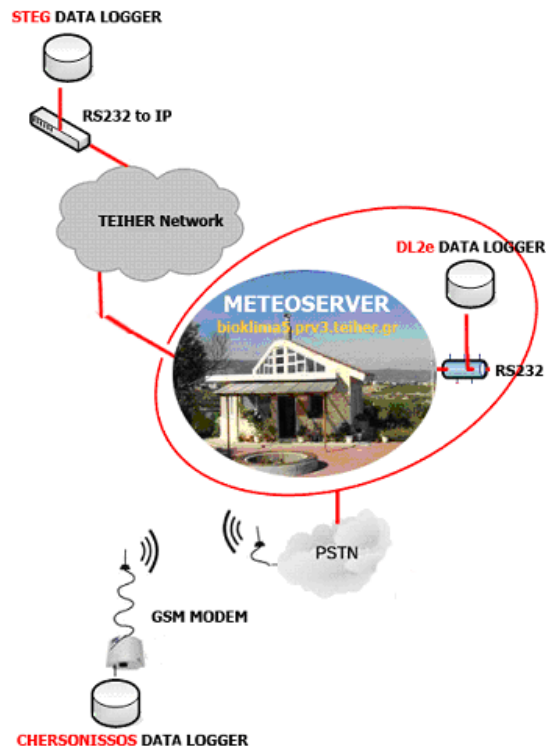


ΑΝΩΤΑΤΟ ΤΕΧΝΟΛΟΓΙΚΟ ΕΚΠΑΙΔΕΥΤΙΚΟ ΙΔΡΥΜΑ ΚΡΗΤΗΣ

Σχολή Τεχνολογικών Εφαρμογών

Τμ. Εφαρμοσμένης Πληροφορικής και Πολυμέσων.

ΑΝΑΠΤΥΞΗ ΒΑΣΗΣ ΔΕΔΟΜΕΝΩΝ ΓΙΑ ΤΗΝ ΕΠΕΞΕΡΓΑΣΙΑ ΜΕΤΕΩΡΟΛΟΓΙΚΩΝ ΔΕΔΟΜΕΝΩΝ ΑΠΟ ΔΙΚΤΥΟ ΜΕΤΕΩΡΟΛΟΓΙΚΩΝ ΣΤΑΘΜΩΝ



Σπουδαστής: Γούναρης Γεώργιος - Α.Μ.:927

Εισηγητής Καθηγητής: Βλησίδης Ανδρέας

Περίληψη

Η πιτυχιακή αυτή αναφέρεται στην σχεδίαση, ανάπτυξη και υλοποίηση του ήδη υπάρχοντος μοντέλου - που είναι εγκατεστημένο στην περιοχή ΤΕΙ Ηρακλείου- για την συλλογή, επεξεργασία και παρουσίαση σε πραγματικό χρόνο Μετεωρολογικών δεδομένων στο διαδίκτυο από 3 σταθμούς που είναι εγκατεστημένοι στο Ν. Ηρακλείου. Ειδικότερα οι 2 από αυτούς βρίσκονται στο περιοχή του ΤΕΙ Κρήτης- Ηράκλειο και ο τρίτος στην Χερσονήσο (30χμ από το κεντρικό μετεωρολογικό σταθμό- meteo server). Ο Σταθμός της Χερσονήσου βρίσκεται σε μικρό υψόμετρο από την επιφάνεια της θάλασσας.

Για την υλοποίηση αυτού του μοντέλου, εκτός της βάσης δεδομένων που ενημερώνεται σε πραγματικό χρόνο (10 λεπτά) θα δημιουργηθούν 2 διαφορετικές τεχνικές σύνδεσης των Data Logger με τον Meteo Server (bioklima5.prv3.teiher.gr) του Μετεωρολογικού σταθμού που βρίσκεται στην Βιοκλιματική κατοικία του ΤΕΙ Κρήτης/Ηράκλειο.

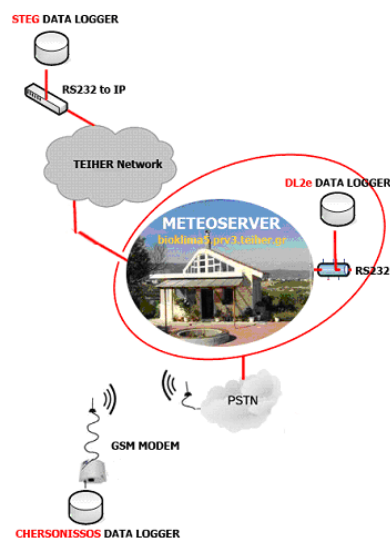
Το Block διάγραμμα του υπό ανάπτυξη συστήματος φαίνεται στο παρακάτω σχήμα. Οι τεχνικές που θα χρησιμοποιηθούν είναι:

- με χρήση εξοπλισμού τεχνολογίας GSM (CHERSONISSOS).
- με χρήση του τοπικού δικτύου Ethernet του ΤΕΙ χρησιμοποιώντας RS232- IP converters, (STEG).

Σκοπός του μοντέλου αυτού είναι η επέκταση της υπάρχουσας βάσης δεδομένων MySQL με δεδομένα και από τους άλλους σταθμούς για την παραγωγή συγκριτικών γραφημάτων, για την on-line παρουσίαση τους στο site του ΤΕΙ στο διαδίκτυο με χρήση δυναμικών γλωσσών. Ακόμη γίνονται υπολογισμοί της παραγόμενης ηλιακής και αιολικής ενέργειας καθώς και η συχνотική κατανομή της ταχύτητας του αέρα(Weibull distribution) στην κάθε περιοχή που υπάρχει data logger. Όλες αυτές οι πληροφορίες είναι πολύ ενδιαφέρουσες για τους ερευνητές και τους μηχανικούς που εργάζονται και αναπτύσσουν συστήματα στο χώρο των Ανανεώσιμων Πηγών Ενέργειας, Περιβάλλον, Γεωπονίας κ.τ.λ.

Λέξεις Κλειδιά:

Συλλογή και επεξεργασία μετεωρολογικών δεδομένα, on-line παρουσίαση στο διαδίκτυο , υπολογισμοί, GSM, PSTN.



Εικόνα 1: Το μπλοκ διάγραμμα

στην αρχή αυτής της πτυχιακής
ευχαριστώ τον Καθηγητή μου Ανδρέα Βλησίδη
που με βοήθησε όχι μόνο για την ολοκλήρωση της
αλλά και γιατί από την αρχή της πορείας μου
στη σχολή ήταν δίπλα μου

Περιεχόμενα

ΑΝΩΤΑΤΟ ΤΕΧΝΟΛΟΓΙΚΟ ΙΔΡΥΜΑ ΚΡΗΤΗΣ	1
1 Εισαγωγή- Ιστορική Αναδρομή	5
1.1 Αντικείμενο	6
1.2 Οργάνωση του τόμου	7
2 Η υπάρχουσα υποδομή του εργαστηρίου.....	8
2.1 Εξοπλισμός Μετρήσεων.....	8
2.2 Πυρανόμετρο τύπου CM 11	8
2.3 Server- Τοπολογία Δικτύου	9
2.4 Βάση Δεδομένων- Παρουσίαση Μετρήσεων.....	10
3 ΣΤΑΘΜΟΣ ΜΕΤΡΗΣΗΣ Σ.ΤΕ.Γ	12
3.1 Γενική Περιγραφή	12
3.2 Εξοπλισμός	13
3.3 Εγκατάσταση.....	17
3.4 Λογισμικό- Πρωτόκολλα	19
3.5 Η Βάση Δεδομένων	21
3.6 Γενική περιγραφή του Πρωτοκόλλου TCP/IP.....	24
4 ΣΤΑΘΜΟΣ ΜΕΤΡΗΣΗΣ ΧΕΡΣΟΝΗΣΟΣ	28
4.1 Γενική Περιγραφή	28
4.2 Εξοπλισμός	29
4.3 Εγκατάσταση.....	31
4.4 Λογισμικό- Πρωτόκολλα	32
4.5 Η Βάση Δεδομένων	33
5 ΣΥΜΠΕΡΑΣΜΑΤΑ ΚΑΙ ΙΔΕΕΣ	38
5.1 Ιδέες που απορρίψαμε	38
5.2 Συμπέρασμα.....	38
6 ΠΑΡΟΥΣΙΑΣΗ	39
6.1 Γενικά	39
6.2 Τύποι Διαγράμματος	40
6.3 Τα γραφήματα μας	42
6.4 Η Ανανεωμένη Ιστοσελίδα μας.....	45
7 ΕΡΓΑΛΕΙΑ.....	48
8 ΠΑΡΑΡΤΗΜΑΤΑ- ΚΩΔΙΚΕΣ	60
8.1 Παράρτημα I- Κώδικες Εφαρμογών	60
8.2 Παράρτημα II- Κώδικες Γραφημάτων	76
9 Βιβλιογραφία- References	83

1 Εισαγωγή- Ιστορική Αναδρομή

Για 2000 περίπου χρόνια το βασικό βοήθημα για την πρόγνωση του καιρού ήταν το βιβλίο του Θεόφραστου (300 π.Χ.), μαθητή του Αριστοτέλη, που είχε τίτλο «Περί σημείων». Εκεί περιγράφονται πάνω από 200 οιωνοί βροχής, ανέμου και καλοκαιρίας που προκύπτουν από το τσίμπημα των μυγών, από την συμπεριφορά των προβάτων, από τον τρόπο που καίγεται ένα λυχνάρι κατά την διάρκεια μιάς θύελλας, και από τον τρόπο που έρπουν οι σαρανταποδαρούσες σ' ένα τοίχο. Πάντως πολλές από τις παρατηρήσεις του Θεόφραστου έχουν επαληθευθεί, όπως, ότι η κόκκινη ανατολή του ηλίου και η άλως γύρω από αυτόν ή την σελήνη είναι προάγγελοι βροχής, ενώ κόκκινο ηλιοβασίλεμα είναι προάγγελος καλοκαιρίας. Η σύγχρονη μετεωρολογική έρευνα επαληθεύει τις παρατηρήσεις αυτές σε ποσοστό 70%. Η πραγματική μετεωρολογική έρευνα άρχισε τον 17ο αιώνα με την ανάπτυξη των οργάνων για ποσοτικές μετρήσεις.



Χρησιμοποιήθηκε το βαρόμετρο που εφευρέθηκε από τον Τορικέλι το 1643, και το οποίο γοήτευσε τόσο τους επιστήμονες της Αναγεννήσεως, ώστε να θεωρηθεί αλάνθαστος προφήτης του καιρού, και ονομάστηκε «γυαλί του καιρού». Οπλισμένοι με το βαρόμετρο, το θερμόμετρο (που κι αυτό είχε προηγηθεί απ' το 1503) και το υγρόμετρο οι ενθουσιώδεις επιστήμονες του αιώνα αυτού άρχισαν να θέτουν τις βάσεις της πρόγνωσης, καταγράφοντας τις

τιμές της πίεσης, της θερμοκρασίας, και της υγρασίας. Σώζονται ακόμα οι καταχωρήσεις των παρατηρήσεων που έγιναν από το 1649 ως το 1651 στο Παρίσι της Γαλλίας και στη Στοκχόλμη της Σουηδίας.

Τά καταγραφόμενα στοιχεία ολοένα πλήθαιναν έκτοτε, τά όργανα μέτρησης τελειοποιούνται και οι μελετητές του καιρού αποκτούσαν ολοένα και σαφέστερη εικόνα του. Οι μετεωρολογικοί χάρτες του 19ου αιώνα απεικόνιζαν τις κυκλωνικές θύελλες και έδειχναν την από δυτικά προς ανατολικά κίνησή τους. Όσο όμως η μεταβίβαση των πληροφοριών αργούσε, η πρακτική αξία όλων αυτών των μετρήσεων ήταν ελάχιστη. Η ανακάλυψη του τηλέγραφου απ' τον Σάμουελ Μόρς, σε συνδυασμό με την πύκνωση του δικτύου μετεωρολογικών μετρήσεων, έδωσε νέα ώθηση και στην εξέλιξη της επιστημονικής πρόγνωσης του καιρού. Το 1856 πρώτοι οι Γάλλοι ίδρυσαν εθνική υπηρεσία προαναγγελίας των θυελλών, αμέσως μετά την καταστροφή των γαλλικών και βρετανικών πολεμικών πλοίων, στο λιμάνι Μπαλακλάβα της Κριμαίας, εξ αιτίας μιάς θύελλας. Η σπουδαιότητα της πρόγνωσης του καιρού έγινε αντιληπτή. Ομοίως, μετά την απώλεια 450 ψυχών σε ναυάγιο, πάλι εξ αιτίας θύελλας στα παράλια της Ουαλίας, ίδρυσαν και οι Βρετανοί το 1860 ανάλογη υπηρεσία.

Στη χώρα μας αρχίζουν να γίνονται οι πρώτες μετεωρολογικές παρατηρήσεις το 1839 από το Αστεροσκοπείο Αθηνών, ενώ το 1890 αρχίζει να λειτουργεί ένα μικρό δίκτυο από μετεωρολογικούς σταθμούς σε όλη τη χώρα. Το 1931 ιδρύεται και μετεωρολογική υπηρεσία.

Ο 1ος παγκόσμιος πόλεμος έδωσε νέα ώθηση στην οργάνωση των μετεωρολογικών υπηρεσιών, αλλά και η ανάπτυξη της Φυσικής και των Μαθηματικών διευκόλυναν στην διατύπωση των θεωριών περί αερίων μαζών, περί νεφών, περί θερμοδυναμικής της ατμόσφαιρας. Μετά τον 2ο παγκόσμιο πόλεμο συμβαίνει πραγματική επανάσταση στη Φυσική της Ατμόσφαιρας. Στή δεκαετία του 1950 τίθενται σε τροχιά και οι πρώτοι μετεωρολογικοί

δορυφόροι, οι οποίοι με τις φωτογραφίες και τις μετρήσεις τους παρέχουν συνολική εικόνα για τη νεφική κάλυψη της γής πολλές φορές την ημέρα. Εκείνο όμως που έδωσε φτερά στην πρόγνωση ήταν η χρησιμοποίηση των ηλεκτρονικών υπολογιστών στην επεξεργασία του φοβερού όγκου πλέον των πληροφοριών που παρέχονται από μετεωρολογικά αερόστατα, αεροπλάνα, πλοία, επίγειους σταθμούς και μετεωρολογικούς δορυφόρους. Χρησιμοποιούνται πλέον αριθμητικά μοντέλα, που προσομοιάζουν τις καιρικές διεργασίες που συντελούνε στην ατμόσφαιρα, και τα αποτελέσματά τους αποτυπώνονται στις οθόνες των υπολογιστών ή στο χαρτί. Ήδη η ακρίβεια της βραχυχρόνιας πρόγνωσης αγγίζει το 85%, ενώ οι οιωνοί είναι αισιόδοξοι για το μέλλον.

Στις μέρες μας η χρήση των υπολογιστών, των νέων τεχνολογιών και της επιστήμης της Πληροφορικής, έχουν κάνει την μέτρηση και καταγραφή των καιρικών συνθηκών όχι μόνο πιο εύκολη αλλά και πιο ακριβή σε ποσοστά 99%. Είναι σημαντικό να έχουμε συνεχή ροή δεδομένων ώστε να έχουμε καλύτερη και έγκαιρη πρόγνωση, καθώς και όλο ένα τα προβλήματα του περιβάλλοντος αυξάνονται.

1.1 Αντικείμενο

Η πτυχιακή αυτή αναφέρεται στην σχεδίαση και ανάπτυξη του ήδη υπάρχοντος μοντέλου - που είναι εγκατεστημένο στην περιοχή ΤΕΙ Ηρακλείου- για την συλλογή, επεξεργασία και παρουσίαση σε πραγματικό χρόνο Μετεωρολογικών δεδομένων στο διαδίκτυο από 3 σταθμούς που είναι εγκατεστημένοι στο Ν. Ηρακλείου. Ειδικότερα οι 2 από αυτούς βρίσκονται στο περιοχή του ΤΕΙ Κρήτης- Ηράκλειο και ο τρίτος στην Χερσόνησο (30χμ από το κεντρικό μετεωρολογικό σταθμό- meteo server). Ο Σταθμός της Χερσονήσου βρίσκεται σε μικρό υψόμετρο από την επιφάνεια της θάλασσας.

Για την υλοποίηση αυτού του μοντέλου, εκτός της βάσης δεδομένων που ενημερώνεται σε πραγματικό χρόνο (10 λεπτά) θα δημιουργηθούν 2 διαφορετικές τεχνικές σύνδεσης των Data Logger με τον Meteo Server (bioklima5.prv3.teiher.gr) του Μετεωρολογικού σταθμού που βρίσκεται στην Βιοκλιματική κατοικία του ΤΕΙ Κρήτης/Ηράκλειο.

Το Block διάγραμμα του υπό ανάπτυξη συστήματος φαίνεται στο παρακάτω σχήμα. Οι τεχνικές που θα χρησιμοποιηθούν είναι:

- με χρήση εξοπλισμού τεχνολογίας GSM (CHERSONISSOS).
- με χρήση του τοπικού δικτύου Ethernet του ΤΕΙ χρησιμοποιώντας RS232- IP converters, (STEG).

Σκοπός του μοντέλου αυτού είναι η επέκταση της υπάρχουσας βάσης δεδομένων MySQL με δεδομένα και από τους άλλους σταθμούς για την παραγωγή συγκριτικών γραφημάτων, για την on-line παρουσίαση τους στο site του ΤΕΙ στο διαδίκτυο με χρήση δυναμικών γλωσσών. Ακόμη γίνονται υπολογισμοί της παραγόμενης ηλιακής και αιολικής ενέργειας καθώς και η συχνотική κατανομή της ταχύτητας του αέρα(Weibull distribution) στην κάθε περιοχή που υπάρχει data logger. Όλες αυτές οι πληροφορίες είναι πολύ ενδιαφέρουσες για τους ερευνητές και τους μηχανικούς που εργάζονται και αναπτύσσουν συστήματα στο χώρο των Ανανεώσιμων Πηγών Ενέργειας, Περιβάλλον, Γεωπονίας κ.τ.λ.

1.2 Οργάνωση του τόμου

Η συνέχεια του τόμου έχει οργανωθεί στα ακόλουθα κεφάλαια:

Στο κεφάλαιο 2, παρουσιάζεται η υπάρχουσα υποδομή και ο μετεωρολογικός σταθμός του εργαστηρίου και πώς αυτός συλλέγει και παρουσιάζει τις μετρήσεις.

Στο κεφάλαιο 3, αναπτύσσεται η εγκατάσταση, το λογισμικό, τα πρωτόκολλα επικοινωνίας και η τεχνολογία που χρησιμοποιήθηκε ώστε ο σταθμός μέτρησης που εγκαταστήσαμε στο Αγρόκτημα της σχολής Τεχνολόγων Γεωπόνων κάνοντας χρήση του εσωτερικού δικτύου του ΤΕΙ, να επικοινωνεί με το server του εργαστηρίου μας.

Στο κεφάλαιο 4, παρουσιάζουμε ο τρίτος σταθμός του δικτύου που βρίσκεται στην Χερσόνησο βασιζόμενοι στο δίκτυο Κινητής Τηλεφωνίας (Group Special Mobile -GSM) και χρησιμοποιώντας ειδικό εξοπλισμό που εγκαταστήσαμε.

Στο κεφάλαιο 5, περιγράφεται η προβολή των μετρήσεων στο Internet όπως αυτές συλλέγονται από κάθε σταθμό, τι λογισμικό έχει χρησιμοποιηθεί, και προτάσεις εξέλιξης.

Στο κεφάλαιο 6, ορίζουμε τα εργαλεία που χρησιμοποιήσαμε για την υλοποίηση του Δικτύου Σταθμών.

Στο κεφάλαιο 7, δίνουμε πληροφορίες σχετικά με τις μεθόδους που απορρίψαμε και αρχικά τις είχαμε λάβει υπ' οψιν μας.

Στο κεφάλαιο 8- Παράρτημα, δημοσιεύονται οι κώδικες όλων των προγραμμάτων που δημιουργήσαμε.

Τέλος, στο κεφάλαιο 9 δίνεται βιβλιογραφία και διάφοροι σύνδεσμοι (web links) στις οποίες μπορεί κανείς να αναζητήσει περισσότερες πληροφορίες για τα θέματα που πραγματευόμαστε.

2 Η υπάρχουσα υποδομή του εργαστηρίου

2.1 Εξοπλισμός Μετρήσεων

Ο κεντρικός μετεωρολογικός σταθμός είναι εγκατεστημένος στη Βιοκλιματική κατοικία και διαθέτει ένα καταγραφικό (Data Logger) της Delta- T Devices, τύπου DL2e όπου καταγράφονται κάθε 10 λεπτά οι τιμές των μετεωρολογικών παραμέτρων. Η επεξεργασία των μετεωρολογικών παραμέτρων γίνεται από κατάλληλα προγράμματα που είναι εγκατεστημένα σε Υπολογιστή, και η επικοινωνία είναι σειριακή.



Ch	Code	Label	Reading	Units
<input checked="" type="checkbox"/>	1	TM1	DL2-degC	28.09 deg C
<input checked="" type="checkbox"/>	2	WV2	Wind Dir	340 deg
<input checked="" type="checkbox"/>	3	LM	TEMP IN2	30 deg
<input checked="" type="checkbox"/>	5	PYR	PYRANOM	0.39970 Kw/m2
<input checked="" type="checkbox"/>	6	Skh	Out Temp	25.96 Deg
<input checked="" type="checkbox"/>	7	%RH	HUMIDITY	62.6 %RH
<input checked="" type="checkbox"/>	8	PYR	Pyran-PV	outside... Kw/m2
<input checked="" type="checkbox"/>	9	PRT	Temp PV	27.13 deg C
<input checked="" type="checkbox"/>	13	BS4	BAROMETE	1004 hPa
<input checked="" type="checkbox"/>	15	CO2	CO2	835 mV
<input checked="" type="checkbox"/>	61	AN1	Speed	2.50000 m/s
<input checked="" type="checkbox"/>	63	W...	RHwarmup	1

Εικόνα 2: Ο Data Logger και το Control Panel με ενδείξεις μετεωρολογικών παραμέτρων.

Όλοι οι αισθητήρες που συνδέονται στο καταγραφικό βρίσκονται τοποθετημένοι σε ιστό ύψους 10 m., εκτός του αισθητήρα εσωτερικής θερμοκρασίας της κατοικίας που βρίσκεται στο εσωτερικό της. Η συλλογή των μετεωρολογικών δεδομένων γίνεται από τους παρακάτω αισθητήρες:

Η συλλογή των μετεωρολογικών δεδομένων γίνεται από τους παρακάτω αισθητήρες.

2.2 Πυρανόμετρο τύπου CM 11

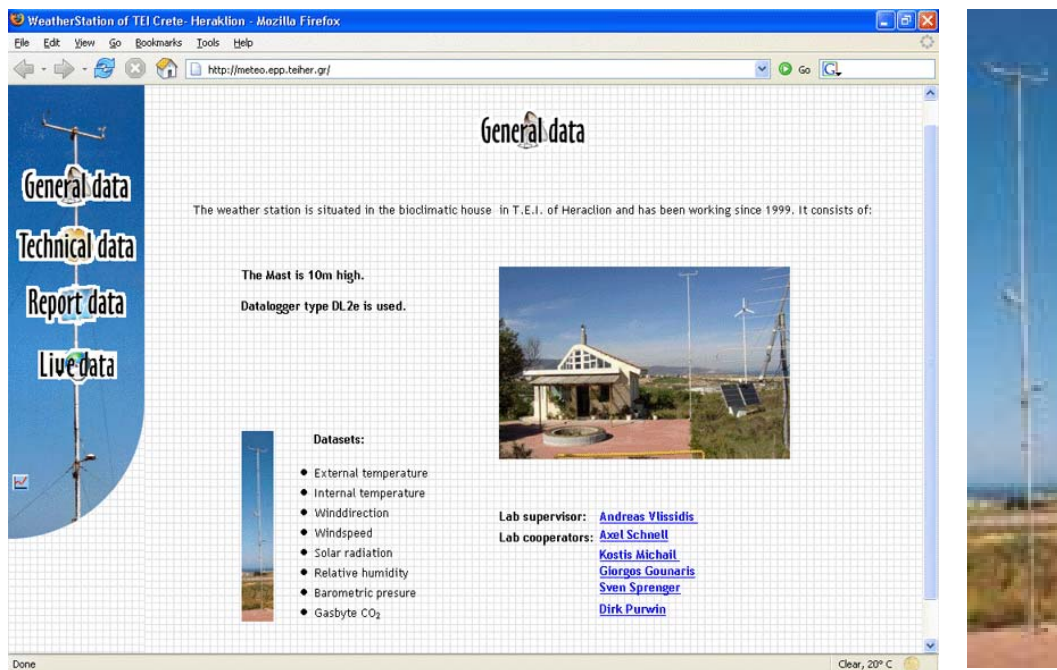
- Ανεμόμετρο DELTA-T Devices / A M 1
- Αισθητήρας Διεύθυνσης Ανέμου DELTA - T Devices / WD 1
- Αισθητήρας Θερμοκρασίας LM35
- Αισθητήρας Θερμοκρασίας PT-100
- Αισθητήρας Θερμοκρασίας Thermistor
- Αισθητήρας Σχετικής Υγρασίας SKYE Instruments / SK H 2011
- Αισθητήρας Βαρομετρικής Πίεσης DELTA - T Devices / BS 4
- Αισθητήρας συγκέντρωσης CO 2 / Gasbyte CO 2

Για περισσότερες πληροφορίες των τεχνικών χαρακτηριστικών του σταθμού, επισκεφθείτε την σελίδα μας <http://meteo.epp.teiher.gr>

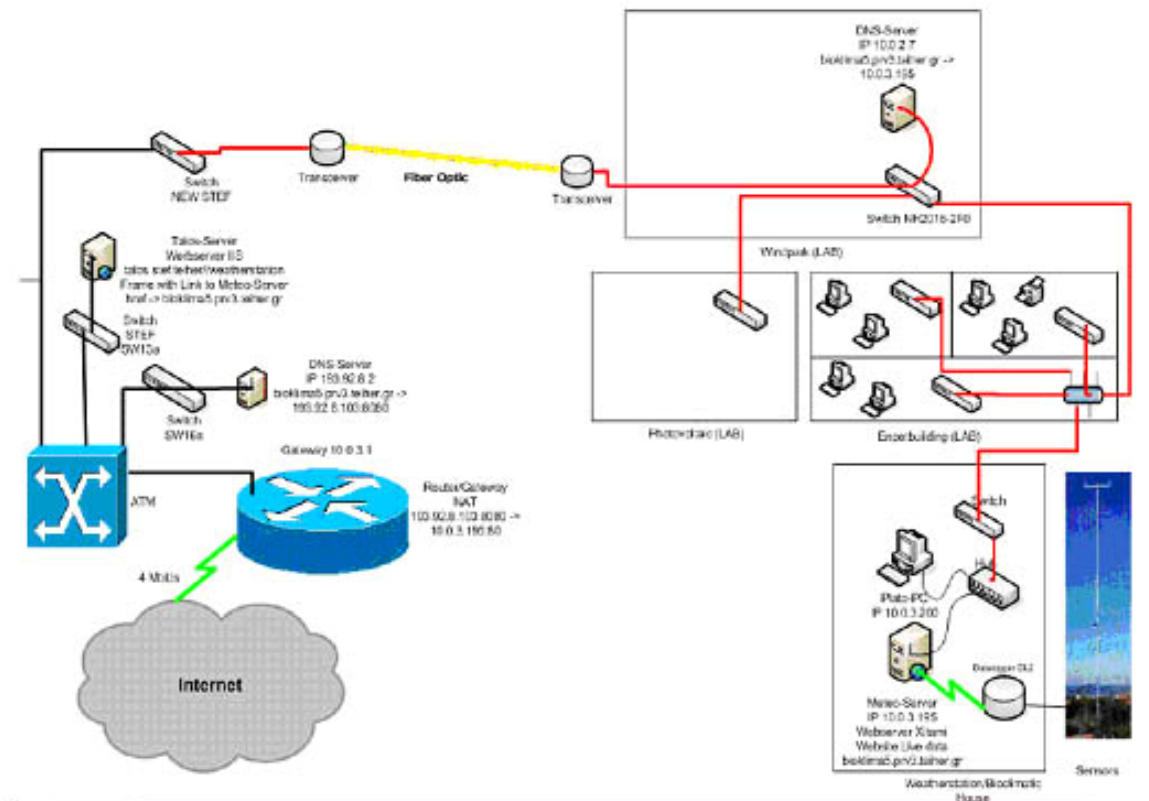
2.3 Server- Τοπολογία Δικτύου

Ο διαδικτυακός ιστοχώρος όπου γίνεται η παρουσίαση όλης της λειτουργίας του εργαστηρίου αποτελείται από δύο μέρη. Το πρώτο είναι το στατικό στο οποίο βρίσκεται και η αρχική σελίδα και φιλοξενείται στο server WEBS του Τμήματος, Εφαρμοσμένης Πληροφορικής και Πολυμέσων. Το δεύτερο βρίσκεται στο web server/ ServerBioHouse που βρίσκεται στο εργαστήριο και περιέχει το δυναμικό κομμάτι, παρουσίασης των μετρήσεων σε πραγματικό χρόνο.

