτερη σύνδεση με τον δέκτη μας. Στο Διάγραμμα 4.11 βλέπουμε τη σχετική φασματική ευαισθησία της φωτοδιόδου που χρησιμοποιούν τα Kilobot, προς το μήκος κύματος. Η συγκεκριμένη φωτοδίοδος είναι πιο ευαίσθητη σε μήκη κύματος μεταξύ 800-1025nm.

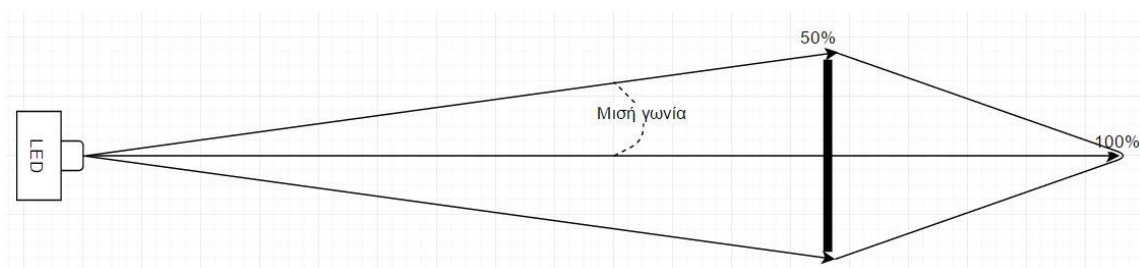


Διάγραμμα 4.11 Μήκος κύματος – Σχετική φασματική ευαισθησία της φωτοδιόδου που χρησιμοποιούν τα Kilobot [11]

Η ακτινοβολούμενη ισχύς ορίζει το πόσο δυνατά εκπέμπουν τα LED σε μονάδες Watt.

Η ένταση ακτινοβολίας είναι η μέτρηση της ακτινοβολούμενης ισχύος ανά στερεά γωνία από μία άκρη της φωτεινής πηγής και μετριέται σε watt/steradian. Το στερακτίνιο (steradian) είναι εκείνη η στερεά γωνία η οποία, όταν γίνει επίκεντρη ορίζει σφαιρική περιοχή, σε οποιαδήποτε σφαίρα, με εμβαδόν ίσο με το τετράγωνο της ακτίνας της [26].

Η μισή γωνία, μετριέται σχετικά με το κέντρο του κώνου εκπομπής του LED και είναι εκείνη η γωνία που η ένταση της ακτινοβολίας πέφτει στο 50% (Σχήμα 4.12). Αυτή η γωνία μας ορίζει και τη μέγιστη περιοχή που μπορούμε να καλύψουμε με ένα LED με ικανοποιητική ισχύ.



Σχήμα 4.12 Αναπαράσταση της μισής γωνίας σχετικά με την ένταση της ακτινοβολίας (50% και 100%).

Σε αυτό το τεστ χρησιμοποιήσαμε ένα κομμάτι ασπροπίνακα για επιφάνεια κίνησης και ο OHC ήταν τοποθετημένος σε ύψος 1,6m από το τραπέζι.

Διαδικασία του τεστ:

- Τοποθέτηση 10 LED του ίδιου τύπου στην πλακέτα (Εικόνα 4.9).
- Τοποθέτηση της πλακέτας στο κέντρο του ασπροπίνακα σε ύψος 1,6m.
- Τοποθέτηση ενός Kilobot στον ασπροπίνακα.
- Αποστολή ενός συνεχούς σήματος (RESET) από τον υπολογιστή (βλέπε Κεφάλαιο 4.3). Όσο το Kilobot λαμβάνει αυτό το σήμα, το LED τριών καταστάσεων παραμένει πράσινο. Όταν χάσει εντελώς επαφή με τον OHC τότε αναβοσβήνει πράσινο.
- Μετακίνηση του Kilobot σε 4 σημεία του ασπροπίνακα (πάνω, κάτω, δεξιά και αριστερά από το κέντρο του OHC), μέχρι να αρχίσουμε να χάνουμε επαφή με τον OHC.
- Μέτρηση της απόστασης που το Kilobot διατηρεί επικοινωνία με τον OHC.
- Καταγραφή των αποτελεσμάτων σε εκατοστά (Πίνακας 4.12).

	LED	#1	#2	#3
Ακτίνα	Δεξιά	110	60	89
	Αριστερά	55.5	37	56
	Πάνω	105	64	79
	Κάτω	44.5	41	48

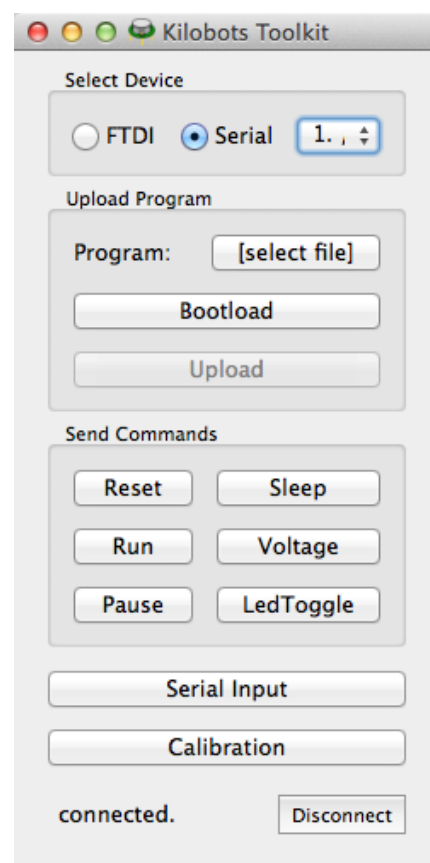
Πίνακας 4.13 Τα αποτελέσματα του τεστ με τα 3 σετ διαφορετικών LED σε εκατοστά. (#1) TSAL6200, (#2) OED-EL-1L1, (#3) TSHG6410.

Όπως φαίνεται από τα αποτελέσματα το LED #1 (TSAL6200) ήταν αυτό που είχε τη μεγαλύτερη διάμετρο (συνδυάζοντας δεξιά-αριστερά και πάνω-κάτω) και αυτό που επιλέξαμε για την πλακέτα LED.

4.3 Γραφικό περιβάλλον KiloGUI

Το KiloGUI είναι ένα γραφικό περιβάλλον που χρησιμοποιούμε, για να ανεβάσουμε νέα προγράμματα στα Kilobots και να εκτελέσουμε μαζικά βασικές λειτουργίες. Επίσης, είναι ένα λογισμικό ανοιχτού κώδικα (open-source) και μπορεί να το κατεβάσει κάποιος από την ιστοσελίδα της εταιρίας [27].

Εντολή	Λειτουργία
FTDI/Serial	Επιλογή του τρόπου σύνδεσης του υπολογιστή με τον OHC (συνήθως η επιλογή γίνεται αυτόματα).
Program [Select File]	Επιλογή του .hex αρχείου που θέλουμε να ανεβάσουμε (ένα .hex αρχείο παράγεται όταν κάνουμε compile ένα πρόγραμμα γραμμένο σε κάποια γλώσσα προγραμματισμού).
Bootload	Ανοίγει τον bootloader του Kilobot για να δεχτεί καινούργιο πρόγραμμα.
Upload	Στέλνει ένα καινούργιο πρόγραμμα στα Kilobot. Το LED του ρομπότ θα αναβοσβήνει μπλε και πράσινο όσο λαμβάνει το πρόγραμμα. Η διαδικασία κρατάει το πολύ 1 λεπτό και όταν ολοκληρωθεί το LED θα αναβοσβήνει πράσινο χρώμα (που σημαίνει ότι είναι έτοιμο να δεχτεί εντολές, άεργη κατάσταση). Αν η αποστολή αποτύχει τότε το LED θα είναι μπλε.
Reset	Πάει στην αρχή του προγράμματος, επαναφέρει τις αρχικές καταστάσεις και μένει άεργο περιμένοντας την επόμενη εντολή. Επίσης “ξυπνάει” τα Kilobot από την κατάσταση Sleep.
Run	Ξεκινάει το πρόγραμμα που έχει στη μνήμη.
Pause	Κάνει μία παύση στο πρόγραμμα (διατηρεί την κατάσταση που είχε, ώστε μόλις πατήσεις το Run να συνεχίσει από εκεί που σταμάτησε).



