

2019

Πανεπιστήμιο Δυτικής Αττικής,
Σχολή μηχανικών, Τμήμα
μηχανικών βιομηχανικής
σχεδίασης και παραγωγής

Βασιλάκη Μαρία

[ΣΤΑΤΙΣΤΙΚΗ ΕΠΕΞΕΡΓΑΣΙΑ ΕΝΕΡΓΙΑΚΩΝ ΜΕΓΕΘΩΝ]

Το πρωτόκολλο Modbus. Βιβλιοθήκες λογισμικού. Visual Studio. SQL. Ανάπτυξη εφαρμογής παρακολούθησης για τον αναλυτή ενέργειας Janitza UMG96RM.

Πίνακας περιεχομένων

1. Ανάθεση	5
2. Εισαγωγή	6
3. Ο αναλυτής ενέργειας UMG96RM-E.....	7
4. Το πρωτόκολλο επικοινωνίας Modbus	11
4.1. Τι είναι το Modbus	11
4.2. Πως χρησιμοποιείται.....	11
4.3. Αρχή λειτουργίας.....	11
4.3.1. Το δεκαεξαδικό σύστημα.....	12
4.3.2. ASCII.....	12
4.4. Αποθήκευση δεδομένων.....	13
4.4.1. Slave ID.....	14
4.4.2. Κωδικός εντολής (function code).....	14
4.4.3. Έλεγχος ακεραιότητας δεδομένων (CRC).....	15
4.4.4. Τύποι δεδομένων.....	15
4.4.5. Αποθήκευση byte και word.....	16
4.4.6. Χάρτης Modbus (Modbus Map).....	16
5. Βιβλιοθήκες λογισμικού.....	18
5.1. Easy Modbus.....	18
5.1. EasyModbus.dll.....	18
5.1.2. Εξομοίωση Modbus TCP Server.....	21
5.1.2.1. Setup - Properties.....	22
5.2. Industrial Gadgets ActiveX library.....	23
5.2.1. Panels.....	23
5.2.2. Meters.....	26

5.2.3. LED Displays.....	30
6. Βάσεις δεδομένων.....	33
6.1. Αποθήκευση δεδομένων.....	33
6.2. Ενδιάμεση μνήμη.....	33
6.3. Ταυτοχρονισμός.....	34
6.4. Ανάκτηση δεδομένων.....	34
6.5. Microsoft SQL Server Management Studio.....	35
6.5.1. Σύνδεση στον SQL Server.....	35
6.5.2. Δημιουργία βάσης.....	38
6.5.3. Δημιουργία πίνακα σε βάση δεδομένων.....	39
7. Προσδιορισμός βασικών εντολών που χρησιμοποιούνται στην εφαρμογή.....	41
7.1. C#.....	41
7.1.1. if statements.....	41
7.1.2. try.....	42
7.1.3. Class.....	42
7.1.4. for loop.....	44
7.2 Visual C#.....	44
7.2.1. Visual components - εμφανιζόμενα αντικείμενα.....	44
7.2.1.1. Windows form.....	45
7.2.1.2. Label.....	46
7.2.1.3. TextBox.....	46
7.2.1.4. Button.....	47
7.2.1.5. DateTimePicker.....	49
7.2.1.6. CheckBox.....	49
7.2.1.7. Chart.....	50

7.2.1.8. DataGridView.....	52
7.2.2. Non visual Components - μη εμφανιζόμενα αντικείμενα.....	54
7.2.2.1. Timer.....	54
7.2.2.2. Class.....	54
8. Δημιουργία εφαρμογής.....	56
8.1. Form1.....	56
8.1.1. Form1.cs[Design].....	56
8.1.1. Form1.cs.....	57
8.2. Παρακολούθηση τιμών.....	59
8.2.1. Janitza_UMG96RM.cs[Design].....	59
8.2.2. Janitza_UMG96RM.cs.....	60
8.3. modBusFloats.cs.....	74
8.4. Statistics.....	77
8.5. SQLInterFace.....	79
8.6. Ευρεση μεγεθών από την βάση δεδομένων.....	81
8.6.1. DataQuery.cs[Design].....	81
8.6.1. DataQuery.cs.....	82
8.7. Πίνακας αποθηκευμένων μεγεθός.....	91
8.7.1. DataGrid.cs[Design].....	91
8.7.1. DataGrid.cs.....	94
8.8. Στατιστικά στοιχεία.....	97
8.8.1. StatisticsData.cs[Design].....	97
8.8.2. StatisticsData.cs.....	101
8.9. Γράφημα αποθηκευμένων μεγεθών.....	103
8.9.1. DataChart.cs[Design].....	103

8.9.2. DataChart.cs.....	106
8.10.Εξομοίωση τιμών.....	111
8.10.1. SetValues.cs[Design].....	111
8.10.2. SetValues.cs.....	112
8.11. SQL Database.....	118
9.Επίλογος.....	120
10. Βιβλιογραφία.....	121

1. Ανάθεση

ΘΕΜΑ:	<u>Ανάθεση πτυχιακής εργασίας.</u>
ΤΙΤΛΟΣ:	<u>Στατιστική επεξεργασία ενεργειακών μεγεθών.</u>
ΣΠΟΥΔΑΣΤΗΣ:	Βασιλάκη Μαρία
ΣΤΟΙΧΕΙΑ:	Τηλ.: 6997125681 (email: basilaki.ma@gmail.com)
ΑΝΑΛΗΨΗ:	Μάρτιος 2019.
ΟΛΟΚΛΗΡΩΣΗ:	.
ΕΠΙΒΛΕΠΩΝ ΚΑΘΗΓΗΤΗΣ:	Χρήστος Δρόσος

Περιεχόμενα :

2. Εισαγωγή.
3. Εγχειρίδιο χρήσης του αναλυτή ενέργειας UMG96RM-E.
4. Το πρωτόκολλο επικοινωνίας Modbus.
5. Βιβλιοθήκες λογισμικού.
6. Οι βάσεις Δεδομένων.
7. Προσδιορισμός βασικών εντολών που χρησιμοποιούνται στην εφαρμογή.
8. Δημιουργία εφαρμογής.
9. Επίλογος.
10. Βιβλιογραφία.

2. Εισαγωγή

Είναι γεγονός ότι οι σύγχρονες ανάγκες της βιομηχανίας, για την κατανάλωση ηλεκτρικής ενέργειας συνεχώς αυξάνονται. Παρόλα αυτά οι πηγές ενέργειας, οι οποίες είναι διαθέσιμες, εξαντλούνται και γίνονται όλο πιο δυσπρόσιτες. Εκτός από την προφανή λύση της αξιοποίησης ήπιων μορφών ενέργειας, υπάρχει και η λύση της βέλτιστης διαχείρισης της παραχωμένης ηλεκτρικής ενέργειας, ειδικά σε βιομηχανικούς καταναλωτές.

Η κατανάλωση ενέργειας είναι ένα από τα μεγαλύτερα ανελαστικά έξοδα, για μία βιομηχανική μονάδα, και έτσι προκειμένου να υπάρχει άμεση παρακολούθηση της κατανάλωσης ενέργειας, η μονάδα προμηθεύεται όργανα παρακολούθησης ενέργειας και ενεργειακών μεγεθών. Όμως, η απλή συλλογή δεδομένων δεν αρκεί, αφού ο σκοπός είναι μείωση του ενεργειακού κόστους. Έτσι, οι ενεργειακοί μετρητές που είναι διασκορπισμένοι σε όλη την βιομηχανική μονάδα διασυνδέονται με υπολογιστή, σε αυτόν, τα μετρούμενα μεγέθη παρακολουθούνται σε πραγματικό χρόνο και καταχωρούνται σε βάση δεδομένων. Στην συνέχεια οι αρμόδιοι, π.χ. για την παραγωγική διαδικασία, μπορούν να έχουν πρόσβαση στα καταγεγραμμένα ενεργειακά δεδομένα, είτε στα πρωτότυπα δεδομένα, είτε σε στατιστικά στοιχεία αυτών.

Για την ανάπτυξη της εφαρμογής θα χρησιμοποιηθεί ως αναλυτής ενέργειας ο UMG 96 RM-E της Janitza.

3. Ο αναλυτής ενέργειας UMG96RM-E.

Για την ανάπτυξη της εφαρμογής θα χρησιμοποιηθεί ο βιομηχανικός αναλυτής ενέργειας πεδίου UMG96RM-E της Janitza. Ο αναλυτής αυτός υποστηρίζει το πρωτόκολλο επικοινωνίας Modbus TCP/IP.

Στο παρόν κεφάλαιο αναφέρεται μόνο ο πίνακας διευθύνσεων του Modbus TCP/IP. Το εγχειρίδιο χρήσης του αναφέρεται στην βιβλιογραφία, ως υπερσύνδεση.

Modbus Address	Τύπος δεδομένων	Τύπος register (Ανάγνωση-Εγγραφή)	Μονάδα μέτρησης	Σημειώσεις
19000	Float	Ανάγνωση	V	Τάση L1
19002	Float	Ανάγνωση	V	Τάση L2
19004	Float	Ανάγνωση	V	Τάση L3
19006	Float	Ανάγνωση	V	Τάση L1-L2
19008	Float	Ανάγνωση	V	Τάση L1-L3
19010	Float	Ανάγνωση	V	Τάση L2-L3
19012	Float	Ανάγνωση	A	Ένταση L1
19014	Float	Ανάγνωση	A	Ένταση L2
19016	Float	Ανάγνωση	A	Ένταση L3
19018	Float	Ανάγνωση	A	Ένταση άθροισμα
19020	Float	Ανάγνωση	W	Ισχύς L1
19022	Float	Ανάγνωση	W	Ισχύς L2
19024	Float	Ανάγνωση	W	Ισχύς L3
19026	Float	Ανάγνωση	W	Ισχύς άθροισμα
19028	Float	Ανάγνωση	VA	Φαινόμενη ισχύς L1
19030	Float	Ανάγνωση	VA	Φαινόμενη ισχύς L2
19032	Float	Ανάγνωση	VA	Φαινόμενη ισχύς L3
19034	Float	Ανάγνωση	VA	Φαινόμενη ισχύς άθροισμα
19036	Float	Ανάγνωση	var	Άεργος ισχύς L1

Modbus Address	Τύπος δεδομένων	Τύπος register (Ανάγνωση-Εγγραφή)	Μονάδα μέτρησης	Σημειώσεις
19038	Float	Ανάγνωση	var	Άεργος ισχύς L2
19040	Float	Ανάγνωση	var	Άεργος ισχύς L3
19042	Float	Ανάγνωση	var	Άεργος ισχύς άθροισμα
19044	Float	Ανάγνωση		Συντελεστής ισχύος - συνφ L1
19046	Float	Ανάγνωση		Συντελεστής ισχύος - συνφ L2
19048	Float	Ανάγνωση		Συντελεστής ισχύος - συνφ L3
19050	Float	Ανάγνωση	Hz	Συχνότητα
19052	Float	Ανάγνωση		Περιστροφή μαγνητικού πεδίου
19054	Float	Ανάγνωση	Wh	Πραγματική ενέργεια L1
19056	Float	Ανάγνωση	Wh	Πραγματική ενέργεια L2
19058	Float	Ανάγνωση	Wh	Πραγματική ενέργεια L3
19060	Float	Ανάγνωση	Wh	Πραγματική ενέργεια Άθροισμα
19062	Float	Ανάγνωση	Wh	Πραγματική ενέργεια - κατανάλωση L1
19064	Float	Ανάγνωση	Wh	Πραγματική ενέργεια - κατανάλωση L2
19068	Float	Ανάγνωση	Wh	Πραγματική ενέργεια - κατανάλωση L3
19070	Float	Ανάγνωση	Wh	Πραγματική ενέργεια - κατανάλωση άθροισμα

Modbus Address	Τύπος δεδομένων	Τύπος register (Ανάγνωση-Εγγραφή)	Μονάδα μέτρησης	Σημειώσεις
19072	Float	Ανάγνωση	Wh	Πραγματική ενέργεια - παραδοτέα L1
19074	Float	Ανάγνωση	Wh	Πραγματική ενέργεια - παραδοτέα L2
19076	Float	Ανάγνωση	Wh	Πραγματική ενέργεια - παραδοτέα L3
19078	Float	Ανάγνωση	VAh	Φαινόμενη ενέργεια L1
19080	Float	Ανάγνωση	VAh	Φαινόμενη ενέργεια L2
19082	Float	Ανάγνωση	VAh	Φαινόμενη ενέργεια L3
19084	Float	Ανάγνωση	VAh	Φαινόμενη ενέργεια άθροισμα
19086	Float	Ανάγνωση	varh	Άεργος ενέργεια L1
19088	Float	Ανάγνωση	varh	Άεργος ενέργεια L2
19090	Float	Ανάγνωση	varh	Άεργος ενέργεια L3
19092	Float	Ανάγνωση	varh	Άεργος ενέργεια άθροισμα
19094	Float	Ανάγνωση	varh	Άεργος ενέργεια – επαγωγική L1
19096	Float	Ανάγνωση	varh	Άεργος ενέργεια – επαγωγική L2
19098	Float	Ανάγνωση	varh	Άεργος ενέργεια – επαγωγική L3
19100	Float	Ανάγνωση	varh	Άεργος ενέργεια – επαγωγική άθροισμα
19102	Float	Ανάγνωση	varh	Άεργος ενέργεια – χωρητική L1

Modbus Address	Τύπος δεδομένων	Τύπος register (Ανάγνωση-Εγγραφή)	Μονάδα μέτρησης	Σημειώσεις
19104	Float	Ανάγνωση	varh	Έργος ενέργεια – χωρητική L2
19106	Float	Ανάγνωση	varh	Έργος ενέργεια – χωρητική L3
19108	Float	Ανάγνωση	varh	Έργος ενέργεια – χωρητική άθροισμα
19110	Float	Ανάγνωση	%	Αρμονική τάσης L1
19112	Float	Ανάγνωση	%	Αρμονική τάσης L2
19114	Float	Ανάγνωση	%	Αρμονική τάσης L3
19116	Float	Ανάγνωση	%	Αρμονική έντασης L1
19118	Float	Ανάγνωση	%	Αρμονική έντασης L2
19120	Float	Ανάγνωση	%	Αρμονική έντασης L3

4. Το πρωτόκολλο επικοινωνίας Modbus

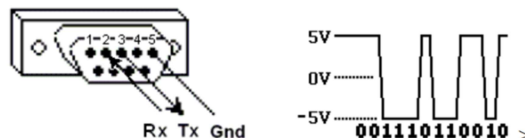
4.1. Τι είναι το Modbus

Το Modbus είναι ένα σειριακό πρωτόκολλο επικοινωνίας, το οποίο αναπτύχθηκε από την Modicon το 1979, για τον προγραμματισμό ελεγκτών πεδίου (PLC – Programmable Logic Controller). Το πρωτόκολλο αυτό μεταδίδει πληροφορίες μεταξύ ηλεκτρονικών συσκευών, μέσω σειριακής σύνδεσης. Η συσκευή που ζητάει δεδομένα λέγεται Modbus Master και η συσκευή που παρέχει δεδομένα λέγεται Modbus Slave. Σε ένα τυπικό δίκτυο Modbus υπάρχει ένας Master και ως διακόσα σαράντα επτά Slaves, ο καθένας με μοναδική διεύθυνση. Ο Master μπορεί και να στέλνει δεδομένα στους Slaves.

4.2. Πως χρησιμοποιείται.

Το Modbus είναι ανοικτό πρωτόκολλο, δηλαδή, παρέχεται σε κατασκευαστές για να χρησιμοποιείται στον εξοπλισμό που κατασκευάζουν, χωρίς να πληρώνουν δικαιώματα. Έτσι έχει επικρατήσει σαν βιομηχανικό πρωτόκολλο επικοινωνίας, και χρησιμοποιείται στην διασύνδεση ηλεκτρονικών συσκευών, από πολλούς κατασκευαστές, σε διάφορους κλάδους της βιομηχανίας. Το Modbus, στις περισσότερες φορές, χρησιμοποιείται για την μετάδοση δεδομένων από διάφορα όργανα σε PLC ή σε σύστημα συλλογής δεδομένων, για παράδειγμα, σε ένα σύστημα που μετράει θερμοκρασία και τα δεδομένα μεταφέρονται σε έναν υπολογιστή. Το Modbus συχνά χρησιμοποιείται στην διασύνδεση υπερκείμενου υπολογιστή (SCADA), με απομακρυσμένα τερματικά (RTU). Υπάρχουν σειριακές εκδόσεις (Modbus RTU and Modbus ASCII) και για δίκτυα Ethernet (Modbus TCP).

4.3. Αρχή λειτουργίας.



Το Modbus μεταδίδει σειριακά μεταξύ συσκευών. Η πιο απλή σύνδεση μέσω των σειριακών θυρών του Master και του Slave. Τα δεδομένα μεταδίδονται σε 0-1 (δυναμικό σύστημα), με «0» είναι -5VDC και «1» είναι 5VDC. Τα δυναμικά δεδομένα μεταδίδονται πολύ γρήγορα. Μία τυπική ταχύτητα μετάδοσης δεδομένων είναι 9600 baud (bit ανά δευτερόλεπτο).

4.3.1. Το δεκαεξαδικό σύστημα.

Σε περίπτωση αντιμετώπισης βλάβης, είναι χρήσιμο να παρακολουθούνται τα δεδομένα ακατέργαστα. Όμως, το δυναμικό σύστημα είναι δυσανάγνωστο, έτσι προέκυψε η ανάγκη τα ακατέργαστα δεδομένα να συνδυάζονται σε τετράδες bits και να αναπαριστούνται στο δεκαεξαδικό σύστημα (HEX).

Ισοτιμία δυναμικού – δεκαεξαδικού συστήματος :

Δυναμικός αριθμός	Δεκαεξαδικός αριθμός
0000	0
0001	1
0010	2
0011	3
0100	4
0101	5
0110	6
0111	7
1000	8
1001	9
1010	A
1011	B
1100	C
1101	D
1110	E
1111	F

Κάθε σύνολο από 8 bit ονομάζεται byte, και αντιπροσωπεύουν έναν από 256 χαρακτήρες (00 ως FF).

4.3.2. ASCII.

Το σύστημα ASCII (American Standard Code for Information Interchange), όπως και το δεκαεξαδικό σύστημα, συνδυάζει 4 bits, και σε κάθε byte αντιστοιχεί ένας από 256 χαρακτήρες. Κάποιοι από τους

χαρακτήρες είναι:

Δεκαδικό (βάση 10)	Δυαδικό (βάση 2)	Δεκαεξαδικό (βάση 16)	ASCII (βάση 256)
0	0000 0000	00	Null
1	0000 0001	01	"
34	0010 0010	22	#
35	0010 0011	23	\$
36	0010 0100	24	%
47	0010 1111	2F	/
48	0010 0000	30	0
49	0011 0001	31	1
56	0011 1000	38	8
57	0011 1001	39	9
58	0011 1010	3A	:
64	0100 0000	40	@
65	0100 0001	41	A
66	0100 0010	42	B
89	0101 1001	59	Y
90	0101 1010	5A	Z
91	0101 1011	5B	[
95	0101 1111	5F	~
96	0110 0000	60	`
97	0110 0001	61	a
122	0111 0001	7A	Z
123	0111 1011	7B	{
174	1010 1110	AE	®
255	1111 1111	FF	

4.4. Αποθήκευση δεδομένων.

Τα δεδομένα σε μία συσκευή Modbus Slave αποθηκεύονται σε τέσσερεις διαφορετικούς πίνακες. Δύο πίνακες αποθηκεύουν δυαδικά δεδομένα (coils) και οι άλλοι δύο αποθηκεύουν αριθμούς (registers). Οι πίνακες είναι ανάγνωσης (read only table) και πίνακες εγγραφής-ανάγνωσης (read - write table). Κάθε πίνακας αποθηκεύει 9999 τιμές. Το μήκος κάθε πίνακα coil έχει διευθύνσεις 0000 ως και 270E. Το μήκος κάθε πίνακα register έχει διευθύνσεις 0000 ως και 270E.

Coil/Register – Αριθμοί	Διεύθυνση	Τύπος	Όνομα πίνακα
1-9999	0000-270E	Εγγραφή- ανάγνωση	Discrete Output Coils
10001-19999	0000-270E	Ανάγνωση	Discrete Input Conducts
30001-39999	0000-270E	Ανάγνωση	Analog Input Registers
40001-49999	0000-270E	Εγγραφή- ανάγνωση	Analog Output Holding Registers

Οι αριθμοί Coil/Register χρησιμοποιούνται σαν ονόματα τοποθεσίας και δεν εμφανίζονται στα μηνύματα που ανταλλάσσονται. Οι διευθύνσεις χρησιμοποιούνται στα μηνύματα.

4.4.1. Slave ID.

Σε κάθε συσκευή Modbus Slave αποδίδεται μία μοναδική διεύθυνση (1 ως 247). Όταν ο Master ζητάει δεδομένα στο πρώτο byte στέλνει την διεύθυνση του Slave, έτσι κάθε συσκευή γνωρίζει ποια πακέτα τον αφορούν, και αγνοεί τα υπόλοιπα.

Για να αρθεί ο περιορισμός των 247 διευθύνσεων το πρωτόκολλο μπορεί να χρησιμοποιεί και δεύτερο byte για την διευθητοδότηση. Αν γίνει αυτή η αλλαγή πρέπει να ενημερωθούν όλοι οι Slave και ο Master. Αν χρησιμοποιηθεί δεύτερο byte, οι διευθύνσεις των Slave μπορεί να φτάσει ως 65535.

4.4.2. Κωδικός εντολής (function code).

Το δεύτερο byte που στέλνει ο Master είναι ο κωδικός εντολής. Αυτός ο αριθμός λέει στον Slave ποιον πίνακα να προσπελάσει και αν πρέπει να διαβάσει ή να γράψει δεδομένα στον πίνακα.

Κωδικός εντολής	Χειρισμός	Όνομα πίνακα
01 (01 16δικό)	Ανάγνωση	Discrete Output Coils
05 (05 16δικό)	Μονή εγγραφή	Discrete Output Coil
15 (0F 16δικό)	Πολλαπλή εγγραφή	Discrete Output Coils
02 (02 16δικό)	Ανάγνωση	Discrete Input Contacts
04 (04 16δικό)	Ανάγνωση	Analog Input Registers
03 (03 16δικό)	Ανάγνωση	Analog Output Holding Registers
06 (06 16δικό)	Μονή εγγραφή	Analog Output Holding Register

Κωδικός εντολής	Χειρισμός	Όνομα πίνακα
16 (10 16δικό)	Πολλαπλή εγγραφή	Analog Output Holding Registers

4.4.3. Έλεγχος ακεραιότητας δεδομένων (CRC).

Ο έλεγχος ακεραιότητας δεδομένων γίνεται με την μέθοδο κυκλικού επαναπροσδιορισμού - CRC (Cycle Redundancy Check), προσθέτοντας δύο byte στο τέλος κάθε μηνύματος, για την ανίχνευση σφαλμάτων. Κάθε byte του μηνύματος υπολογίζεται στο CRC. Η συσκευή που λαμβάνει τα δεδομένα ελέγχει το πακέτο του CRC και το συγκρίνει με αυτό του Master. Εάν υπάρχει σφάλμα τα πακέτα ελέγχου θα είναι διαφορετικά.

4.4.4 Τύποι δεδομένων

Τύπος	Σχόλια
16-bit unsigned integer	Ακέραιος αριθμός μεταξύ 0 και 65535
16-bit signed integer	Ακέραιος αριθμός μεταξύ -32768 και +32767
two character ASCII string	Δύο στοιχεία γραμματοσειράς
discrete on/off value	Δεδομένα σε δυαδική μορφή «0» και «1»
32-bit unsigned integer	Ακέραιος αριθμός μεταξύ 0 και 4294967295
32-bit signed integer	Ακέραιος αριθμός μεταξύ -2147483648 και +2147483648
32-bit IEEE floating point number	Δεκαδικός αριθμός κινητής υποδιαστολής
four character ASCII string	Τέσσερα στοιχεία γραμματοσειράς

4.4.5. Αποθήκευση byte και word.

Το πρωτόκολλο Modbus δεν προσδιορίζει επακριβώς πως τα δεδομένα αποθηκεύονται στους registers. Για αυτό, κάποιοι κατασκευαστές, αποθηκεύουν και μεταδίδουν το πιο σημαντικό byte, ακολουθούμενο από το λιγότερο σημαντικό byte. Όμως αυτός ο τρόπος εκπομπής δεν είναι δεσμευτικός, και έτσι άλλοι κατασκευαστές, αποθηκεύουν πρώτα το λιγότερο σημαντικό byte.

Το εύρος διευθύνσεων των register είναι 40001 ως 49999, αυτό σημαίνει ότι υπάρχουν 9999 registers, όμως σε κάποιες περιπτώσεις το

διαθέσιμο εύρος δεν αρκεί υπάρχει η δυνατότητα υπέρ-εξαπλασιαστούν, δηλαδή, οι διαθέσιμες διευθύνσεις είναι 40001 ως 105536 (65536 registers).

4.4.6. Χάρτης Modbus (Modbus Map).

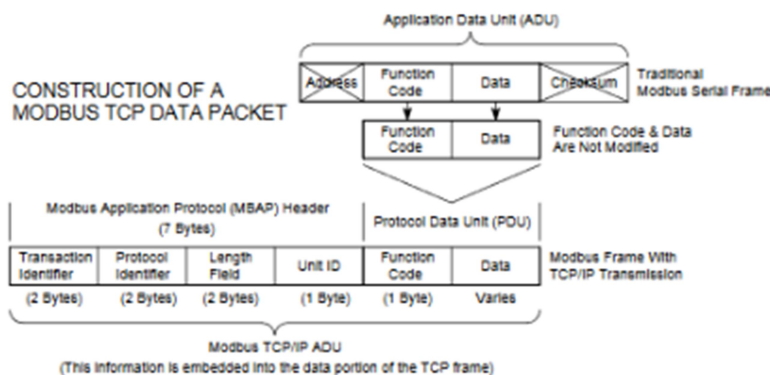
Ο χάρτης Modbus είναι μία λίστα των συσκευών Slave, που καθορίζει:

- Ποία είναι τα δεδομένα
- Που αποθηκεύονται τα δεδομένα (πίνακες και διευθύνσεις)
- Πως αποθηκεύονται τα δεδομένα (τύπος δεδομένων, σειρά byte και word)

Κάποιες συσκευές κατασκευάζονται με προκαθορισμένο χάρτη, ενώ άλλες κατασκευάζονται δίνοντας την δυνατότητα σε κάθε χρήστη, να προσδιορίσουν ή και να προγραμματίσουν τον Modbus Map, σύμφωνα με τις ανάγκες του.

4.5. Modbus TCP/IP.

Το Modbus TCP/IP (γνωστό ως Modbus-TCP) είναι απλός το Modbus RTU συνδεδεμένο με TCP και τρέχει σε ένα δίκτυο Ethernet. Η δομή των μηνυμάτων επικοινωνίας του Modbus είναι το πρωτόκολλο εφαρμογής (application protocol) και προσδιορίζει τους κανόνες οργάνωσης και διερμηνείας των δεδομένων που διακινούνται στο δίκτυο. Το πρωτόκολλο TCP είναι υπεύθυνο για την διακίνηση πακέτων δεδομένων, με έλεγχο σφαλμάτων, ενώ, το πρωτόκολλο IP διασφαλίζει την δρομολόγηση των πακέτων δεδομένων στην σωστή συσκευή. Δηλαδή το Modbus TCP/IP είναι ένα πρωτόκολλο που εμπεριέχεται μέσα σε ένα Ethernet TCP/IP. Αυτό σημαίνει ότι δεν χρησιμοποιείται ο έλεγχος ορθότητας δεδομένων του Modbus, αλλά του TCP/IP.



Επεξήγηση:

- Application. Data Unit(ADU). Οι εντολές του Modbus και τα δεδομένα εσωκλείονται στο τηλεγράφημα του TCP/IP χωρίς να αλλοιώνεται.
- Transaction/invocation Identifier (2 Bytes). Αυτό το πεδίο χρησιμοποιείται όταν στέλνονται πολλαπλά πακέτα δεδομένων στην ίδια σύνδεση TCP χωρίς να περιμένει την απάντηση του client.
- Protocol Identifier (2 bytes). Αυτό το πεδίο έχει πάντα τιμή 0 για το Modbus.
- Length (2 bytes). Αυτό το πεδίο μετρά τα εναπομείναντα πεδία (unit identifier, function code, data).
- Unit Identifier (1 byte). Αυτό το πεδίο χρησιμοποιείται για να αναγνωρίζεται μια συσκευή που δεν βρίσκεται σε Ethernet (serial bridging). Παίρνει τιμές 00 και FF.

5. Βιβλιοθήκες λογισμικού.

5.1. Easy Modbus.

5.1.1. EasyModbus.dll.

Το πακέτο λογισμικού υποστηρίζει τα πρωτόκολλα:

- Modbus TCP/IP.
- Modbus UDP.
- Modbus RTU.

Επίσης υποστηρίζει τις εντολές:

- Read Coils (FC1).
- Read Discrete Inputs (FC2).
- Read Holding Registers (FC3).
- Read Input Registers (FC4).
- Write Single Coil (FC5).
- Write Single Register (FC6).
- Write Multiple Coils (FC15).
- Write Multiple Registers (FC16).

Στην ιστοσελίδα του πακέτου λογισμικού δίδετε το εξής παράδειγμα, για την διαδικασία ανάγνωσης και εγγραφής register:

```
namespace ModbusClientApplication
{
    class Program
    {
        public static void Main(string[] args)
        {
            ModbusClient modbusClient = new ModbusClient("190.201.100.100",
502); //Διεύθυνση και πόρτα του Modbus-TCP-Server
            modbusClient.Connect(); //Σύνδεση με τον Server
            modbusClient.WriteMultipleCoils(4, new bool[] {true, true, true, true, true, true,
true, true, true, true}); //Εγγραφή πολλαπλών Coils (δυναμικά μεγέθη) ξεκινώντας από
την διεύθυνση 4
            bool[] readCoils = modbusClient.ReadCoils(9,10); //Ανάγνωση 10
Coils από τον Server, ξεκινώντας από την διεύθυνση 10
        }
    }
}
```

```

    int[] readHoldingRegisters =
modbusClient.ReadHoldingRegisters(0,10); //Ανάγνωση 10 registers (αναλογικά
μεγέθη) από τον Server, ξεκινώντας από τη διεύθυνση 1

    // Εμφάνιση στην κονσόλα των windows
    for (int i = 0; i < readCoils.Length; i++)
        Console.WriteLine("Value of Coil " + (9 + i + 1) + " " + readCoils[i].ToString());

    for (int i = 0; i < readHoldingRegisters.Length; i++)
        Console.WriteLine("Value of HoldingRegister " + (i + 1) + " "+
readHoldingRegisters[i].ToString());
    modbusClient.Disconnect(); //Αποσύνδεση από τον
Server
    Console.WriteLine("Press any key to continue . . . ");
    Console.ReadKey(true);
}
}
}

```

Κάποιες εντολές που χρησιμοποιούνται στην εφαρμογή που θα αναπτυχθεί είναι :

Εντολές -κλάσεις	Περιγραφή
ModbusClient ()	Η κλάση που αντιστοιχεί στο αντικείμενο .NET που συνδέεται με τον Modbus Server.

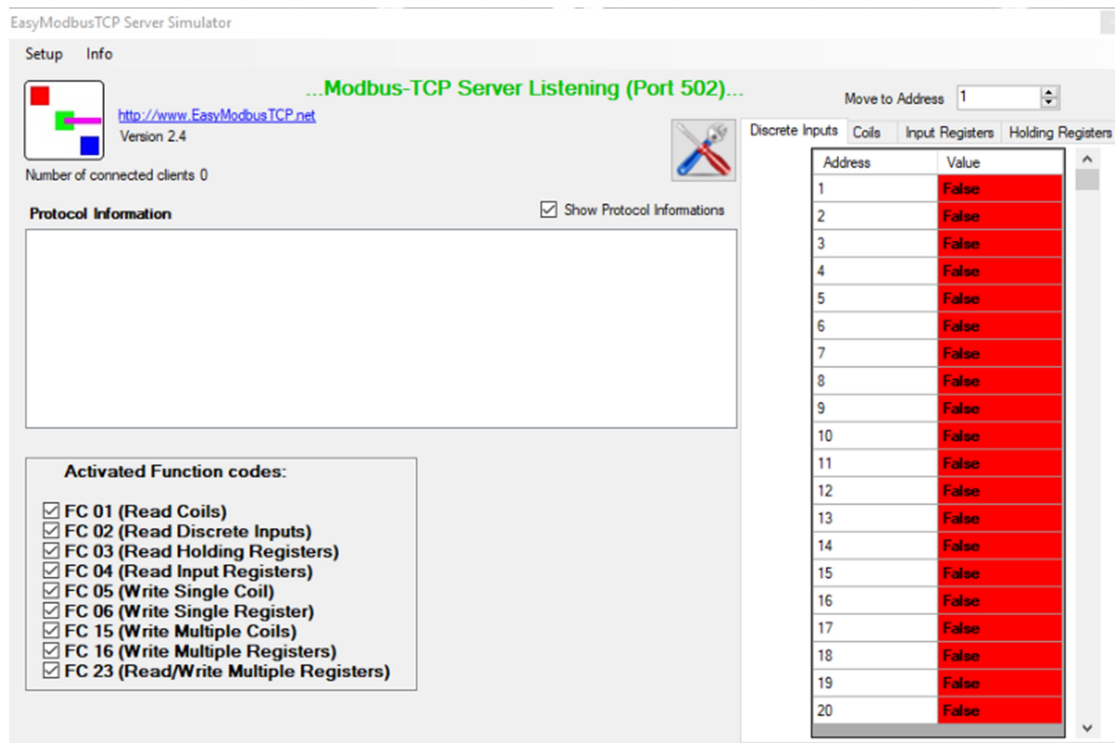
Εντολές -κλάσεις	Περιγραφή
<p>Connect(string iPad dress, int port)</p>	<p>Εντολή σύνδεσης του αντικειμένου .NET με τον Modbus Server. Παράμετροι :</p> <ol style="list-style-type: none"> 1. string iPad dress : η διεύθυνση δικτύου του Server. 2. int port : η διαθέσιμη πόρτα του Server για σύνδεση. Για την εφαρμογή που θα αναπτυχθεί είναι 502. <p>Αν δεν είναι γνωστή η διεύθυνση του Server μπορεί να βρεθεί εισάγοντας την εντολή ipconfig /all στην κονσόλα του Server,το αποτέλεσμα είναι:</p>  <p>Επιλέγετε η διεύθυνση IPv4 Address. (Preferred)</p>

Εντολές -κλάσεις	Περιγραφή
<pre>ReadHoldingRegisters(int startingAddress, int quantity)</pre>	<p>Ανάγνωση πολλαπλών καταχωριτών (register (αναλογικά μεγέθη)). Παράμετροι :</p> <ol style="list-style-type: none"> 1. <code>int startingAddress</code> : η διεύθυνση από την οποία ξεκινά η ανάγνωση. 2. <code>int quantity</code> : ο αριθμός των καταχωριτών που θα αναγνωστούν. <p>Η συνάρτηση επιστέφει πίνακα ακεραίων int[].</p>
<pre>WriteMultipleRegisters(int startingAddress, int[] values)</pre>	<p>Εγγραφή πολλαπλών καταχωριτών (register (αναλογικά μεγέθη)). Παράμετροι :</p> <ol style="list-style-type: none"> 1. <code>int startingAddress</code> : η διεύθυνση από την οποία ξεκινά η εγγραφή. 2. <code>int[] values</code> : πίνακας ακεραίων με τις τιμές που θα εγγραφούν.

