

Τεχνολογικό Εκπαιδευτικό Ίδρυμα Κρήτης

Σχολή Τεχνολογικών Εφαρμογών
Τμήμα Μηχανικών Πληροφορικής



ΤΕΙ Κρήτης
Τεχνολογικό Εκπαιδευτικό Ίδρυμα Κρήτης

Πτυχιακή Εργασία

Κατασκευή ρομποτικού οχήματος με δυνατότητα προγραμματισμού μέσω του περιβάλλοντος scratch.

Καρυωτάκης Βασίλειος 3393

Επιβλέπων καθηγητής: Παναγιωτάκης Σπυρίδων

Ευχαριστίες:

Θα ήθελα να ευχαριστήσω τον επιβλέποντα καθηγητή μου, κύριο Παναγιωτάκης Σπυρίδων, που μου έδωσε την δυνατότητα να ασχοληθώ με ένα τόσο αξιόλογο και ενδιαφέρον θέμα επίσης θα ήθελα να ευχαριστήσω την οικογένειά μου που με στήριξαν με κάθε τρόπο καθ' όλη την διάρκεια των σπουδών μου.

Abstract:

In this thesis we will deal with the construction, implementation and programming of a robotic vehicle based on the Arduino open platform for the introduction of primary / secondary school students into the world of programming, this will be achieved by programming in a Scratch environment that does not need to have programming skills to program.

The end-user at the end of the implementation will be able to control the vehicle through virtual components rather than code through the computer as well as be able to control via Bluetooth, via IR remote control, the vehicle will be autonomous in the room moving autonomously and avoid obstacles without the user's intervention and also be able to recognize a line and follow it again autonomously.

Σύνοψη:

Στην παρούσα πτυχιακή θα ασχοληθούμε με την κατασκευή , την υλοποίηση και τον προγραμματισμό ενός ρομποτικού οχήματος το οποίο είναι βασισμένο στην ανοικτή πλατφόρμα Arduino για την εισαγωγή μαθητών Δημοτικού/Γυμνασίου στον κόσμο του προγραμματισμού, αυτό θα επιτευχθεί με τον προγραμματισμό σε περιβάλλον Scratch το οποίο δεν χρειάζεται να έχει κάποιος γνώσεις προγραμματισμού για να προγραμματίσει.

Ο τελικός χρήστης στο τέλος της υλοποίησης θα μπορεί να ελέγχει το όχημα μέσω εικονικών στοιχείων και όχι κώδικα δια μέσου του ηλεκτρονικού υπολογιστή όπως επίσης θα μπορεί να το ελέγξει μέσω Bluetooth , μέσω IR (τηλεκοντρόλ), το όχημα να είναι αυτόνομο στον χώρο δηλαδή να μπορεί να κινείται αυτόνομα και να αποφεύγει εμπόδια χωρίς την παρέμβαση του χρήστη και επίσης να μπορεί να αναγνωρίζει μια διαδρομή (Line Follow) και να την ακολουθεί πάλι αυτόνομα.

Εισαγωγή:

Στο παρόν θέμα πτυχιακής εργασίας θα εξετάσουμε την συνδρομή της STEM στην εκπαίδευση για την εκμάθηση προγραμματισμού σε μικρές ηλικίες, θα ερευνήσουμε διάφορους μεθόδους block-based προγραμματισμού θα μιλήσουμε για τα μειονεκτήματα και πλεονεκτήματα της κάθε μια από αυτές, επίσης θα δούμε διάφορες υλοποιήσεις για το ρομπότ και πως καταλήξαμε στο τελικό σχέδιο και γιατί.

Θα αναφερθούμε επίσης στα διάφορα λογισμικά που υποστηρίζουν τον block-based προγραμματισμό και τους λόγους που καταλήξαμε σε αυτό που χρησιμοποιήσαμε όπως επίσης για τις δυσκολίες που προέκυψαν κατά την έρευνα και την τελική υλοποίηση.

Θα μιλήσουμε εκτενώς για τον τρόπο που προγραμματίσαμε το ρομπότ να κάνει τις εργασίες για τις οποίες μιλήσαμε παραπάνω όπως επίσης και για την κατασκευή του και τον τρόπο λειτουργίας του.

Σαν τελικό αποτέλεσμα θα έχουμε ένα πλήρως λειτουργικό ρομπότ το οποίο θα είναι πολύ εύκολα προγραμματίσιμο μέσω Block-based προγραμματισμού όπως επίσης και την πλήρη έννοια των STEM και πως βοηθάει στην εκπαίδευση και στην εκμάθηση.

Τα κεφάλαια που χωρίζεται η πτυχιακή είναι τα εξής:

Κεφάλαιο 1:

Αναλύουμε την μέθοδο STEM εξετάζουμε την χρησιμότητα της και τα πλεονεκτήματα της στην εκπαίδευση, επίσης συγκρίνουμε τον block-based με τον text-based προγραμματισμό και αναφερόμαστε στο scratch και τον τρόπο λειτουργίας του.

Κεφάλαιο 2:

Μιλάμε για το Snap4Arduino, το λογισμικό που χρησιμοποιούμε στην παρούσα πτυχιακή την εξέταση του γραφικού περιβάλλοντος του, όπως επίσης για το πρωτόκολλο επικοινωνίας που δουλέψαμε και τον τρόπο μεταφόρτωσης του στο ρομπότ.

Κεφάλαιο 3:

Αναλύουμε το κατασκευαστικό κομμάτι της πτυχιακής, δηλαδή του ρομπότ, τους λόγους για τους οποίους καταλήξαμε στο τελικό σχέδιο και τις διάφορες υλοποιήσεις που θα μπορούσαν να γίνουν.

Επίσης αναλύουμε τις λειτουργίες που αποφασίσαμε να έχει το όχημα και πως θα δουλεύουν αυτές τις λειτουργίες όσο αφορά το κατασκευαστικό κομμάτι και τα εξαρτήματα που χρησιμοποιήσαμε.

Κεφάλαιο 4:

Σε αυτό το κεφάλαιο δείχνουμε πως κατασκευάζουμε το ρομπότ, που θα μπουν τα εξαρτήματα που διαλέξαμε παραπάνω, επίσης αναλύουμε το κάθε εξάρτημα με πιο τρόπο δουλεύει για να επιτευχθεί η λειτουργία που θέλουμε να προσθέσουμε στο όχημα

Κεφάλαιο 5:

Σε αυτό το κεφάλαιο δείχνουμε και αναλύουμε διεξοδικά τον κώδικα που φτιάξαμε για να επιτύχουμε τις λειτουργίες που προαναφέραμε.

Κεφάλαιο 6:

Αναλύουμε το τρόπο λειτουργίας του ρομπότ, πως συμπεριφέρεται δηλαδή στις λειτουργίες που του έχουμε προγραμματίσει να κάνει.

Κεφάλαιο 7:

Αναλύουμε τα συμπεράσματα που προσκομίσαμε στην πάροδο της πτυχιακής τα θετικά και τα αρνητικά όπως επίσης αναφερόμαστε σε μελλοντικές επεκτάσεις.

Περιεχόμενα

Ευχαριστίες:	2
Abstract:	3
Σύνοψη:	4
Εισαγωγή:	5
Κεφάλαιο 1: Η σημασία της ρομποτικής και των ΣΤΕΜ στην εκπαίδευση.	9
1.1 Τι είναι το ΣΤΕΜ	9
1.2 Εκπαιδευτική Ρομποτική.....	9
1.3 Υπολογιστική Σκέψη.....	10
1.4 Εκπαιδευτική Ρομποτική & Υπολογιστική Σκέψη.....	11
1.5 Μια μικρή σύγκριση μεταξύ visual-based και text-based προγραμματισμού:	13
1.6 Λογισμικά που υποστηρίζουν block based προγραμματισμό.	16
Κεφάλαιο 2 Snap4Arduino	21
2.1 Τι είναι το Snap4Arduino:	21
2.2 Το Interface:	21
2.3 Η λογική της λειτουργικότητας πίσω από το Snap.	27
2.3.1 Ανέβασμα πρωτοκόλλου:.....	28
Κεφάλαιο 3: Σχεδιασμός ρομποτικού οχήματος.	30
3.1 Το σασί	30
3.2 Λειτουργίες του οχήματος	32
3.2.1 Αποφυγή εμποδίων.....	32
3.2.2 Ακολούθηση μαύρης γραμμής.....	33
3.2.3 Τηλεχειρισμός είτε από υπολογιστή είτε από τηλεκοντρόλ.....	34
Κεφάλαιο 4: Η υλοποίηση της κατασκευής.....	36
4.1 Αρχικές ρυθμίσεις	36
4.1.1 Αισθητήρας υπερέυθρων Keyes KY-033 IR Line tracer.....	36
4.2 Η-ΓΕΦΥΡΑ L298.....	37
4.3 Η πλακέτα Arduino Uno R3	39
4.4 Arduino Sensor Shield v5.0.....	40
4.5 Αισθητήρας Υπερήχων (Ultrasonic Module HC-SR04)	41
4.6 Σερβοκινητήρας (Servo SG90).....	43
4.7 Ir Receiver και Ir Transmitter	44
Κεφάλαιο 5: Προγραμματίζοντας τις λειτουργίες του οχήματος.....	46
Σενάριο 1: Αυτοκινούμενο όχημα στο χώρο με αποφυγή εμποδίων.....	46

Πτυχιακή Εργασία Τμήματος Μηχανικών Πληροφορικής Καρυωτάκης Βασίλειος

Σενάριο 2: Λειτουργία παρακολούθησης μαύρης γραμμής.....	52
Σενάριο 3: Έλεγχος με χειριστήριο υπερύθρων.....	56
Σενάριο 4: Έλεγχος από υπολογιστή.....	66
Κεφάλαιο 6: Οι λειτουργίες του οχήματος αναλυτικά.....	71
Σενάριο 1: Αυτοκινούμενο όχημα στον χώρο με αποφυγή εμποδίων.....	71
Σενάριο 2: Λειτουργία παρακολούθησης μαύρης γραμμής.....	74
Κεφάλαιο 7: Συμπεράσματα:.....	78
7.1 Γενικά:.....	78
7.2 Συνδυάζοντας παιχνίδι και μάθηση:.....	78
7.3 Μελλοντικές επεκτάσεις:.....	79
Βιβλιογραφία.....	82

Κεφάλαιο 1: Η σημασία της ρομποτικής και των ΣΤΕΜ στην εκπαίδευση.

1.1 Τι είναι το ΣΤΕΜ

Ο όρος “**STEM**” (**S**cience-**T**echnology-**E**ngineering and **M**athematics) είναι το ακρωνύμιο που χρησιμοποιείται από άτομα που σχετίζονται με την εκπαίδευση για τα πεδία που σχετίζονται με την τεχνολογία, τα μαθηματικά, τις φυσικές επιστήμες και την επιστήμη των μηχανικών. Ο όρος “STEM” εμφανίστηκε για πρώτη φορά το 2001 από την βιολόγο Judith Ramaley που ως διευθύντρια στις Ηνωμένες Πολιτείες στο ίδρυμα φυσικών επιστημών ήταν υπεύθυνη για την ανάπτυξη νέων προγραμμάτων σπουδών. Το “STEM” είναι μια σύγκλιση στην εκπαίδευση η οποία σχεδιάστηκε ώστε στην διδασκαλία των φυσικών επιστημών και των μαθηματικών οι οποίες είναι ζωτικής σημασίας για την βασική κατανόηση του σύμπαντος, να εισαχθούν η επιστήμη των μηχανικών και οι τεχνολογίες, οι οποίες αποτελούν τα μέσα αλληλεπίδρασης με το σύμπαν για τον άνθρωπο. Είναι ένας καινούργιος μεταεπιστημονικός κλάδος στην εκπαίδευση

1.2 Εκπαιδευτική Ρομποτική.

Η εκπαιδευτική ρομποτική αφορά μια σχετικά νέα μέθοδο μάθησης η οποία συνενώνει γνώσεις και στοιχεία από τις βασικές επιστήμες όπως τα μαθηματικά, τις φυσικές επιστήμες, της μηχανικής και της τεχνολογίας. Έχει την ικανότητα να οδηγήσει στην εκδίπλωση μεγαλύτερων νοητικά ικανοτήτων και δεξιοτήτων για την επίλυση προβλημάτων, όπως επίσης να καλλιεργήσει την ομαδική σκέψη και την συνεργασία ανάμεσα στους μαθητές. Η εκπαιδευτική ρομποτική έχει ως βάση την θεωρία μάθησης του εποικοδομισμού (Piaget 1974) και συνεχίζει στα βήματα της εποικοδομητικής κατασκευαστικής προσέγγισης της μάθησης σύμφωνα με τις αρχές που διατύπωσε ο Papert το 1993. Πρωτεύον εργαλείο της είναι η λειτουργία και η κατασκευή κάποιου ρομποτικού οχήματος με σκοπό να καλλιεργήσει στους μαθητές τις δεξιότητες επίλυσης προβλημάτων, τον τεχνολογικό γραμματισμό, όπως επίσης και την δημιουργικότητα τους από τα πρώτα στάδια της εκπαίδευσης τους, οι μαθητές τελειώνοντας το νηπιαγωγείο έχουν μάθει να χρησιμοποιούν χειραπτικά υλικά με σκοπό να κατακτήσουν την γνώση. Οι δραστηριότητες που επιλέγει ο δάσκαλος πρέπει να περιέχουν προβλήματα που έχουν κάποιο νόημα και έχουν κάποια συσχέτιση με τον πραγματικό κόσμο, με αυτό τον τρόπο παρακινούν τους μαθητές να συμμετέχουν σε όλη την διαδικασία με σκοπό την κοινωνική αλληλεπίδραση.

Ένα απτά βασικά χαρακτηριστικά της ρομποτικής μάθησης είναι ότι συνδυάζει το παιχνίδι και την μάθηση με αποτέλεσμα να δημιουργεί ένα διασκεδαστικό και ευχάριστο περιβάλλον, έτσι κάνει την μάθηση ουσιαστικότερη και ευκολότερη. Οι μαθητές μπαίνουν στον ρόλο του επιστήμονα και ερευνητή με σκοπό την ανακάλυψη νέων ιδεών και σχεδίων που με την σειρά τους θα τους βοηθήσουν στην επίλυση των προβλημάτων που τους έχουν ανατεθεί. Οι ομάδες χωρίζονται συνήθως μεταξύ 4-5 ατόμων με διαφορετικές επιδόσεις και δεξιότητες, όπου ο καθένας αναλαμβάνει ένα ρόλο. Διαμέσου της συνεργατικής μάθησης οι μαθητές αναπτύσσουν ειδικές δεξιότητες όπως η υπευθυνότητα, η ανάληψη ρόλων και αρμοδιοτήτων, ο διάλογος, η παρατηρητικότητα, ο σεβασμός, η συν κατασκευή της γνώσης καθώς και άλλες διαφο-

ρετικές κοινωνικό γνωστικές διαδικασίες. Όλες οι ομάδες πρέπει να διαθέτουν ορισμένα χαρακτηριστικά για να θεωρηθεί η όλη διαδικασία επιτυχημένη:

- 1) Να υπάρχει μια σχετική αλληλεξάρτηση μεταξύ των μελών της ομάδας για να επιτευχθεί καλύτερο αποτέλεσμα
- 2) Να υπάρχει μια σχετική ανομοιογένεια όσο αφορά τις επιδόσεις των μελών καλοί, μέτριοι και ανεξαρτήτου φύλλου μαθητές
- 3) Ένας μαθητής να αναλαμβάνει τον ρόλο του “αρχηγού” εκ μέρους της ομάδας αλλά όλα τα μέλη να ανταμείβονται ισάξια
- 4) Τέλος η αξιολόγηση να γίνεται ομαδικά και να γίνονται ομαδικές συζητήσεις στην ολομέλεια.

Μεγάλη βαρύτητα και σημασία παίζει ο ρόλος του δασκάλου ο οποίος πρέπει να συντονίζει και να καθοδηγεί τους μαθητές με σκοπό να επιλύσουν τα προβλήματα και να κατασκευάσουν την γνώση. Ως πρώτο βήμα ο δάσκαλος είναι υπεύθυνος να δημιουργήσει τις ομάδες με σκοπό να εξασφαλιστεί η ανομοιογένεια τους, με σκοπό οι πιο καλοί μαθητές να λειτουργούν ως βοηθοί της γνώσης για τους μαθητές χαμηλότερων επιδόσεων, έπειτα πρέπει να εξηγεί στους μαθητές και να ορίζει τους κανόνες και τις κοινωνικές δεξιότητες της συνεργατικής μάθησης, να αναπτύσσει την υπευθυνότητα μεταξύ των μαθητών και να διδάσκει πως μαθαίνουν. Ο δάσκαλος θα πρέπει να μην δίνει εύκολες και έτοιμες απαντήσεις στους μαθητές, αλλά να τους δίνει βοηθήματα και στοιχεία που θα τους βοηθήσουν να ξεκολλήσουν από το πρόβλημα και να φτάσουν εν τέλει στην λύση. Επίσης ο δάσκαλος έχει την υποχρέωση να συγχαίρει και να ενθαρρύνει κάθε είδους προσπάθεια που γίνεται από τους μαθητές όπως επίσης να προωθεί τεχνικές για προβληματισμό και έρευνα και εκτός της αίθουσας.

1.3 Υπολογιστική Σκέψη.

Η ορολογία υπολογιστική σκέψη χρησιμοποιήθηκε αρχικά από τον Papert το 1993 για να μπορέσει να περιγράψει ένα είδος αναλυτικής σκέψης η οποία έχει αρκετές ομοιότητες με τη μηχανική σκέψη (σχεδιασμό και αξιολόγηση διαδικασιών), με τη μαθηματική σκέψη (επίλυση προβλημάτων) όπως επίσης και με την επιστημονική σκέψη (συστηματική ανάλυση). Αφορά την γνωστική διαδικασία που έχει ως στόχο την λύση ενός προβλήματος χρησιμοποιώντας αλγοριθμικό σχεδιασμό, τμηματοποίηση, αξιολόγηση, γενίκευση και αφαίρεση (Selby και Woollard, 2013). Οι δεξιότητες του ατόμου που καλλιεργεί η υπολογιστική σκέψη μπορούν να ενισχυθούν σε όλα τα γνωστικά αντικείμενα χρησιμοποιώντας κατάλληλες δραστηριότητες που με την σειρά τους δίνουν μεγαλύτερη σημασία στην διαδικασία της μάθησης και της σκέψης, δηλαδή όχι στο τι μαθαίνουν αλλά στο πως το μαθαίνουν. Με αυτό τον τρόπο μετατρέπει τους μαθητούμενους από καταναλωτές και χρήστες της τεχνολογίας σε αυτοεκφραστές και δημιουργούς (Κοτίνη και Τζελέπη, 2012).

Οι δεξιότητες που κυριαρχούν στην υπολογιστική σκέψη με βάση το μοντέλο των Atmatzidou & Demetriadis (2016) είναι η Ανάλυση, η Αφαίρεση, η Αλγοριθμική Σκέψη, η Γενίκευση, η Αξιολόγηση και η Επεκτασιμότητα.

- 1) Η Αφαίρεση είναι η δεξιότητα να επεξηγούμε ένα πρόβλημα ή μια λύση του, αφαιρώντας λεπτομέρειες που δεν έχουν σημασία ή είναι χαμηλότερης σημασίας.
- 2) Η Ανάλυση είναι η ικανότητα να “διαμελίζουμε” κάποιο σύνθετο και πολύπλοκο πρόβλημα σε πιο απλά και μικρότερα κομμάτια, με τα οποία διαχειριζόμαστε ευκολότερα το πρόβλημα.

- 3) Η Γενίκευση είναι η δυνατότητα των μαθητευόμενων να “εξαπλώνουν” μια προηγουμένη λύση κάποιου προβλήματος για την κάλυψη περισσότερων περιπτώσεων.
- 4) Η Αλγοριθμική Σκέψη αφορά την σαφή επεξήγηση των βημάτων ενός αλγόριθμου, τον εντοπισμό διαφορετικών αλγόριθμων που μπορούν να επιλύσουν το πρόβλημα και την επιλογή του πιο αποδοτικού.
- 5) Η Επεκτασιμότητα είναι η ικανότητα των μαθητευόμενων να δημιουργούν αυτοτελή κομμάτια κώδικα τα οποία έχουν την ικανότητα να επιλύουν το ίδιο το πρόβλημα όσο και οποιοδήποτε άλλο.
- 6) Η Αξιολόγηση περιλαμβάνει τον έλεγχο για το αν κάποιο πρότυπο λειτουργεί όπως θάπρεπε ή όχι, όπως επίσης την δεξιότητα της εκσφαλμάτωσης στην οποία εντοπίζονται πιθανά σφάλματα στο πρόγραμμα και έπειτα οι λύσεις τους από τον προγραμματιστή.

Με λίγα λόγια η Υπολογιστική Σκέψη ορίζεται ο τρόπος με τον οποίο οι άνθρωποι μπορούν να επιλύσουν πάσης φύσεως προβλήματα, αυτό όμως δεν σημαίνει ότι οι άνθρωποι σκέφτονται όπως οι υπολογιστές αλλά αντιθέτως οι ίδιοι οι άνθρωποι κάνουν τους υπολογιστές συναρπαστικούς γιατί διαθέτουν φαντασία και εξυπνάδα, επίσης η Υπολογιστική Σκέψη δεν αναφέρεται σε αντικείμενα αλλά σε ιδέες διότι δεν περιλαμβάνει μόνο το λογισμικό και υλικό που παράγουν οι προγραμματιστές και οι μηχανικοί αλλά περιλαμβάνει επίσης έννοιες που χρησιμοποιούμε για να προσεγγίσουμε και να επιλύσουμε προβλήματα, να διαχειριστούμε την καθημερινότητα μας όπως επίσης να αλληλοεπιδράσουμε και να επικοινωνήσουμε με τους γύρω μας.

1.4 Εκπαιδευτική Ρομποτική & Υπολογιστική Σκέψη.

Βάση ερευνών που πραγματοποιήθηκαν σε σχολεία Πρωτοβάθμιας αλλά και Δευτεροβάθμιας εκπαίδευσης αλλά και σε Ανώτατα εκπαιδευτικά ιδρύματα η εκπαιδευτική ρομποτική ως καινοτόμο διδακτικό εργαλείο συνεισφέρει για να αναπτυχθούν οι δεξιότητες της υπολογιστικής σκέψης για την λύση διαφορετικών προβλημάτων. Η απασχόληση με την ρομποτική περιέχει δύο ειδών δραστηριότητες: μια προγραμματιστική και μια κατασκευαστική. Οι μαθητευόμενοι στην αρχή κατασκευάζουν ρομποτικές κατασκευές και στην συνέχεια τις προγραμματίζουν, με σκοπό να δοθούν λύσεις στα προβλήματα που έχουν ανατεθεί από τον δάσκαλο. Για την υλοποίηση αυτών των ρομπότ υπάρχουν διάφορα εκπαιδευτικά πακέτα ρομποτικής που μπορούν να χρησιμοποιηθούν όπως π.χ (snap4arduino, scratch4arduino, Beebot, Lego Wedo, Lego Wedo 2.0, Gigo Smart, Machines, Lego Mindstorms NXT, Lego Mindstorms EV3, Arduino, Ardublock κ.α.), τα οποία είναι διαθέσιμα σε διαφορετικές τιμές και ικανοποιούν τις εκπαιδευτικές ανάγκες για όλες τις ηλικίες και βαθμίδες εκπαίδευσης.

Οι καθηγητές έχουν την υποχρέωση να διαλέγουν δραστηριότητες οι οποίες θα επιλύουν προβλήματα από την πραγματική ζωή και την καθημερινότητα και θα εφαρμόζονται τα μοντέλα της ανακαλυπτικής μάθησης αλλά και της μάθησης μέσω της επίλυσης προβλημάτων. Έχουν την υποχρέωση να φτιάχνουν φύλλα εργασίας διασκεδαστικά και ενδιαφέροντα, με σκοπό να ενθαρρύνουν τους μαθητευόμενους να εμπλακούν στην όλη διαδικασία, να ενθαρρύνουν την έκφραση τους στην διαδικασία της μάθησης και να υποστηρίζουν την κοινωνική τους αλληλεπίδραση, επίσης θα πρέπει να διαθέτουν κάποια συγκεκριμένη δομή όπως επίσης

να παρέχουν σαφή και κλιμακούμενης δυσκολίας βήματα προς τους μαθητές οδηγώντας τους έτσι στην επίλυση των προβλημάτων και έπειτα στην κατασκευή της γνώσης.

Τέλος για να εξασφαλιστεί η καλή συνεργασία και επιτυχία της ομάδας καθώς και η ανάπτυξη των κοινωνικών τους δεξιοτήτων έχει ως σημαντικό παράγοντα την ανάθεση των ρόλων στους μαθητές, την καθοδήγηση καθώς και τον καθορισμό των αρμοδιοτήτων, από τον ίδιο τον διδάσκοντα.

Η εκπαίδευση STEM ενσωματώνει έννοιες που συνήθως διδάσκονται ως ξεχωριστά μαθήματα σε διαφορετικές τάξεις και υπογραμμίζει την εφαρμογή της γνώσης σε πραγματικές καταστάσεις. Ένα μάθημα βασισμένο σε STEM έχει καταστάσεις πραγματικού βίου που βοηθούν τον μαθητή να μάθει και να τις ενσωματώσει σε ένα συνεκτικό μοντέλο μάθησης που βασίζεται σε πραγματικές εφαρμογές.

Κατά τη διάρκεια της τελευταίας δεκαετίας, η εκπαίδευση STEM έχει κινηθεί στο προσκήνιο και οι σταδιοδρομίες στον τομέα της Επιστήμης, της Τεχνολογίας, της Μηχανικής και των Μαθηματικών αυξάνονται. Η εκπαίδευση STEM είναι ζωτικής σημασίας για την επιτυχία.

Μερικά ακόμα οφέλη της εκπαίδευσης STEM:

1) Προετοιμάζει τα παιδιά για τις επερχόμενες τεχνολογικές καινοτομίες:

Διδάσκει τα παιδιά να είναι ενεργά για την τεχνολογία. Προετοιμάζει τα παιδιά για τις τεχνολογικές καινοτομίες που αναμφίβολα θα βιώσουν στη ζωή τους. Όλο και περισσότερο, η τεχνολογία αποτελεί αναπόσπαστο μέρος της ζωής που τα παιδιά πρέπει να καταλάβουν.

2) Ενισχύει τις δεξιότητες επίλυσης προβλημάτων:

Η εκπαίδευση STEM αναπτύσσει και ενθαρρύνει τις δεξιότητες επίλυσης προβλημάτων. Στις δραστηριότητες STEM, οι μαθητές μαθαίνουν πώς να αναλύουν τα προβλήματα και στη συνέχεια να δουλεύουν για να τα διορθώσουν και να τα ξεπεράσουν.

