

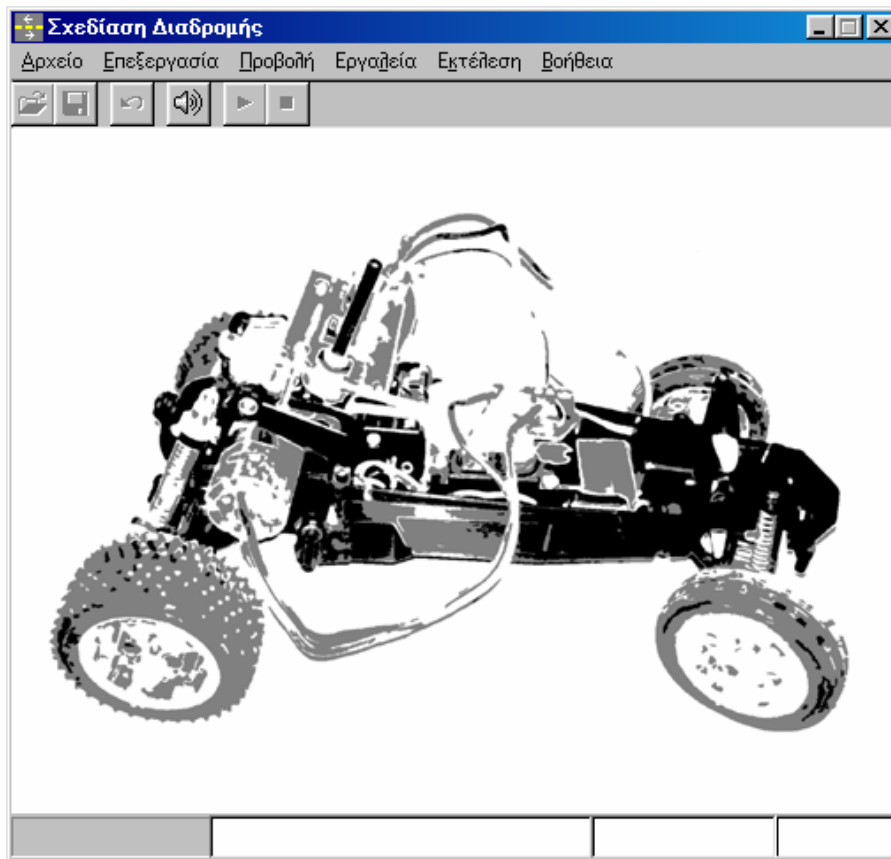


ΤΕΧΝΟΛΟΓΙΚΟ ΕΚΠΑΙΔΕΥΤΙΚΟ ΙΔΡΥΜΑ ΚΡΗΤΗΣ, ΠΑΡΑΡΤΗΜΑ ΧΑΝΙΩΝ  
ΣΧΟΛΗ ΤΕΧΝΟΛΟΓΙΚΩΝ ΕΦΑΡΜΟΓΩΝ  
ΤΜΗΜΑ ΗΛΕΚΤΡΟΝΙΚΗΣ

## “Πρόγραμμα Τηλεκατεύθυνσης Ηλεκτροκίνητου Μοντέλου Οχήματος”

Πτυχιακή Εργασία του σπουδαστή  
Νικόλαου Κομπολάκη

Εισηγητής καθηγητής  
Ιωάννης Εμμ. Τομαδάκης



## “Electric Vehicle Remote Control Program”

Thesis of Nick Kompolakis

Supervisor: Ioannis Emm. Tomadakis

XANIA  
ΔΕΚΕΜΒΡΙΟΣ 2004

## **ΠΡΟΛΟΓΟΣ**

Αυτή η εφαρμογή ελέγχει την κίνηση ενός τηλεχειριζόμενου οχήματος μέσω της σειριακής θύρας ενός Η/Υ. Ο έλεγχος γίνεται στην ταχύτητα και τη γωνία στροφής του οχήματος. Το πρόγραμμα ελέγχου της εφαρμογής έχει υλοποιηθεί με τη VISUAL BASIC.

Ο έλεγχος γίνεται σε πραγματικό και μη πραγματικό χρόνο. Στον έλεγχο μη πραγματικού χρόνου ο χειριστής σχεδιάζει μια διαδρομή, την οποία ακολουθεί στη συνέχεια το όχημα. Στον έλεγχο πραγματικού χρόνου ελέγχονται η ταχύτητα και η γωνία στροφής κατά τη διάρκεια της κίνησης του οχήματος.

Ο έλεγχος του οχήματος επιτυγχάνεται με τη χρησιμοποίηση δύο ψηφιακών ποτενσιόμετρων, τα οποία έχουν αντικαταστήσει δύο αναλογικά στην κονσόλα τηλεχειρισμού του οχήματος.

## **FOREWORD**

Using this program we can remote control an electric model of a car through the serial port of a PC. The program controls the velocity and the angle of the vehicle's turn. The program has been written using VISUAL BASIC.

The control can be in real time and in unreal time. In the unreal time control the user draws a route, which is followed by the model afterwards. In the real time control, the velocity and the angle of the vehicle's turn are controlled during the vehicle's kinesis.

The program controls the vehicle by using two digital potentiometers, which replaced two analog ones on the vehicle's radio control system.

# ΠΕΡΙΕΧΟΜΕΝΑ

<b>1. ΕΙΣΑΓΩΓΗ</b> .....	σελ.5
1.1 Νόμοι και Ορισμοί.....	σελ.5
1.2 Λόγοι χρησιμοποίησης των ρομπότ.....	σελ.6
1.3 Παράμετροι απόδοσης των ρομπότ.....	σελ.6
1.4 Εφαρμογές.....	σελ.7
<b>2. ΑΥΤΟΝΟΜΑ ΑΥΤΟΚΙΝΟΥΜΕΝΑ ΟΧΗΜΑΤΑ</b> .....	σελ.10
2.1 Ορισμοί και εφαρμογές.....	σελ.10
2.2 Δομή και λειτουργία.....	σελ.10
2.3 Προβλήματα.....	σελ.11
<b>3. ΥΛΙΚΟ</b> .....	σελ.13
3.1 Πάτωμα.....	σελ.13
3.2 Κινητήρας.....	σελ.13
3.3 Έλεγχος διεύθυνσης και ταχύτητας.....	σελ.13
3.4 Τηλεκατεύθυνση.....	σελ.14
3.5 Επικοινωνία με τη σειριακή θύρα (RS232).....	σελ.16
<b>4. ΛΟΓΙΣΜΙΚΟ</b> .....	σελ.19
4.1 Εισαγωγή.....	σελ.19
4.2 Έλεγχος σε πραγματικό χρόνο.....	σελ.19
4.2.1 Παρουσίαση φόρμας ελέγχου.....	σελ.19
4.2.2 Τρόπος λειτουργίας.....	σελ.20
4.3 Σχεδίαση διαδρομής.....	σελ.22
4.3.1 Εισαγωγή.....	σελ.22
4.3.2 Εισαγωγή χάρτη-οδηγού.....	σελ.22
4.3.3 Ρύθμιση παραμέτρων σχεδίασης.....	σελ.23
4.3.4 Διαδικασία προσανατολισμού.....	σελ.25
4.3.5 Τρόπος δειγματοληψίας σημείων.....	σελ.30
4.3.6 Διαδικασία υπολογισμού της γωνίας στροφής.....	σελ.30
4.3.7 Προβολή σημείων καμπής.....	σελ.37
4.3.8 Εκκαθάριση-αναίρεση.....	σελ.38
4.3.9 Αποθήκευση διαδρομής.....	σελ.38

4.3.10 Εκτέλεση διαδρομής.....σελ.39
4.3.11 Διαδικασίες σχεδίασης με ελεύθερη γραμμή.....σελ.39
4.3.12 Διαδικασίες σχεδίαση με ευθύγραμμα τμήματα.....σελ.42
<b>5. ΕΠΙΛΟΓΟΣ.....σελ.46</b>
Βιβλιογραφία – Αναφορές.....σελ.48

# 1. ΕΙΣΑΓΩΓΗ

## 1.1 Νόμοι και Ορισμοί

Η πρώτη αναφορά στον όρο **Ρομπότ** έγινε στην δεκαετία του 1920 από τον Τσέχο συγγραφέα *Capek*. Θεμελιωτής της ρομποτικής υπήρξε ο *Ισαάκ Ασίμοφ*, ο οποίος το 1942 διατύπωσε τους «τρεις νόμους της ρομποτικής»

1. Ένα ρομπότ δεν μπορεί να τραυματίσει ένα ανθρώπινο όν, ή, εξαιτίας αδράνειας του, να επιτρέψει τον τραυματισμό ενός ανθρώπινου όντος.
2. Ένα ρομπότ πρέπει να υπακούει στις εντολές οι οποίες του δίνονται από ανθρώπινα όντα εκτός από αυτές που έρχονται σε αντίθεση με τον πρώτο νόμο.
3. Ένα ρομπότ πρέπει να προστατεύει την ύπαρξη του εφόσον αυτή η προστασία δεν έρχεται σε αντίθεση με τον πρώτο ή το δεύτερο νόμο.

Οι ορισμοί που παρατίθενται παρακάτω αφορούν θεμελιώδεις έννοιες που χρησιμοποιούνται στη ρομποτική:

**Μηχανοποίηση** (*mechanization*):είναι η χρήση μηχανών που αναλαμβάνουν μερικές από τις εργασίες που εκτελούσαν οι άνθρωποι ή τα ζώα.

**Αυτοματοποίηση** (*automation*):είναι η αυτοδιευθυνόμενη παραγωγική δραστηριότητα που είναι αποτέλεσμα του συνδυασμού της μηχανοποίησης και της υπολογιστικής ικανότητας. Τα αυτοματοποιημένα συστήματα χρησιμοποιούν πληροφορία από ανάδραση έτσι ώστε να επιτυγχάνεται η επιθυμητή λειτουργία.

**Εργαλείο** (*tool*):είναι μια συσκευή που χρησιμοποιείται για να εκτελεστεί μια δράση. Όταν χρησιμοποιείται από τον άνθρωπο, ο άνθρωπος παρέχει την ισχύ και τον έλεγχο.

**Μηχανή** (*machine*):είναι μια συσκευή που χρησιμοποιεί μη ανθρώπινη ισχύ για να εκτελέσει μια δράση. Ο έλεγχος σε μια απλή μηχανή γίνεται από τον άνθρωπο.

**Τηλεχειριστής (teleoperator):**είναι μια μηχανή ικανή να δράσει από απόσταση με τον έλεγχο του ανθρώπου.

**Ρομπότ (Robot):**είναι μια ευέλικτη μηχανή που ελέγχει τις δράσεις της σε μια ποικιλία εργασιών χρησιμοποιώντας αποθηκευμένα προγράμματα. Η ευελιξία εξασφαλίζεται από τη δυνατότητα προγραμματισμού. Τα νοήμονα (ευφυή-intelligent) ρομπότ μπορούν να θέσουν τους στόχους τους, να προγραμματίσουν τις δράσεις τους και να διορθώσουν τη λειτουργία τους σύμφωνα με τις μεταβολές στο περιβάλλον τους.

## 1.2 Λόγοι χρησιμοποίησης των ρομπότ

Οι λόγοι που έχουν κάνει τη χρήση των ρομπότ τόσο διευρυμένη και σε κάποιες περιπτώσεις επιτακτική κινούνται στον άξονα 4-D (dirty-difficult-dangerous-dull), δηλαδή σε όσες δουλειές θεωρούνται βρόμικες, δύσκολες, επικίνδυνες ή ανιαρές. Επίσης σε εφαρμογές που θεωρείται οικονομικότερη η χρήση των ρομπότ από τους ανθρώπους, ή, η παραγωγικότητα του ρομπότ είναι μεγαλύτερη συγκρινόμενη με αυτή του ανθρώπου, προτιμάται η χρήση των ρομπότ.

## 1.3 Παράμετροι απόδοσης των ρομπότ

Η απόδοση ενός ρομπότ είναι πολύ κρίσιμη για την επιτυχή χρήση του σε κάποια εφαρμογή. Οι πλέον κρίσιμες παράμετροι για τον καθορισμό της απόδοσης του ρομπότ παρατίθενται παρακάτω:

**Διακριτική Ικανότητα (resolution):**είναι το μέτρο της μικρότερης δυνατής μεταβολής στη μεταβλητή έξοδο μιας συσκευής

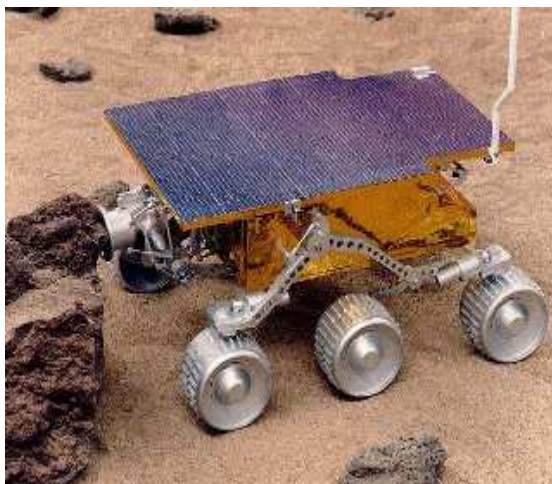
**Ακρίβεια (accuracy):**είναι ο βαθμός στον οποίο η πραγματική τοποθέτηση αντιστοιχεί στην επιθυμητή διατεταγμένη τοποθέτηση.

**Επαναληψιμότητα (repeatability):**είναι το μέτρο της απόκλισης επαναλαμβανόμενων κινήσεων τοποθέτησης (θέσης και προσανατολισμού), κάτω από τις ίδιες συνθήκες περιβάλλοντος, φορτίου και εργασίας.