5.1.2. Εξομοίωση Modbus TCP Server.

Για την ανάπτυξη της εφαρμογής θα χρησιμοποιηθεί εξομοιωτής Modbus TCP/IP Server. Αυτή η μέθοδος αποτελεί μια συνηθισμένη λύση για την ανάπτυξη εφαρμογών, όταν δεν έχει ολοκληρωθεί ακόμα η παραγγελία του υλικού. Επιπλέον, δεν ήταν δυνατόν να γίνει παραγγελία του μετρητή ενέργειας, λόγω αδυναμίας έκδοσης τιμολογίου.

Ο εξομοιωτής που θα χρησιμοποιηθεί είναι ο **Modbus TCP Server simulator.NET**:

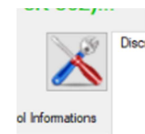


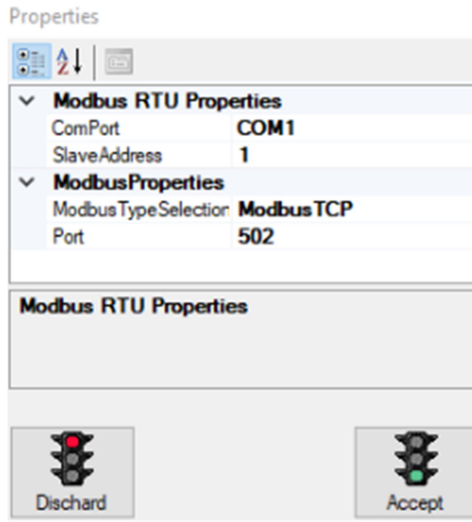
Στο παράθυρο εμφανίζονται οι εξής πληροφορίες και επιλογές:

- **Port 502.** Πόρτα σύνδεσης.
- **Number of connected clients 0.** Το πλήθος των συνδεδεμένων client.
- **Activated Function codes.** Σε αυτή την περιοχή ορίζονται οι ενεργοί πίνακες του server. Οι πίνακες ορίζονται από το πρωτόκολλο Modbus.
- Στο δεξί μέρος εμφανίζονται οι πίνακες μνήμης του εξομοιωτή (**Discrete Input – Coils – Input Registers – Holding Registers**).

5.1.2.1. Setup – Properties.

Πατώντας με το ποντίκι την επιλογή Setup, ή το κουμπί εμφανίζετε το παράθυρο:





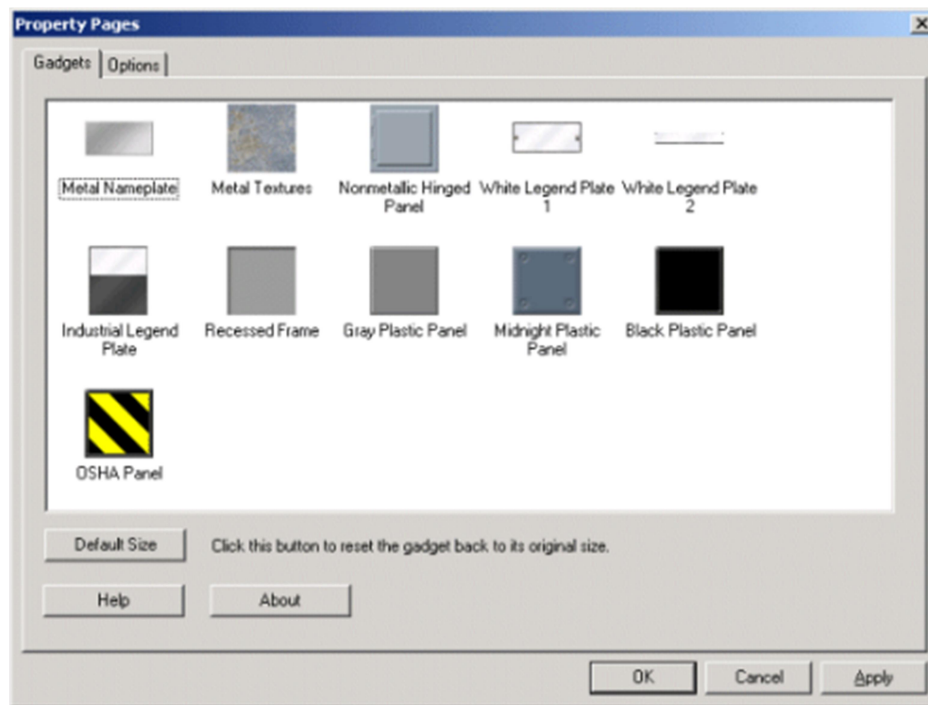
Σε αυτό το παράθυρο ορίζονται οι εξής ιδιότητες του server:

- **Modbus RTU Properties:**
 - **ComPort.** Η σειριακή πόρτα σύνδεσης του server για σειριακή επικοινωνία.
 - **Slave Address.** Η διεύθυνση της σειριακής πόρτας για σειριακό πρωτόκολλο επικοινωνίας.
- **Modbus Properties :**
 - **Modbus Type Selector.** Στο μενού μπορεί να επιλεγεί ο τρόπος σύνδεσης του server (**ModbusTCP – ModbusUDP – ModbusRTU**).
 - **Port.** Πόρτα σύνδεσης του server για επικοινωνία Modbus TCP/IP.

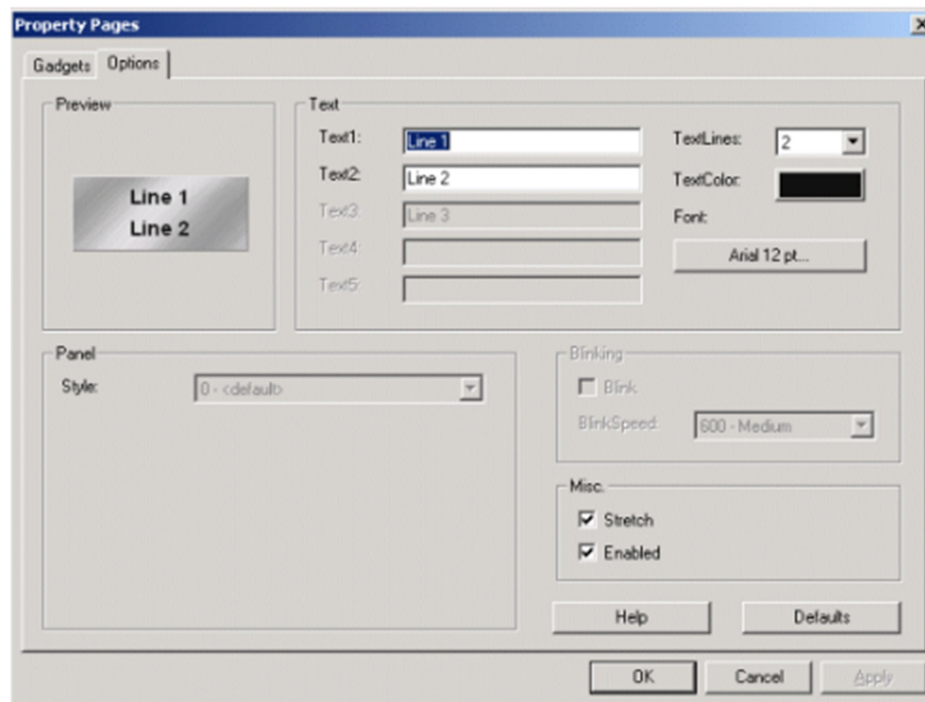
5.2. Industrial Gadgets ActiveX library.

Η βιβλιοθήκη Industrial Gadgets είναι μια συλλογή από χειριστήρια (**Sliders**), πλαίσια απεικόνισης (**Meters, LED Displays, and Message Displays**), πλαίσια (**Panels**). Τα αντικείμενα αυτού του πακέτου έχουν σχεδιαστεί για να μοιάζουν με βιομηχανικά, παρόλα αυτά, το πακέτο χρησιμοποιείται από πολλές διαφορετικές εφαρμογές, επειδή, φαίνονται οικία.

5.2.1. Panels.

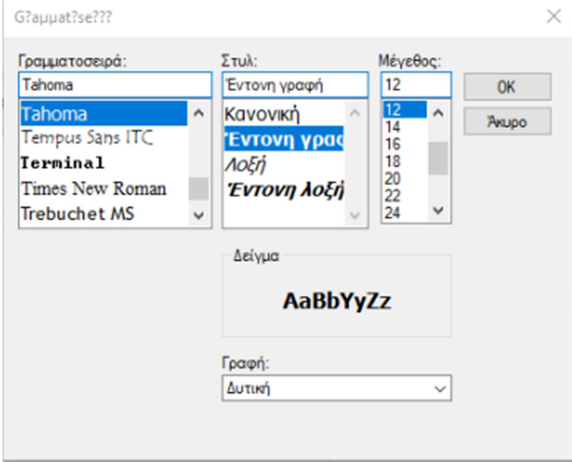


Σε αυτή την καρτέλα επιλέγεται η εμφάνιση του αντικειμένου. Πατώντας το κουμπί **Default Size** επαναφέρει το προκαθορισμένο μέγεθος του αντικειμένου.

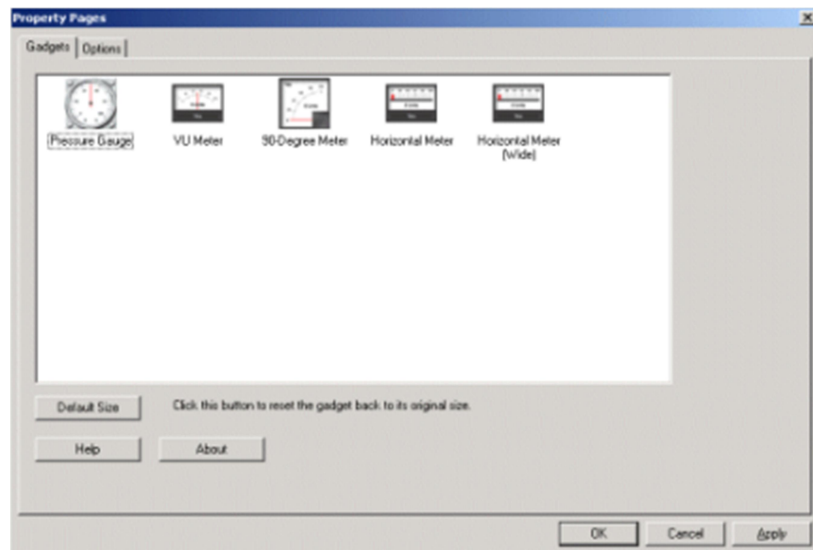


Σε αυτήν την καρτέλα υπάρχουν επιλογές για την εμφάνιση του αντικειμένου:

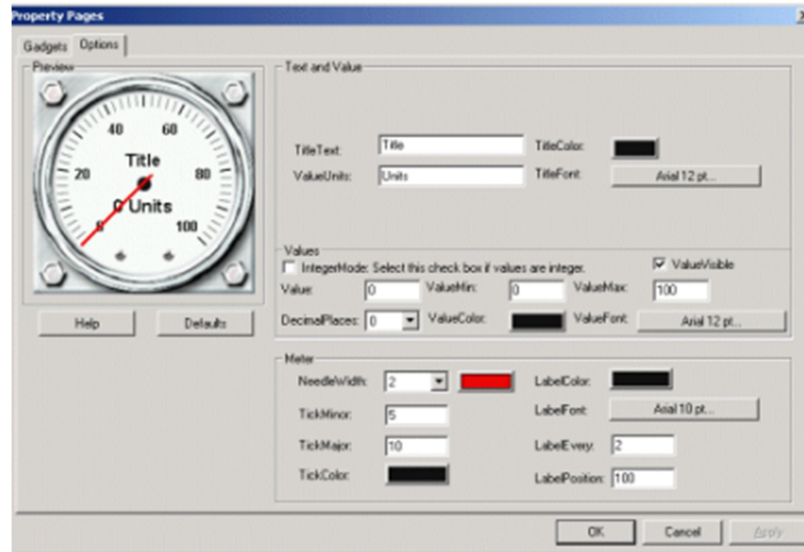
Στοιχείο καρτέλας	Λειτουργία
Text1	Σε αυτά τα πεδία γράφεται το κείμενο που θα εμφανίζεται στο αντικείμενο, χωρισμένο σε γραμμές.
Text2	
Text3	
Text4	
Text5	
TextLines	Σε αυτό το πεδίο επιλέγεται το πλήθος των γραμμών που θα έχει το αντικείμενο.
TextColor	<div data-bbox="816 787 1193 1354" data-label="Image"> </div> <p data-bbox="586 1465 1247 1501">Επιλογή του χρώματος της γραμματοσειράς.</p>

Στοιχείο καρτέλας	Λειτουργία
<p>Font</p>	 <p>Επιλογή της γραμματοσειράς, του μεγέθους και επιλογές εμφάνισης.</p>
<p>Stretch</p>	<p>Αλλοιώνει τις προκαθορισμένες αναλογίες μεγέθους του αντικειμένου.</p>
<p>Enabled</p>	<p>Εμφάνιση – απόκρυψη του αντικειμένου.</p>
<p>Defaults</p>	<p>Επαναφορά του αντικειμένου στις προκαθορισμένες τιμές.</p>

5.2.2. Meters.

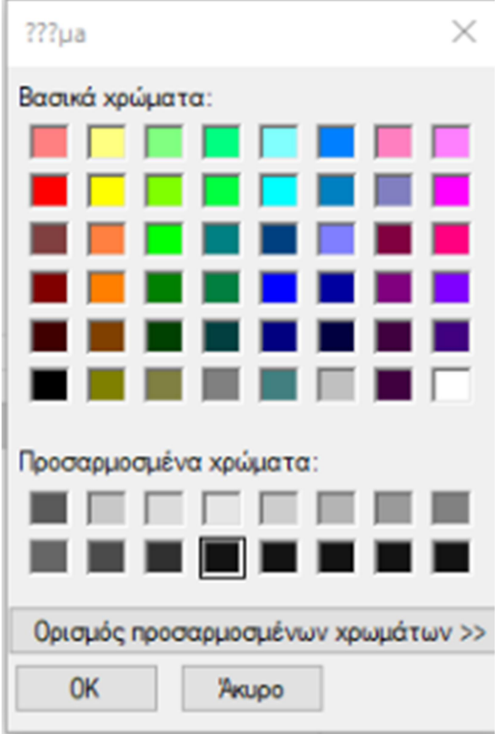


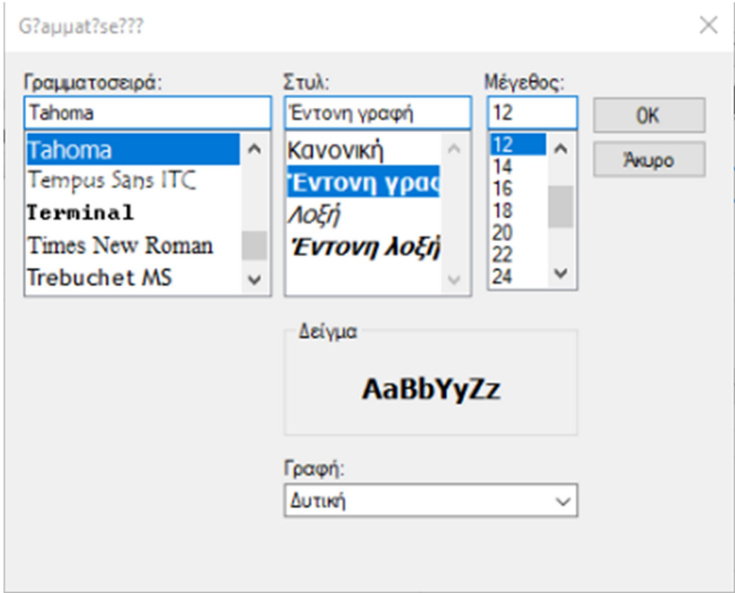
Σε αυτή την καρτέλα επιλέγεται η εμφάνιση του αντικειμένου. Πατώντας το κουμπί **Default Size** επαναφέρει το προκαθορισμένο μέγεθος του αντικειμένου.



Σε αυτήν την καρτέλα υπάρχουν επιλογές για την εμφάνιση του αντικειμένου:

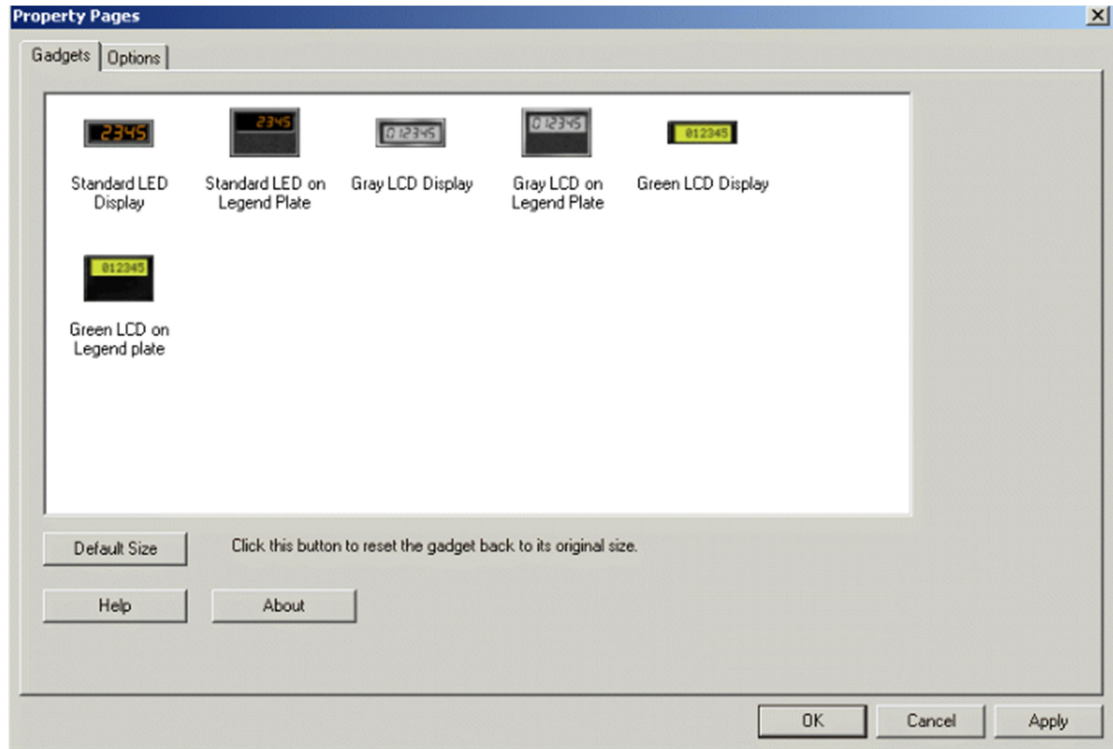
Στοιχείο καρτέλας	Λειτουργία
Defaults	Επαναφορά του αντικειμένου στις προκαθορισμένες τιμές.
TitleTexts	Σε αυτό το πεδίο εισάγεται ο τίτλος του αντικειμένου.
ValueUnits	Σε αυτό το πεδίο εισάγεται η μονάδα μέτρησης του μετρούμενου μεγέθους.

Στοιχείο καρτέλας	Λειτουργία
TitleColor	
ValueColor	
NeedleWidth - color	
TickColor	
LabelColor	<p>Πατώντας τα αντίστοιχα κουμπιά, αλλάζει το χρώμα τις αντίστοιχης ιδιότητας:</p> <ul style="list-style-type: none"> • Χρώμα τίτλου. • Χρώμα τιμών. • Χρώμα βελόνας • Χρώμα γραμμών κλίμακας. <p>Χρώμα μετρήσεων – κλίμακας μέτρησης.</p>

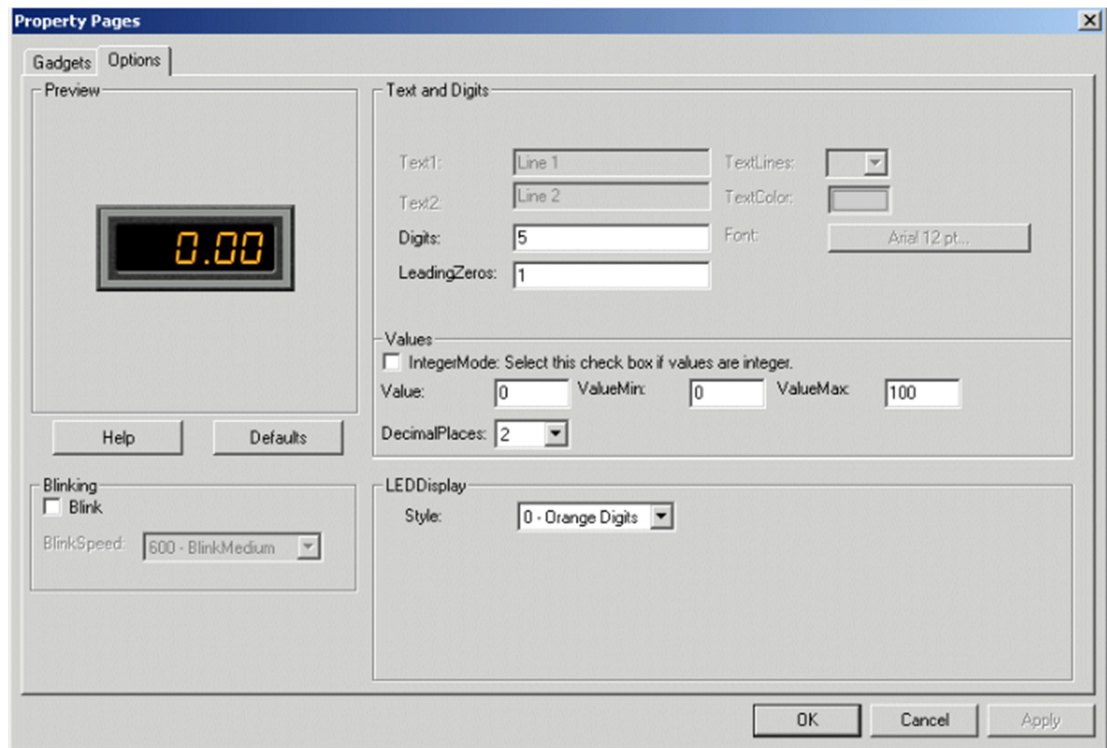
Στοιχείο καρτέλας	Λειτουργία	
TitleFont		
ValueFont		
LabelFont		<p>Επιλογή της γραμματοσειράς, του μεγέθους και επιλογές εμφάνισης για:</p> <ul style="list-style-type: none"> • Τον τίτλο • Τα αριθμητικά μεγέθη • Της κλίμακας μέτρησης.
IntigerMode	Επιλέγοντας αυτή την ετικέτα η απεικονιζόμενη μέτρηση δεν εμφανίζει δεκαδικά ψηφία.	
ValueVisible	Εμφανίζει την μέτρηση του αναλογικού μεγέθους, με αριθμητική απεικόνιση.	
Value	Αρχική τιμή μέτρησης, του μετρούμενου μεγέθους.	
ValueMin	Ελάχιστη τιμή της κλίμακας.	
ValueMax	Μέγιστη τιμή της κλίμακας.	
DecimalPlaces	Πλήθος δεκαδικών ψηφίων, που εμφανίζεται στην κλίμακα του οργάνου.	
NeedleWidth	Ορισμός του πάχους της βελόνας του οργάνου.	
TickMinor	Ορισμός του μήκους των δευτερευόντων σημείων της κλίμακας.	
TickMajor	Ορισμός του μήκους των πρωτευόντων σημείων της κλίμακας.	
LabelEvery	Εμφάνιση αριθμού της κλίμακας σε σχέση με τα Major Ticks .	

Στοιχείο καρτέλας	Λειτουργία
LabelPosition	Θέση των αριθμών της κλίμακας, σε σχέση με το κάτω μέρος του αντικειμένου.

5.2.3. LED Displays.



Σε αυτή την καρτέλα επιλέγεται η εμφάνιση του αντικειμένου. Πατώντας το κουμπί **Default Size** επαναφέρει το προκαθορισμένο μέγεθος του αντικειμένου.



Σε αυτήν την καρτέλα υπάρχουν επιλογές για την εμφάνιση του αντικειμένου:

Στοιχείο καρτέλας	Λειτουργία
Defaults	Επαναφορά του αντικειμένου στις προκαθορισμένες τιμές.
Digits	Ορισμός του μέγιστου αριθμού ψηφίων που θα απεικονίζονται.
LeadingZeros	Ορισμός του μέγιστου αριθμού ψηφίων που θα απεικονίζονται πριν την υποδιαστολή (ακέραιο μέρος).
IntegerMode	Επιλέγοντας αυτή την ετικέτα η απεικονιζόμενη μέτρηση δεν εμφανίζει δεκαδικά ψηφία.
Value	Αρχική τιμή μέτρησης, του μετρούμενου μεγέθους.
ValueMin	Ελάχιστη τιμή που μπορεί να απεικονιστεί.
ValueMax	Μέγιστη τιμή που μπορεί να απεικονιστεί.
DecimalPlaces	Πλήθος δεκαδικών ψηφίων, που εμφανίζεται στην κλίμακα του οργάνου.
Blink	Αναβόσβημα της απεικονιζόμενης τιμής.

Στοιχείο καρτέλας	Λειτουργία
<i>BlinkSpeed</i>	Ταχύτητα αναβοσβήματος: <ul style="list-style-type: none">• 300 – BlinkFast. Κάθε 300ms.• 600 – BlinkMedium. Κάθε 600ms• 900 – BlinkSlow. Κάθε 900ms
<i>Style</i>	Χρώμα απεικόνισης: <ul style="list-style-type: none">• 0 – Orange digits. Πορτοκαλί ψηφία.• 1 – Red digits. Κόκκινα ψηφία.

6. Βάσεις δεδομένων.

Για την αποθήκευση των δεδομένων του ενεργειακού αναλυτή θα χρησιμοποιηθεί ο SQL Server της Microsoft. Ο SQL Server είναι μια σχεσιακή βάση δεδομένων. Οι κύριες γλώσσες που χρησιμοποιούνται είναι η *T-SQL* και η *ANSI SQL*. Η κύρια μονάδα αποθήκευσης στοιχείων είναι μια βάση δεδομένων, η οποία αποτελείται από μια συλλογή πινάκων και κώδικα.

6.1. Αποθήκευση δεδομένων.

Η κεντρική βάση δεδομένων του SQL υποστηρίζει διαφορετικούς τύπους, συμπεριλαμβανομένων των ακεραίων αριθμών, αριθμών κινητής υποδιαστολής (floating point – float), δεκαδικών, αλφαριθμητικών, Varchar, δυαδικών αριθμών, κειμένων. Επιτρέπει επίσης καθορισμένους από το χρήστη σύνθετους τύπους δεδομένων (UDTs), δηλαδή τύπους που βασίζονται στους βασικούς τύπους αλλά μπορούν να τροποποιηθούν. Το αρχείο καταγραφής το οποίο περιέχει όλες τις πρόσφατες αλλαγές στη βάση δεδομένων. Ο χώρος αποθήκευσης που διατίθεται σε μια βάση δεδομένων διαιρείται σε διαδοχικά αριθμημένες σελίδες, κάθε μία από τις οποίες έχει μέγεθος 8 KB.

6.2. Ενδιάμεση μνήμη.

Οι σελίδες αποθηκεύονται στην ενδιάμεση (buffer) μνήμη RAM για να ελαχιστοποιηθεί η μεταφορά δεδομένων προς και από τον σκληρό δίσκο. Οποιαδήποτε σελίδα 8 KB μπορεί να είναι αποθηκευμένη στη μνήμη, και το σύνολο όλων των σελίδων που αποθηκεύονται σε μία περίοδο καλείται λανθάνουσα μνήμη (cache). Το ποσό μνήμης που είναι διαθέσιμο στον κεντρικό διακομιστή SQL αποφασίζει πόσες σελίδες θα εναποθηκευθούν στη λανθάνουσα μνήμη. Ο SQL Server έχει διαφορετικούς αλγόριθμους για την καλύτερη απόδοση της λανθάνουσας μνήμης. Ο κεντρικός διακομιστής SQL εξασφαλίζει ότι οποιαδήποτε αλλαγή στα στοιχεία γίνεται μέσω transactions (συναλλαγές), και εξασφαλίζει ότι οποιαδήποτε λειτουργία είτε ολοκληρώνεται συνολικά, είτε αποτυγχάνει συνολικά, αλλά δεν αφήνει ποτέ τη βάση δεδομένων σε μία ενδιάμεση κατάσταση.

Χρησιμοποιώντας τις συναλλαγές αυτές, μια ακολουθία εντολών μπορεί να αποτελέσει μία ομάδα, με την προϋπόθεση ότι είτε όλες οι εντολές θα πετύχουν ή καμία δεν θα πετύχει. Ο κεντρικός διακομιστής SQL επιτρέπει σε πολλούς πελάτες να χρησιμοποιούν την ίδια βάση δεδομένων ταυτόχρονα. Υπό αυτήν τη μορφή, θα πρέπει να ελέγξει την

ταυτόχρονη πρόσβαση στα κοινά στοιχεία, για να εξασφαλίσει την ακεραιότητα των στοιχείων.

6.3. Ταυτοχρονισμός.

Ο κεντρικός διακομιστής SQL παρέχει δύο τρόπους ελέγχου του ταυτοχρονισμού: απαισιόδοξος ταυτοχρονισμός και αισιόδοξος ταυτοχρονισμός. Όταν ο απαισιόδοξος έλεγχος ταυτοχρονισμού χρησιμοποιείται, ο κεντρικός διακομιστής SQL ελέγχει την ταυτόχρονη πρόσβαση με τη χρησιμοποίηση κλειδώματος (locks). Τα κλειδώματα μπορούν είτε να διαμοιράζονται είτε να είναι αποκλειστικά. Το αποκλειστικό κλειδωμά χορηγεί στο χρήστη την αποκλειστική πρόσβαση στα στοιχεία (συνήθως κατά την αλλαγή στοιχείων) κανένας άλλος χρήστης δεν μπορεί να έχει πρόσβαση στα στοιχεία εφόσον υπάρχει κλειδωμά. Το κοινό κλειδωμά χρησιμοποιείται όταν διαβάζεται κάποιο στοιχείο πολλαπλοί χρήστες μπορούν να διαβάσουν από τα στοιχεία που κλειδώνονται με ένα κοινό κλειδωμά, αλλά να μην αποκτούν αποκλειστικό. Το κλειδωμά μπορεί να εφαρμοστεί σε διαφορετικά επίπεδα σε ολόκληρους πίνακες, σελίδες ή ακόμα και σε διαφορετικές γραμμές. Το επίπεδο που χρησιμοποιείται καθορίζεται σε μια βάση δεδομένων από περίπλοκους αλγόριθμους του SQL Server, οι οποίοι έχουν σκοπό την γρηγορότερη και καλύτερη λειτουργία από χιλιάδες χρήστες η ακόμη και εκατοντάδες χιλιάδες την ίδια στιγμή. Ένας SQL Server είναι δυνατόν να πραγματοποιήσει μέχρι και 1.000.000 transactions το δευτερόλεπτο με το ανάλογο υλικό. Ο SQL Server λειτουργεί σε 32 η 64 bit και υποστηρίζει μέχρι και 256 πυρήνες (CPU) Τυπική χρήση είναι οι 8-16 επεξεργαστές.

6.4. Ανάκτηση δεδομένων.

Η ερώτηση (query) είναι ο κύριος τρόπος για την ανάκτηση στοιχείων από μια βάση δεδομένων. Η ερώτηση εκφράζεται χρησιμοποιώντας μια παραλλαγή της αποκαλούμενου SQL. T-SQL, είναι μια διάλεκτος SQL που αναπτύχθηκε από την Microsoft και Sybase. Η T-SQL είναι πολύ κοντά στα ANSI standards που έχουν καθιερωθεί διεθνώς, σε αντιδιαστολή με άλλες διαλέκτους όπως η PL-SQL της Oracle που διαφέρουν περισσότερο από το ANSI standards. Η ερώτηση διευκρινίζει επεξηγηματικά αυτό που είναι να ανακτηθεί από την βάση δεδομένων. Η ερώτηση υποβάλλεται σε επεξεργασία από τον επεξεργαστή ερώτησης, ο οποίος υπολογίζει τη σειρά των βημάτων που θα είναι απαραίτητα για να ανακτηθούν τα ζητούμενα στοιχεία. Η ακολουθία ενεργειών απαραίτητων να εκτελέσουν μια ερώτηση καλείται σχέδιο ερώτησης.

