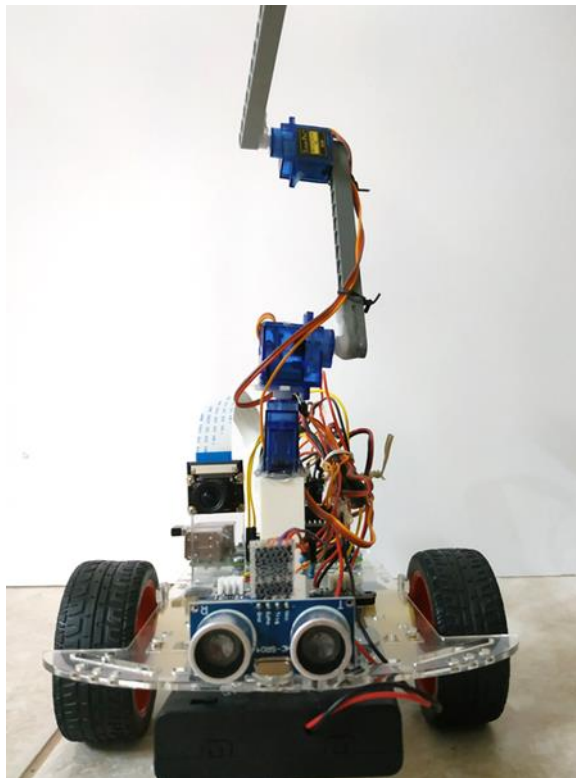


## ΜΕΤΑΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ

Θέμα:

Δημιουργία εκπαιδευτικής ρομποτικής πλατφόρμας για εισαγωγή της ρομποτικής στη δευτεροβάθμια εκπαίδευση



Φοιτητής:

Καραλέκας Γεώργιος (ΑΜ: 14)

Επιβλέπων καθηγητής:

Δρ. Βολογιαννίδης Σταύρος



### **Υπεύθυνη Δήλωση**

Βεβαιώνω ότι είμαι συγγραφέας αυτής της διπλωματικής εργασίας και ότι κάθε βοήθεια την οποία είχα για την προετοιμασία της, είναι πλήρως αναγνωρισμένη και αναφέρεται στην πτυχιακή εργασία. Επίσης έχω αναφέρει τις όποιες πηγές από τις οποίες έκανα χρήση δεδομένων, ιδεών ή λέξεων, είτε αυτές αναφέρονται ακριβώς είτε παραφρασμένες. Επίσης βεβαιώνω ότι αυτή η διπλωματική εργασία προετοιμάστηκε από εμένα προσωπικά ειδικά για τις απαιτήσεις του προγράμματος σπουδών του Μεταπτυχιακού με τίτλο «Ρομποτική»(MSc in Robotics)του Τμήματος Μηχανικών Πληροφορικής Τ.Ε. της Σχολής Τεχνολογικών Εφαρμογών του Τ.Ε.Ι. Κεντρικής Μακεδονίας.

Ο Δηλών

Καραλέκας Γεώργιος

## **ΕΥΧΑΡΙΣΤΙΕΣ**

Η εκπόνηση της διπλωματικής εργασίας αποτελεί το τελικό στάδιο για την απόκτηση μεταπτυχιακού διπλώματος. Είναι μια δουλειά που έγινε με αρκετή αγάπη για πειραματισμό και δημιουργία και οδήγησε στη συγγραφή κι μιας εργασίας που έγινε δεκτή στο διεθνές συνέδριο “10th IEEE International Conference on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications” που θα γίνει στις 18-21 Σεπτεμβρίου 2019 στο Metz της Γαλλίας.

Στην προσπάθεια αυτή θα ήθελα να ευχαριστήσω τον Δρ. Βολογιαννίδη Σταύρο και τον Δρ. Καλόμοιρο Ιωάννη για την υποστήριξη, την καθοδήγηση και τις διορθώσεις τους και φυσικά την παρότρυνσή τους για την συγγραφή της εργασίας. Αλλά και όλους τους συντελεστές και καθηγητές του μεταπτυχιακού προγράμματος Ρομποτικής του ΤΕΙ Κεντρικής Μακεδονίας.

Τέλος ένα μεγάλο ευχαριστώ στο στήριγμά της ζωής μου την γυναίκα μου, Έλενα και τα δύο μου παιδιά Αλέξη και Δημήτρη για την υπομονή και την ανοχή τους στα ατελείωτα ξενύχτια και στην έλλειψη μου από κοντά τους.

Καραλέκας Γεώργιος

Κατερίνη Μάιος 2019

## **Σύνοψη**

Αυτή η εργασία παρουσιάζει ένα πολύ χαμηλού κόστους εκπαιδευτικό ρομπότ που αναπτύχθηκε για να εισαγάγει τους μαθητές της δευτεροβάθμιας εκπαίδευσης στη ρομποτική. Ο στόχος είναι ότι μπορεί να είναι βολικό, φθηνό και επεκτάσιμο. Τα κιτ που διατίθενται επί του παρόντος στην αγορά εάν είναι φθηνά έχουν περιορισμένες δυνατότητες, ενώ στην περίπτωση των ρομπότ αξιόλογων δυνατοτήτων, είναι πολύ πιο ακριβά. Λόγω της σημασίας της ρομποτικής είναι πλέον απαραίτητο να εισαχθεί στη δευτεροβάθμια εκπαίδευση, συνεπώς, το κόστος τους δεν πρέπει να είναι απαγορευτικό για μια σχολική τάξη. Τέλος υπάρχουν έξι προτεινόμενα σχέδια μαθήματος για αξιοποίηση του στη τάξη.

## **Abstract**

This paper presents a very low cost educational robot developed to introduce secondary school students to robotics. The goal is that it can be handy, cheap and extensible. The kits that are currently available on the market if they are inexpensive have limited capabilities, while in the case of robots of worthwhile capabilities they are much more expensive. Due to the importance of robotics for the modern education of young students, it is now necessary to introduce it into secondary education, therefore, their costs should not be prohibitive for a school class. Finally, there are six proposed lesson plans for its use in the classroom.

## Πίνακας περιεχομένων

Υπεύθυνη Δήλωση .....	3
ΕΥΧΑΡΙΣΤΙΕΣ .....	4
Σύνοψη .....	5
Abstract .....	5
Κεφάλαιο 1 <sup>ο</sup> .....	11
Δημιουργία εκπαιδευτικής ρομποτικής πλατφόρμας για εισαγωγή της ρομποτικής στη δευτεροβάθμια εκπαίδευση. ....	11
Κεφάλαιο 2 <sup>ο</sup> .....	15
Εκπαιδευτικές ρομποτικές πλατφόρμες. ....	15
Προσχολική ηλικία.....	15
<b>BeeBot</b> .....	15
WeDraw-Eggy.....	16
Botley .....	17
Clementoni My First Robot .....	18
UARO.....	18
Πρωτοβάθμια εκπαίδευση.....	21
Edison 2.0.....	21
WeDo 2.0 lego.....	22
Pro-Bot .....	23
Thymio .....	24
BinaryBots-Meet DIMM kit .....	26
Gigo S4A Programming Bricks .....	27
Robot Dash Workshop Wonder .....	28
Δευτεροβάθμια εκπαίδευση .....	29
Neuron.....	29
mBot.....	30
Pi2Go Raspberry Pi Programmable Floor Robot.....	32

BQ PrintBot EvolutionH .....	34
LEGO Education MINDSTORMS EV3 .....	36
VEX Classroom and Competition Super Kit .....	38
AlphaBot .....	39
Τριτοβάθμια εκπαίδευση.....	41
Arlo.....	42
SDP Mini RPlidar 2WD .....	43
TurtleBot .....	44
Eddie.....	48
Ρομποτικοί Βραχίονες.....	50
Magician Dobot.....	50
Gauss6 .....	52
Αξιολόγηση και γενικά συμπεράσματα εκπαιδευτικών ρομπότ.....	53
Κεφάλαιο 3° .....	56
Κεντρικός στόχος της εκπαιδευτικής πλατφόρμας .....	56
Επεκτασιμότητα εκπαιδευτικής πλατφόρμας.....	57
Ρομποτικός Βραχίονας .....	60
Κόστος ρομπότ.....	60
Επιπλέον αισθητήρες ρομπότ.....	61
Κεφάλαιο 4° .....	62
Κατασκευή και προγραμματισμός του Ρομπότ.....	62
ROS .....	62
Σασί και κίνηση του ρομπότ. ....	64
Κεντρική μονάδα ελέγχου Raspberry Pi 3+ Model B.....	67
Ρομποτικός βραχίονας.....	67
Αισθητήρες/κάμερα.....	68
Κάμερα.....	69

Motor controller .....	70
Λογισμικό για τον έλεγχο του Ρομπότ .....	71
Scripts τα οποία είναι nodes (διεργασίες) .....	72
A) kg_driver_node_key.....	72
B) joint_state_controller2.py.....	73
Γ)Ultrasound_Publisher .....	75
Τρόπος επικοινωνίας Ρομπότ με τον απομακρυσμένο υπολογιστή.....	75
Έτοιμες εφαρμογές του ROS για τον έλεγχο του ρομπότ.....	77
Προσομοίωση του ρομπότ. ....	78
Εικόνα από την κάμερα του RaspBerry .....	81
Χρησιμοποίηση των αισθητήρων οπτοδιακοπής στους τροχούς .....	82
1η Εφαρμογή rtm.py .....	83
2η Εφαρμογή rtm_move2.py.....	86
Εφαρμογή απλής οδομετρίας με το εικονικό ρομπότ. ....	87
Nodes στην υλοποίηση της πλατφόρμας.....	89
Ηλεκτρονικό σχεδιάγραμμα του ρομπότ.....	91
Η τελική μορφή του ρομπότ.....	92
Κεφάλαιο 5 <sup>ο</sup> .....	94
Λειτουργία και αξιολόγηση του ρομπότ. ....	94
Αξιολόγηση της λειτουργίας της πλατφόρμας.....	98
Προτάσεις δίωρων Μαθημάτων.....	99
1)Εισαγωγικό μάθημα .....	99
2)Παράδοση του Kit στους μαθητές.....	100
3)Εισαγωγή στο ROS.....	100
4)Απλή οδομετρία .....	101
5)Κινηματική του βραχίονα .....	101
6)Τηλεκατεύθυνση του ρομπότ μέσω της κάμερας.....	101



Κεφάλαιο 6° .....	102
Συμβολή της μεταπτυχιακής εργασίας.....	102
Συμπεράσματα της εργασίας.....	102
Μελλοντική επέκταση της δουλειάς .....	103
Βιβλιογραφία.....	105
Παράρτημα.....	109
Κινηματική Ανάλυση Ρομπότ Διαφορικής Οδήγησης.....	109
<b>Ευρετήριο εικόνων</b>	
Εικόνα 1Αγορά των ρομπότ στο μέλλον .....	12
Εικόνα 2Αναδιάταξη της αγοράς των Ρομπότ στο μέλλον .....	13
Εικόνα 3BeeBot .....	15
Εικόνα 4 WeDraw-Eggy .....	16
Εικόνα 5 Botley .....	17
Εικόνα 6ClementoniMyFirstRobot.....	18
Εικόνα 7 UARO.....	18
Εικόνα 8 Edison 2.0 .....	21
Εικόνα 9 WeDo 2.0 lego.....	22
Εικόνα 10 Pro-Bot .....	23
Εικόνα 11 Thymio .....	24
Εικόνα12 BinaryBots-Meet DIMM kit.....	26
Εικόνα 13RobotDashWorkshopWonder.....	28
Εικόνα 14 Neuron .....	29
Εικόνα 15 mBot .....	30
Εικόνα16 Pi2Go Raspberry Pi Programmable Floor Robot .....	32
Εικόνα 17 BQ PrintBot EvolutionH .....	34
Εικόνα18 VEX Classroom and Competition Super Kit.....	38
Εικόνα 19 AlphaBot.....	39
Εικόνα 20 Arlo.....	42
Εικόνα 21SDPMiniRPlidar 2WD .....	43
Εικόνα 22 TurtleBot.....	45
Εικόνα 23 TurtleBot2.....	46

Εικόνα 24 TurtleBot3.....	47
Εικόνα 25 Eddie.....	48
Εικόνα 26 Magician Dobot.....	50
Εικόνα 27 Gauss6 .....	52
Εικόνα 28 ρόδα τύπου castor .....	65
Εικόνα 29 σφαιρικός τροχός.....	66
Εικόνα 30 Σασί, κινητήρες μπαταριοθήκη .....	66
Εικόνα 31 RaspberryPi 3 ModelB .....	67
Εικόνα 32 αισθητήρας υπερήχων απόστασης ο SR04.....	68
Εικόνα 33 Οπτικός Αισθητήρας Στροφών Waveshare 12225 .....	68
Εικόνα 34 δίσκος με διάκενα.....	69
Εικόνα 35 κάμερα.....	69
Εικόνα 36 Η bridgeHUT.....	70
Εικόνα 37 η IP διεύθυνση του υπολογιστή.....	76
Εικόνα 38 Το ρομπότ στο Rviz.....	79
Εικόνα 39 Το ρομπότ στο Gazebo.....	80
Εικόνα 40 Το ρομπότ στο Moveit.....	81
Εικόνα 41 εικόνα κάμερας.....	82
Εικόνα 42 σχήμα κινηματικής ρομπότ .....	85
Εικόνα 43 Σχεδιάγραμμα με τα Nodes και τις συνδέσεις τους.....	90
Εικόνα 44 ηλεκτρονικό σχεδιάγραμμα.....	91
Εικόνα 45 σύνδεση κάμερα και τροφοδοσίας .....	92
Εικόνα 46 Κάτω όψη .....	92
Εικόνα 47 Επάνω όψη. ....	93
Εικόνα 48 Εμπρός όψη .....	93
Εικόνα 49 ρομπότ διαφορικής κίνησης 1 .....	110
Εικόνα 50 ρομπότ διαφορικής κίνησης 2 .....	110
Εικόνα 51 συνεισφορά κάθε τροχού στο ρομπότ διαφορικής κίνησης .....	111
Εικόνα 52 συστήματα συντεταγμένων .....	112

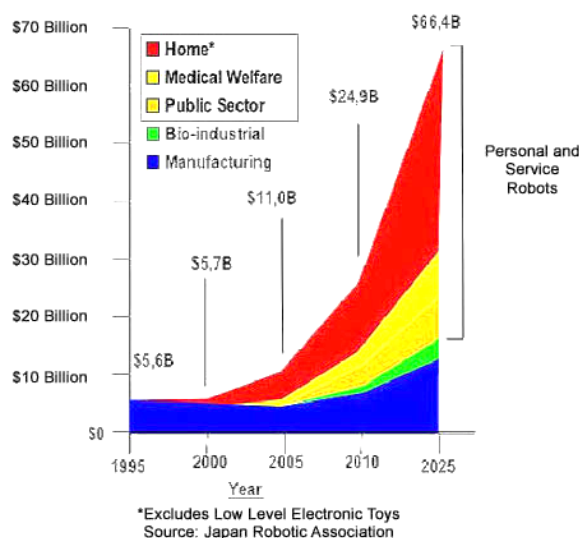
## Κεφάλαιο 1<sup>ο</sup>

### **Δημιουργία εκπαιδευτικής ρομποτικής πλατφόρμας για εισαγωγή της ρομποτικής στη δευτεροβάθμια εκπαίδευση.**

Στην παρούσα εργασία κατασκευάστηκε ένα ρομπότ το οποίο είναι φτιαγμένο με υλικά τα οποία μπορούν να βρεθούν πολύ εύκολα στο εμπόριο και το κόστος τους δεν είναι απαγορευτικό για το εργαστήριο ενός σχολικού συγκροτήματος[25]. Η δυσκολία εύρεσης κατάλληλου hardware καθιστά απαγορευτική την χρήση τους στη σχολική τάξη μιας και τα ατυχήματα και οι απροσεξίες είναι κάτι το συνηθισμένο. Το λογισμικό που χρησιμοποιείται είναι ελεύθερο και βασίζεται σε άδειες τύπου BSD, όπου πρόκειται για μια πολύ ανοικτή άδεια που επιτρέπει την επαναχρησιμοποίηση του κώδικα τόσο σε εμπορικά αλλά και ελεύθερα προϊόντα. Σημειώνεται ότι η GPL, η οποία είναι παρόμοια, με την διαφορά από την άδεια BSD ότι κάθε αντίγραφο, παράγωγο και προϊόν ενός GPL προγράμματος, υποχρεούται να κυκλοφορεί κι αυτό υπό την ίδια άδεια. Το συγκεκριμένο ρομπότ, το οποίο με τα φθηνά υλικά που χρησιμοποιήθηκαν το καθιστά, μια ιδανική επιλογή από πλευράς κόστους για την εισαγωγή στη ρομποτική, και όχι μόνο, για παιδιά της δευτεροβάθμιας εκπαίδευσης.

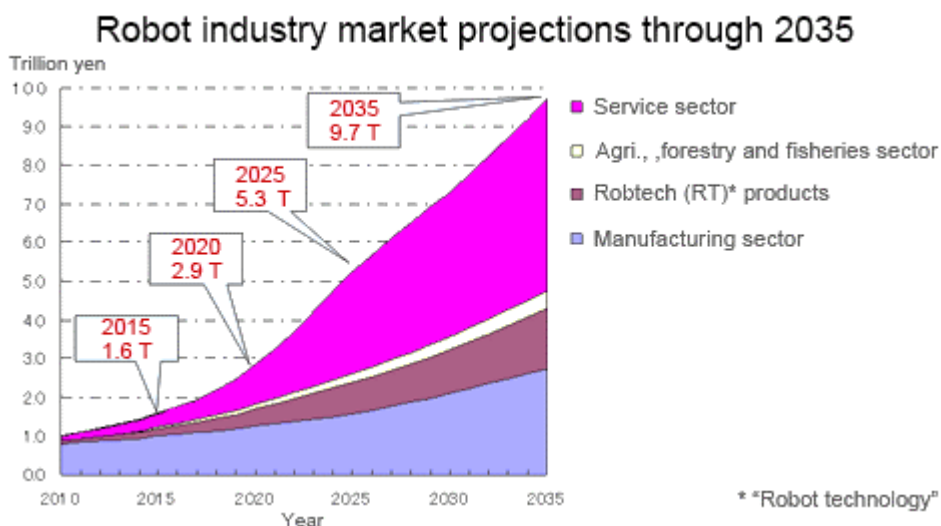
Ταυτόχρονα όμως το συγκεκριμένο ρομπότ παρουσιάζει δυνατότητες οι οποίες απαντούνται σε ρομποτικές πλατφόρμες πολλαπλάσιου κόστους. Οι δυνατότητες επέκτασης των εφαρμογών που μπορούν να εφαρμοστούν επάνω του είναι πάρα πολλές. Το κύριο λογισμικό που χρησιμοποιείται είναι το ROS [26] (RobotOperatingSystem). Το ROS δεν είναι κάποια συγκεκριμένη γλώσσα προγραμματισμού αλλά ένα ανοιχτού κώδικα, μετα-λειτουργικό σύστημα για ρομπότ που βασίζεται σε Linux [28] το οποίο περιέχει πληθώρα έτοιμων εργαλείων. Η γνώση του μας λύνει τα χέρια για πολλές εφαρμογές της ρομποτικής χωρίς να χρειάζεται να ασχοληθούμε με πράγματα τα οποία έχουν ήδη ευρεία εφαρμογή.

Η εκπαίδευση στη ρομποτική είναι το εκκολλητήριο των νέων επιστημόνων οι οποίοι θ' ασχοληθούν αργότερα με αυτό το αντικείμενο. Όπως σήμερα υπάρχει μία τηλεόραση ένα ψυγείο ή κι ένα υπολογιστής, τα επόμενα χρόνια θα υπάρχει ένα ρομπότ. Η απαίτηση για ρομπότ σε τομείς εκτός της βιομηχανίας τα επόμενα χρόνια θα είναι πολλαπλάσια[8,9]. Σύμφωνα με μελέτη του Συνδέσμου Ρομποτικής της Ιαπωνίας (Εικόνα 1) εκτιμάτε τα προσεχή χρόνια η ζήτηση σε οικιακά, ιατρικά δημοσίου συμφέροντος αλλά και άλλα είδη ρομπότ να ξεπεράσουν σε τζίρους αυτά της βιομηχανίας.



Εικόνα 1 Αγορά των ρομπότ στο μέλλον

Σύμφωνα με το ιαπωνικό Υπουργείο Οικονομίας και Εμπορίου, η αγορά της βιομηχανίας ρομπότ θα αναδιαταχτεί δραματικά (Εικόνα. 2). Η κυριαρχία θα φύγει από τα κλασικά βιομηχανικά ρομπότ που χρησιμοποιούνται στον τομέα της μεταποίησης, και θα περάσει στα λεγόμενα ρομπότ υπηρεσιών, τα οποία έχουν ήδη ξεπεράσει στα επιστημονικά περιοδικά και τα συνέδρια η θεματολογία τους τα βιομηχανικά ρομπότ. Αυτό θα συμβαίνει όλο και περισσότερο στο εγγύς μέλλον. Αυτό δείχνει η αύξηση της ζήτησης για ιατρικά και παρόμοια ρομπότ. (Στις περισσότερες περιπτώσεις, πρόκειται για κινητά ρομπότ.) Το γενικό συμπέρασμα που βγαίνει απ' όλα αυτά είναι η μεγάλη ζήτηση η οποία θα παρουσιαστεί στο μέλλον για εργατικό δυναμικό εξειδικευμένο στη ρομποτική.



Εικόνα 2 Αναδιάταξη της αγοράς των Ρομπότ στο μέλλον

Επίσης ο τομέας των ρομπότ υπηρεσιών, μπορεί να μην είναι μια βαριά βιομηχανία όπως παραδείγματος χάρη μια αυτοκινητοβιομηχανία ή και ακόμη μια σύγχρονη βιομηχανία παραγωγής ρομπότ, αλλά αποτελείται από μονάδες μικρότερης κλίμακας - όπου η χώρα μας μπορεί να παίξει κάποιο ρόλο. Ένας από τους στόχους της παρούσας εργασίας είναι να καταδείξει στους μαθητές την ευκολία κατασκευής ενός ρομπότ το οποίο μπορεί να είναι αρκετά επεκτάσιμο με πλήθος εφαρμογών.

Η χρήση της ρομποτικής σήμερα εστιάζεται κυρίως στην χρήση της ως εργαλείο για STEM (ScienceTechnologyEngineering Math) [10-13] προσεγγίσεις της εκπαίδευσης, και όχι σαν αυτούσιο μάθημα. Στην παρούσα εργασία η φιλοδοξία της είναι να αντιστρέψει την εκπαιδευτική προσέγγιση. Δηλαδή μέσα από την ρομποτική οι μαθητές να κατανοούν την ανάγκη σύνδεσης των μαθημάτων και τα εφόδια που τους δίνουν για την κατασκευή, και επίλυση των δυσκολιών που προκύπτουν κατά τη διάρκεια δημιουργίας του ρομπότ[14-18].

Κατά τη διάρκεια κατασκευής του ρομπότ οι μαθητές κατανοούν διάφορες έννοιες φυσικής και μηχανικής οι οποίες μέχρι τη στιγμή αυτή, το ποιο πιθανό είναι να μην γνώριζαν τη χρησιμότητά τους και τις εφαρμογές τους. Κάτι που ακούμε συνεχώς στην τάξη είναι “κύριε αυτό εμένα που θα μου χρειαστεί στη ζωή μου;” Σε πολλά θέματα λοιπόν όπως θα δούμε οι παρούσα πλατφόρμα απαντά. Έννοιες γεωμετρίας, τριγωνομετρίας στα μαθηματικά δύναμης, ροπής, μετατόπισης, ταχύτητας, γωνιακής ταχύτητας, ηλεκτρισμού ,στη φυσική, στατικότητα, κέντρου βάρους, ροπής αδράνειας στην μηχανική και φυσικά προγραμματισμού αποκαλύπτεται η

αναγκαιότητα τους και οι εφαρμογή τους στο σύγχρονο κόσμο. Η χαρά της δημιουργίας, της ανακάλυψης και της επίτευξης του στόχου μπορούν να συνδυαστούν με την υπάρχουσα πλατφόρμα.

## Κεφάλαιο 2<sup>ο</sup>

### Εκπαιδευτικές ρομποτικές πλατφόρμες.

Ο τομέας των εκπαιδευτικών ρομπότ έχει να επιδείξει αρκετές πλατφόρμες με διαφορετικές προσεγγίσεις η κάθε μία και διαφορετικό κοινό - στόχο. Υπάρχουν εκπαιδευτικά ρομπότ για την προσχολική ηλικία, το δημοτικό, την δευτεροβάθμια εκπαίδευση και το πανεπιστήμιο.

### Προσχολική ηλικία

Για την προσχολική ηλικία οι πλατφόρμες χαρακτηρίζονται από την απλότητα κατασκευής και χειρισμού τους. Όπως γίνεται εύκολα κατανοητό το ζητούμενο σ αυτή την κατηγορία είναι η ευκολία χειρισμού του ρομπότ από τα μικρά παιδιά, η στιβαρή του κατασκευή για μηχανικές καταπονήσεις και η απλότητα προγραμματισμού του. Τα χαρακτηριστικά αυτών των ρομπότ είναι να μπορούν να ακολουθούν απλές εντολές οι οποίες εισάγονται στο ρομπότ συνήθως από ένα πληκτρολόγιο επάνω του ή τηλεχειριστήριο. Οι εντολές που εκτελούν είναι κίνησης, εκτέλεσης κάποιων ηχητικών σημάτων και ανάματος κάποιων Led που φέρουν στο σασί τους. Ρομπότ αυτής της κατηγορίας είναι τα BeeBot, Wedraw Eggy, Botley, Clementoni My First Robot, UARO που αναλύονται παρακάτω.

### BeeBot



Εικόνα 3 BeeBot

Το BeeBot, εικόνα 3 η “έξυπνη μέλισσα”, είναι ένα προγραμματιζόμενο ρομπότ δαπέδου ειδικά κατασκευασμένο για να χρησιμοποιείται από παιδιά προσχολικής ηλικίας καθώς και των πρώτων τάξεων δημοτικής εκπαίδευσης. Ο προγραμματισμός της γίνεται με πλήκτρα που βρίσκονται επάνω της (on-board) και μπορεί να προγραμματιστεί για να κινείται με ακρίβεια στο χώρο προχωρώντας μπροστά, πίσω,

στρίβοντας αριστερά και δεξιά. Τα BeeBot έχουν ένα φιλικό και ευχάριστο σχεδιασμό που ικανοποιεί τους μαθητές. Μπορούν να κινούνται σε λείες επίπεδες και ελαφρά επικλινείς επιφάνειες διαφόρων υλικών όπως: χαρτί, μουσαμά, τσιμέντο, πλακάκι, ξύλο, πλαστικό, χαλί.

#### Χαρακτηριστικά του BeeBot

- Στιβαρή κατασκευή και μικρό σε μέγεθος
- Ευκρινή κουμπιά (πλήκτρα εντολών)
- Μνήμη για να προγραμματίσετε μέχρι και 40 βήματα
- Εύκολος προγραμματισμός
- Επιβεβαιώνει τις οδηγίες σας (προγραμματισμό) τόσο με χαρακτηριστικό ήχο όσο και αναβοσβήνοντας τα μάτια
- Μεγάλη ποικιλία και διαθεματικότητα διαθέσιμων σεναρίων
- Κινείται με βήμα 15 εκατοστών (150mm  $\pm$ 8mm) και στρέφεται (αριστερά ή δεξιά) κατά 90° με ακρίβεια ( $\pm$ 4%)

**Κόστος: 62.90 Ευρώ**

#### WeDraw-Eggy



Εικόνα 4 WeDraw-Eggy



Το WeDraw-Eggy εικόνα 4 είναι ένα εκπαιδευτικό παιχνίδι ρομπότ σχεδιασμένο για παιδιά ηλικίας 3-8 ετών. Είναι σαν "δάσκαλος στο σπίτι". Με το WeDraw-Eggy, τα παιδιά μπορούν να μάθουν σχεδίαση, ορθογραφία, αλφάβητα, καταμέτρηση, αναγνώριση σχήματος, προσθήκες, αφαίρεση και μαθηματικά παζλ. Μιλάει, αναβοσβήνει, ζωγραφίζει και γράφει. Οι εντολές δίνονται με ειδικές κάρτες προγραμματισμού.

**Κόστος:69.95 Ευρώ**

### **Botley**



Εικόνα 5 Botley

Το Botley αποτελεί μέλος της οικογένειας Learning Resources, που έχει ως στόχο να εισαγάγει τα παιδιά στο κόσμο του προγραμματισμού με τον πιο απλό και εύκολο τρόπο. Το Botley είναι απαλλαγμένο από οθόνες. Προγραμματίζεται με κάρτες οι οποίες παρέχονται με το κιτ.

Το Botley μπορεί:

- Να αναγνωρίσει αντικείμενα και να κινείται γύρω τους
- Να ακολουθεί επαναλαμβανόμενες εντολές
- Να κινείται σε διαδρομές αποφεύγοντας εμπόδια
- Να ακολουθεί μαύρες γραμμές

**Κόστος:85.00 Ευρώ**

## Clementoni My First Robot



Εικόνα 6 Clementoni My First Robot

Το Clementoni My First Robot έχει παρόμοιες λειτουργίες με το BeeBot και προγραμματίζεται με το ίδιο τρόπο. Επιπλέον χαρακτηριστικά αυτού του κιτ είναι ένας αισθητήρας μαγνητικού πεδίου όπου μπορεί να αναγνωρίσει εάν υπάρχουν μέταλλα μπροστά του, ή δυνατότητα συναρμολόγησης του μιας και είναι αποσυναρμολογημένο αρχικά, και η προσθήκη μολυβιού η στυλό για γραφή της τροχιάς του.

**Κόστος: 32.90 Ευρώ**

## UARO.



Εικόνα 7 UARO

Το UARO. Η εκπαίδευση με το UARO χωρίζεται σε τρία επίπεδα με προσθήκες νέες σε κάθε επίπεδο.

**Επίπεδο 1:** τα παιδιά θα μάθουν να συναρμολογούν ένα ρομπότ βήμα – βήμα χρησιμοποιώντας πλαίσια διαφόρων σχημάτων και μεγεθών, ενώνοντας τα με βίδες και μπουλόνια και προσθέτοντας μοτέρ και ρόδες που θα του δώσουν κίνηση.

**Το πακέτο περιλαμβάνει:**

- 41 Πλαστικά πλαίσια (περιλαμβάνονται επιπλέον 4 κομμάτια που ενώνονται με LEGO DUPLO)
- 39 Βίδες
- 39 Μπουλόνια
- 4 Ρόδες
- 1 Θήκη για μπαταρίες (μπαίνει στη βάση)
- 1 Βάση με κουμπί on/off και τέσσερις θύρες σύνδεσης
- 2 Μοτέρ
- 2 Λαμπάκια LED που συνδέονται στη βάση (αναβοσβήνουν όταν τροφοδοτούνται με ρεύμα)
- 1 Κατσαβίδι
- 1 Σετ διακοσμητικά μάτια με αυτοκόλλητο
- Οδηγό κατασκευών για [12 μοντέλα](#)

**Επίπεδο 2:** Τα παιδιά δίνουν κίνηση στο ρομπότ από απόσταση με το τηλεχειριστήριο και να κατανοήσουν τη χρήση των αισθητήρων.

22 πλαστικά πλαίσια (περιλαμβάνονται επιπλέον 2 κομμάτια που ενώνονται με LEGO DUPLO)

- 8 Διακοσμητικά κομμάτια από υλικό foam
- 1 Σετ διακοσμητικά μάτια με αυτοκόλλητο
- 11 Βίδες

- 14 Μπουλόνια
- 1 CPU βάση (Λειτουργεί ως εγκέφαλος του ρομπότ πάνω στην οποία συνδέονται η πλακέτα δέκτη, τα φωτάκια
- LED, οι αισθητήρες, ο σύρματος δέκτης, τα ηχείο με μελωδίες).
- 1 Τηλεχειριστήριο
- 1 Ασύρματο δέκτη
- 2 Αισθητήρες αφής (ανιχνεύουν την πίεση του κουμπιού)
- 1 Ηχείο με μελωδίες (παράγει διάφορους ήχους)
- 2 Καλώδια σύνδεσης
- Οδηγό κατασκευών για 12 νέα μοντέλα

Επίπεδο 3: τα παιδιά θα μάθουν να προγραμματίζουν το ρομπότ τους τοποθετώντας τα πλακίδια με τις εντολές πάνω στην πλακέτα κωδικοποίησης, με τη σειρά που θέλουν να τις εκτελέσει το ρομπότ

Το πακέτο περιλαμβάνει:

- 9 Πλαστικά πλαίσια (περιλαμβάνει 2 κομμάτια που ενώνονται με τουβλάκια LEGO DUPLO)
- 10 Βίδες
- 13 Μπουλόνια
- 2 Ρόδες
- 2 Διακοσμητικά κομμάτια από υλικό foam
- 1 Αισθητήρα κίνησης
- 1 Πλακέτα κωδικοποίησης
- 17 Πλακίδια εντολών
- 1 Καλώδιο σύνδεσης

- Οδηγό κατασκευών για 12 νέα μοντέλα

**Κόστος:179.80 Ευρώ**

### **Πρωτοβάθμια εκπαίδευση**

Σαν δεύτερη κατηγορία έρχονται πλατφόρμες οι οποίες χαρακτηρίζονται πλέον από την συνδεσιμότητα τους με υπολογιστή, tablet ή smartphone. Εδώ οι ηλικίες που αναφέρονται είναι για τις πρώτες τάξεις του δημοτικού και η πολυπλοκότητα του προγραμματισμού τους αυξάνεται και όπως καταλαβαίνουμε αυξάνονται και οι δυνατότητές τους. Κι εδώ τα ρομπότ είναι κυρίως δίτροχα χρησιμοποιώντας μία πολυ-κατευθυντική μπίλια για την στήριξή τους. Ο τρόπος προγραμματισμού τους είναι συνήθως με οπτικό προγραμματισμό. Κύριος εκφραστής αυτού του τρόπου είναι το Scratch. Υπάρχουν όμως κι άλλες γλώσσες όπου η φιλοσοφία τους διαφέρει λίγο από αυτή του Scratch.

