

ΤΕΧΝΟΛΟΓΙΚΟ ΕΚΠΑΙΔΕΥΤΙΚΟ ΙΔΡΥΜΑ ΚΕΝΤΡΙΚΗΣ ΜΑΚΕΔΟΝΙΑΣ
ΣΧΟΛΗ ΤΕΧΝΟΛΟΓΙΚΩΝ ΕΦΑΡΜΟΓΩΝ
ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ

*Κινηματική ανάλυση του ρομπότ NI Starter Kit και
προγραμματισμός του ελεγκτή SbRIO 9631 για την κίνηση του
ρομπότ στο επίπεδο*

Ταρατσίδης Κωνσταντίνος (3490)
Επιβλέπων : Ιωάννης Καλόμοιρος, Αναπληρωτής Καθηγητής

ΣΕΡΡΕΣ, ΣΕΠΤΕΜΒΡΙΟΣ 2017

Περίληψη

Η παρούσα πτυχιακή εργασία πραγματεύεται την κινηματική ανάλυση του ρομπότ NI Starter Kit και τον προγραμματισμό του ελεγκτή SbRIO 9631 για την κίνηση του ρομπότ στο επίπεδο. Αρχικά γίνεται μια αναφορά στα βασικά υποσυστήματα ενός ρομποτικού οχήματος και παρουσιάζονται τα είδη των κινούμενων ρομποτικών συστημάτων με τις ιδιαιτερότητες τους. Στη συνέχεια, περιγράφεται το ρομποτικό όχημα NI Starter Kit και περιγράφονται αναλυτικά τα δομικά του στοιχεία. Επιπλέον γίνεται μια παρουσίαση της γραφικής γλώσσας προγραμματισμού LabVIEW, καθώς και των βασικών προγραμματιστικών δομών που περιλαμβάνει. Στο τρίτο κεφάλαιο γίνεται ανάλυση των κινηματικών εξισώσεων. Σχολιάζεται εκτενώς ο κώδικας που αναπτύχθηκε για την κίνηση του ρομποτικού οχήματος στο επίπεδο. Στο πέμπτο και τελευταίο κεφάλαιο περιγράφεται το πείραμα που πραγματοποιήθηκε, με την βοήθεια του οποίου καταγράφεται η κίνηση του ρομπότ στο επίπεδο, όπως προκύπτει από την οδομετρία και από ανεξάρτητες μετρήσεις.

Πίνακας περιεχομένων

Περίληψη.....	2
Κεφάλαιο 1 ^ο : Ρομποτική και Ρομποτικά Συστήματα	5
1.1 Εισαγωγή.....	5
1.1.1 Τι είναι η ρομποτική	5
1.1.2 Τι ορίζεται ως ρομπότ	5
1.1.2 Η γέννηση της λέξης ρομπότ	5
1.2 Βασικά υποσυστήματα ενός ρομπότ	5
1.3 Είδη ρομποτικών συστημάτων	6
Κεφάλαιο 2 ^ο : Πρώτη γνωριμία με το Robot NI Starter Kit.....	12
2.1 Βασικά χαρακτηριστικά του ρομποτικού οχήματος.....	12
2.1.1 Ελεγκτής	13
2.1.2 Αισθητήρες	16
2.1.3 Μοτέρ.....	19
2.2 LabVIEW	22
2.2.1 Εισαγωγή στο LabVIEW	22
2.2.2 Front Panel	22
2.2.3 Block Diagram	23
2.2.4 Παρουσίαση βασικών προγραμματιστικών δομών.....	25
Κεφάλαιο 3 ^ο : Ανάλυση κινηματικών εξισώσεων του ρομποτικού οχήματος.....	31
3.1 Εισαγωγή.....	31
3.2 Είδη βασικών τροχών	31

3.3 Θεωρία ελέγχου τροχών	33
3.4 Παρουσίαση και ανάλυση κινηματικών εξισώσεων για το ρομποτικό όχημα NI Starter Kit	37
Κεφάλαιο 4^ο : Παρουσίαση και ανάλυση του προγράμματος	41
4.1 Εισαγωγή.....	41
4.2 Παρουσίαση και ανάλυση των subVIs	45
4.2.1 Servo Rotate	45
4.2.2 Drive Motor	46
4.2.3 Odometry	49
4.3 Network Streams	50
Κεφάλαιο 5^ο : Κίνηση του ρομπότ στο επίπεδο και υπολογισμός θέσης	55
5.1 Αρχικές συνθήκες και στόχοι του πειράματος.....	55
5.2 Έλεγχος ορθότητας των μετρήσεων.....	57
5.3 Εκτέλεση του πειράματος	62
5.4 Ανάλυση και σχολιασμός αποτελεσμάτων.....	63
5.5 Συμπεράσματα Πειράματος	67
5.6 Προτεινόμενες Λύσεις.....	68
Παράρτημα 1^ο: Εγκατάσταση απαραίτητου λογισμικού	69
Παράρτημα 2^ο: Διασύνδεση του ρομποτικού οχήματος με τον υπολογιστή	76
Παράρτημα 3^ο: Τοποθετώντας ένα δρομολογητή στο ρομποτικό όχημα.....	86
Βιβλιογραφία.....	89

Κεφάλαιο 1^ο : Ρομποτική και Ρομποτικά Συστήματα

1.1 Εισαγωγή

1.1.1 Τι είναι η ρομποτική

Η Ρομποτική συνδυάζει αρχές από τις επιστήμες των μαθηματικών, της φυσικής, της μηχανολογίας, της ηλεκτρολογίας, της πληροφορικής, των τηλεπικοινωνιών, της βιολογίας. Εμπεριέχει επίσης την θεωρία συστημάτων αυτομάτου ελέγχου, την επεξεργασία ψηφιακών σημάτων, την υπολογιστική όραση και τα νευρωνικά δίκτυα. Η ρομποτική πραγματεύεται τον σχεδιασμό, την υλοποίηση, τον έλεγχο και την λειτουργία ενός ρομποτικού συστήματος.

1.1.2 Τι ορίζεται ως ρομπότ

Το ρομπότ είναι μια μηχανική συσκευή η οποία μπορεί να πραγματοποιήσει μια σειρά από περίπλοκες ενέργειες, με αυτόματο τρόπο. Ένα ρομπότ μπορεί να δράσει κάτω από τον απ' ευθείας έλεγχο ενός ανθρώπου ή αυτόνομα κάτω από τον έλεγχο μιας προγραμματισμένης κεντρικής μονάδας επεξεργασίας. Ο απώτερος σκοπός είναι η αντικατάσταση του ανθρώπου στην εκτέλεση έργου.

1.1.2 Η γέννηση της λέξης ρομπότ

Η λέξη ρομπότ προέρχεται από τη Σλαβική *robot* που σημαίνει εξαναγκασμένη εργασία. Καθιερώθηκε ως όρος με την σημερινή του έννοια το 1920 από τον Τσέχο θεατρικό συγγραφέα Karel Čapek στο έργο του "R.U.R." (Rossum's Universal Robots), όπου σατιρίζει την εξάρτηση της κοινωνίας από τους μηχανικούς εργάτες (ρομπότ), που τελικά εξοντώνουν τους δημιουργούς τους.

1.2 Βασικά υποσυστήματα ενός ρομπότ

Ένα ρομπότ αποτελείται από :

- Ένα μηχανολογικό υποσύστημα, το οποίο ενσωματώνει τη δυνατότητα του ρομπότ για εκτέλεση έργου. Το υποσύστημα αυτό αποτελείται από μηχανισμούς που επιτρέπουν στο ρομπότ να κινείται όπως αρθρώσεις, συστήματα μετάδοσης κίνησης, κινητήρες, οδηγούς κλπ.

- Ένα υποσύστημα αίσθησης, μέσω του οποίου το ρομπότ συγκεντρώνει πληροφορίες για την κατάσταση στην οποία βρίσκονται τόσο το ίδιο όσο και το περιβάλλον. Οι αισθητήρες που λαμβάνουν δεδομένα για το ρομπότ ονομάζονται εσωτερικοί ελεγκτές (proprioceptive sensors), ενώ αυτοί για το περιβάλλον εξωτερικοί (exteroceptive sensors). Το υποσύστημα είναι υπεύθυνο για την αποδοχή των εξωτερικών εντολών, την επεξεργασία τους, τη μετάφραση τους σε ηλεκτρική ισχύ που θα δοθεί στους κινητήρες του ρομπότ. Καθώς επίσης και για την παραγωγή σημάτων εξόδου που θα πληροφορούν για την κατάσταση του συστήματος. Στο υποσύστημα αίσθησης περιλαμβάνονται όργανα μετρήσεως, αισθητήρες, ηλεκτρονικά στοιχεία κλπ.
- Ένα σύστημα ελέγχου, το οποίο συνδυάζει κατάλληλα την αίσθηση με τη δράση, έτσι ώστε το ρομπότ να λειτουργεί αποτελεσματικά και ακρίβεια. Ο ελεγκτής του ρομπότ επιβλέπει το σύστημα και επεμβαίνει, με διορθωτικές ενέργειες όπου θεωρεί απαραίτητο. Για την σχεδίαση και υλοποίηση του απαιτείται ο συνδυασμός γνώσεων από πολλές γνωστικές περιοχές, όπως είναι ο αυτόματος έλεγχος, η τεχνητή νοημοσύνη, η επιστήμη των υπολογιστών, μαθηματικά, φυσική κλπ.
- Τέλος το σύστημα τροφοδοσίας το οποίο είναι απαραίτητο για την παροχή ενέργειας στο ρομπότ. Το σύστημα τροφοδοσίας είναι αρκετά υποκειμενικό για το κάθε ρομπότ και εξαρτάται κυρίως από το περιβάλλον όπου θα δρα καθώς και το μέγεθός του. Η τροφοδοσία συνήθως γίνεται με μπαταρίες διαφόρων τύπων, με καλώδια τροφοδοσίας (σε περιπτώσεις όπου το βάρος πρέπει να είναι περιορισμένο) και σπάνια με μηχανές εσωτερικής καύσης (αποφεύγονται λόγω βάρους και απαγωγή θερμότητας).

1.3 Είδη ρομποτικών συστημάτων

Με την συνεχή εξέλιξη της επιστήμης της ρομποτικής προέκυψε η ανάγκη για διαφορετικά είδη ρομποτικών μηχανισμών, οι οποίοι διαφέρουν σημαντικά στη μορφή, η οποία προκύπτει συνήθως από το περιβάλλον στο οποίο δρουν και την εργασία που εκτελούν. Η μορφή τους μπορεί να είναι διαφορετική αλλά

αποτελούνται από τα ίδια επιμέρους υποσυστήματα. Παρακάτω κατηγοριοποιούνται βάση με τον τρόπο κίνησης τους.

- **Ρομπότ σταθερής βάσης:** τα ρομπότ αυτά αποτελούνται από διαδοχικά στερεά σώματα, σύνδεσμοι(links) που συνδέονται μέσω αρθρώσεων (joints) σχηματίζοντας μία κινηματική αλυσίδα. Η αλυσίδα αυτή έχει το ένα άκρο της, την βάση (base) , σταθερά συνδεδεμένη στο περιβάλλον εργασίας ή τοποθετημένη σε μια κινητή πλατφόρμα. Στο άλλο άκρο της, το τελικό σημείο δράσης (end effector) είναι εξοπλισμένο με το κατάλληλο εργαλείο κάθε φορά. Ανάλογα με το πώς κινούνται οι σύνδεσμοι χωρίζονται και σε άλλες κατηγορίες. Τα βιομηχανικά ρομπότ είναι ρομπότ σταθερής βάσης.
- **Κινούμενα ρομπότ:** ως κινητά ρομπότ χαρακτηρίζονται όλα τα ρομπότ που έχουν τη δυνατότητα να μετακινήσουν όλα τα σημεία του μηχανισμού τους. Η δυνατότητα αυτή προσφέρεται από ειδικά συστήματα προώθησης, τα οποία μπορεί να είναι είτε απλά (όπως τροχοί) είτε πολύπλοκα (μηχανικά πόδια, προπέλες, έλικες).

a) **Αυτόματα καθοδηγούμενα οχήματα (Automatic Guided Vehicles):** τα AGVs είναι οχήματα με περιορισμένη ελευθερία. Αυτό συμβαίνει καθώς όπως υποδηλώνει και το όνομα είναι καθοδηγούμενα και όχι πλήρως αυτόνομα. Τα AGVs χωρίζονται σε δύο κατηγορίες, σε αυτά που κινούνται σε προκαθορισμένες διαδρομές στο περιβάλλον τους και σε αυτά που χρησιμοποιούν κάποιο είδος εικόνας ώστε να αναγνωρίσουν το περιβάλλον τους και να κινηθούν σε αυτό.

1. Στο χώρο εργασίας της πρώτης κατηγορίας AGV δημιουργούνται λωρίδες κυκλοφορίας, είτε με καλώδια ενσωματωμένα στο έδαφος είτε με κάποιου είδους ταινία μαγνητική ή μη, ώστε το AGV να γνωρίζει την διαδρομή που πρέπει να ακολουθήσει. Αν το μέσο δημιουργίας της λωρίδας κυκλοφορίας είναι καλώδιο τότε το χρησιμοποιείται ένας αισθητήρας ανίχνευσης ραδιοσήματος ενώ αν είναι κάποιου είδους ταινία χρησιμοποιείται αισθητήρας υπέρυθρων ή ακόμα και αισθητήρας laser ανάλογα με την θέση της ταινίας στο επίπεδο. Το μειονέκτημα με αυτή την προσέγγιση είναι ότι αν χρειαστεί να αλλάξει η διαδρομή του AGV θα πρέπει να

αλλάξει και ο χώρος εργασίας με αποτέλεσμα τον προσωρινό τερματισμό της εργασίας του.

2. Στην δεύτερη κατηγορία, τα AGV έχουν μεγαλύτερο βαθμό αυτονομίας από την άποψη ότι ο χώρος εργασίας δεν υπαγορεύει την διαδρομή τους. Τα AGV έχουν την δυνατότητα να επιλέγουν την μικρότερη διαδρομή προς τον προορισμό τους καθώς ενσωματώνουν στερεοσκοπικούς χάρτες με το περιβάλλον εργασίας και με την βοήθεια αισθητήρων όπως γυροσκόπια, laserμετρητές απόστασης, ειδικές κάμερες κ.α.

Τα AGV χρησιμοποιούνται κυρίως σε μεταφορές αντικειμένων ποικίλων διαστάσεων. Διακρίνονται σε εφαρμογές με χαρακτηριστικά όπως απαιτητικά ωράρια παράδοσης, επαναλαμβανόμενες μεταφορές αντικειμένων σε μακρινές αποστάσεις, συνεχή μεταφορά αντικειμένων με σταθερό βάρος και σε διαδικασίες που απαιτείται η τακτική παρακολούθηση των μεταφορών.

b) **Αυτόνομα ρομπότ με ρόδες:** τα ρομπότ αυτά λειτουργούν με υψηλό βαθμό αυτονομίας. Μπορούν και λειτουργούν χωρίς συνεχή εξωτερική επίβλεψη και είναι ικανά να εκτελούν εργασίες αυτόνομα, δεχόμενα μόνο ορισμένες υψηλού επιπέδου εντολές. Χρησιμοποιούν ένα μεγάλο αριθμό αισθητήρων όπως θερμικούς, οπτικούς, ηχητικούς, εγγύτητας, θερμοκρασίας, χημικούς (όπως οσμής) κ.α. Ο αριθμός και ο τύπος των αισθητήρων είναι άρρηκτα συνδεδεμένος με την εργασία την οποία θα εκτελούν. Τα ρομπότ αυτά έχουν τεράστιο φάσμα εφαρμογών από στρατιωτικούς μέχρι ψυχαγωγίας. Το ρομποτικό σύστημα NI Starter Kit που πραγματεύεται η διπλωματική εργασία ανήκει σε αυτή την κατηγορία.

c) **Βαδίζοντα ρομπότ (Legged Robots):** τα ρομπότ αυτά χρησιμοποιούν μηχανικά πόδια για την κίνηση τους στο επίπεδο. Τα κυριότερα πλεονεκτήματα της συγκεκριμένης υλοποίησης είναι η μεγάλη δυνατότητα αποφυγής εμποδίων και η ικανότητα αναρρίχησης σε ανώμαλα εδάφη και μη επίπεδες επιφάνειες. Συνήθως τέτοιου είδους ρομπότ μιμούνται την κίνηση ανθρώπων ή ζώων, όπως θηλαστικών, εντόμων, ερπετών ή και ένα συνδυασμό άκρων με ρόδες. Δύο από τα κυριότερα μειονεκτήματα τους είναι η περίπλοκη κατασκευή καθώς και οι μεγάλες απαιτήσεις τους σε ενέργεια.

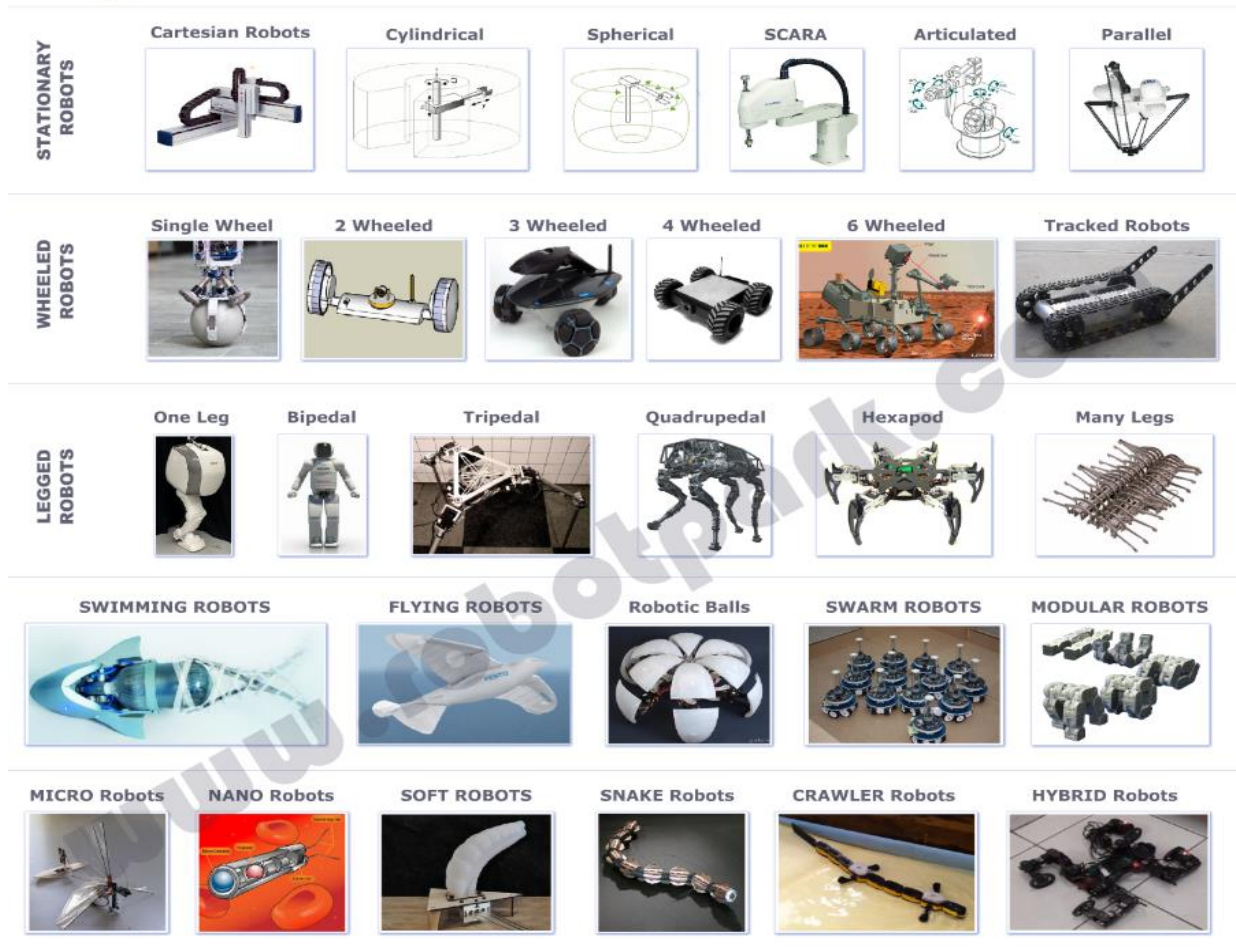
- 1. Δίποδα ρομπότ (Two legged/bipedal robots):** Η κατασκευή τους είναι από τις πιο απαιτητικές καθώς καλούνται να αντιμετωπίσουν δύο σημαντικά προβλήματα. Το πρώτο είναι ο έλεγχος της σταθερότητας, το οποίο αναφέρεται στην ισορροπία του ρομπότ. Η επίτευξη ισορροπίας είναι ένα εξαιρετικά δύσκολο πρόβλημα ,ιδιαίτερα σε δίποδα συστήματα καθώς το ρομπότ δεν μπορεί να διατηρεί την ισορροπία του στατικά ακόμα και όταν δεν κινείται. Απαιτούνται συνεχής υπολογισμοί και μικρές κινήσεις ώστε το ρομπότ ισορροπήσει, δυναμικά, σε ένα σημείο. Για Το δεύτερο πρόβλημα είναι ο έλεγχος της κίνησης όπου περιγράφει την ικανότητα του ρομπότ να κινείται. Για την επίλυση των προβλημάτων οι μηχανικοί προσπαθούν να προσομοιάσουν τον τρόπο με τον οποίο ο άνθρωπος επιτυγχάνει την ισορροπία και την κίνηση. Το ρομπότ χρησιμοποιεί γυροσκόπια, επιταχυνσιόμετρα, αισθητήρες κλίσης και ασκούμενης δύναμης ώστε να υπολογίζεται σε πραγματικό χρόνο η επόμενη κίνηση του. Υπάρχουν αρκετά δίποδα ρομπότ που έχουν καταφέρει να περπατήσουν με αρκετή αξιοπιστία αλλά κανένα από αυτά δεν έχει φτάσει στο σημείο να περπατάει με την ευκολία ενός ανθρώπου. Τα ρομπότ αυτά συνήθως προορίζονται για εφαρμογές όπου θα βοηθούν τους ανθρώπους σε διάφορες καθημερινές εργασίες ή θα αναλαμβάνουν να διεκπεραιώσουν εργασίες κουραστικές και ανιαρές. Στην Ιαπωνία, λόγω χάρη, πολλοί υπάλληλοι υποδοχής σε πανεπιστήμια αντικαθίστανται από τέτοια ρομπότ.
- 2. Τετράποδα ρομπότ (Four legged/quadrupedal robots):** Έχουν το πλεονέκτημα όταν είναι ακίνητα να ισορροπούν στατικά λόγω της κατασκευής τους. Υπάρχουν αρκετοί τρόποι για να κινήσουν τα άκρα τους. Ένας απλός τρόπος είναι κάθε φορά να κινείται ένα από τα τέσσερα άκρα. Έτσι εξασφαλίζεται ότι το κέντρο βάρους του ρομπότ θα είναι πάντα σε ένα νοητό τρίγωνο που σχηματίζεται από τα άλλα τρία σταθερά πόδια. Με αυτή την τεχνική το ρομπότ ισορροπεί ακόμα και όταν κινείται με αποτέλεσμα να μην απαιτεί δυναμικός έλεγχος της κίνησης. Ένας πιο σύνθετος τρόπος είναι να κινούνται τα άκρα ανά ομάδες των δύο. Το αποτέλεσμα είναι ότι κατά την κίνηση του δεν μπορεί να ισορροπεί στατικά αλλά κινείται γρηγορότερα. Το βάδισμα (gait) του ρομπότ καθορίζει αν το

ρομπότ θα έχει δυναμική ή στατική ισορροπία. Μπορεί κανείς να συναντήσει τετράποδα ρομπότ σε στρατιωτικές εφαρμογές ως μεταφορείς αντικειμένων στο πεδίο της μάχης ή ακόμη και σαν ένα είδους ζωάκι που κρατάει συντροφιά σε ηλικιωμένους δίχως να απαιτεί φροντίδα.

d) **Υποβρύχια οχήματα απομακρυσμένου ελέγχου (Remotely Operated underwater Vehicle):** τα ROVs ανήκουν στην κατηγορία των μη επανδρωμένων ρομπότ. Δεν έχουν μεγάλο βαθμό αυτονομίας, μιας και είναι συνδεδεμένα με το μητρικό σκάφος μέσω καλωδίου, το οποίο καλύπτει τις ανάγκες του ρομπότ σε ενέργεια και επικοινωνίες. Είναι ευκολότερα στην κατασκευή και χρησιμοποιούν λιγότερους σε αριθμό αισθητήρες για την κίνηση τους λόγω της απλότητας του περιβάλλοντος. Περιέχουν αισθητήρες για χημική ανάλυση υδάτων, θερμοκρασίας, βάθους κ.α. Χρησιμοποιούνται κυρίως για επιστημονικούς και στρατιωτικούς σκοπούς.

e) **Μη επανδρωμένα ιπτάμενα οχήματα (Unmanned Aerial Vehicle):** τα UAVs έχουν την μορφή αεροπλάνων ή ελικοφόρων οχημάτων. Είναι πλήρως αυτόνομα στον αέρα αλλά συνήθως η απογείωση και προσγείωση γίνεται από άνθρωπο. Τα οχήματα αυτά κατηγοριοποιούνται ανάλογα με το μέγεθος και τις πτητικές τους ιδιότητες. Οι αισθητήρες που περιλαμβάνουν είναι βαρόμετρα, υψόμετρα, γυροσκόπια, επιταχυνσιόμετρα, πυξίδες, δέκτες GPS κ.α. Τα UAVs είναι συστήματα πραγματικού χρόνου για το λόγο αυτό χρησιμοποιούν single-board computers (SBC) ώστε να αντιδρούν άμεσα στα δεδομένα των αισθητήρων. Έχουν διαρκώς αυξανόμενες εφαρμογές, και χρησιμοποιούνται για στρατιωτικούς κυρίως σκοπούς. Τα τελευταία χρόνια τα τετρακόπτερα (quadcopters) έχουν γίνει αρκετά δημοφιλή καθώς αποτελούν σχετικά φθηνές λύσεις για εναέρια φωτογράφιση, μεταφορά προϊόντων κ.α.

All Types of Robots by Locomotion

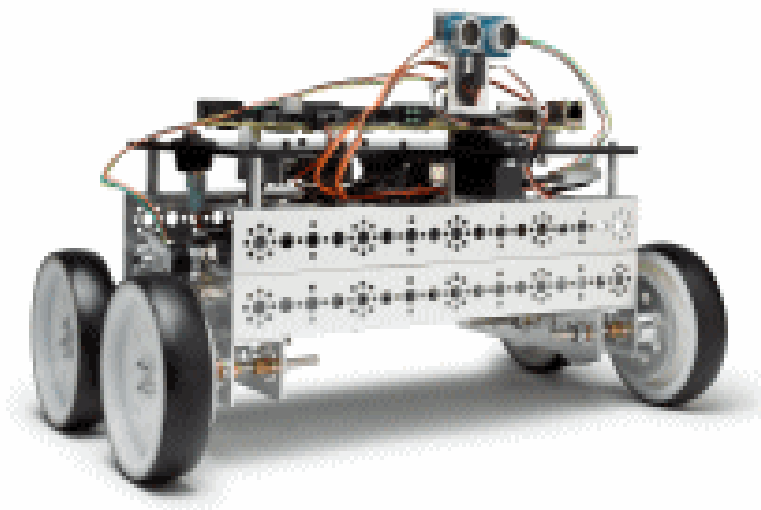


Εικόνα 1.1: Είδη ρομποτικών συστημάτων βάσει του τρόπου που κινούνται

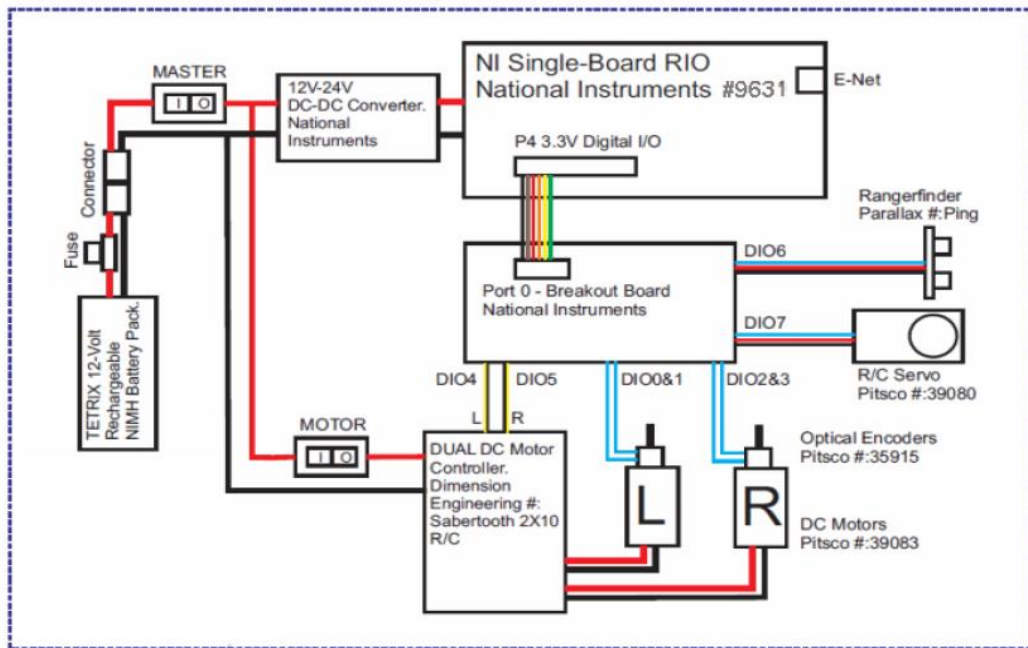
Κεφάλαιο 2^ο : Πρώτη γνωριμία με το Robot NI Starter Kit

2.1 Βασικά χαρακτηριστικά του ρομποτικού οχήματος

Το κιτ της National Instruments (NI) Robotics Starter Kit 1.0 είναι ένα κινητό ρομπότ το οποίο έρχεται συναρμολογημένο και περιλαμβάνει έναν ελεγκτή SbRIO 9631 για τον έλεγχο του, αισθητήρες, δύο μοτέρ και ένα πλήρως αρθρωτό σασί για την προσθήκη επιπλέον αισθητήρων. Η κίνηση που παράγουν τα μοτέρ μεταφέρεται στις ρόδες με ένα σύστημα γραναζιών. Προγραμματίζεται με το λογισμικό NI LabVIEW 2011, ή και μεταγενέστερης έκδοσης, στο οποίο έχουν ενσωματωθεί τα παρακάτω τμήματα λογισμικού : Robotics Module, Real-Time Module, FPGA Module και InternetToolkit. Το LabVIEW είναι μια οπτική γλώσσα προγραμματισμού όπου δίνει την δυνατότητα ακόμα και σε κάποιον που δεν έχει μεγάλη εμπειρία στον παραδοσιακό προγραμματισμό να καταφέρει να το προγραμματίσει. Αυτό δεν σημαίνει βέβαια ότι δεν είναι κατάλληλο και για πιο απαιτητικές εφαρμογές. Παρακάτω θα αναφερθεί αναλυτικότερα η δομή του ρομπότ καθώς και τα επιμέρους εξαρτήματα που το απαρτίζουν.



Εικόνα 2.1: Αναπτυξιακό σύστημα ρομπότ NI Starter Kit 1.0



Εικόνα 2.2: Μπλοκ διάγραμμα υλικού του ρομποτικού οχήματος NI Starter Kit

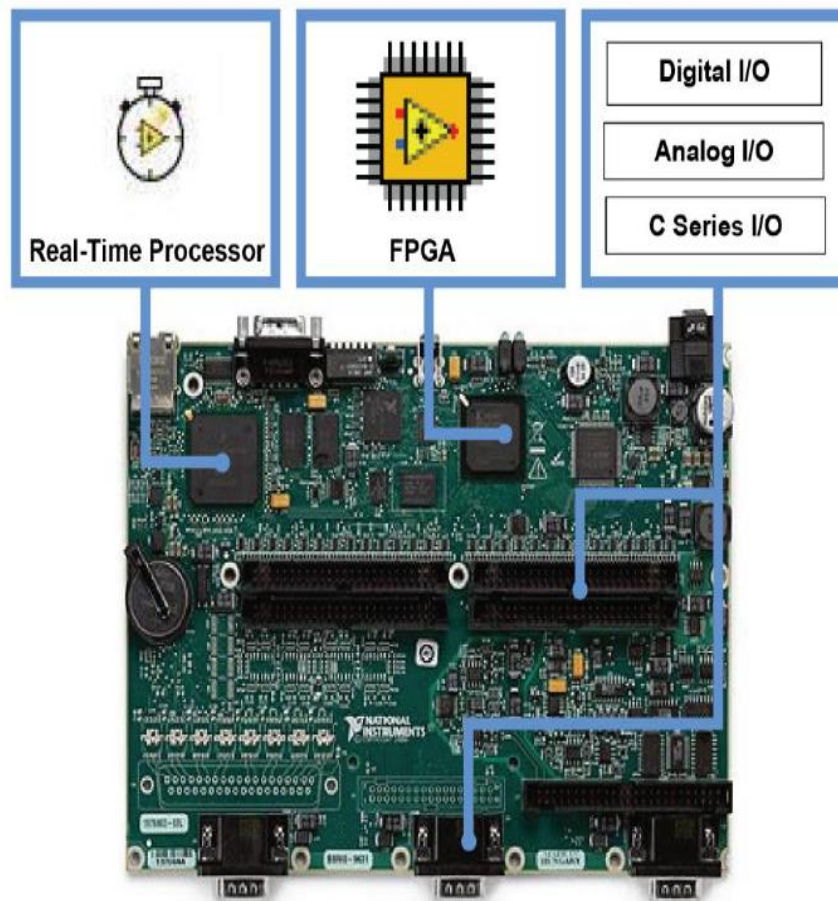
2.1.1 Ελεγκτής

Το ρομποτικό όχημα χρησιμοποιεί τον ελεγκτή NI Single-Board RIO 9631 (SbRIO9631). Ο ελεγκτής αυτός ενσωματώνει έναν επεξεργαστή σημάτων πραγματικού χρόνου (DSP) με συχνότητα λειτουργίας στα 266MHz, μια συστοιχία επιτόπια προγραμματιζόμενων πυλών (FPGA) Xilinx Spartan του ενός εκατομμυρίων πυλών και μια σειρά από αναλογικές-ψηφιακές εισόδους/εξόδους (Analog-Digital I/O) των 3.3V, ανεκτές στα 5V και συμβατές με την οικογένεια ολοκληρωμένων κυκλωμάτων TTL. Οι ενσωματωμένες εισοδοί και έξοδοί μπορούν επεκταθούν με την σειρά C-Series της NI. Το SbRIO 9631 επιπλέον περιλαμβάνει, 64MB μνήμης RAM και 128MB ROM για την αποθήκευση των προγραμμάτων και συλλογή δεδομένων. Διαθέτει θύρα 10/100 Mbits/s Ethernet η οποία μπορεί να χρησιμοποιηθεί για επικοινωνία μέσω του δικτύου και σειριακή θύρα RS232 για έλεγχο περιφερειακών συσκευών. Τέλος ο αναπτυξιακός ελεγκτής μπορεί να τροφοδοτηθεί από ένα εύρος τάσης από 19 ως 30 VDC.