3) Μεγαλύτερη κατανόηση των εννοιών:

Τα παιδιά πρέπει να εμπλέκουν και να εφαρμόζουν τις έννοιες με βαθύτερο και διεξοδικότερο τρόπο, οδηγώντας σε μεγαλύτερη κατανόηση των εννοιών.

4) Εφαρμογή της γνώσης:

Το STEM ενθαρρύνει, και εμπνέει. Η γνώση διδάσκεται με τρόπο με τον οποίο χρησιμοποιείται στον πραγματικό κόσμο, με έννοιες και αντικείμενα να αλληλεπικαλύπτονται άψογα. Τα παιδιά ενσωματώνουν και εφαρμόζουν όλες αυτές τις γνώσεις.

5) Εμπνέει τη δημιουργικότητα και την καινοτομία:

Η δημιουργικότητα μπορεί να συνυπάρχει με την επιστήμη και την τεχνολογία και να οδηγήσει σε δημιουργικές σταδιοδρομίες όπως στην αρχιτεκτονική και το γραφικό σχέδιο. Στον κόσμο της καινοτομίας, η δημιουργικότητα είναι το καύσιμο για νέες τεχνολογίες και ιδέες. Η δημιουργικότητα επιτρέπει στους μαθητές να βλέπουν προβλήματα με νέους τρόπους και να σκέφτονται έξω από το κουτί.

6) Διδάσκει την ομαδική εργασία, τη συνεργασία και την επικοινωνία:

Το STEM το καθιστά προσιτό σε παιδιά όλων των βαθμίδων ικανότητας. Ακόμη και μέσα σε μία ομάδα, τα διάφορα επίπεδα ικανοτήτων μπορούν να συνεργαστούν και να μάθουν από κοινού. Η μεταφορά των διαπιστώσεων και των λύσεων με ακρίβεια και αποτελεσματικότητα είτε μέσω της καταγραφής δεδομένων, της σύνταξης αναφορών ή της παρουσίασης είναι το κλειδί για τη λήψη μέτρων.

7) Ενθαρρύνει τη λήψη κινδύνων:

Η παρακολούθηση των αναγκών, η ανάπτυξη ιδεών, ο πειραματισμός με τις λύσεις ενθαρρύνει ένα σημαντικό επίπεδο ανάληψης κινδύνου που πολλά παιδιά θέλουν να αποφύγουν. Τα παιδιά θα μάθουν να υποστηρίζουν την ιδέα τους, να το κάνουν να συμβούν και, εάν αποτύχει, να προσπαθήσουν ξανά.

8) Πλήρης εκμάθηση της διαδικασίας:

Τα παιδιά θα βιώσουν ολόκληρη τη διαδικασία ολοκλήρωσης ενός έργου από την αρχή μέχρι το τέλος. Οι δεξιότητες για να μάθουν πώς να ερευνούν, να σχεδιάζουν, να συλλέγουν δεδομένα, να σχεδιάζουν και να συνάγουν συμπεράσματα είναι ζωτικής σημασίας για την επιτυχία των παιδιών που θέλουν να λύσουν τις πραγματικές ανάγκες.

9) Χτίζει την επιμονή και τον προσδιορισμό:

Τα παιδιά έχουν την ελευθερία σκέψης κριτικά, δημιουργικά και να καινοτομούν. Και παρέχονται ευκαιρίες στα παιδιά να αποτύχουν και να προσπαθήσουν ξανά σε ένα ασφαλές περιβάλλον. Υπογραμμίζει την αξία της αποτυχίας ως εκπαιδευτική ευκαιρία και βοηθά να αγκαλιάσει τα λάθη. Παρέχει στο παιδί την εμπιστοσύνη για να κάνει στόχους και να λάβει μετρήσιμα βήματα για να τα επιτύχει.

10) Υποστηρίζει την αγάπη της μάθησης:

Το STEM ενθαρρύνει την αγάπη της μάθησης, ενσταλάζει το πάθος και οδηγεί στην εκμάθηση. Οδηγεί την περιέργεια και την πίστη στις ικανότητες του παιδιού και τη δίψα της γνώσης. Με αυτές τις δεξιότητες στην εργαλειοθήκη τους, θα είναι εξοπλισμένα για να αντιμετωπίσουν τις προκλήσεις της τριτοβάθμιας εκπαίδευσης και του πραγματικού κόσμου.

1.5 Μια μικρή σύγκριση μεταξύ visual-based και text-based προγραμματισμού:

Η πλειοψηφία των γλωσσών προγραμματισμού είναι text-based, αλλά υπάρχουν και αρκετές γλώσσες βασισμένες σε visual-based προγραμματισμό όπως αυτές που ήδη έχουν αναφερθεί παραπάνω εν συντομία visual-based είναι η μέθοδος προγραμματισμού που επιτρέπει την χειραγώγηση εικονικών στοιχείων-π.χ drag and drop blocks για την κατασκευή προγράμματος, εν αντίθεση με τον text-based προγραμματισμό που πρέπει να γράφεις γραμμές και γραμμές κώδικα για να φτιαχτεί ένα πρόγραμμα.

Παρακάτω φαίνεται ένα απλό πρόγραμμα που απλά εμφανίζει “Hello word!” σε visual και text based προγραμματισμό:



VS.

```
#include <iostream>
using namespace std;

int main() {
    cout << "Hello world!" << endl;
    return 0;
}
```

Εικόνα 1.4.1.1

Είναι εμφανές γιατί η εισαγωγή στον προγραμματισμό πρέπει να γίνεται σε visual based περιβάλλον .

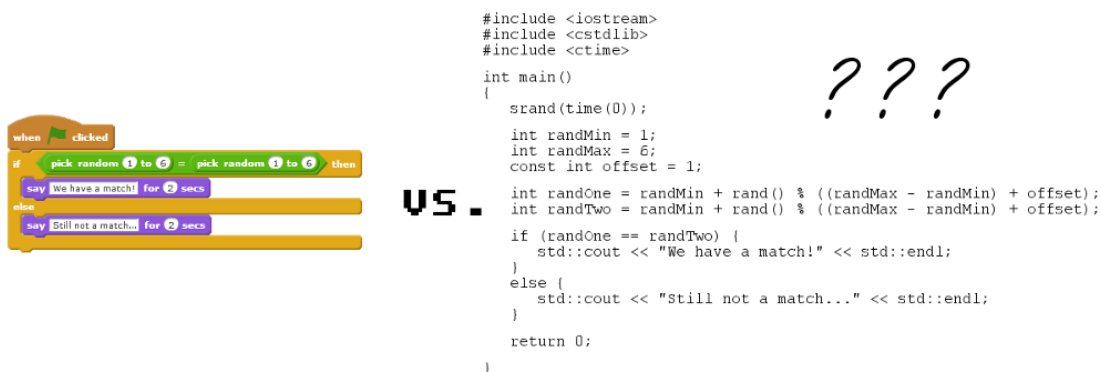
Πιο συγκεκριμένα:

1) Ευαναγνωστικότητα:

Για κάποιον που κάνει την πρώτη εισαγωγή στον προγραμματισμό, το scratch είναι πού πιο ευανάγνωστο. Τα Blocks στο scratch τείνουν να είναι πιο ευθέως προς τον προγραμματιστή και είναι πιο εύκολο να μαντέψεις τι κάνει ένα block, ακόμα και αν την το 'χεις χρησιμοποιήσει ή ακόμα και άμα το βλέπεις πρώτη φορά.

Με ένα παράδειγμα σε C++ όπως αυτό παρακάτω δεν είναι εύκολο να καταλάβεις κατευθείαν τι γίνεται στο text-based προγραμματισμό, θα αναρωτηθεί ο αρχάριος 'Τι είναι το iostream?', 'τι είναι όλα αυτά τα περίεργα σύμβολα?', 'τι είναι η main?'

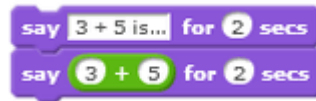
Δεν είναι καθόλου ξεκάθαρο όπως στην visual based γλώσσα:



Εικόνα 1.4.1.2

2) Ευκολία χρήσης

Ο τρόπος που κατασκευάζετε ένα πρόγραμμα στο scratch είναι με την μέθοδο drag προ εγκατεστημένων block που έχουν προ-γεγραμμένο κώδικα. Για παράδειγμα αν θες να εκτυπώσεις την πράξη 3+5 μπορεί να γίνει κάπως έτσι:



Εικόνα 1.4.1.3

Σε αντίθεση με την γλώσσα C++ που το ίδιο ακριβώς θα γινόταν κάπως έτσι:

```
#include <iostream>
using namespace std;

int main() {
    int calc = 3 + 5;
    cout << "3 + 5 is..." << calc << "!" << endl;
    return 0;
}
```

Εικόνα 1.4.1.5

3) Τεχνογνωσία

Όταν προγραμματίζεις με text-based γλώσσα, είναι σημαντικό να μάθεις πρώτα τις βασικές έννοιες του προγραμματισμού, ειδικά αν προσπαθείς να αυτοδιδασκείσαι σχεδόν σε όλες τις γλώσσες τείνουν να υπάρχουν κοινά στοιχεία όπως loop, Boolean, array, με λίγα λόγια άμα δεν είσαι εξοικειωμένος με εννοιές όπως αυτές που είναι πολύ βασικές δεν γίνετε να φτιαχτεί ούτε το πιο απλό πρόγραμμα.

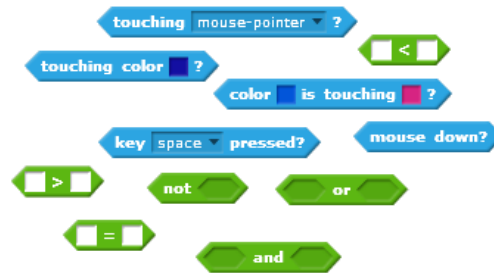
Για πιο πολύπλοκες γλώσσες όπως η C θα πρέπει επίσης να ξέρεις τι είναι οι headers, libraries, functions, dynamic memory management και τόσα άλλα.

Αλλά τι σημαίνουν αυτές οι λέξεις;

Πώς μπορείς να εξηγήσεις σε κάποιο παιδί που προσπαθεί να μπει στον προγραμματισμό αυτές τις έννοιες;

Με το scratch δεν χρειάζεται να ξέρει κάποιος ούτε τις μισές από τις παραπάνω έννοιες χωρίς να χρειάζεται να διαφοροποιείς το ένα Block από άλλο.

Για παράδειγμα μπορείς άνετα να χρησιμοποιήσεις μια μεταβλητή τύπου Boolean στο scratch χωρίς καν να το καταλάβεις, οι Boolean στο scratch ξεχωρίζουν από τριγωνικό σχήμα στο τέλος του block τους.



Εικόνα 1.4.1.6

Μερικές σκέψεις:

Ναι, το Scratch είναι ένας πολύ καλός τρόπος για τους αρχάριους να ξεκινήσουν με τον προγραμματισμό, αλλά δεν είναι όλες οι visual-based γλώσσες προγραμματισμού ίδιες! Το Scratch απλώς τυγχάνει να απευθύνεται σε ανθρώπους που ξεκινούν με τον προγραμματισμό.

Ναι, μπορείτε να κάνετε κάποια εντυπωσιακά πράγματα με το Scratch, αλλά δεν υποκαθιστά τα πράγματα που μπορούν να κάνουν οι περισσότερες γλώσσες που βασίζονται σε κείμενο. Δεν θα είστε σε θέση να κάνετε το επόμενο βέλτιστο λογισμικό σύνθεσης ή την επόμενη πλατφόρμα κοινωνικών μέσων χρησιμοποιώντας το Scratch. Κοιτάξτε το ως εκμάθηση βήμα-βήμα: Είναι ένα εξαιρετικό μέρος για να αναπτύξετε αυτό που είναι πραγματικά σημαντικό - μια στιβαρή προγραμματιστική λογική - χωρίς να χρειάζεται να είστε σε επαφή με όλες τις υπερβολικές τεχνικές γνώσεις, ή να ανησυχείτε μήπως χρειαστεί να εξετάσετε κάθε μια γραμμή του κώδικα σε περίπτωση σφάλματος, απλά για να βεβαιωθείτε ότι δεν υπάρχουν τυπογραφικά λάθη.

Όσο οι μαθητές γίνονται πιο εξοικειωμένοι και πιο απαιτητικοί στον προγραμματισμό μια block-based γλώσσα ξεκινάει να γίνεται όλο και πιο περιορισμένη, γίνεται δύσκολη στο να διαχειριστείς και να επεξεργαστείς μεγάλα προγράμματα, επίσης γίνεται ανιαρό με τόσο drag-and-drop που χρειάζεται ακόμα και για το πιο απλό πρόγραμμα όπως το $x=1+2$.

Με λίγα λόγια όσο καλός και να ακούγεται ο visual-based προγραμματισμός έχει το πολύ σημαντικό μειονέκτημα ότι είναι περιορισμένος σε μετέπειτα στάδια στην εξέλιξη της εκμάθησης προγραμματισμού αλλά αυτό δεν αναιρεί ότι είναι πολύ καλός για την εισαγωγή στον προγραμματισμό και την εκμάθηση προγραμματιστικής σκέψης που είναι παντού ίδια είτε visual-based είτε text-based προγραμματισμό ίσως στα επόμενα χρόνια εξελιχθεί και μπορεί να κάνει περισσότερά αλλά για την ώρα το ιδανικό είναι μετά την εισαγωγή με το scratch είναι σχεδόν υποχρεωτική η μετάβαση σε text-based προγραμματισμό.

1.6 Λογισμικά που υποστηρίζουν block based προγραμματισμό.

1) Scratch

Το Scratch είναι μια διερμηνευόμενη (Στον προγραμματισμό, διερμηνευμένη γλώσσα είναι μια γλώσσα προγραμματισμού η υλοποίηση της οποίας συνήθως αποτελείται από έναν διερμηνέα.) δυναμική οπτική γλώσσα προγραμματισμού βασισμένη και υλοποιημένη σε Squeak. Όντας δυναμική, επιτρέπει σε αλλαγές του κώδικα ακόμη και κατά τη διάρκεια της εκτέλεσης των προγραμμάτων. Έχει ως στόχο τη διδασκαλία εννοιών προγραμματισμού σε παιδιά και εφήβους και να τους επιτρέψει να δημιουργήσουν παιχνίδια, βίντεο και μουσική. Μπορεί να μεταφορτωθεί δωρεάν και χρησιμοποιείται σε μια ευρεία ποικιλία δράσεων εντός και εκτός του σχολείου ανά τον κόσμο.

Η δημοτικότητα του Scratch στην εκπαίδευση οφείλεται στην ευκολία με την οποία μπορούν να δημιουργηθούν προγράμματα: οι εντολές και οι δομές δεδομένων είναι απλές και είναι τουλάχιστον μερικά γραμμένες στην καθομιλουμένη, και η δομή του προγράμματος μπορεί να σχεδιαστεί όπως ένα παζλ, με αποσπώμενα κομμάτια κώδικα που μπορούν να μετακινηθούν και να προσαρμοστούν μαζί.

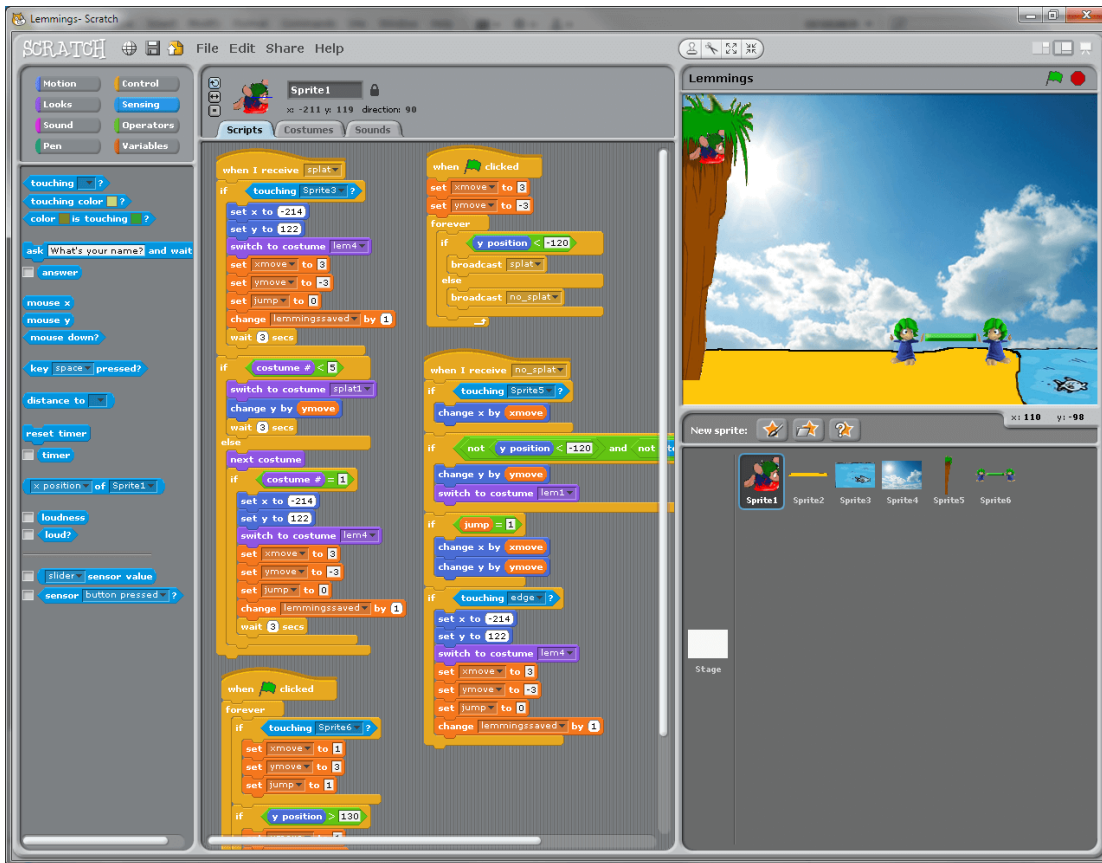
Το Scratch χρησιμοποιείται παγκοσμίως σε διάφορα σχολεία και εκπαιδευτικούς οργανισμούς.

Το scratch χρησιμοποιείται από πάρα πολλά περιβάλλοντα όπως αυτά που αναφέρθηκαν και στο κεφάλαιο 1, όλα τα περιβάλλοντα έχουν σαν κοινό σκοπό το ίδιο πράγμα, την εισαγωγή των παιδιών στον block-based προγραμματισμό αλλά έχουν και πολλά μη κοινά όπως για παράδειγμα τις υποστηριζόμενες πλακέτες και αισθητήρες.

Το scratch ήταν πρώτη επιλογή για την υλοποίηση της παρούσας πτυχιακής αλλά λόγω του αργού σχετικά πρωτόκολλου επικοινωνίας μέσω Bluetooth που διαθέτει για την επικοινωνία Arduino και υπολογιστή είχε κάποιες αργές αντιδράσεις σε καίρια σημεία και για αυτόν το λόγο απορρίφθηκε.

Πέρα από αυτό, το περιβάλλον είναι σχεδόν ίδιο με αυτό που χρησιμοποιήθηκε εν τέλη, τα γραφικά είναι τα ίδια όπως και οι λειτουργίες που διαθέτει.

Παρακάτω φαίνεται το περιβάλλον Scratch:



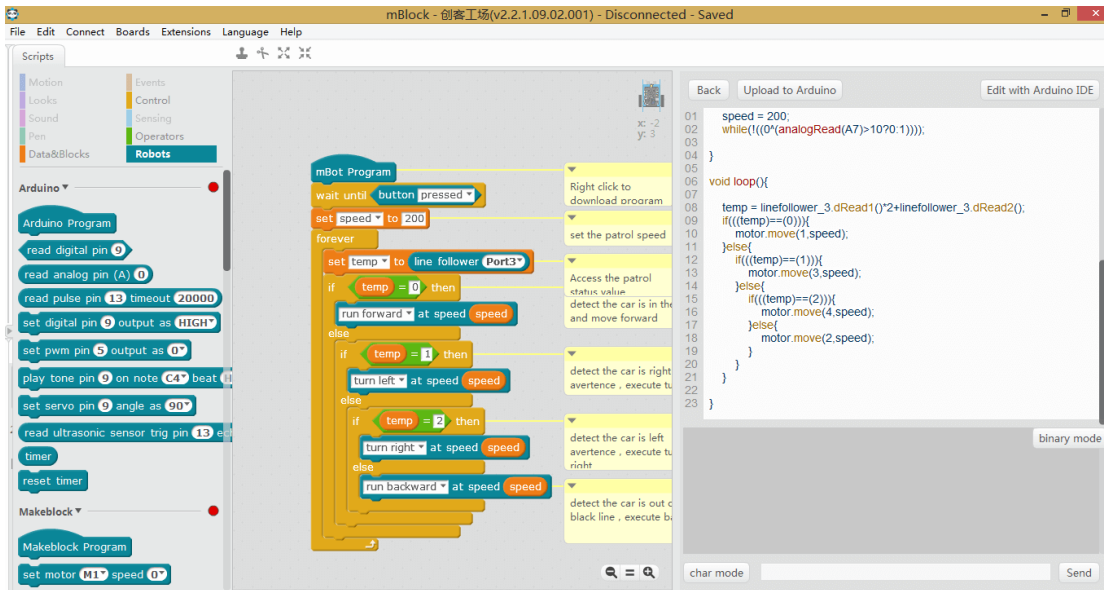
Εικόνα 1.5.1

2) Ardublock

Το Ardublock είναι και αυτό ένα από τα πιο γνωστά περιβάλλοντα block based programming γιατί εκτός από τον προγραμματισμό με εικονικά στοιχεία προσφέρει και την δυνατότητα το πρόγραμμα να μεταφράζει τον εικονικό κώδικα σε κώδικα Arduino και ο κώδικας να ανεβαίνει σε C χωρίς να χρειάζεται κάποιον μεταφραστή για να μεταφράζει τα εικονικά στοιχεία.

Μπορεί αυτό να ακούγεται μεγάλο πλεονέκτημα σε σχέση με τα άλλα, αλλά δεν είναι έτσι γιατί με αυτόν τον τρόπο χάνετε η real-time διαδραστικότητα που μπορεί να έχει ο χρήστης με αυτό, για παράδειγμα άμα ανεβάσεις κάποιο πρόγραμμα στο Arduino κ βρεις κάποιο λάθος δεν δίνετε η δυνατότητα να το κάνεις απευθείας μέσα από πρόγραμμα, πρέπει να ξανα-συνδέσεις το Arduino με τον υπολογιστή να ξαναανεβάσεις τον κώδικα να το ξανατρέξεις κοκ.

Παρακάτω φαίνεται το περιβάλλον Ardublock:

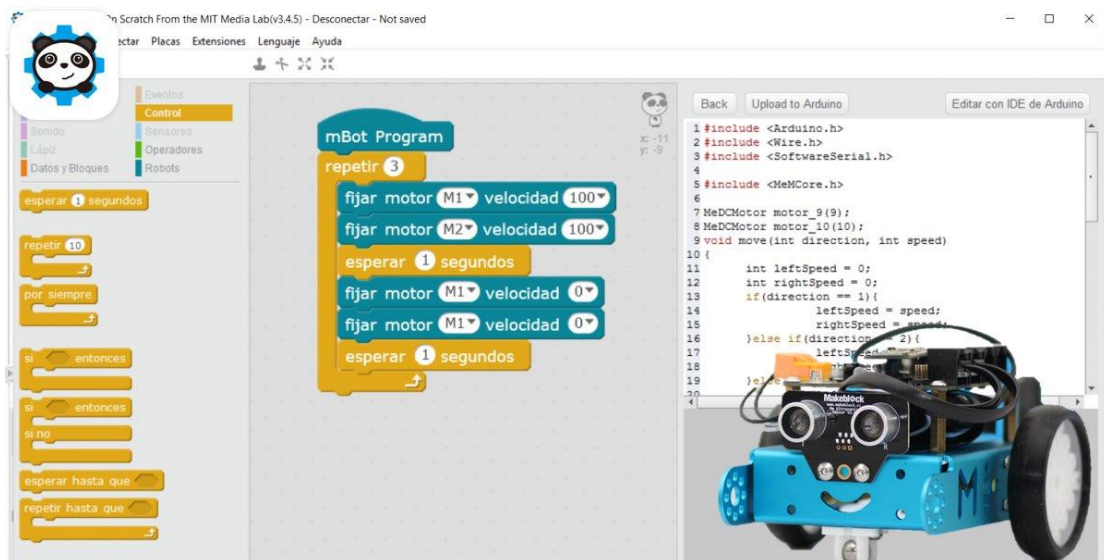


Εικόνα 1.5.2

3) mBlock

Το mBlock θα μπορούσε να είναι ένας συνδυασμός των δυο παραπάνω, διαθέτει block-based programming και μεταφραστή των block σε κώδικα Arduino είναι συμβατό με το Arduino αλλά έχει αρκετές επιλογές οι οποίες είναι αποκλειστικά για το δικό τους ρομπότ που είναι ως πώληση, ναι μεν δηλαδή σου δίνει επιλογές αλλά αυτές που είναι πιο χρήσιμες για την υλοποίηση της πτυχιακής δεν είναι διαθέσιμες ή δεν λειτουργούν όσο σωστά θα έπρεπε.

Παρακάτω φαίνεται το περιβάλλον mBlock:



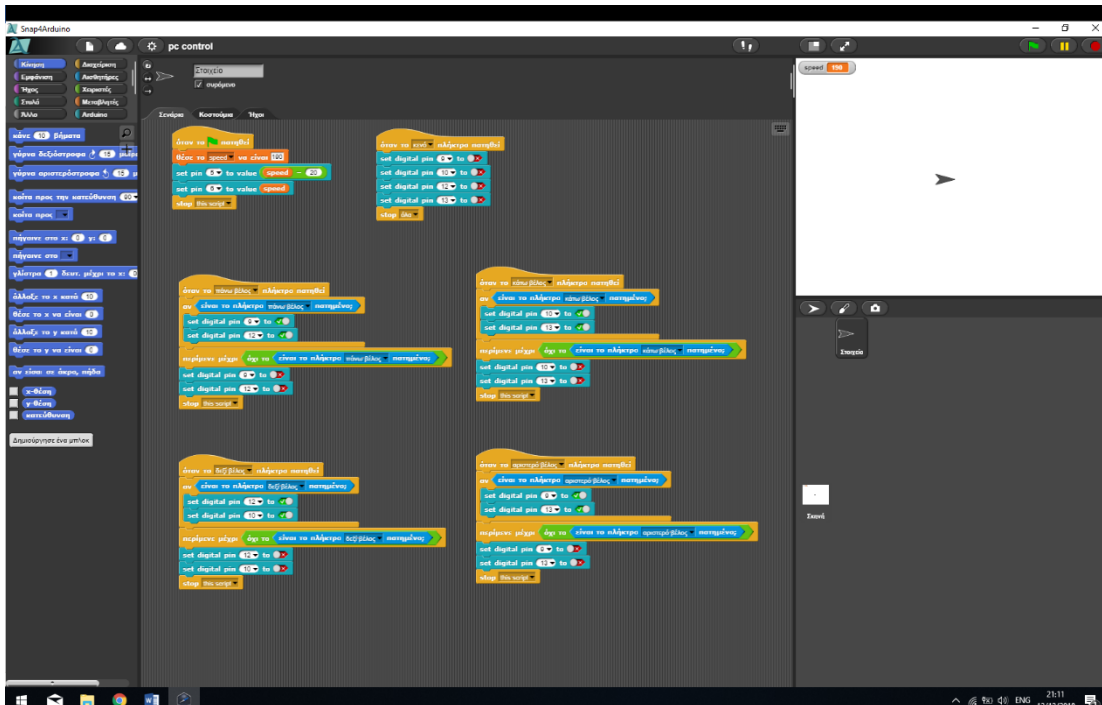
Εικόνα 1.5.3

4) Snap4arduino

Μετά από μελέτη και δοκιμές με τα πιο κοινά και πιο εξελιγμένα περιβάλλοντα καταλήξαμε στο περιβάλλον snap4arduino το οποίο συγκριτικά με τα υπόλοιπα είναι πιο ώριμο τεχνολογικά και έχει τεράστια συμβατότητα με τις πλακέτες Arduino και τους αισθητήρες τους.