### **Edison 2.0**



Εικόνα 8 Edison 2.0

Το ρομπότ Edison2.0, περιλαμβάνει ενσωματωμένους αισθητήρες απόστασης φωτός και ήχου, φώτα led και ήχους καθώς επίσης δίνει τη δυνατότητα αυτόνομης συμπεριφοράς. Μπορεί να ανταποκριθεί στο φως και τους ήχους, να ακολουθήσει μαύρες γραμμές και να αποφεύγει εμπόδια. Προγραμματίζεται εύκολα με το λογισμικό EdBlocks όπου είναι εξαιρετικά εύκολο ακόμα και για μικρές ηλικίες 8 - 12 χρονών. Το EdBlocks βασίζεται στο σύστημα drag and drop blocks και στην γλώσσα προγραμματισμού Scratch και Blockly, είναι δωρεάν και λειτουργεί σε

υπολογιστή ή tablet. Μπορεί επίσης να προγραμματιστεί με το λογισμικό EdPy. Το EdPy βασίζεται στην διαδεδομένη γλώσσα προγραμματισμού Python, είναι δωρεάν online εφαρμογή. Το ρομπότ edison είναι συμβατό και μπορεί να επεκταθεί με τα τουβλάκια Lego. Όπως το πακέτο EdCreate, το οποίο είναι ένα πακέτο κατασκευών το οποίο αποτελείται από 115 αλληλοσυνδεδεμένα δομικά στοιχεία, άξονες, γρανάζια και άλλα. Τα δομικά στοιχεία του EdCreate είναι σχεδιασμένα με τέτοιο τρόπο ώστε να είναι συμβατά με τα Edison robots και κάθε άλλο σύστημα δομικών στοιχείων συμβατό με LEGO. Κάθε πακέτο EdCreate μπορεί να χρησιμοποιηθεί για να κατασκευαστούν πέντε διαφορετικά EdBuilds καθώς και να υλοποιηθούν μια σειρά από δοκιμασίες σχεδιασμού και μηχανικής. Το πακέτο μπορεί επίσης να επεκταθεί χρησιμοποιώντας και άλλα δομικά στοιχεία συμβατά με LEGO. Το EdCreate μπορεί επίσης να αξιοποιηθεί σε ένα εύρος μηχανικών, σχεδιαστικών και προγραμματιστικών δραστηριοτήτων ανοιχτού τύπου. Η μάθηση μέσω project και δοκιμασιών, συμπεριλαμβανομένων και των EdChallenges που εμπεριέχονται στο EdCreate, προωθούν τη δημιουργική επίλυση προβλημάτων και δίνουν στους μαθητές την ευκαιρία να εφαρμόσουν λύσεις STEM σε προβλήματα εμπνευσμένα από τον πραγματικό κόσμο.

**Κόστος:45.00 Ευρώ**

## **WeDo 2.0 lego**



Εικόνα 9 WeDo 2.0 lego

Το σετ *WeDo 2.0 lego* αποτελεί την πιο πρόσφατη εκπαιδευτική πρόταση της LEGO Education. Το σετ παραδίδεται σε ένα κουτί αποθήκευσης και συνοδεύεται από ένα δίσκο διαλογής, αυτοκόλλητα για διευκόλυνση της ταξινόμησης, 1 Smarthub (Εγκέφαλος), 1 κινητήρα (Medium Motor), 1 αισθητήρα κίνησης (Motion Sensor), 1 αισθητήρα κλίσης (Tilt Sensor) και αρκετά δομικά υλικά.

Το λογισμικό WE DO 2, είναι της μορφής οπτικού προγραμματισμού που παρέχεται δωρεάν από τη LEGO, υποστηρίζεται από Laptops, σταθερούς Η/Υ, αλλά και Tablet, παρέχει ένα φιλικό και εύχρηστο προς τους μαθητές περιβάλλον προγραμματισμού, καθώς επίσης και πρόγραμμα σπουδών το οποίο δομείται από μαθήματα με θέματα φυσικής, επιστημών γης και διαστήματος, μηχανικής και πολλών άλλων επίσης δωρεάν. Παρέχεται επίσης το πρόγραμμα E-learning με βίντεο-μαθήματα ώστε να διευκολυνθεί η διαδικασία εκμάθησης των εκπαιδευτικών. Το we do 2 προγραμματίζεται επίσης και με την νεα έκδοση του scratch 3.Τέλος το wedo 2.0 είναι ιδανικό για συμμετοχή σε διαγωνισμούς Jr.FLL και WRO.

**Κόστος:190.00 Ευρώ**

### **Pro-Bot**



Εικόνα 10 Pro-Bot

Το Pro-Bot είναι το βασίζεται στην τεχνολογία Logo. Είναι ένα ρομπότ χελώνα, μεταμφιεσμένο ως αγωνιστικό αυτοκίνητο. Οι εντολές προγραμματισμού εισάγονται μέσω ενός συνόλου πλήκτρων βέλους και αριθμών που είναι τοποθετημένα στην πλάτη. Το Pro-Bot θα ακολουθήσει την ακολουθία των εντολών που εισήχθησαν βήμα προς βήμα.

Το Pro-Bot λειτουργεί σε δύο λειτουργίες. Ως "μεγάλος αδελφός" στο Bee-Bot, το Pro-Bot μπορεί να λειτουργήσει σε λειτουργία Bee-Bot χρησιμοποιώντας μόνο τα πλήκτρα βέλους και το πλήκτρο GO. Κάθε κίνηση και στροφή ενεργοποιείται σε ένα προκαθορισμένο ποσό. Στη λειτουργία Logo τα πλήκτρα βέλους μπορούν να συνδυαστούν με τα αριθμητικά πλήκτρα για εισαγωγή αποστάσεων για την κίνηση

και μοίρες για στροφές. Η ενσωματωμένη οθόνη LCD διευκολύνει τον προγραμματισμό στον πίνακα. Καθώς πατάμε τα πλήκτρα, στην οθόνη εμφανίζεται η αντίστοιχη εντολή Logo. Τα προγράμματα μπορούν να επεξεργαστούν και να τροποποιηθούν στην οθόνη LCD χρησιμοποιώντας τα πλήκτρα επεξεργασίας. Το Pro-Bot μπορεί να κρατήσει ένα στυλό στο κέντρο του, επιτρέποντάς του να σχεδιάζει καθώς κινείται. Το Pro-Bot έχει επίσης τη δυνατότητα αποθήκευσης υπο-διαδικασιών που μπορούν να ενσωματωθούν σε ένα πρόγραμμα Pro-Bot. Εκτός από τη μετακίνηση και την περιστροφή, το Pro-Bot διαθέτει αρκετούς ενσωματωμένους αισθητήρες. Οι εμπρός και πίσω προφυλακτήρες είναι αισθητήρες αφής και είναι εξοπλισμένο με αισθητήρες ήχου και φωτός. Το Pro-Bot λειτουργεί με 3 μπαταρίες AA και έρχεται πλήρες και έτοιμο σε ένα κατάλληλο κουτί αποθήκευσης. Ένας οδηγός χρήσης παρέχει πληροφορίες σχετικά με τη λειτουργία και τη φροντίδα του Pro-Bot.

***Κόστος:159.90 Ευρώ***

## **Thymio**



Εικόνα 11 Thymio

Η πλατφόρμα Thymio, βασίζεται στο φως και στην αφή, ενώ το περιβάλλον προγραμματισμού δίνει στα παιδιά την ευκαιρία να προγραμματίσουν, τόσο σε γραφικό περιβάλλον, όσο και με γραμμή εντολών. Ενσωματώνει μια πληθώρα αισθητήρων σε ένα επιδαπέδιο ρομπότ κλειστού τύπου. Έχει αισθητήρες υπέρυθρων, θερμοκρασίας, ήχου και επιτάχυνσης οι οποίοι δίνουν στους μαθητές τη δυνατότητα να προγραμματίσουν το ρομπότ να κινηθεί σε ένα χώρο αποφεύγοντας εμπόδια ή να



ακολουθήσει μια μαύρη γραμμή, ενώ αν υπάρχουν περισσότερα από ένα τα Thymio μπορούν να επικοινωνήσουν μεταξύ τους και να συνεργαστούν. Το Thymio έχει ήδη αρχικές συμπεριφορές με τις οποίες έχει προγραμματιστεί και του επιτρέπουν να χρησιμοποιηθεί μόλις βγει από το κουτί, ακολουθώντας οδηγίες ή αποφεύγοντας εμπόδια που θα βρει μπροστά του, ενώ μπορεί να τηλεκατευθυνθεί με το χειριστήριο υπερύθρων. Ο προγραμματισμός του γίνεται σε οποιοδήποτε λειτουργικό σύστημα μέσα από το γραφικό περιβάλλον Aseba, ενώ είναι συμβατό με Blockly καθώς και με Scratch. Τα παιδιά με αυτόν τον τρόπο έχουν τη δυνατότητα να εξελίξουν τις προγραμματιστικές τους δεξιότητες μέσα από μια σειρά από δραστηριότητες αλλά και αξιοποιώντας τις θεματικές πίστες που διατίθενται για το Thymio, αναπτύσσοντας τη φαντασία και τη δημιουργικότητά τους.

#### Χαρακτηριστικά

- Περισσότεροι από 20 Αισθητήρες
- 40 φώτα
- 2 κινητήρες
- Συμβατό με τουβλάκια LEGO
- Δυνατότητα Τηλεχειρισμού
- Έτοιμα προγράμματα και δυνατότητα προγραμματισμού
- Ισχυρή κοινότητα με περισσότερες από 90 δημοσιευμένες εργασίες
- Επαναφορτιζόμενη μπαταρία, η οποία φορτίζει όσο προγραμματίζουμε
- Φορητό, μπορείτε να το πάρετε παντού μαζί σας!

**Κόστος: 111.30 Ευρώ**

## BinaryBots-Meet DIMM kit



Εικόνα12 BinaryBots-Meet DIMM kit

Το BinaryBots-Meet DIMM kit περιλαμβάνει το ολοκαίνουργιο BBC microbit, μια πλακέτα που είναι σε ένα Microprocessor: 32-bit ARM® Cortex™ M0 CPU. Επιπλέον, περιλαμβάνει αισθητήρες φωτός, κίνησης και θερμοκρασίας. Διαθέτει ελληνικό εικονογραφημένο εγχειρίδιο, και μπορεί να συνδεθεί ενσύρματα με το PC ή ασύρματα με το smartphone και το tablet μέσω Bluetooth. Πάνω σε αυτό μπορείς να συνδέσουμε τους αισθητήρες φωτός, κίνησης και θερμοκρασίας. Παράλληλα, δίνεται η δυνατότητα δωρεάν πρόσβασης σε online δραστηριότητες και βίντεο ώστε να μάθει ο μαθητής να γράφει κώδικα σε Scratch, Microsoft Block Editor, Python, JavaScript and Microsoft PXT or Touch Develop.

**Κόστος:40.32Ευρώ**

## Gigo S4A Programming Bricks



Εικόνα.13GigoS4AProgrammingBricks

Το πακέτο Gigo S4A Programming Bricks δίνει τη δυνατότητα στους μαθητές να κατασκευάσουν και να προγραμματίσουν ρομποτικά μοντέλα μηχανών. Η πλατφόρμα, περιλαμβάνει αισθητήρες φωτός, υπέρυθρων, δύο σερβομοτέρ περιστροφής 360° & και ένα 180° καθώς και ένα κουτί ελέγχου (πλακέτα) συμβατό με Arduino Leonardo το οποίο συνδέεται με τον Η/Υ και δρα ως “εγκέφαλος” της κατασκευής. Το Arduino Leonardo που περιλαμβάνει το πακέτο, προγραμματίζεται με το γραφικό περιβάλλον [S4A \(Scratch for Arduino\)](#). Τα εκατοντάδες δομικά στοιχεία που περιλαμβάνονται, δίνουν τη δυνατότητα κατασκευής μιας πληθώρας μοντέλων, από απλά ρομποτικά οχήματα 2 τροχών, μέχρι πολύπλοκες κατασκευές με σύνθετες λειτουργίες. Παρέχεται δωρεάν οδηγός με τα αναλυτικά σχέδια μαθήματος και τα παραδείγματα προγραμμάτων.

**Κόστος: 189.92 Ευρώ**

## Robot Dash Workshop Wonder



Εικόνα 13 Robot Dash Workshop Wonder

Το Robot Dash Workshop Wonder είναι ένα ρομπότ που μπορεί να αντιληφθεί το περιβάλλον του και μπορεί να προγραμματιστεί ασύρματα χρησιμοποιώντας ένα tablet. Μπορεί να κινηθεί αποφεύγοντας εμπόδια, να καταλάβει ένα κάποιος το κινεί, να λάμψει με διάφορα χρώματα, να κάνει ήχους και να αλληλοεπιδράσει με άλλα παρόμοια ρομπότ. Διαθέτει δύο κινητήριους τροχούς και μία μπάλα ισορροπίας που του επιτρέπει να κινείται προς τα εμπρός, προς τα πίσω, και να περιστρέφεται και ακόμη δύο σερβοκινητήρες που του επιτρέπουν να κινεί το κεφάλι του. Από αισθητήρες έχει τρεις απόστασης και ένα ήχου και ένα επιτάχυνσης και γυροσκόπιο. Το Dash προγραμματίζεται με οπτικό προγραμματισμό, τη Blockly. Η οποία είναι μια drag-and-drop γλώσσα προγραμματισμού που συνενώνει τις εντολές σαν κομμάτια παζλ, παρόμοια με το Scratch. Το Dash μπορεί να δεχτεί και κάποια πρόσθετα, όπως το Dashes Xylophone και να παίζει ένα τραγούδι, να τραβήξουμε βίντεο χρησιμοποιώντας τη κάμερα του smartphone, να εκτοξεύσει μπαλάκια με ένα καταπέλτη και να προσθέσουμε τουβλάκια με τα Building Brick Connectors δίνοντας του ένα πιο προσωπικό χαρακτήρα.

**Κόστος: 179.00 Ευρώ**

## Δευτεροβάθμια εκπαίδευση

Στην τρίτη κατηγορία η οποία ανήκει σε παιδιά δευτεροβάθμιας εκπαίδευσης όπως γίνεται εύκολα αντιληπτό η πολυπλοκότητα ανεβαίνει σε επίπεδο κατασκευής αλλά και προγραμματισμού. Αυτό σημαίνει και περισσότερες δυνατότητες για τους χρήστες των πακέτων. Εδώ υπάρχουν πλατφόρμες οι οποίες δεν επιτρέπουν στον χρήστη να επέμβει στο ηλεκτρονικό κομμάτι της κατασκευής π.χ. Lego Mindstorm αλλά και εντελώς στην αντίθετη πλευρά όπως το Alphanot. Από πλευράς προγραμματισμού κι εδώ υπάρχουν επιλογές σε σχέση με την ανοικτότητα του λογισμικού που χρησιμοποιείται. Πχ το Lego Mindstorm που προγραμματίζεται με λογισμικό της εταιρίας και το Alphanot που προγραμματίζεται με ανοικτό λογισμικό.

## Neuron



Εικόνα 14 Neuron

Το κιτ Neuron προσφέρει πάνω από 30 ηλεκτρονικά Block, όπως αισθητήρα φωνής, αισθητήρα φωτός και αισθητήρα υπερύθρων. Τα Neuron Blocks συνδέονται μεταξύ τους μέσω μαγνητών δημιουργώντας έτσι γρήγορες αλλά και ασφαλείς συνδέσεις. Επίσης το σύστημα μαγνητών αποτρέπει οποιαδήποτε λανθασμένη σύνδεση. Το Neuron συνοδεύεται με το λογισμικό Neuron App μέσω του οποίου τα παιδιά

μπορούν να δώσουν ζωή στις δημιουργίες τους μέσα σε λίγα δευτερόλεπτα. Επίσης μπορεί αν συνδυαστεί με το mBlock 5, για γραφική γλώσσα προγραμματισμού που αναπτύχθηκε βάσει του Scratch 3.0. Επίσης προχωρημένα χαρακτηριστικά όπως Wi-Fi, Κάμερες και ηχεία, σε συνδυασμό με το Neuron App ή το mBlock 5, δίνουν τη δυνατότητα στα παιδιά να δημιουργήσουν πολύπλοκες εφαρμογές σχεδιάζοντας αυτοματισμούς καθώς και πρώιμα ευφυή συστήματα.

***Κόστος:129.90 Ευρώ***

## **mBot**



Εικόνα 15 mBot

Το mBot αποτελεί μια εκπαιδευτική πρόταση για εισαγωγή στον προγραμματισμό και την ηλεκτρονική. Οι δυνατότητές του με έξτρα υλικά καθώς και η ευκολία της συναρμολόγησής του το καθιστούν μια ολοκληρωμένη λύση για την εισαγωγή στο STEM. Η συναρμολόγηση μπορεί να γίνει σε μικρό χρόνο. Περιλαμβάνει 38 εξαρτήματα για τη συναρμολόγηση του ρομπότ. Η σχεδιάσή του έχει γίνει με τέτοιο τρόπο ώστε η καλωδίωση να είναι εύκολη μέσω των χρωματιστών θυρών του. Τα μηχανικά μέρη του mBot είναι πλήρως συμβατά με όλα τα εξαρτήματα της Makeblock καθώς και με εξαρτήματα της LEGO®. Τα ηλεκτρονικά μέρη του mBot

βασίζονται στο Arduino. Αποτέλεσμα αυτών των δύο, η μεγάλη επεκτασιμότητα και οι δυνατότητες που προσφέρονται για περαιτέρω ανάπτυξη.

### **Χαρακτηριστικά:**

- Είσοδοι: Αισθητήρας φωτός, Button, Δέκτης υπέρυθρων, Αισθητήρας απόστασης, Αισθητήρας ακολουθίας γραμμής
- Έξοδοι: Buzzer, πολύχρωμο LED, Πομπός υπέρυθρων, δύο κινητήρες, επιπλέον θύρες
- Μικροελεγκτής: Βασισμένο στον Arduino Uno
- Τροφοδοσία: 3.7V Μπαταρία λιθίου (περιλαμβάνει φορτιστή ενσωματωμένος στην πλακέτα) ή Μπαταρίες 4xAA (Η μπαταρία δεν περιλαμβάνεται)
- Ασύρματη Επικοινωνία: Bluetooth
- 17 x 13 x 9 cm (συναρμολογημένο)
- Βάρος 1034g (συναρμολογημένο)

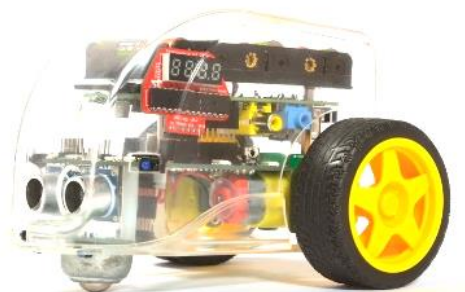
### **Το σετ περιλαμβάνει**

- 1 × Πλατφόρμα Αλουμινίου χρώματος μπλέ
- 1 × mCore Κεντρική πλακέτα
- 1 × mCore κουτί
- 1 × Bluetooth
- 1 × Me Αισθητήρα ακολουθίας γραμμής
- 1 × Me Αισθητήρα απόστασης/αποφυγής εμποδίων
- 1 × Πίστα ακολουθίας γραμμής
- 1 × AA Μπαταριοθήκη
- 1 × Κουτί για μπαταρία λιθίου
- 2 × Κινητήρες DC

- 2 × Ρόδες με λάστιχο
- 1 × Ρόδα εμπρός
- 2 × RJ25 Καλώδια
- 2 × Velcro Αυτοκόλλητο
- 1 × Κατσαβίδι
- 4 × M4x25mm Αποστάτες
- 1 × USB Καλώδιο
- 15 × M4x8mm Βίδες
- 4 × M2.2x9mm Βίδες
- 8 × M3 Παξιμάδια
- 6 × M3x25mm βίδες
- 1 × Τηλεχειριστήριο IR

***Κόστος: 99.90 Ευρώ***

### **Pi2Go Raspberry Pi Programmable Floor Robot**



Εικόνα16 Pi2Go Raspberry Pi Programmable Floor Robot



Pi2Go Raspberry Pi Programmable Floor Robot Είναι ένα προγραμματιζόμενο ρομπότ το οποίο σχεδιάστηκε ώστε να αναδείξει την δύναμη που κρύβει το Raspberry Pi. Συμβατό με τα μοντέλα από το B+ έως το 3, μπορεί να προγραμματιστεί μέσω του Scratch ή της Python. Το Pi2Go είναι προ- συναρμολογημένο, το μόνο που χρειάζεται είναι να τοποθετηθεί το Raspberry Pi και να προγραμματιστεί. Το διάφανο κάλυμμα του προστατεύει τα εξαρτήματά του αλλά ταυτόχρονα επιτρέπει στο χρήστη να δει και στο εσωτερικό του για το τι ακριβώς συμβαίνει καθώς το ρομπότ ακολουθεί τις οδηγίες το οποίο προγραμματίστηκε .

Χαρακτηριστικά:

- Δυνατότητα παρακολούθησης εισόδων – εξόδων (line follower, ultra-sonic sensor, RGB LEDs, motors, light sensor)
- Αισθητήρες εισόδου που επιτρέπουν το χειρισμό μεταβλητών όπως απόσταση και φωτεινότητα
- Επικοινωνία μέσω Wi-Fi ή απευθείας μέσω USB
- Συμβατό με Scratch και Python
- Εύκολη προσαρμογή κάμερας του Raspberry Pi
- Προ συναρμολογημένο
- Διάφανο προστατευτικό κάλυμμα

Το σετ προϋποθέτει να υπάρχει ένα Raspberry Pi, ώστε να μπορείτε να το συνδέσετε με το Pi2Go γιατί δεν παρέχεται μαζί με το kit.

***Κόστος:141.13Ευρώ***

## BQ PrintBot EvolutionH



Εικόνα 17 BQ PrintBot EvolutionH

BQ PrintBot EvolutionH παρουσιάζει μια νέα ρομποτική πλατφόρμα για παιδιά και μεγάλους. Με αυτή θα μπορέσετε να συναρμολογήσετε ένα πλήρες ρομπότ, να μάθετε ηλεκτρονικά, προγραμματισμό και να διασκεδάσετε με την εξέλιξή του με δικές σας ιδέες και προσθήκες.

Το πακέτο περιλαμβάνει όλα τα μηχανικά και ηλεκτρονικά μέρη αλλά και τα 3D εκτυπωμένα τμήματα του σκελετού. Εάν έχουμε 3D εκτυπωτή μπορούμε να σχεδιάσουμε και να εκτυπώσουμε δικό μας σκελετό.

Συνδέοντάς το με ένα tablet και την εφαρμογή της BQ, μπορούμε να το ελέγχουμε και να το οδηγούμε. Τα ενσωματωμένα ηλεκτρονικά μας δίνουν τη δυνατότητα για διάφορους αυτοματισμούς και τρόπους λειτουργίας του ρομπότ, όπως το να ακολουθεί μια μαύρη γραμμή, να ακολουθεί το φως ή και να αποφεύγει εμπόδια.

Το κιτ περιλαμβάνει.

- Τρισδιάστατο πλαίσιο
- Σώμα (κεντρικό τεμάχιο για την τοποθέτηση τόσο των συνεχών κινητήρων περιστροφής, του βομβητή και του πίνακα ZUM)
- 2 τροχούς
- Υποστήριξη μίνι σερβο
- Μπαταρία
- 2 καλύμματα της κατασκευής
- Υποστήριξη υπερήχων
- 2 πτερύγια τροχών
- 2 σερβοκινητήρες συνεχούς περιστροφής
- 2 αισθητήρες φωτός ZUMbloq
- Miniservo
- Καλώδιο USB
- Πίνακας ελεγκτών ZUM BT-328
- Μίνι σέρβο
- ZUMbloq βομβητή
- ZUMbloq υπερέθρων 2 ακολουθία γραμμή
- Αισθητήρας υπερήχων
- Καλώδιο USB

Άλλα εξαρτήματα

- 4 Εξαγωνικοί διαχωριστές 15mm με M3 σε ένα από τα άκρα
- Μεθακρυλική βάση

- 6 βίδες M3 x 12mm
- 34 παξιμάδια M3
- 26 βίδες M3 x 8mm
- 2 βίδες M3 x 25mm
- 2 δακτύλιοι στεγανοποίησης (πάχος 4 mm και εσωτερική διάμετρος 56 mm)
- κλειδί άλεν

**Κόστος:82.56Ευρώ**

### **LEGO Education MINDSTORMS EV3**



**Εικόνα19LEGOEducationMINDSTORMSEV3**

Η πλατφόρμα LEGO Education MINDSTORMS EV3 αποτελεί την πιο διαδεδομένη λύση εκπαιδευτικής ρομποτικής. Κατάλληλη για έμπειρους αλλά και για νέους δημιουργούς. Με το συγκεκριμένο κιτ οι μαθητές εκτός των γνώσεων

προγραμματισμού και ρομποτικής που παίρνουν μπορούν να ασχοληθούν και με τον τομέα της μηχανικής σε μεγάλο βαθμό. Το πακέτο χαρακτηρίζεται από την μεγάλη δυνατότητα αλλαγών της κατασκευής του ρομπότ, μιας κι όλη η κατασκευή στηρίζεται σε κομματάκια LEGO τα οποία συνδυάζονται με διάφορους τρόπους. Ο εγκέφαλος του μπορεί να συνδεθεί ταυτόχρονα μέχρι τέσσερις κινητήρες και τέσσερις αισθητήρες. Ο προγραμματισμός του γίνεται μέσω του λογισμικού που παρέχει η εταιρία και είναι του τύπου του οπτικού προγραμματισμού αποτελεί και στηρίζεται στο LABVIEW. Το πακέτο μπορεί να προγραμματιστεί και από Python, Basic, με τα κατάλληλα extensions αλλά κι άλλες γλώσσες. Το συγκεκριμένο πακέτο είναι κατάλληλο και για τους διαγωνισμούς ρομποτικής FLL και WRO.

Το βασικό πακέτο περιλαμβάνει

- Εγκέφαλος
- δυο μεγάλους σερβοκινητήρες
- ένα μικρό σερβοκινητήρα
- ένα αισθητήρα υπερήχων
- ένα αισθητήρα χρώματος φωτός
- ένα αισθητήρα στροφής
- δύο αισθητήρες επαφής
- 532 Lego bricks

Επιτρέπει Bluetooth και Wi-Fi επικοινωνία και παρέχει συλλογή δεδομένων και προγραμματισμό.

Στο πακέτο περιλαμβάνονται γραπτές οδηγίες κατασκευής του βασικού μοντέλου και 4 επιπλέον μοντέλων στο λογισμικό.

Το βασικό πακέτο LEGO Education MINDSTORMS EV3 περιλαμβάνει ένα ειδικό πλαστικό κάδο με θήκες διαχωρισμού για εύκολη χρήση στην τάξη και οργάνωση. Είναι συμβατό με τους αισθητήρες Hi-Technic.

***Κόστος: 480.00 Ευρώ***

## VEX Classroom and Competition Super Kit



Εικόνα18 VEX Classroom and Competition Super Kit

Το VEX Classroom and Competition Super Kit είναι μία ρομποτική πλατφόρμα με φιλοσοφία σαν και αυτά της LEGO αλλά με εξαρτήματα που απαιτούν πιο λεπτό χειρισμό. Οι συνδέσεις των εξαρτημάτων για την κατασκευή των ρομπότ δεν γίνονται με σφήνες αλλά με βίδες και αυτό τα κάνει πιο ανθεκτικά αλλά αυξάνει τη δυσκολία συναρμολόγησης και μετέπειτα μετατροπής τους. Το λογισμικό προγραμματισμού μπορούμε να το κατεβάσουμε μετά από την εγγραφή του στον ιστότοπο της VEX. Η εταιρία παρέχει πρόγραμμα σπουδών το VEX EDR το οποίο ξεκινά από τα βασικά της μηχανικής, και της ρομποτικής. Επίσης το πακέτο μπορεί να χρησιμοποιηθεί και για τους ρομποτικούς αγώνες με VEX ρομπότ.

Το κιτ περιλαμβάνει:

- VEX Cortex Microcontroller

- Motor Controllers 29
- τηλεχειριστήριο (joystick)
- αισθητήρες επαφής
- Διακόπτες
- Ποτενσιόμετρα
- αισθητήρα ακολούθησης γραμμής
- αισθητήρα υπερήχων
- αισθητήρες οπτοδιακοπής για έλεγχο των κινητήρων
- κινητήρες
- τροχούς απλούς και σουηδικούς τροχούς
- γρανάζια και αλυσίδες
- αρκετά υλικά συναρμολόγησης όπως άξονες, δοκάρια, βίδες κτλ

**Κόστος:931.00 Ευρώ**

### **AlphaBot**



Εικόνα 19 AlphaBot

Το AlphaBot είναι μια ρομποτική πλατφόρμα ανάπτυξης συμβατή με το Raspberry Pi και το Arduino. Αποτελείτε από δύο κινητήριους τροχούς και έναν περιστρεφόμενο τροχό (σφαιρικό τροχό) για ευθυγράμμιση και ισορροπία. Περιλαμβάνει υποστήριξη διπλής πλακέτας. Ο εγκέφαλος του μπορεί να είναι μια πλακέτα Arduino ή μια πλακέτα Raspberry Pi ή για πιο απαιτητικές εφαρμογές, και τις δύο μαζί. Επιτρέπει

δηλαδή τη διανομή εργασιών. Την πιο δύσκολη υπολογιστικά για το Raspberry Pi (διαχείριση της κάμερας και των ασύρματων ζεύξεων RF), ενώ ο Arduino μπορεί να φροντίσει τη διασύνδεση με οτιδήποτε άλλο π.χ. τον έλεγχο των κινητήριων τροχών ή την ανάγνωση κάποιων αισθητηρίων. Ο προγραμματισμός του γίνεται ένα υπάρχει μόνο το Arduino με τη γλώσσα Wiring η οποία είναι C++ like ενώ εάν υπάρχει και το Raspberry Pi ή μόνο αυτό ο προγραμματισμός μπορεί να γίνει και σε Python. Η εταιρία που το εμπορεύεται παρέχει διάφορα παραδείγματα κώδικα μαζί με το AlphaBot, μερικά από τα οποία είναι δοκιμές για αισθητήρες και ηλεκτρονικά και μηχανικά μέρη όπως οι κινητήρες, ορισμένα άλλα είναι πλήρη προγράμματα βασικά για την έναρξη της ανάπτυξης του ρομπότ σας. Για τη λειτουργία του ,μπορεί να χρησιμοποιεί και η βιβλιοθήκη AlphaBot, η οποία μπορεί να μεταφορτωθεί από την επίσημη ιστοσελίδα του AlphaBot και περιέχει αναγκαία κομμάτια κώδικα για την λειτουργία του ρομπότ. Η όλη κατασκευή είναι open source και αυτό μας δίνει την δυνατότητα να επεμβούμε και να κάνουμε τις δικές μας προσθήκες, βελτιώσεις .

Το βασικό kit περιέχει :

- Kit σασί AlphaBot:
- mainboard
- τροχούς
- οδηγούς
- βίδες
- 5 IR Αισθητήρες
- Μονάδα παρακολούθησης γραμμής
- Αισθητήρας διακοπής φωτός
- Μονάδα μέτρησης ταχύτητας (2τμ)
- Υπέρυθρος αισθητήρας εγγύτητας
- Μονάδα αποφυγής εμποδίων (2 τεμάχια)
- Αισθητήρας υπερήχων.
- IR τηλεχειριστήριο για έλεγχο εξ αποστάσεως του ρομπότ
- LM298P Η γέφυρα για οδηγό των κινητήρων



- TLC1543:10 bit AD, που επιτρέπει στο Pi να χρησιμοποιεί αναλογικούς αισθητήρες
- Θήκη μπαταρίας: δέχεται δύο 18650 μπαταρίες.
- SG90 servo x2
- 2 DOF pan and tilt kit x1
- IR τηλεχειριστήριο υπερύθρων x1

**Κόστος: 89.90 Ευρώ**

### **Τριτοβάθμια εκπαίδευση**

Τελευταία κατηγορία εκπαιδευτικών ρομπότ είναι αυτά που προορίζονται για χρήση σε προπτυχιακό επίπεδο στην τριτοβάθμια εκπαίδευση. Εδώ οι λύσεις είναι επίσης πάρα πολλές και με κόστη που μπορούν να ανέβουν αρκετά πάνω από τα 1000 Ευρώ. Το κύριο χαρακτηριστικό εδώ είναι η έρευνα οπότε και οι πλατφόρμες υποστηρίζουν αναπτυξιακά εργαλεία για πιο πολύπλοκες εργασίες σε σχέση με τις προηγούμενες κατηγορίες. Εδώ π.χ. οι αισθητήρες απόστασης που είδαμε πριν να χρησιμοποιούνται μπαίνουν σε δεύτερη μοίρα και την θέση τους την παίρνουν Lidar τα οποία χρησιμοποιούν αλγόριθμους χαρτογράφησης. Επίσης χρησιμοποιούνται κάμερες όχι απλά για να μεταφέρουν την εικόνα σε κάποιο απομακρυσμένο υπολογιστή με σκοπό τον έλεγχο του ρομπότ αλλά χρησιμοποιούν αλγόριθμους αναγνώρισης εικόνας για την αυτόνομη κίνησή τους. Εδώ βλέπουμε και ρομπότ με ρομποτικούς βραχίονες όπου το ζητούμενο είναι η κατεύθυνση του άκρου του βραχίονα σε συγκεκριμένες θέσεις μέσα από συγκεκριμένες τροχιές, με σκοπό την επιτέλεση κάποιας κίνησης.

