

**ΑΤΕΙ ΚΡΗΤΗΣ – ΠΑΡΑΡΤΗΜΑ ΧΑΝΙΩΝ**  
**ΤΜΗΜΑ ΗΛΕΚΤΡΟΝΙΚΗΣ**

**ΑΝΑΛΥΣΗ ΜΕΤΡΗΣΕΩΝ ΚΥΨΕΛΩΤΩΝ ΔΙΚΤΥΩΝ**  
**ΤΡΙΤΗΣ ΓΕΝΙΑΣ (3G)**

**Data analysis in 3<sup>rd</sup> generation cellular systems (3G)**



**ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ**  
**ΧΑΡΙΤΑΚΗΣ ΔΗΜΗΤΡΙΟΣ**  
**ΚΟΥΤΣΟΥΡΗΣ ΣΤΕΛΙΟΣ**

**ΕΠΙΒΛΕΠΩΝ ΚΑΘΗΓΗΤΗΣ**

**Δρ. Ευάγγελος Κόκκινος**

**Μέλη επιτροπής εποπτείας**

**Ηλ. Μηχ. Μαυρεδάκης Μιχαήλ**

**Δρ. Βαρδιάμπασης Ιωάννης**

**ΧΑΝΙΑ**  
**Μάρτιος 2009**

# Περιεχόμενα

## Εισαγωγή

### Κεφάλαιο 1 - Παρουσίαση των λειτουργιών του προγράμματος

• 1.1 Λίγα λόγια για την Anite plc.....	2
• 1.2 Σκοπός και λειτουργίες του 3G Analysis.....	10
• 1.3 Γραφήματα.....	12
• 1.4 Πίνακες Μετρήσεων.....	15
• 1.5 Αρχεία Μετρήσεων – Σταθεροί Παράμετροι – Ανάλυση.....	17
➢ Γεγονός ECN0.....	19
➢ Γεγονός CAD.....	23
➢ Γεγονός CAF.....	24
• 1.6 Εισαγωγή στην Visual Basic .net.....	25

### Κεφάλαιο 2 – Ανάλυση Κώδικα

• 2.1 Ανάλυση Κώδικα Στο Κεντρικό παράθυρο ( Form1) .....	27
• 2.2 Δημιουργία Πινάκων .....	29
• 2.3 Ανάλυση αρχείου *.dt2.....	
➢ Γεγονός ECN0.....	31
➢ Γεγονός CAD.....	34
➢ Γεγονός CAF.....	35
• 2.4 Εύρεση των Scrambling Codes.....	38
• 2.5 Γραφήματα.....	40
• 2.6 Drop Call Rate.....	66
• 2.7 Γενικά κομμάτια κώδικα στο form1 .....	66
• 2.8 Ανάλυση κώδικα στο form2.....	74
• 2.9 Ανάλυση κώδικα στο form3.....	80
• 2.10 Ανάλυση κώδικα στο usersettings.....	82

### Κεφάλαιο 3 – Λειτουργία του προγράμματος

• 3.1 Εξοικείωση με την εφαρμογή.....	83
• 3.2 Γραφήματα διαδρομής κινητού.....	83
➢ Απεικόνιση διαδρομής με χρωματικό κώδικα.....	83
➢ Απεικόνιση διαδρομής για συγκεκριμένο scrambling code.....	83
• 3.3 Γραφήματα παρεμβολής.....	86
• 3.4 Γραφήματα CAD και CAF.....	87
• 3.5 Γραφήματα μεταβολής ισχύος.....	88
• 3.6 Λοιπά γραφήματα και λειτουργίες.....	89
• 3.7 Μελέτες RSCP και ECN0.....	90
• 3.8 Βάσεις δεδομένων.....	92

<u>Επίλογος</u> .....	93
-----------------------	----

<u>Βιβλιογραφία</u> .....	94
---------------------------	----

## **ΕΙΣΑΓΩΓΗ**

Με το GSM να εξαφανίζεται σιγά σιγά απ το τηλεπικοινωνιακό προσκήνιο της Ελλάδος και την εμφάνιση κινητών τηλεφώνων εφοδιασμένων με πολυμεσικά πρόσθετα τελευταίας τεχνολογίας (κάμερα, wifi, antenna, gps antenna κτλ). Τα κυψελωτά συστήματα τρίτης γενιάς αποτελούν πλέον τα πιο διαδεδομένα στην χώρα μας με το ποσοστό πληθυσμιακής κάλυψης να αγγίζει το 99%. Ειδικότερα σε χώρες της δυτικής Ευρώπης το ποσοστό αυτό ενδέχεται να πλησιάζει και το 100%.

Το γεγονός αυτό μας ώθησε στην εκπόνηση της παρούσας πτυχιακής εργασίας, στην οποία γίνεται μελέτη και ανάλυση δεδομένων τα οποία προκύπτουν από μετρήσεις πάνω στα κυψελωτά συστήματα τρίτης γενιάς. Βασικό κομμάτι της εργασίας μας είναι η υλοποίηση μιας εφαρμογής γραμμένης σε Visual Basic .NET με την βοήθεια της οποίας ο χρήστης μπορεί να πραγματοποιήσει αναλύσεις μέσα από παραγόμενα γραφήματα και τους πίνακες τους οποίους τα τροφοδοτούν με δεδομένα.

Η εφαρμογή αυτή απευθύνεται στον ερασιτέχνη αλλά και στον επαγγελματία μηχανικό και θα μπορούσε κάλλιστα να συγκριθεί με αντίστοιχες ακριβές εμπορικές εφαρμογές . Επιπρόσθετα θα μπορούσε να αποτελέσει σημαντικό εργαστηριακό βοήθημα σε φοιτητές του κλάδου της Ηλεκτρονικής και των Τηλεπικοινωνιών.

## **Λίγα λόγια για την Anite plc**

Η παρούσα πτυχιακή εργασία σχετίζεται άμεσα με την εταιρία Nemo Technologies ( πρόσφατα μετονομάστηκε σε Anite plc) η οποία παρέχει λύσεις πληροφορικής κυρίως στον τομέα των τηλεπικοινωνιών και συγκεκριμένα των ασύρματων. Με έδρα το Ηνωμένο Βασίλειο και έχοντας εισαχθεί στο Χρηματιστήριο του Λονδίνου απασχολεί περίπου 1300 άτομα σε 10 διαφορετικές χώρες της Αμερικής, της Ευρώπης και της Ανατολικής Ασίας.

Η Anite παρέχει ένα περιεκτικό εύρος επαγγελματικών λύσεων σε οργανισμούς τόσο του ιδιωτικού όσο και του δημόσιου τομέα. Αναπτύσσει λογισμικά, γνωμοδοτεί, ολοκληρώνει συστήματα και υπηρεσίες για να διασφαλίσει πως οι πελάτες της λειτουργούν με αποτελεσματικότητα και με ασφάλεια. Κάνοντας χρήση τεχνολογιών αιχμής η Anite επιτυγχάνει να προσφέρει ποιότητα σε καλή τιμή με προϋποθέσεις που έχει ορίσει ο πελάτης.

Έχοντας κατανοήσει εις βάθος τις αγορές των πελατών της είναι σε θέση να παραδώσει σε κορυφαίους τηλεπικοινωνιακούς φορείς και κατασκευαστές λογισμικό το οποίο προορίζεται για χρήση σε κινητά τηλέφωνα, λογισμικό που έχει σκοπό την μελέτη διαφόρων τύπων ασυρμάτων δικτύων καθώς και συστήματα κρατήσεων για ταξιδιωτικά γραφεία. Αξίζει να σημειωθεί επίσης η συνεργασία της με δημόσιους φορείς με σκοπό την παροχή υπηρεσιών στους πολίτες. Η ομάδα της Anite εστιάζει στο να κάνει την διαφορά στον χώρο των τηλεπικοινωνιών και στην υπέρβαση κάθε προσδοκίας!

Στην συνέχεια παρουσιάζονται μερικά από τα βασικά προϊόντα της εταιρίας που έχουν σχέση με την μελέτη δικτύων.

- Nemo Q
- Nemo Handy
- Nemo Outdoor
- Nemo Autonomus
- Nemo Analyze

## Nemo Q

### Επισκόπηση

Το Nemo Q είναι μία εφαρμογή βασισμένη στην πλατφόρμα Symbian. Δίνει στην έννοια φορητότητα όλη τη σημασία της καθώς κάθε μηχανικός μπορεί πλέον με το κινητό του τηλέφωνο να λύσει προβλήματα που αφορούν ασύρματα δίκτυα διαφόρων τεχνολογιών. Υποστηρίζει μετρήσεις στα δίκτυα GSM, GPRS, EDGE, WCDMA και HSPDA.

Το Nemo Q είναι εύκολο στην χρήση και παράλληλα αποτελεσματικό στην παροχή λύσεων σε προβλήματα δικτύου. Καθώς το πρόγραμμα τρέχει στο προσκήνιο, ο χρήστης μπορεί να χρησιμοποιήσει το τηλέφωνο του κανονικά. Δηλαδή μπορεί να συνομιλεί, να στέλνει μηνύματα πολυμέσων, να 'σερφάρει' στο διαδίκτυο και να πραγματοποιεί βίντεο-κλήσεις τη στιγμή που το Nemo Q, με το πάτημα ενός κουμπιού, είναι έτοιμο να καταγράψει οποιοδήποτε πρόβλημα σε ένα αρχείο κειμένου και να το προωθήσει στον πάροχο δικτύου.

Την παρούσα στιγμή η εφαρμογή είναι διαθέσιμη σε 3 μοντέλα κινητών τηλεφώνων της NOKIA. Πρόκειται για τα N80, N95 και N75.

### Nemo Q και τελικός χρήστης

Ο χρήστης δεν χρειάζεται να δώσει ιδιαίτερη προσοχή κατά την χρήση της εφαρμογής καθώς αυτή τρέχει στο παρασκήνιο. Η διαδικασία για την αποστολή λαθών στον φορέα αποτελείται από 3 στάδια.

1. Παρουσιάζεται το πρόβλημα, π.χ. Drop Call, και ο χρήστης επιλέγει Q στο μενού του προγράμματος.
2. Ο χρήστης επιλέγει το κουμπί Capture και επιλέγει το Send log files.
3. Τέλος επιλέγει το επιθυμητό access point και προωθεί τα αρχεία στον φορέα δικτύου.

### Nemo Q και φορέας δικτύου

Τα τεμαχικά θα αποστείλουν το log file αυτομάτως στον εξυπηρετητή που τους έχει αντιστοιχηθεί. Τα αρχεία αυτά περιέχουν δεδομένα όσον αφορά το δίκτυο την στιγμή του προβλήματος αλλά και 5 λεπτά πριν αυτό. Μετά την αποστολή των αρχείων στον ftp server είναι δυνατό να ανακτηθούν και να επεξεργαστούν με

προγράμματα που υποστηρίζουν το format του Nemo File όπως το Nemo Analyze και το Nemo Outdoor.

## **Nemo Handy**

### **Επισκόπηση**

Το Nemo Handy είναι όπως και το παραπάνω μία εφαρμογή βασισμένη στην πλατφόρμα Symbian που σκοπό έχει τον έλεγχο ποιότητας φωνής στα ασύρματα δίκτυα GSM, GPRS, EDGE, WCDMA, HSPDA και WiFi 802.11b/g καθώς και ο έλεγχος του QoS\* και του QoE\*. Υποστηρίζει έλεγχο ποιότητας φωνής σε κανονικές αλλά και σε βίντεο-κλήσεις, έλεγχο για τη σωστή λειτουργία του πρωτοκόλλου WAP, έλεγχο της υπηρεσίας SMS και MMS και τέλος έλεγχο των συνδέσεων ftp και http. Η εφαρμογή, όπως και το Nemo Q, τρέχει στα κινητά τηλέφωνα N80, N75 και N95.

Όπως λέει και το όνομά του το Nemo Handy είναι ένα χρήσιμο εργαλείο για κάθε μηχανικό δικτύου. Πραγματοποιεί μετρήσεις σε κτίρια και τοποθεσίες όπου είναι δύσκολη και ασύμφορη η χρήση ογκώδους εξοπλισμού. Επίσης μπορεί να συνδυαστεί με δέκτες GPS της αγοράς για την συγκέντρωση γεωγραφικών συντεταγμένων.

Στην συνέχεια παραθέτουμε τις τρεις εναλλακτικές εκδόσεις του Nemo Handy καθώς και μερικά παράθυρα λειτουργίας.

### **Handy Field Test**

Περιλαμβάνει όλες τις διαθέσιμες επιλογές γραφημάτων χωρίς όμως να έχει την δυνατότητα καταγραφής δεδομένων και υποστήριξη για GPS. Αποτελεί ιδανικό εργαλείο για τεχνικούς.

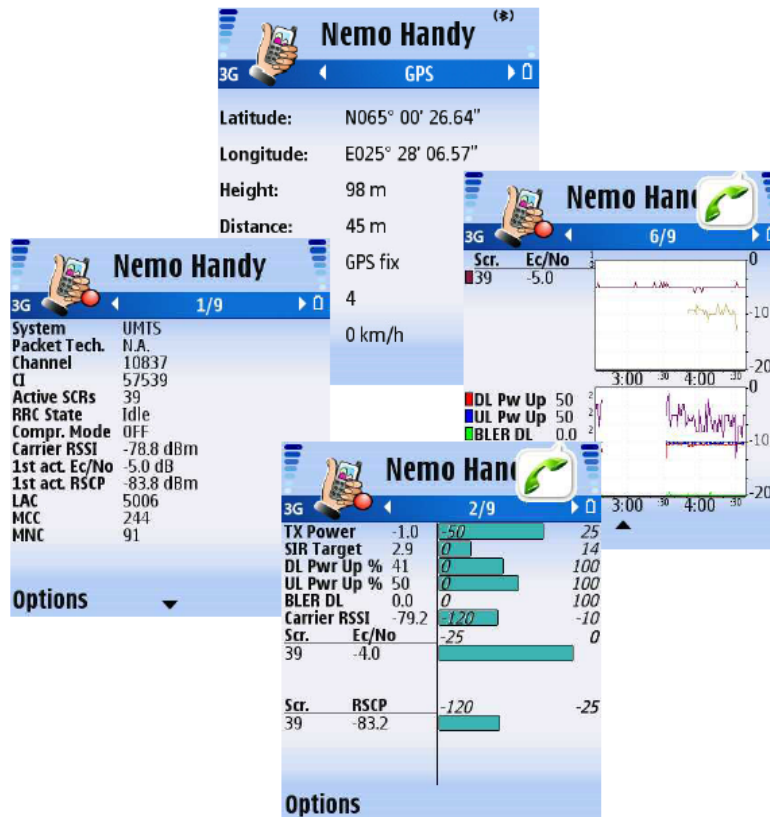
### **Handy**

Εκτός από το γραφικό περιβάλλον αυτή η έκδοση εισάγει την καταγραφή δεδομένων και την χρήση GPS. Παρολαυτά το scripting και άλλες προχωρημένες λειτουργίες λείπουν από την έκδοση αυτή η οποία απευθύνεται σε αυτούς που θέλουν απλά να τεστάρουν ή να λύσουν περιστασιακά προβλήματα.

### **Handy Professional**

Τέλος έχουμε το Handy στην πλήρη του μορφή όπου πλέον έχουμε στην διάθεσή μας εργαλεία scripting, δυνατότητα αυτόματης μεταφοράς αρχείων και την δυνατότητα

χρήσης εσωτερικού χάρτη. Ως εκ τούτου το Handy Professional είναι ιδανικό και για εξεύρεση λύσεων αλλά και για ελαφρύ έλεγχο.



σχ. 1.1

## Nemo Outdoor

### Επισκόπηση

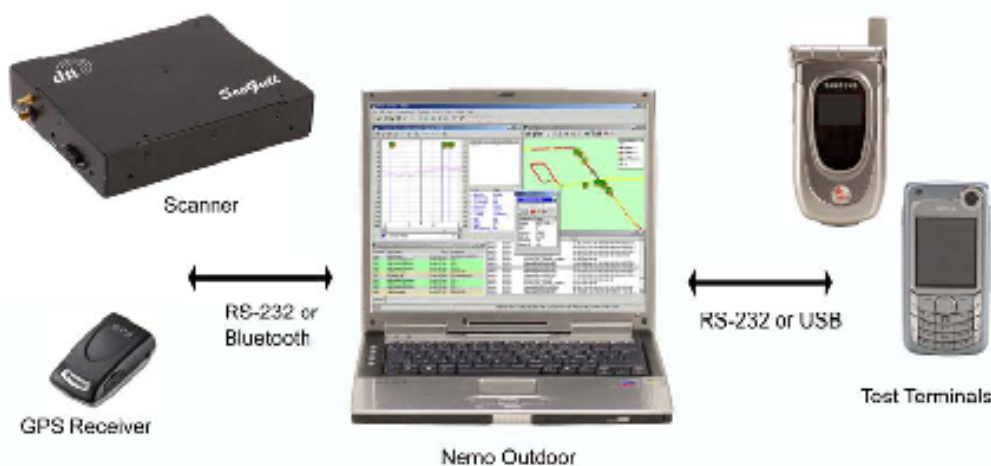
Το Nemo Outdoor είναι ένα υπερβολικά «εύκαμπτο» και φορητό εργαλείο για την μέτρηση και παρακολούθηση ασύρματων δικτύων. Φορείς δικτύου με το Nemo outdoor έχουν την δυνατότητα να πραγματοποιούν μετρήσεις και δοκιμές όχι μόνο σε πολλαπλά δίκτυα αλλά και σε διαφορετικές τεχνολογίες την ίδια στιγμή. Υποστηρίζει μετρήσεις στα δίκτυα TETRA, TDMA, AMPS, cdmaOne, GSM, GPRS, EDGE, WCDMA, CDMA2000, HSPDA, HSPDA 16QAM, HSUPA, TD-SCDMA και τέλος σε δίκτυα UMA.

Παρά το όνομά του, το Nemo Outdoor είναι εξίσου ικανό και για εσωτερικές μετρήσεις. Συνοδεύεται από ένα tablet pc με το οποίο ο χρήστης μπορεί να κάνει

οτιδήποτε μ ένα ειδικό στυλό. Υποστηρίζει πληθώρα νέων συσκευών τόσο στον τομέα των κινητών τηλεφώνων όσο και στον τομέα των scanners.

### Δουλεύοντας με το Nemo Outdoor

Το σύστημα μετρήσεων Nemo Outdoor αποτελείται από ένα υπολογιστή ο οποίος έχει εγκατεστημένα τα Windows XP, κινητά τηλέφωνα ή σαρωτές συχνοτήτων( ή ακόμα και τον συνδυασμό τους), δέκτη GPS καθώς και όλα τα απαραίτητα καλώδια που χρειάζονται για την σύνδεση των παραπάνω εξαρτημάτων.



σχ. 1.2

- Έχει την δυνατότητα να καταγράφει διάφορες παραμέτρους μετρήσεων φωνής όπως είναι η ισχύς εκπομπής του κινητού τερματικού, την κλάση της ποιότητας αλλά και το B(it) E(rror) R(ate).
- Καταγράφει μετρήσεις που σχετίζονται με το πρωτόκολλο μετάδοσης ψηφιακού βίντεο από φορητές συσκευές (DVB-H), πρωτόκολλο το οποίο επιτρέπει για παράδειγμα την παρακολούθηση τηλεοπτικών προγραμμάτων απ το κινητό μας τηλέφωνο.
- Επιτρέπει την πραγματοποίηση μετρήσεων σε εσωτερικούς χώρους. Ο χρήστης κάνοντας χρήση ενός table pc αντί ενός Laptop έχει την δυνατότητα να πραγματοποιεί μετρήσεις πολύ απλά και γρήγορα με τη βοήθεια του ειδικού στυλό.
- Επιτρέπει επίσης την πολλαπλή καταγραφή μετρήσεων έχοντας συνδεδεμένα 5 κινητά τηλέφωνα, ίδιας ή διαφορετικής τεχνολογίας, έναν δέκτη GPS και πολλαπλούς σαρωτές.



- Είναι αξιοσημείωτο ότι το Nemo Outdoor αναγνωρίζει αυτόματα οποιαδήποτε συσκευή συνδεθεί στον φορητό Η/Υ και συνεργάζεται άψογα με αυτή.

## Nemo Autonomus

### Επισκόπηση

Το Nemo Autonomus αποτελεί την ύψιστη λύση για αυτοματοποιημένες μεγάλης κλίμακας μετρήσεις σε ένα ασύρματο δίκτυο. Επιτρέπει την απομακρυσμένη διαχείριση εξοπλισμού είτε σε καθορισμένες τοποθεσίες όπως π.χ. κτίρια είτε σε οχήματα εν κινήσει. Οι μονάδες είναι προγραμματισμένες να κάνουν επανεκκίνηση σε περίπτωση αποτυχίας είτε του λογισμικού είτε του τηλεφώνου. Οι μονάδες αφού πραγματοποιήσουν τις προκαθορισμένες από χρήστη μετρήσεις αποστέλλουν τα log files σ έναν κεντρικό server για ανάλυση από το Nemo Analyze. Τέλος κάθε χρήστης έχει την δυνατότητα παρακολούθησης έως και 100 κινητών και ακίνητων μονάδων επισπεύδοντας σημαντικά τον χρόνο που απαιτείται για την λύση ενός προβλήματος. Η εφαρμογή αποτελείται από το **Nemo Wardener** (μονάδα μετρήσεων πεδίου), το **Nemo Analyze** και το **Nemo Server**.



σχ. 1.3

### Nemo Wardener

Το Nemo Wardener είναι στην ουσία ένα κουτί το οποίο περιέχει μέσα ένα συμβατό κινητό τηλέφωνο βασιζόμενο στην πλατφόρμα Symbian και τοποθετείτε συνήθως σε κάποιον τοίχο. Είναι ικανό για μετρήσεις RF δεδομένων και για σηματοδότηση. Αφού τοποθετηθεί σε έναν κινητό ή σε έναν ακίνητο σταθμό είναι ικανό να πάρει μετρήσεις χωρίς την παρουσία τεχνικού αν και αυτή ενδέχεται να είναι απαραίτητη περιστασιακά.

### Nemo Analyze

Περιλαμβάνει έναν διαχειριστή «στόλου» (Fleet Manager), ένα Server και την δυνατότητα της μετέπειτα επεξεργασίας των αρχείων μετρήσεων.

Ο διαχειριστής στόλου έχει την δυνατότητα διαχείρισης και ελέγχου όλων των κινητών και ακίνητων μονάδων που πραγματοποιούν μετρήσεις. Η τοποθεσία και η κατάσταση τους φαίνεται σε μια οθόνη. Τα log files που αποστέλλουν οι μονάδες μετρήσεων (Nemo Wardener) αποθηκεύονται σε έναν κεντρικό server και μπορούν να ανακτηθούν ανά πάσα στιγμή για επεξεργασία.

### Nemo Server

Τέλος το Nemo Server είναι το κέντρο επικοινωνίας μεταξύ των διαφορετικών ενοτήτων του Nemo Analyze, των μονάδων μετρήσεων Nemo Wardener και των άλλων εργαλείων όπως το Nemo Outdoor και το Nemo Handy.

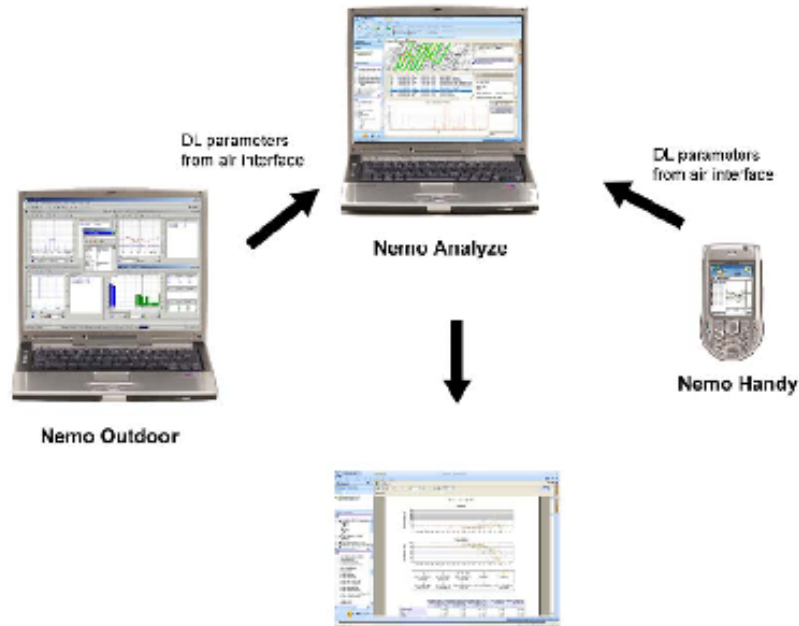
---

## **Nemo Analyze**

### Επισκόπηση

Όπως φανερώνει και το όνομά του το Nemo analyze είναι ένα πολύ σημαντικό εργαλείο ανάλυσης. Ένα από τα πιο δυνατά του σημεία είναι ότι υποστηρίζει όλες τις σημαντικές ασύρματες τεχνολογίες. (TDMA, AMPS, GSM, WCDMA, EDGE, GPRS, cdmaOne, UMA, DVB-H, HSUPA, HSPDA, TD-CDMA, TETRA και CDMA2000).

Αποτελείται από δική του μηχανή βάσης και υποστηρίζει την ταυτόχρονη διαχείριση μέχρι και χιλιάδων αρχείων που περιέχουν μετρήσεις. Η εγκατάσταση του προγράμματος είναι πολύ απλή, η εφαρμογή τρέχει σε ένα κανονικό PC και δεν απαιτεί απ' τον χρήστη να ξέρει οτιδήποτε σχετικά με την μηχανή βάσης δεδομένων. Το Nemo Analyze είναι σχεδιασμένο έτσι ώστε να αναλύει τα δεδομένα που παράγονται από τα προηγούμενα εργαλεία μετρήσεων που αναφέραμε.



σχ. 1.4

### Δουλεύοντας με το Nemo Analyze

Το Nemo Analyze περιλαμβάνει ένα σετ εργαλείων το οποίο βοηθάει στην γρήγορη και αποτελεσματική εξεύρεση λύσεων ψάχνοντας την ρίζα της αιτίας μέσω της ανάλυσης. Υποστηρίζει ανάλυση των κλήσεων ομιλίας και κλήσεων βίντεο. Οι διάφορες αποτυχίες που προκύπτουν σημειώνονται καθώς και οι αιτίες τους. Ένα άλλο χαρακτηριστικό γνώρισμα της εφαρμογής είναι η συγκριτική μέτρηση επιδόσεων (benchmarking). Είναι δυνατή η σύγκριση δεδομένων μεταξύ διαφορετικών παρόχων, τεχνολογιών, κυψελών, αλλά και πολυγωνικών περιοχών. Παρέχει επίσης την δυνατότητα στατιστικής αναφοράς βασιζόμενο στην εφαρμογή Crystal Reports επιτρέποντας την εξαγωγή των αναφορών σε αρχείο pdf ή xls.

Στο Nemo Analyze τα δεδομένα συσσωρεύονται σε μια SQL βάση δεδομένων όπου μπορούν να αποθηκευτούν μεγάλα κομμάτια δεδομένων τα οποία μπορούν να διαχειριστούν και μετέπειτα να επεξεργασθούν. Το μέγεθος της βάσης δεδομένων εξαρτάται από το μέγεθος του σκληρού δίσκου.

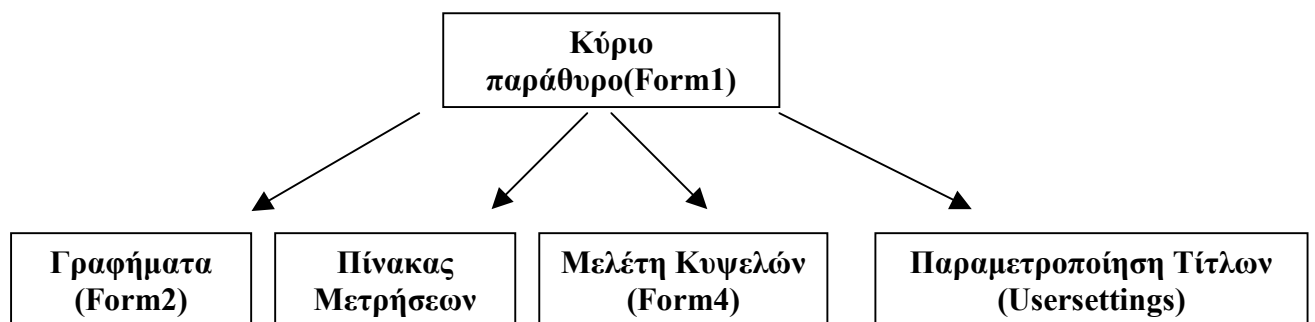


Οι λειτουργίες του προγράμματος μας είναι οι εξής :

1. **Ανάλυση των αρχείων \*.dt2 και εισαγωγή δεδομένων σε πίνακες.**
2. **Δημιουργία γραφημάτων.**
3. **Αποθήκευση γραφημάτων.**
4. **Αποθήκευση των πινάκων που έχει δημιουργήσει το πρόγραμμα σε βάση δεδομένων.**
5. **Εισαγωγή βάσης δεδομένων.**
6. **Δυνατότητα εκτύπωσης γραφημάτων.**

Το πρόγραμμα αποτελείται από 5 φόρμες:

- **Κύριο Παράθυρο**
- **Γραφήματα**
- **Πίνακες Μετρήσεων**
- **Μελέτη Κυψελών**
- **Παραμετροποίηση Τίτλων**

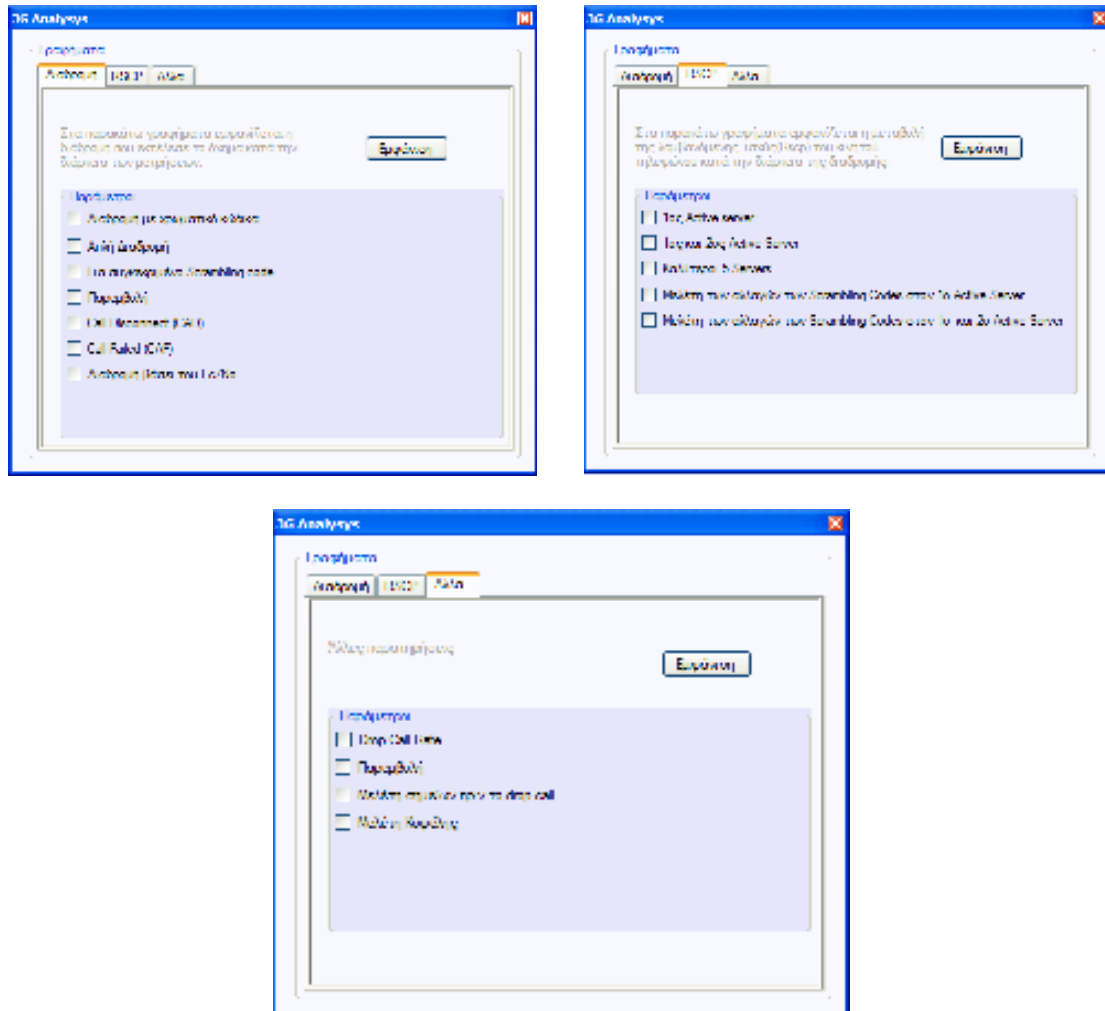


Σχ. 1.6

*Κύριο Παράθυρο* : Το κύριο παράθυρο είναι το πρώτο παράθυρο που βλέπουμε όταν τρέξουμε το πρόγραμμα μας. Εκεί βρίσκονται όλα τα εργαλεία που χρειαζόμαστε για να το λειτουργήσουμε. Σε αυτό θα απεικονίζονται οι γραφικές παραστάσεις που θα δημιουργούμε καθώς και κάποια μενού επιλογών.