Ένας επεξεργαστής ψηφιακού σήματος ή DSP (Digital Signal Processor), είναι ένας εξειδικευμένος μικροεπεξεργαστής που έχει μια αρχιτεκτονική βελτιστοποιημένη για τις λειτουργικές ανάγκες της γρήγορης επεξεργασίας ψηφιακού σήματος. Ένας επεξεργαστής ψηφιακού σήματος είναι ικανός να επεξεργάζεται δεδομένα σε πραγματικό χρόνο, και είναι ιδανικός για εφαρμογές που δεν μπορούν να ανεχθούν καθυστερήσεις. Οι ψηφιακοί επεξεργαστές σήματος λαμβάνουν ένα ψηφιακό σήμα και το επεξεργάζονται για να βελτιώσουν ένα ηχητικό σήμα σε καθαρότερο ήχο, να μεταδώσουν ταχύτερα δεδομένα ή να αποδώσουν ευκρινέστερες εικόνες. Οι επεξεργαστές ψηφιακού σήματος χρησιμοποιούν σήματα βίντεο, φωνής, ήχου, θερμοκρασίας ή θέσης τα οποία έχουν ψηφιοποιηθεί και τα επεξεργάζονται με μαθηματικό τρόπο. Ένας επεξεργαστής ψηφιακών σημάτων έχει σχεδιαστεί για να εκτελεί αυτές τις μαθηματικές λειτουργίες σε ελάχιστο χρόνο. Τα σήματα επεξεργάζονται έτσι ώστε οι πληροφορίες που περιέχονται σε αυτά να μπορούν να εμφανιστούν ή να μετατραπούν σε άλλο τύπο σήματος. Βρίσκουν εφαρμογές στην συμπίεση και επεξεργασία ήχου και βίντεο, την ψηφιακή επεξεργασία εικόνας, τις ψηφιακές επικοινωνίες, τις εφαρμογές ραντάρ και γενικότερα όπου χρειάζεται επεξεργασία ψηφιακών δεδομένων δίχως χρονικές απώλειες.

Η συστοιχία των επιτόπια προγραμματιζόμενων πυλών (FPGA) είναι μια συσκευή ημιαγωγών η οποία μετά από κατάλληλο προγραμματισμό μπορεί να υλοποιήσει σχεδόν οποιοδήποτε ψηφιακό κύκλωμα. Τα FPGA, για παραγωγές χαμηλού ως μεσαίου όγκου, παρέχουν μια φθηνότερη και πιο άμεση υλοποίηση του προϊόντος σε σχέση με τα κυκλώματα ειδικού σκοπού (ASIC) που συνήθως απαιτούν πολλούς πόρους από άποψη χρόνου και χρημάτων για να κατασκευαστούν. Βασικό χαρακτηριστικό των FPGA είναι ότι μπορούν εύκολα να επαναπρογραμματιστούν με αποτέλεσμα η αναβάθμιση πρωτότυπων κυκλωμάτων να γίνεται εύκολα και με μηδενικό κόστος. Λόγω των προγραμματιζόμενων διασυνδέσεων που περιέχονται στο FPGA όπου δεσμεύουν γύρω στο 90% της συνολικής επιφάνειας του, προκύπτει ότι οι υπόλοιποι πόροι υλικού είναι μειωμένοι. Από την άλλη μεριά, τα διάφορα κυκλώματα ειδικού σκοπού καταναλώνουν λίγη ενέργεια, είναι μικρά σε μέγεθος καθώς και πιο γρήγορα. Τα FPGA παρουσιάζουν μια εναλλακτική λύση για την

υλοποίηση ενός ψηφιακού συστήματος, λόγω του μικρότερου χρόνου που απαιτείται για την διάθεση του στην αγορά καθώς και του ιδιαίτερα μικρού κόστους ανάπτυξης. Ο προγραμματισμός των FPGA γίνεται με γλώσσες περιγραφής υλικού όπως τις VHDL και Verilog, όπου ο προγραμματισμός γίνεται βάσει κειμένου, και το LabVIEW όπου η ανάπτυξη του κώδικα γίνεται μέσω γραφικής απεικόνισης.



Εικόνα 2.3: *Single-Board RIO 9631*

2.1.2 Αισθητήρες

Το ρομπότ είναι αρχικά εξοπλισμένο με δύο βασικούς αισθητήρων, ένα αισθητήρα υπερήχων και ένα σετ από οπτικούς κωδικοποιητές που βρίσκονται στα μοτέρ το ρομπότ. Βέβαια υπάρχει μπορούν να προστεθούν και άλλου είδους αισθητήρες όπως GPS, γυροσκόπια, κάμερες κ.α.

1. Ο αισθητήρας υπερήχων της Parallax όπου εντοπίζει αντικείμενα, εκπέμποντας μικρής διάρκειας παλμούς τους οποίους περιμένει να ακούσει αν ανακλαστούν (ηχώ). Με την βοήθεια ενός μικροελεγκτή, ο αισθητήρας εκπέμπει ένα παλμό συχνότητας 40kHz διάρκειας 200μs. Ο παλμός ταξιδεύει στον αέρα με ταχύτητα 343m/s, χτυπάει πάνω σε ένα αντικείμενο και ανακλάται πίσω στον αισθητήρα. Ο αισθητήρας στέλνει ένα σήμα εξόδου στον ελεγκτή που αντιστοιχεί στο χρόνο που έκανε ο παλμός να επιστρέψει στον αισθητήρα. Έτσι υπολογίζεται εύκολα η απόσταση που απέχει το αντικείμενο. Ο αισθητήρας μπορεί να εντοπίσει εμπόδια που απέχουν από αυτόν 2 εκατοστά μέχρι και 3 μέτρα. Επιπλέον ο αισθητήρας υπερήχων είναι τοποθετημένος σε ένα servo μοτέρ με δυνατότητα περιστροφής σε ένα εύρος -90 με 90 μοιρών εντοπίζοντας για εμπόδια γύρω από

Προδιαγραφές αισθητήρα υπερήχων Parallax (PING)))	
Συχνότητα Παλμού	40 kHz για 200μs
Ελάχιστη ηχώ παλμού	115 μs
Μέγιστη ηχώ παλμού	18.5 ms
Καθυστέρηση μέχρι την επόμενη μέτρηση	200 μs

το ρομπότ.



Εικόνα 2.4: Αισθητήρας Υπερήχων τοποθετημένος σε *servomotor*

Ο χρόνος από τη στιγμή που εκπέμπεται το κύμα ως την στιγμή που επιστρέφει το ανακλώμενο σήμα στον αισθητήρα ονομάζεται time-of-flight. Ο χρόνος στο FPGA μετριέται σε ticks. Το tick υπολογίζεται σε συνάρτηση με την συχνότητα του FPGA, στην συγκεκριμένη περίπτωση 40MHz. Έτσι 1 tick ισούται με 25ns. Το FPGA μετατρέπει τα ticks σε time-of-flight και έπειτα σε απόσταση. Ο τύπος που χρησιμοποιείται είναι

$$d = c * t/2$$

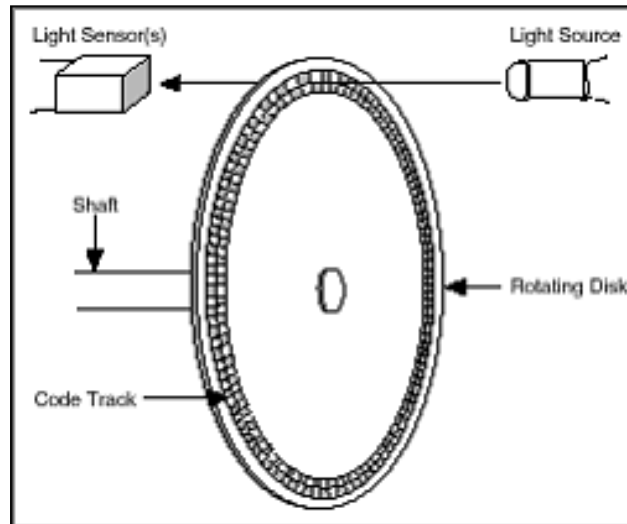
Όπου

d = απόσταση του αντικειμένου που προκαλεί την ανάκλαση

c = ταχύτητα του ήχου στο αέρα με συνηθισμένη πίεση στους 20°C

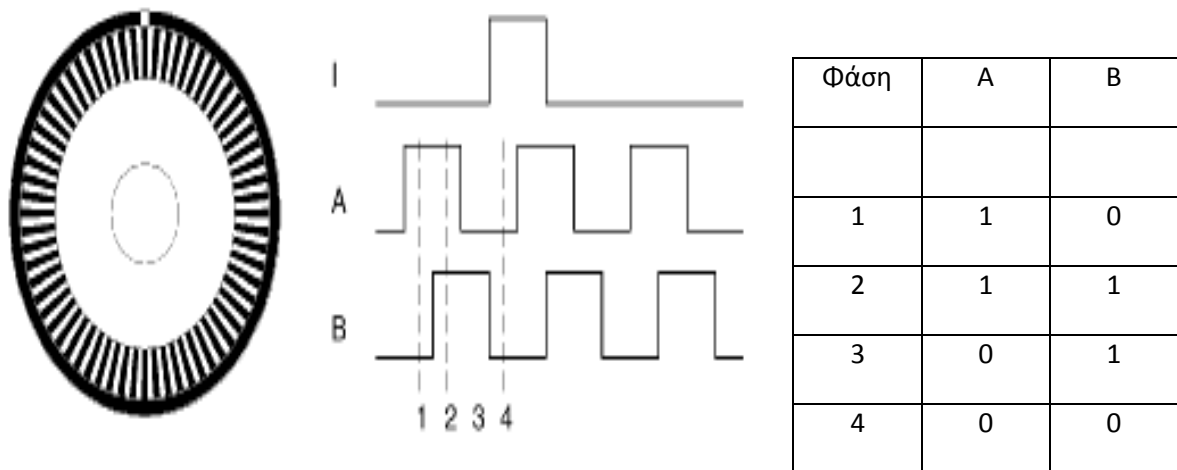
t = time-of-flight

2. Οι δύο οπτικοί κωδικοποιητές της Tetrix που βρίσκονται στο ρομπότ επιστρέφουν κάθε στιγμή τις γωνιακές ταχύτητες των δύο μοτέρ. Γνωρίζοντας τις γωνιακές ταχύτητες κάθε στιγμή μπορούμε να υπολογίσουμε την θέση, την κατεύθυνση και την επιτάχυνση του ρομπότ κάθε στιγμή. Ένας κωδικοποιητής είναι μια ηλεκτρομηχανική συσκευή η οποία μετατρέπει την γωνιακή θέση ή κίνηση ενός άξονα σε ψηφιακό ή παλμικό σήμα. Ένας οπτικός κωδικοποιητής χρησιμοποιεί ένα περιστρεφόμενο δίσκο, μια πηγή φωτός και ένα αισθητήρα φωτός. Ο δίσκος χωρίζεται σε διαφανή και αδιαφανή τμήματα και στερεώνεται στο άξονα κίνησης του μοτέρ. Ο αισθητήρας φωτός τοποθετείται πίσω από δίσκο ενώ από μπροστά βρίσκεται η πηγή φωτός. Καθώς ο δίσκος περιστρέφεται, οι αδιάφανοι τομείς διακόπτουν το φως που εκπέμπεται παράγοντας μια έξοδο ψηφιακού ή παλμικού σήματος.



Εικόνα 2.5: Οπτικός κωδικοποιητής

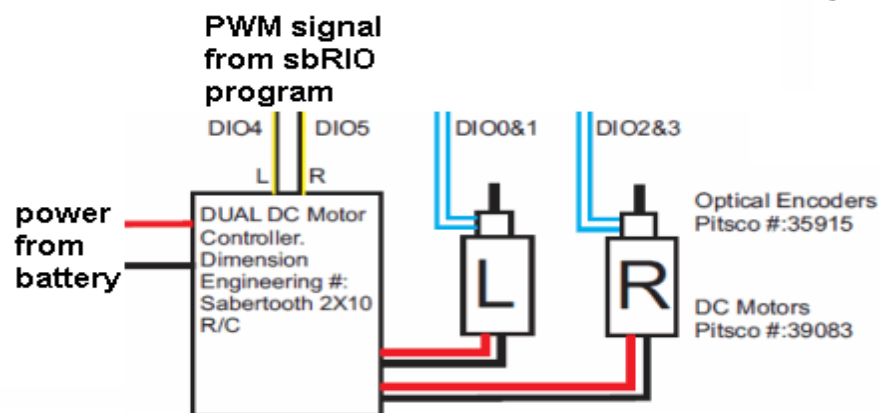
Οι οπτικοί κωδικοποιητές χωρίζονται σε απόλυτους και σε σταδιακούς. Τα κανάλια (channels) του απόλυτου κωδικοποιητή, συνήθως τέσσερα με έξι, χρησιμοποιούνται για την παραγωγή δυαδικών εξόδων όπως BCD και κώδικα Gray. Ενώ ο σταδιακός κωδικοποιητής δημιουργεί ένα ημιτονοειδές κύμα για κάθε περιστροφή του άξονα το οποίο στην συνέχεια μετατρέπεται σε διακριτό τετραγωνικό παλμό. Για την μετατροπή του χρησιμοποιείται ένα κατώφλι (threshold) για να επιλεγθούν ποιες καταστάσεις θεωρούνται σκοτεινές ή φωτεινές. Η ακρίβεια του σταδιακού οπτικού κωδικοποιητή μετριέται σε κύκλους ανά περιστροφή (Cycles Per Revolution). Αυτού του είδους κωδικοποιητή χρησιμοποιούνται κατά κόρον λόγω του φθηνού κόστους και της ακρίβειας των μετρήσεων τους. Σε ρομποτικές εφαρμογές χρησιμοποιείται ο σταδιακός οπτικός κωδικοποιητής τετραγωνισμού (quadrature incremental optical encoder), ο οποίος χρησιμοποιεί ένα δεύτερο κανάλι και αισθητήρα φωτός με διαφορά 90 μοιρών από το πρώτο σετ. Οι δύο τετραγωνικοί παλμοί, ονομάζονται A και B, που προκύπτουν δίνουν μεγαλύτερο όγκο πληροφορίας καθώς για παράδειγμα αν ο A προηγείται του B τότε άξονας κινήθηκε με τη φορά του ρολογιού. Έτσι μελετώντας τον αριθμό των περιστροφών αλλά και την σχετική φάση των δύο παλμών μπορεί κανείς να γνωρίζει και την θέση αλλά και την διεύθυνση τη περιστροφής. Πιο σύνθετοι τετραγωνικοί κωδικοποιητές περιέχουν ένα τρίτο κανάλι εξόδου Z, το οποίο παρέχει ένα παλμό ανά περιστροφή και χρησιμοποιείται σαν σημείο αναφοράς.



Εικόνα 2.6: Σταδιακός κωδικοποιητής τετραγωνισμού με κανάλια A, B και Z. Το εξωτερικό κομμάτι δημιουργεί ένα δείκτη (αναφορά) ανά περιστροφή.

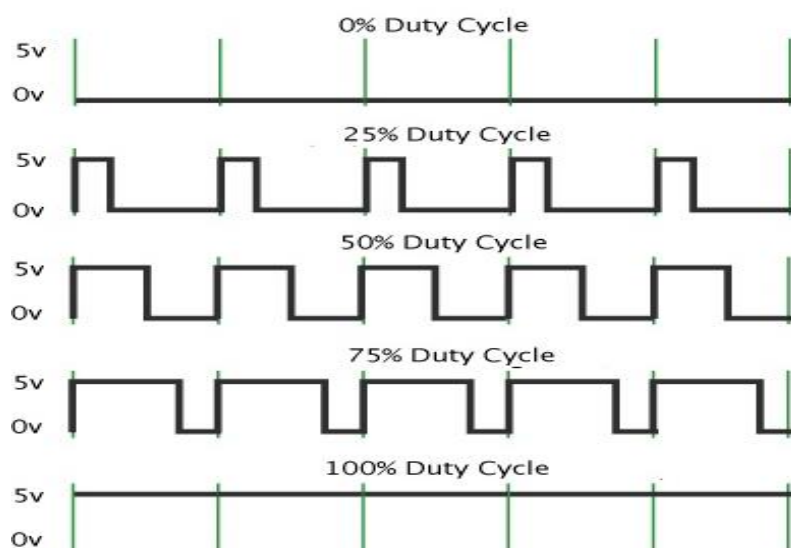
2.1.3 Μοτέρ

Το ρομποτικό όχημα NI Starter Kit είναι εξοπλισμένο με δύο 12VDC μοτέρ της Tetrrix τα οποία συνδέονται με το ελεγκτή SbRIO 9631 μέσω του οδηγού των μοτέρ SaberTooth 2x10 R/C. Ο οδηγός των μοτέρ συνδέεται με τις δύο μπαταρίες χωρητικότητας 3000mAh και μπορεί να παρέχει μέχρι 10A σε κάθε ένα μοτέρ ξεχωριστά. Ο ελεγκτής στέλνει PWM σήματα ελέγχου στα μοτέρ μέσω των ψηφιακών εισόδων εξόδων (Digital Input/Output). Το σήμα από την DIO4 ελέγχει το αριστερό μοτέρ ενώ το σήμα DIO5 το δεξί μοτέρ.



Εικόνα 2.7: Block διάγραμμα του κυκλώματος ελέγχου των μοτέρ

Η διαμόρφωση εύρους παλμού (PWM) είναι μια τεχνική στην οποία μια σειρά ψηφιακών παλμών χρησιμοποιείται για τον έλεγχο ενός αναλογικού κυκλώματος. Το μήκος και η συχνότητα αυτών των παλμών καθορίζει τη συνολική ισχύ που παραδίδεται στο κύκλωμα. Τα σήματα PWM είναι ιδανικά για τον έλεγχο DC μοτέρ. Ο ελεγκτής των μοτέρ παράγει ένα σήμα PWM με σταθερή συχνότητα και μεταβάλλει το πλάτος παλμού για να αλλάξει τη μέση ισχύ του σήματος. Η αναλογία του πλάτους παλμού προς την περίοδο αναφέρεται ως ο κύκλος λειτουργίας (duty cycle) του σήματος. Για παράδειγμα, εάν ένα σήμα PWM έχει μια περίοδο 10 ms και οι παλμοί του έχουν μήκος 2,5 ms, το σήμα αυτό λέγεται ότι έχει ένα 25% κύκλο λειτουργίας όπως φαίνεται στην εικόνα 2.8, η οποία δείχνει πέντε σήματα PWM με διαφορετικούς κύκλους λειτουργίας. Τέλος πρέπει να αναφερθεί πως ο κώδικας ελέγχου των μοτέρ έχει ήδη υλοποιηθεί για τον χρήστη ως συνάρτηση στο LabVIEW και φορτώνεται με την δημιουργία ενός νέου project.

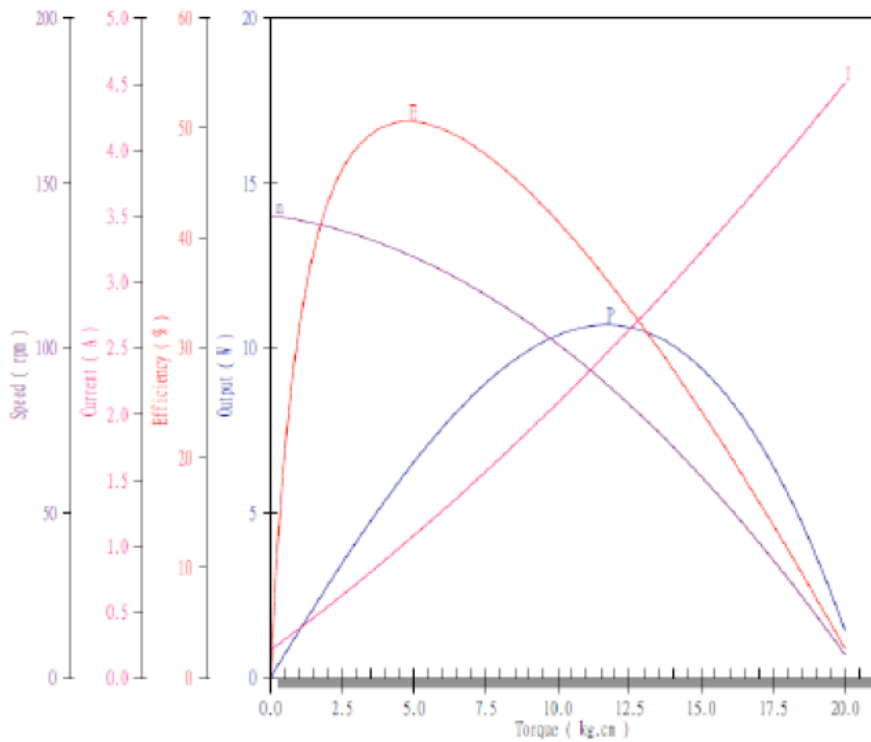


Εικόνα 2.8: PWM σήματα με διαφορετικά duty cycles

Μελετώντας το διάγραμμα της εικόνας 2.9, παρατηρείται πως το μοτέρ μπορεί να αποδώσει με δύο διαφορετικούς τρόπους ανάλογα με τις ανάγκες του εδάφους και του χρήστη. Αν ο χρήστης ενδιαφέρεται για την μέγιστη δυνατή ταχύτητα τότε το μοτέρ μπορεί να αποδώσει 128 στροφές το λεπτό, ροπή στρέψης 4.85kg-cm ενώ η ισχύς που απαιτείται είναι 6,38 Watts. Τέλος η αποδοτικότητα του μοτέρ φτάνει το 50%. Αν όμως το έδαφος είναι ανηφορικό ή το ρομπότ μεταφέρει επιπλέον βάρος

τότε απαιτείται επιπλέον ροπή στρέψης ώστε αυτό να κινηθεί. Έτσι σε αυτό το σενάριο η ροπή στρέψης φτάνει στα 11.72 kg-cm, η ισχύς των μοτέρ στα 10.7 Watts, η ταχύτητα στις 89 περιστροφές το λεπτό ενώ το ρεύμα που απαιτούν τα μοτέρ είναι 2,46 A. Η αποδοτικότητα του μοτέρ πέφτει στο 36.22%.

(B) MOTOR PERFORMANCE CURVES AND CHARACTERISTICS:



Supply Voltage	= 12	VDC
<u>AT NO LOAD</u>		
SPEED	= 146	RPM
CURRENT	= 0.17	AMP
<u>AT STALL EXTRAPOLATION</u>		
TORQUE	= 20	kg-cm
CURRENT	= 4.55	AMP
<u>AT MAXIMUM EFFICIENCY</u>		
EFFICIENCY	= 50	%
SPEED	= 128	RPM
TORQUE	= 4.85	kg-cm
CURRENT	= 1.05	AMP
OUTPUT	= 6.38	WATTS
<u>AT MAXIMUM OUTPUT</u>		
SPEED	= 89	RPM
TORQUE	= 11.72	kg-cm
CURRENT	= 2.46	AMP
OUTPUT	= 10.7	WATTS
EFFICIENCY	= 36.22	%

Torque Conversions:
 1.0 kgf-cm = 0.098 Nm = 98 mNm
 = 13.887 oz-in = 0.867 lb-in

Εικόνα 2.9: Τεχνικά χαρακτηριστικά των μοτέρ

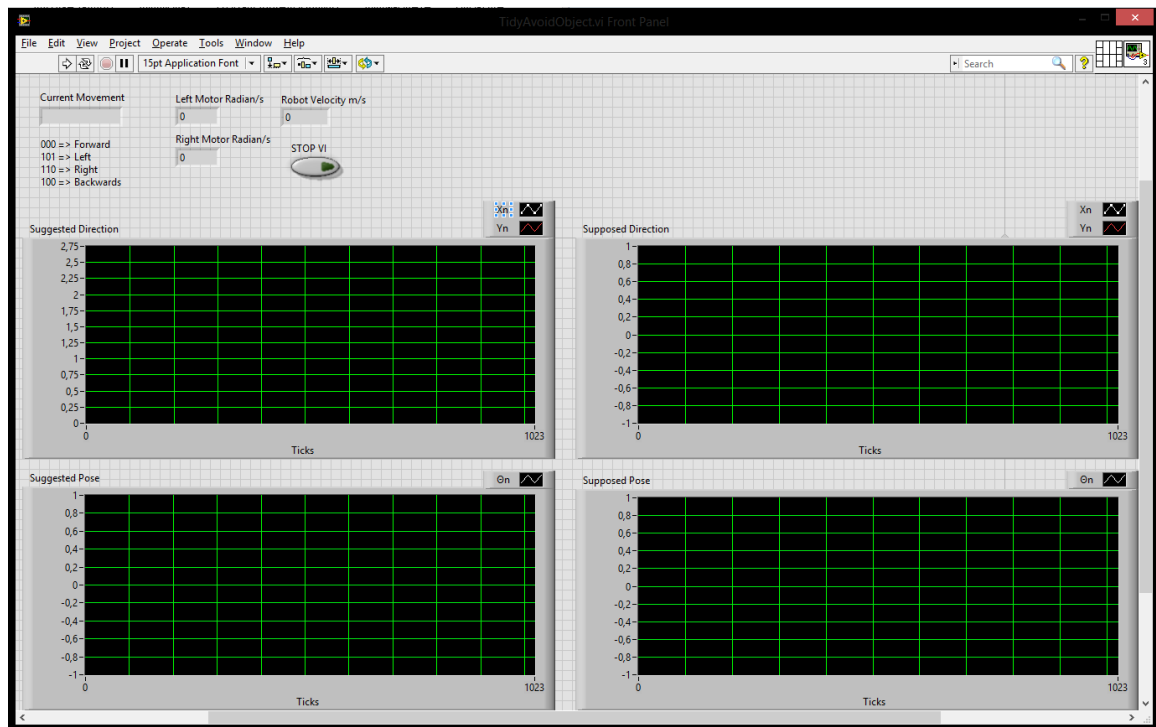
2.2 LabVIEW

2.2.1 Εισαγωγή στο LabVIEW

Το LabVIEW (Laboratory Virtual Instrument Engineering Workbench) είναι μια γραφική γλώσσα προγραμματισμού μετρήσεων, ελέγχου και ανάλυσης δεδομένων όπου χρησιμοποιούνται εικονίδια και όχι λεκτικές εντολές για την δημιουργία εφαρμογών. Η εκτέλεση του εκάστοτε προγράμματος στο LabVIEW καθορίζεται από τον προγραμματισμό ροής (data flow programming) όπου η ροή των δεδομένων καθορίζει την εκτέλεση του προγράμματος. Ο προγραμματισμός σε LabVIEW χωρίζεται σε τρία βασικά μέρη. Στην δημιουργία μιας διεπαφής με τον χρήστη που ονομάζεται εμπρόσθιο πλαίσιο (Front Panel) όπου προβάλλονται δεδομένα, γραφήματα και δίνεται η δυνατότητα αλληλεπίδρασης με την εφαρμογή. Ο χώρος όπου τοποθετείται ο κώδικας, είναι το δομικό διάγραμμα (Block Diagram), το οποίο σε πολλές περιπτώσεις μοιάζει με διάγραμμα ροής και είναι αναπόσπαστο κομμάτι με το Front Panel. Τέλος, υπάρχουν οι παλέτες εργαλείων και συναρτήσεων όπου βρίσκονται τα εικονικά όργανα για την συγγραφή της εφαρμογής. Τα εικονικά όργανα θα μπορούσαν να παρομοιαστούν με τις συναρτήσεις και τις υπορουτίνες μιας κλασσικής γλώσσας προγραμματισμού όπως η C.

2.2.2 Front Panel

Όπως αναφέρθηκε και πριν, το Front Panel είναι το μέρος όπου προβάλλονται όλοι οι έξοδοι του προγράμματος με την βοήθεια ψηφιακών οθονών, διαγραμμάτων και LEDs. Επιπλέον δίνεται η δυνατότητα ελέγχου του προγράμματος από το χρήστη με την χρήση διακοπών, πλήκτρων, ροοστατών και άλλων μέσων. Κάθε αντικείμενο που τοποθετείται στο Front Panel εμφανίζεται αυτόματα και στο Block Diagram ώστε να συνδεθεί κατάλληλα στην εφαρμογή.

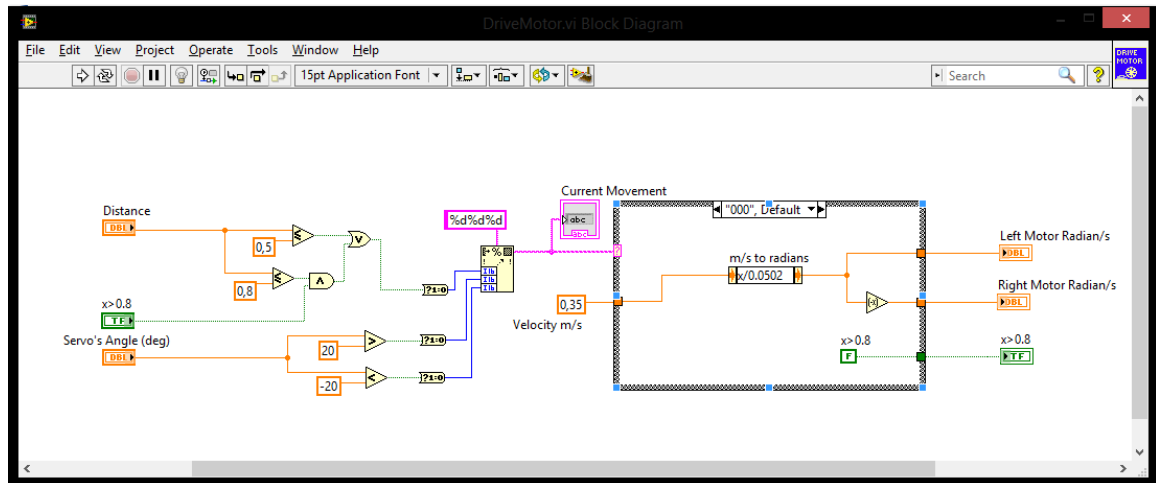


Εικόνα 2.10: Παράθυρο *Front Panel*

Η τοποθέτηση κάποιου αντικειμένου στο front panel γίνεται με την εμφάνιση του πίνακα *Controls Palette* όπου περιέχει όλα τα στοιχεία εισόδου και εξόδου. Τα στοιχεία εισόδου περιλαμβάνουν πλήκτρα, ποτενσιόμετρα και άλλα στοιχεία τα οποία δίνουν την δυνατότητα εισόδου τιμών στην εφαρμογή από τον έξω κόσμο. Ενώ τα στοιχεία εξόδου περιλαμβάνουν γραφήματα, πίνακες και άλλα στοιχεία απεικόνισης διάφορων τύπων δεδομένων.

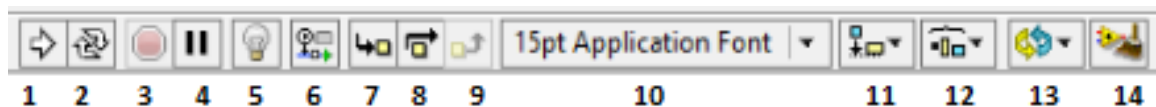
2.2.3 Block Diagram

Το βασικό δεύτερο μέρος, το *Block Diagram*, περιέχει το κώδικα προγράμματος στη γραφική γλώσσα G και θυμίζει οπτικά τη μορφή ενός ηλεκτρικού κυκλώματος. Στο *Block Diagram* βρίσκονται όλα τα στοιχεία που βρίσκονται στο *Front Panel* καθώς και αντικείμενα όπως συναρτήσεις, σταθερές, προγραμματιστικές δομές, καλώδια που συνδέουν τα αντικείμενα μεταξύ τους.



Εικόνα 2.11: Παράθυρο *Block Diagram*

Η γραμμή εργαλείων που βρίσκεται στο πάνω μέρος του *Block Diagram* είναι υπεύθυνη για την σύνταξη, ανάπτυξη (deployment) και αποσφαλμάτωση του κώδικα.



Εικόνα 2.12: Βασική γραμμή εργαλείων

Τα αριθμημένα στοιχεία παίζουν τον παρακάτω ρόλο:

1. Μεταγλώττιση και εκτέλεση του προγράμματος. Αν το πρόγραμμα έχει κάποιο συντακτικό σφάλμα τότε το βέλος θα είναι γκριζό και σπασμένο στην μέση.
2. Συνεχή εκτέλεση του προγράμματος μέχρι να διακοπεί από τον χρήστη
3. Διακόπτει άμεσα την εκτέλεση του προγράμματος. Συνιστάται η χρήση του μόνο αν το πρόγραμμα δεν αποκρίνεται. Αν εκτελείται σε κάποιο εξωτερικό υλικό (hardware) , όπως το SbRIO-9631 υπάρχει μεγάλη πιθανότητα να αφήσει τους πόρους που έχει δεσμεύσει σε άγνωστη κατάσταση. Συνιστάται κατά τον σχεδιασμό της εφαρμογής να λαμβάνεται υπόψη ο τρόπος που θα διακόπτεται αυτή από τον χρήστη ώστε να αποφεύγεται το παραπάνω πρόβλημα.
4. Σταματάει προσωρινά ή συνεχίζει την εκτέλεση του προγράμματος.

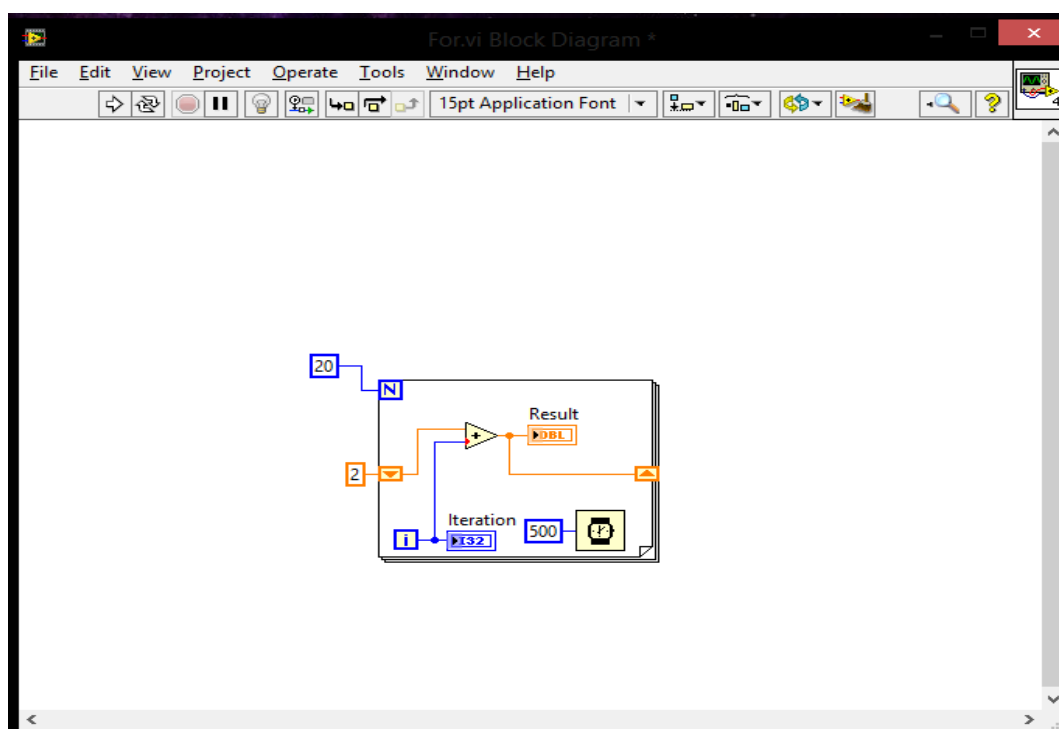
5. Επισημαίνει την ροή εκτέλεσης του προγράμματος στο Block Diagram.
6. Αποθηκεύει προσωρινά τις τιμές των δεδομένων την συγκεκριμένη στιγμή όπου πατήθηκε το κουμπί. Στην συνέχεια υπάρχει η δυνατότητα με ένα probe ο χρήστης να μελετήσει σε κάθε καλώδιο την τιμή που είχε εκείνη την στιγμή.
7. Ανοίγει ένα κόμβο και εκτελεί με κάθε πάτημα του κουμπιού ένα κομμάτι εικονοστοιχείο τη φορά.
8. Εκτελεί ολόκληρο τον κόμβο και σταματάει στον επόμενο.
9. Τελειώνει την εκτέλεση όλων των κόμβων και σταματάει.
10. Αλλάζει το μέγεθος, το χρώμα και το είδος της γραμματοσειράς
11. Ευθυγραμμίζει τα εικονοστοιχεία μεταξύ τους
12. Στοιχίζει τα εικονοστοιχεία μεταξύ τους
13. Αναδιατάσσει τα αντικείμενα μεταξύ τους. Χρησιμοποιείται αυτή η λειτουργία όταν υπάρχει επικάλυψη εικονοστοιχείων.
14. Οργανώνει τον υπάρχοντα κώδικα με τέτοιο τρόπο ώστε να μην καταλαμβάνει μεγάλο χώρο στο Block Diagram. Δεν συνιστάται η χρήση του καθώς επηρεάζει το πόσο ευανάγνωστος είναι ο κώδικας.