Η σύνδεση του Μετεωρολογικού Σταθμού μας με το υπόλοιπο Τ.Ε.Ι. γίνεται με οπτική ίνα που αντικατέστησε την ασύρματη ζεύξη ταχύτητας 5,5 mbits/s με χρήση του συστήματος εκπομπής/λήψης Aironet 340 series της Cisco. Και τα δύο αυτά μέρη παρουσιάζονται στην ηλεκτρονική διεύθυνση <http://meteo.epp.teiher.gr>. Σκοπός μας είναι στο μέλλον όλη η ιστοσελίδα να μεταφερθεί στο web server του εργαστηρίου για μεγαλύτερες δυνατότητες αφού πρώτα διασφαλιστεί η σταθερότητα του δικτύου αλλά και του server.



Εικόνα 3: Η αρχική σελίδα στο διαδίκτυο.



Εικόνα 4: Παρουσίαση του δικτύου και η σύνδεση του με το υπόλοιπο δίκτυο του ΤΕΙ

2.4 Βάση Δεδομένων- Παρουσίαση Μετρήσεων.

```
Select C:\WINNT\system32\cmd.exe - mysql
```

Date	time	Speed	degC	WindDir	TempIN	PYRANOM	TempOUT	hUMIDITY	PyranPU	TempPU	BAROMETE	CO2
2005-04-07	21:40:05	2.90	20.040	280.600	17.370	-0.004	14.610	81.920	-80.400	7.600	1006.850	966
2005-04-07	21:50:05	2.50	20.260	280.300	17.235	-0.004	14.580	82.120	-76.200	7.570	1006.910	1027
2005-04-07	22:00:05	2.60	20.470	277.500	17.126	-0.004	14.530	82.430	-81.510	7.610	1006.970	984
2005-04-07	22:10:05	3.00	20.690	286.100	17.165	-0.004	14.530	83.510	-85.490	7.460	1006.850	948
2005-04-07	22:20:05	2.60	20.890	282.400	17.062	-0.005	14.530	83.970	-85.930	7.320	1006.850	951
2005-04-07	22:30:05	3.10	20.950	288.900	17.030	-0.005	14.520	85.500	-83.060	7.180	1006.670	951
2005-04-07	22:40:05	3.30	21.070	280.200	17.062	-0.005	14.520	87.400	-84.300	7.190	1006.560	950
2005-04-07	22:50:05	2.10	21.110	287.300	16.986	-0.004	14.480	87.810	-77.310	7.360	1006.730	1017
2005-04-07	23:00:05	1.60	21.220	277.000	16.851	-0.004	14.490	87.760	-78.190	7.120	1006.670	1011
2005-04-07	23:10:05	1.50	21.300	279.000	16.922	-0.004	14.520	87.760	-85.930	7.510	1006.620	953
2005-04-07	23:20:05	1.80	21.380	301.200	16.595	-0.005	14.520	87.810	-85.930	7.250	1006.620	946
2005-04-07	23:30:05	1.10	21.380	318.900	16.749	-0.004	14.540	87.140	-87.040	7.320	1006.730	942
2005-04-07	23:40:05	0.90	21.270	319.300	16.787	-0.005	14.530	86.580	-74.870	6.950	1006.730	999
2005-04-07	23:50:05	3.50	21.090	281.300	16.589	-0.005	14.490	85.910	-79.960	7.030	1006.790	930
2005-04-08	00:10:05	2.40	20.790	331.200	16.864	-0.004	14.600	89.400	-82.390	7.730	1006.730	940
2005-04-08	00:20:05	2.90	20.770	326.700	16.864	-0.003	14.640	90.320	-80.400	7.980	1006.790	947
2005-04-08	00:30:05	2.10	20.690	309.300	16.755	-0.003	14.570	90.060	-68.680	7.820	1006.910	1026
2005-04-08	00:40:05	3.50	20.550	55.050	16.800	-0.004	14.640	90.880	-70.010	8.310	1007.030	982
2005-04-08	00:50:05	2.50	20.590	216.000	16.947	-0.004	14.680	90.010	-76.200	8.230	1007.200	947

```
19 rows in set (0.01 sec)
mysql> created_by R0stis
```

Εικόνα 5: Η Βάση Δεδομένων του σταθμού.

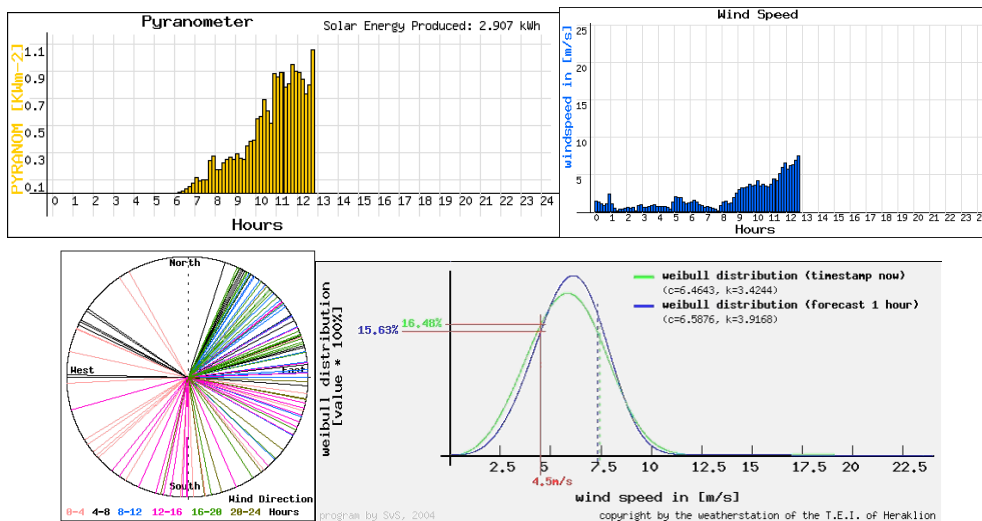
Η επικοινωνία του Data Logger με τον server γίνεται με την θύρα RS232. Ο DL διαθέτει 63 αναλογικά και ψηφιακά κανάλια, για την σύνδεση των αισθητηρίων και μια μεγάλη βάση δεδομένων με τους τύπους των αισθητηρίων μέτρησης μετεωρολογικών παραμέτρων. Παρέχεται ακόμη η δυνατότητα τοποθέτησης και άλλων αισθητηρίων που δεν βρίσκονται

στην βάση δεδομένων (open configuration). Ο DL διαθέτει μνήμη αποθήκευσης 64kB , και με την οργάνωση λειτουργίας (configuration), έχει δυνατότητα αποθήκευσης Μετεωρολογικών δεδομένων 2 μηνών.

Τα αποθηκευμένα δεδομένα “κατεβαίνουν” στον Η/Υ μέσω του λογισμικού της εταιρείας σε μορφή .bin. Ακόμη υπάρχει η δυνατότητα να γίνει άμεσα η μετατροπή σε αρχείο μορφής .dat. Στη συνέχεια με το σχετικό import wizard μετατρέπουμε τα αρχεία μορφής .bin σε έτοιμες φόρμες Excel, και έτσι στην συνέχεια δημιουργούνται τα γραφήματα που προορίζονται για το διαδίκτυο (στατική παρουσίαση μετεωρολογικών δεδομένων) σε εβδομαδιαία και μηνιαία βάση.

Αυτός ο τρόπος παρουσίασης σχεδιάζεται σε λειτουργία πραγματικού χρόνου έτσι ώστε τα μετεωρολογικά δεδομένα να παρουσιάζονται με δυναμικό τρόπο. Ηδη υπάρχει ένας εγκατεστημένος Server στο εργαστήριο όπου γίνεται η καταγραφή και η παρουσίαση των μετρήσεων σε πραγματικό χρόνο, κάθε 10 min.

Στην συνέχεια παρουσιάζονται εικόνες όπως αυτές εμφανίζονται στην σελίδα μας.

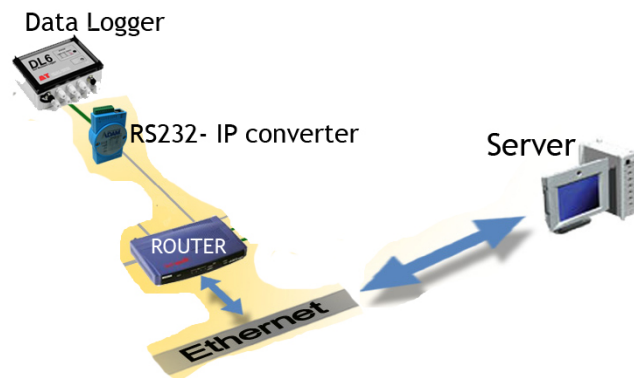


Εικόνα 6: Διαγράμματα Ηλιακής Ακτινοβολίας, Ταχύτητας Ανέμου, Διεύθυνσης Ανέμου, και πρόγνωσης αέρα.

3 ΣΤΑΘΜΟΣ ΜΕΤΡΗΣΗΣ Σ.ΤΕ.Γ

3.1 Γενική Περιγραφή

Ο σταθμός μέτρησης που εγκαταστήσαμε στα Αγρόκτημα της σχολή Τεχνολόγων Γεωπονίας του ΤΕΙ Κρήτης εκτός του εξοπλισμού καταγραφής – data logger- κάνει χρήση του εσωτερικού δικτύου του ΤΕΙ. Τα δεδομένα καταγράφονται ανα 10λεπτά στην βάση του server μας χρησιμοποιώντας το πρωτόκολλο επικοινωνίας TCP/IP. Το λογισμικό για την κλήση των δεδομένων γράφτηκε σε Visual Basic και τα βοηθητικά- δευτερεύοντα προγράμματα σε C. Για την υλοποίηση της επικοινωνίας του server μας με το καταγραφικό χρησιμοποιήθηκε τεχνολογία Virtual Private Network VPN, και αυτό γιατί ο server και το καταγραφικό βρίσκονται σε διαφορετικό VLAN έτσι από το Κέντρο Δικτύων (NOC) δεν μας επέτρεπαν κάποια άλλη μέθοδο (IP Mapping).



Εικόνα 3.1: Παράδειγμα δικτύου με μετατροπή RS232-IP

3.2 Εξοπλισμός

Ο εξοπλισμός που χρησιμοποιήθηκε για τον σταθμό μέτρησης βασίστηκε στις απαιτήσεις του καταγραφικού μας μηχανήματος που είναι Stylitis- 41 της εταιρίας symmetron με τα παρακάτω χαρακτηριστικά



Εικόνα 3.2: Το καταγραφικό μηχανήμα

Αυτόνομο καταγραφικό μπαταρίας κατάλληλο για τα πεδία των Μετεωρολογικών, Ανεμολογικών, Υδρολογικών και άλλων περιβαλλοντικών μετρήσεων.

- (4) Αναλογικές εισοδοί τάσης.
- (3) Είσοδοι απαριθμητών.
- (3) Ξεοδοί διέγερσης αισθητήρων ακριβείας 5V.
- Ενσωματωμένη προστασία υπέρτασης για όλες τις εισόδους και εξόδους.
- Μετράει: τάση, ρεύμα, συχνότητα, θερμοκρασία, πίεση, υγρασία, υψόμετρο, ηλιακή ακτινοβολία, ταχύτητα και κατεύθυνση ανέμου, ύψος βροχής, ταχύτητα ροής και βάθος νερού, κλπ.
- Δειγματοληψία ανά δευτερόλεπτο.
- Στατιστικά από 1 λεπτό έως 1 ώρα: Ελάχιστο, Μέγιστο, Μέσος Όρος, Τυπική Απόκλιση.
- Καταγραφή σε εσωτερική μνήμη ή κάρτα PCMCIA Flash.
- Τηλεμετάδοση δεδομένων μέσω δικτύου GSM (προαιρετικά).
- Λειτουργεί με μπαταρία σε θερμοκρασίες -30~+70 C.

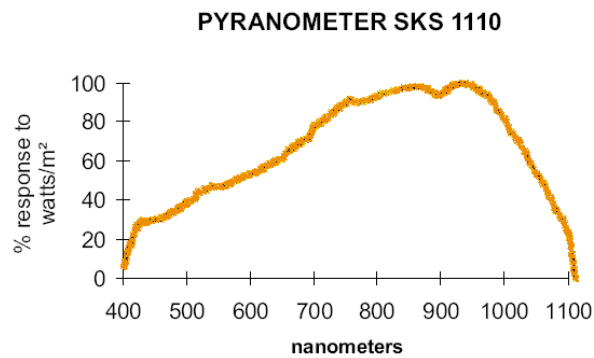
Ο υπόλοιπος εξοπλισμός αποτελείται από:

3.2.1 Αισθητήρας Ηλιακής Ακτινοβολίας SKS 1110 της Skye.

Χαρακτηριστικά	
Sensitivity έντασης	5μA / 100 W/m ²
Sensitivity τάσης	1μV / 100 W/m ²
Όρια μέτρησης	0- 5000 Wm ²
Σφάλμα Γραμμικότητας	<0,2%
Σφάλμα εργοστασιακής ρύθμισης	<3%, max 5%
Σφάλμα συνημίτονου	3%
Σφάλμα κλίσης	<1%
Παράγοντας θερμοκρασίας	±0.2%/οC
Σταθερότητα	±2%
Χρόνος Απόκρισης	10ns
Εσωτερική αντίσταση	c.200 Ohms
Όρια λειτουργίας- Θερμότητα	-30 οC, +75 οC
Όρια λειτουργίας- Υγρασία	0- 100%RH



Εικόνα 3.3: Το Αισθητήριο



Εικόνα 3.4: Γράφημα απόκρισης αισθητηρίου

3.2.2 Αισθητήρας Διεύθυνσης Ανέμου NRG #200P της NRGSYSTEMS.

Χαρακτηριστικά

Εύρος μέτρησης	360°
Τύπος σήματος	Analog DC voltage, 10K ohms
Ακρίβεια	1%
Έυρος σήματος εξόδου	0 V to excitation voltage
Threshold	1 m/s (2.2 miles per hour)
Τροφοδοσία	1 V- 15 V DC
Όρια λειτουργίας- Θερμοκρασία	-55 °C, +60°C
Όρια λειτουργίας- Υγρασία	0- 100%RH



Εικόνα 3.5: Το Αισθητήριο

3.2.3 Αισθητήρας Ταχύτητας Ανέμου NRG #40S της NRGSYSTEMS.

Χαρακτηριστικά

Εύρος μέτρησης	1 m/s- 96 m/s
Τύπος σήματος	Τετραγωνικός παλμός
Ακρίβεια	0.1 m/s
Έυρος σήματος εξόδου	0 Hz - 125 Hz
Threshold	0.78 m/s
Τροφοδοσία	5 V - 24 V
Όρια λειτουργίας- Θερμοκρασία	-55 °C, +60°C
Όρια λειτουργίας- Υγρασία	0- 100%RH



Εικόνα 3.6: Το Αισθητήριο

3.2.4 Αισθητήρας Θερμοκρασίας NRG #110S της NRGSYSTEMS.

Χαρακτηριστικά	
Εύρος μέτρησης	-40 °C - 52.5 °C
Τύπος σήματος	linear analog voltage
Ακρίβεια	offset is +/- 0.8 °C max nonlinearity is +/- 0.33 °C max total error +/- 1.1 °C max
Ευρος σήματος εξόδου	0 V - 2.5 V DC
Threshold	0.78 m/s
Τροφοδοσία	4 V - 35 V DC
Όρια λειτουργίας- Θερμοκρασία	-55 °C, +60°C
Όρια λειτουργίας- Υγρασία	0- 100%RH



Εικόνα 3.7: Το Αισθητήριο

3.2.5 Μετατροπέας Serial to Ethernet Digi One SP της Digi.

Λόγω των απαιτήσεων και της τοπολογίας του σταθμού μέτρησης χρησιμοποιήσαμε ένα μετατροπέα serial to Ethernet για να μπορούμε να έχουμε πρόσβαση του σταθμού από το εσωτερικό δίκτυο του ΤΕΙ. Με την ίδια τεχνική θα μπορούσαμε να έχουμε προσβάσιμο το καταγραφικό μας μηχάνημα από οποιοδήποτε μέρος στον πλανήτη αρκεί να είχαμε πρόσβαση στο Ίντερνετ. Η IP που ορίσαμε στον μετατροπέα είναι στατική και είναι η 10.3.2.200, και με TCP Port την 2101.

Χαρακτηριστικά:

- RS-232/422/485 switch
- DB-9M serial connector
- Baud rates up to 230 Kbps
- Hardware flow control
- Modem emulation
- 10/100Base-T Ethernet support
- TCP/UDP Socket Services
- UDP Multicast
- Telnet and Reverse Telnet
- DHCP/RARP/ARP-Ping for
- IP address assignment
- HTTP configuration
- RealPort technology for COM and TTY ports
- LED status for link and power



Εικόνα 3.8: Ο μετατροπέας

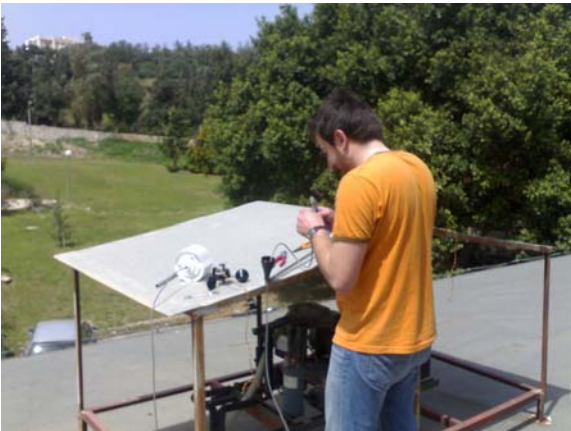
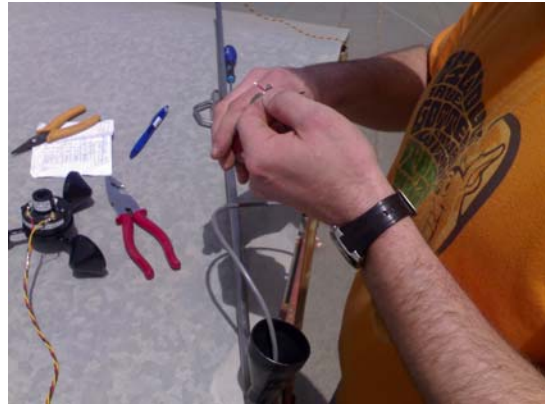
3.3 Εγκατάσταση

Η ρύθμιση των παραμέτρων των αισθητηρίων με το καταγραφικό έγινε στο εργαστήριο μας. Έπειτα μεταφέρθηκαν στο αγρόκτημα της Σ.Τ.Ε.Γ. για την μόνιμη εγκατάστασή τους, όπως είναι μέχρι σήμερα.

Ο καταγραφικό και ο μετατροπέας εγκαταστάθηκαν στο εσωτερικό κτιρίου για την αποφυγή κλοπής. Τα αισθητήρια προσαρμόστηκαν σε ειδικό μετεωρολογικό ιστό στην οροφή του κτιρίου και προσανατολίστηκαν με το βορρά με ειδική συσκευή GPS ώστε η μετρήσεις τις διεύθυνσης ανέμου να είναι αξιόπιστες.

Στον Ιστό προσαρμόσαμε υδατοστεγές κουτί ηλεκτρολογικού τύπου όπου καταλήγουν οι απολήξεις των καλωδίων των αισθητήρων. Για να συνδεθούν τα αισθητήρια με το καταγραφικό μηχανήμα χρησιμοποιήσαμε καλώδιο UTP κατηγορίας 5e όπου σύμφωνα με τα διεθνή στάνταρ κατασκευής καλωδίων είναι κατάλληλα για χρήση σε δίκτυα συχνότητες μέχρι τα 100 MHz των 100 Mbit/s αλλά κάνει και για δίκτυα των 1Gigabit. Ακολουθεί σχετικός πίνακας για τις κατηγορίες των καλωδίων UTP. Για την καλύτερη προφύλαξη των καλωδίων από τις εξωτερικές συνθήκες χρησιμοποιήθηκε σπιδράλ "Duroflex" μεσαίου τύπου 16mm.

<u>Κατηγορία</u>	<u>Χρήση</u>
1	καλώδιο PSTN ή ISDN
2	4 Mbit/s token ring
3	συχνότητες μέχρι τα 16 MHz των 10 Mbit/s.
4	μέχρι τα 20 MHz των 10 Mbit/s.
5	συχνότητες μέχρι τα 100 MHz των 100 Mbit/s.
5e	συχνότητες μέχρι τα 100 MHz των 100 Mbit/s αλλά κάνει και για δίκτυα των 1Gigabit.
6	συχνότητες μέχρι τα 250 MHz και είναι τα πλέον κατάλληλα για δίκτυα του 1Gigabit.
7	συχνότητες μέχρι τα 600 MHz των 10Gigabit



Κατά την εγκατάσταση των αισθητήρων, στον ιστό.

StylOut.txt

```
Site: STEG card number: 0
cardtype: DOWNLOADED
filename: 1 Math interval: 10 min
length of file: 295 number of records: 3
input type rate slope offset
counter 1 wind speed .765 .35
counter 2 wind speed 1 0
counter 3 wind speed 1 0
analog 1 analog 55.55 -86.39
analog 2 vane 1 0
analog 3 analog 100 0
analog 4 vane 1 15
All vane values are referenced to North.
8/ 4/ 2008
C1-min C1-max C1-av C1-sdv C2-min C2-max
C2-av C2-sdv C3-min C3-max C3-av C3-sdv A1 -min A1 -max A1
-av A1 -sdv A2 -min A2 -max A2 -av A2 -sdv A3 -min A3 -max A3 -
av A3 -sdv A4 -min A4 -max A4 -av A4 -sdv
08042008 1450 0.3 0.3 0.3 0 0 0 0 0 0 0 -85.915 -
0.066 0 0 0 0 0 3.907 8.059 6.403
1.453 15 15 15 0
```

Στις πρώτες γραμμές παίρνουμε πληροφορίες για το στήσιμο του σταθμού, το όνομα, το ρυθμό καταγραφής, το αριθμό των μετρήσεων κτλ.

Στις τελευταίες γραμμές βλέπουμε τα κανάλια καταγραφής και ακριβώς από κάτω την ημερομηνία, την ώρα και την μέτρηση για κάθε κανάλι. Να σημειωθεί πώς για κάθε κανάλι μέτρησης παίρνουμε 4 τιμές, την μέγιστη (max), την ελάχιστη (min), την μέση (av)- που χρησιμοποιούμε κατά κόρον- και την Τυπική Απόκλιση (sdv).

Στην συνέχεια αφού έχουμε επιτυχημένη επικοινωνία με το καταγραφικό και η παραπάνω διαδικασία έχει ολοκληρωθεί χωρίς προβλήματα εκτελείται στο server το πρόγραμμα read.exe που παίρνει σαν όρισμα το αποσυμπίεσμένο αρχείο StylOut.txt από την προηγούμενη διαδικασία και δίνει σαν αποτέλεσμα μόνο την τελευταία γραμμή του προγράμματος στοιχισμένη και με διαχωρισμό των τιμών με tab (/t).

Η εφαρμογή read.exe είναι γραμμένη σε C χρησιμοποιώντας τον Compiler της Borland 6 C++. Ο κώδικας της εφαρμογής βρίσκεται στο Παράρτημα I (8^ο Κεφάλαιο).

Η εκτέλεση του read.exe μας δίνει σαν αποτέλεσμα το τελικό αρχείο out.txt που χρησιμοποιείται σαν όρισμα σε query από την MySQL σε για να εισαχθεί η εγγραφή στη βάση. Βλέπουμε στο αρχείο τα δύο πρώτα ορίσματα είναι η ημερομηνία και η ώρα σε μορφή ddmmyyyy και hhmm αντίστοιχα. Στην συνέχεια δίνονται η τέσσερις τιμές ανά κανάλι μέτρησης του data logger.

out.txt

```
08042008 1450 0.3 0.3 0.3 0 0 0 0 0 0 0 -85.915 -
85.78 -85.83 0.066 0 0 0 0 3.907 8.059 6.403 1.453 15 15 15
0
```

sql.bat

```
cd\  
cd mysql  
cd bin  
mysql stdb < C:/EtherCon/sqlcom.sql  
exit  
exit
```

sqlcom.sql

```
LOAD DATA LOCAL INFILE 'C:/EtherCon/out.txt' INTO TABLE data;  
exit
```

3.5 Η Βάση Δεδομένων

3.5.1 Γενικά για την σχεδίαση της βάσης.

Στις μέρες μας όλοι αναζητούν την πληροφορία και προτιμούν να τους παρέχεται από έναν μόνο πάροχο. Η όλη ιδέα και το όλο εγχείρημα γίνεται για να καλυφθούν οι περισσότερες δικτυακές συνδέσεις, που μπορούν να γίνουν με ένα Data Logger. Αυτό λοιπόν αποσκοπεί, ότι ο κάθε ένας που έχει ένα καταγραφικό μηχάνημα να έχει τον τρόπο να συνδεθεί με το server μας, έτσι όλα τα δεδομένα θα είναι συγκεντρωμένα σε μια κοινή βάση δεδομένων που ο κάθε ένας, θα μπορεί να έχει πρόσβαση.