Το κύριο παράθυρο έχει οριστεί στον κώδικα του προγράμματος ως Form1.

*Γραφήματα* : Στην φόρμα αυτή μας δίνεται η δυνατότητα να επιλέξουμε την γραφική παράσταση που θέλουμε να μελετήσουμε. Για την εμφάνιση του μενού γραφήματα επιλέγουμε από το κύριο παράθυρο : Tools → Γραφήματα. Το μενού Γραφήματα έχει οριστεί στον κώδικα του προγράμματος ως Form2.



σχ. 1.7

Στο menu Γραφήματα μπορούμε να επιλέξουμε ανάμεσα σε τρία διαφορετικά tabs, ανάλογα με αυτό που θέλουμε να μελετήσουμε.

## Διαδρομή

### 1. Διαδρομή με χρωματικό κώδικα.

Στο γράφημα αυτό παρατηρούμε την διαδρομή που ακολούθησε το όχημα κατά την διάρκεια των μετρήσεων καθώς και την ισχύ (σύμφωνα με τον χρωματικό κώδικα) που λάμβανε το κινητό τηλέφωνο.

## 2. Απλή διαδρομή.

Στο γράφημα αυτό παρατηρούμε απλά την διαδρομή που ακολούθησε το όχημα κατά την διάρκεια των μετρήσεων.

## 3. Για συγκεκριμένο scrambling code.

Στο γράφημα αυτό παρατηρούμε τα σημεία της διαδρομής στα οποία το κινητό τηλέφωνο λάμβανε σήμα από το scrambling code που εμείς έχουμε επιλέξει. (ουσιαστικά μελετούμε τα σημεία που καλύπτει μια κυψέλη καθώς και την ισχύ που λαμβάνεται σε αυτά τα σημεία.)

## 4. Παρεμβολή.

Στο γράφημα αυτό παρατηρούμε τα σημεία της διαδρομής στα οποία έχουμε παρεμβολή. Θεωρούμε ότι υπάρχει παρεμβολή όταν έχουμε μία καλή τιμή ισχύος ( $R_{scp}$ ) και μία κακή τιμή του  $ECN_0$ . Από default αυτές οι τιμές έχουν οριστεί ως εξής :  $rscp > -86$  και  $ECN_0 < -11$ . Στο πρόγραμμα υπάρχει η δυνατότητα αλλαγής των παραπάνω τιμών.

## 5. Call Disconnect.

Στο γράφημα αυτό παρατηρούμε τα σημεία της διαδρομής στα οποία έγινε διακοπή της κλήσης για κάποιο λόγο (βλέπε σελ 22 ). (Είτε από κανονική διακοπή είτε από απόρριψη από το δίκτυο).

## 6. Call Failed.

Στο γράφημα αυτό παρατηρούμε τα σημεία της διαδρομής στα οποία δεν μπόρεσε να ξεκινήσει η κλήση για κάποιο λόγο (βλέπε σελ 23 ).

## 7. Διαδρομή Βάσει του Ec/No.

Στο γράφημα αυτό παρατηρούμε την διαδρομή που ακολούθησε το όχημα κατά την διάρκεια των μετρήσεων τονίζοντας όμως και την τιμή του Ec/No.

# RSCP

### 1. 1<sup>ος</sup> Active server.

Στο γράφημα αυτό παρατηρούμε την μεταβολή της ισχύος του πρώτου Active server κατά την διάρκεια των μετρήσεων.

## 2. 1<sup>ος</sup> και 2<sup>ος</sup> Active server.

Στο γράφημα αυτό παρατηρούμε την μεταβολή της ισχύος όλων των Active servers κατά την διάρκεια των μετρήσεων.

## 3. Καλύτεροι 5 servers.

Στο γράφημα αυτό παρατηρούμε την μεταβολή της ισχύος των 5 καλύτερων servers (ανάμεσα σε Active, Monitored, Detected και Undetected) κάθε μέτρησης κατά την διάρκεια των μετρήσεων.

## 4. Μελέτη των αλλαγών των Scrambling Codes στον 1ο Active Server.

Στο γράφημα αυτό παρατηρούμε την μεταβολή της ισχύος του πρώτου Active server καθώς και τις αλλαγές (με κόκκινο χρώμα) των scrambling codes κατά την διάρκεια των μετρήσεων.

## 5. Μελέτη των αλλαγών των Scrambling Codes στον 1ο και 2ο Active Server.

Στο γράφημα αυτό παρατηρούμε την μεταβολή της ισχύος του πρώτου και δεύτερου Active server καθώς και τις αλλαγές (με κόκκινο χρώμα) των scrambling codes κατά την διάρκεια των μετρήσεων.

# ΑΛΛΑ

## 1. Drop Call Rate

Είναι το ποσοστό επί τοις εκατό των κλήσεων που απορρίφθηκαν από το δίκτυο.

## 2. Παρεμβολή

Στο γράφημα αυτό παρατηρούμε την ισχύ των σημείων που μετρήθηκαν σε σχέση με την τιμή του ECN0 στα αντίστοιχα σημεία.

## 3. Μελέτη σημείων πριν το drop call

Στην επιλογή αυτή μας δίνεται η δυνατότητα της μελέτης των σημείων πριν από το drop call ώστε να μπορέσουμε να προσεγγίσουμε το πρόβλημα των απορριφθέντων κλήσεων.



#### 4. Μελέτη Κυψέλης.

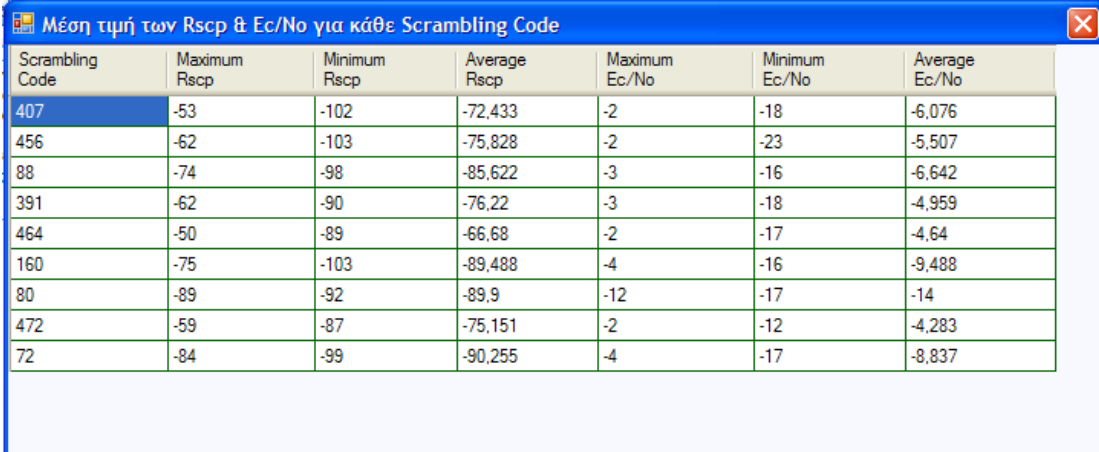
Τέλος, αυτή η γραφική παρουσιάζει την μέγιστη, ελάχιστη και μέση τιμή των Rscp και Ec/No για την κυψέλη που εμείς επιλέξουμε.

*Πίνακες Μετρήσεων* : Στην φόρμα αυτή παρουσιάζονται οι πίνακες που δημιουργεί το πρόγραμμα μας κατά την ανάλυση των μετρήσεων. Σε αυτή την φόρμα υπάρχει μενού επιλογών ώστε να μπορεί ο χρήστης να επιλέξει μεμονωμένα τον πίνακα που τον ενδιαφέρει. Τα δεδομένα που περιέχουν οι πίνακες χρησιμοποιούνται για την δημιουργία των γραφικών παραστάσεων. Για την εμφάνιση των Πινάκων μετρήσεων επιλέγουμε από το κύριο παράθυρο : Tools → Πίνακες Μετρήσεων. Το μενού Πίνακας Μετρήσεων έχει οριστεί στον κώδικα του προγράμματος ως Form3.

id	Status	Cellular number	Code	Ec/No	STTD	RSCP
1	Active	11111	10	-10	1	-101
1	Monitored	11111	11	-11	0	-101
1	Monitored	11111	150	-15	0	-102
1	Monitored	11111	10	-12	0	-101
1	Monitored	11111	111	-12	0	-101
2	Active	11111	10	-10	1	-101
2	Monitored	11111	11	-11	0	-102
2	Monitored	11111	150	-15	0	-102
2	Monitored	11111	10	-16	0	-101
2	Monitored	11111	111	-18	0	-101
3	Active	11111	10	-10	1	-101
3	Monitored	11111	11	-11	0	-102
3	Monitored	11111	150	-15	0	-102
3	Monitored	11111	10	-12	0	-101
3	Monitored	11111	111	-18	0	-101
4	Active	11111	10	-10	0	-101
4	Monitored	11111	11	-11	0	-101

σχ. 1.8

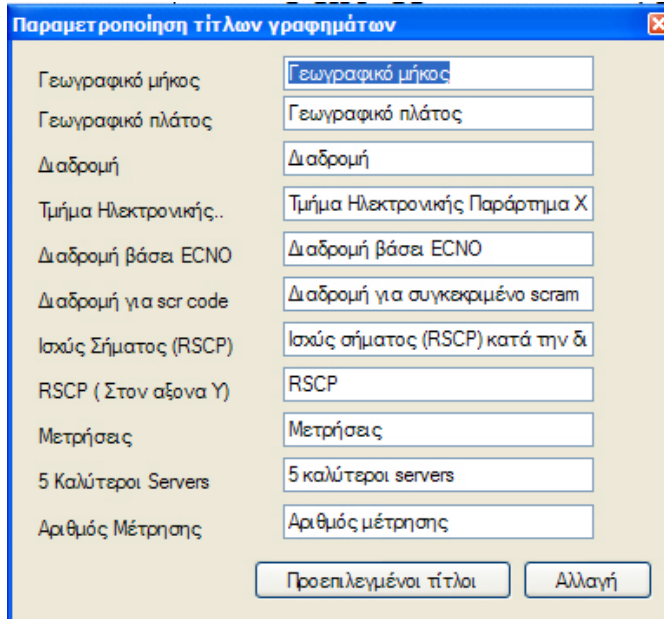
*Μελέτη Κυψελών* : Στη φόρμα αυτή παρουσιάζεται ένας πίνακας ο οποίος μας δίνει κάποια στατιστικά στοιχεία για τις τιμές των Rscp και Ec/No σε κάθε κυψέλη ξεχωριστά. Οι πληροφορίες αυτές είναι η μέγιστη, η ελάχιστη και η μέση τιμή των παραπάνω στοιχείων. Τα δεδομένα που περιέχει ο πίνακας χρησιμοποιούνται για την δημιουργία της γραφικής παράστασης “Μελέτη Κυψέλης”. Για την εμφάνιση του πίνακα επιλέγουμε από το κύριο παράθυρο : Tools →Μελέτη Κυψελών. Το μενού Μελέτη Κυψελών έχει οριστεί στον κώδικα του προγράμματος ως Form4.



Scrambling Code	Maximum Rscp	Minimum Rscp	Average Rscp	Maximum Ec/No	Minimum Ec/No	Average Ec/No
407	-53	-102	-72,433	-2	-18	-6,076
456	-62	-103	-75,828	-2	-23	-5,507
88	-74	-98	-85,622	-3	-16	-6,642
391	-62	-90	-76,22	-3	-18	-4,959
464	-50	-89	-66,68	-2	-17	-4,64
160	-75	-103	-89,488	-4	-16	-9,488
80	-89	-92	-89,9	-12	-17	-14
472	-59	-87	-75,151	-2	-12	-4,283
72	-84	-99	-90,255	-4	-17	-8,837

σχ. 1.9

*Παραμετροποίηση Τίτλων* : Στη φόρμα αυτή ο χρήστης μπορεί εύκολα αν το επιθυμεί να δώσει νέες ονομασίες στις λεζάντες των γραφημάτων. (π.χ. μετονομασία στην αγγλική γλώσσα).



Γεωγραφικό μήκος	Γεωγραφικό μήκος
Γεωγραφικό πλάτος	Γεωγραφικό πλάτος
Διαδρομή	Διαδρομή
Τμήμα Ηλεκτρονικής..	Τμήμα Ηλεκτρονικής Παράρτημα Χ
Διαδρομή βάσει ECNO	Διαδρομή βάσει ECNO
Διαδρομή για scr code	Διαδρομή για συγκεκριμένο scram
Ισχύς Σήματος (RSCP)	Ισχύς σήματος (RSCP) κατά την δι
RSCP ( Στον άξονα Υ)	RSCP
Μετρήσεις	Μετρήσεις
5 Καλύτεροι Servers	5 καλύτεροι servers
Αριθμός Μέτρησης	Αριθμός μέτρησης

Προεπιλεγμένοι τίτλοι      Αλλαγή

σχ. 1.10

## Αρχεία Μετρήσεων

Τα αρχεία που περιέχουν τις μετρήσεις και τα οποία βασιστήκαμε για να φτιάξουμε το πρόγραμμα μας έχουν κατάληξη \*.dt2. Ουσιαστικά είναι αρχεία κειμένου τα οποία μπορούν να διαβαστούν από οποιονδήποτε επεξεργαστή κειμένου. Παρακάτω μπορείτε να παρατηρήσετε ένα κομμάτι ενός τέτοιου αρχείου.

```

////////////////////////////////////
ECN0 23.634943 37.932362 55 219 1 6 5 16:29:13.907 12 1 10764 -87 14 1 10764
464 -4 0 -91 -1 255 255 255 255 255 255 -1 -1.0 4 10764 90 -16 0 -103 -1 255 255
255 255 255 255 -1 -1.0 10764 40 -16 0 -103 -1 255 255 255 255 255 255 -1 -1.0
10764 88 -22 0 -109 -1 255 255 255 255 255 255 -1 -1.0 10764 391 -25 0 -112 -1 255
255 255 255 255 255 -1 -1.0 1 10764 130 -22 1 -109 -1 255 255 255 255 255 255 -1 -
1.0 0
RXPC 23.634943 37.932362 55 219 1 6 5 16:29:13.907 12 255 255 2.3 255.0 0 412
1088 27.5 -1
TXPC 23.634943 37.932362 55 219 1 6 5 16:29:13.907 12 -12 0 1 0 751 749 50.1
BER 23.634943 37.932362 55 219 1 6 5 16:29:13.907 12 1.3 0.0
ECN0 23.634962 37.932346 55 221 1 6 11 16:29:14.906 12 1 10764 -84 14 1 10764
464 -3 0 -87 -1 255 255 255 255 255 255 -1 -1.0 4 10764 90 -17 0 -101 -1 255 255
255 255 255 255 -1 -1.0 10764 40 -20 0 -104 -1 255 255 255 255 255 255 -1 -1.0
10764 88 -25 0 -109 -1 255 255 255 255 255 255 -1 -1.0 10764 391 -25 0 -109 -1 255
255 255 255 255 255 -1 -1.0 1 10764 130 -23 1 -107 -1 255 255 255 255 255 255 -1 -
1.0 0
RXPC 23.634962 37.932346 55 221 1 6 11 16:29:14.906 12 255 255 2.3 255.0 0 27
1473 1.8 -1
TXPC 23.634962 37.932346 55 221 1 6 11 16:29:14.906 12 -14 0 1 0 751 749 50.1
PHDAS 23.634962 37.932346 55 221 1 6 11 16:29:14.906 12 22200
BER 23.634962 37.932346 55 221 1 6 11 16:29:14.906 12 0.0 0.0
CAD 23.634962 37.932346 55 221 1 6 11 16:29:15.283 1 00:01:14 -1
CAF 23.632915 37.929775 46 700 1 6 12 16:31:12.306 9 10 -1
////////////////////////////////////

```

### Παράδειγμα 1.1

Κάθε αρχείο της παραπάνω μορφής μπορεί να αποτελείται από δεκάδες έως χιλιάδες μετρήσεις. Κάθε γραμμή είναι μια μέτρηση ενός γεγονότος κάποια συγκεκριμένη στιγμή. Στην αρχή κάθε γραμμής μπορούμε να παρατηρήσουμε έναν κωδικό. Ουσιαστικά αυτός ο κωδικός είναι μια συντομογραφία η οποία μας δηλώνει το γεγονός στο οποίο αναφέρεται η γραμμή. Κατά την διάρκεια των μετρήσεων έχουν μελετηθεί πολλών ειδών γεγονότα. Αυτά όμως που μας ενδιαφέρουν φαίνονται στο παράδειγμα 1.1 με έντονο κόκκινο χρώμα και είναι τα εξής : **ECN0** , **CAD** και **CAF**.

Για να κατανοήσουμε πλήρως την λογική που δημιουργήθηκε το πρόγραμμα πρέπει πρώτα να έχουμε καταλάβει την δομή ενός αρχείου μετρήσεων. Οι πρώτοι οχτώ

παράμετροι που βρίσκονται μετά την συντομογραφία είναι ίδιοι για όλα τα γεγονότα. Για καλύτερη κατανόηση αυτών των παραμέτρων ας δούμε το παράδειγμα που ακολουθεί :

**CAF 23,632915 37,929775 46 700 1 6 12 16:31:12.306 9 10 -1**

*Παράδειγμα 1.2*

- **CAF** : Είναι το γεγονός στο οποίο αναφέρονται οι μετρήσεις που θα ακολουθήσουν. Το γεγονός CAF είναι τα αρχικά των λέξεων **CALL FAILED**.
- **23,632915** : Το γεωγραφικό μήκος στο οποίο έγινε η μέτρηση.
- **37,929775** : Το γεωγραφικό πλάτος στο οποίο έγινε η μέτρηση.
- **46** : Το γεωγραφικό ύψος στο οποίο έγινε η μέτρηση.
- **700** : Η απόσταση από το αρχικό σημείο των μετρήσεων (σε μέτρα).
- **1** : Ένδειξη GPS
- **6** : Πλήθος διαθέσιμων δορυφόρων
- **12** : Η ταχύτητα του οχήματος που κάνει τις μετρήσεις (σε χιλιόμετρα).
- **16:31:12.306** : Η ώρα την στιγμή της μέτρησης στην ακόλουθη μορφή

[Ωρα]:[Λεπτό]:[Δευτερόλεπτο].[Χιλιοστό δευτερολέπτου].

Όλοι οι παράμετροι που βρίσκονται μετά την μέτρηση της ώρας διαφοροποιούνται για κάθε γεγονός. Όλα τα γεγονότα και οτιδήποτε έχει σχέση με τα περιεχόμενα των αρχείων \*.dt2 αναλύονται πλήρως στο εγχειρίδιο που έχει δημιουργήσει η εταιρεία Nemo Technologies.

Εμείς θα αναλύσουμε μόνο τα γεγονότα που μας ενδιαφέρουν.

## Ανάλυση γεγονότων

**Γεγονός ECNO :** Το ECNO είναι μία παράμετρος που μας δείχνει την διαφορά της ισχύς του φέροντος σήματος με την λαμβανόμενη ισχύ της κυψέλης που μελετάμε.

Μία μέτρηση ECNO έχει την παρακάτω μορφή :

```

////////////////////////////////////
ECNO 23.634943 37.932362 55 219 1 6 5 16:29:13.907 12 1 10764 -87 14 1 10764
464 -4 0 -91 -1 255 255 255 255 255 255 -1 -1.0 4 10764 90 -16 0 -103 -1 255 255
255 255 255 255 -1 -1.0 10764 40 -16 0 -103 -1 255 255 255 255 255 255 -1 -1.0
10764 88 -22 0 -109 -1 255 255 255 255 255 255 -1 -1.0 10764 391 -25 0 -112 -1 255
255 255 255 255 255 -1 -1.0 1 10764 130 -22 1 -109 -1 255 255 255 255 255 255 -1 -
1.0 0
////////////////////////////////////

```

Παράδειγμα 1.3

Τα πρώτα εννέα στοιχεία(κόκκινο χρώμα) έχουν σχολιαστεί παραπάνω (Παρ. 1.2). Τα επόμενα στοιχεία φαίνονται στον πίνακα που ακολουθεί (Τα στοιχεία με το μπλε χρώμα δεν αναλύονται γιατί μας είναι αδιάφορα).

### Παράμετροι

Όνομα	Περιγραφή	Τιμή στο παράδειγμα
Σύστημα	Cellular system 12 = UMTS FDD 2100 13 = UMTS TDD 2100 20 = UMTS FDD 1900 21 = UMTS TDD 1900	12
#Chs	Πλήθος καναλιών	1
Channel	Αριθμός καναλιού	10764
Carrier RSSI	Ισχύς φέροντος	-87
#Params / cell	Πλήθος παραμέτρων για κάθε κυψέλη	14

Πίνακας 1.1

Τα πρώτα πέντε στοιχεία μας δίνουν πληροφορίες για τον τύπο του συστήματος καθώς και για το φέρων σήμα. Στην συνέχεια δίνονται πληροφορίες για τους Active servers.

Όνομα	Περιγραφή	Τιμή στο παράδειγμα
#Active cells	Πλήθος των Active κυψελών	1
Channel	Ο αριθμός του Active καναλιού	10764
Scrambling code	Ο αριθμός του Scrambling code	464
Ec/No	Η τιμή του Ec/No	-4
STTD	STTD status active set 0 = STTD not active on PCCPCH. 1 = STTD active on PCCPCH	0
RSCP	Η τιμή της ισχύος Κυμαίνεται από -115 έως -25 dBm.	-91

Πίνακας 1.2

Ακολουθούν πληροφορίες για τους Monitored servers.

Όνομα	Περιγραφή	Τιμή στο παράδειγμα
#Monit. cells	Πλήθος των Monitored κυψελών	4
Channel	Ο αριθμός του Monitored καναλιού	10764

Scrambling code	Ο αριθμός του Scrambling code	90
Ec/No	Η τιμή του Ec/No της συγκεκριμένης κυψέλης	-16
STTD	STTD status active set 0 = STTD not active on PCCPCH. 1 = STTD active on PCCPCH	0
RSCP	Η τιμή της ισχύος της συγκεκριμένης κυψέλης Κυμαίνεται από -115 έως -25 dBm.	-103

Πίνακας 1.3

Σε αυτό το σημείο πρέπει να παρατηρήσουμε ότι το πλήθος των monitored κυψελών είναι **τέσσερα**. Έτσι πρέπει να μας δώσει τις πληροφορίες για κάθε κυψέλη ξεχωριστά. Στον παραπάνω πίνακα φαίνονται οι πληροφορίες για την κυψέλη με scrambling code **-16**.

Οι υπόλοιπες (δηλαδή για τα scrambling codes 40, 88 και 391) δίνονται στην συνέχεια της γραμμής με την ίδια ακριβώς δομή (βλέπε μωβ γράμματα στο παράδειγμα 1.3).

Όνομα	Περιγραφή	Τιμή στο παράδειγμα
#Detect. cells	Πλήθος των Detected κυψελών	1
Channel	Ο αριθμός του Detected καναλιού	10764
Scrambling code	Ο αριθμός του Scrambling code	130
Ec/No	Η τιμή του Ec/No της συγκεκριμένης κυψέλης	-22
STTD	STTD status active set 0 = STTD not active on PCCPCH. 1 = STTD active on PCCPCH	1
RSCP	Η τιμή της ισχύς της συγκεκριμένης κυψέλης Κυμαίνεται από -115 έως -25 dBm.	-109

Πίνακας 1.4

Στον πίνακα 1.4 μας δίνονται πληροφορίες για τους detected servers. Ο τελευταίος μας δείχνει το πλήθος των undetected servers όπου στο παράδειγμα μας έχει την τιμή 0.



**Γεγονός CAD:** Το γεγονός CAD είναι τα αρχικά των λέξεων **CALL DISCONNECT**

Μία μέτρηση CAD έχει την παρακάτω μορφή :

**CAD** 23.634962 37.932346 55 221 1 6 11 16:29:15.283 1 00:01:14 -1

### Παράμετροι

Όνομα	Περιγραφή
CS disc. status	<p>CS call disconnect status</p> <p>1 = Normal disconnect</p> <p>2 = Dropped call. (i.e., network release)</p> <p>3 = Dropped out of service.</p> <p>4 = Dropped during handover/handoff/hard handover</p> <p>5 = Reserved</p> <p>6 = Transfer timeout during data or fax test call</p> <p>7 = Protocol error during data or fax test call</p> <p>8 = Transfer failed during data or fax test call</p> <p>9 = Modem connection failure.</p> <p>Dedicated connection already achieved</p> <p>10= Network busy (GSM) When network has not sent ALERT or CONNECT message before it sends DISCONNECT message</p> <p>20= PPP Error</p>
CS call dur.	<p>CS call duration</p> <p>Time (in hh:mm:ss) elapsed from call attempt to return to control channel</p>
CS disc. cause	<p>CS call disconnect cause</p> <p>If call is dropped due to a dial-up connection error (status code 20), the value is an OS RAS cause (see chapter 7.7 for a list of OS RAS cause values).</p> <p>Otherwise CC cause code. See chapter 7.3 for a list of possible causes. (GSM/UMTS). Value is set -1 if cause code is not available.</p>

**Γεγονός CAF** : Το γεγονός CAF είναι τα αρχικά των λέξεων **CALL FAILURE**

Μία μέτρηση CAF έχει την παρακάτω μορφή :

**CAF 23.632915 37.929775 46 700 1 6 12 16:31:12.306 9 10 -1**

### Παράμετροι

Name	Type	Description
CS fail. status	Integer	CS call attempt failure status 1 = timeout before connection 2 = call was released before connection 3 = service not available (call attempt is made when MS is not in coverage area) 4 = incoming call not received 5 = test system failure (rejected in Quality Survey Reports) 6 = SDCCCH blocking (GSM) (MS sends CHANNEL REQUEST, but network does not send IMMEDIATE ASSIGNMENT to the MS during “Call Attempt Timeout”) 7 = TCH blocking (GSM) (network does not send ASSIGNMENT COMMAND during “Call Attempt Timeout”) 8 = RRC connection failed 9 = Radio bearer setup failed 11 = Reserved 20 = PPP error
CS fail. time	Integer	CS call failure time Time (in seconds) elapsed from call attempt or call receiving to failing
CS fail. cause	Integer	CS call failure cause If call fails due to dial-up connection error (failure status 20), the value is an OS RAS cause (see chapter 7.7 for a list of OS RAS cause values). Value is set -1 if cause code is not available.

## Εισαγωγή στην Visual Basic .net

Η **Visual Basic .NET** είναι μια αντικειμενοστραφής γλώσσα προγραμματισμού και αποτελεί την εξέλιξη της Microsoft Visual Basic. Η εισαγωγή της στην κοινότητα των προγραμματιστών δεν την έκανε άμεσα αποδεκτή λόγω της ασυμβατότητας της με προηγούμενες εκδόσεις του προγραμματιστικού πακέτου. Το μεγαλύτερο ποσοστό των προγραμματιστών χρησιμοποιούν για την ανάπτυξη προγραμμάτων σε VB.net την σουίτα προγραμματιστικών εργαλείων Visual Studio .NET. Όπως για όλες τις .net γλώσσες προγραμματισμού προγράμματα γραμμένα σε VB.NET απαιτούν για την εκτέλεσή τους να υπάρχει εγκατεστημένο το πακέτο .net **FRAMEWORK**. Μέχρι και τον Νοέμβριο του 2006 υπάρχουν 3 εκδόσεις της Visual Basic .NET.

### **1. Visual Basic .NET**

Η αυθεντική VB .NET η οποία κυκλοφόρησε μαζί με την Visual# και το ASP .NET και αποτέλεσε την απάντηση της Microsoft στην Java. Παρολαυτά δεν βρήκε μεγάλη ανταπόκριση στην προγραμματιστική κοινότητα.

### **2. Visual Basic .NET 2003**

Κυκλοφόρησε μαζί με το πακέτο 1.1 του .NET Framework. Βελτιωμένη έκδοση σε σύγκριση με τον προκάτοχό της. Σαφώς καλύτερος οδηγός αναβάθμισης της VB, περισσότερη αξιοπιστία και καλύτερη απόδοση του IDE .NET καθώς και η ενσωμάτωση μερικών καινούριων επιλογών.

### **3. Visual Basic 2005**

Η γλώσσα στην οποία υλοποιήσαμε το project της πτυχιακής μας. Σ αυτή την έκδοση της γλώσσας χάνεται το προσωνύμιο .NET από τον τίτλο και

προστίθενται πολλά νέα χαρακτηριστικά σε μια προσπάθεια να τονιστεί η ευκολία που διακατέχει την συγκεκριμένη γλώσσα προγραμματισμού απευθυνόμενη κυρίως στους ερασιτέχνες και «πρωτάρηδες» προγραμματιστές. Μερικά από αυτά τα χαρακτηριστικά είναι:

- «Επεξεργασία και συνέχεια» , πιθανότατα το μεγαλύτερο χαρακτηριστικό που έλειπε επιτρέποντας πλέον την επεξεργασία του κώδικα και άμεση συνέχιση της εκτέλεσης του.
- «Data source binding» επιτρέποντας την εύκολη δημιουργία βάσεων δεδομένων.
- «Just my code» εμφανίζοντας μόνο τον κώδικα που γράψαμε εμείς.
- Εύκολη πρόσβαση σε περιοχές του .NET framework όπου σε διαφορετική περίπτωση θα απαιτούσε αξιοσημείωτο μέγεθος κώδικα για να γίνει.

Η Visual Basic 2005 εισάγει χαρακτηριστικά τα οποία εμφανίζονται για να συμπληρωθούν τα κενά που υπάρχουν μεταξύ της VB 2005 και των άλλων πιο ισχυρών γλωσσών του πακέτου. Η σχέση της με την Visual Basic που γνωρίσαμε παλαιότερα δε μπορεί να προσδιοριστεί με σαφήνεια. Σαφώς και η βασική σύνταξη δεν έχει δει τρομερές αλλαγές αλλά τα νέα στοιχεία που προστίθενται μας κάνουν να μιλάμε για μια νέα δυνατή γλώσσα η οποία αποτελεί την εξέλιξη της ερασιτεχνικής Visual Basic.

## Ανάλυση κώδικα στο Form1

```
Option Strict Off
Option Explicit On
Imports VB = Microsoft.VisualBasic
Imports System.Drawing
Imports System.Drawing.Printing
Imports ZedGraph
Imports System.Windows.Forms
Imports System.Text
```

**Module** mymodule

**Public** gr **As Integer**

Ορίζουμε το στοιχείο gr ως ακέραιο. Το gr δηλώνει το πλήθος των γεγονότων ECN0.

**Public** mytable **As New DataTable**

Ορίζουμε το mytable ως πίνακα ο οποίος θα περιέχει τους υποπίνακες active, monitored, detected και undetected servers.

**Public** general **As New DataTable**

Ορίζουμε το general ως πίνακα ο οποίος θα περιέχει γενικές πληροφορίες (τις πρώτες 8, βλεπε παράδειγμα 1.2) των γεγονότων ECN0.

**Public** general2 **As New DataTable**

Ορίζουμε το general2 ως πίνακα ο οποίος θα περιέχει τα δεδομένα του γεγονότος CAD.

**Public** generalcaf **As New DataTable**

Ορίζουμε το general2 ως πίνακα ο οποίος θα περιέχει τα δεδομένα του γεγονότος CAF.

**Public** avgtable **As New DataTable**

Ορίζουμε το avgtable ως πίνακα ο οποίος θα περιέχει την μέση τιμή του rscp και Ec/N0 για κάθε κυψέλη (scrambling code).

**Public** countscrcodes **As Integer**

Ορίζουμε το countscrcodes ως ακέραιο. Το countscrcodes δηλώνει το πλήθος των Active servers.