Εικόνα 4.14 Γραφικό περιβάλλον χρήστη KiloGUI (Graphical User Interface, GUI)

Sleep	Βάζει τα Kilobot σε κατάσταση χαμηλής κατανάλωσης (δεν δέχονται εντολές και αναβοσβήνει το LED λευκό χρώμα κάθε 8 δευτερόλεπτα).
Voltage	Απεικόνιση της τάσης του Kilobot χρησιμοποιώντας το LED (πράσινο/μπλε = πλήρως φορτισμένο, κίτρινο/κόκκινο = χρειάζεται φόρτιση).
Led Toggle	Αλλάζει την κατάσταση του LED που βρίσκεται στον OHC (χρησιμοποιείται για έλεγχο της επικοινωνίας μεταξύ υπολογιστή και OHC).
Serial Input	Δείχνει τα μηνύματα που στάλθηκαν από το Kilobot μέσω ενός σειριακού καλωδίου.
Calibration	Εμφανίζει ένα νέο παράθυρο και μπορείς να θέσεις ταυτότητα και να δώσεις τιμές στους κινητήρες ενός Kilobot (σε περίπτωση άνισης κίνησής τους).

4.4 Περιβάλλον προσομοίωσης

Ένα πείραμα με Kilobot μπορεί να διαρκέσει από μερικά λεπτά, μέχρι μερικές ώρες ανάλογα με τον πληθυσμό των ρομπότ και την πολυπλοκότητα του πειράματος, εξαιτίας της αργής κίνησής τους. Εκτός από τη διάρκεια του πειράματος, χρειάζεται ειδική προετοιμασία πριν από κάθε πείραμα (π.χ. φόρτιση όλων των ρομπότ, βαθμονόμηση των κινητήρων, καθάρισμα της επιφάνειας κίνησής τους, κ.α.).

Για την εξοικονόμηση χρόνου, πριν περάσουμε στην εκτέλεση του κώδικά μας σε πραγματικό χρόνο, μπορούμε να προσομοιώσουμε τον κώδικα στο περιβάλλον του V-Rep [28].

Το **V-Rep** είναι ένα περιβάλλον προσομοίωσης για γρήγορη ανάπτυξη αλγορίθμων σε ρομπότ. Χρησιμοποιείται κυρίως για προσομοίωση αυτοματοποιημένων εργοστασιακών εφαρμογών, εξοικείωση με τη ρομποτική στην εκπαίδευση, ελέγχους ασφαλείας, κ.α..

4.4.1 Εγκατάσταση του V-Rep και αρχικές ρυθμίσεις

Για να μπορέσουμε να προσομοιώσουμε τα Kilobot σε αυτό το λογισμικό πρέπει να ακολουθήσουμε τα εξής βήματα:

1. Κατέβασμα του λογισμικού (εμείς θα χρησιμοποιήσουμε την εκπαιδευτική έκδοση) [29].
2. Εγκατάσταση του λογισμικού και των βιβλιοθηκών που θα ζητηθούν από το πρόγραμμα την ώρα της εγκατάστασης.

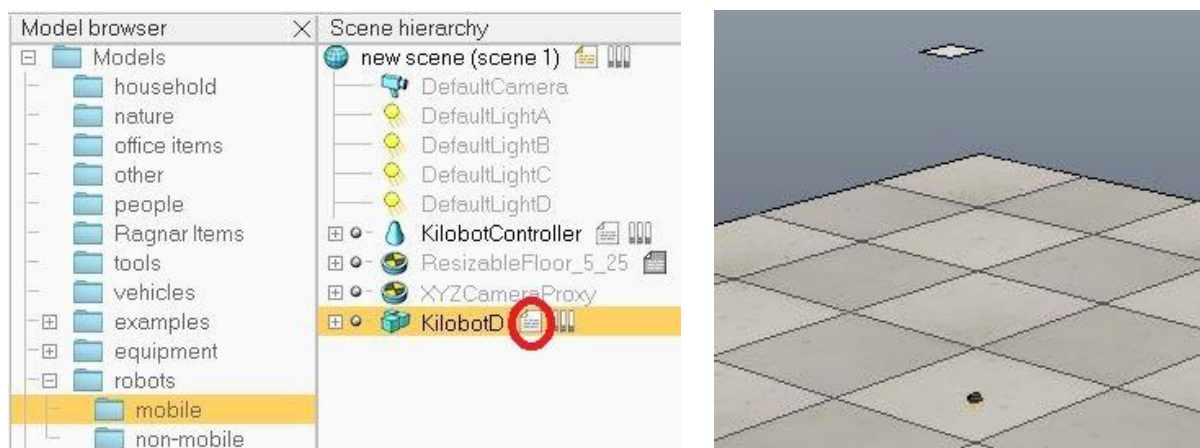
3. Κατέβασμα του μοντέλου των Kilobot (Kilobot model for V-REP simulator, 3 αρχεία) [30].
4. Αντικατάσταση του μοντέλου των Kilobot στο φάκελο εγκατάστασης του V-Rep (C:\Program Files (x86)\V-REP3\V-REP_PRO_EDU\models\robots\mobile) με τα αρχεία του προηγούμενου βήματος.

4.4.2 Χειρισμός του V-Rep

Μόλις ξεκινήσει το πρόγραμμα θα βρείτε στην αναζήτηση των μοντέλων (model browser), το μοντέλο του Kilobot και του OHC (το μοντέλο βρίσκεται στο φάκελο που βάλαμε τα αρχεία στο βήμα 4 στο Κεφάλαιο 4.4.1, model\robots\mobile).

Για να ξεκινήσει η προσομοίωση πρέπει:

- Να τοποθετήσουμε το μοντέλο του Kilobot και του OHC στο περιβάλλον προσομοίωσης (τα Kilobot πρέπει να βρίσκονται κάτω από τον OHC για να πάρουν τις εντολές).
- Να αλλάξουμε το πρόγραμμα του Kilobot (κάνοντας διπλό κλικ στο εικονίδιο δίπλα από το αντικείμενο, Εικόνα 4.15).
- Να ξεκινήσουμε την προσομοίωση, είτε από το μενού (Simulations -> Start Simulation), είτε από το εικονίδιο στη γραμμή εργαλείων.
- Να πατήσουμε το “Run” στο παράθυρο που άνοιξε στην προσομοίωση.



Εικόνα 4.15 (Αριστερά) Η αναζήτηση των μοντέλων και το παράθυρο των αντικειμένων που υπάρχουν στο περιβάλλον προσομοίωσης με κυκλωμένο το εικονίδιο για την επεξεργασία του κώδικα! (Δεξιά) Περιβάλλον προσομοίωσης με ένα Kilobot και τον OHC.

4.5 Προγραμματισμός των Kilobot

Αρχικά, πρέπει να γράψουμε το πρόγραμμα, με τη χρήση μιας τροποποιημένης έκδοσης της γλώσσας προγραμματισμού C, και να το μετατρέψουμε σε δεκαεξαδικό σύστημα (compile). Υπάρχουν 2 τρόποι για να γίνει αυτό: τοπικά ή στο διαδίκτυο.

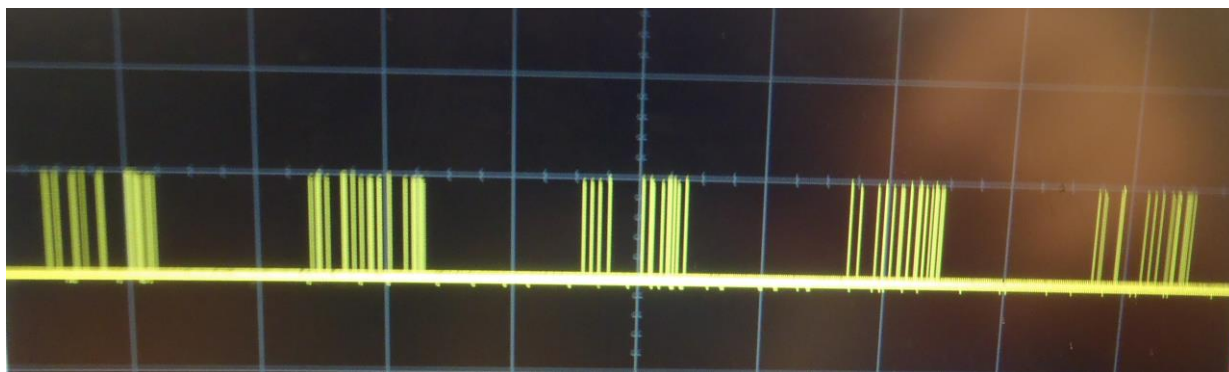
Τοπικά μπορεί να γίνει με οποιοδήποτε πρόγραμμα της αρεσκείας μας (π.χ. Eclipse [31]), αρκεί να έχουμε τις βιβλιοθήκες των Kilobot οι οποίες είναι ανοιχτού κώδικα [25].