Υπάρχουν πολλοί τρόποι για να υποβληθεί σε επεξεργασία η ίδια ερώτηση. Π.χ., μια ερώτηση που ενώνει 2 πίνακες A & B μαζί θα

μπορούσε να γίνει ή πηγαίνοντας πρώτα στο A βρίσκοντας τα στοιχεία και μετά ενώνοντας με τα στοιχεία του πίνακα B, ή πρώτα από το B και μετά πίνακα A. Ο SQL Server και όχι ο Προγραμματιστής παίρνει την απόφαση μέσω στατιστικών στοιχείων και άλλων στοιχείων που διαθέτει για τους 2 πίνακες να ακολουθήσει την διαδρομή A-B γιατί αυτός είναι ο γρηγορότερος τρόπος. Στις procedural γλώσσες προγραμματισμού πχ C, Pascal, .NET, ο προγραμματιστής θα πρέπει να κάνει την απόφαση το οποίο πολλές φορές σημαίνει λάθος απόφαση. Αυτό καλείται βελτιστοποίηση ερώτησης και εκτελείται από τον επεξεργαστή ερώτησης. Ο SQL παίρνει πολύ περίπλοκες αποφάσεις βασισμένος σε αλγόριθμους που έχουν αναπτυχθεί τα τελευταία 20-30 χρόνια πάνω στο αντικείμενο. Μια παρόμοια ερώτηση σε μια προηγούμενη έκδοση του SQL Server θα μπορούσε να πάρει 10-20 φορές περισσότερο χρόνο λόγω του διαφορετικού αλγόριθμου. Γενικά κάθε καινούργια έκδοση βελτιώνει την αποτελεσματικότητα, και άρα τον χρόνο εκτέλεσης. Εδώ ακριβώς είναι ένα από τα πλεονεκτήματα προγραμματισμού ενός προβλήματος σε μια βάση δεδομένων σε σύγκριση με C, Pascal, C++ κτλ. Βέβαια όλα τα προβλήματα δεν είναι κατάλληλα για εφαρμογή σε βάση δεδομένων.

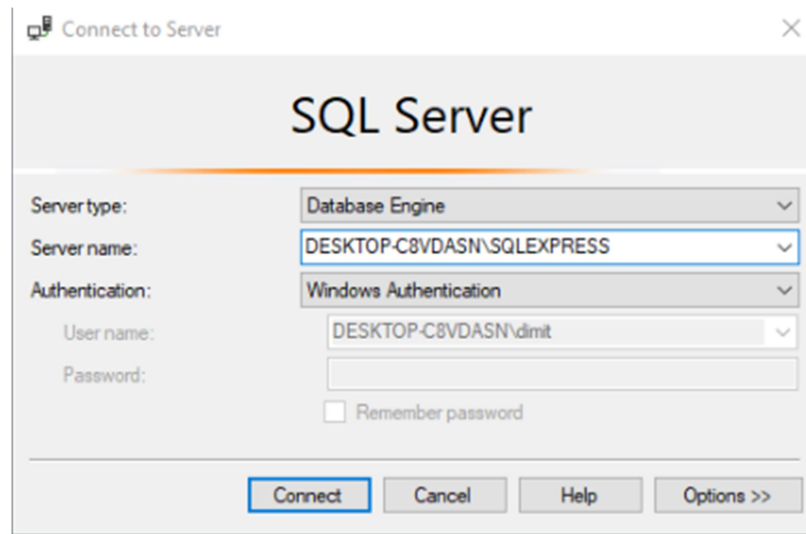
Ο SQL Server επιτρέπει επίσης αποθηκευμένες ερωτήσεις (stored procedures). Οι αποθηκευμένες ερωτήσεις είναι ερωτήσεις με παραμέτρους T-SQL, οι οποίες αποθηκεύονται στον SQL Server. Μια stored procedure τυπικά είναι από 100-1000 γραμμές κώδικα SQL. Οι αποθηκευμένες ερωτήσεις μπορούν να δεχτούν τιμές που στέλνονται από τον πελάτη ως παράμετροι εισαγωγής, και να στείλουν τα αποτελέσματα ως παραμέτρους παραγωγής. Μπορούν να καλέσουν functions, και άλλες stored procedures. Οι stored procedures είναι γρηγορότερες από κατασκευή σε σχέση με απλές ακολουθίες SQL.

6.5. Microsoft SQL Server Management Studio.

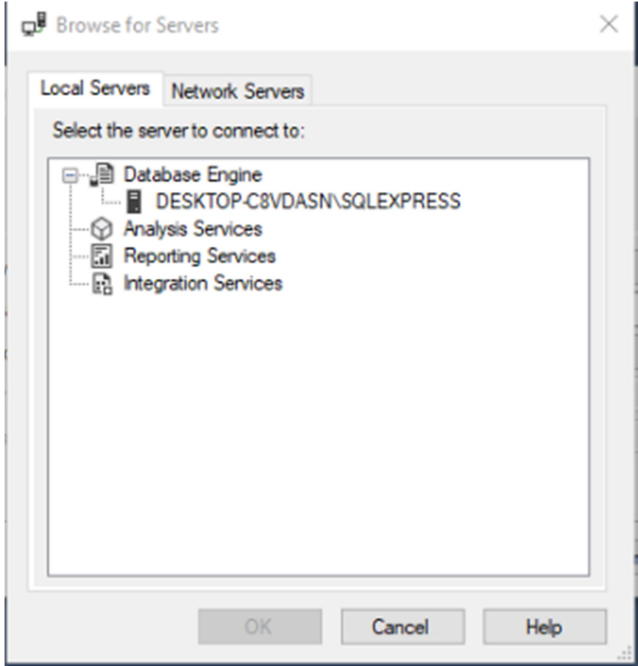
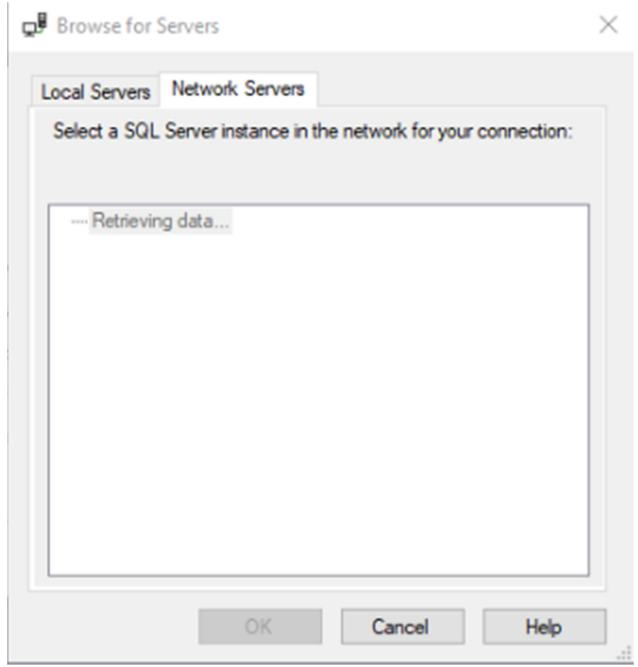
Στην παρούσα παράγραφο παρουσιάζεται ο τρόπος σύνδεσης στον SQL Server, και ο τρόπος δημιουργίας νέας βάσης δεδομένων και η δημιουργία πινάκων στην δημιουργηθήσα βάση. Η διαδικασία που περιγράφεται αφορά το περιβάλλον του **Microsoft SQL Server Management Studio**.

6.5.1. Σύνδεση στον SQL Server.

Κατά την εκκίνηση εμφανίζεται το παράθυρο σύνδεσης:



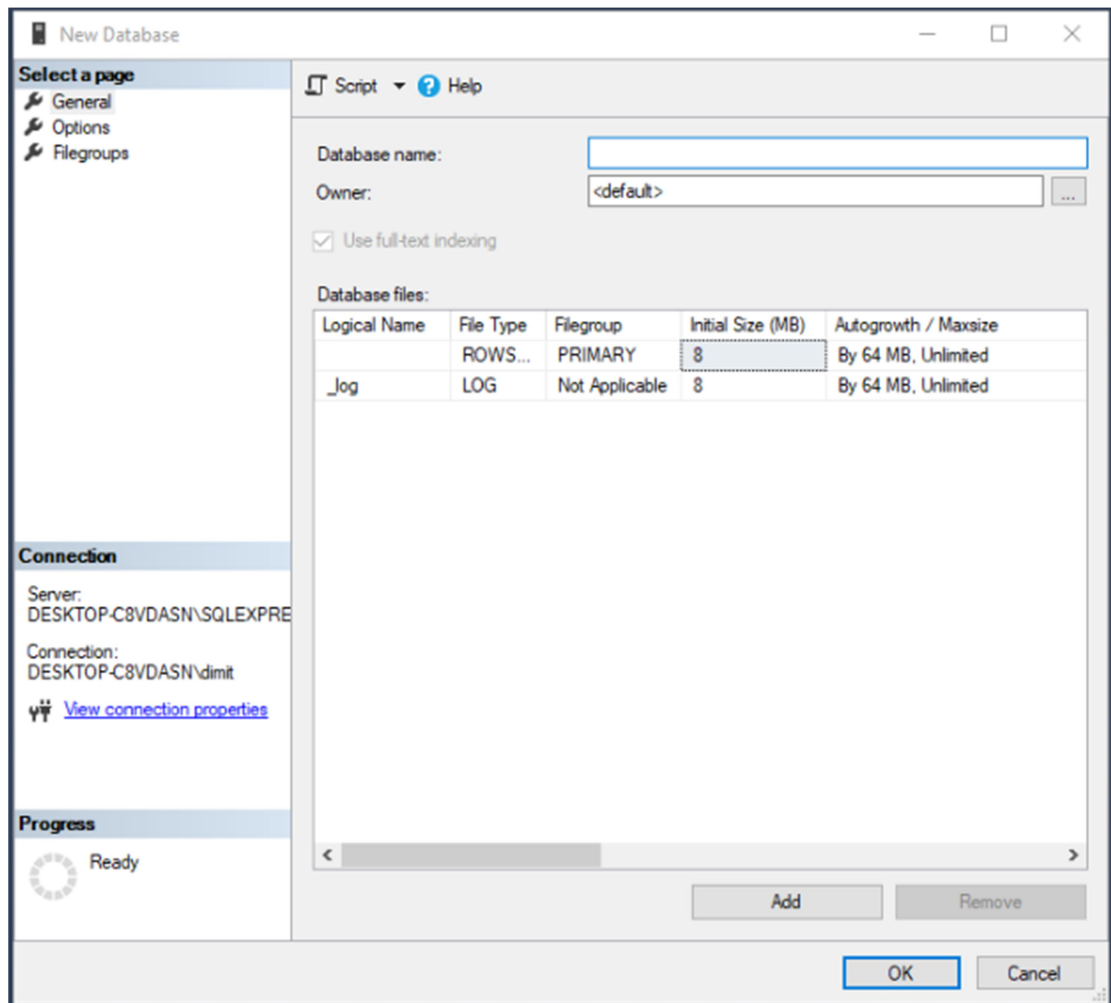
Στοιχείο παραθύρου	Επιλογές
<p>Server type</p>	<p>Επιλογή του τύπου server:</p> <ul style="list-style-type: none"> • DataBase Server. – Σύνδεση σε server με βάση δεδομένων. • Analysis Services. – Σύνδεση σε server που παρέχει δεδομένα με την μορφή διαγραμμάτων, αναφορών, και διάφορα άλλα εργαλεία απεικόνισης δεδομένων. • Reporting Services. – Σύνδεση σε server διαχείρισης και απόδοσης δεδομένων, σε χρήστες, μέσω, κινητού τηλεφώνου (mobile reports), διαδικτυακά (web portal) κ.α. • Intergation Services. Σύνδεση σε server παροχής δεδομένων για περαιτέρω επεξεργασία.

Στοιχείο παραθύρου	Επιλογές
<p>Server name</p>	<p>Επιλογή του υπολογιστή και του server στον οποίο θα γίνει η σύνδεση.</p> <p>Επιλογή του τύπου server που είναι εγκατεστημένοι στο τοπικό υπολογιστή:</p>  <p>Επιλογή υπολογιστή από το τοπικό δίκτυο:</p> 

Στοιχείο παραθύρου	Επιλογές
Authentication	Επιλογή μεθόδου ταυτοποίησης χρήστη: <ul style="list-style-type: none"> • Windows Authentication. – Ταυτοποίηση με βάση τον χρήστη των Windows • SQL Server Authentication. – Ταυτοποίηση με βάση τους χρήστες του SQL Server. Ενεργοποιούνται: <ul style="list-style-type: none"> ○ User name. – Εισαγωγή ονόματος χρήστη. ○ Password. – Εισαγωγή του κωδικού πρόσβασης που αντιστοιχεί στο όνομα χρήστη. ○ Remember password. – Εάν επιλεγεί η ετικέτα , τα αποθηκεύονται τα στοιχεία User name και Password.

6.5.2. Δημιουργία βάσης.

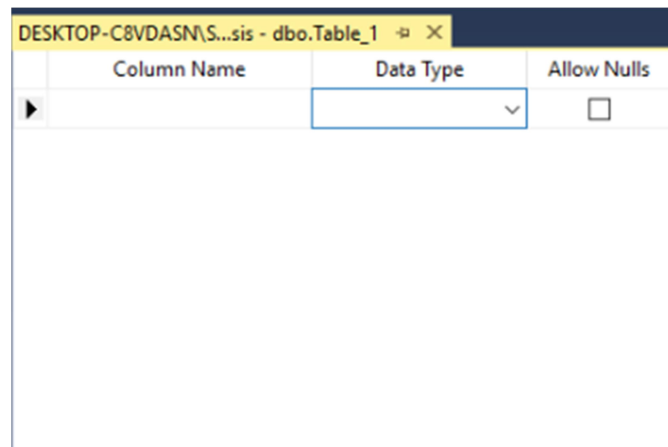
Στον **object explorer**, πατώντας με το δεξί πλήκτρο του ποντικού στον φάκελο **Database**, εμφανίζεται αναδυόμενο μενού. Σε αυτό επιλέγετε **New Database...** , εμφανίζεται το παράθυρο:



Στις περισσότερες των περιπτώσεων συνιστάται να εισάγεται μόνο το όνομα της νέας βάσης, αφήνοντας τις υπόλοιπες επιλογές στις προκαθορισμένες ρυθμίσεις. Εάν χρειαστεί οι επιλογές μπορούν να αλλάξουν σε μεταγενέστερο χρόνο.

6.5.3. Δημιουργία πίνακα σε βάση δεδομένων.

Στον **object explorer**, αν επεκταθεί του εικονίδιο της βάσης δεδομένων (με όνομα αυτό που καταχωρήθηκε στο προηγούμενο παράθυρο), εμφανίζεται ο φάκελος **Tables**, πατώντας με το δεξί πλήκτρο του ποντικού εμφανίζεται αναδυόμενο μενού. Σε αυτό επιλέγεται **Table...**. Εμφανίζεται η καρτέλα:



The image shows a screenshot of a table design tool window titled "DESKTOP-C8VDASN\S...sis - dbo.Table_1". The table has three columns: "Column Name", "Data Type", and "Allow Nulls". The "Data Type" column contains a dropdown menu with a downward arrow. The "Allow Nulls" column contains an unchecked checkbox. The table is currently empty of data rows.

Column Name	Data Type	Allow Nulls
		<input type="checkbox"/>

Σε αυτήν την καρτέλα προσδιορίζονται οι στήλες του πίνακα:

- Το όνομα (**Column Name**)
- ο τύπος των δεδομένων της στήλης (**Data Type**)
- αν μπορεί ή όχι η στήλη να είναι κενή (**Allow Nulls**).

7. Προσδιορισμός βασικών εντολών που χρησιμοποιούνται στην εφαρμογή.

Σε αυτό το κεφάλαιο παρουσιάζονται τα αντικείμενα, οι ιδιότητες, τα συμβάντα των αντικειμένων και οι βασικές εντολές που χρησιμοποιούνται για την ανάπτυξη της εφαρμογής.

7.1. C#.

7.1.1. if statments.

Καθορίζει ποιός βρόγχος κώδικα θα εκτελεστεί με βάση μια λογική πράξη (True/false).

Παράδειγμα απλού ελέγχου, στο οποίο αν δεν ισχύει η συνθήκη παραλείπεται ο κώδικας του αποτελέσματος True:

```
if (condition)
{
    true-statement;
}
```

Παράδειγμα ελέγχου, στο οποίο αν δεν ισχύει η συνθήκη, εκτελείται ο κώδικας του αποτελέσματος False:

```
if (condition)
{
    true-statement;
}
else
{
    false-statement;
}
```

Παράδειγμα διπλού έλεγχου, για δύο συνθήκες με βρόγχο έλεγχου που εκτελείται αν και οι δύο συνθήκες έχουν αποτέλεσμα false:

```
if (Condition1)
{
    // Condition1 is true.
}
else if (Condition2)
{
    // Condition1 is false and Condition2 is true.
}
else
{
    // Condition1 and Condition2 are false.
}
```

7.1.2. try.

Έλεγχος αν ο βρόγχος του κώδικα μπορεί να εκτελεστεί.

```
try
{
    // Code with possible Error or inconsistencies
}
```

Σε πολλές περιπτώσεις αν δεν μπορεί να εκτελεστεί ο κώδικας, το πρόγραμμα πρέπει να προβλέπει αυτήν την πιθανότητα, για αυτό, συνήθως η δήλωση try ακολουθείτε:

```
try
{
    // Code with possible Error or inconsistencies
}
catch (ArgumentException e)
{
    //Handlein error or exception
}
```

7.1.3. Class.

Με τις κλάσεις δημιουργούνται βρόγχοι κώδικα, που εκτελούν σύνθετες λειτουργίες του προγράμματος και μπορούν να χρησιμοποιηθούν σε πολλά σημεία της εφαρμογής.

Η κλάσεις δηλώνονται ως εξής:

```
public class Customer
{
    // Fields, properties, methods and events go here...
}
```

Οι κλάσεις δεν είναι αντικείμενα, επομένως, όταν χρησιμοποιηθούν πρέπει να δηλωθούν ως εξής:

```
Customer object1 = new Customer();
```

Μια κλάση μπορεί να βασίζεται σε μία άλλη κλάση, ή και να είναι βάση μιας τρίτης κλάσης. Αυτή η ιδιότητα ονομάζεται **κληρονομικότητα**. Η δήλωση μίας κλάσης που κληρονομεί ιδιότητες από μια άλλη κλάση δηλώνεται ως εξής:

```
public class Manager : Employee
{
    // Employee fields, properties, methods and events are inherited
    // New Manager fields, properties, methods and events go here...
}
```

Παράδειγμα δήλωσης και χρήσης κλάσεων.

```
using System;

public class Person
{
    // Constructor that takes no arguments:
    public Person()
    {
        Name = "unknown";
    }

    // Constructor that takes one argument:
    public Person(string name)
    {
        Name = name;
    }
}
```

```
// Auto-implemented readonly property:
public string Name { get; }

// Method that overrides the base class (System.Object)
implementation.
public override string ToString()
{
    return Name;
}
}
class TestPerson
{
    static void Main()
    {
        // Call the constructor that has no parameters.
        var person1 = new Person();
        Console.WriteLine(person1.Name);

        // Call the constructor that has one parameter.
        var person2 = new Person("Sarah Jones");
        Console.WriteLine(person2.Name);
        // Get the string representation of the person2 instance.
        Console.WriteLine(person2);

        Console.WriteLine("Press any key to exit.");
        Console.ReadKey();
    }
}

// Output:
// unknown
// Sarah Jones
// Sarah Jones
```

7.1.4. for loop.

Με την χρήση του for εκτελείται ένας βρόγχος κώδικα για το διάστημα που ισχύει μία λογική συνθήκη. Δηλώνεται ως εξής:

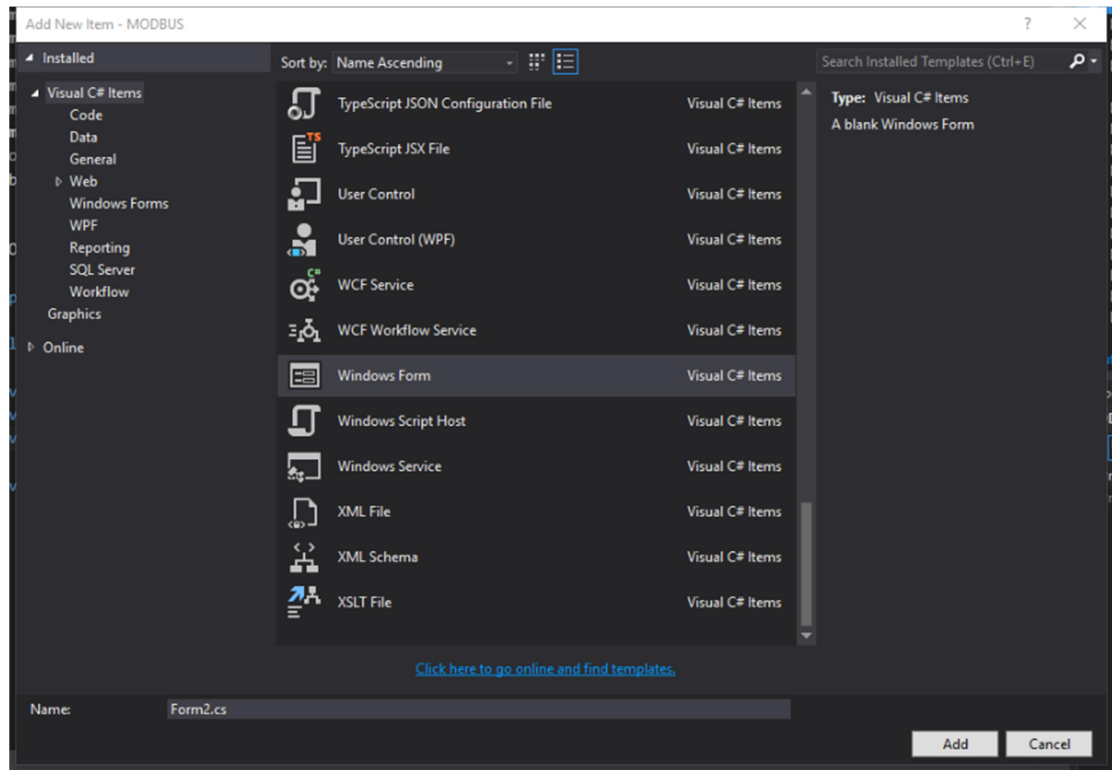
```
for (initializer; condition; iterator)
    body
```

7.2. Visual C#.

7.2.1. Visual components - εμφανιζόμενα αντικείμενα.

7.2.1.1. Windows form.

Αποτελεί το παράθυρο των windows, τα παράθυρα τοποθετούνται εντός του **project**. Δημιουργείται από παράθυρο:



Σε αυτό το παράθυρο ορίζεται το όνομα της φόρμας.

Ιδιότητα		Επεξήγηση
BackColor		Επιλογή του χρώματος του παραθύρου.
FormBorderStyle		Επιλογή εμφάνισης του πλαισίου του παραθύρου
MaximizeBox		Επιλογή εμφάνισης κουμπιού μεγιστοποίησης του παραθύρου.
MinimizeBox		Επιλογή εμφάνισης κουμπιού ελαχιστοποίησης του παραθύρου
Size	Width	Καθορισμός του πλάτους του παραθύρου.
	Height	Καθορισμός του ύψους του παραθύρου.
Text		Καθορισμός του τίτλου του παραθύρου.

7.2.1.2. *Label*.

Το αντικείμενο αυτό χρησιμοποιείται για την απεικόνιση κειμένου.

Ιδιότητα		Επεξήγηση
Name		Το όνομα του αντικειμένου.
AutoSize		Αυτόματος καθορισμός του μεγέθους του αντικειμένου με βάση το μέγεθος του κειμένου.
BackColor		Επιλογή του χρώματος του φόντου του αντικειμένου.
Font	Name	Επιλογή της γραμματοσειράς.
	Size	Επιλογή του μεγέθους της γραμματοσειράς.
	Bold	Επιλογή έντονης γραφής.
ForeColor		Επιλογή του χρώματος της γραμματοσειράς.
Location	X	Τοποθέτηση στον οριζόντιο άξονα, με βάση το παράθυρο.
	Y	Τοποθέτηση στον κάθετο άξονα, με βάση το παράθυρο.
Text		Καθορισμός του αναγραφόμενου κειμένου του αντικειμένου.

7.2.1.3. *TextBox*.

Το αντικείμενο αυτό χρησιμοποιείται για την εισαγωγή κειμένου.

Ιδιότητα		Επεξήγηση
Name		Το όνομα του αντικειμένου.
BackColor		Επιλογή του χρώματος του φόντου του αντικειμένου.
BorderStyle		Επιλογή εμφάνισης του πλαισίου του αντικειμένου.
Font	Name	Επιλογή της γραμματοσειράς.
	Size	Επιλογή του μεγέθους της γραμματοσειράς.
ForeColor		Επιλογή του χρώματος της γραμματοσειράς.
Lines	[0]	Το κείμενο του αντικειμένου χωρισμένο με βάση της γραμμές.

Ιδιότητα		Επεξήγηση
Location	X	Τοποθέτηση στον οριζόντιο άξονα, με βάση το παράθυρο.
	Y	Τοποθέτηση στον κάθετο άξονα, με βάση το παράθυρο.
MaxLength		Καθορισμός του μέγιστου αριθμού χαρακτήρων που μπορούν να εισαχθούν στο αντικείμενο.
Size	Width	Καθορισμός του πλάτους του αντικειμένου.
	Height	Καθορισμός του ύψους του αντικειμένου.
Text		Καθορισμός του αναγραφόμενου κειμένου του αντικειμένου.

Event	Κλήση	Επεξήγηση
Enter	<code>private void adressInputTbx_Enter(object sender, EventArgs e)</code>	Εκτέλεση της μεθόδου όταν πατηθεί το πλήκτρο ENTER .
KeyDown	<code>private void adressInputTbx_KeyDown(object sender, KeyEventArgs e)</code>	Εκτέλεση της μεθόδου για κάθε πλήκτρο που είναι πατημένο.
KeyPress	<code>private void adressInputTbx_KeyPress(object sender, KeyPressEventArgs e)</code>	Εκτέλεση της μεθόδου για κάθε πλήκτρο που πατητέ

7.2.1.4. Button.

Το αντικείμενο αυτό χρησιμοποιείται για την εκτέλεση χειρισμών μέσω του ποντικιού.

Ιδιότητα		Επεξήγηση
Name		Το όνομα του αντικειμένου.
BackColor		Επιλογή του χρώματος του φόντου του αντικειμένου.
FlatAppearance	BorderColor	Καθορισμός του χρώματος του πλαισίου του αντικειμένου.

Ιδιότητα		Επεξήγηση
	BorderSize	Καθορισμός του μεγέθους του πλαισίου του αντικειμένου.
	MouseDownBackColor	Επιλογή του χρώματος του φόντου του αντικειμένου, όταν πατητέ με το ποντίκι.
	MouseOverBackColor	Επιλογή του χρώματος του φόντου του αντικειμένου, όταν το ποντίκι, βρίσκεται πάνω από το αντικείμενο.
FlatStyle		Επιλογή του στυλ του αντικειμένου.
Font	Name	Επιλογή της γραμματοσειράς.
	Size	Επιλογή του μεγέθους της γραμματοσειράς.
	Bold	Επιλογή έντονης γραφής.
ForeColor		Επιλογή του χρώματος της γραμματοσειράς.
Location	X	Τοποθέτηση στον οριζόντιο άξονα, με βάση το παράθυρο.
	Y	Τοποθέτηση στον κάθετο άξονα, με βάση το παράθυρο.
Size	Width	Καθορισμός του πλάτους του αντικειμένου.
	Height	Καθορισμός του ύψους του αντικειμένου.
TabIndex		Ορισμός της σειράς που το αντικείμενο εστιάζεται πατώντας το πλήκτρο TAB . Ο αριθμός αυτός είναι μοναδικός για όλα τα αντικείμενα που εμφανίζονται στο παράθυρο.

Ιδιότητα	Επεξήγηση
TabStop	Επιλογή εάν το αντικείμενο μπορεί να εστιάσει πατώντας το πλήκτρο TAB .
Text	Καθορισμός του αναγραφόμενου κειμένου του αντικειμένου.

Event	Κλήση	Επεξήγηση
Click	<code>private void connectBtn_Click(object sender, EventArgs e)</code>	Εκτέλεση αυτής της μεθόδου όταν στο αντικείμενο πατηθεί το ποντίκι.

7.2.1.5. DateTimePicker.

Με αυτό το αντικείμενο επιλέγεται ημερομηνία και ώρα, **DateTime**.

Ιδιότητα	Επεξήγηση	
Name	Τα όνομα του αντικειμένου.	
CustomFormat	Καθορισμός ειδικής μορφοποίηση των τιμών που εισάγονται.	
Font	Name	Επιλογή της γραμματοσειράς.
	Size	Επιλογή του μεγέθους της γραμματοσειράς.
Format	Επιλογή της μορφοποίησης των τιμών που εισάγονται.	
Location	X	Τοποθέτηση στον οριζόντιο άξονα, με βάση το παράθυρο.
	Y	Τοποθέτηση στον κάθετο άξονα, με βάση το παράθυρο.
Size	Width	Καθορισμός του πλάτους του αντικειμένου.
	Height	Καθορισμός του ύψους του αντικειμένου.

7.2.1.6. CheckBox.

Το αντικείμενο αυτό χρησιμοποιείται για να επιλέγει ο χρήστης την τιμή μιας λογικής μεταβλητής.

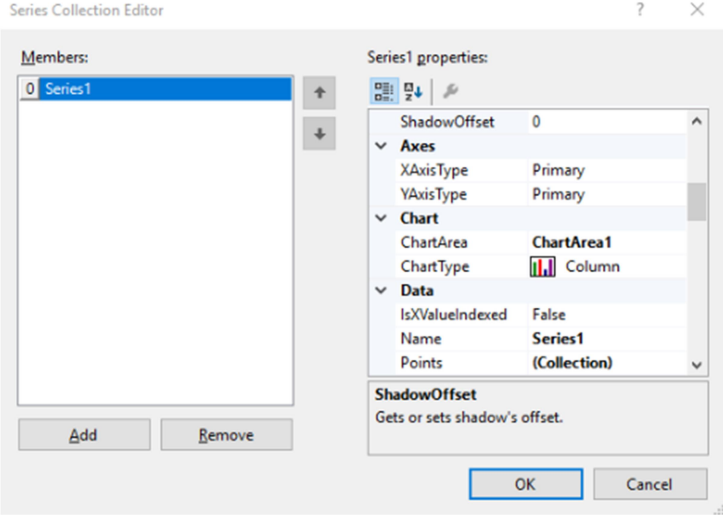
Ιδιότητα		Επεξήγηση
Name		Τα όνομα του αντικειμένου.
AutoSize		Αυτόματος καθορισμός του μεγέθους του αντικειμένου με βάση το μέγεθος του κειμένου.
BackColor		Επιλογή του χρώματος του φόντου του αντικειμένου.
Font	Name	Επιλογή της γραμματοσειράς.
	Size	Επιλογή του μεγέθους της γραμματοσειράς.
ForeColor		Επιλογή του χρώματος της γραμματοσειράς.
Location	X	Τοποθέτηση στον οριζόντιο άξονα, με βάση το παράθυρο.
	Y	Τοποθέτηση στον κάθετο άξονα, με βάση το παράθυρο.
Text		Καθορισμός του αναγραφόμενου κειμένου του αντικειμένου.

Event	Κλήση	Επεξήγηση
CheckedChanged	<pre>private void checkBoxSelection_CheckedChanged(object sender, EventArgs e)</pre>	Εκτέλεση αυτής της μεθόδου όταν η ετικέτα είτε επιλεγεί, είτε αποεπιλεγεί.

7.2.1.7. Chart.

Αυτό το αντικείμενο χρησιμοποιείται για απεικόνιση δεδομένων σε μορφή γραφήματος.

Ιδιότητα		Επεξήγηση
Name		Το όνομα του αντικειμένου.
BackColor		Επιλογή του χρώματος του φόντου του αντικειμένου.
Location	X	Τοποθέτηση στον οριζόντιο άξονα, με βάση το παράθυρο.
	Y	Τοποθέτηση στον κάθετο άξονα, με βάση το παράθυρο.

