

Τμήμα Μηχανικών Πληροφορικής, Υπολογιστών και Τηλεπικοινωνιών
Διπλωματική Εργασία, Πρόγραμμα Μεταπτυχιακών Σπουδών στην Ρομποτική

Κατασκευή, Τεκμηρίωση και προγραμματισμός εκπαιδευτικών
εφαρμογών με την χρήση της αναπτυξιακής ρομποτικής πλατφόρμας
NVIDIA Kaya

Σαρίδης Λεωνίδα
Σέρρες 2023

Επιβλέπων Καθηγητής: Σταύρος Βολογιαννίδης

Πίνακας Περιεχομένων

Πίνακας Περιεχομένων.....	1
Πίνακας εικόνων.....	3
Περίληψη.....	4
Abstract.....	5
1. Εισαγωγή.....	6
1.1 Συνοπτική περιγραφή του Nvidia Kaya.....	6
1.2 Nvidia Isaac	7
1.2.1 Isaac Sight.....	8
1.2.2 Isaac Sim.....	9
1.3 Jetson Nano.....	10
1.3.1 Jetpack SDK	11
1.4 Intel RealSense D435	12
1.5 Κινητήρες Dynamixel	13
1.6 Πανκατευθυντικοί τροχοί.....	14
1.7 Επιλογές υλοποίησης του Kaya Robot.....	15
2. Υλοποίηση της κατασκευής.....	17
2.1 Λίστα και κόστος εξαρτημάτων	17
2.2 Κατασκευή 3D printed εξαρτημάτων	18
2.2.1 Τεχνολογία πρόσθεσης υλικού (Additive Manufacturing)	18
2.2.2 Εκτύπωση εξαρτημάτων	19
2.3 Συναρμολόγηση του Kaya Robot.....	22
2.3.1 Προετοιμασία και τοποθέτηση κινητήρων.....	22
2.3.2 Προετοιμασία καλωδίων και ηλεκτρολογικών συνδέσεων	26
2.3.3 Συναρμολόγηση σκελετού.....	28
2.3.4 Τοποθέτηση κινητήρων στον σκελετό.....	29
2.3.5 Τοποθέτηση IMU και μετατροπέων τάσεως	30
2.3.6 Τοποθέτηση κεντρικής βάσης Jetson Nano.....	31
2.3.7 Τοποθέτηση Jetson Nano	33
2.3.8 Τοποθέτηση κάμερας Realsense	33
2.3.9 Τοποθέτηση τροχών κίνησης.....	34
2.3.10 Τελικές ηλεκτρολογικές συνδέσεις.....	34
2.4 Εγκατάσταση λογισμικού	37

2.4.1	Εγκατάσταση Jetpack SDK στο Jetson Nano	37
2.4.2	Εγκατάσταση Isaac SDK στο σύστημα Host	38
2.4.3	Εγκατάσταση dependencies στο Jetson Nano.....	39
3	Εκτέλεση εφαρμογών	39
3.1	Εφαρμογή Joystick	39
3.2	Εφαρμογή Follow Me.....	41
3.3	Εφαρμογή Gmapping.....	44
4	Συμπεράσματα και μελλοντική εξέλιξη.....	47
4.1	Δυσκολίες κατά την κατασκευή του Kaya Robot.....	47
4.2	Προτάσεις για μελλοντική εξέλιξη.....	47
4.2.1	Isaac ROS.....	47
	Βιβλιογραφία.....	49

Πίνακας εικόνων

Figure 1: Isaac SIM environment	7
Figure 2: Isaac Sight environment.....	8
Figure 3: Isaac Omniverse	9
Figure 4: Jetson Nano Developer Kit.....	10
Figure 5: Intel RealSense D435	12
Figure 6: Dynamixel MX-12W	13
Figure 7: VEXpro 3.25 Omni Wheel	14
Figure 8: Κατευθύνσεις κίνησης Omni Wheel	14
Figure 9: Υλοποίηση του Kaya Robot με την κάμερα Realsense υπό γωνία	15
Figure 10: Υλοποίηση του Kaya με την κάμερα Realsense και καπέλο	15
Figure 11: Ψηφιακή απεικόνιση της υλοποίησης του Kaya	16
Figure 12: Στιγμιότυπα από την διαδικασία 3D εκτύπωσης	21
Figure 13: Άνοιγμα θύρας COM.....	22
Figure 14: Αναζήτηση κινητήρων.....	23
Figure 15: Παραμετροποίηση ID κινητήρα.....	24
Figure 16: Προετοιμασία στερέωσης κινητήρα.....	24
Figure 17: Στερέωση κινητήρα στην βάση.....	25
Figure 18: Στερέωση βάσης τροχού στον κινητήρα	26
Figure 19: Περιγραφή καλωδίων	27
Figure 20: Τοποθέτηση καλωδίων τροφοδοσίας στην βάση μπαταρίας	28
Figure 21: Έδρανα κινητήρων & MakerBeams	28
Figure 22: Τοποθέτηση προφίλ MakerBeam.....	29
Figure 23: Διάταξη κινητήρων	30
Figure 24: Σύνδεση κινητήρων	30
Figure 25: Τοποθέτηση IMU & μετατροπέων τάσης	31
Figure 26: Κεντρικός σκελετός	31
Figure 27: Στερέωση βάσης Jetson	32
Figure 28: Τοποθέτηση κεντρικού διακόπτη & Dynamixel Power Hub.....	32
Figure 29: Τοποθέτηση τερματικού συνδέσεων	33
Figure 30: Τοποθέτηση κάρτας δικτύου και στερέωση Jetson Nano.....	33
Figure 31: Στερέωση κάμερας RealSense	34
Figure 32: Τοποθέτηση τροχών κίνησης.....	34
Figure 33: Ρύθμιση τροφοδοσίας Jetson Nano	35
Figure 34: Ηλεκτρολογικό διάγραμμα Kaya Robot.....	36
Figure 35: Ολοκλήρωση συναρμολόγησης Kaya Robot	36
Figure 36: Εγγραφή λειτουργικού Jetpack	37
Figure 37: Εκτέλεση εφαρμογής Joystick (1)	40
Figure 38: Εκτέλεση εφαρμογής Joystick (2)	41
Figure 39: Πρότυπα AprilTags	41
Figure 40: Εκτέλεση εφαρμογής Follow Me (1).....	43
Figure 41: Εκτέλεση εφαρμογής Follow Me (2).....	43
Figure 42: Απεικόνιση εικόνας βάθους στο Kaya Robot	45
Figure 43: Κατασκευή χάρτη στο σύστημα Host	46
Figure 44: Δομή Isaac ROS	48

Περίληψη

Η παρούσα εργασία εκπονήθηκε στο Τμήμα Μηχανικών Πληροφορικής, Υπολογιστών και Τηλεπικοινωνιών του Διεθνούς Πανεπιστημίου Ελλάδος στα πλαίσια της διπλωματικής εργασίας του φοιτητή Σαρίδη Λεωνίδα.

Στόχος αυτής της εργασίας ήταν η κατασκευή της αναπτυξιακής ρομποτικής πλατφόρμας της NVIDIA, Kaya Robot η τεκμηρίωση της και η εκτέλεση εκπαιδευτικών εφαρμογών.

Η παρούσα εργασία χωρίζεται σε τέσσερα κεφάλαια. Στο πρώτο κεφάλαιο γίνεται μία συνοπτική εισαγωγή στην οποία περιγράφονται αναλυτικά όλα τα δομικά μέρη του NVIDIA Kaya. Γίνεται αναφορά και επεξήγηση των εξαρτημάτων του, καθώς και του λειτουργικού συστήματος που χρησιμοποιείται για την λειτουργία του ρομπότ. Στο δεύτερο κεφάλαιο γίνεται αναλυτική περιγραφή στην διαδικασία κατασκευής του ρομπότ τις μεθόδους που χρησιμοποιήθηκαν για αυτή και την ανάλυση του κόστους των εξαρτημάτων του. Στο τρίτο κεφάλαιο περιγράφεται η διαδικασία εγκατάστασης και εκτέλεσης των εκπαιδευτικών εφαρμογών που συνοδεύουν την πλατφόρμα του Kaya. Τέλος, στο τέταρτο κεφάλαιο γίνεται αναφορά στις δυσκολίες που προέκυψαν κατά την κατασκευή και σε νέες τεχνολογίες που έχουν αναπτυχθεί και θα μπορούσαν να ενσωματωθούν μελλοντικά στην πλατφόρμα του Kaya.

Abstract

This paper was written for the Department of Information Technology Engineering of the International Hellenic University in the framework of the student's thesis Saridis Leonidas.

The aim of this project was to cover the build process of the development robotic platform of NVIDIA, Kaya Robot, the documentation and the showcase of example applications.

This thesis is divided along four chapters. The first chapter contains an introduction in the robotic platform of Kaya robot, in which the hardware, parts and the software of it are being presented in detail. The second chapter describes the build process of the robot, the construction methods that were used and the cost of the parts that were used. The third chapter contains the description of the installation and execution procedure of example applications and use cases of the robot. Finally, the fourth chapter lists the issues that were face during the build of the robot and introduces newly developed technologies that could be used in future implementations of the Kaya robot.

1. Εισαγωγή

1.1 Συνοπτική περιγραφή του Nvidia Kaya

Το NVIDIA Kaya είναι μια ρομποτική πλατφόρμα που δημιουργήθηκε με σκοπό να αναδείξει τις δυνατότητες που παρέχει η πλατφόρμα ανάπτυξης λογισμικού, Isaac SDK χρησιμοποιώντας ως κεντρική υπολογιστική μονάδα την αναπτυξιακή υπολογιστική πλατφόρμα της NVIDIA, Jetson Nano. Η κατασκευή του Kaya έχει σχεδιαστεί με γνώμονα την προσβασιμότητα σε ερασιτέχνες, φοιτητές αλλά και ερευνητές ώστε να είναι εύκολο να υλοποιηθεί χωρίς να απαιτείται ειδικός εξοπλισμός. Για αυτό τον λόγο, τα δομικά του μέρη είναι κατασκευασμένα με την χρήση της τεχνολογίας πρόσθεσης υλικού (Additive Manufacturing), ευρύτερα γνωστή ως 3D printing, το οποίο κάνει εύκολη την κατασκευή των εξαρτημάτων του. Εκτός του Jetson Nano, χρησιμοποιεί την στερεοσκοπική κάμερα βάθους της Intel, Realsense D435 η οποία επιτρέπει στο ρομπότ να λαμβάνει δεδομένα εισόδου τα οποία στην συνέχεια επεξεργάζεται έτσι ώστε να σχεδιάσει το πως θα κινηθεί στον χώρο στον οποίο έχει τοποθετηθεί. Για την λήψη στοιχείων οδομετρίας και τον καθορισμό της θέσης του ρομπότ χρησιμοποιείται η μονάδα μέτρησης αδράνειας (Inertial Measurement Unit) της Bosch, BMI160 σε συνδυασμό με την θέση περιστροφής των κινητήριων τροχών.

Το σύστημα μετάδοσης κίνησης του Kaya βασίζεται σε ένα πανκατευθυντικό σύστημα κίνησης αποτελούμενο από τρεις Omnidirectional τροχούς, το οποίο του επιτρέπει να κινείται προς οποιαδήποτε κατεύθυνση, ανεξαρτήτως από την φορά περιστροφής του. Ως κινητήρες των τροχών, χρησιμοποιούνται οι Dynamixel MX-12W, γνωστοί στον χώρο των ρομποτικών κατασκευών για την συμβατότητα και την ακρίβειά τους. Για την επικοινωνία με άλλα συστήματα και τον χειρισμό του, το Kaya ενσωματώνει στην πλατφόρμα του Jetson την κάρτα δικτύου και Bluetooth της Intel, 8265. Όλο το σύνολο ηλεκτρονικών εξαρτημάτων του NVIDIA Kaya τροφοδοτείται από μια μπαταρία ηλεκτρικών εργαλείων χειρός της DeWalt της τάσης των 18V και στην συνέχεια με την χρήση μετατροπέων τάσης διοχετεύεται η απαιτούμενη τάση σε κάθε ξεχωριστό τμήμα του ρομπότ.

1.2 Nvidia Isaac

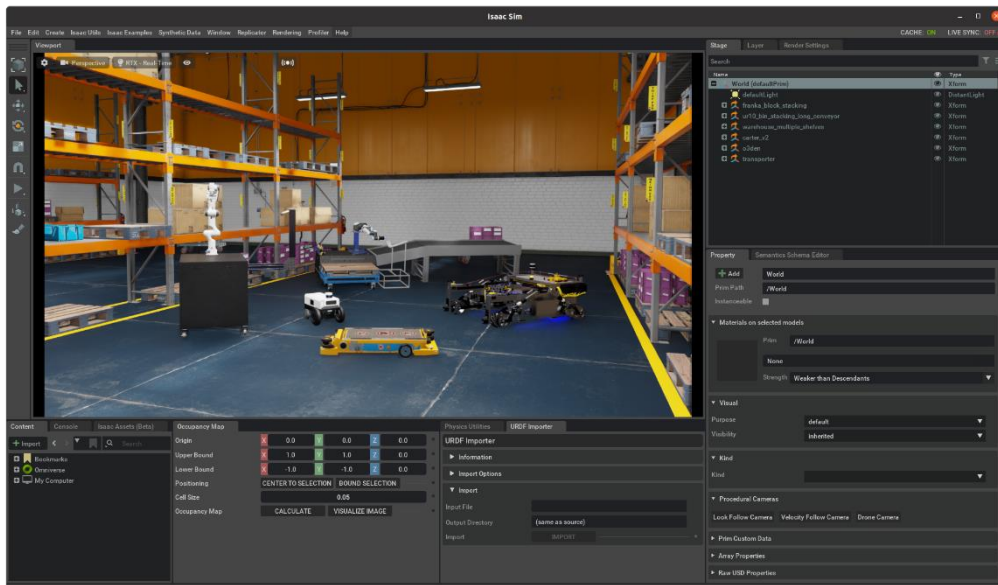


Figure 1: Isaac SIM environment

Το πακέτο ανάπτυξης λογισμικού Isaac SDK (Software Development Kit) της NVIDIA είναι μία πλατφόρμα που παρέχει τα απαραίτητα εργαλεία για την σχεδίαση, την ανάπτυξη και την διαχείριση της λειτουργίας ρομποτικών συστημάτων. Προσφέρει μία πολύ μεγάλη συλλογή έτοιμων αλγορίθμων που καλύπτουν βασικές αλλά και κάποιες πιο σύνθετες λειτουργικότητες ρομποτικών συστημάτων, όπως η πλοήγηση, ο εντοπισμός θέσης και η αναγνώριση αντικειμένων μέσω καμερών. Στους συγκεκριμένους αλγόριθμους έχει εφαρμοστεί διαδικασίες βελτιστοποίησής τους με σκοπό η εκτέλεσή τους να γίνεται με την χρήση μονάδων επεξεργασίας γραφικών (GPU accelerated algorithms) στοχεύοντας στην υψηλή αποδοτικότητα και χαμηλούς χρόνους εκτέλεσης. Το Isaac SDK Engine παρέχει την δυνατότητα εύκολης ανάπτυξης παραμετροποιήσιμων εφαρμογών και της τελικής ενσωμάτωσής τους σε ένα πραγματικό ρομποτικό σύστημα το οποίο μπορεί να χρησιμοποιηθεί σε βιομηχανικούς και εκπαιδευτικούς χώρους. Αυτό επιτυγχάνεται με την βοήθεια ενός συνόλου εργαλείων το οποίο αποτελείται από βιβλιοθήκες λογισμικού, τεχνικά εγχειρίδια, τμήματα κώδικα με την μορφή παραδειγμάτων, αυτοματοποιημένες διεργασίες και έγγραφα καθοδήγησης (documentation) τα οποία οι προγραμματιστές μπορούν να χρησιμοποιήσουν κατά την διαδικασία ανάπτυξης και εξέλιξης των δικών τους εφαρμογών. Το Isaac SDK συνδυάζεται με το Isaac SIM (Figure 1) έτσι ώστε να συμβάλλει στην ταχύτερη ανάπτυξη, τον έλεγχο, και την εκπαίδευση των ρομπότ σε ένα εικονικό περιβάλλον προσομοίωσης πριν περάσει το τελικό σχέδιο του ρομπότ στην διαδικασία κατασκευής του έτσι ώστε να εξαλείφεται η ανάγκη για την κατασκευή πολλών πρωτοτύπων μειώνοντας έτσι το κόστος και βελτιώνοντας το τελικό αποτέλεσμα.