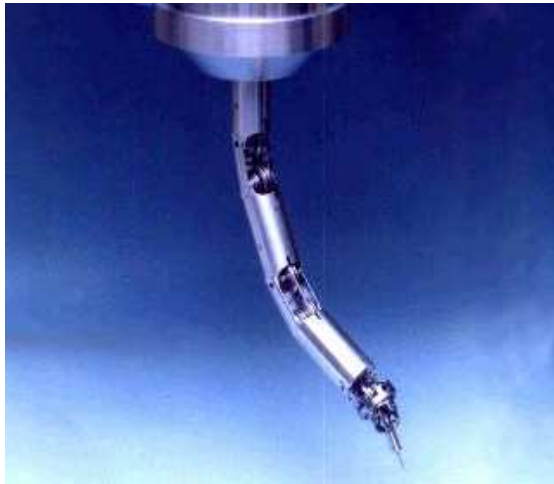
## 1.4 Εφαρμογές

Σε ορισμένους τομείς της επιστήμης και του εμπορίου η συχνότητα χρήσης ρομπότ είναι πολύ μεγάλη. Σε μια σειρά από βιομηχανικές εφαρμογές οι οποίες είναι επικίνδυνες για τον άνθρωπο και η παραγωγικότητα του ρομπότ είναι καλύτερη έχει επικρατήσει η καθολική εφαρμογή των ρομπότ. Η διαχείριση επικίνδυνων υλικών (όπως τα χημικά και τα ραδιενεργά), ο ψεκασμός, η ηλεκτροσυγκόλληση, οι μηχανουργικές κατεργασίες, η συναρμολόγηση οχημάτων, η φόρτωση/ εκφόρτωση και η βαφή είναι εφαρμογές που η χρήση ρομπότ είναι καθολική και θεωρείται αυτονόητη. Στη διαστημική και υποβρύχια εξερεύνηση καθώς και σε επιστημονικές αποστολές σε δυσπρόσιτες περιοχές (όπως ηφαιστεια, υποθαλάσσιες βιοσφαιρές σε μεγάλα βάθη) η χρήση των ρομπότ είναι επίσης πολύ διαδεδομένη. Στην ιατρική έχει αρχίσει η χρήση ρομποτικής τεχνολογίας. Ρομποτικοί βραχίονες χρησιμοποιούνται σαν βοηθοί σε χειρουργικές επεμβάσεις. Επίσης, κατασκευάζονται τεχνητά μέλη με την ικανότητα της αφής και της όρασης. Στην βιομηχανία ημιαγωγών, η χρήση ρομπότ έχει σαν αποτέλεσμα μεγαλύτερη ακρίβεια στο σχεδιασμό και μεγαλύτερη ασφάλεια στην υλοποίηση των σχεδιασμένων ολοκληρωμένων κυκλωμάτων. Έχουν κατασκευαστεί ρομπότ για οικιακές εφαρμογές (όπως είναι ο καθαρισμός δαπέδων) και παιχνίδια ρομπότ για την ψυχαγωγία των παιδιών. Τα τελευταία χρόνια έχουν αναπτυχθεί ειδικά ρομπότ για τη φύλαξη χώρων (σε μουσεία και αποθήκες εργοστασίων).

Στις φωτογραφίες που ακολουθούν φαίνονται κάποιες από τις εφαρμογές που περιγράφονται παραπάνω:



**Σχ.1:** αυτόνομο αυτοκινούμενο όχημα που χρησιμοποιείται για τη μελέτη και την εξερεύνηση πλανητών



Σχ.2: ρομπότ το οποίο χρησιμοποιείται σε χειρουργικές επεμβάσεις σαν βοηθός



Σχ.3: ρομπότ που χρησιμοποιείται βοηθητικά στη διδασκαλία φυσικής σε αμερικάνικο πανεπιστήμιο



Σχ.4: ρομποτικός βραχίονας ο οποίος χρησιμοποιείται στην αυτοκινητοβιομηχανία





Σχ.5: όχημα παιχνίδι το οποίο έχει ενσωματωμένη κάμερα



Σχ.6: ρομποτικό όχημα που χρησιμοποιείται στην μεταφορά κορμών δέντρων για την κοπή τους

## 2. ΑΥΤΟΝΟΜΑ ΑΥΤΟΚΙΝΟΥΜΕΝΑ ΟΧΗΜΑΤΑ

### 2.1 Ορισμοί και εφαρμογές

**Αυτοκινούμενο όχημα** ονομάζεται το όχημα το οποίο περιλαμβάνει ενσωματωμένο κινητήρα ο οποίος του επιτρέπει να κινείται με ενεργειακή αυτονομία. Ένα τέτοιο όχημα μπορεί επιπλέον να τηλεκατευθύνεται από άνθρωπο ή υπολογιστή. Τότε καλείται **τηλεχειριζόμενο αυτοκινούμενο όχημα**. Ένα ευφυές αυτοκινούμενο όχημα μπορεί επιπλέον να διορθώνει ή και να καθορίζει το ίδιο την πορεία που θα ακολουθήσει προκειμένου να καταλήξει σε συγκεκριμένη θέση του χώρου που του ζητείται από το χρήστη. Τα ευφυή αυτοκινούμενα οχήματα συνήθως καλούνται **αυτόνομα αυτοκινούμενα οχήματα**.

Τα αυτόνομα αυτοκινούμενα οχήματα έχουν αρχίσει να έχουν μεγάλη διάδοση τα τελευταία χρόνια. Συναντώνται σε ερευνητικές αποστολές στο διάστημα (*Pathfinder*), σε πεδία στρατιωτικών επιχειρήσεων κυρίως ως κατασκοπευτικά μέσα ή ανιχνευτές ναρκών και επικίνδυνων αερίων, σε επιχειρήσεις διάσωσης (π.χ. σε πυρκαγιές, σε ναυάγια κι αλλού) αλλά και ως έξυπνα παιχνίδια.

### 2.2 Δομή και λειτουργία

Το σύστημα που συνήθως αποτελεί ένα αυτόνομο αυτοκινούμενο όχημα είναι πολύπλοκο. Αποτελείται από μηχανικά, ηλεκτρικά και ηλεκτρονικά μέρη, καθώς επίσης και από αισθητήρες διαφόρων ειδών. Οι αισθητήρες χρησιμοποιούνται για την ανίχνευση του περιβάλλοντος χώρου και την παρακολούθηση των παραμέτρων κίνησης του οχήματος, δηλαδή της ταχύτητας, της επιτάχυνσης, της γωνίας στροφής κ.λ.π. Τα ηλεκτρονικά κυκλώματα του οχήματος χρησιμοποιούνται για την επεξεργασία των σημάτων που δίνουν οι αισθητήρες, για τον προσδιορισμό της θέσης του και της απόστασής του από γειτονικά εμπόδια, τη διόρθωση της πορείας του και τις επικοινωνίες. Επειδή οι συνθήκες που συναντά το όχημα κατά τη διάρκεια της κίνησής του σε ένα πραγματικό περιβάλλον είναι συνεχώς μεταβαλλόμενες επιβάλλεται η μονάδα ελέγχου θέσης του να διαθέτει αρκετή λογική που θα του εξασφαλίσει μια ομαλή κι ασφαλή πορεία προς το τελικό σημείο στόχο.

Η λειτουργία των αυτοκινούμενων οχημάτων διαφέρει ανάλογα με την κατηγορία τους. Στα μη ευφυή οχήματα, αισθητήρες προσδιορίζουν τη θέση του οχήματος κατά τη διάρκεια της κίνησής του κι εντοπίζουν τυχόν εμπόδια. Με τη βοήθεια μιας μονάδας ελέγχου το όχημα διαγράφει προσχεδιασμένες πορείες βάσει χρονικού διαγράμματος, δηλαδή πορείες τις οποίες έχει σχεδιάσει ο χρήστης πριν την έναρξη της κίνησης του οχήματος. Αυτή η κατηγορία οχημάτων χαρακτηρίζεται από ανελαστικότητα διότι μόλις εντοπίσουν κάποιο άγνωστο από πριν εμπόδιο ακινητοποιούνται. Δεν μπορούν να το παρακάμψουν παρά μόνο με τη βοήθεια του χρήστη. Τα ευφυή οχήματα διαφέρουν από τα μη ευφυή στη λειτουργία τους σε αυτό το σημείο. Σε αντίθεση με τα μη ευφυή, όταν συναντήσουν κάποιο εμπόδιο μπορούν να το παρακάμψουν. Τα ευφυή οχήματα περιλαμβάνουν λογική μονάδα η οποία ελέγχει συνεχώς τη θέση τους κι αν εντοπιστεί παρέκκλιση από την αρχικά σχεδιασμένη πορεία ακολουθείται διαδικασία διόρθωσής της.

Οι πιο διαδεδομένες διατάξεις που χρησιμοποιούνται για την αντίληψη του περιβάλλοντος και την καθοδήγηση του οχήματος είναι το **sonar** (SOund Navigation And Ranging) που λειτουργεί με υπέρηχους, το **lidar** (LIght Detection And Ranging) που λειτουργεί με υπέρυθρες ακτίνες, διατάξεις λήψης εικόνων (συστήματα με **κάμερες**) και το σύστημα πλοήγησης με δορυφόρους **GPS** (Global Positioning System).

### 2.3 Προβλήματα

Τα αυτοκινούμενα οχήματα αντιμετωπίζουν διάφορα προβλήματα κατά τη διάρκεια της κίνησης τους. Αυτά μπορεί να οφείλονται είτε στα μηχανικά τους μέρη, είτε στα ηλεκτρικά, είτε στα ηλεκτρονικά. Τα λάθη που οφείλονται στα μηχανικά μέρη του οχήματος προκαλούνται από ροπές που αναπτύσσονται κατά την κίνησή του και από ανοχές των μερών αυτών. Αυτά έχουν σαν αποτέλεσμα την παρέκκλιση από την πορεία που πρέπει να ακολουθήσει το όχημα. Τα κυκλώματα τροφοδοσίας του οχήματος εισάγουν σφάλματα στην πορεία του εξαιτίας της μη σταθερής παρεχόμενης τάσης στα ηλεκτρονικά κυκλώματα και στους αισθητήρες. Η αστάθεια στην τροφοδοσία των ηλεκτρονικών μερών του οχήματος μπορεί να προκαλέσει σφάλματα στον προσδιορισμό της θέσης του οχήματος, καθώς επίσης και στους υπολογισμούς που κάνει η μονάδα ελέγχου για την απόσταση από εμπόδια, την

ταχύτητα και άλλων παραμέτρων, των οποίων ο καθορισμός είναι απαραίτητος για να υλοποιηθεί η διαδρομή σωστά και με ασφάλεια για το όχημα.

### **3. ΥΛΙΚΟ**

#### **3.1 Πάτωμα**

Το όχημα που χρησιμοποιείται για την εφαρμογή αυτή είναι ένα πλαστικό τηλεχειριζόμενο μοντέλο της εταιρίας MINICRAFT MODELS INC, το ROAD RUNNER II σε κλίμακα 1/10.

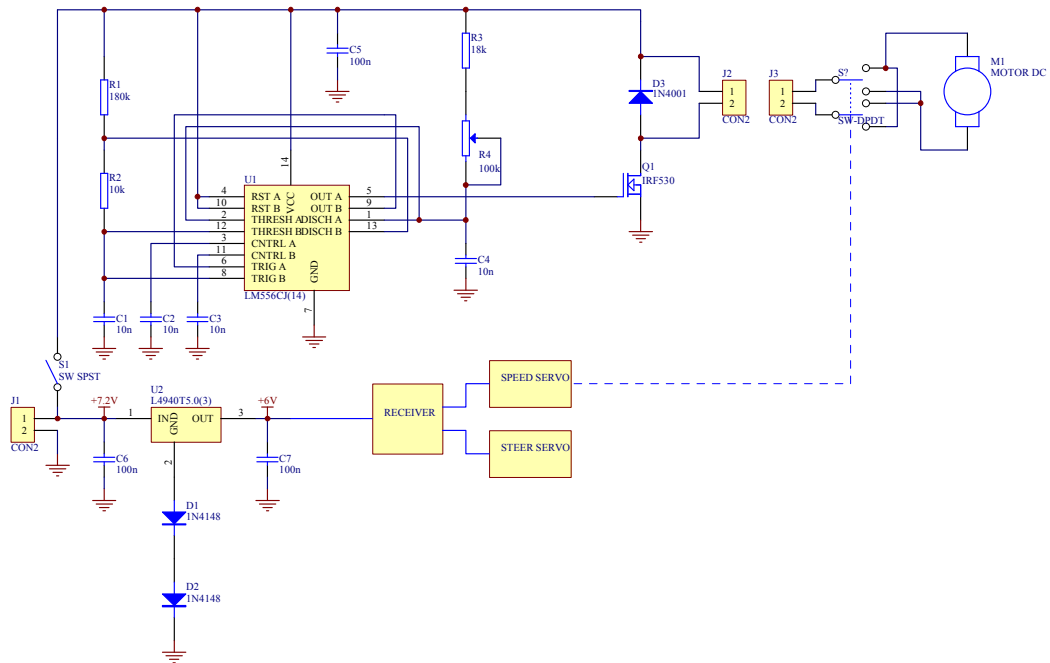
#### **3.2 Κινητήρας**

Ο κινητήρας του οχήματος είναι ηλεκτρικός (DC 7,2V). Τροφοδοτείται από μια συστοιχία από 6 επαναφορτιζόμενες μπαταρίες Ni-Cd, 1600 mAh. Η κίνηση από τη μηχανή μεταδίδεται στους πίσω τροχούς. Το μοντέλο αυτό έχει ανεξάρτητη ανάρτηση σε κάθε τροχό.

#### **3.3 Έλεγχος διεύθυνσης και ταχύτητας**

Στο όχημα υπάρχουν δύο σερβομηχανισμοί με τη χρήση των οποίων καθορίζονται η ταχύτητα και η γωνία στροφής του οχήματος. Οι σερβομηχανισμοί οδηγούνται από ένα ραδιοδέκτη που είναι τοποθετημένος στο όχημα. Ο σερβομηχανισμός ελέγχου της γωνίας στροφής μπορεί να στρέψει το σύστημα διεύθυνσης του οχήματος έως 30° δεξιά και 30° αριστερά. Ο σερβομηχανισμός ελέγχου της ταχύτητας, καθορίζει την τιμή της ταχύτητας αλλάζοντας τη θέση ενός επιλογέα από τον οποίο επιλέγεται η τιμή της αντίστασης  $R_s$  που θα παρεμβληθεί σε σειρά με τον κινητήρα. Η ταχύτητα κίνησης του οχήματος μπορεί να λάβει τρεις διαφορετικές τιμές: την ελάχιστη ( $R_s=0.3\Omega$ ), τη μέση ( $R_s=2 \times 0.3\Omega=0.6\Omega$ ), και τη μέγιστη ( $R_s=0\Omega$ ).

Χρησιμοποιώντας όμως αντιστάτες για την επίτευξη χαμηλής ταχύτητας κίνησης προκαλείται άσκοπη κατανάλωση ενέργειας. Επειδή στο πλαίσιο αυτής της εργασίας το όχημα θα πρέπει να κινείται με πολύ μικρή ταχύτητα, το κύκλωμα ρύθμισής της με αντιστάτες αντικαταστάθηκε από έναν ρυθμιστή στροφών DC κινητήρα του οποίου η λειτουργία που στηρίζεται στη διαμόρφωση εύρους παλμών (PWM, Pulse Width Modulation), Σχ 7.



Σχ.7

Από τον ροοστάτη R4 μεταβάλλεται το εύρος των παλμών οδήγησης του MOSFET Q1, το οποίο με τη σειρά του καθορίζει την ταχύτητα του οχήματος.