**Public** binding **As New BindingSource**

**Public** binding2 **As New BindingSource**

**Public** developing **As Boolean = False**

**Public** meas(2000, 2) **As String** ????

**Public** best **As New DataTable**

Ορίζουμε το best ως πίνακα ο οποίος θα περιέχει τα δεδομένα των πέντε καλύτερων servers κάθε γεγονότος ECN0.

**Public** codes(1000) **As String**

Ορίζουμε το codes(1000) ως πίνακα στήλης χιλίων θέσεων ο οποίος θα περιέχει τα scrambling codes του πρώτου Active server κάθε γεγονότος ECN0.

**Public** allcodes(1000) **As String**

```
Public pd As New PrintDocument
```

Ορίζουμε το pd ως ένα έγγραφο προς εκτύπωση.

```
Public tlplatos As String
Public tlmikos As String
Public tldiadromi As String
Public tltmima As String
Public tldiadromiecno As String
Public tldiadromiscrcode As String
Public tlrscproute As String
Public tlrscp As String
Public tlmetriseis As String
Public tlbest5 As String
Public tlmeasurement As String
```

Ορίζουμε αυτές τις μεταβλητές με σκοπό να αποθηκεύουμε κάποιους τίτλους γραφημάτων.

```
End Module
```

Σε αυτό το σημείο πρέπει να παρατηρήσουμε ότι όλα τα στοιχεία που έχουμε δηλώσει μέχρι τώρα τα έχουμε ορίσει ως public. Αυτό το κάναμε με σκοπό να μπορούμε να χρησιμοποιούμε τα παραπάνω στοιχεία και στις άλλες τρεις φόρμες.

```
Public Class Form1
```

Σε αυτό σημείο δηλώνουμε το Form1.

```
Inherits System.Windows.Forms.Form
```

```
Private Sub DoPrintPreview()
```

```
Dim ppd As PrintPreviewDialog = New PrintPreviewDialog
AddHandler pd.PrintPage, AddressOf Graph_PrintPage
ppd.Document = pd
ppd.Show()
End Sub
```

```
Private Sub Graph_PrintPage(ByVal sender As Object, ByVal e As
PrintPageEventArgs)
```

```
Dim mypane As GraphPane = ZedGraphControl1.GraphPane
mypane.Draw(e.Graphics)
End Sub
```

Το DoPrintPreview σε συνδιασμό με το Graph\_Printpage είναι δύο ρουτίνες οι οποίες μας επιτρέπουν να εκτυπώσουμε τα γραφήματα που έχουμε δημιουργήσει.

## Δημιουργία Πινάκων

**Sub createtables ()**

Συνάρτηση την οποία καλούμε με σκοπό την δημιουργία των πινάκων που δηλώσαμε στην αρχή του κώδικα.

```
Dim db, int, strin As System.Type
db = System.Type.GetType("System.Double")
int = System.Type.GetType("System.Int32")
strin = System.Type.GetType("System.String")
```

Οι διάφορες τιμές με τις οποίες θα γεμίζουν οι πίνακες. Μπορεί να είναι ακαιρέου τύπου πχ ισχύς, τύπου string (ενδεχομένως κάποια λέξη) και με υποδιαστολή πχ γεωγραφικό μήκος και πλάτος.

```
If mytable.Rows.Count <> 0 Then
    mytable.Clear()
End If
If general.Rows.Count <> 0 Then
    general.Clear()
End If
If general2.Rows.Count <> 0 Then
    general2.Clear()
End If
If generalcaf.Rows.Count <> 0 Then
    generalcaf.Clear()
End If
If best.Rows.Count <> 0 Then
    best.Clear()
End If

If avgtable.Rows.Count <> 0 Then
    avgtable.Clear()
End If
```

```
If mytable.Columns.Count = 0 Then
    mytable.TableName = "All"
    mytable.Columns.Add(New DataColumn("Measurement", int))
    mytable.Columns.Add(New DataColumn("Status", strin))
    mytable.Columns.Add(New DataColumn("Chanell number", int))
    mytable.Columns.Add(New DataColumn("Code", int))
    mytable.Columns.Add(New DataColumn("Ec/No", int))
    mytable.Columns.Add(New DataColumn("STTD", int))
    mytable.Columns.Add(New DataColumn("RSCP", int))
End If
```

Έχοντας δημιουργήσει κενούς πίνακες στο προηγούμενο βήμα έρχεται η στιγμή που πρέπει να τους αρχικοποιήσουμε. Το παραπάνω μέρος του κώδικα αναλαμβάνει να τους προσθέσει τα χαρακτηριστικά τους, τα οποία είναι το όνομα του πίνακα, τα ονόματα των στηλών καθώς και τον τύπο του στοιχείου που θα περιέχει κάθε στήλη.

Στον παρακάτω κομμάτι για παράδειγμα δίνουμε στον πίνακα το όνομα **General info** και του προσθέτουμε 11 στήλες. Η ίδια διαδικασία επαναλαμβάνεται για όλους τους πίνακες.

Σε αυτό το σημείο του κώδικα γίνεται έλεγχος αν οι πίνακες είναι άδειοι. Στην περίπτωση που δεν είναι τότε γίνεται ο καθαρισμός τους με την εντολή *table.clear* όπου *table* το όνομα του πίνακα που θέλουμε να αδειάσουμε.

```
If general.Columns.Count = 0 Then
    general.TableName = "Generalinfo"
    general.Columns.Add(New DataColumn("Measurement", int))
    general.Columns.Add(New DataColumn("Longitude", db))
    general.Columns.Add(New DataColumn("Latitude", db))
    general.Columns.Add(New DataColumn("Height", int))
    general.Columns.Add(New DataColumn("Distance", int))
    general.Columns.Add(New DataColumn("GPS Number", int))
    general.Columns.Add(New DataColumn("Velocity", int))
    general.Columns.Add(New DataColumn("Time", strin))
    general.Columns.Add(New DataColumn("A", int))
    general.Columns.Add(New DataColumn("B", int))
    general.Columns.Add(New DataColumn("C", int))
    general.Columns.Add(New DataColumn("D", int))
    general.Columns.Add(New DataColumn("E", int))
    general.Columns.Add(New DataColumn("F", int))
End If

If general2.Columns.Count = 0 Then
    general2.TableName = "Cadgeneral"
    general2.Columns.Add(New DataColumn("Measurement", int))
    general2.Columns.Add(New DataColumn("Longitude", db))
    general2.Columns.Add(New DataColumn("Latitude", db))
    general2.Columns.Add(New DataColumn("Height", int))
    general2.Columns.Add(New DataColumn("Distance", int))
    general2.Columns.Add(New DataColumn("GPS Number", int))
    general2.Columns.Add(New DataColumn("Velocity", int))
    general2.Columns.Add(New DataColumn("Time", strin))
    general2.Columns.Add(New DataColumn("Disconnection Cause",
strin))
    general2.Columns.Add(New DataColumn("Disconnection Status",
strin))
    general2.Columns.Add(New DataColumn("Call Duration", strin))
End If

If generalcaf.Columns.Count = 0 Then
    generalcaf.TableName = "Cafgeneral"
    generalcaf.Columns.Add(New DataColumn("Measurement", int))
    generalcaf.Columns.Add(New DataColumn("Longitude", db))
    generalcaf.Columns.Add(New DataColumn("Latitude", db))
    generalcaf.Columns.Add(New DataColumn("Height", int))
    generalcaf.Columns.Add(New DataColumn("Distance", int))
    generalcaf.Columns.Add(New DataColumn("GPS Number", int))
    generalcaf.Columns.Add(New DataColumn("Velocity", int))
    generalcaf.Columns.Add(New DataColumn("Time", strin))
    generalcaf.Columns.Add(New DataColumn("Failure Status",
strin))
    generalcaf.Columns.Add(New DataColumn("Call failure time",
int))
    generalcaf.Columns.Add(New DataColumn("Disconnection Cause",
strin))
End If

If best.Columns.Count = 0 Then
    best.TableName = "Best5Servers"
    best.Columns.Add(New DataColumn("First", int))
    best.Columns.Add(New DataColumn("Second", int))
    best.Columns.Add(New DataColumn("Third", int))
    best.Columns.Add(New DataColumn("Fourth", int))
    best.Columns.Add(New DataColumn("Fifth", int))
End If
```



```

If Avgtable.Columns.Count = 0 Then
    Avgtable.TableName = "Average Rscp & Ec/No"
    Avgtable.Columns.Add(New DataColumn("Scrambling Code",
int))
    avgtable.Columns.Add(New DataColumn("Maximum Rscp", int))
    avgtable.Columns.Add(New DataColumn("Minimum Rscp", int))
    avgtable.Columns.Add(New DataColumn("Average Rscp", db))
    avgtable.Columns.Add(New DataColumn("Maximum Ec/No",
int))
    avgtable.Columns.Add(New DataColumn("Minimum Ec/No",
int))
    avgtable.Columns.Add(New DataColumn("Average Ec/No", db))
End If
End Sub

```

## Ανάλυση αρχείου \*.dt2 ECN0

```
Sub analyse1()
```

Συνάρτηση την οποία καλούμε με σκοπό την ανάλυση του αρχείου μετρήσεων όσον αφορά τα γεγονότα ECN0.

```
Dim inputdata As String
```

Δηλώνεται η μεταβλητή inputdata ως string.

```
Dim maxMonitor, maxDetected, maxUndetected, strlen, i, j, k As  
Integer
```

Η μεταβλητή maxMonitor αποθηκεύει τον αριθμό των monitored servers που υπάρχουν στο αρχείο που επεξεργαζόμαστε, η maxDetected τον αριθμό των detected servers και η maxUndetected τον αριθμό των undetected servers.

```
gr =0
```

Αρχικοποίηση της μεταβλητής gr. Όπως έχουμε προαναφέρει στην μεταβλητή gr αποθηκεύεται το πλήθος των γεγονότων ECN0(ουσιαστικά ο αριθμός των γραμμών ECN0).

```
Createtables()
```

Καλούμε την συνάρτηση που αρχικοποιεί και δημιουργεί τους πίνακες δεδομένων.

```
Dim Filenames() As String = OpenFileDialog1.FileNames
```

Ορίζουμε ένα νέο file dialog το οποίο μας επιτρέπει να ανοίγουμε αρχεία δεδομένων προς επεξεργασία.

```
For j = 0 To Filenames.Length - 1
```

Ξεκινάει μία διαδικασία η οποία επαναλαμβάνεται τόσες φορές όσες είναι και οι γραμμές του αρχείου μετρήσεων.

```
FileOpen(1, Filenames(j), OpenMode.Input)
```

```
Do While Not EOF(1)
```

Εκτέλεση εφόσον δεν είμαστε στο τέλος του αρχείου.

```
inputdata = LineInput(1)
```

Δηλώνουμε από πού έρχονται τα δεδομένα.

```
If VB.Left(inputdata, 4) = "ECN0" Then
```

Εάν τα τέσσερα πρώτα αριστερά γράμματα κάθε γραμμής δημιουργούν την λέξη ECN0 τότε ο κώδικας ξεκινά την ανάλυση της γραμμής. Σε διαφορετική περίπτωση προχωράει και ελέγχει την επόμενη γραμμή.

```
gr = gr + 1
```

Ο αριθμός των γραμμών αυξάνεται κατά ένα

```
Dim str As String()
```

Δηλώνουμε την μεταβλητή str ως string

```
Dim dr As DataRow
```

Δηλώνουμε την μεταβλητή dr ως νέα στήλη δεδομένων.

```
Dim ct As Integer
```

Δηλώνουμε την μεταβλητή ct ως ακέραιο.

```
str = inputdata.Split()
```

Με αυτή την εντολή χωρίζουμε τα δεδομένα σε κομμάτια. Τον τρόπο με τον οποίο θα τα διαχωρίσουμε τον ορίζουμε στην παρένθεση. Στη συγκεκριμένη περίπτωση είναι ο χαρακτήρας κενού. Οπότε όταν συναντάμε κενό καταλαβαίνουμε ότι πρόκειται για άλλο είδος δεδομένου!

```
strLen = str.Length - 1
```

Η μεταβλητή αποθηκεύει το μήκος του αρχείου μειωμένο κατά ένα.

```
dr = general.NewRow
```

Το dr ορίζεται ως στήλη.

```
dr.Item(0) = gr
```

Το item 0 παίρνει τον A/A

```
dr.Item(1) = str(1)
```

Το item 1 το γεωγρ. Μήκος.

```
dr.Item(2) = str(2)
```

Το item 2 το γεωγρ. Πλάτος.

```
dr.Item(3) = str(3)
```

Ομοίως εισάγονται και τα υπόλοιπα δεδομένα στις κατάλληλες στήλες κάθε γραμμής.

```
dr.Item(4) = str(4)
```

```
dr.Item(5) = str(5)
```

```
dr.Item(6) = str(6)
```

```
dr.Item(7) = str(8)
```

```
dr.Item(8) = str(7)
```

```
dr.Item(9) = str(9)
```

```
dr.Item(10) = str(10)
```

```
dr.Item(11) = str(11)
```

```
dr.Item(12) = str(12)
```

```
dr.Item(13) = str(13)
```

```
general.Rows.Add(dr)
```

Μεταφέρονται τα στοιχεία στον πίνακα general.

```
ct = str(14)
```

Η “λέξη” 14 περιέχει το πλήθος των active servers.

Δήλωση της dr ως νέα στήλη. Στο πρώτο της στοιχείο θα μπει ο αύξων αριθμός των γραμμών. Το δεύτερο και τρίτο στοιχείο αποθηκεύουν το γεωγραφικό μήκος και πλάτος.

```
maxMonitor = str((14 * ct) + 15)
```

Το πλήθος των monitored servers βρίσκεται στην θέση που μας υποδουκνεί η μαθηματική συνάρτηση της παραπάνω συνάρτησης.

```
maxDetected = str((14 * ct + 14 * maxMonitor) + 16)
```

Ομοίως η παραπάνω μαθηματική συνάρτηση κρύβει το πλήθος των detected servers και η αμέσως επόμενη το πλήθος των undetected. Αν παρατηρήσουμε τις συναρτήσεις θα καταλάβουμε ότι οι θέσεις του κάθε είδους server (εκτός Active) είναι άμεσα εξαρτημένες απ την θέση των active servers.

```
maxUndetected = str((14 * ct + 14 * maxMonitor + 14 *
maxDetected) + 17)
```

```
i = 1
```

```
While 1 = 1
```

Όπως και παραπάνω έτσι κι εδώ η μεταβλητή dr μας βοηθάει να αποθηκεύσουμε τα δεδομένα μας.

```
dr = mytable.NewRow
```

Δήλωση νέας στήλης για τον mytable.

```
dr.Item(0) = gr
```

Το στοιχείο 0 έχει τον αριθμό γραμμών.

```
If ct >= i Then
```

```
  k = (15 * i) - i + 1
```

```
  dr.Item(1) = "Active"
```

```
ElseIf maxMonitor + ct >= i Then
```

```
  k = (14 * ct) + 16 + 14 * (i - ct - 1)
```

```
  dr.Item(1) = "Monitored"
```

```
ElseIf maxDetected + maxMonitor + ct >= i Then
```

```
  k = (14 * ct + 14 * maxMonitor) + 17 + 14 * (i -
```

```
maxMonitor - ct - 1)
```

```
  dr.Item(1) = "Detected"
```

```
Else
```

```
Exit While
```

Έξοδος απ τον βρόγχο.

```
End If
```

```
dr.Item(2) = str(k)
```

```
dr.Item(3) = str(k + 1)
```

```
dr.Item(4) = str(k + 2)
```

```
dr.Item(5) = str(k + 3)
```

```
dr.Item(6) = str(k + 4)
```

```
mytable.Rows.Add(dr)
```

Πρόσθεση της νέας στήλης.

```
i += 1
```

Το i αυξάνεται κατά ένα.

```
End While
```

```
End If
```

```
Loop
```

Τέλος βρόγχου

Στην περίπτωση που ο αριθμός των active servers είναι μεγαλύτερος του ενός τότε κάνουμε τα ακόλουθα σύμφωνα με τις μαθηματικές σχέσεις που έχουμε υπολογίσει. Προσθέτουμε τα στοιχεία των active servers στον πίνακα δεδομένων. Αν έχουμε έναν μόνο active server τότε προχωράμε στον έλεγχο των monitored servers και κάνουμε την εισαγωγή δεδομένων. Ομοίως για τους detected & undetected.

**FileClose(1)**  
Κλείσιμο του αρχείου που ανοίξαμε προς επεξεργασία

**Next**

**cad()**  
Καλείται η συνάρτηση cad

**caf()**  
Καλείται η συνάρτηση caf

**Me.Text = " 3G Analysis " & OpenFileDialog1.FileName & ""**  
Στο filedialog που εμφανίζεται θα φαίνεται ως τίτλος η διαδρομή του επιλεγμένου αρχείου.

**ToolStripButton2.Enabled = True**  
**ToolStripButton3.Enabled = True**  
**ToolStripButton4.Enabled = True**  
**ToolStripButton5.Enabled = True**  
**ToolStripButton6.Enabled = True**  
**ToolStripButton7.Enabled = True**  
**ToolStripButton8.Enabled = True**  
**ToolStripMenuItem6.Enabled = True**  
**AnalyzeToolStripMenuItem.Enabled = True**  
**SaveToolStripButton.Enabled = True**  
**PrintToolStripButton.Enabled = True**  
**SettingsToolStripMenuItem1.Enabled = True**

Από τη στιγμή που ανοίχτηκε το αρχείο και γίνανε όλες οι απαραίτητες διαδικασίες τότε όλα τα κουμπιά της γραμμής εργασιών και όλα τα στοιχεία του μενού ενεργοποιούνται ώστε να μπορεί να είναι δυνατή η ανάλυση των δεδομένων.

## CAD

**Sub cad()**  
Συνάρτηση την οποία καλούμε με σκοπό την ανάλυση του αρχείου μετρήσεων όσον αφορά τα γεγονότα CAD.

**Dim inputdata As String**  
Δηλώνεται η μεταβλητή inputdata ως string.

**Dim strLen, j As Integer**  
**Dim Filenames() As String = OpenFileDialog1.FileNames**  
Ορίζουμε ένα νέο file dialog το οποίο μας επιτρέπει να ανοίγουμε αρχεία δεδομένων προς επεξεργασία.

**gr = 0**  
**For j = 0 To Filenames.Length - 1**  
Ξεκινάει μία διαδικασία η οποία επαναλαμβάνεται τόσες φορές όσες είναι και οι γραμμές του αρχείου μετρήσεων.

**FileOpen(1, Filenames(j), OpenMode.Input)**  
**Do While Not EOF(1)**  
Εκτέλεση εφόσον δεν είμαστε στο τέλος του αρχείου.

**inputdata = LineInput(1)**  
**If VB.Left(inputdata, 3) = "CAD" Then**  
Εάν τα τρία πρώτα αριστερά γράμματα κάθε γραμμής δημιουργούν την λέξη CAD τότε ο κώδικας ξεκινά την ανάλυση της γραμμής. Σε διαφορετική περίπτωση προχωράει και ελέγχει την επόμενη γραμμή.

**gr = gr + 1**  
**Dim str As String()**

```

Dim de As DataRow
str = inputdata.Split()
strLen = str.Length - 1
de = general2.NewRow
de.Item(0) = gr
de.Item(1) = str(1)
de.Item(2) = str(2)
de.Item(3) = str(3)
de.Item(4) = str(4)
de.Item(5) = str(5)
de.Item(6) = str(6)
de.Item(7) = str(8)
de.Item(8) = str(7)
de.Item(9) = str(9)
de.Item(10) = str(10)
general2.Rows.Add(de)
End If
Loop

```

Τέλος βρόγχου

```
FileClose(1)
```

Κλείσιμο του αρχείου που ανοίξαμε προς επεξεργασία

```
Next
End Sub
```

## CAF

```
Sub caf()
```

Συνάρτηση την οποία καλούμε με σκοπό την ανάλυση του αρχείου μετρήσεων όσον αφορά τα γεγονότα CAF.

```
Dim inputdata As String
```

Δηλώνεται η μεταβλητή inputdata ως string.

```
Dim strLen As Integer
```

```
gr = 0
```

```
FileOpen(1, (OpenFileDialog1.FileName), OpenMode.Input)
```

Ορίζουμε ένα νέο file dialog το οποίο μας

επιτρέπει να ανοίγουμε αρχεία δεδομένων προς επεξεργασία.

```
Do While Not EOF(1)
```

```
inputdata = LineInput(1)
```

```
If VB.Left(inputdata, 3) = "CAF" Then
```

Εάν τα τρία πρώτα αριστερά γράμματα κάθε γραμμής δημιουργούν την λέξη CAF τότε ο κώδικας ξεκινά την ανάλυση της γραμμής. Σε διαφορετική περίπτωση προχωράει και ελέγχει την επόμενη γραμμή.

```
gr = gr + 1
```

```
Dim str As String()
```

```
Dim da As DataRow
```

```
str = inputdata.Split()
```

```
strLen = str.Length - 1
```

```
da = generalcaf.NewRow
```

```
da.Item(0) = gr
```

```
da.Item(1) = str(1).Replace(".", ",")
```

```
da.Item(2) = str(2).Replace(".", ",")
```

```
da.Item(3) = str(3)
```

```
da.Item(4) = str(4)
```

```
da.Item(5) = str(5)
```

Για την ανάλυση των γραμμών (γεγονότων) CAD χρησιμοποιούμε την ίδια λογική κώδικα όπως περιγράψαμε παραπάνω για τα γεγονότα ECN0(βλέπε σελ 31)

Για την ανάλυση των γραμμών (γεγονότων) CAF χρησιμοποιούμε την ίδια λογική κώδικα όπως περιγράψαμε για τα δύο προηγούμενα γεγονότα.

```

    da.Item(6) = str(6)
    da.Item(7) = str(8)
    da.Item(8) = str(7)
    da.Item(9) = str(9)
    da.Item(10) = str(10)
    generalcaf.Rows.Add(da)
End If
Loop

FileClose(1)
End Sub

Sub Avg()
    Συνάρτηση την οποία καλούμε με σκοπό την εύρεση των
    πληροφοριών που χρειάζονται για να συμπληρωθεί ο πίνακας
    “Μελέτη Κυψελών”.

    avgtable.Clear()
        Καθαρισμός του πίνακα avgtable.

    Dim i, counta, rscp, ecno, maxrscp, maxecno, minrscp, minecno, scr, k
    As Integer
        Δηλώνονται οι μεταβλητές ως ακέραιοι.
        i      : Χρησιμοποιείται για να προχωράμε στην επόμενη
                γραμμή ECNO όταν είναι απαραίτητο.
        counta : Το πλήθος των ίδιων scrambling codes.
        rscp   : Το άθροισμα των τιμών των rscp με ίδιο scrambling code.
        ecno   : Το άθροισμα των τιμών των ecno με ίδιο scrambling code.
        maxrscp : Αποθήκευση της μέγιστης τιμής του rscp.
        maxecno : Αποθήκευση της μέγιστης τιμής του ecno.
        minrscp : Αποθήκευση της ελάχιστης τιμής του rscp.
        minecno : Αποθήκευση της ελάχιστης τιμής του ecno.
        scr    : Αποθήκευση της τιμής του scrambling code που έχει
                επιλέξει ο χρήστης.
        k      : Χρησιμοποιείται ως μετρητής για να επαναλαμβάνουμε
                μία διαδικασία.

    Dim avgrscp, avgecno As Double
        Δηλώνονται οι μεταβλητές ως double επειδή θα αποθηκεύουν τιμές
        με υποδιαστολή.
        avgrscp : Αποθήκευση της μέσης τιμής των rscp (για καθορισμένο
                scrambling code).
        avgecno : Αποθήκευση της μέσης τιμής των ecno (για καθορισμένο
                scrambling code).

    avgrscp = avgecno = 0
    maxrscp = -150
    maxecno = -25
    rscp = 0
    counta = 0
    ecno = 0
    minrscp = 0
    minecno = 0

    Dim de As DataRow

    For k = 1 To countscrcodes

```

Αρχικοποίηση με τις απαραίτητες τιμές  
για να ξεκινήσει η διαδικασία.

```

scr = codes(k)
For i = 0 To mytable.Rows.Count - 1
  If mytable.Rows(i).Item(1) = "Active" Then
    If mytable.Rows(i).Item(3) = scr Then
      rscp = mytable.Rows(i).Item(6) + rscp
      ecno = mytable.Rows(i).Item(4) + ecno
      counta = counta + 1
      If maxrscp < mytable.Rows(i).Item(6) Then
        maxrscp = mytable.Rows(i).Item(6)
      End If
      If minrscp > mytable.Rows(i).Item(6) Then
        minrscp = mytable.Rows(i).Item(6)
      End If
      If maxecno < mytable.Rows(i).Item(4) Then
        maxecno = mytable.Rows(i).Item(4)
      End If
      If minecno > mytable.Rows(i).Item(4) Then
        minecno = mytable.Rows(i).Item(4)
      End If
    End If
  End If
Next i
de = avgtable.NewRow
avgrscp = rscp / counta
avgrscp = Math.Round(avgrscp, 3)
avgecno = ecno / counta
avgecno = Math.Round(avgecno, 3)
de.Item(0) = scr
de.Item(1) = maxrscp
de.Item(2) = minrscp
de.Item(3) = avgrscp
de.Item(4) = maxecno
de.Item(5) = minecno
de.Item(6) = avgecno
avgtable.Rows.Add(de)
rscp = 0
counta = 0
ecno = 0
minrscp = 0
minecno = 0
maxrscp = -150
maxecno = -25
Next k
End Sub

```

Η διαδικασία επαναλαμβάνεται τόσες φορές όσες είναι και τα διαφορετικά scrambling codes που έχουν βρεθεί(countscr codes).

Εισάγουμε το πρώτο scr code στο scr και στη συνέχεια τσεκάρουμε, ένα ένα, όλα τα γεγονότα ECNO (*MONO GIA TOYΣ ACTIVE SERVERS*).

Αν το scr. code της γραμμής που ελέγχουμε είναι ίδιο με την μεταβλητή scr τότε :

- 1) Προσθέτουμε το rscp στην υπάρχουσα τιμή του rscp.
- 2) Προσθέτουμε το ecno στην υπάρχουσα τιμή του ecno.
- 3) Αυξάνουμε την τιμή του counta κατά 1.
- 4) Συγκρίνουμε και βρίσκουμε τα maxrscp, maxecno, minrscp και minecno.

Όταν ελέγχουμε όλες τις γραμμές τότε :

- 1) Δημιουργείται μια νέα γραμμή στον πίνακα avgtable.
- 2) Βρίσκουμε τις μέσες τιμές διαιρώντας το άθροισμα προς το πλήθος.
- 3) Εισάγουμε στις κατάλληλες θέσεις του πίνακα τα δεδομένα.
- 4) Αρχικοποιούμε πάλι τις μεταβλητές.
- 5) Προχωράμε στο επόμενο k, δηλαδή επαναλαμβάνουμε την διαδικασία για το επόμενο scr. code.

## Εύρεση των scrambling codes

**Sub Findcodes ()**

Συνάρτηση την οποία καλούμε με σκοπό την εύρεση των διαφορετικών scrambling codes των Active servers ενός αρχείου μετρήσεων.

**ComboBox1.Items.Clear ()**

**Form2.ComboBox1.Items.Clear ()**

**Form2.ComboBox2.Items.Clear ()**

Καθαρίζουμε τα περιεχόμενα των παραπάνω τριών combobox

**Dim i, cn, aa, temp As Integer**

Δηλώνονται οι μεταβλητές ως ακέραιοι.

**i** : Χρησιμοποιείται για να προχωράμε στην επόμενη γραμμή ECNO όταν είναι απαραίτητο.

**cn** : Χρησιμοποιείται ως δείκτης του πίνακα *codes* όταν συγκρίνουμε τα περιεχόμενα του πίνακα *codes* με το *mytable.Rows(i).Item(3)*.

**aa** : Αποθήκευση προσωρινού scrambling code.

**temp** : Μας δείχνει την γραμμή του πίνακα *mytable* στην οποία βρισκόμαστε (π.χ 450).

**countscrcodes = 1**

**cn = 1**

**temp = -1**

Αρχικοποίηση τιμών.

**i = 0**

**codes(countscrcodes) = mytable.Rows(i).Item(3)**

Εισαγωγή του scrambling code της πρώτης μέτρησης στην πρώτη γραμμή του πίνακα *codes*.

**For i = 0 To mytable.Rows.Count - 1**

Ξεκινάει μία διαδικασία η οποία θα επαναληφθεί όσες φορές είναι και οι γραμμές του πίνακα *mytable*.

**If temp <> mytable.Rows(i).Item(0) And mytable.Rows(i).Item(1) = "Active" Then**

Αν η τιμή του *temp* είναι διαφορετική από την τιμή του *mytable.Rows(i).Item(0)*, δηλαδή αν έχουμε περάσει στην επόμενη μέτρηση και η γραμμή που ελέγχουμε περιέχει Active Server (*mytable.Rows(i).Item(1)*), τότε εκτελείται η ρουτίνα που βρίσκεται μέσα στο if.

**For cn = 0 To countscrcodes**

**aa = mytable.Rows(i).Item(3)**

**If mytable.Rows(i).Item(3) = Val(codes(cn)) Then**



```

        GoTo 1
    End If
Next cn
countscrcodes = countscrcodes + 1
codes(countscrcodes) =
mytable.Rows(i).Item(3)
End If

```

Σε αυτό το σημείο του κώδικα συγκρίνουμε το scrambling code της γραμμής ( i ) που μελετάμε με τις τιμές των scrambling codes που βρίσκονται μέσα στον πίνακα codes. Αν υπάρχει τότε προχωράμε στην επόμενη μέτρηση. Αν δεν υπάρχει τότε εισάγουμε την τιμή του scrambling code στην επόμενη ελεύθερη θέση του πίνακα codes.

```

1:   temp = mytable.Rows(i).Item(0)
      Εισάγουμε στο temp την τιμή της
      μέτρησης που μελετήσαμε.

```

```

Next i
      Τέλος βρόγχου.

```

```

For i = 1 To countscrcodes
    Form2.ComboBox1.Items.Add(codes(i))
    Form2.ComboBox2.Items.Add(codes(i))
    ComboBox1.Items.Add(codes(i))
Next i

```

Εισαγωγή των scrambling codes του πίνακα codes στα παραπάνω Combobox.

```

MsgBox(" Βρέθηκαν " & d & " scrambling codes.",
MsgBoxStyle.Information, "3G Analysys")

```

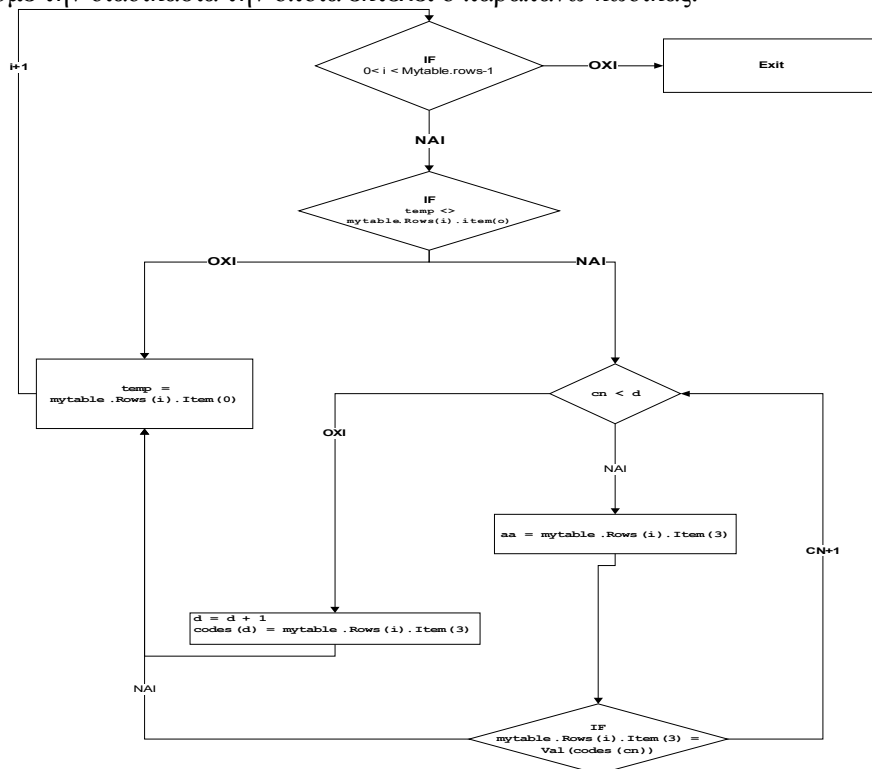
Μήνυμα το οποίο εμφανίζεται και μας ενημερώνει για το πλήθος των διαφορετικών scrambling codes που βρέθηκαν στο αρχείο μετρήσεων.

```

End Sub
      Τέλος συνάρτησης.