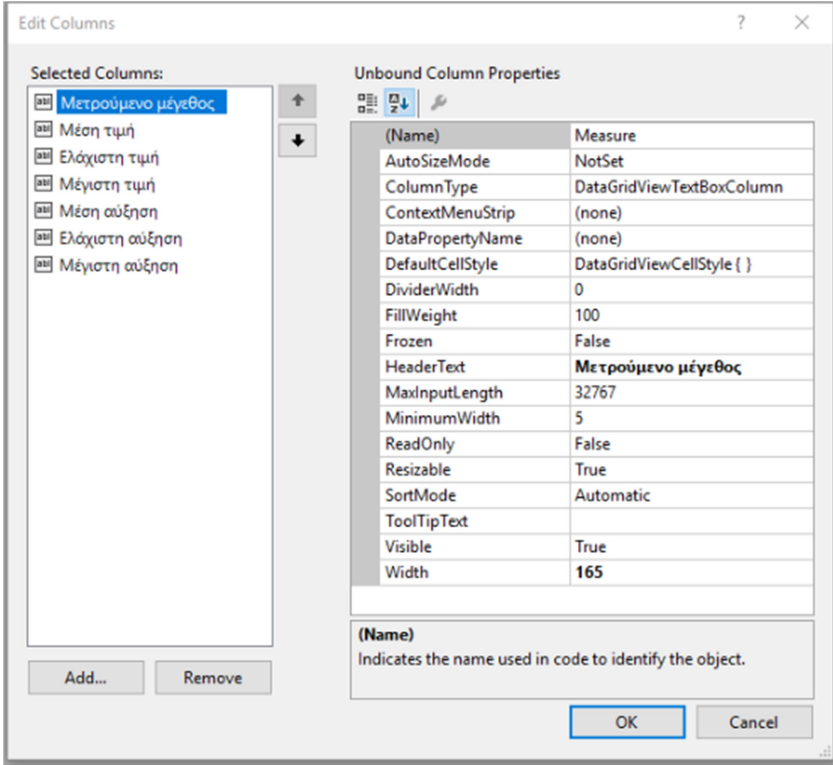
Ιδιότητα	Επεξήγηση	
Series	<p>Καθορισμός των στοιχείων (σειρών –Series) του γραφήματος. Εμφανίζει το παράθυρο:</p> 	
	Ιδιότητα	Επεξήγηση
	BorderWidth	Ορισμός του πάχους της σειράς στο γράφημα.
	Name	Το όνομα της σειράς.
ChartType	<p>Επιλογή του τύπου της σειράς. Οι υποστηριζόμενοι τύποι είναι:</p> <ul style="list-style-type: none"> • Point • FastPoint • Bubble • Line • Spline • StepLine • FastLine • Bar • StackedBar • StackedBar100 • Column • StackedColumn • StackedColumn100 • Area • SplineArea • StackedArea • StackedArea100 • Pie • Doughnut 	

Ιδιότητα		Επεξήγηση
		<ul style="list-style-type: none"> • Stock • CandleStic • Range • SplineRange • RangeBar • RangeColumn • Radar • Polar • ErrorBar • BoxPlot • Renko • ThreeLineBrake • Kagi • PointAndFigure • Funel • Pyramid
Size	Width	Καθορισμός του πλάτους του αντικειμένου.
	Height	Καθορισμός του ύψους του αντικειμένου.

7.2.1.8. *DataGridView*.

Το αντικείμενο αυτό χρησιμοποιείται για την απεικόνιση δεδομένων υπό την μορφή πίνακα.

Ιδιότητα	Επεξήγηση
Name	Το όνομα του αντικειμένου.
BackColor	Επιλογή του χρώματος του φόντου του αντικειμένου.

Ιδιότητα		Επεξήγηση						
Column		<p>Καθορισμός των στοιχείων του πίνακα (Column – Στήλες). Εμφανίζει το παράθυρο:</p> 						
		<table border="1"> <thead> <tr> <th>Ιδιότητα</th> <th>Επεξήγηση</th> </tr> </thead> <tbody> <tr> <td>Name</td> <td>Το όνομα της στήλης.</td> </tr> <tr> <td>HeaderText</td> <td>Καθορισμός του κειμένου του τίτλου της στήλης</td> </tr> </tbody> </table>	Ιδιότητα	Επεξήγηση	Name	Το όνομα της στήλης.	HeaderText	Καθορισμός του κειμένου του τίτλου της στήλης
	Ιδιότητα	Επεξήγηση						
Name	Το όνομα της στήλης.							
HeaderText	Καθορισμός του κειμένου του τίτλου της στήλης							
Location	<p>X Τοποθέτηση στον οριζόντιο άξονα, με βάση το παράθυρο.</p> <p>Y Τοποθέτηση στον κάθετο άξονα, με βάση το παράθυρο.</p>							
Size	Width	Καθορισμός του πλάτους του αντικειμένου.						
	Height	Καθορισμός του ύψους του αντικειμένου.						

Event	Κλήση	Επεξήγηση
DataBindingCompleted	<code>private void dataDGV_DataBindingComplete(object sender, DataGridViewBindingCompleteEventArgs e)</code>	Εκτέλεση αυτής της μεθόδου όταν ολοκληρωθεί η σύνδεση με την πηγή δεδομένων.

7.2.2. Non visual Components – μη εμφανιζόμενα αντικείμενα.

7.2.2.1 Timer.

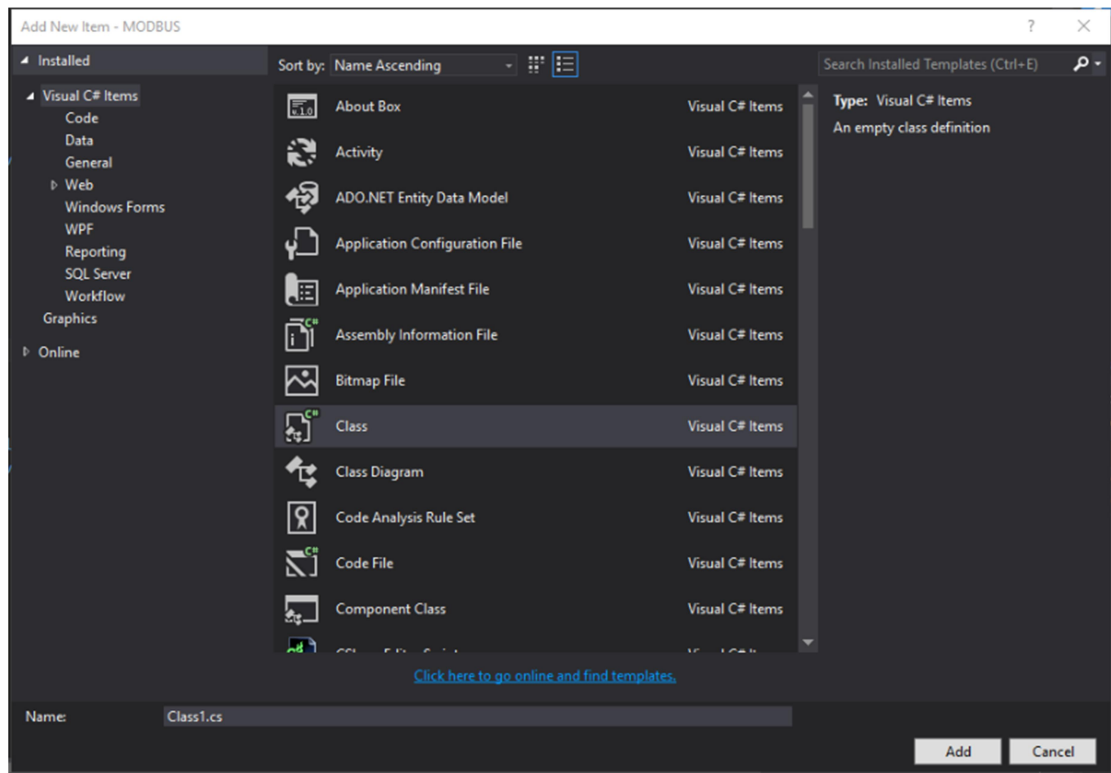
Το αντικείμενο αυτό χρησιμοποιείτε για την εκτέλεση κώδικα σε συγκεκριμένο χρόνο.

Ιδιότητα	Επεξήγηση
Name	Το όνομα του αντικειμένου.
Enabled	Καθορισμός αν ο timer ξεκινάει να μετράει χρόνο αυτόματα.
Interval	Ορισμός του κύκλου του χρονικού (σε msec).

Event	Κλήση	Επεξήγηση
Tick	<code>private void checkInstrumentTmr_Tick(object sender, EventArgs e)</code>	Εκτέλεση αυτής της μεθόδου κάθε φορά που ολοκληρώνεται ο κύκλος του χρονικού.

7.2.2.2 Class.

Το αντικείμενο αυτό χρησιμοποιείται για την δημιουργία κλάσεων, εντός του project, ή εντός του solution, ή εξωτερικά. Δημιουργείται από το παράθυρο:

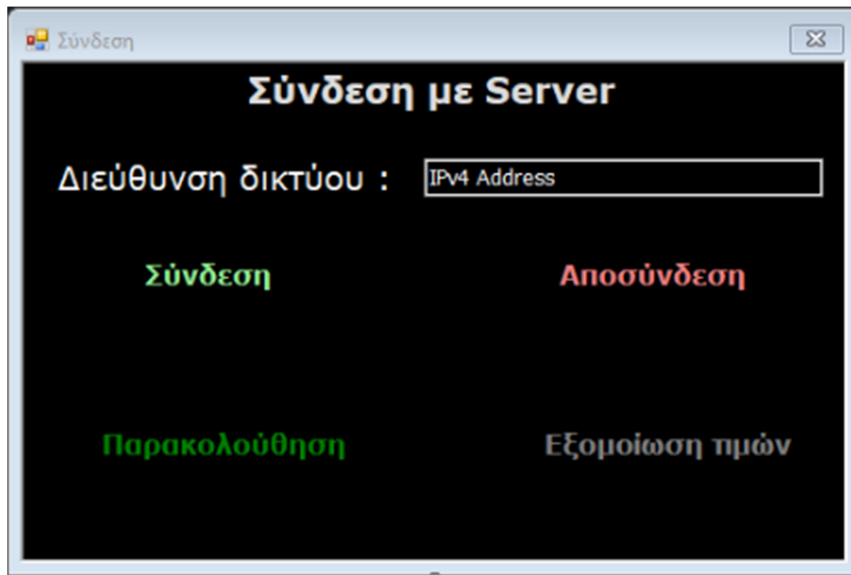


Σε αυτό το παράθυρο προσδιορίζεται το όνομα της κλάσης.

8. Δημιουργία εφαρμογής.

8.1. Form1.

8.1.1. Form1.cs[Design].



Στοιχείο παραθύρου	Λειτουργία
Διεύθυνση δικτύου	Σε αυτό το πεδίο πληκτρολογείται η διεύθυνση δικτύου του μετρητή ενέργειας. Το πρωτόκολλο επικοινωνίας είναι το IPv4. Για την παρούσα εφαρμογή πληκτρολογείται η τοπική IPv4, όπου τρέχει ο Modbus Server εξομοίωσης. Η διαδικασία με την οποία βρίσκουμε την σωστή διεύθυνση έχει περιγραφεί ανωτέρω (5.1.1. EasyModbus.dll.)
Σύνδεση	Σύνδεση με τον Server, την διεύθυνση του οποίου έχουμε ορίσει ανωτέρω.
Αποσύνδεση	Αποσύνδεση από τον Server.
Παρακολούθηση	Εμφανίζει το παράθυρο Janitza_UMG96RM.
Εξομοίωση τιμών	Εμφανίζει το παράθυρο Εξομοίωση τιμών.

8.1.2.Form1.cs.

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;
using EasyModbus;

namespace MODBUS
{
    public partial class Form1 : Form
    {
        public Form1()
        {
            InitializeComponent();
        }

        public static int numOfRecs = 1;

        public string ip;

        private void adressInputTbx_Enter(object sender, EventArgs e)
        {
            adressInputTbx.Text = "";
        }

        private void adressInputTbx_KeyDown(object sender, KeyEventArgs e)
        {
            if (e.KeyCode == Keys.Enter)
            {
                string newInput;
                newInput = adressInputTbx.Text.Replace(",", ".");

                if (newInput.Length > 10)
                {
                    ip = newInput;
                    connectBtn.Enabled = true;
                    connectBtn.Text = "Connect to server : "+newInput;
                }
                else
                    adressInputTbx.Text = "";
            }
        }

        private void adressInputTbx_KeyPress(object sender, KeyPressEventArgs e)
        {
            char ck = e.KeyChar;

            if (Char.IsLetter(ck))
            {
                e.Handled = true;
            }
        }
    }
}
```

```

    }

    public static ModbusClient mBC = new ModbusClient();
    private void connectBtn_Click(object sender, EventArgs e)
    {
        try
        {
            mBC.Connect(addressInputTbx.Text, 502);
        }
        catch
        {
            MessageBox.Show("ModBusTCP Client not connected to : " +
mBC.IPAddress + " Check if server is running");
        }

        if (mBC.Connected)
        {
            connectBtn.Enabled = false;
            disconnectBtn.Enabled = true;
            watchInstrumentBtn.Enabled = true;
            setValuesBtn.Enabled = true;
        }
    }

    private void disconnectBtn_Click(object sender, EventArgs e)
    {
        if (mBC.Connected)
            mBC.Disconnect();
        else
            disconnectBtn.Enabled = false;
    }

    private void setValuesBtn_Click(object sender, EventArgs e)
    {
        SetValues newSV = new SetValues();
        newSV.Show();
    }

    private void watchInstrumentBtn_Click(object sender, EventArgs e)
    {
        Janitza_UMG96RM newJanitza = new Janitza_UMG96RM();
        newJanitza.Show();
    }
}
}

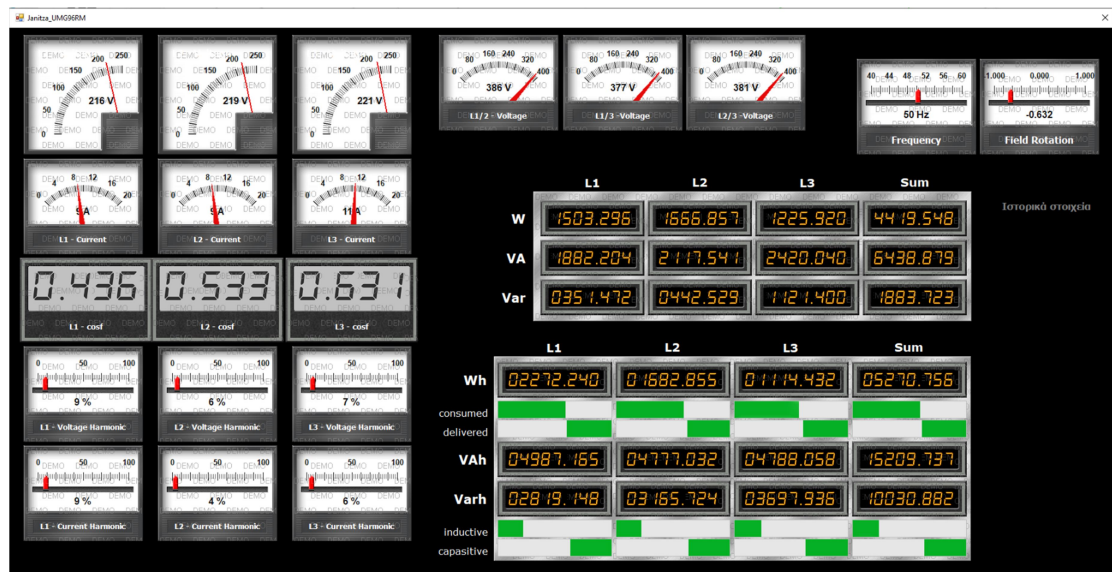
```

Μεταβλητές - Μέθοδοι – Events	Περιγραφή
<code>public Form1()</code>	Αρχικοποίηση παραθύρου.
<code>public string ip;</code>	Η διεύθυνση που πληκτρολογείτε στο πεδίο Διεύθυνση δικτύου
<code>private void addressInputTbx_Enter(object sender, EventArgs e)</code>	Όταν το πεδίο Διεύθυνση δικτύου επιλεγεί τότε διαγράφεται το επεξηγηματικό κείμενο IPv4 Address και ο χρήστης πληκτρολογεί την διεύθυνση

Μεταβλητές - Μέθοδοι – Events	Περιγραφή
<pre>private void addressInputTbx_KeyDown(object sender, KeyEventArgs e)</pre>	Έλεγχος αν τα πλήκτρα που πατιούνται στο πεδίο Διεύθυνση δικτύου είναι αριθμοί
<pre>private void addressInputTbx_KeyPress(object sender, KeyPressEventArgs e)</pre>	Έλεγχος αν το μήκος των χαρακτήρων αντιστοιχεί σε διεύθυνση IP. Εμφάνιση του κουμπιού Σύνδεση
<pre>public static ModbusClient mBC = new ModbusClient();</pre>	Καθολική στατική μεταβλητή που ορίζεται στην βιβλιοθήκη Easy Modbus .
<pre>private void connectBtn_Click(object sender, EventArgs e)</pre>	Σύνδεση, μέσω της μεταβλητής mBC , με τον Modbus Server. Εμφανίζει τα κουμπιά: <ul style="list-style-type: none"> • Αποσύνδεση. • Παρακολούθηση. • Εξομοίωση τιμών.
<pre>private void disconnectBtn_Click(object sender, EventArgs e)</pre>	Αποσύνδεση από συνδεδεμένο Server, μέσω της μεταβλητής mBC .
<pre>private void watchInstrumentBtn_Click(object sender, EventArgs e)</pre>	Εμφανίζει το παράθυρο Janitza_UMG96RM .
<pre>private void setValuesBtn_Click(object sender, EventArgs e)</pre>	Εμφανίζει το παράθυρο Εξομοίωση τιμών .

8.2. Παρακολούθηση τιμών.

8.2.1. Janitza_UMG96RM.cs[Design].



Σε αυτό παράθυρο εμφανίζονται όλα τα μεγέθη του μετρητή ενέργειας. Το κουμπι **Ιστορικά στοιχεία** εμφανίζει το παράθυρο **Εύρεση μεγεθών από την βάση δεδομένων**.

8.2.2. Janitza_UMG96RM.cs.

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;
using EasyModbus;
using exModbus;
using System.Data.SqlClient;

namespace MODBUS
{
    public partial class Janitza_UMG96RM : Form
    {
        public Janitza_UMG96RM()
        {
            InitializeComponent();
        }

        private simReadWriteFloat inst = new simReadWriteFloat();
        public List<float> V1 = new List<float>();
    }
}
```

```
public List<float> V2 = new List<float>();
public List<float> V3 = new List<float>();
public List<float> V12 = new List<float>();
public List<float> V13 = new List<float>();
public List<float> V23 = new List<float>();
public List<float> I1 = new List<float>();
public List<float> I2 = new List<float>();
public List<float> I3 = new List<float>();
public List<float> Isum = new List<float>();
public List<float> cosf1 = new List<float>();
public List<float> cosf2 = new List<float>();
public List<float> cosf3 = new List<float>();
public List<float> frequency = new List<float>();
public List<float> fieldRotation = new List<float>();
public List<float> P1 = new List<float>();
public List<float> P2 = new List<float>();
public List<float> P3 = new List<float>();
public List<float> Psum = new List<float>();
public List<float> A1 = new List<float>();
public List<float> A2 = new List<float>();
public List<float> A3 = new List<float>();
public List<float> Asum = new List<float>();
public List<float> R1 = new List<float>();
public List<float> R2 = new List<float>();
public List<float> R3 = new List<float>();
public List<float> Rsum = new List<float>();
public List<float> RE1 = new List<float>();
public List<float> RE2 = new List<float>();
public List<float> RE3 = new List<float>();
public List<float> REsum = new List<float>();
public List<float> RECons1 = new List<float>();
public List<float> RECons2 = new List<float>();
public List<float> RECons3 = new List<float>();
public List<float> REConssum = new List<float>();
public List<float> REDe11 = new List<float>();
public List<float> REDe12 = new List<float>();
public List<float> REDe13 = new List<float>();
public List<float> REDelsum = new List<float>();
public List<float> AE1 = new List<float>();
public List<float> AE2 = new List<float>();
public List<float> AE3 = new List<float>();
public List<float> AEsun = new List<float>();
public List<float> RaE1 = new List<float>();
public List<float> RaE2 = new List<float>();
public List<float> RaE3 = new List<float>();
public List<float> RaEsum = new List<float>();
public List<float> RaEInd1 = new List<float>();
public List<float> RaEInd2 = new List<float>();
public List<float> RaEInd3 = new List<float>();
public List<float> RaEIndsum = new List<float>();
public List<float> RaECap1 = new List<float>();
public List<float> RaECap2 = new List<float>();
public List<float> RaECap3 = new List<float>();
public List<float> RaECapsum = new List<float>();
public List<float> HV1 = new List<float>();
public List<float> HV2 = new List<float>();
public List<float> HV3 = new List<float>();
public List<float> HI1 = new List<float>();
public List<float> HI2 = new List<float>();
public List<float> HI3 = new List<float>();
private Statistics stat = new Statistics();
private SQLInterface iSQL = new SQLInterface();
```

```

public static SqlConnection SQLCon;
private void aquisitionTmr_Tick(object sender, EventArgs e)
{
    if (SQLCon == null)
    {
        SQLCon = new SqlConnection("Data Source=DESKTOP-
C8VDASN\\SQLEXPRESS;Initial Catalog=thesis;Integrated Security=True");
    }

    DateTime nowDT = DateTime.Now;

    //mean Values

//V1,V2,V3,V12,V13,V23,I1,I2,I3,Isum,cosf1,cosf2,cosf3,frequency,fieldRotation,P1,P2
,P3,Psum,A1,A2,A3,Asum,R1,R2,R3,
    //Rsum,HV1, HV2, HV3, HI1, HI2, HI3

    float V1Value, V2Value, V3Value, V12Value, V13Value, V23Value, I1Value,
I2Value, I3Value, IsumValue, cosf1Value,
        cosf2Value, cosf3Value, frequencyValue, fieldRotationValue,
P1Value, P2Value, P3Value, PsumValue, A1Value, A2Value,
        A3Value, AsumValue, R1Value, R2Value, R3Value, RsumValue,
HV1Value, HV2Value, HV3Value, HI1Value, HI2Value, HI3Value;
    V1Value = stat.GetMeanValue(V1);
    V2Value = stat.GetMeanValue(V2);
    V3Value = stat.GetMeanValue(V3);
    V12Value = stat.GetMeanValue(V12);
    V13Value = stat.GetMeanValue(V13);
    V23Value = stat.GetMeanValue(V23);
    I1Value = stat.GetMeanValue(I1);
    I2Value = stat.GetMeanValue(I2);
    I3Value = stat.GetMeanValue(I3);
    IsumValue = stat.GetMeanValue(Isum);
    cosf1Value = stat.GetMeanValue(cosf1);
    cosf2Value = stat.GetMeanValue(cosf2);
    cosf3Value = stat.GetMeanValue(cosf3);
    frequencyValue = stat.GetMeanValue(frequency);
    fieldRotationValue = stat.GetMeanValue(fieldRotation);
    P1Value = stat.GetMeanValue(P1);
    P2Value = stat.GetMeanValue(P2);
    P3Value = stat.GetMeanValue(P3);
    PsumValue = stat.GetMeanValue(Psum);
    A1Value = stat.GetMeanValue(A1);
    A2Value = stat.GetMeanValue(A2);
    A3Value = stat.GetMeanValue(A3);
    AsumValue = stat.GetMeanValue(Asum);
    R1Value = stat.GetMeanValue(R1);
    R2Value = stat.GetMeanValue(R2);
    R3Value = stat.GetMeanValue(R3);
    RsumValue = stat.GetMeanValue(Rsum);
    HV1Value = stat.GetMeanValue(HV1);
    HV2Value = stat.GetMeanValue(HV2);
    HV3Value = stat.GetMeanValue(HV3);
    HI1Value = stat.GetMeanValue(HI1);
    HI2Value = stat.GetMeanValue(HI2);
    HI3Value = stat.GetMeanValue(HI3);
    //actual values
    //RE1,RE2,RE3,REsum, RECons1, RECons2, RECons3, REConssum, REDe11,
REDe12, REDe13, REDe1sum, AE1, AE2, AE3, AEsuM,RaE1,
    //RaE2, RaE3, RaEsum, RaEInd1, RaEInd2, RaEInd3, RaEIndsum, RaECap1,
RaECap2, RaECap3, RaECapsum

```

```

float RE1Value, RE2Value, RE3Value, RSumValue, RECons1Value,
RECons2Value, RECons3Value, REConssumValue, REDel1Value,
REDel2Value, REDel3Value, REDelsumValue, AE1Value, AE2Value,
AE3Value, ASumValue, RaE1Value, RaE2Value, RaE3Value,
RaEsumValue, RaEInd1Value, RaEInd2Value, RaEInd3Value,
RaEIndsumValue, RaECap1Value, RaECap2Value, RaECap3Value,
RaECapsumValue;
RE1Value = RE1.Last();
RE2Value = RE2.Last();
RE3Value = RE3.Last();
RSumValue = RSum.Last();
RECons1Value = RECons1.Last();
RECons2Value = RECons2.Last();
RECons3Value = RECons3.Last();
REConssumValue = REConssum.Last();
REDel1Value = REDel1.Last();
REDel2Value = REDel2.Last();
REDel3Value = REDel3.Last();
REDelsumValue = REDelsum.Last();
AE1Value = AE1.Last();
AE2Value = AE2.Last();
AE3Value = AE3.Last();
ASumValue = ASum.Last();
RaE1Value = RaE1.Last();
RaE2Value = RaE2.Last();
RaE3Value = RaE3.Last();
RaEsumValue = RaEsum.Last();
RaEInd1Value = RaEInd1.Last();
RaEInd2Value = RaEInd2.Last();
RaEInd3Value = RaEInd3.Last();
RaEIndsumValue = RaEIndsum.Last();
RaECap1Value = RaECap1.Last();
RaECap2Value = RaECap2.Last();
RaECap3Value = RaECap3.Last();
RaECapsumValue = RaECapsum.Last();

bool savedData = iSQL.JanitzaUMG96R_SaveToDB(Form1.numOfRecs ,SQLCon,
nowDT, V1Value, V2Value, V3Value, V12Value, V13Value, V23Value, I1Value, I2Value,
I3Value, IsumValue, cosf1Value, cosf2Value, cosf3Value, frequencyValue,
fieldRotationValue, P1Value, P2Value, P3Value, PsumValue, A1Value, A2Value, A3Value,
AsumValue, R1Value, R2Value, R3Value, RsumValue, HV1Value, HV2Value, HV3Value,
HI1Value, HI2Value, HI3Value, RE1Value, RE2Value, RE3Value, RSumValue,
RECons1Value, RECons2Value, RECons3Value, REConssumValue, REDel1Value, REDel2Value,
REDel3Value, REDelsumValue, AE1Value, AE2Value, AE3Value, ASumValue, RaE1Value,
RaE2Value, RaE3Value, RaEsumValue, RaEInd1Value, RaEInd2Value, RaEInd3Value,
RaEIndsumValue, RaECap1Value, RaECap2Value, RaECap3Value, RaECapsumValue);
Form1.numOfRecs++;
if (savedData)
{
V1.Clear();
V2.Clear();
V3.Clear();
V12.Clear();
V13.Clear();
V23.Clear();
I1.Clear();
I2.Clear();
I3.Clear();
Isum.Clear();
cosf1.Clear();
cosf2.Clear();

```



```
        cosf3.Clear();
        frequency.Clear();
        fieldRotation.Clear();
        P1.Clear();
        P2.Clear();
        P3.Clear();
        Psum.Clear();
        A1.Clear();
        A2.Clear();
        A3.Clear();
        Asum.Clear();
        R1.Clear();
        R2.Clear();
        R3.Clear();
        Rsum.Clear();
        HV1.Clear();
        HV2.Clear();
        HV3.Clear();
        HI1.Clear();
        HI2.Clear();
        HI3.Clear();
        RE1.Clear();
        RE2.Clear();
        RE3.Clear();
        REsum.Clear();
        RECons1.Clear();
        RECons2.Clear();
        RECons3.Clear();
        REConssum.Clear();
        REDe11.Clear();
        REDe12.Clear();
        REDe13.Clear();
        REDelsum.Clear();
        AE1.Clear();
        AE2.Clear();
        AE3.Clear();
        Asum.Clear();
        RaE1.Clear();
        RaE2.Clear();
        RaE3.Clear();
        RaEsum.Clear();
        RaEInd1.Clear();
        RaEInd2.Clear();
        RaEInd3.Clear();
        RaEIndsum.Clear();
        RaECap1.Clear();
        RaECap2.Clear();
        RaECap3.Clear();
        RaECapsum.Clear();
    }
}

private void checkInstrumentTmr_Tick(object sender, EventArgs e)
{
    float iVar;

    iVar = inst.SimReadFloat(Form1.mBC, 19000);
    L1VGXM.Value = iVar;
    V1.Add(iVar);
    iVar = inst.SimReadFloat(Form1.mBC, 19002);
    L2VGXM.Value = iVar;
}
```

```
V2.Add(iVar);
iVar = inst.SimReadFloat(Form1.mBC, 19004);
L3VGXM.Value = iVar;
V3.Add(iVar);
iVar = inst.SimReadFloat(Form1.mBC, 19006);
L12VGXM.Value = iVar;
V12.Add(iVar);
iVar = inst.SimReadFloat(Form1.mBC, 19008);
L23VGXM.Value = iVar;
V23.Add(iVar);
iVar = inst.SimReadFloat(Form1.mBC, 19010);
L13VGXM.Value = iVar;
V13.Add(iVar);
iVar = inst.SimReadFloat(Form1.mBC, 19012);
L1CGXM.Value = iVar;
I1.Add(iVar);
iVar = inst.SimReadFloat(Form1.mBC, 19012);
L2CGXM.Value = iVar;
I2.Add(iVar);
iVar = inst.SimReadFloat(Form1.mBC, 19016);
L3CGXM.Value = iVar;
I3.Add(iVar);
iVar = inst.SimReadFloat(Form1.mBC, 19018);
Isum.Add(iVar);
iVar = inst.SimReadFloat(Form1.mBC, 19044);
L1cosfGXLEDD.Value = iVar;
cosf1.Add(iVar);
iVar = inst.SimReadFloat(Form1.mBC, 19046);
L2cosfGXLEDD.Value = iVar;
cosf2.Add(iVar);
iVar = inst.SimReadFloat(Form1.mBC, 19048);
L3cosfGXLEDD.Value = iVar;
cosf3.Add(iVar);
iVar = inst.SimReadFloat(Form1.mBC, 19050);
FrequencyGXM.Value = iVar;
frequency.Add(iVar);
iVar = inst.SimReadFloat(Form1.mBC, 19052);
FRotGXM.Value = iVar;
fieldRotation.Add(iVar);
iVar = inst.SimReadFloat(Form1.mBC, 19020);
P1GXLEDD.Value = iVar;
P1.Add(iVar);
iVar = inst.SimReadFloat(Form1.mBC, 19022);
P2GXLEDD.Value = iVar;
P2.Add(iVar);
iVar = inst.SimReadFloat(Form1.mBC, 19024);
P3GXLEDD.Value = iVar;
P3.Add(iVar);
iVar = inst.SimReadFloat(Form1.mBC, 19026);
PSGXLEDD.Value = iVar;
Psum.Add(iVar);
iVar = inst.SimReadFloat(Form1.mBC, 19028);
A1GXLEDD.Value = iVar;
A1.Add(iVar);
iVar = inst.SimReadFloat(Form1.mBC, 19030);
A2GXLEDD.Value = iVar;
A2.Add(iVar);
iVar = inst.SimReadFloat(Form1.mBC, 19032);
A3GXLEDD.Value = iVar;
A3.Add(iVar);
iVar = inst.SimReadFloat(Form1.mBC, 19034);
ASGXLEDD.Value = iVar;
```

```

Asum.Add(iVar);
iVar = inst.SimReadFloat(Form1.mBC, 19036);
R1GXLEDD.Value = iVar;
R1.Add(iVar);
iVar = inst.SimReadFloat(Form1.mBC, 19038);
R2GXLEDD.Value = iVar;
R2.Add(iVar);
iVar = inst.SimReadFloat(Form1.mBC, 19040);
R3GXLEDD.Value = iVar;
R3.Add(iVar);
iVar = inst.SimReadFloat(Form1.mBC, 19042);
RSGXLEDD.Value = iVar;
Rsum.Add(iVar);
//real energy
iVar = inst.SimReadFloat(Form1.mBC, 19054);
RE1GXLEDD.Value = iVar;
RE1CprBar.Maximum = (int)(Math.Abs(iVar));
RE1DprBar.Maximum = (int)(iVar);
RE1.Add(iVar);
iVar = inst.SimReadFloat(Form1.mBC, 19056);
RE2GXLEDD.Value = iVar;
RE2CprBar.Maximum = (int)(iVar);
RE2DprBar.Maximum = (int)(iVar);
RE2.Add(iVar);
iVar = inst.SimReadFloat(Form1.mBC, 19058);
RE3GXLEDD.Value = iVar;
RE3CprBar.Maximum = (int)(iVar);
RE3DprBar.Maximum = (int)(iVar);
RE3.Add(iVar);
iVar = inst.SimReadFloat(Form1.mBC, 19060);
RESGXLEDD.Value = iVar;

if (iVar > 0)
{
    RESCprBar.Maximum = (int)(iVar);
    RESDprBar.Maximum = (int)(iVar);
}

REsum.Add(iVar);
//real energy consumed
iVar = inst.SimReadFloat(Form1.mBC, 19062);

if (iVar <= RE1CprBar.Maximum)
    RE1CprBar.Value = (int)(iVar);

RECons1.Add(iVar);
iVar = inst.SimReadFloat(Form1.mBC, 19064);

if (iVar <= RE2CprBar.Maximum)
    RE2CprBar.Value = (int)(iVar);

RECons2.Add(iVar);
iVar = inst.SimReadFloat(Form1.mBC, 19066);

if (iVar<=RE3CprBar.Maximum)
    RE3CprBar.Value = (int)(iVar);

RECons3.Add(iVar);
iVar = inst.SimReadFloat(Form1.mBC, 19068);

if (iVar <= RESCprBar.Maximum)
    RESCprBar.Value = (int)(iVar);