### 3.4 Τηλεκατεύθυνση

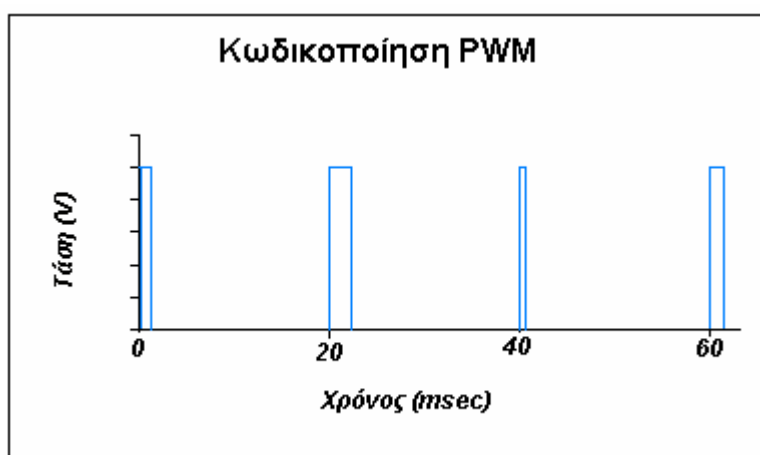
Το σύστημα τηλεκατεύθυνσης του οχήματος είναι το LYNX, 2 καναλιών FM. Τα βασικά του χαρακτηριστικά είναι τα παρακάτω:

- Τάση λειτουργίας: 12VDC
- Ρεύμα κατανάλωσης: 180mA
- Ισχύς εξόδου RF: 500 mW
- Δείκτης κατάστασης μπαταριών: NAI (με 3 LED)
- Συχνότητα λειτουργίας: 48MHz
- Κανάλια τηλεχειρισμού: 2 αναλογικά (ένα για την ταχύτητα κι ένα για τη διεύθυνση).
- Διακόπτες αντιστροφής των σέρβο: NAI (και στα δυο κανάλια).
- Ρύθμισης μηδενός: NAI (και για τα δύο κανάλια).

Το σύστημα της τηλεκατεύθυνσης λειτουργεί με έναν πομπό και έναν δέκτη οι οποίοι συντονίζονται να λειτουργούν στην ίδια συχνότητα χρησιμοποιώντας κατάλληλο ζεύγος όμοιων κρυστάλλων (48MHz).

Στον πομπό υπάρχουν δύο αναλογικά ποτενσιόμετρα. Ένα για τον καθορισμό της ταχύτητας κι ένα για τον καθορισμό και της γωνίας στροφής του οχήματος. Το ποτενσιόμετρο καθορισμού της γωνίας στροφής καθορίζει (ανάλογα με τις τιμές που παίρνει) προς ποια κατεύθυνση θα στρίψει το όχημα (δεξιά ή αριστερά) και πόσο.

Οι σερβομηχανισμοί οι οποίοι ρυθμίζουν, ο ένας την ταχύτητα και ο άλλος τη γωνία στροφής ελέγχονται μέσω του συστήματος πομπού-δέκτη. Στέλνονται σήματα ελέγχου με κωδικοποίηση PWM (Pulse Width Modulation) όπως φαίνεται παρακάτω στο σχήμα 8.



Σχ. 8

Η περίοδος επανάληψης των παλμών ελέγχου των σερβομηχανισμών είναι 20 msec. Το εύρος των παλμών κυμαίνεται από το 1 msec έως τα 2 msec (σε ορισμένους σερβομηχανισμούς η διάρκεια των παλμών ελέγχου μεταβάλλεται από 0.7 msec έως 2.5 msec). Οι ελεγκτές των τηλεκατευθυνόμενων μοντέλων λειτουργούν με δύο τρόπους. Στους αναλογικούς ελεγκτές επιτρέπεται η χρήση όλων των ενδιάμεσων τιμών για τη διάρκεια των παλμών ελέγχου των σερβομηχανισμών. Οι μη αναλογικοί ή δυαδικοί ελεγκτές μετάγονται από τη μια ακραία κατάσταση στην άλλη, δηλαδή χρησιμοποιούν παλμούς με διάρκειες 1 και 2 msec μόνο. Ο σερβομηχανισμός περιέχει ένα μικρό ηλεκτρικό κινητήρα που οδηγεί την ελεγχόμενη διάταξη. Η

περιστροφή του κινητήρα μεταβάλλει τη θέση ενός ενσωματωμένου ποτενσιόμετρου. Το ποτενσιόμετρο, που έχει αντίσταση περίπου 5 kΩ, ελέγχει ένα μονοσταθί πολυδονητή που επίσης είναι ενσωματωμένος στο σερβομηχανισμό. Γενικά, τα ενσωματωμένα ηλεκτρονικά κρατούν τη διάρκεια του παλμού που παράγεται εσωτερικά, ίση με αυτήν του παλμού εισόδου. Επειδή υπάρχει μια καθορισμένη σχέση μεταξύ της γωνίας περιστροφής του ποτενσιόμετρου και της διάρκειας του παλμού, η περιστροφή του σερβοκινητήρα μπορεί να ελεγχθεί απευθείας από τη διάρκεια του παλμού ελέγχου.

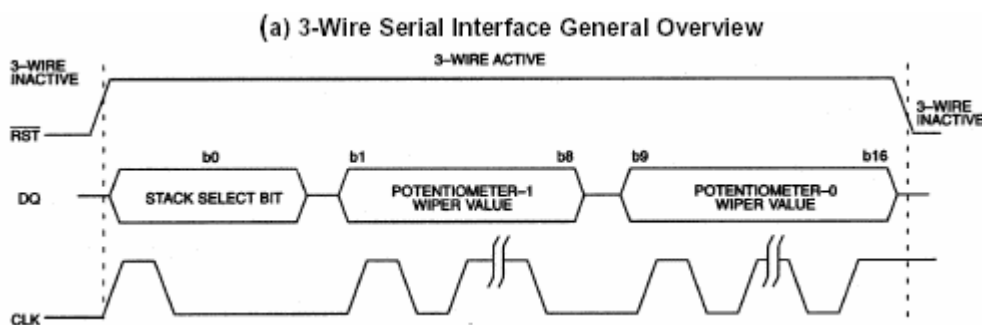
### 3.5 Επικοινωνία με τη σειριακή θύρα (RS232)

Για να επιτευχθεί ο έλεγχος της κίνησης του οχήματος από τον υπολογιστή αντικαταστάθηκαν τα αναλογικά ποτενσιόμετρα που βρίσκονταν στην τηλεκατεύθυνση και έλεγχαν την ταχύτητα και τη γωνία στροφής, με ένα διπλό ψηφιακό ποτενσιόμετρο. Το ποτενσιόμετρο που χρησιμοποιήθηκε είναι το DS1867 (DIP-14) της MAXIM. Το ποτενσιόμετρο αυτό δέχεται κωδικές λέξεις των 17 bits. Στην εφαρμογή αυτή χρησιμοποιούνται και τα δύο ενσωματωμένα στο ολοκληρωμένο ποτενσιόμετρα, με εύρος τιμών 0-10 kΩ, το ένα για τον έλεγχο της ταχύτητας και το άλλο για τον έλεγχο της γωνίας στροφής. Τα bits 2-9 της κωδικής λέξης προγραμματίζουν το ένα ποτενσιόμετρο (στην εφαρμογή αυτό που ελέγχει την ταχύτητα) και τα bits 10-17 προγραμματίζουν το άλλο ποτενσιόμετρο (στην εφαρμογή αυτό που ελέγχει τη γωνία στροφής). Το πρώτο bit χρησιμοποιείται μόνο όταν το ολοκληρωμένο συνδεθεί για να λειτουργήσει σαν μονό ψηφιακό ποτενσιόμετρο με διπλάσια ονομαστική τιμή.

Στο πρόγραμμα, για να κατέβουν οι κωδικές λέξεις στη σειριακή πόρτα, χρησιμοποιείται το ελεγκτήριο MSCComm της VISUAL BASIC για τη σειριακή. Προτιμήθηκε αυτός ο τρόπος ελέγχου των σημάτων της σειριακής γιατί είναι άμεσος και εύκολος. Δεν χρειάζονται άλλα εξαρτήματα, εκτός από κάποιες διόδους ζένερ και αντιστάτες για τον περιορισμό των τάσεων της σειριακής σε επίπεδα TTL. Το ελεγκτήριο κατεβάζει τα σήματα προγραμματισμού των ποτενσιόμετρων στη σειριακή αμέσως μόλις παραχθούν από το πρόγραμμα. Τέλος με αυτόν τον τρόπο η λειτουργία της εφαρμογής παραμένει η ίδια σε όλα τα περιβάλλοντα που τρέχει (λειτουργεί με τον ίδιο τρόπο σε όλες τις εκδόσεις των Windows).

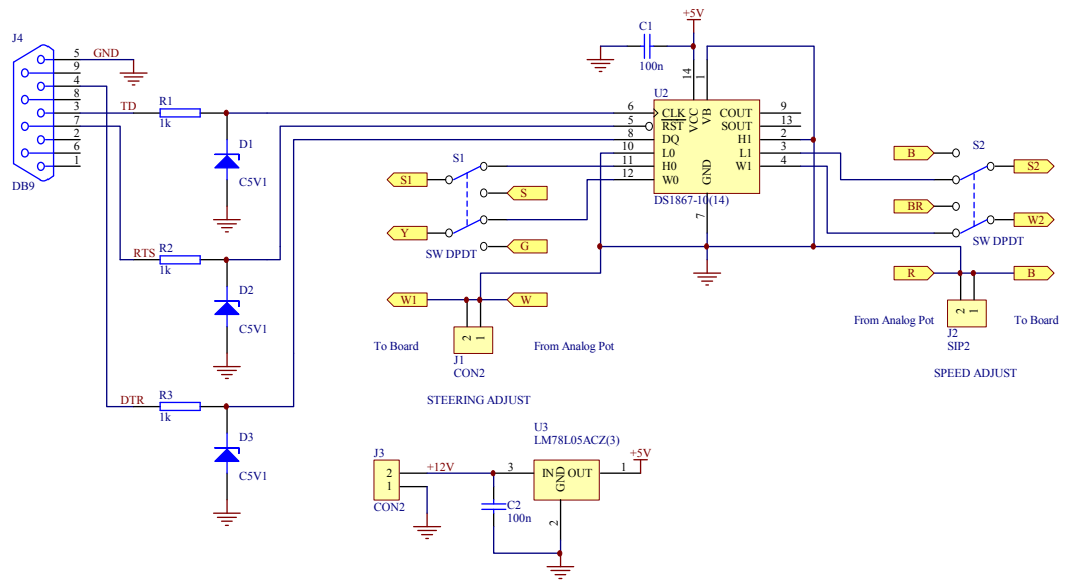


Τα σήματα της σειριακής που ελέγχονται από το ελεγκτήριο MScComm είναι τα RTS (Request To Send), DTR (Data Terminal Ready) και TXD (Transmit Data). Όπως φαίνεται και στο παρακάτω σχήμα, χρησιμοποιούνται τρεις γραμμές για τον προγραμματισμό του ποτενσιόμετρου. Η Reset ενεργοποιεί τη γραμμή DQ. Η ενεργοποίηση γίνεται με την άνοδο του παλμού reset και η απενεργοποίηση με την πτώση του. Η DQ μεταφέρει τα bit που θα κωδικοποιήσουν τα ποτενσιόμετρα και η clock καθορίζει πότε θα περάσει το επόμενο bit στη γραμμή DQ. Με την κάθε άνοδο του παλμού clock περνάει το επόμενο bit στη γραμμή DQ.



Σχ. 9

Το ελεγκτήριο MScComm μεταβάλλει τα σήματα που ελέγχει από λογικό “1,, σε λογικό “0,, και ανάποδα. Οι ιδιότητες DTREnable, RTSEnable και Break του ελεγκτηρίου κάνουν τον έλεγχο στα σήματα DTR, RTS και TXD αντίστοιχα. Το DTR χρησιμοποιείται για να μεταφέρει τα δεδομένα στο ολοκληρωμένο μέσο της εισόδου DQ. Το TXD χρησιμοποιείται σαν σήμα ρολογιού clock μια και η διάρκεια διατήρησης του σε λογικό “1,, είναι μικρή. Τέλος το RTS εφαρμόζεται στην είσοδο Reset.



Σχ. 10

## 4. ΛΟΓΙΣΜΙΚΟ

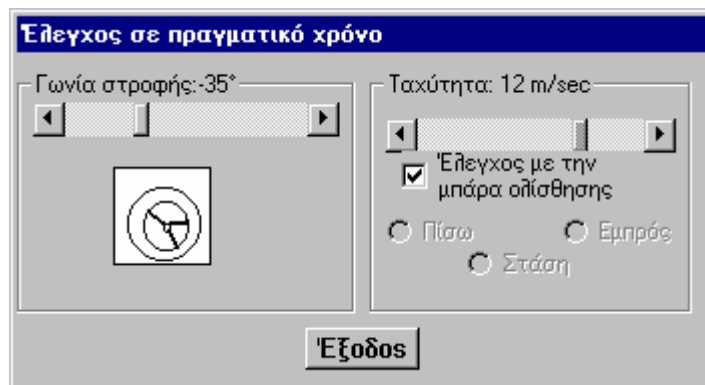
### 4.1 Εισαγωγή

Για την υλοποίηση αυτής της εφαρμογής χρησιμοποιήθηκε η γλώσσα προγραμματισμού VISUAL BASIC 6.0. Υλοποιήθηκαν δύο προγράμματα. Στο πρώτο ο έλεγχος της κίνησης του οχήματος γίνεται σε πραγματικό χρόνο κατά τη διάρκεια της κίνησης του. Στο δεύτερο σχεδιάζεται μια διαδρομή με ελεύθερο τρόπο ή με μικρά ευθύγραμμα τμήματα, η οποία ακολουθείται στη συνέχεια από το όχημα.

### 4.2 Έλεγχος σε πραγματικό χρόνο

#### 4.2.1 Παρουσίαση φόρμας ελέγχου

Σε αυτό το πρόγραμμα ο έλεγχος γίνεται σε πραγματικό χρόνο, δηλαδή κατά τη διάρκεια της κίνησης του οχήματος. Ο έλεγχος γίνεται σε δύο παραμέτρους του. Στην ταχύτητα και την γωνία στροφής του. Για αυτό το λόγο χρησιμοποιούνται δύο μπάρες ολίσθησης. Μία για τον έλεγχο της ταχύτητας και μία για τον έλεγχο της γωνίας στροφής. Για τον ευκολότερο έλεγχο του οχήματος έχει προστεθεί η δυνατότητα κίνησης με σταθερή ταχύτητα, είτε μπροστά, είτε πίσω.



Σχ.11

Στο παραπάνω σχήμα φαίνεται η φόρμα του προγράμματος. Στα αριστερά βρίσκεται η μπάρα ελέγχου της γωνίας στροφής και στα δεξιά της ταχύτητας. Ανάλογα με την τιμή της αντίστοιχης μπάρας μεταβάλλεται η ελεγχόμενη παράμετρος.

Το πλαίσιο ελέγχου στα δεξιά μεταφέρει τον έλεγχο της ταχύτητας από την μπάρα στα κουμπιά επιλογής. Με τα κουμπιά επιλογής το όχημα προχωράει μπροστά

ή πίσω με σταθερή και μικρή ταχύτητα. Αυτή η επιλογή μας επιτρέπει να επικεντρωθούμε κατά την εκτέλεση της εφαρμογής κυρίως στον έλεγχο της γωνίας στροφής.