1.2.1 Isaac Sight

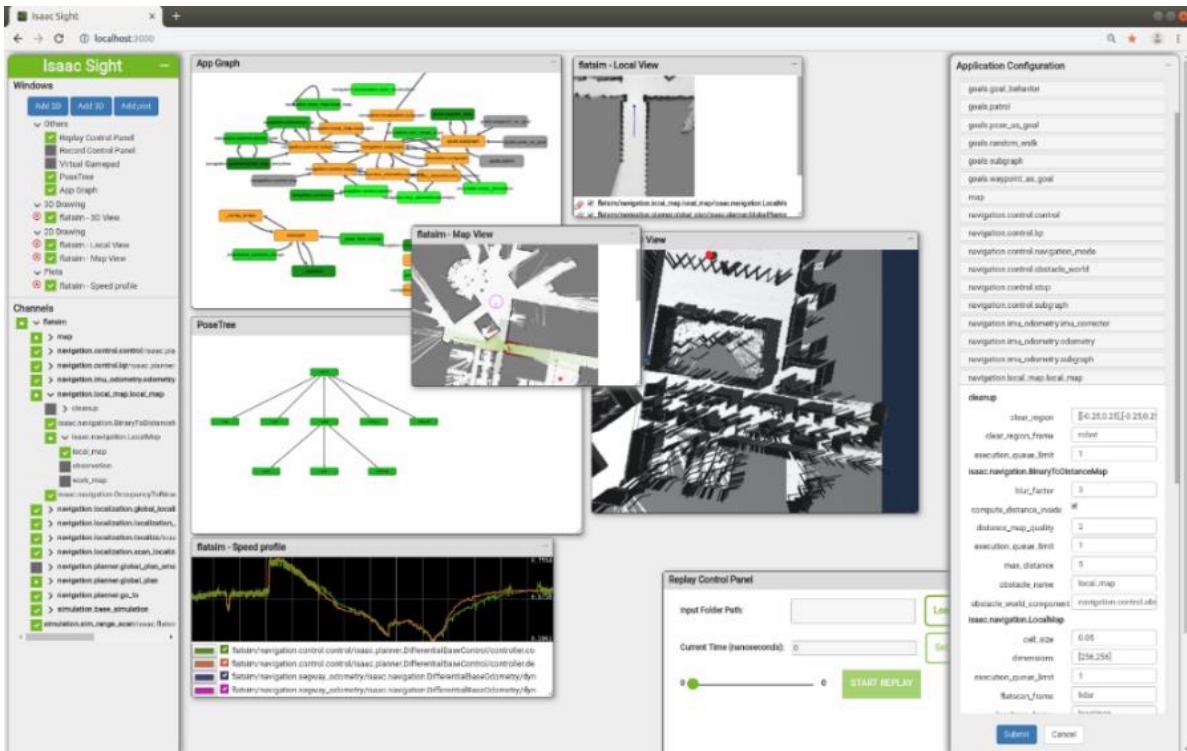


Figure 2: Isaac Sight environment

Για την επικοινωνία με τα ρομποτικά συστήματα και τον απομακρυσμένο έλεγχο αυτών το Isaac SDK χρησιμοποιεί το Isaac Sight (Figure 2). Το Isaac Sight είναι ένα API (Application Programmable Interface) το οποίο υλοποιείται με την μορφή μίας Web Based εφαρμογής η οποία παρέχει την δυνατότητα επικοινωνίας με το ρομπότ μέσω ενός εξωτερικού συστήματος οικοδεσπότη (Host system) μέσω δικτύου για τον έλεγχο αλλά και την οπτικοποίηση των λειτουργιών των ρομπότ.

Στο κεντρικό τμήμα του Sight, τοποθετούνται δισδιάστατες απεικονίσεις διαφόρων τμημάτων της στοίβας πλοήγησης του Isaac. Τα γραφήματα σχεδιάζονται χρησιμοποιώντας τα κανάλια επικοινωνίας με το ρομπότ μέσω δικτύου για την λήψη δεδομένων.

Στο αριστερό τμήμα, είναι διαθέσιμα για επιλογή τα κανάλια επικοινωνίας που μπορούν να παρέχουν δεδομένα οπτικοποίησης. Επιλέγοντας ή αποεπιλέγοντας κάποιο από τα διαθέσιμα κανάλια, καθορίζεται εάν το ρομπότ θα στέλνει δεδομένα για το συγκεκριμένο κανάλι ή όχι.

1.2.2 Isaac Sim

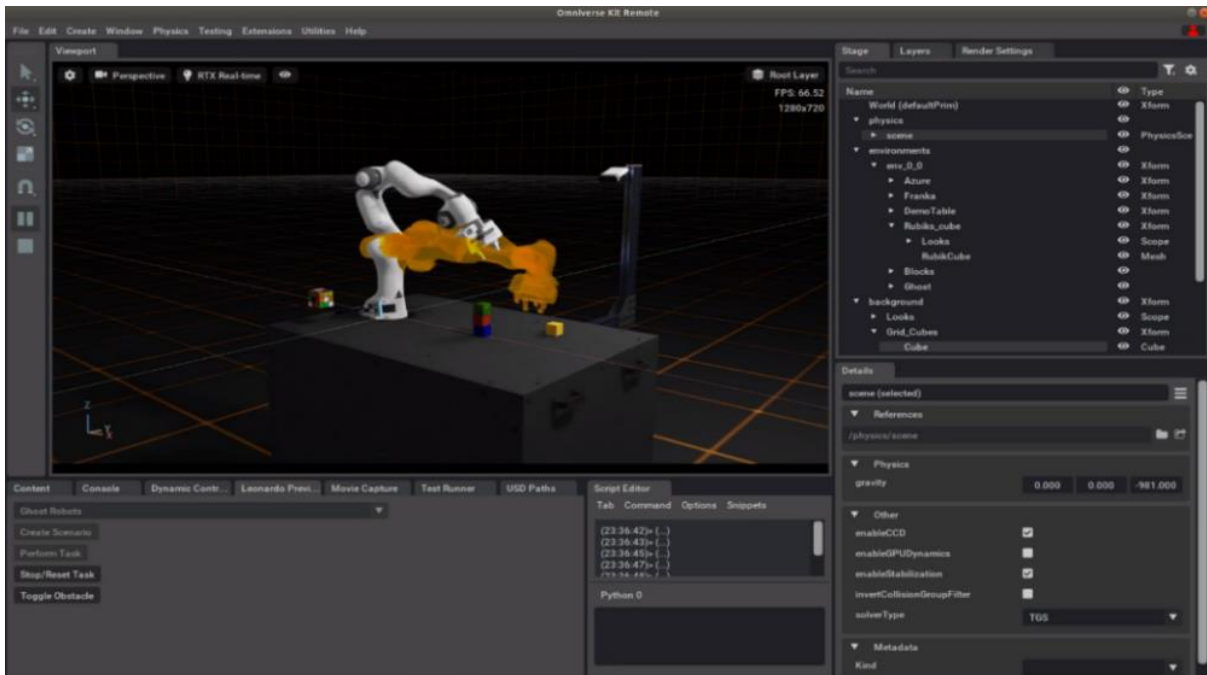


Figure 3: Isaac Omniverse

Το Isaac Sim είναι ένα περιβάλλον προσομοίωσης ρομποτικών συστημάτων το οποίο έχει στηριχτεί στην πλατφόρμα NVIDIA Omniverse (Figure 3). Παρέχει τα απαραίτητα εργαλεία για την ανάπτυξη εικονικών ρομπότ και χώρων στους οποίους μπορούν να τοποθετηθούν και να διεξαχθούν πειράματα και δοκιμές. Με την χρήση της τεχνολογίας γραφικών RTX προσφέρει φωτορεαλιστικά γραφικά τα οποία σε συνδυασμό με την μηχανή φυσικής PhysX της NVIDIA η οποία παρέχει την δυνατότητα προσομοίωσης φυσικών ιδιοτήτων στα ρομπότ όπως βαρύτητα και μοντέλο συγκρούσεων, αποτελούν ένα περιβάλλον στο οποίο μπορούν να αναπτυχθούν κατασκευές με μεγάλη ακρίβεια αναπαράστασης της δομής και της λειτουργίας τους στον πραγματικό κόσμο.

1.3 Jetson Nano



Figure 4: Jetson Nano Developer Kit

Η αναπτυξιακή πλατφόρμα Jetson Nano της NVIDIA είναι ένας υπολογιστής μικρού μεγέθους ο οποίος έχει σχεδιαστεί με βασικό σκοπό την εκτέλεση εφαρμογών οι οποίες απαιτούν υψηλή υπολογιστική ισχύ. Με την χρήση μονάδας επεξεργασίας γραφικών διαθέτει ισχύ που του επιτρέπει να εκτελεί σύγχρονα νευρωνικά δίκτυα τα οποία αποφέρουν βέλτιστα αποτελέσματα σε εφαρμογές όπως την αναγνώριση προτύπων σε εικόνες, ανίχνευση αντικειμένων και επεξεργασία ομιλίας. Η ισχύς του σε συνδυασμό με την φορητότητα που παρέχει το μικρό του μέγεθος καθώς και το σχετικά μικρό του κόστος, το καθιστούν ιδανικό για την υιοθέτηση του σε ρομποτικά συστήματα. Συναντάται συχνά σε ερευνητικές εφαρμογές οι οποίες απαιτούν πολλαπλές δοκιμές και συνεχή εξέλιξη πρωτοτύπων συστημάτων. Ταυτόχρονα, αποτελεί μια καλή λύση για την εισαγωγή φοιτητών στους τομείς της ρομποτικής και της τεχνητής νοημοσύνης.

Κύρια τεχνικά χαρακτηριστικά:

- CPU: Quad-core ARM A57 @ 1.43 GHz
- GPU: 128-core Maxwell
- Μνήμη: 4 GB 64-bit LPDDR4 25.6 GB/s
- Αποθηκευτικός χώρος: microSD Card
- Σύνδεση δικτύου: Gigabit Ethernet, M.2 Key E
- Θύρες απεικόνισης: HDMI & Display port
- USB: 4x USB 3.0, USB 2.0 Micro-B
- Δίαυλοι επικοινωνίας: GPIO, I2C, I2S, SPI, UART
- Τροφοδοσία: 5V 2.5A Micro-USB adapter | 5V 4A DC barrel jack adapter

1.3.1 Jetpack SDK

Το Jetpack SDK (Software Development Kit) είναι ένα λειτουργικό σύστημα βασισμένο σε Linux το οποίο έχει αναπτυχθεί από την NVIDIA για την διευκόλυνση ανάπτυξης εφαρμογών στην οικογένεια των Jetson platforms, όπως το Jetson Nano, Jetson Xavier και Jetson TX1/TX2. Προσφέρει μια σειρά από βιβλιοθήκες, εργαλεία και πόρους που διευκολύνουν την ανάπτυξη λογισμικού για αυτές τις πλατφόρμες, ειδικά για εφαρμογές που σχετίζονται με την τεχνητή νοημοσύνη (AI), ρομποτική και μηχανική όραση.

Κύρια Χαρακτηριστικά:

- CUDA Toolkit: Περιλαμβάνει ένα πλήρες περιβάλλον ανάπτυξης εφαρμογών σε C και C++ με την χρήση GPU, παρέχοντας compiler για τις μονάδες επεξεργασίας γραφικών της NVIDIA.
- cuDNN (CUDA Deep Neural Network): Η βιβλιοθήκη νευρωνικών δικτύων cuDNN παρέχει μια σειρά από εφαρμογές για την υλοποίηση θεμελιωδών διαδικασιών μηχανικής μάθησης.
- Multimedia API: Παρέχει εργαλεία για την καταγραφή και αναπαραγωγή βίντεο, καθώς και για την επεξεργασία εικόνας.

- Μηχανική όραση: Η βιβλιοθήκη λογισμικού VPI (Vision Programming Interface) παρέχει μία σειρά αλγορίθμων Μηχανικής Όρασης και Επεξεργασίας εικόνας που εκτελούνται κάνοντας χρήση μονάδων επεξεργασίας γραφικών που παρέχει το Jetson Nano.

1.4 Intel RealSense D435



Figure 5: Intel RealSense D435

Η κάμερα της Intel, RealSense D435 είναι μία στερεοσκοπική κάμερα βάθους ικανή να χρησιμοποιηθεί σε ένα μεγάλο πλήθος εφαρμογών. Διαθέτει ευρύ οπτικό πεδίο το οποίο την καθιστά ιδανική για χρήση σε εφαρμογές ρομποτικών συστημάτων, επαυξημένης πραγματικότητας και εικονικής πραγματικότητας. Παρέχοντας εμβέλεια λήψης έως δέκα μέτρα και λόγω του μικρού της μεγέθους και της ανθεκτικής της κατασκευής μπορεί να τοποθετηθεί με ευκολία σε κατασκευές που απαιτούν φορητότητα και αντοχή σε συνθήκες εξωτερικού χώρου. Η απόδοση της σε συνθήκες χαμηλού φωτισμού διευκολύνει την χρήση της σε εφαρμογές πλοήγησης αυτόνομων ρομποτικών συστημάτων και αναγνώρισης αντικειμένων.

Κύρια χαρακτηριστικά:

- Ανάλυση εικόνας βάθους: Μέχρι και 1280 x 720
- Frame Rate: Μέχρι και 90 FPS
- Οπτικό πεδίο (FOV): 87° x 58°
- Εμβέλεια: 10 μέτρα
- Συνδεσιμότητα: USB 3.0

1.5 Κινητήρες Dynamixel



Figure 6: Dynamixel MX-12W

Οι σερβοκινητήρες Dynamixel κατασκευάζονται από την εταιρεία Robotis. Συναντώνται συχνά σε εφαρμογές όπως ρομποτικά συστήματα, αυτόνομα οχήματα, βραχίονες και σε ένα ευρύτερο πλαίσιο κατασκευών που απαιτούν περιστροφική κίνηση, λόγω της υψηλής ακρίβειας που διαθέτουν. Παρέχουν την δυνατότητα προγραμματισμού της ταχύτητας περιστροφής, θέσης και ροπής. Προσφέρουν ακριβή έλεγχο σε μικρές αλλαγές θέσης. Χρησιμοποιούν διασυνδέσεις TTL ή RS-485 για την επικοινωνία με μονάδες ελεγκτών για την διαχείριση τους. Πολλά από τα μοντέλα τους περιλαμβάνουν ενσωματωμένους αισθητήρες για τη μέτρηση της ταχύτητας, της θέσης και της θερμοκρασίας καθιστώντας τους ασφαλής επιλογή για τόσο για πειραματικές όσο και για βιομηχανικές εφαρμογές.