## Arlo



Εικόνα 20 Arlo

Το Arlo είναι μια χερσαία πλατφόρμα ρομπότ που υποστηρίζει πολύπλοκες καθοδηγούμενες ή αυτόνομες αποστολές. Περιλαμβάνει όλα τα μηχανικά και ηλεκτρονικά εξαρτήματα, για τον έλεγχο της κίνησης του, καθώς και τα καλώδια διασύνδεσης, το υλικό, ακόμα και οι 12 V μπαταρίες και το φορτιστή. Αυτό το kit είναι σχεδιασμένο για χρήση με τον ελεγκτή της δικιάς μας επιλογής, περιέχει όμως κι έναν ελεγκτή τον Propeller Activity Board WX και καλώδιο προγραμματισμού. Μπορεί να συνεργαστεί με διάφορες πλακέτες όπως Arduino, Raspberry κτλ. Η βασική του λειτουργία είναι να μπορεί να δεχτεί διάφορα εξαρτήματα επάνω του και να κινείται σε εσωτερικούς χώρους. Η αυτονομία του εξαρτάται από τον κυρίως ελεγκτή που θα του προσθέσουμε και τι αισθητήρες μπορεί αυτός να δεχτεί.

Το kit περιέχει :

- σασί
- 2 Caster τροχούς
- 2 κινητήρες με τις βάσεις τους
- πάνω βάση
- πλακέτα διανομής ρευμάτων για επεκτάσεις
- [Propeller Activity Board WX](#) πλακέτα ελεγκτή

- [USB A to Mini B Cable](#)
- Βίδες και βάσεις
- Φορτιστή της μπαταρίας
- Dual H-Bridge ελεγκτής κινητήρων
- 4 Αισθητήρες υπερήχων με τις βάσεις τους
- 12 Volt, 7.2Ah Μπαταρία
- DVM810 Πολύμετρο
- 3M Γυαλιά Ασφαλείας

***Κόστος:846.80 Ευρώ***

### **SDP Mini RPlidar 2WD**



**Εικόνα 21SDPMiniRPlidar 2WD**

Η πειραματική πλατφόρμα ρομπότ SDP Mini RPlidar 2WD είναι μια μικρή πλατφόρμα ανάπτυξης που σχεδιάστηκε από την Slamtec με την οποία οι χρήστες μπορούν να έχουν μια γενική κατανόηση σχετικά με τη λύση Slamware η οποία αναπτύχθηκε από την εταιρία και να εξετάσουν τη βασική της απόδοση. Το SDP Mini είναι μια συμπαγής κατασκευή κατάλληλη για την προκαταρκτική αξιολόγηση

της λύσης Slamware ή την ανάπτυξη ρομπότ με βασικές ανάγκες τον εντοπισμό και τη λειτουργία πλοήγησης του σε αχαρτογράφητο περιβάλλον. Χρησιμοποιεί ένα lidar αισθητήρα και την πλακέτα Slamware η οποία περιέχει μαγνητική πυξίδα και επιχυνσιόμετρα και μπορεί να δεχτεί δεδομένα επίσης από κάμερα βάρους, αισθητήρες υπερήχων, κλίσης και επαφής. Τα βασικά εξαρτήματα του είναι, το Slamware Core, το RPLidar A2 πλακέτα για κοντρόλ των τροχών, και απαιτεί μόνο 8 μπαταρίες AA. Το SDP Mini, υποστηρίζεται για χρήση σε περιβάλλον Windows / Linux / IOS και Android, ενσωματώνει ολόκληρη τη λύση Slamware και πρόσθετες διεπαφές αισθητήρων.

**Κόστος: 484.36 Ευρώ**

## **TurtleBot**

Το TurtleBot είναι ένα κιτ ρομπότ με λογισμικό ανοικτής πηγής χαμηλού κόστους. Το TurtleBot δημιουργήθηκε στο Willow Garage από τον Melonee Wise και τον Tully Foote τον Νοέμβριο του 2010. Το TurtleBot είναι λειτουργεί με την πλατφόρμα ROS η οποία διατηρείται από το Ίδρυμα Ρομποτικής Ανοικτού Κώδικα ([Open Source Robotics Foundation](#)). Αποτελείται από μια κινητή βάση, έναν αισθητήρα 2D ή 3D, ένα υπολογιστή και το κιτ εξοπλισμού TurtleBot. Εκτός από το κιτ TurtleBot, οι χρήστες μπορούν να κάνουν λήψη του TurtleBot SDK (Software Development Kit) από το wiki του ROS. Το TurtleBot έχει σχεδιαστεί ώστε να είναι εύκολο να αποκτηθεί, να συναρμολογηθεί, και να χτίσουμε ρομποτικές εφαρμογές επάνω του. Ως πλατφόρμα κινητής ρομποτικής, TurtleBot διαθέτει πολλές από τις ίδιες δυνατότητες των μεγαλύτερων πλατφορμών ρομποτικής της εταιρείας, όπως το PR2, Care-O-Bot. Με το TurtleBot, οι χρήστες μπορούν να οδηγήσουν και να χαρτογραφήσουν το περιβάλλον τους, να δουν σε 3D και να έχουν μεγάλη ελευθερία για να δημιουργήσουν τις δικές τους εφαρμογές. Ανάλογα την έκδοσή του μπορεί να έχει ή ένα αισθητήρα LIDAR ή μια κάμερα KINECT για χαρτογράφηση και εντοπισμό της θέσης του. Όπως και με πολλές άλλες πλατφόρμες ROS, ένα από τα μεγαλύτερα πλεονεκτήματα του TurtleBot είναι η κοινότητα υποστήριξής του. Το TurtleBot έχει μια παγκόσμια κοινότητα η οποία αυξάνεται συνεχώς με χιλιάδες TurtleBots σε όλο τον κόσμο. Το TurtleBot έχει υιοθετηθεί από πολλά ερευνητικά εργαστήρια για την έρευνα πολλαπλών ρομπότ και την έρευνα αλληλεπίδρασης

ανθρώπων με ρομπότ. Επιπλέον, πολλά πανεπιστήμια χρησιμοποιούν το TurtleBot για να διδάξουν εισαγωγικά μαθήματα ρομποτικής.

Οι εκδόσεις TurtleBot είναι οι εξής :



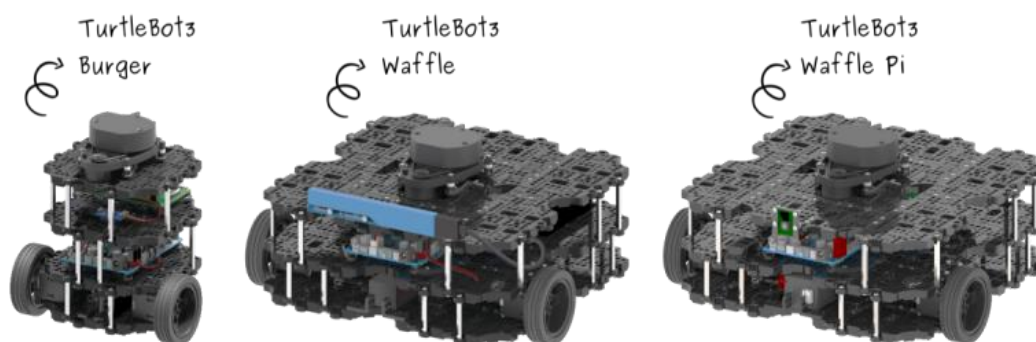
Εικόνα 22 TurtleBot

Το TurtleBot1 το οποίο αποτελείται από βάση iRobot Create, μπαταρία 3.000mAh, πλακέτα TurtleBot με γυροσκόπιο, αισθητήρα Kinect, φορητό υπολογιστή Asus 1215N με επεξεργαστή διπλού πυρήνα και κιτ τοποθέτησης υλικού που συνδέει τα πάντα μαζί και έχει υποδοχές για να προσθέσουμε μελλοντικά κι άλλους αισθητήρες. Το πρώτο TurtleBot δημιουργήθηκε στο Willow Garage από τον Melonee Wise και τον Tully Foote τον Νοέμβριο του 2010 Δεν διατίθεται πλέον.



Εικόνα 23 TurtleBot2

Το TurtleBot2 αποτελείται από μια βάση Yujin Kobuki, μια μπαταρία 2.200mAh, έναν αισθητήρα Kinect, έναν φορητό υπολογιστή Asus 1215N με επεξεργαστή διπλού πυρήνα, γρήγορο φορτιστή, βάση φόρτισης και ένα κιτ τοποθέτησης υλικού ο οποίος συνδέει όλα τα ηλεκτρονικά μαζί και έχει υποδοχές για να προσθέσουμε μελλοντικά κι άλλους αισθητήρες. Υπάρχει και σε έκδοση με ρομποτικό βραχίονα. Το Turtlebot2 κυκλοφόρησε τον Οκτώβριο του 2012.



Εικόνα 24 TurtleBot3

Το TurtleBot3 αποτελείται από αρθρωτές πλάκες όπου οι χρήστες μπορούν να προσαρμόσουν το σχήμα του ρομπότ τους. Διατίθεται σε τρεις τύπους: Burger μικρού μεγέθους και βαρύ μεσαίο μέγεθος, Waffle και Waffle Pi. Το TurtleBot3 αποτελείται από μία βάση, δύο κινητήρες Dynamixel, μια μπαταρία 1.800mAh, ένα LIDAR 360 μοιρών, μια κάμερα (κάμερα RealSense για το Waffle, Pi κάμερας για το Waffle Pi), ένα SBC (single board computer: Raspberry PI 3 και Intel Joule 570x) και ένα κιτ σύνδεσης όλων των ηλεκτρονικών του ρομπότ και έχει υποδοχές για να προσθέσουμε μελλοντικά κι άλλους αισθητήρες. Το Turtlebot3 κυκλοφόρησε τον Μάιο του 2017.

**Κόστος: 486-576 Ευρώ**

## Eddie



Εικόνα 25 Eddie

Το ρομπότ Eddie είναι ένα αυτόνομο κινητό ρομπότ που μεταφέρει ένα laptop και ένα Microsoft Kinect. Το ρομπότ Eddie είναι η πλατφόρμα αναφοράς που επέλεξε η Microsoft για την κυκλοφορία του νέου της Microsoft Robotics Studio που ενσωματώνει το Kinect SDK και επιτρέπει τη χρήση ισχυρών αλγορίθμων που παρέχονται από τη Microsoft που μπορούν να επωφεληθούν από την κάμερα Kinect όσον αφορά την αναγνώριση 3D ανθρώπινων σχημάτων. Το ρομπότ του Eddie πωλείται μη συναρμολογημένο, χωρίς την κάμερα Kinect και χωρίς φορητό υπολογιστή. Το ρομπότ Eddie είναι πλήρως συμβατό με το Microsoft Robotics



Developer Studio 4. Το λογισμικό προγραμματισμού είναι δωρεάν και μπορείτε να το κατεβείτε από την ιστοσελίδα του Microsoft Robotics Studio για να προγραμματιστεί το ρομπότ. Το Microsoft Robotics Developer Studio είναι ένα πλήρες σύνολο λογισμικού με το IDE (Visual Studio), ένα γραφικό περιβάλλον προγραμματισμού (Microsoft Visual Language), έναν προσομοιωτή (Microsoft Visual Environment) και το παράλληλο μοντέλο προγραμματισμού: CCR / DSS.

Τεχνικά χαρακτηριστικά του ρομπότ:

Σχεδίαση διπλής πλατφόρμας με άφθονο χώρο για ηλεκτρονικά και πρόσβαση στο lap-top για τον έλεγχο του ρομπότ. Δύο κινητήριои τροχοί με κινητήρες 12 VDC υψηλής ροπής και ελεγκτές θέσης. Εμπρόσθιοι και πίσω τροχοί διπλού ελαστικού για την σταθερότητα της πλατφόρμας με ανθεκτικά ελαστικά από καουτσούκ τα οποία μπορούν να κυλίσουν εύκολα σε σκληρά δάπεδα και χαλιά.

Πέντε αισθητήρες απόστασης (τρεις υπέρυθρες και δύο υπερήχων) για αποφυγή σύγκρουσης.

Ο πλήρως ενσωματωμένος πίνακας ελέγχου ο οποίος χειρίζεται όλες τις λειτουργίες των υλικών του χαμηλότερου επιπέδου για την οδήγηση κινητήρων και τη συλλογή δεδομένων αισθητήρων.

Απλή σύνδεση USB μεταξύ του πίνακα ελέγχου και του φορητού σας υπολογιστή. 12 V, 14,4 Ah μπαταρίες κλειστού τύπου για 4-7 ώρες συνεχούς λειτουργίας.

***Κόστος:1300 Ευρώ***

## Ρομποτικοί Βραχίονες

Μια άλλη κατηγορία εκπαιδευτικών ρομπότ είναι αυτά των ρομποτικών βραχιόνων. Και εδώ υπάρχουν αρκετές πλατφόρμες με αρκετές δυνατότητες. Οι διαφοροποιήσεις τους είναι κυρίως στους βαθμούς ελευθερίας του καθενός και στα περιβάλλοντα διεπαφής του χρήστη με τον βραχίονα.

### Magician Dobot



Εικόνα 26 Magician Dobot

Η εκπαιδευτική εκδοχή του Magician Dobot είναι ένα ρομποτικός βραχίονας σχεδιασμένος αποκλειστικά για ιδρύματα τριτοβάθμιας εκπαίδευσης, συμπεριλαμβανομένων των σχολών μηχανικής και των πανεπιστημιακών ινστιτούτων τεχνολογίας - ή ακόμα και μεταπτυχιακούς φοιτητές που σπουδάζουν επιστήμες σχετικές με τη ρομποτική τεχνολογία. Αυτός ο ρομποτικός βραχίονας έχει 4 βαθμούς ελευθερίας και είναι ικανός για πολλές εφαρμογές. Αλλάζοντας τα εργαλεία που μπορούν να προστεθούν στο άκρο του μπορεί να μετατραπεί το Dobot Magician σε έναν εκτυπωτή 3D, ένα εργαλείο βιομηχανικού σχεδιασμού και γραφής, ένα σφυρί Μπορεί να συγκρατήσει πράγματα μέχρι 0.5 κιλά και επίσης μπορεί με μια μικρή βεντούζα για να πάρει μικρά αντικείμενα. Ο ρομποτικός βραχίονας αναπτύχθηκε για

επιστημονικό σκοπό παραδίδεται έτοιμος για εγκατάσταση. Προσφέρει μια πληθώρα δυνατοτήτων συμπεριλαμβανομένης μιας μονάδας WiFi, μιας μονάδας Bluetooth και λειτουργία εκτός σύνδεσης. Επιπλέον, έχει 10 διαμορφωμένες εισόδους / εξόδους, 4 ελεγχόμενες εξόδους 12V και πολλές διεπαφές επικοινωνίας. Ο ρομποτικός βραχίονας έρχεται με ένα ιδανικό περιβάλλον προγραμματισμού για τους φοιτητές. Έχει ειδικό λογισμικό - το DobotStudio - το οποίο περιλαμβάνει μια σειρά από παραμέτρους ελέγχου, και υπάρχει και ένας επεξεργαστής εικονικού προγραμματισμού ρομποτικού βραχίονα για τους νέους προγραμματιστές.

Αυτός ο ρομποτικός βραχίονας έρχεται με το Repetier-Host, μια πλήρη βιβλιοθήκη κωδικών, ένα εργαλείο τύπου Scratch (Blockly) για την ανάπτυξη του ρομποτικού σας βραχίονα. Το λογισμικό και οι εφαρμογές που κυκλοφορούν με αυτό το ρομποτικό βραχίονα για εκπαίδευση εκπληρώνουν το μεγαλύτερο μέρος της εργασίας που χρειάζεται να κάνουμε επιτρέποντάς μας να τον ελέγξουμε με ακρίβεια χιλιοστού από τον υπολογιστή μας (Mac ή Windows), tablet ή smartphone. Μπορούμε να αναπτύξουμε ακόμη και εντολές φωνής και χειρονομίας, συνδέοντάς το με τη συσκευή Leap Motion, για παράδειγμα.

- Τεχνικές προδιαγραφές του βραχίονα
- 4 άξονες
- Μέγιστο ωφέλιμο φορτίο: 500g
- Μέγιστη εμβέλεια: 320 mm
- Επαναληψιμότητα θέσης (έλεγχος): 0,2 mm
- Επικοινωνία: USB / WiFi / Bluetooth
- Τροφοδοσία: 100-240V, 50/60 Hz
- Ισχύς: 12V / 7A DC
- Κατανάλωση: Μέγιστο 60W.
- Θερμοκρασία λειτουργίας: -10 έως 60 ° C
- Καθαρό βάρος: 3,4 kg
- Μεικτό βάρος: 8 kg
- Διαστάσεις βάσης: 158 x 158 mm
- Υλικά: αλουμίνιο (κράμα 6061), ABS μηχανικό πλαστικό
- Ενσωματωμένο ελεγκτή Dobot
- Διαστάσεις: 307 x 224 x 330 mm

- Επεκτάσιμες διεπαφές I / O: 10 I / O (μπορούν να διαμορφωθούν ως αναλογικές είσοδοι ή έξοδοι PWM).
- 4 ελεγχόμενες έξοδοι ισχύος 12V.
- διεπαφές επικοινωνίας (UART, Reset, Stop, 12V, 5V και 2 εισόδου / εξόδου).
- Διαθέσιμα αξεσουάρ (έκδοση εκπαίδευσης):
- Συσκευή αναρρόφησης κενού.
- Gripper Αρπάγη.
- Ακροφύσιο 3D kit εκτυπωτή.
- Σετ γραφής και σχεδίασης το οποίο δέχεται στυλό/μολύβι/μαρκαδόρο.
- Ακίδα λέιζερ.
- Μονάδα Bluetooth.
- Μονάδα WiFi.
- Χειριστήριο.

**Κόστος:1240.20 Ευρώ**

## **Gauss6**



Εικόνα 27 Gauss6

Το Gauss6 είναι μια ακόμη πλατφόρμα ρομποτικού βραχίονα με 6 άξονες ειδικά σχεδιασμένη για τους χρήστες της κοινότητας ROS και της ρομποτικής. Το Gauss6 είναι ένα τυπικό εργαλείο μελέτης που βοηθά τους φοιτητές να γνωρίζουν τη δομή ενός χειρισμού και του εγγενούς προγραμματισμού ROS. Μπορούμε να μάθουμε την τυπική δομή, τον κινηματικό και τον ρομποτικό έλεγχο ενός βραχίονα 6 αξόνων από το Gauss6. Χ ROS, συμπεριλαμβανομένου του σχεδιασμού διαδρομής χρησιμοποιεί πολλές εφαρμογές του ROS για την κινηματική του και υπάρχει και σε προσομοίωση στο GAZEBO. Επιπλέον, μπορούμε να δημιουργήσουμε εφαρμογές που απαντούν στις δικές μας ανάγκες, προσθέτοντας στο άκρο του ειδικά εργαλεία. Το Gauss6 λόγω της οικονομικής του τιμής μπορεί να υπάρχει σε κάθε εργαστήριο ρομποτικής σε πολλά κομμάτια έτσι ώστε να δίνει την δυνατότητα στους φοιτητές να πειραματίζονται μαζί του ατομικά ή με μικρές ομάδες.

#### **Χαρακτηριστικά:**

Ανοιχτό λογισμικό και τα εξαρτήματα που τον αποτελούν είναι 3D εκτυπώσιμα. Η τιμή του Gauss6 αποτελεί σημαντικό προσόν για εξοπλισμό ενός εργαστηρίου με πολλές μονάδες.

Η κοινότητα του Gauss6 συνδέει τους χρήστες του να μοιραστούν τις ιδέες, τα παραδείγματα, τις εμπειρίες τους, τους πηγαίους κώδικες, τη γνώση, την τεχνογνωσία, και τη δημιουργικότητα τους.

Το Gauss6 έρχεται με το Tony Robot Studio το οποίο είναι αρκετά εύχρηστο για τη διαχείριση του βραχίονα και το RoboWare Studio για εύκολο προγραμματισμό με την χρήση πακέτων του ROS.

***Κόστος: 700.00 Ευρώ***

#### **Αξιολόγηση και γενικά συμπεράσματα εκπαιδευτικών ρομπότ**

Ο κατάλογος των ρομποτικών βραχιόνων είναι αρκετά μεγάλος και η παρουσίαση δύο βασικών απλών βραχιόνων στην παρούσα μελέτη ήταν απλά για να συνδεθεί το ρομπότ που σχεδιάστηκε με την υπάρχουσα κατάσταση στα εκπαιδευτικά ρομπότ. Γι' αυτό δεν θα επεκταθούμε με περισσότερες παρουσιάσεις κι άλλων. Ο τομέας των

ρομποτικών βραχιόνων είναι ένα μεγάλο κομμάτι της ρομποτικής οπότε δεν μπορεί να λείπει από το θέμα της εκπαιδευτικής ρομποτικής.

Η παρουσίαση των παραπάνω ρομπότ δεν καλύπτει ολόκληρη την αγορά πράγμα που δεν θα μπορούσε να χωρέσει σε μια μελέτη μερικών σελίδων αλλά αποτελεί μία παρουσίαση των πιο διαδεδομένων πλατφόρμων που κυκλοφορούν κι είναι ευρέως αποδεκτά. Τα κύρια χαρακτηριστικά τους είναι:

Στις προσχολικές ηλικίες η ευκολία χειρισμού τους η στιβαρότητα της κατασκευής για αντοχή στις μηχανικές καταπονήσεις και η μικρή πολυπλοκότητα των κινήσεων τους.

Για τις ηλικίες του δημοτικού η πολυπλοκότητα ανεβαίνει . Καινούργια δυνατότητα εδώ είναι η διασύνδεση των ρομπότ με υπολογιστή, ταμπλέτα ή κινητό τηλέφωνο. Ο προγραμματισμός γίνεται κυρίως με οπτικό προγραμματισμό τύπου Scratch για εξοικείωση των μαθητών με έννοιες του προγραμματισμού. Τα ρομπότ εδώ, μπορούν να επιτελέσουν πιο πολύπλοκες αποστολές και χρησιμοποιούν αισθητήρες απόστασης ήχου, επαφής ακόμη και φωτός. Τα ρομπότ όμως, είναι έτσι φτιαγμένα ώστε τα παιδιά να μη μπορούν να δουν πως λειτουργεί η όλη κατασκευή. Αυτό είναι λογικό αφού έννοιες όπως το ρεύμα και η τάση είναι άγνωστες για αυτή την ηλικία.

Σαν τρίτη κατηγορία αυτή της δευτεροβάθμιας εκπαίδευσης η κύρια διαφορά με την προηγούμενη είναι η δυνατότητα επέμβασης στο ρομπότ. Εδώ τώρα βασικό χαρακτηριστικό είναι ότι ο χρήστης μπορεί να μετατρέψει αισθητά τον χαρακτήρα αλλά και τις εφαρμογές του ρομπότ. Επίσης υπάρχει η δυνατότητα να προστεθούν κι άλλοι αισθητήρες κινητήρες ακόμη και κάμερες στα ρομπότ βάσης. Ο προγραμματισμός κι εδώ είναι οπτικός, με scripts ή και με τους δύο τρόπους. Υπάρχει μεγάλη γκάμα πλατφόρμων με πληθώρα αισθητήρων αλλά και κινητήρων. Αυτό βεβαίως έχει αντανάκλαση και στις τιμές των ρομπότ.

Στην κατηγορία των εκπαιδευτικών ρομπότ για την τριτοβάθμια εκπαίδευση τα κιτ και πάλι είναι πολυάριθμα. Εδώ το κεντρικό ενδιαφέρον εστιάζεται στην ανάπτυξη δομών ελέγχου των κινήσεων του ρομπότ ,στην ανάπτυξη αλγορίθμων χαρτογράφησης, αντίληψης της θέσης, του αυτόνομης κίνησης σε άγνωστα περιβάλλοντα, ανάπτυξη αλγορίθμων αναγνώρισης εικόνας και άλλα .Ο προγραμματισμός σ αυτή την κατηγορία γίνεται μόνο με scripts σε διάφορες

γλώσσες αλλά κυριαρχούν η C++ και η Python. Ένα ενδιαφέρον framework που παρουσιάζεται σ αυτή την κατηγορία είναι το ROS το οποίο είναι ένα ανοιχτό περιβάλλον προγραμματισμού ρομπότ με τεράστια κοινότητα από developers. Το σημαντικό με το συγκεκριμένο πακέτο είναι η ευρεία αποδοχή του από την παγκόσμια κοινότητα και η μεγάλη γκάμα ετοιμών εργαλείων για προσομοίωση έλεγχο και αξιολόγηση των εργασιών ενός ρομπότ.

Τελευταία κατηγορία αυτή των ρομποτικών βραχιόνων έχει ενδιαφέρον γιατί σχετίζεται με τα πιο διαδεδομένα ρομπότ αυτή τη στιγμή στη βιομηχανία. Το θέμα που διδάσκεται εδώ είναι η εύρεση της τροχιάς που πρέπει να κάνει ο βραχίονας έτσι ώστε να επιτελέσει μια συγκεκριμένη αποστολή. Υπάρχουν προγράμματα τα οποία κάνουν αυτούς τους υπολογισμούς αλλά από εκπαιδευτικής άποψης καλά θα ήταν οι μαθητές να παίρνουν και μία ιδέα από ευθύ και ανάστροφο κινηματικό πρόβλημα.

## Κεφάλαιο 3<sup>ο</sup>

### Κεντρικός στόχος της εκπαιδευτικής πλατφόρμας

Όπως είδαμε στο προηγούμενο κεφάλαιο η κατηγορία των εκπαιδευτικών ρομπότ είναι πολυάριθμη, και με πολλές προσεγγίσεις ανάλογα με την ηλικία στην οποία απευθύνονται. Η παρούσα εργασία εστιάζει το ενδιαφέρον της σε ένα εκπαιδευτικό ρομπότ για την δευτεροβάθμια εκπαίδευση, με δυνατότητες οι οποίες μπορούν να προσεγγίσουν ρομπότ ακόμη και σε προπτυχιακό επίπεδο.

Το πρώτο σημείο που λήφθηκε υπόψη ήταν το ρομπότ που θα φτιαχτεί να μην χρησιμοποιείται ως εργαλείο για προσεγγίσεις άλλων μαθημάτων με τη STEM μέθοδο, αλλά και ως πλατφόρμας εκπαίδευσης των μαθητών στην ρομποτική, όπου μέσα από αυτή να διδάσκονται και έννοιες τεχνολογίας, φυσικής, μαθηματικών, και μηχανικής έτσι ώστε να οδηγεί τελικά σε STEM[19,20]. Γι' αυτό θα πρέπει να μπορεί να προσεγγίζει πολλούς τομείς της ρομποτικής χωρίς να τους κρύβει από τους μαθητές. Για να επιτευχθεί κάτι τέτοιο το ρομπότ πρέπει να κατασκευάζεται από τους μαθητές και να μην είναι προκατασκευασμένο. Κατά την διάρκεια της συναρμολόγησης στους μαθητές εξηγείται η αναγκαιότητα του κάθε καλωδίου και της κάθε σύνδεσης. Σε πολλές πλατφόρμες όπου τα ρομπότ είναι προκατασκευασμένα αυτό δεν γίνεται σαφές, και οι μαθητές μένουν με την εντύπωση ότι τις εντολές το ρομπότ τις καταλαβαίνει από το μόνο του. Ακόμη και σε πλατφόρμες με μη προκατασκευασμένα ρομπότ τα καλώδια δεν είναι ξεχωριστά. Πχ τα καλώδια που οδηγούν τους κινητήρες στο LEGO είναι τετραπλά αλλά αυτό ο μαθητής δεν το καταλαβαίνει γιατί απλά δεν του χρειάζεται για να λειτουργήσει η συγκεκριμένη πλατφόρμα. Αντί αυτού εάν ο μαθητής διδαχτεί ότι αυτός ο κινητήρας για να λειτουργήσει, χρειάζεται δύο καλώδια, και εάν θέλουμε και δεδομένα από αυτόν, π.χ. για το πόσο περιστράφηκε, ακόμη περισσότερα, η γνώση του είναι σφαιρική κι όχι επιφανειακή. Επίσης πολύ ίσως οι περισσότεροι κινητήρες που χρησιμοποιούνται στη ρομποτική είναι βηματικοί. Κι εδώ η προσέγγιση από τις περισσότερες πλατφόρμες είναι να αποκρύπτουν τον τρόπο λειτουργίας τους. Ο μαθητής νομίζει πως μαγικά ο βραχίονας θα πάει στη θέση που του περιγράψαμε στο κώδικά μας πχ στις 90 μοίρες. Το ίδιο συμβαίνει και με τους αισθητήρες κανένας αισθητήρας δεν δουλεύει με ένα και μόνο καλώδιο. Ακόμη και ο πιο απλοί αυτοί της επαφής χρειάζονται δύο. Σημαντικό κομμάτι λοιπόν της εκπαίδευσης των παιδιών



στη ρομποτική είναι να αποκαλύπτονται λεπτομέρειες της κατασκευής του ρομπότ χωρίς να προσπερνιούνται[19-22]. Αυτό που προσφέρει η ρομποτική είναι η σύνδεση του προγραμματισμού και της αλγοριθμικής σκέψης με τον πραγματικό κόσμο. Εάν πολλά κομμάτια αυτής της σύνδεσης τα κρύβουμε, τότε δεν χρειάζεται να χρησιμοποιούμε πραγματικά ρομπότ αλλά μόνο προσομοιώσεις. Με αυτόν τον τρόπο όμως χάνουμε την έννοια του STEM τουλάχιστον σε αυτούς της φυσικής της τεχνολογίας και της μηχανικής. Εάν παίζουμε ένα απλό παιχνίδι με έννοιες φυσικής πχ AGRY BIRDS τότε θα παρατηρήσουμε ότι εάν είμαστε προσεκτικοί και ξεκινήσουμε από την ίδια θέση την εκτόξευση και με την ίδια δύναμη τότε το αποτέλεσμα θα είναι ακριβώς το ίδιο. Εάν αυτό γινόταν στην πραγματικότητα το αποτέλεσμα σχεδόν πάντα θα ήταν διαφορετικό. Για να πετυχαίναμε το ίδιο αποτέλεσμα έπρεπε οι αρχικές μας συνθήκες να ήταν ακριβώς ίδιες κάθε φορά και για να το επιτύχουνε αυτό οι μαθητές θα μάθαιναν την έννοια των αρχικών συνθηκών και πόσο σημαντικές είναι. Θα έψαχναν να βρουν τα αίτια που διαφοροποιούν τις αρχικές συνθήκες και με αυτόν τον τρόπο θα μάθαιναν πολύ καλύτερα τι σημαίνει τριβή τι κέντρο βάρους ,αδράνεια επιτάχυνση κτλ. Έτσι καταλαβαίνουν την έννοια της αιτίας και του αποτελέσματος. Σε αυτήν την ηλικία όπου οι μαθητές διδάσκονται τις έννοιες της δύναμης, της ταχύτητας, της επιτάχυνσης, του ρεύματος, της τάσης, της ροπής, κτλ. η ρομποτική είναι το πιο χρήσιμο εργαλείο για να δείξουμε τις εφαρμογές τους στον πραγματικό κόσμο[23,24].

### **Επεκτασιμότητα εκπαιδευτικής πλατφόρμας**

Δεύτερο σημείο που δόθηκε έμφαση στο σχεδιασμό και την επιλογή του υλικού ήταν η επεκτασιμότητα του ρομπότ. Επειδή αναφέρεται σε εκπαιδευτική κοινότητα και σε μαθητές θα πρέπει να υπάρχει η δυνατότητα οι μαθητές αφού κατανοήσουν τα βασικά της ρομποτικής να μπορούν κι οι ίδιοι να προτείνουν προσθήκες, βελτιώσεις και εφαρμογές πέρα από αυτά που διδάσκονται[5,12,14-16]. Αυτό σημαίνει ότι το ρομπότ που θα φτιαχτεί, θα πρέπει να μπορεί να δεχτεί τροποποιήσεις και προσθήκες είτε σε υλικό π.χ. αισθητήρες είτε σε λογισμικό. Η λογική του ρομπότ είναι το λογισμικό του. Συνεπώς αφού θέλαμε μια εισαγωγή στην πραγματική ρομποτική θέλαμε κι ένα λειτουργικό σύστημα που να μπορεί να είναι εύχρηστο και αρκετά ευέλικτο για πολλαπλές εφαρμογές. Εδώ το Framework του ROS[26] ήταν μια από τις πιο κατάλληλες επιλογές.