#### 4.2.2 Τρόπος λειτουργίας

Ανάλογα με την τιμή που θα έχει η μπάρα ολίσθησης μπορεί να αυξηθεί ή να μειωθεί η ελεγχόμενη παράμετρος. Αυτό γίνεται με τον παρακάτω τρόπο:

Χρησιμοποιείται μια μεταβλητή στην οποία καταχωρείται η τιμή της μπάρας ολίσθησης και πιο συγκεκριμένα η μεταβλητή **vbspeed** για την ταχύτητα και η μεταβλητή **vturn** για τη γωνία στροφής. Κατόπιν χρησιμοποιώντας τη σχέση **freq = vbspeed \* 256 + vturn** εξάγουμε έναν δεκαδικό που τον καταχωρούμε στη βοηθητική μεταβλητή **freq**. Ο αριθμός αυτός θα κατέβει στη σειριακή για να προγραμματίσει τα ηλεκτρονικά ποτενσιόμετρα και θα πρέπει να είναι σε δυαδική μορφή. Για να κατέβουν οι σωστοί αριθμοί από τις μπάρες στα ποτενσιόμετρα είναι απαραίτητη η ολίσθηση του ενός αριθμού στη δυαδική του μορφή κατά οκτώ θέσεις, η οποία επιτυγχάνεται με τον παραπάνω τρόπο. Τα ποτενσιόμετρα προγραμματίζονται με «κωδικές» λέξεις μήκους 17 bit. Με αυτό τον τρόπο ο αριθμός που ελέγχει την ταχύτητα (στη δυαδική μορφή) θα μεταβεί στο αντίστοιχο ποτενσιόμετρο χωρίς να αλλοιωθεί στα 8 πρώτα bit της κωδικής λέξης. Η βοηθητική μεταβλητή **freq** χρησιμοποιείται στον παρακάτω βρόγχο, ο οποίος κάνει τη μετατροπή του αριθμού από δεκαδικό σε δυαδικό:

```
MSComm1.RTSEnable = True
```

```
Do While i >= 0
```

```
X = freq / 2 ^ i
```

```
If X >= 1 Then
```

```
MSComm1.DTREnable = 1
```

```
MSComm1.Break = True
```

```
MSComm1.DTREnable = 0
```

```
MSComm1.Break = False
```

```
freq = freq - 2 ^ i
```

```
Else
```

**MSComm1.Break = True**

**MSComm1.Break = False**

**End If**

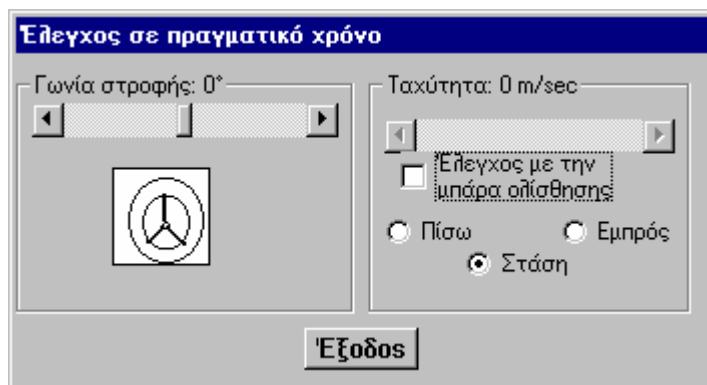
**i = i - 1**

**Loop**

**MSComm1.RTSEnable = False**

Σε αυτή τη διαδικασία διαιρείται η μεταβλητή **freq** με πολλαπλάσια του 2 και το πηλίκο συγκρίνεται με τη μονάδα. Αν το πηλίκο είναι μικρότερο από τη μονάδα τότε στη σειριακή «κατεβαίνει» σήμα που αντιστοιχεί σε bit 0, αλλιώς «κατεβαίνει» σήμα που αντιστοιχεί σε bit 1. Η κωδική λέξη που σχηματίζεται στην έξοδο από το βρόχο έχει μήκος 17 bit. Για τον καθορισμό των επαναλήψεων στο βρόχο έχει τοποθετηθεί ένας μετρητής (η μεταβλητή **i**) ο οποίος μειώνεται σε κάθε επανάληψη. Μόλις μηδενιστεί τερματίζεται ο βρόχος.

Για να μεταβεί ο έλεγχος της ταχύτητας από τη μπάρα ολίσθησης στα κουμπιά επιλογής πρέπει να γίνει κλικ στο πλαίσιο ελέγχου “**Έλεγχος με την μπάρα ολίσθησης**...”. Τότε απενεργοποιείται η λειτουργία της μπάρας και ενεργοποιούνται οι λειτουργίες των κουμπιών επιλογής. Για την αντίστροφη διαδικασία γίνεται και πάλι κλικ στο πλαίσιο ελέγχου.



**Σχ.12**

Στα κουμπιά επιλογής – ανάλογα με την επιλογή, δηλαδή Πίσω, Στάση ή Εμπρός – καταχωρείται μια τιμή στη μεταβλητή **vbspeed** και μετατρέπεται με τον τρόπο που αναφέρεται παραπάνω σε δυαδική για να «κατεβεί» στη σειριακή. Η τιμή

αυτή είναι κατάλληλη για να κινηθεί το όχημα πίσω ή μπροστά με σταθερή ταχύτητα ή να σταματήσει.

### 4.3 Σχεδίαση διαδρομής

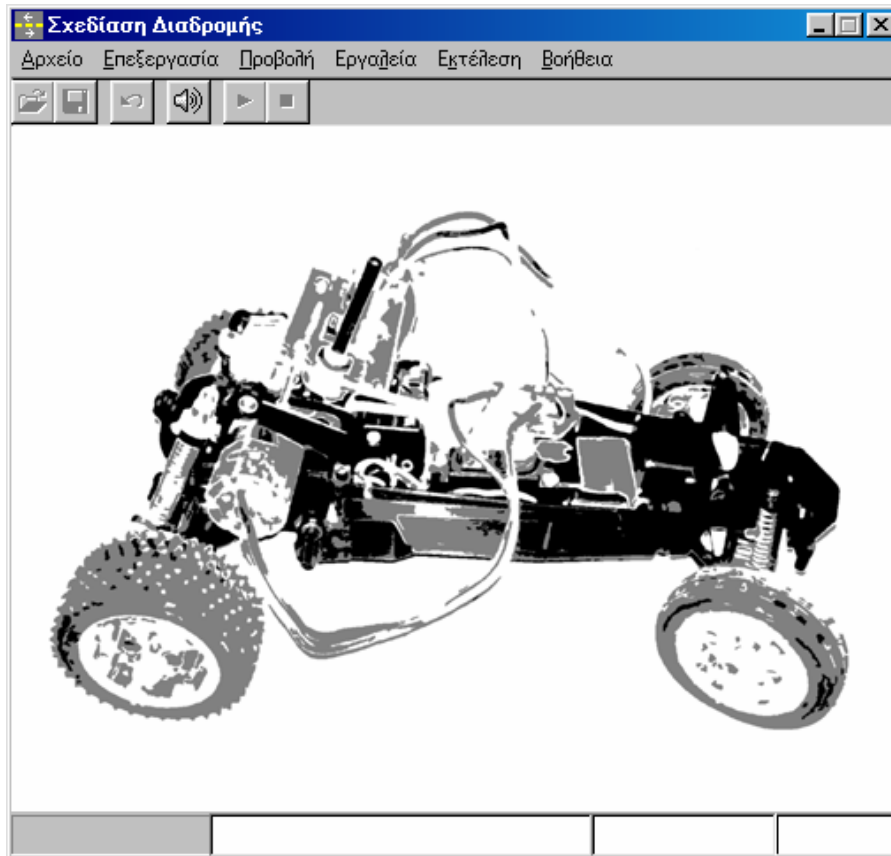
#### 4.3.1 Εισαγωγή

Σε αυτό το πρόγραμμα ο έλεγχος στην κίνηση του οχήματος, δεν γίνεται σε πραγματικό χρόνο. Αφού καθοριστεί η διαδρομή την οποία θα ακολουθήσει, ξεκινάει ο έλεγχος στην ταχύτητα και τη γωνία στροφής του οχήματος για την υλοποίηση της διαδρομής αυτής. Υπάρχει η δυνατότητα αποθήκευσης κάποιας διαδρομής για να χρησιμοποιηθεί κάποια άλλη στιγμή χωρίς να χρειαστεί να την σχεδιάσουμε ξανά από την αρχή.

Σε αυτό το πρόγραμμα ο έλεγχος περιορίζεται στη γωνία στροφής αφού έχει επιλεγεί μια μικρή τιμή για την ταχύτητα για όλη τη διάρκεια της κίνησης του οχήματος. Αυτή η επιλογή έγινε για την ομαλότερη κίνηση του οχήματος αφού το μοντέλο που χρησιμοποιείται στην εφαρμογή δεν παρέχει ακρίβεια στη μεταβολή της ταχύτητας του.

#### 4.3.2 Εισαγωγή χάρτη-οδηγού

Το πρόγραμμα δεν επιτρέπει τη σχεδίαση διαδρομής πριν από την εισαγωγή ενός χάρτη-οδηγού. Θα πρέπει να επιλεγεί κάποιος χάρτης σαν βάση για να μπορεί ο χρήστης να σχεδιάσει την επιθυμητή διαδρομή. Η εισαγωγή του χάρτη γίνεται ακολουθώντας τη διαδρομή **Αρχείο→Προσθήκη Χάρτη**. Στο σχήμα που ακολουθεί φαίνεται η κύρια φόρμα πριν τη «φόρτωση» κάποιου χάρτη.



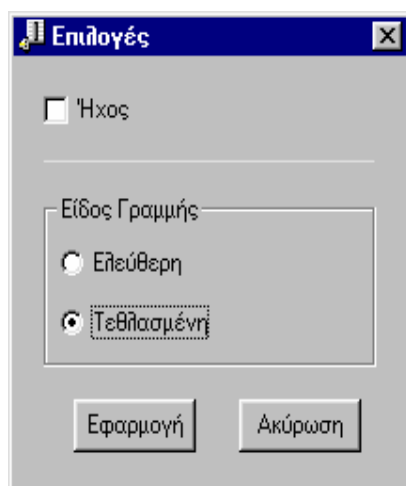
Σχ.13

Αφού επιλεγεί κάποιος χάρτης που έχει αποθηκευτεί ως αρχείο εικόνας του τύπου .bmp, .gif ή .jpeg είναι έτοιμη η φόρμα για τη σχεδίαση. Για τη διαδικασία σχεδίασης πρέπει να ρυθμιστούν κάποιες παράμετροι. Για αυτό έχουν προστεθεί δυο ακόμα φόρμες. Είναι η φόρμα «Επιλογές» και η φόρμα «Ρυθμίσεις».

#### 4.3.3 Ρύθμιση παραμέτρων σχεδίασης

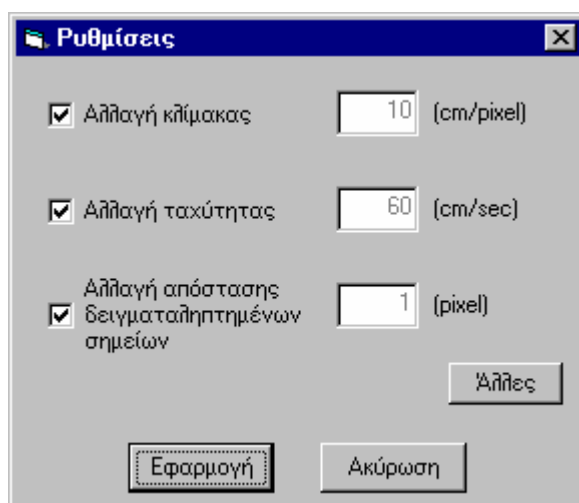
Στη φόρμα «Επιλογές» επιλέγεται αν είναι επιθυμητό να ακούγεται ήχος χαρακτηριστικά κατά τη διάρκεια κίνησης του οχήματος. Επίσης επιλέγεται το είδος της γραμμής σχεδίασης της διαδρομής. Με την επιλογή «Ελεύθερη» σχεδιάζεται η διαδρομή με ελεύθερη και καμπύλη γραμμή, ενώ με την επιλογή «Τεθλασμένη» σχεδιάζεται η διαδρομή με ευθύγραμμα τμήματα. Ανάλογα με την επιλογή του είδους της γραμμής σχεδίασης ενεργοποιείται ή απενεργοποιείται μια σημαία (**DrawFlag**). Η σημαία ενεργοποιεί τις αντίστοιχες διαδικασίες, για το κάθε είδος σχεδίασης, οι οποίες χρησιμοποιούνται για το σχεδιασμό της διαδρομής και την επιλογή των σημείων της διαδρομής, που θα χρησιμοποιούνται στον υπολογισμό των αποστάσεων που θα καλύψει το όχημα. Με την επιλογή του πλαισίου ελέγχου για την

ενεργοποίηση του ήχου ή την απενεργοποίηση του, ρυθμίζεται μια σημαία (**SoundFlag**) η οποία επιτρέπει ή απαγορεύει την αποστολή ενός σήματος στο μεγάφωνο του υπολογιστή για την παραγωγή ή όχι ήχου. Η ρύθμιση του ήχου μπορεί να γίνει και κατά τη διάρκεια εκτέλεσης της διαδρομής.



Σχ.14

Στη φόρμα «Ρυθμίσεις», καθορίζονται η τιμή που θα έχει η ταχύτητα του οχήματος, η τιμή της κλίμακας του χάρτη καθώς επίσης και η απόσταση των επιλεγμένων σημείων της σχεδιασμένης διαδρομής, τα οποία χρησιμοποιούνται για τον υπολογισμό των αποστάσεων που θα καλύψει το όχημα. Η παράμετρος της απόστασης χρησιμοποιείται μόνο στις διαδικασίες της σχεδίασης με ελεύθερο τρόπο. Μόλις φορτωθεί ο χάρτης θα πρέπει ο χρήστης να ενημερώσει την κλίμακα στη φόρμα.



Σχ.15



Κάνοντας κλικ στο πλήκτρο **Άλλες** μεταβαίνουμε σε μια φόρμα ρυθμίσεων στην οποία δηλώνεται η ταχύτητα που θα έχει το όχημα κατά τη διάρκεια της στροφής του, καθώς επίσης και οι τιμές – σε δεκαδική μορφή – που θα «κατεβαίνουν» στα ποτενσιόμετρα κατά τη διάρκεια της κίνησης του οχήματος για να καθορίσουν τη θέση που θα έχουν για τη δεξιά στροφή και την αριστερή στροφή και για την ταχύτητα του οχήματος στην περίπτωση στάσης, κίνησης εμπρός ή κίνηση πίσω. Στο παρακάτω σχήμα φαίνεται αυτή η φόρμα. Η πρώτη σειρά αφορά την τιμή του ποτενσιόμετρου για τη γωνία. Η δεύτερη την τιμή του ποτενσιόμετρου για την ταχύτητα και η τρίτη για την ταχύτητα στροφής του οχήματος. Κάνοντας κλικ σε κάποιο από τα πλαίσια ελέγχου, μπορεί ο χρήστης να αλλάξει τις αρχικές ρυθμίσεις των παραμέτρων που περιλαμβάνονται στη φόρμα.