Με λίγα λόγια, θα μπορούμε να αναζητούμε, να ταξινομούμε και να ανακαλούμε τα δεδομένα αποτελεσματικά. Ο MySQL διακομιστής ελέγχει την πρόσβαση στα δεδομένα, για να μπορούν να δουλεύουν ταυτόχρονα πολλοί χρήστες, για να παρέχει γρήγορη πρόσβαση και να διασφαλίσει ότι μόνο πιστοποιημένοι χρήστες μπορούν να έχουν πρόσβαση. Τα αποτελέσματα κάθε περιοχής θα αποθηκεύονται σε ξεχωριστό πίνακα για να είναι πιο εύκολες οι αναζητήσεις. Όλα τα καταγραφικά θα χρησιμοποιούν την ίδια βάση με τα ίδια πεδία. Ο Server θα είναι Non-transaction για να είναι πολύ γρήγορος και τα δεδομένα να καταλαμβάνουν λιγότερο χώρο στο δίσκο του. Επίσης το όλο μοντέλο δεν έχει την απαίτηση να γίνονται update, κάτι που απαιτεί περισσότερη μνήμη. Απαραίτητη κρίνεται και η δυνατότητα merge για να μπορούμε να συνοψίζουμε τους πίνακες και να μπορούμε να κάνουμε σύνθετα ερωτήματα στη βάση. Δεν μας ενδιαφέρει το Rollback και αλλά από τα θετικά που έχει μια βάση με transaction.

Στην αρχή οι ταυτόχρονες συνδέσεις θα περιοριστούν στις 20 μετά ανάλογα με την ανάπτυξη του έργου μπορεί αυτό να ανέβει. Μπορεί να είναι ένας απλός διακομιστής, χωρίς πολλές απαιτήσεις σε μνήμη. Αν όμως τα καταγραφικά είναι πολλά, θα γίνει Dedicated Myself server ο οποίος θα διαχειρίζεται όλη τη μνήμη και θα μπορούσε να καλύψει ευρύτερες ανάγκες.

Η γνώση της SQL είναι πολύτιμη για οποιονδήποτε θελήσει να αποθηκεύσει ή να ανακτήσει δεδομένα. Η βάση μπορεί εύκολα να εξάγει τα δεδομένα σε φύλλα του excel για όσους δεν γνωρίζουν. Η επιλογή της MySQL δεν ήταν τυχαία, έγινε μετά από αρκετή έρευνα στο διαδίκτυο, συμβουλές από άλλους χρηστές και εξαντλητικές δοκιμές. Στο διαδίκτυο χαρακτηρίζεται ως η καλύτερη επιλογή από πολλούς που στηρίζουν το ανοικτό λογισμικό.

3.5.2 Ο Πίνακας data του σταθμού.

Server: localhost ▶ Database: stdb ▶ Table: data

	Field	Type	Attributes	Null	Default	Extra	Action						
<input type="checkbox"/>	date	varchar(8)		No									
<input type="checkbox"/>	time	time		No	00:00:00								
<input type="checkbox"/>	C1_min	varchar(7)		No									
<input type="checkbox"/>	C1_max	varchar(7)		No									
<input type="checkbox"/>	C1_av	varchar(7)		No									
<input type="checkbox"/>	C1_sdv	varchar(7)		No									
<input type="checkbox"/>	C2_min	varchar(7)		No									
<input type="checkbox"/>	C2_max	varchar(7)		No									
<input type="checkbox"/>	C2_av	varchar(7)		No									
<input type="checkbox"/>	C2_sdv	varchar(7)		No									
<input type="checkbox"/>	C3_min	varchar(7)		No									
<input type="checkbox"/>	C3_max	varchar(7)		No									
<input type="checkbox"/>	C3_av	varchar(7)		No									
<input type="checkbox"/>	C3_sdv	varchar(7)		No									
<input type="checkbox"/>	A1_min	varchar(7)		No									
<input type="checkbox"/>	A1_max	varchar(7)		No									
<input type="checkbox"/>	A1_av	varchar(7)		No									
<input type="checkbox"/>	A1_sdv	varchar(7)		No									
<input type="checkbox"/>	A2_min	varchar(7)		No									
<input type="checkbox"/>	A2_max	varchar(7)		No									
<input type="checkbox"/>	A2_av	varchar(7)		No									
<input type="checkbox"/>	A2_sdv	varchar(7)		No									
<input type="checkbox"/>	A3_min	varchar(7)		No									
<input type="checkbox"/>	A3_max	varchar(7)		No									
<input type="checkbox"/>	A3_av	varchar(7)		No									
<input type="checkbox"/>	A3_sdv	varchar(7)		No									
<input type="checkbox"/>	A4_min	varchar(7)		No									
<input type="checkbox"/>	A4_max	varchar(7)		No									
<input type="checkbox"/>	A4_av	varchar(7)		No									
<input type="checkbox"/>	A4_sdv	varchar(7)		No									
<input type="checkbox"/>	aa	int(11)		No		auto_increment							

With selected:

Εικόνα 3.10: Η δομή του πίνακα data όπως φαίνεται από το phpmyadmin.

Ο πίνακας έχει σχεδιαστεί έτσι ώστε σε μελλοντική αλλαγή των ρυθμίσεων του καταγραφικού να μην επηρεαστεί η ροή των δεδομένων. Έτσι, όπως φαίνεται και στη πιο πάνω εικόνα, στην στήλη των πεδίων έχουμε στις δύο πρώτες θέσεις, το πεδίο date, και το πεδίο time με default μορφή ddmmyyy και hh:mm:ss (22:30:00) αντίστοιχα.

Στη συνέχεια είναι τα πεδία των καναλιών C1, C2, C3 που είναι τα 4 ψηφιακά κανάλια του σταθμού. Όπως έχει αναφερθεί και παραπάνω για κάθε κανάλι παίρνουμε 4 τιμές. Στην εξαγωγή των γραφημάτων χρησιμοποιούμε την τιμή _av που είναι η μέση τιμή μιας περιόδου καταγραφής.

Τα κανάλια A1, A2, A3, A4 είναι τα αναλογικά κανάλια, και πάλι έχουμε 4 τιμές ανα κανάλι. Όπως και στα ψηφιακά έτσι και εδώ χρησιμοποιούμε πάντα την τιμή _av.

Ο τύπος των πεδίων είναι varchar(7) ώστε να μην έχουμε κάποιο σφάλμα σε περίπτωση που το καταγραφικό μας, αντί για αριθμό στείλει κάποιο χαρακτήρα (flag). Το μήκος σε 7 χαρακτήρες ορίστηκε γιατί πάντα τα δεδομένα είναι της μορφής xxx.yyy.

Το 31^ο πεδίο όπου έχουμε το πεδίο aa για την αρίθμηση των εγγραφών στη βάση με extra τον χαρακτηρισμό του ως auto_increment.

Στον σχεδιασμό της βάσης αξίζει να σημειωθεί ότι γίνεται χρήση της MySQL που είναι το πιο δημοφιλές Open Source SQL σύστημα διαχείρισης βάσεων δεδομένων. Περισσότερες πληροφορίες για την MySQL δίνεται στο τέλος του τόμου.

3.6 Γενική περιγραφή του Πρωτοκόλλου TCP/IP

Το TCP/IP είναι μια σουίτα πρωτοκόλλων οργανωμένη σε επίπεδα. Για να κατανοήσουμε τη σημασία αυτού, είναι χρήσιμο να δοθεί ένα παράδειγμα. Μια τυπική κατάσταση e_mail ή αποστολή ταχυδρομείου. Αρχικά, υπάρχει το πρωτόκολλο για το μήνυμα, το οποίο ορίζει το σύνολο των εντολών που μια μηχανή θα στέλνει σε κάποια άλλη, π.χ. εντολές για τον καθορισμό του αποστολέα του μηνύματος, του παραλήπτη και το τέλος του κειμένου του μηνύματος. Το πρωτόκολλο αυτό υποθέτει βέβαια, ότι υπάρχει τρόπος αξιόπιστης επικοινωνίας μεταξύ των δυο υπολογιστών. Το ταχυδρομείο, όπως άλλα πρωτόκολλα εφαρμογών απλά καθορίζει ένα σετ εντολών και μηνυμάτων για αποστολή. Είναι σχεδιασμένο να χρησιμοποιείται μαζί με τα TCP και IP. Το TCP είναι υπεύθυνο για το πέρασμα των εντολών στο άλλο άκρο. Κρατάει λογαριασμό του τι αποστέλλεται και επανεκπέμπει οτιδήποτε δεν πέρασε. Αν ένα μήνυμα είναι πολύ μεγάλο για ένα datagram π.χ. το κείμενο του ταχυδρομείου, το TCP θα το χωρίσει σε πολλαπλά datagrams και θα φροντίσει ώστε να φτάσουν σωστά. Εφόσον αυτές οι λειτουργίες απαιτούνται για πολλές εφαρμογές, ομαδοποιούνται σε ένα ξεχωριστό πρωτόκολλο αντί να αποτελούν μέρος των προδιαγραφών για την αποστολή ταχυδρομείου. Μπορούμε να σκεφτούμε ότι το TCP φτιάχνει μια βιβλιοθήκη ρουτινών που οι εφαρμογές μπορούν να χρησιμοποιούν, όταν χρειάζονται, αξιόπιστες επικοινωνίες με άλλο υπολογιστή. Όμοια το TCP καλεί τις υπηρεσίες του IP. Αν και οι υπηρεσίες που παρέχει το TCP απαιτούνται από πολλές εφαρμογές, υπάρχουν ακόμα μερικά είδη εφαρμογών που δεν τις χρειάζονται. Σίγουρα όμως υπάρχουν και υπηρεσίες που απαιτούνται από όλες τις εφαρμογές. Οι υπηρεσίες αυτές ομαδοποιούνται στο IP. Όπως και με το TCP, μπορούμε να σκεφτούμε το IP σαν μια βιβλιοθήκη ρουτινών που καλεί το TCP και που είναι επίσης διαθέσιμη σε εφαρμογές που δεν χρησιμοποιούν το TCP. Η στρατηγική αυτή της οικοδόμησης πολλαπλών επιπέδων του πρωτοκόλλου καλείται "layering". Έτσι, μπορούμε να θεωρήσουμε τα διάφορα προγράμματα εφαρμογών στη θέση του ταχυδρομείου και τα TCP, IP σαν ξεχωριστά επίπεδα, κάθε ένα από τα οποία καλεί τις υπηρεσίες του αμέσως πιο χαμηλού επιπέδου. Γενικά οι TCP/IP εφαρμογές έχουν τέσσερα επίπεδα: Το TCP/IP είναι βασισμένο στο "catenet model" (περιγράφεται με λεπτομέρεια στο IEN 48). Το μοντέλο αυτό υποθέτει ότι υπάρχει ένας μεγάλος αριθμός ανεξάρτητων δικτύων συνδεδεμένων μεταξύ τους με gateways. Ο χρήστης πρέπει να μπορεί να προσπελάσει υπολογιστές ή άλλα αγαθά σε οποιοδήποτε από αυτά τα δίκτυα. Τα datagrams περνούν συχνά από δεκάδες διαφορετικά δίκτυα πριν φτάσουν στον τελικό τους προορισμό. Η διαδρομή που χρειάζεται για να επιτευχθεί αυτό, πρέπει να είναι αόρατη στο χρήστη.

Όσον αφορά το χρήστη, το μόνο που χρειάζεται να γνωρίζει για την προσπέλαση ενός άλλου συστήματος είναι η Internet διεύθυνσή του. Αυτή είναι μια διεύθυνση με την εξής μορφή: 128.6.4.194. Στην πραγματικότητα, η διεύθυνση είναι ένας 32-bit αριθμός, αν και γράφεται σαν τέσσερα δεκαδικά ψηφία χωρισμένα μεταξύ τους με μια τελεία, που ο καθένας αντιστοιχεί σε 8 bits της διεύθυνσης. (Για τέτοιες ομάδες των 8 bits χρησιμοποιείται ο όρος "octets". Ο όρος "byte" δεν χρησιμοποιείται, επειδή το TCP/IP υποστηρίζεται από μερικούς υπολογιστές που έχουν μήκος byte διαφορετικό των 8 bits). Γενικά, η δομή της διεύθυνσης δίνει πληροφορίες για το πως θα φτάσουμε στο σύστημα. Για παράδειγμα, το 128.6 είναι το νούμερο του δικτύου που έχει ανατεθεί στο Rutgers University. Το Rutgers χρησιμοποιεί το επόμενο octet, για να δηλώσει σε ποιο από τα Ethernets του πανεπιστημίου γίνεται αναφορά. Για παράδειγμα, 128.6.4 είναι το Ethernet που χρησιμοποιείται από το Computer & Science Dept. Το τελευταίο octet επιτρέπει μέχρι και 254 συστήματα σε κάθε Ethernet. (Είναι 254, γιατί το 0 και το 255 δεν επιτρέπονται, για λόγους που θα συζητηθούν αργότερα). Φυσικά, αναφερόμαστε συνήθως σε συστήματα, με όνομα, παρά με τη διεύθυνση Internet. Όταν δίνουμε ένα όνομα, το λογισμικό του δικτύου συμβουλευείται μια βάση δεδομένων και βρίσκει την αντίστοιχη διεύθυνση Internet. (Το RFC 882 περιγράφει την τεχνολογία των name servers που χρησιμοποιούνται για το χειρισμό του εντοπισμού των διευθύνσεων).

Το TCP/IP είναι χτισμένο με τεχνολογία "χωρίς σύνδεση" (connectionless). Η πληροφορία μεταφέρεται σαν μια ακολουθία datagrams. Το datagram είναι μια ομάδα δεδομένων που στέλνεται σαν ξεχωριστό μήνυμα. Κάθε ένα από τα datagrams, στέλνεται ατομικά μέσω του δικτύου. Αφού επιτευχθεί εγκατάσταση σύνδεσης, η πληροφορία διασπάται σε datagrams, τα οποία αντιμετωπίζονται από το δίκτυο απολύτως ξεχωριστά. Για παράδειγμα, υποθέστε ότι θέλετε να μεταφέρετε ένα αρχείο 15000 octets. Τα περισσότερα δίκτυα δεν μπορούν να χειριστούν ένα datagram 15000 octets. Έτσι, τα πρωτόκολλα θα το διασπάσουν π.χ. σε 30 datagrams των 500 octets. Κάθε ένα από αυτά τα datagrams θα σταλεί στο άλλο άκρο. Στο σημείο αυτό θα επανασυνδεθούν για να διαμορφώσουν το πρωτότυπο αρχείο των 15000 octets. Όσο όμως τα datagrams αυτά είναι σε μεταφορά, το δίκτυο δεν γνωρίζει ότι υπάρχει κάποια σύνδεση μεταξύ τους. Έτσι, είναι πολύ πιθανόν να φτάσει το datagram 14, πριν το 13. Είναι επίσης πιθανό, ότι κάπου στο δίκτυο θα συμβεί κάποιο λάθος και κάποιο datagram δεν θα περάσει καθόλου. Στην περίπτωση αυτή το datagram πρέπει να σταλεί ξανά. Σημειώστε, ότι οι όροι "datagram" και "packet" φαίνονται να έχουν την ίδια έννοια. Τεχνικά, το datagram είναι ο σωστός όρος που πρέπει να χρησιμοποιείται όταν περιγράφεται το TCP/IP. Το datagram είναι μια μονάδα δεδομένων με την οποία ασχολούνται τα πρωτόκολλα. Το πακέτο είναι κάτι φυσικό, που παρουσιάζεται στο Ethernet ή κάποιο καλώδιο. Στις περισσότερες περιπτώσεις το πακέτο απλώς περιέχει ένα datagram κι έτσι η διαφορά είναι πολύ μικρή. Μπορεί όμως και να διαφέρουν. Όταν το TCP/IP χρησιμοποιείται πάνω το X.25 το X.25 interface διασπά τα datagrams σε 128-byte πακέτα. Αυτό δεν το βλέπει το IP, επειδή τα πακέτα συντίθεται ξανά σε ένα datagram στην άλλη άκρη πριν τα επεξεργαστεί το TCP/IP. Έτσι στην περίπτωση αυτή 1 IP datagram θα μεταφέρεται από πολλά πακέτα. Στις περισσότερες περιπτώσεις βέβαια, υπάρχουν αρκετά πλεονεκτήματα στην αποστολή ενός datagram / πακέτο και έτσι οι διακρίσεις τείνουν να εξαλειφθούν.

ΤΟ ΕΠΙΠΕΔΟ TCP

Υπάρχουν δυο διαφορετικά πρωτόκολλα που χειρίζονται τα TCP/IP datagrams. Το TCP (Transmission control protocol) είναι υπεύθυνο για τη διάσπαση του μηνύματος σε datagrams και την επανασύνδεσή τους στο άλλο άκρο επαναστέλλοντας οτιδήποτε έχει χαθεί και ταξινομώντας τα στη σωστή σειρά. Το IP (Internet protocol) είναι υπεύθυνο για τη δρομολόγηση των datagrams. Ίσως φαίνεται ότι το TCP κάνει όλη τη δουλειά. Στα μικρά δίκτυα, αυτό πράγματι συμβαίνει. Παρ' όλα αυτά στο Internet, το να φτάσει απλώς ένα datagram στον προορισμό του μπορεί να είναι μια δύσκολη δουλειά. Το datagram συνήθως πρέπει να περάσει από πολλά διαφορετικά είδη δικτύων όπου απαιτείται χειρισμός των ασυμβατοτήτων μεταξύ των διαφορετικών μέσων μετάδοσης. Σημειώστε ότι το Interface μεταξύ του TCP και IP είναι σχετικά απλό. Το TCP απλά παραδίδει στο IP ένα datagram και τον προορισμό του. Το IP δεν γνωρίζει πως το datagram αυτό συνδέεται με το προηγούμενο ή το επόμενο του. Μέχρι εδώ έχουμε μιλήσει για τις διευθύνσεις του Internet, αλλά όχι για το πως κρατιέται λογαριασμός των πολλαπλών συνδέσεων σε ένα σύστημα. Σίγουρα δεν είναι αρκετό να φτάσει ένα datagram στο σωστό προορισμό. Το TCP πρέπει να γνωρίζει ποιας σύνδεσης είναι μέρος, το συγκεκριμένο datagram. Η διαδικασία αυτή αναφέρεται σαν "απόπλεξη" (Demultiplexing). Στην πραγματικότητα, υπάρχουν πολλά επίπεδα απόπλεξης στο TCP/IP. Οι πληροφορίες που χρειάζονται για να γίνει η απόπλεξη αυτή περιέχεται σε μια σειρά επικεφαλίδων (headers). Η επικεφαλίδα είναι απλώς μερικά επιπλέον octets που προσαρτούνται στην αρχή των datagrams από ένα πρωτόκολλο. Αμέσως παρακάτω, φαίνεται πως ακριβώς προσαρτούνται επικεφαλίδες σε ένα μήνυμα που περνά μέσα από ένα τυπικό TCP/IP δίκτυο.

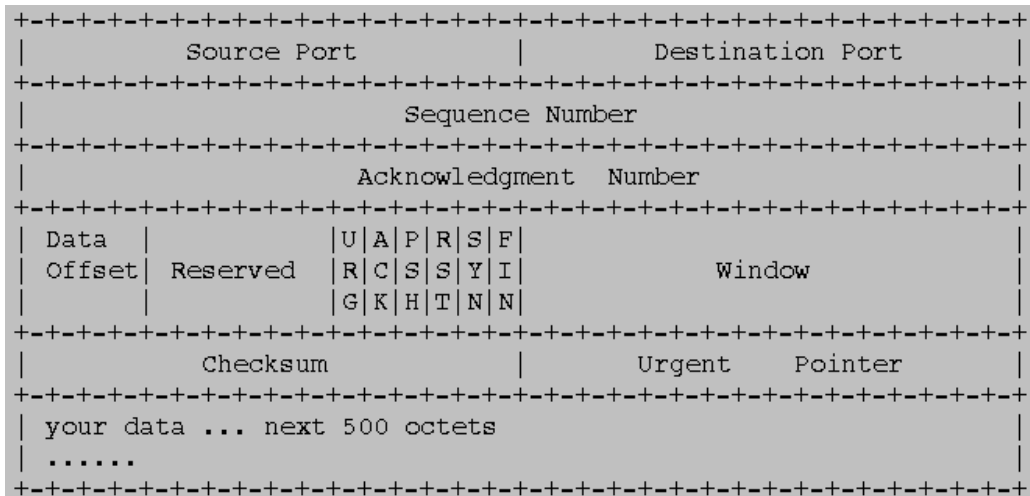
Έστω ότι έχουμε ένα αρχείο που πρέπει να το στείλουμε σε έναν άλλο υπολογιστή:

Το TCP χωρίζει το αρχείο σε κομμάτια ώστε να μπορεί να τα χειριστεί (Για να γίνει αυτό πρέπει το TCP να γνωρίζει το μεγαλύτερο μήκος datagram που μπορεί να χειριστεί το συγκεκριμένο δίκτυο. Στην πραγματικότητα τα TCP στις δύο άκρες δηλώνουν το μεγαλύτερο datagram που μπορούν να χειριστούν και επιλέγεται το μικρότερο από τα δύο).

*** **

Το TCP βάζει μια επικεφαλίδα στην αρχή του κάθε datagram . Η επικεφαλίδα αυτή περιέχει τουλάχιστον 20 octets, αλλά τα πιο σημαντικά είναι ο αριθμός της πόρτας πηγής και προορισμού και ο αριθμός σειράς (source and destination port number - sequence number) Τα port numbers χρησιμοποιούνται για την ταυτοποίηση των διαφόρων συνδιαλέξεων. Ας υποθέσουμε ότι 3 διαφορετικοί άνθρωποι μεταφέρουν αρχεία. Το TCP αναθέτει τα port number 1000, 1001, και 1002 στις μεταφορές αυτές. Όταν αποστέλλεται ένα datagram, τα νούμερα αυτά αποτελούν τα source port numbers. Φυσικά το TCP στο άλλο άκρο, έχει αναθέσει ένα δικό του port number για την συνδιάλεξη. Το TCP στο σύστημα αποστολής του αρχείου πρέπει να γνωρίζει το port number που χρησιμοποιείται στο άλλο άκρο, το οποίο και τοποθετεί στο πεδίο πόρτας προορισμού (destination port field). Φυσικά, αν από το άλλο άκρο σταλεί πίσω ένα datagram τα source και destination ports θα αντιστραφούν. Κάθε datagram έχει ένα sequence number. Αυτός ο αριθμός χρησιμοποιείται ώστε το άλλο άκρο να παίρνει τα datagrams στη σωστή σειρά και να εξασφαλίζεται ότι δεν υπάρχουν απώλειες. Το TCP δεν αριθμεί τα datagrams αλλά τα octets. Έτσι αν υπάρχουν 500 octets δεδομένων σε κάθε datagram, το πρώτο datagram θα έχει αριθμό 0 το δεύτερο 500, το επόμενο 1000 κ.ο.κ. Τέλος, θα αναφερθεί το checksum. Το checksum είναι ένας αριθμός που υπολογίζεται προσθέτοντας όλα τα octets σε ένα datagram. Το αποτέλεσμα μπαίνει στην επικεφαλίδα. Το TCP στο άλλο άκρο υπολογίζει ξανά το checksum. Αν τα δύο checksum δεν συμφωνούν, τότε κάτι έχει συμβεί στο datagram κατά τη διάρκεια της μεταφοράς και απορρίπτεται από το λαμβάνων σύστημα.

Στο παρακάτω σχήμα φαίνεται η τελική μορφή του datagram.



Αν θεωρήσουμε "T" τη συντομογραφία της επικεφαλίδας του TCP το αρχείο μας θα έχει τελικά τη μορφή:

T*** T*** T*** T*** T*** T*** T*** T*** T*** T***

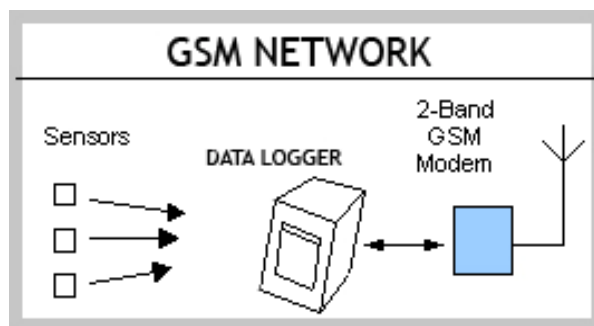
Σημειώνεται, ότι υπάρχουν κομμάτια της επικεφαλίδας που δεν έχουν περιγραφεί. Γενικά, αυτά έχουν να κάνουν με την διαχείριση της σύνδεσης. Για να εξασφαλιστεί το γεγονός ότι το datagram έχει φτάσει στον προορισμό του, ο λήπτης στέλνει ένα "acknowledgment" (επιβεβαίωση). Αυτό είναι ένα datagram του οποίου το πεδίο "acknowledgment number" είναι συμπληρωμένο. Για παράδειγμα, στέλνοντας ένα πακέτο με ackn. 1500, σημαίνει ότι έχουν φτάσει όλα τα data έως το octet με νούμερο 1500. Αν ο αποστολέας δεν πάρει acknowledgment μέσα σε ένα συγκεκριμένο χρονικό διάστημα, αποστέλλει ξανά τα δεδομένα. Το πεδίο window χρησιμοποιείται για να ελέγχεται τα πόσα δεδομένα μπορεί να βρίσκονται υπό μεταφορά ταυτόχρονα. Προφανώς, είναι μη πρακτικό να περιμένουμε acknowledgment για κάθε datagram που έχει αποσταλεί για να στείλουμε το επόμενο. Κάτι τέτοιο θα επιβράδυνε πολύ την όλη διαδικασία. Σίγουρα όμως δεν θα μπορούσαμε να στέλναμε συνεχώς δεδομένα διότι ένας γρήγορος αποστολέας θα ξεπερνούσε τη δυνατότητα ενός αργού παραλήπτη να απορροφήσει τα δεδομένα. Έτσι κάθε άκρο δηλώνει πόσα νέα δεδομένα μπορεί να απορροφήσει βάζοντας το νούμερο των octets στο πεδίο "window". Όσο ο υπολογιστής λαμβάνει δεδομένα, το ποσό του χώρου που μένει ελεύθερος στο παράθυρό του μειώνεται. Όταν φτάσει στο μηδέν, ο αποστολέας πρέπει να σταματήσει. Όταν ο λήπτης επεξεργάζεται τα δεδομένα αυξάνει το παράθυρό του, δηλώνοντας ότι είναι έτοιμος να δεχτεί νέα δεδομένα. Συχνά το ίδιο datagram μπορεί να χρησιμοποιηθεί για να επιβεβαιώσει τη λήψη ενός συνόλου δεδομένων και να δώσει την άδεια για επιπρόσθετα νέα δεδομένα. Το πεδίο urgent επιτρέπει στο ένα άκρο να πεί στο άλλο να προχωρήσει στην επεξεργασία ενός συγκεκριμένου octet. Είναι επίσης χρήσιμο για το χειρισμό ασύγχρονων γεγονότων π.χ. διακοπή της εξόδου με την πληκτρολόγηση control χαρακτήρα ή άλλης εντολής. Τα άλλα πεδία δεν θα μελετηθούν διότι κάτι τέτοιο ξεφεύγει από τους σκοπούς του κειμένου αυτού.

4 ΣΤΑΘΜΟΣ ΜΕΤΡΗΣΗΣ ΧΕΡΣΟΝΗΣΟΥ

4.1 Γενική Περιγραφή

Ο σταθμός μέτρησης ήταν ήδη εγκατεστημένος σε χώρο της σχολής Τεχνολόγων Γεωπονίας του ΤΕΙ Κρήτης στην περιοχή της Χερσονήσου και έκανε καταγραφή συνθηκών θερμοκρασίας, και βροχόπτωσης, έτσι το μόνο που έμενε ήταν να εγκαταστήσουμε τον ειδικό εξοπλισμό επικοινωνίας για να είναι διαθέσιμος ο σταθμός από το εργαστήριο μας, και να μην χρειάζεται πλέον η φυσική παρουσία στο χώρο μέτρησης, για να πάρει τα δεδομένα που είχαν αποθηκευτεί στην μνήμη του καταγραφικού μηχανήματος. Αλλά και σε περίπτωση κάποιας ελαφριάς βλάβης, θα μπορούσαμε να τον επαναφέρουμε. Ο σταθμός μέτρησης της Χερσονήσου λειτουργεί αυτόνομα, και από πλευρά επικοινωνίας αλλά και ενεργειακά καθώς είναι εξοπλισμένος με φωτοβολταϊκό πάνελ και μπαταρίες ώστε η λειτουργία του να γίνεται απρόσκοπτα.