1.6 Πανκατευθυντικοί τροχοί



Figure 7: VEXpro 3.25 Omni Wheel

Οι πανκατευθυντικοί τροχοί γνωστοί και ως Omnidirectional Wheels επιτρέπουν την κίνηση ενός οχήματος προς οποιαδήποτε κατεύθυνση. Η σχεδίαση τους στηρίζεται στην διάταξη ενός συμβατικού τροχού με την προσθήκη κυλίνδρων στην περίμετρό του.



Figure 8: Κατευθύνσεις κίνησης Omni Wheel

1.7 Επιλογές υλοποίησης του Kaya Robot

Ο σχεδιασμός του NVIDIA Kaya παρέχει την δυνατότητα υλοποίησής του σε κάποιες διαφορετικές μορφές, χρησιμοποιώντας εναλλακτικό Hardware και εξαρτήματα ανάλογα με την χρήση για την οποία προορίζεται η συγκεκριμένη υλοποίηση του ρομπότ:



Figure 9: Υλοποίηση του Kaya Robot με την κάμερα Realsense υπό γωνία

- **RealSense υπό γωνία:** Είναι η βασική μορφή του Kaya Robot. Η κάμερα της Intel, RealSense τοποθετείται με γωνία λήψης προς το επίπεδο με σκοπό το ρομπότ να έχει καλύτερη οπτική σε αντικείμενα που συναντά μπροστά του κατά την κίνησή του.
- **RealSense σε ευθεία:** Η συγκεκριμένη υλοποίηση τοποθετεί την κάμερα RealSense με κάθετη γωνία προς το επίπεδο προσφέροντας μεγαλύτερο οπτικό πεδίο στο ρομπότ δίνοντάς του την δυνατότητα να λάβει δεδομένα για μεγαλύτερες αποστάσεις στον χώρο που κινείται.



Figure 10: Υλοποίηση του Kaya με την κάμερα Realsense και καπέλο

- **RealSense & Καπέλο:** Μια υλοποίηση (Figure 10) που καθιστά το ρομπότ ασφαλέστερο για χρήση σε χώρους με κοινό, καλύπτοντας και προστατεύοντας τα κύρια ηλεκτρονικά και ευαίσθητα εξαρτήματά του.

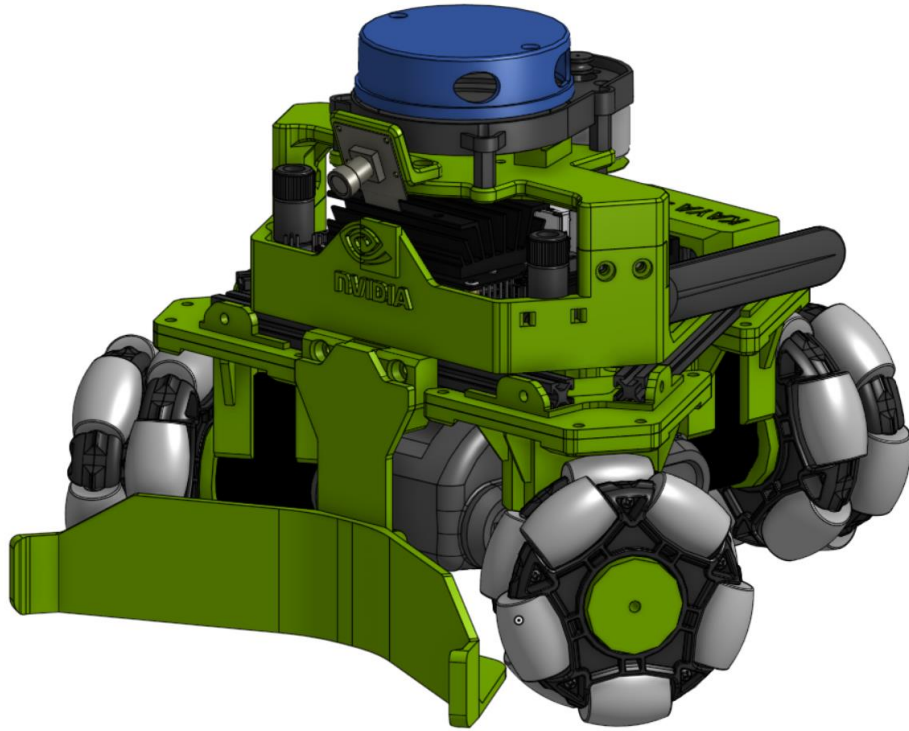


Figure 11: Ψηφιακή απεικόνιση της υλοποίησης του Kaya με αισθητήρα Lidar και την κάμερα Leopard IMX219

- **Lidar & Κάμερα:** Εναλλακτική υλοποίηση (Figure 11) του Kaya Robot στην οποία χρησιμοποιείται ένας αισθητήρας lidar (YDLidar X3) για την πλοήγηση και η κάμερα Leopard IMX219 για την επεξεργασία εικόνας.

2. Υλοποίηση της κατασκευής

2.1 Λίστα και κόστος εξαρτημάτων

Το τελικό κόστος κάθε εξαρτήματος περιλαμβάνει έξοδα αποστολής και τελωνειακούς δασμούς όπου εφαρμόστηκαν. Το κόστος των εκτυπωμένων (3d printed) βασικών τμημάτων της κατασκευής του ρομπότ, έχει υπολογιστεί με βάση το βάρος του κάθε εξαρτήματος και το κόστος του υλικού κατασκευής. Τα εξαρτήματα εκτυπώθηκαν σε ερασιτεχνικό πλαίσιο για εξοικονόμηση κόστους και το τελικό τους κόστος δεν αντικατοπτρίζει το κόστος λιανικής για την παροχή της συγκεκριμένης υπηρεσίας.

	Τίτλος εξαρτήματος	Σειριακός αριθμός	QTY	Πηγή αγοράς	Κόστος
1	3D Printer PLA+ Filament 1.0KG		1	grobotronics.gr	€ 25.00
2	3D Printed Parts				€ 19.80
3	NVIDIA Jetson Nano Developer Kit	945-13450-0000-000	1	mouser.gr	€ 105.00
4	MicroSD Card (32GB)	SDSQUNC-032G-GN6MA	1	eshop.gr	€ 8.00
5	Intel Dual Band Wireless-Ac w/Bluetooth	8265NGW	1	hellasphone.gr	€ 28.00
6	Dual Band Wi-Fi Antennas & Cables	CHAOHANG 6484109	1	ebay.com	€ 18.00
7	Intel RealSense Depth Camera D435	961448	1	ebay.com (used)	€ 125.00
8	Bosch GY-BMI160 6DOF IMU Board	GY-BMI160	1	cableworks.gr	€ 6.90
9	Dynamixel MX-12W σερβοκινητήρες	902-0086-000	3	mybotshop.de	€ 250.00
10	Robotis U2D2 ελεγκτής κινητήρων	RO-902-0132-000	1		
11	Dynamixel 6 Port AX/MX Power Hub	IL-3PHUB	1	robosavvy.com	€ 50.00
12	VEX 3.25" Omni-Directional Wheel	217-4775	3	vex robotics uk	€ 160.00
13	DeWalt 20V MAX 2AH μπαταρία	DCB203	1	skroutz.gr	€ 50.00
14	DeWalt 20V MAX φορτιστής	DCB107	1	skroutz.gr	€ 32.00
15	5V 6A DC-DC μετατροπέας τάσης	D24V60F5	1	skpang.co.uk	€ 32.00
16	12V 5A DC-DC μετατροπέας τάσης	EA60-12V	1	ebay	€ 16.00
17	Τερματικό ηλεκτρογικών συνδέσεων	7527K44	1	marinlife.gr	€ 10.29
18	Τερματικά καλωδίων	HC28-6		ABC Store	€ 3.00
19	MakerBeam 150x10x10mm	BAMB150P06	3	tinkerforge.com	
20	MakerBeam βίδες, παξιμάδια T-slot n	101619			€ 40.00
21	Ηλεκτρολογικό καλώδιο 18AWG			grobotronics.gr	
22	Βύσμα τροφοδοσίας 2.1 x 5.5mm		1		
23	Διακόπτης ON - OFF		1		
24	M3 18mm αποστάτης		3		
25	Βίδες M3 x 35mm		3		
26	Βίδες M3 x 8mm				
27	Βίδες M3 x 5mm				
28	Παξιμάδια M3				
29	Βίδες M2 x 6mm				10 €
	Τελικό κόστος				€ 988.99

2.2 Κατασκευή 3D printed εξαρτημάτων

2.2.1 Τεχνολογία πρόσθεσης υλικού (Additive Manufacturing)

Το Additive Manufacturing αναφέρεται σε μία ομάδα τεχνολογιών που χρησιμοποιούνται για την κατασκευή γεωμετρικών εξαρτημάτων προσθέτοντας υλικό σε στρώσεις. Χρησιμοποιείται για την κατασκευή πολύπλοκων εξαρτημάτων, χωρίς να απαιτείται η χρήση καλουπιών, μητρών ή άλλων ειδικών εργαλείων. Ο καθορισμός των σημείων εναπόθεσης υλικού γίνεται κατά την κίνηση μιας κεφαλής εξαγωγής υλικού η οποία κινείται σε ένα καρτεσιανό σύστημα (X, Y, Z). Ο έλεγχος της κίνησης γίνεται μέσω ψηφιακών ελεγκτών. Μία από τις μορφές με την οποία η συγκεκριμένη τεχνολογία είναι ευρύτερα γνωστή και διαδεδομένη είναι το 3D printing. Με το 3D printing δίνεται η δυνατότητα παραγωγής εξαρτημάτων ακόμη και σε ερασιτεχνικό πλαίσιο καθώς οι τρισδιάστατοι εκτυπωτές πολύ συχνά συναντώνται ως προϊόντα εμπορικής μορφής για ιδιωτική χρήση.


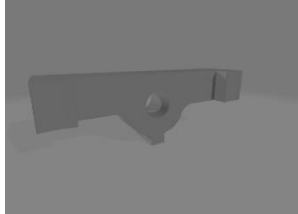
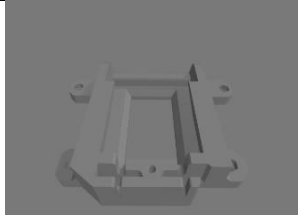
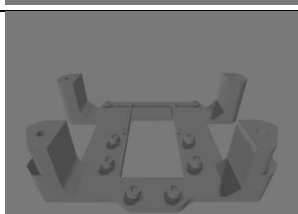
Κύρια πλεονεκτήματα:

- Ελευθερία Σχεδιασμού: Δυνατότητα σχεδιασμού εξαρτημάτων χωρίς μεγάλους περιορισμούς.
- Ταχύτητα: Μείωση του χρόνου μετάβασης από τον σχεδιασμό του εξαρτήματος στην γραμμή παραγωγής.
- Βέλτιστη Ανάπτυξη Πρωτοτύπου: Επιτρέπει τη δημιουργία πολλών διαφορετικών πρωτοτύπων που μπορούν να παραχθούν ταχύτερα και φθηνότερα σε σύγκριση με τις παραδοσιακές μεθόδους.
- Εξατομίκευση για τις ανάγκες του καταναλωτή: Μεγάλη εφαρμογή στους τομείς της ιατρικής για την κατασκευή προσθετικών μελών.
- Εξοικονόμηση ενέργειας και υλικού: Χρήση λιγότερου υλικού κατά την κατασκευή, ελαχιστοποίηση απώλειας υλικού και χρήση μηχανημάτων με χαμηλότερο ενεργειακό κόστος.


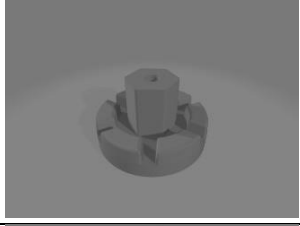
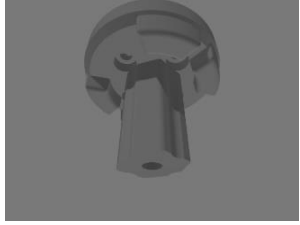
2.2.2 Εκτύπωση εξαρτημάτων

Για την κατασκευή των εκτυπωμένων εξαρτημάτων χρησιμοποιήθηκαν συνδυαστικά οι εκτυπωτές τρισδιάστατης εκτύπωσης, Prusa i3 MK3S+ και Artillery X1. Το υλικό που χρησιμοποιήθηκε για την εκτύπωση ήταν το πλαστικό πολυγαλακτικού οξέος, γνωστό ως PLA. Ο συνολικός αριθμός των τμημάτων που αποτελούν την κατασκευή του Kaya Robot ανέρχεται σε δεκατέσσερα εξαρτήματα, συμπεριλαμβανομένων των δύο επιλογών στήριξης για την κάμερα Realsense, υπό γωνία ή σε ευθεία. Ο συνολικός χρόνος εκτύπωσης όλων των εξαρτημάτων ανήλθε σε 41 ώρες.

Αναλυτικός πίνακας 3D printed εξαρτημάτων

Εικόνα εξαρτήματος	Περιγραφή εξαρτήματος	Χρόνος εκτύπωσης (minutes)	Βάρος (gramms)	Κόστος
	Kaya_Motor_Bottom_Plate	570	96	3.00€
	Kaya_Battery_Terminal_Holder	10	10	0.10€
	Kaya_Battery_Holder	120	30	2.00€
	Kaya_Electronics_Bay	130	20	1.00€

	Kaya_Motor_Top_Plate (x3)	210	30	2.00€
	Kaya_Nano_Base	370	66	3.00€
	Kaya_Pusher	340	40	2.20€
	Kaya_Pusher_Mount	50	10	0.50€
	Kaya_Realsense_Bridge	170	25	1.50€
	Kaya_Realsense_Straight	70	10	1.00€
	Kaya_Realsense_Angled	70	10	1.00€

	Kaya_Terminal_Block_Cover	60	10	0.50€
	Kaya_Wheel_Hub_Outside	130	30	1.00€
	Kaya_Wheel_Hub_Inside	160	30	1.00€
Σύνολα		2460	417	19.80€