```

Το σχήμα που ακολουθεί είναι ένα block διάγραμμα στο οποίο μπορούμε εύκολα να διακρίνουμε την διαδικασία την οποία εκτελεί ο παραπάνω κώδικας.



## ΓΡΑΦΗΜΑΤΑ

Στη συνέχεια θα αναλύσουμε τον κώδικα που χρησιμοποιήσαμε για να δημιουργήσουμε τις γραφικές μας παραστάσεις. Επειδή όμως πολλές γραμμές κώδικα επαναλαμβάνονται, θα αποφύγουμε στην συνέχεια να τις ξανα-αναλύσουμε. Για την καλύτερη κατανόηση των παρακάτω γραμμών θα ακολουθήσουμε το παρακάτω τρόπο απεικόνισης...

**Με έντονη γραφή** : Ο κώδικας που αναλύεται για πρώτη φορά.

Με απλή γραφή : Ο κώδικας που έχει αναλυθεί σε προηγούμενη γραμμή.

Με πράσινο χρώμα : Η ανάλυση του κώδικα.

### Απλή Διαδρομή :

```
Sub SimpleRoute()
```

```
    ZedGraphControl1.Visible = True
```

Το zedgraphcontrol1 (Το παράθυρο που θα εμφανιστεί η γραφική παράσταση) γίνεται ορατό.

```
    Dim oldPane As ZedGraph.GraphPane = ZedGraphControl1.GraphPane
    ZedGraphControl1.GraphPane = New ZedGraph.GraphPane(oldPane.Rect,
oldPane.Title.Text, oldPane.XAxis.Title.Text,
oldPane.YAxis.Title.Text)
```

Καθαρίζει ό,τι γράφημα υπάρχει στο zedgraph control.

```
    Dim mypane As ZedGraph.GraphPane = New ZedGraph.GraphPane
    mypane = Me.ZedGraphControl1.GraphPane
```

Ορίζουμε μία καινούρια γραφική με όνομα mypane και την ορίζουμε ως γράφημα του zedgraphcontrol1.

```
    mypane.Title.Text = tldiadromi
    mypane.XAxis.Title.Text = tlplatos
    mypane.YAxis.Title.Text = tlmikos
```

Ορίζουμε τον τίτλο του γραφήματος, τον τίτλο του άξονα X και τέλος τον τίτλο του άξονα Y.

```
    Dim list1 As ZedGraph.PointPairList = New ZedGraph.PointPairList
```

Ορίζουμε το list1 ως μία 'λίστα' του zedgraph στην οποία αποθηκεύουμε ζευγάρια τιμών (X,Y).

Οι 10 γραμμές κώδικα που ακολουθούν είναι αυτές που "επιλέγουν" τα στοιχεία X και Y που εισάγουμε στο list1. Στην πράξη ελέγχουμε όλες τις γραμμές του πίνακα mytable (For i =0 To mytable.Rows.Count - 1) και αν βρισκόμαστε σε μία μέτρηση για πρώτη φορά (temp <> mytable.Rows(i).Item(0)) και αυτή η μέτρηση περιέχει Active server (mytable.Rows(i).Item(1) = "Active") τότε εισάγουμε το γεωγραφικό μήκος και πλάτος (general.Rows(j).Item(2), general.Rows(j).Item(1) αντίστοιχα) στο list1.

```

Dim i, j As Integer
Dim temp As Integer = -1
j = 0
For i = 0 To mytable.Rows.Count - 1
    If temp <> mytable.Rows(i).Item(0) And mytable.Rows(i).Item(1)
= "Active" Then
        j = mytable.Rows(i).Item(0) - 1
        list1.Add(general.Rows(j).Item(2), general.Rows(j).Item(1),
"Μέτρηση:" & j + 1 & "")
        End If
        temp = mytable.Rows(i).Item(0)
    Next i

```

```

Dim start As TextObj = New TextObj("Start
Point", general.Rows(0).Item(2),
general.Rows(0).Item(1))
start.FontSpec.FontColor = Color.DarkMagenta
start.FontSpec.Border.IsVisible = True
start.FontSpec.Fill.IsVisible = True
start.FontSpec.Size = 8
mypane.GraphObjList.Add(start)

```

Δημιουργούμε ένα αντικείμενο κειμένου και το τοποθετούμε στην θέση της πρώτης μέτρησης. Στη συνέχεια ορίζουμε τις ιδιότητες του κειμένου και του πλαισίου του (χρώμα, μέγεθος, αν είναι ορατό). Τέλος το εμφανίζουμε στο γράφημα μας (mypane).

```

Dim text As TextObj
text = New TextObj(tltmima, 0.93F, -0.04F)
text.Location.CoordinateFrame = CoordType.ChartFraction
text.FontSpec.Angle = 15.0F

```

Δίνουμε στο text μία κλίση 15 μοιρών.

```

text.FontSpec.FontColor = Color.DarkGray
text.FontSpec.IsBold = True
text.FontSpec.Size = 10

```

Το text γίνεται σκούρο γκρι, bold, και έχει μέγεθος 10 pt.

```

text.FontSpec.Border.IsVisible = False
text.FontSpec.Fill.IsVisible = False

```

```

text.Location.AlignH = AlignH.Center
text.Location.AlignV = AlignV.Bottom

```

Ευθυγραμμίζουμε την θέση του κειμένου.

```
mypane.GraphObjList.Add(text)
```

```

Dim mycurve1 As ZedGraph.LineItem = mypane.AddCurve(tldiadromi,
list1, Color.Blue, ZedGraph.SymbolType.Circle)
mycurve1.Symbol.Fill = New ZedGraph.Fill(Color.Blue)
mycurve1.Symbol.Size = 1

```

Ορίζουμε το mycurve1 ως ένα νέο αντικείμενο 'γραμμής', χρώματος μπλε, με επιγραφή "Διαδρομή" και σύμβολο του κάθε σημείου ένα κύκλο.

Ορίζουμε το εσωτερικό του με χρώμα μπλε καθώς και το μέγεθος του συμβόλου σε 1.

```
mypane.Fill = New ZedGraph.Fill(Color.WhiteSmoke, Color.Lavender,
0.0F)
```

Ορίζουμε τα χρώματα του mypane.

```

mypane.YAxis.Scale.FontSpec.FontColor = Color.Black
mypane.YAxis.Scale.FontSpec.Size = 8
mypane.YAxis.Scale.FontSpec.IsBold = False
mypane.XAxis.Scale.FontSpec.FontColor = Color.Black
mypane.XAxis.Scale.FontSpec.Size = 8
mypane.XAxis.Scale.FontSpec.IsBold = False
mypane.XAxis.Title.FontSpec.Size = 12
mypane.XAxis.Title.FontSpec.IsItalic = True
mypane.XAxis.Title.FontSpec.FontColor = Color.Chocolate
mypane.YAxis.Title.FontSpec.Size = 12
mypane.YAxis.Title.FontSpec.IsItalic = True
mypane.YAxis.Title.FontSpec.FontColor = Color.Chocolate
mycurve1.Line.IsVisible = False
mypane.XAxis.MajorGrid.IsVisible = True
mypane.YAxis.MajorGrid.IsVisible = True
mypane.Y2Axis.IsVisible = False
mypane.Chart.Fill = New ZedGraph.Fill(Color.White, Color.LightGray,
45.0F)

```

Στις πρώτες 6 γραμμές ορίζουμε το κείμενο των τμών στους άξονες X και Y σε μαύρο, μέγεθος 8 και απλή γραφή.

Στις επόμενες 6 ορίζουμε το κείμενο των τίτλων στους άξονες X και Y σε Chocolate, μέγεθος 12 και πλάγια γραφή. Κάνουμε ορατή την γραμμή του mycurve1 όπως επίσης και το πλέγμα των αξόνων X και Y.

Ορίζουμε τα χρώματα του διαγράμματος(εντός των αξόνων X και Y).

```

Dim g As Graphics = Me.CreateGraphics
mypane.AxisChange(g)
Me.ZedGraphControl1.AxisChange()
g.Dispose()

```

Εμφανίζουμε την γραφική στο zedgraphcontrol1.

```
End Sub
```

### Διαδρομή με Χρωματικό Κώδικα :

```

Sub routecodecolour()
    ZedGraphControl1.Visible = True
    Dim oldPane As ZedGraph.GraphPane = ZedGraphControl1.GraphPane
    ZedGraphControl1.GraphPane = New ZedGraph.GraphPane(oldPane.Rect,
oldPane.Title.Text, oldPane.XAxis.Title.Text,oldPane.YAxis.Title.
Text)
    Dim mypane As ZedGraph.GraphPane = New ZedGraph.GraphPane
    mypane = Me.ZedGraphControl1.GraphPane

    mypane.Title.Text = tldiadromi
    mypane.XAxis.Title.Text = tlplatos
    mypane.YAxis.Title.Text = tlmikos

    Dim list1 As ZedGraph.PointPairList = New ZedGraph.PointPairList
    Dim list2 As ZedGraph.PointPairList = New ZedGraph.PointPairList
    Dim list3 As ZedGraph.PointPairList = New ZedGraph.PointPairList
    Dim list4 As ZedGraph.PointPairList = New ZedGraph.PointPairList
    Dim list5 As ZedGraph.PointPairList = New ZedGraph.PointPairList
    Dim list6 As ZedGraph.PointPairList = New ZedGraph.PointPairList

    Dim i, j As Integer
    Dim temp As Integer = -1
    j = 0
    For i = 0 To mytable.Rows.Count - 1
        If temp <> mytable.Rows(i).Item(0) And mytable.Rows(i).Item(1) =
"Active" Then
            j = mytable.Rows(i).Item(0) - 1
            If mytable.Rows(i).Item(6) >= -69 Then
                list1.Add(general.Rows(j).Item(2), general.Rows(j).Item(1),
"Μέτρηση:" & j + 1 & "")
            End If
        End If
    Next i
End Sub

```

Δημιουργούμε 6 λίστες.  
1 λίστα για κάθε χρώμα.

```

    ElseIf mytable.Rows(i).Item(6) >= -72 And mytable.Rows(i).
Item(6) <= -70 Then
        list2.Add(general.Rows(j).Item(2), general.Rows(j).Item(1),
"Μέτρηση:" & j + 1 & "")
        ElseIf mytable.Rows(i).Item(6) >= -82 And
mytable.Rows(i).Item(6) <= -73 Then
            list3.Add(general.Rows(j).Item(2), general.Rows(j).Item(1),
"Μέτρηση:" & j + 1 & "")
            ElseIf mytable.Rows(i).Item(6) >= -90 And
mytable.Rows(i).Item(6) <= -83 Then
                list4.Add(general.Rows(j).Item(2), general.Rows(j).Item(1),
"Μέτρηση:" & j + 1 & "")
                ElseIf mytable.Rows(i).Item(6) >= -100 And
mytable.Rows(i).Item(6) <= -91 Then
                    list5.Add(general.Rows(j).Item(2), general.Rows(j).Item(1),
"Μέτρηση:" & j + 1 & "")
                    ElseIf mytable.Rows(i).Item(6) >= -130 And
mytable.Rows(i).Item(6) <= -101 Then
                        list6.Add(general.Rows(j).Item(2), general.Rows(j).Item(1),
"Μέτρηση:" & j + 1 & "")
                        End If
                    End If
                End If
            End If
        End If
    temp = mytable.Rows(i).Item(0)
Next i

```

Σε αυτό το σημείο επαναλάβαμε την ίδια διαδικασία προσθήκης τιμών στις λίστες όπως στην 'Απλή Διαδρομή' με τη διαφορά ότι έχουμε πάνω από μία λίστες. Ελέγχουμε την τιμή του Rscp (mytable.Rows(i).Item(6)) και ανάλογα της τιμής του προσθέτουμε τα γεωγραφικά μήκη και πλάτη στην κατάλληλη λίστα.

```

Dim start As TextObj = New TextObj("Start Point",
general.Rows(0).Item(2), general.Rows(0).Item(1))
start.FontSpec.FontColor = Color.DarkMagenta
start.FontSpec.Border.IsVisible = True
start.FontSpec.Fill.IsVisible = True
start.FontSpec.Size = 8
mypane.GraphObjList.Add(start)

Dim text As TextObj
text = New TextObj(tltmima, 0.93F, -0.04F)
text.Location.CoordinateFrame = CoordType.ChartFraction
text.FontSpec.Angle = 15.0F
text.FontSpec.FontColor = Color.DarkGray
text.FontSpec.IsBold = True
text.FontSpec.Size = 10
text.FontSpec.Border.IsVisible = False
text.FontSpec.Fill.IsVisible = False
text.Location.AlignH = AlignH.Center
text.Location.AlignV = AlignV.Bottom
mypane.GraphObjList.Add(text)

```

Ο κώδικας που ακολουθεί είναι ίδιος με αυτόν της 'Απλής διαδρομής'. Το μόνο που αλλάζει είναι ότι έχουμε προσθέσει πέντε ακόμα mycurve(ένα για κάθε λίστα) στα οποία πρέπει να καθορίσουμε τις ιδιότητες τους.

```

Dim mycurve1 As ZedGraph.LineItem = mypane.AddCurve("0 έως -69",
list1, Color.Brown, ZedGraph.SymbolType.Circle)
mycurve1.Symbol.Fill = New ZedGraph.Fill(Color.Brown)
mycurve1.Symbol.Size = 2
Dim mycurve2 As ZedGraph.LineItem = mypane.AddCurve("-70 έως -72 ",
list2, Color.Red, ZedGraph.SymbolType.Circle)
mycurve2.Symbol.Fill = New ZedGraph.Fill(Color.Red)
mycurve2.Symbol.Size = 2

```

```

Dim mycurve3 As ZedGraph.LineItem = mypane.AddCurve("-73 έως -82",
list3, Color.Orange, ZedGraph.SymbolType.Circle)
mycurve3.Symbol.Fill = New ZedGraph.Fill(Color.Orange)
mycurve3.Symbol.Size = 2
Dim mycurve4 As ZedGraph.LineItem = mypane.AddCurve("-83 έως -90",
list4, Color.Blue, ZedGraph.SymbolType.Circle)
mycurve4.Symbol.Fill = New ZedGraph.Fill(Color.Blue)
mycurve4.Symbol.Size = 2
Dim mycurve5 As ZedGraph.LineItem = mypane.AddCurve("-91 έως -100",
list5, Color.Navy, ZedGraph.SymbolType.Circle)
mycurve5.Symbol.Fill = New ZedGraph.Fill(Color.Navy)
mycurve5.Symbol.Size = 2
Dim mycurve6 As ZedGraph.LineItem = mypane.AddCurve("-101 έως -130",
list6, Color.Green, ZedGraph.SymbolType.Circle)
mycurve6.Symbol.Fill = New ZedGraph.Fill(Color.Green)
mycurve6.Symbol.Size = 2

mypane.Fill = New ZedGraph.Fill(Color.WhiteSmoke, Color.Lavender,
0.0F)
mypane.YAxis.Scale.FontSpec.FontColor = Color.Black
mypane.YAxis.Scale.FontSpec.Size = 8
mypane.YAxis.Scale.FontSpec.IsBold = False
mypane.XAxis.Scale.FontSpec.FontColor = Color.Black
mypane.XAxis.Scale.FontSpec.Size = 8
mypane.XAxis.Scale.FontSpec.IsBold = False
mypane.XAxis.Title.FontSpec.Size = 12
mypane.XAxis.Title.FontSpec.IsItalic = True
mypane.XAxis.Title.FontSpec.FontColor = Color.Chocolate
mypane.YAxis.Title.FontSpec.Size = 12
mypane.YAxis.Title.FontSpec.IsItalic = True
mypane.YAxis.Title.FontSpec.FontColor = Color.Chocolate
mycurve1.Line.IsVisible = False
mycurve2.Line.IsVisible = False
mycurve3.Line.IsVisible = False
mycurve4.Line.IsVisible = False
mycurve5.Line.IsVisible = False
mycurve6.Line.IsVisible = False
mypane.XAxis.MajorGrid.IsVisible = True
mypane.Chart.Fill = New ZedGraph.Fill(Color.White, Color.LightGray,
45.0F)
Dim g As Graphics = Me.CreateGraphics
mypane.AxisChange(g)
Me.ZedGraphControl1.AxisChange()
g.Dispose()
End Sub

```

### ***Διαδρομή Βάσει του Ec/No :***

```

Sub routecodecolourECN0()
    ZedGraphControl1.Visible = True
    Dim oldPane As ZedGraph.GraphPane = ZedGraphControl1.GraphPane
    ZedGraphControl1.GraphPane = New ZedGraph.GraphPane(oldPane.Rect,
oldPane.Title.Text, oldPane.XAxis.Title.Text,
oldPane.YAxis.Title.Text)
    Dim mypane As ZedGraph.GraphPane = New ZedGraph.GraphPane
    mypane = Me.ZedGraphControl1.GraphPane
    mypane.Title.Text = tldiadromiecno
    mypane.XAxis.Title.Text = tlplatos
    mypane.YAxis.Title.Text = tlmikos

```

```
Dim list1 As ZedGraph.PointPairList = New ZedGraph.PointPairList
Dim list2 As ZedGraph.PointPairList = New ZedGraph.PointPairList
Dim list3 As ZedGraph.PointPairList = New ZedGraph.PointPairList
```

```
Dim i, j As Integer
Dim temp As Integer = -1
j = 0
```

Σε αυτό το σημείο επαναλαμβάνουμε την ίδια διαδικασία προσθήκης τιμών στις λίστες όπως στην **‘Διαδρομή με χρωματικό κώδικα’** με τη διαφορά ότι ελέγχουμε την τιμή του Ec/No (mytable.Rows(i).Item(4)) και ανάλογα της τιμής του προσθέτουμε τα γεωγραφικά μήκη και πλάτη στην κατάλληλη λίστα.

```
For i = 0 To mytable.Rows.Count - 1
    If temp <> mytable.Rows(i).Item(0) And mytable.Rows(i).Item(1) =
"Active") Then
        j = mytable.Rows(i).Item(0) - 1
        If mytable.Rows(i).Item(4) >= -12 Then
            list1.Add(general.Rows(j).Item(2), general.Rows(j).Item(1),
"Μέτρηση:" & j + 1 & "")
        ElseIf mytable.Rows(i).Item(4) >= -17 And mytable.Rows(i).
Item(4) <= -13 Then
            list2.Add(general.Rows(j).Item(2), general.Rows(j).Item(1),
"Μέτρηση:" & j + 1 & "")
        ElseIf mytable.Rows(i).Item(4) <= -18 Then
            list3.Add(general.Rows(j).Item(2), general.Rows(j).Item(1),
"Μέτρηση:" & j + 1 & "")
        End If
    End If
    temp = mytable.Rows(i).Item(0)
Next i
```

```
Dim start As TextObj = New TextObj("Start Point",
general.Rows(0).Item(2), general.Rows(0).Item(1))
start.FontSpec.FontColor = Color.DarkMagenta
start.FontSpec.Border.IsVisible = True
start.FontSpec.Fill.IsVisible = True
start.FontSpec.Size = 8
mypane.GraphObjList.Add(start)
```

```
Dim text As TextObj
text = New TextObj(tltmima, 0.93F, -0.04F)
text.Location.CoordinateFrame = CoordType.ChartFraction
text.FontSpec.Angle = 15.0F
text.FontSpec.FontColor = Color.DarkGray
text.FontSpec.IsBold = True
text.FontSpec.Size = 10
text.FontSpec.Border.IsVisible = False
text.FontSpec.Fill.IsVisible = False
text.Location.AlignH = AlignH.Center
text.Location.AlignV = AlignV.Bottom
mypane.GraphObjList.Add(text)
```

```
Dim mycurve1 As ZedGraph.LineItem = mypane.AddCurve("0 έως -12",
list1, Color.Green, ZedGraph.SymbolType.Circle)
mycurve1.Symbol.Fill = New ZedGraph.Fill(Color.Green)
mycurve1.Symbol.Size = 2
Dim mycurve2 As ZedGraph.LineItem = mypane.AddCurve("-13 έως -17 ",
list2, Color.Yellow, ZedGraph.SymbolType.Circle)
```

```

mycurve2.Symbol.Fill = New ZedGraph.Fill(Color.Yellow)
mycurve2.Symbol.Size = 3
Dim mycurve3 As ZedGraph.LineItem = mypane.AddCurve("-18 έως -25",
list3, Color.Red, ZedGraph.SymbolType.Circle)
mycurve3.Symbol.Fill = New ZedGraph.Fill(Color.Red)
mycurve3.Symbol.Size = 3

mypane.Fill = New ZedGraph.Fill(Color.WhiteSmoke, Color.Lavender,
0.0F)
mypane.YAxis.Scale.FontSpec.FontColor = Color.Black
mypane.YAxis.Scale.FontSpec.Size = 8
mypane.YAxis.Scale.FontSpec.IsBold = False
mypane.XAxis.Scale.FontSpec.FontColor = Color.Black
mypane.XAxis.Scale.FontSpec.Size = 8
mypane.XAxis.Scale.FontSpec.IsBold = False
mypane.XAxis.Title.FontSpec.Size = 12
mypane.XAxis.Title.FontSpec.IsItalic = True
mypane.XAxis.Title.FontSpec.FontColor = Color.Chocolate
mypane.YAxis.Title.FontSpec.Size = 12
mypane.YAxis.Title.FontSpec.IsItalic = True
mypane.YAxis.Title.FontSpec.FontColor = Color.Chocolate
mycurve1.Line.IsVisible = False
mycurve2.Line.IsVisible = False
mycurve3.Line.IsVisible = False
mypane.XAxis.MajorGrid.IsVisible = True
mypane.Chart.Fill = New ZedGraph.Fill(Color.White, Color.LightGray,
45.0F)
Dim g As Graphics = Me.CreateGraphics
mypane.AxisChange(g)
Me.ZedGraphControll1.AxisChange()
g.Dispose()
End Sub

```

### Για συγκεκριμένο scrambling code :

```

Sub coderoute()
    Dim codenum As Integer
    If Form2.Visible = True Then
        codenum = Val(Form2.ComboBox1.Text)
    Else
        codenum = Val(ComboBox1.Text)
    End If
    If codenum = 0 Then
        MsgBox("Δεν έχει επιλεγεί κανένα scrambling code", MsgBoxStyle.
Exclamation, "3G Analysis")
    Else

```

Ορίζουμε το codenum ως ακέραιο.  
Σε αυτό το αντικείμενο θα εισάγουμε την τιμή του scr. code που θέλουμε να μελετήσουμε.  
Το if που ακολουθεί ελέγχει από ποιο combobox θα επιλεγεί το scr. code.

Αν το codenum=0 τότε εμφανίζεται το παραπάνω μήνυμα και δεν εμφανίζεται κανένα γράφημα.

```

ZedGraphControll1.Visible = True
Dim oldPane As ZedGraph.GraphPane = ZedGraphControll1.GraphPane
ZedGraphControll1.GraphPane = New ZedGraph.GraphPane(oldPane.Rect,
oldPane.Title.Text, oldPane.XAxis.Title.Text,
oldPane.YAxis.Title.Text)
Dim mypane As ZedGraph.GraphPane = New ZedGraph.GraphPane
mypane = Me.ZedGraphControll1.GraphPane
mypane.Title.Text = "" & tldiadromiscrcode & "(" & codenum & ")"

```

Να παρατηρήσουμε ότι μέσα στον τίτλο έχουμε προσθέσει το αντικείμενο codenum. Με αυτό τρόπο εμφανίζουμε στον τίτλο την τιμή του Scr. Code που έχει επιλέξει ο χρήστης.



```

mypane.XAxis.Title.Text = tlplatos
mypane.YAxis.Title.Text = tlmikos

Dim text As TextObj
text = New TextObj(tltmima, 0.93F, -0.04F)
text.Location.CoordinateFrame = CoordType.ChartFraction
text.FontSpec.Angle = 15.0F
text.FontSpec.FontColor = Color.DarkGray
text.FontSpec.IsBold = True
text.FontSpec.Size = 10
text.FontSpec.Border.IsVisible = False
text.FontSpec.Fill.IsVisible = False
text.Location.AlignH = AlignH.Center
text.Location.AlignV = AlignV.Bottom
mypane.GraphObjList.Add(text)

Dim list1 As ZedGraph.PointPairList = New ZedGraph.PointPairList
Dim list2 As ZedGraph.PointPairList = New ZedGraph.PointPairList
Dim list3 As ZedGraph.PointPairList = New ZedGraph.PointPairList
Dim list4 As ZedGraph.PointPairList = New ZedGraph.PointPairList
Dim list5 As ZedGraph.PointPairList = New ZedGraph.PointPairList
Dim list6 As ZedGraph.PointPairList = New ZedGraph.PointPairList
Dim j As Integer = 0
Dim temp As Integer = -1
Dim i As Integer = 0

```

Σε αυτό το σημείο επαναλαμβάνουμε την ίδια διαδικασία προσθήκης τιμών στις λίστες όπως στην **‘Διαδρομή με χρωματικό κώδικα’** με τη διαφορά ότι ελέγχουμε ένα παραπάνω στοιχείο πριν προσθέσουμε τα γεωγραφικά μήκη και πλάτη στις λίστες. Το στοιχείο αυτό είναι το `Scr. Code` και βρίσκεται στην θέση `mytable.Rows(i).Item(3)`.

```

For i = 0 To mytable.Rows.Count - 1
  If temp <> mytable.Rows(i).Item(0) And mytable.Rows(i).Item(1) = "Active" Then
    j = mytable.Rows(i).Item(0) - 1
    If mytable.Rows(i).Item(3) = codenum Then
      If mytable.Rows(i).Item(6) >= -69 Then
        list1.Add(general.Rows(j).Item(2), general.Rows(j).Item(1))
      ElseIf mytable.Rows(i).Item(6) >= -72 And mytable.Rows(i).Item(6) <= -70 Then
        list2.Add(general.Rows(j).Item(2), general.Rows(j).Item(1))
      ElseIf mytable.Rows(i).Item(6) >= -82 And mytable.Rows(i).Item(6) <= -73 Then
        list3.Add(general.Rows(j).Item(2), general.Rows(j).Item(1))
      ElseIf mytable.Rows(i).Item(6) >= -90 And mytable.Rows(i).Item(6) <= -83 Then
        list4.Add(general.Rows(j).Item(2), general.Rows(j).Item(1))
      ElseIf mytable.Rows(i).Item(6) >= -100 And mytable.Rows(i).Item(6) <= -91 Then
        list5.Add(general.Rows(j).Item(2), general.Rows(j).Item(1))
      ElseIf mytable.Rows(i).Item(6) >= -130 And mytable.Rows(i).Item(6) <= -101 Then
        list6.Add(general.Rows(j).Item(2), general.Rows(j).Item(1))
      End If
    End If
    temp = mytable.Rows(i).Item(0)
  End If
Next i

Dim myCurve1 As ZedGraph.LineItem = mypane.AddCurve("0 έως -69",
list1, Color.Brown, ZedGraph.SymbolType.Circle)
myCurve1.Symbol.Fill = New ZedGraph.Fill(Color.Brown)
myCurve1.Symbol.Size = 2
Dim myCurve2 As ZedGraph.LineItem = mypane.AddCurve("-70 έως -72 ",
list2, Color.Red, ZedGraph.SymbolType.Circle)
myCurve2.Symbol.Fill = New ZedGraph.Fill(Color.Red)
myCurve2.Symbol.Size = 2
Dim myCurve3 As ZedGraph.LineItem = mypane.AddCurve("-73 έως -82",
list3, Color.Orange, ZedGraph.SymbolType.Circle)
myCurve3.Symbol.Fill = New ZedGraph.Fill(Color.Orange)
myCurve3.Symbol.Size = 2

```

```

Dim myCurve4 As ZedGraph.LineItem = mypane.AddCurve("-83 έως -90",
list4, Color.Blue, ZedGraph.SymbolType.Circle)
myCurve4.Symbol.Fill = New ZedGraph.Fill(Color.Blue)
myCurve4.Symbol.Size = 2
Dim myCurve5 As ZedGraph.LineItem = mypane.AddCurve("-91 έως -100",
list5, Color.Navy, ZedGraph.SymbolType.Circle)
myCurve5.Symbol.Fill = New ZedGraph.Fill(Color.Navy)
myCurve5.Symbol.Size = 2
Dim myCurve6 As ZedGraph.LineItem = mypane.AddCurve("-101 έως -130",
list6, Color.Green, ZedGraph.SymbolType.Circle)
myCurve6.Symbol.Fill = New ZedGraph.Fill(Color.Green)
myCurve6.Symbol.Size = 2

mypane.Fill = New ZedGraph.Fill(Color.WhiteSmoke, Color.Lavender,
0.0F)
mypane.YAxis.Scale.FontSpec.FontColor = Color.Black
mypane.YAxis.Scale.FontSpec.Size = 8
mypane.YAxis.Scale.FontSpec.IsBold = False
mypane.XAxis.Scale.FontSpec.FontColor = Color.Black
mypane.XAxis.Scale.FontSpec.Size = 8
mypane.XAxis.Scale.FontSpec.IsBold = False
mypane.XAxis.Title.FontSpec.Size = 12
mypane.XAxis.Title.FontSpec.IsItalic = True
mypane.XAxis.Title.FontSpec.FontColor = Color.Chocolate
mypane.YAxis.Title.FontSpec.Size = 12
mypane.YAxis.Title.FontSpec.IsItalic = True
mypane.YAxis.Title.FontSpec.FontColor = Color.Chocolate
myCurve1.Line.IsVisible = False
myCurve2.Line.IsVisible = False
myCurve3.Line.IsVisible = False
myCurve4.Line.IsVisible = False
myCurve5.Line.IsVisible = False
myCurve6.Line.IsVisible = False
mypane.XAxis.MajorGrid.IsVisible = True
mypane.YAxis.MajorGrid.IsVisible = True
mypane.Chart.Fill = New ZedGraph.Fill(Color.White, Color.LightGray,
45.0F)
Dim g As Graphics = Me.CreateGraphics
mypane.AxisChange(g)
Me.ZedGraphControl1.AxisChange()
g.Dispose()
End If
End Sub

```

### ***Call Disconnect :***

```

Sub cadroute()
SimpleRoute()

```

Καλούμε την συνάρτηση που δημιουργεί την *Απλή Διαδρομή*. Στην συνέχεια παρατηρούμε ότι λείπει ο κώδικας που καθαρίζει το υπάρχον γράφημα από το zedgraph control. Αυτό γίνεται διότι θα προσθέσουμε το γράφημα που φτιάχνουμε πάνω στο γράφημα της απλής Διαδρομής.

```

Dim mypane As ZedGraph.GraphPane = New ZedGraph.GraphPane
mypane = Me.ZedGraphControl1.GraphPane
mypane.Title.Text = tldiadromi
mypane.XAxis.Title.Text = tlplatos
mypane.YAxis.Title.Text = tlmikos

```

```

Dim list1 As ZedGraph.PointPairList = New ZedGraph.PointPairList
Dim list2 As ZedGraph.PointPairList = New ZedGraph.PointPairList

Dim i, j As Integer
Dim temp As Integer = -1
j = 0
For i = 0 To general2.Rows.Count - 1
    If temp <> general2.Rows(i).Item(0) Then
        j = general2.Rows(i).Item(0) - 1
        If general2.Rows(j).Item(9) <> 1 Then
            list2.Add(general2.Rows(j).Item(2), general2.Rows(j).
Item(1))
        Else
            list1.Add(general2.Rows(j).Item(2), general2.Rows(j).
Item(1))
        End If
    End If
    temp = general2.Rows(i).Item(0)
Next i

Dim mycurve1 As ZedGraph.LineItem = mypane.AddCurve("Normal
Release", list1, Color.Green, ZedGraph.SymbolType.Circle)
mycurve1.Symbol.Fill = New ZedGraph.Fill(Color.Green)
mycurve1.Symbol.Size = 4
Dim mycurve2 As ZedGraph.LineItem = mypane.AddCurve("Drop Call",
list2, Color.Red, ZedGraph.SymbolType.Circle)
mycurve2.Symbol.Fill = New ZedGraph.Fill(Color.Red)
mycurve2.Symbol.Size = 4
mypane.Fill = New ZedGraph.Fill(Color.WhiteSmoke, Color.Lavender,
0.0F)
mypane.YAxis.Scale.FontSpec.FontColor = Color.Black
mypane.YAxis.Scale.FontSpec.Size = 8
mypane.YAxis.Scale.FontSpec.IsBold = False
mypane.XAxis.Scale.FontSpec.FontColor = Color.Black
mypane.XAxis.Scale.FontSpec.Size = 8
mypane.XAxis.Scale.FontSpec.IsBold = False
mypane.XAxis.Title.FontSpec.Size = 12
mypane.XAxis.Title.FontSpec.IsItalic = True
mypane.XAxis.Title.FontSpec.FontColor = Color.Chocolate
mypane.YAxis.Title.FontSpec.Size = 12
mypane.YAxis.Title.FontSpec.IsItalic = True
mypane.YAxis.Title.FontSpec.FontColor = Color.Chocolate
mycurve1.Line.IsVisible = False
mycurve2.Line.IsVisible = False
mypane.XAxis.MajorGrid.IsVisible = True
mypane.YAxis.MajorGrid.IsVisible = True
mypane.Y2Axis.Scale.FontSpec.FontColor = Color.Blue
mypane.Y2Axis.Title.FontSpec.FontColor = Color.Blue
mypane.Chart.Fill = New ZedGraph.Fill(Color.White, Color.LightGray,
45.0F)
Dim g As Graphics = Me.CreateGraphics
mypane.AxisChange(g)
Me.ZedGraphControl1.AxisChange()
g.Dispose()
End Sub

```

Ελέγχουμε το ένατο αντικείμενο κάθε γραμμής του general2(πίνακας που έχουμε αποθηκεύσει τα γεγονότα CAD) και ανάλογα με την τιμή του, εισάγουμε τα γεωγραφικά μήκη και πλάτη στα list1 και list2.