Η επικοινωνία με τον server γίνεται μέσω του τηλεφωνικού δικτύου. Ο server “καλεί” τον datalogger, μέσω του Modem που είναι συνδεδεμένο πάνω του, και είναι εξοπλισμένο με κάρτα SIM εταιρίας κινητής τηλεφωνίας για data transfer.



Εικόνα 4.1: Παράδειγμα δικτύου με GSM Modem

4.2 Εξοπλισμός

Ο εξοπλισμός που χρησιμοποιήθηκε για τον σταθμό μέτρησης βασίστηκε στις απαιτήσεις του καταγραφικού μας μηχανήματος που είναι ο DataHog2 της εταιρίας skye με τα παρακάτω χαρακτηριστικά



Εικόνα 4.2: Το καταγραφικό DataHog2 της skye.

- Συμβατό με όλα τα δημοφιλή αισθητήρια.
- Αυτόνομη μονάδα, με διάρκεια μπαταριών για μεγάλο χρονικό διάστημα.
- Επικοινωνία μεσω RS232
- **Απομακρυσμένη πρόσβαση με GSM remote data link.**
- Δυνατότητα μέτρησης τάσης, έντασης, ψηφιακή είσοδος, αναλογικά αισθητήρια.

4.2.1 GSM MODEM - Wavecom Fastrack M1306B



Εικόνα 4.3: Το Modem που χρησιμοποιήσαμε

Μέσα από την πληθώρα των χαρακτηριστικών του, η επιλογή του έγινε για τις παρακάτω λειτουργίες, και ειδικά για το *Remote control by AT commands* κάτι που χρησιμοποιούμε συνέχεια για την επικοινωνία με το modem.

- Dual Band GSM/GPRS modem
- Input voltage: 5V-32V
- Full Rate, Enhanced Full Rate and Half
- Dual Tone Multi Frequency
- Function (DTMF)
- Fixed Dialling Number
- RS-232 and audio through mini sub-D
- **Remote control by AT commands**
- Baud rate from 300 to 115,200 bits/s
- Power supply through micro-FIT 4-pin connector
- SMA antenna connector
- Sliding SIM holder(3V/5V SIM interface)

Επίσης χρησιμοποιήσαμε μια εξωτερική GSM κεραία Triband 900/1800/1900 για να έχουμε καλύτερη λήψη σήματος, και άρα μικρότερη κατανάλωση στις μπαταρίες του σταθμού.

4.3 Εγκατάσταση

Η εγκατάσταση του modem στον σταθμό έγινε μέσα στο κουτί του datalogger, ώστε να παραμείνει ανέπαφο με το εξωτερικό περιβάλλον και να προστατευθεί από τις καιρικές συνθήκες. Η τροφοδοσία του modem γίνεται από την μπαταρία του σταθμού μέτρησης, που τροφοδοτείται από ένα φωτοβολταϊκό στοιχείο 12 volt.

Η κεραία εγκαταστάθηκε σε ύψος 3 μέτρα από το έδαφος στην κάτω πλευρά του φωτοβολταϊκού στοιχείου, ώστε να πετύχουμε την καλύτερη δυνατή λήψη σήματος από την εταιρία παροχής της υπηρεσίας.



Εικόνα 4.4: Μαζί με τους υπεύθυνους καθηγητές του προγράμματος της Σ.Τ.Ε.Γ.



Εικόνα 4.5: κατά την προσαρμογή της κεραίας στην πίσω όψη του φωτοβολταϊκού στοιχείου.

4.4 Λογισμικό- Πρωτόκολλα

Το κυρίως πρόγραμμα DialXer.exe που εκτελείται στο server για να παίρνει τα δεδομένα από το καταγραφικό γράφηκε σε Visual Basic έκδοσης 6, και βασίζεται στο ActiveComport ActiveX control της εταιρίας λογισμικού Macrovision. Μετά από δοκιμές σε διάφορα ActiveX κρίναμε πώς είναι το πιο σταθερό για τις ανάγκες μας.

Το DialXer.exe εκτελείται κάθε 4 ώρες ή ανάλογα με τις ανάγκες και τις ρυθμίσεις μας, κάνοντας κλήση στον απομακρυσμένο σταθμό. Μόλις έχουμε συγχρονισμό των modem, ο data logger απαντάει με τα δεδομένα που έχει στη μνήμη του και αυτά γράφονται σε ένα log αρχείο.

Σε περίπτωση αποτυχίας η κλήση θα επαναληφθεί στην επόμενη εκτέλεση της εφαρμογής.

Ο κώδικας της παραπάνω εφαρμογής βρίσκεται στο παράρτημα Ι (8^ο Κεφάλαιο)

Στη συνέχεια η διαδικασία είναι ίδια με την περίπτωση του σταθμού μέτρησης Σ.ΤΕ.Γ. Το log αρχείο χρησιμοποιείται σαν όρισμα σε παραλλαγή της εφαρμογής read.exe και για έξοδο παίρνουμε ένα αρχείο με ημερομηνία, ώρα και τις τιμές ανα κανάλι μέτρησης.

Στο παρακάτω αρχείο βλέπουμε την δομή του αρχείου καταγραφής πριν γίνει η εγγραφή στην βάση δεδομένων. Φαίνεται η ημερομηνία, η ώρα της μέτρησης, καθώς και οι τιμές των 3 καναλιών, μαζί με τον αριθμό του καναλιού να προηγείται.

out.txt

11102007	1230	01	25.2	02	0.0	03	0.0
----------	------	----	------	----	-----	----	-----

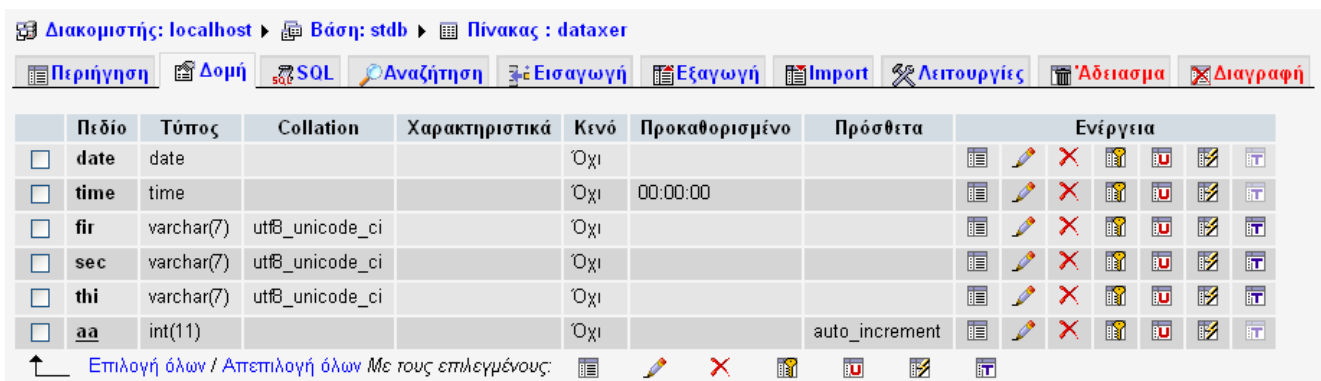
Έπειτα το sql αρχείο αναλαμβάνει να κάνει την εγγραφή στη βάση.

sqlcommand.sql

LOAD DATA LOCAL INFILE 'C:/DialXer/out.txt' INTO TABLE dataXer
--

4.5 Η Βάση Δεδομένων

4.5.1 Ο Πίνακας dataXer του σταθμού.



Πεδίο	Τύπος	Collation	Χαρακτηριστικά	Κενό	Προκαθορισμένο	Πρόσθετα	Ενέργεια
<input type="checkbox"/> date	date			Όχι			
<input type="checkbox"/> time	time			Όχι	00:00:00		
<input type="checkbox"/> fir	varchar(7)	utf8_unicode_ci		Όχι			
<input type="checkbox"/> sec	varchar(7)	utf8_unicode_ci		Όχι			
<input type="checkbox"/> thi	varchar(7)	utf8_unicode_ci		Όχι			
<input type="checkbox"/> aa	int(11)			Όχι		auto_increment	

Επιλογή όλων / Απεπιλογή όλων Με τους επιλεγμένους

Εικόνα 4.6: Η δομή του πίνακα dataXer όπως φαίνεται από το phrmyadmin.

Ο πίνακας dataXer όπως και ο πίνακας data έχει σχεδιαστεί με την ίδια λογική, έτσι ώστε σε μελλοντική αλλαγή των ρυθμίσεων του καταγραφικού να μην επηρεαστεί η ροή των δεδομένων. Έτσι, όπως φαίνεται και στη πιο πάνω εικόνα, στην στήλη των πεδίων έχουμε στις δύο πρώτες θέσεις, το πεδίο date, και το πεδίο time με default μορφή ddmmyyyy και hh:mm:ss αντίστοιχα.

Στη συνέχεια είναι τα πεδία των καναλιών fir, sec, thi τύπου varchar(7) ώστε να μην έχουμε κάποιο σφάλμα σε περίπτωση που το καταγραφικό μας, αντί για αριθμό στείλει κάποιο χαρακτήρα (flag), αντί για αριθμητικό ψηφίο. Το μήκος σε 7 χαρακτήρες ορίστηκε γιατί πάντα τα δεδομένα είναι της μορφής xxx.yyy.

Στο τελευταίο πεδίο έχουμε το πεδίο aa για την αρίθμηση των εγγραφών στη βάση με extra τον χαρακτηρισμό του, και εδώ, ως auto_increment.



Global System for Mobile communications

Το Ευρωπαϊκό Τηλεπικοινωνιακό Συμβούλιο το 1982, άρχισε την μελέτη για την δημιουργία ενός κοινού Ευρωπαϊκού ψηφιακού συστήματος κινητής τηλεφωνίας δεύτερης γενιάς (2G) και τότε αυτό το σύστημα ονομάστηκε αρχικά Group Special Mobile (GSM). Το GSM είναι ένα κυψελοειδές ψηφιακό σύστημα κινητής τηλεφωνίας δεύτερης γενιάς (2G) κάνοντας χρήση ηλεκτρομαγνητικών σημάτων χρησιμοποιώντας την τεχνική πολλαπλής πρόσβασης με διαχωρισμό του διαθέσιμου φάσματος συχνοτήτων σε ένα αριθμό καναλιών και την διαίρεση αυτών σε χρονοθυρίδες για την μετάδοση σημάτων. Το 1989 η ευθύνη του GSM ανατέθηκε στο Ευρωπαϊκό Τηλεπικοινωνιακό Ινστιτούτο Προτύπων (ETSI) και το 1990 ανακοινώθηκε επίσημα για πρώτη φορά το πρότυπο και τα χαρακτηριστικά του GSM. Το 1991 λοιπόν άρχισε η εμπορική του διάθεση στην Ευρώπη, ενώ στην Ελλάδα το 1993 από την WIND (πρώην TIM ή πρώην TELESTET).

Το πρότυπο GSM δεν ήταν μόνο ένα Ευρωπαϊκό πρότυπο μόνο, αφού υιοθετήθηκε από πολλές άλλες χώρες των άλλων Ηπείρων, εκμεταλεύοντας διάφορες ζώνες συχνοτήτων.

Ζώνες Συχνοτήτων

GSM 900

Τα πρώτα δίκτυα GSM το 1990 άρχισαν να λειτουργούν στη ζώνη συχνοτήτων των 900MHz. Η Διεθνής Ένωση Τηλεπικοινωνιών (ITU) παραχώρησε ένα ζεύγος συχνοτήτων, από τα 890 έως τα 915 MHz και από τα 935 έως τα 960 MHz. Η πρώτη περιοχή χρησιμοποιείται για την επικοινωνία του κινητού με τον σταθμό βάσης (Uplink), ενώ η δεύτερη για την επικοινωνία του σταθμού βάσης με το κινητό (downlink). Οι περιοχές(ζώνες) των 25MHz υποδιαιρούνται η καθεμία σε 124 + (1 ελεύθερο) κανάλια συχνότητας και κάθε κανάλι έχει εύρος ζώνης 200 KHz. Όλο αυτό το σύστημα ονομάστηκε GSM 900 ή Standard GSM.

GSM 1800

Στη συνέχεια, το 1991, αναπτύχθηκε το DCS 1800 σύστημα, όπου διατηρείται η δομή ενός GSM 900 δικτύου αλλά χρησιμοποιούνται διαφορετικά ζεύγη συχνοτήτων, από τα 1710 έως τα 1785 MHz Uplink και από τα 1805 έως τα 1880 MHz Downlink. Οι περιοχές των 75MHz υποδιαιρούνται η καθεμία σε 374+ (1 ελεύθερο) κανάλια και κάθε κανάλι έχει εύρος ζώνης 200KHz. Αυτή η αλλαγή στην ζώνη συχνοτήτων έγινε διότι οι ζώνες του GSM 900 στην Ευρώπη ήταν πιασμένες από άλλους παροχής κινητής τηλεφωνίας. Όπως και στην χώρα μας σήμερα όλες οι εταιρίες κινητής τηλεφωνίας χρησιμοποιούν και τα δύο συστήματα(GSM 900/GSM 1800) στα δίκτυα τους αυξάνοντας αισθητά τη χωρητικότητά στα δίκτυα τους. Στα τέλη δεκαετίας του 1990 το GSM World Association αποφάσισε να μετονομάσει το DCS 1800 σε GSM 1800 για να φανεί η δυναμικότητα και η παγκοσμιότητα του GSM.

GSM 1900

Στο GSM 1900 χρησιμοποιείται σε αρκετές χώρες της Αμερικής, διατηρείται και πάλι η δομή ενός GSM 900 δικτύου, αλλά χρησιμοποιούνται και εδώ διαφορετικά ζεύγη συχνοτήτων: Από τα 1850 έως τα 1910 MHz για Uplink και από τα 1930 έως τα 1990 MHz για Downlink. Οι περιοχές των 60MHz υποδιαιρούνται η καθεμία σε 299+ (1 ελεύθερο) κανάλια συχνότητας και κάθε κανάλι έχει εύρος ζώνης 200KHz. Στα τέλη δεκαετίας του 1990 το GSM World Association απεφάσισε να

μετονομάσει το PCS 1900 που λεγότανε παλιότερα σε GSM 1900 για να φανεί η δυναμικότητα και η παγκοσμιότητα του GSM.

E-GSM • Extended-GSM 900 - Εκτεταμένη ζώνη GSM

Το E-GSM καθορίστηκε από την Ευρωπαϊκή Επιτροπή Ράδιο Επικοινωνιών στα τέλη της δεκαετίας του 1990 για να «αντικαταστήσει» το κλασικό GSM 900 διατηρώντας βέβαια την δομή του αυξάνοντας όμως τις περιοχές συχνοτήτων από 880 έως 915 MHz για Uplink και 925 έως 960 MHz Downlink. Έτσι επέτρεψε στα δίκτυα κινητής τηλεφωνίας να αυξηθούν τη χωρητικότητά τους και να καλύψουν τις ανάγκες από την αυξημένη κίνηση των πελατών τους.

Κυψελοειδής Δομή Δικτύου

Η εμβέλεια ενός δικτύου GSM σε μία γεωγραφική περιοχή για να γίνει, η περιοχή αυτή διαμελίζεται σε μικρότερες περιοχές που λέγονται κυψέλες, οι οποίες εφάπτονται μεταξύ τους με κάθε κυψέλη να έχει και ένα σταθμό βάσης (Base Station), συνθέτοντας έτσι μια δομή κυψελών. Η δομή αυτή επαναλαμβάνεται όσες φορές χρειάζεται για την απαιτούμενη κάλυψη της μιας περιοχής κάνοντας επαναχρησιμοποίηση συχνοτήτων. Με την μέθοδο αυτή αυξάνεται η χωρητικότητα του δικτύου αλλά πρέπει η ισχύς κάθε κυψέλης να είναι όση χρειάζεται ώστε να μην ξεπερνάει τα όρια της και να υπερχειλίζει άλλες κυψέλες της ίδιας δομής ενώ για να μην δημιουργείται ενδοκαναλική παρεμβολή σε γειτονικές κυψέλες η επαναχρησιμοποίηση συχνοτήτων πρέπει να σχεδιάζεται έτσι ώστε να απέχουν επαρκή απόσταση οι κυψέλες μιας δομής που έχουν την ίδια συχνότητα με τις κυψέλες μιας άλλης δομής. Η ενδοκαναλική παρεμβολή μειώνεται όσο αυξάνει ο αριθμός των κυψελών της δομής. Η ακτίνα κάθε κυψέλης σε αραιοκατοικημένες περιοχές είναι έως και 35Km ενώ σε πυκνοκατοικημένες περιοχές δεν ξεπερνά τα 300 μέτρα. Σε περιοχές με πολύ μεγάλη ζήτηση χωρητικότητας δικτύου όπως σε αστικά κέντρα, οι σταθμοί βάσης υπερφορτώνονται και έτσι υπάρχει ανάγκη για μεγαλύτερη χωρητικότητα του δικτύου. Έτσι για να επιτευχθεί αυτός ο σκοπός γίνεται διάσπαση των υπάρχοντων κυψελών σε μικρότερες, ενώ γι' αυτές χρησιμοποιούνται κεραίες μικρότερης ισχύος (macro bs - micro- bs - pico bs) όπως σε κτίρια, στο μετρό, Δημόσιους Οργανισμούς, οδικές αρτηρίες κτλ..

Αρχιτεκτονική

Ένα GSM δίκτυο χωρίζεται σε 3 βασικά μέρη:

1) Τον Κινητό Σταθμό (Mobile Station): Έχει οπωσδήποτε πομπό-δέκτη, κεραία, οθόνη και την κάρτα SIM. Η μέγιστη επιτρεπόμενη ισχύς εκπομπής στην Ευρώπη μιας κινητής μονάδας είναι στα 2 Watt ενώ σε Αυστραλία και Αμερική είναι 1,6W, οι τιμές αυτές καθορίστηκαν από την Διεθνή Επιτροπή για την προστασία από τη μη ιονίζουσα ακτινοβολία.

2) Το Βασικό Υποσύστημα Σταθμού (Base Station Subsystem): Το BSS διαχειρίζεται τις κλήσεις σε μια γεωγραφική περιοχή όπου καλύπτεται από ένα σύνολο κεραιών διαφόρων μεγεθών σε σειρά σαν αυτούς που βλέπουμε σε λόφους, ταράτσες πολυκατοικιών-εταιριών-σχολείων-οργανισμών κτλ. και κάθε τέτοια κεραία εξυπηρετεί και από μια κυψέλη. Το BSS χωρίζεται στο βασικό σταθμό πομπό-δέκτη Base Transceiver Station (BTS) και στο βασικό σταθμό ελέγχου Base Station Controller (BSC).

• Το Βασικό Υποσύστημα Σταθμού (BTS) φροντίζει την επικοινωνία μεταξύ του δικτύου GSM και του κινητού σταθμού. Ένα BTS μπορεί να ελέγχει μια ή περισσότερες κεραίες. Η ισχύς των κεραιών σε ένα BTS μπορεί να είναι 40W έως 500W. Όταν ένας χρήστης A θέλει να πραγματοποιήσει μια κλήση σε έναν άλλο συνδρομητή B, ο σταθμός βάσης μεταβιβάζει το σήμα με το αίτημά του A για αναζήτηση και εντοπισμό του άλλου συνδρομητή B στο τηλεπικοινωνιακό κέντρο της εταιρείας του A. Το κέντρο της εταιρείας εντοπίζει την κυψέλη στην οποία βρίσκεται ο

B και στέλνει το σήμα στον πλησιέστερο σταθμό βάσης. Από εκεί, πάλι με τη χρήση των διαθέσιμων συχνοτήτων, στέλνεται το σήμα στο κινητό του B κι έτσι μπορεί να επικοινωνήσει μαζί του ο A. Το πεδίο μιας GSM κεραίας ενός σταθμού βάσης ή κινητής μονάδας, είναι παλμικό με κανάλια διάρκειας 4,616 ή 9,232 msec το καθένα, που είναι χωρισμένα σε 8 ή 16 διαστήματα-χρονοθυρίδες, διάρκειας 0.577 msec η καθεμία (8X0,577 ή 16X0,577) . Κάθε χρήστης χρησιμοποιεί για μια τηλεφωνική κλήση από μια χρονοθυρίδα άρα ένα κανάλι μπορεί να χρησιμοποιηθεί μέχρι και από 8 ή 16 συνδρομητές. Οι 8 ή 16 χρονοθυρίδες που χωρίζονται σε ένα κανάλι αποκαλούνται πλαίσιο TDMA ενώ κάθε χρονοθυρίδα αντιστοιχεί σε 156 bits.

- Το BSC (βασικός σταθμός ελέγχου) ελέγχει τα σήματα παίρνοντας τα από ένα ή περισσότερα BTS ενώ εκχωρεί και απελευθερώνει κανάλια. Τα σήματα που λαμβάνει τα κατευθύνει στο MSC-Mobile Switching Centre και όταν χρειάζεται μετατρέπει τα 16kbps φωνής που είναι στην κινητή τηλεφωνία σε 64kbps που χρησιμοποιείται στην σταθερή τηλεφωνία.(σχήμα)

3) Το Υποσύστημα Δικτύου μεταγωγής (NNS- Network Switching Subsystem) που αποτελείται από:

Το Κέντρο Διαμονής (Mobile Switching Center), είναι υπεύθυνο για την διασύνδεση, τον έλεγχο και την δρομολόγηση εισερχόμενων/εξερχόμενων κλήσεων μεταξύ του δικτύου κινητής τηλεφωνίας και ενός άλλου δικτύου ή άλλων. Όταν ένα MSC συνδέεται με ένα δίκτυο σταθερής τηλεφωνίας θα πρέπει να δέχεται 64kbps φωνής. Όταν όμως ο MSC συνδέεται με ένα δίκτυο κινητής τηλεφωνίας τότε θα πρέπει να γνωρίζει που βρίσκεται εκείνη τη δεδομένη χρονική στιγμή ο χρήστης, αυτό επιτυγχάνεται με την βοήθεια καταχωρητών VLR (Visitor Locator Register), Home Locator Register (HLR). Ο πάτριος καταχωρητής θέσης αναζήτησης ή τοπικά κέντρα εγγραφής-HLR έχει μια Βάση Δεδομένων που κρατά στοιχεία προφίλ ενός συνδρομητή και πληροφορίες για την τρέχουσα θέση του, κάθε τέτοιο κέντρο η εμβέλεια του είναι σε τοπικό επίπεδο. Έτσι π.χ. όταν ένας συνδρομητής από το Πέραμα το HLR του χρήστη είναι το "HLR Πέραμα", επίσης σε μια πιο πυκνοκατοικημένη περιοχή μπορεί να υπάρχουν περισσότερα από ένα τοπικά κέντρα εγγραφής πχ. το Περιστερί. Ο καταχωρητής θέσης αναζήτησης επισκεπτών ή εικονικό κέντρο εγγραφής χρήστη (VLR): Όταν ο συνδρομητής βγει από τα όρια της τοπικής περιοχής που καλύπτει το HLR δηλαδή είναι πολύ μακριά από το σπίτι του τότε αναλαμβάνει τον χρήστη ο καταχωρητής θέσης αναζήτησης ή εικονικό κέντρο εγγραφής - VLR ο οποίος έχει μια βάση δεδομένων, ο οποίος συγκρατεί προσωρινά δεδομένα καθώς και την τρέχουσα θέση του, αναλαμβάνοντας τις κλήσεις του καλύτερα κατά τις ώρες αιχμής στο κέντρο της πόλης. Το κέντρο πιστοποίησης (Authentication Centre – AuC) ο ρόλος του οποίου έγκειται στη διαχείριση δεδομένων για την πιστοποίηση της ταυτότητας του χρήστη.

AT Command List

Κάποιες από τις βασικές AT εντολές που χρησιμοποιήθηκαν στο λογισμικό που γράφτηκε για την επικοινωνία με το σταθμό μέτρησης. Γενικά οι εντολές AT είναι εντολές που δίνονται στην command line ώστε να να πραγματοποιηθεί η περιγραφή τους, επίσης μπορούν να χρησιμοποιηθούν για να γίνει αλλαγή των ρυθμίσεων του Modem

AT A Απάντηση. Το modem θα βγει σε off-hook κατάσταση και θα προσπαθήσει να απαντήσει την κλήση.

AT D Κλήση. Η εντολή βγάζει το modem σε off-hook, και κάνει κλήση ανάλογα με τι συνεχίζετε. Σε περίπτωση που δεν του δώσουμε τηλεφωνικό αριθμό .

Παράδειγμα

```
ATD6977376650;
```

Πιθανοί παράμετροι της εντολής ATD είναι:

0-9 Tone and pulse dial digits 0 to 9.

L επανάκληση του τελευταίου αριθμού

Παράδειγμα

```
ATDL;
```

P Επιλογή παλμικής γραμμής

T Επιλογή τονικής γραμμής

W Αναμονή για ήχος κλήσης

@ Αναμονή για σιγή. Το modem περιμένει για τουλάχιστον 5 δευτερόλεπτα πριν κάνει την κλήση.

; Επιστροφή σε κατάσταση αναμονής εντολών

AT E0 Ακύρωση του echo στο terminal window του υπολογιστή.

AT E1 Ενεργοποίηση του echo στο terminal window.

AT H Διακοπή, και κλείσιμο της γραμμής

AT L0 Speaker off.

5 ΣΥΜΠΕΡΑΣΜΑΤΑ ΚΑΙ ΙΔΕΕΣ

5.1 Ιδέες που απορρίψαμε

Για να υλοποιήσουμε το παραπάνω εγχείρημα με την μορφή που παρουσιάζετε σε αυτή την πτυχιακή εργασία, προτείναμε και κάποιες λύσεις, τις οποίες στο τέλος απορρίψαμε αιτιολογώντας το γιατί.

5.1.1 Ασύρματο Δίκτυο 2.4 GHz

Η χρήση αυτή θα μας επέτρεπε την εύκολη διασύνδεση του datalogger (3) με τον κεντρικό υπολογιστή αλλά λόγω του ότι δεν υπάρχει οπτική επαφή μεταξύ τους η εφαρμογή θα απαιτούσε εγκατάσταση repeater κάτι που θα ανέβαζε το ήδη αυξημένο κόστος σε αρκετά υψηλά επίπεδα.

5.1.2 Δορυφορική σύνδεση του μέσω τεχνολογίας VSAT

Αν και η χρήση αυτής της τεχνολογίας θα μας έδινε πολλές δυνατότητες για ανάπτυξη του συστήματος σε μελλοντικό σχεδιασμό, απορρίφθηκε για τρεις λόγους, ο κυριότερος είναι το κόστος, που ανέβαινε πάρα πολύ, το ότι θα χρειαζόταν σε κάθε σταθμό μέτρησης και ένας υπολογιστής, αλλά και το ότι χρειάζεται μεγάλη ποσότητα ενέργειας, που εμείς δεν μπορούμε ένα έχουμε σε ένα σταθμό στο βουνό. Αξίζει να αναφερθεί πως με την χρήση τέτοιας τεχνολογίας θα μπορούσαμε να αναπτύξουμε συστήματα άλλης φύσεως εκτός των μετεωρολογικών μετρήσεων, όπως υπηρεσίες internet (ISP), VoIP (Voice over IP), κ.α.