2.2.4 Παρουσίαση βασικών προγραμματιστικών δομών

Όπως στην γλώσσα C, έτσι και στο LabVIEW υπάρχουν οι τρεις βασικές προγραμματιστικές δομές For, While και If-else όπου αποτελούν αναπόσπαστο κομμάτι οποιουδήποτε προγράμματος που επιτελεί κάποιας μορφής υπολογισμού. Έτσι θεωρήθηκε απαραίτητο να παρουσιαστούν οι παραπάνω δομές ώστε ο αναγνώστης να πάρει μια καλή ιδέα για το πώς υλοποιούνται στο LabVIEW.

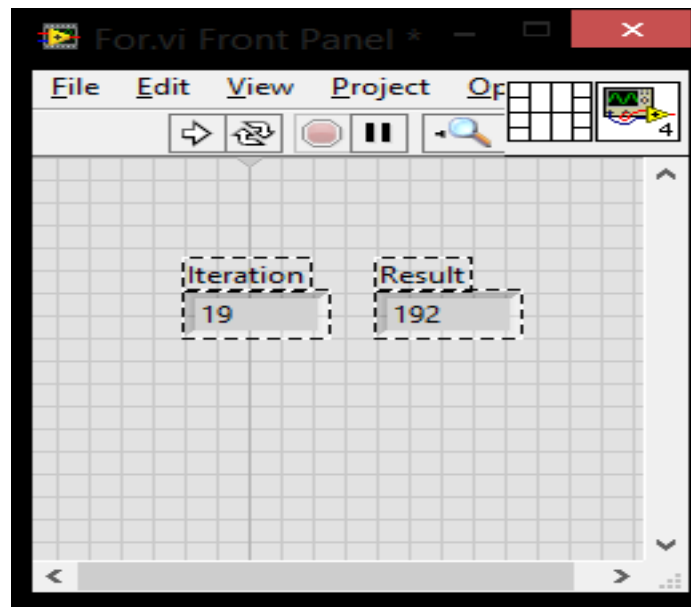
1. For

Η προγραμματιστική δομή For είναι ένας βρόχος επανάληψης που επιτρέπει την επαναλαμβανόμενη εκτέλεση ενός συγκεκριμένου μέρους κώδικα τόσες φορές όσες ορίζει ο χρήστης.



Εικόνα 2.13: Παράδειγμα υλοποίησης της δομής *For*

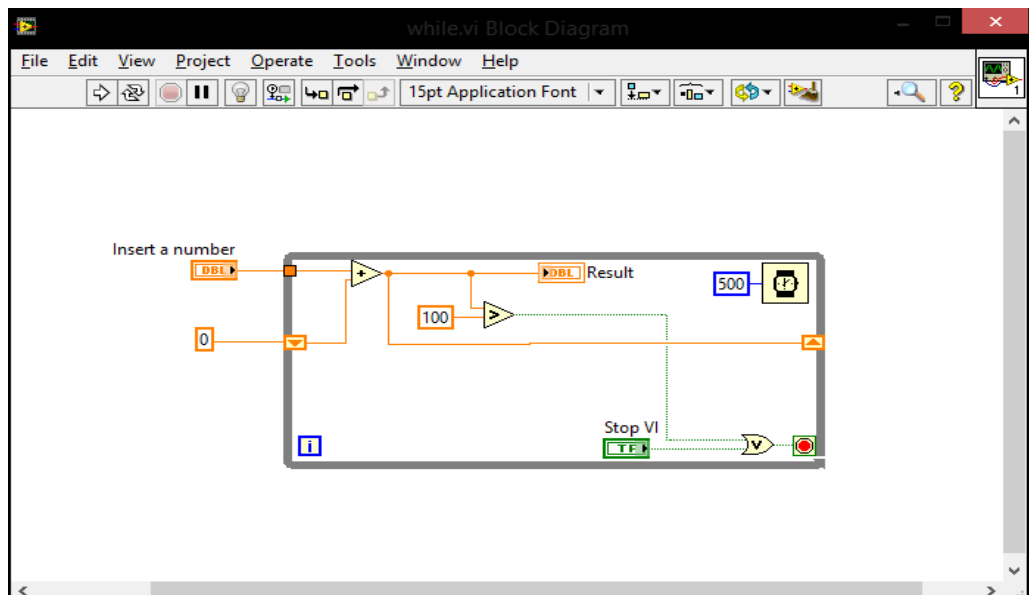
Ο κώδικας που υπάρχει μέσα στο πλαίσιο πρόκειται να εκτελεστεί 20 φορές. Στην πρώτη επανάληψη προστίθεται η αρχική τιμή 2 με το i που δηλώνει το πόσες επαναλήψεις έχουν εκτελεστεί. Την πρώτη φορά εκτέλεσης της επανάληψης το i έχει την τιμή 0. Όπως και στην C έτσι και στο LabVIEW η αρίθμηση των επαναλήψεων των πινάκων και των επαναλήψεων ξεκινάει από το 0. Οπότε το i μέχρι το τέλος της εκτέλεσης του προγράμματος θα πάρει τιμές από το 0 μέχρι και το 19. Στην συνέχεια με την βοήθεια ενός shift register το αποτέλεσμα αποθηκεύεται και ανακτάται κατά την εκτέλεση της δεύτερης επανάληψης, με αποτέλεσμα στην δεύτερη επανάληψη να γίνεται η πρόσθεση του 2 (το αποτέλεσμα της πρώτης πρόσθεσης) με το 1 που δηλώνει την δεύτερη εκτέλεση της επανάληψης. Η συνάρτηση με το ρολόι ονομάζεται Wait και παίρνει ως όρισμα μια τιμή σε millisecond. Μετά την εκτέλεση μιας επανάληψης καθυστερεί την επόμενη επανάληψη 500ms. Έτσι δίνει την δυνατότητα στον χρήστη στο Front Panel να παρακολουθεί τα αποτελέσματα καθώς χωρίς αυτή η ταχύτητα εκτέλεσης του προγράμματος εξαρτάται από την συχνότητα λειτουργίας του επεξεργαστή όπου εκτελείται το πρόγραμμα.



Εικόνα 2.14: Front Panel με τα τελικά αποτελέσματα της επανάληψης For

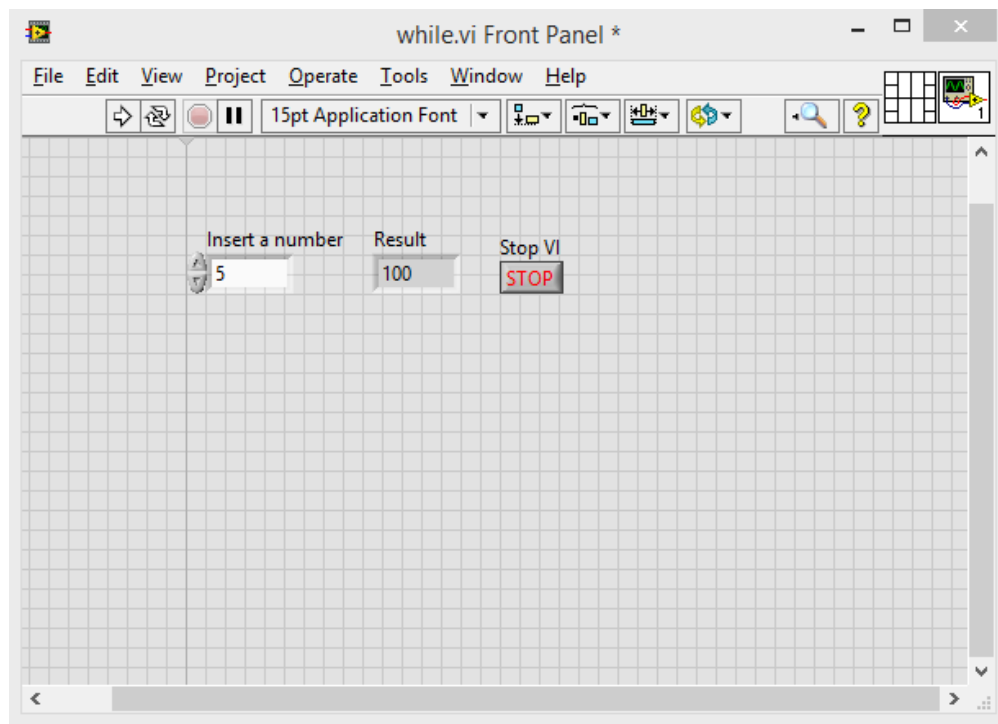
2. While

Η προγραμματιστική δομή While είναι ένας ακόμα βρόχος επανάληψης που επιτρέπει την επαναλαμβανόμενη εκτέλεση ενός συγκεκριμένου μέρους κώδικα όσο η συνθήκη τερματισμού παραμένει αληθής.



Εικόνα 2.13: Παράδειγμα υλοποίησης της δομής While

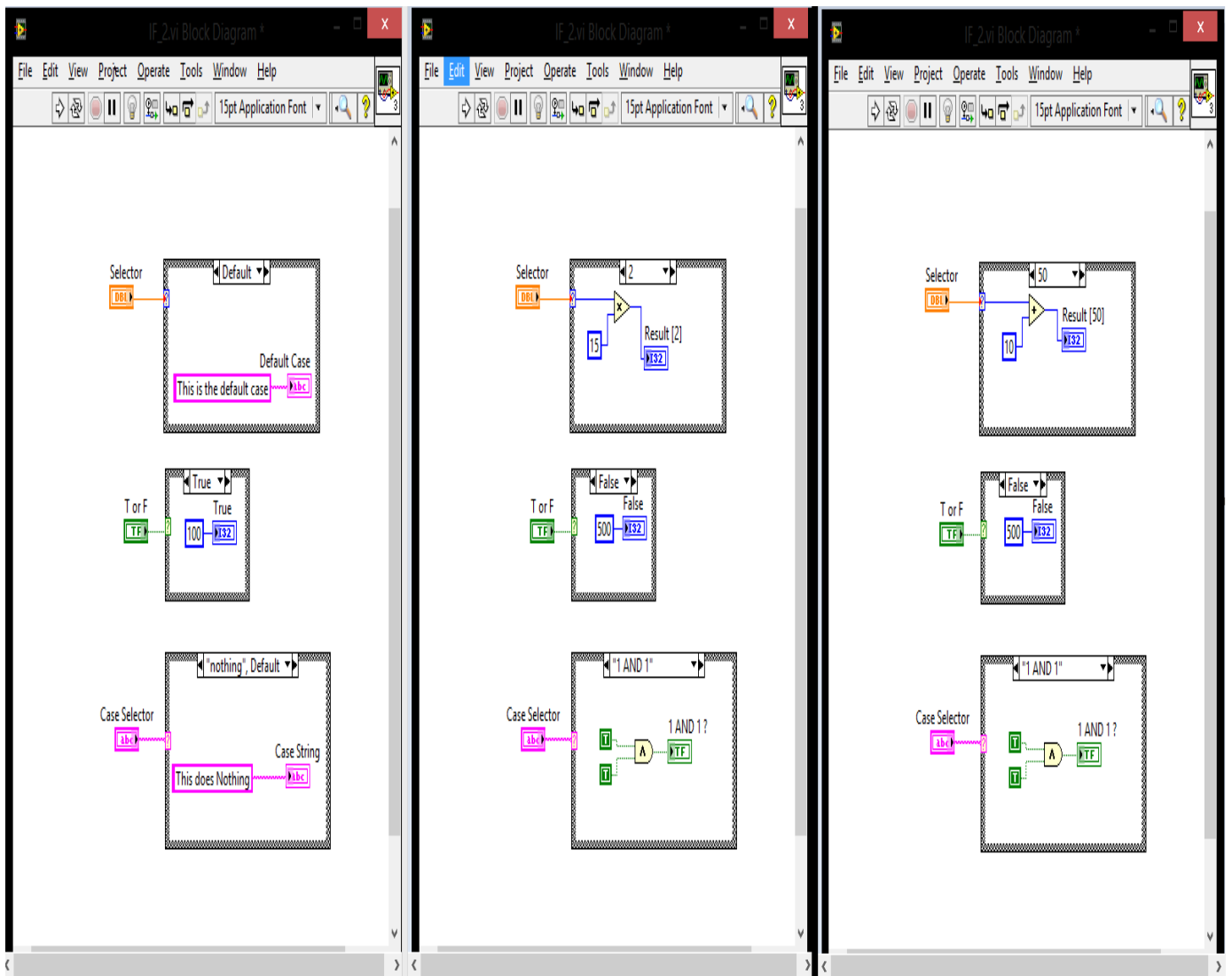
Ο κώδικας στο πλαίσιο προσθέτει σε κάθε επανάληψη της while έναν αριθμό που έθεσε ο χρήστης από το Front Panel μέχρι να το άθροισμα να αυξηθεί περισσότερο από την τιμή 100 ή μέχρι ο χρήστης να πατήσει το κουμπί “Stop VI” ώστε να σταματήσει το πρόγραμμα πριν το άθροισμα ξεπεράσει το 100. Η συνθήκη τερματισμού μπορεί να μετατραπεί έτσι ώστε αντί να τερματίζεται η εκτέλεση της While όταν η συνθήκη γίνει εσφαλμένη, να μπορεί να συνεχίζει όσο είναι αληθής. Αυτό μπορεί να επιτευχθεί αλλάζοντας τις ρυθμίσεις εκτέλεσης της while.



Εικόνα 2.15: *Front Panel με τα τελικά αποτελέσματα της επανάληψης While*

3. IF-Else

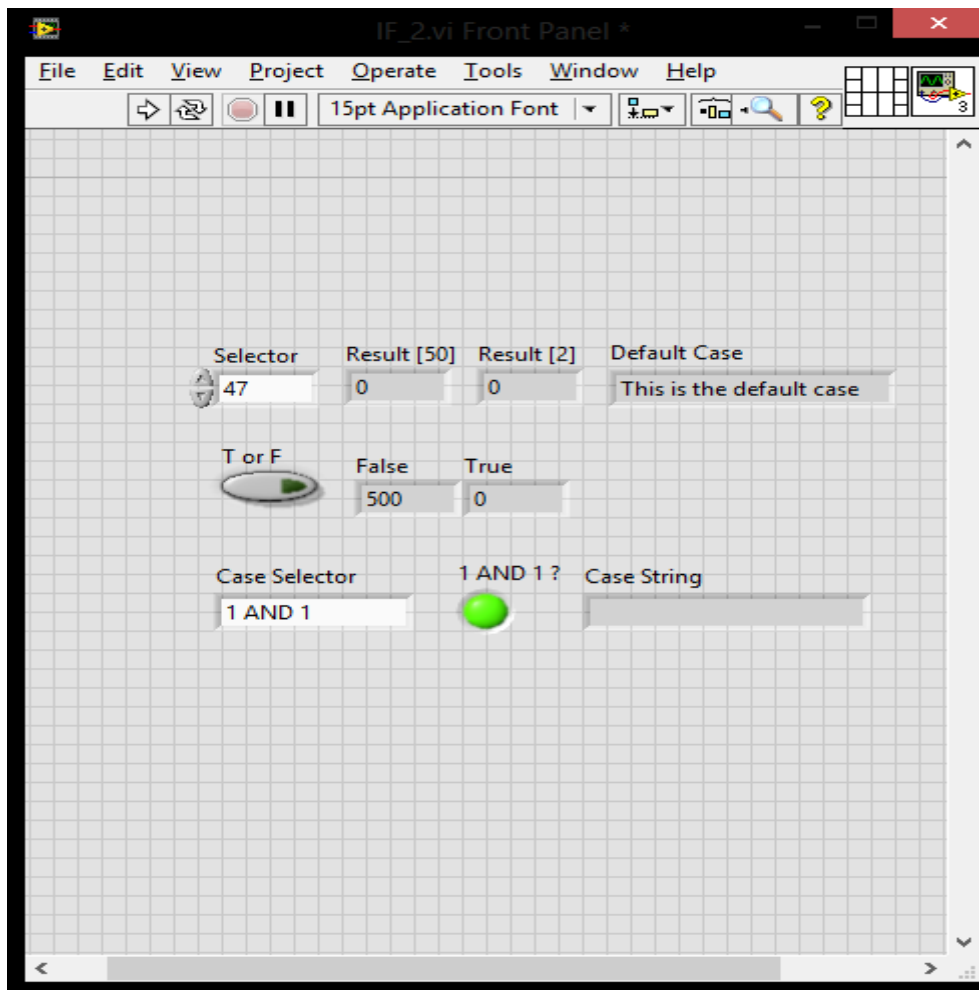
Η συνθήκη ελέγχου If-else επιτρέπει την εκτέλεση διαφορετικού μέρους κώδικα ανάλογα με το αν ισχύει η συνθήκη ελέγχου που καθορίζεται από τον χρήστη. Η συνθήκη If-else στο LabVIEW ονομάζεται “Case Structure” καθώς περιλαμβάνει και την αντίστοιχη επιλογή ελέγχου στη C, Switch Case.



Εικόνα 2.16: Παράδειγμα υλοποίησης της δομής If-Else

Η Case Structure έχει μια είσοδο όπου ονομάζεται “Case Selector”, μέσα από αυτή μπορεί να επιλεγεί κάθε φορά η εκάστοτε περίπτωση όπου θα τρέξει. Σαν είσοδο μπορεί να πάρει έναν αριθμό, μια Boolean μεταβλητή και συμβολοσειρά (string). Στο LabVIEW όποιο κομμάτι κώδικα συμβολίζεται με μπλε χρώμα τότε ο τύπος μεταβλητής είναι 32-bit integer και με πορτοκαλί 64-bit double για πολύ μεγαλύτερη ακρίβεια. Με πράσινο χρώμα ορίζονται οι μεταβλητές Boolean και με ροζ ότι σχετίζεται με τις συμβολοσειρές. Η κάθε δομή ελέγχου μπορεί να έχει από δύο συνθήκες και πάνω και υποχρεωτικά μια από αυτές θα πρέπει να είναι και η προεπιλεγμένη αν καμία από τις συνθήκες δεν επαληθεύεται. Αν στην είσοδο του “Case Selector” υπάρχει Boolean μεταβλητή τότε η προεπιλεγμένη συνθήκη είναι εσφαλμένη. Κάθε μια από τις Case Structures παίρνει ως είσοδο από το Front Panel

δεδομένα από τον χρήστη, κάθε μια αναλύει τα δεδομένα που δέχεται και οδηγεί το αποτέλεσμα στην έξοδο.



Εικόνα 2.17: Front Panel με τα τελικά αποτελέσματα της δομής ελέγχου If-Else

Η πρώτη Case Structure λαμβάνει ως είσοδο την τιμή 47 όπου επειδή δεν ισούται ούτε με το 2 αλλά ούτε με το 50 καλύπτεται από την προεπιλεγμένη συνθήκη όπου και εμφανίζει το μήνυμα “This is the default case”. Στη δεύτερη Case Structure το κουμπί “T or F” δεν χρησιμοποιείται και έτσι στην έξοδο εμφανίζεται η τιμή 500. Ενώ τέλος στην τρίτη Case Structure στην είσοδο της συμβολοσειράς γράφεται “1 AND 1” και ενεργοποιείται το LED.

Κεφάλαιο 3^ο : Ανάλυση κινηματικών εξισώσεων του ρομποτικού οχήματος

3.1 Εισαγωγή

Κινηματική είναι γενικά ο κλάδος της μηχανικής που μελετάει την κίνηση των στερεών σωμάτων χωρίς να αναφέρεται στην μάζα τους ή στις δυνάμεις που προκαλούν την κίνηση. Η κινηματική ανάλυση ενός ρομποτικού οχήματος ασχολείται με την κίνηση των μερών του σε σχέση με ένα παγκόσμιο σύστημα συντεταγμένων, χωρίς να αναφέρεται στις δυνάμεις που προκαλούν την κίνησή του. Η ρομποτική κινηματική αναφέρεται στην κίνηση του ρομπότ μέσα στο χώρο και στις σχέσεις που προκύπτουν μεταξύ των μεταβλητών του.

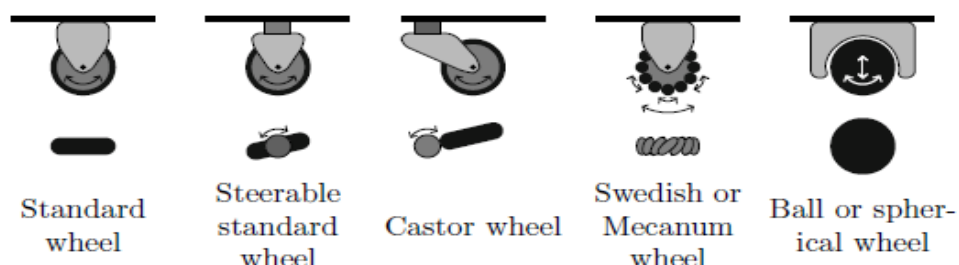
Η κινηματική διακρίνεται σε δύο κλάδους:

1. Στην ορθή κινηματική που επιτρέπει στον προσδιορισμό της θέσης και της ταχύτητας του ρομπότ γνωρίζοντας την κίνηση των τροχών του.
2. Στην αντίστροφη κινηματική, όπου γνωρίζοντας την ταχύτητα και την θέση του ρομπότ σε μια συγκεκριμένη στιγμή προσδιορίζονται οι παράμετροι της κίνησης των τροχών.

Έτσι μπορεί κανείς να συμπεραίνει πως ανάλογα την γεωμετρία, το είδος της κίνησης και των τροχών κάθε ρομπότ έχει διαφορετικές κινηματικές εξισώσεις.

3.2 Είδη βασικών τροχών

Οι βασικοί τύποι τροχών που μπορεί κάποιο ρομπότ να εξοπλιστεί απεικονίζονται στην εικόνα 3.1.



Εικόνα 3.1: Βασικά είδη τροχών

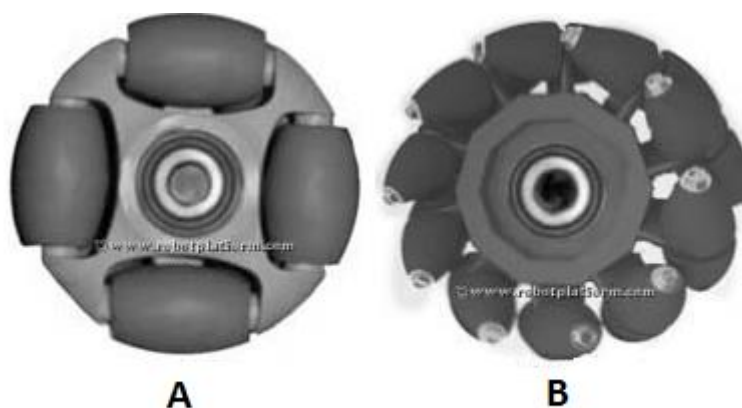
Η κινηματική των διαφόρων τροχών διαφέρει πολύ και ως εκ τούτου η επιλογή των τροχών καθορίζει σε μεγάλο βαθμό και την συνολική κινηματική του ρομποτικού οχήματος.

Οι δύο πρώτοι τροχοί, ο τυπικός τροχός και ο τυπικός κατευθυνόμενος τροχός έχουν έναν κύριο άξονα περιστροφής. Έχουν δύο βαθμούς ελευθερίας και μπορούν να οδηγούν το ρομποτικό όχημα εμπρός ή πίσω. Το κέντρο του τροχού είναι στερεωμένο στο πλαίσιο του ρομπότ, όπως φαίνεται και στην εικόνα 3.1. Η γωνία μεταξύ του πλαισίου του ρομπότ και του επιπέδου των τροχών είναι σταθερή.

Το τρίτο είδος βασικού τροχού ονομάζεται προσανατολιζόμενος τροχός (Castor Wheel). Ο κατακόρυφος άξονας του τροχού, δεν διέρχεται από το κέντρο του, αλλά είναι ελαφρώς μετατοπισμένος εκτός του κεντρικού άξονα. Κάποιες υλοποιήσεις του τροχού περιλαμβάνουν έναν περιστρεφόμενο σύνδεσμο μεταξύ του τροχού και του πιρουνιού ώστε να μπορεί να περιστρέφεται ελεύθερα με 360° ελευθερίας. Ένα από τα μεγαλύτερα μειονεκτήματα αυτών των περιστρεφόμενων τροχών είναι πτερυγισμός (flutter). Το φαινόμενο του πτερυγισμού εμφανίζεται όταν ο τροχός χάνει την επαφή του με το έδαφος και περιστρέφεται ελεύθερα προς οποιαδήποτε κατεύθυνση. Αυτό έχει ως αποτέλεσμα να μην είναι ο πλέον κατάλληλος τύπος τροχού αν χρησιμοποιούνται οπτικοί κωδικοποιητές για την συλλογή δεδομένων, καθώς από την στιγμή που ο τροχός θα έχανε την επαφή με το έδαφος θα πρόκυπτε σφάλμα στις μετρήσεις. Τέλος κατά την επαφή με το έδαφος, ο τροχός μπορεί να προσανατολίζεται προς οποιαδήποτε κατεύθυνση και να κινείται προς οποιαδήποτε κατεύθυνση. Τέτοιου είδους τροχοί συναντιούνται και στις καρέκλες γραφείου.

Το τέταρτο είδος τροχού ονομάζεται πολυ-κατευθυντικός τροχός (omnidirectional wheel) και χωρίζεται σε δύο είδη με μια βασική διαφορά. Δίνει στο ρομπότ την δυνατότητα κινείται σε οποιαδήποτε κατεύθυνση καθώς αποτελούνται από έναν κανονικό τροχό με παθητικούς τροχούς (κυλίνδρους) προσαρτημένους στην επιφάνεια του κεντρικού τροχού. Οι κύλινδροι είναι στερεωμένοι κατά τέτοιο τρόπο ώστε ο άξονας των μικρών τροχών να είναι κάθετος προς τον άξονα του μεγαλύτερου κεντρικού τροχού, ο οποίος καθιστά τον τροχό να περιστρέφεται παράλληλα με τον δικό του άξονα Έτσι παρέχεται η δυνατότητα στο ρομποτικό όχημα να κινείται σε οποιαδήποτε κατεύθυνση με χαμηλή αντίσταση. Οι

Mechanum τροχοί έχουν προσαρτημένους τους κύλινδρους με γωνία 45° γύρω από την περιφέρεια του κεντρικού τροχού ενώ οι Swedish με γωνία 90° .



Εικόνα 3.2: A) *Mechanum Wheel* B) *Swedish Wheel*

Τέλος, ο σφαιρικός τροχός αποτελείται από μια σφαίρα από μέταλλο ή πλαστικό τοποθετημένη μέσα σε μια θήκη. Η μπάλα έχει 360° ελευθερίας και κανονικά χρησιμοποιείται σαν βοηθητικός τροχός για την εξισορρόπηση ενός ρομπότ και όχι σαν κινητήριος τροχός. Το μειονέκτημα είναι ότι αυτός ο τροχός έχει υψηλή πρόσφυση και απαιτεί περισσότερη δύναμη για να σπρώξει και να στηρίξει τους κινητήριους τροχούς. Επίσης είναι ακατάλληλος για ανώμαλες, σκονισμένες και λιπαρές επιφάνειες.

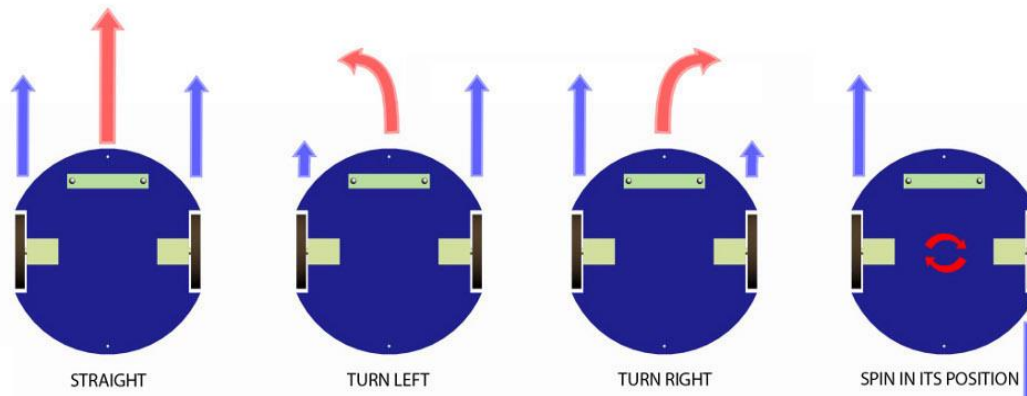
3.3 Θεωρία ελέγχου τροχών

Είναι ο τρόπος με τον οποίο ορίζεται το πως θα οδηγείται και θα κατευθύνεται το ρομποτικό όχημα στο χώρο. Υπάρχουν αρκετοί τρόποι ελέγχου αλλά θα παρουσιαστούν οι τρεις βασικοί όπου θα αναλυθούν παρακάτω.

Differential Drive – Διαφορική Οδήγηση: Είναι ο πιο κοινός μηχανισμός ελέγχου στα ρομποτικά οχήματα. Η διαφορά ταχύτητας μεταξύ δύο μοτέρ οδηγεί το ρομποτικό όχημα σε οποιαδήποτε απαιτούμενη διαδρομή και κατεύθυνση. Ανάλογα με τις γωνιακές ταχύτητες που εφαρμόζονται στους τροχούς μπορεί να συμβεί μια από τις περιπτώσεις που ακολουθούν.

1. Εάν οι γωνιακές ταχύτητες έχουν την ίδια τιμή και το πρόσημο, δηλαδή εάν και οι δύο τροχοί κινούνται με την ίδια ταχύτητα και την ίδια κατεύθυνση (είτε δεξιόστροφα είτε αριστερόστροφα) τότε το ρομπότ τείνει να περιστρέφεται γύρω από τον κάθετο άξονά του. Αυτή η πλήρης δυνατότητα περιστροφής είναι ένα από τα μεγαλύτερα πλεονεκτήματα ενός ρομπότ με διαφορετική κίνηση και του επιτρέπει να εκτελεί στροφές με μηδενική ακτίνα.
2. Οι γωνιακές ταχύτητες έχουν την ίδια τιμή αλλά διαφορετικό πρόσημο, δηλαδή αν και οι δύο τροχοί κινούνται με την ίδια ταχύτητα αλλά στην αντίθετη κατεύθυνση (Ένα δεξιόστροφο και άλλο αριστερόστροφο), τότε το ρομπότ είναι πιο πιθανό να ακολουθήσει μια γραμμική διαδρομή, είτε προς τα εμπρός είτε προς τα πίσω με βάση την περιστροφή των μοτέρ.
3. Οι γωνιακές ταχύτητες έχουν διαφορετικές τιμές με ίδιο ή διαφορετικό πρόσημο, δηλαδή εάν οι τροχοί κινούνται με διαφορετικές ταχύτητες στην ίδια κατεύθυνση ή προς την αντίθετη κατεύθυνση, τότε το ρομπότ κάνει καμπύλη κίνηση. Τέλος, εάν ένας από τους τροχούς περιστρέφεται και ο άλλος παραμένει ακίνητος τότε το ρομπότ κάνει σχεδόν στροφή 90° .

Ένα από τα σημαντικότερα μειονεκτήματα του διαφορικού ελέγχου είναι ότι το ρομπότ δεν συμπεριφέρεται ακριβώς όπως αναμένει κανείς. Δεν οδηγεί κατά μήκος μιας ευθείας γραμμής ούτε στρέφεται ακριβώς στις αναμενόμενες γωνίες, ειδικά όταν γίνεται χρήση μοτέρ συνεχούς ρεύματος (DC). Αυτό οφείλεται στη διαφορά στον αριθμό περιστροφών του κάθε τροχού σε δεδομένο χρονικό διάστημα. Για να αντιμετωπιστεί αυτό το πρόβλημα, πρέπει να υπολογιστεί ένας διορθωτικός παράγοντας στην γωνιακή ταχύτητα του κινητήρα. Παραδείγματος χάρη, αν το ρομπότ σε μια ευθεία διαδρομή και το ρομπότ παρεκκλίνει προς μια πλευρά τότε θα πρέπει να υπολογιστεί ένας διορθωτικός παράγοντας ώστε να μειώσει την γωνιακή ταχύτητα του μοτέρ που είναι υπεύθυνος για αυτή την απόκλιση.



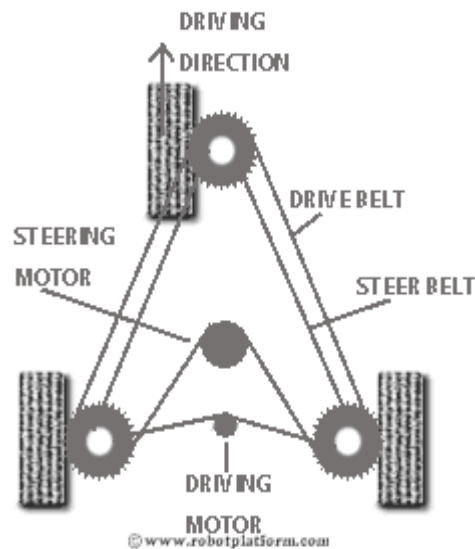
Εικόνα 3.3: Διαφορετικές κινήσεις ρομπότ με differential drive

Ackerman Steering – Οδήγηση Ackerman: Η διαμόρφωση αυτή συναντάται και στα αυτοκίνητα αλλά και στα ρομποτικά οχήματα. Συντονίζει μηχανικά τη γωνία των δύο εμπρόσθιων τροχών, που είναι στερεωμένοι σε έναν κοινό άξονα, που χρησιμοποιείται για την μεταβολή της κατεύθυνσης και δύο τροχούς στο πίσω μέρος του ρομπότ που είναι στερεωμένοι σε άλλο άξονα για την μετάδοση της κίνησης. Το σύστημα Ackerman έχει σχεδιαστεί με τέτοιο τρόπο, ώστε όταν υπάρχει στροφή, το εσωτερικό ελαστικό στρέφεται με μεγαλύτερη γωνία από το εξωτερικό ελαστικό για να αποφευχθεί η ολίσθηση των ελαστικών. Τα πλεονεκτήματα του σχεδιασμού αυτού είναι ο αυξημένος έλεγχος, η καλύτερη σταθερότητα και η ευελιξία στο πεδίο δράσης του ρομπότ. Επιπλέον προσφέρει μικρότερη ολίσθηση και λιγότερη κατανάλωση ενέργειας σε σχέση με το differential drive καθώς είναι απαραίτητο μόνο ένα μοτέρ για την κίνηση του ρομποτικού οχήματος.



Εικόνα 3.4: Ρομποτικό όχημα με Ackerman steering

Synchronous Drive – Συγχρονισμένη Οδήγηση: Αν το έδαφος στο οποίο κινείται το ρομπότ είναι επίπεδο τότε ο σχεδιασμός αυτός μπορεί να θεωρηθεί ως ο καλύτερος, καθώς η πλοήγηση που προσφέρει στο ρομπότ είναι πολύ ακριβής. Το ρομπότ μπορεί να κατασκευαστεί με τρεις, τέσσερις ή περισσότερους τροχούς, αλλά όλοι οι τροχοί θα πρέπει να είναι από κοινού συνδεδεμένοι και να περιστρέφονται μαζί με την ίδια ταχύτητα και να έχουν την ίδια κατεύθυνση. Λόγω της αυξημένης πολυπλοκότητας ο αριθμός των τροχών παραμένει ο ελάχιστος. Έτσι τα περισσότερα ρομπότ που κάνουν χρήση αυτού του ελέγχου εξοπλίζονται με τρεις τροχούς. Συχνά τα ρομπότ αυτά χρησιμοποιούν ένα μόνο μοτέρ το οποίο μέσω ενός ιμάντα μεταφέρει την κίνηση στους τροχούς. Τέλος, τα ρομπότ που χρησιμοποιούν ως επί το πλείστον αυτού του είδους τον έλεγχο είναι οι ρομποτικές σκούπες.