Το ROS είναι ένα μετα-λειτουργικό σύστημα ανοιχτού κώδικα για τα ρομπότ. Αν και τα αρχικά ROS σημαίνουν Robot Operating System, στην πραγματικότητα είναι ένα πλαίσιο που τρέχει σε υπάρχον λειτουργικό σύστημα, όπως το GNU/Linux. Παρέχει τις υπηρεσίες που παρέχει ένα λειτουργικό σύστημα, συμπεριλαμβανομένης της λειτουργίας του υλικού, του ελέγχου των συσκευών χαμηλού επιπέδου, της εφαρμογής κοινώς χρησιμοποιούμενων λειτουργιών, της μετάδοσης μηνυμάτων μεταξύ των διαδικασιών και της διαχείρισης των πακέτων λογισμικού και κώδικα. Παρέχει επίσης εργαλεία και βιβλιοθήκες για την απόκτηση, κατασκευή, γραφή και εκτέλεση κώδικα σε πολλούς υπολογιστές. Το ROS είναι παρόμοιο σε ορισμένα σημεία με τα "framework ρομπότ", όπως το Player, το YARP, το Orocos, το CARMEN, το Orca, το MOOS και το Microsoft Robotics Studio. Το γράφημα ROS "runtime" είναι ένα δίκτυο ομότιμων δικτύων διεργασιών (δυνητικά καταναμεμένων μεταξύ των μηχανών) που είναι συζευγμένα χρησιμοποιώντας την υποδομή επικοινωνίας ROS. Το ROS υλοποιεί διάφορες διαφορετικές μορφές επικοινωνίας, όπως σύγχρονη επικοινωνία τύπου RPC μέσω υπηρεσιών(services), ασύγχρονη ροή δεδομένων μέσω θεμάτων (topics) και αποθήκευση δεδομένων σε διακομιστή παραμέτρων(parameter server). Το ROS δεν είναι ένα πλαίσιο σε πραγματικό χρόνο, αν και είναι δυνατή η ενσωμάτωση του ROS με κώδικα σε πραγματικό χρόνο. Το ROS επιτρέπει στα διάφορα κομμάτια του λογισμικού να επικοινωνούν μεταξύ ενός ή περισσότερων υπολογιστών και μικροελεγκτών, και επιτρέπουν να ελέγχουμε ένα ή περισσότερα δίκτυα ρομπότ από ένα desktop, ένα web browser, ή κάποια άλλη συσκευή. Η χρήση του ROS μας γλυτώνει από τη δουλειά της δημιουργίας πολλών πολύπλοκου κώδικα για την λειτουργία του ρομπότ, μιας και παρέχει όλο τον κώδικα που είναι απαραίτητος για την πλοήγηση και τον χειρισμό του με κάποιες μικρές μετατροπές εάν θέλουμε να τον χρησιμοποιήσουμε για το δικό μας custom ρομπότ. Όσο πιο εξελιγμένο με περισσότερες δυνατότητες θέλουμε να είναι το ρομπότ μας, τόσο περισσότερο το ROS θα είναι χρήσιμο[3]. Για παράδειγμα, με το ROS μπορούμε να μην το τηλεκατευθύνουμε με το joystick, αλλά να προσπαθήσει να βρει το δρόμο, μόνο του. Η διαφορά στην πολυπλοκότητα σε σύγκριση με τον χειροκίνητο έλεγχο, είναι ότι μπορεί το ρομπότ να δημιουργήσει και να χρησιμοποιήσει χάρτες για να αποφύγει εμπόδια[1,2]. Για παράδειγμα, ο έλεγχος ενός ρομπότ μέσω joystick μπορεί να φτιαχτεί αρκετά γρήγορα, χρησιμοποιώντας ένα Arduino[6]. Για την αυτόνομη κίνηση όμως, το ROS έχει διαθέσιμες λειτουργίες, όπως δημιουργία χαρτών, χειρισμό βάθους χάρτη και εντοπισμού εμποδίων, για να

μπορούμε να δώσουμε στο ρομπότ μας αυτονομία[7]. Μια άλλη δυνατότητα που μας έδινε το ROS ήταν αυτή της οπτικοποίησης του ρομπότ. Το ROS συνεργάζεται με το Rviz[39] (ROS οπτικοποίηση) το οποίο είναι ένας τρισδιάστατος οπτικοποιητής για την εμφάνιση δεδομένων αισθητήρων και πληροφοριών κατάστασης από το ROS. Χρησιμοποιώντας το rviz, μπορούμε να απεικονίσουμε την τρέχουσα κατάσταση του ρομπότ σε ένα εικονικό μοντέλο. Το οποίο έχει περιγραφεί προηγουμένως σε ένα αρχείο XML. Μπορούμε επίσης να παρατηρούμε ζωντανές αποτυπώσεις τιμών αισθητήρων που προέρχονται από topics του ROS, όπως δεδομένα κάμερας, μετρήσεις απόστασης υπέρυθρων, δεδομένα σόναρ και πολλά άλλα. Ένα δεύτερο εργαλείο που μας δίνει το ROS είναι το Gazebo[29]. Από την στιγμή που έχουμε περιγράψει το ρομπότ μπορούμε να το παρατηρήσουμε μέσα από το Rviz και να βλέπουμε τα δεδομένα των αισθητηρίων του. Αυτό όμως δεν είναι πραγματική προσομοίωση αλλά οπτικοποίηση. Περιγράφοντας ακόμη περισσότερους παραμέτρους του ρομπότ όπως βάρος, ροπές αδράνειας, τριβές, ενώσεις μεταξύ των κινούμενων μερών του μπορούμε να έχουμε και την προσομοίωση του σε εικονικό περιβάλλον με όρους φυσικής δηλαδή σαν να κινείτε το ρομπότ μας στον πραγματικό κόσμο. Το Gazebo είναι ο ένας προσομοιωτής φυσικής πραγματικού κόσμου όπου μπορούμε να δημιουργήσουμε έναν περιβάλλον να τοποθετήσουμε το ρομπότ μας και το Gazebo προσομοιώνει το ρομπότ μας να κινείται μέσα σ αυτό. Το εκπαιδευτικό όφελος εδώ είναι μεγάλο μιας και δύσκολες έννοιες όπως ροπή και ροπή αδράνειας είναι αναγκαίες για την σωστή κίνηση του εικονικού ρομπότ μέσα στο εικονικό περιβάλλον.

Όπως γίνεται αντιληπτό από τη στιγμή που αποφασίστηκε να χρησιμοποιηθεί το ROS σαν λειτουργικό σύστημα του ρομπότ αυτό σήμαινε ότι η καρδιά του ρομπότ θα έπρεπε να είναι ένας υπολογιστής κι όχι ένα microcontroller όπως το Arduino. Θα μπορούσε βέβαια να είναι ένας απλός microcontroller κι απλός να επικοινωνούσε με ένα υπολογιστή μέσω WIFI ή Bluetooth, αλλά αυτό θα μείωνε τις δυνατότητες επέκτασης του ρομπότ πχ δεν θα μπορούσε να δεχτεί κάμερα. Κι εδώ αποφασίστηκε να χρησιμοποιηθεί ένας single board pc το ευρέως διαδεδομένο Raspberry Pi 3+[4]. Το Raspberry Pi 3 διαθέτει Quad Core Processor 1.2GHz 64-Bit, 1GB RAM, 4 θύρες USB 2.0 για σύνδεση με πληκτρολόγιο, ποντίκι και άλλα περιφερειακά, θύρα Ethernet, WiFi, Bluetooth 4.1, έξοδο HDMI, έξοδο ήχου mini jack και micro-USB υποδοχή για να την τροφοδοσία του. Η επεξεργαστική

δυνατότητα του Raspberry κρίθηκε ικανοποιητική για τις εφαρμογές που θέλαμε να τρέχουμε. Η on Board WiF θύρα του ήταν αναγκαία για την ασύρματη επικοινωνία του ρομπότ με ένα υπολογιστή είτε για τον έλεγχο είτε για την μεταφορά δεδομένων.

### **Ρομποτικός Βραχίονας**

Τρίτο σημείο για την τελική μορφή του ρομπότ ήταν η επιλογή να έχει επάνω του και ένα απλό ρομποτικό βραχίονα. Όπως αναφέρθηκε και στο δεύτερο κεφάλαιο οι ρομποτικοί βραχίονες είναι σημαντικό κομμάτι της ρομποτικής βιομηχανίας σήμερα[8]. Οι μαθητές έχοντας στα χέρια τους ένα τέτοιο εργαλείο μπορούν να μελετήσουν απλές κινηματικές ασκήσεις[41]. Επίσης μπορούν να καταλάβουν τι σημαίνει ευθύ και ανάστροφο κινηματικό πρόβλημα. Ο βραχίονας επιλέχθηκε να είναι όσο πιο απλός γίνεται για να μπορούν οι μαθητές να αντιμετωπίσουν ένα απλό ευθύ κινηματικό πρόβλημα. Στο ρομπότ τοποθετήθηκε ένα βραχίονας τριών βαθμών ελευθερίας ο οποίος ελέγχεται από τρεις σερβοκινητήρες[32].

### **Κόστος ρομπότ**

Τέταρτο σημείο που δόθηκε μεγάλη βαρύτητα ήταν αυτό του κόστους και της ευκολίας εύρεσης των εξαρτημάτων που αποτελούν το ρομπότ[25]. Το ρομπότ φιλοδοξεί να είναι μια πρόταση για εισαγωγή στη ρομποτική μαθητών οπότε η αστοχίες και πολλές φορές οι απροσεξίες θα είναι συχνές, συνεπώς ότι μπορεί να χαλάσει θα πρέπει εύκολα να μπορεί να αντικατασταθεί και να μην είναι απαγορευτική η τιμή των ανταλλακτικών του. Επίσης το λογισμικό είναι καλό να είναι ελεύθερο και να μην απαιτεί άδειες για την χρησιμοποίησή του. Το κόστος επιλέχθηκε να είναι χαμηλό ώστε να μην αποτελεί πρόβλημα στην απόκτηση πολλών μονάδων έτσι ώστε να δίνεται η δυνατότητα σε μικρές ομάδες μαθητών να έχει η κάθε μια από ένα και να μην μοιράζεται όλο το τμήμα ένα ρομπότ.

<b>Εξάρτημα</b>	<b>Κόστος σε Ευρώ</b>
Raspberry 3 B+ board x 1	41.90
motor controller shield x 1	5.20
Raspberry Pi Camera x 1	21.20
Servo motors x 3	3x2.5=7.50
Led photodiode encoders x 2	2x3.90=7.80
Ultrasonic Sensor x 1	2.50
DC motors with Encoder disks x 2	2x1.8=3.60
Plexiglas base x 1	1.40
Caster Ball x 1	1.60
Rechargeable Battery x 1	9.90
4xAA Battery Holder x 1	0.80
Wheel x 2	2x1.5=3.00
Cables and Lego Parts	4.00
<b>Συνολικό Κόστος</b>	<b>110.40</b>

Πίνακας 1 κόστος εξαρτημάτων ρομπότ

### **Επιπλέον αισθητήρες ρομπότ**

Τέλος αποφασίστηκε να τοποθετηθεί ένας αισθητήρας υπερήχων [34] και μια κάμερα με σκοπό την αποφυγή εμποδίων από το ρομπότ και τον απομακρυσμένο έλεγχο του ρομπότ. Σαν περαιτέρω βελτίωση του ρομπότ μπορούν να εφαρμοστούν αλγόριθμοι ανάλυσης εικόνας για αυτόνομη οδήγηση και πολλά άλλα. Οι κινητήρες είναι απλοί DC για να κρατηθεί το κόστος χαμηλά, αλλά για τον έλεγχο της κίνησης επιλέχτηκαν και δύο αισθητήρες οπτοδιακοπής [33] που σε συνδυασμό με δύο δίσκους μ' εγκοπές που εφαρμόζουν στους άξονες των τροχών μπορούμε να έχουμε πληροφορία για την περιστροφή των τροχών του ρομπότ και από αυτή να εξάγουμε πληροφορία για την θέση και την ταχύτητά του. Για να εξαχθούν αυτές οι πληροφορίες για την κίνηση του ρομπότ οι μαθητές θα χρησιμοποιήσουν γνώσεις μαθηματικών, γεωμετρίας και φυσικής.

## Κεφάλαιο 4<sup>ο</sup>

### Κατασκευή και προγραμματισμός του Ρομπότ.

#### ROS

Όπως έγινε κατανοητό από το προηγούμενο κεφάλαιο το βασικό χαρακτηριστικό της όλης υλοποίησης είναι το ROS συνεπώς αρχική δουλειά μας ήταν η εγκατάσταση του στο Raspberry.

Το ROS τρέχει σε περιβάλλον LINUX συνεπώς το λειτουργικό σύστημα που θα έτρεχε στο Raspberry θα έπρεπε να είναι τέτοιου τύπου. Το ROS μπορεί να εγκατασταθεί και σε λειτουργικό Raspbian, το επίσημο λειτουργικό του Raspberry αλλά αποφεύχθηκε, γιατί σαν λειτουργικό είναι βαρύ για τον επεξεργαστή του, και οι δυνατότητες που παρέχει επιβαρύνουν τη λειτουργία του και δεν είναι αναγκαίες για την υλοποίηση του ρομπότ. Στην επίσημη σελίδα του ROS για την εγκατάσταση του στο Raspberry αναφέρει ότι είναι πιο εύκολο και πιο γρήγορο να εγκατασταθεί το Ubuntu Mate 16.04 και έπειτα το ROS από ότι να εγκατασταθεί στο Raspbian.<sup>1</sup> Ο λόγος είναι τα dependencies. Για εγκατάσταση του ROS στο Raspbian πρέπει να εγκατασταθούν αρκετά προαπαιτούμενα κομμάτια λογισμικού για την σωστή λειτουργία του. Στο site προσφέρεται και ένα image σε **SD Card** με προεγκατεστημένο το Ubuntu 16.04 (LXDE) και ROS Kinetic. Και το Ubuntu 16.04 όμως είναι κι αυτό σχετικά βαρύ κι όχι αναγκαίο για την εφαρμογή μας.

Μια αρκετά ελαφριά έκδοση του Ubuntu, και με προσανατολισμό για IoT εφαρμογές είναι το Ubuntu Core ή οποία και δοκιμάστηκε. Η σύνδεση γινόταν με Ssh απομακρυσμένα από το Raspberry και το ROS έτρεχε ικανοποιητικά. Υπήρχαν όμως δύο προβλήματα. Το Ubuntu Core λειτουργεί headless, που σημαίνει ότι δεν υποστηρίζει σύνδεση του Raspberry με οθόνη πληκτρολόγιο και ποντίκι. Επειδή το ρομπότ προορίζεται για χρήση από μαθητές οι οποίοι δεν είναι εξοικονομημένοι με χρήση από τερματικό κι έχουν συνηθίσει την χρήση παραθύρων, η λύση αυτή αν και βολική και αρκετά ελαφριά ήταν προβληματική. Ένα δεύτερο σημείο που το συγκεκριμένο λειτουργικό είχε πρόβλημα ήταν αυτό της υποστήριξης κάμερας.

---

<sup>1</sup><http://wiki.ros.org/ROSBerryPi/Installing%20ROS%20Kinetic%20on%20the%20Raspberry%20Pi>

Επειδή το Ubuntu Core λειτουργεί Headless δεν υποστήριζε την λειτουργία της κάμερας. Μετά από αυτόν τον περιορισμό η επιλογή του εγκαταλείφθηκε.

Τελικά επιλέχθηκε η εγκατάσταση του Lubuntu ενός λειτουργικού που βασίζεται στο Ubuntu 16.04 είναι σαν το Ubuntu Mate αλλά είναι πιο ελαφρύ και δεν έχει κάποιες δυνατότητες του Mate οι οποίες δεν κρίνονταν απαραίτητες. Οι πληροφορίες αντλήθηκαν από [Ubuntu Pi Flavour Maker](https://ubuntu-pi-flavour-maker.org/) το οποίο είναι ένας ιστότοπος με developers για το Ubuntu και το Raspberry. Αυτό το λειτουργικό χρησιμοποιεί κι ένα άλλο ρομπότ της εταιρίας Ubiquity Robotics η οποία κατασκευάζει ρομπότ με ROS και Raspberry. Η εταιρία μάλιστα προσφέρει το image της εγκατάστασης της στο δικό της ρομπότ χωρίς κανένα περιορισμό[27]. Το συγκεκριμένο image τελικά αφού δοκιμάστηκε, επιλέχθηκε να είναι αυτό που θα χρησιμοποιούσαμε. Η έκδοση του είναι η **Kinect**.

Ένα αρκετά χρήσιμο χαρακτηριστικό με τη συγκεκριμένη εγκατάσταση είναι ότι κάνει το Raspberry να είναι (WAP, Wireless Access Point). Στα [δίκτυα υπολογιστών](#) καλούμε **ασύρματο σημείο πρόσβασης** ή **σταθμό βάσης** (WAP, Wireless Access Point) μια συσκευή που συνδέει μεταξύ τους ασύρματες συσκευές επικοινωνίας για τον σχηματισμό ενός [ασύρματου δικτύου](#).<sup>2</sup> Με αυτόν τον τρόπο δημιουργεί το δικό του δίκτυο, πράγμα που σημαίνει ότι μπορούσαμε πολύ εύκολα να συνδεθούμε επάνω του χωρίς να χρειάζεται να έχουμε ήδη κάποιο δίκτυο και να συνδεθούν σ αυτό και το ρομπότ μας αλλά και ο απομακρυσμένος υπολογιστής μας για να κάνουμε τον έλεγχο του ρομπότ. Όπως καταλαβαίνουμε αυτή είναι πραγματικά μια πολύ χρήσιμη λειτουργία αφού μας αποδεσμεύει από την ύπαρξη ασύρματου δικτύου μέσα στην τάξη. Για να συνδεθούμε στο ασύρματο δίκτυο που δημιουργεί το Raspberry το όνομα του είναι ubiquityrobotXXXX όπου XXXX είναι το κομμάτι της MAC διεύθυνσης και ο κωδικός πρόσβασης είναι robotseverywhere

Το συγκεκριμένο image είχε κάποιες διαδικασίες οι οποίες έτρεχαν στο περιθώριο για την λειτουργία του ρομπότ της εταιρίας. Αυτές όμως η εταιρεία μας δίνει την δυνατότητα να τις απενεργοποιήσουμε εκτελώντας από την γραμμή εντολών

```
sudo systemctl disable magni-base
```

---

<sup>2</sup>[https://el.wikipedia.org/wiki/%CE%91%CF%83%CF%8D%CF%81%CE%BC%CE%B1%CF%84%CE%BF\\_%CF%83%CE%B7%CE%BC%CE%B5%CE%AF%CE%BF\\_%CF%80%CF%81%CF%8C%CF%83%CE%B2%CE%B1%CF%83%CE%B7%CF%82](https://el.wikipedia.org/wiki/%CE%91%CF%83%CF%8D%CF%81%CE%BC%CE%B1%CF%84%CE%BF_%CF%83%CE%B7%CE%BC%CE%B5%CE%AF%CE%BF_%CF%80%CF%81%CF%8C%CF%83%CE%B2%CE%B1%CF%83%CE%B7%CF%82)

Το image κατέβηκε από το site της ubiquity<sup>3</sup> και εγγράφηκε στο ssd με την αντίστοιχη εφαρμογή GNOME σε υπολογιστή ο οποίος έτρεχε LINUX. Εάν δεν έχουμε την εφαρμογή τότε από την γραμμή εντολών μπορούμε να τρέξουμε

```
sudo apt install gnome-disk-utility
```

Σε λειτουργικό σύστημα Windows μπορούμε να χρησιμοποιήσουμε το [etcher](#) για να εγγράψουμε το image στην ssd. Αφού συνδέσουμε τον υπολογιστή μας με το δίκτυο που έχει δημιουργηθεί μπορούμε να συνδέσουμε και τον υπολογιστή μας με το Raspberry πληκτρολογώντας στην γραμμή εντολών του υπολογιστή μας ssh ubuntu@10.42.0.1 και κωδικό ubuntu. Από τη στιγμή που μπορούσαμε να συνδεθούμε ασύρματα και να έχουμε πρόσβαση στο ρομπότ, αλλά είχαμε και γραφικό περιβάλλον η εργασία που είχαμε να κάνουμε, από πλευράς λογισμικού στο ρομπότ γινόταν πιο εύκολα.

Ένας άλλος λόγος για τον οποίο επιλέχτηκε το παραπάνω image ήταν ότι μπορούσε να επικοινωνεί το ρομπότ με ένα απομακρυσμένο υπολογιστή στον οποίο έτρεχε το ROS και στο ρομπότ να τρέχουν μόνο κάποια Nodes (διεργασίες) για την λειτουργία του. Έτσι οι προσομοίωση του ρομπότ έτρεχε στον απομακρυσμένο υπολογιστή και επικοινωνούσε με το ρομπότ το οποίο ελεγχόταν από αυτήν. Η φιλοσοφία του ROS, η οποία κεντρικός της πυρήνας είναι οι διεργασίες nodes, μπορούν να τρέχουν σε διαφορετικούς υπολογιστές ή και μικροελεγκτές και όλες ενώνονται μεταξύ τους επικοινωνώντας, ανταλλάσσοντας δεδομένα και πληροφορίες μέσω του RosMaster. Η ανάπτυξη του δικτύου από το Raspberry ευνοούσε πάρα πολλή τη σύνδεση με το ρομπότ με ένα υπολογιστή ή και δεύτερου ρομπότ με τον υπολογιστή ή τη σύνδεση των δύο ρομπότ ή και περισσότερων μεταξύ τους.

### **Σασί και κίνηση του ρομπότ.**

Από τη στιγμή που το ρομπότ μας αναφέρετε σε παιδιά για την εισαγωγή τους στη ρομποτική θα έπρεπε ο τρόπος κίνησής του ρομπότ να είναι εύκολος στην κατανόηση. Να έχει εκπαιδευτικό χαρακτήρα τον οποίον οι μαθητές τον μαθηματικό φορμαλισμό που θα χρησιμοποιήσουμε να είναι του επιπέδου τους να μπορεί να εφαρμοστεί και να συν επικουρεί της γνώσεις τους. Ο πιο διαδεδομένος τρόπος κίνησης ενός ρομπότ βασίζεται σ αυτόν της διαφορικής κίνησης[βλέπε Παράρτημα].

---

<sup>3</sup><https://downloads.ubiquityrobotics.com/pi.html>



Στο ρομπότ υπάρχουν δύο τροχοί οι οποίοι μπορούν να κινούνται ανεξάρτητα ο ένας από τον άλλον. Ένας κινητήρας είναι συνδεδεμένος με τον κάθε τροχό. Η κατεύθυνση του ρομπότ αλλάζει δίνοντας διαφορετικές ταχύτητες περιστροφής στον κάθε τροχό. Εάν κι οι δύο τροχοί κινούνται με την ίδια φορά και την ίδια ταχύτητα το ρομπότ πηγαίνει ευθεία. Εάν ο αριστερός τροχός είναι ακίνητος και περιστρέφεται ο δεξιός τότε το ρομπότ στρέφεται με κέντρο τον αριστερό τροχό και φορά ανάλογα την φορά περιστροφής του δεξιού τροχού. Προς τα αριστερά εάν ο δεξιός στρέφεται αντίθετα από τους δείκτες του ρολογιού και προς τα δεξιά εάν στρέφεται όπως οι δείκτες του ρολογιού. Το αντίθετο συμβαίνει εάν είναι ακίνητος ο δεξιός τροχός και περιστρέφεται ο αριστερός. Το ρομπότ για να κρατά την ισορροπία του στηρίζεται και σ ένα τρίτο τροχό όπου μπορεί να είναι :

Γ) ρόδα τύπου castor



shutterstock.com • 743632066

Εικόνα 28 ρόδα τύπου castor

- Ο οποίος έχει τρεις βαθμούς ελευθερίας
- Περιστροφή γύρω από τον άξονα της ρόδας
- Ελεύθερη περιστροφή γύρω από το σημείο επαφής με το έδαφος
- Περιστροφή γύρω από τον άξονα στήριξης

## II) Σφαιρικός τροχός



Εικόνα 29 σφαιρικός τροχός

- Ο οποίος έχει τρεις βαθμούς ελευθερίας
- Περιστροφή γύρω από το κέντρο της σφαίρας σε οποιαδήποτε κατεύθυνση.

Στο ρομπότ τελικά επιλέχθηκε ο σφαιρικός τροχός λόγω απλότητας ,λιγότερης αδράνειας και καλύτερης κατευθυντικότητας. Εάν ο τροχός είναι στραμμένος, με το που ξεκινά το ρομπότ την κίνηση αρχικά στρεβλώνει την τροχιά του μέχρι να ευθυγραμμιστεί με την κίνηση. Αυτό το φαινόμενο δεν είναι εμφανές σε μεγάλα και βαριά ρομπότ αλλά σε μικρά όπως το δικό μας η τροχιά αλλάζει αισθητά.



Εικόνα 30 Σασί, κινητήρες μπαταριοθήκη

Το σασί που επιλέχθηκε να χρησιμοποιηθεί ήταν ένα απλό πλεξιγκλάς με τρύπες έτοιμες για τους κινητήρες οι οποίοι περιέχονταν στο κιτ. Οι κινητήρες είναι απλοί

κινητήρες 5V, οι οποίοι περιστρέφονται με συχνότητα 200στροφές/min Το συγκεκριμένο kit επιλέχτηκε για την ευκολία εύρεσης του και την οικονομική του τιμή. Ο castor τύπου τροχός που περιείχε το kit δεν χρησιμοποιήθηκε αλλά όπως προαναφέραμε χρησιμοποιήθηκε ένας τύπου σφαίρας.

### **Κεντρική μονάδα ελέγχου Raspberry Pi 3+ Model B**



Εικόνα 31RaspberriPi 3 ModelB

ΤοRaspberriPi 3 ModelB είναι ένας υπολογιστής σε υλοποίηση μιας πλακέτας (singleboardcomputer) .Διαθέτει ένα QuadCoreProcessor 1.2GHz 64-Bit, 1GBRAM, 4 θύρες USB 2.0 για σύνδεση με πληκτρολόγιο, ποντίκι και άλλα περιφερειακά, θύρα Ethernet, WiFi, Bluetooth 4.1, έξοδο HDMI, έξοδο ήχου minijack έξοδο για TFT οθόνη ,είσοδο για κάμερα και microUSB υποδοχή για να την τροφοδοσία του.

Για να λειτουργήσει το Pi 3 χρειάζεται τροφοδοσία 5V 2.5A και κάρτα microSD όπου θα εγκατασταθεί το λειτουργικό σύστημα.

### **Ρομποτικός βραχίονας**

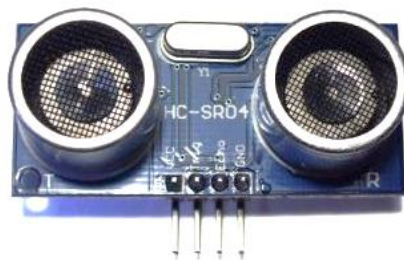
Ο ρομποτικός βραχίονας αποτελείται από τρεις σερβοκινητήρες SG 90

- τάση λειτουργίας 5 volt
- γωνία περιστροφής 0°-180° μοίρες,
- ροπή 2.5kg/cm,

- γωνιακή ταχύτητα 60°/0.1s

οι οποίοι είναι και οι αρθρώσεις του. Σαν δομικά του μέρη χρησιμοποιούνται κομμάτια LEGO [38] τα οποία αποτελούν την βάση του και τους βραχίονες του. Η παραπάνω κατασκευή είχε και πάλι σαν σκεπτικό την ευκολία εύρεσης των υλικών και την οικονομική τους τιμή.

### Αισθητήρες/κάμερα



Εικόνα 32 αισθητήρας υπερήχων απόστασης ο SR04

Οι αισθητήρες που χρησιμοποιήθηκαν στο ρομπότ είναι ένας αισθητήρας υπερήχων απόστασης ο SR04[34] ο οποίος μπορεί να υπολογίσει αποστάσεις από 2εκ. έως 400εκ. με ακρίβεια ενός εκατοστού. Ο συγκεκριμένος αισθητήρας μπορεί να χρησιμοποιηθεί για εφαρμογή αποφυγής εμποδίων η οποία είναι μία βασική λειτουργία του κάθε αυτόνομου ρομπότ και κρίνεται απαραίτητη.



Εικόνα 33 Οπτικός Αισθητήρας Στροφών Waveshare 12225

Δύο Οπτικούς Αισθητήρες Στροφών Waveshare 12225 (εικόνα 35)[33]. Οι αισθητήρες αυτοί έχουν ένα led υπέρυθρης κι ένα δέκτη υπέρυθρης τα οποία βρίσκονται σε οπτική επαφή. Όταν ένα σώμα παρεμβληθεί ανάμεσά τους αυτή η επαφή διακόπτεται και το κύκλωμα του αισθητήρα αλλάζει κατάσταση. Αυτή την εναλλαγή την πετυχαίνουμε, με την τοποθέτηση δύο δίσκων με διάκενα (εικόνα36), στους άξονες των τροχών.



Εικόνα 34 δίσκος με διάκενα

Όταν ανάμεσα στον πομπό και το δέκτη παρεμβάλλεται διάκενο ο αισθητήρας είναι ON ενώ όταν διακόπτεται η οπτική επαφή OFF. Οι συγκεκριμένοι αισθητήρες μπήκαν για να μπορέσουν οι μαθητές να κατανοήσουν πως μπορεί αν γίνει η κωδικοποίηση της κίνησης του ρομπότ. Η επιλογή τους έγινε με βάση την απλότητα κατανόησης της λειτουργίας τους η οποία είναι πιο βασική από ότι ενός αισθητήρα Hall. Μετρώντας τον αριθμό των διάκενων οι μαθητές μπορούν να υπολογίσουν το πόσο περιστράφηκε η ρόδα και από εκεί να υπολογίσουν την μετατόπιση και την ταχύτητα του ρομπότ.

### **Κάμερα**



Εικόνα 35 κάμερα

Η κάμερα (εικόνα37)[31] που χρησιμοποιείται είναι η επίσημη κάμερα για το Raspberry

Pi Camera Module Adjustable-Focus (5MP,1080p)

**Κατασκευαστής :Waveshare χώρα προέλευσης Κίνα**

5 megapixelOV5647 sensor

### **Χαρακτηριστικά Κάμερας**

- CCD size : 1/4inch
- Aperture (οπή)(F) : 1.8
- Εστιακό μήκος : 3.6MM (προσαρμοζόμενο)
- Διαγώνιος : 75.7 degree
- Μεγίστη ανάλυση : 1080p.

Η κάμερα ένα ρόλο που εξυπηρετεί κατ' αρχάς είναι ο εντυπωσιασμός των μαθητών, αλλά χωρίς να είναι ο βασικός της. Το ρομπότ αρχικά μπορεί να μεταδίδει εικόνα αυτών που συλλαμβάνει η κάμερα σε απομακρυσμένο υπολογιστή και ο χρήστης να τηλε-κατευθύνει το ρομπότ χωρίς ο ίδιος να βλέπει την θέση του στο χώρο όπου κινείται παρά μόνο μέσω της εικόνας που μεταδίδει η κάμερα στον υπολογιστή. Η συγκεκριμένη λειτουργία είναι αρκετά ενδιαφέρουσα για τα παιδιά και η αξία της είναι στην παρακίνηση του ενδιαφέροντος τους. Ένας δεύτερος ρόλος της κάμερας είναι αυτός της αναγνώρισης αντικειμένων μέσω εφαρμογών σε Open CV οι οποίες είναι συμβατές με το περιβάλλον του ROS. Οι εφαρμογές αυτές δεν είναι για να εξηγηθούν πλήρως και να κατανοήσουν οι μαθητές την πλήρη λειτουργία τους. Μπορούν όμως εισαγωγικά να κατανοήσουν την λειτουργία των πινάκων εξαγωγής χαρακτηριστικών και να εισαχθούν σε έννοιες όπως τα αυτόνομα κινούμενα οχήματα, αναγνώριση προσώπου κτλ.