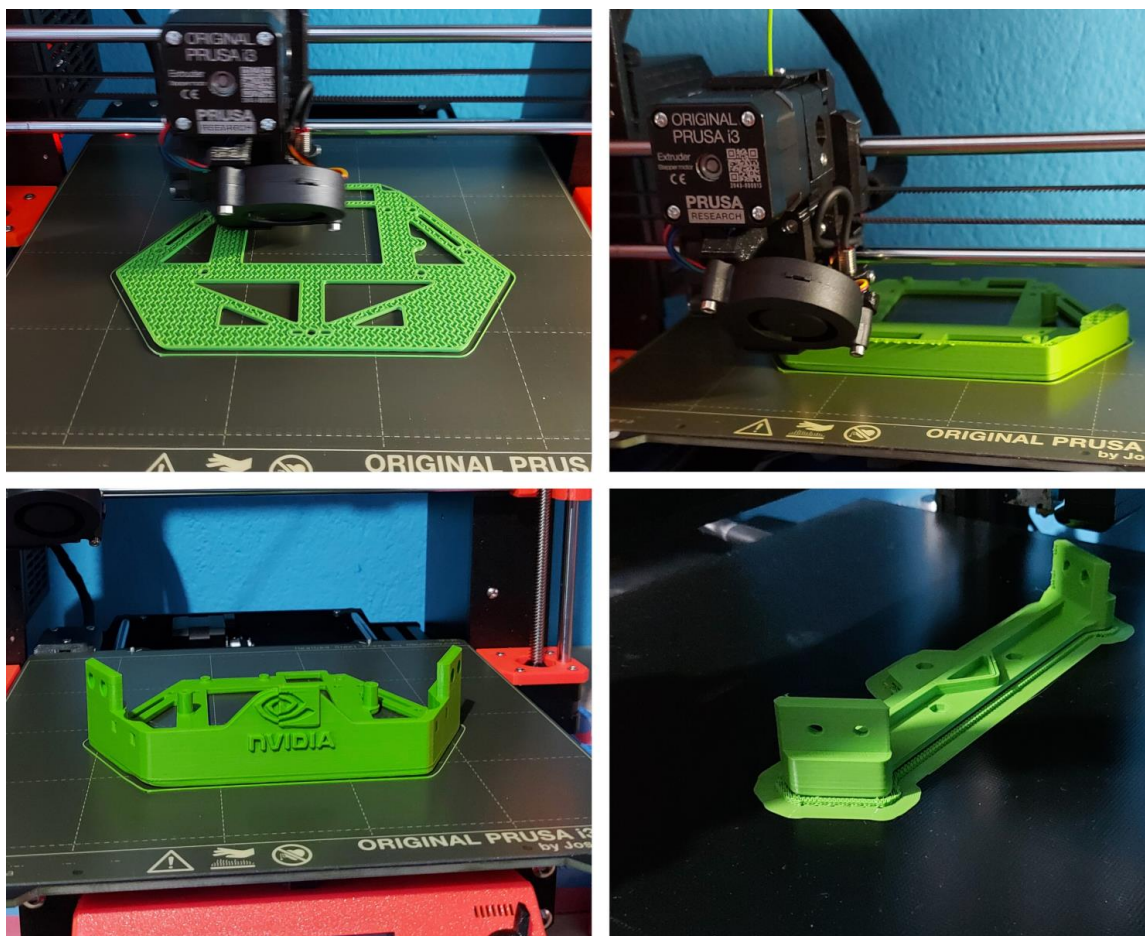


Figure 12: Στιγμιότυπα από την διαδικασία 3D εκτύπωσης

2.3 Συναρμολόγηση του Kaya Robot

2.3.1 Προετοιμασία και τοποθέτηση κινητήρων

Αρχικό βήμα για την διαδικασία συναρμολόγησης είναι ο καθορισμός της ταυτότητας ID σε κάθε κινητήρα Dynamixel με την χρήση ενός απλού συστήματος αρίθμησης. Για την συγκεκριμένη διαδικασία είναι απαραίτητη η εγκατάσταση του λογισμικού Dynamixel Wizard, το οποίο είναι διαθέσιμο στον επίσημο διαδικτυακό τόπο της εταιρείας Robotis ([σύνδεσμος λογισμικού](#)). Χρησιμοποιώντας τα συνδετικά καλώδια τροφοδοσίας και δεδομένων που παρέχονται μαζί με τους κινητήρες Dynamixel στις συσκευασίες τους, πρέπει να υλοποιηθεί η σύνδεσή τους με τον ελεγκτή U2D2, ένας κινητήρας ξεχωριστά κάθε φορά και στην συνέχεια μέσω του καλωδίου USB σε κάποιο υπολογιστή. Αφού όλες οι συνδέσεις ολοκληρωθούν κατά τον σωστό τρόπο, επόμενο βήμα είναι η εκκίνηση του λογισμικού Dynamixel Wizard, στην συνέχεια είναι απαραίτητο να οριστεί στο λογισμικό η θύρα COM στην οποία βρίσκεται συνδεδεμένο το USB που προέρχεται από το U2D2.

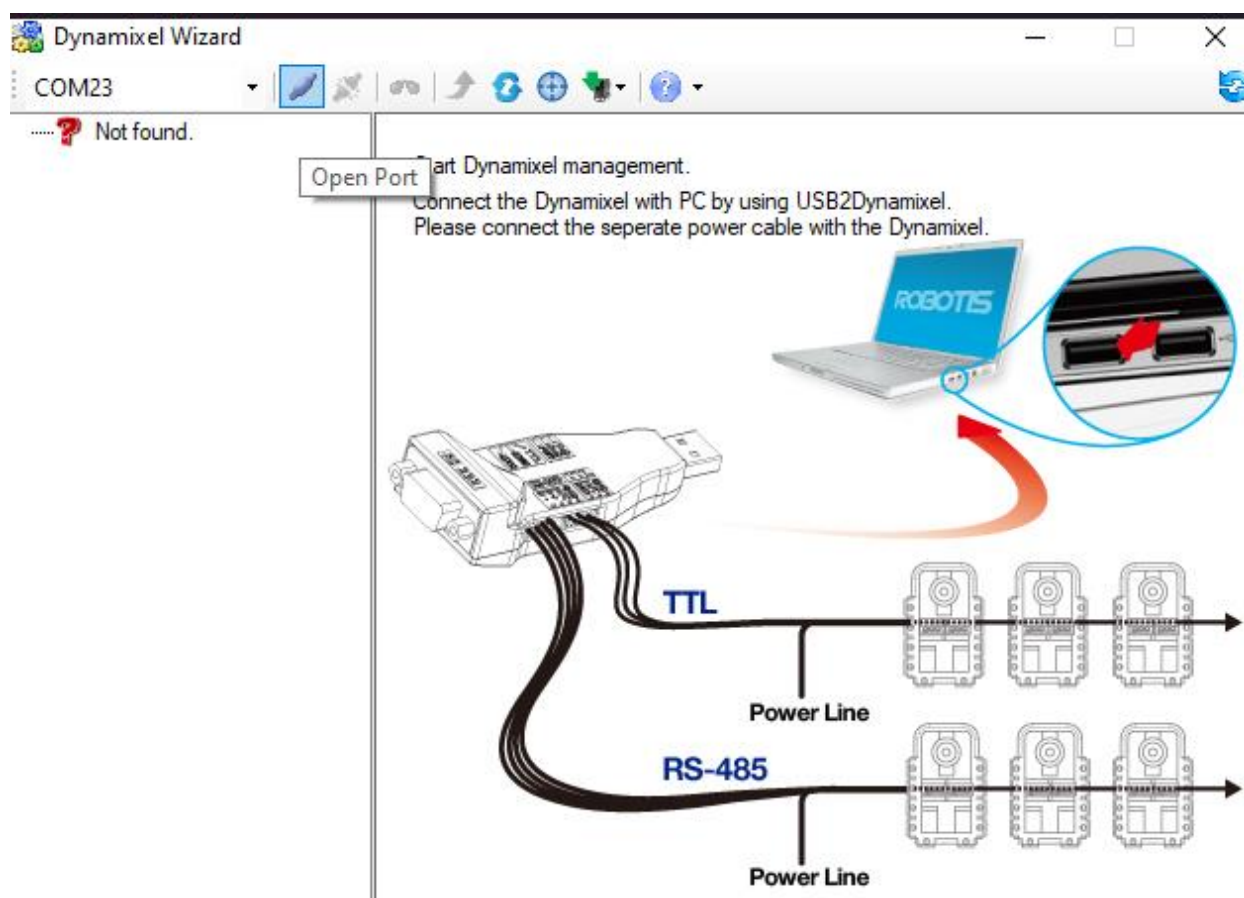


Figure 13: Άνοιγμα θύρας COM

Με την επιτυχής επιλογή της σωστής θύρας COM, ακολουθεί η εκκίνηση της διαδικασίας αναζήτησης του σερβοκινητήρα επιλέγοντας το αντίστοιχο πλήκτρο εντός του λογισμικού με την ένδειξη 'Start Searching'.

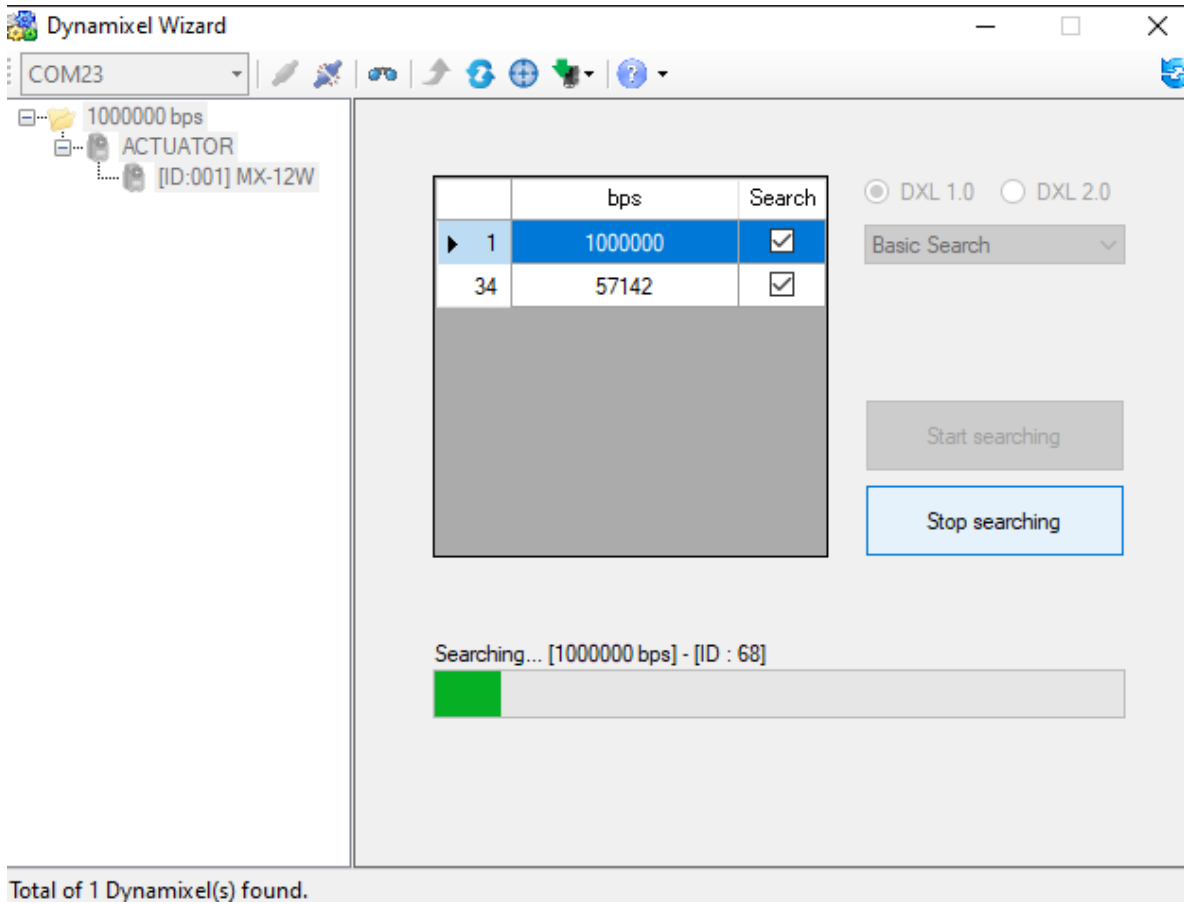


Figure 14: Αναζήτηση κινητήρων

Με την επιτυχής αναζήτηση και εύρεση του συνδεδεμένου κινητήρα, είναι πλέον διαθέσιμος μέσω του λογισμικού ώστε να παραμετροποιηθούν τα χαρακτηριστικά του ή για να πραγματοποιηθεί έλεγχος ορθής λειτουργίας του. Στο πεδίο ID, ορίζεται ο αριθμός που έχει επιλεγθεί για τον συγκεκριμένο κινητήρα. Το Kaya διαθέτει τρεις κινητήρες Dynamixel, επομένως πρέπει να γίνει η αντίστοιχη αρίθμηση των κινητήρων σε σειρά χρησιμοποιώντας τους αριθμούς 1, 2, 3. Η διαδικασία πρέπει να επαναληφθεί ακολουθώντας τα ίδια βήματα για κάθε έναν κινητήρα ξεχωριστά.

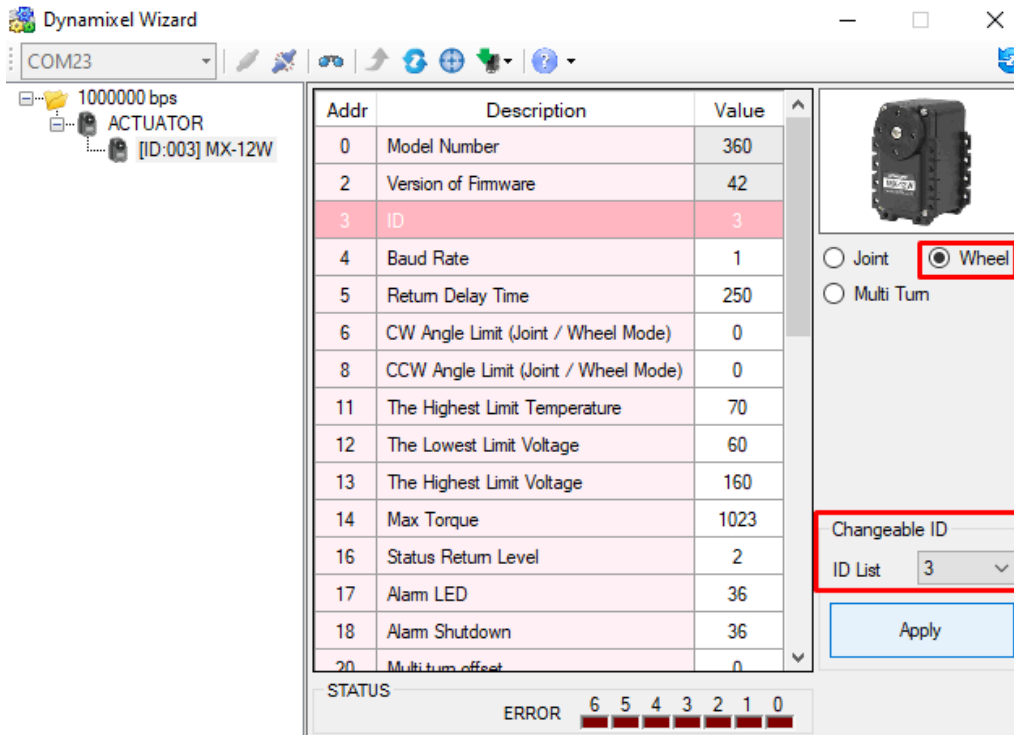


Figure 15: Παραμετροποίηση ID κινητήρα

Για την στήριξη κάθε κινητήρα απαιτείται η τοποθέτηση τεσσάρων παξιμαδιών μεγέθους M2, σε κάθε δεύτερη εσοχή από την βάση του (όπου βάση ορίζεται ως το τμήμα με τον άξονα κίνησης του κινητήρα στο κάτω μέρος), ένα σε κάθε γωνία του κινητήρα. Η διαδικασία επαναλαμβάνεται για κάθε έναν από τους τρεις κινητήρες ξεχωριστά.



Figure 16: Προετοιμασία στερέωσης κινητήρα

Με την διαδικασία προετοιμασίας των κινητήρων να έχει ολοκληρωθεί, ακολουθεί η τοποθέτησή τους στις βάσεις τους **Kaya_Motor_Bottom_Plate**. Η στερέωση τους γίνεται με την χρήση τεσσάρων βιδών τύπου M2 οι οποίες παρέχονται ως συνοδευτικά εξαρτήματα στην συσκευασία κάθε κινητήρα Dynamixel. Η διαδικασία επαναλαμβάνεται για κάθε κινητήρα.



Figure 17: Στερέωση κινητήρα στην βάση

Στην συνέχεια γίνεται η τοποθέτηση της εσωτερικής βάσης στήριξης του τροχού στον κινητήρα. Στην εσοχή της βάσης, τοποθετείται ένα παξιμάδι τύπου M3 x 18mm και στην συνέχεια γίνεται η στερέωση του στον άξονα του κινητήρα με την χρήση τεσσάρων βιδών M2 από αυτές που παρέχονται με την συσκευασία του κινητήρα. Η σύσφιξη τους πρέπει να εφαρμοστεί με διαγώνια διάταξη έτσι ώστε οι δύο επιφάνειες να εφάπτονται κατά τον σωστό τρόπο μεταξύ τους. Η διαδικασία επαναλαμβάνεται για κάθε κινητήρα.