Το snap είναι μια βελτιωμένη έκδοση του scratch, διαθέτει ακριβώς τις ίδιες λειτουργίες αλλά με αρκετές extra, όπως η πρόσθεση καινούργιων block, λειτουργίες που δεν βρίσκουμε στα υπόλοιπα όπως την προσθήκη βιβλιοθηκών για την υλοποίηση οποιουδήποτε project χωρίς κανένα συμβιβασμό και επίσης έχει ταχύτερη μεταφορά δεδομένων σε σχέση με το scratch με αποτέλεσμα να έχει το ρομπότ πολύ καλύτερη απόκριση σε στιγμές που είναι απαραίτητο όπως την ανίχνευση εμποδίου και την γρήγορη απόκριση αποφυγής.

Παρακάτω φαίνεται το περιβάλλον snap4arduino:



Εικόνα 1.5.4

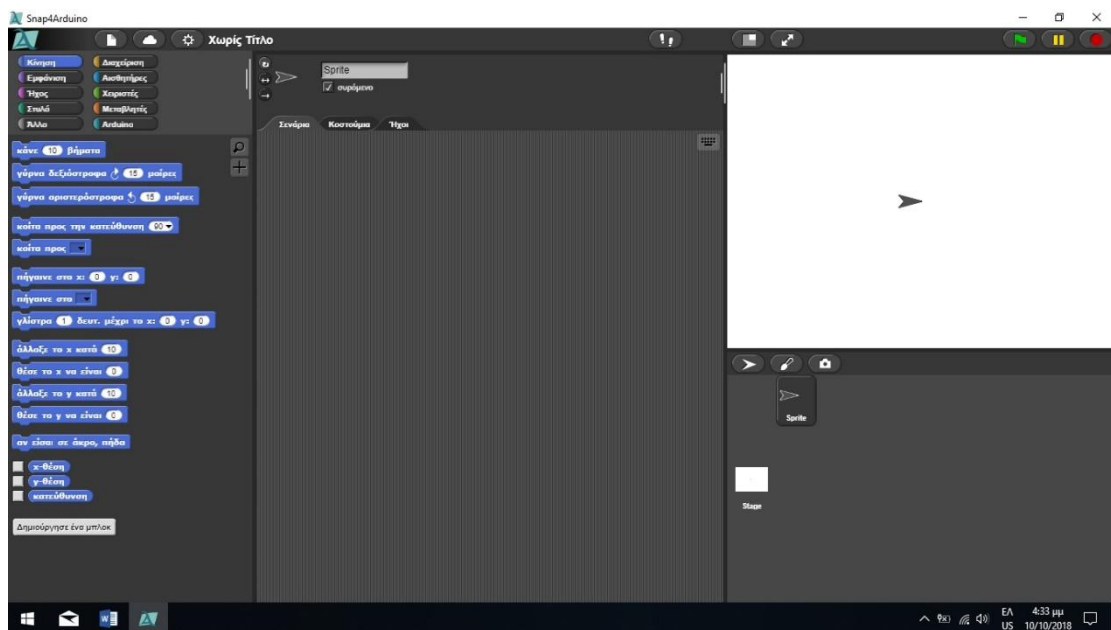
Κεφάλαιο 2 Snap4Arduino

2.1 Τι είναι το Snap4Arduino:

Το Snap4Arduino το οποίο διανέμεται δωρεάν (<http://snap4arduino.rocks/>) βασίζεται στην τεχνολογία του Scratch που αναφέραμε παραπάνω με την διαφορά ότι είναι τροποποιημένο ώστε να επικοινωνεί με την πλατφόρμα του Arduino.

Σε σχέση με το απλό Scratch το Snap4Arduino παρέχει νέα blocks για να μπορεί ο χρήστης να ελέγχει τους διάφορους αισθητήρες και τα διάφορα Shields του Arduino ,παρέχει blocks για τον έλεγχο των αναλογικών και ψηφιακών reads και writes.

2.2 To Interface:

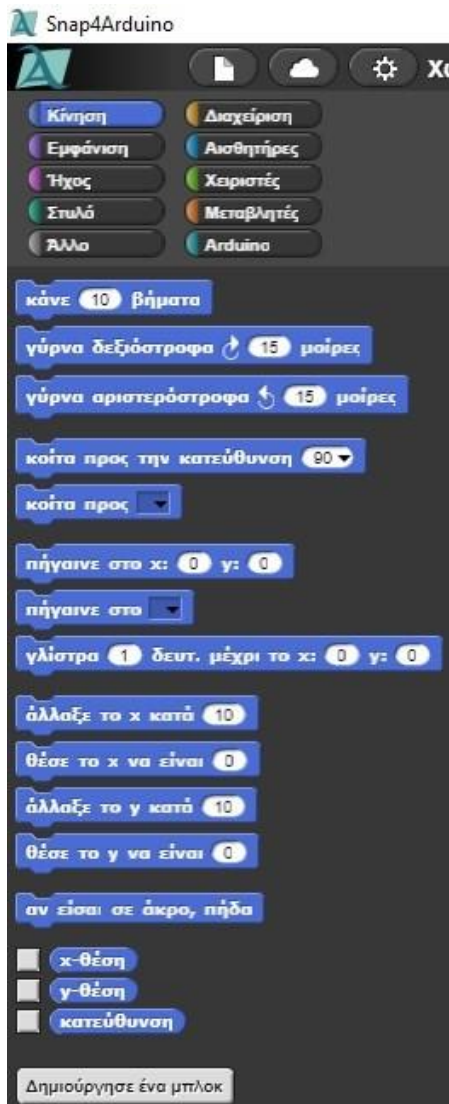


Εικόνα 2.2

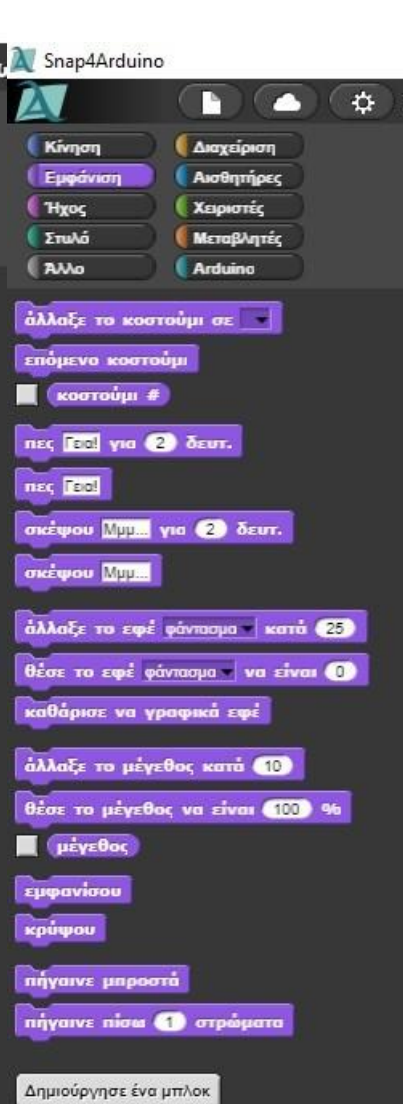
Όπως βλέπουμε και από την εικόνα το Snap είναι μεταφρασμένο και στην Ελληνική γλώσσα χάρη στη μεγάλη έκταση της κοινότητας του.

Όπως παρατηρούμε από την εικόνα το Interface χωρίζεται σε 10 διαφορετικές κατηγορίες. Για αρχή βλέπουμε την πάνω αριστερή γωνία η οποία περιλαμβάνει τις 9 κύριες κατηγορίες των Blocks που διαθέτει το πρόγραμμα και ακριβώς από κάτω τα Blocks της κατηγορίας που έχουμε επιλέξει.

Blocks που αφορούν την κίνηση και την εμφάνιση:

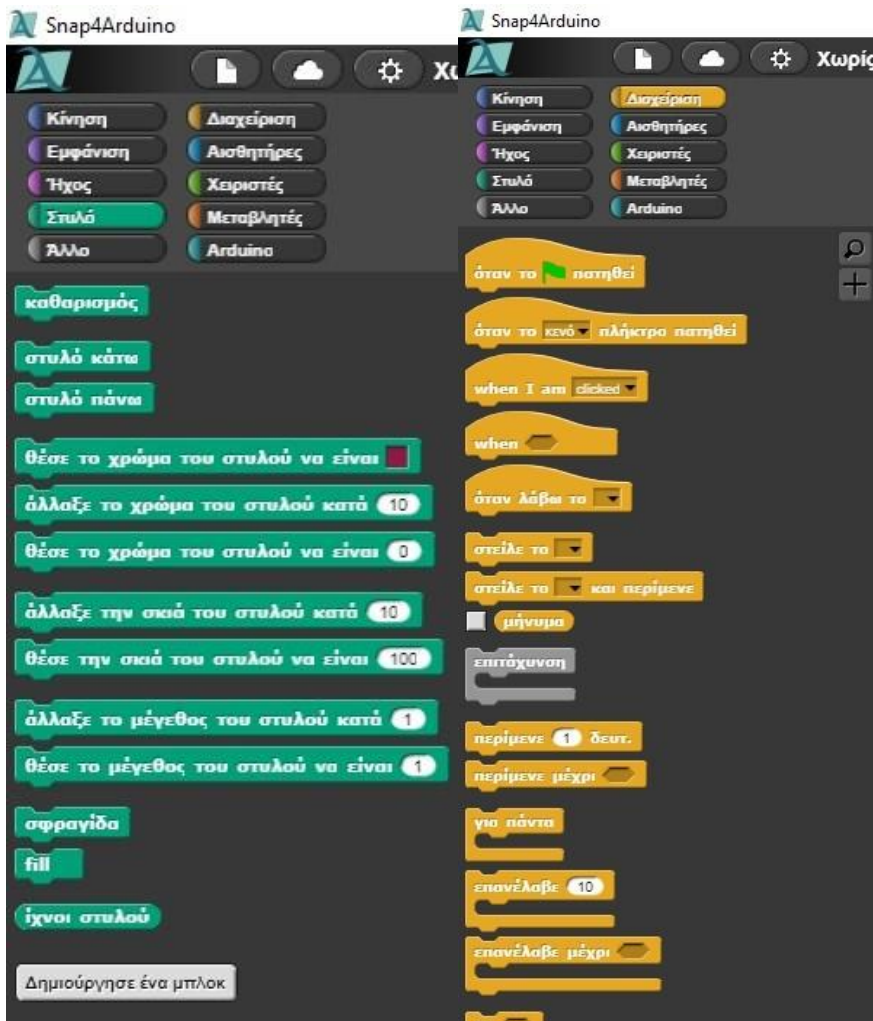


Εικόνα 2.3



Εικόνα 2.4

Blocks που αφορούν τον στυλό και την διαχείριση:



Εικόνα 2.5

Εικόνα 2.6

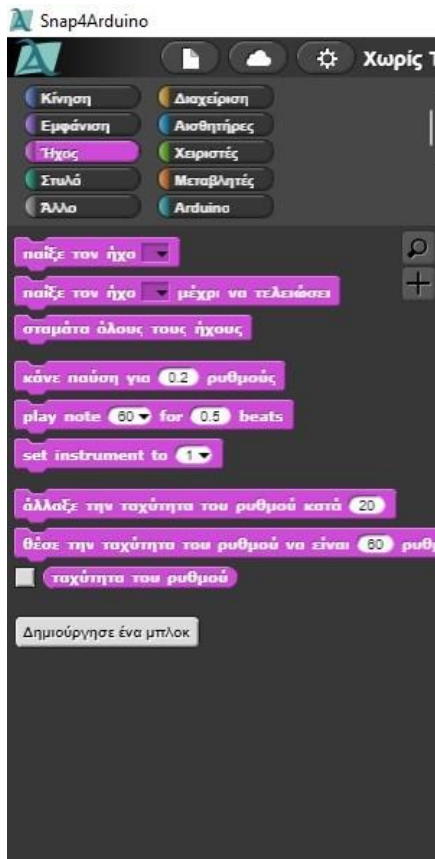
Blocks που αφορούν τον έλεγχο αισθητήρων και τελεστών:



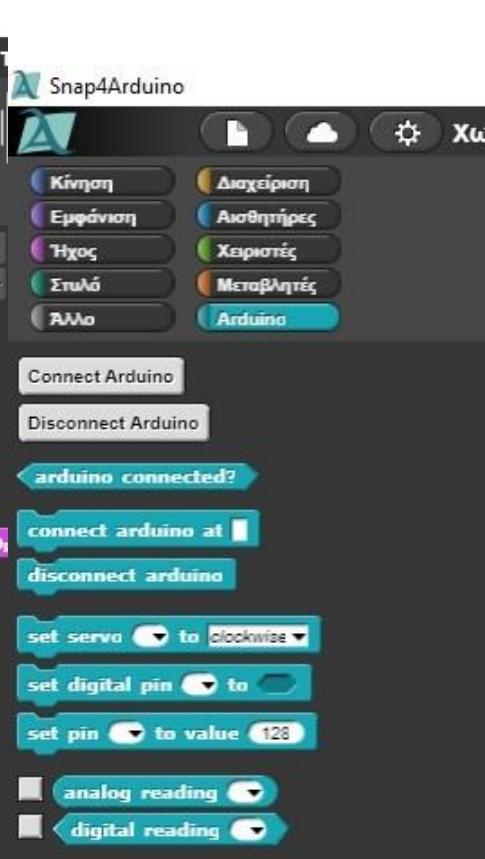
Εικόνα 2.7

Εικόνα 2.8

Blocks που αφορούν τον ήχο και την διαχείριση των pin του Arduino:



Εικόνα 2.9



Εικόνα 2.10

Blocks που αφορούν τις μεταβλητές

Εδώ μπορούμε να δημιουργήσουμε δικές μας μεταβλητές και λίστες για να χρησιμοποιήσουμε στο πρόγραμμά μας.



Εικόνα 2.11

Κεντρική περιοχή

Εδώ ουσιαστικά «γράφουμε» τον κώδικα μας όπως φαίνεται στην παρακάτω εικόνα.



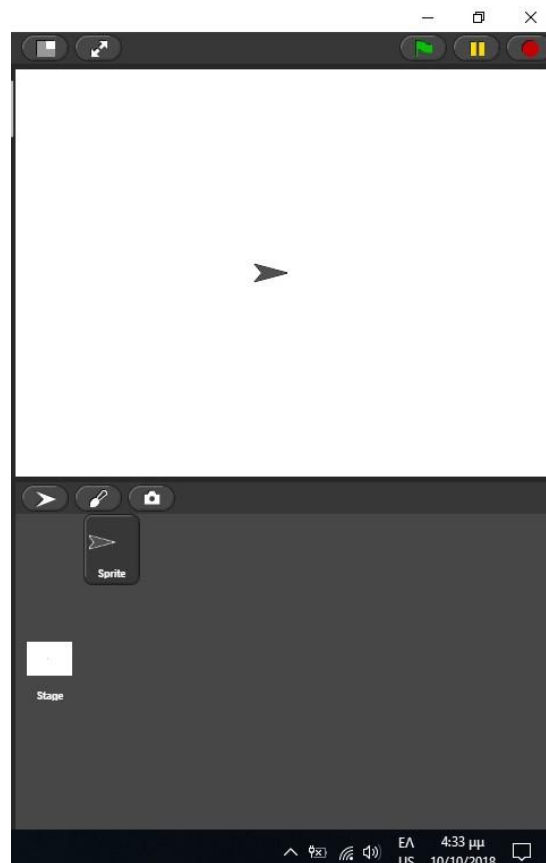
Εικόνα 2.12

Όπου με λίγα λόγια το παραπάνω όταν εκτελεστεί ενεργοποιεί τα ψηφιακά Pins 10 και 11 και στέλνει έναν παλμό στα Pins 5 και 6 με τιμή 255 όπου το εύρος 0-255 αντιστοιχεί σε ακινησία (0) και φουλ ταχύτητα (255), έπειτα μετά από 5 δευτερόλεπτα που στέλνουμε φουλ ταχύ-

τητα στους τροχούς στέλνουμε 0 με αποτέλεσμα να σταματάνε οι τροχοί και με αυτήν την διεργασία το πρόγραμμά μας σταματάει.

Interface

Τέλος έχουμε την τελευταία περιοχή του Interface η οποία απεικονίζει εικονικά ένα «όχημα». Αυτό χρησιμεύει στην περίπτωση που δεν έχουμε πραγματικό ρομπότ να ελέγξουμε χρησιμοποιούμε αυτό το εικονικό το οποίο μπορεί να λειτουργήσει για την προσομοίωση κάποιας εργασίας αποκλειστικά μέσα από interface χωρίς την ανάγκη hardware.



Εικόνα 2.13

2.3 Η λογική της λειτουργικότητας πίσω από το Snap.

Εφόσον αναλύσαμε το γραφικό περιβάλλον του Interface ήρθε η ώρα να εξηγήσουμε τον τρόπο λειτουργίας του.

Όπως αναφέραμε και παραπάνω το Snap είναι μια τροποποιημένη έκδοση του Scratch το οποίο είναι μια διερμηνευόμενη δυναμική οπτική γλώσσα προγραμματισμού η οποία χρειάζεται έναν «μεταφραστή», το ίδιο λοιπόν ισχύει και για το Snap.

Το Interface που εξηγήσαμε παραπάνω είναι ο «μεταφραστής», με λίγα λόγια όταν εμείς σχεδιάζουμε το πρόγραμμά μας και το εκτελούμε ο γραφικός κώδικας δεν μπορεί να γίνει κατανοητός από το Arduino έτσι το Snap πριν επικοινωνήσει με την πλατφόρμα «μεταφράζει» τον γραφικό κώδικα σε κώδικα κατανοητό ως προς την πλατφόρμα, αυτό για να γίνει όμως πρέπει το Snap και το Arduino να μιλάνε την ίδια «γλώσσα», αυτό επιτυγχάνεται με το να «φορτώσουμε» στο Arduino ένα πρόγραμμα το οποίο θα μιλάει με το Snap και θα στέλνει τις διάφο-

Πτυχιακή Εργασία Τμήματος Μηχανικών Πληροφορικής Καρυωτάκης Βασίλειος

ρες ενδείξεις της πλατφόρμας πίσω στο Snap κατανοητές προς αυτό, την διαδικασία της «φόρτωσης» στην πλατφόρμα θα την εξηγήσουμε παρακάτω.

2.3.1 Ανέβασμα πρωτοκόλλου:

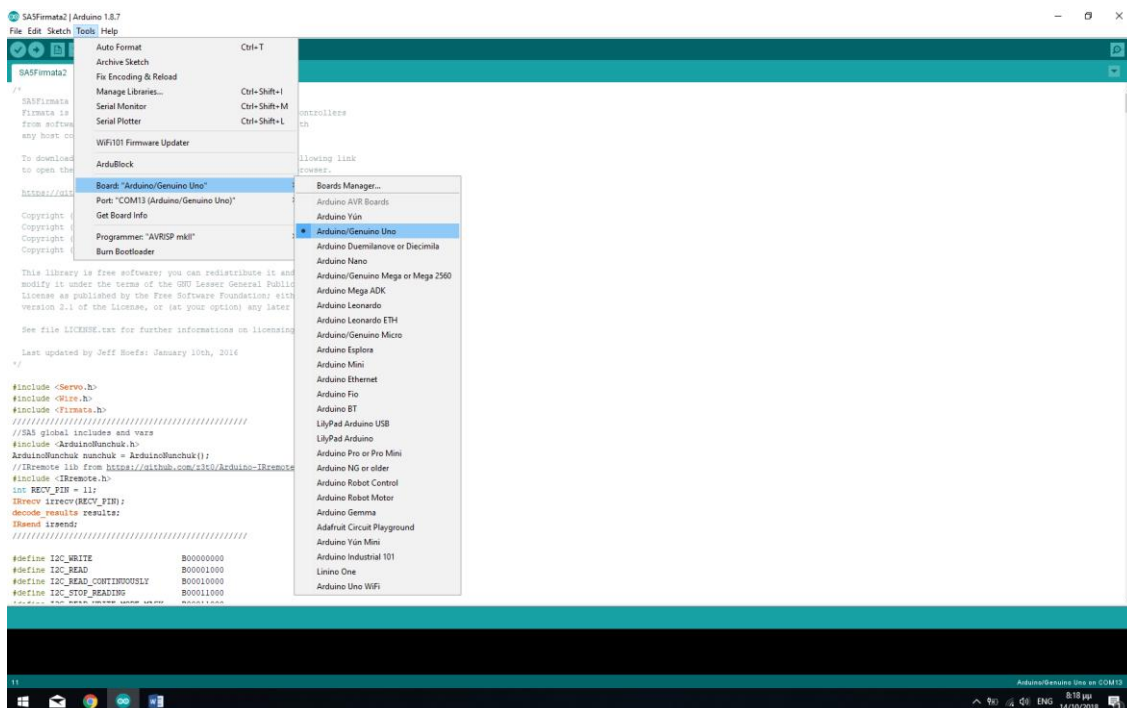
Εφόσον δείξαμε τι περιλαμβάνει η εφαρμογή snap4arduino και πια είναι η δουλεία της ήρθε η ώρα να φορτώσουμε στο Arduino το πρωτόκολλο που είναι υπεύθυνο για την επικοινωνία software-hardware που όπως αναφέρθηκε και σε προηγούμενο κεφάλαιο ουσιαστικά μεταφράζει τα blocks σε κώδικα ώστε να ναι κατανοητό στο Arduino.

Το πρωτόκολλο αυτό είναι το S5AFirmata(<https://github.com/jguille2/SA5/>)

το οποίο είναι μια εξελιγμένη μορφή του StandardFirmata που χρησιμοποιείται για τέτοιου είδους εργασίες επικοινωνίας software-hardware.

Μετά το κατέβασμα του πρωτοκόλλου ανοίγουμε το αρχείο .ino με το interface του Arduino και ενώ έχουμε συνδεδεμένο την πλακέτα με τον υπολογιστή κάνουμε τα παρακάτω βήματα:

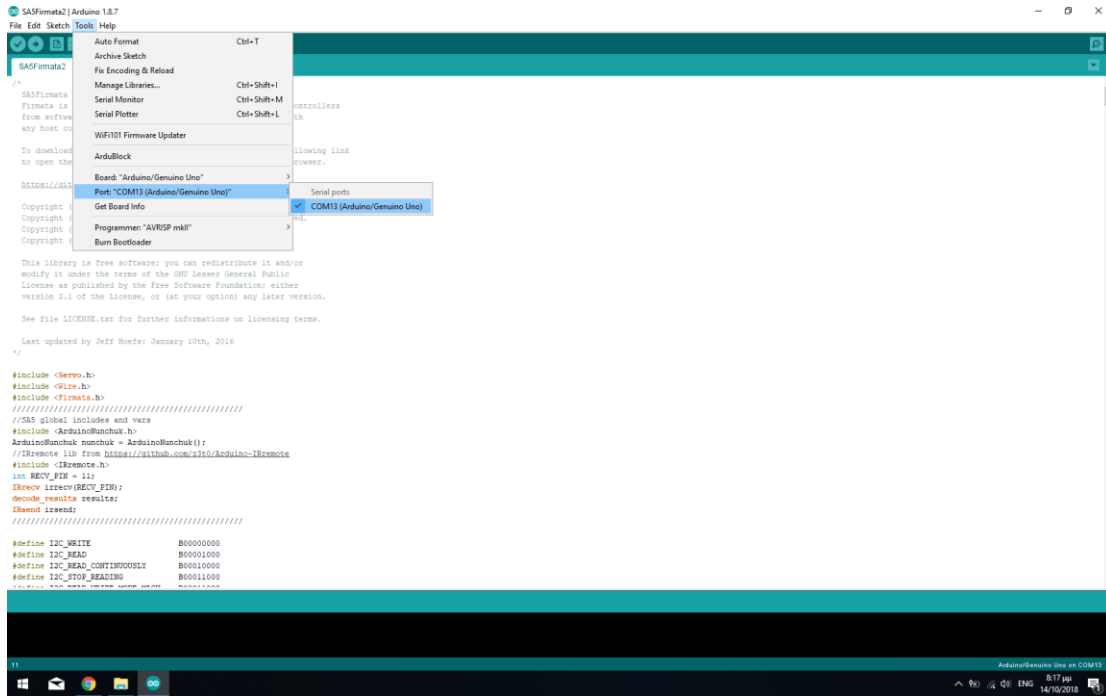
Επιλέγουμε την πλακέτα που χρησιμοποιούμε από περιβάλλον IDE



Εικόνα 2.3.1

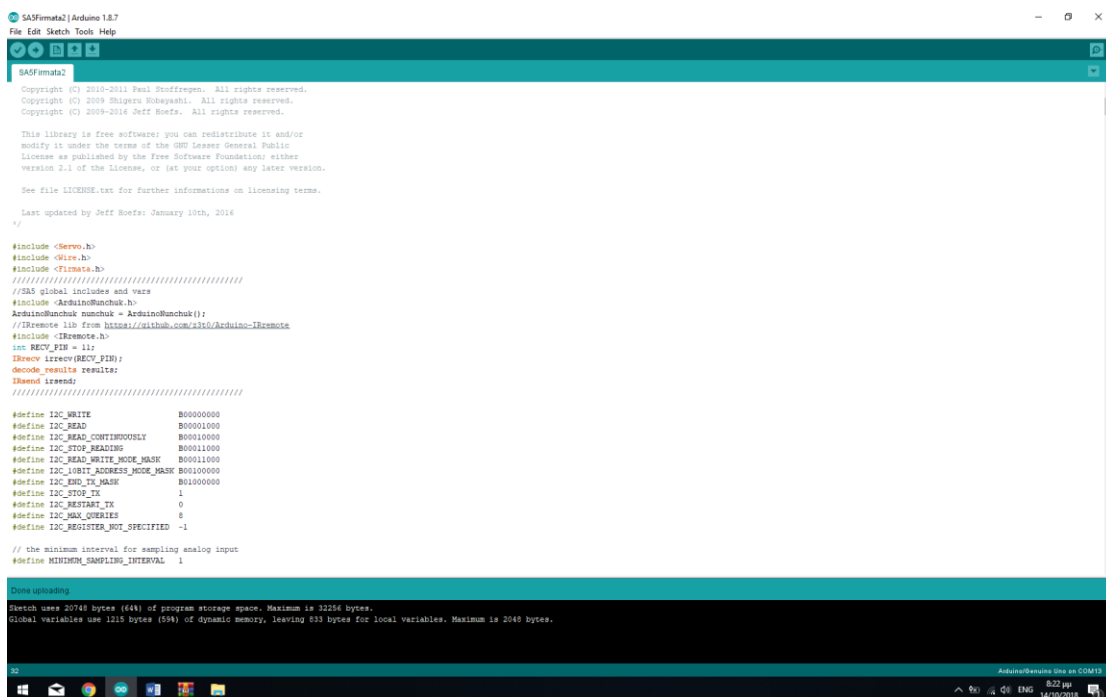
Επιλέγουμε την σωστή πόρτα που είναι συνδεδεμένη η πλακέτα

Πτυχιακή Εργασία Τμήματος Μηχανικών Πληροφορικής Καρυωτάκης Βασίλειος



Εικόνα 2.3.2

Πατάμε upload και περιμένουμε να ανέβει ο κώδικας στην πλακέτα



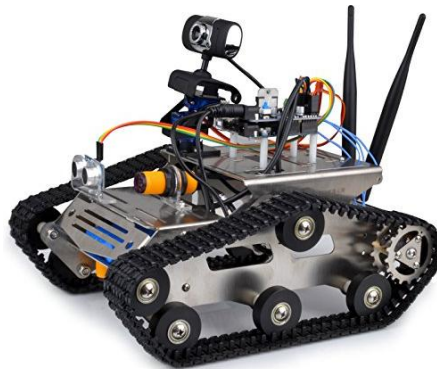
Εικόνα 2.3.3

Κεφάλαιο 3: Σχεδιασμός ρομποτικού οχήματος.