5.1.3 UHF 3.4 GHz

Η τεχνολογία αυτή θα μπορούσε να εφαρμοστεί σε όλες τις πιο πάνω ζεύξεις καθώς μπορεί να καλύψει περιοχές μεγάλων αποστάσεων. Όμως η χρήση τέτοιων συστημάτων απαιτεί ειδική άδεια καθώς η συγκεκριμένη band συχνοτήτων χρησιμοποιείτε ευρέως. Επίσης άλλο πρόβλημα που θα προέκυπτε, είναι το μεγάλο μέγεθος από εσφαλμένα δεδομένα (σκουπίδια) που θα παίρναμε λόγω του θορύβου από παρεμβολές καθώς αυτή η band είναι ελεύθερη για όλους. Το τελευταίο πρόβλημα θα μπορούσε να μειωθεί στο ελάχιστο με την χρήση κατευθυντικών κεραιών αλλά το κόστος τους είναι αρκετά υψηλό.

5.1.4 5.4 RS424

Ουσιαστικά αυτός ο τρόπος διασύνδεσης είναι ένας μετατροπέας RS232 σε RS424 ο οποίος επιτρέπει την μεταφορά δεδομένων σε μεγάλες αποστάσεις έως 2,5 χμ. χωρίς απώλειες. Αυτό όμως απαιτεί την χρήση ενός δικτύου όπου το σύστημα μας θα κάνει αποκλειστική χρήση. Η χρήση του τοπικού δικτύου απορρίφθηκε για την συγκεκριμένη τεχνολογία καθώς δεν γίνεται να έχουμε αποκλειστική χρήση του τοπικού δικτύου του TEI κάθε 10 λεπτά που θα παίρναμε δεδομένα από το καταγραφικό μας.

5.2 Συμπέρασμα

Από την χρήση των τεχνικών που αναπτύχθηκαν, μπορούμε να πούμε πως λιγότερο αξιόπιστο κρίθηκε ο σταθμός που έκανε χρήση του κυψελωτού δικτύου GSM, καθώς σε καταστάσεις χαμηλής στάθμης λήψης ή σε σύνδεση που ρυθμό μετάδοσης ήταν μικρότερος από τον απαιτούμενο, η επικοινωνία τερματιζόταν ανεπιτυχώς, με αποτέλεσμα να μην έχουμε δεδομένα στην βάση μας ενώ καταγράφονταν κανονικά από τον data logger.

Οι άλλες δύο τεχνικές (μεσω τοπικού δικτύου, και σειριακή) δοκιμάστηκαν και συνεχίζουν να λειτουργούν, χωρίς κανένα πρόβλημα. Οι όποιες απώλειες ή δυσλειτουργίες οφείλονται και μόνο σε εξωγενείς παράγοντες.

6 ΠΑΡΟΥΣΙΑΣΗ

6.1 Γενικά

Για την παρουσίαση των δεδομένων που καταγράφουμε από το δίκτυο σταθμών που αναπτύξαμε χρησιμοποιήσαμε μια γλώσσα προγραμματισμού που θα μας έδινε τις περισσότερες επιλογές παρουσίασης.

Ένας παρατηρητής μπορεί πιο εύκολα να βγάλει συμπεράσματα βλέποντας ένα σχήμα, μια εικόνα ή ένα γράφημα παρά όταν βλέπει ένα πίνακα με αριθμούς, ειδικά σε περιπτώσεις που ο όγκος των δεδομένων είναι τεράστιος. Έτσι για να βοηθήσουμε τον επισκέπτη της σελίδας μας να βγάλει χρήσιμα συμπεράσματα και να κατανοήσει καλύτερα το νόημα της πληροφορίας χρησιμοποιήσαμε την PHP για να παράγουμε δυναμικά γραφήματα. Η επιλογή της γλώσσας έγινε γιατί τα χαρακτηριστικά της όπως είναι η ευκολία επικοινωνίας με μια βάση δεδομένων, η δυναμική της και η ικανότητα της να δημιουργεί εικόνες «on the fly», μας οδήγησαν σε αυτή την επιλογή σχεδόν χωρίς δεύτερη σκέψη.

Εκτός όμως από την γλώσσα που θα χρησιμοποιούσαμε έπρεπε να επιλέξουμε και μια βιβλιοθήκη γραφημάτων που να μπορούσε να ανταποκριθεί στις ανάγκες μας και να εκμεταλλευόταν στο έπακρο την PHP. Έτσι γεννηθήκαν κάποια ερωτήματα σχετικά με τα χαρακτηριστικά των πακέτων δημιουργίας γραφημάτων (**Graphing Packages**) όπως

- Αν ανανεώνεται ένα τέτοιο πακέτο
- Αν είναι ελεύθερο προς χρήση – freeware
- Αν υπάρχει βιβλιογραφία και υποστήριξη από τους δημιουργούς
- Αν υπάρχουν βοηθήματα
- Τι τύπους γραφήματος μας παρέχει
- Αν ο κώδικας του πακέτου μπορεί εύκολα να ερμηνευτεί
- Και διάφορα άλλα λιγότερο σημαντικά ερωτήματα.

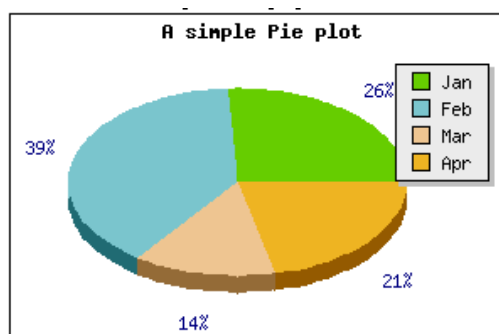
Έτσι καταλήξαμε στο πακέτο γραφικών **JpGraph** που μας έδινε περισσότερα από αυτά που θα χρειαζόμασταν, και ήταν ευρέως γνωστό στο κόσμο, οπότε θα είχαμε καλύτερη υποστήριξη. Επίσης είναι μια σουίτα με πάρα πολλά παραδείγματα δημιουργίας γραφημάτων, και κώδικα εύκολο προσαρμόσιμο στα δικά μας δεδομένα.

6.2 Τύποι Διαγράμματος

Οι επιλογές παρουσίασης ήταν πολλές καθώς υπάρχουν διάφοροι τύποι για να αναπαραστήσουν την πληροφορία. Μερικά παραδείγματα φαίνονται παρακάτω.

Διάγραμμα Πίτας

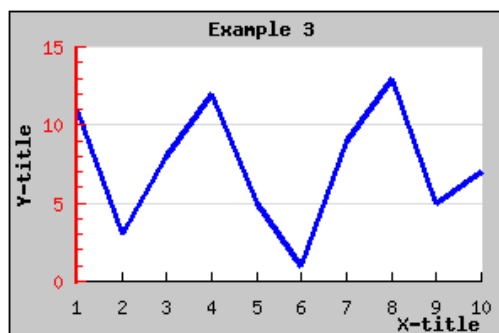
Ένα διάγραμμα αυτής της μορφής είναι γενικά ένας κύκλος χωρισμένος σε 2 ή περισσότερα μέρη που ονομάζονται φέτες, και είναι συνήθως διαφορετικά χρωματισμένα αναπαριστούν το σχετικό μέγεθος της πληροφορίας σε ποσοστά.



Εικόνα 6.1: Παράδειγμα διαγράμματος πίτας

Γράφημα Γραμμής

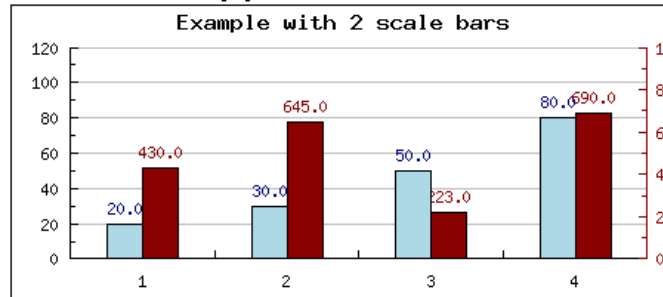
Σε αυτού του είδους τα γραφήματα, που είναι και τα πιο συνηθισμένα τα δεδομένα αναπαρίστανται σε δύο άξονες χ, ψ.



Εικόνα 6.2: Παράδειγμα γραφήματος γραμμής

Γράφημα Μπάρας

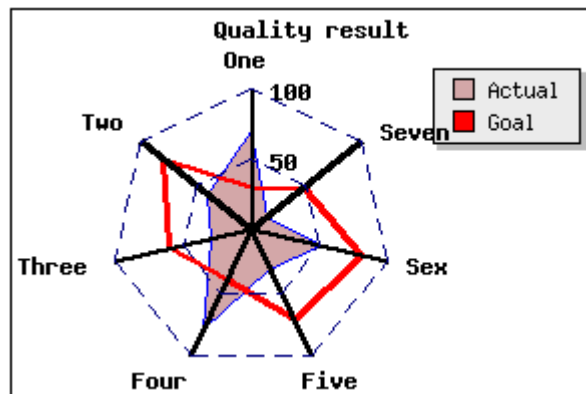
Στα γραφήματα μπάρας χρησιμοποιούμε ορθογώνια για να προσδιορίσουμε το μέγεθος της μέτρησης. Αυτού του είδους τα γραφήματα, θεωρούνται κατατοπιστικά για τον αναγνώστη γιατί γίνεται αναφορά για την τιμή της κάθε μπάρας ξεχωριστά, άρα είναι πιο εύκολο κάποιος να βγάλει χρήσιμα συμπεράσματα. Τα γραφήματα μπάρας, συχνά αναπαριστούν κατηγορίες ή ομάδες δεδομένων σε σχέση με τον χρόνο, time- series data.



Εικόνα 6.3: Παράδειγμα γραφήματος μπάρας

Γράφημα Ραντάρ

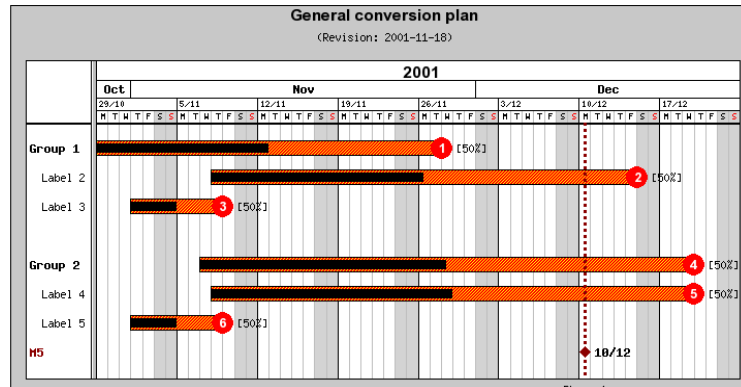
Τα γραφήματα ραντάρ παρουσιάζουν δεδομένα σε πολλούς άξονες ξεκινώντας από την μέση κάθε φορά. Οι ξεχωριστές τιμές των δεδομένων συνήθως ενώνονται με γραμμές και το πολύγωνο που δημιουργείται χρωματίζεται για καλύτερο τονισμό. Τα γραφήματα αυτά είναι γνωστά και ως γράφημα αράχνη.



Εικόνα 6.4: Παράδειγμα γραφήματος ραντάρ

Διαγράμματα Gantt

Αυτού του είδους τα διαγράμματα παρουσιάζουν χρησιμότητα σε προγραμματισμό ενεργειών με αρχή και τέλος κάθε φορά ανάλογες ημερομηνίες. Ο οριζόντιος άξονας αναπαριστά το χρόνο και ο κάθετος γενικά αναπαριστά πολλαπλές διεργασίες ενός project.



Εικόνα 6.5: Παράδειγμα διαγράμματος gantt

Υπάρχουν και άλλοι τύποι διαγράμματος καθώς και πιο εξελιγμένα, όπως είναι τα γραφήματα 3 διαστάσεων. Εδώ κάναμε μια αναφορά στα πιο ευρέως χρησιμοποιούμενα.

Πριν προχωρήσουμε όμως θα πρέπει να γίνεται διαχωρισμός στις έννοιες «Διάγραμμα» και «Γράφημα», επειδή πολύ συχνά συγχέονται ενώ στην πραγματικότητα είναι διαφορετικά πράγματα. Διάγραμμα είναι ένας γενικός όρος για κάθε γραφική πληροφορία. Ένα διάγραμμα είναι το μέσο για να ομαδοποιήσουμε και να παρουσιάσουμε την πληροφορία με σκοπό την ανάλυση, τη προβολή, τη πρόβλεψη ή την επικοινωνία. Ένα γράφημα είναι ειδικός τύπος διαγράμματος.

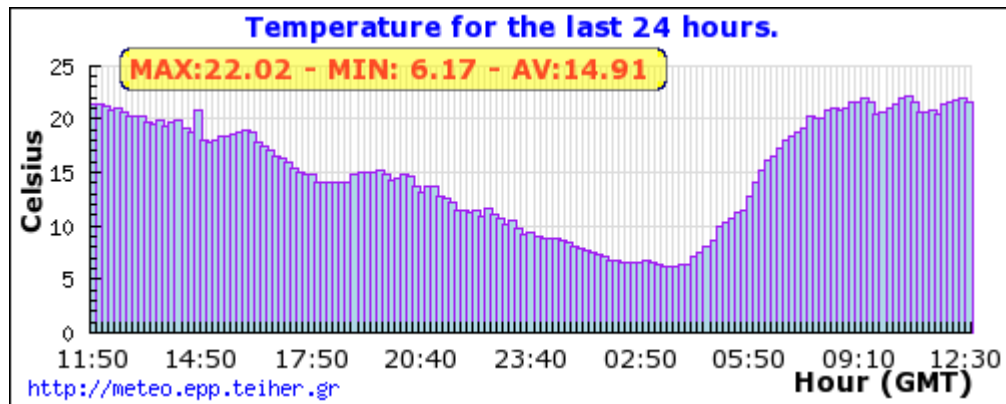
6.3 Τα γραφήματα μας

Για την παρουσίαση των δεδομένων μας, αξιολογήσαμε πρώτα πώς θα ήταν πιο εύκολο για έναν αναγνώστη ή επισκέπτη της ιστοσελίδας μας να εξάγει κατάλληλα συμπεράσματα από την παρουσίαση, ποια είναι τα διεθνή δεδομένα για αντίστοιχες εργασίες συνυπολογίζοντας και των όγκο των δεδομένων.

Έτσι καταλήξαμε στο να παράγουμε τις μετεωρολογικές μετρήσεις ανά αισθητήριο και ένα γράφημα, και στο τέλος κάποια συνδυαστικά όπως το γράφημα θερμοκρασίας και ταχύτητας ανέμου.

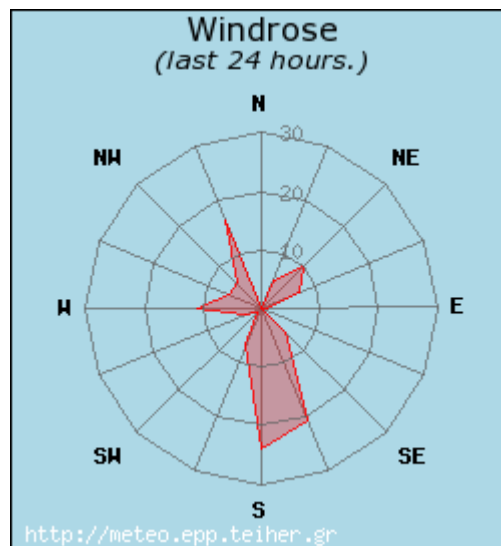
Η προβολή των δεδομένων γίνεται ανά 24 ώρες, δηλαδή μόλις κάποιος επισκεφτεί την ιστοσελίδα του εργαστηρίου, δημιουργούνται εκ νέου τα γραφήματα και ο επισκέπτης λαμβάνει δεδομένα για τις τελευταίες 24 ώρες. Ο αριθμός των τιμών που προβάλλονται σε κάθε γράφημα είναι 144 και αυτό γιατί έχουμε δειγματοληψία από κάθε σταθμό ανα 10 λεπτά, άρα 6 φορές σε μια ώρα. Έτσι για ένα ολόκληρο 24ωρο φτάνουμε τις 144 τιμές.

Είναι σημαντικό να αναφερθεί πώς τα γραφήματα δημιουργούνται με κάθε επίσκεψη που έχουμε στην ιστοσελίδα μας, και δεν υπάρχει κάποια διεργασία στο server που δημιουργεί τα γραφήματα, καταναλώνοντας επεξεργαστική ισχύς και πόρους συστήματος χωρίς να υπάρχει λόγος. Τα γραφήματα δημιουργούνται on the fly, όπως ορίζετε και από την PHP, ανακαλώντας τιμές από την βάση κάθε φορά.



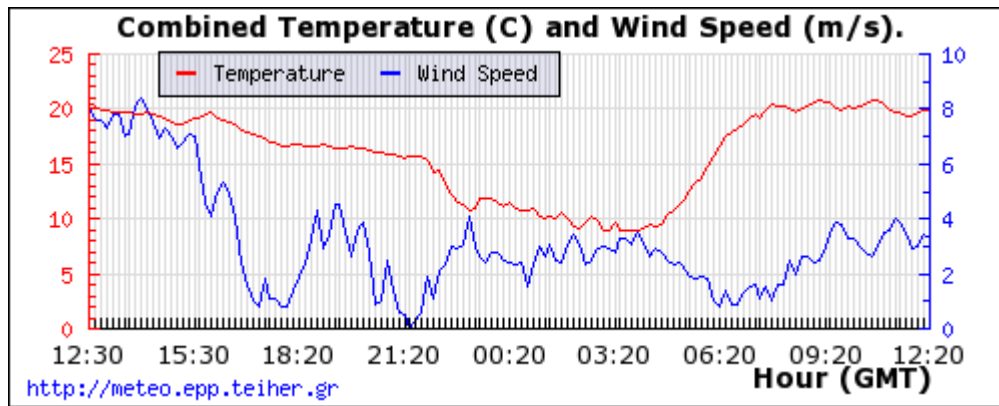
Εικόνα 6.6: Γράφημα θερμοκρασίας για ένα 24ωρο.

Οι κώδικες των γραφημάτων δίνονται το Παράρτημα II στο 8^ο Κεφάλαιο

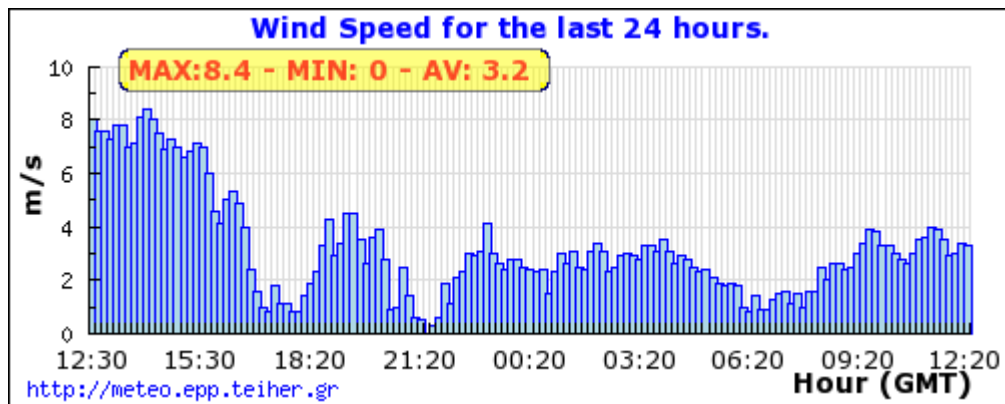


Εικόνα 6.7: Στατιστικό για την διεύθυνση ανέμου για ένα 24ωρο.

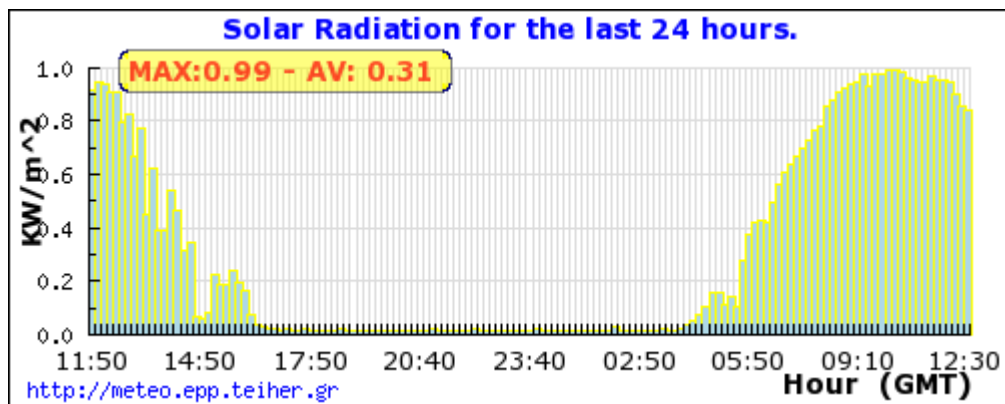
Για κάθε γράφημα υπάρχει η μέση τιμή, η μέγιστη αλλά και η ελάχιστη για τις μετρήσεις που προβάλλονται κάθε φορά, εκτός από το παραπάνω γράφημα της διεύθυνσης ανέμου.



Εικόνα 6.8: Γράφημα συνδυασμού Ταχύτητας Ανέμου και Θερμοκρασίας για ένα 24ωρο.



Εικόνα 6.9: Γράφημα Ταχύτητας Ανέμου για ένα 24ωρο.

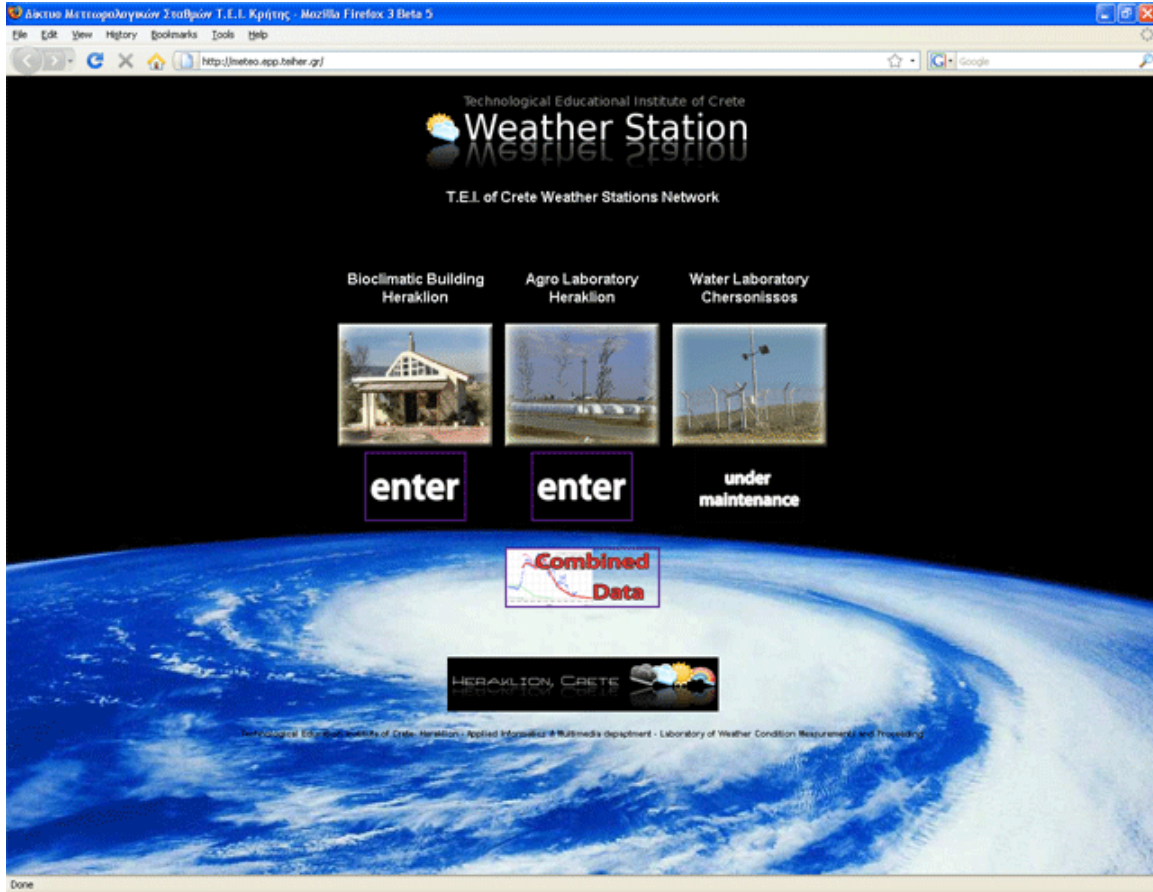


Εικόνα 6.10: Γράφημα Ηλιακής Ακτινοβολίας για ένα 24ωρο.

Τα γραφήματα είναι αρχεία PHP όπου εύκολα προσαρμόζονται μέσα στο κώδικα απλής HTML με image tags όπως φαίνεται παρακάτω.

```
<html>  
  
</html>
```

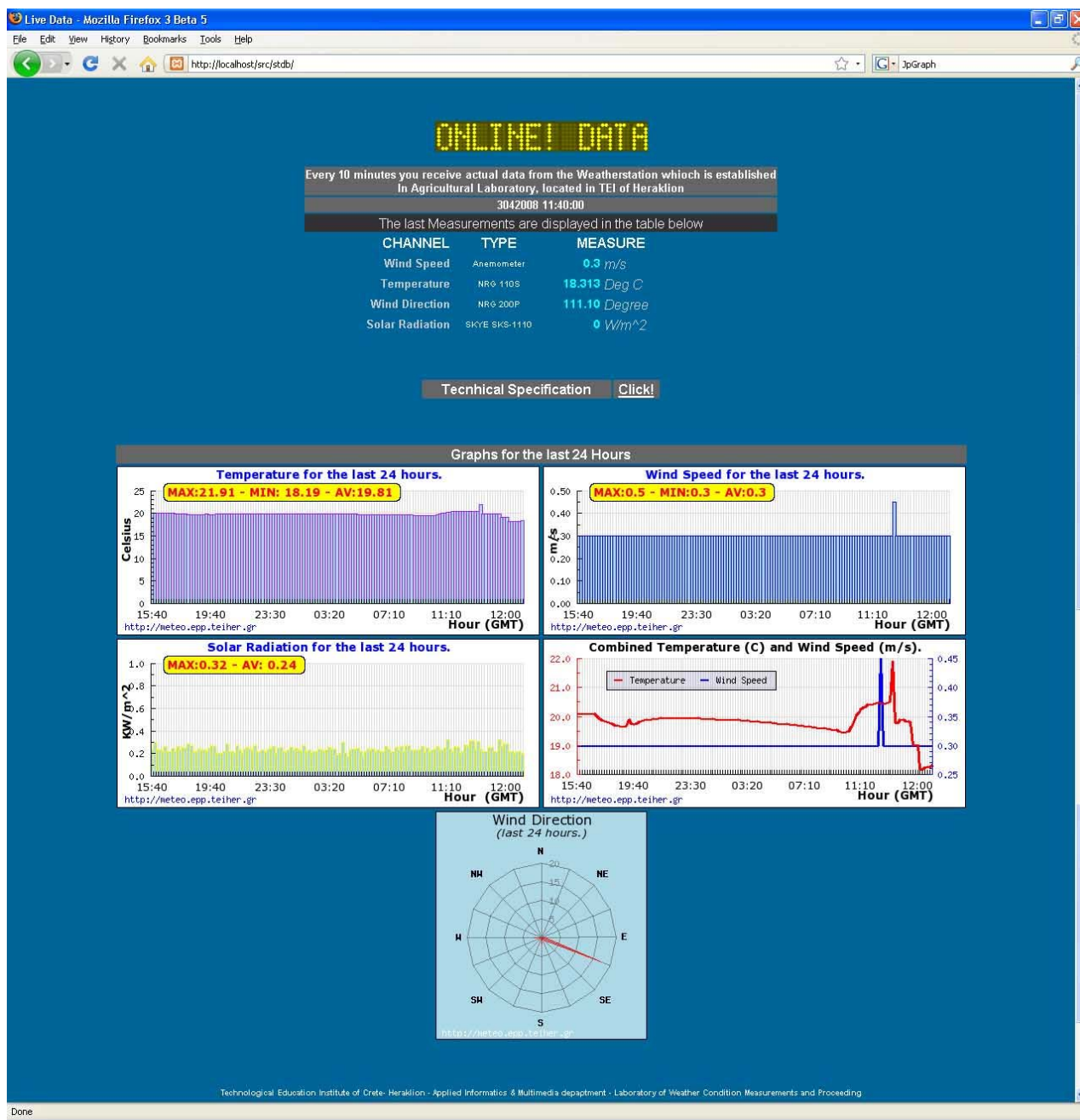
6.4 Η Ανανεωμένη Ιστοσελίδα μας



Εικόνα 6.11: Η αρχική ιστοσελίδα του Δικτύου Μετεωρολογικών Σταθμών Τ.Ε.Ι. Κρήτης.