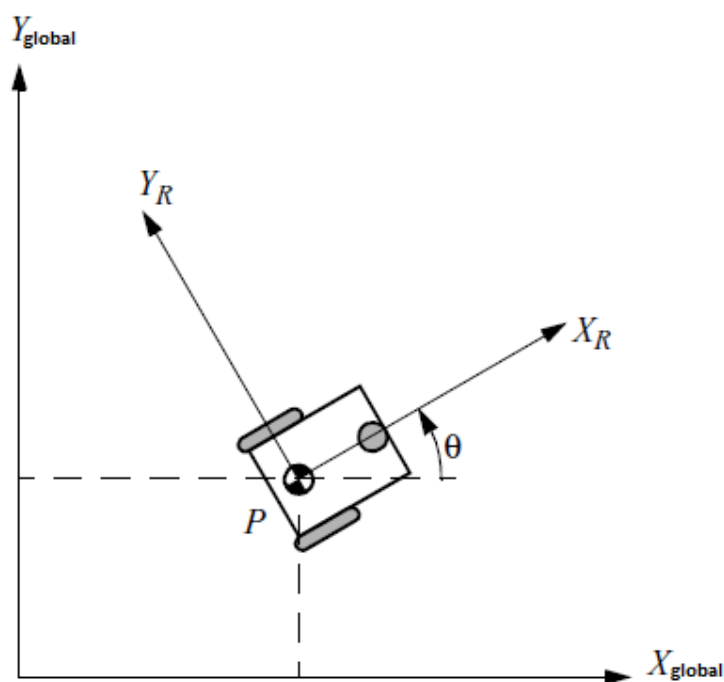
Εικόνα 3.5: Ρομποτικό όχημα με *Synchronous Drive*

3.4 Παρουσίαση και ανάλυση κινηματικών εξισώσεων για το ρομποτικό όχημα NI Starter Kit

Το ρομποτικό όχημα NI Starter Kit υιοθετεί το έλεγχο Differential Drive καθώς εξοπλίζεται με τέσσερις τυπικούς τροχούς όπου τοποθετούνται σε δύο άξονες πάνω στο πλαίσιο του. Η ακτίνα των τροχών μετρήθηκε στα $R= 4.9$ εκατοστά ενώ η απόσταση των δύο αντικριστών τροχών στα $l= 36.9$ εκατοστά.

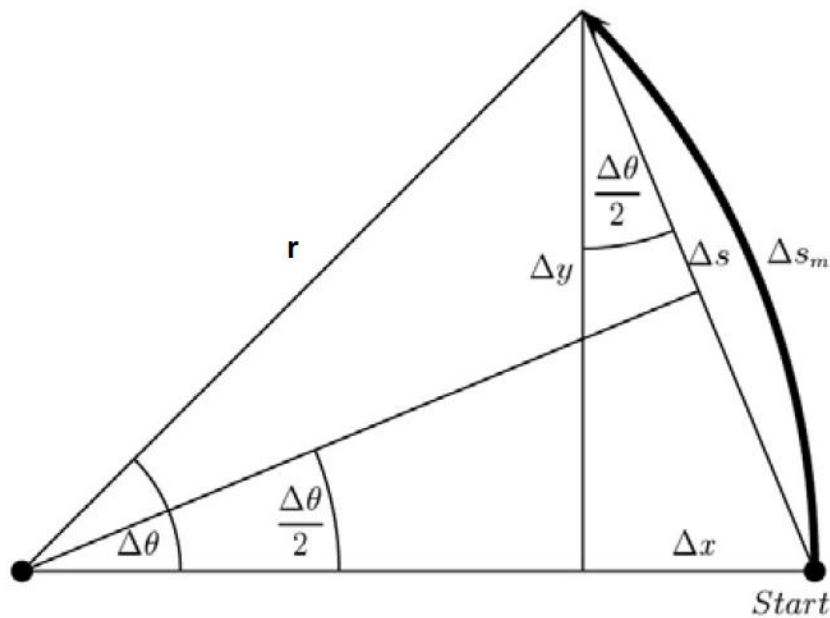
Αρχικά πρέπει να ορισθεί ένα παγκόσμιο σύστημα συντεταγμένων X - Y στο οποίο βρίσκεται το ρομπότ. Για να καθορισθεί η θέση του ρομπότ, επιλέγεται ένα σημείο P στο πλαίσιο (σασί) του ως το σημείο αναφοράς της θέσης. Στο σημείο P σχηματίζεται ένα δεύτερο σύστημα συντεταγμένων, το οποίο σχηματίζει γωνία θ με το παγκόσμιο σύστημα. Με αποτέλεσμα η θέση του σημείου P στο παγκόσμιο σύστημα συντεταγμένων να καθορίζεται από τις συντεταγμένες x και y και την γωνία θ . Η θέση του ρομπότ κάθε στιγμή περιγράφεται από ένα διάνυσμα με τρία στοιχεία.

$$\text{Pose (θέση)} = \begin{bmatrix} x \\ y \\ \theta \end{bmatrix}$$



Εικόνα 3.6: Παγκόσμιο σύστημα συντεταγμένων και σύστημα συντεταγμένων του ρομπότ

Ένα ρομπότ με differential drive αποτελείται από δύο ανεξάρτητα κινούμενους τροχούς με αποτέλεσμα να είναι δυνατές μόνο κυκλικές τροχιές. Επομένως δύο είναι οι ιδιαίτερες περιπτώσεις, η ακτίνα της κυκλικής τροχιάς να είναι ίση με $r = 0$ είτε $r = \infty$. Το πρώτο ενδεχόμενο είναι το ρομπότ να κάνει μια επιτόπια περιστροφή επί τόπου, ενώ δεύτερο το ρομπότ να διαγράφει μια ευθεία διαδρομή. Στο μέσο των κινητήριων τροχών βρίσκεται το κινηματικό κέντρο του ρομπότ. Λαμβάνοντας υπόψη τις γραμμικές ταχύτητες U_{left} και U_{right} του αριστερού και του δεξιού τροχού αντίστοιχα, και ένα χρονικό διάστημα Δt , μπορεί να υπολογιστεί το διάστημα ΔS_m που διένυσε ο κάθε τροχός και το κινηματικό κέντρο.



Εικόνα 3.7: Γεωμετρική λύση της κινηματικής ανάλυσης του Differential Drive

Έτσι προκύπτουν τα διαστήματα που διένυσαν ο αριστερός και δεξιός τροχός.

$$\begin{aligned}\Delta s_{left} &= U_{left} \cdot \Delta t \\ \Delta s_{right} &= U_{right} \cdot \Delta t \\ \Delta s_m &= \frac{(\Delta s_{left} + \Delta s_{right})}{2}\end{aligned}\quad (3.1)$$

Οι γραμμικές ταχύτητες μπορούν να εκφραστούν ως:

$$U_{left} = R * \Omega_{left} = R * \dot{\varphi}_{left}$$

$$U_{right} = R * \Omega_{right} = R * \dot{\varphi}_{right}$$

Η γωνιακή ταχύτητα του ρομπότ κατά την διεύθυνση της κίνησης του προκύπτει από την σχέση 3.1 αν διαιρεθεί με το χρονικό διάστημα Δt .

$$U_{robot} = \frac{\Delta s_m}{\Delta t} = \frac{(\Delta s_{left} + \Delta s_{right})}{2\Delta t} = \frac{R * \dot{\varphi}_{left} * \Delta t + R * \dot{\varphi}_{right} * \Delta t}{2\Delta t}$$

$$U_{robot} = R \frac{(\dot{\varphi}_{left} + \dot{\varphi}_{right})}{2} \quad (3.2)$$

Η ταχύτητα του ρομπότ εκφράζεται σαν γωνιακή ώστε να μην χρειαστεί να μετατραπεί κατά την εκτέλεση του κώδικα. Διότι το ρομπότ διαβάζει τις γωνιακές ταχύτητες μέσω των οπτικών κωδικοποιητών.

Κατά την κίνηση του ρομπότ διαγράφεται ένα κυκλικό τόξο με ακτίνα r όπως φαίνεται και στην εικόνα 3.7. Για να υπολογιστεί η ακτίνα πρέπει να ληφθεί υπόψη το διάστημα Δs_m και η απόσταση l των δύο τροχών μεταξύ τους.

$$r = l * \frac{\Delta s_{left} + \Delta s_{right}}{\Delta s_{left} - \Delta s_{right}}$$

Έχοντας ως δεδομένη την ακτίνα μπορεί να υπολογιστεί η μεταβολή της γωνίας θ , $\Delta\theta$. Το πρόσημο του $\Delta\theta$ εξαρτάται από το τι στροφή πραγματοποιεί το ρομποτικό όχημα, αριστερόστροφη ή δεξιόστροφη.

$$\Delta\theta = \frac{\Delta s_m}{r} = \frac{\Delta s_{left} - \Delta s_{right}}{l} \quad (3.3)$$

Αντικαθιστώντας τα διαστήματα Δs_{left} και Δs_{right} με τις γωνιακές ταχύτητες η εξίσωση 3.3 ισοδυναμεί

$$\Delta\theta = \frac{R}{l} * (\varphi_{\text{left}} - \varphi_{\text{right}}) \quad (3.4)$$

Για τον υπολογισμό των μετατοπίσεων του ρομποτικού οχήματος στο παγκόσμιο σύστημα αναφοράς, οι εξισώσεις που προκύπτουν είναι :

$$\Delta x = U_{Rx} * \Delta t = U_R * \cos\left(\frac{\Delta\theta}{2} + \theta_0\right) * \Delta t \quad (3.5)$$

$$\Delta y = U_{Ry} * \Delta t = U_R * \sin\left(\frac{\Delta\theta}{2} + \theta_0\right) * \Delta t \quad (3.6)$$

Όπου θ_0 είναι η αρχική γωνία θ .

Τέλος, αν θέλει κανείς να υπολογίσει λόγω χάρη τον τελικό προσανατολισμό και μετατόπιση του ρομπότ για ένα διάστημα $\Delta t = 5$, δεν έχει παρά μόνο σε κάθε επόμενο Δt να προσθέτει τις προηγούμενες μετατοπίσεις και προσανατολισμό στις επόμενες. Έτσι οι εξισώσεις γίνονται στο πρώτο βήμα

$$\theta_1 = \theta_0 + \Delta\theta = \theta_0 + \left[\frac{R}{l} * (\varphi_{\text{left}} - \varphi_{\text{right}})\right]$$

$$x_1 = x_0 + \Delta x = x_0 + U_R * \cos\left(\frac{\Delta\theta}{2} + \theta_0\right) * \Delta t$$

$$y_1 = y_0 + \Delta y = y_0 + U_R * \sin\left(\frac{\Delta\theta}{2} + \theta_0\right) * \Delta t$$

Ενώ στο δεύτερο βήμα, αντίστοιχα

$$\theta_2 = \theta_1 + \Delta\theta = \theta_1 + \left[\frac{R}{l} * (\varphi_{\text{left}} - \varphi_{\text{right}})\right]$$

$$x_2 = x_1 + \Delta x = x_1 + U_R * \cos\left(\frac{\Delta\theta}{2} + \theta_1\right) * \Delta t$$

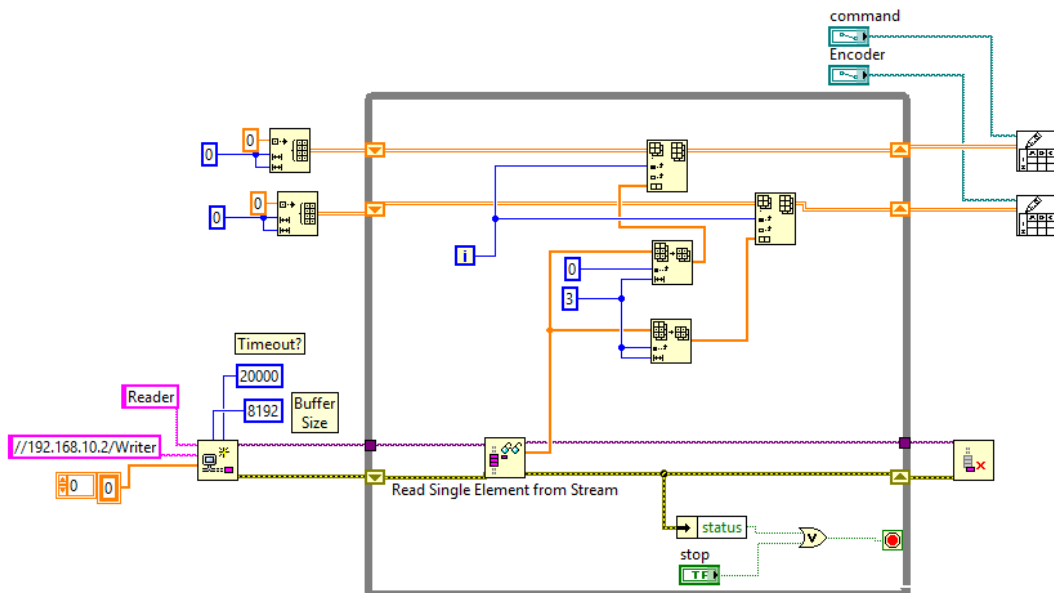
$$y_2 = y_1 + \Delta y = y_1 + U_R * \sin\left(\frac{\Delta\theta}{2} + \theta_1\right) * \Delta t$$

και ούτω καθεξής στα επόμενα βήματα.

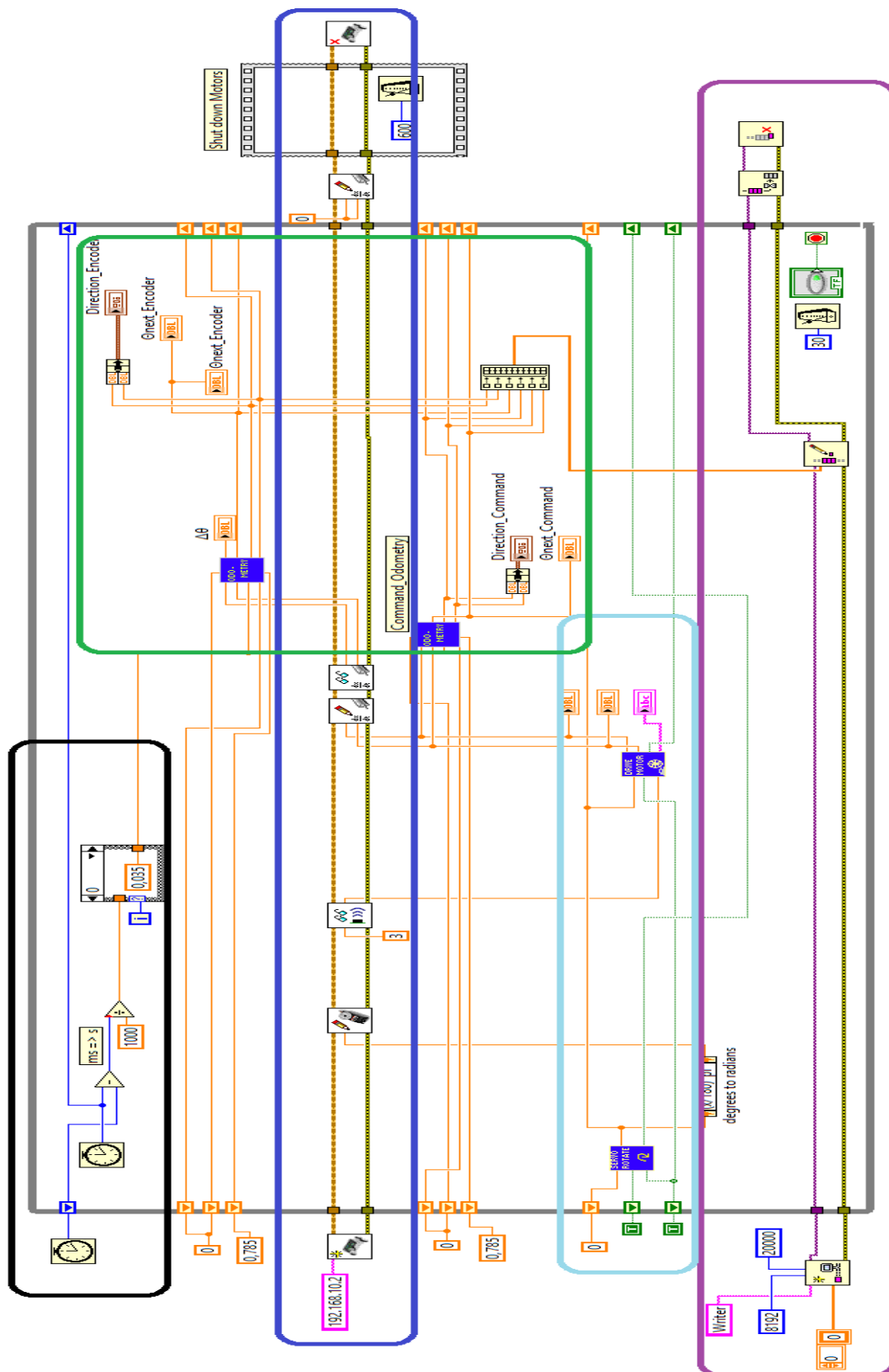
Κεφάλαιο 4^ο : Παρουσίαση και ανάλυση του προγράμματος

4.1 Εισαγωγή

Στο παρόν κεφάλαιο θα αναλυθεί ο κώδικας που συγγράφηκε και εκτελείται από την μεριά του ρομποτικού οχήματος για την περιπλάνηση, αποστολή και καταγραφή των μετρήσεων καθώς και ο κώδικας που εκτελείται από την μεριά του host υπολογιστή και είναι υπεύθυνος για την αποθήκευση των μετρήσεων σε αρχείο. Ο κώδικας που εκτελείται στο ρομποτικό όχημα αποτελείται από τρία subVIs, τα οποία είναι υπεύθυνα για την κίνηση του αισθητήρα υπερήχων, την λήψη αποφάσεων όταν εντοπιστεί εμπόδιο και την οδομετρία του οχήματος. Τα subVIs, που σχεδιάστηκαν για την πτυχιακή εργασία, μαζί με άλλα βασικά VIs της NI εκτελούνται σε ένα ατέρμονα βρόγχο while, ο οποίος τερματίζεται μόνο από τον χρήστη. Ο κώδικας θα μπορούσε να χωριστεί σε πέντε κομμάτια για την καλύτερη επεξήγηση και κατανόηση του αναγνώστη (Εικόνα 4.2). Το πρώτο κομμάτι (με μαύρο χρώμα) είναι υπεύθυνο να υπολογίζει το χρονικό διάστημα Δt όπου χρησιμοποιείται για τον υπολογισμό των διάφορων μεγεθών αργότερα.



Εικόνα 4.1: Κώδικας που από τον host υπολογιστή



Εικόνα 4.2: Κώδικας που εκτελείται στο ρομποτικό όχημα

Το δεύτερο κομμάτι (με σκούρο μπλε) αποτελείται από τα VI που παρέχει η NI για την αρχικοποίηση της ζεύξης επικοινωνίας με το FPGA, για τον έλεγχο του

servomotor που κινεί τον αισθητήρα υπερήχων , για τον έλεγχο των μοτέρ κίνησης, για την ανάγνωση δεδομένων από τους αισθητήρες και τα μοτέρ και τον τερματισμό της επικοινωνίας με το ρομποτικό όχημα μέσω του FPGA. Το τρίτο κομμάτι (με γαλάζιο) απαρτίζεται από δύο subVIs. Το πρώτο, με όνομα Servo Rotate, ελέγχει τον τρόπο με τον οποίο το servomotor κινεί τον αισθητήρα υπερήχων και πως αυτός θα συμπεριφέρεται στον εντοπισμό ενός εμποδίου. Ενώ το δεύτερο, Drive Motor, αναλύει τις πληροφορίες που δέχεται κάθε στιγμή από το Servo Rotate και αποφασίζει σε κάθε εκτέλεση του βρόγχου ποια κίνηση θα ακολουθήσει το ρομποτικό όχημα. Το τέταρτο κομμάτι (με πράσινο) είναι υπεύθυνο για την καταγραφή των δεδομένων και την παρουσίαση τους στο front panel. Περιλαμβάνει και το τρίτο subVI με όνομα Odometry όπου περιλαμβάνει τις εξισώσεις της κινηματικής που αναλύθηκαν στο προηγούμενο κεφάλαιο. Τέλος στο πέμπτο κομμάτι κώδικα (με μωβ) υλοποιείται η σύνδεση του VI που εκτελείται στον host υπολογιστή με το VI που εκτελείται στο ρομποτικό όχημα ώστε να μπορεί να γίνει η αποθήκευση των δεδομένων στον υπολογιστή.

Από την μεριά του υπολογιστή, ο κώδικας που εκτελείται είναι υπεύθυνος για την δημιουργία δύο αρχείων στα οποία αποθηκεύονται τα δεδομένα της μετατόπισης και της γωνίας θ που υπολογίστηκαν με γωνιακές ταχύτητες που μετρήθηκαν από τα μοτέρ καθώς και τα αντίστοιχα δεδομένα που υπολογίστηκαν από τις γωνιακές ταχύτητες που δόθηκαν ως ορίσματα.

Συνοπτικά η εκτέλεση του προγράμματος ακολουθεί την παρακάτω σειρά:

Αρχικοποίηση:

1. Επιστρέφεται μια χρονική στιγμή από το αντικείμενο Tick Count για τον υπολογισμό του Δt .
2. Ορίζονται οι αρχικές τιμές στους Shift Registers, που είναι υπεύθυνοι για την μεταφορά των μετρήσεων των κωδικοποιητών από το ένα βρόγχο στον επόμενο.
3. Ορίζεται η IP του SbRIO-9631 ώστε αυτό να συνδεθεί με τον host υπολογιστή

4. Ορίζονται οι αρχικές τιμές στους Shift Registers, που είναι υπεύθυνοι για την μεταφορά των μετρήσεων που προκύπτουν από τον υπολογισμό της οδομετρίας με τις γωνιακές ταχύτητες που ορίζονται από το πρόγραμμα.
5. Αρχικοποιούνται οι είσοδοι του Servo Rotate subVI (Περισσότερα για το subVI θα αναφερθούν παρακάτω)
6. Δίνονται αρχικές τιμές για την δημιουργία του Writer Stream Endpoint VI το οποίο θα αναμένει να δημιουργηθεί ζεύξη από το Reader Stream Endpoint VI που βρίσκεται στο Read Stream VI και εκτελείται στον host υπολογιστή.

Εκτέλεση:

1. Υπολογίζεται το Δt .
2. Υπολογίζεται η επόμενη γωνία στο subVI Servo Rotate, στην οποία θα οδηγηθεί το servomotor.
3. Η γωνία αυτή μετατρέπεται σε rad και δίνεται σαν όρισμα στο Write Sensor Servo VI όπου οδηγεί το servomotor σε αυτή και κατά συνέπεια και τον αισθητήρα υπερήχων.
4. Το Read Ping Servo Angle VI καταγράφει την απόσταση από τον αισθητήρα υπερήχων και οδηγεί την μέτρηση στο Drive Motor subVI.
5. Το Drive Motor subVI ανάλογα με τα δεδομένα που παίρνει ως ορίσματα από το Servo Rotate και το Read Ping Servo Angle αποφασίζει την επόμενη κίνηση του ρομποτικού οχήματος ώστε να αντιμετωπίσει κάποιο πιθανό εμπόδιο. Το VI σε κάθε εκτέλεση του δίνει ως έξοδο δύο γωνιακές ταχύτητες όπου θα εφαρμοστούν στα μοτέρ. Επιπλέον οι τιμές αυτές οδηγούνται και στο Odometry subVI ώστε να υπολογισθούν οι μετατοπίσεις κατά τους άξονες X και Y καθώς και η γωνία θ .
6. Οι γωνιακές ταχύτητες που υπολογίστηκαν εφαρμόζονται στα μοτέρ με την βοήθεια του Motor Velocity Setpoints VI.
7. Το Read Motor Velocity Setpoints VI επιστρέφει λόγω των κωδικοποιητών τις τρέχουσες γωνιακές ταχύτητες των μοτέρ και στην συνέχεια τις οδηγεί στο Odometry subVI ώστε να υπολογισθούν οι μετατοπίσεις κατά τους άξονες X και Y καθώς και η γωνία θ .
8. Στην συνέχεια τα αποτελέσματα που παράχθηκαν από τα δύο Odometry subVIs παρουσιάζονται σε αντίστοιχα διαγράμματα καθώς και

ομαδοποιούνται σε ένα πίνακα όπου σε κάθε επανάληψη του βρόγχου αποστέλλεται η τελευταία γραμμή στον host υπολογιστή ώστε να αποθηκευτούν σε αρχεία.

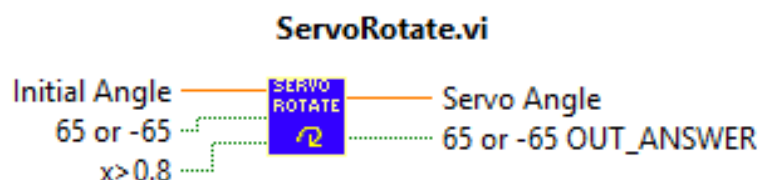
Τερματισμός:

1. Με τον τερματισμό του βρόγχου, το Flush Stream VI αποστέλλει όσες εγγραφές του πίνακα δεν πρόλαβαν να φτάσουν στον host υπολογιστή.
2. Τερματίζεται η σύνδεση του Read Stream VI που εκτελείται στον host υπολογιστή με το SbRIO-9631.
3. Ορίζονται μηδενικές γωνιακές ταχύτητες στα μοτέρ ώστε σταματήσει να κινείται ρομποτικό όχημα.
4. Τερματίζεται η σύνδεση του host υπολογιστή με το SbRIO-9631.

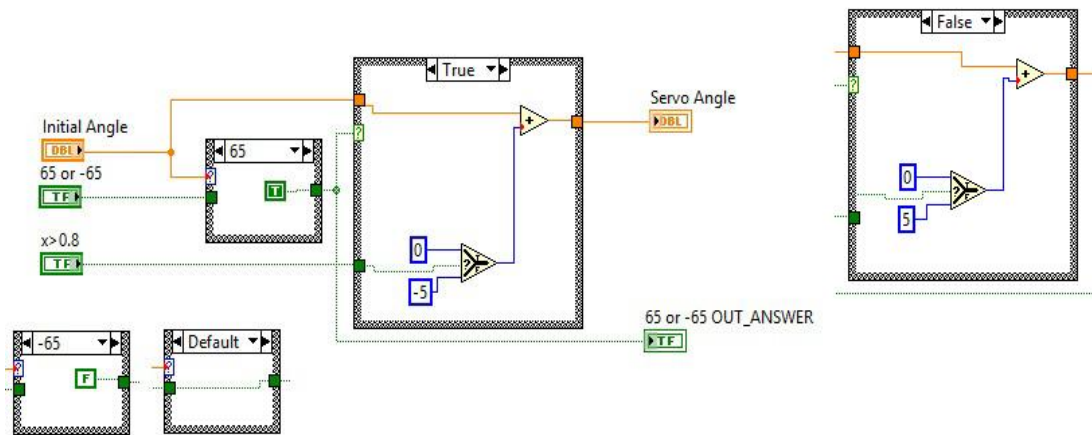
4.2 Παρουσίαση και ανάλυση των subVIs

4.2.1 Servo Rotate

Το VI αυτό είναι υπεύθυνο για την σωστή σάρωση του χώρου από τον αισθητήρα υπερήχων ο οποίος κινείται με την βοήθεια του servo motor. Ως εισόδους έχει την αρχική γωνία που βρίσκεται ο αισθητήρας υπερήχων που ισούται με 0 καθώς στην αρχή είναι στο μέσο του οχήματος και δύο boolean μεταβλητές που χρησιμοποιούνται για τον έλεγχο των δομών case. Η μία δείχνει αν το servo motor βρίσκεται στις -65ο με 65ο ενώ η άλλη αν το ρομπότ απέχει από εμπόδιο περισσότερο από 0.8m. Τέλος η έξοδος Servo Angle περιέχει την γωνία του servo motor σε μοίρες.



Εικόνα 4.3: Διάγραμμα εισόδων/εξόδων του Servo Rotate subVI

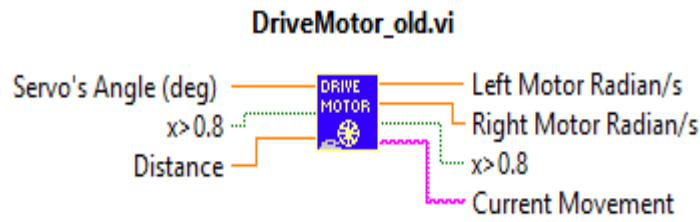


Εικόνα 4.4: Block διάγραμμα του Servo Rotate με όλες τις περιπτώσεις των Case Structures

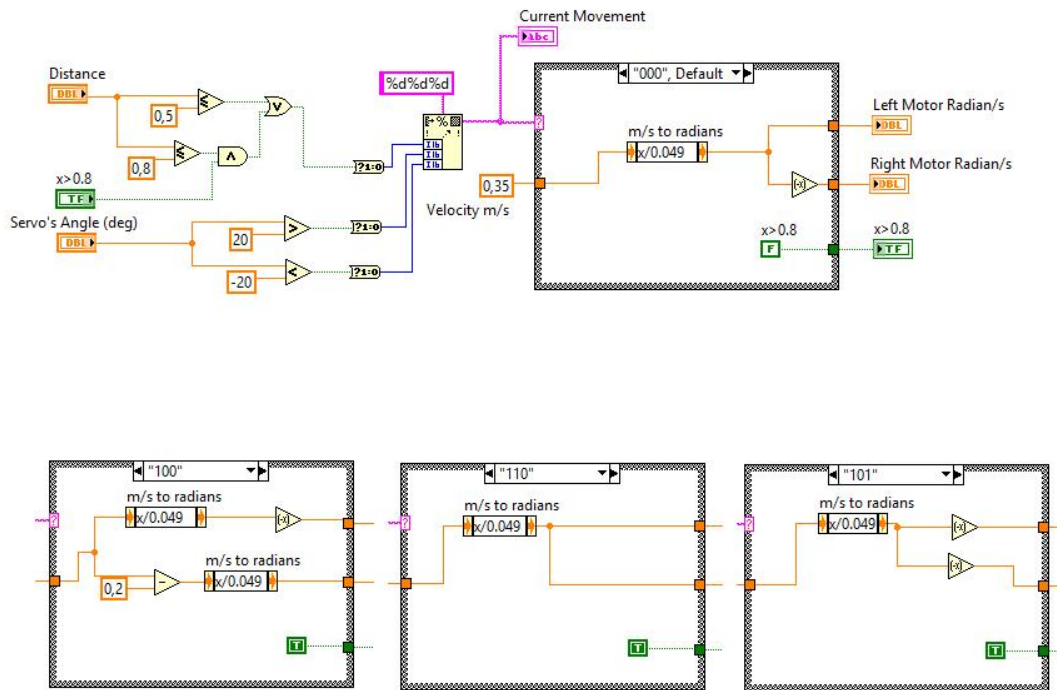
Το subVI αποτελείται από δύο δομές Case, με την πρώτη να είναι υπεύθυνη να περιορίζει το servomotor να κινείται από τις -65° ως 65° και την δεύτερη να αυξάνει την γωνία κίνησης ανά -5° αν δεν έχει φτάσει στις -65° ή 5° αν δεν έχει φτάσει στις 65° . Επιπλέον αν το ρομποτικό όχημα έχει εντοπίσει εμπόδιο σταματάει να μετατοπίζει τον αισθητήρα υπερήχων μέχρι αυτό να απέχει $0.8m$ από αυτό. Έτσι βεβαιώνεται ότι το όχημα θα καταφέρει να αποφύγει με επιτυχία το εμπόδιο καθώς επικεντρώνει την προσοχή του σε αυτό.

4.2.2 Drive Motor

Ίσως το πιο σημαντικό VI σε ολόκληρο το πρόγραμμα διότι επωμίζεται την ευθύνη για το πώς θα δράσει το ρομπότ αν εντοπίσει κάποιο εμπόδιο στην πορεία του. Δέχεται ως είσοδο την γωνία (σε rad) του αισθητήρα υπερήχων που υπολογίστηκε στο προηγούμενο VI, το αν απέχει το ρομποτικό όχημα παραπάνω από $0.8m$ από κάποιο εμπόδιο και την απόσταση που καταγράφει ο αισθητήρας υπερήχων. Σαν εξόδους παράγει δύο τις γωνιακές ταχύτητες για το αριστερό και δεξί μοτέρ, το αν το ρομποτικό όχημα απέχει πάνω από $0.8m$ και μια συμβολοσειρά (string) που ενημερώνει τον χρήστη στο front panel για την τρέχουσα πορεία του οχήματος.



Εικόνα 4.5: Διάγραμμα εισόδων/εξόδων του Drive Motor subVI



Εικόνα 4.6: Block διάγραμμα του Drive Motor με όλες τις περιπτώσεις του Case Structure

Αρχικά γίνεται μια σύγκριση της απόστασης που μετρήθηκε από τον αισθητήρα υπερήχων με δύο προκαθορισμένες αποστάσεις. Αν το εμπόδιο απέχει 0.5m ή και λιγότερο τότε το ρομπότ πρέπει να πάρει απόφαση για την αλλαγή της πορείας του. Το ποια πορεία θα επιλέξει εξαρτάται από την γωνία όπου βρίσκεται εκείνη την στιγμή ο αισθητήρας υπερήχων. Αφότου επιλέξει την τρόπο δράσης του τότε θα συνεχίσει να τον ακολουθεί μέχρι να απέχει από το εμπόδιο απόσταση πάνω από

0.8m. Στην συνέχεια αφότου ολοκληρωθούν οι συγκρίσεις, δημιουργείται μια συμβολοσειρά όπου γίνεται όρισμα στην επιλογή περίπτωσης στη δομή case. Η επεξήγηση της σημασίας κάθε συμβολοσειράς παρουσιάζεται στον παρακάτω πίνακα.

Εμπόδιο $\leq 0.5m$	Γωνία αισθητήρα $> 20^\circ$	Γωνία αισθητήρα $< -20^\circ$	Κατεύθυνση
0	0	0	Μπροστά
1	0	1	Αριστερά
1	1	0	Δεξιά
1	0	0	Πίσω και δεξιά

Πίνακας 4.1: Πίνακας σχηματισμού συμβολοσειράς

Αν ο αισθητήρας υπερήχων βρίσκεται από τις 20° ως τις 65° και εντοπίσει εμπόδιο αυτό σημαίνει πως το εμπόδιο βρίσκεται στην αριστερή πλευρά του ρομπότ. Έτσι το ρομποτικό όχημα πρέπει να κατευθυνθεί δεξιά και να απομακρυνθεί ως ότου απέχει παραπάνω από 0.8m. Αντίστοιχα αν εντοπιστεί εμπόδιο από -20° ως -65° τότε εμπόδιο βρίσκεται στην δεξιά πλευρά και το ρομπότ να κατευθυνθεί προς την αριστερή. Τέλος αν το ρομποτικό όχημα βρίσκεται στο διάστημα από -20° ως 20° τότε πρακτικά σημαίνει ότι το εμπόδιο βρίσκεται ευθεία μπροστά του. Έτσι για να το αποφύγει μετακινείται προς τα πίσω στρίβοντας ταυτόχρονα δεξιά αποφεύγοντας αποτελεσματικά το εμπόδιο.

Για τον υπολογισμό των γωνιακών ταχυτήτων για τα δύο μοτέρ ήταν απαραίτητη η μελέτη του εγχειριδίου χρήσεων του Write DC Motor Velocity Setpoints VI. Αναφέρει πως για κινηθεί το ρομποτικό όχημα ευθεία πρέπει οι γωνιακές ταχύτητες των μοτέρ να είναι ίσες κατά απόλυτη τιμή ενώ το πρόσημο της τιμής στο αριστερό μοτέρ να είναι θετικό ενώ στο δεξιό αρνητικό. Στην συνέχεια υπολογίστηκαν τα πρόσημα των μοτέρ ώστε το ρομποτικό όχημα να αλλάζει την κατεύθυνση του.