Αφού κατεβάσουμε τις βιβλιοθήκες εντοπίζουμε μέσα το αρχείο *'blank.c'*, το οποίο περιέχει ένα πρότυπο της μορφής που πρέπει να έχει το πρόγραμμα. Στη συνέχεια, μπορούμε να χρησιμοποιήσουμε το αρχείο *'Makefile'* που παρέχεται για να κάνουμε compile το πρόγραμμα (ώστε να πάρουμε το αρχείο με επέκταση *'hex'*, το οποίο αναγνωρίζει ο μικροελεγκτής). Πριν κάνουμε compile ένα πρόγραμμα θα πρέπει να έχουμε εγκατεστημένο στον υπολογιστή μας τον avr-gcc compiler και τις βιβλιοθήκες avr-libc (οι οποίες υπάρχουν ήδη από την εγκατάσταση του WinAVR στο Κεφάλαιο 4.1.1).

Ένας πιο εύκολος τρόπος για μικρά και απλά προγράμματα είναι να γίνει αυτή η διαδικασία στο διαδίκτυο. Χρησιμοποιώντας την διαδικτυακή πλατφόρμα των Kilobot [32] και ένα λογαριασμό στο Dropbox [33] μπορούμε να γράψουμε τα προγράμματα που θέλουμε για τα Kilobot, να τα κάνουμε compile και να τα αποθηκεύσουμε στο λογαριασμό μας.

4.5.1 Επικοινωνία OHC με Kilobot

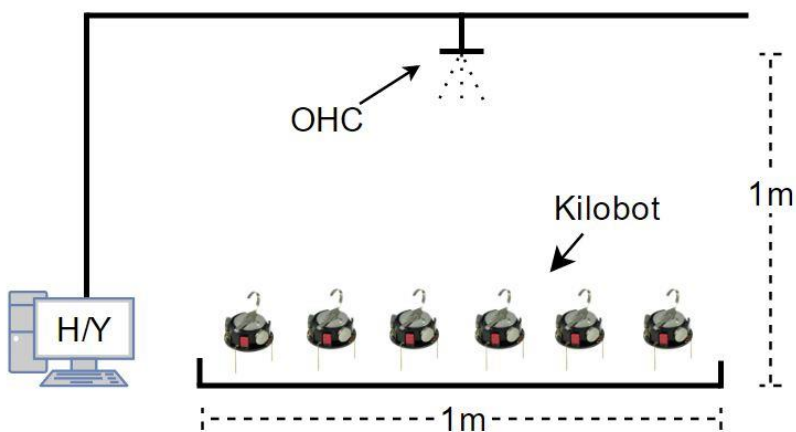
Όπως και στη επικοινωνία μεταξύ Kilobot, η επικοινωνία με τον OHC γίνεται μέσω υπέρυθρης ακτινοβολίας. Ο OHC παίρνει το αρχείο που στείλαμε από το KiloGUI και το στέλνει στα Kilobot μέσω μορφοποιημένων τετραγωνικών παλμών (Pulse Width Modulation, PWM) που δημιουργούνται από το γρήγορο άναμμα και σβήσιμο των υπέρυθρων LED (βλέπε Εικόνα 4.16). Στη συνέχεια, το πρόγραμμα αποθηκεύεται στη μνήμη των Kilobot και ο Bootloader αναλαμβάνει τότε θα τρέξει τις υπόλοιπες λειτουργίες.



Εικόνα 4.16 Τετραγωνικός παλμός της εντολής Reset.

Για να ανεβάσουμε ένα πρόγραμμα στα Kilobot πρέπει:

1. Να βρίσκονται σε άεργη κατάσταση (LED αναβοσβήνει πράσινο), δηλαδή να μην βρίσκονται σε Sleep mode.
2. Να γυρίσουμε τα Kilobot στον Bootloader, ώστε να μπορεί να δεχθεί καινούργιο πρόγραμμα ο μικροελεγκτής.
3. Να επιλέξουμε το αρχείο με επέκταση '.hex' που παράχθηκε από τον compiler και να πατήσουμε την εντολή 'Upload' από το KiloGUI, ώστε να αρχίσει η διαδικασία ανεβάσματος του κώδικα. Στη συνέχεια, το LED στα Kilobot θα παραμείνει μπλε αν το ανέβασμα απέτυχε και θα πάει σε άεργη κατάσταση αν ολοκληρώθηκε σωστά.



Εικόνα 4.17 Διαδικασία προγραμματισμού των Kilobot και εμβέλεια του OHC της K-Team

4.6 Βελτίωση της υπέρυθρης επικοινωνίας

Με την εμπορική έκδοση του OHC, η απόσταση του OHC από το τραπέζι και ο περιβάλλοντας φωτισμός παίζει σημαντικό ρόλο για την ποιότητα και την αποτελεσματικότητα της επικοινωνίας ανάμεσα στα ρομπότ, αλλά και τον προγραμματισμό τους από τον OHC. Επίσης, η δημιουργία

σκιών στην επιφάνεια του τραπέζιου από τα Kilobot και από άτομα που βρίσκονταν δίπλα στο τραπέζι δημιουργούσε παρεμβολές στην υπέρυθρη επικοινωνία.

Προκειμένου να επιλύσουμε αυτά τα προβλήματα με το καινούργιο σύστημα OHC, πραγματοποιήσαμε μερικά τεστ για να δοκιμάσουμε διαφορετικούς τύπους λαμπτήρων και να αναγνωρίσουμε την αιτία των παρεμβολών.

Σε αυτά τα τεστ χρησιμοποιήσαμε τον OHC της K-Team, για να μελετήσουμε τον φωτισμό του δωματίου και να δοκιμάσουμε διαφορετικούς τύπους λαμπτήρων. Τα τεστ πραγματοποιήθηκαν στο τραπέζι της Sheffield Robotics [34] που είναι ένα κομμάτι γυαλιού 2,2m². Για να μελετήσουμε πώς το φως επηρεάζει την υπέρυθρη επικοινωνία, διαιρέσαμε το τραπέζι σε 4 κελιά και τους δώσαμε σημεία κατευθυντικότητας (βλέπε Εικόνα 4.18).

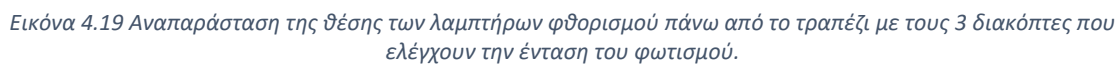


Εικόνα 4.18 Το τραπέζι της Sheffield Robotics χωρισμένο σε 4 κελιά. Στα κελιά A3 και A4 βλέπετε τα 4 σημεία κατευθυντικότητας (Πάνω, Κάτω, Δεξιά, Αριστερά).

Αφού χωρίσαμε το τραπέζι σε κελιά, μετρήσαμε (1) την ένταση της φωτεινότητας κάθε κελιού σε Lumen με ένα φωτόμετρο (με τον υπάρχων φωτισμό του δωματίου) και (2) τις διαφορές στην εμβέλεια της επικοινωνίας με άλλους 2 τύπους λαμπτήρων σε 2 διαφορετικά ύψη. Πραγματοποιήσαμε αυτά τα τεστ με **λάμπες φθορίου**, ένα **προβολέα** (λάμπα αλογόνου) και μία **ταινία LED**.

Οι λάμπες φθορίου είναι μόνιμα τοποθετημένες στην οροφή του δωματίου. Επίσης το δωμάτιο έχει 3 διακόπτες που ελέγχουν την ένταση του φωτός πάνω από το τραπέζι. Χρησιμοποιήσαμε αυτούς τους διακόπτες, για να μελετήσουμε την επίδραση που έχει η ένταση του φωτός στην υπέρυθρη επικοινωνία (βλέπε Εικόνες 4.19).