### **Motor controller**



Εικόνα 36 H bridgeHUT

Ο ελεγκτής των κινητήρων [30] προέρχεται από το PiHut.<sup>4</sup>είναι από ένα κιτ για ρομπότ με δυο κινητήριους τροχούς και ένα σφαιρικό τροχό για ισορροπία. Το κιτ περιέχει εκτός από την πλακέτα με την γέφυρα και δύο κινητήρες, ένα αποστασιόμετρο υπερήχων, ένα οπτικό αισθητήρα μαύρης γραμμής, μια μικρή breadboard και τέσσερις αντιστάσεις. Το χιπτης γέφυρας είναι το DVR8833 και βρίσκεται σε ένα HUT το οποίο κουμπώνει επάνω στα πιν του Raspberry εικόνα 38. Ο ελεγκτής συνδέεται με τα pin του raspberry

Ο Α κινητήρας για εμπρός κίνηση στο pin10

Ο Α κινητήρας για πίσω κίνηση στο pin9

Ο Β κινητήρας για εμπρός κίνηση στο pin8

Ο Β κινητήρας για πίσω κίνηση στο pin7

### **Λογισμικό για τον έλεγχο του Ρομπότ**

Δημιουργία catkin\_ws φακέλου εργασίας εδώ δεν χρειάζεται γιατί προϋπάρχει ήδη. Και είναι στο path: /home /ubuntu/catkin\_ws

Εάν θέλουμε όμως να τον δημιουργήσουμε, η διαδικασία από την γραμμή εντολών είναι ως εξής:

```
$ mkdir -p ~/catkin_ws/src
```

```
$ cd ~/catkin_ws/src
```

```
$ catkin_init_workspace
```

Σε αυτό το φάκελο εργασίας θα φτιάξουμε ένα υποφάκελο όπου θα βάλουμε όλα τα scripts (nodes) που θα χρειαστούμε για τον έλεγχο του ρομπότ. Ο υποφάκελος αυτός έχει τέσσερις dependencies (εξαρτήσεις) δηλαδή βιβλιοθήκες rospy, std\_msgs, geometry\_msgs, και python-rpi.gpio

```
$ cd ~/catkin_ws/src
```

```
$ catkin_create_pkg kg_edukit_bot rospy std_msgs geometry_msgs -s python-rpi.gpio
```

---

<sup>4</sup>(<https://thepihut.com/collections/raspberry-pi-kits-and-bundles/products/camjam-edukit-3-robotics>)

Εδώ στον υποφάκελο src γράφουμε τα προγράμματα σε python2 γιατί το ROS είναι συμβατό μόνο με την python2 και όχι με την python3 τα οποία τα ορίζουμε ως εκτελέσιμα. Αυτά μπορούμε να τα γράψουμε κατευθείαν στο raspberry με ένα editor. Στην έκδοση του λειτουργικό που έχουμε έχει εγκατεστημένο των GNU Emacs24(GUI) με παραθυρικό περιβάλλον. Έχει επιλεγεί αυτός γιατί είναι πιο εύχρηστος και κατανοητός για τα παιδιά. Η διαδικασία αυτή επαναλαμβάνεται και στον απομακρυσμένο υπολογιστή μας. Δηλαδή η δημιουργία του catkin\_ws φακέλου εργασίας και του υποφακέλου kg\_edukit\_bot.

## Scripts τα οποία είναι nodes (διεργασίες)

### A) kg\_driver\_node\_key

Το πρώτο script είναι για τον έλεγχο των κινητήριων τροχών, και η ονομασία του είναι kg\_driver\_node\_key. Το συγκεκριμένο πρόγραμμα δέχεται μηνύματα Twist από το topic cmd\_vel. Στο topic αυτό εγγραφήκαμε με την εντολή :

```
rospy.Subscriber('cmd_vel', Twist ,self.velocity_received_callback)
```

όπου με την velocity\_received\_callback πέρνουμε τις ταχύτητες

linear=message.linear.x γραμμική ως προς τον X άξονα

angular= 3.3\*message.angular.z γωνιακή γύρω από τον Z άξονα.(εδώ ο συντελεστής 3.3 έχει υπολογιστεί εμπειρικά για να συμβαδίζει το εικονικό με το πραγματικό ρομπότ.)

Στην κλάση Motor ορίζονται τα δεδομένα για τους κινητήρες, όπως τα pin που τους ελέγχουν και οι PWM τιμές για τις ταχύτητες που θα παίρνουν.

Στην κλάση Driver ορίζονται διάφορες σταθερές όπως ο χρόνος timeout οποίος είναι χρόνος ασφαλείας μιας και αν το ρομπότ χάσει το σήμα από την πηγή που ελέγχεται να μην συνεχίσει να κινείται με την προηγούμενη ταχύτητα. Το max\_speed η μέγιστη ταχύτητα για να μπορούμε να ορίσουμε τις υπόλοιπες σαν ποσοστό αυτής. Ενώ το wheel\_base είναι το πλάτος του ρομπότ από το κέντρο της επιφάνειας που πατά ο ένας τροχός έως το κέντρο του άλλου. Γίνεται η εγγραφή στο topic cmd\_vel και γίνεται και ο υπολογισμός των γωνιακών ταχυτήτων που θα δώσουμε στους τροχούς για αν κινηθεί το ρομπότ. Θεωρώντας Θετική φορά περιστροφής αυτή των δεικτών



του ρολογιού η ταχύτητα του αριστερού τροχού θα είναι η γραμμική που πήραμε μείον την γωνιακή επί την βάση του ρομπότ διά δύο.

```
left_speed= linear – angular*self.wheel_base/2
```

για τον δεξιό τροχό αντί για αφαίρεση έχουμε πρόσθεση.

```
right_speed= linear + angular*self.wheel_base/2
```

Τέλος στη συνάρτηση run γίνεται ο έλεγχος εάν έχει ληφθεί σήμα για κίνηση ένα ναι τότε κινείται με τις υπολογισμένες ταχύτητες αλλιώς σταματά.

Αφού το σώσουμε το κάνουμε εκτελέσιμο τρέχοντας την εντολή

```
$ chmod a+x ~/catkin_ws/src/kg_edukit_bot/src/kg_driver_node
```

## **B) joint\_state\_controller2.py**

Το δεύτερο script είναι για τον έλεγχο των σερβοκινητήρων του ρομποτικού βραχίονα και η ονομασία του είναι joint\_state\_controller2.py. Το συγκεκριμένο πρόγραμμα εγγράφεται στο topic /kg\_robot\_arm/joint\_states όπου τα μηνύματα είναι του τύπου JointState και ξεκινά το node wonder. Τα μηνύματα εδώ προέρχονται από την εφαρμογή του Moveit.

Για να ελεγχθούν εδώ οι σερβοκινητήρες σύμφωνα με τον κατασκευαστή λειτουργούν με σήματα PWM στα 50 Hz, δηλαδή η περίοδος του παλμού PWM είναι 20ms. Η θέση του servo εξαρτάται από το εύρος του παλμού ON. Τα περισσότερα servo βρίσκονται στην ακραία αριστερή θέση με διάρκεια παλμού ON περίπου 1ms. Αυτό αντιστοιχεί στο 5% της συνολικής περιόδου, δηλαδή το Duty cycle στην περίπτωση αυτή είναι 5 με μέγιστο το 100. Τα περισσότερα servo βρίσκονται στην μεσαία θέση με διάρκεια παλμού περίπου 1.5ms. Αυτό αντιστοιχεί στο 7.5% της συνολικής διάρκειας ή αλλιώς, το duty cycle είναι 7.5 με μέγιστο το 100. Τα περισσότερα servo βρίσκονται στην ακραία δεξιά θέση με διάρκεια παλμού περίπου 2ms. Αυτό αντιστοιχεί στο 10% της συνολικής διάρκειας, ή αλλιώς σε duty cycle 10 με μέγιστο το 100. Άρα, η παράμετρος duty cycle θα πρέπει να μεταβάλλεται γραμμικά από 5 έως 10 για να αλλάξει διαδοχικά η θέση του άξονα του μοτέρ από τη θέση αριστερά μέχρι τη θέση δεξιά. Οι τιμές αυτές ήταν λίγο διαφορετικές στην περίπτωση μας. Στην περίπτωση μας υπήρχε κι ένα άλλο πρόβλημα αυτό της

ταλάντωσης των σέρβο όταν παρέμεναν σε μία θέση. Με την απλή βιβλιοθήκη RPI.GPIO η οποία περιείχε την συνάρτηση `pwm.ChangeDutyCycle( )` εάν δινόταν μία σταθερή τιμή το σέρβο ταλαντωνόταν. Αυτό συμβαίνει επειδή το Raspberry Pi είναι ένας πλήρης υπολογιστής και όχι ένας μικροελεγκτής. Αντί να κάνει μόνο ένα πράγμα κάθε φορά, το Pi έχει ένα λειτουργικό σύστημα πολλαπλών εργασιών που πρέπει να κάνει πολλά πράγματα ταυτόχρονα. Για αυτό το λόγο χρησιμοποιείτε εδώ η βιβλιοθήκη `piGPIO`<sup>56</sup> ή οποία είναι μια βιβλιοθήκη χαμηλού επιπέδου για έλεγχο των pin του Pi. Πριν τρέξουμε το πρόγραμμα `joint_state_controller2.py` για τον έλεγχο των σερβοκινητήρων τρέχουμε στην γραμμή εντολών:

```
$ sudo pigpiod
```

Τελικά για τον κάθε σερβοκινητήρα γίνεται μία μαθηματική μετατροπή από γωνίες που παίρνουμε σε ακτίνια σε από το `topic` σε τιμές από 1500 για 0°, 2500 για 90° και 3500 για 180°.

Υπολογισμός γωνίας σε μοίρες

```
gwnFirstj = 90.+(joint_first/1.57)*90
```

εύρεση τιμής για την συνάρτηση που δίνει το `duty_cycle`

```
y1= (200./18.)*gwnFirstj+500.
```

Μετατροπή του σε ακέραιο

```
y1=int(y1)
```

Η τελική τιμή στη συνάρτηση που ελέγχει τον πρώτο σερβοκινητήρα, το 17 είναι το pin που του αντιστοιχεί.

```
pFirstj.set_servo_pulsewidth(17,y1)
```

Αφού το σώσουμε το κάνουμε εκτελέσιμο τρέχοντας την εντολή όπως πριν.

```
$ chmod a+x ~/catkin_ws/src/kg_edukit_bot/src/joint_state_controller2.py
```

---

<sup>5</sup> (<https://github.com/joan2937/pigpio/blob/master/README>)

<sup>6</sup> (<http://abyz.me.uk/rpi/pigpio/index.html>)

Άλλα δύο scripts που υπάρχουν στον φάκελο είναι το Servo\_Publisher και το Servo\_Subscriber. Είναι παραδείγματα απλά για την επίδειξη ενός Publisher κι ενός Subscriber του ROS στους μαθητές κατά την διάρκεια των μαθημάτων. Η έννοια του Publisher και του Subscriber είναι θεμελιώδης για το ROS. Το μεν πρώτο είναι ένα πρόγραμμα το οποίο δημιουργεί ένα NODE το οποίο εδώ δημοσιεύει σ ένα TOPIC κάποια μηνύματα με ένα ρυθμό. Το δεύτερο δημιουργεί ένα NODE το οποίο εγγράφεται στο προηγούμενο TOPIC που δημιουργήθηκε και ακούει τα μηνύματα τα οποία στέλνονται, και τελικός ελέγχει το σερβοκινητήρα. Η ένωση αυτών των δύο NODE γίνεται μέσω του ROSmaster. Αυτή είναι κι όλη η φιλοσοφία του ROS μπορούμε να τρέχουμε διάφορα NODES τα οποία να συντελούν διαφορετικές διαδικασίες κι όλα μαζί να επικοινωνούν και να συνεργάζονται. Το σημαντικό είναι ότι δεν χρειάζεται να είναι στον ίδιο των υπολογιστή. Ο ROSmaster μπορεί να τρέχει είτε στο ρομπότ είτε σ ένα άλλο υπολογιστή.

### **Γ)Ultrasound\_Publisher**

Το Ultrasound\_Publisher δημιουργεί ένα NODE με αυτό το όνομα το οποίο δημοσιεύει της αποστάσεις που βρίσκει μπροστά από το ρομπότ ο αισθητήρας υπερήχων, σε ένα topic με την ονομασία distance. Για να ενεργοποιηθεί ο αισθητήρας και να διαβαστεί η απόσταση έχει δημιουργηθεί μια συνάρτηση η Read\_Distance η οποία διεγείρει το πιν αποστολής σήματος κι αναμένει την επιστροφή της ανάκλασης του ήχου και υπολογίζει με βάση το χρόνο από την στιγμή της αποστολής μέχρι τη στιγμή επιστροφής της ανάκλασης την απόσταση μπροστά από το εμπόδιο. Στο παρόν πρόγραμμα ξεκινάει ένα Node με την εντολή

```
rospy.init_node('Ulltrasound_Publisher') και δημοσιεύει τις τιμές με
```

```
pub= rospy.Publisher('distance',Float32,queue_size=10)
```

### **Τρόπος επικοινωνίας Ρομπότ με τον απομακρυσμένο υπολογιστή.**

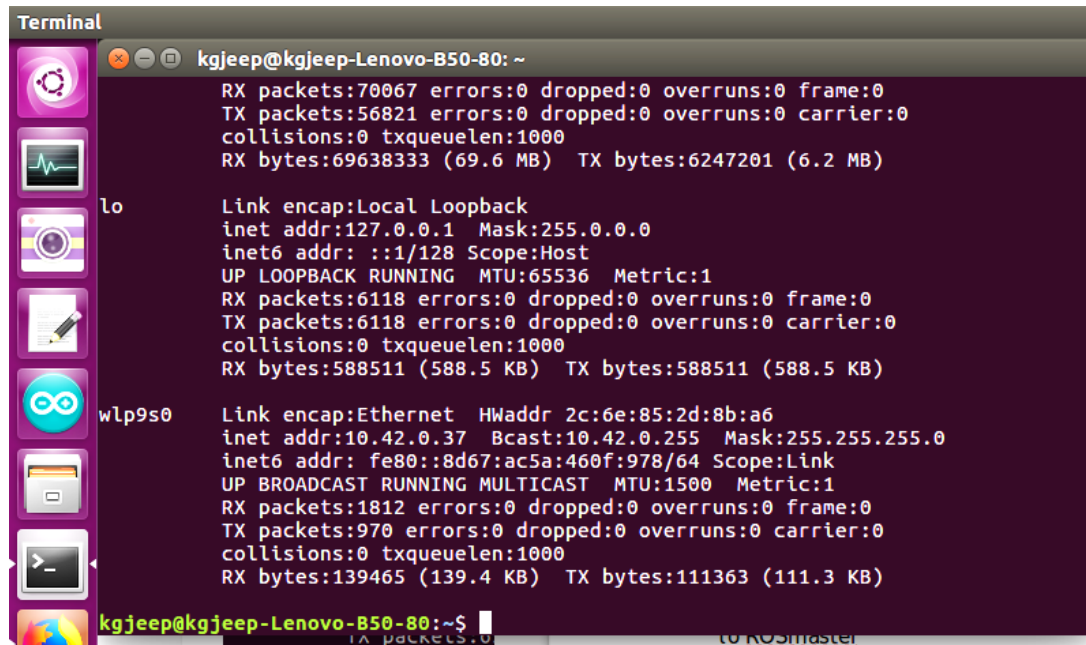
Για να γίνεται ή επικοινωνία όμως μέσω του δικτύου που δημιουργεί το Raspberry θα πρέπει να πληροφορεί ο υπολογιστής αλλά και το ρομπότ για το που να βρίσκεται το ένα το άλλο.<sup>7</sup>

---

<sup>7</sup>(<https://razbotics.wordpress.com/>)

Στον υπολογιστή όπου θα τρέχουμε τον ROSmaster συμπληρώνουμε κάποιες γραμμές στο αρχείο .bashrc στο root του υπολογιστή.

Αυτό γίνεται με τον παρακάτω τρόπο. Συνδεόμαστε με το δίκτυο που έχει δημιουργήσει το raspberry.



```
kgjeep@kgjeep-Lenovo-B50-80: ~
RX packets:70067 errors:0 dropped:0 overruns:0 frame:0
TX packets:56821 errors:0 dropped:0 overruns:0 carrier:0
collisions:0 txqueuelen:1000
RX bytes:69638333 (69.6 MB) TX bytes:6247201 (6.2 MB)

lo
Link encap:Local Loopback
inet addr:127.0.0.1 Mask:255.0.0.0
inet6 addr: ::1/128 Scope:Host
UP LOOPBACK RUNNING MTU:65536 Metric:1
RX packets:6118 errors:0 dropped:0 overruns:0 frame:0
TX packets:6118 errors:0 dropped:0 overruns:0 carrier:0
collisions:0 txqueuelen:1000
RX bytes:588511 (588.5 KB) TX bytes:588511 (588.5 KB)

wlp9s0
Link encap:Ethernet HWaddr 2c:6e:85:2d:8b:a6
inet addr:10.42.0.37 Bcast:10.42.0.255 Mask:255.255.255.0
inet6 addr: fe80::8d67:ac5a:460f:978/64 Scope:Link
UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1
RX packets:1812 errors:0 dropped:0 overruns:0 frame:0
TX packets:970 errors:0 dropped:0 overruns:0 carrier:0
collisions:0 txqueuelen:1000
RX bytes:139465 (139.4 KB) TX bytes:111363 (111.3 KB)

kgjeep@kgjeep-Lenovo-B50-80:~$
```

Εικόνα 37 η IP διεύθυνση του υπολογιστή

Πληκτρολογούμε στη γραμμή εντολών στο υπολογιστή όπου θα τρέχουμε το ROSmaster

```
$ ifconfig
```

για να μάθουμε την IP διεύθυνση του υπολογιστή. Στη δικιά μας περίπτωση ήταν 10.42.0.37 και μετά πληκτρολογούμε στην γραμμή εντολών:

```
$ gedit .bashrc
```

Πρέπει να πούμε στον υπολογιστή μας ποιος είναι ο κύριος υπολογιστής που τρέχει το ROS και τι IP διεύθυνση έχει στο δίκτυο. Δεδομένου ότι οι υπολογιστές που χρησιμοποιούν ROS δεν γνωρίζουν τη διεύθυνση IP. Πηγαίνουμε στο τέλος του αρχείου και προσθέτουμε τις επόμενες τρεις γραμμές:

```
export ROS_MASTER_URI=http://localhost:11311/
```

```
export ROS_HOSTNAME= 10.42.0.37
```

```
export ROS_IP=10.42.0.37
```

Στην πρώτη γραμμή καθορίζουμε τη διεύθυνση του υπολογιστή που τρέχει ο ROSmaster μαζί με τη θύρα που θα λειτουργήσει. Βασικά λέμε στον κύριο υπολογιστή μας ότι είναι ο HOST έτσι ώστε για να καθορίσει τη διεύθυνση IP του λέμε χρησιμοποιήσει τη δική σου localhost διεύθυνση IP.

Τελικά πρέπει να σώσουμε το αλλαγμένο αρχείο και να εκτελέσουμε την παρακάτω εντολή.

```
$ source .bashrc
```

Την ίδια δουλειά σχεδόν πρέπει να κάνουμε και στο Raspberry. Αφού βρούμε με τον ίδιο τρόπο όπως πριν την IP διεύθυνση που έχει δώσει στον εαυτό του. Η διεύθυνση είναι 10.42.0.1 και εδώ πρέπει να πάμε στο αρχείο .bashrc στο root του raspberry

```
$ nano .bashrc
```

και να συμπληρώσουμε στο τέλος

```
export ROS_MASTER_URI=http://10.42.0.37:11311/
```

```
export ROS_HOSTNAME=10.42.0.1
```

```
export ROS_IP=10.42.0.1
```

Τώρα η αλλαγή που κάναμε εδώ είναι ορίζουμε το ROS\_MASTER\_URI ως http://10.42.0.37:11311/ όπου 10.42.0.37 είναι η διεύθυνση IP του master PC όπου τρέχει ο ROSmaster. Λέμε δηλαδή τη διεύθυνση IP του κύριου υπολογιστή στο Raspberry. Η υπόλοιπες δύο γραμμές καθορίζουν τη διεύθυνση IP της Raspberry. Τέλος κι εδώ πρέπει να σώσουμε το αλλαγμένο αρχείο.

```
$ source .bashrc
```

### **Έτοιμες εφαρμογές του ROS για τον έλεγχο του ρομπότ.**

Με την εφαρμογή teleop\_twist\_keyboard.py [36] μπορούμε να ελέγχουμε το ρομπότ μας από τον απομακρυσμένο υπολογιστή. Η εφαρμογή πατώντας τα κατάλληλα πλήκτρα δημιουργεί μηνύματα twist. Τα μηνύματα twist είναι μηνύματα τα οποία εκφράζουν γραμμικές και γωνιακές ταχύτητες. Στην περίπτωσή μας δίνουν τιμές σε

ταχύτητες στον X άξονα και γωνιακές ως προς τον Z άξονα. Τα μηνύματα διαχέονται στο Topic `cmd_vel`, τα οποία ακούει το node (`kg_driver_node_key`) που τρέχει στο ρομπότ και μ αυτόν τον τρόπο ελέγχεται. Η εφαρμογή τρέχει κατευθείαν από τη γραμμή εντολών:

Πρώτα σε ένα παράθυρο στη γραμμή εντολών ξεκινάμε το Ros master

```
$ roscore
```

Κι έπειτα ανοίγοντας νέο παράθυρο τρέχουμε

```
$ roslaunch teleop_twist_keyboard teleop_twist_keyboard.py
```

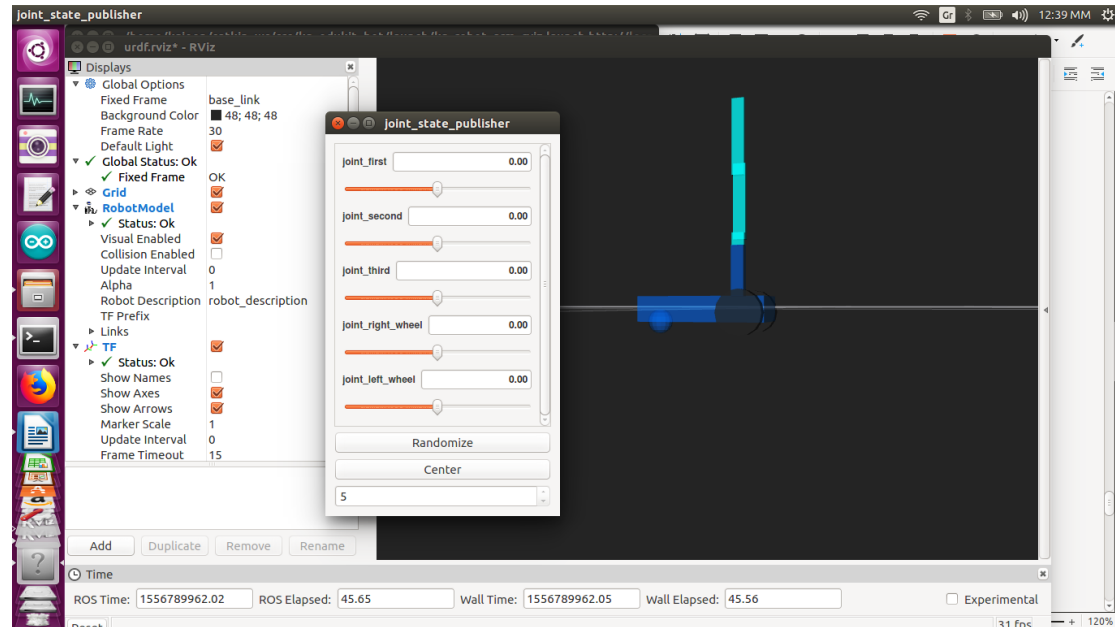
Άλλη μια εφαρμογή για τον έλεγχο της κίνησης του ρομπότ ή οποία προκαλεί μεγάλη εντύπωση στους μαθητές είναι από τι κινητό τηλέφωνο[42]. Πηγαίνοντας με το Android κινητό μας στο Google Playstore μπορούμε να κατεβάσουμε εντελώς δωρεάν την εφαρμογή Ros Control, η οποία δημιουργεί ένα node στο κινητό όπως πριν στον υπολογιστή, και να επικοινωνεί κι αυτό με το ρομπότ μέσω του topic `cmd_vel` στέλνοντας του μηνύματα `twist`. Η εφαρμογή λειτουργεί σαν joystick. Πριν την ανοίξουμε πρέπει να έχουμε συνδέσει το κινητό με το ασύρματο δίκτυο το οποίο έχει δημιουργήσει το ρομπότ μας και είναι συνδεδεμένος κι ο υπολογιστής στον οποίο τρέχει ο Ros master.

### **Προσομοίωση του ρομπότ.**

Το Ros μας δίνει και την δυνατότητα να προσομοιώσουμε το ρομπότ μας και σε εικονικό περιβάλλον. Για να γίνει αυτό πρέπει πρώτα να περιγραφεί το ρομπότ σ ένα αρχείο URDF σε γλώσσα XML. Αυτό το αρχείο μαζί με άλλα αρχεία του θα χρειαστούν για τις προσομοιώσεις θα μουν σ' ένα φάκελο `kg_edukit_bot` κάτω από τον φάκελο `~/catkin_ws/src/`. Στον φάκελο αυτό θα δημιουργηθούν άλλοι τέσσερις υποφάκελοι οι `urdf`, `config`, `launch` ,και `src`. Το URDF με ονομασία `kg_robot_arm1_new.urdf` θα μπει στο φάκελο `urdf`. Το αρχείο αυτό περιέχει μια γεωμετρική κατ' αρχάς περιγραφή του ρομπότ το οποίο μπορούμε να παρακολουθήσουμε στο περιβάλλον Rviz[39]. Μαζί με την προσομοίωση ξεκινάει και ένα γραφικό περιβάλλον όπου έχει όλες τις κινητές ενώσεις στο ρομπότ τις οποίες μπορούμε να τις ελέγχουμε. Εάν δεν θέλουμε να ξεκινάει τότε στο τέλος της παρακάτω εντολής στο παράθυρο εντολών συμπληρώνουμε `gui:=False`. Το Rviz

υπάρχει εγκατεστημένο ήδη με την εγκατάσταση του Ros. Η εντολή για να τρέξει η προσομοίωση:

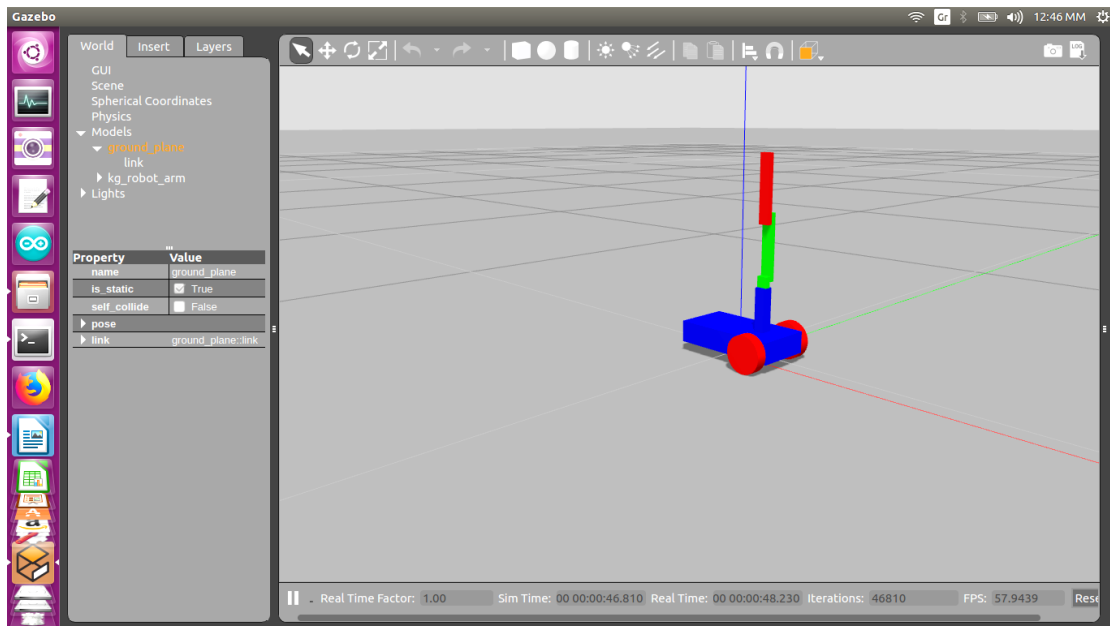
```
$ roslaunch kg_edukit_bot kg_robot_arm_rviz.launch
```



Εικόνα 38 Το ρομπότ στο Rviz.

Η περιγραφή μας όμως στο αρχείο `kg_robot_arm1_new.urdf` περιέχει και άλλα χαρακτηριστικά όπως βάρη, ροπές αδράνειας των διαφόρων μερών του, τρόπους κίνησης και ελεγκτές των τροχών του και των βραχιόνων του τριβών κτλ. Αυτά υπάρχουν για να προσομοιώσουμε το ρομπότ μας και στο εικονικό περιβάλλον του Gazebo[29]. Το Gazebo υπάρχει εγκατεστημένο ήδη με την εγκατάσταση του Ros. Όπου εδώ η προσομοίωση γίνεται με όρους φυσικής και μπορούμε να παρατηρούμε την κίνηση του ρομπότ σε εικονικό “πραγματικό” περιβάλλον. Εκτός από τα δεδομένα που περιέχονται στην περιγραφή του ρομπότ μας στο `kg_robot_arm1_new.urdf` για να τρέξει η προσομοίωση στο gazebo χρειάζονται και μερικά ακόμη στοιχεία. Η εντολή για να τρέξει η προσομοίωση:

```
$ roslaunch kg_edukit_bot kg_robot_arm_xacro_gazebo.launch
```



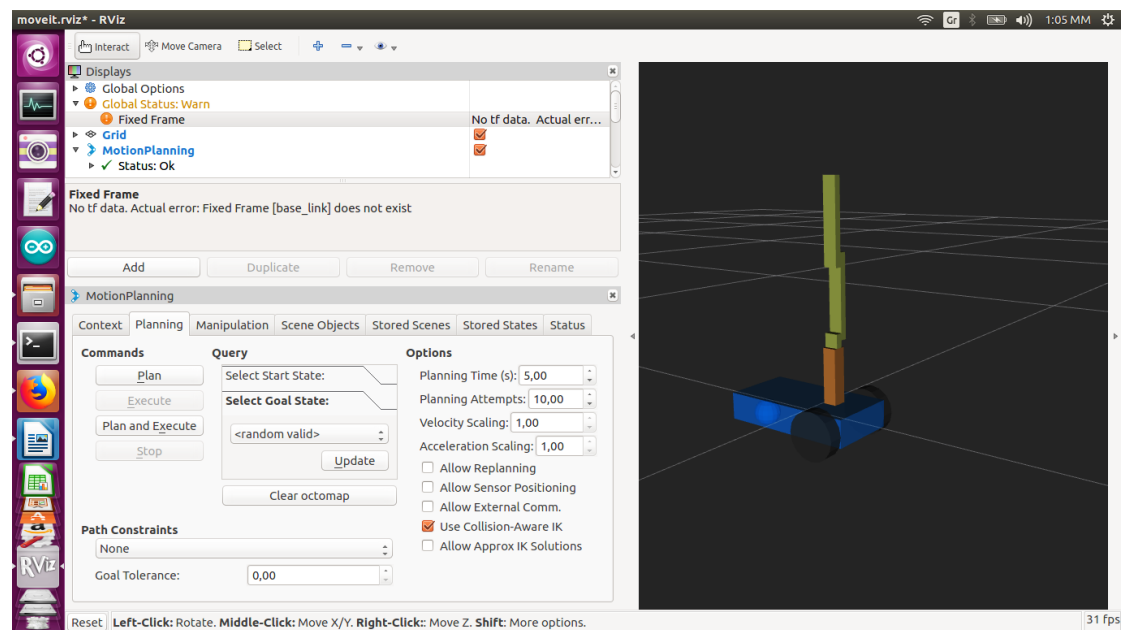
Εικόνα 39 Το ρομπότ στο Gazebo.

Προσομοίωση του ρομπότ και στο Moveit[40]. Το Moveit παρέχει λειτουργίες για κινηματική, σχεδιασμό κίνησης / διαδρομής, έλεγχο σύγκρουσης, 3D αντίληψη, αλληλεπίδραση ρομπότ με το περιβάλλον και πολλά άλλα. Το MoveIt αποτελεί την κύρια πηγή πολλών λειτουργιών για έλεγχο της κίνησης ρομποτικών βραχιόνων στο ROS. Το MoveIt βασίζεται στα μηνύματα ROS και δημιουργεί αντίστοιχα (URDF αρχεία XML) τα οποία βασίζονται στα αρχικά (URDF αρχεία XML) που έχουμε δημιουργήσει για την περιγραφή του ρομπότ μας. Χρησιμοποιεί και το Visualizer ROS (RVIZ). Με το Moveit αυτό που καταφέρνουμε είναι να ελέγχουμε τον ρομποτικό βραχίονα επάνω στο ρομπότ. Η εύρεση της διαδρομής από μία θέση του βραχίονα σε μια άλλη γίνεται μέσω επιλυτών τροχιών που έχει ή εφαρμογή. Μ' αυτή την εφαρμογή γίνεται ο έλεγχος του εικονικού αλλά και του πραγματικού βραχίονα, που βρίσκεται επάνω στο ρομπότ μας. Η κίνηση του βραχίονα είναι μία δύσκολη υπόθεση μιας κι έχει πολλά κινητά μέρη και η εκτέλεση της διαδρομής θέλει συνεχείς υπολογισμούς. Λύσεις τέτοιων κινηματικών προβλημάτων ιδίως ανάστροφων είναι ιδιαίτερα δύσκολα και απαιτούν προχωρημένες γνώσεις γραμμικής άλγεβρας και πράξεις πινάκων. Η κίνηση του βραχίονα από μια αρχική θέση σε μία άλλη δίνοντας στους σερβοκινητήρες κάποιες τιμές γωνιών είναι απλή αλλά εάν θέλουμε να πάμε από μια τυχαία θέση σε μια άλλη θέση η εύρεση των τιμών των γωνιών που πρέπει να πάρουν οι σερβοκινητήρες δεν είναι απλή. Εάν προσθέσουμε και κάποια εμπόδια στον χώρο κίνησης του, τότε η αποστολή μας δυσκολεύει ακόμη αρκετά. Αυτή την δουλειά την κάνει το Moveit κι εμείς απλά θέτουμε τα σημεία που θέλουμε να



περάσει ο βραχίονας. Η δημιουργία των κατάλληλων αρχείων για την εφαρμογή αυτή γίνεται πολύ απλά ακολουθώντας το Tutorial του Moveit Assistant, αφού πρώτα έχουμε εγκαταστήσει το Moveit στον υπολογιστή μας <https://moveit.ros.org/install/>. Η εντολή για να τρέξει η προσομοίωση:

```
$ roslaunch kg_edukit_moveit demo_planning_execution.launch
```



Εικόνα 40 Το ρομπότ στο Moveit.