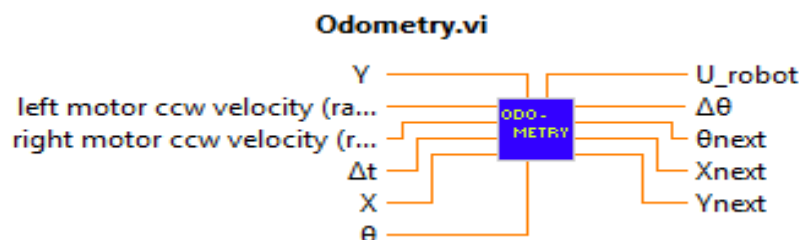
Πρόσημο Αριστερού Μοτέρ	Πρόσημο Δεξιού Μοτέρ	Κατεύθυνση
+	-	Μπροστά
+	+	Δεξιά
-	-	Αριστερά
-	+	Πίσω

Πίνακας 4.2: Πίνακας πρόσημων για την κατεύθυνση του ρομποτικού οχήματος

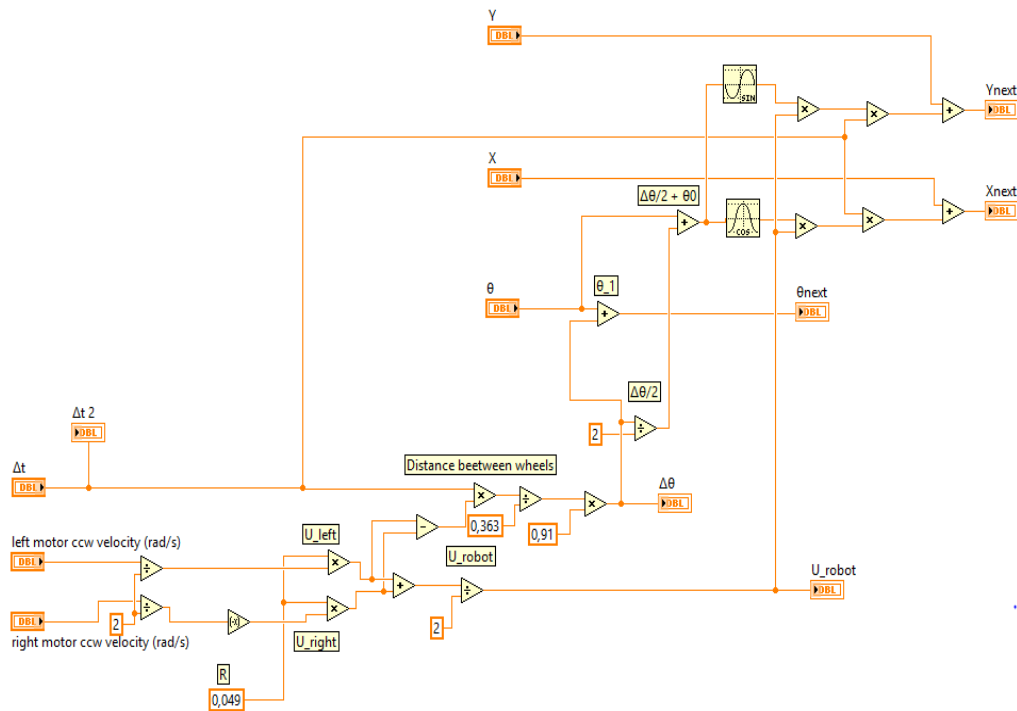
Έπειτα δίνοντας μια αρχική ταχύτητα της τάξεως στα 0.35m/s , στη δομή case μετατρέπεται σε γωνιακή ταχύτητα διαιρώντας την με την ακτίνα της ρόδας του ρομποτικού οχήματος ($R= 0.049m$) και προκύπτει η γωνιακή ταχύτητα των μοτέρ κατά απόλυτη τιμή. Τα πρόσημα των γωνιακών ταχυτήτων ορίζονται κατάλληλα στην αντίστοιχη περίπτωση του case.

4.2.3 Odometry

Το τρίτο subVI είναι υπεύθυνο για τον υπολογισμό της μετατόπισης του ρομποτικού οχήματος στο χώρο καθώς και για τον υπολογισμό της γωνίας θ . Στο block διάγραμμα του αναλύονται οι κινηματικές εξισώσεις που παρουσιάστηκαν στο κεφάλαιο 3. Το VI δέχεται ως εισόδους τις γωνιακές ταχύτητες των δύο μοτέρ, το χρονικό διάστημα Δt που υπολογίζεται σε κάθε εκτέλεση του βρόγχου καθώς και τις αρχικές τιμές της μετατόπισης κατά τους άξονες X,Y και την γωνία θ . Ενώ στις εξόδους του εμφανίζει την συνολική ταχύτητα του ρομποτικού οχήματος, τις επόμενες μετατοπίσεις κατά τους άξονες X,Y και την επόμενη γωνία θ .



Εικόνα 4.7: Διάγραμμα εισόδων/εξόδων του Odometry subVI



Εικόνα 4.8: Block διάγραμμα του Odometry subVI

4.3 Network Streams

Παρόλο που το SbRIO-9631 προσφέρει μεγάλες δυνατότητες στην επεξεργασία μετρήσεων σε πραγματικό χρόνο, πολλές φορές κρίνεται απαραίτητη η αποθήκευση των δεδομένων και η περαιτέρω επεξεργασία σε κάποιον υπολογιστή. Κατά την διάρκεια εκτέλεσης κάποιου VI ο υπολογιστής έχει πρόσβαση στα δεδομένα μέσω του Front Panel. Έτσι ο χρήστης έχει την δυνατότητα να παρατηρεί τις μεταβολές στις μετρήσεις μέσω γραφημάτων ή απλής απεικόνισης των τιμών. Μέρος του VI που τρέχει στο SbRIO-9631 τρέχει και στον host υπολογιστή για την απεικόνιση των δεδομένων στο Front Panel. Αν και ο υπολογιστής φαίνεται πως έχει πρόσβαση στα δεδομένα που εμφανίζονται, δεν είναι σε θέση να τα αποθηκεύσει ή να τα επεξεργαστεί. Η χρήση ροών δικτύου μπορεί να προσφέρει λύσεις μεταφοράς δεδομένων ανάμεσα σε VIs που εκτελούνται σε διαφορετικούς υπολογιστές στην ίδια εφαρμογή ή στο ίδιο υπολογιστή αλλά σε διαφορετικές εφαρμογές. Ακόμα η χρήση των ροών δεδομένων είναι η προτεινόμενη όταν η μεταφορά δεδομένων απαιτείται να γίνει:

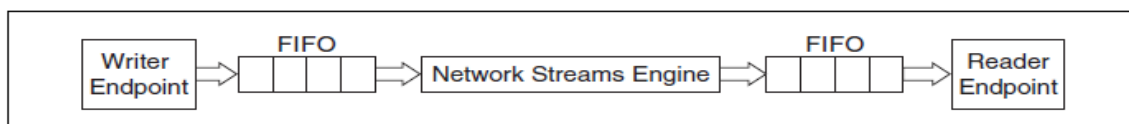
1. χωρίς απώλειες μεταξύ του SbRIO και του host υπολογιστή.

2. για καταγραφή δεδομένων στον host υπολογιστή.
3. για επεξεργασία και ανάλυση δεδομένων όπου απαιτείται περισσότερη μνήμη από αυτή που διαθέτει το SbRIO.

Υποστηρίζονται όλοι οι τύποι δεδομένων για μεταφορά άλλα οι παρακάτω τύποι δεδομένων μπορούν να αποστέλλονται με ταχύτερους ρυθμούς.

1. αριθμητικές μεταβλητές
2. μεταβλητές boolean
3. μονοδιάστατοι πίνακες αριθμητικών μεταβλητών
4. μονοδιάστατοι πίνακες μεταβλητών boolean

Κάθε τελικό σημείο για την αποστολή ή την λήψη των δεδομένων χρησιμοποιεί ένα FIFO buffer για τη μεταφορά δεδομένων. Ένα buffer είναι μια περιοχή μνήμης που αποθηκεύει δεδομένα. Το FIFO (First In First Out) αναφέρεται στον τρόπο όπου ελευθερώνεται ο buffer δηλαδή το πρώτο δεδομένο που εισέρχεται στην περιοχή μνήμης είναι και το πρώτο όπου αποστέλλεται στο τελικό σημείο λήψης. Τα δύο τελικά σημεία, ο αποστολέας και ο παραλήπτης μπορούν να λειτουργούν με διαφορετικούς ρυθμούς. Έτσι πρέπει να διασφαλιστεί πως το μέγεθος του buffer είναι ικανό ώστε να μην υπάρξει κάποια απώλεια δεδομένων. Στο παρακάτω σχήμα γίνεται μια γραφική αναπαράσταση στο πως μεταφέρονται τα δεδομένα.

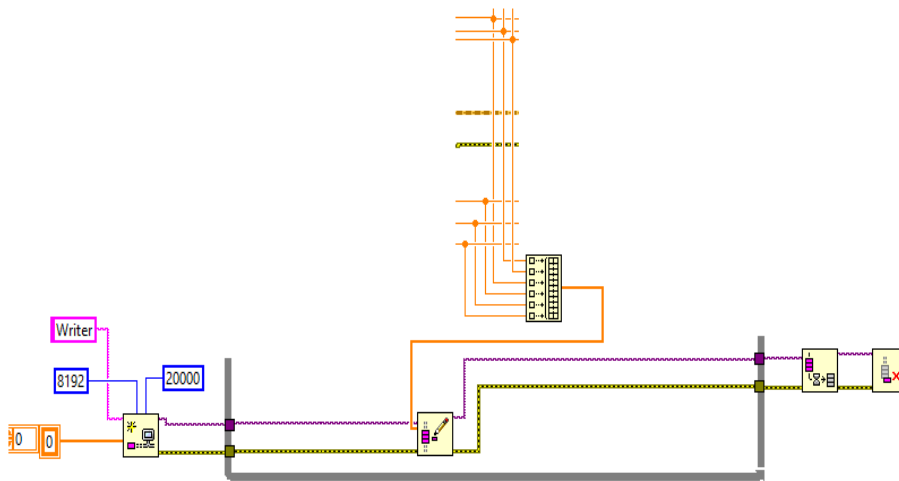


Εικόνα 4.9: Γραφική αναπαράσταση ροής δεδομένων

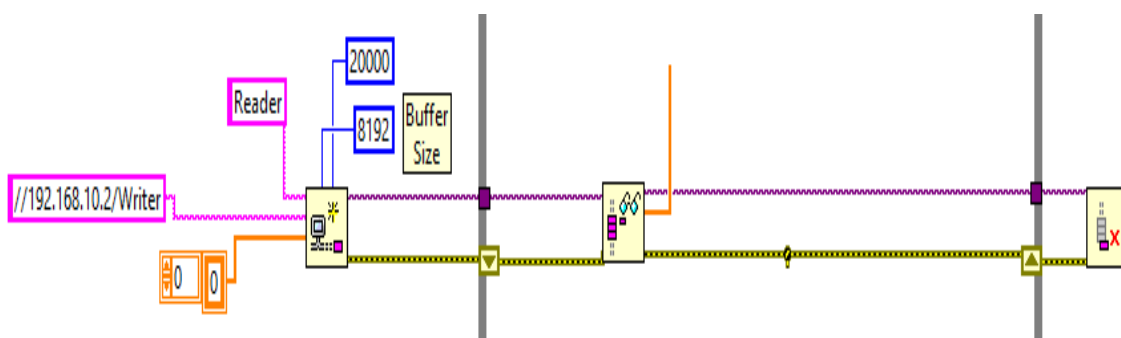
1. Το τελικό σημείο του αποστολέα γράφει δεδομένα σε ένα FIFO buffer, στο SbRIO-9631.
2. Το λογισμικό ροής δικτύου (που ονομάζεται Network Streams Engine ή NSE) μεταφέρει δεδομένα μέσω δικτύου στο άλλο FIFO buffer στο τελικό σημείο του παραλήπτη, στον host υπολογιστή σε αυτήν την περίπτωση.
3. Το τελικό σημείο του παραλήπτη διαβάζει τα δεδομένα από τον δικό του FIFO buffer.

Για την ολοκλήρωση μια ροής δεδομένων σε δύο εφαρμογές απαιτείται πρώτα η κατασκευή των δύο τελικών σημείων (endpoints) και η αρχικοποίηση της σύνδεσης των δύο αυτών σημείων. Έπειτα ξεκινάει η μεταφορά δεδομένων μεταξύ των δύο τελικών σημείων. Τέλος με την ολοκλήρωση της μεταφοράς δεδομένων, τα δύο τελικά σημεία καταστρέφονται και σταματάει η μεταξύ τους σύνδεση.

Από την στιγμή που απαιτείται η μεταφορά των μετρήσεων από το SbRIO-9631 στον host υπολογιστή τότε τελικό σημείο εγγραφής δεδομένων πρέπει να ενσωματωθεί στον κώδικα του ρομποτικού οχήματος ενώ το τελικό σημείο ανάγνωσης στον host υπολογιστή.

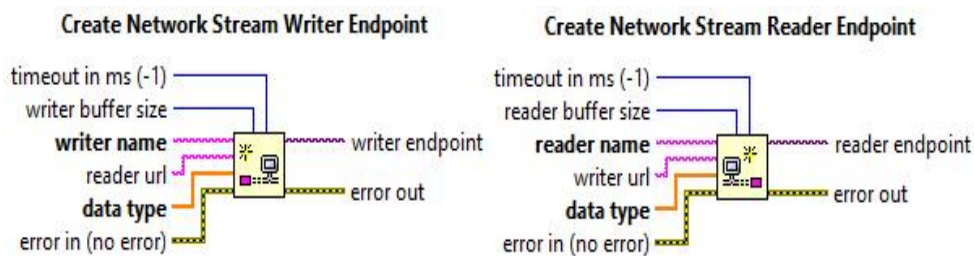


Εικόνα 4.10: Κώδικας εγγραφής ροής δεδομένων



Εικόνα 4.11: Κώδικας ανάγνωσης ροής δεδομένων

Για την υλοποίηση της ροής απαιτείται να εισαχθούν VI, όπου δημιουργούν τα endpoints του αποστολέα και του παραλήπτη. Τα VIs αυτά, ονομάζονται Create Network Stream Writer Endpoint και Create Network Stream Reader Endpoint. Αρχικά είναι απαραίτητο να ονομαστούν με μοναδικά ονόματα καθώς με αυτά θα αναγνωρίζονται από το LabVIEW. Στην συνέχεια το LabVIEW χρησιμοποιεί το όνομα που δόθηκε ώστε να δημιουργήσει ένα μοναδικό URL για το κάθε endpoint. Το είδος δεδομένων το οποίο θα διαχειρίζονται πρέπει να δηλωθεί και να είναι κοινό και για τα δύο endpoints. Το ίδιο συμβαίνει και για τις εισόδους 'reader/writer buffer size' και 'timeout in ms'. Καλό θα ήταν να δοθεί στα endpoints ένα ικανοποιητικό χρονικό διάστημα για να επιτευχθεί η ζεύξη λόγω ότι ο χρόνος δεν είναι πάντα ο ίδιος. Τέλος, σε ένα από τα δύο endpoints πρέπει να δηλωθεί το όνομα και η IP του άλλου ώστε να συνδεθούν τα δύο endpoints.



Εικόνα 4.12: Block διαγράμματα των Writer/Reader Endpoints

Για να εκτελεστεί ο κώδικας του ρομποτικού οχήματος θα πρέπει τα δύο endpoints να συνδεθούν μεταξύ τους στο χρονικά διάστημα του timeout. Αν δεν υπάρξει η ζεύξη, τότε εμφανίζεται ένα timeout σφάλμα που ενημερώνει τον χρήστη ότι ο κώδικας δεν θα εκτελεστεί.

Με την ολοκλήρωση των υπολογισμών, τα αποτελέσματα ομαδοποιούνται σε ένα πίνακα 2x6 και στο τέλος της ολοκλήρωσης κάθε επανάληψης του βρόγχου αποστέλλονται στο Read Stream VI που εκτελείται στο host υπολογιστή.

Στο πρόγραμμα που εκτελείται στον υπολογιστή (Εικόνα 4.1), γίνεται αρχικοποίηση δύο δισδιάστατων πινάκων 2x3. Με την άφιξη κάθε νέας γραμμής δεδομένων από το ρομποτικό όχημα, το VI είναι υπεύθυνο για την ανάγνωση των δεδομένων, την διάσπαση της γραμμής των έξι δεδομένων σε δύο γραμμές με τρία δεδομένα και την αποθήκευση αυτών των γραμμών σε δύο πίνακες με διαστάσεις 2x3. Στο τέλος κάθε εκτέλεσης του βρόγχου οι τελευταίες γραμμές των πινάκων εγγράφονται σε δύο αρχεία excel. Έτσι το πρώτο αρχείο περιέχει όλα τα δεδομένα της οδομετρίας που προέκυψαν από τις εντολές προς τα μοτέρ (Command Data) ενώ το δεύτερο τα δεδομένα από τους οπτικούς κωδικοποιητές (Encoder Data).

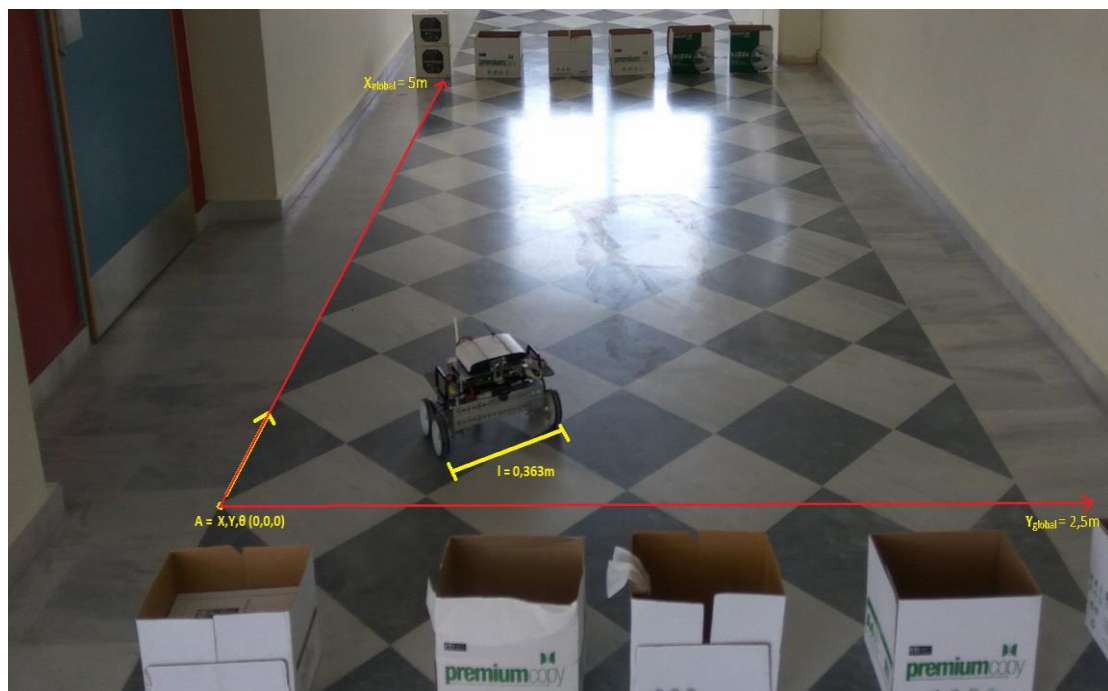
Κεφάλαιο 5^ο : Κίνηση του ρομπότ στο επίπεδο και υπολογισμός θέσης

5.1 Αρχικές συνθήκες και στόχοι του πειράματος

Αρχικά οριοθετήθηκε ένα χώρος με μήκος 5 μέτρων και πλάτος 2,5 μέτρων στον οποίο θα κινείται το ρομπότ. Ορίστηκε το σημείο εκκίνησης A του ρομπότ στο παγκόσμιο σύστημα συντεταγμένων ως $X_{global}=0$, $Y_{global}=0$, $\theta_{global}=0$. Έτσι το ρομπότ πριν την εκτέλεση οποιουδήποτε πειράματος πρέπει να τοποθετείται στο σημείο A ως εξής :

Το κέντρο του ρομπότ θα πρέπει να συμπίπτει με το σημείο A. Η απόσταση που απέχουν οι δύο αντικριστοί τροχοί του ρομπότ είναι $l=0,363m$. Οπότε το σημείο A θα πρέπει να απέχει περίπου $l/2$ από κάθε ρόδα.

Ο προσανατολισμός του ρομπότ αναφέρεται ως γωνία θ . Η γωνία αυτή θεωρείται μηδέν όταν ο άξονας X_R του συστήματος συντεταγμένων που προσαρμόζεται στο ρομπότ ταυτίζεται με τον άξονα X_{global} του παγκόσμιου συστήματος συντεταγμένων (βλέπε και κεφ. 3...).

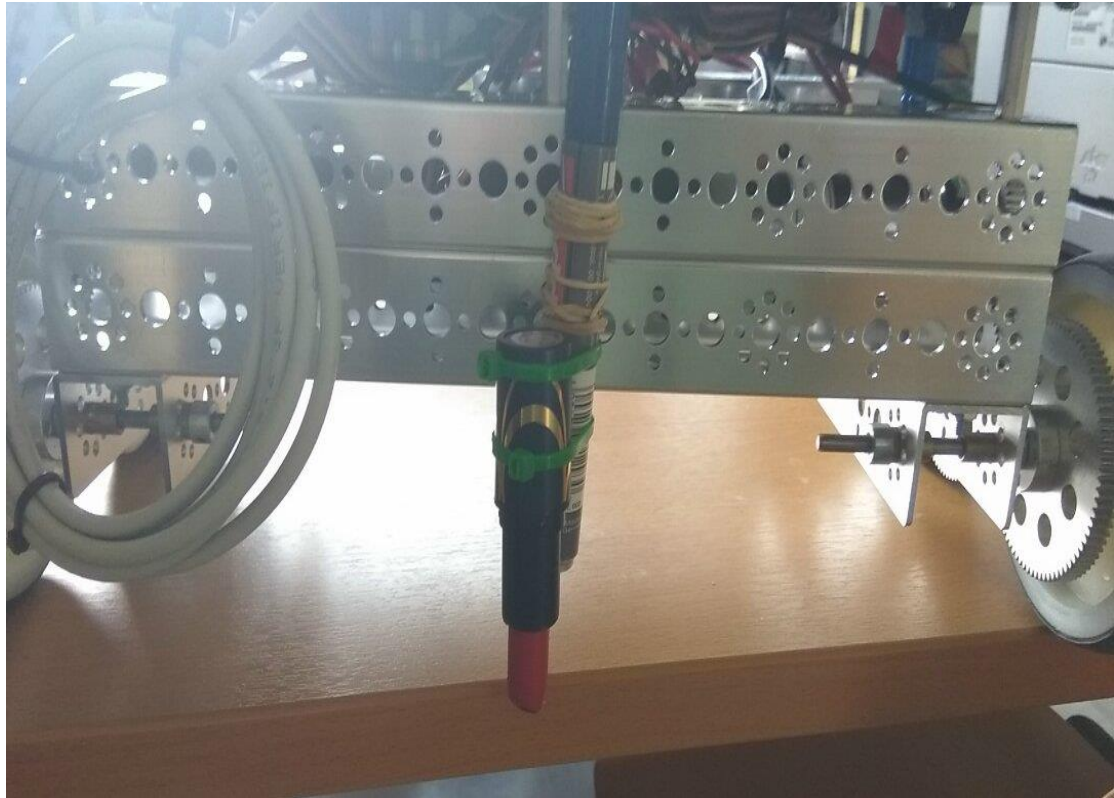


Εικόνα 5.1: Σημείο εκκίνησης A στο παγκόσμιο σύστημα συντεταγμένων

Για να θεωρηθεί μια διαδρομή κατάλληλη για μελέτη, κρίθηκε ότι το ρομπότ πρέπει να επιστρέψει, αφού εκτελέσει μια διαδρομή, όσο το δυνατόν πιο κοντά στο σημείο εκκίνησης A, δηλαδή να ολοκληρώσει τουλάχιστον ένα βρόχο κίνησης. Καθώς το ρομπότ κινείται, σημειώνει στο πάτωμα τη διαδρομή που ακολούθησε με την βοήθεια ενός κραγιόν, το οποίο τοποθετήθηκε στο μέσο του αμαξώματος. Τα πραγματικά σημεία της διαδρομής προσδιορίζονται στο παγκόσμιο σύστημα συντεταγμένων, με τη βοήθεια ενός μέτρου, εκτελώντας μετρήσεις στο πάτωμα πάνω στη γραμμή που χαράσσει το κραγιόν. Τα σημεία της διαδρομής αποθηκεύονται με τη μορφή αρχείου. Παράλληλα, καταγράφεται η κίνηση του ρομπότ, όπως προκύπτει από τις εντολές κίνησης που δίνονται στους τροχούς. Επίσης, καταγράφονται οι μετρήσεις που λαμβάνουν οι οπτικοί κωδικοποιητές, που είναι προσαρμοσμένοι στις ρόδες.

Έτσι, δίνεται η δυνατότητα να γίνει σύγκριση ανάμεσα στα σημεία της διαδρομής, όπως αυτά υπολογίζονται με τους εξής τρεις τρόπους: 1. από τις εντολές κίνησης που δίδονται στο ρομπότ (Command Data) 2. από τους οπτικούς κωδικοποιητές (Encoder Data) και 3. από την πραγματική διαδρομή που χαράσσει στο πάτωμα το κραγιόν (Actual Data).

Στο τέλος, μετά την ανάλυση των δεδομένων, θα μπορεί κανείς να παρατηρήσει πόσο ακριβείς τελικά είναι οι μετρήσεις που λαμβάνονται από τους κωδικοποιητές και αν μόνοι τους, δίχως άλλους αισθητήρες, είναι ικανοί να προσδιορίζουν με ακρίβεια την θέση του ρομπότ κάθε στιγμή.



Εικόνα 5.2: Καταγραφέας χάραξης διαδρομής

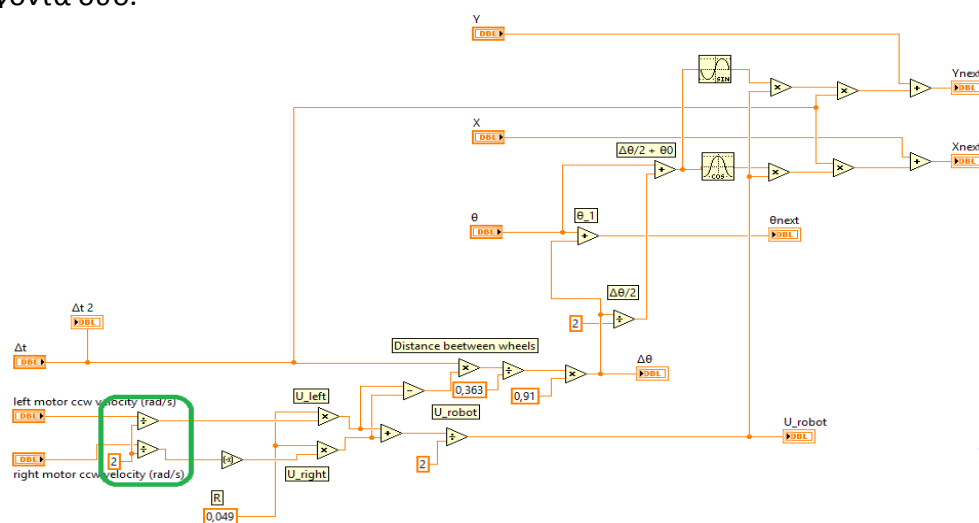
5.2 Έλεγχος ορθότητας των μετρήσεων

Πριν την έναρξη του πειράματος θεωρήθηκε σκόπιμο να γίνει ένας έλεγχος του υλικού και του λογισμικού, δηλαδή των μετρήσεων των κωδικοποιητών καθώς και του κώδικα που είναι υπεύθυνος για τις μετρήσεις. Αρχικά, έγινε η παρατήρηση ότι η γωνιακή ταχύτητα που διαβάζουν οι κωδικοποιητές ήταν διπλάσια από την γωνιακή ταχύτητα που δινόταν ως εντολή στα μοτέρ. Αυτό οφειλόταν στον τρόπο μετάδοσης της κίνησης από τα μοτέρ στις ρόδες του οχήματος. Η μετάδοση γίνεται με ένα σύστημα από γρανάζια διαφορετικού μεγέθους, όπως φαίνεται στην Εικόνα 5.3



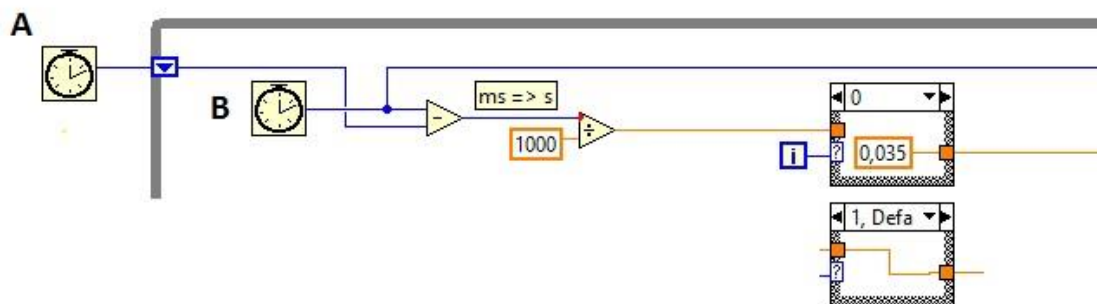
Εικόνα 5.3: Παρουσίαση του συστήματος μετάδοσης της κίνησης από τα μοτέρ στους τροχούς του οχήματος

Μετά από την αρίθμηση των εγκοπών και στα δύο γρανάζια ανακαλύφθηκε πως ο λόγος των εγκοπών ήταν δύο προς ένα. Για το λόγο αυτό, διαιρούμε τις γωνιακές ταχύτητες των δύο μοτέρ που καταγράφουν οι οπτικοί κωδικοποιητές με τον παράγοντα δύο.



Εικόνα 5.4: Διόρθωση γωνιακών ταχυτήτων στον κώδικα της οδομετρίας (Odometry subVI)

Παρατηρήθηκε επίσης, ότι κατά την εκτέλεση του προγράμματος το χρονικό διάστημα εκτέλεσης του πρώτου βρόγχου διαρκούσε πολύ περισσότερο σε σχέση με όλες τις υπόλοιπες επαναλήψεις. Αυτό οφειλόταν στο γεγονός ότι κατά την πρώτη εκτέλεση του βρόγχου, γινόταν μια καταγραφή του χρόνου από την συνάρτηση Tick Count. Έπειτα ο βρόγχος ανέμενε την σύνδεση των δύο Network Endpoints ώστε να είναι εφικτή η αποστολή των μετρήσεων στο host υπολογιστή. Με αποτέλεσμα να υπάρχει μια αρκετά μεγάλη καθυστέρηση στην λήψη της δεύτερης χρονικής στιγμής από την συνάρτηση Tick Count, όπου βρίσκεται μέσα στο βρόγχο του προγράμματος. Το πρόβλημα επιλύθηκε με την χρήση μια δομής Case. Κατά την πρώτη εκτέλεση του βρόγχου το διάστημα Δt δεν υπολογίζεται αλλά δίνεται ως όρισμα. Η τιμή που επιλέχθηκε είναι κατά προσέγγιση ο μέσος χρόνος που απαιτείται για την εκτέλεση ενός βρόγχου. Καθ' όλη την υπόλοιπη διάρκεια εκτέλεσης του προγράμματος το Δt υπολογίζεται κανονικά.



Εικόνα 5.5: Διόρθωση γωνιακών ταχυτήτων στον κώδικα της οδομετρίας (Odometry subVI)

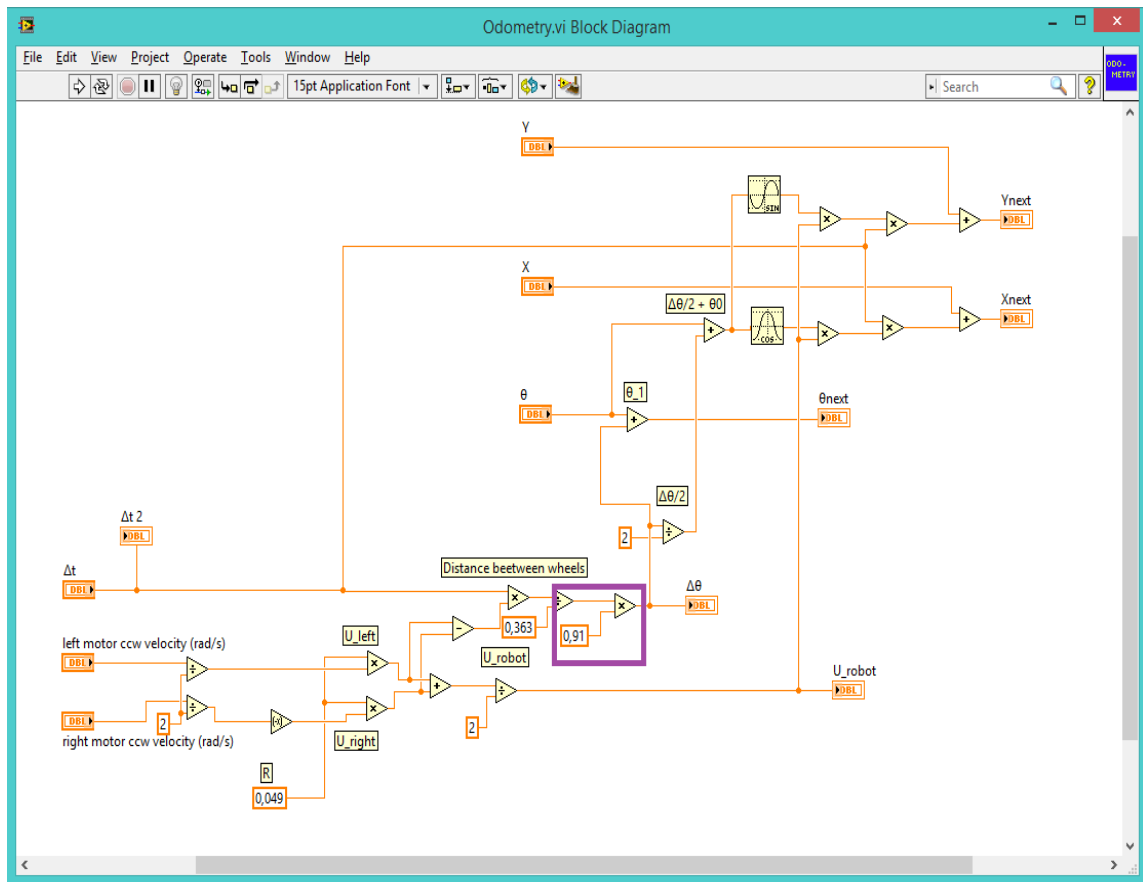
Στη συνέχεια, ελέγχθηκε το πόσο ακριβής είναι η μεταβολή της απόστασης κατά τον άξονα X_{global} , για να μετακινηθεί κατά τον άξονα αυτό χρειάστηκε και τα δύο μοτέρ του ρομπότ να έχουν την ίδια γωνιακή ταχύτητα. Στο ρομπότ δόθηκε η εντολή να μετακινηθεί για 2 μέτρα από το σημείο εκκίνησης A και μετά να σταματήσει. Μετά την ολοκλήρωση της διαδικασίας μετρήθηκε η πραγματική απόσταση που διένυσε το ρομπότ, η οποία ήταν 2.07 μέτρα. Η απόκλιση της θεωρητικής απόστασης που διένυσε με την πραγματική θεωρήθηκε ανεκτή καθώς θα πρέπει να ληφθεί υπόψη

πως οι κωδικοποιητές επηρεάζονται από την ολισθηρότητα του εδάφους που κινείται το ρομπότ, στην προκειμένη περίπτωση το έδαφος αποτελούταν από μάρμαρο. Επιπλέον είναι πολύ πιθανό ο κωδικοποιητής να μην είναι απόλυτα ακριβής στις μετρήσεις του. Κατά την διάρκεια του ελέγχου αυτού η θεωρητική απόσταση κατά τον άξονα Y_{global} ήταν 2 εκατοστά ενώ η πραγματική απόσταση ήταν στα 3.8 εκατοστά. Αυτό συνέβη καθώς κατά την εκκίνηση του ρομπότ το ένα μοτέρ ξεκινάει πιο γρήγορα και δίνει μια μικρή αρχική μετατόπιση κατά τον άξονα Y_{global} .