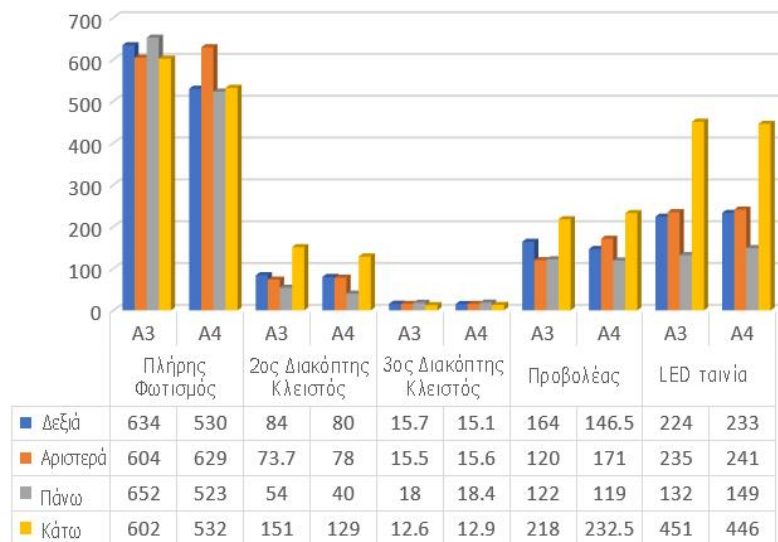
Η LED ταινία είχε λευκό χρώμα (3.500K – 4.500K), 120 LED με συνολική ισχύ 9,6W/m και ήταν τοποθετημένη σε ένα διαχωριστικό που βρισκόταν 25cm μακριά από το τραπέζι (στην πλευρά της περιοχής A3 και A4) και σε ύψος 1,5m από το έδαφος (βλέπε Εικόνα 4.20). Η γωνία της ταινίας ήταν ρυθμισμένη, ώστε να καλύπτει τη μεγαλύτερη δυνατή περιοχή στο τραπέζι και να μην δημιουργεί αντανακλάσεις στις κάμερες που βρίσκονται πάνω από το τραπέζι.



Πραγματοποιήσαμε όλα τα τεστ στις περιοχές A3 και A4 του τραπεζιού (βλέπε Εικόνα 4.18). Έτσι, συλλέξαμε δεδομένα και από διαφορετικές πηγές φωτός και από διαφορετικές εντάσεις στη φωτεινότητά τους (βλέπε Διάγραμμα 4.21).

4.6.1 Τεστ – 1: Μέτρηση της φωτεινότητας στο τραπέζι

Σε αυτό το τεστ μετρήσαμε τη φωτεινότητα που υπάρχει στην επιφάνεια του τραπεζιού με ένα φωτόμετρο, ώστε να μπορούμε να το συγκρίνουμε με τα αποτελέσματα της υπέρυθρης επικοινωνίας που είχαμε σε κάθε περιοχή.



Διάγραμμα 4.21 Ένταση ακτινοβολίας (σε Lumen) στις περιοχές A3 και A4 με 3 διαφορετικούς τύπους λαμπτήρων.

Στο διάγραμμα 4.21 φαίνεται ότι στον πλήρη φωτισμό οι 2 περιοχές δεν έχουν την ίδια ποσότητα φωτός (Lumen) και ως αποτέλεσμα υπήρχαν διαφορές στην έκταση που μπορούσε να καλύψει ο ΟΗC (όπως θα παρατηρήσουμε στα αποτελέσματα του Τεστ-2 στο Κεφάλαιο 4.6.2).

Για να μετρήσουμε την έκταση που είχαμε επικοινωνία με έναν ΟΗC, τοποθετήσαμε τον ΟΗC στο κέντρο της επιθυμητής περιοχής σε 2 διαφορετικά ύψη (102cm και 157cm από την επιφάνεια του τραπεζιού). Έπειτα, τοποθετούσαμε μερικά Kilobot ως δέκτες στην περιοχή για να μετρήσουμε το εύρος της ακτίνας που κάλυπτε ο ΟΗC.

4.6.2 Τεστ – 2: Περιοχή κάλυψης της υπέρυθρης επικοινωνίας

Στο δεύτερο τεστ, καταγράψαμε τα αποτελέσματα από την έκταση που κάλυπτε ο ΟΗC σε 2 διαφορετικά ύψη για τους τύπους λαμπτήρα που περιγράψαμε στο Κεφάλαιο 4.6.1. Για επιφάνεια κίνησης χρησιμοποιήσαμε ένα κομμάτι ασπροπίνακα 1m² το οποίο μετακινούσαμε στο τραπέζι, όταν χρειαζόταν, διατηρώντας το κέντρο του ΟΗC σταθερό.

102cm										
	A3	A4	A3	A4	A3	A4	A3	A4	A3	A4
	Πλήρης Φωτισμός		2ος Διακόπτης Κλειστός		3ος Διακόπτης Κλειστός		Προβολέας		LED ταινία	
■ Δεξιά	66	54	45	42.5	34	35	70	54	11	14.5
■ Αριστερά	57	68	57	38	38	34	57	68	13	15
■ Πάνω	76	77	55	43	39	40	76	73	17	14
■ Κάτω	57	57	57	49	34	29	57	57	11	14

Διάγραμμα 4.22 Ακτίνα επικοινωνίας, σε εκατοστά (cm), στις περιοχές A3 και A4 με διαφορετικούς τύπους λαμπτήρων σε ύψος 102cm από το τραπέζι.

157cm										
	A3	A4	A3	A4	A3	A4	A3	A4	A3	A4
	Πλήρης Φωτισμός		2ος Διακόπτης Κλειστός		3ος Διακόπτης Κλειστός		Προβολέας		LED ταινία	
■ Δεξιά	71.5	64	16	15.5	0	0	72.5	64	29	24
■ Αριστερά	53.5	68	25.5	14.5	0	0	53.5	66	32	20
■ Πάνω	70	68	13	14	0	0	68.5	70	21	18
■ Κάτω	47	61	45	16	0	0	47	61	60	53

Διάγραμμα 4.23 Ακτίνα επικοινωνίας, σε εκατοστά (cm), στις περιοχές A3 και A4 με διαφορετικούς τύπους λαμπτήρων σε ύψος 157cm από το τραπέζι.

Από τα διαγράμματα 4.22 και 4.23 φαίνεται ότι στις 2 περιοχές ο ΟΗC είχε διαφορετική κάλυψη και ότι η διαφορά ύψους δεν άλλαξε και πολύ τα αποτελέσματα. Αυτό που είναι σημαντικό είναι η κάλυψη που είχαμε με κλειστούς τους διακόπτες. Γενικά, παρατηρήθηκε μια σταθερότητα στην διάμετρο που είχαμε επικοινωνία (συνδυάζοντας Δεξιά-Αριστερά και Πάνω-Κάτω) μεταξύ των 102cm και των 157cm, αλλά με κλειστούς τους διακόπτες η επικοινωνία γινόταν πάρα πολύ μικρή στα 157cm και στην περίπτωση του 3^{ου} διακόπτη μηδενική. Επίσης, παρατηρήσαμε ότι όταν ο ΟΗC ήταν στα 157cm, η υπέρυθρη επικοινωνία γινόταν πιο ευάλωτη σε παρεμβολές (σκιές).

Εντυπωσιακά είναι και τα αποτελέσματα με την LED ταινία, γιατί είναι τα μοναδικά που έδειξαν εμφανή βελτίωση στην υπέρυθρη επικοινωνία. Τα αποτελέσματα φαίνεται να δείχνουν ότι τα σημεία κατευθυντικότητας που ήταν πιο κοντά στην ταινία έχουν μεγαλύτερη ακτίνα (βλέπε Εικόνα 4.20 για κατανόηση της θέσης της ταινίας).

4.6.3 Συμπεράσματα

Τα αποτελέσματα των Τεστ-1 και Τεστ-2, δείχνουν ότι η ένταση της φωτεινότητας (lumen) δεν επηρεάζει την έκταση της επικοινωνίας. Οι λάμπες φθορίου και η λάμπα αλογόνου, από τον προβολέα, καλύπτουν σχεδόν την ίδια περιοχή, παρά το γεγονός ότι έχουν πολύ διαφορετική ένταση (βλέπε Διάγραμμα 4.21). Το κοινό στοιχείο των 2 λαμπτήρων είναι ότι παράγουν μεγάλες ποσότητες υπέρυθρης ακτινοβολίας. Η φωτοδίοδος των Kilobot, για να ανιχνεύσει το κωδικοποιημένο σήμα πρέπει να δεχθεί σήμα μεγαλύτερης ισχύος από το «κατώφλι» της φωτοδίοδου (threshold). Το σήμα του OHC χάνει την ισχύ του σταδιακά, όσο απομακρύνεται από την πηγή (όπως κάθε ηλεκτρομαγνητικό κύμα). Έτσι, οι λάμπες φθορίου και αλογόνου καλύπτουν μεγάλο μέρος του κατωφλίου της φωτοδίοδου και χρειάζεται μικρότερη διέγερση από τον OHC, ώστε να ανιχνεύσει το σήμα. Το φαινόμενο αυτό μπορεί να επιβεβαιωθεί με τα αποτελέσματα της LED ταινίας. Τα LED έχουν μικρό εύρος μήκους κύματος και με δυσκολία ξεπερνά τα 750nm, που είναι η αρχή της υπέρυθρης ακτινοβολίας. Έτσι συμπεράναμε, ότι τα LED εκπέμπουν τη μικρότερη ποσότητα υπέρυθρου φωτός και έτσι είχαν τη μικρότερη κάλυψη στο τραπέζι (βλέπε Διαγράμματα 4.21 και 4.22). Επομένως η ποσότητα της ακτινοβολίας που δεν μπορούμε να δούμε είναι αυτή που έχει σημασία (υπέρυθρη) και όχι η ορατή (φως).