Figure 18: Στερέωση βάσης τροχού στον κινητήρα

2.3.2 Προετοιμασία καλωδίων και ηλεκτρολογικών συνδέσεων

Για την πραγματοποίηση των ηλεκτρολογικών συνδέσεων του Kaya απαιτούνται εννέα τμήματα καλωδίων, τέσσερα μαύρου χρώματος και πέντε κόκκινου. Η κοπή των απαιτούμενων τμημάτων καθώς και η τοποθέτηση των κατάλληλων τερματικών στα άκρα τους απεικονίζεται στον παρακάτω πίνακα και την σχετική εικόνα (Figure 19).

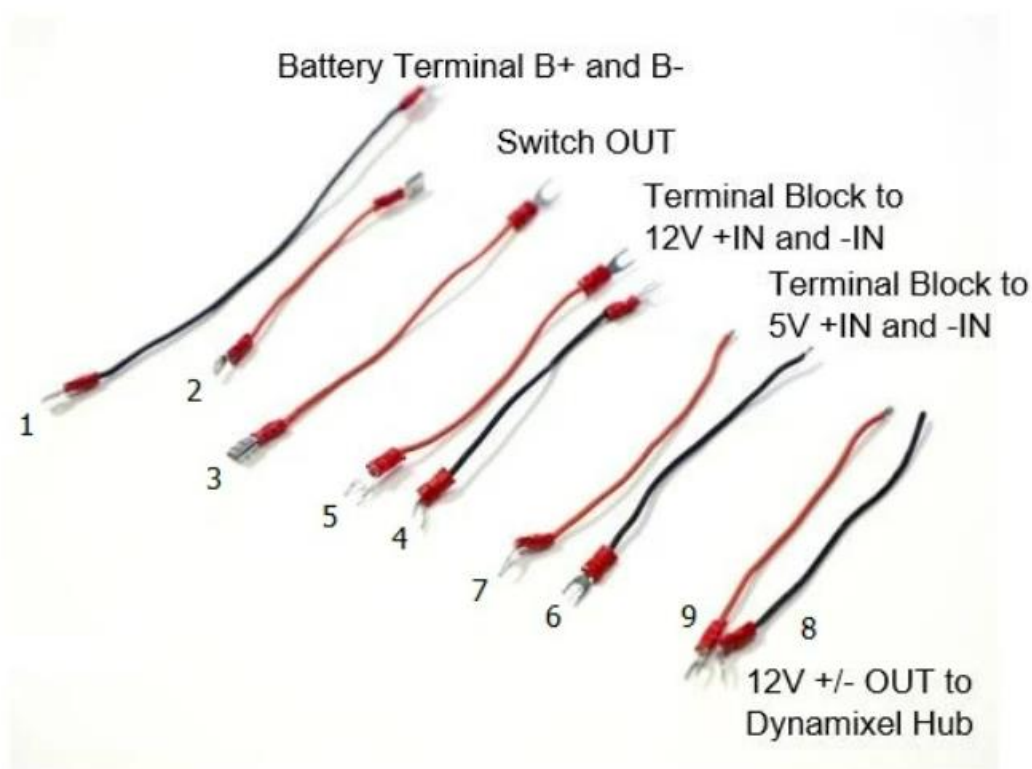


Figure 19: Περιγραφή καλωδίων

Αναλυτικός πίνακας ηλεκτρολογικών καλωδίων

#	Χρώμα	Μήκος	Σύνδεσμος τερματικού 1	Σύνδεση τερματικού 1	Σύνδεσμος τερματικού 2	Σύνδεση τερματικού 2
1	Μαύρο	140mm	Αρσενικό	Battery B-	Άγκιστρο	Τερματικό συνδέσεων B-
2	Κόκκινο	80mm	Αρσενικό	Battery B+	Γυμνό καλώδιο	Διακόπτης ON - OFF
3	Κόκκινο	140mm	Θηλυκό	Διακόπτης ON - OFF	Άγκιστρο	Τερματικό συνδέσεων B+
4	Μαύρο	120mm	Άγκιστρο	Τερματικό συνδέσεων B-	Άγκιστρο	12V Regulator -IN
5	Κόκκινο	120mm	Άγκιστρο	Τερματικό συνδέσεων B+	Άγκιστρο	12V Regulator +IN
6	Μαύρο	120mm	Άγκιστρο	Τερματικό συνδέσεων B-	Γυμνό καλώδιο	5V Regulator -IN
7	Κόκκινο	120mm	Άγκιστρο	Τερματικό συνδέσεων B+	Γυμνό καλώδιο	5V Regulator +IN
8	Μαύρο	120mm	Άγκιστρο	12V Regulator -OUT	Γυμνό καλώδιο	Dynamixel Hub -IN
9	Κόκκινο	120mm	Άγκιστρο	12V Regulator +OUT	Γυμνό καλώδιο	Dynamixel Hub +IN

Με την ολοκλήρωση της προετοιμασίας των επιμέρους καλωδίων σειρά έχει η τοποθέτησή τους. Τα καλώδια #1 και #2 τοποθετούνται στις εγκοπές του εξαρτήματος **Kaya_Battery_Holder** και στην συνέχεια ασφαλιζονται με την χρήση του εξαρτήματος συγκράτησής τους **Kaya_Battery_Terminal_Holder** χρησιμοποιώντας μία βίδα τύπου M3 x 8mm. Σε αυτό το σημείο είναι σημαντικό οι ακροδέκτες των καλωδίων να ευθυγραμμιστούν με τις εσοχές της μπαταρίας ώστε η τοποθέτηση και αφαίρεση της μπαταρίας να πραγματοποιείται με τον βέλτιστο τρόπο.

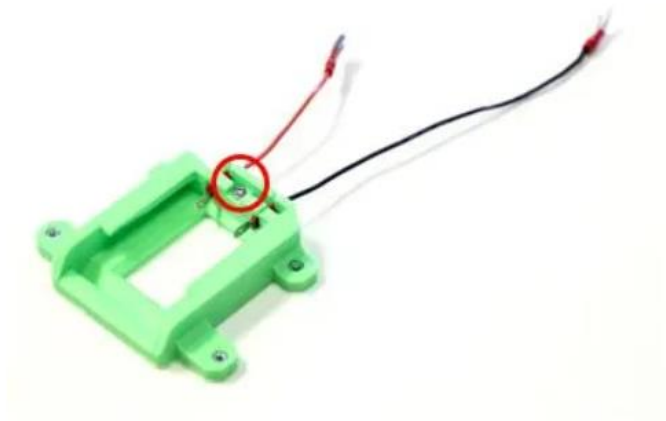


Figure 20: Τοποθέτηση καλωδίων τροφοδοσίας στην βάση μπαταρίας

2.3.3 Συναρμολόγηση σκελετού

Για την συναρμολόγηση του κεντρικού σκελετού του Kaya Robot απαιτούνται τα τρία έδρανα των κινητήρων **Kaya_Motor_Top_Plate** και τα προφίλ αλουμινίου **MakerBeams**. Απαραίτητα επίσης είναι τα παξιμάδια τύπου **T - Slot** και τέσσερις βίδες **M3 x 5mm** για κάθε ένα από τα τρία άκρα του σκελετού. Αφού τοποθετηθούν τα παξιμάδια **T - Slot** στις δύο πλευρές των προφίλ αλουμινίου, το προφίλ αλουμινίου τοποθετείται στις εσοχές του εδράνου και στην συνέχεια γίνεται η σύσφιξη του με τις δύο βίδες σε κάθε άκρο του. Η διαδικασία επαναλαμβάνεται για τα υπόλοιπα έδρανα για τα δύο άκρα που απομένουν.



Figure 21: Έδρανα κινητήρων & MakerBeams

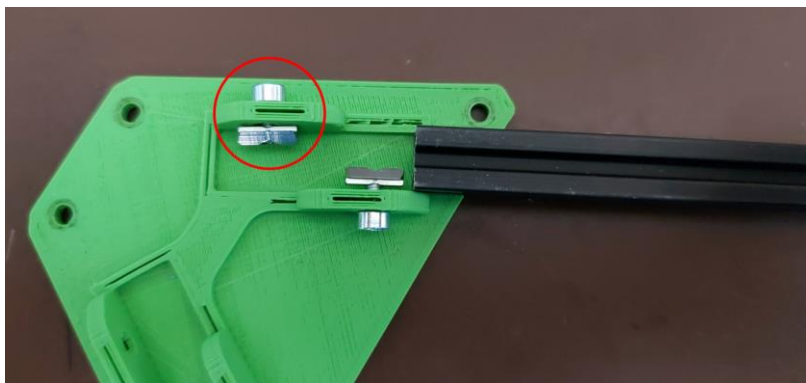


Figure 22: Τοποθέτηση προφίλ MakerBeam

Με την ολοκλήρωση του κεντρικού τμήματος του σκελετού τοποθετείται στο εμπρός τμήμα του η βάση του προωθητήρα του ρομπότ **Kaya_Pusher_Mount** με την χρήση δύο βιδών M3 x 5mm.



2.3.4 Τοποθέτηση κινητήρων στον σκελετό

Ακολουθεί η τοποθέτηση των κινητήρων που έχουν προετοιμαστεί, με τις βάσεις τους, **Kaya_Motor_Bottom_Plate** στον σκελετό του ρομπότ ακολουθώντας την σωστή διάταξη με βάση την αρίθμησή τους. Ο κινητήρας με ID2 πρέπει να τοποθετηθεί στο οπίσθιο άκρο του ρομπότ, ο κινητήρας με ID1 στο εμπρός δεξί άκρο και ο κινητήρας με ID3 στο εμπρός αριστερό άκρο που απομένει. Η μεταξύ σύσφιξη των δύο βάσεων κάθε κινητήρα γίνεται με τέσσερα παξιμάδια M3 και τέσσερις βίδες M3 x 8mm για κάθε έναν.



Figure 23: Διάταξη κινητήρων

Με τους κινητήρες τοποθετημένους στον σκελετό, πραγματοποιείται η σύνδεση μεταξύ τους με τα συνδετικά καλώδια που παρέχονται στην συσκευασία τους έτσι ώστε να διαμοιράζεται η απαραίτητη ισχύς και τα δεδομένα για την λειτουργία τους. Μπορούν να συνδεθούν υπό οποιαδήποτε σειρά εξυπηρετεί με ικανοποιητικότερο τρόπο την χωροταξική διάταξη των καλωδίων.



Figure 24: Σύνδεση κινητήρων

2.3.5 Τοποθέτηση IMU και μετατροπών τάσεως

Με την χρήση του εξαρτήματος **Kaya_Electronics_Bay** ως βάση, τοποθετείται η μονάδα IMU με δύο βίδες M2 x 6mm με τους ακροδέκτες να έχουν φορά προς το κέντρο της βάσης. Στην συνέχεια, τοποθετούνται οι μετατροπές τάσεως 5V και 12V με την χρήση πέντε βιδών M2 x 6mm.

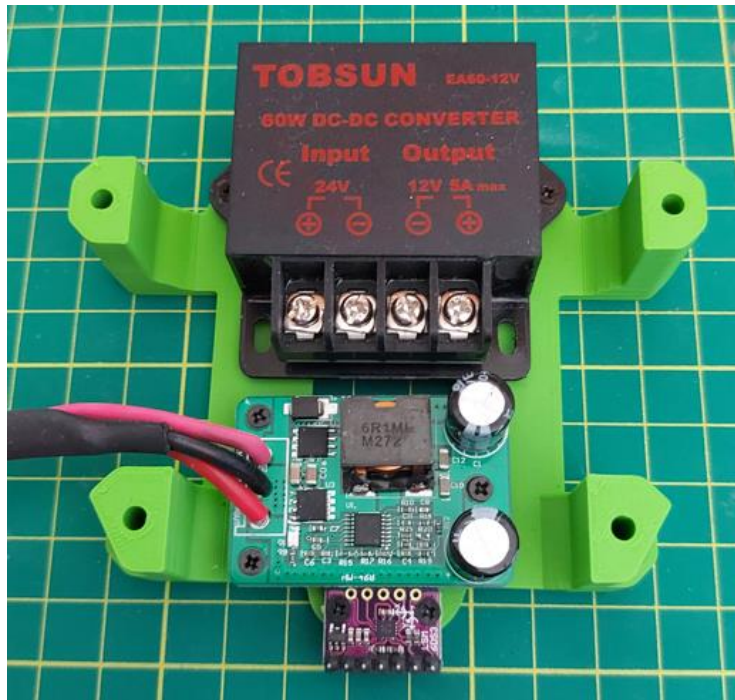


Figure 25: Τοποθέτηση IMU & μετατροπών τάσης

2.3.6 Τοποθέτηση κεντρικής βάσης Jetson Nano

Ως προετοιμασία, τοποθετούνται στα προφίλ αλουμινίου του σκελετού τα παξιμάδια τύπου T - Slot, δύο σε κάθε πλευρικό προφίλ και ένα στο προφίλ του μπροστινού τμήματος.

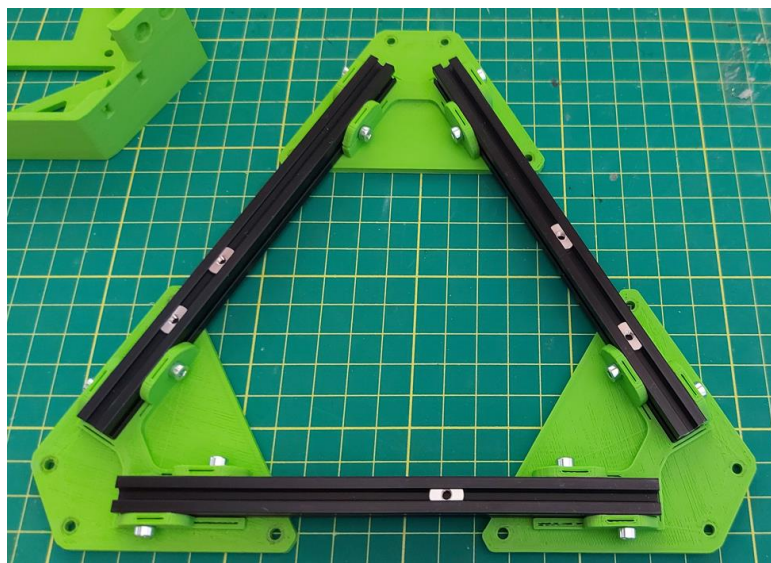


Figure 26: Κεντρικός σκελετός

Στην συνέχεια η κεντρική βάση του ρομπότ που αποτελεί και σημείο στήριξης του Jetson Nano τοποθετείται στον σκελετό χρησιμοποιώντας πέντε βίδες M3 x 5mm. Ταυτόχρονα με την χρήση τεσσάρων βιδών M3 x 35mm τοποθετούνται στην κεντρική βάση, η βάση των επιμέρους ηλεκτρονικών τμημάτων **Kaya_Electronics_Bay** και η βάση της μπαταρίας **Kaya_Battery_Holder**.