Ο δεύτερος έλεγχος που πραγματοποιήθηκε ήταν για την ορθή μέτρηση της γωνίας θ . Καθώς το ρομπότ δεν έχει κάποιο γυροσκόπιο ή πυξίδα για την μέτρηση και σύγκριση των αποτελεσμάτων, ο έλεγχος έγινε με το εξής πείραμα. Δόθηκε η εντολή στο ρομπότ να κάνει ένα πλήρη δεξιόστροφο κύκλο. Με την ολοκλήρωση του κύκλου η γωνία θ θα έπρεπε να έχει πάρει την τιμή γύρω στα 6.28 rad, όπου θα σήμαινε και την επιστροφή του ρομπότ στην αρχική θέση A. Για να γίνουν πιο ακριβείς οι έλεγχοι, το ρομπότ αρχικά έκανε μια κλειστή δεξιά στροφή και στην συνέχεια μια πολύ πιο ανοικτή ώστε να μελετηθεί η διαφορά υπολογισμού της γωνίας θ , αν υπήρχε. Έτσι με την ολοκλήρωση και των δύο κύκλων και την επιστροφή του ρομπότ στο σημείο εκκίνησης A, στην περίπτωση του κλειστού δεξιόστροφου κύκλου παρατηρήθηκε η γωνία $\theta = -6.873$ rad ενώ στο πιο ανοικτού κύκλου $\theta = -6.914$ rad. Η διαδικασία αυτή εκτελέστηκε τρεις φορές και οι μετρήσεις που πάρθηκαν ήταν πάντα παραπλήσιες με διαφορά από το δεύτερο δεκαδικό ψηφίο και μετά. Έτσι κρίθηκε σκόπιμο να εισαχθεί στον κώδικα υπολογισμού της γωνίας θ ένας διορθωτικός παράγοντας ώστε η γωνία θ να παίρνει την σωστή τιμή όταν ολοκληρώνει τον κύκλο κάθε φορά. Ο διορθωτικός παράγοντας, μετά από την διαίρεση του 6.28 με την εκάστοτε γωνία θ που πρόκυπτε προσεγγίστηκε στην τιμή 0.91 ώστε να διορθώνει και τα δύο είδη στροφών. Μετά πραγματοποιήθηκε μια ακόμα φορά η ίδια διαδικασία και παρατηρήθηκε πως η γωνία θ για την κλειστή στροφή ισούταν με -6.229 rad, ενώ για την πιο ανοικτή στροφή -6.297 rad.

Η τόσο μεγάλη απόκλιση της πραγματικής γωνίας θ από αυτή που υπολογίζεται με την βοήθεια των κωδικοποιητών, ίσως οφείλεται στις κινηματικές εξισώσεις που χρησιμοποιούμε και το είδος του ρομποτικού οχήματος. Οι κινηματικές εξισώσεις που χρησιμοποιήθηκαν, είχαν ως πρότυπο ένα όχημα το οποίο είχε ένα ζευγάρι

αντικριστούς τροχούς και μια ρόδα στο πίσω μέρος για ισορροπία ενώ το ρομποτικό όχημα της πτυχιακής εργασίας αποτελείται από δύο ζεύγη αντικριστών τροχών.



Εικόνα 5.6: Εισαγωγή διορθωτικού παράγοντα στον κώδικα της οδομετρίας

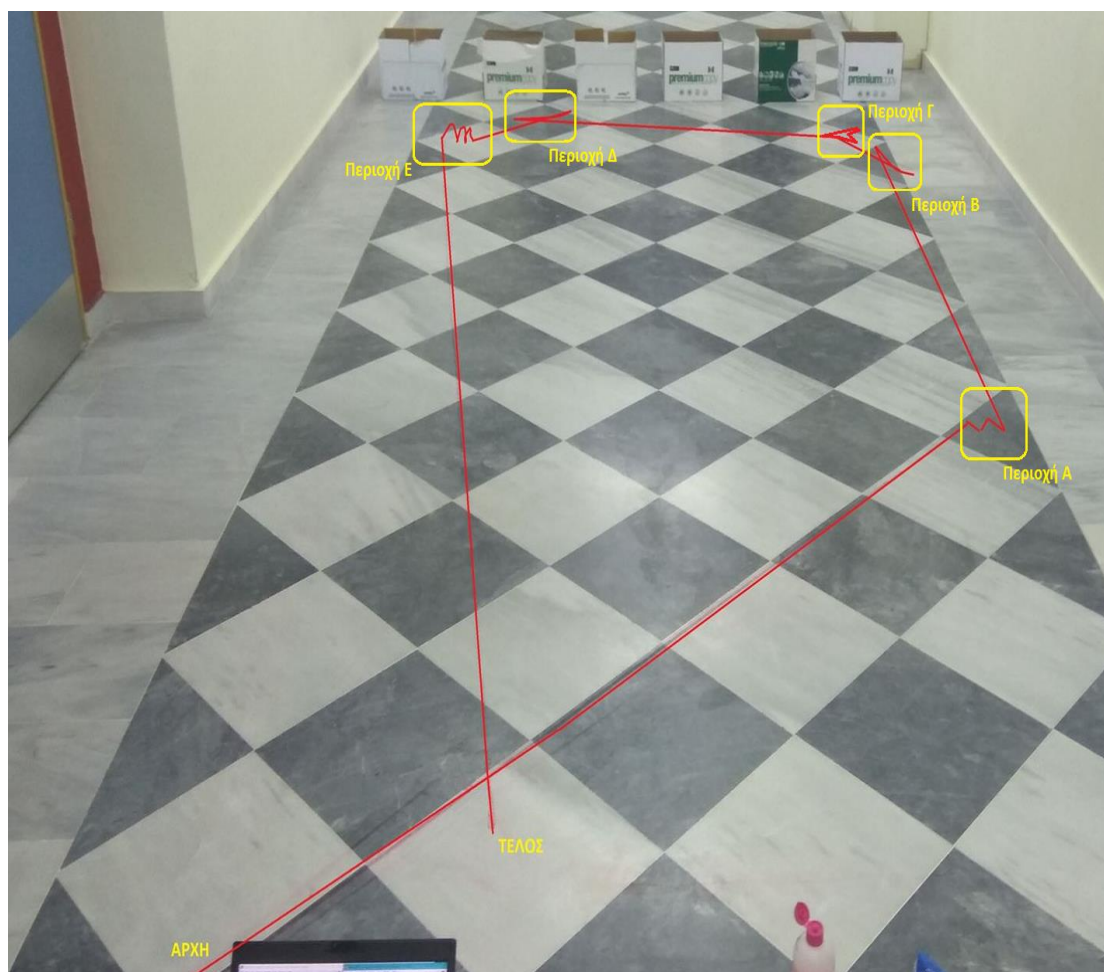
Το αρνητικό πρόσημο στην γωνία θ υποδηλώνει ότι ο κύκλος που διέγραψε ήταν δεξιόστροφος, καθώς όταν η ίδια διαδικασία πραγματοποιήθηκε με αριστερόστροφους κύκλους η γωνία θ είχε θετικές τιμές. Το πρόσημο της γωνίας θ καθορίζεται από τον υπολογισμό του $\Delta\theta$, που δίνεται από την σχέση:

$$\Delta\theta = \frac{R}{l} (\dot{\phi}_l - \dot{\phi}_r) * \Delta t \quad (\text{Βλέπε εξίσωση 3.4})$$

Όπου αν αντιστραφεί η διαφορά των γωνιακών ταχυτήτων των μοτέρ αλλάζει και το πρόσημο του $\Delta\theta$ και κατ'έπекταση και το πρόσημο της γωνίας θ .

5.3 Εκτέλεση του πειράματος

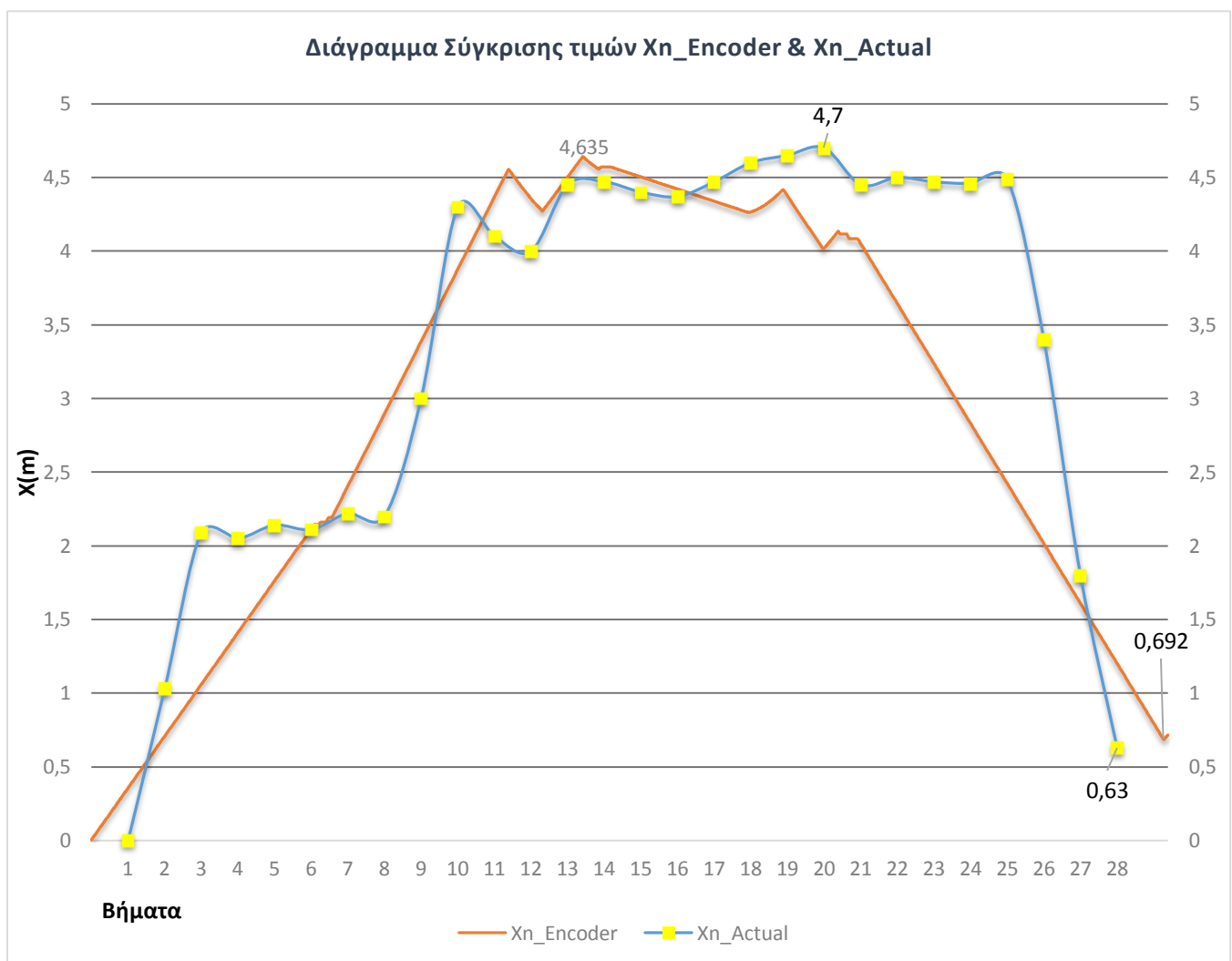
Για την πραγματοποίηση του πειράματος, το ρομπότ τοποθετήθηκε στο σημείο εκκίνησης A με αρχική γωνία $\theta = 45^\circ$ ώστε να διανύσει σε ένα κύκλο όσο μεγαλύτερη απόσταση γίνεται. Με την προσθήκη αρχικής γωνίας θ , είναι απαραίτητο να ενημερωθεί και το πρόγραμμα που εκτελείται, για τη νέα αρχική τιμή. Στην συνέχεια το ρομπότ είναι έτοιμο να εκτελέσει την διαδρομή του. Έτσι ξεκινάει η εκτέλεση του προγράμματος πλοήγησης του ρομπότ στο επίπεδο. Από την μεριά του host υπολογιστή εκτελείται το πρόγραμμα που είναι υπεύθυνο για τη συλλογή και προβολή των δεδομένων που καταγράφονται και αποστέλλονται από το ρομπότ προς τον υπολογιστή για ανάλυση. Με την άφιξη του ρομπότ κοντά στο σημείο εκκίνησης A ολοκληρώνεται το πείραμα. Συνολικά καταγράφηκαν 28 σημεία πάνω στην πραγματική διαδρομή καθ' όλη τη διάρκεια της κίνησης του ρομπότ και 5 περιοχές που συγκεντρώνουν το μεγαλύτερο ενδιαφέρον για ανάλυση.



Εικόνα 5.7: Η διαδρομή όπου ακολούθησε το ρομπότ

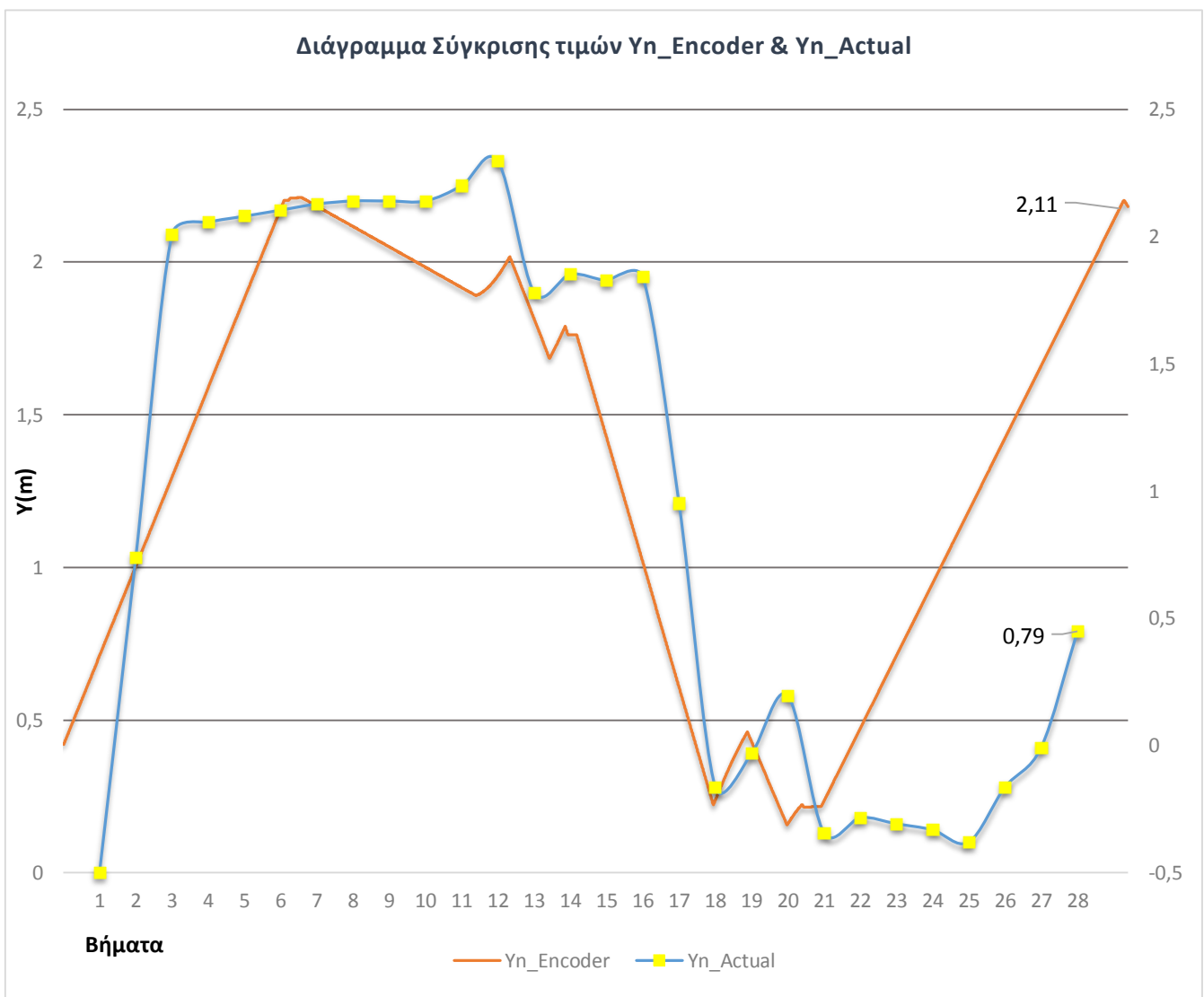
5.4 Ανάλυση και σχολιασμός αποτελεσμάτων

Στα γραφήματα που ακολουθούν με πράσινο χρώμα απεικονίζονται οι τιμές που δίνονται από τον κώδικα ως εντολές για την κίνηση του ρομπότ (Command Data), με κόκκινο χρώμα οι τιμές που υπολογίζονται από τα δεδομένα των οπτικών κωδικοποιητών (Encoder Data) και με μπλε χρώμα οι τιμές που καταγράφηκαν με το κραγιόν στο επίπεδο (Actual Data).



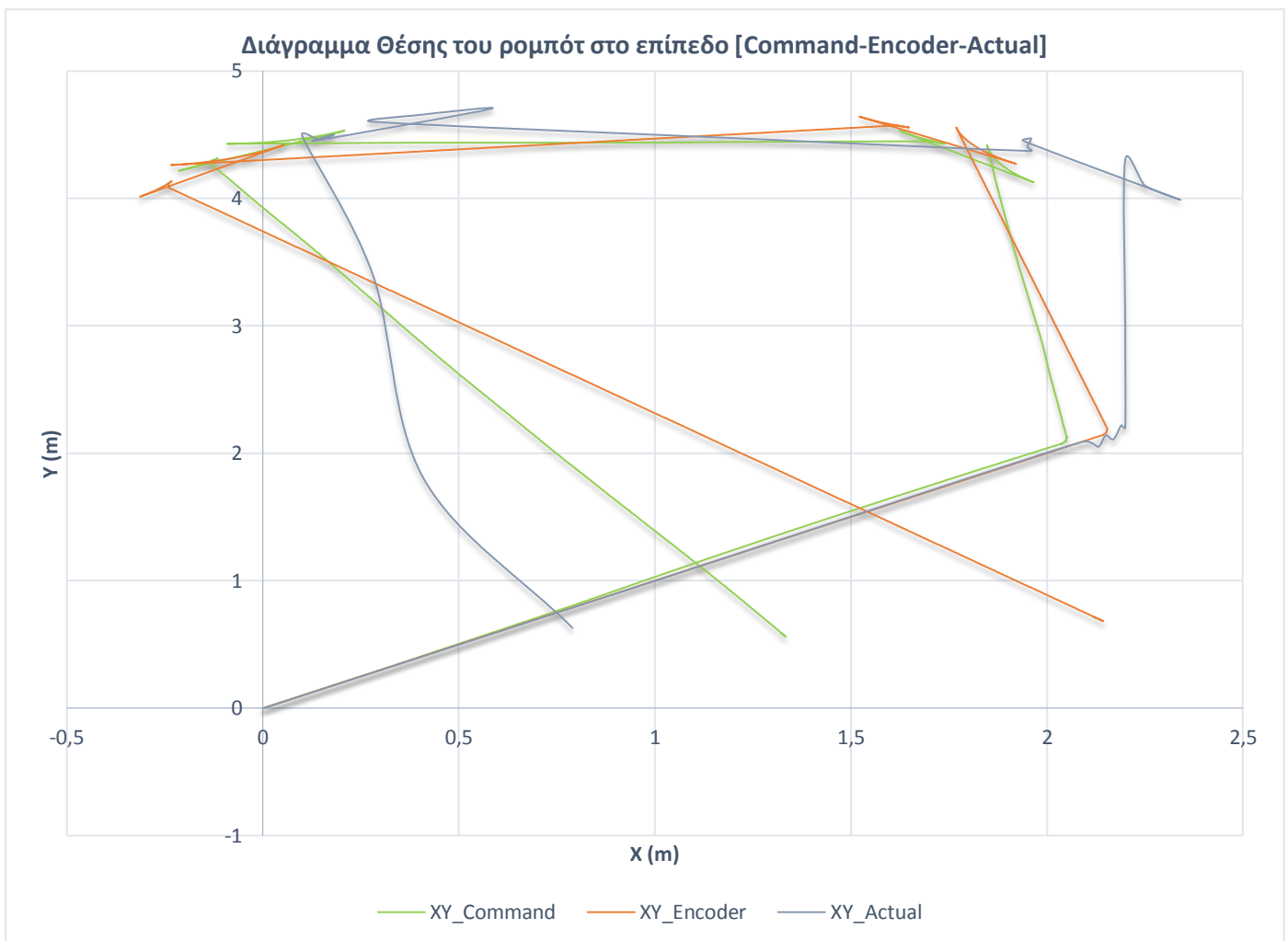
Γράφημα 5.1: Διάγραμμα σύγκρισης τιμών Xn_Encoder (κόκκινη γραμμή) και Xn_Actual (μπλε γραμμή)

Στο 5.1 γράφημα παρουσιάζεται η μετατόπιση κατά τον άξονα Χ στο παγκόσμιο σύστημα συντεταγμένων, όπως υπολογίστηκε από τους κωδικοποιητές και τα σημεία όπου καταγράφηκαν με το χέρι σε συνάρτηση με το χρόνο καταγραφής (βήμα του ρομπότ-σύνολο περίπου 30 βήματα). Παρατηρούμε ότι η τελική τιμή του $X_{n_Encoder}$ ισούται με 0.69m ενώ η τελική τιμή του X_{n_Actual} με 0.63m. Έτσι, αν και υπάρχουν ενδιάμεσες αποκλίσεις παρατηρείται μια αρκετά μεγάλη ακρίβεια των δύο μετρήσεων στην τελική θέση. Η μέγιστη τιμή για το X_{n_Actual} είναι 4.7m ενώ για το $X_{n_Encoder}$ είναι 4.635m.



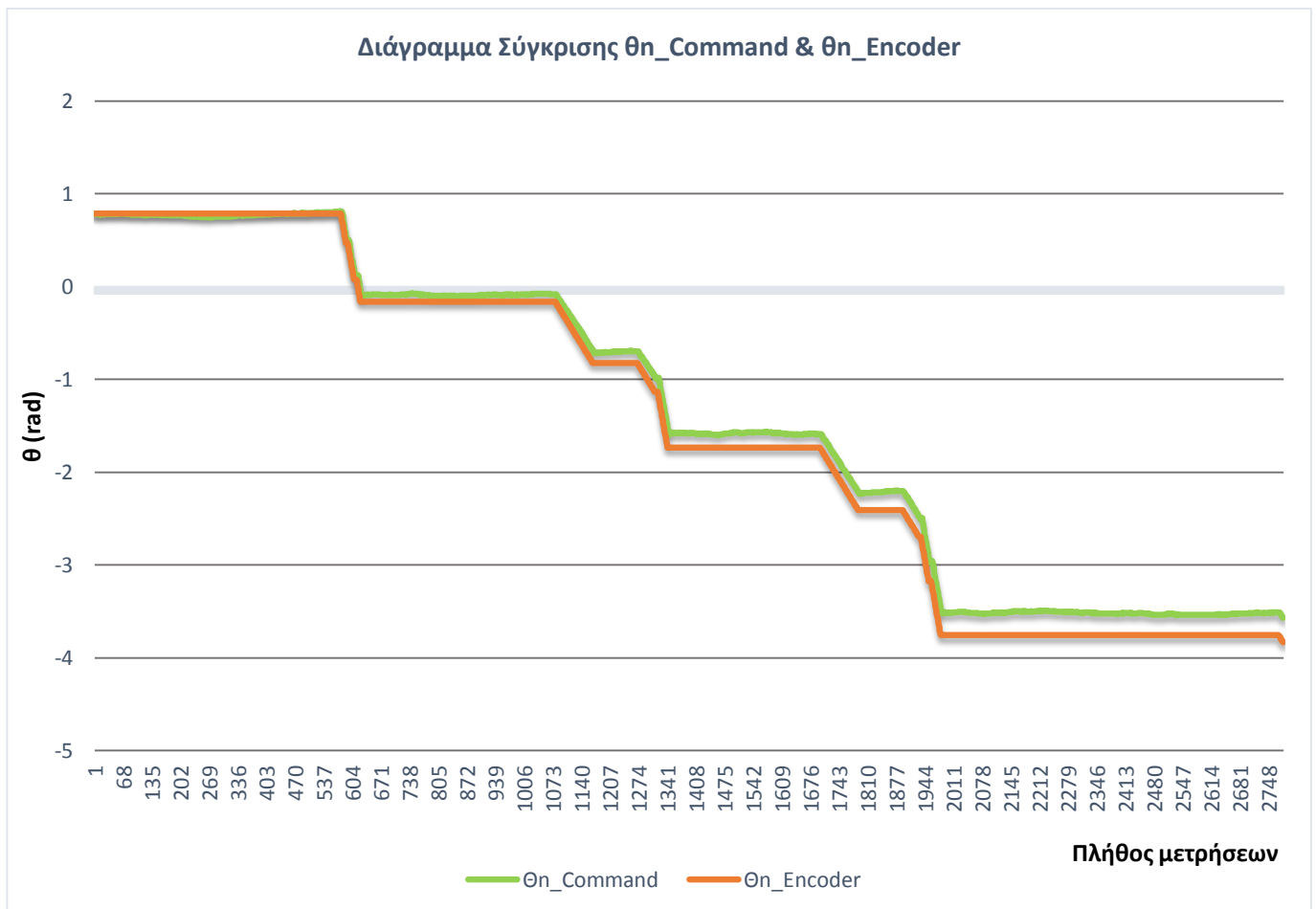
Γράφημα 5.2: Διάγραμμα σύγκρισης τιμών $Y_n_Encoder$ (κόκκινη γραμμή) και Y_n_Actual (μπλε γραμμή)

Αντίστοιχα στο γράφημα 5.2 γίνεται η σύγκριση των μετατοπίσεων κατά τον άξονα Y. Παρατηρείται ότι όσο το ρομπότ δεν κάνει πολλές επιτόπιες κινήσεις για την αποφυγή κάποιου εμποδίου η μεταβολή των ευθειών γίνεται με παραπλήσιο ρυθμό. Στην περιοχή A όπου μετρήθηκαν πολύ μικρές μεταβολές στην μετατόπιση κατά τον άξονα Y οι ίδιες μεταβολές δεν αποτυπώνονται και στην ευθεία των σημείων όπου υπολογίστηκαν με την βοήθεια των κωδικοποιητών. καθώς η μεταβολή στην γωνία θ είναι τόσο απότομη που επηρεάζει τον υπολογισμό της μετατόπισης κατά τον άξονα αυτό. Η μέγιστη για το Y_n_Actual είναι 0.79m ενώ για το $Y_n_Encoder$ είναι 2.11m.



Γράφημα 5.3: Διάγραμμα μετατόπισης στο χώρο με χρήση δεδομένων *Command Data* (πράσινο χρώμα), *Actual Data* (μπλε γραμμή) και *Encoder Data* (κόκκινη γραμμή)

Το διάγραμμα 5.3 παρουσιάζει την μετατόπιση του ρομπότ στο χώρο όπως υπολογίστηκε από τις εντολές που δόθηκαν στα μοτέρ (με πράσινο χρώμα), από τους κωδικοποιητές (με κόκκινο χρώμα) και όπως καταγράφηκε από τις μετρήσεις όπου πάρθηκαν πάνω στην πραγματική διαδρομή (με μπλε χρώμα). Παρατηρείται μια αδυναμία των οπτικών κωδικοποιητών να καταγράψουν της απότομες μεταβολές της γωνίας θ . Έτσι καθ' όλη την διάρκεια της διαδρομής συσσωρεύονται λανθασμένες μετρήσεις και προκύπτει η μεγάλη απόκλιση στην τελική μετατόπιση του ρομποτικού οχήματος σε σχέση με την μετατόπιση που υπολογίστηκε με τα δεδομένα XY_Actual . Το ίδιο ισχύει και με τα δεδομένα από τις εντολές των μοτέρ αλλά το σφάλμα πιθανό να είναι μικρότερο επειδή η γωνιακή ταχύτητα είναι σταθερή και δεν προκύπτει από τις μετρήσεις των οπτικών κωδικοποιητών.



Γράφημα 5.4: Διάγραμμα σύγκρισης γωνίας θ υπολογισμένη από *Command Data* (πράσινη γραμμή) και *Encoder Data* (κόκκινη γραμμή)

Στο τελευταίο διάγραμμα γίνεται σύγκριση των γωνιών θ που υπολογίζονται από τις γωνιακές ταχύτητες, που δίνονται ως εντολές στα μοτέρ ώστε να κινήσουν το ρομπότ (με πράσινο χρώμα) και τις γωνιακές ταχύτητες που διαβάζουν οι κωδικοποιητές κατά την κίνηση του (με κόκκινο χρώμα). Οι γωνιακές ταχύτητες που δίνονται ως εντολές μεταβάλλονται μόνο αν το ρομπότ αποφασίσει ότι πρέπει να αλλάξει την κατεύθυνσή του ώστε να αποφύγει κάποιο εμπόδιο. Η απόκλιση στις δύο ευθείες αυξάνεται κάθε φορά όταν προκύπτουν απότομες μεταβολές στην μετατόπιση του οχήματος.

5.5 Συμπεράσματα Πειράματος

Με την ολοκλήρωση της ανάλυσης του πειράματος εξήχθησαν κάποια χρήσιμα και συνάμα ενδιαφέροντα συμπεράσματα.

1. Τα μοτέρ που χρησιμοποιούνται για την κίνηση του ρομποτικού οχήματος μπορεί να παρέχονται από τον ίδιο κατασκευαστή και να είναι ίδιου τύπου όμως αυτό δεν σημαίνει πως είναι κατασκευασμένα με πανομοιότυπο τρόπο. Έτσι παρατηρούνται μικρές αποκλίσεις στην απόδοση και απόκριση τους.
2. Οι κωδικοποιητές είναι επιρρεπείς σε λανθασμένες μετρήσεις αν το έδαφος στο οποίο κινείται το ρομποτικό όχημα δεν προσφέρει επαρκή πρόσφυση. Είναι πιθανό ότι μετρήσεις από τους κωδικοποιητές πρέπει να λαμβάνονται σε πιο πυκνά χρονικά διαστήματα.
3. Το ρομποτικό όχημα δεν εξοπλίζεται με κάποιου είδους αισθητήρα στροφής (πυξίδα) για τον ακριβή υπολογισμό της γωνίας θ , ώστε να μπορεί να υπάρξει μια σύγκριση με τη γωνία θ που υπολογίζεται με την βοήθεια των κωδικοποιητών.
4. Είναι απαραίτητο να οριστεί επακριβώς ο χρόνος εκτέλεσης του κύριου βρόγχου προγράμματος ώστε αυτός να μην επηρεάζει τυχόν μετρήσεις που εξαρτώνται από αυτόν.
5. Θα πρέπει να πραγματοποιηθούν επιπλέον πειράματα με διαφορετικές διαδρομές και αρχικές συνθήκες ώστε να συγκριθούν με το ήδη υπάρχον

πείραμα. Έτσι θα μπορεί κανείς να έχει μια καλύτερη εικόνα για το ποια είναι τα αίτια της απόκλισης των τιμών.

5.6 Προτεινόμενες Λύσεις

1. Για την διαφορά της απόκρισης των δύο μοτέρ, που έχει ως αποτέλεσμα τη μερική μετατόπιση του ρομπότ, σε ευθείες διαδρομές, θα μπορούσε να προστεθεί κώδικας διόρθωσης σφάλματος στο μοτέρ που προκαλεί αυτή τη μεταβολή. Για να υλοποιηθεί αυτό θα πρέπει να βρεθεί η διαφορά των δύο γωνιακών ταχυτήτων και να υπολογιστεί ο παράγοντας διόρθωσης.
2. Θα μπορούσε να βελτιωθεί ο κώδικας αποφυγής εμποδίων ώστε το ρομποτικό όχημα να τα αποφεύγει με πιο ομαλές μανούβρες. Έτσι θα μειωνόταν η πιθανότητα να μην υπολογίζεται σωστά η γωνία θ .
3. Είναι απαραίτητο να προστεθεί κάποιος αισθητήρας ή άλλη τεχνική υπολογισμού στροφής για τον ακριβή υπολογισμό της γωνίας θ . Έτσι θα δοθεί η δυνατότητα να συγκριθούν οι υπολογιζόμενες γωνίες θ με αυτές που θα καταγράφει ο αισθητήρας.

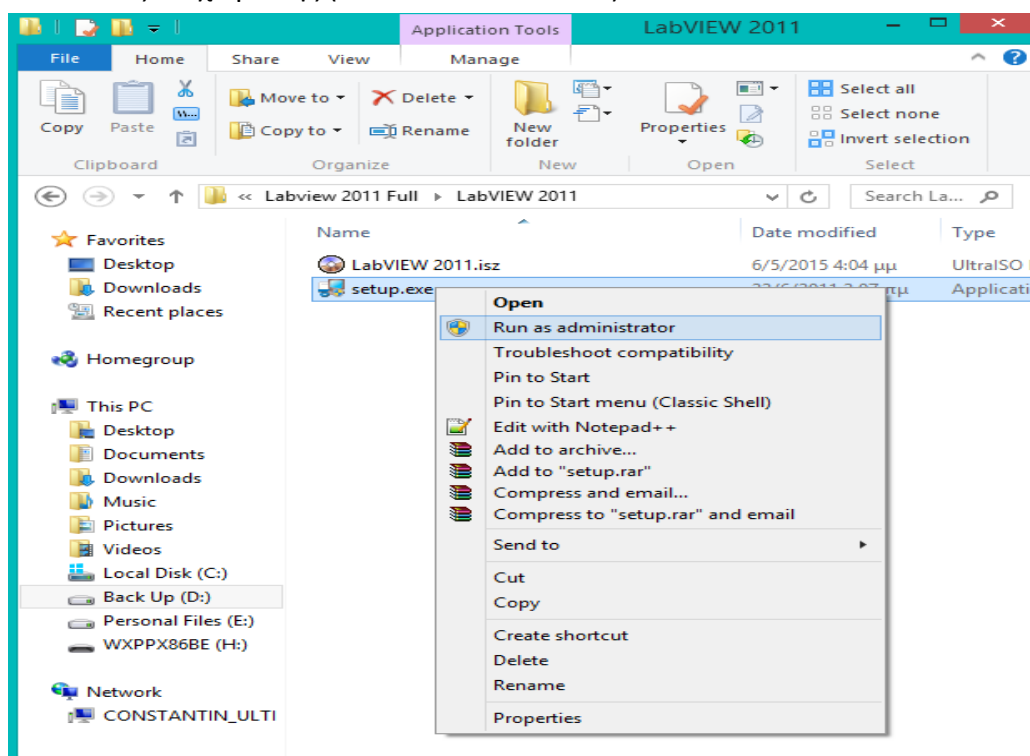
Παράρτημα 1^ο: Εγκατάσταση απαραίτητου λογισμικού

Για την περάτωσης της πτυχιακής εργασίας , από μεριά λογισμικού, απαιτήθηκε να εγκατασταθεί στον υπολογιστή τα εξής προγράμματα:

1. LabVIEW 2011 (32-Bit)
2. Robotics Module
3. FPGA Module
4. Real Time Module
5. Internet Utilities
6. LabVIEW Hardware Support

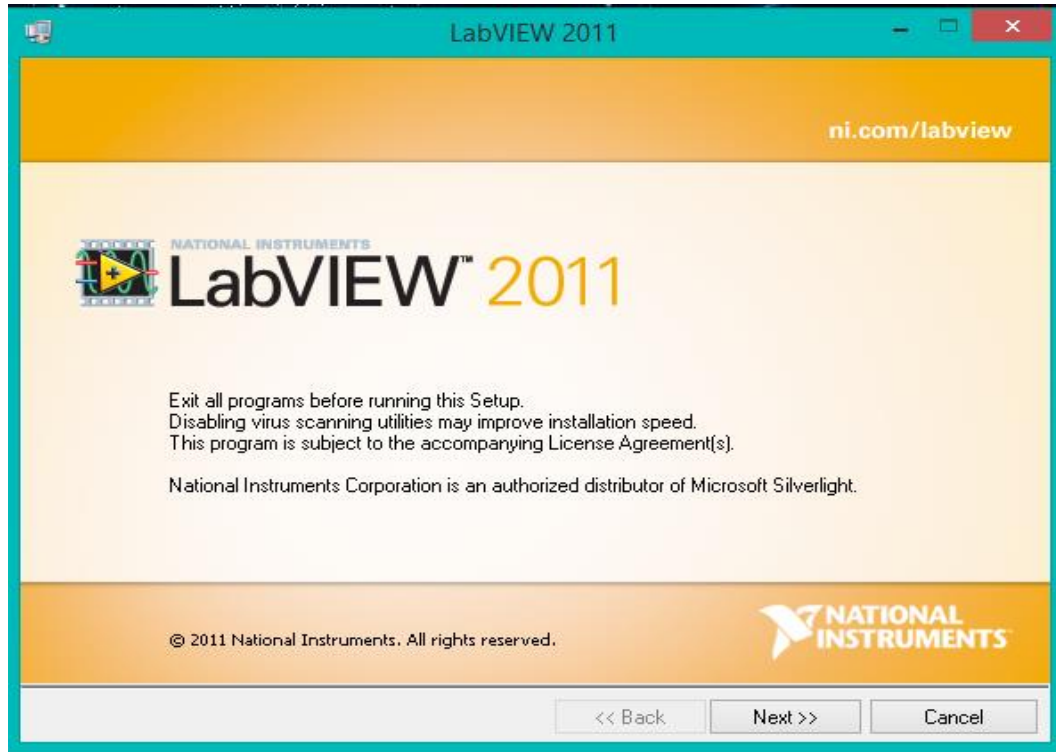
Στην συνέχεια ακολουθούν οι οδηγίες εγκατάστασης του LabVIEW. Με την ίδια διαδικασία εγκαθιστούνται και τα υπόλοιπα τμήματα (modules) του LabVIEW που είναι απαραίτητα για την απρόσκοπτη λειτουργία του προγράμματος που συγγράφηκε.