Συνοψίζοντας, παρατηρήσαμε ότι οι λάμπες φθορίου που είναι τοποθετημένες στο δωμάτιο παρέχουν κατάλληλο ορατό φως για την καταγραφή του πειράματος από τις κάμερες και υπέρυθρο φως για να λαμβάνουν τα Kilobot το σήμα από τον OHC. Ωστόσο, ήταν πιο ευάλωτο σε παρεμβολές. Ο προβολέας φαίνεται να είναι η καλύτερη επιλογή για την βέλτιστη επικοινωνία με τις λιγότερες παρεμβολές, αλλά όχι τόσο καλή με την βιντεοσκόπηση του πειράματος. Η LED ταινία, δεν είναι εύκολο να τοποθετηθεί γύρω από το τραπέζι και ταυτόχρονα να υπάρχει εύκολη πρόσβαση στα Kilobot, ενώ παράλληλα δεν παρέχει τον κατάλληλο φωτισμό για το πείραμα. Επιπλέον, ανάμεσα στις άλλες 2 επιλογές η LED ταινία κάλυπτε την μικρότερη περιοχή στο τραπέζι.

4.7 Πρόσθετες πλακέτες υπέρυθρων LED

Παρατηρώντας τα αποτελέσματα των τεστ 1 και 2, αποφασίσαμε να κατασκευάσουμε 4 καινούργιες πλακέτες οι οποίες θα στέλνουν συνεχόμενα υπέρυθρη ακτινοβολία. Οι νέες πλακέτες υπέρυθρων είναι παρόμοιες με τις πλακέτες που περιγράψαμε στο Κεφάλαιο 4.2 με τη διαφορά ότι αυτές δεν στέλνουν κωδικοποιημένα σήματα. Για το λόγο αυτό, οι πλακέτες αυτές είναι συνδεδεμένες απευθείας στην τροφοδοσία και χρησιμοποιούν αντιστάσεις 330Ωm.

Οι πρόσθετες πλακέτες καλύπτουν μεγαλύτερο κομμάτι του κατωφλίου της φωτοδιόδου των Kilobot και ανεξάρτητα από το πόσο αδύνατο είναι το κωδικοποιημένο σήμα, είναι αρκετό ώστε να καλύψει το υπόλοιπο του κατωφλίου και να το αναγνωρίσει το Kilobot.

Επιπλέον, αυτές οι πλακέτες λύσανε το πρόβλημα με τις παρεμβολές. Με 4 τέτοιες πλακέτες τοποθετημένες στις 4 πλευρές του τραπέζιού βελτιώθηκε, όχι μόνο η αντοχή του σήματος στις σκιές, αλλά και ο έλεγχος των Kilobot, ακόμα και με την απουσία του ορατού φωτός. Παρόμοιο αποτέλεσμα θα μπορούσε να πραγματοποιηθεί αν αντικαθιστούσαμε αυτές τις πλακέτες με έναν υπέρυθρο προβολέα χαμηλής ισχύος.

4.8 Αποτελέσματα

Έχοντας μελετήσει τους διάφορους τύπους λαμπτήρων (π.χ. φθορισμού, αλογόνου και LED) και με την ολοκλήρωση των τριών τεστ, συμπεραίνουμε ότι αυξάνοντας την ποσότητα της υπέρυθρης ακτινοβολίας στο τραπέζι, καταφέραμε να λύσουμε τις παρεμβολές που υπήρχανε στην υπέρυθρη επικοινωνία μεταξύ Kilobot-OHC και Kilobot-Kilobot.

Με το νέο σύστημα καταφέραμε να:

- Καλύπτουμε περιοχές μεγαλύτερες από 1m^2 , χρησιμοποιώντας 10 πλακέτες υπέρυθρου φωτισμού (6 για κωδικοποιημένα σήματα και 4 για την αύξηση της υπέρυθρης ακτινοβολίας στην περιοχή).
- Λύσουμε τα προβλήματα που υπήρχανε από τη διαφορετική ένταση του φωτισμού στο χώρο και από τις σκιές που δημιουργούσαν τα Kilobot και ο κόσμος που παρατηρούσε το πείραμα.
- Ελέγχουμε τα Kilobot ακόμα και στην απουσία ορατού φωτός.

4.9 Προβλήματα που προέκυψαν

Τα περισσότερα προβλήματα που προέκυψαν ήταν συμβατότητας λογισμικού. Κυρίως αυτά τα προβλήματα προέρχονταν από περιβάλλοντα Windows. Μία εύκολη λύση στο πρόβλημα, ήταν η εγκατάσταση εικονικού μηχανήματος (virtual machine) με λειτουργικό σύστημα Ubuntu 16.0.4, που είναι μια διανομή των Linux.

Ανέβασμα του Bootloader

Για την ενσωμάτωση του Bootloader, από περιβάλλον Windows, χρησιμοποιώντας την συντομογραφία του μικροελεγκτή που είναι η “m328”, το πρόγραμμα WinAVR δεν την αναγνώριζε. Για να την αναγνωρίσει πρέπει να προσθέσουμε την ταυτότητα του μικροελεγκτή στο πρόγραμμα WinAVR. Για να το κάνουμε αυτό πρέπει:

- Να ανοίξουμε το αρχείο “avrdude.conf” (συνήθως βρίσκεται στη διαδρομή: C:\WinAVR-20100110\bin).
- Να αφαιρέσουμε το γράμμα “p” από τις γραμμές : 8544, 8548, 8549.

- Να αλλάξουμε την ταυτότητα του μικροελεγκτή στη σειρά 8557. Η ταυτότητα του Atmega328 είναι “0x1e 0x95 0x14”.

Επίσης παρατηρούμε ότι όταν ανεβάζαμε το πρόγραμμα στο μικροελεγκτή χρησιμοποιήσαμε την ταυτότητα του “m328p” αντί για την “m328” που χρειάστηκε για το ανέβασμα του Bootloader.

KiloGUI στα Windows

Το πρόγραμμα που ελέγχει τη λειτουργία των Kilobot (το KiloGUI) δεν ανιχνεύει τον OHC σε περιβάλλοντα Windows. Αφού εγκαταστήσαμε όλους του οδηγούς (drivers) για την πλακέτα το πρόβλημα εξακολουθούσε να υφίσταται. Η λύση βρέθηκε όταν δοκιμάσαμε το πρόγραμμα από περιβάλλον Linux, που λειτουργήσε κανονικά.

4.10 Μελλοντικά σχέδια

Έχοντας ένα ολοκληρωμένο σύστημα ελέγχου για τα Kilobot, θα θέλαμε να πραγματοποιήσουμε ένα πείραμα συλλογικής συμπεριφοράς στο σκοτάδι για να μάθουμε παραπάνω για την συλλογική συμπεριφορά, να ελέγξουμε την αποτελεσματικότητα του νέου συστήματος και νέα πιθανά προβλήματα.

Διαδικασία του πειράματος:

1. Τυχαία τοποθέτηση 200 βαθμονομημένων Kilobot στην επιφάνεια.
2. Εγγραφή και compile του κώδικα που θα περιγράφει μία συλλογική συμπεριφορά.
3. Ανέβασμα του κώδικα.
4. Παρατήρηση του πειράματος και καταγραφή αποτελεσμάτων.