Στο παρόν κεφάλαιο θα μελετήσουμε την κατασκευή του οχήματος και με πιο τρόπο ήρθε εις πέρας η κατασκευή του, ποιες λειτουργίες θα έχει και πως θα δουλεύουν όπως επίσης και τις επιλογές που είχαμε για την επιλογή του σχεδίου κατασκευής του οχήματος.

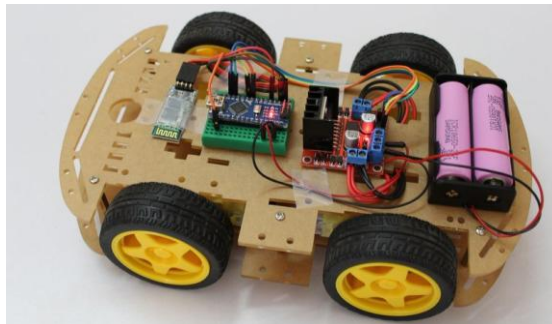
3.1 Το σασί

Το πρώτο πράγμα που χρειάζεται να γίνει είναι να δούμε τι σχήμα θα έχει το όχημα, γιατί σίγουρα υπάρχουν πολλά σχέδια αλλά δεν είναι όλα κατάλληλα για το έργο που θέλουμε να κάνουμε, για παράδειγμα υπάρχουν υλοποιήσεις όπως αυτή:



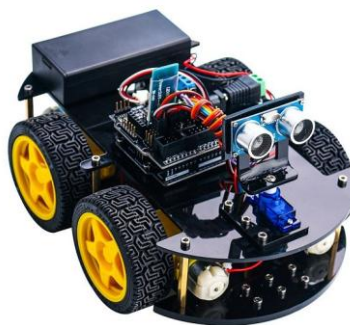
Εικόνα 3.1.1

Ή αυτή:



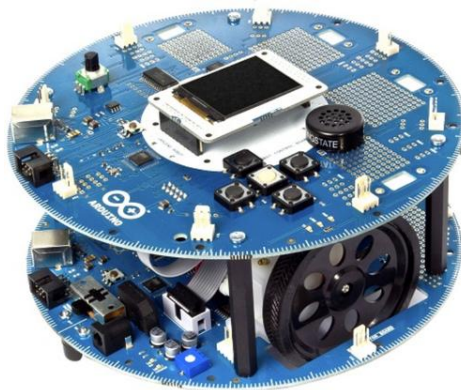
Εικόνα 3.1.2

Αυτή:



Εικόνα 3.1.3

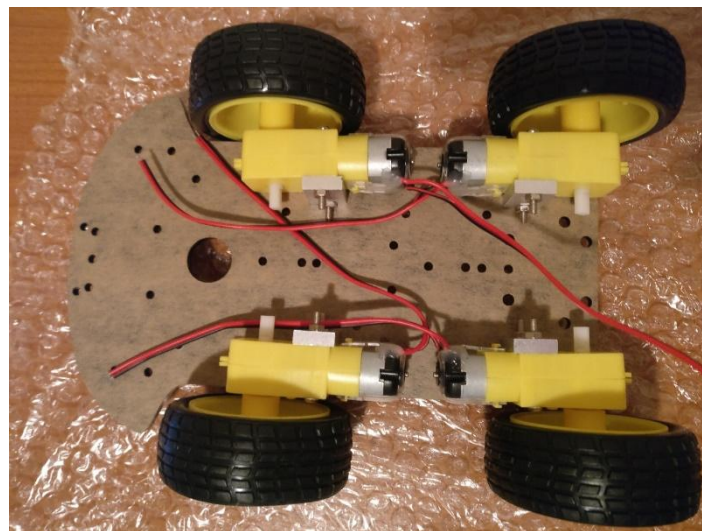
Ακόμα και αυτή



Εικόνα 3.1.4

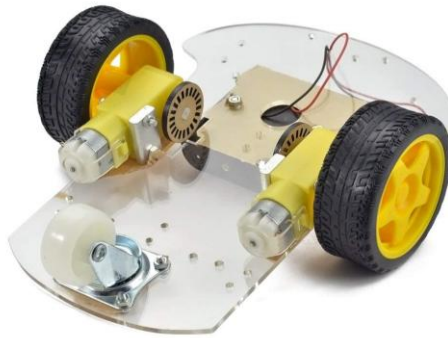
Όλες οι παραπάνω υλοποιήσεις είναι Arduino based με αισθητήρες για να κάνουν κάποια Α πράγματα, εμείς ψάχνουμε κάτι μικρό και ευέλικτο στον χώρο που να μπορεί να υποστηρίξει τις λειτουργίες και τους αισθητήρες που θέλουμε, με βάση τα παραπάνω το πιο σωστό σαν υλοποίηση θα ήταν ένα όχημα όπως στην εικόνα 2 ή 3, ένα απλό δηλαδή τετρακίνητο αμαξάκι.

Στην αρχή υλοποιήσαμε ένα αμαξάκι με 4 τροχούς όπως φαίνεται και παρακάτω:



Εικόνα 3.1.5

Αλλά μετά από δοκιμές και μελέτη του σχεδιασμού παρατηρήθηκε ότι το όχημα δεν μπορεί να πάρει απότομες στροφές και να κάνει απότομους ελιγμούς δεξιά ή αριστερά για αυτό αφαιρέθηκαν οι 2 πίσω τροχοί και στην θέση τους προστέθηκε μια μεταλλική μπίλια που είναι μέσα σε γρανάζι(caster) που επιτρέπει στο όχημα να κάνει απότομους ελιγμούς χωρίς καμία δυσκολία και χωρίς να σπαταλάει ενέργεια όπως έκαναν οι 2 τροχοί, η υλοποίηση φαίνεται παρακάτω.



Εικόνα 3.1.6

Εφόσον καταλήξαμε στο σασί είναι ώρα να δούμε τι λειτουργίες θα προσθέσουμε στο όχημα.

3.2 Λειτουργίες του οχήματος

Με μια γρήγορη αναζήτηση στο google μπορούμε να δούμε αμέτρητες λειτουργίες που μπορούμε να προσθέσουμε στο όχημα αλλά και πάλι δεν είναι όλες κατάλληλες για το σκοπό της παρούσας πτυχιακής.

Μερικές από τις πιο χρήσιμες λειτουργίες που μπορεί κάποιος να βρει είναι οι εξής:

- 1) Αποφυγή εμποδίων.
- 2) Ακολούθηση μαύρης γραμμής
- 3) Τηλεχειρισμό είτε από υπολογιστή είτε από τηλεκοντρόλ

Με αυτές λοιπόν τις λειτουργίες θα εργαστούμε παρακάτω για να τις προσαρμόσουμε στο ρομποτάκι.

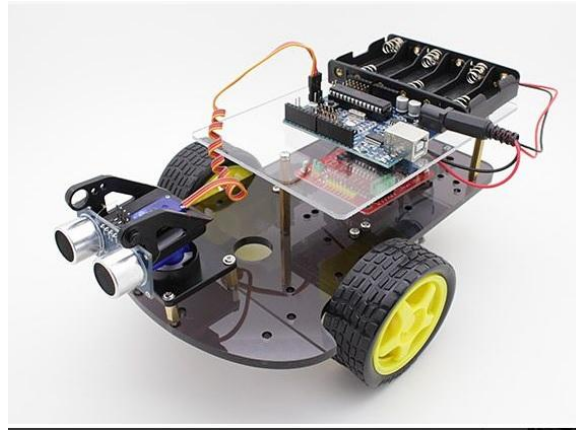
3.2.1 Αποφυγή εμποδίων

Η αποφυγή εμποδίων μπορεί να γίνει με 2 τρόπους, είτε χρησιμοποιώντας υπέρυθρες είτε υπέρηχους, και οι 2 τρόποι μπορούν να φέρουν εις πέρας την λειτουργία αυτή αλλά όχι το ίδιο αποτελεσματικά.

Οι υπέρυθρες θα ανιχνεύσουν ένα εμπόδιο όταν είναι σε πάρα πολύ κοντινή απόσταση 2-4 εκατοστών κάτι που δεν φαίνεται να είναι και τόσο ασφαλές για το όχημα σε περίπτωση λάθους υπολογισμού επίσης οι υπέρυθρες επηρεάζονται από φως που εύκολα οδηγεί σε λάθος υπολογισμό του αισθητήρα, από την άλλη οι υπέρηχοι μπορούν να ανιχνεύσουν εμπόδια σε απόσταση μέχρι και 400 εκατοστά κάτι που δίνει στο όχημα πολύ μικρότερη πιθανότητα λάθους μέτρησης της τελευταίας στιγμής επίσης δεν επηρεάζεται από το φως ή από άλλους εξωτερικούς παράγοντες.

Με βάση τα παραπάνω λοιπόν η επιλογή των υπέρηχων είναι σχεδόν μονόδρομος.

Η τοποθέτηση του αισθητήρα θα γίνει στο μπροστινό τμήμα όπως περίπου φαίνεται στην παρακάτω φωτογραφία:

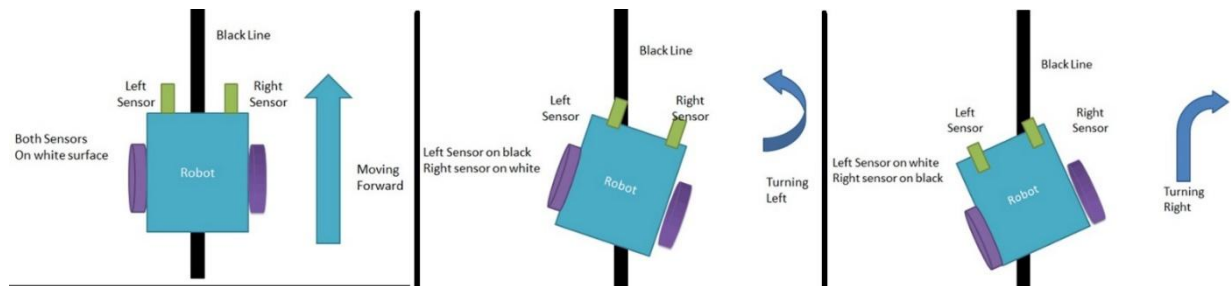


Εικόνα 3.3

3.2.2 Ακολούθηση μαύρης γραμμής.

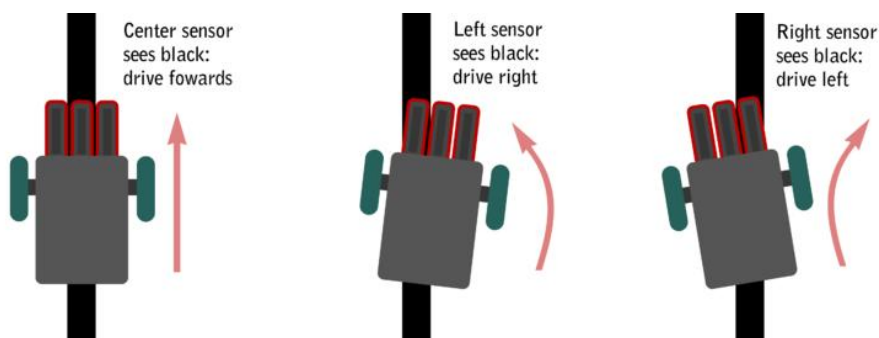
Στην συγκεκριμένη λειτουργία δεν υπάρχουν διαφορετικές υλοποιήσεις με διαφορετικά αισθητήρια, η χρήση αισθητήρα υπέρυθρων είναι η μόνη λύση, η ανίχνευση της μαύρης γραμμής γίνεται πολύ εύκολα από τους συγκεκριμένους αισθητήρες και με εξαιρετική ακρίβεια, όπως είναι λογικό οι αισθητήρες αυτοί θα τοποθετηθούν στην κάτω μεριά του οχήματος.

Μπορεί η επιλογή του αισθητήρα να ήταν εύκολη αλλά και εδώ υπάρχουν διάφορες υλοποιήσεις όπως να τοποθετηθούν 2 αισθητήρες, αριστερά και δεξιά του μπροστινού μέρους του οχήματος η υλοποίηση και ο τρόπος λειτουργίας φαίνεται στην παρακάτω εικόνα:



Εικόνα 3.4.1

Μια διαφορετική υλοποίηση περιλαμβάνει 3 αισθητήρες αντί για 2 η οποία φαίνεται παρακάτω:



Εικόνα 3.4.1

Ουσιαστικά και οι 2 υλοποιήσεις κάνουν το ίδιο πράγμα με μόνη διαφορά ότι στην υλοποίηση με τους 3 αισθητήρες προσθέτουμε μεγαλύτερη ακρίβεια και μικρότερη πιθανότητα λάθους αφού παίρνουμε μετρήσεις από 3 και όχι από 2 αισθητήρες.

3.2.3 Τηλεχειρισμός είτε από υπολογιστή είτε από τηλεκοντρόλ

1) Τηλεχειρισμός από υπολογιστή

Ο τηλεχειρισμό από υπολογιστή μπορεί να γίνει με 2 τρόπους, είτε ασύρματα είτε ενσύρματα, η ενσύρματη επιλογή όπως είναι λογικό έχει τον περιορισμό του καλώδιου σε θέμα απόστασης όπως επίσης το πρόβλημα ότι μπορεί να μπερδευτεί είτε με το όχημα είτε σε κάποιο αντικείμενο.

Αρα πάμε στην επιλογή της ασύρματης σύνδεσης η οποία μπορεί να γίνει με 2 τρόπους είτε με Bluetooth είτε με WIFI.

Η επιλογή του WIFI είναι αρκετά πιο πολύπλοκη από του Bluetooth όπως επίσης η λειτουργία με το snar και τα blocks όχι τόσο σίγουρη και αποτελεσματική.

Η τοποθέτηση στο όχημα δεν έχει σημασία γιατί το σήμα δεν μπλοκάρτε από κάτι άλλο.

2) Τηλεχειρισμός από τηλεκοντρόλ

Ο τηλεχειρισμό από τηλεκοντρόλ γίνεται από ένα συγκεκριμένο αισθητήρα υπέρυθρων ο οποίος λειτουργεί ως ματάκι για να λαμβάνει τις ενδείξεις των τηλεκοντρόλ, όπως ακριβώς έχουν και οι τηλεοράσεις.

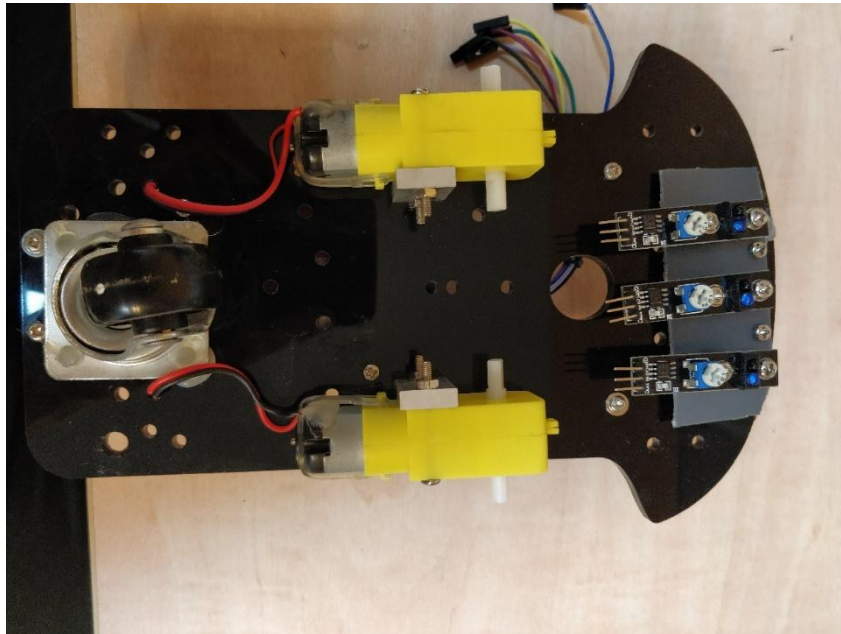
Η τοποθέτηση θα γίνει κάπου στο πίσω μέρος του οχήματος για να μπορεί το ματάκι να λαμβάνει εντολές όταν αυτό θα είναι πιο μπροστά από τον χειριστή.

Κεφάλαιο 4: Η υλοποίηση της κατασκευής

4.1 Αρχικές ρυθμίσεις

Όπως προ είπαμε το σασί του οχήματος είναι 2 κομμάτια plexiglass για να σχηματίζουν ουσιαστικά 2 ορόφους για τις πλακέτες και τους αισθητήρες που θα χρησιμοποιήσουμε , για αρχή πρέπει να τοποθετηθούν οι 2 ρόδες και το carter όπως επίσης και οι 3 αισθητήρες υπερύθρων. Όλα αυτά επειδή βρίσκονται στο κάτω μέρος του οχήματος τα βάζουμε πρώτα.

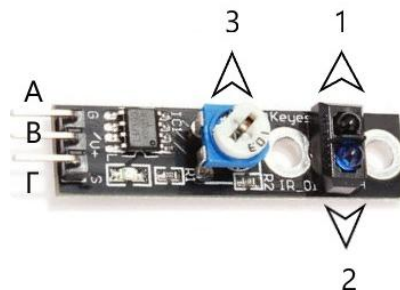
Η τελική εικόνα είναι κάπως έτσι:



Εικόνα 4.1

4.1.1 Αισθητήρας υπερύθρων Keyes KY-033 IR Line tracer.

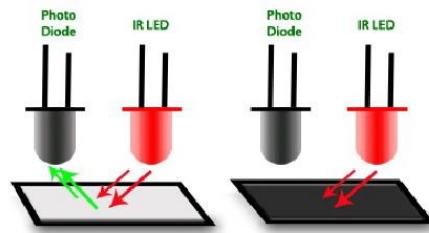
Το συγκεκριμένο module ανιχνεύει άμα υπάρχει μια αντανακλαστική επιφάνεια φώτος ή μια μη-αντανακλαστική στο σημείο που είναι ο αισθητήρας.



Εικόνα 4.1.1

Ο τρόπος που δουλεύει είναι: από το Led1 εκπέμπει μια υπεριώδης δέσμη φώτος και από το Led2 περιμένει να λάβει αυτήν την δέσμη, αυτό γίνεται όταν το υπεριώδες φως χτυπήσει πάνω σε μια σκούρα και σκληρή επιφάνεια αντανακλάται και το Led2 λαμβάνει το φως που αντανακλάστηκε και στέλνει αυτήν την πληροφορία στην πλακέτα μας, στην αντίθετη περίπτω-

ση το υπερίσως φως χάνεται και το Led2 δεν ανιχνεύει τίποτα και στέλνει αυτήν την πληροφορία στην πλακέτα μας.



Εικόνα 4.1.2

Τα επόμενα επισημασμένα σημεία του module είναι:

3: Μια κινητή ροδέλα η οποία ρυθμίζει την ευαισθησία του αισθητήρα. Με λίγα λόγια άμα ο αισθητήρας δεν ανιχνεύει απλά τον κάνουμε πιο ευαίσθητο και το ανάποδο άμα ισχύει το αντίστροφο.

A: Η σύνδεση για την γείωση του Module

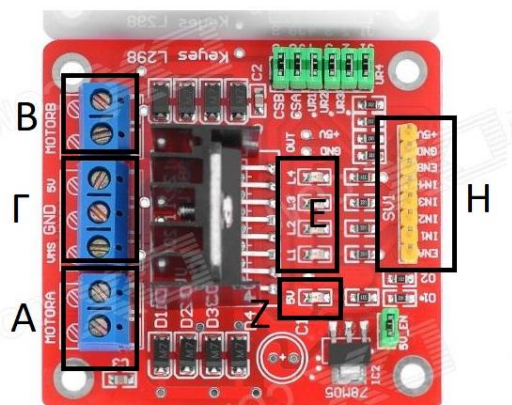
B: Η σύνδεση για την τροφοδοσία.

Γ: Η έξοδος του Module η οποία πάει στο Arduino.

4.2 Η-ΓΕΦΥΡΑ L298

Αφού λοιπόν τελειώσαμε το κάτω κομμάτι του οχήματος ήρθε η ώρα να τοποθετήσουμε την πλακέτα που θα ελέγχει τα 2 μοτέρ.

Επειδή το Arduino δεν μπορεί να ελέγξει αποτελεσματικά τα μοτέρ χρειάζεται να προσθέσουμε μια πλακέτα για να κάνει αυτή την δουλειά, αυτή η πλακέτα είναι η H-BRIDGE Keyes L298N Motor Shield η οποία φαίνεται παρακάτω.



Εικόνα 4.2

Πτυχιακή Εργασία Τμήματος Μηχανικών Πληροφορικής Καρυωτάκης Βασίλειος

Η μονάδα οδήγησης Double H χρησιμοποιεί διπλό οδηγό πλήρους γέφυρας ST L298N, ένα ολοκληρωμένο μονολιθικό κύκλωμα σε πακέτα Multiwatt και PowerSO20 15 καναλιών. Πρόκειται για ένα υψηλής τάσης, υψηλού ρεύματος διπλό οδηγό πλήρους γέφυρας που έχει σχεδιαστεί για να δέχεται στάνταρ επίπεδα λογικής TTL και να κατευθύνει επαγωγικά φορτία όπως ηλεκτρονόμοι, ηλεκτρομαγνητικές βαλβίδες, DC και βηματικούς κινητήρες. Παρέχονται δύο εισοδοί ενεργοποίησης για να ενεργοποιούν ή να απενεργοποιούν τη συσκευή ανεξάρτητα από τα σήματα εισόδου.

A: Δείχνει το σημείο που συνδέουμε το ένα μοτέρ(A).

B: Δείχνει το σημείο που συνδέουμε το άλλο μοτέρ(B).

Γ:

- VMS: Δείχνει την θετική είσοδο(+) της εξωτερικής τροφοδοσίας μας.
- GND: Δείχνει την γείωση(-) της εξωτερικής τροφοδοσίας.
- 5V: Δείχνει μια έξοδο 5V την οποία άμα θέλουμε μπορούμε από εκεί να τροφοδοτήσουμε την πλακέτα Arduino από την εξωτερική τροφοδοσία που έχουμε βάλει στο L298N.

Z: Ένα LED το οποίο ενημερώνει άμα χρησιμοποιούμε την έξοδο 5V.

H: Τα παρακάτω PINS είναι PINS εισόδων που λαμβάνουν σήματα από την πλακέτα Arduino και είναι υπεύθυνα για την κίνηση του κάθε τροχού.

ENA + ENB : Αυτά τα 2 Pins χρησιμεύουν στο να αφήσουμε τα μοτέρ A και B αντίστοιχα να συνδεθούν με την τροφοδοσία(όχι να ενεργοποιηθούν), παίρνουν τιμές HIGH ή LOW για σύνδεση και αποσύνδεση αντίστοιχα.

IN1+IN2+IN3+IN4: Τα συγκεκριμένα PINS είναι για να κινούν τους τροχούς. Το κάθε μοτέρ έχει 2 PINS, στο μοτέρ A αντιστοιχούν τα PINS IN1 και IN2 και στο μοτέρ B τα PINS IN3 και IN4. Τα συγκεκριμένα PINS μπορούν να πάρουν τιμές HIGH ή LOW αλλά είναι συνετό να χρησιμοποιούμε PWM Inputs (Pulse Width Modulation) για να ελέγχουμε ακριβώς την ταχύτητα των μοτέρ, διότι άμα εισάγουμε HIGH ή LOW τα μοτέρ θα είναι ή φουλ ταχύτητα ή ανενεργά ενώ με PWM μπορούμε να ελέγξουμε ακριβώς την ταχύτητα με τιμές από 0 έως 255. Το κάθε ένα από τα 2 PINS που αντιστοιχεί στο κάθε μοτέρ το πρώτο θέτει το μοτέρ να κινείται μπροστά και το άλλο να κινείται αντίστροφα με την προϋπόθεση ότι μόνο ένα από τα 2 PINS είναι ενεργοποιημένο κάθε φορά στον παρακάτω πίνακα θα δείξουμε όλους τους συνδυασμούς για την κίνηση του οχήματος.