### ***Call Failed :***

```

Sub cafroute()
    SimpleRoute()

```

```

Dim mypane As ZedGraph.GraphPane = New ZedGraph.GraphPane
mypane = Me.ZedGraphControll1.GraphPane
mypane.Title.Text = tldiadromi
mypane.XAxis.Title.Text = tlplatos
mypane.YAxis.Title.Text = tlmikos

Dim text As TextObj
text = New TextObj(tltmima, 0.93F, -0.04F)
text.Location.CoordinateFrame = CoordType.ChartFraction
text.FontSpec.Angle = 15.0F
text.FontSpec.FontColor = Color.DarkGray
text.FontSpec.IsBold = True
text.FontSpec.Size = 10
text.FontSpec.Border.IsVisible = False
text.FontSpec.Fill.IsVisible = False
text.Location.AlignH = AlignH.Center
text.Location.AlignV = AlignV.Bottom
mypane.GraphObjList.Add(text)

Dim list1 As ZedGraph.PointPairList = New ZedGraph.PointPairList

Dim i, j As Integer
Dim temp As Integer = -1

j = 0
For i = 0 To generalcaf.Rows.Count - 1
    If temp <> generalcaf.Rows(i).Item(0)
Then
        j = generalcaf.Rows(i).Item(0) - 1
        list1.Add(generalcaf.Rows(j).Item(2), generalcaf.Rows(j).
Item(1))
        End If
        temp = generalcaf.Rows(i).Item(0)
    Next i

Dim mycurve1 As ZedGraph.LineItem = mypane.AddCurve("Access
Failure", list1, Color.Red, ZedGraph.SymbolType.Circle)
mycurve1.Symbol.Fill = New ZedGraph.Fill(Color.Red)
mycurve1.Symbol.Size = 4
mypane.Fill = New ZedGraph.Fill(Color.WhiteSmoke, Color.Lavender,
0.0F)
mypane.YAxis.Scale.FontSpec.FontColor = Color.Black
mypane.YAxis.Scale.FontSpec.Size = 8
mypane.YAxis.Scale.FontSpec.IsBold = False
mypane.XAxis.Scale.FontSpec.FontColor = Color.Black
mypane.XAxis.Scale.FontSpec.Size = 8
mypane.XAxis.Scale.FontSpec.IsBold = False
mypane.XAxis.Title.FontSpec.Size = 12
mypane.XAxis.Title.FontSpec.IsItalic = True
mypane.XAxis.Title.FontSpec.FontColor = Color.Chocolate
mypane.YAxis.Title.FontSpec.Size = 12
mypane.YAxis.Title.FontSpec.IsItalic = True
mypane.YAxis.Title.FontSpec.FontColor = Color.Chocolate
mycurve1.Line.IsVisible = False
mypane.XAxis.MajorGrid.IsVisible = True
mypane.YAxis.MajorGrid.IsVisible = True
mypane.Chart.Fill = New ZedGraph.Fill(Color.White, Color.LightGray,
45.0F)
Dim g As Graphics = Me.CreateGraphics
mypane.AxisChange(g)
Me.ZedGraphControll1.AxisChange()

```

Εισάγουμε τα γεωγραφικά μήκη και πλάτη στο list1 από τον πίνακα generalcaf (πίνακας που έχουμε αποθηκεύσει τα γεγονότα CAF)

```
g.Dispose()
End Sub
```

### Παρεμβολή :

```
Sub routeinter()
    ZedGraphControll1.Visible = True
    Dim oldPane As ZedGraph.GraphPane = ZedGraphControll1.GraphPane
    ZedGraphControll1.GraphPane = New ZedGraph.GraphPane(oldPane.Rect,
oldPane.Title.Text, oldPane.XAxis.Title.Text,
oldPane.YAxis.Title.Text)

    Dim mypane As ZedGraph.GraphPane = New ZedGraph.GraphPane
    mypane = Me.ZedGraphControll1.GraphPane
    mypane.Title.Text = tldiadromi
    mypane.XAxis.Title.Text = tlplatos
    mypane.YAxis.Title.Text = tlmikos

    Dim text As TextObj
    text = New TextObj(tltmima, 0.93F, -0.04F)
    text.Location.CoordinateFrame = CoordType.ChartFraction
    text.FontSpec.Angle = 15.0F
    text.FontSpec.FontColor = Color.DarkGray
    text.FontSpec.IsBold = True
    text.FontSpec.Size = 10
    text.FontSpec.Border.IsVisible = False
    text.FontSpec.Fill.IsVisible = False
    text.Location.AlignH = AlignH.Center
    text.Location.AlignV = AlignV.Bottom
    mypane.GraphObjList.Add(text)

    Dim list1 As ZedGraph.PointPairList = New ZedGraph.PointPairList
    Dim list2 As ZedGraph.PointPairList = New ZedGraph.PointPairList

    Dim rscp, ecn0 As Integer
    Dim i, j, count As Integer
    Dim interpoints(10000) As String
    Dim temp As Integer = -1
    j = 0
    count = 0
    If Form2.CheckBox10.Checked = True Then
        rscp = Int32.Parse(Form2.TextBox1.Text)
        ecn0 = Int32.Parse(Form2.TextBox2.Text)
    Else
        rscp = -86
        ecn0 = -11
    End If

    For i = 0 To mytable.Rows.Count - 1
        If temp <> mytable.Rows(i).Item(0) And mytable.Rows(i).Item(1) =
"Active" Then
            j = mytable.Rows(i).Item(0) - 1
            If mytable.Rows(i).Item(6) > rscp And mytable.Rows(i).Item(4) <
ecn0 Then
                list2.Add(general.Rows(j).Item(2), general.Rows(j).Item(1),
"Rscp:" & mytable.Rows(i).Item(6) & "" & Chr(10) & "Ec/No:" &
mytable.Rows(i).Item(4) & "")
                count = count + 1
                interpoints(count) = mytable.Rows(i).Item(0)
            Else
                list1.Add(general.Rows(j).Item(2), general.Rows(j).Item(1))
            End If
        End If
    End For
End Sub
```

Με το If ελέγχουμε αν το checkbox10 είναι επιλεγμένο. Αν είναι, τότε τα rscp και ecn0 παίρνουν τις τιμές που έχει διαλέξει ο χρήστης. Αν όχι, παίρνουν τις default τιμές.

Στην συνέχεια συγκρίνουμε τις παραπάνω τιμές με τις ανάλογες τιμές κάθε γραμμής ECNO(MONO ΓΙΑ ΤΟΝ 1<sup>0</sup> ACTIVE) και βάσει του αποτελέσματος τοποθετούνται οι συντεταγμένες τους στην ανάλογη λίστα.

```

    End If
    temp = mytable.Rows(i).Item(0)
Next i

Dim start As TextObj = New TextObj("Αρχή", general.Rows(0).Item(2),
general.Rows(0).Item(1))
start.FontSpec.FontColor = Color.DarkMagenta
start.FontSpec.Border.IsVisible = True
start.FontSpec.Fill.IsVisible = True
start.FontSpec.Size = 8
mypane.GraphObjList.Add(start)

Dim mycurve1 As ZedGraph.LineItem = mypane.AddCurve("Διαδρομή",
list1, Color.Blue, ZedGraph.SymbolType.Circle)
mycurve1.Symbol.Fill = New ZedGraph.Fill(Color.Blue)
mycurve1.Symbol.Size = 1
Dim mycurve2 As ZedGraph.LineItem = mypane.AddCurve("Παρεμβολή",
list2, Color.Red, ZedGraph.SymbolType.Diamond)
mycurve2.Symbol.Fill = New ZedGraph.Fill(Color.Red)
mycurve2.Symbol.Size = 3

mypane.Fill = New ZedGraph.Fill(Color.WhiteSmoke, Color.Lavender,
0.0F)
mypane.YAxis.Scale.FontSpec.FontColor = Color.Black
mypane.YAxis.Scale.FontSpec.Size = 8
mypane.YAxis.Scale.FontSpec.IsBold = False
mypane.XAxis.Scale.FontSpec.FontColor = Color.Black
mypane.XAxis.Scale.FontSpec.Size = 8
mypane.XAxis.Scale.FontSpec.IsBold = False
mypane.XAxis.Title.FontSpec.Size = 12
mypane.XAxis.Title.FontSpec.IsItalic = True
mypane.XAxis.Title.FontSpec.FontColor = Color.Chocolate
mypane.YAxis.Title.FontSpec.Size = 12
mypane.YAxis.Title.FontSpec.IsItalic = True
mypane.YAxis.Title.FontSpec.FontColor = Color.Chocolate
mycurve1.Line.IsVisible = False
mycurve2.Line.IsVisible = False
mypane.XAxis.MajorGrid.IsVisible = True
mypane.YAxis.MajorGrid.IsVisible = True
mypane.Chart.Fill = New ZedGraph.Fill(Color.White, Color.LightGray,
45.0F)
Dim g As Graphics = Me.CreateGraphics
mypane.AxisChange(g)
Me.ZedGraphControl1.AxisChange()
g.Dispose()
MsgBox(" Βρέθηκαν " & count & " σημεία παρεμβολής.", MsgBoxStyle.
Information, "3G Analysys")
End Sub

```

### ***Rscp 1<sup>ο</sup> Active server :***

```

Sub RSCP()
    ZedGraphControl1.Visible = True
    Dim oldPane As ZedGraph.GraphPane = ZedGraphControl1.GraphPane
    ZedGraphControl1.GraphPane = New ZedGraph.GraphPane(oldPane.Rect,
oldPane.Title.Text, oldPane.XAxis.Title.Text,
oldPane.YAxis.Title.Text)

    Dim mypane As ZedGraph.GraphPane = New ZedGraph.GraphPane
    mypane = Me.ZedGraphControl1.GraphPane

```

```

mypane.Title.Text = tlrsproute
mypane.XAxis.Title.Text = tlmetriseis
mypane.YAxis.Title.Text = tlrsctp

Dim text As TextObj
text = New TextObj(tltmima, 0.93F, -0.04F)
text.Location.CoordinateFrame = CoordType.ChartFraction
text.FontSpec.Angle = 15.0F
text.FontSpec.FontColor = Color.DarkGray
text.FontSpec.IsBold = True
text.FontSpec.Size = 10
text.FontSpec.Border.IsVisible = False
text.FontSpec.Fill.IsVisible = False
text.Location.AlignH = AlignH.Center
text.Location.AlignV = AlignV.Bottom
mypane.GraphObjList.Add(text)

Dim list1 As ZedGraph.PointPairList = New ZedGraph.PointPairList
Dim check As Integer = -1
Dim i, j As Integer
j = 0

For i = 0 To mytable.Rows.Count - 1
    If check <> mytable.Rows(i).Item(0) And mytable.Rows(i).Item(1) =
"Active" Then
        j = mytable.Rows(i).Item(0) - 1
        list1.Add(general.Rows(j).Item(0), mytable.Rows(i).Item(6))
    End If
    check = mytable.Rows(i).Item(0)
Next i

    Σε αυτό το σημείο ελέγχουμε κάθε γραμμή γεγονότος Ec/No και για
    κάθε 1° Active server που βρίσκουμε εισάγουμε στην λίστα τον A/A
    της μέτρησης και την τιμή του Rscp. Συνεπώς θα φτιάξουμε μια
    γραφική παράσταση που θα μας δείχνει την μεταβολή της
    λαμβανόμενης ισχύος συναρτήσει του χρόνου.

Dim myCurve1 As ZedGraph.LineItem = mypane.AddCurve(tlrsctp, list1,
Color.Blue, ZedGraph.SymbolType.Circle)
myCurve1.Symbol.Fill = New ZedGraph.Fill(Color.Blue)
myCurve1.Symbol.Size = 1

mypane.Fill = New ZedGraph.Fill(Color.WhiteSmoke, Color.Lavender,
0.0F)
mypane.YAxis.Scale.FontSpec.FontColor = Color.Black
mypane.YAxis.Scale.FontSpec.Size = 8
mypane.YAxis.Scale.FontSpec.IsBold = False
mypane.XAxis.Scale.FontSpec.FontColor = Color.Black
mypane.XAxis.Scale.FontSpec.Size = 8
mypane.XAxis.Scale.FontSpec.IsBold = False
mypane.XAxis.Title.FontSpec.Size = 12
mypane.XAxis.Title.FontSpec.IsItalic = True
mypane.XAxis.Title.FontSpec.FontColor = Color.Chocolate
mypane.YAxis.Title.FontSpec.Size = 12
mypane.YAxis.Title.FontSpec.IsItalic = True
mypane.YAxis.Title.FontSpec.FontColor = Color.Chocolate
myCurve1.Line.IsVisible = True
mypane.XAxis.MajorGrid.IsVisible = True
mypane.YAxis.MajorGrid.IsVisible = True
mypane.Y2Axis.IsVisible = False
mypane.Y2Axis.Scale.FontSpec.FontColor = Color.Blue
mypane.Y2Axis.Title.FontSpec.FontColor = Color.Blue

```

```

mypane.Chart.Fill = New ZedGraph.Fill(Color.White, Color.LightGray,
45.0F)
Dim g As Graphics = Me.CreateGraphics
mypane.AxisChange(g)
Me.ZedGraphControl1.AxisChange()
g.Dispose()
End Sub

```

### ***Rscp 1<sup>ο</sup>* και 2<sup>ο</sup> Active server :**

```

Sub rscp2()
    ZedGraphControl1.Visible = True
    Dim oldPane As ZedGraph.GraphPane = ZedGraphControl1.GraphPane
    ZedGraphControl1.GraphPane = New ZedGraph.GraphPane(oldPane.Rect,
oldPane.Title.Text, oldPane.XAxis.Title.Text,
oldPane.YAxis.Title.Text)

    Dim mypane As ZedGraph.GraphPane = New ZedGraph.GraphPane
    mypane = Me.ZedGraphControl1.GraphPane

    mypane.Title.Text = tlrsctroute
    mypane.XAxis.Title.Text = tlmetriseis
    mypane.YAxis.Title.Text = tlrsctp

    Dim text As TextObj
    text = New TextObj(tltmima, 0.93F, -0.04F)
    text.Location.CoordinateFrame = CoordType.ChartFraction
    text.FontSpec.Angle = 15.0F
    text.FontSpec.FontColor = Color.DarkGray
    text.FontSpec.IsBold = True
    text.FontSpec.Size = 10
    text.FontSpec.Border.IsVisible = False
    text.FontSpec.Fill.IsVisible = False
    text.Location.AlignH = AlignH.Center
    text.Location.AlignV = AlignV.Bottom
    mypane.GraphObjList.Add(text)

    Dim list1 As ZedGraph.PointPairList = New ZedGraph.PointPairList
    Dim list2 As ZedGraph.PointPairList = New ZedGraph.PointPairList

    Dim check As Integer = -1
    Dim i, j As Integer
    j = 0

```

Σε αυτό το σημείο επαναλαμβάνουμε την ίδια διαδικασία προσθήκης τιμών στις λίστες όπως με το RSCP και επιπλέον ελέγχουμε αν υπάρχει 2<sup>ο</sup>s Active. Αν υπάρχει προσθέτουμε την τιμή την του rscp στο list1.

```

    For i = 0 To mytable.Rows.Count - 1
        If check <> mytable.Rows(i).Item(0) Then
            j = mytable.Rows(i).Item(0) - 1
            list2.Add(general.Rows(j).Item(0), mytable.Rows(i).Item(6),
"Μέτρηση:" & j + 1 & "" & Chr(10) & "Scrambling Code:" &
mytable.Rows(i).Item(3) & "")
            If i > mytable.Rows.Count - 2 Then
                Else
                    If mytable.Rows(i + 1).Item(1) = "Active" And mytable.Rows(i
+ 1).Item(0) = mytable.Rows(i).Item(0) Then
                        list1.Add(general.Rows(j).Item(0), mytable.Rows(i +
1).Item(6), "Μέτρηση:" & j + 1 & "" & Chr(10) & "Scrambling Code:" &
mytable.Rows(i + 1).Item(3) & "")

```



```

        End If
    End If
End If
check = mytable.Rows(i).Item(0)
Next i

Dim myCurve2 As ZedGraph.LineItem = mypane.AddCurve("1ος Active",
list2, Color.Blue, ZedGraph.SymbolType.Circle)
myCurve2.Symbol.Fill = New ZedGraph.Fill(Color.Blue)
myCurve2.Symbol.Size = 1
Dim myCurve1 As ZedGraph.LineItem = mypane.AddCurve("2ος Active",
list1, Color.Green, ZedGraph.SymbolType.Circle)
myCurve1.Symbol.Fill = New ZedGraph.Fill(Color.Green)
myCurve1.Symbol.Size = 2
mypane.Fill = New ZedGraph.Fill(Color.WhiteSmoke, Color.Lavender,
0.0F)
mypane.YAxis.Scale.FontSpec.FontColor = Color.Black
mypane.YAxis.Scale.FontSpec.Size = 8
mypane.YAxis.Scale.FontSpec.IsBold = False
mypane.XAxis.Scale.FontSpec.FontColor = Color.Black
mypane.XAxis.Scale.FontSpec.Size = 8
mypane.XAxis.Scale.FontSpec.IsBold = False
mypane.XAxis.Title.FontSpec.Size = 12
mypane.XAxis.Title.FontSpec.IsItalic = True
mypane.XAxis.Title.FontSpec.FontColor = Color.Chocolate
mypane.YAxis.Title.FontSpec.Size = 12
mypane.YAxis.Title.FontSpec.IsItalic = True
mypane.YAxis.Title.FontSpec.FontColor = Color.Chocolate
myCurve1.Line.IsVisible = False
myCurve2.Line.IsVisible = True
mypane.XAxis.MajorGrid.IsVisible = True
mypane.YAxis.MajorGrid.IsVisible = True
mypane.Chart.Fill = New ZedGraph.Fill(Color.White, Color.LightGray,
45.0F)
Dim g As Graphics = Me.CreateGraphics
mypane.AxisChange(g)
Me.ZedGraphControl1.AxisChange()
g.Dispose()
End Sub

```

### Καλύτεροι 5 servers :

```

Sub test5best()
    Dim array As New ArrayList
    Δηλώνεται το array ως ArrayList.
    Dim i, check As Integer
    best.Clear()
    For i = 0 To mytable.Rows.Count - 1
        array.Clear()
        Γίνεται καθαρισμός του array.
        check = mytable.Rows(i).Item(0)
        Do While check =
mytable.Rows(i).Item(0)
            array.Add(mytable.Rows(i).Item(6))
            i = i + 1
            If i = mytable.Rows.Count Then
                Exit Do
            End If
        Loop
        i = i - 1
    
```

Η διαδικασία που ακολουθείται για την εύρεση των πέντε καλύτερων servers είναι η εξής :

Όπως πάντα ξεκινάμε από την 1<sup>η</sup> γραμμή του πίνακα mytable και εισάγουμε στο check τον A/A της πρώτης μέτρησης. Στην συνέχεια όσο βρισκόμαστε στην 1<sup>η</sup> μέτρηση προσθέτουμε στο array τις τιμές του rscr.

Με το array.Sort() ταξινομούμε σε σειρά τις τιμές που περιέχει το array και επειδή είναι αρνητικές, αντιστρέφουμε την σειρά τους με την array.Reverse().

Ελέγχουμε το πλήθος του array και ανάλογα της τιμής του κάνουμε τα ανάλογα.

Τέλος μεταφέρουμε τα δεδομένα στον πίνακα best.

```

array.Sort()
array.Reverse()

While array.Count > 5
    array.RemoveAt(array.Count - 1)
End While

While array.Count < 5
    array.Add(0)
End While

Dim values As Object()
values = array.ToArray
best.Rows.Add(values)
Next i

ZedGraphControll1.Visible = True
Dim oldpane As ZedGraph.GraphPane = ZedGraphControll1.GraphPane
ZedGraphControll1.GraphPane = New ZedGraph.GraphPane(oldpane.Rect,
oldpane.Title.Text, oldpane.XAxis.Title.Text,
oldpane.YAxis.Title.Text)

Dim mypane As ZedGraph.GraphPane = New ZedGraph.GraphPane
mypane = Me.ZedGraphControll1.GraphPane

mypane.Title.Text = tlbest5
mypane.XAxis.Title.Text = tlmeasurement
mypane.YAxis.Title.Text = tlrscp

Dim text As TextObj
text = New TextObj(tlmtima, 0.93F, -0.04F)
text.Location.CoordinateFrame = CoordType.ChartFraction
text.FontSpec.Angle = 15.0F
text.FontSpec.FontColor = Color.DarkGray
text.FontSpec.IsBold = True
text.FontSpec.Size = 10
text.FontSpec.Border.IsVisible = False
text.FontSpec.Fill.IsVisible = False
text.Location.AlignH = AlignH.Center
text.Location.AlignV = AlignV.Bottom
mypane.GraphObjList.Add(text)

Dim list10 As ZedGraph.PointPairList = New ZedGraph.PointPairList
Dim list11 As ZedGraph.PointPairList = New ZedGraph.PointPairList
Dim list12 As ZedGraph.PointPairList = New ZedGraph.PointPairList
Dim list13 As ZedGraph.PointPairList = New ZedGraph.PointPairList
Dim list14 As ZedGraph.PointPairList = New ZedGraph.PointPairList
Dim v, h As Integer
Dim temp2 As Integer = -1
h = 0
For v = 0 To best.Rows.Count - 1
    If best.Rows(v).Item(0) <> 0 Then
        list10.Add(general.Rows(v).Item(0), best.Rows(v).Item(0))
    End If
    If best.Rows(v).Item(1) <> 0 Then
        list11.Add(general.Rows(v).Item(0), best.Rows(v).Item(1))
    End If
    If best.Rows(v).Item(2) <> 0 Then
        list12.Add(general.Rows(v).Item(0), best.Rows(v).Item(2))
    End If
    If best.Rows(v).Item(3) <> 0 Then

```

```

        list13.Add(general.Rows(v).Item(0), best.Rows(v).Item(3))
    End If
    If best.Rows(v).Item(4) <> 0 Then
        list14.Add(general.Rows(v).Item(0), best.Rows(v).Item(4))
    End If
Next v

Dim mycurve1 As ZedGraph.LineItem = mypane.AddCurve("Πρώτος",
list10, Color.Blue, ZedGraph.SymbolType.Circle)
Dim mycurve2 As ZedGraph.LineItem = mypane.AddCurve("Δεύτερος",
list11, Color.Green, ZedGraph.SymbolType.Circle)
Dim mycurve3 As ZedGraph.LineItem = mypane.AddCurve("Τρίτος",
list12, Color.Red, ZedGraph.SymbolType.Circle)
Dim mycurve4 As ZedGraph.LineItem = mypane.AddCurve("Τέταρτος",
list13, Color.Yellow, ZedGraph.SymbolType.Circle)
Dim mycurve5 As ZedGraph.LineItem = mypane.AddCurve("Πέμπτος",
list14, Color.Orange, ZedGraph.SymbolType.Circle)

mycurve1.Symbol.Fill = New ZedGraph.Fill(Color.Blue)
mycurve1.Symbol.Size = 1
mycurve2.Symbol.Fill = New ZedGraph.Fill(Color.Green)
mycurve2.Symbol.Size = 1
mycurve3.Symbol.Fill = New ZedGraph.Fill(Color.Red)
mycurve3.Symbol.Size = 1
mycurve4.Symbol.Fill = New ZedGraph.Fill(Color.Yellow)
mycurve4.Symbol.Size = 1
mycurve5.Symbol.Fill = New ZedGraph.Fill(Color.Orange)
mypane.Fill = New ZedGraph.Fill(Color.WhiteSmoke, Color.Lavender,
0.0F)
mypane.YAxis.Scale.FontSpec.FontColor = Color.Black
mypane.YAxis.Scale.FontSpec.Size = 8
mypane.YAxis.Scale.FontSpec.IsBold = False
mypane.XAxis.Scale.FontSpec.FontColor = Color.Black
mypane.XAxis.Scale.FontSpec.Size = 8
mypane.XAxis.Scale.FontSpec.IsBold = False
mypane.XAxis.Title.FontSpec.Size = 12
mypane.XAxis.Title.FontSpec.IsItalic = True
mypane.XAxis.Title.FontSpec.FontColor = Color.Chocolate
mypane.YAxis.Title.FontSpec.Size = 12
mypane.YAxis.Title.FontSpec.IsItalic = True
mypane.YAxis.Title.FontSpec.FontColor = Color.Chocolate
mycurve1.Line.IsVisible = True
mycurve2.Line.IsVisible = False
mycurve3.Line.IsVisible = False
mycurve4.Line.IsVisible = False
mycurve5.Line.IsVisible = False
mypane.XAxis.MajorGrid.IsVisible = True
mypane.YAxis.MajorGrid.IsVisible = True
mypane.Chart.Fill = New ZedGraph.Fill(Color.White, Color.LightGray,
45.0F)
Dim g As Graphics = Me.CreateGraphics
mypane.AxisChange(g)
Me.ZedGraphControll1.AxisChange()
g.Dispose()
End Sub

```

### ***Μελέτη των αλλαγών των Scrambling Codes στον Io Active Server :***

```

Sub rscpscrcode()
    ZedGraphControll1.Visible = True

```

```
Dim oldPane As ZedGraph.GraphPane = ZedGraphControll1.GraphPane
ZedGraphControll1.GraphPane = New ZedGraph.GraphPane(oldPane.Rect,
oldPane.Title.Text, oldPane.XAxis.Title.Text,
oldPane.YAxis.Title.Text)
```

```
Dim mypane As ZedGraph.GraphPane = New ZedGraph.GraphPane
mypane = Me.ZedGraphControll1.GraphPane
```

```
mypane.Title.Text = tlrscproute
mypane.XAxis.Title.Text = tlmetriseis
mypane.YAxis.Title.Text = tlrscp
```

```
Dim text As TextObj
text = New TextObj(tltmima, 0.93F, -0.04F)
text.Location.CoordinateFrame = CoordType.ChartFraction
text.FontSpec.Angle = 15.0F
text.FontSpec.FontColor = Color.DarkGray
text.FontSpec.IsBold = True
text.FontSpec.Size = 10
text.FontSpec.Border.IsVisible = False
text.FontSpec.Fill.IsVisible = False
text.Location.AlignH = AlignH.Center
text.Location.AlignV = AlignV.Bottom
mypane.GraphObjList.Add(text)
```

```
Dim list1 As PointPairList = New ZedGraph.PointPairList
Dim list2 As PointPairList = New ZedGraph.PointPairList
Dim check As Integer = -1
Dim temp As Integer
Dim i, j As Integer
j = i = 0
temp = mytable.Rows(i).Item(3)
```

Σε αυτό το σημείο επαναλαμβάνουμε την ίδια διαδικασία προσθήκης τιμών στις λίστες όπως με το RSCP και επιπλέον ελέγχουμε αν αλλάζει το scrambling code. Αν αλλάζει τότε προσθέτουμε την τιμή την του rscp στο list2.

```
For i = 0 To mytable.Rows.Count - 1
  If check <> mytable.Rows(i).Item(0) And mytable.Rows(i).Item(1) =
"Active" Then
    j = mytable.Rows(i).Item(0) - 1
    list1.Add(general.Rows(j).Item(0), mytable.Rows(i).Item(6),
"Μέτρηση:" & j + 1 & "" & Chr(10) & "Scrambling Code:" &
mytable.Rows(i).Item(3) & "")
```

```
  If temp <> mytable.Rows(i).Item(3) Then
    list2.Add(general.Rows(j).Item(0), mytable.Rows(i).Item(6),
"Μέτρηση:" & j + 1 & "" & Chr(10) & "Scrambling Code:" &
mytable.Rows(i).Item(3) & "")
```

```
  End If
  temp = mytable.Rows(i).Item(3)
End If
check = mytable.Rows(i).Item(0)
Next i
```

```
Dim myCurve1 As ZedGraph.LineItem = mypane.AddCurve("RSCP", list1,
Color.Blue, ZedGraph.SymbolType.Circle)
myCurve1.Symbol.Fill = New ZedGraph.Fill(Color.Blue)
myCurve1.Symbol.Size = 1
```

```

Dim myCurve2 As ZedGraph.LineItem = mypane.AddCurve("Αλλαγή
Scrambling Code (Handover)", list2, Color.Red,
ZedGraph.SymbolType.Circle)
myCurve2.Symbol.Fill = New ZedGraph.Fill(Color.Red)
myCurve2.Symbol.Size = 3
mypane.Fill = New ZedGraph.Fill(Color.WhiteSmoke, Color.Lavender,
0.0F)
mypane.YAxis.Scale.FontSpec.FontColor = Color.Black
mypane.YAxis.Scale.FontSpec.Size = 8
mypane.YAxis.Scale.FontSpec.IsBold = False
mypane.XAxis.Scale.FontSpec.FontColor = Color.Black
mypane.XAxis.Scale.FontSpec.Size = 8
mypane.XAxis.Scale.FontSpec.IsBold = False
mypane.XAxis.Title.FontSpec.Size = 12
mypane.XAxis.Title.FontSpec.IsItalic = True
mypane.XAxis.Title.FontSpec.FontColor = Color.Chocolate
mypane.YAxis.Title.FontSpec.Size = 12
mypane.YAxis.Title.FontSpec.IsItalic = True
mypane.YAxis.Title.FontSpec.FontColor = Color.Chocolate
myCurve1.Line.IsVisible = True
myCurve2.Line.IsVisible = False
mypane.XAxis.MajorGrid.IsVisible = True
mypane.YAxis.MajorGrid.IsVisible = True
mypane.Chart.Fill = New ZedGraph.Fill(Color.White, Color.LightGray,
45.0F)
Dim g As Graphics = Me.CreateGraphics
mypane.AxisChange(g)
Me.ZedGraphControll1.AxisChange()
g.Dispose()
End Sub

```

### ***Μελέτη των αλλαγών των Scrambling Codes στον 1ο και 2ο Active Server :***

```

Sub rscpscrcode2()
ZedGraphControll1.Visible = True
Dim oldPane As ZedGraph.GraphPane = ZedGraphControll1.GraphPane
ZedGraphControll1.GraphPane = New ZedGraph.GraphPane(oldPane.Rect,
oldPane.Title.Text, oldPane.XAxis.Title.Text,
oldPane.YAxis.Title.Text)

Dim mypane As ZedGraph.GraphPane = New ZedGraph.GraphPane
mypane = Me.ZedGraphControll1.GraphPane

mypane.Title.Text = tlrsctroute
mypane.XAxis.Title.Text = tlmetriseis
mypane.YAxis.Title.Text = tlrsctp

Dim text As TextObj
text = New TextObj(tltmima, 0.93F, -0.04F)
text.Location.CoordinateFrame = CoordType.ChartFraction
text.FontSpec.Angle = 15.0F
text.FontSpec.FontColor = Color.DarkGray
text.FontSpec.IsBold = True
text.FontSpec.Size = 10
text.FontSpec.Border.IsVisible = False
text.FontSpec.Fill.IsVisible = False
text.Location.AlignH = AlignH.Center
text.Location.AlignV = AlignV.Bottom
mypane.GraphObjList.Add(text)

```

```

Dim list1 As ZedGraph.PointPairList = New ZedGraph.PointPairList
Dim list2 As ZedGraph.PointPairList = New ZedGraph.PointPairList
Dim list3 As ZedGraph.PointPairList = New ZedGraph.PointPairList

Dim temp As Integer
Dim check As Integer = -1
Dim i, j As Integer
i = 0
temp = mytable.Rows(i).Item(3)
j = 0