Παράρτημα 1 - Φωτογραφίες από το ανέβασμα του Bootloader

Εκτέλεση της εντολής “`sudo avrdude -b 57600 -c usbtiny -p m328 -v -e -U efuse:w:0xFF:m -U hfuse:w:0xDA:m -U lfuse:w:0xFF:m`” για τον προγραμματισμό των ασφαλειών του μικροελεγκτή πριν δεχτεί το πρόγραμμα του Bootloader.

```
@ -U: ~
-U:~$ sudo avrdude -b 57600 -c usbtiny -p m328 -v -e -U efuse:w:0xFF:m -U hfuse:w:0xDA:m -U lfuse:w:0xFF:m
[sudo] password for :
avrdude: Version 6.2
Copyright (c) 2000-2005 Brian Dean, http://www.bdmicro.com/
Copyright (c) 2007-2014 Joerg Wunsch

System wide configuration file is "/etc/avrdude.conf"
User configuration file is "/home/./avrduderc"
User configuration file does not exist or is not a regular file, skipping

Using Port                : usb
Using Programmer           : usbtiny
Overriding Baud Rate       : 57600
avrdude: usbdev_open(): Found USBtinyISP, bus:device: 001:002
AVR Part                   : ATmega328
Chip Erase delay           : 9000 us
PAGEL                      : PD7
BS2                        : PC2
RESET disposition         : dedicated
RETRY pulse               : SCK
serial program mode       : yes
parallel program mode     : yes
Timeout                   : 200
StabDelay                 : 100
CmdexeDelay               : 25
SyncLoops                 : 32
ByteDelay                 : 0
PollIndex                 : 3
PollValue                 : 0x53
Memory Detail              :

                Block Poll
Memory Type Mode Delay Size  Indx Paged  Size  Page
-----
eeprom      65   20    4    0 no    1024   4
flash      65    6  128    0 yes   32768 128
lfuse       0    0    0    0 no     1    0
hfuse       0    0    0    0 no     1    0
efuse       0    0    0    0 no     1    0
lock        0    0    0    0 no     1    0
calibration 0    0    0    0 no     1    0
signature   0    0    0    0 no     3    0
                Size #Pages MinW  MaxW      ReadBack
-----
eeprom      0 3600 3600 0xff 0xff
flash      256 4500 4500 0xff 0xff
lfuse       0 4500 4500 0x00 0x00
hfuse       0 4500 4500 0x00 0x00
efuse       0 4500 4500 0x00 0x00
lock        0 4500 4500 0x00 0x00
calibration 0    0    0 0x00 0x00
signature   0    0    0 0x00 0x00

Programmer Type : USBtiny
Description      : USBtiny simple USB programmer, http://www.ladyada.net/make/usbtin
yisp/
avrdude: programmer operation not supported

avrdude: Using SCK period of 10 usec
avrdude: AVR device initialized and ready to accept instructions

Reading | ##### | 100% 0.00s

avrdude: Device signature = 0x1e9514 (probably m328)
avrdude: safemode: hfuse reads as D9
avrdude: safemode: efuse reads as FF
avrdude: erasing chip
avrdude: Using SCK period of 10 usec
```

Ολοκλήρωση της προηγούμενης εντολής για τον προγραμματισμό των ασφαλειών. Όπως βλέπετε στο αποτέλεσμα “avrdude: safemode: Fuses OK (E:FD, H:DA, L:FF)” οι ασφάλειες τέθηκαν ως: efuse = FD, hfuse = DA, lfuse = FF.

```

Reading | ##### | 100% 0.00s

avrdude: Device signature = 0x1e9514 (probably m328)
avrdude: safemode: hfuse reads as D9
avrdude: safemode: efuse reads as FF
avrdude: erasing chip
avrdude: Using SCK period of 10 usec
avrdude: reading input file "0xFD"
avrdude: writing efuse (1 bytes):

Writing | ##### | 100% 0.01s

avrdude: 1 bytes of efuse written
avrdude: verifying efuse memory against 0xFD:
avrdude: load data efuse data from input file 0xFD:
avrdude: input file 0xFD contains 1 bytes
avrdude: reading on-chip efuse data:

Reading | ##### | 100% 0.00s

avrdude: verifying ...
avrdude: 1 bytes of efuse verified
avrdude: reading input file "0xDA"
avrdude: writing hfuse (1 bytes):

Writing | ##### | 100% 0.01s

avrdude: 1 bytes of hfuse written
avrdude: verifying hfuse memory against 0xDA:
avrdude: load data hfuse data from input file 0xDA:
avrdude: input file 0xDA contains 1 bytes
avrdude: reading on-chip hfuse data:

Reading | ##### | 100% 0.00s

avrdude: verifying ...
avrdude: 1 bytes of hfuse verified
avrdude: reading input file "0xFF"
avrdude: writing lfuse (1 bytes):

Writing | ##### | 100% 0.00s

avrdude: 1 bytes of lfuse written
avrdude: verifying lfuse memory against 0xFF:
avrdude: load data lfuse data from input file 0xFF:
avrdude: input file 0xFF contains 1 bytes
avrdude: reading on-chip lfuse data:

Reading | ##### | 100% 0.00s

avrdude: verifying ...
avrdude: 1 bytes of lfuse verified

avrdude: safemode: hfuse reads as DA
avrdude: safemode: efuse reads as FD
avrdude: safemode: Fuses OK (E:FD, H:DA, L:FF)

avrdude done. Thank you.

-U:~$
```

Εκτέλεση εντολής “`sudo avrdude -b 57600 -c usbtiny -p m328 -v -e -U flash:w:ATmegaBOOT_168_atmega328.hex -U lock:w:0x0F:m`” για το ανέβασμα και κλείδωμα του κώδικα του Bootloader στη μνήμη του μικροελεγκτή. Όπως βλέπετε, το πρώτο κομμάτι της εκτέλεσης αυτής της εντολής είναι ίδιο με αυτό της προηγούμενης εντολής. Αυτό συμβαίνει, γιατί το πρώτο βήμα για την εκτέλεση μια εντολής είναι η αναγνώριση του μικροελεγκτή και του προγραμματιστή, ώστε να δοθούν οι κατάλληλες ταυτότητες από το πρόγραμμα (WinAVR) και να γίνει ο έλεγχος για το αν υπάρχει επικοινωνία με τον μικροελεγκτή.

```

-U: ~
-U: ~/Desktop$ sudo avrdude -b 57600 -c usbtiny -p m328 -v -e -U flash:w:ATmegaBOOT_168_atmega328
.hex -U lock:w:0x0F:m
[sudo] password for :
avrdude: Version 6.2
Copyright (c) 2000-2005 Brian Dean, http://www.bdmicro.com/
Copyright (c) 2007-2014 Joerg Wunsch

System wide configuration file is "/etc/avrdude.conf"
User configuration file is "/home/~/avrduderc"
User configuration file does not exist or is not a regular file, skipping

Using Port                : usb
Using Programmer           : usbtiny
Overriding Baud Rate      : 57600
avrdude: usbdev_open(): Found USBtinyISP, bus:device: 001:002
AVR Part                   : ATmega328
Chip Erase delay           : 9000 us
PAGEL                      : PD7
BS2                        : PC2
RESET disposition         : dedicated
RETRY pulse               : SCK
serial program mode       : yes
parallel program mode     : yes
Timeout                   : 200
StabDelay                 : 100
CmdexeDelay               : 25
SyncLoops                 : 32
ByteDelay                 : 0
PollIndex                 : 3
PollValue                 : 0x53
Memory Detail

      Block Poll
Memory Type Mode Delay Size  Indx Paged  Size   Page
-----
EEPROM      65   20    4    0 no    1024    4
Flash      65    6  128    0 yes   32768  128
lfuse       0    0    0    0 no     1    0
hfuse       0    0    0    0 no     1    0
efuse       0    0    0    0 no     1    0
lock        0    0    0    0 no     1    0
calibration 0    0    0    0 no     1    0
signature   0    0    0    0 no     3    0
      #Pages MinW  MaxW      ReadBack
-----
EEPROM      0 3600  3600 0xff 0xff
Flash      256 4500  4500 0xff 0xff
lfuse       0 4500  4500 0x00 0x00
hfuse       0 4500  4500 0x00 0x00
efuse       0 4500  4500 0x00 0x00
lock        0 4500  4500 0x00 0x00
calibration 0    0    0 0x00 0x00
signature   0    0    0 0x00 0x00

Programmer Type : USBtiny
Description      : USBtiny simple USB programmer, http://www.ladyada.net/make/usbtin
yisp/
avrdude: programmer operation not supported

avrdude: Using SCK period of 10 usec
avrdude: AVR device initialized and ready to accept instructions

Reading | ##### | 100% 0.00s

avrdude: Device signature = 0x1e9514 (probably m328)
avrdude: safemode: hfuse reads as D9
avrdude: safemode: efuse reads as FF
avrdude: erasing chip
avrdude: Using SCK period of 10 usec