```

```

REConssum.Add(iVar);
//real energy delivered
iVar = inst.SimReadFloat(Form1.mBC, 19070);

if(iVar<=RE1DprBar.Maximum)
    RE1DprBar.Value = (int)(iVar);

REDe11.Add(iVar);
iVar = inst.SimReadFloat(Form1.mBC, 19072);

if (iVar <= RE2DprBar.Maximum)
    RE2DprBar.Value = (int)(iVar);

REDe12.Add(iVar);
iVar = inst.SimReadFloat(Form1.mBC, 19074);

if (iVar <= RE3DprBar.Maximum)
    RE3DprBar.Value = (int)(iVar);
REDe13.Add(iVar);
iVar = inst.SimReadFloat(Form1.mBC, 19076);

if (iVar <= RESDprBar.Maximum)
    RESDprBar.Value = (int)(iVar);

REDe1sum.Add(iVar);
//aparent energy
iVar = inst.SimReadFloat(Form1.mBC, 19078);
AE1GXLEDD.Value = iVar;
AE1.Add(iVar);
iVar = inst.SimReadFloat(Form1.mBC, 19080);
AE2GXLEDD.Value = iVar;
AE2.Add(iVar);
iVar = inst.SimReadFloat(Form1.mBC, 19082);
AE3GXLEDD.Value = iVar;
AE3.Add(iVar);
iVar = inst.SimReadFloat(Form1.mBC, 19084);
AESGXLEDD.Value = iVar;
AESum.Add(iVar);
//reactive energy
iVar = inst.SimReadFloat(Form1.mBC, 19086);
RaE1GXLEDD.Value = iVar;
RaE1IprBar.Maximum = (int)(iVar);
RaE1CprBar.Maximum = (int)(iVar);
RaE1.Add(iVar);
iVar = inst.SimReadFloat(Form1.mBC, 19088);
RaE2GXLEDD.Value = iVar;
RaE2IprBar.Maximum = (int)(iVar);
RaE2CprBar.Maximum = (int)(iVar);
RaE2.Add(iVar);
iVar = inst.SimReadFloat(Form1.mBC, 19090);
RaE3GXLEDD.Value = iVar;
RaE3IprBar.Maximum = (int)(iVar);
RaE3CprBar.Maximum = (int)(iVar);
RaE3.Add(iVar);
iVar = inst.SimReadFloat(Form1.mBC, 19092);
RaESGXLEDD.Value = iVar;
RaESIprBar.Maximum = (int)(Math.Abs(iVar));
RaESCprBar.Maximum = (int)(Math.Abs(iVar));
RaESum.Add(iVar);
iVar = inst.SimReadFloat(Form1.mBC, 19094);

```

```
if (iVar<= RaE1IprBar.Maximum)
    RaE1IprBar.Value = (int)(iVar);

RaEInd1.Add(iVar);
iVar = inst.SimReadFloat(Form1.mBC, 19096);

if (iVar <= RaE2IprBar.Maximum)
    RaE2IprBar.Value = (int)(iVar);

RaEInd2.Add(iVar);
iVar = inst.SimReadFloat(Form1.mBC, 19098);

if (iVar <= RaE3IprBar.Maximum)
    RaE3IprBar.Value = (int)(iVar);

RaEInd3.Add(iVar);
iVar = inst.SimReadFloat(Form1.mBC, 19100);

if (iVar <= RaE3IprBar.Maximum)
    RaESIprBar.Value = (int)(iVar);

RaEIndsum.Add(iVar);
iVar = inst.SimReadFloat(Form1.mBC, 19102);

if (iVar <= RaE1CprBar.Maximum)
    RaE1CprBar.Value = (int)(iVar);

RaECap1.Add(iVar);
iVar = inst.SimReadFloat(Form1.mBC, 19104);

if (iVar <= RaE2CprBar.Maximum)
    RaE2CprBar.Value = (int)(iVar);

RaECap2.Add(iVar);
iVar = inst.SimReadFloat(Form1.mBC, 19106);

if (iVar <= RaE3CprBar.Maximum)
    RaE3CprBar.Value = (int)(iVar);

RaECap3.Add(iVar);
iVar = inst.SimReadFloat(Form1.mBC, 19108);

if (iVar <= RaESCprBar.Maximum)
    RaESCprBar.Value = (int)(iVar);

RaECapsum.Add(iVar);
iVar = inst.SimReadFloat(Form1.mBC, 19110);
HV1GXM.Value = iVar;
HV1.Add(iVar);
iVar = inst.SimReadFloat(Form1.mBC, 19112);
HV2GXM.Value = iVar;
HV2.Add(iVar);
iVar = inst.SimReadFloat(Form1.mBC, 19114);
HV3GXM.Value = iVar;
HV3.Add(iVar);
iVar = inst.SimReadFloat(Form1.mBC, 19116);
HI1GXM.Value = iVar;
HI1.Add(iVar);
iVar = inst.SimReadFloat(Form1.mBC, 19118);
HI2GXM.Value = iVar;
HI2.Add(iVar);
iVar = inst.SimReadFloat(Form1.mBC, 19120);
```

```

        HI3GXM.Value = iVar;
        HI3.Add(iVar);
    }

    private void showDBData_Click(object sender, EventArgs e)
    {
        DataQuery dataquery = new DataQuery();
        dataquery.Show();
    }
}

```

Μεταβλητές - Μέθοδοι – Events	Περιγραφή
<code>public Janitza_UMG96RM()</code>	Αρχικοποίηση παραθύρου.
<code>private simReadWriteFloat inst = new simReadWriteFloat();</code>	Τοπική μεταβλητή που ορίζεται στο αρχείο modBusFloats.
<pre> public List<float> V1 = new List<float>(); public List<float> V2 = new List<float>(); public List<float> V3 = new List<float>(); public List<float> V12 = new List<float>(); public List<float> V13 = new List<float>(); public List<float> V23 = new List<float>(); public List<float> I1 = new List<float>(); public List<float> I2 = new List<float>(); public List<float> I3 = new List<float>(); public List<float> Isum = new List<float>(); public List<float> cosf1 = new List<float>(); public List<float> cosf2 = new List<float>(); public List<float> cosf3 = new List<float>(); public List<float> frequency = new List<float>(); public List<float> fieldRotation = new List<float>(); public List<float> P1 = new List<float>(); public List<float> P2 = new List<float>(); public List<float> P3 = new List<float>(); public List<float> Psum = new List<float>(); public List<float> A1 = new List<float>(); public List<float> A2 = new List<float>(); </pre>	<p>Ορίζονται δημόσιες λίστες από floats (μεταβλητές κυλιόμενης υποδιαστολής) που αντιστοιχούν στα φυσικά μετρούμενα μεγέθη.</p>

Μεταβλητές - Μέθοδοι – Events	Περιγραφή
<pre> List<float>(); public List<float> A3 = new List<float>(); public List<float> Asum = new List<float>(); public List<float> R1 = new List<float>(); public List<float> R2 = new List<float>(); public List<float> R3 = new List<float>(); public List<float> Rsum = new List<float>(); public List<float> RE1 = new List<float>(); public List<float> RE2 = new List<float>(); public List<float> RE3 = new List<float>(); public List<float> RESum = new List<float>(); public List<float> RECons1 = new List<float>(); public List<float> RECons2 = new List<float>(); public List<float> RECons3 = new List<float>(); public List<float> REConssum = new List<float>(); public List<float> REDel1 = new List<float>(); public List<float> REDel2 = new List<float>(); public List<float> REDel3 = new List<float>(); public List<float> REDelsum = new List<float>(); public List<float> AE1 = new List<float>(); public List<float> AE2 = new List<float>(); public List<float> AE3 = new List<float>(); public List<float> AEsSum = new List<float>(); public List<float> RaE1 = new List<float>(); public List<float> RaE2 = new List<float>(); public List<float> RaE3 = new List<float>(); public List<float> RaESum = new List<float>(); public List<float> RaEInd1 = new List<float>(); public List<float> RaEInd2 = new List<float>(); public List<float> RaEInd3 = new List<float>(); public List<float> RaEIndsum = new List<float>(); </pre>	

Μεταβλητές - Μέθοδοι – Events	Περιγραφή
<pre> public List<float> RaECap1 = new List<float>(); public List<float> RaECap2 = new List<float>(); public List<float> RaECap3 = new List<float>(); public List<float> RaECapsum = new List<float>(); public List<float> HV1 = new List<float>(); public List<float> HV2 = new List<float>(); public List<float> HV3 = new List<float>(); public List<float> HI1 = new List<float>(); public List<float> HI2 = new List<float>(); public List<float> HI3 = new List<float>(); </pre>	
<pre> private Statistics stat = new Statistics(); </pre>	<p>Τοπική μεταβλητή ορίζεται στο αρχείο Statistics.cs.</p>
<pre> private SQLInterface iSQL = new SQLInterface(); </pre>	<p>Τοπική μεταβλητή ορίζεται στο αρχείο SQLInterface.cs.</p>
<pre> public static SqlConnection SQLCon; </pre>	<p>Καθολική στατική μεταβλητή που ορίζεται στην βιβλιοθήκη συστήματος System.Data.SqlClient.</p>
<pre> private void aquisitionTmr_Tick(object sender, EventArgs e) </pre>	<p>Καταγραφή μετρήσεων στην βάση δεδομένων. Εάν δεν υπάρχει σύνδεση στην βάση, δημιουργείται, μέσω της μεταβλητής SQLCon.</p> <p>Οι καταγραφόμενες τιμές είναι:</p> <ul style="list-style-type: none"> • Μέσος όρος τιμών για χρονικό διάστημα που έχει οριστεί στον timer (sender)(1 λεπτό): <ol style="list-style-type: none"> 1. Τάση L1. 2. Τάση L2. 3. Τάση L3. 4. Τάση L1-L2. 5. Τάση L1-L3. 6. Τάση L2-L3. 7. Ένταση L1. 8. Ένταση L2. 9. Ένταση L3. 10. Άθροισμα έντασης. 11. Συντελεστής ισχύος L1. 12. Συντελεστής ισχύος L2. 13. Συντελεστής ισχύος L3

Μεταβλητές - Μέθοδοι – Events	Περιγραφή
	<ul style="list-style-type: none"> 14. Συχνότητα. 15. Περιστροφή μαγνητικού πεδίου 16. Ισχύς L1. 17. Ισχύς L2. 18. Ισχύς L3. 19. Άθροισμα ισχύος. 20. Φαινόμενη ισχύς L1. 21. Φαινόμενη ισχύς L2. 22. Φαινόμενη ισχύς L3. 23. Φαινόμενη ισχύς Άθροισμα. 24. Άεργη ισχύς L1. 25. Άεργη ισχύς L2. 26. Άεργη ισχύς L3. 27. Άθροισμα άεργης ισχύος. 28. Αρμονική τάσης L1. 29. Αρμονική τάσης L2. 30. Αρμονική τάσης L3. 31. Αρμονική έντασης L1. 32. Αρμονική έντασης L2. 33. Αρμονική έντασης L3. • Καταγραφή της τρέχουσας τιμής: <ul style="list-style-type: none"> 1. Πραγματική ενέργεια L1. 2. Πραγματική ενέργεια L2. 3. Πραγματική ενέργεια L3. 4. Άθροισμα πραγματικής ενέργειας. 5. Κατανάλωση πραγματικής ενέργειας L1. 6. Κατανάλωση πραγματικής ενέργειας L2. 7. Κατανάλωση πραγματικής ενέργειας L3. 8. Άθροισμα κατανάλωσης πραγματικής ενέργειας L1. 9. Παραδοτέα πραγματική ενέργεια L1. 10. Παραδοτέα πραγματική ενέργεια L2. 11. Παραδοτέα πραγματική ενέργεια L3. 12. Άθροισμα παραδοτέας πραγματικής ενέργειας. 13. Φαινόμενη ενέργεια L1.

Μεταβλητές - Μέθοδοι – Events	Περιγραφή
	<ol style="list-style-type: none"> 14. Φαινόμενη ενέργεια L2. 15. Φαινόμενη ενέργεια L3. 16. Άθροισμα φαινόμενης ενέργειας. 17. Άεργη ενέργεια L1. 18. Άεργη ενέργεια L2. 19. Άεργη ενέργεια L3. 20. Άθροισμα άεργης ενέργειας. 21. Άεργη επαγωγική ενέργεια L1. 22. Άεργη επαγωγική ενέργεια L2. 23. Άεργη επαγωγική ενέργεια L3. 24. Άθροισμα άεργης επαγωγικής ενέργειας. 25. Άεργη χωρητική ενέργεια L1. 26. Άεργη χωρητική ενέργεια L2. 27. Άεργη χωρητική ενέργεια L3. 28. Άθροισμα άεργης χωρητικής ενέργειας.
<pre>private void checkInstrumentTmr_Tick(object sender, EventArgs e)</pre>	<p>Ενημέρωση των γραφικών στοιχείων του παραθύρου Janitza-UMG96RM.</p> <p>Τα στοιχεία:</p> <ul style="list-style-type: none"> • RE1CprBar. • RE2CprBar. • RE3CprBar. • RESCprBar. • RE1DprBar. • RE2DprBar. • RE3DprBar. • RESDprBar. • RaE1IprBar. • RaE2IprBar. • RaE3IprBar. • RaESIprBar. • RaE1CprBar. • RaE2CprBar.

Μεταβλητές - Μέθοδοι – Events	Περιγραφή
	<ul style="list-style-type: none"> • RaE3CprBar. • RaESCprBar. <p>Έχουν έλεγχο τιμής (Value) σε σχέση με την μέγιστη τιμή (Maximum) λόγω της εξομοίωσης. Σε κανονική λειτουργία μπορούν να παραληφθούν.</p>
<pre>private void showDBData_Click(object sender, EventArgs e)</pre>	<p>Εμφανίζει το παράθυρο Εύρεση μεγεθών από την βάση δεδομένων.</p>

8.3. modBusFloats.cs

Σε αυτή την κλάση ορίζονται οι μέθοδοι ανάγνωσης και εγγραφής στον **Easy Modbus Server Simulator**. Ο server δεν μπορεί να διαχειριστεί μεταβλητές 4 bytes, έτσι οι floating point μεταβλητές διαχειρίζονται σαν 2 ξεχωριστοί ακέραιοι αριθμοί. Επίσης υπάρχουν άδειες μέθοδοι (χωρίς κώδικα) για την περίπτωση που ο μετρητής ενέργειας Janitza-UMG96RM, διαχειρίζεται διαφορετικά τις μεταβλητές τεσσάρων bytes.

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;
using EasyModbus;

namespace exModbus
{
    public class simReadWriteFloat
    {
        int intPart;
        int decimicalPart;

        public void SimWriteFloat(Int32 min, Int32 max, ModbusClient client, Int32
startAddress,int wh, int dec)
        {
            Random rValue = new Random();
            int[] valueToWrite = new int[2];

            if ((startAddress== 19044)|| (startAddress == 19046)|| (startAddress ==
19048)) //cosf
```

```

        {
            valueToWrite[0] = 0;
            valueToWrite[1] = (int)(rValue.Next(400, 900));
        }
        else if ((wh == 0) && (dec == 0))
        {
            valueToWrite[0] = (int)(rValue.Next((int)(min), (int)(max)));
            valueToWrite[1] = (int)(rValue.Next(0, 999));
        }
        else if ((wh == 0) && (dec > 0))
        {
            decimicalPart += dec;

            if (decimicalPart >= 999)
            {
                decimicalPart = 0;
                intPart++;
            }

            valueToWrite[0] = intPart;
            valueToWrite[1] = decimicalPart;
        }
        else if ((min == -1) && (max == -1))
        {
            valueToWrite[0] = (int)(rValue.Next((int)(min), (int)(max)));

            int valS;

            if (valueToWrite[0] < 0)
                valS = -1;
            else
                valS = 1;

            valueToWrite[1] = (int)(rValue.Next(0, 999)*valS);
        }

        client.WriteMultipleRegisters(startAddress-1, valueToWrite);
    }

    public void SimWritePowerAndEnergy(float floatToWrite, ModbusClient client,
Int32 startAddress)
    {
        int[] valueToWrite = new int[2];

        valueToWrite[0] = (int) Math.Abs(Math.Truncate(floatToWrite));

        string valStr = (floatToWrite - Math.Truncate(floatToWrite)).ToString();
        float mid = (float) (floatToWrite -
Math.Truncate(floatToWrite)*((valStr.Length-2)*10));

        valueToWrite[1] = (int) (mid);

        client.WriteMultipleRegisters(startAddress - 1, valueToWrite);
    }

    public void WriteFloat()
    {

```

```

    public float SimReadFloat(ModbusClient client, Int32 startAddress)
    {
        int[] valInt = client.ReadHoldingRegisters(startAddress-1, 2);
        //.ReadHoldingRegisters(19000, 1);

        float rVal = (float)(valInt[0] + (valInt[1] * 0.001));
        return (rVal);
    }

    public float ReadFloat()
    {
        return (0f);
    }
}

```

Μεταβλητές - Μέθοδοι – Events	Περιγραφή
<pre> public void SimWriteFloat(Int32 min, Int32 max, ModbusClient client, Int32 startAddress,int wh, int dec) </pre>	<p>Γράφει μία μεταβλητή, με τυχαία τιμή, στον server εξομοίωσης. Παράμετροι:</p> <ul style="list-style-type: none"> • Int32 min. Ελάχιστη τιμή για εγγραφή. • Int32 max. Μέγιστη τιμή για εγγραφή • ModbusClient client. Ο ενεργός client. • Int32 startAddress. Η διεύθυνση του server του πρώτου μέρους του αριθμού που θα εγγραφεί. • int wh και int dec. Μεταβλητές ειδικού ελέγχου.
<pre> public void SimWritePowerAndEnergy(float floatToWrite, ModbusClient client, Int32 startAddress) </pre>	<p>Γράφει μία μεταβλητή, με καθορισμένη τιμή, στον server εξομοίωσης. Παράμετροι:</p> <ul style="list-style-type: none"> • float floatToWrite. Τιμή που θα γραφτεί στον Server εξομοίωσης. • ModbusClient client. Ο ενεργός client. • Int32 startAddress. Η διεύθυνση του server του πρώτου μέρους του αριθμού που θα εγγραφεί.

Μεταβλητές - Μέθοδοι – Events	Περιγραφή
<code>public void WriteFloat()</code>	Μέθοδος για μελλοντική χρήση.
<code>public float SimReadFloat(ModbusClient client, Int32 startAddress)</code>	<p>Διαβάζει μια μεταβλητή από τον server εξομοίωσης. Παράμετροι:</p> <ul style="list-style-type: none"> • <code>ModbusClient</code> client. Ο ενεργός client. • <code>Int32</code> startAddress. Η διεύθυνση του server του πρώτου μέρους του αριθμού που θα αναγνωσθεί.
<code>public float ReadFloat()</code>	Μέθοδος για μελλοντική χρήση.

8.4. Statistics.

Σε αυτή την κλάση ορίζονται μαθηματικές συναρτήσεις για στατιστική επεξεργασία των μετρούμενων μεγεθών.

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace MODBUS
{
    class Statistics
    {
        public float GetMeanValue(List<float> inputList)
        {
            float tmp = 0, returnVal = 0;

            for (int z = 0; z < inputList.Count; z++)
                tmp += inputList[z];

            returnVal = tmp / inputList.Count;

            return (returnVal);
        }

        public float GetMinValue(List<float> inputList)
        {
            float tmp = inputList[0], returnVal = 0;

            for (int z=0; z<inputList.Count;z++)
            {
                if (inputList[z]<tmp)
                    tmp = inputList[z];
            }
        }
    }
}

```

```
        returnVal = tmp;
        return (returnVal);
    }

    public float GetMaxValue(List<float> inputList)
    {
        float tmp = inputList[0], returnVal = 0;

        for (int z = 0; z < inputList.Count; z++)
        {
            if (inputList[z]>tmp)
                tmp = inputList[z];
        }

        returnVal = tmp;
        return (returnVal);
    }

    public float GetMeanIncrease(List<float> inputList)
    {
        float tmp = 0, returnVal = 0;

        for (int z = 0; z < inputList.Count - 1; z++)
            tmp += inputList[z + 1] - inputList[z];

        returnVal = (float)(tmp / inputList.Count);
        return (returnVal);
    }

    public float GetMinIncrease(List<float> inputList)
    {
        float tmp = 0, returnVal = 0;

        for (int z = 0; z < inputList.Count - 1; z++)
        {
            tmp = inputList[z + 1] - inputList[z];

            if (tmp < returnVal)
                returnVal = tmp;
        }
        return (returnVal);
    }

    public float GetMaxIncrease(List<float> inputList)
    {
        float tmp = 0, returnVal = 0;

        for (int z = 0; z < inputList.Count - 1; z++)
        {
            tmp = inputList[z + 1] - inputList[z];

            if (tmp > returnVal)
                returnVal = tmp;
        }
        return (returnVal);
    }
}
```

Μεταβλητές - Μέθοδοι - Events	Περιγραφή
<code>public float GetMeanValue(List<float> inputList)</code>	Επιστρέφει την μέση τιμή μιας λίστας τιμών.
<code>public float GetMinValue(List<float> inputList)</code>	Επιστρέφει την ελάχιστη τιμή μιας λίστας τιμών.
<code>public float GetMaxValue(List<float> inputList)</code>	Επιστρέφει την μέγιστη τιμή μιας λίστας τιμών.
<code>public float GetMeanIncrease(List<float> inputList)</code>	Επιστρέφει την μέση τιμή απόκλισης μιας λίστας τιμών.
<code>public float GetMinIncrease(List<float> inputList)</code>	Επιστρέφει την ελάχιστη τιμή απόκλισης μιας λίστας τιμών.
<code>public float GetMaxIncrease(List<float> inputList)</code>	Επιστρέφει την μέγιστη τιμή απόκλισης μιας λίστας τιμών.

8.5. SQLInterface.cs

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Data.SqlClient;
using System.Data;

namespace MODBUS
{
    class SQLInterface
    {
        public Boolean JanitzaUMG96R_SaveToDB(int recordNum, SqlConnection con,
        DateTime now, float V1, float V2, float V3, float V12, float V13, float V23,
        float I1, float I2, float I3, float Isum, float cosf1, float cosf2,
        float cosf3, float frequency, float fieldRotation, float P1,
        float P2, float P3, float Psum, float A1, float A2, float A3, float
        Asum, float R1, float R2, float R3, float Rsum,
        float HV1, float HV2, float HV3, float HI1, float HI2, float HI3, float
        RE1, float RE2, float RE3, float REsum,
        float RECons1, float RECons2, float RECons3, float REConssum, float
        REDe11, float REDe12, float REDe13, float REDe1sum,
        float AE1, float AE2, float AE3, float AEsu, float RaE1, float RaE2,
        float RaE3, float RaEsum, float RaEInd1,
        float RaEInd2, float RaEInd3, float RaEIndsum, float RaECap1, float
        RaECap2, float RaECap3, float RaECapsum)
        {
            string query = "Insert into Measurements(id, DateTime, V1, V2, V3, V12,
            V13, V23, I1, I2, I3, Isum, cosf1, cosf2, cosf3, frequency, fieldRotation, P1, P2,
            P3, Psum, A1, A2, A3, Asum, R1, R2, R3, Rsum, HV1, HV2, HV3, HI1, HI2, HI3, RE1,
            RE2, RE3, REsum, RECons1, RECons2, RECons3, REConssum, REDe11, REDe12, REDe13,

```



```

REDElsum, AE1, AE2, AE3, AEsuM, RaE1, RaE2, RaE3, RaEsuM, RaEInd1, RaEInd2, RaEInd3,
RaEIndsum, RaECap1, RaECap2, RaECap3, RaECapsum) values (''+ recordNum + "','" + now
+ "','" + V1 + "','" + V2 + "','" + V3 + "','" + V12 + "','" + V13 + "','" + V23 +
','' + I1 + "','" + I2 + "','" + I3 + "','" + Isum + "','" + cosf1 + "','" + cosf2
+ "','" + cosf3 + "','" + frequency + "','" + fieldRotation + "','" + P1 + "','" +
P2 + "','" + P3 + "','" + Psum + "','" + A1 + "','" + A2 + "','" + A3 + "','" + Asum
+ "','" + R1 + "','" + R2 + "','" + R3 + "','" + Rsum + "','" + HV1 + "','" + HV2 +
','' + HV3 + "','" + HI1 + "','" + HI2 + "','" + HI3 + "','" + RE1 + "','" + RE2 +
','' + RE3 + "','" + REsum + "','" + RECons1 + "','" + RECons2 + "','" + RECons3 +
','' + REConssum + "','" + REDe11 + "','" + REDe12 + "','" + REDe13 + "','" +
REDElsum + "','" + AE1 + "','" + AE2 + "','" + AE3 + "','" + AEsuM + "','" + RaE1 +
','' + RaE2 + "','" + RaE3 + "','" + RaEsuM + "','" + RaEInd1 + "','" + RaEInd2 +
','' + RaEInd3 + "','" + RaEIndsum + "','" + RaECap1 + "','" + RaECap2 + "','" +
RaECap3 + "','" + RaECapsum + "')";
    con.Open();
    SqlCommand sqlcmd = new SqlCommand(query, con);
    int res = sqlcmd.ExecuteNonQuery();
    con.Close();

    if (res != 0)
        return (true);
    else
        return (false);
}

public DataTable JanitzaUMG96R_GetDataTable(SqlConnection con, DateTime
from, DateTime to)
{
    con.Open();
    SqlDataAdapter dataAdapt = new SqlDataAdapter("Select DateTime, V1, V2,
V3, V12, V13, V23, I1, I2, I3, Isum, cosf1, cosf2, cosf3, frequency, fieldRotation,
P1, P2, P3, Psum, A1, A2, A3, Asum, R1, R2, R3, Rsum, HV1, HV2, HV3, HI1, HI2, HI3,
RE1, RE2, RE3, REsum, RECons1, RECons2, RECons3, REConssum, REDe11, REDe12, REDe13,
REDElsum, AE1, AE2, AE3, AEsuM, RaE1, RaE2, RaE3, RaEsuM, RaEInd1, RaEInd2, RaEInd3,
RaEIndsum, RaECap1, RaECap2, RaECap3, RaECapsum from Measurements where DateTime
Between '' + from + '' and '' + to + '' order by DateTime", con);
    DataTable returnTable = new DataTable();
    dataAdapt.Fill(returnTable);
    con.Close();
    return (returnTable);
}
}
}

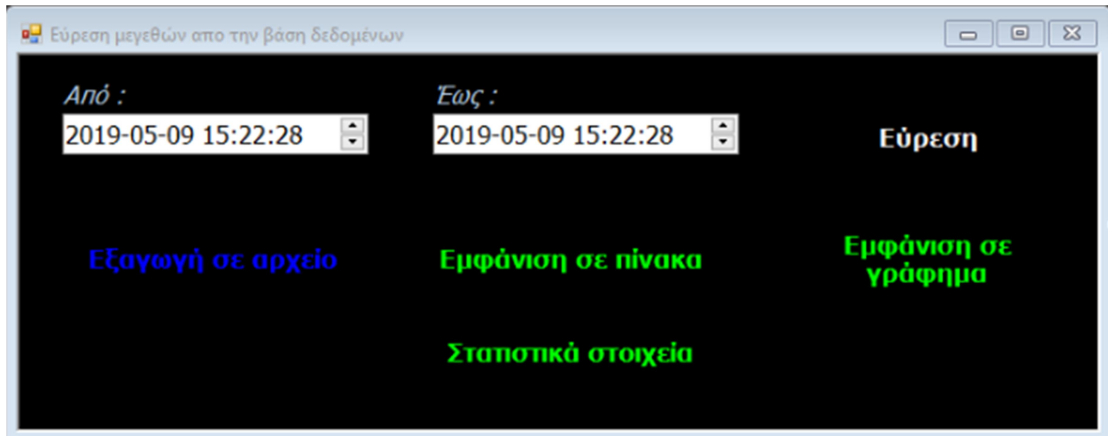
```

Μεταβλητές - Μέθοδοι – Events	Περιγραφή
<pre> public Boolean JanitzaUMG96R_SaveToDB(int recordNum, SqlConnection con, DateTime now, float V1, float V2, float V3, float V12, float V13, float V23, float I1, float I2, float I3, float Isum, float cosf1, float cosf2, float cosf3, float frequency, float fieldRotation, float P1, float P2, float P3, float Psum, float A1, float A2, float A3, float Asum, float R1, float R2, float R3, float Rsum, float HV1, float HV2, float HV3, float HI1, float HI2, float HI3, float RE1, float RE2, float RE3, float REsum, float RECons1, float RECons2, float RECons3, </pre>	<p>Αυτή η μέθοδος καταγράφει τα μετρούμενα στοιχεία στην βάση δεδομένων. Παράμετροι:</p> <ul style="list-style-type: none"> • int recordNum. Αριθμός εγγραφής. • SqlConnection con Σύνδεση με την βάση δεδομένων. • DateTime now Χρονική στιγμή που γίνονται οι

Μεταβλητές - Μέθοδοι – Events	Περιγραφή
<p>float REConssum, float REDe11, float REDe12, float REDe13, float REDe1sum, float AE1, float AE2, float AE3, float AEsom, float RaE1, float RaE2, float RaE3, float RaEsum, float RaEInd1, float RaEInd2, float RaEInd3, float RaEIndsum, float RaECap1, float RaECap2, float RaECap3, float RaECapsum)</p>	<p>εγγραφές στην βάση δεδομένων.</p> <p>Επιστρέφει TRUE αν η εγγραφή έχει πετύχει.</p>
<p>public DataTable JanitzaUMG96R_GetDataTable(SqlConnection con, DateTime from, DateTime to)</p>	<p>Επιστρέφει έναν πίνακα δεδομένων, μέσω SQL query . Παράμετροι:</p> <ul style="list-style-type: none"> • SqlConnection con. Σύνδεση με την βάση δεδομένων. • DateTime from...DateTime to. Η χρονική περίοδος που αφορά το ερώτημα στην βάση δεδομένων.

8.6. Εύρεση μεγεθών από την βάση δεδομένων.

8.6.1. DataQuery.cs[Design].



Σε αυτό παράθυρο εμφανίζονται επιλογές για την εμφάνιση καταγεγραμμένων στοιχείων στην βάση δεδομένων:

Στοιχείο παραθύρων	Περιγραφή
--------------------	-----------

Στοιχείο παραθύρων	Περιγραφή
Από:	Επιλογή του χρονικού διαστήματος (ημερομηνία και ώρα) για το οποίο θα εμφανιστούν στοιχεία από την βάση δεδομένων.
Έως	
Εύρεση	Αφού επιλεγεί το χρονικό διάστημα, εκτελεί ερώτημα στην βάση δεδομένων και αν υπάρξει απάντηση από την βάση, εμφανίζει τα κουμπιά: <ul style="list-style-type: none"> • Εξαγωγή σε αρχείο • Εμφάνιση σε πίνακα • Στατιστικά στοιχεία • Εμφάνιση σε γράφημα
Εξαγωγή σε αρχείο	Εμφανίζει επιλογές για την εξαγωγή στοιχείων.
Εμφάνιση σε πίνακα	Εμφανίζει το παράθυρο Πίνακας αποθηκευμένων μεγεθών .
Στατιστικά στοιχεία	Εμφανίζει το παράθυρο Στατιστικά στοιχεία .
Εμφάνιση σε γράφημα	Εμφανίζει το παράθυρο Γράφημα αποθηκευμένων μεγεθών .

8.6.1. DataQuery.cs.