### Εικόνα από την κάμερα του Raspberry

Για να μπορεί να μεταδοθεί εικόνα από την κάμερα με τον ίδιο τρόπο όπως και τα μηνύματα των ταχυτήτων πρέπει να τρέξει ένα node [35] και να μεταδίδει μηνύματα βίντεο. Το Ros έχει μια έτοιμη εφαρμογή γι' αυτό το πράγμα η οποία αργότερα μπορεί να χρησιμοποιηθεί και για αναγνώριση εικόνας ή ακόμη και για πιο προχωρημένες εφαρμογές, όπως το να ακολουθεί ένα άνθρωπο, από το ρομπότ, ή και αυτόνομη οδήγηση. Θα εγκαταστήσουμε το `raspicam_node` πληκτρολογώντας στην γραμμή εργασιών του raspberry :

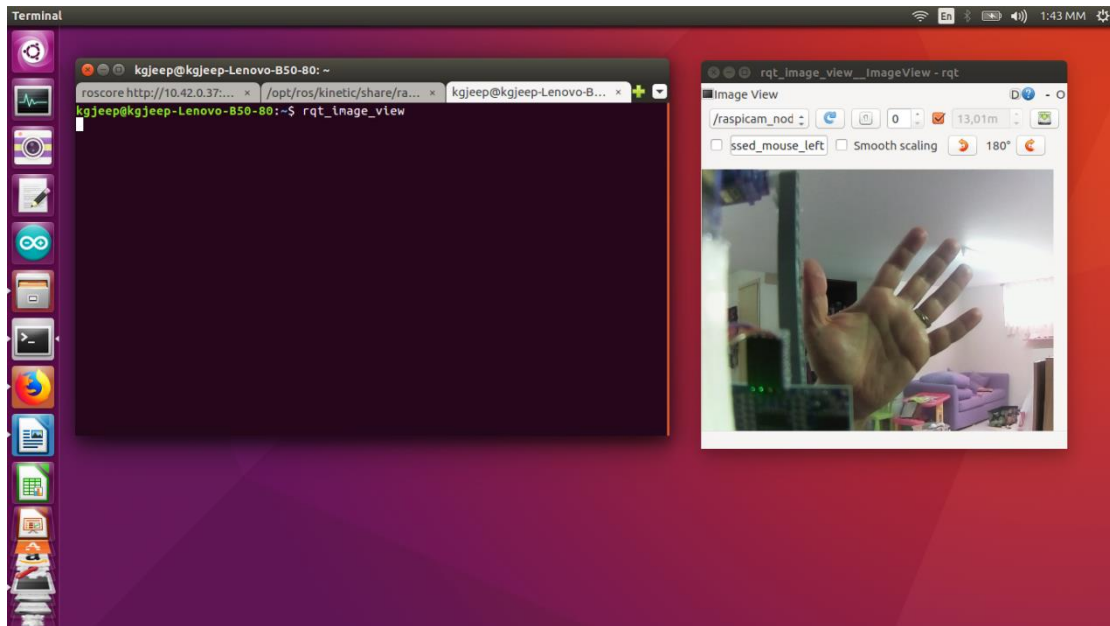
```
$ sudo apt install ros-kinetic-raspicam-node
```

Η παραπάνω εντολή εγκαθιστά το κατάλληλο λογισμικό για την λειτουργία της κάμερας τα Launch αρχεία εγκαθίστανται στον φάκελο `/opt/ros/kinetic/share/raspicam_node/launch`. Για να αρχίσει η μετάδοση εικόνας πρέπει να τρέξουμε το κατάλληλο node ανάλογα με την ανάλυση που θέλουμε. Από την γραμμή των εργασιών του Raspberry πληκτρολογούμε :

```
$ roslaunch raspicam_node camerav2_1280x960.launch
```

και στην γραμμή εργασιών του απομακρυσμένου υπολογιστή :

```
$ rqt_image_view
```



Εικόνα 41 εικόνα κάμερας

### **Χρησιμοποίηση των αισθητήρων οπτοδιακοπής στους τροχούς.**

Οι αισθητήρες οπτοδιακοπής είναι πολύ χρήσιμοι για την μέτρηση της ταχύτητας και της μετατόπισης του ρομπότ μας. Κάθε φορά που στρέφεται ο τροχός, ο αισθητήρας φωτός εναλλάξ την μια δέχεται φως και μια όχι. Οι δίσκοι που χρησιμοποιούμε έχουν 20 διάκενα, που σημαίνει ότι σε μια πλήρη περιστροφή του ο αισθητήρας έχει 20 ανάμματα και 20 σβήσιματα. Άρα ένα άναμμα ή σβήσιμο αντιστοιχεί σε  $18^\circ$  μοίρες περιστροφής του τροχού, και σε  $2\pi/20 * 3.25 = 1.02\text{cm}$  μετατόπισης, όπου  $3.25\text{cm}$  η ακτίνα του τροχού μας. Η ακρίβεια που μπορούμε να πετύχουμε είναι ακόμη μεγαλύτερη εάν τους μετρητές των στροφών δεν τους βάζουμε να μετράνε τα ανάμματα ή τα σβήσιματα αλλά την κάθε αλλαγή, δηλαδή το άναμμα-σβήσιμο η το σβήσιμο άναμμα. Τώρα ο μετρητής θα μετρά 40 τέτοιες καταστάσεις ανά περιστροφή του τροχού και η ακρίβεια της περιστροφής θα γίνει  $9^\circ$  μοίρες και αυτή της μετατόπισης  $0.501\text{cm}$ .

## 1η Εφαρμογή rpm.py

Μία εφαρμογή η οποία μας δίνει την μεταφορική ,και γωνιακή ταχύτητα του ρομπότ και μας δίνει και την θέση εάν υποθέσουμε ότι ξεκινά από την θέση ( $\chi=0,\psi=0$ ) και με προσανατολισμό  $\phi=0$  είναι η rpm.py η οποία βρίσκεται στην επιφάνεια εργασίας(desktop) του Raspberry. Την συγκεκριμένη εφαρμογή μπορούμε να την χρησιμοποιήσουμε με την τηλε-κατεύθυνση του ρομπότ και να βρίσκουμε κάθε στιγμή την ταχύτητα και την θέση του. Είναι ένα απλό παράδειγμα οδομετρίας που χρήσιμο για την εισαγωγή αυτής της έννοιας της ρομποτικής στους νέους μαθητές. Αυτό που κάνουμε στο συγκεκριμένο πρόγραμμα είναι να υπολογίζουμε την γωνιακή συχνότητα του κάθε τροχού και να την μετατρέπουμε σε γραμμική με τον τύπο:

$$vel\_R=(rpm\_r*2*\pi*Wheel\_Radius)/60$$

όπου vel\_R η γραμμική ταχύτητα του δεξιού τροχού, rpm\_r η συχνότητα σε στροφές ανά λεπτό και Wheel\_Radius η ακτίνα των τροχών. Η διαίρεση διά 60 γίνεται για μετατροπή της ταχύτητας από cm/min σε cm/sec. Ομοίως υπολογίζουμε και την γραμμική ταχύτητα του αριστερού τροχού

$$vel\_L=(rpm\_l*2*\pi*Wheel\_Radius)/60$$

Ο υπολογισμός rpm\_r και rpm\_l γίνεται με τον εξής τρόπο. Στο πρόγραμμα έχουμε θέση ένα μετρητή ο οποίος ενεργοποιείται κάθε φορά που ο αισθητήρας οπτοδιακοπής του κάθε τροχού από κατάσταση LOW πάει σε κατάσταση HIGH. Αυτό στην python με εφαρμογή στο Raspberry γίνεται με μια συνάρτηση Interrupt την GPIO.add\_event\_detect(sensor\_l, GPIO.RISING, callback=get\_rpm\_l). Τα πρώτο όρισμα στην συνάρτηση είναι το pin που είναι συνδεδεμένος ο οποίος αλλάζει κατάσταση κι ενεργοποιεί την συνάρτηση, το δεύτερο δηλώνει ότι θα ενεργοποιείται η συνάρτηση κάθε φορά που ο αισθητήρας θα πηγαίνει από κατάσταση LOW σε κατάσταση HIGH και το τρίτο όρισμα έχει την συνάρτηση που θα ξεκινά να εκτελείτε κάθε φορά που συμβαίνει το γεγονός που είχαμε στο δεύτερο όρισμα. Εδώ θα μπορούσαμε να είχαμε βάλει σαν γεγονός αλλαγής αντί το GPIO.RISING το GPIO.CHANGE το οποίο θα ενεργοποιούνταν είτε ο αισθητήρας πήγαινε από LOW σε HIGH ή από HIGH σε LOW, έχοντας όπως είχαμε αναφέρει και πιο πάνω μεγαλύτερη ακρίβεια ή οποία όμως στην παρούσα φάση δεν κρίνεται

απαραίτητη. Με το που συμπληρώσει 20 τέτοιες καταστάσεις καταλαβαίνει ότι έχει γίνει μια περιστροφή του τροχού. Στο ξεκίνημα του μετρήματος υπολογίσαμε το χρόνο καθώς και στο τέλος των 20 αλλαγών κατάστασης. Αφαιρώντας από την τελική χρονική στιγμή -όταν συμπληρώθηκε ο αριθμός 20 – την αρχική χρονική στιγμή προκύπτει ο χρόνος που χρειάστηκε για μια πλήρη περιστροφή ο τροχός. Η συχνότητα rpm\_l ή rpm\_r υπολογίζεται από τους παρακάτω τύπους:

$$\text{rpm}_r = (20 / \text{delta}_r) / 20 \quad \text{ή} \quad 1 / \text{delta}_r$$

$$\text{rpm}_l = (20 / \text{delta}_l) / 20 \quad \text{ή} \quad 1 / \text{delta}_l$$

όπου delta\_r ο χρόνος που χρειάστηκε για μια περιστροφή ο δεξιός τροχός και delta\_l ο αριστερός.

Η συνολική μεταφορική ταχύτητα του ρομπότ είναι το άθροισμα των δύο προηγούμενων ταχυτήτων διά δύο

$$V = (\text{vel}_R + \text{vel}_L) / 2$$

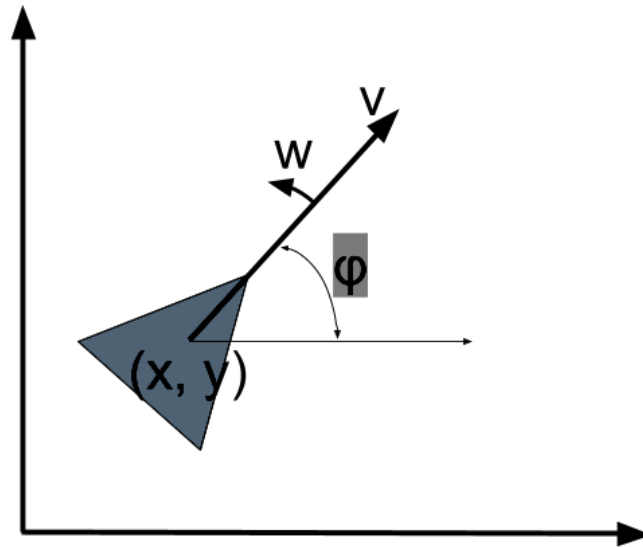
όπου V η γενική μεταφορική ταχύτητα του ρομπότ σε μονάδες cm/sec.

Ο υπολογισμός της γωνιακής ταχύτητας θεωρώντας θετική την δεξιόστροφη στροφή (όπως των δεικτών του ρολογιού) γίνεται με τον παρακάτω τύπο:

$$W = (\text{vel}_R - \text{vel}_L) / \text{Platos}$$

όπου W η γωνιακή ταχύτητα σε rad/sec και Platos το πλάτος του ρομπότ από το κέντρο της επιφάνειας επαφής του αριστερού τροχού έως το κέντρο της επιφάνειας επαφής του δεξιού τροχού.

Το ρομπότ μας κινείται στο επίπεδο γι' αυτό έχει κάθε στιγμή μια μεταφορική ταχύτητα στον X άξονα καθώς και στο Ψ. Για να υπολογίσουμε τη θέση στο επίπεδο του ρομπότ μας αρχικά πρέπει να υπολογίσουμε αυτές τις ταχύτητες. Παρατηρώντας το παρακάτω σχήμα (εικόνα 44) καταλαβαίνουμε ότι οι δύο ταχύτητες που ζητάμε προκύπτουν από ένα πολλαπλασιασμό της γενικής μεταφορικής ταχύτητας με το συν(φ) της γωνίας φ για την ταχύτητα στον X άξονα και με το ημ(φ) για την ταχύτητα στον Ψ άξονα.



Εικόνα 42 σχήμα κινηματικής ρομπότ

Συνεπώς:

$$V_x = V \cdot \cos(\varphi) \text{ και}$$

$$V_y = V \cdot \sin(\varphi)$$

Η γωνία  $\varphi$  υπολογίζεται κάθε φορά προσθέτοντας την προηγούμενη γωνιακή μετατόπιση  $d\varphi$  με την επόμενη. Η  $d\varphi$  προκύπτει πολλαπλασιάζοντας την γωνιακή ταχύτητα με το χρονικό διάστημα που χρειάστηκε το τροχός να γυρίσει μία στροφή. Στο πρόγραμμα αυτό γράφεται ως εξής :

$\varphi = \varphi + \text{γωνιακή} \cdot (\text{delta}_r + \text{delta}_l) / 2$  όπου  $\text{delta}_r$  ο χρόνος που χρειάστηκε για μια περιστροφή ο δεξιός τροχός και  $\text{delta}_l$  ο αριστερός. Εδώ υπολογίζουμε την μέση τιμή των δύο χρόνων.

Παρόμοια υπολογίζεται και η θέση  $X, Y$  του ρομπότ στο επίπεδο .

$$X = X + V_x \cdot (\text{delta}_r + \text{delta}_l) / 2 \text{ και}$$

$$Y = Y + V_y \cdot (\text{delta}_r + \text{delta}_l) / 2$$

Τέλος τυπώνει όλα τα δεδομένα στην οθόνη .

Για να εκτελεστεί η συγκεκριμένη εφαρμογή πάμε στο φάκελο Desktop από την γραμμή εντολών

```
$ cd Desktop/
```

και μετά πληκτρολογούμε

```
~/Desktop$ python rpm.py
```

## 2η Εφαρμογή rpm\_move2.py

Και αυτή η εφαρμογή χρησιμοποιεί τους αισθητήρες οπτοδιακοπής με τις ίδιες Interrupt συναρτήσεις όπως προηγουμένως. Αυτή όμως η εφαρμογή κάνει κάτι διαφορετικό δεν περιμένει από τον χρήστη να τηλεκατευθύνει το ρομπότ αλλά ζητάει από τον χρήστη μια θέση στο επίπεδο και κατευθύνει το ρομπότ σ αυτή την θέση. Παρέχοντας ο χρήστης την θέση X και Θέση Ψ υπολογίζεται η απόσταση που πρέπει να μετακινηθεί το ρομπότ.

$$\text{Distance} = \sqrt{(x^2 + y^2)}$$

$$\text{Distance} = \text{math.sqrt}(x^{**2} + y^{**2})$$

επίσης υπολογίζει και την γωνία που πρέπει να στραφεί

$$\Theta = \pi/2 - \arctan(y/x)$$

$$\Theta = (\text{math.pi}/2 - \text{math.atan2}(y/x))$$

Επίσης υπολογίζει και το τόξο που θα στραφεί κάθε τροχός για να περιστραφεί το ρομπότ σ αυτή την γωνία

$S = \Theta * 0.14 / 2$  όπου 0.14 το πλάτος του ρομπότ από το κέντρο του ενός τροχού στο άλλο .

Για να κάνει την κίνηση όμως πρέπει να του πούμε για πόσες εναλλαγές της κατάστασης του πιν που είναι συνδεδεμένο με τον αισθητήρα οπτοδιακοπής να περιστρέφεται. Άρα θα περιστραφεί για count -αριθμός εναλλαγών- πηγαίνοντας ευθεία και count\_turn στρίβοντας.

$$\text{count} = (\text{Distance}/0.2) * 20$$

$$\text{count\_turn} = (S/0.2) * 20$$

Όπου 0.2 η περίμετρος του τροχού σε μέτρα και 20 ο αριθμός των διάκενων των τροχών.

Τελικά αυτό που κάνει το πρόγραμμα είναι να στρίβει πρώτα το ρομπότ προς τη κατεύθυνση του συγκεκριμένου σημείου και έπειτα να συνεχίζει ευθεία μέχρι το ζητούμενο σημείο.

Για να εκτελεστεί το πρόγραμμα πάμε στο φάκελο Desktop από την γραμμή εντολών

```
$ cd Desktop/
```

και μετά πληκτρολογούμε

```
~/Desktop$ python rpm_move2.py
```

### **Εφαρμογή απλής οδομετρίας με το εικονικό ρομπότ.**

Στον φάκελο `catkin_ws/src` δημιουργήθηκε ένας άλλος υποφάκελος ο `move_controller` στον οποίο υπάρχει ένας υποφάκελος `src` που περιέχει το `move_controller.py` το οποίο είναι παρόμοιας φιλοσοφίας με το προηγούμενο αλλά εδώ ελέγχεται το εικονικό ρομπότ και με τις τιμές της ταχύτητας που παίρνει αυτό κινείται και τα πραγματικό ρομπότ. Το πρόγραμμα εδώ ζητάει την θέση  $X, \Psi$ , που θέλουμε να πάει το ρομπότ μα και υπολογίζει και πάλι την απόσταση και την γωνιά που πρέπει να στραφεί το ρομπότ. Εδώ η διαφορά όμως είναι ότι δεν πηγαίνει για μια φορά στη θέση όπου του δώσαμε και μετά τελειώνει το πρόγραμμα αλλά συνεχώς. Δηλαδή από τη στιγμή που θα φτάσει στο πρώτο σημείο που του θέσαμε μετά μας ζητάει ξανά μια καινούργια θέση και πηγαίνει στην καινούργια και ούτω καθεξής.

Εδώ η οδομετρία προέρχεται από το ίδιο το ROS η θέση και η κατεύθυνση του ρομπότ παρακολουθείται και υπάρχει στο topic `Odom`. Άρα κάθε στιγμή μπορούμε να τα γνωρίζουμε. Συνεπώς αφού με το που τελειώσει η προηγούμενη κίνηση γνωρίζουμε την θέση και τον προσανατολισμό του ρομπότ στο τέλος της μπορούμε με την εισαγωγή της νέας θέσης από τον χρήστη να υπολογίσουμε την νέα μετατόπιση και στροφή. Εδώ λοιπόν έχουμε την εγγραφή του node `speed_controller` που δημιουργούμε σ αυτό το script στο topic `/odom` το οποίο περιέχει μηνύματα οδομετρίας τα οποία περιέχουν την θέση και τον προσανατολισμό του ρομπότ.

```
sub = rospy.Subscriber("/odom", Odometry, newOdom)
```

Με τις εντολές  $x = \text{msg.pose.pose.position.x}$  και  $y = \text{msg.pose.pose.position.y}$  παίρνουμε κάθε φορά τις θέσεις  $X, Y$  του ρομπότ. Η γωνία όμως περιστροφής του είναι λίγο πιο δύσκολο να εξαχθεί κι αυτό γιατί εδώ οι γωνίες είναι εκφρασμένες σε τετραδόνια (quaternion)<sup>8</sup> οπότε αυτό που πρέπει πρώτα να γίνει είναι αν μετατραπούν αυτές οι γωνίες σε γωνίες Euler. Αυτό συμβαίνει με την συνάρτηση

```
rot_q = msg.pose.pose.orientation
```

παίρνουμε τις γωνίες σε Τετραδόνια

```
(roll ,pitch, theta) = euler_from_quaternion([rot_q.x, rot_q.y, rot_q.z, rot_q.w])
```

Μετατρέπουμε τις γωνίες από τετραδόνια σε γωνίες Euler<sup>9</sup> και η ζητούμενη γωνία είναι η γωνία γύρω από την  $Z$  άξονα  $\theta$ .

Αυτό που γίνεται στη συνέχεια είναι το πρόγραμμα ζητάει από τον χρήστη μια θέση και υπολογίζει την γωνία στροφής και την απόσταση που πρέπει να πάει ευθεία. Πρώτα στρίβει στην κατάλληλη γωνία και έπειτα πηγαίνει ευθεία μέχρι το ζητούμενο σημείο. Η διαφορά με το προηγούμενο πρόγραμμα είναι ότι τώρα το feedback προέρχεται από την οδομετρία του εικονικού ρομπότ κι όχι από τους αισθητήρες στους τροχούς του πραγματικού ρομπότ. Οι απόσταση που πρέπει να διανύσει το ρομπότ είναι κάθε φορά ή δοσμένη

```
X - x = msg.pose.pose.position.x
```

ομοίως και για το  $Y - y = \text{msg.pose.pose.position.y}$

αυτό υλοποιείται στο πρόγραμμα

```
dif_x = goal.x - x και
```

---

<sup>8</sup><https://el.wikipedia.org/wiki/%CE%A4%CE%B5%CF%84%CF%81%CE%B1%CE%B4%CF%8C%CE%BD%CE%B9%CE%BF>

<sup>9</sup><https://eclass.teicrete.gr/modules/document/file.php/TM168/%CE%A6%CE%91%CE%A3%CE%9F%CE%A5%CE%9B%CE%91%CE%A3%20%CE%94%CE%99%CE%91%CE%A6%CE%91%CE%9D%CE%95%CE%99%CE%95%CE%A3/2-%CE%A0%CE%95%CE%A1%CE%99%CE%A3%CE%A3%CE%9F%CE%A4%CE%95%CE%A1%CE%91%20%CE%93%CE%99%CE%91%20%CE%A4%CE%9F%CE%A5%CE%A3%20%CE%A0%CE%99%CE%9D%CE%91%CE%9A%CE%95%CE%A3%20%CE%A3%CE%A4%CE%A1%CE%9F%CE%A6%CE%97%CE%A3.ppt.pdf>



$dif\_y = goal.y - y$

Ενώ η απόσταση τώρα είναι

$Distance = (dif\_x^{**2} + dif\_y^{**2})$

Ομοίως και για την γωνία

$angle\_to\_goal = atan2(dif\_y, dif\_x)$  η γωνία στόχος

$angle\_to\_goal\_moires = angle\_to\_goal * 360.0 / 6.28$  μετατροπή της σε μοίρες

η αναζήτηση του σημείου σταματά όταν το τετράγωνο της απόστασης του ρομπότ από το σημείο στόχο είναι μικρότερη από 0.02. Η τιμές των ταχυτήτων εκφράζονται ως:

$speed.linear.x$  η τιμή της γραμμικής ταχύτητας στον X άξονα του ρομπότ.

$speed.angular.z$  η τιμή της γωνιακής ταχύτητας ως προς τον Z άξονα.

Η τιμές αυτές διαχέονται στο εικονικό αλλά και στο πραγματικό ρομπότ μέσω του topic `cmd_vel` στο οποίο κάνουμε Publish

```
pub = rospy.Publisher("/cmd_vel", Twist, queue_size=1)
```

### **Nodes στην υλοποίηση της πλατφόρμας**

A) Τρέχουν αποκλειστικά στο ρομπότ

1) node για να παίρνει πληροφορίες για τη κίνηση του ρομπότ και να τις μεταφράζει κατάλληλα για τον έλεγχο των κινητήρων.

2) node για να παίρνει πληροφορίες για τις θέσεις του ρομποτικού βραχίονα και να τις μεταφράζει κατάλληλα για την κίνηση των σερβοκινητήρων.

3) node για την λήψη βίντεο από την κάμερα .

4) node για την αποστολή των μετρήσεων απόστασης από το αποστασιόμετρο.

5) node για την αποστολή των δεδομένων από τους κωδικοποιητές των τροχών με στοιχεία οδομετρίας.

Β) Τρέχουν στο απομακρυσμένο υπολογιστή ή και στο ίδιο το ρομπότ εάν είναι συνδεδεμένο με οθόνη και πληκτρολόγιο.

1) node για την κίνηση του ρομπότ από το πληκτρολόγιο.

2) node για την κίνηση του ρομποτικού βραχίονα.

3) node για την παρακολούθηση του βίντεο που λαμβάνεται από την κάμερα του ρομπότ.

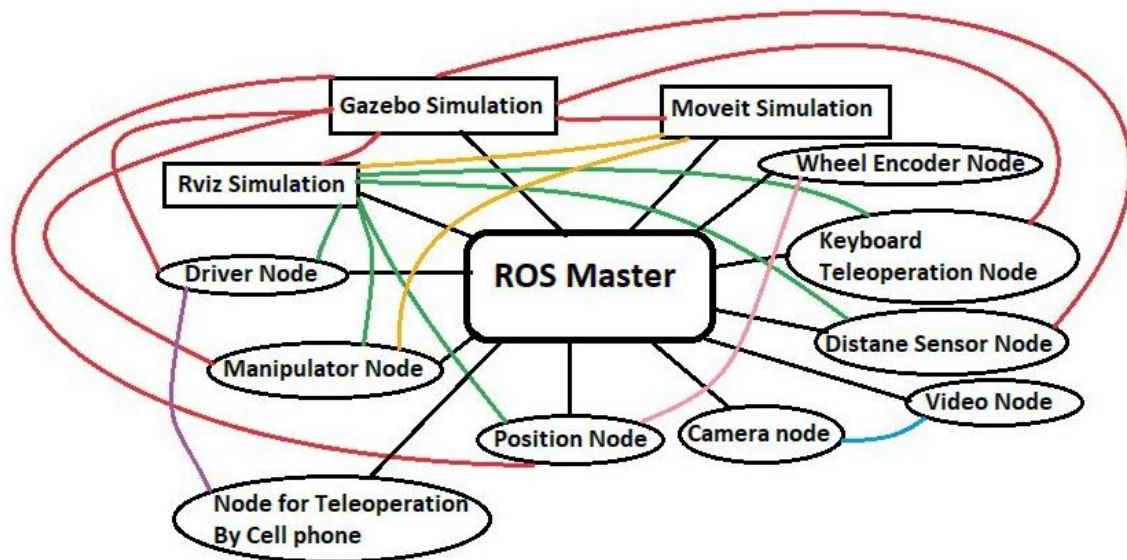
4) node για την κίνηση του ρομπότ σε καθοριζόμενη θέση πάνω στο επίπεδο χ,ψ.

Γ) Προσομοιώσεις τρέχουν αποκλειστικά στο απομακρυσμένο υπολογιστή.

1) προσομοίωση του ρομπότ σε εικονικό περιβάλλον μέσω της εφαρμογής Gazebo

2) προσομοίωση των κινήσεων και της οδομετρίας του ρομπότ σε εικονικό περιβάλλον μέσω της εφαρμογής Rviz

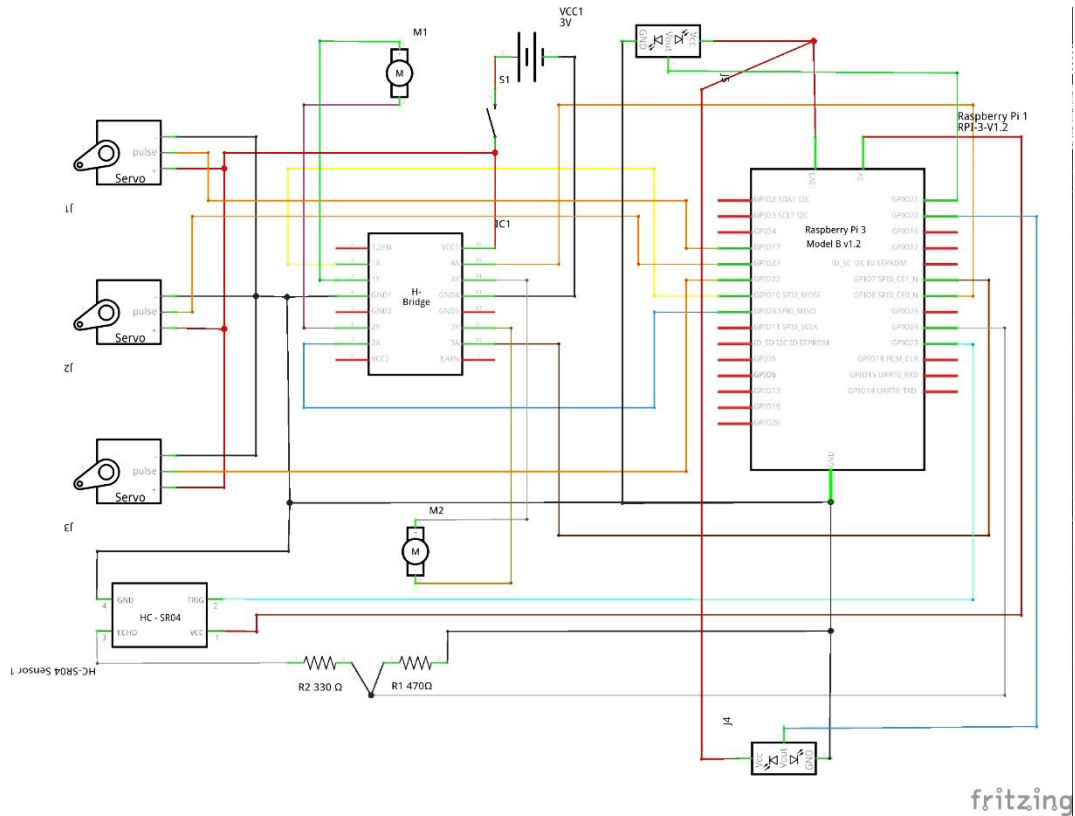
3) προσομοίωση και έλεγχος των κινήσεων του ρομποτικού βραχίονα μέσω της εφαρμογής του Moveit



Εικόνα 43 Σχεδιάγραμμα με τα Nodes και τις συνδέσεις τους

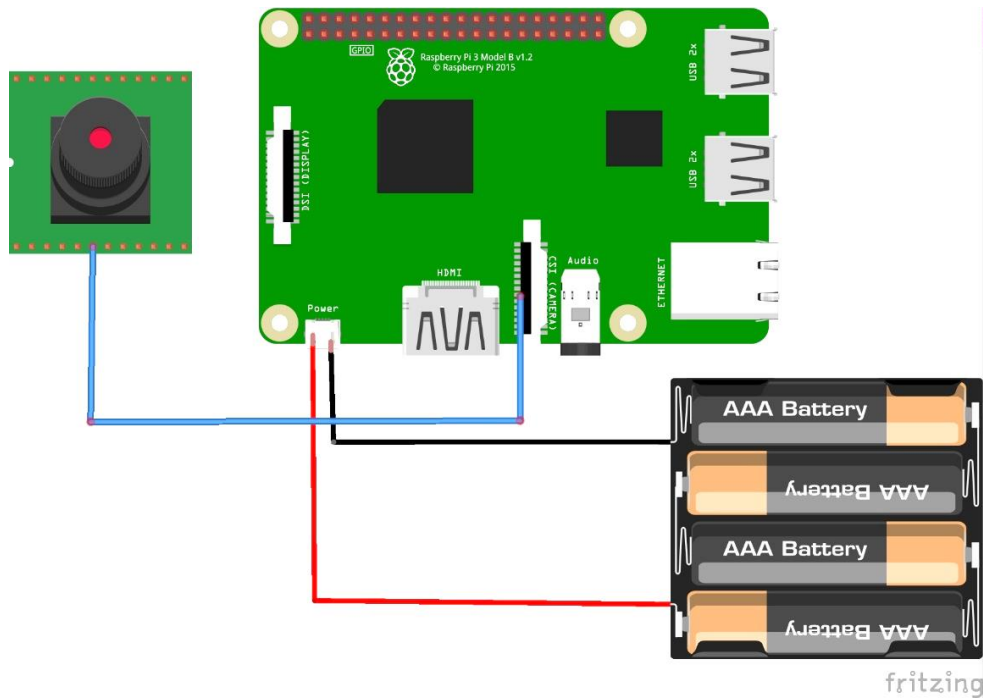
## Ηλεκτρονικό σχεδιάγραμμα του ρομπότ

Οι συνδέσεις που έγιναν με τα ηλεκτρονικά εξαρτήματα και τα ριπτού Raspberry φαίνονται στο παρακάτω σχεδιάγραμμα (εικόνα 46).