```

Ολοκλήρωση της εντολής για το ανέβασμα του Bootloader στη μνήμη του μικροελεγκτή.

```
Reading | ##### | 100% 0.00s
avrdude: Device signature = 0x1e9514 (probably m328)
avrdude: safemode: hfuse reads as DA
avrdude: safemode: efuse reads as FD
avrdude: erasing chip
avrdude: Using SCK period of 10 usec
avrdude: reading input file "ATmegaBOOT_168_atmega328.hex"
avrdude: input file ATmegaBOOT_168_atmega328.hex auto detected as Intel Hex
avrdude: writing flash (32670 bytes):

Writing | ##### | 100% 0.00s
avrdude: 32670 bytes of flash written
avrdude: verifying flash memory against ATmegaBOOT_168_atmega328.hex:
avrdude: load data flash data from input file ATmegaBOOT_168_atmega328.hex:
avrdude: input file ATmegaBOOT_168_atmega328.hex auto detected as Intel Hex
avrdude: input file ATmegaBOOT_168_atmega328.hex contains 32670 bytes
avrdude: reading on-chip flash data:

Reading | ##### | 100% 0.00s
avrdude: verifying ...
avrdude: 32670 bytes of flash verified
avrdude: reading input file "0x0F"
avrdude: writing lock (1 bytes):

Writing | ##### | 100% 0.01s
avrdude: 1 bytes of lock written
avrdude: verifying lock memory against 0x0F:
avrdude: load data lock data from input file 0x0F:
avrdude: input file 0x0F contains 1 bytes
avrdude: reading on-chip lock data:

Reading | ##### | 100% 0.00s
avrdude: verifying ...
avrdude: 1 bytes of lock verified

avrdude: safemode: hfuse reads as DA
avrdude: safemode: efuse reads as FD
```

```
~ -u:~$ █
```

Παράρτημα 2 – Επεξήγηση όρων

Arduino pro mini – Είναι μία πλακέτα που έχει το μικροελεγκτή Atmega328 και μπορεί να προγραμματιστεί ώστε να εκτελέσει διάφορες λειτουργίες. Χαρακτηριστικό αυτής της πλακέτας είναι ο Bootloader που είναι της εταιρίας Arduino.

AVR – Είναι μια σειρά μικροελεγκτών της εταιρίας Atmel.

ASCII – Συντομογραφία του αμερικάνικου στάνταρ κώδικα για ανταλλαγή πληροφοριών (American Standard code for Information Interchange).

Bit – Βασική μονάδα πληροφορίας στους υπολογιστές και στις ψηφιακές συσκευές. Μπορεί να πάρει μόνο 2 τιμές και άρα μπορεί να αναπαρασταθεί από οποιαδήποτε συσκευή 2 καταστάσεων (π.χ. ένα διακόπτη).

Compiler – Είναι ένα πρόγραμμα των υπολογιστών (ή ένα σύνολο προγραμμάτων) που μετατρέπει πηγαίο κώδικα γραμμένο σε μία προγραμματιστική γλώσσα σε μία άλλη γλώσσα υπολογιστή. Με το αποτέλεσμα συνήθως να είναι σε δυαδική μορφή.

Drone – Ιπτάμενη συσκευή η οποία μπορεί να πετάει αυτόνομα ή κάτω από τον έλεγχο κάποιου ανθρώπου.

Dropbox – Υπηρεσία του διαδικτύου για την αποθήκευση και τον συγχρονισμό αρχείων στο “cloud” (δηλαδή αποθηκεύονται στο διαδίκτυο και είναι πάντα διαθέσιμα).

GitHub – Υπηρεσία του διαδικτύου για αποθήκευση και τον χειρισμό προγραμματιστικού κώδικα. Χρησιμοποιείται κυρίως από προγραμματιστές ώστε να έχουν εύκολη πρόσβαση του κώδικά τους από κάθε συσκευή και να μπορούν άλλοι χρήστες να συμβάλουν στην βελτίωση του κώδικα.

ISP και ICSP – In System Programming και In Circuit Serial Programming είναι δύο έννοιες που χρησιμοποιούνται για συσκευές που μπορούν να προγραμματιστούν επάνω στο τελικό προϊόν (πλακέτα) χωρίς να χρειάζεται αφαίρεση του μικροελεγκτή για τον προγραμματισμό του.

Linux – Λειτουργικό σύστημα το οποίο είναι ανοιχτού κώδικα και μπορεί ο καθένας να συμβάλει σε αυτό.

MOSFET (Metal-Oxide-Semiconductor Field-Effect-Transistor) – Είδος τρανζίστορ επίδρασης πεδίου (FET). Έχει μία μονωμένη πύλη, της οποίας η τάση ελέγχει την αγωγιμότητα της συσκευής. [Wikipedia]

Open-Source (Ανοιχτού κώδικα) – Ονομάζεται κάθε λογισμικό υπολογιστή, με τον πηγαίο κώδικά του, που είναι διαθέσιμο με μία άδεια που επιτρέπει στο χρήστη να το μελετήσει, να κάνει αλλαγές και να το διανέμει στον καθένα και για κάθε σκοπό.

Sleep-mode – Μία κατάσταση χαμηλής κατανάλωσης για ηλεκτρονικές συσκευές, όπως υπολογιστές, τηλεοράσεις και τηλεχειριστήρια στην οποία μέρη της συσκευής απενεργοποιούνται μέχρι να ζητηθούν από το χρήστη.

Μνήμη EEPROM (Electrically Erasable Programmable Read Only Memory) – Μη-πτητική μνήμη που χρησιμοποιείται στους υπολογιστές και άλλες ηλεκτρονικές συσκευές για να

αποθηκεύει σχετικά μικρή ποσότητα δεδομένων, αλλά επιτρέπει συγκεκριμένα μέρη να διαγράφονται και να ξαναπρογραμματίζονται.

Μνήμη Flash – Είδος μνήμης που διατηρεί τα δεδομένα της ακόμα και στην απουσία τροφοδοσίας.

Μνήμη ROM (Read Only Memory) – Είδος μνήμης που όταν προγραμματιστεί επιτρέπει μόνο την ανάγνωση των δεδομένων που έχει αποθηκευμένα και τα διατηρεί ακόμα και στην απουσία τροφοδοσίας.

Μνήμη SDRAM (Synchronous Dynamic Random-Access Memory) – Γενικό όνομα διαφόρων ειδών δυναμικής μνήμης τυχαίας προσπέλασης, που συγχρονίζεται με την ταχύτητα του επεξεργαστή που έχει σχεδιαστεί. [Wikipedia]

Τερματικό – Ηλεκτρονική ή ηλεκτρομηχανική συσκευή για τη εισαγωγή και την εμφάνιση δεδομένων από και προς ένα ή περισσότερα υπολογιστικά συστήματα. [Wikipedia]

Τεχνητή νοημοσύνη – Κλάδος της πληροφορικής που ασχολείται με τη σχεδίαση και την υλοποίηση υπολογιστικών συστημάτων που μιμούνται στοιχεία της ανθρώπινης συμπεριφοράς τα οποία υπονοούν έστω και στοιχειώδη ευφυΐα: μάθηση, προσαρμοστικότητα, εξαγωγή συμπερασμάτων, κατανόηση από συμφραζόμενα, επίλυση προβλημάτων κλπ. [Wikipedia]