```

using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;
using System.Data.SqlClient;
using System.IO;

namespace MODBUS
{
    public partial class DataQuery : Form
    {
        public DataQuery()
        {
            InitializeComponent();
        }
    }
}

```

```
}

private SQLInterface iSql = new SQLInterface();
public static DataTable answerDataTable = new DataTable();

public static List<DateTime> dtTimeStamp = new List<DateTime>();
public static List<float> dtV1 = new List<float>();
public static List<float> dtV2 = new List<float>();
public static List<float> dtV3 = new List<float>();
public static List<float> dtV12 = new List<float>();
public static List<float> dtV13 = new List<float>();
public static List<float> dtV23 = new List<float>();
public static List<float> dtI1 = new List<float>();
public static List<float> dtI2 = new List<float>();
public static List<float> dtI3 = new List<float>();
public static List<float> dtIsum = new List<float>();
public static List<float> dtcosf1 = new List<float>();
public static List<float> dtcosf2 = new List<float>();
public static List<float> dtcosf3 = new List<float>();
public static List<float> dtfrequency = new List<float>();
public static List<float> dtfieldRotation = new List<float>();
public static List<float> dtP1 = new List<float>();
public static List<float> dtP2 = new List<float>();
public static List<float> dtP3 = new List<float>();
public static List<float> dtPsum = new List<float>();
public static List<float> dtA1 = new List<float>();
public static List<float> dtA2 = new List<float>();
public static List<float> dtA3 = new List<float>();
public static List<float> dtAsum = new List<float>();
public static List<float> dtR1 = new List<float>();
public static List<float> dtR2 = new List<float>();
public static List<float> dtR3 = new List<float>();
public static List<float> dtRsum = new List<float>();
public static List<float> dtHV1 = new List<float>();
public static List<float> dtHV2 = new List<float>();
public static List<float> dtHV3 = new List<float>();
public static List<float> dtHI1 = new List<float>();
public static List<float> dtHI2 = new List<float>();
public static List<float> dtHI3 = new List<float>();
public static List<float> dtRE1 = new List<float>();
public static List<float> dtRE2 = new List<float>();
public static List<float> dtRE3 = new List<float>();
public static List<float> dtREsum = new List<float>();
public static List<float> dtRECons1 = new List<float>();
public static List<float> dtRECons2 = new List<float>();
public static List<float> dtRECons3 = new List<float>();
public static List<float> dtREConssum = new List<float>();
public static List<float> dtREDel1 = new List<float>();
public static List<float> dtREDel2 = new List<float>();
public static List<float> dtREDel3 = new List<float>();
public static List<float> dtREDelsum = new List<float>();
public static List<float> dtAE1 = new List<float>();
public static List<float> dtAE2 = new List<float>();
public static List<float> dtAE3 = new List<float>();
public static List<float> dtAEsum = new List<float>();
public static List<float> dtRaE1 = new List<float>();
public static List<float> dtRaE2 = new List<float>();
public static List<float> dtRaE3 = new List<float>();
public static List<float> dtRaEsum = new List<float>();
public static List<float> dtRaEInd1 = new List<float>();
public static List<float> dtRaEInd2 = new List<float>();
public static List<float> dtRaEInd3 = new List<float>();
```

```
public static List<float> dtRaEIndsum = new List<float>();
public static List<float> dtRaECap1 = new List<float>();
public static List<float> dtRaECap2 = new List<float>();
public static List<float> dtRaECap3 = new List<float>();
public static List<float> dtRaECapsum = new List<float>();

private void Xquery_Click(object sender, EventArgs e)
{
    dtTimeStamp.Clear();
    dtV1.Clear();
    dtV2.Clear();
    dtV3.Clear();
    dtV12.Clear();
    dtV13.Clear();
    dtV23.Clear();
    dtI1.Clear();
    dtI2.Clear();
    dtI3.Clear();
    dtIsum.Clear();
    dtcosf1.Clear();
    dtcosf2.Clear();
    dtcosf3.Clear();
    dtfrequency.Clear();
    dtfieldRotation.Clear();
    dtP1.Clear();
    dtP2.Clear();
    dtP3.Clear();
    dtPsum.Clear();
    dtA1.Clear();
    dtA2.Clear();
    dtA3.Clear();
    dtAsum.Clear();
    dtR1.Clear();
    dtR2.Clear();
    dtR3.Clear();
    dtRsum.Clear();
    dtHV1.Clear();
    dtHV2.Clear();
    dtHV3.Clear();
    dtHI1.Clear();
    dtHI2.Clear();
    dtHI3.Clear();
    dtRE1.Clear();
    dtRE2.Clear();
    dtRE3.Clear();
    dtREsum.Clear();
    dtRECons1.Clear();
    dtRECons2.Clear();
    dtRECons3.Clear();
    dtREConsum.Clear();
    dtREDel1.Clear();
    dtREDel2.Clear();
    dtREDel3.Clear();
    dtREDelsum.Clear();
    dtAE1.Clear();
    dtAE2.Clear();
    dtAE3.Clear();
    dtAsum.Clear();
    dtRaE1.Clear();
    dtRaE2.Clear();
    dtRaE3.Clear();
}
```

```

dtRaEsum.Clear();
dtRaEInd1.Clear();
dtRaEInd2.Clear();
dtRaEInd3.Clear();
dtRaEIndsum.Clear();
dtRaECap1.Clear();
dtRaECap2.Clear();
dtRaECap3.Clear();
dtRaECapsum.Clear();

if (Janitza_UMG96RM.SQLCon == null)
{
    Janitza_UMG96RM.SQLCon = new SqlConnection("Data Source=DESKTOP-
C8VDASN\\SQLEXPRESS;Initial Catalog=thesis;Integrated Security=True");
}

answerDataTable =
iSql.JanitzaUMG96R_GetDataTable(Janitza_UMG96RM.SQLCon, FromDTP.Value, ToDTP.Value);

showGridBtn.Enabled = true;

for (int z = 0; z<answerDataTable.Rows.Count; z++)
{
    dtTimeStamp.Add((DateTime)(answerDataTable.Rows[z]["DateTime"]));
    dtV1.Add(float.Parse(answerDataTable.Rows[z]["V1"].ToString()));
    dtV2.Add(float.Parse(answerDataTable.Rows[z]["V2"].ToString()));
    dtV3.Add(float.Parse(answerDataTable.Rows[z]["V3"].ToString()));
    dtV12.Add(float.Parse(answerDataTable.Rows[z]["V12"].ToString()));
    dtV13.Add(float.Parse(answerDataTable.Rows[z]["V13"].ToString()));
    dtV23.Add(float.Parse(answerDataTable.Rows[z]["V23"].ToString()));
    dtI1.Add(float.Parse(answerDataTable.Rows[z]["I1"].ToString()));
    dtI2.Add(float.Parse(answerDataTable.Rows[z]["I2"].ToString()));
    dtI3.Add(float.Parse(answerDataTable.Rows[z]["I3"].ToString()));
    dtIsum.Add(float.Parse(answerDataTable.Rows[z]["Isum"].ToString()));

    dtcosf1.Add(float.Parse(answerDataTable.Rows[z]["cosf1"].ToString()));

    dtcosf2.Add(float.Parse(answerDataTable.Rows[z]["cosf2"].ToString()));

    dtcosf3.Add(float.Parse(answerDataTable.Rows[z]["cosf3"].ToString()));

    dtfrequency.Add(float.Parse(answerDataTable.Rows[z]["frequency"].ToString()));

    dtfieldRotation.Add(float.Parse(answerDataTable.Rows[z]["fieldRotation"].ToString())
);

    dtP1.Add(float.Parse(answerDataTable.Rows[z]["P1"].ToString()));
    dtP2.Add(float.Parse(answerDataTable.Rows[z]["P2"].ToString()));
    dtP3.Add(float.Parse(answerDataTable.Rows[z]["P3"].ToString()));
    dtPsum.Add(float.Parse(answerDataTable.Rows[z]["Psum"].ToString()));
    dtA1.Add(float.Parse(answerDataTable.Rows[z]["A1"].ToString()));
    dtA2.Add(float.Parse(answerDataTable.Rows[z]["A2"].ToString()));
    dtA3.Add(float.Parse(answerDataTable.Rows[z]["A3"].ToString()));
    dtAsum.Add(float.Parse(answerDataTable.Rows[z]["Asum"].ToString()));
    dtR1.Add(float.Parse(answerDataTable.Rows[z]["R1"].ToString()));
    dtR2.Add(float.Parse(answerDataTable.Rows[z]["R2"].ToString()));
    dtR3.Add(float.Parse(answerDataTable.Rows[z]["R3"].ToString()));
    dtRsum.Add(float.Parse(answerDataTable.Rows[z]["Rsum"].ToString()));
    dtHV1.Add(float.Parse(answerDataTable.Rows[z]["HV1"].ToString()));
    dtHV2.Add(float.Parse(answerDataTable.Rows[z]["HV2"].ToString()));
    dtHV3.Add(float.Parse(answerDataTable.Rows[z]["HV3"].ToString()));
    dtHI1.Add(float.Parse(answerDataTable.Rows[z]["HI1"].ToString()));
    dtHI2.Add(float.Parse(answerDataTable.Rows[z]["HI2"].ToString()));
}

```

```

        dtHI3.Add(float.Parse(answerDataTable.Rows[z]["HI3"].ToString()));
        dtRE1.Add(float.Parse(answerDataTable.Rows[z]["RE1"].ToString()));
        dtRE2.Add(float.Parse(answerDataTable.Rows[z]["RE2"].ToString()));
        dtRE3.Add(float.Parse(answerDataTable.Rows[z]["RE3"].ToString()));

dtREsum.Add(float.Parse(answerDataTable.Rows[z]["REsum"].ToString()));

dtRECons1.Add(float.Parse(answerDataTable.Rows[z]["RECons1"].ToString()));

dtRECons2.Add(float.Parse(answerDataTable.Rows[z]["RECons2"].ToString()));

dtRECons3.Add(float.Parse(answerDataTable.Rows[z]["RECons3"].ToString()));

dtREConssum.Add(float.Parse(answerDataTable.Rows[z]["REConssum"].ToString()));

dtREDe11.Add(float.Parse(answerDataTable.Rows[z]["REDe11"].ToString()));

dtREDe12.Add(float.Parse(answerDataTable.Rows[z]["REDe12"].ToString()));

dtREDe13.Add(float.Parse(answerDataTable.Rows[z]["REDe13"].ToString()));

dtREDe1sum.Add(float.Parse(answerDataTable.Rows[z]["REDe1sum"].ToString()));
        dtAE1.Add(float.Parse(answerDataTable.Rows[z]["AE1"].ToString()));
        dtAE2.Add(float.Parse(answerDataTable.Rows[z]["AE2"].ToString()));
        dtAE3.Add(float.Parse(answerDataTable.Rows[z]["AE3"].ToString()));

dtAEsum.Add(float.Parse(answerDataTable.Rows[z]["AEsum"].ToString()));
        dtRaE1.Add(float.Parse(answerDataTable.Rows[z]["RaE1"].ToString()));
        dtRaE2.Add(float.Parse(answerDataTable.Rows[z]["RaE2"].ToString()));
        dtRaE3.Add(float.Parse(answerDataTable.Rows[z]["RaE3"].ToString()));

dtRaEsum.Add(float.Parse(answerDataTable.Rows[z]["RaEsum"].ToString()));

dtRaEInd1.Add(float.Parse(answerDataTable.Rows[z]["RaEInd1"].ToString()));

dtRaEInd2.Add(float.Parse(answerDataTable.Rows[z]["RaEInd2"].ToString()));

dtRaEInd3.Add(float.Parse(answerDataTable.Rows[z]["RaEInd3"].ToString()));

dtRaEIndsum.Add(float.Parse(answerDataTable.Rows[z]["RaEIndsum"].ToString()));

dtRaECap1.Add(float.Parse(answerDataTable.Rows[z]["RaECap1"].ToString()));

dtRaECap2.Add(float.Parse(answerDataTable.Rows[z]["RaECap2"].ToString()));

dtRaECap3.Add(float.Parse(answerDataTable.Rows[z]["RaECap3"].ToString()));

dtRaECapsum.Add(float.Parse(answerDataTable.Rows[z]["RaECapsum"].ToString()));
    }

    showGraphBtn.Enabled = true;
    showStatisticsBtn.Enabled = true;
    XportBtn.Enabled = true;
}

private void showGridBtn_Click(object sender, EventArgs e)
{
    DataGrid datagrid = new DataGrid();
    datagrid.Show();
}

private void DataQuery_Load(object sender, EventArgs e)

```

```

    {
        DateTime today = DateTime.Now;
        DateTime fromday = today.AddDays(-today.Day + 1);
        fromday = fromday.AddHours(-today.Hour);
        fromday = fromday.AddMinutes(-today.Minute);
        fromday = fromday.AddSeconds(-today.Second);

        FromDTP.Value = fromday;
    }

    private void showGraphBtn_Click(object sender, EventArgs e)
    {
        DataChart datachart = new DataChart();
        datachart.Show();
    }

    private void showStatisticsBtn_Click(object sender, EventArgs e)
    {
        StatistisData dataStats = new StatistisData();
        dataStats.Show();
    }

    private async void XportBtn_Click(object sender, EventArgs e)
    {
        using (SaveFileDialog SD = new SaveFileDialog() { Filter = "CSV|*.csv",
        ValidateNames = true })
        {
            if (SD.ShowDialog()==DialogResult.OK)
            {
                using (StreamWriter stwrite = new StreamWriter(new
        FileStream(SD.FileName, FileMode.Create), Encoding.UTF8))
                {
                    StringBuilder txt = new StringBuilder();

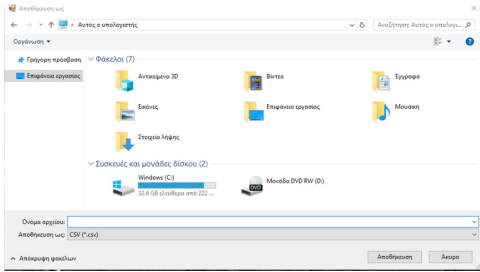
                    txt.AppendLine("Ημερομηνία - ώρα;Ταση L1 (V);Ταση L2(V);Ταση
        L3(V);Ταση L1 - L2(V);Ταση L1 - L3(V);Ταση L2 - L3(V);Ένταση L1(A);Ένταση
        L2(A);Ένταση L3(A);Ένταση Άθροισμα(A);cosf L1;cosf L2;cosf
        L3;Συχνότητα(Hz);Περ.μαγν.πεδίου;Ισχύς L1(W);Ισχύς L2(W);Ισχύς L3(W);Ισχύς
        Άθροισμα(W);Φ.Ισχύς L1(VA);Φ.Ισχύς L2(VA);Φ.Ισχύς L3(VA);Φ.Ισχύς
        Άθροισμα(VA);A.Ισχύς L1(var);A.Ισχύς L2(var);A.Ισχύς L3(var);A.Ισχύς
        Άθροισμα(var);Αρμ.Τασης L1(%);Αρμ.Τασης L2(%);Αρμ.Τασης L3(%);Αρμ.Έντασης
        L1(%);Αρμ.Έντασης L2(%);Αρμ.Έντασης L3(%);Π.Ενέργεια L1(Wh);Π.Ενέργεια
        L2(Wh);Π.Ενέργεια L3(Wh);Π.Ενέργεια Άθροισμα(Wh);Π.Εν.Κατανάλωση
        L1(Wh);Π.Εν.Κατανάλωση L2(Wh);Π.Εν.Κατανάλωση L3(Wh);Π.Εν.Κατανάλωση
        Άθροισμα(Wh);Π.Εν.Παραδοτέα L1(Wh);Π.Εν.Παραδοτέα L2(Wh);Π.Εν.Παραδοτέα
        L3(Wh);Π.Εν.Παραδοτέα Άθροισμα(Wh);Φ.Ενέργεια L1(VAh);Φ.Ενέργεια L2(VAh);Φ.Ενέργεια
        L3(VAh);Φ.Ενέργεια Άθροισμα(VAh);A.Ενέργεια L1(varh);A.Ενέργεια L2(varh);A.Ενέργεια
        L3(varh);A.Ενέργεια Άθροισμα(varh);A.Εν.Επαγωγική L1(varh);A.Εν.Επαγωγική
        L2(varh);A.Εν.Επαγωγική L3(varh);A.Εν.Επαγωγική Άθροισμα(varh);A.Εν.Χωρητική
        L1(varh);A.Εν.Χωρητική L2(varh);A.Εν.Χωρητική L3(varh);A.Εν.Χωρητική
        Άθροισμα(varh);");

                    for (int z=0; z< dtTimeStamp.Count; z++)
                        txt.AppendLine(dtTimeStamp[z].ToString() + ";" +
        dtV1[z].ToString() + ";" + dtV2[z].ToString() + ";" + dtV3[z].ToString() + ";" +
        dtV12[z].ToString() + ";" + dtV13[z].ToString() + ";" + dtV23[z].ToString() + ";" +
        dtI1[z].ToString() + ";" + dtI2[z].ToString() + ";" + dtI3[z].ToString() + ";" +
        dtIsum[z].ToString() + ";" + dtcosf1[z].ToString() + ";" + dtcosf2[z].ToString() +
        ";" + dtcosf3[z].ToString() + ";" + dtfrequency[z].ToString() + ";" +
        dtfieldRotation[z].ToString() + ";" + dtP1[z].ToString() + ";" + dtP2[z].ToString()
        + ";" + dtP3[z].ToString() + ";" + dtPsum[z].ToString() + ";" + dtA1[z].ToString() +
        ";" + dtA2[z].ToString() + ";" + dtA3[z].ToString() + ";" + dtAsum[z].ToString() +

```


Μεταβλητές - Μέθοδοι – Events	Περιγραφή
<pre> public static List<float> dtV1 = new List<float>(); public static List<float> dtV2 = new List<float>(); public static List<float> dtV3 = new List<float>(); public static List<float> dtV12 = new List<float>(); public static List<float> dtV13 = new List<float>(); public static List<float> dtV23 = new List<float>(); public static List<float> dtI1 = new List<float>(); public static List<float> dtI2 = new List<float>(); public static List<float> dtI3 = new List<float>(); public static List<float> dtIsum = new List<float>(); public static List<float> dtcosf1 = new List<float>(); public static List<float> dtcosf2 = new List<float>(); public static List<float> dtcosf3 = new List<float>(); public static List<float> dtfrequency = new List<float>(); public static List<float> dtfieldRotation = new List<float>(); public static List<float> dtP1 = new List<float>(); public static List<float> dtP2 = new List<float>(); public static List<float> dtP3 = new List<float>(); public static List<float> dtPsum = new List<float>(); public static List<float> dtA1 = new List<float>(); public static List<float> dtA2 = new List<float>(); public static List<float> dtA3 = new List<float>(); public static List<float> dtAsum = new List<float>(); public static List<float> dtR1 = new List<float>(); public static List<float> dtR2 = new List<float>(); public static List<float> dtR3 = new List<float>(); public static List<float> dtRsum = new List<float>(); public static List<float> dtHV1 = new List<float>(); public static List<float> dtHV2 = new List<float>(); public static List<float> dtHV3 = new List<float>(); public static List<float> dtHI1 = new List<float>(); public static List<float> dtHI2 = new List<float>(); public static List<float> dtHI3 = new List<float>(); public static List<float> dtRE1 = new List<float>(); public static List<float> dtRE2 = new List<float>(); public static List<float> dtRE3 = new List<float>(); public static List<float> dtREsum = new List<float>(); public static List<float> dtRECons1 = new List<float>(); public static List<float> dtRECons2 = new List<float>(); public static List<float> dtRECons3 = new List<float>(); public static List<float> dtREConssum = new List<float>(); public static List<float> dtREDe11 = new List<float>(); public static List<float> dtREDe12 = new List<float>(); public static List<float> dtREDe13 = new List<float>(); public static List<float> dtREDelsum = new List<float>(); public static List<float> dtAE1 = new List<float>(); public static List<float> dtAE2 = new List<float>(); public static List<float> dtAE3 = new List<float>(); public static List<float> dtAEsum = new List<float>(); public static List<float> dtRaE1 = new List<float>(); public static List<float> dtRaE2 = new List<float>(); public static List<float> dtRaE3 = new List<float>(); public static List<float> dtRaEsum = new List<float>(); public static List<float> dtRaEInd1 = new List<float>(); public static List<float> dtRaEInd2 = new List<float>(); public static List<float> dtRaEInd3 = new List<float>(); public static List<float> dtRaEIndsum = new List<float>(); public static List<float> dtRaECap1 = new List<float>(); public static List<float> dtRaECap2 = new List<float>(); public static List<float> dtRaECap3 = new List<float>(); public static List<float> dtRaECapsum = new List<float>(); </pre>	<p>Ορίζονται δημόσιες λίστες από floats (μεταβλητές κυλιόμενης υποδιαστολής) οποία θα μπει η αντίστοιχη στήλη από το ερώτημα της βάσης δεδομένων</p>
89	

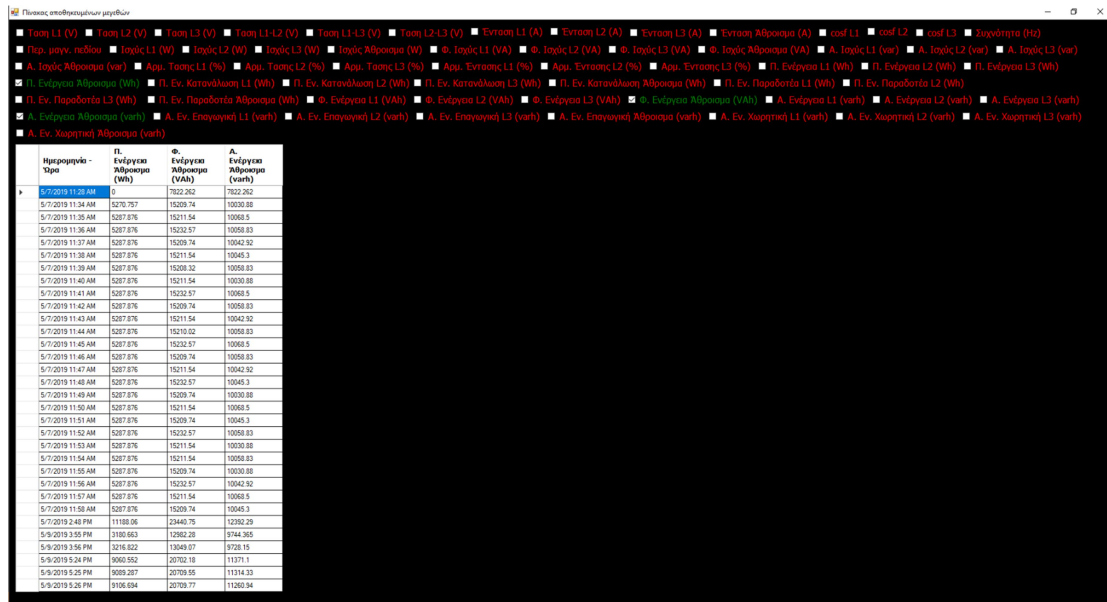
Μεταβλητές - Μέθοδοι – Events	Περιγραφή
<pre>private void Xquery_Click(object sender, EventArgs e)</pre>	<p>Αδειάζει τις ανωτέρω λίστες (DateTime και floats). Ελέγχει αν υπάρχει ενεργή σύνδεση με την τοπική βάση δεδομένων, εάν δεν υπάρχει την δημιουργεί. Μεταφέρει τα δεδομένα της απάντησης από την βάση δεδομένων, από τον answerTable, στις αντίστοιχες λίστες. Εμφανίζει τα κουμπιά:</p> <ul style="list-style-type: none"> • Εμφάνιση σε γράφημα. • Εμφάνιση σε πίνακα. • Στατιστικά στοιχεία. • Εξαγωγή σε αρχείο.
<pre>private void showGridBtn_Click(object sender, EventArgs e)</pre>	<p>Εμφανίζει το παράθυρο Πίνακας αποθηκευμένων μεγεθών.</p>
<pre>private void DataQuery_Load(object sender, EventArgs e)</pre>	<p>Ορίζει το αρχικό χρονικό διάστημα από την αρχή του μήνα.</p>
<pre>private void showGraphBtn_Click(object sender, EventArgs e)</pre>	<p>Εμφανίζει το παράθυρο Γράφημα αποθηκευμένων μεγεθών.</p>
<pre>private void showStatisticsBtn_Click(object sender, EventArgs e)</pre>	<p>Εμφανίζει το παράθυρο Στατιστικά στοιχεία.</p>

Μεταβλητές - Μέθοδοι – Events	Περιγραφή
<pre>private async void XportBtn_Click(object sender, EventArgs e)</pre>	<p>Εμφανίζει το παράθυρο αποθήκευσης των windows:</p>  <p>Στο παράθυρο αυτό ορίζετε η διαδρομή και το όνομα του αρχείου που θα εξαχθεί. Ο τύπος του αρχείου είναι .csv.</p>

8.7. Πίνακας αποθηκευμένων μεγεθών.

Σε αυτό το παράθυρο εμφανίζει, σε μορφή πίνακα, στοιχεία από την βάση δεδομένων. Το ερώτημα προς την βάση δεδομένων εκτελείται από το παράθυρο **Εύρεση μεγεθών από την βάση δεδομένων**.

8.7.1. DataGridView[Design].



Στοιχείο παραθύρου	Ενεργειακό μέγεθος	Περιγραφή
Τάση L1 (V)	Τάση L1.	Επιλέγοντας κάποια από αυτές τις ετικέτες, εμφανίζονται οι αντίστοιχες τιμές τις απάντησης του ερωτήματος προς την βάση δεδομένων, στον πίνακα. Όταν ένα μέγεθος εμφανίζεται στον πίνακα, η ετικέτα έχει χρώμα πράσινο, διαφορετικά έχει χρώμα κόκκινο.
Τάση L2 (V)	Τάση L2.	
Τάση L3 (V)	Τάση L3.	
Τάση L1-L2 (V)	Τάση L1-L2.	
Τάση L1-L3 (V)	Τάση L1-L3.	
Τάση L2-L3 (V)	Τάση L2-L3.	
Ένταση L1 (A)	Ένταση L1.	
Ένταση L2 (A)	Ένταση L2.	
Ένταση L3 (A)	Ένταση L3.	
Ένταση Άθροισμα (A)	Άθροισμα έντασης.	
cosφ L1	Συντελεστής ισχύος L1	
cosφ L2	Συντελεστής ισχύος L2	
cosφ L3	Συντελεστής ισχύος L3	
Συχνότητα (Hz)	Συχνότητα.	
Περ. μαγν. πεδίου	Περιστροφή μαγνητικού πεδίου	
Ισχύς L1 (W)	Ισχύς L1.	
Ισχύς L2 (W)	Ισχύς L2.	
Ισχύς L3 (W)	Ισχύς L3.	
Ισχύς Άθροισμα (W)	Άθροισμα ισχύος.	
Φ. Ισχύς L1 (VA)	Φαινόμενη ισχύς L1.	
Φ. Ισχύς L2 (VA)	Φαινόμενη ισχύς L2.	
Φ. Ισχύς L3 (VA)	Φαινόμενη ισχύς L3.	
Φ. Ισχύς Άθροισμα (VA)	Φαινόμενη ισχύς Άθροισμα.	

Στοιχείο παραθύρου	Ενεργειακό μέγεθος	Περιγραφή
A. Ισχύς L1 (var)	Άεργη ισχύς L1.	
A. Ισχύς L2 (var)	Άεργη ισχύς L2.	
A. Ισχύς L3(var)	Άεργη ισχύς L3.	
A. Ισχύς Άθροισμα (var)	Άθροισμα άεργης ισχύος.	
Αρμ. Τάσης L1 (%)	Αρμονική τάσης L1.	
Αρμ. Τάσης L2 (%)	Αρμονική τάσης L2.	
Αρμ. Τάσης L3 (%)	Αρμονική τάσης L3.	
Αρμ. Έντασης L1 (%)	Αρμονική έντασης L1.	
Αρμ. Έντασης L2 (%)	Αρμονική έντασης L2.	
Αρμ. Έντασης L3 (%)	Αρμονική έντασης L3.	
Π. Ενέργεια L1 (Wh)	Πραγματική ενέργεια L1	
Π. Ενέργεια L2 (Wh)	Πραγματική ενέργεια L2.	
Π. Ενέργεια L3 (Wh)	Πραγματική ενέργεια L3.	
Π. Ενέργεια Άθροισμα (Wh)	Άθροισμα πραγματικής ενέργειας.	
Π. Εν. Κατανάλωση L1 (Wh)	Κατανάλωση πραγματικής ενέργειας L1.	
Π. Εν. Κατανάλωση L2 (Wh)	Κατανάλωση πραγματικής ενέργειας L2.	
Π. Εν. Κατανάλωση L3 (Wh)	Κατανάλωση πραγματικής ενέργειας L3.	
Π. Εν. Κατανάλωση Άθροισμα (Wh)	Άθροισμα κατανάλωσης πραγματικής ενέργειας L1.	
Π. Εν. Παραδοτέα L1 (Wh)	Παραδοτέα πραγματική ενέργεια L1.	
Π. Εν. Παραδοτέα L2 (Wh)	Παραδοτέα πραγματική ενέργεια L2.	
Π. Εν. Παραδοτέα L3 (Wh)	Παραδοτέα πραγματική ενέργεια L3.	
Π. Εν. Παραδοτέα Άθροισμα (Wh)	Άθροισμα παραδοτέας πραγματικής ενέργειας.	

Στοιχείο παραθύρου	Ενεργειακό μέγεθος	Περιγραφή
Φ. Ενέργεια L1 (VAh)	Φαινόμενη ενέργεια L1.	
Φ. Ενέργεια L2 (VAh)	Φαινόμενη ενέργεια L2.	
Φ. Ενέργεια L3 (VAh)	Φαινόμενη ενέργεια L3.	
Φ. Ενέργεια Άθροισμα (VAh)	Άθροισμα φαινόμενης ενέργειας.	
A. Ενέργεια L1 (varh)	Άεργη ενέργεια L1.	
A. Ενέργεια L2 (varh)	Άεργη ενέργεια L2.	
A. Ενέργεια L3 (varh)	Άεργη ενέργεια L3.	
A. Ενέργεια Άθροισμα (varh)	Άθροισμα άεργης ενέργειας.	
A. Ev. Επαγωγική L1 (varh)	Άεργη επαγωγική ενέργεια L1.	
A. Ev. Επαγωγική L2 (varh)	Άεργη επαγωγική ενέργεια L2.	
A. Ev. Επαγωγική L3 (varh)	Άεργη επαγωγική ενέργεια L3.	
A. Ev. Επαγωγική Άθροισμα (varh)	Άθροισμα άεργης επαγωγικής ενέργειας.	
A. Ev. Χωρητική L1 (varh)	Άεργη χωρητική ενέργεια L1.	
A. Ev. Χωρητική L2 (varh)	Άεργη χωρητική ενέργεια L2.	
A. Ev. Χωρητική L3 (varh)	Άεργη χωρητική ενέργεια L3.	
A. Ev. Χωρητική Άθροισμα (varh)	Άθροισμα άεργης χωρητικής ενέργειας.	

8.7.1. DataGrid.cs

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;
```

```

namespace MODBUS
{
    public partial class DataGrid : Form
    {
        public DataGrid()
        {
            InitializeComponent();
        }

        private void DataGrid_Load(object sender, EventArgs e)
        {
            dataDGV.DataSource = DataQuery.answerDataTable;

            dataDGV.Size = new Size((this.Size.Width - dataDGV.Location.X - 30),
            (this.Size.Height - 50 - dataDGV.Location.Y));
        }

        private void dataDGV_DataBindingComplete(object sender,
        DataGridViewBindingCompleteEventArgs e)
        {
            dataDGV.Columns[0].HeaderText = "Ημερομηνία - Ώρα";
            dataDGV.Columns[1].HeaderText = "Ταση L1 (V)";
            dataDGV.Columns[2].HeaderText = "Ταση L2 (V)";
            dataDGV.Columns[3].HeaderText = "Ταση L3 (V)";
            dataDGV.Columns[4].HeaderText = "Ταση L1-L2 (V)";
            dataDGV.Columns[5].HeaderText = "Ταση L1-L3 (V)";
            dataDGV.Columns[6].HeaderText = "Ταση L2-L3 (V)";
            dataDGV.Columns[7].HeaderText = "Ένταση L1 (A)";
            dataDGV.Columns[8].HeaderText = "Ένταση L2 (A)";
            dataDGV.Columns[9].HeaderText = "Ένταση L3 (A)";
            dataDGV.Columns[10].HeaderText = "Ένταση Άθροισμα (A)";
            dataDGV.Columns[11].HeaderText = "cosf L1";
            dataDGV.Columns[12].HeaderText = "cosf L2";
            dataDGV.Columns[13].HeaderText = "cosf L3";
            dataDGV.Columns[14].HeaderText = "Συχνότητα (Hz)";
            dataDGV.Columns[15].HeaderText = "Περ. μαγν. πεδίου";
            dataDGV.Columns[16].HeaderText = "Ισχύς L1 (W)";
            dataDGV.Columns[17].HeaderText = "Ισχύς L2 (W)";
            dataDGV.Columns[18].HeaderText = "Ισχύς L3 (W)";
            dataDGV.Columns[19].HeaderText = "Ισχύς Άθροισμα (W)";
            dataDGV.Columns[20].HeaderText = "Φ. Ισχύς L1 (VA)";
            dataDGV.Columns[21].HeaderText = "Φ. Ισχύς L2 (VA)";
            dataDGV.Columns[22].HeaderText = "Φ. Ισχύς L3 (VA)";
            dataDGV.Columns[23].HeaderText = "Φ. Ισχύς Άθροισμα (VA)";
            dataDGV.Columns[24].HeaderText = "Α. Ισχύς L1 (var)";
            dataDGV.Columns[25].HeaderText = "Α. Ισχύς L2 (var)";
            dataDGV.Columns[26].HeaderText = "Α. Ισχύς L3(var)";
            dataDGV.Columns[27].HeaderText = "Α. Ισχύς Άθροισμα (var)";
            dataDGV.Columns[28].HeaderText = "Αρμ. Τασης L1 (%)";
            dataDGV.Columns[29].HeaderText = "Αρμ. Τασης L2 (%)";
            dataDGV.Columns[30].HeaderText = "Αρμ. Τασης L3 (%)";
            dataDGV.Columns[31].HeaderText = "Αρμ. Έντασης L1 (%)";
            dataDGV.Columns[32].HeaderText = "Αρμ. Έντασης L2 (%)";
            dataDGV.Columns[33].HeaderText = "Αρμ. Έντασης L3 (%)";
            dataDGV.Columns[34].HeaderText = "Π. Ενέργεια L1 (Wh)";
            dataDGV.Columns[35].HeaderText = "Π. Ενέργεια L2 (Wh)";
            dataDGV.Columns[36].HeaderText = "Π. Ενέργεια L3 (Wh)";
            dataDGV.Columns[37].HeaderText = "Π. Ενέργεια Άθροισμα (Wh)";
            dataDGV.Columns[38].HeaderText = "Π. Εν. Κατανάλωση L1 (Wh)";
        }
    }
}

```



```

dataDGV.Columns[39].HeaderText = "Π. Εν. Κατανάλωση L2 (Wh)";
dataDGV.Columns[40].HeaderText = "Π. Εν. Κατανάλωση L3 (Wh)";
dataDGV.Columns[41].HeaderText = "Π. Εν. Κατανάλωση Άθροισμα (Wh)";
dataDGV.Columns[42].HeaderText = "Π. Εν. Παραδοτέα L1 (Wh)";
dataDGV.Columns[43].HeaderText = "Π. Εν. Παραδοτέα L2 (Wh)";
dataDGV.Columns[44].HeaderText = "Π. Εν. Παραδοτέα L3 (Wh)";
dataDGV.Columns[45].HeaderText = "Π. Εν. Παραδοτέα Άθροισμα (Wh)";
dataDGV.Columns[46].HeaderText = "Φ. Ενέργεια L1 (VAh)";
dataDGV.Columns[47].HeaderText = "Φ. Ενέργεια L2 (VAh)";
dataDGV.Columns[48].HeaderText = "Φ. Ενέργεια L3 (VAh)";
dataDGV.Columns[49].HeaderText = "Φ. Ενέργεια Άθροισμα (VAh)";
dataDGV.Columns[50].HeaderText = "Α. Ενέργεια L1 (varh)";
dataDGV.Columns[51].HeaderText = "Α. Ενέργεια L2 (varh)";
dataDGV.Columns[52].HeaderText = "Α. Ενέργεια L3 (varh)";
dataDGV.Columns[53].HeaderText = "Α. Ενέργεια Άθροισμα (varh)";
dataDGV.Columns[54].HeaderText = "Α. Εν. Επαγωγική L1 (varh)";
dataDGV.Columns[55].HeaderText = "Α. Εν. Επαγωγική L2 (varh)";
dataDGV.Columns[56].HeaderText = "Α. Εν. Επαγωγική L3 (varh)";
dataDGV.Columns[57].HeaderText = "Α. Εν. Επαγωγική Άθροισμα (varh)";
dataDGV.Columns[58].HeaderText = "Α. Εν. Χωρητική L1 (varh)";
dataDGV.Columns[59].HeaderText = "Α. Εν. Χωρητική L2 (varh)";
dataDGV.Columns[60].HeaderText = "Α. Εν. Χωρητική L3 (varh)";
dataDGV.Columns[61].HeaderText = "Α. Εν. Χωρητική Άθροισμα (varh)";

for (int z = 1; z < dataDGV.ColumnCount; z++)
{
    if ((z == 37) || (z == 49) || (z == 53))
        dataDGV.Columns[z].Visible = true;
    else
        dataDGV.Columns[z].Visible = false;
}

private void checkBoxSelection_CheckedChanged(object sender, EventArgs e)
{
    CheckBox current = sender as CheckBox;

    dataDGV.Columns[current.TabIndex].Visible = current.Checked;

    if (current.Checked)
        current.ForeColor = Color.Green;
    else
        current.ForeColor = Color.Red;
}
}
}

```

Μεταβλητές - Μέθοδοι – Events	Περιγραφή
<code>public DataGrid()</code>	Αρχικοποίηση παραθύρου.
<code>private void DataGrid_Load(object sender, EventArgs e)</code>	Ορισμός ως πηγής δεδομένων του πίνακα, η μεταβλητή answerDataTable που ορίζεται στο παράθυρο Εύρεση μεγεθών από την βάση δεδομένων .