Στην αρχική σελίδα δίνεται η επιλογή στο επισκέπτη να επισκεφτεί τον κάθε σταθμό μέτρησης ξεχωριστά, να βρει τα τεχνικά χαρακτηριστικά του κάθε σταθμού, και να δει τα γραφήματα των μετρήσεων, καθώς και τις μετρήσεις τις στιγμής της επίσκεψης.

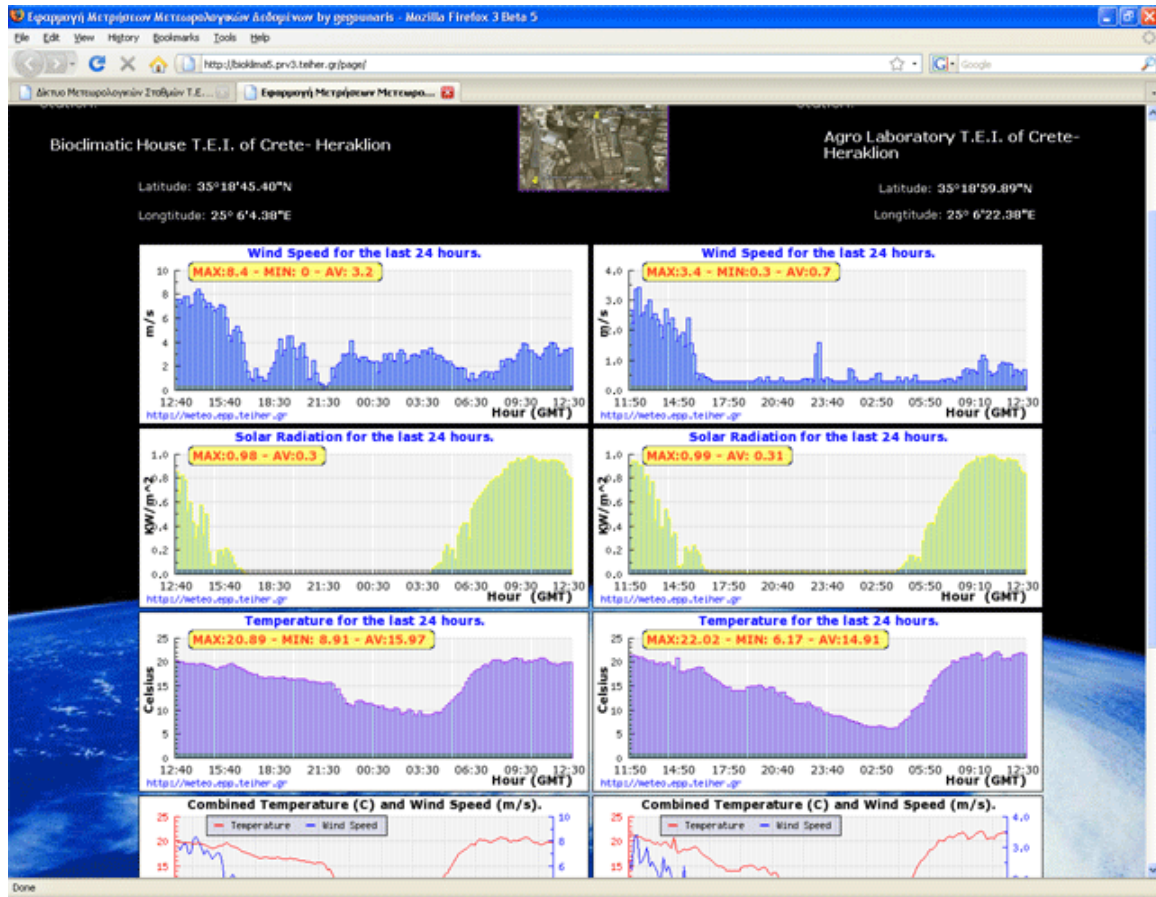
Παρακάτω φαίνονται εικόνες τραβηγμένες από την σελίδα μας, και τις επιλογές που έχει ο επισκέπτης μας.



Εικόνα 6.12: Η Ιστοσελίδα του σταθμού μέτρησης του Αγροκτήματος ΣΤΕΓ.

Οι πληροφορίες που παίρνει ο αναγνώστης μας από την σελίδα έχουν να κάνουμε με τις μετρήσεις και τις συνθήκες που επικρατούν την χρονική στιγμή που επισκέπτεται τον ιστοτόπο μας, για τα τεχνικά χαρακτηριστικά του σταθμού και τι εξοπλισμό χρησιμοποιούμε, και τέλος τα γραφήματα θερμοκρασίας, ταχύτητας ανέμου, ηλιακής ακτινοβολίας, στατιστικού διεύθυνσης

ανέμου, καθώς και το γράφημα συνδυασμού ταχύτητας ανέμου και θερμοκρασίας. Όλα τα γραφήματα αναφέρονται στις μετρήσεις του τελευταίου 24ώρου.



Εικόνα 6.13: Η σελίδα όπου γίνεται αντιπαράθεση των μετρήσεων των σταθμών.

7 ΕΡΓΑΛΕΙΑ

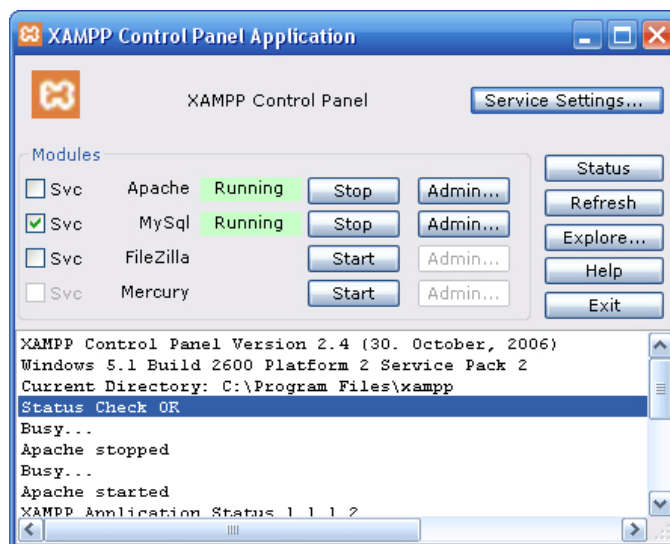
Για την υλοποίηση της πτυχιακής εργασίας χρησιμοποιήθηκαν τα ακόλουθα εργαλεία:



Η εφαρμογή XAMPP είναι ένας εύκολος τρόπος εγκατάστασης και ρύθμισης του apache και του MySQL server.

Η κονσόλα του Xampp μας δίνει τη δυνατότητα να ξεκινήσουμε τις 4 εφαρμογές (για την εργασία θα χρησιμοποιήσουμε τις 2 από τις 4 εφαρμογές: τον apache server και τον MySQL server).

Έχει το πλεονέκτημα κλείνοντας το παράθυρο να μην απενεργοποιείται ο apache server και όποια άλλη εφαρμογή "τρέχει" εκείνη τη στιγμή. Επίσης πατώντας το κουμπί admin... που υπάρχει δίπλα από κάθε εφαρμογή, εμφανίζει το administrator menu κάθε εφαρμογής, δίνοντας μας τη δυνατότητα για περαιτέρω διαχείριση.



Εικόνα 7.1: Το control panel του XAMPP.



Apache

HTTP SERVER PROJECT

Ο Apache HTTP Server είναι μία εφαρμογή open-source, η οποία μας βοηθά να αναπτύξουμε και να συντηρήσουμε έναν open-source HTTP server για λειτουργικά συστήματα όπως το UNIX και τα WINDOWS. Ο Apache έχει συμβάλει στη ραγδαία ανάπτυξη του παγκόσμιου ιστού. Η όλη εφαρμογή μας παρέχει ένα ασφαλές και επεκτάσιμο server ο οποίος μας παρέχει HTTP υπηρεσίες, οι οποίες είναι συμβατές με τα τρέχοντα HTTP δεδομένα. Από τη στιγμή που έγινε διαθέσιμος, έχει γίνει ο πιο συχνά χρησιμοποιούμενος και δημοφιλής web server. Επιπλέον έγινε ο πρώτος εναλλακτικός server πέρα από τον Netscape web server που ήταν κατασκευασμένος πάνω σε java. Σήμερα πλέον ο Apache χρησιμοποιείται από το 58% των ιστοσελίδων.

Ο apache χρησιμοποιείται για να εξυπηρετήσει κυρίως δυναμικές σελίδες του ιστοχώρου. Πολλές εφαρμογές είναι σχεδιασμένες ώστε να προσαρμοστούν στο περιβάλλον και στα χαρακτηριστικά γνωρίσματα που ο apache παρέχει. Ο apache είναι ανακαταναεμημένος ως ένα μέρος μιας μεγάλης γκάμας από πακέτα λογισμικού, συμπεριλαμβανομένου του Oracle RDBMS ή του IBM WebSphere application server.

Επίσης χρησιμοποιείται για να πραγματοποιήσει πολλούς στόχους, όπου απαιτείται ένας ασφαλής και αξιόπιστος τρόπος χρήσης. Ένα παράδειγμα χρήσης είναι το μοίρασμα αρχείων από έναν προσωπικό υπολογιστή μέσα στο διαδίκτυο. Ο χρήστης που έχει εγκατεστημένο τον apache server στον υπολογιστή, μπορεί να εισάγει διάφορα αρχεία στο One example is sharing files from a personal computer over the Internet. A user who has Apache installed on their desktop can put arbitrary files in the Apache's document root which can then be shared.

Ακόμα πολλοί προγραμματιστές που χρησιμοποιούν διαδικτυακές εφαρμογές, χρησιμοποιούν μια έκδοση του apache για την προεπισκόπηση και τον έλεγχο του κώδικα καθώς αυτός αναπτύσσεται.

Γενικά Ο Apache HTTP γνωστός και απλά σαν Apache είναι ένας εξυπηρετητής του παγκόσμιου ιστού (web). Όταν επισκέπτεστε ένας ιστότοπο ο πλοηγός σας επικοινωνεί με έναν διακομιστή HTTP. Ο Apache είναι ένας από τους δημοφιλέστερους, εν μέρει γιατί λειτουργεί σε διάφορες πλατφόρμες σαν τα Windows, το Linux, το Unix, και το Mac OS X. Διατηρείται τώρα από μια κοινότητα ανοιχτού κώδικα με επιτήρηση από το Ίδρυμα Λογισμικού Apache (Apache Software Foundation).

Η πρώτη του έκδοση, γνωστή ως NCSA HTTPd, δημιούργησε από τον Robert McCool και κυκλοφόρησε το 1993. Θεωρείται ότι έπαιξε σημαντικό ρόλο στην αρχική επέκταση του ιστού. Ήταν η πρώτη βιώσιμη εναλλακτική επιλογή που παρουσιάστηκε απέναντι στον εξυπηρετητή http της εταιρείας Netscape και από τότε έχει εξελιχθεί στο σημείο να ανταγωνίζεται άλλους εξυπηρετητές βασισμένους στο Unix σε λειτουργικότητα και απόδοση. Από το 1996 ήταν από τους πιο δημοφιλείς όμως από τον Μάρτιο του 2006 έχει μειωθεί το ποσοστό της εγκατάστασής του κυρίως από τον Microsoft Internet Information Services και τη πλατφόρμα .NET. Τον Οκτώβριο του 2007 το μερίδιο του ήταν 47.73% από όλους τους ιστοτόπους



Η PHP, της οποίας τα αρχικά αντιπροσωπεύουν το "PHP: Hypertext Preprocessor" είναι μια ευρέως χρησιμοποιούμενη, ανοιχτού κώδικα, γενικού σκοπού scripting γλώσσα προγραμματισμού, η οποία είναι ειδικά κατάλληλη για ανάπτυξη εφαρμογών για το Web και μπορεί να ενσωματωθεί στην HTML.

Απλή απάντηση, αλλά τι σημαίνει; Ένα παράδειγμα:

Παράδειγμα. Ένα εισαγωγικό παράδειγμα

```
<html>
  <head>
    <title>Example</title>
  </head>
  <body>

    <?php
      echo "Hi, I'm a PHP script!";
    ?>

  </body>
</html>
```

Παρατηρήστε πως αυτό είναι διαφορετικό από ένα script γραμμένο σε άλλες γλώσσες προγραμματισμού όπως η Perl ή η C : Αντί να γράφετε ένα πρόγραμμα με πολλές εντολές για να εξάγετε HTML, γράφετε ένα HTML script με κάποιο ενσωματωμένο κώδικα για να κάνει κάτι (σε αυτή την περίπτωση, να εμφανίζει κάποιο κείμενο). Ο κώδικας PHP είναι εσώκλειστος σε ειδικά tags (ετικέτες) αρχής και τέλους που σας επιτρέπουν να μεταφέρεστε μέσα και έξω από το "PHP mode" (PHP τρόπο λειτουργίας).

Αυτό που διαχωρίζει την PHP από κάτι σαν client-side Javascript είναι ότι ο κώδικας εκτελείται στον server (εξυπηρετητή). Αν είχατε ένα script σαν το παραπάνω στον server σας, ο client θα έπαιρνε τα αποτελέσματα της εκτέλεσης αυτού του script, χωρίς να υπάρχει κανένας τρόπος να καταλάβει τι κώδικας υπάρχει από κάτω. Μπορείτε ακόμη να ρυθμίσετε τον web server σας να χειρίζεται όλα τα HTML αρχεία σας με την PHP, και τότε πραγματικά δεν υπάρχει τρόπος ο χρήστης να καταλάβει τι έχετε κάτω από το μανίκι σας.

Τα καλύτερο πράγμα στην PHP είναι ότι είναι εξαιρετικά απλή για ένα νεοφερμένο αλλά προσφέρει πολλές προηγμένα χαρακτηριστικά για ένα επαγγελματία προγραμματιστή. Μην τρομάζετε όταν διαβάζετε την μακροσκελή λίστα με τα χαρακτηριστικά της PHP. Μπορείτε να εξοικειωθείτε μέσα σε πολύ λίγο χρόνο και να αρχίσετε να γράφετε απλά script σε λίγες ώρες.

Αν και η ανάπτυξη της PHP εστιάζεται σε server-side scripting, μπορείτε να κάνετε πολύ περισσότερα με αυτή.

Η PHP επικεντρώνεται κυρίως στο server-side scripting, έτσι μπορείτε να κάνετε οτιδήποτε ένα άλλο CGI πρόγραμμα μπορεί να κάνει, όπως να μαζέψει δεδομένα, να παράγει δυναμικό περιεχόμενο σελίδων, ή να στείλει και να πάρει cookies. Αλλά η PHP μπορεί να κάνει πολύ περισσότερα.

Υπάρχουν τρεις κύριοι τομείς που χρησιμοποιείται ένα PHP script.

- Server-side scripting. Αυτό είναι το πιο παραδοσιακό και το κύριο πεδίο για την PHP. Χρειάζεστε τρία πράγματα για να δουλέψει αυτό. Τον PHP μεταγλωττιστή (parser) (CGI ή server module), ένα webserver (εξηγηρητή σελίδων) και ένα web browser ("φυλλομετρητή"). Πρέπει να τρέξετε τον webserver, με μια συνδεδεμένη εγκατάσταση της PHP. Μπορείτε να προσπελάσετε τα αποτελέσματα του PHP προγράμματος με ένα web browser, βλέποντας την σελίδα PHP μέσα από τον server. Για περισσότερες πληροφορίες, δείτε την παράγραφο οδηγίες εγκατάστασης.
- Command line scripting. Μπορείτε να φτιάξετε ένα PHP script για να το τρέχετε χωρίς server ή browser. Χρειάζεστε μόνο τον PHP μεταγλωττιστή για να την χρησιμοποιήσετε με αυτό τον τρόπο. Αυτός ο τύπος είναι ιδανικός για script που εκτελούνται συχνά με τη χρήση της cron (σε *nix ή Linux) ή με τον Task Scheduler (στα Windows). Αυτά τα script μπορούν επίσης να χρησιμοποιηθούν για απλές εργασίες επεξεργασίας κειμένου. Δείτε την ενότητα σχετικά με την Command line χρήση της PHP για περισσότερες πληροφορίες.
- Εγγραφή client-side GUI εφαρμογών (Γραφικά περιβάλλοντα χρηστών). Η PHP ίσως να μην είναι η πιο καλή γλώσσα για να γράψει κανείς παραθυρικές εφαρμογές, αλλά αν ξέρετε PHP πολύ καλά και θέλετε να χρησιμοποιήσετε κάποια προχωρημένα χαρακτηριστικά της PHP στις client-side εφαρμογές σας, μπορείτε επίσης να χρησιμοποιήσετε το PHP-GTK για αυτού του είδους τα προγράμματα. Έχετε επίσης τη δυνατότητα να γράφετε cross-platform εφαρμογές με αυτό τον τρόπο. Το PHP-GTK είναι μια επέκταση της PHP και δεν συμπεριλαμβάνεται στην κύρια διανομή. Αν ενδιαφέρεστε για το PHP-GTK, επισκεφτείτε » την δική του ιστοσελίδα.

Η PHP μπορεί να χρησιμοποιηθεί σε όλα τα κύρια λειτουργικά συστήματα, συμπεριλαμβανομένου του Linux, πολλών εκδοχών του Unix (HP-UX, Solaris και OpenBSD), Microsoft Windows, Mac OS X, RISC OS και πιθανώς σε άλλα. Η PHP υποστηρίζει επίσης τους Apache, Microsoft Internet Information Server, Personal Web Server, Netscape και iPlanet servers, Oreilly Website Pro server, Caudium, Xitami, OmniHTTPd, και πολλούς άλλους webserver. Για την πλειοψηφία των server η PHP έχει ένα module, για τους υπόλοιπους η PHP μπορεί να λειτουργήσει ως ένας CGI επεξεργαστής.

Έτσι με την PHP έχετε την ελευθερία επιλογής ενός λειτουργικού συστήματος και ενός web server. Επιπλέον, έχετε επίσης την ελευθερία να χρησιμοποιήσετε συναρτησιακό (procedural) ή αντικειμενοστραφή (object oriented) προγραμματισμό ή μια ανάμειξη τους. Αν και η παρούσα έκδοση δεν υποστηρίζει όλα τα πρότυπα χαρακτηριστικά, μεγάλες βιβλιοθήκες κώδικα και μεγάλες εφαρμογές (συμπεριλαμβανομένης και της βιβλιοθήκης PEAR) είναι γραμμένες μόνο με αντικειμενοστραφή κώδικα.

Με την PHP δεν είστε περιορισμένοι να εξάγετε HTML. Οι δυνατότητες της PHP συμπεριλαμβάνουν την εξαγωγή εικόνων, αρχείων PDF, ακόμη και ταινίες Flash (χρησιμοποιώντας τα libswf και Ming) παράγονται αμέσως. Μπορείτε επίσης να εξάγετε εύκολα οποιοδήποτε κείμενο όπως XHTML και οποιοδήποτε άλλο XML αρχείο. Η PHP μπορεί να δημιουργεί αυτόματα αυτά τα αρχεία και να τα αποθηκεύει στο σύστημα αρχείων, αντί να τα εκτυπώνει, αποτελώντας έτσι μια server-side cache για το δυναμικό σας περιεχόμενο.

Ένα από τα πιο δυνατά και σημαντικά χαρακτηριστικά της PHP είναι η υποστήριξη που έχει για ένα μεγάλο σύνολο βάσεων δεδομένων. Η συγγραφή μιας σελίδας που υποστηρίζει βάσεις δεδομένων είναι εξαιρετικά απλή. Οι εξής βάσεις δεδομένων υποστηρίζονται μέχρι στιγμής:

Adabas D	Ingres	Oracle (OCI7 and OCI8)
dBase	InterBase	Ovrimos
Empress	FrontBase	PostgreSQL
FilePro (read-only)	mSQL	Solid
Hyperwave	Direct MS-SQL	Sybase
IBM DB2	MySQL	Velocis
Informix	ODBC	Unix dbm

Έχουμε επίσης μια αφαιρετική επέκταση DBX βάσεων δεδομένων (DBX database abstraction extension) που σας επιτρέπει διάφανα να χρησιμοποιείτε οποιαδήποτε βάση δεδομένων υποστηρίζεται από αυτή την επέκταση. Επιπλέον η PHP υποστηρίζει το ODBC, το Open Database Connection standard (Ανοιχτό πρότυπο Σύνδεσης Βάσεων δεδομένων) έτσι μπορείτε να συνδεθείτε σε οποιαδήποτε βάση δεδομένων που υποστηρίζει αυτό το παγκόσμιο πρότυπο.

Η PHP έχει επίσης υποστήριξη για επικοινωνία με άλλες υπηρεσίες χρησιμοποιώντας πρωτόκολλα όπως LDAP, IMAP, SNMP, NNTP, POP3, HTTP, COM (στα Windows) και αμέτρητα άλλα. Μπορείτε επίσης να ανοίξετε raw network sockets και να αλληλεπιδράσετε με οποιοδήποτε άλλο πρωτόκολλο. Η PHP έχει ακόμη υποστήριξη για την περίπλοκη ανταλλαγή δεδομένων WDDX μεταξύ σχεδόν όλων των Web programming γλωσσών. Μιλώντας για δια-επικοινωνία, η PHP υποστηρίζει instantiation αντικειμένων Java και τα χρησιμοποιεί διάφανα σαν αντικείμενα PHP. Μπορείτε επίσης να χρησιμοποιήσετε την CORBA επέκταση μας για να προσπελάσετε remote (απομακρυσμένα) αντικείμενα.

Η PHP έχει εξαιρετικά χρήσιμα χαρακτηριστικά επεξεργασίας κειμένων, από την POSIX επέκταση ή τις Perl regular expressions μέχρι XML parsing αρχείων. Για τη μεταγλώττιση και την πρόσβαση αρχείων XML, υποστηρίζουμε τα πρότυπα SAX και DOM. Μπορείτε να χρησιμοποιήσετε την XSLT επέκταση μας για να μετατρέψετε τα XML αρχεία σε άλλες μορφές.

Καθώς χρησιμοποιείτε την PHP στον τομέα του ecommerce, θα βρείτε τις Cybercash payment, CyberMUT, VeriSign Payflow Pro και C CVS συναρτήσεις χρήσιμες για τα online προγράμματα πληρωμής σας.

Τελευταίο αλλά σημαντικό, έχουμε πολλές άλλες ενδιαφέρουσες επεκτάσεις, τις mhoGoSearch search engine συναρτήσεις, πολλά εργαλεία συμπίεσης (gzip, bz2), μετατροπές ημερολογίου, μεταφράσεις...



JpGraph is a Object-Oriented Graph creating library for PHP $\geq 4.3.1$ The library is completely written in PHP and ready to be used in any PHP scripts (both CGI/APXS/CLI versions of PHP are supported).

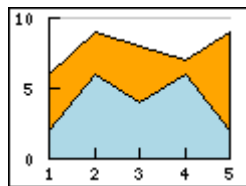
The library can be used to create numerous types of graphs either on-line or written to a file. JpGraph makes it easy to draw both "quick and dirty" graphs with a minimum of code as well as complex graphs which requires a very fine grained control. The library assigns context sensitive default values for most of the parameters which minimizes the learning curve. The features are there when you need them - not as an obstacle to overcome!