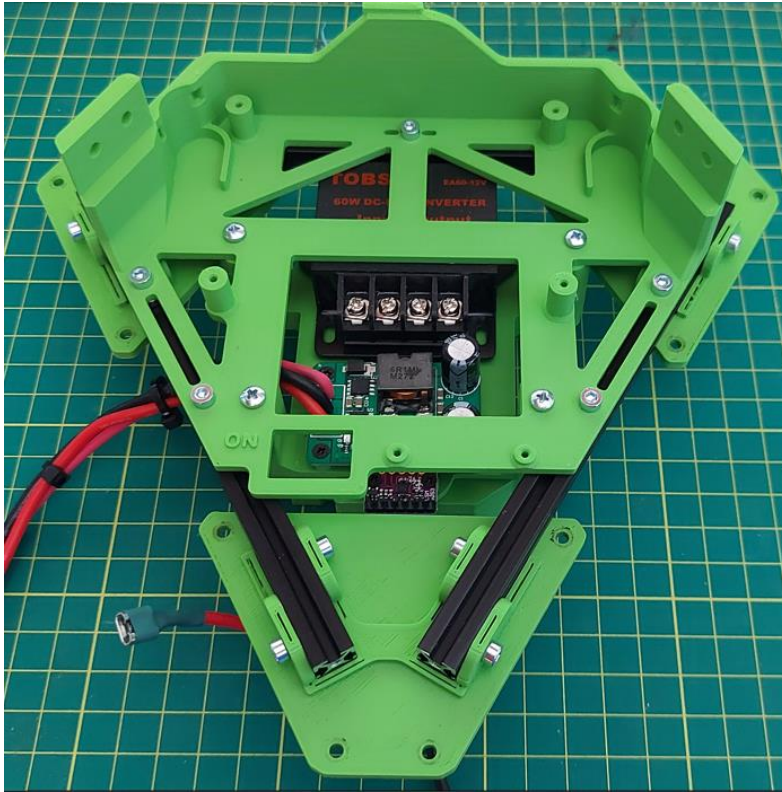


Figure 27: Στερέωση βάσης Jetson

Στην συνέχεια τοποθετείται ο διακόπτης τύπου ON – OFF στην αντίστοιχη εσοχή της κεντρικής βάσης καθώς και η μονάδα διανομής ενέργειας και δεδομένων των κινητήρων Dynamixel.

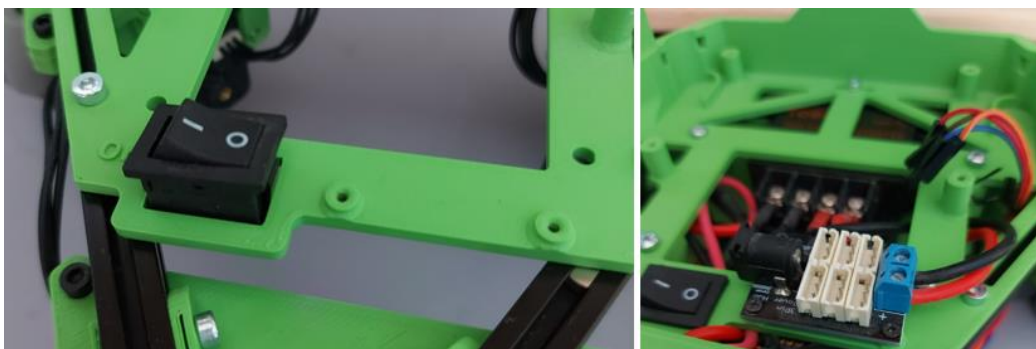


Figure 28: Τοποθέτηση κεντρικού διακόπτη & Dynamixel Power Hub

Στο τερματικό ηλεκτρολογικών συνδέσεων τοποθετούνται δύο ηλεκτρολογικές γέφυρες (jumper wires) και στην συνέχεια στερεώνεται στο πίσω τμήμα του σκελετού με δύο βίδες M3 x 10mm.

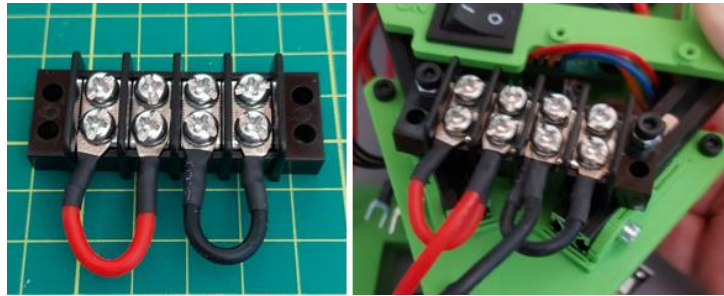


Figure 29: Τοποθέτηση τερματικού συνδέσεων

2.3.7 Τοποθέτηση Jetson Nano

Αφού αφαιρεθεί το κεντρικό module επεξεργασίας του Jetson Nano στερεώνεται η κάρτα δικτύου στην κατάλληλη εσοχή και ασφαρίζεται με την βίδα που βρίσκεται στο σημείο σύσφιξης. Τα καλώδια για τις κεραιές δικτύου και Bluetooth τοποθετούνται στην κάρτα. Στην συνέχεια το Jetson Nano στερεώνεται στην κεντρική βάση με τέσσερις βίδες M2 x 6mm.

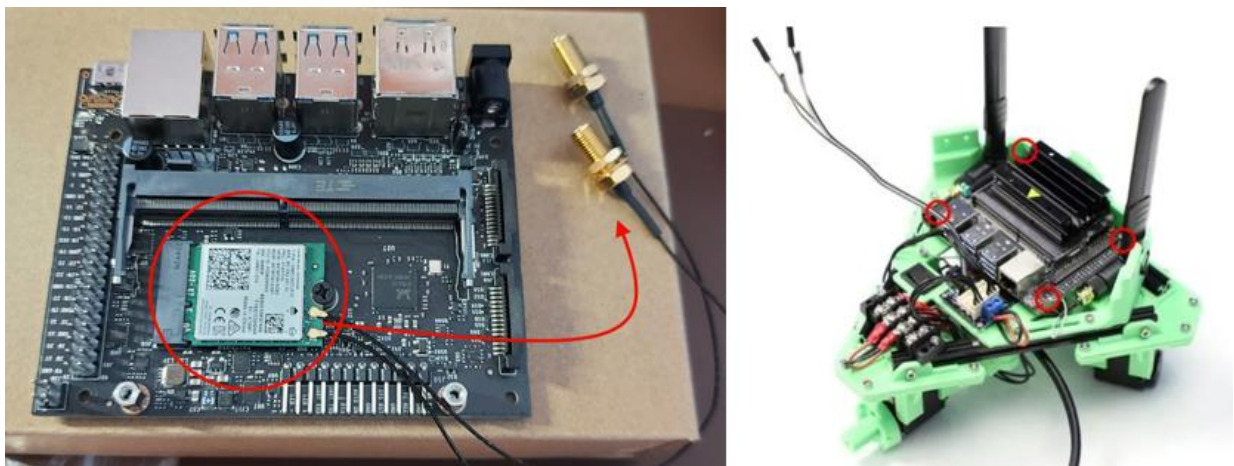


Figure 30: Τοποθέτηση κάρτας δικτύου και στερέωση Jetson Nano

2.3.8 Τοποθέτηση κάμερας Realsense

Με χρήση του εξαρτήματος **Kaya_Realsense_Mount** και δύο βιδών M3 x 5mm στερεώνεται η κάμερα Realsense. Στην συνέχεια η κάμερα μαζί με την βάση της στερεώνεται στο εξάρτημα **Kaya_Realsense_Bridge** με χρήση τριών βιδών M3 x 8mm. Το τμήμα του Bridge τοποθετείται στην υπόλοιπη κατασκευή του ρομπότ με τέσσερις βίδες M3 x 8mm.



Figure 31: Στερέωση κάμερας RealSense

2.3.9 Τοποθέτηση τροχών κίνησης

Ο κάθε τροχός τοποθετείται στην βάση έδρασης **Kaya_Wheel_Hub_Inside** που ήδη έχει εγκατασταθεί στον άξονα κάθε κινητήρα στα προηγούμενα βήματα, χρησιμοποιώντας μια βίδα M3 x 35mm για κάθε τροχό και ένα εξάρτημα **Kaya_Wheel_Hub_Outside** για την σύσφιξη και την ασφάλιση της στερέωσής του.



Figure 32: Τοποθέτηση τροχών κίνησης

2.3.10 Τελικές ηλεκτρολογικές συνδέσεις

Με την βασική κατασκευή ολοκληρωμένη όλα τα τμήματα απαιτείται να συνδεθούν μεταξύ τους με τις προκαθορισμένες ηλεκτρολογικές συνδέσεις. Αυτό γίνεται ακολουθώντας το ηλεκτρολογικό διάγραμμα του Kaya.

- Το τελικό καλώδιο από την συστοιχία των κινητήρων συνδέεται σε οποιαδήποτε θύρα του Dynamixel hub, αυτό στην συνέχεια συνδέεται στο U2D2 όπου καταλήγει μέσω σύνδεσης USB στο Jetson Nano. Στην είσοδο τροφοδοσίας του Dynamixel Hub συνδέονται τα καλώδια από την έξοδο του μετατροπέα τάσης 12V στους αντίστοιχους θετικούς και αρνητικούς πόλους.
- Η έξοδος του μετατροπέα τάσης 5V συνδέεται μέσω του βύσματος τροφοδοσίας τύπου barrel jack στην είσοδο τροφοδοσίας του Jetson Nano. Είναι σημαντικό το Jumper J48 να συνδέει τους ακροδέκτες στο Jetson Nano ώστε να γνωρίζει ότι του παρέχεται τροφοδοσία μέσω εξωτερικής πηγής και όχι μέσω καλωδίου USB.



Figure 33: Ρύθμιση τροφοδοσίας Jetson Nano

- Η μπαταρία συνδέεται στο τερματικό ηλεκτρολογικών συνδέσεων στους θετικούς και αρνητικούς ακροδέκτες αντιστοίχως. Στο καλώδιο του θετικού ακροδέκτη μεσολαβεί ο κεντρικός διακόπτης τροφοδοσίας.
- Η μονάδα IMU συνδέεται ως εξής:
 - Ακροδέκτης 3V στον ακροδέκτη #1 από την θύρα GPIO του Jetson Nano.
 - Ακροδέκτης GND στον ακροδέκτη #6 από την θύρα GPIO του Jetson Nano.
 - Ακροδέκτης SCL στον ακροδέκτη #5 από την θύρα GPIO του Jetson Nano.
 - Ακροδέκτης SDA στον ακροδέκτη #3 από την θύρα GPIO του Jetson Nano.

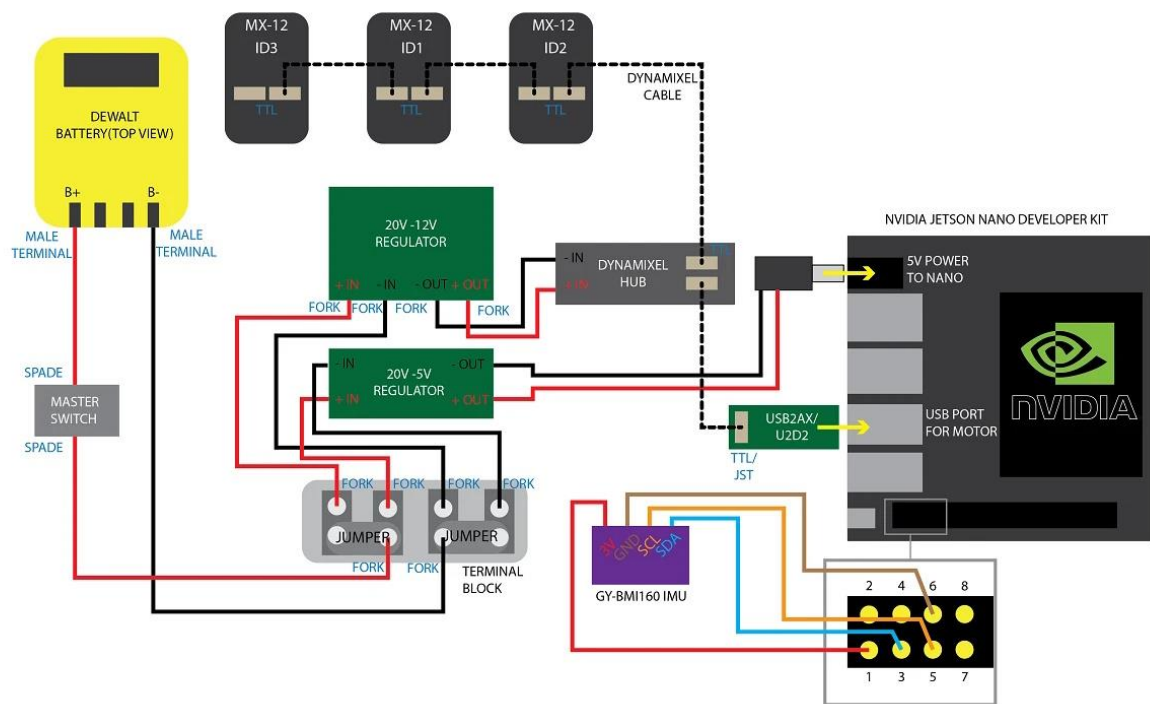


Figure 34: Ηλεκτρολογικό διάγραμμα Kaya Robot

Σε αυτό το στάδιο έχει ολοκληρωθεί η συναρμολόγηση του Kaya.



Figure 35: Ολοκλήρωση συναρμολόγησης Kaya Robot

2.4 Εγκατάσταση λογισμικού

2.4.1 Εγκατάσταση Jetpack SDK στο Jetson Nano

Για την λειτουργία του Jetson Nano απαιτείται η εγκατάσταση ενός λειτουργικού συστήματος. Όπως αναφέρθηκε η NVIDIA έχει δημιουργήσει το Jetpack SDK για αυτό τον σκοπό. Στην συγκεκριμένη υλοποίηση χρησιμοποιήθηκε η έκδοση 4.3 του Jetpack. Αφού ολοκληρωθεί η λήψη του από τον διαδικτυακό τόπο της NVIDIA απαιτείται η εγγραφή του στην κάρτα microSD. Αυτή η διαδικασία υλοποιείται με την χρήση του προγράμματος Balena Etcher.

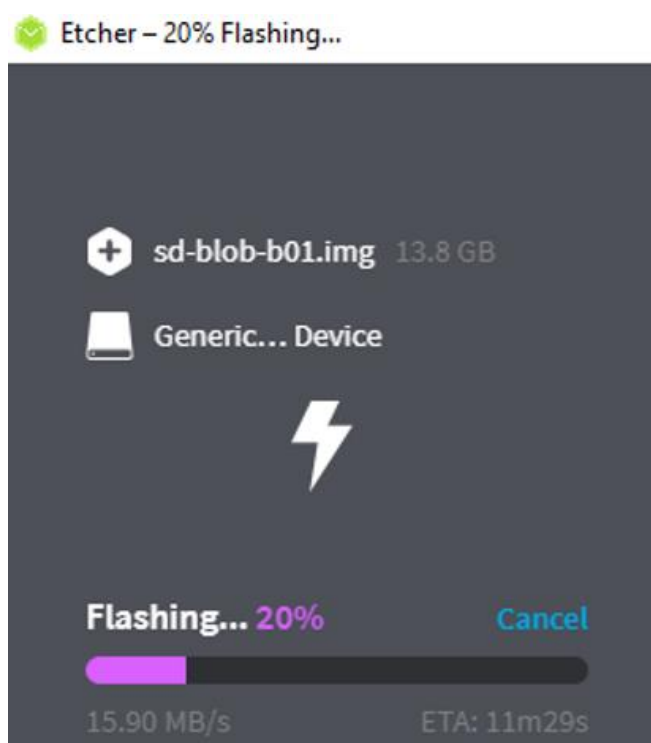


Figure 36: Εγγραφή λειτουργικού Jetpack

Με την ολοκλήρωση της εγγραφής, η κάρτα πρέπει να τοποθετηθεί στο Jetson Nano και να συνδεθεί σε μία πηγή τροφοδοσίας και μία οθόνη. Ακολουθώντας τα βήματα που εμφανίζονται στην οθόνη, ολοκληρώνεται η εγκατάσταση του λειτουργικού στο Jetson.