```

Σε αυτό το σημείο επαναλαμβάνουμε την ίδια διαδικασία προσθήκης τιμών στις λίστες όπως με το RSCP2 και επιπλέον ελέγχουμε αν αλλάζει το scrambling code. Αν αλλάζει τότε προσθέτουμε την τιμή την του rscp στο list3.

```

For i = 0 To mytable.Rows.Count - 1
    If check <> mytable.Rows(i).Item(0) And mytable.Rows(i).Item(1) =
"Active" Then
        j = mytable.Rows(i).Item(0) - 1
        list2.Add(general.Rows(j).Item(0), mytable.Rows(i).Item(6),
"Μέτρηση:" & j + 1 & "" & Chr(10) & "Scrambling Code:" &
mytable.Rows(i).Item(3) & "")
        If temp <> mytable.Rows(i).Item(3) Then
            list3.Add(general.Rows(j).Item(0), mytable.Rows(i).Item(6),
"Μέτρηση:" & j + 1 & "" & Chr(10) & "Scrambling Code:" &
mytable.Rows(i).Item(3) & "")
        End If
        If i > mytable.Rows.Count - 2 Then
            Else
                If mytable.Rows(i + 1).Item(1) = "Active" And mytable.Rows(i
+ 1).Item(0) = mytable.Rows(i).Item(0) Then
                    list1.Add(general.Rows(j).Item(0), mytable.Rows(i +
1).Item(6), "Μέτρηση:" & j + 1 & "" & Chr(10) & "Scrambling Code:" &
mytable.Rows(i + 1).Item(3) & "")
                End If
            End If
            temp = mytable.Rows(i).Item(3)
        End If
        check = mytable.Rows(i).Item(0)
    Next i

Dim myCurve2 As ZedGraph.LineItem = mypane.AddCurve("1ος Active",
list2, Color.Blue, ZedGraph.SymbolType.Circle)
myCurve2.Symbol.Fill = New ZedGraph.Fill(Color.Blue)
myCurve2.Symbol.Size = 1
Dim myCurve1 As ZedGraph.LineItem = mypane.AddCurve("2ος Active",
list1, Color.Green, ZedGraph.SymbolType.Circle)
myCurve1.Symbol.Fill = New ZedGraph.Fill(Color.Green)
myCurve1.Symbol.Size = 2
Dim myCurve3 As ZedGraph.LineItem = mypane.AddCurve("Αλλαγή
Scrambling Code (Handover)", list3, Color.Red,
ZedGraph.SymbolType.Circle)
myCurve3.Symbol.Fill = New ZedGraph.Fill(Color.Red)
myCurve3.Symbol.Size = 3

mypane.Fill = New ZedGraph.Fill(Color.WhiteSmoke, Color.Lavender,
0.0F)
mypane.YAxis.Scale.FontSpec.FontColor = Color.Black
mypane.YAxis.Scale.FontSpec.Size = 8
mypane.YAxis.Scale.FontSpec.IsBold = False

```

```

mypane.XAxis.Scale.FontSpec.FontColor = Color.Black
mypane.XAxis.Scale.FontSpec.Size = 8
mypane.XAxis.Scale.FontSpec.IsBold = False
mypane.XAxis.Title.FontSpec.Size = 12
mypane.XAxis.Title.FontSpec.IsItalic = True
mypane.XAxis.Title.FontSpec.FontColor = Color.Chocolate
mypane.YAxis.Title.FontSpec.Size = 12
mypane.YAxis.Title.FontSpec.IsItalic = True
mypane.YAxis.Title.FontSpec.FontColor = Color.Chocolate
myCurve1.Line.IsVisible = False
myCurve2.Line.IsVisible = True
myCurve3.Line.IsVisible = False
mypane.XAxis.MajorGrid.IsVisible = True
mypane.YAxis.MajorGrid.IsVisible = True
mypane.Chart.Fill = New ZedGraph.Fill(Color.White, Color.LightGray,
45.0F)
Dim g As Graphics = Me.CreateGraphics
mypane.AxisChange(g)
Me.ZedGraphControl1.AxisChange()
g.Dispose()
End Sub

```

### Παρεμβολή :

```

Sub rscpecn0inter()
Dim rscp, ecn0 As Integer
If Form2.CheckBox12.Checked = True Then
rscp = Int32.Parse(Form2.TextBox4.Text)
ecn0 = Int32.Parse(Form2.TextBox3.Text)
Else
rscp = -86
ecn0 = -11
End If

ZedGraphControl1.Visible = True
Dim oldPane As ZedGraph.GraphPane = ZedGraphControl1.GraphPane
ZedGraphControl1.GraphPane = New ZedGraph.GraphPane(oldPane.Rect,
oldPane.Title.Text, oldPane.XAxis.Title.Text,
oldPane.YAxis.Title.Text)
Dim mypane As ZedGraph.GraphPane = New ZedGraph.GraphPane
mypane = Me.ZedGraphControl1.GraphPane
mypane.Title.Text = "Rscp/EcNo για Rscp< " & rscp & " και Ec/No>" &
ecn0 & " "
mypane.XAxis.Title.Text = "ECN0"
mypane.YAxis.Title.Text = "RSCP"

Dim text As TextObj
text = New TextObj("Τμήμα Ηλεκτρονικής " & Chr(10) & " Παράρτημα
Χανίων", 0.93F, -0.04F)
text.Location.CoordinateFrame = CoordType.ChartFraction
text.FontSpec.Angle = 15.0F
text.FontSpec.FontColor = Color.DarkGray
text.FontSpec.IsBold = True
text.FontSpec.Size = 10
text.FontSpec.Border.IsVisible = False
text.FontSpec.Fill.IsVisible = False
text.Location.AlignH = AlignH.Center
text.Location.AlignV = AlignV.Bottom
mypane.GraphObjList.Add(text)

Dim list1 As ZedGraph.PointPairList = New ZedGraph.PointPairList

```

Με το If ελέγχουμε αν το checkbox12 είναι επιλεγμένο. Αν είναι, τότε τα rscp και ecn0 παίρνουν τις τιμές που έχει διαλέξει ο χρήστης. Αν όχι, παίρνουν τις default τιμές.

```

Dim list2 As ZedGraph.PointPairList = New ZedGraph.PointPairList
Dim check As Integer = -1

Dim i, j As Integer
j = 0
For i = 0 To mytable.Rows.Count - 1
    If check <> mytable.Rows(i).Item(0) And mytable.Rows(i).Item(1) =
"Active" Then
        j = mytable.Rows(i).Item(0) - 1
        If mytable.Rows(i).Item(6) > rscp And mytable.Rows(i).Item(4) <
ecn0 Then
            list1.Add(mytable.Rows(i).Item(4), mytable.Rows(i).Item(6))
        Else
            list2.Add(mytable.Rows(i).Item(4), mytable.Rows(i).Item(6))
        End If
    End If
    check = mytable.Rows(i).Item(0)
Next i

Dim myCurve1 As ZedGraph.LineItem = mypane.AddCurve("Παρεμβολή",
list1, Color.Red, ZedGraph.SymbolType.Star)
myCurve1.Symbol.Fill = New ZedGraph.Fill(Color.Red)
myCurve1.Symbol.Size = 2
Dim myCurve2 As ZedGraph.LineItem = mypane.AddCurve("", list2,
Color.Blue, ZedGraph.SymbolType.Circle)
myCurve2.Symbol.Fill = New ZedGraph.Fill(Color.Blue)
myCurve2.Symbol.Size = 2
mypane.Fill = New ZedGraph.Fill(Color.WhiteSmoke, Color.Lavender,
0.0F)
mypane.YAxis.Scale.FontSpec.FontColor = Color.Black
mypane.YAxis.Scale.FontSpec.Size = 8
mypane.YAxis.Scale.FontSpec.IsBold = False
mypane.XAxis.Scale.FontSpec.FontColor = Color.Black
mypane.XAxis.Scale.FontSpec.Size = 8
mypane.XAxis.Scale.FontSpec.IsBold = False
mypane.XAxis.Title.FontSpec.Size = 12
mypane.XAxis.Title.FontSpec.IsItalic = True
mypane.XAxis.Title.FontSpec.FontColor = Color.Chocolate
mypane.YAxis.Title.FontSpec.Size = 12
mypane.YAxis.Title.FontSpec.IsItalic = True
mypane.YAxis.Title.FontSpec.FontColor = Color.Chocolate
myCurve1.Line.IsVisible = False
myCurve2.Line.IsVisible = False
mypane.XAxis.MajorGrid.IsVisible = True
mypane.Chart.Fill = New ZedGraph.Fill(Color.White, Color.LightGray,
45.0F)
Dim g As Graphics = Me.CreateGraphics
mypane.AxisChange(g)
Me.ZedGraphControll1.AxisChange()
g.Dispose()
End Sub

```

### ***Μελέτη σημείων πριν το drop call :***

```

Sub cadpoints()
    SimpleRoute()
    Dim mypane As ZedGraph.GraphPane = New ZedGraph.GraphPane
    mypane = Me.ZedGraphControll1.GraphPane
    mypane.Title.Text = tldiadromi
    mypane.XAxis.Title.Text = tlplatos

```



```

mypane.YAxis.Title.Text = tlmikos

Dim list1 As ZedGraph.PointPairList = New ZedGraph.PointPairList
Dim list2 As ZedGraph.PointPairList = New ZedGraph.PointPairList

Dim ct As Integer = 1
Dim i, j As Integer
Dim temp As Integer = -1
j = 0

For i = 0 To general2.Rows.Count - 1
    If temp <> general2.Rows(i).Item(0) Then
        j = general2.Rows(i).Item(0) - 1
        If general2.Rows(j).Item(9) <> 1 Then
            list2.Add(general2.Rows(j).Item(2), general2.Rows(j).Item(1))
        Else
            list1.Add(general2.Rows(j).Item(2), general2.Rows(j).Item(1))
        End If
    End If
    temp = general2.Rows(i).Item(0)
Next i

Dim xOffset As Single = CType((mypane.XAxis.Scale.Max -
mypane.XAxis.Scale.Min), Single) * 0.01F
Dim yOffset As Single = CType((mypane.YAxis.Scale.Max -
mypane.YAxis.Scale.Min), Single) * 0.01F
For Each pt As PointPair In list2
    Dim label As String = pt.ToString
    Dim text As TextObj = New TextObj("Drop Call " & ct & "",
CType(pt.X, Single) + xOffset, CType(pt.Y, Single) + yOffset,
CoordType.AxisXYScale, AlignH.Left, AlignV.Bottom)
    text.FontSpec.Border.IsVisible = False
    text.Location.AlignH = AlignH.Center
    text.Location.AlignV = AlignV.Bottom
    text.FontSpec.FontColor = Color.Red
    text.FontSpec.Border.IsVisible = True
    text.FontSpec.Fill.IsVisible = True
    text.FontSpec.Size = 7
    mypane.GraphObjList.Add(text)
    ct = ct + 1
Next

Dim start As TextObj = New TextObj("Start Point",
general.Rows(0).Item(2), general.Rows(0).Item(1))
start.FontSpec.FontColor = Color.DarkMagenta
start.FontSpec.Border.IsVisible = True
start.FontSpec.Fill.IsVisible = True
start.FontSpec.Size = 8
mypane.GraphObjList.Add(start)

Dim mycurve1 As ZedGraph.LineItem = mypane.AddCurve("Normal
Release", list1, Color.Green, ZedGraph.SymbolType.Circle)
mycurve1.Symbol.Fill = New ZedGraph.Fill(Color.Green)
mycurve1.Symbol.Size = 4
Dim mycurve2 As ZedGraph.LineItem = mypane.AddCurve("Drop Call",
list2, Color.Red, ZedGraph.SymbolType.Circle)
mycurve2.Symbol.Fill = New ZedGraph.Fill(Color.Red)
mycurve2.Symbol.Size = 4

```

Σε αυτό το σημείο ελέγχουμε το ένατο αντικείμενο του πίνακα general2 και αν είναι διάφορο του 1 (δηλαδή έχουμε drop call, βλέπε παράρτημα .....), βάζουμε τις συντεταγμένες του σημείου στο list2.

Ο κώδικας με τα έντονα γράμματα σχεδιάζει ένα αντικείμενο κειμένου για κάθε σημείο του list2 με τίτλο "Drop Call ct" όπου ct ο αύξων αριθμός του Drop Call.

```

mypane.Fill = New ZedGraph.Fill(Color.WhiteSmoke, Color.Lavender,
0.0F)
mypane.YAxis.Scale.FontSpec.FontColor = Color.Black
mypane.YAxis.Scale.FontSpec.Size = 8
mypane.YAxis.Scale.FontSpec.IsBold = False
mypane.XAxis.Scale.FontSpec.FontColor = Color.Black
mypane.XAxis.Scale.FontSpec.Size = 8
mypane.XAxis.Scale.FontSpec.IsBold = False
mypane.XAxis.Title.FontSpec.Size = 12
mypane.XAxis.Title.FontSpec.IsItalic = True
mypane.XAxis.Title.FontSpec.FontColor = Color.Chocolate
mypane.YAxis.Title.FontSpec.Size = 12
mypane.YAxis.Title.FontSpec.IsItalic = True
mypane.YAxis.Title.FontSpec.FontColor = Color.Chocolate
mycurve1.Line.IsVisible = False
mycurve2.Line.IsVisible = False
mypane.XAxis.MajorGrid.IsVisible = True
mypane.Chart.Fill = New ZedGraph.Fill(Color.White, Color.LightGray,
45.0F)
Dim g As Graphics = Me.CreateGraphics
mypane.AxisChange(g)
Me.ZedGraphControl1.AxisChange()
g.Dispose()
End Sub

```

### **Μελέτη Κυψέλης :**

```

Sub avgbar()
ZedGraphControl1.Visible = True
Dim oldPane As ZedGraph.GraphPane = ZedGraphControl1.GraphPane
ZedGraphControl1.GraphPane = New ZedGraph.GraphPane(oldPane.Rect,
oldPane.Title.Text, oldPane.XAxis.Title.Text,
oldPane.YAxis.Title.Text)
Dim mypane As ZedGraph.GraphPane = New ZedGraph.GraphPane
mypane = Me.ZedGraphControl1.GraphPane

If Form2.Visible = True Then
mypane.Title.Text = "Μελέτη των Rscp & Ec/No της κυψέλης με
scrambling code (" & Form2.ComboBox2.Text & ") "
Else
mypane.Title.Text = "Μελέτη των Rscp & Ec/No της κυψέλης με
scrambling code (" & ComboBox1.Text
& ") "
End If

```

Το παραπάνω if χρησιμοποιείται για να καθοριστεί ο τίτλος του γραφήματος ανάλογα με την πηγή που ο χρήστης έχει επιλέξει το scr. code.

```

mypane.XAxis.Title.Text = ""
mypane.YAxis.Title.Text = ""
Dim text As TextObj
text = New TextObj(tlmtima, 0.93F, -0.04F)
text.Location.CoordinateFrame = CoordType.ChartFraction
text.FontSpec.Angle = 15.0F
text.FontSpec.FontColor = Color.DarkGray
text.FontSpec.IsBold = True
text.FontSpec.Size = 10
text.FontSpec.Border.IsVisible = False
text.FontSpec.Fill.IsVisible = False
text.Location.AlignH = AlignH.Center
text.Location.AlignV = AlignV.Bottom
mypane.GraphObjList.Add(text)

```

```

Dim i As Integer
Dim x, y As Integer

If Form2.Visible = True Then
    If Val(Form2.ComboBox2.Text) = 0 Then
        MsgBox("Δεν έχει επιλεγεί κανένα scrambling code",
MsgBoxStyle.Exclamation, "3G Analysis")
    Else
        x = Form2.ComboBox2.Text
    End If
Else
    If Val(ComboBox1.Text) = 0 Then
        MsgBox("Δεν έχει επιλεγεί κανένα scrambling code",
MsgBoxStyle.Exclamation, "3G Analysis")
    Else
        x = ComboBox1.Text
    End If
End If

If Val(Form2.ComboBox2.Text) = 0 And Val(ComboBox1.Text) = 0 Then
Else
    For i = 0 To countsccrcodes
        If x = codes(i) Then
            y = i
            GoTo 1
        End If
    Next
1:
    Dim avegrscp As Double() = {avgtable.Rows(y - 1).Item(3) * (-1)}
    Dim avegecn0 As Double() = {avgtable.Rows(y - 1).Item(6) * (-1)}
    Dim maxrscp As Double() = {avgtable.Rows(y - 1).Item(1) * (-1)}
    Dim maxecno As Double() = {avgtable.Rows(y - 1).Item(4) * (-1)}
    Dim minrscp As Double() = {avgtable.Rows(y - 1).Item(2) * (-1)}
    Dim minecno As Double() = {avgtable.Rows(y - 1).Item(5) * (-1)}

    Dim myCurve1 As BarItem = mypane.AddBar("Rscp", Nothing, maxrscp,
Color.SkyBlue)
    myCurve1.Bar.Border.IsVisible = True
    Dim myCurve2 As BarItem = mypane.AddBar("", Nothing, minrscp,
Color.SkyBlue)
    myCurve2.Bar.Border.IsVisible = True
    Dim myCurve3 As BarItem = mypane.AddBar("Μέση Τιμή", Nothing,
avegrscp, Color.Red)
    myCurve3.Bar.Border.IsVisible = True
    Dim myCurve4 As BarItem = mypane.AddBar("Ec/No", Nothing,
maxecno, Color.LightGreen)
    myCurve4.Bar.Border.IsVisible = True
    Dim myCurve5 As BarItem = mypane.AddBar("", Nothing, minecno,
Color.LightGreen)
    myCurve5.Bar.Border.IsVisible = True
    Dim myCurve6 As BarItem = mypane.AddBar("", Nothing, avegecn0,
Color.Red)
    myCurve6.Bar.Border.IsVisible = True
    BarItem.CreateBarLabels(mypane, True, "f0")
    mypane.XAxis.MajorTic.IsBetweenLabels = True
    mypane.XAxis.Type = AxisType.Text
    mypane.Chart.Fill = New Fill(Color.LightPink, Color.SteelBlue,
45.0F)
    mypane.Fill = New Fill(Color.LightCyan, Color.FromArgb(255, 255,
225), 45)
    mypane.Legend.IsVisible = True

```

Εισάγουμε στην μεταβλητή x το scr. code που έχει επιλέξει ο χρήστης (αν υπάρχει).

Ψάχνουμε να βρούμε για ποια τιμή του i το  $x = \text{codes}(i)$  και όταν το βρούμε βγαίνουμε από τον βρόγχο, δηλώνουμε τις μεταβλητές που θέλουμε και εισάγουμε σε αυτές τις τιμές που βρίσκονται στον πίνακα avgtable.

```

    Dim g As Graphics = Me.CreateGraphics
    mypane.AxisChange(g)
    Me.ZedGraphControll1.AxisChange()
    g.Dispose()
End If
End Sub

```

### Drop Call Rate :

```

Sub CADrate()
    Dim cdr As Double
    Dim i, count, check, countall As Integer
    count = 0
    countall = 0
    check = 100
    For i = 0 To general2.Rows.Count - 1
        If general2.Rows(i).Item(9) <> 1 Then
            count = count + 1
        End If
    Next
    For i = 0 To mytable.Rows.Count - 1
        If check <> mytable.Rows(i).Item(0) And
mytable.Rows(i).Item(1) = "Active" Then
            countall = countall + 1
        End If
        check = mytable.Rows(i).Item(0)
    Next i
    Debug.Print(countall)
    cdr = (count / countall) * 100
    cdr = Math.Round(cdr, 3)

```

Ελέγχουμε το ένατο αντικείμενο κάθε γραμμής του πίνακα general2 και για κάθε διαφορετική τιμή από 1, αυξάνουμε κατά μία μονάδα την τιμή του count. Συνεπώς στο **count** έχουμε αποθηκεύσει το πλήθος των drop calls.

Στον επόμενο βρόγχο ελέγχουμε όλες τις γραμμές του πίνακα mytable και αποθηκεύουμε στο **countall** το πλήθος των Active servers.

Τέλος εισάγουμε στην μεταβλητή cdr το πηλίκο του count με το countall πολλαπλασιασμένο επί 100. Με απλά λόγια έχουμε βρει το ποσοστό των drop call στην διάρκεια της διαδρομής.

Με αυτή την εντολή περιορίζουμε τον δεκαδικό αριθμό cdr στα **τρία** δεκαδικά ψηφία.

```

MsgBox(" To Drop Call Rate είναι " & cdr & "%", MsgBoxStyle.
Information, "3G Analysys")
End Sub

```

Σε αυτό το σημείο τελειώσαμε με την ανάλυση κώδικα των γραφημάτων και θα συνεχίσουμε στο υπόλοιπο Form1.

```

Sub findcodesnofill()
    Dim i, cn, aa, temp As Integer
    countscrcodes = 1
    cn = 1
    temp = -1
    i = 0
    codes(countscrcodes) = mytable.Rows(i).Item(3)
    For i = 0 To mytable.Rows.Count - 1
        If temp <> mytable.Rows(i).Item(0) And mytable.Rows(i).Item(1) =
"Active" Then
            For cn = 0 To countscrcodes
                aa = mytable.Rows(i).Item(3)
                If mytable.Rows(i).Item(3) = Val(codes(cn)) Then
                    GoTo 1
                End If
            Next cn
            countscrcodes = countscrcodes + 1

```

Η συνάρτηση αυτή είναι ίδια με την συνάρτηση Findcodes με την διαφορά ότι δεν γεμίζει τα combobox.

```

        codes(countsrcodes) = mytable.Rows(i).Item(3)
    End If
1: temp = mytable.Rows(i).Item(0)
    Next i
End Sub

Sub showdetails()
    Dim i, j, temp As Integer
    DataGridView1.Visible = True
    DataGridView1.Rows.Clear()
    DataGridView1.Rows.Add(7)
    DataGridView1.Rows(0).Cells(0).Value = "Longitude"
    DataGridView1.Rows(1).Cells(0).Value = "Latitude"
    DataGridView1.Rows(2).Cells(0).Value = "Channel"
    DataGridView1.Rows(3).Cells(0).Value = "Scrambling Code"
    DataGridView1.Rows(4).Cells(0).Value = "RSCP"
    DataGridView1.Rows(5).Cells(0).Value = "ECNO"
    DataGridView1.Rows(6).Cells(0).Value = "Μέτρηση"
    temp = -1
    i = 0
    For i = 0 To mytable.Rows.Count - 1
        If temp <> mytable.Rows(i).Item(0) Then
            j = mytable.Rows(i).Item(0) - 1
            If mytable.Rows(i).Item(1) = "Active" Then
                If j = TextBox2.Text - 1 Then
                    DataGridView1.Rows(0).Cells(1).Value = general.
Rows(TextBox2.Text - 1).Item(1)
                    DataGridView1.Rows(1).Cells(1).Value = general.
Rows(TextBox2.Text - 1).Item(2)
                    DataGridView1.Rows(2).Cells(1).Value = mytable.
Rows(i).Item(2)
                    DataGridView1.Rows(3).Cells(1).Value = mytable.
Rows(i).Item(3)
                    DataGridView1.Rows(4).Cells(1).Value = mytable.
Rows(i).Item(6)
                    DataGridView1.Rows(5).Cells(1).Value = mytable.
Rows(i).Item(4)
                    DataGridView1.Rows(6).Cells(1).Value = mytable.
Rows(i).Item(0)
                End If
            End If
        End If
        temp = mytable.Rows(i).Item(0)
    Next i
End Sub

Private Sub OpenToolStripMenuItem1_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles OpenToolStripMenuItem1.Click
    OpenFileDialog1.ShowDialog()
End Sub

Private Sub SaveAsToolStripMenuItem1_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles SaveAsToolStripMenuItem1.Click
    SaveFileDialog1.ShowDialog()
End Sub

Private Sub PrintToolStripButton_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles PrintToolStripButton.Click
    PrintDialog1.ShowDialog()
End Sub

Private Sub ImportDatabaseToolStripMenuItem_Click(ByVal sender As System.Object, ByVal
e As System.EventArgs) Handles ImportDatabaseToolStripMenuItem.Click
    OpenFileDialog2.ShowDialog()
End Sub

```

Όταν καλούμε την συνάρτηση showdetails εμφανίζεται ένα πλαίσιο με τα βασικά χαρακτηριστικά του σημείου που έχει επιλέξει ο χρήστης.

```
Private Sub ExportDatabaseToolStripMenuItem_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles ExportDatabaseToolStripMenuItem.Click
    SaveFileDialog2.ShowDialog()
End Sub

Private Sub PrintPreviewToolStripMenuItem_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles PrintPreviewToolStripMenuItem.Click
    PrintPreviewDialog1.ShowDialog()
End Sub

Private Sub PrintToolStripMenuItem1_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles PrintToolStripMenuItem1.Click
    PrintDialog1.ShowDialog()
    Dim pd As PrintDocument = New PrintDocument
    Dim psd As New PageSetupDialog
    AddHandler pd.PrintPage, AddressOf Graph_PrintPage
    PrintDialog1.Document = pd
    PrintDialog1.Document.Print()
End Sub

Private Sub ExitToolStripMenuItem1_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles ExitToolStripMenuItem1.Click
    End
End Sub

Private Sub ΓραφήματαToolStripMenuItem_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles ΓραφήματαToolStripMenuItem.Click
    If Form2.TopMost = True Then
        Form2.TopMost = False
        ΓραφήματαToolStripMenuItem.Checked = False
    Else
        Form2.TopMost = True
        ΓραφήματαToolStripMenuItem.Checked = True
    End If
End Sub

Private Sub ΠίνακεςΜετρήσεωνToolStripMenuItem_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles ΠίνακεςΜετρήσεωνToolStripMenuItem.Click
    If Form3.TopMost = True Then
        Form3.TopMost = False
        ΠίνακεςΜετρήσεωνToolStripMenuItem.Checked = False
    Else
        Form3.TopMost = True
        ΠίνακεςΜετρήσεωνToolStripMenuItem.Checked = True
    End If
End Sub

Private Sub ΜελέτηΚυψελώνToolStripMenuItem1_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles ΜελέτηΚυψελώνToolStripMenuItem1.Click
    If Form4.TopMost = True Then
        Form4.TopMost = False
        ΜελέτηΚυψελώνToolStripMenuItem1.Checked = False
    Else
        Form4.TopMost = True
        ΜελέτηΚυψελώνToolStripMenuItem1.Checked = True
    End If
End Sub

Private Sub settingsToolStripMenuItem_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles settingsToolStripMenuItem.Click
    If Groupbox1.Visible = True And GroupBox2.Visible = True Then
        Groupbox1.Visible = False
        GroupBox2.Visible = False
        settingsToolStripMenuItem.Checked = False
    Else
        Groupbox1.Visible = True
        GroupBox2.Visible = True
        settingsToolStripMenuItem.Checked = True
    End If
End Sub

Private Sub AnalyzeToolStripMenuItem1_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles AnalyzeToolStripMenuItem1.Click
    analyse1()
End Sub
```

```
Private Sub RouteStatisticsToolStripMenuItem_Click(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles RouteStatisticsToolStripMenuItem.Click
    Form2.Show()
End Sub

Private Sub TableToolStripMenuItem_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles TableToolStripMenuItem.Click
    Form3.Show()
    Form3.DataGridView1.DataSource = binding
    binding.DataSource = mytable
End Sub

Private Sub Table2ToolStripMenuItem1_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Table2ToolStripMenuItem1.Click
    findcodesnofill()
    Avg()
    Form4.DataGridView1.DataSource = binding2
    binding2.DataSource = avgtable
    Form4.Show()
End Sub

Private Sub ΣχετικάToolStripMenuItem1_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles ΣχετικάToolStripMenuItem1.Click
    AboutBox1.Show()
End Sub

Private Sub OpenFileDialog1_FileOk(ByVal sender As System.Object, ByVal e As
System.ComponentModel.CancelEventArgs) Handles OpenFileDialog1.FileOk
    analyse1()
    ToolStripButton1.Enabled = True
    ToolStripButton10.Enabled = True
    ToolStripButton11.Enabled = True
    ToolStripButton12.Enabled = True
    ToolStripButton13.Enabled = True
    ToolStripButton14.Enabled = True
    ToolStripButton15.Enabled = True
    AnalyzeToolStripMenuItem1.Enabled = True
    RouteStatisticsToolStripMenuItem.Enabled = True
    TableToolStripMenuItem.Enabled = True
    Table2ToolStripMenuItem1.Enabled = True
    settingsToolStripMenuItem.Enabled = True
    settingsToolStripMenuItem.Enabled = True
    Button6.Enabled = False
End Sub

Private Sub ToolStripMenuItem6_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles ToolStripMenuItem6.Click
    Form2.Show()
End Sub

Private Sub SaveAsToolStripMenuItem_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles SaveAsToolStripMenuItem.Click
    SaveFileDialog1.ShowDialog()
End Sub

Private Sub SaveToolStripButton_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles SaveToolStripButton.Click
    SaveFileDialog1.ShowDialog()
End Sub

Private Sub Form1_Load(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles MyBase.Load
    If developing Then
        analyse1()
    End If
    ToolStripProgressBar1.Visible = False
    tlplatos = "Γεωγραφικό πλάτος"
    tlmikos = "Γεωγραφικό μήκος"
    tldiadromi = "Διαδρομή"
    tltmima = "Τμήμα Ηλεκτρονικής " & Chr(10) & " Παράρτημα Χανίων"
    tldiadromiecno = "Διαδρομή βάσει το ECNO"
    tldiadromiscrcode = "Διαδρομή για συγκεκριμένο Scrambling Code"
    tlrscproute = "Ισχύς σήματος (RSCP) κατά την διαδρομή"
    tlrscp = "RSCP"
    tlmetriseis = "Μετρήσεις"
    tlbest5 = "5 Καλύτεροι Servers"
    tlmeasurement = "Αριθμός Μέτρησης"
End Sub
```

```
Private Sub ToolStripButton2_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles ToolStripButton2.Click
    Form2.Show()
End Sub

Private Sub ToolStripMenuItem7_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles ToolStripMenuItem7.Click
    Form3.Show()
    Form3.DataGridView1.DataSource = binding
End Sub

Private Sub ToolStripButton3_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs)
    Form3.Show()
End Sub

Private Sub AnalyzeToolStripMenuItem_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles AnalyzeToolStripMenuItem.Click
    Dialog1.Show()
End Sub

Private Sub ToolStripMenuItem2_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles ToolStripMenuItem2.Click
    PrintPreviewDialog1.ShowDialog()
End Sub

Private Sub ΒοήθειαToolStripMenuItem_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs)
End Sub

Private Sub ΣχετικάToolStripMenuItem_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles ΣχετικάToolStripMenuItem.Click
    AboutBox1.Show()
End Sub

Private Sub ToolStripButton9_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles ToolStripButton9.Click
    OpenFileDialog1.ShowDialog()
    Form2.ComboBox1.Items.Clear()
    Form2.ComboBox1.Text = ""
    ComboBox1.Items.Clear()
    ComboBox1.Text = ""
    Form2.ComboBox2.Items.Clear()
    Form2.ComboBox2.Text = ""
End Sub

Private Sub ToolStripButton10_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles ToolStripButton10.Click
    SaveFileDialog1.ShowDialog()
End Sub

Private Sub ToolStripButton11_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles ToolStripButton11.Click
    PrintDialog1.ShowDialog()
End Sub

Private Sub ToolStripButton12_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles ToolStripButton12.Click
    Form2.Show()
End Sub

Private Sub ToolStripButton1_Click_3(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles ToolStripButton1.Click
    GroupBox2.Visible = True
    Groupbox1.Visible = True
End Sub

Private Sub ΑνλήΔιαδρομήToolStripMenuItem1_Click(ByVal sender As System.Object, ByVal
e As System.EventArgs) Handles ΑνλήΔιαδρομήToolStripMenuItem1.Click
    SimpleRoute()
End Sub

Private Sub ΒάσειΤουEcNoToolStripMenuItem_Click(ByVal sender As System.Object, ByVal
e As System.EventArgs) Handles ΒάσειΤουEcNoToolStripMenuItem.Click
    routecodecolourECN0()
End Sub
```



```

Private Sub ΙαΣυγκεκριμένοScramblingCodeToolStripMenuItem_Click(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles
ΙαΣυγκεκριμένοScramblingCodeToolStripMenuItem.Click
    Form2.Visible = False
    coderoute()
End Sub

Private Sub ΜεΧρωματικόΚώδικαToolStripMenuItem_Click(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles ΜεΧρωματικόΚώδικαToolStripMenuItem.Click
    routecodecolour()
End Sub

Private Sub ΠαρεμβολήToolStripMenuItem_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles ΠαρεμβολήToolStripMenuItem.Click
    routeinter()
End Sub

Private Sub CallDisconnectToolStripMenuItem1_Click(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles CallDisconnectToolStripMenuItem1.Click
    cadroute()
End Sub

Private Sub CallFailedToolStripMenuItem1_Click(ByVal sender As System.Object, ByVal e
As System.EventArgs) Handles CallFailedToolStripMenuItem1.Click
    cafroute()
End Sub

Private Sub ToolStripButton13_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles ToolStripButton13.Click
    GroupBox2.Visible = True
    Groupbox1.Visible = True
End Sub

Private Sub ΟςActiveServerToolStripMenuItem_Click(ByVal sender As System.Object, ByVal
e As System.EventArgs) Handles ΟςActiveServerToolStripMenuItem.Click
    RSCP()
End Sub

Private Sub ΟςΚαι2οςActiveServerToolStripMenuItem_Click(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles ΟςΚαι2οςActiveServerToolStripMenuItem.Click
    rscp2()
End Sub

Private Sub Καλύτεροι5ServersToolStripMenuItem_Click(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles Καλύτεροι5ServersToolStripMenuItem.Click
    test5best()
End Sub

Private Sub ToolStripButton15_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles ToolStripButton15.Click
    GroupBox2.Visible = True
    Groupbox1.Visible = True
End Sub

Private Sub DropCallRateToolStripMenuItem_Click(ByVal sender As System.Object, ByVal e
As System.EventArgs) Handles DropCallRateToolStripMenuItem.Click
    CADrate()
End Sub

Private Sub ΜελέτηΣημείωνΠρινToCADToolStripMenuItem_Click(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles
ΜελέτηΣημείωνΠρινToCADToolStripMenuItem.Click
    cadpoints()
End Sub
Private Sub ΜελέτηΚυψέληςToolStripMenuItem_Click(ByVal sender As System.Object, ByVal
e As System.EventArgs) Handles ΜελέτηΚυψέληςToolStripMenuItem.Click
    Form2.Visible = False
    Avg()
    avgbar()
End Sub

Private Sub ΠίνακεςΜετρήσεωνToolStripMenuItem1_Click(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles ΠίνακεςΜετρήσεωνToolStripMenuItem1.Click
    Form3.DataGridView1.DataSource = binding
    binding.DataSource = mytable
    Form3.Show()