Μεταβλητές - Μέθοδοι – Events	Περιγραφή
<pre>private void dataDGV_DataBindingComplete(object sender, DataGridViewBindingCompleteEventArgs e)</pre>	<p>Όταν ολοκληρωθεί η φόρτωση δεδομένων στον πίνακα, αλλάζει τους τίτλους των στηλών, για να εμφανίζονται περιγραφές και όχι το όνομα της στήλης του ερωτήματος προς την βάση δεδομένων. Αποκρύπτει όλες τις στήλες εκτός:</p> <ol style="list-style-type: none"> 1. Π. Ενέργεια Άθροισμα (Wh) 2. Φ. Ενέργεια Άθροισμα (VAh) 3. Α. Ισχύς Άθροισμα (var)
<pre>private void checkBoxSelection_CheckedChanged(object sender, EventArgs e)</pre>	<p>Όταν πατηθεί κάποια από τις ετικέτες του παραθύρου που δεν είναι επιλεγμένη, επιλέγει την αντίστοιχη ετικέτα (object sender), εμφανίζει το αντίστοιχο μέγεθος στον πίνακα και αλλάζει το χρώμα της ετικέτας σε πράσινο. Σε αντίθετη περίπτωση, αποεπιλέγει την αντίστοιχη ετικέτα (object sender), αποκρύπτει το αντίστοιχο μέγεθος από τον πίνακα και αλλάζει το χρώμα της ετικέτας σε κόκκινο.</p>

8.8. Στατιστικά στοιχεία

Σε αυτό το παράθυρο εμφανίζονται στατικά στοιχεία, από τα στοιχεία που είναι αποθηκευμένα στην βάση δεδομένων.

8.8.1. StatisticsData.cs[Design].

Μετρούμενο μέγεθος	Μέση τιμή	Ελάχιστη τιμή	Μέγιστη τιμή	Μέση αύξηση	Ελάχιστη αύξηση	Μέγιστη αύξηση
Τάση L1(V)	217.435	121.6649	222.5737			
Τάση L2 (V)	217.5533	121.1522	223.953			
Τάση L3 (V)	217.8159	124.0141	222.8963			
Τάση L1-L2 (V)	376.3594	211.2742	387.0042			
Τάση L1-L3 (V)	375.758	213.4518	386.3234			
Τάση L2-L3 (V)	375.5885	207.6344	385.1371			
Ένταση L1 (A)	10.29433	5.235111	11.7442			
Ένταση L2 (A)	10.29433	5.235111	11.7442			
Ένταση L3 (A)	10.12652	5.257333	11.20525			
Ένταση Άθροισμα (A)	30.98526	18.51778	35.2532			
cosφ L1	0.6260284	0.2863333	0.7654285			
cosφ L2	0.6451402	0.2794444	0.79575			
cosφ L3	0.6426156	0.3902222	0.87925			
Συχνότητα (Hz)	49.506	27.92733	50.69366			
Περ. μαγν. πεδίου	0.02400686	-0.7506	0.5913333			
Ισχύς L1 (W)	1429.235	465.5483	1843.318			
Ισχύς L2 (W)	1507.376	466.8663	1993.949			
Ισχύς L3 (W)	1446.724	743.5612	1889.022			
Ισχύς Άθροισμα (W)	4383.237	1686.084	5000.65			
Φ. Ισχύς L1 (VA)	2264.417	1153.766	2566.653			
Φ. Ισχύς L2 (VA)	2320.304	1406.155	2764.48			
Φ. Ισχύς L3 (VA)	2230.372	1171.534	2480.287			
Φ. Ισχύς Άθροισμα (VA)	6813.015	3724.624	7599.979			
Α. Ισχύς L1 (var)	833.8978	548.065	1326.749			
Α. Ισχύς L2 (var)	815.5146	413.1357	1268.593			
Α. Ισχύς L3 (var)	785.7732	357.9515	994.6758			
Α. Ισχύς Άθροισμα (var)	2432.668	2045.497	3188.993			
Αρμ. Τάσης L1 (%)	7.547304	3.950667	9.893714			
Αρμ. Τάσης L2 (%)	7.638092	5.861625	10.001			
Αρμ. Τάσης L3 (%)	7.729388	4.383889	10.15425			
Αρμ. Έντασης L1 (%)	7.356416	3.099222	9.68225			
Αρμ. Έντασης L2 (%)	7.800291	4.197667	9.57525			
Αρμ. Έντασης L3 (%)	7.380307	3.866111	10.51			
Π. Ενέργεια L1 (Wh)				63.93934	-3155.135	2272.241
Π. Ενέργεια L2 (Wh)				69.24402	-2993.243	2338.019
Π. Ενέργεια L3 (Wh)				96.71056	-1629.1	1991.091
Π. Ενέργεια Άθροισμα (Wh)				238.9859	-8007.396	5900.184
Π. Εν. Κατανάλωση L1 (Wh)				57.14797	-1899.245	1643.34
Π. Εν. Κατανάλωση L2 (Wh)				62.95803	-1794.382	1202.403
Π. Εν. Κατανάλωση L3 (Wh)				86.36658	-878.951	1105.458
Π. Εν. Κατανάλωση Άθροισμα (Wh)				215.3796	-4637.499	3826.24
Π. Εν. Παραδοτέα L1 (Wh)				6.974293	-1290.516	1107.027
Π. Εν. Παραδοτέα L2 (Wh)				7.410171	-1210.241	1166.121
Π. Εν. Παραδοτέα L3 (Wh)				9.714098	-1141.856	958.546
Π. Εν. Παραδοτέα Άθροισμα (Wh)				23.5249	-3314.97	3121.981
Φ. Ενέργεια L1 (VAh)				119.0426	-3806.637	2888.414
Φ. Ενέργεια L2 (VAh)				89.19954	-3173.534	2768.65
Φ. Ενέργεια L3 (VAh)				108.4638	-3123.154	2544.885
Φ. Ενέργεια Άθροισμα (VAh)				295.5636	-10458.47	8231.01
Α. Ενέργεια L1 (varh)				46.26822	-717.4761	993.571
Α. Ενέργεια L2 (varh)				17.7109	-1878.07	867.4341
Α. Ενέργεια L3 (varh)				0.8165373	-1814.141	940.457
Α. Ενέργεια Άθροισμα (varh)				64.80727	-3392.647	2576.969

Στοιχείο πίνακα	Ενεργειακό μέγεθος	Στατιστική επεξεργασία
Τάση L1 (V)	Τάση L1.	1. Μέση τιμή 2. Ελάχιστη τιμή. 3. Μέγιστη τιμή.
Τάση L2 (V)	Τάση L2.	
Τάση L3 (V)	Τάση L3.	
Τάση L1-L2 (V)	Τάση L1-L2.	
Τάση L1-L3 (V)	Τάση L1-L3.	
Τάση L2-L3 (V)	Τάση L2-L3.	
Ένταση L1 (A)	Ένταση L1.	
Ένταση L2 (A)	Ένταση L2.	
Ένταση L3 (A)	Ένταση L3.	
Ένταση Άθροισμα (A)	Άθροισμα έντασης.	
cosφ L1	Συντελεστής ισχύος L1	
cosφ L2	Συντελεστής ισχύος L2	
cosφ L3	Συντελεστής ισχύος L3	

Στοιχείο πίνακα	Ενεργειακό μέγεθος	Στατιστική επεξεργασία
Συχνότητα (Hz)	Συχνότητα.	
Περ. μαγν. πεδίου	Περιστροφή μαγνητικού πεδίου	
Ισχύς L1 (W)	Ισχύς L1.	
Ισχύς L2 (W)	Ισχύς L2.	
Ισχύς L3 (W)	Ισχύς L3.	
Ισχύς Άθροισμα (W)	Άθροισμα ισχύος.	
Φ. Ισχύς L1 (VA)	Φαινόμενη ισχύς L1.	
Φ. Ισχύς L2 (VA)	Φαινόμενη ισχύς L2.	
Φ. Ισχύς L3 (VA)	Φαινόμενη ισχύς L3.	
Φ. Ισχύς Άθροισμα (VA)	Φαινόμενη ισχύς Άθροισμα.	
A. Ισχύς L1 (var)	Άεργη ισχύς L1.	
A. Ισχύς L2 (var)	Άεργη ισχύς L2.	
A. Ισχύς L3 (var)	Άεργη ισχύς L3.	
A. Ισχύς Άθροισμα (var)	Άθροισμα άεργης ισχύος.	
Αρμ. Τάσης L1 (%)	Αρμονική τάσης L1.	
Αρμ. Τάσης L2 (%)	Αρμονική τάσης L2.	
Αρμ. Τάσης L3 (%)	Αρμονική τάσης L3.	
Αρμ. Έντασης L1 (%)	Αρμονική έντασης L1.	
Αρμ. Έντασης L2 (%)	Αρμονική έντασης L2.	
Αρμ. Έντασης L3 (%)	Αρμονική έντασης L3.	
Π. Ενέργεια L1 (Wh)	Πραγματική ενέργεια L1	
Π. Ενέργεια L2 (Wh)	Πραγματική ενέργεια L2.	
Π. Ενέργεια L3 (Wh)	Πραγματική ενέργεια L3.	
Π. Ενέργεια Άθροισμα (Wh)	Άθροισμα πραγματικής ενέργειας.	1. Μέση αύξηση.
Π. Εν. Κατανάλωση L1 (Wh)	Κατανάλωση πραγματικής ενέργειας L1.	2. Ελάχιστη αύξηση.
Π. Εν. Κατανάλωση L2 (Wh)	Κατανάλωση πραγματικής ενέργειας L2.	3. Μεγίστη αύξηση.
Π. Εν. Κατανάλωση L3 (Wh)	Κατανάλωση πραγματικής ενέργειας L3.	

Στοιχείο πίνακα	Ενεργειακό μέγεθος	Στατιστική επεξεργασία
Π. Εν. Κατανάλωση Άθροισμα (Wh)	Άθροισμα κατανάλωσης πραγματικής ενέργειας L1.	
Π. Εν. Παραδοτέα L1 (Wh)	Παραδοτέα πραγματική ενέργεια L1.	
Π. Εν. Παραδοτέα L2 (Wh)	Παραδοτέα πραγματική ενέργεια L2.	
Π. Εν. Παραδοτέα L3 (Wh)	Παραδοτέα πραγματική ενέργεια L3.	
Π. Εν. Παραδοτέα Άθροισμα (Wh)	Άθροισμα παραδοτέας πραγματικής ενέργειας.	
Φ. Ενέργεια L1 (VAh)	Φαινόμενη ενέργεια L1.	
Φ. Ενέργεια L2 (VAh)	Φαινόμενη ενέργεια L2.	
Φ. Ενέργεια L3 (VAh)	Φαινόμενη ενέργεια L3.	
Φ. Ενέργεια Άθροισμα (VAh)	Άθροισμα φαινόμενης ενέργειας.	
Α. Ενέργεια L1 (varh)	Άεργη ενέργεια L1.	
Α. Ενέργεια L2 (varh)	Άεργη ενέργεια L2.	
Α. Ενέργεια L3 (varh)	Άεργη ενέργεια L3.	
Α. Ενέργεια Άθροισμα (varh)	Άθροισμα άεργης ενέργειας.	
Α. Εν. Επαγωγική L1 (varh)	Άεργη επαγωγική ενέργεια L1.	
Α. Εν. Επαγωγική L2 (varh)	Άεργη επαγωγική ενέργεια L2.	
Α. Εν. Επαγωγική L3 (varh)	Άεργη επαγωγική ενέργεια L3.	
Α. Εν. Επαγωγική Άθροισμα (varh)	Άθροισμα άεργης επαγωγικής ενέργειας.	
Α. Εν. Χωρητική L1 (varh)	Άεργη χωρητική ενέργεια L1.	
Α. Εν. Χωρητική L2 (varh)	Άεργη χωρητική ενέργεια L2.	
Α. Εν. Χωρητική L3 (varh)	Άεργη χωρητική ενέργεια L3.	

Στοιχείο πίνακα	Ενεργειακό μέγεθος	Στατιστική επεξεργασία
A. Εν. Χωρητική Άθροισμα (varh)	Άθροισμα άεργης χωρητικής ενέργειας.	

8.8.2 StatisticsData.cs.

```

using System;
using System.Collections;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;

namespace MODBUS
{
    public partial class StatistisData : Form
    {
        public StatistisData()
        {
            InitializeComponent();
        }

        private void StatistisData_Load(object sender, EventArgs e)
        {
            // statsGV.Rows.RemoveAt(0);
            fillDataGrid();
        }

        void fillDataGrid()
        {
            addStatRow(false, 0, "Ταση L1(V)", DataQuery.dtV1);
            addStatRow(false, 1, "Ταση L2 (V)", DataQuery.dtV2);
            addStatRow(false, 2, "Ταση L3 (V)", DataQuery.dtV3);
            addStatRow(false, 3, "Ταση L1-L2 (V)", DataQuery.dtV12);
            addStatRow(false, 4, "Ταση L1-L3 (V)", DataQuery.dtV13);
            addStatRow(false, 5, "Ταση L2-L3 (V)", DataQuery.dtV23);
            addStatRow(false, 6, "Ένταση L1 (A)", DataQuery.dtI1);
            addStatRow(false, 7, "Ένταση L2 (A)", DataQuery.dtI2);
            addStatRow(false, 8, "Ένταση L3 (A)", DataQuery.dtI3);
            addStatRow(false, 9, "Ένταση Άθροισμα (A)", DataQuery.dtIsum);
            addStatRow(false, 10, "cosf L1", DataQuery.dtcosf1);
            addStatRow(false, 11, "cosf L2", DataQuery.dtcosf2);
            addStatRow(false, 12, "cosf L3", DataQuery.dtcosf3);
            addStatRow(false, 13, "Συχνότητα (Hz)", DataQuery.dtfrequency);
            addStatRow(false, 14, "Περ. μαγν. πεδίου", DataQuery.dtfieldRotation);
            addStatRow(false, 15, "Ισχύς L1 (W)", DataQuery.dtP1);
            addStatRow(false, 16, "Ισχύς L2 (W)", DataQuery.dtP2);
            addStatRow(false, 17, "Ισχύς L3 (W)", DataQuery.dtP3);
            addStatRow(false, 18, "Ισχύς Άθροισμα (W)", DataQuery.dtPsum);
        }
    }
}

```

```

        addStatRow(false, 19, "Φ. Ισχύς L1 (VA)", DataQuery.dtA1);
        addStatRow(false, 20, "Φ. Ισχύς L2 (VA)", DataQuery.dtA2);
        addStatRow(false, 21, "Φ. Ισχύς L3 (VA)", DataQuery.dtA3);
        addStatRow(false, 22, "Φ. Ισχύς Άθροισμα (VA)", DataQuery.dtAsum);
        addStatRow(false, 23, "Α. Ισχύς L1 (var)", DataQuery.dtR1);
        addStatRow(false, 24, "Α. Ισχύς L2 (var)", DataQuery.dtR2);
        addStatRow(false, 25, "Α. Ισχύς L3(var)", DataQuery.dtR3);
        addStatRow(false, 26, "Α. Ισχύς Άθροισμα (var)", DataQuery.dtRsum);
        addStatRow(false, 27, "Αρμ. Τάσης L1 (%)", DataQuery.dtHV1);
        addStatRow(false, 28, "Αρμ. Τάσης L2 (%)", DataQuery.dtHV2);
        addStatRow(false, 29, "Αρμ. Τάσης L3 (%)", DataQuery.dtHV3);
        addStatRow(false, 30, "Αρμ. Έντασης L1 (%)", DataQuery.dtHI1);
        addStatRow(false, 31, "Αρμ. Έντασης L2 (%)", DataQuery.dtHI2);
        addStatRow(false, 32, "Αρμ. Έντασης L3 (%)", DataQuery.dtHI3);

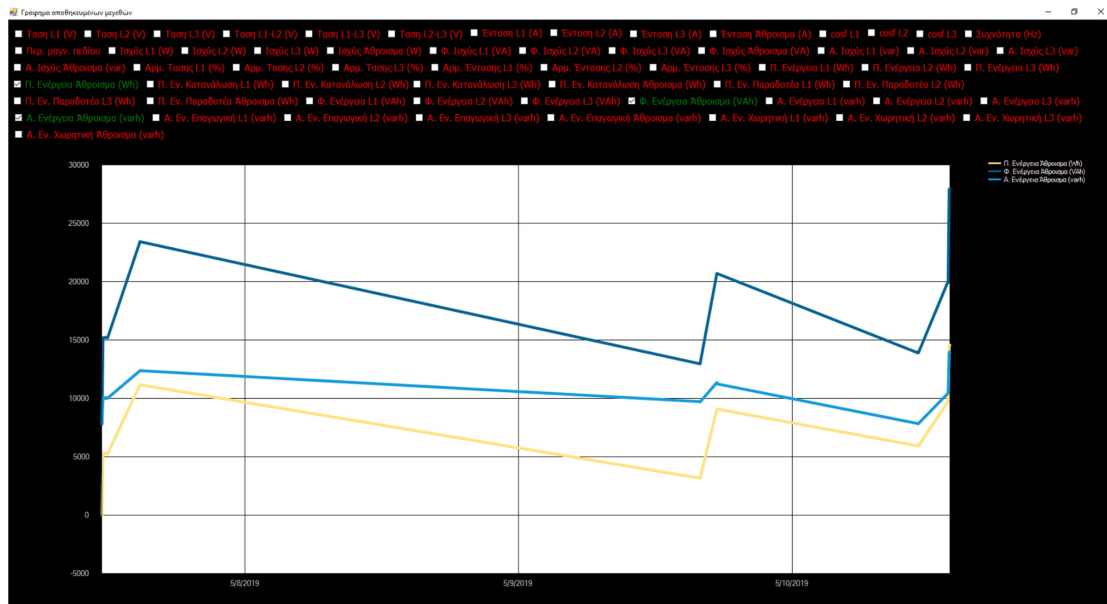
        addStatRow(true, 33, "Π. Ενέργεια L1 (Wh)", DataQuery.dtRE1);
        addStatRow(true, 34, "Π. Ενέργεια L2 (Wh)", DataQuery.dtRE2);
        addStatRow(true, 35, "Π. Ενέργεια L3 (Wh) ", DataQuery.dtRE3);
        addStatRow(true, 36, "Π. Ενέργεια Άθροισμα (Wh)", DataQuery.dtREsum);
        addStatRow(true, 37, "Π. Εν. Κατανάλωση L1 (Wh)", DataQuery.dtRECons1);
        addStatRow(true, 38, "Π. Εν. Κατανάλωση L2 (Wh)", DataQuery.dtRECons2);
        addStatRow(true, 39, "Π. Εν. Κατανάλωση L3 (Wh)", DataQuery.dtRECons3);
        addStatRow(true, 40, "Π. Εν. Κατανάλωση Άθροισμα (Wh)",
DataQuery.dtREConssum);
        addStatRow(true, 41, "Π. Εν. Παραδοτέα L1 (Wh)", DataQuery.dtREde1);
        addStatRow(true, 42, "Π. Εν. Παραδοτέα L2 (Wh)", DataQuery.dtREde2);
        addStatRow(true, 43, "Π. Εν. Παραδοτέα L3 (Wh)", DataQuery.dtREde3);
        addStatRow(true, 44, "Π. Εν. Παραδοτέα Άθροισμα (Wh)",
DataQuery.dtREde1sum);
        addStatRow(true, 45, "Φ. Ενέργεια L1 (VAh)", DataQuery.dtAE1);
        addStatRow(true, 46, "Φ. Ενέργεια L2 (VAh)", DataQuery.dtAE2);
        addStatRow(true, 47, "Φ. Ενέργεια L3 (VAh)", DataQuery.dtAE3);
        addStatRow(true, 48, "Φ. Ενέργεια Άθροισμα (VAh)", DataQuery.dtAEsum);
        addStatRow(true, 49, "Α. Ενέργεια L1 (varh)", DataQuery.dtRaE1);
        addStatRow(true, 50, "Α. Ενέργεια L2 (varh)", DataQuery.dtRaE2);
        addStatRow(true, 51, "Α. Ενέργεια L3 (varh)", DataQuery.dtRaE3);
        addStatRow(true, 52, "Α. Ενέργεια Άθροισμα (varh) ",
DataQuery.dtRaEsum);
        addStatRow(true, 53, "Α. Εν. Επαγωγική L1 (varh)", DataQuery.dtRaEInd1);
        addStatRow(true, 54, "Α. Εν. Επαγωγική L2 (varh)", DataQuery.dtRaEInd2);
        addStatRow(true, 55, "Α. Εν. Επαγωγική L3 (varh)", DataQuery.dtRaEInd3);
        addStatRow(true, 56, "Α. Εν. Επαγωγική Άθροισμα (varh)",
DataQuery.dtRaEIndsum);
        addStatRow(true, 57, "Α. Εν. Χωρητική L1 (varh)", DataQuery.dtRaECap1);
        addStatRow(true, 58, "Α. Εν. Χωρητική L2 (varh)", DataQuery.dtRaECap2);
        addStatRow(true, 59, "Α. Εν. Χωρητική L3 (varh)", DataQuery.dtRaECap3);
        addStatRow(true, 60, "Α. Εν. Χωρητική Άθροισμα (varh)",
DataQuery.dtRaECapsum);

    }
    private Statistics getStat = new Statistics();

    void addStatRow(bool isEnergy, int InToAdd, string description, List<float>
listForStats)
    {
        statsGV.Rows.Add();

        if (isEnergy)
        {
            statsGV.Rows[InToAdd].Cells[0].Value = description;
            statsGV.Rows[InToAdd].Cells[4].Value =
getStat.GetMeanIncrease(listForStats);

```

Στοιχείο παραθύρου	Ενεργειακό μέγεθος	Περιγραφή
Τάση L1 (V)	Τάση L1.	Επιλέγοντας κάποια από αυτές τις ετικέτες, εμφανίζονται οι αντίστοιχες τιμές τις απάντησης του ερωτήματος προς την βάση δεδομένων, στο γράφημα. Όταν ένα μέγεθος εμφανίζεται στο γράφημα, η ετικέτα έχει χρώμα πράσινο, διαφορετικά έχει χρώμα κόκκινο.
Τάση L2 (V)	Τάση L2.	
Τάση L3 (V)	Τάση L3.	
Τάση L1-L2 (V)	Τάση L1-L2.	
Τάση L1-L3 (V)	Τάση L1-L3.	
Τάση L2-L3 (V)	Τάση L2-L3.	
Ένταση L1 (A)	Ένταση L1.	
Ένταση L2 (A)	Ένταση L2.	
Ένταση L3 (A)	Ένταση L3.	
Ένταση Άθροισμα (A)	Άθροισμα έντασης.	
cosf L1	Συντελεστής ισχύος L1	
cosf L2	Συντελεστής ισχύος L2	
cosf L3	Συντελεστής ισχύος L3	
Συχνότητα (Hz)	Συχνότητα.	
Περ. μαγν. πεδίου	Περιστροφή μαγνητικού πεδίου	
Ισχύς L1 (W)	Ισχύς L1.	
Ισχύς L2 (W)	Ισχύς L2.	
Ισχύς L3 (W)	Ισχύς L3.	
Ισχύς Άθροισμα (W)	Άθροισμα ισχύος.	
Φ. Ισχύς L1 (VA)	Φαινόμενη ισχύς L1.	
Φ. Ισχύς L2 (VA)	Φαινόμενη ισχύς L2.	
Φ. Ισχύς L3 (VA)	Φαινόμενη ισχύς L3.	
Φ. Ισχύς Άθροισμα (VA)	Φαινόμενη ισχύς Άθροισμα.	

Στοιχείο παραθύρου	Ενεργειακό μέγεθος	Περιγραφή
A. Ισχύς L1 (var)	Άεργη ισχύς L1.	
A. Ισχύς L2 (var)	Άεργη ισχύς L2.	
A. Ισχύς L3(var)	Άεργη ισχύς L3.	
A. Ισχύς Άθροισμα (var)	Άθροισμα άεργης ισχύος.	
Αρμ. Τάσης L1 (%)	Αρμονική τάσης L1.	
Αρμ. Τάσης L2 (%)	Αρμονική τάσης L2.	
Αρμ. Τάσης L3 (%)	Αρμονική τάσης L3.	
Αρμ. Έντασης L1 (%)	Αρμονική έντασης L1.	
Αρμ. Έντασης L2 (%)	Αρμονική έντασης L2.	
Αρμ. Έντασης L3 (%)	Αρμονική έντασης L3.	
Π. Ενέργεια L1 (Wh)	Πραγματική ενέργεια L1	
Π. Ενέργεια L2 (Wh)	Πραγματική ενέργεια L2.	
Π. Ενέργεια L3 (Wh)	Πραγματική ενέργεια L3.	
Π. Ενέργεια Άθροισμα (Wh)	Άθροισμα πραγματικής ενέργειας.	
Π. Εν. Κατανάλωση L1 (Wh)	Κατανάλωση πραγματικής ενέργειας L1.	
Π. Εν. Κατανάλωση L2 (Wh)	Κατανάλωση πραγματικής ενέργειας L2.	
Π. Εν. Κατανάλωση L3 (Wh)	Κατανάλωση πραγματικής ενέργειας L3.	
Π. Εν. Κατανάλωση Άθροισμα (Wh)	Άθροισμα κατανάλωσης πραγματικής ενέργειας L1.	
Π. Εν. Παραδοτέα L1 (Wh)	Παραδοτέα πραγματική ενέργεια L1.	
Π. Εν. Παραδοτέα L2 (Wh)	Παραδοτέα πραγματική ενέργεια L2.	
Π. Εν. Παραδοτέα L3 (Wh)	Παραδοτέα πραγματική ενέργεια L3.	
Π. Εν. Παραδοτέα Άθροισμα (Wh)	Άθροισμα παραδοτέας πραγματικής ενέργειας.	

Στοιχείο παραθύρου	Ενεργειακό μέγεθος	Περιγραφή
Φ. Ενέργεια L1 (VAh)	Φαινόμενη ενέργεια L1.	
Φ. Ενέργεια L2 (VAh)	Φαινόμενη ενέργεια L2.	
Φ. Ενέργεια L3 (VAh)	Φαινόμενη ενέργεια L3.	
Φ. Ενέργεια Άθροισμα (VAh)	Άθροισμα φαινόμενης ενέργειας.	
A. Ενέργεια L1 (varh)	Άεργη ενέργεια L1.	
A. Ενέργεια L2 (varh)	Άεργη ενέργεια L2.	
A. Ενέργεια L3 (varh)	Άεργη ενέργεια L3.	
A. Ενέργεια Άθροισμα (varh)	Άθροισμα άεργης ενέργειας.	
A. Ev. Επαγωγική L1 (varh)	Άεργη επαγωγική ενέργεια L1.	
A. Ev. Επαγωγική L2 (varh)	Άεργη επαγωγική ενέργεια L2.	
A. Ev. Επαγωγική L3 (varh)	Άεργη επαγωγική ενέργεια L3.	
A. Ev. Επαγωγική Άθροισμα (varh)	Άθροισμα άεργης επαγωγικής ενέργειας.	
A. Ev. Χωρητική L1 (varh)	Άεργη χωρητική ενέργεια L1.	
A. Ev. Χωρητική L2 (varh)	Άεργη χωρητική ενέργεια L2.	
A. Ev. Χωρητική L3 (varh)	Άεργη χωρητική ενέργεια L3.	
A. Ev. Χωρητική Άθροισμα (varh)	Άθροισμα άεργης χωρητικής ενέργειας.	

8.9.2. DataChart.cs[Design].

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;
```

```

namespace MODBUS
{
    public partial class DataChart : Form
    {
        public DataChart()
        {
            InitializeComponent();
        }

        private void DataChart_Load(object sender, EventArgs e)
        {
            // chart1.Location = new Point(0, 0);
            chart1.Size = new Size((this.Size.Width), (this.Size.Height-30-
            chart1.Location.Y));

            GetSData();
        }

        private void GetSData()
        {
            for (int z =0; z<DataQuery.dtTimeStamp.Count; z++)
            {
                chart1.Series[0].Points.AddXY(DataQuery.dtTimeStamp[z],
                DataQuery.dtV1[z]);
                chart1.Series[1].Points.AddXY(DataQuery.dtTimeStamp[z],
                DataQuery.dtV2[z]);
                chart1.Series[2].Points.AddXY(DataQuery.dtTimeStamp[z],
                DataQuery.dtV3[z]);
                chart1.Series[3].Points.AddXY(DataQuery.dtTimeStamp[z],
                DataQuery.dtV12[z]);
                chart1.Series[4].Points.AddXY(DataQuery.dtTimeStamp[z],
                DataQuery.dtV13[z]);
                chart1.Series[5].Points.AddXY(DataQuery.dtTimeStamp[z],
                DataQuery.dtV23[z]);
                chart1.Series[6].Points.AddXY(DataQuery.dtTimeStamp[z],
                DataQuery.dtI1[z]);
                chart1.Series[7].Points.AddXY(DataQuery.dtTimeStamp[z],
                DataQuery.dtI2[z]);
                chart1.Series[8].Points.AddXY(DataQuery.dtTimeStamp[z],
                DataQuery.dtI3[z]);
                chart1.Series[9].Points.AddXY(DataQuery.dtTimeStamp[z],
                DataQuery.dtIsum[z]);
                chart1.Series[10].Points.AddXY(DataQuery.dtTimeStamp[z],
                DataQuery.dtcosf1[z]);
                chart1.Series[11].Points.AddXY(DataQuery.dtTimeStamp[z],
                DataQuery.dtcosf2[z]);
                chart1.Series[12].Points.AddXY(DataQuery.dtTimeStamp[z],
                DataQuery.dtcosf3[z]);
                chart1.Series[13].Points.AddXY(DataQuery.dtTimeStamp[z],
                DataQuery.dtfrequency[z]);
                chart1.Series[14].Points.AddXY(DataQuery.dtTimeStamp[z],
                DataQuery.dtfieldRotation[z]);
                chart1.Series[15].Points.AddXY(DataQuery.dtTimeStamp[z],
                DataQuery.dtP1[z]);
                chart1.Series[16].Points.AddXY(DataQuery.dtTimeStamp[z],
                DataQuery.dtP2[z]);
                chart1.Series[17].Points.AddXY(DataQuery.dtTimeStamp[z],
                DataQuery.dtP3[z]);
            }
        }
    }
}

```

```
        chart1.Series[18].Points.AddXY(DataQuery.dtTimeStamp[z],
DataQuery.dtPsum[z]);
        chart1.Series[19].Points.AddXY(DataQuery.dtTimeStamp[z],
DataQuery.dtA1[z]);
        chart1.Series[20].Points.AddXY(DataQuery.dtTimeStamp[z],
DataQuery.dtA2[z]);
        chart1.Series[21].Points.AddXY(DataQuery.dtTimeStamp[z],
DataQuery.dtA3[z]);
        chart1.Series[22].Points.AddXY(DataQuery.dtTimeStamp[z],
DataQuery.dtAsum[z]);
        chart1.Series[23].Points.AddXY(DataQuery.dtTimeStamp[z],
DataQuery.dtr1[z]);
        chart1.Series[24].Points.AddXY(DataQuery.dtTimeStamp[z],
DataQuery.dtr2[z]);
        chart1.Series[25].Points.AddXY(DataQuery.dtTimeStamp[z],
DataQuery.dtr3[z]);
        chart1.Series[26].Points.AddXY(DataQuery.dtTimeStamp[z],
DataQuery.dtrsum[z]);
        chart1.Series[27].Points.AddXY(DataQuery.dtTimeStamp[z],
DataQuery.dthv1[z]);
        chart1.Series[28].Points.AddXY(DataQuery.dtTimeStamp[z],
DataQuery.dthv2[z]);
        chart1.Series[29].Points.AddXY(DataQuery.dtTimeStamp[z],
DataQuery.dthv3[z]);
        chart1.Series[30].Points.AddXY(DataQuery.dtTimeStamp[z],
DataQuery.dthi1[z]);
        chart1.Series[31].Points.AddXY(DataQuery.dtTimeStamp[z],
DataQuery.dthi2[z]);
        chart1.Series[32].Points.AddXY(DataQuery.dtTimeStamp[z],
DataQuery.dthi3[z]);
        chart1.Series[33].Points.AddXY(DataQuery.dtTimeStamp[z],
DataQuery.dtre1[z]);
        chart1.Series[34].Points.AddXY(DataQuery.dtTimeStamp[z],
DataQuery.dtre2[z]);
        chart1.Series[35].Points.AddXY(DataQuery.dtTimeStamp[z],
DataQuery.dtre3[z]);
        chart1.Series[36].Points.AddXY(DataQuery.dtTimeStamp[z],
DataQuery.dtresum[z]);
        chart1.Series[37].Points.AddXY(DataQuery.dtTimeStamp[z],
DataQuery.dtrecons1[z]);
        chart1.Series[38].Points.AddXY(DataQuery.dtTimeStamp[z],
DataQuery.dtrecons2[z]);
        chart1.Series[39].Points.AddXY(DataQuery.dtTimeStamp[z],
DataQuery.dtrecons3[z]);
        chart1.Series[40].Points.AddXY(DataQuery.dtTimeStamp[z],
DataQuery.dtreconssum[z]);
        chart1.Series[41].Points.AddXY(DataQuery.dtTimeStamp[z],
DataQuery.dtrede11[z]);
        chart1.Series[42].Points.AddXY(DataQuery.dtTimeStamp[z],
DataQuery.dtrede12[z]);
        chart1.Series[43].Points.AddXY(DataQuery.dtTimeStamp[z],
DataQuery.dtrede13[z]);
        chart1.Series[44].Points.AddXY(DataQuery.dtTimeStamp[z],
DataQuery.dtrede1sum[z]);
        chart1.Series[45].Points.AddXY(DataQuery.dtTimeStamp[z],
DataQuery.dtae1[z]);
        chart1.Series[46].Points.AddXY(DataQuery.dtTimeStamp[z],
DataQuery.dtae2[z]);
        chart1.Series[47].Points.AddXY(DataQuery.dtTimeStamp[z],
DataQuery.dtae3[z]);
        chart1.Series[48].Points.AddXY(DataQuery.dtTimeStamp[z],
DataQuery.dtaesum[z]);
```