2.4.2 Εγκατάσταση Isaac SDK στο σύστημα Host

Για την λειτουργία του NVIDIA Isaac SDK είναι προαπαιτούμενο ένα σύστημα Ubuntu για να εγκατασταθεί. Το NVIDIA Isaac λειτουργεί με την έκδοση Ubuntu 18.04. Για την εγκατάσταση απαιτείται η λήψη της επιθυμητής έκδοσης του Isaac από τον διαδικτυακό τόπο της NVIDIA. Στην συγκεκριμένη υλοποίηση χρησιμοποιήθηκε η έκδοση 2019.03.

Αφού ολοκληρωθεί η λήψη απαιτείται η δημιουργία ενός νέου directory. Αυτό γίνεται με την εντολή:

```
user@host:~/$ mkdir isaac
```

Στην συνέχεια το ληφθέν συμπιεσμένο αρχείο πρέπει να τοποθετηθεί στο παραπάνω directory και να αποσυμπιεστεί με την χρήση της εντολής:

```
user@host:~/isaac$ tar xf <isaac filename>
```

Αφού ολοκληρωθεί η εξαγωγή των αρχείων εγκαθίστανται τα απαραίτητα dependencies με χρήση της εντολής:

```
user@host:~/isaac$ engine/build/scripts/install_dependencies.sh
```

2.4.3 Εγκατάσταση dependencies στο Jetson Nano

Για την λειτουργία του Isaac SDK απαιτείται η εγκατάσταση των απαραίτητων dependencies σε αυτό. Αυτό γίνεται από το σύστημα host με χρήση της εντολής:

```
user@host:~/isaac$ engine/build/scripts/install_dependencies_jetson.sh -u <jetson_username> -h <jetson_ip>
```

3 Εκτέλεση εφαρμογών

3.1 Εφαρμογή Joystick

Η εφαρμογή πρέπει να μεταφερθεί από το σύστημα Host στο Jetson Nano (deployment). Για να ολοκληρωθεί η συγκεκριμένη διαδικασία χρησιμοποιείται η εντολή:

```
user@host:~/isaac$ ./engine/build/deploy.sh --remote_user <username_on_robot> -p //apps/kaya:joystick-pkg -d jetpack43 -h <robot_ip>
```

Στην συνέχεια πρέπει να δημιουργηθεί μια σύνδεση με το ρομπότ μέσω SSH για την εκτέλεση της εφαρμογής:

```
user@host:~/isaac$ ssh <username_on_robot>@<robot_ip>
```

Με την πραγματοποίηση της σύνδεσης ακολουθεί η εκτέλεση της εφαρμογής, αφού πρώτα γίνει η μετάβαση στο σχετικό directory.

```
kaya@jetson:~//$ cd deploy/<bob>/joystick-pkg/  
kaya@jetson:~/deploy/user/joystick-pkg$ ./apps/kaya/joystick
```

Μόλις η εφαρμογή εκτελεστεί επιτυχώς, από σύστημα Host πρέπει να γίνει εκκίνηση του Isaac Sight για τον έλεγχο του ρομπότ. Αυτό γίνεται μέσω ενός Web Browser ανατρέχοντας στην διεύθυνση IP του ρομπότ και την θύρα (port) 3000:

`<robot_ip>:3000`

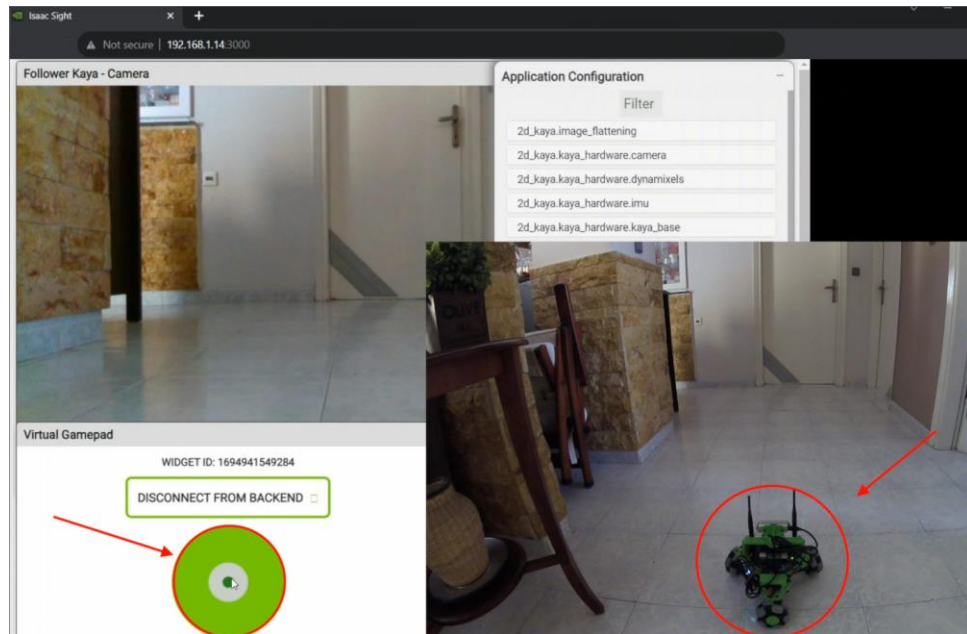


Figure 37: Εκτέλεση εφαρμογής Joystick (1)

Για την εξοικονόμηση κόστους, για την εκτέλεση της συγκεκριμένης εφαρμογής χρησιμοποιήθηκε ένα εικονικό Joystick για την κίνηση του ρομπότ που παρέχει το Isaac Sight. Εναλλακτικά μπορούν να χρησιμοποιηθούν ασύρματα χειριστήρια κόνσολών παιχνιδιού που διαθέτουν συνδεσιμότητα Bluetooth. Μετακινώντας το εικονικό χειριστήριο μέσω του κέρσορα τ ρομπότ κινείται προς την επιθυμητή κατεύθυνση.

Η λειτουργία της εφαρμογής απεικονίζεται στο συνοδευτικό αρχείο βίντεο: ***kaya_virtual_gamepad.mp4***

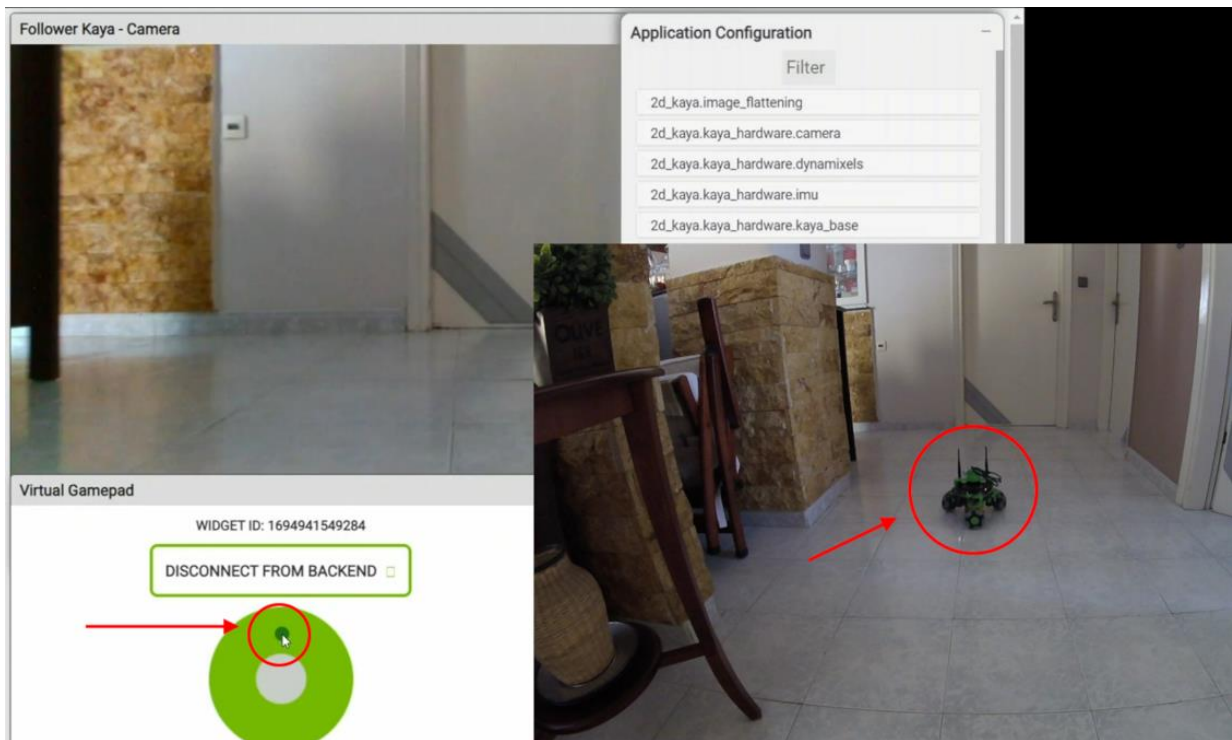


Figure 38: Εκτέλεση εφαρμογής Joystick (2)

3.2 Εφαρμογή Follow Me

Ομοίως ακολουθείται η διαδικασία για την εκτέλεση της εφαρμογής Follow Me, κατά την οποία με την χρήση της κάμερας Realsense το Kaya έχει την δυνατότητα να εντοπίζει κωδικούς AprilTags και να κινείται προς αυτά. Τα AprilTags είναι οπτικοί σηματοδότες τύπου κωδικών QR και χρησιμοποιούνται ευρύτερα για την ανίχνευση αντικειμένων, εύρεση θέσης και ως στόχοι αναφοράς για την βαθμονόμηση καμερών. Το Isaac SDK υποστηρίζει το πρότυπο Tag36h11 για την ενσωμάτωση των AprilTags.

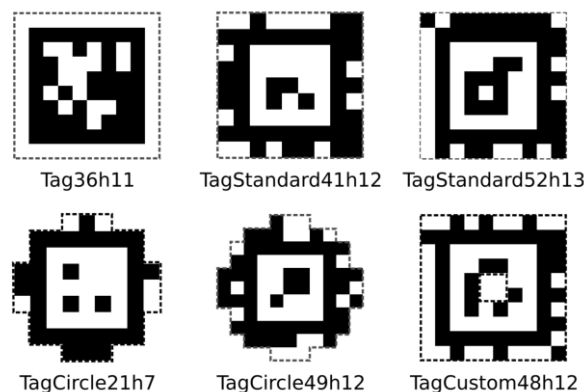


Figure 39: Πρότυπα AprilTags

Για την μεταφορά της εφαρμογής από το σύστημα Host στο Jetson Nano (deployment) ακολουθείται η εντολή:

```
user@host:~/isaac$ ./engine/build/deploy.sh --remote_user <username_on_robot> -p //apps/kaya:follow_me-pkg -d jetpack43 -h <robot_ip>
```

Στην συνέχεια πρέπει να δημιουργηθεί μια σύνδεση με το ρομπότ μέσω SSH για την εκτέλεση της εφαρμογής:

```
user@host:~/isaac$ ssh <username_on_robot>@<robot_ip>
```

Με την πραγματοποίηση της σύνδεσης ακολουθεί η εκτέλεση της εφαρμογής, αφού πρώτα γίνει η μετάβαση στο σχετικό directory.

```
kaya@jetson:~//$ cd deploy/user/joystick-pkg/  
kaya@jetson:~/deploy/user/follow_me-pkg$ ./apps/kaya/follow_me
```

Μόλις η εφαρμογή εκτελεστεί επιτυχώς, από σύστημα Host πρέπει να γίνει εκκίνηση του Isaac Sight για τον έλεγχο του ρομπότ. Αυτό γίνεται μέσω ενός Web Browser ανατρέχοντας στην διεύθυνση IP του ρομπότ και την θύρα (port) 3000:

```
<robot_ip>:3000
```

Με την εκκίνηση της εφαρμογής στις παραμέτρους της πρέπει να οριστεί το ID της ετικέτας που έχει τοποθετηθεί στον χώρο.

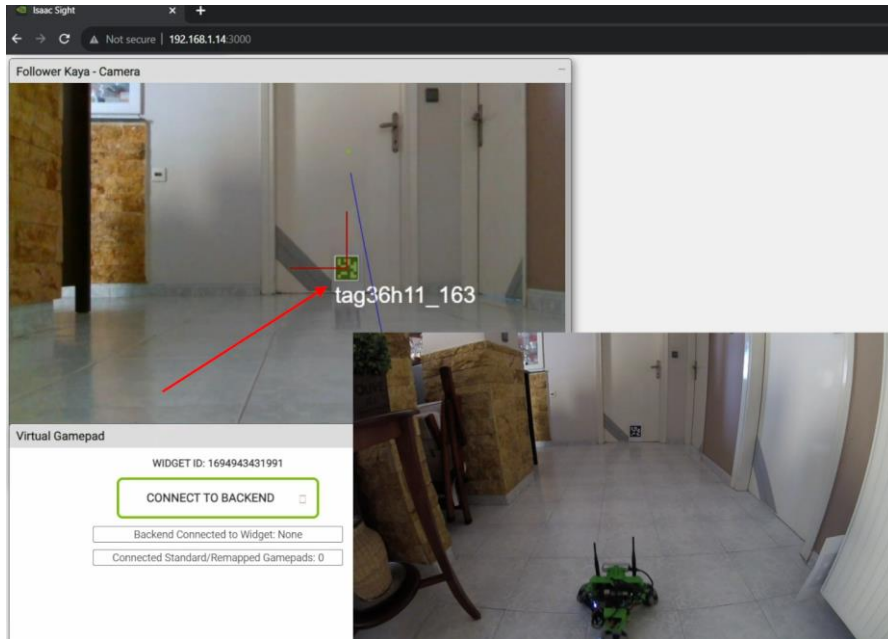


Figure 40: Εκτέλεση εφαρμογής Follow Me (1)

Στην συνέχεια, αν η ετικέτα βρίσκεται στο οπτικό πεδίο του ρομπότ γίνεται η ανίχνευσή της από αυτό και σχεδιάζεται η διαδρομή προς αυτή, η οποία στην συνέχεια ακολουθείται για να προσεγγίσει την ετικέτα.