- Anti-aliasing for Pie charts.
- More advanced formatting of graph titles including 3D Bevel effects.
- Additional 3D Bevel formatting feature for the entire Graph
- Gradient fills for bars are now also displayed in the legend
- Improvements on legend formatting. It is now possible to have several columns in the legend
- Improved callback capability for Plot marks. The new callbacks will provide both X and Y coordinates.
- Footer text can now be used on all graph types
- Added possibility to use the Freely available Vera TTF fonts. These fonts are available from <http://www.gnome.org/fonts/>
- Built in rotation of graphs
- Fine tuning of legend layout, size and positioning of individual legends and markers.
- Added possibility to use object methods as callbacks and not only static global functions
- Full support for color alpha blending
- Automatic generation of client side image maps to make it possible to generate drill-down graphs.
- Advanced interpolation with cubic splines to get smooth curves from just a few data points.
- Several different fill styles for line plots
- Advanced image 3D effects built-in without external image manipulation programs
- Supports unlimited number of plots in each graph, makes it easy to compose complex graph which consists of several plot types

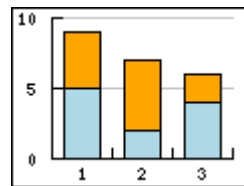
- Additional built-in images for plot marks including 3D rendered markers like diamonds, squares, bevels, balls, pins etc
- Text strings can now be added to the plot using scale coordinates
- Support for all primitive URL parameter types with CSIM graphs.
- Autoscaling will now also consider added lines and texts.
- Hare/Niemeyer Integer compensation for Pie Plots
- Added the possibility to use Vertical Gradient fill for line plots.
- Improved error handling. The visual appearance of the error handling now tries to mimic any windows system window (in graphic)

- Builtin support to display over 200 country flag and the possibility to use them as icons or markers in the graphs. All the flag images are builtin with JpGraph in an efficient pre-compiled data format.
- Supports caching (with timeout) of generated graphs to lessen burden of a HTTP server.
- Intelligent autoscaling which gravitates towards aesthetical values.
- Fully supports manual scaling, with fine grain control of position of ticks
- Supports color-gradient fill for bar graphs
- Supports both vertical and horizontal grids (with image depth specification)
- Supports both on-the fly image generation and batch processing.
- Extensive documentation
- Advanced text formatting using TTF fonts
- Support for both Chinese and Japanese character sets

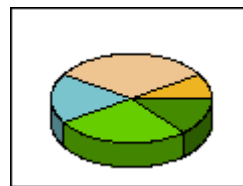
Basic supported graphs



1. Line



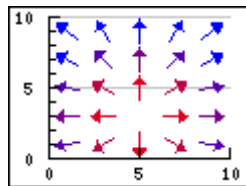
2. Bar



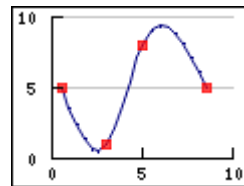
3. Pie (2D & 3D)



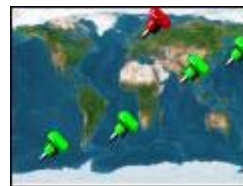
4. Ring



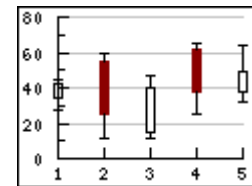
5. Field



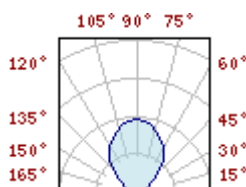
6. Splines



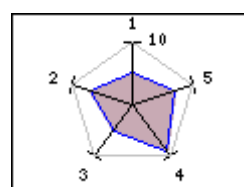
7. Geo Maps



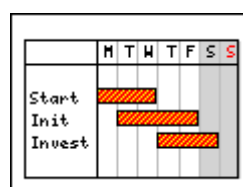
8. Stock



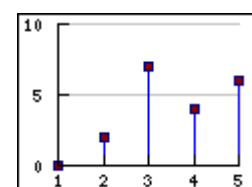
9. Polar



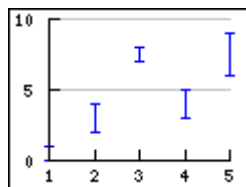
10. Radar



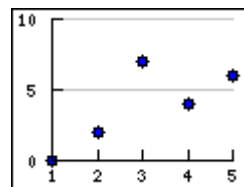
11. Gantt



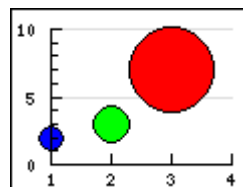
12. Impuls



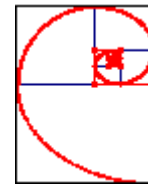
13. Error



14. Scatter



15. Balloon

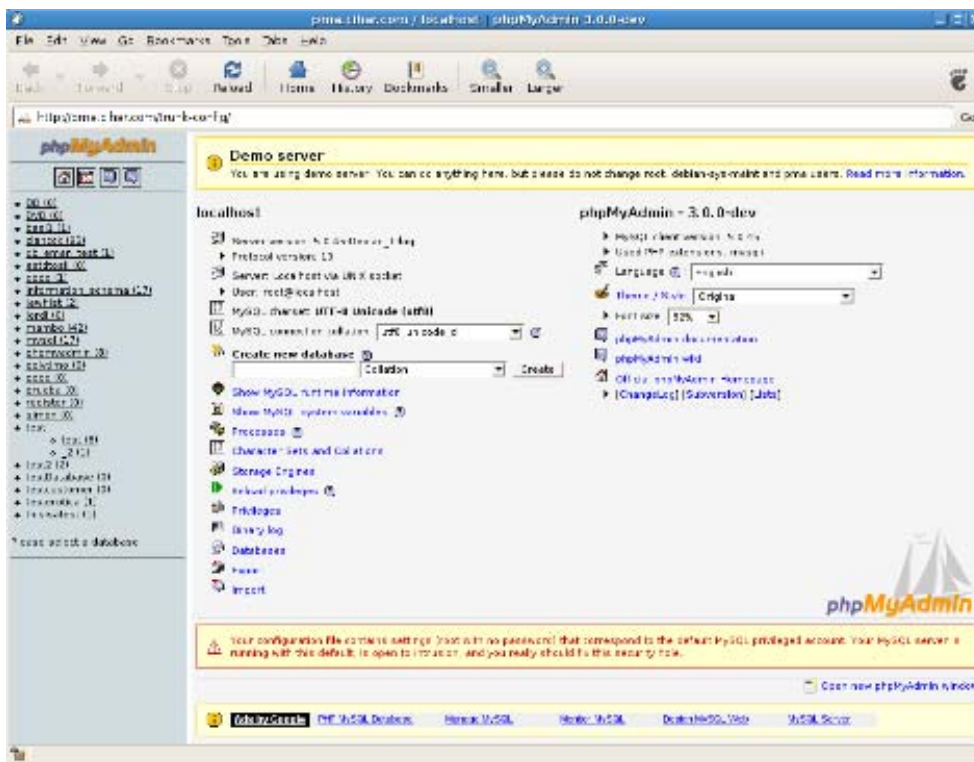


6. Canvas



Το phpMyAdmin είναι ένα εργαλείο γραμμένο σε php που μας επιτρέπει να διαχειριζόμαστε τις βάσεις δεδομένων MySQL. Μπορούμε να εκτελέσουμε ερωτήματα SQL, να εργαστούμε με fields (add, edit, delete), με tables (create, alter, drop), να δημιουργήσουμε πρόσθετες βάσεις δεδομένων και άλλα. Μπορούμε εύκολα να διαχειριστούμε τους πίνακες και τα πεδία που περιέχονται στη βάση δεδομένων μας, να αλλάξουμε τύπους πεδίων και να ενημερώσουμε τους πίνακες της βάσης μας.

Ο Tobias Ratschiller , σύμβουλος της IT και αργότερα ιδρυτής της εταιρίας Maguma , ξεκίνησε να εργάζεται βασιζόμενος σε php περιβάλλον το 1998 και όταν το 2000 εγκατέλειψε το project εξαιτίας έλλειψης χρόνου , το phpMyAdmin είχε ήδη γίνει ένα από τις πιο δημοφιλείς php εφαρμογές.



Εικόνα 7.2: Το phpMyAdmin σε λειτουργία.



Οι επιχειρήσεις προσπαθούν να έχουν οργανωμένες πληροφορίες , στατιστικά αλλά και να μπορούν να διαχειρίζονται τους πελάτες τους. Προς την κατεύθυνση αυτή βοήθησε η ανάπτυξη της MySQL , η οποία έδωσε την δυνατότητα να πραγματοποιήσουν τους στόχους αυτούς.

Η MySQL είναι το πιο δημοφιλές Open Source SQL σύστημα διαχείρισης βάσεων δεδομένων και Web εφαρμογών. Αναπτύχθηκε και υποστηρίχθηκε από το MySQL AB. Το MySQL AB είναι μια 2ης γενιάς ανοιχτού κώδικα εταιρία , η οποία ιδρύθηκε από τους δημιουργούς της MySQL. Τέλος συνεργάζεται με την php και αυτό μπορούμε να το διαπιστώσουμε σε πάρα πολλές εφαρμογές (e-shop , Wikipedia κ.τ.λ.).



About Visual Studio

The Microsoft Visual Studio development system is a suite of development tools designed to aid software developers—whether they are novices or seasoned professionals—face complex challenges and create innovative solutions. Every day, software developers break through tough problems to create software that makes a difference in the lives of others. Visual Studio's role is to improve the process of development to make the work of achieving those breakthroughs easier and more satisfying.

How Visual Studio improves the process of development:

Productive

Visual Studio-branded tools continually deliver better ways for software developers to do more with less energy wasted on repetition and drudgery. From efficient code editors, IntelliSense, Wizards, and multiple coding languages in one integrated development environment (IDE) to high-end application life-cycle management (ALM) products in Microsoft® Visual Studio® Team System. New versions of Visual Studio keep bringing innovative tools to help developers focus on solving problems, not waste time on minutiae.

Integrated

With Visual Studio, software developers benefit from an integrated product experience that spans tools, servers, and services. Visual Studio products work well together—not just with one another,

but also with other Microsoft software, such as Microsoft server products and the Microsoft Office system.

Comprehensive

Visual Studio offers a choice of tools for all phases of software development—development, testing, deployment, integration, and management—and for every kind of developer—from the novice to the skilled professional. Visual Studio is also engineered to support development across all types of devices—PCs, servers, the Web, and mobile devices.

Reliable

Visual Studio is engineered and tested to be consistently dependable, secure, interoperable, and compatible. Visual Studio offers an unmatched combination of security features, scalability, and interoperability. Although Visual Studio always incorporates forward-thinking features, it is designed to ensure backward-compatibility wherever possible.

Visual Studio and the Microsoft Application Platform

The Microsoft Application Platform is a portfolio of technology capabilities, core products, and best practice guidance focused on helping IT and development departments partner with the business to maximize opportunity.

As one of the core products of the Microsoft Application Platform, Visual Studio can help you drive the right business efficiencies, customer connections, and value-added services by providing a single, fully integrated development environment for all types of development, including Microsoft Windows, Microsoft Office, Web, and mobile applications. Use Visual Studio development solutions to give your development team powerful ways to:

- Increase productivity and quality through integrated and familiar tools.
- Deploy, secure, and support your critical Web applications and infrastructure.
- Reduce costs through better visibility of your development process.
- Provide better predictability and planning through integrated process and methodology support.

Borland C++ Builder™

Borland C++ Builder 6 combines the most powerful C++ Windows development environment for building desktop and database applications with native Internet components for building high-throughput Web applications. C++Builder increases productivity and manages the entire development cycle with over 150 reusable components, the XML-based Project Manager and a true ANSI/ISO C++ high-speed 32-bit optimizing compiler to deliver reliable applications to market faster. C++Builder includes a professional IDE, WebBroker, Native Internet Components, InterBase Express, and robust debugging tools. Use the CodeGuard run-time error detection tool to easily locate and diagnose memory and resource leaks on the spot. Understand and navigate your code and data modules with the Project Browser, Code Explorer, and Data Module Designer views. Visually build components for your business with Frames. Build and reuse COM objects and servers, including a complete suite of MS Office automation controllers.

Features:

- Rapid C++ e-business development with Web Services
- Create SOAP/XML Web Services and connections
- Build Web Services-enabled middleware
- Develop cross-platform applications
- Integrate corporate data into your applications



Το Dreamweaver MX 2004 της Macromedia αποτελεί την επιλογή των επαγγελματιών για την κατασκευή ιστοσελίδων και εφαρμογών. Συνεχίζει την παράδοση του προϊόντος στην καινοτομία παρέχοντας ένα στιβαρό θεμέλιο για ευρεία υιοθέτηση του Cascading Style Sheet (CSS), ενός ευρέως αποδεκτού web standard, το οποίο προηγουμένως ήταν δύσκολο να εφαρμοστεί. Το Dreamweaver MX 2004 αποτελεί το μοναδικό εκτενές εργαλείο για να αποδώσει γερή υποστήριξη CSS σε ένα οπτικό περιβάλλον ανάπτυξης, ώστε να παράγουν ιστοσελίδες και εφαρμογές για το web.

Το Dreamweaver MX 2004 ενσωματώνει οπτικά εργαλεία περιγράμματος, χαρακτηριστικά ανάπτυξης εφαρμογών και υποστήριξη επεξεργασίας κώδικα μέσω ενός περιβάλλοντος τεχνολογίας ανεξαρτήτου πλατφόρμας. Υποστηρίζοντας τις σημαντικότερες τεχνολογίες HTML και server, το Dreamweaver MX 2004 απελευθερώνει τη δύναμη της προσβάσιμης ανάπτυξης βασισμένης σε πρότυπα για την κοινότητα web, ενώ βελτιώνει τα βασικά εργαλεία, τα οποία χρειάζονται οι επαγγελματίες. Το Dreamweaver MX 2004 περιλαμβάνει επίσης το SecureFTP, μια δυναμική λειτουργία επικύρωσης λειτουργικότητας με οποιονδήποτε browser, ενσωματωμένη επεξεργασία γραφικών, ομαλή συνεργασία με τα Word και Excel της Microsoft και ενημερωμένη υποστήριξη για τεχνολογίες ASP.NET, PHP και ColdFusion server.

Το Dreamweaver MX το 2004 επίσης περιλαμβάνει τα MX Elements για HTML, για κατασκευή blocks εφαρμογών που επιτρέπουν στους developers να παράγουν γρήγορα εντυπωσιακά αποτελέσματα με συνεπές ύφος και αίσθηση.

Επίσης ενσωματώνει ένα βελτιωμένο περιβάλλον σχεδίου και ανάπτυξης για να βοηθήσει τους επαγγελματίες του web να βρουν ό,τι χρειάζονται γρηγορότερα, αυξάνοντας την απόδοση, την παραγωγικότητα, και την ποιότητα της εργασίας τους.


```

S40Ctl1.Decompress.Execute
End Sub

Private Sub cmdDownload_Click()
S40Ctl1.Download.ClearBufferAfter = False
S40Ctl1.Download.TargetFileName = Trim(txtSource)
StatusBar1.Panels.Item(1).Text = "Downloading: " & Trim(txtSource) & "..."
mnuConnect Click

S40Ctl1.Download.Execute
End Sub

Private Sub cmdEnSaveMode Click()
StatusBar1.Panels.Item(1).Text = "Working. Please wait..."
mnuConnect Click

S40Ctl1.Logger.EnterEnSaveMode
End Sub

Private Sub cmdPWD Click()
mnuConnect Click

S40Ctl1.Logger.Password = "123456"
S40Ctl1.Logger.EnterPassword
End Sub

Private Sub cmdLoad Click()
CommonDialog1.CancelError = True
On Error GoTo AbandonLoad
CommonDialog1.DialogTitle = "Select a Stylitis-40 setup file to load"
CommonDialog1.FileName = App.Path + "\set40.dat"
CommonDialog1.DefaultExt = ".dat"
CommonDialog1.ShowOpen
SelectedFile = CommonDialog1.FileName
If SelectedFile <> "" Then S40Ctl1.LoadFile (SelectedFile)
StatusBar1.Panels.Item(1).Text = "Setup loaded: " + SelectedFile
Me.Caption = Cap$ + SelectedFile
RefreshTable
Exit Sub
AbandonLoad:
    If Err.Number <> 0 Then MsgBox Err.Description
End Sub

Private Sub cmdSave Click()
CommonDialog1.CancelError = True
On Error GoTo AbandonSave
CommonDialog1.DialogTitle = "Select a save name for a Stylitis40 setup file"
CommonDialog1.FileName = App.Path + "\set40.dat"
CommonDialog1.DefaultExt = ".dat"
CommonDialog1.Flags = &H2 And &H4 'Prompt to overwrite, Hide read-only check box
CommonDialog1.ShowSave
SelectedFile = CommonDialog1.FileName
S40Ctl1.SaveFile (SelectedFile)
StatusBar1.Panels.Item(1).Text = "Setup saved: " + SelectedFile
Me.Caption = Cap$ + SelectedFile
Exit Sub
AbandonSave:
    If Err.Number <> 0 Then MsgBox Err.Description
End Sub

Private Sub cmdServer Click()
frmDLServer.Show

End Sub

Private Sub cmdSleep Click()
StatusBar1.Panels.Item(1).Text = "Working. Please wait..."
mnuConnect Click

S40Ctl1.Logger.Sleep

```

```

End Sub

Private Sub cmdTime Click()
StatusBar1.Panels.Item(1).Text = "Working. Please wait..."
mnuConnect_Click

S40Ctl1.Logger.EnterTime

End Sub

Private Sub cmdWrite_Click()
StatusBar1.Panels.Item(1).Text = "Working. Please wait..."
mnuConnect_Click

S40Ctl1.WriteSetup
End Sub

Private Sub cmdContMode Click()
StatusBar1.Panels.Item(1).Text = "Working. Please wait..."
mnuConnect_Click

S40Ctl1.Logger.EnterContMode
End Sub

Private Sub ComboLines Click()
S40Ctl1.ComPort.Port = ComboLines.Text

End Sub

Private Sub Form_QueryUnload(Cancel As Integer, UnloadMode As Integer)
'Prevent unloading the form when communication in progress
If S40Ctl1.ComPort.PortBusy = True Then Cancel = True

End Sub

Private Sub Image2 Click()

End Sub

Private Sub mnuAcq_Click()
mnuConnect_Click

StatusBar1.Panels.Item(1).Text = "Quering Logger..."
StatusBar1.Panels.Item(1).Text = "Logger ACQ ON: " & S40Ctl1.Logger.AcqON

End Sub

Private Sub mnuAcqOff Click()
mnuConnect_Click

StatusBar1.Panels.Item(1).Text = "Setting Logger..."
S40Ctl1.Logger.AcqON = False
StatusBar1.Panels.Item(1).Text = "Logger ACQ ON: False"

End Sub

Private Sub mnuAcqOn Click()
mnuConnect_Click

StatusBar1.Panels.Item(1).Text = "Setting Logger..."
S40Ctl1.Logger.AcqON = True
StatusBar1.Panels.Item(1).Text = "Logger ACQ ON: True"

End Sub

Private Sub mnuConnect_Click()
S40Ctl1.ComPort.PortOpen = True

If Not S40Ctl1.ComPort.PortOpen Then
    StatusBar1.Panels.Item(1).Text = S40Ctl1.ComPort.Status

```



```

Private Sub cboPortRate Click()
S40Ctl1.ComPort.BaudRate = Val(cboPortRate.Text)
End Sub
Private Sub cmdSetup_Click()
StatusBar1.Panels.Item(1).Text = "Working. Please wait..."
mnuConnect Click

S40Ctl1.ReadSetup
If S40Ctl1.OperationEvent = 0 Then RefreshTable
End Sub
Private Sub RefreshTable()
SetGrid.Clear
SetGrid.Rows = 1
SetGrid.Cols = 5
SetGrid.Row = 0
SetGrid.Col = 0: SetGrid.Text = "Channel"
SetGrid.Col = 1: SetGrid.Text = "Mode"
SetGrid.Col = 2: SetGrid.Text = "Type"
SetGrid.Col = 3: SetGrid.Text = "Slope"
SetGrid.Col = 4: SetGrid.Text = "Offset"

For i = 1 To 3
SetGrid.AddItem "C" & i & vbTab & S40Ctl1.Counter.inModeLiteral(i) & vbTab &
S40Ctl1.Counter.inTypeLiteral(i) & vbTab & S40Ctl1.Counter.inSlope(i) & vbTab &
S40Ctl1.Counter.inOffset(i)
Next i
For i = 1 To 4
SetGrid.AddItem "A" & i & vbTab & S40Ctl1.Analog.inModeLiteral(i) & vbTab &
S40Ctl1.Analog.inTypeLiteral(i) & vbTab & S40Ctl1.Analog.inSlope(i) & vbTab &
S40Ctl1.Analog.inOffset(i)
Next i

txtLoggerSite.Text = S40Ctl1.Logger.SiteName
cboMathInterval.Text = S40Ctl1.Logger.MathInterval
End Sub
Private Sub cmdStatus Click()
StatusBar1.Panels.Item(1).Text = "Working. Please wait..."
mnuConnect_Click

S40Ctl1.Logger.GetStatus
If S40Ctl1.OperationEvent > 0 Then Exit Sub
List1.Clear: List1.Font.Name = "Courier": List1.Font.Size = 10
List1.AddItem "S" & S40Ctl1.Logger.Model & " Stylitis Logger Status. "
List1.AddItem "At: " & S40Ctl1.Logger.StatusAtClientClock
List1.AddItem String$(42, ".")
List1.AddItem "Firmware status: " & Trim(Left$(S40Ctl1.Logger.StatusInfo, 24))
List1.AddItem " " & Trim(Right$(S40Ctl1.Logger.StatusInfo, 24))
List1.AddItem String$(42, ".")
List1.AddItem "Battery status: " & Trim(Left$(S40Ctl1.Logger.StatusBattery, 24))
List1.AddItem " " & Trim(Right$(S40Ctl1.Logger.StatusBattery,
24))
List1.AddItem String$(42, ".")
List1.AddItem "Card status: " & Trim(Left$(S40Ctl1.Logger.StatusCard, 24))
List1.AddItem " " & Trim(Right$(S40Ctl1.Logger.StatusCard, 24))
List1.AddItem String$(42, ".")
List1.AddItem "Buffer status: " & Trim(Left$(S40Ctl1.Logger.StatusBuffer, 24))
List1.AddItem " " & Trim(Right$(S40Ctl1.Logger.StatusBuffer, 24))
List1.AddItem String$(42, ".")
List1.AddItem "Activity status: " & Trim(Left$(S40Ctl1.Logger.StatusActivity,
24))
List1.AddItem " " & Trim(Right$(S40Ctl1.Logger.StatusActivity,
24))
List1.AddItem String$(42, ".")
List1.AddItem "Clock status: " & Trim(Left$(S40Ctl1.Logger.StatusClock, 24))
List1.AddItem " " & Trim(Right$(S40Ctl1.Logger.StatusClock, 24))
List1.AddItem String$(42, "-")
End Sub

```



```

Private Sub Form Load()

Me.Caption = Cap$ + "Untitled"
SetGrid.Rows = 1
SetGrid.Cols = 5
SetGrid.Row = 0
SetGrid.ColWidth(0) = 1000
SetGrid.ColWidth(1) = 1300
SetGrid.ColWidth(2) = 1000
SetGrid.ColWidth(3) = 900
SetGrid.ColWidth(4) = 900
DatGrid.Rows = 1
DatGrid.Cols = 2
DatGrid.Row = 0
DatGrid.ColWidth(0) = 500
DatGrid.ColWidth(1) = 1000

Label2.Caption = "ActiveX: S40Ctl v" & S40Ctl1.Version & "." & S40Ctl1.Build

cboPortRate.Clear
For k = 0 To 3: cboPortRate.AddItem (2 ^ k) * 1200: Next k
cboPortRate.Text = 9600

cboMathInterval.Clear
For k = 0 To UBound(S40Ctl1.Logger.MathIntervalOptions)
    cboMathInterval.AddItem S40Ctl1.Logger.MathIntervalOptions(k)
Next k
cboMathInterval.Text = ""

For k = 1 To 3
    ImageCombol.ComboItems.Add k, "C" & Trim(Str$(k)), "C" & Trim(Str$(k))
    ImageCombol.ComboItems("C" & Trim(Str$(k))).Image = "Stop"
Next k
For k = 1 To 4
    ImageCombol.ComboItems.Add k + 3, "A" & Trim(Str$(k)), "A" & Trim(Str$(k))
    ImageCombol.ComboItems("A" & Trim(Str$(k))).Image = "Stop"
Next k

MSChart1.ColumnCount = 0
MSChart1.Title.Text = "15 second real-time plot"
MSChart1.Plot.Axis(VtChAxisIdx).CategoryScale.LabelTick = True
MSChart1.ShowLegend = True
Timer1.Interval = 300

S40Ctl1.ErrorReportMethod = 1
S40Ctl1.ComPort.PortType = ptEthernet
S40Ctl1.ComPort.Number = 1
S40Ctl1.ComPort.BaudRate = 9600
S40Ctl1.ComPort.RemoteHost = Trim(txtIPAddr.Text)
S40Ctl1.ComPort.RemotePort = Val(txtIPPort.Text)

Dim vLine As Variant
For Each vLine In S40Ctl1.ComPort.Ports
    ComboLines.AddItem vLine
Next vLine
ComboLines.Text = S40Ctl1.ComPort.Port
S40Ctl1.ComPort.PhoneNumber = Text1.Text
S40Ctl1.ComPort.BaudRate = Val(cboPortRate.Text)

StatusBar1.Panels.Item(2).Text = "DISCONNECTED"
ProgressBar1.ZOrder
End Sub

Private Sub SetGrid DblClick()
CellText = SetGrid.TextMatrix(SetGrid.Row, SetGrid.Col)
If CellText = "" Then Exit Sub
    FRow = SetGrid.Row: FCol = SetGrid.Col
If SetGrid.Col > 0 And SetGrid.Col < 3 Then

```

```

SetGrid.RowHeight(FRow) = cboGridEdit.Height
cboGridEdit.Top = SetGrid.CellTop + SetGrid.Top
cboGridEdit.Width = SetGrid.CellWidth
cboGridEdit.Left = SetGrid.CellLeft + SetGrid.Left
cboGridEdit.Visible = True
FillCombo
cboGridEdit.Text = CellText
cboGridEdit.ZOrder
Else
SetGrid.RowHeight(FRow) = mebGridEdit.Height
mebGridEdit.Top = SetGrid.CellTop + SetGrid.Top
mebGridEdit.Width = SetGrid.CellWidth
mebGridEdit.Left = SetGrid.CellLeft + SetGrid.Left
mebGridEdit.Visible = True
mebGridEdit.Text = CellText
mebGridEdit.ZOrder
End If
End Sub
Private Sub SetGrid MouseUp(Button As Integer, Shift As Integer, X As Single, Y
As Single)
If cboGridEdit.Visible = True Then
SetGrid.TextMatrix(FRow, FCol) = cboGridEdit.Text
UpdateSettings
cboGridEdit.Visible = False
SetGrid.RowHeight(FRow) = -1
End If
If mebGridEdit.Visible = True Then
SetGrid.TextMatrix(FRow, FCol) = mebGridEdit.Text
UpdateSettings
mebGridEdit.Visible = False
SetGrid.RowHeight(FRow) = -1
End If
End Sub
Private Sub FillCombo()
cboGridEdit.Clear
If SetGrid.Row < 4 Then 'counter
If SetGrid.Col = 1 Then 'mode
opt = S40Ctl1.Counter.inModeOptions(SetGrid.Row)
For k = 0 To UBound(opt)
cboGridEdit.AddItem opt(k)
Next k
End If
If SetGrid.Col = 2 Then 'type
opt = S40Ctl1.Counter.inTypeOptions(SetGrid.Row)
For k = 0 To UBound(opt)
cboGridEdit.AddItem opt(k)
Next k
End If
Else 'analog
If SetGrid.Col = 1 Then 'mode
opt = S40Ctl1.Analog.inModeOptions(SetGrid.Row - 3)
For k = 0 To UBound(opt)
cboGridEdit.AddItem opt(k)
Next k
End If
If SetGrid.Col = 2 Then
opt = S40Ctl1.Analog.inTypeOptions(SetGrid.Row - 3)
For k = 0 To UBound(opt)
cboGridEdit.AddItem opt(k)
Next k
End If
End If
End Sub
Private Sub UpdateSettings()
If FRow < 4 Then 'counter
Select Case FCol
Case 1 'mode
S40Ctl1.Counter.inModeLiteral(FRow) = cboGridEdit.Text
Case 2 'type
S40Ctl1.Counter.inTypeLiteral(FRow) = cboGridEdit.Text

```

```

        Case 3 'slope
            S40Ctl1.Counter.inSlope(FRow) = mebGridEdit.Text
        Case 4 'offset
            S40Ctl1.Counter.inOffset(FRow) = mebGridEdit.Text
    End Select
Else 'analog
    Select Case FCol
        Case 1 'mode
            S40Ctl1.Analog.inModeLiteral(FRow - 3) = cboGridEdit.Text
        Case 2 'type
            S40Ctl1.Analog.inTypeLiteral(FRow - 3) = cboGridEdit.Text
        Case 3 'slope
            S40Ctl1.Analog.inSlope(FRow - 3) = mebGridEdit.Text
        Case 4 'offset
            S40Ctl1.Analog.inOffset(FRow - 3) = mebGridEdit.Text
    End Select
End If

End Sub
Private Sub ShowMeasurements()
    DatGrid.Clear
    DatGrid.Rows = 1
    DatGrid.Cols = 2
    DatGrid.Row = 0
    DatGrid.Col = 0: DatGrid.Text = "Ch"
    DatGrid.Col = 1: DatGrid.Text = "Data"

    For i = 1 To 3
        DatGrid.AddItem "C" & i & vbTab & S40Ctl1.Counter.inData(i)
    Next i
    For i = 1 To 4
        DatGrid.AddItem "A" & i & vbTab & S40Ctl1.Analog.inData(i)
    Next i
End Sub
Private Sub S40Ctl1_OnError()
    Beep
    Txt1$ = S40Ctl1.OperationErrorLiteral
    If S40Ctl1.Logger.Channel > 0 Then Txt1$ = Txt1$ & ", Ch=" &
    Str$(S40Ctl1.Logger.Channel)
    StatusBar1.Panels.Item(1).Text = Txt1$
    ProgressBar1.Value = 0
    ProgressBar1.Visible = False
End Sub
Private Sub S40Ctl1_OnEvent()
    Select Case S40Ctl1.OperationEvent
        Case S40EvOperationComplete
            StatusBar1.Panels.Item(1).Text = S40Ctl1.OperationEventLiteral
            ProgressBar1.Value = 0
            ProgressBar1.Visible = False
        Case S40EvBusy
            ProgressBar1.Visible = True
            ProgressBar1.Value = S40Ctl1.PercentProgress
        Case S40EvPasswordRequest
            S40Ctl1.Logger.Password = "1234"
        Case s40EvConnectionClosed
            mnuDisconnect Click
            Beep
    End Select
End Sub
Private Sub ImageCombo1_Click()
    If ImageCombo1.SelectedItem.Image = "Check" Then
        ImageCombo1.SelectedItem.Image = "Stop"
        ChaNum = ChaNum - 1
    Else
        ImageCombo1.SelectedItem.Image = "Check"
        ChaNum = ChaNum + 1
    End If
    Label7.Caption = "# of channels selected: " & ChaNum
    ReDim arrData(14, ChaNum + 1)
    ReDim arrCol(ChaNum + 1)

```

```

End Sub
Private Sub cmdPlot_Click()
If cmdPlot.Caption = "Stop Plot" Then
    cmdPlot.Caption = "Plot Data"
    cmdData.Enabled = True
Else
    If ChaNum = 0 Then
        StatusBar1.Panels.Item(1).Text = "Select a channel to plot": Beep
        Exit Sub
    End If
    mnuConnect_Click

    Timer1.Enabled = True
    cmdPlot.Caption = "Stop Plot"
    cmdData.Enabled = False
End If
End Sub
Private Sub Timer1_Timer()
If cmdPlot.Caption = "Plot Data" Then
    Timer1.Enabled = False
    Exit Sub
End If

Dim Column As Integer, j As Integer
S40Ctl1.ReadMeasurement
With MSChart1
    For i = 0 To 13
        arrData(i, 1) = arrData(i + 1, 1) 'Labels set in first series.
        For j = 1 To ChaNum
            arrData(i, j + 1) = arrData(i + 1, j + 1)
        Next j
    Next i
    arrData(14, 1) = Time
    k = 0
    For j = 1 To 3
        If ImageCombo1.ComboItems("C" & Trim(Str$(j))).Image = "Check" Then
            k = k + 1: arrData(14, k + 1) = S40Ctl1.Counter.inData(j)
            arrCol(k) = "C" & Trim(Str$(j))
        End If
    Next j
    For j = 1 To 4
        If ImageCombo1.ComboItems("A" & Trim(Str$(j))).Image = "Check" Then
            k = k + 1: arrData(14, k + 1) = S40Ctl1.Analog.inData(j)
            arrCol(k) = "A" & Trim(Str$(j))
        End If
    Next j
    If ChaNum = 0 Then Exit Sub
    .ChartData = arrData
    For j = 1 To ChaNum: .Column = j: .ColumnLabel = arrCol(j): Next j
End With
End Sub
Private Sub cmdConfiguration_Click()
S40Ctl1.ComPort.Port = Trim(ComboLines.Text)
S40Ctl1.ComPort.Config
End Sub

```

8.1.2 Ο κώδικας της εφαρμογής read.exe γραμμένος σε C.