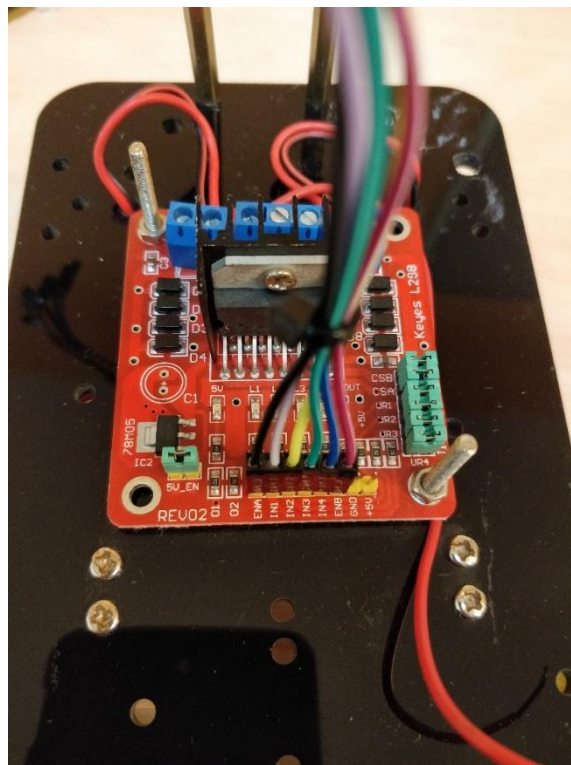
IN1+IN3=HIGH , IN2+IN4=LOW	Μπροστά
IN1=HIGH IN2+IN3+IN4=LOW	Δεξιά άμα έχουμε συνδέσει στο MotorA τον δεξιό τροχό, διαφορετικά αριστερά.
IN3=HIGH IN2+IN1+IN4=LOW	Αριστερά άμα έχουμε συνδέσει στο MotorB τον αριστερό τροχό, διαφορετικά δεξιά.
IN2+IN4=HIGH IN1+IN3=LOW	Όπισθεν
IN2=HIGH	Όπισθεν αριστερά(ισχύει ο ίδι-

IN1+IN3+IN4=LOW	ος κανόνας με το δεξιά-αριστερά παραπάνω)
IN4=HIGH IN1+IN3+IN2=LOW	Όπισθεν δεξιά(ισχύει ο ίδιος κανόνας με το δεξιά-αριστερά παραπάνω)

Σε όλους τους παραπάνω συνδυασμούς είναι προφανές όπως εξηγήσαμε παραπάνω ότι στην κάθε στιγμή τα ENA + ENB είναι ρυθμισμένα σε HIGH.

E: Δείχνει με ενδείξεις LED πιο IN χρησιμοποιείται την εκάστοτε στιγμή.

Η τοποθέτηση στο σασί φαίνεται παρακάτω:



Εικόνα 4.2.1

4.3 Η πλακέτα Arduino Uno R3

Το μυαλό του οχήματος θα το αναλάβει η πλακέτα Arduino Uno που είναι μικρή σε μέγεθος και κατάλληλη για την υλοποίησή μας, επίσης πολύ φθηνή και πάρα πολύ αναγνωρισμένη σε κατασκευές τέτοιου είδους, μια τέτοια πλακέτα φαίνεται παρακάτω:

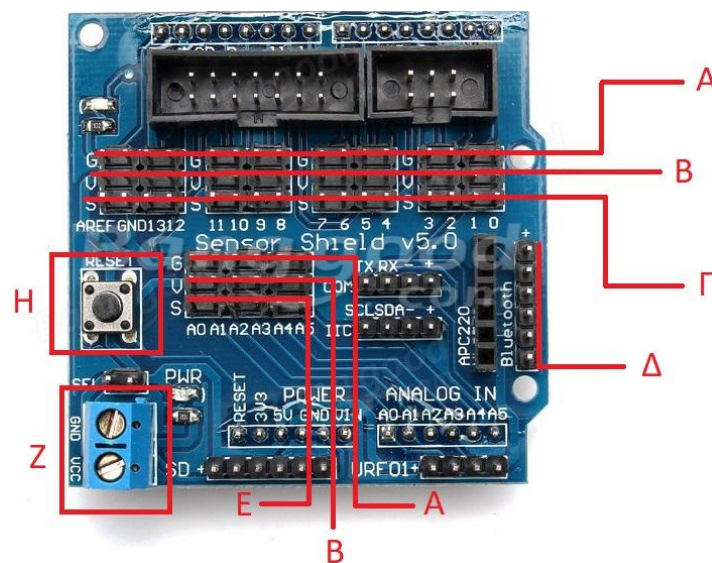


Εικόνα 4.3

Το Arduino είναι ένας μικροελεγκτής μονής πλακέτας, δηλαδή μια απλή μητρική πλακέτα ανοικτού κώδικα με ενσωματωμένο μικροελεγκτή και εισόδους/εξόδους, η οποία μπορεί να προγραμματιστεί με τη γλώσσα Wiring (ουσιαστικά πρόκειται για τη γλώσσα προγραμματισμού C++ και ένα σύνολο από βιβλιοθήκες, υλοποιημένες επίσης στην C++). Το Arduino μπορεί να χρησιμοποιηθεί για την ανάπτυξη ανεξάρτητων διαδραστικών αντικειμένων αλλά και να συνδεθεί με υπολογιστή μέσω προγραμμάτων σε Processing, Max/MSP, Pure Data, SuperCollider.

4.4 Arduino Sensor Shield v5.0

Για να αποφύγουμε την ταλαιπωρία με breadboard και τον κακό χαμό από καλώδια που μπερδεύονται μεταξύ τους χρησιμοποιήσαμε μια πλακέτα shield η οποία κουμπώνει πάνω στο Arduino και μας δίνει την δυνατότητα επέκτασης των Pins και προσθήκης περισσότερων επιλογών που θα χρειαστούν για την υλοποίηση του οχήματος, η οποία φαίνεται παρακάτω:



Εικόνα 4.4.1

Πτυχιακή Εργασία Τμήματος Μηχανικών Πληροφορικής Καρυωτάκης Βασίλειος

A και B: Δείχνουν τα Pins που είναι διαθέσιμα για τροφοδοσία 5V(A) και γείωσης(B). Να θυμίσουμε ότι η πλακέτα Arduino είχε 3 Pin γείωσης και ένα τροφοδοσίας 5V, που σημαίνει ότι άμα θέλαμε να τροφοδοτήσουμε πάνω από ένα όργανο μέσω της πλακέτας θα έπρεπε είτε να χρησιμοποιήσουμε Breadboard είτε να κάνουμε περιττές συνδέσεις μεταξύ των καλωδίων αλλά το shield μας δίνει 1 Pin τροφοδοσίας και ένα Pin γείωσης για κάθε ένα από τα 20 Pin σήματος.

Γ: Τα συγκεκριμένα Pins(0-13 και gnd και aref) αντιστοιχούν στα αντίστοιχα που βρίσκονται πάνω στην πλακέτα του Arduino.

Δ: Τα συγκεκριμένα Pins θα τα χρησιμοποιήσουμε αργότερα για να συνδέσουμε το Bluetooth module με το Arduino.

Ε: Τα A0-A5 αντιστοιχούν στις αναλογικές εισόδους οι οποίες βρίσκονται στην πλακέτα του Arduino.

Ζ: Είσοδος για εξωτερική τροφοδοσία του Sensor Shield.

Η: Push Button για reset της πλακέτας.

Μια πλακέτα Arduino μαζί με το sensor shield συνδεδεμένο φαίνεται παρακάτω:



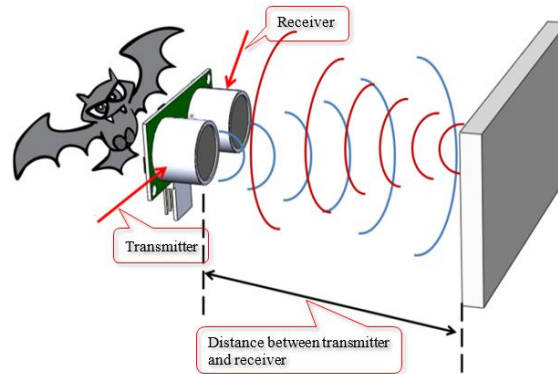
Εικόνα 4.4.2

4.5 Αισθητήρας Υπερήχων (Ultrasonic Module HC-SR04)

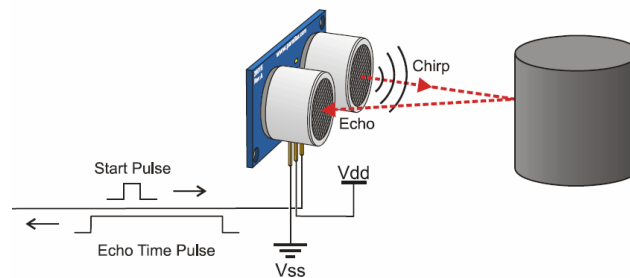


Εικόνα 4.5.1

Ο αισθητήρας υπερήχων χρησιμοποιεί την τεχνολογία σόναρ για να διευκρινίσει την απόσταση από ένα αντικείμενο, όπως ακριβώς κάνουν τα καράβια, οι νυχτερίδες και τα δελφίνια.



Εικόνα 4.5.2



Εικόνα 4.5.3

Προσφέρει εξαιρετική ανίχνευση εύρους χωρίς επαφή με υψηλή ακρίβεια και σταθερές αναγνώσεις. Μπορεί να ανιχνεύσει εύρος από 2 εκατοστά έως 400 εκατοστά και η λειτουργία του δεν επηρεάζεται από το φως του ήλιου ή τα σκούρα μαύρα αντικείμενα αλλά ακουστικά μαλακά αντικείμενα όπως τα ρούχα για παράδειγμα είναι δύσκολο να ανιχνευτούν γιατί αντί να ανακλαστεί, ο ήχος απορροφάται. Το σύστημα είναι πλήρες με υπερηχητικό πομπό και δέκτη.

Συνδεσμολογία:

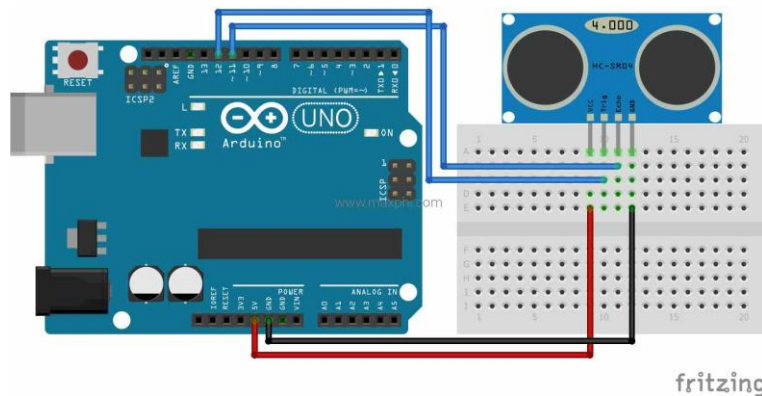
VCC: Τροφοδοσία 5V

TRIG: Η έξοδος υπερήχων

ECHO: Η είσοδος υπερήχων

GND: Γείωση

Στην παρακάτω εικόνα φαίνεται η σύνδεση του αισθητήρα με μια πλακέτα Arduino.



Εικόνα 4.5.4

Όπως βλέπουμε είναι συνδεδεμένα τα Pins Vcc και Gnd του Module στα 5V και Gnd αντίστοιχα στην πλακέτα του Arduino και το Echo Pin στο Pin 11 του Arduino το οποίο είναι PWM και το Trig Pin στο Pin 12 του Arduino.

Όπως είδαμε και παραπάνω το Echo Pin του αισθητήρα επειδή επιστρέφει μια τιμή και όχι απλά High ή Low πρέπει οπωσδήποτε να συνδεθεί με κάποιο PWM ψηφιακό Pin του Arduino ή με κάποιο αναλογικό.

4.6 Σερβοκινητήρας (Servo SG90)



Εικόνα 4.6.1

Ο αισθητήρας υπερήχων για τον οποίο μιλήσαμε στην προηγούμενη ενότητα, είναι προσαρμοσμένος επάνω σε ένα σερβοκινητήρα ο οποίος έχει δυνατότητα στρέψης εκατό ογδόντα μοιρών, ενενήντα αριστερά και ενενήντα δεξιά.

Με τον τρόπο αυτό μπορούμε κάθε φορά που το όχημά μας συναντάει εμπόδιο και σταματάει να μπορεί να ελέγχει δεξιά είτε αριστερά για άλλα τυχόν εμπόδια και να πράττει αναλόγως.

Παράδειγμα : Εφόσον το όχημά μας ανίχνευσε εμπόδιο μπροστά του και σταμάτησε στρίβει τον σερβοκινητήρα δεξιά ελέγχει άμα υπάρχει εμπόδιο, άμα δεν υπάρχει το όχημα στρίβει δεξιά και μετά από λίγο συνεχίζει ευθεία αλλιώς ο σερβοκινητήρας στρίβει αριστερά ελέγχει πάλι άμα υπάρχει εμπόδιο και πράττει αναλόγως.

Συνδεσμολογία:

Servo +5V(Κόκκινο καλώδιο): Arduino +5V

Servo Ground(Καφέ καλώδιο): Arduino Ground

Servo Signal(Πορτοκαλί καλώδιο): Κάποιο PWM Pin του Arduino



Εικόνα 4.6.2

4.7 Ir Receiver και Ir Transmitter

Όπως αναφέραμε και στην αρχή ο χρήστης θα έχει την δυνατότητα να χειρίζεται το όχημα μέσω τηλεκοντρόλ για αυτόν τον λόγο χρειαζόμαστε έναν πομπό που θα τον κρατάει ο χρήστης και θα δίνει τις εντολές και έναν δέκτη ο οποίος θα λαμβάνει τα σήματα του πομπού και θα βρίσκειτε πάνω στο όχημα.



Εικόνα 4.7.1

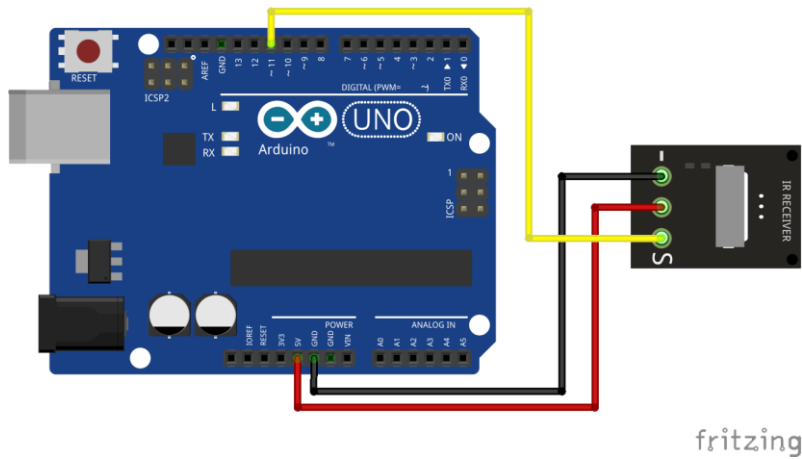


Εικόνα 4.7.2

Παραπάνω βλέπουμε τον δέκτη (Εικόνα 4.7.1) και δίπλα τον πομπό (Εικόνα 4.7.2), σαν πομπό μπορούμε να χρησιμοποιήσουμε οτιδήποτε εκπέμπει υπέρυθρο σήμα όπως για παράδειγμα τηλεκοντρόλ τηλεόρασης και air-condition.

Ο τρόπος που διαβάζουμε τις ενδείξεις του κάθε πομπού θα τον δούμε παρακάτω σε επόμενο κεφάλαιο.

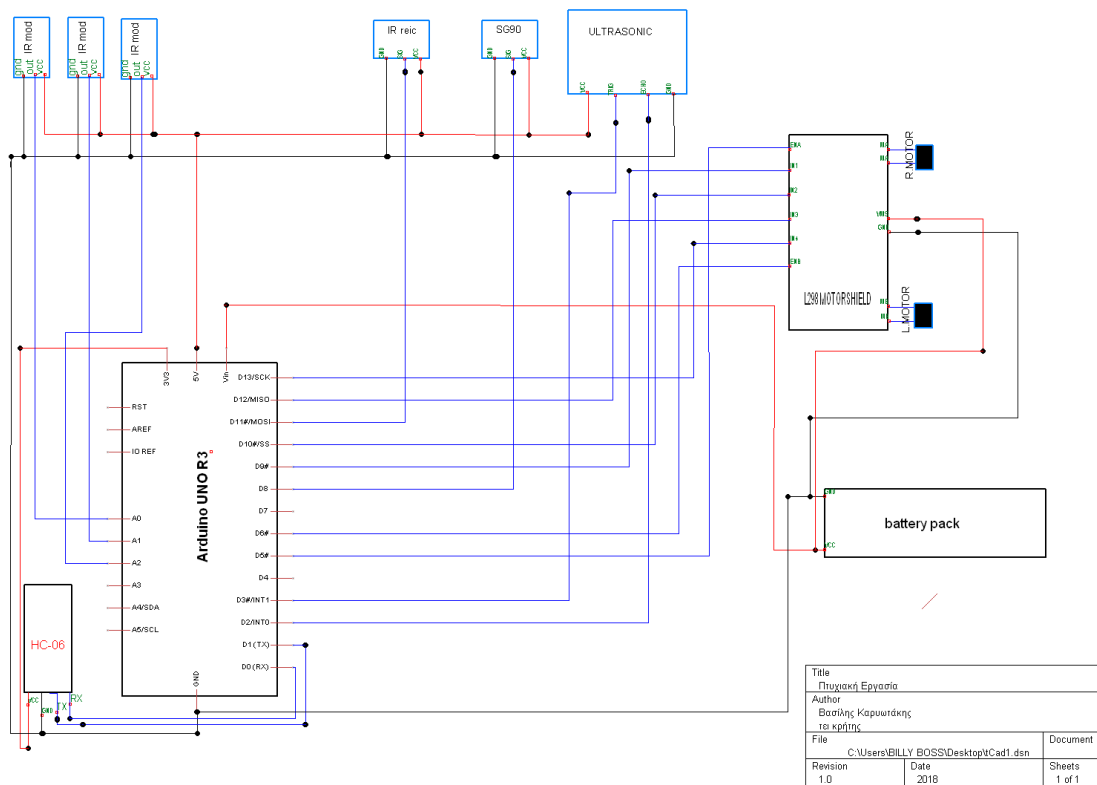
Σύνδεση του δέκτη στην πλακέτα Arduino:



fritzing

Εικόνα 4.7.3

Παρακάτω φαίνεται η συνδεσμολογία όλου του οχήματος και η τελική εικόνα:



Εικόνα 4.7.4

Κεφάλαιο 5: Προγραμματίζοντας τις λειτουργίες του οχήματος

Τα σενάρια που θα εξεταστούν στην παρούσα πτυχιακή όπως προ είπαμε είναι τα εξής:

Σενάριο 1: Αυτοκινούμενο όχημα στον χώρο με αποφυγή εμποδίων.

Σενάριο 2: Λειτουργία παρακολούθησης μαύρης γραμμής.

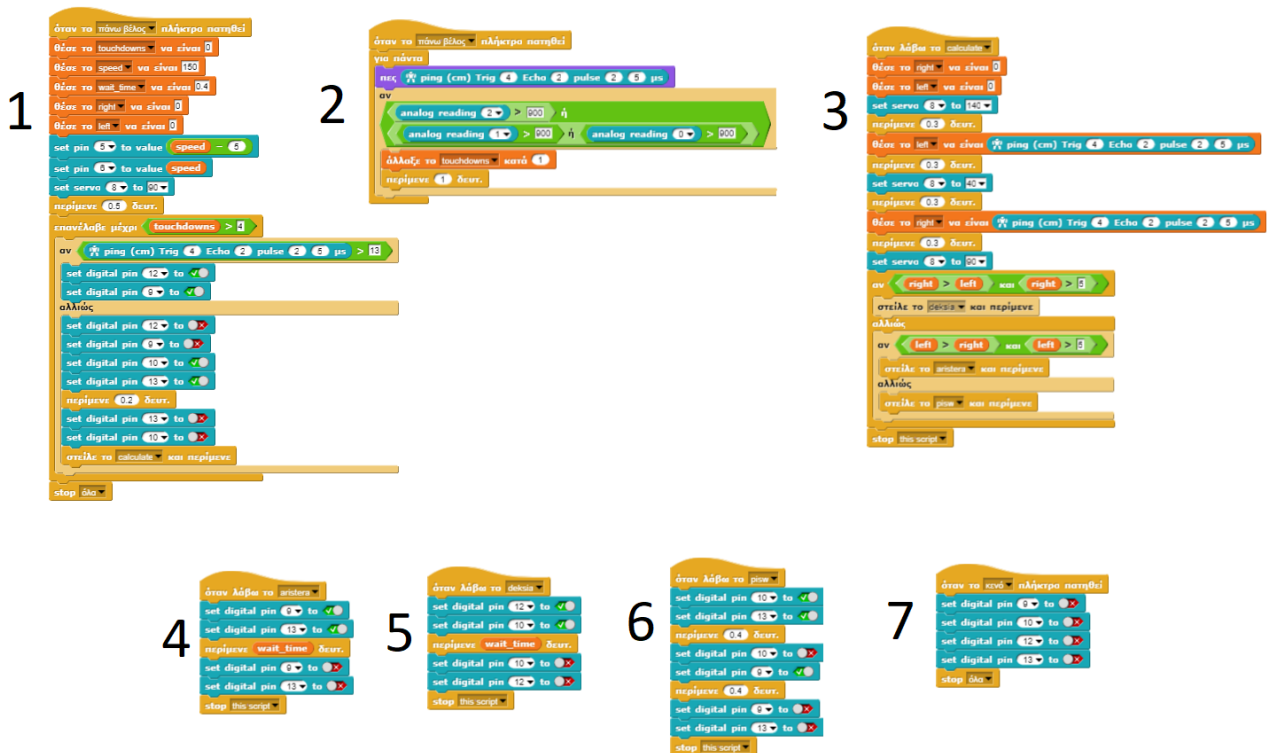
Σενάριο 3: Έλεγχος με χειριστήριο υπέρυθρων.

Σενάριο 4: Έλεγχος από υπολογιστή.

Σενάριο 1: Αυτοκινούμενο όχημα στο χώρο με αποφυγή εμποδίων

Σε αυτό το σενάριο θα προγραμματίσουμε το όχημα να μπορεί να κινείται στον χώρο αυτόνομα, να αποφεύγει εμπόδια με την χρήση του σόναρ και να υπολογίζει την απόσταση εμπόδιων είτε δεξιά είτε αριστερά και να παίρνει την σωστή απόφαση ως την κατεύθυνση που πρέπει να ακολουθήσει.

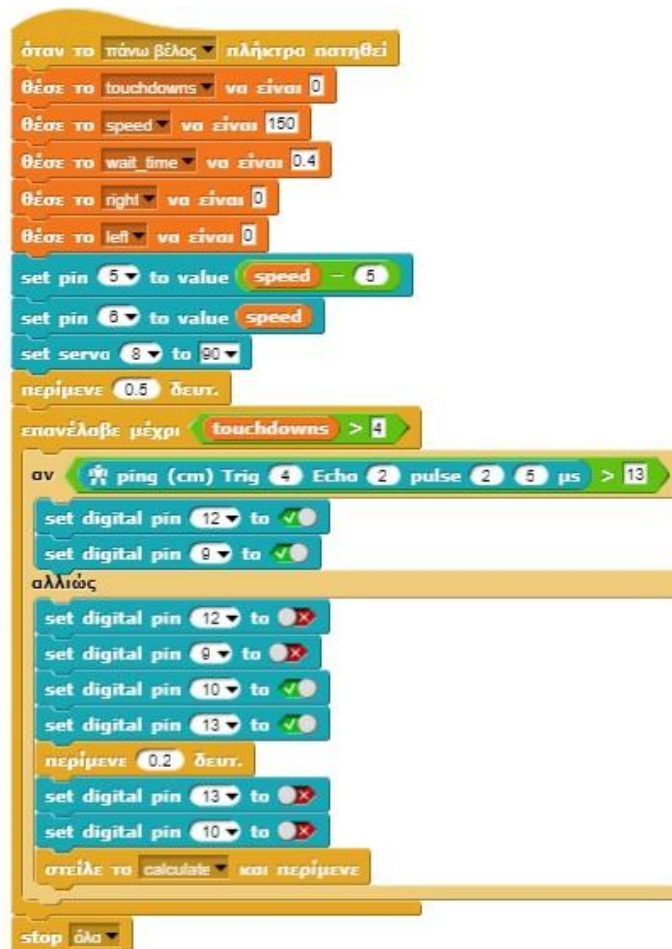
Στην παρακάτω εικόνα βλέπουμε το συγκεκριμένο σενάριο.



Εικόνα 5.1

Κομμάτι 1:

1



Εικόνα 5.2

Το συγκεκριμένο κομμάτι του προγράμματος είναι ουσιαστικά η main του προγράμματος, δηλαδή το κύριο μέρος του προγράμματος εκεί γίνονται οι αρχικοποιήσεις των μεταβλητών που θα χρησιμοποιηθούν στο πρόγραμμα όπως επίσης είναι το κομμάτι που ξεκινάει πρώτο και μόλις σταματήσει σταματάει και όλο το πρόγραμμα.

Ας πάρουμε μια μια τις γραμμές του κώδικα:

1)Είναι ο τρόπος εκκίνησης του προγράμματος εμείς βάλαμε όταν πατηθεί το πάνω βελάκι του πληκτρολογίου ξεκινάει.

2-6) Γραμμή 2 έως 6 γίνεται η αρχικοποίηση των μεταβλητών που θα χρησιμοποιηθούν.

7-8) Στην 7 και 8 δίνουμε τιμή στα πίν 5 και 6 του Arduino στα οποία είναι συνδεδεμένα τα EnA και EnB από το Motor Shield που όπως προείπαμε είναι υπεύθυνα για την ταχύτητα των τροχών. Ο λόγος που στο πιν 5 έχει μπει η μεταβλητή με την ταχύτητα να μειώνεται κατά 5 είναι διότι λόγω κατασκευαστικού λάθους ο ένας τροχός γέρνει λίγο προς τα έξω με αποτέλεσμα το όχημα να παρεκκλίνει της πορείας του όταν είναι να πάει ευθεία, και με αυτόν τον τρόπο μειώνοντας δηλαδή λίγο την ταχύτητα του μοτέρ να διορθώνεται η απόκλιση που υπάρχει.

9) Θέτουμε το Servo στις 90 μοίρες, δηλαδή στο κέντρο αφού το εύρος του Servo είναι από 0 έως 180.

10) Δίνουμε ένα μικρό delay 0,5 δευτερόλεπτα.

11) Εδώ ξεκινάει η επανάληψη, επειδή δεν θέλουμε το όχημα να δουλεύει επ άπειρον μαζί με την αποφυγή εμποδίων έχουμε προσαρμόσει την λειτουργία να ανιχνεύει σημεία που έχουμε τοποθετήσει στο πάτωμα με μαύρη ταινία μέσω των ανιχνευτών που έχει από κάτω και όταν περάσει πάνω από 4 σημεία σταματάει.

12) Εδώ έχουμε μια If (αν) στην οποία ελέγχουμε άμα το σόναρ εντοπίζει κάποιο εμπόδιο, εμπόδιο θεωρείται ότι εντοπίζει σε απόσταση μικρότερη ή ίση από 13 εκατοστά.