```

```
End Sub
```

```
Private Sub ΜελέτηΚυψελώνToolStripMenuItem_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles ΜελέτηΚυψελώνToolStripMenuItem.Click
    findcodesnofill()
    Avg()
    Form4.DataGridView1.DataSource = binding2
    binding2.DataSource = avgtable
    Form4.Show()
End Sub
```

```
Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button1.Click
    Findcodes()
End Sub
```

```
Private Sub PrintPreviewDialog1_Load_1(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles PrintPreviewDialog1.Load
    Dim pd As PrintDocument = New PrintDocument
    AddHandler pd.PrintPage, AddressOf Graph_PrintPage
    PrintPreviewDialog1.Document = pd
    PrintPreviewDialog1.Show()
End Sub
```

```
Private Sub SaveFileDialog1_FileOk(ByVal sender As System.Object, ByVal e As System.ComponentModel.CancelEventArgs) Handles SaveFileDialog1.FileOk
    Dim mypane As ZedGraph.GraphPane = ZedGraphControl1.GraphPane
    Dim saveas As String
    saveas = SaveFileDialog1.FileName
    mypane.GetImage.Save(saveas, System.Drawing.Imaging.ImageFormat.Jpeg)
End Sub
```

```
Private Sub Button6_Click_1(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button6.Click
    Dim j As Integer
    j = Val(TextBox2.Text)
    If j > general.Rows.Count Then
        MsgBox("Δεν υπάρχει η μέτρηση που επιλέξατε", MsgBoxStyle.Information, "Info")
    End If
    showdetails()
End Sub
```

```
Private Sub SaveFileDialog2_FileOk(ByVal sender As System.Object, ByVal e As System.ComponentModel.CancelEventArgs) Handles SaveFileDialog2.FileOk
    Dim export As String
    export = SaveFileDialog2.FileName
    If export <> "" Then
        mytable.WriteXml(export, True)
    End If
End Sub
```

```
Private Sub OpenFileDialog2_FileOk_1(ByVal sender As System.Object, ByVal e As System.ComponentModel.CancelEventArgs) Handles OpenFileDialog2.FileOk
    Dim import As String
    import = OpenFileDialog2.FileName
    If import <> "" Then
        createtables()
        mytable.ReadXml(import)
    End If
End Sub
```

```
Private Sub TextBox2_KeyPress(ByVal sender As Object, ByVal e As System.Windows.Forms.KeyPressEventArgs) Handles TextBox2.KeyPress
    If Not IsNumeric(e.KeyChar) Then
        e.Handled = True
    End If
End Sub
```

```
Private Sub TextBox2_TextChanged(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles TextBox2.TextChanged
    If TextBox2.Text = "" Then
        Button6.Enabled = False
    Else
        If Convert.ToDecimal(TextBox2.Text) > mytable.Rows(mytable.Rows.Count - 1).Item(0) Then
            MsgBox("Οι μετρήσεις κυμαίνονται από 1 έως " & mytable.Rows(mytable.Rows.Count - 1).Item(0) & "", MsgBoxStyle.Exclamation, "Σφάλμα - Τιμή εκτός ορίων")
        End If
    End If
End Sub
```

```
        TextBox2.Text = mytable.Rows(mytable.Rows.Count - 1).Item(0)
    End If
    Button6.Enabled = True
End If
End Sub

Private Sub ZedGraphControl1_Load(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles ZedGraphControl1.Load
    ZedGraphControl1.Visible = True
    Dim oldPane As ZedGraph.GraphPane = ZedGraphControl1.GraphPane
    ZedGraphControl1.GraphPane = New ZedGraph.GraphPane(oldPane.Rect,
oldPane.Title.Text, oldPane.XAxis.Title.Text, oldPane.YAxis.Title.Text)
    Dim mypane As ZedGraph.GraphPane = New ZedGraph.GraphPane
    mypane = Me.ZedGraphControl1.GraphPane
    mypane.Title.Text = "Γράφημα"
    mypane.XAxis.Title.Text = ""
    mypane.YAxis.Title.Text = ""
    mypane.Chart.Fill = New Fill(Color.LightPink, Color.SteelBlue, 45.0F)
    mypane.Fill = New Fill(Color.LightCyan, Color.FromArgb(255, 255, 225), 45)
    Dim g As Graphics = Me.CreateGraphics
    mypane.AxisChange(g)
    Me.ZedGraphControl1.AxisChange()
    g.Dispose()
End Sub
Private Sub SettingsToolStripMenuItem1_Click_2(ByVal sender As System.Object, ByVal e
As System.EventArgs) Handles SettingsToolStripMenuItem1.Click
    usersettings.Show()

End Sub
End Class
```

## Ανάλυση κώδικα στο Form2

Το form2 είναι εκείνο το παράθυρο στο οποίο ο χρήστης επιλέγει την γραφική παράσταση την οποία θέλει να επεικονήσει. Οι εντολές που χρησιμοποιούνται είναι απλές εντολές της VB.Net οι οποίες ελέγχουν τα διάφορα checkboxes και groupboxes οι οποίες χρησιμοποιούνται κατάλληλα για να βγει το ποθητό αποτέλεσμα. Γι' αυτό τον λόγο δεν υπάρχει κάποια ιδιαίτερη σημασία να κάνουμε ανάλυση του κώδικα. Στην συνέχεια φαίνεται ολόκληρος ο κώδικας του form 2.

```
Public Class Form2

    Private Sub Button2_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
        Me.Hide()
    End Sub

    Private Sub CheckBox5_CheckedChanged(ByVal sender As System.Object, ByVal e As System.EventArgs)
        If CheckBox5.Checked = True Then
            GroupBox2.Visible = True
            CheckBox6.Checked = False
            CheckBox7.Checked = False
        Else
            GroupBox2.Visible = False
        End If
    End Sub

    Private Sub CheckBox6_CheckedChanged(ByVal sender As System.Object, ByVal e As System.EventArgs)
        If CheckBox6.Checked = True Then
            GroupBox2.Visible = False
            CheckBox5.Checked = False
            CheckBox7.Checked = False
        End If
    End Sub

    Private Sub CheckBox7_CheckedChanged(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles CheckBox7.CheckedChanged
        If CheckBox7.Checked = True Then
            CheckBox1.Checked = False
            CheckBox2.Checked = False
            CheckBox3.Checked = False
            CheckBox4.Checked = False
            CheckBox8.Checked = False
            CheckBox17.Checked = False
            CheckBox10.Visible = True
        Else
            CheckBox10.Visible = False
            CheckBox10.Checked = False
        End If
    End Sub

    Private Sub CheckBox3_CheckedChanged(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles CheckBox3.CheckedChanged
        If CheckBox3.Checked = True Then
            ComboBox1.Visible = True
            CheckBox1.Checked = False
            CheckBox2.Checked = False
            CheckBox4.Checked = False
        End If
    End Sub

End Class
```

```
        CheckBox7.Checked = False
        CheckBox8.Checked = False
        CheckBox17.Checked = False
        ComboBox1.Visible = True
        Form1.Findcodes()
    Else
        ComboBox1.Visible = False
    End If
End Sub

Private Sub CheckBox9_CheckedChanged(ByVal sender As System.Object,
ByVal e As System.EventArgs)
    Form1.Findcodes()
End Sub

Private Sub Button1_Click_1(ByVal sender As System.Object, ByVal e
As System.EventArgs) Handles Button1.Click
    If CheckBox1.Checked = True Then
        Form1.routecodecolour()
    ElseIf CheckBox2.Checked = True Then
        Form1.SimpleRoute()
    ElseIf CheckBox3.Checked = True Then
        Form1.coderoute()
    ElseIf CheckBox4.Checked = True Then
        Form1.cadroute()
    ElseIf CheckBox7.Checked = True Then
        Form1.routeinter()
    ElseIf CheckBox8.Checked = True Then
        Form1.cafroute()
    ElseIf CheckBox17.Checked = True Then
        Form1.routecodecolourECN0()
    End If
    Me.Close()
End Sub

Private Sub CheckBox1_CheckedChanged(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles CheckBox1.CheckedChanged
    If CheckBox1.Checked = True Then
        CheckBox2.Checked = False
        CheckBox3.Checked = False
        CheckBox4.Checked = False
        CheckBox7.Checked = False
        CheckBox8.Checked = False
        CheckBox17.Checked = False
    End If
End Sub

Private Sub CheckBox2_CheckedChanged_1(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles
CheckBox2.CheckedChanged
    If CheckBox2.Checked = True Then
        CheckBox1.Checked = False
        CheckBox3.Checked = False
        CheckBox4.Checked = False
        CheckBox7.Checked = False
        CheckBox8.Checked = False
        CheckBox17.Checked = False
    End If
End Sub
```

```
Private Sub CheckBox4_CheckedChanged_1(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles
CheckBox4.CheckedChanged
    If CheckBox4.Checked = True Then
        CheckBox1.Checked = False
        CheckBox2.Checked = False
        CheckBox3.Checked = False
        CheckBox7.Checked = False
        CheckBox8.Checked = False
        CheckBox17.Checked = False
    End If
End Sub

Private Sub CheckBox8_CheckedChanged(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles CheckBox8.CheckedChanged
    If CheckBox8.Checked = True Then
        CheckBox1.Checked = False
        CheckBox2.Checked = False
        CheckBox3.Checked = False
        CheckBox4.Checked = False
        CheckBox7.Checked = False
        CheckBox17.Checked = False
    End If
End Sub

Private Sub CheckBox10_CheckedChanged(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles
CheckBox10.CheckedChanged
    If CheckBox10.Checked = True Then
        Label1.Visible = True
        Label2.Visible = True
        TextBox1.Clear()
        TextBox2.Clear()
        TextBox1.Visible = True
        TextBox2.Visible = True
    Else
        Label1.Visible = False
        Label2.Visible = False
        TextBox1.Visible = False
        TextBox2.Visible = False
    End If
End Sub

Private Sub CheckBox5_CheckedChanged_1(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles
CheckBox5.CheckedChanged
    If CheckBox5.Checked = True Then
        CheckBox6.Checked = False
        CheckBox9.Checked = False
        CheckBox15.Checked = False
        CheckBox16.Checked = False
    End If
End Sub

Private Sub CheckBox6_CheckedChanged_1(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles
CheckBox6.CheckedChanged
    If CheckBox6.Checked = True Then
        CheckBox5.Checked = False
        CheckBox9.Checked = False
        CheckBox15.Checked = False
    End If
End Sub
```

```
        CheckBox16.Checked = False
    End If
End Sub

Private Sub CheckBox9_CheckedChanged_1(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles
CheckBox9.CheckedChanged
    If CheckBox9.Checked = True Then
        CheckBox5.Checked = False
        CheckBox6.Checked = False
        CheckBox15.Checked = False
        CheckBox16.Checked = False
    End If
End Sub

Private Sub Button2_Click_1(ByVal sender As System.Object, ByVal e
As System.EventArgs) Handles Button2.Click
    If CheckBox5.Checked = True Then
        Form1.RSCP()
    ElseIf CheckBox6.Checked = True Then
        Form1.rscp2()
    ElseIf CheckBox9.Checked = True Then
        Form1.test5best()
    ElseIf CheckBox15.Checked = True Then
        Form1.rscpscrcode()
    ElseIf CheckBox16.Checked = True Then
        Form1.rscpscrcode2()
    End If
    Me.Close()
End Sub

Private Sub CheckBox13_CheckedChanged_1(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles
CheckBox13.CheckedChanged
    If CheckBox13.Checked = True Then
        CheckBox12.Visible = True
        CheckBox12.Checked = False
        CheckBox11.Checked = False
        CheckBox14.Checked = False
        CheckBox18.Checked = False
    Else
        CheckBox12.Visible = False
        CheckBox12.Checked = False
    End If
End Sub

Private Sub CheckBox11_CheckedChanged_1(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles
CheckBox11.CheckedChanged
    If CheckBox11.Checked = True Then
        CheckBox13.Checked = False
        CheckBox14.Checked = False
        CheckBox18.Checked = False
    End If
End Sub

Private Sub CheckBox14_CheckedChanged(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles
CheckBox14.CheckedChanged
    If CheckBox14.Checked = True Then
        CheckBox13.Checked = False
    End If
End Sub
```

```
        CheckBox11.Checked = False
        CheckBox18.Checked = False
    End If
End Sub

Private Sub CheckBox12_CheckedChanged(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles
CheckBox12.CheckedChanged
    If CheckBox12.Checked = True Then
        Label3.Visible = True
        Label4.Visible = True
        TextBox3.Clear()
        TextBox4.Clear()
        TextBox3.Visible = True
        TextBox4.Visible = True
    Else
        Label3.Visible = False
        Label4.Visible = False
        TextBox3.Visible = False
        TextBox4.Visible = False
    End If
End Sub

Private Sub Button3_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button3.Click
    If CheckBox11.Checked = True Then
        Form1.CADrate()
    ElseIf CheckBox13.Checked = True Then
        Form1.rscpecn0inter()
    ElseIf CheckBox14.Checked = True Then
        Form1.Groupbox1.Visible = True
        Form1.cadpoints()
    ElseIf CheckBox18.Checked = True Then
        Form1.Avg()
        Form1.avgbar()
    End If
    Me.Close()
End Sub

Private Sub ToolTip1_Popup_1(ByVal sender As System.Object, ByVal e
As System.Windows.Forms.PopupEventArgs) Handles ToolTip1.Popup
    ToolTip1.ToolTipTitle = "qqqq"
End Sub

Private Sub CheckBox15_CheckedChanged(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles
CheckBox15.CheckedChanged
    If CheckBox15.Checked = True Then
        CheckBox5.Checked = False
        CheckBox6.Checked = False
        CheckBox9.Checked = False
        CheckBox16.Checked = False
    End If
End Sub

Private Sub CheckBox16_CheckedChanged(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles
CheckBox16.CheckedChanged
    If CheckBox16.Checked = True Then
        CheckBox5.Checked = False
        CheckBox6.Checked = False
    End If
End Sub
```



```
        CheckBox9.Checked = False
        CheckBox15.Checked = False
    End If
End Sub

Private Sub CheckBox17_CheckedChanged(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles
CheckBox17.CheckedChanged
    If CheckBox17.Checked = True Then
        CheckBox1.Checked = False
        CheckBox2.Checked = False
        CheckBox3.Checked = False
        CheckBox4.Checked = False
        CheckBox7.Checked = False
        CheckBox8.Checked = False
    End If
End Sub

Private Sub CheckBox18_CheckedChanged(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles
CheckBox18.CheckedChanged
    If CheckBox18.Checked = True Then
        CheckBox13.Checked = False
        CheckBox14.Checked = False
        CheckBox11.Checked = False
        ComboBox2.Visible = True
        Form1.Findcodes()
    Else
        ComboBox2.Visible = False
    End If
End Sub
End Class
```

## Ανάλυση κώδικα στο Form3

Το παράθυρο form3 μας παρουσιάζει τους πίνακες με τα στοιχεία που έχουμε δημιουργήσει στο form1. Στην ουσία είναι δεδομένα που εμείς φιλτράραμε από το αρχείο μετρήσεων (dt2).

```
Public Class Form3
```

```
Private Sub ComboBox1_SelectedIndexChanged(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles
```

```
ComboBox1.SelectedIndexChanged
```

```
binding.RemoveFilter()
```

```
Select Case ComboBox1.SelectedIndex
```

```
Case 0
```

```
binding.DataSource = general
```

```
TextBox1.Visible = True
```

```
Case 1
```

```
binding.DataSource = mytable
```

```
TextBox1.Visible = True
```

```
Case 2
```

```
binding.DataSource = mytable
```

```
binding.Filter = "Status LIKE 'Active'"
```

```
TextBox1.Visible = False
```

```
Case 3
```

```
binding.DataSource = mytable
```

```
binding.Filter = "Status LIKE 'Monitored'"
```

```
TextBox1.Visible = False
```

```
Case 4
```

```
binding.DataSource = mytable
```

```
binding.Filter = "Status LIKE 'Detected'"
```

```
TextBox1.Visible = False
```

```
Case 5
```

```
binding.DataSource = mytable
```

```
binding.Filter = "Status LIKE 'Undetected'"
```

```
TextBox1.Visible = False
```

```
Case 6
```

```
binding.DataSource = general2
```

```
TextBox1.Visible = False
```

```
Case 7
```

```
binding.DataSource = generalcaf
```

```
TextBox1.Visible = False
```

```
Case 8
```

```
binding.DataSource = best
```

```
TextBox1.Visible = False
```

```
End Select
```

```
End Sub
```

```
Private Sub Form3_Load(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles MyBase.Load
```

```
ComboBox1.SelectedIndex = 1
```

Μόλις ανοίγει το form3 εμφανίζεται ο πίνακας mytable. Αλλάζοντας τον αριθμο 1 θα μπορούσαμε να ορίσουμε κάποιον άλλο πίνακα.

```
End Sub
```

Χρησιμοποιώντας την ρουτίνα case, σε αυτό το κομμάτι του κώδικα, μπορεί ο χρήστης να επιλέγει τον πίνακα που επιθυμεί να δει.

Με την εντολη binding.Datasource επιλέγουμε τον πίνακα. Στην περίπτωση όμως που ένας πίνακας έχει διαφορετικές γραμμές τις οποίες θέλουμε να τις δούμε (όπως είναι ο πίνακας mytable) ομαδοποιημένες τότε προσθέτουμε και την γραμμή binding.Filter ώστε να φιλτράρουμε τον πίνακα και να εμφανίσουμε μόνο τις επιθυμητές γραμμές.

```

Private Sub TextBox1_KeyPress(ByVal sender As Object, ByVal e As
System.Windows.Forms.KeyPressEventArgs) Handles TextBox1.KeyPress
    If Not IsNumeric(e.KeyChar) Then
        e.Handled = True
    End If

```

Ελέγχουμε αν το πλήκτρο που θα πατήσει ο χρήστης θα είναι αριθμός. Αν δεν είναι τότε δεν εμφανίζεται τίποτα στο textbox.

```

End Sub

```

```

Private Sub TextBox1_TextChanged(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles TextBox1.TextChanged
    Dim j, i As Integer
    j = Val(TextBox1.Text)
    If j > general.Rows.Count Then
        MsgBox("Δεν υπάρχει η μέτρηση που επιλέξατε",
MsgBoxStyle.Information, "Info")
    Else
        If ComboBox1.Text = "General Info" Then
            For i = 0 To general.Rows.Count - 1
                If j = general.Rows(i).Item(0) Then
                    DataGridView1.CurrentCell = DataGridView1.Rows(j -
1).Cells(0)
                End If
            Next i
        ElseIf ComboBox1.Text = "All Servers" Then
            For i = 0 To mytable.Rows.Count - 1
                If j = mytable.Rows(i).Item(0) Then
                    DataGridView1.CurrentCell =
DataGridView1.Rows(i).Cells(0)
                End If
            Next i
        End If
    End Sub
End Class

```

Σε αυτό το σημείο έχουμε τοποθετήσει ένα textbox που σκοπό έχει να βρίσκει την μέτρηση που επιθυμεί ο χρήστης εισάγοντας απλά την τιμή της. Ο κώδικας αυτός ελέγχει αν η τιμή υπάρχει και στην συνέχεια μεταφέρεται ο πίνακας στην ανάλογη γραμμή. Αν η τιμή δεν υπάρχει τότε εμφανίζεται μήνυμα λάθους. Να τονίσουμε ότι αυτή η λειτουργία είναι ενεργή στους πίνακες General Info και mytable.

## Ανάλυση κώδικα στο Usersettings

```
Public Class usersettings
```

```
    Private Sub Button1_Click(ByVal sender As System.Object, ByVal e
As System.EventArgs) Handles Button1.Click
        tlplatos = TextBox2.Text
        tlmikos = TextBox1.Text
        tldiadromi = TextBox3.Text
        tltmima = TextBox4.Text
        tldiadromiecnocno = TextBox5.Text
        tldiadromiscrcode = TextBox6.Text
        tlrscproute = TextBox7.Text
        tlrscp = TextBox8.Text
        tlmetriseis = TextBox9.Text
        tlbest5 = TextBox10.Text
        tlmeasurement = TextBox11.Text
        Me.Hide()
        MsgBox(" Οι αλλαγές ολοκληρώθηκαν επιτυχώς.Επιλέξτε εκ νέου
την εμφάνιση του γραφήματος που σας ενδιαφέρει.",
MsgBoxStyle.Information, "3G Analysys")
```

```
    End Sub
```

```
    Private Sub TextBox4_TextChanged(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles TextBox4.TextChanged
```

```
    End Sub
```

```
    Private Sub Button2_Click(ByVal sender As System.Object, ByVal e
As System.EventArgs) Handles Button2.Click
        TextBox2.Text = "Γεωγραφικό πλάτος"
        TextBox1.Text = "Γεωγραφικό μήκος"
        TextBox3.Text = "Διαδρομή"
        TextBox4.Text = "Τμήμα Ηλεκτρονικής " & Chr(10) & " Παράρτημα
Χανίων"
        TextBox5.Text = "Διαδρομή βάσει το ECNO"
        TextBox6.Text = "Διαδρομή για συγκεκριμένο Scrambling Code"
        TextBox7.Text = "Ισχύς σήματος (RSCP) κατά την διαδρομή"
        TextBox8.Text = "RSCP"
        TextBox9.Text = "Μετρήσεις"
        TextBox10.Text = "5 Καλύτεροι Servers"
        TextBox11.Text = "Αριθμός Μέτρησης"
```

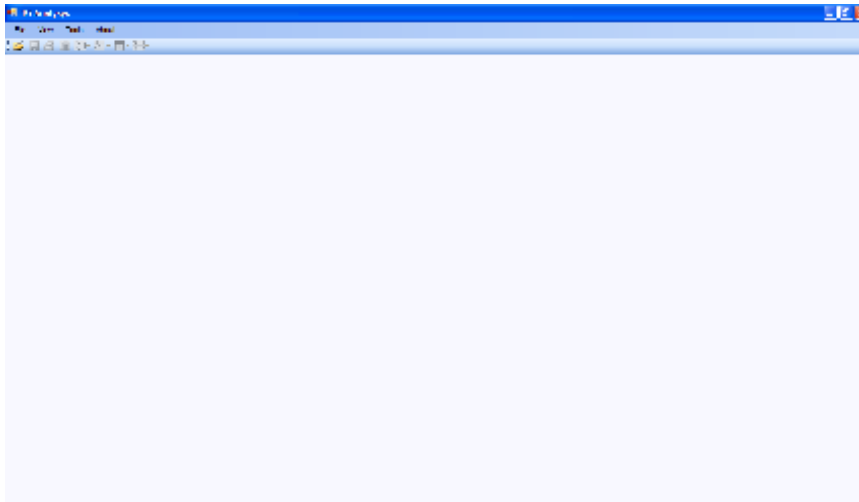
```
    End Sub
```

```
End Class
```

## Λειτουργία του προγράμματος

Σε αυτό το σημείο θα παρουσιάσουμε τις βασικές λειτουργίες του προγράμματος βήμα προς βήμα.

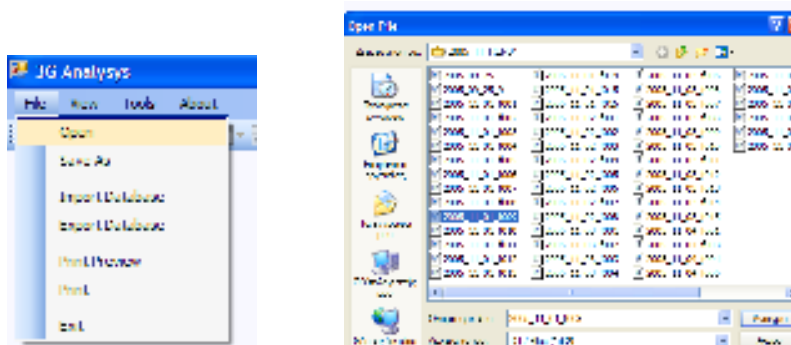
Ανοίγοντας για πρώτη φορά το πρόγραμμα το βλέπουμε στην παρακάτω μορφή.



σχ. 3.1

Για ν' ανοίξουμε το αρχείο μετρήσεων dt2 πηγαίνουμε File-> Open και προκύπτει το παρακάτω παράθυρο από το οποίο διαλέγουμε το ανάλογο αρχείο.

( Το πρόγραμμα υποστηρίζει και το ταυτόχρονο άνοιγμα πολλαπλών αρχείων. )

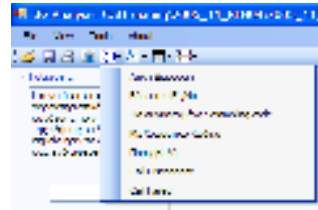


σχ. 3.2

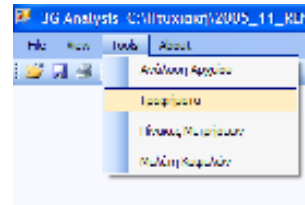
Στο παράδειγμα μας ανοίγουμε το αρχείο **2005\_11\_01\_f009.dt2**.

Αφού λοιπόν ανοίξουμε το αρχείο μας, επιστρέφουμε στο κεντρικό παράθυρο του προγράμματος. Σε αυτό το σημείο παρατηρούμε ότι διάφορα κουμπιά τα οποία ήταν ανενεργά πριν το άνοιγμα, τώρα έχουν ενεργοποιηθεί.

Από αυτό το σημείο και μετά είναι πολύ εύκολο να δημιουργήσουμε τα γραφήματα που μας ενδιαφέρουν. Να τονίσουμε ότι αυτό μπορεί να γίνει με 2 διαφορετικούς τρόπους. Είτε από τις συντομεύσεις του προγράμματος, είτε από το μενού.



Συντομεύσεις

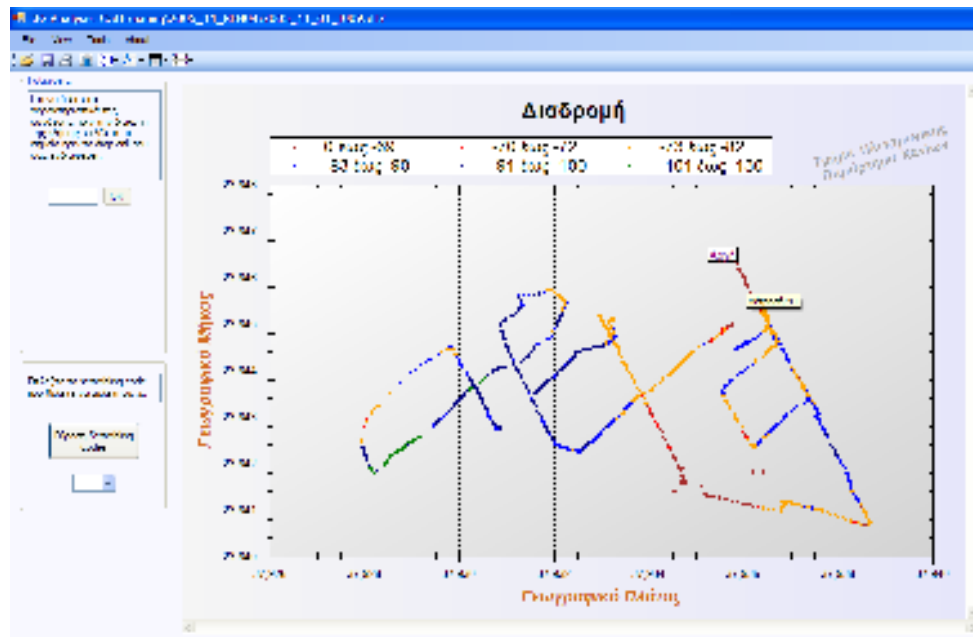


Μενού

Σχ 3.3

### Διαδρομή με χρωματικό κώδικα

Επιλέγουμε λοιπόν να εμφανίσουμε το γράφημα με την διαδρομή που διένυσε το όχημα την περίοδο των μετρήσεων και την ίδια στιγμή έχουμε ομαδοποιήσει τα σημεία σύμφωνα με τον χρωματικό κώδικα ο οποίος φαίνεται πάνω από το γράφημα μας.



σχ 3.4

Βλέπουμε επίσης ότι υπάρχει ένδειξη για το πρώτο σημείο των μετρήσεων καθώς και ένδειξη για το κάθε σημείο που βρίσκεται το ποντίκι. Κάτι τέτοιο γίνεται για να μπορεί ο χρήστης να έχει όσο το δυνατόν γίνεται καλύτερη εικόνα της διαδρομής και να μην μπερδεύεται από γραμμές (της διαδρομής δηλαδή) που περνάνε από το ίδιο σημείο.

Μπορούμε λοιπόν εύκολα με αυτό το γράφημα να δούμε τα σημεία (περιοχές) τα οποία δεν έχουν καλή κάλυψη του δικτύου που μελετάμε. Αν είχαμε στην διάθεση μας τους χάρτες των περιοχών που μελετάμε σε ψηφιακή μορφή θα μπορούσαμε να τοποθετήσουμε αυτόματα στο background την συγκεκριμένη περιοχή. Αυτό θα μας έδινε ακόμα καλύτερη προσέγγιση στο πρόβλημα που προσπαθούμε να λύσουμε. Κάτι τέτοιο όμως δεν ήταν δυστυχώς δυνατόν.