Εικόνα 44 ηλεκτρονικό σχεδιάγραμμα

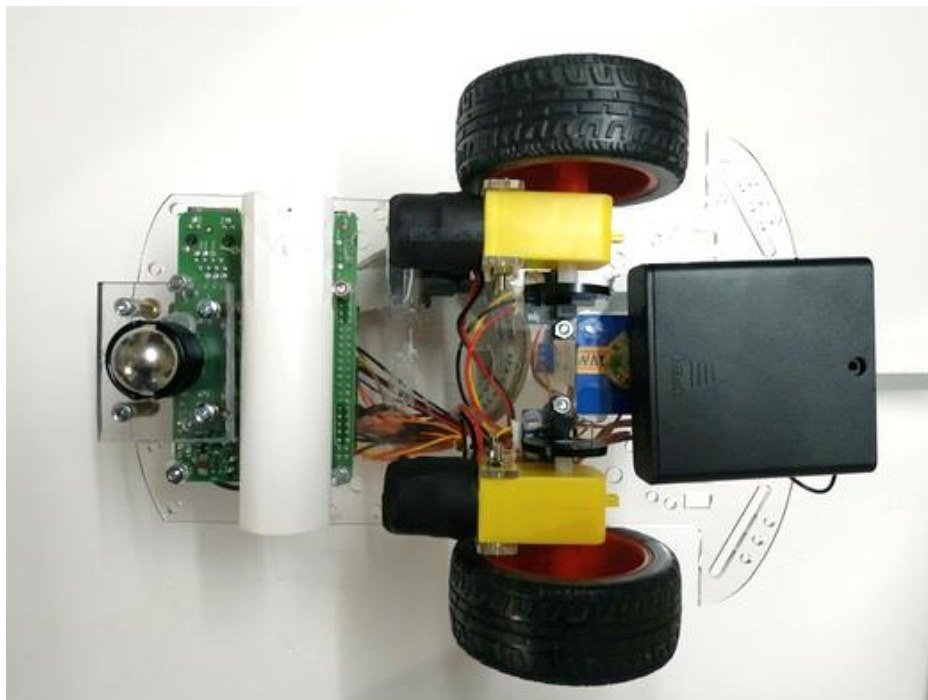
Οι μόνες συνδέσεις οι οποίες δεν φαίνονται εδώ είναι αυτή της κάμερας με την πλακέτα του Raspberry και της τροφοδοσίας του από μία επαναφορτιζόμενη μπαταρία 5volt. Οι συνδέσεις της κάμερας και της τροφοδοσίας φαίνονται στην παρακάτω εικόνα 47.



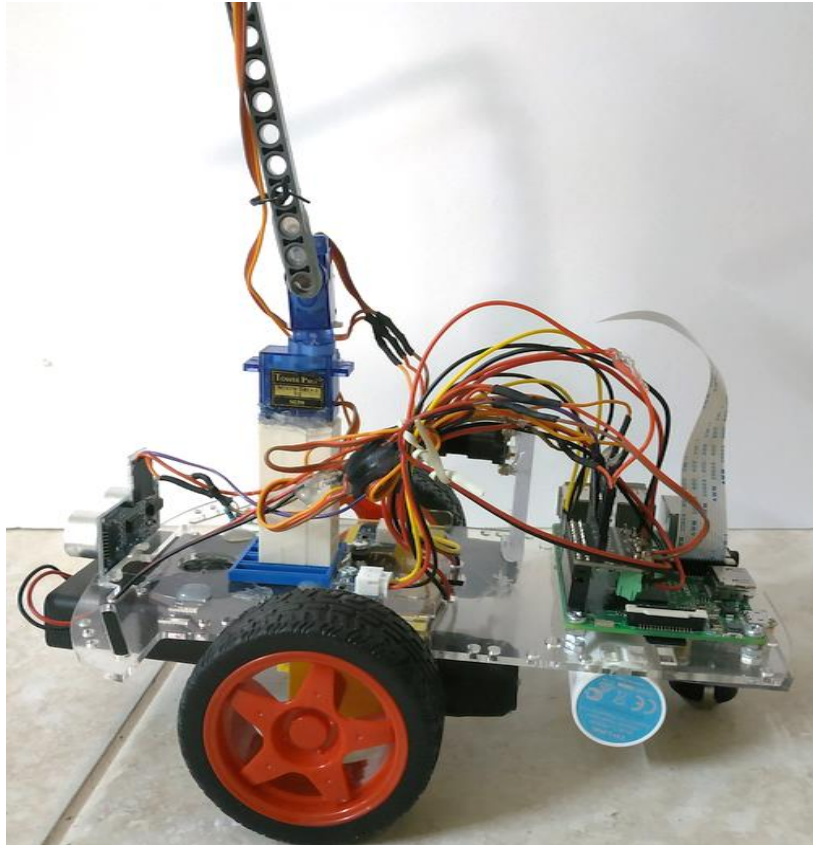
Εικόνα 45 σύνδεση κάμερα και τροφοδοσία

Στην κατασκευή δεν χρησιμοποιήθηκε breadboard αλλά όπου χρειάστηκε έγιναν κολλήσεις μεταξύ των καλωδίων κι ενώθηκαν όλα πάνω στην πλακέτα του Raspberry.

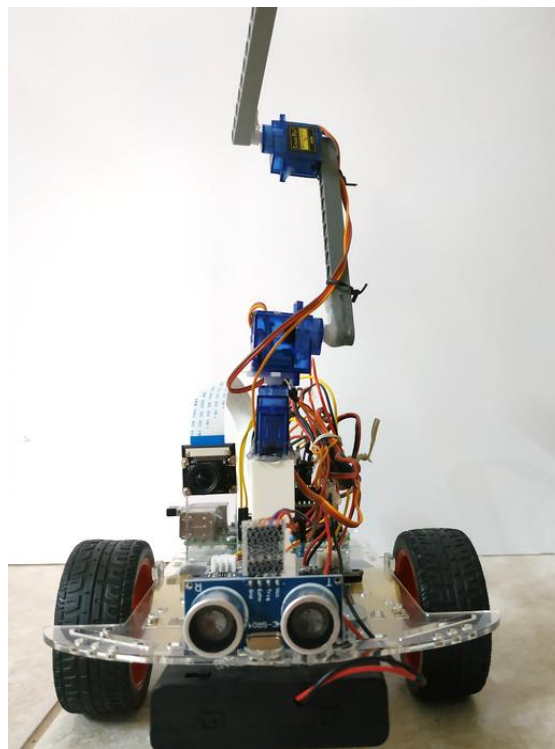
**Η τελική μορφή του ρομπότ.**



Εικόνα 46Κάτω όψη



Εικόνα 47Επάνω όψη.



Εικόνα 48Εμπρός όψη

## Κεφάλαιο 5<sup>ο</sup>

### Λειτουργία και αξιολόγηση του ρομπότ.

Αφού συνδέσουμε την επαναφορτιζόμενη μπαταρία στην MicroUsb θύρα τροφοδοσίας του Pi ανοίγουμε και την τροφοδοσία από το δεύτερο πακέτο μπαταριών για την τροφοδοσία των κινητήρων. Με το που ολοκληρωθεί η εκκίνηση του λειτουργικού συστήματος, το ρομπότ ανοίγει και το δίκτυο του, το οποίο έχει όνομα `ubiquityrobot`. Από αυτή τη στιγμή μπορούμε να συνδέσουμε τον υπολογιστή μας με το δίκτυο για να υπάρχει επικοινωνία μεταξύ του ρομπότ και του υπολογιστή. Για να συνδεθούμε με το δίκτυο ο κωδικός είναι `robotseverywhere`. Αφού συνδεθούμε με το δίκτυο μετά μπορούμε να συνδεθούμε και με το ρομπότ και συγκεκριμένα με την γραμμή εντολών του μέσω `ssh`.

Πληκτρολογούμε `ssh ubuntu@10.42.0.1` στη γραμμή εντολών του υπολογιστή μας. Με αυτόν τον τρόπο έχουμε αποκτήσει έλεγχο του ρομπότ κι από εδώ μπορούμε να ενεργοποιούμε και να απενεργοποιούμε διεργασίες.

Επόμενο βήμα είναι να ενεργοποιήσουμε τον `rigprio daemon` για να ενεργοποιηθεί η βιβλιοθήκη `rigprio` που χρησιμοποιούμε στον ελεγκτή των σερβοκινητήρων. Στο παράθυρο εντολών του ρομπότ πληκτρολογούμε

```
$ sudo rigpriad - θα μας ζητήσει κωδικό ο οποίος είναι ubuntu.
```

Επόμενο βήμα στην γραμμή εργασιών του ρομπότ είναι η απενεργοποίηση των ενεργειών βάσης που ξεκινούν μαζί με την ενεργοποίηση του ρομπότ και προϋπάρχουν από την εταιρία την οποία κατεβάσαμε το `image` πληκτρολογούμε:

```
$ sudo systemctl disable magni-base
```

κι εάν δεν το κάνουμε το ρομπότ μας λειτουργεί κανονικά.

Στην συνέχεια ανοίγουμε ένα νέο παράθυρο στο παράθυρο εντολών με `ctrl+shift+t` και εδώ ενεργοποιούμε τον ROS Master πληκτρολογούμε :

```
$ roscore
```

Από αυτή τη στιγμή ότι Node δημιουργείται επικοινωνεί με το ROS Master.

Ανοίγουμε νέο παράθυρο στο παράθυρο εντολών και συνδεόμαστε με το ρομπότ και πάλι και τώρα ανεβάζουμε το Node που ελέγχει τους κινητήριους τροχούς ώστε να υπακούει το ρομπότ. Πληκτρολογούμε:

```
$ rosrn kg_edukit_bot kg_driver_node_key
```

Από αυτή τη στιγμή μπορούμε να ελέγχουμε το ρομπότ με τρεις τρόπους .

α)πληκτρολογώντας σε νέο παράθυρο στην γραμμή εντολών του υπολογιστή μας

```
$ rosrn rqt_gui rqt_gui[37]
```

και ανοίγει ένα παράθυρο όπου μπορούμε να εισάγουμε τιμές ταχυτήτων στο κατάλληλο topic.

β)πληκτρολογώντας σε νέο παράθυρο στην γραμμή εντολών του υπολογιστή μας

```
$ rosrn teleop_twist_keyboard teleop_twist_keyboard.py
```

τόρα ελέγχουμε την κίνηση του ρομπότ με το πληκτρολόγιο του υπολογιστή μας τα πλήκτρα που χρησιμοποιούμε εμφανίζονται στην οθόνη.

γ) Από το android κινητό τηλέφωνο.

Συνδέουμε το τηλέφωνο με το δίκτυο του Pi και ανοίγουμε τη εφαρμογή ros control εκεί βρίσκουμε το ρομπότ μας και απλά πατάμε επάνω του. Για να συνδεθεί θα πρέπει να βάλουμε την διεύθυνση του RosMaster μας αυτή μπορούμε να τη δούμε στο παράθυρο που τρέχει το roscore ως: ROS\_MASTER\_URI=http://10.42.0.37:11311/ συνήθως πρέπει να αλλάξουμε εκεί που λέει localhost με τη διεύθυνση στην περίπτωση μας 10.42.0.37. Ακόμη πρέπει να επιλέξουμε και το Show advanced Options και να αλλάξουμε το όνομα του joystick Topic σε /cmd\_vel Το ρομπότ τώρα ελέγχεται από το κινητό μας με το joystick που έχει η εφαρμογή. Η εφαρμογή λειτουργεί και με tilt.

Για να ενεργοποιήσουμε την κάμερα ανοίγουμε νέο παράθυρο στο παράθυρο εντολών όπου συνδεόμαστε με το ρομπότ, πάλι και τώρα ανεβάζουμε το Node για την κάμερα πληκτρολογούμε :

```
$ roslaunch raspicam_node camerav2_1280x960.launch
```

και για να παρακολουθήσουμε το τι μεταδίδει η κάμερα ανοίγουμε νέο παράθυρο στην γραμμή εργασιών του υπολογιστή και πληκτρολογούμε :

```
$ rqt_image_view
```

βλέπουμε μόνο την εικόναή

```
$ rosrnrqt_guirqt_gui
```

όπου ανοίγει το γραφικό περιβάλλον ελέγχου όλων το topic όπου βλέπουμε και το τι μεταδίδει και η κάμερα.

Για να ελέγχουμε το βραχίονα ανοίγουμε ένα παράθυρο στη γραμμή εργασιών του υπολογιστή μας και πληκτρολογούμε:

```
$ roslaunch kg_edukit_bot kg_robot_arm_rviz.launch
```

αυτή η εφαρμογή μας οπτικοποιεί το ρομπότ μας κι ανοίγει κι ένα γραφικό μενού με όλες τις ενώσεις του ρομπότ.

Ανοίγουμε νέο παράθυρο στη γραμμή εργασιών του υπολογιστή μας και συνδεόμαστε με το ρομπότ κι εκεί πληκτρολογούμε :

```
$ rosrn kg_edukit_bot joint_state_controller1.py
```

και μετακινώντας από το γραφικό περιβάλλον τα joints βλέπουμε να κινείται και η προσομοίωση αλλά και ο πραγματικός βραχίονας.

Για να ελέγξουμε τον ρομποτικό βραχίονα στο πραγματικό αλλά και στο εικονικό ρομπότ και να του δίνουμε προκαθορισμένες θέσεις και όχι να τα πηγαίνουμε όπου θέλουμε χειροκίνητα, ανοίγουμε τρία νέα παράθυρα στην γραμμή εργασιών και πληκτρολογούμε στο πρώτο:

```
$ roslaunch kg_edukit_bot kg_robot_arm_xacro_gazebo.launch
```

αυτό το πρόγραμμα ξεκινά την προσομοίωση του ρομπότ μας στο εικονικό περιβάλλον φυσικής gazebo και ενεργοποιεί τους ελεγκτές των ενώσεων των σερβοκινητήρων.

Στο δεύτερο:



```
$ roslaunch kg_edukit_moveit demo_planning_execution.launch
```

Και στο παράθυρο όπου τρέξαμε το :

```
$ rosrn kg_edukit_bot joint_state_controller1.py
```

Το κλείνουμε με control+C

Και πληκτρολογούμε

```
$ rosrn kg_edukit_bot joint_state_controller2.py
```

Ο λόγος είναι ότι οι τιμές των θέσεων των joints τώρα τρέχουν σε διαφορετικό topic.

Το πρόγραμμα αυτό είναι η προσομοίωση του ρομποτικού βραχίονα για να εκτελούμε προκαθορισμένες κινήσεις. Τις κινήσεις τις βλέπουμε να τις εκτελεί και το εικονικό ρομπότ αλλά και το πραγματικό.

Σε αυτό το σημείο αφού δουλεύει η προσομοίωση του ρομπότ μπορούμε να δοκιμάσουμε και την τηλεκατεύθυνση που αναφέραμε πιο πάνω, και θα βλέπουμε το πραγματικό αλλά και το εικονικό ρομπότ να κινούνται.

Για να εφαρμόσουμε απλά προγράμματα οδομετρίας υπάρχουν όπως αναφέραμε στο προηγούμενο κεφάλαιο τρεις εφαρμογές.

Ανοίγουμε νέο παράθυρο στην γραμμή εργασιών του υπολογιστή μας και συνδεόμαστε με το ρομπότ και πάλι. Εκεί πρέπει να μπούμε στο φάκελο Desktop και να τρέξουμε τα προγράμματα rpm.py ή rpm\_move2.py πληκτρολογούμε :

```
$ cd Desktop/
```

```
$ python rpm.py
```

```
$ python rpm_move2.py.
```

Η τρίτη είναι σε ένα φάκελο στον απομακρυσμένο υπολογιστή όπου τρέχει κι ο RosMaster. Κλείνουμε τις προηγούμενες εφαρμογές οδομετρίας και ανοίγουμε νέο παράθυρο στην γραμμή εντολών του υπολογιστή μας και τρέχουμε πληκτρολογώντας:

```
$ rosrun move_controller.py
```

Εδώ θα βλέπουμε το εικονικό ρομπότ στο Gazebo, στο Rviz και το πραγματικό να κινούνται και να προσπαθούν να φτάσουν στην δοσμένη θέση.

Τελική εφαρμογή είναι αυτή της ενεργοποίησης του αισθητήρα υπερήχων και η αποστολή σημάτων στο Ros ανοίγουμε ένα νέο παράθυρο στην γραμμή εργασιών και αφού συνδεθούμε με το ρομπότ πληκτρολογούμε στη γραμμή εργασιών του:

```
$ rosrun kg_edukit_bot Ultrasound_Publisher
```

Εδώ μπορούμε να βλέπουμε τις τιμές της απόστασης που βλέπει μπροστά του ο αισθητήρας απόστασης αλλά μπορούμε να τις βλέπουμε και από το σύστημα του RoS στον απομακρυσμένο υπολογιστή πληκτρολογώντας σε νέο παράθυρο της γραμμής εντολών του υπολογιστή μας

```
$ rostopic echo /distance.
```

### **Αξιολόγηση της λειτουργίας της πλατφόρμας**

- 1) Η λειτουργία της είναι αξιόπιστη και επαναλήψιμη χωρίς προβλήματα σταθερότητας, αρκεί η απόσταση του ρομπότ που δημιουργεί το δίκτυο να μην είναι εκτός εμβέλειας του υπολογιστή.
- 2) Με τη μπαταρία που έχουμε για την λειτουργία του raspberry επιτρέπει την αδιάλειπτη λειτουργία του για 45 λεπτά περίπου. Οι μπαταρίες για τον έλεγχο των κινητήρων ο χρόνος λειτουργίας τους εξαρτάται από τον τρόπο χρήσεις του αλλά και σε συνεχή χρήση αντέχουν για μια ώρα ικανοποιητικά.
- 3) Η λειτουργία της, η οποία γίνεται βήμα-βήμα επιτρέπει τη κατανόηση όλων των κομματιών που την απαρτίζουν και την χρησιμοποίηση του καθενός ξεχωριστά. Εάν όλες οι διεργασίες τρέχανε ταυτόχρονα θα ήταν πιο δύσκολο να αποσυνδεθούν και να εξηγηθούν.
- 4) Όλα τα κομμάτια του κώδικα είναι ανοικτού λογισμικού και η άδεια χρήσης τους είναι ελεύθερη, που σημαίνει ότι οι τροποποιήσεις είναι ευπρόσδεκτες και οι πειραματισμοί. Με σκοπό την μάθηση ο καθηγητής έχει ένα εργαλείο στα χέρια του με πολλές δυνατότητες.

5) Το κόστος του είναι κάτω από 120 ευρώ και τα κομμάτια που το αποτελούν είναι πολύ εύκολα και οικονομικά, να βρεθούν σε περίπτωση αστοχιών και δυσλειτουργιών.

6) Η εργασία αυτή έχει ως στόχο την εισαγωγή στη ρομποτική σε πραγματικές συνθήκες και η χρήση των ρομπότ που θα κατασκευάσουν οι μαθητές για να ανακαλύψουν τις έννοιες μαθηματικών, φυσικής, της μηχανικής, και της πληροφορικής με τη μέθοδο STEM. Για αυτό έχουν δημιουργηθεί κώδικες οι οποίοι είναι εντελώς ανοικτοί για πειραματισμό και επέκτασή τους. Οι προσεγγίσεις της σχολικής ρομποτικής μέχρι σήμερα επικεντρώνονται κυρίως στο κομμάτι του γραπτού κώδικα για τον έλεγχο των ρομπότ ή και της κατασκευής Lego[38], αλλά δεν μπαίνουν βαθύτερα στο Hardware. Όταν συνδέουν έναν αισθητήρα ή έναν κινητήρα με τον εγκέφαλο του ρομπότ οι μαθητές, δεν καταλαβαίνουν την πολυπλοκότητα που κρύβεται από πίσω. Είναι σαν κάποιος που γνωρίζει τον κώδικα που γράφει για παράδειγμα σε C και δεν ξέρει τη σημαίνει δυαδική λογική. Στα Lego, για παράδειγμα, τα καλώδια που χρησιμοποιούνται είναι τετραπλά, αλλά οι μαθητές δεν το γνωρίζουν και αν ακόμη το ξέρουν δεν γνωρίζουν τη χρήση τους. Μέσα από αυτά διέρχονται τα ρεύματα ελέγχου του κινητήρα και οι πληροφορίες από τους κωδικοποιητές τροχών. Πώς όλα αυτά μεταφράζονται από τάσεις και ρεύματα σε πληροφορίες 0 και 1 δεν έχουν ιδέα. Αυτή η πλατφόρμα φιλοδοξεί να δώσει αυτές τις έννοιες στους μαθητές, αφού ο απώτερος στόχος της είναι να προετοιμάσει τους αυριανούς επιστήμονες για την ρομποτική.

## **Προτάσεις δίωρων Μαθημάτων.**

### **1)Εισαγωγικό μάθημα**

Στο εισαγωγικό μάθημα ξεκινάμε με την εξήγηση του τι είναι ρομποτική. Στη συνέχεια παρουσιάζουμε τους αισθητήρες που χρησιμοποιούμε ,και περιγράφουμε την λειτουργία των κινητήρων των απλών αλλά και των σερβοκινητήρων που φέρει το ρομπότ. Αρχικά, κάνουμε αναφορά στην τάση, το ρεύμα και τη λειτουργία των κινητήρων DC και όλα αυτά συνδέονται με τη γνώση των παλαιότερων τάξεων. Το επόμενο βήμα είναι η εξήγηση της λειτουργίας των σερβοκινητήρων και τι είναι το PWM (Pulse Width Modulation). Στη συνέχεια επεξηγήθηκε η έννοια του αισθητήρα και εφαρμόζουμε την εφαρμογή του αισθητήρα απόστασης, όπου οι μαθητές χρησιμοποιούν το χρόνο πτήσης για τον υπολογισμό της απόστασης. Αυτό το μάθημα

σχετίζεται άμεσα με τη φυσική της πρώτης τάξης. Η κατανόηση από τους μαθητές της λειτουργίας του συγκεκριμένου αισθητήρα είναι απαραίτητη επειδή είναι βασική για την κατανόηση άλλων πιο περίπλοκων αισθητήρων που χρησιμοποιούνται στη ρομποτική. Ένας άλλος αισθητήρας που χρησιμοποιεί το ρομπότ είναι ο Photo Interrupter, ένας υπέρυθρος αισθητήρας όπου εξηγείτε κι αυτός πώς λειτουργεί και τέλος αναλύεται η λειτουργία της κάμερας. Η οποία αποτελείται από πολλούς αισθητήρες φωτός.

## **2)Παράδοση του Kit στους μαθητές**

Στο δεύτερο μάθημα αφού αναλύθηκαν, τα βασικά συστατικά του ρομπότ παραδίδονται τα kit των ρομπότ και κατασκευάζουν οι μαθητές τα ρομπότ τους. Έπειτα παρουσιάζεται η εικονική μορφή του ρομπότ σε περιβάλλον Rviz και Gazebo. Οι μαθητές παίρνουν επίσης το urdf αρχείο / που περιγράφει το ρομπότ. Το αρχείο αναλύεται και εξηγείται στους μαθητές για να κατανοήσουν τη βασική δομή του. Στη συνέχεια οι ίδιοι οι μαθητές παροτρύνονται να πειραματιστούν με αλλαγές στο υπάρχον αρχείο με προσοχή αποθηκεύοντας το τροποποιημένο αρχείο τους με νέο όνομα για να μην χάσουν το πρωτότυπο. Μ αυτό τον τρόπο βλέπουν πώς επηρεάζουν το ρομπότ στην κίνηση του, σε ένα εικονικό περιβάλλον οι αλλαγές σε τιμές του βάρους της ροπής αδράνειας των τριβών κτλ. Σε αυτό το σημείο, δεν γίνεται αναφορά στο σύστημα αρχείων ROS, καθώς είναι λίγο περίπλοκο, και θα γίνει αργότερα.

## **3)Εισαγωγή στο ROS**

Στο τρίτο μάθημα γίνεται εξήγηση της λειτουργίας του Ros, δηλαδή εξηγείται η βασική δομή επικοινωνίας του RosMaster με τα Nodes και τα topics. Έπειτα τους δείχνουμε το αρχείο Node που λαμβάνει τα μηνύματα από το πληκτρολόγιο του υπολογιστή και τα μεταφράζει σε εντολές κίνησης ρομπότ. Εδώ στους μαθητές γίνεται μια σύντομη επανάληψη των γνώσεων από την κυκλική κίνηση και επανεξετάζουν πρακτικές έννοιες όπως η γωνιακή ταχύτητα και η γραμμική ταχύτητα για εφαρμογή σε πραγματικές συνθήκες. Σε αυτό το σημείο οι μαθητές εφαρμόζουν για πρώτη φορά τις γνώσεις τους από τη φυσική και τα μαθηματικά όπου μέχρι στιγμής δεν τα χρησιμοποιούσαν σε πρακτικά θέματα παρά μόνο σε λύση θεωρητικών ασκήσεων.

#### **4)Απλή οδομετρία**

Το τέταρτο μάθημα είναι η κατεύθυνση του ρομπότ σε μια συγκεκριμένη θέση. Εδώ στους μαθητές γίνεται μια επανάληψη του Πυθαγορείου Θεωρήματος και των βασικών τριγωνομετρικών εξισώσεων. Και πάλι, το αρχείο που εφαρμόζει αυτές τις αρχές των μαθηματικών είναι παραδίδεται στους μαθητές έτοιμο έτσι ώστε το ρομπότ να πάει σε αυτή μια συγκεκριμένη θέση. Σε αυτό το σημείο, να σημειωθεί ότι οι μαθητές μπορούν να παρακολουθούν τις κινήσεις του εικονικού αλλά και του πραγματικού ρομπότ.

#### **5)Κινηματική του βραχίονα**

Στο πέμπτο μάθημα, οι μαθητές ασχολούνται με το ρομποτικό βραχίονα του ρομπότ. Εδώ γίνεται εφαρμογή γεωμετρίας και τριγωνομετρίας για να βρεθεί η τελική θέση του άκρου του βραχίονα. Επίσης εδώ σημαντικό είναι να γίνει μια αναφορά στη ροπή και να εξηγηθεί αυτό το τεχνικό χαρακτηριστικό των σερβοκινητήρων. Μετά από αυτό, πραγματοποιείται μια απλή κίνηση του βραχίονα, αφού τους δοθεί το αρχείο ελέγχου του βραχίονα το οποίο αναλύεται κι εξηγείται για να κατανοήσουν τη χρήση του. Εδώ οι μαθητές δίνουν μέσα από το γραφικό περιβάλλον ελέγχου των ενώσεων του ρομποτικού βραχίονα κάποιες γωνίες για τις ενώσεις και προσπάθησαν να βρουν τη θέση όπου θα καταλήξει το άκρο του ρομποτικού βραχίονα. Το επόμενο βήμα σε αυτό το μάθημα είναι να προσπαθήσουμε να κάνουμε την αντίστροφη εργασία από πριν. Δηλαδή να υπολογίσουμε τις γωνίες που πρέπει να στρέψουν οι σερβοκινητήρες των ρομποτικών βραχιόνων για να μεταβεί το άκρο του σε μια συγκεκριμένη θέση. Εδώ οι μαθητές θα καταλάβουν τη δυσκολία του αντίστροφου κινηματικού προβλήματος. Η λύση τέτοιων εξισώσεων δεν είναι στους στόχους αυτού του μαθήματος, αλλά γίνεται για την κατανόηση της δυσκολίας του εγχειρήματος. Μετά την κατανόηση της δυσκολίας του προβλήματος, παρουσιάζεται το πακέτο Ros Moveit όπου ο βραχίονας μπορεί να εκτελεί πολύπλοκες κινήσεις χρησιμοποιώντας έτοιμες βιβλιοθήκες με κινηματικές λύσεις. Εδώ τώρα οι μαθητές βλέπουν το εικονικό και το πραγματικό ρομπότ να κάνει το ίδιο πράγμα και να πηγαίνει από μία συγκεκριμένη θέση σε μία άλλη.

#### **6)Τηλεκατεύθυνση του ρομπότ μέσω της κάμερας**

Ένα τελευταίο μάθημα μπορεί να είναι αυτό του τηλεχειρισμού του ρομπότ μέσα από την εικόνα που μεταδίδεται από την κάμερα που υπάρχει στο ρομπότ και ανάγνωση των αποστάσεων που μεταδίδει το Node του μετρητή απόστασης. Εδώ δίνεται και εξηγείται και πάλι το αρχείο με το πρόγραμμα του μετρητή απόστασης. Οι μαθητές παρακολουθούν ζωντανό βίντεο από την κάμερα του ρομπότ στον υπολογιστή τους και μέσω αυτής της εικόνας μπορούν να τηλεκατευθύνουν το ρομπότ τους. Έχοντας τις τιμές των αποστάσεων από το μετρητή απόστασης μπορούν να πλησιάσουν αντικείμενα και με το ρομποτικό βραχίονα να τα ακουμπήσουν προσομοιώνοντας διάφορες κινήσεις.

## **Κεφάλαιο 6<sup>ο</sup>**

### **Συμβολή της μεταπτυχιακής εργασίας**

Τα κύρια σημεία συμβολής της παρούσας μεταπτυχιακής εργασίας περιγράφονται παρακάτω:

Αναπτύχθηκε ένα αρκετά οικονομικό ρομπότ με αρκετές δυνατότητες για το κόστος κατασκευής του και ακόμη περισσότερες δυνατότητες επέκτασης των εφαρμογών του, χωρίς απαραίτητα την προσθήκη περαιτέρω υλικού.

Δημιουργήθηκε λογισμικό για την πλοήγησή του, τον έλεγχο των σερβοκινητήρων και την συλλογή δεδομένων από τον αισθητήρα απόστασης και από τους οπτικούς κωδικοποιητές των τροχών με σκοπό τον έλεγχο της οδομετρίας του.

Δημιουργήθηκε προσομοίωση του σε εικονικά περιβάλλοντα όπου δεν απαιτείται η παρουσία του φυσικού ρομπότ για την λειτουργία του.

Σχεδιάστηκαν έξι προτάσεις μαθημάτων με σκοπό την εισαγωγή μαθηματικών και φυσικών εννοιών, με έμφαση στην STEM προσέγγιση.

Το παρόν ρομπότ ελπίζω να αποτελέσει εφαλτήριο για περαιτέρω μελέτη και βελτίωσή των εκπαιδευτικών ρομπότ.

### **Συμπεράσματα της εργασίας**

Οι δυνατότητες αυτού του συγκεκριμένου πακέτου είναι πάρα πολλές, και ο λόγος είναι το ROS, το οποίο έχει ως κύριο λειτουργικό σύστημα δημιουργίας και ανάπτυξης του. Αυτή είναι και η μεγάλη διαφορά του με τις υπάρχουσες εκπαιδευτικές ρομποτικές πλατφόρμες.

Το ROS έχει έτοιμες βιβλιοθήκες και εφαρμογές που υποστηρίζονται από την ανοιχτή κοινότητα ROS και συνεχώς επεκτείνονται. Δίνοντας στον χρήστη την δυνατότητα να επεκτείνει τις εφαρμογές του .

Επιπλέον η πλακέτα του Raspberry έχει μια πολύ μεγάλη κοινότητα με πολυάριθμες εφαρμογές και απεριόριστα παραδείγματα ανάπτυξης τους.

Μπορεί να εφαρμοστεί άμεσα στην εκπαιδευτική διαδικασία, χωρίς πολλές απαιτήσεις στην εκπαίδευση των καθηγητών που θα το χρησιμοποιήσουν. Παράλληλα μπορεί να αναπτυχθεί ένας ιστότοπος, ο οποίος θα περιέχει βίντεο με παρουσιάσεις της κατασκευής και των μαθημάτων που μπορούν να υλοποιηθούν. Επίσης θα υπάρχει δυνατότητα αποστολής μηνυμάτων για τυχόν απορίες και προβλήματα κατά τη διαδικασία συναρμολόγησης του και λειτουργίας του.

Το κόστος του περιορίζεται κάτω από τα 120 ευρώ και το υλικό που το αποτελούν είναι πολύ εύκολο να βρεθούν και να αντικατασταθούν εάν κάτι χαλάσει.

Το λογισμικό όλο είναι ελεύθερο και χωρίς κόστος για τον χρήστη με αποτέλεσμα τη δυνατότητα ενασχόλησης των χρηστών ακόμη κι από το σπίτι τους με το εικονικό ρομπότ.

Είναι το μόνο σ αυτή την κατηγορία τιμής με προσομοίωση σε εικονικό περιβάλλον.

Γενικά παρατηρώντας τις δυνατότητες της συγκεκριμένης πλατφόρμας που προτείνουμε, σε σχέση με το τι υπάρχει στην αγορά και συγκρίνοντας τις τιμές αγοράς τους η πρότασή μας είναι σε πολύ πλεονεκτική θέση.

### **Μελλοντική επέκταση της δουλειάς**

Περαιτέρω βελτίωση στη παρούσα υλοποίηση θα ήταν η πλήρη ενσωμάτωση με την οδομετρία του Ros και η χρησιμοποίησή του για εύρεσή της διαδρομής μέσα από μια διαδρομή με εμπόδια.