Βιβλιογραφία

- [1] E. Nikolaidis, C. Sabo, J. A. R. Marshal και A. Reina, «Characterisation and upgrade of the communication between overhead controllers and Kilobots», White Rose Research Online, April 2017.
- [2] T. Telegraph, «Ο υπερ-υπολογιστής της Κίνας», 2013. [Ιστοσελίδα]. Διαθέσιμη: <http://www.telegraph.co.uk/technology/news/10129285/Chinese-supercomputer-is-worlds-fastest-at-33860-trillion-calculations-per-second.html>.
- [3] T. N. Interest, «Ο υπερ-υπολογιστής της Κίνας», 2016. [Ιστοσελίδα]. Διαθέσιμη: <http://nationalinterest.org/blog/the-buzz/chinas-new-super-computer-can-perform-93000-trillion-16660>.
- [4] «Ρομπότ για ηλεκτροκόλληση», [Ιστοσελίδα]. Διαθέσιμη: <https://www.robots.com/applications/arc-welding>.
- [5] «Βιομηχανικό ρομπότ για συναρμολόγηση προϊόντων», Acieta, [Ιστοσελίδα]. Διαθέσιμη: <http://www.acieta.com/automation-application/assembly-robotics/>.
- [6] «Αυτόματο ρομπότ για βαφή προϊόντων», RobotWorx, [Ιστοσελίδα]. Διαθέσιμη: <https://www.robots.com/applications/painting-automation>.
- [7] «Ρομπότ "CarryPick" για τη μετακίνηση και ταξινόμηση αντικειμένων σε αποθήκες», [Ιστοσελίδα]. Διαθέσιμη: <http://www.swisslog.com/carrypick>.
- [8] «Εκπαιδευτικά ρομπότ "Lego"», Lego, [Ιστοσελίδα]. Διαθέσιμη: <https://www.lego.com/en-us/mindstorms>.
- [9] «Εκπαιδευτικό ρομπότ "e-puck"», [Ιστοσελίδα]. Διαθέσιμη: <http://www.e-puck.org/>.
- [10] «Ρομπότ "Roomba" για σκούπισμα εσωτερικών χώρων», iRobot, [Ιστοσελίδα]. Διαθέσιμη: <http://www.irobot.com/For-the-Home/Vacuuming/Roomba.aspx>.
- [11] «Οικιακό ρομπότ για μαγείρεμα», Moley Robotics, [Ιστοσελίδα]. Διαθέσιμη: <http://www.moley.com/>.
- [12] «Προσωπικός βοηθός "Care-O-bot"», Fraunhofer IPA, [Ιστοσελίδα]. Διαθέσιμη: <http://www.care-o-bot.de/en/care-o-bot-4.html>.
- [13] «Θεραπευτικό ρομπότ "PARO"», [Ιστοσελίδα]. Διαθέσιμη: <http://www.parorobots.com/index.asp>.
- [14] «Διαδραστικό ρομπότ "Zeno"», Hanson Robotics, [Ιστοσελίδα]. Διαθέσιμη: <http://www.hansonrobotics.com/robot/zeno/>.
- [15] «Ανθρωποειδές ρομπότ "Pepper" που μπορεί να ανιχνεύει συναισθήματα», [Ιστοσελίδα]. Διαθέσιμη: <https://www.ald.softbankrobotics.com/en/cool-robots/pepper>.
- [16] D. J. T. Sumpter, Collective Animal Behavior, Princeton University Press, 2010.

- [17] A. Reina, A. J. Cope, E. Nikolaidis, J. A. R. Marshall και C. Sabo, «ARK: Augmented Reality for Kilobots», *IEEE Robotics and Automation Letters*, p. in press, 2017.
- [18] M. Rubenstein, C. Ahler και R. Nagpal, «Kilobot: A Low Cost Scalable Robot System for Collective Behaviors», *2012 IEEE International Conference on Robotics and Automation*, p. 3293–3298.
- [19] A. Antoun, G. Valentini, E. Hocquard, B. Wiandt, V. Trianni και M. Dorigo, «Kilogrid: A modular virtualization environment for the Kilobot robot», *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 3809-3814, 2016.
- [20] «The Kilobot», K-Team, [Ιστοσελίδα]. Διαθέσιμη: <http://www.k-team.com/mobile-robotics-products/kilobot#introduction>.
- [21] «Πληροφορίες για τα Kilobot και συνδέσμοι για κατέβασμα των σχηματικών και του λογισμικού», [Ιστοσελίδα]. Διαθέσιμη: <https://www.eecs.harvard.edu/ssr/projects/progSA/kilobot.html#make>.
- [22] «Προγραμματιστής μικροελεγκτών AVR: Pocket AVR Programmer», Sparkfun, [Ιστοσελίδα]. Διαθέσιμη: <https://www.sparkfun.com/products/9825>.
- [23] «Πρόγραμμα WinAVR για compile και ανέβασμα προγραμμάτων σε μικροελεγκτές της Atmel», [Ιστοσελίδα]. Διαθέσιμη: <https://sourceforge.net/projects/winavr>.
- [24] «Αρχείο Bootloader για το Arduino pro mini», Arduino, [Ιστοσελίδα]. Διαθέσιμη: <https://github.com/arduino/Arduino/tree/master/hardware/arduino/avr/bootloaders/atmega>.
- [25] A. Cornejo, «Πρόγραμμα ανοιχτού κώδικα και βιβλιοθήκες για τον OHC και τα Kilobot», [Ιστοσελίδα]. Διαθέσιμη: <https://github.com/acornejo/kilolib>.
- [26] «Θεμελιώδεις μονάδες μέτρησης του Διεθνούς Συστήματος Μονάδων», Wikipedia, [Ιστοσελίδα]. Διαθέσιμη: https://el.wikipedia.org/wiki/Θεμελιώδεις_μονάδες_μέτρησης_του_Διεθνούς_Συστήματος_Μονάδων.
- [27] «Γραφικό περιβάλλον KiloGUI για τον έλεγχο και τον προγραμματισμό των Kilobot», [Ιστοσελίδα]. Διαθέσιμη: <https://www.kilobotics.com/download>.
- [28] «Εταιρία διανομής του λογισμικού V-Rep για προσωμοίωση ρομποτικών συστημάτων», Coppelia Robotics, [Ιστοσελίδα]. Διαθέσιμη: <http://www.coppeliarobotics.com/>.
- [29] «Κατέβασμα του λογισμικού V-Rep», [Ιστοσελίδα]. Διαθέσιμη: <http://www.coppeliarobotics.com/downloads.html>.
- [30] «Υλικό για τον προγραμματισμό των Kilobot», [Ιστοσελίδα]. Διαθέσιμη: <http://www.k-team.com/mobile-robotics-products/kilobot#manual>.
- [31] «Ιστοσελίδα για το κατέβασμα του λογισμικού Eclipse», [Ιστοσελίδα]. Διαθέσιμη: <https://eclipse.org/downloads>.

- [32] «Διαδικτυακή πλατφόρμα για compile του κώδικα για τα Kilobot», [Ιστοσελίδα]. Διαθέσιμη: <https://www.kilobotics.com/editor>.
- [33] «Ιστότοπος Dropbox», [Ιστοσελίδα]. Διαθέσιμη: www.dropbox.com.
- [34] «Ιστότοπος της Sheffield Robotics», [Ιστοσελίδα]. Διαθέσιμη: <http://www.sheffieldrobotics.ac.uk/>.
- [35] «Γραφικό περιβάλλον για το Arduino (IDE)», [Ιστοσελίδα]. Διαθέσιμη: <https://www.arduino.cc/en/main/software>.
- [36] «Πληροφορίες για τις βιβλιοθήκες των Kilobot», [Ιστοσελίδα]. Διαθέσιμη: <https://www.kilobotics.com/docs/index.html>.
- [37] «Πρότυπα προγράμματα για τον προγραμματισμό των Kilobot», [Ιστοσελίδα]. Διαθέσιμη: <https://www.kilobotics.com/labs>.
- [38] «Ανέβασμα του Bootloader σε ένα Arduino Mini», Arduino, [Ιστοσελίδα]. Διαθέσιμη: <https://www.arduino.cc/en/Hacking/MiniBootloader>.
- [39] «Φύλλο δεδομένων φωτοδιόδου πυριτίου PIN TEMD7100X01», Vishay Semiconductors, [Ιστοσελίδα]. Διαθέσιμη: <http://www.vishay.com/docs/81960/temd7100.pdf>.
- [40] «Ιστότοπος της εταιρίας K-Team που διανέμει τα Kilobot», [Ιστοσελίδα]. Διαθέσιμη: <http://www.k-team.com>.
- [41] «Πληροφορίες για τη λειτουργία του συστήματος των Kilobot», [Ιστοσελίδα]. Διαθέσιμη: <https://www.kilobotics.com/documentation>.
- [42] «Διαδραστικό ρομπότ για συντροφιά "NAO"», [Ιστοσελίδα]. Διαθέσιμη: <https://www.ald.softbankrobotics.com/en/cool-robots/nao>.
- [43] «Φωτογραφία φωλιάς μυρμηγκιών», [Ιστοσελίδα]. Διαθέσιμη: <http://www.terro.com/articles/how-to-find-an-ant-nest>.
- [44] «Φωτογραφία κυκλοφοριακής συμφόρησης από την Ισπανία», [Ιστοσελίδα]. Διαθέσιμη: <http://www.greenguidespain.com/andalucia/2015/11/spanish-drivers-spending-hours-and-billions-in-traffic-jams/>.
- [45] «Ρομπότ "marXbot" για μελέτη της συλλογικής συμπεριφοράς», [Ιστοσελίδα]. Διαθέσιμη: <http://mobots.epfl.ch/marxbot.html>.
- [46] D. J. T. Sumpter, The principles of collective animal behaviour, τόμ. 361, The Royal Society, 2006.