<input checked="" type="checkbox"/> Δεξιά	<input checked="" type="checkbox"/> Ευθεία	<input checked="" type="checkbox"/> Αριστερά
122	97	72
<input checked="" type="checkbox"/> Πίσω	<input checked="" type="checkbox"/> Στάση	<input checked="" type="checkbox"/> Εμπρός
112	126	140
<input checked="" type="checkbox"/> Αλλαγή ταχύτητας στροφής	70 (cm/sec)	
<input type="button" value="Εφαρμογή"/>		<input type="button" value="Ακύρωση"/>

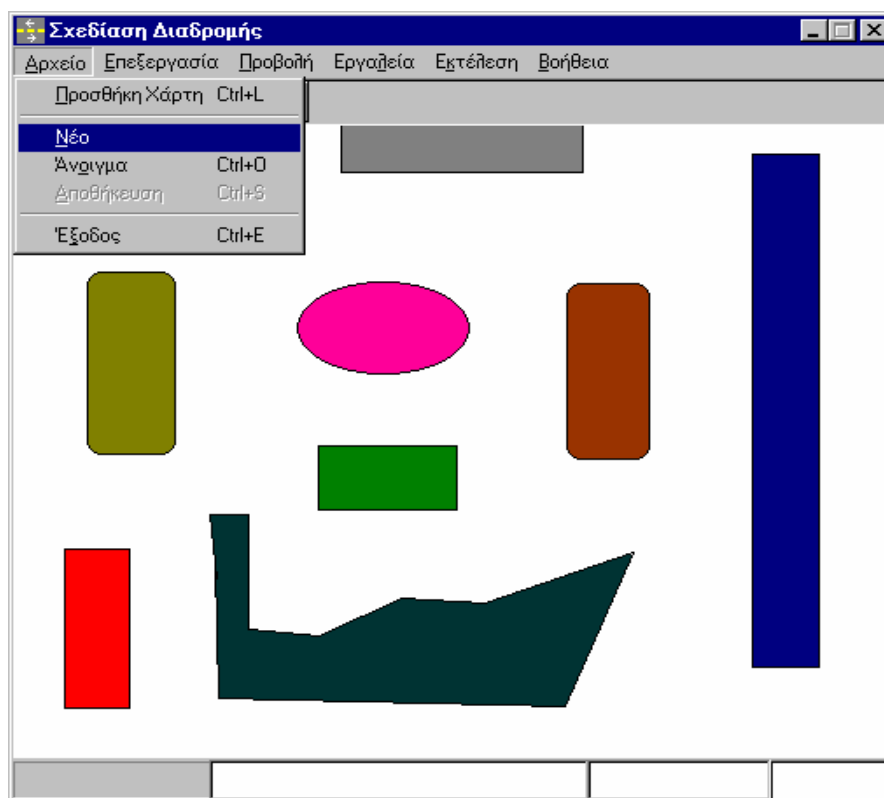
Σχ.16

Αφού καθοριστεί το είδος της γραμμής σχεδίασης, η κλίμακα του χάρτη και οι λοιπές μεταβλητές μπορεί ο χρήστης να σχεδιάσει μια νέα διαδρομή ακολουθώντας τη διαδρομή **Αρχείο**→**Νέο** όπως φαίνεται και στο σχήμα 17.

#### 4.3.4 Διαδικασία προσανατολισμού

Αμέσως μετά την επιλογή για σχεδίαση νέας διαδρομής αρχίζει μια διαδικασία για τη δήλωση του προσανατολισμού του οχήματος στο χάρτη. Ο προσανατολισμός του οχήματος γίνεται, προσαρμόζοντας πάνω του μια πυξίδα η οποία δείχνει τη γωνία που σχηματίζει ο άξονας του οχήματος με το βορά. Πρώτα ορίζεται η διεύθυνση του βορά στο χάρτη. Σχεδιάζουμε μια γραμμή η οποία θα είναι

παράλληλη προς το βορά του χάρτη-οδηγού. Η τιμή της γωνίας που σχηματίζει η γραμμή αυτή ως προς τον οριζόντιο άξονα της φόρμας υπολογίζεται από τις συντεταγμένες των άκρων της. Κατόπιν η γωνία αυτή καταχωρείται σε μια μεταβλητή με το όνομα **NorthAngle**. Κατόπιν σημειώνεται το σημείο εκκίνησης του οχήματος. Μετά σχεδιάζεται η διεύθυνση του άξονα του οχήματος ως προς το βορά. Αυτό γίνεται σχεδιάζοντας μια γραμμή με αρχικό σημείο το σημείο εκκίνησης και φέρνοντάς την παράλληλη στον άξονα του οχήματος. Από τις συντεταγμένες των άκρων της γραμμής αυτής υπολογίζεται η γωνία που σχηματίζει ο άξονας του οχήματος με τον οριζόντιο άξονα της φόρμας. Αυτή η γωνία καταχωρείται στη μεταβλητή **CarAngle**. Από τις δύο αυτές γωνίες (**NorthAngle** και **CarAngle**) το πρόγραμμα εξάγει τη γωνία, η οποία ταυτίζεται με την ένδειξη της πυξίδας που είναι προσαρμοσμένη πάνω στο όχημα.



Σχ.17

Η διαδικασία για την εξαγωγή της γωνίας αυτής, συσχετίζει τις **NorthAngle** και **CarAngle**, καθώς επίσης τις συντεταγμένες των σημείων από τα οποία υπολογίστηκαν αυτές οι δύο γωνίες. Η διαδικασία περιγράφεται παρακάτω.

Ανάλογα με το πρόσημο της διαφοράς των συντεταγμένων του τελικού από το αρχικό σημείο, καθώς επίσης και από τη σχέση που έχουν οι γωνίες **NorthAngle** και **CarAngle** συγκρινόμενες μεταξύ τους χρησιμοποιείται η ανάλογη μαθηματική σχέση για να προκύψει η γωνία που θα ταυτιστεί με την ένδειξη της πυξίδας. Παρακάτω παρατίθεται η διαδικασία που μόλις περιγράφηκε:

**If CarAngle > 0 And NorthAngle < 0 Then**

**If (Car(1, 1) >= DataX(1) And NorthPoint(1, 3) < NorthPoint(1, 1)) Or (Car(1, 1) < DataX(1) And NorthPoint(1, 3) >= NorthPoint(1, 1)) Then**

**nc = Int(((CarAngle - NorthAngle) - 180) + 0.5)**

**If 0 > nc And nc >= -180 Then**

**nc = 360 + nc**

**End If**

**Text1.Text = "Δφπιξ(°)=" + Str(nc)**

**Elseif (Car(1, 1) >= DataX(1) And NorthPoint(1, 3) >= NorthPoint(1, 1)) Or (Car(1, 1) < DataX(1) And NorthPoint(1, 3) < NorthPoint(1, 1)) Then**

**nc = Int((CarAngle - NorthAngle) + 0.5)**

**If 0 > nc And nc >= -180 Then**

**nc = 360 + nc**

**End If**

**Text1.Text = "Δφπιξ(°)=" + Str(nc)**

**End If**

**Elseif CarAngle < 0 And NorthAngle > 0 Then**

**If (Car(1, 1) < DataX(1) And NorthPoint(1, 3) >= NorthPoint(1, 1)) Or (Car(1, 1) >= DataX(1) And NorthPoint(1, 3) < NorthPoint(1, 1)) Then**

**nc = Int((180 + (CarAngle - NorthAngle)) + 0.5)**

**If 0 > nc And nc >= -180 Then**

**nc = 360 + nc**

**End If**

**Text1.Text = "Δφπιξ(°)=" + Str(nc)**

**Elseif (Car(1, 1) >= DataX(1) And NorthPoint(1, 3) >= NorthPoint(1, 1)) Or (Car(1, 1) < DataX(1) And NorthPoint(1, 3) < NorthPoint(1, 1)) Then**

**nc = Int((CarAngle - NorthAngle) + 0.5)**

**If 0 > nc And nc >= -180 Then**

```

    nc = 360 + nc
End If
Text1.Text = "Αφπιξ(°)=" + Str(nc)
End If
ElseIf CarAngle >= 0 And NorthAngle >= 0 Then
    If (Car(1, 1) >= DataX(1) And NorthPoint(1, 3) >= NorthPoint(1, 1)) Or_
(Car(1,1) < DataX(1) And NorthPoint(1, 3) < NorthPoint(1, 1)) Then
        nc = Int((CarAngle - NorthAngle) + 0.5)
        If 0 > nc And nc >= -180 Then
            nc = 360 + nc
        End If
        Text1.Text = "Αφπιξ(°)=" + Str(nc)
    ElseIf (Car(1, 1) < DataX(1) And NorthPoint(1, 3) >= NorthPoint(1, 1)) Or_
(Car(1, 1) >= DataX(1) And NorthPoint(1, 3) < NorthPoint(1, 1)) Then
        If CarAngle < NorthAngle Then
            nc = Int(((CarAngle - NorthAngle) + 180) + 0.5)
            If 0 > nc And nc >= -180 Then
                nc = 360 + nc
            End If
            Text1.Text = "Αφπιξ(°)=" + Str(nc)
        ElseIf CarAngle > NorthAngle Then
            nc = Int(((CarAngle - NorthAngle) - 180) + 0.5)
            If 0 > nc And nc >= -180 Then
                nc = 360 + nc
            End If
            Text1.Text = "Αφπιξ(°)=" + Str(nc)
        End If
    End If
ElseIf CarAngle <= 0 And NorthAngle <= 0 Then
    If (Car(1, 1) >= DataX(1) And NorthPoint(1, 3) >= NorthPoint(1, 1)) Or_
(Car(1,1) < DataX(1) And NorthPoint(1, 3) < NorthPoint(1, 1)) Then
        nc = Int((CarAngle - NorthAngle) + 0.5)
        If 0 > nc And nc >= -180 Then
            nc = 360 + nc

```

```

End If
Text1.Text = "Δφπιξ(°)=" + Str(nc)
ElseIf (Car(1, 1) < DataX(1) And NorthPoint(1, 3) >= NorthPoint(1, 1)) Or_
(Car(1, 1) >= DataX(1) And NorthPoint(1, 3) < NorthPoint(1, 1)) Then
If CarAngle < NorthAngle Then
nc = Int(((CarAngle - NorthAngle) + 180) + 0.5)
If 0 > nc And nc >= -180 Then
nc = 360 + nc
End If
Text1.Text = "Δφπιξ(°)=" + Str(nc)
ElseIf CarAngle > NorthAngle Then
nc = Int(((CarAngle - NorthAngle) - 180) + 0.5)
If 0 > nc And nc >= -180 Then
nc = 360 + nc
End If
Text1.Text = "Δφπιξ(°)=" + Str(nc)
End If
End If
End If

```

Τα **Car(1, 1)** και **DataX(1)** είναι οι μεταβλητές στις οποίες καταχωρούνται η τετμημένη του τελικού και του αρχικού σημείου αντίστοιχα για τον υπολογισμό της **CarAngle**. Τα **NorthPoint(1, 3)** και **NorthPoint(1, 1)** είναι οι μεταβλητές στις οποίες καταχωρούνται η τετμημένη του τελικού και του αρχικού σημείου αντίστοιχα για τον υπολογισμό της **NorthAngle**. Στη μεταβλητή **nc** καταχωρείται η τιμή της γωνίας που δείχνει η πυξίδα.

Όπως φαίνεται από τον κώδικα, στη διαδικασία αυτή υπάρχουν τέσσερις διαφορετικές περιπτώσεις για τον υπολογισμό της **nc**. Οι περιπτώσεις αυτές προκύπτουν από τα πρόσημα των γωνιών **CarAngle** και **NorthAngle**. Η κάθε περίπτωση από αυτές έχει κάποιες υποπεριπτώσεις στις οποίες συγκρίνονται οι τετμημένες των αρχικών σημείων με τα τελικά σημεία και ανάλογα με τα αποτελέσματα των συγκρίσεων αυτών χρησιμοποιούνται διαφορετικοί τύποι για να υπολογιστεί η γωνία **nc** από τις **CarAngle** και **NorthAngle**. Αν από τους

υπολογισμούς προκύπτει η γωνία **nc** αρνητική τότε με τον τύπο  $nc = 360 + nc$  την προσαρμόζουμε σε θετικές τιμές για να είναι συμβατή με τις τιμές που προκύπτουν από την πυξίδα, δηλαδή από 0°-360°.

#### 4.3.5 Τρόπος δειγματοληψίας σημείων

Κατά τη διάρκεια της σχεδίασης επιλέγονται κάποια σημεία τα οποία σαρώνει ο κέρσορας για όση ώρα κινείται το ποντίκι και οι συντεταγμένες αυτών των σημείων της φόρμας αποθηκεύονται. Οι συντεταγμένες των σημείων αυτών χρησιμοποιούνται για τον υπολογισμό των αποστάσεων που θα καλύψει ο όχημα. Ανάλογα με το είδος της σχεδίασης που έχει επιλεγεί διαφέρει και ο τρόπος που επιλέγονται τα σημεία. Όταν γίνεται σχεδίαση με τεθλασμένη γραμμή τα σημεία που θα επιλεγούν κατά τη διάρκεια της σχεδίασης της διαδρομής είναι εκείνα όταν αφήνεται το αριστερό πλήκτρο του ποντικιού. Όταν γίνεται ελεύθερη σχεδίαση η συχνότητα με την οποία θα επιλέγονται σημεία της σχεδιαζόμενης διαδρομής για τους υπολογισμούς καθορίζεται από το χρήστη. Στη φόρμα «Ρυθμίσεις» ο καθορισμός της παραμέτρου “**Απόσταση των δειγματοληπτημένων σημείων**,” ενημερώνει μια μεταβλητή η οποία ορίζει την απόσταση που θα έχουν μεταξύ τους δύο διαδοχικά επιλεγμένα σημεία.