Σε περίπτωση προσθήκης και ενός lidar θα μπορούσαμε να χρησιμοποιήσουμε και μεθόδους SLAM. Αλλά για την κατηγορία στην οποία αναφέρεται το ρομπότ μας , δηλαδή τη δευτεροβάθμια εκπαίδευση, εφαρμογές τέτοιου τύπου κρίνονται αρκετά απαιτητικές για την ηλικία των παιδιών. Τέτοιου είδους εφαρμογές βέβαια μπορούν να εφαρμοστούν σε προπτυχιακό επίπεδο σε σχολές πληροφορικής, μηχανικών και γενικά μαθημάτων στην τριτοβάθμια εκπαίδευση που ασχολούνται με την ρομποτική.

Μία άλλη προσθήκη η οποία απλά χρειάζεται το κατάλληλο λογισμικό για να λειτουργήσει είναι αυτή της αναγνώρισης εικόνας. Το MIT έχει αναπτύξει ένα παρόμοιο ρομπότ το οποίο από πλευράς υλικού, είναι ίδιο με το δικό μας. Το λογισμικό που αναπτύσσει κι είναι ελεύθερο για να το χρησιμοποιήσουμε, είναι αλγόριθμοι τεχνητής νοημοσύνης για αυτόνομη οδήγηση. Ακόμη κι αν δεν επεκταθούμε τόσο πολύ, πιο απλές εφαρμογές αναγνώρισης εικόνας χρησιμοποιώντας OPEN CV μπορούν να εφαρμοστούν απλά και μόνο με την προσθήκη επιπλέον λογισμικού. Από πλευράς hardware δεν χρειάζεται κάτι επιπλέον.

Μια περαιτέρω βελτίωση από πλευράς υλικού, θα ήταν αυτή της δημιουργίας ενός πιο στιβαρού ρομποτικού βραχίονα με άλλους σερβοκινητήρες και η προσθήκη μίας αρπάγης(gripper) στο άκρο του, το οποίο θα βελτιώνει την χρηστικότητά του και τις εφαρμογές του.

Ανάπτυξη λογισμικού για την αυτόνομη κίνησή του μέσα σε άγνωστο περιβάλλον. Θα μπορούσε να γραφτεί ένα απλό πρόγραμμα για απλή αποφυγή εμποδίων και εάν προθέτονταν και lidar να είχαμε αυτόνομη πλοήγησή του.

Επιπλέον, το υπάρχον ρομπότ μπορεί να χρησιμοποιηθεί άμεσα και σε μικρότερες ηλικίες. Εάν είναι προσυναρμολογημένο , αντί του image στην κάρτα μνήμης του με το ROS και το Ubuntu, μπορεί να χρησιμοποιηθεί ένα image με το rasbian το οποίο υποστηρίζει οπτικό προγραμματισμό μέσω Scratch. Η μόνη προσθήκη που θα απαιτούταν για να μην ξαναγράψουμε την SD κάρτα θα ήταν μια άλλη κάρτα με προ-εγκατεστημένο το Raspbian εγκατεστημένη με την τελευταία έκδοση του Scratch .



## Βιβλιογραφία

1. SLAM for Dummies A Tutorial Approach to Simultaneous Localization and Mapping By the ‘dummies’ Søren Riisgaard and Morten Rufus Blas
2. Introduction to Mobile Robotics SLAM: Simultaneous Localization and Mapping Wolfram Burgard, Cyrill Stachniss, Kai Arras, Maren Bennewitz
3. Programming Robots with ROS by Morgan Quigley, Brian Gerkey, and William D. Smart Copyright © 2015 Morgan Quigley, Brian Gerkey, and William D. Smart.
4. Learn Robotics Programming Build and control autonomous robots using Raspberry Pi 3 and Python Danny Staple BIRMINGHAM -2018 Packt Publishing
5. Computational Thinking for Teacher Education Aman Yadav, Chris Stephenson, and Hai Hong Communications of the ACM April 2017 | vol. 60 | no. 4
6. Introduction to ROS (Robot Operating System) with Arduino - Arjun Bhasin (Mentor Robotics Club IIT-k) Ros.org
7. ROS Robotics Projects Build a variety of awesome robots that can see, sense, move, and do a lot more using the powerful Robot Operating System Lentin Joseph BIRMINGHAM -Packt Publishing
8. Manshi Shukla, Amar Nath Shukla, “Growth of Robotics Industry Early in 21st Century”, International Journal Of Computational Engineering Research (ijceronline.com), Vol. 2 Issue 5, September 2012.
9. Luigi Pagliarini, Henrik Hautop Lund, “The future of Robotics Technology”, International Conference on Artificial Life and Robotics (ICAROB 2017), 2017.
10. D. Alimisis, M. Moro, “Special issue on educational robotics”, Robotics and Autonomous Systems, vol. 77, 2016, pp. 74-75.
11. B. S. Barker, G. Nugent, N. Grandgenett, & V. I. Adamchuk, “Robots in K-12 education: a new technology for learning,” IGI Global, 2012, pp. 1-402.
12. F. B. V. Benitti, “Exploring the educational potential of robotics in schools: A systematic review,” Computers & Education, vol.58, issue 3, 2012, pp. 978-988.

13. Y. Kim and P. Namje, “The effect of STEAM education on elementary school student’s creativity improvement”, *Computer Applications for Security, Control and System Engineering*, Springer, Berlin Heidelberg, 2012, pp. 115-121.
14. Dr. Mack Shelley & Dr. S. Ahmet Kiray, “Integrated STEM Education: Promoting STEM Literacy and 21st Century Learning” *Research Highlights in STEM Education*, ISRES Publishing, 2018.
15. Arpad Takacs, Gyorgy Eigner, Levente Kovacs, Imre J.Rudas and Tamas Haidegger, “Teachers’Kit Development, Usability and Communities of Modular Robotic Kits For Classroom Education”, *IEEE Robotics & Automation Magazine*, Vol 23, No.2, June 2016.
16. M. Naya, G. Varela, L. Llamas, M. Bautista, J. C. Becerra, F. Bellas, A. Prieto, A. Deibe, R. Duro, “A Versatile Robotic Platform for Educational Interaction”, *9th International Conference on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications*, Bucharest Romania, 21-13 September, 2017.
17. S. Atmatzidou and S. Demetriadis, “Advancing students’ computational thinking skills through educational robotics: A study on age and gender relevant differences,” *Robot. Auton. Syst.*, vol. 75, pp. 661–670, Jan. 2016.
18. R. B.-B. Levy and M. M. Ben-Ari, “Robotics activities—Is the investment worthwhile?” in *Informatics in Schools. Curricula, Competences, and Competitions*, vol. 9378. Cham, Switzerland: Springer, 2015, pp. 22–31.
19. Cecily Heiner, “A Robotics Experience for All the Students in an Elementary School”, *SIGCSE ’18: 49th ACM Technical Symposium on Computer Science Education*, Baltimore, MD, USA, Feb. 21–24, 2018.
20. Shakir Hussain, Jorgen Lindh, and Ghazi Shukur. “The effect of lego training on pupils’ school performance in mathematics, problem solving ability and attitude: Swedish data”, *Educational Technology & Society*, 9(3), 2006, pp. 182–194.
21. Jorgen Lindh and Thomas Holgersson, “Does lego training stimulate pupils ability to solve logical problems?” *Computers & Education*, 49(4), 2007, pp. 1097–1111.
22. Mohammad Ehsanul Karim, Severin Lemaignan, Francesco Mondada, “A review: Can robots reshape K-12 STEM education?”, *2015 IEEE International*

Workshop on Advanced Robotics and its Social Impacts, Lyon, France, 30 June-2 July 2015.

23. Francesco Mondada, Michael Bonani, Fanny Riedo, Manon Briod, Léa Pereyre, Philippe Rétornaz, and Stéphane Magnenat “Bringing Robotics to Formal Education The Thymio Open-Source Hardware Robot”, IEEE Robotics and Automation Magazine, March 2017.
24. S. Kradolfer, S. Dubois, F. Riedo, F. Mondada, F. Fassa, “A sociological contribution to understanding the use of robots in schools: The Thymio robot”, Social Robotics, Cham, Switzerland:Springer, 2014, pp. 217-228.
25. Affordable Mobile Robotic Platforms for Teaching Computer Science at African Universities Ernest B.B. Gyebi, Marc Hanheide and Grzegorz Cielniak School of Computer Science University of Lincoln, UK
26. M. Quigley, K. Conley, B. Gerkey, J. Faust, T. B. Foote, J. Leibs, R. Wheeler, and A.Y. Ng, “ROS: An open-source robot operating system,” Proc. ICRA Open-Source Softw. Workshop, 2009.
27. <https://downloads.ubiquityrobotics.com/>
28. List of Linux adopters, [https://en.wikipedia.org/wiki/List\\_of\\_Linux\\_adopters](https://en.wikipedia.org/wiki/List_of_Linux_adopters)
29. <http://gazebosim.org/>
30. DRV8833 Dual H-Bridge Motor Driver,  
<http://www.ti.com/lit/ds/symlink/drv8833.pdf>
31. Raspberry Pi Camera User Manual,  
<https://www.waveshare.com/w/upload/6/61/RPi-Camera-User-Manual.pdf>
32. Feetech FS90,  
<https://www.tme.eu/gb/Document/110379ac6163e3201b107a9051363338/FS90.pdf>
33. Photo Interrupter Sensor, Speed Measuring,  
<https://static.chipdip.ru/lib/663/DOC001663882.pdf>
34. Ultrasonic Ranging Module HC - SR04,  
<https://cdn.sparkfun.com/datasheets/Sensors/Proximity/HCSR04.pdf>
35. ROS node for camera module of Raspberry Pi,  
[https://github.com/UbiquityRobotics/raspicam\\_node](https://github.com/UbiquityRobotics/raspicam_node)
36. teleop\_twist\_keyboard, Package summary,  
[http://wiki.ros.org/teleop\\_twist\\_keyboard](http://wiki.ros.org/teleop_twist_keyboard)
37. rqt\_rviz, Package summary, [http://wiki.ros.org/rqt\\_rviz](http://wiki.ros.org/rqt_rviz)

38. Lego EV3, <https://www.lego.com/en-us/mindstorms/about-ev3>
39. Rviz - 3D visualization tool for ROS, <http://wiki.ros.org/rviz>
40. Ioan A. Sutan and Sachin Chitta, "MoveIt!", [Online] Available:  
<http://moveit.ros.org>
41. Ρομποτική Ι Κινηματική/Στατική/Δυναμική Ανάλυση και  
Έλεγχος Ρομποτικών Χειριστών Κων/νος Τζαφέστας Τομέας Σημάτων,  
Ελέγχου & Ρομποτικής Σχολή Ηλεκτρ. Μηχ/κών & Μηχ/κών Υπολ., Ε.Μ.Π.
42. <http://wiki.ros.org/turtlebot/tutorials/AndroidControl>

## Παράρτημα

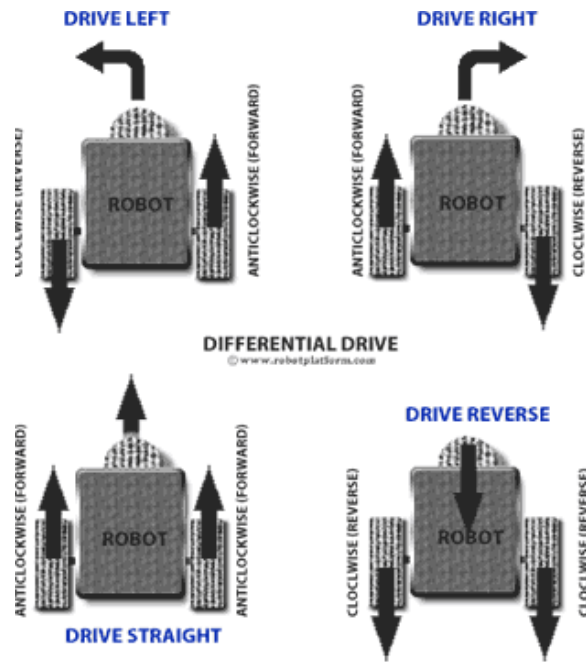
### Κινηματική Ανάλυση Ρομπότ Διαφορικής Οδήγησης

Όχημα διαφορικής οδήγησης, ονομάζεται κάθε όχημα με δύο τροχούς, μη κατευθυντήριους, στον ίδιο άξονα, όπου η κίνηση του καθενός από αυτούς είναι ανεξάρτητη και η φορά κίνησής των δύναται να είναι και η ωρολογιακή και η αντίστροφη αυτής. Από τη στιγμή που οι τροχοί των ρομπότ διαφορικής οδήγησης δεν είναι κατευθυντήριοι, η κατευθυντικότητα τους είναι  $\delta s = 0$  ενώ η κινητικότητα τους είναι  $\delta m = 2$  μιας και η κίνησή τους γίνεται μόνο στην επιφάνεια που ορίζεται από τους άξονες  $x, y$ . Οι βαθμοί ελευθερίας  $\text{dof}_{\text{dd}}$  των ρομπότ διαφορικής οδήγησης είναι άρα :

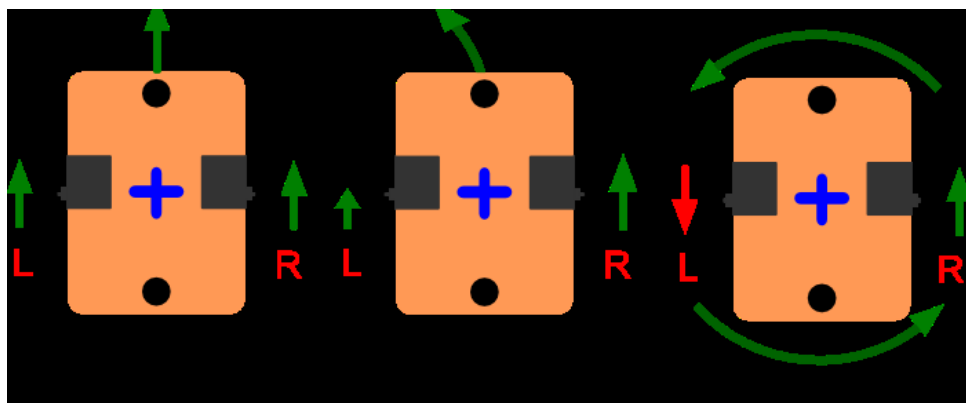
$$\text{dof}_{\text{dd}} = \delta m + \delta s = 2 + 0 = 2$$

Θα πρέπει να τονιστεί εδώ πως αν και η κίνηση γίνεται στην επιφάνεια που ορίζεται από τους άξονες  $x, y$ , παρόλα αυτά, το διάνυσμα κίνησης του ρομπότ δεν μπορεί να είναι παράλληλο με τον άξονα που ορίζουν οι 2 τροχοί συνεπώς δεν έχουμε ταχύτητα  $\sigma$  αυτόν των άξονα. Στην παρούσα υλοποίηση θεωρείται αυτός ο άξονας παράλληλος με τον  $\psi^R$ . Για να κινηθεί το όχημα ευθεία, θα πρέπει να περιστραφούν και οι δύο τροχοί του με την ίδια ταχύτητα. Για να στρίψει δεξιά θα πρέπει να περιστραφεί ο αριστερός τροχός με μεγαλύτερη ταχύτητα από τον δεξιό. Για να στρίψει ακαριαία δεξιά θα πρέπει ο δεξιός τροχός να περιστραφεί προς τα πίσω και ο αριστερός προς τα εμπρός. Παρακάτω παρουσιάζονται σχηματικά ο τρόπος κατεύθυνσης ενός ρομπότ με διαφορική κίνηση.

Στην εικόνα 51, μπορούμε να δούμε πάνω δεξιά την επιτόπια δεξιά στροφή, πάνω αριστερά την επιτόπια αριστερή στροφή, κάτω δεξιά την κίνηση ευθεία προς τα πίσω, και κάτω αριστερά την κίνηση ευθεία μπροστά. Όπως αναφέρθηκε προηγουμένως υπάρχει η δυνατότητα να επιτύχουμε δεξιά ή αριστερή στροφή επιταχύνοντας ή επιβραδύνοντας έναν από τους δυο τροχούς. Στην εικόνα 52, μπορούμε να δούμε αριστερή στροφή με επιβράδυνση του αριστερού τροχού ή επιτάχυνση του δεξιού τροχού. Και στα δυο σχήματα τα βέλη πάνω στους τροχούς αντιπροσωπεύουν την ταχύτητα και την φορά περιστροφής της συγκεκριμένης ρόδας, ενώ τα βέλη μπροστά ή πίσω από το όχημα αντιπροσωπεύουν την τελική κίνηση του οχήματος δεδομένης της εφαρμογής των συγκεκριμένων ταχυτήτων και περιστροφών στους τροχούς.



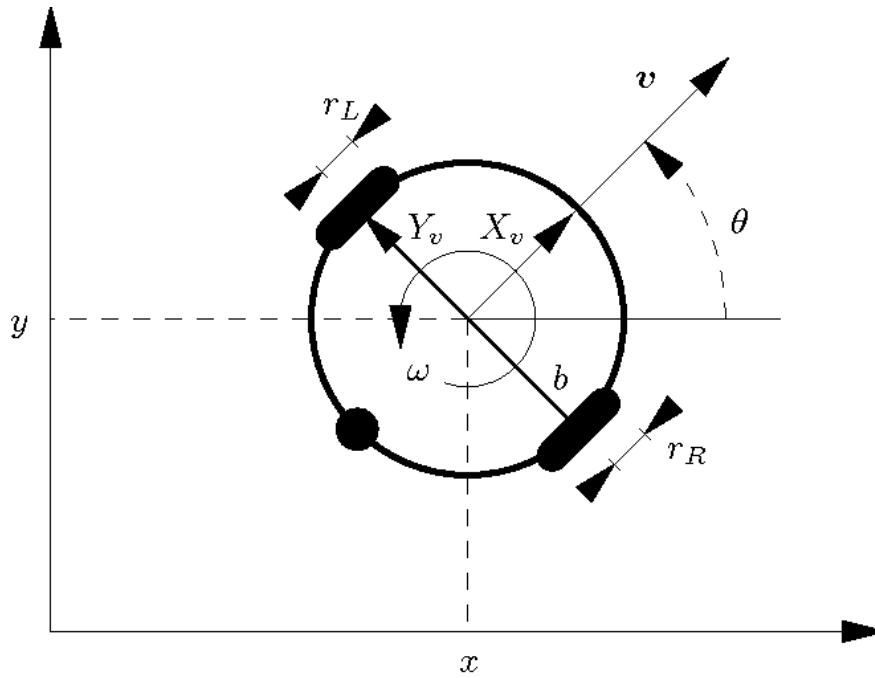
Εικόνα 49 ρομπότ διαφορικής κίνησης 1



Εικόνα 50 ρομπότ διαφορικής κίνησης 2

### Μελέτη κινηματικού προβλήματος.

Η συνεισφορά του κάθε τροχού στην κίνηση του ρομπότ εικόνα 53.



Εικόνα 51 συνεισφορά κάθε τροχού στο ρομπότ διαφορικής κίνησης

$$\dot{x}_R = \frac{r \cdot \dot{\phi} r}{2} \text{ (δεξιός τροχός)}, \quad \dot{x}_L = \frac{r \cdot \dot{\phi} l}{2} \text{ (αριστερός τροχός)}.$$

Στο ρομπότ διαφορικής κίνησης οι δύο αυτές ταχύτητες προστίθενται για να δώσουν την συνολική συνιστώσα  $\dot{x}_R$  του διάνυσματος  $\xi_R$  που είναι το διάνυσμα ταχυτήτων ως προς το σύστημα συντεταγμένων του ρομπότ.

Η συνιστώσα  $\dot{y}_R$  είναι πάντοτε μηδέν γιατί κανένας τροχός δεν συνεισφέρει στην πλαϊνή κίνηση του ρομπότ. Έχουμε ευθυγραμμίσει τον άξονα  $yy'$  με την ευθεία που συνδέει τα κέντρα των τροχών κι αφού δεν δεχόμαστε ολίσθηση άρα δεν υπάρχει συνιστώσα στον  $yy'$  άξονα.

Η 3<sup>η</sup> συνιστώσα στο διάνυσμα ταχυτήτων του ρομπότ ως προς το σύστημα συντεταγμένων του είναι αυτή της γωνιακής ταχύτητάς του. Ο υπολογισμός γίνεται πάλι ξεχωριστά για τον κάθε τροχό.

Για τον δεξί τροχό:  $\omega_R = \frac{r \cdot \dot{\phi} r}{2 \cdot b}$  (όπου  $b$  η απόσταση του κέντρου του ρομπότ από το κέντρο του τροχού).

Για τον αριστερό :  $\omega_L = -\frac{r \cdot \dot{\phi} l}{2 \cdot b}$  το (-) γιατί εάν περιστρέφεται θετικά ο τροχός περιστρέφει αρνητικά το ρομπότ σύμφωνα με τον κανόνα του δεξιόστροφου κοχλίου.

Άρα τελικά έχουμε  $\dot{x}_R = \frac{r \cdot \dot{\phi} r}{2} + \frac{r \cdot \dot{\phi} l}{2}$ ,  $\dot{y}_R = 0$  και  $\omega_R = \frac{r \cdot \dot{\phi} r}{2 \cdot b} - \frac{r \cdot \dot{\phi} l}{2 \cdot b}$  αυτό σε μορφή πίνακα.

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} r/2 & r/2 \\ 0 & 0 \\ r/2b & -r/2b \end{bmatrix} \begin{bmatrix} \dot{\phi}r \\ \dot{\phi}l \end{bmatrix} \text{ αυτό είναι το } \dot{\xi}_R.$$

Για το καθολικό η αδρανειακό σύστημα συντεταγμένων ισχύει  $R(\theta)\dot{\xi}_I = \dot{\xi}_R$  όπου  $R(\theta)$  ένας πίνακας στροφής ως προς των Z άξονα μιας και το ρομπότ κινείται μόνο στο επίπεδο  $xy$ .

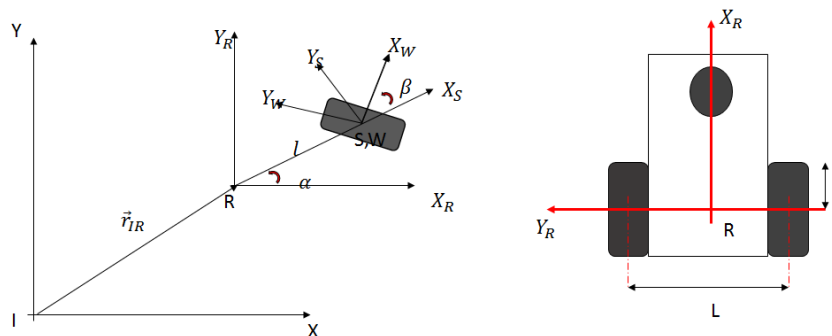
$$R(\theta) = \begin{bmatrix} \cos \theta & \sin \theta & 0 \\ -\sin \theta & \cos \theta & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$\text{άρα } \dot{\xi}_I = R(\theta)^{-1} \dot{\xi}_R = \begin{bmatrix} \cos \theta & -\sin \theta & 0 \\ \sin \theta & \cos \theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \frac{r*\dot{\phi}r}{2} + \frac{r*\dot{\phi}l}{2} \\ 0 \\ \frac{r*\dot{\phi}r}{2*b} - \frac{r*\dot{\phi}l}{2*b} \end{bmatrix}$$

Για μια πιο λεπτομερή και ακριβής περιγραφή του κινηματικού προβλήματος, θα πρέπει να περιγραφούν οι περιορισμοί που εισάγονται από τους τροχούς του ρομπότ.

Για να γίνει η έκφραση των περιορισμών θα υποθεθούν κάποιοι περιορισμοί στην κίνησή του .Α) Το επίπεδο του τροχού παραμένει πάντοτε κάθετο Β) υπάρχει πάντοτε ένα σημείο επαφής μεταξύ του τροχού και του επιπέδου του εδάφους. Γ) Δεν υπάρχει ολίσθηση στο σημείο επαφής Δηλαδή η τριβές που αναπτύσσονται είναι ΜΟΝΟ στατικές. Δ) ο τροχός κινείται μόνο κυλίνοντας και η περιστροφή του είναι σε επίπεδο κάθετο στο επίπεδο κίνησης.

Μ αυτά έχουμε λοιπόν ότι οι 1)τροχοί κυλίνουνται μόνο προς την αντίστοιχη κατεύθυνση και 2)δεν ολισθαίνουν πλευρικά.



Εικόνα 52 συστήματα συντεταγμένων

Στην εικόνα 54 βλέπουμε τα συστήματα συντεταγμένων I αδρανειακό (καθολικό), R το σασί του ρομπότ, και W του τροχού. Η θέση του τροχού ως προς το τοπικό πλαίσιο αναφοράς R καθορίζεται με τις πολικές συντεταγμένες  $(l, \alpha)$ . Η γωνία του επιπέδου του τροχού ως προς το σώμα του ρομπότ είναι η  $\beta$  που στην περίπτωση μας είναι σταθερή αν ο τροχός μπορούσε να περιστραφεί γύρω από κατακόρυφο άξονα που περνά από το σημείο επαφής και το κέντρο του, τότε η  $\beta$  θα ήταν  $\beta(t)$ , δηλαδή συναρτήση του χρόνου. Ο τροχός έχει ακτίνα  $r$  και η κίνησή του μεταβάλλεται με γωνιακή ταχύτητα  $\dot{\phi}(t)$ .

Ο περιορισμός κύλισης για τον κάθε τροχό είναι :



$$[\sin(\alpha + \beta) \quad -\cos(\alpha + \beta) \quad (-l) \cos \beta] \cdot R(\theta) \cdot \dot{\xi}_I - r \cdot \dot{\phi}(t) = 0$$

Ο περιορισμός ολίσθησης για τον κάθε τροχό είναι:

$$[\cos(\alpha + \beta) \quad \sin(\alpha + \beta) \quad l \sin \beta] \cdot R(\theta) \cdot \dot{\xi}_I = 0$$

Εάν τα γράψουμε όλα αυτά υπό μορφή πινάκων

Θα πάρουμε :

$\begin{bmatrix} J_1(\beta_s) \\ C_1(\beta_s) \end{bmatrix} \cdot R(\theta) \cdot \dot{\xi}_I = \begin{bmatrix} J_2 \cdot \dot{\phi} \\ 0 \end{bmatrix}$  στην περίπτωση μας  $\begin{bmatrix} J_{1f} \\ C_{1f} \end{bmatrix} \cdot R(\theta) \cdot \dot{\xi}_I = \begin{bmatrix} J_2 \cdot \dot{\phi} \\ 0 \end{bmatrix}$  γιατί οι τροχοί μας είναι σταθεροί δεν είναι π.χ. οδηγούμενοι να αλλάζουν κατεύθυνση.

Ο πίνακας  $J_2$  είναι ένας διαγώνιος πίνακας  $2 \times 2$  με τιμές τις ακτίνες  $r$  των τροχών.

Ο πίνακας  $J_1(\beta_s)$  δηλώνει ένα πίνακα με τις προβολές των τροχών στα επιμέρους επίπεδά τους.

$J_1(\beta_s) = \begin{bmatrix} J_{1f} \\ J_{1s}(\beta_s) \end{bmatrix}$  ο υποπίνακας  $J_{1s}(\beta_s)$  του προηγούμενου πίνακα στην περίπτωσή μας είναι μηδενικός.

Ομοίως και για τον  $C_1(\beta_s) = \begin{bmatrix} C_{1f} \\ C_{1s}(\beta_s) \end{bmatrix}$  ο υποπίνακας  $C_{1s}(\beta_s)$  στην περίπτωσή μας είναι μηδενικός.

Συνεπώς για το Ρομπότ μας έχουμε :

Δεξιός τροχός  $\alpha = -\pi/2$ ,  $\beta = \pi$ . Αριστερός τροχός  $\alpha = \pi/2$ ,  $\beta = 0$ .

$$\text{Άρα ο } \begin{bmatrix} J_{1f} \\ C_{1f} \end{bmatrix} = \begin{bmatrix} 1 & 0 & b \\ 1 & 0 & -b \\ 0 & 1 & 0 \end{bmatrix}$$

$$\text{Τελικά } \begin{bmatrix} 1 & 0 & b \\ 1 & 0 & -b \\ 0 & 1 & 0 \end{bmatrix} \cdot R(\theta) \cdot \dot{\xi}_I = \begin{bmatrix} r & 0 & 0 \\ 0 & r & 0 \\ 0 & 0 & 0 \end{bmatrix} \cdot \begin{bmatrix} \dot{\phi}_r \\ \dot{\phi}_l \\ 0 \end{bmatrix}$$

### Ευθύ κινηματικό πρόβλημα

Για να βρούμε λοιπόν την  $\dot{\xi}_I$  πρέπει να πολλαπλασιάσουμε από τα αριστερά με τον αντίστροφο του  $\begin{bmatrix} J_{1f} \\ C_{1f} \end{bmatrix}^{-1}$  και τον αντίστροφο του  $R(\theta)^{-1}$ . Τελικά παίρνουμε

$$\dot{\xi}_I = R(\theta)^{-1} \cdot \begin{bmatrix} 1/2 & 1/2 & 0 \\ 0 & 0 & 1 \\ 1/2b & -1/2b & 0 \end{bmatrix} \cdot \begin{bmatrix} r & 0 & 0 \\ 0 & r & 0 \\ 0 & 0 & 0 \end{bmatrix} \cdot \begin{bmatrix} \dot{\phi}_r \\ \dot{\phi}_l \\ 0 \end{bmatrix}$$

$$\text{ή } \dot{\xi}_I = \begin{bmatrix} \cos \theta(t) & -\sin \theta(t) & 0 \\ \sin \theta(t) & \cos \theta(t) & 0 \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} 1/2 & 1/2 & 0 \\ 0 & 0 & 1 \\ 1/2b & -1/2b & 0 \end{bmatrix} \cdot \begin{bmatrix} r & 0 & 0 \\ 0 & r & 0 \\ 0 & 0 & 0 \end{bmatrix} \cdot \begin{bmatrix} \dot{\phi}_r \\ \dot{\phi}_l \\ 0 \end{bmatrix}$$

προσοχή εδώ σε κάθε χρονικό διάστημα θα υπολογίζεται νέο  $R(\theta)^{-1}$  γιατί η  $\theta$  αλλάζει συνεχώς αφού το ρομπότ στρέφεται.

Αφού έχουμε υπολογίσει σε κάθε χρονική στιγμή τις ταχύτητες  $\dot{x}$ ,  $\dot{y}$ ,  $\dot{\phi}$  υπολογίζουμε τα  $\Delta x$ ,  $\Delta y$ ,  $\Delta \phi$  δηλαδή τις μετατοπίσεις στα συγκεκριμένα χρονικά διαστήματα.

$\Delta X = \dot{x} \cdot \Delta t$ ,  $\Delta y = \dot{y} \cdot \Delta t$ ,  $\Delta \phi = \dot{\phi} \cdot \Delta t$  ( θεωρείται ότι οι γωνιακές ταχύτητες των τροχών παραμένουν σταθερές σε κάθε χρονικό διάστημα)

Οι τελικές θέσεις και γωνία προκύπτουν από την άθροιση κάθε φορά των  $\Delta x$ ,  $\Delta y$ ,  $\Delta \phi$  σωρευτικά.

### Ανάστροφο κινηματικό πρόβλημα.

Από τις διαδοχικές θέσεις και γωνίες που προέκυψαν στο προηγούμενο κομμάτι της εργασίας υπολογίζω τα  $\dot{x}$ ,  $\dot{y}$ ,  $\dot{\phi}$  για κάθε χρονικό διάστημα, ως προς το καθολικό (αδρανειακό) σύστημα ως εξής :

$$\dot{x} = \frac{x_{i+1} - x_i}{t_{i+1} - t_i}, \dot{y} = \frac{y_{i+1} - y_i}{t_{i+1} - t_i}, \dot{\phi} = \frac{\phi_{i+1} - \phi_i}{t_{i+1} - t_i}$$

Αυτές οι ταχύτητες αποτελούν το διάνυσμα ταχυτήτων  $\dot{\xi}_I = \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\phi} \end{bmatrix}$

Από την εξίσωση  $\begin{bmatrix} J_{1f} \\ C_{1f} \end{bmatrix} \cdot R(\theta) \cdot \dot{\xi}_I = \begin{bmatrix} J_2 \cdot \dot{\phi} \\ 0 \end{bmatrix}$  το δεύτερο μέλος αποτελεί το διάνυσμα

ταχυτήτων στο σύστημα του Ρομπότ  $\dot{\xi}_R = \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix}$  άρα για να προκύψουν οι γωνιακές

ταχύτητες πρέπει να πολλαπλασιάσουμε από τα αριστερά και τα δύο μέλη της εξίσωσης με  $J_2^{-1}$  ο οποίος είναι

$$J_2^{-1} = \begin{bmatrix} 1/r & 0 & 0 \\ 0 & 1/r & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

Τελικά έχουμε την εξίσωση:

$$\dot{\phi} = \begin{bmatrix} 1/r & 0 & 0 \\ 0 & 1/r & 0 \\ 0 & 0 & 0 \end{bmatrix} \cdot \begin{bmatrix} 1 & 0 & b \\ 1 & 0 & -b \\ 0 & 1 & 0 \end{bmatrix} \cdot \begin{bmatrix} \cos \theta & \sin \theta & 0 \\ -\sin \theta & \cos \theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \cdot \dot{\xi}_I \text{ από όπου προκύπτουν}$$

οι αρχικές στροφές ανά λεπτό.