Αρχικά σε κάθε αρχείο εκκίνησης εγκατάστασης (setup.exe) θα πρέπει να εκτελείται ως διαχειριστής (Run as Administrator).



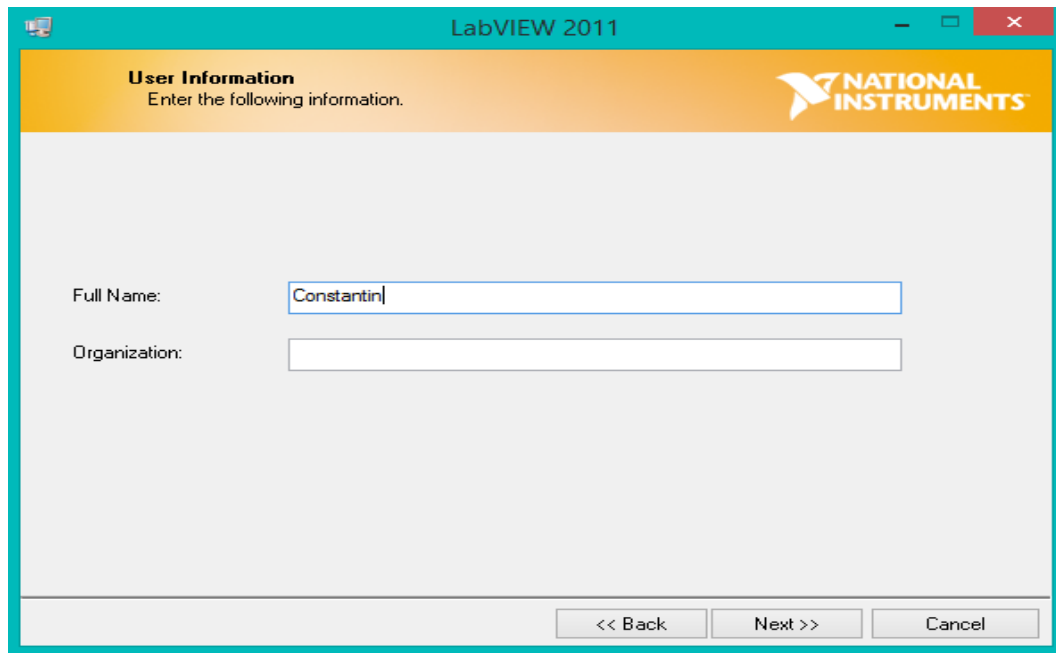
Εικόνα 1: Εκτέλεση αρχείου ως διαχειριστής

Έπειτα εμφανίζεται η αρχική εικόνα εγκατάστασης όπου μας αναφέρει τυπικές οδηγίες για την σωστή εγκατάσταση του λογισμικού όπως την απενεργοποίηση του αντικού προγράμματος (antivirus).



Εικόνα 2: Αρχική εικόνα εγκατάστασης LabVIEW

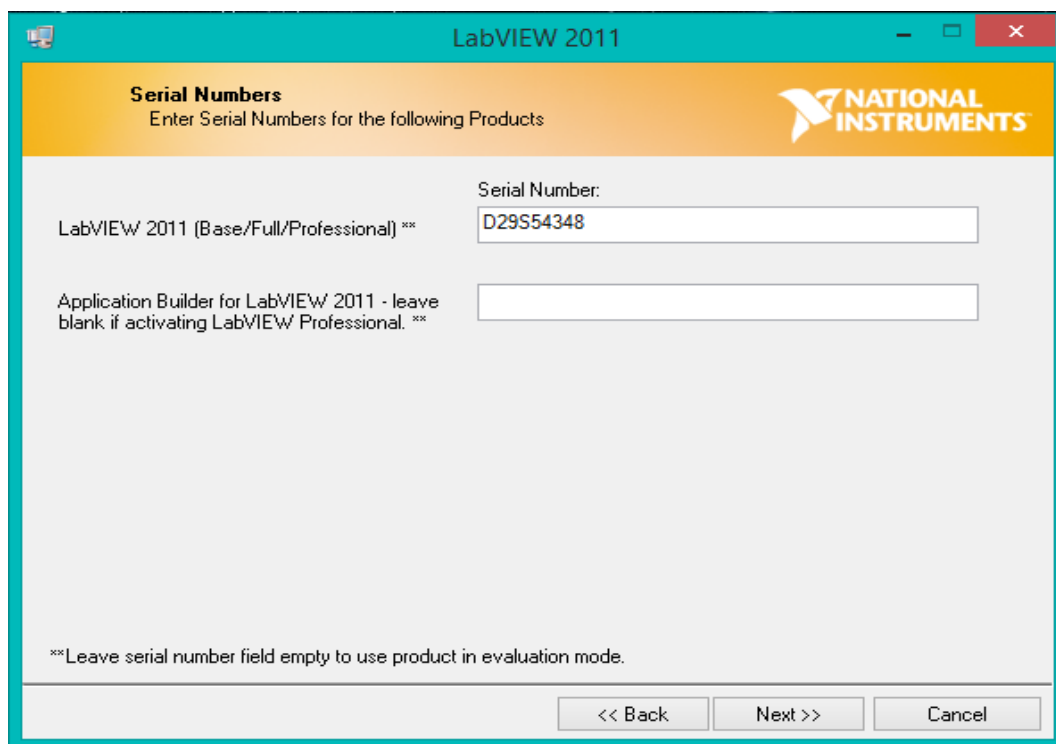
Το επόμενο βήμα είναι η είσοδος του οργανισμού και το όνομα του χρήστη που θα χρησιμοποιεί το πρόγραμμα. (Ο ενδιαφερόμενος θα μπορούσε να συμπληρώσει με ένα οποιοδήποτε τυπικό όνομα το πρώτο πεδίο ενώ το δεύτερο δεν είναι απαραίτητο να συμπληρωθεί).



The image shows the 'User Information' dialog box in LabVIEW 2011. The window title is 'LabVIEW 2011'. The header bar is orange and contains the text 'User Information' and 'Enter the following information.' on the left, and the National Instruments logo on the right. The main area is light gray and contains two text input fields: 'Full Name:' with the value 'Constantin' and 'Organization:' which is empty. At the bottom, there are three buttons: '<< Back', 'Next >>', and 'Cancel'.

Εικόνα 3: Εισαγωγή προσωπικών στοιχείων

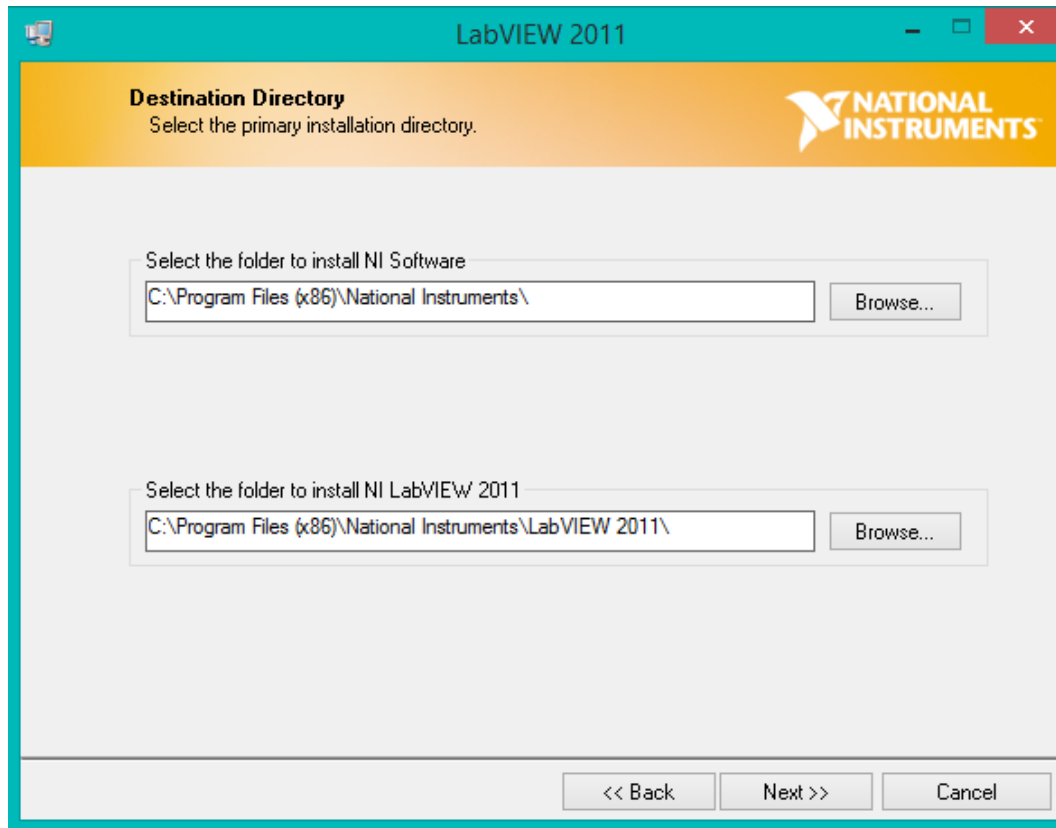
Το επόμενο βήμα είναι η εισαγωγή του σειριακού αριθμού του προγράμματος όπου ταυτοποιείται και η γνησιότητα του λογισμικού



The image shows the 'Serial Numbers' dialog box in LabVIEW 2011. The window title is 'LabVIEW 2011'. The header bar is orange and contains the text 'Serial Numbers' and 'Enter Serial Numbers for the following Products' on the left, and the National Instruments logo on the right. The main area is light gray and contains two rows of text input fields. The first row is for 'LabVIEW 2011 (Base/Full/Professional) **' with the serial number 'D29S54348'. The second row is for 'Application Builder for LabVIEW 2011 - leave blank if activating LabVIEW Professional. **' with an empty field. At the bottom, there are three buttons: '<< Back', 'Next >>', and 'Cancel'. A footnote at the bottom left reads: '**Leave serial number field empty to use product in evaluation mode.'

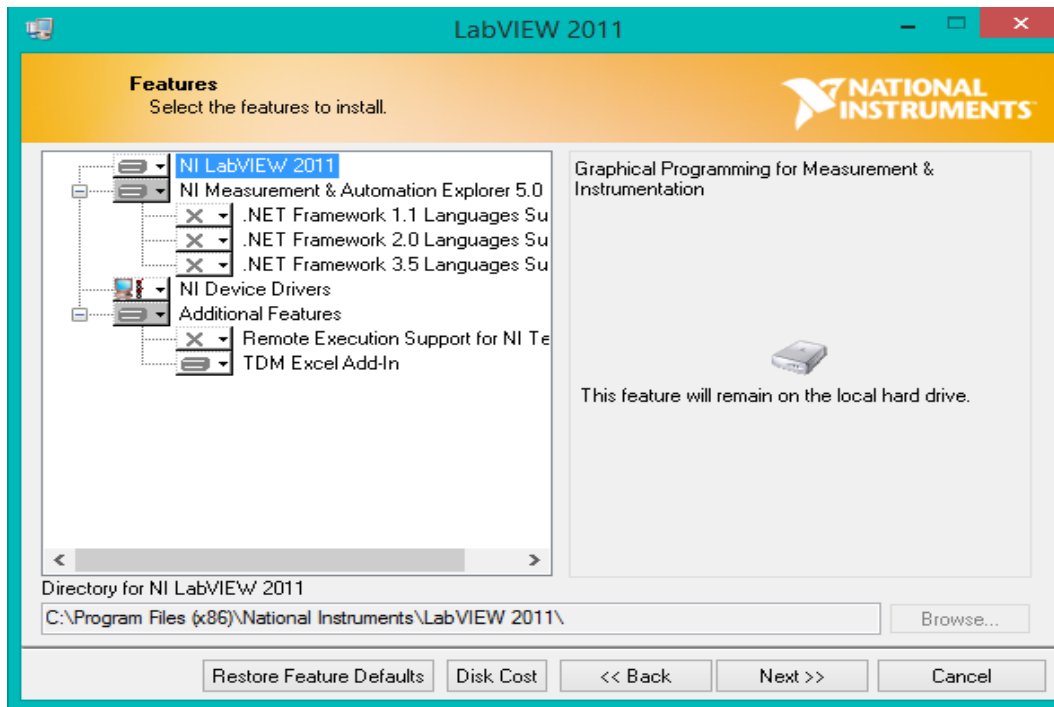
Εικόνα 4: Εισαγωγή σειριακού αριθμού

Στην συνέχεια ο χρήστης ορίζει σε ποια διαδρομή (path) ή σκληρό δίσκο επιθυμεί να εγκαταστήσει το λογισμικό.



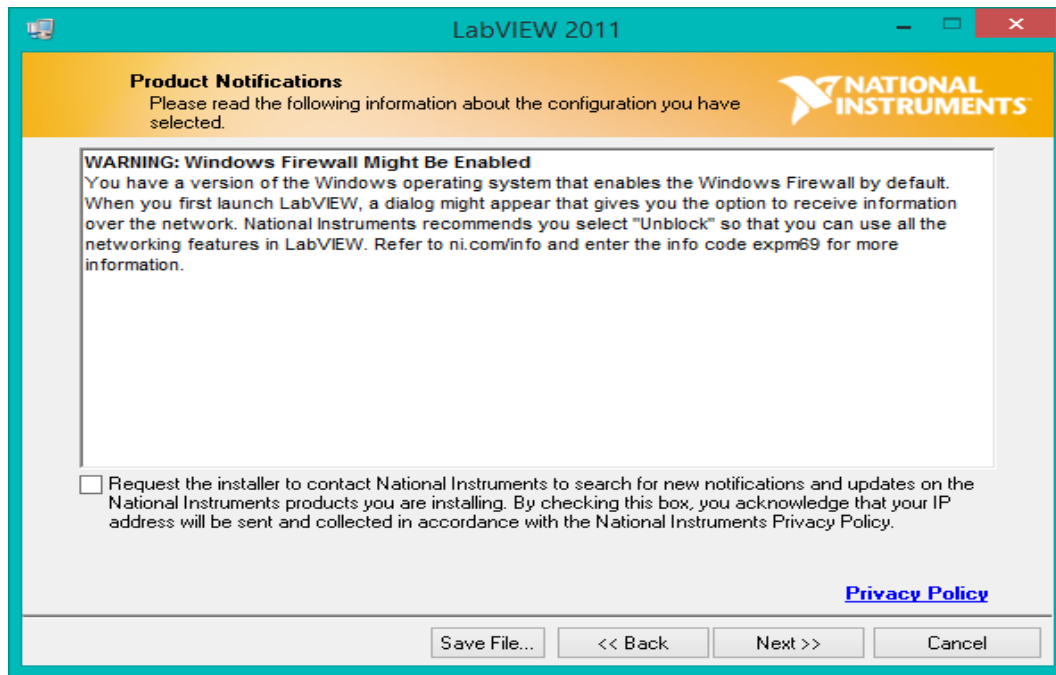
Εικόνα 5: Ορισμός διαδρομών εγκατάστασης λογισμικού

Μετάπειτα δίνεται η επιλογή στον χρήστη να εγκαταστήσει κατά το δοκούν επιπλέον βοηθητικά προγράμματα. Είναι απαραίτητο να εγκατασταθούν όσα είναι προεπιλεγμένα ώστε το LabVIEW να είναι όσο πιο πλήρες γίνεται. Ο χρήστης αν θέλει να μην εγκαταστήσει κάποιο από αυτά θα πρέπει να γνωρίζει αν όντως το χρειάζεται για να μην προκληθούν προβλήματα στην ανάπτυξη του κώδικα του.



Εικόνα 6: Επιλογή βοηθητικών προγραμμάτων προς εγκατάσταση

Το επόμενο παράθυρο ενημερώνει τον χρήστη ότι κατά την πρώτη φορά εκτέλεσης του LabVIEW θα του ζητηθεί να εξαιρέσει το πρόγραμμα από τείχος προστασίας των Windows (firewall). Καλό θα ήταν κατά την εκτέλεση του LabVIEW το τείχος προστασίας να είναι απενεργοποιημένο για τα ιδιωτικά δίκτυα διότι ακόμα και αν το LabVIEW εξαιρεθεί από το τείχος προστασίας, υπάρχει μεγάλη πιθανότητα κάποια άλλη υπηρεσία του (service) να είναι ακόμα περιορισμένη από αυτό. Επίσης ζητάει την άδεια του χρήστη για την αποστολή ειδοποιήσεων και μηνυμάτων ενημέρωσης για τα προϊόντα της National Instruments που εγκαθιστούνται. Καλό θα ήταν να μην δοθεί αυτή η επιλογή.



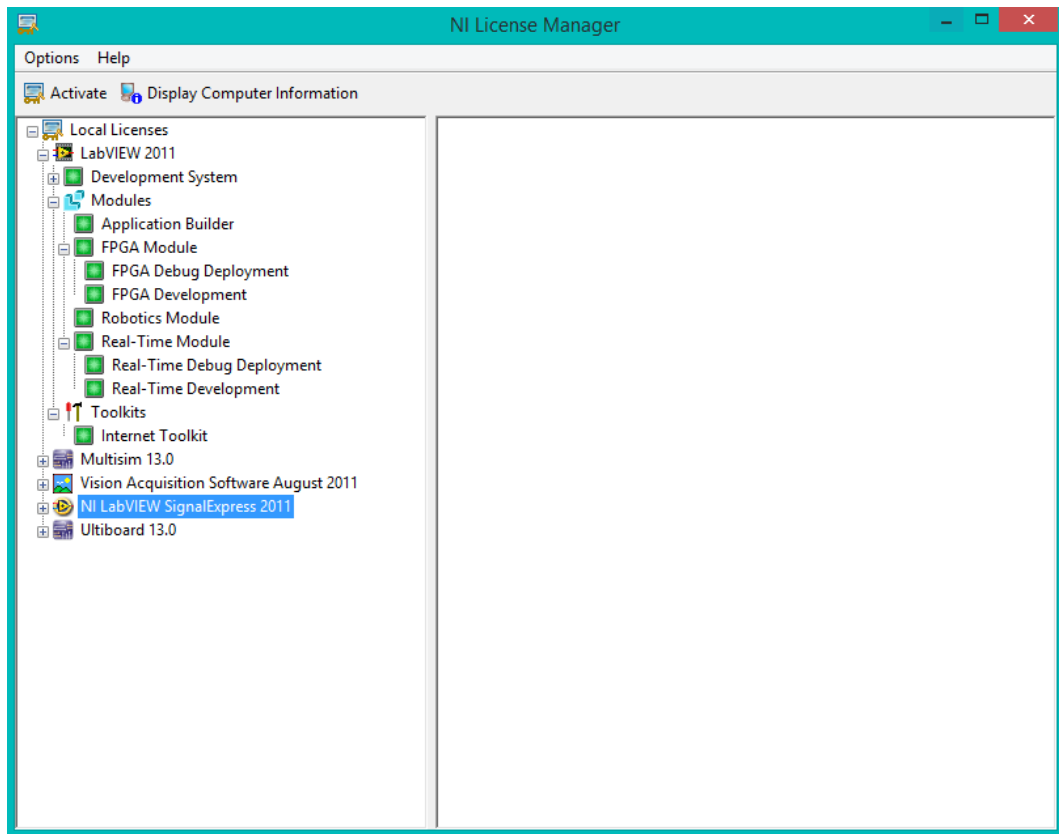
Εικόνα 7: Μήνυμα απενεργοποίησης του τείχους προστασίας

Περνώντας στο επόμενο βήμα και αφού τελειώσει η εγκατάσταση του λογισμικού, εμφανίζεται ένα παράθυρο που ενημερώνει τον χρήστη αν επιθυμεί να εγκαταστήσει επιπλέον οδηγούς υλικού (hardware drivers) στον υπολογιστή του. Ο χρήστης πρέπει να εντοπίσει τους οδηγούς στον φάκελο εγκατάστασης του LabVIEW και στην συνέχεια να τους εγκαταστήσει.



Εικόνα 8: Μήνυμα εγκατάστασης επιπλέον οδηγών

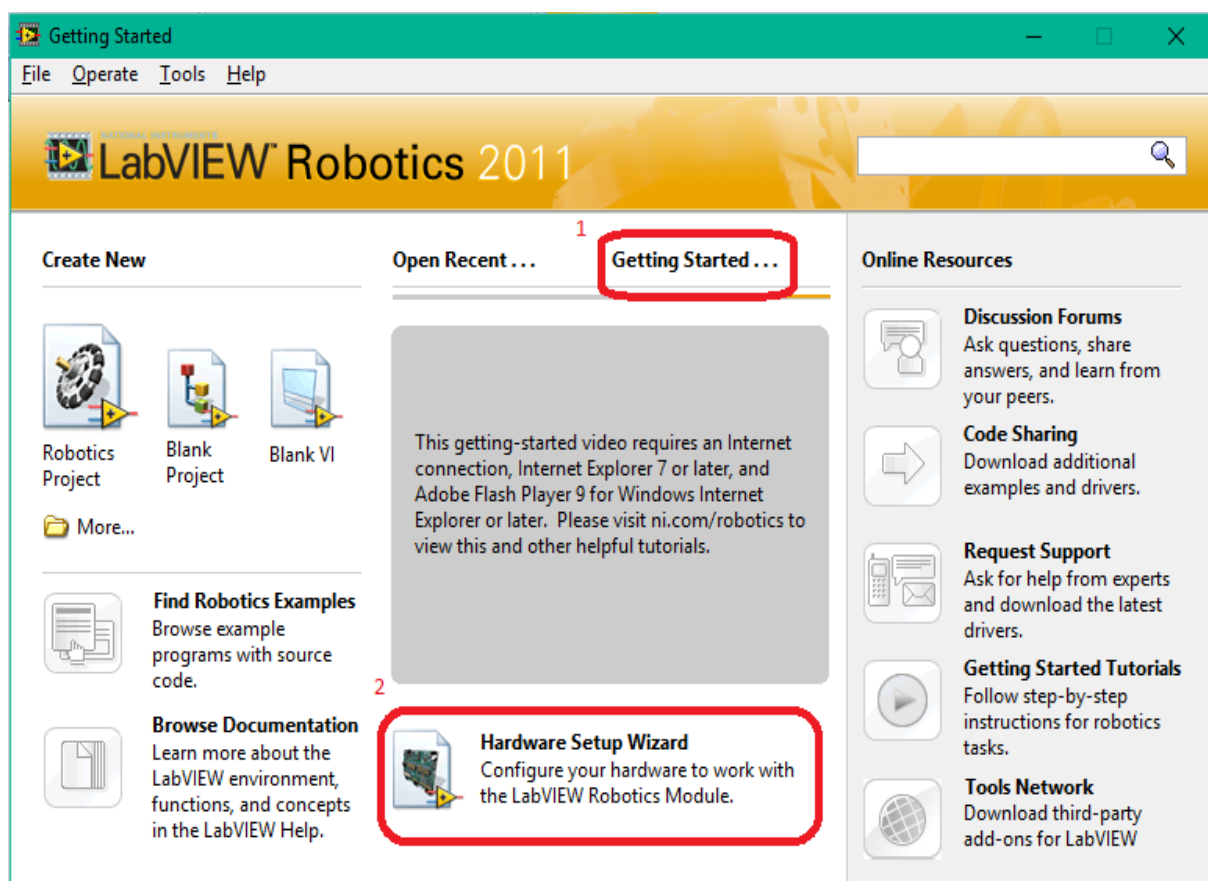
Με την περάτωση του τελευταίου βήματος ολοκληρώνεται και η εγκατάσταση του LabVIEW. Η διαδικασία της εγκατάστασης των υπόλοιπων τμημάτων του λογισμικού είναι η ίδια έτσι αν ακολουθηθεί ο ίδιος οδηγός δεν θα υπάρξει καμιά δυσκολία. Με το τέλος της εγκατάστασης όλων των τμημάτων λογισμικού, ανοίγοντας το βοηθητικό πρόγραμμα NI License Manager μπορεί κανείς να δει όλα τα εγκαταστημένα προγράμματα που φέρουν την υπογραφή της National Instruments. Έτσι ο χρήστης μπορεί να ενημερωθεί για ποια τμήματα λογισμικού έχουν εγκατασταθεί και ενσωματωθεί σωστά κάτω από το γονικό πρόγραμμα, στην περίπτωση αυτή το LabVIEW. Το λογισμικό το οποίο είναι έτοιμο προς χρήση μαρκάρεται με ένα πράσινο τετράγωνο.



Εικόνα 9: Παράθυρο NI License Manager

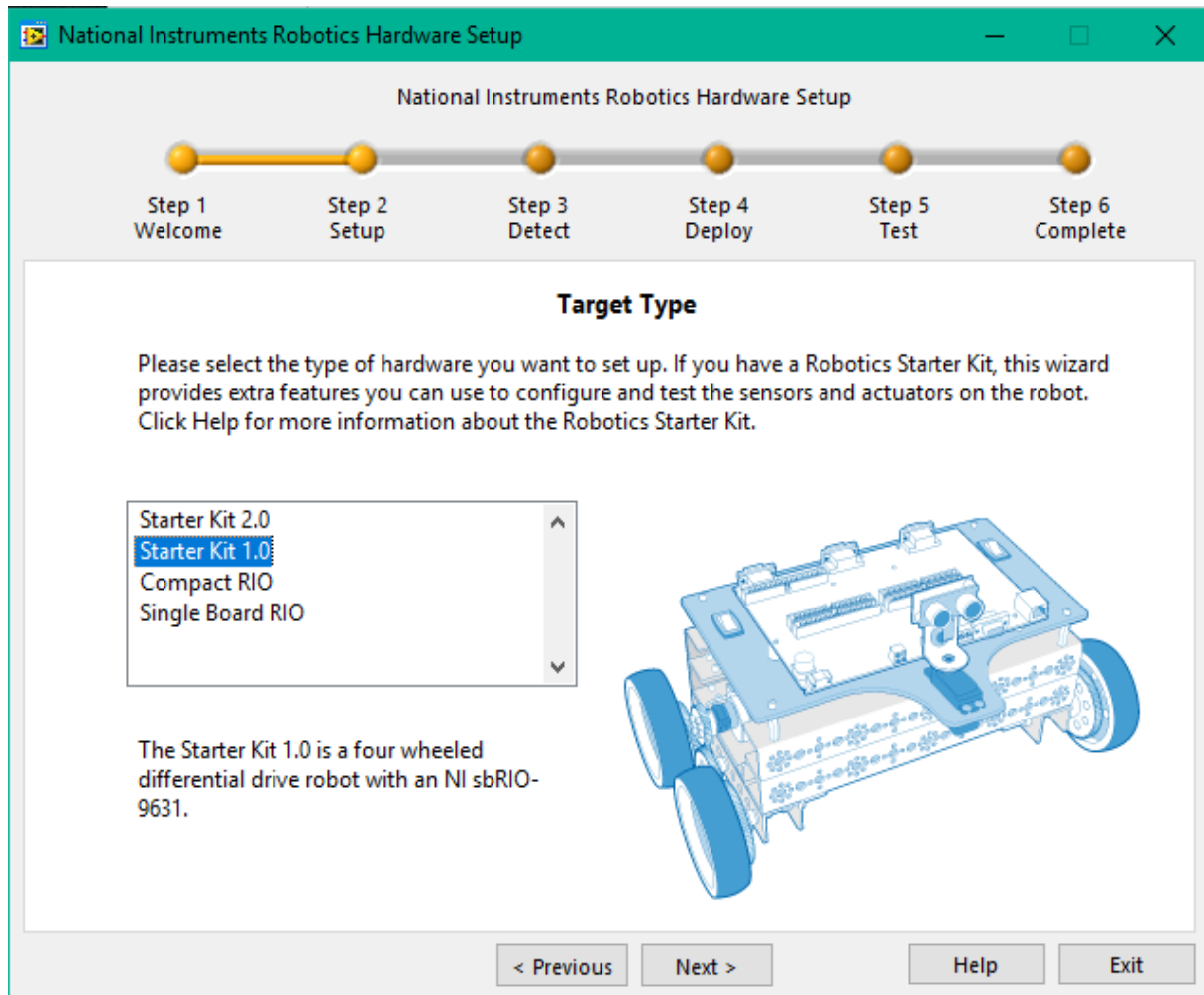
Παράρτημα 2^ο: Διασύνδεση του ρομποτικού οχήματος με τον υπολογιστή

Για να εκτελεστεί ο κώδικας που συγγράφηκε στο LabVIEW είναι απαραίτητη η ζεύξη του ρομποτικού οχήματος με τον host υπολογιστή του χρήστη. Για την εγκατάσταση των δικτυακών παραμέτρων του ρομποτικού οχήματος είναι απαραίτητη η εκκίνηση του Robotics Module στον host υπολογιστή, να επιλεγεί η καρτέλα “Getting Started...” και στην συνέχεια να ξεκινήσει ο οδηγός “Hardware Setup Wizard”. Κατά την διάρκεια της διασύνδεσης θα πρέπει ο διακόπτης «MASTER” στο ρομποτικό όχημα να είναι στην θέση ON.



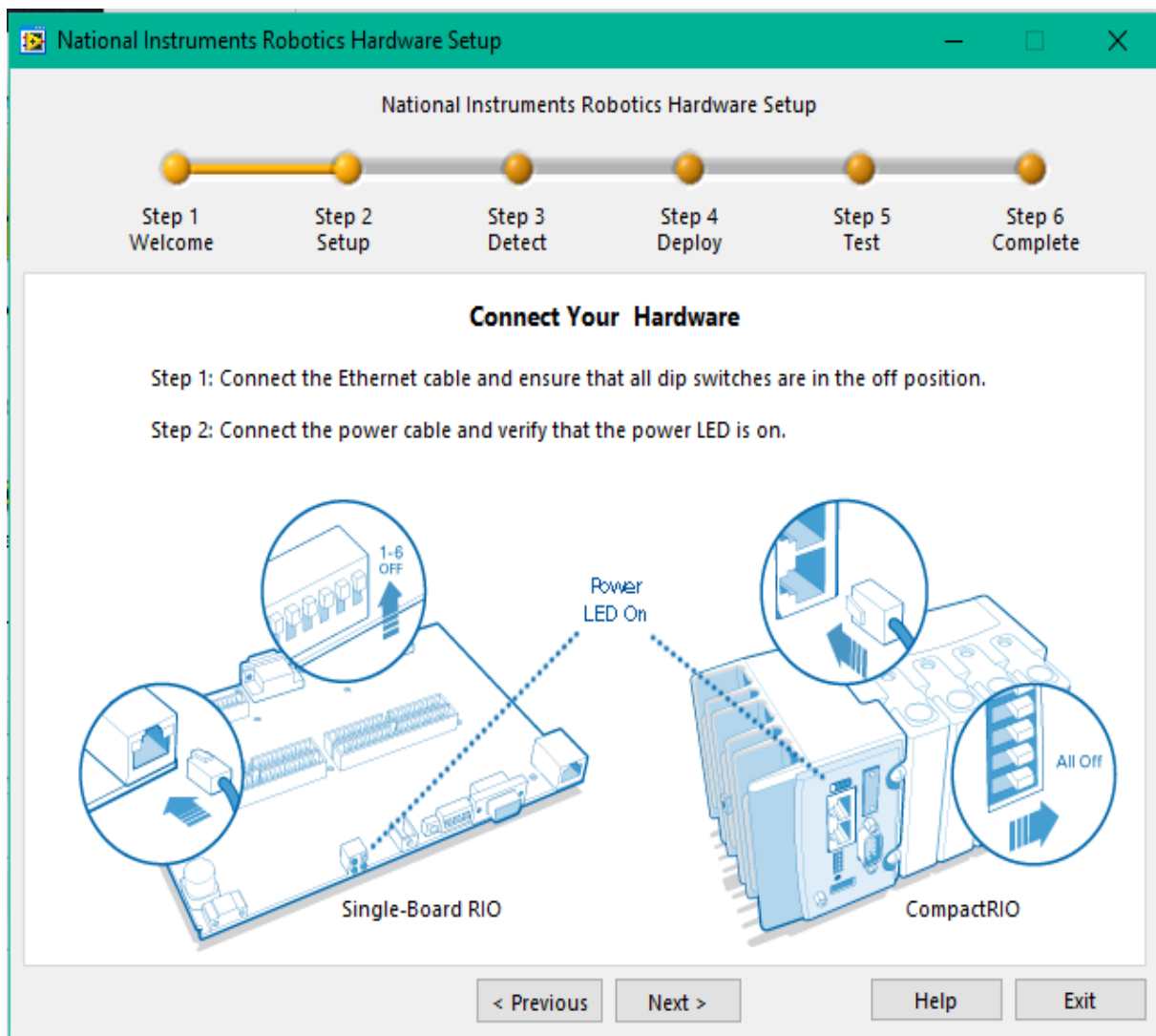
Εικόνα 1: Παράθυρο εκκίνησης του Robotics Module του LabVIEW

Επιλέγοντας το “Hardware Setup Wizard” εμφανίζεται ένα παράθυρο όπου δίνει στον χρήστη τη δυνατότητα να επιλέξει την πλατφόρμα που επιθυμεί, στην προκειμένη περίπτωση την πλατφόρμα “Starter Kit 1.0”.