#### 4.3.6 Διαδικασία υπολογισμού της γωνίας στροφής

Κατά τη διάρκεια της σχεδίασης της διαδρομής γίνεται ο υπολογισμός της γωνίας στροφής του οχήματος. Η διαδικασία που ακολουθείται για αυτόν τον υπολογισμό είναι παρόμοια με τη διαδικασία που περιγράφηκε παραπάνω για τη γωνία της διεύθυνσης του οχήματος. Οι συντεταγμένες των σημείων που επιλέγονται καταχωρούνται σε πίνακες δεδομένων – στο **DataX** η τετμημένη και στο **DataY** η τεταγμένη – οι οποίες χρησιμοποιούνται για τον υπολογισμό της κλίσης κάθε τμήματος της διαδρομής που βρίσκεται ανάμεσα στα σημεία δειγματοληψίας ως προς τον οριζόντιο άξονα της φόρμας. Οι κλίσεις αυτές καταχωρούνται σε πίνακα δεδομένων (πίνακας **g**) και χρησιμοποιούνται για τον υπολογισμό της γωνίας της διεύθυνσης του τελευταίου τμήματος της σχεδιαζόμενης διαδρομής ως προς τη διεύθυνση του αμέσως προηγούμενου τμήματος. Παρακάτω παρατίθεται η διαδικασία υπολογισμού της γωνίας στροφής του οχήματος:

```

If i > 2 Then
  If DataX(i) = DataX(i - 1) And DataY(i) > DataY(i - 1) Then
    j = j + 1
    ReDim Preserve g(j)
    g(j) = 90
  ElseIf DataX(i) = DataX(i - 1) And DataY(i) < DataY(i - 1) Then
    j = j + 1
    ReDim Preserve g(j)
    g(j) = -90
  Else
    j = j + 1
    ReDim Preserve g(j)
    g(j) = 180 * Atn((DataY(i) - DataY(i - 1)) / (DataX(i) - DataX(i - 1))) / Pi
  End If
End If
If j = 1 Then
  If g(j) > 0 And CarAngle < 0 Then
    If i = 2 Then
      If (DataX(i) >= DataX(i - 1) And Car(1, 1) < DataX(i - 1)) Or (DataX(i) <_
DataX(i - 1) And Car(1, 1) >= DataX(i - 1)) Then
        AngleCount = AngleCount + 1
        ReDim Preserve Angles(AngleCount)
        Angles(AngleCount) = (g(j) - CarAngle) - 180
        Angles(AngleCount) = Int(Angles(AngleCount) + 0.5)
        Text1.Text = "Δφ(°)=" + Str(Angles(AngleCount))
      ElseIf (DataX(i) >= DataX(i - 1) And Car(1, 1) >= DataX(i - 1)) Or (DataX(i) <_
DataX(i - 1) And Car(1, 1) < DataX(i - 1)) Then
        AngleCount = AngleCount + 1
        ReDim Preserve Angles(AngleCount)
        Angles(AngleCount) = g(j) - CarAngle
        Angles(AngleCount) = Int(Angles(AngleCount) + 0.5)
        Text1.Text = "Δφ(°)=" + Str(Angles(AngleCount))
      End If
    End If
  End If

```

```

ElseIf g(j) < 0 And CarAngle > 0 Then
  If i = 2 Then
    If (DataX(i) < DataX(i - 1) And Car(1, 1) >= DataX(i - 1)) Or (DataX(i) >=
DataX(i - 1) And Car(1, 1) < DataX(i - 1)) Then
      AngleCount = AngleCount + 1
      ReDim Preserve Angles(AngleCount)
      Angles(AngleCount) = 180 + (g(j) - CarAngle)
      Angles(AngleCount) = Int(Angles(AngleCount) + 0.5)
      Text1.Text = "Δφ(°)=" + Str(Angles(AngleCount))
    ElseIf (DataX(i) >= DataX(i - 1) And Car(1, 1) >= DataX(i - 1)) Or (DataX(i) <
DataX(i - 1) And Car(1, 1) < DataX(i - 1)) Then
      AngleCount = AngleCount + 1
      ReDim Preserve Angles(AngleCount)
      Angles(AngleCount) = g(j) - CarAngle
      Angles(AngleCount) = Int(Angles(AngleCount) + 0.5)
      Text1.Text = "Δφ(°)=" + Str(Angles(AngleCount))
    End If
  End If
ElseIf g(j) >= 0 And CarAngle >= 0 Then
  If i = 2 Then
    If (DataX(i) >= DataX(i - 1) And Car(1, 1) >= DataX(i - 1)) Or (DataX(i) <
DataX(i - 1) And Car(1, 1) < DataX(i - 1)) Then
      AngleCount = AngleCount + 1
      ReDim Preserve Angles(AngleCount)
      Angles(AngleCount) = g(j) - CarAngle
      Angles(AngleCount) = Int(Angles(AngleCount) + 0.5)
      Text1.Text = "Δφ(°)=" + Str(Angles(AngleCount))
    ElseIf (DataX(i) < DataX(i - 1) And Car(1, 1) >= DataX(i - 1)) Or (DataX(i) >=
DataX(i - 1) And Car(1, 1) < DataX(i - 1)) Then
      If g(j) < CarAngle Then
        AngleCount = AngleCount + 1
        ReDim Preserve Angles(AngleCount)
        Angles(AngleCount) = (g(j) - CarAngle) + 180
        Angles(AngleCount) = Int(Angles(AngleCount) + 0.5)
      End If
    End If
  End If

```



```

Text1.Text = "Δφ(°)=" + Str(Angles(AngleCount))
ElseIf g(j) > CarAngle Then
    AngleCount = AngleCount + 1
    ReDim Preserve Angles(AngleCount)
    Angles(AngleCount) = (g(j) - CarAngle) - 180
    Angles(AngleCount) = Int(Angles(AngleCount) + 0.5)
    Text1.Text = "Δφ(°)=" + Str(Angles(AngleCount))
End If
End If
End If
ElseIf g(j) <= 0 And CarAngle <= 0 Then
    If i = 2 Then
        If (DataX(i) >= DataX(i - 1) And Car(1, 1) >= DataX(i - 1)) Or (DataX(i) <_
DataX(i - 1) And Car(1, 1) < DataX(i - 1)) Then
            AngleCount = AngleCount + 1
            ReDim Preserve Angles(AngleCount)
            Angles(AngleCount) = g(j) - CarAngle
            Angles(AngleCount) = Int(Angles(AngleCount) + 0.5)
            Text1.Text = "Δφ(°)=" + Str(Angles(AngleCount))
        ElseIf (DataX(i) < DataX(i - 1) And Car(1, 1) >= DataX(i - 1)) Or (DataX(i) >=_
DataX(i - 1) And Car(1, 1) < DataX(i - 1)) Then
            If g(j) < CarAngle Then
                AngleCount = AngleCount + 1
                ReDim Preserve Angles(AngleCount)
                Angles(AngleCount) = (g(j) - CarAngle) + 180
                Angles(AngleCount) = Int(Angles(AngleCount) + 0.5)
                Text1.Text = "Δφ(°)=" + Str(Angles(AngleCount))
            ElseIf g(j) > CarAngle Then
                AngleCount = AngleCount + 1
                ReDim Preserve Angles(AngleCount)
                Angles(AngleCount) = (g(j) - CarAngle) - 180
                Angles(AngleCount) = Int(Angles(AngleCount) + 0.5)
                Text1.Text = "Δφ(°)=" + Str(Angles(AngleCount))
            End If
        End If
    End If
End If

```

```

End If
End If
End If
Elseif j > 1 Then
If g(j) > 0 And g(j - 1) < 0 Then
If i > 2 Then
If (DataX(i) >= DataX(i - 1) And DataX(i - 1) < DataX(i - 2)) Or (DataX(i) <_
DataX(i - 1) And DataX(i - 1) >= DataX(i - 2)) Then
AngleCount = AngleCount + 1
ReDim Preserve Angles(AngleCount)
Angles(AngleCount) = (g(j) - g(j - 1)) - 180
Angles(AngleCount) = Int(Angles(AngleCount) + 0.5)
Text1.Text = "Δφ(°)=" + Str(Angles(AngleCount))
Elseif (DataX(i) >= DataX(i - 1) And DataX(i - 1) >= DataX(i - 2)) Or_
(DataX(i) < DataX(i - 1) And DataX(i - 1) < DataX(i - 2)) Then
AngleCount = AngleCount + 1
ReDim Preserve Angles(AngleCount)
Angles(AngleCount) = g(j) - g(j - 1)
Angles(AngleCount) = Int(Angles(AngleCount) + 0.5)
Text1.Text = "Δφ(°)=" + Str(Angles(AngleCount))
End If
End If
Elseif g(j) < 0 And g(j - 1) > 0 Then
If i > 2 Then
If (DataX(i) < DataX(i - 1) And DataX(i - 1) >= DataX(i - 2)) Or (DataX(i) >=_
DataX(i - 1) And DataX(i - 1) < DataX(i - 2)) Then
AngleCount = AngleCount + 1
ReDim Preserve Angles(AngleCount)
Angles(AngleCount) = 180 + (g(j) - g(j - 1))
Angles(AngleCount) = Int(Angles(AngleCount) + 0.5)
Text1.Text = "Δφ(°)=" + Str(Angles(AngleCount))
Elseif (DataX(i) >= DataX(i - 1) And DataX(i - 1) >= DataX(i - 2)) Or_
(DataX(i) < DataX(i - 1) And DataX(i - 1) < DataX(i - 2)) Then
AngleCount = AngleCount + 1

```

```

ReDim Preserve Angles(AngleCount)
Angles(AngleCount) = g(j) - g(j - 1)
Angles(AngleCount) = Int(Angles(AngleCount) + 0.5)
Text1.Text = "Δφ(°)=" + Str(Angles(AngleCount))
End If
End If
ElseIf g(j) >= 0 And g(j - 1) >= 0 Then
If i > 2 Then
If (DataX(i) >= DataX(i - 1) And DataX(i - 1) >= DataX(i - 2)) Or (DataX(i) <_
DataX(i - 1) And DataX(i - 1) < DataX(i - 2)) Then
AngleCount = AngleCount + 1
ReDim Preserve Angles(AngleCount)
Angles(AngleCount) = g(j) - g(j - 1)
Angles(AngleCount) = Int(Angles(AngleCount) + 0.5)
Text1.Text = "Δφ(°)=" + Str(Angles(AngleCount))
ElseIf (DataX(i) < DataX(i - 1) And DataX(i - 1) >= DataX(i - 2)) Or (DataX(i)_
>= DataX(i - 1) And DataX(i - 1) < DataX(i - 2)) Then
If g(j) < g(j - 1) Then
AngleCount = AngleCount + 1
ReDim Preserve Angles(AngleCount)
Angles(AngleCount) = (g(j) - g(j - 1)) + 180
Angles(AngleCount) = Int(Angles(AngleCount) + 0.5)
Text1.Text = "Δφ(°)=" + Str(Angles(AngleCount))
ElseIf g(j) > g(j - 1) Then
AngleCount = AngleCount + 1
ReDim Preserve Angles(AngleCount)
Angles(AngleCount) = (g(j) - g(j - 1)) - 180
Angles(AngleCount) = Int(Angles(AngleCount) + 0.5)
Text1.Text = "Δφ(°)=" + Str(Angles(AngleCount))
End If
End If
End If
ElseIf g(j) <= 0 And g(j - 1) <= 0 Then
If i > 2 Then

```

```

If (DataX(i) >= DataX(i - 1) And DataX(i - 1) >= DataX(i - 2)) Or (DataX(i) <_
DataX(i - 1) And DataX(i - 1) < DataX(i - 2)) Then
    AngleCount = AngleCount + 1
    ReDim Preserve Angles(AngleCount)
    Angles(AngleCount) = g(j) - g(j - 1)
    Angles(AngleCount) = Int(Angles(AngleCount) + 0.5)
    Text1.Text = "Δφ(°)=" + Str(Angles(AngleCount))
ElseIf (DataX(i) < DataX(i - 1) And DataX(i - 1) >= DataX(i - 2)) Or (DataX(i)_
>= DataX(i - 1) And DataX(i - 1) < DataX(i - 2)) Then
    If g(j) < g(j - 1) Then
        AngleCount = AngleCount + 1
        ReDim Preserve Angles(AngleCount)
        Angles(AngleCount) = (g(j) - g(j - 1)) + 180
        Angles(AngleCount) = Int(Angles(AngleCount) + 0.5)
        Text1.Text = "Δφ(°)=" + Str(Angles(AngleCount))
    ElseIf g(j) > g(j - 1) Then
        AngleCount = AngleCount + 1
        ReDim Preserve Angles(AngleCount)
        Angles(AngleCount) = (g(j) - g(j - 1)) - 180
        Angles(AngleCount) = Int(Angles(AngleCount) + 0.5)
        Text1.Text = "Δφ(°)=" + Str(Angles(AngleCount))
    End If
End If
End If
End If
End If

```

Στον πίνακα δεδομένων **Angles** καταχωρούνται οι γωνίες στροφής του οχήματος. Στον πίνακα δεδομένων **g** καταχωρούνται οι γωνίες που σχηματίζουν τα επιμέρους τμήματα της σχεδιασμένης διαδρομής με τον οριζόντιο άξονα της φόρμας. Στους πίνακες δεδομένων **DataX** και **DataY** καταχωρούνται οι συντεταγμένες των επιλεγμένων σημείων της διαδρομής τα οποία θα χρησιμοποιηθούν στους υπολογισμούς των γωνιών στροφής του οχήματος.

Στη διαδικασία που ακολουθείται για τον υπολογισμό των γωνιών ελέγχεται το πρόσημο των γωνιών που σχηματίζουν τα δύο τελευταία τμήματα της διαδρομής με τον οριζόντιο άξονα της φόρμας και κατόπιν η σχέση των συντεταγμένων των άκρων των δύο τελευταίων τμημάτων της διαδρομής. Ανάλογα με τα αποτελέσματα των συγκρίσεων αυτών χρησιμοποιείται και άλλος τύπος για τον προσδιορισμό της γωνίας στροφής του τελευταίου τμήματος της διαδρομής.

Όπως συμβαίνει και στη διαδικασία για το καθορισμό της διεύθυνσης του οχήματος, από το πρόσημο των γωνιών των επιμέρους τμημάτων της διαδρομής που σχηματίζουν με τον οριζόντιο άξονα της φόρμας προκύπτουν τέσσερις περιπτώσεις. Οι περιπτώσεις αυτές διαιρούνται σε άλλες περιπτώσεις, οι οποίες προκύπτουν από τη σύγκριση των συντεταγμένων των σημείων που αποτελούν τα άκρα των επιμέρους τμημάτων της σχεδιασμένης διαδρομής. Η γωνία που προκύπτει από τη διαδικασία αυτή κυμαίνεται από  $-180^\circ$  έως  $180^\circ$ . Οι αρνητικές γωνίες που προκύπτουν αναφέρονται σε αριστερή στροφή ενώ οι θετικές σε δεξιά. Αυτή η διαδικασία υπολογισμού της γωνίας χρησιμοποιείται και στους δυο τρόπους σχεδίασης.