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <iostream>
#include <fstream>
#ifdef __BORLANDC__
    #pragma hdrstop
#endif
#ifdef __BORLANDC__
    #pragma argsused
#endif

using namespace std;
int main(int argc, char* argv[])
{
    FILE      *Stream;
    int        nLongTable = 0;
    long       lLength, lFileLength, lLineLen, lPrintIdx, lTable[2],
lFloatTable = 1;
    char       *ptrArray = NULL, *ptrToken= NULL, *ptrTokenDivider, st_r;
    float      *ptrFloat   = NULL;
    long tes;
    if((Stream=fopen("stylOut.txt","rb"))==NULL)
        printf(" Cannot open file for read - %s\n",argv[1]);
    else
    {
        /* Find file length the ANSI-C way ... */
        if(fseek(Stream,0,SEEK_END)!=0)
            printf(" Cannot seek to EOF - %s\n",argv[1]);
        else
            if((lLength=ftell(Stream))===-1L)
                printf(" Cannot find file length - %s\n",argv[1]);
            else
            {
                /* Find last line length (line should have a leading \n) */
                lFileLength = lLength;

                for(;lLength>0;lLength--)
                {
                    static int nNewLineCount = 0,
                        nChar = 0;

                    fseek(Stream,lLength,SEEK_SET);

                    if((nChar = fgetc(Stream))=='\n')
                        if(++nNewLineCount>1)
                            break;
                }

                lLineLen = lFileLength - lLength;

                if((ptrArray=(char*)calloc(lLineLen+1,sizeof(char)))==NULL)
                    printf(" Not enough memory!\n");
                else
                {
                    /* Load Line */
                    if(fseek(Stream,lLength,SEEK_SET)!=0)
                        printf(" Cannot seek to Line offset - %s\n",argv[1]);
                    else
                    {
```

```

        if(fread(ptrArray,lLineLen,1,Stream)!=1)
            printf(" Cannot read line to memory - %s\n",argv[1]);
        else
        {
            /*
            the first
            two entries are integers!
            */
            if((ptrToken = strtok(ptrArray," "))!=NULL) /*
version 2:" "->," (bug-fix) */
            {
                do{
                    /* Store Integer into Long table */
                    if(nLongTable<2)
                    {
                        lTable[nLongTable++] = atoi(ptrToken);
                        continue;
                    }

                    /*
0.217
                    (if you use/support setlocale this may
become redudant :-D)
                    */
                    ptrTokenDivider =
(char*)memchr(ptrToken,',',strlen(ptrToken));
                    if(ptrTokenDivider!=NULL)
                        ptrToken[ptrTokenDivider-ptrToken] = '.';

                    /* Store float into Float table */
                    ptrFloat =
(float*)realloc(ptrFloat,lFloatTable*sizeof(float));
                    memset(&ptrFloat[lFloatTable-
1],0,sizeof(float));

                    ptrFloat[lFloatTable-1] = atof(ptrToken);

                    lFloatTable++;
                }while((ptrToken=strtok(NULL," "))!=NULL);
            }

            /* Print integer (long) table */
            //printf(" Integer Table:\n");
            //for(lPrintIdx=0;lPrintIdx<nLongTable;lPrintIdx++)
            // printf("[%d] =
%d\n",lPrintIdx,lTable[lPrintIdx]);
            // printf("\n");

            /* Print float table with up to 2 digits accuracy
*/
            // printf(" Float Table:\n");
            // for(lPrintIdx=0;lPrintIdx<lFloatTable-
1;lPrintIdx++)
                // printf("[%d] =
%.3f\n",lPrintIdx,ptrFloat[lPrintIdx]);
        }
    }
}

```

```

tes= lTable[1] *100;
string st_r = (string) lTable[0];

ofstream myfile;
myfile.open ("out.txt");
    myfile << lTable[0] <<"\t" << tes <<"\t" << ptrFloat[0] <<"\t"
<< ptrFloat[1] <<"\t" << ptrFloat[2] <<"\t" << ptrFloat[3] <<"\t" <<
ptrFloat[4] <<"\t" << ptrFloat[5] <<"\t" << ptrFloat[6] <<"\t" <<
ptrFloat[7] <<"\t" << ptrFloat[8] <<"\t" << ptrFloat[9] <<"\t" <<
ptrFloat[10] <<"\t" << ptrFloat[11] <<"\t" << ptrFloat[12] <<"\t" <<
ptrFloat[13] <<"\t" << ptrFloat[14] <<"\t" << ptrFloat[15] <<"\t" <<
ptrFloat[16] <<"\t" << ptrFloat[17] <<"\t" << ptrFloat[18] <<"\t" <<
ptrFloat[19] <<"\t" << ptrFloat[20] <<"\t" << ptrFloat[21] <<"\t" <<
ptrFloat[22] <<"\t" << ptrFloat[23] <<"\t" << ptrFloat[24] <<"\t" <<
ptrFloat[25] <<"\t" << ptrFloat[26] <<"\t" << ptrFloat[27];
//myfile << printf("%d  %d",lTable[0], lTable[1]);
myfile.close();

free(ptrArray);
free(ptrFloat);
fclose(Stream);
std::remove("stylOut.txt");
std::remove("Downtest.stb");
//printf(" Press Enter to quit...");
//fgetc(stdin);

return 0;

```

8.1.3 Ο κώδικας VB της κύριας εφαρμογής DialXer.exe

```
Public lTimerID As Long
Public objComport As ACOMPORTLib.ComPort

Private Sub buttonNext_Click()
    'textReceived = ""
    textSend2 = "3"
    objComport.WriteString (textSend2)

    GetResult
End Sub

Private Sub buttonSubmit_Click()
    textReceived = ""
    objComport.WriteString (textSend)

    GetResult
End Sub

Private Sub buttonView_Click()
    If FileExists(textLogfile.Text) = True Then
        Shell "notepad " + textLogfile.Text, vbNormalFocus
    End If
End Sub

Private Sub CheckDTR_Click()

    objComport.RaiseDTR (CheckDTR.Value)
    GetResult

End Sub

Private Sub CheckRTS_Click()

    objComport.RaiseRTS (CheckRTS.Value)
    GetResult

End Sub

Public Function FileExists(sFileName As String) As Boolean
    FileExists = CBool(Len(Dir$(sFileName))) And CBool(Len(sFileName))
End Function

Private Sub Command1_Click()

End Sub

Private Sub Form_Load()
    textSend = "ATDT6977487104"

    Set objComport = CreateObject("ActiveXperts.ComPort")

    objComport.ComTimeout = 100

    For i = 0 To objComport.GetDeviceCount() - 1
        comboDevice.AddItem (objComport.GetDevice(i))
    Next

    'comboDevice.AddItem "COM1"
    'comboDevice.AddItem "COM2"
```



```

'comboDevice.AddItem "COM3"
'comboDevice.AddItem "COM4"
'comboDevice.AddItem "COM5"
' comboDevice.AddItem "COM6"
'comboDevice.AddItem "COM7"
'comboDevice.AddItem "COM8"

comboDevice.ListIndex = 0

comboSpeed.AddItem "Default"
comboSpeed.AddItem "110"
comboSpeed.AddItem "300"
comboSpeed.AddItem "600"
comboSpeed.AddItem "1200"
comboSpeed.AddItem "2400"
comboSpeed.AddItem "4800"
comboSpeed.AddItem "9600"
comboSpeed.AddItem "14400"
comboSpeed.AddItem "19200"
comboSpeed.AddItem "38400"
comboSpeed.AddItem "57600"
comboSpeed.AddItem "64000"
comboSpeed.AddItem "115200"
comboSpeed.AddItem "128000"
comboSpeed.AddItem "256000"

comboSpeed.ListIndex = 0

comboHWFlowControl.AddItem "Default"
comboHWFlowControl.AddItem "Disable"
comboHWFlowControl.AddItem "Enable"

comboHWFlowControl.ListIndex = 0

comboSWFlowControl.AddItem "Default"
comboSWFlowControl.AddItem "Disable"
comboSWFlowControl.AddItem "Enable"

comboSWFlowControl.ListIndex = 0

comboDataFormat.AddItem "Default"
comboDataFormat.AddItem "8,n,1"
comboDataFormat.AddItem "7,e,1"

comboDataFormat.ListIndex = 0

textReceived = ""

EnableControls

End Sub

Private Sub buttonOPEN_Click()

    objComport.Device = comboDevice.List(comboDevice.ListIndex)

    If (comboSpeed.Text = "Default") Then
        objComport.BaudRate = 0
    Else
        objComport.BaudRate = comboSpeed.Text
    End If

```

```

objComport.LogFile = textLogfile.Text

objComport.HardwareFlowControl = comboHWFlowControl.ListIndex
objComport.SoftwareFlowControl = comboSWFlowControl.ListIndex

If (comboDataFormat.ListIndex = 0) Then
    objComport.DataBits = objComport.asDATABITS_DEFAULT
    objComport.StopBits = objComport.asSTOPBITS_DEFAULT
    objComport.Parity = objComport.asPARITY_DEFAULT
End If

If (comboDataFormat.ListIndex = 1) Then
    objComport.DataBits = objComport.asDATABITS_8
    objComport.StopBits = objComport.asSTOPBITS_1
    objComport.Parity = objComport.asPARITY_NONE
End If

If (comboDataFormat.ListIndex = 2) Then
    objComport.DataBits = objComport.asDATABITS_7
    objComport.StopBits = objComport.asSTOPBITS_1
    objComport.Parity = objComport.asPARITY_EVEN
End If

objComport.Open

getResult

EnableControls

End Sub

Private Sub buttonClose_Click()
    objComport.Close

    getResult

    EnableControls
End Sub

Private Sub Form_Unload(Cancel As Integer)
    Timer1.Enabled = False
End Sub

Private Sub EnableControls()

    Dim bOpened

    bOpened = objComport.IsOpened

    CheckDTR.Enabled = bOpened
    CheckRTS.Enabled = bOpened
    checkCTS.Enabled = bOpened
    checkDCD.Enabled = bOpened
    checkRI.Enabled = bOpened
    checkDSR.Enabled = bOpened
    buttonOpen.Enabled = bOpened + 1
    buttonClose.Enabled = bOpened
    buttonSubmit.Enabled = bOpened

```

```

        Timer1.Enabled = bOpened

End Sub

Private Sub linkErrorCodes_Click()
    Shell "Explorer
http://www.activexperts.com/support/errorcodes/Index.asp"
End Sub

Private Sub linkWebsite_Click()
    Shell "Explorer http://www.activexperts.com"
End Sub

Private Sub textSend_Change()

End Sub

Private Sub Timer1_Timer()
    Dim strString

    strString = objComport.ReadString

    If (strString <> "") Then
        textReceived = textReceived & strString & vbCrLf
    End If
    checkDCD.Value = Abs(objComport.QueryDCD)
    checkCTS.Value = Abs(objComport.QueryCTS)
    checkDSR.Value = Abs(objComport.QueryDSR)
    checkRI.Value = Abs(objComport.QueryRI)

End Sub

Private Sub GetResult()
    If objComport.LastError = 0 Then
        textResult.Caption = "SUCCESS"
    Else
        textResult.Caption = "ERROR " & objComport.LastError & " ( " &
objComport.GetErrorDescription(objComport.LastError) & " )"
    End If
End Sub

```

8.2 Παράρτημα II- Κώδικες Γραφημάτων

8.2.1 Ο κώδικας του γραφήματος Ταχύτητας Ανέμου μαζί με το αρχείο query της βάση.

```
<?php
include ("connection.php");

$max_Speed=0;
$max_PYRANOM=0;
$max_degC=0;

$min_Speed=50;
$min_PYRANOM=50;
$min_degC=50;

$sqlquery = "SELECT AA FROM data";

$result = mysql_query($sqlquery);
$num=mysql_numrows($result);

//mysql_close();

$i=$num-1;

$aa=mysql_result($result,$i,"AA");

$aa=$aa-145;

$sqlquery = "SELECT Date, time,Speed, PYRANOM, degC FROM data WHERE
AA>'$aa'";

$result = mysql_query($sqlquery);
$num=mysql_numrows($result);
mysql_close();

if ( $num > 0 ){

$i=0;
while ($i < $num){

$Date[$i]=mysql_result($result,$i,"Date");
$time[$i]=mysql_result($result,$i,"time");

list($hour,$min) = explode(":",$time[$i]);
$array = array($hour, $min);
$time[$i] = implode(":", $array);

$Speed[$i]=mysql_result($result,$i,"Speed");
$av_Speed+=$Speed[$i];
$PYRANOM[$i]=mysql_result($result,$i,"PYRANOM");
$av_PYRANOM+=$PYRANOM[$i];
```

```

$degC[$i]=mysql_result($result,$i,"degC");
$sav_degC+=$degC[$i];

if ($Speed[$i]>=$max_Speed)
    $max_Speed=$Speed[$i];

if ($PYRANOM[$i]>=$max_PYRANOM)
    $max_PYRANOM=$PYRANOM[$i];

if ($degC[$i]>=$max_degC)
    $max_degC=$degC[$i];

if ($Speed[$i]<$min_Speed)
    $min_Speed=$Speed[$i];

if ($PYRANOM[$i]<$min_PYRANOM)
    $min_PYRANOM=$PYRANOM[$i];

if ($degC[$i]<$min_degC)
    $min_degC=$degC[$i];

$i++;
}
}

$sav_Speed=$sav_Speed/$i;
$sav_PYRANOM=$sav_PYRANOM/$i;
$sav_degC=$sav_degC/$i;

?>

```

```

<?php
include ("../jpggraph.php");
include ("../jpggraph_line.php");
include ("../jpggraph_scatter.php");
include ("../jpggraph_bar.php");

include ("sqlall.php");

// New graph with a drop shadow
$graph = new Graph(500,200);
//$graph->SetShadow();

// Use a "text" X-scale
$graph->SetScale("textlin");
$graph->SetMarginColor('white');

// Specify X-labels
$graph->xaxis->SetFont(FF_VERDANA,FS_NORMAL);
$graph->xaxis->SetTickLabels($time);
$graph->xaxis->SetTextLabelInterval(23);

```

```

$graph->xgrid->Show();

// Set title and subtitle
$graph->title->Set("Wind Speed for the last 24 hours.");
$graph->title->SetColor('blue');
// Use built in font
$graph->title->SetFont(FF_VERDANA,FS_BOLD);

$graph->xaxis->title->Set('Hour (GMT)');
$graph->xaxis->title->SetFont(FF_VERDANA,FS_BOLD);
$graph->xaxis->title->SetColor('black');

$graph->yaxis->title->Set('m/s');
$graph->yaxis->title->SetFont(FF_VERDANA,FS_BOLD);
$graph->yaxis->title->SetColor('black');

// Create the bar plot
$b1 = new BarPlot($Speed);
//$b1->SetLegend("Temperature");
$b1->SetAbsWidth(4);
$b1->SetColor('blue');
//$b1->SetShadow();

$txt2=new Text("http://meteo.epp.teiher.gr");
$txt2->Pos(10,182);
$txt2->SetColor("blue");
$graph->AddText($txt2);

$av_Speed=round($av_Speed, 1);
$max_Speed=round($max_Speed, 1);
$min_Speed=round($min_Speed, 1);

// Create and add a new text
$txt=new Text("MAX:$max_Speed - MIN: $min_Speed - AV: $av_Speed");
$txt->Pos(60,25);
$txt->SetFont(FF_VERDANA,FS_BOLD);
$txt->SetBox('yellow','navy');
$txt->SetColor("red");
$graph->AddText($txt);

// The order the plots are added determines who's on top
$graph->Add($b1);

// Finally output the image
$graph->Stroke();

?>

```

8.2.2 Ο κώδικας του συνδυαστικού γραφήματος Ταχύτητας Ανέμου και Θερμοκρασίας

```
<?php
include ("../jpgraph.php");
include ("../jpgraph_line.php");

include ("sqlall.php");

// Create the graph. These two calls are always required
$graph = new Graph(500,200,"auto");
$graph->img->SetMargin(40,40,20,40);
$graph->SetMarginColor('white');
$graph->SetScale("textlin");
$graph->SetY2Scale("lin");

// Create the linear plot
$lineplot=new LinePlot($degC);
$lineplot2=new LinePlot($Speed);

// Add the plot to the graph
$graph->Add($lineplot);
$graph->AddY2($lineplot2);

$lineplot2->SetWeight(2);
$graph->y2axis->SetColor("blue");

$graph->xgrid->Show();

$graph->title->Set("Combined Temperature (C) and Wind Speed (m/s).");

//$graph->yaxis->title->Set("Y-title");
$lineplot->SetLegend("Temperature");
$lineplot2->SetLegend("Wind Speed");
$graph->legend->SetLayout(LEGEND_HOR);
$graph->legend->Pos(0.35,0.3,"center","bottom");
$graph->legend->SetFillColor('navy@0.9');
$graph->legend->SetShadow(false);

$graph->title->SetFont(FF_VERDANA,FS_BOLD);
$graph->yaxis->title->SetFont(FF_VERDANA,FS_BOLD);

$graph->xaxis->title->Set('Hour (GMT)');
$graph->xaxis->title->SetFont(FF_VERDANA,FS_BOLD);
$graph->xaxis->title->SetColor('black');
$graph->xaxis->SetTickLabels($time);
$graph->xaxis->SetFont(FF_VERDANA);
$graph->xaxis->SetTextLabelInterval(23);

$txt=new Text("http://meteo.epp.teiher.gr");
$txt->Pos(10,182);
$txt->SetColor("blue");
$graph->AddText($txt);

$lineplot->SetColor("red");
$lineplot->SetWeight(2);

$lineplot2->SetColor("blue");
```

```

$lineplot2->SetWeight(2);

$graph->yaxis->SetColor("red");

// Display the graph
$graph->Stroke();
?>

```

8.2.3 Ο κώδικας του Στατιστικού γραφήματος Διεύθυνσης Ανέμου μαζί με το αρχείο query της βάση

```

<?php
include ("connection.php");

$Ss=0;

$sqlquery = "SELECT AA FROM data";

$result = mysql_query($sqlquery);
$num=mysql_numrows($result);

//mysql_close();

$i=$num-1;

$aa=mysql_result($result,$i,"AA");

$aa=$aa-145;

$sqlquery = "SELECT * FROM data WHERE AA>'$aa'";

$result = mysql_query($sqlquery);
$num=mysql_numrows($result);
mysql_close();

if ( $num > 0 ){

$i=0;
while ($i < $num){

$WindDir[$i]=mysql_result($result,$i,"WindDir");

if ($WindDir[$i]<11.25  && $WindDir[$i]>=348.75)
    $N++;
else if ($WindDir[$i]<348.75  && $WindDir[$i]>=326.25)
    $NWW++;
else if ($WindDir[$i]<326.25  && $WindDir[$i]>=303.75)
    $NW++;
else if ($WindDir[$i]<303.75  && $WindDir[$i]>=281.25)
    $NNW++;
else if ($WindDir[$i]<281.25  && $WindDir[$i]>=258.75)
    $W++;
else if ($WindDir[$i]<258.75  && $WindDir[$i]>=236.25)
    $SWW++;
else if ($WindDir[$i]<236.25  && $WindDir[$i]>=213.75)
    $SW++;
else if ($WindDir[$i]<213.75  && $WindDir[$i]>=191.25)

```



```

    $SSW++;
else if ($WindDir[$i]<191.25 && $WindDir[$i]>=168.75)
    $S++;
else if ($WindDir[$i]<168.75 && $WindDir[$i]>=146.25)
    $SSE++;
else if ($WindDir[$i]<146.25 && $WindDir[$i]>=123.75)
    $SE++;
else if ($WindDir[$i]<123.75 && $WindDir[$i]>=101.25)
    $SEE++;
else if ($WindDir[$i]<101.25 && $WindDir[$i]>=78.75)
    $E++;
else if ($WindDir[$i]<78.75 && $WindDir[$i]>=56.25)
    $EWE++;
else if ($WindDir[$i]<56.25 && $WindDir[$i]>=33.75)
    $NE++;
else if ($WindDir[$i]<33.75 && $WindDir[$i]>=22.25)
    $NEN++;

$i++;
}
}
?>

```

```

<?php
include ("../jpggraph.php");
include ("../jpggraph_radar.php");
include ("sqlallWD.php");

$titles=array("N", '|', "NW", '|', "W", '|', "SW", '|', 'S', '|', "SE", '|',
"E", '|', "NE", '|');
$data=array($N, $NWW, $NW, $NNW, $W, $Sww, $SW, $SSW, $S, $SSE, $SE,
$SEE, $E, $EWE, $NE, $NEN);

$graph = new RadarGraph (250,270,"auto");

$graph->title->Set("Wind Direction");
$graph->title->SetFont(FF_VERDANA,FS_NORMAL,12);

$graph->subtitle->Set("(last 24 hours.)");
$graph->subtitle->SetFont(FF_VERDANA,FS_ITALIC,10);

$txt2=new Text("http://meteo.epp.teiher.gr");
$txt2->Pos(8,254);
$txt2->SetColor("white");
$graph->AddText($txt2);

$graph->SetTitles($titles);
$graph->SetCenter(0.5,0.55);
$graph->HideTickMarks();
$graph->SetColor('lightblue');
$graph->axis->SetColor('darkgray@0.3');
$graph->grid->SetColor('darkgray@0.3');
$graph->grid->Show();

```

```
$graph->SetGridDepth(DEPTH_BACK);  
  
$plot = new RadarPlot($data);  
$plot->SetColor('red@0.2');  
$plot->SetLineWeight(1);  
$plot->SetFillColor('red@0.7');  
$graph->Add($plot);  
$graph->Stroke();  
?>
```

9 Βιβλιογραφία- References

Παρακάτω αναφέρεται η βιβλιογραφία στην οποία βασιστήκαμε για να υλοποιήσουμε την πτυχιακή εργασία, καθώς επίσης και τις διάφορες ηλεκτρονικές διευθύνσεις.

ΕΠΕΞΕΡΓΑΣΙΑ ΜΕΤΕΩΡΟΛΟΓΙΚΩΝ ΔΕΔΟΜΕΝΩΝ ΚΑΙ ON-LINE ΠΑΡΟΥΣΙΑΣΗ ΣΤΟ ΔΙΑΔΙΚΤΥΟ, Ανδρέας Βλησίδης ,Κωνσταντίνος Μιχαήλ, Γιώργος Γούναρης, Annual Conference on Telecommunications & Multimedia- TEMU 2005 , ISBN:960-88785-1-9

WEATHER STATIONS DISTRIBUTED MODEL FOR DATA PROCEES AND ON-LINE INTERNET PRESENTATION Georgios Gounaris, Andreas Vlissidis, Konstantinos Michail, Annual Conference on Telecommunications & Multimedia- TEMU 2006

Ανάπτυξη Web Εφαρμογών με PHP & MYSQL, WELLING 2002 ΓΚΙΟΥΡΔΑΣ

PHP Graphics Handbook, Allan Kent, Mitja Slenc, Jason E. Sweat 2003 Wronx Press Ltd
My SQL Tutorial, Laura,Thomson, Robert L.,Lineberry, Luke,Welling, Pearson Education (US), 2003

Programming Microsoft Visual Basic 6.0, Francesco Balena, Microsoft Press

Visual Basic 6 Complete, Steve,Sybex, John Wiley and Sons Ltd 1999

Java Network Programming, Elliotte Rusty,Harold, O'Reilly Media, Inc, USA, 2004

Programming PHP, Rasmus,Lerdorf, Kevin,Tatroe,Peter, MacIntyre O'Reilly Media, Inc, USA, 2006

PHP Manual Aug 2007 από <http://www.php.net/manual/el/>

Γενική περιγραφή των Πρωτοκόλλων TCP/IP

από http://web.teipir.gr/new/ecs/pelab_1/tcp/inter2.htm#LABEL2.1

About Visual Studio

από <http://msdn2.microsoft.com/ens/vstudio/products/bb931214.aspx>

About Borland C++ Builder 6 από <http://www.borland.com/>

GSM απο <http://el.wikipedia.org/wiki/GSM>

JpGraph από <http://www.aditus.nu/jpgraph/index.php>