13-14) Ενεργοποιούνται τα πίν 12 και 13 τα οποία είναι συνδεδεμένα στο IN1 και IN3 στο motorshield αντίστοιχα, δηλαδή τα μοτέρ ξεκινάνε και το όχημα πάει προς τα εμπρός. Το όχημα συνεχίζει την ευθεία πορεία του μέχρι να εντοπίσει ένα αντικείμενο.

15-16) Μόλις εντοπιστεί το εμπόδιο το πρώτο πράγμα που κάνει το πρόγραμμα είναι να κλείσει τα μοτέρ έπειτα

16-17) Ανοίγει τα μοτέρ αντίστροφα από ότι πριν, δηλαδή όπισθεν για πολύ λίγο διάστημα

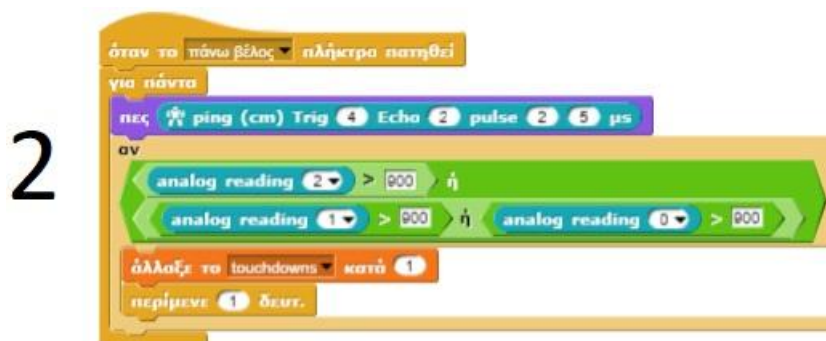
18) 0,2 δευτερόλεπτα, ο λόγος που γίνεται αυτό είναι για να έχουμε ακαριαίο σταμάτημα ώστε από την ταχύτητα να μην προσκρούσει το όχημα στο εμπόδιο που έχει μπροστά του

19-20) Έπειτα κλείνουμε πάλι τα μοτέρ.

21) Εδώ δίνεται η εντολή να ξεκινήσει ένα δευτερεύον πρόγραμμα με την ονομασία calculate το οποίο θα αναλυθεί μετά.

22) Δίνεται εντολή να σταματήσουν όλα τα προγράμματα.

Κομμάτι 2:



Εικόνα 5.3

1) Όπως και στο προηγούμενο το συγκεκριμένο κομμάτι κώδικα ξεκινάει όταν πατηθεί το πάνω βέλος, μαζί δηλαδή με το κυρίως πρόγραμμα.

2) Ξεκινάει μια επανάληψη για όσο τρέχει το πρόγραμμα.

3) Με την εντολή «πες» λέμε στο Snaρ να δείχνει στην οθόνη δεξιά τις μετρήσεις που λαμβάνει από σόναρ.

4) Εντολή αν.

5) Εδώ ελέγχουμε τους αισθητήρες υπέρυθρων που βρίσκονται κάτω από όχημα αν ένας από τους τρεις ανιχνεύσει κάτι, σε περίπτωση που ανιχνεύσει κάτι

6) Αλλάζει την τιμή της μεταβλητής touchdowns κατά +1

7) Έπειτα περιμένει 1 δευτερόλεπτο για να μπορέσει να ξανά μετρήσει ώστε να αποφευχθεί να διαβαστεί το ίδιο σημάδι 2 φορές στον ίδιο χρόνο.

Κομμάτι 3:



Εικόνα 5.4

1) Το κομμάτι αυτό του προγράμματος ξεκινάει όταν θα δοθεί η εντολή calculate από το main πρόγραμμα που δείξαμε παραπάνω.

2-3) Θέτει τις μεταβλητές right και left ίσες με το 0

4) Βάζει το σέρβο στην θέση 140 μοιρών, δηλαδή στα αριστερά.

5) Περιμένει 0,3 δευτερόλεπτα.

6) Έπειτα παίρνει μια μέτρηση από σόναρ για πιθανά εμπόδια στα αριστερά και μετά την αποθηκεύει στην μεταβλητή left

7) Περιμένει πάλι 0,3 δευτερόλεπτα.

8)Γυρνάει το σέρβο στα δεξιά.

9) Περιμένει πάλι 0,3 δευτερόλεπτα.

10)Παίρνει μέτρηση από σόναρ και την αποθηκεύει στην μεταβλητή right.

11) Περιμένει πάλι 0,3 δευτερόλεπτα.

12)Το σέρβο γυρνάει στην αρχική του θέση, στο κέντρο.

13)Εδώ ελέγχουμε τις 2 μεταβλητές right και left για να κρίνουμε σε πια από τις 2 κατεύθυνσης είναι πιο ελεύθερη από εμπόδια, επίσης εκτός από το πια μεταβλητή είναι μεγαλύτερη πρέπει να ελέγξουμε άμα η ίδια η μεταβλητή είναι μεγαλύτερη από 5 εκατοστά, για να αποφύγουμε την περίπτωση να έχουμε και στις 2 κατεύθυνσης κάποιο εμπόδιο πολύ κοντά και παρόλα αυτά να στρίψει.

14)Αμα ισχύει το right μεγαλύτερο του left και right μεγαλύτερο του 5 στέλνει την εντολή deksia και περιμένει αλλιώς

15)Ελέγχει την αντίθετη περίπτωση με την παραπάνω, για το left δηλαδή και άμα ισχύει

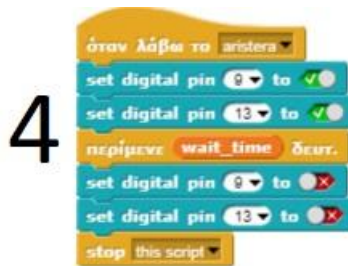
16) στέλνει aristera και περιμένει

17) Άμα τώρα δεν ισχύει καμία από τις 2 περιπτώσεις

18)Στέλνει την εντολή risw και περιμένει

19) Τέλος σταματάει το script.

Κομμάτι 4:



Εικόνα 5.5

1)Όταν λάβει την εντολή aristera ξεκινάει.

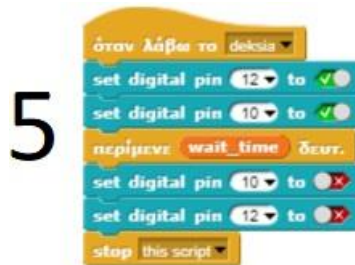
2-3)Ενεργοποιεί τα πιν 9 και 13, δηλαδή ο δεξιός τροχός κινείται προς τα εμπρός και ο αριστερός προς τα πίσω. Ο λόγος που γίνεται αυτό είναι γιατί έτσι έχουμε ακαριαίο στρίψιμο όπως ακριβώς κάνουν τα τανκς.

4)Περιμένει ένα χρονικό διάστημα που έχουμε ορίσει στο main πρόγραμμα.

5-6)Κλείνει τα μοτέρ και τέλος

7)Σταματάει το script.

Κομμάτι 5:



Εικόνα 5.6

1)Όταν λάβει την εντολή deksia ξεκινάει.

2-3)Ενεργοποιεί τα πιν 12 και 10, δηλαδή ο αριστερός τροχός κινείται προς τα εμπρός και ο δεξιός προς τα πίσω.

4)Περιμένει ένα χρονικό διάστημα που έχουμε ορίσει στο main πρόγραμμα.

5-6)Κλείνει τα μοτέρ και τέλος

7)Σταματάει το script.

Κομμάτι 6:



Εικόνα 5.7

1)Όταν λάβει την εντολή pisw ξεκινάει.

2-3)Ενεργοποιεί τα πιν 10 και 13, δηλαδή και τα 2 μοτέρ πάνε όπισθεν

4) Περιμένει 0.4 δευτερόλεπτα

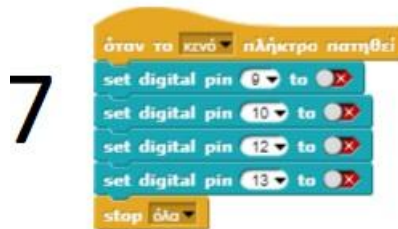
5-6) Κλείνει το ένα μοτέρ ενώ ταυτόχρονα αφήνει το άλλο να δουλεύει έτσι ώστε να κάνει το όχημα μια απότομη στροφή. Ο λόγος που γίνεται αυτό είναι για να αλλάξουμε στο όχημα τελείως την κατεύθυνση του

7) Περιμένει 0.4 δευτερόλεπτα

8-9)Κλείνει τα μοτέρ και τέλος

10)Σταματάει το script.

Κομμάτι 7:



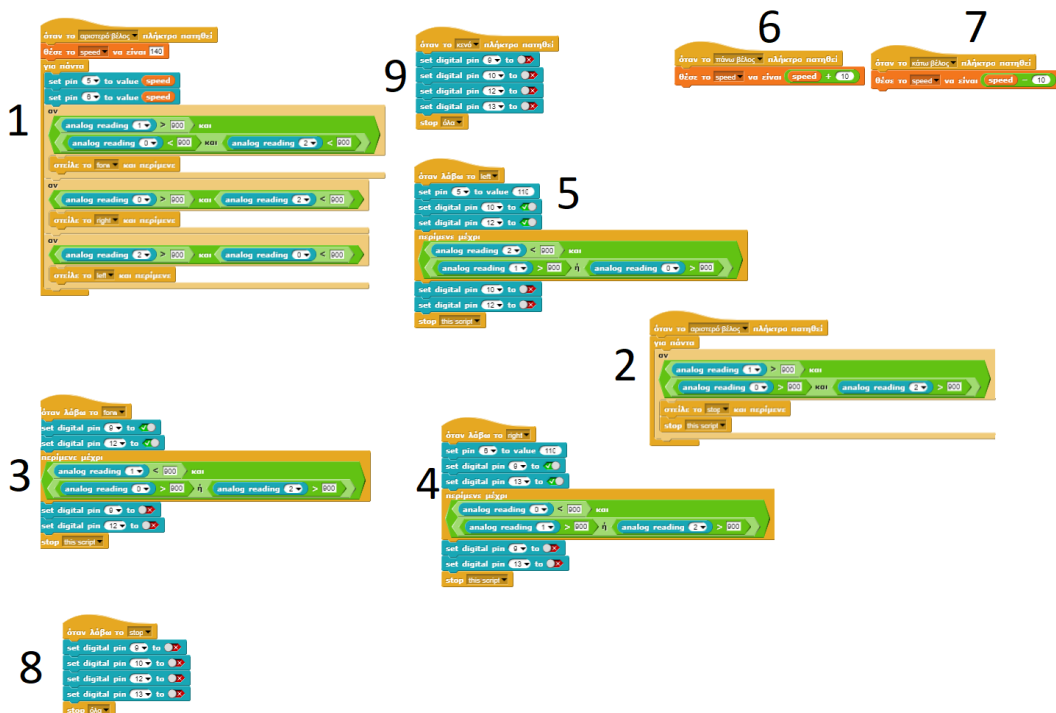
Εικόνα 5.8

Αυτό δημιουργήθηκε σε περίπτωση κάποιου σφάλματος και άμα πατηθεί το πλήκτρο space κλείνουν όλα τα μοτέρ.

Σενάριο 2: Λειτουργία παρακολούθησης μαύρης γραμμής

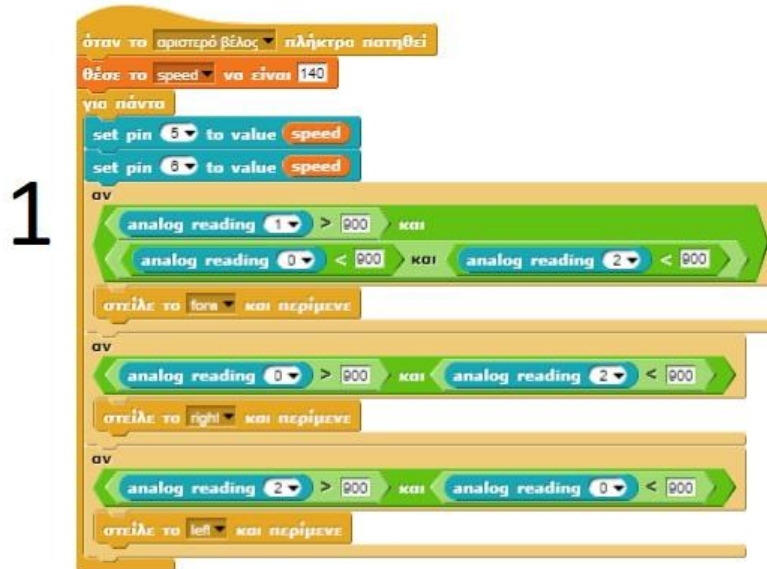
Στο παρόν σενάριο βάζουμε το όχημα να ακολουθεί μια μαύρη γραμμή την οποία έχουμε φτιάξει στο πάτωμα με μαύρη ταινία. Αυτή η λειτουργία χρησιμοποιεί τους 3 αισθητήρες υπέρυθρων που είναι τοποθετημένοι στο κάτω μέρος του οχήματος.

Η συγκεκριμένη λειτουργία φαίνεται στην παρακάτω εικόνα.



Εικόνα 5.9

Κομμάτι 1:



Εικόνα 5.10

- 1) Όταν πατηθεί το αριστερό κουμπί ξεκινάει το main πρόγραμμα
- 2) Θέτουμε την μεταβλητή speed σε 140.
- 3) Ξεκινάει μια επανάληψη.
- 4-5) Θέτουμε τα πίν 5 και 6 ίσα με την μεταβλητή speed.
- 6) Ελέγχουμε αν ο μεσαίος αισθητήρας ανιχνεύει την γραμμή και οι άλλοι 2 δεν την ανιχνεύουν, άμα ισχύει αυτό τότε
- 7) στέλνουμε την εντολή forw και περιμένουμε.
- 8) Ελέγχουμε αν ο αριστερός αισθητήρας ανιχνεύει μαύρη γραμμή και ο δεξιός όχι, τότε
- 9) Στέλνουμε την εντολή right και περιμένουμε.
- 10) Ελέγχουμε αν ο δεξιός αισθητήρας ανιχνεύει μαύρη γραμμή και ο αριστερός όχι, τότε
- 9) Στέλνουμε την εντολή left και περιμένουμε.

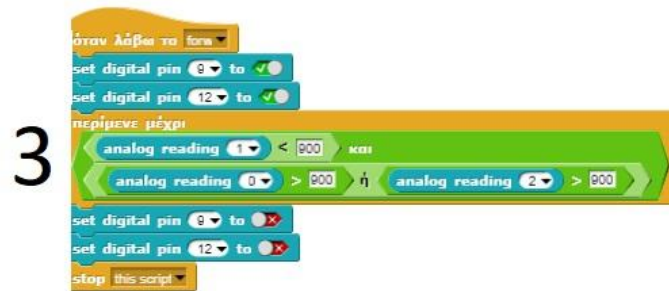
Κομμάτι 2:



Εικόνα 5.11

- 1) Ξεκινάει μαζί με το κυρίως πρόγραμμα
- 2) Ξεκινάει μία επανάληψη
- 3) Αν και οι 3 αισθητήρες ακουμπήσουν πάνω σε μαύρη γραμμή
- 4) στέλνει stop
- 5) Σταματάει να εκτελείται αυτό το script.

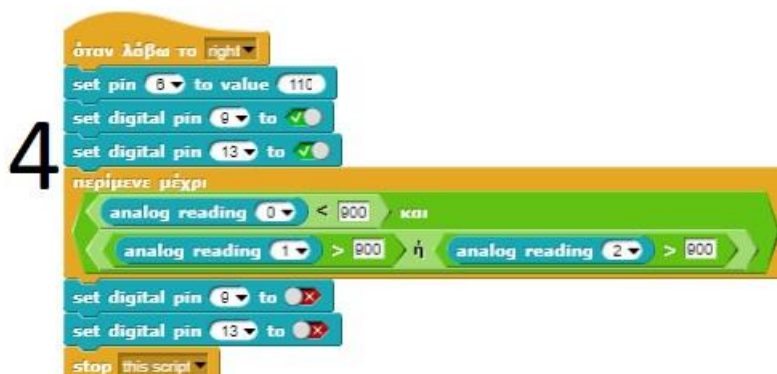
Κόμματι 3:



Εικόνα 5.12

- 1) Όταν λάβει την εντολή forw ξεκινάει
- 2-3) Ενεργοποιεί τα πιν 9 και 12, δηλαδή ξεκινάει και πάει προς τα εμπρός
- 4) Περιμένει μέχρι ο μεσαίος αισθητήρας να βγει εκτός της μαύρης γραμμής και ο αριστερός ή ο δεξιός αισθητήρας να ανιχνεύσει την μαύρη γραμμή.
- 5-6) Απενεργοποιεί τα πιν 9 και 12
- 7) Σταματάει να εκτελείται αυτό το script.

Κομμάτι 4:



Εικόνα 5.13

- 1) Όταν λάβει την εντολή right ξεκινάει
- 2) Αλλάζει την ταχύτητα του αριστερού μοτέρ σε 110, χαμηλότερά δηλαδή από ότι ήταν αρχικά. Ο λόγος που γίνεται αυτό είναι διότι θέλουμε όταν στρίψει ο ένας τροχός να έχει αντίθετη κατεύθυνση για να μπορεί το όχημα να στρίψει ακαριαία.

3-4) Ενεργοποιεί τα πιν 9 και 13

5) Περιμένει μέχρι ο δεξιός αισθητήρας να βγει εκτός της μαύρης γραμμής και ο μεσαίος ή ο αριστερός αισθητήρας να ανιχνεύσει την μαύρη γραμμή.

6-7) Απενεργοποιεί τα πιν 9 και 13

8) Σταματάει να εκτελείται αυτό το script

Κομμάτι 5:



Εικόνα 5.14

1) Όταν λάβει την εντολή left ξεκινάει

2) Αλλάζει την ταχύτητα του δεξιού μοτέρ σε 110, χαμηλότερα δηλαδή από ότι ήταν αρχικά. Ο λόγος που γίνεται αυτό είναι ο ίδιος όπως στο προηγούμενο κομμάτι.

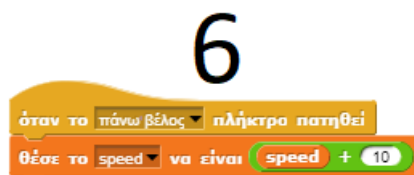
3-4) Ενεργοποιεί τα πιν 10 και 12

5) Περιμένει μέχρι ο αριστερός αισθητήρας να βγει εκτός της μαύρης γραμμής και ο μεσαίος ή ο δεξιός αισθητήρας να ανιχνεύσει την μαύρη γραμμή.

6-7) Απενεργοποιεί τα πιν 10 και 12

8) Σταματάει να εκτελείται αυτό το script

Κομμάτι 6:



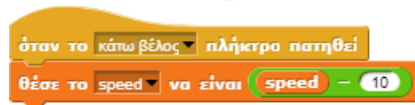
Εικόνα 5.15

1) Όταν το πάνω βέλος πατηθεί

2) Αλλάζει η μεταβλητή speed κατά +10

Κομμάτι 7:

7

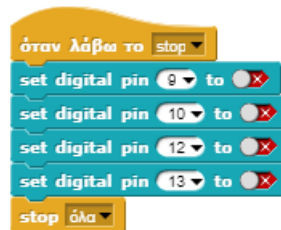


Εικόνα 5.16

- 1) Όταν το κάτω βέλος πατηθεί
- 2) Αλλάζει η μεταβλητή speed κατά -10

Κόμματι 8:

8

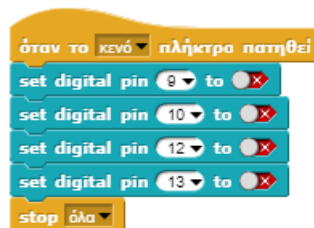


Εικόνα 5.17

- 1) Όταν λάβει την εντολή stop
- 2-5) Κλείνει τα πιν 9,10,12 και 13
- 6) Σταματάει να εκτελείται αυτό το script

Κομμάτι 9:

9



Εικόνα 5.18

Αυτό δημιουργήθηκε σε περίπτωση κάποιου σφάλματος και άμα πατηθεί το πλήκτρο space κλείνουν όλα τα μοτέρ.

Σενάριο 3: Έλεγχος με χειριστήριο υπερύθρων

Στο παρόν σενάριο ελέγχουμε το όχημα με ένα τηλεχειριστήριο υπέρυθρων. Αυτό μπορεί να είναι είτε αυτό που δείξαμε παραπάνω είτε οποιοδήποτε τηλεχειριστήριο από τηλεόρασης μέχρι air-condition.

Πτυχιακή Εργασία Τμήματος Μηχανικών Πληροφορικής Καρυωτάκης Βασίλειος

Όλα τα τηλεχειριστήρια υπέρυθρων λειτουργούν με τον ίδιο τρόπο, στέλνουν ένα δεκαεξαδικό κωδικό στον δέκτη και άμα αυτός τον αναγνωρίσει πράττει αναλόγως, έτσι και εδώ για να χρησιμοποιήσουμε αυτή την λειτουργία πρέπει πρώτα να βρούμε τους 16δικούς κωδικούς κάθε κουμπιού του χειριστηρίου μας.

Πρώτα συνδέουμε το arduino με τον υπολογιστή μας, έπειτα ανοίγουμε το arduino IDE και ανεβάζουμε τον παρακάτω κώδικα στην πλακέτα.

```
#include <IRremote.h>
const int RECV_PIN = 11;
IRrecv irrecv(RECV_PIN);
decode_results results;
void setup(){
  Serial.begin(9600);
  irrecv.enableIRIn();
  irrecv.blink13(true);
}
void loop(){
  if (irrecv.decode(&results)){
    Serial.println(results.value, HEX);
    irrecv.resume();
  }
}
```

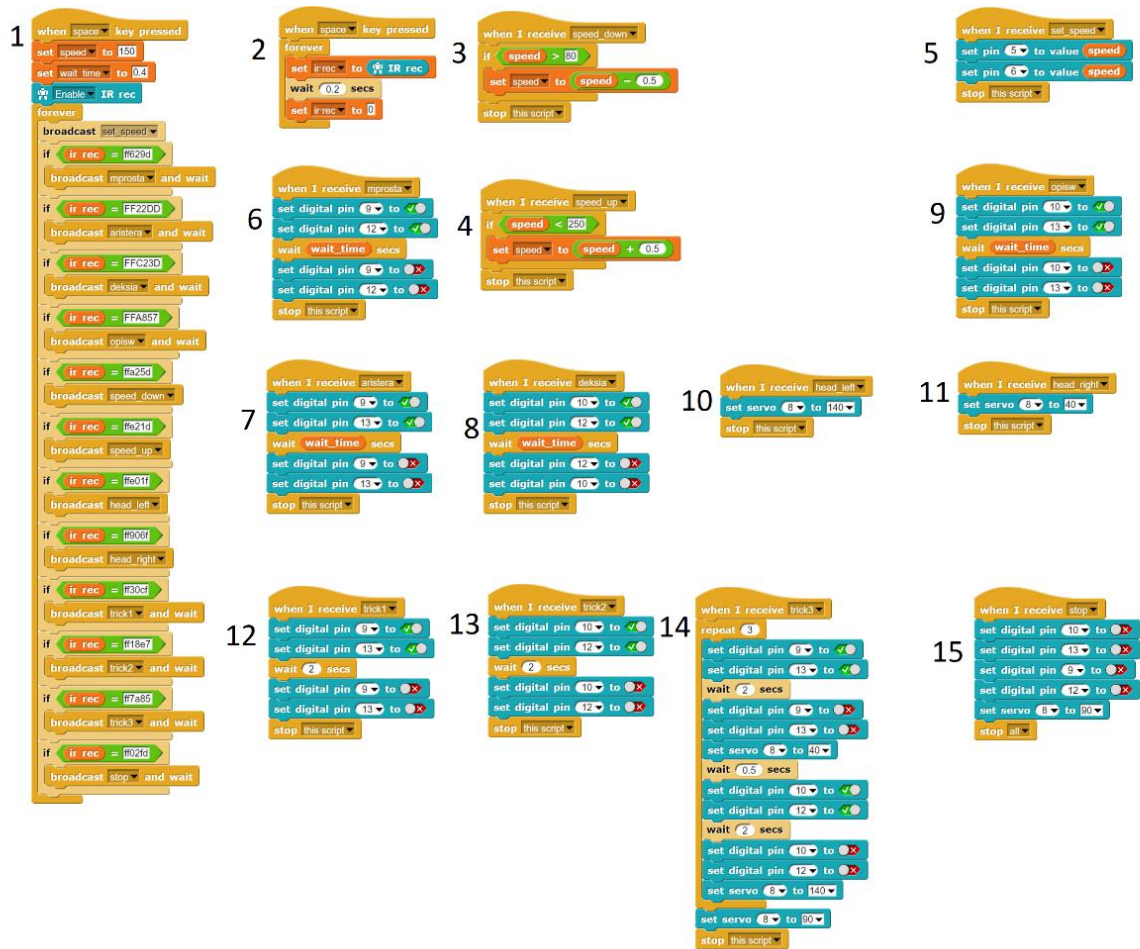
Έπειτα ανοίγουμε το Serial Monitor που βρίσκεται στο tools και ξεκινάμε να πατάμε ένα ένα τα κουμπιά και να καταγράφουμε τους κωδικούς που μας δείχνει. Παρακάτω φαίνεται ο πίνακας καταγραφής

Key	Code
CH-	FFA25D
CH	FF629D
CH+	FFE21D
<<	FF22DD
>>	FF02FD
>	FFC23D
-	FFE01F

+	FFA857
EQ	FF906F
100+	FF9867
200+	FFB04F
0	FF6897
1	FF30CF
2	FF18E7
3	FF7A85
4	FF10EF
5	FF38C7
6	FF5AA5
7	FF42BD
8	FF4AB5
9	FF52AD

Με βάση αυτόν τον πίνακα μπορούμε πλέον να φτιάξουμε το πρόγραμμα μας.