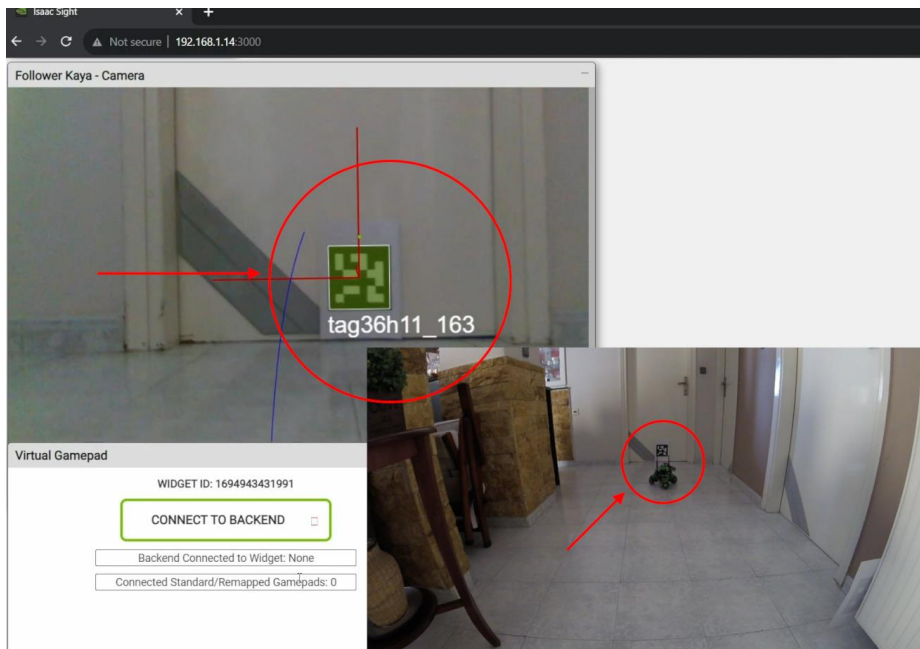


Figure 41: Εκτέλεση εφαρμογής Follow Me (2)

Η λειτουργία της εφαρμογής απεικονίζεται στο συνοδευτικό αρχείο βίντεο: **kaya_follow_me.mp4**

3.3 Εφαρμογή Gmapping

Η εφαρμογή Gmapping χρησιμοποιείται για την δημιουργία ενός χάρτη του χώρου του οποίου κινείται το Kaya Robot. Με την χρήση της κάμερας RealSense λαμβάνονται δεδομένα για το βάθος των εμποδίων και με την χρήση της οδομετρίας των τροχών και της μονάδας IMU καταγράφεται η πορεία που χαράζεται στον χώρο. Στην συνέχεια, τα δεδομένα αυτά αποστέλλονται στο σύστημα host στο οποίο δημιουργείται ο τελικός χάρτης του χώρου.

Για την εγκατάσταση και την εκτέλεση της εφαρμογής απαιτούνται δύο βήματα. Η εγκατάσταση του πακέτου `gmapping_distributed_kaya`, το οποίο εκτελείται στο ρομπότ και η εγκατάσταση του πακέτου `gmapping_distributed_host`, το οποίο εκτελείται στο σύστημα host.

Για την εγκατάσταση του πακέτου στο ρομπότ, στο σύστημα host χρησιμοποιείται η εντολή:

```
kaya@jetson:~/isaac$ ./engine/build/deploy.sh --remote_user <username_on_robot> -p  
//apps/kaya:gmapping_distributed_kaya-pkg -d jetpack43 -h <robot_ip>
```

Στην συνέχεια το τμήμα της εφαρμογής στο ρομπότ είναι έτοιμο για εκτέλεση με την χρήση των εντολών:

```
cd deploy/user/gmapping_distributed_kaya-pkg/  
./apps/kaya/gmapping_distributed_kaya
```

Ακολουθώντας την διεύθυνση `<nano_ip>:3000` σε έναν περιηγητή ιστού στο σύστημα host, εκκινεί το Isaac Sight στο οποίο απεικονίζεται η εικόνα βάθους μέσω της RGB κάμερας του ρομπότ και η οδομετρία του.

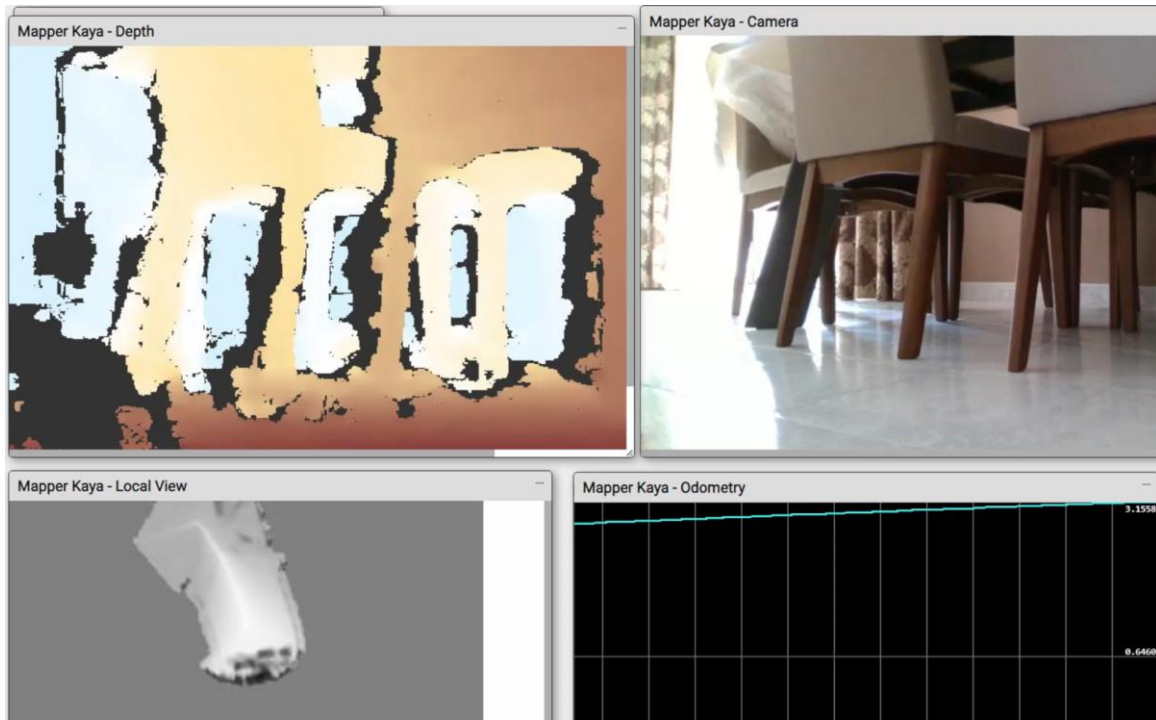


Figure 42: Απεικόνιση εικόνας βάθους στο Kaya Robot

Στο σύστημα host, είναι απαραίτητο να τροποποιηθεί το πεδίο 'tcp-subscriber' στο αρχείο 'gmapping_distributed_host.app.json'. Το συγκεκριμένο αρχείο βρίσκεται στο directory 'app/kaya/gmapping_distributed_host.app.json'. Η επεξεργασία του μπορεί να γίνει με κάποια εφαρμογή κειμενογράφου και το πεδίο 'tcp-subscriber' πρέπει να οριστεί στην διεύθυνση IP του ρομπότ.

Στην συνέχεια μπορεί να γίνει η εγκατάσταση και η εκτέλεση της εφαρμογής στο σύστημα host με την χρήση των εντολών:

```
bazel build //apps/kaya:gmapping_distributed_host  
bazel run //apps/kaya:gmapping_distributed_host
```

Στην διεύθυνση localhost:3000 του συστήματος host, εκκινεί το Isaac Sight και γίνεται λήψη των δεδομένων από το ρομπότ για την δημιουργία του χάρτη του περιβάλλοντός του. Η κατασκευή του χάρτη γίνεται ορατή σε πραγματικό χρόνο.

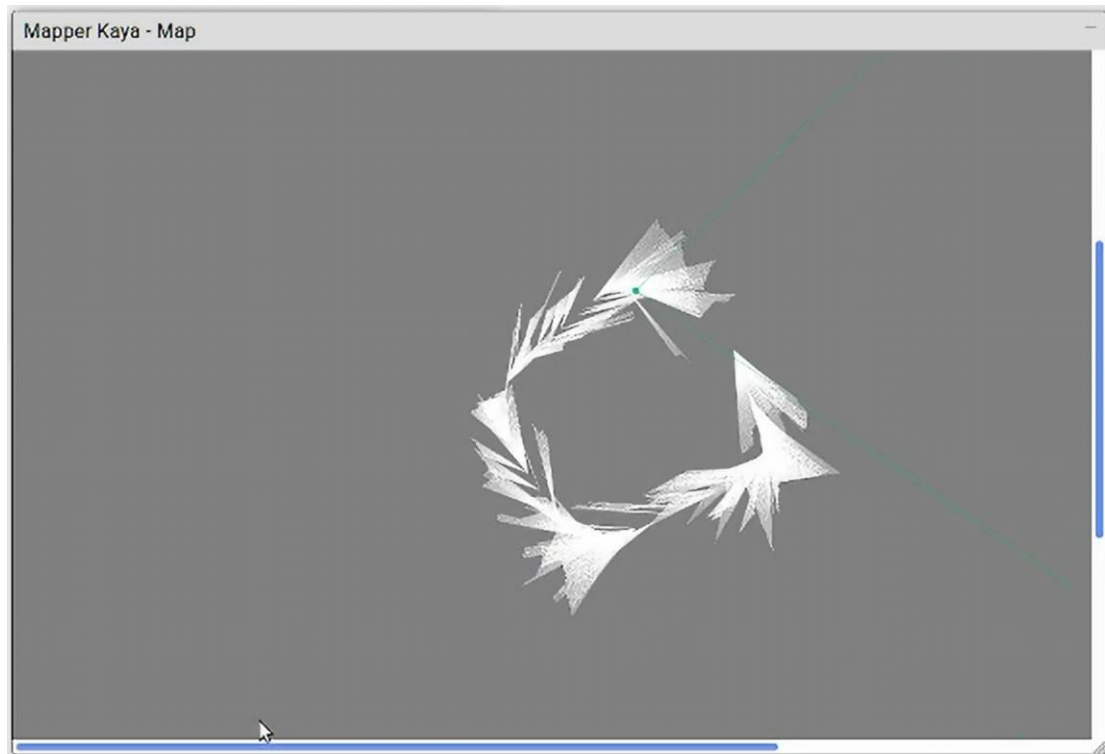


Figure 43: Κατασκευή χάρτη στο σύστημα Host

Η λειτουργία της εφαρμογής απεικονίζεται στο συνοδευτικό αρχείο βίντεο: ***kaya_gmapping.mp4***

4 Συμπεράσματα και μελλοντική εξέλιξη

4.1 Δυσκολίες κατά την κατασκευή του Kaya Robot

Κατά την πορεία κατασκευής του Kaya Robot αντιμετωπίστηκαν διάφορες δυσκολίες οι οποίες έπρεπε να επιλυθούν για την ολοκλήρωση της τελικής κατασκευής του ρομπότ. Κάποιες από αυτές ήταν:

- Ελλιπής βιβλιογραφία και documentation στους ιστότοπους της NVIDIA
- Ανεπαρκής υποστήριξη μέσω των NVIDIA developer forums
- Ασυμβατότητα των εκδόσεων του Isaac SDK με τις εκδόσεις του Jetpack SDK
- Ανάγκη για επανασχεδιασμό κάποιο τμημάτων των 3D parts

Ως μέθοδος επίλυσης των προβλημάτων χρησιμοποιήθηκαν οι επαναλαμβανόμενες δοκιμές διαφορετικών εκδόσεων του λογισμικού Isaac SDK και του λειτουργικού συστήματος Jetpack.

4.2 Προτάσεις για μελλοντική εξέλιξη

Παρά την παύση εξέλιξης και υποστήριξης του Isaac SDK από την NVIDIA, η κατασκευή του Kaya Robot παραμένει ένα πολύ ικανό σύνολο εξαρτημάτων και αισθητήρων το οποίο προσφέρεται και ανάπτυξη ρομποτικών εφαρμογών.

4.2.1 Isaac ROS

Η NVIDIA έχει προχωρήσει στην ενσωμάτωση του πακέτου Robot Operating System (ROS) στο Isaac, η μελλοντική εξέλιξη του Kaya θα μπορούσε να περιλαμβάνει την αντικατάσταση του Isaac SDK από το Isaac ROS την διαχείριση του αισθητήρων και κινητήρων του μέσω αυτού και την ανάπτυξη εφαρμογών σε αυτό.

Το Isaac ROS αποτελεί μία πολύ ικανή πλατφόρμα ανάπτυξης ρομποτικών εφαρμογών που δίνει την δυνατότητα στους προγραμματιστές να αναπτύξουν να δοκιμάσουν και να ενσωματώσουν σε χώρους εργασίας ρομποτικές εφαρμογές. Το Isaac ROS προήλθε από την συνεργασία της NVIDIA με την πλατφόρμα λογισμικού ανοιχτού κώδικα Robot Operating System (ROS). Έχει σχεδιαστεί με βασικό σκοπό να κάνει πιο εύκολη την

δημιουργία ρομποτικών συστημάτων με την δυνατότητα να αλληλοεπιδρούν με το περιβάλλον τους, να αναγνωρίζουν αντικείμενα, να κινούνται στον χώρο που θα τοποθετηθούν και να ολοκληρώσουν πολύπλοκες διαδικασίες. Το Isaac ROS έχει την δυνατότητα να επεξεργαστεί δεδομένα μέσω ενός ταχύτερου και αποδοτικότερου τρόπου συγκριτικά με τα παραδοσιακά ρομποτικά συστήματα.

Η δομή του Isaac ROS στηρίζεται στην χρήση κόμβων (nodes) τους ROS2 και των πακέτων του Isaac για την εκπλήρωση ειδικών λειτουργιών (GEMs).

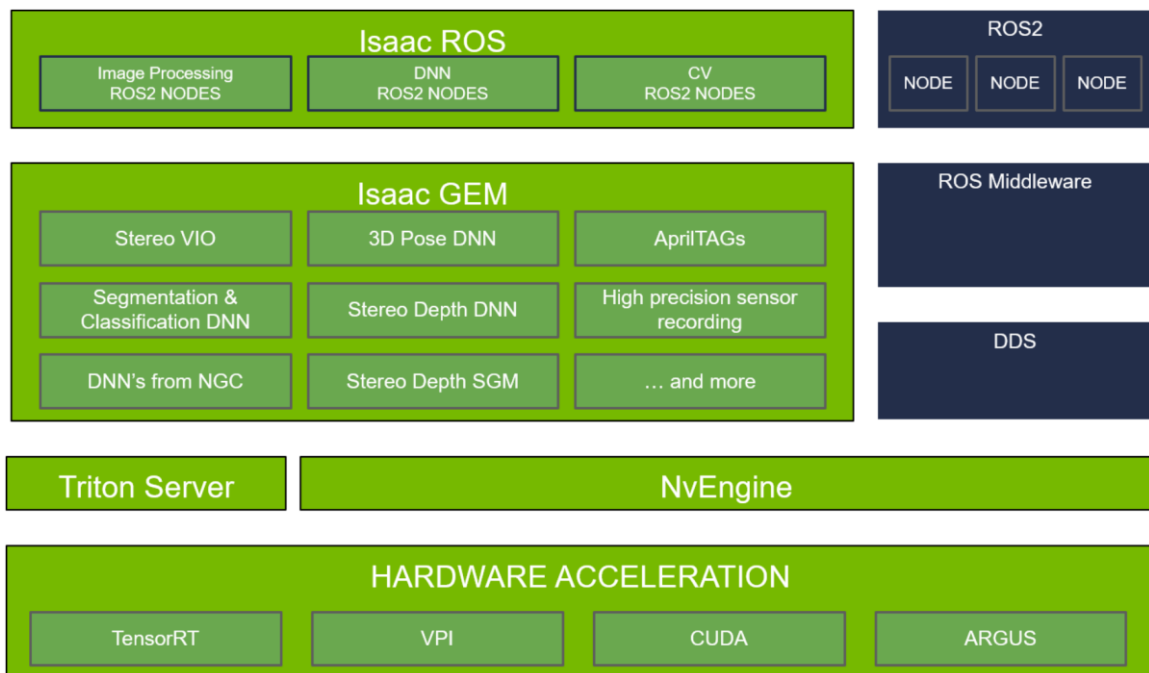


Figure 44: Δομή Isaac ROS

Βιβλιογραφία

[1] <https://github.com/nvidia-isaac/kaya-robot>

[2] https://docs.nvidia.com/isaac/archive/2020.1/doc/tutorials/kaya_software.html

[3] https://docs.nvidia.com/isaac/archive/2020.1/doc/tutorials/assemble_kaya.html

[4] <https://forums.developer.nvidia.com/>