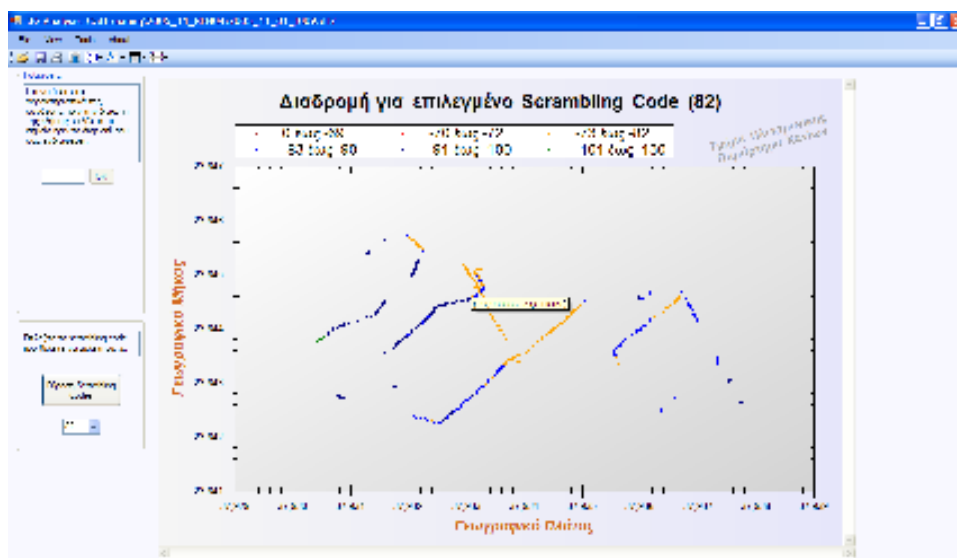
### Διαδρομή για συγκεκριμένο scrambling code

Από το σχ.... Βλέπουμε ότι υπάρχει ένα γράφημα που λέγεται *διαδρομή για συγκεκριμένο scrambling code*. Για να υπάρχει δυνατότητα απεικόνισης αυτού του γραφήματος θα πρέπει πρώτα να βρούμε όλα τα scrambling codes της διαδρομής. Σε αντίθετη περίπτωση θα εμφανιστεί ένα μήνυμα λάθους. Μόλις λοιπόν πατήσουμε το κουμπί **Εύρεση των scrambling codes**, το οποίο βρίσκεται στο αριστερό - κάτω μέρος της οθόνης θα εμφανιστεί το παρακάτω μήνυμα.



σχ. 3.5

Στην συνέχεια πατάμε OK, επιλέγουμε ένα scrambling code από το combobox που βρίσκεται στο ίδιο σημείο και πατάμε από τις συντομεύσεις το κουμπί που λέει *διαδρομή για συγκεκριμένο scrambling code*. Το αποτέλεσμα φαίνεται στην παρακάτω εικόνα.

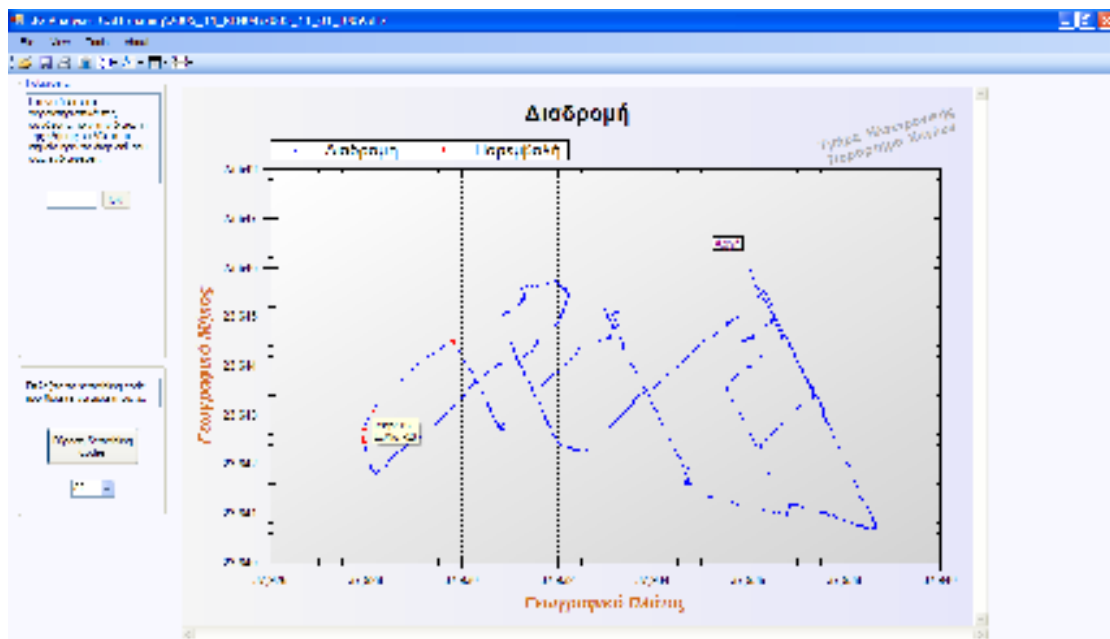


σχ. 3.6

## Παρεμβολή

Άλλη μια ενδιαφέρουσα γραφική παράσταση είναι αυτή η οποία μας δείχνει τα σημεία παρεμβολής. Έχουμε ορίσει τα σημεία που έχουμε παρεμβολή ως τα σημεία που έχουμε καλή λαμβανόμενη ισχύ και μεγάλο EC/NO (πιο συγκεκριμένα, είναι τα σημεία με  $rscp > -86$  και  $ecno < -11$ ). Δηλαδή θεωρούμε ότι δεν γίνεται σ' ένα σημείο να έχουμε καλό σήμα και ταυτόχρονα μεγάλη διαφορά του σήματος από το φέρον.

Το πρόγραμμα μας επιτρέπει την αλλαγή των προκαθορισμένων αυτών τιμών. (Tools – Γραφήματα – Παρεμβολή – Ρύθμιση RSCP και ECNO). Το γράφημα που προκύπτει φαίνεται παρακάτω. Τα σημεία παρεμβολής φαίνονται με κόκκινο χρώμα. Παρατηρείστε ότι αν μετακινήσετε το ποντίκι πάνω στο σημείο παρεμβολής μπορείτε να δείτε τα στοιχεία που σας ενδιαφέρουν.



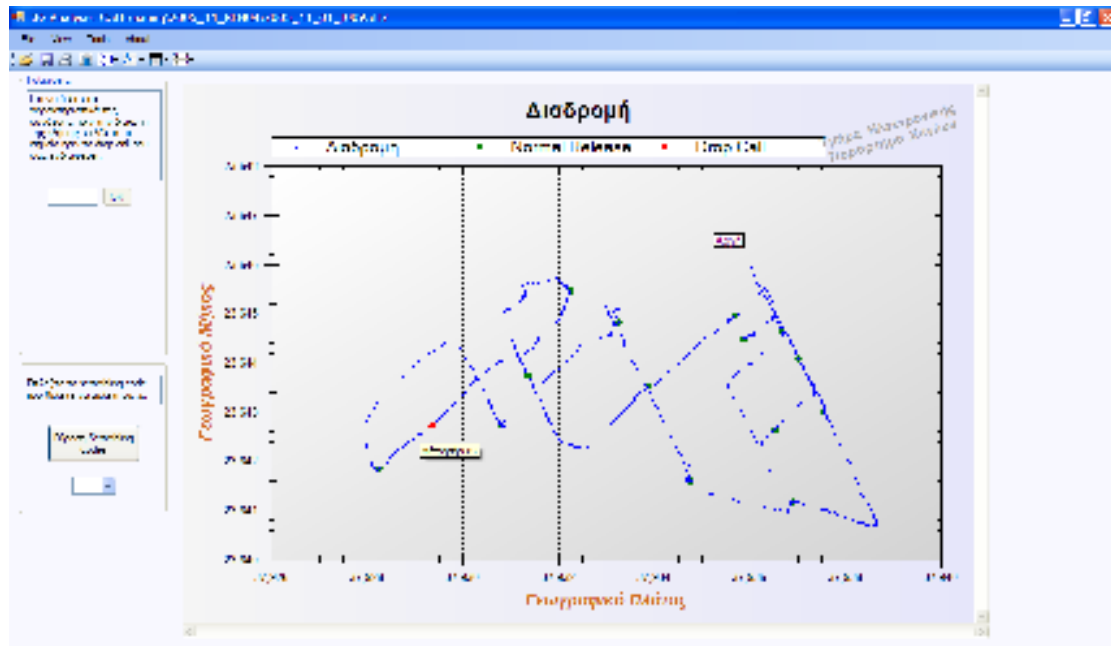
σχ. 3.7

## Call Disconnect και Call Failed ( CAD & CAF )

Τα 2 αυτά γραφήματα μας δείχνουν τα σημεία στα οποία είτε έγινε διακοπή της κλήσης (*CAD*), είτε έγινε αποτυχία πραγματοποίησης της κλήσης (*CAF*).



CAD

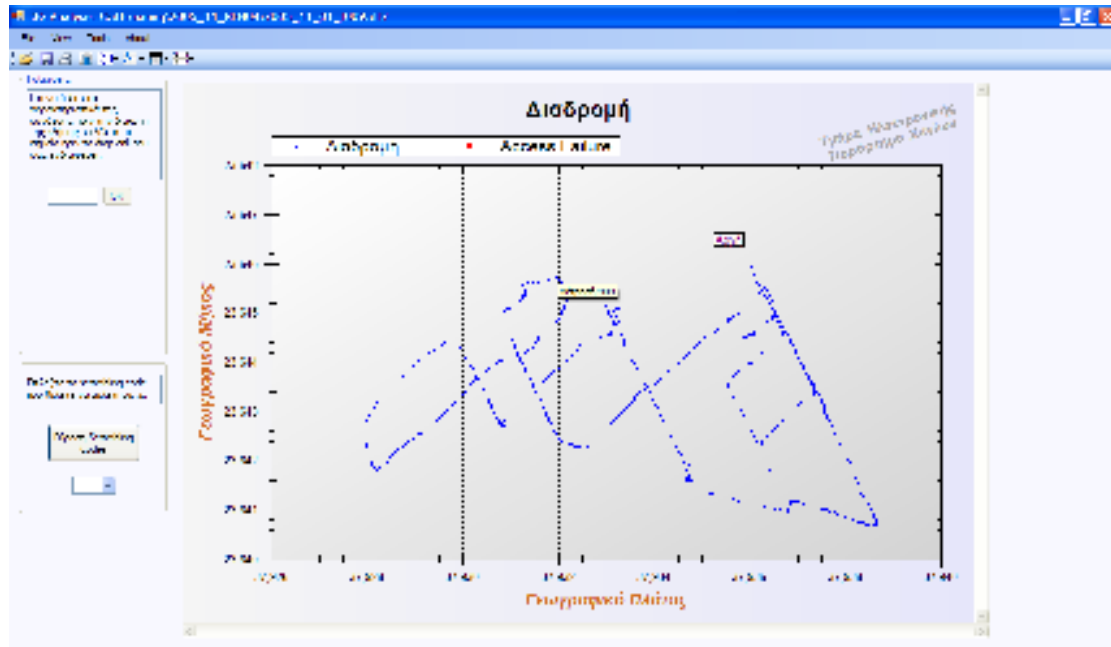


σχ. 3.8

Με **πράσινο** χρώμα φαίνονται τα σημεία που έγινε κανονική διακοπή της κλήσης.

Με **κόκκινο** χρώμα φαίνονται τα σημεία που έπεσε απότομα η γραμμή.

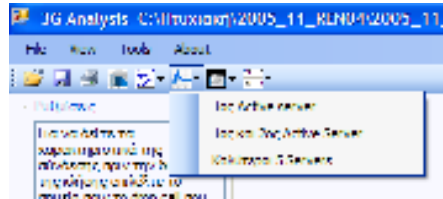
CAF



σχ. 3.9

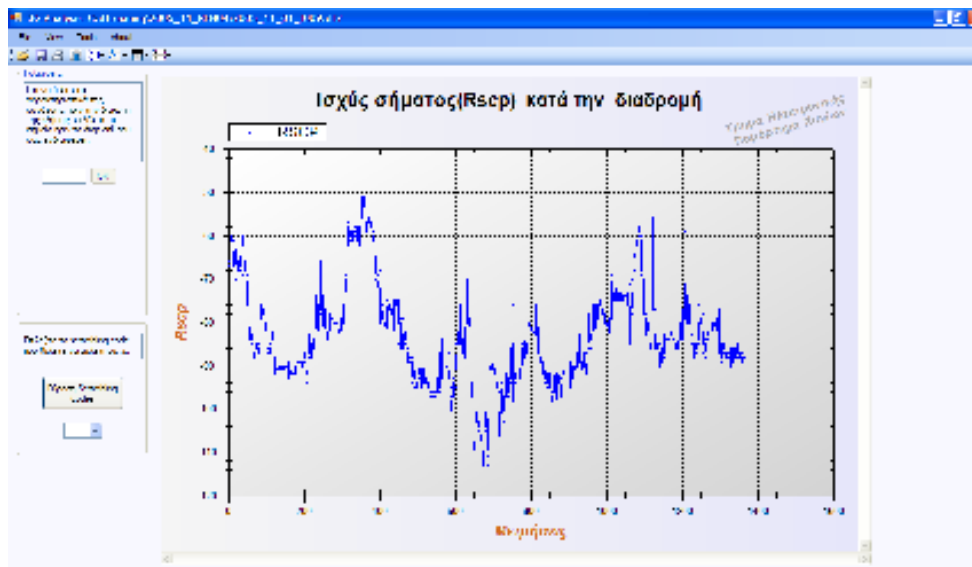
Στην συγκεκριμένη περίπτωση δεν είχαμε κάποιο τέτοιο σημείο στο οποίο να απέτυχε κάποια κλήση μας.

Στην συνέχεια θα δούμε γραφήματα που μας δείχνουν την μεταβολή της ισχύος ( είτε του 1<sup>ου</sup> Active Server, είτε του 1<sup>ου</sup> και 2<sup>ου</sup> Active Server, είτε των 5 καλύτερων Servers) στην διάρκεια της διαδρομής. Αυτό το πετυχαίνουμε κάνοντας κλικ στις συντομεύσεις που φαίνονται παρακάτω.



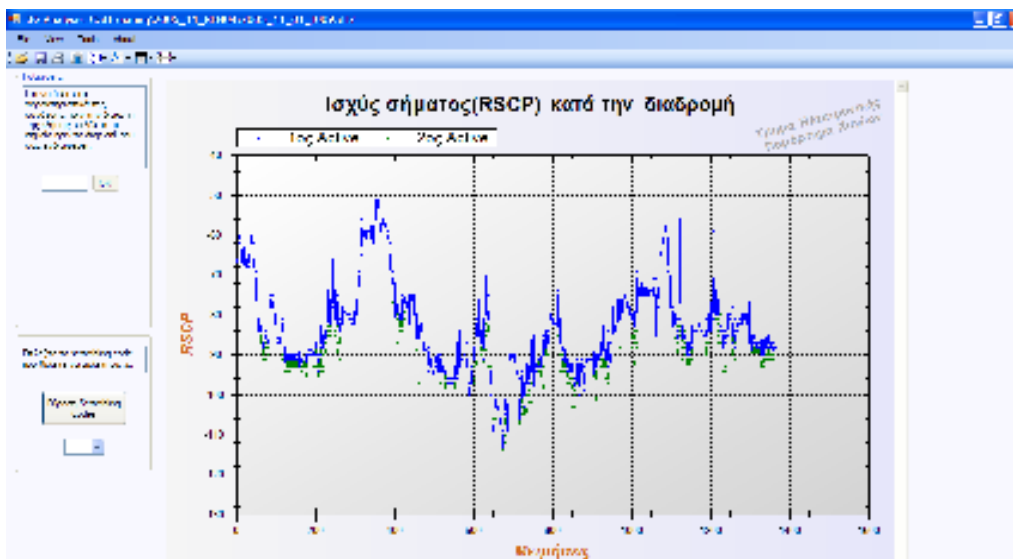
σχ. 3.10

**1<sup>ος</sup> Active Server**



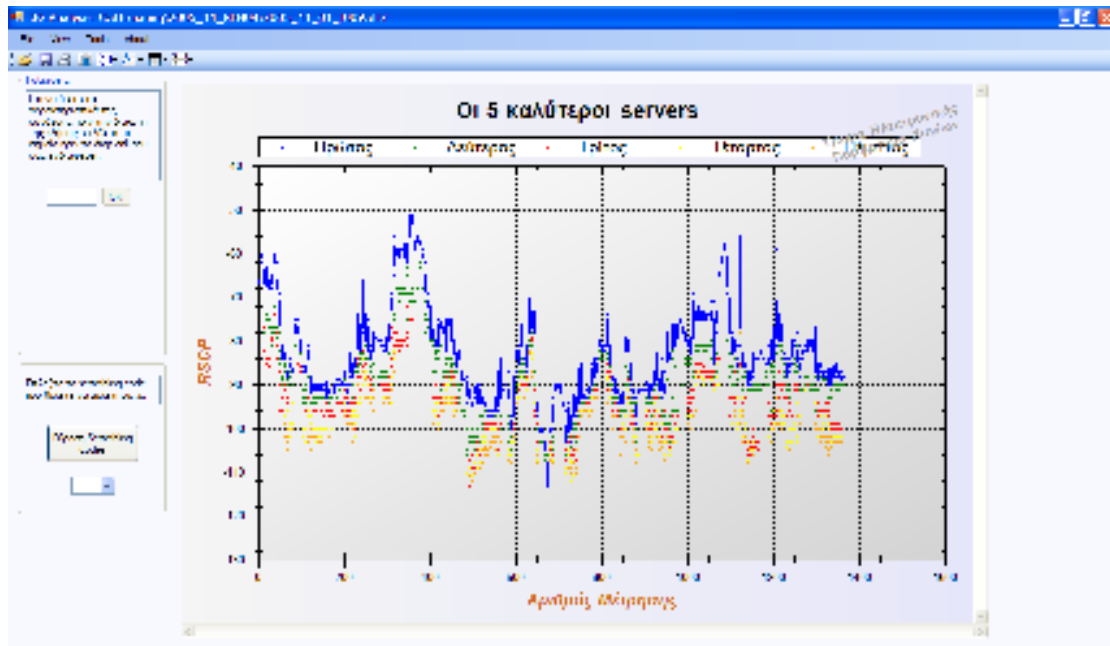
σχ. 3.11

**1<sup>ος</sup> και 2<sup>ος</sup> Active Server**



σχ. 3.12

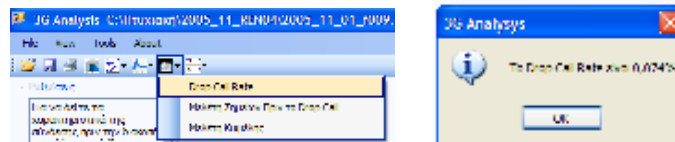
### 5 καλύτεροι Servers



σχ. 3.13

### Άλλα γραφήματα και λειτουργίες.

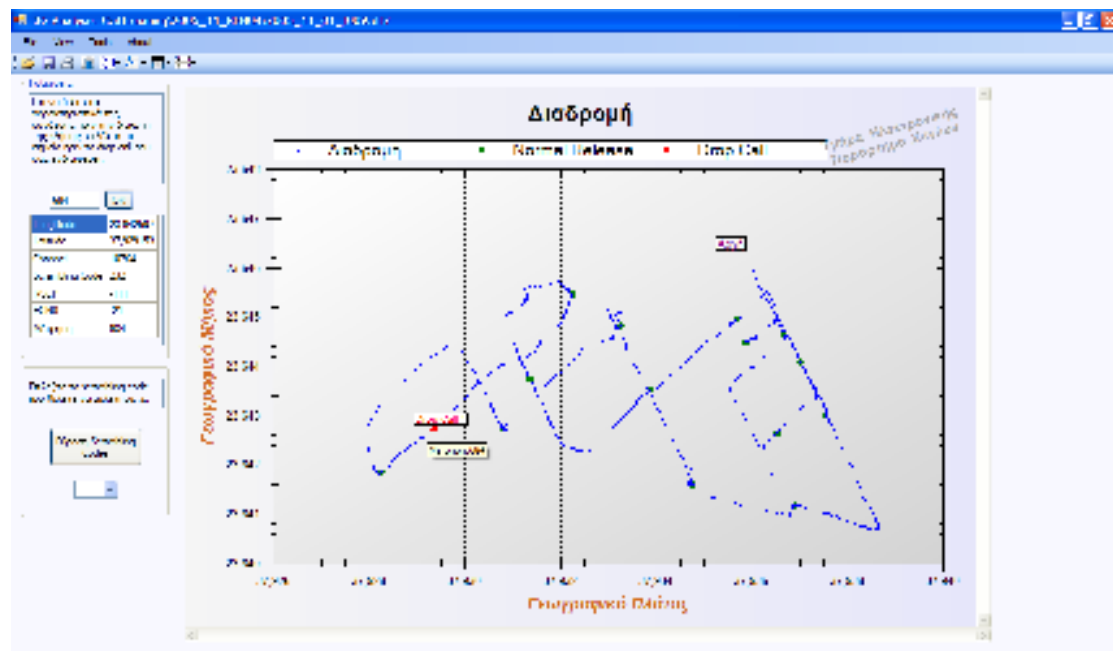
Το *drop call rate* είναι μια τιμή που μας δείχνει τον *λόγο* του αριθμού των σημείων των παρεμβολών προς τον αριθμό όλων των σημείων της διαδρομής (Πάντα μιλάμε για τους Active servers).



σχ. 3.14

Μία επίσης πολύ ενδιαφέρουσα λειτουργία είναι η *μελέτη σημείων πριν το drop call*. Το γράφημα αυτό εμφανίζεται κάνοντας κλικ στην συντόμευση **Άλλα** και στην συνέχεια στο κουμπί *μελέτη σημείων πριν το drop call*. Σε αυτό το γράφημα μας δίνεται η δυνατότητα να δούμε μόνο τα σημεία που συνέβη drop call και ταυτόχρονα να μελετήσουμε τα χαρακτηριστικά των σημείων που βρίσκονται πριν από αυτό. Για να το κάνουμε αυτό βλέπουμε ποιο σημείο (αριθμός) βρίσκεται πριν το drop call και το εισάγουμε στο πεδίο κειμένου που βρίσκεται στα αριστερά της οθόνης μας.

Για να το καταλάβουμε καλύτερα ας κοιτάξουμε το γράφημα που ακολουθεί.



σχ. 3.15

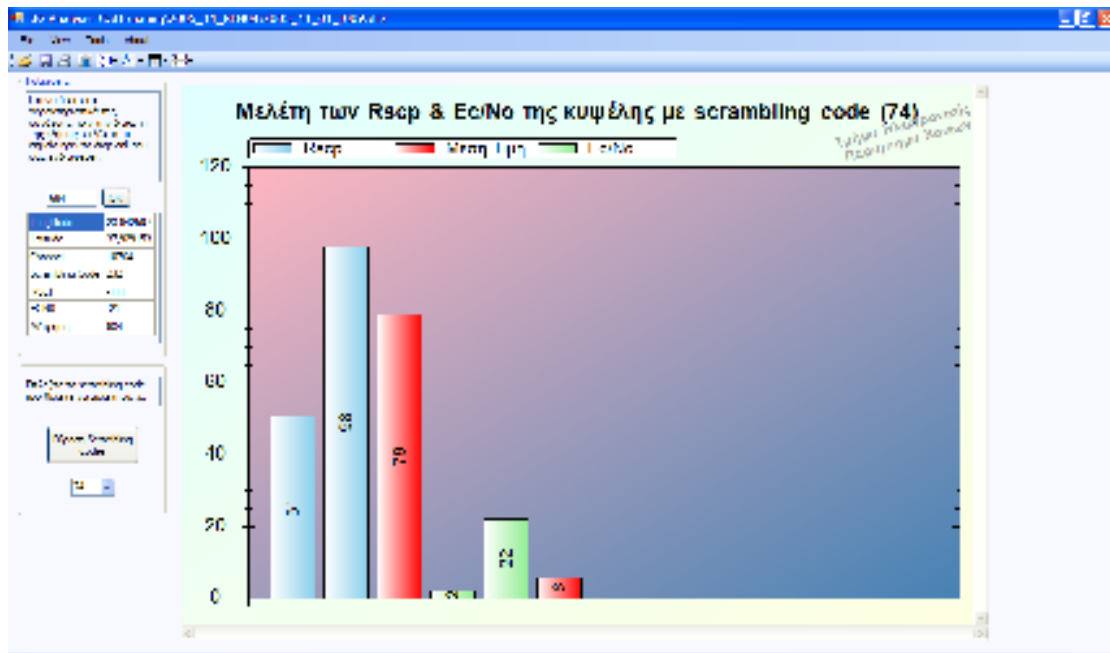
- Βλέπουμε το drop call (Μέτρηση 684)
- Τοποθετούμε το ποντίκι μας πριν από το drop call και βλέπουμε ότι είναι το σημείο 684.
- Εισάγουμε το 684 στο αριστερό πλαίσιο και πατάμε OK.
- Στην συνέχεια εμφανίζεται ο πίνακας με τα βασικά στοιχεία του σημείου.

(Να παρατηρήσουμε ότι το πρόγραμμα δεν επιτρέπει στον χρήστη να εισάγει κάποιο άλλο σύμβολο εκτός από αριθμό καθώς και μέτρηση η οποία δεν υπάρχει.)

### Μελέτη RSCP και ECNO

Το γράφημα αυτό εμφανίζεται κάνοντας κλικ στην συντόμευση **Άλλα** και στην συνέχεια στο κουμπί **Μελέτη Κυψέλης**. Το χρησιμοποιούμε για να δούμε την μέγιστη, ελάχιστη και μέση τιμή του rscp και του ecno μιας συγκεκριμένης κυψέλης που έχουμε επιλέξει εμείς.

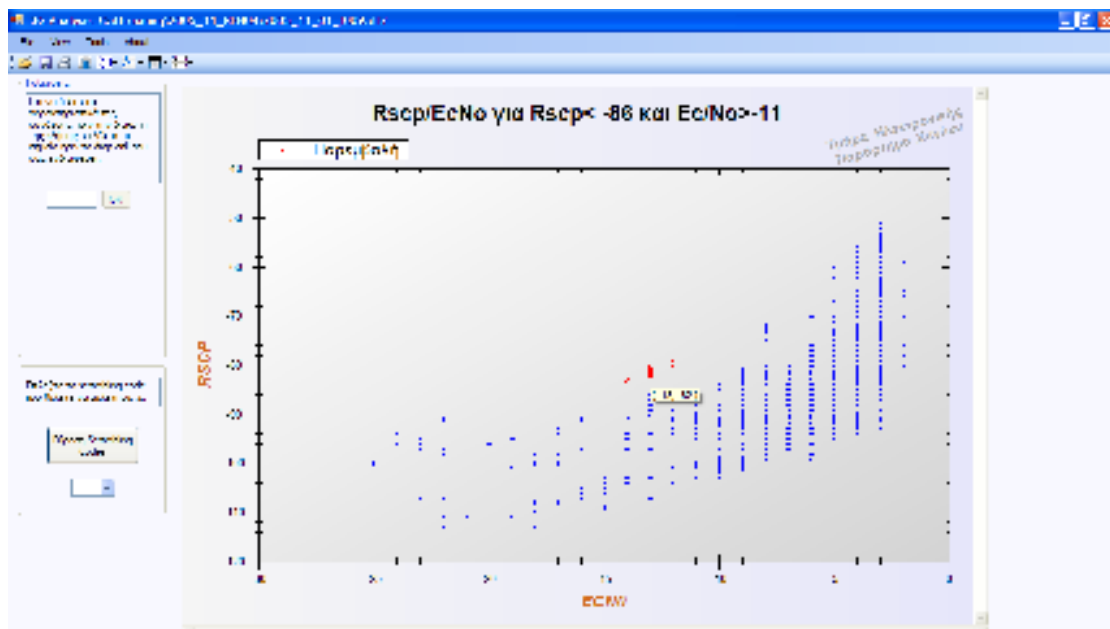
Όπως καταλαβαίνετε από το παρακάτω σχήμα με **μπλε** χρώμα φαίνονται οι τιμές του rscp, με **πράσινο** του ecno και με **κόκκινο** οι αντίστοιχες μέσες τιμές τους. Να τονίσουμε ότι όλες αυτές οι τιμές είναι **αρνητικές**. Απλά για λόγους καλύτερης κατανόησης του γραφήματος, έχουμε αφαιρέσει το - ( πλην ) .



σχ. 3.16

Ένα άλλο γράφημα το οποίο μας δείχνει τα ίδια στοιχεία αλλά με διαφορετικό τρόπο είναι αυτό της παρεμβολής.

Θα το βρείτε στο Tools – Γραφήματα – Άλλα – Παρεμβολή και το αποτέλεσμα του δίνεται παρακάτω.

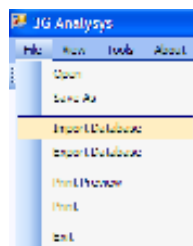


σχ. 3.17

## Βάσεις δεδομένων

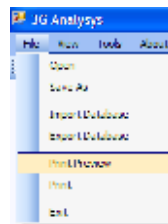
Μία άλλη δυνατότητα που μας δίνει το πρόγραμμα είναι η εισαγωγή και η εξαγωγή των στοιχείων σε βάση δεδομένων. Κάτι τέτοιο είναι χρήσιμο στην περίπτωση που ο χρήστης θέλει να δώσει τα στοιχεία ενός αρχείου μετρήσεων dt2 σε κάποιον άλλον χωρίς όμως να του δώσει το αρχείο dt2 (Τα αρχεία αυτά περιέχουν χιλιάδες ακόμα δεδομένα που ενδεχομένως να μην θέλουμε να μοιραστούμε). Έτσι λοιπόν θα πρέπει να υπάρχει τρόπος εισαγωγής και εξαγωγής αυτών των δεδομένων. Για να κάνουμε **εξαγωγή** σ' ένα xml αρχείο ανοίγουμε το αρχείο dt2 και στην συνέχεια από το File κάνουμε export database.

Για να κάνουμε εισαγωγή απλά πάμε File και import database.



σχ. 3.18

Τέλος ν' αναφέρουμε ότι υπάρχει και η δυνατότητα εκτύπωσης των γραφημάτων από το menu File



σχ. 3.19

Όλα τα γραφήματα που έχουμε κατασκευάσει μέχρι τώρα προέκυψαν από δεδομένα που έχουμε ταξινομήσει σε πίνακες. Οι πίνακες αυτοί μπορούν να εμφανιστούν είτε από τις συντομεύσεις, είτε από το menu tools – Πίνακες Μετρήσεων.

 A screenshot of the 'JG Analysis' application window showing a data table. The table has columns for 'Αντικείμενο', 'Όνομα', 'Εύρος', 'Τύπος', and 'Εύρος'. The data is organized into a grid with alternating light and dark rows. The table is titled 'Πίνακες Μετρήσεων' and contains multiple rows of data points.

σχ. 3.20

## **Επίλογος**

Μέσα από την ενασχόληση μας με την παρούσα εργασία είχαμε την ευκαιρία να γνωρίσουμε από μια διαφορετική οπτική γωνία τα κυψελωτά συστήματα τρίτης γενιάς, μια τεχνολογία η οποία ωριμάζει σιγά σιγά στην χώρα μας,. Μπορέσαμε επίσης να κατανοήσουμε πλήρως την λειτουργία των συστημάτων αυτών και τον τρόπο με τον οποίο έγινε και συνεχίζει να γίνεται η μετάβαση από το παλαιό πλέον GSM στο 3G.

Επιπρόσθετα καθ' όλο το διάστημα στο οποίο εκπονούσαμε την εργασία μας βελτιώναμε παράλληλα και τις ικανότητες μας στον προγραμματισμό και μπορούμε πλέον να παρέχουμε λύσεις για απλά αλλά και πιο σύνθετα προβλήματα.

Εν καιρώ η εφαρμογή η οποία υλοποιήσαμε, όπως και όλες οι εφαρμογές, θα χρειαστεί βελτιώσεις, αλλαγές και προσθήκες οι οποίες ευελπιστούμε να αποτελέσουν πρόκληση στους νέους συναδέλφους μας και να αγαπήσουν το συγκεκριμένο θέμα όπως κάναμε και εμείς αρχικά.

Τέλος θα θέλαμε να ευχαριστήσουμε θερμά τον καθηγητή μας Κο Κόκκινο Ευάγγελο για τη βοήθεια, την καθοδήγηση και την υποστήριξη την οποία μας παρείχε όλο αυτό το διάστημα.

## **ΒΙΒΛΙΟΓΡΑΦΙΑ**

### **Κινητές επικοινωνίες 3<sup>ης</sup> Γενιάς**

Σημειώσεις μαθήματος του προγράμματος σπουδών του Τμήματος Ηλεκτρονικής. Καθηγητής Δρ. Ευάγγελος Κόκκινος

<http://msdn.microsoft.com/en-us/vbasic/default.aspx>

Online παροχή βοήθειας για το προγραμματιστικό πακέτο Visual Studio 2005

[www.anite.com](http://www.anite.com)

Πληροφορίες σχετικά με την εταιρία Anite PLC και τις εφαρμογές τις οποίες υλοποιεί για ανάλυση δεδομένων συστημάτων κινητής τηλεφωνίας.

[www.zedgraph.org](http://www.zedgraph.org)

Plugin ανοιχτού κώδικα το οποίο παράγει γραφικές παραστάσεις.

[www.vbdotnetforums.com](http://www.vbdotnetforums.com)

Fora υποστήριξης του προγραμματιστικού πακέτου Visual studio 2005