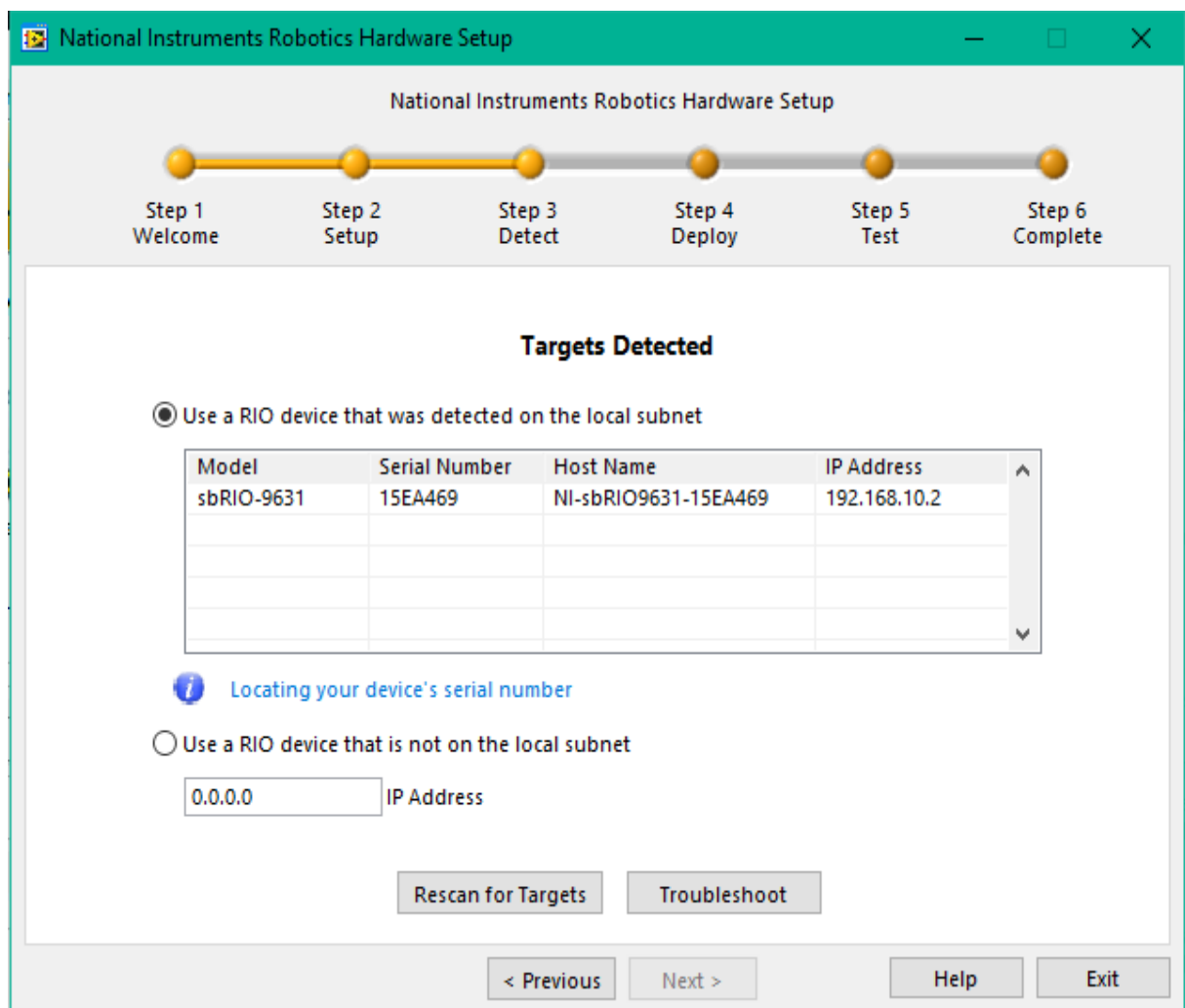
Εικόνα 2: Παράθυρο επιλογής αναπτυξιακής πλατφόρμας

Έπειτα το επόμενο παράθυρο ενημερώνει τον χρήστη να συνδέσει το καλώδιο Ethernet στο SbRIO και να απενεργοποιήσει τα DIP switches. φέροντας τα προς τα πάνω. Στην συγκεκριμένη πτυχιακή εργασία κρίθηκε σκόπιμο το ρομπότ να μην είναι συνδεδεμένο ενσύρματα με τον υπολογιστή με αποτέλεσμα να χρησιμοποιηθεί ένας δρομολογητής , ο οποίος συνδέεται ενσύρματα με το SbRIO και στην συνέχεια ο δρομολογητής ασύρματα με τον υπολογιστή, αυτό όμως δεν αλλάζει την διαδικασία διασύνδεσης. Εκείνη την στιγμή στο SbRIO θα πρέπει να είναι ενεργό το πάνω δεξιά LED μόνο, το οποίο υποδηλώνει την ορθή τροφοδοσία του.



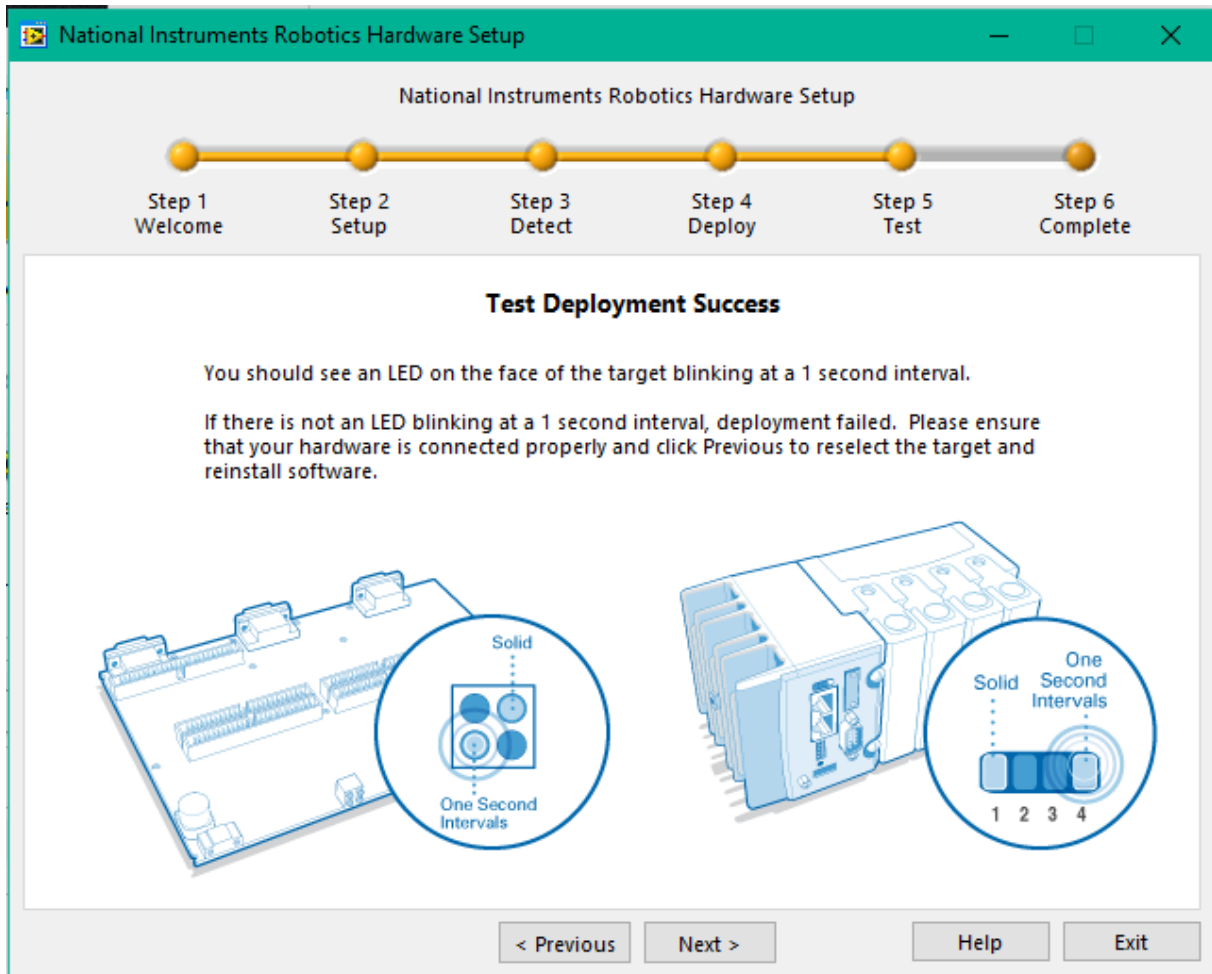
Εικόνα 3: Παράθυρο σύνδεσης του Ethernet

Στην συνέχεια περιμένοντας μερικά δευτερόλεπτα ώστε να ολοκληρωθεί η εύρεση των συσκευών ο χρήστης βρίσκεται αντιμέτωπος με το παράθυρο των ενεργών συσκευών στο δίκτυο. Το LabVIEW είναι ικανό να διασυνδεθεί με πολλαπλές συσκευές ταυτόχρονα. Έτσι κάθε συσκευή επιστρέφει στο LabVIEW πέρα από την IP σύνδεσης και το σειριακό αριθμό ώστε να μπορεί γίνει φυσική αναγνώριση από τον χρήστη (στο SbRIO-9631 βρίσκεται στο πίσω μέρος της πλακέτας). Μόλις επιλεγθεί μια συσκευή από τον χρήστη τότε θα μπορέσει να συνεχίσει τον οδηγό εγκατάστασης.



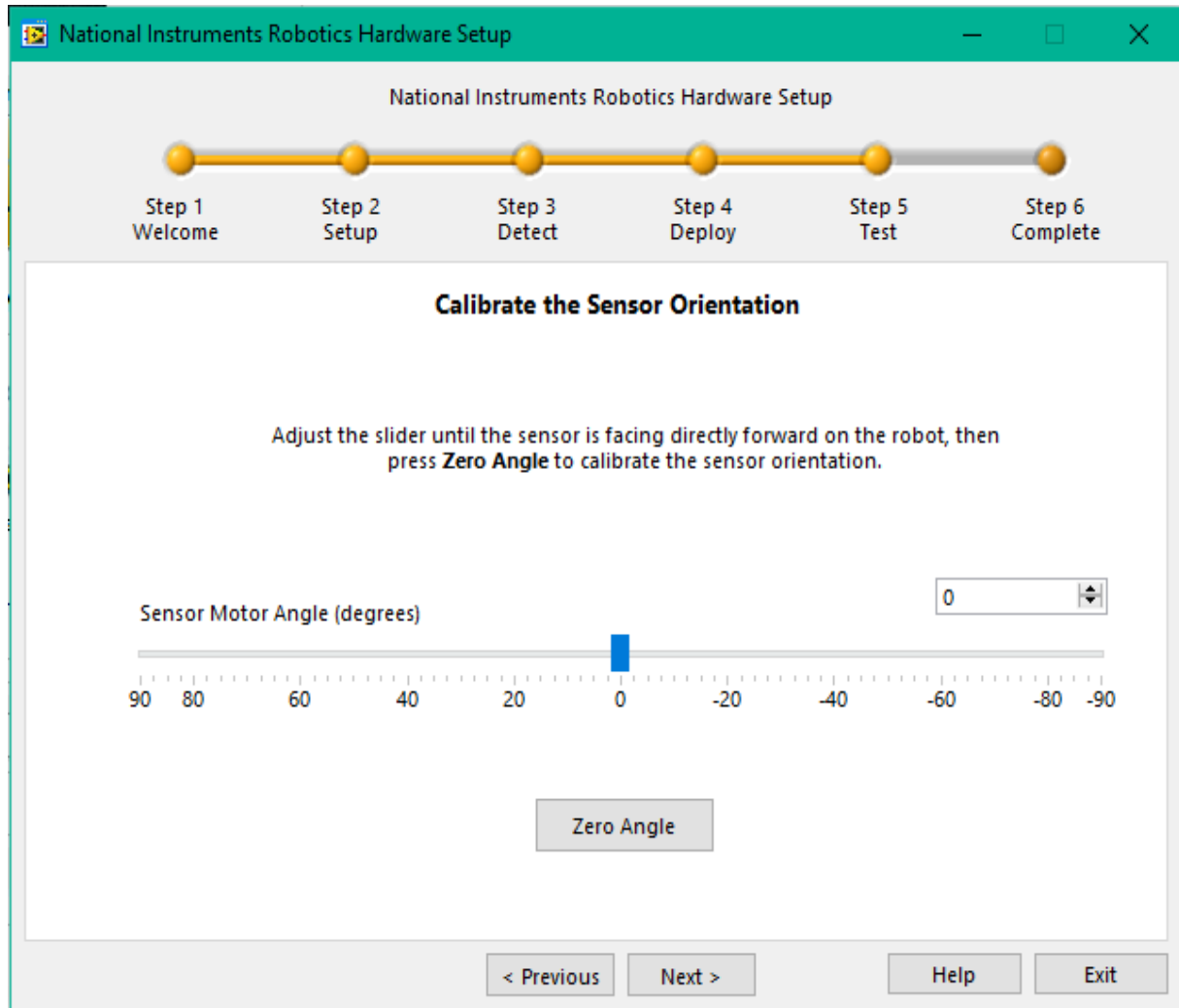
Εικόνα 4: Παράθυρο εύρεσης συσκευών

Ο χρόνος για την εγκατάσταση του NI Starter Kit είναι μερικά λεπτά. Αν η εγκατάσταση είναι επιτυχής τότε ο χρήστης θα είναι σε θέση να ελέγξει αν οι αισθητήρες και τα μοτέρ λειτουργούν σωστά. Αν αποτύχει η εγκατάσταση των ρυθμίσεων για το ρομπότ τότε θα εμφανιστεί ένα παράθυρο που θα περιέχει ένα κωδικό σφάλματος καθώς και μια περιεκτική επεξήγηση του. Στο διαδίκτυο θα μπορεί κανείς να βρει περισσότερες πληροφορίες για το εκάστοτε σφάλμα.



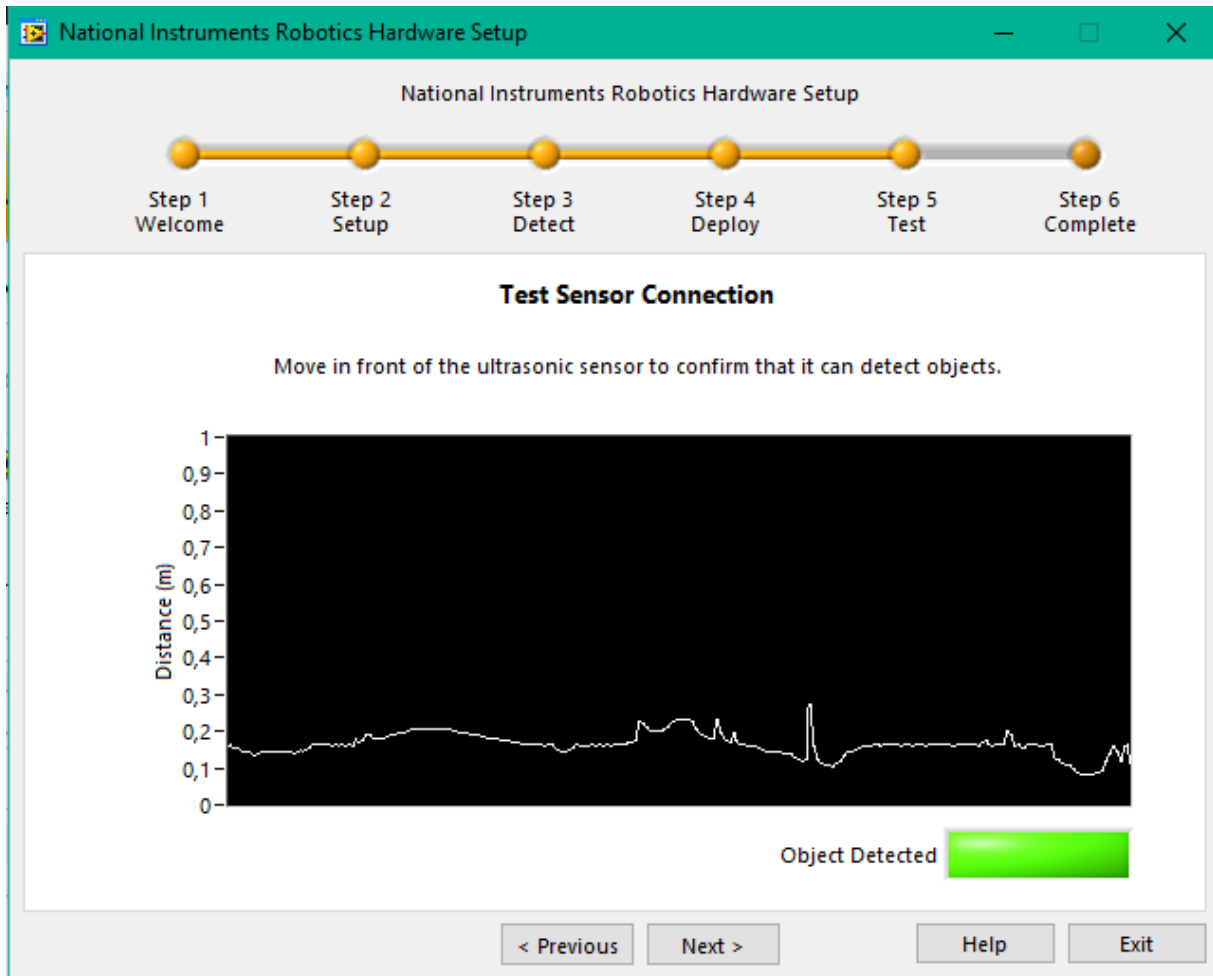
Εικόνα 5: Παράθυρο επιτυχημένης σύνδεσης του NI Starter Kit

Το επόμενο παράθυρο είναι υπεύθυνο για τον ορθό προσανατολισμό του αισθητήρα υπερήχων του ρομπότ, όπου κινείται με την βοήθεια ενός σερβομοτέρ, σε ευθεία θέση.



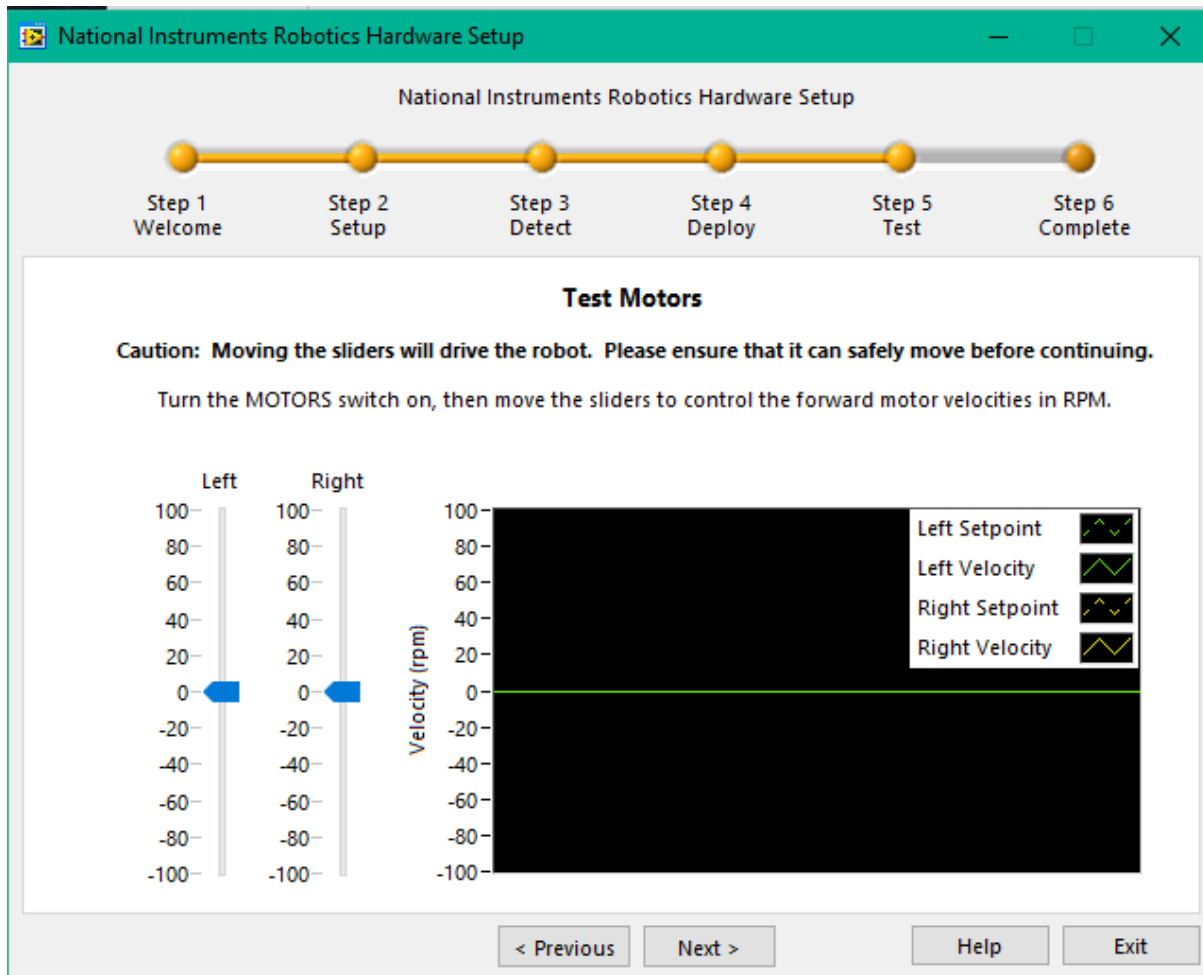
Εικόνα 6: Παράθυρο αρχικοποίησης του σερβομοτέρ

Στην συνέχεια εξακριβώνεται αν λειτουργεί ο αισθητήρας υπερήχων με ένα γράφημα που αναπαράγει σε πραγματικό χρόνο αν εντοπίζεται κάποιο εμπόδιο. Για τον καλύτερο έλεγχο του αισθητήρα συνιστάται να τοποθετηθεί το ρομπότ μπροστά από έναν τοίχο και στην συνέχεια ανάμεσα στον τοίχο και το ρομπότ ένα αντικείμενο ώστε να υπολογισθεί η απόσταση του ρομπότ από τον τοίχο.



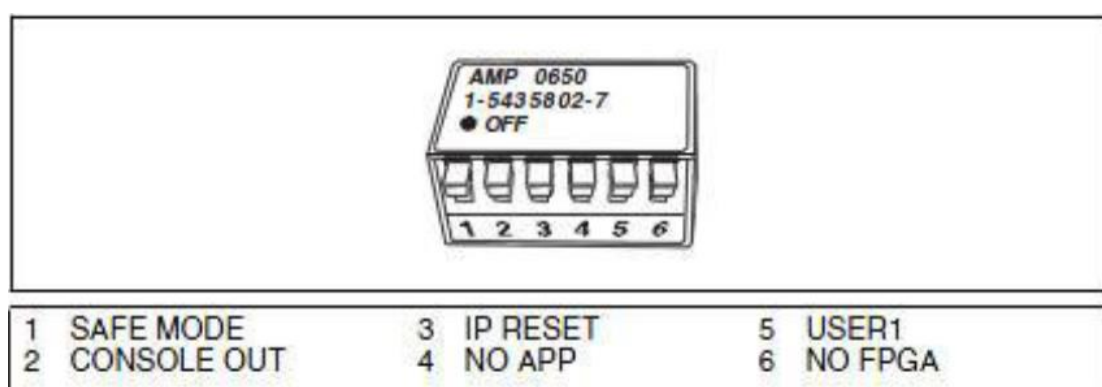
Εικόνα 7: Παράθυρο ελέγχου λειτουργίας αισθητήρα

Τέλος γίνεται ένας έλεγχος των οπτικών κωδικοποιητών και των μοτέρ. Μπορεί κανείς να τοποθετήσει το NI Starter Kit σε ένα κουτί έτσι ώστε οι ρόδες του να μην έχουν επαφή με το έδαφος και έπειτα να μετακινήσει τους sliders και να δει τα αποτελέσματα του γραφήματος.



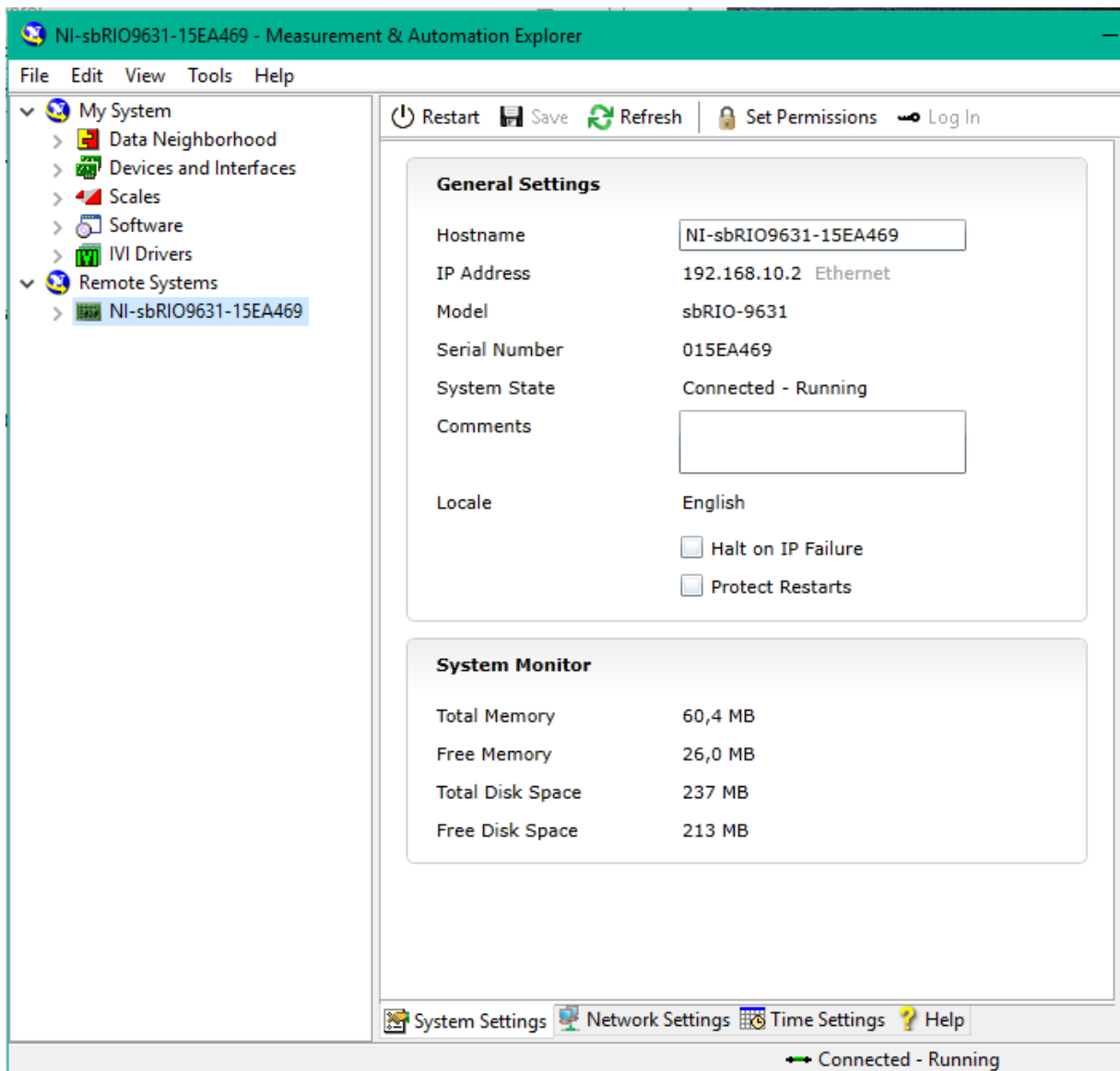
Εικόνα 8: Παράθυρο ελέγχου οπτικών κωδικοποιητών και μοτέρ

Αν κριθεί απαραίτητο να αλλαχθεί για κάποιο λόγο η IP διεύθυνση του ρομπότ αυτό είναι εφικτό με την βοήθεια των DIP διακοπών που βρίσκονται στο μπροστά του μέρος. Ρυθμίζοντας τον διακόπτη Ασφαλούς λειτουργίας και τον διακόπτη IP RESET στη θέση ON και πατώντας το διακόπτη επαναφοράς (reset) του SbRIO πραγματοποιείται η επαναφορά της διεύθυνση IP στο 0.0.0.0. Η ενδεικτική λυχνία κατάστασης θα αναβοσβήνει 3 φορές διαδοχικά για να δείξει ότι το SbRIO βρίσκεται σε ασφαλή λειτουργία.



Εικόνα 9: Διακόπτες DIP στην αναπτυξιακή πλατφόρμα SbRIO-9631

Για να ανατεθεί μια νέα διεύθυνση IP είναι απαραίτητο οι διακόπτες να επανέλθουν στη θέση OFF. Έτσι το SbRIO είναι έτοιμο να δεχθεί μια δυναμική ή στατική IP. Για να ελεγχθεί αν το SbRIO είναι συνδεδεμένο με τον υπολογιστή δίνεται ένα πρόγραμμα από την National Instruments που ονομάζεται Measurement And Automation (MAX) που εγκαθίσταται μαζί με το LabVIEW. Με το πρόγραμμα αυτό μπορεί κανείς να αλλάξει τις ρυθμίσεις δικτύου, να ελέγξει πόση RAM ή ROM χρησιμοποιεί εκείνη την στιγμή το SbRIO ή να βρει άλλες χρήσιμες πληροφορίες για την κατάσταση του.

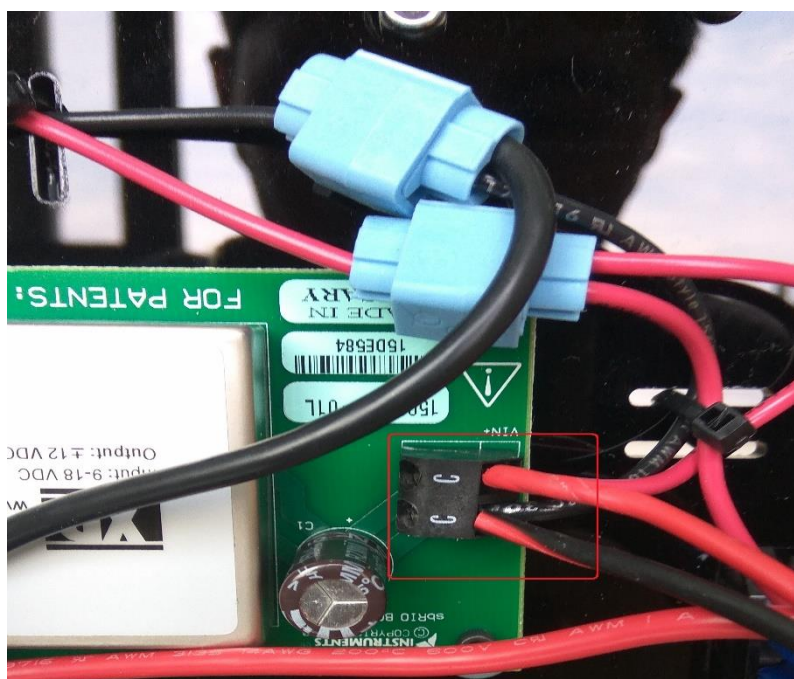


Εικόνα 10: Παράθυρο Measurement And Automation

Παράρτημα 3^ο: Τοποθετώντας ένα δρομολογητή στο ρομποτικό όχημα

Μια έλλειψη που παρατηρήθηκε στην ρομποτική πλατφόρμα της National Instruments ήταν η ανάγκη ο υπολογιστής να παραμένει φυσικά συνδεδεμένος με τον ελεγκτή SbRIO-9631 για την αποστολή δεδομένων στον υπολογιστή καθώς ο δεύτερος δεν ήταν σε θέση να μεταδώσει ασύρματα τα δεδομένα. Έτσι τοποθετήθηκε ένας δρομολογητής ανάμεσα στον ελεγκτή και στον host υπολογιστή. Ο δρομολογητής συνδέθηκε ενσύρματα με το SbRIO-9631 και ασύρματα με τον host υπολογιστή μέσω WIFI δικτύου. Απαραίτητη προϋπόθεση είναι ο υπολογιστής του χρήστη να έχει ασύρματη κάρτα δικτύου.

Η τοποθέτηση του δρομολογητή έγινε στο πάνω μέρος του ρομποτικού οχήματος ενώ για την τροφοδοσία του χρησιμοποιήθηκε η μπαταρία του οχήματος. Η τάση λειτουργίας του δρομολογητή είναι 12V καθώς και το ρεύμα λειτουργίας στο 1A. Με την βοήθεια ενός βολτόμετρου εντοπίστηκε πως στην είσοδο του οδηγού των μοτέρ η τάση έφτανε τα 12,1V. Έτσι σε εκείνο το σημείο συνδέθηκαν τα καλώδια για την τροφοδοσία του δρομολογητή.



Εικόνα 1: Σημείο σύνδεσης τροφοδοσίας δρομολογητή

Στην συνέχεια έγιναν οι απαραίτητες ρυθμίσεις στον δρομολογητή ώστε να επιτευχθεί η απαραίτητη ζεύξη μεταξύ του SbRIO-9631 και του hostυπολογιστή. Οι παρακάτω οδηγίες περιγράφουν τις αλλαγές στις ρυθμίσεις του δρομολογητή Technicolor TG703. Αρχικά απενεργοποιήθηκε το τείχος προστασίας του δρομολογητή ώστε όλη η δικτυακή κίνηση να επιτρέπεται να περνά από αυτό. Συνήθως αυτή η λειτουργία βρίσκεται στις ρυθμίσεις ασφάλειας του κάθε δρομολογητή.

Home > Toolbox > Firewall Overview | Configure

Firewall

This page summarizes the overall security policy configured on your Technicolor Gateway.

- Security Settings**
 - Security Level: Disabled [Details...](#)

Disable the firewall. All traffic is allowed to pass through your gateway. Game and Application sharing is allowed by the firewall.

Pick a task...

- Assign a game or application to a local network device

Εικόνα 2: Απενεργοποίηση τείχους προστασίας του δρομολογητή

Βλέποντας την τοπολογία του δικτύου και ποιες συσκευές είναι συνδεδεμένες πρέπει να ορίσουμε για το SbRIO-9631 να έχει στατική διεύθυνση IP καθώς και στον υπολογιστή να ανατίθεται κάθε φορά η ίδια IP διεύθυνση.

Home > Home Network > Devices Overview

Local Network Devices

- Detected Device(s)**

The table below contains the list of devices the Technicolor Gateway detected on your local network. Click on a device name to get more information on a device.

Name	IP Address	Interface
Technicolor TG703	192.168.10.254	
DESKTOP-6F248U2	192.168.10.4	WLAN
NI-sbRIO9631-15EA469	192.168.10.2	ethport1

Pick a task...

- Assign a game or application to a local network device
- Assign the public IP address of a connection to a device

Εικόνα 3: Τοπολογία δικτύου



Εικόνα 4: Ρυθμίσεις host υπολογιστή



Εικόνα 5: Ρυθμίσεις SbRIO-9631

Αν οι ρυθμίσεις έχουν ολοκληρωθεί σωστά τότε με την βοήθεια της εντολής Ping με όρισμα την IP του ελεγκτή SbRIO-9631 μέσα από το command window των Windows είναι δυνατόν να ελεγχθεί η επιτυχία της ζεύξης του SbRIO-9631 με τον δρομολογητή καθώς και η ζεύξη του δρομολογητή με τον υπολογιστή.

```

CA Command Prompt
Microsoft Windows [Version 10.0.15063]
(c) 2017 Microsoft Corporation. All rights reserved.

C:\Users\Constantin719>ping 192.168.10.2

Pinging 192.168.10.2 with 32 bytes of data:
Reply from 192.168.10.2: bytes=32 time=4ms TTL=64
Reply from 192.168.10.2: bytes=32 time=1ms TTL=64
Reply from 192.168.10.2: bytes=32 time=1ms TTL=64
Reply from 192.168.10.2: bytes=32 time=1ms TTL=64

Ping statistics for 192.168.10.2:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 1ms, Maximum = 4ms, Average = 1ms
  
```

Εικόνα 6: Εκτέλεση εντολής Ping με όρισμα την IP του ελεγκτή SbRIO-9631

Βιβλιογραφία

1. “Autonomous Land Vehicles”, Karsten Berns | Ewald von Puttkamer
2. “Mobile Robotics Experiments with DaNI”, DR. Robert King
3. “Autonomous Mobile Robots”, Roland Siegwart | Illah Nourbakhsh
4. “Αρχές Προγραμματισμού Πραγματικού Χρόνου”, Ιωάννης Καλόμοιρος
5. “Προηγμένα Ψηφιακά Συστήματα”, Ιωάννης Καλόμοιρος
6. “Συστήματα Συλλογής Πληροφοριών και Μετρήσεων”, Νικόλας Αρμπατζάνης
7. <http://www.robotpark.com>
8. <http://www.robotplatform.com/>
9. <https://en.wikipedia.org/>
10. <http://www.ni.com/>
11. <https://www.arduino.cc/en/Tutorial/PWM>
12. <https://barrgroup.com/Embedded-Systems/How-To/PWM-Pulse-Width-Modulation>
13. <http://www.dspguide.com/>
14. <https://www.xilinx.com/>