#### 4.3.7 Προβολή σημείων καμπής

Ο χρήστης μπορεί να παρακολουθεί τις γωνίες στροφής που προκύπτουν ταυτόχρονα με τις συντεταγμένες των σημείων που έχουν επιλεγεί για τον υπολογισμό των γωνιών στροφής ακολουθώντας τη διαδρομή **Προβολή→Σημεία καμπής**. Με την επιλογή αυτή εμφανίζεται μια φόρμα στην οποία φαίνονται τα δεδομένα αυτά σε μορφή στήλης. Η συμπλήρωση των κελιών γίνεται αμέσως μόλις καταχωρηθούν στους αντίστοιχους πίνακες δεδομένων οι συντεταγμένες και οι γωνίες. Οι συντεταγμένες είναι σε pixel και σε μέτρα. Το σημείο με συντεταγμένες  $(x,y)=(0,0)$  είναι το πάνω πιο αριστερό σημείο της φόρμας.

Σημεία Καμπής Διαδρομής					
	Δφ(°)	x(pixels)	y(pixels)	x(m)	y(m)
Σημείο 1		158	208	15.8	20.8
Σημείο 2	0	162	208	16.2	20.8
Σημείο 3	-24	208	185	20.8	18.5
Σημείο 4	37	266	196	26.6	19.6
Σημείο 5	35	298	229	29.8	22.9

Σχ.18

#### 4.3.8 Εκκαθάριση-αναίρεση

Κατά τη διάρκεια της σχεδίασης μπορεί ο χρήστης να καθαρίσει τη φόρμα ή να διαγράψει μέρος της διαδρομής. Αυτές οι διαδικασίες υπάρχουν στην επιλογή του μενού της κύριας φόρμας **Επεξεργασία**. Με την επιλογή **Εκκαθάριση** διαγράφει τη σχεδιασμένη διαδρομή από τη φόρμα. Με αυτή την επιλογή καθαρίζονται ειδικά τα περιεχόμενα των πινάκων που περιέχουν τις συντεταγμένες, τις κλίσεις και τις γωνίες στροφής. Με την επιλογή **Αναίρεση** διαγράφεται το τελευταίο τμήμα της διαδρομής μόνο. Για να διαγραφούν κι άλλα τμήματα από τη σχεδιασμένη διαδρομή θα πρέπει να γίνει κλικ στην **Αναίρεση** όσες φορές είναι απαραίτητο. Με την επιλογή αυτή καθαρίζονται όσα από τα περιεχόμενα των πινάκων για τις συντεταγμένες, τις κλίσεις και τις γωνίες στροφής έχουν σχέση με το τμήμα της διαδρομής που σβήστηκε.

#### 4.3.9 Αποθήκευση διαδρομής

Μια επιπλέον δυνατότητα του προγράμματος ελέγχου είναι η δυνατότητα αποθήκευσης κάποιας διαδρομής για τη χρησιμοποίησή της κάποια άλλη στιγμή. Επιλέγοντας **Αρχείο→Αποθήκευση** από το μενού, αποθηκεύεται η σχεδιασμένη διαδρομή σε ένα αρχείο κειμένου (.txt). Για την αποθήκευση ανοίγει το πλαίσιο διαλόγου των Windows για αποθήκευση αρχείων. Ο χρήστης μπορεί να χρησιμοποιήσει όποιο όνομα θέλει. Αντίστοιχα, για το άνοιγμα κάποιου αρχείου που περιέχει μια αποθηκευμένη διαδρομή, ο χρήστης πρέπει να επιλέξει

**Αρχείο**→**Άνοιγμα** από το μενού. Τότε ανοίγει το πλαίσιο διαλόγου των Windows για το άνοιγμα αρχείων και ο χρήστης μπορεί να επιλέξει το αρχείο που επιθυμεί.

Για τη διαδικασία αποθήκευσης και ανοίγματος κάποιου αρχείου διαδρομής χρησιμοποιείται το ελεγκτήριο **CommonDialog** της **Visual Basic**.

Κατά την αποθήκευση της διαδρομής αποθηκεύονται τα περιεχόμενα των πινάκων δεδομένων που περιέχουν τις συντεταγμένες των σημείων δειγματοληψίας, τις κλίσεις και τις γωνίες στροφής, οι διαστάσεις αυτών των πινάκων, η γωνία της πυξίδα και η σημαία που καθορίζει το είδος της σχεδίασης. Από τα δεδομένα αυτά, ανοίγοντας το συγκεκριμένο αρχείο σχεδιάζεται η διαδρομή και καταχωρούνται σε πίνακα με τις κατάλληλες διαστάσεις οι γωνίες στροφής. Επίσης μπορεί ο χρήστης να προσανατολίσει το όχημα αφού εμφανίζεται στην οθόνη η γωνία που αντιστοιχεί στη πυξίδα. Τέλος αποθηκεύονται το αρχείο του χάρτη οδηγού, η κλίμακά του και η γραμμική ταχύτητα που χρησιμοποιήθηκε για τον υπολογισμό των χρόνων κάλυψης της διαδρομής από το όχημα. Μόλις ανοίξει το αρχείο ενημερώνονται οι μεταβλητές που χρησιμοποιούνται για την ταχύτητα και την κλίμακα και φορτώνεται ο χάρτης.

#### 4.3.10 Εκτέλεση διαδρομής

Αφού σχεδιαστεί η επιθυμητή διαδρομή από το χρήστη, για να αρχίσει να την εκτελεί το όχημα, ο χρήστης πρέπει να επιλέξει από το μενού **Εκτέλεση**→**Εκκίνηση**. Ο χρήστης μπορεί να παρακολουθεί την πορεία του οχήματος στην οθόνη του υπολογιστή με τη βοήθεια ενός κόκκινου κύκλου ο οποίος κινείται πάνω στη σχεδιασμένη πορεία κατά τη διάρκεια εκτέλεσης της από το όχημα. Αυτό επιτυγχάνεται υπολογίζοντας τις συντεταγμένες των σημείων που βρίσκονται ενδιάμεσα από τα σημεία δειγματοληψίας σε συνάρτηση με την ταχύτητα του οχήματος και σχεδιάζοντας τον κύκλο στα σημεία αυτά. Ο χρόνος που θα φαίνεται σε κάθε σημείο πάνω στη διαδρομή ο κύκλος είναι ίσος με ένα δευτερόλεπτο.

#### 4.3.11 Διαδικασίες σχεδίασης με ελεύθερη γραμμή

Σε αυτό τον τρόπο σχεδίασης η επιλογή των σημείων της σχεδιαζόμενης διαδρομής που χρησιμοποιούνται στους υπολογισμούς γίνεται καθορίζοντας ο χρήστης την παράμετρο απόσταση δειγματοληπτημένων σημείων. Ο καθορισμός της

παραμέτρου αυτής έχει σχέση με τη διαφορά σε pixel των σημείων που δειγματοληπτούνται. Ο τρόπος επιλογής σημείων σε αυτό το είδος σχεδίασης παρατίθεται παρακάτω:

Στη μεταβλητή **Diafora** καταχωρείται η σχετική θέση του τρέχοντος σημείου ως προς την αρχή των αξόνων της φόρμας όπως φαίνεται στους παρακάτω τύπους:

$$\mathbf{Diafora(1, 1) = X - Anafora(1, 1)}$$

$$\mathbf{Diafora(1, 2) = Y - Anafora(1, 2)}$$

όπου **X** η τετμημένη του τρέχοντος σημείου, η **Y** η τεταγμένη, και οι μεταβλητές **Anafora(1, 1)** και **Anafora(1, 2)** οι συντεταγμένες του αρχικού σημείου των αξόνων της φόρμας.

Το επόμενο σημείο το οποίο θα καταχωρηθεί θα πρέπει να διαφέρει από το τελευταίο καταχωρημένο σε τουλάχιστον μια από τις συντεταγμένες του κατά την τιμή της παραμέτρου “**Απόσταση δειγματοληπτημένων σημείων**”,. Από τον τρόπο επιλογής των σημείων είναι φανερό πως όσο αυξάνεται η παράμετρος “**Απόσταση δειγματοληπτημένων σημείων**”, τόσο η δειγματοληψία που γίνεται έχει μεγάλες αποκλίσεις από τη σχεδιαζόμενη διαδρομή. Σε περίπτωση αποθήκευσης της διαδρομής και χρησιμοποίηση της ξανά, οι αποκλίσεις των επιλεγμένων σημείων από την αρχικά σχεδιασμένη διαδρομή θα οδηγήσουν στο λάθος επανασχεδιασμό της στο χάρτη οδηγό και σε σφάλματα στους χρόνους και στις αποστάσεις που έχει να καλύψει το όχημα. Για αυτό το λόγο θα πρέπει η παράμετρος της απόστασης να παίρνει μικρές τιμές.

Με την επιλογή **Εκτέλεση→Στάση** από το μενού κατά τη διάρκεια εκτέλεσης της διαδρομής, το όχημα κάνει μια στάση στο σημείο στο οποίο δείχνει ο κόκκινος κύκλος. Για να γίνει πάλι η έναρξη της κίνησης του οχήματος θα γίνει κάνοντας κλικ ξανά στο **Εκτέλεση→Εκκίνηση**. Τότε το όχημα θα ξεκινήσει να κινείται για να ολοκληρώσει το τμήμα της σχεδιασμένης διαδρομής που έχει απομείνει.

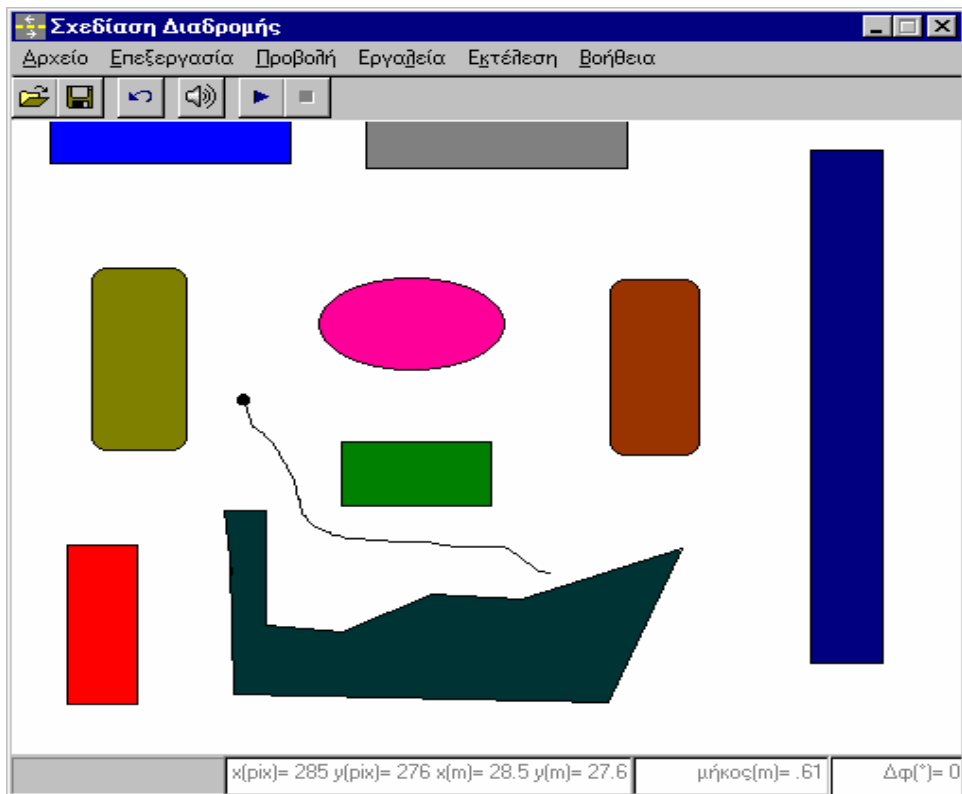
Από τις συντεταγμένες των δειγματοληπτημένων σημείων υπολογίζονται οι αποστάσεις τις οποίες θα καλύψει το όχημα. Από τις αποστάσεις θα προκύψουν οι



χρόνοι κατά τη διάρκεια των οποίων το ποτενσιόμετρο για τη στροφή θα διατηρεί μια συγκεκριμένη τιμή. Οι αποστάσεις καταχωρούνται σε ένα πίνακα δεδομένων (**Length**) και υπολογίζονται από τη τετραγωνική ρίζα του αθροίσματος των τετραγώνων της διαφοράς των συντεταγμένων των άκρων του κάθε τμήματος της διαδρομής. Παρακάτω ο τύπος υπολογισμού των αποστάσεων (τύπος 1). Οι χρόνοι καταχωρούνται σε ένα πίνακα δεδομένων (**Times**) και προκύπτουν από τη διαίρεση των αποστάσεων με τη ταχύτητα (τύπος 2).

1.  $\text{Length}(\text{LengthCount}) = \text{Sqr}((\text{DataX}(i) - \text{DataX}(i - 1))^2 + (\text{DataY}(i) - \text{DataY}(i - 1))^2)$
2.  $\text{Times}(\text{TimesCount}) = (\text{Length}(\text{LengthCount}) * \text{Klimaka}) / (100 * \text{Velocity})$

Μόλις γίνει η εκκίνηση της εκτέλεσης του προγράμματος ενεργοποιείται ένας **Timer** (ελεγκτήριο της **Visual Basic**), ο οποίος λειτουργεί σαν χρονόμετρο. Αρχίζει ταυτόχρονα η μετατροπή των γωνιών από δεκαδική σε δυαδική μορφή και η αποστολή τους στη σειριακή. Οι χρόνοι που έχουν υπολογιστεί για την εκτέλεση των επιμέρους τμημάτων της σχεδιασμένης διαδρομής, τα οποία έχουν καταχωρηθεί στον πίνακα δεδομένων **Times** ενημερώνουν την ιδιότητα **Interval** του **Timer**. Έτσι διατηρείται η κάθε τιμή της γωνίας και ταχύτητας για χρόνο ίσο με τον αντίστοιχο για κάθε τμήμα της διαδρομής. Αυτό συμβαίνει γιατί η ιδιότητα **Interval** καθορίζει το χρόνο για τον οποίο ο χρονιστής **Timer** θα εκτελεί συγκεκριμένες ενέργειες και θα διατηρεί σταθερές τις εξόδους του συστήματος που ελέγχονται από το πρόγραμμα. Παρακάτω στο σχήμα φαίνεται μια πορεία σχεδιασμένη με ελεύθερο τρόπο:

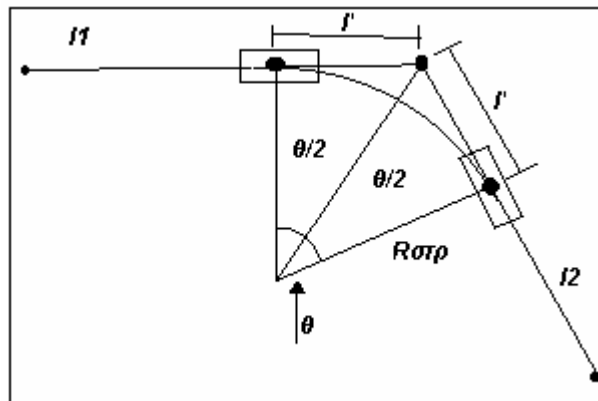


Σχ.19

#### 4.3.12 Διαδικασίες σχεδίασης με ευθύγραμμα τμήματα

Σε αυτόν τον τρόπο σχεδίασης, σε αντίθεση με τον ελεύθερο, η δειγματοληψία γίνεται μόλις αφηθεί το αριστερό κουμπί του ποντικιού. Τότε αποθηκεύονται οι συντεταγμένες του, και υπολογίζονται οι αποστάσεις και οι χρόνοι που είναι απαραίτητοι για την εκτέλεση της διαδρομής από το όχημα.