```

        chart1.Series[49].Points.AddXY(DataQuery.dtTimeStamp[z],
DataQuery.dtRaE1[z]);
        chart1.Series[50].Points.AddXY(DataQuery.dtTimeStamp[z],
DataQuery.dtRaE2[z]);
        chart1.Series[51].Points.AddXY(DataQuery.dtTimeStamp[z],
DataQuery.dtRaE3[z]);
        chart1.Series[52].Points.AddXY(DataQuery.dtTimeStamp[z],
DataQuery.dtRaEsum[z]);
        chart1.Series[53].Points.AddXY(DataQuery.dtTimeStamp[z],
DataQuery.dtRaEInd1[z]);
        chart1.Series[54].Points.AddXY(DataQuery.dtTimeStamp[z],
DataQuery.dtRaEInd2[z]);
        chart1.Series[55].Points.AddXY(DataQuery.dtTimeStamp[z],
DataQuery.dtRaEInd3[z]);
        chart1.Series[56].Points.AddXY(DataQuery.dtTimeStamp[z],
DataQuery.dtRaEIndsum[z]);
        chart1.Series[57].Points.AddXY(DataQuery.dtTimeStamp[z],
DataQuery.dtRaECap1[z]);
        chart1.Series[58].Points.AddXY(DataQuery.dtTimeStamp[z],
DataQuery.dtRaECap2[z]);
        chart1.Series[59].Points.AddXY(DataQuery.dtTimeStamp[z],
DataQuery.dtRaECap3[z]);
        chart1.Series[60].Points.AddXY(DataQuery.dtTimeStamp[z],
DataQuery.dtRaECapsum[z]);
    }

    for (int z =0; z < chart1.Series.Count; z++)
    {
        if ((z == 36) || (z == 48) || (z == 52))
        {
            chart1.Series[z].Enabled = true;
            chart1.Series[z].IsVisibleInLegend = true;
        }
        else
        {
            chart1.Series[z].Enabled = false;
            chart1.Series[z].IsVisibleInLegend = false;
        }
    }
}

private void checkBoxSelection_CheckedChanged(object sender, EventArgs e)
{
    CheckBox current = sender as CheckBox;

    chart1.Series[current.Text].Enabled = current.Checked;
    chart1.Series[current.Text].IsVisibleInLegend = current.Checked;

    if (current.Checked)
        current.ForeColor = Color.Green;
    else
        current.ForeColor = Color.Red;

    double maxSc = 0;
    double minSc = 0;

    for (int z = 0; z < chart1.Series.Count; z++)
    {
        if (chart1.Series[z].Enabled == true)
        {
            for (int y = 0; y<chart1.Series[z].Points.Count;y++)
            {

```

```

        if (chart1.Series[z].Points[y].YValues[0] > maxSc)
            maxSc = chart1.Series[z].Points[y].YValues[0];

        if (chart1.Series[z].Points[y].YValues[0] < minSc)
            minSc = chart1.Series[z].Points[y].YValues[0];
    }
}

chart1.ChartAreas[0].AxisY.Minimum = Math.Ceiling(minSc * 0.9);
chart1.ChartAreas[0].AxisY.Maximum = Math.Floor(maxSc * 1.1);
}
}
}
}

```

Μεταβλητές - Μέθοδοι - Events	Περιγραφή
<code>public DataChart()</code>	Αρχικοποίηση παραθύρου.
<code>private void DataGrid_Load(object sender, EventArgs e)</code>	Προσδιορισμός της θέσης και του μεγέθους του γραφήματος σε σχέση με το παράθυρο.
<code>private void GetSData()</code>	<p>Προσθέτει τα δεδομένα που έχουν οριστεί στις λίστες στοιχείων στο παράθυρο Εύρεση μεγεθών από την βάση δεδομένων. Όταν ολοκληρωθεί η φόρτωση δεδομένων στο γράφημα, αποκρύπτει όλες τις σειρές εκτός:</p> <ol style="list-style-type: none"> 1. Π. Ενέργεια Άθροισμα (Wh) 2. Φ. Ενέργεια Άθροισμα (VAh) 3. Α. Ισχύς Άθροισμα (var)

Μεταβλητές - Μέθοδοι - Events	Περιγραφή
<pre>private void checkBoxSelection_CheckedChanged(object sender, EventArgs e)</pre>	<p>Όταν πατηθεί κάποια από τις ετικέτες του παραθύρου που δεν είναι επιλεγμένη, επιλέγει την αντίστοιχη ετικέτα (object sender), εμφανίζει το αντίστοιχο μέγεθος στο γράφημα και αλλάζει το χρώμα της ετικέτας σε πράσινο. Σε αντίθετη περίπτωση, αποεπιλέγει την αντίστοιχη ετικέτα (object sender), αποκρύπτει το αντίστοιχο μέγεθος από το γράφημα και αλλάζει το χρώμα της ετικέτας σε κόκκινο.</p>

8.10. Εξομοίωση τιμών.

Σε αυτό το παράθυρο, γράφονται στοιχεία στον server εξομοίωσης.

8.10.1. SetValues.cs[Design].

Εξομοίωση τιμών																			
L1 - Voltage	215	225	223.204	L1 - Current	8	13	8.282	L2 - Voltage	215	225	223.204	L2 - Current	8	13	8.282				
L3 - Voltage	215	225	215.3	L3 - Current	8	13	12.767	L1/L2 - Voltage	370	390	375.396	Sum - Current			27.233				
L2/L3 - Voltage	370	390	382.091	Frequency	49	51	49.233	L3/L1 - Voltage	370	390	386.187								
Power 1	1141.433	A.Power 1	1824.697	R.Power 1	657.044	cosf L1-N	0.813	Power 2	1273.541	A.Power 2	2635.3	R.Power 2	1334.809	cosf L2-N	0.672				
Power 3	1594.564	A.Power 3	2748.669	R.Power 3	1138.056	cosf L3-N	0.688	Sum - Power	4005.326	Sum - A.Power	7221.849	Sum - R.Power	3149.866	Rotation field	0.024				
R.Energy 1	6774.604	A.Energy 1	11089.99	Rea.Energy 1	4143.038	Harmonic U1	3	12	7.86	R.Energy 2	6566.748	A.Energy 2	11770.9	Rea.Energy 2	5242.282	Harmonic U2	3	12	9.956
R.Energy 3	7178.099	A.Energy 3	11866.35	Sum - Rea.Energy	14958.41	Harmonic U3	3	12	7.346	Sum - R.Energy	21033.74	Sum - A.Energy	30082.32	Rea.Energy 3	5296.087	Harmonic I1	3	12	8.442
R.Energy 1 - con	4039.137	Rea.Energy 1 - ind	3657.664	Harmonic I2	3	12	10.538	R.Energy 2 - con	3929.893	Rea.Energy 2 - ind	4618.434	Harmonic I3	3	12	3.634				
R.Energy 3 - con	4295.399	Rea.Energy 3 - ind	4641.195	Sum - Rea.Energy - ind	13124.68	R.Energy 1 - del	2714.31	Rea.Energy 1 - cap	515.728										
Sum - R.Energy - con	12615.8	Rea.Energy 2 - cap	634.663	R.Energy 2 - del	2641.187	Rea.Energy 3 - cap	627.77												
R.Energy 3 - del	2857.032	Sum - Rea.Energy - cap	1848.768	Sum - R.Energy - del	8432.082														

Στα πεδία με πλαίσιο, εισάγονται η μέγιστη και η ελάχιστη τιμή για την εξομοίωση. Στα μεγέθη που δεν υπάρχουν πεδία εισαγωγής αριθμών, η τιμή τους καθορίζεται από τους αντίστοιχους τύπους:

Στοιχείο παραθύρου	Ενεργειακό μέγεθος	Ορισμός τιμής
Sum - Current	Άθροισμα έντασης.	$I_{sum} = I_{L1} + I_{L2} + I_{L3}$
Power 1	Ισχύς L1	$P = V * I * \cos\phi$
Power 2	Ισχύς L2	
Power 3	Ισχύς L3	
Sum - Power	Άθροισμα ισχύος.	$P_{sum} = P_1 + P_2 + P_3$
A.Power 1	Φαινόμενη ισχύς L1.	$P_A = V * I$
A.Power 2	Φαινόμενη ισχύς L2.	
A.Power 3	Φαινόμενη ισχύς L3.	
Sum - A.Power	Άθροισμα φαινόμενης ισχύος	$P_{Asum} = P_{A1} + P_{A2} + P_{A3}$
R.Power 1	Άεργη ισχύς L1.	$P_R = V * I * \sin\phi$
R.Power 2	Άεργη ισχύς L2.	
R.Power 3	Άεργη ισχύς L3.	
Sum - R.Power	Άθροισμα άεργης ισχύος.	$P_{Rsum} = P_{R1} + P_{R2} + P_{R3}$
R.Energy 1	Πραγματική ενέργεια L1	$E_R = (V * I * \cos\phi) / t = P / t$
R.Energy 2	Πραγματική ενέργεια L2.	
R.Energy 3	Πραγματική ενέργεια L3.	
Sum - R.Energy	Άθροισμα πραγματικής ενέργειας.	$E_{Rsum} = E_{R1} + E_{R2} + E_{R3}$
A.Energy 1	Φαινόμενη ενέργεια L1.	$E_A = (V * I) / t = P_A / t$
A.Energy 2	Φαινόμενη ενέργεια L2.	
A.Energy 3	Φαινόμενη ενέργεια L3.	
Sum - A.Energy	Άθροισμα φαινόμενης ενέργειας.	$E_{Asum} = E_{A1} + E_{A2} + E_{A3}$
Rea.Energy 1	Άεργη ενέργεια L1.	$E_{RA} = (V * I * \sin\phi) / t = P_R / t$
Rea.Energy 2	Άεργη ενέργεια L2.	
Rea.Energy 3	Άεργη ενέργεια L3.	
Sum - Rea.Energy	Άθροισμα άεργης ενέργειας.	$E_{RAsum} = E_{RA1} + E_{RA2} + E_{RA3}$

8.10.2. SetValue.cs.

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
```

```

using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;
using EasyModbus;
using exModbus;

namespace MODBUS
{
    public partial class SetValues : Form
    {
        public SetValues()
        {
            InitializeComponent();

            private simReadWriteFloat inst = new simReadWriteFloat();
            private bool firstRead = true;
            private void writeValuesTmr_Tick(object sender, EventArgs e)
            {

                inst.SimWriteFloat(Int32.Parse(l1LowLimitTbx.Text),
Int32.Parse(l1HighLimitTbx.Text), Form1.mBC, 19000,0,0);
                inst.SimWriteFloat(Int32.Parse(l2LowLimitTbx.Text),
Int32.Parse(l2HighLimitTbx.Text), Form1.mBC, 19002, 0, 0);
                inst.SimWriteFloat(Int32.Parse(l3LowLimitTbx.Text),
Int32.Parse(l3HighLimitTbx.Text), Form1.mBC, 19004, 0, 0);
                inst.SimWriteFloat(Int32.Parse(l1l2LowLimitTbx.Text),
Int32.Parse(l1l2HighLimitTbx.Text), Form1.mBC, 19006, 0, 0);
                inst.SimWriteFloat(Int32.Parse(l2l3LowLimitTbx.Text),
Int32.Parse(l2l3HighLimitTbx.Text), Form1.mBC, 19008, 0, 0);
                inst.SimWriteFloat(Int32.Parse(l3l1LowLimitTbx.Text),
Int32.Parse(l3l1HighLimitTbx.Text), Form1.mBC, 19010, 0, 0);
                inst.SimWriteFloat(Int32.Parse(l1CurrentLowLimitTbx.Text),
Int32.Parse(l1CurrentHighLimitTbx.Text), Form1.mBC, 19012, 0, 0);
                inst.SimWriteFloat(Int32.Parse(l2CurrentLowLimitTbx.Text),
Int32.Parse(l2CurrentHighLimitTbx.Text), Form1.mBC, 19014, 0, 0);
                inst.SimWriteFloat(Int32.Parse(l3CurrentLowLimitTbx.Text),
Int32.Parse(l3CurrentHighLimitTbx.Text), Form1.mBC, 19016, 0, 0);
                //current sum
                inst.SimWritePowerAndEnergy((inst.SimReadFloat(Form1.mBC, 19012) +
inst.SimReadFloat(Form1.mBC, 19014) + inst.SimReadFloat(Form1.mBC, 19016)),
Form1.mBC, 19018);

                //powers
                //real power = V*I*cosf
                inst.SimWritePowerAndEnergy((inst.SimReadFloat(Form1.mBC, 19000)*
inst.SimReadFloat(Form1.mBC, 19012)* inst.SimReadFloat(Form1.mBC, 19044)),
Form1.mBC, 19020);
                inst.SimWritePowerAndEnergy((inst.SimReadFloat(Form1.mBC, 19002) *
inst.SimReadFloat(Form1.mBC, 19014) * inst.SimReadFloat(Form1.mBC, 19046)),
Form1.mBC, 19022);
                inst.SimWritePowerAndEnergy((inst.SimReadFloat(Form1.mBC, 19004) *
inst.SimReadFloat(Form1.mBC, 19016) * inst.SimReadFloat(Form1.mBC, 19048)),
Form1.mBC, 19024);
                inst.SimWritePowerAndEnergy((inst.SimReadFloat(Form1.mBC, 19020) +
inst.SimReadFloat(Form1.mBC, 19022) + inst.SimReadFloat(Form1.mBC, 19024)),
Form1.mBC, 19026);
            }
        }
    }
}

```

```

//apparent power = V*I
inst.SimWritePowerAndEnergy((inst.SimReadFloat(Form1.mBC, 19000) *
inst.SimReadFloat(Form1.mBC, 19012)), Form1.mBC, 19028);
inst.SimWritePowerAndEnergy((inst.SimReadFloat(Form1.mBC, 19002) *
inst.SimReadFloat(Form1.mBC, 19014)), Form1.mBC, 19030);
inst.SimWritePowerAndEnergy((inst.SimReadFloat(Form1.mBC, 19004) *
inst.SimReadFloat(Form1.mBC, 19016)), Form1.mBC, 19032);
inst.SimWritePowerAndEnergy((inst.SimReadFloat(Form1.mBC, 19028) +
inst.SimReadFloat(Form1.mBC, 19030) + inst.SimReadFloat(Form1.mBC, 19032)),
Form1.mBC, 19034);

//reactive power = V*I*sinf
inst.SimWritePowerAndEnergy((inst.SimReadFloat(Form1.mBC, 19000) *
inst.SimReadFloat(Form1.mBC, 19012) * (1-inst.SimReadFloat(Form1.mBC, 19044))),
Form1.mBC, 19036);
inst.SimWritePowerAndEnergy((inst.SimReadFloat(Form1.mBC, 19002) *
inst.SimReadFloat(Form1.mBC, 19014) * (1-inst.SimReadFloat(Form1.mBC, 19046))),
Form1.mBC, 19038);
inst.SimWritePowerAndEnergy((inst.SimReadFloat(Form1.mBC, 19004) *
inst.SimReadFloat(Form1.mBC, 19016) * (1-inst.SimReadFloat(Form1.mBC, 19048))),
Form1.mBC, 19040);
inst.SimWritePowerAndEnergy((inst.SimReadFloat(Form1.mBC, 19036) +
inst.SimReadFloat(Form1.mBC, 19038) + inst.SimReadFloat(Form1.mBC, 19040)),
Form1.mBC, 19042);

//cosf
inst.SimWriteFloat(0, 1, Form1.mBC, 19044, 0, 0);
inst.SimWriteFloat(0, 1, Form1.mBC, 19046, 0, 0);
inst.SimWriteFloat(0, 1, Form1.mBC, 19048, 0, 0);

//Hz
inst.SimWriteFloat(Int32.Parse(FrequencyLowLimitTbx.Text),
Int32.Parse(FrequencyHighLimitTbx.Text), Form1.mBC, 19050, 0, 0);

//rotation field 1=right, 0=none, -1=left
inst.SimWriteFloat(-1, 1, Form1.mBC, 19052, 0, 0);

//energy
float energyPertime = 0;
if (firstRead)
{
    firstRead = false;
    energyPertime = 1;
}
else
    energyPertime = (1 / 36000);

//real
inst.SimWritePowerAndEnergy(((inst.SimReadFloat(Form1.mBC, 19020) *
energyPertime) + inst.SimReadFloat(Form1.mBC, 19054)), Form1.mBC, 19054);
inst.SimWritePowerAndEnergy(((inst.SimReadFloat(Form1.mBC, 19022) *
energyPertime) + inst.SimReadFloat(Form1.mBC, 19056)), Form1.mBC, 19056);
inst.SimWritePowerAndEnergy(((inst.SimReadFloat(Form1.mBC, 19024) *
energyPertime) + inst.SimReadFloat(Form1.mBC, 19058)), Form1.mBC, 19058);
inst.SimWritePowerAndEnergy(((inst.SimReadFloat(Form1.mBC, 19026) *
energyPertime) + inst.SimReadFloat(Form1.mBC, 19060)), Form1.mBC, 19060);

//real consumption
Random rConsumption = new Random();
float consumption = (float)rConsumption.NextDouble(); //value 0 TO 1

```

```

        if (consumption < 0.6)
            consumption = 0.6f;

        inst.SimWritePowerAndEnergy((inst.SimReadFloat(Form1.mBC, 19054) *
consumption), Form1.mBC, 19062);
        inst.SimWritePowerAndEnergy((inst.SimReadFloat(Form1.mBC, 19056) *
consumption), Form1.mBC, 19064);
        inst.SimWritePowerAndEnergy((inst.SimReadFloat(Form1.mBC, 19058) *
consumption), Form1.mBC, 19066);
        inst.SimWritePowerAndEnergy((inst.SimReadFloat(Form1.mBC, 19060) *
consumption), Form1.mBC, 19068);

        //real delivered
        inst.SimWritePowerAndEnergy((inst.SimReadFloat(Form1.mBC, 19054) * (1 -
consumption)), Form1.mBC, 19070);
        inst.SimWritePowerAndEnergy((inst.SimReadFloat(Form1.mBC, 19056) * (1 -
consumption)), Form1.mBC, 19072);
        inst.SimWritePowerAndEnergy((inst.SimReadFloat(Form1.mBC, 19058) * (1 -
consumption)), Form1.mBC, 19074);
        inst.SimWritePowerAndEnergy((inst.SimReadFloat(Form1.mBC, 19060) * (1 -
consumption)), Form1.mBC, 19076);

        //aparent
        inst.SimWritePowerAndEnergy(((inst.SimReadFloat(Form1.mBC, 19028) *
energyPertime) + inst.SimReadFloat(Form1.mBC, 19078)), Form1.mBC, 19078);
        inst.SimWritePowerAndEnergy(((inst.SimReadFloat(Form1.mBC, 19030) *
energyPertime) + inst.SimReadFloat(Form1.mBC, 19080)), Form1.mBC, 19080);
        inst.SimWritePowerAndEnergy(((inst.SimReadFloat(Form1.mBC, 19032) *
energyPertime) + inst.SimReadFloat(Form1.mBC, 19082)), Form1.mBC, 19082);
        inst.SimWritePowerAndEnergy(((inst.SimReadFloat(Form1.mBC, 19034) *
energyPertime) + inst.SimReadFloat(Form1.mBC, 19084)), Form1.mBC, 19084);

        //reactive
        inst.SimWritePowerAndEnergy(((inst.SimReadFloat(Form1.mBC, 19036) *
energyPertime) + inst.SimReadFloat(Form1.mBC, 19086)), Form1.mBC, 19086);
        inst.SimWritePowerAndEnergy(((inst.SimReadFloat(Form1.mBC, 19038) *
energyPertime) + inst.SimReadFloat(Form1.mBC, 19088)), Form1.mBC, 19088);
        inst.SimWritePowerAndEnergy(((inst.SimReadFloat(Form1.mBC, 19040) *
energyPertime) + inst.SimReadFloat(Form1.mBC, 19090)), Form1.mBC, 19090);
        inst.SimWritePowerAndEnergy(((inst.SimReadFloat(Form1.mBC, 19042) *
energyPertime) + inst.SimReadFloat(Form1.mBC, 19092)), Form1.mBC, 19092);

        //inductive factor
        Random rInd = new Random();
        float inductive = (float)rInd.NextDouble(); //value 0 TO 1

        //inductive
        inst.SimWritePowerAndEnergy((inst.SimReadFloat(Form1.mBC, 19086) *
inductive), Form1.mBC, 19094);
        inst.SimWritePowerAndEnergy((inst.SimReadFloat(Form1.mBC, 19088) *
inductive), Form1.mBC, 19096);
        inst.SimWritePowerAndEnergy((inst.SimReadFloat(Form1.mBC, 19090) *
inductive), Form1.mBC, 19098);
        inst.SimWritePowerAndEnergy((inst.SimReadFloat(Form1.mBC, 19092) *
inductive), Form1.mBC, 19100);

        //capasitive
        inst.SimWritePowerAndEnergy((inst.SimReadFloat(Form1.mBC, 19086) * (1-
inductive)), Form1.mBC, 19102);
        inst.SimWritePowerAndEnergy((inst.SimReadFloat(Form1.mBC, 19088) * (1 -
inductive)), Form1.mBC, 19104);

```

```

        inst.SimWritePowerAndEnergy((inst.SimReadFloat(Form1.mBC, 19090) * (1 -
inductive)), Form1.mBC, 19106);
        inst.SimWritePowerAndEnergy((inst.SimReadFloat(Form1.mBC, 19092) * (1 -
inductive)), Form1.mBC, 19108);

        //voltage harmonics
        inst.SimWriteFloat(Int32.Parse(HU1minTbx.Text),
Int32.Parse(HU1maxTbx.Text), Form1.mBC, 19110, 0, 0);
        inst.SimWriteFloat(Int32.Parse(HU2minTbx.Text),
Int32.Parse(HU2maxTbx.Text), Form1.mBC, 19112, 0, 0);
        inst.SimWriteFloat(Int32.Parse(HU3minTbx.Text),
Int32.Parse(HU3maxTbx.Text), Form1.mBC, 19114, 0, 0);

        //current harmonics
        inst.SimWriteFloat(Int32.Parse(HI1minTbx.Text),
Int32.Parse(HI1maxTbx.Text), Form1.mBC, 19116, 0, 0);
        inst.SimWriteFloat(Int32.Parse(HI2minTbx.Text),
Int32.Parse(HI2maxTbx.Text), Form1.mBC, 19118, 0, 0);
        inst.SimWriteFloat(Int32.Parse(HI3minTbx.Text),
Int32.Parse(HI3maxTbx.Text), Form1.mBC, 19120, 0, 0);
    }

    private void readValuesTmr_Tick(object sender, EventArgs e)
    {
        l1VValueTxt.Text = (inst.SimReadFloat(Form1.mBC, 19000)).ToString();
        l2VValueTxt.Text = (inst.SimReadFloat(Form1.mBC, 19002)).ToString();
        l3VValueTxt.Text = (inst.SimReadFloat(Form1.mBC, 19004)).ToString();
        l1l2VValueTxt.Text = (inst.SimReadFloat(Form1.mBC, 19006)).ToString();
        l2l3VValueTxt.Text = (inst.SimReadFloat(Form1.mBC, 19008)).ToString();
        l3l1VValueTxt.Text = (inst.SimReadFloat(Form1.mBC, 19010)).ToString();
        l1CValueTxt.Text = (inst.SimReadFloat(Form1.mBC, 19012)).ToString();
        l2CValueTxt.Text = (inst.SimReadFloat(Form1.mBC, 19012)).ToString();
        l3CValueTxt.Text = (inst.SimReadFloat(Form1.mBC, 19016)).ToString();
        SCValueTxt.Text = (inst.SimReadFloat(Form1.mBC, 19018)).ToString();

        P1ValueTxt.Text = (inst.SimReadFloat(Form1.mBC, 19020)).ToString();
        P2ValueTxt.Text = (inst.SimReadFloat(Form1.mBC, 19022)).ToString();
        P3ValueTxt.Text = (inst.SimReadFloat(Form1.mBC, 19024)).ToString();
        PSValueTxt.Text = (inst.SimReadFloat(Form1.mBC, 19026)).ToString();
        S1ValueTxt.Text = (inst.SimReadFloat(Form1.mBC, 19028)).ToString();
        S2ValueTxt.Text = (inst.SimReadFloat(Form1.mBC, 19030)).ToString();
        S3ValueTxt.Text = (inst.SimReadFloat(Form1.mBC, 19032)).ToString();
        SSVValueTxt.Text = (inst.SimReadFloat(Form1.mBC, 19034)).ToString();

        R1ValueTxt.Text = (inst.SimReadFloat(Form1.mBC, 19036)).ToString();
        R2ValueTxt.Text = (inst.SimReadFloat(Form1.mBC, 19038)).ToString();
        R3ValueTxt.Text = (inst.SimReadFloat(Form1.mBC, 19040)).ToString();
        RSValueTxt.Text = (inst.SimReadFloat(Form1.mBC, 19042)).ToString();

        cosfl1NTxt.Text = (inst.SimReadFloat(Form1.mBC, 19044)).ToString();
        cosfl2NTxt.Text = (inst.SimReadFloat(Form1.mBC, 19046)).ToString();
        cosfl3NTxt.Text = (inst.SimReadFloat(Form1.mBC, 19048)).ToString();

        FrequencyValueTxt.Text = (inst.SimReadFloat(Form1.mBC,
19050)).ToString();

        RotFValueTbx.Text = (inst.SimReadFloat(Form1.mBC, 19052)).ToString();

        RE1ValueTxt.Text = (inst.SimReadFloat(Form1.mBC, 19054)).ToString();
        RE2ValueTxt.Text = (inst.SimReadFloat(Form1.mBC, 19056)).ToString();
        RE3ValueTxt.Text = (inst.SimReadFloat(Form1.mBC, 19058)).ToString();
        RESumValueTxt.Text = (inst.SimReadFloat(Form1.mBC, 19060)).ToString();
    }

```

```

RE1cValueTxt.Text = (inst.SimReadFloat(Form1.mBC, 19062)).ToString();
RE2cValueTxt.Text = (inst.SimReadFloat(Form1.mBC, 19064)).ToString();
RE3cValueTxt.Text = (inst.SimReadFloat(Form1.mBC, 19066)).ToString();
RESumcValueTxt.Text = (inst.SimReadFloat(Form1.mBC, 19068)).ToString();

RE1dValueTxt.Text = (inst.SimReadFloat(Form1.mBC, 19070)).ToString();
RE2dValueTxt.Text = (inst.SimReadFloat(Form1.mBC, 19072)).ToString();
RE3dValueTxt.Text = (inst.SimReadFloat(Form1.mBC, 19074)).ToString();
RESumdValueTxt.Text = (inst.SimReadFloat(Form1.mBC, 19076)).ToString();

AE1ValueTxt.Text = (inst.SimReadFloat(Form1.mBC, 19078)).ToString();
AE2ValueTxt.Text = (inst.SimReadFloat(Form1.mBC, 19080)).ToString();
AE3ValueTxt.Text = (inst.SimReadFloat(Form1.mBC, 19082)).ToString();
AESumValueTxt.Text = (inst.SimReadFloat(Form1.mBC, 19084)).ToString();

RAE1ValueTxt.Text = (inst.SimReadFloat(Form1.mBC, 19086)).ToString();
RAE2ValueTxt.Text = (inst.SimReadFloat(Form1.mBC, 19088)).ToString();
RAE3ValueTxt.Text = (inst.SimReadFloat(Form1.mBC, 19090)).ToString();
RAESValueTxt.Text = (inst.SimReadFloat(Form1.mBC, 19092)).ToString();
RAEi1ValueTxt.Text = (inst.SimReadFloat(Form1.mBC, 19094)).ToString();
RAEi2ValueTxt.Text = (inst.SimReadFloat(Form1.mBC, 19096)).ToString();
RAEi3ValueTxt.Text = (inst.SimReadFloat(Form1.mBC, 19098)).ToString();
RAEiSValueTxt.Text = (inst.SimReadFloat(Form1.mBC, 19100)).ToString();
RAEc1ValueTxt.Text = (inst.SimReadFloat(Form1.mBC, 19102)).ToString();
RAEc2ValueTxt.Text = (inst.SimReadFloat(Form1.mBC, 19104)).ToString();
RAE3cValueTxt.Text = (inst.SimReadFloat(Form1.mBC, 19106)).ToString();
RAEScValueTxt.Text = (inst.SimReadFloat(Form1.mBC, 19108)).ToString();

HU1ValueTxt.Text = (inst.SimReadFloat(Form1.mBC, 19110)).ToString();
HU2ValueTxt.Text = (inst.SimReadFloat(Form1.mBC, 19112)).ToString();
HU3ValueTxt.Text = (inst.SimReadFloat(Form1.mBC, 19114)).ToString();
HI1ValueTxt.Text = (inst.SimReadFloat(Form1.mBC, 19116)).ToString();
HI2ValueTxt.Text = (inst.SimReadFloat(Form1.mBC, 19118)).ToString();
HI3ValueTxt.Text = (inst.SimReadFloat(Form1.mBC, 19120)).ToString();
}
}
}

```

Μεταβλητές - Μέθοδοι – Events	Περιγραφή
<code>public SetValues()</code>	Αρχικοποίηση παραθύρου.
<code>private simReadWriteFloat inst = new simReadWriteFloat()</code>	Τοπική μεταβλητή ανάγνωσης - εγγραφής που ορίζεται στο αρχείο modBusFloats
<code>private bool firstRead</code>	Τοπική μεταβλητή που αλλάζει τις παραμέτρους της ενέργειας κατά την πρώτη εγγραφή του αρχείου.
<code>private void writeValuesTmr_Tick(object sender, EventArgs e)</code>	Εγγραφή δεδομένων στον server εξομοίωσης.
<code>private void readValuesTmr_Tick(object sender, EventArgs e)</code>	Ανάγνωση δεδομένων από τον server εξομοίωσης.

8.11. SQL Database.

Στο **Microsoft SQL Server Management Studio**, δημιουργείται μία τοπική βάση δεδομένων με όνομα **thesis**. Στην βάση δημιουργείται ένας πίνακας με το όνομα **Measurments**. Συμπληρώνονται οι παρακάτω στήλες:

Column Name	Data Type	Allow Nulls
id	int	not null
DateTime	datetime	not null
V1	float	null
V2	float	null
V3	float	null
V12	float	null
V23	float	null
I1	float	null
I2	float	null
I3	float	null
Isum	float	null
cosf1	float	null
cosf2	float	null
cosf3	float	null
freaquency	float	null
fieldRotation	float	null
P1	float	null
P2	float	null
P3	float	null
Psum	float	null
A1	float	null
A2	float	null
A3	float	null
Asum	float	null
R1	float	null
R2	float	null
R3	float	null
Rsum	float	null
HV1	float	null
HV2	float	null
HV3	float	null
HI1	float	null
HI2	float	null
HI3	float	null
RE1	float	null
RE2	float	null

RE3	float	null
REsum	float	null
RECons1	float	null
RECons2	float	null
RECons3	float	null
REConssum	float	null
REDel1	float	null
REDel2	float	null
REDel3	float	null
REDelsum	float	null
AE1	float	null
AE2	float	null
AE3	float	null
AEsum	float	null
RaE1	float	null
RaE2	float	null
RaE3	float	null
RaEsum	float	null
RaEInd1	float	null
RaEInd2	float	null
RaEInd3	float	null
RaEIndsum	float	null
RaECap1	float	null
RaECap2	float	null
RaECap3	float	null
RaECapsum	float	null

9.Επίλογος

Η πτυχιακή εργασία αποτελεί το τελικό στάδιο των σπουδών μου στο τμήμα της αυτοματισμού. Η ολοκλήρωση της πτυχιακής μου εργασίας ήταν ο τελευταίος στόχος για να ολοκληρώσω τις σπουδές μου και αποτελεί μια απόδειξη ότι η φοίτηση μου στη σχολή αυτή και οι γνώσεις που αποκόμισα, θα αποτελούν σημαντικά εφόδια για το μέλλον.

Η ενασχόληση μου με την πτυχιακή εργασία μου έδωσε τη δυνατότητα να ασχοληθώ με ένα εξαιρετικά ενδιαφέρον θέμα. Η επιλογή του θέματος ήταν μια επιθυμία μου να ασχοληθώ εκτενέστερα με την στατιστική επεξεργασία ενεργειακών μεγεθών.

10. Βιβλιογραφία

- ▶ Εγχειρίδιο χρήσης αναλυτή ενέργειας:
[Janitza UMG96RM-E](#)
- ▶ Modbus:
<http://modbus.org/>
<http://www.simplymodbus.ca/FAQ.htm>
https://www.prosoft-technology.com/kb/assets/intro_modbustcp.pdf .
- ▶ EasyModbus:
<http://easymodbustcp.net/en/>
- ▶ Εξομοιωτής Modbus TCP/IP Server:
<https://sourceforge.net/projects/easymodbustcpserver/>
- ▶ [Software Toolbox Library1](#) και [Software Toolbox Library 2](#).
- ▶ SQL Server :
<https://docs.microsoft.com/en-us/sql/sql-server/sql-server-technical-documentation?view=sql-server-2017> .
- ▶ C#:
<https://docs.microsoft.com/en-us/dotnet/csharp/>.
- ▶ Visual Studio C#:
<https://docs.microsoft.com/en-us/dotnet/core/tutorials/with-visual-studio-code>.