Πριν κλείσουμε το IDE πρέπει να ξανανεβάσουμε το πρωτόκολλο S5A όπως κάναμε και πιο πριν. Η λειτουργία IR remote φαίνεται παρακάτω:



Εικόνα 5.19

Κόμματι 1:



Εικόνα 5.20

Όταν το space πατηθεί ξεκινάει το main πρόγραμμα

- 1) Θέτει την μεταβλητή speed σε αρχική τιμή ίση με 150
- 2) Θέτει την μεταβλητή wait_time σε 0.4 δευτερόλεπτα
- 3) Ενεργοποιεί το IR
- 4) Ξεκινάει την επανάληψη

- 5) Στέλνει την εντολή set_speed
- 6) Άμα ο κωδικός που στέλνουμε είναι ff629d στέλνει την εντολή mprosta και περιμένει
- 7) Άμα ο κωδικός που στέλνουμε είναι FF22DD στέλνει την εντολή aristera και περιμένει
- 8) Άμα ο κωδικός που στέλνουμε είναι FFC23D στέλνει την εντολή deksia και περιμένει
- 9) Άμα ο κωδικός που στέλνουμε είναι FFA857 στέλνει την εντολή opisw και περιμένει
- 10) Άμα ο κωδικός που στέλνουμε είναι ffa25d στέλνει την εντολή speed_down και περιμένει
- 11) Άμα ο κωδικός που στέλνουμε είναι ffe21d στέλνει την εντολή speed_up και περιμένει
- 12) Άμα ο κωδικός που στέλνουμε είναι ffe01f στέλνει την εντολή head_left και περιμένει
- 13) Άμα ο κωδικός που στέλνουμε είναι ff906f στέλνει την εντολή head_right και περιμένει
- 14) Άμα ο κωδικός που στέλνουμε είναι ff30cf στέλνει την εντολή trick1 και περιμένει
- 15) Άμα ο κωδικός που στέλνουμε είναι ff18e7 στέλνει την εντολή trick2 και περιμένει
- 16) Άμα ο κωδικός που στέλνουμε είναι ff7a85 στέλνει την εντολή trick3 και περιμένει
- 17) Άμα ο κωδικός που στέλνουμε είναι ff02fd στέλνει την εντολή stop και περιμένει

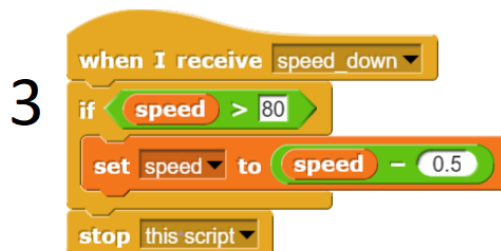
Κομμάτι 2:



Εικόνα 5.21

- 1)Όταν πατηθεί το space
- 2)Ξεκινάει μια επανάληψη για όσο τρέχει το main πρόγραμμα
- 3)Θέτει την μεταβλητή ir rec ίση με τον δεκαεξαδικό που στέλνουμε εκείνη την στιγμή
- 4)Περιμένει 0.2 seconds
- 5)Θέτει την μεταβλητή ir rec ίση με το 0

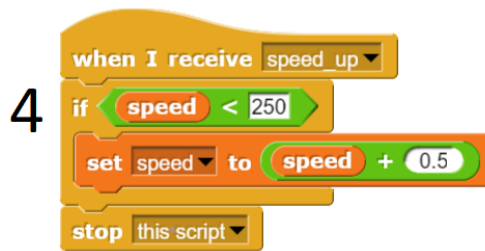
Κομμάτι 3:



Εικόνα 5.22

- 1)Όταν λάβει την εντολή speed_down
- 2)Άμα η μεταβλητή speed είναι μεγαλύτερη του 80 τότε
- 3)Θέτει την μεταβλητή speed ίση με την υπάρχουσα τιμή μείον 0.5
- 4)Σταματάει το script

Κομμάτι 4:



Εικόνα 5.23

- 1)Όταν λάβει την εντολή speed_up
- 2)Αμα η μεταβλητή speed είναι μικρότερη του 250 τότε
- 3)Θέτει την μεταβλητή speed ίση με την υπάρχουσα τιμή συν 0.5
- 4)Σταματάει το script

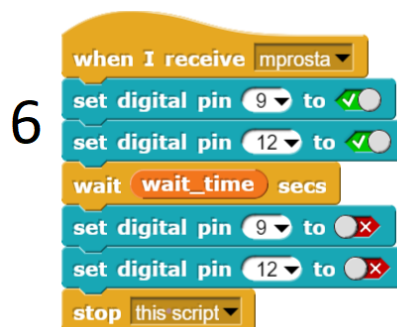
Κομμάτι 5:



Εικόνα 5.24

- 1)Όταν λάβει την εντολή set_speed
- 2-3)Θέτει τα πιν 5 και 6 ίσα με την μεταβλητή speed
- 4)Σταματάει αυτό το script.

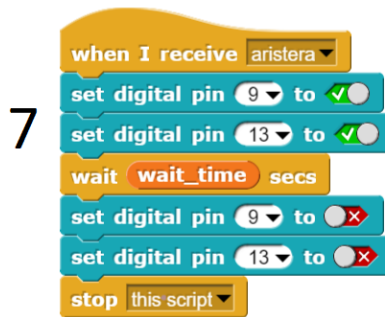
Κομμάτι 6:



Εικόνα 5.25

- 1)Όταν λάβει την εντολή mprosta
- 2-3)Ανοίγει τα πιν 9 και 12
- 4)Περιμένει για όσο έχουμε θέσει την μεταβλητή wait_time
- 5-6)Κλείνει τα πιν 9 και 12
- 7)Σταματάει αυτό το script

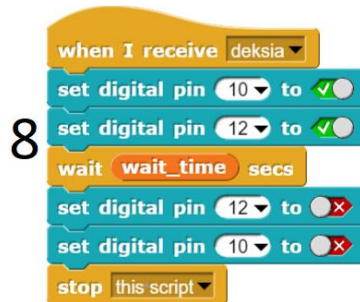
Κομμάτι 7:



Εικόνα 5.26

- 1)Όταν λάβει την εντολή aristera
- 2-3)Ανοίγει τα πιν 9 και 13
- 4) Περιμένει για όσο έχουμε θέσει την μεταβλητή wait_time
- 5)Κλείνει τα πιν 9 και 13
- 6)Σταματάει αυτό το script

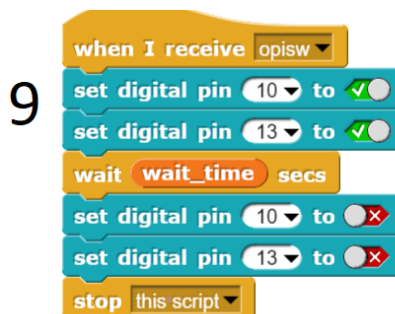
Κομμάτι 8:



Εικόνα 5.27

- 1)Όταν λάβει την εντολή deksia
- 2-3)Ανοίγει τα πιν 10 και 12
- 4) Περιμένει για όσο έχουμε θέσει την μεταβλητή wait_time
- 5)Κλείνει τα πιν 10 και 12
- 6)Σταματάει αυτό το script

Κομμάτι 9:



Εικόνα 5.28

- 1)Όταν λάβει την εντολή orisw
- 2-3)Ανοίγει τα πιν 10 και 13
- 4) Περιμένει για όσο έχουμε θέσει την μεταβλητή wait_time
- 5)Κλείνει τα πιν 10 και 13
- 6)Σταματάει αυτό το script

Κομμάτι 10:



Εικόνα 5.29

- 1)Όταν λάβει την εντολή head_left
- 2)Θέτει το servo στις 140 μοίρες
- 3)Σταματάει αυτό το script

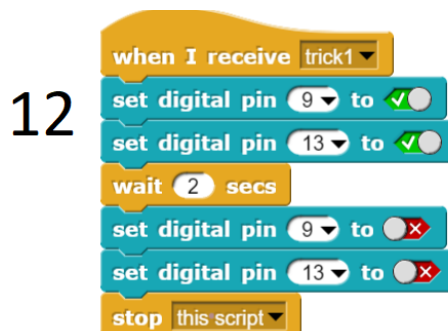
Κομμάτι 11:



Εικόνα 5.30

- 1)Όταν λάβει την εντολή head_right
- 2)Θέτει το servo στις 40 μοίρες
- 3)Σταματάει αυτό το script

Κομμάτι 12:



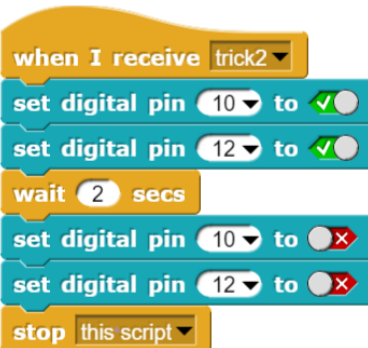
Εικόνα 5.31

- 1)Όταν λάβει την εντολή trick1
- 2-3)Ανοίγει τα πιν 9 και 13
- 4)Περιμένει για 2 δευτερόλεπτα

- 5)Κλείνει τα πιν 9 και 13
- 6)Σταματάει αυτό το script

Κομμάτι 13:

13



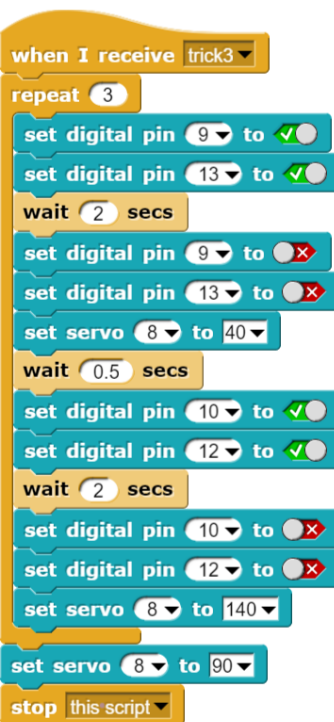
```
when I receive trick2
set digital pin 10 to ✓
set digital pin 12 to ✓
wait 2 secs
set digital pin 10 to ✗
set digital pin 12 to ✗
stop this script
```

Εικόνα 5.32

- 1)Όταν λάβει την εντολή trick2
- 2-3)Ανοίγει τα πιν 10 και 12
- 4)Περιμένει για 2 δευτερόλεπτα
- 5)Κλείνει τα πιν 10 και 12
- 6)Σταματάει αυτό το script

Κομμάτι 14:

14



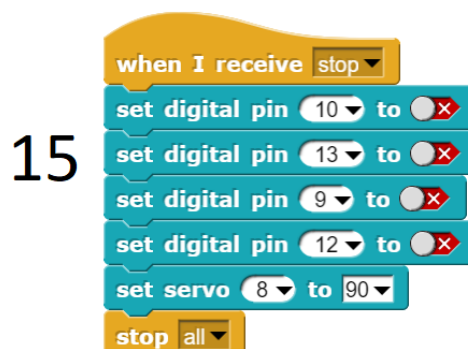
```
when I receive trick3
repeat 3
set digital pin 9 to ✓
set digital pin 13 to ✓
wait 2 secs
set digital pin 9 to ✗
set digital pin 13 to ✗
set servo 8 to 40
wait 0.5 secs
set digital pin 10 to ✓
set digital pin 12 to ✓
wait 2 secs
set digital pin 10 to ✗
set digital pin 12 to ✗
set servo 8 to 140
set servo 8 to 90
stop this script
```

Εικόνα 5.33

- 1)Όταν λάβει την εντολή trick3

- 2-3)Ανοίγει το πιν 9 και 13
- 4)Περιμένει 2 δευτερόλεπτα
- 5-6)Κλείνει τα πιν 9 και 13
- 7)Θέτει το servo στις 40 μοίρες
- 8)Περιμένει 0.5 δευτερόλεπτα
- 9-10)Ανοίγει τα πιν 10 και 12
- 11)Περιμένει 2 δευτερόλεπτα
- 12-13)Κλείνει τα πιν 10 και 12
- 14)Θέτει το servo στις 140 μοίρες
- 15)Θέτει το servo στις 40 μοίρες
- 16)Σταματάει αυτό το script

Κομμάτι 15:

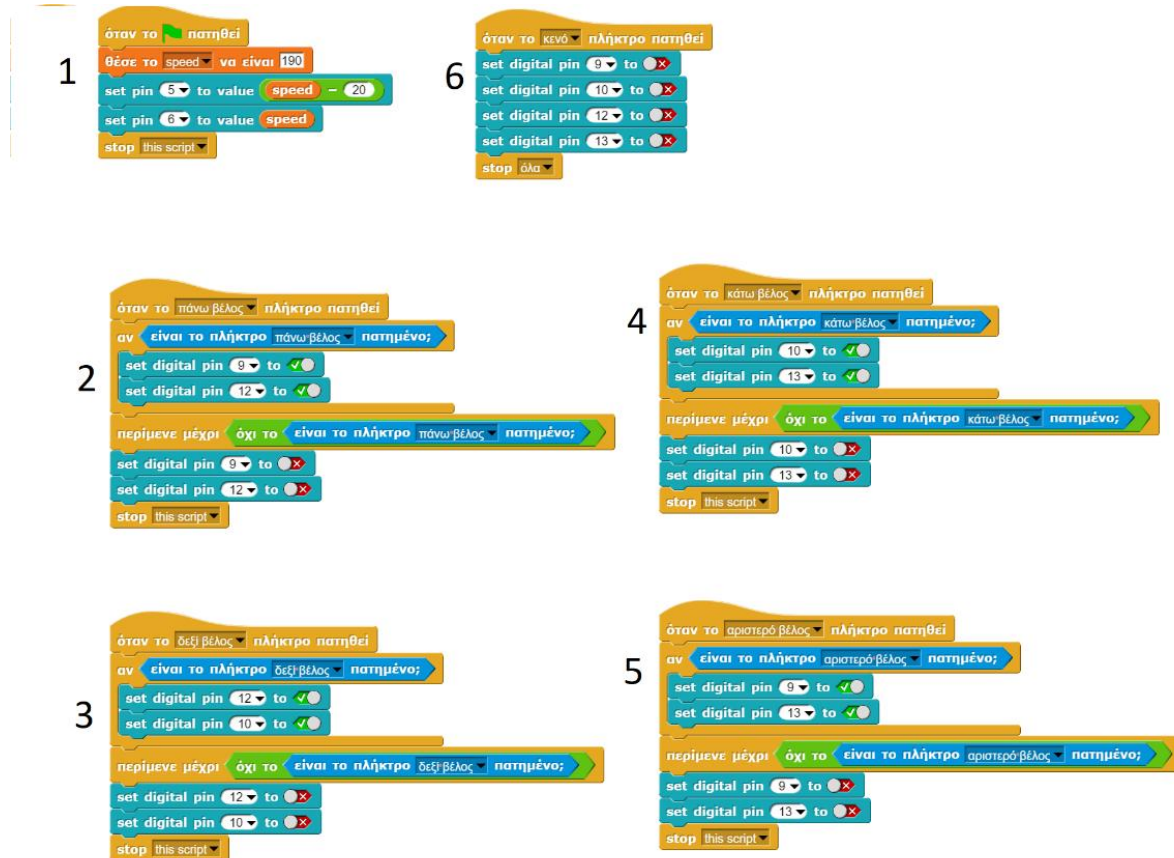


Εικόνα 5.34

- 1)Όταν λάβει την εντολή stop
- 2-3-4-5)Κλείνει τα πιν 10,13,9 και 12
- 6)Θέτει το servo στις 90 μοίρες
- 7)Σταματάει αυτό το script

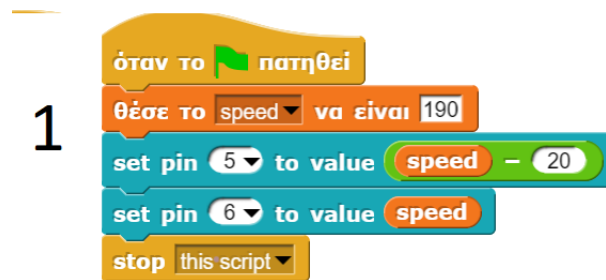
Σενάριο 4: Έλεγχος από υπολογιστή

Στο παρόν σενάριο ελέγχουμε το όχημα με τον υπολογιστή μέσω Bluetooth απ' τα βελάκια που έχει το πληκτρολόγιο.



Εικόνα 5.35

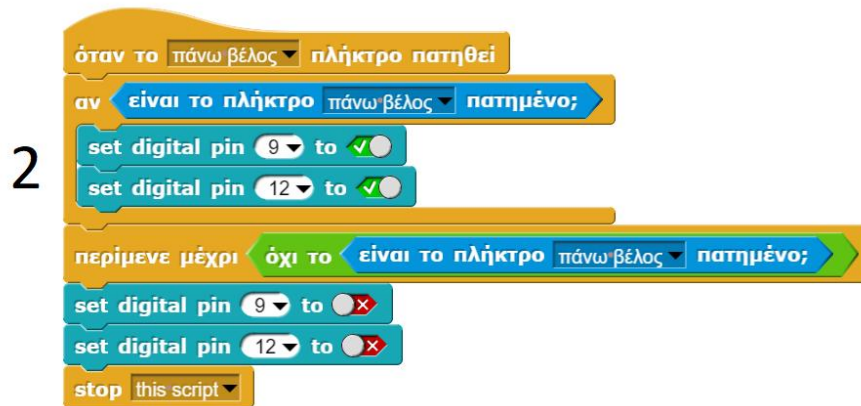
Κομμάτι 1:



Εικόνα 5.36

- 1) Όταν πατηθεί
- 2) Θέτει την μεταβλητή speed σε 190
- 3-4) Θέτει τα πιν 5 και 6 ίσα με την μεταβλητή speed
- 5) Σταματάει αυτό το script

Κομμάτι 2:



Εικόνα 5.37

- 1)Όταν το πάνω βέλος πατηθεί
- 2)Αν το πάνω βέλος εξακολουθεί να μένει πατημένο
- 3-4)Θέτει τα πιν 9 και 12 σε κατάσταση ON
- 5)Περιμένει μέχρι το πλήκτρο πάνω βέλος να σταματήσει να είναι πατημένο
- 6-7)Θέτει τα πιν 9 και 12 σε κατάσταση OFF
- 8)Σταματάει αυτό το script

Κομμάτι 3:



Εικόνα 5.38

- 1)Όταν το δεξί βέλος πατηθεί
- 2)Αν το δεξί βέλος εξακολουθεί να μένει πατημένο
- 3-4)Θέτει τα πιν 10 και 12 σε κατάσταση ON
- 5)Περιμένει μέχρι το πλήκτρο δεξί βέλος να σταματήσει να είναι πατημένο
- 6-7)Θέτει τα πιν 10 και 12 σε κατάσταση OFF
- 8)Σταματάει αυτό το script

Κομμάτι 4:



Εικόνα 5.39

- 1)Όταν το κάτω βέλος πατηθεί
- 2)Αν το κάτω βέλος εξακολουθεί να μένει πατημένο
- 3-4)Θέτει τα πιν 10 και 13 σε κατάσταση ON
- 5)Περιμένει μέχρι το πλήκτρο κάτω βέλος να σταματήσει να είναι πατημένο
- 6-7)Θέτει τα πιν 10 και 12 σε κατάσταση OFF
- 8)Σταματάει αυτό το script

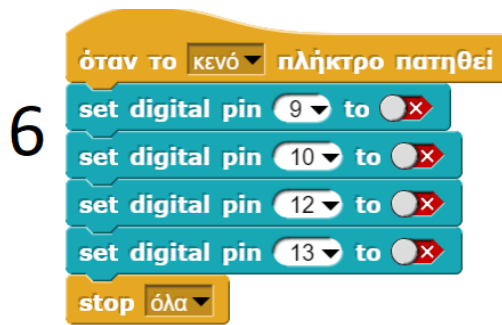
Κομμάτι 5:



Εικόνα 5.40

- 1)Όταν το αριστερό βέλος πατηθεί
- 2)Αν το αριστερό βέλος εξακολουθεί να μένει πατημένο
- 3-4)Θέτει τα πιν 9 και 13 σε κατάσταση ON
- 5)Περιμένει μέχρι το πλήκτρο αριστερό βέλος να σταματήσει να είναι πατημένο
- 6-7)Θέτει τα πιν 9 και 12 σε κατάσταση OFF
- 8)Σταματάει αυτό το script

Κομμάτι 6:



Εικόνα 5.41

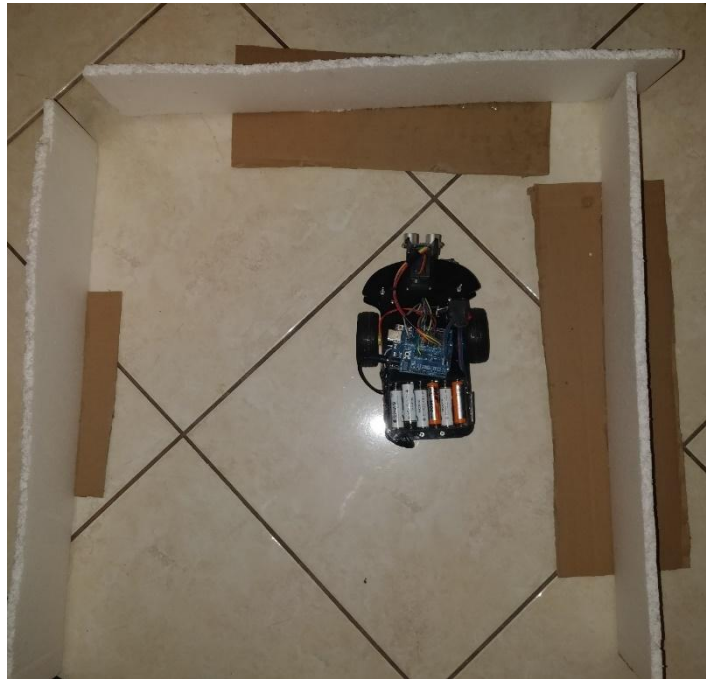
- 1)Όταν πατηθεί το κενό
- 2-3-4-5) Κλείνει τα πιν 9,10,12 και 13
- 6)Σταματάει αυτό το script

Κεφάλαιο 6: Οι λειτουργίες του οχήματος αναλυτικά.

Σενάριο 1: Αυτοκινούμενο όχημα στον χώρο με αποφυγή εμποδίων

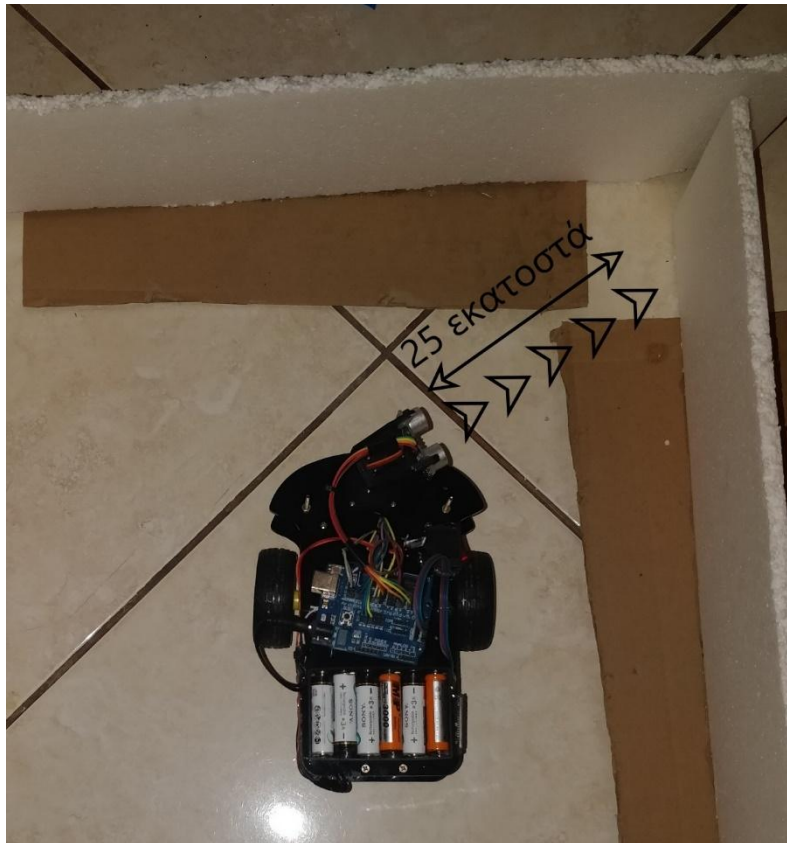
Όπως προ είπαμε το όχημα θα έχει την δυνατότητα να είναι αυτοκινούμενο στο χώρο ανιχνεύοντας τα εμπόδια και αποφεύγοντας τα, θα έχει την δυνατότητα να μπορεί να διαλέγει την σωστή διαδρομή αριστερά είτε δεξιά στην περίπτωση που συναντήσει κάποιο εμπόδιο μπροστά του.

Με αυτή την δυνατότητα θα μπορούσε για παράδειγμα να βγει από έναν λαβύρινθο. Παρακάτω φαίνεται πως γίνεται αυτό.



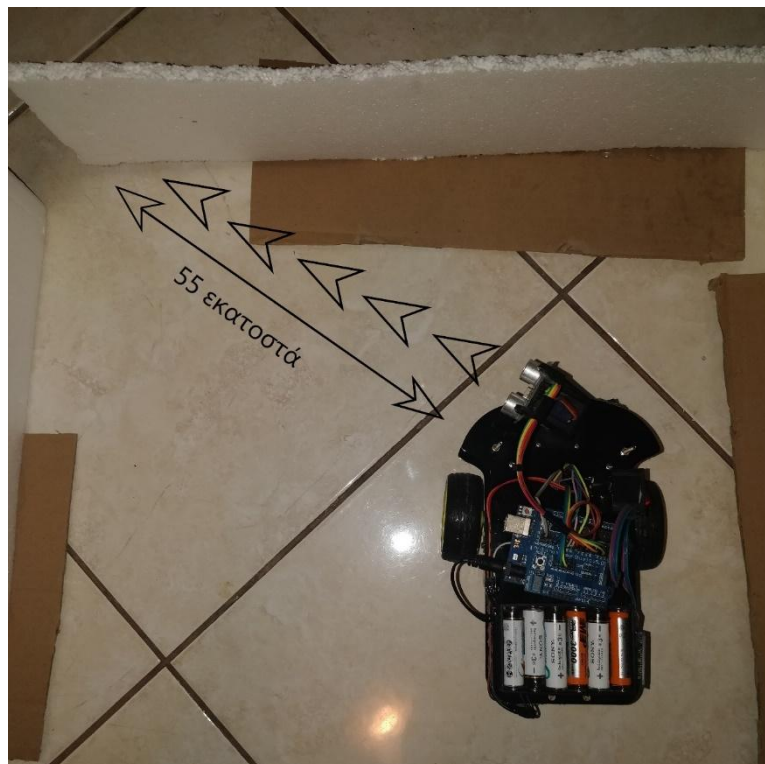
Εικόνα 6.1.1

Στη εικόνα βλέπουμε την στιγμή που το όχημα συναντάει ένα εμπόδιο μπροστά του, η πρώτη κίνηση που κάνει είναι να σταματήσει, έπειτα ελέγχει δεξιά για να δει άμα υπάρχει εμπόδιο και παίρνει μια μέτρηση από τον αισθητήρα σόναρ την οποία στην συνέχεια θα την συγκρίνει για να αποφασίσει άμα θα πάει δεξιά ή αριστερά, η διαδικασία φαίνεται παρακάτω:



Εικόνα 6.1.2

Αφού πάρει την μέτρηση στην συνέχεια θα στρέψει τον αισθητήρα στην αριστερή πλευρά για να ελέγξει και από κει άμα υπάρχει κάποιο εμπόδιο όπως έκανε και προηγούμενός:



Εικόνα 6.1.3

Όπως φαίνεται και από τις εικόνες μετά τις μετρήσεις έχει διαβάσει το πρόγραμμα 1 μέτρηση από τα δεξιά η οποία είναι 25 εκατοστά και άλλη μια από αριστερά η οποία είναι 55 εκατοστά, στην συνέχεια συγκρίνει αυτές τις 2 μετρήσεις, υπολογίζει πια από τις 2 είναι μεγαλύτερη, δηλαδή πιο καθαρή από εμπόδια και στρίβει στην συγκεκριμένη κατεύθυνση, στο παράδειγμά μας είναι η αριστερή κατεύθυνση όπως φαίνεται παρακάτω



Εικόνα 6.1.4

Εφόσον στρίψει από εκεί και έπειτα κάνει ότι και στην αρχή μέχρι να βρει την έξοδο.