Η διαδικασία που χρησιμοποιείται για τον υπολογισμό των αποστάσεων και των χρόνων διαφέρει σε αυτό το είδος σχεδίασης. Επιλέχθηκε για την υλοποίηση της διαδρομής στις στροφές το όχημα να στρίβει με την ίδια πάντα στροφή του συστήματος διεύθυνσης. Η επιθυμητή γωνία επιτυγχάνεται αφήνοντας για χρόνο κατάλληλο το όχημα να στρίβει ώστε να ευθυγραμμιστεί ο άξονας του με τον άξονα της σχεδιασμένης πορείας. Στο παρακάτω σχήμα απεικονίζεται η πορεία που έχει το όχημα κατά τη διάρκεια της στροφής, καθώς και πως πρέπει να υπολογιστούν οι αποστάσεις και οι χρόνοι:



Σχ.20

Από τους μαθηματικούς τύπους που ακολουθούν υπολογίζονται οι αποστάσεις και οι χρόνοι για τη στροφή του οχήματος:

$I' = R_{στρ} * εφ(θ/2)$  όπου  $R_{στρ}$  = σταθερό, η ακτίνα του κύκλου που διαγράφει το όχημα με τη σταθερή ίδια στροφή του συστήματος διεύθυνσης που έχει επιλεγεί για την υλοποίηση όλων των στροφών που κάνει το όχημα για να ακολουθήσει κάθε σχεδιασμένη διαδρομής –  $θ$  η σχεδιαζόμενη γωνία στροφής που πρέπει να εκτελέσει το όχημα στο τελευταίο τμήμα της διαδρομής που ακολουθεί

$I_2 = I - I'$  όπου  $I$  = το ολικό μήκος του σχεδιαζόμενου τμήματος –  $I'$  = το μήκος του τμήματος της διαδρομής κατά το οποίο το όχημα ακολουθεί καμπύλη τροχιά για να ευθυγραμμίσει τον άξονα του με τη διεύθυνση του επόμενου τμήματος της διαδρομής –  $I_2$  = το μήκος που θα διανύσει τελικά το όχημα σε ευθεία

$I_θ = θ * 2 * π * R_{στρ} / 360°$  όπου  $I_θ$  = το μήκος της τροχιάς που θα διανύσει στρίβοντας το όχημα για να ευθυγραμμίσει τον άξονα του με τη διεύθυνση του επόμενου τμήματος της σχεδιαζόμενης διαδρομής

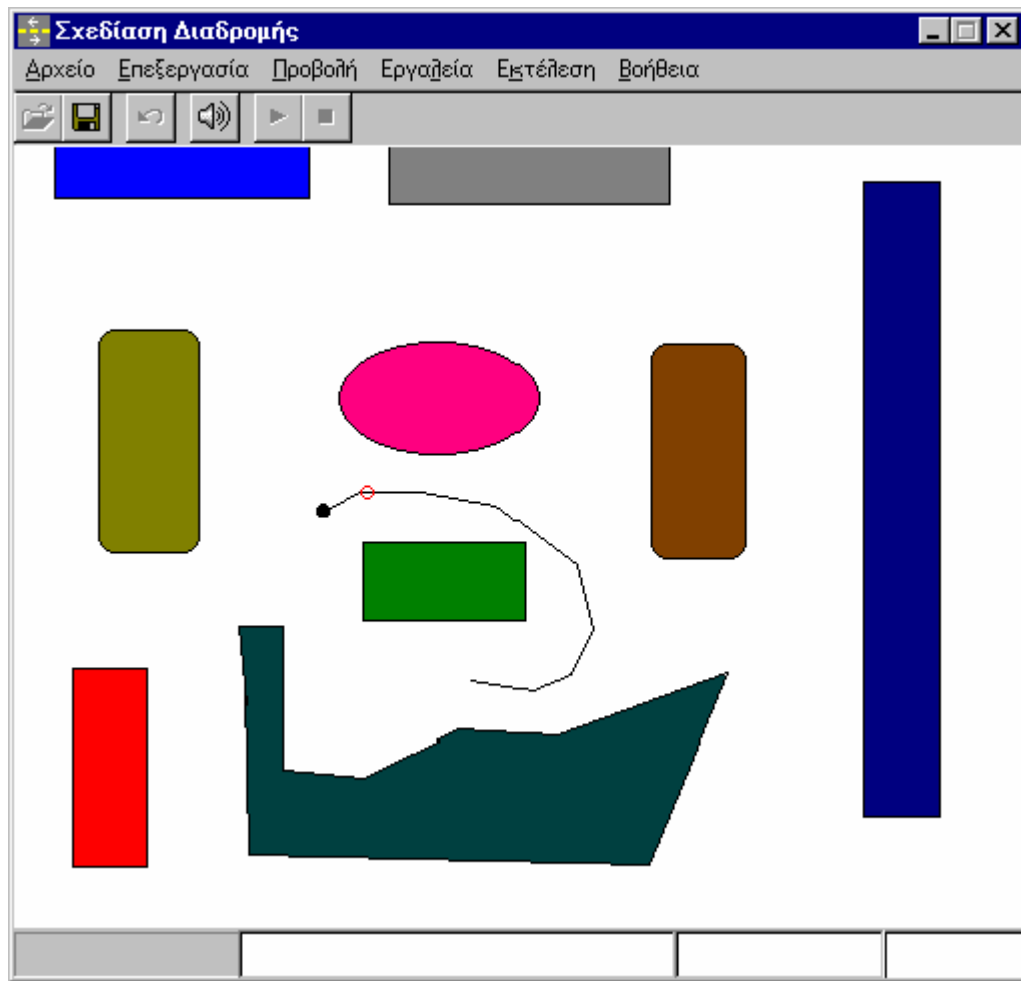
Από τα  $I_θ$ ,  $I_2$  υπολογίζονται οι χρόνοι διαιρώντας με την ταχύτητα που έχει το όχημα όταν στρίβει και όταν προχωράει ευθεία αντίστοιχα.

Η διαδικασία που ακολουθείται όταν αρχίσει να «τρέχει» η εφαρμογή είναι παρόμοια με τη διαδικασία που περιγράφεται στην ελεύθερη σχεδίαση. Οι

αποστάσεις των τμημάτων της διαδρομής υπολογίζονται με τον ίδιο τρόπο και καταχωρούνται σε ένα πίνακα δεδομένων (**Length**) κατά τη διαδικασία σχεδίασης. Κατόπιν υπολογίζονται με τη βοήθεια των γωνιών τα καμπύλα τμήματα που πρέπει να διανύσει το όχημα κατά τη διάρκεια στροφής καθώς επίσης και τα νέα τμήματα που απομένουν μετά την ευθυγράμμιση του οχήματος με την εκάστοτε νέα διεύθυνση. Οι τύποι που χρησιμοποιούνται για αυτό το σκοπό παρατέθηκαν παραπάνω (τύποι για τα **I**, **I2** και **Iθ**). Μετά υπολογίζονται οι χρόνοι για τους οποίους το όχημα θα πηγαίνει ευθεία και τους χρόνους για τους οποίους θα στρίβει. Αυτοί οι χρόνοι καταχωρούνται σε ένα πίνακα δεδομένων με το όνομα **Times**.

Μόλις αρχίσει να «τρέχει» η εφαρμογή ενεργοποιείται ένας **Timer** ο οποίος χρησιμοποιείται σαν χρονόμετρο στην ιδιότητα **Interval** του οποίου καταχωρούνται οι υπολογισμένοι χρόνοι που αναφέρονται παραπάνω για τα επιμέρους τμήματα της διαδρομής. Με την ίδια διαδικασία όπως και στην ελεύθερη σχεδίαση – με μόνη διαφορά ότι εδώ οι τιμές που πηγαίνουν στο ποτενσιόμετρο της στροφής είναι μόνο τρεις, δηλαδή στροφή αριστερά στροφή δεξιά και ευθεία – γίνεται η μετατροπή δε δυαδικό των αριθμών που θα κατέβουν στα ποτενσιόμετρα για τη γωνία και την ταχύτητα. Οι τιμές αυτές θα παραμείνουν για χρόνο ίσο με αυτόν που έχει υπολογιστεί.

Σε αυτόν τον τρόπο σχεδίασης δεν υπάρχει η δυνατότητα να ακινητοποιηθεί το όχημα κατά τη διάρκεια της κίνησης του, γιατί η ακρίβεια κατά την έναρξη ξανά της κίνησης του δεν είναι μεγάλη. Αυτό συμβαίνει γιατί χρησιμοποιούνται δυο **Timer** στη διαδικασία εκτέλεσης του προγράμματος – ο ένας για τους χρόνους και τις τιμές των ποτενσιόμετρων και ο άλλος για τη σχεδίαση του κύκλου παρακολούθησης του οχήματος κατά τη διάρκεια της εκτέλεσης – από τους οποίους ο δεύτερος μετράει και το συνολικό χρόνο εκτέλεσης της διαδρομής. Η ακρίβεια του μετρητή αυτού είναι ένα δευτερόλεπτο. Αν ακινητοποιηθεί το όχημα ο **Timer** αυτός θα χρησιμοποιηθεί για το σωστό χρόνο έναρξης της κίνησης του οχήματος με σκοπό την ολοκλήρωση της διαδρομής, ο οποίος λόγω μικρής ακρίβειας του μετρητή θα έχει μια σημαντική απόκλιση από το σωστό εναπομείναντα χρόνο. Παρακάτω φαίνεται η εκτέλεση μιας διαδρομής σχεδιασμένης με αυτόν τον τρόπο:



Σχ.21

## 5 ΕΠΙΛΟΓΟΣ

Σε αυτή την εφαρμογή γίνεται ο έλεγχος της κίνησης του οχήματος μέσω της σειριακής θύρας του υπολογιστή. Έγινε προσπάθεια να υλοποιηθεί ένα σύστημα ελέγχου του οχήματος απλό και αξιόπιστο. Τα ηλεκτρονικά μέρη του συστήματος είναι απλά. Χρησιμοποιήθηκε ένα διπλό ψηφιακό ποτενσιόμετρο, το οποίο αντικατέστησε δυο αναλογικά ποτενσιόμετρα που βρίσκονταν στην τηλεκατεύθυνση για τον έλεγχο των παραμέτρων κίνησης του οχήματος (ταχύτητα και γωνία στροφής). Από τη σειριακή θύρα του υπολογιστή «κατεβαίνουν» κωδικές λέξεις οι οποίες προγραμματίζουν το ψηφιακό ποτενσιόμετρο.

Υλοποιήθηκαν δυο προγράμματα για τον έλεγχο του οχήματος. Στο ένα ο έλεγχος γίνεται σε πραγματικό χρόνο, και στο άλλο όχι. Στο πρόγραμμα πραγματικού χρόνου ο έλεγχος των παραμέτρων γίνεται με τη βοήθεια δυο μάρων ολίσθησης. Στο άλλο σχεδιάζεται μια διαδρομή η οποία ακολουθείται στη συνέχεια από το όχημα.

Η εφαρμογή αυτή παρουσιάζει αρκετές δυσκολίες και η αξιοπιστία της, για τις πραγματικές συνθήκες κίνησης των οχημάτων, είναι περιορισμένη. Ο βασικός λόγος είναι ό,τι δε γίνεται ο σαφής καθορισμός της θέσης του οχήματος στο χώρο κατά τη διάρκεια της κίνησής του.

Μια άλλη δυσκολία που παρουσιάστηκε ήταν κατά τη διάρκεια εκτέλεσης της εφαρμογής για την υλοποίηση μιας διαδρομής. Ο θεωρητικός χρόνος διάρκεια της κίνησης του οχήματος – υπολογιζόμενος από την απόσταση που έχει να διανύσει το όχημα και την καθοριζόμενη από το χρήστη σταθερή ταχύτητα – διέφερε από το χρόνο που χρειαζόταν το όχημα για να υλοποιήσει τη διαδρομή. Αυτή η διαφοροποίηση στους χρόνους οφείλεται σε δύο παράγοντες. Κυρίως στον Η/Υ στον οποίο «τρέχει» η εφαρμογή και στον τρόπο που υλοποιήθηκε το πρόγραμμα ελέγχου του οχήματος.

Στο πρόγραμμα, κατά τη διάρκεια εκτέλεσης κάποιας διαδρομής σαρώνονται κάποιοι πίνακες δεδομένων στους οποίους έχουν καταχωρηθεί οι χρόνοι και οι τιμές που θα κατέβουν στα ποτενσιόμετρα της ταχύτητας και της στροφής. Όσο μεγαλύτεροι είναι αυτοί οι πίνακες τόσο ο Η/Υ στον οποίο «τρέχει» η εφαρμογή

προσθέτει καθυστέρηση στην εκτέλεση η οποία οφείλεται στη διαδικασία ανάγνωσης των δεδομένων στους πίνακες. Η καθυστέρηση αυτή προκαλεί μικρές αποκλίσεις στη διαδρομή που ακολουθεί το όχημα με αποτέλεσμα το τελικό σημείο στόχος να μην μπορεί να προσεγγιστεί με πολύ μεγάλη ακρίβεια.

Οι μηχανικές ατέλειες που υπάρχουν στο όχημα προσθέτουν αποκλίσεις στην ακολουθούμενη από το όχημα διαδρομή. Στις δεξιές στροφές η τροχιά είναι ελαφρώς μεγαλύτερη από την τροχιά που ακολουθεί το όχημα στις αριστερές στροφές.

## **ΒΙΒΛΙΟΓΡΑΦΙΑ - ΑΝΑΦΟΡΕΣ**

1. Εγχειρίδιο της VISUAL BASIC 4 – Nathan Gurewich & Ori Gurewich, 1996
2. Πλήρες εγχειρίδιο της VISUAL BASIC 5 – Evangelos Petroustos, 1998
3. Πλήρες εγχειρίδιο της VISUAL BASIC 6 – E. Petroustos, 1999
4. ΡΟΜΠΟΤΙΚΗ – ΔΗΜΗΤΡΙΟΣ Μ. ΕΜΙΡΗΣ, σελ. 1-94, 1999
5. Designing Autonomous Mobile Robots – John M. Holland, 2004
6. Ελεγκτής σερβομηχανισμών – Ελέκτορ, 5/99
7. DIRECT CONTROL OF PC SERIAL LINES USING MSCOMM – Everyday practical Electronics, Robert Penfold, February 2003
8. UPDATE ON DIRECT CONTROL OF PC SERIAL LINES USING MSCOMM – Everyday practical Electronics, Robert Penfold, April 2003