Στην περίπτωση τώρα που και από τις 2 μεριές έχει εμπόδιο και η απόσταση και των 2 εμποδίων είναι ίση ή μικρότερη από κάποιο όριο (αυτό το όριο έχει μπει για να εξασφαλίσουμε ότι το όχημα δεν θα προσπαθήσει να στρίψει σε κάποιο στενό σημείο που πιθανώς δεν χωράει), όπως φαίνεται και παρακάτω:



Εικόνα 6.1.5

Τότε το όχημα θα γυρίσει 180 μοίρες, με αυτόν τον τρόπο αλλάζει τελείως κατεύθυνση και σε περίπτωση που ήταν σε κάποιο αδιέξοδο ξεφεύγει από αυτό και συνεχίζει κανονικά την πορεία του.

Σενάριο 2: Λειτουργία παρακολούθησης μαύρης γραμμής

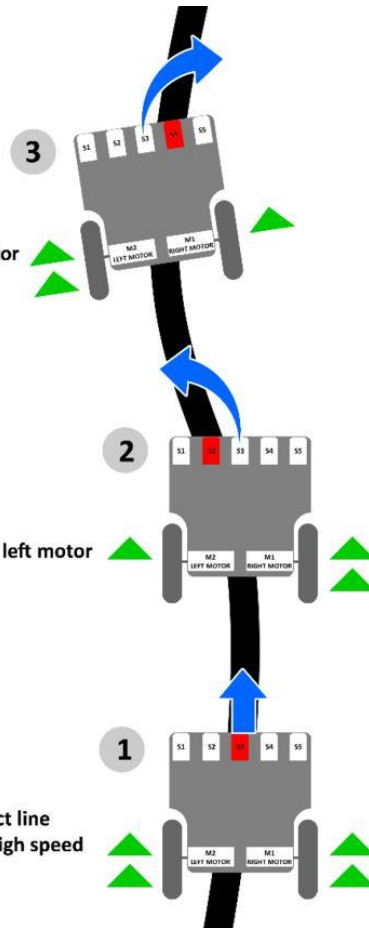
Με την λειτουργία αυτή το όχημα μπορεί να ακολουθεί μια μαύρη γραμμή που έχουμε σχηματίσει στο πάτωμα.

Αυτό επιτυγχάνεται με την χρήση 3 υπέρυθρων αισθητήρων που έχουμε τοποθετήσει κάτω από το όχημα, των οποίων η λειτουργία έχει εξηγηθεί παραπάνω.

Το όχημα για να μπορεί να μην ξεφεύγει από την γραμμή κινείται πλαγίως, κάνει δηλαδή ζιγκ ζαγκ, μόνο έτσι δεν χάνει το όχημα την γραμμή, η κίνηση αυτή φαίνεται στο παρακάτω διάγραμμα:

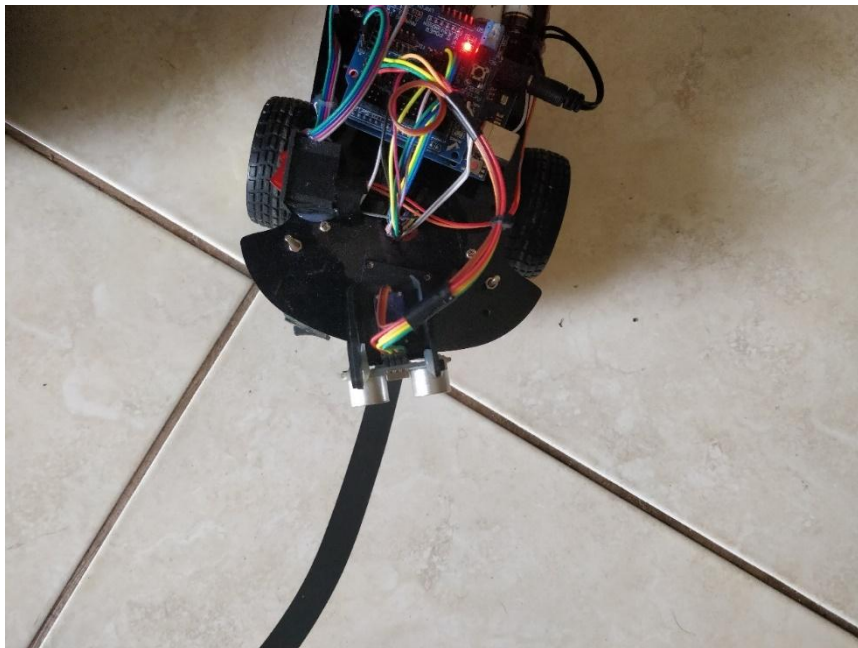
**Very Basic Control Algorithm
(Using 3 sensors)**

S4 (Sensor on right of middle sensor) detect line
Action : Turn right with left motor rotating faster than right motor



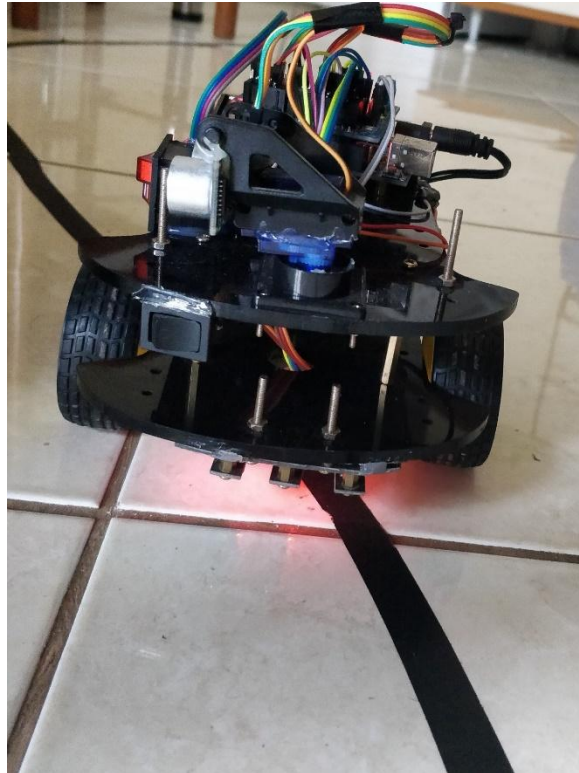
Εικόνα 6.1.6

Μια πιο ρεαλιστική εικόνα φαίνεται παρακάτω με το κανονικό ρομπότ:



Εικόνα 6.1.7

Εδώ το όχημα ανιχνεύει στο μεσαίο αισθητήρα την μαύρη γραμμή και προχωράει ευθεία.



Εικόνα 6.1.8

Εδώ το όχημα ανιχνεύει την γραμμή στον αριστερό αισθητήρα και έτσι κινείται δεξιά.



Εικόνα 6.1.9

Εδώ παρομοίως με πριν ανιχνεύει την γραμμή στον δεξιό αισθητήρα και θα κινηθεί αριστερά



Εικόνα 6.1.10

Τέλος επειδή κάποια στιγμή η μαύρη γραμμή σταματάει το ίδιο πρέπει να κάνει και το όχημα, αυτό επιτυγχάνεται με το να βάλουμε μια στρώση από οριζόντιες μαύρες ταινίες στο τέλος της μαύρης γραμμής, με αυτό τον τρόπο και οι 3 αισθητήρες αντιλαμβάνονται μαύρη γραμμή την ίδια στιγμή και με αυτό τον τρόπο μπορούμε να ελέγξουμε πότε θα σταματήσει το όχημα.

Οι επόμενες λειτουργίες δεν έχουν κάποια ιδιαιτερότητα εκτός από πάτημα ενός κουμπιού και η κίνηση του οχήματος, για αυτό και δεν θα αναλυθούν περαιτέρω.

Κεφάλαιο 7: Συμπεράσματα:

7.1 Γενικά:

Όπως προαναφέρθηκε η μέθοδος της εκμάθησης με STEM είναι χρήσιμη στην εισαγωγή στον προγραμματισμό, πέρα από αυτό για να συναγωνιστεί τον text-based προγραμματισμό θέλει πολύ δουλειά στο μέλλον για να επιτευχθεί κάτι τέτοιο, σε περίπλοκα προγράμματα δεν μπορεί να τα στηρίξει επαρκώς, όχι την συγκεκριμένη στιγμή τουλάχιστον.

Όσο για το κατασκευαστικό κομμάτι της πτυχιακής μετά την πάροδο της υλοποίησης σαν σχέδιο είναι κάτι πολύ εύκολο σε γενικές γραμμές, αρκετά διασκεδαστικό για την μικρή ηλικία και πλήρως λειτουργικό για τον σκοπό που προορίζεται, στην εισαγωγική εκμάθηση προγραμματισμού δηλαδή.

Προσφέρει όλα όσα προσπαθούμε να αναδείξουμε στην παρούσα πτυχιακή, την μάθηση και την διασκέδαση μαζί με αποτέλεσμα την εκμάθηση προγραμματισμού.

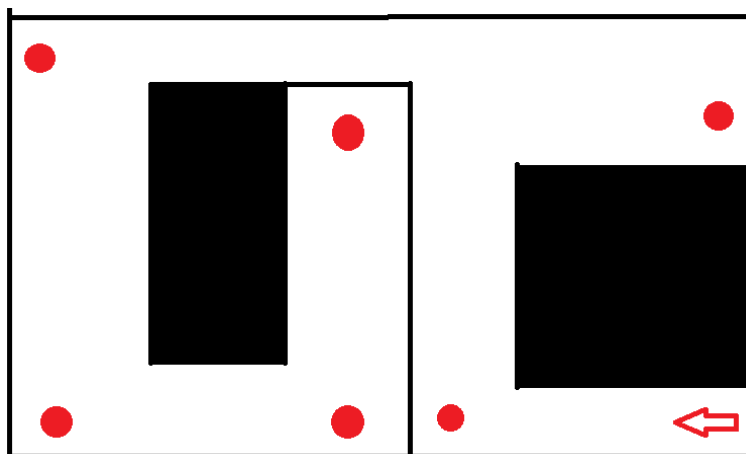
7.2 Συνδυάζοντας παιχνίδι και μάθηση:

Μια διαφορετική τύπου προσέγγιση θα ήταν η κατασκευή ενός λαβύρινθου με σημεία (points) που θα πρέπει ο μαθητής να προγραμματίσει το ρομποτάκι να κινηθεί μέσα από τον λαβύρινθο «συλλέγοντας» points με αποτέλεσμα να μαζέψει ένα συγκεκριμένο αριθμό Points για να βγει εν τέλει έξω από τον λαβύρινθο.

Με αυτό το σχέδιο ο μαθητής χρησιμοποιεί συγκεκριμένες εντολές if, while και for για να επιτύχει την ολοκλήρωση της άσκησης.

Με αυτόν τον τρόπο συνδυάζεται παιχνίδι και μάθηση ταυτόχρονα πράγμα που είναι πολύ σημαντικό στις νεαρές ηλικίες γιατί ο μαθητής πειραματίζεται μόνος του και μαθαίνει να λύνει προγραμματιστικά προβλήματα, κάτι που είναι πολύ σημαντικό στην εισαγωγή στον προγραμματισμό.

Παρακάτω φαίνεται ένα τέτοιο σχέδιο με τους κόκκινους κύκλους να συμβολίζουν τα points:

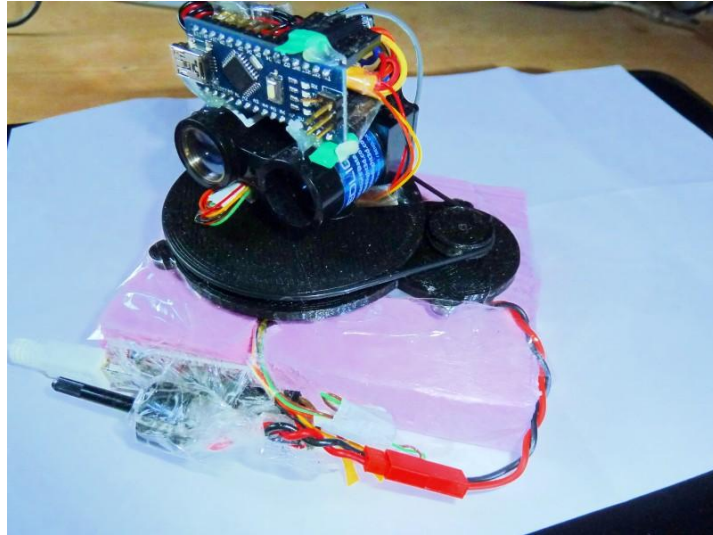


Εικόνα 7.1.1

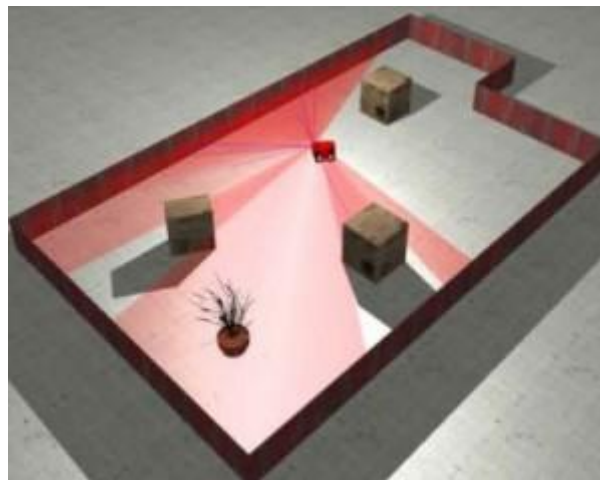
7.3 Μελλοντικές επεκτάσεις:

Όπως είπαμε και πιο πάνω στο προγραμματιστικό κομμάτι ο visual προγραμματισμός θέλει αρκετή δουλειά για να ανταγωνιστεί τον text-based σε θέμα περίπλοκων προγραμμάτων αλλά πέρα από αυτό υστερεί και σε θέματα hardware υποστήριξης.

1) Χαρτογράφηση χώρου με την μέθοδο lidar :



Εικόνα 7.1.2



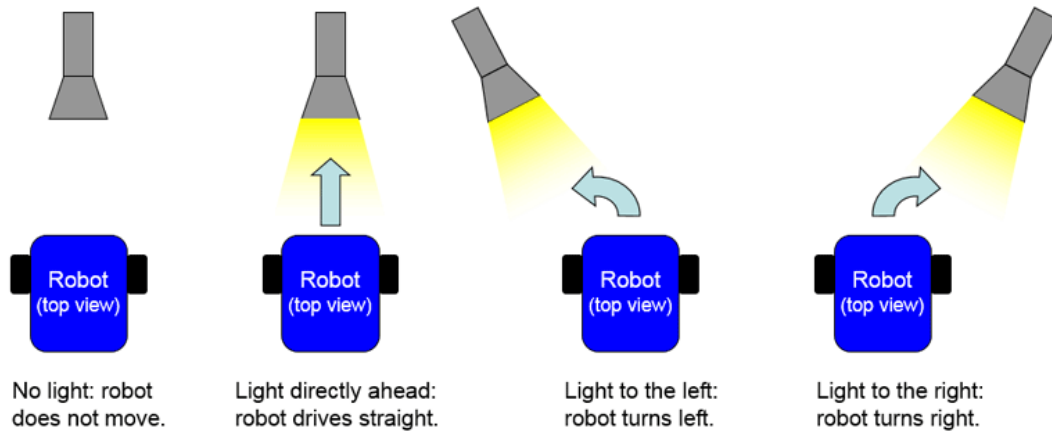
Εικόνα 7.1.3

Η λέξη LIDAR είναι ένας συνδυασμός από τις λέξεις “light” και “radar” επίσης και ακρωνύμιο για το “LIght Detection And Ranging” or “Laser Imaging, Detection, And Ranging.” Σε γενικές γραμμές το LIDAR λειτουργεί ρίχνοντας laser σε ένα αντικείμενο και έπειτα μετράει τον χρόνο που έκανε το φως να επιστρέψει στον σένσορα, με αυτό τον τρόπο μπορεί να μετρήσει την απόσταση από ένα αντικείμενο με πολύ μεγάλη ακρίβεια.

Περιστρέφοντας μια μονάδα Lidar μπορεί να κάνει μια πιστή χαρτογράφηση ενός χώρου.

2) *Light follower*

Με την προσθήκη ενός ή περισσότερων φωτοευαίσθητων αισθητήρων μπορούμε να κάνουμε το όχημα να μπορεί να ακολουθεί μια δέσμη φωτός αναλόγως την κατεύθυνση που βρίσκεται αυτή η δέσμη, όπως δείχνει η παρακάτω εικόνα.



Εικόνα 7.1.4

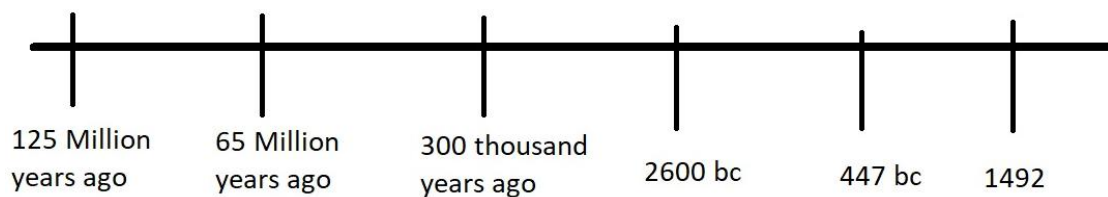
3) *Game-based learning*

Άλλες επεκτάσεις θα μπορούσε να είναι η δημιουργία κατασκευών παρόμοιες όπως το όχημα που κατασκευάστηκε στην παρούσα πτυχιακή για την βοήθεια εκμάθησης μαθημάτων όπως η γεωγραφία ή η ιστορία.

A) Ιστορία

Με την σχεδίαση ενός αλληλεπιδραστικού χρονοαυτοκινήτου (STEAM), που θα μεταφέρει τη γνώση στη αντίστοιχη χρονική περίοδο επάνω σε μια χρονογραμμή.

Με την κατασκευή μιας χρονογραμμής στο πάτωμα για παράδειγμα όπως φαίνεται παρακάτω

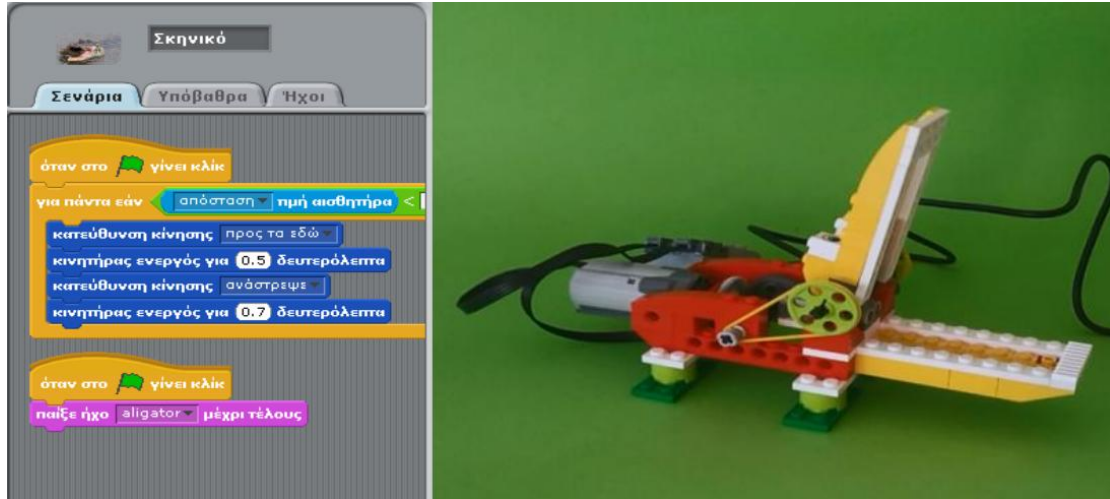


Εικόνα 7.1.5

Θα μπορούσαμε να φτιάξουμε ένα πρόγραμμά στο Scratch το οποίο όταν του δίνουμε ένα κοσμοϊστορικό γεγονός θα τοποθετεί το όχημα στην κατάλληλη χρονική περίοδο πάνω στην χρονογραμμή που έχουμε φτιάξει παράδειγμα άμα του δίνουμε δεινόσαυρους θα τοποθετούσε το όχημα στο 125 million years ago ενώ άμα του δίνουμε Παρθενώνας θα το έβαζε στο 447 bc.

B) Γεωγραφία

Με την βοήθεια των Lego και της Scratch θα μπορούσε να κατασκευαστεί ένα είδος ζώου με τα Lego και να γίνει ο προγραμματισμός του μέσω scratch για να προσομοιώσουμε αυτό το ζώο όπως ακριβώς φαίνεται παρακάτω.



Εικόνα 7.1.6

Βιβλιογραφία

- 1) <http://www.etpe.gr/custom/pdf/etpe2537.pdf>
- 2) <https://robotics-edu.gr/to-steam-%CF%89%CF%82-%CE%B5%CF%80%CE%B9%CF%84%CE%B1%CF%87%CF%85%CE%BD%CF%84%CE%AE%CF%82-%CF%83%CF%84%CE%BF-%CE%BC%CE%AC%CE%B8%CE%B7%CE%BC%CE%B1-%CF%84%CE%B7%CF%82-%CE%B9%CF%83%CF%84%CE%BF%CF%81%CE%AF/>
- 3) http://grauonline.de/wordpress/?page_id=1233
- 4) <https://code.likeagirl.io/text-based-vs-block-based-coding-part-1-ab1ce60bc943>
- 5) https://www.researchgate.net/publication/320678674_Comparing_Block-Based_and_Text-Based_Programming_in_High_School_Computer_Science_Classrooms
- 6) <https://www.codingkids.com.au/blog/visual-based-vs-text-based-programming-languages/>
- 7) http://sv1ahh1.blogspot.com/2018/07/stem_7.html
- 8) <https://www.makeblock.com/coding-robot/208984.html>
- 9) [https://en.wikipedia.org/wiki/Scratch_\(programming_language\)#Features_and_derivatives](https://en.wikipedia.org/wiki/Scratch_(programming_language)#Features_and_derivatives)
- 10) <https://scratch.mit.edu/projects/10128431/editor>
- 11) <http://scratch.redware.com/lessonplan/conclusion>
- 12) <http://epri.korinthos.uop.gr/BlogsPortal/mps2018/2018/04/25/%CE%B5%CE%BA%CF%80%CE%B1%CE%B9%CE%B4%CE%B5%CF%85%CF%84%CE%B9%CE%BA%CE%AE-%CF%81%CE%BF%CE%BC%CF%80%CE%BF%CF%84%CE%B9%CE%BA%CE%AE-%CF%85%CF%80%CE%BF%CE%BB%CE%BF%CE%B3%CE%B9%CF%83%CF%84%CE%B9-3/#more-583>
- 13) <http://epri.korinthos.uop.gr/BlogsPortal/mps2018/2018/04/15/%CE%B7-%CE%B5%CE%B9%CF%83%CE%B1%CE%B3%CF%89%CE%B3%CE%AE-%CF%80%CF%81%CE%BF%CE%B3%CF%81%CE%B1%CE%BC%CE%BC%CE%AC%CF%84%CF%89%CE%BD-stemsteam-%CE%BA%CE%B1%CE%B9-%CF%80%CF%81%CE%BF%CE%B3%CF%81%CE%B1/>
- 14) <https://github.com/jguille2/SA5/blob/master/firmata/SA5Firmata.md>
- 15) http://sensorkit.joy-it.net/index.php?title=KY-033_Tracking_Sensor_Modul
- 16) <https://www.dx.com/p/ir-receiver-module-wireless-remote-control-kit-for-arduino-1-x-cr2025-2015666#.XD3IDlwza70>
- 17) https://www.banggood.com/PTZ-Bracket-Kit-With-SG90-Servo-And-SR04-Ultrasonic-Module-p-996832.html?cur_warehouse=CN
- 18) <https://www.brainy-bits.com/ir-remote-arduino/>
- 19) <https://megaeshop.pk/>
- 20) https://docs.google.com/document/d/1Y-yZnNhMYy7rwhAgyL_pfa39RsB-x2qR4vP8saG73rE/edit#
- 21) <http://e-belajarelektronika.com/definisi-dan-bagian-robot-line-follower/>
- 22) <http://www.themakersworkbench.com/tutorial/triggering-servo-using-hc-sr04-distance-sensor-and-arduino>

- 23) <https://randomnerdtutorials.com/complete-guide-for-ultrasonic-sensor-hc-sr04/>
- 24) <http://www.icstation.com/tracking-sensor-module-infrared-obstacle-avoidance-arduino-project-p-2787.html>
- 25) http://yourduino.com/sunshop//index.php?l=product_detail&p=407
- 26) <https://robottini.altervista.org/tag/line-follower>
- 27) <http://www.maxphi.com/ultrasonic-sensor-interfacing-arduino-tutorial>
- 28) <http://aimagin.com/blog/pwm-capture-data-acquisition-and-ultrasonic-sensor/>
- 29) <https://create.arduino.cc/projecthub/16336/line-follower-robot-using-arduino-5f5698>
- 30) http://www.geeetech.com/wiki/index.php/L298N_Motor_Driver_Board
- 31) <https://gr.pinterest.com/pin/369365606912052091/>
- 32) <http://robotechshop.com/shop/arduino/arduino-shield/arduino-sensor-shield-v5-0/?v=f214a7d42e0d>
- 33) <http://tinkbox.ph/store/modules/keyes-h-bridge-driver-l298n-v2>
- 34) <https://www.amazon.com/Makerfire-4-wheel-Chassis-Encoder-Arduino/dp/B00NAT3VF4>
- 35) <https://market24.gr/el/raspberry-arduino/2303-arduino-arduino-car-tracing-obstacle-avoidance-4wd-5200250811257.html>
- 36) https://www.banggood.com/Smart-Robot-Car-Chassis-Kit-Speed-Encoder-Battery-Box-For-Arduino-p-981975.html?cur_warehouse=CN
- 37) <https://www.quora.com/What-are-Bluetooth-profiles>
- 38) <https://en.wikipedia.org/wiki/Arduino>
- 39) <https://deltahacker.gr/arduino-intro/>